























**Team members:** (Top Left to right) Cailín Smith, Amogelang P Moloko, Christopher J Oakes  
(Bottom Left to Right) Achal Seechoonparsad, Jacqueline L Lawler

## Team Information

 15022014	 Jacqueline L Lawler	 u15022014@tuks.co.za	 0797581409
 15213626	 Christopher J Oakes	 u15213626@tuks.co.za	 0834074027
 14284783	 Amogelang P Moloko	 u14284783@tuks.co.za	 0741020360
 15035892	 Cailín Smith [Project Leader]	 u15035892@tuks.co.za	 0769363737
 15278043	 Achal Seechoonparsad	 u15278043@tuks.co.za	 0718894149

## Functional Specification

The purpose of this document is to give the reader a clear overview of all the functional elements of the proposed system. The functional specification includes use case diagrams and use case narratives to completely describe functional requirements. It also contains a full set of process models, UML models and data models to graphically depict all the functional components of the system. Later in the description, the interface and reports of the proposed system are described, after which the requirements are validated against the functional specification.



# Contents

Table of Figures .....	iv
1. Introduction .....	1
2. Use Cases .....	3
2.1 Introduction .....	3
2.2 Use Case Diagrams.....	4
2.3 Use Case Narratives .....	14
2.3.1 User Subsystem Narratives.....	14
2.3.2 Volunteer Subsystem Narratives .....	23
2.3.3 Student Subsystem Narratives.....	42
2.3.4 Event Subsystem Narratives .....	61
2.3.5 Function Subsystem Narratives .....	99
2.3.6 Community Engagement Subsystem Narratives .....	115
2.3.7 Lecture Subsystem Narratives .....	127
2.3.8 Marketing Subsystem Narratives.....	139
2.3.9 Report Subsystem Narratives .....	146
2.3.10 Donation Subsystem Narratives .....	160
2.4 Conclusion.....	164
3. Process Models .....	165
3.1 Introduction .....	165
3.2 Context Diagram .....	165
3.3 Functional Decomposition Diagram .....	175
3.4 Data Flow Diagrams .....	185
3.4.1 High-Level Data Flow Diagrams .....	185
3.4.2 Mid-Level Data Flow Diagrams .....	195
3.4.3 Primitive-Level Data Flow Diagrams .....	217
3.5 Conclusion.....	278
4. UML Modelling.....	279
4.1 Introduction .....	279
4.2 Activity Diagrams .....	280

4.3 Conclusion .....	341
5. Data Model.....	343
5.1 Introduction .....	343
5.2 Entity Relationship Diagram .....	344
5.3 Conclusion .....	345
6. Interfaces and Other Inputs .....	347
6.1 Introduction .....	347
6.2 Input Interface Description .....	348
6.2.1 User Input Interface Description .....	348
6.2.2 Volunteer Input Interface Description.....	349
6.2.3 Student Input Interface Description .....	352
6.2.4 Events Input Interface Description .....	356
6.2.5 Function Input Interface Description .....	362
6.2.6 Community Engagement Input Interface Description.....	365
6.2.7 Lecture Input Interface Description .....	368
6.2.8 Marketing Input Interface Description .....	371
6.2.9 Reports Input Interface Description .....	372
6.2.10 Donation Input Interface Description .....	374
6.3 Conclusion .....	374
7. Reports and Other Outputs .....	377
7.1 Introduction .....	377
7.2 Output Description .....	377
7.2.1 User Output Description .....	377
7.2.2 Volunteer Output Description .....	379
7.2.3 Student Output Description.....	382
7.2.4 Events Output Description .....	386
7.2.5 Function Output Description .....	393
7.2.6 Community Engagement Output Description .....	397
7.2.7 Lecture Output Description .....	400
7.2.8 Marketing Output Description .....	403
7.2.9 Reports Output Description .....	405

7.2.10 Donation Output Description .....	407
7.3 Conclusion .....	407
8. Validation .....	409
8.1 Introduction .....	409
8.2 Validation against Requirements.....	409
8.2.1 User Subsystem.....	409
8.2.2 Volunteer Subsystem .....	412
8.2.3 Student Subsystem .....	418
8.2.4 Event Subsystem .....	424
8.2.5 Function Subsystem .....	432
8.2.6 Community Engagement Subsystem .....	436
8.2.7 Lecture Content Subsystem .....	441
8.2.8 Marketing Subsystem .....	445
8.2.9 Report Subsystem .....	447
8.2.10 Donation Subsystem .....	450
8.3 Conclusion .....	450
9. Conclusion .....	451
10. Complexity .....	453
11. Sign-off by Client .....	461
12. Sign-off by Team .....	462

## Table of Figures

Figure 1- Use Case Diagram 1. User Subsystem .....	4
Figure 2- Use Case Diagram 2. Personnel Subsystem.....	5
Figure 3- Use Case Diagram 3. Student Subsystem .....	7
Figure 4- Use Case Diagram 4. Manage Events Subsystem .....	7
Figure 5- Use Case Diagram 5. Function Subsystem.....	8
Figure 6- Use Case Diagram 6. Community Engagement Subsystem.....	9
Figure 7- Use Case Diagram 7. Lecture Subsystem.....	10
Figure 8- Use Case Diagram 8. Marketing Subsystem .....	11
Figure 9- Use Case Diagram 9. Reporting Subsystem .....	12
Figure 10- Use Case Diagram 10. Donation Subsystem.....	13
Figure 11- Context Diagram User Subsystem .....	165
Figure 12- Context Diagram Volunteer Subsystem .....	166
Figure 13- Context Diagram Student Subsystem.....	167
Figure 14- Context Diagram Event Subsystem .....	168
Figure 15- Context Diagram Function Subsystem .....	169
Figure 16- Context Diagram Community Engagement Subsystem .....	170
Figure 17- Context Diagram Lecture Subsystem .....	171
Figure 18- Context Diagram Marketing Subsystem.....	172
Figure 19- Context Diagram Report Subsystem.....	173
Figure 20- Context Diagram Donation Subsystem.....	174
Figure 21- Decomposition Diagram Level 0.....	175
Figure 22- Decomposition Diagram User Function 1.....	175
Figure 23- Decomposition Diagram Volunteer Function 2 .....	176
Figure 24- Decomposition Diagram Student Function 3 .....	177
Figure 25- Decomposition Diagram Manage Events Function 4 .....	178
Figure 26- Decomposition Diagram Function Function 5 .....	179
Figure 27- Decomposition Diagram Community Engagement Function 6 .....	180
Figure 28- Decomposition Diagram Lecture Function 7 .....	181
Figure 29- Decomposition Diagram Function 8 .....	182
Figure 30- Decomposition Diagram Report Function 9 .....	183
Figure 31- Decomposition Diagram Donation Function 10 .....	184
Figure 32- High-Level Data Flow Diagram 1. User Subsystem.....	185
Figure 33- High-Level Data Flow Diagram 2. Volunteer Subsystem .....	186
Figure 34- High-Level Data Flow Diagram 3. Student Subsystem .....	187
Figure 35- High-Level Data Flow Diagram 4. Event Subsystem .....	188
Figure 36- High-Level Data Flow Diagram 5. Function Subsystem .....	189
Figure 37- High-Level Data Flow Diagram 6. Community Engagement.....	190
Figure 38- High-Level Data Flow Diagram 7. Lecture Subsystem .....	191
Figure 39- High-Level Data Flow Diagram 8. Marketing Subsystem.....	192

Figure 40- High-Level Data Flow Diagram 9. Reporting Subsystem .....	193
Figure 41- High-Level Data Flow Diagram 10. Donation Subsystem .....	194
Figure 42- Mid-Level Data Flow Diagram 1.1 Check Forgotten Password .....	195
Figure 43- Mid-Level Data Flow Diagram 1.2 Forgotten Password .....	195
Figure 44- Mid-Level Data Flow Diagram 1.3 Login .....	196
Figure 45- Mid-Level Data Flow Diagram 1.4 Abstract Register User .....	196
Figure 46- Mid-Level Data Flow Diagram 2.1 Register Volunteer .....	197
Figure 47- Mid-Level Data Flow Diagram 2.2 Search Volunteer.....	197
Figure 48- Mid-Level Data Flow Diagram 2.3 Update Volunteer.....	197
Figure 49- Mid-Level Data Flow Diagram 2.4 Delete Volunteer .....	198
Figure 50- Mid-Level Data Flow Diagram 2.5 Add Volunteer Type .....	198
Figure 51- Mid-Level Data Flow Diagram 2.6 Search Volunteer Type .....	198
Figure 52- Mid-Level Data Flow Diagram 2.7 Update Volunteer Type .....	199
Figure 53- Mid-Level Data Flow Diagram 2.8 Delete Volunteer Type .....	199
Figure 54- Mid-Level Data Flow Diagram 2.9 Generate Unique Code .....	199
Figure 55- Mid-Level Data Flow Diagram 3.1 Register Student.....	200
Figure 56- Mid-Level Data Flow Diagram 3.2 Search Student .....	200
Figure 57- Mid-Level Data Flow Diagram 3.3 Update Student .....	200
Figure 58- Mid-Level Data Flow Diagram 3.4 Delete Student .....	201
Figure 59- Mid-Level Data Flow Diagram 3.5 Generate Graduate List.....	201
Figure 60- Mid-Level Data Flow Diagram 3.6 Add Student Type.....	201
Figure 61- Mid-Level Data Flow Diagram 3.7 Search Student Type .....	202
Figure 62- Mid-Level Data Flow Diagram 3.8 Update Student Type .....	202
Figure 63- Mid-Level Data Flow Diagram 3.9 Delete Student Type .....	202
Figure 64- Mid-Level Data Flow Diagram 4.1 Create Venue .....	203
Figure 65- Mid-Level Data Flow Diagram 4.2 Search Venue .....	203
Figure 66- Mid-Level Data Flow Diagram 4.3 Update Venue .....	203
Figure 67- Mid-Level Data Flow Diagram 4.4 Delete Venue.....	204
Figure 68- Mid-Level Data Flow Diagram 4.5 Create Event.....	204
Figure 69- Mid-Level Data Flow Diagram 4.6 Search Event.....	204
Figure 70- Mid-Level Data Flow Diagram 4.7 RSVP to an Event.....	205
Figure 71- Mid-Level Data Flow Diagram 4.8 View TRWLA Timeline .....	205
Figure 72- Mid-Level Data Flow Diagram 4.9 Update Event Information .....	205
Figure 73- Mid-Level Data Flow Diagram 4.10 Cancel Event.....	206
Figure 74- Mid-Level Data Flow Diagram 4.11 Log Event Attendance .....	206
Figure 75- Mid-Level Data Flow Diagram 4.12 Send Notification.....	206
Figure 76- Mid-Level Data Flow Diagram 5.1 Register Guest Speaker .....	207
Figure 77- Mid-Level Data Flow Diagram 5.2 Search Guest Speaker .....	207
Figure 78- Mid-Level Data Flow Diagram 5.3 Update Guest Speaker .....	207
Figure 79- Mid-Level Data Flow Diagram 5.4 Delete Guest Speaker.....	208
Figure 80- Mid-Level Data Flow Diagram 5.5 Invite Guest Speaker .....	208

