



OUR PROJECT:

ARTECH SYSTEMS



GALLERY ADMINISTRATION AND BOOKING SYSTEM



MEET OUR TEAM



TEAM MEMBER (L TO R)	STUDENT NUMBER	EMAIL	CONTACT NUMBER
Dean Botha	u18002197	u18002197@tuks.co.za	082 381 7663
Trishla Pathak	u18198644	u18198644@tuks.co.za	074 613 2732
Reagile Matloga	u18201696	u18201696@tuks.co.za	066 153 9415
Gamuchirai Dambanjera	u17272522	u17272522@tuks.co.za	082 071 7209
Reinhardt Smit	u18131451	u18131451@tuks.co.za	061 498 8472



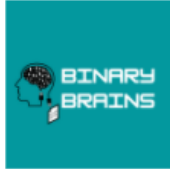
FUNCTIONAL SPECIFICATIONS



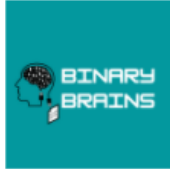
This Document is our Functional Specifications (Deliverable 2) document. The document contains more in-depth information on our system Artech Systems. The document includes Use Case Diagrams, Context Diagrams, Use Case Narratives, High to Low-level Data Flow Diagrams, Activity Diagrams, detailed Validation table as well as detailed descriptions of our interfaces involving input & output as well as report documentation for each use case.

Table of Contents

1. Document Introduction	10
2. Use Cases	11
Introduction	11
2.1 Use Case Diagram	12
2.1.1 Administration Subsystem	12
.....	12
2.1.2 Art Class Subsystem	17
2.1.3 Reporting Subsystem	18
2.1.4 User Subsystem.....	19
2.2.5 Artist Subsystem	20
2.2 Use Case Narratives	21
2.2.1 Administration Subsystem	21
2.2.2 Art Class Subsystem	84
2.2.3 Report Subsystem	97
2.2.4 User Subsystem.....	103
2.3.5 Artist Subsystem	116
Conclusion.....	126
3.Process Models of the New Proposed System	127
Introduction	127
3.1 Context Diagram	128
3.2 Functional Decomposition Diagram.....	140
3.2.1 Administration Subsystem	140
3.2.2 Art Class Subsystem	142
3.2.3 Reporting Subsystem	143



3.2.4 User Subsystem.....	143
3.2.5 User Subsystem.....	144
3.3 Data Flow Diagrams (High to Low).....	145
3.3.1 High Level Data Flow Diagrams.....	145
3.3.1.1 Administration Subsystem	145
3.3.1.2 Art Class Subsystem	154
3.3.1.3 Reporting Subsystem	156
3.3.1.4 User Subsystem.....	157
3.3.1.5 Artist Subsystem	158
3.3.2 Middle Level Data Flow Diagrams.....	159
3.3.2.1 Administration Subsystem	159
3.3.2.2 Art Class Subsystem	186
3.3.2.3 Reporting Subsystem	190
3.3.2.4 User Subsystem.....	192
3.3.2.5 Artist Subsystem	196
3.3.3 Low Level Data Flow Diagrams	198
3.3.3.1 Administration Subsystem	199
3.3.3.2 Art Class Subsystem	239
3.3.3.3 Reporting Subsystem	248
3.3.3.4 User Subsystem.....	254
3.3.3.5 Artist Subsystem	261
3.4 Data Dictionary	268
Conclusion.....	275
4. UML Modeling.....	276
Introduction	276
4.1 Activity Diagrams	277
4.1.1 Administration Subsystem	277
4.1.2 Art Class Subsystem	318
4.1.3 Reporting Subsystem	326
4.1.4 User Subsystem.....	330
4.1.4 Artist Subsystem	338
Conclusion.....	346
5. Data Model	347



Introduction	347
5.1 Logical Data Model (ERD).....	348
Conclusion.....	349
6. Interfaces and Other Inputs	350
Introduction	350
6.1 Input Interface Description	351
Conclusion.....	424
7. Reports & Other Outputs	425
Introduction	425
7.1 Output Description	425
Conclusion.....	501
8. Validation	502
Introduction	502
8.1 Validation of functional specifications against requirements	502
Conclusion.....	522
9. Sign off by Client	523
10. Complexity	524
11. General.....	527
Introduction	527
11.1 Team Sign-Off.....	527
Conclusion.....	527
12. Document Conclusion	528

Table of Figures

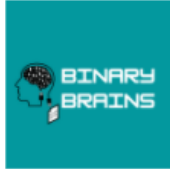
Figure 1: Use Case Diagram – Administration subsystem	12
Figure 2: Use Case Diagram – Administration subsystem	13
Figure 3: Use Case Diagram – Administration subsystem	14
Figure 4: Use Case Diagram – Administration subsystem	15
Figure 5: Use Case Diagram – Administration subsystem	16
Figure 6: Use Case Diagram – Art Class subsystem	17
Figure 7: Use Case Diagram – Reporting subsystem.....	18
Figure 8: Use Case Diagram – User subsystem	19
Figure 9: Use Case Diagram – Artist subsystem.....	20
Figure 10: Context Diagram part 1 – Subsystem 1.....	128
Figure 11: Context Diagram part 2 – Subsystem 1.....	129
Figure 12: Context Diagram part 3 – Subsystem 1.....	130
Figure 13: Context Diagram part 4 – Subsystem 1.....	131
Figure 14: Context Diagram part 5 – Subsystem 1.....	132
Figure 15: Context Diagram part 6– Subsystem 1	133
Figure 16: Context Diagram part 7 – Subsystem 1.....	134
Figure 17: Context Diagram part 8 – Subsystem 1.....	135
Figure 18: Context Diagram part 9 – Subsystem 2.....	136
Figure 19: Context Diagram part 10 – Subsystem 3	137
Figure 20: Context Diagram part 11 – Subsystem 4	138
Figure 21: Context Diagram part 12 – Subsystem 5	139
Figure 22: Decomposition Diagram part 1 – Subsystem 1.....	140
Figure 23: Decomposition Diagram part 2 – Subsystem 1.....	140
Figure 24: Decomposition Diagram part 3 – Subsystem 1.....	141
Figure 25: Decomposition Diagram part 4 – Subsystem 1.....	141
Figure 26: Decomposition Diagram part 5 – Subsystem 1.....	142
Figure 27: Decomposition Diagram part 6 – Subsystem 2.....	142
Figure 28: Decomposition Diagram part 7 – Subsystem 3.....	143
Figure 29: Decomposition Diagram part 8 – Subsystem 4.....	143
Figure 30: Decomposition Diagram part 9 – Subsystem 5.....	144
Figure 31: High Level Diagram part 1 - Subsystem 1	145
Figure 32: High Level Diagram part 2 - Subsystem 1	146
Figure 33: High Level Diagram part 3 - Subsystem 1	147
Figure 34: High Level Diagram part 4 - Subsystem 1	148
Figure 35: High Level Diagram part 5 - Subsystem 1	149
Figure 36: High Level Diagram part 6 - Subsystem 1	150
Figure 37: High Level Diagram part 7 - Subsystem 1	151
Figure 38: High Level Diagram part 8 - Subsystem 1	152
Figure 39: High Level Diagram part 9 - Subsystem 1	153
Figure 40: High Level Diagram part 1 - Subsystem 2	154

Figure 41: High Level Diagram part 2 - Subsystem 2	155
Figure 42: High Level Diagram - Subsystem 3	156
Figure 43: High Level Diagram- Subsystem 4	157
Figure 44: High Level Diagram- Subsystem 5	158
Figure 45: Middle Level DFD: 1.1. Approve Exhibition User	159
Figure 46: Middle Level DFD: 1.2 Search Exhibition User	159
Figure 47: Middle Level DFD: 1.3 Delete Exhibition User	160
Figure 48: Middle Level DFD: 1.4 Create New Exhibition	160
Figure 49: Middle Level DFD: 1.5 Search Exhibition	160
Figure 50: Middle Level DFD: Update Exhibition	161
Figure 51: Middle Level DFD: 1.7 Delete Exhibition.....	161
Figure 52: Middle Level DFD: 1.8 View Artist	162
Figure 53: Middle Level DFD: 1.9 Add Artwork.....	162
Figure 54: Middle Level DFD: 1.10 Search Artwork	163
Figure 55: Middle Level DFD: 1.11 Delete Artwork	163
Figure 56: Middle Level DFD: 1.12 Update Artwork	164
Figure 57: Middle Level DFD: 1.13 Create Artist Exhibition Invite	164
Figure 58: Middle Level DFD: 1.14 Print Tags	165
Figure 59: Middle Level DFD: 1.15 View Schedule.....	165
Figure 60: Middle Level DFD: 1.16 Print Schedule.....	166
Figure 61: Middle Level DFD: 1.17 Create Class	166
Figure 62: Middle Level DFD: 1.18 Search Class	167
Figure 63: Middle Level DFD: 1.19 Update Class	167
Figure 64: Middle Level DFD: 1.20 Delete Class.....	168
Figure 65: Middle Level DFD: 1.21 Generate Payment Reminder	168
Figure 66: Middle Level DFD: 1.22 Generate Payment Reminder	169
Figure 67: Middle Level DFD: 1.23 Create Class Teacher.....	169
Figure 68: Middle Level DFD: 1.24 Search Class Teacher	170
Figure 69: Middle Level DFD: 1.25 Update Class Teacher	170
Figure 70: Middle Level DFD: 1.26 Delete Class Teacher.....	171
Figure 71: Middle Level DFD: 1.27 Generate Booking Notification	171
Figure 72: Middle Level DFD: 1.28 Create Medium Detail	172
Figure 73: Middle Level DFD: 1.29 Search Medium Detail	173
Figure 74: Middle Level DFD: 1.30 Update Medium Detail	174
Figure 75: Middle Level DFD: 1.31 Delete Medium	175
Figure 76: Middle Level DFD: 1.32 Create Artwork Detail	176
Figure 77: Middle Level DFD: 1.33 Search Artwork	177
Figure 78: Middle Level DFD: 1.34 Update Artwork Detail.....	178
Figure 79: Middle Level DFD: 1.35 Delete Artwork	179
Figure 80: Middle Level DFD: 1.36 Archive Status	180
Figure 81: Middle Level DFD: 1.37 Create Surface type Detail	181
Figure 82: Middle Level DFD: 1.38 Search Surface Type.....	182
Figure 83: Middle Level DFD: 1.39 Update Surface Type detail	183
Figure 84: Middle Level DFD: 1.40 Delete Surface type	184

Figure 85: Middle Level DFD: 1.41 View Art Class Feedback Form.....	185
Figure 86: Middle Level DFD: 2.1 Book Art Class	186
Figure 87: Middle Level DFD: 2.2 View Existing Classes	186
Figure 88: Middle Level DFD: 2.3 Cancel Booking.....	186
Figure 89: Middle Level DFD: 2.4 Generate Booking Confirmation Message.....	187
Figure 90: Middle Level DFD: 2.5 Refund Status.....	187
Figure 91: Middle Level DFD: 2.6 Capture Class Feedback	188
Figure 92: Middle Level DFD: 2.7 Generate Class Teacher Information	188
Figure 93: Middle Level DFD: 2.8 Make Payment.....	189
Figure 94: Middle Level DFD: 2.9 Request Refund	190
Figure 95: Middle Level DFD: 3.1 Generate New User Report	190
Figure 96: Middle Level DFD: 3.2 Generate Art Class Report	190
Figure 97: Middle Level DFD: 3.3 Generate Class_Teacher Report	191
Figure 98: Middle Level DFD: 3.4 Generate Exhibition Report	191
Figure 99: Middle Level DFD: 3.5 Generate Class Survey Report	191
Figure 100: Middle Level DFD: 3.6 Generate Payment Report.....	192
Figure 101: Middle Level DFD: 4.1 Register User.....	192
Figure 102: Middle Level DFD: 4.2 Log in.....	192
Figure 103: Middle Level DFD: 4.3 Log out	193
Figure 104: Middle Level DFD: 4.4 Reset Password.....	193
Figure 105: Middle Level DFD: 4.5 Update User	194
Figure 106: Middle Level DFD: 4.6 View Guide.....	194
Figure 107: Middle Level DFD: 4.7 View Notification	195
Figure 108: Middle Level DFD: 4.8 Generate Frame Info	195
Figure 109: Middle Level DFD: 5.1 Apply for Exhibition	196
Figure 110: Middle Level DFD: 5.2 Apply Teach Class	196
Figure 111: Middle Level DFD: 5.3 Add Picture	196
Figure 112: Middle Level DFD: 5.4 Update Picture.....	197
Figure 113: Middle Level DFD: 5.5 Delete Picture	197
Figure 114: Middle Level DFD: 5.6 View Invitation.....	197
Figure 115: Middle Level DFD: 5.7 View Exhibitions.....	198
Figure 116: Middle Level DFD: 5.8 Accept Invitation.....	198
Figure 117: Low Level DFD: 1.1 Approve Exhibition User.....	199
Figure 118: Low Level DFD: 1.2 Search Exhibition User.....	200
Figure 119: Low Level DFD: 1.3 Notify Delete Exhibition User	201
Figure 120: Low Level DFD: 1.4 Create New Exhibition	202
Figure 121: Low Level DFD: 1.5 Search Exhibition	203
Figure 122: Low Level DFD: 1.6 Update Exhibition	204
Figure 123: Low Level DFD: 1.7 Delete Exhibition	205
Figure 124: Low Level DFD: 1.8 View Artist	206
Figure 125: Low Level DFD: 1.9 Add Artwork	207
Figure 126v: Low Level DFD: 1.10 Search Artwork	208
Figure 127: Low Level DFD: 1.11 Delete Artwork	209
Figure 128: Low Level DFD: 1.12 Update Artwork.....	210

Figure 129: Low Level DFD: 1.13 Create Exhibition Invitation.....	211
Figure 130: Low Level DFD: 1.14 Print Tags.....	212
Figure 131: Low Level DFD: 1.15 View Schedule	213
Figure 132: Low Level DFD: 1.16 Print Schedule.....	213
Figure 133: Low Level DFD: 1.17 Create Class	214
Figure 134: Low Level DFD: 1.18 Search Class	215
Figure 135: Low Level DFD: 1.19 Update Class	216
Figure 136: Low Level DFD: 1.20 Delete Class	217
Figure 137: Low Level DFD: 1.21 Generate Payment Reminder.....	218
Figure 138: Low Level DFD: 1.22 Generate Payment Reminder.....	219
Figure 139: Low Level DFD: 1.23 Create Class Teacher	220
Figure 140: Low Level DFD: 1.24 Search Class Teacher	221
Figure 141: Low Level DFD: 1.25 Update Class Teacher	222
Figure 142: Low Level DFD: 1.26 Delete Class Teacher	223
Figure 143: Low Level DFD: 1.27 Generate Booking Notification.....	224
Figure 144: Low Level DFD: 1.28 Create Medium Detail	225
Figure 145: Low Level DFD: 1.29 Update Medium Detail	226
Figure 146: Low Level DFD: 1.30 Update Medium detail	227
Figure 147: Low Level DFD: 1.31 Delete Medium.....	228
Figure 148: Low Level DFD: 1.32 Create Artwork Detail.....	229
Figure 149: Low Level DFD: 1.33 Search Artwork	230
Figure 150: Low Level DFD: 1.34 Update Artwork.....	231
Figure 151: Low Level DFD: 1.35 Delete Artwork	232
Figure 152: Low Level DFD: 1.36 Archive Status.....	233
Figure 153: Low Level DFD: 1.37 Create Surface type	234
Figure 154: Low Level DFD: 1.38 Search surface type	235
Figure 155: Low Level DFD: 1.39 Update Surface detail.....	236
Figure 156: Low Level DFD: 1.40 Delete Surface type	237
Figure 157: Low Level DFD: 1.41 View Art Class Feedback form	238
Figure 158: Low Level DFD: 2.1 Book Art Class.....	239
Figure 159: Low Level DFD: 2.2 View Existing Classes	240
Figure 160: Low Level DFD: 2.3 Cancel Booking	241
Figure 161: Low Level DFD: 2.4 Generate Booking Confirmation.....	242
Figure 162: Low Level DFD: 2.5 Refund Status	243
Figure 163: Low Level DFD: 2.6 Capture Class Feedback.....	244
Figure 164: Low Level DFD: 2.7 Generate Class Teacher	245
Figure 165: Low Level DFD: 2.8 Make Payment.....	246
Figure 166: Low Level DFD: 2.9 Request Refund	247
Figure 167: Low Level DFD: 3.1 Generate New User Report	248
Figure 168: Low Level DFD: 3.2 Generate Art Class Report.....	249
Figure 169: Low Level DFD: 3.3 Generate Class Teacher Report.....	250
Figure 170: Low Level DFD: 3.4 Generate Exhibition Report.....	251
Figure 171: Low Level DFD: 3.5 Generate Class Survey Report	252
Figure 172: Low Level DFD: 3.6 Generate Payment Report.....	253

Figure 173: Low Level DFD: 4.1 Register User.	254
Figure 174: Low Level DFD: 4.2 Log in	255
Figure 175: Low Level DFD: 4.3 Log Out	256
Figure 176: Low Level DFD: 4.4 Reset Password.....	257
Figure 177: Low Level DFD: 4.5 Update User.....	258
Figure 178: Low Level DFD: 4.7 View Notification	259
Figure 179: Low Level DFD: 4.8 Generate Frame Info	260
Figure 180: Low Level DFD: 5.1 Apply for Exhibition	261
Figure 181: Low Level DFD: 5.2 Apply to Teach	262
Figure 182: Low Level DFD: 5.3 Add Picture	263
Figure 183: Low Level DFD: 5.4 Update Picture.....	264
Figure 184: Low Level DFD: 5.5 Delete Picture	265
Figure 185: Low Level DFD: 5.6 View Invitation.....	266
Figure 186: Low Level DFD: 5.8 Accept Invitation Exhibition	267
Figure 187: Rich Picture Current System (Exhibitions)	348
Figure 188: Rich Picture Current System (Classe)Figure 189: Rich Picture Current System (Exhibitions)	348
Figure 190: Rich Picture Current System (Classe)	348
Figure 191: Rich Picture Proposed System (Exhibitions)Figure 192: Rich Picture Current System	
(Classe)Figure 193: Rich Picture Current System (Exhibitions)	348
Figure 194: Rich Picture Current System (Classe)Figure 195: Rich Picture Current System (Exhibitions)	
.....	348
Figure 196: Rich Picture Current System (Classe)	348
Figure 197: Rich Picture Proposed System (Exhibitions)Figure 198: Rich Picture Current System	
(Classe)	348
Figure 199: Rich Picture Proposed System (Exhibitions)Figure 200: Rich Picture Current System	
(Classe)	348
Figure 201: Rich Picture Proposed System (Exhibitions)Figure 202: Rich Picture Current System	
(Classe)	348
Figure 203: Rich Picture Proposed System (Exhibitions)	348



1. Document Introduction

This deliverable will be providing a complete overview of the functional specifications of the proposed system. It will entail the Dataflow diagrams (context diagram, decomposition diagram, high level diagram, middle level diagram and primitive level diagrams), use case diagrams and use case narratives. All the data flow diagrams will visually give more insight into how the system will be used and how each process will be carried out. This document will also, in detail outline the UML Modelling and Entity Relationship diagram of the proposed system. In order to explain the interfaces, inputs, outputs, reports, and validations; there will be tables describing each one in detail for each use case, respectively.



2. Use Cases

Introduction

This section will consist of the use case diagram and use case narratives for each requirement in the proposed system. It is generally shown as a graphic depiction of interactions among different elements in a system. Use case diagrams will specify the events in a system and how those events flow. They show the interaction between the system and its users.

2.1 Use Case Diagram

2.1.1 Administration Subsystem

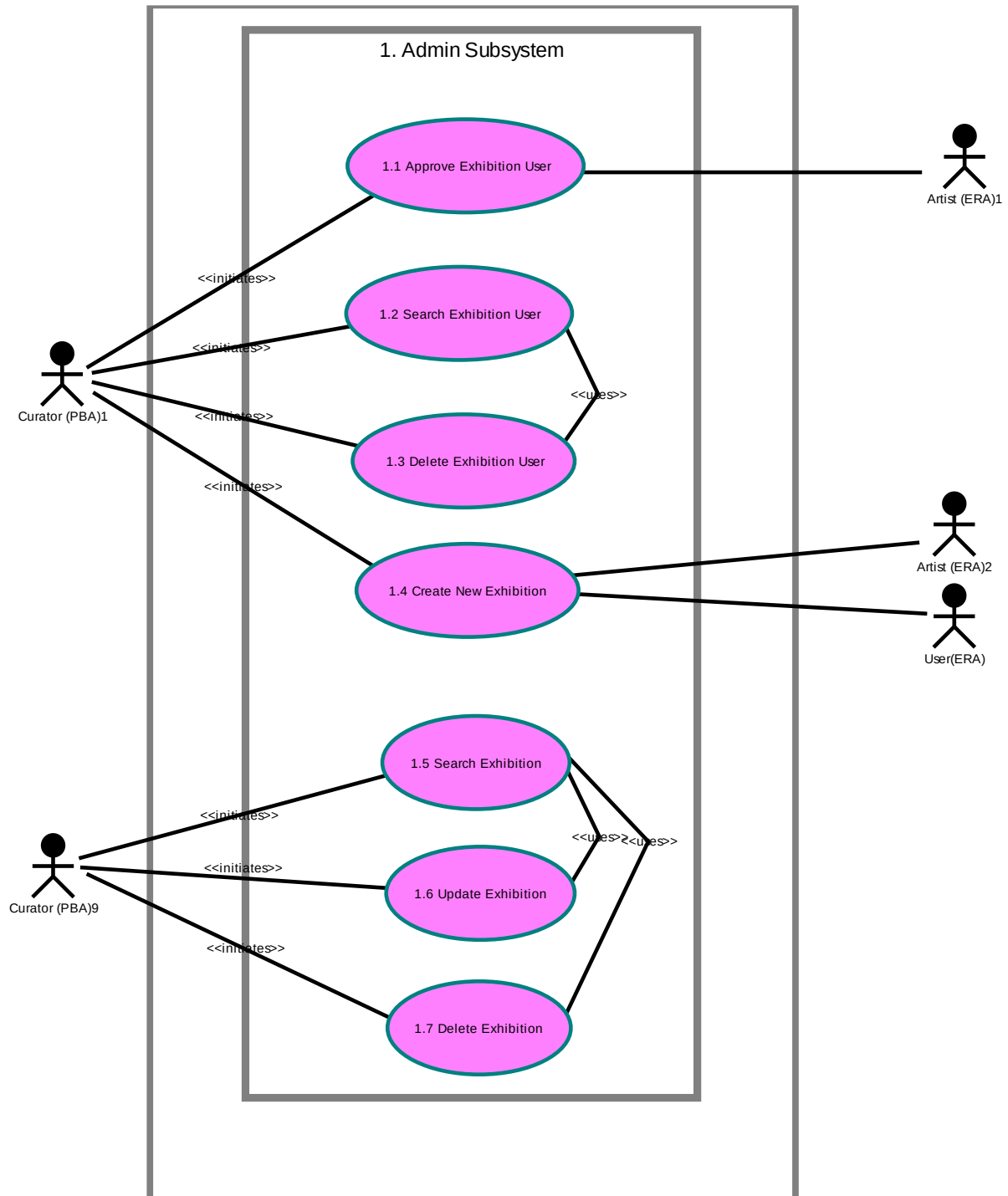


Figure 1: Use Case Diagram – Administration subsystem

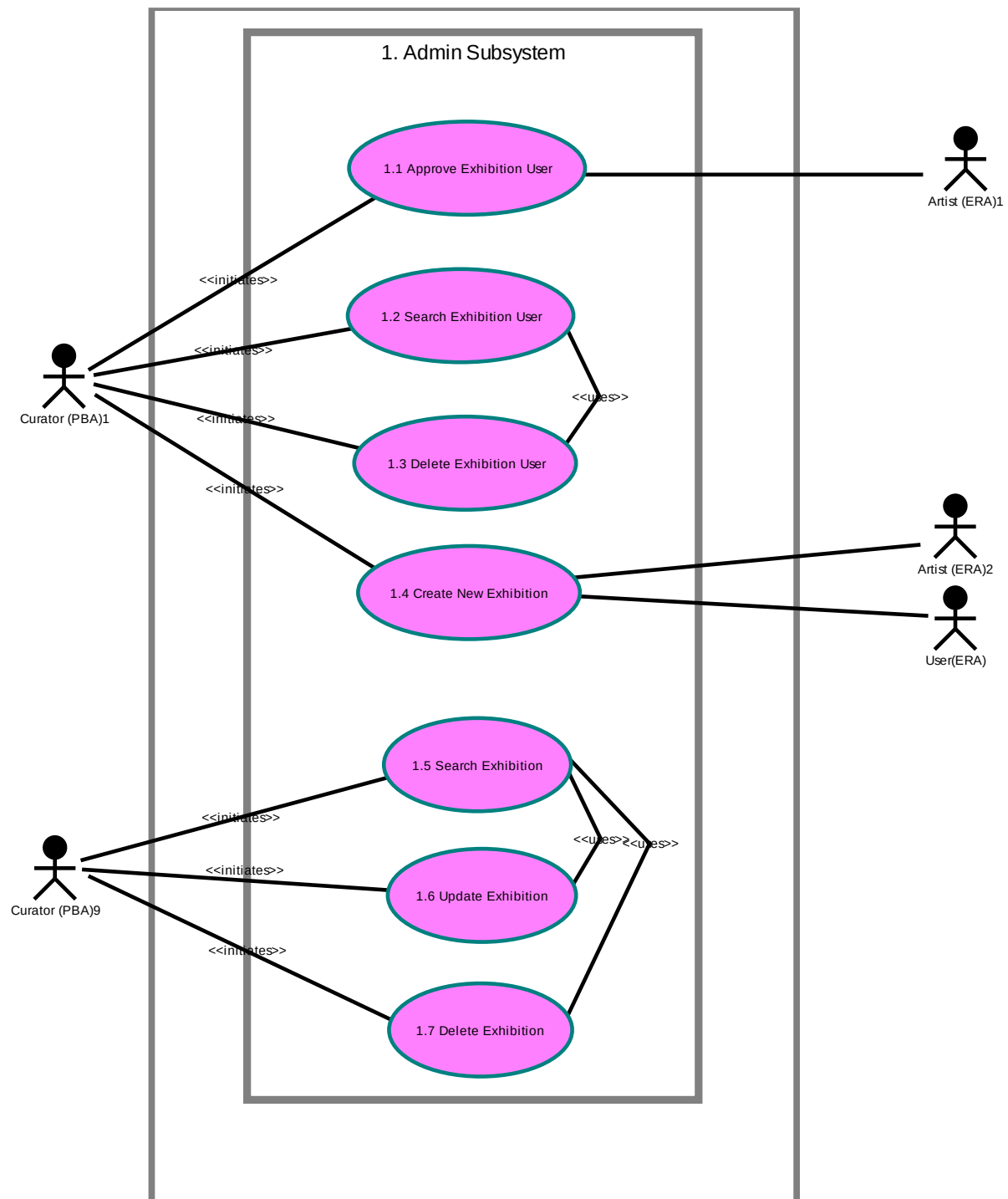


Figure 2: Use Case Diagram – Administration subsystem

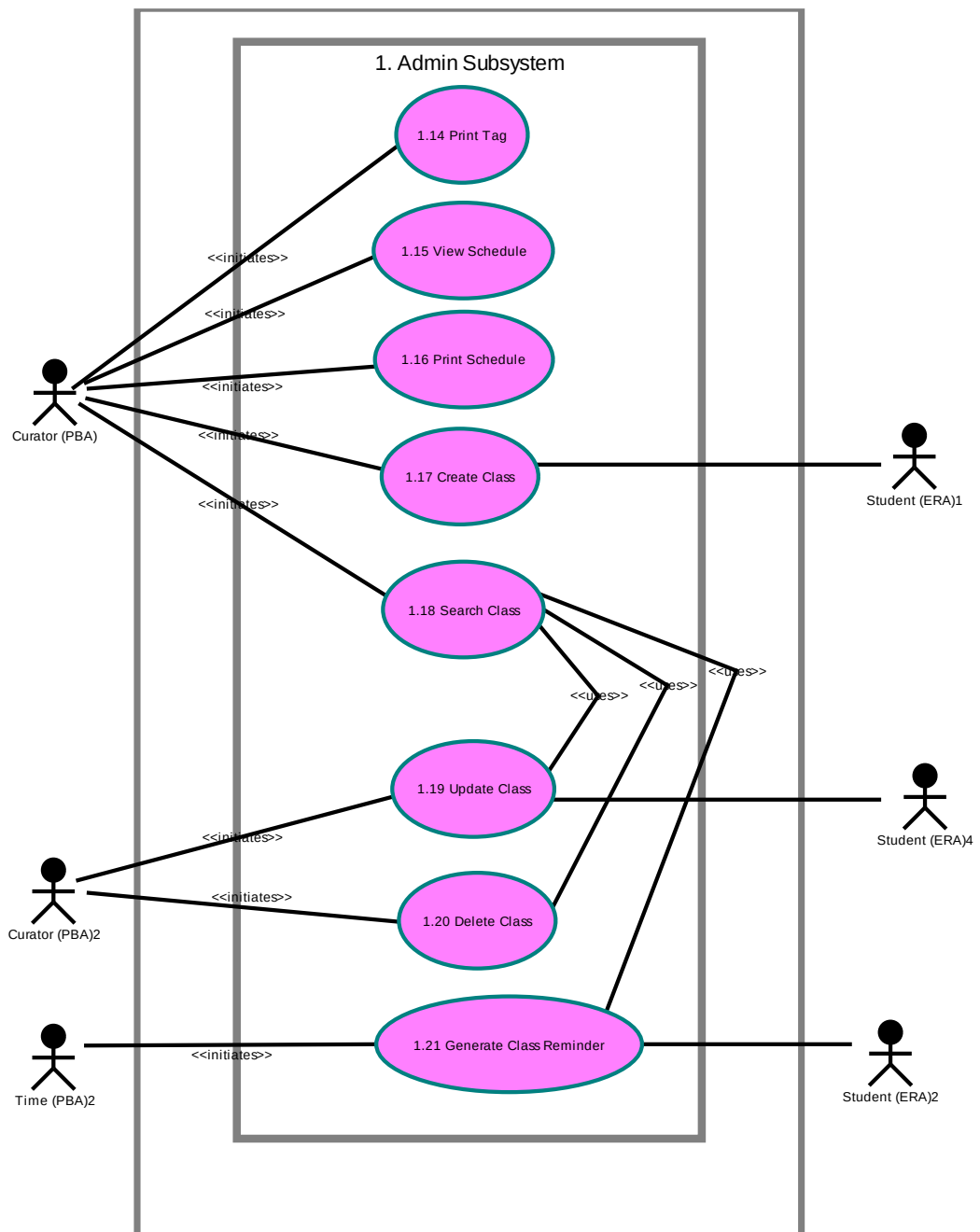


Figure 3: Use Case Diagram – Administration subsystem

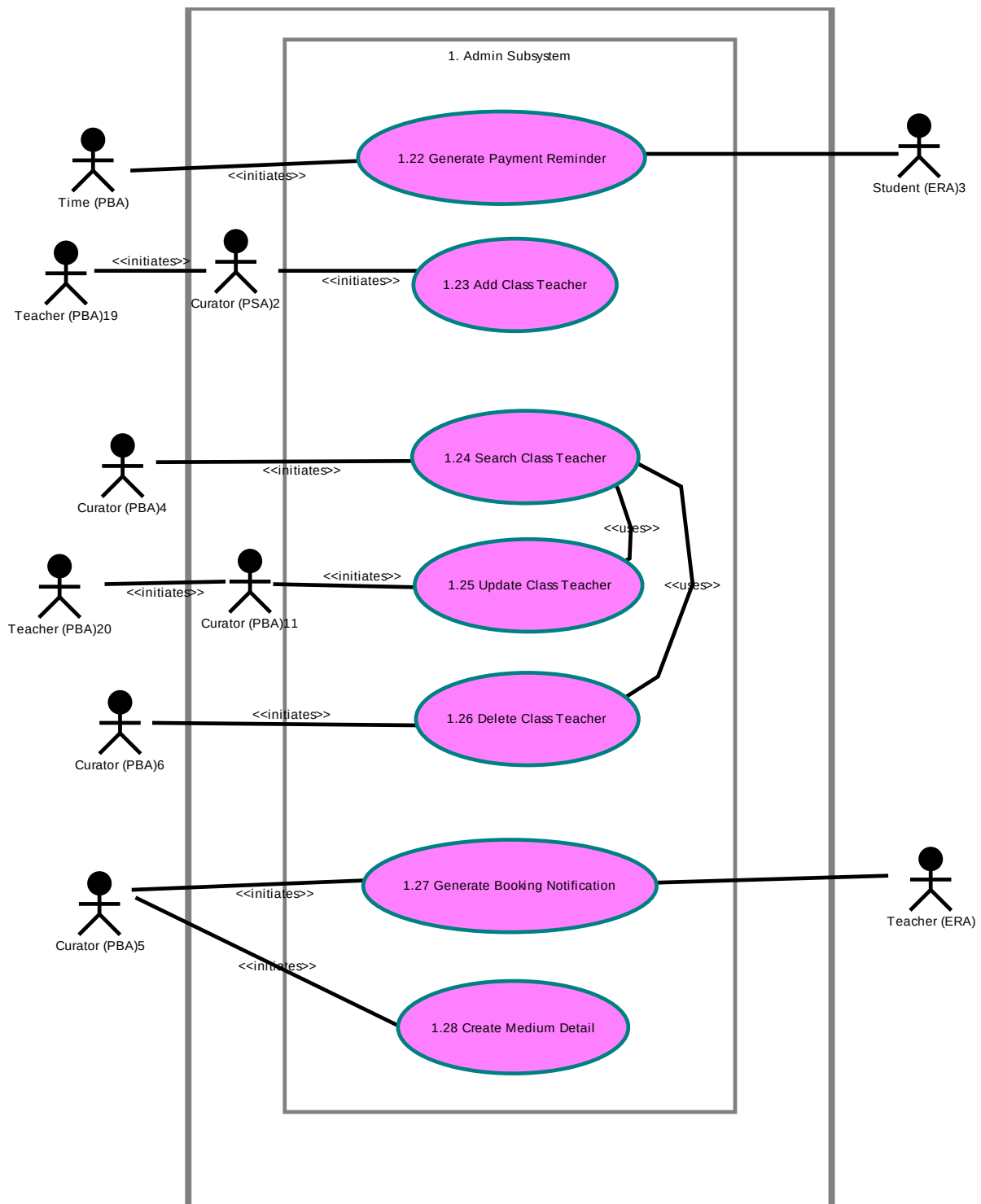


Figure 4: Use Case Diagram – Administration subsystem

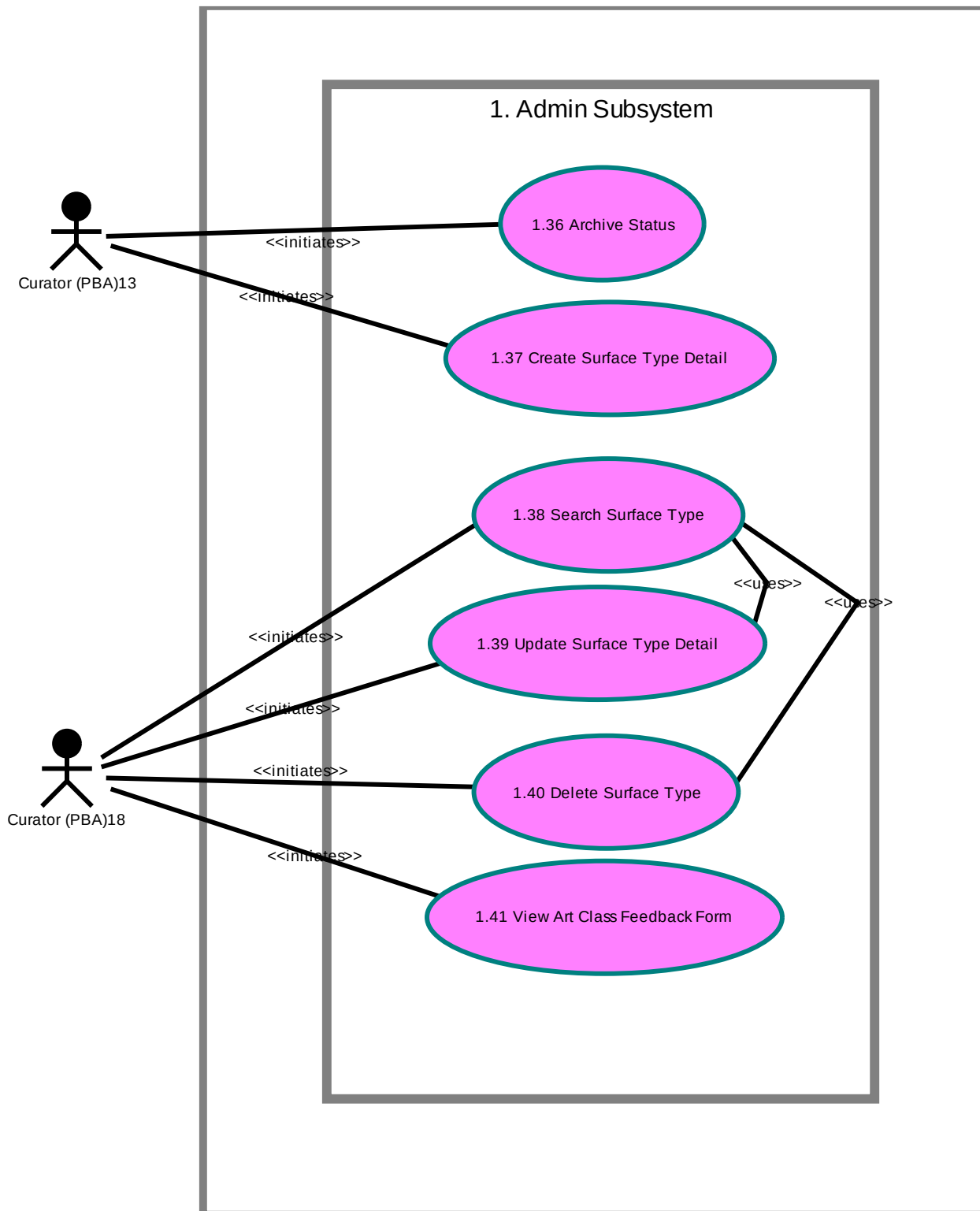


Figure 5: Use Case Diagram – Administration subsystem

2.1.2 Art Class Subsystem

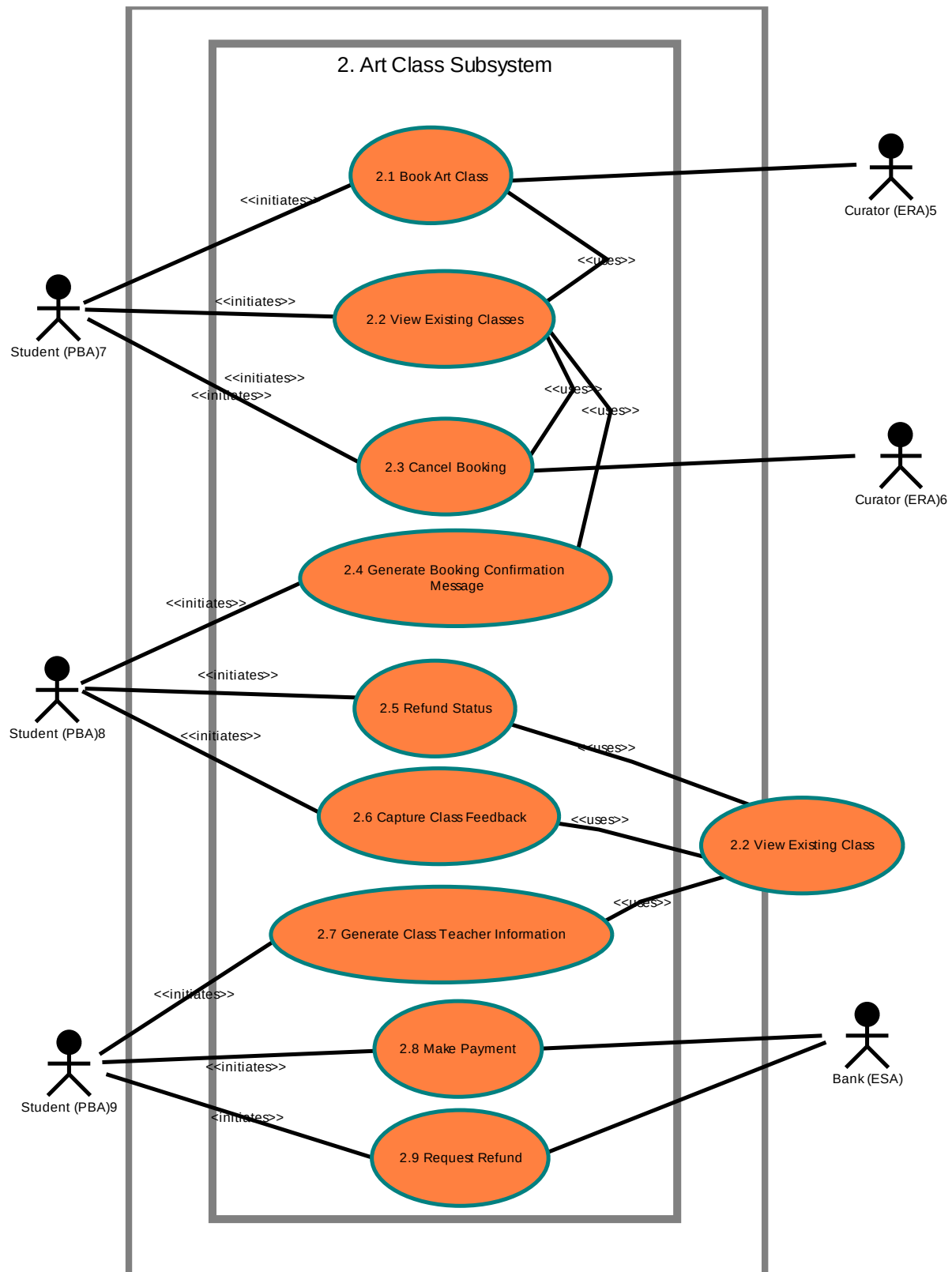


Figure 6: Use Case Diagram – Art Class subsystem

2.1.3 Reporting Subsystem

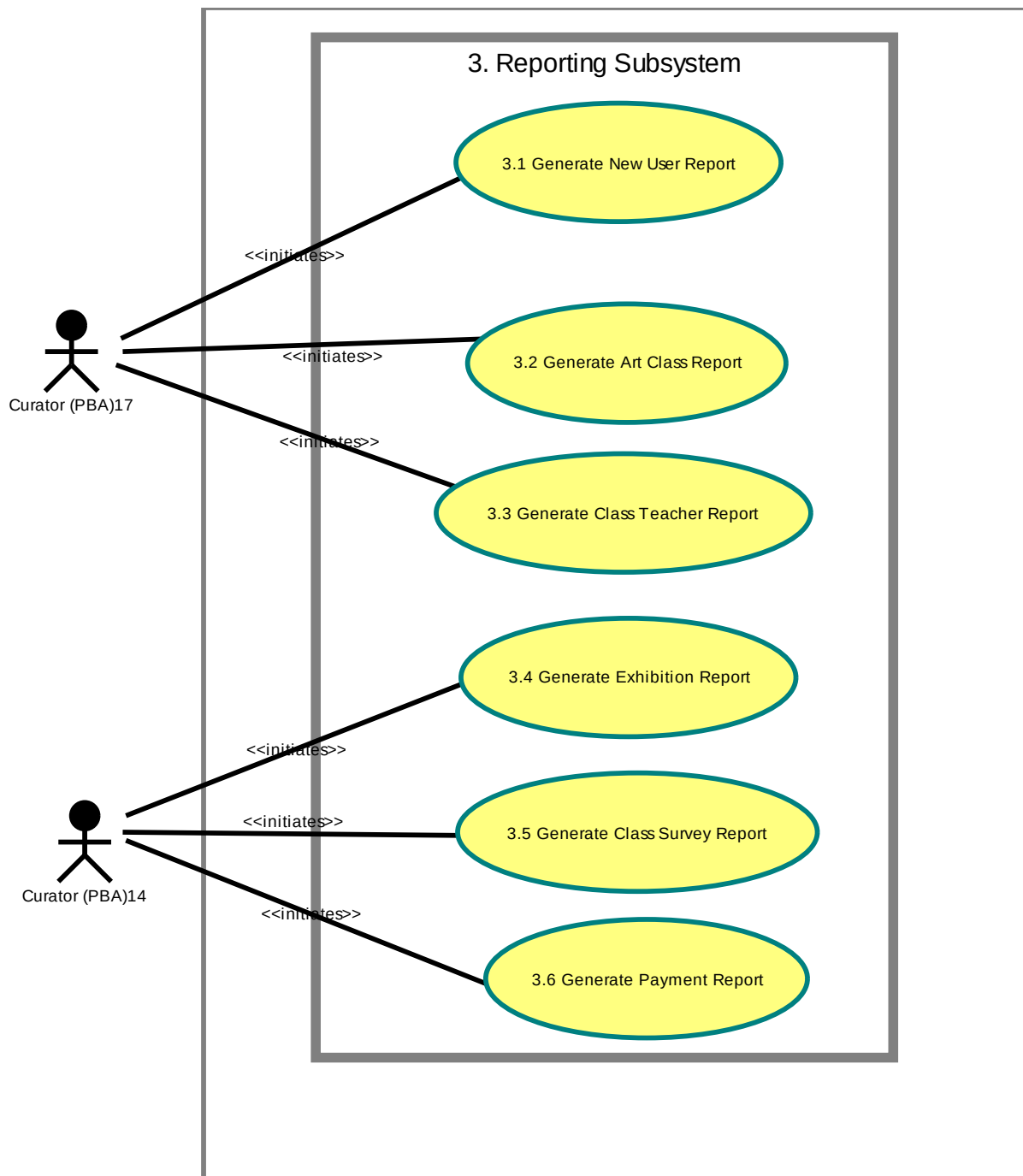


Figure 7: Use Case Diagram – Reporting subsystem

2.1.4 User Subsystem

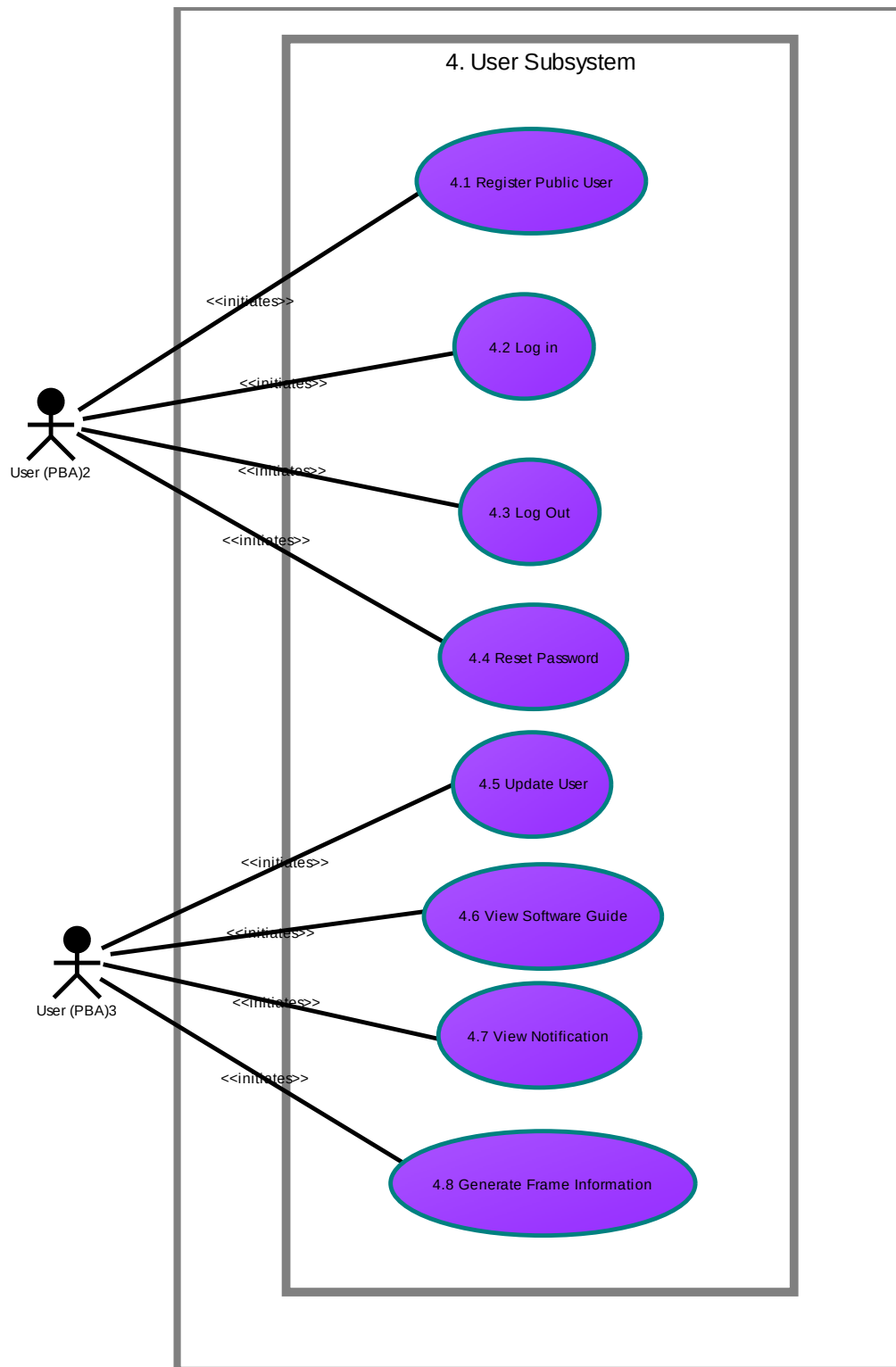


Figure 8: Use Case Diagram – User subsystem

2.2.5 Artist Subsystem

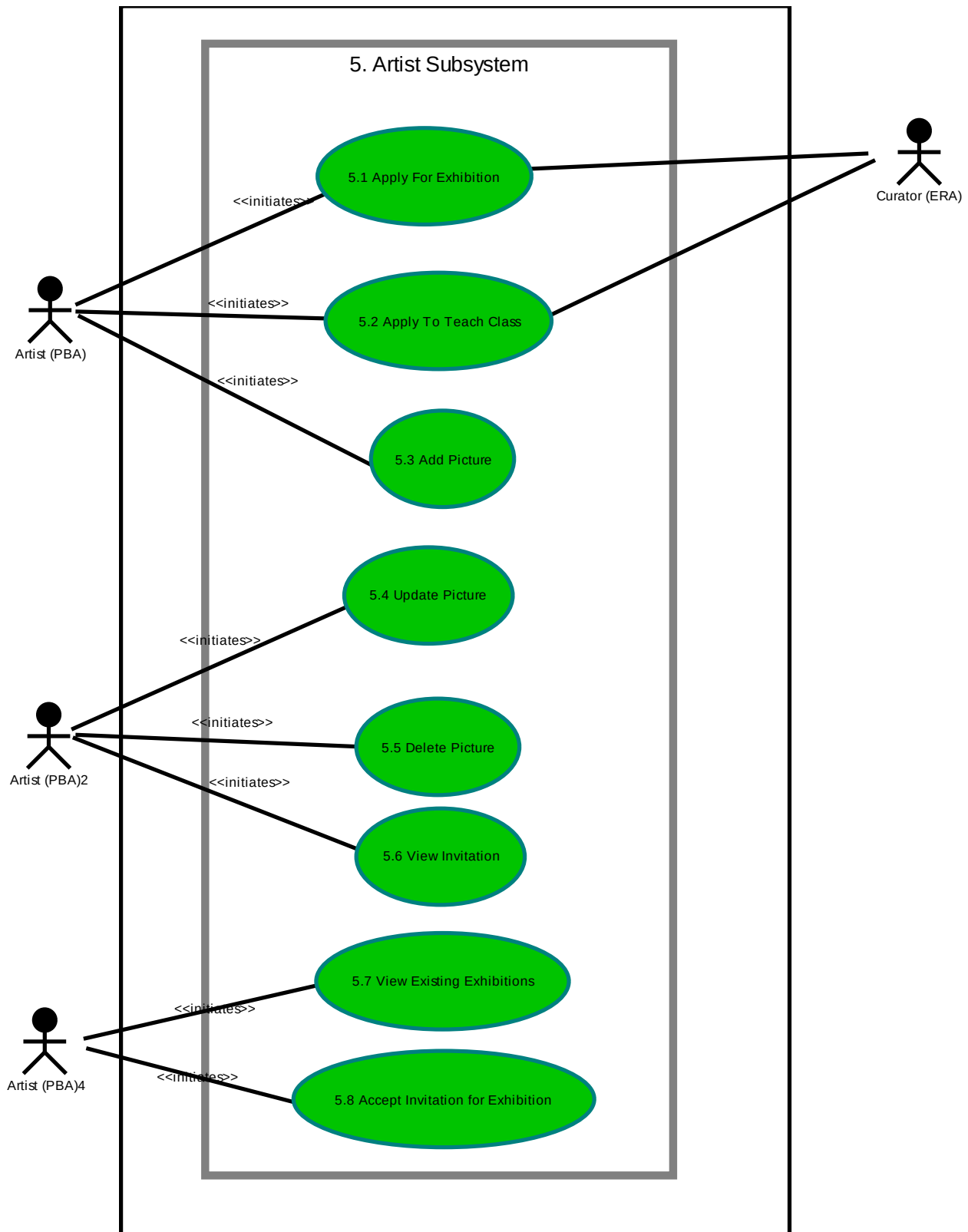


Figure 9: Use Case Diagram – Artist subsystem

2.2 Use Case Narratives

2.2.1 Administration Subsystem

Artech Systems

Author (s): Dean Botha
2021/05/03

Date: _____
Version: 1.0

USE CASE NAME:	Approve exhibition User	USE CASE TYPE Business Requirements: ☐ System Analysis: ☒ System Design: ☐	
USE CASE ID:	1.1		
PRIORITY:	High		
SOURCE:	Artech Requirements List		
PRIMARY BUSINESS ACTOR	Curator/Administration		
PRIMARY SYSTEM ACTOR			
OTHER PARTICIPATING ACTORS:	External Receiver Actor: Artist		
OTHER INTERESTED STAKEHOLDERS:	None		
DESCRIPTION:	This Use Case describes the action of the Curator approving the application of an Artist to join in on an exhibition. The Use Case ends where the Artist receives a notification that he is been accepted and should fill in his/her tags.		
PRE-CONDITION:	The Artist is registered on the system. The Curator is logged on to the system		
TRIGGER:	The Artist applies to be part of an exhibition.		
TYPICAL COURSE OF EVENTS:	Actor Action	System Response	
	Step 1: The Curator clicks to view the Applications for Exhibitions	Step 2: System loads and shows a table with applications grouped by exhibition from the Application table	
	Step 3: The Curator clicks on application to view the details.	Step 4: The system opens up all the details onto the screen of an application as well as the option to approve.	

		Step 5: The system sends a popup to confirm the Curator want to approve the Application after validating all information.
	Step 6: The Curator reads the application and clicks on Approve Application.	Step 7: The system adds the Artist to an Exhibition and sends a notification to the Artist to fill in his/her tags for the art to be displayed.
ALTERNATE COURSES:	ALT Step 6: The Curator reads the application and clicks on Deny Application.	
	ALT Step 7: The system sends a notification to the artist informing him/her that the application is denied.	
	ALT Step 7: The system notifies the Curator that the Exhibition User cannot be saved	
CONCLUSION:	The Artist can fill in tags for their artwork to be part of the exhibition	
POST-CONDITION:	The artist may access the Fill in Tags part of an exhibition	
BUSINESS RULES	The Person must be registered as an artist to apply for exhibitions	
IMPLEMENTATION CONSTRAINTS AND SPECIFICATIONS	None	
ASSUMPTIONS:	The Curator works through the Approvals in time for the exhibition.	
OPEN ISSUES:	None	



GROUP 23 DELIVERABLE 02



Artech Systems

Author (s): Dean Botha
2021/05/03

Date:

Version: 1.0

USE CASE NAME:	Search Exhibition User	USE CASE TYPE Business Requirements: ☐ System Analysis: ☒ System Design: ☐	
USE CASE ID:	1.2		
PRIORITY:	Low		
SOURCE:	Artech Requirements List		
PRIMARY BUSINESS ACTOR	Curator/Administration		
PRIMARY SYSTEM ACTOR	None		
OTHER PARTICIPATING ACTORS:	ERA: Curator		
OTHER INTERESTED STAKEHOLDERS:	None		
DESCRIPTION:	This Use Case describes the event where the Curator or Administration wants to search for		
PRE-CONDITION:	The Curator is logged into the system. At least one exhibition has been created An artist has been accepted to an exhibition		
TRIGGER:	The curator wants to search an exhibition user		
TYPICAL COURSE OF EVENTS:	Actor Action	System Response	
	Step 1: The Curator clicks on the Search Exhibition User button	Step 2: The system loads the Exhibition Artists table and details into the system in a table format.	
	Step 3: The user selects the Exhibition in question and enters Artist Name details into the search bar. Then clicks the search button	Step 4: The system matches this data with the ArtistName as well as ExhibitionName attribute from the Exhibitions Artists table after validation.	
		Step 5: The system returns all matched data to the Curator in a table format.	
	Step 6: The Curator receives the data on the screen in a table format		

ALTERNATE COURSES:	ALT Step 4: The System does not find any results and sends a message to the user that no matches were found, or data was not inputted correctly.	
CONCLUSION:	The Curator has received a list of the artist in question for an exhibition	
POST-CONDITION:	The Curator has received the Exhibition User details.	
BUSINESS RULES	An artist can appear only once in each exhibition but can be in many exhibitions	
IMPLEMENTATION CONSTRAINTS AND SPECIFICATIONS	None	
ASSUMPTIONS:	None	
OPEN ISSUES:		

Artech Systems

Author (s): Dean Botha
2021/05/03

Date:

Version: 1.0

USE CASE NAME:	Delete Exhibition User	USE CASE TYPE	
USE CASE ID:	1.3	Business Requirements:	☐
PRIORITY:	High	System Analysis:	☒
SOURCE:	Artech Requirements List	System Design:	☐
PRIMARY BUSINESS ACTOR	Curator/Administration		
PRIMARY SYSTEM ACTOR			
OTHER PARTICIPATING ACTORS:	None		
OTHER INTERESTED STAKEHOLDERS:	None		
DESCRIPTION:	This Use Case describes the process of deleting an exhibition user after the artist has registered for a specific exhibition. The Use Case ends with the table being updated to remove the Exhibition User.		
PRE-CONDITION:	The Artist must be part of a specific exhibition		
TRIGGER:	The Curator wants to delete a artist from an exhibition		
TYPICAL COURSE OF EVENTS:	Actor Action	System Response	
	Step 1: The Curator clicks on the Delete Exhibition User button.	Step 2: The system Invokes Use Case 1.2 Search Exhibition User to locate the specific Exhibition User	
	Step 3: The user selects the Exhibition in question and enters Artist Name details into the search bar. Then clicks the search button	Step 4: The system matches this data with the ArtistName as well as ExhibitionName attribute from the Exhibitions Artists table.	
	Step 5: The Curator selects a specific search result row and once again clicks on the Delete Exhibition User.	Step 6: The system prompts the user if he/she is certain the Exhibition User must be deleted	
	Step 7: The Curator provides confirmation that the Exhibition User must be deleted.	Step 8: The system deletes the record from the Exhibition Artist table.	
		Step 9: The system sends a popup that the record has been deleted.	
	Step 10: The User confirms the popup, and the Use case ends.		

ALTERNATE COURSES:	ALT Step 9: The system notifies the User that the Delete was not successful	
	ALT Step 7: The Curator denies the confirmation and the Use Case ends	
CONCLUSION:	The Curator has successfully deleted a entry in the Exhibition Artist table	
POST-CONDITION:	The Table Exhibition Users has been updated to delete the appropriate entry.	
BUSINESS RULES	The Curator may not delete an artist unless specific decline of the event is given.	
IMPLEMENTATION CONSTRAINTS AND SPECIFICATIONS	None	
ASSUMPTIONS:	The Artists will contact the Curator when they cannot attend the exhibition.	
OPEN ISSUES:		

Artech Systems

Author (s): Dean Botha
2021/05/03

Date: _____
Version: 1.0

USE CASE NAME:	Create New Exhibition	USE CASE TYPE	
USE CASE ID:	1.4	Business Requirements:	☐
PRIORITY:	High	System Analysis:	☒
SOURCE:	Artech Requirements List	System Design:	☐
PRIMARY BUSINESS ACTOR	Curator/Administration		
PRIMARY SYSTEM ACTOR			
OTHER PARTICIPATING ACTORS:	External Receiver Actor: Public Users & Artists		
OTHER INTERESTED STAKEHOLDERS:	None		
DESCRIPTION:	This Use Case describes the process of creating an exhibition on the system. The Curator creates an exhibition with the specific details which is added to the database and system. The Use Case ends when the Users receives a notification that a new Exhibition has been added to the list of Available exhibitions and is added to the table to view.		
PRE-CONDITION:	The Curator is logged on to the system		
TRIGGER:	The Curator wants to create a new exhibition		
TYPICAL COURSE OF EVENTS:	Actor Action	System Response	
	Step 1: The Curator clicks on Create New Exhibition.	Step 2: The system loads a form with the appropriate input fields for the Name, Date, Description, Venue, Announcement, and time as well as prompts the user to enter the details.	
	Step 3: The Curator fills in the appropriate details for the Exhibition.	Step 4: The system saves the details in the Exhibition table as well as creating an ExhibitionID and adding the Exhibition to the Schedule table. The system then invokes Use Case 1.12 Create Artist Exhibition Invite.	
		Step 5: The system displays a popup with a message that the Exhibition is loaded, and an announcement has been sent out to the Users.	
	Step 6: The user clicks the OK button and the window closes.		

ALTERNATE COURSES:	ALT Step 4: The system displays a popup that some characters or details are invalid. The Use Case goes back to Step 2	
	ALT Step 5: The system displays a popup that the data did not save correctly in the Exhibition table. The Use Case goes back to Step 3	
CONCLUSION:	The Exhibition has been saved to the appropriate tables and an announcement is sent to the Users.	
POST-CONDITION:	The table has been updated and the exhibition can be viewed on a website.	
BUSINESS RULES		
IMPLEMENTATION CONSTRAINTS AND SPECIFICATIONS	The Exhibition gets assigned an ExhibitionID on creation.	
ASSUMPTIONS:	None	
OPEN ISSUES:		



GROUP 23 DELIVERABLE 02



Artech Systems

Author (s): Dean Botha
2021/05/03

Date:

Version: 1.0

USE CASE NAME:	Search Exhibition	USE CASE TYPE	
USE CASE ID:	1.5	Business Requirements:	☐
PRIORITY:	High	System Analysis:	✓
SOURCE:	Artech Requirements List	System Design:	☐
PRIMARY BUSINESS ACTOR	Curator/Administration		
PRIMARY SYSTEM ACTOR			
OTHER PARTICIPATING ACTORS:	ERA: Curator/Administration		
OTHER INTERESTED STAKEHOLDERS:	None		
DESCRIPTION:	This Use Case describes the search of an Exhibition from the Exhibition table. The Use Case ends where the Curator has received the Exhibition details.		
PRE-CONDITION:	At least one exhibition has been created. The Curator is logged into the system		
TRIGGER:	The Curator wants to search a existing exhibition		
TYPICAL COURSE OF EVENTS:	Actor Action	System Response	
	Step 1: The Curator clicks on the Search Exhibition button.	Step 2: The system prompts the user to enter the search details for the Exhibition Name.	
	Step 3: The Curator enters the details into the search bar and presses the Search button again.	Step 4: The system processes the query and looks for a match in the Exhibition Name table and the entered details after validating all details.	
		Step 5: The system returns all matched data to the Curator in a table format.	
	Step 6: The Curator receives the data on the screen in a table format.		

ALTERNATE COURSES:	ALT Step 5: The System does not find any results and sends a message to the user that no matches were found. The user then presses the OK button on the popup and can search again.	
CONCLUSION:	The Curator has received a list of the exhibitions in question for the Gallery.	
POST-CONDITION:	The Curator has received the Exhibition details.	
BUSINESS RULES	None	
IMPLEMENTATION CONSTRAINTS AND SPECIFICATIONS	None	
ASSUMPTIONS:	None	
OPEN ISSUES:		



GROUP 23 DELIVERABLE 02



Artech Systems

Author (s): Dean Botha
2021/05/03

Date:

Version: 1.0

USE CASE NAME:	Update Exhibition	USE CASE TYPE	
USE CASE ID:	1.6	Business Requirements:	☐
PRIORITY:	High	System Analysis:	☒
SOURCE:	Artech Requirements List	System Design:	☐
PRIMARY BUSINESS ACTOR	Curator/Administration		
PRIMARY SYSTEM ACTOR			
OTHER PARTICIPATING ACTORS:	None		
OTHER INTERESTED STAKEHOLDERS:	None		
DESCRIPTION:	This Use Case describes the update of Exhibition details. The Use Case ends when the system has updated the exact details in the Exhibition table		
PRE-CONDITION:	The Curator is logged in At least one Artwork has been created		
TRIGGER:	The Curator wants to update the Artwork table		
TYPICAL COURSE OF EVENTS:	Actor Action	System Response	
	Step 1: The User clicks on the Update Artwork button	Step 2: The system invokes Use Case 1.5 Search Exhibition	
	Step 3: The Curator selects a specific search result row and once again clicks on the Update Exhibition.	Step 4: The system loads the information in the Artwork table of the specific Exhibition. The system prompts the user to change the details to Update	
	Step 5: The User enters the details to change from an Exhibition and clicks save details	Step 6: The system prompts the user if he/she is certain the Exhibition must be updated	
	Step 7: The Curator provides confirmation that the Exhibition must be updated.	Step 8: The system updates the record from the Exhibition table after validating all details entered.	

		Step 9: The system sends a popup that the record has been updated
	Step 10: The User confirms the popup, and the Use case ends.	
ALTERNATE COURSES:	ALT Step 6: The system displays a popup that some characters or details are invalid. The Use Case goes back to Step 4	
	ALT Step 7: The Curator denies the confirmation and the Use Case ends	
	ALT Step 5: The system displays a popup that the data did not save correctly in the Exhibition table. The Use Case goes back to Step 1	
CONCLUSION:	The Curator has successfully updated the Exhibition record.	
POST-CONDITION:	The record in the Exhibition Table has been updated.	
BUSINESS RULES	None	
IMPLEMENTATION CONSTRAINTS AND SPECIFICATIONS	None	
ASSUMPTIONS:	None	
OPEN ISSUES:		



GROUP 23 DELIVERABLE 02



Artech Systems

Author (s): Dean Botha
2021/05/03

Date:

Version: 1.0

USE CASE NAME:	Delete Exhibition	USE CASE TYPE Business Requirements: ☐ System Analysis: ☒ System Design: ☐	
USE CASE ID:	1.7		
PRIORITY:	High		
SOURCE:	Artech Requirements List		
PRIMARY BUSINESS ACTOR	Curator/Administration		
PRIMARY SYSTEM ACTOR			
OTHER PARTICIPATING ACTORS:	None		
OTHER INTERESTED STAKEHOLDERS:	None		
DESCRIPTION:	This Use Case describes the process of deleting an exhibition. The Use Case ends with the table being updated to remove the Exhibition.		
PRE-CONDITION:	At least one Exhibition must have been created The Curator is logged into the system		
TRIGGER:	The Curator wants to delete an exhibition from the system		
TYPICAL COURSE OF EVENTS:	Actor Action	System Response	
	Step 1: The Curator clicks on the Delete Exhibition button.	Step 2: The system Invokes Use Case 1.7 Search Exhibition to locate the specific Exhibition.	
	Step 3: The Curator selects a specific search result row and once again clicks on the Delete Exhibition User.	Step 4: The system prompts the user if he/she is certain the Exhibition must be deleted	
	Step 5: The Curator provides confirmation that the Exhibition User must be deleted.	Step 6: The system deletes the record from the Exhibition Artist table.	
		Step 7: The system sends a popup that the record has been deleted.	

	Step 8: The User confirms the popup and the Use case ends.	
ALTERNATE COURSES:	ALT Step 6: The System shows a popup that an error has occurred, and the Exhibition could not be deleted	
	ALT Step 7: The Curator denies the confirmation and the Use Case ends	
	ALT Step 4: The system displays a popup that some characters or details are invalid. The Use Case goes back to Step 2	
CONCLUSION:	The Curator has successfully deleted an entry in the Exhibition table	
POST-CONDITION:	The Table Exhibition has been updated to delete the appropriate entry.	
BUSINESS RULES	• .None	
IMPLEMENTATION CONSTRAINTS AND SPECIFICATIONS	• None	
ASSUMPTIONS:	• None	
OPEN ISSUES:		



GROUP 23 DELIVERABLE 02



Artech Systems

Author (s): Dean Botha
2021/05/03

Date:

Version: 1.0

USE CASE NAME:	View Artist	USE CASE TYPE Business Requirements: ☐ System Analysis: ☒ System Design: ☐	
USE CASE ID:	1.8		
PRIORITY:	High		
SOURCE:	Artech Requirements List		
PRIMARY BUSINESS ACTOR	Curator/Administration		
PRIMARY SYSTEM ACTOR			
OTHER PARTICIPATING ACTORS:	ERA: Curator/Administration		
OTHER INTERESTED STAKEHOLDERS:	None		
DESCRIPTION:	This Use Case describes the View of an Artist from the Artist table. The Use Case ends where the Curator has received the Artist details.		
PRE-CONDITION:	At least one Artist has been created. The Curator is logged into the system		
TRIGGER:	The Curator wants to view an existing Artist		
TYPICAL COURSE OF EVENTS:	Actor Action	System Response	
	Step 1: The Curator clicks on the View Artist button.	Step 2: The system prompts the user to enter the search details for the Artist Name and Artist Surname.	
	Step 3: The Curator enters the details into the search bar and presses the Search button.	Step 4: The system processes the query and looks for a match in the Artist table and the entered details after validation.	
		Step 5: The system returns all matched data to the Curator in a table format.	
	Step 6: The Curator receives the data on the screen in a table format.		

ALTERNATE COURSES:	ALT Step 5: The System does not find any results and sends a message to the user that no matches were found. The user then presses the OK button on the popup and can search again.	
CONCLUSION:	The Curator has received a list of the Artist in question for the Gallery.	
POST-CONDITION:	The Curator has received the Artist details.	
BUSINESS RULES	None	
IMPLEMENTATION CONSTRAINTS AND SPECIFICATIONS	None	
ASSUMPTIONS:	None	
OPEN ISSUES:		

Artech Systems

Author (s): Dean Botha
2021/05/03

Date: _____
 Version: 1.0

USE CASE NAME:	Add Artwork	USE CASE TYPE	
USE CASE ID:	1.9	Business Requirements:	☐
PRIORITY:	High	System Analysis:	☒
SOURCE:	Artech Requirements List	System Design:	☐
PRIMARY BUSINESS ACTOR	Curator/Administration		
PRIMARY SYSTEM ACTOR			
OTHER PARTICIPATING ACTORS:	None		
OTHER INTERESTED STAKEHOLDERS:	None		
DESCRIPTION:	This Use Case describes the process of creating an artwork on the system. The Curator creates an artwork with the specific details which is added to the database and system. The Use Case ends when the Users receives a notification that a new Artwork has been added to the list of Available Artworks and is added to the table to view.		
PRE-CONDITION:	The Curator is logged on to the system		
TRIGGER:	The Curator wants to create a new artwork		
TYPICAL COURSE OF EVENTS:	Actor Action Step 1: The Curator clicks on Create New Artwork. Step 3: The Curator fills in the appropriate details for the Exhibition.	System Response Step 2: The system loads a form with the appropriate input fields for the Title, Price, Picture, Description, SurfaceType,,MediumType,FrameColour, and FrameDimensions as well as prompts the user to enter the details. Step 4: The system saves the details in the Artwork table as well as adding an ExhibitionID,ArtistID and unique ArtworkID after validating all details. Step 5: The system displays a popup with a message that the Artwork is saved. The system automatically extracts the Tag information and saves it to the Tag table	

	Step 6: The user clicks the OK button and the window closes.	
ALTERNATE COURSES:	ALT Step 4: The system displays a popup that some characters or details are invalid. The Use Case goes back to Step 2	
	ALT Step 4: The System shows a popup that an error has occurred, and the Artwork could not be deleted. The Use Case is then ended.	
CONCLUSION:	The Artwork has been saved to the appropriate tables as well as the Tag information in the Tag table	
POST-CONDITION:	The tables has been updated and the Artwork can be viewed on a website.	
BUSINESS RULES	•	
IMPLEMENTATION CONSTRAINTS AND SPECIFICATIONS	• The Artwork gets assigned an ArtworkID on creation. • The system automatically separates the Tag information for printing later on	
ASSUMPTIONS:	• None	
OPEN ISSUES:		

Artech Systems

Author (s): Dean Botha
2021/05/03

Date: _____
 Version: 1.0

USE CASE NAME:	Search Artwork	USE CASE TYPE Business Requirements: ☐ System Analysis: ☒ System Design: ☐	
USE CASE ID:	1.10		
PRIORITY:	High		
SOURCE:	Artech Requirements List		
PRIMARY BUSINESS ACTOR	Curator/Administration		
PRIMARY SYSTEM ACTOR			
OTHER PARTICIPATING ACTORS:	None		
OTHER INTERESTED STAKEHOLDERS:	None		
DESCRIPTION:	This Use Case describes the search of an Artwork from the Artwork table. The Use Case ends where the Curator has received the Artwork details.		
PRE-CONDITION:	At least one Artwork has been created. The Curator is logged into the system		
TRIGGER:	The Curator wants to search a existing Artwork		
TYPICAL COURSE OF EVENTS:	Actor Action	System Response	
	Step 1: The Curator clicks on the Search Artwork button.	Step 2: The system prompts the user to enter the search details for the Artwork Name , ArtistName.SurfaceType or MediumType to search	
	Step 3: The Curator enters the details into the search bar and presses the Search button again.	Step 4: The system processes the query and looks for a match in the Artwork table and the entered details after validating all details.	
		Step 5: The system returns all matched data to the Curator in a table format.	
	Step 6: The Curator receives the data on the screen in a table format.		

ALTERNATE COURSES:	ALT Step 5: The System does not find any results and sends a message to the user that no matches were found. The user then presses the OK button on the popup and can search again. ALT Step 4: The system displays a popup that some characters or details are invalid. The Use Case goes back to Step 2	
CONCLUSION:	The Curator has received a list of the artworks in question for the Gallery.	
POST-CONDITION:	The Curator has received the artwork details.	
BUSINESS RULES	None	
IMPLEMENTATION CONSTRAINTS AND SPECIFICATIONS	None	
ASSUMPTIONS:	None	
OPEN ISSUES:		

Artech Systems

Author (s): Dean Botha
2021/05/03

Date: _____
 Version: 1.0

USE CASE NAME:	Delete Artwork	USE CASE TYPE	
USE CASE ID:	1.11	Business Requirements:	☐
PRIORITY:	High	System Analysis:	☒
SOURCE:	Artech Requirements List	System Design:	☐
PRIMARY BUSINESS ACTOR	Curator/Administration		
PRIMARY SYSTEM ACTOR			
OTHER PARTICIPATING ACTORS:	•		
OTHER INTERESTED STAKEHOLDERS:	•		
DESCRIPTION:	This Use Case describes the process of deleting an Artwork. The Use Case ends with the table being updated to remove the Artwork.		
PRE-CONDITION:	At least one Artwork must have been created The Curator is logged into the system		
TRIGGER:	The Curator wants to delete an Artwork from the system		
TYPICAL COURSE OF EVENTS:	Actor Action	System Response	
	Step 1: The Curator clicks on the Delete Artwork button.	Step 2: The system Invokes Use Case 1.9 Search Artwork to locate the specific Artwork	
	Step 3: The Curator selects a specific search result row and once again clicks on the Delete Artwork User.	Step 4: The system prompts the user if he/she is certain the Artwork must be deleted	
	Step 5: The Curator provides confirmation that the Artwork must be deleted.	Step 6: The system deletes the record from the Artwork table.	
		Step 7: The system sends a popup that the record has been deleted.	
	Step 8: The User confirms the popup, and the Use case ends.		

ALTERNATE COURSES:	ALT Step 4: The System shows a popup that an error has occurred, and the Artwork could not be deleted	
	ALT Step 7: The Curator denies the confirmation and the Use Case ends	
CONCLUSION:	The Curator has successfully deleted a entry in the Artwork table	
POST-CONDITION:	The Table Artwork has been updated to delete the appropriate entry.	
BUSINESS RULES		
IMPLEMENTATION CONSTRAINTS AND SPECIFICATIONS	None	
ASSUMPTIONS:	The Artwork will only be deleted in case an artist cannot deliver it or is damaged to the point of destruction	
OPEN ISSUES:		

Artech Systems

Author (s): Dean Botha
2021/05/03

Date:

Version: 1.0

USE CASE NAME:	Update Artwork	USE CASE TYPE	
USE CASE ID:	1.12	Business Requirements:	☐
PRIORITY:	Low	System Analysis:	☒
SOURCE:	Artech Requirements List	System Design:	☐
PRIMARY BUSINESS ACTOR	Curator/Administration		
PRIMARY SYSTEM ACTOR			
OTHER PARTICIPATING ACTORS:	ERA: Artist		
OTHER INTERESTED STAKEHOLDERS:	None		
DESCRIPTION:	This Use Case describes the update of Artwork details. The Use Case ends when the system has updated the exact details in the Artwork table and notified the User		
PRE-CONDITION:	The Curator is logged in At least one Artwork has been created		
TRIGGER:	The Curator wants to update the Artwork table		
TYPICAL COURSE OF EVENTS:	Actor Action	System Response	
	Step 1: The User clicks on the Update Artwork button	Step 2: The system invokes Use Case 1.9 Search Artwork	
	Step 3: The Curator selects a specific search result row and once again clicks on the Update Artwork.	Step 4: The system loads the information in the Artwork table of the specific artwork. The system prompts the user to change the details to Update	
	Step 5: The User enters the details to change from an Artwork and clicks save details	Step 6: The system prompts the user if he/she is certain the Artwork must be updated after validating all details.	
	Step 7: The Curator provides confirmation that the Artwork must be updated.	Step 8: The system updates the record from the Artwork table and updates the Tag table.	

		Step 9: The system sends a popup that the record has been updated and all Tag information has been updated.
	Step 10: The User confirms the popup, and the Use case ends.	
ALTERNATE COURSES:	ALT Step 6: The system displays a popup that some characters or details are invalid. The Use Case goes back to Step 4	
	ALT Step 7: The Curator denies the confirmation and the Use Case ends	
CONCLUSION:	The Curator has successfully updated the Artwork record.	
POST-CONDITION:	The record in the Artwork Table has been updated.	
BUSINESS RULES	None	
IMPLEMENTATION CONSTRAINTS AND SPECIFICATIONS	The Artist must receive a notification of the update and the relevant Tag table details must be updated	
ASSUMPTIONS:		
OPEN ISSUES:		

Artech Systems

Author (s): Dean Botha
2021/05/03

Date:

Version: 1.0

USE CASE NAME:	Create Artist Exhibition Invitation	USE CASE TYPE	
USE CASE ID:	1.13	Business Requirements:	☐
PRIORITY:	High	System Analysis:	☒
SOURCE:	Artech Requirements List	System Design:	☐
PRIMARY BUSINESS ACTOR	Curator/Administration		
PRIMARY SYSTEM ACTOR			
OTHER PARTICIPATING ACTORS:	ERA: Artist		
OTHER INTERESTED STAKEHOLDERS:			
DESCRIPTION:	This Use Case describes the creation of an artist exhibition invitations. This Use Case is called at the end of creating an Exhibition. The Use Case ends when the Artist receives the invite and a notification of the invite.		
PRE-CONDITION:	The Curator is logged on to the system		
TRIGGER:	The Curator wants to create a new artwork		
TYPICAL COURSE OF EVENTS:	Actor Action	System Response	
	Step 1: The Curator clicks on Send Exhibition Invitation button.	Step 2: The system loads a form with the appropriate input fields for the Invitation Details , Invitation Date , Invitation Status , and the selection of artists to send it to as well as prompting the user to enter the details.	
	Step 3: The Curator fills in the appropriate details for the Exhibition.	Step 4: The system saves the details in the Artwork table as well as adding an ExhibitionID , ArtistID and unique InvitationID . The system saves the invitation under the InvitationsPerArtist table as well.	
		Step 5: The system displays a popup with a message if the Curator is sure the details are correct, and the Invitation can be saved and sent after validating all characters.	
	Step 6: The user clicks the Confirm button and the window closes.	Step 7: The system displays a popup with a message that the Invitation is saved and sent.	

ALTERNATE COURSES:	ALT Step 4: The system displays a popup that some characters or details are invalid. The Use Case goes back to Step 2	
	ALT Step 6: The Curator denies the confirmation and the Use Case ends	
	ALT Step 7: The System shows a popup that an error has occurred, and the Invite could not be saved. The Use Case is then ended.	
CONCLUSION:	The invitation has been saved under the Invitation table, sent to the Artists, and saved under the InvitationsPerArtist table	
POST-CONDITION:	The Artist can view the invitation to accept or reject which changes the status.	
BUSINESS RULES		
IMPLEMENTATION CONSTRAINTS AND SPECIFICATIONS	When the status is updated, the Curator must be notified and added to the Exhibition table. The creation of Artworks must also be made available	
ASSUMPTIONS:		
OPEN ISSUES:		

Artech Systems

Author (s): Dean Botha
2021/05/03

Date:

Version: 1.0

USE CASE NAME:	Print Tags	USE CASE TYPE Business Requirements: ☐ System Analysis: ☒ System Design: ☐	
USE CASE ID:	1.14		
PRIORITY:	High		
SOURCE:	Artech Requirements List		
PRIMARY BUSINESS ACTOR	Curator/Administration		
PRIMARY SYSTEM ACTOR			
OTHER PARTICIPATING ACTORS:	External Receiver Actor: Curator		
OTHER INTERESTED STAKEHOLDERS:			
DESCRIPTION:	This Use Case describes the process of printing the tags for Artworks that must be hanged on the wall. The Use Case ends when the Curator receives the printed list of tags.		
PRE-CONDITION:	The Artists have filled in and submitted their tag information.		
TRIGGER:	The Curator wants to print the tags for the artwork to be displayed for an exhibition.		
TYPICAL COURSE OF EVENTS:	Actor Action		System Response
	Step 1: The Curator clicks on the Print Tags button to print the tags of a specific exhibition.		Step 2: The system retrieves all data in the Tag table with the selected ExhibitionID , compiles them and prompts the user for printer preferences
	Step 3: The Curator sets printer and paper preferences and clicks the Print button.		Step 4: The system sends a print command to the chosen printer system. The tags are printed in PDF format.
			Step 5: The system sends a message displaying the print was made successfully.

ALTERNATE COURSES:	ALT Step 2: The system sends a popup message that no tag information is available to print and returns to the exhibition screen.	
CONCLUSION:	The Curator receives the physical printout of all tag information for an exhibition.	
POST-CONDITION:	The Curator has the physical printout and can assign tags to the physical artworks	
BUSINESS RULES	All tags must be filled in and submitted 2 days beforehand	
IMPLEMENTATION CONSTRAINTS AND SPECIFICATIONS	Some form of notification must be sent if not all artists have submitted a tag 2 days beforehand.	
ASSUMPTIONS:		
OPEN ISSUES:		



GROUP 23 DELIVERABLE 02



Artech Systems

Author (s): Dean Botha
2021/05/03

Date:

Version: 1.0

USE CASE NAME:	View Schedule	USE CASE TYPE Business Requirements: ☐ System Analysis: ☒ System Design: ☐
USE CASE ID:	1.15	
PRIORITY:	Low	
SOURCE:	Artech Requirements List	
PRIMARY BUSINESS ACTOR	Curator/Administration	
PRIMARY SYSTEM ACTOR		
OTHER PARTICIPATING ACTORS:	External Receiver Actor: Curator/Administration	
OTHER INTERESTED STAKEHOLDERS:		
DESCRIPTION:	This Use Case describes the Curator viewing the Schedule where the parameters of dates can be chosen to view. The Use Case ends where the System outputs the chosen timeframe.	
PRE-CONDITION:	There are classes or exhibitions already loaded onto the system.	
TRIGGER:		
TYPICAL COURSE OF EVENTS:	Actor Action	System Response
	Step 1: The Curator clicks the button View Schedule and chooses the dates to filter between.	Step 2: The system processes the query using the DATE parameter from the Classes and Exhibitions Tables in the Schedule to show the details in a calendar format.
	Step 3: The User confirms the popup, and the Use case ends.	

ALTERNATE COURSES:	ALT Step 1: The Curator does not choose a filter timeframe. The system will show the full schedule in table format.	
CONCLUSION:	The Curator receives a table view of everything in the schedule in the date parameter set.	
POST-CONDITION:	The Curator receives a detailed table view of the schedule.	
BUSINESS RULES	None	
IMPLEMENTATION CONSTRAINTS AND SPECIFICATIONS	The system will only allow to set a date to view the timeframe if there is an event up to the chosen dates.	
ASSUMPTIONS:	None	
OPEN ISSUES:		

Artech Systems

Author (s): Dean Botha
2021/05/03

Date:

Version: 1.0

USE CASE NAME:	Print Schedule	USE CASE TYPE	
USE CASE ID:	1.16	Business Requirements:	☐
PRIORITY:	Low	System Analysis:	☒
SOURCE:	Artech Requirements List	System Design:	☐
PRIMARY BUSINESS ACTOR	Curator/Administration		
PRIMARY SYSTEM ACTOR			
OTHER PARTICIPATING ACTORS:	ERA: Curator		
OTHER INTERESTED STAKEHOLDERS:			
DESCRIPTION:	This Use Case describes the Curator printing the Schedule containing either booked Exhibitions or Classes. The Use Case ends where the Curator receives the printed schedule between the start and end date chosen.		
PRE-CONDITION:	The Curator is logged in and Classes or Exhibitions have been created.		
TRIGGER:	The Curator wants to view the schedule of activities for a certain period.		
TYPICAL COURSE OF EVENTS:	Actor Action	System Response	
	Step 1: The Curator clicks on the View Schedule button, sets the timeframe to view the schedule and clicks Set Timeframe	Step 2: The system loads all details from the Exhibition and Art Class tables where the Date attribute is in between the chosen Timeframe and displays it in a calendar format. The system prompts the user for printer preferences.	
	Step 3: The Curator sets printer and paper preferences and clicks the Print button.	Step 4: The system sends a print command to the chosen printer system. The schedule is printed in PDF format.	
		Step 5: The system sends a message displaying the print was made successfully.	
	Step 6: The Curator receives the printed Schedule and presses the OK button		

ALTERNATE COURSES:	ALT Step 1: The Curator does not choose a filter timeframe. The system will print the full schedule in PDF format.	
CONCLUSION:	The Curator receives a printed table view of everything in the schedule in the date parameter set.	
POST-CONDITION:	The Curator receives a detailed printed table view of the schedule.	
BUSINESS RULES	None	
IMPLEMENTATION CONSTRAINTS AND SPECIFICATIONS	The system will only allow to set a date to print the timeframe only if there is an event up to the chosen dates.	
ASSUMPTIONS:	None	
OPEN ISSUES:		

Artech Systems

Author (s): Dean Botha
2021/05/03

Date:

Version: 1.0

USE CASE NAME:	Create Class	USE CASE TYPE	
USE CASE ID:	1.17	Business Requirements:	☐
PRIORITY:	High	System Analysis:	☒
SOURCE:	Artech Requirements List	System Design:	☐
PRIMARY BUSINESS ACTOR	Curator/Administration		
PRIMARY SYSTEM ACTOR			
OTHER PARTICIPATING ACTORS:	None		
OTHER INTERESTED STAKEHOLDERS:	None		
DESCRIPTION:	This Use Case describes the process of creating a Class on the system. The Curator creates a Class with the specific details which is added to the database and system. The Use Case ends when the Users receives a notification that a new Art Class has been added to the list of Available Art Classes and is added to the table to view.		
PRE-CONDITION:	The Curator is logged on to the system		
TRIGGER:	The Curator wants to create a new artwork		
TYPICAL COURSE			
OF EVENTS:	Actor Action	System Response	
	Step 1: The Curator clicks on Create New Artwork.	Step 2: The system loads a form with the appropriate input fields for the ArtClassName , ArtClassDescription , ArtClassStartDate , ArtClassEndDate , ClassPrice , VenueID , ArtClassAnnouncement , ArtClassType_ID , ClassTeacher as well as prompts the user to enter the details.	
	Step 3: The Curator fills in the appropriate details for the Class.	Step 4: The system saves the details in the Artwork table as well as adding an OrganisationID and unique ArtClassID	
		Step 5: The system displays a popup with a message if the Curator is sure the details are correct, and the Class can be saved.	

	Step 6: The user clicks the Confirm button and the window closes.	Step 7: The system displays a popup with a message that the Class is saved, and an announcement has been made to the Users.
ALTERNATE COURSES:	ALT Step 4: The system displays a popup that some characters or details are invalid. The Use Case goes back to Step 2	
	ALT Step 7: The System shows a popup that an error has occurred, and the Artwork could not be deleted. The Use Case is then ended.	
	ALT Step 6: The Curator denies the confirmation and the Use Case ends	
CONCLUSION:	The Class has been saved to the appropriate tables as well as announcement has been sent out	
POST-CONDITION:	The table has been updated and the Class can be viewed on the website.	
BUSINESS RULES	None	
IMPLEMENTATION CONSTRAINTS AND SPECIFICATIONS	- The Class gets assigned an ArtClassID on creation. - The system automatically sends the announcement after creation.	
ASSUMPTIONS:	None	
OPEN ISSUES:		

{Artech Systems}

Author (s): Reagile Matloga Date: 2021/05/02
Version: 2.0

USE CASE NAME:	Search Class	USE CASE TYPE Business Requirements: ☐ • System Analysis: ☒ System Design: ☐ •
USE CASE ID:	1.18	
PRIORITY:	Medium	
SOURCE:	Artech Requirements List	
PRIMARY BUSINESS ACTOR	Curator	
PRIMARY SYSTEM ACTOR	None	
OTHER PARTICIPATING ACTORS:	None	
OTHER INTERESTED STAKEHOLDERS:	None	
DESCRIPTION:	This use case narrative describes the process in which the curator wants to view a certain class they have created. It begins with the curator, who will search the class name from the list of all classes schedule she has created from the Art_Class table for the art classes that has been added into the table. The process is concluded once the search results of art class names are presented.	
PRE-CONDITION:	The Curator must be logged into the system.	
TRIGGER:	The Curator wants to search an Art Class.	
TYPICAL COURSE OF EVENTS:	Actor Action	System Response
	Step 1: The Curator as the admin requests to search a class.	Step 2: The system requires/prompts the curator to enter the following search parameters: - ArtClassID - ArtClassName
	Step3: The curator inputs the requested search parameter.	Step4: The system captures the art class name search parameter.
		Step5: The system retrieves and displays the search results of what was called from the Art_Class table. [ALT]
	Step6: The curator selects the class name from the search results.	Step7: The system displays the following information of the class from the Art_Class table: - ArtClassID - ArtClassName - ArtClassStartDate - ArtClassEndDate
ALTERNATE COURSES:	ALT STEP5: There is no Art Class in the system that corresponds with name that was typed in the search parameter. - The system notifies the curator that no search results were found in the system. - Terminate the use case.	
CONCLUSION:	This use case is concluded when the system displays the search results of art class name to the curator.	
POST-CONDITION:	The Art_Class table was searched for an entry that corresponds with the search parameter.	
BUSINESS RULES	Only the Curator can search a class on the system.	

IMPLEMENTATION CONSTRAINTS AND SPECIFICATIONS	None
ASSUMPTIONS:	None
OPEN ISSUES:	None

{Artech Systems}

Author (s): _____ Reagile Matloga _____ Date: _____ 2021/05/01
Version: __ 1.0 _____

USE CASE NAME:	Update Class	USE CASE TYPE Business Requirements: ☐ • System Analysis: ☒ System Design: ☐ •
USE CASE ID:	1.19	
PRIORITY:	Low	
SOURCE:	Artech Systems Requirements List	
PRIMARY BUSINESS ACTOR	Curator	
PRIMARY SYSTEM ACTOR	None	
OTHER PARTICIPATING ACTORS:	None	
OTHER INTERESTED STAKEHOLDERS:	None	
DESCRIPTION:	This use case describes the process in which the Curator updates the class she has created. The use case begins when the curator requests to update the details of the Art Class. The system will display the details from the Art_Class table. The curator can update things like the name, description, or date. The curator will update the necessary changes and the system will update the Art_Class table.	
PRE-CONDITION:	- The Curator must be logged into the system. - The Art Class the curator wants to update must already exist in the system.	
TRIGGER:	The curator wants to update an art class.	
TYPICAL COURSE OF EVENTS:	Actor Action	System Response
	Step 1: The curator wants to update the details of an Art Class.	Step 2: The system invokes Use Case 1.18 “Search Art Class”
		Step3: The system will ask the user to update any of the following details: - Art Class Name - Art Class Description - Art Class Period - Class Price - Art Class Type - Art Class Teacher - Art Class Type - Venue
	Step4: The curator will apply the necessary changes.	Step5: The system captures and validates the changes applied by checking whether all fields have been filled in and done correctly. [ALT]
		Step6: The system will prompt the curator to confirm the changes to the changes they made.

	Step7: The Curator confirms the changes made to the art class details. [ALT]	Step8: The system will save the changes made to the Art_Class table in the database: • Art Class Name • Art Class Description • Art Class Period • Class Price • Art Class Type • Art Class Teacher • Art Class Type • Venue
		Step9: The system will notify that the changes have been successfully made.
ALTERNATE COURSES:	ALT STEP 5: The details the curator has changed are not in the correct format and not all fields are completed. • The system displays a validation error to the user. • Returns to Step 3. ALT STEP 7: The Curator does not confirm the changes made to the art class details. • Returns to Step 3.	
CONCLUSION:	The use case concludes when the curator is notified by the system that changes have been made successfully.	
POST-CONDITION:	The updated Art Class information is saved in the Art_Class table.	
BUSINESS RULES	• Only the Curator can update Art Class details.	
IMPLEMENTATION CONSTRAINTS AND SPECIFICATIONS	• None	
ASSUMPTIONS:	• None	
OPEN ISSUES:	None	

{Artech Systems}

Author (s): _____ Reagile Matloga Date: 2021/05/01
Version: 2.0

USE CASE NAME:	Delete Class	USE CASE TYPE Business Requirements: ☐ System Analysis: ☒ System Design: ☐
USE CASE ID:	1.20	
PRIORITY:	Low	
SOURCE:	Artech Systems Requirements List	
PRIMARY BUSINESS ACTOR	Curator	
PRIMARY SYSTEM ACTOR	None	
OTHER PARTICIPATING ACTORS:	• None	
OTHER INTERESTED STAKEHOLDERS:	• None	
DESCRIPTION:	This use case describes how the Curator removes an Art Class from a slot that they have created it from. The system will display schedule of all the classes and their slots. The Curator will delete the specific class from the schedule.	

PRE-CONDITION:	The Curator must be logged into the system.	
TRIGGER:	The Curator wants to delete an art class they have created.	
TYPICAL COURSE OF EVENTS:	Actor Action	System Response
	Step 1: The curator wants to delete/remove an art class they have created.	Step 2: System invokes use case 1.18 "Search Class"
	Step 3: The Curator selects the Art Class to be deleted.	Step4: The system captures the following : • ArtClassID • ArtClassStartDate • ArtClassEndDate
		Step5: The System asks the curator if they wish to delete the selected Art Class.
	Step6: The Curator confirms to delete the Art Class. [ALT]	Step7: The system deletes the details of the Art Class from the table/schedule.
		Step8: The system notifies the curator that the art class has been deleted successfully.
ALTERNATE COURSES:	ALT Step 5: The Curator does not confirm to delete the Art Class. • Terminate the use case.	
CONCLUSION:	This use case concludes when the Curator is informed that the Art Class has been deleted successfully.	
POST-CONDITION:	The details of the art class are removed from the table/schedule.	
BUSINESS RULES	• Only the Curator can delete the Art Class. • The Curator cannot simply delete the Art Class that has art class bookings made to it.	
IMPLEMENTATION CONSTRAINTS AND SPECIFICATIONS	• None	
ASSUMPTIONS:	• None	
OPEN ISSUES:	None	

{Artech Systems }

Author (s): _____ Reagile Matloga Date: 2021/05/01
Version: 2.0

USE CASE NAME:	Generate Class Reminder	USE CASE TYPE Business Requirements: ☐ • System Analysis: ☒ System Design: ☐ •
USE CASE ID:	1.21	
PRIORITY:	High	
SOURCE:	Artech System Requirements List	
PRIMARY BUSINESS ACTOR	Time	
PRIMARY SYSTEM ACTOR	None	
OTHER PARTICIPATING ACTORS:	• None	
OTHER INTERESTED STAKEHOLDERS:	• Student(ERA)	
DESCRIPTION:	This use case describes the process of automatically generating a reminder that reminds the art class students of the art class coming up .The use case begins when the wants to create a full reminder of their upcoming class that has been scheduled. The system will capture the description of the reminder that will be sent.	

PRE-CONDITION:	The curator must be logged into the system	
TRIGGER:	The curator wants to create a reminder message description to send to the students to remind them of the art class.	
TYPICAL COURSE OF EVENTS:	Actor Action	System Response
	Step 1: The curator wants to create a class reminder.	Step2: The system prompts the curator to generate the Class reminder.
		Step4: System Invokes "Search ArtClass" use case
		Step5: : The system reads information from the ArtClass , ArtClass_Type , USER and Booking tables : • UserId • UserEmailAddress • ArtClassName • ArtClassStartDate • ArtClassEndDate • ArtClass_TypeId • PaymentStatus • Date • Time • ArtClassType are read from their respective tables
		Step5: Invokes use case 4.1. "Register User"
		Step 6: The system captures the email of the user from User Table
		Step 7: The system sends report to the user.
	Step 8: The client receives the generated report from the system.	
ALTERNATE COURSES:	• None	
CONCLUSION:	This use case concludes when the system notifies the Curator that a class reminder has been created successfully.	
POST-CONDITION:	The system has generated and sent out the class reminder and the client receives it.	
BUSINESS RULES	• The report is automatically generated a week before the client's booking is due and they have not paid the booking.	

IMPLEMENTATION CONSTRAINTS AND SPECIFICATIONS	None
ASSUMPTIONS:	None
OPEN ISSUES:	None

{Artech Systems}

Author (s): _____ Reagile Matloga Date: 2021/05/01

Version: 2.0

USE CASE NAME:	Generate Payment Reminder	USE CASE TYPE Business Requirements: ☐ • System Analysis: ☒ System Design: ☐ •
USE CASE ID:	1.22	
PRIORITY:	High	
SOURCE:	Artech Systems Requirements List	
PRIMARY BUSINESS ACTOR	Time (PBA)	
PRIMARY SYSTEM ACTOR	None	
OTHER PARTICIPATING ACTORS:	• Student (ERA)	
OTHER INTERESTED STAKEHOLDERS:	• None	
DESCRIPTION:	This use case describes the process where a reminder is generated automatically to remind the Art Class students of the upcoming class to pay their outstanding fee, given that they have paid a deposit for their Art Class Booking. This report will put the class booking, date of the booking, time of the booking and description of what they will need.	
PRE-CONDITION:	The Student must have made a booking and paid deposit.	
TRIGGER:	There is a week to go to the class.	
TYPICAL COURSE OF EVENTS:	Actor Action	System Response
	Step 1: There are two weeks to go to the class.	Step 2: The system will show the generate the payment reminder.
		Step3: The system invokes “2.1 Book Art Class” Use Case
		Step 4: The system reads information from the Booking, Art Class, Payment , User tables: • UserID • UserEmailAddress • BookingDate • BookingTime • ArtClassPrice • PaymentAmount • PaymentDate And read from their tables and generate a report.
		Step5: The System invokes “Search Student”. The UserEmailAddress will be used.
		Step6: The system sends the report generated to the client.

	Step7: The Student receives the report from the system.	
ALTERNATE COURSES:	None	
CONCLUSION:	The student receives the generated report from the system.	
POST-CONDITION:	The system will generate and send out the Payment Reminder and show the student what they still need to pay.	
BUSINESS RULES	The report is generated automatically a week before the Art Class takes place and the students have not paid yet.	
IMPLEMENTATION CONSTRAINTS AND SPECIFICATIONS	None	
ASSUMPTIONS:	None	
OPEN ISSUES:	None	

{Artech Systems}

Author (s): _____ Reagile Matloga Date: 2021/05/01
Version: 2.0

USE CASE NAME:	Create Class Teacher	USE CASE TYPE Business Requirements: ☐ • System Analysis: ☒ System Design: ☐ •
USE CASE ID:	1.23	
PRIORITY:	High	
SOURCE:	Artech Systems Requirements List	
PRIMARY BUSINESS ACTOR	Art Class Teacher (PBA)	
PRIMARY SYSTEM ACTOR	Curator (PSA)	
OTHER PARTICIPATING ACTORS:	• None	
OTHER INTERESTED STAKEHOLDERS:	• None	
DESCRIPTION:	This use case describes the process of adding an Art Class Teacher to the system. The Use Case begins when a potential teacher calls or contacts the curator requesting to become a teacher for an Art class. The Curator will request the potential teacher's details, capture and validate them and add them into the system. Then the person will be added as a teacher.	
PRE-CONDITION:	• The Curator must be logged into the system. • The potential teacher is not an existing teacher on the system.	
TRIGGER:	A potential Art Class teacher calls in and requests to become a teacher for an art class.	

TYPICAL COURSE OF EVENTS:	Actor Action	System Response
	Step1: A potential Art Class teacher calls in and requests to become a teacher for an art class.	Step2: The Curator selects to add an Art Class teacher to the system.
		Step3: The system requests the details of the Art Class Teacher.
	Step4: The Art Class teacher provides their specific details like: • Name • Surname • Email Address • AddressLine1 • AddressLine2 • Medium	Step5: The Curator enters the details of the Art Class teacher onto the System
		Step 4: The system requests the Curator to confirm if the details entered are correct.
	Step5: Step 9: The Art Class Teacher confirms that his/her details are correct. [ALT]	Step 10: The Curator confirms that the details are correct.
		Step7: The system will ask the Curator to confirm whether the details are correct.
	Step8: The Teacher confirms that their details are correct.	Step9: The Curator confirms the details are correct.
		Step10: The system captures and validates the filled in fields checking whether all fields have been filled in and done correctly. [ALT]
		Step 11: The system generates a new ClassTeacher_ID
		Step 12: (ALT)The system saves the details of the Art Class Teacher as follows: In the ClassTeacher table • ClassTeacher_ID • ClassTeacher_Name • ClassTeacher_Surname • ClassTeacher_EmailAddress • ClassTeacher_AddressLine1 • ClassTeacher_AddressLine2 • ClassTeacher_Medium
		Step14: The system notifies the Curator that the Class Teacher has been created successfully.
		Step15: The Curator notifies the Teacher that have been added successfully.
ALTERNATE COURSES:	ALT STEP 9: The Art Class Teacher does not confirm his/her details. → Return to Step 4.	
	ALT STEP 10: The entered details are not in the correct format or not all the required fields are completed.	

	→ The system displays a notification of the validation error to the System Administrator → Returns to Step 4.
	ALT Step 14: The saving of the Art Class teacher information was unsuccessful. → The system displays a notification indicating the creation of the Art Class Teacher was unsuccessful. → The Curator notifies the Art Class Teacher that the creation of their profile was unsuccessful. → Terminate this use case
CONCLUSION:	This use case concludes when the Curator informs the Art Class teacher that they have been added successfully.
POST-CONDITION:	The Art Class Teacher has been added to the ClassTeacher table.
BUSINESS RULES	Only a Curator can add a new teacher to the system.
IMPLEMENTATION CONSTRAINTS AND SPECIFICATIONS	None
ASSUMPTIONS:	None
OPEN ISSUES:	None

{Artech Systems}

Author (s): _____ Reagile Matloga Date: 2021/05/01 _____
 Version: _____ 2.0 _____

USE CASE NAME:	Search Class Teacher	USE CASE TYPE Business Requirements: ☐ • System Analysis: ☒ System Design: ☐ •
USE CASE ID:	1.24	
PRIORITY:	Medium	
SOURCE:	Artech System Requirements List	
PRIMARY BUSINESS ACTOR	Curator(PBA)	
PRIMARY SYSTEM ACTOR	None	
OTHER PARTICIPATING ACTORS:	None	
OTHER INTERESTED STAKEHOLDERS:	None	
DESCRIPTION:	None	
PRE-CONDITION:	This use case describes the process where Curator wants to search for an Art Class Teacher. The Curator will view the list of saved Art Class teachers and search for the specific teacher. The system will search the ClassTeacher table for an Art Class Teacher based on what was entered in the search parameters. It will conclude when the search result of the requested Art Class Teacher is presented.	
TRIGGER:	The Curator wants to search for an Art Class Teacher.	
TYPICAL COURSE OF EVENTS:	Actor Action	System Response
	Step 1: The curator wants to search for an Art Class Teacher.	Step2: The system prompts the user to enter either one of the following search parameters: Name Surname Email Address Medium

	Step3: The curator inputs the search parameters into the system.	Step4: The system captures and validates that the search parameters are in the proper format and fully completed [ALT]
		Step5: The system retrieves and displays the search result based on the search parameters from Class_Teacher table. [ALT]
	Step6: The Curator selects the employee from search results.	Step7: The system displays the following details of the Art Class Teacher from the Class_Teacher table: • ClassTeacher_Name • ClassTeacher_Surname • ClassTeacher_EmailAddress • ClassTeacher_Medium • ClassTeacher_AddressLine1 • ClassTeacher_AddressLine2
ALTERNATE COURSES:	ALT STEP 4: The filled in search parameters are not in the correct format or not completed. • The system displays a validation error to the Curator.	
	ALT STEP 5: There is no Art Class Teacher that corresponds with what was searched for in the search parameters. • The notifies the Curator that no search results were found in the system • Terminates the use case.	
CONCLUSION:	This use case concludes when the curator is presented with the search results.	
POST-CONDITION:	The Class_Teacher table was searched with the entry that corresponds with the search parameters.	
BUSINESS RULES	• Only the curator can search the different art class teachers.	
IMPLEMENTATION CONSTRAINTS AND SPECIFICATIONS	• None	
ASSUMPTIONS:	• None	
OPEN ISSUES:	None	

{Artech Systems}

Author (s): _____ Reagile Matloga _____ Date: _____ 2021/05/01 _____
Version: _____ 2.0 _____

USE CASE NAME:	Update Class Teacher	USE CASE TYPE
USE CASE ID:	1.25	Business Requirements: ☐ •
PRIORITY:	Low	System Analysis: ☒
SOURCE:	Artech Systems Requirements List	System Design: ☐ •
PRIMARY BUSINESS ACTOR	System Administrator (PBA)	
PRIMARY SYSTEM ACTOR	Curator (PSA)	
OTHER PARTICIPATING ACTORS:	Teacher (ERA)	
OTHER INTERESTED STAKEHOLDERS:	None	
DESCRIPTION:	This use case describes the process where the curator updates the Art Class Teacher details on the system. The use case begins when the curator requests to update and make changes to the Art Class Teacher details. The system will display the Art Class teacher details from the Class_Teacher table. The curator can update the Class teacher . The curator will update the necessary changes and the system will update the Class_Teacher table	
PRE-CONDITION:	The Curator must be logged into the system. The Class_Teacher the curator wants to update must already exist in the system.	
TRIGGER:	The curator wants to update an art class teacher.	
TYPICAL COURSE OF EVENTS:	Actor Action	System Response
	Step1: The teacher wants the curator to update an art class teacher detail.	Step2: The system invokes use case 1.24 "Search Class Teacher"
		Step3: The system will ask the curator to update any one of the following: - ClassTeacher Name - ClassTeacher Surname - ClassTeacher Email Address - ClassTeacher Medium - ClassTeacher AddressLine1 - ClassTeacher AddressLine2
	Step4: The Art class teacher will state which details they want updated	Step5: The curator updates the details given.
		Step6: The system captures and validates the changes applied by checking whether all fields have been filled in and done correctly. [ALT]
		Step7: The system will prompt the curator to confirm the changes to the changes they made.
		Step8: The Curator asks the Art Class teacher to confirm changes.
	Step9: The Art Class Teacher confirms the changes made to the art class teacher details. [ALT]	Step8: The system will save the changes made to the Class_teacher table in the database: - ClassTeacher_ID - ClassTeacher_Name - ClassTeacher_Surname - ClassTeacher_EmailAddress - ClassTeacher_AddressLine1 - ClassTeacher_AddressLine2

		ClassTeacher_Medium
		Step9: The system will notify the changes have been successfully made.
ALTERNATE COURSES:	ALT STEP 6: The details the curator has changed are not in the correct format and not all fields are completed. - The system displays a validation error to the user. - Returns to Step 3.	
	ALT STEP 9: The Art Class teacher r does not confirm the changes made to the art class teacher details Returns to Step 3.	
CONCLUSION:	The use case concludes when the curator is notified by the system that changes have been made successfully.	
POST-CONDITION:	The updated Art Class information is saved in the Class_Teacher table.	
BUSINESS RULES	- Only the Curator can update Art Class details. - The Art Class teacher must already be in the system to be updated.	
IMPLEMENTATION CONSTRAINTS AND SPECIFICATIONS	None	
ASSUMPTIONS:	None	
OPEN ISSUES:	None	

{Artech Systems}

Author (s): _____ Reagile Matloga _____ Date: _____ 2021/05/01 _____
Version: _____ 2.0 _____

USE CASE NAME:	Delete Class Teacher		USE CASE TYPE Business Requirements: ☐ • System Analysis: ☒ System Design: ☐ •
USE CASE ID:	1.26		
PRIORITY:	Low		
SOURCE:	Artech Systems Requirements List		
PRIMARY BUSINESS ACTOR	Curator (PBA)		
PRIMARY SYSTEM ACTOR	None		
OTHER PARTICIPATING ACTORS:	None		
OTHER INTERESTED STAKEHOLDERS:	None		
DESCRIPTION:	This use case describes how the Curator removes an Art Class Teacher from the list of teachers in the system. The system will display a list of all Class teachers on the system. The Curator will delete the specific Class Teacher from the schedule.		
PRE-CONDITION:	- The Curator must be logged into the system. - The Class Teacher must exist on the system already.		
TRIGGER:	The Curator wants to delete an art class teacher they have created.		
TYPICAL COURSE OF EVENTS:	Actor Action	System Response	
	Step 1: The curator wants to delete/remove an art class teacher they have created.	Step 2: System displays the exiting Class Teachers.	
	Step 3: The Curator selects the Art Class Teacher to be deleted.	Step4: The System asks the curator if they are certain they want to delete the teacher.	
	Step5: The Curator confirms to delete the Art Class Teacher. [ALT]	Step6: The system deletes the description of the Art Class Reminder from the ClassTeacher table.	
		Step7: The system notifies the curator that the art class teacher has been deleted successfully.	
ALTERNATE COURSES:	ALT Step 5: The Curator does not confirm to delete the Art ClassTeacher.		
	Terminate the use case.		
CONCLUSION:	This use case concludes when the Curator is informed that the Art Class Teacher has been deleted successfully.		
POST-CONDITION:	The details of the art class teacher are removed from the table.		

BUSINESS RULES	Only the Curator can delete the Art Class teacher.
IMPLEMENTATION CONSTRAINTS AND SPECIFICATIONS	None
ASSUMPTIONS:	None
OPEN ISSUES:	None

{Artech Systems}

Author (s): Trishla Pathak Date: 2021/05/01
Version: 2.0

USE CASE NAME:	Create Medium Detail	USE CASE TYPE Business Requirements: " System Analysis: " x System Design: "	
USE CASE ID:	1.28		
PRIORITY:	High		
SOURCE:	Artech Systems Requirement List		
PRIMARY BUSINESS ACTOR	Curator (PBA)		
PRIMARY SYSTEM ACTOR	None		
OTHER PARTICIPATING ACTORS:	None		
OTHER INTERESTED STAKEHOLDERS:	None		
DESCRIPTION:	The use case describes the event when a curator creates a medium detail to be used in order to be able to search for artworks. The Curator enters the medium details and the system then creates a new medium with details.		
PRE-CONDITION:	Artwork must exist in order for mediums to be assigned to them.		
TRIGGER:	Curator wants to create Medium Detail.		
TYPICAL COURSE OF EVENTS:	Actor Action	System Response	
	1. The curator requests to create a medium detail.	2. The system displays the Create Medium page.	
	3. The Curator enters information of the medium.	4. System Validates if the Details entered are right. Allows Curator to proceed. [alt]	
	5. Curator confirms the process by submitting medium detail. [alt]	6. The system captures information into the MediumType Table as follows: [alt] - Medium Type ID - Medium Type Description Display Message confirming details being saved.	
		7. Medium successfully created. Shows Medium Details created. [alt]	
ALTERNATE COURSES:	Alt 4: System Validates the entered details and displays error message asking to fill in the correct information in the appropriate format. Return to Step 3		
	Alt 5: Curator cancels the process. Use Case is Terminated.		

	Alt 6: System unable to capture details due to system error. Show Error Message. Return to Step 3.
	Alt 7: Medium could not be created. Error Message is shown. Terminate UseCase.
CONCLUSION:	Medium details are successfully created and added to the system.
POST-CONDITION:	The medium details should be part of an artwork.
BUSINESS RULES	- Artwork must exist in order to create a medium. - Only Curator can create Medium. - The medium must not exist in the system
IMPLEMENTATION CONSTRAINTS AND SPECIFICATIONS	None
ASSUMPTIONS:	None
OPEN ISSUES:	None

{Artech Systems}

Author (s): Trishla Pathak Date: 2021/05/01
Version: 2.0

USE CASE NAME:	Search Medium	USE CASE TYPE Business Requirements: "

		7. System searches for medium and displays the artworks related to that medium. [alt]
ALTERNATE COURSES:	Alt 4: System Validates the entered details and displays error message asking to fill in the correct information in the appropriate format. Return to Step 3. Alt 5: Curator cancels the process. Use Case is Terminated. Alt 6: System unable to extract medium because medium does not exist. Return to Step 3. Alt 7: System unable to display Medium due to System Error. Shows Error Message. Return to Step 3.	
CONCLUSION:	Artworks related to that medium are displayed.	
POST-CONDITION:	The Medium should exist in order to be able to search for it.	
BUSINESS RULES	- Artwork must exist in order to search for a medium. - Medium must exist on the system to be found - Only curator can search for a Medium	
IMPLEMENTATION CONSTRAINTS AND SPECIFICATIONS	None	
ASSUMPTIONS:	None	
OPEN ISSUES:	None	

{Artech Systems}

Author (s): Trishla Pathak Date: 2021/05/01
 Version: 2.0

USE CASE NAME:	Update Medium Detail	USE CASE TYPE Business Requirements: "" System Analysis: "" System Design: ""	
USE CASE ID:	1.30		
PRIORITY:	Low		
SOURCE:	Artech Systems Requirement List		
PRIMARY BUSINESS ACTOR	Curator (PBA)		
PRIMARY SYSTEM ACTOR	None		
OTHER PARTICIPATING ACTORS:	None		
OTHER INTERESTED STAKEHOLDERS:	None		
DESCRIPTION:	The use case describes the event when a curator wants to update the medium detail to be used in order to be able to search for artworks. The Curator enters the medium description and the system then updates the medium with details.		
PRE-CONDITION:	Medium should exist in order to be updated.		
TRIGGER:	Curator wants to update for the medium details.		
TYPICAL COURSE OF EVENTS:	Actor Action		System Response
	1. Curator wants to update the medium details.		2. System Invokes Use case 1.29 search medium and displays the update medium page.
	3. Curator selects medium to update.		4. The system displays medium to update by extracting information from

		the MediumType Table as follows: - Medium Type ID - Medium Type Description
	5. Curator enters details that need to be updated.	6. System Validates if the Details entered are right. Allows Curator to proceed. [alt]
	7. Curator confirms the process by submitting medium detail. [alt]	8. System successfully captures updated information into MediumType Table as follows: - Medium Type ID - Medium Type Description
		9. System shows message of medium being successfully updated. [alt]
ALTERNATE COURSES:	Alt 6: System Validates the entered details and displays error message asking to fill in the correct information in the appropriate format. Return to Step 5 Alt 7: Curator cancels the process. Use Case is Terminated. Alt 9: System Shows Message of Medium details not being able to be Updated. Return to Step 7.	
CONCLUSION:	Medium details are updated.	
POST-CONDITION:	Artworks must contain updated medium details.	
BUSINESS RULES	- Artwork must exist in order to update a medium - Only curator can update a Medium. - A medium must exist on the system to be updated.	
IMPLEMENTATION CONSTRAINTS AND SPECIFICATIONS	None	
ASSUMPTIONS:	None	
OPEN ISSUES:	None	

{Artech Systems}

Author (s): Trishla Pathak Date: 2021/05/01
Version: 2.0

USE CASE NAME:	Delete Medium	USE CASE TYPE Business Requirements: " System Analysis: "x System Design: "
USE CASE ID:	1.31	
PRIORITY:	Low	
SOURCE:	Artech Systems Requirement List	
PRIMARY BUSINESS ACTOR	Curator (PBA)	
PRIMARY SYSTEM ACTOR	None	
OTHER PARTICIPATING ACTORS:	• None	
OTHER INTERESTED STAKEHOLDERS:	• None	

DESCRIPTION:	The use case describes the event when a curator wants to delete the medium. The Curator searches for the medium and selects to delete it. The system then deletes the medium.	
PRE-CONDITION:	Medium should exist in order to be deleted.	
TRIGGER:	Curator wants to delete the medium.	
TYPICAL COURSE OF EVENTS:	Actor Action	System Response
	1. Curator requests to delete medium.	2. System Invokes Use case 1.29 search medium and displays all the mediums
	3. Curator selects the mediums that need to be deleted and submits it.	4. System displays message asking if the curator confirms deleting the medium.
	5. Curator confirms deletion and submits selection. [alt]	6. System successfully deletes the medium from the MediumType table as follows: - Medium Type ID - MediumType Description
		7. System shows message of medium being successfully deleted. [alt]
ALTERNATE COURSES:	Alt 5: Curator cancels the process. Use Case is Terminated.	
	Alt 7: System Shows Message of medium not being able to be deleted. Terminate UseCase.	
CONCLUSION:	Medium deleted.	
POST-CONDITION:	Artworks containing this medium must be updated.	
BUSINESS RULES	Only curator can delete a medium from the system	
IMPLEMENTATION CONSTRAINTS AND SPECIFICATIONS	None	
ASSUMPTIONS:	None	
OPEN ISSUES:	None	

{Artech Systems}

Author (s): Trishla Pathak Date: 2021/05/01
Version: 2.0

USE CASE NAME:	Create Artwork Detail	USE CASE TYPE Business Requirements: " System Analysis: "x System Design: "	
USE CASE ID:	1.32		
PRIORITY:	High		
SOURCE:	Artech Systems Requirement List		
PRIMARY BUSINESS ACTOR	Curator (PBA)		
PRIMARY SYSTEM ACTOR	None		
OTHER PARTICIPATING ACTORS:	• None		
OTHER INTERESTED STAKEHOLDERS:	• None		
DESCRIPTION:	The use case describes the event when a curator wants to create an Artwork. The Curator creates the artwork by entering artwork details. The system captures the information and creates the artwork detail, attaching it to the specific artwork.		

PRE-CONDITION:	Artwork needs to exist in order to create Artwork Details	
TRIGGER:	Curator wants to create Artwork Details.	
TYPICAL COURSE OF EVENTS:	Actor Action	System Response
	1. Curator requests to create Artwork details.	2. System displays the Art Work create page.
	3. Curator enters details of the art work.	4. System Validates if the Details entered are right. Allows Curator to proceed. [alt]
	5. Curator confirms the process by submitting medium detail. [alt]	6. System Captures the artwork details in the Artwork table as follows: [alt] • Artwork ID • Artwork Title • Price • Art Picture • Artwork Status • Surface Type ID • Medium Type • Exhibition ID • Artist ID • Frame Colour ID • Frame Dimension ID
		7. System successfully creates the artwork details. Shows Artwork Details created. [alt]
ALTERNATE COURSES:	Alt 4: System Validates the entered details and displays error message asking to fill in the correct information in the appropriate format. Return to Step 3	
	Alt 5: Curator cancels the process. Use Case is Terminated.	
	Alt 6: System unable to capture details of the Artwork due to system Error. Error Message is shown. Return to Step 3	
	Alt 7: Medium could not be created. Error Message is shown. Terminate UseCase.	
CONCLUSION:	Artwork Details successfully created.	
POST-CONDITION:	Artwork details are attached to the specific artwork.	
BUSINESS RULES	• Every artwork has unique artwork details. • Only Curator can create a new artwork with details • Artwork must not already exist on the system.	
IMPLEMENTATION CONSTRAINTS AND SPECIFICATIONS	• None	
ASSUMPTIONS:	• None	
OPEN ISSUES:	None	

{Artech Systems}

Author (s): Trishla Pathak Date: 2021/05/01
Version: 2.0

USE CASE NAME:	Search Artwork	USE CASE TYPE	
USE CASE ID:	1.33	Business Requirements: "	
PRIORITY:	High	System Analysis: "x	
SOURCE:	Artech Systems Requirement List	System Design: "	
PRIMARY BUSINESS ACTOR	Curator (PBA)		
PRIMARY SYSTEM ACTOR	None		
OTHER PARTICIPATING ACTORS:	None		
OTHER INTERESTED STAKEHOLDERS:	None		
DESCRIPTION:	The use case describes the event when a curator wants to search for an Artwork. The Curator searches for the Artwork by entering the artwork details. The system captures the information and displays the artwork.		
PRE-CONDITION:	Artwork needs to exist in order to search for Artwork.		
TRIGGER:	Curator wants to search for an Artwork.		
TYPICAL COURSE OF EVENTS:	Actor Action	System Response	
	1. Curator requests to search for Artwork.	2. System displays the Art Work page.	
	3. Curator enters details of the art work.	4. System Validates if the Details entered are right. Allows Curator to proceed. [alt]	
	5. Curator confirms the process by submitting art work detail. [alt]	6. System Captures and extracts the artwork details in the Artwork table as follows: [alt] - Artwork ID - Artwork Title - Price - Art Picture - Artwork Status - Surface Type ID - Medium Type - Exhibition ID - Artist ID - Frame Colour ID - Frame Dimension ID Display message of item found	
		7. System successfully searches and displays the artwork. [alt]	
ALTERNATE COURSES:	Alt: 4: System Validates the entered details and displays error message asking to fill in the correct information in the appropriate format. Return to Step 3.		
	Alt 5: Curator cancels the process. Use Case is Terminated.		
	Alt 6: System unable to extract artworks because artwork does not exist. Return to Step 3.		
	Alt 7: System unable to display Artwork due to System Error. Shows Error Message. Return to Step 3.		
CONCLUSION:	Artwork successfully displayed.		
POST-CONDITION:	Artwork is viewed.		
BUSINESS RULES	Every artwork has unique artwork details.		

	Artwork must exist on the system to be found.
IMPLEMENTATION CONSTRAINTS AND SPECIFICATIONS	None
ASSUMPTIONS:	None
OPEN ISSUES:	None

{Artech Systems}

Author (s): Trishla Pathak Date: 2021/05/01

Version: 2.0

USE CASE NAME:	Update Artwork Detail	USE CASE TYPE Business Requirements: "" System Analysis: "" x System Design: ""
USE CASE ID:	1.34	
PRIORITY:	Low	
SOURCE:	Artech Systems Requirement List	
PRIMARY BUSINESS ACTOR	Curator (PBA)	
PRIMARY SYSTEM ACTOR	None	
OTHER PARTICIPATING ACTORS:	• None	
OTHER INTERESTED STAKEHOLDERS:	• None	
DESCRIPTION:	The use case describes the event when a curator wants to update the Artwork details. The Curator searches for the Artwork by entering the artwork details. The system captures the information and prompts the curator to enter information that is to be updated. The system then successfully updates the Artwork Details.	
PRE-CONDITION:	Artwork Details need to exist in order to update Artwork Details	
TRIGGER:	Curator wants to update Artwork Details.	
TYPICAL COURSE OF EVENTS:	Actor Action	System Response
	1. Curator requests to Update Artwork details.	2. System Invokes use case 1.33 search artwork and displays the Art Work page.
	3. Curator selects artwork to be updated.	4. System Captures and extracts the artwork details from the Artwork table as follows: • Artwork ID • Artwork Title • Price • Art Picture • Artwork Status • Surface Type ID • Medium Type • Exhibition ID • Artist ID • Frame Colour ID • Frame Dimension ID
	5. Curator enters details that need to be updated.	6. System Validates if the Details entered are right. Allows Curator to proceed. [alt]

	7. Curator confirms the process by submitting medium detail. [alt]	8. System successfully captures updated information into Artwork table as follows: • Artwork ID • Artwork Title • Price • Art Picture • Artwork Status • Surface Type ID • Medium Type • Exhibition ID • Artist ID • Frame Colour ID • Frame Dimension ID
		9. System shows message of artwork being successfully updated. [alt]
ALTERNATE COURSES:	Alt 6: System Validates the entered details and displays error message asking to fill in the correct information in the appropriate format. Return to Step 5 Alt 7: Curator cancels the process. Use Case is Terminated. Alt 9: System Shows Message of Artwork not being able to be Updated. Return to Step 7.	
CONCLUSION:	Artwork Details successfully updated.	
POST-CONDITION:	Artwork details attached to the specific artwork are updated.	
BUSINESS RULES	• Only Curator can update Artwork details • Artwork must exist on the system to be updated	
IMPLEMENTATION CONSTRAINTS AND SPECIFICATIONS	• None	
ASSUMPTIONS:	• None	
OPEN ISSUES:	None	

{Artech Systems}

Author (s): Trishla Pathak Date: 2021/05/01
Version: 2.0

USE CASE NAME:	Delete Artwork	USE CASE TYPE Business Requirements: "" System Analysis: "" System Design: ""
USE CASE ID:	1.35	
PRIORITY:	Low	
SOURCE:	Artech Systems Requirement List	
PRIMARY BUSINESS ACTOR	Curator (PBA)	
PRIMARY SYSTEM ACTOR	None	
OTHER PARTICIPATING ACTORS:	• None	
OTHER INTERESTED STAKEHOLDERS:	• None	
DESCRIPTION:	The use case describes the event when a curator wants to delete an Artwork. The Curator searches for the artwork by entering artwork details. The system captures the information and displays the artwork. The Curator then selects which artwork is to be deleted, and the system deletes the selected artwork.	

PRE-CONDITION:	Artwork needs to exist in order to be deleted	
TRIGGER:	Curator wants to delete the Artwork.	
TYPICAL COURSE OF EVENTS:	Actor Action	System Response
	1. Curator requests to delete Artwork details.	2. System Invokes Use case 1.33 Search artwork and displays the Art Work based on entered details to be deleted.
	3. Curator selects artwork to be deleted and submits the selection to be deleted.	4. System displays message asking if the curator confirms deleting the medium.
	5. Curator confirms deletion and submits selection. [alt]	6. System successfully deletes the artwork detail from the Artwork table as follows: • Artwork ID • Artwork Title • Price • Art Picture • Artwork Status ID • Surface Type ID • Medium Type • Exhibition ID • Artist ID • Frame Colour ID • Frame Dimension ID
		7. System shows message of artwork being successfully deleted. [alt]
ALTERNATE COURSES:	Alt 5: Curator cancels the process. Use Case is Terminated.	
	Alt 7: System Shows Message of Artwork not being able to be deleted. Terminate UseCase.	
CONCLUSION:	Artwork successfully deleted.	
POST-CONDITION:	Artwork details attached to that specific artwork are also deleted.	
BUSINESS RULES	• Only the curator can delete Artwork from the system.	
IMPLEMENTATION CONSTRAINTS AND SPECIFICATIONS	• None	
ASSUMPTIONS:	• None	
OPEN ISSUES:	None	

{Artech Systems}

Author (s): Trishla Pathak Date: 2021/05/01
Version: 2.0

USE CASE NAME:	Archive Status	USE CASE TYPE Business Requirements: "" System Analysis: "" System Design: ""
USE CASE ID:	1.36	
PRIORITY:	High	
SOURCE:	Artech Systems Requirement List	
PRIMARY BUSINESS ACTOR	Curator (PBA)	
PRIMARY SYSTEM ACTOR	None	

OTHER PARTICIPATING ACTORS:	None						
OTHER INTERESTED STAKEHOLDERS:	Artist (ERA)						
DESCRIPTION:	This use case describes an event that explains the status of an artwork once it comes into the gallery. Based on where and how the artwork is used physically or digitally, the curator can change the status of the artwork within the system.						
PRE-CONDITION:	Artwork must be in the gallery.						
TRIGGER:	Curator receives the artwork and wants to update artwork status.						
TYPICAL COURSE OF EVENTS:	<table> <tr> <th>Actor Action</th><th>System Response</th></tr> <tr> <td>1. Curator exhibits the artwork in an exhibition and requests to change status. [alt]</td><td>2. System displays artwork details.</td></tr> <tr> <td>3. Curator updates artwork status to "Exhibition" [alt]</td><td>4. System successfully captures information in the ArtworkStatus table as follows: [alt] - Artwork Status ID - Artwork Status Description Displays status.</td></tr> </table>	Actor Action	System Response	1. Curator exhibits the artwork in an exhibition and requests to change status. [alt]	2. System displays artwork details.	3. Curator updates artwork status to "Exhibition" [alt]	4. System successfully captures information in the ArtworkStatus table as follows: [alt] - Artwork Status ID - Artwork Status Description Displays status.
Actor Action	System Response						
1. Curator exhibits the artwork in an exhibition and requests to change status. [alt]	2. System displays artwork details.						
3. Curator updates artwork status to "Exhibition" [alt]	4. System successfully captures information in the ArtworkStatus table as follows: [alt] - Artwork Status ID - Artwork Status Description Displays status.						
ALTERNATE COURSES:	Alt 1: Curator displays artwork on the wall where it stays for one week only. Alt 1: Curator stores artwork in the storeroom. Alt 1: Curator requests to return the artwork to the artist if the artwork has been displayed on the wall for over a week Alt 1.1: The artist gets an automatic e-mail when the status changes asking them to come pick their artwork. Alt 1: Curator returns the artwork to the artist. Alt 3: Curator updates artwork status to "Display" Alt 3: Curator updates artwork status to "Store Room" Alt 3: Curator updates artwork status to "Returned" Alt 3: Curator updates artwork status to "Request Return" Alt 4: System could not capture information successfully due to system error. Show error Message. Terminate Use Case.						
CONCLUSION:	Artwork Status changed.						
POST-CONDITION:	Artwork must exist in order to create an archive status.						
BUSINESS RULES	- Every artwork has an archive status attached to it. - Only the curator can change the archive status						
IMPLEMENTATION CONSTRAINTS AND SPECIFICATIONS	None						
ASSUMPTIONS:	None						
OPEN ISSUES:	None						

{Artech Systems}

Author (s): Trishla Pathak Date: 2021/05/01
Version: 2.0

USE CASE NAME:	Create Surface Type Details	USE CASE TYPE	
USE CASE ID:	1.37	Business Requirements: "	
PRIORITY:	High	System Analysis: "x	
SOURCE:	Artech Systems Requirement List	System Design: "	
PRIMARY BUSINESS ACTOR	Curator (PBA)		
PRIMARY SYSTEM ACTOR	None		
OTHER PARTICIPATING ACTORS:	None		
OTHER INTERESTED STAKEHOLDERS:	None		
DESCRIPTION:	The use case describes the event when a curator creates a surface type detail to be used in order to be able to search for artworks. The Curator enters the surface type details and the system then creates a new surface type with details.		
PRE-CONDITION:	Artwork must exist in order for surface types to be assigned to them.		
TRIGGER:	Curator wants to create Surface type Details		
TYPICAL COURSE OF EVENTS:	Actor Action	System Response	
	1. The curator requests to create surface type detail.	2. The system displays the Create Surface type page.	
	3. The Curator enters information of the surface type.	4. System Validates if the Details entered are right. Allows Curator to proceed. [alt]	
	5. Curator confirms the process by submitting surface type details [alt]	6. The system captures information into the SurfaceTypeTable as follows: [alt] - Surface Type ID - Surface Type Description	
		7. Surface Type successfully created. Shows Confirmation Message. [alt]	
ALTERNATE COURSES:	Alt 4: System Validates the entered details and displays error message asking to fill in the correct information in the appropriate format. Return to Step 3		
	Alt 5: Curator cancels the process. Use Case is Terminated.		
	Alt 6: System unable to capture details due to system error. Return to Step 3.		
	Alt 7: Surface Type could not be created. Error Message is shown. Terminate UseCase.		
CONCLUSION:	Surface Type details are successfully created and added to the system.		
POST-CONDITION:	The surface type details should be part of an artwork.		
BUSINESS RULES	- Artwork must exist in order to create a surface type - Only Curator can create surface type - The Surface type must not exist in the system		
IMPLEMENTATION CONTRAINTS AND SPECIFICATIONS	None		
ASSUMPTIONS:	None		
OPEN ISSUES:	None		

{Artech Systems}

Author (s): Trishla Pathak Date: 2021/05/01
Version: 2.0

USE CASE NAME:	Search Surface Type	USE CASE TYPE Business Requirements: " System Analysis: " x System Design: "	
USE CASE ID:	1.38		
PRIORITY:	High		
SOURCE:	Artech Systems Requirement List		
PRIMARY BUSINESS ACTOR	Curator (PBA)		
PRIMARY SYSTEM ACTOR	None		
OTHER PARTICIPATING ACTORS:	None		
OTHER INTERESTED STAKEHOLDERS:	None		
DESCRIPTION:	The usecase describes the event when a curator searches for a surface type to be used in order to be able to search for artworks. The Curator enters the surface type details and the system then searches for the surface type based on that.		
PRE-CONDITION:	Artwork must exist in order for surface types to be assigned to them.		
TRIGGER:	Curator wants to search for the surface type .		
TYPICAL COURSE OF EVENTS:	Actor Action	System Response	
	1. The curator requests to search for surface type in order to find classes or artwork	2. The system displays the Search surface type page.	
	3. The Curator enters details of the surface type.	4. System Validates if the Details entered are right. Allows Curator to proceed. [alt]	
	5. Curator confirms the process by submitting surface type detail. [alt]	6. The system extracts and captures information from the SurfaceType Table as follows: [alt] - Surface Type ID - Surface Type Description Display message of item found	
		7. System successfully searches for surface type and displays the artworks related to that surface type. [alt]	
ALTERNATE COURSES:	Alt: 4: System Validates the entered details and displays error message asking to fill in the correct information in the appropriate format. Return to Step 3		
	Alt 5: Curator cancels the process. Use Case is Terminated.		
	Alt 6: System unable to extract surface type because surface type does not exist. Return to Step 3.		
	Alt 7: System unable to display artworks or art classes due to System Error. Error Message Shows. Return to step 3.		
CONCLUSION:	Artworks related to that surface type are displayed.		

POST-CONDITION:	The surface type should exist in order to be able to search for it.
BUSINESS RULES	- Artwork must exist in order to search for a surface type. - Only curator can search for a surface type - A surface type must exist on the system to be found
IMPLEMENTATION CONSTRAINTS AND SPECIFICATIONS	None
ASSUMPTIONS:	None
OPEN ISSUES:	None

{Artech Systems}

Author (s): Trishla Pathak Date: 2021/05/01
Version: 2.0

USE CASE NAME:	Update surface type Detail	USE CASE TYPE Business Requirements: " System Analysis: "x System Design: "	
USE CASE ID:	1.39		
PRIORITY:	Low		
SOURCE:	Artech Systems Requirement List		
PRIMARY BUSINESS ACTOR	Curator (PBA)		
PRIMARY SYSTEM ACTOR	None		
OTHER PARTICIPATING ACTORS:	None		
OTHER INTERESTED STAKEHOLDERS:	None		
DESCRIPTION:	The usecase describes the event when a curator wants to update the surface type detail to be used in order to be able to search for artworks. The Curator enters the surface type description and the system then updates the surface type with details.		
PRE-CONDITION:	Medium should exist in order to be updated.		
TRIGGER:	Curator wants to update for the medium details.		
TYPICAL COURSE OF EVENTS:	Actor Action		System Response
	1. Curator wants to update the surface type details		2. System Invoke usecase 1.38 Search Surface Type and displays the update surface type page.
	3. Curator selects surface type to update.		4. The system extracts information from the SurfaceType Table as follows: Surface Type ID SurfaceType Description
	5. Curator enters information of the surface type that need to be updated.		6. System Validates if the Details entered are right. Allows Curator to proceed. [alt]

	7. Curator confirms the process by submitting medium detail. [alt]	8. System successfully captures updated information into SurfaceType Table as follows: - Surface Type ID - Medium Type Description
		9. System shows message of surface type being successfully updated. [alt]
	Alt 6 System Validates the entered details and displays error message asking to fill in the correct information in the appropriate format. Return to Step 5	
	Alt 7: Curator cancels the process. Use Case is Terminated	
	Alt 9: System Shows Message of Surface type details not being able to be Updated. Return to Step 7	
CONCLUSION:	Surface type details are updated.	
POST-CONDITION:	Artworks must contain updated surface type details.	
BUSINESS RULES	- Artwork must exist in order to update a surface type - Only curator can update a surface type - A surface type must exist on the system to be updated.	
IMPLEMENTATION CONSTRAINTS AND SPECIFICATIONS	None	
ASSUMPTIONS:	None	
OPEN ISSUES:	None	

{Artech Systems}

Author (s): Trishla Pathak Date: 2021/05/01

Version: 2.0

USE CASE NAME:	Delete Surface type	USE CASE TYPE Business Requirements: "" System Analysis: "" x System Design: ""	
USE CASE ID:	1.40		
PRIORITY:	Low		
SOURCE:	Artech Systems Requirement List		
PRIMARY BUSINESS ACTOR	Curator (PBA)		
PRIMARY SYSTEM ACTOR	None		
OTHER PARTICIPATING ACTORS:	None		
OTHER INTERESTED STAKEHOLDERS:	None		
DESCRIPTION:	The usecase describes the event when a curator wants to delete the surface type. The Curator searches for the surface type and selects to delete it. The system then deletes the surface type.		
PRE-CONDITION:	Surface type should exist in order to be deleted.		
TRIGGER:	Curator wants to delete the surface type.		
TYPICAL COURSE OF EVENTS:	Actor Action	System Response	
	1. Curator wants to delete surface type.	2. System Invoke usecase 1.38 Search Surface Type and displays the surface type page.	

	3. Curator selects the surface type that needs to be deleted and submits it.	4. System displays message asking if the curator confirms deleting the surface type.
	5. Curator confirms deletion and submits selection. [alt]	6. System successfully deletes the surface type from the SurfaceType table as follows: [alt] - Surface type Type ID - Surface type Description
		7. System shows message of surface type being successfully deleted. [alt]
ALTERNATE COURSES:	Alt 5: Curator cancels the process. Use Case is Terminated. Alt 6: System unable to delete the surface type due to system error. Return to Step 3. Alt 7: System Shows Message of surface type not being able to be deleted. Terminate UseCase.	
CONCLUSION:	Surface type deleted.	
POST-CONDITION:	Artworks containing this surface type must be updated.	
BUSINESS RULES	- Artwork must exist in order to update a surface type - Only curator can delete a surface type from the system	
IMPLEMENTATION CONSTRAINTS AND SPECIFICATIONS	None	
ASSUMPTIONS:	None	
OPEN ISSUES:	None	

{Artech Systems}

Author (s): Trishla Pathak Date: 2021/05/01
Version: 2.0

USE CASE NAME:	View Art Class Feedback Form	USE CASE TYPE Business Requirements: " System Analysis: "x System Design: "	
USE CASE ID:	1.41		
PRIORITY:	High		
SOURCE:	Artech Systems Requirement List		
PRIMARY BUSINESS ACTOR	Curator (PBA)		
PRIMARY SYSTEM ACTOR	None		
OTHER PARTICIPATING ACTORS:	• None		
OTHER INTERESTED STAKEHOLDERS:	• None		
DESCRIPTION:	The usecase describes an event where the curator requests to view the Art Class Feedback form which is the feedback of the users on the art classes. The system provides the curator with the forms. The curator can then view the feedback forms.		
PRE-CONDITION:	User has entered a feedback form for an art class.		
TRIGGER:	Curator wants to view feedback form.		
TYPICAL COURSE OF EVENTS:	Actor Action	System Response	
	1. Curator requests to view feedback form.	2. System displays the art class feedback form page.	

	3. Curator selects the form she wants to view. [alt]	4. System successfully extracts and displays information of the form from the Feedback Table as follows: [alt] - Feedback Page ID - Public User ID - Artist ID - Art Class ID - Feedback Comment - Rating Teacher - Rating Food - Rating Difficulty - Overall Rating
ALTERNATE COURSES:	Alt 3: Curator cannot select any form to view as no form exists. Terminate Use Case. Alt 4: System unable to extract information due to system error. Show Error message. Terminate Use Case.	
CONCLUSION:	Feedback form is viewed.	
POST-CONDITION:	None	
BUSINESS RULES	One User can fill only one feedback form, once.	
IMPLEMENTATION CONSTRAINTS AND SPECIFICATIONS	None	
ASSUMPTIONS:	None	
OPEN ISSUES:	None	

2.2.2 Art Class Subsystem

Author (s): <u>Reinhardt Smit</u>		Artech Systems	
Date: <u>2021/05/03</u>		Version: <u>1.0</u>	
USE CASE NAME:	Book Art Class	USE CASE TYPE Business Requirements: ☐ • System Analysis: ☒ • System Design: ☐ •	
USE CASE ID:	2.1		
PRIORITY:	High		
SOURCE:	Binary Brains – Artech System Requirements		
PRIMARY BUSINESS ACTOR	User		
PRIMARY SYSTEM ACTOR	None		
OTHER PARTICIPATING ACTORS:	Curator (ERA)		
OTHER INTERESTED STAKEHOLDERS:	None		
DESCRIPTION:	This use case describes how the User uses the system to book for an Art class by firstly selecting the desired class they want to participate in. Once a class has been selected, they will see if there are any seats available. If a class has open seats, they request to make a booking. When making the booking they will provide all the required booking details into the system which is captured and validated. The system will prompt the user to make a payment to confirm their booking seat by		

	either making a 50% deposit or a full payment. Once payment is made all their booking information is stored into the system in the Booking table.	
PRE-CONDITION:	User must have an account and be signed into the system.	
TRIGGER:	User request to Book an Art Class	
TYPICAL COURSE OF EVENTS:	Actor Action Step 1: User request to book for a class. Step 3: User selects Book for Class. Step 5: User enters booking details. Step 8: User makes booking payment. [ALT] Step 10: User submits the Booking.	System Response Step 2: System captures request and invokes Use Case 2.2 View Existing Class Step 4: System captures action and prompts the user to enter booking details in the required input fields. Step 6: System captures all the details entered and validates each input field to determine if the correct information has been entered in the correct format and style. [ALT] Step 7: The system prompts the user to specify if they would like to make payment for booking and displays which payment option, they would like to make to confirm booking seat. Step 9: Invokes Use case 2.8 Make Payment Step 11: System captures all the information they entered and stores it in the Booking table with the following attributes, BookingID , PublicUserID , ArtClassID , Date and Time . [ALT] Step 12: System notifies the user that the Booking was successful.
ALTERNATE COURSES:	ALT-Step 6: Information entered in the input fields are invalid making it unavailable for the User to submit the form. - Return to Step 5. ALT-Step 8: User decides not to make any payment which will not confirm their seat but booking process proceeds. ALT-Step 11: Storing booking details was unsuccessful. - Return to Step 5.	
CONCLUSION:	User is successfully added to the class.	
POST-CONDITION:	The booking is stored in the Booking table in the database	
BUSINESS RULES	Must make a 50% deposit or full payment to confirm booking in art class	
IMPLEMENTATION CONSTRAINTS AND SPECIFICATIONS	System is offline	
ASSUMPTIONS:	None	
OPEN ISSUES:	None	

Author (s): Reinhardt Smit Artech Systems Date: 2021/05/03

Version: 1.0

USE CASE NAME:	View Existing Classes	USE CASE TYPE	
USE CASE ID:	2.2	Business Requirements: ☐	
PRIORITY:	High	System Analysis: ☒	
SOURCE:	Binary Brains – Artech System Requirements	System Design: ☐	
PRIMARY BUSINESS ACTOR	User		
PRIMARY SYSTEM ACTOR	None		
OTHER PARTICIPATING ACTORS:	None		
OTHER INTERESTED STAKEHOLDERS:	None		
DESCRIPTION:	This use case describes how the User uses the system to view all the details of an existing class on the system. The user selects the class that they want to view, system gathers all the class information from the Art_Class table.		
PRE-CONDITION:	Class must exist on the system		
TRIGGER:	User requests to View Art Class		
TYPICAL COURSE OF EVENTS:	Actor Action	System Response	
	Step 1: User requests to View Art Class.	Step 2: System Captures the action and displays all the available classes on the system by gather the list of Classes from the Art_Class table displaying all available classes by looping through each ArtClassID .	
	Step 3: User selects a class	Step 4: System gathers all information form the Art_Class table for that specific class using the ArtClassID to display the following attributes, ArtClassName , ArtClassDescription , ArtClassPeriod , ClassPrice , ArtClassType , ClassTeacher and Venue .	
		Step 5: System successfully displays all the Art Class information to the user.	
ALTERNATE COURSES:	None		
CONCLUSION:	System successfully displays Art Class		
POST-CONDITION:	The Art Class in the Art_Class table has been viewed		

BUSINESS RULES	None
IMPLEMENTATION CONSTRAINTS AND SPECIFICATIONS	System is offline
ASSUMPTIONS:	None
OPEN ISSUES:	None

Artech Systems
 Author (s): Reinhardt Smit Date: 2021/05/03 Version: 1.0

USE CASE NAME:	Cancel Booking	USE CASE TYPE Business Requirements: ☐ System Analysis: ☒ System Design: ☐
USE CASE ID:	2.3	
PRIORITY:	High	
SOURCE:	Binary Brains – Artech System Requirements	
PRIMARY BUSINESS ACTOR	User	
PRIMARY SYSTEM ACTOR	None	
OTHER PARTICIPATING ACTORS:	Curator (ERA)	
OTHER INTERESTED STAKEHOLDERS:	None	
DESCRIPTION:	This use case describes how the User uses the system to cancel an Art Class booking. The user selects the class they booked for and cancels the booking. A confirmation for cancel booking will be required. Once the user confirms cancellation, they will be removed from the booking in the Booking table.	
PRE-CONDITION:	- A Booking for a class must be made. - User must be logged in	
TRIGGER:	User requests to cancel their Booked class	
TYPICAL COURSE OF EVENTS:	Actor Action	System Response
	Step 1: User requests to cancel booking	Step 2: System captures request and invokes Use Case 2.2 View Existing Class
	Step 3: User cancels their booking	Step 4: System Captures the action and gathers Booking information from the Booking table and displays all the Booking details, Date and Time by BookingID .
		Step 5: System request confirmation to cancel the booking.
	Step 6: User confirms cancel booking [ALT]	Step 7: System captures the confirmation and removes the user from the Booking table by deleting all booking details BookingID , PublicUserID , ArtClassID , Date and Time .
		Step 8: System notifies that the user has successfully cancelled their booking.
ALTERNATE COURSES:	ALT-Step 3: User rejects confirm cancellation of booking. Use case is terminated	

CONCLUSION:	System successfully removes User from the art class.
POST-CONDITION:	Cancellation of booking has been successfully saved in the database.
BUSINESS RULES	Bookings can only be cancelled until 2 days before class.
IMPLEMENTATION CONSTRAINTS AND SPECIFICATIONS	System is offline
ASSUMPTIONS:	None
OPEN ISSUES:	None

Author (s): Reinhardt Smit Artech Systems Date: 2021/05/03 Version: 1.0

USE CASE NAME:	Cancel Booking	USE CASE TYPE Business Requirements: ☐ System Analysis: ☒ System Design: ☐
USE CASE ID:	2.3	
PRIORITY:	High	
SOURCE:	Binary Brains – Artech System Requirements	
PRIMARY BUSINESS ACTOR	User	
PRIMARY SYSTEM ACTOR	None	
OTHER PARTICIPATING ACTORS:	Curator (ERA)	
OTHER INTERESTED STAKEHOLDERS:	None	
DESCRIPTION:	This use case describes how the User uses the system to cancel an Art Class booking. The user selects the class they booked for and cancels the booking. A confirmation for cancel booking will be required. Once the user confirms cancellation, they will be removed from the booking in the Booking table.	
PRE-CONDITION:	- A Booking for a class must be made. - User must be logged in	
TRIGGER:	User requests to cancel their Booked class	
TYPICAL COURSE OF EVENTS:	Actor Action	System Response
	Step 1: User requests to cancel booking	Step 2: System captures request and invokes Use Case 2.2 View Existing Class
	Step 3: User cancels their booking	Step 4: System Captures the action and gathers Booking information from the Booking table and displays all the Booking details, Date and Time by BookingID .
		Step 5: System request confirmation to cancel the booking.
	Step 6: User confirms cancel booking [ALT]	Step 7: System captures the confirmation and removes the user form the Booking table by deleting all booking details BookingID , PublicUserID , ArtClassID , Date and Time .
		Step 8: System notifies that the user has successfully cancelled their booking.
ALTERNATE COURSES:	ALT-Step 3: User rejects confirm cancellation of booking. Use case is terminated	

CONCLUSION:	System successfully removes User from the art class.
POST-CONDITION:	Cancellation of booking has been successfully saved in the database.
BUSINESS RULES	Bookings can only be cancelled until 2 days before class.
IMPLEMENTATION CONSTRAINTS AND SPECIFICATIONS	System is offline
ASSUMPTIONS:	None
OPEN ISSUES:	None

Author (s): Reinhardt Smit Artech Systems Date: 2021/05/03 Version: 1.0

USE CASE NAME:	Generate Booking Confirmation Message	USE CASE TYPE	
USE CASE ID:	2.4	Business Requirements: ☐ • System Analysis: ☒ • System Design: ☐ •	
PRIORITY:	High		
SOURCE:	Binary Brains – Artech System Requirements		
PRIMARY BUSINESS ACTOR	User		
PRIMARY SYSTEM ACTOR	None		
OTHER PARTICIPATING ACTORS:	None		
OTHER INTERESTED STAKEHOLDERS:	None		
DESCRIPTION:	This use case describes how the User uses the system to generate a confirmation message of the class they booked for containing all the important details of that class. The system will gather all information for that specific booking using BookingID from the Booking table displaying all needed information for that class.		
PRE-CONDITION:	A Booking for a class must be made		
TRIGGER:	User requests Generate Booking Confirmation Message		
TYPICAL COURSE OF EVENTS:	Actor Action	System Response	
	Step 1: User requests Generate Booking.	Step 2: System captures request and invokes Use Case 2.2 View Existing Class	
	Step 3: User Generates Booking confirmation message.	Step 4: System Captures the action and gathers Booking information from the Booking table and Art_Class table and displays all the necessary details to the user, Date and Time , ArtClassPeriod , ArtClassType , VenueID , ClassTeacher and ArtClassName .	

		Step 5: System successfully generates a Booking Confirmation Message.
ALTERNATE COURSES:	None	
CONCLUSION:	System successfully generates a Booking Confirmation Message	
POST-CONDITION:	User can view all booking confirmation details.	
BUSINESS RULES	None	
IMPLEMENTATION CONSTRAINTS AND SPECIFICATIONS	System is offline	
ASSUMPTIONS:	None	
OPEN ISSUES:	None	

Author (s): Reinhardt Smit Artech Systems Date: 2021/05/03 Version: 1.0

USE CASE NAME:	Refund Status	USE CASE TYPE Business Requirements: ☐ • System Analysis: ☒ • System Design: ☐ •
USE CASE ID:	2.5	
PRIORITY:	High	
SOURCE:	Binary Brains – Artech System Requirements	
PRIMARY BUSINESS ACTOR	User	
PRIMARY SYSTEM ACTOR	None	
OTHER PARTICIPATING ACTORS:	• None	
OTHER INTERESTED STAKEHOLDERS:	• None	
DESCRIPTION:	This use case describes how the user uses the system to view their refund status on the system. The User selects the class they paid for and requests to view the class refund status. Once the refund status load is gathering all information from the Refund table and displays RefundStatus to the User by using their BookingID from the Booking table.	
PRE-CONDITION:	• A Booking for a class must be made. • User must have made a payment for the class	
TRIGGER:	User requests to view refund status	
TYPICAL COURSE OF EVENTS:	Actor Action Step 1: User requests to view refund status Step 3: User selects to view refund status	System Response Step 2: System captures request and invokes Use Case 2.2 View Existing Class Step 4: System Captures the action and gathers information from the Refund table displaying the RefundStatus for that specific User by finding their Booking in the Booking table through the PublicUserID and BookingID . Step 5: System successfully displays the refund status to the user.
ALTERNATE COURSES:	None	

CONCLUSION:	System successfully displays the refund status
POST-CONDITION:	User can view their refund status to keep track.
BUSINESS RULES	Can only refund for a class until 2 days before the class starts.
IMPLEMENTATION CONSTRAINTS AND SPECIFICATIONS	System is offline
ASSUMPTIONS:	None
OPEN ISSUES:	None

Author (s): Reinhardt Smit Artech Systems Date: 2021/05/03 Version: 1.0

USE CASE NAME:	Capture Class Feedback	USE CASE TYPE Business Requirements: ☐ System Analysis: ☒ System Design: ☐	
USE CASE ID:	2.6		
PRIORITY:	High		
SOURCE:	Binary Brains – Artech System Requirements		
PRIMARY BUSINESS ACTOR	User		
PRIMARY SYSTEM ACTOR	None		
OTHER PARTICIPATING ACTORS:	Curator (ERA)		
OTHER INTERESTED STAKEHOLDERS:	None		
DESCRIPTION:	This use case describes how the User uses the system to provide feedback on their experience in the class. Once the user completes a class the system will tag that class on the users account as completed opening the chance for the user to provide feedback for their class experience. User will be prompted to enter a comment, rate the teacher, difficulty of class and overall class rating in the required input fields. User will then submit this form and stores all feedback information in the Feedback table.		
PRE-CONDITION:	Must first finish their class experience. Must be logged into the system		
TRIGGER:	User requests to Provide Feedback		
TYPICAL COURSE	Actor Action	System Response	
OF EVENTS:	Step 1: User requests to Provide Feedback.	Step 2: System captures request and invokes Use Case 2.2 View Existing Class. Class is tagged as 'Completed' and opens feedback for class.	
	Step 3: User select provide feedback for their class.	Step 2: System Captures the action and prompts the user to enter information into the necessary input fields by providing a comment, Teacher, Class difficulty and overall class rating.	
	Step 3: User enters feedback details	Step 4: System captures feedback information in all necessary input fields and validates the input values to ensure all values are in the correct format. [ALT]	

	Step 5: User submits feedback form	Step 6: System prompts the user to confirm all details entered for feedback form.
	Step 7: User confirms feedback details. [ALT]	Step 8: System captures all entered information and stores a new feedback form in the Feedback table generating a new form with FeedbackPageID for that specific form and stores all necessary values in the following attributes, Feedbackcomment , RatingTeacher and RatingDifficulty . [ALT]
		Step 9: System notifies the user that the Feedback form is successfully submitted.
ALTERNATE COURSES:	ALT-Step 7: Rejects confirm feedback form. Terminates Use case.	
	ALT-Step 4: Information entered in the input fields are invalid making it unavailable for the User to submit the form. Return to Step 3.	
	ALT-Step 8: Storing booking details was unsuccessful. Return to Step 3.	
CONCLUSION:	User successfully submits a feedback form	
POST-CONDITION:	Feedback form is successfully stored in the Feedback table in the database.	
BUSINESS RULES	- Can only provide feedback once. - Has 2 days to provide feedback	
IMPLEMENTATION CONSTRAINTS AND SPECIFICATIONS	System is offline	
ASSUMPTIONS:	None	
OPEN ISSUES:	None	

Author (s): Reinhardt Smit Artech Systems Date: 2021/05/03

Version: 1.0

USE CASE NAME:	Generate Class Teacher infomation	USE CASE TYPE Business Requirements: ☐ ▪ System Analysis: ☒ ▪ System Design: ☐ ▪
USE CASE ID:	2.7	
PRIORITY:	High	
SOURCE:	Binary Brains – Artech System Requirements	
PRIMARY BUSINESS ACTOR	User	
PRIMARY SYSTEM ACTOR	None	
OTHER PARTICIPATING ACTORS:	• None	
OTHER INTERESTED STAKEHOLDERS:	• None	

DESCRIPTION:	This use case describes how the User uses the system to generate a class teacher information. The user firstly invokes Use case 3.2 View Existing class. Once the user views the class they have the option to view the teacher's details representing the class. System successfully displays all class teacher details to the user with the Class_Teacher and Artist table.	
PRE-CONDITION:	None	
TRIGGER:	User requests generate class teacher information	
TYPICAL COURSE OF EVENTS:	Actor Action Step 1: User requests generate class teacher information Step 3: User selects view class teacher information.	System Response Step 2: System captures request and invokes Use Case 2.2 View Existing Class . Step 4: System finds class teacher information from the Class_Teacher table ClassTeacherID , ArtistID , ArtClassID and displays the information through the Artist table with ArtistBio , ArtistBioPictures , Name and Surname .
ALTERNATE COURSES:	None	
CONCLUSION:	System successfully generates class teacher information	
POST-CONDITION:	User can view all class teacher information.	
BUSINESS RULES	None	
IMPLEMENTATION CONSTRAINTS AND SPECIFICATIONS	System is offline	
ASSUMPTIONS:	None	
OPEN ISSUES:	None	

Author (s): Reinhardt Smit Artech Systems Date: 2021/05/03

Version: 1.0

USE CASE NAME:	Make Payment	USE CASE TYPE Business Requirements: ☐ System Analysis: ☒ System Design: ☐
USE CASE ID:	2.8	
PRIORITY:	High	
SOURCE:	Binary Brains – Artech System Requirements	
PRIMARY BUSINESS ACTOR	User	
PRIMARY SYSTEM ACTOR	None	
OTHER PARTICIPATING ACTORS:	Bank (ESA) Virtual Card Services (ESA)	
OTHER INTERESTED STAKEHOLDERS:	None	
DESCRIPTION:	This use case describes how the User uses the system to make a payment for an art class they are booking or booked for to confirm their seat. This use case gets invoked by Use Case 2.1 Book Art Class if the user decides to make payment while making their booking, or when the user decides to make payment later, they are opened to that same class given the option to make payment. Once the user requests to make payment the system prompts the user to select a 50% deposit or make	

	full payment. Depending on the Class the system will generate an amount after the user selected their payment option from the Payment table and prompt the user to enter card details through a virtual card service on the system to complete the payment process. After payment has been made the users, seat has been confirmed for the class and a refund will be available for only until 2 days before the class starts.	
PRE-CONDITION:	A Booking for a class must be made	
TRIGGER:	User requests to make payment.	
TYPICAL COURSE OF EVENTS:	Actor Action Step 1: User requests to make payment. Step 3: User selects payment option. Step 5: User selects preferred payment method. Step 8: User enters card details Step 10: User submits card details Step 12: User confirms amount due. [ALT] Step 14: User enters one-time-pin	System Response Step 2: System captures request and displays Payment amount due for that specific class by displaying Price from the Art_Class table. System then prompts the user to select 50% deposit or make full payment option. [ALT] Step 4: System captures action and prompts the user to select a payment method. Step 6: System captures from the PaymentType table PaymentTypeID and PaymentTypeDescription and validates selection. System updates PaymentType in Payment table [ALT] Step 7: System loads payment gateway with the amount due for the Class and virtual card services prompts user for card details. Step 9: Virtual card services captures and validates card details. [ALT] Step 11: System captures action and Virtual card services prompts user to confirm amount due. Step 13: Virtual card services contacts bank and requests the user to enter the one-time-pin the Bank to allow payment from Bank. Step 15: Virtual card services captures and validates the one-time-pin. [ALT] Step 16: System captures payment and generates new PaymentID in Payment table and saves Time and Date of payment and stores PaymentType in the Payment table and Update PaymentStatus in Booking table. [ALT] System displays payment was successful notification. Step 17: System sends payment confirmation email to user.
ALTERNATE COURSES:	ALT-Step 2: If a user decides to make payment later and not in the booking process the system will then take the user to the Classes they have booked for but not paid yet by using the PaymentStatus in the Booking table and then direct the user back into the booking process with the BookingID ALT-Step 6: System validation failed. - Return to Step 3. ALT-Step 9: Incorrect card details entered. - Return to Step 8. ALT-Step 12: User rejects amount due. - Terminates use case. ALT-Step 15: Incorrect one-time-pin entered. - Return to Step 14. ALT- Step 16: Payment was unsuccessful.	

	Return to Step 3.
CONCLUSION:	System successfully makes payment for Class Booking
POST-CONDITION:	Users booking seat is now confirmed.
BUSINESS RULES	- Can only make a 50% deposit or full payment. - Payment must be made to secure seat.
IMPLEMENTATION CONSTRAINTS AND SPECIFICATIONS	System is offline
ASSUMPTIONS:	None
OPEN ISSUES:	None

Page Break Artech Systems

Author (s): Reinhardt Smit Date: 2021/05/03

Version: 1.0

USE CASE NAME:	Request Refund	USE CASE TYPE Business Requirements: ☐ System Analysis: ☒ System Design: ☐
USE CASE ID:	2.9	
PRIORITY:	High	
SOURCE:	Binary Brains – Artech System Requirements	
PRIMARY BUSINESS ACTOR	User	
PRIMARY SYSTEM ACTOR	None	
OTHER PARTICIPATING ACTORS:	Bank (ESA)	
OTHER INTERESTED STAKEHOLDERS:	None	
DESCRIPTION:	This use case describes how the User uses the system to generate a confirmation message of the class they booked for containing all the important details of that class. The system will gather all information for that specific booking using BookingID from the Booking table displaying all needed information for that class.	
PRE-CONDITION:	A Booking for a class must be made	
TRIGGER:	User requests to refund payment	
TYPICAL COURSE OF EVENTS:	Actor Action Step 1: User requests to refund payment Step 3: User selects refund request Step 5: User confirms refund request. [ALT]	System Response Step 2: System captures request and finds payment made via PaymentID in the Payment table and displays all payment details for that specific class through BookingID in the Booking table and checks if refund is available by determining the current date and date when class starts at Date and Time in Booking table. [ALT] Step 4: System Captures the action and prompts the user to confirm request refund. Step 6: System captures action and will then request the bank to reverse the transaction for the class that has been booked and generate a new RefundID for the Refund table. Step 7: The system will notify the art user that the refund is in process. System displays refund request is successful notification.

ALTERNATE COURSES:	ALT- Step 2: Refund not available. User requested refund too late. - Use case Terminated ALT-Step 5: User rejects refund request. - Terminates Use case.
CONCLUSION:	System successfully requested a refund
POST-CONDITION:	User patiently awaits the return of their money through the refund process.
BUSINESS RULES	Refunds can only be made until 2 days before the class starts
IMPLEMENTATION CONSTRAINTS AND SPECIFICATIONS	System is offline
ASSUMPTIONS:	None
OPEN ISSUES:	None

Artech Systems

Author (s): Trishla Pathak Date: 2021/05/01
Version: 2.0

USE CASE NAME:	Generate Art Class Feedback Form	USE CASE TYPE	
USE CASE ID:	2.10	Business Requirements: "	
PRIORITY:	High	System Analysis: "x	
SOURCE:	Artech system Usecase Requirement	System Design: "	
PRIMARY BUSINESS ACTOR	User (PBA)		
PRIMARY SYSTEM ACTOR	None		
OTHER PARTICIPATING ACTORS:	None		
OTHER INTERESTED STAKEHOLDERS:	Curator (ERA)		
DESCRIPTION:	The usecase describes an event where an Art Class feedback form is generated by the User to provide feedback on the art classes. The system provides the user with the form and the user fills in the details. The system then saves these details and the curator is able to view the feedback.		
PRE-CONDITION:	Art Class Period has ended.		
TRIGGER:	User wants to provide feedback on the art class.		
TYPICAL COURSE OF EVENTS:	Actor Action	System Response	
	1. User wants to provide feedback on the art class.	2. System displays the Feedback form page.	
	1. Users enter all the information they need to in the Feedback form and submit.	2. System successfully captures information into the Feedback Table as follows: [alt] - Feedback Page ID - Public User ID - Artist ID - Art Class ID - Feedback Comment - Rating Teacher - Rating Food - Rating Difficulty - Overall Rating	

ALTERNATE COURSES:	Alt 4: System unable to capture information due to system error. Return to Step 3.
CONCLUSION:	Art Class Feedback form is filled in.
POST-CONDITION:	None
BUSINESS RULES	One User can fill only one feedback form, once
IMPLEMENTATION CONSTRAINTS AND SPECIFICATIONS	None
ASSUMPTIONS:	None
OPEN ISSUES:	None

2.2.3 Report Subsystem

Author (s): Reinhardt Smit Artech Systems Date: 2021/05/03

Version: 1.0

USE CASE NAME:	Generate New User Report	USE CASE TYPE Business Requirements: ☐ • System Analysis: ☒ • System Design: ☐ •
USE CASE ID:	3.1	
PRIORITY:	High	
SOURCE:	Binary Brains – Artech System Requirements	
PRIMARY BUSINESS ACTOR	Curator	
PRIMARY SYSTEM ACTOR	None	
OTHER PARTICIPATING ACTORS:	• None	
OTHER INTERESTED STAKEHOLDERS:	• None	
DESCRIPTION:	This use case describes how the Curator uses the system to generate a New Users report that report how many users have registered unto the system in the selected time period. The Curator will be required to select a time period or can set a automated period to generate reports.	
PRE-CONDITION:	• The Curator must firstly be logged in	
TRIGGER:	Curator requests to generate New User Report	
TYPICAL COURSE	Actor Action	System Response
OF EVENTS:	Step 1: Curator requests to generate New User Report	Step 2: System captures request, gathers user details from the User table and prompts the Curator to select a period for report.
	Step 3: The Curator provides a period range for the New User Report.	Step 4: System Captures the action and gathers all information from User table counting all UserID that registered unto the system in the provided period. [ALT] System successfully generates report and displays it to the Curator with all its necessary details.
		Step 5: System determines the Curators details through the Administration table using the UserRoleID to get the contact details of an

	Administrator. The EmailAddress from the User table is read and is used to send report to the Curator.
	Step 6: System successfully sends the generated report to the Curator.
ALTERNATE COURSES:	ALT-Step 4: Selects a period that is out of range. Return to Step 3.
CONCLUSION:	System successfully generates report and displays it to the Curator
POST-CONDITION:	The system has generated the New User report and business intelligence can now be elicited from the report.
BUSINESS RULES	- Only the Curator can generate this report. - Only Curator can set period for generating report
IMPLEMENTATION CONSTRAINTS AND SPECIFICATIONS	System is offline
ASSUMPTIONS:	None
OPEN ISSUES:	None

Author (s): Reinhardt Smit Artech Systems Date: 2021/05/03

Version: 1.0

USE CASE NAME:	Generate Art Class Report	USE CASE TYPE Business Requirements: ☐ System Analysis: ☒ System Design: ☐
USE CASE ID:	3.2	
PRIORITY:	High	
SOURCE:	Binary Brains – Artech System Requirements	
PRIMARY BUSINESS ACTOR	Curator	
PRIMARY SYSTEM ACTOR	None	
OTHER PARTICIPATING ACTORS:	None	
OTHER INTERESTED STAKEHOLDERS:	None	
DESCRIPTION:	This use case describes how the Curator uses the system to generate a monthly art class report	
PRE-CONDITION:	The Curator must firstly be logged in	
TRIGGER:	Curator requests to generate Art Class report	
TYPICAL COURSE OF EVENTS:	Actor Action	System Response
	Step 1: Curator requests to generate Art Class report	Step 2: System prompts the Curator to select a period for report.
	Step 3: The Curator provides a period range for the Art Class Report.	Step 4: System Captures the action and gathers all information from Art_Class table counting all ArtClassType's and how many users per type is attending on the in system in the provided period by also counting PublicUserID in Booking table. [ALT] System successfully generates report and displays it to the Curator.

		Step 5: System determines the Curators details through the Administration table using the UserRoleID to get the contact details of an Administrator. The EmailAddress from the User table is read and is used to send report to the Curator. Step 6: System successfully sends the generated report to the Curator.
ALTERNATE COURSES:	ALT-Step 4: Selects a period that is out of range. Return to Step 3.	
CONCLUSION:	System successfully generates report and displays it to the Curator	
POST-CONDITION:	The system has generated the Art Class report and business intelligence can now be elicited from the report.	
BUSINESS RULES	- Only the Curator can generate this report. - Only Curator can set period for generating report	
IMPLEMENTATION CONSTRAINTS AND SPECIFICATIONS	System is offline	
ASSUMPTIONS:	None	
OPEN ISSUES:	None	

Author (s): Reinhardt Smit Artech Systems Date: 2021/05/03

Version: 1.0

USE CASE NAME:	Generate Class Teacher Report	USE CASE TYPE Business Requirements: ☐ • System Analysis: ☒ • System Design: ☐ •
USE CASE ID:	3.3	
PRIORITY:	High	
SOURCE:	Binary Brains – Artech System Requirements	
PRIMARY BUSINESS ACTOR	Curator	
PRIMARY SYSTEM ACTOR	None	
OTHER PARTICIPATING ACTORS:	None	
OTHER INTERESTED STAKEHOLDERS:	None	
DESCRIPTION:	This use case describes how the Curator uses the system to generate a monthly Class Teacher report	
PRE-CONDITION:	The Curator must firstly be logged in	
TRIGGER:	Curator requests to generate Class Teacher Report	
TYPICAL COURSE OF EVENTS:	Actor Action Step 1: Curator requests to generate Class Teacher Report Step 3: The Curator provides a period range for the Class Teacher Report.	System Response Step 2: System prompts the Curator to select a period for report. Step 2: System Captures the action and gathers all information from the Art_Class table to count class ClassTeacherID to generate a report to review the amount each teacher teaches a class in the provided period. [ALT] System successfully generates report and displays it to the Curator.

		Step 5: System determines the Curators details through the Administration table using the UserRoleID to get the contact details of an Administrator. The EmailAddress from the User table is read and is used to send report to the Curator. Step 6: System successfully sends the generated report to the Curator.
ALTERNATE COURSES:	ALT-Step 4: Selects a period that is out of range. Return to Step 3.	
CONCLUSION:	System successfully generates report and displays it to the Curator	
POST-CONDITION:	The system has generated the Class Teacher report and business intelligence can now be elicited from the report.	
BUSINESS RULES	- Only the Curator can generate this report. - Only Curator can set period for generating report	
IMPLEMENTATION CONSTRAINTS AND SPECIFICATIONS	System is offline	
ASSUMPTIONS:	None	
OPEN ISSUES:	None	

Author (s): Reinhardt Smit Artech Systems Date: 2021/05/03

Version: 1.0

USE CASE NAME:	Generate Exhibition Report	USE CASE TYPE Business Requirements: ☐ • System Analysis: ☒ • System Design: ☐ •
USE CASE ID:	3.4	
PRIORITY:	High	
SOURCE:	Binary Brains – Artech System Requirements	
PRIMARY BUSINESS ACTOR	Curator	
PRIMARY SYSTEM ACTOR	None	
OTHER PARTICIPATING ACTORS:	• None	
OTHER INTERESTED STAKEHOLDERS:	• None	
DESCRIPTION:	This use case describes how the Curator uses the system to generate an Exhibition Report.	
PRE-CONDITION:	• The Curator must firstly be logged in	
TRIGGER:	Curator requests to generate Exhibition Report	
TYPICAL COURSE	Actor Action	System Response
OF EVENTS:	Step 1: Curator requests to generate Exhibition Report	Step 2: System prompts the Curator to select a period for report.
	Step 3: The Curator provides a period range for the Exhibition Report Report.	Step 2: System Captures the action and gathers all Exhibition information from the Exhibition table counting all exhibitions that have taken place by using the ExhibitionID in the provided period. [ALT]

		System successfully generates report and displays it to the Curator.
		Step 5: System determines the Curators details through the Administration table using the UserRoleID to get the contact details of an Administrator. The EmailAddress from the User table is read and is used to send report to the Curator.
		Step 6: System successfully sends the generated report to the Curator.
ALTERNATE COURSES:	ALT-Step 4: Selects a period that is out of range. Return to Step 3.	
CONCLUSION:	System successfully generates report and displays it to the Curator	
POST-CONDITION:	The system has generated the Generate Exhibition report and business intelligence can now be elicited from the report.	
BUSINESS RULES	- Only the Curator can generate this report. - Only Curator can set period for generating report	
IMPLEMENTATION CONSTRAINTS AND SPECIFICATIONS	System is offline	
ASSUMPTIONS:	None	
OPEN ISSUES:	None	

Author (s): Reinhardt Smit Artech Systems Date: 2021/05/03 Version: 1.0

USE CASE NAME:	Generate Class Survey Report	USE CASE TYPE Business Requirements: ☐ • System Analysis: ✓• System Design: ☐ •
USE CASE ID:	3.5	
PRIORITY:	High	
SOURCE:	Binary Brains – Artech System Requirements	
PRIMARY BUSINESS ACTOR	Curator	
PRIMARY SYSTEM ACTOR	None	
OTHER PARTICIPATING ACTORS:	• None	
OTHER INTERESTED STAKEHOLDERS:	• None	
DESCRIPTION:	This use case describes how the Curator uses the system to generate a monthly Class Survey report	
PRE-CONDITION:	• The Curator must firstly be logged in	
TRIGGER:	Curator requests to generate Class Survey Report	
TYPICAL COURSE OF EVENTS:	Actor Action	System Response
	Step 1: Curator requests to generate Class Survey Report	Step 2: System prompts the Curator to select a period for report.
	Step 3: The Curator provides a period range for the Class Survey Report.	Step 2: System Captures the action and gathers all information from Feedback table capturing all RatingClass values and calculation an overall

	average rating per class in the given time period. [ALT]
	System successfully generates report and displays it to the Curator.
	Step 5: System determines the Curators details through the Administration table using the UserRoleId to get the contact details of an Administrator. The EmailAddress from the User table is read and is used to send report to the Curator.
	Step 6: System successfully sends the generated report to the Curator.
ALTERNATE COURSES:	ALT-Step 4: Selects a period that is out of range. Return to Step 3.
CONCLUSION:	System successfully generates report and displays it to the Curator
POST-CONDITION:	The system has generated the Class Survey report and business intelligence can now be elicited from the report.
BUSINESS RULES	- Only the Curator can generate this report. - Only Curator can set period for generating report
IMPLEMENTATION CONSTRAINTS AND SPECIFICATIONS	System is offline
ASSUMPTIONS:	None
OPEN ISSUES:	None

Author (s): Reinhardt Smit Artech Systems Date: 2021/05/03

Version: 1.0

USE CASE NAME:	Generate Payment Report	USE CASE TYPE Business Requirements: ☐ • System Analysis: ☒ • System Design: ☐ •
USE CASE ID:	3.6	
PRIORITY:	High	
SOURCE:	Binary Brains – Artech System Requirements	
PRIMARY BUSINESS ACTOR	Curator	
PRIMARY SYSTEM ACTOR	None	
OTHER PARTICIPATING ACTORS:	None	
OTHER INTERESTED STAKEHOLDERS:	None	
DESCRIPTION:	This use case describes how the Curator uses the system to generate a monthly Class Survey Report.	
PRE-CONDITION:	The Curator must firstly be logged in	
TRIGGER:	Curator requests to generate Payment Report	
TYPICAL COURSE	Actor Action	System Response
	Step 1: Curator requests to generate Payment Report	Step 2: System prompts the Curator to select a period for report.
OF EVENTS:	Step 3: The Curator provides a period range for the Payment Report.	Step 2: System Captures the action and gathers all Payment information from the Payment table

	keeping track of all payments and refunds that have been made in the given period. [ALT]
	System successfully generates report and displays it to the Curator.
	Step 5: System determines the Curators details through the Administration table using the UserRoleId to get the contact details of an Administrator. The EmailAddress from the User table is read and is used to send report to the Curator.
	Step 6: System successfully sends the generated report to the Curator.
ALTERNATE COURSES:	ALT-Step 4: Selects a period that is out of range. Return to Step 3.
CONCLUSION:	System successfully generates report and displays it to the Curator
POST-CONDITION:	The system has generated the Payment report and business intelligence can now be elicited from the report.
BUSINESS RULES	- Only the Curator can generate this report. - Only Curator can set period for generating report
IMPLEMENTATION CONSTRAINTS AND SPECIFICATIONS	System is offline
ASSUMPTIONS:	None
OPEN ISSUES:	None

2.2.4 User Subsystem

Artech Systems

Author (s): Gamuchirai Dambanjera

Date: 2021/05/03

Version: 1.0

USE CASE NAME:	Register User	USE CASE TYPE	
USE CASE ID:	4.1	Business Requirements:	☐
PRIORITY:	High	System Analysis:	☒
SOURCE:	Artech System Requirements List	System Design:	☐
PRIMARY BUSINESS ACTOR	User		
PRIMARY SYSTEM ACTOR	None		
OTHER PARTICIPATING ACTORS:	None		
OTHER INTERESTED STAKEHOLDERS:	None		

DESCRIPTION:	This use case describes the event where a user is registered on the system. The use case begins when the user provides the system with all of the required information and enters the information into the system. The use case ends when the user is notified by the system that they have been successfully registered.	
PRE-CONDITION:	The user does not exist on the system.	
TRIGGER:	The user enters his/her personal information	
TYPICAL COURSE OF EVENTS:	Actor Action	System Response
	1. The user requests to be registered on the system	2. The system captures the request and displays the registration page and prompts the user to enter their details into required input fields.
	3. The user enters his/her personal details	4. The system captures user details and validates that the values in the input fields are in the correct format and that no fields are empty. [ALT]
	5. The user submits their registration details.	6. The system captures user details and verifies that the user does not already exist on the system and that the username is unique. [ALT] Then the system stores the values in the User table as follows: • Name • Surname • Password • DOB • Email Address • Phone Number • Username • Password And from User Role table • User Role Name • User Role Description The system adds the new User information into the User table.
		7. The system notifies the user that he/she has been successfully registered. [ALT]
ALTERNATE COURSES:	ALT step 4: The validation is unsuccessful and an incorrect format was inserted into the system or a field was left empty. • Display notification that a format is incorrect or a field was left empty • Return to step 3 ALT Step 6: The system does not add the new user on the system as they already exist. The system notifies the user of the error. • Use case is terminated.	

	ALT Step 7: The system fails to add the new user on the system. A failure notification appears to the user. • Terminate Use Case.
CONCLUSION:	A new user is successfully registered.
POST-CONDITION:	The User can now log into the system.
BUSINESS RULES	The user can only have one account
IMPLEMENTATION CONSTRAINTS AND SPECIFICATIONS	None
ASSUMPTIONS:	None
OPEN ISSUES:	None

Artech Systems

Author (s): Gamuchirai Dambanjera

Date: 2021/05/03

Version: 1.0

USE CASE NAME:	Login	USE CASE TYPE	
USE CASE ID:	4.2	Business Requirements:	☐
PRIORITY:	High	System Analysis:	✓
SOURCE:	Artech System Requirements List	System Design:	☐
PRIMARY BUSINESS ACTOR	User		
PRIMARY SYSTEM ACTOR	None		
OTHER PARTICIPATING ACTORS:	None		
OTHER INTERESTED STAKEHOLDERS:	None		
DESCRIPTION:	The use case describes the event where the person (user) wants to log in to the system. The use case begins where the person (user) enters their login details. If the login details are correct, the use case will conclude by logging the person (user) into the system.		
PRE-CONDITION:	The person must have been registered and approved as a member.		
TRIGGER:	The person wants to log in to the system.		
TYPICAL COURSE OF EVENTS:	Actor Action	System Response	
	1. The User requests to log in to the system	2. System displays the Login page and prompts the User to enter their login details	
	3. The User enters login details.	4. System validates the input fields that they are in the correct format [alt]	
	5. The user selects Login	6. The system checks that the Username and Password are correct by looking it up from the User table based on the UserID . [alt] User successfully logs in	
ALTERNATE COURSES:	Alt Step 4: Validation error, due to the User entering incorrect login details. The system notifies the user that the details that they have entered are incorrect. Return to step 3.		
CONCLUSION:	System logs the User into the system.		
POST-CONDITION:	The User may access the areas of the system available to them.		
BUSINESS RULES	The person must be registered on the system in order to Log in		
IMPLEMENTATION CONTRAINTS AND SPECIFICATIONS	None		
ASSUMPTIONS:	None		
OPEN ISSUES:	None		

Artech Systems

Author (s): Gamuchirai Dambanjera

Date: 2021/05/03

Version: 1.0

USE CASE NAME:	Logout	USE CASE TYPE	
USE CASE ID:	4.3	Business Requirements:	☐
PRIORITY:	High	System Analysis:	☒
SOURCE:	Artech System Requirements List	System Design:	☐
PRIMARY BUSINESS ACTOR	User		
PRIMARY SYSTEM ACTOR	None		
OTHER PARTICIPATING ACTORS:	None		
OTHER INTERESTED STAKEHOLDERS:	None		
DESCRIPTION:	The use case describes the event where the user wants to log out of the system. The use case begins where the person clicks the logout option. The system will confirm with the user if they are sure that they want to log out. The use case concludes when the person confirms the logout.		
PRE-CONDITION:	- User must be logged in to the system. - User must exist on the system.		
TRIGGER:	The user wants to logout of the system.		
TYPICAL COURSE OF EVENTS:	Actor Action	System Response	
	1. The user selects the logout option.	2. System requests the user confirm that they want to logout of the system.	
	3. The user selects the option to confirm that they want to logout of the system. [alt]	4. System logs out the user from the system. [alt]	
		5. The system displays the logout page	
ALTERNATE COURSES:	Alt step 3: The user selects the option to not logout from the system. Terminate Use Case.		
	Alt Step 4: System fails to logout the user. Notify user of the error. Return to Step 2.		
CONCLUSION:	The use case ends when the user is successfully logged out and the login page is displayed.		
POST-CONDITION:	The user is successfully logged out of the system.		
BUSINESS RULES	None		



GROUP 23 DELIVERABLE 02



IMPLEMENTATION CONSTRAINTS AND SPECIFICATIONS	None
ASSUMPTIONS:	None
OPEN ISSUES:	None

Artech Systems

Author (s): Gamuchirai Dambanjera

Date: 2021/05/03

Version: 1.0

USE CASE NAME:	Reset Password	USE CASE TYPE	
USE CASE ID:	4.4	Business Requirements:	☐
PRIORITY:	Low	System Analysis:	☒
SOURCE:	Artech System Requirements List	System Design:	☐
PRIMARY BUSINESS ACTOR	User		
PRIMARY SYSTEM ACTOR	None		
OTHER PARTICIPATING ACTORS:	None		
OTHER INTERESTED STAKEHOLDERS:	None		
DESCRIPTION:	This use case describes the event where the user wants to reset their password because they have forgotten it. The use case begins where the user clicks the reset password option in the case where their password is forgotten. The system prompts the user to enter a new password and to confirm it. If those two passwords match, the system will confirm that the user is sure that they would like to reset it. System will send person a One Time Password (OTP) and if the person enters the correct OTP, the use case concludes with the user confirming the password reset.		
PRE-CONDITION:	- User must be logged in to the system. - User must exist on the system.		
TRIGGER:	The user wants to logout of the system.		
TYPICAL COURSE OF EVENTS:	Actor Action	System Response	
	1. The user has forgotten their password and selects the forgot password option	2. The system displays the forgot password page and prompts the user to enter their Username from the User table with a corresponding Email Address	
	3. User clicks the reset password option in the case where their password is forgotten.	4. System captures and validates username and birthday by reading the correct Username and Email Address from the User table based on the UserID . [alt]	
		5. System prompts the user to enter a new password and confirm it.	
	6. User enters a new password and a confirmation thereof.	7. System captures and validates the passwords. [alt]	
		8. System prompts the User to confirm that they	
	9. User confirms that they want to reset their password. [alt]		

ALTERNATE COURSES:	Alt Step 4: Validation error, due to the user entering an invalid username or email. System alerts person that the username and email that they entered is incorrect. Return to step 3.
	Alt Step 7: Validation error, due to the user entering a non-matching password. System alerts the user that the passwords that they've entered did not match. Return to step 6.
CONCLUSION:	The use case ends when the user is notified that their password has been reset successfully
POST-CONDITION:	The user logs in with their new password.
BUSINESS RULES	None
IMPLEMENTATION CONTRAINTS AND SPECIFICATIONS	None
ASSUMPTIONS:	The User resetting the password is the owner of that username.
OPEN ISSUES:	None

Artech Systems

Author (s): Gamuchirai Dambanjera

Date: 2021/05/03

Version: 1.0

USE CASE NAME:	Update User	USE CASE TYPE	
USE CASE ID:	4.5	Business Requirements:	☐
PRIORITY:	High	System Analysis:	☒
SOURCE:	Artech System Requirements List	System Design:	☐
PRIMARY BUSINESS ACTOR	User		
PRIMARY SYSTEM ACTOR	None		
OTHER PARTICIPATING ACTORS:	None		
OTHER INTERESTED STAKEHOLDERS:	None		
DESCRIPTION:	The use case describes the event where the User wants to update his/her details. The use case begins when the system wants to update user details and the system prompts the user to look for User fields they would like to update. The User enters or selects their search query and then select the user fields they would like to update. The system will display that user’s details and the user will be able to enter the updated version of those details. If these details are valid the user must confirm the update and when they do, the use case concludes where the system will update the details.		

PRE-CONDITION:	- User must be logged in to the system. - User must exist on the system.	
TRIGGER:	The user wants to update his/her details.	
TYPICAL COURSE OF EVENTS:	Actor Action	System Response
	1. The user wants to update his/her details	2. System prompts the user to enter or select the user fields they would like to update.
	3. User enters their details	4. System captures and validates the details entered by the user. [alt]
		5. System prompts the user to confirm the update.
	6. User confirms the update [alt]	7. System saves updated, User table as follows: Name Surname Password DOB Email Address Phone Number Username Password And from User Role table User Role Name User Role Description
ALTERNATE COURSES:	Alt Step 4: Validation error, due to the user entering invalid user details. The system notifies the user of their invalid user details. Return to step 3.	
	Alt Step 6: Person cancels update. Return to step 5.	
CONCLUSION:	System updates user details.	
POST-CONDITION:	The updated version of the user's details is seen when viewed.	
BUSINESS RULES	None	
IMPLEMENTATION CONSTRAINTS AND SPECIFICATIONS	None	
ASSUMPTIONS:	The updated version of the details are the correct details.	
OPEN ISSUES:	None	



GROUP 23 DELIVERABLE 02



Artech Systems

Author (s): Gamuchirai Dambanjera

Date: 2021/05/03

Version: 1.0

USE CASE NAME:	View Software Guide	USE CASE TYPE Business Requirements: ☐ System Analysis: ☒ System Design: ☐	
USE CASE ID:	4.6		
PRIORITY:	High		
SOURCE:	Artech System Requirements List		
PRIMARY BUSINESS ACTOR	User		
PRIMARY SYSTEM ACTOR	None		
OTHER PARTICIPATING ACTORS:	None		
OTHER INTERESTED STAKEHOLDERS:	None		
DESCRIPTION:	The use case describes the event where a user wishes to view the software guide on the system.		
PRE-CONDITION:	• User must be logged in to the system. • User must exist on the system.		
TRIGGER:	The user wants to view software guide.		
TYPICAL COURSE OF EVENTS:	Actor Action	System Response	
	1. The user wants to view the software guide.	2. System displays a link to Viewing the software guide.	
	3. The user clicks on the link to view the Software guide in detail.	4. System displays the full software guide in PDF Format.	

	5. User views the software guide and reads through it.	
ALTERNATE COURSES:	NONE	
CONCLUSION:	The use case ends when the user has successfully viewed the Software guide	
POST-CONDITION:	The user is successfully logged out of the system.	
BUSINESS RULES	None	
IMPLEMENTATION CONTRAINTS AND SPECIFICATIONS	None	
ASSUMPTIONS:	None	
OPEN ISSUES:	None	

Artech Systems

Author (s): Gamuchirai Dambanjera

Date: 2021/05/03

Version: 1.0

USE CASE NAME:	View Notifications	USE CASE TYPE	
USE CASE ID:	4.7	Business Requirements:	☐
PRIORITY:	High	System Analysis:	☒
SOURCE:	Artech System Requirements List	System Design:	☐
PRIMARY BUSINESS ACTOR	User		
PRIMARY SYSTEM ACTOR	None		
OTHER PARTICIPATING ACTORS:	None		
OTHER INTERESTED STAKEHOLDERS:	None		
DESCRIPTION:	The use case describes the event where a user wishes to view all notifications sent to them.		
PRE-CONDITION:	• User must be logged in to the system. • User must exist on the system.		
TRIGGER:	The user wants to view notifications sent them		
TYPICAL COURSE			
OF EVENTS:	Actor Action	System Response	
	1. The user wants to view all the notifications in their inbox.	2. System searches for notifications sent to the user that is currently logged in and displays them, retrieving all the notifications	

		that have been received up to the date of viewing them. [alt]
	3. The user clicks on a specific notification to view it in detail.	4. The system displays the notification as "Viewed"
ALTERNATE COURSES:	Alt step 2: Popup message displays that no new notifications are in the user's inbox.	
CONCLUSION:	The system displays the notifications details for the notification clicked on by the user and changes the Notification status to "Viewed".	
POST-CONDITION:	The status of the notification that was viewed is updated to "Viewed"	
BUSINESS RULES	None	
IMPLEMENTATION CONSTRAINTS AND SPECIFICATIONS	None	
ASSUMPTIONS:	None	
OPEN ISSUES:	None	

Artech Systems

Author (s): Gamuchirai Dambanjera

Date: 2021/05/03

Version: 1.0

USE CASE NAME:	Generate Frame Information		USE CASE TYPE	
USE CASE ID:	4.8		Business Requirements:	☐
PRIORITY:	High		System Analysis:	☒
SOURCE:	Artech System Requirements List		System Design:	☐
PRIMARY BUSINESS ACTOR	User			
PRIMARY SYSTEM ACTOR	None			
OTHER PARTICIPATING ACTORS:	None			
OTHER INTERESTED STAKEHOLDERS:	None			
DESCRIPTION:	The use case describes the event where a user wishes to generate frame information			
PRE-CONDITION:	- User must be logged in to the system. - User must exist on the system.			
TRIGGER:	The user wants to view notifications sent them			
TYPICAL COURSE OF EVENTS:	Actor Action	System Response		
	1. The user wants to generate frame information	2. System searches for all the frames stored in the system.		
	3. The user clicks on a specific frame to generate information for it	4. The system generates and displays frame information.		
ALTERNATE COURSES:	None			
CONCLUSION:	The system generates frame information and displays it to the user			
POST-CONDITION:	User must be logged into the system			
BUSINESS RULES	None			
IMPLEMENTATION CONSTRAINTS AND SPECIFICATIONS	None			
ASSUMPTIONS:	None			
OPEN ISSUES:	None			

2.3.5 Artist Subsystem

Artech Systems

Author (s): Gamuchirai Dambanjera

Date: 2021/05/03

Version: 1.0

USE CASE NAME:	Apply for Exhibition	USE CASE TYPE	
USE CASE ID:	5.1	Business Requirements:	☐
PRIORITY:	High	System Analysis:	☒
SOURCE:	Artech System Requirements List	System Design:	☐
PRIMARY BUSINESS ACTOR	Artist		
PRIMARY SYSTEM ACTOR	Curator		
OTHER PARTICIPATING ACTORS:	None		
OTHER INTERESTED STAKEHOLDERS:	None		
DESCRIPTION:	This use case describes an event where an artist applies for an exhibition. The artist could already be registered on the system as a user. The artist will apply for a particular exhibition and the curator will either accept or decline their application. The use case comes to an end when the user has been accepted or declined.		
PRE-CONDITION:	The curator must approve an artist's application to exhibit his/her artist		
TRIGGER:	The user wants to logout of the system.		
TYPICAL COURSE OF EVENTS:	Actor Action	System Response	
	1. An artist wants to apply for an exhibition.	2. The system prompts the artist to fill in the form with their exhibition details.	
	3. The artist fills in their exhibition details	4. System captures and validates the details. [alt]	
		5. System prompts artist to fill in tags	
	6. Artist fills in tag.	7. System captures and validates tag information. [alt]	
		8. Curator goes through an artist's exhibition application and informs them that their application has been successful [alt]	
ALTERNATE COURSES:	Alt Step 4: Validation error, due to the artist filling in the artist form wrongly. Return to step 3.		

	Alt Step 7: Validation error, due to the artist wrong tag information. System alerts the artist that their tag information has been wrongly entered. Return to step 6. Alt step 8: Curator goes through the artist's exhibition form and declines their application. The use case comes to an end.
CONCLUSION:	The use case ends when the artist has been notified that their application is either successful or not
POST-CONDITION:	The artist must be logged on into the system
BUSINESS RULES	None
IMPLEMENTATION CONSTRAINTS AND SPECIFICATIONS	None
ASSUMPTIONS:	The artist can be registered on the system as a user
OPEN ISSUES:	None

Artech Systems

Author (s): Gamuchirai Dambanjera

Date: 2021/05/03

Version: 1.0

USE CASE NAME:	Apply to teach class	USE CASE TYPE	
USE CASE ID:	5.2	Business Requirements:	☐
PRIORITY:	High	System Analysis:	☒
SOURCE:	Artech System Requirements List	System Design:	☐
PRIMARY BUSINESS ACTOR	Artist		
PRIMARY SYSTEM ACTOR	Curator		
OTHER PARTICIPATING ACTORS:	None		
OTHER INTERESTED STAKEHOLDERS:	None		
DESCRIPTION:	This use case describes the event where an artist wants to apply to teach a class. The artist applies to teacher an art class of which they are familiar with e.g. an oil painter will apply to teach an oil painting class because that is what they are familiar with. The use case ends when the artist has been accepted or declined to teacher a particular art class		

PRE-CONDITION:	- Art teacher must already exist on the system as an Artist	
TRIGGER:	Art teacher wants to teach a class.	
TYPICAL COURSE OF EVENTS:	Actor Action	System Response
	1. Artist wants to apply to teach a class	2. The system request the artist to fill in the art class form.
	3. Artist fills in the art class form	4. System captures and validates that the art class form has been correctly filled. [alt]
		5. The curator goes the artist application to teach class and informs the artist that their application has been accepted. [alt]
ALTERNATE COURSES:	Alt Step 4: Validation error, due to the artist entering the wrong art class info. Return to step 3.	
	Alt Step 5: Artist's application to teach class has been declined. Use case comes to an end.	
CONCLUSION:	The use case ends when the artist has been accepted or declined to become an art teacher	
POST-CONDITION:	Artist must be registered on the system	
BUSINESS RULES	None	
IMPLEMENTATION CONTRAINTS AND SPECIFICATIONS	None	
ASSUMPTIONS:	None	
OPEN ISSUES:	None	

Artech Systems

Author (s): Gamuchirai Dambanjera

Date: 2021/05/03

Version: 1.0

USE CASE NAME:	Add Picture	USE CASE TYPE	
USE CASE ID:	5.3	Business Requirements:	☐
PRIORITY:	Medium	System Analysis:	☒
SOURCE:	Artech System Requirements List	System Design:	☐
PRIMARY BUSINESS ACTOR	Artist		

PRIMARY SYSTEM ACTOR	None	
OTHER PARTICIPATING ACTORS:	Curator (ERA)	
OTHER INTERESTED STAKEHOLDERS:	None	
DESCRIPTION:	This use case describes the event where the artist want to add a picture to their artist profile.	
PRE-CONDITION:	- Artist must be logged in to the system. - Artist must exist on the system.	
TRIGGER:	Artist wants to add picture	
TYPICAL COURSE OF EVENTS:	Actor Action	System Response
	1. Artist wants to add a picture of the artwork on their profile	2. The system displays a Browse option from which the artist can select a folder he/she wants to upload the picture from
	3. Artist selects the picture and clicks the upload button	4. System captures and validates whether the picture is the right format [alt]
		5. System sends a confirmation dialog asking the artist to confirm if they want to add this particular picture.
	6. Artist confirms that they want to upload that picture	7. System upload and saves that picture into the system and an alert message will pop up notifying the artist that the picture has been successfully added and saved.
ALTERNATE COURSES:	Alt Step 4: Validation error, due to the artist entering the wrong picture format. Return to step 3.	
CONCLUSION:	The use case ends when the artist is notified that the picture has been successfully added.	
POST-CONDITION:	The artist can see their added picture on their profile	
BUSINESS RULES	None	
IMPLEMENTATION CONTRAINTS AND SPECIFICATIONS	None	
ASSUMPTIONS:	Non	
OPEN ISSUES:	None	

Artech Systems

Author (s): Gamuchirai Dambanjera

Date: 2021/05/03

Version: 1.0

USE CASE NAME:	Update Picture	USE CASE TYPE	
USE CASE ID:	5.4	Business Requirements:	☐
PRIORITY:	Low	System Analysis:	☒
SOURCE:	Artech System Requirements List	System Design:	☐
PRIMARY BUSINESS ACTOR	Artist		
PRIMARY SYSTEM ACTOR	None		
OTHER PARTICIPATING ACTORS:	Curator (ERA)		
OTHER INTERESTED STAKEHOLDERS:	None		
DESCRIPTION:	The use case describes the event where the artist wants to update their picture on the system. The use case begins where the artist requests to update a picture. If a new picture in the right format is correctly uploaded, the system will conclude by updating the picture on the system.		
PRE-CONDITION:	- Artist must be logged onto the system - A picture must exist so that it can be updated		
TRIGGER:	The user wants to logout of the system.		
TYPICAL COURSE OF EVENTS:	Actor Action	System Response	
	1. Artist requests to update their picture on the system	2. The system displays the update picture screen	
	3. Artist selects the picture and clicks the upload button	4. System captures and validates whether the picture is the right format [alt]	
		5. System sends a confirmation dialog asking the artist to confirm if they want to update this particular picture	
	6. Artist confirms that they want to upload that picture.	7. System upload and saves that picture into the system and an alert message will pop up notifying the artist that the picture has been successfully Updated and saved into the system.	

ALTERNATE COURSES:	Alt Step 4: Validation error, due to the artist entering the wrong picture format. Return to step 3.
CONCLUSION:	The use case ends when the artist is notified that their picture has been successfully updated
POST-CONDITION:	The artist must be able to see their updated picture
BUSINESS RULES	None
IMPLEMENTATION CONTRAINTS AND SPECIFICATIONS	None
ASSUMPTIONS:	None
OPEN ISSUES:	None

Artech Systems

Author (s): Gamuchirai Dambanjera

Date: 2021/05/03

Version: 1.0

USE CASE NAME:	Delete Picture	USE CASE TYPE Business Requirements: ☐ System Analysis: ☒ System Design: ☐	
USE CASE ID:	5.5		
PRIORITY:	Low		
SOURCE:	Artech System Requirements List		
PRIMARY BUSINESS ACTOR	Artist		
PRIMARY SYSTEM ACTOR	None		
OTHER PARTICIPATING ACTORS:	Curator (ERA)		
OTHER INTERESTED STAKEHOLDERS:	None		
DESCRIPTION:	The use case describes the event where the artist wants to delete their picture from the system. The use case is initiated when an artist decides that they want to delete a picture from the system. The use case ends when the system displays a message saying that the picture has been successfully deleted from the system.		
PRE-CONDITION:	- Artist must be logged onto the system - A picture must exist so that it can be deleted		
TRIGGER:	The user wants to logout of the system.		
TYPICAL COURSE	Actor Action	System Response	
OF EVENTS:	1. Artist requests to delete their picture from the system	2. The system displays the delete picture screen	

	3. Artist navigates the system and select the picture they would like to delete	
	4. Artist clicks the "Delete" button to unload the picture from the system.	5. System asks the artist to confirm if they are sure that they want to delete this particular picture
	6. Artist confirms that they want to delete that particular picture.	7. System deletes and saves the changes into the system and an alert message will pop up notifying the artist that the picture has been successfully Deleted from the system.
ALTERNATE COURSES:	None	
CONCLUSION:	The use case ends when the artist is notified that their picture has been successfully Deleted	
POST-CONDITION:	The artist must be able to see their picture has been Deleted from the system	
BUSINESS RULES	None	
IMPLEMENTATION CONTRAINTS AND SPECIFICATIONS	None	
ASSUMPTIONS:	None	
OPEN ISSUES:	None	

Artech Systems

Author (s): Gamuchirai Dambanjera

Date: 2021/05/03

Version: 1.0

USE CASE NAME:	View Invitation	USE CASE TYPE	
USE CASE ID:	5.6	Business Requirements:	☐
PRIORITY:	Medium	System Analysis:	☒
SOURCE:	Artech System Requirements List	System Design:	☐
PRIMARY BUSINESS ACTOR	Artist		
PRIMARY SYSTEM ACTOR	None		
OTHER PARTICIPATING ACTORS:	Curator (ERA)		
OTHER INTERESTED STAKEHOLDERS:	None		
DESCRIPTION:	The use case describes the event where an artist wishes to view all invitations sent specifically to them.		

PRE-CONDITION:	Artist must be logged onto the system	
TRIGGER:	The artist wants to view all invitations in their inbox.	
TYPICAL COURSE OF EVENTS:	Actor Action	System Response
	1. The artist wants to view all invitations in their inbox	2. System searches for invitations sent to the artist that is currently logged in by retrieving all invitations in the Invite table where the InvitationID in the Artist/Invite entity is the same as the InvitationID of the invitation in the Invite entity and displays them. [alt]
	3. The artist selects to Add the event to their calendar.	4. The system uses the google calendar API to create a Google Calendar event in the Google account of the logged in artist.
	5. The artist selects to dismiss the Invitation.	6. The InvitationStatus of the invitation in the Artist_Invitation entity is set to "Viewed" using SQL update queries
ALTERNATE COURSES:	ALT Step 2: Popup message displays that no new Invitations are in the artist's inbox.	
CONCLUSION:	The use case ends when the invitation status changes to "Viewed"	
POST-CONDITION:	The status of the invitation that was viewed is updated to "Viewed"	
BUSINESS RULES	None	
IMPLEMENTATION CONTRAINTS AND SPECIFICATIONS	None	
ASSUMPTIONS:	None	
OPEN ISSUES:	None	

Artech Systems

Author (s): Gamuchirai Dambanjera

Date: 2021/05/03

Version: 1.0

USE CASE NAME:	View existing Exhibitions		USE CASE TYPE
USE CASE ID:	5.7	Business Requirements:	☐
PRIORITY:	High	System Analysis:	☒
SOURCE:	Artech System Requirements List	System Design:	☐
PRIMARY BUSINESS ACTOR	Artist and User		
PRIMARY SYSTEM ACTOR	None		
OTHER PARTICIPATING ACTORS:	None		
OTHER INTERESTED STAKEHOLDERS:	None		
DESCRIPTION:	This use case describes the process where an artist or user requests to view information regarding exhibitions. The system will retrieve exhibition info from the Exhibition table and displays it for the artist or user to view.		
PRE-CONDITION:	- Artist or user must be logged into the system. - Artist or user must already exist on the system.		
TRIGGER:	The artist or user want to view exhibition information		
TYPICAL COURSE OF EVENTS:	Actor Action	System Response	
	1. Artist or user requests to view exhibition information	2. The system searches the Exhibition table and retrieves the exhibition information	
	3. Artist or user clicks the "View" Exhibition Info button	4. The system displays the exhibition details with the following information from the Exhibition table: - ExhibitionID (PK) - ExhibitionName - ExhibitionDescription - ExhibitionDate - OrganisationID (FK) - ExhibitionTime - ScheduleID (FK) - VenueID (FK) - ExhibAnnounce_ID (FK)	
ALTERNATE COURSES:	None		
CONCLUSION:	This use case concludes when the system displays Exhibition Information		

POST-CONDITION:	Exhibition Information was retrieved from the from Exhibition table.
BUSINESS RULES	Only an artist or user may perform this function.
IMPLEMENTATION CONSTRAINTS AND SPECIFICATIONS	None
ASSUMPTIONS:	None
OPEN ISSUES:	None

Artech Systems

Author (s): Gamuchirai Dambanjera

Date: 2021/05/03

Version: 1.0

USE CASE NAME:	Accept Invitation for Exhibition	USE CASE TYPE
USE CASE ID:	5.8	Business Requirements: ☐
PRIORITY:	High	System Analysis: ☒
SOURCE:	Artech System Requirements List	System Design: ☐
PRIMARY BUSINESS ACTOR	Artist	
PRIMARY SYSTEM ACTOR	None	
OTHER PARTICIPATING ACTORS:	Curator (ERA)	
OTHER INTERESTED STAKEHOLDERS:	None	
DESCRIPTION:	This use case describes the event when an artist uses the system to accept an invitation for a specific exhibition, they received from the Curator by opening the invitation and accepts or declines it.	
PRE-CONDITION:	Artist must be logged into the system.	
TRIGGER:	Artist selects invitation received	
TYPICAL COURSE	Actor Action	System Response
OF EVENTS:	Step 1: Artist request to 'accept' the invitation received from the Curator.	Step 2: System displays all the invitation information by gathering all the necessary invite information for that specific Artist through the Invite table displaying all the invite information using - InvitationDetails - InvitationDate - Field

		InvitationStatus that will be displayed as 'Pending' and InvitationPerArtist table using the InvitationID ArtistID. System then displays all information contained in the invitation to the artist.
	Step 3: Artist 'accepts' the invitation. [ALT]	Step 4: System captures the Artists action and updates the InvitationStatus to 'Accepted' in the Invite table and successfully adds the artist to that exhibition list in the Exhibition_Artists table with ArtistID .
ALTERNATE COURSES:	ALT-Step 4: Artist selects 'decline'. System captures decline and updates Invitation status to 'Declined' in Invite table.	
CONCLUSION:	Artist accepts invitation and is successfully added to the exhibition	
POST-CONDITION:	Artist is added to the Exhibition table and stored in the database.	
BUSINESS RULES	Only Artists can receive Exhibition Invites.	
IMPLEMENTATION CONTRAINTS AND SPECIFICATIONS	None	
ASSUMPTIONS:	None	
OPEN ISSUES:	None	

Conclusion

In conclusion the above section details the use case diagrams and gives further explanations of them using the use case narratives for this proposed system.



3.Process Models of the New Proposed System

Introduction

This section consists of all the process models of the proposed system from the Decomposition diagram to all the Data Flow Diagrams and the Data Dictionaries. They will all visually illustrate the flow of all the processes between the system and its actors as well as the flows into the data stores.