Figure 81- Mid-Level Data Flow Diagram 5.6 Confirm Guest Speaker .....	208
Figure 82- Mid-Level Data Flow Diagram 6.1 Upload Community Engagement Content .....	209
Figure 83- Mid-Level Data Flow Diagram 6.2 Search Community Engagement Content .....	209
Figure 84- Mid-Level Data Flow Diagram 6.3 Delete Community Engagement Content .....	210
Figure 85- Mid-Level Data Flow Diagram 6.4 Lock Community Engagement Content .....	210
Figure 86- Mid-Level Data Flow Diagram 6.5 Unlock Community Engagement Content .....	210
Figure 87- Mid-Level Data Flow Diagram 7.1 Upload Lecture Content .....	211
Figure 88- Mid-Level Data Flow Diagram 7.2 Search Lecture Content .....	211
Figure 89- Mid-Level Data Flow Diagram 7.3 Delete Lecture Content .....	211
Figure 90- Mid-Level Data Flow Diagram 7.4 Lock Lecture Content .....	211
Figure 91- Mid-Level Data Flow Diagram 7.5 Unlock Lecture Content .....	212
Figure 92- Mid-Level Data Flow Diagram 7.6 Content Review .....	212
Figure 93- Mid-Level Data Flow Diagram 8.1 Upload Photo .....	212
Figure 94- Mid-Level Data Flow Diagram 8.2 Post Photo .....	213
Figure 95- Mid-Level Data Flow Diagram 8.3 Delete Photo .....	213
Figure 96- Mid-Level Data Flow Diagram 9.1 Generate Class Attendance Report .....	213
Figure 97- Mid-Level Data Flow Diagram 9.2 Generate Function Attendance Report .....	214
Figure 98- Mid-Level Data Flow Diagram 9.3 Generate Community Engagement Report .....	214
Figure 99- Mid-Level Data Flow Diagram 9.4 Generate Demographics Report .....	214
Figure 100- Mid-Level Data Flow Diagram 9.5 Generate Event Popularity Report .....	215
Figure 101- Mid-Level Data Flow Diagram 9.6 Generate Donations Report .....	215
Figure 102- Mid-Level Data Flow Diagram 9.7 Generate User Statistics Report .....	215
Figure 103- Mid-Level Data Flow Diagram 10.1 Add Donor .....	216
Figure 104- Mid-Level Data Flow Diagram 10.2 Add Donation .....	216
Figure 105- Primitive Level Data Flow Diagram 1.1 Check Forgotten Password .....	217
Figure 106- Primitive Level Data Flow Diagram 1.2 Change Password .....	218
Figure 107- Primitive Level Data Flow Diagram 1.3 Login .....	219
Figure 108- Primitive Level Data Flow Diagram AUC 1 Register User .....	220
Figure 109- Primitive Level Data Flow Diagram 2.1 Register Volunteer .....	221
Figure 110- Primitive Level Data Flow Diagram 2.2 Search Volunteer .....	222
Figure 111- Primitive Level Data Flow Diagram 2.3 Update Volunteer .....	223
Figure 112- Primitive Level Data Flow Diagram 2.4 Delete Volunteer .....	224
Figure 113- Primitive Level Data Flow Diagram 2.5 Add Volunteer Type .....	225
Figure 114- Primitive Level Data Flow Diagram 2.6 Search Volunteer Type .....	226
Figure 115- Primitive Level Data Flow Diagram 2.7 Update Volunteer Type .....	227
Figure 116- Primitive Level Data Flow Diagram 2.8 Delete Volunteer Type .....	228
Figure 117- Primitive Level Data Flow Diagram 2.9 Generate Unique Code .....	229
Figure 118- Primitive Level Data Flow Diagram 3.1 Register Student .....	230
Figure 119- Primitive Level Data Flow Diagram 3.2 Search Student .....	231
Figure 120- Primitive Level Data Flow Diagram 3.3 Update Student .....	232
Figure 121- Primitive Level Data Flow Diagram 3.4 Delete Volunteer .....	233



Figure 122- Primitive Level Data Flow Diagram 3.5 Generate Graduate List .....	234
Figure 123- Primitive Level Data Flow Diagram 3.6 Add Student Type .....	235
Figure 124- Primitive Level Data Flow Diagram 3.7 Search Student Type .....	236
Figure 125- Primitive Level Data Flow Diagram 3.8 Update Student Type .....	237
Figure 126- Primitive Level Data Flow Diagram 3.9 Delete Student Type.....	238
Figure 127- Primitive Level Data Flow Diagram 4.1 Add New Venue.....	239
Figure 128- Primitive Level Data Flow Diagram 4.2 Search Venue.....	240
Figure 129- Primitive Level Data Flow Diagram 4.3 Update Venue .....	241
Figure 130- Primitive Level Data Flow Diagram 4.4 Delete Venue.....	242
Figure 131- Primitive Level Data Flow Diagram 4.5 Create New Event.....	243
Figure 132- Primitive Level Data Flow Diagram 4.6 Search Event.....	244
Figure 133- Primitive Level Data Flow Diagram 4.7 RSVP to an Event .....	245
Figure 134- Primitive Level Data Flow Diagram 4.8 View TRWLA Timeline .....	245
Figure 135- Primitive Level Data Flow Diagram 4.9 Update Event Information .....	246
Figure 136- Primitive Level Data Flow Diagram 4.10 Cancel Event .....	247
Figure 137- Primitive Level Data Flow Diagram 4.11 Log Event Attendance .....	248
Figure 138- Primitive Level Data Flow Diagram 4.12 Send Notification.....	249
Figure 139- Primitive Level Data Flow Diagram 5.1 Register Guest Speaker .....	250
Figure 140- Primitive Level Data Flow Diagram 5.2 Search Guest Speaker.....	251
Figure 141- Primitive Level Data Flow Diagram 5.3 Update Guest Speaker .....	252
Figure 142- Primitive Level Data Flow Diagram 5.4 Delete Guest Speaker.....	253
Figure 143- Primitive Level Data Flow Diagram 5.5 Invite Guest Speaker .....	254
Figure 144- Primitive Level Data Flow Diagram 5.6 Confirm Guest Speaker .....	255
Figure 145- Primitive Level Data Flow Diagram 6.1 Upload Community Engagement Content	256
Figure 146- Primitive Level Data Flow Diagram 6.2 Search Community Engagement Content.	257
Figure 147- Primitive Level Data Flow Diagram 6.3 Delete Community Engagement Content .	258
Figure 148- Primitive Level Data Flow Diagram 6.4 Lock Community Engagement Content ....	259
Figure 149- Primitive Level Data Flow Diagram 6.5 Unlock Community Engagement Content.	260
Figure 150- Primitive Level Data Flow Diagram 7.1 Upload Lecture Content .....	261
Figure 151- Primitive Level Data Flow Diagram 7.2 Upload Lecture Content .....	262
Figure 152- Primitive Level Data Flow Diagram 7.3 Delete Lecture Content .....	263
Figure 153- Primitive Level Data Flow Diagram 7.4 Lock Lecture Content .....	264
Figure 154- Primitive Level Data Flow Diagram 7.5 Unlock Lecture Content .....	265
Figure 155- Primitive Level Data Flow Diagram 7.6 Content Review .....	266
Figure 156- Primitive Level Data Flow Diagram 8.1 Upload Photo .....	267
Figure 157- Primitive Level Data Flow Diagram 8.2 Post Photo .....	268
Figure 158- Primitive Level Data Flow Diagram 8.3 Delete Photo .....	269
Figure 159- Primitive Level Data Flow Diagram 9.1 Generate Class Attendance Report.....	270
Figure 160- Primitive Level Data Flow Diagram 9.2 Generate Function Attendance Report.....	271
Figure 161- Primitive Level Data Flow Diagram 9.3 Generate CommEngagement Attendance Report .....	272

Figure 162- Primitive Level Data Flow Diagram 9.4 Generate Demographic Report .....	273
Figure 163- Primitive Level Data Flow Diagram 9.5 Generate Event Popularity Report .....	274
Figure 164- Primitive Level Data Flow Diagram 9.6 Generate Donation Report.....	275
Figure 165- Primitive Level Data Flow Diagram 9.7 Generate User Statistics Report .....	276
Figure 166- Primitive Level Data Flow Diagram 10.1 Add Donor .....	277
Figure 167- Primitive Level Data Flow Diagram 10.2 Receive Donation .....	278
Figure 168- Activity Diagram 1.1 Check Forgotten Password .....	280
Figure 169- Activity Diagram 1.2 Change Password .....	281
Figure 170- Activity Diagram 1.3 Login .....	282
Figure 171- Activity Diagram 2.1 Register Volunteer .....	283
Figure 172- Activity Diagram 2.2 Search Volunteer.....	284
Figure 173- Activity Diagram 2.3 Update Volunteer.....	285
Figure 174- Activity Diagram 2.4 Delete Volunteer .....	286
Figure 175- Activity Diagram 2.5 Add Volunteer Type .....	287
Figure 176- Activity Diagram 2.6 Search Volunteer Type .....	288
Figure 177- Activity Diagram 2.7 Update Volunteer Type.....	289
Figure 178- Activity Diagram 2.8 Delete Volunteer Type .....	290
Figure 179- Activity Diagram 2.9 Generate Unique Code.....	291
Figure 180- Activity Diagram 3.1 Register Student.....	292
Figure 181- Activity Diagram 3.2 Search Student .....	293
Figure 182- Activity Diagram 3.3 Update Student .....	294
Figure 183- Activity Diagram 3.4 Delete Student .....	295
Figure 184- Activity Diagram 3.5 Generate Graduate List .....	296
Figure 185- Activity Diagram 3.6 Add Student Type .....	297
Figure 186- Activity Diagram 3.7 Search Student Type .....	298
Figure 187- Activity Diagram 3.8 Update Student Type .....	299
Figure 188- Activity Diagram 3.9 Delete Student Type.....	300
Figure 189- Activity Diagram 4.1 Add New Venue .....	301
Figure 190- Activity Diagram 4.2 Search Venue .....	302
Figure 191- Activity Diagram 4.3 Update Venue .....	303
Figure 192- Activity Diagram 4.4 Delete Venue.....	304
Figure 193- Activity Diagram 4.5 Create New Event.....	305
Figure 194- Activity Diagram 4.6 Search Event.....	306
Figure 195- Activity Diagram 4.7 RSVP to an Event .....	307
Figure 196- Activity Diagram 4.8 View TRWLA Timeline .....	308
Figure 197- Activity Diagram 4.9 Update Event Information .....	309
Figure 198- Activity Diagram 4.10 Cancel Event.....	310
Figure 199- Activity Diagram 4.11 Log Event Attendance .....	311
Figure 200- Activity Diagram 4.12 Send Notification.....	312
Figure 201- Activity Diagram 5.1 Register Guest Speaker .....	313
Figure 202- Activity Diagram 5.2 Search Guest Speaker .....	314