3.1 Context Diagram

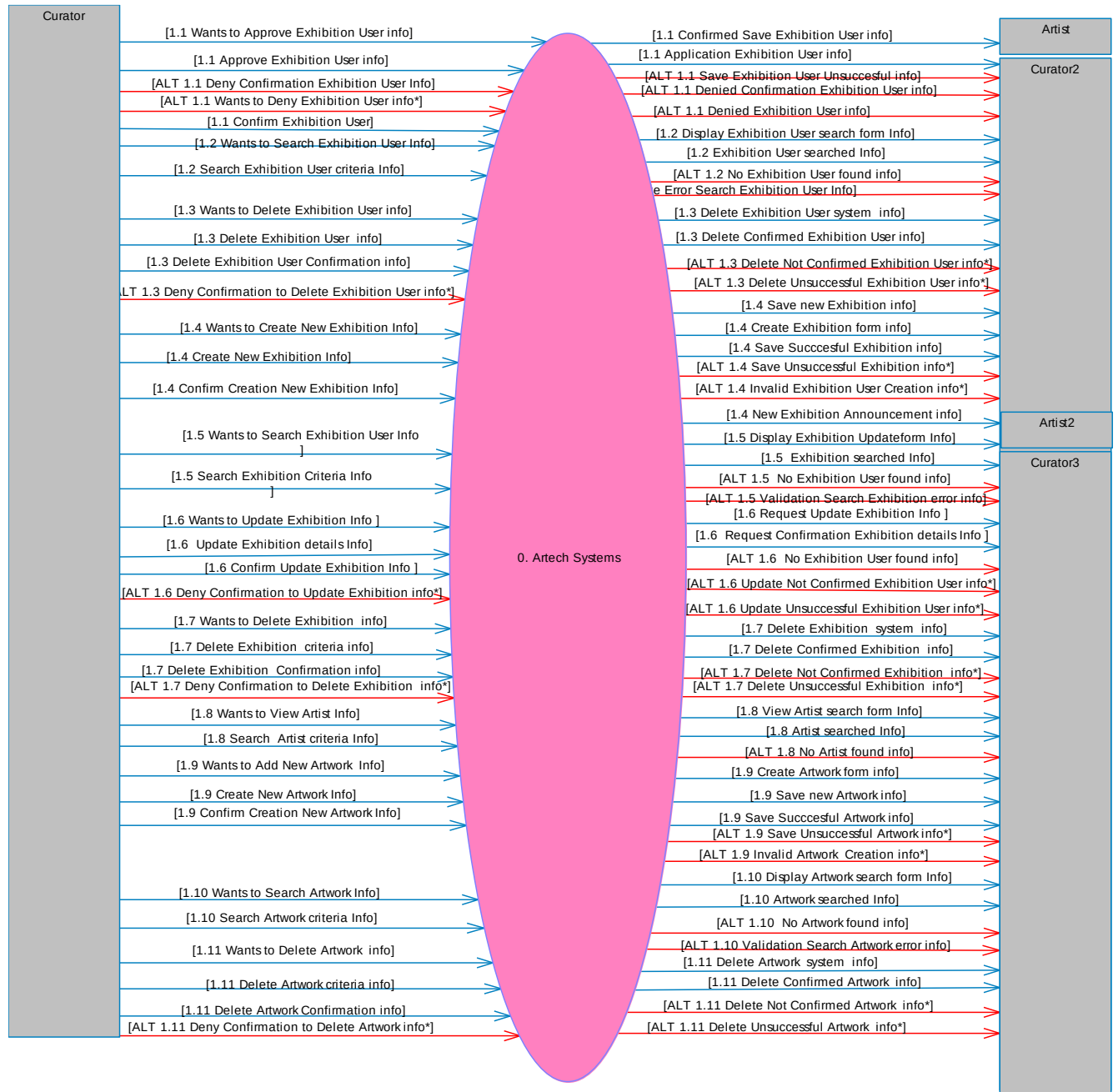


Figure 10: Context Diagram part 1 – Subsystem 1

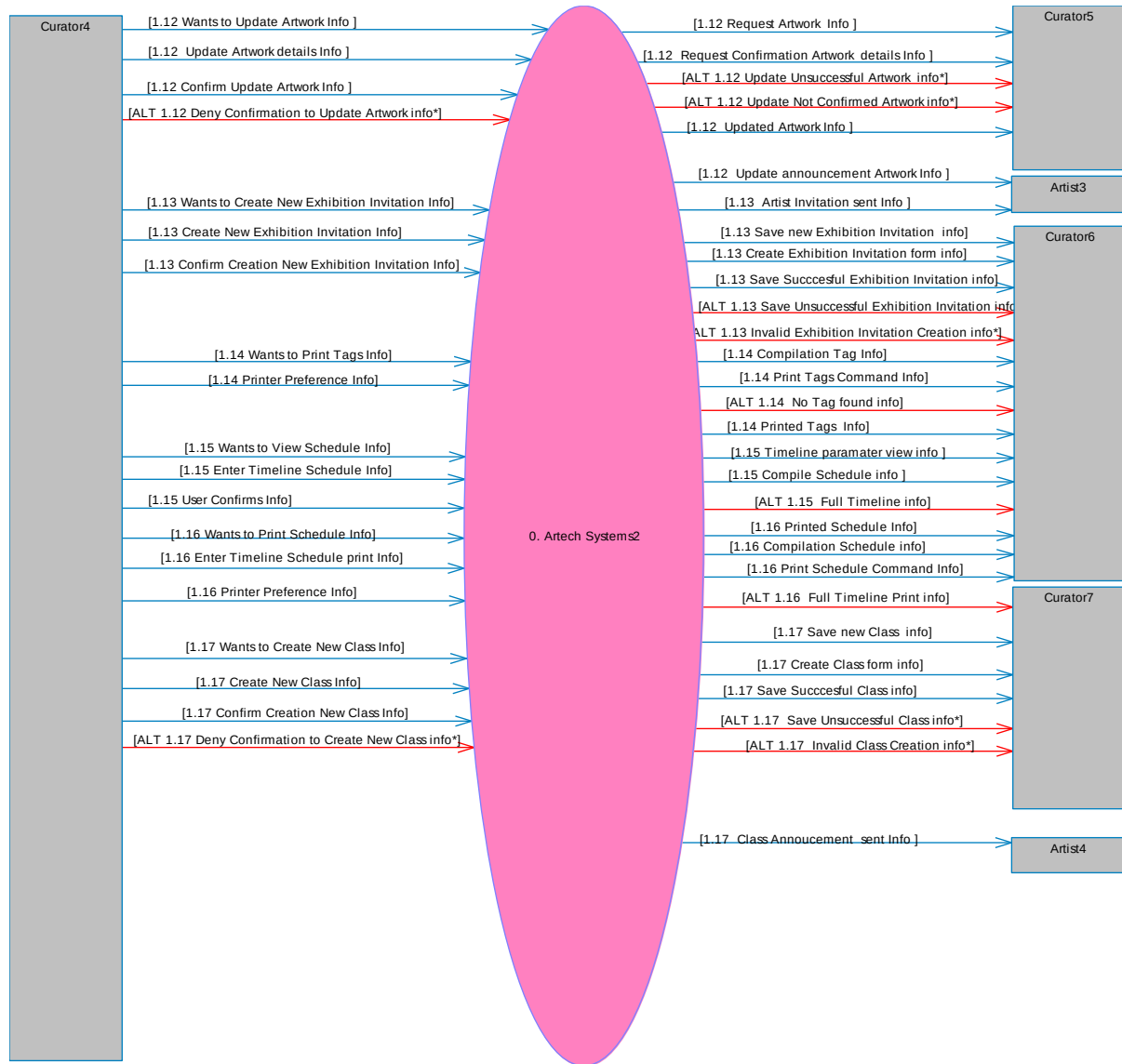


Figure 11: Context Diagram part 2 – Subsystem 1

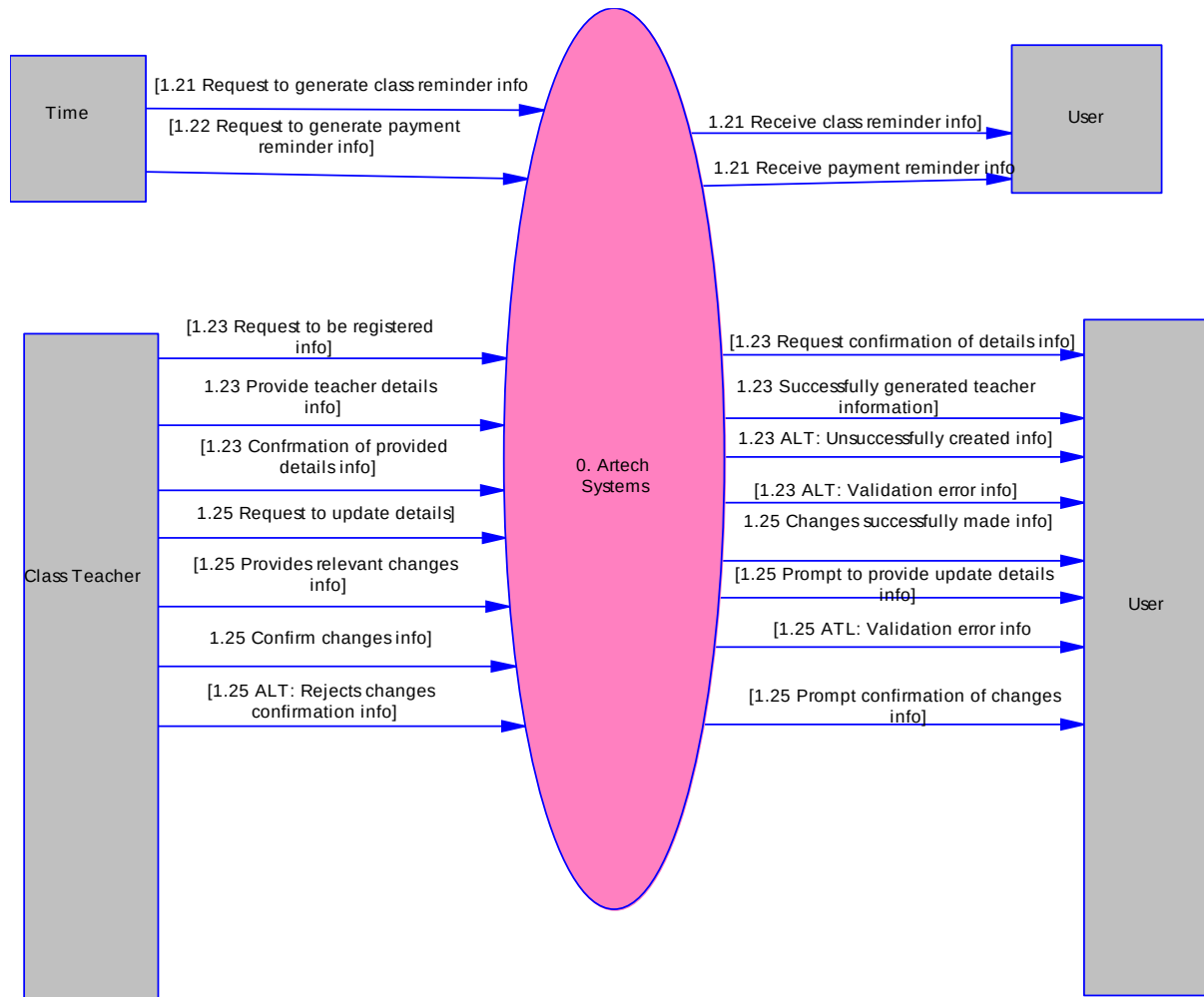


Figure 12: Context Diagram part 3 – Subsystem 1

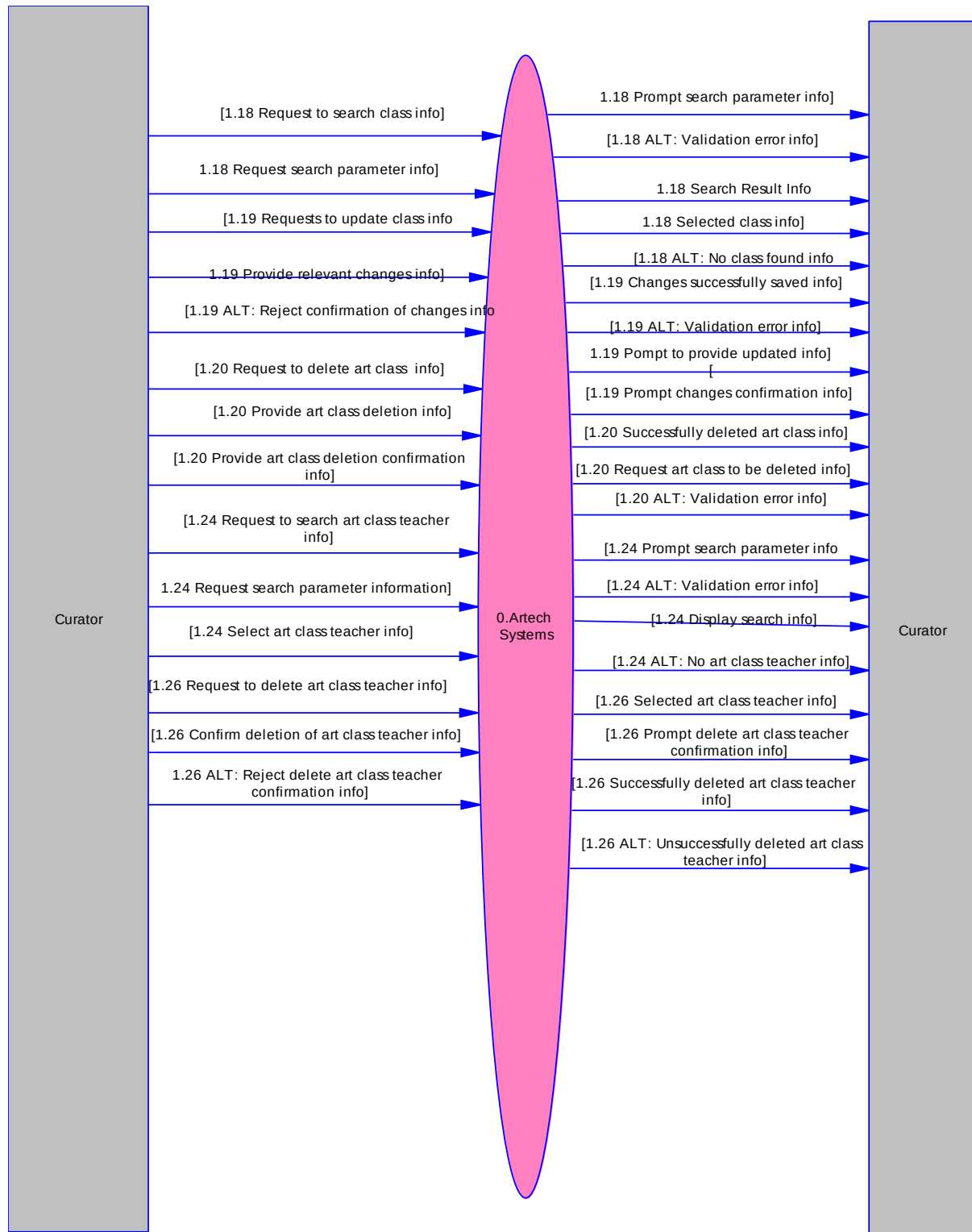


Figure 13: Context Diagram part 4 – Subsystem 1

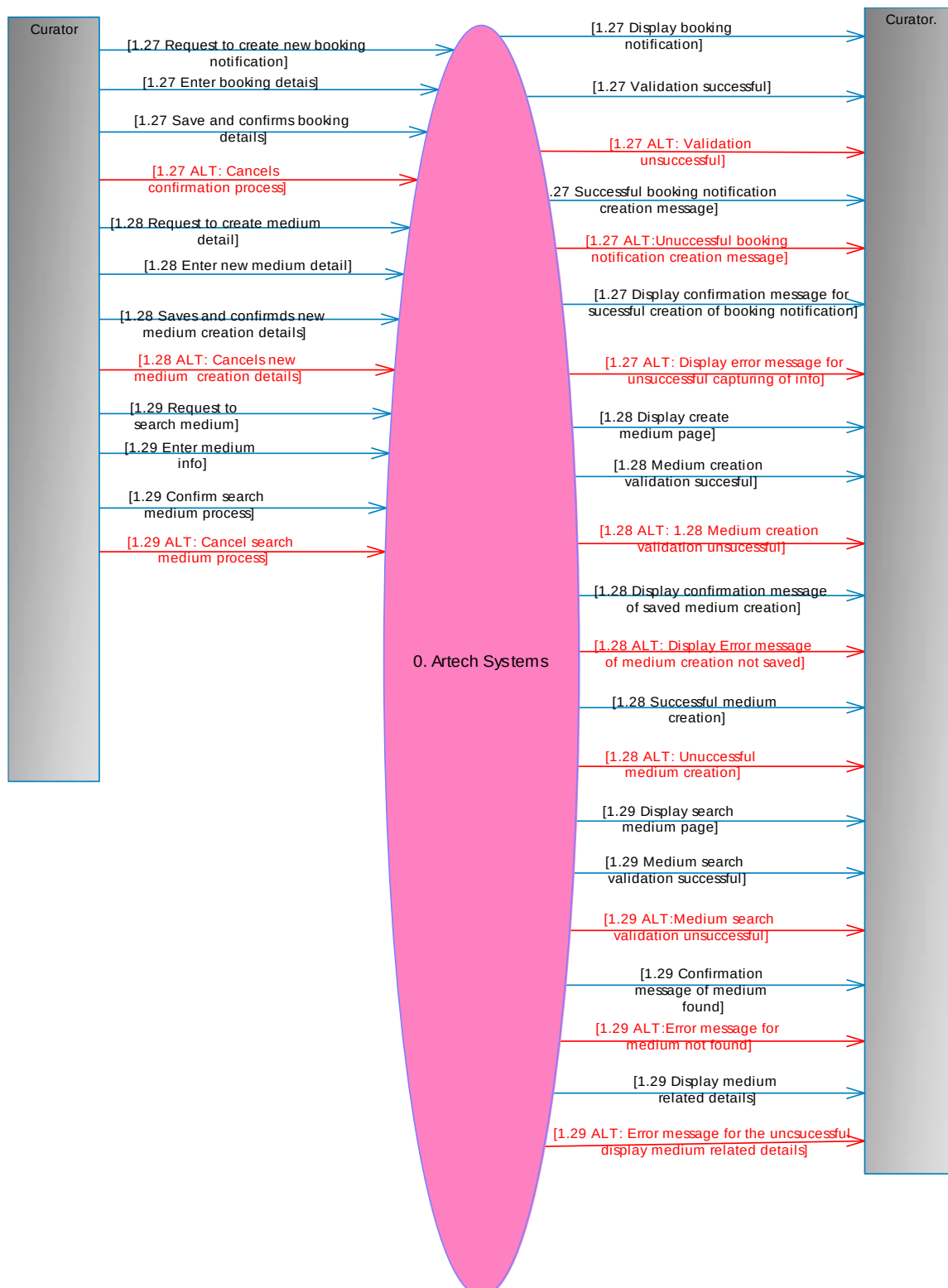


Figure 14: Context Diagram part 5 – Subsystem 1

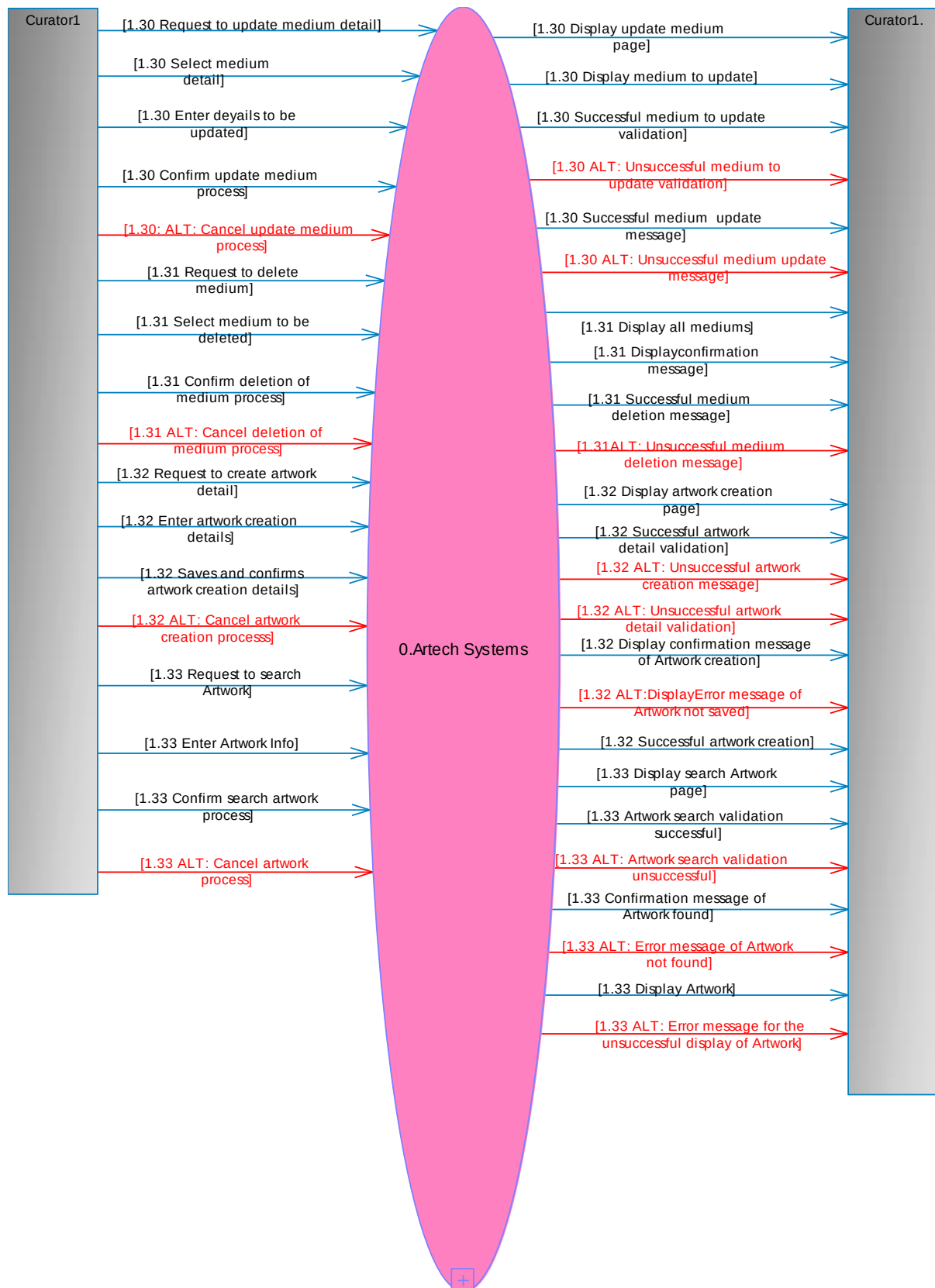


Figure 15: Context Diagram part 6– Subsystem 1

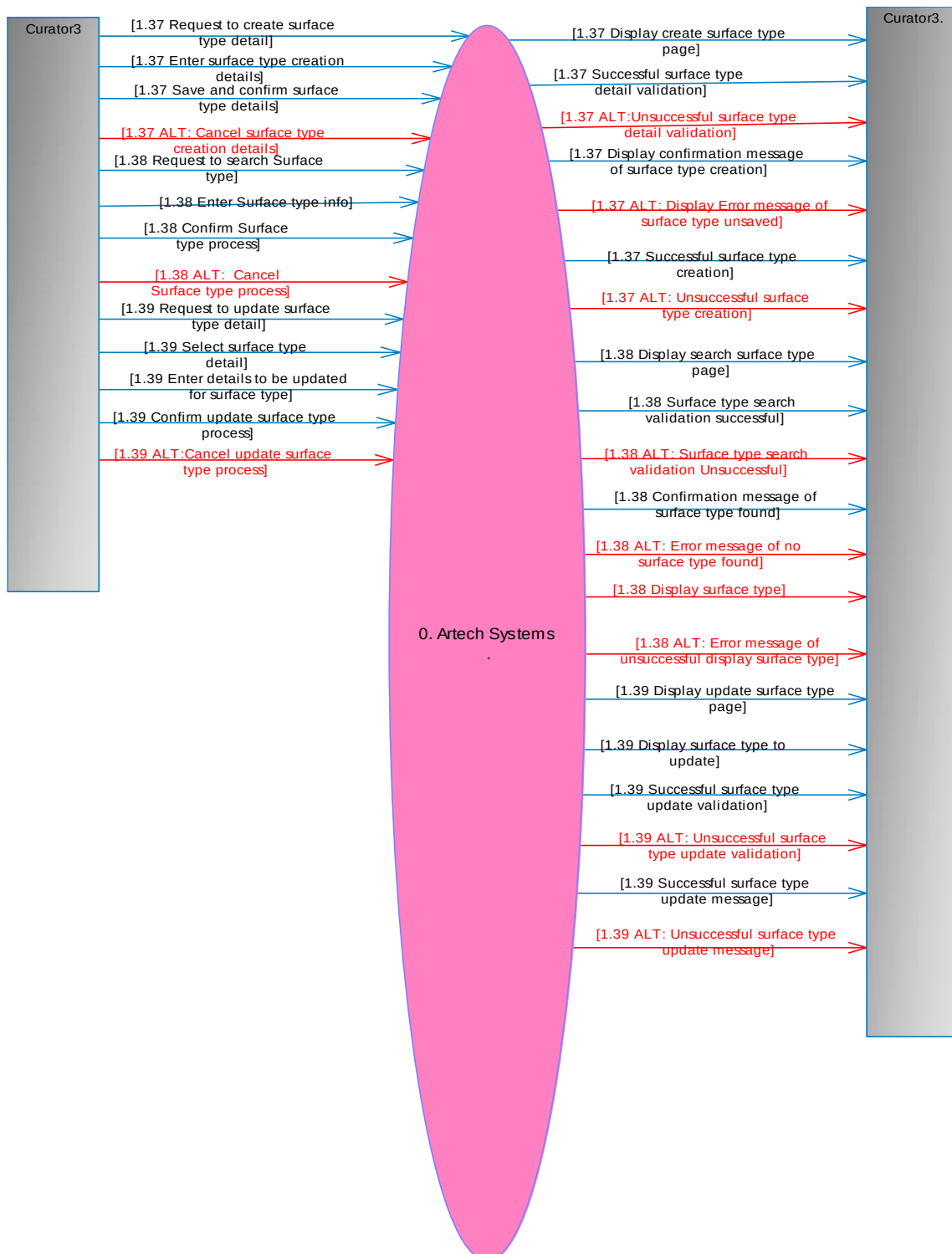


Figure 16: Context Diagram part 7 – Subsystem 1

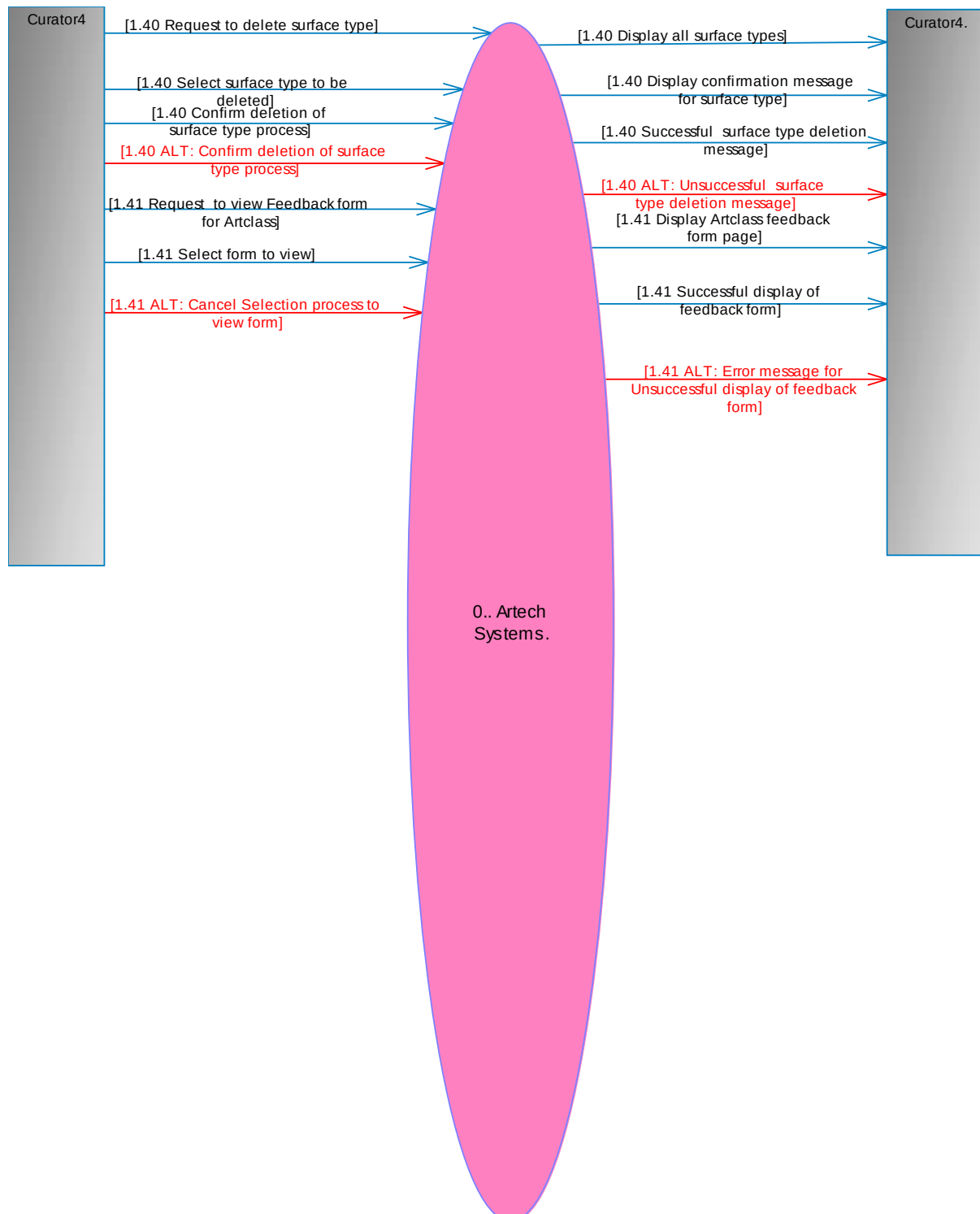


Figure 17: Context Diagram part 8 – Subsystem 1

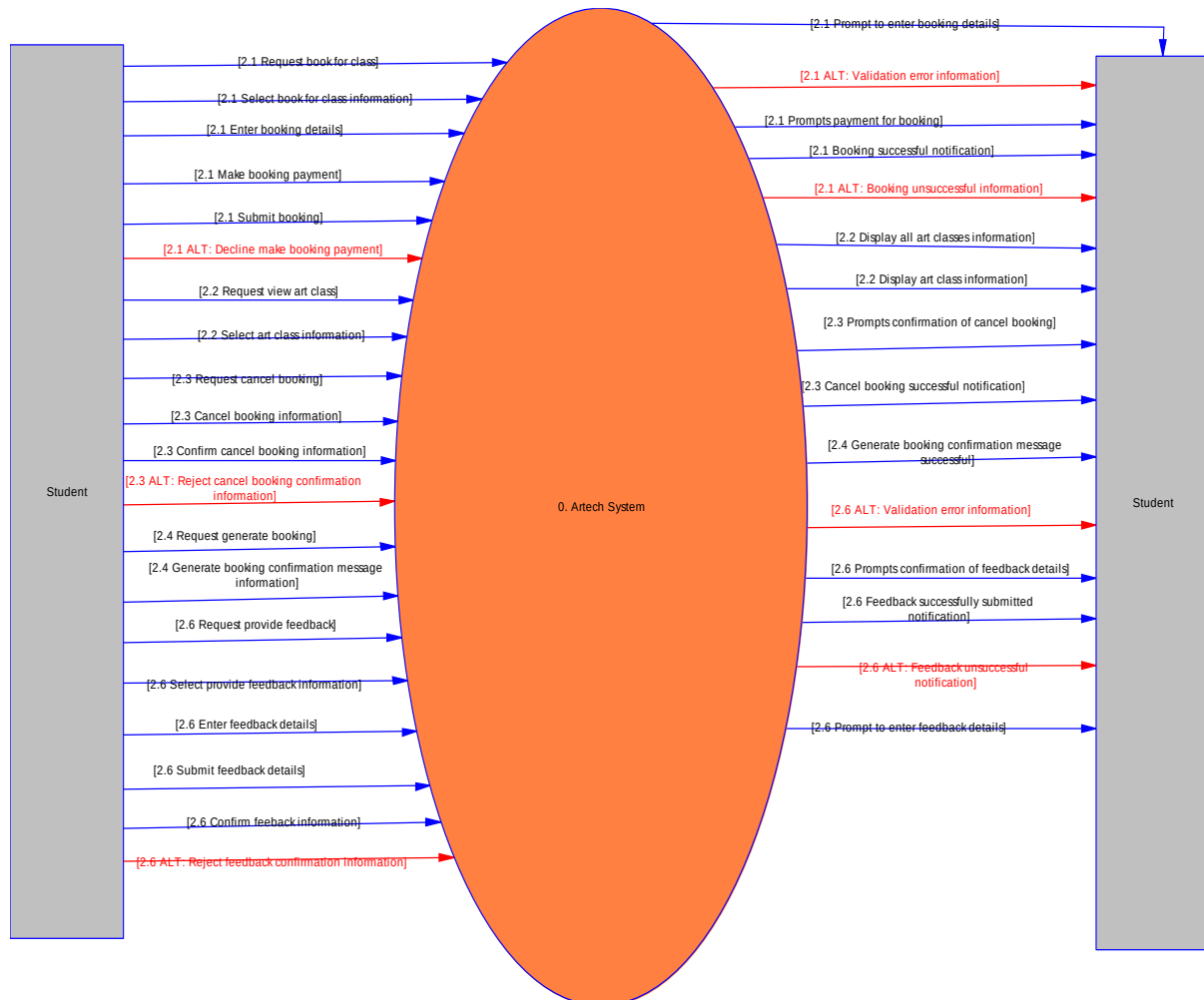


Figure 18: Context Diagram part 9 – Subsystem 2

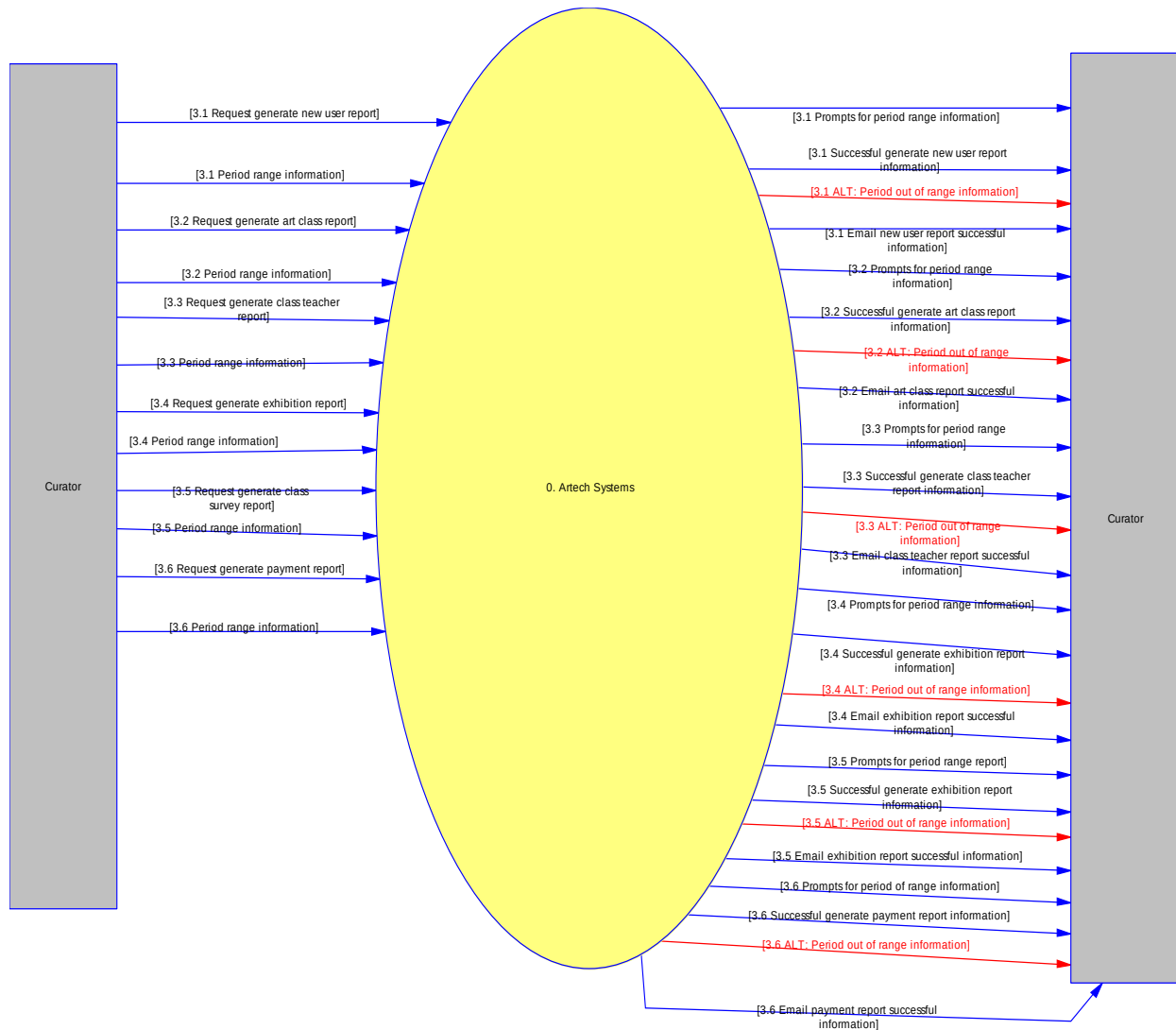


Figure 19: Context Diagram part 10 – Subsystem 3

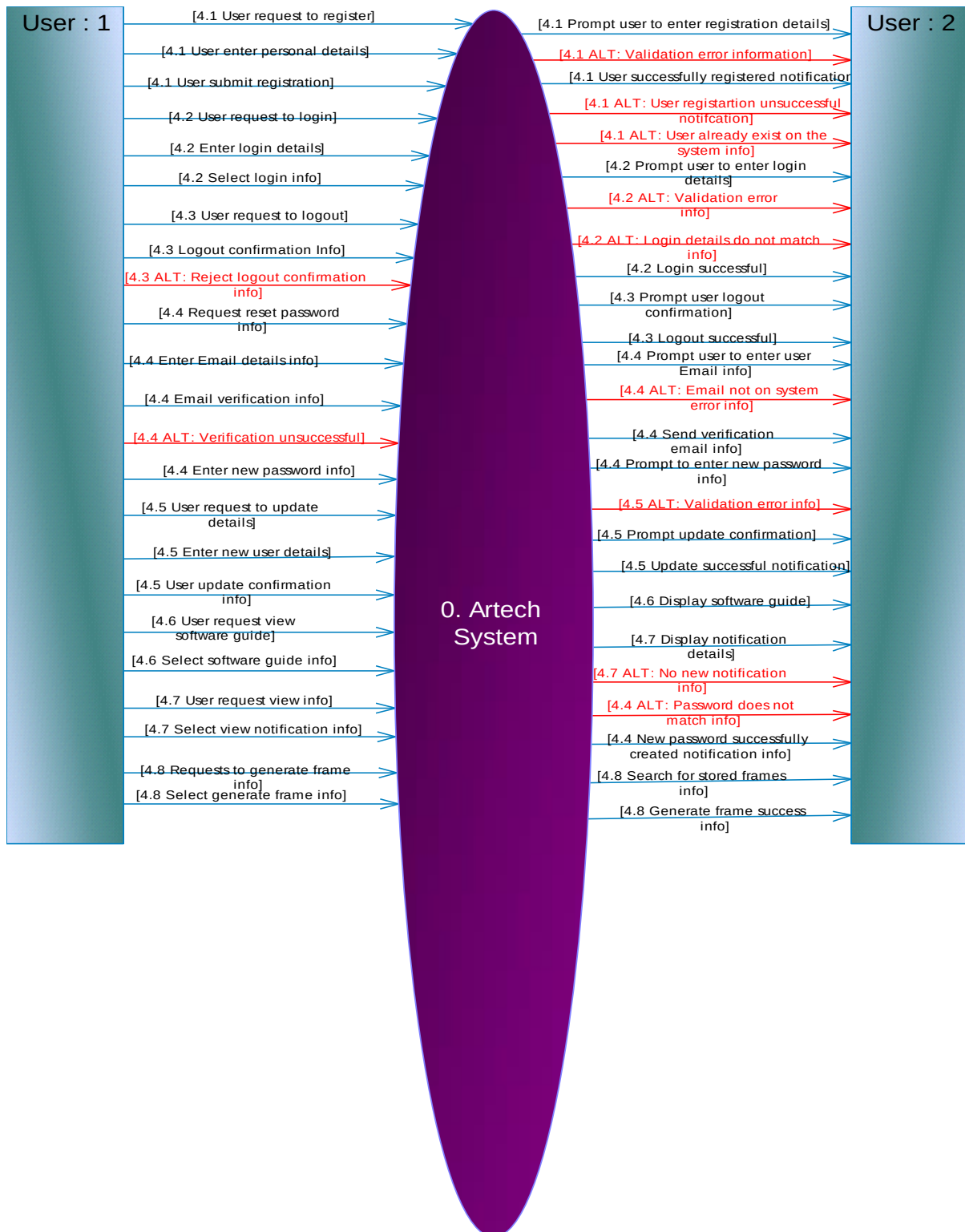


Figure 20: Context Diagram part 11 – Subsystem 4

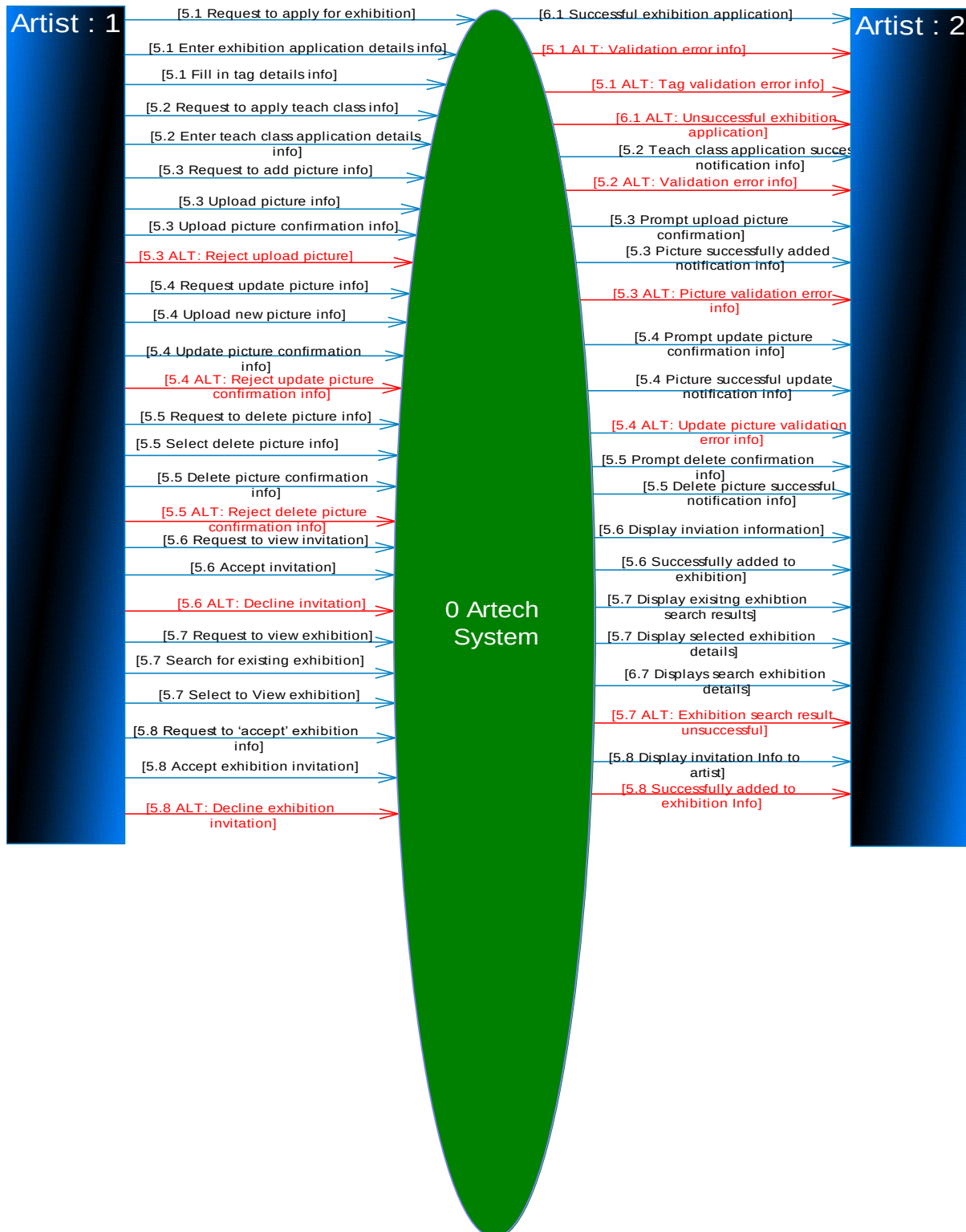


Figure 21: Context Diagram part 12 – Subsystem 5

3.2 Functional Decomposition Diagram

3.2.1 Administration Subsystem

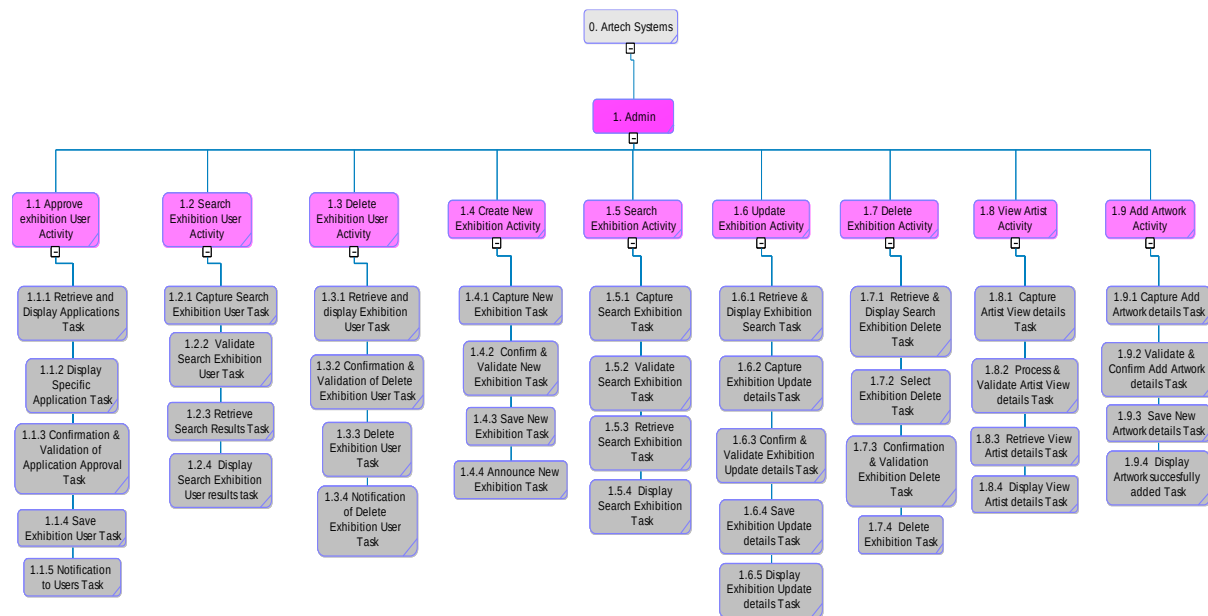


Figure 22: Decomposition Diagram part 1 – Subsystem 1

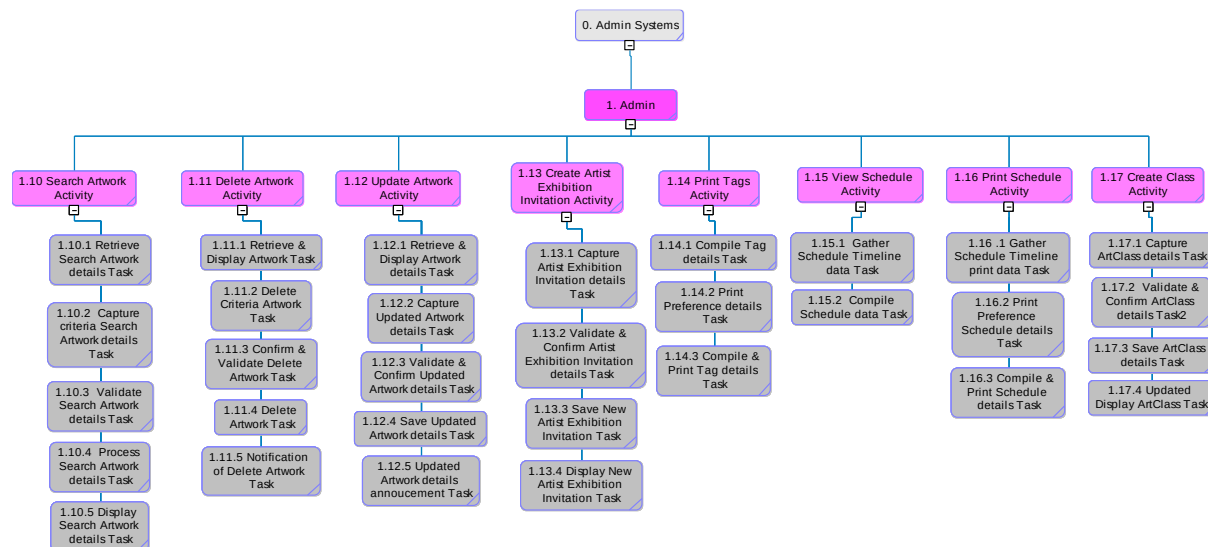


Figure 23: Decomposition Diagram part 2 – Subsystem 1

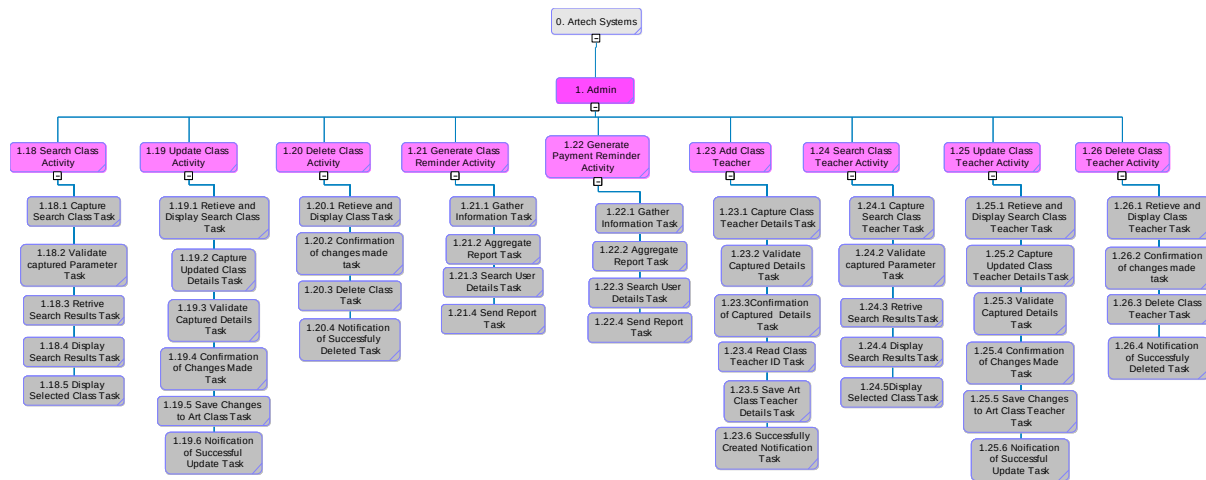


Figure 24: Decomposition Diagram part 3 – Subsystem 1

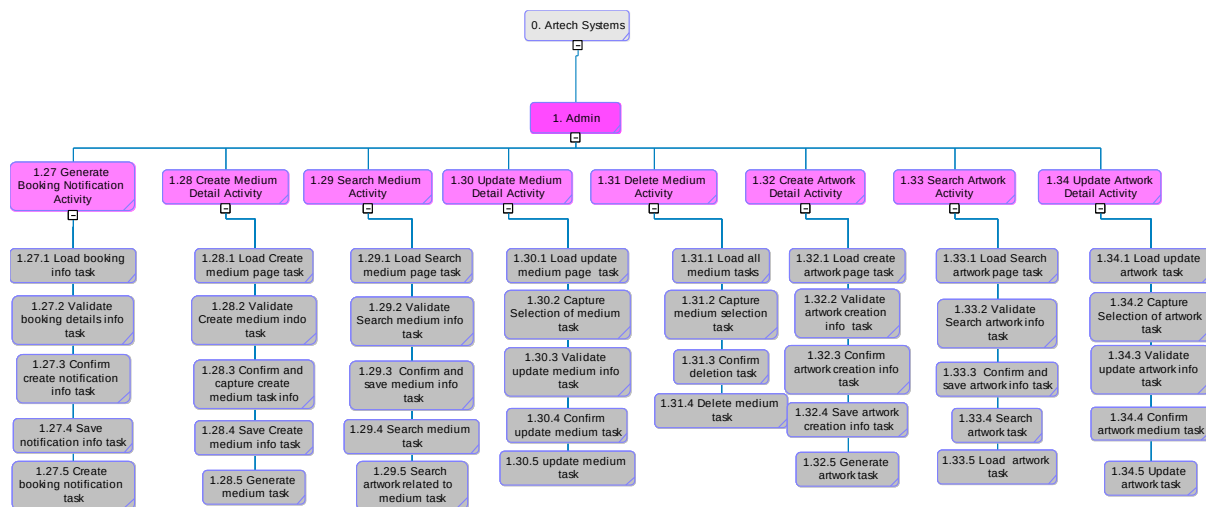


Figure 25: Decomposition Diagram part 4 – Subsystem 1

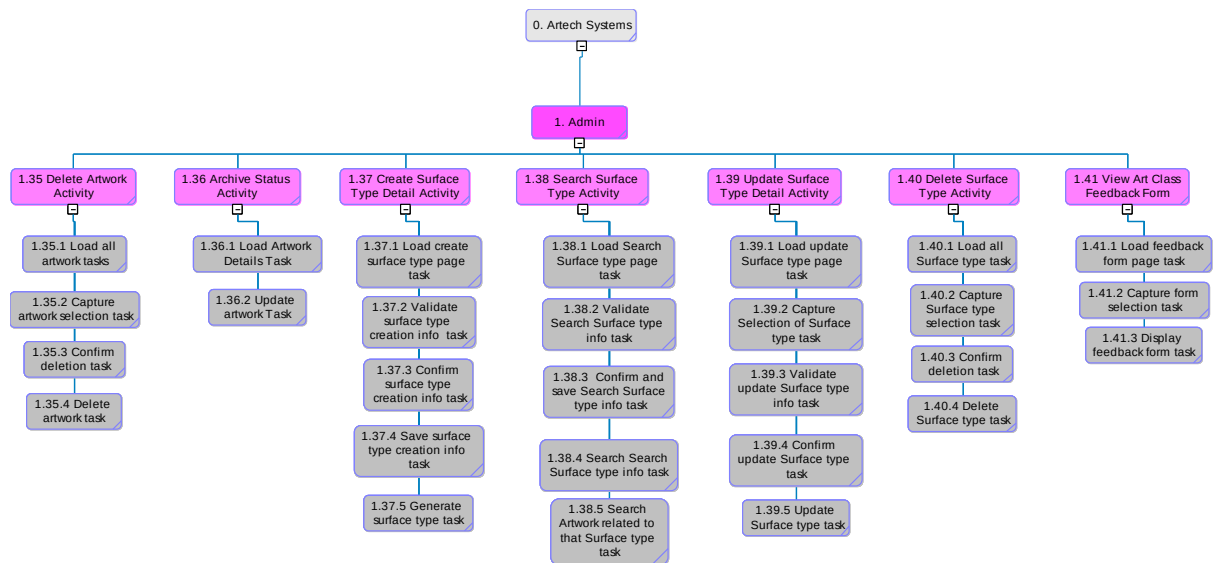


Figure 26: Decomposition Diagram part 5 – Subsystem 1

3.2.2 Art Class Subsystem

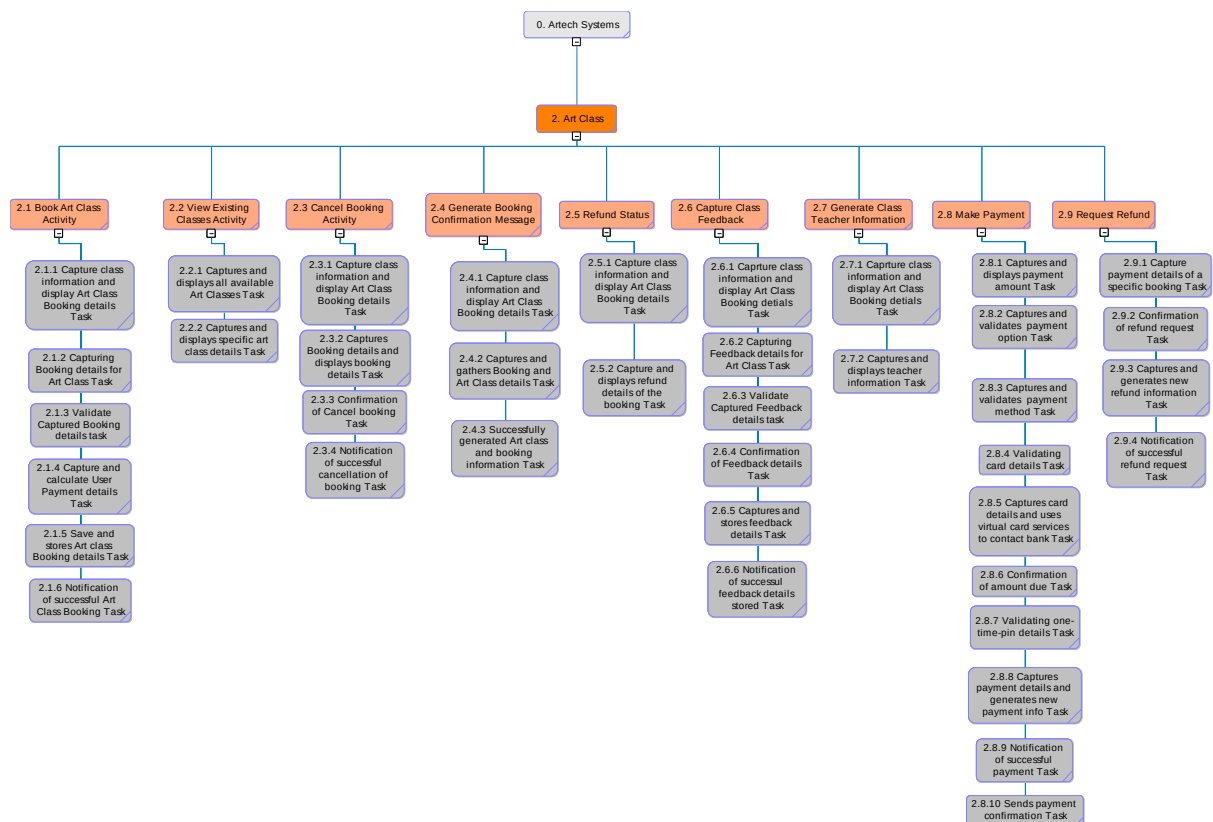


Figure 27: Decomposition Diagram part 6 – Subsystem 2

3.2.3 Reporting Subsystem

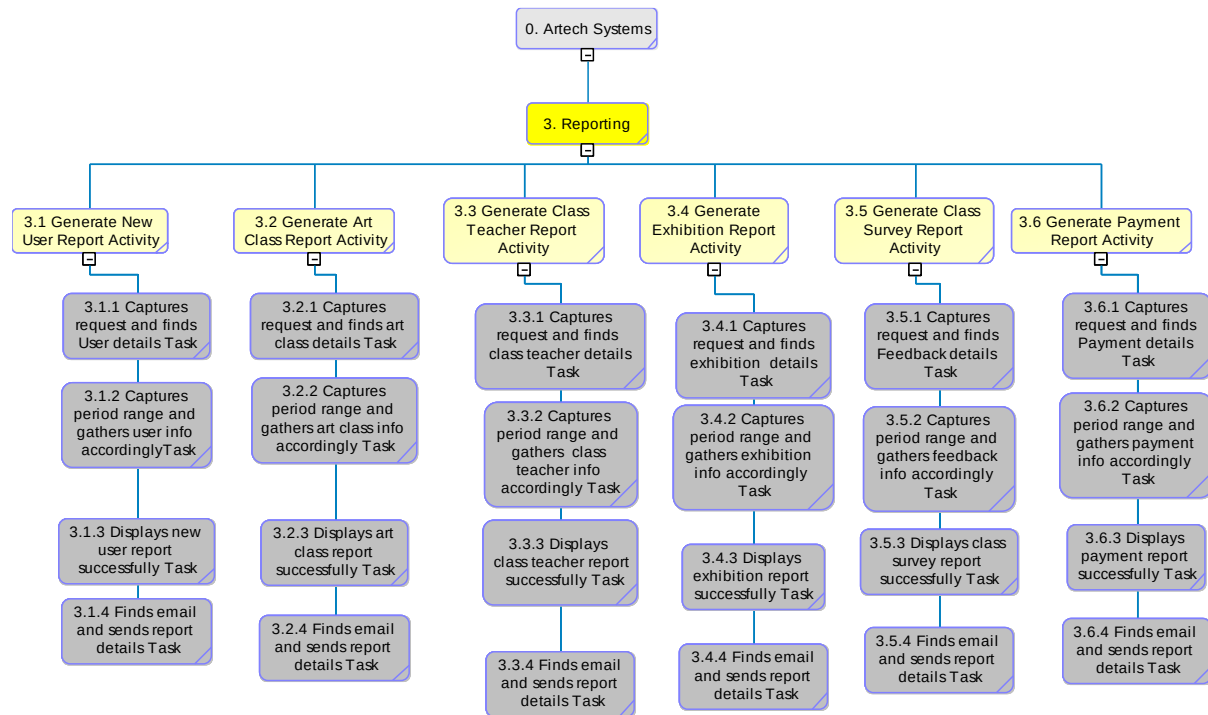


Figure 28: Decomposition Diagram part 7 – Subsystem 3

3.2.4 User Subsystem

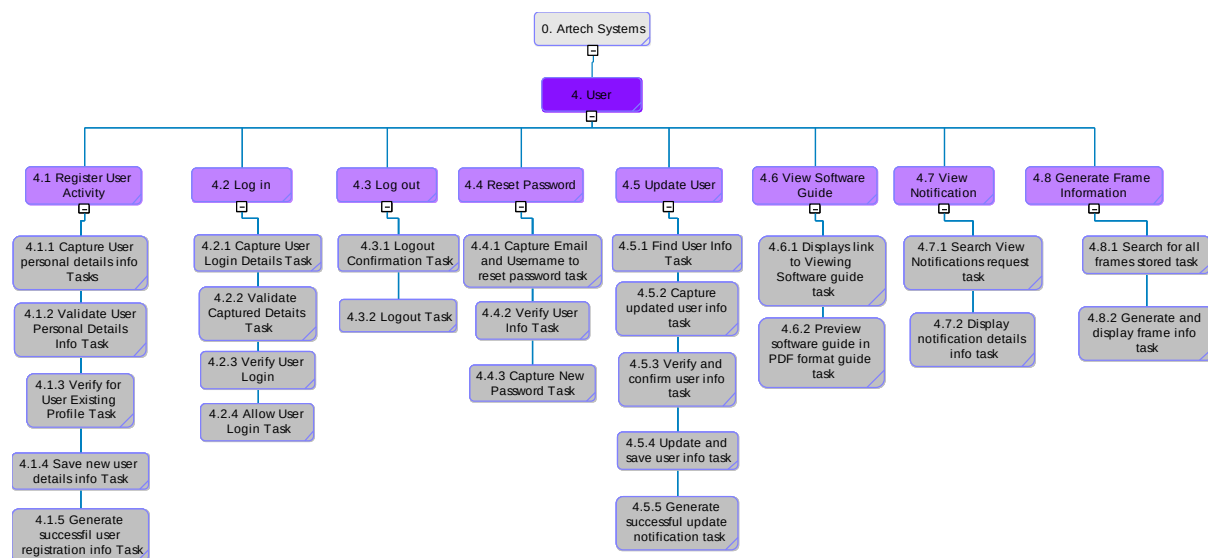


Figure 29: Decomposition Diagram part 8 – Subsystem 4

3.2.5 User Subsystem

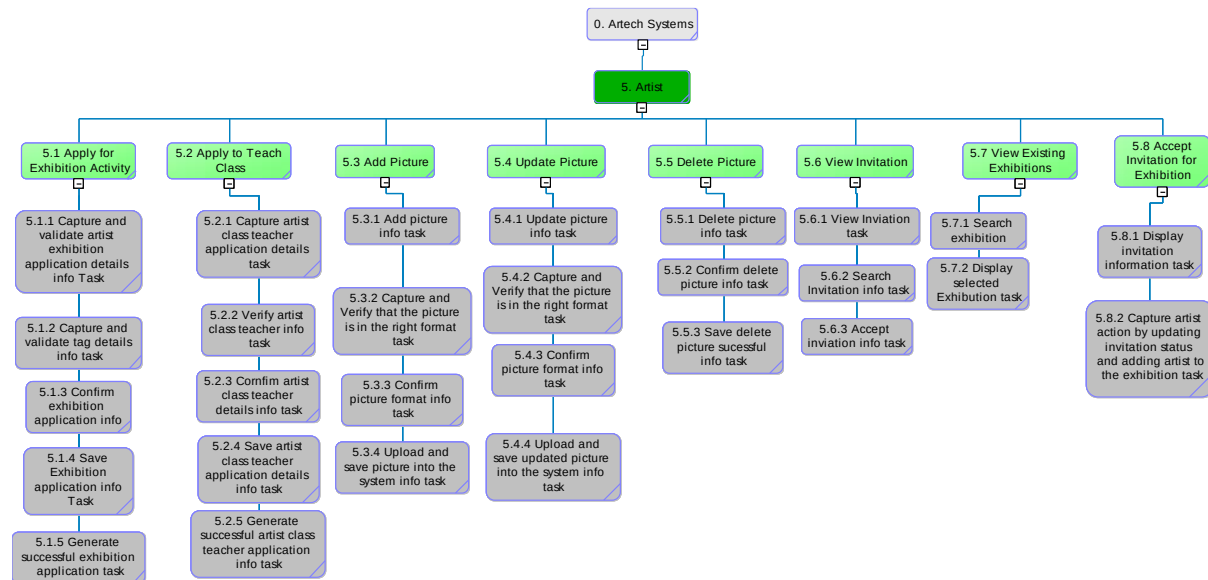


Figure 30: Decomposition Diagram part 9 – Subsystem 5

3.3 Data Flow Diagrams (High to Low)

3.3.1 High Level Data Flow Diagrams

3.3.1.1 Administration Subsystem



Figure 31: High Level Diagram part 1 - Subsystem 1

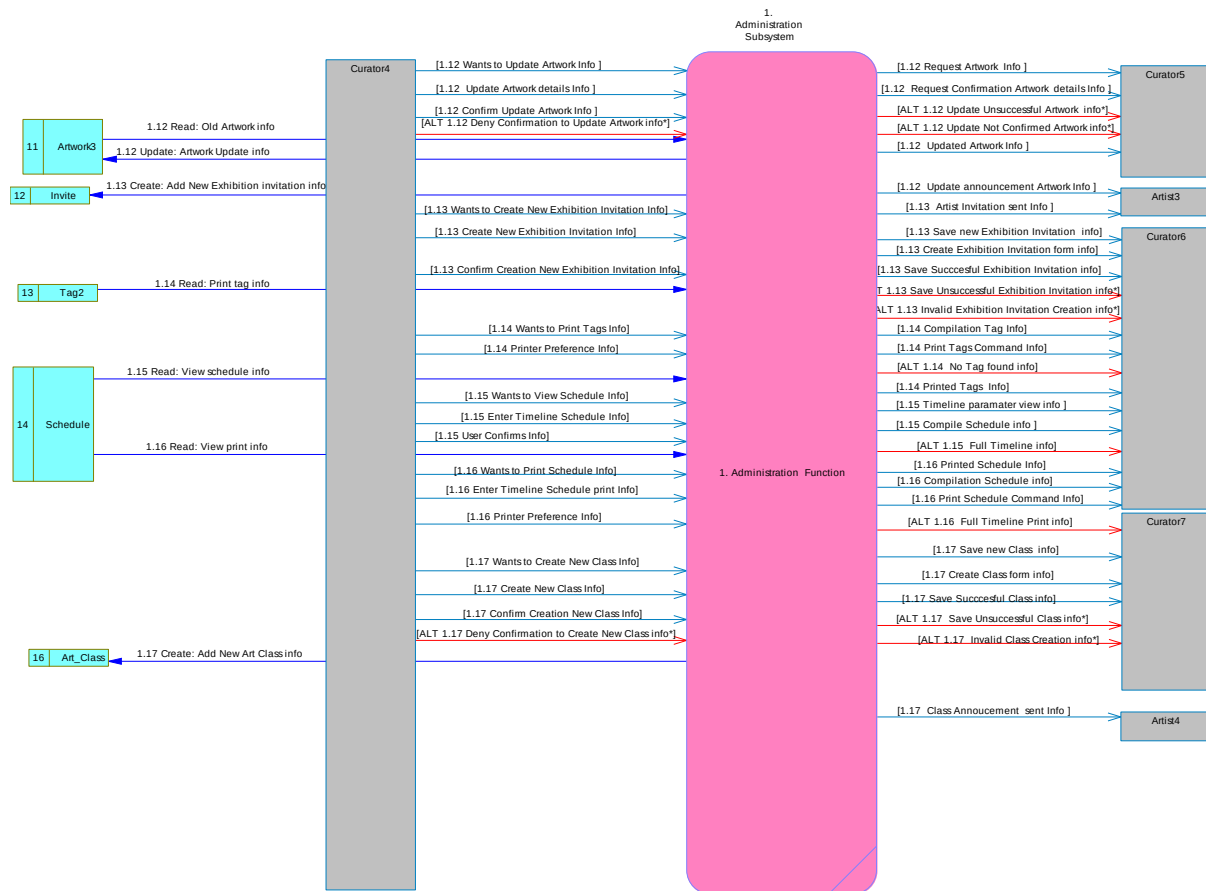


Figure 32: High Level Diagram part 2 - Subsystem 1

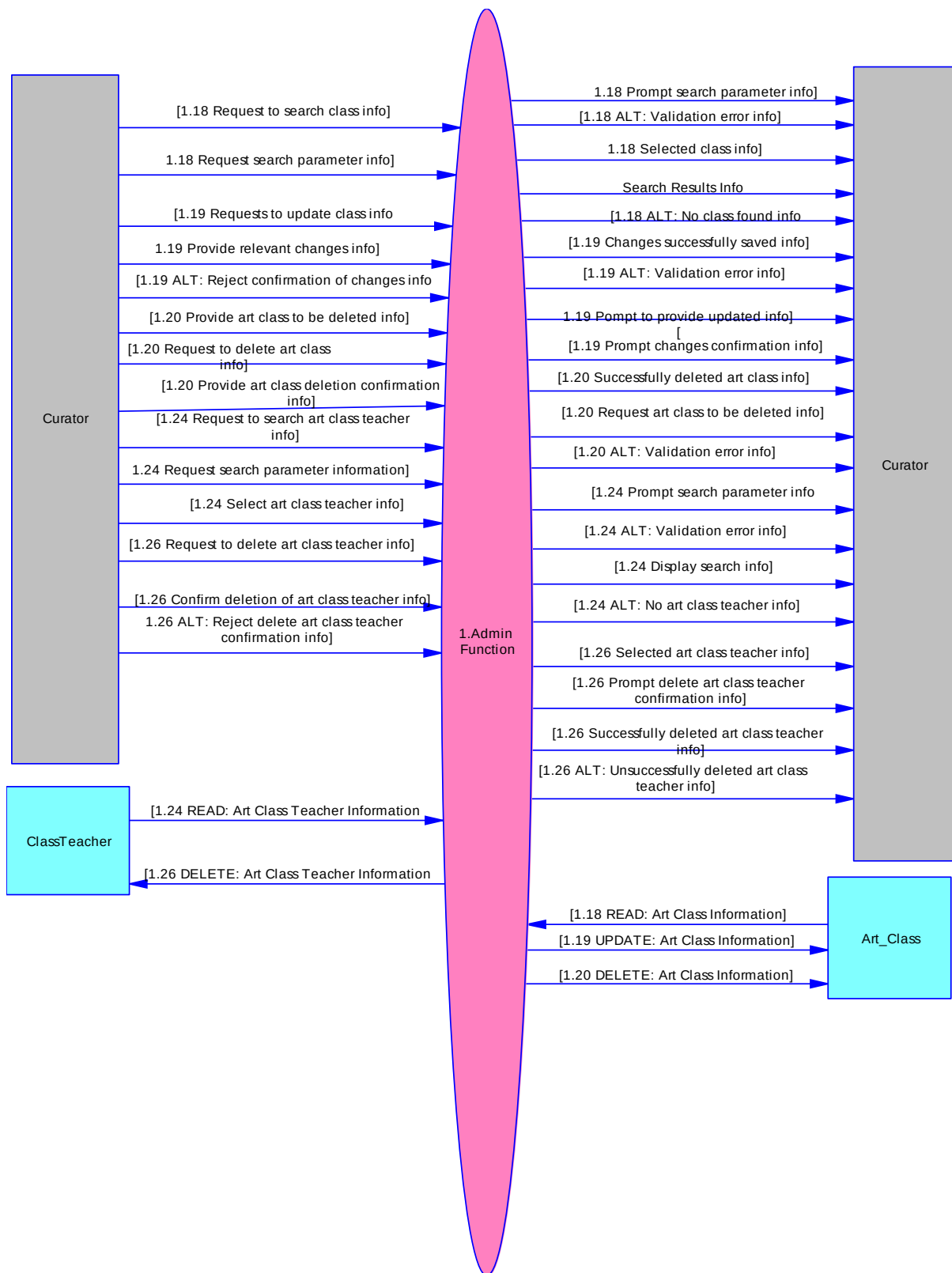


Figure 33: High Level Diagram part 3 - Subsystem 1

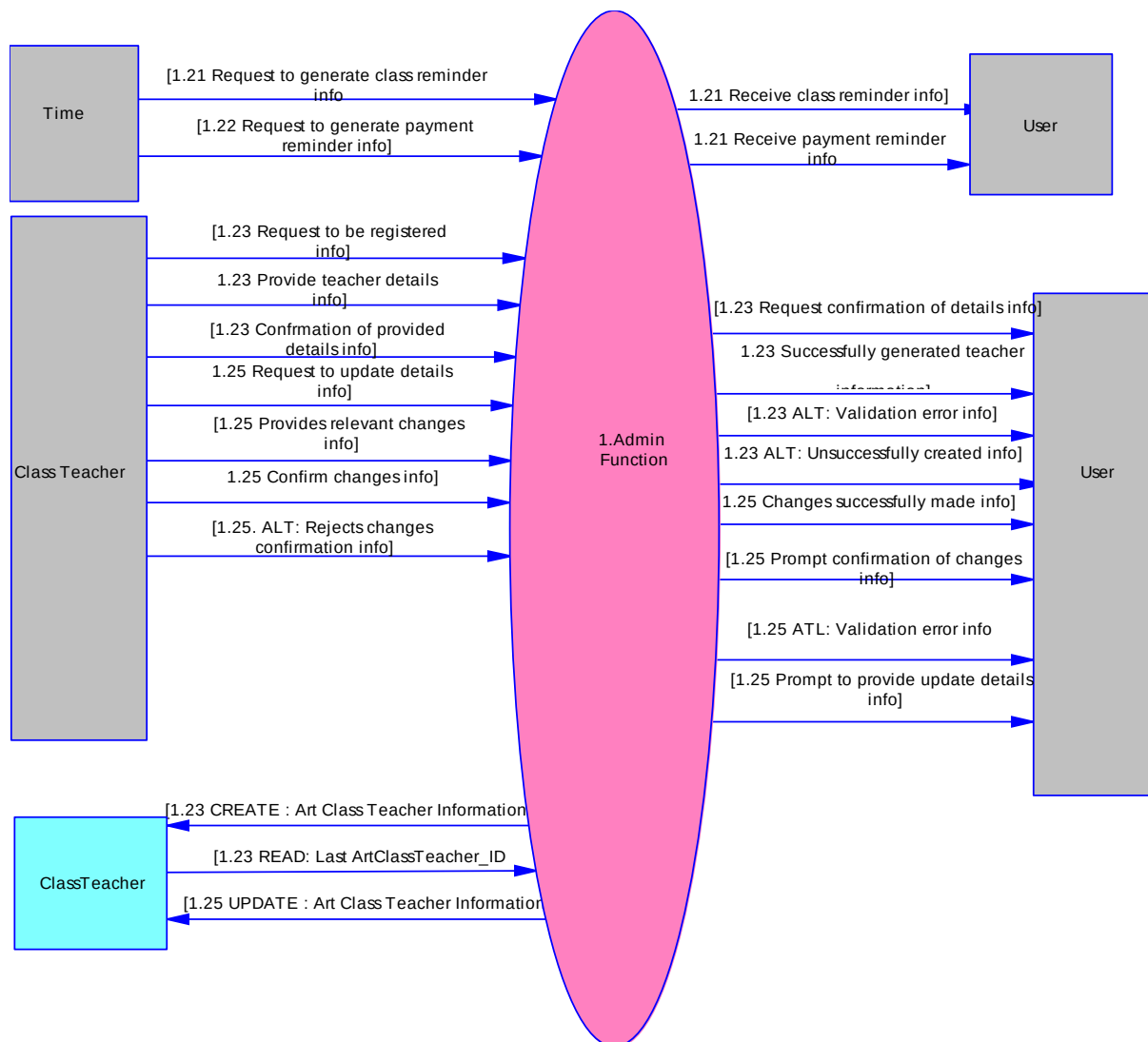


Figure 34: High Level Diagram part 4 - Subsystem 1

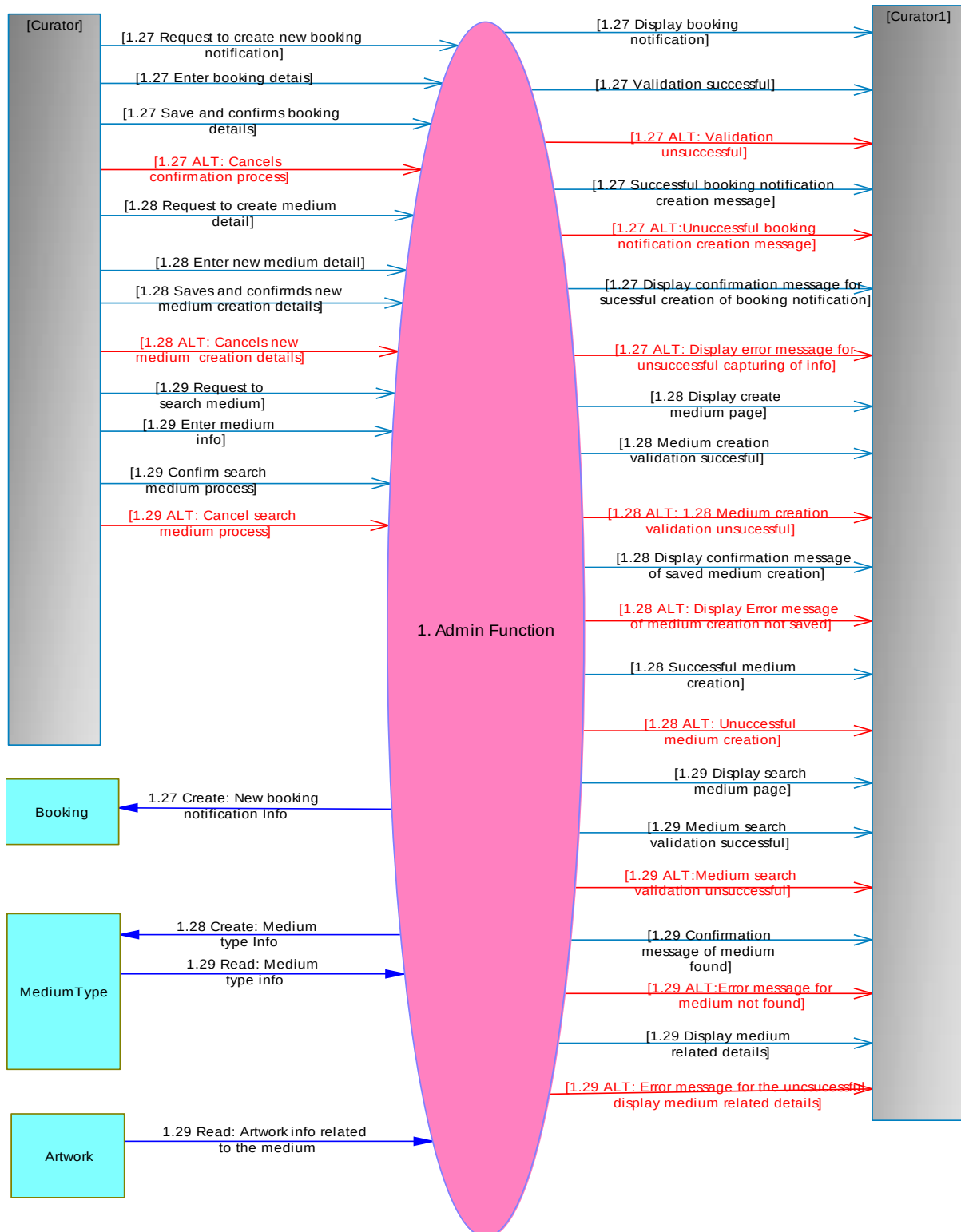


Figure 35: High Level Diagram part 5 - Subsystem 1

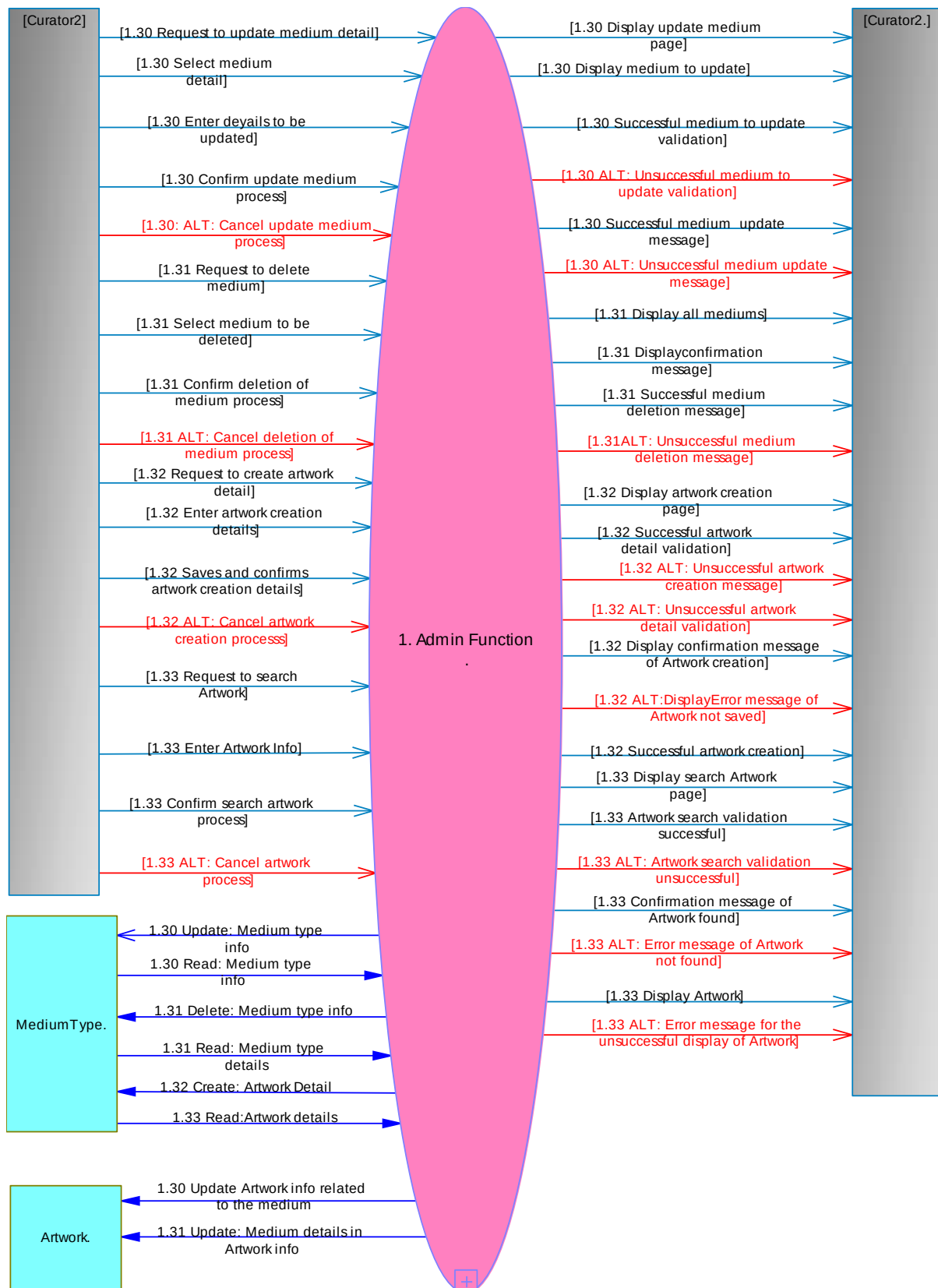


Figure 36: High Level Diagram part 6 - Subsystem 1

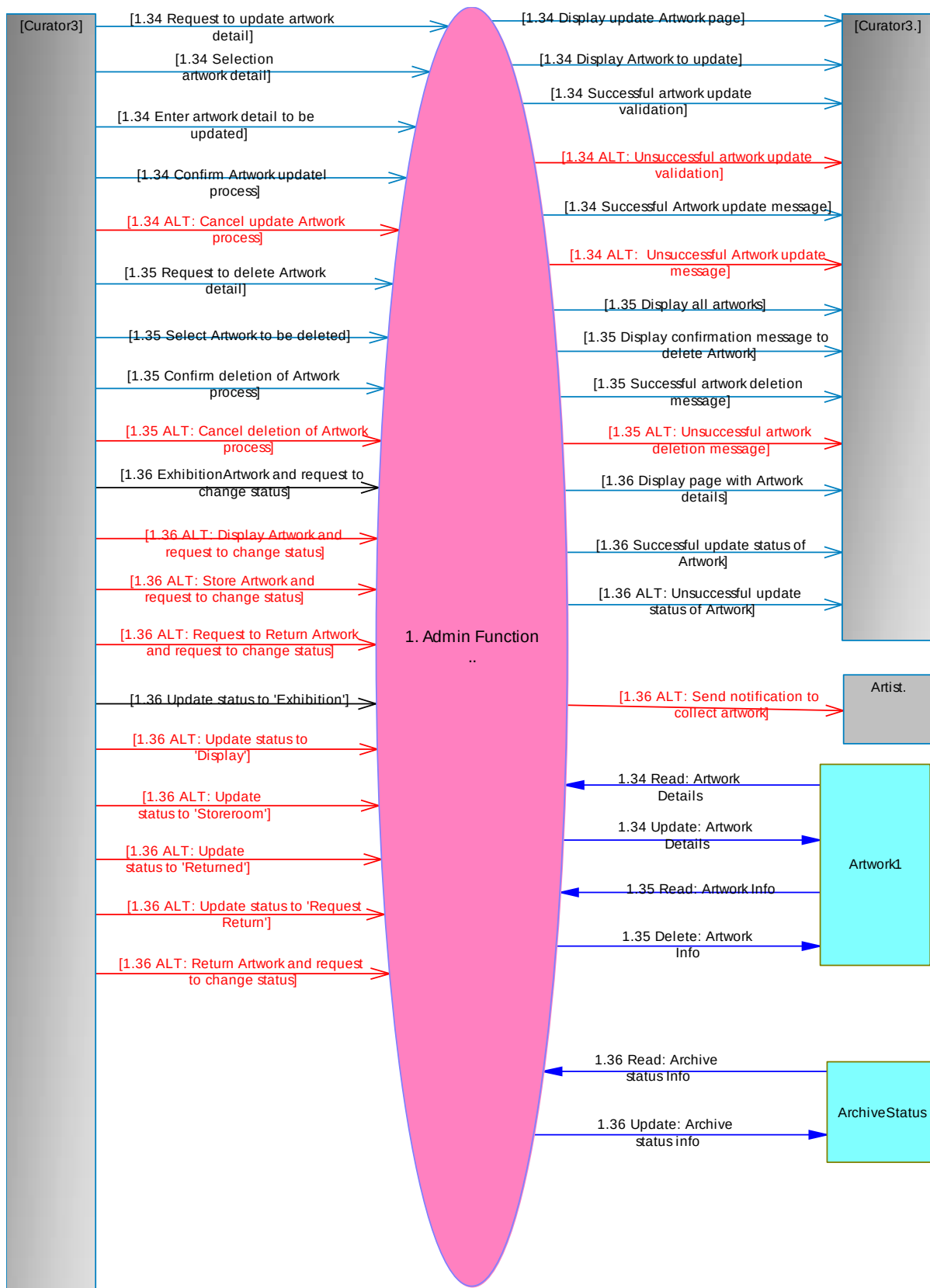


Figure 37: High Level Diagram part 7 - Subsystem 1

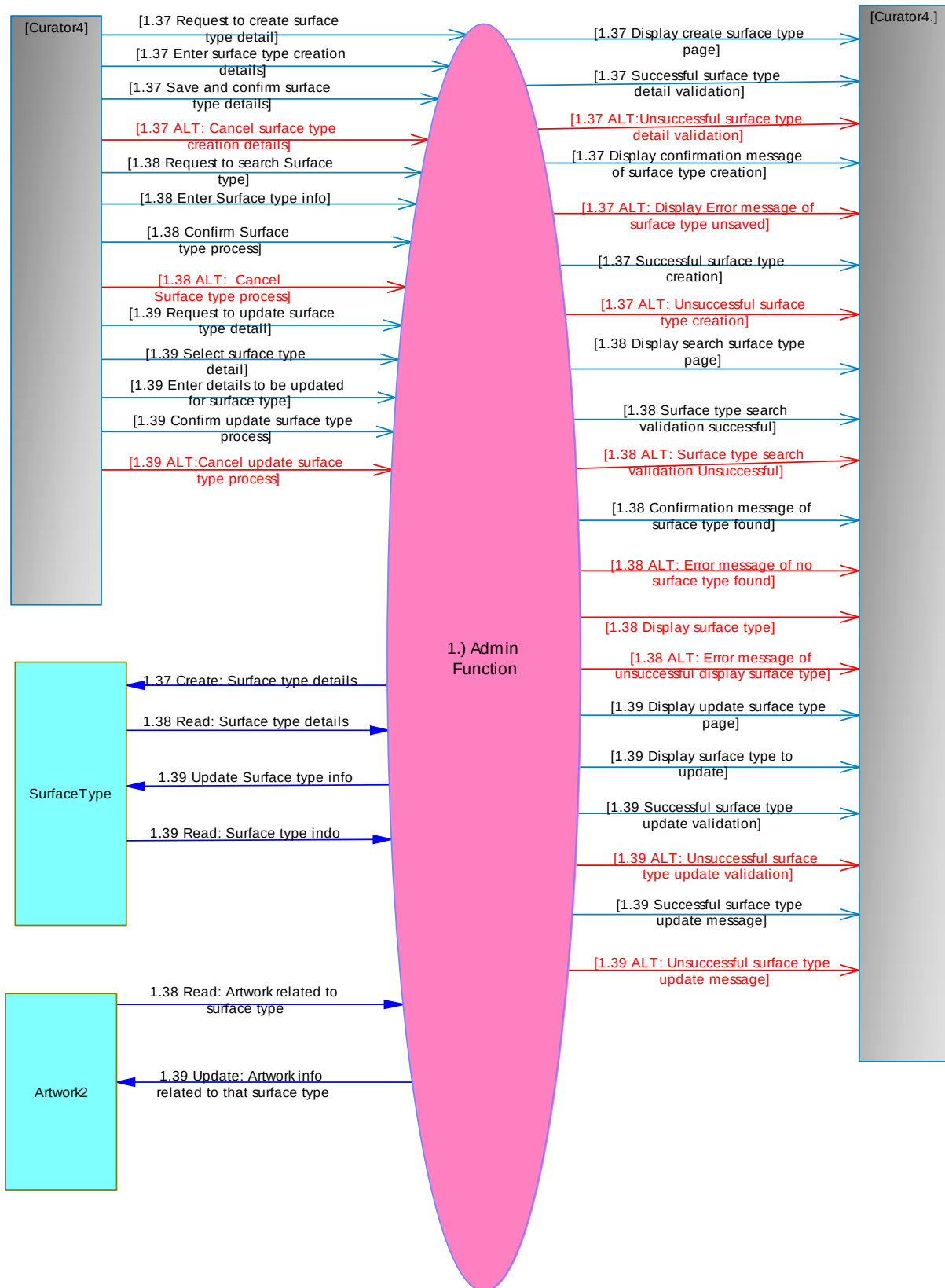


Figure 38: High Level Diagram part 8 - Subsystem 1

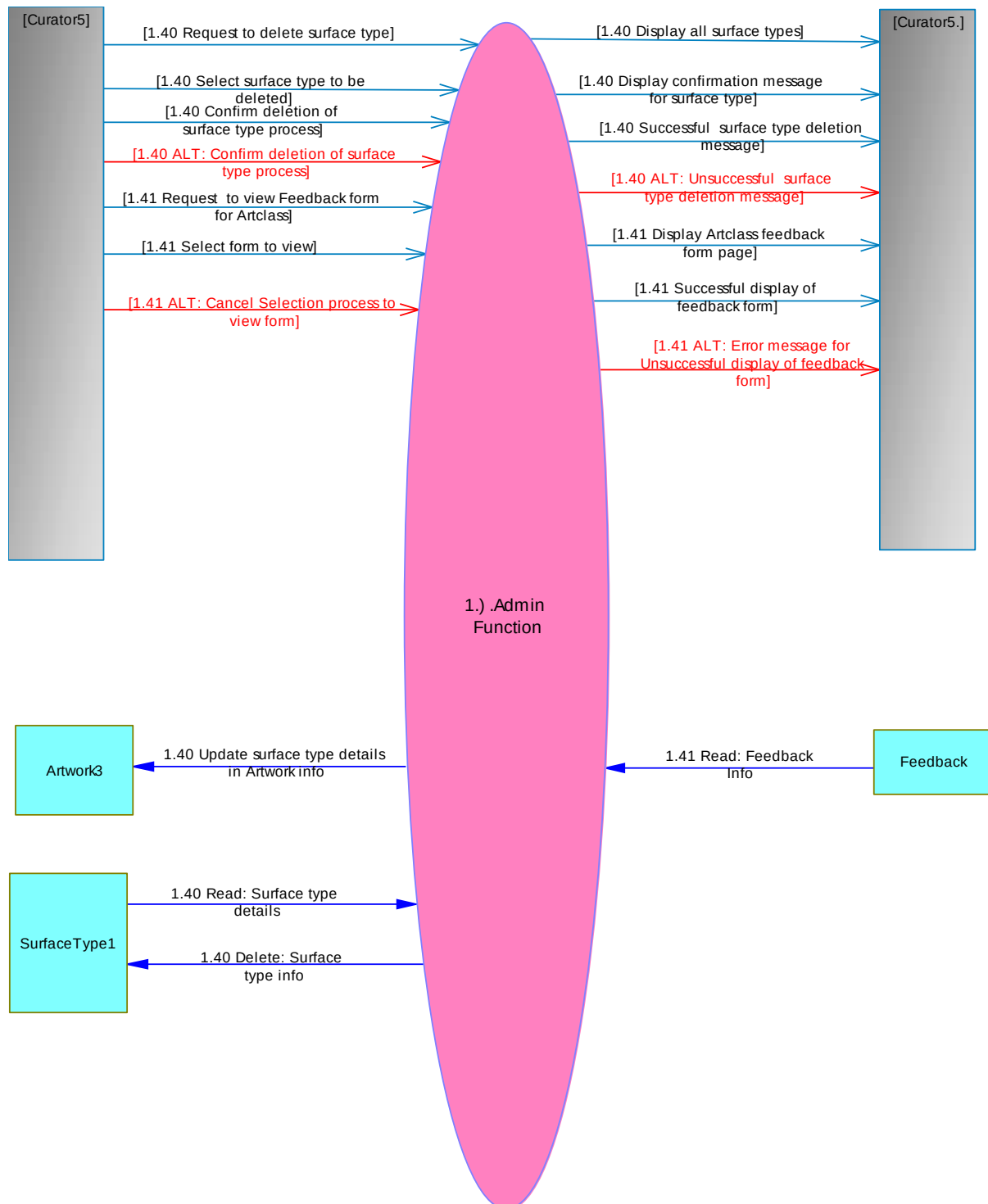


Figure 39: High Level Diagram part 9 - Subsystem 1

3.3.1.2 Art Class Subsystem

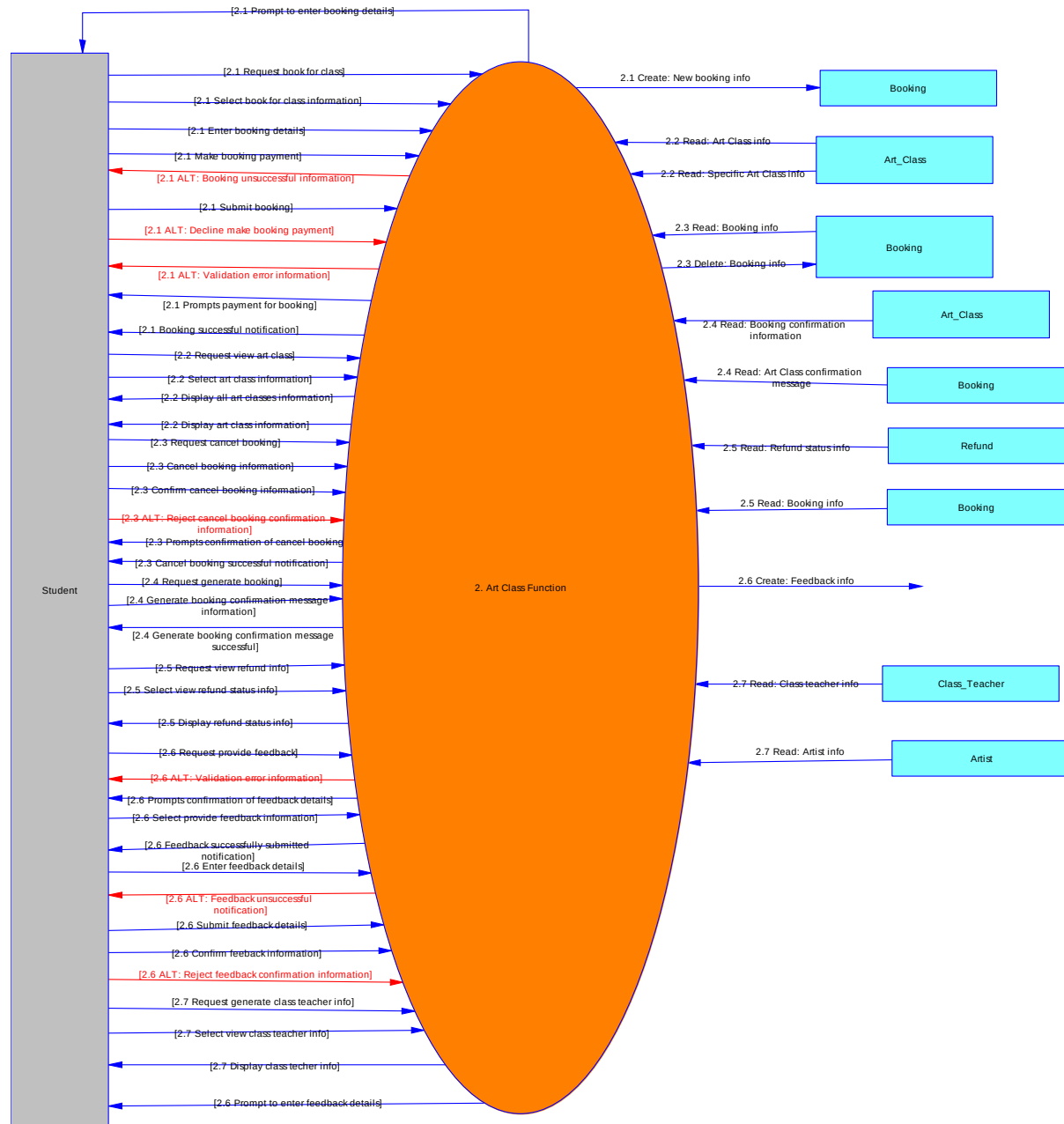


Figure 40: High Level Diagram part 1 - Subsystem 2

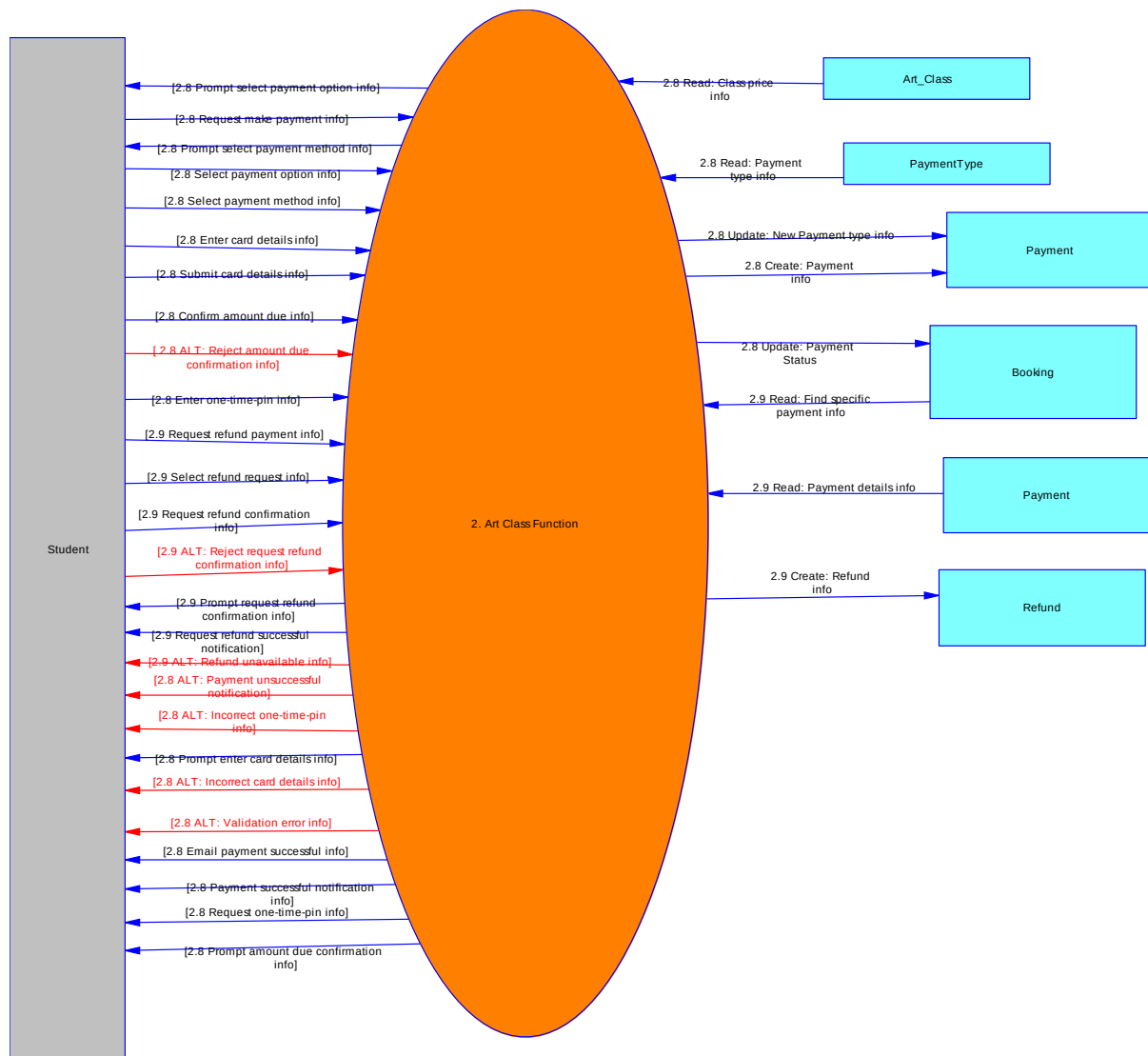


Figure 41: High Level Diagram part 2 - Subsystem 2

3.3.1.3 Reporting Subsystem

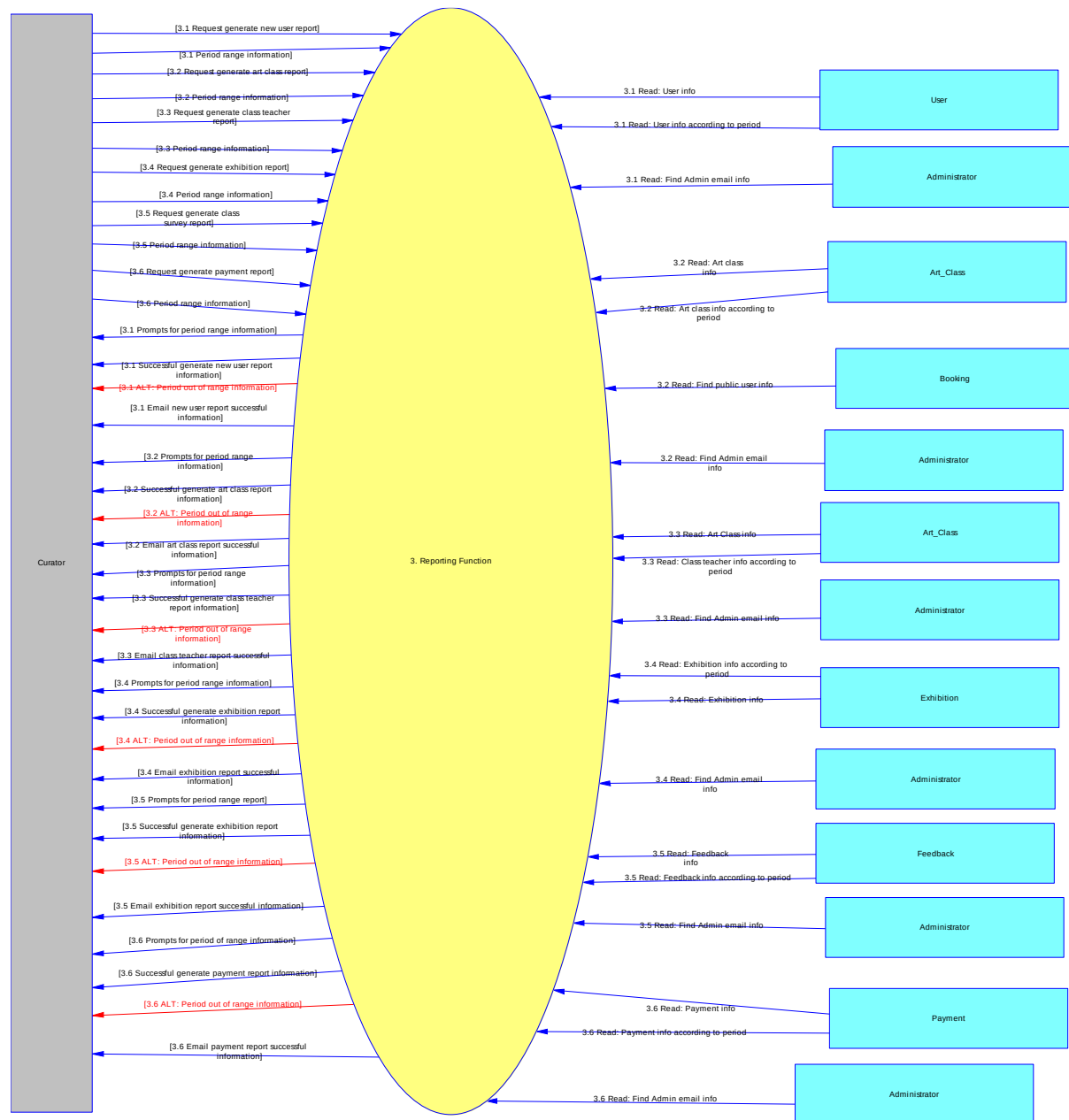


Figure 42: High Level Diagram - Subsystem 3

3.3.1.4 User Subsystem

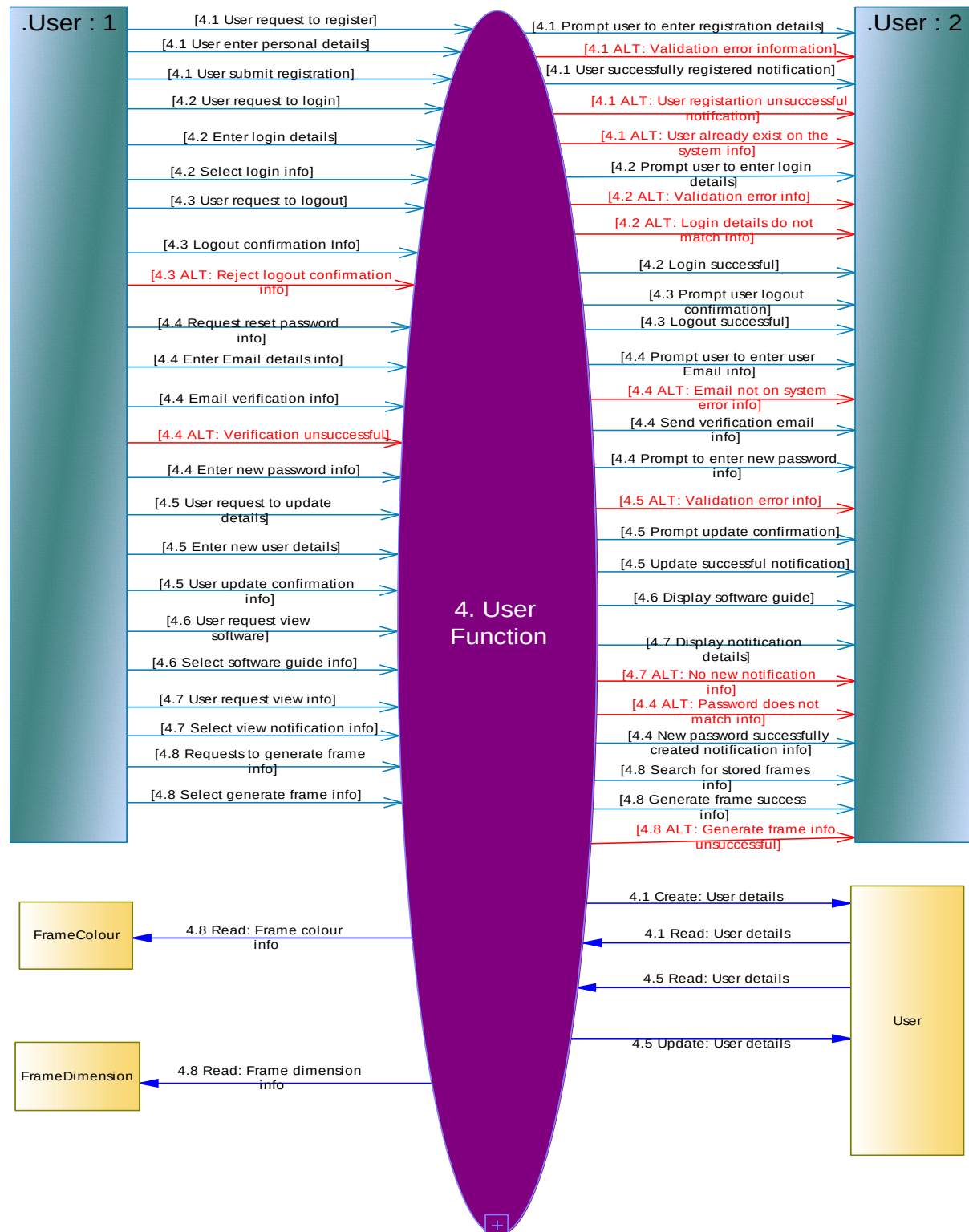


Figure 43: High Level Diagram- Subsystem 4

3.3.1.5 Artist Subsystem

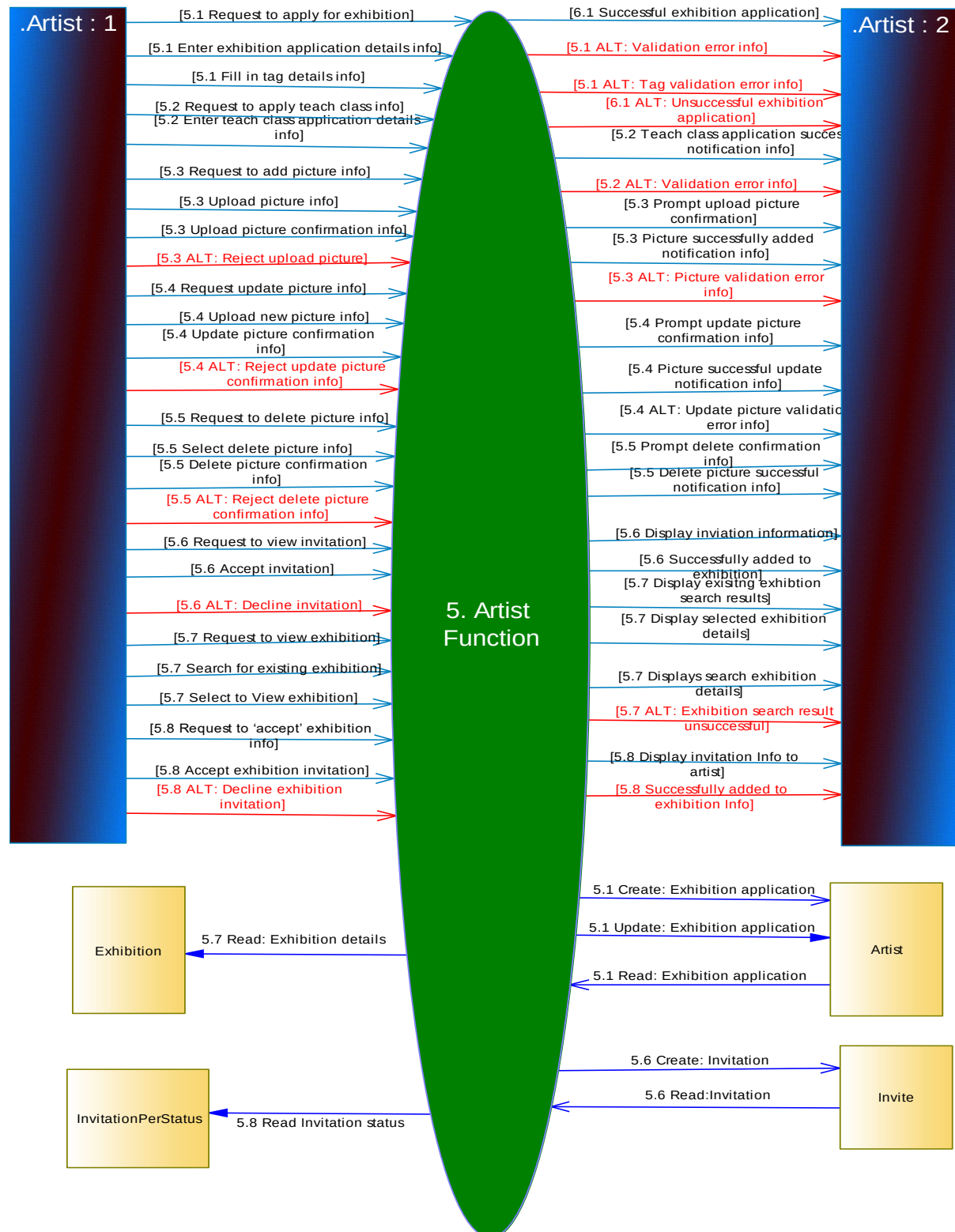


Figure 44: High Level Diagram- Subsystem 5

3.3.2 Middle Level Data Flow Diagrams

3.3.2.1 Administration Subsystem

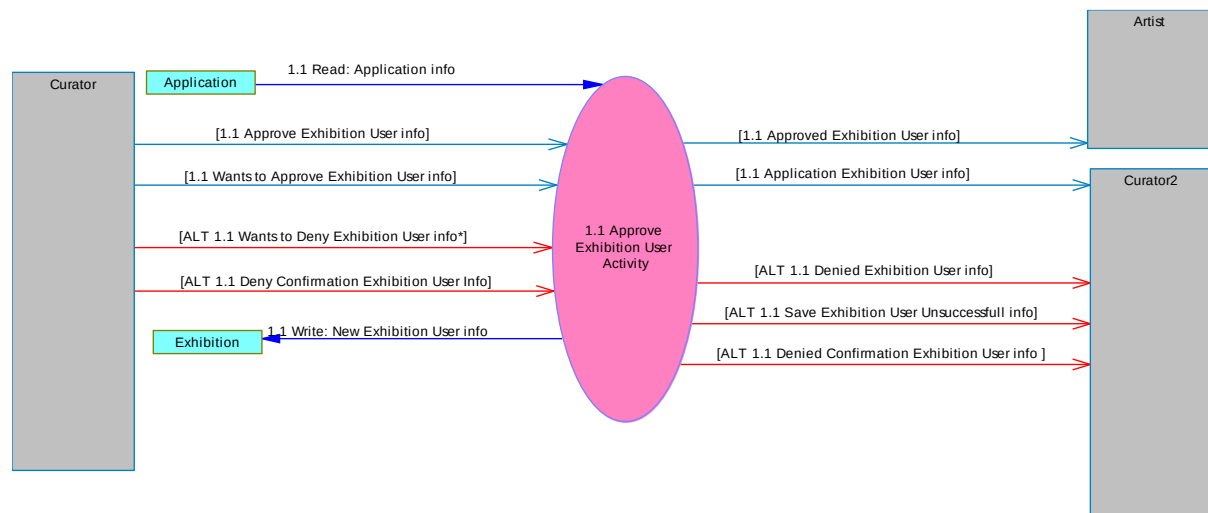


Figure 45: Middle Level DFD: 1.1. Approve Exhibition User

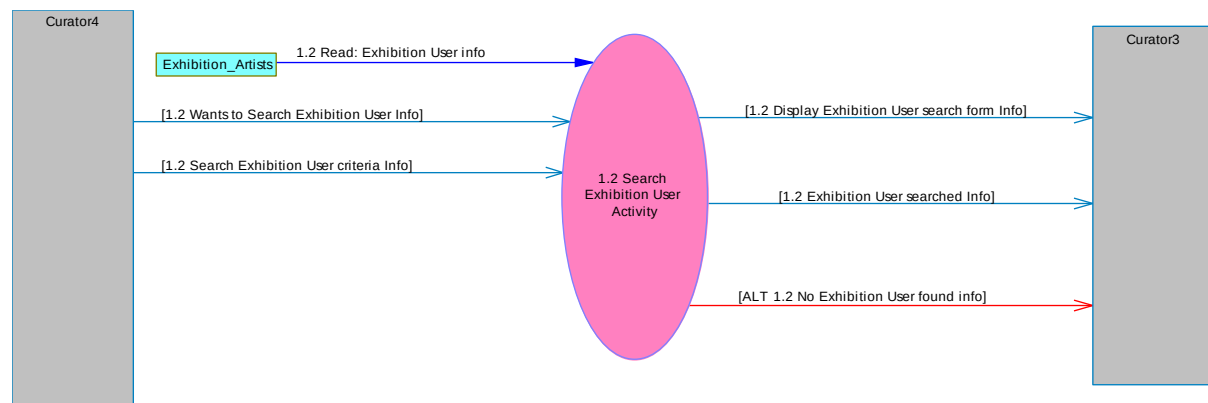


Figure 46: Middle Level DFD: 1.2 Search Exhibition User

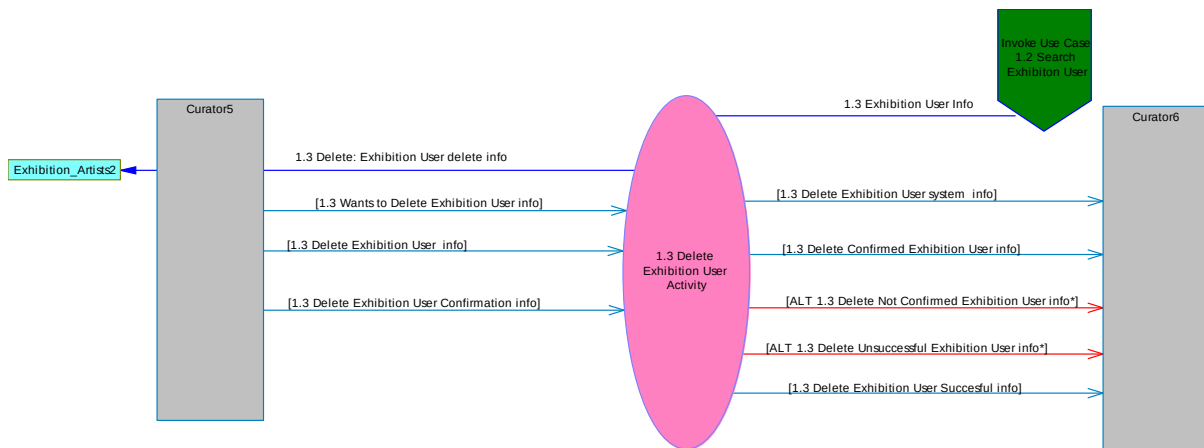


Figure 47: Middle Level DFD: 1.3 Delete Exhibition User

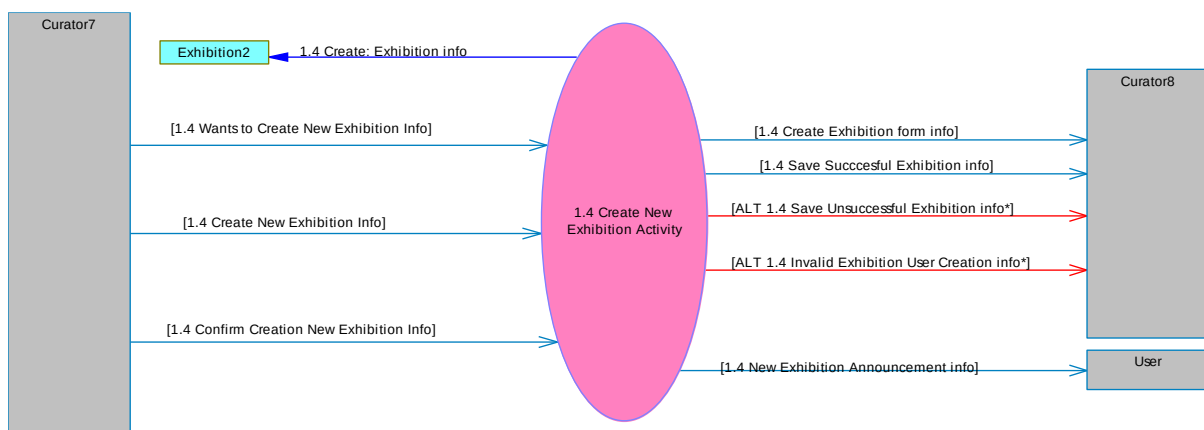


Figure 48: Middle Level DFD: 1.4 Create New Exhibition

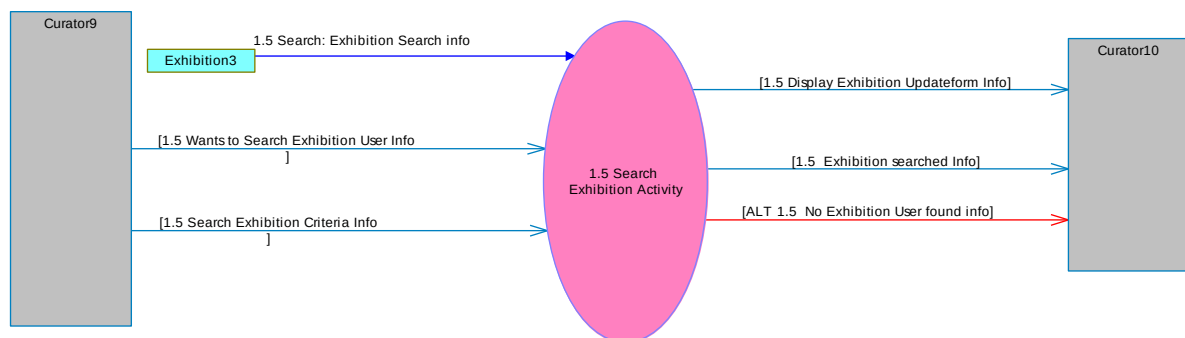


Figure 49: Middle Level DFD: 1.5 Search Exhibition

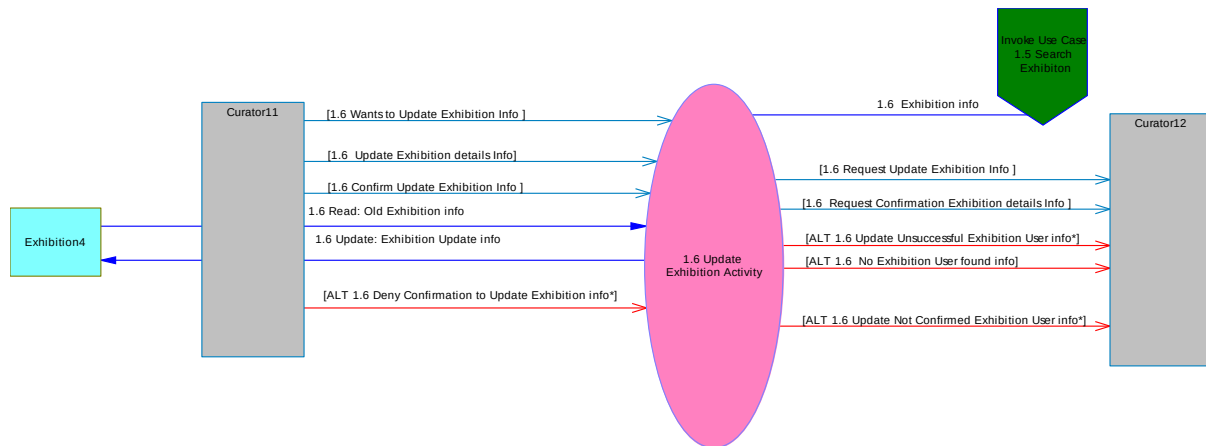


Figure 50: Middle Level DFD: Update Exhibition

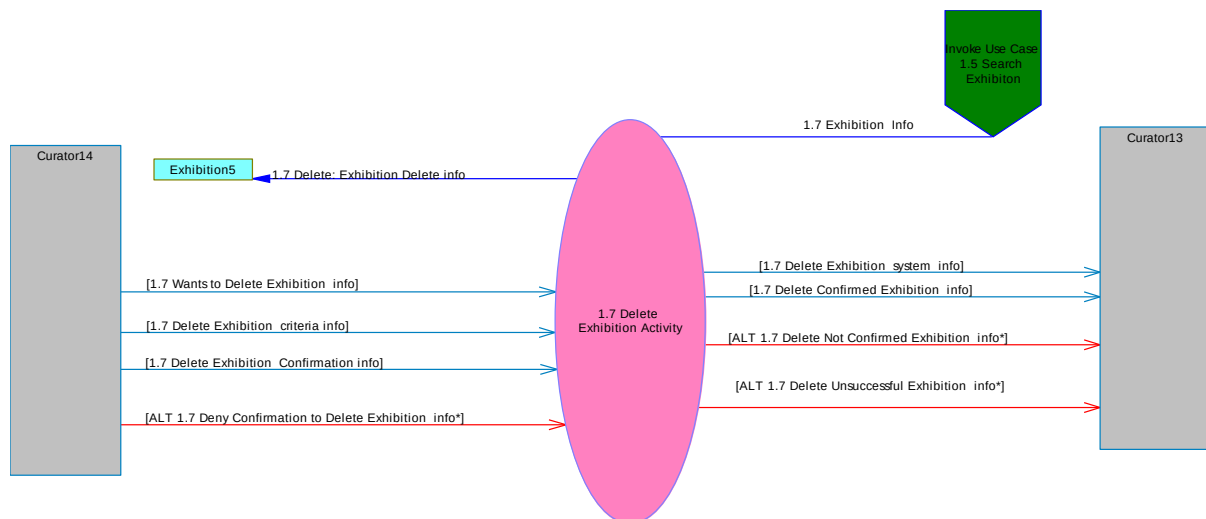


Figure 51: Middle Level DFD: 1.7 Delete Exhibition

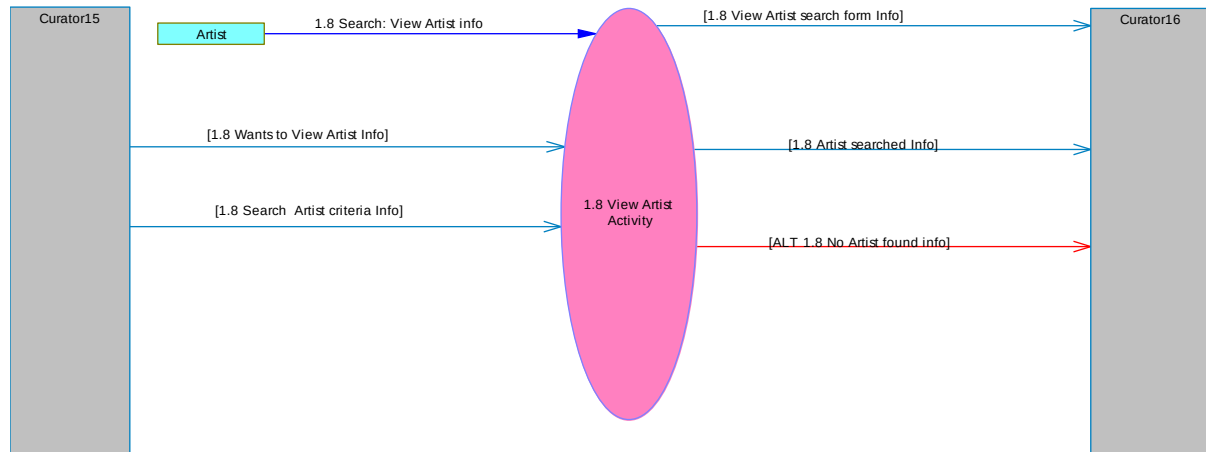


Figure 52: Middle Level DFD: 1.8 View Artist

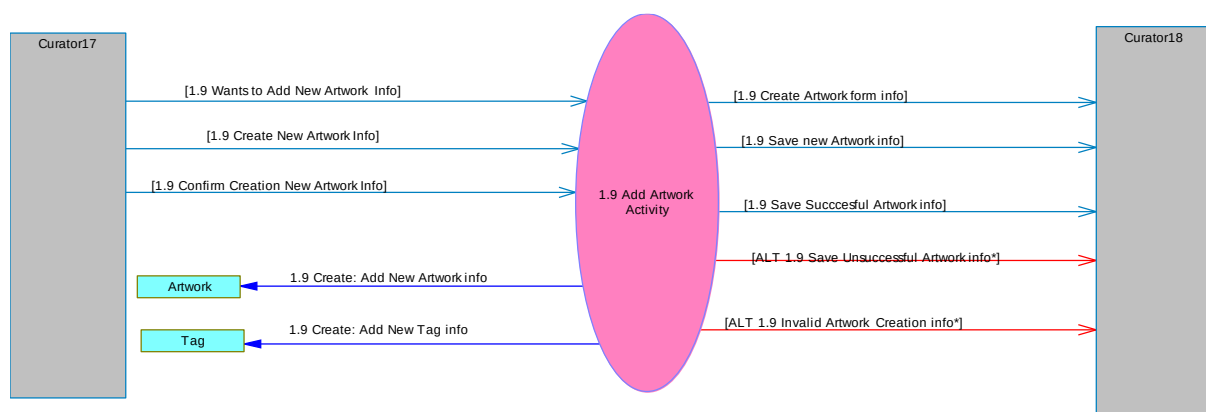


Figure 53: Middle Level DFD: 1.9 Add Artwork

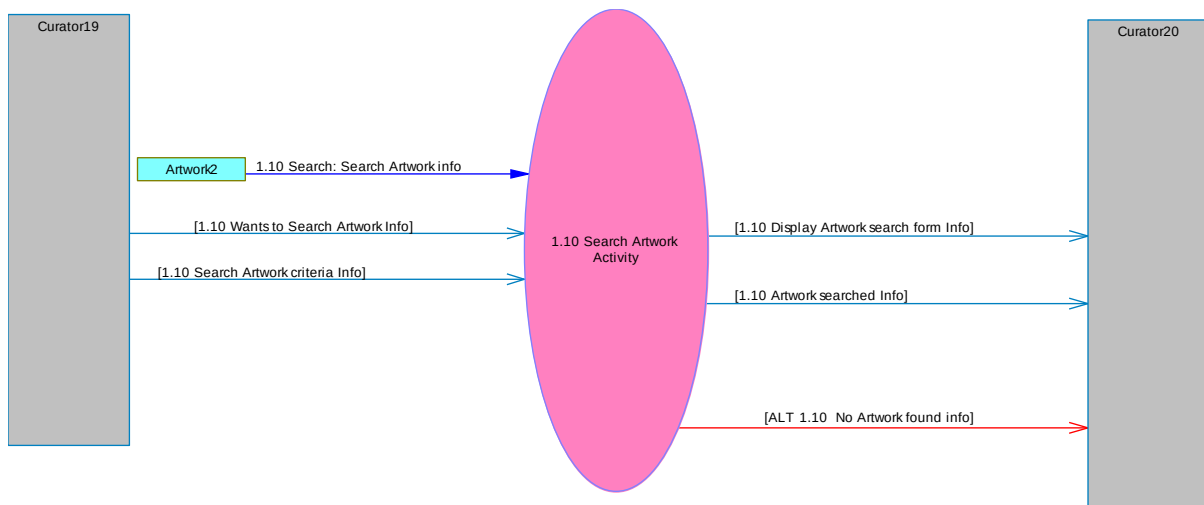


Figure 54: Middle Level DFD: 1.10 Search Artwork

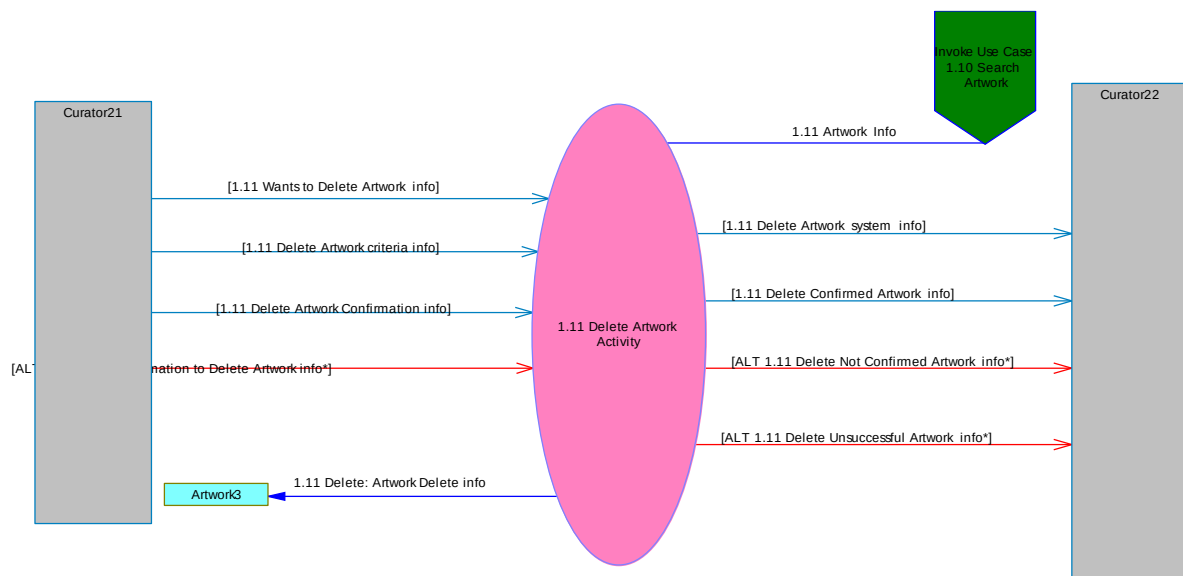


Figure 55: Middle Level DFD: 1.11 Delete Artwork

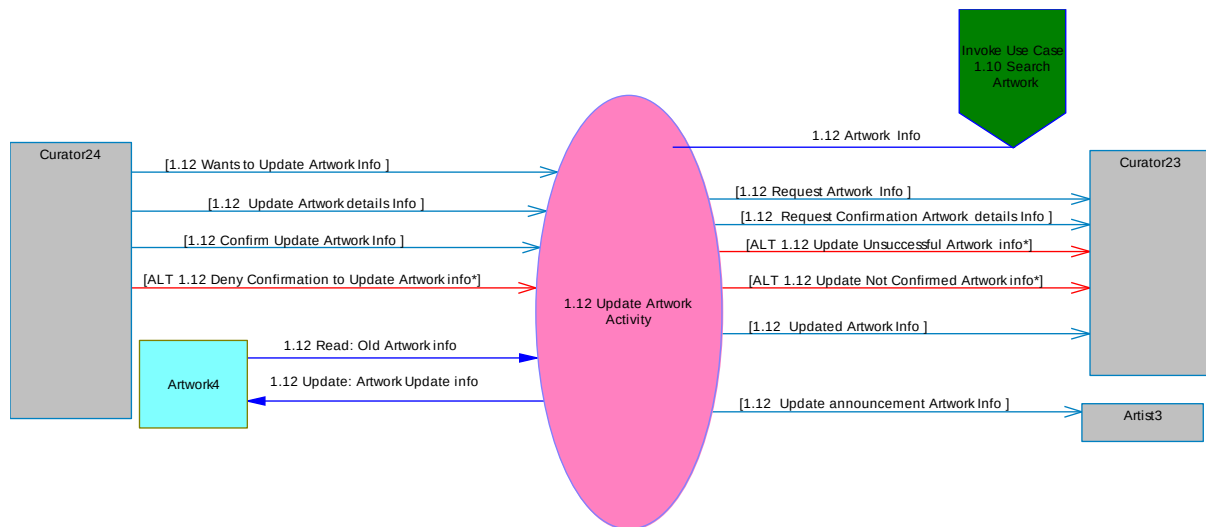


Figure 56: Middle Level DFD: 1.12 Update Artwork

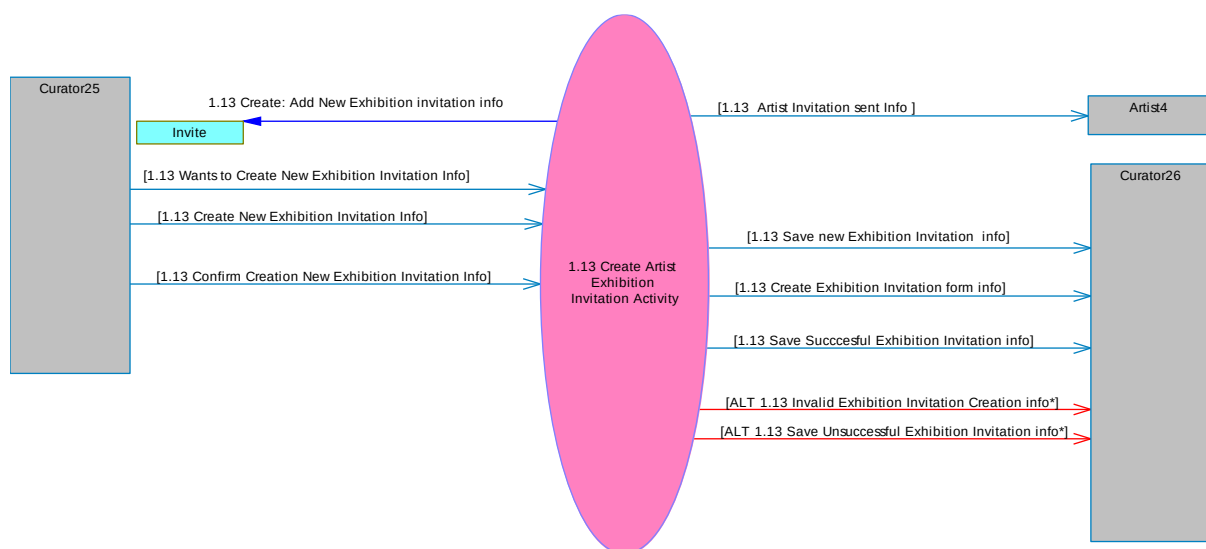


Figure 57: Middle Level DFD: 1.13 Create Artist Exhibition Invite

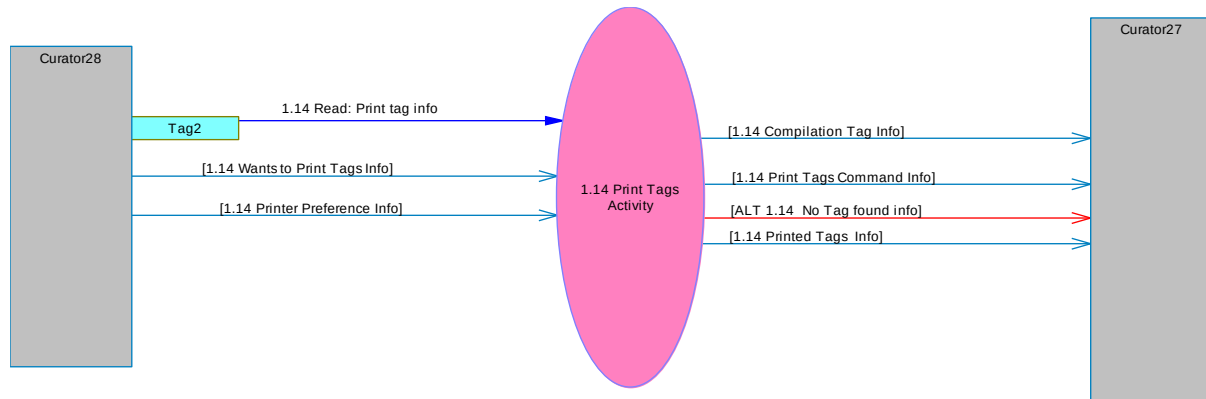


Figure 58: Middle Level DFD: 1.14 Print Tags

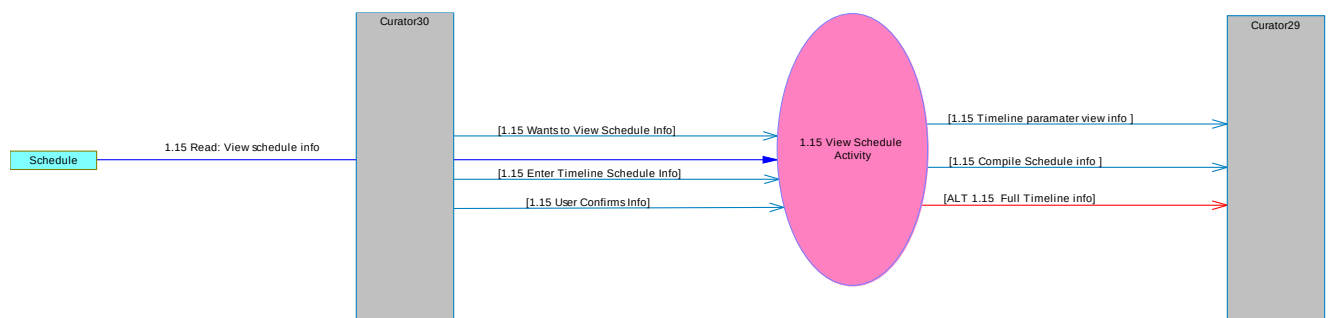


Figure 59: Middle Level DFD: 1.15 View Schedule

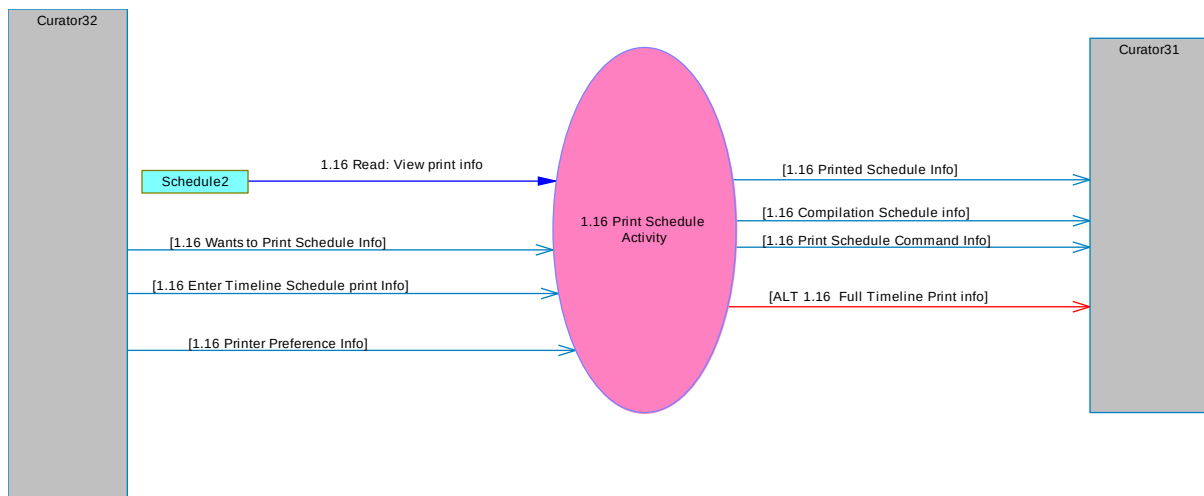


Figure 60: Middle Level DFD: 1.16 Print Schedule

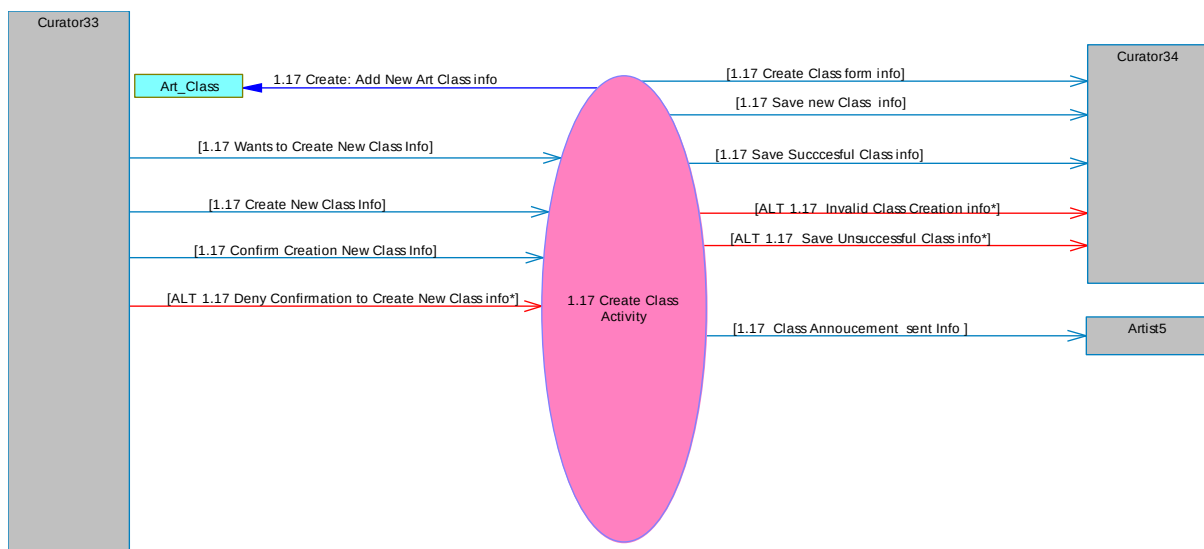


Figure 61: Middle Level DFD: 1.17 Create Class

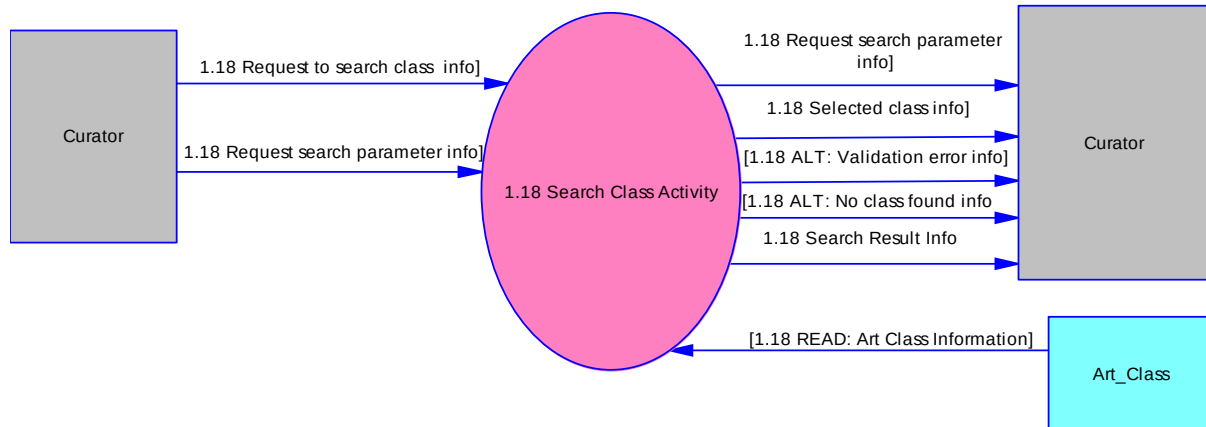


Figure 62: Middle Level DFD: 1.18 Search Class

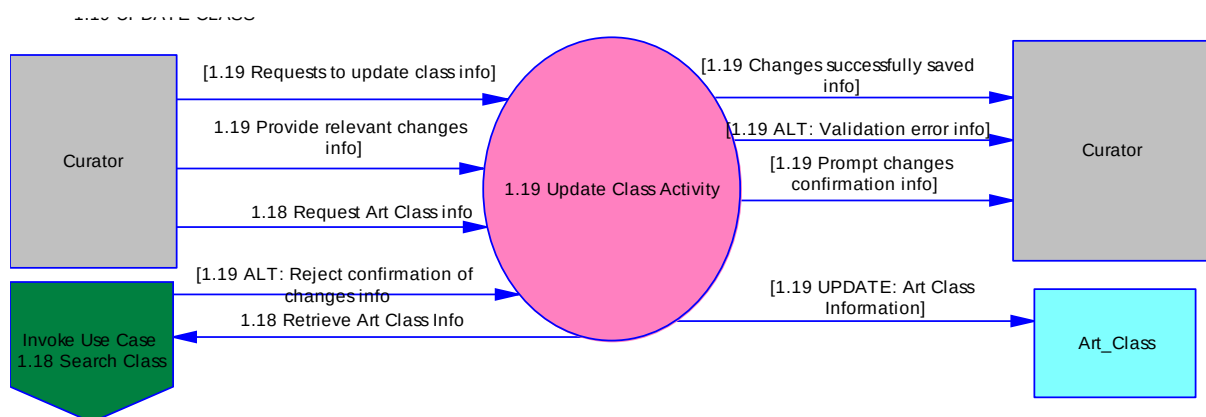


Figure 63: Middle Level DFD: 1.19 Update Class

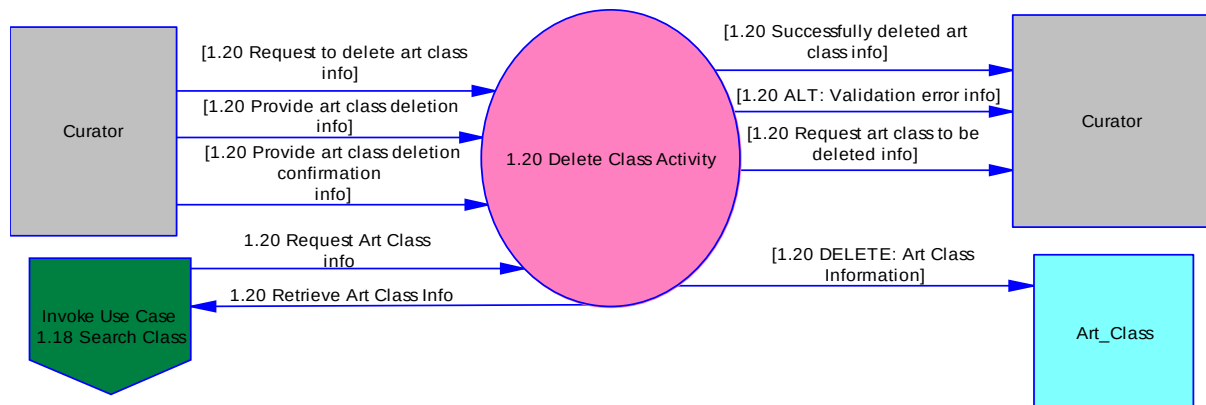


Figure 64: Middle Level DFD: 1.20 Delete Class

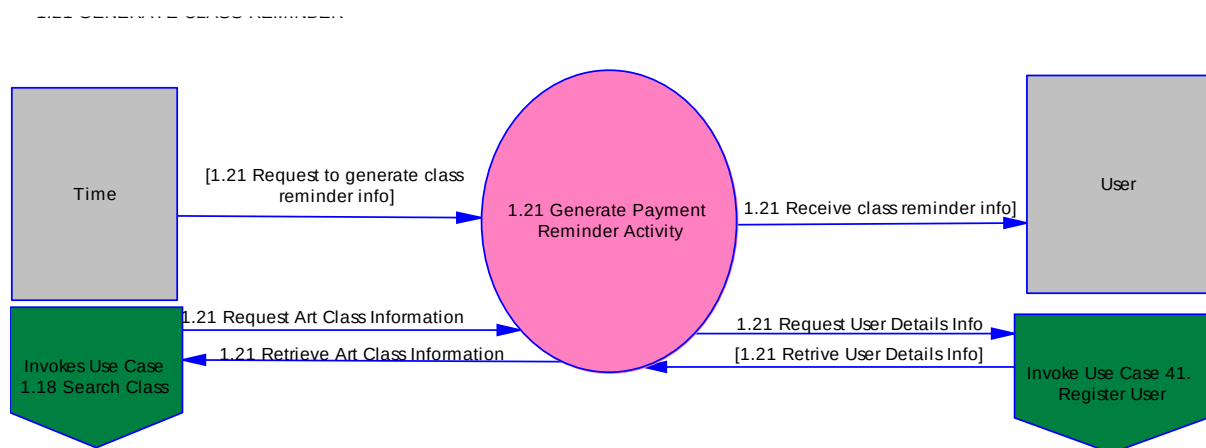


Figure 65: Middle Level DFD: 1.21 Generate Payment Reminder

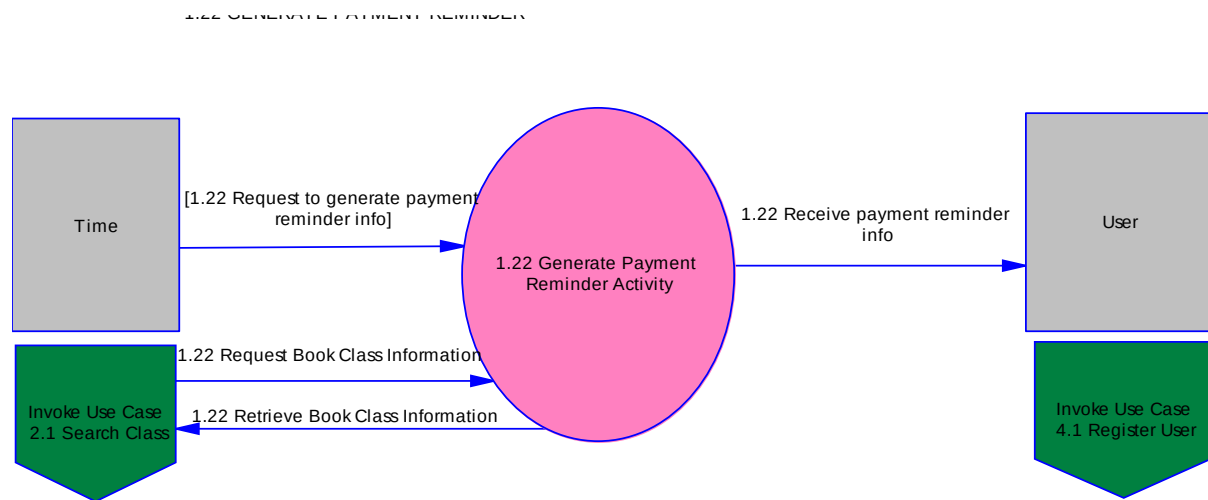


Figure 66: Middle Level DFD: 1.22 Generate Payment Reminder

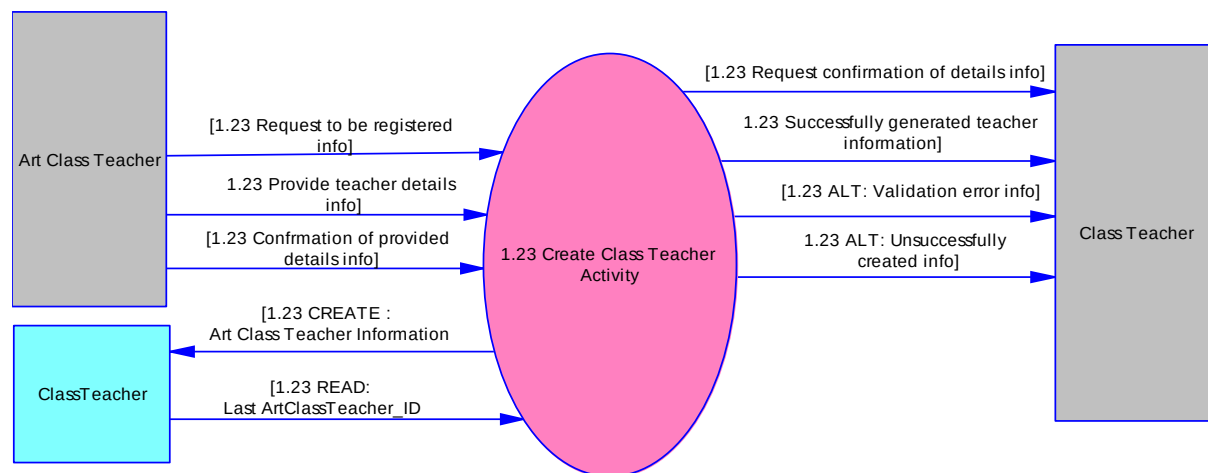


Figure 67: Middle Level DFD: 1.23 Create Class Teacher

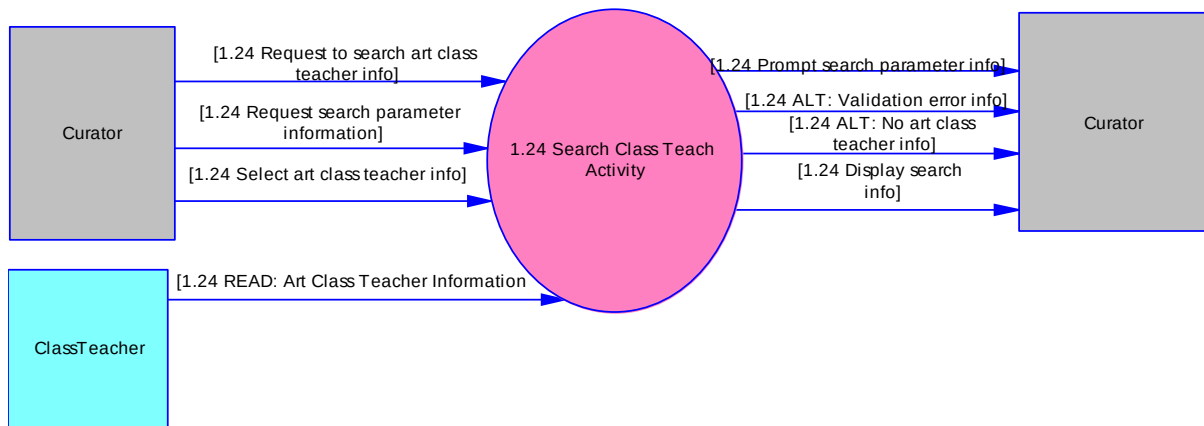


Figure 68: Middle Level DFD: 1.24 Search Class Teacher

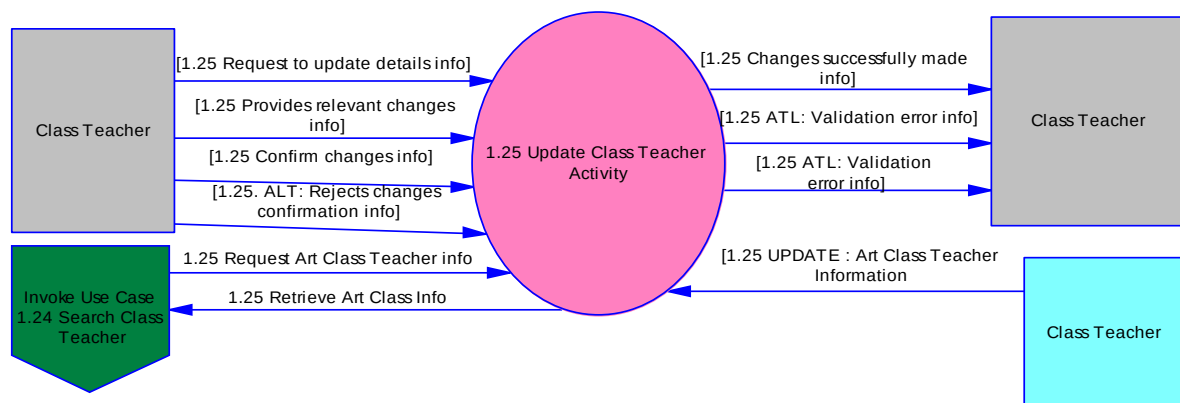


Figure 69: Middle Level DFD: 1.25 Update Class Teacher

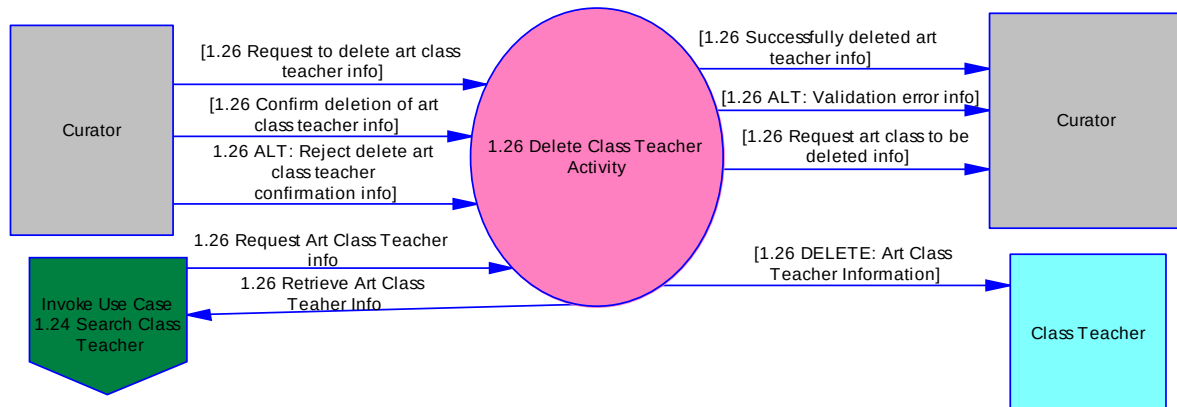


Figure 70: Middle Level DFD: 1.26 Delete Class Teacher

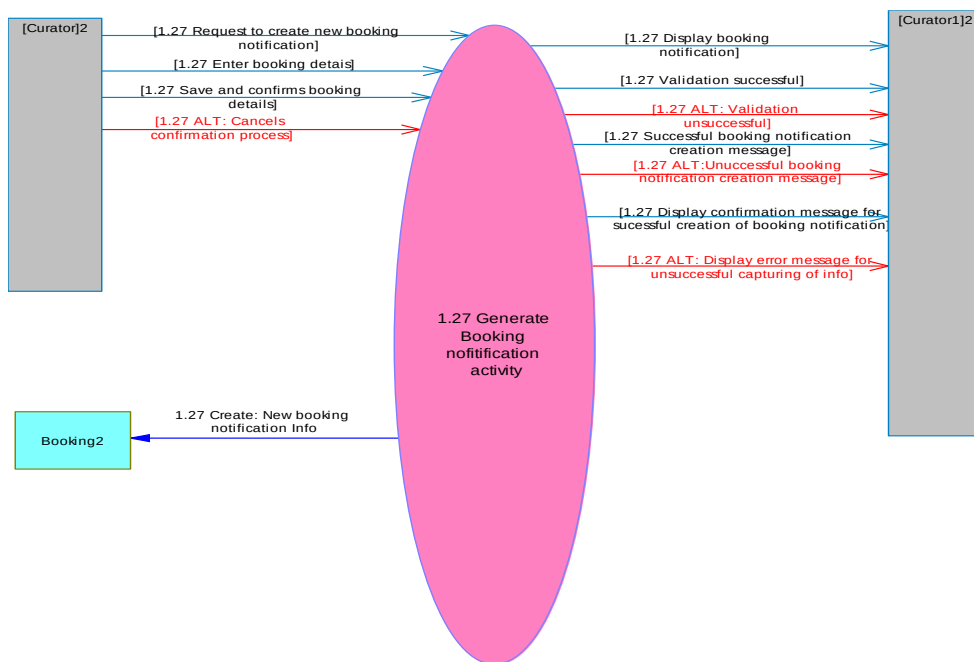


Figure 71: Middle Level DFD: 1.27 Generate Booking Notification

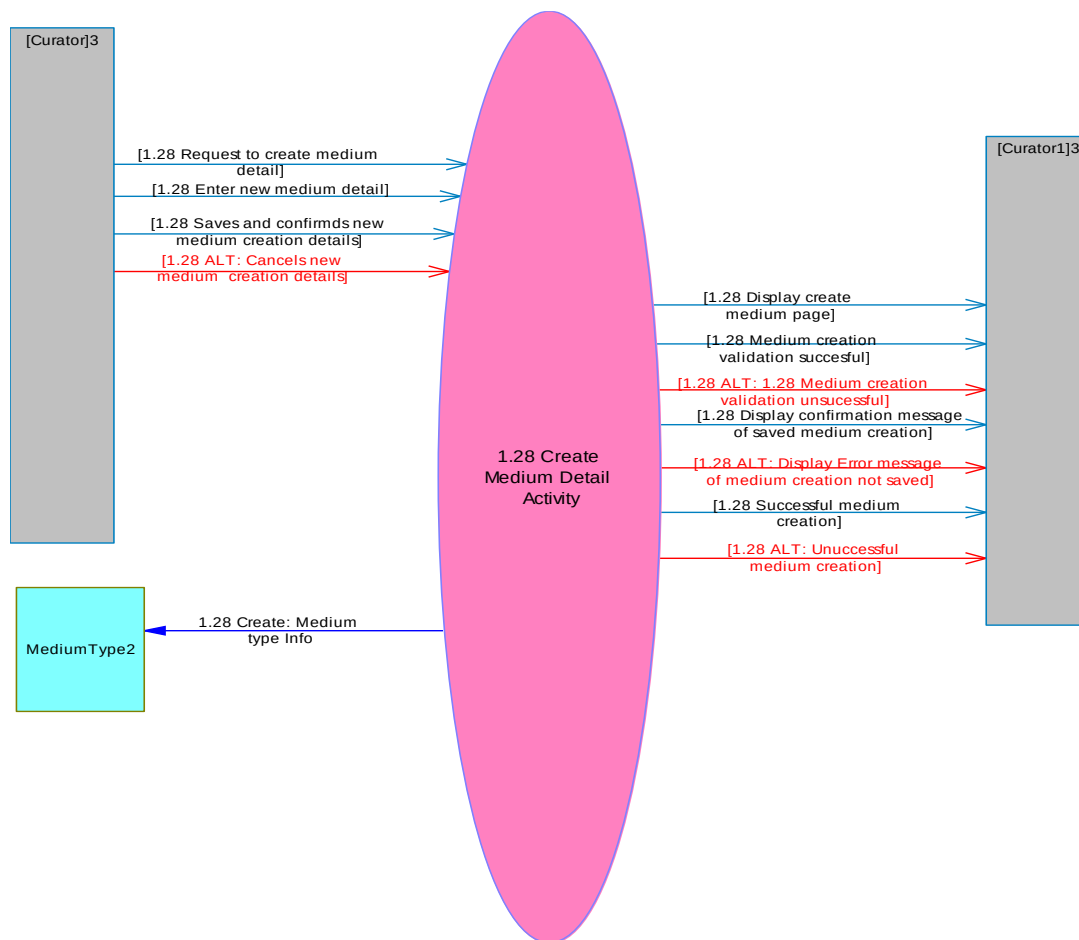


Figure 72: Middle Level DFD: 1.28 Create Medium Detail

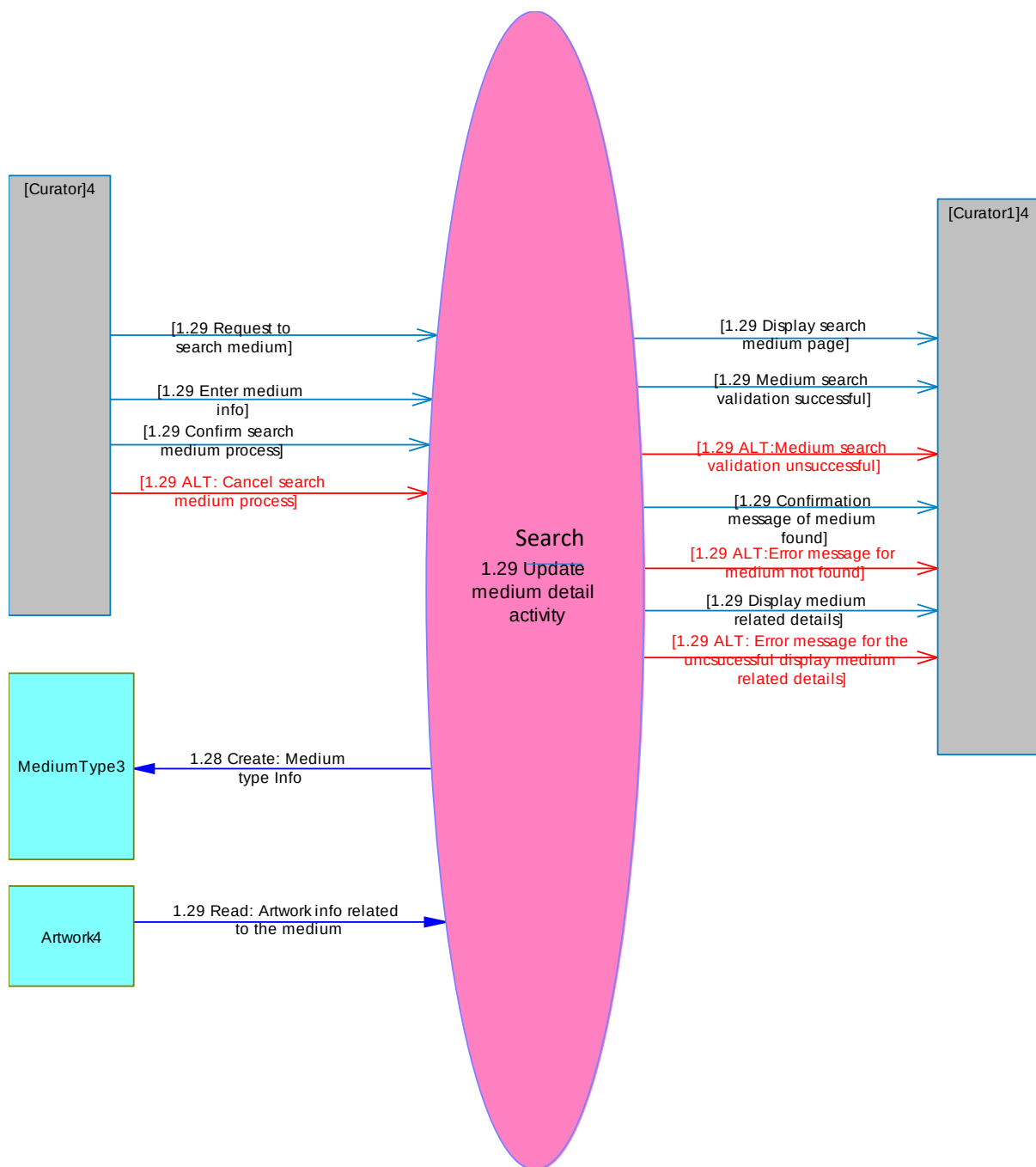


Figure 73: Middle Level DFD: 1.29 Search Medium Detail

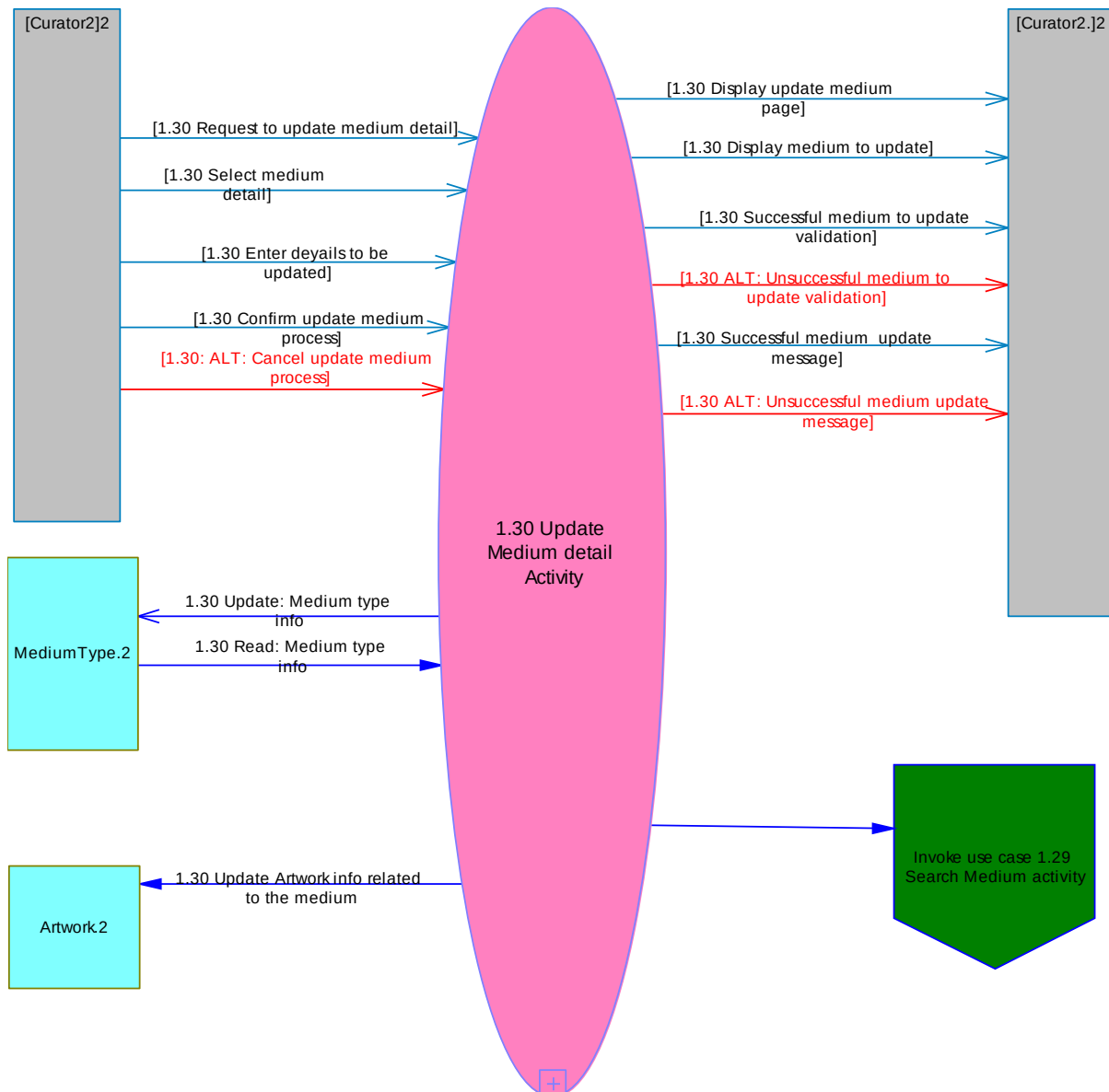


Figure 74: Middle Level DFD: 1.30 Update Medium Detail

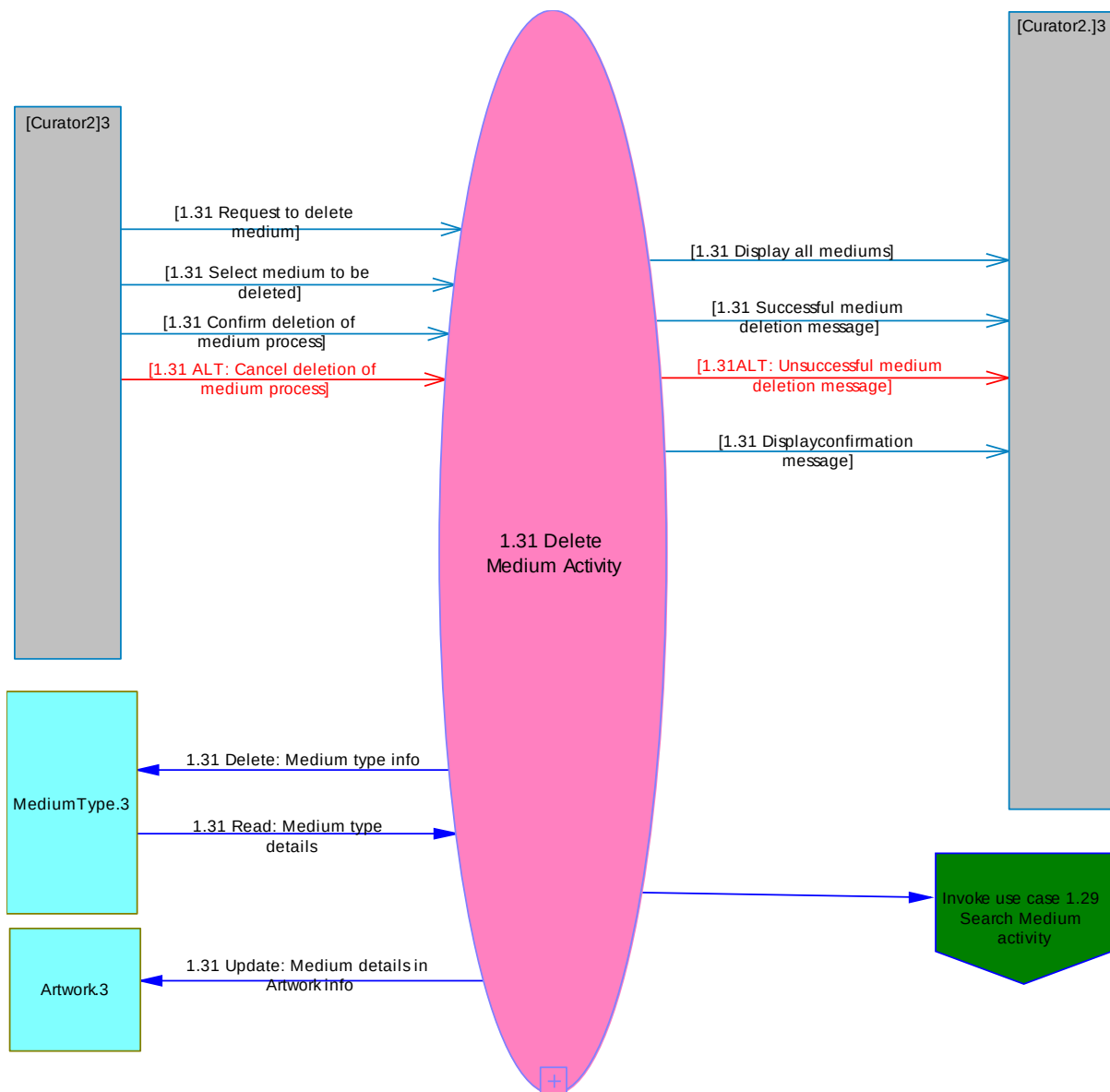


Figure 75: Middle Level DFD: 1.31 Delete Medium

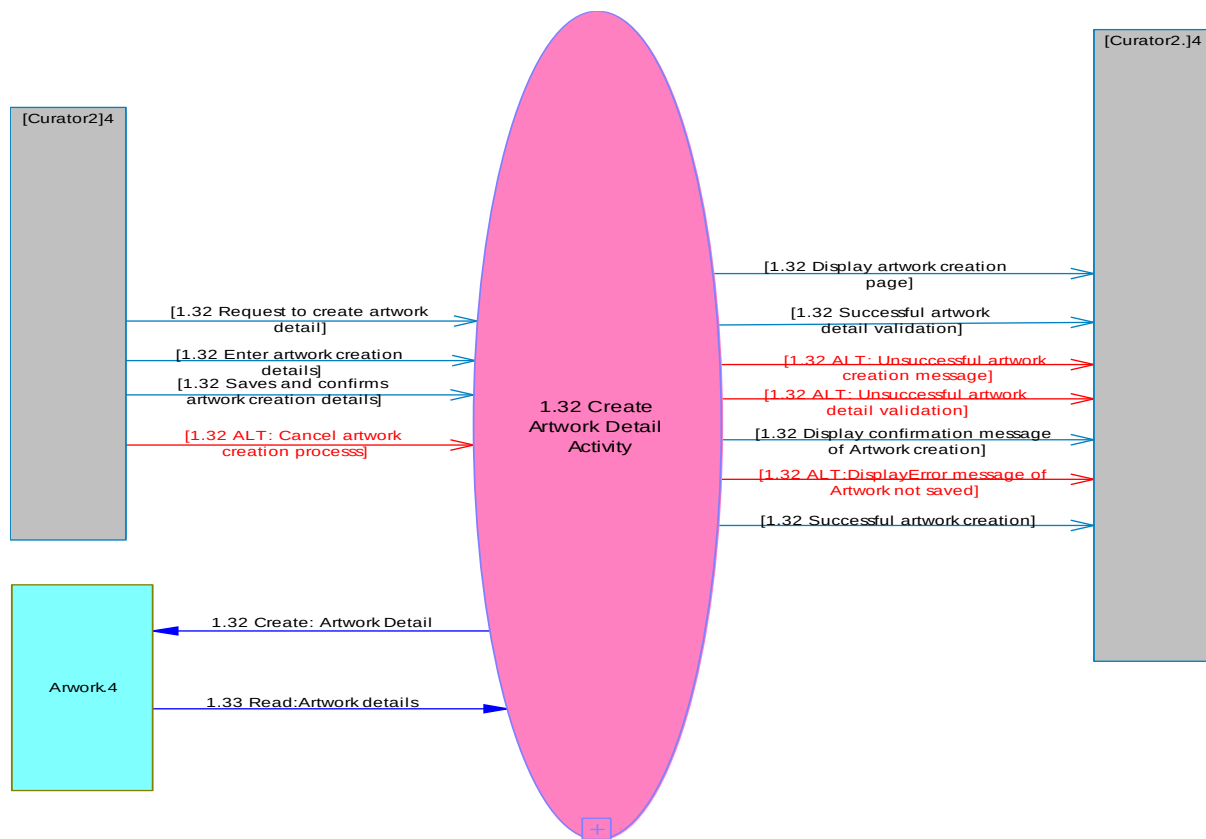


Figure 76: Middle Level DFD: 1.32 Create Artwork Detail

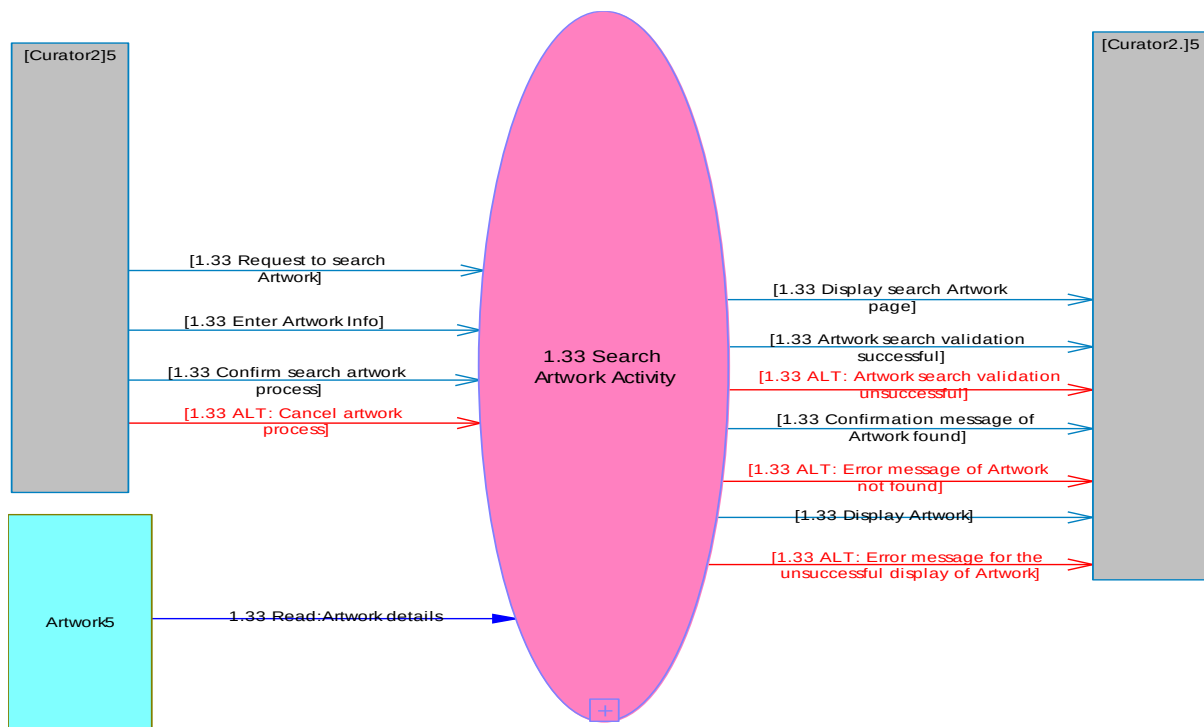


Figure 77: Middle Level DFD: 1.33 Search Artwork

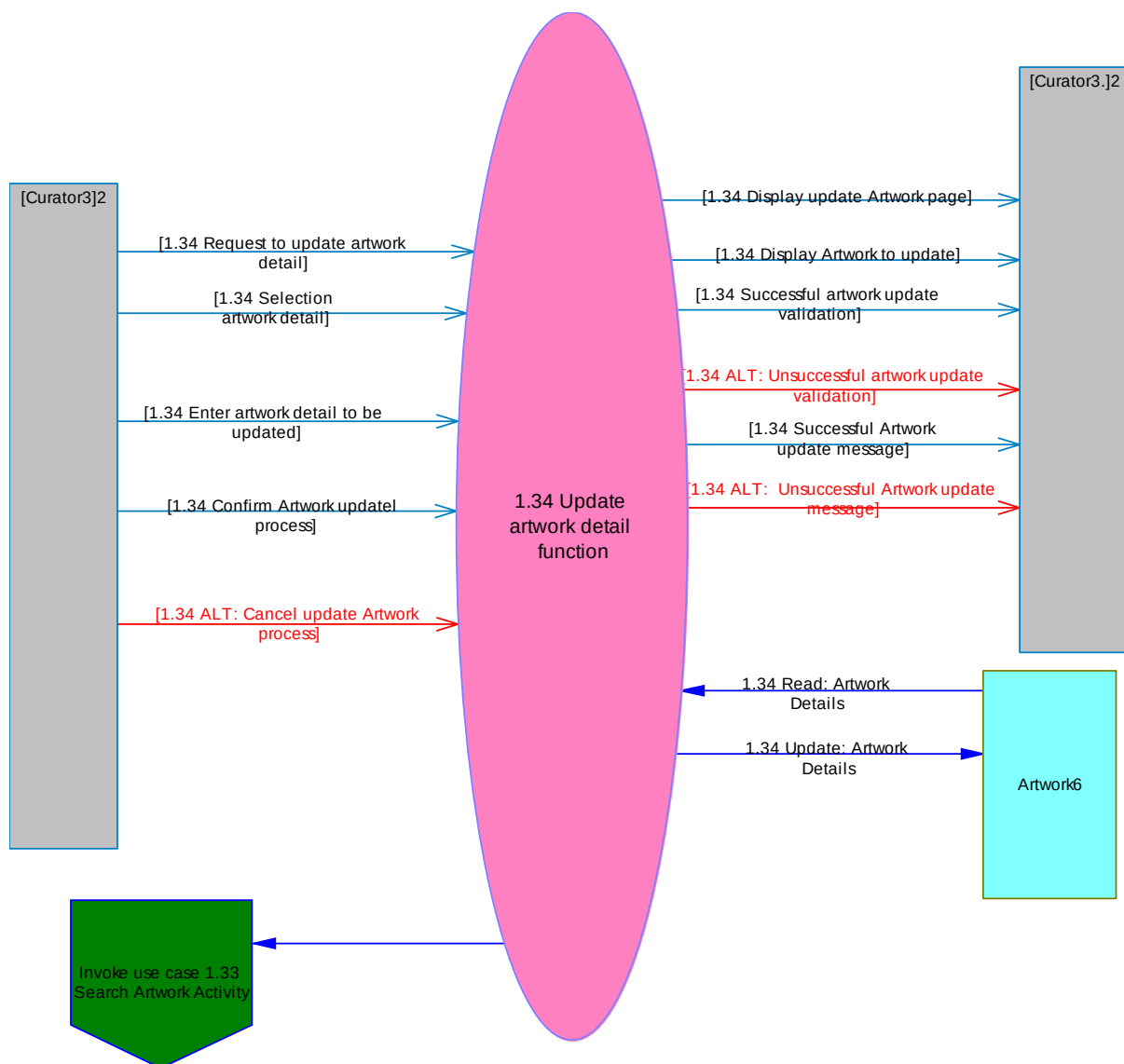


Figure 78: Middle Level DFD: 1.34 Update Artwork Detail

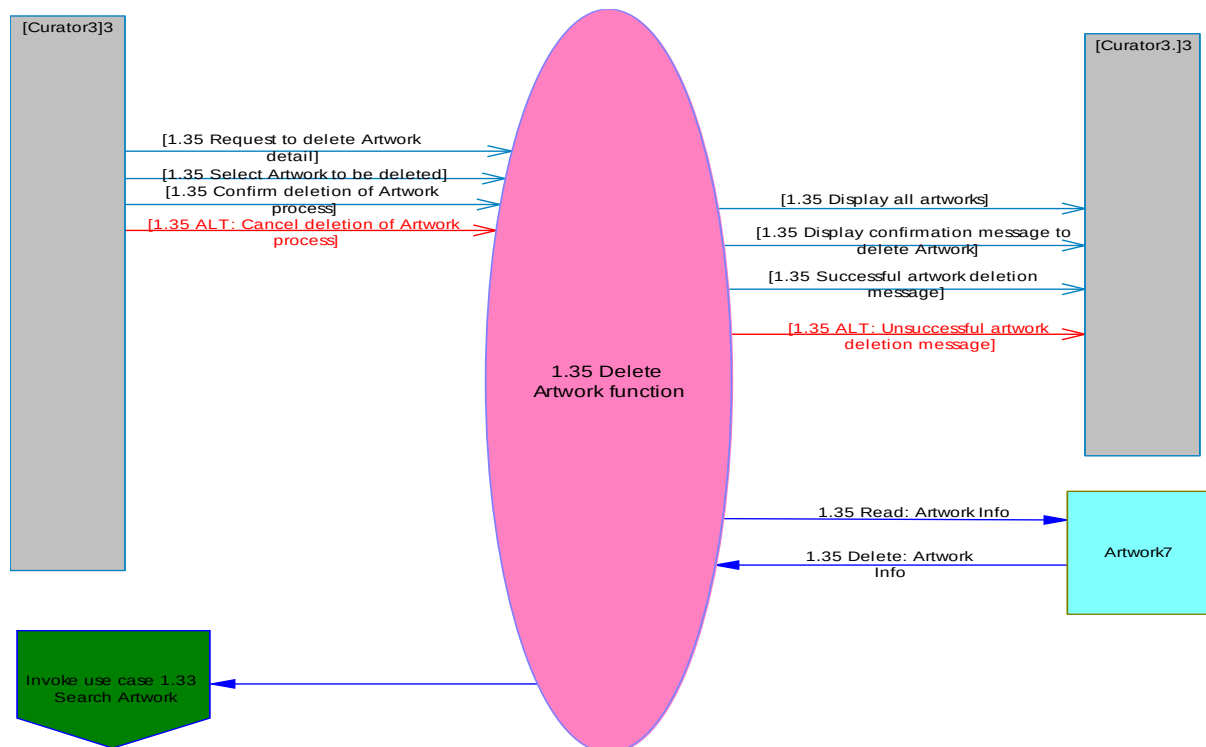


Figure 79: Middle Level DFD: 1.35 Delete Artwork

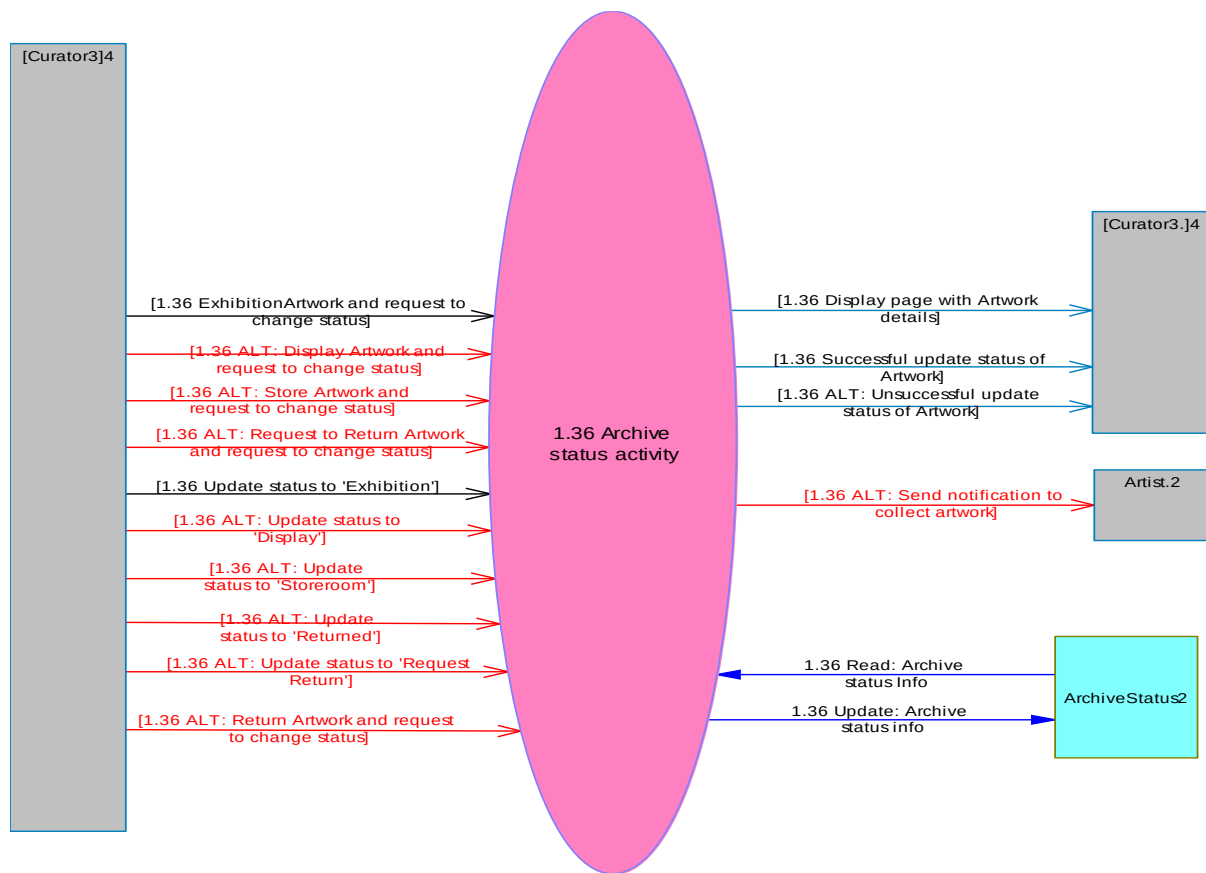


Figure 80: Middle Level DFD: 1.36 Archive Status

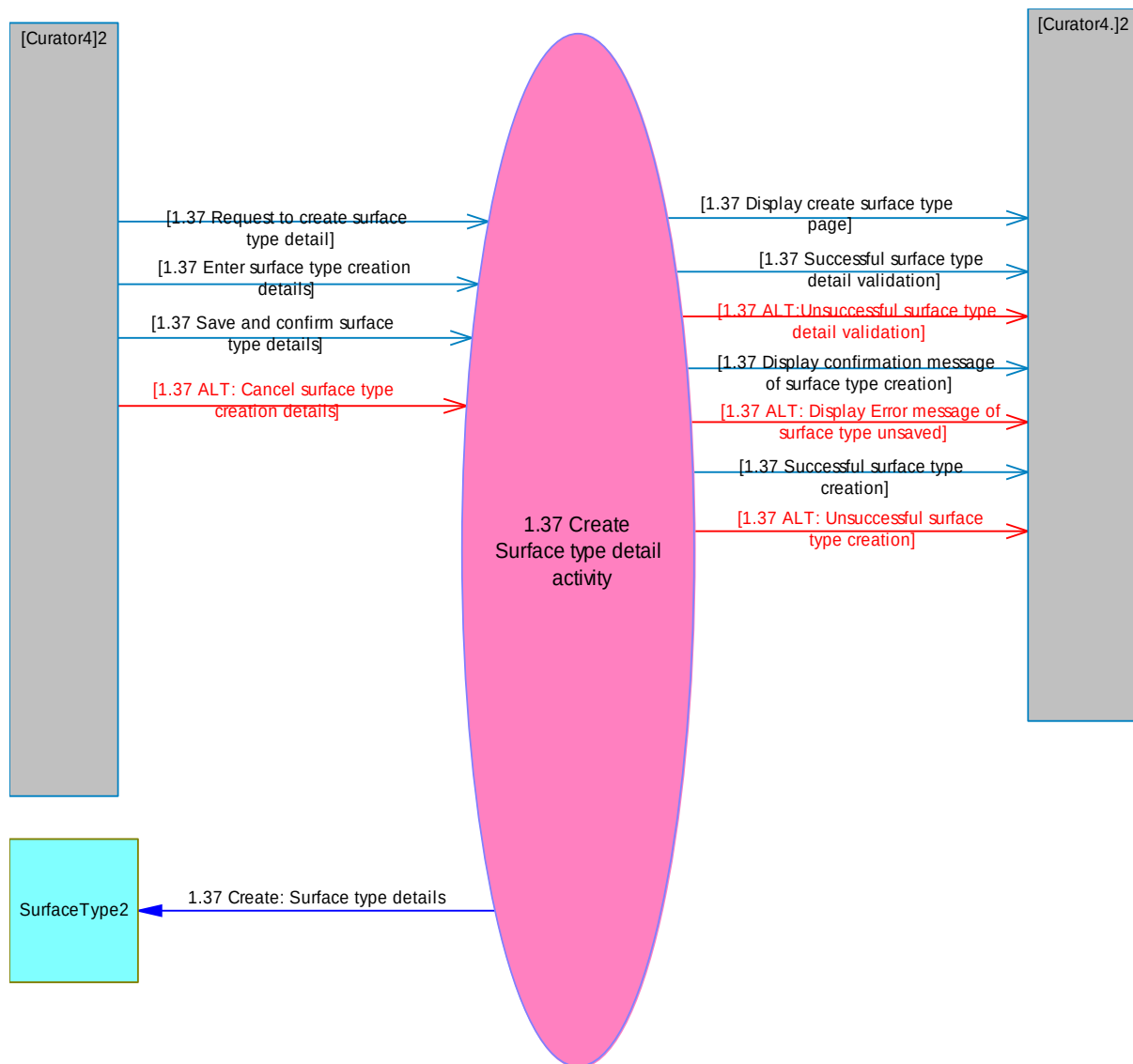


Figure 81: Middle Level DFD: 1.37 Create Surface type Detail

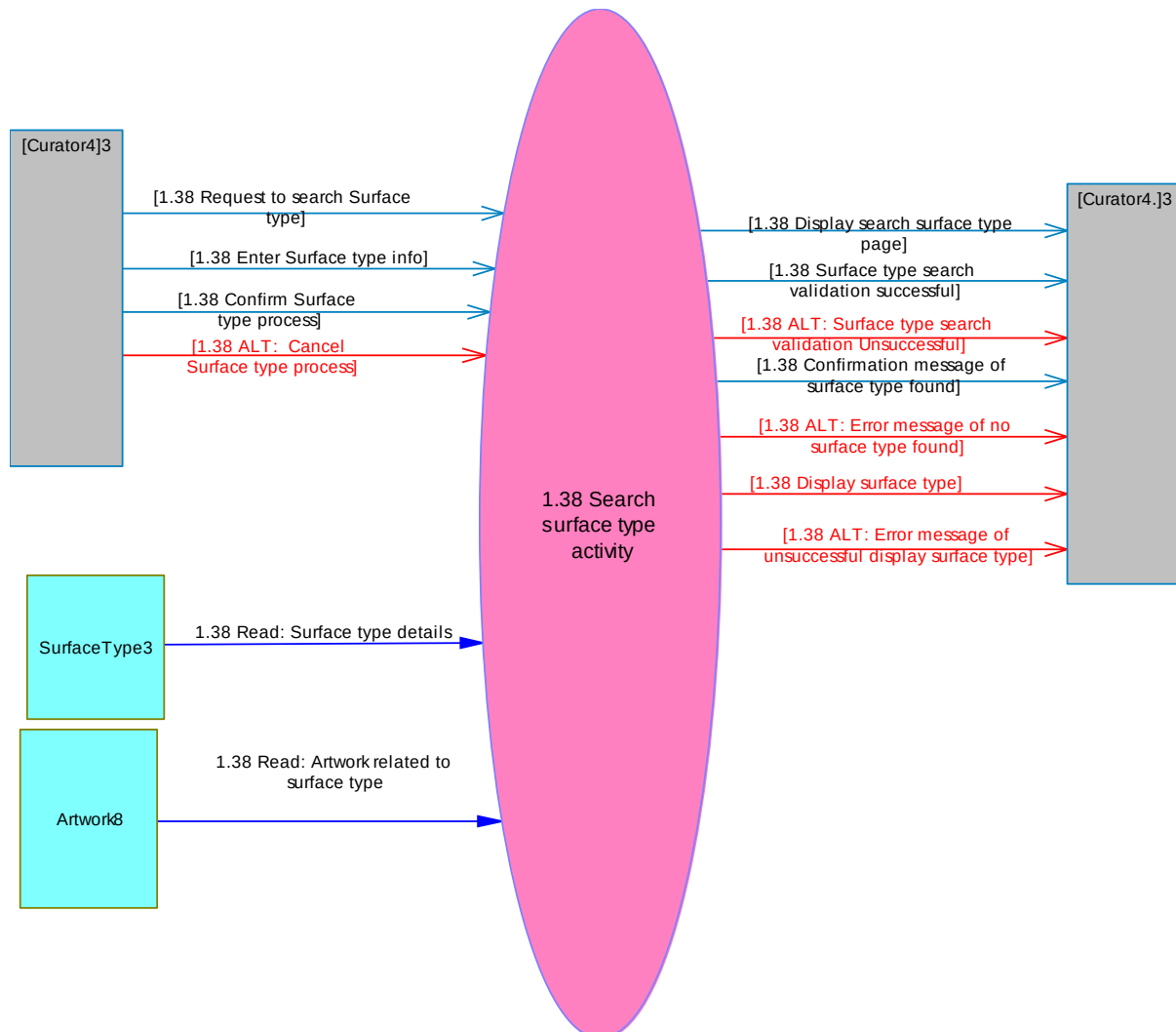


Figure 82: Middle Level DFD: 1.38 Search Surface Type

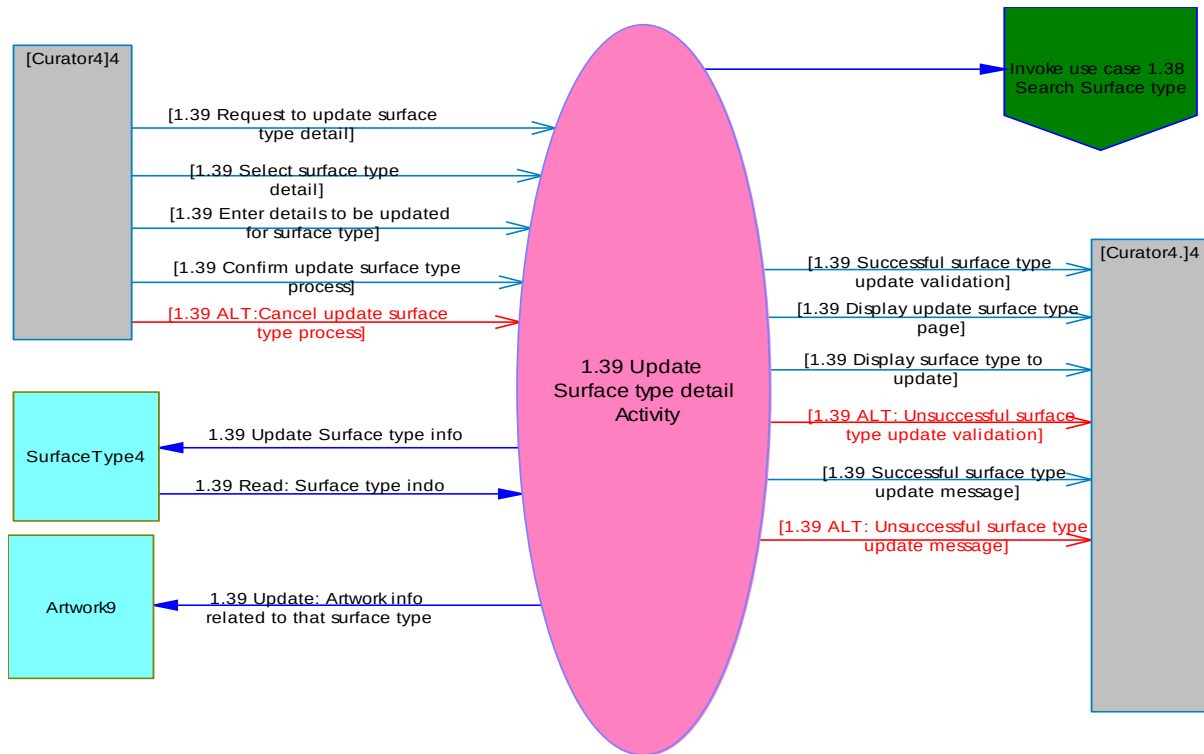


Figure 83: Middle Level DFD: 1.39 Update Surface Type detail

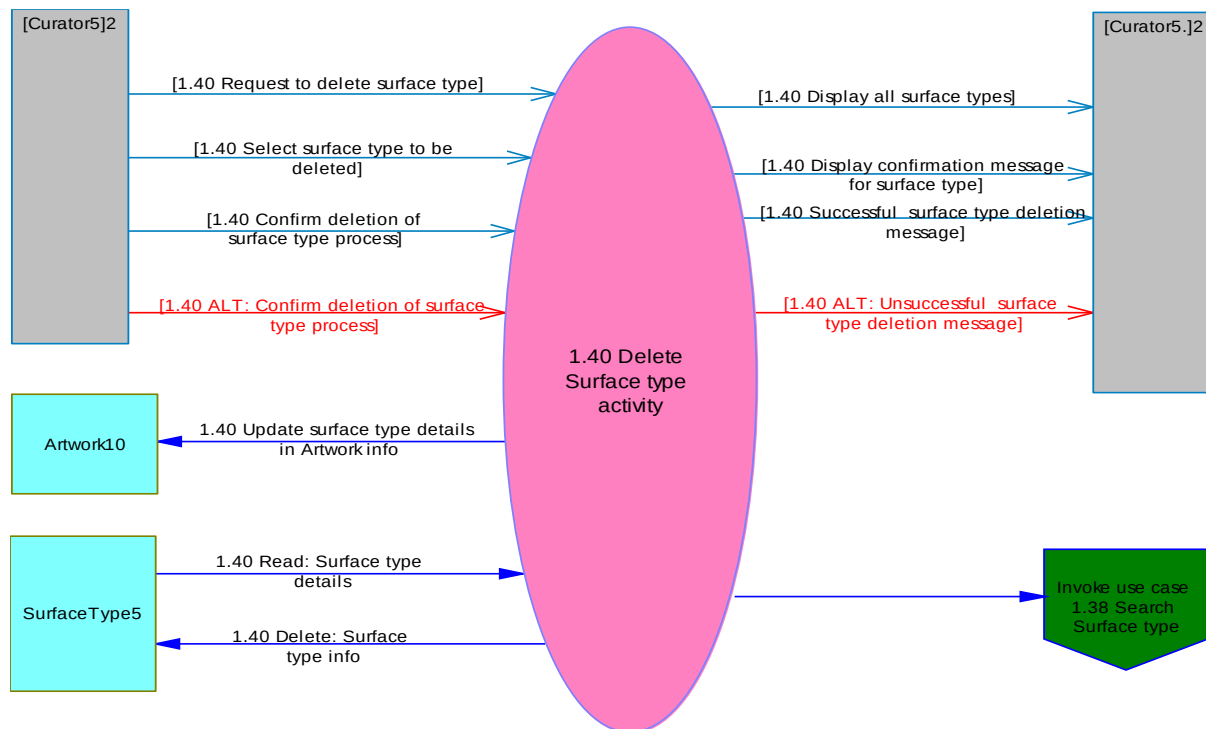


Figure 84: Middle Level DFD: 1.40 Delete Surface type

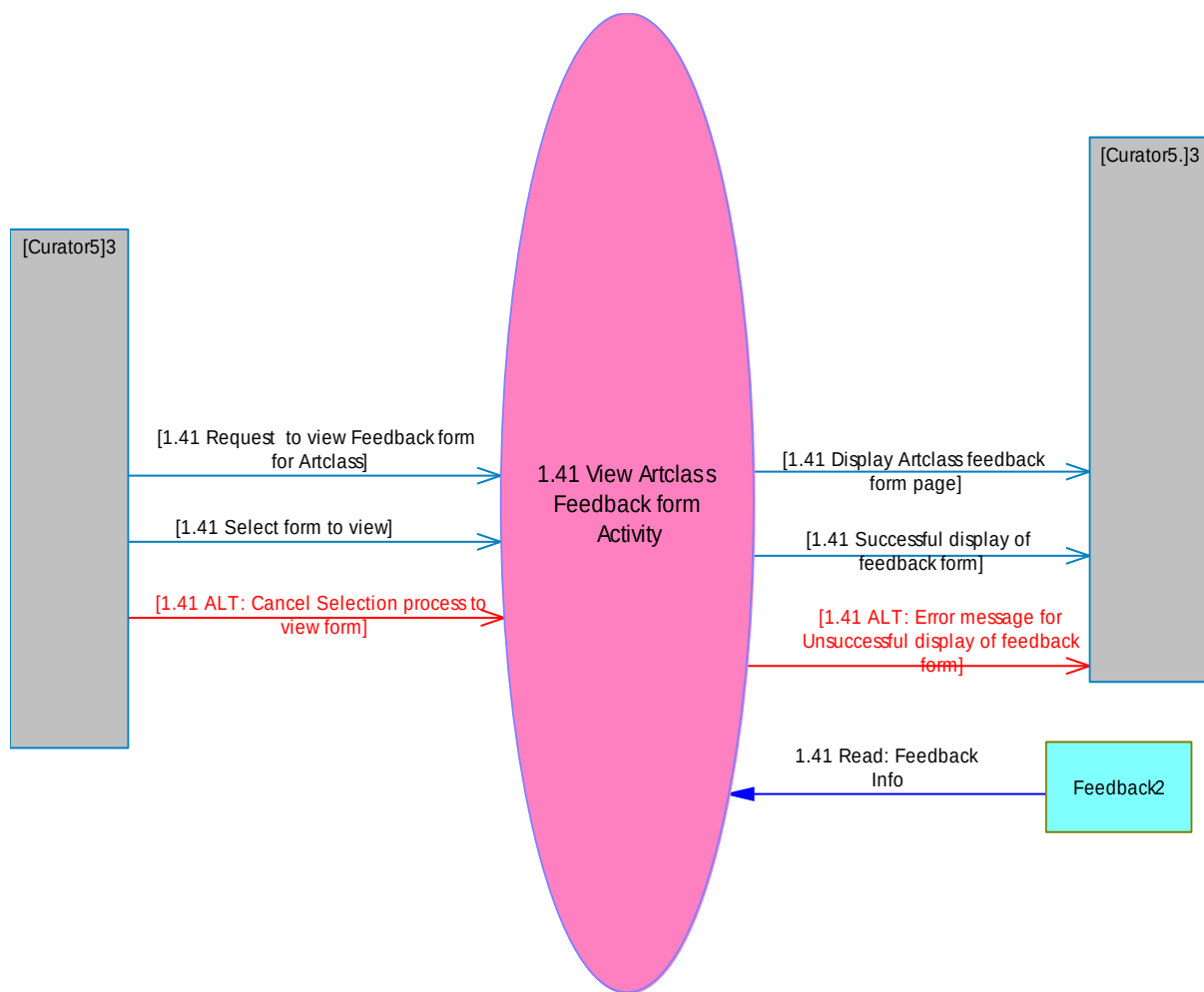


Figure 85: Middle Level DFD: 1.41 View Art Class Feedback Form

3.3.2.2 Art Class Subsystem

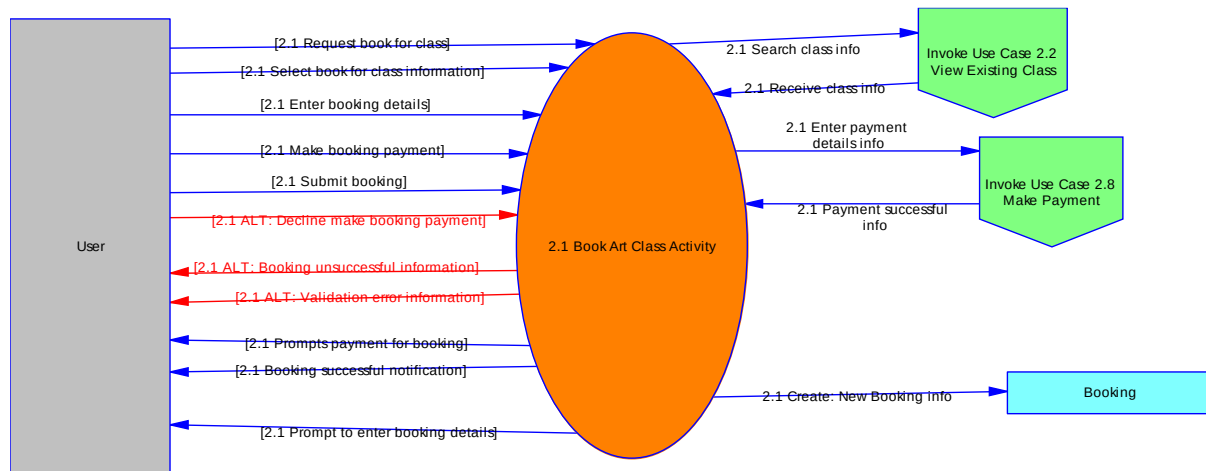


Figure 86: Middle Level DFD: 2.1 Book Art Class

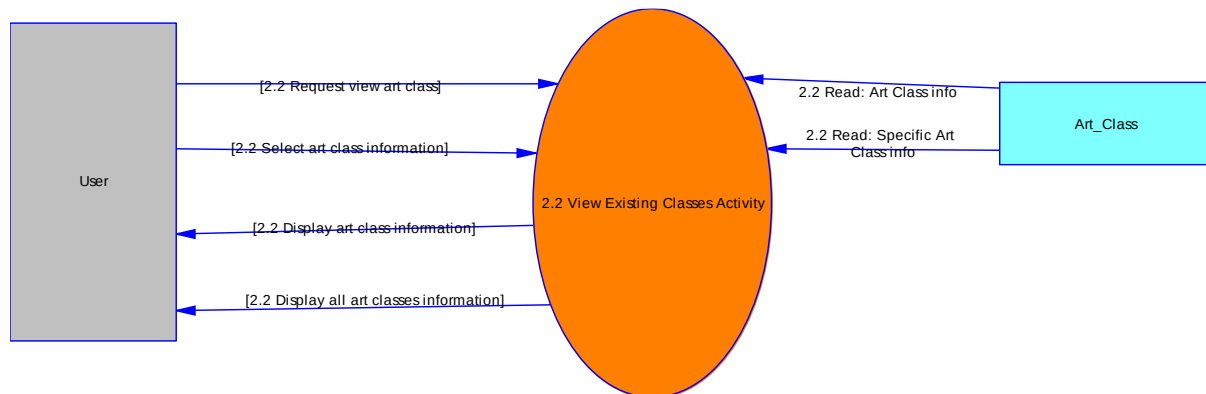


Figure 87: Middle Level DFD: 2.2 View Existing Classes

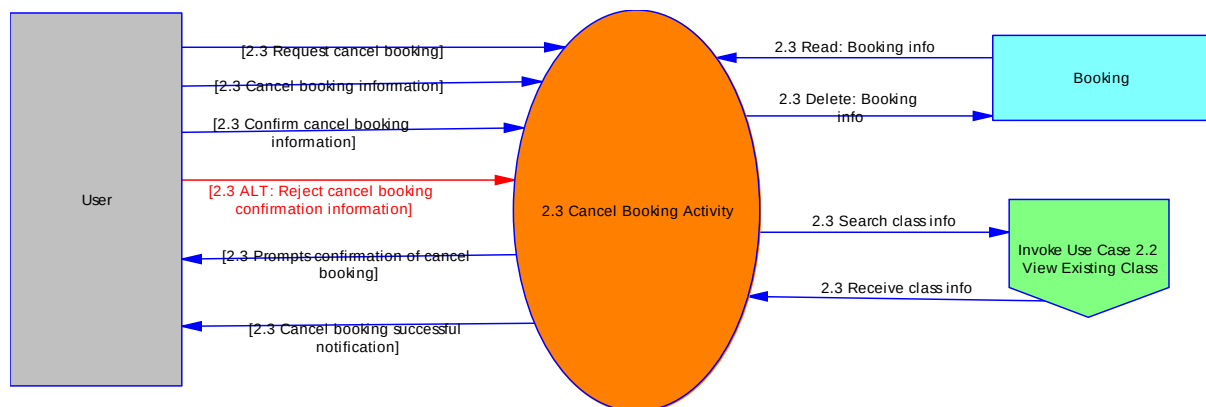


Figure 88: Middle Level DFD: 2.3 Cancel Booking

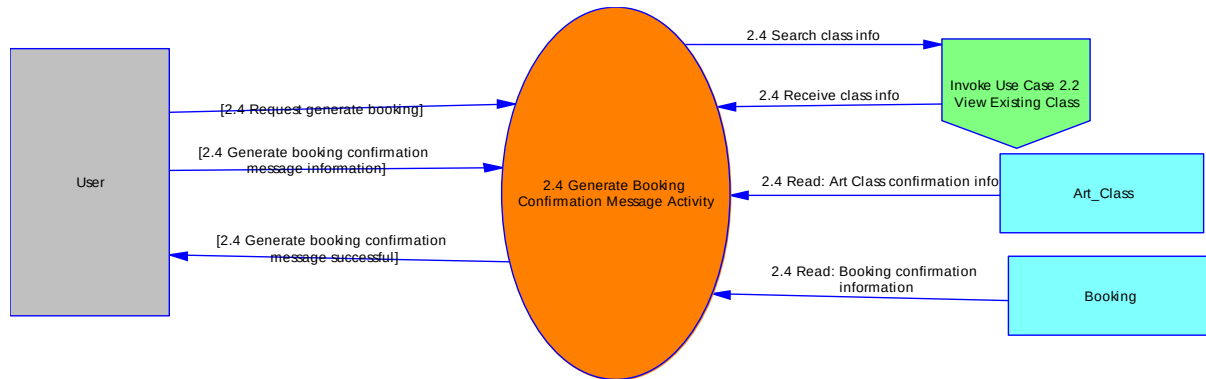


Figure 89: Middle Level DFD: 2.4 Generate Booking Confirmation Message

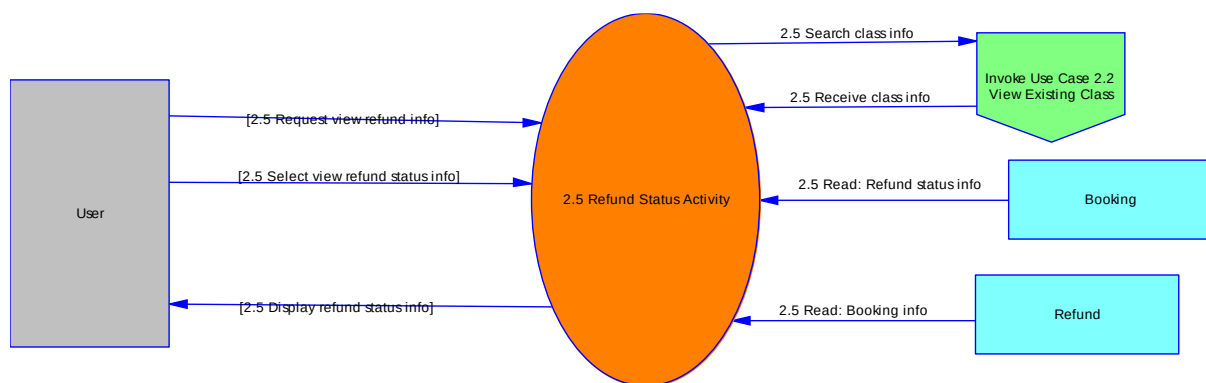


Figure 90: Middle Level DFD: 2.5 Refund Status

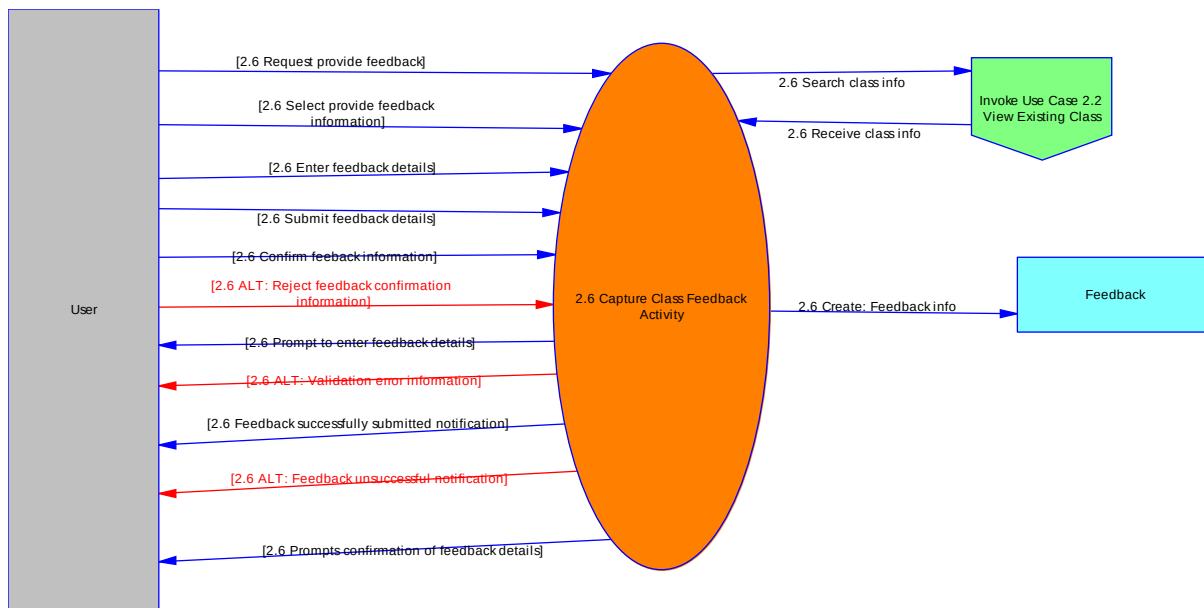


Figure 91: Middle Level DFD: 2.6 Capture Class Feedback

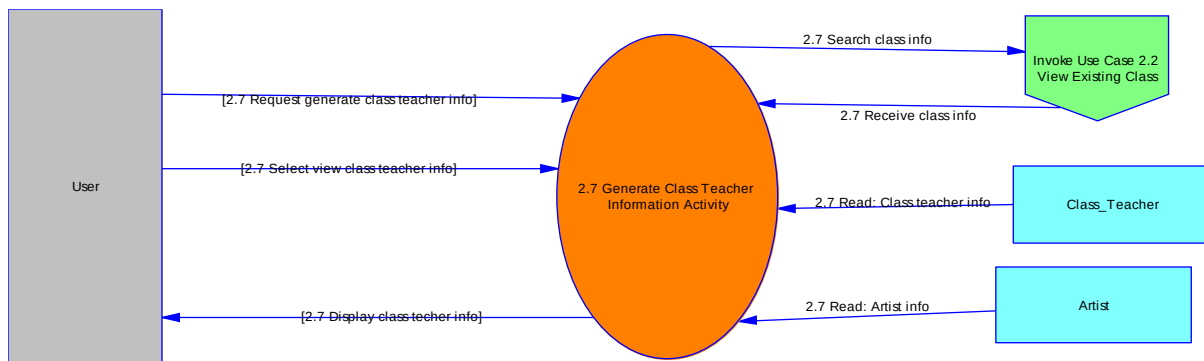


Figure 92: Middle Level DFD: 2.7 Generate Class Teacher Information

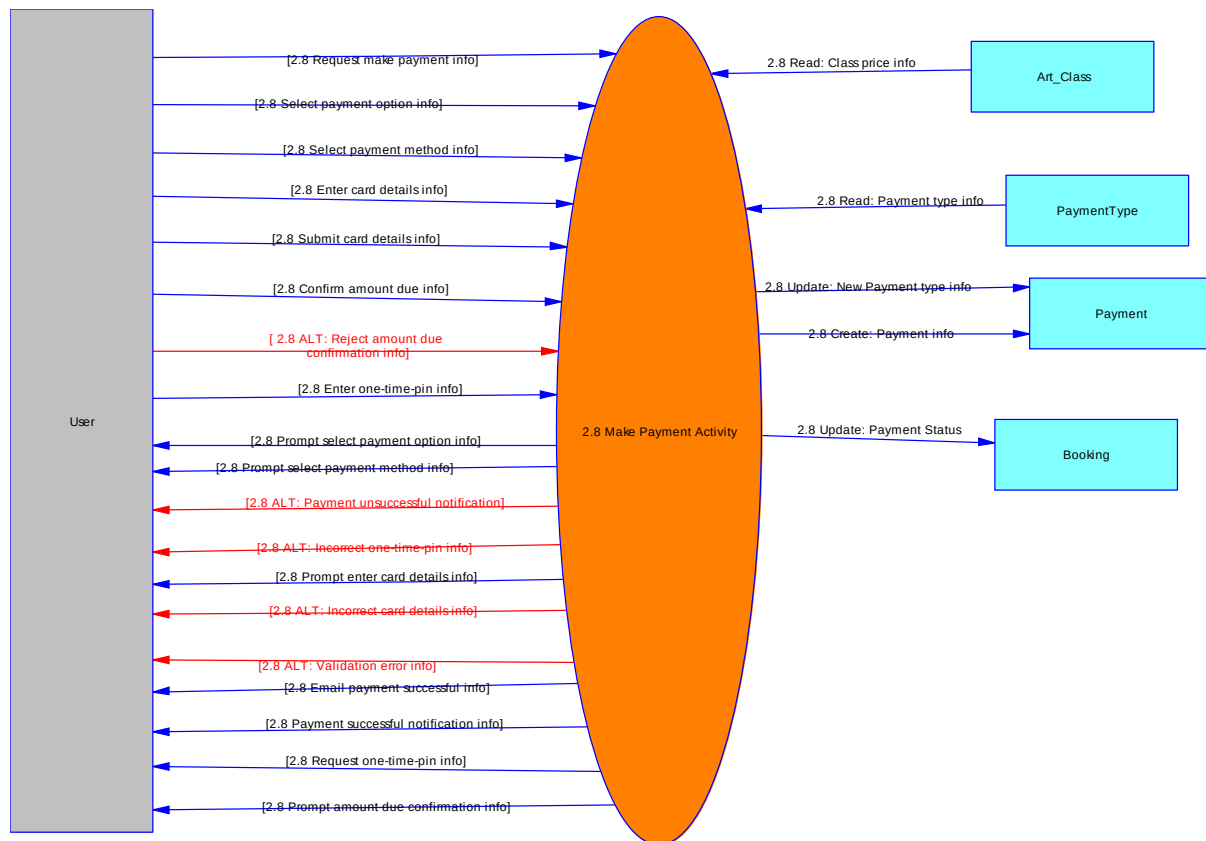


Figure 93: Middle Level DFD: 2.8 Make Payment

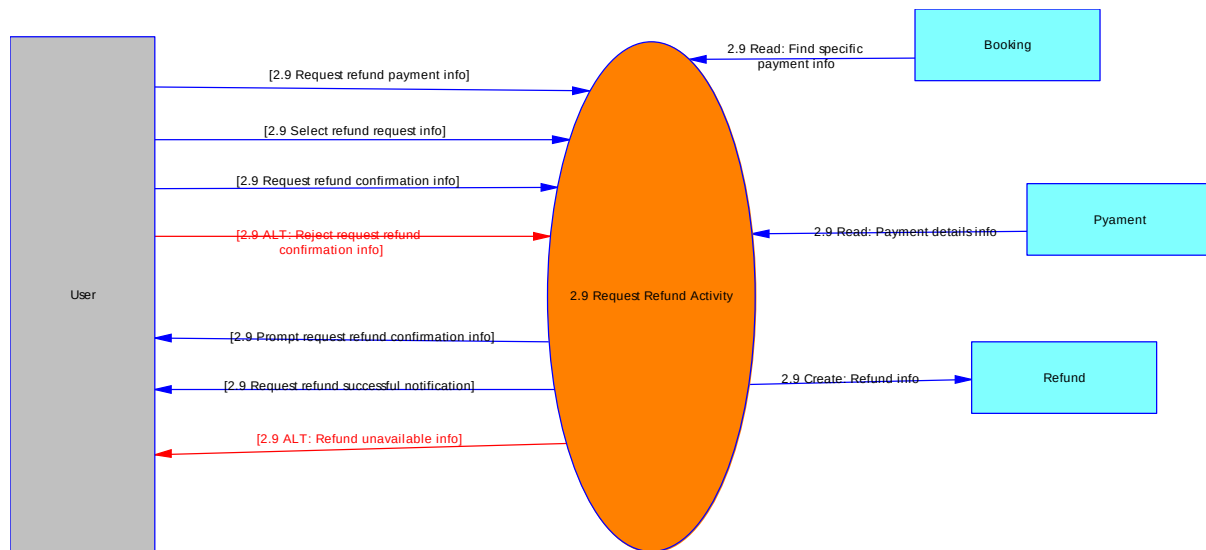


Figure 94: Middle Level DFD: 2.9 Request Refund

3.3.2.3 Reporting Subsystem

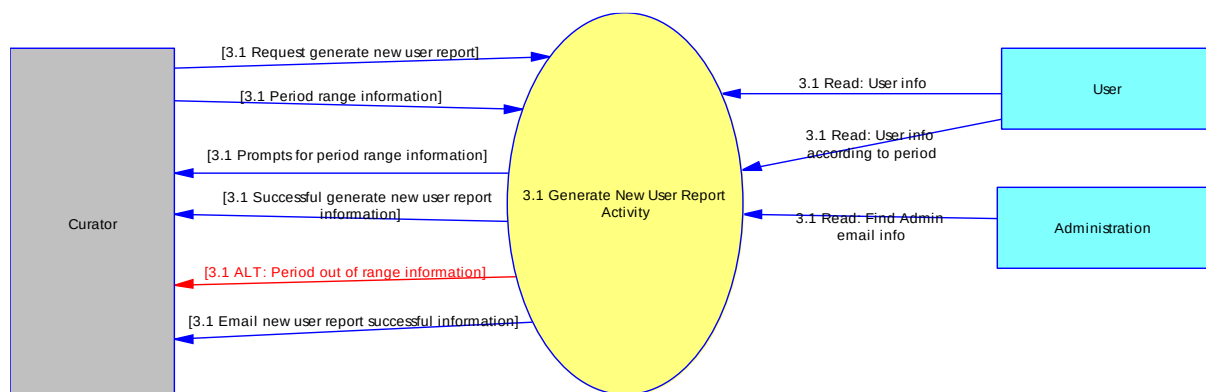


Figure 95: Middle Level DFD: 3.1 Generate New User Report

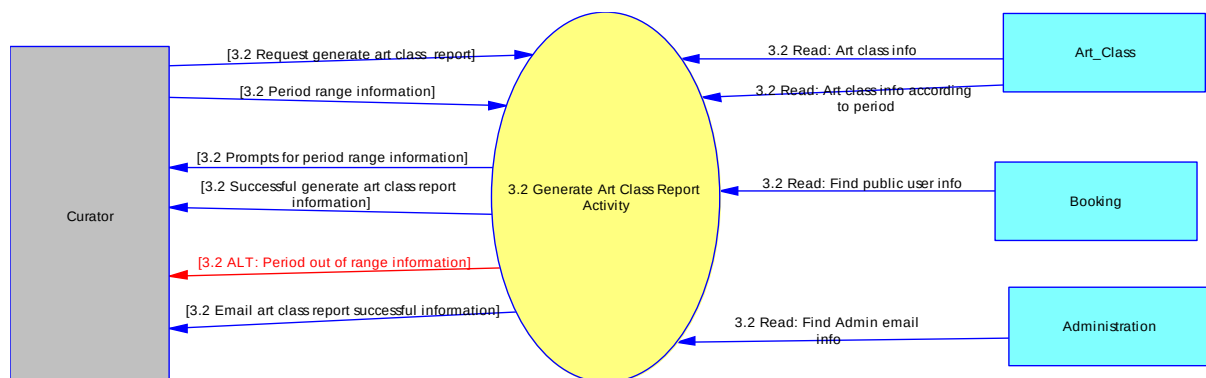


Figure 96: Middle Level DFD: 3.2 Generate Art Class Report

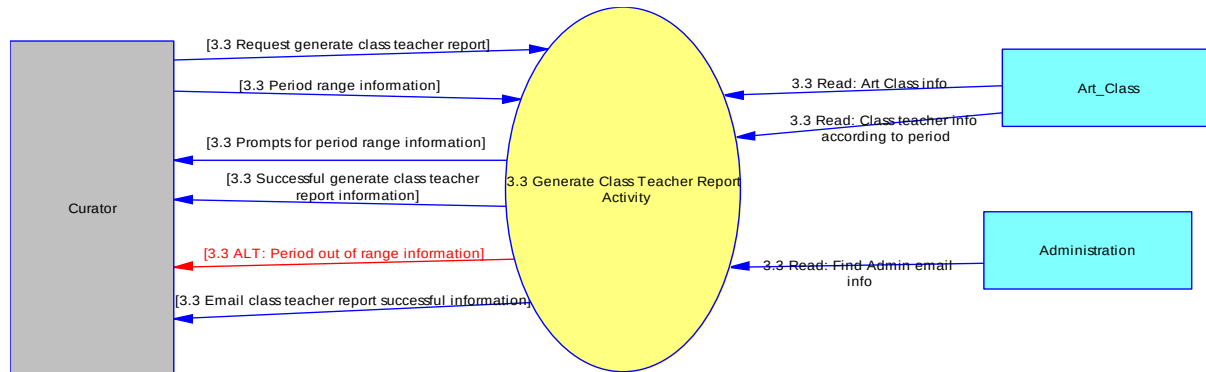


Figure 97: Middle Level DFD: 3.3 Generate Class_Teacher Report

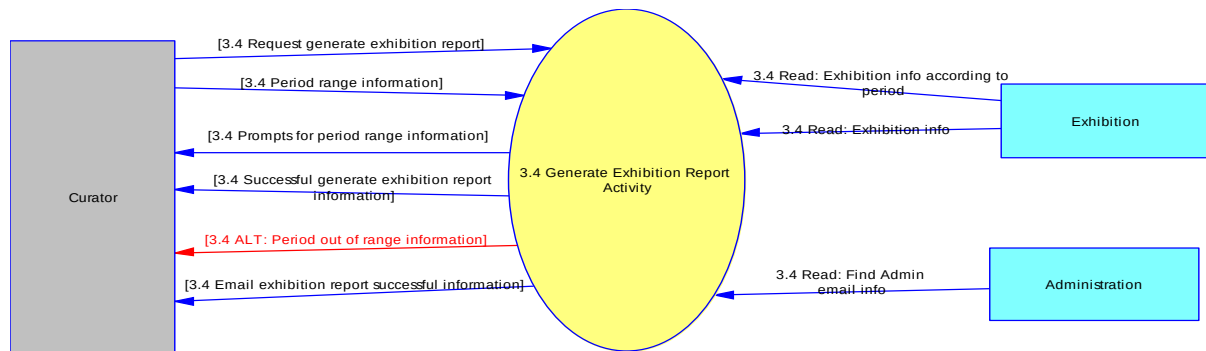


Figure 98: Middle Level DFD: 3.4 Generate Exhibition Report