Figure 203- Activity Diagram 5.3 Update Guest Speaker .....	315
Figure 204- Activity Diagram 5.4 Delete Guest Speaker .....	316
Figure 205- Activity Diagram 5.5 Invite Guest Speaker .....	317
Figure 206- Activity Diagram 5.6 Confirm Guest Speaker .....	318
Figure 207- Activity Diagram 6.1 Upload Community Engagement Content .....	319
Figure 208- Activity Diagram 6.2 Search Community Engagement Content.....	320
Figure 209- Activity Diagram 6.3 Delete Community Engagement Content .....	321
Figure 210- Activity Diagram 6.4 Lock Community Engagement Content .....	322
Figure 211- Activity Diagram 6.5 Unlock Community Engagement Content .....	323
Figure 212- Activity Diagram 7.1 Upload Lecture Content .....	324
Figure 213- Activity Diagram 7.2 Search Lecture Content.....	325
Figure 214- Activity Diagram 7.3 Delete Lecture Content .....	326
Figure 215- Activity Diagram 7.4 Lock Lecture Content .....	327
Figure 216- Activity Diagram 7.5 Unlock Lecture Content .....	328
Figure 217- Activity Diagram 8.1 Upload Photo .....	329
Figure 218- Activity Diagram 8.2 Post Photo .....	330
Figure 219- Activity Diagram 8.3 Delete Photo .....	331
Figure 220- Activity Diagram 9.1 Generate Class Attendance Report.....	332
Figure 221- Activity Diagram 9.2 Generate Function Attendance Report.....	333
Figure 222- Activity Diagram 9.3 Generate Community Outreach Attendance Report.....	334
Figure 223- Activity Diagram 9.4 Generate Demographic Report.....	335
Figure 224- Activity Diagram 9.5 Generate Event Popularity .....	336
Figure 225- Activity Diagram 9.6 Generate Donations Report .....	337
Figure 226- Activity Diagram 9.7 Generate User Statistics Report.....	338
Figure 227- Activity Diagram 10.1 Add Donor .....	339
Figure 228- Activity Diagram 10.2 Receive Donation .....	340
Figure 229- Entity Relationship Diagram .....	344



# 1. Introduction

1.1 This document entails the functional specification for the TuksRes Women in Leadership Academy system. In the first deliverable, the business problem was addressed whereby the group identified that the organization was in dire need of automating most of their business processes. This document is a continuation of the project proposal whereby the group proposes a solution for the organization. This document addresses all the functional requirements and specifications that need to be met in to create the TRWLA system. The first section describes the to-be system which is illustrated using use case diagrams. Each use case from the use case diagram is then discussed and explained in the use case narratives. The process models are then illustrated and explained to describe the proposed system. Process models included are the context diagram, a complete functional decomposition diagram and a full set of high-level, mid-level and primitive-level data flow diagrams. UML modelling is then illustrated using activity diagrams which are depicted from the process models. The data modelling is then illustrated using an entity-relationship diagram. The full input interface descriptions and the complete output descriptions from the context diagram are then explained in tables. A validation table is then presented whereby all the details of the functional specification are validated against the requirements. The document is then concluded by the sign-off by each of the members, as well as the client.



## 2. Use Cases

### 2.1 Introduction

2.1.1 This section of the functional specification illustrates the functional requirements by means of a use case diagram. The use case diagram explains who interacts directly with the system and who is an external actor. The use case diagram includes the user, volunteer, student, event, function, community engagement, lecture, marketing, reports and donation subsystem. These diagrams are illustrated in figures 1 to 10. Each use case in the use case diagrams are then discussed in depth in the use case narratives to determine the steps involved in completing each use case.

## 2.2 Use Case Diagrams

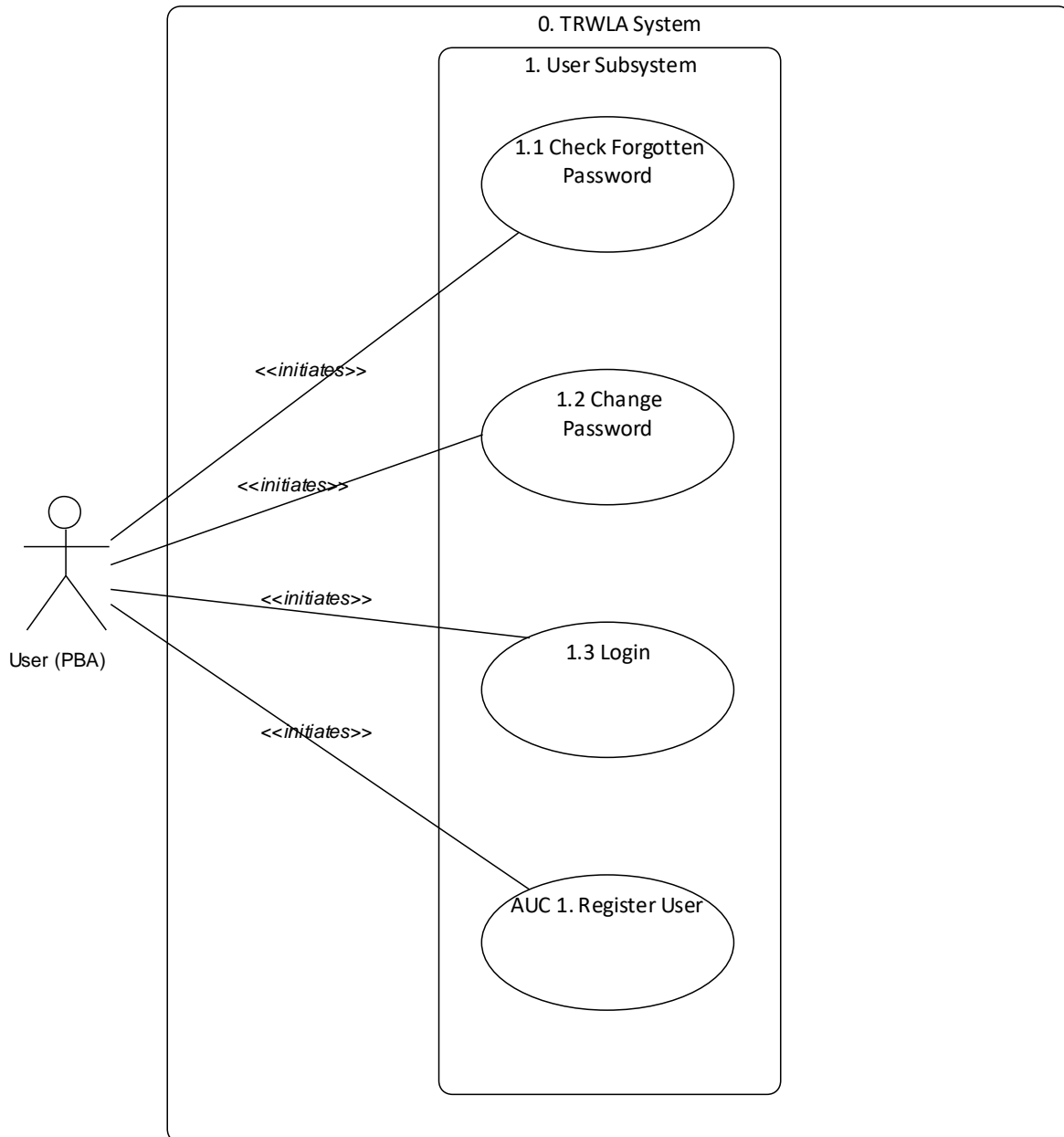


FIGURE 1- USE CASE DIAGRAM 1. USER SUBSYSTEM



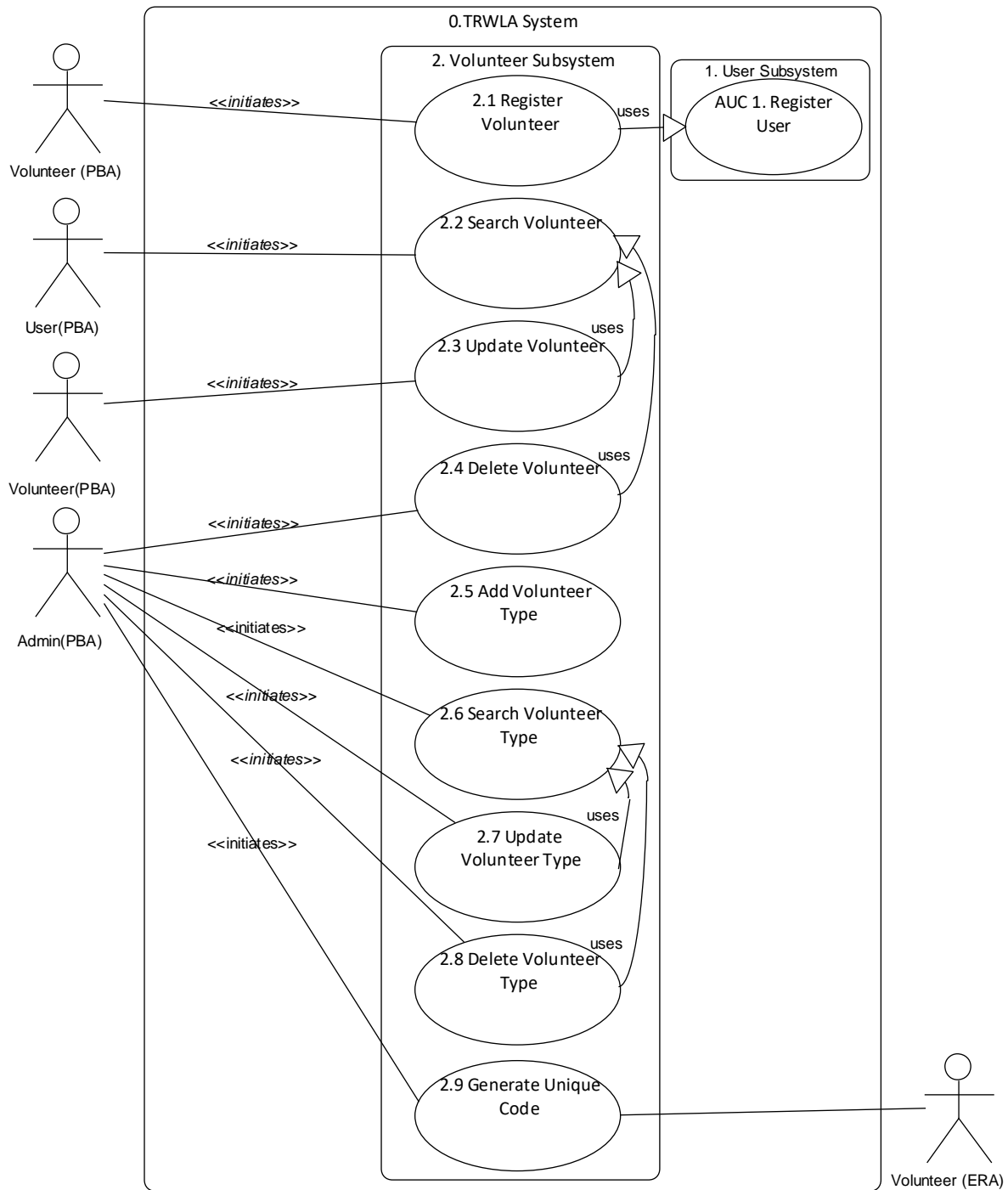


FIGURE 2- USE CASE DIAGRAM 2. PERSONNEL SUBSYSTEM

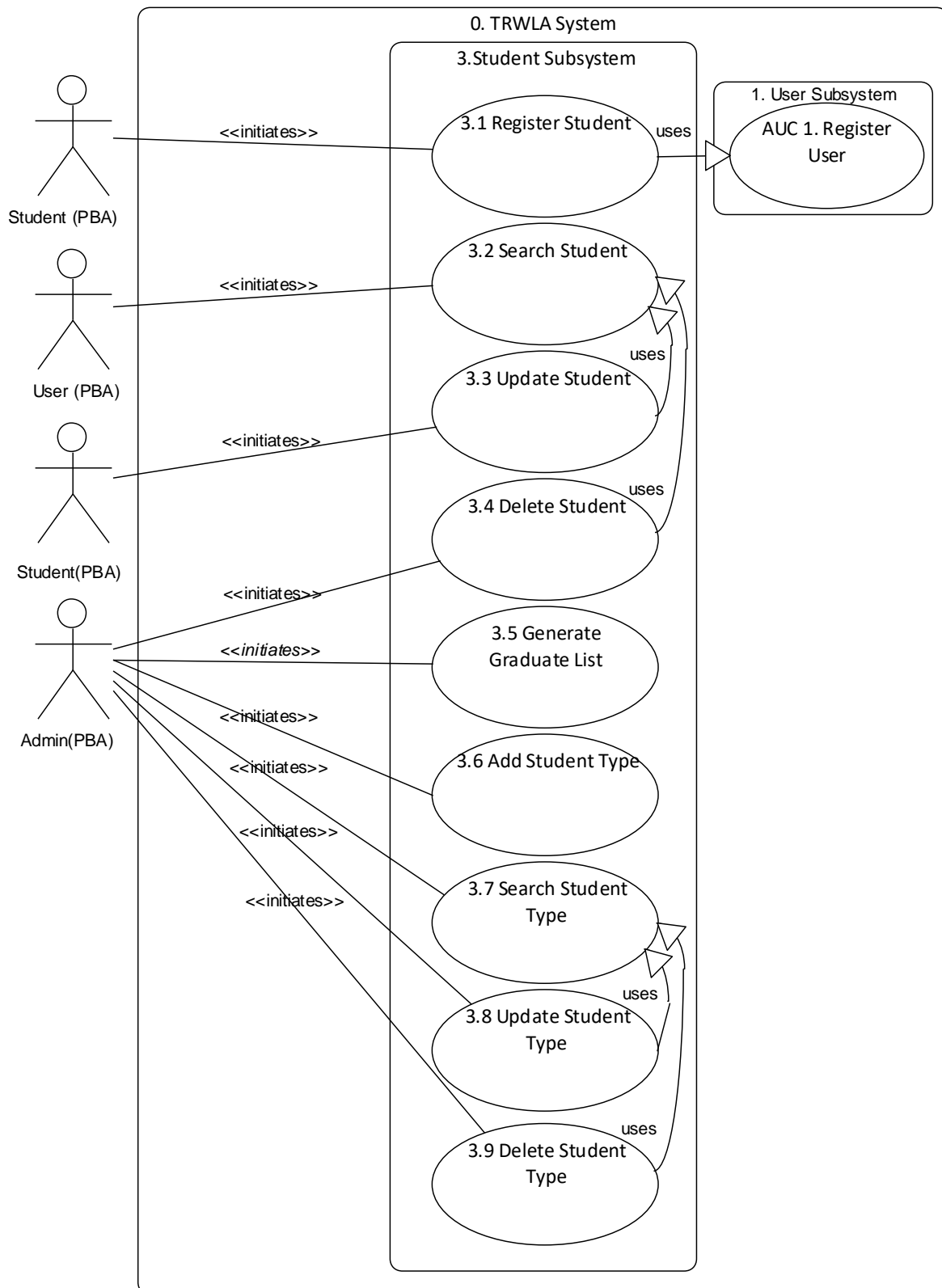


FIGURE 3- USE CASE DIAGRAM 3. STUDENT SUBSYSTEM

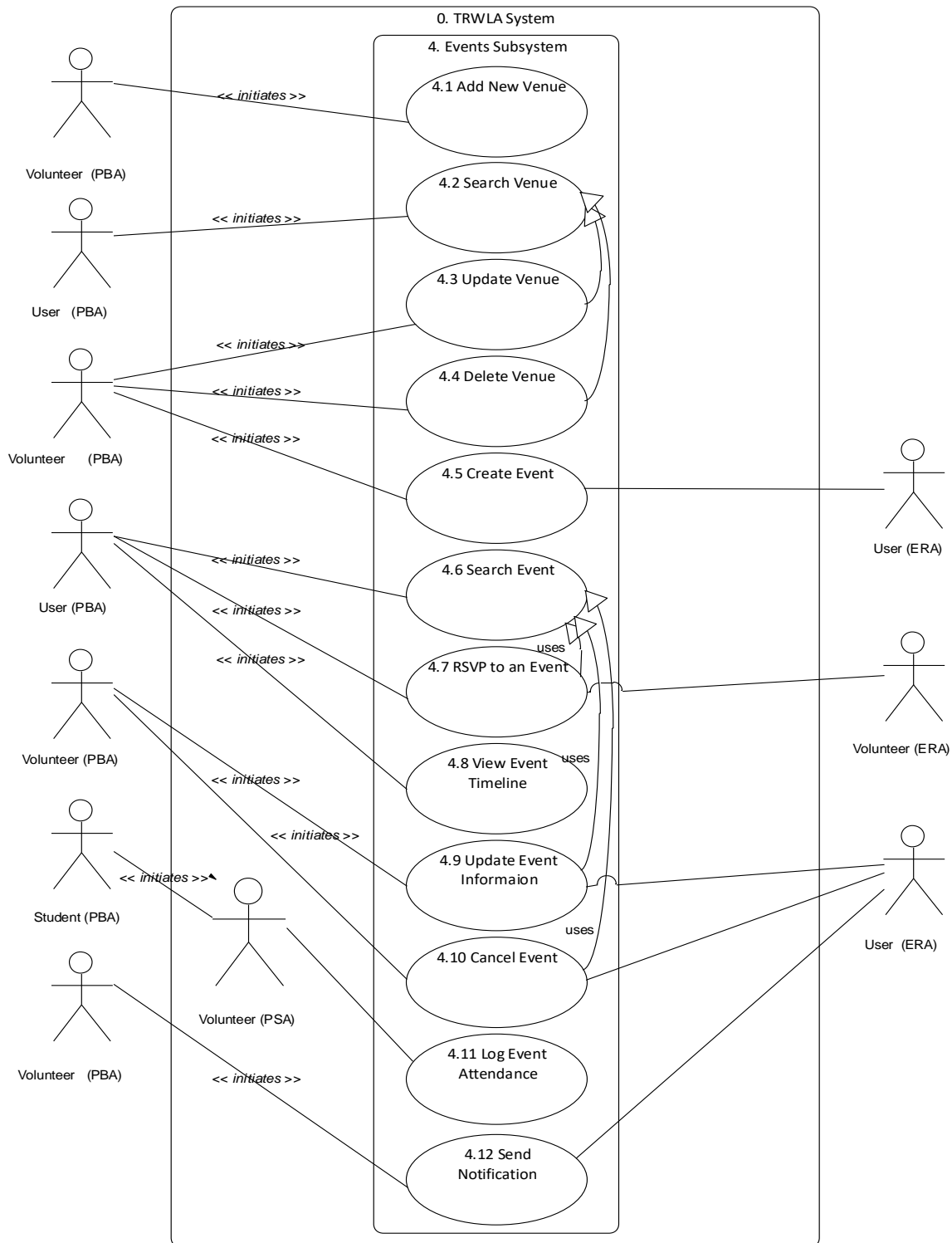


FIGURE 4- USE CASE DIAGRAM 4. MANAGE EVENTS SUBSYSTEM

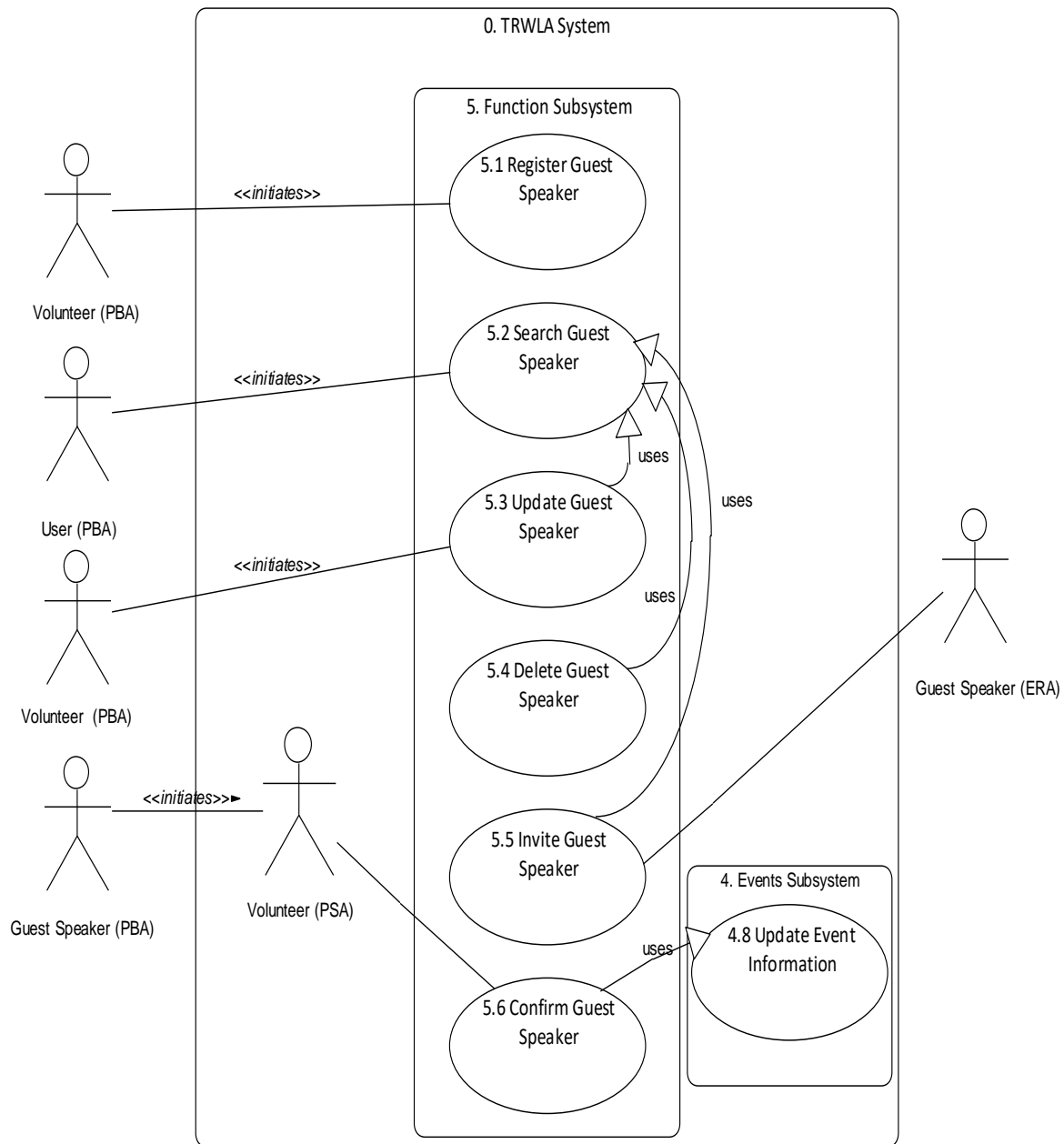


FIGURE 5- USE CASE DIAGRAM 5. FUNCTION SUBSYSTEM

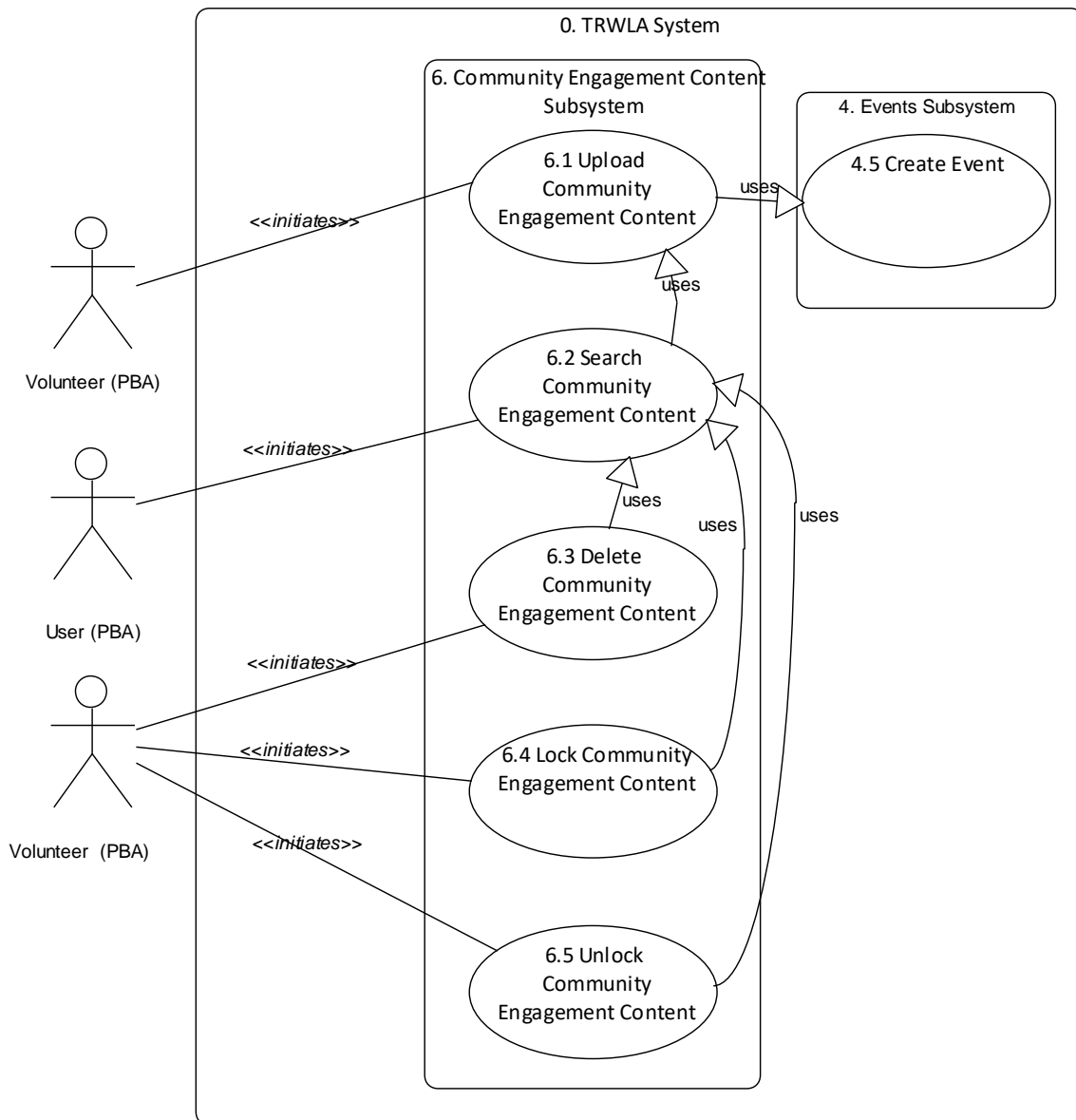


FIGURE 6- USE CASE DIAGRAM 6. COMMUNITY ENGAGEMENT SUBSYSTEM

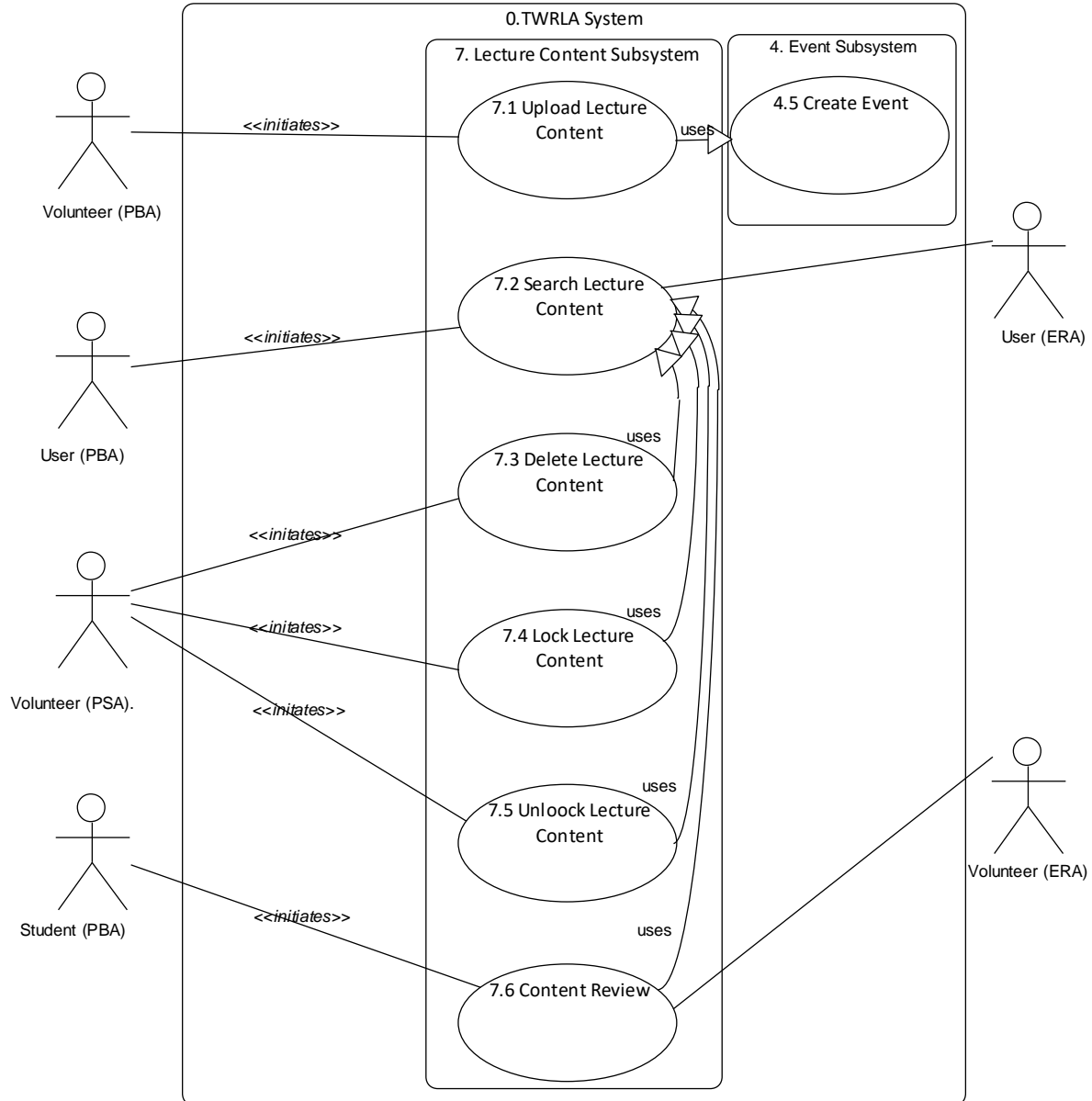


FIGURE 7- USE CASE DIAGRAM 7. LECTURE SUBSYSTEM

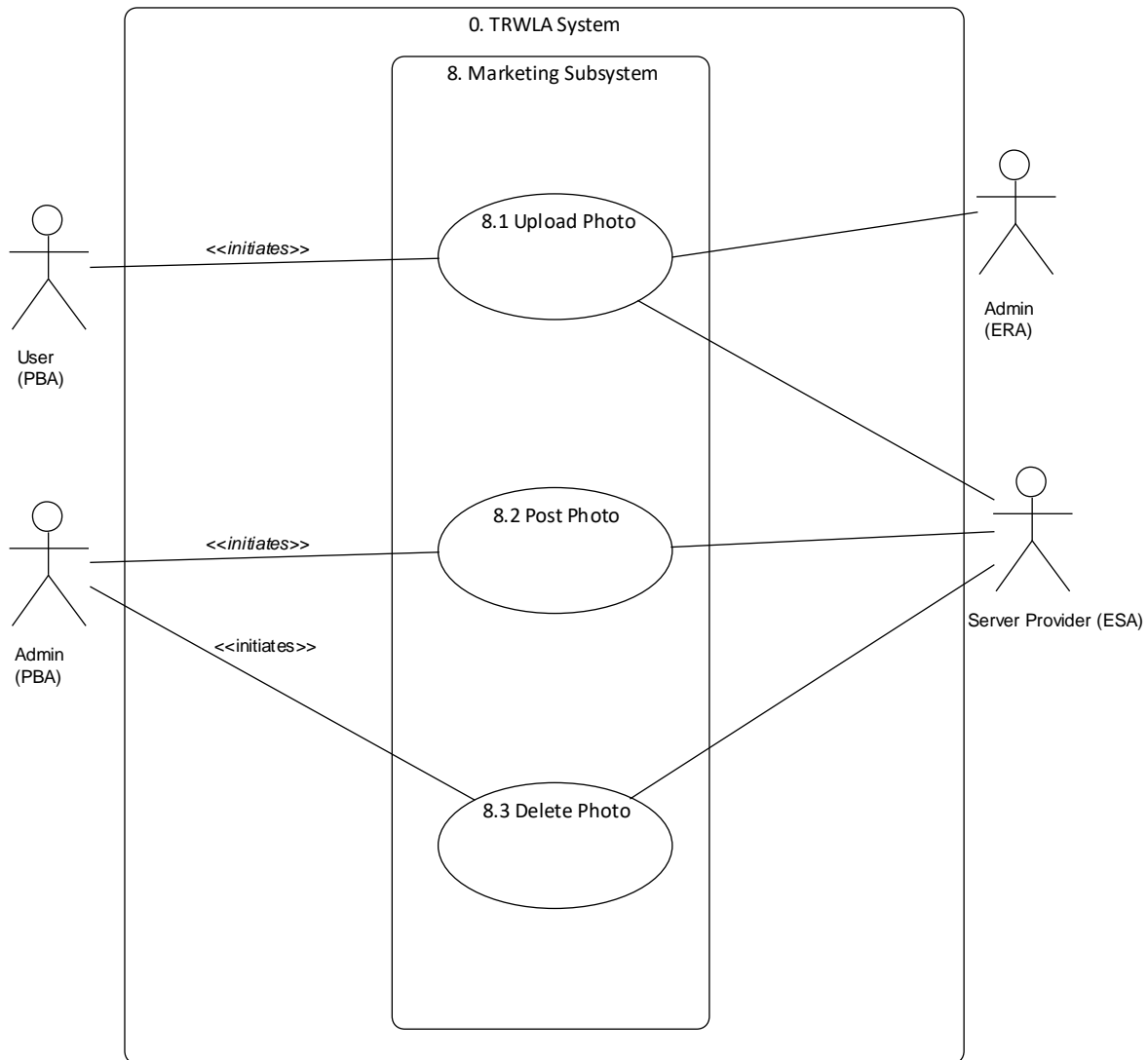


FIGURE 8- USE CASE DIAGRAM 8. MARKETING SUBSYSTEM

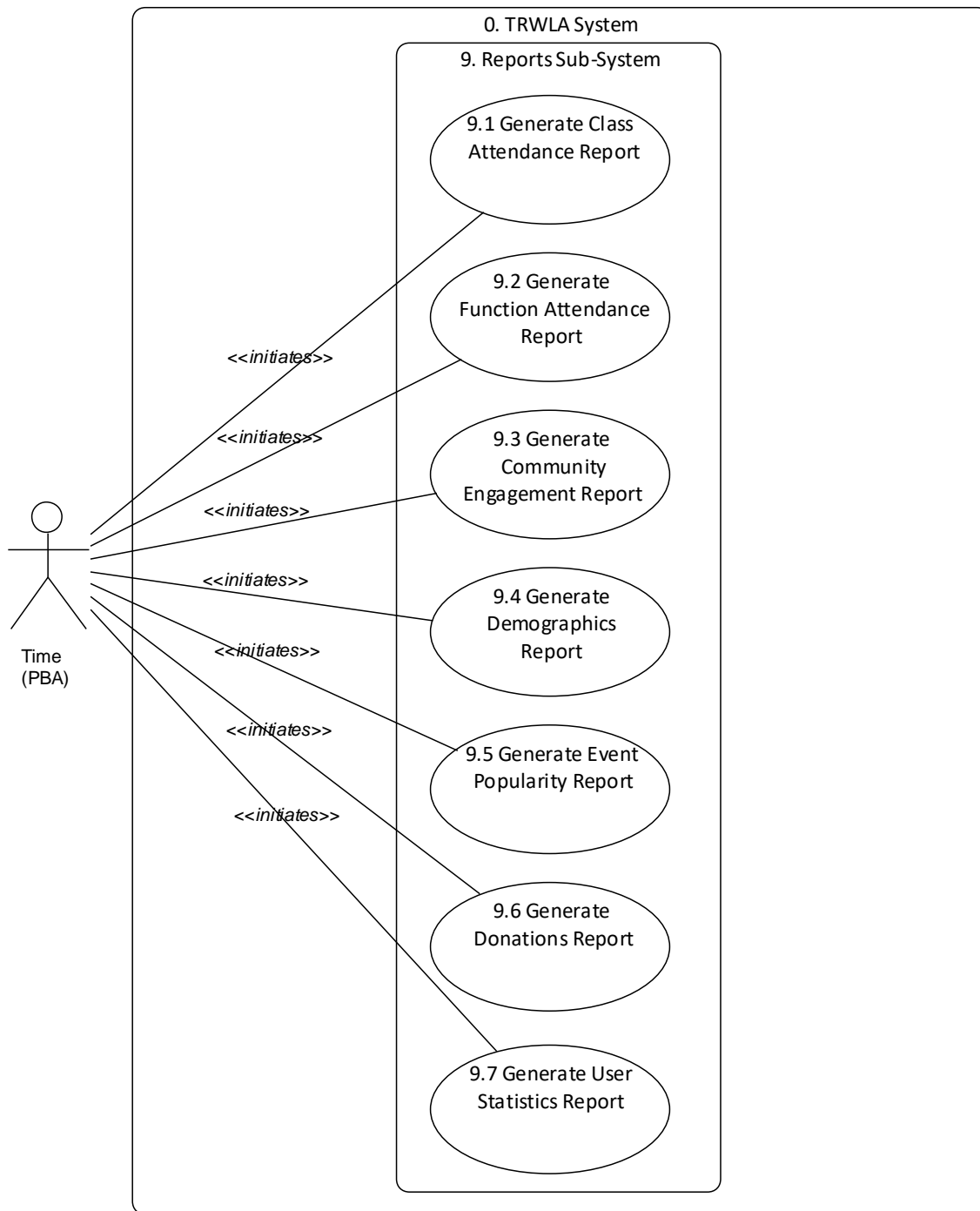


FIGURE 9- USE CASE DIAGRAM 9. REPORTING SUBSYSTEM



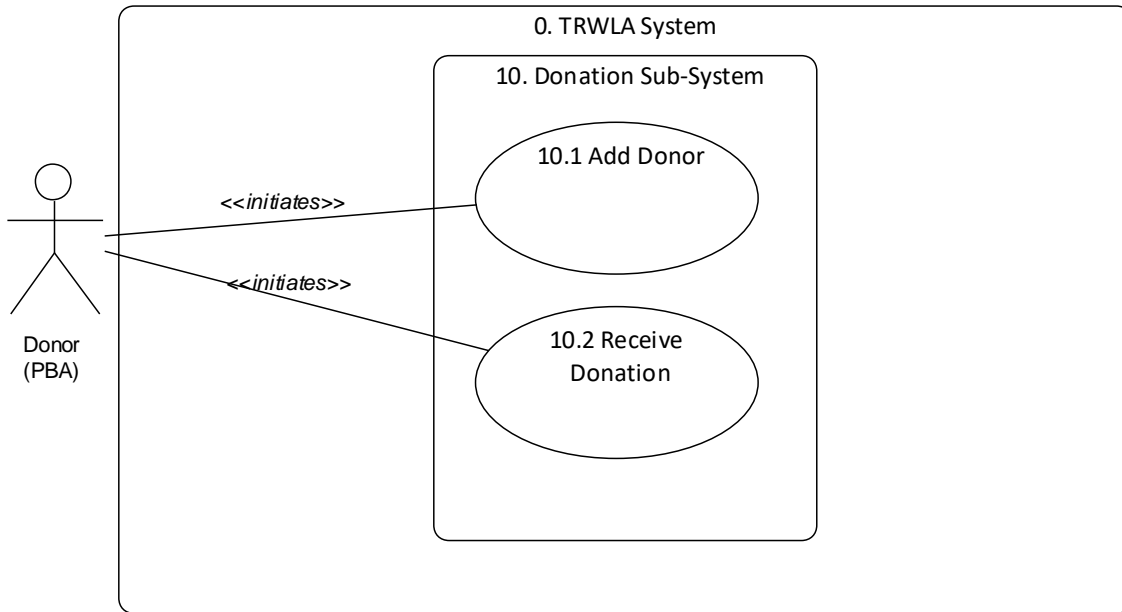


FIGURE 10- USE CASE DIAGRAM 10. DONATION SUBSYSTEM

## 2.3 Use Case Narratives

### TRWLA System

Authors: Cailin Smith, Chris Oakes, Jackie Lawler, Achal Seechoonparsing, Amo Moloko

Date: 11/04/2017

Version: 1

### 2.3.1 User Subsystem Narratives

<b>USE CASE NAME:</b>	Check Forgotten Password	<b>USE CASE TYPE</b>
<b>USE CASE ID:</b>	1.1	Business Requirements: <input type="checkbox"/>
<b>PRIORITY:</b>	Low	System Analysis: <input checked="" type="checkbox"/>
<b>SOURCE:</b>	TuksRes Women In Leadership Academy	System Design: <input type="checkbox"/>
<b>PRIMARY BUSINESS ACTOR</b>	User	
<b>PRIMARY SYSTEM ACTOR</b>	None	
<b>OTHER PARTICIPATING ACTORS:</b>	<ul style="list-style-type: none"> <li>None</li> </ul>	
<b>OTHER INTERESTED STAKEHOLDERS:</b>	<ul style="list-style-type: none"> <li>None</li> </ul>	
<b>DESCRIPTION:</b>	<p>This use case describes the event of a user checking their forgotten password. The use case begins when the user wants to log onto the system cannot remember their password. The user selects the option to 'Check Forgotten Password' and the system prompts the user to enter their email address. The system prompts the user to answer the security questions. If the user answers the security questions correctly, the system sends them a notification which allows them to change their password. The user enters and new password as well as a confirmation password. The use case concludes when the system updates the users' password and confirms that the password has been updated on the system.</p>	