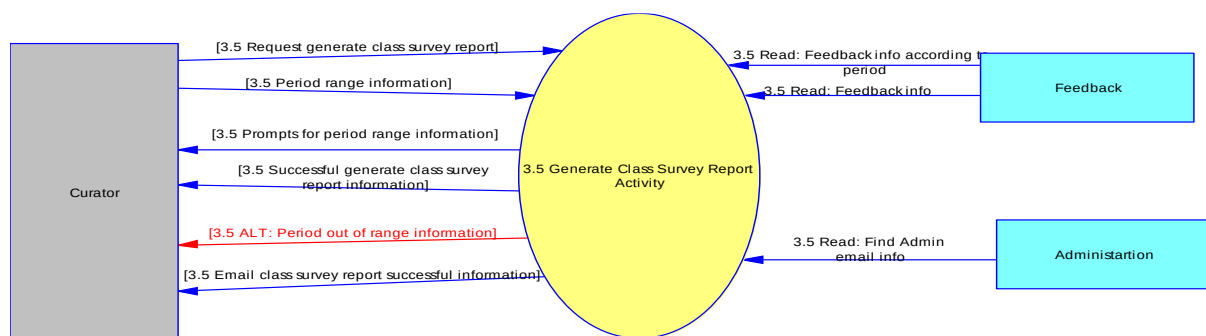


Figure 99: Middle Level DFD: 3.5 Generate Class Survey Report

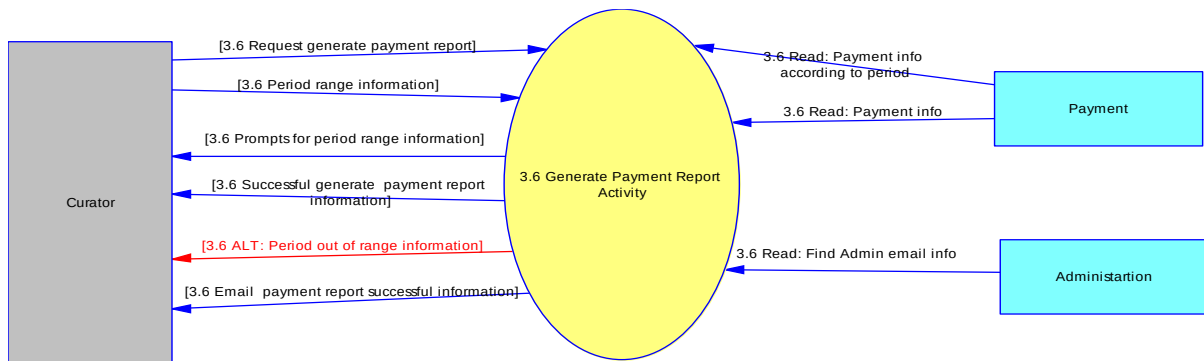


Figure 100: Middle Level DFD: 3.6 Generate Payment Report

3.3.2.4 User Subsystem

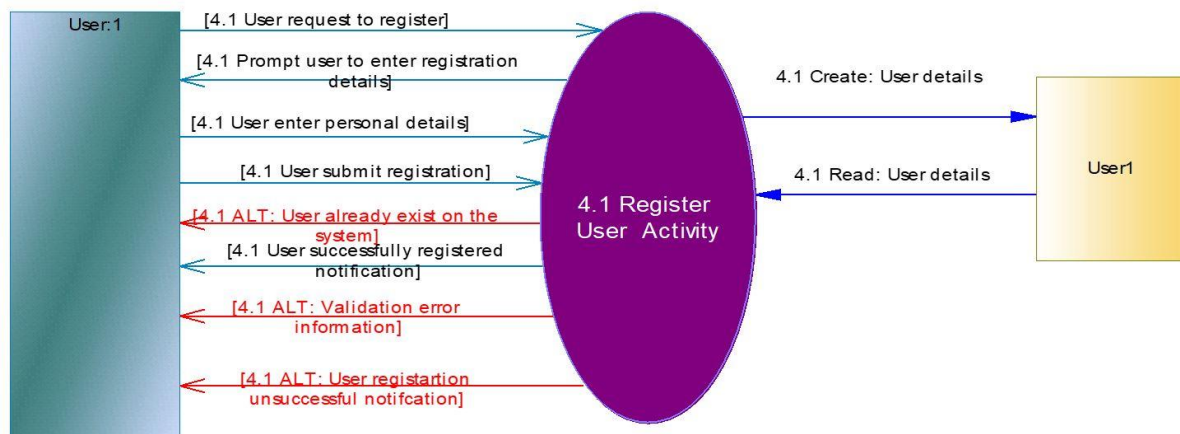


Figure 101: Middle Level DFD: 4.1 Register User

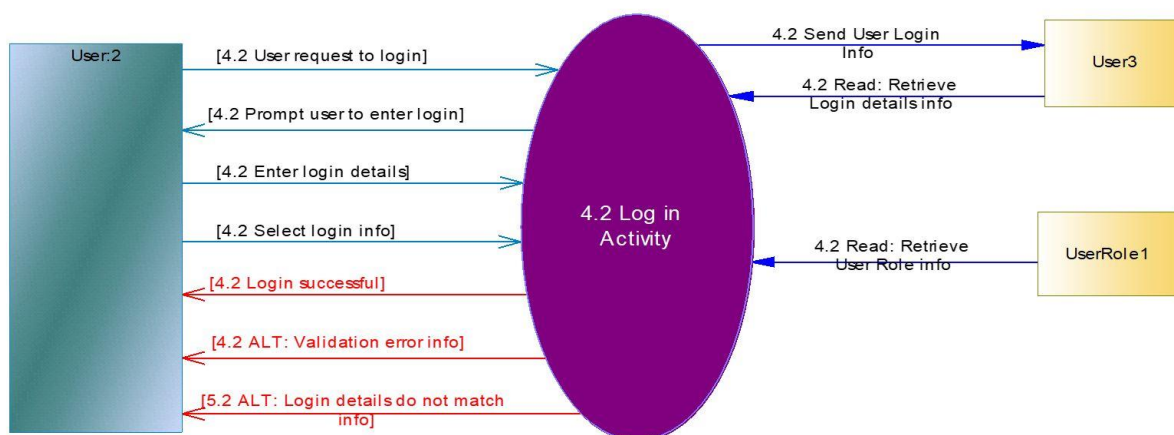


Figure 102: Middle Level DFD: 4.2 Log in

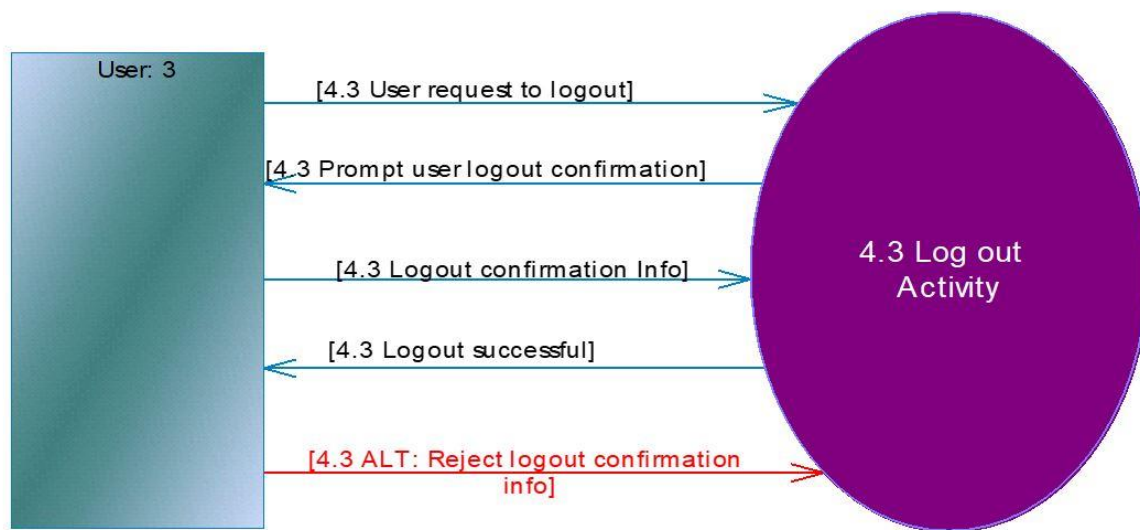


Figure 103: Middle Level DFD: 4.3 Log out

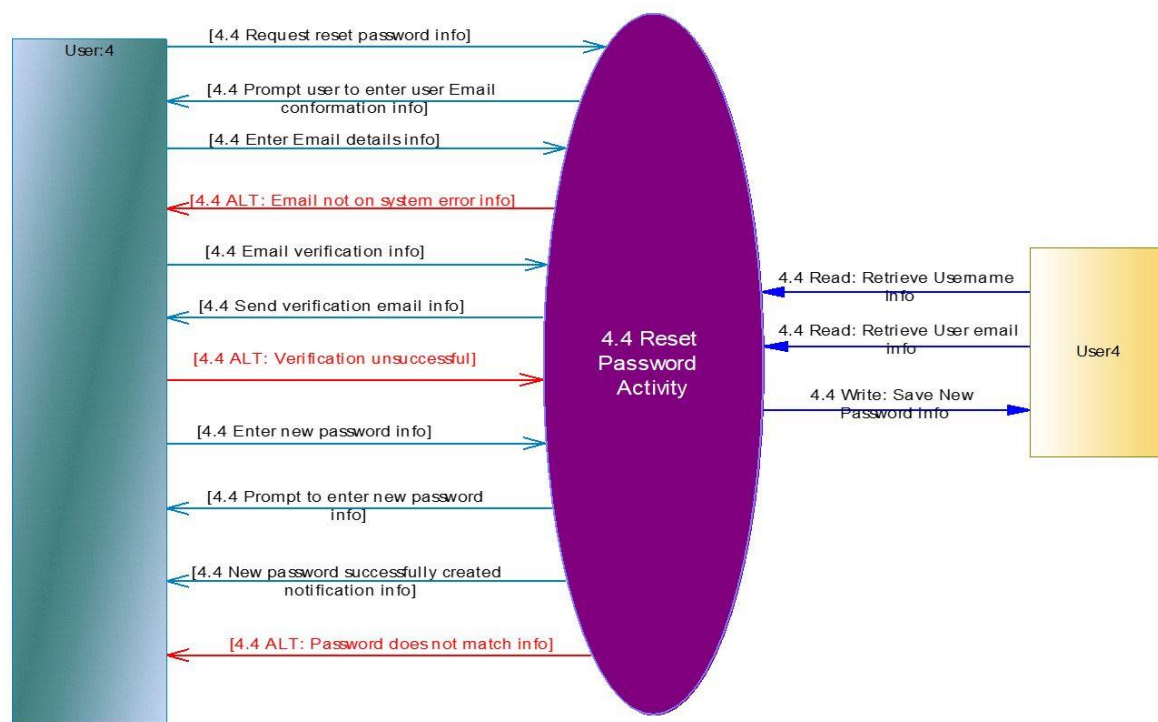


Figure 104: Middle Level DFD: 4.4 Reset Password

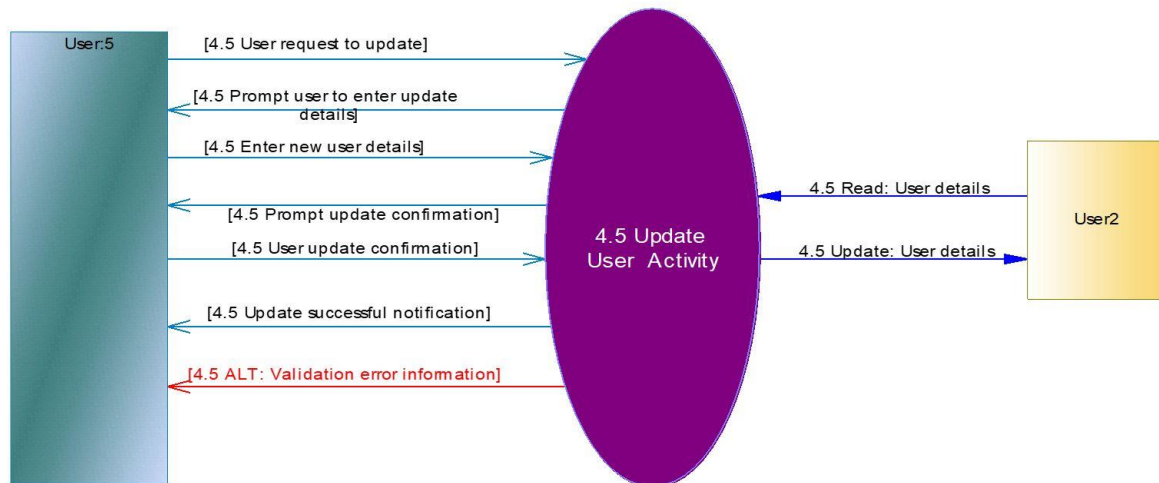


Figure 105: Middle Level DFD: 4.5 Update User

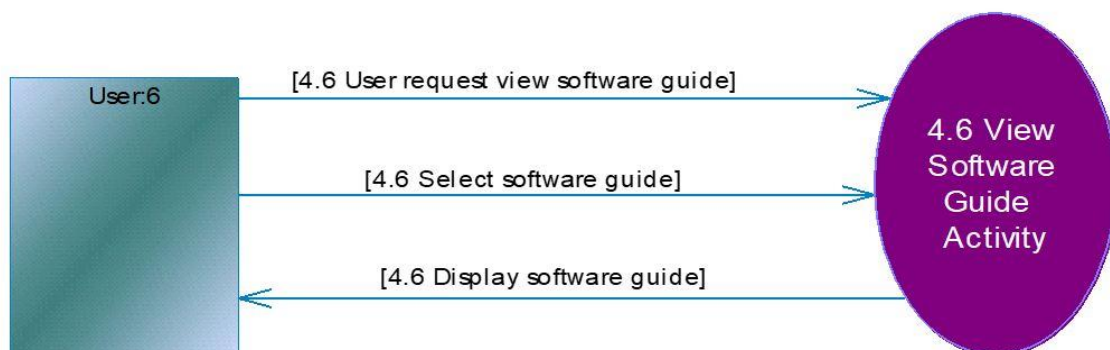


Figure 106: Middle Level DFD: 4.6 View Guide

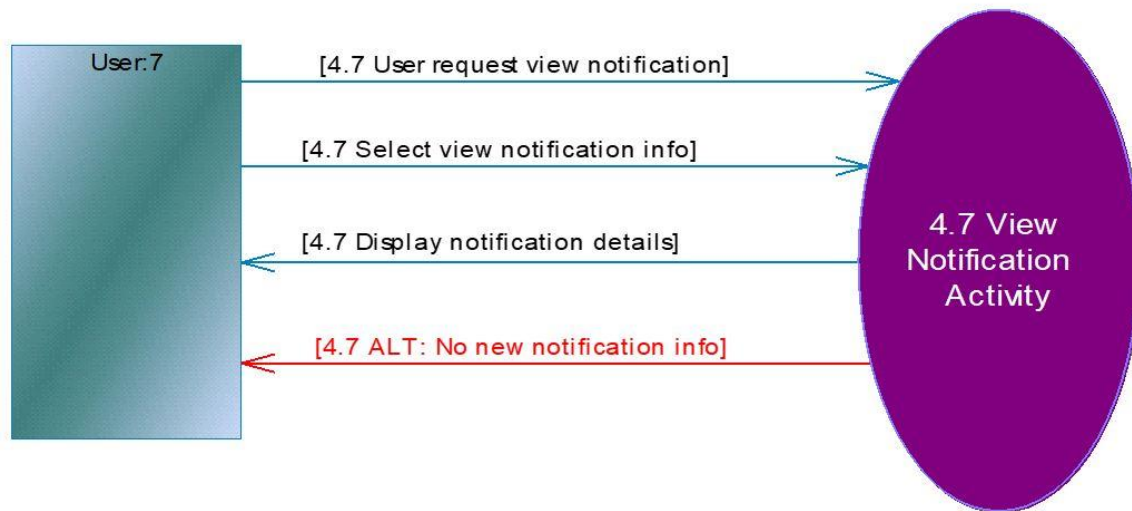


Figure 107: Middle Level DFD: 4.7 View Notification

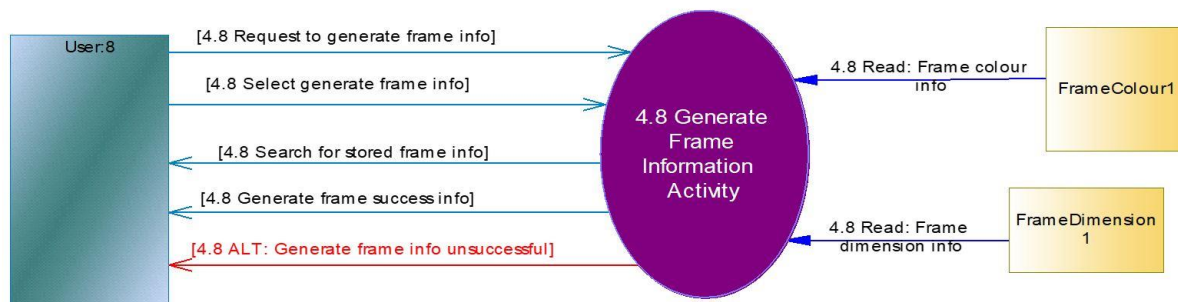


Figure 108: Middle Level DFD: 4.8 Generate Frame Info

3.3.2.5 Artist Subsystem

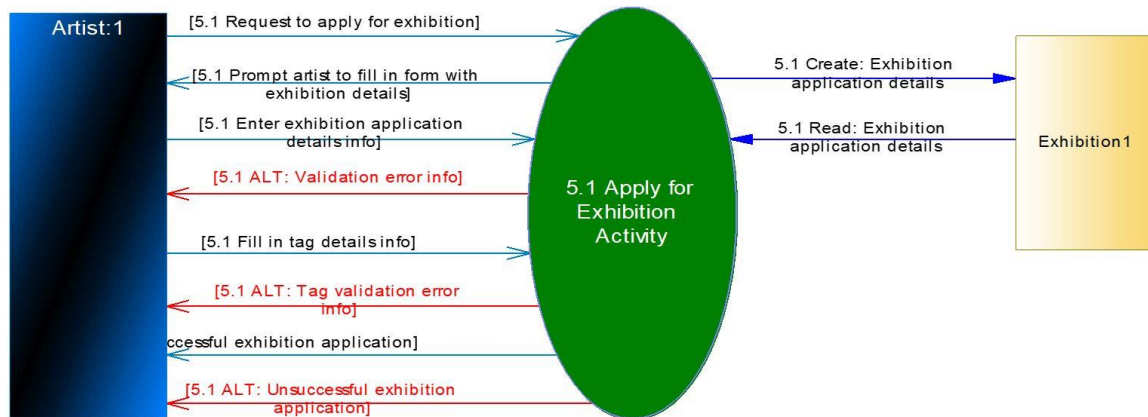


Figure 109: Middle Level DFD: 5.1 Apply for Exhibition

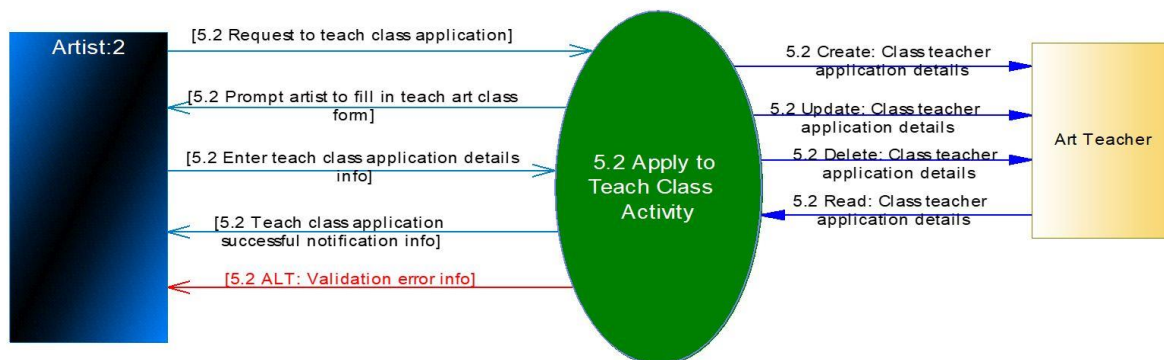


Figure 110: Middle Level DFD: 5.2 Apply Teach Class

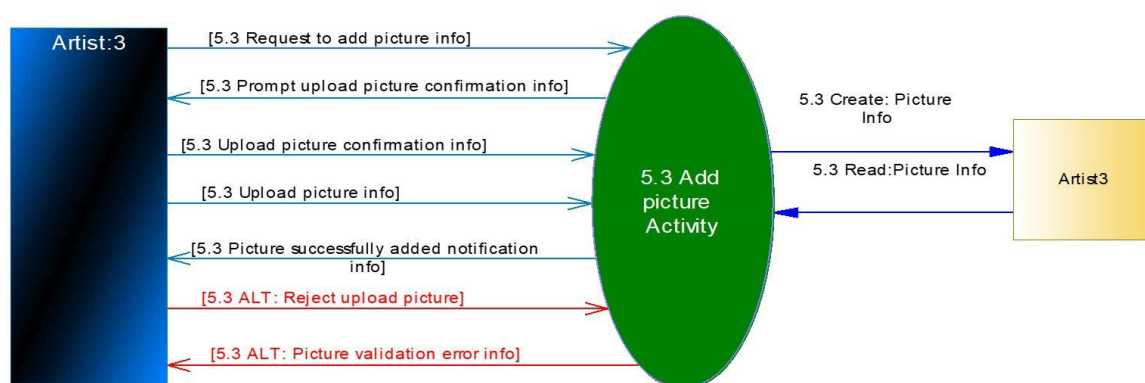


Figure 111: Middle Level DFD: 5.3 Add Picture

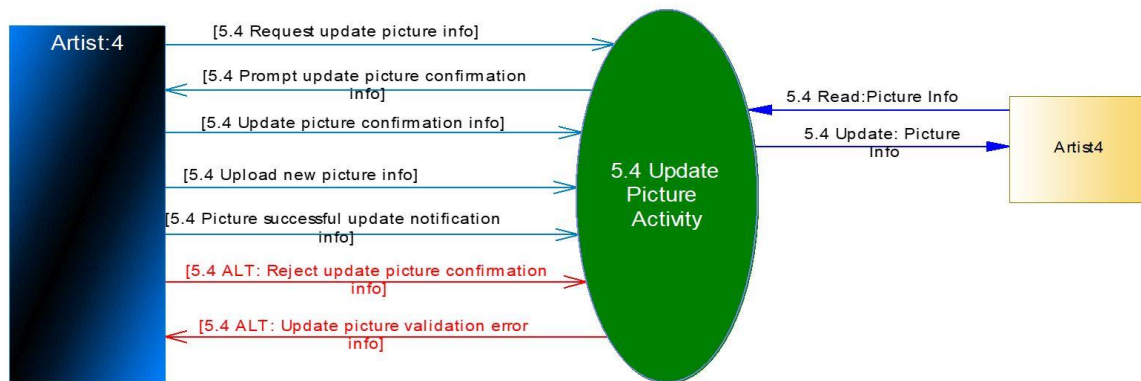


Figure 112: Middle Level DFD: 5.4 Update Picture

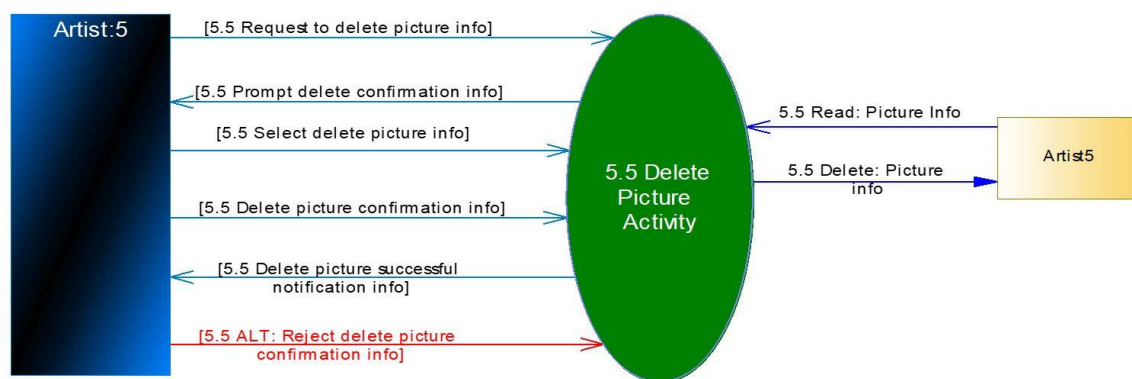


Figure 113: Middle Level DFD: 5.5 Delete Picture

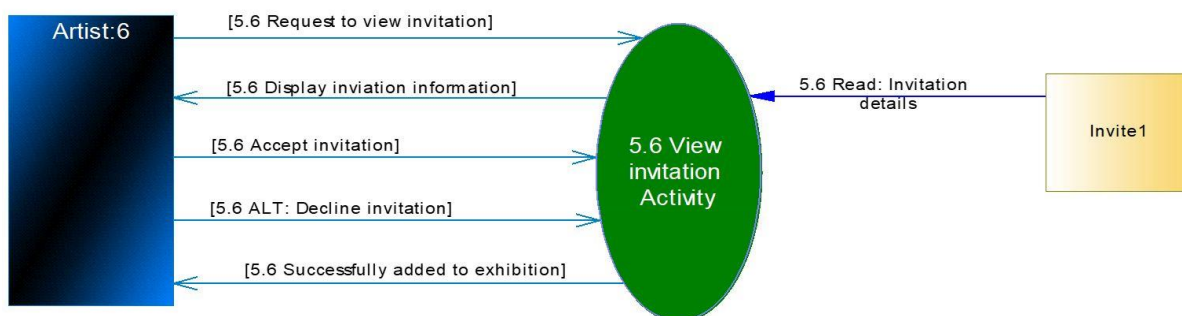


Figure 114: Middle Level DFD: 5.6 View Invitation

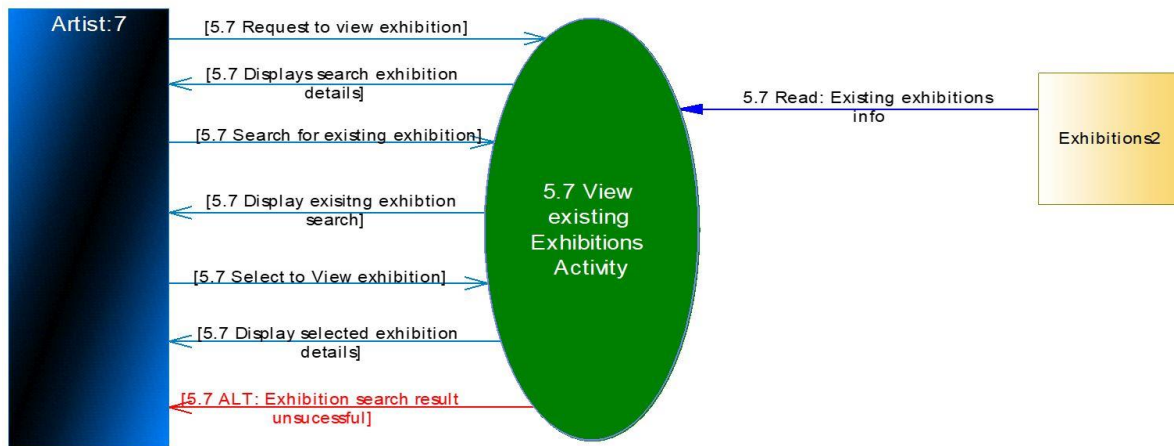


Figure 115: Middle Level DFD: 5.7 View Exhibitions

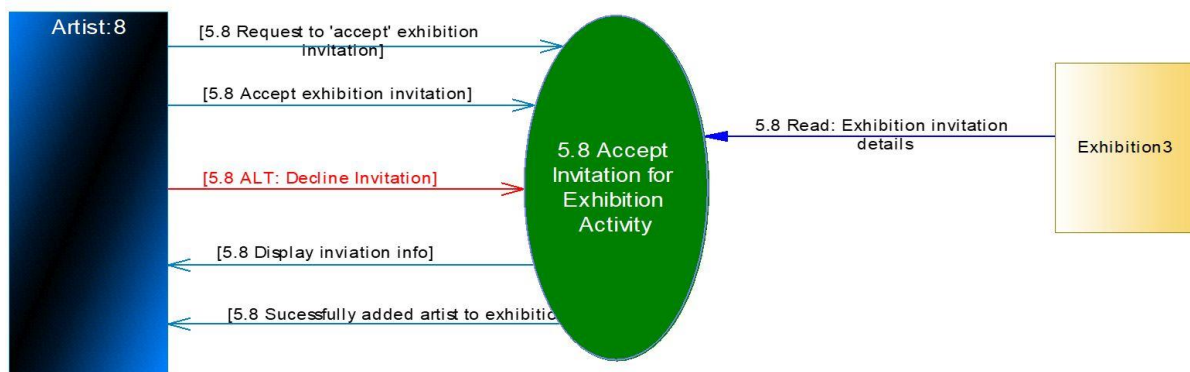


Figure 116: Middle Level DFD: 5.8 Accept Invitation

3.3.3 Low Level Data Flow Diagrams

3.3.3.1 Administration Subsystem

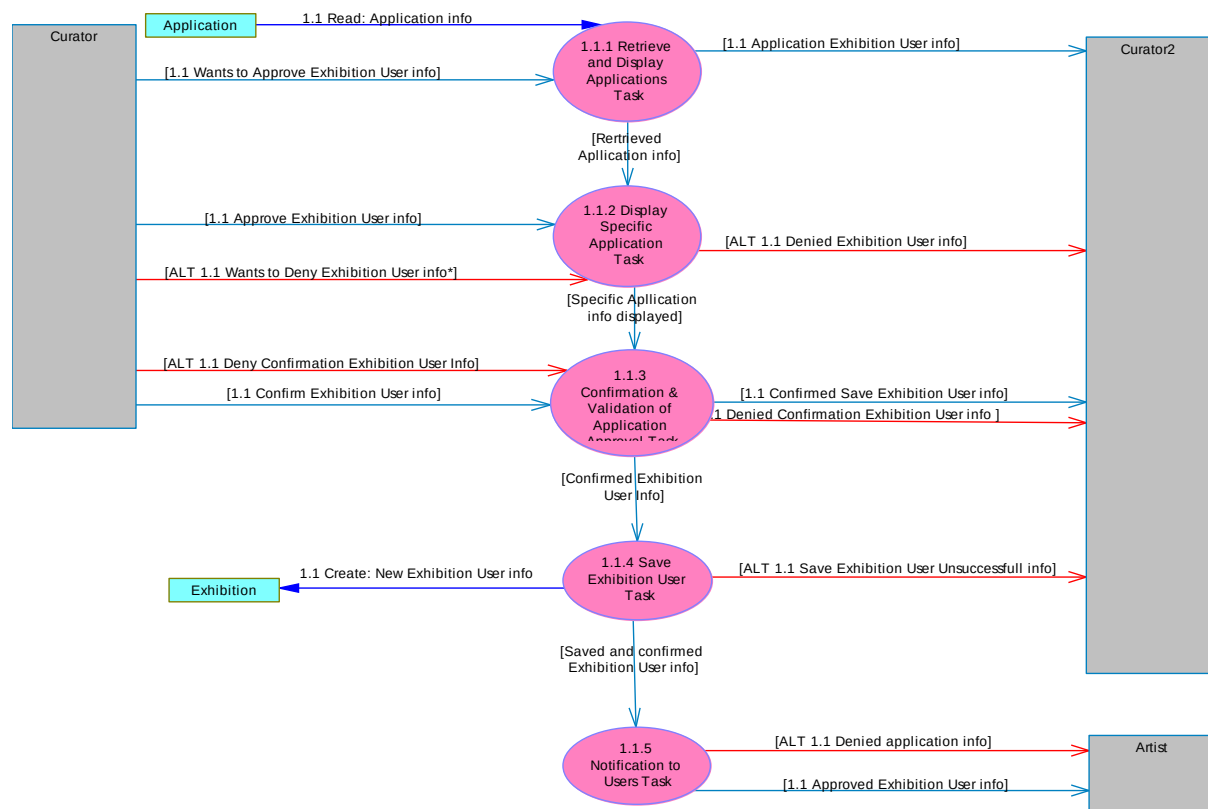


Figure 117: Low Level DFD: 1.1 Approve Exhibition User

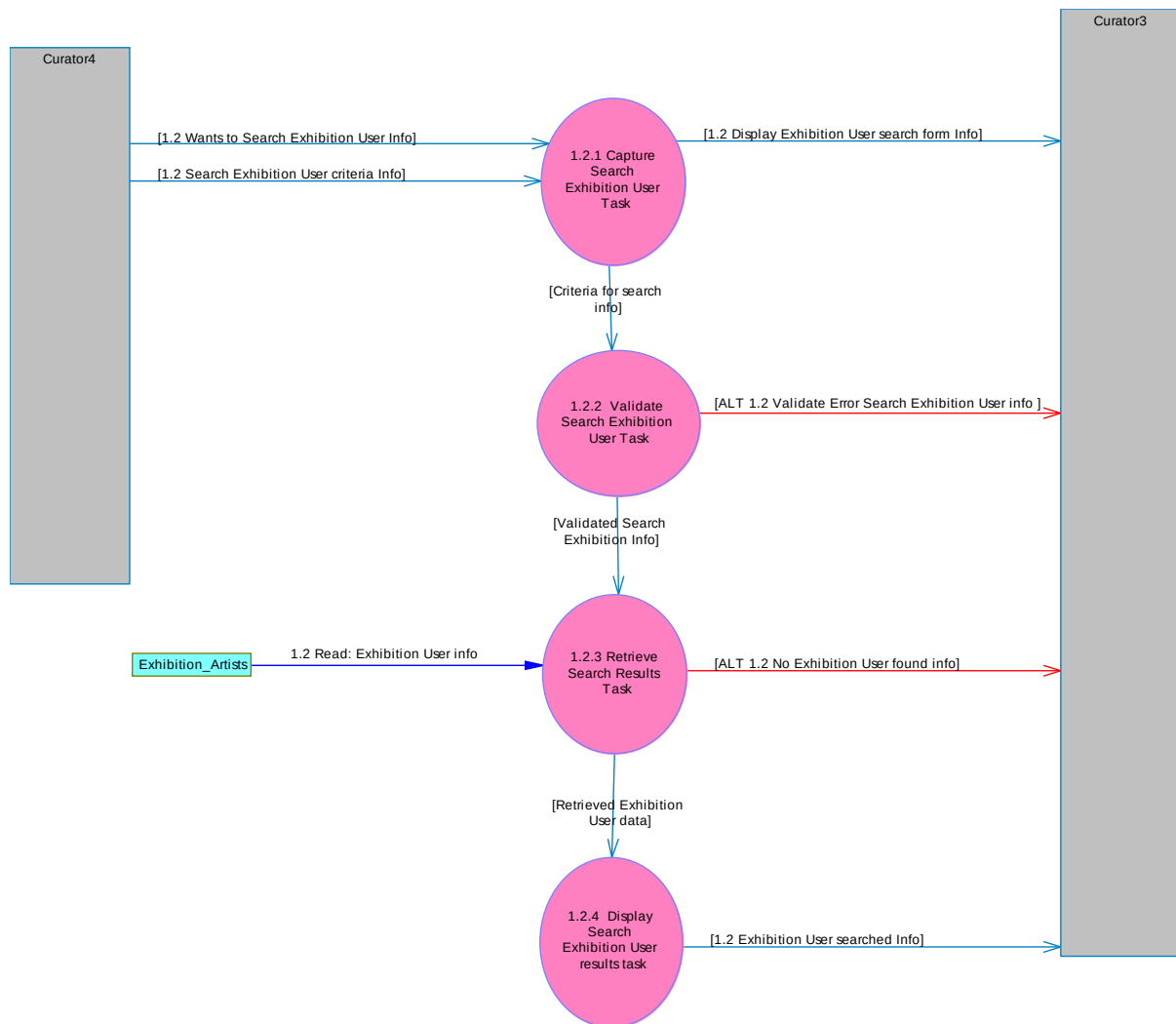


Figure 118: Low Level DFD: 1.2 Search Exhibition User

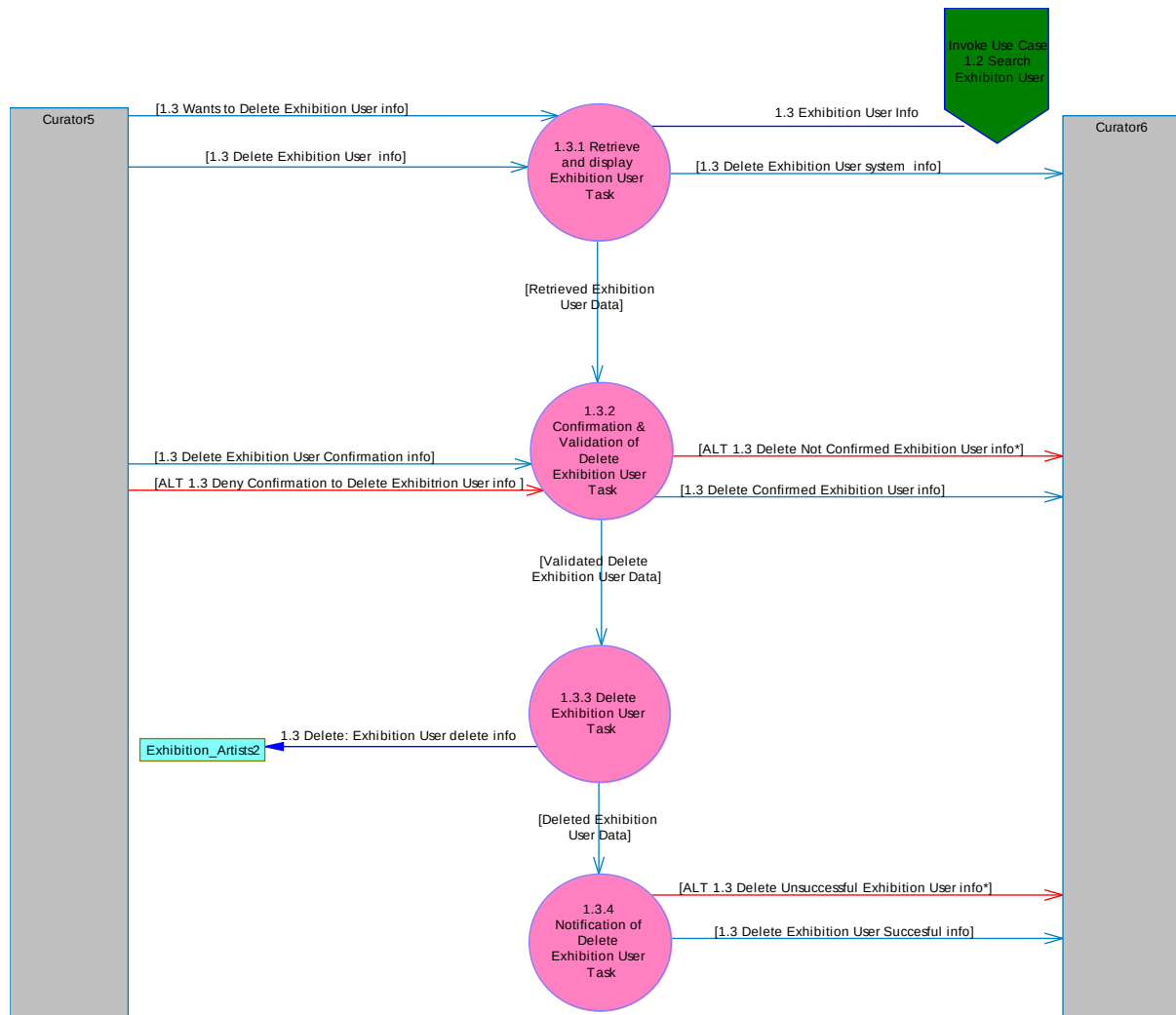


Figure 119: Low Level DFD: 1.3 Notify Delete Exhibition User

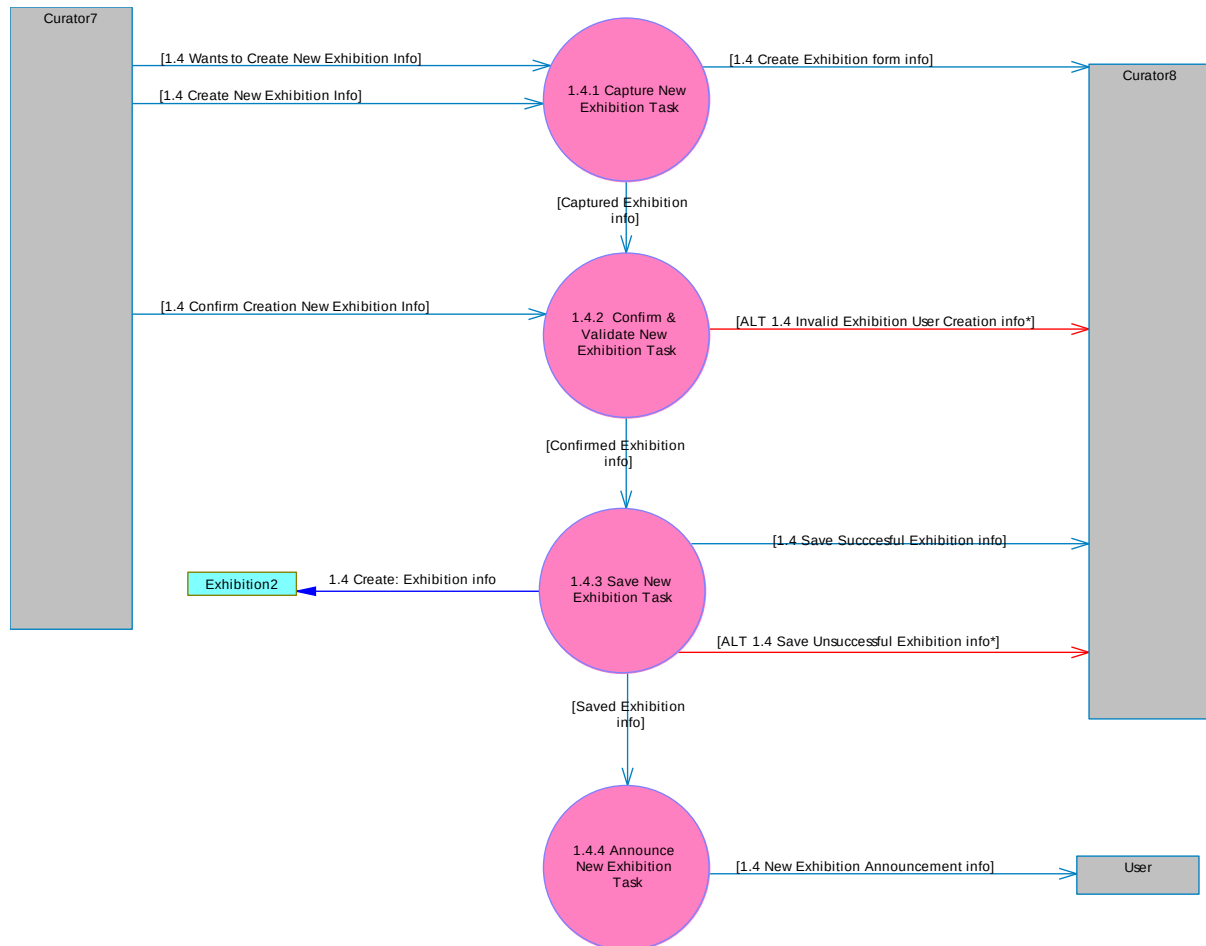


Figure 120: Low Level DFD: 1.4 Create New Exhibition

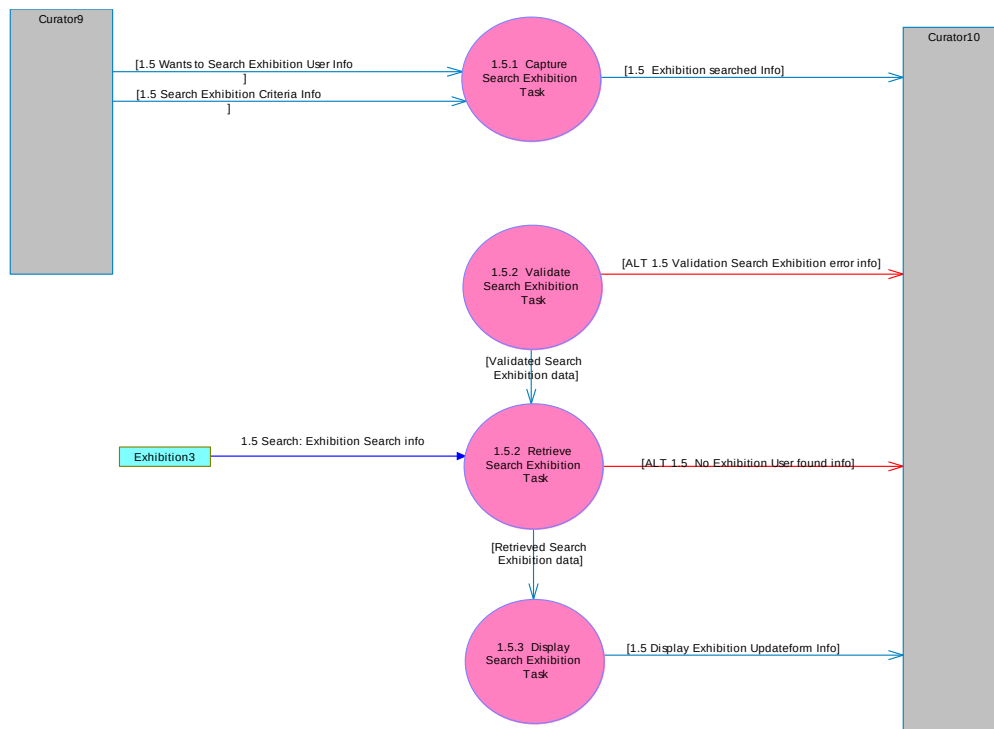


Figure 121: Low Level DFD: 1.5 Search Exhibition

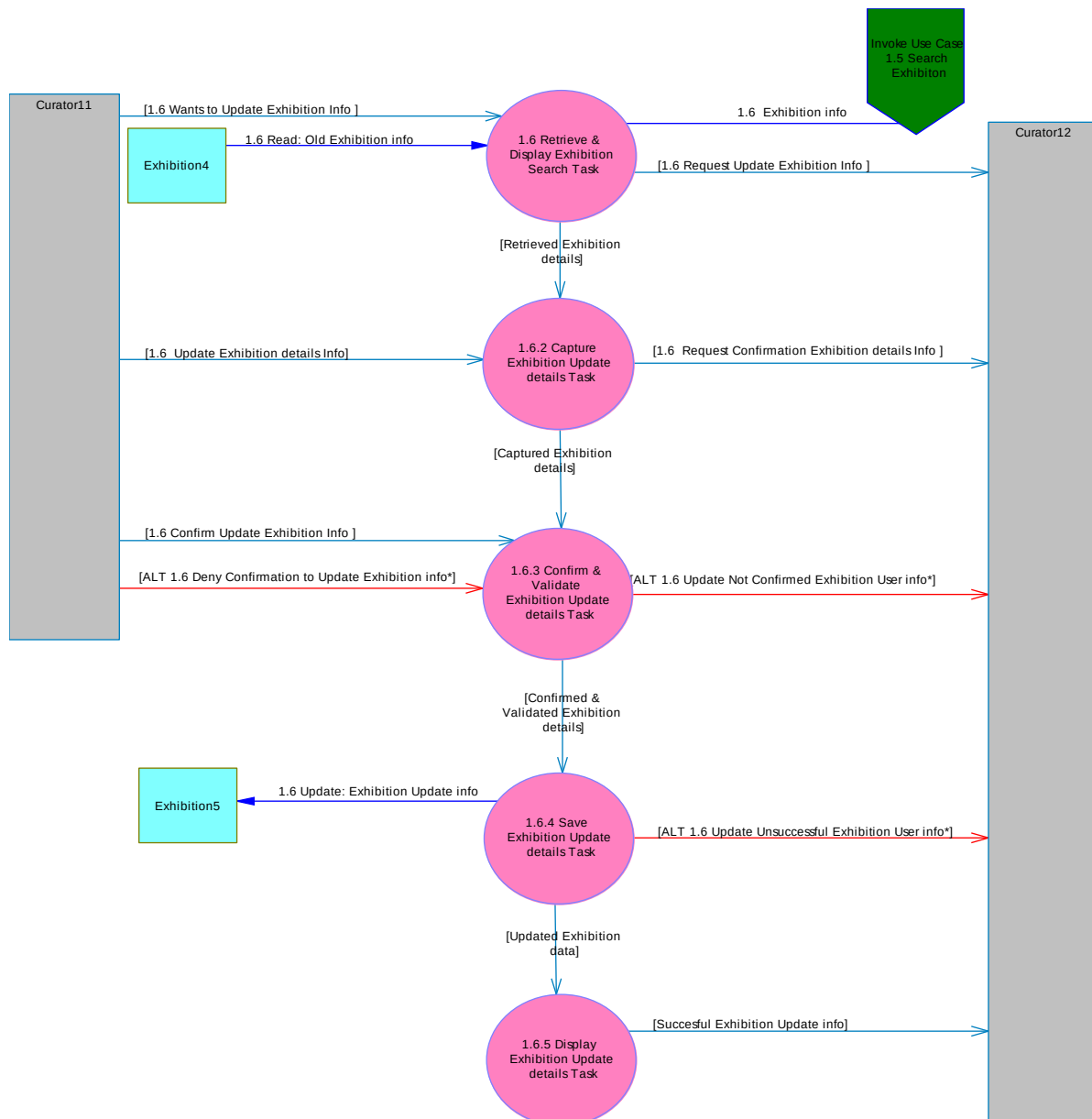


Figure 122: Low Level DFD: 1.6 Update Exhibition

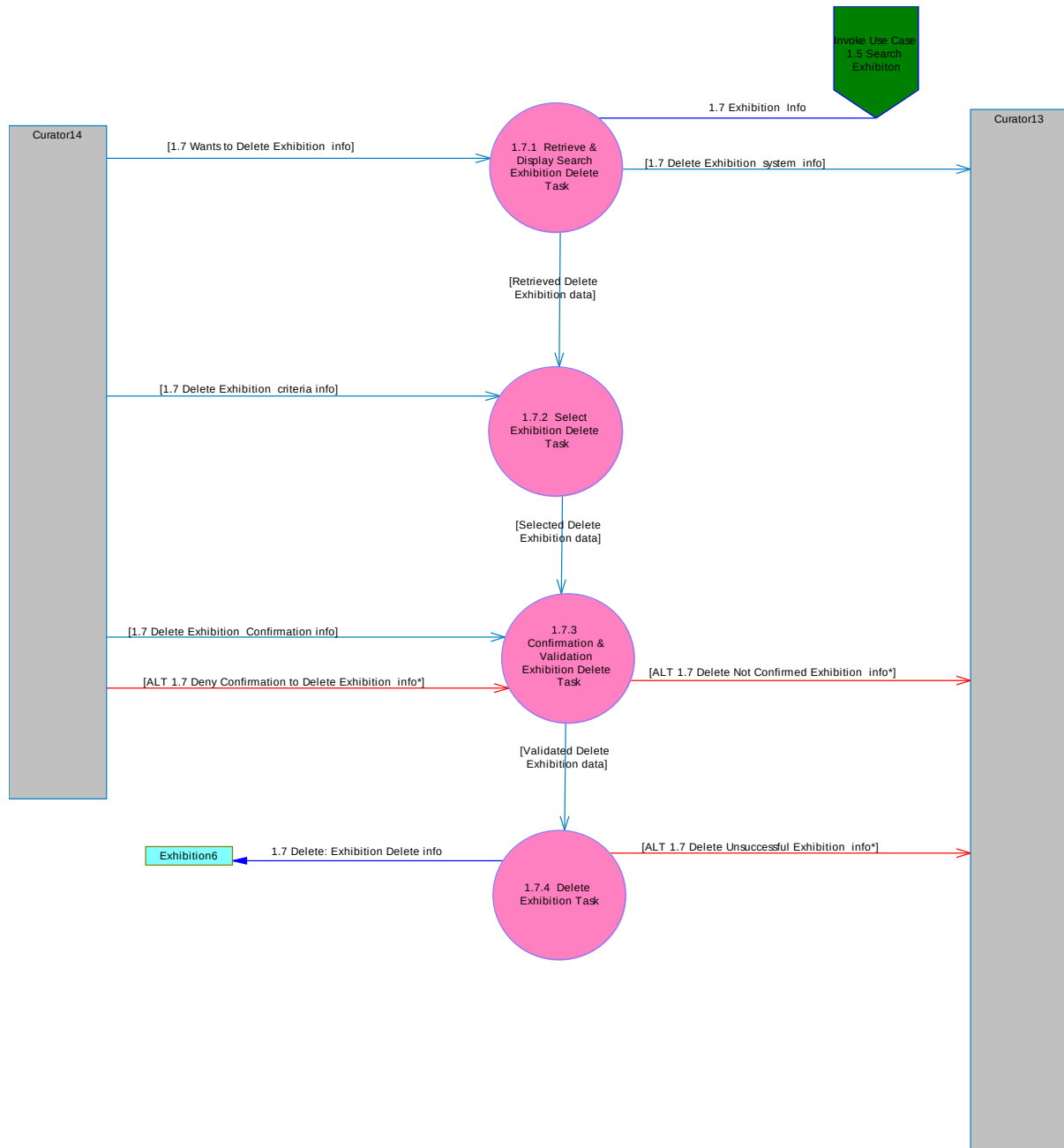


Figure 123: Low Level DFD: 1.7 Delete Exhibition

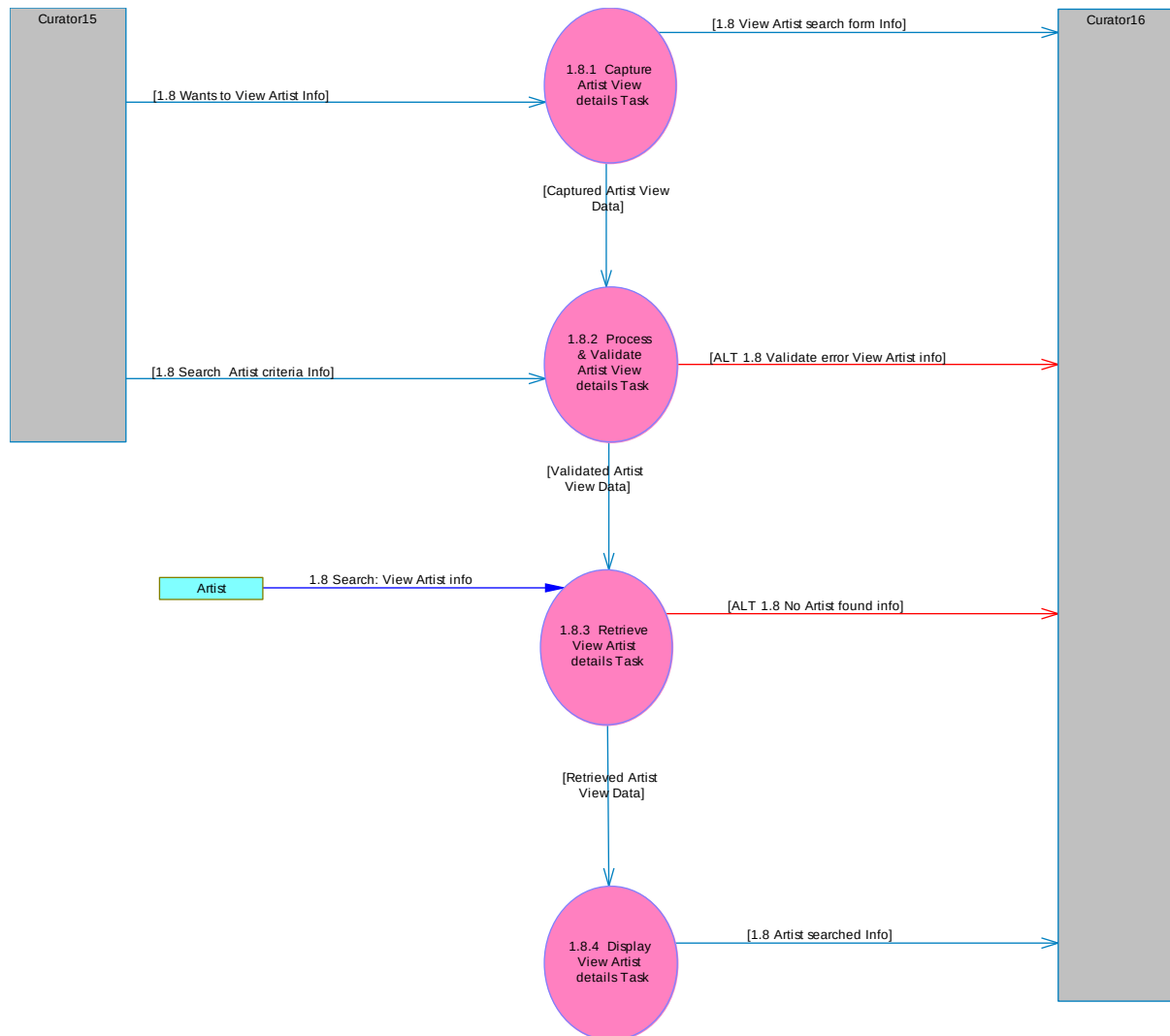


Figure 124: Low Level DFD: 1.8 View Artist

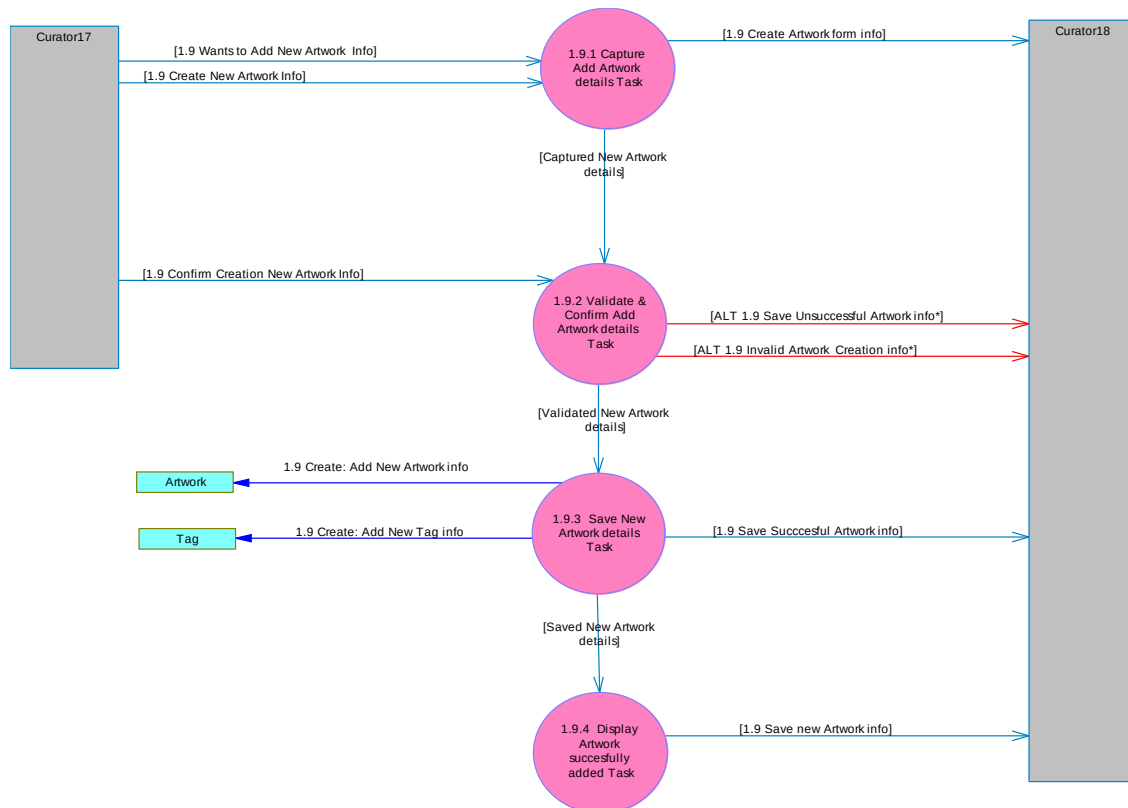


Figure 125: Low Level DFD: 1.9 Add Artwork

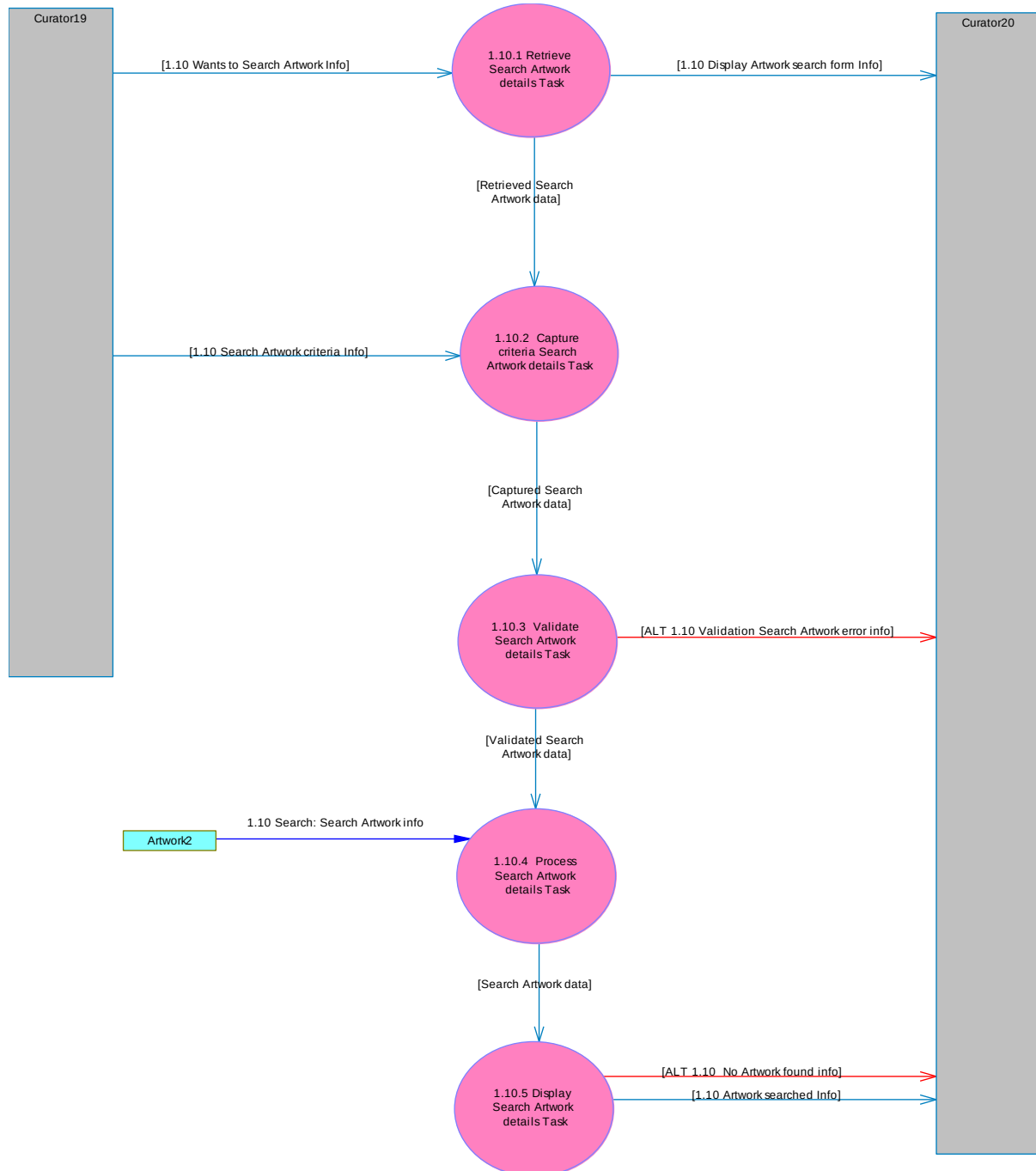


Figure 126v: Low Level DFD: 1.10 Search Artwork

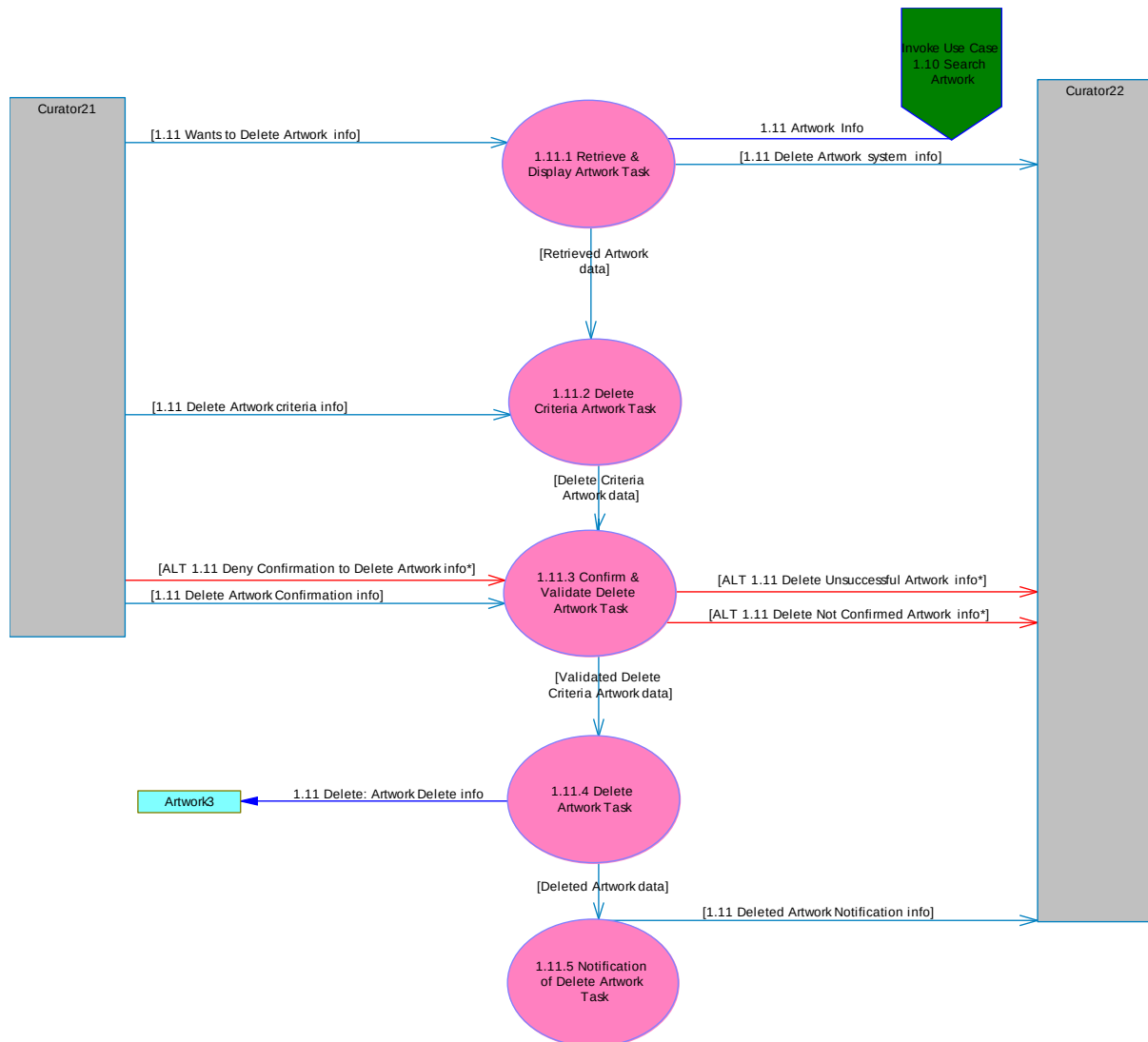


Figure 127: Low Level DFD: 1.11 Delete Artwork

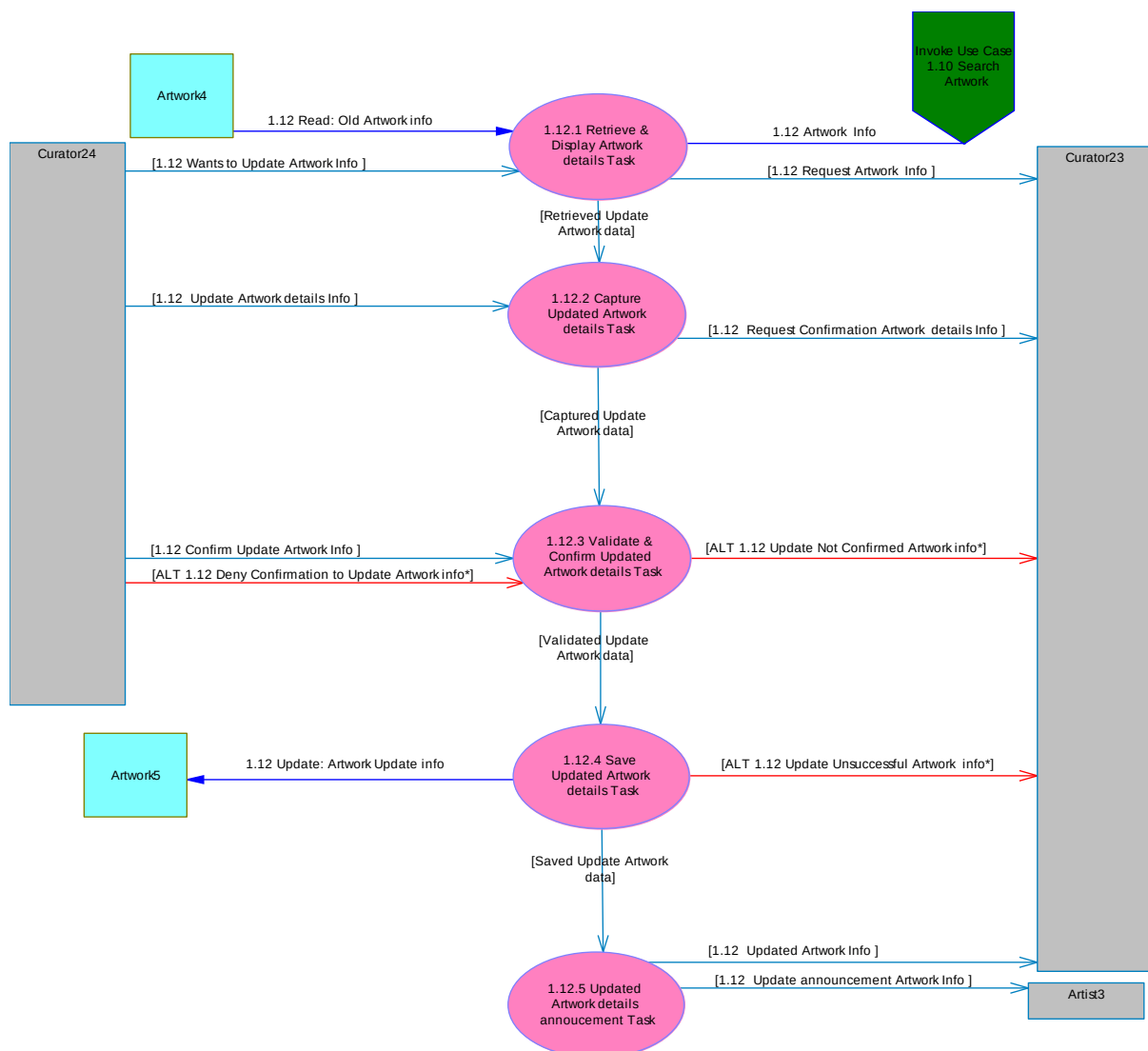


Figure 128: Low Level DFD: 1.12 Update Artwork

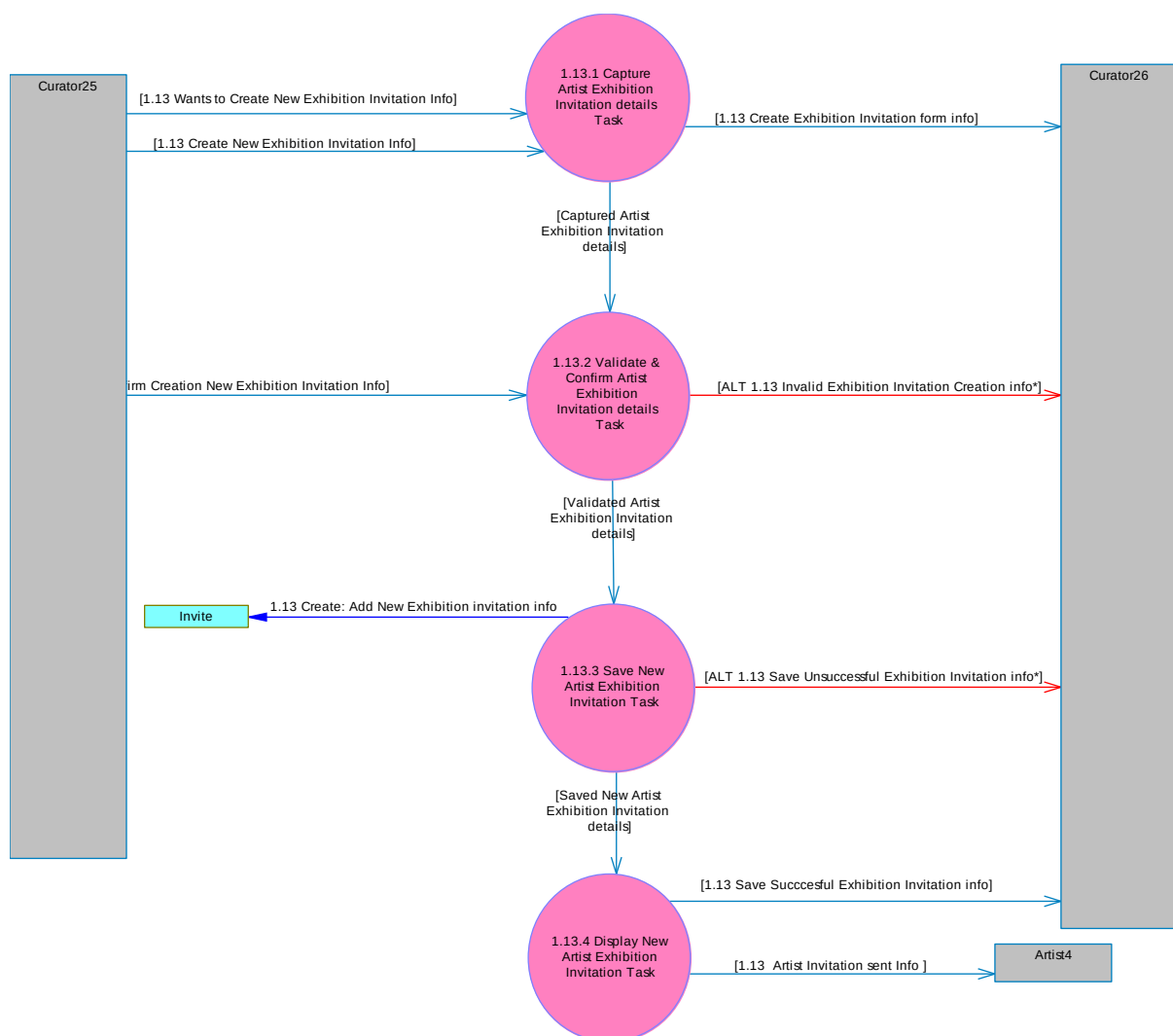


Figure 129: Low Level DFD: 1.13 Create Exhibition Invitation

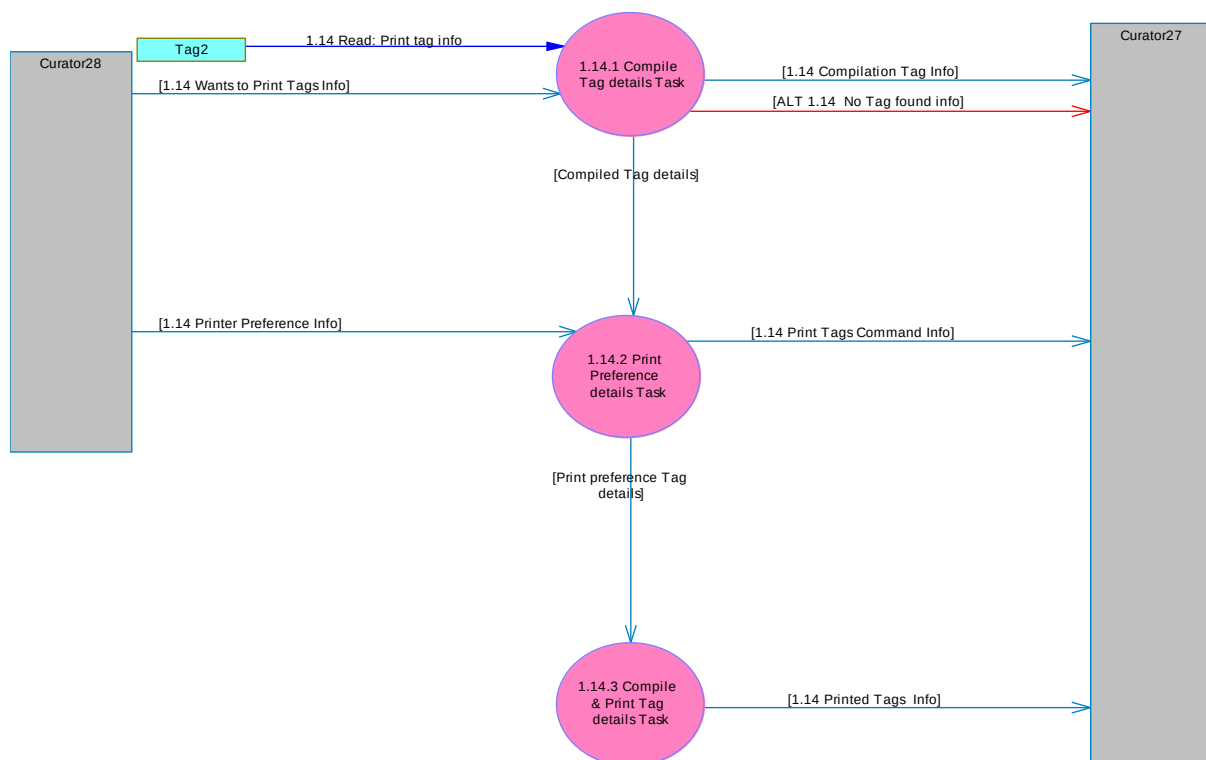


Figure 130: Low Level DFD: 1.14 Print Tags

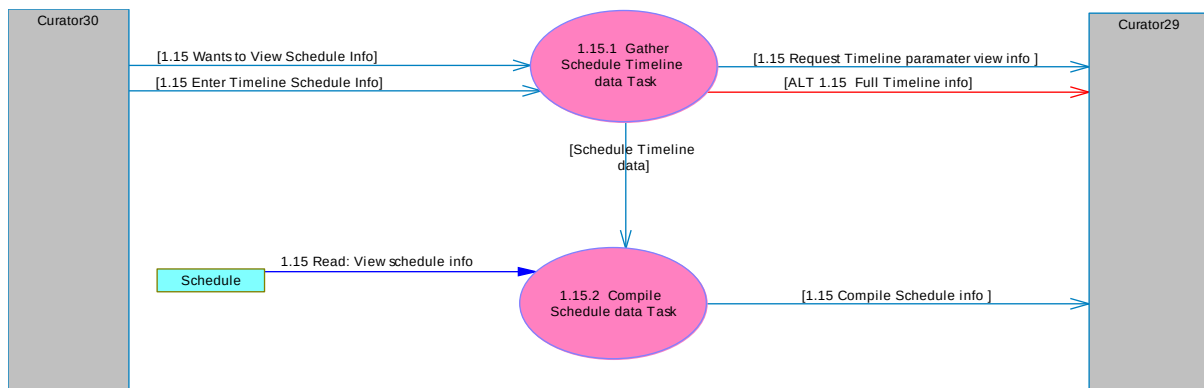


Figure 131: Low Level DFD: 1.15 View Schedule

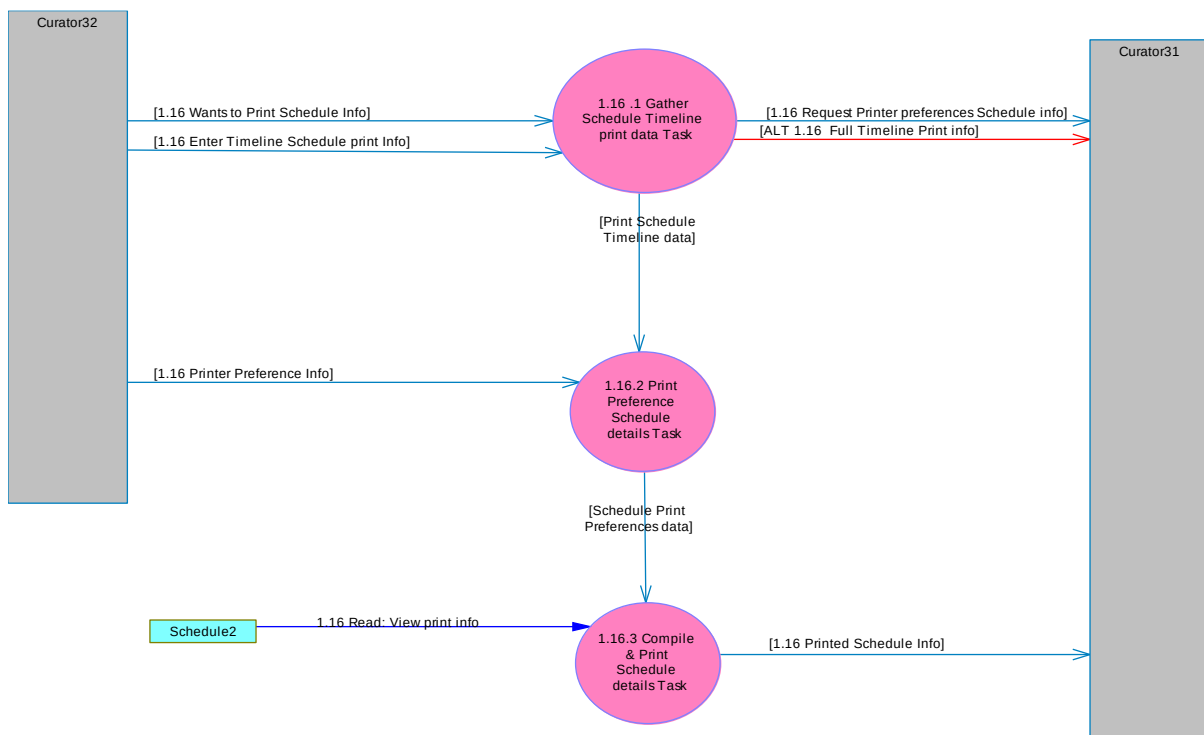


Figure 132: Low Level DFD: 1.16 Print Schedule

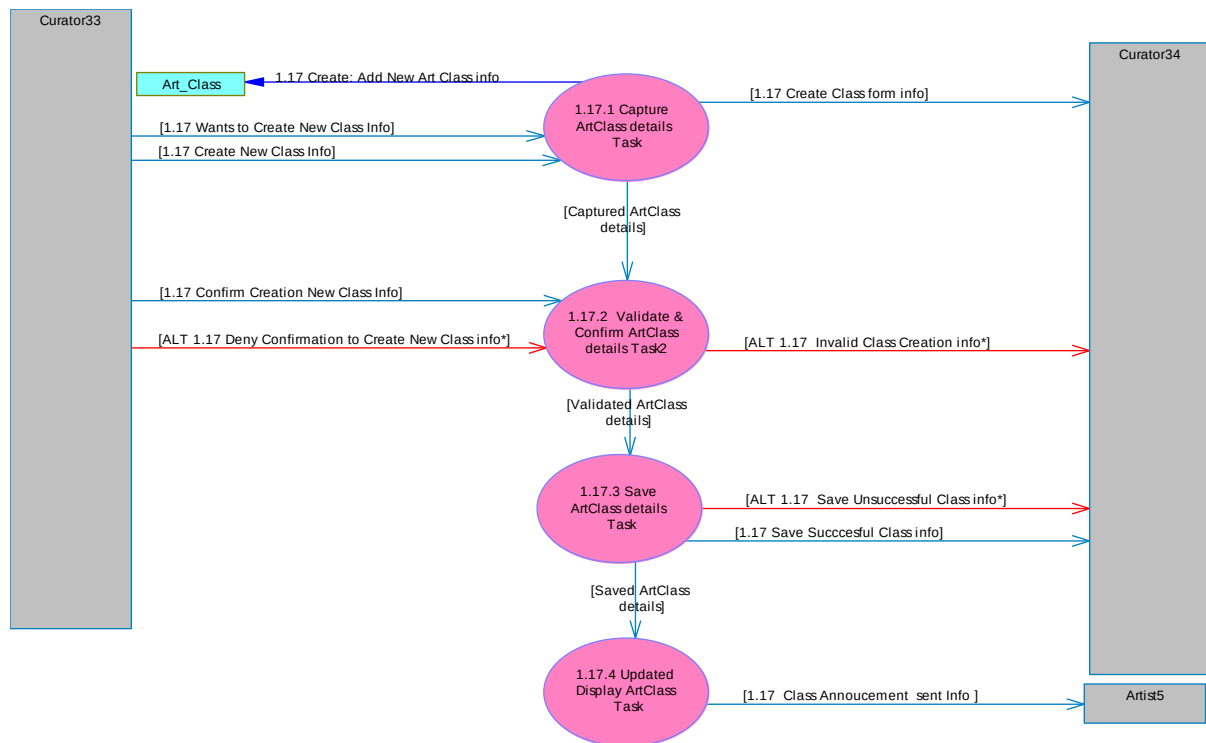


Figure 133: Low Level DFD: 1.17 Create Class

PRIMITIVE DIAGRAMS

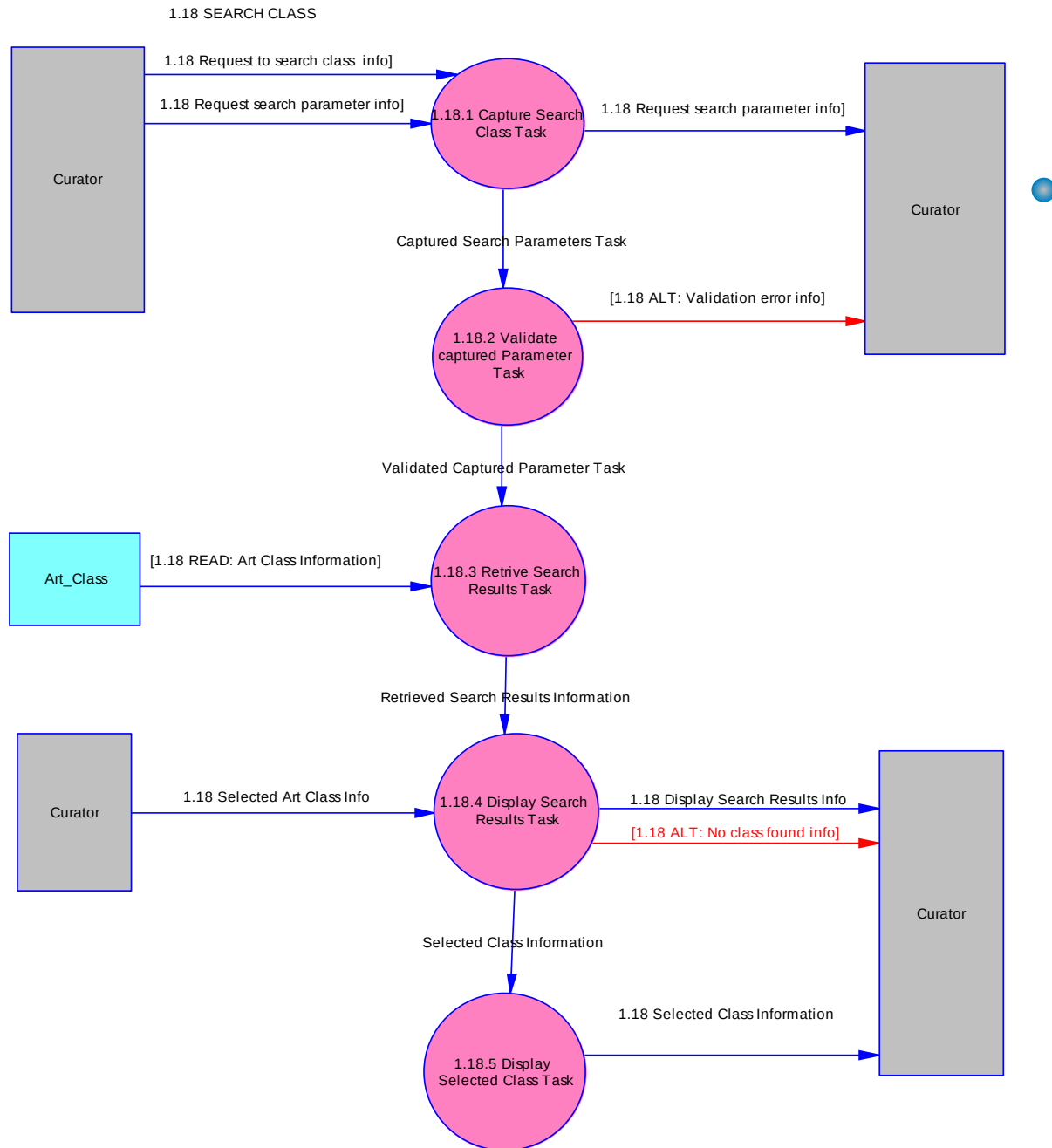


Figure 134: Low Level DFD: 1.18 Search Class

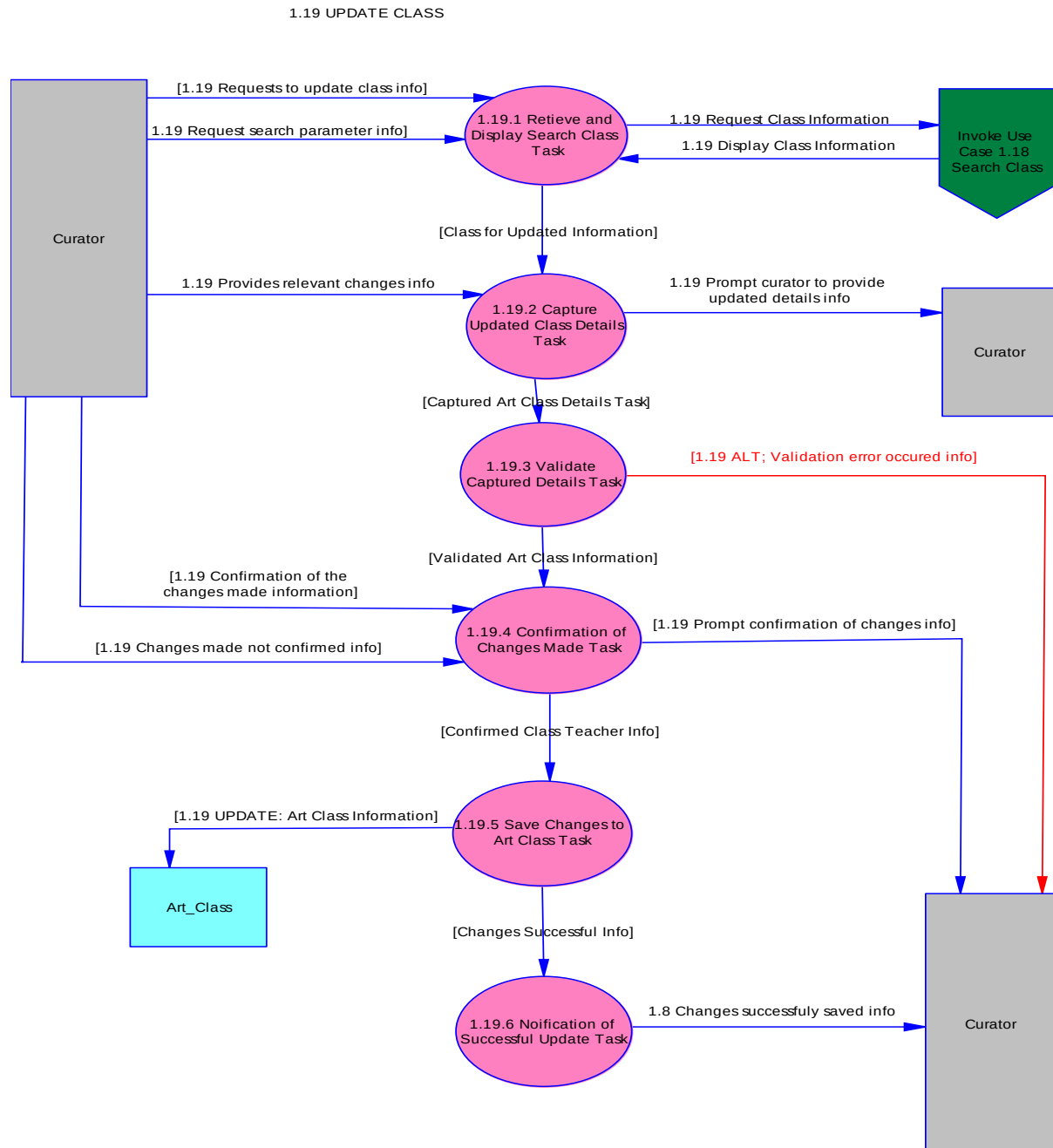


Figure 135: Low Level DFD: 1.19 Update Class

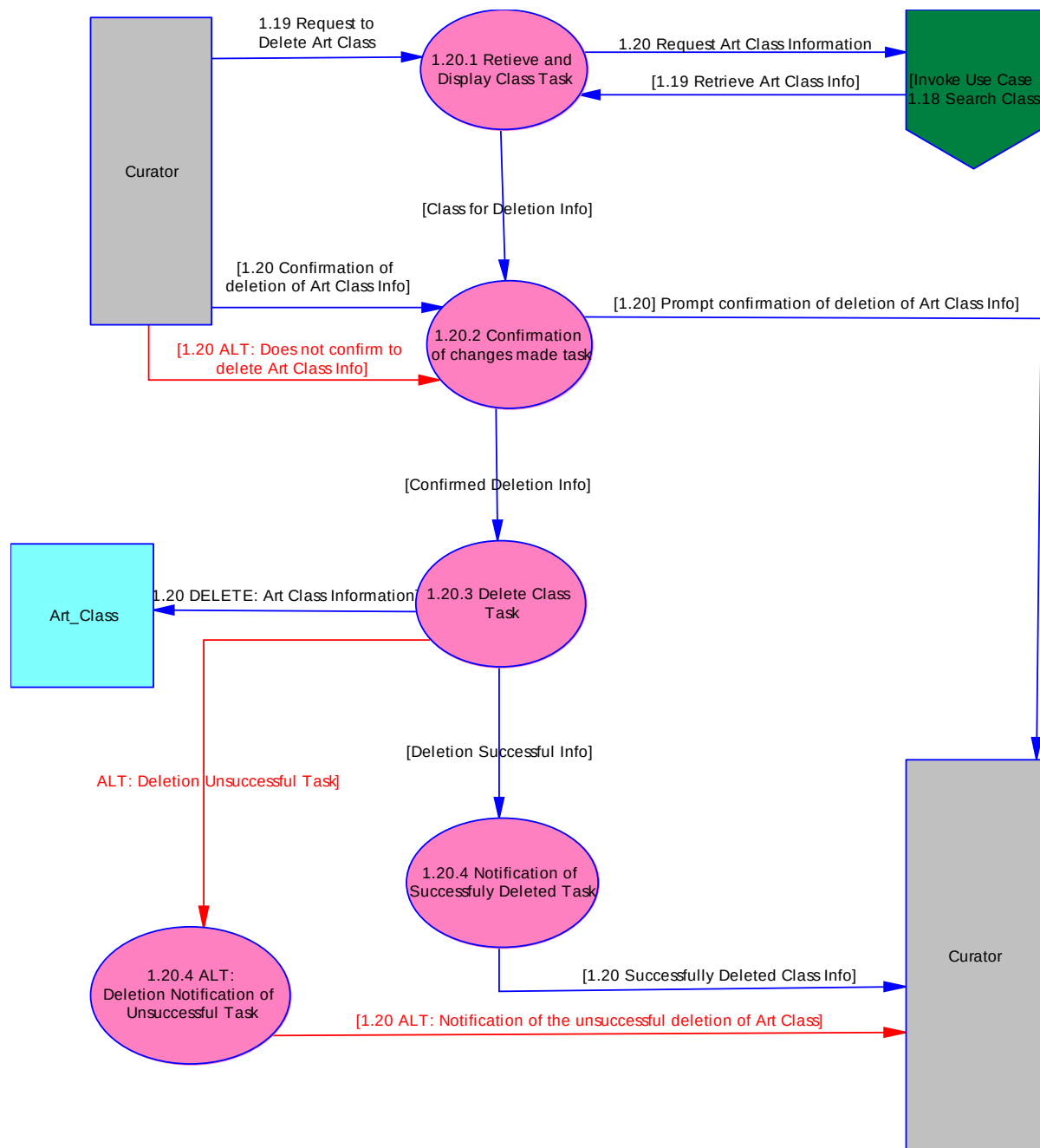


Figure 136: Low Level DFD: 1.20 Delete Class

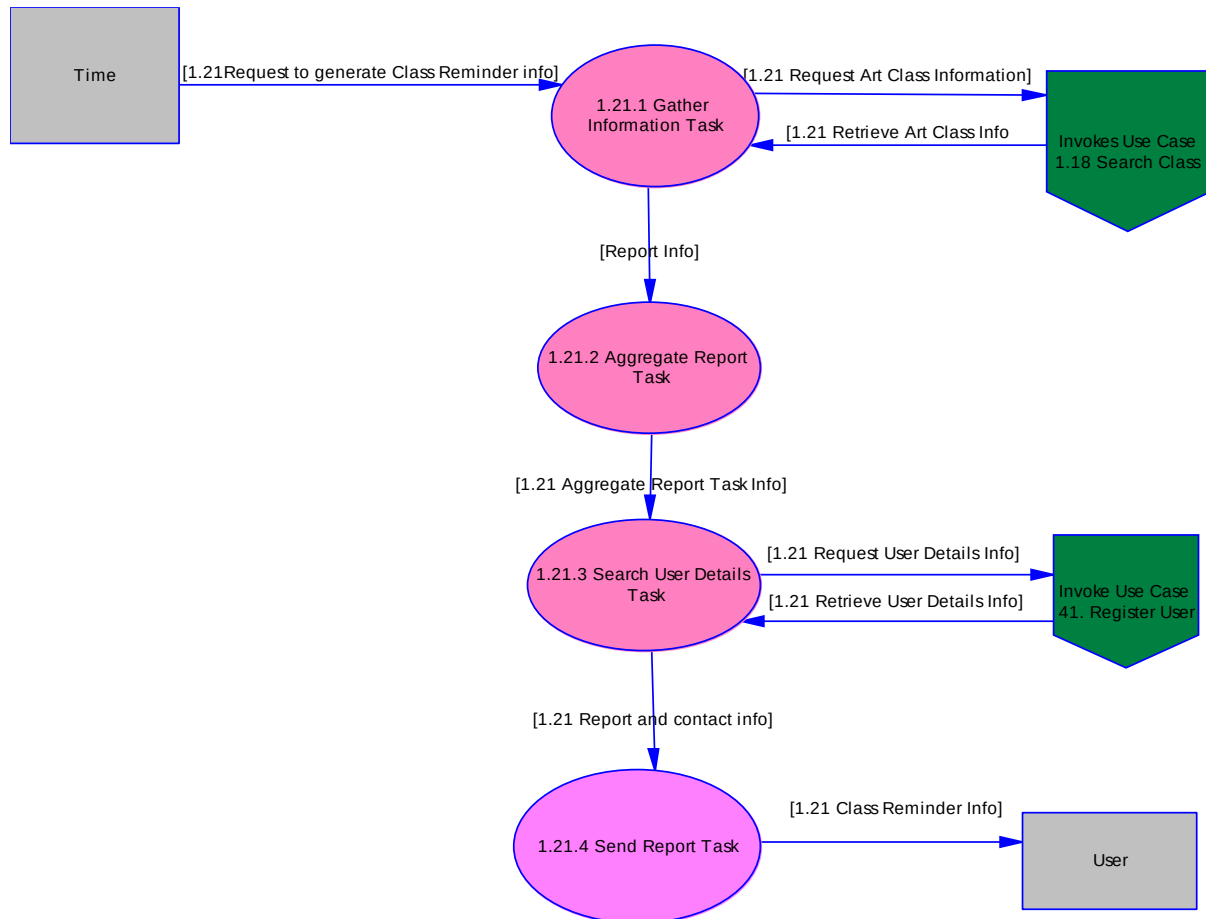


Figure 137: Low Level DFD: 1.21 Generate Payment Reminder

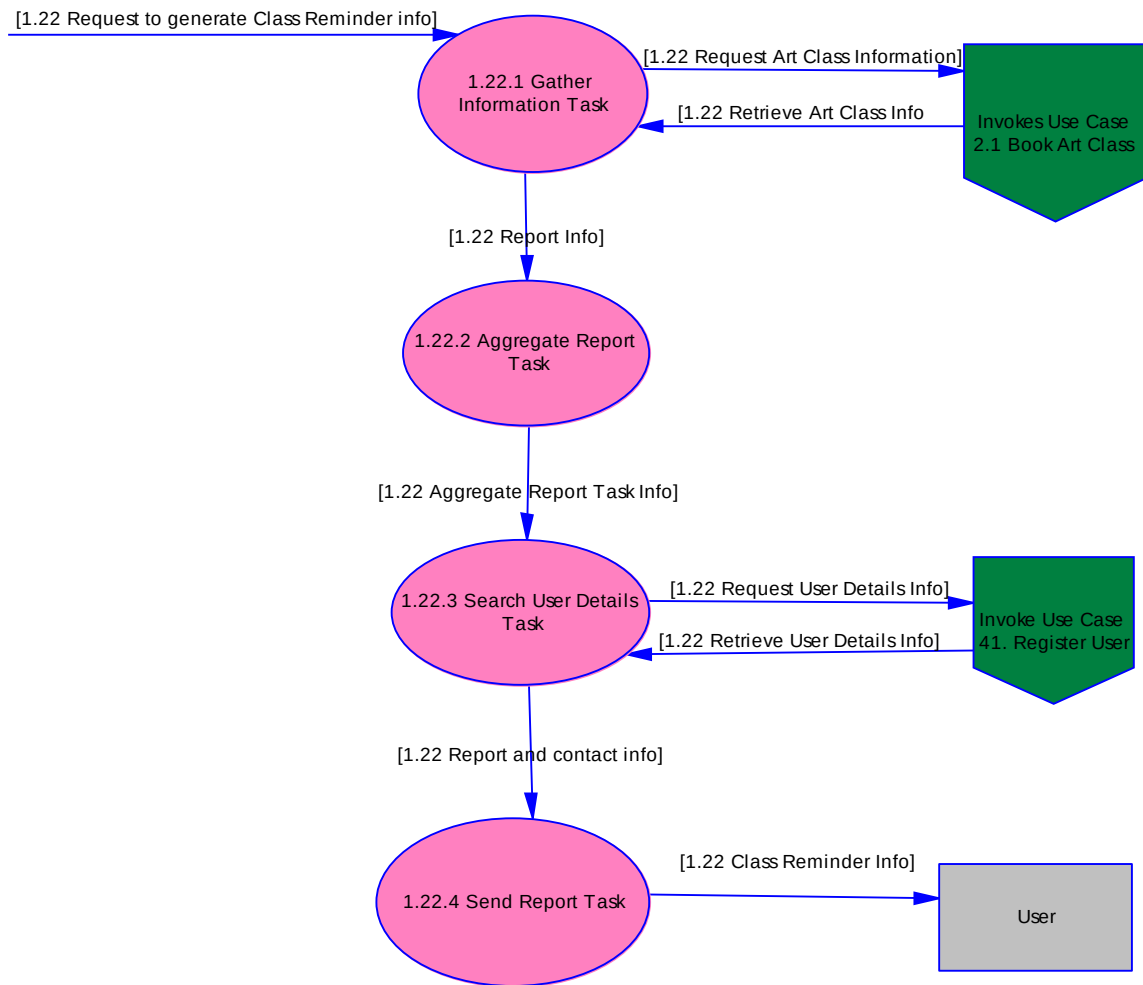


Figure 138: Low Level DFD: 1.22 Generate Payment Reminder

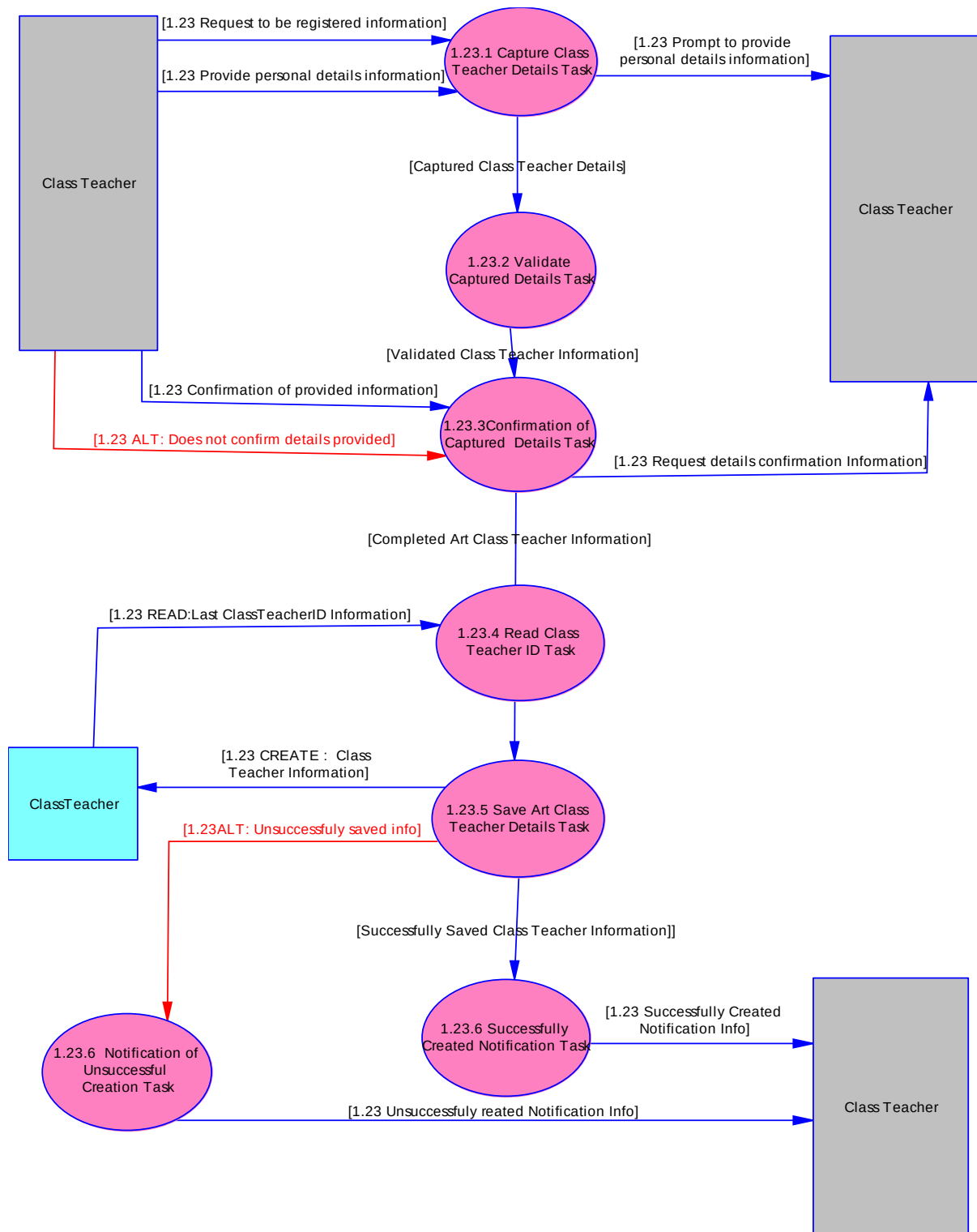


Figure 139: Low Level DFD: 1.23 Create Class Teacher

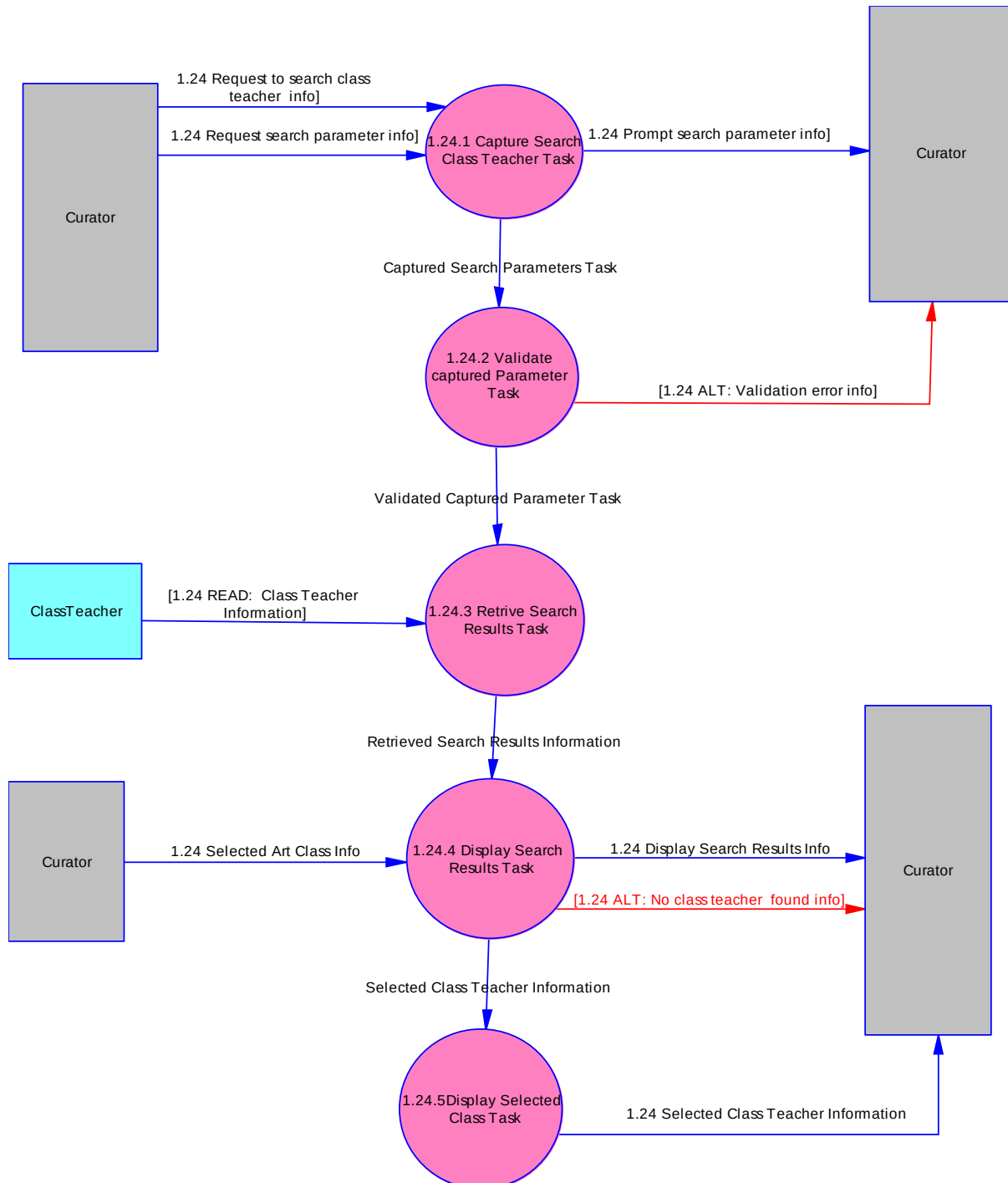


Figure 140: Low Level DFD: 1.24 Search Class Teacher

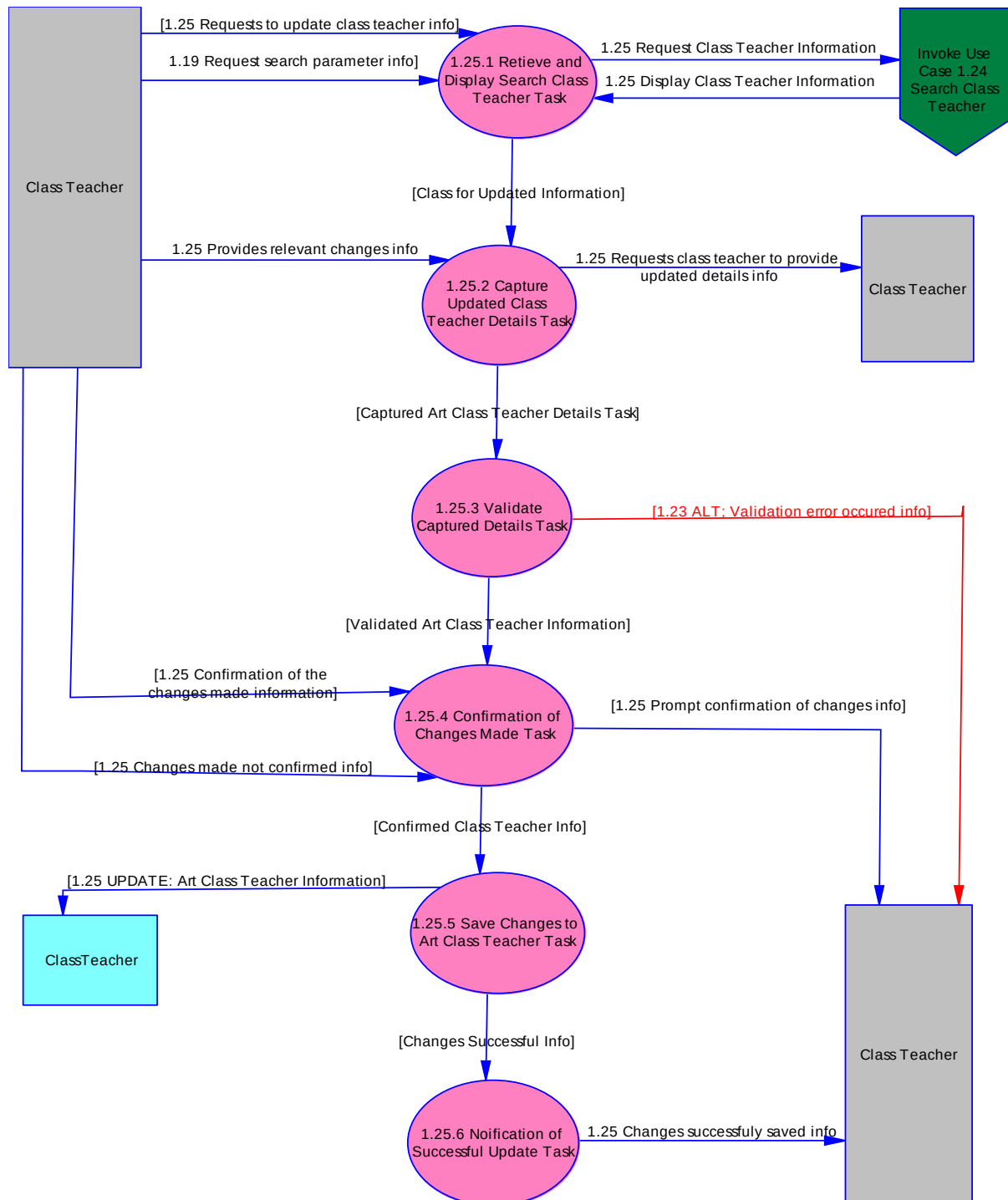


Figure 141: Low Level DFD: 1.25 Update Class Teacher

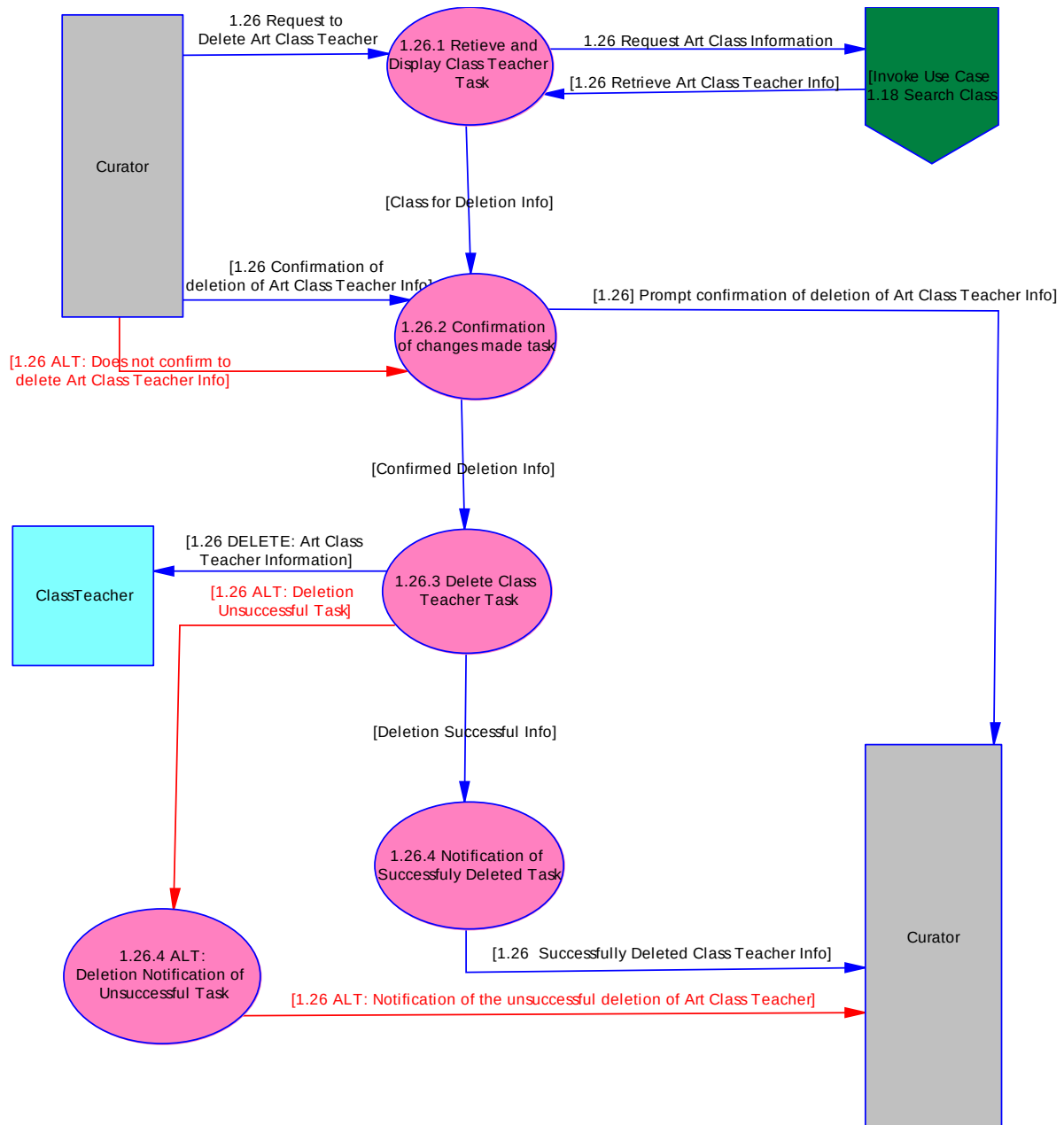


Figure 142: Low Level DFD: 1.26 Delete Class Teacher

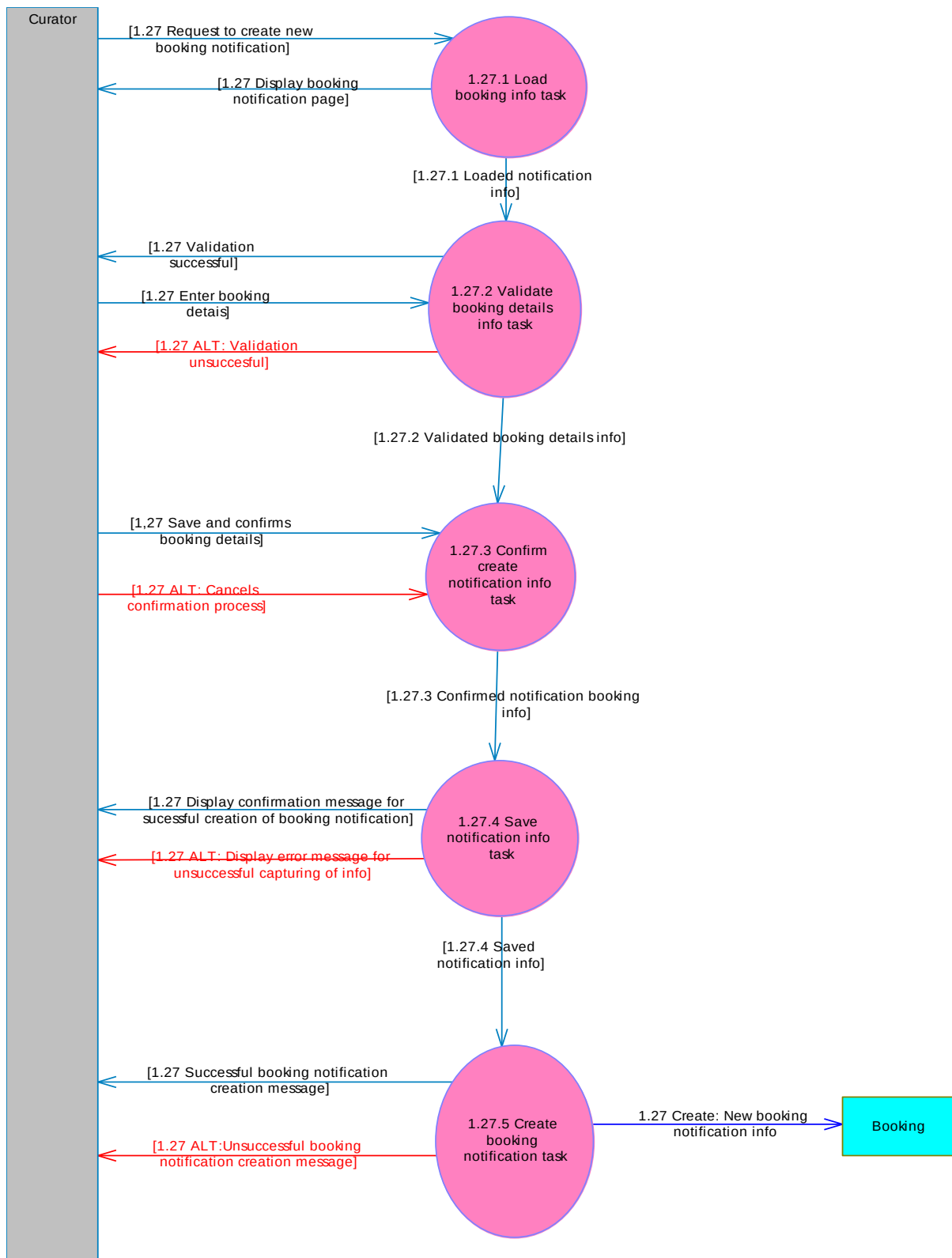


Figure 143: Low Level DFD: 1.27 Generate Booking Notification

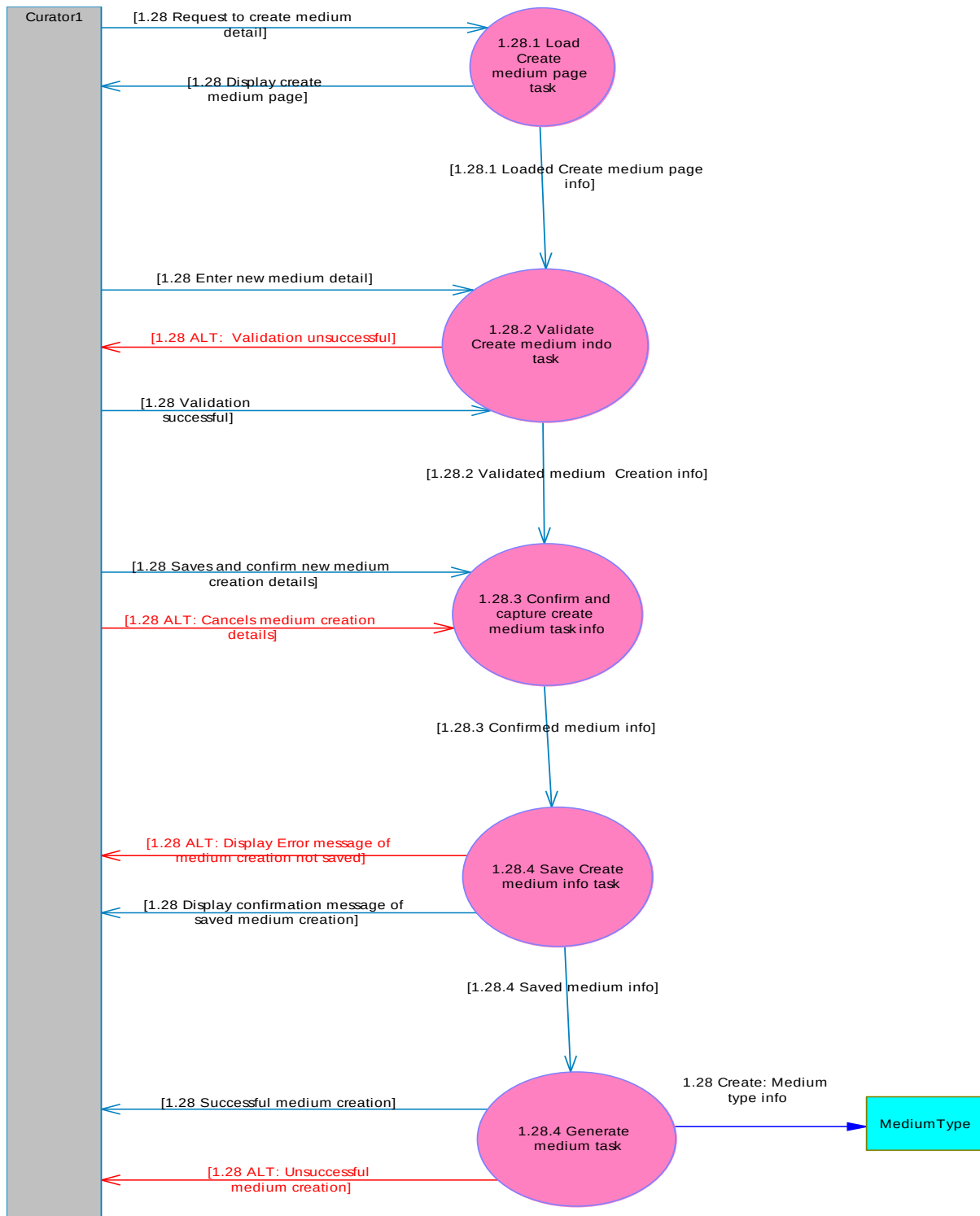


Figure 144: Low Level DFD: 1.28 Create Medium Detail

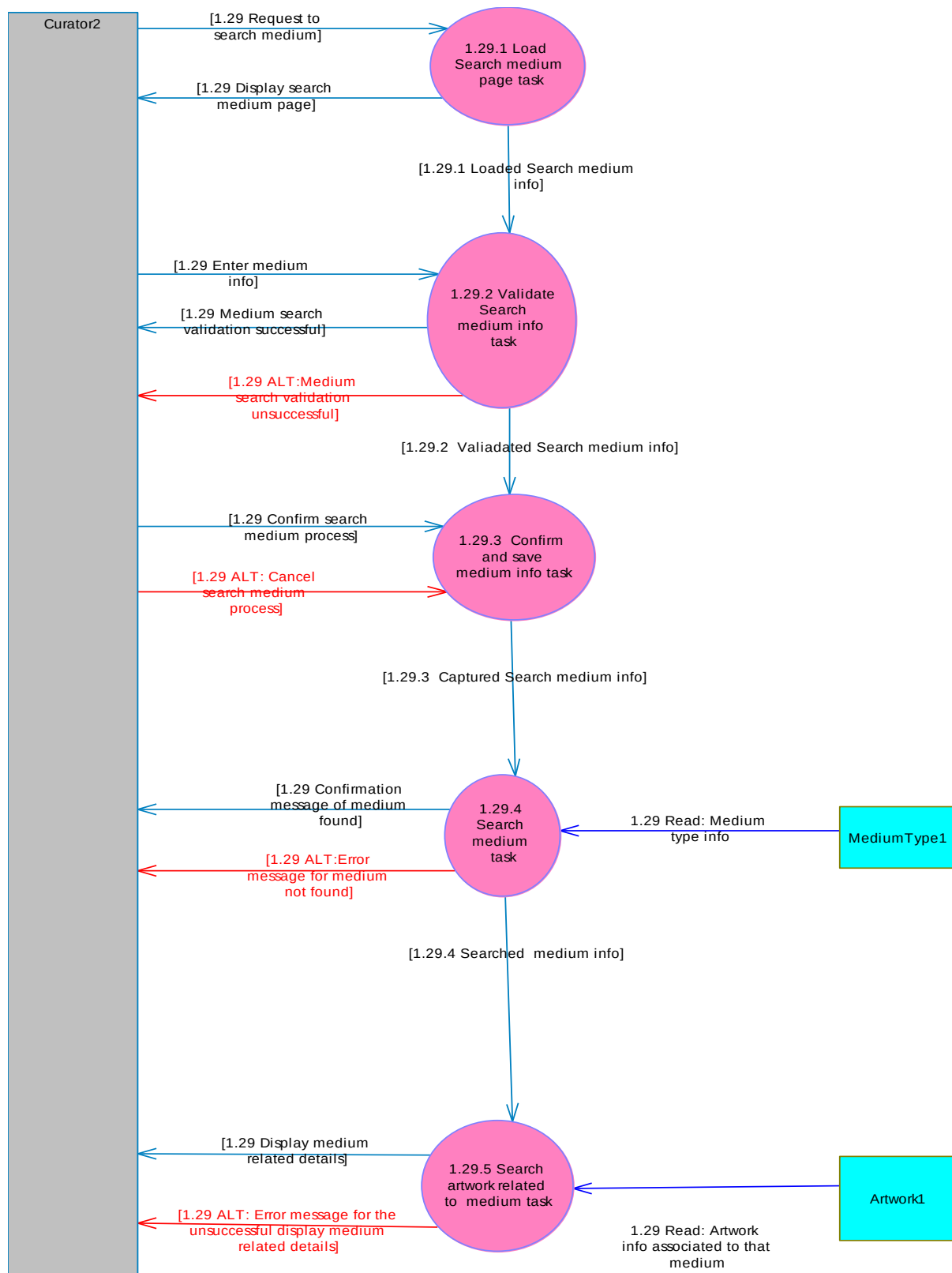


Figure 145: Low Level DFD: 1.29 Update Medium Detail

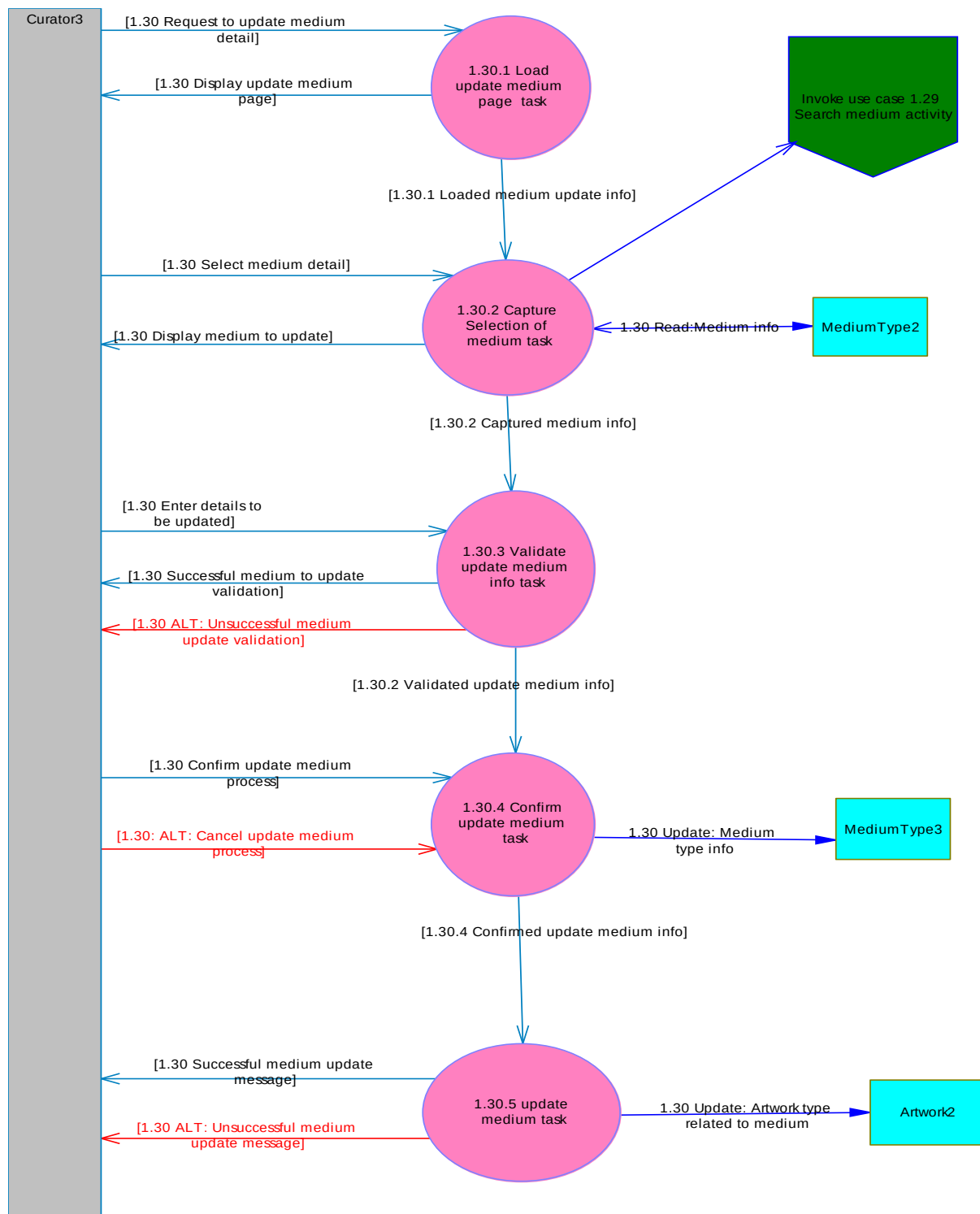


Figure 146: Low Level DFD: 1.30 Update Medium detail

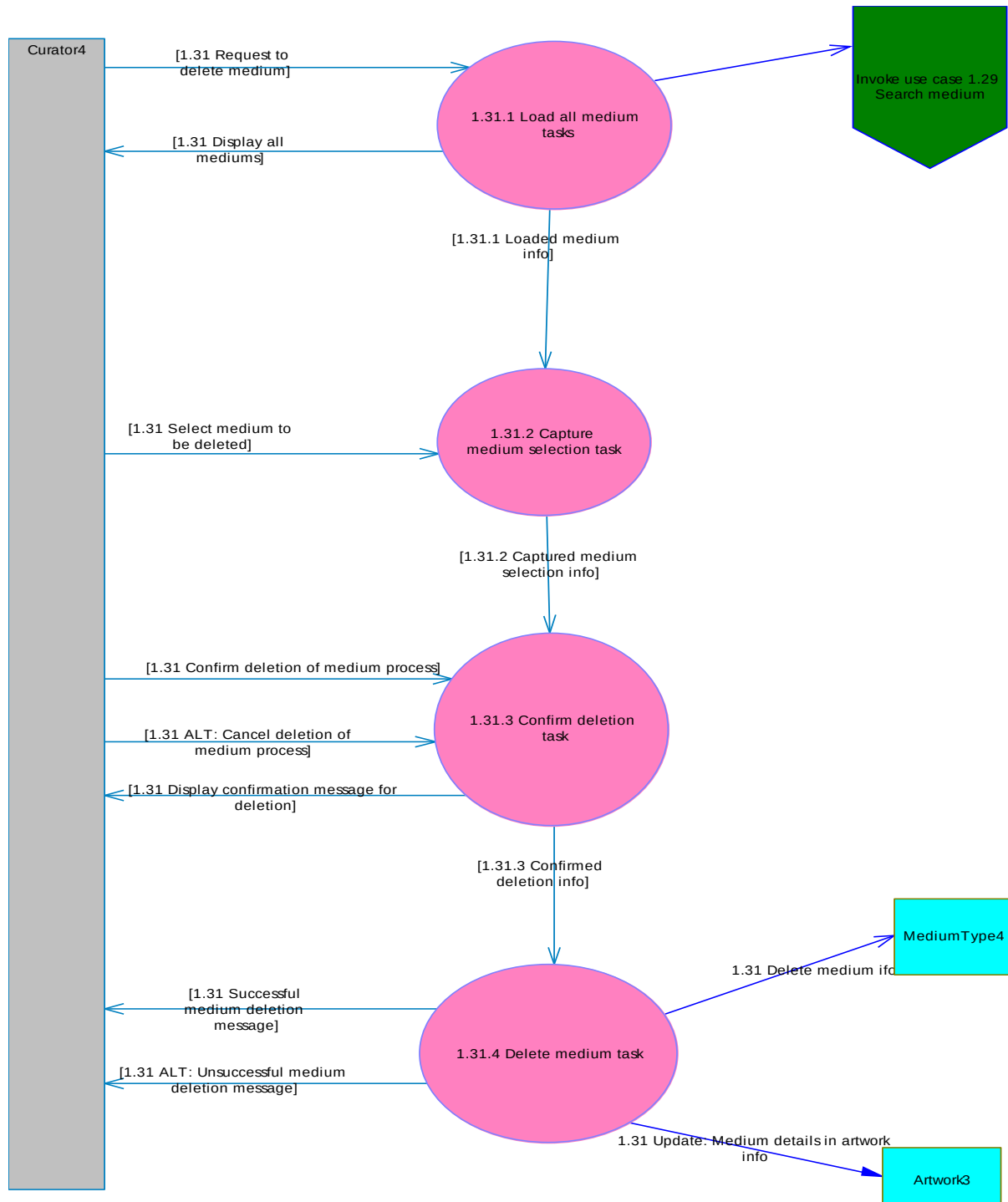


Figure 147: Low Level DFD: 1.31 Delete Medium

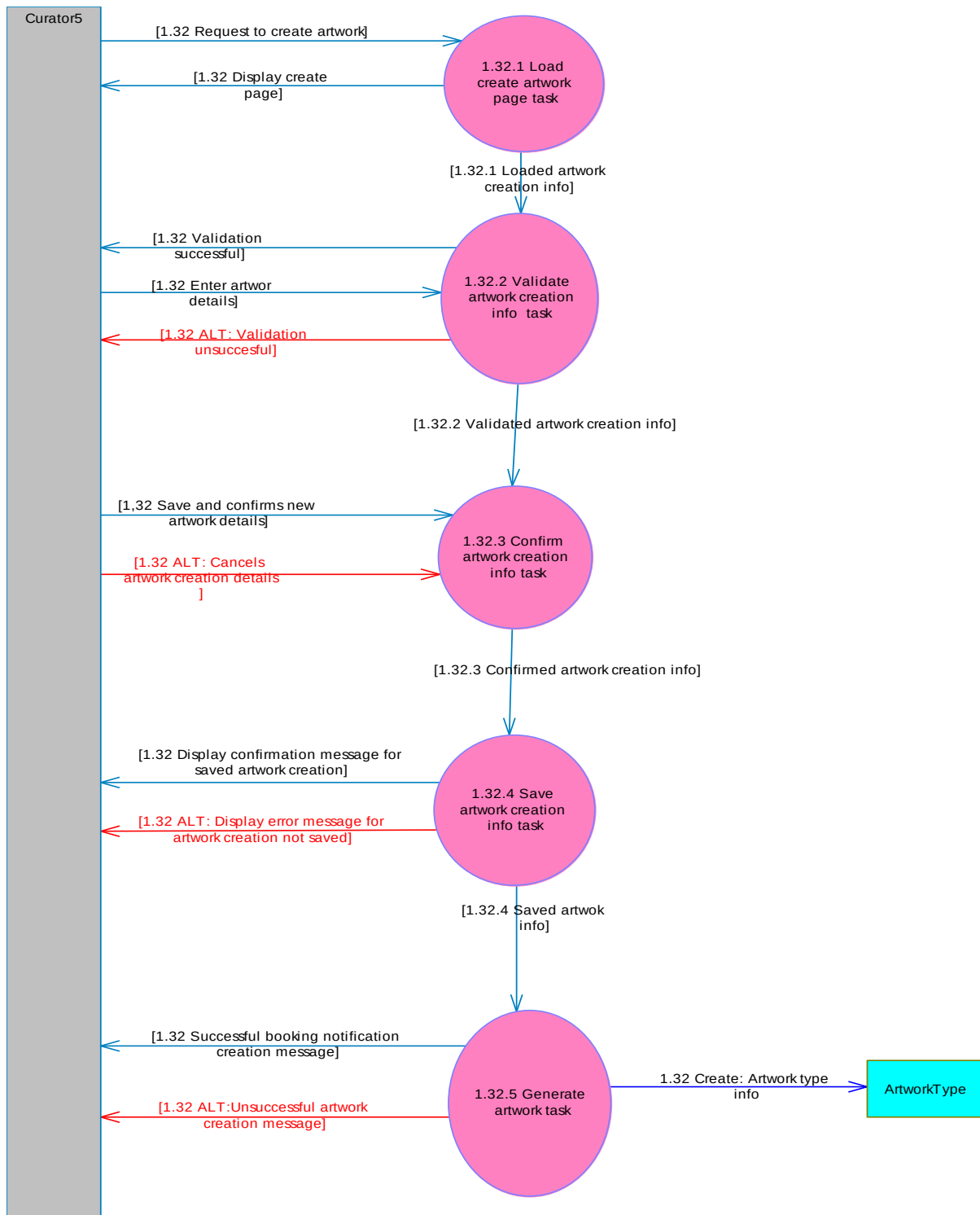


Figure 148: Low Level DFD: 1.32 Create Artwork Detail

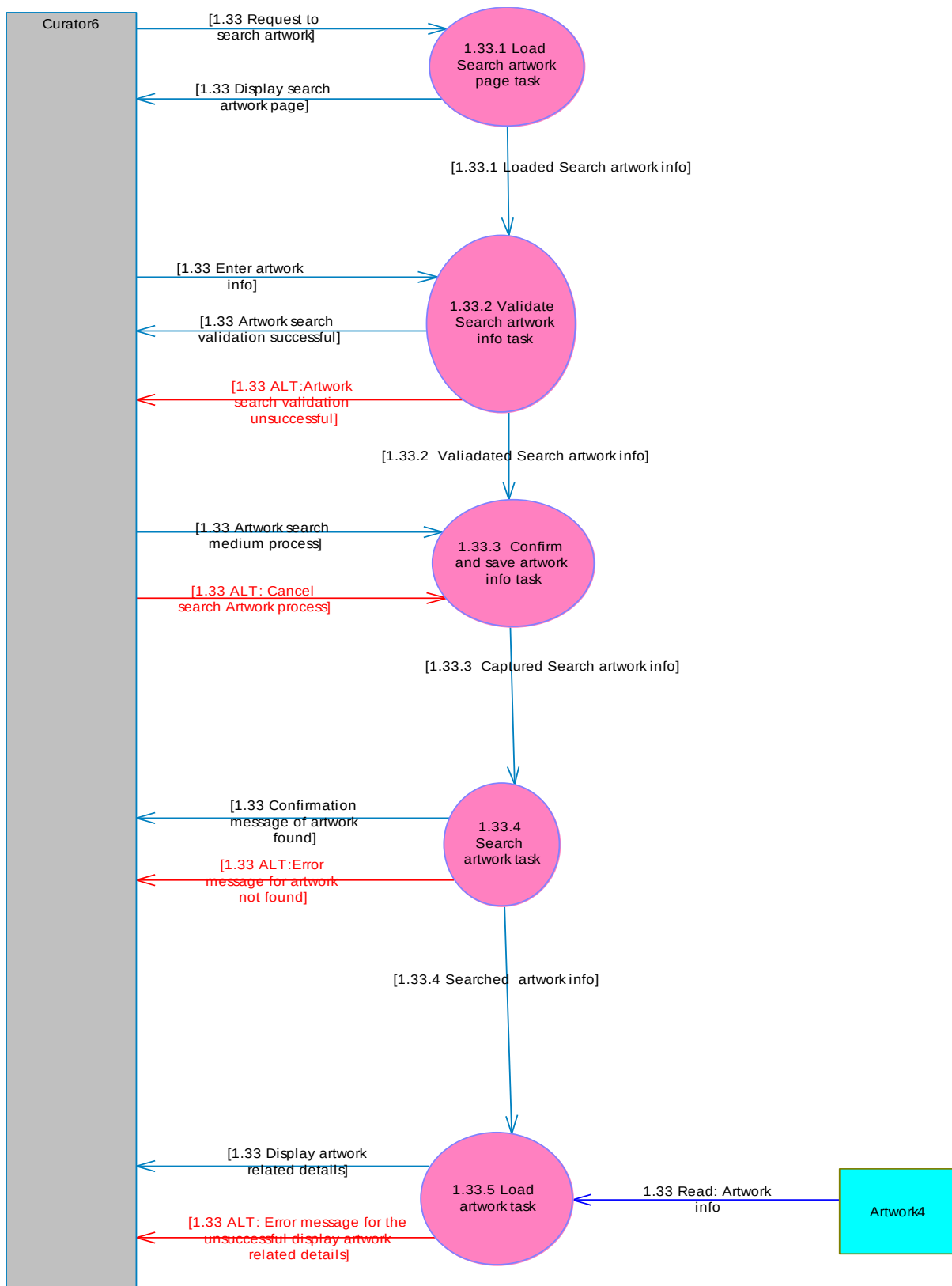


Figure 149: Low Level DFD: 1.33 Search Artwork

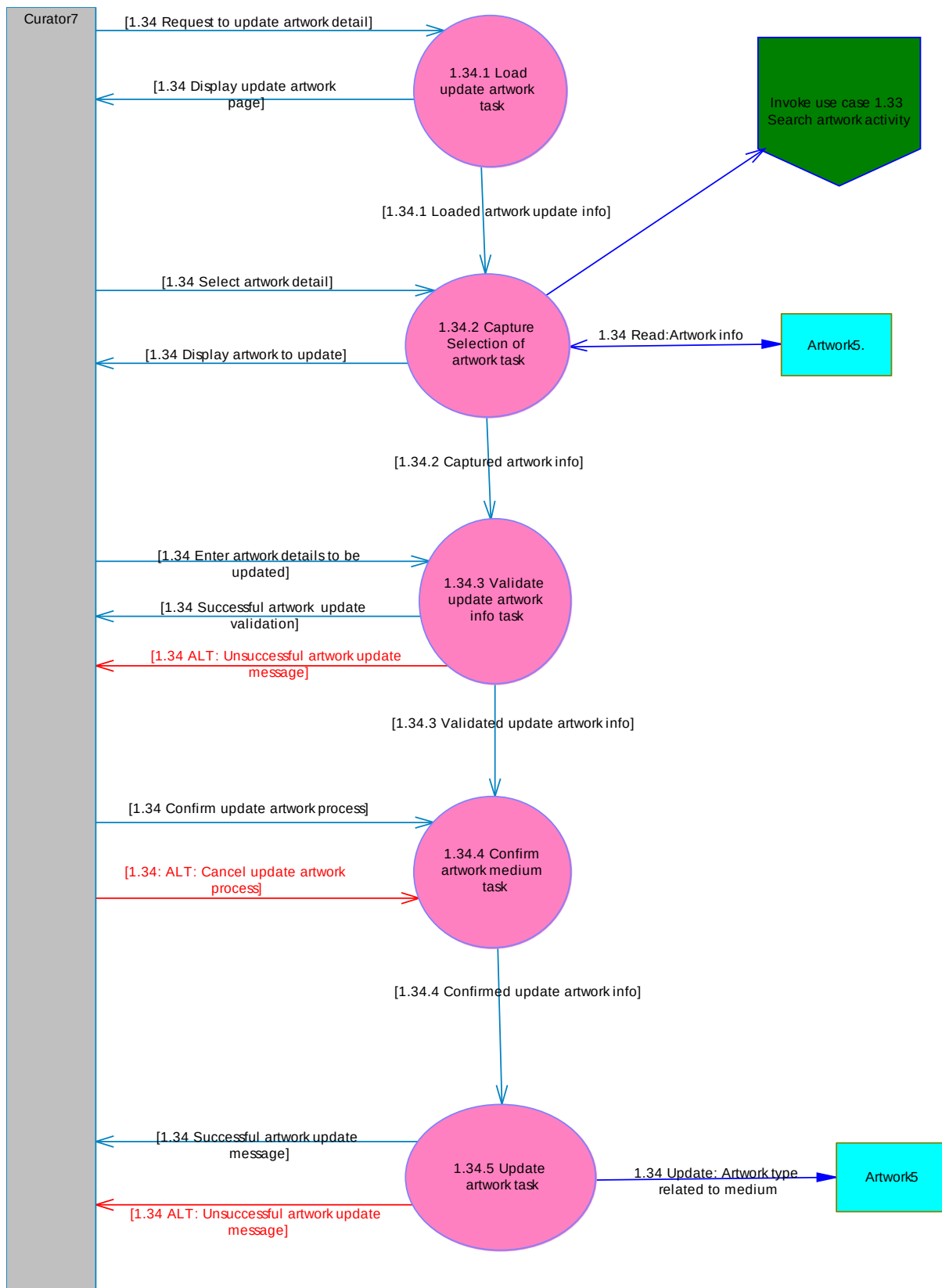


Figure 150: Low Level DFD: 1.34 Update Artwork

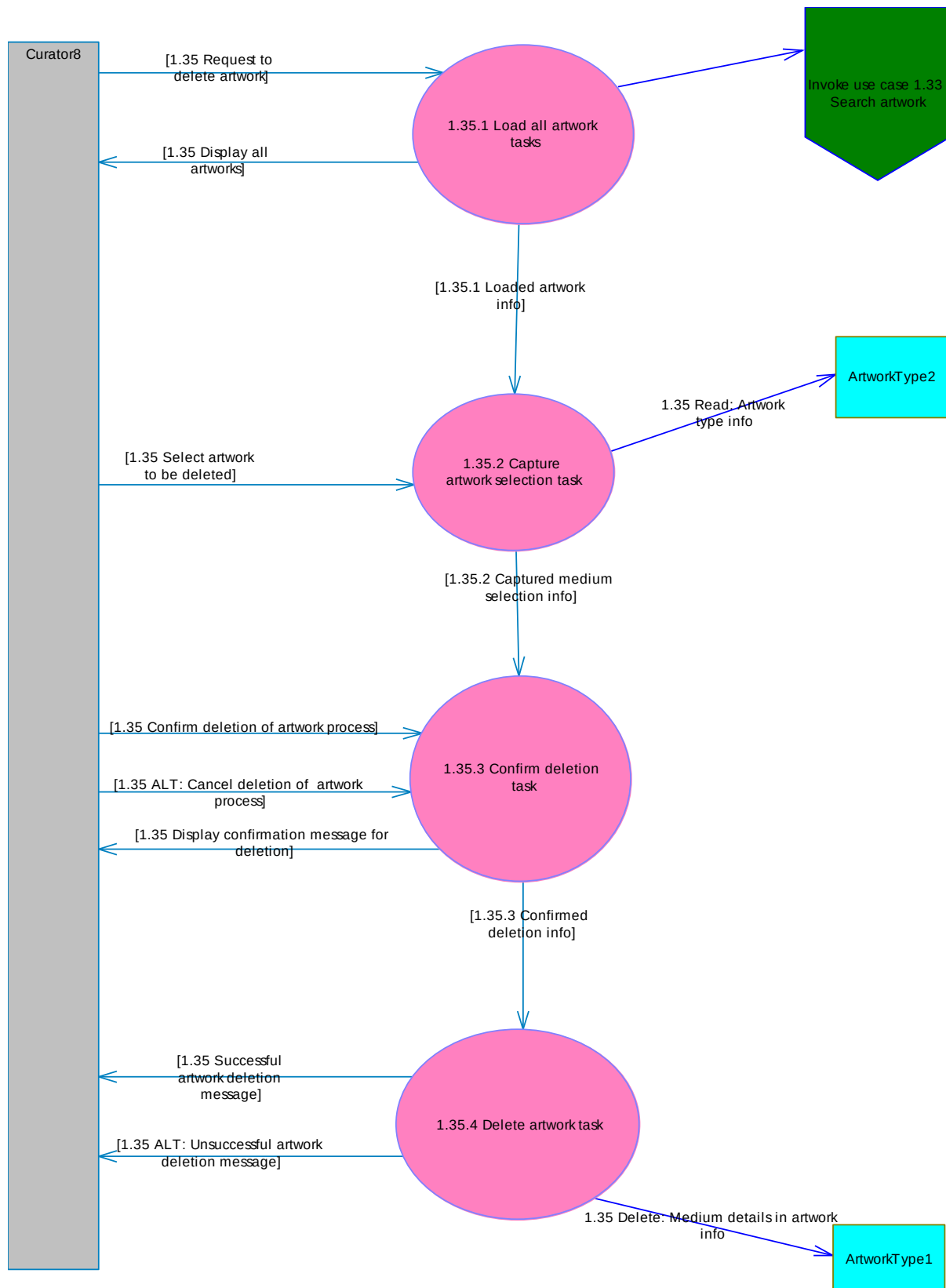


Figure 151: Low Level DFD: 1.35 Delete Artwork

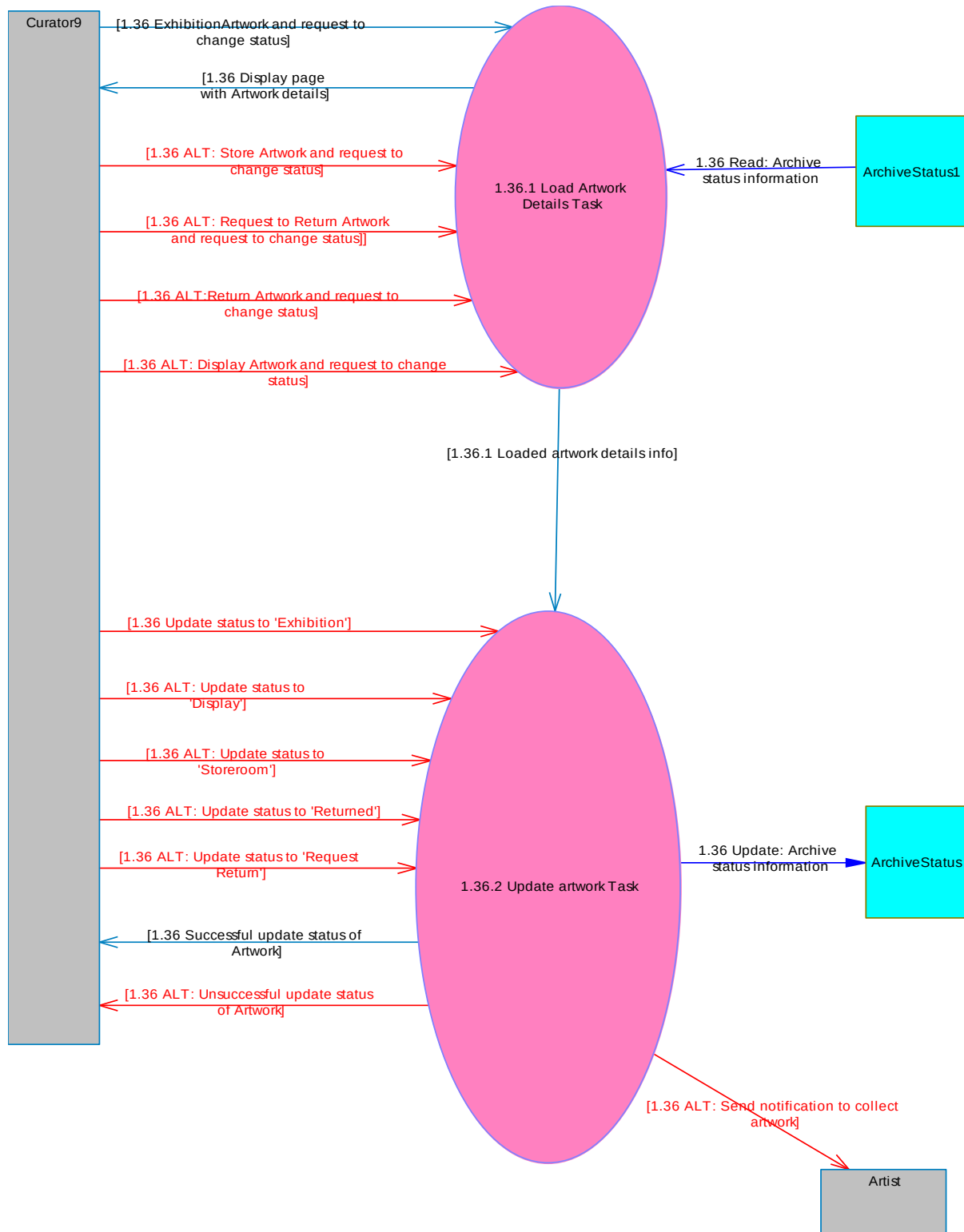


Figure 152: Low Level DFD: 1.36 Archive Status

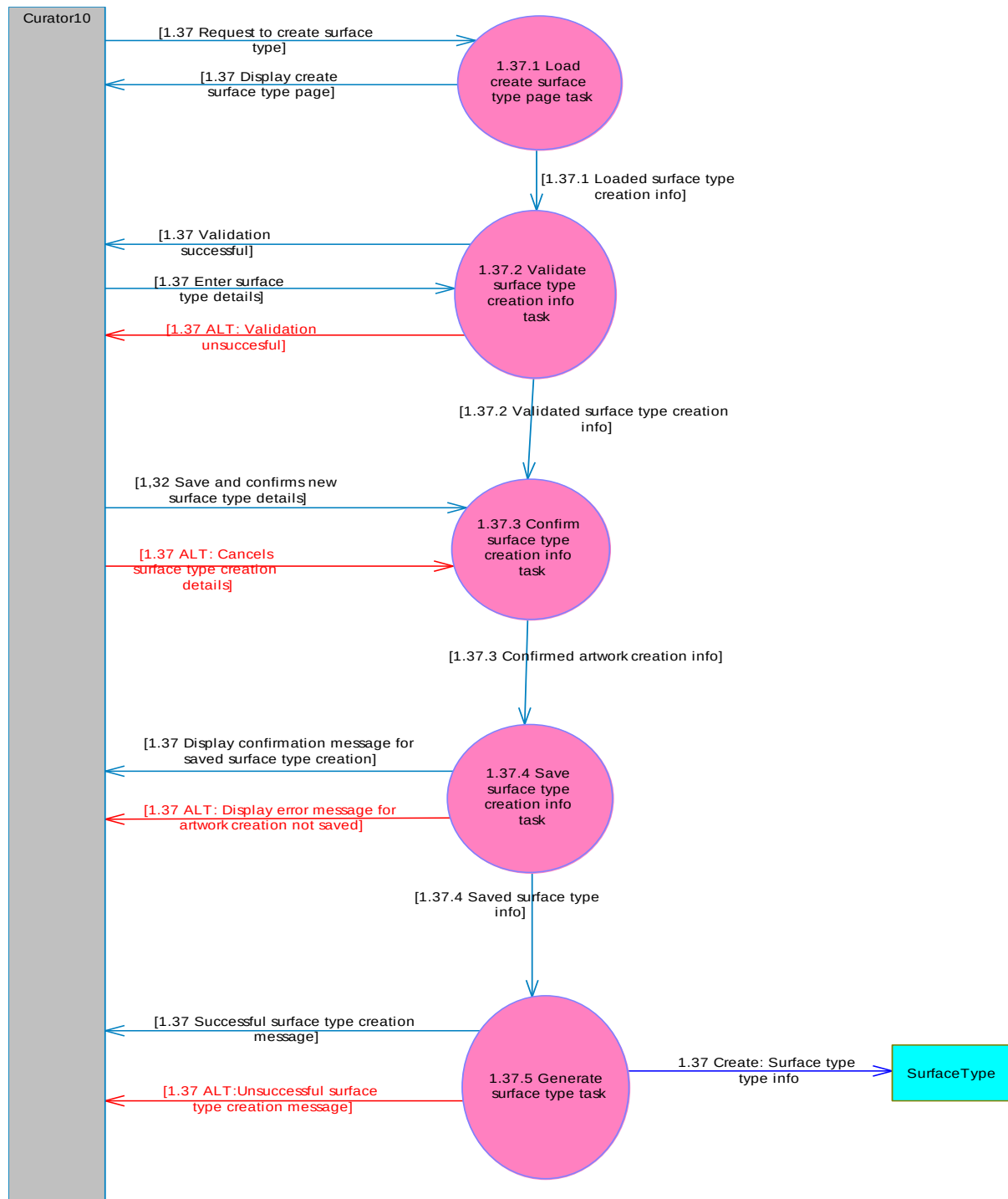


Figure 153: Low Level DFD: 1.37 Create Surface type

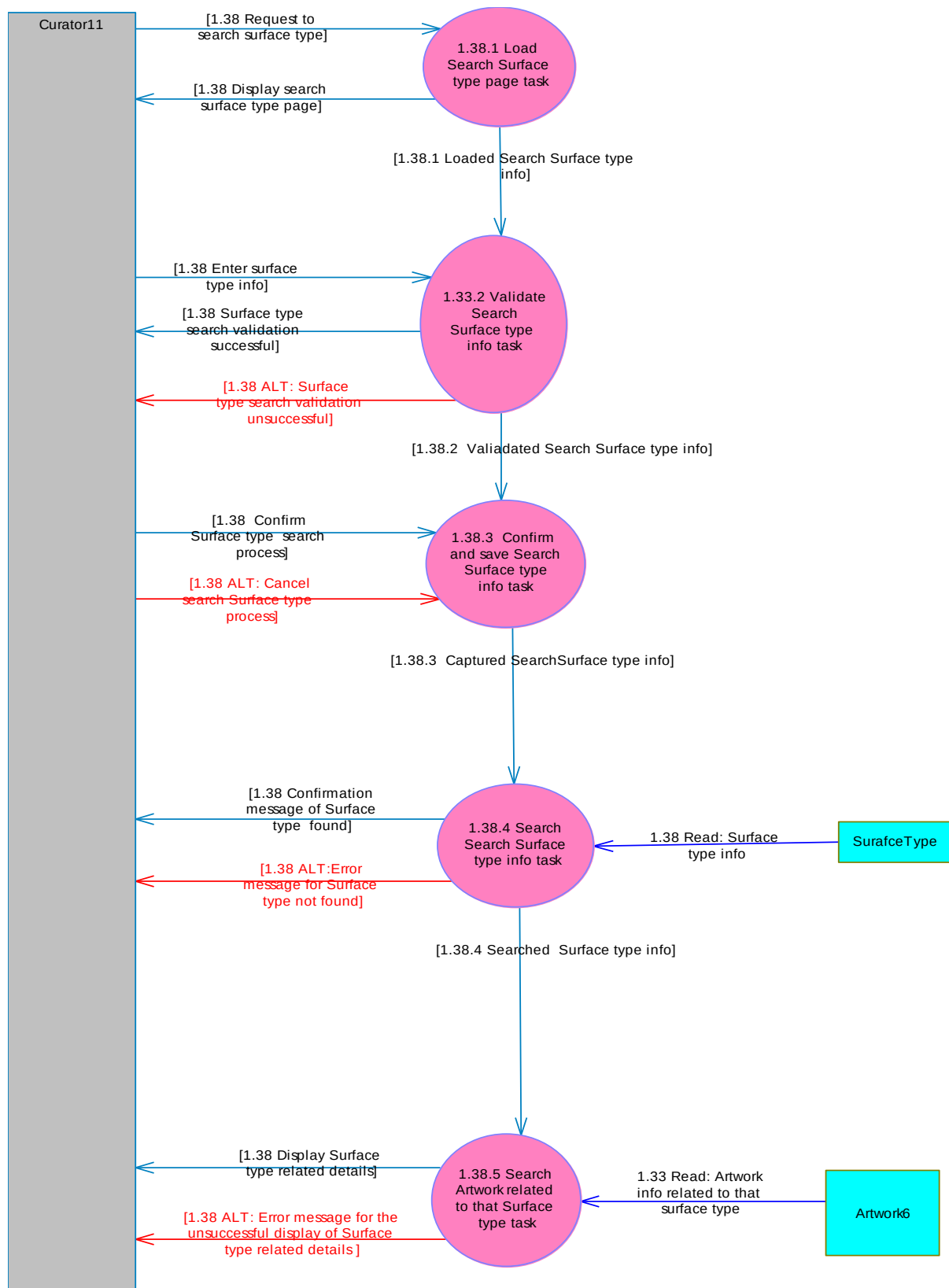


Figure 154: Low Level DFD: 1.38 Search surface type

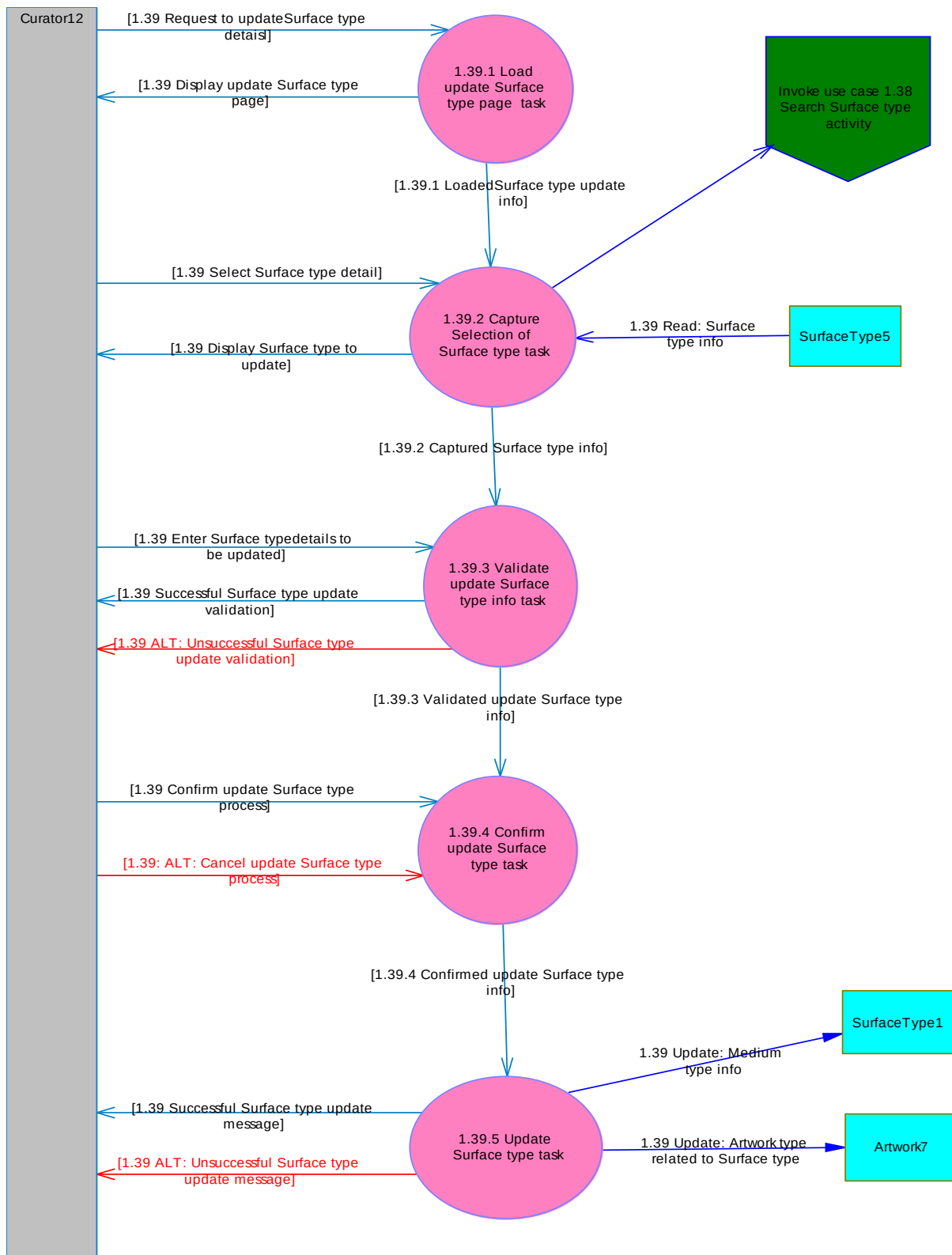


Figure 155: Low Level DFD: 1.39 Update Surface detail

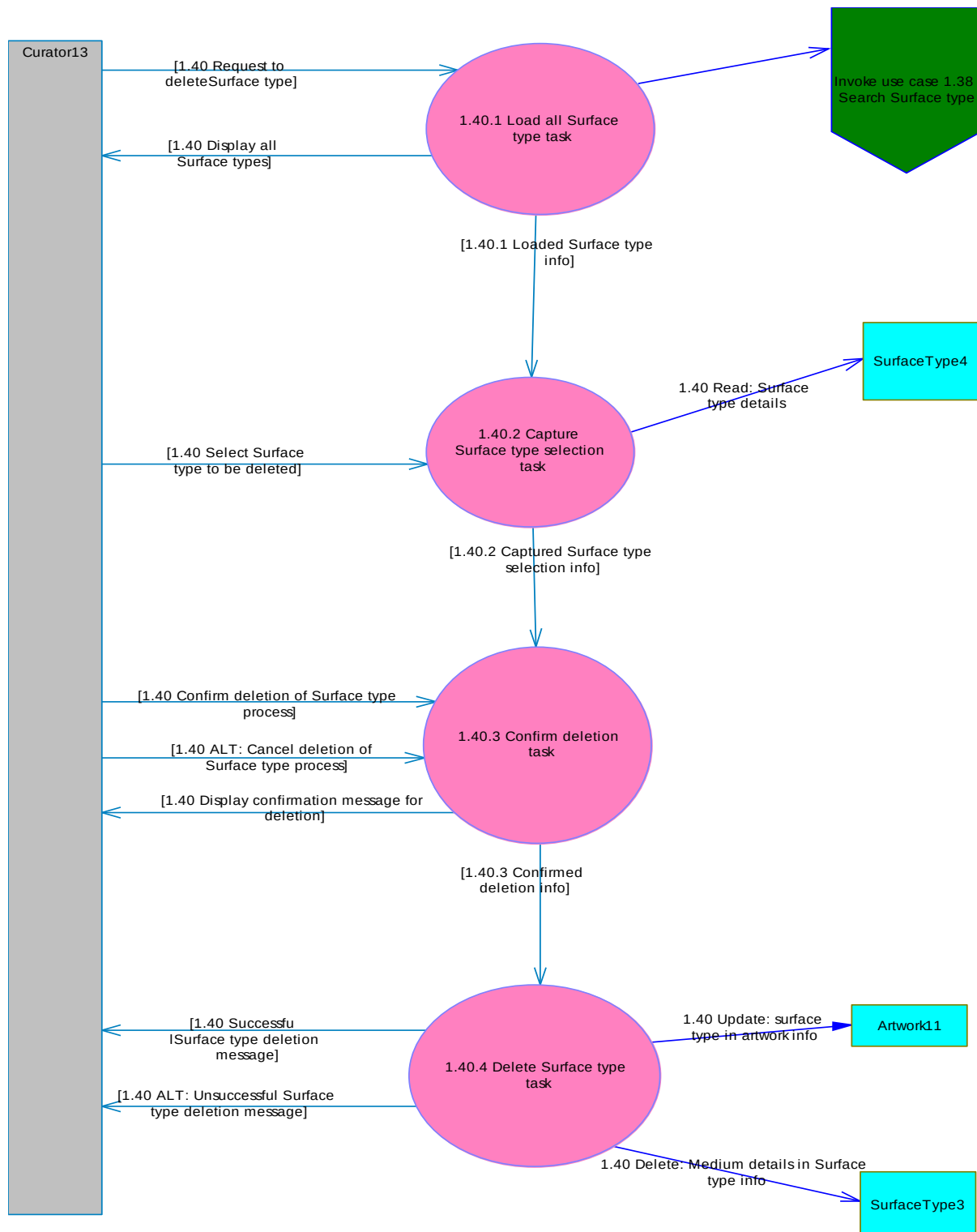


Figure 156: Low Level DFD: 1.40 Delete Surface type

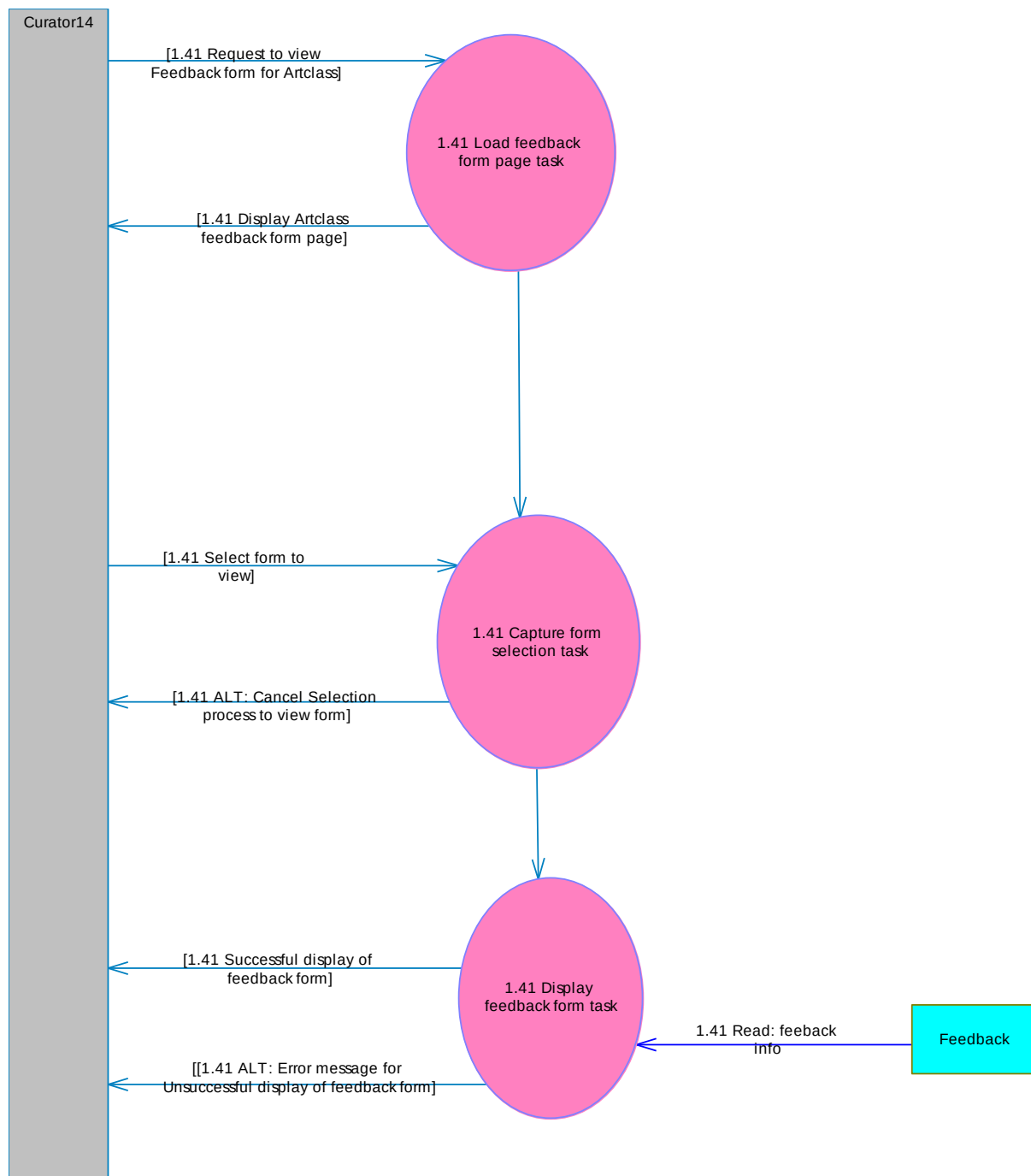


Figure 157: Low Level DFD: 1.41 View Art Class Feedback form

3.3.3.2 Art Class Subsystem

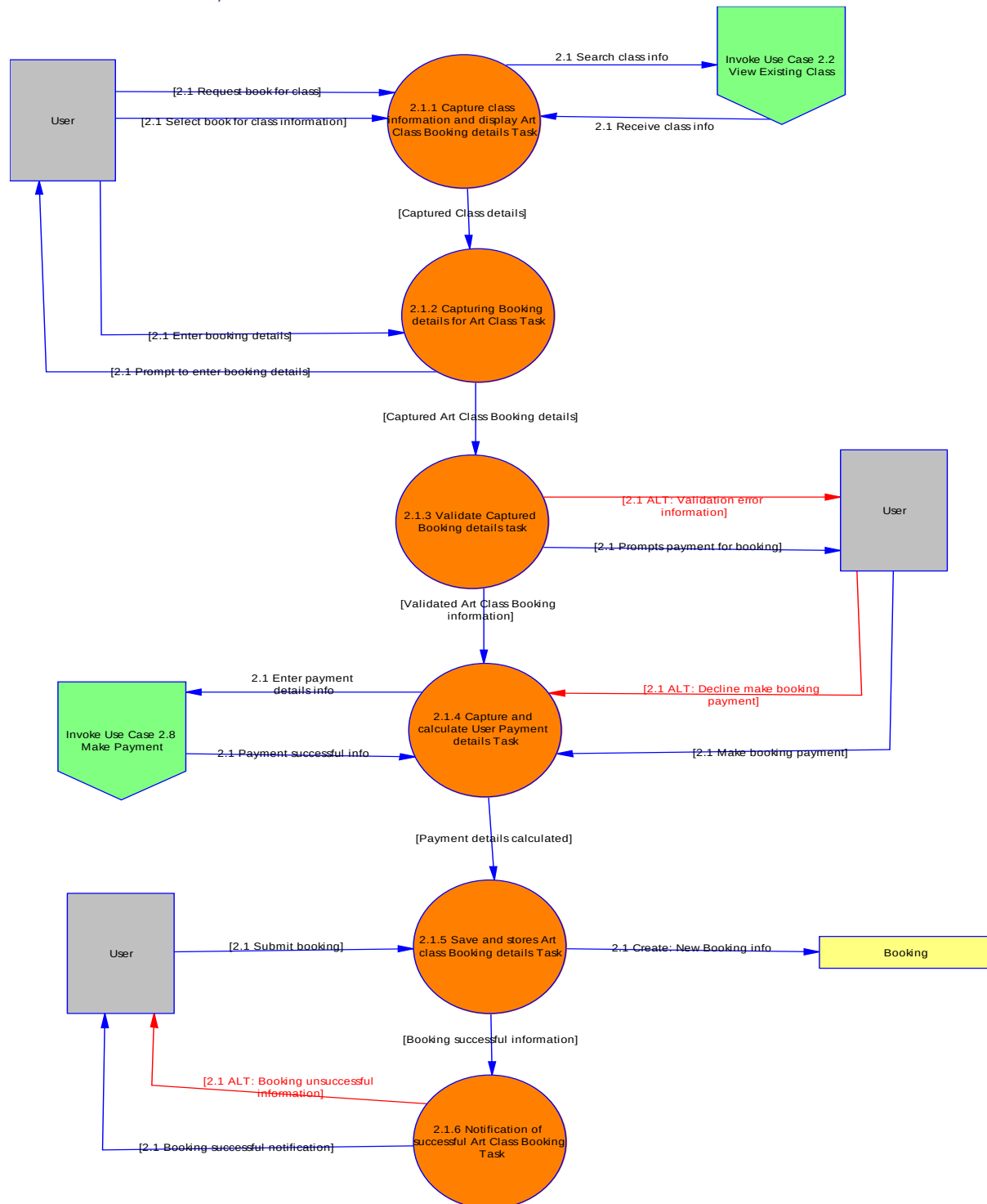


Figure 158: Low Level DFD: 2.1 Book Art Class.

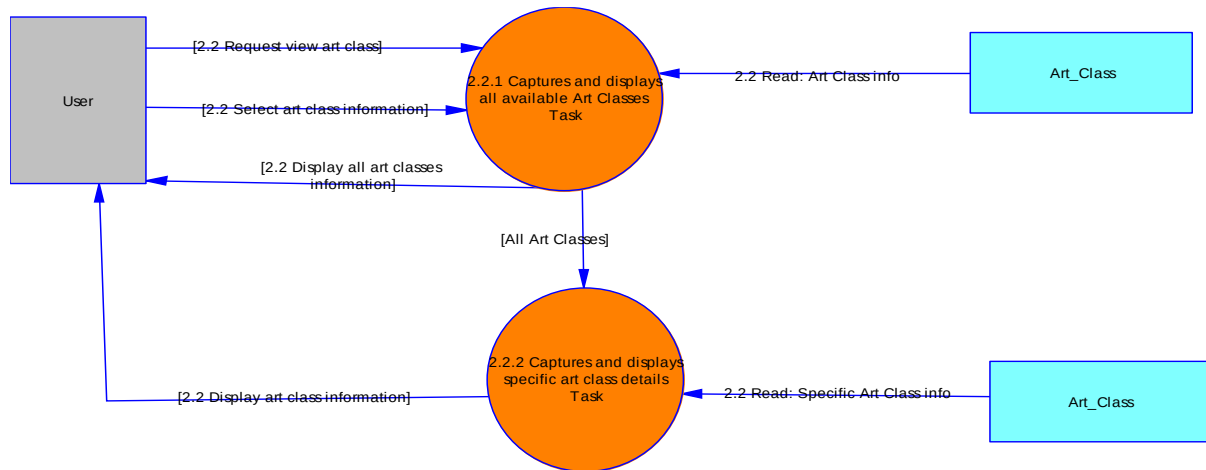


Figure 159: Low Level DFD: 2.2 View Existing Classes

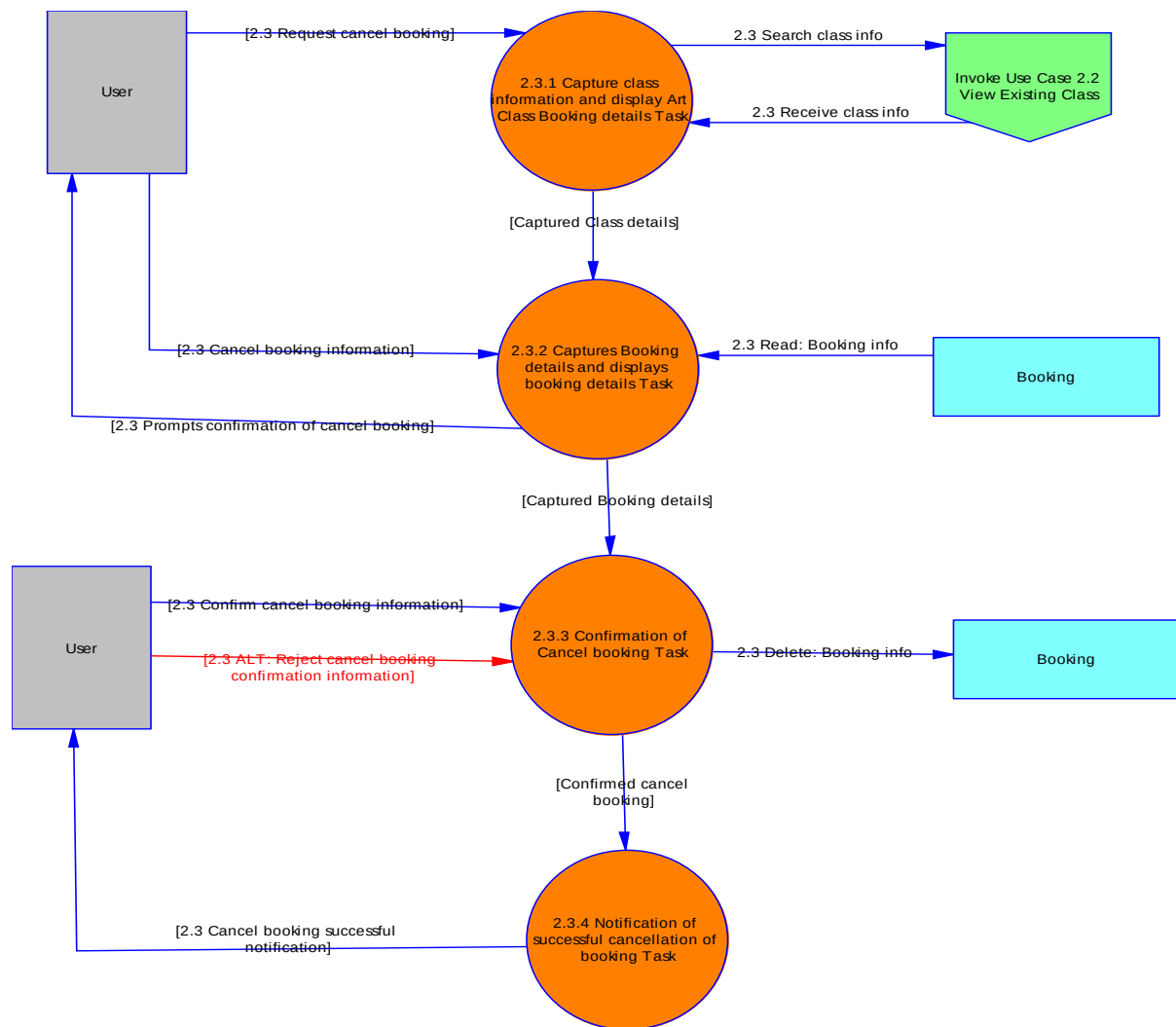


Figure 160: Low Level DFD: 2.3 Cancel Booking

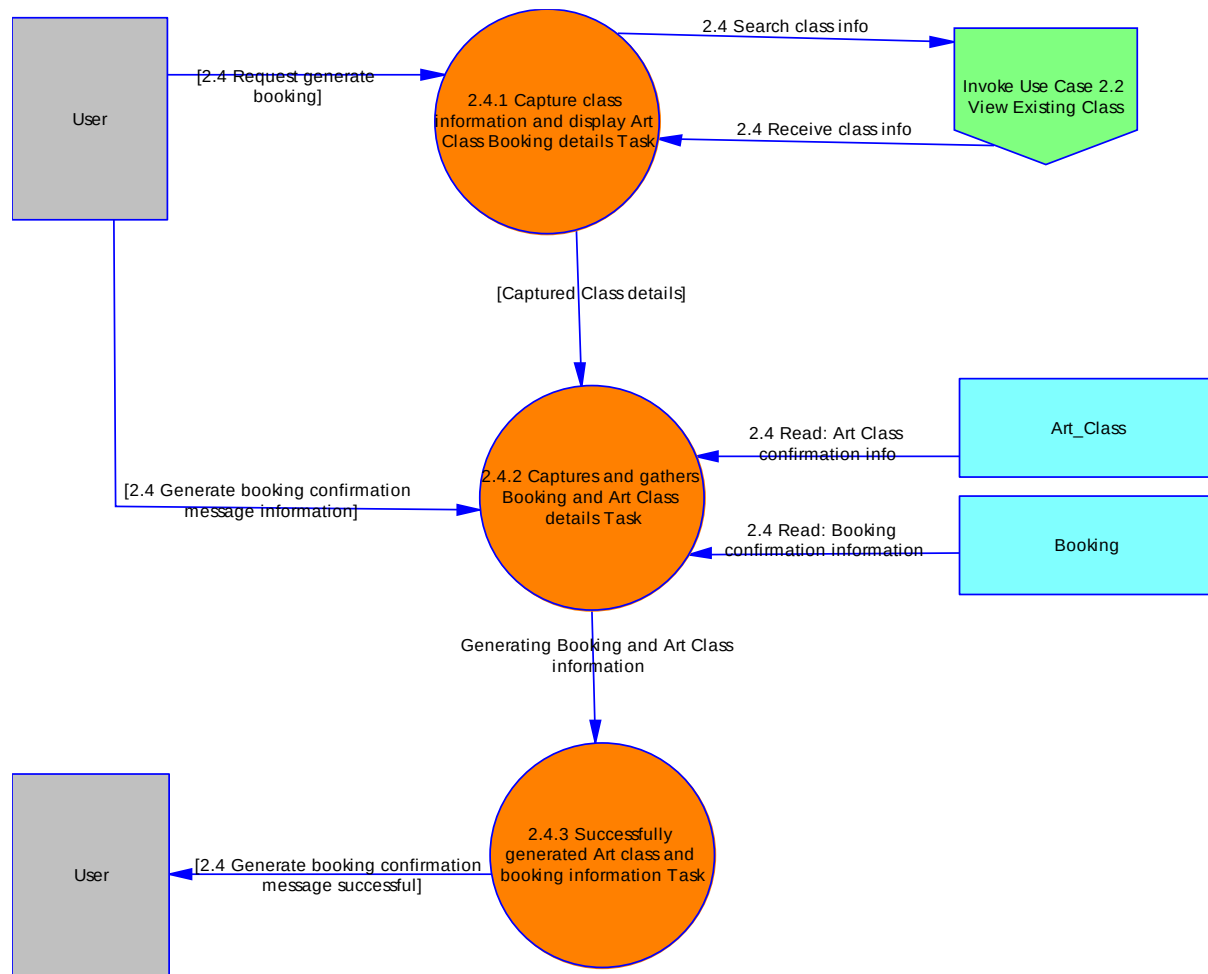


Figure 161: Low Level DFD: 2.4 Generate Booking Confirmation

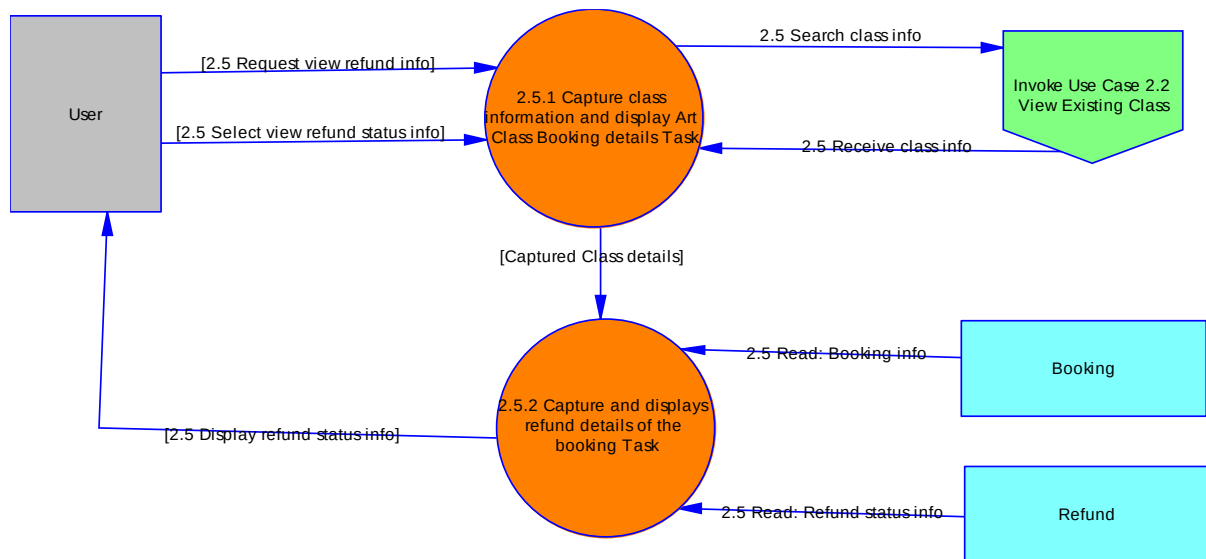


Figure 162: Low Level DFD: 2.5 Refund Status

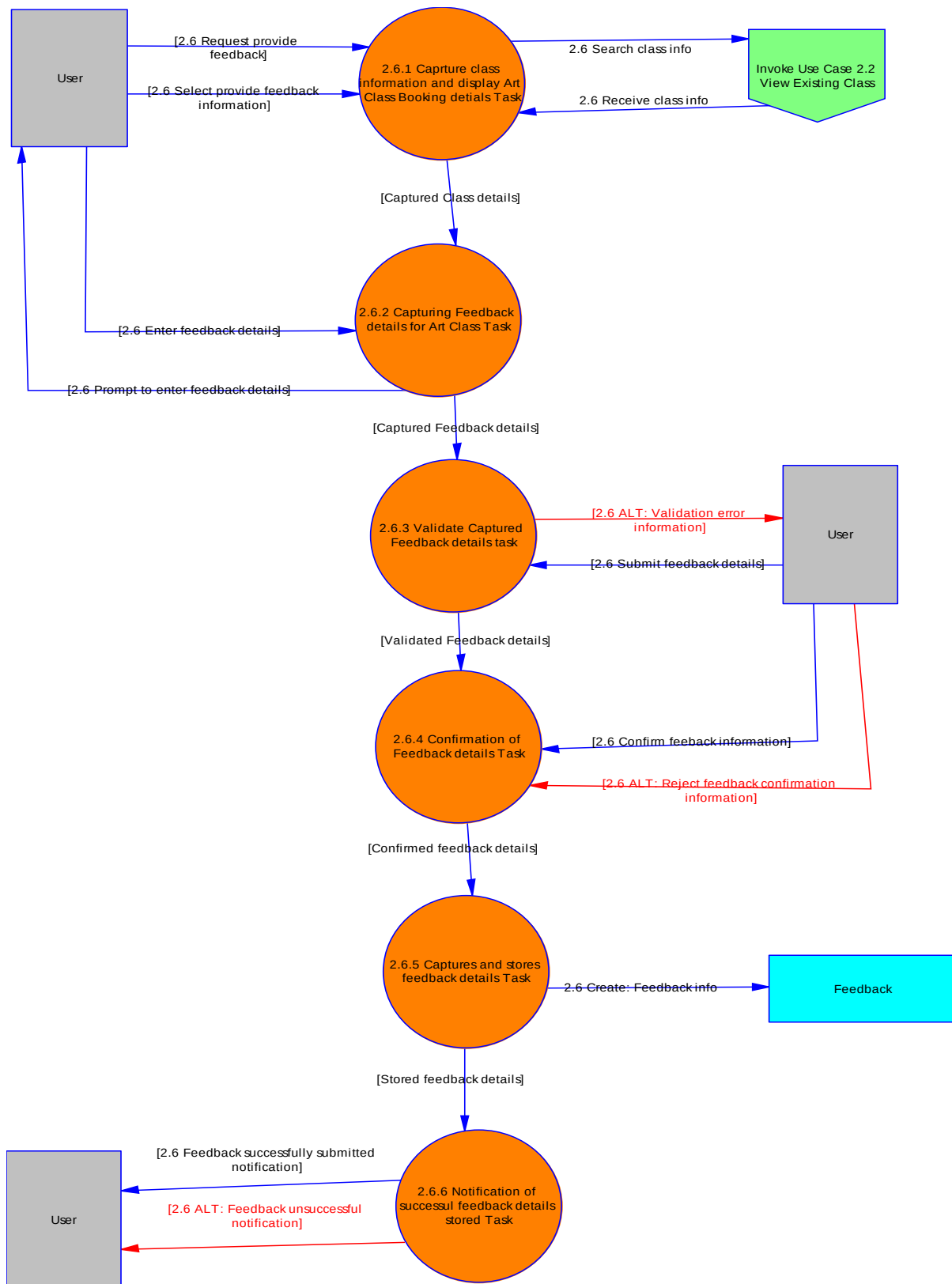


Figure 163: Low Level DFD: 2.6 Capture Class Feedback

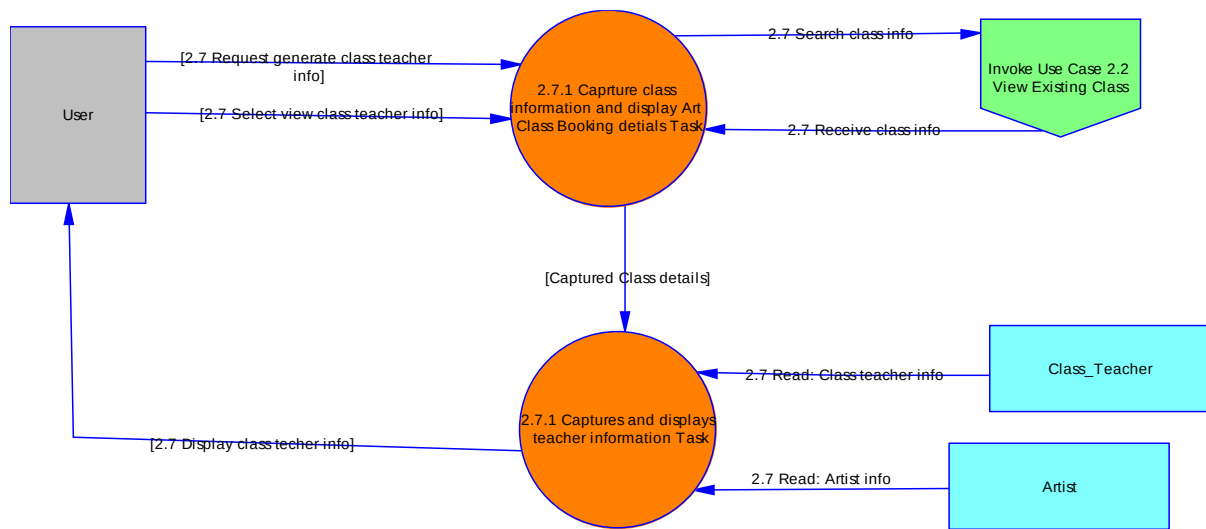


Figure 164: Low Level DFD: 2.7 Generate Class Teacher

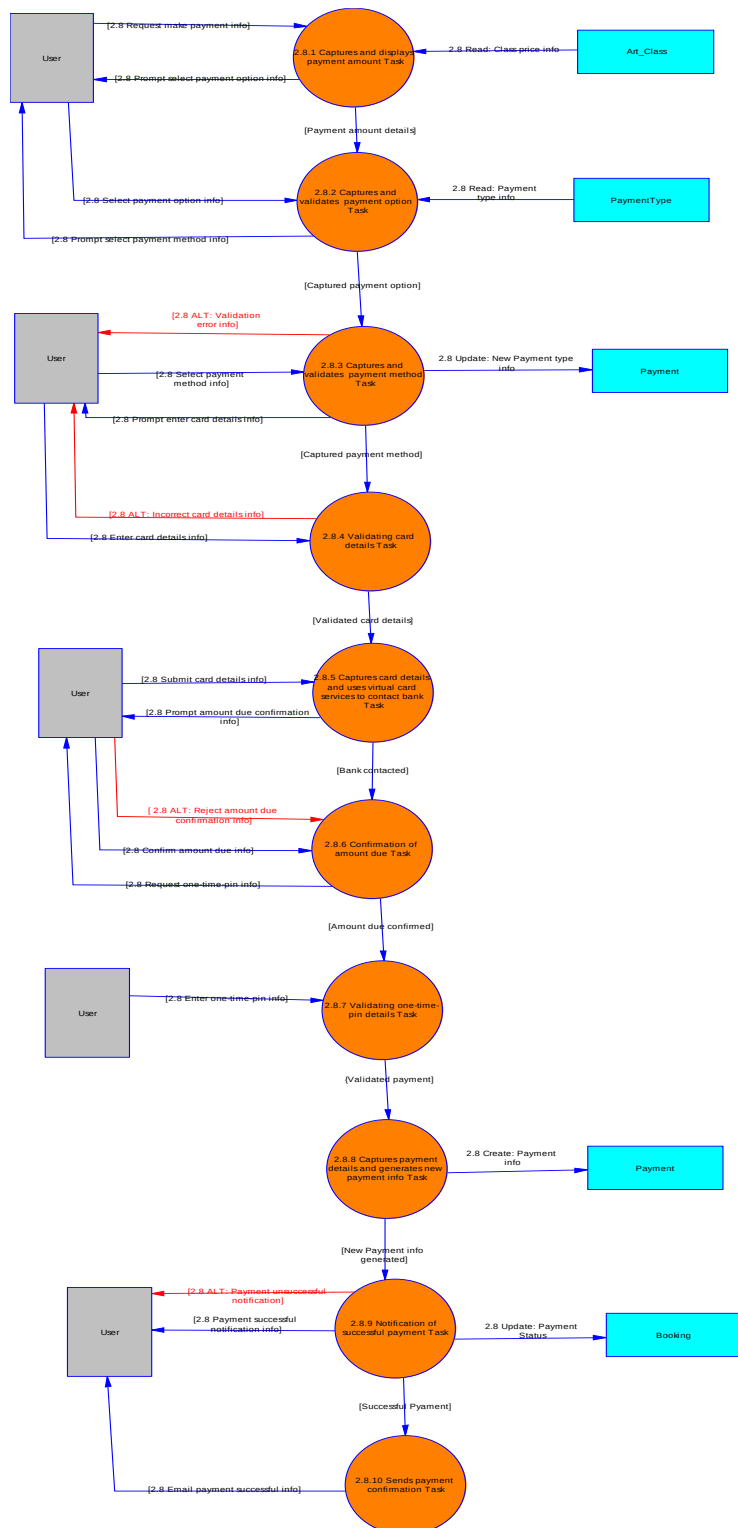


Figure 165: Low Level DFD: 2.8 Make Payment

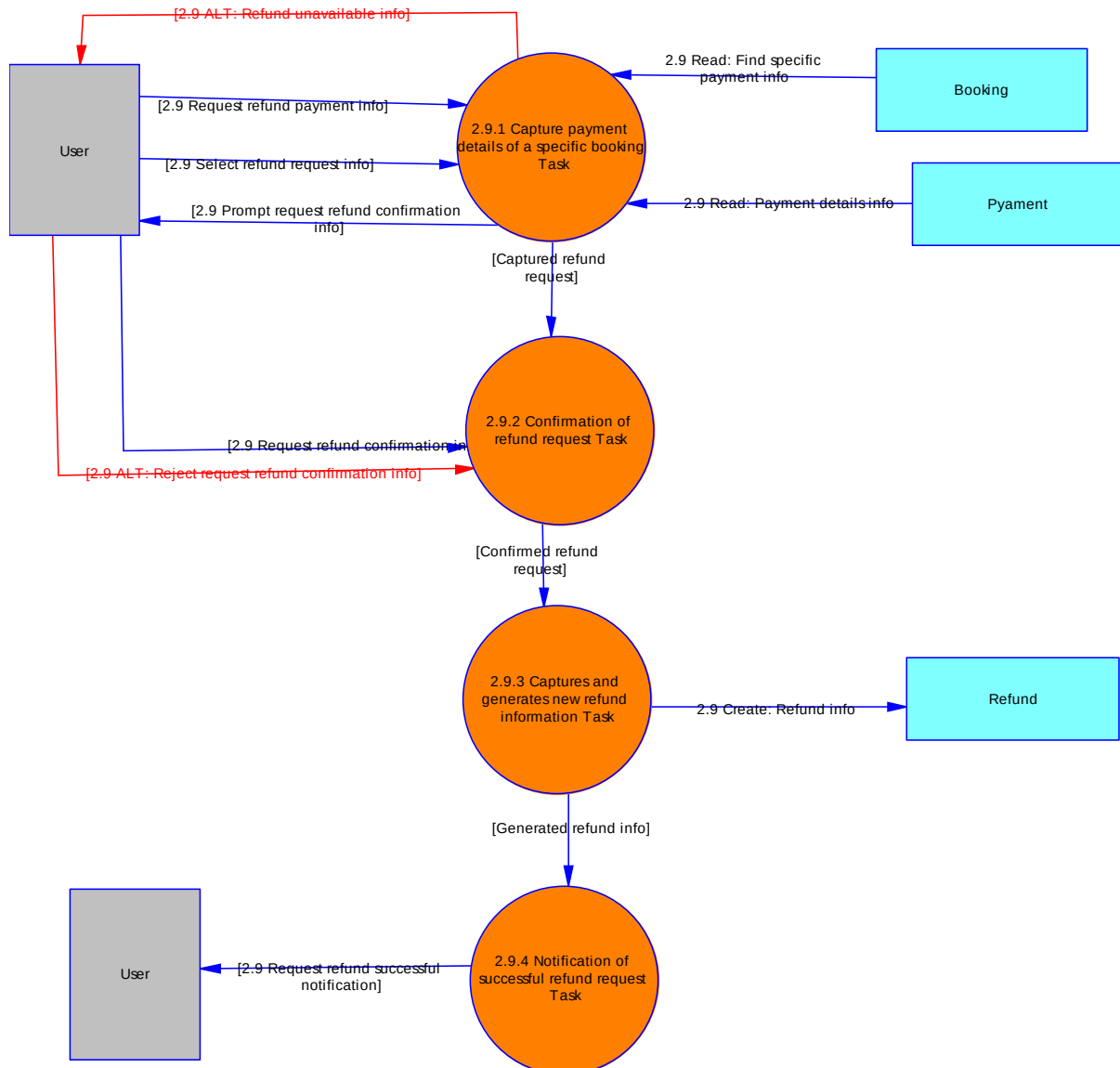


Figure 166: Low Level DFD: 2.9 Request Refund

3.3.3.3 Reporting Subsystem

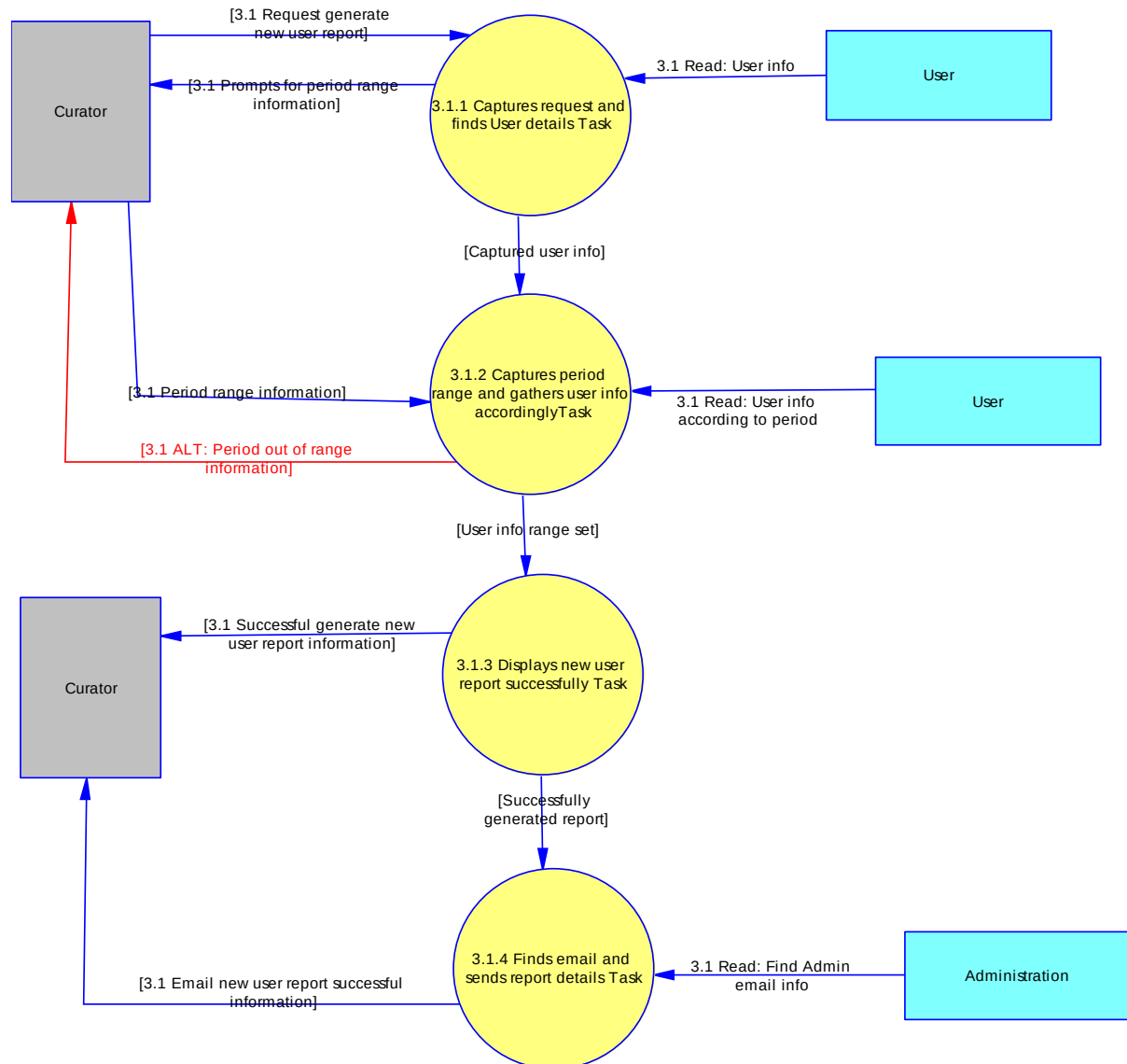


Figure 167: Low Level DFD: 3.1 Generate New User Report

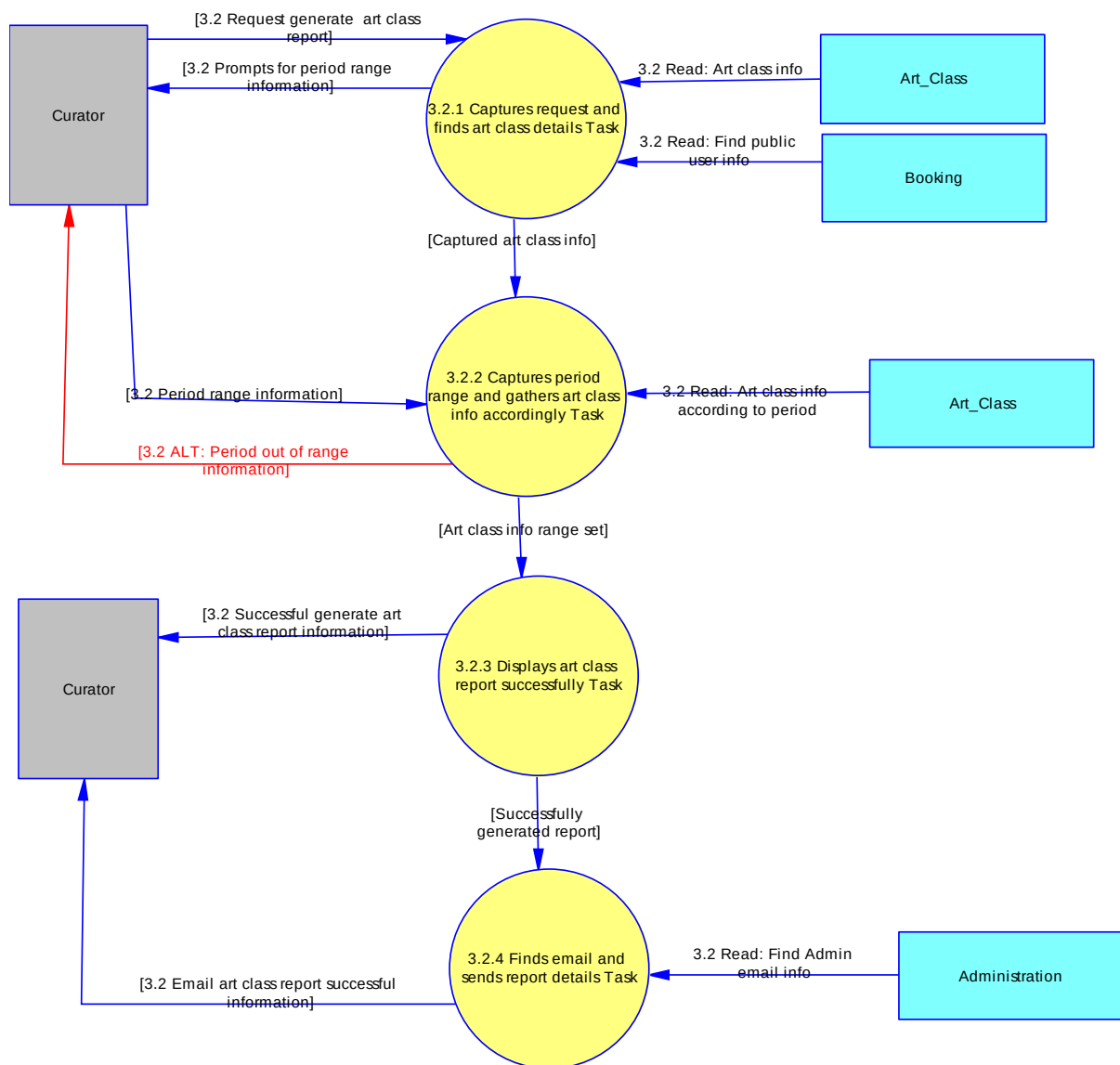


Figure 168: Low Level DFD: 3.2 Generate Art Class Report

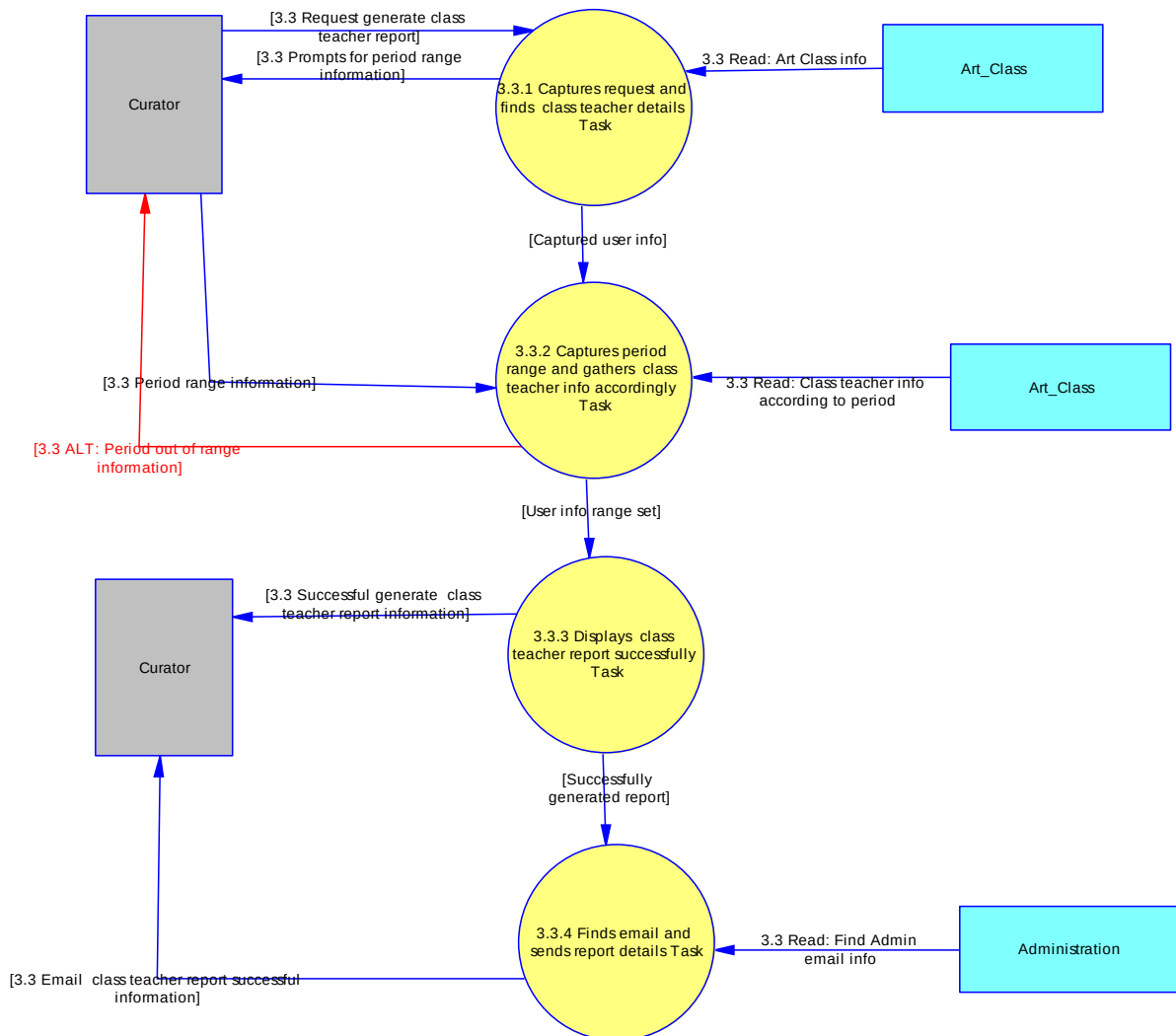


Figure 169: Low Level DFD: 3.3 Generate Class Teacher Report

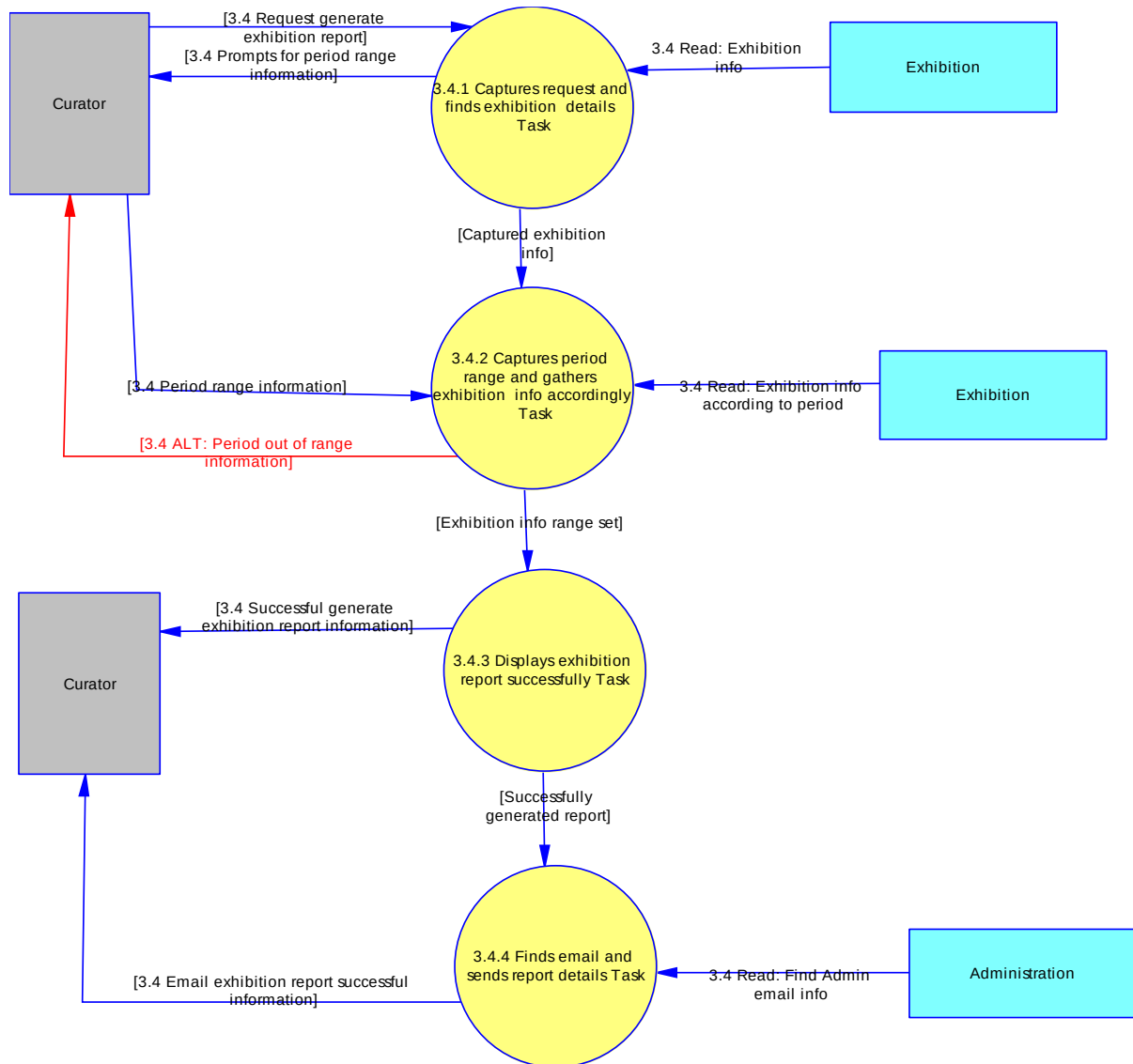


Figure 170: Low Level DFD: 3.4 Generate Exhibition Report

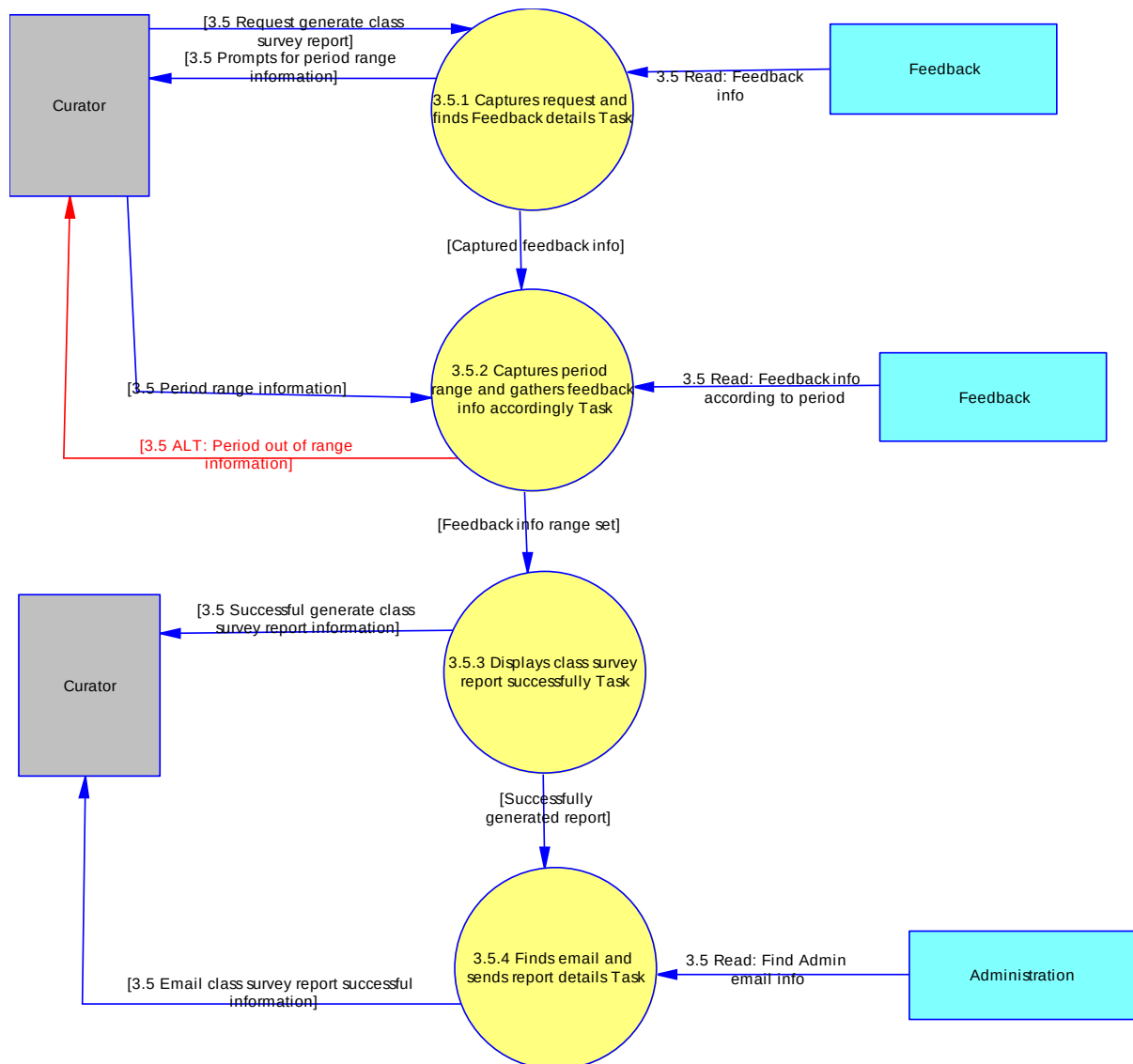


Figure 171: Low Level DFD: 3.5 Generate Class Survey Report

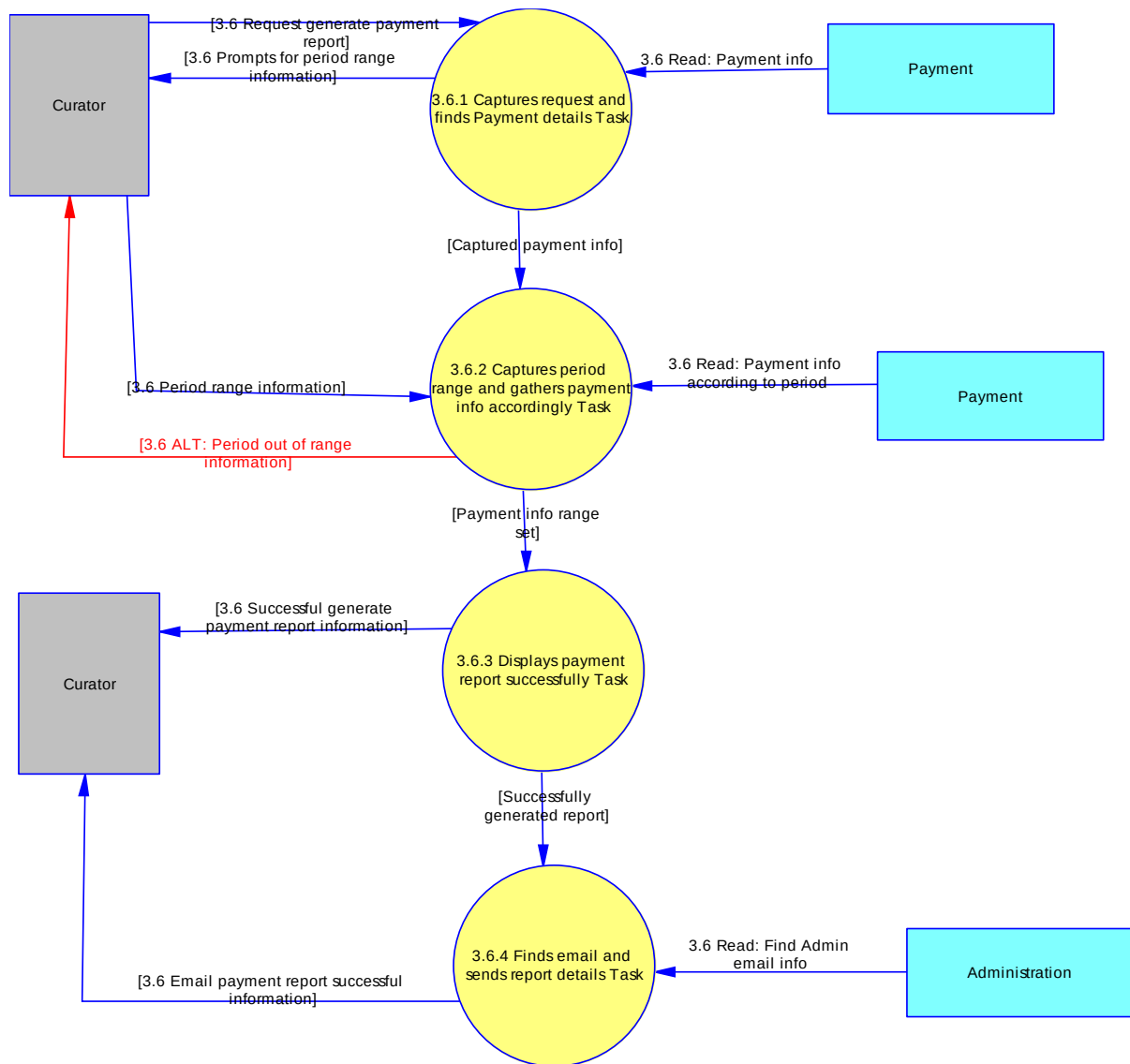


Figure 172: Low Level DFD: 3.6 Generate Payment Report

3.3.3.4 User Subsystem

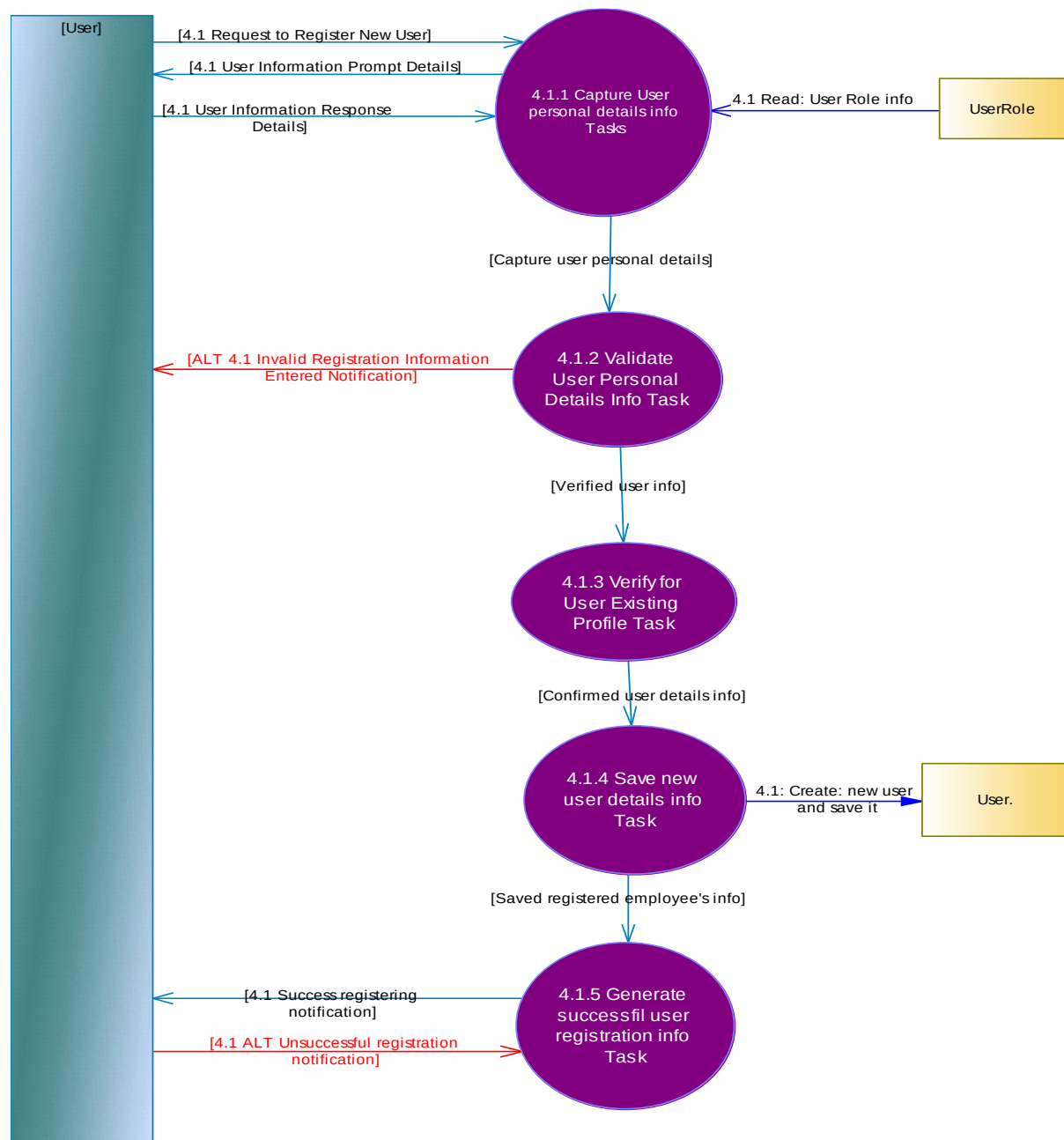


Figure 173: Low Level DFD: 4.1 Register User.

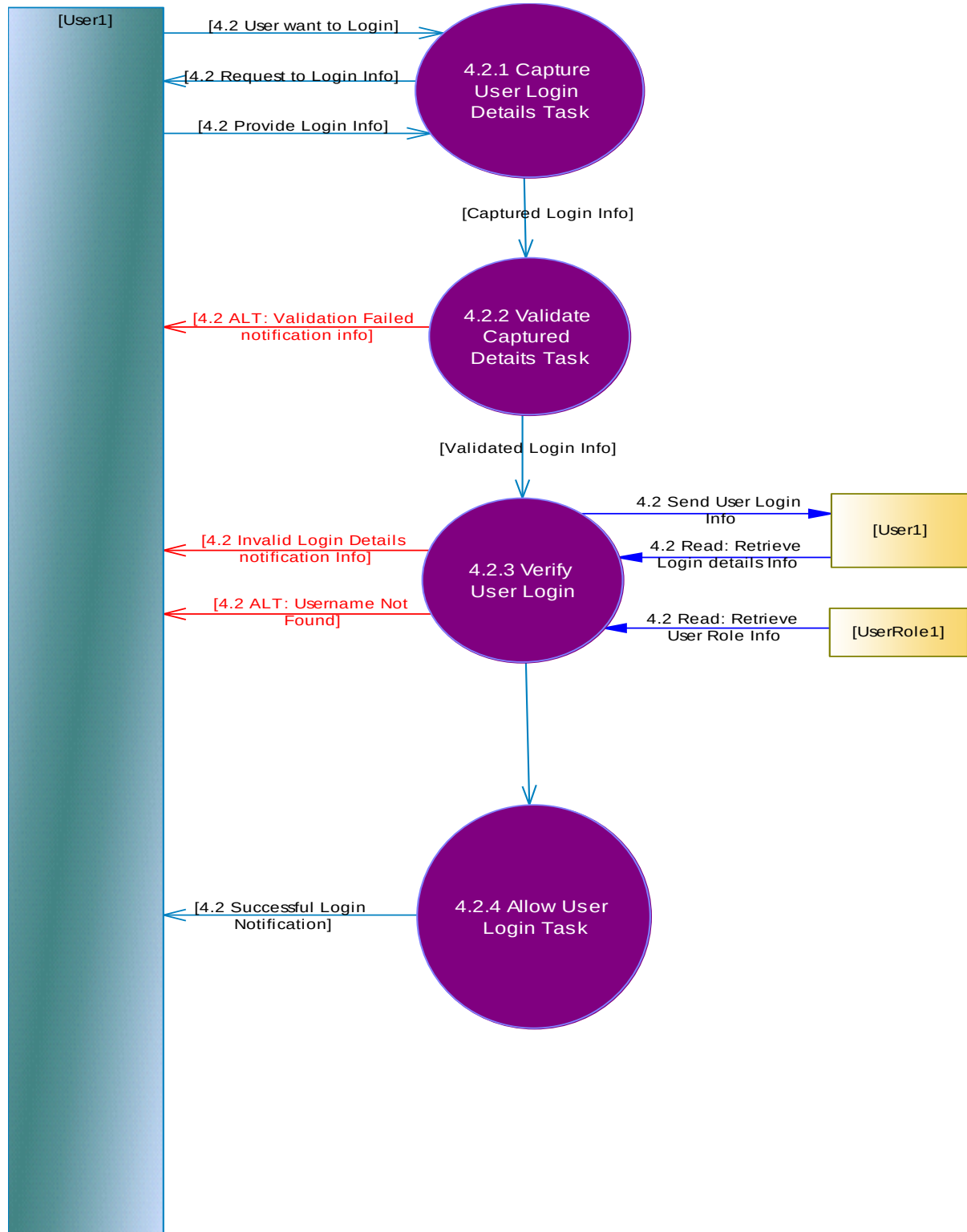


Figure 174: Low Level DFD: 4.2 Log in

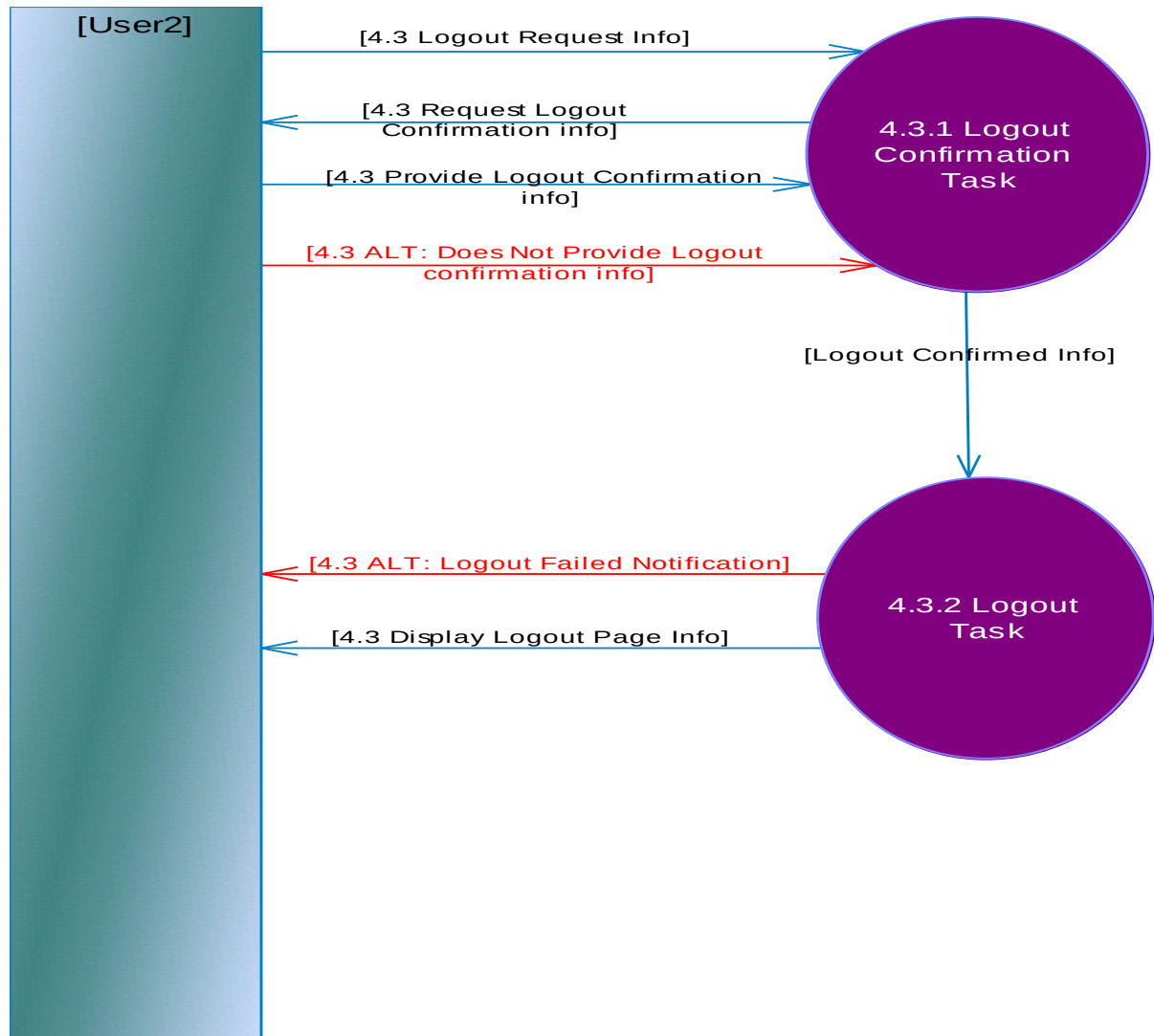


Figure 175: Low Level DFD: 4.3 Log Out

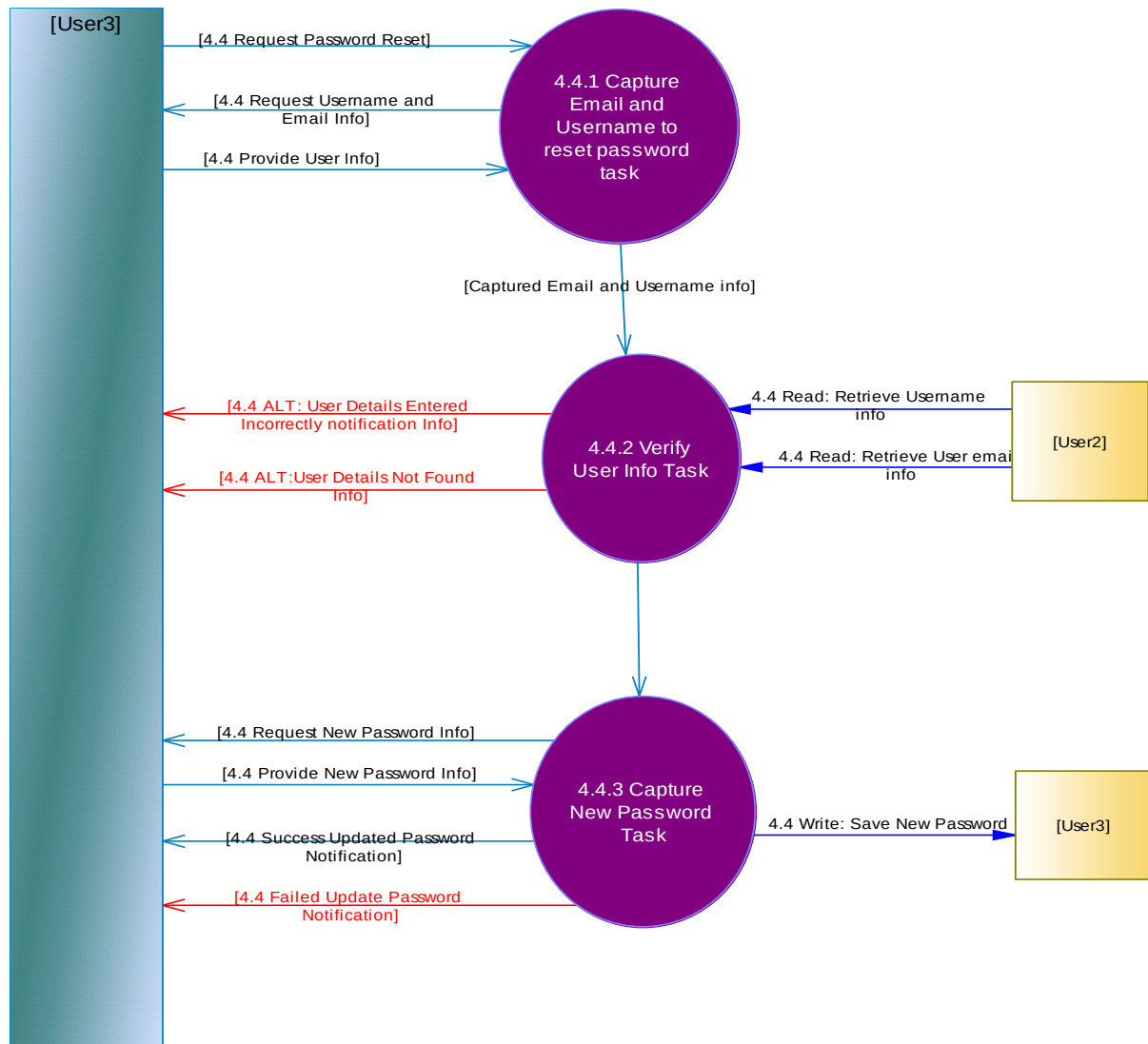


Figure 176: Low Level DFD: 4.4 Reset Password

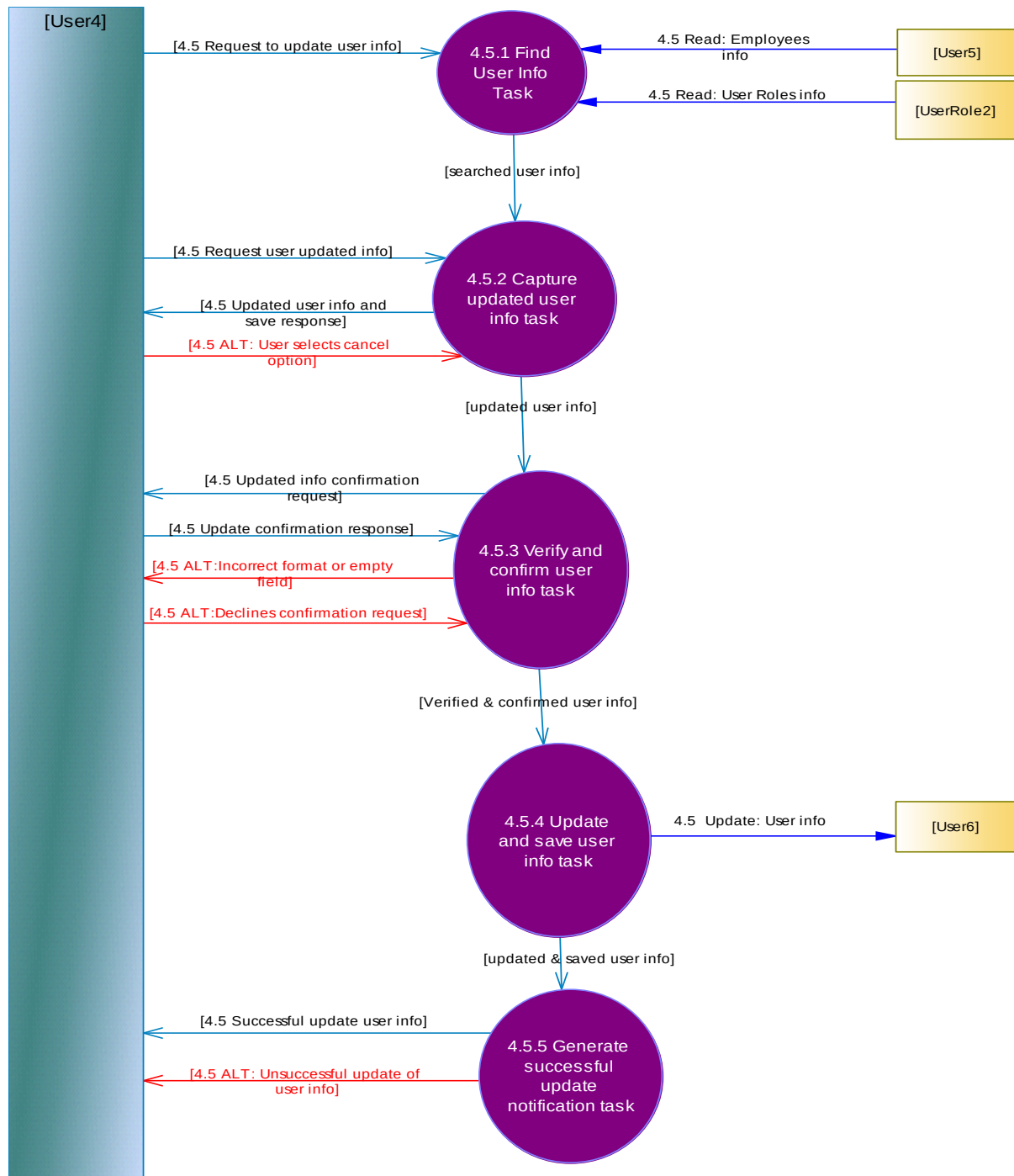


Figure 177: Low Level DFD: 4.5 Update User

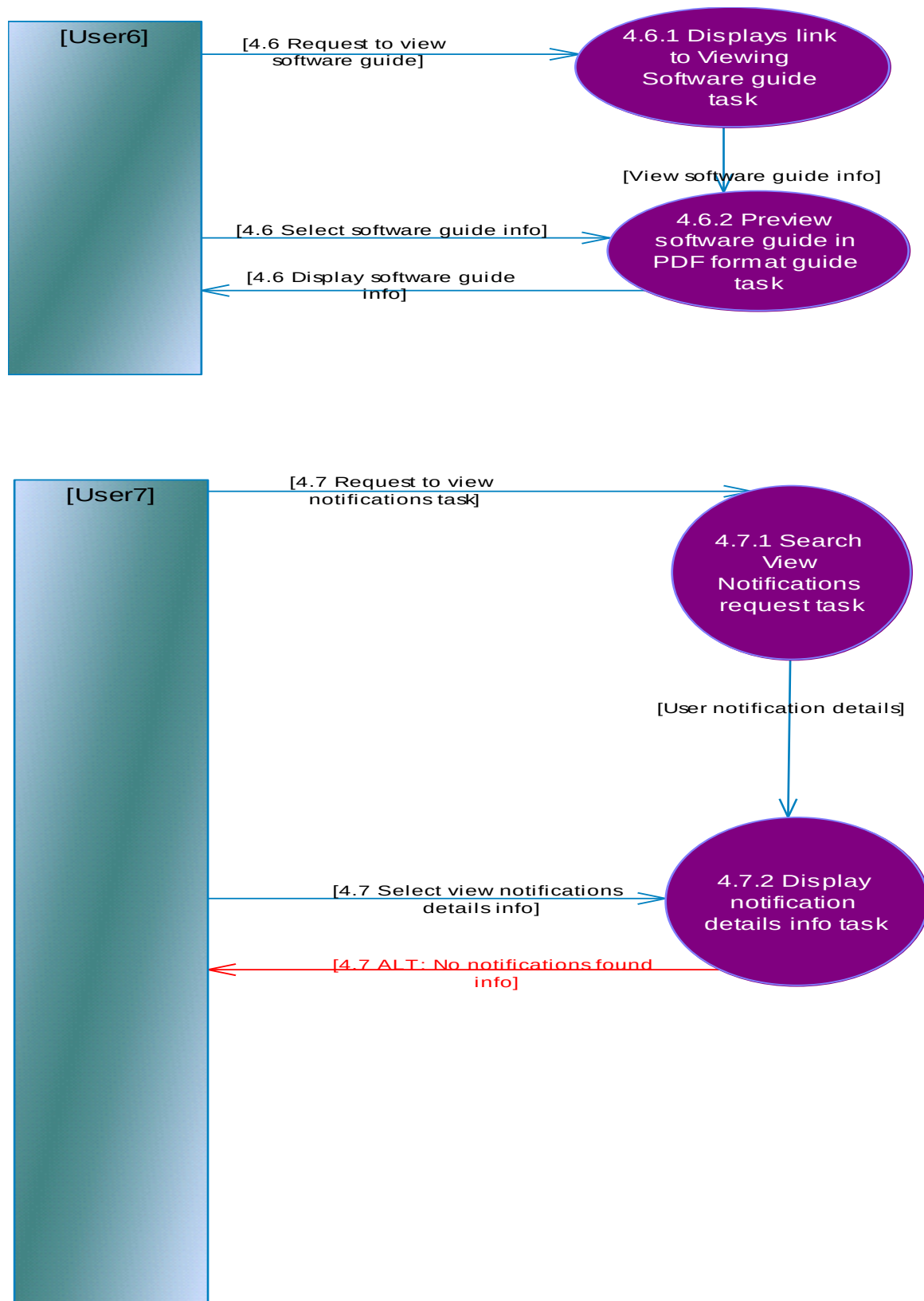


Figure 178: Low Level DFD: 4.7 View Notification

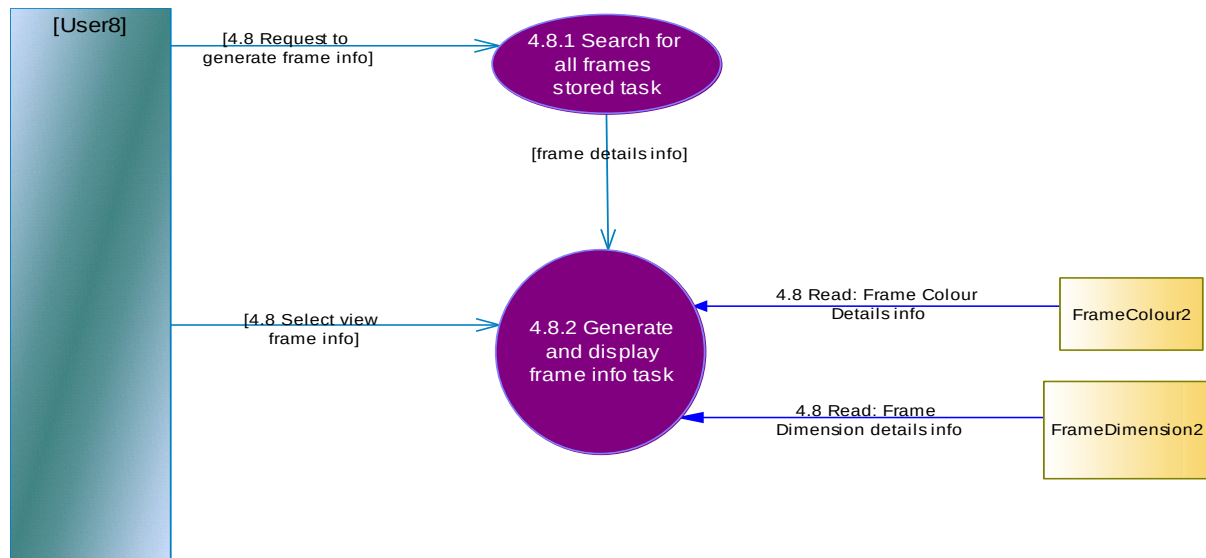


Figure 179: Low Level DFD: 4.8 Generate Frame Info

3.3.3.5 Artist Subsystem

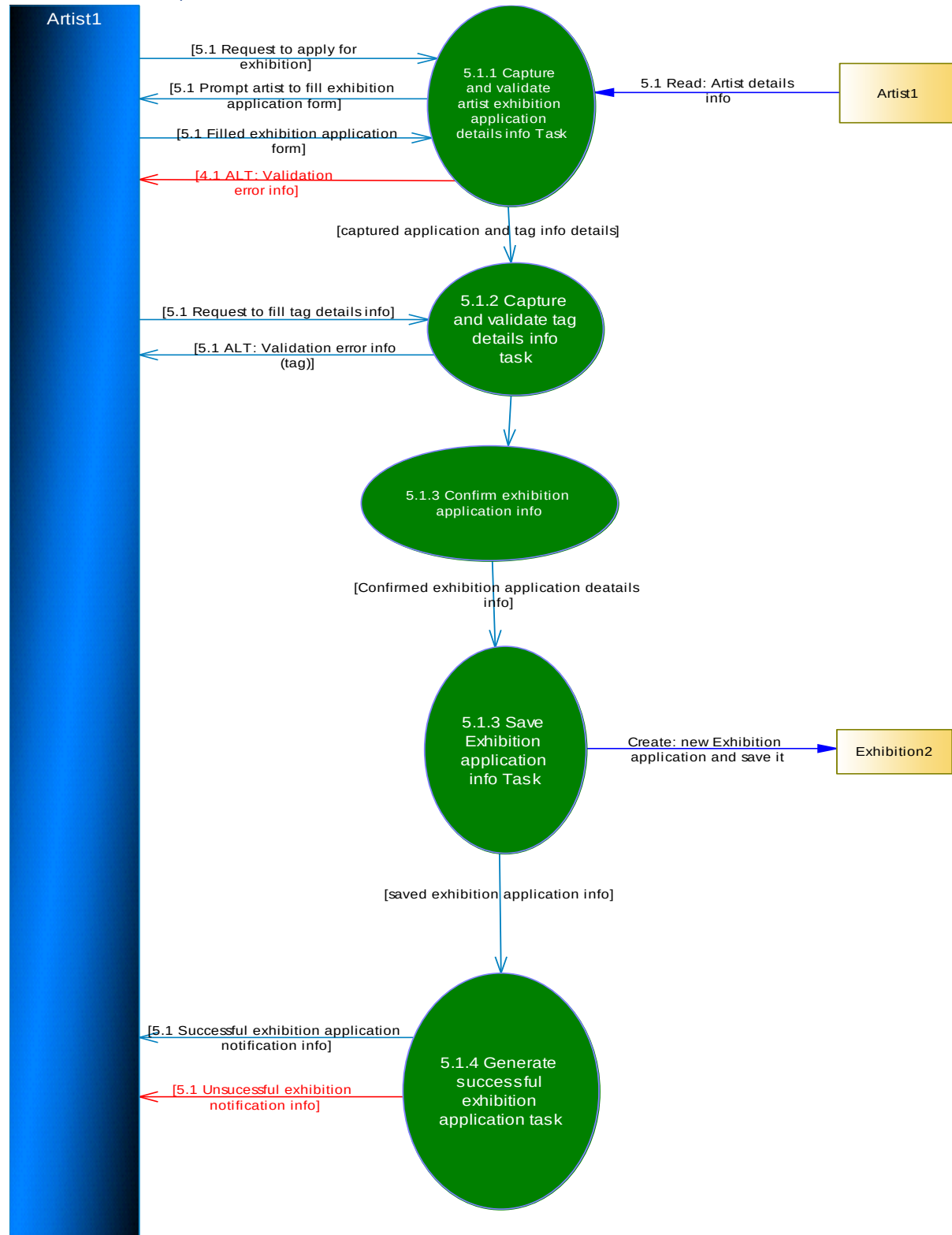


Figure 180: Low Level DFD: 5.1 Apply for Exhibition

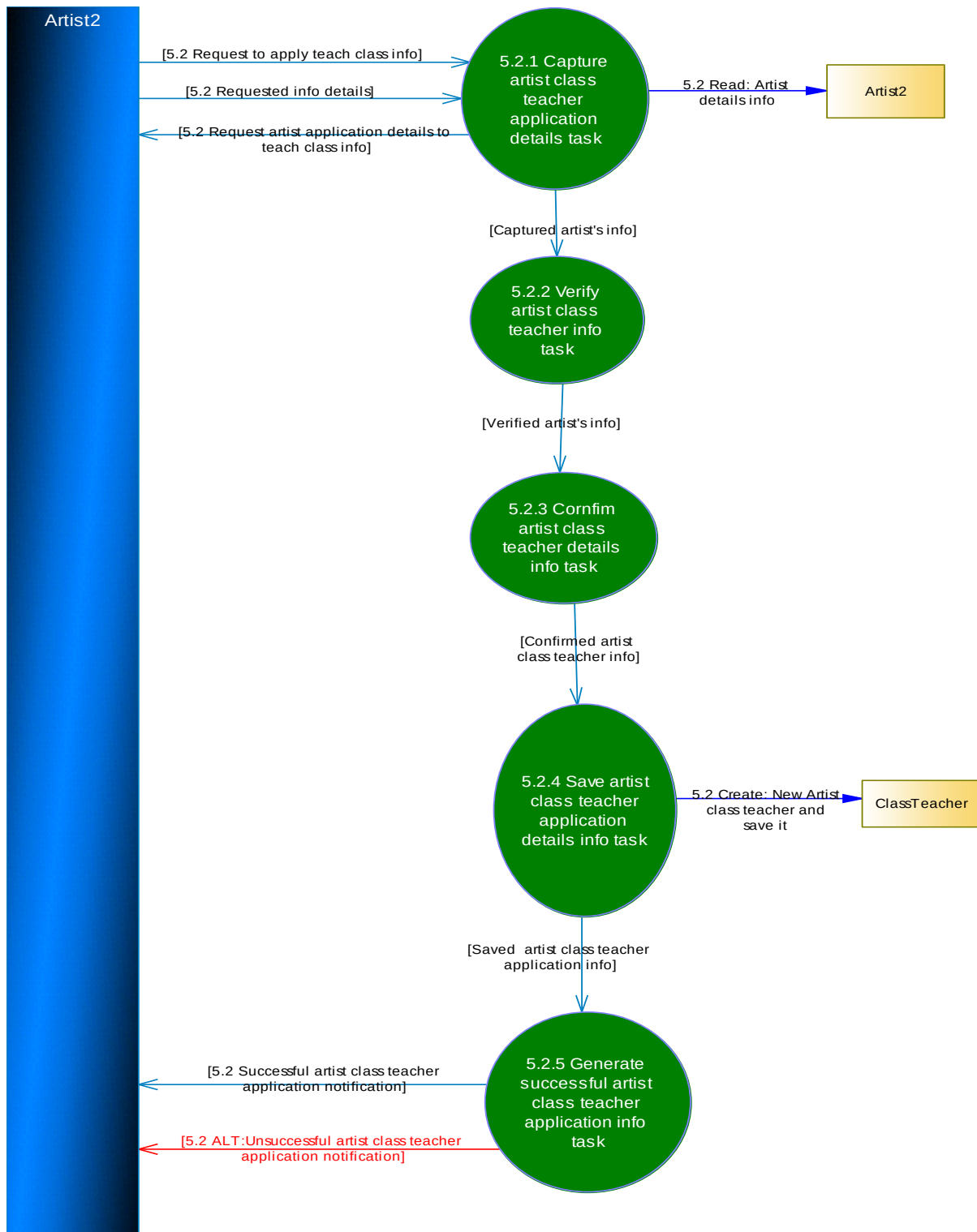


Figure 181: Low Level DFD: 5.2 Apply to Teach

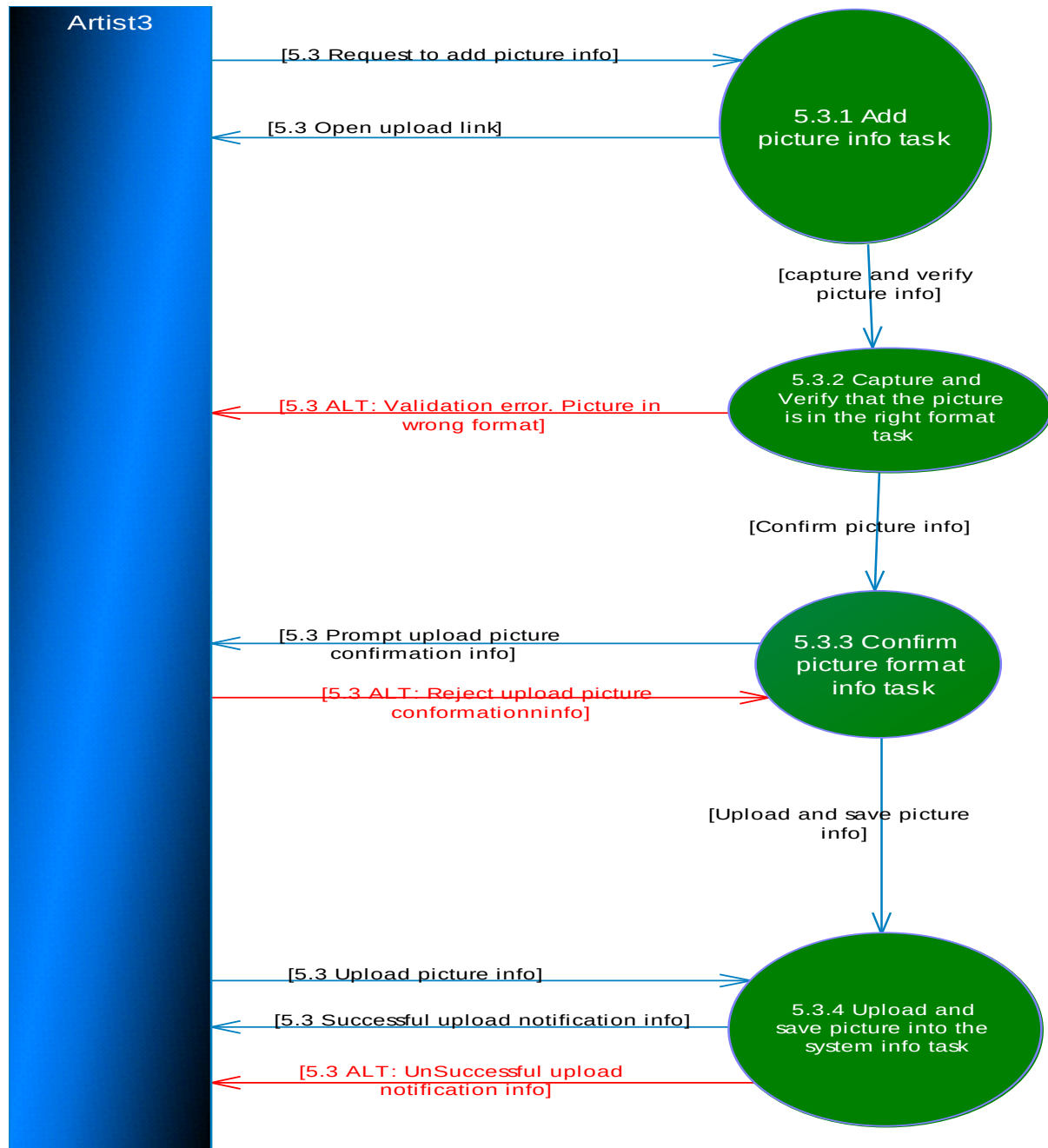


Figure 182: Low Level DFD: 5.3 Add Picture

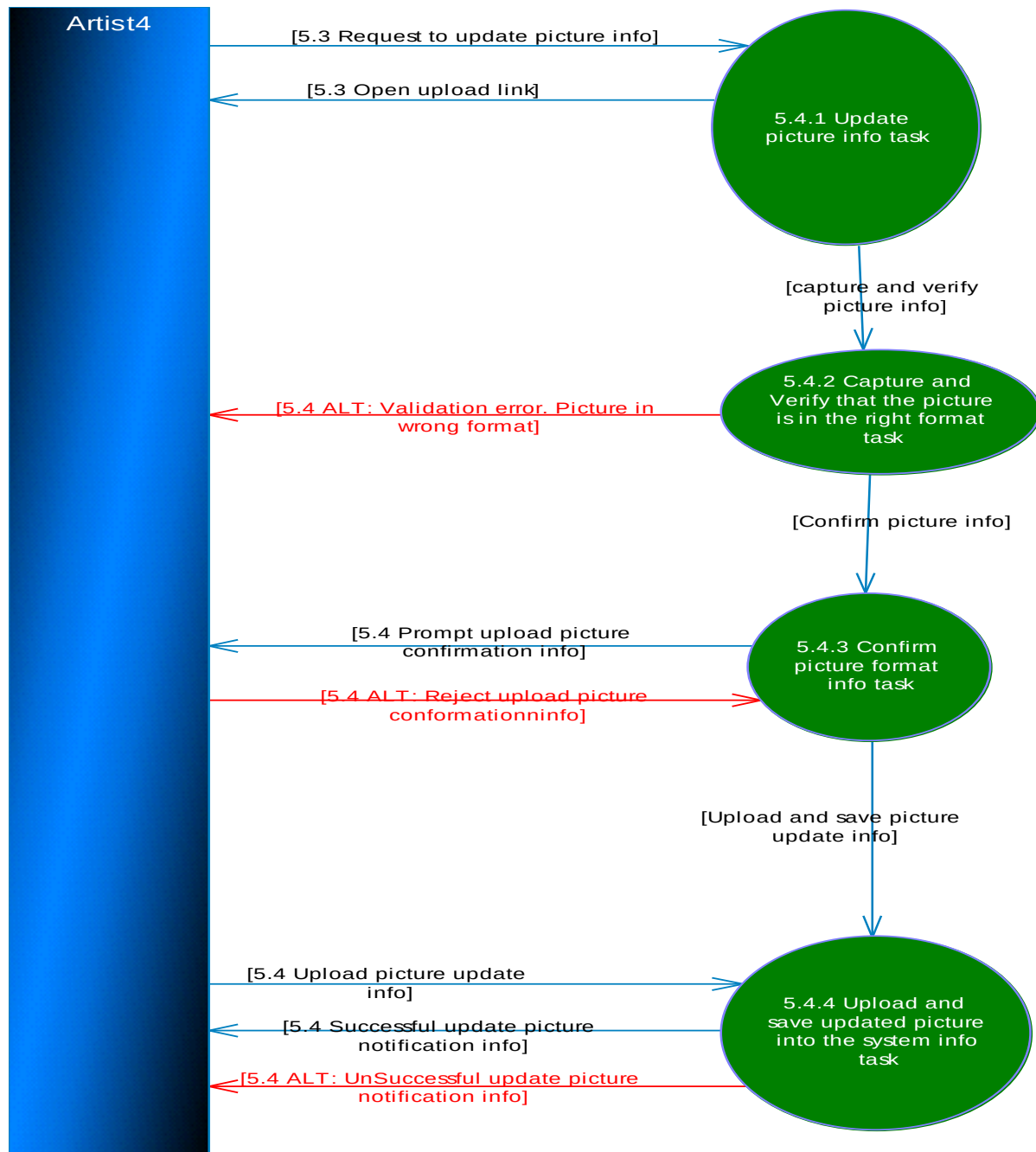


Figure 183: Low Level DFD: 5.4 Update Picture

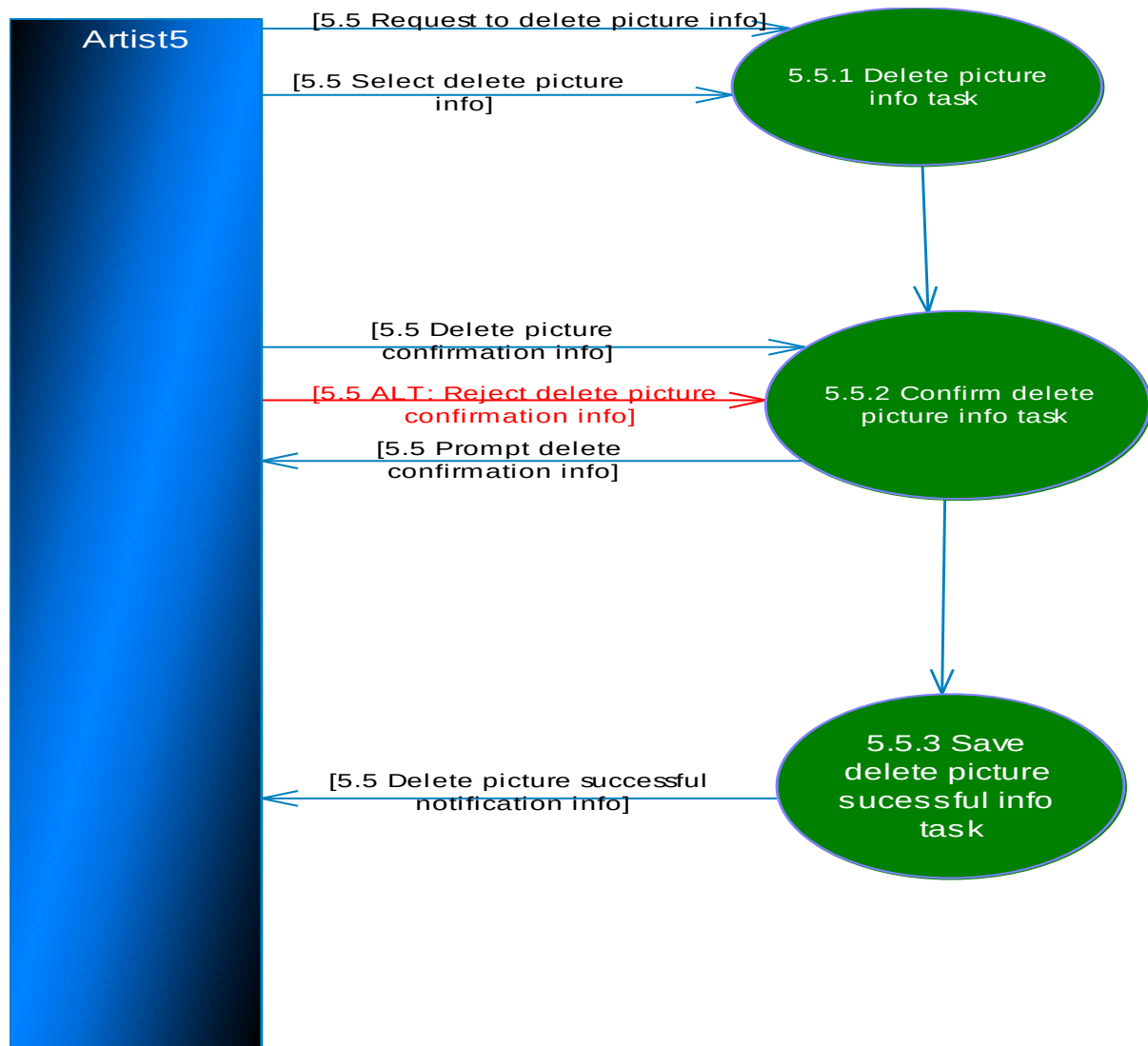


Figure 184: Low Level DFD: 5.5 Delete Picture

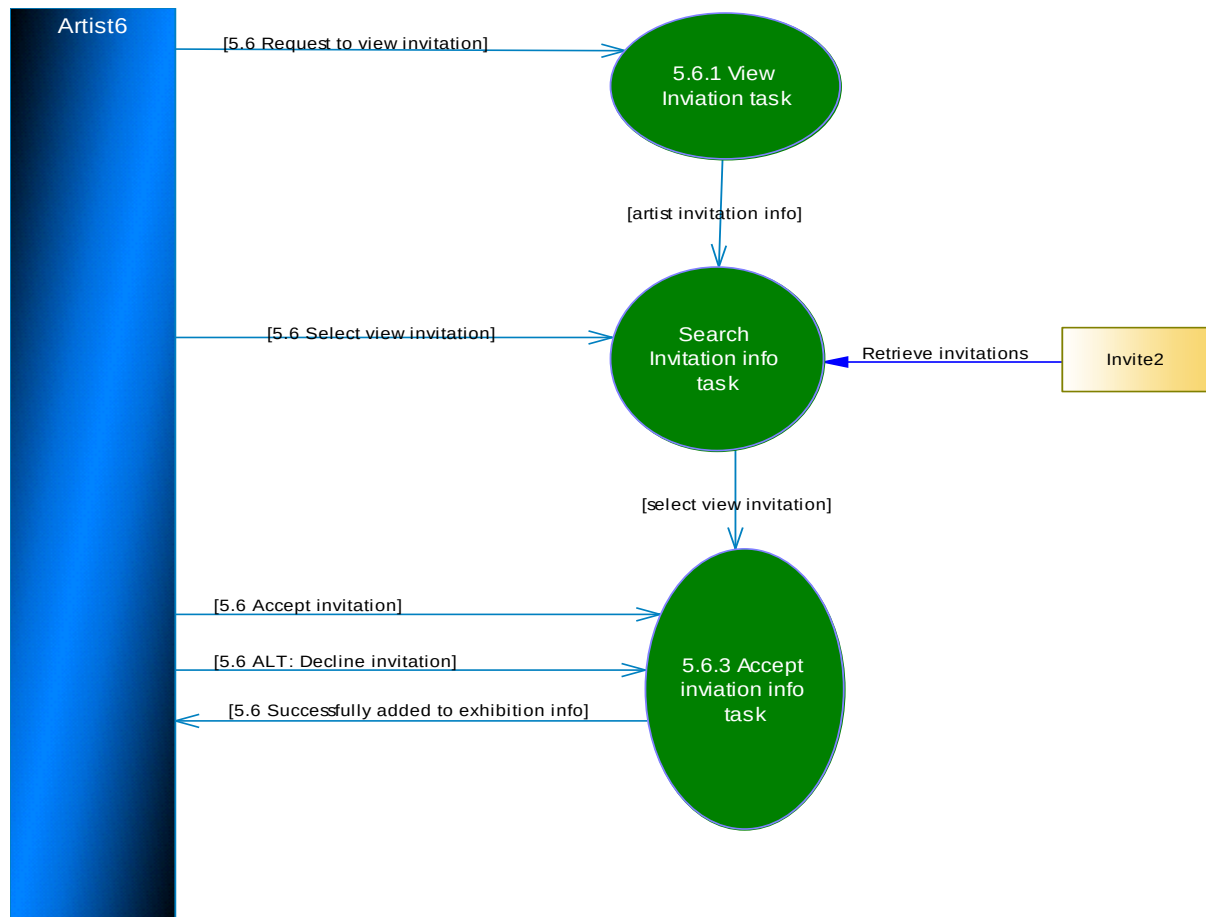


Figure 185: Low Level DFD: 5.6 View Invitation

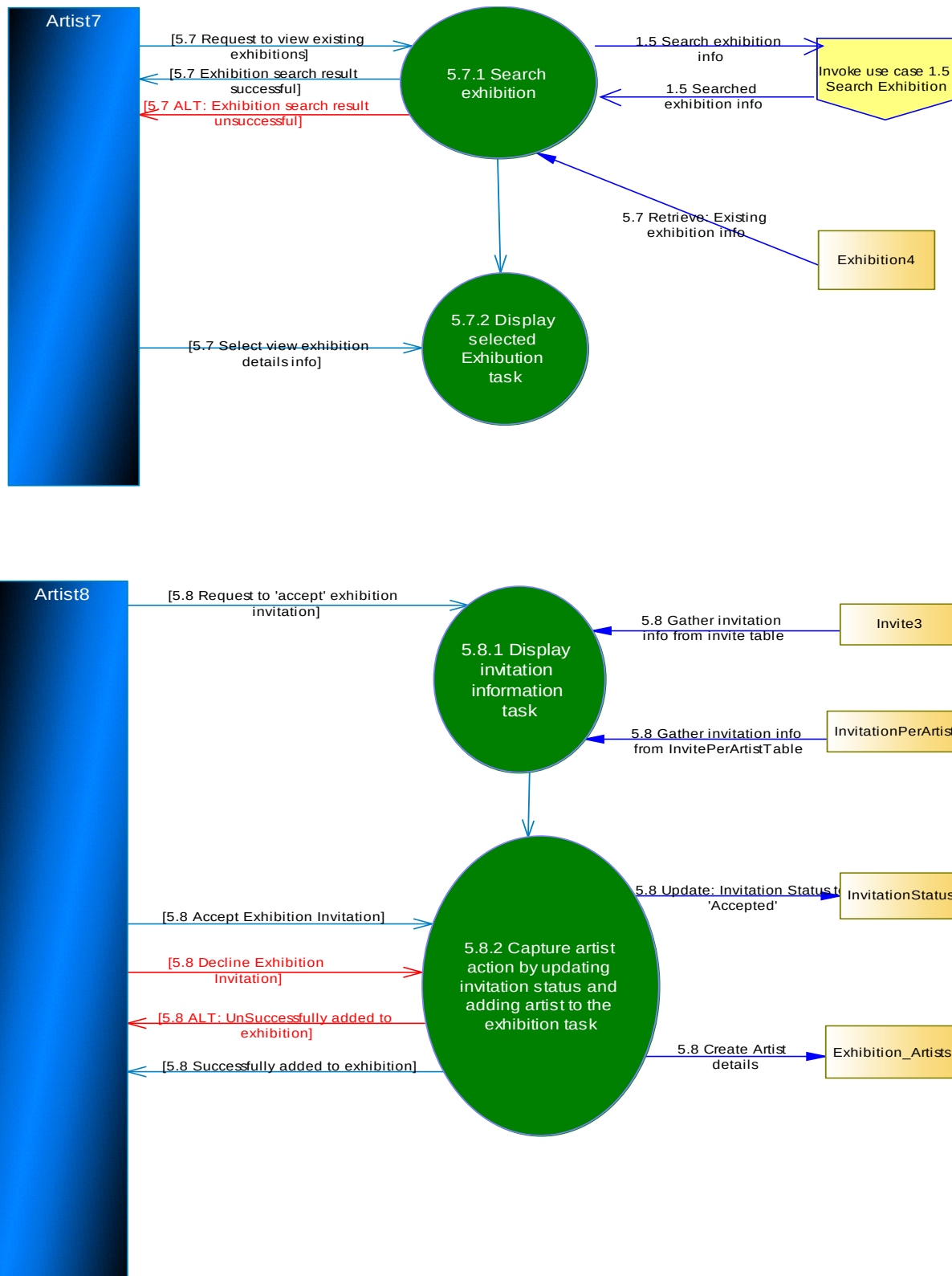


Figure 186: Low Level DFD: 5.8 Accept Invitation Exhibition

3.4 Data Dictionary

Requirement Number:	Attribute	Description	Entities	Example	Action :
1.1	ExhibitionApplicationID	Unique for each application	Exhibition Application	1.	Read
1.1	ArtistID	Unique for each Artist	Exhibition Application	1.	Read
1.1	ExhibitionID	Unique for each Exhibition	Exhibition Application	1.	Read
1.1	ExhibitionPicture1	Picture to apply for Exhibition	Exhibition Application	Image	Read
1.1	ExhibitionPicture2	Picture to apply for Exhibition	Exhibition Application	Image	Read