<b>PRE-CONDITION:</b>	None	
<b>TRIGGER:</b>	A user wants to check their password that they've forgotten.	
<b>TYPICAL COURSE</b>	<b>Actor Action</b>	<b>System Response</b>
<b>OF EVENTS:</b>	<b>Step 1:</b> A user wants to check their password that they've forgotten and goes to the login section of the system.	<b>Step 2:</b> The system prompts the user with the login section, with the options to 'Login', 'Check Forgotten Password', 'Register' or 'Change password'.

	<b>Step 3:</b> The user selects the option to 'Check Forgotten Password'.	<b>Step 4:</b> The system prompts the user to enter their email address.
	<b>Step 5:</b> The user enters their email address.	<b>Step 6:</b> The system verifies that a valid email address has been entered, such that it exists as a <ul style="list-style-type: none"> <li>Email Address</li> </ul> In the <b>Person</b> table, and that it is a minimum of 7 characters, maximum of 255 characters and a full validated email address.
		<b>Step 7:</b> The system displays a security question to the user and prompts the user to enter the correct answer into the designated field.
	<b>Step 8:</b> The user enters the correct answer into the designated field.	<b>Step 9:</b> The system verifies that the answer the user has entered into the field is identical as the answer: <ul style="list-style-type: none"> <li>Answer</li> </ul> Which is retrieved from the <b>SecurityAnswer</b> table according to the users email address: <ul style="list-style-type: none"> <li>Email Address</li> </ul> From the <b>Person</b> table. The <b>Person</b> and <b>SecurityAnswer</b> tables are connected by the PersonID Key.
		<b>Step 10:</b> The system sends the user a notification which gives the user permission to access the 'Change Password' section.
	<b>Step 11:</b> The user proceeds to access the 'Change Password' section via the notification.	<b>Step 12:</b> The system prompts the user to change their password. (Invoke UC 1.2)
	<b>Step 13:</b> The user changes their password	
<b>ALTERNATE COURSES:</b>	<b>Alt-Step 7:</b> An invalid email address has been entered, the system prompts the user to enter a valid email address. → Go to <b>Step 5</b>	
	<b>Alt-Step 10:</b> An invalid security answer has been entered, the system prompts the user to answer a new security question. → Go to <b>Step 8</b>	

	<b>Alt-Step 11:</b> The user does not open the link, cannot continue to change password.
<b>CONCLUSION:</b>	The user has changed their password. The Password attribute in the <u>Person</u> table has been updated.
<b>POST-CONDITION:</b>	The user has changed their password that they forgot.
<b>BUSINESS RULES</b>	<ul style="list-style-type: none"><li>• Only people with a valid email address registered on the system and that can answer the security questions correctly may change their forgotten password.</li></ul>
<b>IMPLEMENTATION CONSTRAINTS AND SPECIFICATIONS</b>	<ul style="list-style-type: none"><li>• None</li></ul>
<b>ASSUMPTIONS:</b>	<ul style="list-style-type: none"><li>• None</li></ul>
<b>OPEN ISSUES:</b>	None

<b>USE CASE NAME:</b>	Change Password	<b>USE CASE TYPE</b>
<b>USE CASE ID:</b>	1.2	Business Requirements: <input type="checkbox"/>
<b>PRIORITY:</b>	Medium	System Analysis: <input checked="" type="checkbox"/>
<b>SOURCE:</b>	TuksRes Women In Leadership Academy	System Design: <input type="checkbox"/>
<b>PRIMARY BUSINESS ACTOR</b>	User	
<b>PRIMARY SYSTEM ACTOR</b>	None	
<b>OTHER PARTICIPATING ACTORS:</b>	<ul style="list-style-type: none"> <li>None</li> </ul>	
<b>OTHER INTERESTED STAKEHOLDERS:</b>	<ul style="list-style-type: none"> <li>None</li> </ul>	
<b>DESCRIPTION:</b>	<p>This use case describes the event of a user changing their password. The use case begins when a user wants to change their password. The user will proceed to the login section and login to the TRWLA system. The user will then go to their profile and select the option to change their password. The system prompts the user to enter their old password, a new password and a confirmation of the new password. Once the user has entered the required information, the system verifies that the password is valid, and updates it in the <b>Person</b> table. The use case concludes when the system confirms that the password has been successfully changed.</p>	
<b>PRE-CONDITION:</b>	The user must be logged in.	
<b>TRIGGER:</b>	A user wants to change their password.	
<b>TYPICAL COURSE OF EVENTS:</b>	<b>Actor Action</b>	<b>System Response</b>