1.1	ExhibitionPicture2	Picture to apply for Exhibition	Exhibition Application	Image	Read
1.1	ApplicationDescription	Artist Description of application	Exhibition Application	"I would like to exhibit because..."	Read
1.1	ArtistID	Unique for each Artist	Exhibition_Artist	1.	Create
1.1	ExhibitionID	Unique for each Exhibition	Exhibition_Artist	1.	Create

Requirement Number:	Attribute	Description	Entities	Example	Action :
1.2	ArtistID	Unique for each Artist	Exhibition_Artist	1.	Read
1.2	ExhibitionID	Unique for each Exhibition	Exhibition_Artist	1.	Read

Requirement Number:	Attribute	Description	Entities	Example	Action:
1.3	ArtistID	Unique for each Artist	Exhibition_Artist	1.	Delete

1.3	ExhibitionID	Unique for each Exhibition	Exhibition_Artist	1.	Delete

Requirement Number:	Attribute	Description	Entities	Example	Action:
1.4	ExhibitionID	Unique for each Exhibition	Exhibition	1.	Create
1.4	ExhibitionTypeID	Unique for each Exhibition type	Exhibition	1.	Create
1.4	ScheduleID	Unique for each Schedule	Exhibition	1.	Create
1.4	ExhibitionName	Name for an Exhibition	Exhibition	Forester	Create
1.4	ExhibitionDescription	Description for an Exhibition	Exhibition	"Creatures of the forest..."	Create
1.4	ExhibitionDate	Date for an Exhibition	Exhibition	09/08/2021	Create
1.4	OrganisationID	Unique for the Organisation	Exhibition	1.	Create
1.4	ExhibitionTime	Time for an Exhibition	Exhibition	10:00	Create
1.4	ExhibitionAnnouncement_ID	Unique for each Exhibition Announcement	Exhibition	1.	Create
1.4	VenueID	Unique for each Venue	Exhibition	1.	Create

Requirement Number:	Attribute	Description	Entities	Example	Action:
1.5	ExhibitionID	Unique for each Exhibition	Exhibition	1.	Read
1.5	ExhibitionTypeID	Unique for each Exhibition type	Exhibition	1.	Read
1.5	ScheduleID	Unique for each Schedule	Exhibition	1.	Read
1.5	ExhibitionName	Name for an Exhibition	Exhibition	Forester	Read
1.5	ExhibitionDescription	Description for an Exhibition	Exhibition	"Creatures of the forest..."	Read
1.5	ExhibitionDate	Date for an Exhibition	Exhibition	09/08/2021	Read

Requirement Number:	Attribute	Description	Entities	Example	Action:
1.6	ExhibitionTypeID	Unique for each Exhibition type	Exhibition	1.	Update
1.6	ExhibitionName	Name for an Exhibition	Exhibition	Forester	Update
1.6	ExhibitionDescription	Description for an Exhibition	Exhibition	"Creatures of the forest..."	Update
1.6	ExhibitionDate	Date for an Exhibition	Exhibition	09/08/2021	Update
1.6	ExhibitionTime	Time for an Exhibition	Exhibition	10:00	Update
1.6	ExhibitionAnnouncement_ID	Unique for each Exhibition Announcement	Exhibition	1.	Update
1.6	VenueID	Unique for each Venue	Exhibition	1.	Update

Requirement Number:	Attribute	Description	Entities	Example	Action:
1.7	ExhibitionTypeID	Unique for each Exhibition type	Exhibition	1.	Update
1.7	ExhibitionName	Name for an Exhibition	Exhibition	Forester	Update
1.7	ExhibitionDescription	Description for an Exhibition	Exhibition	"Creatures of the forest..."	Update
1.7	ExhibitionDate	Date for an Exhibition	Exhibition	09/08/2021	Update
1.7	ExhibitionTime	Time for an Exhibition	Exhibition	10:00	Update
1.7	ExhibitionAnnouncement_ID	Unique for each Exhibition Announcement	Exhibition	1.	Update
1.7	VenueID	Unique for each Venue	Exhibition	1.	Update

Requirement Number:	Attribute	Description	Entities	Example	Action:
1.8	ArtistID	Unique for each Artist	Artist	1.	Read
1.8	ArtistTypeID	Unique for each Artist type	Artist	1.	Read
1.8	UserRoleID	Unique for each User Role	Artist	1.	Read
1.8	DateofBirth	Date of Birth	Artist	09/08/2000	Read
1.8	ArtistBio	Artist description of talents and education	Artist	"I studied Fine Arts at UP"	Read
1.8	ArtistPicture1	Picture for Artist Profile	Artist	Image	Read
1.8	ArtistPicture2	Picture for Artist Profile	Artist	Image	Read
1.8	ArtistPicture3	Picture for Artist Profile	Artist	Image	Read
1.8	ArtistBankDetails	Artist Banking details	Artist	Account:3840 5555 Standard Bank	Read

Requirement Number:	Attribute	Description	Entities	Example	Action:
1.9	ArtworkID	Unique for each Artwork	Artwork Tag	1.	Create
1.9	ArtworkTitle	Title for an Artwork	Artwork Tag	Ivy the Man	Create
1.9	ArtworkPicture	Image for reference of Artwork	Artwork Tag	Image	Create
1.9	ArtworkPrice	The Price of an Artwork	Artwork Tag	R3000	Create
1.9	ArtworkStatus_ID	Unique for each Artwork Status	Artwork	1.	Create
1.9	SurfaceTypeID	Unique for each Surface Type	Artwork	1.	Create
1.9	MediumTypeID	Unique for each Medium Type	Artwork Tag	1.	Create
1.9	ExhibitionID	Unique for each Exhibition	Artwork	1.	Create
1.9	ArtistID	Unique for each Artist	Artwork	1.	Create
1.9	FrameColourID	Unique for each Frame Color	Artwork	1.	Create

1.9	FrameDimensionID	Unique for each Frame Dimension	Artwork	1.	Create
1.9	TagID	Unique for each Tag	Tag	1.	Create

Requirement Number:	Attribute	Description	Entities	Example	Action:
1.10	ArtworkID	Unique for each Artwork	Artwork	1.	Read
1.10	ArtworkTitle	Title for an Artwork	Artwork	Ivy the Man	Read
1.10	ArtworkPrice	The Price of an Artwork	Artwork	R3000	Read
1.10	ArtworkStatus_ID	Unique for each Artwork Status	Artwork	1.	Read
1.10	SurfaceTypeID	Unique for each Surface Type	Artwork	1.	Read
1.10	MediumTypeID	Unique for each Medium Type	Artwork	1.	Read
1.10	ExhibitionID	Unique for each Exhibition	Artwork	1.	Read
1.10	ArtistID	Unique for each Artist	Artwork	1.	Read

Requirement Number:	Attribute	Description	Entities	Example	Action:
1.11	ArtworkID	Unique for each Artwork	Artwork Tag	1.	Delete
1.11	ArtworkTitle	Title for an Artwork	Artwork Tag	Ivy the Man	Delete
1.11	ArtworkPicture	Image for reference of Artwork	Artwork Tag	Image	Delete
1.11	ArtworkPrice	The Price of an Artwork	Artwork Tag	R3000	Delete
1.11	ArtworkStatus_ID	Unique for each Artwork Status	Artwork	1.	Delete
1.11	SurfaceTypeID	Unique for each Surface Type	Artwork	1.	Delete
1.11	MediumTypeID	Unique for each Medium Type	Artwork Tag	1.	Delete
1.11	ExhibitionID	Unique for each Exhibition	Artwork	1.	Delete
1.11	ArtistID	Unique for each Artist	Artwork	1.	Delete
1.11	FrameColourID	Unique for each Frame Color	Artwork	1.	Delete

1.11	FrameDimensionID	Unique for each Frame Dimension	Artwork	1.	Delete
1.11	TagID	Unique for each Tag	Tag	1.	Delete

Requirement Number:	Attribute	Description	Entities	Example	Action:
1.12	ArtworkID	Unique for each Artwork	Artwork Tag	1.	Update
1.12	ArtworkTitle	Title for an Artwork	Artwork Tag	Ivy the Man	Update
1.12	ArtworkPicture	Image for reference of Artwork	Artwork Tag	Image	Update
1.12	ArtworkPrice	The Price of an Artwork	Artwork Tag	R3000	Update
1.12	ArtworkStatus_ID	Unique for each Artwork Status	Artwork	1.	Update
1.12	SurfaceTypeID	Unique for each Surface Type	Artwork	1.	Update
1.12	MediumTypeID	Unique for each Medium Type	Artwork Tag	1.	Update
1.12	ExhibitionID	Unique for each Exhibition	Artwork	1.	Update
1.12	ArtistID	Unique for each Artist	Artwork	1.	Update
1.12	FrameColourID	Unique for each Frame Color	Artwork	1.	Update
1.12	FrameDimensionID	Unique for each Frame Dimension	Artwork	1.	Update

Requirement Number:	Attribute	Description	Entities	Example	Action:
1.13	InvitationID	Unique for each Invitation	Invite	1.	Create
1.13	InvitationDetails	Details to what invited and what to showcase	Invite	"Invite to showcase at the Forester"	Create
1.13	InvitationDate	Date of Invitation	Invite	09/08/2021	Create
1.13	ArtistID	Unique for each Artist	Invite	1.	Create
1.13	ExhibitionID	Unique for each Exhibition	Invite	1.	Create
1.13	InvitationStatusID	Unique for each InvitationStatus	Invite	1.	Create

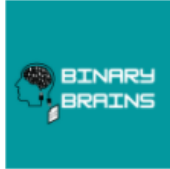
Requirement Number:	Attribute	Description	Entities	Example	Action:
1.15	InvitationID	Unique for each Invitation	Invite	1.	Create
1.15	InvitationDetails	Details to what invited and what to showcase	Invite	"Invite to showcase at the Forester"	Create
1.15	InvitationDate	Date of Invitation	Invite	09/08/2021	Create
1.15	ArtistID	Unique for each Artist	Invite	1.	Create
1.15	ExhibitionID	Unique for each Exhibition	Invite	1.	Create
1.15	InvitationStatusID	Unique for each InvitationStatus	Invite	1.	Create

Requirement Number:	Attribute	Description	Entities	Example	Action:
1.17	ArtClassName	The name of a class.	Art_Class	Sketching	Create
1.17	ArtClassStartDate	The date of when the art class begins.	Art_Class	09/08/2021	Create
1.17	Art Class Description	Describes the art class taking place.	Art_Class	"This art class will be about sketching"	Create
1.17	ArtClassEndDate	The date of when the art class ends.	Art_Class	09/08/2021	Create
1.17	Class Price	The price of the class	Art_Class	R600	Create

1.17	Art Class Type	The type of art class	Art_Class	Sketching	Create
1.17	Art Class Teacher	The teacher of the art class	Art_Class	Ms Jane van Heerden	Create
1.17	ArtClassAnnouncement_ID	Unique ID for every Annoucement	Art_Class	1.	Create
1.17	ClassTeacherID	Unique ID for every Teacher	Art_Class	1.	Create
1.17	ClassLimit	The limit of people in a class	Art_Class	8	Create
1.17	VenueID	The venue of the art class	Art_Class	Dawn Room 3	Create

Conclusion

In conclusion the above section helps outline which processes go where to in order to build the proposed system.



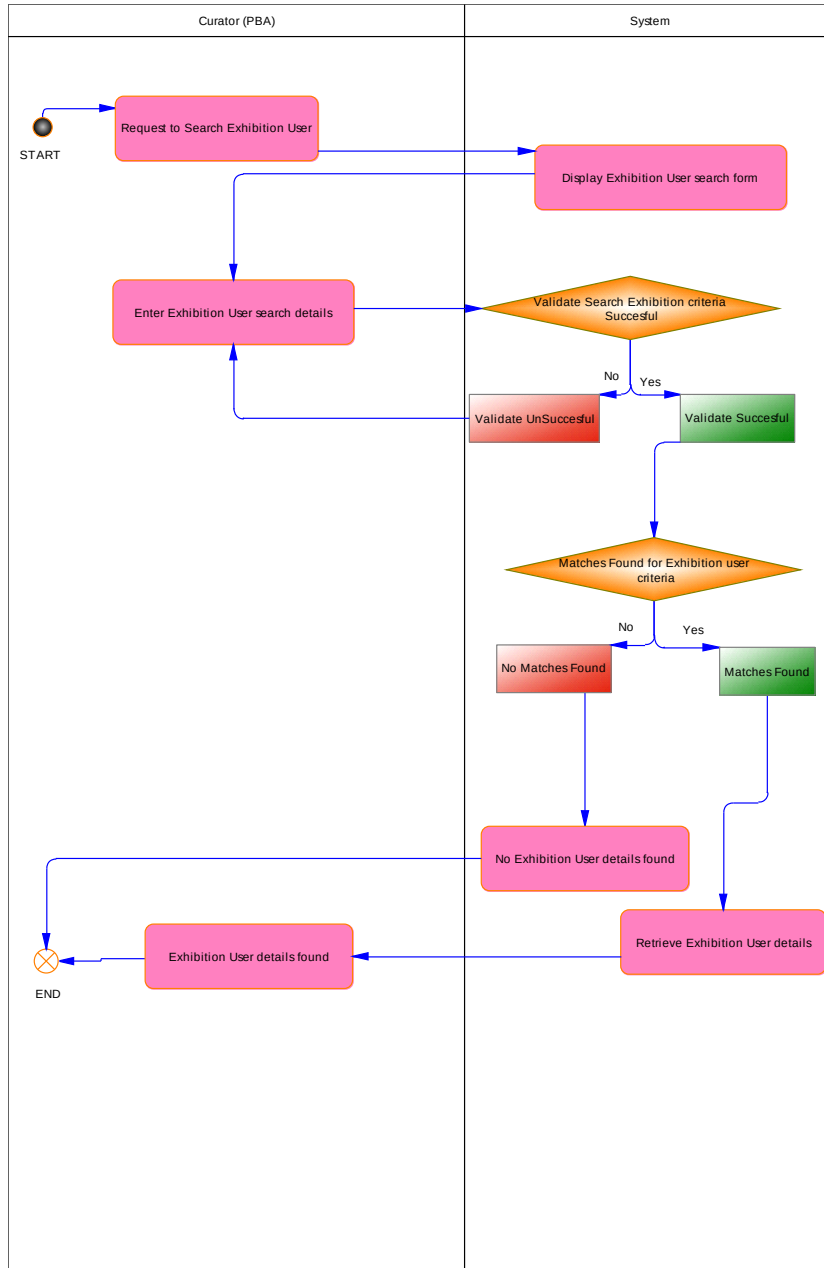
4. UML Modeling

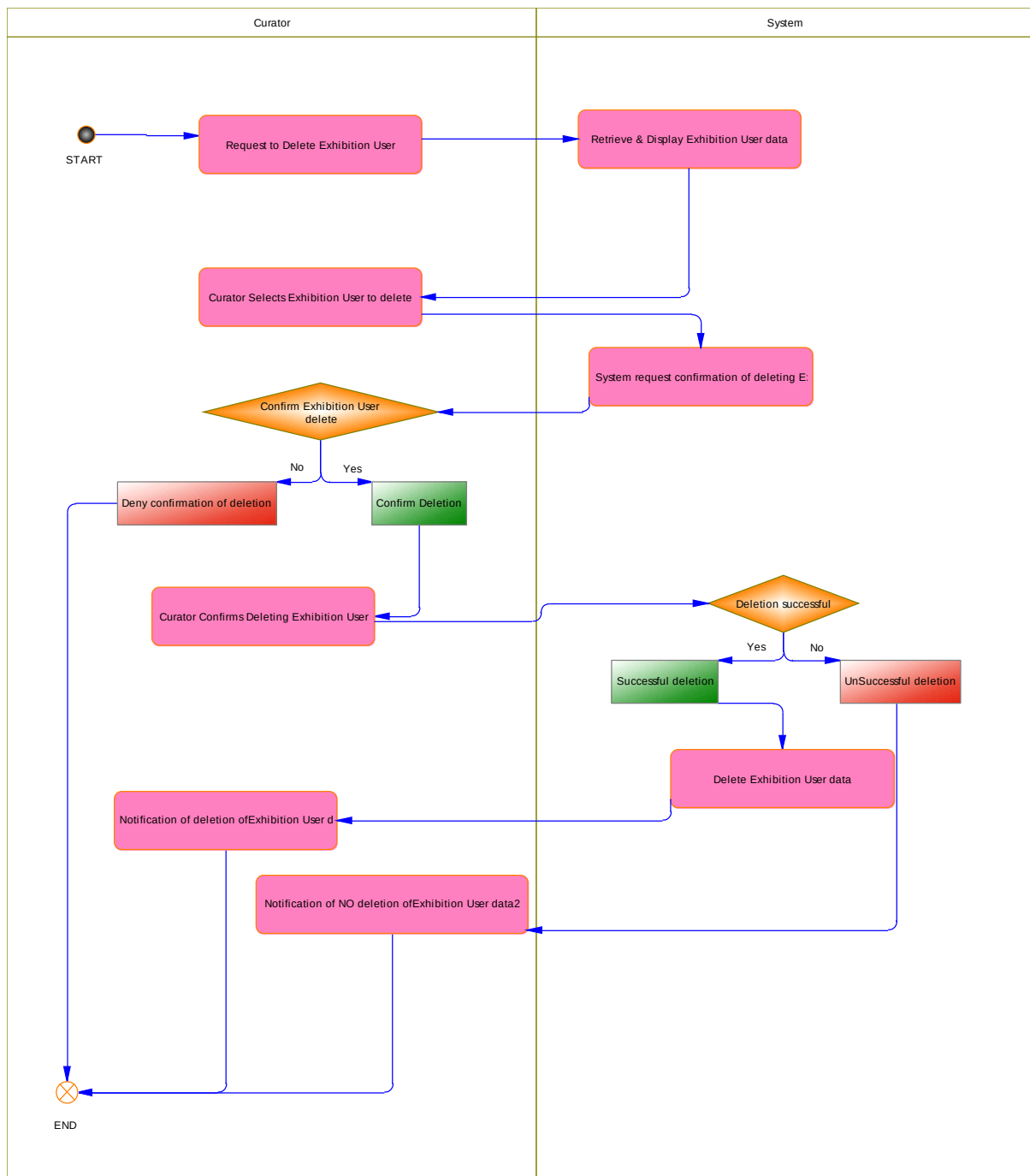
Introduction

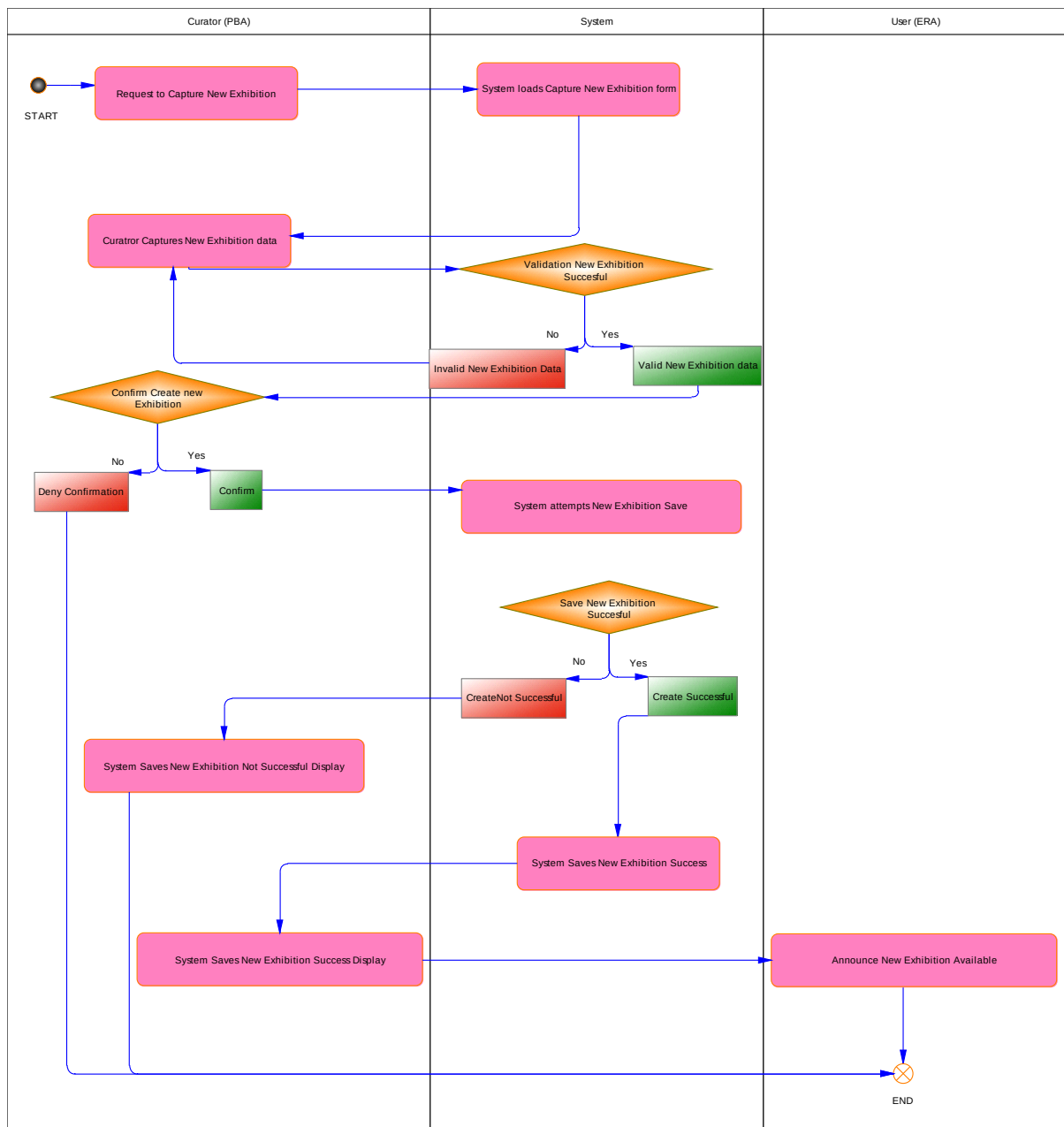
This section contains the set of activity diagrams for each requirement and use case in the proposed system. UML Modelling is developed to help system and software developers for specifying, visualizing, constructing, and documenting the artifacts of software systems. Activity diagram is basically a flowchart to represent the flow from one *activity* to another *activity*. The *activity* can be described as an operation of the proposed system.