	<b>Step 1:</b> A user wants to change their password and proceeds to the main menu.	<b>Step 2:</b> The system prompts the user with a main menu with an option 'My Profile'
	<b>Step 3:</b> The user selects the 'My Profile' option.	<b>Step 4:</b> The system prompts the user with their profile including an option to 'Change Password'
	<b>Step 5:</b> The user selects the option to 'Change Password'.	<b>Step 6:</b> The system prompts the user to enter the required information with the option to 'Change Password' or 'Cancel'.
	<b>Step 7:</b> The user enters the required information: <ul style="list-style-type: none"> <li>Old Password</li> <li>New password</li> <li>Confirmation Password</li> </ul>	<b>Step 8:</b> The system verifies that the information entered is valid, in that the old password matches the: <ul style="list-style-type: none"> <li>Password</li> </ul> in the <b>Person</b> table, the new password is a minimum of 8 characters, a

	and selects the option to 'Change Password'.	maximum of 35 characters and contains at least one numeric value, and that the confirmation password is identical to the new password.
		<b>Step 9:</b> The system updates the <ul style="list-style-type: none"> <li>• Password</li> </ul> In the <b>Person</b> table and confirms with a notification that the password has been successfully updated.
<b>ALTERNATE COURSES:</b>	<b>Alt-Step 8:</b> The user selects the 'Cancel' option. → Go to <b>Step 5</b>	
	<b>Alt-Step 9:</b> The information entered was not valid, the system prompts the user to enter a valid password and make sure that the confirmation password is the same as the password. → Go to <b>Step 7</b>	
<b>CONCLUSION:</b>	The user has changed their password and the new password has been updated to the <b>Person</b> table. The system confirms with a notification that the user's password was successfully updated.	
<b>POST-CONDITION:</b>	The user has changed their password.	
<b>BUSINESS RULES</b>	<ul style="list-style-type: none"> <li>• Only users who can login to the system with an existing email address and password may change their password.</li> </ul>	
<b>IMPLEMENTATION CONSTRAINTS AND SPECIFICATIONS</b>	<ul style="list-style-type: none"> <li>• None</li> </ul>	
<b>ASSUMPTIONS:</b>	<ul style="list-style-type: none"> <li>• None</li> </ul>	
<b>OPEN ISSUES:</b>	None	