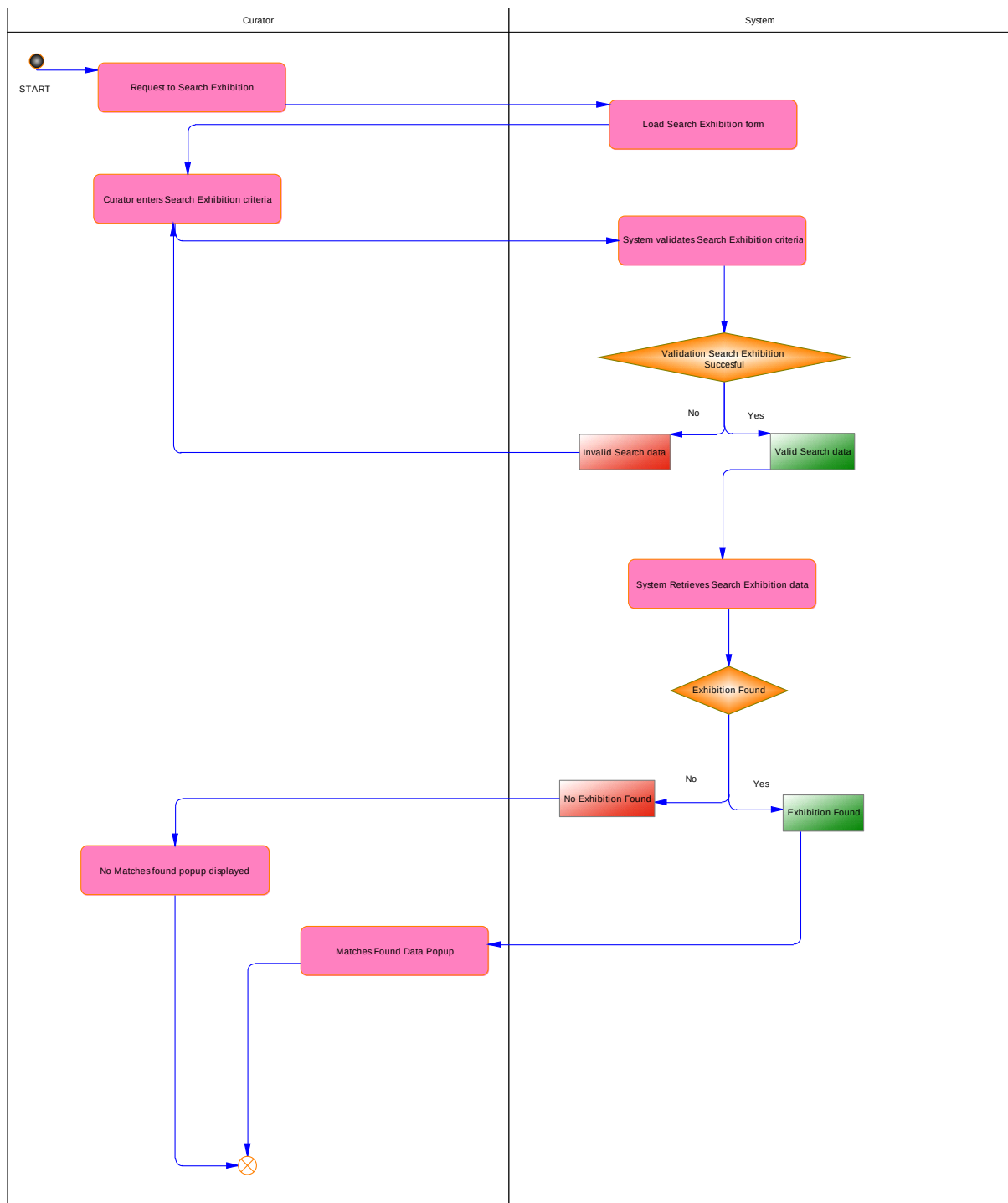
4.1 Activity Diagrams

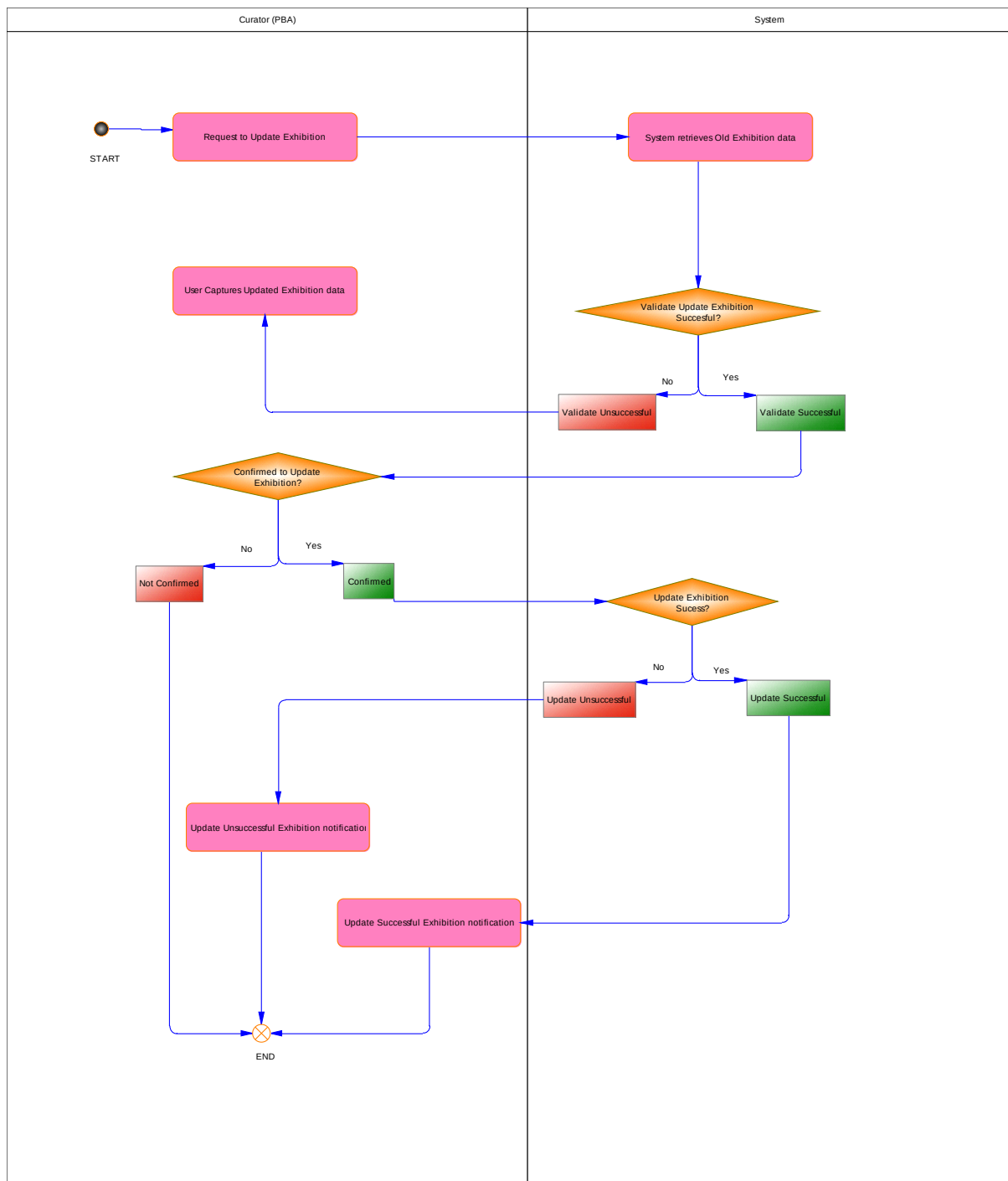
4.1.1 Administration Subsystem

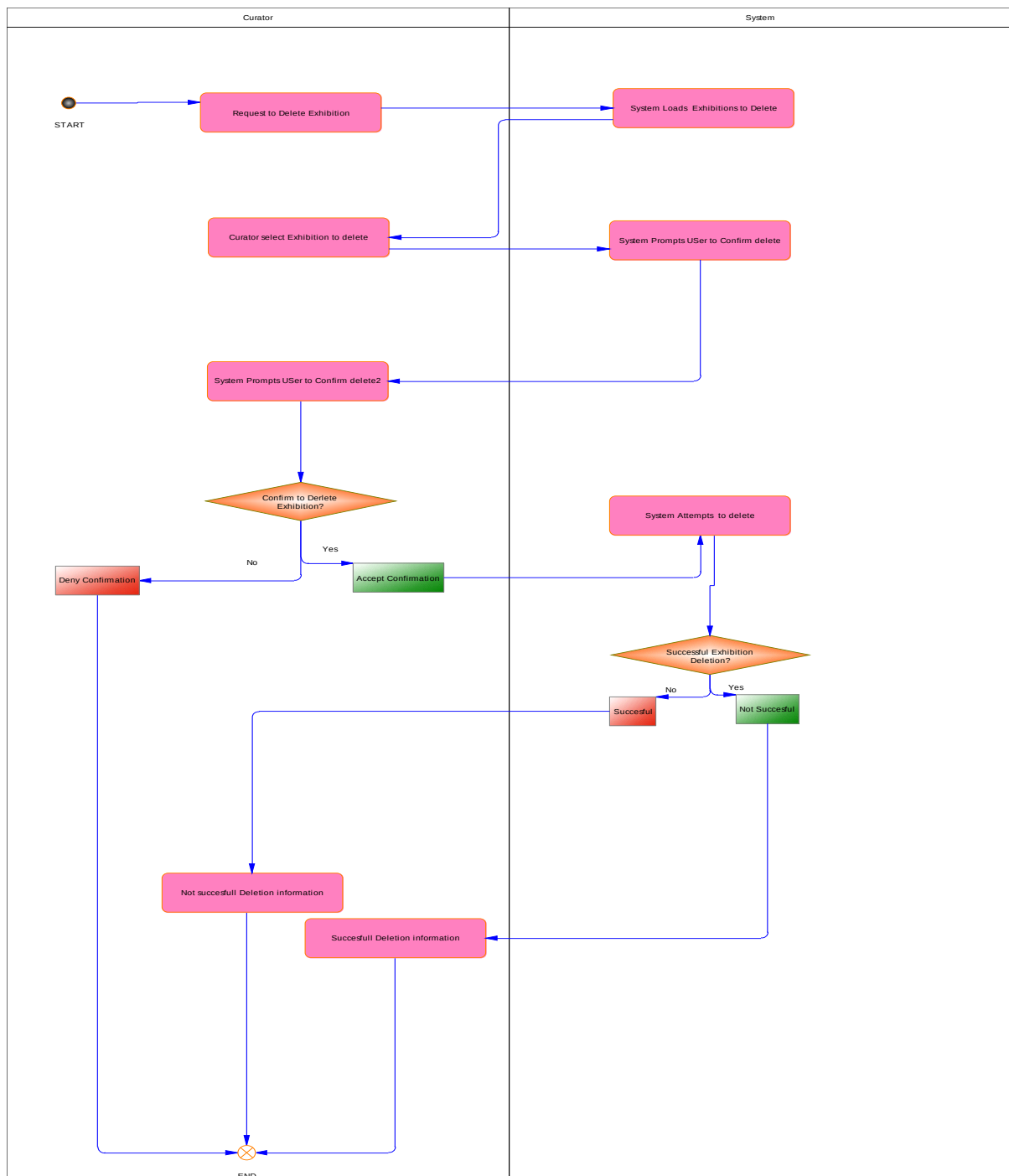


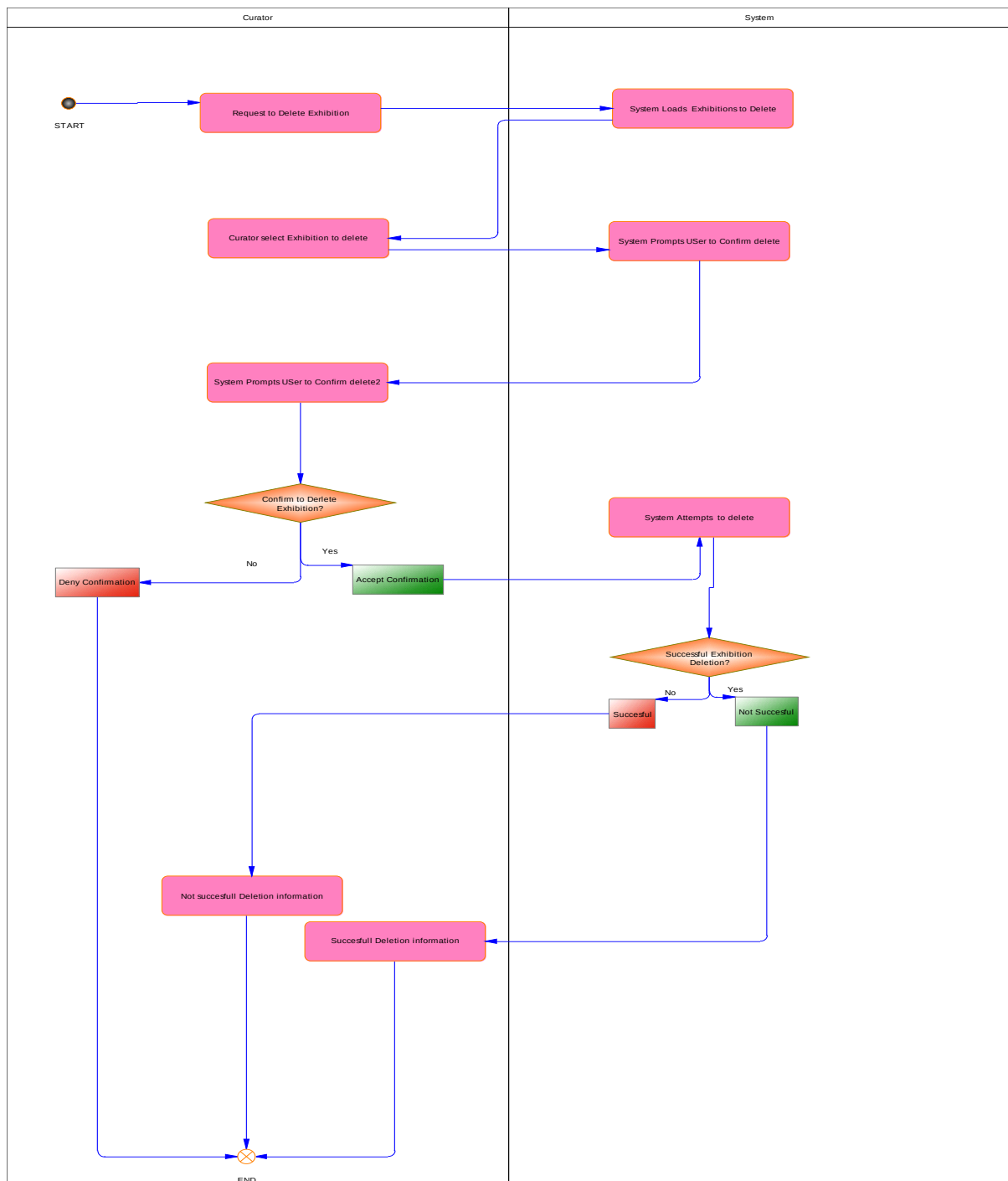


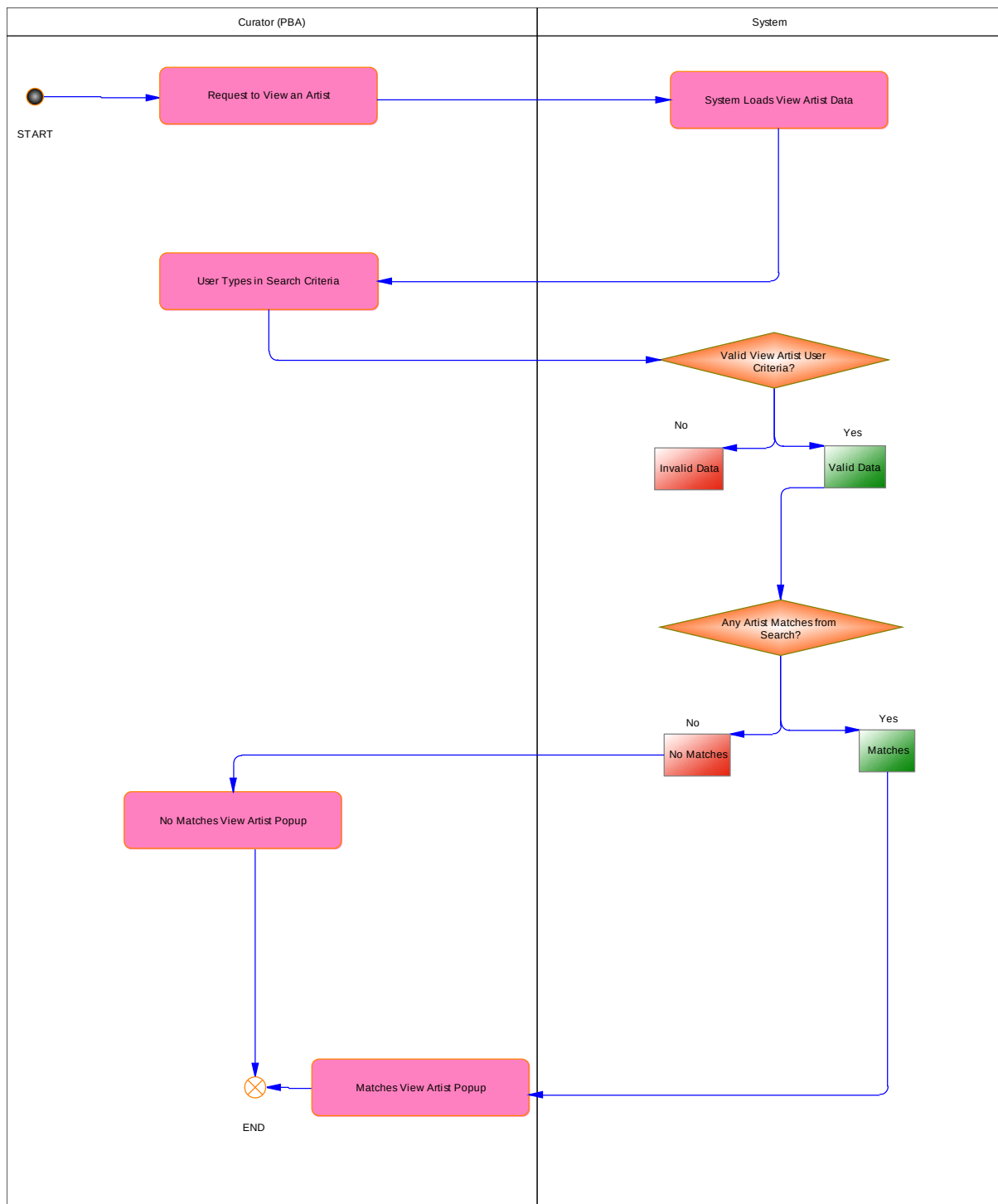


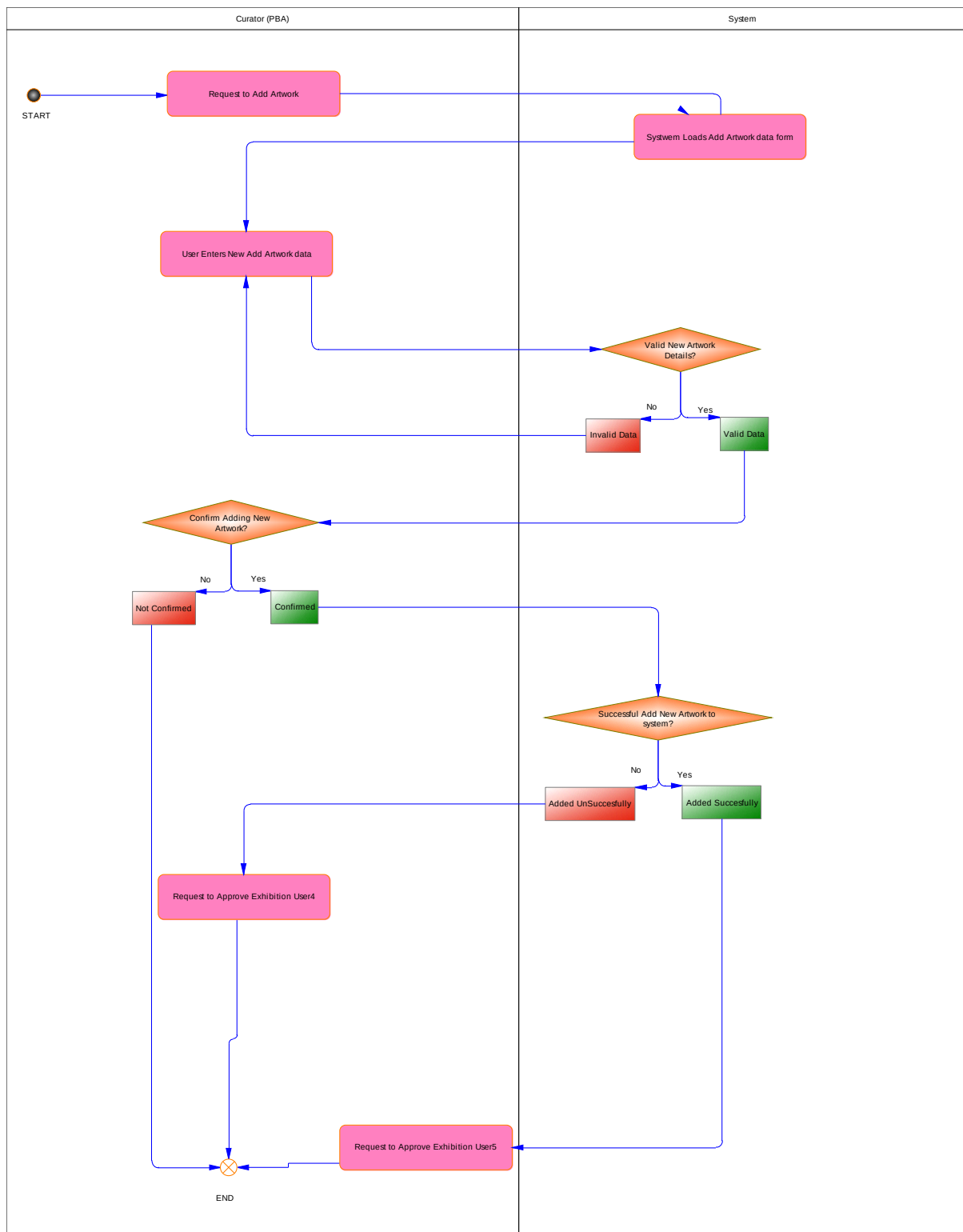


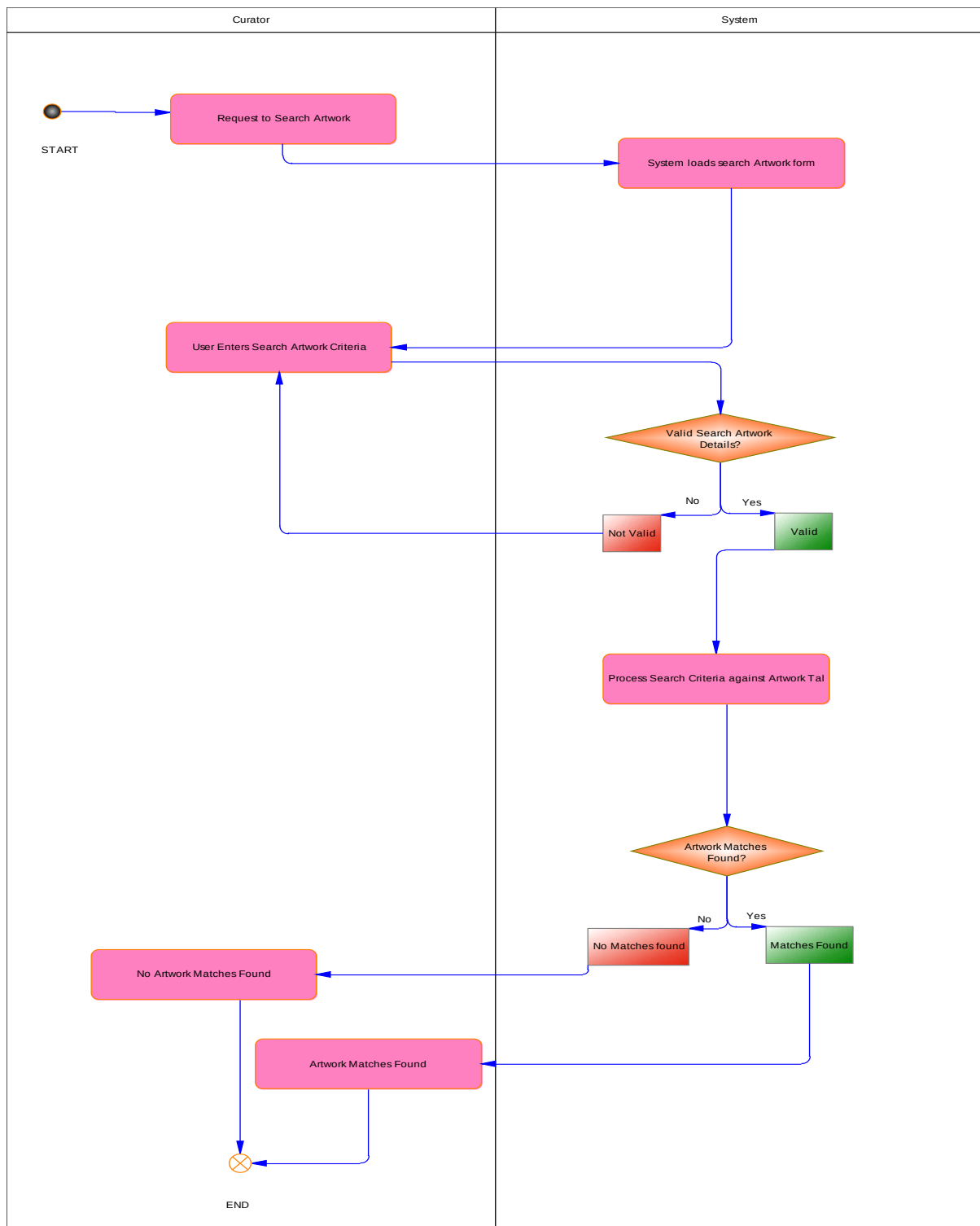


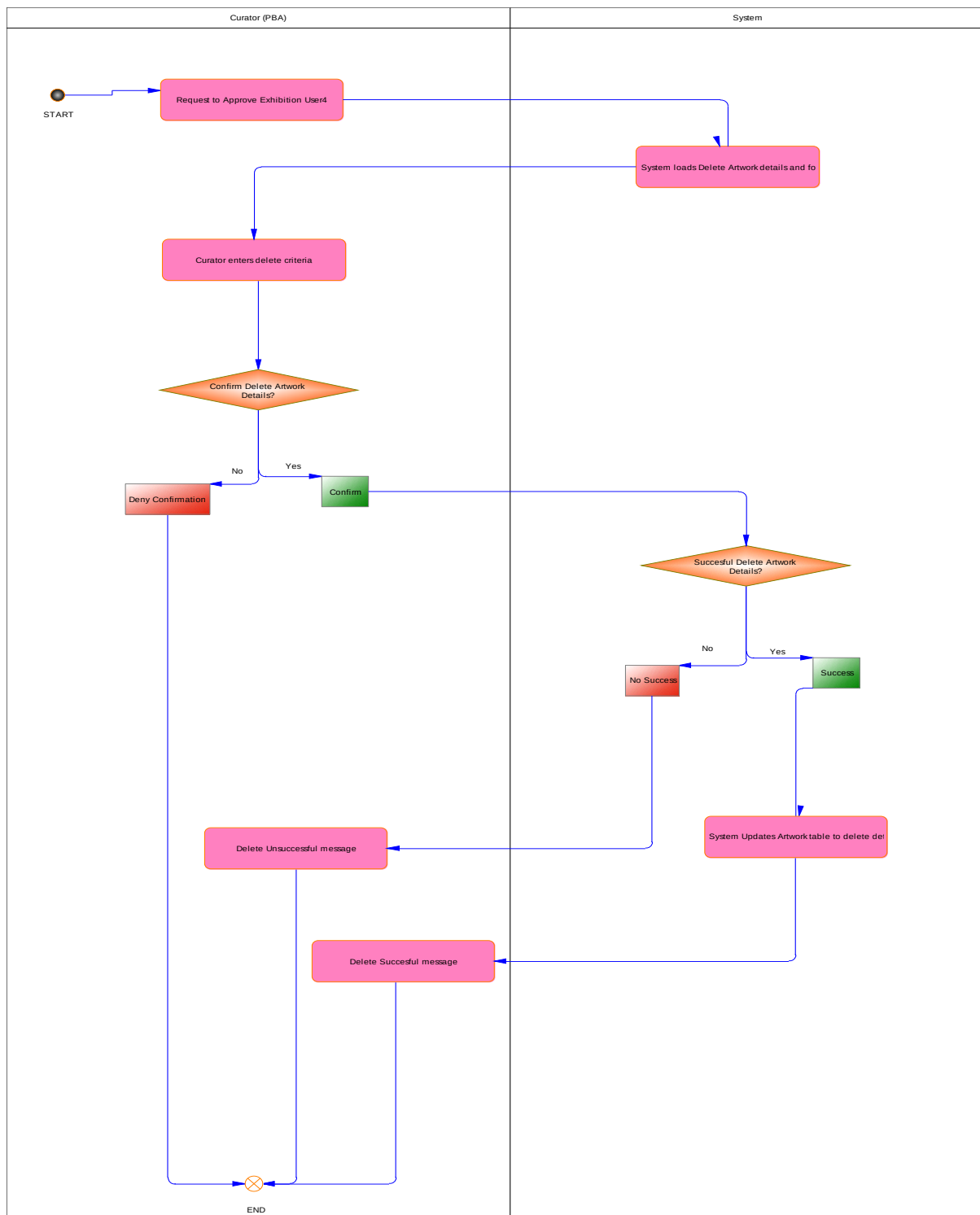


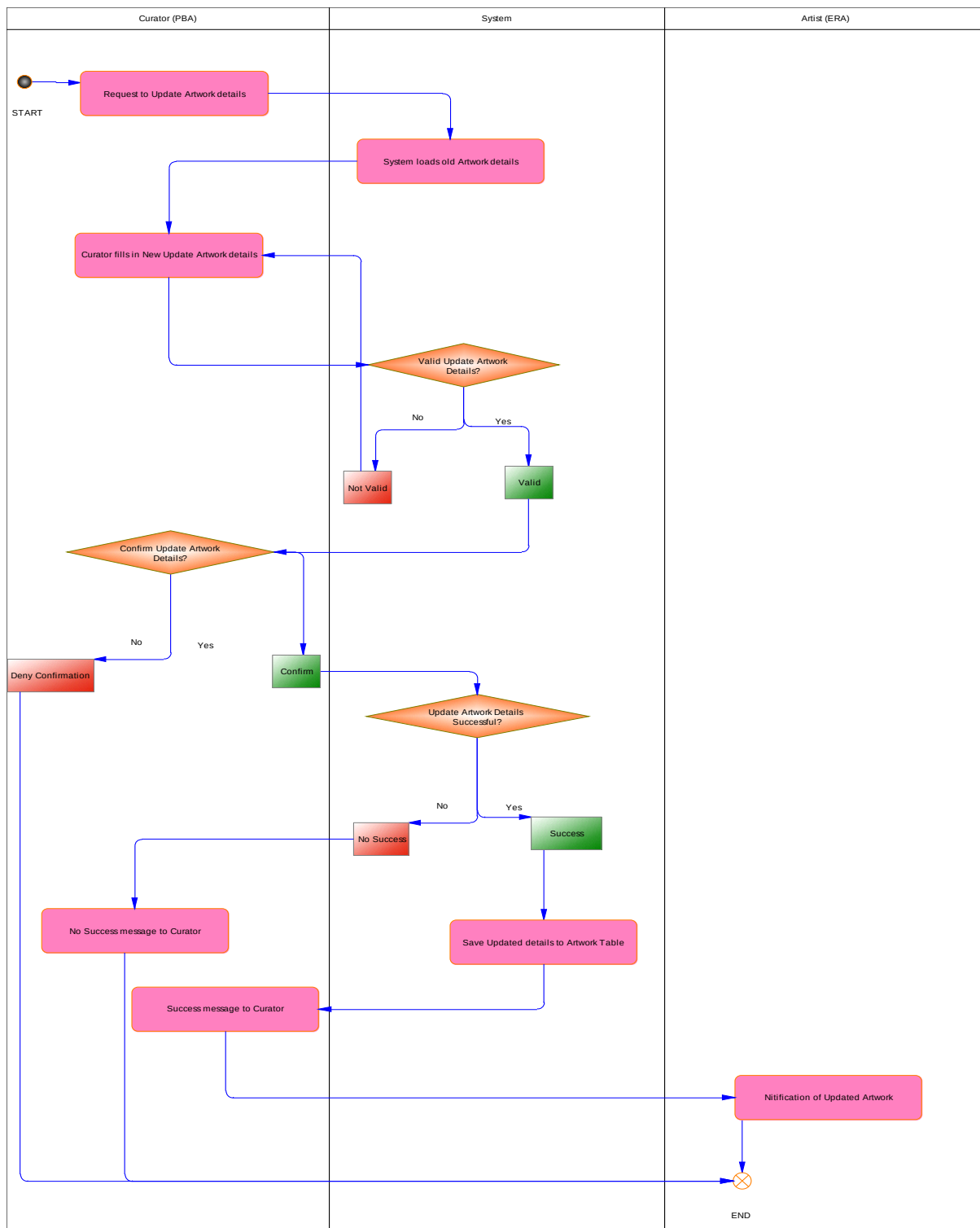


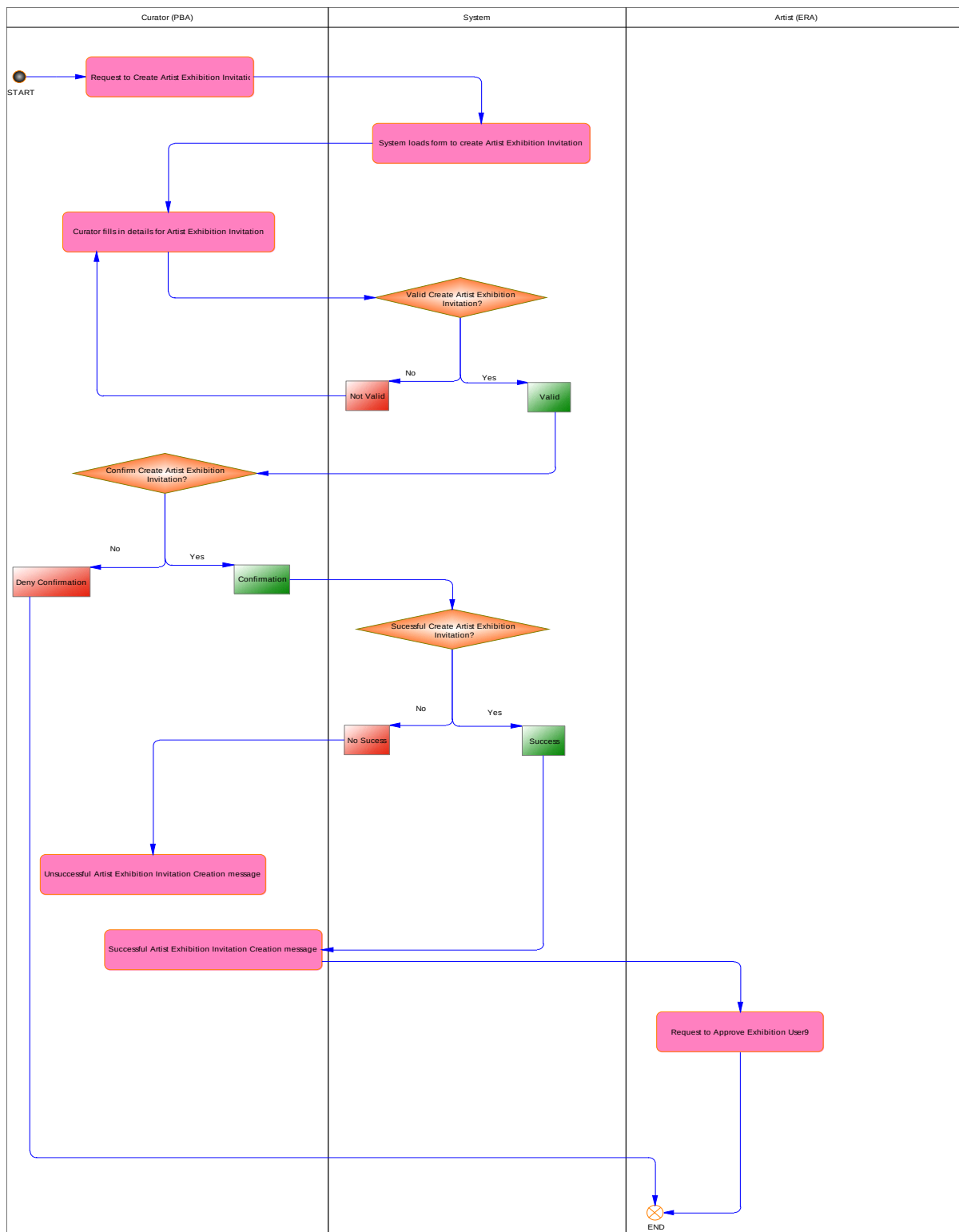


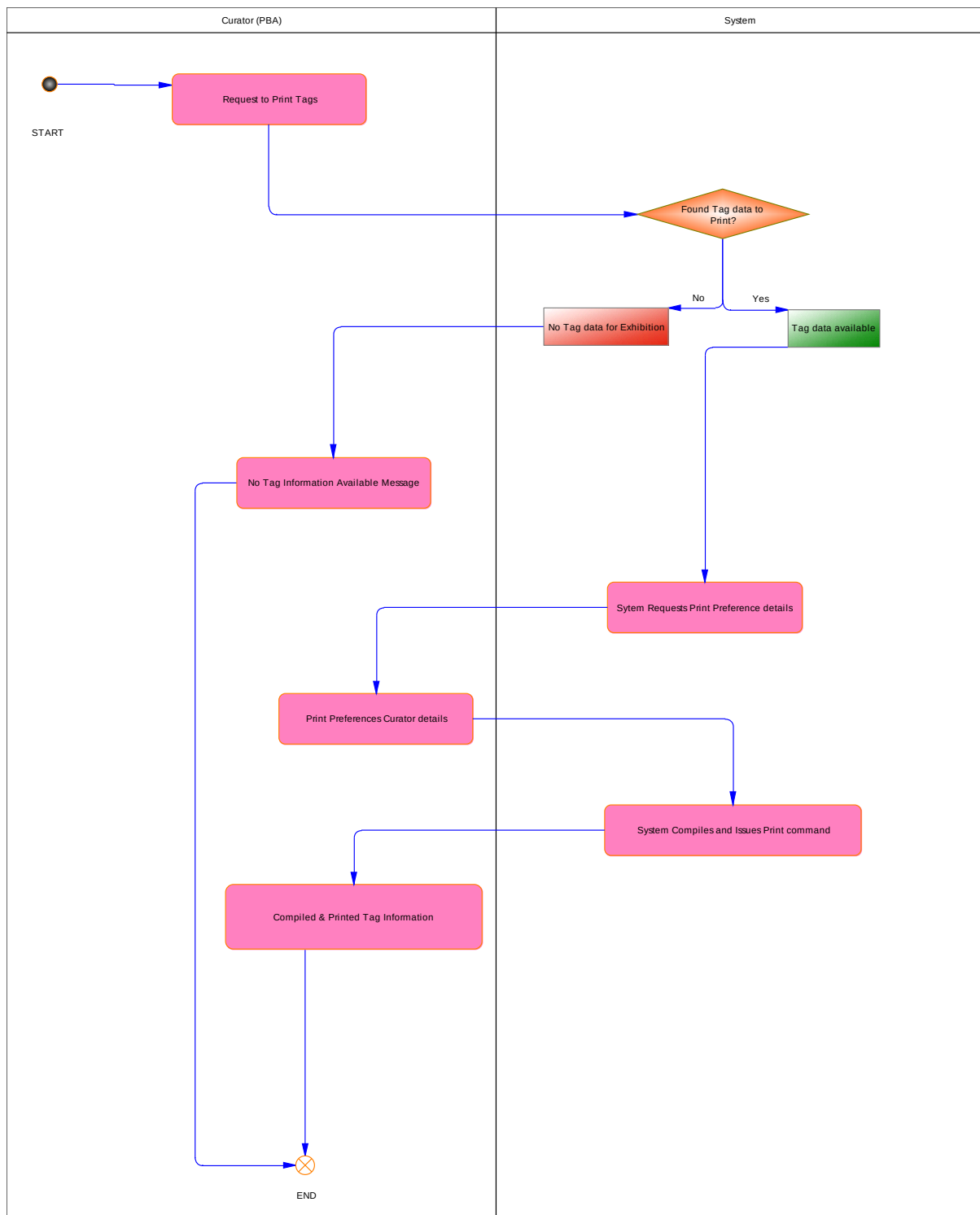


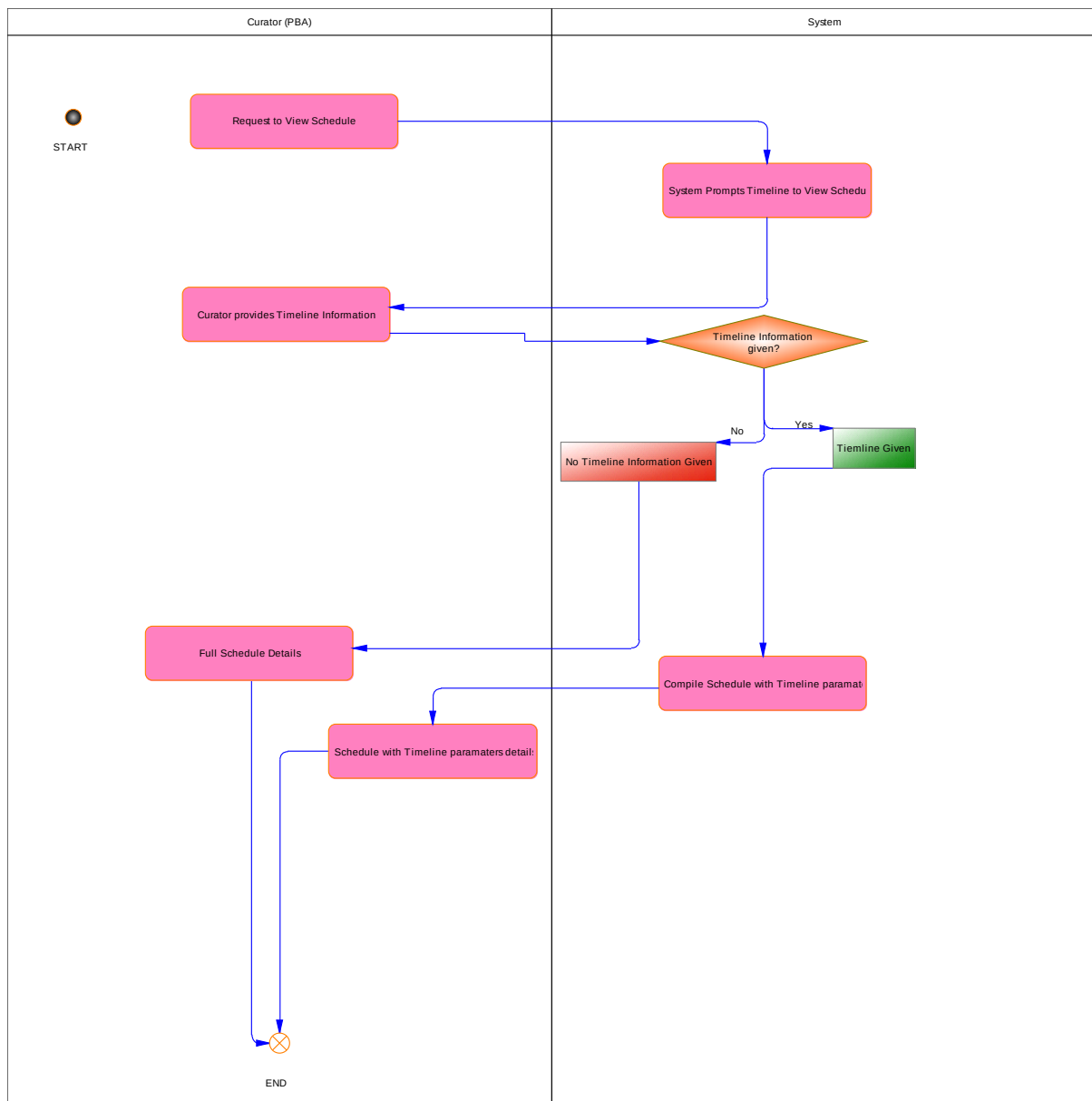


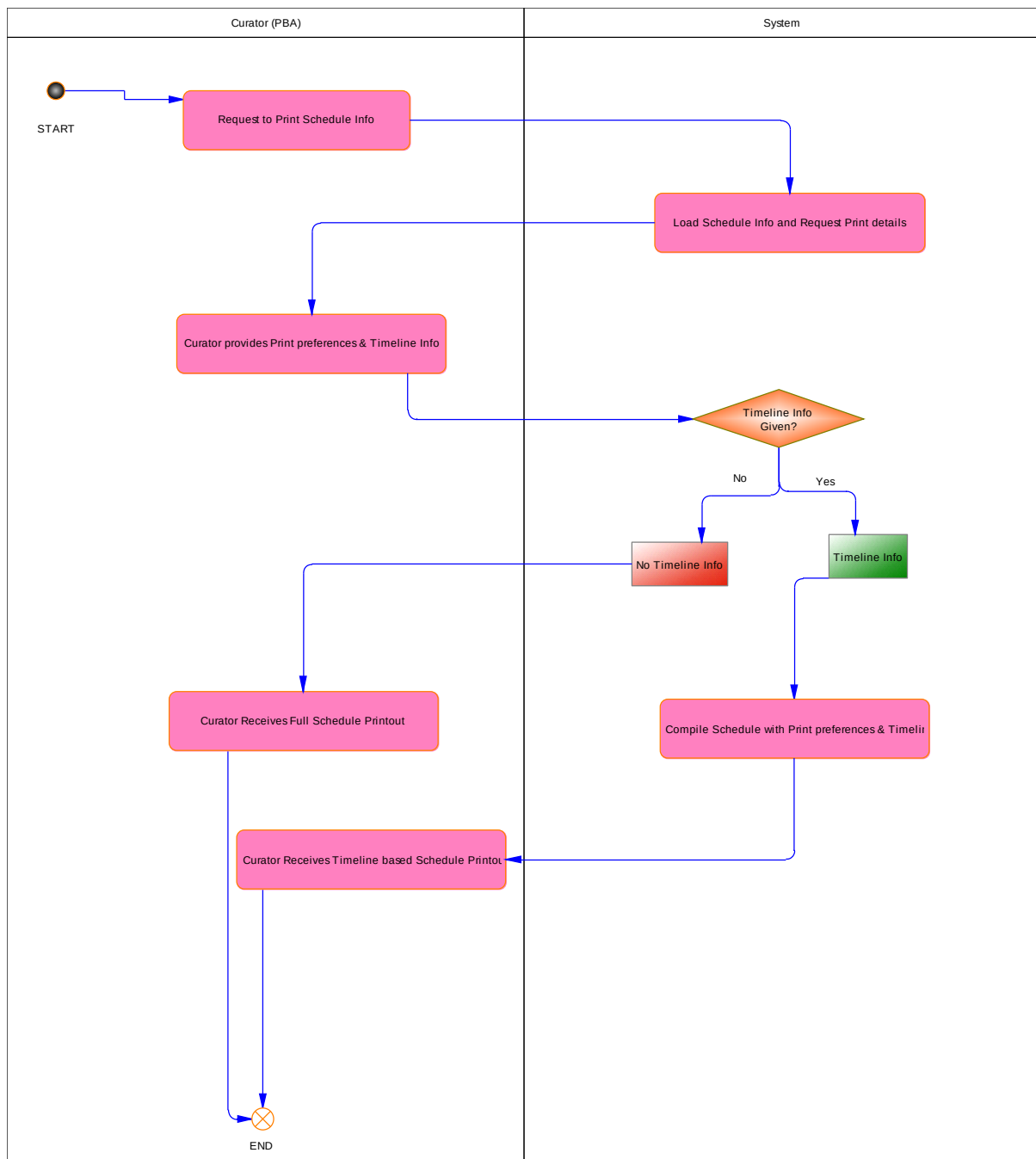


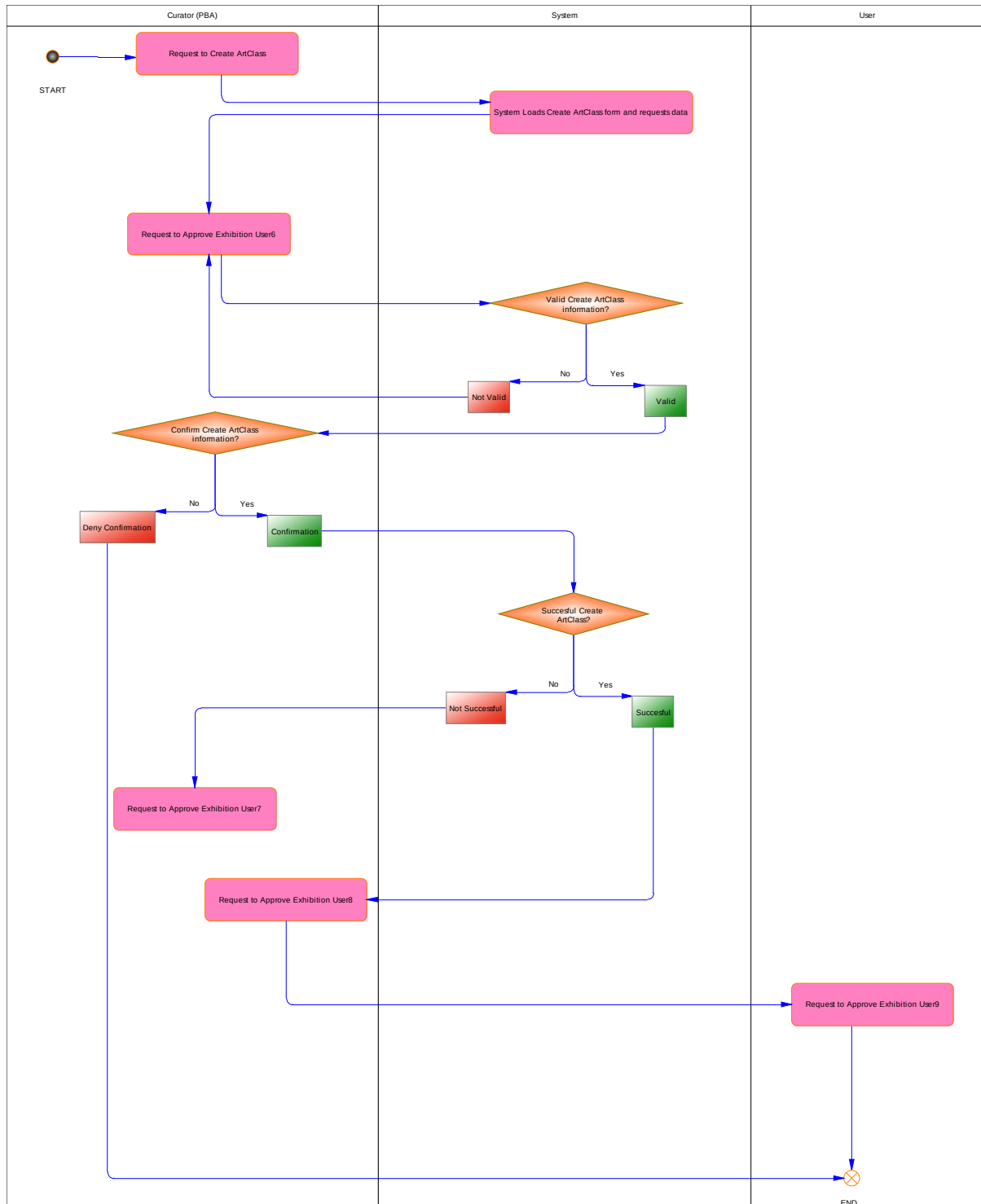


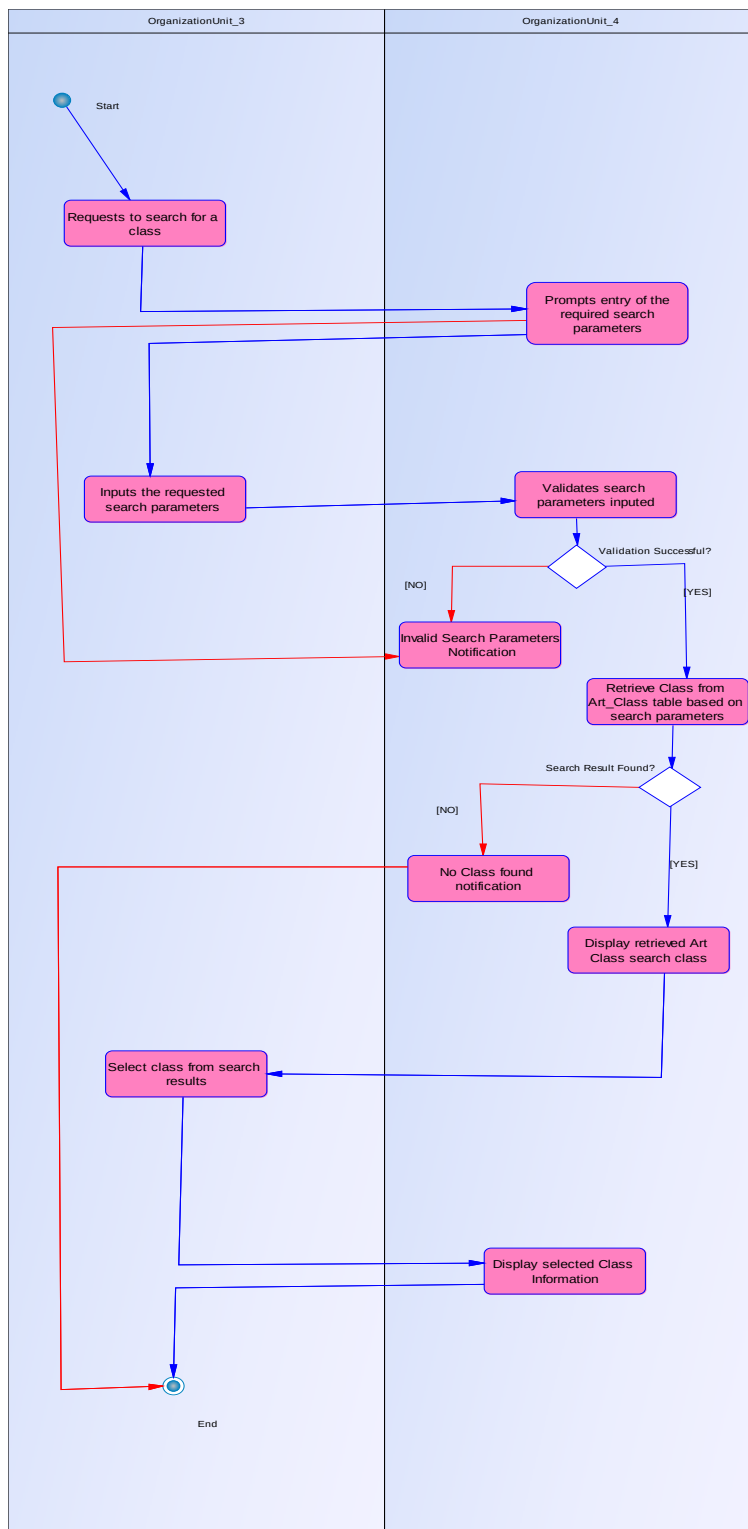


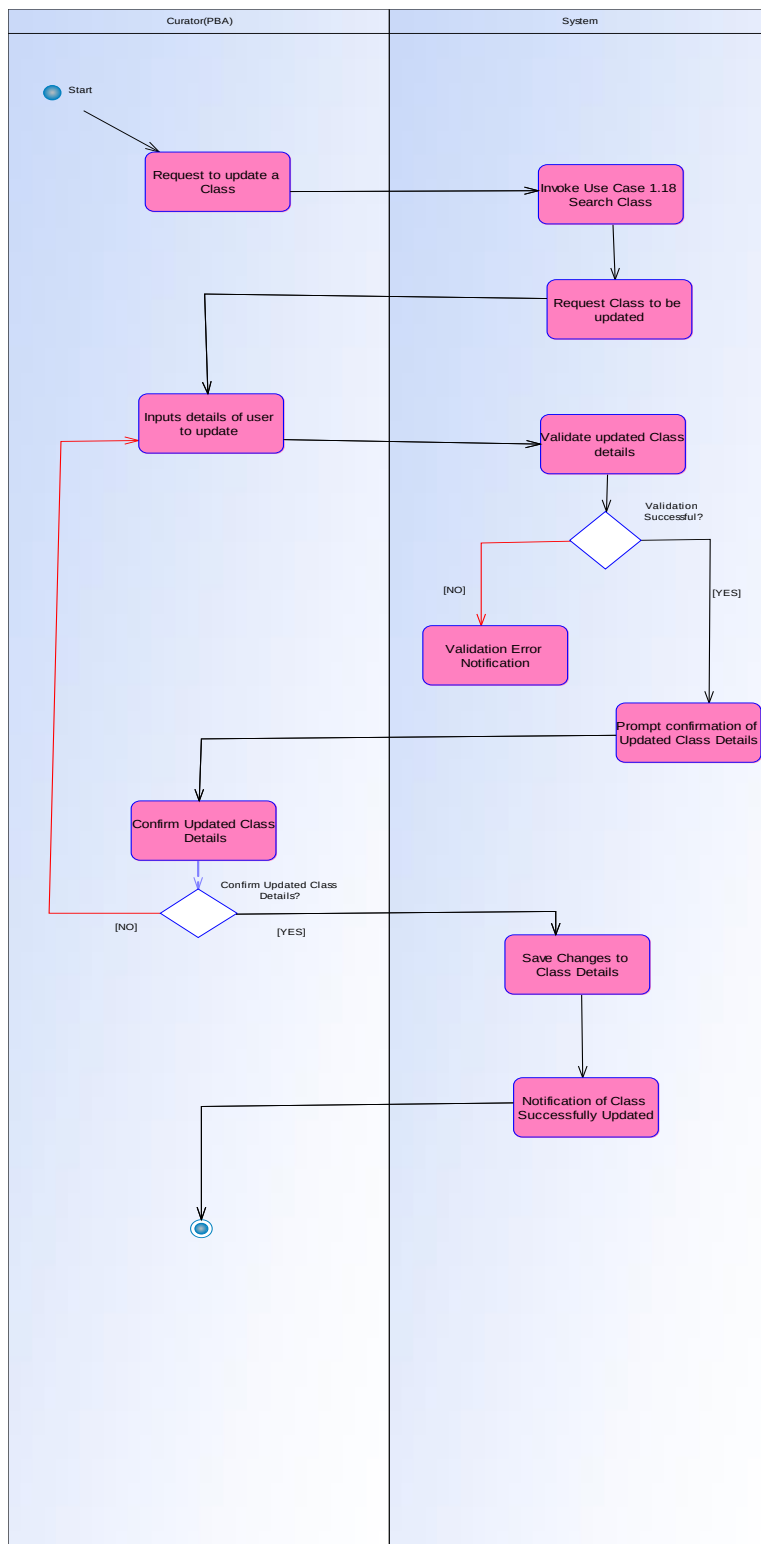


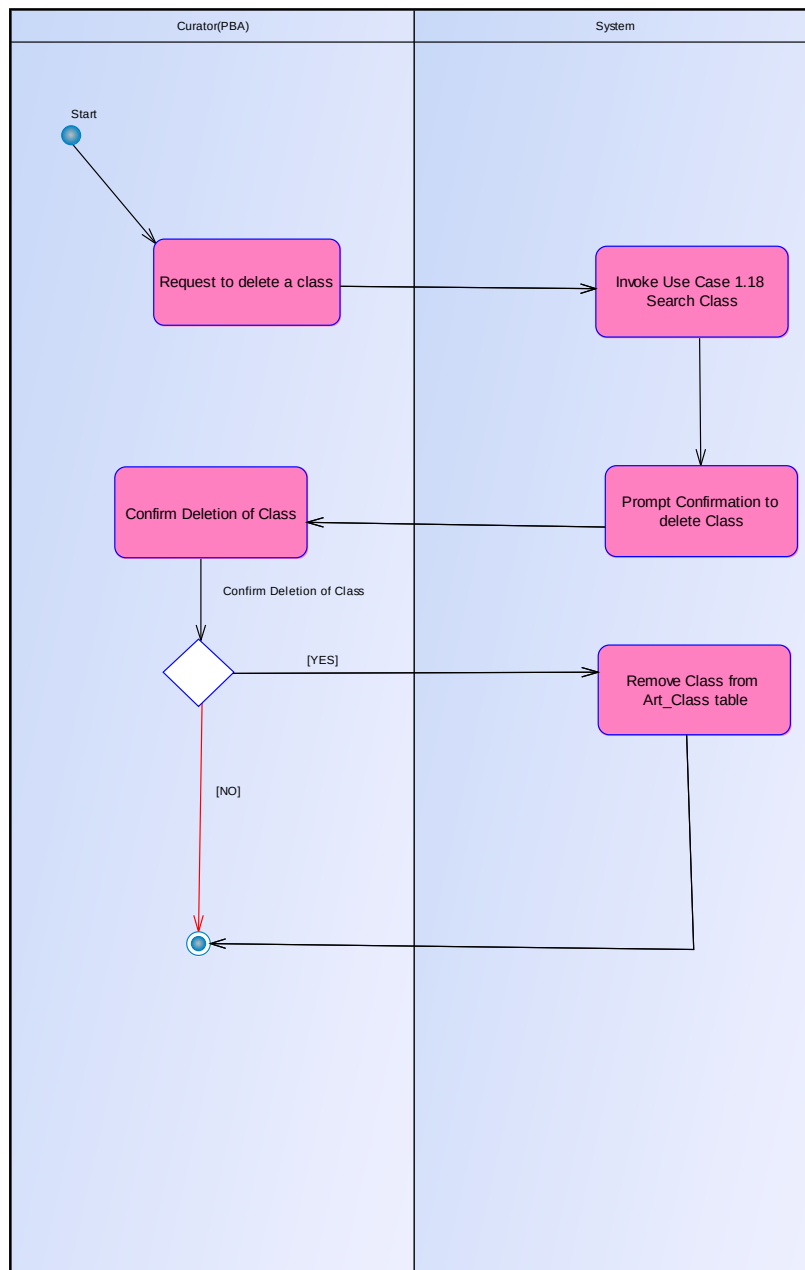


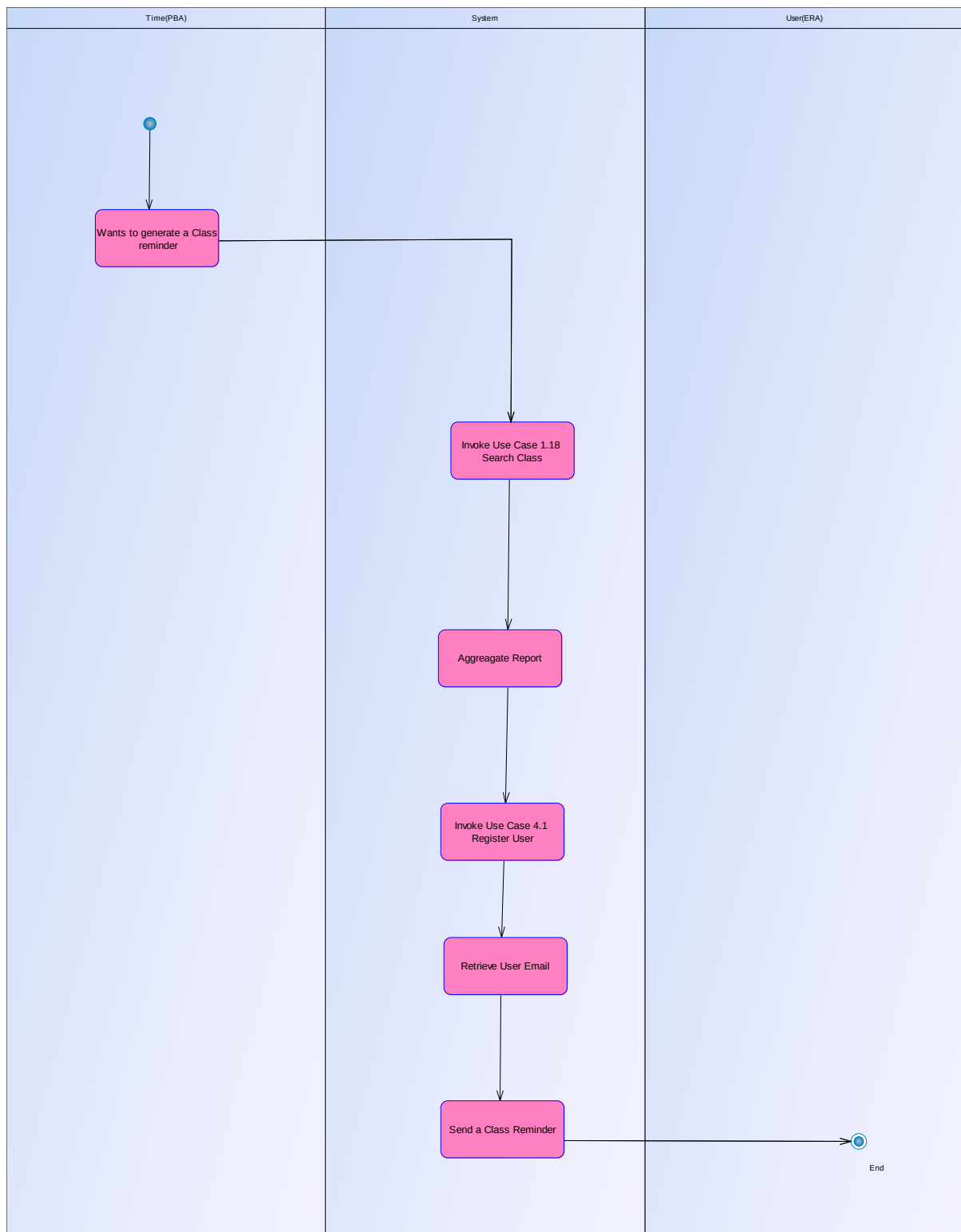


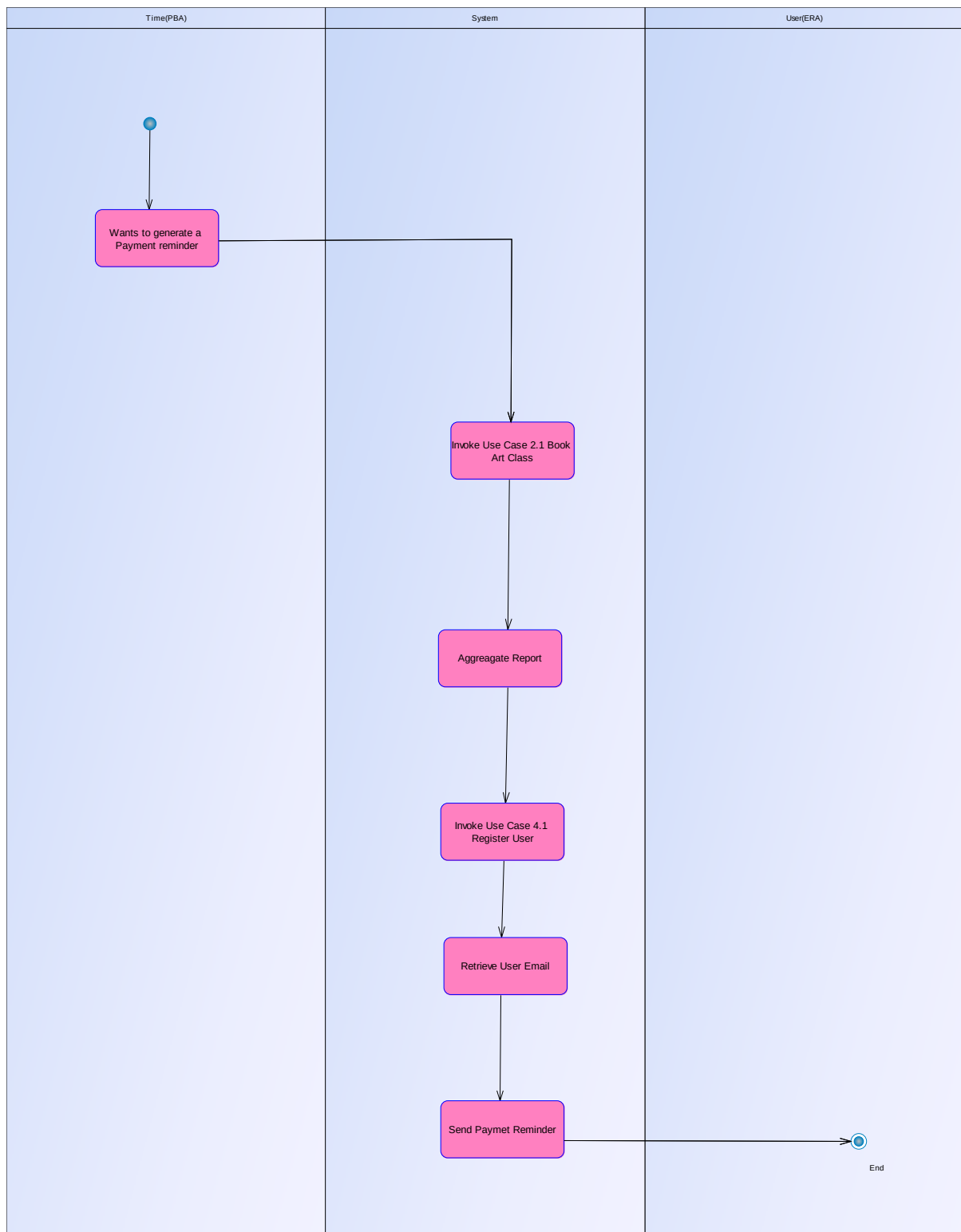


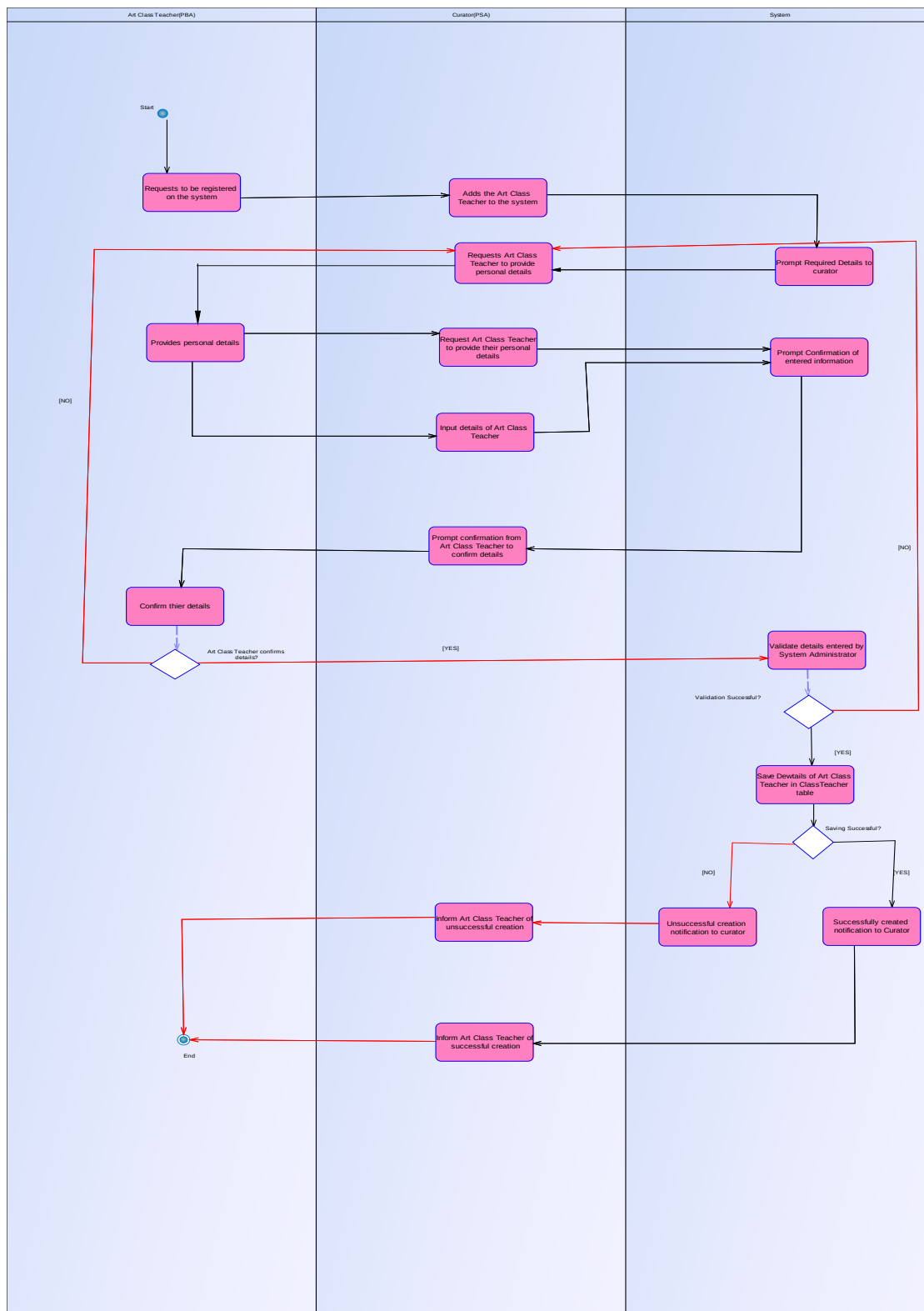


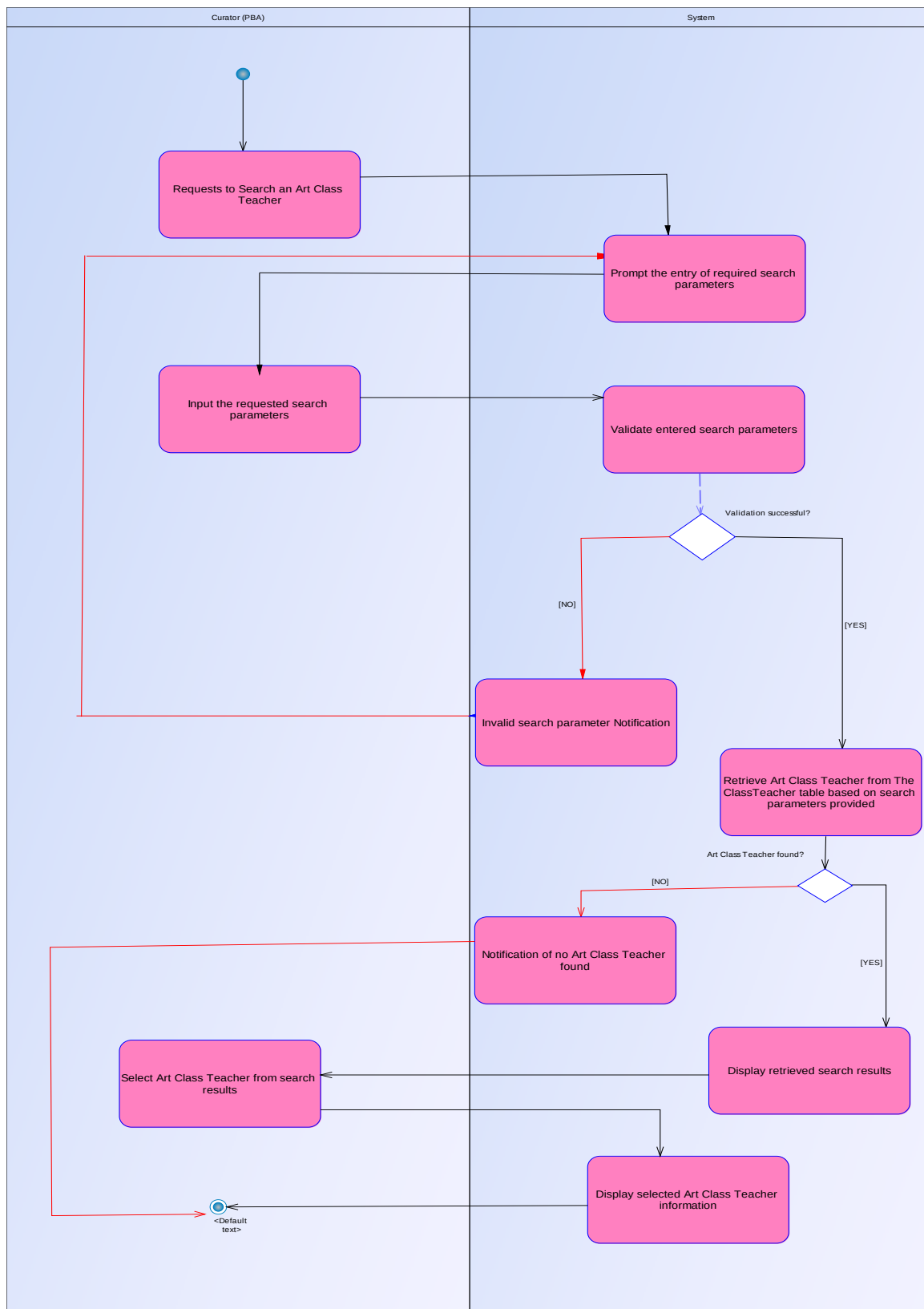


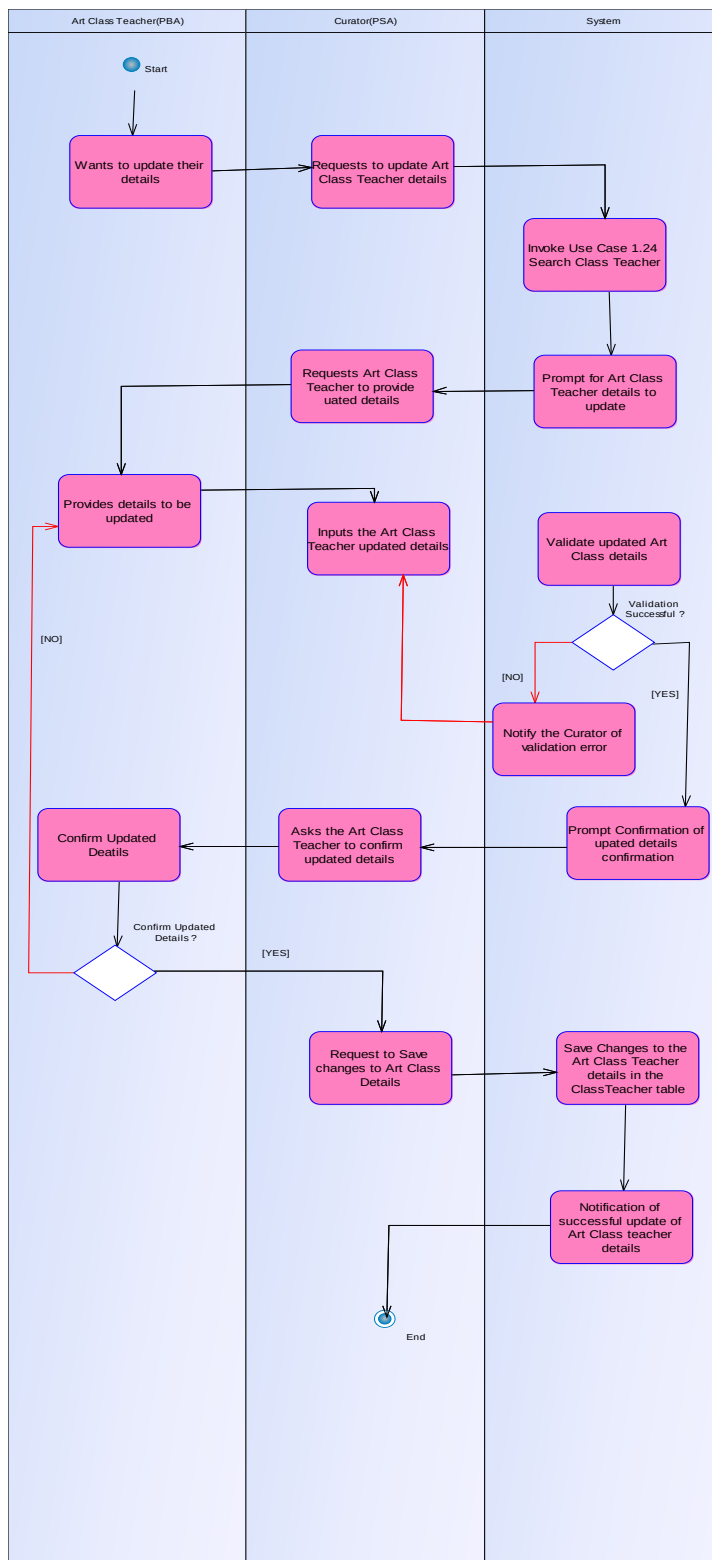


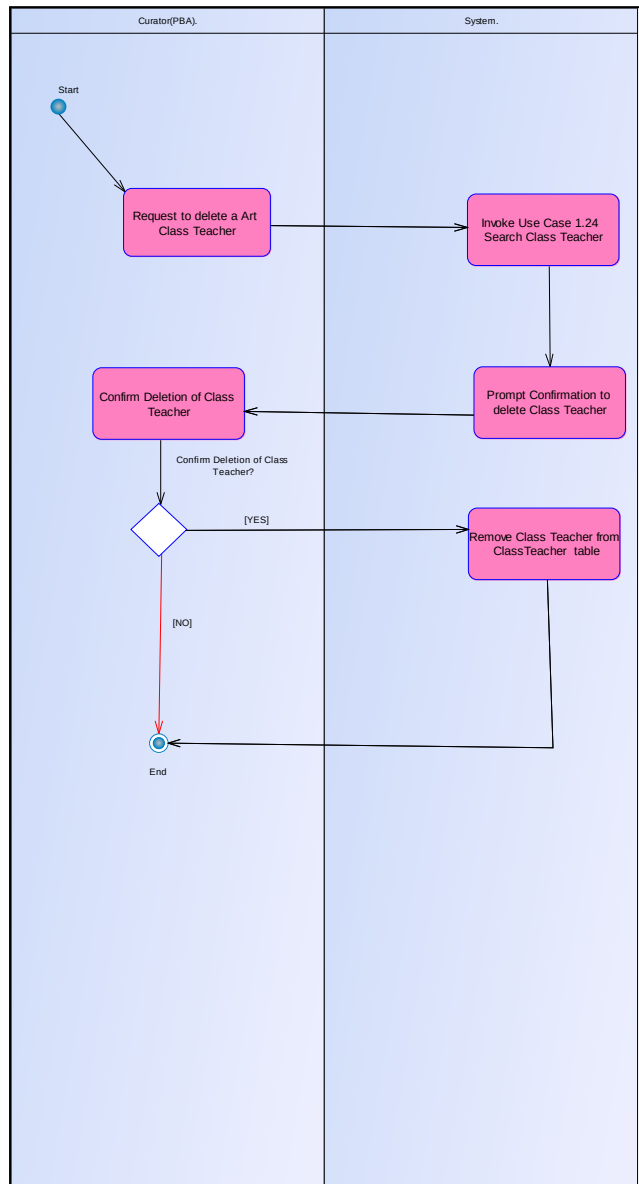


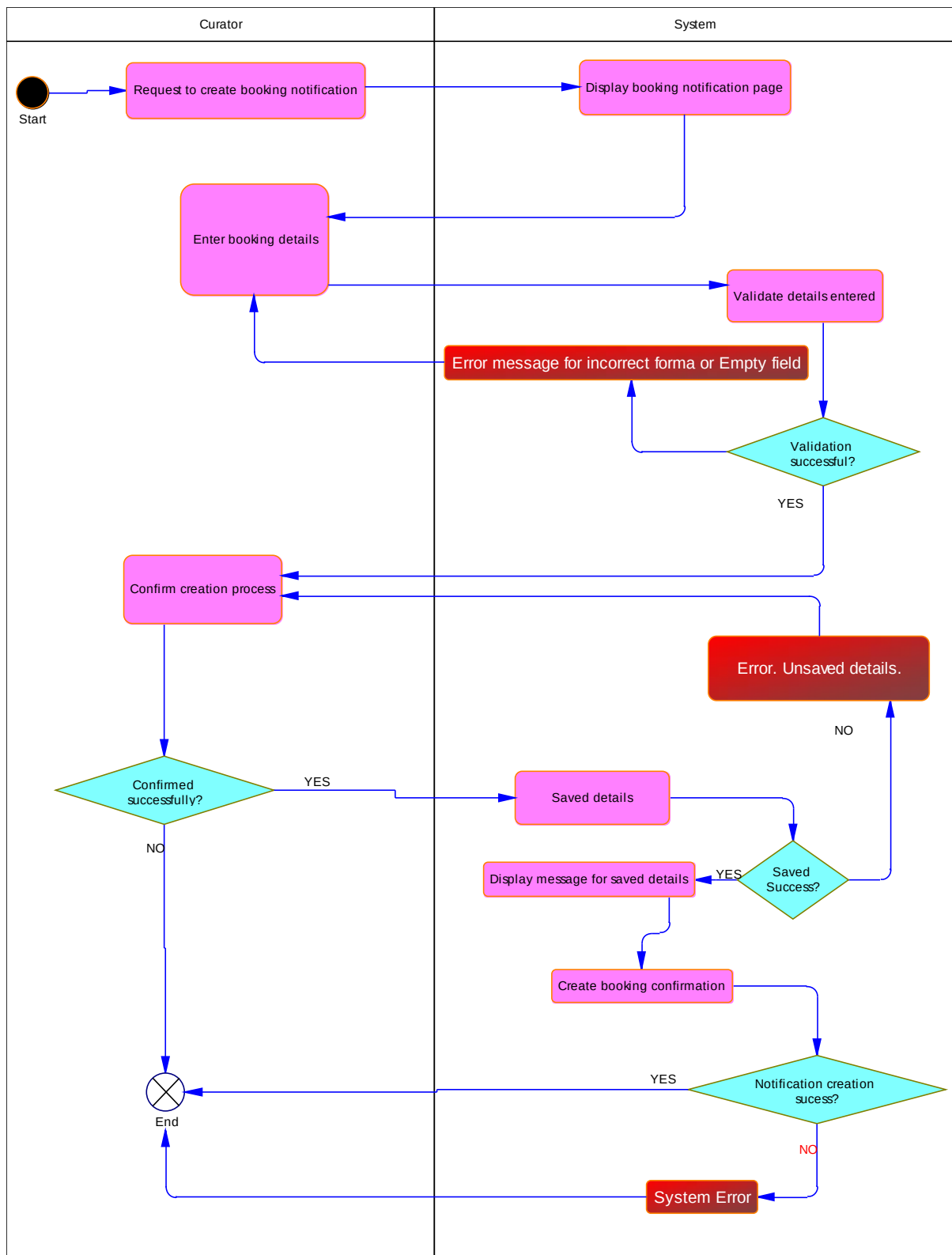


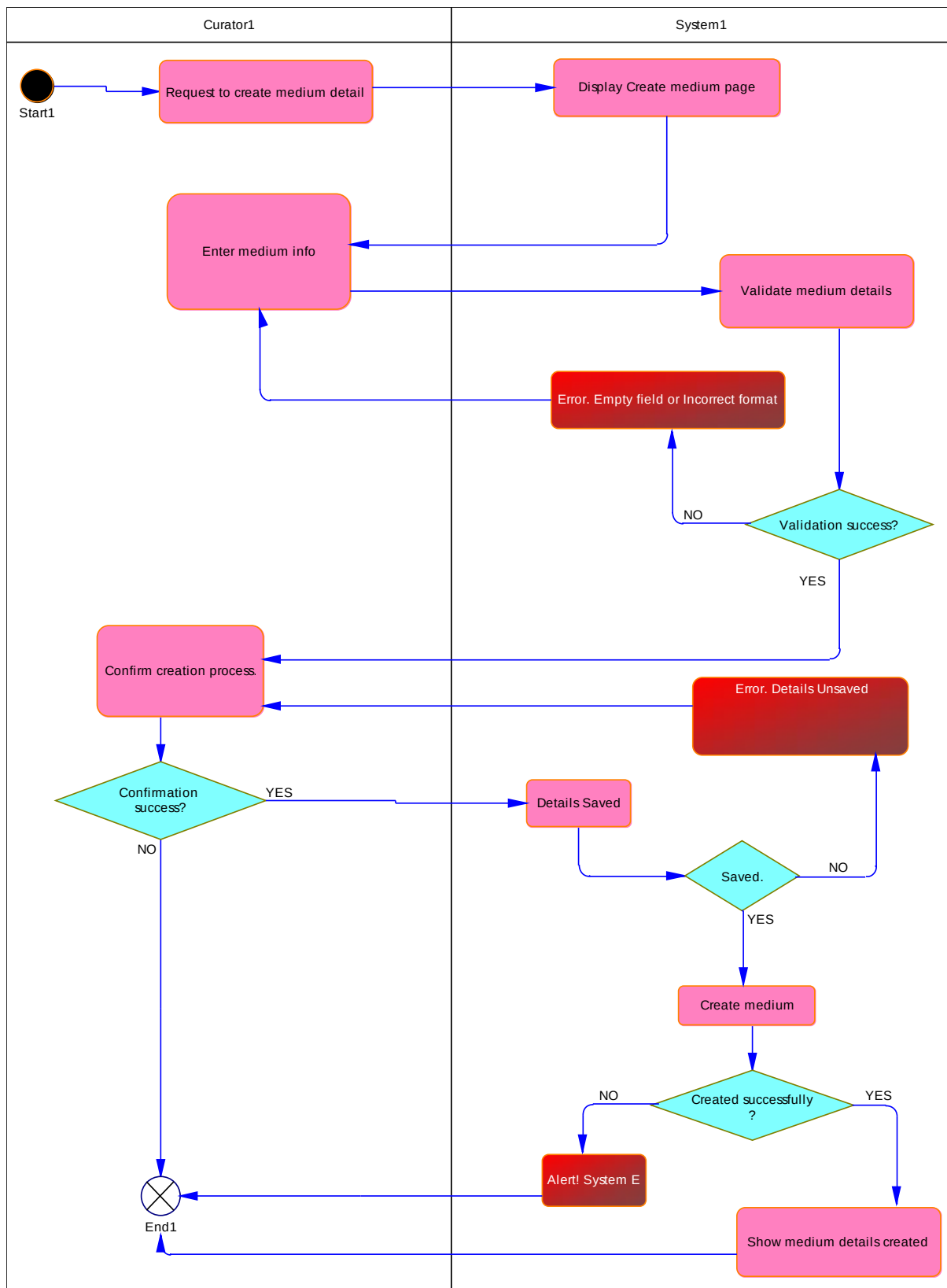


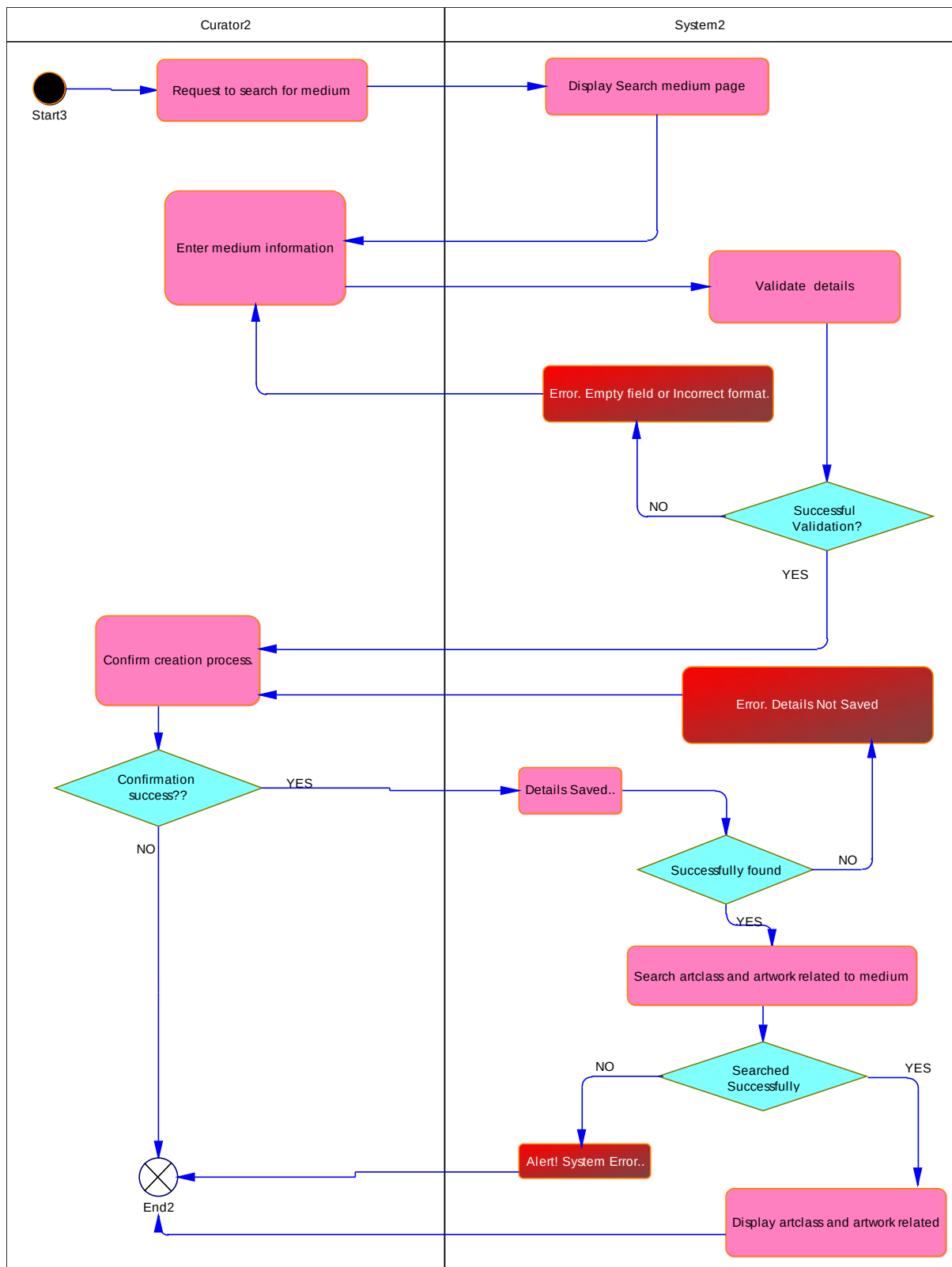


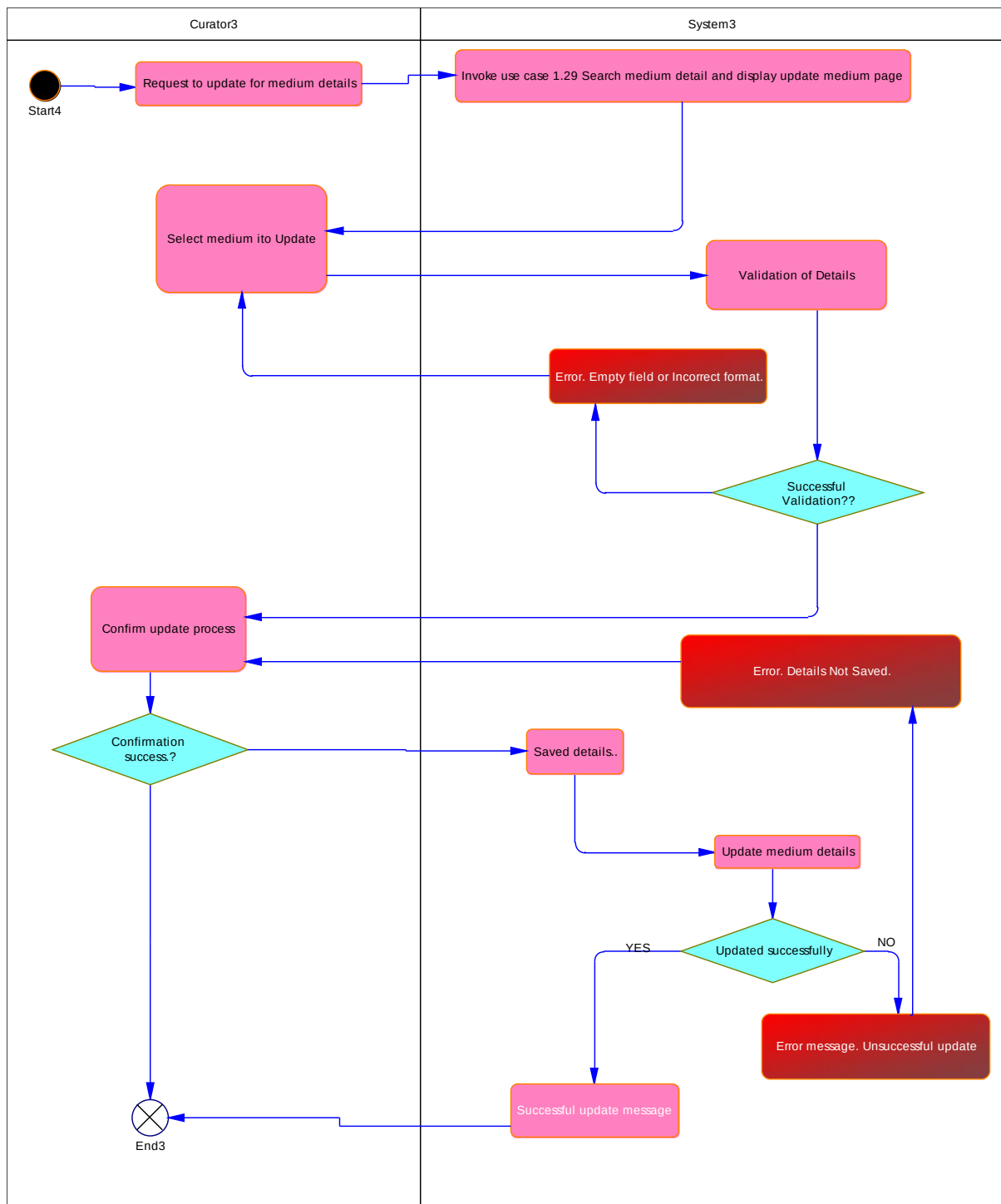


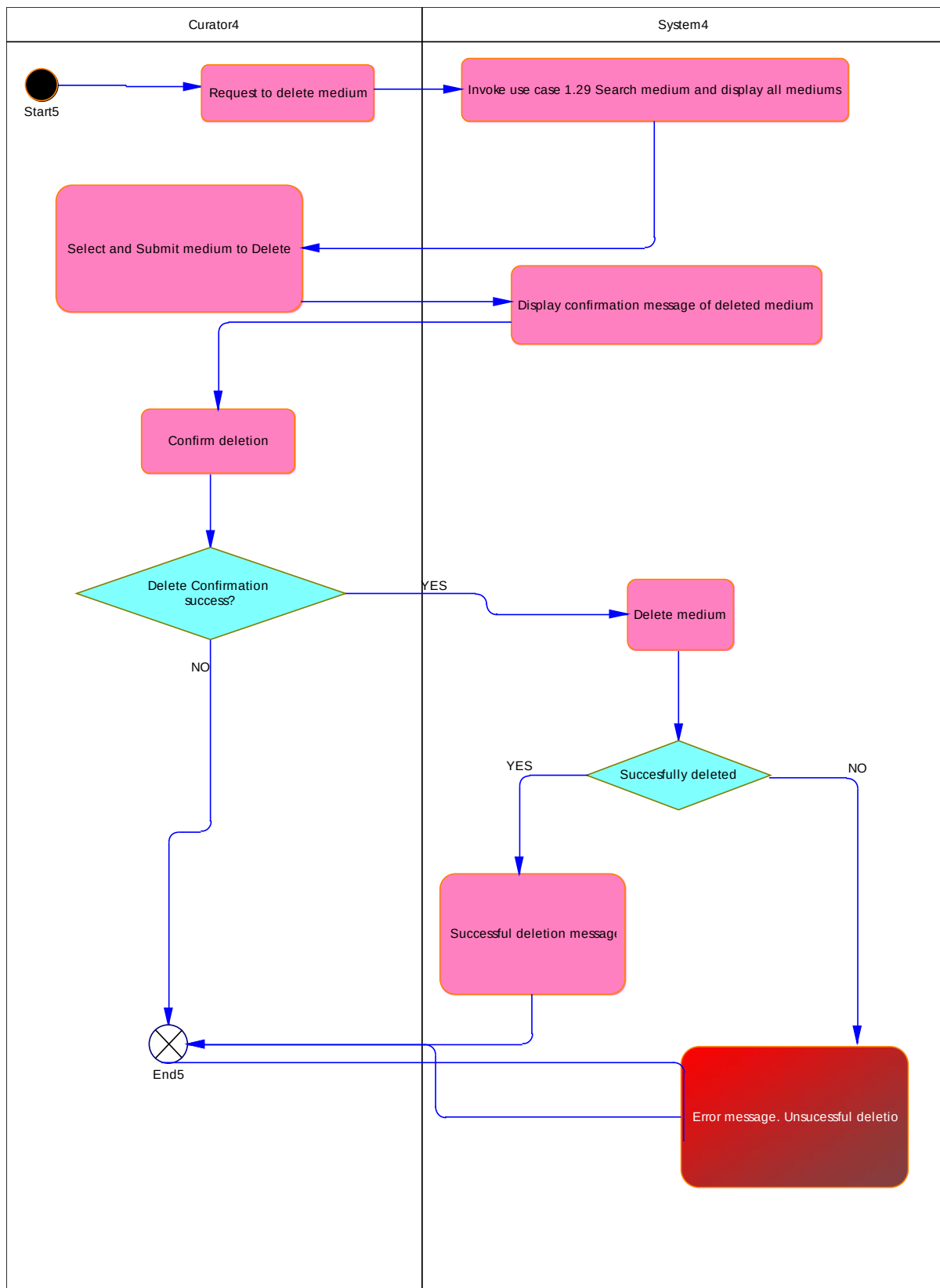


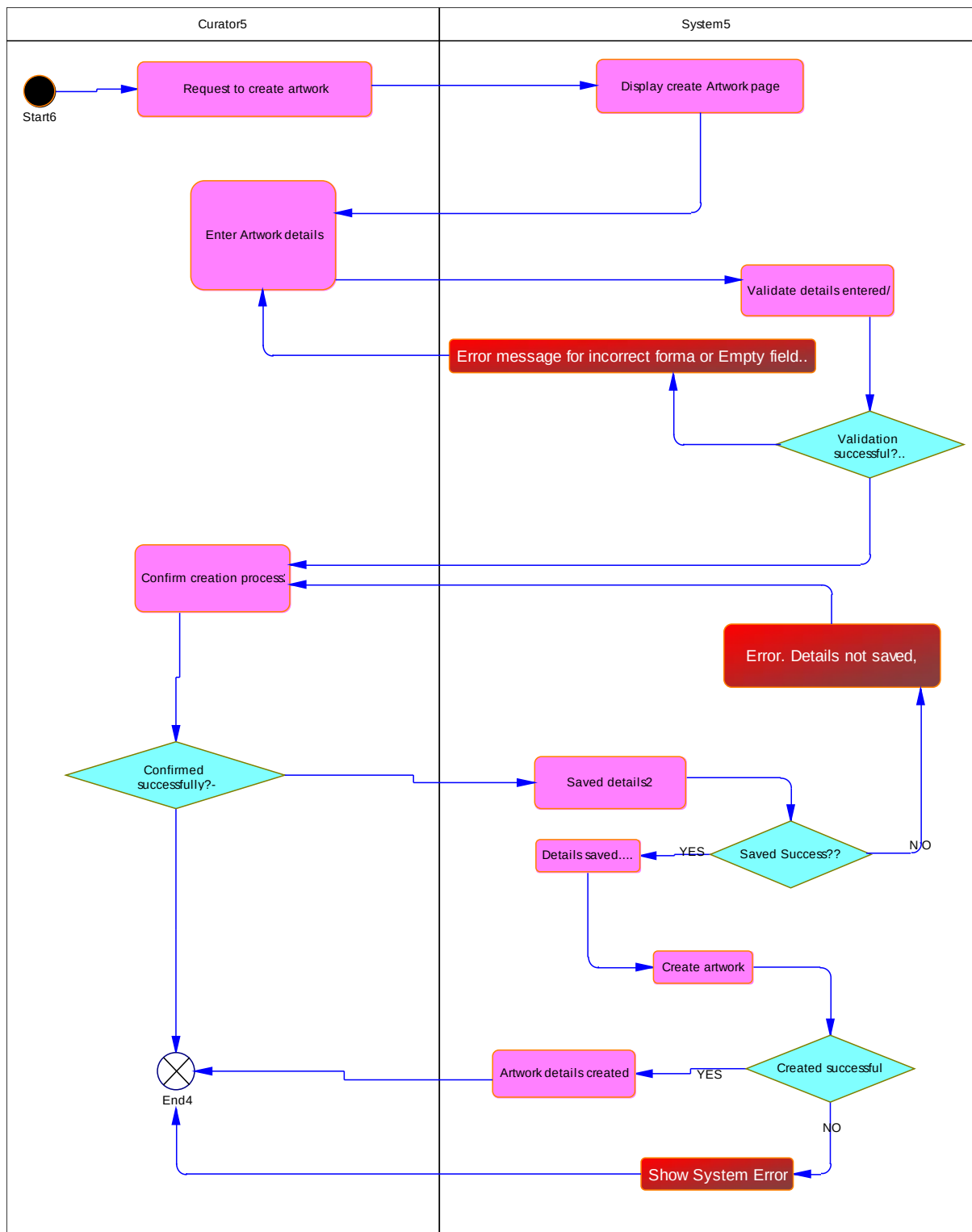


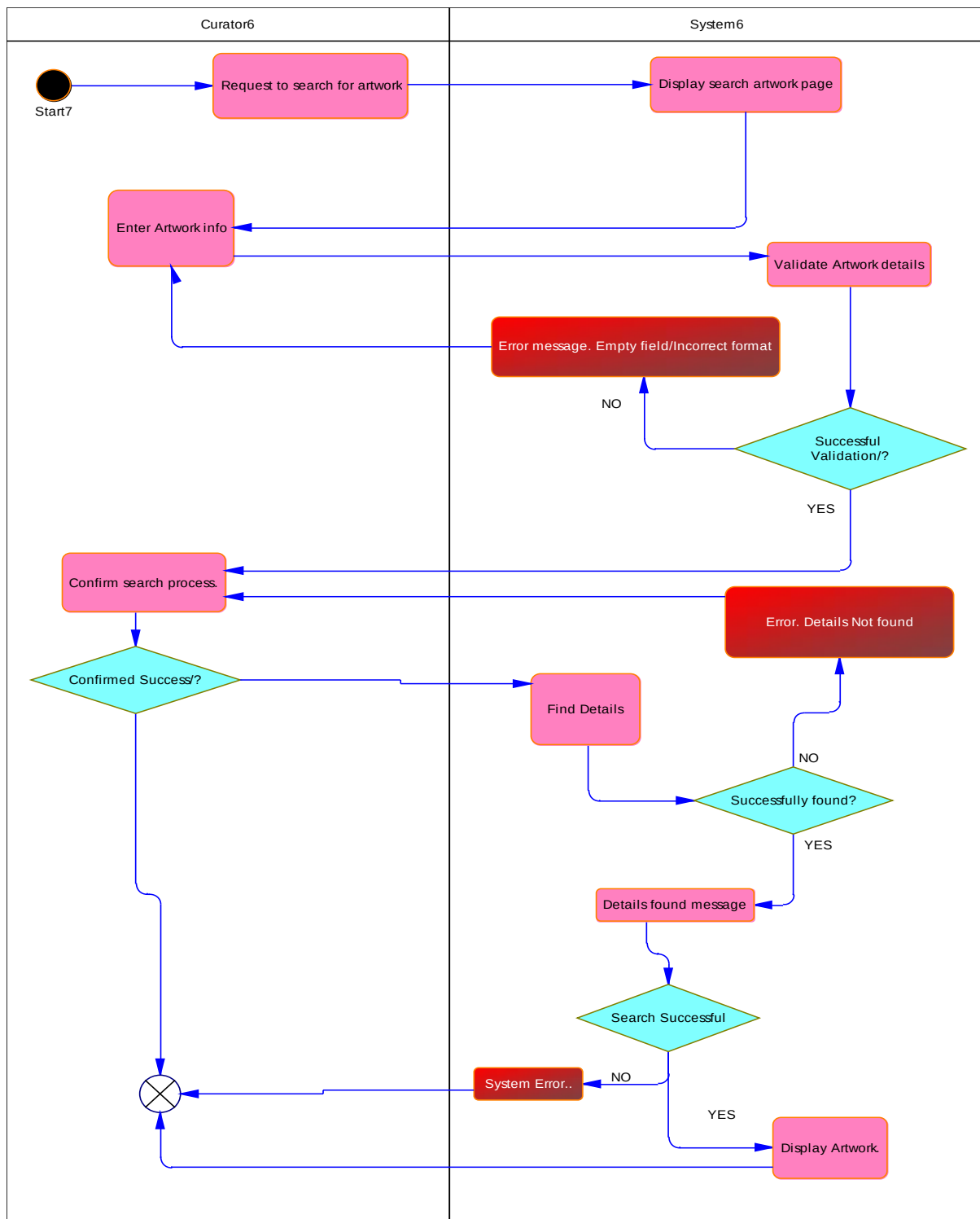


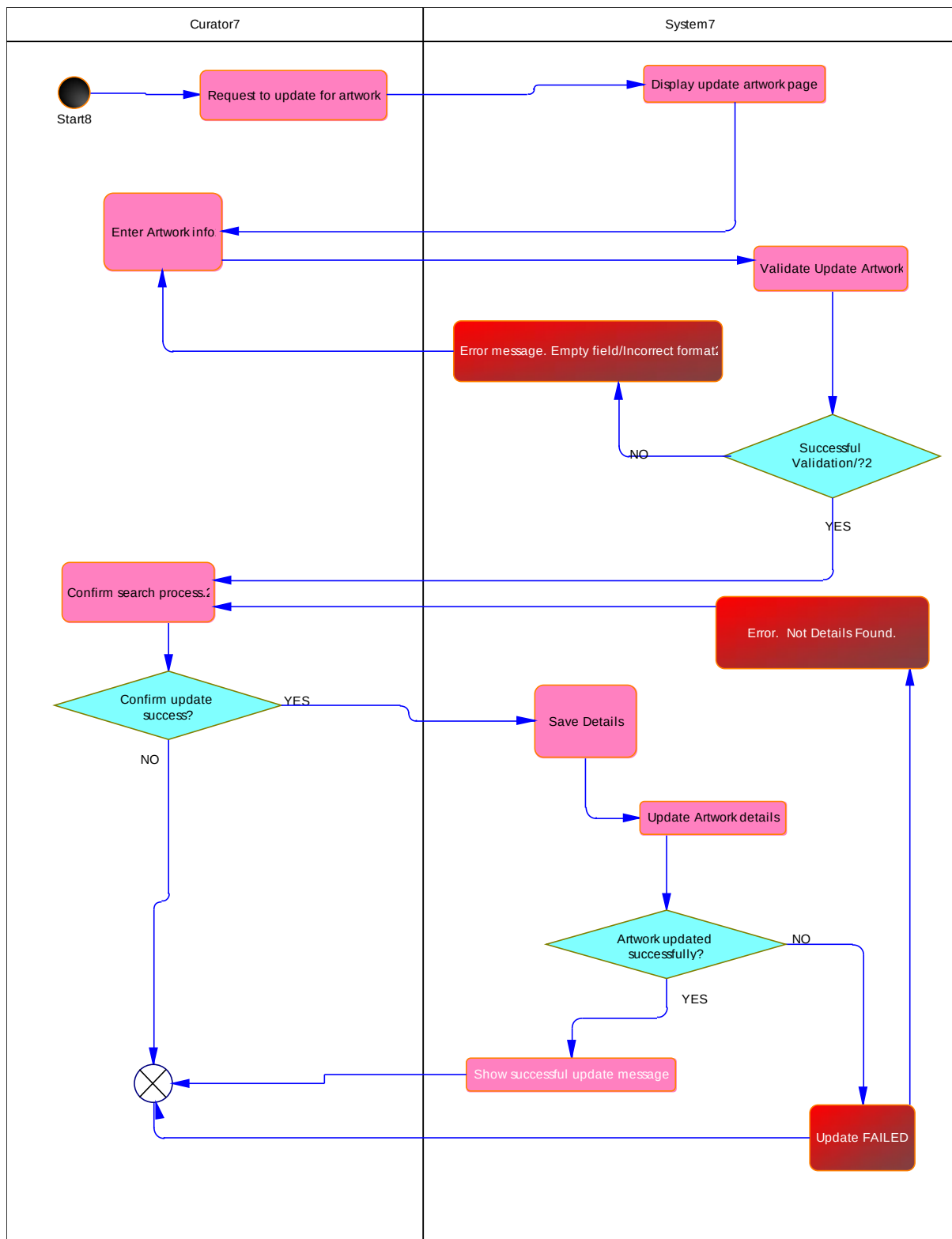


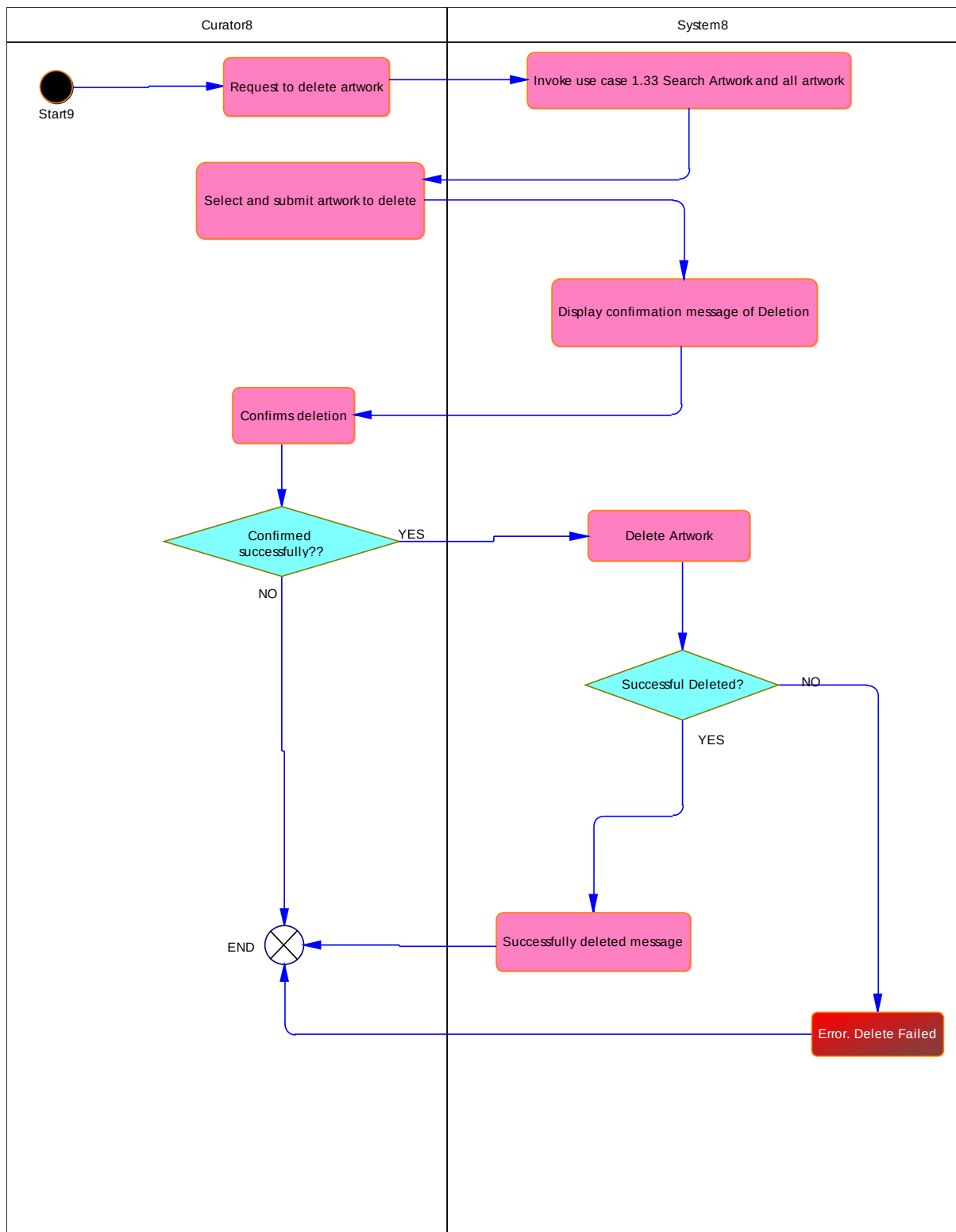


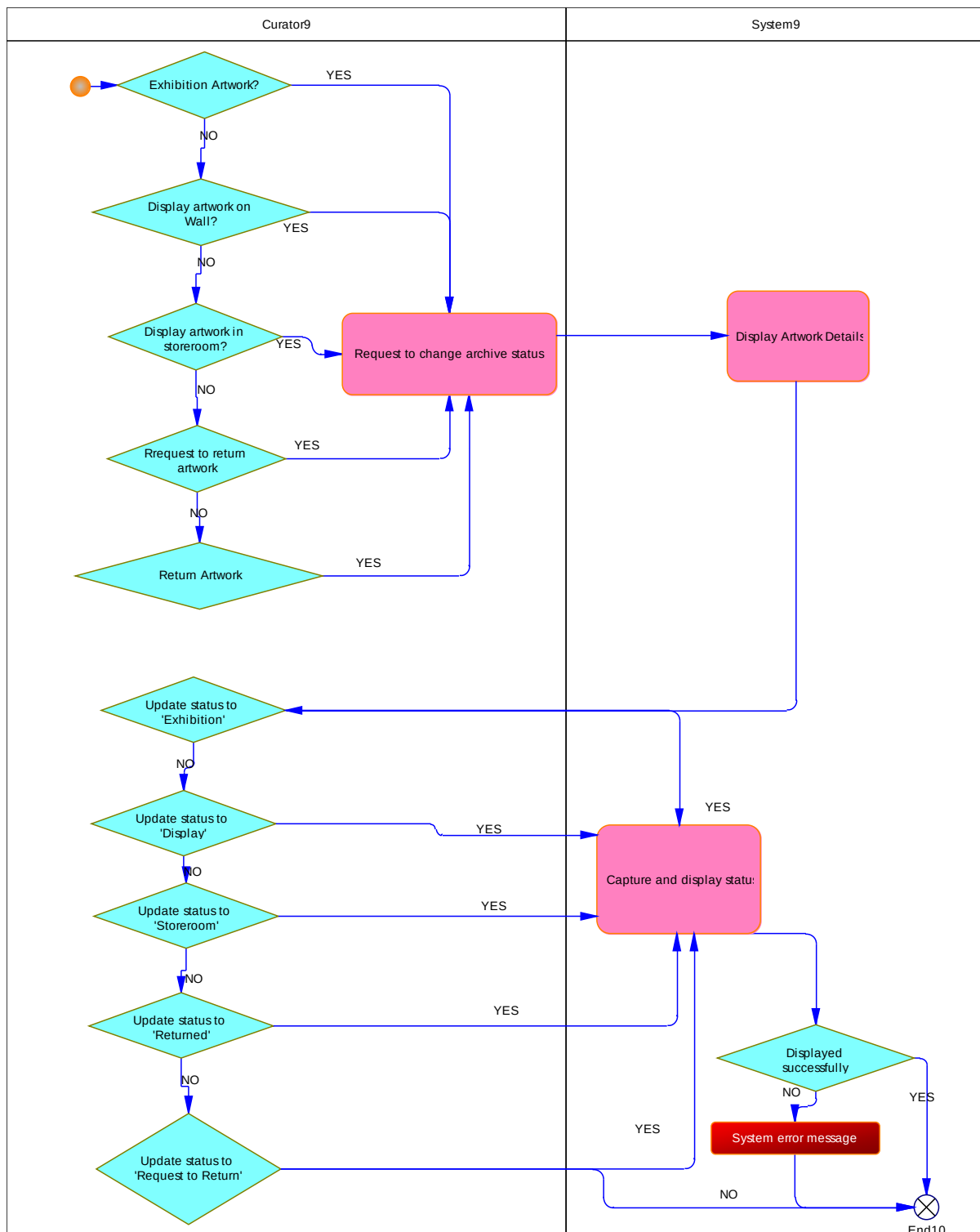


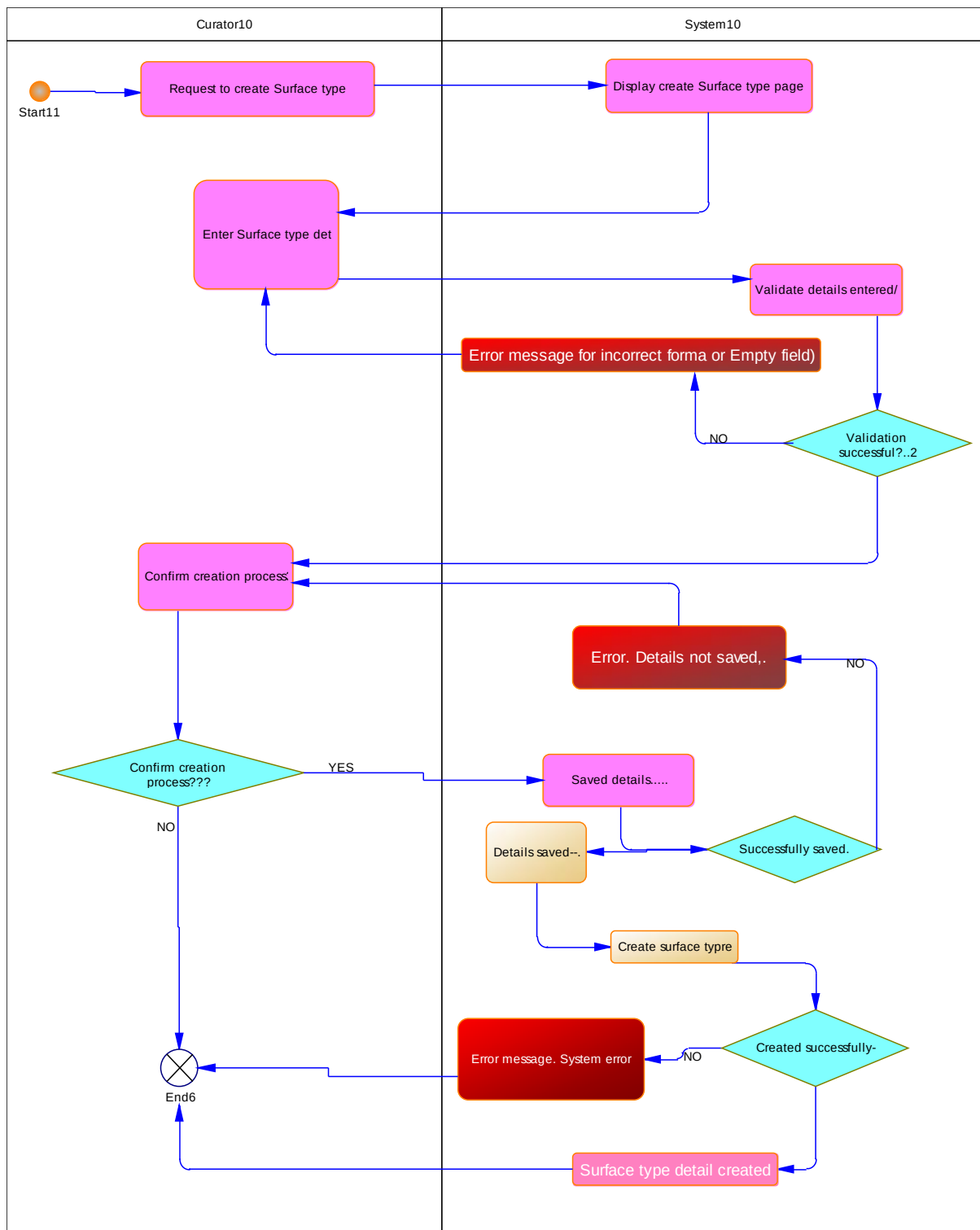


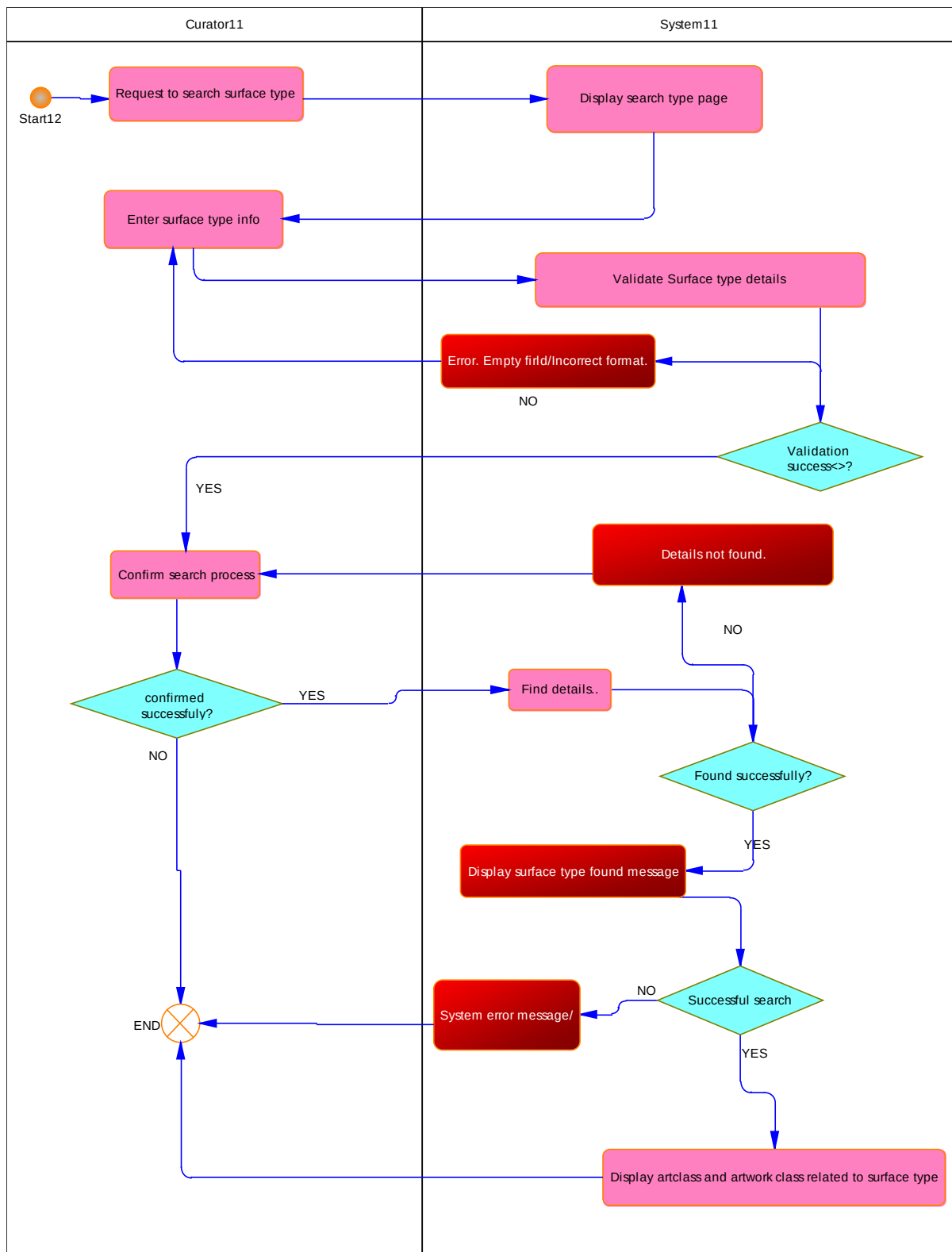


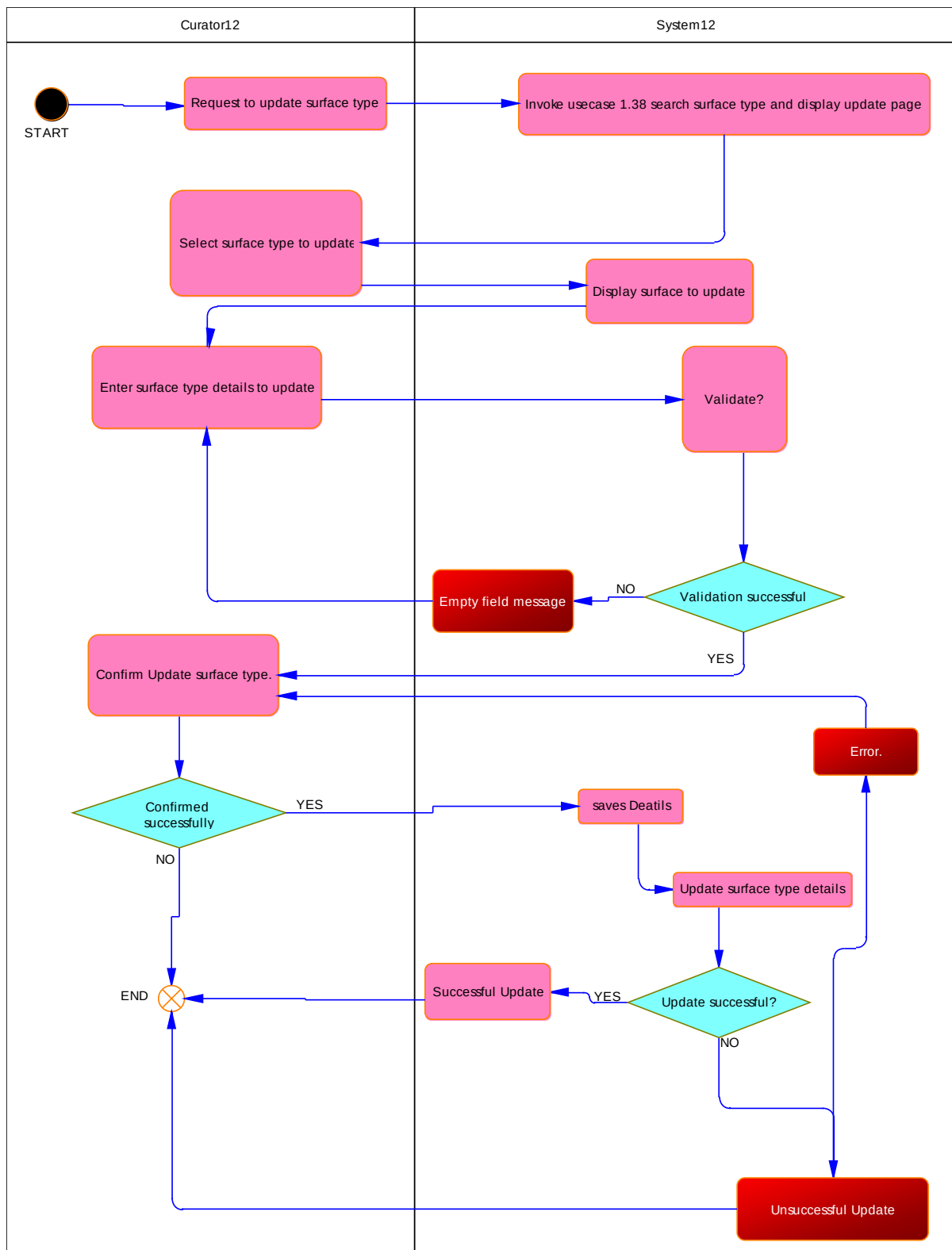


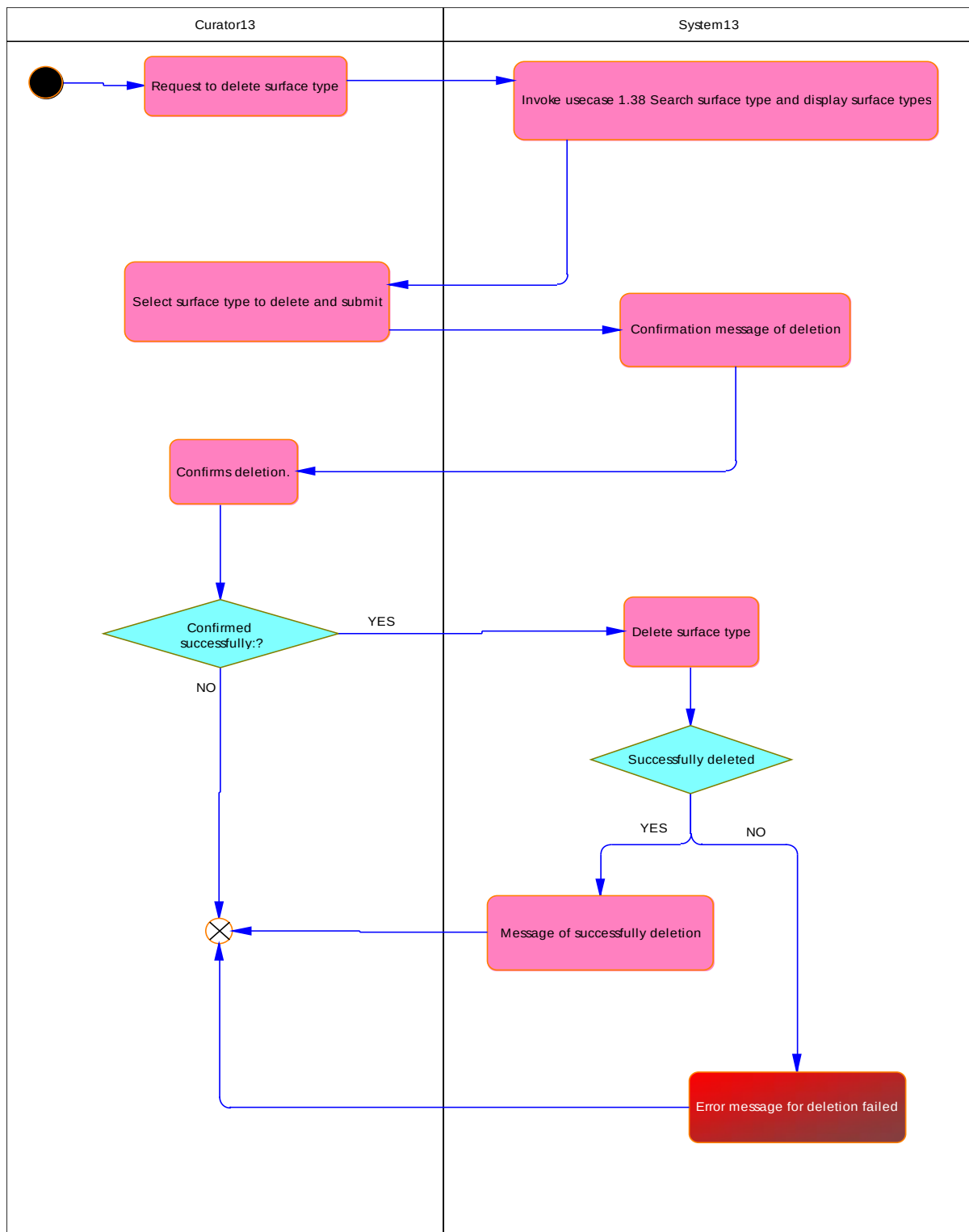


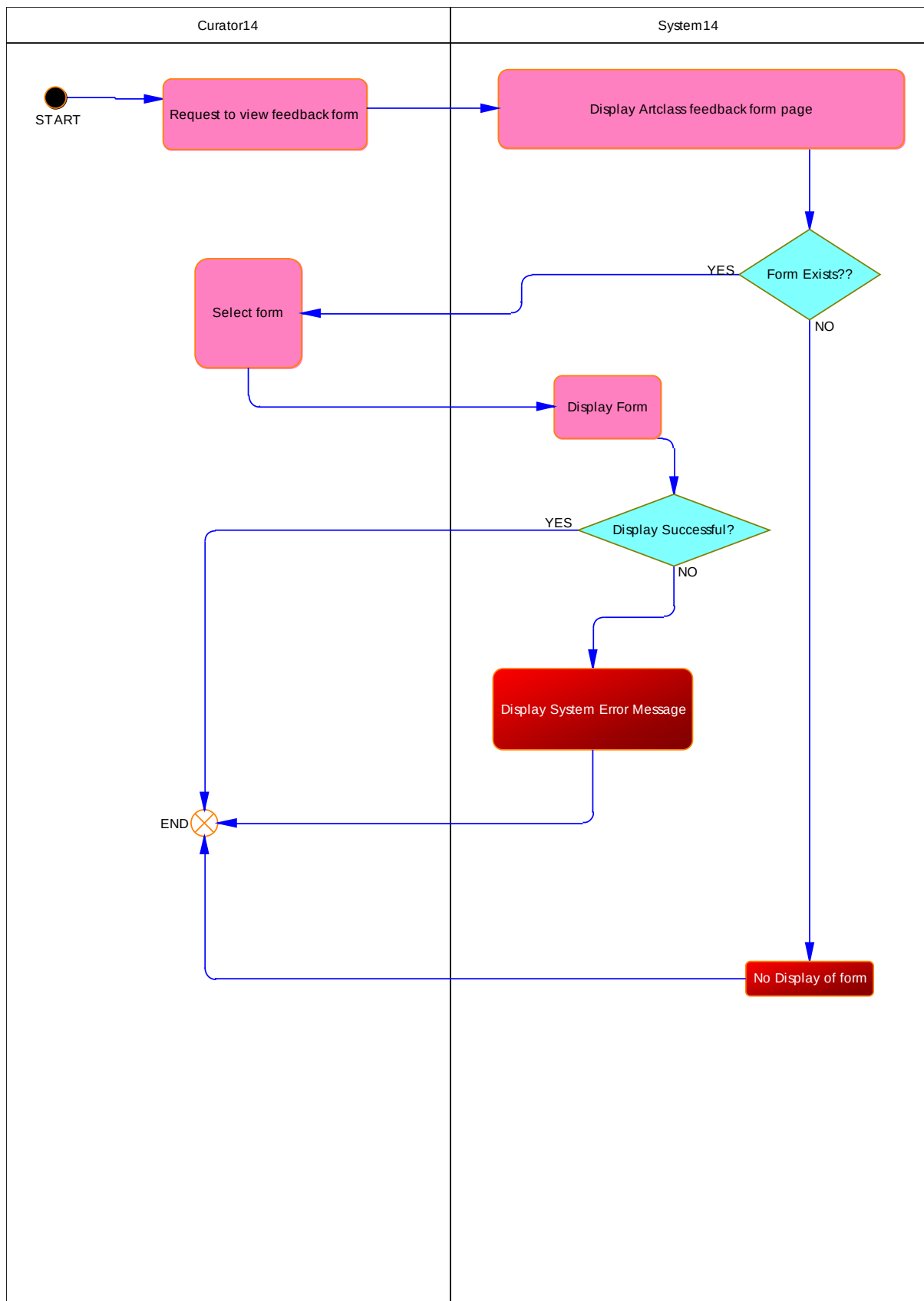




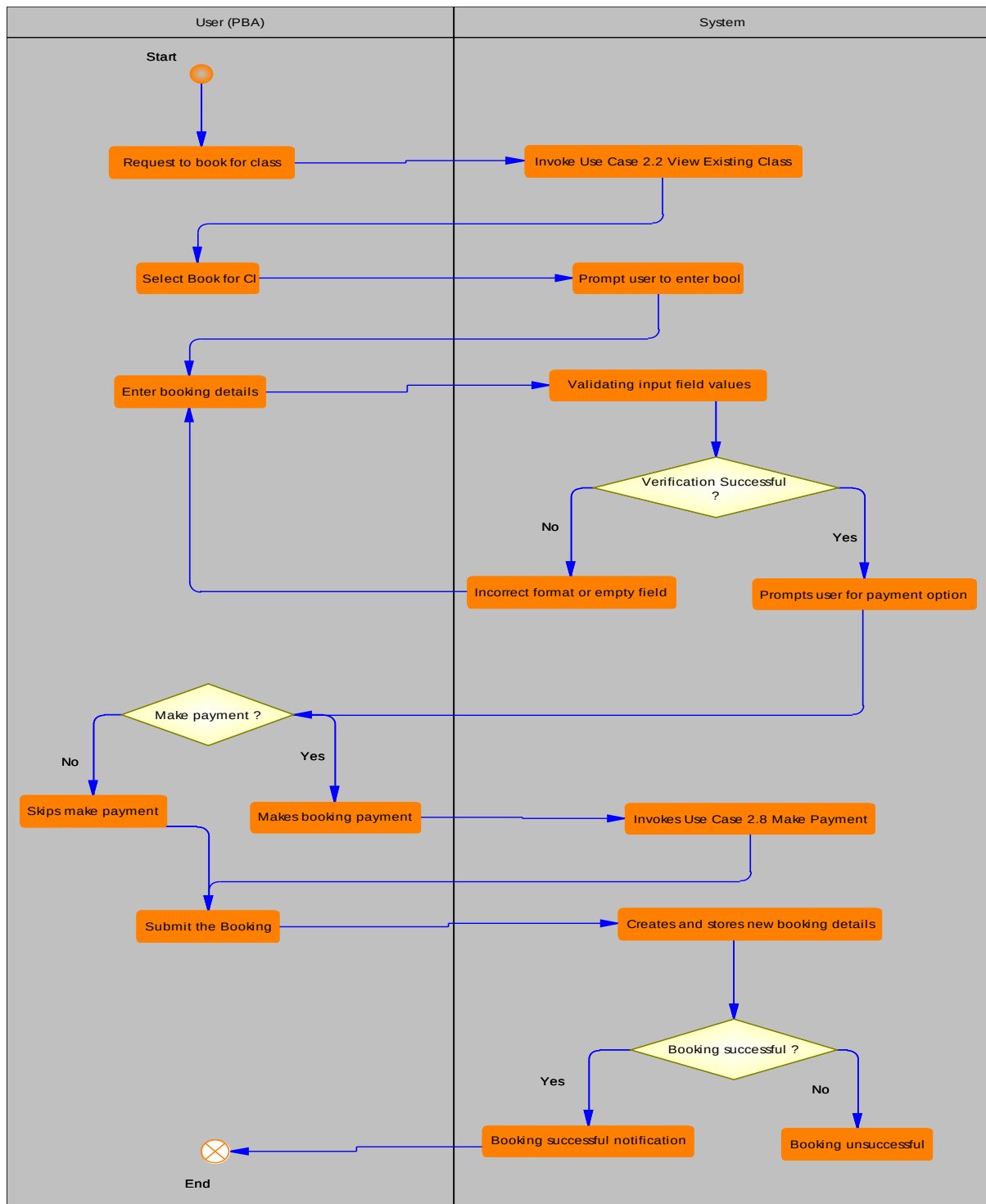


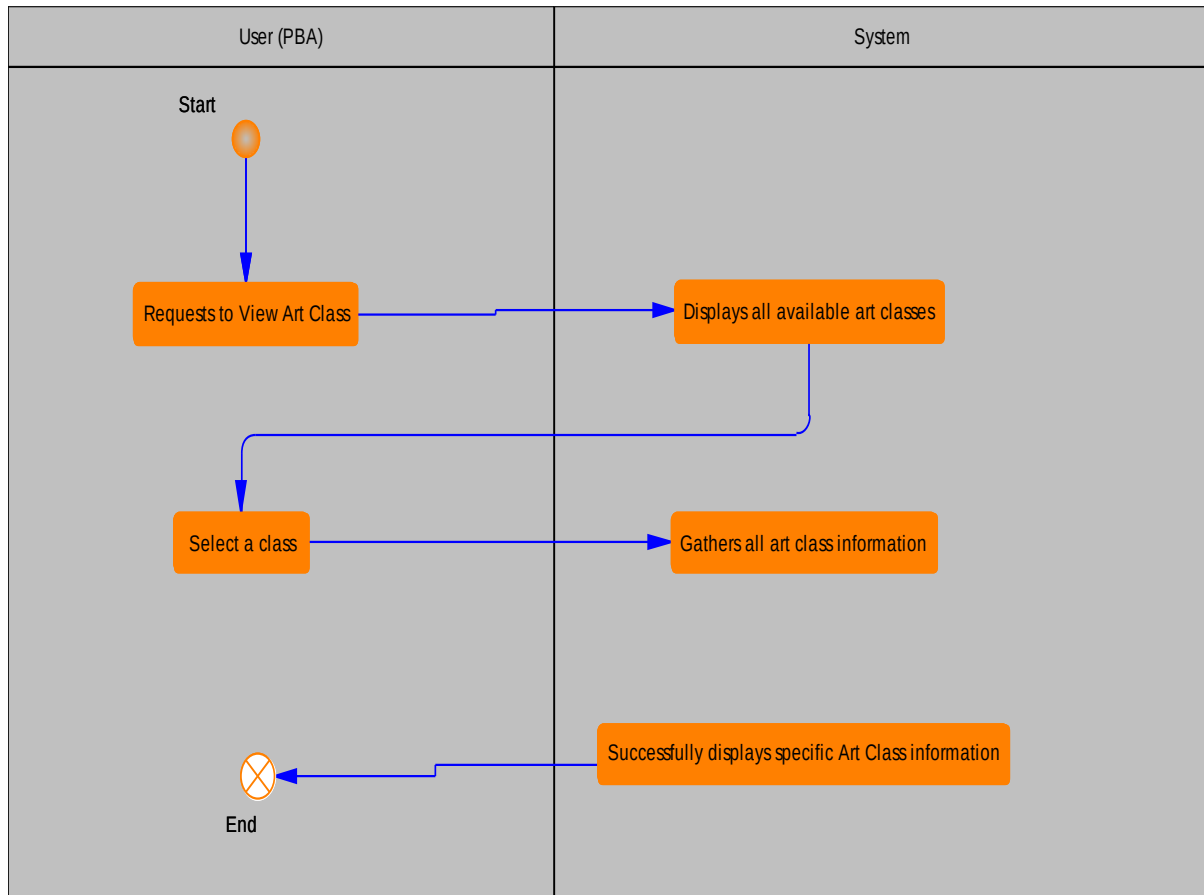


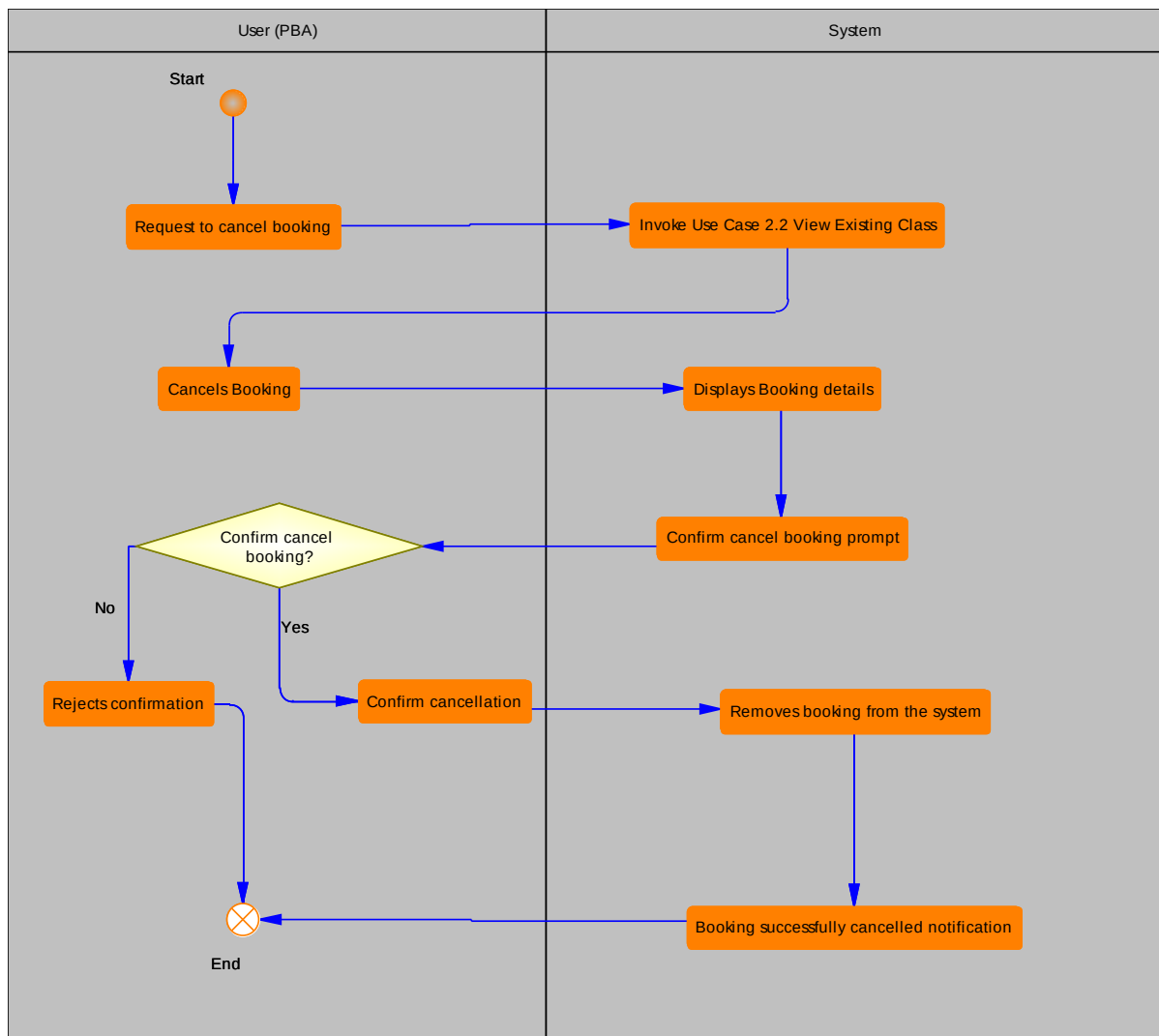


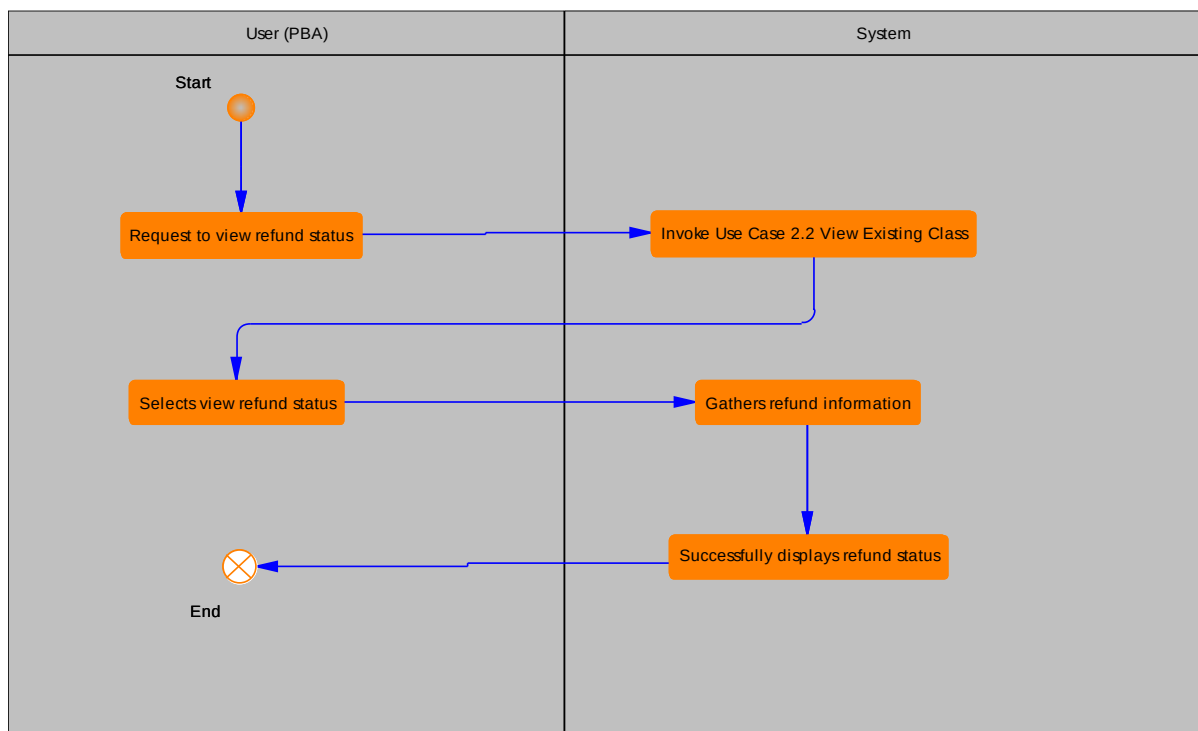
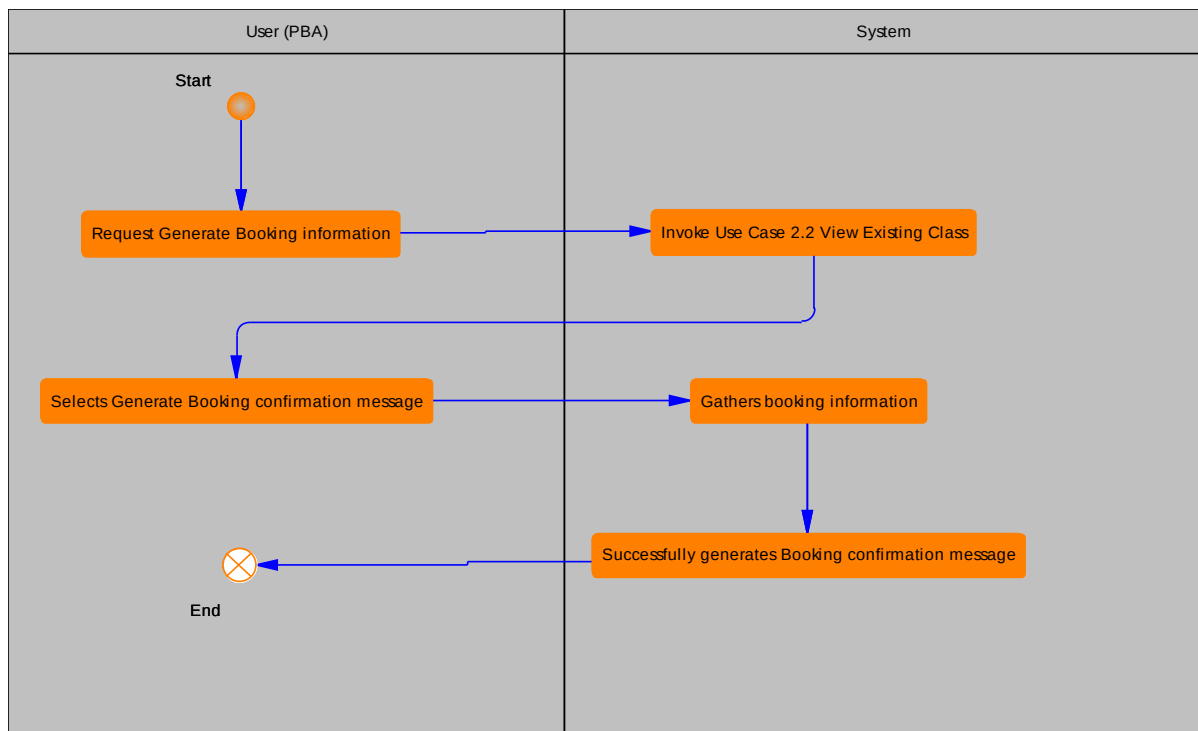


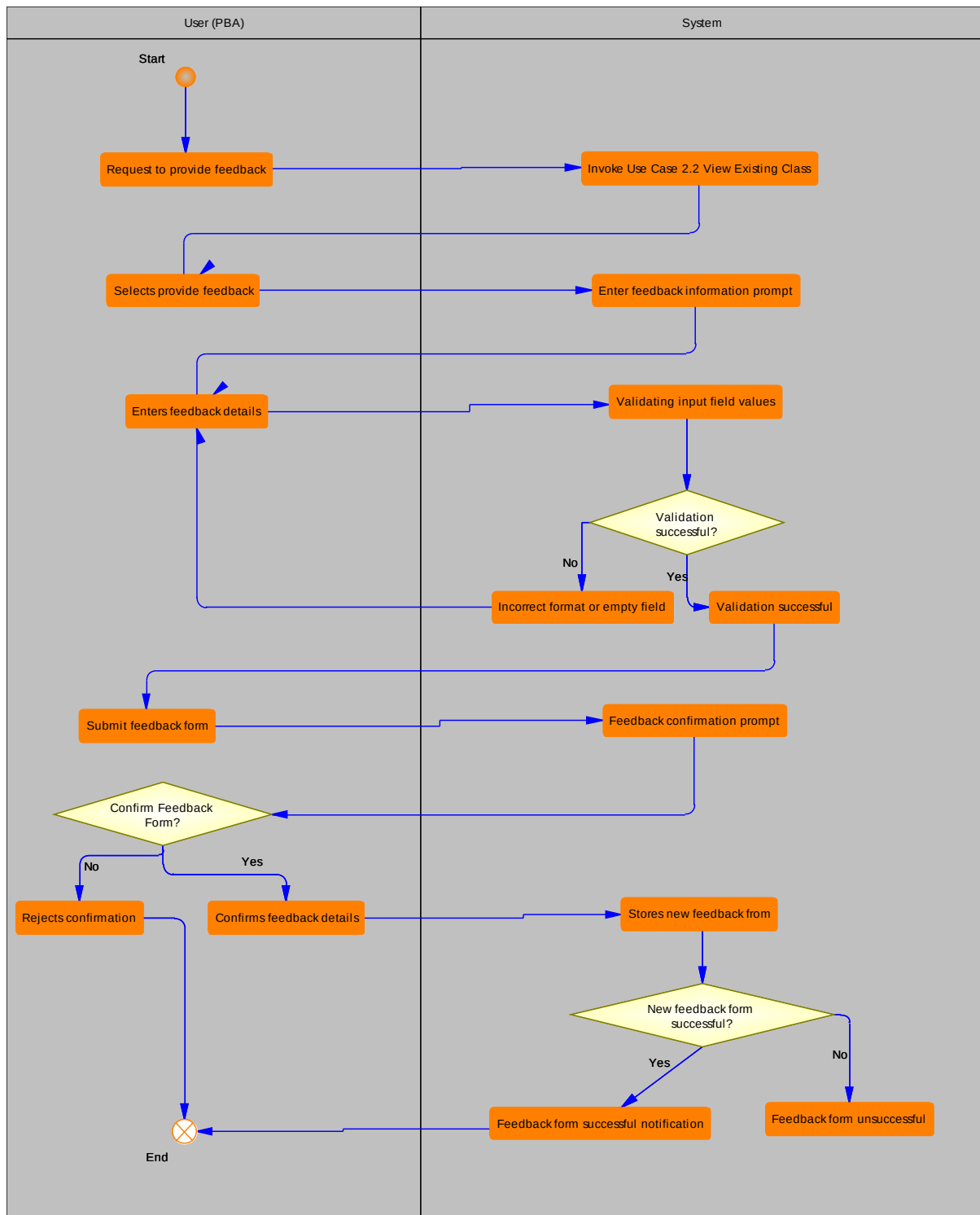
4.1.2 Art Class Subsystem

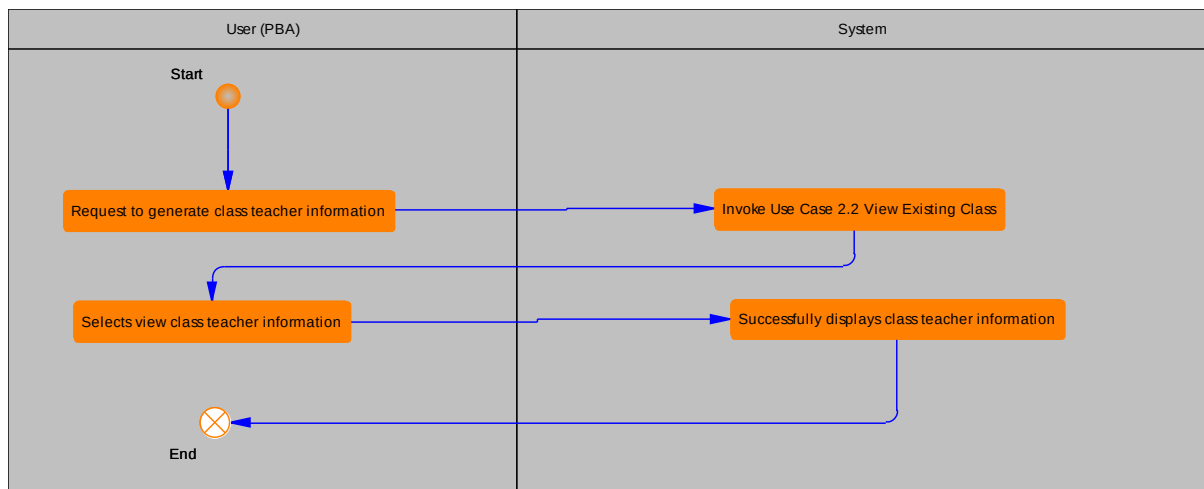


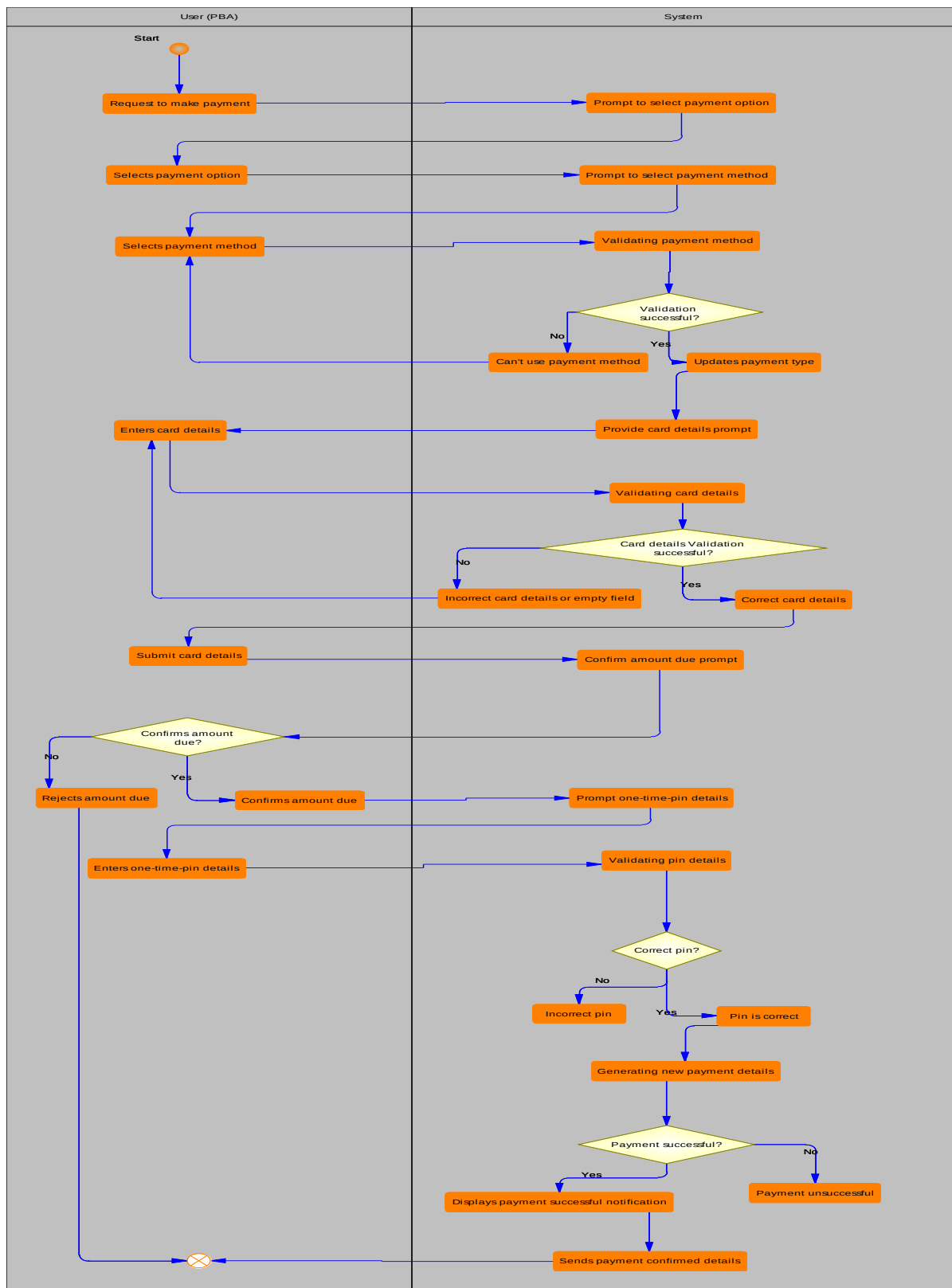


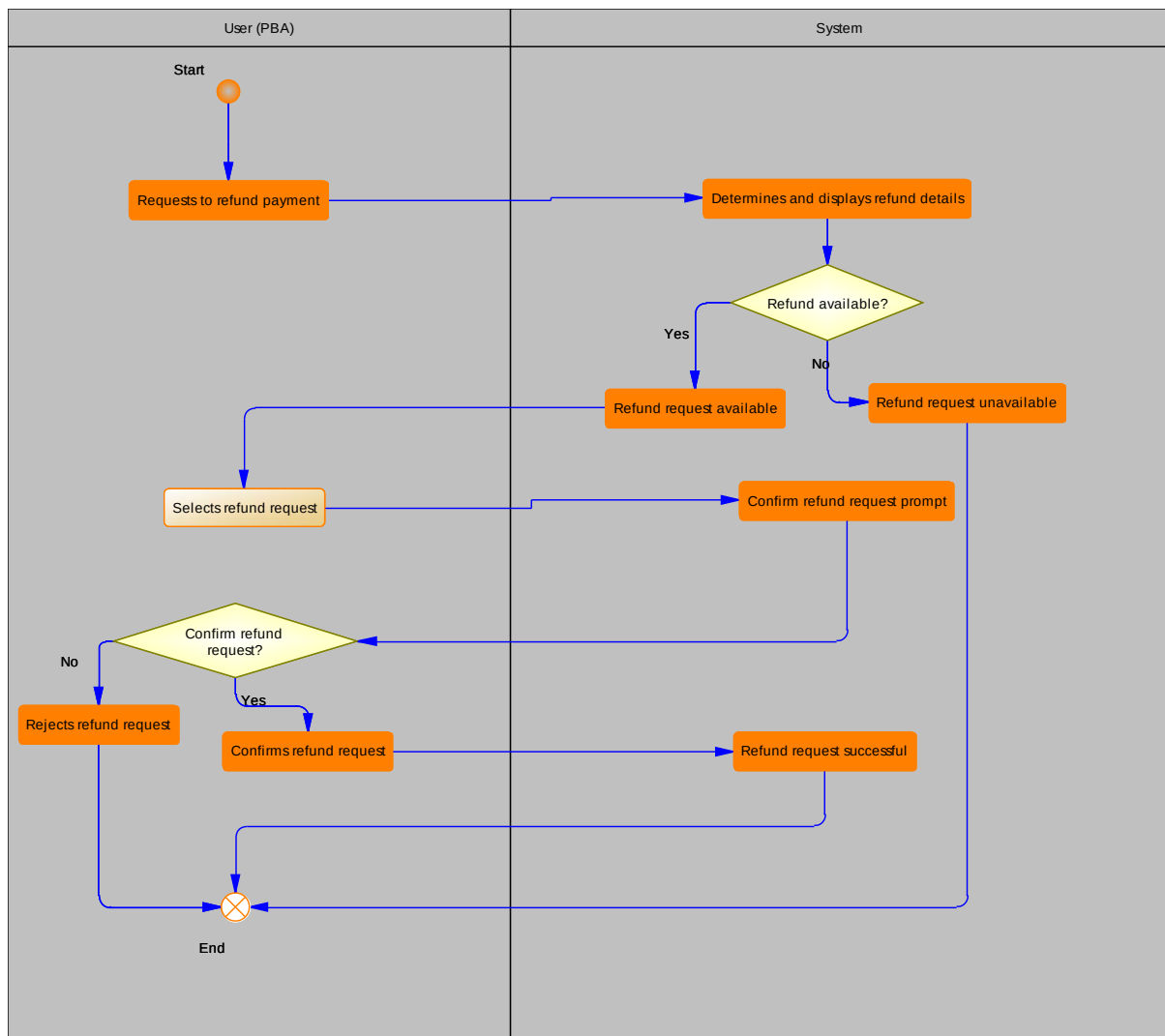




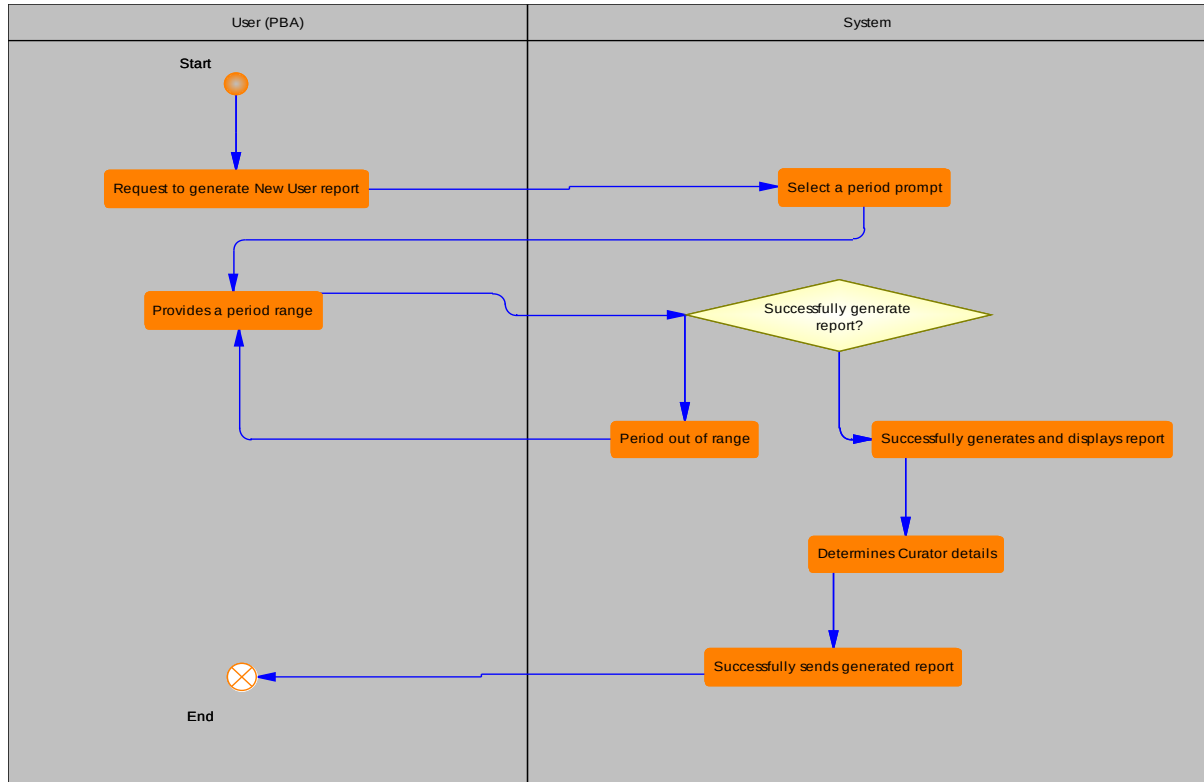


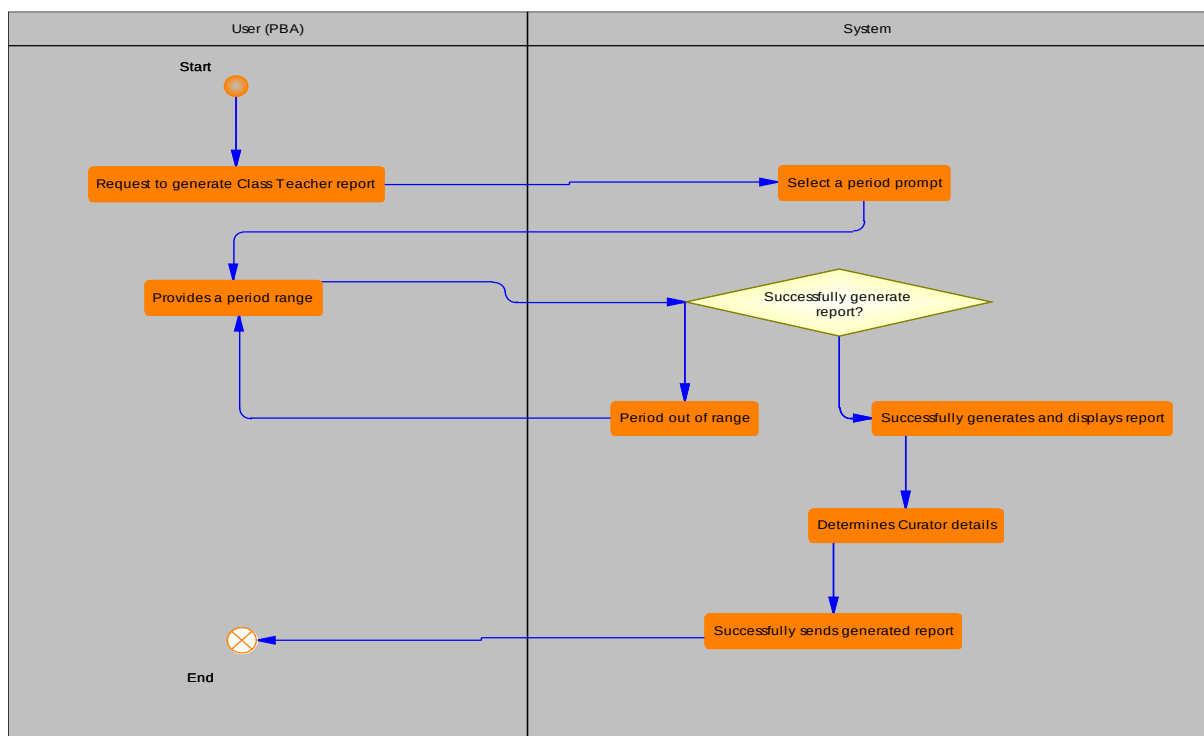
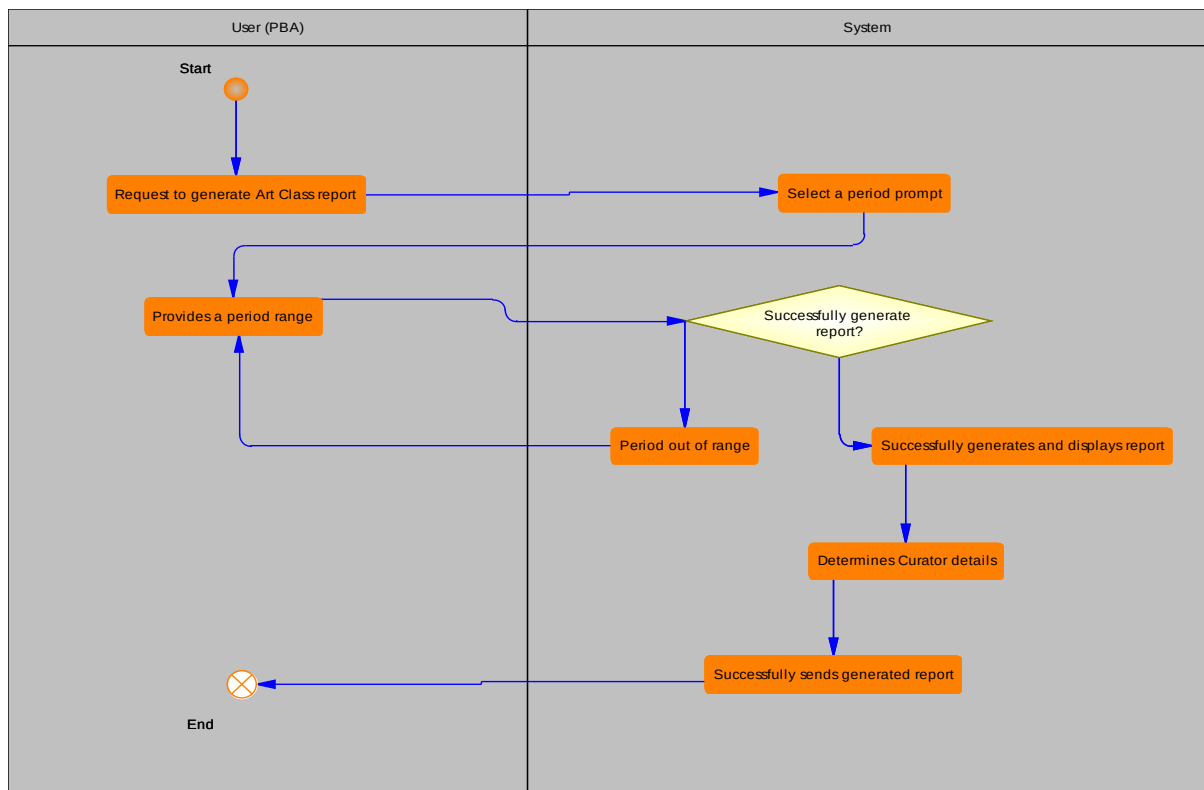


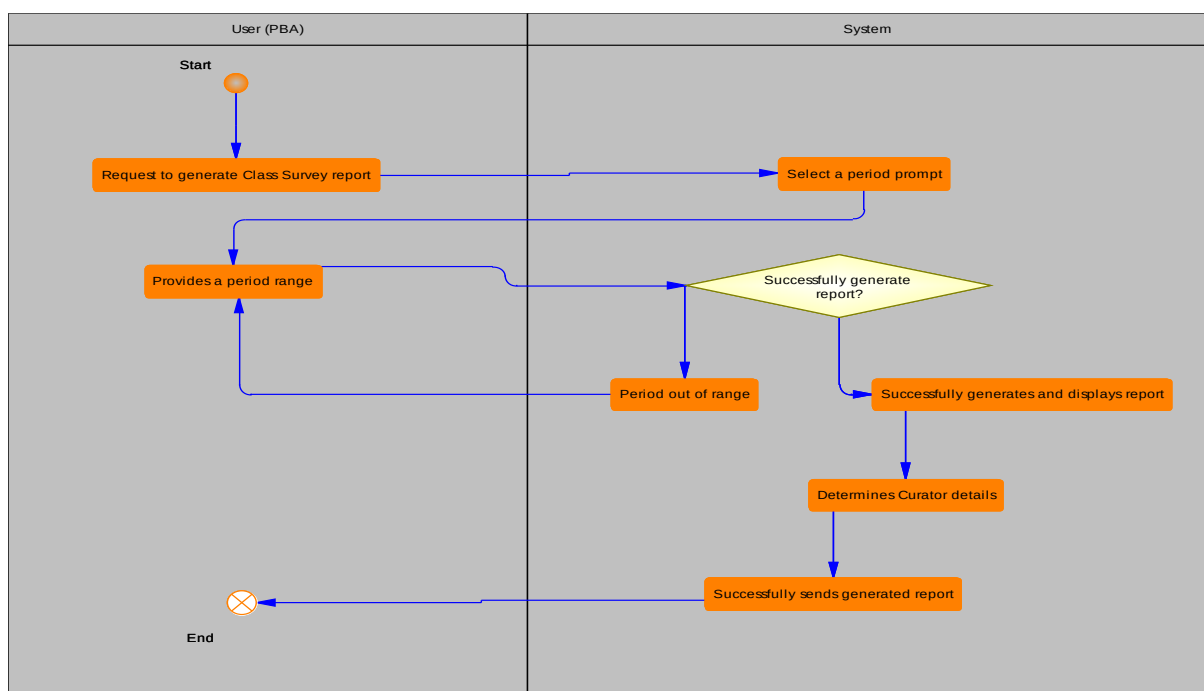
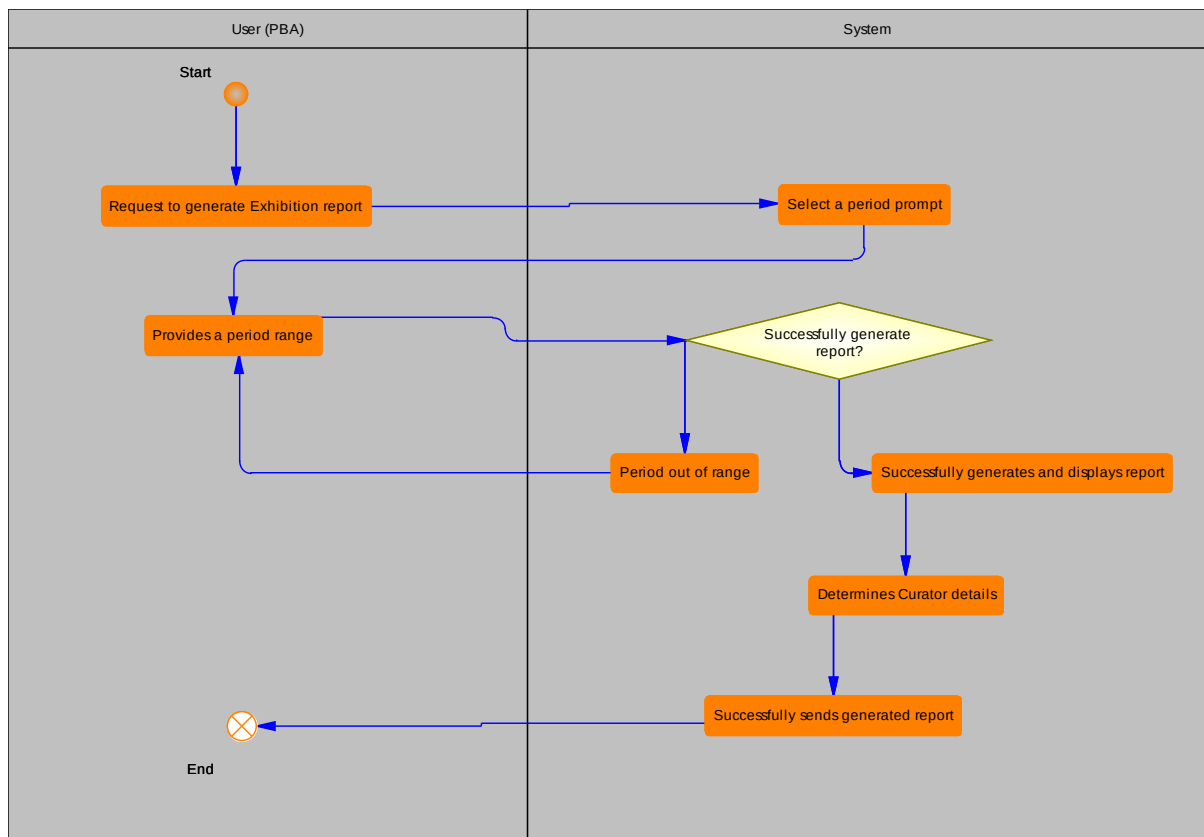


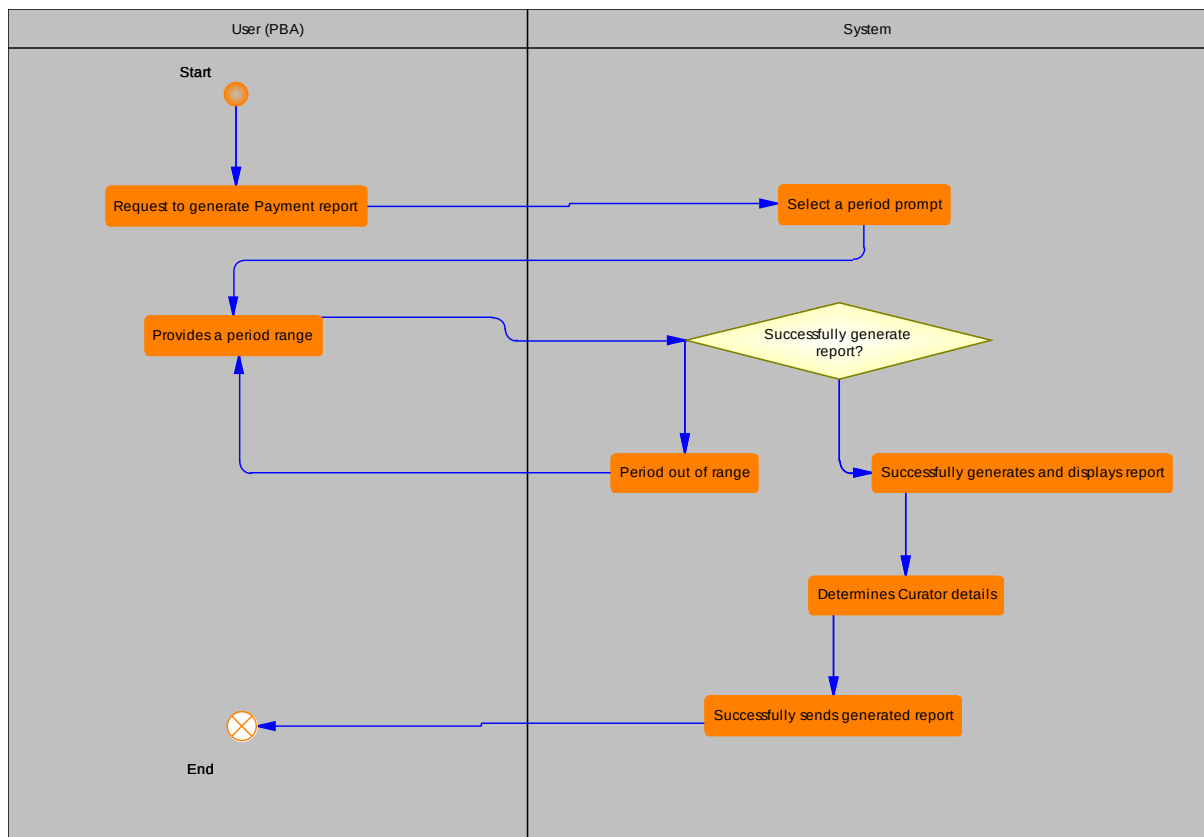


4.1.3 Reporting Subsystem

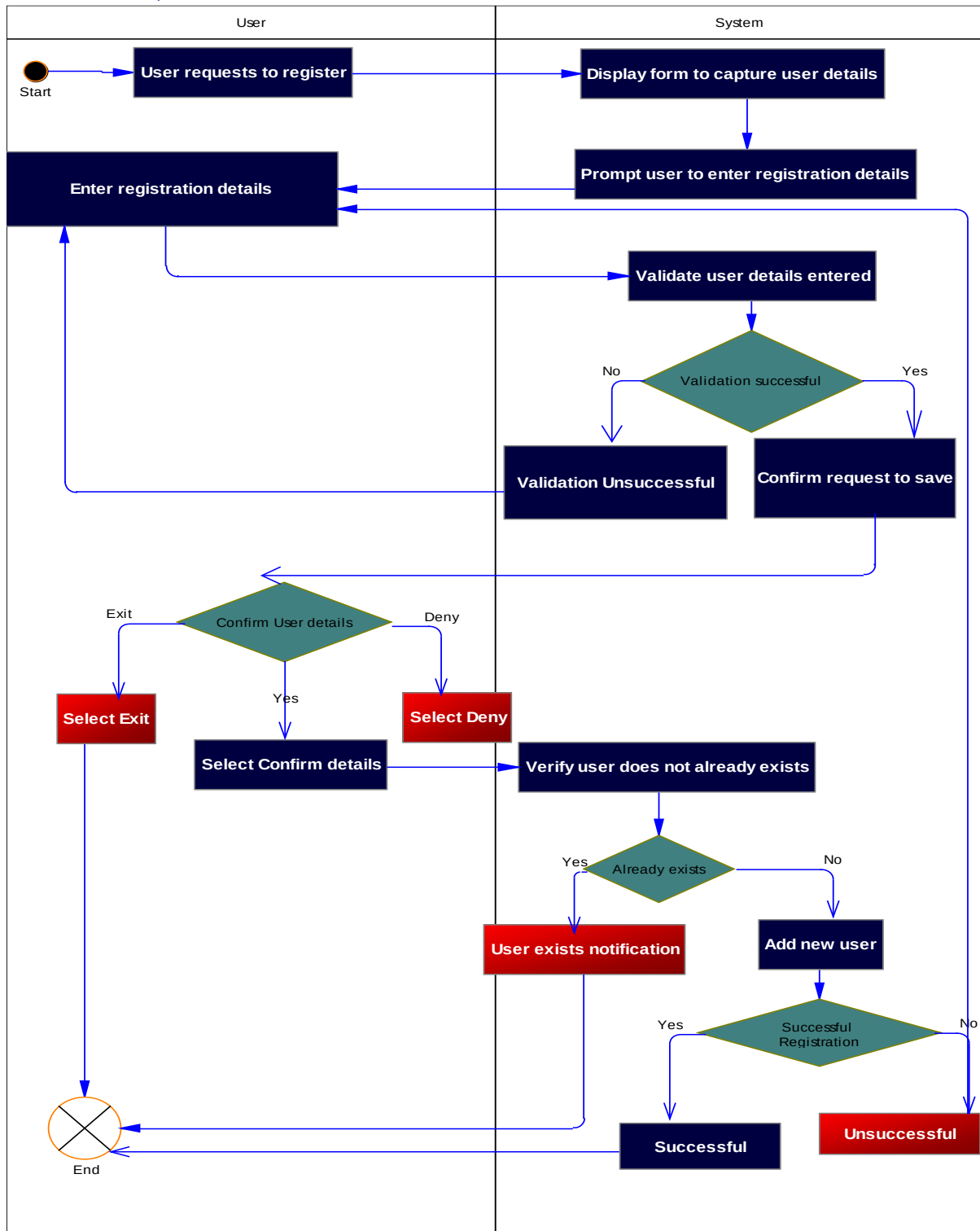


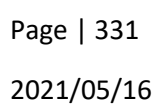


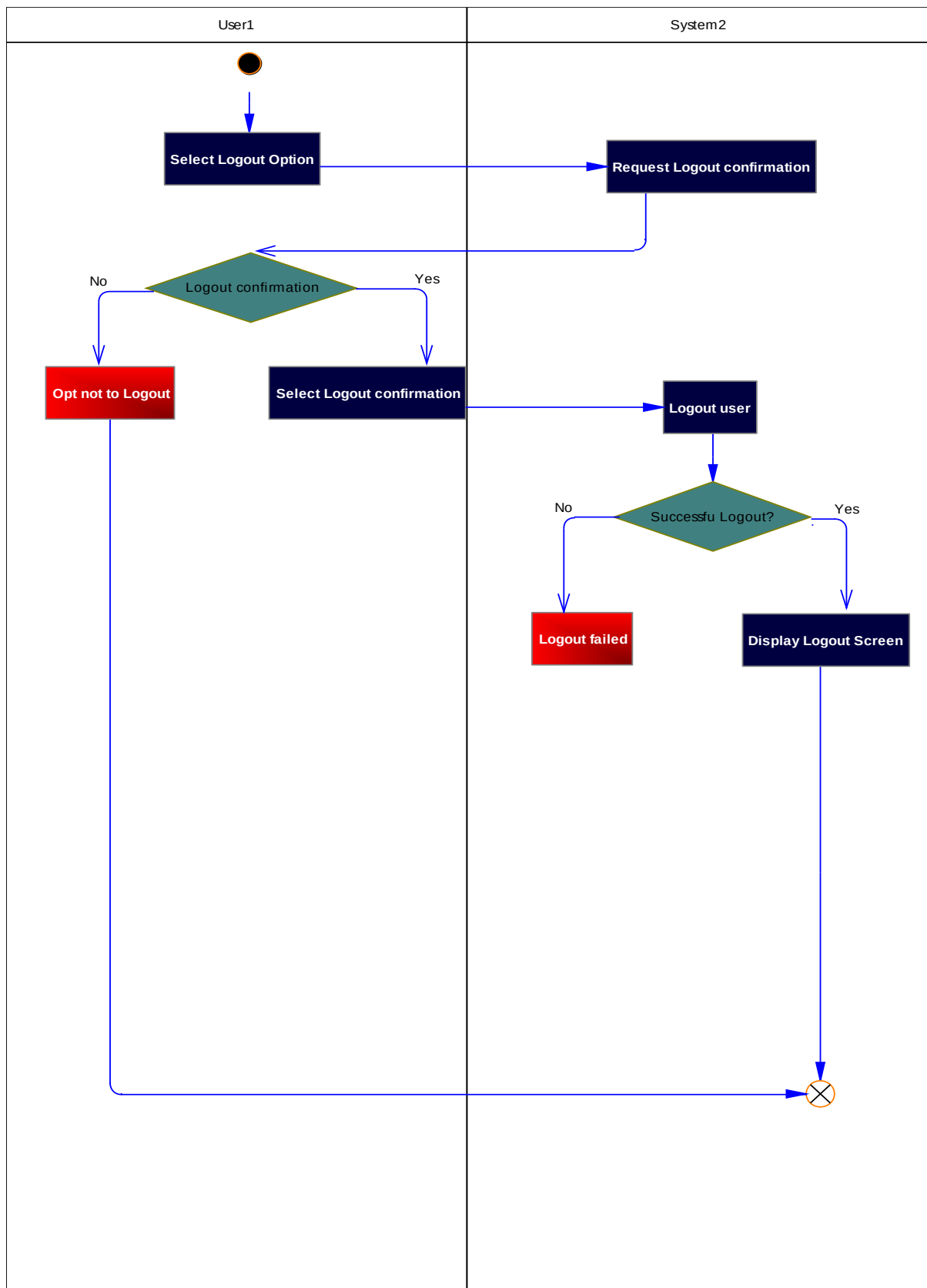


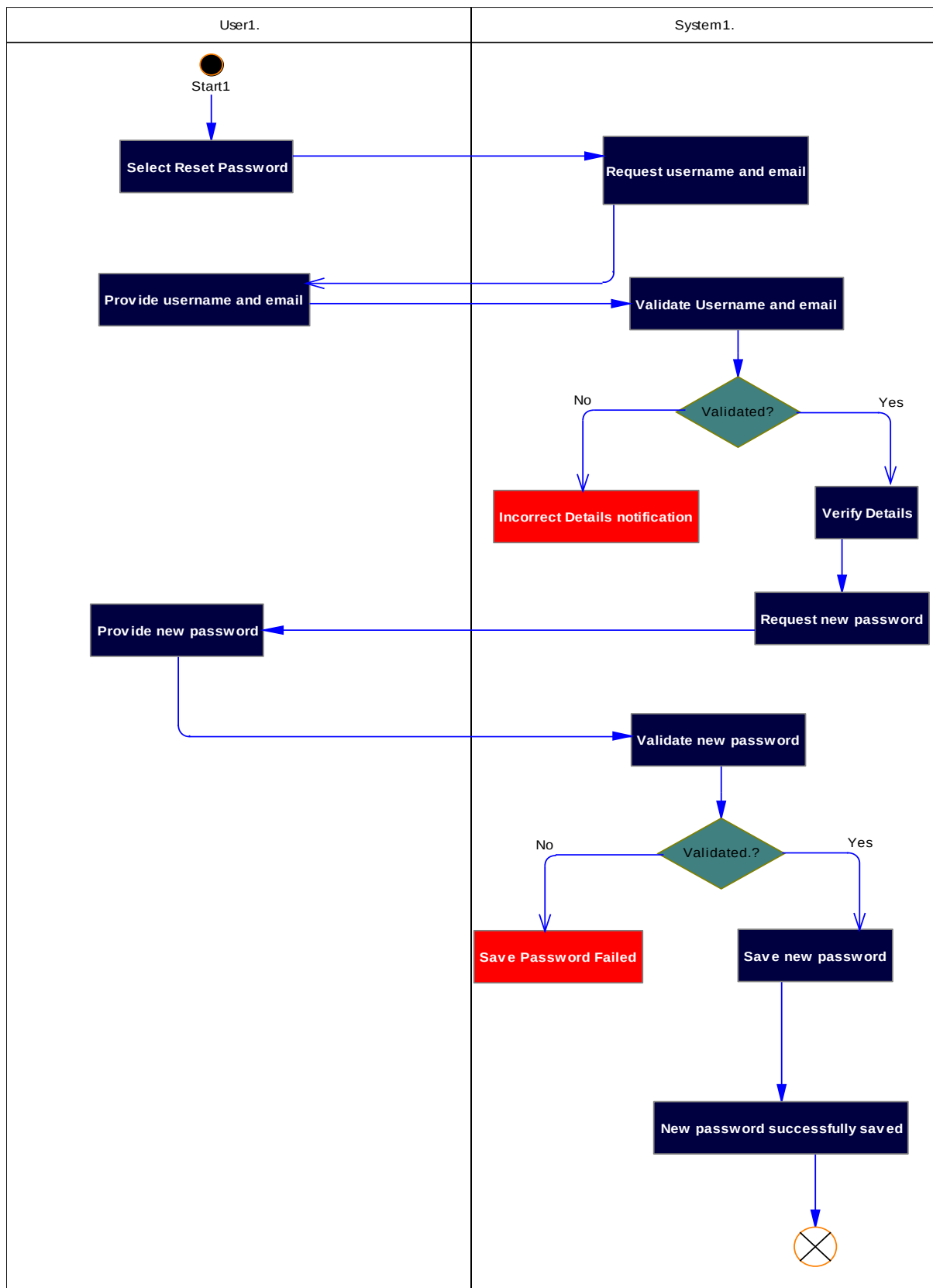


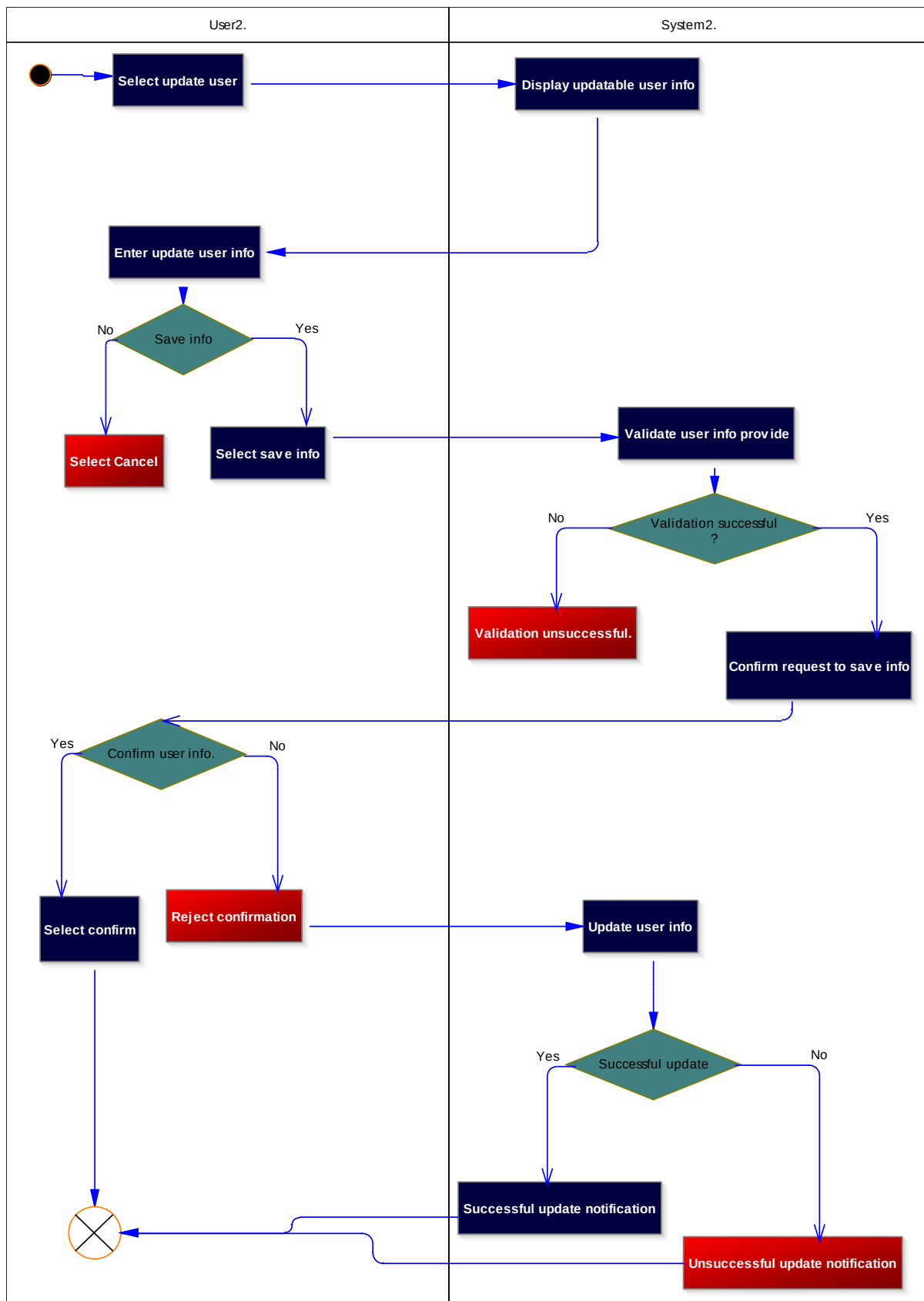
4.1.4 User Subsystem

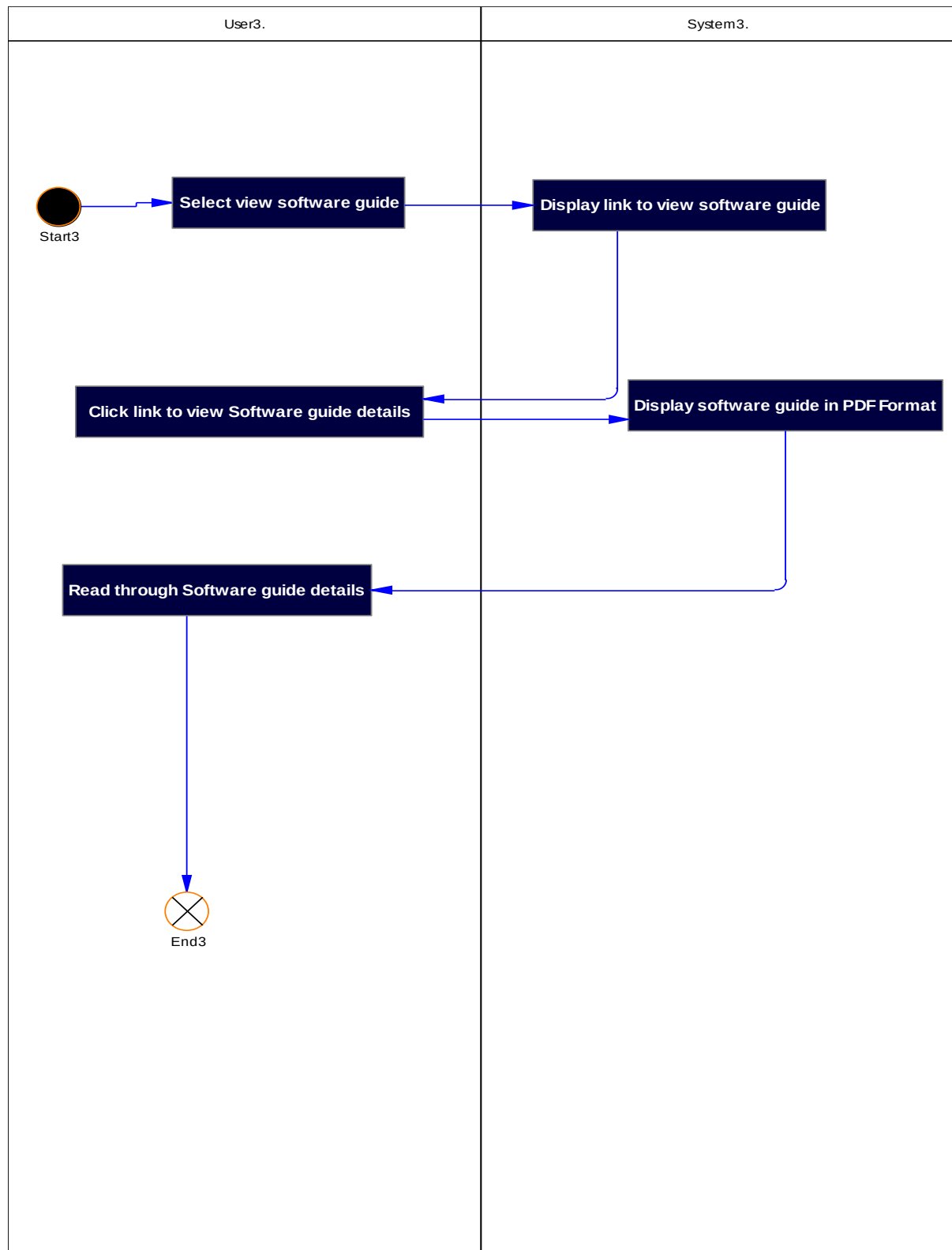


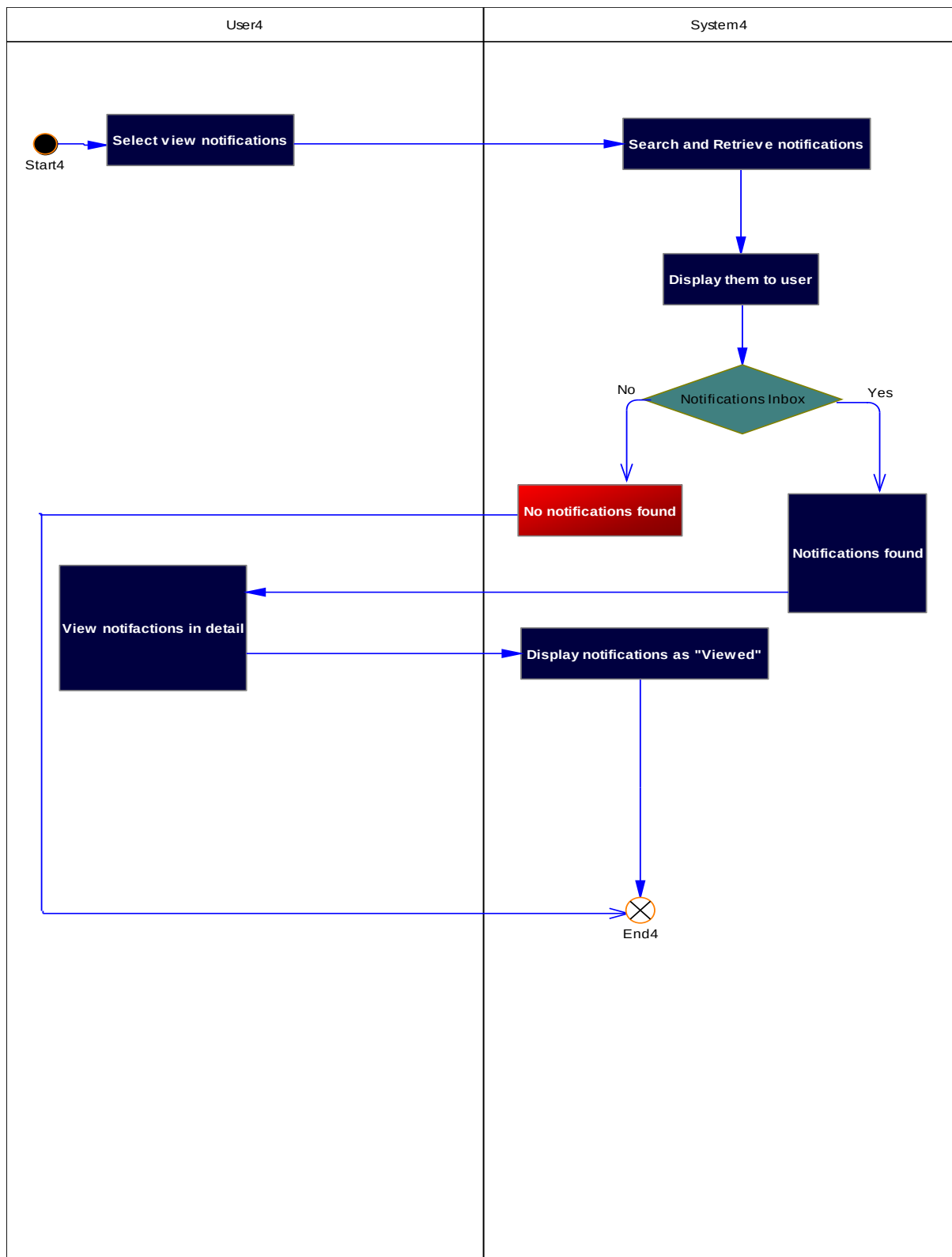


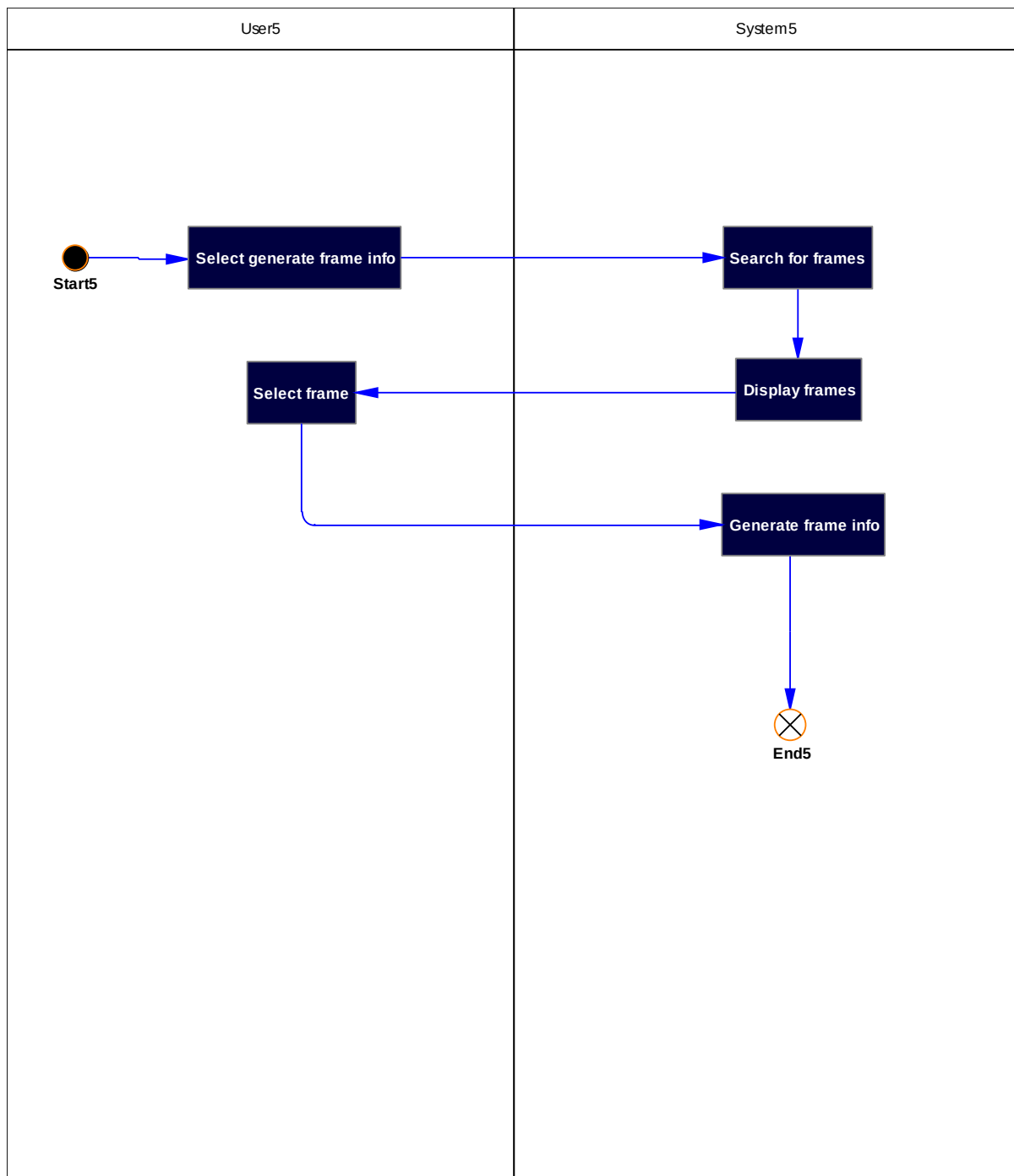




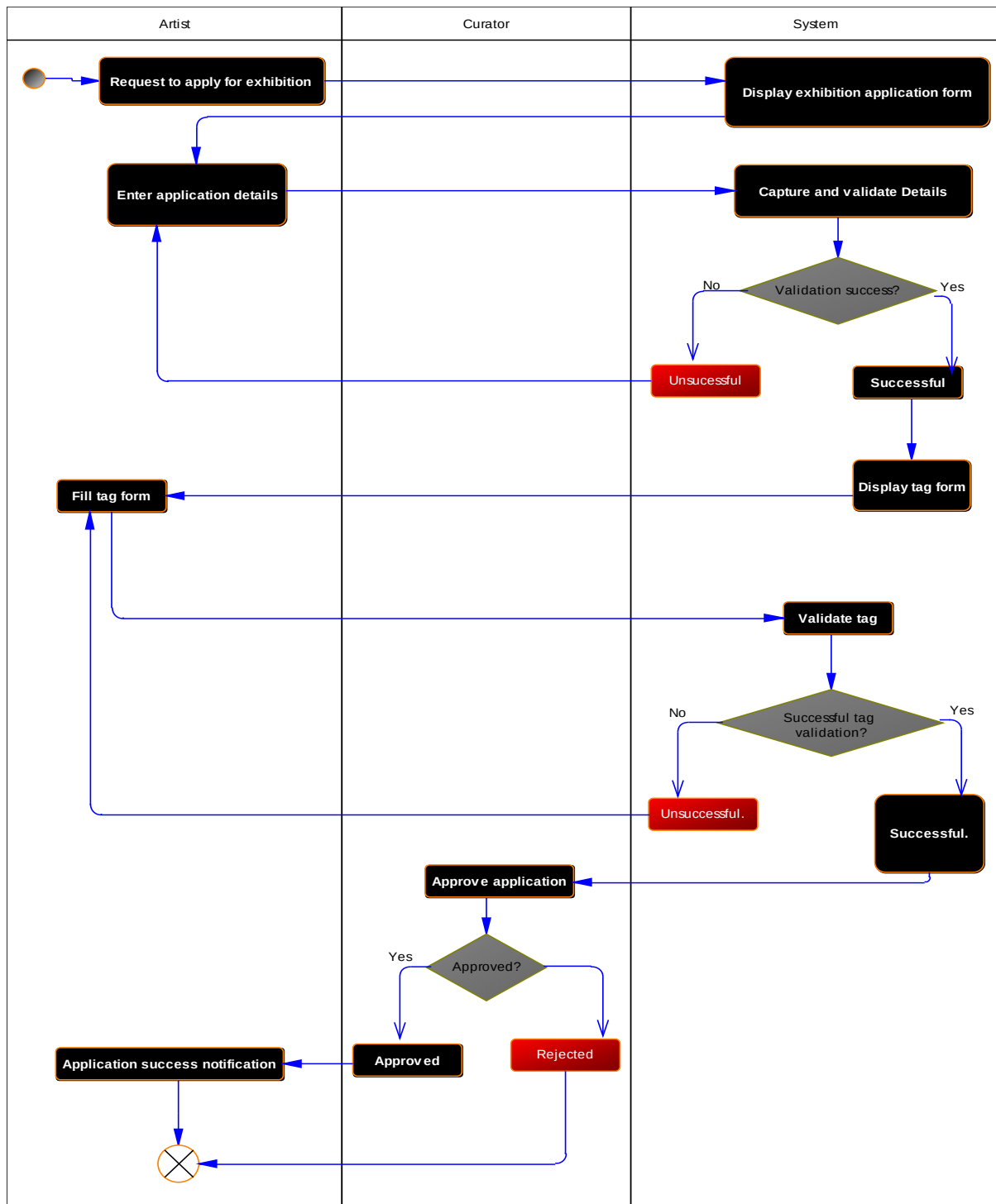


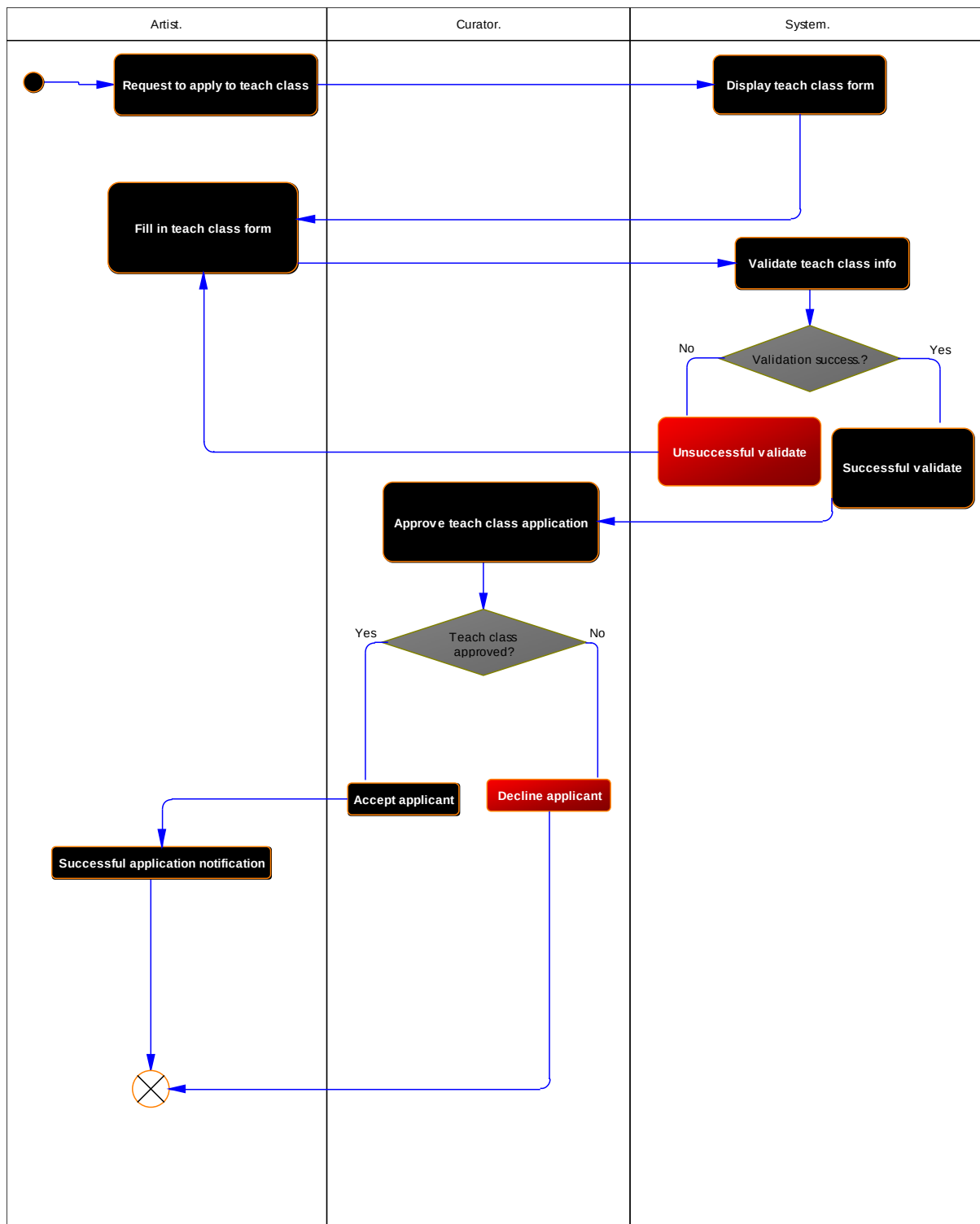


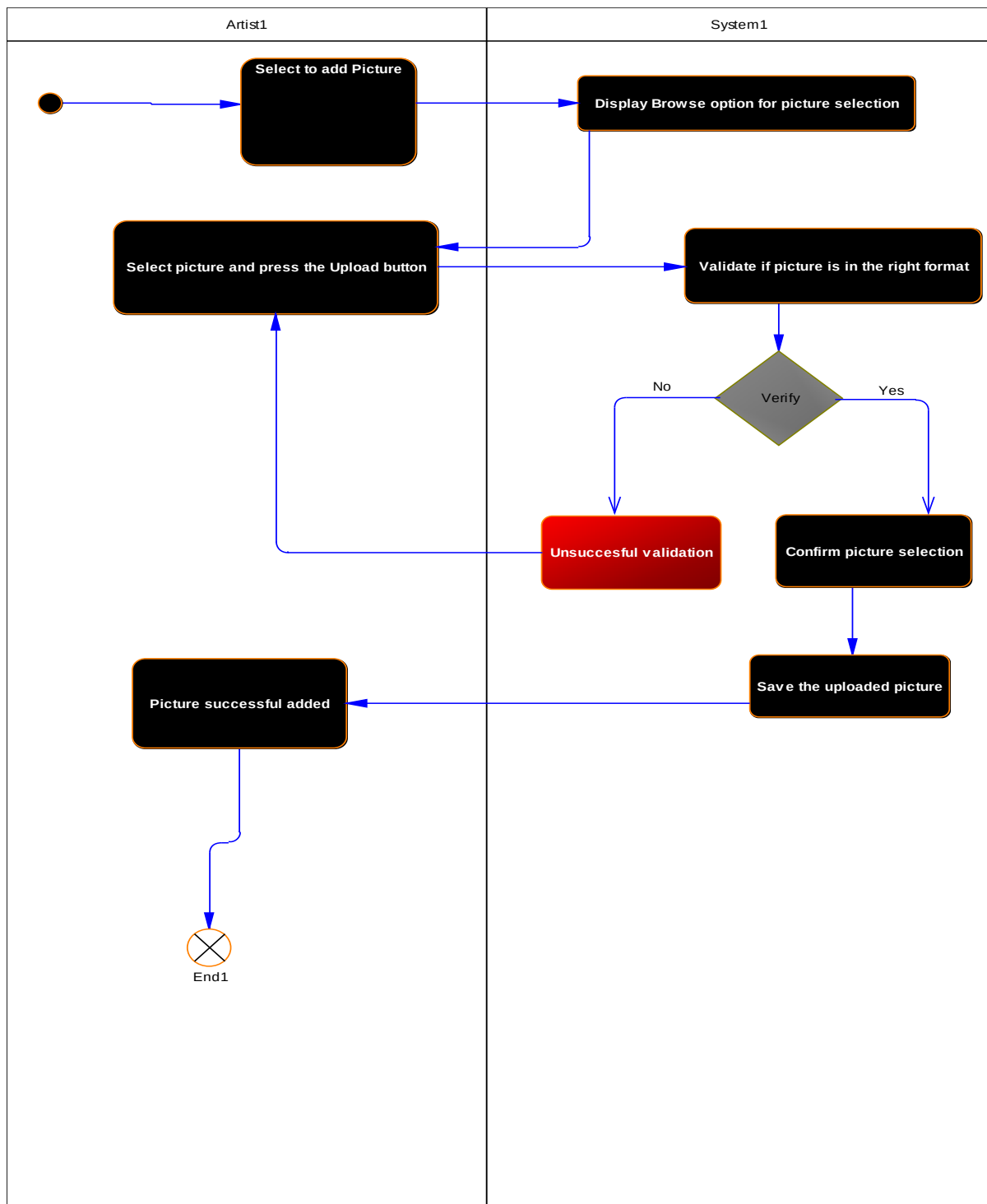


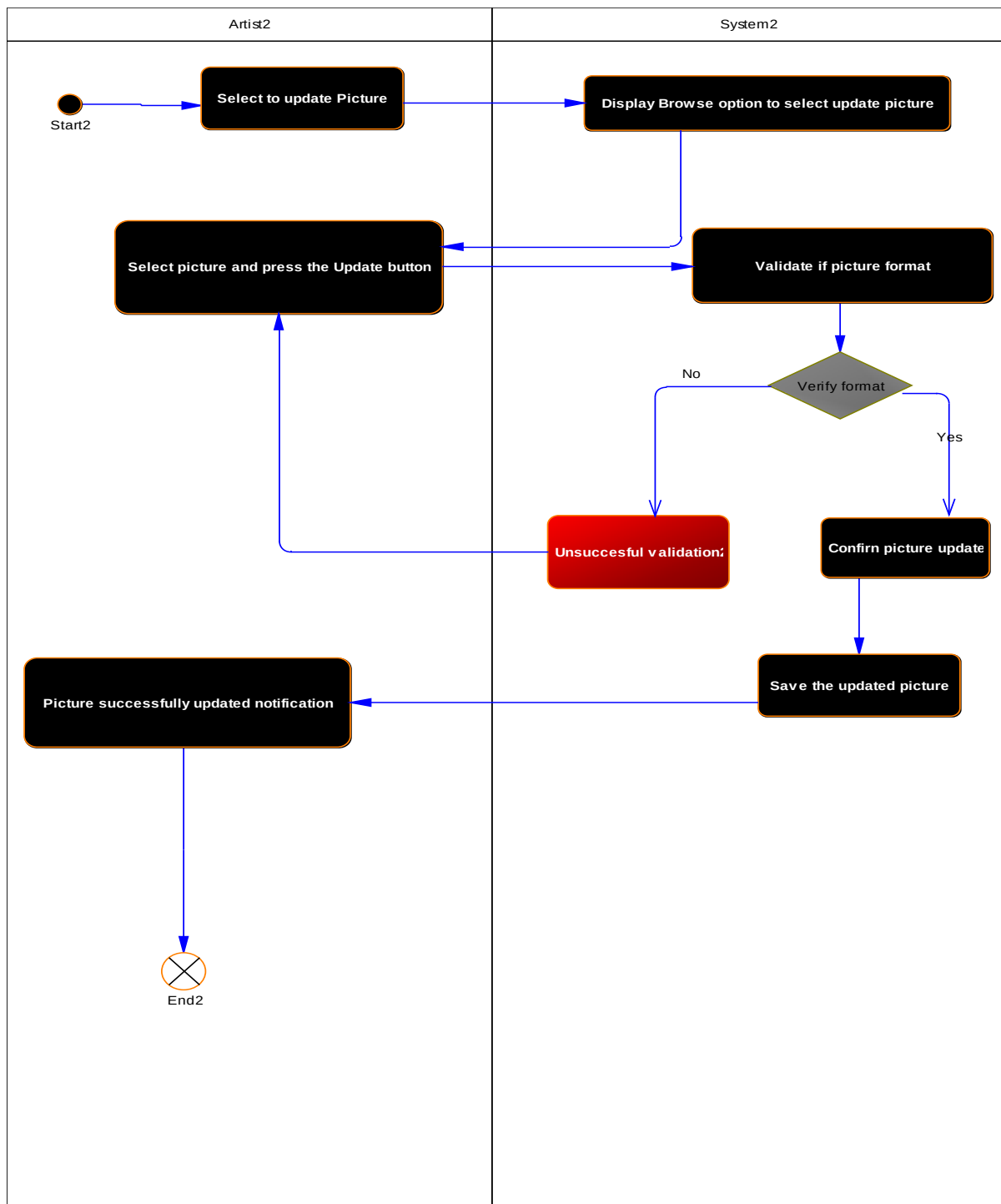


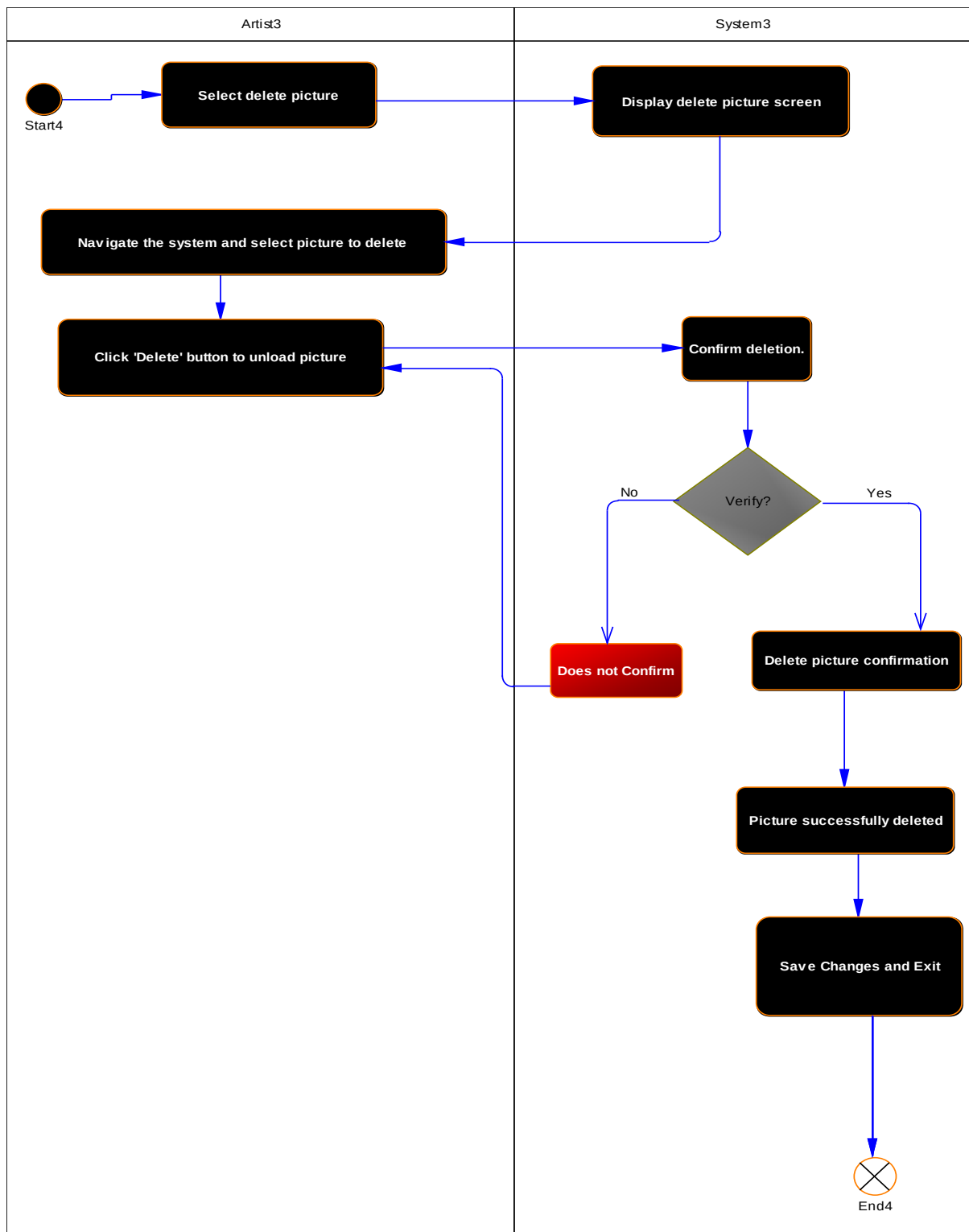
4.1.4 Artist Subsystem

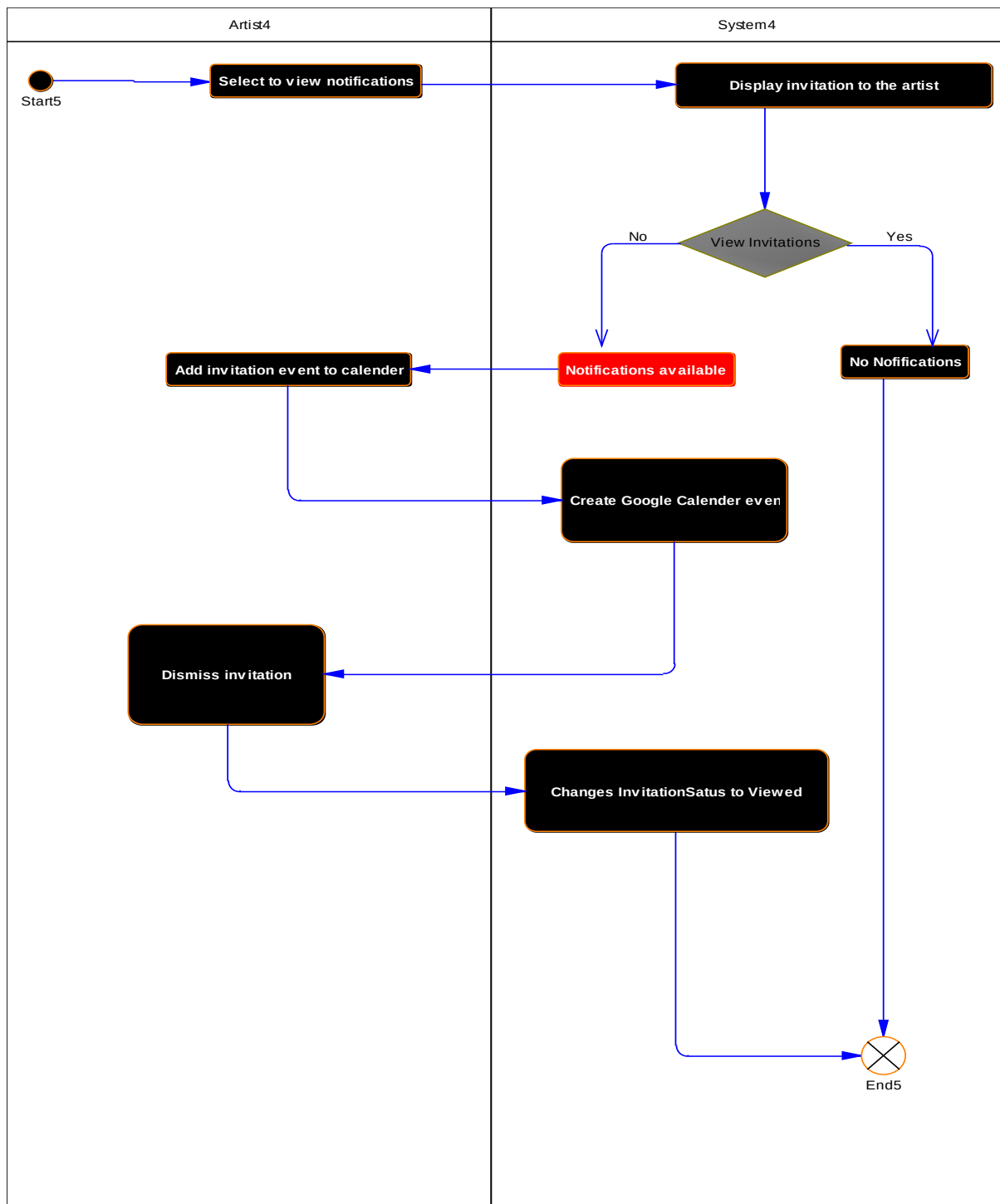


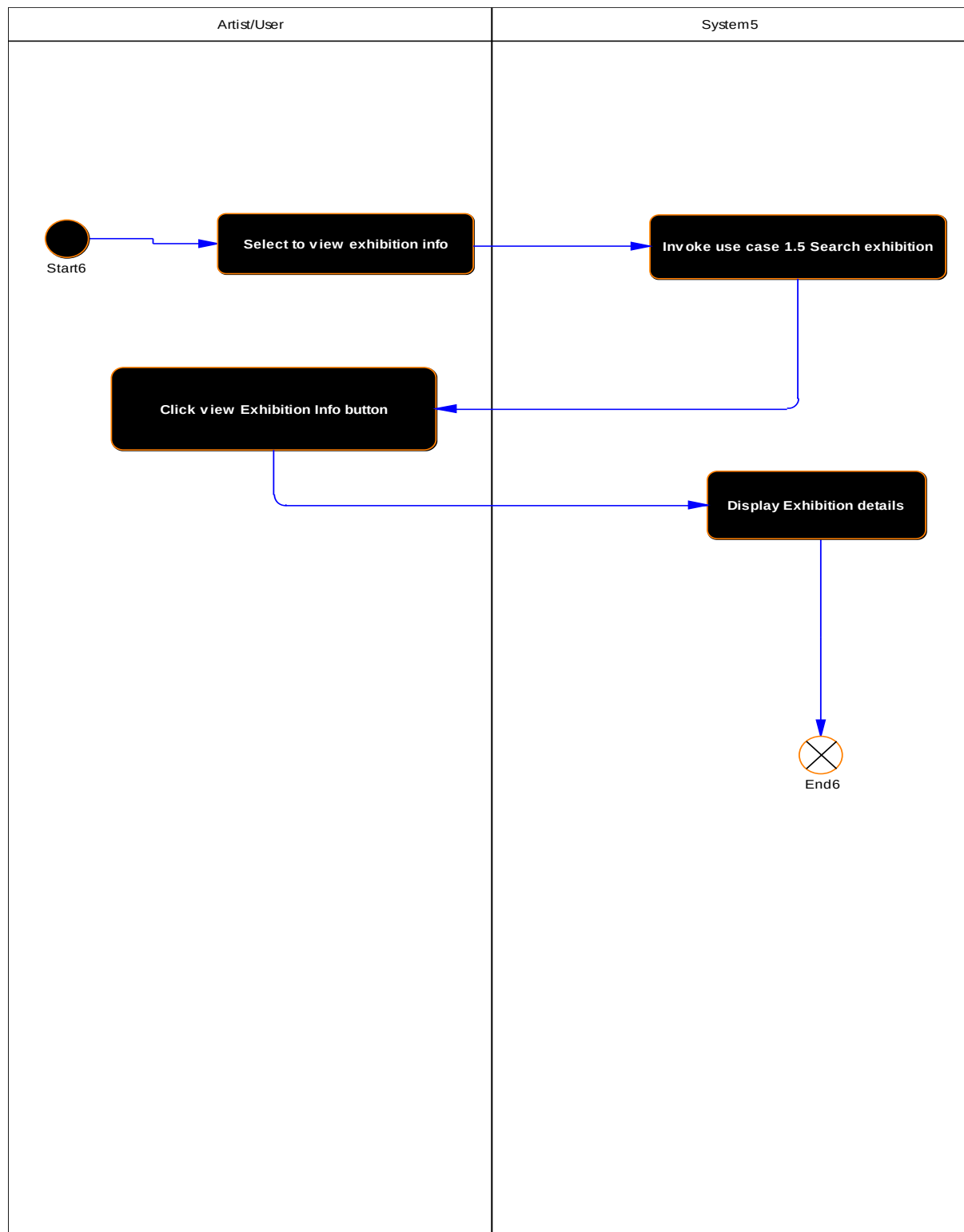


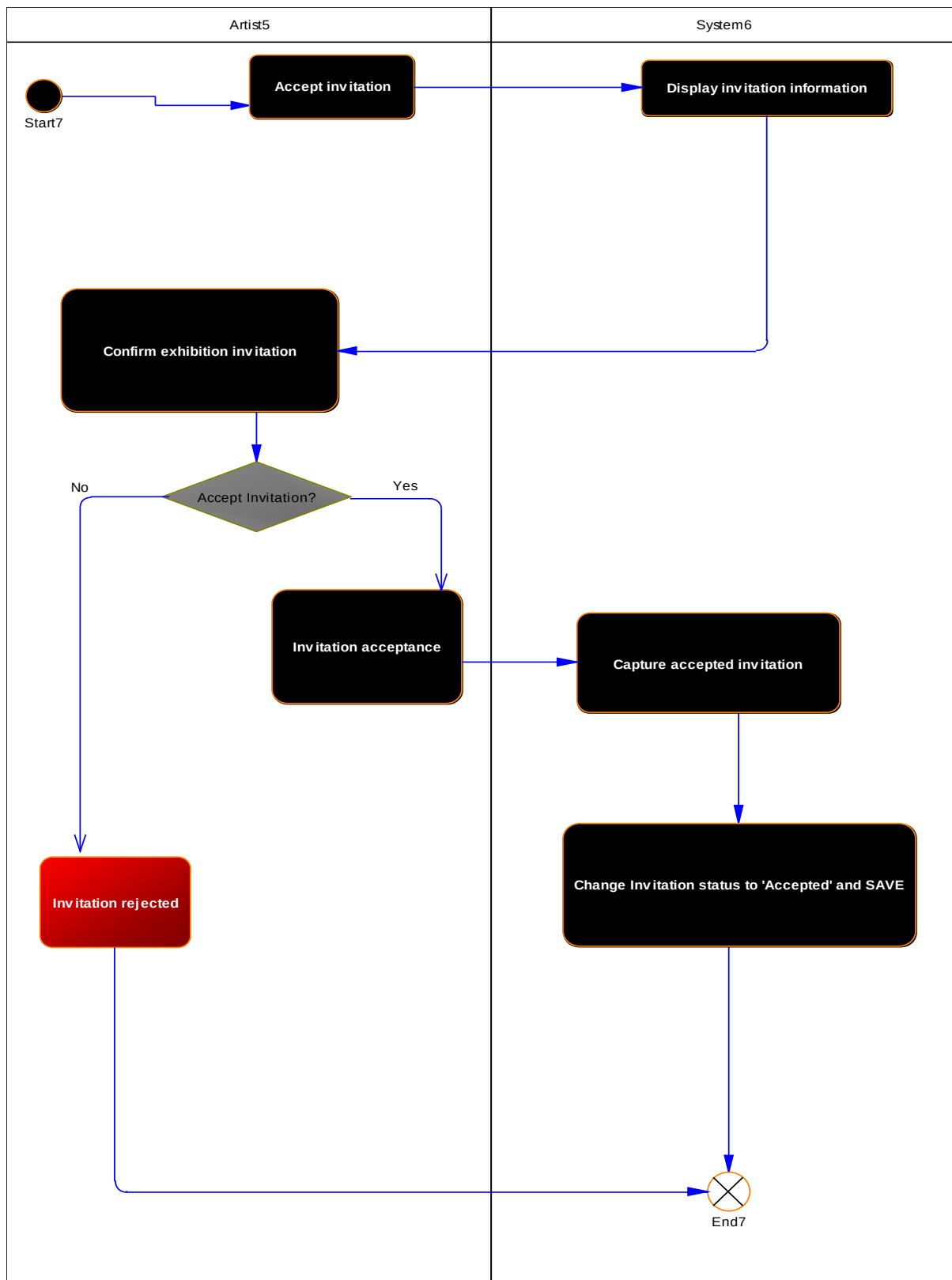












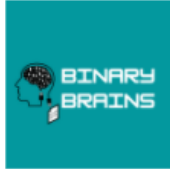


GROUP 23 DELIVERABLE 02



Conclusion

In conclusion, this section outlined the activity diagrams of the system to illustrate the flows and activities in the proposed system. Each task had its own processes carried out.

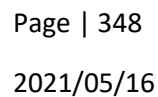


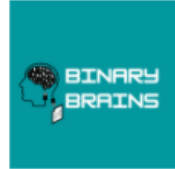
5. Data Model

Introduction

This section looks at the system through the use of data modelling. This contains the Entity Relationship Diagram which is used to show where all the data flows to and from and saves it in the system. The ERD contains different objects, places, people, and concepts in the system.

5.1 Logical Data Model (ERD)



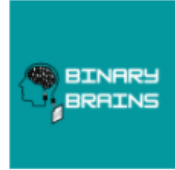


GROUP 23 DELIVERABLE 02



Conclusion

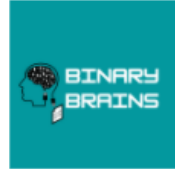
This brings us to a conclusion regarding our analysis of the data model where a determination was made regarding where all the system data is saved and where it flows to and from within the system.



6. Interfaces and Other Inputs

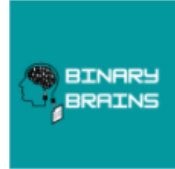
Introduction

This section contains the in-depth descriptions and analysis of the different inputs and interfaces in the proposed system depicted in the use case diagrams and data flow diagrams.



6.1 Input Interface Description

Requirement Number:	Requirement Title:	Description:	Purpose:	Usage:	Entities & Attributes:	Logical Layout:
1.1	Approve exhibition User	1.1 The Curator clicks to view the Applications for Exhibitions	This contains the information that is related to approving a user who applies for an exhibition	This is used when the curator views an exhibition application that has just been submitted to her	None	Clicked by the Curator
		1.1 The Curator clicks on application to view the details.	This contains all the details related to the applications for exhibition that the Curator wants to view	This is used when the curator views exhibition application details	None	Viewed by the curator
		1.1 The Curator reads the application and clicks on Approve Application	This used when the curator goes through an application and reads it through and approve the application for the artist to exhibit.	This is used when the curator approves exhibitions applications after reading them.	None	Read and approved by the curator



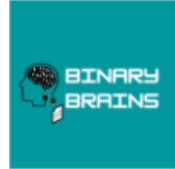
GROUP 23 DELIVERABLE 02



		1.1 ALT Step 6: The Curator reads the application and clicks on Deny Application.	This is used when the curator still reads through the application bit then denies the application	This is used when the Curator denies exhibition applications after reading them.	None	Read and disapproved by the curator
Requirement Number:	Requirement Title:	Description:	Purpose:	Usage:	Entities & Attributes:	Logical Layout:
1.2	Search Exhibition User	1.2 The Curator clicks on the Search Exhibition User button	This is used to gather information about the exhibition user in question	This is used when the curator wishes to search for an exhibition user	None	Selected by the Curator

		1.2 Search Exhibition user info inserted	Contains the keyword or word to search	This is when a user is requested to insert their search information	Exhibition Artists table: - ArtistID - ExhibitionID	Provided by the User
		1.2 The Curator receives the data on the screen in a table format	This used when the searched data is displayed to the curator	This is used when the search results are displayed to the curator	None	Data is displayed to the Curator

Requirement Number:	Requirement Title:	Description:	Purpose:	Usage:	Entities & Attributes:	Logical Layout:
1.3	Delete Exhibition User	1.3 Request to Delete Exhibition User	To delete the selected Exhibition user from the Exhibition list	This is used to remove an Exhibition user from the Exhibition table storage list.	NONE	Performed by the Curator
		1.3 Delete specific Exhibition user	Contains the action to delete an Exhibition user	This is used when the Curator would like to view the specific User details.	NONE	Provided by the curator
		1.3 Select search result found and click on the Delete Exhibition User	To delete the specific search result selected by the Curator.	To remove the selected search result from the Exhibition table.	Exhibition Artists table: - ArtistID - ExhibitionID	Selected by the Curator
		1.3 Confirm to delete Exhibition User	Contains Curator confirmation response	When information would like to be deleted but needs confirmation	None	Confirmed by the Curator

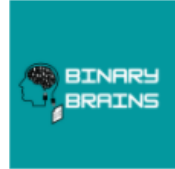


GROUP 23 DELIVERABLE 02



		1.3 Confirm exhibition to be deleted	Contains Exhibition User confirmation response	When information would like to be deleted but needs confirmation	None	Confirmed by the Exhibition User
		1.3 ALT Step 7: The Curator denies the confirmation and the Use Case ends	Contains Curator not being responsive to the confirmation	When information would like to be deleted but no confirmation has been provided.	None	No confirmation from the Curator

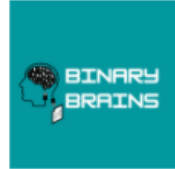
Requirement Number:	Requirement Title:	Description:	Purpose:	Usage:	Entities & Attributes:	Logical Layout:
1.4	Create New Exhibition	1.4 Request to create Exhibition	Contains the selected option to create a new skill level	When the curator/admin would like to add a new exhibition to the system		Provided by Curator/Admin
		1.4 Provide new exhibition details	Contains the exhibition details provided by the curator/admin	When the curator/admin provides new exhibition information	Exhibition table: - ExhibitionID - ExhibitionTypeID - ScheduleID - ExhibitionName - ExhibitionDescription - ExhibitionDate	Provided by Curator/Admin



GROUP 23 DELIVERABLE 02



					• OrganisationID • ExhibitionTime • ExhibitionAnnoucement • IDVenueID	
		1.4 Confirm new exhibition creation response by clicking OK	Contains action of successfully creating new exhibition	When the curator creates a new exhibition.	• NONE	Provided by the Curator



GROUP 23 DELIVERABLE 02



Requirement Number:	Requirement Title:	Description:	Purpose:	Usage:	Entities & Attributes:	Logical Layout:
1.5	Search Exhibition	1.5 Request to search for exhibition	Contains the selected option to search for an exhibition	When an admin/curator would like to search for a specific exhibition	None	Provided by the Curator
		1.5 Search exhibition details inserted	Contains the keyword or word to search	When an admin/curator is requested to insert search exhibition information	Exhibition table: - ExhibitionID - ExhibitionTypeID - ScheduleID - ExhibitionName - ExhibitionDescription - ExhibitionDate - OrganisationID - ExhibitionTime - ExhibitionAnnouncement - VenueID	Provide by the Curator
		1.5 Display search results on the screen for the Curator to view	Contains search results detailed on the screen to be viewed by the curator	When an admin/curator views the final search results on the screen	None	Provide by the Curator

Requirement Number:	Requirement Title:	Description:	Purpose:	Usage:	Entities & Attributes:	Logical Layout:
1.6	Update Exhibition	1.6 Request to update Exhibition	Contains the selected option to update a specific exhibition	When the admin /Curator requests to update an exhibition	None	Provided by the Curator/Admin
		1.6 Search exhibition details inserted	Contains the keyword or word to search	When an admin/curator is requested to insert search exhibition information	Exhibition table: •ExhibitionID •ExhibitionTypeID •ScheduleID •ExhibitionName •ExhibitionDescription •ExhibitionDate •OrganisationID •ExhibitionTime •ExhibitionAnnouncement •IDVenueID	Provided by the Curator/Admin
		1.6 Select specific exhibition details to update to update	Contains details of the exhibition class to be updated	When update details are requested for update	None	Provided by the Curator/Admin
		1.6 Confirm exhibition update	Contains a confirmation action to update exhibition	When an update needs to be confirmed first to	None	Confirmed by the curator

				proceed to update		
		1.6 ALT: Data not saved correctly. Display error	Contains error message for data not saved correctly	When data is not saved correctly, an error message pops up	None	Displayed to the Curator
		1.6 ALT: Confirmation failed	Contains message update confirmation failure	When the curator does not confirm update. The use case ends	None	Supposed to be confirmed by the Curator

Requirement Number	Requirement Title:	Description:	Purpose:	Usage:	Entities & Attributes:	Logical Layout:
1.7	Delete Exhibition	1.7 Requests to delete an exhibition from the system.	To delete a selected exhibition from the system.	This is to remove an exhibition from the system	None	Requested by the curator.
		1.7 Deletes an exhibition from the system.	The action to delete the exhibition.	When the curator selects to delete the	Exhibition table: - ExhibitionID - ExhibitionName	Requested by the curator.

				exhibition .	Exhibition Description	
		1.7 Confirm to delete exhibition.	Curator confirms to delete exhibition.	When the deletion of the exhibition details needs to be confirmed.	None	Selected by curator.
		1.7 [ALT] Does not confirm to delete exhibition.	This is an option by the curator to not delete the exhibition.	To not confirm to exhibition .	None	Selected by curator

Requirement Number :	Requirement Title:	Description:	Purpose:	Usage:	Entities & Attributes:	Logical Layout:
1.8	View Artist	1.8 Requests to view an artist.	This entails the information that is	This is used when the curator wants to	None	Selected by the curator.

			required to view an artist.	view an artist.		
		1.18 Requested Search Parameter	The parameters which the artist can be searched from.	This is used when the curator wants to input the search parameter of an artist.	• Artist table: • ArtistID • ArtistName • ArtistSurname • ArtistType • ArtistBio • ArtistUserRole • ArtistPicture1	Provided by the curator.
		1.18 Display Selected Artist	This is used to select the artist from view results displayed to the curator.	This is used when a curator wants to view an artist from displayed view results search.	• None	Provided by the curator.

Requirement Number :	Requirement Title:	Description:	Purpose:	Usage:	Entities & Attributes:	Logical Layout:
1.9	Add Artwork	1.9 Request to add artwork	This is the when the creation of artwork is initiated.	This when the curator wishes adds artwork to the system.	None	Provided by the curator.
		1.9 Provide required details.	This contains all the details in the fields for adding an artwork.	This is used when the curator wants to add an artwork on the system.	Artwork table: - ArtworkTitle - ArtworkPrice - ArtworkPicture - ArtworkDescription SurfaceType table: - SurfaceType Frame table: - FrameColour	Provided by the curator
		1.9 Confirmation of provided details	To confirm the details that have been inputted onto the system	When the curator confirms that the captured details are correct.	None	Selected by the curator.

			are correct and accurate.			
		1.9[ALT]Does not confirm the process.	This is an option for the curator to not confirm the details.	Used when the curator does not confirm the details.	None	Selected by curator

Requirement Number:	Requirement Title:	Description:	Purpose:	Usage:	Entities & Attributes:	Logical Layout:
1.10	Search Artwork	1.10 Requests to search an artwork.	This entails the information that is required to search for an artwork.	This is used when the curator wants to search for an artwork	None	Selected by the curator.
		1.10 Requested Search Parameter	The parameters which the artwork can be	This is used when the curator wants to input the search	Artwork table: ArtworkTitle	Provided by the curator.

			searched from.	parameter of an artwork.		
		1.10 Selected Artwork	This is used to select the artwork from search results displayed to the curator.	This is used when a curator wants to select an artwork from displayed search results.	None	Selected by the curator.

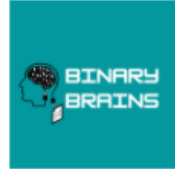
Requirement Number :	Requirement Title:	Description:	Purpose:	Usage:	Entities & Attributes:	Logical Layout:
1.11	Delete Artwork	1.11 Requests to delete an artwork from the system.	To delete a selected artwork from the system.	This is to remove an artwork from the system	None	Requested by the curator.

		1.11 Deletes an artwork from the system.	The action to delete the artwork.	When the curator selects to delete the artwork.	Artwork table: • ArtworkTitle • ArtworkPrice • ArtworkPicture • ArtworkDescription SurfaceType table: • SurfaceType Frame table: • FrameColour	Requested by the curator.
		1.11 Confirm to delete artwork.	Curator confirms to delete artwork.	When the deletion of the artwork details needs to be confirmed.	• None	Selected by curator.
		1.11 [ALT] Does not confirm to delete artwork.	This is an option by the curator to not delete the artwork.	To not confirm to exhibition.	• None	Selected by curator
					•	

Requirement Number:	Requirement title:	Description:	Purpose:	When Used:	Entities & Attributes:	Logical Layout:
1.12 Update Artwork Input						
1.12	Update Artwork	1.12 Wants to update artwork info	This contains the information required to initiate updating an artwork	This is used when the curator wants to make changes on existing artwork details	None	Selected by the Curator.
		1.12 Update Artwork Details info	This contains the new information entered the existing artwork details.	This is used when the curator is entering the new details into the existing artwork	Artwork table: • ArtworkTitle • Price • ArtPicture • MediumType	
		1.12 Confirm Update Artwork Info	This contains the confirmation details that is required	This is used when the user is satisfied	None	Selected by the Curator.

			to complete the update of the artwork details	with the new details for the specific artwork.		
		1.12 Deny Confirmation to update Artwork info	This contains the information for not confirming to update an artwork on the system.	This is used when the curator decides not to confirm to update the specific artwork.	None	Selected by Curator.
1.13 Create Artist Exhibition Invitation Input						
1.13	Create Artist Exhibition Invitation	1.13 Want to Create new Exhibition Invitation info	This contains the information required to initiate creating an exhibition invitation for an artist	This is used by the curator when the curator wants to invite qualified artists to an exhibition	None	Selected by the Curator.

		1.13 Create new Exhibition Invitation info	This contains the invitation details required to send to the artist	This is used by the curator when they want to provide all the invitation details about the exhibition	Invitation table: • InvitationDetails • InvitationDate • InvitationStatus • ArtistID • ExhibitionID	Entered by the Curator.
		1.13 Confirm Creation New Exhibition Invitation info	This contains the confirmation details that is required to complete the creation of the invitation	This is used when the user is satisfied with the new details for the invitation.	None	Selected by the Curator.

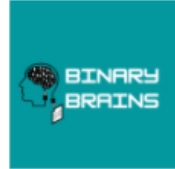


GROUP 23 DELIVERABLE 02



Requirement Number:	Requirement Title:	Description:	Purpose:	Usage:	Entities & Attributes:	Logical Layout:
1.14	Print Tags	Request to print Tag by clicking on print tags button.	This contains the information required to print tags of a specific exhibition	This is when the curator wishes to print tags of a specific exhibition.	None	Requested by the curator
		Set Printer, paper preference and Click Print.	This contains the information needed to print the tags	Used when the curator enters the details of the tags format, printer and paper.	None	Entered by the curator

Requirement Number:	Requirement Title:	Description:	Purpose:	Usage:	Entities & Attributes:	Logical Layout:
1.15	View Schedule	Request to view schedule and chooses to Filter dates	This contains the information where the curator clicks the View schedule button and choose the dates to filter between.	This is when the curator wishes to view the schedule of a period of time.	None	Requested by curator
		[ALT] Request to view Schedule with no dates used as filters.	This contains the information where the curator clicks the View schedule button and does not choose the dates to filter between.	This is when the curator wishes to view the entire schedule.	None	Entered by curator

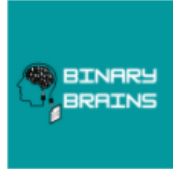


GROUP 23 DELIVERABLE 02



Requirement Number:	Requirement Title:	Description:	Purpose:	Usage:	Entities & Attributes:	Logical Layout:
1.16	Print Schedule	Request to view schedule and chooses to Filter dates	This contains the information where the curator clicks the View schedule button and choose the dates to filter between.	This is when the curator wishes to view the schedule of a period of time.	None	Requested by curator
		[ALT] Request to view schedule and not choose to Filter dates	This contains the information where the curator clicks the View schedule button and does not choose the dates to filter between.	This is when the curator wishes to view the full schedule.	None	Requested by curator
		Set Printer, paper preference and Click Print.	This contains the information needed to print the tags	Used when the curator enters the details of the tags format, printer and paper.	None	Entered by the curator

Requirement Number:	Requirement Title:	Description:	Purpose:	Usage:	Entities & Attributes:	Logical Layout:
1.17	Create Class	Request to create new artwork.	This contains the information required to create a new artwork.	This is when the curator clicks on Create New Artwork.	None	Requested by curator



GROUP 23 DELIVERABLE 02



		Enter class details	This contains the information required to create a specific type of class.	This is when the curator enters details about a class.	None	Entered by the curator
		End create Class process by confirming creation.	This contains the information required to confirm that new class is created.	Used to confirm that new class is created.	None	Entered by the curator

+

Requirement Number :	Requirement Title:	Description:	Purpose:	Usage:	Entities & Attributes:	Logical Layout:
1.18	Search Class	1.18 Requests to search a class.	This entails the information that is required to search for an art class.	This is used when the curator wants to search for an art class.	None	Selected by the curator.
		1.18 Requested Search Parameter	The parameters which the class can be searched from.	This is used when the curator wants to input the search parameter of a class.	Art_Class table: - Art_Cl assName - ArtClass SrtartDate	Provided by the curator.

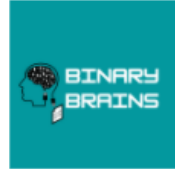
					ArtClass EndDate	
		1.18 Display Selected Class	This is used to select the class from search results displayed to the curator.	This is used when a curator wants to select a class from displayed search results.	None	Provided by the curator.

Requirement Number :	Requirement Title:	Description:	Purpose:	Usage:	Entities & Attributes:	Logical Layout:
1.19	Update Class	1.19Request to update a class.	This is the process of updating existing information about the class.	This is used when a curator wants to update the details of a class.	None	Selected by the curator.
		1.19 Update the class details	The details of the class that the curator	This is used when the curator enters the	- Art- _ClassName - Art_ClassDescription	Provided by the curator.

			wants to update.	updated details of the class.	• ArtClassSrt artDate • ArtClassEndDate	
		1.19 Confirm Changes made to the Class	The act of confirming the updated class details	This happens when the curator confirms the update of the class details.	• None	Selected by the curator
		1.19 [ALT] Does not confirm to changes made to the class.	The process of not confirming to update details.	This happens when the curator does not confirm to update the class.	None	Selected by curator

Requirement Number :	Requirement Title:	Description:	Purpose:	Usage:	Entities & Attributes:	Logical Layout:
1.20	Delete Class	1.20 Requests to delete a class	To delete a selected art	This is to remove an art	None	Requested by the curator.

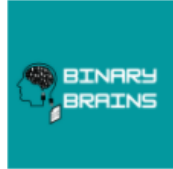
		from the system.	class from the system.	class from the system		
		Deletes a class from the system.	The action to delete the art class.	When the curator selects the delete button to delete the art class.	Art_Class table: • ArtClassID • ArtClassName • ArtClassStartDate • ArtClassEndDate	Requested by the curator.
		1.20 Confirm to delete art class.	Curator confirms to delete art class.	When the deletion of art class details needs to be confirmed.	• None	Selected by curator.
		1.20 [ALT] Does not confirm to delete art class	This is an option by the curator to not delete the art class.	To not confirm to delete class.	• None	Selected by curator



GROUP 23 DELIVERABLE 02



Requirement Number:	Requirement Title:	Description:	Purpose:	Usage:	Entities & Attributes:	Logical Layout:
1.21	Generate Class reminder	1.20 Wants to create a class reminder	To prompt the system of wanting to generate a report.	This happens two weeks before a class takes place.	None	Prompted by time.
		1.20 Retrieve Art Class booking information.	Retrieves Art Class information from the Invoke "Search class" to create the report. When time is prompted by the system for the creation of the report.	This happens when there are two weeks to go to the art class.	Art_Class table: - ArtClassName - ArtClassStartDate - ArtClassEndDate - ArtClass_Typeld Booking table: - Date - Time Payment table: - PaymentStatus	System Searches for the information
		1.20 Retrieve User information	When time is prompted by the system for the creation	This happens when time prompts	User table: userEmailAddress	System searches the information.

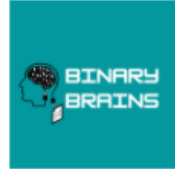


GROUP 23 DELIVERABLE 02



			of the report, to retrieve the User table with the Register User invoke	the system to create a class reminder report.		

Requirement Number:	Requirement Title:	Description:	Purpose:	Usage:	Entities & Attributes:	Logical Layout:
1.22	Generate Payment Reminder	1.21 Wants to create a payment reminder	To prompt the system of wanting to generate a report.	This happens a few days before a class takes place.	None	Prompted by time
		1.21 Retrieves Art Class information from the Invoke "Search class" to	When time is prompted by the system for the creation of the report.	This happens when time prompts the system to create a payment	Payment table: • Payment Amount • Payment Date Art_Class table: • ArtClassPrice	System searches for the information.



GROUP 23 DELIVERABLE 02



		create the report.		reminder report.	Booking Table: • BookingDate • BookinTime	
		1.21 Retrieve User information	When time is prompted by the system for the creation of the report, To retrieve the User table with the Register User invoke	This happens when time prompts the system to create a class reminder report.	• User table: UserEmailAddresses	System searches the information.

Requirement Number :	Requirement Title:	Description:	Purpose:	Usage:	Entities & Attributes:	Logical Layout:
1.23	Create Class Teacher	1.23 Request to be registered	This is the when the creation	This when the art class teacher	• None	Provided by the art class teacher.

			of a class teacher is initiated.	wishes to be added onto the system.		
		1.23 Provide required details.	This contains all the details in the fields for creating an art class teacher.	This is used when the curator wants to create a class teacher on the system, that requests to become a teacher.	ClassTeacher table: • ClassTeacher_ID • ClassTeacher_Name • ClassTeacher_Surname • ClassTeacher_EmailAddress • ClassTeacher_AddressLine1 • ClassTeacher_AddressLine2 • ClassTeacher_Medium	Provided by the art class teacher to the curator.
		1.23 Confirmation of provided details	To confirm the details that have been inputted onto the system are correct and accurate.	When the curator confirms that the captured details are correct.	• None	Selected by the curator.

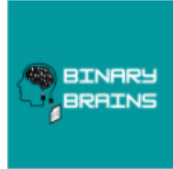
		1.23 [ALT]Does not confirm the process.	This is an option for the curator to not confirm the details.	Used when the curator does not confirm the details.	None	Selected by curator

Requirement Number:	Requirement Title:	Description:	Purpose:	Usage:	Entities & Attributes:	Logical Layout:
1.24	Search Class Teacher	1.24 Requests to search a class teacher.	This entails the information that is required to search for an art class teacher.	This is used when the curator wants to search for an art class teacher.	None	Selected by the curator.
		1.24 Requested Search Parameter	The parameters which the class teacher can be searched from.	This is used when the curator wants to input the search parameter	ClassTeacher table: - ClassTeacher_Name - ClassTeacher_Surname - ClassTeacher_EmailAddress	Provided by the curator.

				of a class teacher.	• ClassTeacher_Medium • ClassTeacher_AddressLine1 • ClassTeacher_AddressLine2	
		1.24 Selected Class	This is used to select the class teacher from search results displayed to the curator.	This is used when a curator wants to select a class teacher from displayed search results.	• None	Selected by the curator.

Requirement Number:	Requirement Title:	Description:	Purpose:	Usage:	Entities & Attributes:	Logical Layout:
1.25	Update Class Teacher	1.25Request to update a class teacher.	This is the process of updating existing details and information	This is used when a curator wants to update the details of a	• None	Selected by the curator.

			about the class teacher.	class teacher.		
		1.25 Provide updated changes that the class teacher wants to update.	The details of the class teacher that the curator wants to update.	This is used when the curator enters the updated details of the class teacher.	• ClassTeacher Name • ClassTeacher Surname • ClassTeacher Email Address • ClassTeacher Medium • ClassTeacher AddressLine1 • ClassTeacher AddressLine2	Provided by the curator.
		1.25 Confirm Changes made	The act of confirming the updated class teacher details.	This happens when the curator confirms the update of the class teacher details.	• None	Selected by the curator
		1.25 [ALT] Does not confirm to changes made	The process of not confirming to update their details.	This happens when the curator does not confirm to	None	Selected by curator.

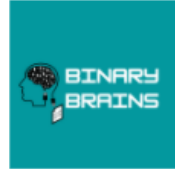


GROUP 23 DELIVERABLE 02



				update the class.		

Requirement Number:	Requirement Title:	Description:	Purpose:	Usage:	Entities & Attributes:	Logical Layout:
1.26	Delete Class	1.26 Requests to delete a class from the system.	To delete a selected art class from the system.	This is to remove an art class from the system	None	Requested by the curator.
		1.26 Deletes a class from the system.	The action to delete the art class.	When the curator selects the delete button to delete the art class.	Art_Class table: • ArtClassID • ArtClassName • ArtClassStartDate • ArtClassEndDate	Requested by the curator.
		1.26 Confirm to delete art class.	Curator confirms to delete art class.	When the deletion of art class details needs to be confirmed.	• None	Selected by curator.
		1.26 ALT] Does not	This is an option by	To not confirm to	• None	Selected by curator

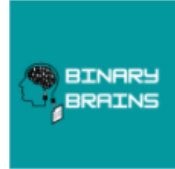


GROUP 23 DELIVERABLE 02



		confirm to delete art class	the curator to not delete the art class.	delete class.		

Requirement Number:	Requirement Title:	Description:	Purpose:	Usage:	Entities & Attributes:	Logical Layout:
1.27	Generate Booking Information	Request to create a new booking notification	This contains the information required to create a new booking notification	This is when the curator wishes to create a booking notification for a specific class that will help the user get notified of the details of the art class that they have booked for.	None	Requested by the curator
		Details of the booking	This contains the details of the booking notification.	Used when the curator enters the details of the booking notification in order to create it. These details will be unique of the type of class booked for.	Booking Table: • Booking ID • Public User ID • Art Class ID • Date • Time • Payment Status • Booking Status	Entered by the curator
		Confirmation of the creation process.	An option that is selected by the curator to confirm the captured information.	When the curator confirms that the captured information is correct to proceed with the creation process.	None	Selected by the curator

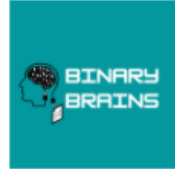


GROUP 23 DELIVERABLE 02



		[ALT] Does not confirm the process	This is an option selected by the curator to reject the captured information or cancel the process.	Used when the curator wishes to not confirm that the captured details are correct or to not move ahead with the creation process.	None	Selected by the curator

Requirement Number:	Requirement Title:	Description:	Purpose:	Usage:	Entities & Attributes:	Logical Layout:
1.28	Create Medium Detail	Request to create a medium detail.	This contains the information required to create a new medium detail.	This is when the curator wishes to create a medium to use it to find related artworks.	None	Requested by curator
		Details of the medium	This contains the details of the medium	Used when the curator enters the details of the medium in order to create it.	Medium Type Table: - Medium Type ID - Medium Type Description	Entered by curator
		Confirmation of the creation process.	An option that is selected by the curator to confirm the captured information.	When the curator confirms that the captured information is correct and proceeds with the creation process.	None	Selected by the curator
		[ALT] Does not confirm the process	This is an option selected by the curator to reject the captured information or cancel the process.	Used when the curator wishes to not confirm that the captured details are correct or to not move ahead with the creation process.	None	Selected by the curator

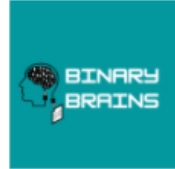


GROUP 23 DELIVERABLE 02



Requirement Number:	Requirement Title:	Description:	Purpose:	Usage:	Entities & Attributes:	Logical Layout:
1.29	Search Medium	Request to search a medium.	This contains the information required to search for a medium.	This is when the curator wishes to search for a medium to use it to find related artworks.	None	Requested by curator
		Details of the medium	This contains the details of the medium entered in order to search for it.	Used when the curator enters details of the medium in order to search for its related artworks.	Medium Type Table: - Medium Type ID - Medium Type Description	Entered by curator
		Confirmation of the search process.	An option that is selected by the curator to confirm the captured information.	When the curator confirms that the captured information is correct and proceeds with the search process.	None	Selected by the curator
		[ALT] Does not confirm the process	This is an option selected by the curator to reject the captured information or cancel the process.	Used when the curator wishes to not confirm that the captured details are correct or to not move ahead with the search process.	None	Selected by the curator

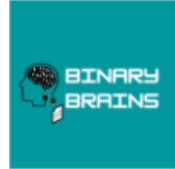
Requirement Number:	Requirement Title:	Description:	Purpose:	Usage:	Entities & Attributes:	Logical Layout:
1.30	Update Medium Detail	Request to update	This contains the information required to	This is when the curator wishes to update the details of a medium to	None	Requested by curator



GROUP 23 DELIVERABLE 02



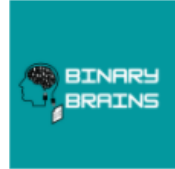
		medium detail.	update a medium detail.	use it to find related artworks better.		
		Selection of medium.	This contains the information required to update a specific type of medium detail after the results of the search are returned to the curator.	This is when the curator selects a specific medium detail that is needed to be updated.	Medium Type Table: • Medium Type ID • Medium Type Description	Selected by the curator
		Details of Medium.	This contains the details of the medium entered in order to update it.	Used to update a mediums details in order to find the artworks related to that medium. New details are entered.	Medium Type Table: • Medium Type ID • Medium Type Description	Entered by the curator
		Confirmation of the update process.	An option that is selected by the curator to confirm the captured information.	When the curator confirms that the captured information is correct and proceeds with the update process.	None	Selected by the curator
		[ALT] Does not confirm the process	This is an option selected by the curator to reject the captured information or cancel the process.	Used when the curator wishes to not confirm that the captured details are correct or to not move ahead with the update process.	None	Selected by the curator



GROUP 23 DELIVERABLE 02



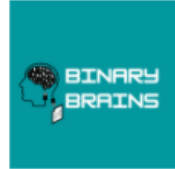
Requirement Number:	Requirement Title:	Description:	Purpose:	Usage:	Entities & Attributes:	Logical Layout:
1.31	Delete Medium	Request to delete medium.	This contains the information required to delete a medium.	This is when the curator wishes to delete a medium that is no longer required, as well as delete all of its attachments with the artworks.	None	Requested by curator
		Selection of medium.	This contains the information required to delete a specific type of medium after the results of the search are returned to the curator.	This is when the curator selects a specific medium that is needed to be delete.	Medium Type Table: - Medium Type ID - Medium Type Description	Selected by the curator
		Confirmation of the delete process.	An option that is selected by the curator to confirm the captured information.	When the curator confirms that the captured information is correct and proceeds with the delete process.	None	Selected by the curator
		[ALT] Does not confirm the process	This is an option selected by the curator to reject the captured information or cancel the process.	Used when the curator wishes to not confirm that the captured details are correct or to not move ahead with the delete process.	None	Selected by the curator



GROUP 23 DELIVERABLE 02



Requirement Number:	Requirement Title:	Description:	Purpose:	Usage:	Entities & Attributes:	Logical Layout:
1.32	Create Artwork Detail	Request to create an artwork detail.	This contains the information required to create a new artwork detail.	This is when the curator wishes to create an artwork in order to display art pieces that can later be used by users to view.	None	Requested by curator
		Details of the artwork	This contains the details of the artwork	Used when the curator enters the details of the artwork in order to create it.	Artwork Table: • Artwork ID • Artwork Title • Price • Art Picture • Artwork Status • Surface Type ID • Medium Type • Exhibition ID • Artist ID • Frame Colour ID • Frame Dimension ID	Entered by curator
		Confirmation of the creation process.	An option that is selected by the curator to confirm the captured information.	When the curator confirms that the captured information is correct and proceeds with the creation process.	None	Selected by the curator
		[ALT] Does not confirm the process	This is an option selected by the curator to reject the captured information or cancel the process.	Used when the curator wishes to not confirm that the captured details are correct or to	None	Selected by the curator

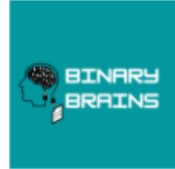


GROUP 23 DELIVERABLE 02



				not move ahead with the creation process.		

Requirement Number:	Requirement Title:	Description:	Purpose:	Usage:	Entities & Attributes:	Logical Layout:
1.33	Search artwork	Request to search an artwork.	This contains the information required to search for an artwork.	This is when the curator wishes to search for an artwork to use it to find art pieces.	None	Requested by curator
		Details of the artwork	This contains the details of the artwork entered in order to search for it.	Used when the curator enters the details of the artwork in order to search for art pieces that match the entered details.	Artwork Table: • Artwork ID • Artwork Title • Price • Art Picture • Artwork Status • Surface Type ID • Medium Type • Exhibition ID • Artist ID • Frame Colour ID • Frame Dimension ID	Entered by curator
		Confirmation of the search process.	An option that is selected by the curator to confirm the captured information.	When the curator confirms that the captured information is correct and proceeds with the search process.	None	Selected by the curator

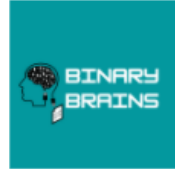


GROUP 23 DELIVERABLE 02



		[ALT] Does not confirm the process	This is an option selected by the curator to reject the captured information or cancel the process.	Used when the curator wishes to not confirm that the captured details are correct or to not move ahead with the search process.	None	Selected by the curator

Requirement Number:	Requirement Title:	Description:	Purpose:	Usage:	Entities & Attributes:	Logical Layout:
1.34	Update artwork Detail	Request to update artwork detail.	This contains the information required to update an artwork detail.	This is when the curator wishes to update the details of an artwork to use it to find art pieces more efficiently.	None	Requested by curator
		Selection of artwork.	This contains the information required to update a specific type of medium detail after the results of the search are returned to the curator.	This is when the curator selects a specific artwork detail that is needed to be updated.	Artwork Table: • Artwork ID • Artwork Title • Price • Art Picture • Artwork Status • Surface Type ID • Medium Type • Exhibition ID • Artist ID • Frame Colour ID • Frame Dimension ID	Selected by the curator
		Details of artwork.	This contains the details of the artwork entered in order to update it.	Used to update an artworks details with the new information.	Artwork Table: • Artwork ID • Artwork Title • Price • Art Picture	Entered by the curator

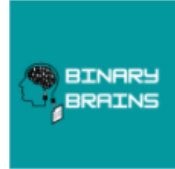


GROUP 23 DELIVERABLE 02



					• Artwork Status • Surface Type ID • Medium Type • Exhibition ID • Artist ID • Frame Colour ID • Frame Dimension ID	
		Confirmation of the update process.	An option that is selected by the curator to confirm the captured information.	When the curator confirms that the captured information is correct and proceeds with the update process.	None	Selected by the curator
		[ALT] Does not confirm the process	This is an option selected by the curator to reject the captured information or cancel the process.	Used when the curator wishes to not confirm that the captured details are correct or to not move ahead with the update process.	None	Selected by the curator

Requirement Number:	Requirement Title:	Description:	Purpose:	Usage:	Entities & Attributes:	Logical Layout:
1.35	Delete artwork	Request to delete artwork.	This contains the information required to delete an artwork.	This is when the curator wishes to delete an artwork that is no longer required or is outdated and needs to be removed.	None	Requested by curator

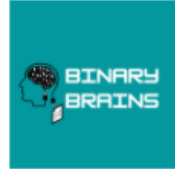


GROUP 23 DELIVERABLE 02



		Selection of artwork.	This contains the information required to delete a specific type of artwork after the results of the search are returned to the curator.	This is when the curator selects a specific artwork that is needed to be delete.	Artwork Table: • Artwork ID • Artwork Title • Price • Art Picture • Artwork Status • Surface Type ID • Medium Type • Exhibition ID • Artist ID • Frame Colour ID • Frame Dimension ID	Selected by the curator
		Confirmation of the delete process.	An option that is selected by the curator to confirm the captured information.	When the curator confirms that the captured information is correct and proceeds with the delete process.	None	Selected by the curator
		[ALT] Does not confirm the process	This is an option selected by the curator to reject the captured information or cancel the process.	Used when the curator wishes to not confirm that the captured details are correct or to not move ahead with the delete process.	None	Selected by the curator

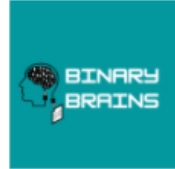
Requirement Number:	Requirement Title:	Description:	Purpose:	Usage:	Entities & Attributes:	Logical Layout:



GROUP 23 DELIVERABLE 02



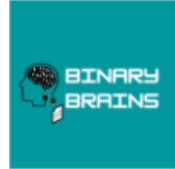
1.36	Archive Status	Request to exhibit the artwork in an exhibition and change status.	This contains the information required to initiate the change of an artwork status.	This is used when the curator wishes to exhibit an artwork.	None	Requested by curator
		[ALT] Request to display artwork on the wall and change status.	This contains the information required to initiate the change of an artwork status.	This is used when the curator wishes to display an artwork.	None	Requested by curator
		[ALT] Request to store artwork in the storeroom and change status.	This contains the information required to initiate the change of an artwork status.	This is used when the curator wishes to store an artwork.	None	Requested by curator
		[ALT] Request to return the artwork to the artist and change status.	This contains the information required to initiate the change of an artwork status.	This is used when the curator wishes to request the return of an artwork.	None	Requested by curator
		[ALT] Request to return artwork and change status.	This contains the information required to initiate the change of an artwork status.	This is used when the curator has returned an artwork.	None	Requested by curator
		Update Artwork status to "Exhibition"	When artworks first come in and needs to be a part of an exhibition. This is when	It shows the status of the artworks that have not been bought in an exhibition. This helps	Artwork Status Table Artwork Status ID	Updated by curator



GROUP 23 DELIVERABLE 02



			the status will be needed to be updated.	keep track of what to do with the artworks left from the exhibition.	• Artwork Status Description	
		[ALT] Update Artwork status to "Display"	After the exhibition the artwork is presented on the wall for a week. This is when the status will be needed to be updated.	It shows the status of the artwork if it has not been bought in an exhibition, and keep track of the artworks left on display in the gallery.	Artwork Status Table • Artwork Status ID • Artwork Status Description	Updated by curator
		[ALT] Update Artwork status to "Store room"	After a week of the artwork displayed on the wall, the artwork is put in a store room. This is when the status will be needed to be updated.	It shows the status of the artwork and helps keep track of how many artworks are kept in the store room to be returned to their respective artists.	Artwork Status Table • Artwork Status ID • Artwork Status Description	Updated by curator
		[ALT] Update Artwork status to "Request Return"	From the storeroom, the artwork is to be returned to the owner. This is when the status will be needed to be updated.	It shows the status of the artworks that have been requested to be returned already. This helps keep track how many artists to inform to come collect their artwork.	Artwork Status Table • Artwork Status ID • Artwork Status Description	Updated by curator
		[ALT] Update Artwork status to "Returned"	Once the artwork has been returned to their respective artists, the status will be needed to be updated.	It shows the status of the artwork when it has been returned in order to keep track of the artworks that are still left within the gallery.	Artwork Status Table • Artwork Status ID • Artwork Status Description	Updated by curator

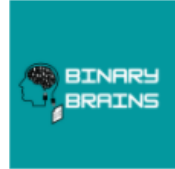


GROUP 23 DELIVERABLE 02



Requirement Number:	Requirement Title:	Description:	Purpose:	Usage:	Entities & Attributes:	Logical Layout:
1.37	Create Surface Type Detail	Request to create a Surface Type detail.	This contains the information required to create a new Surface Type detail.	This is when the curator wishes to create a Surface Type to use it to find related artworks.	None	Requested by curator
		Details of the Surface Type	This contains the details of the Surface Type	Used when the curator enters the details of the Surface Type in order to create it.	Surface Type Table: - Surface Type ID - Surface Type Description	Entered by curator
		Confirmation of the creation process.	An option that is selected by the curator to confirm the captured information.	When the curator confirms that the captured information is correct and proceeds with the creation process.	None	Selected by the curator
		[ALT] Does not confirm the process	This is an option selected by the curator to reject the captured information or cancel the process.	Used when the curator wishes to not confirm that the captured details are correct or to not move ahead with the creation process.	None	Selected by the curator

Requirement Number:	Requirement Title:	Description:	Purpose:	Usage:	Entities & Attributes:	Logical Layout:

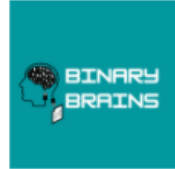


GROUP 23 DELIVERABLE 02



1.38	Search Surface Type	Request to search a Surface Type.	This contains the information required to search for a Surface Type.	This is when the curator wishes to search for a Surface Type to use it to find related artworks.	None	Requested by curator
		Details of the Surface Type	This contains the details of the Surface Type entered in order to search for it.	Used when the curator enters details of the Surface Type in order to search for its related artworks.	Surface Type Table: • Surface Type ID • Surface Type Description	Entered by curator
		Confirmation of the search process.	An option that is selected by the curator to confirm the captured information.	When the curator confirms that the captured information is correct and proceeds with the search process.	None	Selected by the curator
		[ALT] Does not confirm the process	This is an option selected by the curator to reject the captured information or cancel the process.	Used when the curator wishes to not confirm that the captured details are correct or to not move ahead with the search process.	None	Selected by the curator

Requirement Number:	Requirement Title:	Description:	Purpose:	Usage:	Entities & Attributes:	Logical Layout:
1.39	Update Surface Type Detail	Request to update Surface Type detail.	This contains the information required to update a Surface Type detail.	This is when the curator wishes to update the details of a Surface Type to use it to find related artworks better.	None	Requested by curator



GROUP 23 DELIVERABLE 02

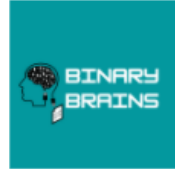


		Selection of Surface Type.	This contains the information required to update a specific type of Surface Type detail after the results of the search is returned to the curator.	This is when the curator selects a specific Surface Type detail that is needed to be updated.	Surface Type Table: • Surface Type ID • Surface Type Description	Selected by the curator
		Details of Surface Type.	This contains the details of the Surface Type entered in order to update it.	Used to update a Surface Type details in order to find the artworks related to that Surface Type. New details are entered.	Surface Type Table: • Surface Type ID • Surface Type Description	Entered by the curator
		Confirmation of the update process.	An option that is selected by the curator to confirm the captured information.	When the curator confirms that the captured information is correct and proceeds with the update process.	None	Selected by the curator
		[ALT] Does not confirm the process	This is an option selected by the curator to reject the captured information or cancel the process.	Used when the curator wishes to not confirm that the captured details are correct or to not move ahead with the update process.	None	Selected by the curator

Requirement Number:	Requirement Title:	Description:	Purpose:	Usage:	Entities & Attributes:	Logical Layout:

1.40	Delete Surface Type	Request to delete Surface Type.	This contains the information required to delete a Surface Type.	This is when the curator wishes to delete a Surface Type that is no longer required, as well as delete all of its attachments with the artworks.	None	Requested by curator
		Selection of Surface Type.	This contains the information required to delete a specific type of Surface Type after the results of the search are returned to the curator.	This is when the curator selects a specific Surface Type that is needed to be delete.	Surface Type Table: - Surface Type ID - Surface Type Description	Selected by the curator
		Confirmation of the delete process.	An option that is selected by the curator to confirm the captured information.	When the curator confirms that the captured information is correct and proceeds with the delete process.	None	Selected by the curator
		[ALT] Does not confirm the process	This is an option selected by the curator to reject the captured information or cancel the process.	Used when the curator wishes to not confirm that the captured details are correct or to not move ahead with the delete process.	None	Selected by the curator

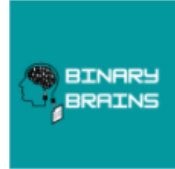
Requirement Number:	Requirement Title:	Description:	Purpose:	Usage:	Entities & Attributes:	Logical Layout:
1.41	View Art Class	Request to View	This contains the information required to	This is when the curator wishes to view the feedback form for the	None	Requested by curator



GROUP 23 DELIVERABLE 02



	Feedback Form	Feedback form.	view an art class feedback form.	art classes and make changes accordingly to those art classes.		
		Selection of Feedback form	This contains the information required to view a specific type of Feedback form after the results of the request are returned to the curator.	This is when the curator selects a specific feedback form that is to be viewed.	Feedback Table: • Feedback Page ID • Public User ID • Artist ID • Art Class ID • Feedback Comment • Rating Teacher • Rating Food • Rating Difficulty • Overall Rating	Selected by Curator
		[ALT] No selection of feedback form made	This contains information that shows that no selection has been made to view any art class feedback form.	This is when there is no form that exists for the curator to select from.	None	Selected by curator



GROUP 23 DELIVERABLE 02



Requirement Number:	Requirement Title:	Description:	Purpose:	When Used:	Entities & Attributes:	Logical Layout:
2.1 Book Art Class Input						
2.1	Book Art Class	2.1 Request to Book for an Art class	This contains the information required to initiate the booking for an Art Class.	This is used when the user wants to book for an Art Class.	None	Selected by the User.
		2.1 Selects to Book for an Art Class	Option that is selected to Book for the Art Class to open the booking.	This is used when the user has selected a class they want to book for.	None	Selected by the User.
		2.1 Details of the Art Class Booking	This contains the details of the Art Class Booking which is a prerequisite for the booking.	This is used when the user enters their details when booking for an Art Class	PublicUser table: • UserName • UserSurname • UserEmailaddress Art_Class table: • ArtClassType • ArtClassStartDate • ArtClassEndDate • ClassPrice • ArtClassName	Entered by the User.

		2.1 Makes payment for Booking	This contains the action to make payment for booking which is required to confirm seat.	This is used when the user decides to make payment	None	Selected by User.
		2.1 ALT: Does not make payment for booking	This is an option selected by the user when they reject to make payment	This is used when the user decides not to make a payment for the booking.	None	Selected by the User.
		2.1 User submits their booking details	This is an option selected by the user to submit the captured information	This is used when the user is completed with the booking details and submits the information.	None	Selected by the User.
2.2 View Existing Classes Input						
2.2	View Existing Classes	2.2 Request to View an Art Class	This contains the information required to	This is used when the user wants to view an	None	Selected by the user.

			initiate to View an Existing Art Class.	existing art class.		
		2.2 Selects to view details of an existing class	This an option selected by the user to view all the details of the existing class	This is used when the user wants to see more information and options of an existing class	None	Selected by the user.
2.3 Cancel Booking Input						
2.3	Cancel Booking	2.3 Request to Cancel a Booking	This contains the information required to initiate Cancel booking	This is used when the user wants to cancel a booking, they made	None	Selected by the user.
		2.3 Selects to cancel the booking	This is an option the user selects to cancel the booking they made	This is used when the user cancels a specific booking they made.	None	Selected by the user
		2.3 Confirms cancellation of booking	This is an option to complete the cancellation	This is used when the user confirms	None	Selected by the User

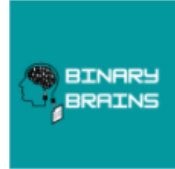
			by confirming cancel booking.	cancel booking		
		2.3 ALT: Does not confirm cancel booking	This is the option where the user does not confirm the cancellation of a booking	This is used when the user does not confirm to cancel booking	None	Selected by the user.
2.4 Generate Booking Confirmation Message Input						
2.4	Generate Booking Confirmation Message	2.4 Requests to generate booking confirmation message	This contains the information required to initiate generate a booking confirmation message	This is used when the user wants to generate a booking confirmation message	None	Selected by the user
		2.4 Selects to generate booking confirmation message	This is the option where the user selects to generate the booking confirmation message	This is used when the user selects to generate a booking confirmation message	None	Selected by the user
2.5 Refund request Input						

2.5	Refund request	2.5 Requests to view refund status	This contains the information required to initiate to view refund request	This is used when the user wants to view their refund request status	None	Selected by the user.
		2.5 Selects to view refund request	This is the option the user selects to view their requested refund status	This is used when the user wants to view their specific refund requested status	None	Selected by the user.
2.6 Capture Class Feedback Input						
2.6	Capture Class Feedback	2.6 Requests to provide feedback	This contains the information required to initiate provide feedback details	This is used when the user wants to provide feedback details on a class, they had	None	Selected by the User.
		2.6 Selects Provide feedback for class	This is the option the user selects to provide feedback for	This is used when the user wants to provide their feedback	None	Selected by the user.

			that specific class	details for the class, they had		
		2.6 Details of the feedback form	This contains the details of the feedback provided by the user	This is used when the user enters their feedback details in the feedback form	Feedback table: - FeedbackComment - RatingTeacher - RatingFood - RatingDifficulty - OverallRating	Entered by the user.
		2.6 Submit feedback form	This is the option to submit the entered feedback details	This is used when the user is satisfied with their provided details and submits their feedback form	None	Selected by the user.
		2.6 Confirms feedback details	This is the option selects to confirm feedback details into the system which is a	This is used when the user wants confirms their feedback details	None	Selected by the user

			prerequisite to submit details			
		2.6 ALT: Does not confirm feedback details	This is the option selected to not confirm feedback details into the system	This is used when the user is not satisfied with their feedback details and does not confirm the feedback form	None	Selected by the user.
2.7 Generate Class Teacher Information Input						
2.7	Generate Class Teacher information	2.7 Requests to generate class teacher information	This contains the information required to initiate generate class teacher information	This is used when the user wants to generate a class teachers' information	None	Selected by the User.
		2.7 Selects to view class teacher information	This is the option selected to generate the class teacher's information	This is used when the user wants to view the selected class	None	Selected by the user.

				teacher's information		
2.8 Make Payment Input						
2.8	Make Payment	2.8 Requests to make payment	This contains the information required to initiate make payment	This is used when the user wants to make a payment for their booking	None	Selected by the user.
		2.8 Selects payment option	This is an option to select a provided payment option for the booking which is a prerequisite to make the payment	This is used when the user wants to select a payment option for the booking	None	Selected by the user.
		2.8 Selects payment method	This is an option to select a payment method for their payment of the booking	This is used when the user wants to select a payment method for the booking	None	Selected by the user.



GROUP 23 DELIVERABLE 02



		2.8 Card details	This contains the details of the user's bank card which is required to make payment	This is used when the user wants to enter their card details to proceed payment	None	Entered by the user.
		2.8 Submit card details	This is an option for the user to submit their card details.	This is used when the user is satisfied and wants to submit their card details.	None	Selected by the user.
		2.8 Confirm amount due	This is an option to confirm the amount that is due to the user for the booking	This is used when the user confirms the amount that is due for the booking, they made	None	Selected by the user.
		2.8 ALT: Does not confirm amount due	This is an option to not confirm the amount that is due.	This is used when the user does not agree with the amount due leading to	None	Selected by the user.

				not confirming		
		2.8 One-Time-Pin details	This contains the one-time-pin details to complete the transaction	This is used when the user enters the one-time-pin they received from the bank.	None	Entered by the user.
2.9 Request Refund Input						
2.9	Request Refund	2.9 Requests refund payment	This contains the information to initiate the request for a refund	This is used when the user wants a refund for a payment made	None	Selected by the user
		2.9 Selects refund request	This is an option to make a request for a refund	This is used when the user requests to make a refund for a specific class.	None	Selected by the user.
		2.9 Confirms refund request	This option is selected when the user wants to	This is used when the user wants to confirm	None	Selected by the user.

			confirm their refund request	their refund request		
		2.9 ALT: Does not confirm refund request	This option is selected when the user does not want to confirm their refund request	This is used when the user does not confirm their refund request.	None	Selected by the user

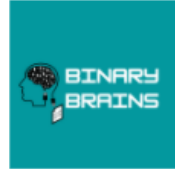
Requirement Number:	Requirement Title:	Description:	Purpose:	When Used:	Entities & Attributes:	Logical Layout:
3.1 Generate New User Report Input						
3.1	Generate New User Report	3.1 Request to generate New User report	This contains the information required to initiate generate new user report	This is used when the curator wants to generate new user	None	Selected by the Curator

				report details		
		3.1 Provides period range details	This contains the period details to capture report details within the given range	This is used when the user provides the range in which they want to receive the report details	None	Entered by the curator
3.2 Generate Art Class Report Input						
3.2	Generate Art Class Report	3.2 Request to generate Art Class report	This contains the information required to initiate generate Art Class report	This is used when the curator wants to generate Art Class report details	None	Selected by the Curator
		3.2 Provides period range details	This contains the period details to capture report details	This is used when the user provides the range in which	None	Entered by the curator

			within the given range	they want to receive the report details		
3.3 Generate Class Teacher Report Input						
3.3	Generate Class Teacher Report	3.3 Request to generate Class Teacher report	This contains the information required to initiate generate Class Teacher report	This is used when the curator wants to generate Class Teacher report details	None	Selected by the Curator
		3.3 Provides period range details	This contains the period details to capture report details within the given range	This is used when the user provides the range in which they want to receive the report details	None	Entered by the curator
3.4 Generate Exhibition Report Input						
3.4	Generate Exhibition Report	3.4 Request to generate	This contains the information	This is used when the	None	Selected by the Curator

		Exhibition report	required to initiate generate Exhibition report	curator wants to generate Exhibition report details		
		3.4 Provides period range details	This contains the period details to capture report details within the given range	This is used when the user provides the range in which they want to receive the report details	None	Entered by the curator
3.5 Generate Class Survey Report Input						
3.5	Generate Class Survey Report	3.5 Request to generate Class Survey report	This contains the information required to initiate generate Class Survey report	This is used when the curator wants to generate Class Survey report details	None	Selected by the Curator

		3.5 Provides period range details	This contains the period details to capture report details within the given range	This is used when the user provides the range in which they want to receive the report details	None	Entered by the curator
3.6 Generate Payment Report Input						
3.6	Generate Payment Report	3.6 Request to generate Payment report	This contains the information required to initiate generate Payment report	This is used when the curator wants to generate Payment report details	None	Selected by the Curator
		3.6 Provides period range details	This contains the period details to capture report details within the given range	This is used when the user provides the range in which they want to receive	None	Entered by the curator



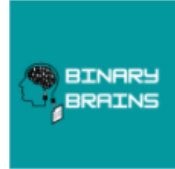
GROUP 23 DELIVERABLE 02



				the report details		

Requirement Number:	Requirement Title:	Description:	Purpose:	Usage:	Entities & Attributes:	Logical Layout:
4.1	Register User	Request to be registered on the system	This contains the information required for a user to register on the system	This is when the user wants to register into the system.	None	Requested by the User
		Enter details.	This contains the details that the user enters into the system in order to be registered	Used when the user enters their personal details on the system so that they can be registered.	None	Entered by the User
		Submit details	This contains information that allows the user details to be submitted to the system.	When the user wants to submit his/her details onto the system in order to be saved and successfully registered.	None	Submitted by the User

Requirement Number:	Requirement Title:	Description:	Purpose:	Usage:	Entities & Attributes:	Logical Layout:
4.2	Log in	Request to be log in to the system	This contains the information required for a user to log in the system	This is when the user wants to log in into the system.	None	Requested by the User
		Enter details.	This contains the details that the user	Used when the user enters their personal	None	Entered by the User



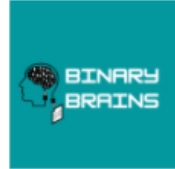
GROUP 23 DELIVERABLE 02



			enters into the system in order to log in	details on the system so that they log in		
		Submit details	This contains information that allows the user details to be submitted to the system.	When the user wants to submit his/her details onto the system in order to be saved and successfully log in.	None	Selected by the User

Requirement Number:	Requirement Title:	Description:	Purpose:	Usage:	Entities & Attributes:	Logical Layout:
4.3	Log out	Request to be log out of the system	This contains the information required for a user to log out of the system	This is when the user wants to log out of the system.	None	Requested by the User
		Select to log out	This contains the details of the option that the user selects to log out of the system	Used when the user selects the option to log out of the system	None	Selected by the User
		[ALT] Select to not log out	This contains the details of the option that the user selects to not log out of the system	Used when the user selects the option to cancel the log out process.	None	Selected by the User

Requirement Number:	Requirement Title:	Description:	Purpose:	Usage:	Entities & Attributes:	Logical Layout:
4.4	Reset Password	Request to get a new password	This contains the information that helps users get a new password	This is when the user forgets their password and requests for a new one.	None	Requested by the User
		Select option to Reset password	This contains the details that the user selects in order to reset their password	Used when the user selects the option to reset their password	None	Selected by the User
		Enter Password Details	This contains information that allows the password details to be entered to the system.	When the user wants to enter details of their new password onto the system	None	Entered by the User
		Confirm Password reset process	This contains the details that the user selects in order to confirm the password.	When the User is sure to proceed with the setting of a new password.	None	Selected by the user
		[ALT] Cancels Password reset process	This contains the details that the user selects in order to cancel the password reset process.	When the User is sure to cancel the setting of a new password.	None	Selected by the user
		Enter OTP details	This contains information that allows the OTP details to be entered to the system.	When the user wants to enter details of their OTP onto the system	None	Entered by the User
		[ALT] Request new OTP	This contains the information that	This is when the user requests to receive a new OTP.	None	Requested by the User



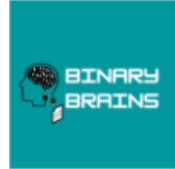
GROUP 23 DELIVERABLE 02



			helps users get a new OTP			

Requirement Number:	Requirement Title:	Description:	Purpose:	Usage:	Entities & Attributes:	Logical Layout:
4.5	Update User	Request to update user details	This contains the information required for a user to update their details on the system	This is when the user wants to update their personal details on the system.	None	Requested by the User
		Enter details.	This contains the details that the user enters into the system to be updated	Used when the user enters their personal details on the system so that their profile can be updated.	None	Entered by the User
		Confirm update process	This contains the details that the user selects in order to confirm the update.	When the User is sure to proceed with the update process.	None	Selected by the user
		[ALT] Cancels Update process	This contains the details that the user selects in order to cancel the update process.	When the User is sure to cancel the update.	None	Selected by the user

Requirement Number:	Requirement Title:	Description:	Purpose:	Usage:	Entities & Attributes:	Logical Layout:
4.6	View software Guide	Request to be view software guide	This contains the information required for a user to view software guide	This is when the user wants to view software guide	None	Requested by the User

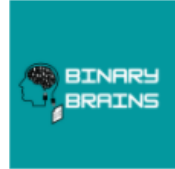


GROUP 23 DELIVERABLE 02



		Click on Link	This contains the details of the selection of the link in order to view software guide	Used when the user selects the software guide link to view it.	None	Selected by the User
		View software guide	This contains information that allows the user to get details of the software guise	When the link has successfully been clicked on and the user views software guide.	None	Viewed by the User

Requirement Number:	Requirement Title:	Description:	Purpose:	Usage:	Entities & Attributes:	Logical Layout:
4.7	View Notification	Request to be view all notifications	This contains the information required for a user to view all notification	This is when the user wants to view all their notifications	None	Requested by the User
		Click on notification	This contains the details of the selection of the notification to be viewed.	Used when the user selects the notification they want to view	None	Selected by the User

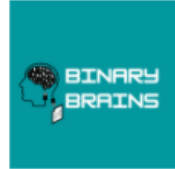


GROUP 23 DELIVERABLE 02



Requirement Number:	Requirement Title:	Description:	Purpose:	Usage:	Entities & Attributes:	Logical Layout:
4.8	Generate Frame information	Request to view frame information	This contains the information required for a user to view the frame information	This is when the user wants to view the kind of frames available	None	Requested by the User
		View frame information	This contains information that allows the user to get details of the frames	When the user is able to successfully view the frame information	None	Viewed by the User

Requirement Number:	Requirement Title:	Description:	Purpose:	Usage:	Entities & Attributes:	Logical Layout:
5.1	Apply for exhibition	Request to apply for exhibition	This contains the information required for an artist to apply for exhibition.	This is when the artist wants to apply for an exhibition	None	Requested by the Artist
		Enter exhibition details.	This contains the details that the artist enters into the system in order to apply for exhibition	Used when the artist enters their details on the system so that they can be selected to be part of an exhibition	None	Entered by the Artist
		Enter tag details.	This contains the tag details that the artist enters into the system in order to apply for exhibition	Used when the artist enters their artwork details on the tags	None	Entered by the Artist

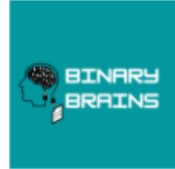


GROUP 23 DELIVERABLE 02



Requirement Number:	Requirement Title:	Description:	Purpose:	Usage:	Entities & Attributes:	Logical Layout:
5.2	Apply to teach class	Request to be teach class	This contains the information required for an art teacher to apply to teach a class	This is when an art teacher wants to teach an art class	None	Requested by the user
		Enter details.	This contains the details that the user enters into the system in order to apply as a class Teacher	Used when the user enters their personal details on the system so that they can apply to teach a class.	None	Entered by the user

Requirement Number:	Requirement Title:	Description:	Purpose:	Usage:	Entities & Attributes:	Logical Layout:
5.3	Add Picture	Request to add picture on the system	This contains the information required for an artist to add pictures on their profile	This is when the artist wants to upload pictures onto the system	None	Requested by the Artist
		Submit picture	This contains details that allows the artist to submit all the pictures that they wish to upload	When the Artist selects pictures to upload and wants to put it onto the system.	None	Submitted by the Artist
		Confirm picture upload process	This contains the details that the user selects in order to confirm the upload.	When the User is sure to proceed with the uploading of the pictures	None	Selected by the artist

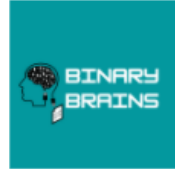


GROUP 23 DELIVERABLE 02



Requirement Number:	Requirement Title:	Description:	Purpose:	Usage:	Entities & Attributes:	Logical Layout:
5.4	Update Picture	Request to update picture on the system	This contains the information required for an artist to update pictures on their profile	This is when the artist wants to upload updated pictures onto the system	None	Requested by the Artist
		Select and Submit picture	This contains details that allows the artist to select and submit all the pictures that they wish to upload	When the Artist selects pictures to upload and wants to put it onto the system.	None	Submitted by the Artist
		Confirm picture upload process	This contains the details that the user selects in order to confirm the upload of new pictures	When the artist is sure to proceed with the uploading of the pictures	None	Selected by the artist.

Requirement Number:	Requirement Title:	Description:	Purpose:	Usage:	Entities & Attributes:	Logical Layout:
5.5	Delete Picture	Request to delete picture	This contains the information required to delete the pictures from the system	This is when the artist wants to delete pictures the system.	None	Requested by the Artist
		Select picture to delete	This contains the selection information required to delete the pictures from the system	This is when the artist selects to delete a specific picture from the system	None	Selected by the Artist



GROUP 23 DELIVERABLE 02



		Click Delete	This contains information of the artist selecting the delete process.	When the artist is sure of which picture they want to delete and unload from the system	None	Clicked by the Artist
		Confirm deletion process	This contains information of the confirmation of the deletion process.	When the artist confirms that they want to delete the picture.	None	Confirmed by the Artist

Requirement Number:	Requirement Title:	Description:	Purpose:	Usage:	Entities & Attributes:	Logical Layout:
5.6	View Invitation	Request to be view all invitations	This contains the information required for an artist to view all invitations	This is when the artist wants to view all their invitations	None	Requested by the Artist
		Add event	This contains the details of the event that the artist wants to attend to be added to their calendar	Used when the artist wants to attend an event	None	Selected by the Artist
		Dismiss invitations	This contains information that allows the artist to dismiss the invitations	When the artist has viewed the invitation.	None	Selected by the Artist

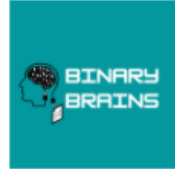
Requirement Number:	Requirement Title:	Description:	Purpose:	Usage:	Entities & Attributes:	Logical Layout:

5.7	View existing exhibitions	Request to be view exhibition information	This contains the information required for an artist or a user to view exhibition information	This is when the user or artist wants to view exhibitions.	None	Requested by the Artist, User
		View Exhibition.	This contains the details that the user or artist enters to be able to view the exhibition	Used when the artist or user selects the exhibition to view	None	Selected by the Artist, User

Requirement Number:	Requirement Title:	Description:	Purpose:	Usage:	Entities & Attributes:	Logical Layout:
5.8	Accept invitation for exhibition	Request to be accept invitation	This contains the information required for an artist to accept the invitation	This is when the artist has been invited to an exhibition	None	Requested by the Artist
		Accept invitation	This contains the details where the artist has accepted the invitation	Used when the artist accepts invitation to an exhibition.	None	Entered by the Artist

Conclusion

In conclusion, the above section contains the in-depth descriptions of all the different reports and other outputs of the proposed system as depicted in the context diagram and the data flow diagrams

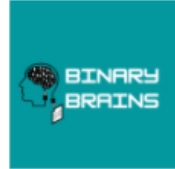


7. Reports & Other Outputs

Introduction

This section of the deliverable contains an in-depth description of the different reports and other outputs for the proposed system.

7.1 Output Description



GROUP 23 DELIVERABLE 02



Requirement Number:	Requirement Title:	Description:	Purpose:	Usage:	Entities & Attributes:	Logical Layout:
1.1	Approve exhibition User	1.1 The Curator clicks to view the Applications for Exhibitions	This contains the information that is related to approving a user who applies for an exhibition	This is used when the curator views an exhibition application that has just been submitted to her	None	Clicked by the Curator
		1.1 The Curator clicks on application to view the details.	This contains all the details related to the applications for exhibition that the Curator wants to view	This is used when the curator views exhibition application details	None	Viewed by the curator
		1.1 The Curator reads the application and clicks on Approve Application	This used when the curator goes through an application and reads it through and approve the application for the artist to exhibit.	This is used when the curator approves exhibitions applications after reading them.	None	Read and approved by the curator

		1.1 ALT Step 6: The Curator reads the application and clicks on Deny Application.	This is used when the curator still reads through the application bit then denies the application	This is used when the Curator denies exhibition applications after reading them.	None	Read and disapproved by the curator
Requirement Number:	Requirement Title:	Description:	Purpose:	Usage:	Entities & Attributes:	Logical Layout:
1.1	Approve exhibition User	1.1 Load and show table with applications	Contains a table with application grouped by Exhibition	Used a curator wants to approve exhibition application	Application table: - AdminID - UserRoleID - DateofBirth	Approved by the Curator
		1.1 Open a screen with all the details with an option to approve	Contains all the application details displayed on a screen	When a Curator wants to approve an exhibition application, all the datils are presented on a screen for her to view	None	Displayed to the System
		1.1 Confirm and validate	Contains confirmation and validation information for the Curator to see	When the curator wants to add the new user, she is asked to confirm and	None	Displayed to the System

				validate her action		
		1.1 Add and notify Artist to exhibition	Contains information regarding the adding of artist to the exhibition as well as notifying them	When the Curator adds an artist to an exhibition, he/she is notified of their approval by being requested to fill in a Tag	Tag table: • TagID • ArtworkID • ArtworkTitle • ArtistID • MediumTypeID • ArtworkPicture • ArtworkPrice	Approved by Curator after Artist fills in tag.
		1.1 ALT Step 7: Notify artist that application is denied.	Contains info informing the artist that their application has been denied.	When an artist application has been denied, a notification message is sent to them.	None	Sent by the Curator
Requirement Number:	Requirement Title:	Description:	Purpose:	Usage:	Entities & Attributes:	Logical Layout:
1.2	Search Exhibition User	1.2 Load ExhibitionArtists table and display their details	Contains information provided from searching for the artist user from the artist table	This is used to search for the exhibition user	ExhibitionArtists: • ArtistID • ExhibitionID	Provided for by the Curator

		1.2 Match search results data with related attributes	Contains all the data that matches with the search results data found.	Used when the curator wants to match the searched fields	None	Displayed to the System
		1.2 Return all the matched data in table format	To view the data, it is returned to the curator in table format	Used when the search results have been returned and the curator views the data in table format.	None	Displayed to the System
		1.2 ALT STEP:4 No results matching the inserted details were found	Contains information relating to messaging indicating that no search results were found	Used when no results corresponding to the provided details are found	None	Displayed to the System
Requirement Number:	Requirement Title:	Description:	Purpose:	Usage:	Entities & Attributes:	Logical Layout:
1.3	Delete Exhibition User	1.3 Locate the specific user	Contains the information about the user that the curator wants deleted	Used when the curator wants to delete the user from the system	ExhibitionArtists: - ArtistID - ExhibitionID	Initiated by the Curator
		1.3 Match search results data with related attributes	Contains all the data that matches with the search	Used when the curator wants to match the searched fields	None	Displayed to the System

			results data found.			
		1.3 Prompt user to confirm whether are sure they should be deleted	Contains information that confirms whether the user wants to be deleted or not.	Used when there is a need to verify whether the artist is certain that they should be deleted.	None	Displayed by the system to the User.
		1.3 System deleted the user from the system	Contains information about the deleted Exhibition user.	Used when the user has been removed from the system.	None	Displayed by the system
		1.3 Pop up message notifying the user that they have been successfully deleted	Contains information about the deleted user.	Used to notify user that they have been successfully removed from the system.	None	Displayed to the user by the system
		1.3 ALT Step7: Curator denies confirmation and the use case ends	Contains information that relates to the Curator denying confirmation	Used when deletion confirmation has been denied	None	Provided by the Curator and displayed by the system.
Requirement Number:	Requirement Title:	Description:	Purpose:	Usage:	Entities & Attributes:	Logical Layout:
1.4	Create new Exhibition	1.4 Loads form with appropriate input fields and prompts user to	Contains information about new exhibition creation details.	Used when a new exhibition needs to be made	None	Initiated by the Curator and displayed by the system

		enter his/her details				
		1.4 Save the details to the exhibition table	Contains the saved details of the exhibition creation.	Used when the system saves the new exhibition creation details.	Exhibition table: - ExhibitionID - ExhibitionName - ExhibitionDescription - ExhibitionDate - ExhibitionTime - ExhibitionAnnouncementID - VenueID Schedule table: - ScheduleID - ScheduleName - ScheduleDescription - ScheduleTypeID	Captured and saved by the system
		1.4 Display message for successful exhibition creation and send an announcement to users	Contains information about the successful creation of exhibition	Used to inform the Curator that a new exhibition has been successfully created	None	Displayed by the system.
		1.4 ALT Step4: Invalid characters or details	Contains information about the invalidity of details or characters	Used to inform the Curator that invalid details have been provided	None	Displayed by the system

		1.4 ALT step6: Data not correctly saved	Contains information about the unsuccessful saving of data	Used when a message displaying that details have not been saved	None	Displayed by the system
Require ment Number:	Requirement Title:	Description:	Purpose:	Usage:	Entities & Attributes:	Logical Layout:
1.5	Search exhibition	1.5 Prompt user to enter search details	Contains information about the details relating to the exhibition parameters	Used when the curator wants to search for an exhibition	None	Provided by the Curator
		1.5 Match fields and validate details	Contains all the information about the matching attributes	Used to match details and validating them	Exhibition table: - ExhibitionID - ExhibitionName - ExhibitionDescription - ExhibitionDate - ExhibitionTime - ExhibitionAnnoucementID - VenueID	Displayed buy the system.
		1.5 Return all the matched data	Contains information about all the matching data attributes	Used gather all the data retrieved from the search	None	Displayed by the System

				exhibition command		
		1.5 ALT Step5: No data is returned	Contains information about the failure to return the data from the search command	Used when no data has been returned after search commands details have been entered.	None	Displayed by system
Requirement Number:	Requirement Title:	Description:	Purpose:	Usage:	Entities & Attributes:	Logical Layout:
1.6	Update Exhibition	1.6 Initiate by Searching exhibition	Search exhibition to update it	Used to identify the exhibition to delete.	None	Displayed by the system
		1.6 Load exhibition Information	Contains information relating to the loaded exhibition	Used to select the parameters to update.	None	Loaded by the system
		1.6 Prompt user to update exhibition details	Contains update information for the user to confirm	Used to confirm whether the user wants to update their details	None	Provided by the user
		1.6 Update details in the user table	Contains information about the	Used to store updated info	Exhibition table: - ExhibitionID - ExhibitionName	Updated by the system

			updated details		- ExhibitionDescription - ExhibitionDate - ExhibitionTime - ExhibitionAnnouncementID - VenueID	
		1.6 ALT Step6: Invalid details	Contains invalid details info	Used when the system displays that the details are invalid	None	Displayed by the System
Requirement Number:	Requirement Title:	Description:	Purpose:	Usage:	Entities & Attributes:	Logical Layout:
1.7	Delete Exhibition	1.7 Prompt confirmation of the deletion of Exhibition.	This contains the details gathered to delete an exhibition.	It is used when the system needs to receive confirmation to proceed with the deletion of an exhibition.	None	Prompted by the system.
		1.7 Notification of the successful deletion of	This contains the information used to indicate	This is used when an exhibition is deleted from the	None	Prompt displayed by system.

		an exhibition.	whether an exhibition has been deleted successfully.	system successfully.		
		1.7 ALT: Notification of the exhibition being unsuccessfully deleted.	This contains information to inform the curator that the exhibition has been deleted unsuccessfully.	This is used when the operation to delete an exhibition is unsuccessful.	None	Prompt displayed by the system.
Requirement Number:	Requirement Title:	Description :	Purpose:	Usage:	Entities & Attributes:	Logical Layout:
1.8	View Artist	1.8 Prompt the entry of search parameters .	This process prompts the curator to provide the search parameters of artist details.	This is used when the curator wants to view a specific artist.	None	Prompt displayed by the system.

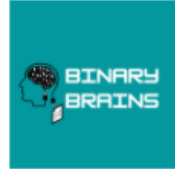
		1.8 [ALT] Validation error	Identifies and notifies the curator that the entered search parameters are invalid.	This occurs when a curator enters invalid search parameters .	None	Displayed by the system.
		1.8 Display search results details	The process that displays the search results requested by the curator.	This is used when the system displays matching responses to the search.	None	Displayed by the system.
		1.8 [ALT]No Artist found results	This process is used to inform the curator that the artist they searched to view for was not found.	This is used to show the curator the artist that matches with entered search parameters .	None	Displayed by the system.
		1.8 Selected Artist	This consists of the art artist details	This is used when the curator	- Artist table: - ArtistID - ArtistName	Displayed by the system.

		information	that was selected.	selects an artist to view.	• ArtistSurname • ArtistType • Artist Bio • ArtistUserRole • ArtistPicture1	
					1.	
Requirement Number:	Requirement Title:	Description :	Purpose:	Usage:	Entities & Attributes:	Logical Layout:
1.9	Add Artwork	1.9 Prompt art class teacher to provide artwork details information	This contains the details required to prompt the curator to provide artwork details.	This is used when the curator requests for an artwork to be added to the system.	• None	Prompted by the curator.
		1.9 Request confirmation of artwork detail's information	This contains the details used to confirm the creation of the Artwork within the system.	This is used when the curator confirms artwork details	• None	Prompted by the curator.
		1.9 Notification of the successful	This contains the information to notify the successful	This is used when the artwork has been successfully	• None	Provided by the curator.

		creation information	creation of an artwork.	created in the system.		
		1.9 [ALT]Notification of the unsuccessful creation information	Contains the information to notify the curator of the unsuccessful creation.	This is used when the artwork is unsuccessful.	None	Provided by the curator
Requirement Number:	Requirement Title:	Description :	Purpose:	Usage:	Entities & Attributes:	Logical Layout:
1.10	Search Class Artwork	1.10 Prompt the entry of search parameters .	This process prompts the curator to provide the search parameters of artwork details.	This is used when the curator wants to search for an artwork.	None	Prompt displayed by the system.
		1.10 [ALT] Validation error	Identifies and notifies the curator that the entered search parameters are invalid.	This occurs when a curator enters invalid search parameters .	None	Displayed by the system.

		1.10 Display search results details	The process that displays the search results requested by the curator.	This is used when the system displays matching responses to the search.	None	Displayed by the system.
		1.10 [ALT]No Artwork found results	This process is used to inform the curator that the artwork they searched for was not found.	This is used to show the curator the artwork that matches with entered search parameters .	None	Displayed by the system.
		1.10 Selected Artwork informatio n	This consists of the artwork details that was selected.	This is used when the curator selects an artwork	• Artist table: • ArtistID • ArtistName • ArtistSurname • ArtistType • Artist Bio • ArtistUserRole • ArtistPicture1	Displayed by the system.
Requirement Number:	Requirement Title:	Description:	Purpose:	Usage:	Entities & Attributes:	Logical Layout:

1.11	Delete Artwork	1.11 Prompt confirmation of the deletion of Artwork.	This contains the details gathered to delete an artwork.	It is used when the system needs to receive confirmation to proceed with the deletion of an artwork.	None	Prompted by the system.
		1.11 Notification of the successful deletion of an artwork.	This contains the information used to indicate whether an artwork has been deleted successfully.	This is used when an artwork is deleted from the system successfully.	None	Prompt displayed by system.
		1.11 ALT: Notification of the artwork being unsuccessfully deleted.	This contains information to inform the curator that the artwork has been deleted.	This is used when the operation to delete an artwork is unsuccessful.	None	Prompt displayed by the system.



GROUP 23 DELIVERABLE 02

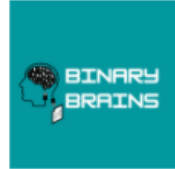


			unsuccessfully.			
Requirement Number:	Requirement title:	Description:	Purpose:	When Used:	Entities & Attributes:	Logical Layout:
1.12 Update Artwork Output						
1.12	Update Artwork	1.12 Request artwork info	This contains the information to prompt the curator for new artwork details	This is used when the curator wants to make changes on existing artwork details	None	Prompted by the system.
		1.12 Request confirmation artwork details info	This contains the information to confirm the updated details made by the curator.	This is used when the curator entered all the new details of the artwork.	None	Generated by the system.
		1.12 [ALT] Update Not confirmed artwork info	This contains the confirmation details when the curator did not	This is used when the curator did not confirm to update	None	Generated by the system.

			confirm to update the artwork details.	the artwork details.		
		1.12 Updated artwork info	This contains the information to notify the curator that the artwork details have been updated.	This is used when the curator confirmed the updated details, and the update was successful.	None	Generated by the system.
		1.12 Update announcement artwork info	This contains the information to notify the artist that an artworks details have been changed.	This is used when the curator successfully changed the artwork details and notifies all artists about the changes	None	Generated by the system.
1.13 Create Artist Exhibition Invitation Output						
1.13	Create Artist Exhibition Invitation	1.13 Create Exhibition Invitation form info	This contains the information to prompt the curator to enter all	This is used by the curator when the curator wants to	None	Prompted by the System.

			invitation details.	create the invitation for the artists		
		1.13 [ALT] Invalid Exhibition Invitation Creation info	This contains the information to notify the curator that the invitation is invalid	This is used when the information entered the invitation is invalid	None	Generated by the system.
		1.13 [ALT] Save Unsuccessful Exhibition Invitation info	This contains the information to notify the curator that the creation of the invitation was unsuccessful	This is used when the curator saves the invitation, and the process was unsuccessful.	None	Generated by the system.
		1.13 Save successful Exhibition Invitation info	This contains the information to notify the curator that the creation of the invitation	This is used when the curator saves the invitation, and the process was successful	None	Generated by the system.

			was successful.			
		1.13 Artist Invitation sent info	This contains the information to notify the curator that the invitation was sent to the artists.	This is used when the curator sent the invitation the artists.	None	Generated by the system.
Requirement Number:	Requirement Title:	Description:	Purpose:	Usage:	Entities & Attributes:	Logical Layout:
1.14	Print Tags	Retrieve, Compile and Prompt user for Printer Preference.	This contains the details of retrieving, compiling and prompting user for details.	This is when the system results in the requests made by the curator and prompts the curator to enter details.	Tag Table: - Tag ID - Artwork title - Medium Name - Artist ID - Art Picture	Retrieved, compiled and displayed by the system
		[ALT] Display error message showing no tags available	Consists of the information required to print tags for an exhibition.	This is used when the system processes that no tag information is available to be printed.	None	Displayed by the System.
		Print Tags	This contains the details of the information needed to print the tags in a PDF format.	Used when the system sends a print command to the chosen printer system. The tags are printed in PDF format.	None	Displayed by the System.
		Display message of	This contains the details of the confirmation of the	Used when the system was able to print all the tags needed.	None	Displayed by the system

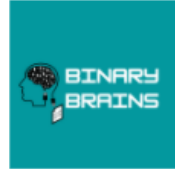


GROUP 23 DELIVERABLE 02



		successful printing.	successful printing of tags			
Requirement Number:	Requirement Title:	Description:	Purpose:	Usage:	Entities & Attributes:	Logical Layout:
1.15	View Schedule	Display Schedule.	This contains the details needed to show the schedule in a calendar format.	This is when the system processes the query using the enters details about the schedule to show the details in a calendar format.	Schedule Table - Schedule ID - Schedule Name - Schedule Description - Schedule Type ID	Displayed by the system
Requirement Number:	Requirement Title:	Description:	Purpose:	Usage:	Entities & Attributes:	Logical Layout:
1.16	Print Schedule	Display Schedule.	This contains the details needed to show the schedule in a calendar format and prompt user for printer preference	This is when the system processes the query using the enters details about the schedule to show the details in a calendar format and requires printer preference details.	None.	Displayed by the system
		Print Schedule	This contains the details of the information needed to print the schedule in a PDF format.	Used when the system sends a print command to the chosen printer system. The schedule within a time range is printed in PDF format.	Schedule Table - Schedule ID - Schedule Name - Schedule Description - Schedule Type ID	Displayed by the System.
		Display message of successful printing.	This contains the details of the confirmation of the successful printing of schedule.	Used when the system was able to print the schedule needed.	None	Displayed by the system

Requirement Number:	Requirement Title:	Description:	Purpose:	Usage:	Entities & Attributes:	Logical Layout:
1.17	Create Class	Prompt user to Enter Details.	This contains the details needed to create a new class.	This is when the system displays the information needed to create a new class.	Art Class Table • Art Class ID • Art Class Name • Art Class Description • Art Class Start Date • Art Class End Date • Class Price • Art Class Type ID • Class Teacher • Art Class Type • Price Organisation ID • Venue ID • Art Class Announce ID	Displayed by the system
		Save Art Class Details.	This contains details of the art class that need to be saved by the system.	To save the details of an art class in order to create it.	None.	Saved and Validated by the system
		[ALT] Error in Saving Art Class Details	Consists of the information to show error message of unsuccessful creation of art class.	This is used when the system captures that there is an error in the information provided by the curator.	None	Displayed by the System.
		Show message of successful	Contains information showing the curator the art class has been saved and created,	This is when the creation of an art class is successful.	None	Displayed by the system



GROUP 23 DELIVERABLE 02



		creation of Art Class	with an announcement sent to all the users.			
Requirement Number:	Requirement Title:	Description:	Purpose:	Usage:	Entities & Attributes:	Logical Layout:
1.18	Search Class	1.18 Prompt the entry of search parameters.	This process prompts the curator to provide the search parameters of class details.	This is used when the curator wants to search for a class.	None	Prompt displayed by the system.
		1.18 [ALT] Validation error	Identifies and notifies the curator that the entered search parameters are invalid.	This occurs when a curator enters invalid search parameters.	None	Displayed by the system.
		1.18 Display search results details	The process that displays the search results requested by the curator.	This is used when the system displays matching responses to the search.	None	Displayed by the system.
		1.18 [ALT]No Art Class found results	This process is used to inform the curator that the class they searched for	This is used to show the curator the art class that matches with entered	None	Displayed by the system.

			was not found.	search parameters.		
		1.18 Selected Art Class information	This consists of the art class details that was selected.	This is used when the curator selects a art class to view.	Art_Class table: • ArtClassID • ArtClassName • ArtClassStartDate • ArtClassEndDate	Displayed by the system.
Requirement Number:	Requirement Title:	Description:	Purpose:	Usage:	Entities & Attributes:	Logical Layout:
1.19	Update Class	1.19 Request Class updated information	Contains the request to acquire Class information.	It is used when the curator requests to update the class information	• None	Displayed by the system.
		1.19 Prompt confirmation of changes	This contains the information to notify the curator of the successful class update.	This is used when the system receives confirmation to update the class.	• None	Prompted by the system.
		1.19 Successfully	This contains the information	This is used when the class has	• None	Prompted by the system.

		changed notification	to notify the curator of the successful update of a class.	been successfully updated in the system.		
		1.19 [ALT] Unsuccessful update of Class information	Contains the unsuccessful update notification	When the system failed to update the Class information	None	Displayed by the system.
		1.19 [ALT] Incorrect format or empty field	Contains the details that the format is incorrect, or a field is empty	When the provided information fails the validation check	None	Displayed by the system.
Requirement Number:	Requirement Title:	Description:	Purpose:	Usage:	Entities & Attributes:	Logical Layout:
1.20	Delete Class	1.20 Prompt confirmation of the deletion of Art Class.	This contains the details gathered to delete an art class.	It is used when the system needs to receive confirmation to proceed with the	None	Prompted by the system.

				deletion of an art class.		
		1.20 Notification of the successful deletion of an art class.	This contains the information used to indicate whether an art class has been deleted successfully.	This is used when an art class is deleted from the system successfully.	None	Prompt displayed by system.
		1.20 ALT: Notification of the art class being unsuccessfully deleted.	This contains information to inform the curator that the art class has been deleted unsuccessfully.	This is used when the operation to delete an art class teacher is unsuccessful.	None	Prompt displayed by the system.
Requirement Number:	Requirement Title:	Description:	Purpose:	Usage:	Entities & Attributes:	Logical Layout:
1.21	Generate Class Reminder	1.21 Request booking info	To request the search art class invoke	This happens when time triggers the	None	Request made by the system.

			information to generate a report.	system to generate a report.		
		1.21 Request User details	To request from a search user invoke, the user information to send the report to.	When the report has been aggregated.	None	Request made by the system.
		1.21 Class Reminder information	To send the report to the user.	Once the report has been aggregated and contact information is received.	None	Sent by the system
Requirement Number:	Requirement Title:	Description:	Purpose:	Usage:	Entities & Attributes:	Logical Layout:
1.22	Generate Payment Reminder	1.22 Request art class info	To request the search book art class, invoke information to generate a report.	This happens when time triggers the system to generate a report.	None	Request made by the system.

		1.22 Request User details	To request from a search user invoke, the user information to send the report to.	When the report has been aggregated.	None	Request made by the system.
		1.22 Payment Reminder information	To send the report to the user.	Once the report has been aggregated and contact information is received.	None	Sent by the system
Requirement Number:	Requirement Title:	Description :	Purpose:	Usage:	Entities & Attributes:	Logical Layout:
1.23	Create Art Class Teacher	1.23 Prompt art class teacher to provide his/her personal details information	This contains the details required to prompt the Art Class teacher to provide his/her personal details.	This is used when the art class teacher requests to be added to the system.	None	Prompted by the curator.

		1.23 5 Request confirmatio n of their personal details information	This contains the details used to confirm the creation of the Art Class teacher within the system.	This is used when the system administrat or needs to receive confirmatio n from the Art class teacher to proceed.	None	Prompted by the curator.
		1.23 Notification of the successful creation information	This contains the information to notify the art class teacher of the successful	This is used when the art class teacher has been successfully created in the system.	None	Provided by the curator.
		1.23 [ALT]Notific ation of the unsuccessf ul creation information	Contains the information to notify the art class teacher of the unsuccessful creation.	This is used when the art class teacher creation is unsuccessfu l	None	Provided by the curator
Requirem ent Number:	Requireme nt Title:	Description :	Purpose:	Usage:	Entities & Attributes:	Logical Layout:

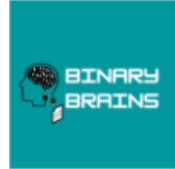
1.24	Search Class Teacher	1.24 Prompt the entry of search parameters .	This process prompts the curator to provide the search parameters of class teacher details.	This is used when the curator wants to search for a class teacher.	None	Prompt displayed by the system.
		1.24 [ALT] Validation error	Identifies and notifies the curator that the entered search parameters are invalid.	This occurs when a curator enters invalid search parameters .	None	Displayed by the system.
		1.24 Display search results details	The process that displays the search results requested by the curator.	This is used when the system displays matching responses to the search.	None	Displayed by the system.
		1.24 [ALT]No Art Class teacher	This process is used to inform the curator that the class	This is used to show the curator the art class	None	Displayed by the system.

		found results	teacher they searched for was not found.	teacher that matches with entered search parameters .		
		1.24 Selected Art Class Teacher information	This consists of the art class teacher details that was selected.	This is used when the curator selects an art class teacher to view.	ClassTeacher table: • ClassTeacher_Name • ClassTeacher_Surname • ClassTeacher_EmailAddress • ClassTeacher_Medium • ClassTeacher_AddressLine1 • ClassTeacher_AddressLine2	Displayed by the system.
Requirement Number:	Requirement Title:	Description:	Purpose:	Usage:	Entities & Attributes:	Logical Layout:
1.25	Update Class Teacher	1.25 Request Class Teacher updated information	Contains the request to acquire Class teacher information.	It is used when the curator requests to update the class teacher information	• None	Displayed by the system.

		1.25 Prompt confirmation of changes	This contains the information to notify the curator of the successful class teacher update.	This is used when the system receives confirmation to update the class teacher.	None	Prompted by the system.
		1.25 Successfully changed notification	This contains the information to notify the curator of the successful update of a class teacher.	This is used when the class has been successfully updated in the system.	None	Prompted by the system.
		1.25 [ALT] Unsuccessful update of Class Teacher information	Contains the unsuccessful update notification	When the system failed to update the Class information	None	Displayed by the system.
		1.25 [ALT] Incorrect format or empty field	Contains the details that the format is	When the provided information fails the	None	Displayed by the system.

			incorrect, or a field is empty	validation check		
Requirement Number:	Requirement Title:	Description:	Purpose:	Usage:	Entities & Attributes:	Logical Layout:
1.26	Delete Class Teacher	1.26 Prompt confirmation of the deletion of Art Class teacher.	This contains the details gathered to delete an art class teacher.	It is used when the system needs to receive confirmation to proceed with the deletion of an art class teacher.	None	Prompted by the system.
		1.26 Notification of the successful deletion of an art class teacher.	This contains the information used to indicate whether an art class teacher has been deleted successfully.	This is used when an art class teacher is deleted from the system successfully.	None	Prompt displayed by system.

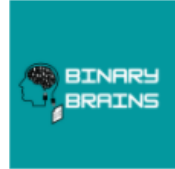
		1.20 ALT: Notification of the art class teacher being unsuccessfully deleted.	This contains information to inform the curator that the art class teacher has been deleted unsuccessfully.	This is used when the operation to delete an art class teacher is unsuccessful.	None	Prompt displayed by the system.
Requirement Number:	Requirement Title:	Description:	Purpose:	Usage:	Entities & Attributes:	Logical Layout:
1.27	Generate Booking Information	Display Booking Notification Page	This contains the details of the request of being able to generate a booking.	This is when the system displays the Booking Notification Page.	None	Displayed by the system
		[ALT] Validation Error encountered	Consists of the information required to validate the booking notification details provided.	This is used when the system captures that there is an error in the information provided by the curator.	None	Validated by the System.
		Display confirmation Message of information captured.	This contains the details of the confirmation of all the details being captured successfully.	Used when the system saves all the information being successfully captured into the system.	None	Displayed by the System.
		[ALT] Display Error Message	This contains the details of the confirmation of the	Used when the system was not able to save all the information which	None	Displayed by the system



GROUP 23 DELIVERABLE 02



		of Information not captured.	unsuccessful capture of information.	was supposed to be captured into the system.		
		Show confirmation message of successfully Creating Booking notification	Contains information to notify the curator that the booking notification is successfully created.	This is when the booking notification creation is successful.	None	Displayed by the system
		[ALT] Show confirmation message of unsuccessful creation of booking notification.	Contains information to notify the curator that the booking notification creation is unsuccessful.	This is when the booking notification creation is unsuccessful.	None	Displayed by the system
Requirement Number:	Requirement Title:	Description:	Purpose:	Usage:	Entities & Attributes:	Logical Layout:
1.28	Create Medium Detail	Display Create Medium Page	This contains the details of the request of being able to create a medium.	This is when the system displays the Create medium Page.	None	Displayed by the system
		[ALT] Validation Error encountered	Consists of the information required to validate the medium details provided.	This is used when the system captures that there is an error in the information provided by the curator.	None	Validated by the System.
		Display confirmation Message of	This contains the details of the confirmation of all the	Used when the system saves all the information being	None	Displayed by the System.

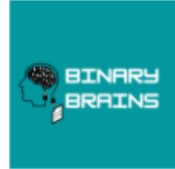


GROUP 23 DELIVERABLE 02



		information captured.	details being captured successfully.	successfully captured into the system.		
		[ALT] Display Error Message of Information not captured.	This contains the details of the confirmation of the unsuccessful capture of information.	Used when the system was not able to save all the information which was supposed to be captured into the system.	None	Displayed by the system
		Create and display new medium details.	Contains information showing the curator the new medium details created.	This is when the medium creation is successful.	None	Displayed by the system
		[ALT] Show error message of unsuccessful creation of new medium details.	Contains information to notify the curator that the medium creation is unsuccessful.	This is when the medium creation is unsuccessful.	None	Displayed by the system
Requirement Number:	Requirement Title:	Description:	Purpose:	Usage:	Entities & Attributes:	Logical Layout:
1.29	Search Medium Detail	Display Search Medium Page	This contains the details of the request of being able to search for a medium.	This is when the system displays the Search medium Page.	None	Displayed by the system
		[ALT] Validation Error encountered	Consists of the information required to validate the medium details provided.	This is used when the system captures that there is an error in the information provided by the curator.	None	Validated by the System.
		Display confirmation	This contains the details of the	Used when the system saves all the	None	Displayed by the System.

		Message of information captured.	confirmation of all the details being captured successfully.	information being successfully captured into the system.		
		[ALT] Display Error Message of Information not captured.	This contains the details of the confirmation of the unsuccessful capture of information.	Used when the system was not able to save all the information which was supposed to be captured into the system.	None	Displayed by the system
		Search and display selected medium details.	Contains information showing the curator the medium details searched for.	This is when the medium search is successful.	None	Displayed by the system
		[ALT] Show error message of unsuccessful search of medium details.	Contains information to notify the curator that the medium search is unsuccessful.	This is when the medium search is unsuccessful.	None	Displayed by the system
Requirement Number:	Requirement Title:	Description:	Purpose:	Usage:	Entities & Attributes:	Logical Layout:
1.30	Update Medium Detail	Display Update Medium Page	This contains the details of the search request of being able to update a medium.	This is when the system displays the Update medium Page.	None	Displayed by the system
		Display medium detail	This contains details of the specific medium that has been selected to be updated.	To update a specific medium detail.	Medium Type Table: - Medium Type ID - Medium Type Description	Displayed by the system



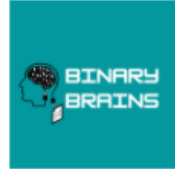
GROUP 23 DELIVERABLE 02



		[ALT] Validation Error encountered	Consists of the information required to validate the medium details provided.	This is used when the system captures that there is an error in the information provided by the curator.	None	Validated by the System.
		Show message of medium details being successfully updated	Contains information showing the curator the medium details are updated.	This is when the medium is updated successfully	None	Displayed by the system
		[ALT] Show error message of unsuccessful update of medium details.	Contains information to notify the curator that the medium update is unsuccessful.	This is when the medium update is unsuccessful.	None	Displayed by the system
Requirement Number:	Requirement Title:	Description:	Purpose:	Usage:	Entities & Attributes:	Logical Layout:
1.31	Delete Medium	Display Delete Medium Page	This contains the details of the search request of being able to delete a medium.	This is when the system displays the Delete medium Page.	None	Displayed by the system
		Display medium detail	This contains details of the specific medium that has been selected to be deleted.	To delete a specific medium.	Medium Type Table: • Medium Type ID • Medium Type Description	Displayed by the system

		Show message of medium details being successfully deleted	Contains information showing the curator the medium details are deleted.	This is when the medium is deleted successfully	None	Displayed by the system
		[ALT] Show error message of unsuccessful deletion of medium details.	Contains information to notify the curator that the medium deletion is unsuccessful.	This is when the medium deletion is unsuccessful.	None	Displayed by the system
Requirement Number:	Requirement Title:	Description:	Purpose:	Usage:	Entities & Attributes:	Logical Layout:
1.32	Create Artwork Detail	Display Create Artwork Page	This contains the details of the request of being able to create an Artwork.	This is when the system displays the Create Artwork Page.	None	Displayed by the system
		[ALT] Validation Error encountered	Consists of the information required to validate the Artwork details provided.	This is used when the system captures that there is an error in the information provided by the curator.	None	Validated by the System.
		Display confirmation Message of information captured.	This contains the details of the confirmation of all the details being captured successfully.	Used when the system saves all the information being successfully captured into the system.	None	Displayed by the System.
		[ALT] Display Error Message of	This contains the details of the confirmation of the	Used when the system was not able to save all the information which was supposed to be	None	Displayed by the system

		Information not captured.	unsuccessful capture of information.	captured into the system.		
		Create and display new Artwork details.	Contains information showing the curator the new Artwork details created.	This is when the Artwork creation is successful.	None	Displayed by the system
		[ALT] Show error message of unsuccessful creation of new Artwork details.	Contains information to notify the curator that the Artwork creation is unsuccessful.	This is when the Artwork creation is unsuccessful.	None	Displayed by the system
Requirement Number:	Requirement Title:	Description:	Purpose:	Usage:	Entities & Attributes:	Logical Layout:
1.33	Search Artwork Detail	Display Search Artwork Page	This contains the details of the request of being able to search for an Artwork.	This is when the system displays the Search Artwork Page.	None	Displayed by the system
		[ALT] Validation Error encountered	Consists of the information required to validate the Artwork details provided.	This is used when the system captures that there is an error in the information provided by the curator.	None	Validated by the System.
		Display confirmation Message of information captured.	This contains the details of the confirmation of all the details being captured successfully.	Used when the system saves all the information being successfully captured into the system.	None	Displayed by the System.
		[ALT] Display Error Message of	This contains the details of the confirmation of the	Used when the system was not able to save all the information which	None	Displayed by the system



GROUP 23 DELIVERABLE 02

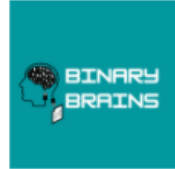


		Information not captured.	unsuccessful capture of information.	was supposed to be captured into the system.		
		Search and display selected Artwork details.	Contains information showing the curator the Artwork details searched for.	This is when the Artwork search is successful.	None	Displayed by the system
		[ALT] Show error message of unsuccessful search of Artwork details.	Contains information to notify the curator that the Artwork search is unsuccessful.	This is when the Artwork search is unsuccessful.	None	Displayed by the system
Requirement Number:	Requirement Title:	Description:	Purpose:	Usage:	Entities & Attributes:	Logical Layout:
1.34	Update artwork Detail	Display Update artwork Page	This contains the details of the search request of being able to update an artwork.	This is when the system displays the Update artwork Page.	None	Displayed by the system
		Display artwork detail	This contains details of the specific artwork that has been selected to be updated.	To update a specific artwork detail.	Artwork table: • Artwork ID • Artwork Title • Price • Art Picture • Artwork Status • Surface Type ID • Medium Type • Exhibition ID • Artist ID	Displayed by the system

					- Frame Colour ID - Frame Dimension ID	
		[ALT] Validation Error encountered	Consists of the information required to validate the artwork details provided.	This is used when the system captures that there is an error in the information provided by the curator.	None	Validated by the System.
		Show message of artwork details being successfully updated	Contains information showing the curator the artwork details are updated.	This is when the artwork is updated successfully	None	Displayed by the system
		[ALT] Show error message of unsuccessful update of artwork details.	Contains information to notify the curator that the artwork update is unsuccessful.	This is when the artwork update is unsuccessful.	None	Displayed by the system
Requirement Number:	Requirement Title:	Description:	Purpose:	Usage:	Entities & Attributes:	Logical Layout:
1.35	Delete artwork	Display Delete artwork Page	This contains the details of the search request of being able to delete a artwork.	This is when the system displays the Delete artwork Page.	None	Displayed by the system
		Display artwork detail	This contains details of the specific artwork that has been selected to be deleted.	To delete a specific artwork.	Artwork table: - Artwork ID - Artwork Title - Price - Art Picture	Displayed by the system

					- Artwork Status - Surface Type ID - Medium Type - Exhibition ID - Artist ID - Frame Colour ID - Frame Dimension ID	
		Show message of artwork details being successfully deleted	Contains information showing the curator the artwork details are deleted.	This is when the artwork is deleted successfully	None	Displayed by the system
		[ALT] Show error message of unsuccessful deletion of artwork details.	Contains information to notify the curator that the artwork deletion is unsuccessful.	This is when the artwork deletion is unsuccessful.	None	Displayed by the system
Requirement Number:	Requirement Title:	Description:	Purpose:	Usage:	Entities & Attributes:	Logical Layout:
1.36	Archive Status	Display Archive Status details	This contains the details of the request of being able to alter the status of an artwork.	This is when the system displays the Archive Status page.	None	Displayed by the system
		Change and display new status.	Contains information showing the curator the status has been changed successfully.	This is when the status has been changed successful.	Artwork Status Table: Artwork Status ID	Displayed by the system

					Artwork Status Description	
		[ALT] Show error message of unsuccessful update of new status.	Contains information to notify the curator that the process of changing status was unsuccessful.	This is when the archive status is not changed.	None	Displayed by the system
Requirement Number:	Requirement Title:	Description:	Purpose:	Usage:	Entities & Attributes:	Logical Layout:
1.37	Create Surface Type Detail	Display Create Surface Type Page	This contains the details of the request of being able to create a Surface Type.	This is when the system displays the Create Surface Type Page.	None	Displayed by the system
		[ALT] Validation Error encountered	Consists of the information required to validate the Surface Type details provided.	This is used when the system captures that there is an error in the information provided by the curator.	None	Validated by the System.
		Display confirmation Message of information captured.	This contains the details of the confirmation of all the details being captured successfully.	Used when the system saves all the information being successfully captured into the system.	None	Displayed by the System.
		[ALT] Display Error Message of Information not captured.	This contains the details of the confirmation of the unsuccessful capture of information.	Used when the system was not able to save all the information which was supposed to be captured into the system.	None	Displayed by the system

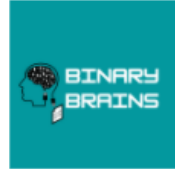


GROUP 23 DELIVERABLE 02



		Create and display new Surface Type details.	Contains information showing the curator the new Surface Type details created.	This is when the Surface Type creation is successful.	None	Displayed by the system
		[ALT] Show error message of unsuccessful creation of new Surface Type details.	Contains information to notify the curator that the Surface Type creation is unsuccessful.	This is when the Surface Type creation is unsuccessful.	None	Displayed by the system
Requirement Number:	Requirement Title:	Description:	Purpose:	Usage:	Entities & Attributes:	Logical Layout:
1.38	Search Surface Type Detail	Display Search Surface Type Page	This contains the details of the request of being able to search for a Surface Type.	This is when the system displays the Search Surface Type Page.	None	Displayed by the system
		[ALT] Validation Error encountered	Consists of the information required to validate the Surface Type details provided.	This is used when the system captures that there is an error in the information provided by the curator.	None	Validated by the System.
		Display confirmation Message of information captured.	This contains the details of the confirmation of all the details being captured successfully.	Used when the system saves all the information being successfully captured into the system.	None	Displayed by the System.
		[ALT] Display Error Message of Information not captured.	This contains the details of the unsuccessful capture of information.	Used when the system was not able to save all the information which was supposed to be captured into the system.	None	Displayed by the system

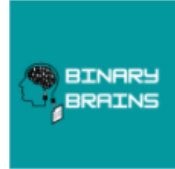
		Search and display selected Surface Type details.	Contains information showing the curator the Surface Type details searched for.	This is when the Surface Type search is successful.	None	Displayed by the system
		[ALT] Show error message of unsuccessful search of Surface Type details.	Contains information to notify the curator that the Surface Type search is unsuccessful.	This is when the Surface Type search is unsuccessful.	None	Displayed by the system
Requirement Number:	Requirement Title:	Description:	Purpose:	Usage:	Entities & Attributes:	Logical Layout:
1.39	Update surface type Detail	Display Update surface type Page	This contains the details of the search request of being able to update a surface type.	This is when the system displays the Update surface type Page.	None	Displayed by the system
		Display surface type detail	This contains details of the specific surface type that has been selected to be updated.	To update a specific surface type detail.	Surface Type Table: - Surface Type ID - Medium Type Description	Displayed by the system
		[ALT] Validation Error encountered	Consists of the information required to validate the surface type details provided.	This is used when the system captures that there is an error in the information provided by the curator.	None	Validated by the System.
		Show message of surface type	Contains information showing the curator	This is when the surface type is updated successfully	None	Displayed by the system



GROUP 23 DELIVERABLE 02



		details being successfully updated	the surface type details are updated.			
		[ALT] Show error message of unsuccessful update of surface type details.	Contains information to notify the curator that the surface type update is unsuccessful.	This is when the surface type update is unsuccessful.	None	Displayed by the system
Requirement Number:	Requirement Title:	Description:	Purpose:	Usage:	Entities & Attributes:	Logical Layout:
1.40	Delete surface type	Display Delete surface type Page	This contains the details of the search request of being able to delete a surface type.	This is when the system displays the Delete surface type Page.	None	Displayed by the system
		Display surface type detail	This contains details of the specific surface type that has been selected to be deleted.	To delete a specific surface type.	Surface Type Table: - Surface Type ID - Medium Type Description	Displayed by the system
		Show message of surface type details being successfully deleted	Contains information showing the curator the surface type details are deleted.	This is when the surface type is deleted successfully	None	Displayed by the system
		[ALT] Show error message of unsuccessful deletion of	Contains information to notify the curator that the surface type deletion is unsuccessful.	This is when the surface type deletion is unsuccessful.	None	Displayed by the system



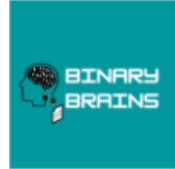
GROUP 23 DELIVERABLE 02



		surface type details.				
Requirement Number:	Requirement Title:	Description:	Purpose:	Usage:	Entities & Attributes:	Logical Layout:
1.41	View Art Class Feedback Form	Display View Feedback Form Page.	This contains the details of the request to view an art class feedback form.	This is when the system displays the feedback form page for the art classes.	None	Displayed by the system
		Display Feedback Form.	Contains information showing the curator the feedback form.	This is when the feedback form is displayed successfully	Feedback Table: • Feedback Page ID • Public User ID • Artist ID • Art Class ID • Feedback Comment • Rating Teacher • Rating Food • Rating Difficulty • Overall Rating	Displayed by the system
		[ALT] Show error message of unsuccessful display of feedback form	Contains information showing the curator the unsuccessful display of feedback form due to system error.	This is when the feedback form is not displayed.	None	Displayed by the system
Requirement Number:	Requirement Title	Description:	Purpose:	When Used:	Entities & Attributes:	Logical Layout:
2.1 Book Art Class Output						

2.1	Book Art Class	2.1 Prompt the user to enter booking details	This consists of the information required to prompt the user to provide the booking details	Used when the system wants the user to enter all required details for the booking of an art class	None	Prompted by the system
		2.1 ALT: Validation error information	Contains the information used to notify the user that the input values are invalid	This is used when the user enters invalid values into the input fields.	None	Generated by the system
		2.1 Prompts payment for booking	This consists of the information required to prompt the user to make a payment	This is used when the user completed entering their booking details to confirm seat	None	Prompted by the system
		2.1 Booking successful notification	This contains the information to notify the	This is used when the booking was successfully	None	Generated by the system.

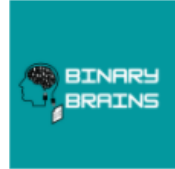
			user the booking was successful	made on the system		
		2.1 ALT: Booking unsuccessful notification	This contains the information to notify the user the booking was unsuccessful	This is used when a problem occurs, and the booking was unsuccessful	None	Generated by the system
2.2 View Existing Classes Output						
2.2	View Existing Classes	2.2 Displays all art classes information	This consists the information of all the available art classes to the user	This is used when the user wants to view all available classes on the system	None	Generated by the system.
		2.2 Displays art class information	This consists the information of a specific art class with all the details in that class	This is used when the user selects to view a specific art class	Art Class table: • ArtClassID • ArtClassName • ArtClassDescription • ArtClassPeriod • ClassPrice • ArtClassType • ClassTeacher • Venue	Generated by the system
2.3 Cancel Booking Output						



GROUP 23 DELIVERABLE 02



2.3	Cancel Booking	2.3 Prompts confirmation to cancel booking	This consists of the information required to prompt the user to confirm if they want to cancel the booking	This is used when the system needs confirmation from the user to proceed.	None	Prompted by the system
		2.3 Cancel Booking successful notification	This contains the information to notify the user that the cancellation of the booking was successful	This is used when the booking was successfully cancelled.	None	Generated by the system
2.4 Generate Booking Confirmation Message Output						
2.4	Generate Booking Confirmation Message	Generate booking confirmation message successful	This contains the information of the successfully generated booking confirmation message	This is used when the booking confirmation message has been successfully generated	ArtClass table: • ArtClassPeriod • ArtClassType • Venue • ClassTeacher • ArtClassName	Generated by the system



GROUP 23 DELIVERABLE 02



			and displays it to the user			
2.5 Refund Status Output						
2.5	Refund Status	2.5 Display refund status info	This contains the refund status information and displays it to the user.	This is used when the system successfully finds and displays the refund status information	Refund table: RefundStatus	Generated by the System
2.6 Capture Class Feedback						
2.6	Capture Class Feedback	2.6 Prompt to enter feedback details	This contains the details required to prompt the user to provide feedback details	Used when the user wants to add feedback of a class to the system	None	Prompted by the system
		2.6 ALT: Validation error information	Contains the information used to notify the user that the input values are invalid	This is used when the user enters invalid values into the input fields.	None	Generated by the system

		2.6 Prompts confirmation of feedback details	This consists of the information required to prompt the user to confirm if they want to submit their feedback details	This is used when the system needs confirmation from the user to proceed.	None	Prompted by the system
		2.6 Feedback successfully submitted notification	This contains the information to notify the user that the feedback form has successfully been submitted	This is used when the user submit their feedback form without any errors.	None	Generated by the system.
		2.6 ALT: Feedback unsuccessful notification	This contains the information to notify the user that the submission of the feedback	This is used when a problem occurs, and the feedback form has errors.	None	Generated by the system

			form was unsuccessful			
2.7 Generate Class Teacher Information Output						
2.7	Generate Class Teacher information	2.7 Display class teacher info	This contains the information of the successfully generated class teacher information and displays it to the user	This is used when the class teacher information has been successfully generated	Art_Class table: - ClassTeacher Artist table: - ArtistBio - ArtistName - ArtistSurname	Generated by the system
2.8 Make Payment Output						
2.8	Make Payment	2.8 Prompt select payment option info	This contains the details to prompt the user to select a payment option	This is used when the system requires the user to select what payment packaged to buy to proceed	None	Prompted by the system
		2.8 Prompt select payment method info	This contains the details to prompt the user to select a	This is used when the system requires the user to select what	None	Prompted by the system

			payment method	payment method, they would like to make		
		2.6 ALT: Validation error information	Contains the information used to notify the user that the input values are invalid	This is used when the user enters invalid values into the input fields.	None	Generated by the system
		2.8 Prompt enter card details info	This contains the details to prompt the user to enter their bank card details	This is used when the system requires the bank card details to verify the card is valid	None	Prompted by the system
		2.8 ALT: Incorrect card details	This contains the details to notify the user that the card details are invalid	This is used when the user enters their card details and the details are incorrect or invalid	None	Generated by the system
		2.8 Prompt amount due	This consists of the information	This is used when the system	None	Prompted by the system

		confirmation info	required to prompt the user to confirm if the amount due is correct	needs confirmation from the user to proceed with the payment.		
		2.8 Request one-time-pin info	This consist of the information to prompt the user to enter a one-time-pin to proceed with payment.	This is used when confirms their amount due to complete the payment	None	Prompted by the system
		2.8 ALT: Incorrect one-time-pin info	This consist of information to notify the user that the one-time-pin that was entered was incorrect	This is used when the user enters the wrong one-time-pin information.	None	Generated by the system
		2.8 Payment successful	This contains the information	This is used when the payment for	None	Generated by the system.

		notification info	to notify the user that the payment was successful	the art class went through without any problems		
		2.8 ALT: Payment unsuccessful notification	This contains the information to notify the user that the payment was unsuccessful	This is used when a problem occurs, during the payment or the user insufficient funds	None	Generated by the system
		2.8 Email payment successful info	This contains the information to notify the user that all payment details have been emailed	This is used when the payment was successful and sends all details contained in the payment to the user	PublicUser table: UserEmailAddress	Generated by the system
2.9 Request Refund Output						
2.9	Request Refund	2.9 ALT: Refund Unavailable	This contains the information to notify the user that a	This is used when the user wants to make a refund but it	None	Generated by the system

			refund request is unavailable	is too late and refund is unavailable		
		2.9 Prompt request refund confirmation info	This consists of the information required to prompt the user to confirm if they want to make a refund	This is used when the system needs confirmation from the user to request a refund	None	Prompted by the system
		2.9 Request refund successful notification	This contains the information to notify the user that the request for a refund was successful	This is used when the request for a refund has successfully been sent through	None	Generated by the system.
Requirement Number:	Requirement Title	Description:	Purpose:	When Used:	Entities & Attributes:	Logical Layout:
3.1 Generate New User Report Output						
3.1	Generate New User Report	3.1 Prompts for period range information	This contains the information	This is used when the system requires	None	Prompted by the system

			to prompt the curator to provide a period range for generating the report	the curator to provide a period range to generate the report		
		3.1 ALT: Period out of range information	This contains the information to notify the curator that the period provided is out of range	This is used when the system can not generate a report because of the period being out of range	None	Generated by the system
		3.1 Successfully generate new user report information	This contains the information of the new generated report of each new user in the given period range	This is used when the system successfully generates a new user report and displays it to the curator	None	Generated by the system

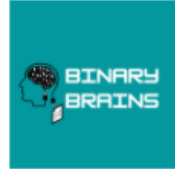
		3.1 Email new user report successful information	This contains the information to notify the curator that the report has also been emailed to their account	This is used when the new user report has successfully been generated and emails this report to the curators account	None	Generated by the system
3.2 Generate Art Class Report Output						
3.2	Generate Art Class Report	3.2 Prompts for period range information	This contains the information to prompt the curator to provide a period range for generating the report	This is used when the system requires the curator to provide a period range to generate the report	None	Prompted by the system
		3.2 ALT: Period out of range information	This contains the information to notify the curator	This is used when the system cannot generate a report	None	Generated by the system

			that the period provided is out of range	because of the period being out of range		
		3.2 Successfully generate art class report information	This contains the information of the new generated report of each art class in the given period range	This is used when the system successfully generates a art class report and displays it to the curator	None	Generated by the system
		3.2 Email art class report successful information	This contains the information to notify the curator that the report has also been emailed to their account	This is used when the art class report has successfully been generated and emails this report to the curators account	None	Generated by the system
3.3 Generate Class Teacher Report Output						

3.3	Generate Class Teacher Report	3.3 Prompts for period range information	This contains the information to prompt the curator to provide a period range for generating the report	This is used when the system requires the curator to provide a period range to generate the report	None	Prompted by the system
		3.3 ALT: Period out of range information	This contains the information to notify the curator that the period provided is out of range	This is used when the system cannot generate a report because of the period being out of range	None	Generated by the system
		3.3 Successfully generate class teacher report information	This contains the information of the new generated report of each class	This is used when the system successfully generates a class teacher report and	None	Generated by the system

			teacher in the given period range	displays it to the curator		
		3.3 Email class teacher report successful information	This contains the information to notify the curator that the report has also been emailed to their account	This is used when the class teacher report has successfully been generated and emails this report to the curators account	None	Generated by the system
3.4 Generate Exhibition Report Output						
3.4	Generate Exhibition Report	3.4 Prompts for period range information	This contains the information to prompt the curator to provide a period range for generating the report	This is used when the system requires the curator to provide a period range to generate the report	None	Prompted by the system

		3.4 ALT: Period out of range information	This contains the information to notify the curator that the period provided is out of range	This is used when the system cannot generate a report because of the period being out of range	None	Generated by the system
		3.4 Successfully generate exhibition report information	This contains the information of the new generated report of each exhibition in the given period range	This is used when the system successfully generates an exhibition report and displays it to the curator	None	Generated by the system
		3.4 Email exhibition report successful information	This contains the information to notify the curator that the	This is used when the exhibition report has successfully been generated	None	Generated by the system



GROUP 23 DELIVERABLE 02



			report has also been emailed to their account	and emails this report to the curators account		
3.5 Generate Class Survey Report Output						
3.5	Generate Class Survey Report	3.5 Prompts for period range information	This contains the information to prompt the curator to provide a period range for generating the report	This is used when the system requires the curator to provide a period range to generate the report	None	Prompted by the system
		3.5 ALT: Period out of range information	This contains the information to notify the curator that the period provided is out of range	This is used when the system cannot generate a report because of the period being out of range	None	Generated by the system

		3.5 Successfully generate class survey report information	This contains the information of the new generated report of each class survey in the given period range	This is used when the system successfully generates a class survey report and displays it to the curator	None	Generated by the system
		3.5 Email class survey report successful information	This contains the information to notify the curator that the report has also been emailed to their account	This is used when the class survey report has successfully been generated and emails this report to the curators account	None	Generated by the system
3.6 Generate Payment Report Output						
3.6	Generate Payment Report	3.6 Prompts for period range information	This contains the information	This is used when the system requires	None	Prompted by the system

			to prompt the curator to provide a period range for generating the report	the curator to provide a period range to generate the report		
		3.6 ALT: Period out of range information	This contains the information to notify the curator that the period provided is out of range	This is used when the system cannot generate a report because of the period being out of range	None	Generated by the system
		3.6 Successfully generate payment report information	This contains the information of the new generated report of each payment in the given period range	This is used when the system successfully generates a payment report and displays it to the curator	None	Generated by the system

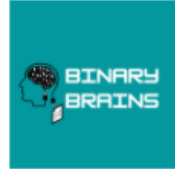
		3.6 Email payment report successful information	This contains the information to notify the curator that the report has also been emailed to their account	This is used when the payment report has successfully been generated and emails this report to the curators account	None	Generated by the system
Requirement Number:	Requirement Title:	Description:	Purpose:	Usage:	Entities & Attributes:	Logical Layout:
4.1	Register User	Prompt user to enter details.	This contains the details needed to be filled in.	This is when the system displays the Registration Page and wants the user to enter their details	None	Prompted by the system
		[ALT] Validation Error encountered	Consists of the information required to validate the registration information.	This is used when the system captures that there is an error in the information provided by the User.	None	Validated by the System.
		[ALT] Notify user of Error in registration	This contains information showing that the validation was unsuccessful and an error has occurred.	Used when system detects validation to be unsuccessful and notify user.	None	Displayed by the system

		Show confirmation message of successful registration	Contains information to notify the user that the registration was successful	This is when the registration is successful.	None	Displayed by the system
		[ALT] Show error message of unsuccessful registration.	Contains information to notify the user that the registration was successful	This is when the registration is unsuccessful.	None	Displayed by the system
Requirement Number:	Requirement Title:	Description:	Purpose:	Usage:	Entities & Attributes:	Logical Layout:
4.2	login	Prompt user to enter details.	This contains the information needed to allow a user to log in	This is when the system displays the Login Page and wants the user to enter their details.	None	Prompted by the system
		[ALT] Validation Error encountered	Consists of the information required to validate the login details provided.	This is used when the system captures that there is an error in the information provided by the user .	None	Validated by the System.
Requirement Number:	Requirement Title:	Description:	Purpose:	Usage:	Entities & Attributes:	Logical Layout:
4.3	Logout	Prompt user to enter details.	This contains the details needed to be filled in.	This is when the system displays the Registration Page and wants to enter their details	None	Prompted by the system
		Display Confirm logout process	Consists of the information required to confirm the log out process.	This is used when the system requests user to confirm the logout process	None	Displayed by the System.

		[ALT] Show error message to unsuccessful log out	This contains of the information required to notify the user of unsuccessful system logout	This is when the system cannot log out the user	None	Displayed by the system
Requirement Number:	Requirement Title:	Description:	Purpose:	Usage:	Entities & Attributes:	Logical Layout:
4.4	Reset Password	Prompt user to enter details.	This contains the details needed to be filled in.	This is when the system displays the Forgot Password page and wants to enter their details	User	Prompted by the system
		[ALT] Validation Error encountered	Consists of the information required to validate the details provided.	This is used when the system captures that there is an error in the information provided by the user.	User	Validated by the System.
		Prompt user to enter new password	This contains the details of the new password.	Used when the system saves requests user to enter new password details.	None	Displayed by the System.
		[ALT] Validation Error encountered with password details	Consists of the information required to validate the details provided.	This is used when the system captures that there is an error in the information provided by the user.	None	Validated by the System.
		Prompt user to confirm process	Contains information of process being confirmed.	When new password process needs to be confirmed	None	Prompted by the system
Requirement Number	Requirement Title:	Description:	Purpose:	Usage:	Entities & Attributes:	Logical Layout:

4.5	Update User	Prompt user to select details they want to update.	This contains all the user details that will be needed to be updated.	This is when the system displays the Update Page and wants to enter their details	None	Prompted by the system
		[ALT] Validation Error encountered	Consists of the information required to validate the update user details provided.	This is used when the system captures that there is an error in the information provided by the user.	None	Validated by the System.
		Prompt user to confirm details that they want to update.	This contains the confirmation of all the details that will be needed to be updated.	Used when the system wants to save all the information and continue with the update process.	None	Prompted by the System.
Requirement Number:	Requirement Title:	Description:	Purpose:	Usage:	Entities & Attributes:	Logical Layout:
4.6	View Software Guide	Display software guide link	This contains the details needed to display the software guide information	This is when the system displays the software Guide link that is to be viewed	None	Displayed by the system
		Display Software in PDF format	This contains the details needed to display the software guide information as a PDF	This is when the system displays the software Guide in a PDF format to be able to be viewed	None	Validated by the System.
Requirement Number:	Requirement Title:	Description:	Purpose:	Usage:	Entities & Attributes:	Logical Layout:
4.7	View Notifications	Display notifications	This contains the details that are needed in order to display the	This is when the system displays the Notifications.	None	Displayed by the system

			notifications to the user			
		[ALT] Pop up message displayed showing no notifications	Consists of information that is needed in order to show the user that there are no available notifications	This is used when the system shows that no notifications have been displayed.	None	Displayed by the System.
		Display view notification	Consists of information that shows that the notification has been viewed by the user	This is when the user as viewed the notification and the system displays it as viewed.	None	Displayed by the system.
Requirement Number:	Requirement Title:	Description:	Purpose:	Usage:	Entities & Attributes:	Logical Layout:
4.8	Generate Frame information	Search for frames	Contains information to search for frames based on search results.	When user enters information and wants the result of the frames searched for.	None	Searched by the system.
		Display Frame information	Contains of information to display the searched result	When the system displays all the frame information as a result of the search	None	Displayed by the system
Requirement Number:	Requirement Title:	Description:	Purpose:	Usage:	Entities & Attributes:	Logical Layout:
5.1	Apply for exhibition	Prompt artist to enter details.	This contains the details needed to be filled in.	This is when the system displays the form and wants details to be entered.	None	Prompted by the system
		[ALT] Validation	Consists of the information required to validate the	This is used when the system captures that there is an error in the	None	Validated by the System.

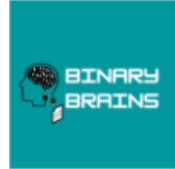


GROUP 23 DELIVERABLE 02



		Error encountered	exhibition details provided.	information provided by the artist.		
		Prompt artist to enter tag details.	This contains the details needed to be filled in the tags	This is when the system displays the tags and wants details to be entered.	None	Prompted by the system
		[ALT] Tag Validation Error encountered	Consists of the information required to validate the exhibition details provided.	This is used when the system captures that there is an error in the tag information provided by the artist.	None	Validated by the System.
Requirement Number:	Requirement Title:	Description:	Purpose:	Usage:	Entities & Attributes:	Logical Layout:
5.2	Apply to Teach Class	Request to fill in form.	This contains the details needed to be filled in the form.	This is when the system displays the form and wants to enter their details	None	Prompted by the system
		[ALT] Validation Error encountered	Consists of the information required to validate the form details provided.	This is used when the system captures that there is an error in the information provided by the user.	None	Validated by the System.
Requirement Number:	Requirement Title:	Description:	Purpose:	Usage:	Entities & Attributes:	Logical Layout:
5.3	Add Picture	Display Browse option	This contains the details needed to display the browse option for the artist to select a picture	This is when the system displays the files that need to be selected in order to be added to the system	None	Displayed by the system
		[ALT] Validation Error encountered	Consists of the information required to validate the format of the picture	This is used when the system captures that there is an error in the	None	Validated by the System.

				information provided by the artist.		
		Display confirmation Message of picture being uploaded.	This contains the details of the confirmation that the picture selected is to be uploaded	Used when the system wants to confirm the process of uploading the selected picture onto the system.	None	Displayed by the System.
		Show confirmation message of successfully adding a picture	Contains information to notify the curator that a picture has been successfully added.	This is when the picture was added successfully.	None	Displayed by the system
Requirement Number:	Requirement Title:	Description:	Purpose:	Usage:	Entities & Attributes:	Logical Layout:
5.4	Update Picture	System displays update picture screen	To update the picture	This is when the system displays the Update picture screen	None	Displayed by the system
		[ALT] Validation Error encountered	Consists of the information required to validate the update format.	This is used when the system captures that there is an error in the information provided by the artist.	None	Validated by the System.
		Display confirmation Message of picture being uploaded.	This contains the details of the confirmation that the picture selected is to be uploaded	Used when the system wants to confirm the process of uploading the selected picture onto the system.	None	Displayed by the System.
		Show confirmation message of successfully	Contains information to notify the curator that a picture has	This is when the picture was added successfully.	None	Displayed by the system

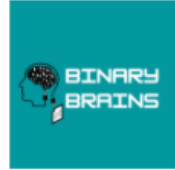


GROUP 23 DELIVERABLE 02



		adding a picture	been successfully added.			
Requirement Number:	Requirement Title:	Description:	Purpose:	Usage:	Entities & Attributes:	Logical Layout:
5.5	Delete Picture	System displays delete picture screen	To delete the picture	This is when the system displays the Delete picture screen	None	Displayed by the system
		Display confirmation Message of confirming the delete process	This contains the details of the confirmation whether all the details should be deleted.	Used when the system is about to delete all the selected info	None	Displayed by the System.
		Show confirmation message of successfully deleting picture	Contains information to notify the artist that the picture has been deleted successfully	This is when the picture has been deleted	None	Displayed by the system
Requirement Number:	Requirement Title:	Description:	Purpose:	Usage:	Entities & Attributes:	Logical Layout:
5.6	View invitation	Search and display invitation	This contains the details needed to search and display the invitation	This is when the system displays the Invitations of the artist	Invite Table: Invitation ID	Displayed by the system
		[ALT] Display error message of no new invitations available	To show the user that there are no invitations available	When there are no invitations available to view	None	Displayed by the system
Requirement Number:	Requirement Title:	Description:	Purpose:	Usage:	Entities & Attributes:	Logical Layout:

5.7	View Existing Exhibitions	Display Exhibition Details	To display the exhibition details needed to be viewed by the artist	When exhibitions exist and are requested to be viewed	Exhibition Table: - Exhibition ID - Exhibition Name - Exhibition Description - Exhibition Date - Organization ID - Exhibition Time - Schedule ID - Venue ID - Exhib Announce ID	Displayed by the system
Requirement Number:	Requirement Title:	Description:	Purpose:	Usage:	Entities & Attributes:	Logical Layout:
5.8	Accept invitation for Exhibition	Display invitation information	This contains the details for all the invitations	This is when the system displays the Invitations Page and wants to accept an invitation	Invitation Table: - Invitation Details - Invitation Date - Field - Invitation Status - InvitationID - Artist ID	Displayed by the system
		Update invitation status	This contains information when the invitation is accepted.	When the invitation is accepted	Exhibition_Artist Table - Invitation Status - Artist Id	Displayed by system
		[ALT] Update invitation status	This contains information when the invitation is not accepted.	When the invitation is not accepted	Exhibition_Artist Table - Invitation Status - Artist Id	Displayed by system

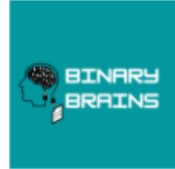


GROUP 23 DELIVERABLE 02



Conclusion

In conclusion, the above section contains in-depth descriptions of all the different reports and other outputs of the proposed system as depicted in the narratives and the complex diagrams.



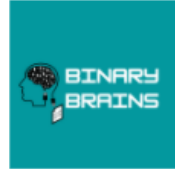
8. Validation

Introduction

This section contains the complete validation table for each requirement in the proposed system and shows the use cases, processes, and entities that satisfy the mentioned requirements.

8.1 Validation of functional specifications against requirements

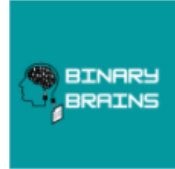
Subsystem	Requirement:	Use Case:	Process:	Entities:
1.Admin	1.1 Approve Exhibition User	1.1 Approve Exhibition User	1.1.1 Retrieve and Display Applications Task	Application
			1.1.2 Display specific Application Task	None
			1.1.3 Confirmation and Validation of Application Approval Task	None
			1.1.4 Save Exhibition User Task	Exhibition
			1.1.5 Notification to Users Task	None
	1.2 Search Exhibition User	1.2 Search Exhibition User	1.2.1 Capture Search Exhibition User Task	None
			1.2.2 Validate Search Exhibition User Task	None
			1.2.3 Retrieve Search results Task	Exhibition Artist
			1.2.4 Display Search Exhibition User Results task	None
	1.3 Delete Exhibition User	1.3 Delete Exhibition User	1.3.1 Retrieve and Display Exhibition User Task	None



GROUP 23 DELIVERABLE 02



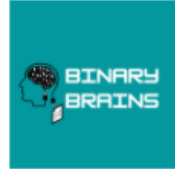
			1.3.2 Confirmation and Validation of Delete Exhibition User Task	None
			1.3.3 Delete Exhibition User Task	Exhibition Artist
			1.3.4 Notification of Delete Exhibition User Task	None
	1.4 Create new Exhibition	1.4 Create new Exhibition	1.4.1 Capture New exhibition Task	None
			1.4.2 Confirm and Validate New Exhibition Task	None
			1.4.3 Save New Exhibition Task	Exhibition
			1.4.4 Announce New Exhibition Task	None
	1.5 Search Exhibition	1.5 Search Exhibition	1.5.1 Capture Search Exhibition Task	None
			1.5.2 Validate Search Exhibition Task	None
			1.5.3 Retrieve Search Exhibition Task	Exhibition
			1.5.4 Display Search Exhibition Task	None
	1.6 Update Exhibition	1.6 Update Exhibition	1.6.1 Retrieve and display Exhibition Search Task	Exhibition
			1.6.2 Capture Exhibition Update Details Task	None
			1.6.3 Confirm and Validate Exhibition Update Details Task	None
			1.6.4 Save Exhibition Update Details Task	Exhibition
			1.6.5 Display Exhibitions Update Details Task	None
	1.7 Delete Exhibition	1.7 Delete Exhibition	1.7.1 Retrieve and Display Search Exhibition Delete Task	None
			1.7.2 Select Exhibition Delete Task	None
			1.7.3 Confirmation and Validation Exhibition Delete Task	None
			1.7.4 Delete Exhibition Task	Exhibition
	1.8 View Artist	1.8 View Artist	1.8.1 Capture artist view Details Task	None
			1.8.2 Process and Validate Artist View Details Task	None
			1.8.3 Retrieve View Artist Details Task	Artist
			1.8.4 Display View Artist Details Task	None
	1.9 Add Artwork	1.9 Add Artwork	1.9.1 Capture Add Artwork Details Task	None
			1.9.2 Validate and Confirm Add Artwork Details Task	None
			1.9.3 Save New Artwork Details Task	Artwork, Tag
			1.9.4 Display Artwork successfully added Task	None
	1.10 Search Artwork	1.10 Search Artwork	1.10.1 Retrieve search artwork details Task	None
			1.10.2 Capture criteria Search artwork Detail's task	None



GROUP 23 DELIVERABLE 02



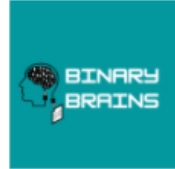
			1.10.3 Validate Search Artwork Details Task	None
			1.10.4 Process Search Artwork Details Task	Artwork
			1.10.5 Display Search Artwork Details Task	None
	1.11 Delete Artwork	1.11 Delete Artwork	1.11.1 Retrieve and display Artwork Task	None
			1.11.2 Delete criteria Artwork Task	None
			1.11.3 Confirm and Validate Details artwork Task	None
			1.11.4 Delete artwork Task	Artwork
			1.11.5 Notification of Delete Artwork Task	None
	1.12 Update Artwork	1.12 Update Artwork	1.12.1 Retrieve and Display Artwork Details Task	Artwork
			1.12.2 Capture Updated Artwork Details Task	None
			1.12.3 Validate and Confirm Updated Artwork Details Task	None
			1.12.4 Save Updated Artwork Details Task	Artwork
			1.12.5 Updated Artwork details announcement Task	None
	1.13 Create artist Exhibition Invitation	1.13 Create artist Exhibition Invitation	1.13.1 Capture Artist Exhibition Invitation Details Task	None
			1.13.2 Validate and Confirm Artist Exhibition Invitation Details Task	None
			1.13.3 Save New Artist Exhibition Invitation Task	Invite
			1.13.4 Display New Artist Exhibition Invitation Task	None
	1.14 Print Tags	1.14 Print Tags	1.14.1 Compile Tag details Task	Tag
			1.14.2 Print preference details Task	None
			1.14.3 Compile and Print Tag details Task	None
	1.15 View Schedule	1.15 View Schedule	1.15.1 Gather Schedule Timeline Data Task	None
			1.15.2 Compile Schedule Data Task	Schedule
	1.16 Print Schedule	1.16 Print Schedule	1.16.1 Gather Schedule timeline print data Task	None
			1.16.2 Print Preference Schedule details Task	None
			1.16.3 Compile and Print Schedule Details Task	Schedule
	1.17 Create Class	1.17 Create Class	1.17.1 Capture Art Class Details Task	Art Class
			1.17.2 Validate and Confirm Art Class Details Task	None
			1.17.3 Save Art Class Details Task	None
			1.17.4 Updated Display Art Class Task	None



GROUP 23 DELIVERABLE 02



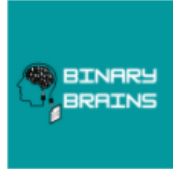
Subsystem	Requirement	Use Case	Purpose(DFD)	Entities
Admin Subsystem	1.18 Search Class	1.18 Search Class	1.18.1 Capture Search Class Task	None
			1.18.2 Validate captured Parameter Task	None
			1.18.3 Retrieve Search Results Task	Art Class
			1.18.4 Display Search Results Task	None
			1.18.5 Display Selected Class Task	None
	1.19 Update Class	1.19 Update Class	1.19.1 Retrieve and Display Search Class Task	None
			1.19.2 Capture Updated Class Details Task	None
			1.19.3 Validate Captured Details Task	None
			1.19.4 Confirmation of Changes Made Task	None
			1.19.5 Save Changes to Art Class Task	Art Class
			1.19.6 Notification of Successful Update Task	None
	1.20 Delete Class	1.20 Delete Class	1.20.1 Retrieve and Display Class Task	None
			1.20.2 Confirmation of changes made task	None
			1.20.3 Delete Class Task	Art Class



GROUP 23 DELIVERABLE 02



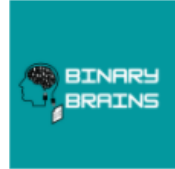
			1.20.4 Notification of Successfully Deleted Task	None
			1.20.4 ALT: Deletion Notification of Unsuccessful Task	None
	1.21 Generate Class Reminder	1.21 Generate Class Reminder	1.21.1 Gather Information Task	None
			1.21.2 Aggregate Report Task	None
			1.21.3 Search User Details Task	None
			1.21.4 Send Report Task	None
	1.22 Generate Payment Reminder	1.22 Generate Payment Reminder	1.22.1 Gather Information Task	None
			1.22.2 Aggregate Report Task	None
			1.22.3 Search User Details Task	None
			1.22.4 Send Report Task	None
	1.23 Create Class Teacher	1.23 Create Class Teacher	1.23.1 Capture Class Teacher Details Task	None
			1.23.2 Validate Captured Details Task	None
			1.23.3 Confirmation of Captured Details Task	None
			1.23.4 Read Class Teacher ID Task	ClassTeacher
			1.23.5 Save Art Class Teacher Details Task	None
			1.23.6 Successfully Created Notification Task	None



GROUP 23 DELIVERABLE 02



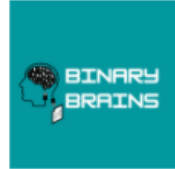
	1.24 Search Class Teacher	1.24 Search Class Teacher	1.24.1 Capture Search Class Teacher Task	None
			1.24.2 Validate captured Parameter Task	None
			1.24.3 Retrieve Search Results Task	ClassTeacher
			1.24.4 Display Search Results Task	None
			1.24.5 Display Selected Class Task	None
	1.25 Update Class Teacher	1.25 Update Class Teacher	1.25.1 Retrieve and Display Search Class Teacher Task	None
			1.25.2 Capture Updated Class Teacher Details Task	None
			1.25.3 Validate Captured Details Task	None
			1.25.4 Confirmation of Changes Made Task	None
			1.25.5 Save Changes to Art Class Teacher Task	ClassTeacher
			1.25.6 Notification of Successful Update Task	None
	1.26 Delete Class Teacher	1.26 Delete Class Teacher	1.26.1 Retrieve and Display Class Teacher Task	None
			1.26.2 Confirmation of changes made task	None
			1.26.3 Delete Class Teacher Task	ClassTeacher
			1.26.4 Notification of Successfully Deleted Task	None
			1.26.4 ALT: Deletion Notification of Unsuccessful Task	None



GROUP 23 DELIVERABLE 02



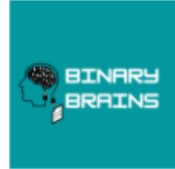
Subsystem	Requirement:	Use Case:	Process:	Entities:
	1.27 Generate Booking Information	1.27 Generate Booking Information	1.27.1 Load Booking Info Task	None
			1.27.2 Validate Booking Details Task	None
			1.27.3 Confirm create notification booking task	None
			1.27.4 Save Create notification info Task	None
			1.27.5 Create booking Notification Task	Booking Table
	1.28 Create Medium Detail	1.28 Create Medium Detail	1.28.1 Load Create Medium Page Task	None
			1.28.2 Validate Create Medium Info Task	None
			1.28.3 Confirm and Capture create medium info Task	None
			1.28.4 Save and Create Medium Info Task	None
			1.28.5 Create Medium Task	Medium Type
	1.29 Search Medium	1.29 Search Medium	1.29.1 Load Search medium page Task	None
			1.28.2 Validate search medium info Task	None
			1.29.3 Confirm and save medium info Task	None
			1.29.4 Search Medium Task	Medium Type
			1.29.5 Search artworks related to that medium Task	Artwork
	1.30 Update Medium Detail	1.30 Update Medium Detail	1.30.1 Load update medium page Task	None
			1.30.2 Capture selection of medium Task	Medium Type



GROUP 23 DELIVERABLE 02



			1.30.3 Validate update medium info Task	None
			1.30.4 Confirm update Medium Task	None
			1.30.5 Update medium Task	Medium Type, Artwork
	1.31 Delete medium	1.31 Delete medium	1.31.1 Delete all mediums Task	None
			1.31.2 Capture medium selection Task	Medium Type
			1.31.3 Confirm Deletion Task	None
			1.31.4 Delete Medium Task	Medium Type, Artwork
	1.32 Create Artwork Detail	1.32 Create Artwork Detail	1.32.1 Load Create Artwork Page Task	None
			1.32.2 Validate Create Artwork Info Task	None
			1.32.3 Confirm and Capture create Artwork info Task	None
			1.32.4 Save and Create Artwork Info Task	None
			1.32.5 Create Artwork Task	Artwork
	1.33 Search Artwork	1.33 Search Artwork	1.33.1 Load Search Artwork Page Task	None
			1.33.2 Validate search Artwork info Task	None
			1.33.3 Confirm and save Artwork info Task	None
			1.33.4 Search Artwork Task	None
			1.33.5 Load Artwork Task	Artwork
	1.34 Update Artwork detail	1.34 Update Artwork detail	1.34.1 Load update Artwork page Task	None
			1.34.2 Capture selection of Artwork Task	Artwork
			1.34.3 Validate update Artwork info Task	None
			1.34.4 Confirm update Artwork Task	None
			1.34.5 Update Artwork Task	Artwork
	1.35 Delete Artwork	1.35 Delete Artwork	1.35.1 Delete all Artworks Task	None
			1.35.2 Capture Artwork selection Task	Artwork
			1.35.3 Confirm Deletion Task	None
			1.35.4 Delete Artwork Task	Artwork
	1.36 Archive Status	1.36 Archive Status	1.36.1 Load Artwork Details Task	Archive Status
			1.36.2 Update Artwork Task	Archive Status, Artist
	1.37 Create surface Type Details	1.37 Create surface Type Details	1.37.1 Load Create surface Type Page Task	None



GROUP 23 DELIVERABLE 02



			1.37.2 Validate Create surface Type Info Task	None
			1.37.3 Confirm and Capture create surface Type info Task	None
			1.37.4 Save and Create surface Type Info Task	None
			1.37.5 Create surface Type Task	Surface Type
	1.38 Search surface Type	1.38 Search surface Type	1.38.1 Load Search surface Type page Task	None
			1.38.2 Validate search surface Type info Task	None
			1.38.3 Confirm and save surface Type info Task	None
			1.38.4 Search surface Type Task	Surface Type
			1.38.5 Search artworks related to that surface Type Task	Artwork
	1.39 Update surface Type Details	1.39 Update surface Type Details	1.39.1 Load update surface Type page Task	None
			1.39.2 Capture selection of surface Type Task	Surface Type
			1.39.3 Validate update surface Type info Task	None
			1.39.4 Confirm update surface Type Task	None
			1.39.5 Update surface Type Task	Surface Type, Artwork
	1.40 Delete surface Type	1.40 Delete surface Type	1.40.1 Delete all surface Type Task	None
			1.40.2 Capture surface Type selection Task	Surface Type
			1.40.3 Confirm Deletion Task	None
			1.40.4 Delete surface Type Task	Surface Type, Artwork
	1.41 View Art Class Feedback form	1.41 View Art Class Feedback form	1.41.1 Load Feedback Form page	None
			1.41.2 Capture form selection Task	None
			1.41.3 Display Feedback Form Task	Feedback

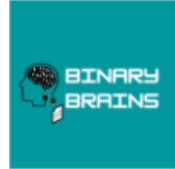
Subsystem	Requirement	Use Case	Process (DFD)	Entities (ERD)
2. Art Class	2.1 Book Art Class	2.1 Book Art Class	2.1.1 Capture class information	

			and display Art Class Booking details Task	
			2.1.2 Capturing Booking details for Art Class Task	
			2.1.3 Validate Captured Booking details task	
			2.1.4 Capture and calculate User Payment details Task	
			2.1.5 Save and stores Art class Booking details Task	Booking
			2.1.6 Notification of successful Art Class Booking Task	
	2.2 View Existing Classes	2.2 View Existing Classes	2.2.1 Captures and displays all available Art Classes Task	Art_Class
			2.2.2 Captures and displays specific art class details Task	Art_Class

	2.3 Cancel Booking	2.3 Cancel Booking	2.3.1 Capture class information and display Art Class Booking details Task	
			2.3.2 Captures Booking details and displays booking details Task	Booking
			2.3.3 Confirmation of Cancel booking Task	Booking
			2.3.4 Notification of successful cancellation of booking Task	
	2.4 Generate Booking Confirmation Message	2.4 Generate Booking Confirmation Message	2.4.1 Capture class information and display Art Class Booking details Task	
			2.4.2 Captures and gathers Booking and Art Class details Task	Art_Class Booking
			2.4.3 Successfully generated Art	

			class and booking information Task	
	2.5 Refund Status	2.5 Refund Status	2.5.1 Capture class information and display Art Class Booking details Task	
			2.5.2 Capture and displays refund details of the booking Task	Booking Refund
	2.6 Capture Class Feedback	2.6 Capture Class Feedback	2.6.1 Capture class information and display Art Class Booking details Task	
			2.6.2 Capturing Feedback details for Art Class Task	
			2.6.3 Validate Captured Feedback details task	
			2.6.4 Confirmation of Feedback details Task	
			2.6.5 Captures and stores	Feedback

			feedback details Task	
			2.6.6 Notification of successful feedback details stored Task	
	2.7 Generate Class Teacher information	2.7 Generate Class Teacher Information	2.7.1 Capture class information and display Art Class Booking details Task	
			2.7.1 Captures and displays teacher information Task	Class_Teacher Artist
	2.8 Make Payment	2.8 Make Payment	2.8.1 Captures and displays payment amount Task	Art_Class
			2.8.2 Captures and validates payment option Task	PaymentType
			2.8.3 Captures and validates payment method Task	Payment
			2.8.4 Validating card details Task	



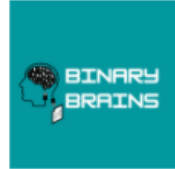
GROUP 23 DELIVERABLE 02



			2.8.5 Captures card details and uses virtual card services to contact bank Task	
			2.8.6 Confirmation of amount due Task	
			2.8.7 Validating one-time-pin details Task	
			2.8.8 Captures payment details and generates new payment info Task	Payment
			2.8.9 Notification of successful payment Task	Booking
			2.8.10 Sends payment confirmation Task	
	2.9 Request Refund	2.9 Request Refund	2.9.1 Capture payment details of a specific booking Task	Booking Payment
			2.9.2 Confirmation of	

			refund request Task	
			2.9.3 Captures and generates new refund information Task	Refund
			2.9.4 Notification of successful refund request Task	

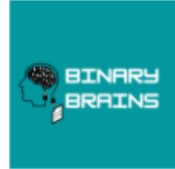
Subsystem	Requirement	Use Case	Process (DFD)	Entities (ERD)
3. Reporting	3.1 Generate New User Report	3.1 Generate New User Report	3.1.1 Captures request and finds User details Task	User
			3.1.2 Captures period range and gathers user info accordingly Task	User
			3.1.3 Displays new user report successfully Task	
			3.1.4 Finds email and sends report details Task	Administration
	3.2 Generate Art Class Report	3.2 Generate Art Class Report	3.2.1 Captures request and finds	Art_Class Booking



GROUP 23 DELIVERABLE 02



			art class details Task	
			3.2.2 Captures period range and gathers art class info accordingly Task	Art_Class
			3.2.3 Displays art class report successfully Task	
			3.2.4 Finds email and sends report details Task	Administration
	3.3 Generate Class Teacher Report	3.3 Generate Class Teacher Report	3.3.1 Captures request and finds class teacher details Task	Art_Class
			3.3.2 Captures period range and gathers class teacher info accordingly Task	Art_Class
			3.3.3 Displays class teacher report successfully Task	
			3.3.4 Finds email and sends report details Task	Administration



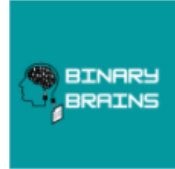
GROUP 23 DELIVERABLE 02



	3.4 Generate Exhibition Report	3.4 Generate Exhibition Report	3.4.1 Captures request and finds exhibition details Task	Exhibition
			3.4.2 Captures period range and gathers exhibition info accordingly Task	Exhibition
			3.4.3 Displays exhibition report successfully Task	
			3.4.4 Finds email and sends report details Task	Administration
	3.5 Generate Class Survey Report	3.5 Generate Class Survey Report	3.5.1 Captures request and finds Feedback details Task	Feedback
			3.5.2 Captures period range and gathers feedback info accordingly Task	Feedback
			3.5.3 Displays class survey report successfully Task	

			3.5.4 Finds email and sends report details Task	Administration
	3.6 Generate Payment Report	3.6 Generate Payment Report	3.6.1 Captures request and finds Payment details Task	Payment
			3.6.2 Captures period range and gathers payment info accordingly Task	Payment
			3.6.3 Displays payment report successfully Task	
			3.6.4 Finds email and sends report details Task	Administration

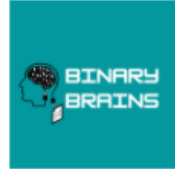
Subsystem	Requirement:	Use Case:	Process:	Entities:
4.User	4.1 Register User	4.1 Register User	4.1.1 Capture User Personal Details Info Task	User role
			4.1.2 Validate User Personal Details Info Task	None



GROUP 23 DELIVERABLE 02



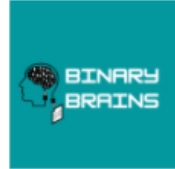
			4.1.3 Verify for User Existing Profile Task	None
			4.1.4 Save new user details info Task	User
			4.1.5 Generate successful user registration info Task	None
	4.2 Log in	4.2 Log in	4.2.1 Capture User Login Details Task	None
			4.2.2 Validate Captured Details Info Task	None
			4.2.3 Verify User Login Task	User, User Role
			4.2.4 Allow User Login Task	None
			4.2.5 Generate successful user registration info Task	None
	4.3 Log out	4.3 Log out	4.3.1 Logout Confirmation Task	None
			4.3.2 Logout Task	None
	4.4 Reset Password	4.4 Reset Password	4.4.1 Capture Email and Username to reset password Task	None
			4.4.2 Verify User Info Task	User
			4.4.3 Capture New Password Task	User
	4.5 Update User	4.5 Update User	4.5.1 Find User Info Task	User, User Role
			4.5.2 Capture Updated User Info Task	None
			4.5.3 Verify and Confirm User Info Task	None
			4.5.4 Update and Save User Info Task	User
			4.5.5 Generate successful update notification Task	None
	4.6 View Software Guide	4.6 View Software Guide	4.6.1 Display link to viewing software guide Task	None
			4.6.2 Preview Software Guide in PDF form at guide Task	None
	4.7 View notification	4.7 View notification	4.7.1 Search View Notifications Request Task	None
			4.7.2 Display notification details info Task	None
	4.8 Generate Frame Information	4.8 Generate Frame Information	4.8.1 Search for all frames stored Task	None
			4.8.2 Generate and Display Frame info Task	Frame Colour, Frame Dimension



GROUP 23 DELIVERABLE 02



Subsystem	Requirement:	Use Case:	Process:	Entities:
5.Artist	5.1 Apply for Exhibition	5.1 Apply for Exhibition	5.1.1 Capture and Validate artist Exhibition Application Details Info Task	Artist
			5.1.2 Capture and Validate tag details info Task	None
			5.1.3 Confirm exhibition application info	None
			5.1.4 Save Exhibition Application Info Task	Exhibition
			5.1.5 Generate Successful Exhibition Application Task	None
	5.2 Apply to Teach Class	5.2 Apply to Teach Class	5.2.1 Capture artist Class Teacher application details Task	Artist
			5.2.2 Verify art class teacher info Task	None
			5.2.3 Confirm artist class teacher details info Task	None
			5.2.4 Save artist class teacher application details info Task	Class Teacher
			5.2.5 Generate successful artist class teacher application info Task	None
	5.3 Add Picture	5.3 Add Picture	5.3.1 Add picture info Task	None
			5.3.2 Capture and Verify that the picture is in the right format Task	None
			5.3.3 Confirm picture format info Task	None
			5.3.4 Upload and Save Picture into the system info Task	None
	5.4 Update Picture	5.4 Update Picture	5.4.1 Update Picture info Task	None
			5.4.2 Capture and Verify that the picture is in the right format Task	None
			5.4.3 Confirm picture format info Task	None
			5.4.4 Upload and save updated picture into the system info Task	None
	5.5 Delete Picture	5.5 Delete Picture	5.5.1 Delete picture info Task	None
			5.5.2 Confirm delete picture info Task	None
			5.5.3 Save delete picture successful info Task	None
	5.6 View invitation	5.6 View invitation	5.6.1 View Invitation Task	None
			5.6.2 Search Invitation Task info	Invite
			5.6.3 Accept invitation info Task	None



GROUP 23 DELIVERABLE 02



	5.7 View existing exhibitions	5.7 View existing exhibitions	5.7.1 Search exhibition	Exhibition
			5.7.2 Display selected exhibition Task	None
	5.8 Accept invitation for Exhibition	5.8 Accept invitation for Exhibition	5.8.1 Display invitation information Task	Invite, Invitation per artist
			5.8.2 Capture artist action by updating invitation status and adding artist to the exhibition Task	Invitation Status, Exhibition Artist

Conclusion

In conclusion, the above section contains the complete validation table for each requirement in the proposed system and includes the use cases, processes from the primitive level data flow diagrams and entities that satisfy the requirement.

9. Sign off by Client

Deliverable Name: System Requirements

Deliverable Number: 2

Delivered on Date: 16 May 2021

Client name: Johann van Heerden Art Gallery

Approving Client Representative: Miekie Lourens

Project Leader: Dean Botha

Deliverable Approval

Signing off this document signifies that the client is satisfied with the deliverables listed above. The client acknowledges the accuracy of all aspects related to the Gallery. The client is also satisfied in the way we approached the proposed system and how we have worked together with the client in a respectful and efficient manner.

Client Signature



Project Leader Signature



Signed on: 10 April 2021

10. Complexity

Topic	Level		Marks	MAX
1. Special GUI	For online applications: Responsive web design For desktop applications: Form design according to design principles (Schneiderman's golden rule on navigation applies here)	*	3	40
	Appropriate use of grids/tables	✓	3	
	Appropriate use of tabs/links	✓	3	
	Use of graphs in an appropriate business context	✓	4	
	The storage and display of graphical information, like photos with a good business reason	✓	3	
	Working e-mail automatically generated from the database in an appropriate business context	✓	2	
	Extensive user-friendly search facility	✓	3	
	At least one use of a tree to display data from the database	✓	3	
	Able to dynamically modify a data tree structure and in doing so adjusting the data in the database	✓	4	
	At least one use of a calendar view of data (not a date/time picker; not a plug-in such as Google calendar)	✓	3	
	Uploading a file into the system with appropriate business reason	✓	3	
	The use of audio/video in an appropriate business context	✓	3	
	At least one use of an administrator configurable timer in an appropriate business context	✓	3	
2. Database access	At least 30 tables used (4 member groups) or 40. tables used (5 member groups)	*	6	15
	Full referential integrity on all tables	*	6	
	At least one use of master-detail table relationships (Schneiderman's golden rule on system status applies here)	*	3	
3. Reports	At least 3 simple list reports in a reporting tool (no control breaks, no graphs, single table)	*	3	15
	At least 2 transactional report with 2 or more control breaks (with heading and calculated values/totals, multiple tables)	*	6	
	At least 1 report with adjustable criteria	✓	3	
	At least 1 management report using a graph	✓	3	
4. Flexibility	All data that can change in future should not be hard coded but maintained in a sub-module of the system (e.g. Lookup tables)	✓	6	12
	Some business rules are not hard coded but maintained in a sub-module of the system.	✓	6	
5. Error handling	All system-generated errors are trapped, and consistent, user-friendly error messages are displayed	✓	6	12
	Appropriate data validation on all input fields	✓	6	
	At least one menu item or other control that opens up a complete help document (HTML, PDF, Help-file)	✓	3	

6. Help	Extensive context-sensitive help. E.g. calling Help on a specific screen/function will automatically open the specific help for that screen/function.	✓	6	15
Topic	Level		Marks	MAX
	Search Facility on Help	✓	3	
	Extensive use of hints	✓	3	
7. Security	Logon screen with user ID and password and fixed user profiles	✓	3	13
	Applying two factor authentication with applicable business reason.	✓	3	
	Encrypted passwords in database	*	1	
	Flexible user profiles (i.e. you can dynamically add user profiles that will enable/disable access to certain parts of the system)	✓	6	
8. Deployment	For an online application: Deployment of application to a publicly accessible web server	✓	3	6
	Deployment of the database to a remote database server	✓	3	
9. Backup and Restore	A backup and restore subsystem exists that backup/restore all data (system may exit during restore)	✓	3	3
10. Import/Export Data	Able to open Word or Excel and automatically place data in it based on the parameters provided (with a good business reason)	✓	6	6
11. External APPLICATION / Services	Integrate an existing web service into your application (with good business reason)	✓	3	3

Topic	Level		Marks	MAX
12. Multiplatform processing for an appropriate business reason	Appropriate business use of static views on an alternative platform.	✓	3	18
	Appropriate use of dynamic views on an alternative platform (i.e. data is displayed from the system's database)	✓	3	
	Appropriate use of substantial dynamic views on an alternative platform (i.e. both reading and writing data from the system's database)	✓	9	
	Uploading a file through an alternative platform onto the system's database.	✓	3	
15. Programming Principles	The use of a data layer to facilitate interaction between your database and your business layer	✓	3	12
	The use of an API to facilitate interaction between your business layer and your presentation layer	✓	6	
	Comprehensive use of stored procedures and/or triggers and/or jobs.	✓	3	

Total Complexity Marks

170

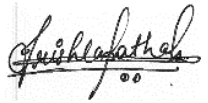
11. General

Introduction

This section proves that the team members stated below has read and agrees with the information stated within this document. By signing, each member gives proof that he/she is satisfied with the contents of this document and that he/she has contributed a fair share of content to this document.

11.1 Team Sign-Off

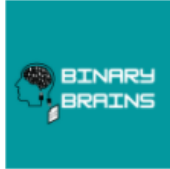
I,

Full name	Student number	Date	Signature
Dean Botha	u18002197	5/16/2021	
Trishla Pathak	u18198644	5/16/2021	
Reagile Matloga	u18201696	5/16/2021	
Gamuchirai Dambanjera	u17272522	5/16/2021	
Reinhardt Smit	u18131451	5/16/2021	

, declare and agree that I am satisfied and contributed to the content of this deliverable and produced work of a professional matter and high standards.

Conclusion

In conclusion, this section consists of the signatures of each team member as a sign of their recognition and approval regarding the content of this deliverable.



12. Document Conclusion

This concludes the second deliverable of the proposed system. This deliverable aimed to provide a broad overview of the functional specifications of each use case, along with descriptions, data models and data flow diagrams. The document is finalised with the client sign-off, group sign-off and the complexity matrix