<b>USE CASE NAME:</b>	Login	<b>USE CASE TYPE</b>
<b>USE CASE ID:</b>	1.3	Business Requirements: <input type="checkbox"/>
<b>PRIORITY:</b>	High	System Analysis: <input checked="" type="checkbox"/>
<b>SOURCE:</b>	TuksRes Women In Leadership Academy	System Design: <input type="checkbox"/>
<b>PRIMARY BUSINESS ACTOR</b>	User	
<b>PRIMARY SYSTEM ACTOR</b>	None	
<b>OTHER PARTICIPATING ACTORS:</b>	<ul style="list-style-type: none"> <li>None</li> </ul>	
<b>OTHER INTERESTED STAKEHOLDERS:</b>	<ul style="list-style-type: none"> <li>None</li> </ul>	
<b>DESCRIPTION:</b>	<p>This use case describes the event of a user logging into the system. The user will enter a registered email address and password and the system will verify that the email address exists on the system, and that the password entered matches up with the email address in the <b>Person</b> table. The system will then display the appropriate main menu according to the type of user that is logging in. The use case concludes when the user has successfully logged in and their main menu is displayed.</p>	
<b>PRE-CONDITION:</b>	The user is registered on the TRWLA system.	
<b>TRIGGER:</b>	A user wants to login to the TRWLA system.	
<b>TYPICAL COURSE OF EVENTS:</b>	<b>Actor Action</b>	<b>System Response</b>
	<b>Step 1:</b> A user wants to login to the TRWLA system and proceeds to the login section.	<b>Step 2:</b> The system displays the login section, which prompts the user to enter the required login details with the options to 'Login', 'Check Forgotten Password' or 'Register'.
	<b>Step 3:</b> The user enters the required login details namely: <ul style="list-style-type: none"> <li>Email Address</li> <li>Password</li> </ul>	<b>Step 4:</b> The system verifies that the: <ul style="list-style-type: none"> <li>Email Address exists in the <b>Person</b> table, and that the password entered matches the</li> <li>Password related to the above-mentioned email address in the <b>Person</b> table.</li> </ul>
		<b>Step 5:</b> The system determines which type of user the user is, according to the <b>UserType</b> table which is linked to the Person table via the UserTypeID key, and displays the appropriate main menu according to the user type.

		<b>Step 6:</b> The system updates the <b>AuditLog</b> table with the following information: <ul style="list-style-type: none"> <li>• PersonID (Determined according to the current logged in user)</li> <li>• UserID</li> <li>• LoginTime (Current timestamp)</li> <li>• LoginDate (Current date)</li> <li>• LoginDuration (Time between login and logout).</li> </ul>
		<b>Step 6:</b> The system displays the appropriate main menu according to the user type.
<b>ALTERNATE COURSES:</b>	<b>Alt-Step 5:</b> The user entered an invalid email address or password, the system prompts the user to try login again. <b>→ Go to Step 3</b>	
<b>CONCLUSION:</b>	The user has entered a registered and valid email address and password, and has successfully logged into the system. The system displays an appropriate main menu according to the type of user logging in.	
<b>POST-CONDITION:</b>	The user has logged into the TRWLA system.	
<b>BUSINESS RULES</b>	<ul style="list-style-type: none"> <li>• Only registered users may login to the TRWLA system.</li> </ul>	
<b>IMPLEMENTATION CONSTRAINTS AND SPECIFICATIONS</b>	<ul style="list-style-type: none"> <li>• The user must use a valid email address as their login 'username'</li> </ul>	
<b>ASSUMPTIONS:</b>	<ul style="list-style-type: none"> <li>• None</li> </ul>	
<b>OPEN ISSUES:</b>	None	



<b>USE CASE NAME:</b>	Register User	<b>USE CASE TYPE</b>
<b>USE CASE ID:</b>	AUC 1	Abstract: <input checked="" type="checkbox"/>
<b>PRIORITY:</b>	High	Extension: <input type="checkbox"/>
<b>SOURCE:</b>	TuksRes Women In Leadership Academy	
<b>PARTICIPATING ACTORS:</b>	<ul style="list-style-type: none"> <li>User (PBA)</li> </ul>	
<b>DESCRIPTION:</b>	<p>This use case describes the event of the user registering themselves on the system. The use case begins when the user wants to register on the TRWLA system and proceeds to the login section. They will then select the option to register. The system will prompt the user to enter the required login details. Once the user has entered the required information the system will verify that the information is complete and valid. The system will the insert this information into the <u>Person</u> and <u>SecurityAnswer</u> tables, and prompt the user to select their user type. The use case concludes when the system prompts the user to select a user type.</p>	
<b>PRE-CONDITION:</b>	None	
<b>TYPICAL COURSE OF EVENTS:</b>	<b>Step 1:</b> The user wants to register on the system and proceeds to the login section.	
	<b>Step 2:</b> The system prompts the user with the options to 'Login', 'Check Forgotten Password' or 'Register' for the system.	
	<b>Step 3:</b> The user selects the option to 'Register' for the system.	
	<b>Step 4:</b> The system prompts the user to enter their desired login details and to answer the display security question.	
	<b>Step 5:</b> The user enters their required login details: email address, password, confirm password and answer to the security question, and selects the option to 'Check Login Details'.	
	<b>Step 6:</b> The system verifies that all required fields have been filled in, that the fields contain valid details: <ul style="list-style-type: none"> <li>Email Address (Minimum 7 characters, Maximum 255 characters. Must be a full validated email address. Cannot already exist in the <u>Person</u> table)</li> <li>Password (Maximum 35 characters, minimum 8 characters, must contain at least one numeric value)</li> <li>Confirm Password (Identical to the password)</li> <li>Security Answer (Max 200 characters)</li> </ul>	
	<b>Step 7:</b> The system inserts the validated information into the EmailAddress and Password attributes in the <u>Person</u> table and the Answer attribute in the <u>SecurityAnswer</u> table. The <u>Person</u> and <u>SecurityAnswer</u> tables are connected via the PersonID key.	

	<b>Step 8:</b> The system prompts the user to select their user type. The available user types are retrieved from the <b>UserType</b> table.
<b>ALTERNATE COURSES:</b>	Alt <b>Step 7:</b> The user has entered invalid login details, the system prompts the user to enter valid information → Go to <b>Step 5</b>
<b>POST-CONDITION:</b>	The user has created login details and selected a user type.

## 2.3.2 Volunteer Subsystem Narratives

<b>USE CASE NAME:</b>	Register Volunteer	<b>USE CASE TYPE</b>
<b>USE CASE ID:</b>	2.1	Business Requirements: <input type="checkbox"/>
<b>PRIORITY:</b>	High	System Analysis: <input checked="" type="checkbox"/>
<b>SOURCE:</b>	TuksRes Women In Leadership Academy	System Design: <input type="checkbox"/>
<b>PRIMARY BUSINESS ACTOR</b>	Volunteer	
<b>PRIMARY SYSTEM ACTOR</b>	None	
<b>OTHER PARTICIPATING ACTORS:</b>	<ul style="list-style-type: none"> <li>None</li> </ul>	
<b>OTHER INTERESTED STAKEHOLDERS:</b>	<ul style="list-style-type: none"> <li>Board Members and Management</li> </ul>	
<b>DESCRIPTION:</b>	<p>This use case describes the event of a volunteer registering themselves on the system. The use case begins when the volunteer is officially employed by TRWLA and wants to register on the system. The volunteer will go to the login section of the system, select the option to register and then register their required details onto the system as well as create a username and password. The system will have to verify that the volunteer is registering valid details. The use case concludes when the volunteer has successfully registered on the system.</p>	
<b>PRE-CONDITION:</b>	The volunteer has been employed by TRWLA and wants to register as a volunteer on the system	
<b>TRIGGER:</b>	The volunteer wants to register on the TRWLA system.	
<b>TYPICAL COURSE OF EVENTS:</b>	<b>Actor Action</b>	<b>System Response</b>
	<b>Step 1:</b> The volunteer wants to register on the system and proceeds to the login section.	<b>Step 2:</b> The system prompts the volunteer to register their login details. (Invoke AUC 1. Register User)
	<b>Step 3:</b> The volunteer selects the Volunteer option.	<b>Step 4:</b> The system prompts the volunteer to enter a unique code.
	<b>Step 5:</b> The volunteer enters a unique code which was provided to them by management	<b>Step 6:</b> The system verifies the unique code, such that it is a 5 digit numeric value that exists in the <u>UniqueCode</u> table and is marked as 'unused'.
		<b>Step 7:</b> The system marks the unique code as 'used' in the <u>UniqueCode</u> table.
		<b>Step 8:</b> The system prompts the user to enter the required personal information with the option to 'Save'

	<p><b>Step 9:</b> The volunteer enters their personal details:</p> <ul style="list-style-type: none"> <li>• Name</li> <li>• Surname</li> <li>• Date of Birth</li> <li>• Phone Number</li> <li>• Race</li> <li>• Language</li> <li>• VolunteerType</li> </ul> <p>into the given fields and selects the option to 'Save'.</p>	<p><b>Step 10:</b> The system verifies that all required fields have been filled in and that the fields contain valid details:</p> <ul style="list-style-type: none"> <li>• Name (Maximum 35 characters)</li> <li>• Surname (Maximum 35 characters)</li> <li>• Date of Birth (Maximum 10 characters in the format CCYY-MM-DD)</li> <li>• Phone Number (Max 15 characters (No spaces))</li> <li>• Race (Maximum 35 characters)</li> <li>• Home Language (Maximum 35 characters)</li> <li>• Language (Selected from a predetermined set of options as retrieved from the <u>Language</u> table)</li> <li>• VolunteerType (Selected from a list of options provided as retrieved from the <u>VolunteerType</u> table)</li> </ul>
		<ul style="list-style-type: none"> <li>• <b>Step 11:</b> The system inserts the information into the <u>Person</u>, <u>PersonLanguage</u> and <u>Volunteer</u> tables which are linked via the PersonID key and confirms with a notification that registration was successful.</li> </ul>
<b>ALTERNATE COURSES:</b>	<p><b>Alt-Step 7:</b> The verification of the unique code was unsuccessful, the system prompts the user to enter a valid unique code</p> <p>→ Go to <b>Step 5</b></p>	
	<p><b>Alt-Step 11:</b> The verification of the volunteers' personal details was unsuccessful and the system prompts the user to enter all their information in the correct format.</p> <p>→ Go to <b>Step 19</b></p>	
<b>CONCLUSION:</b>	<p>The system verifies that the volunteer has entered valid and complete personal information, and inserts the information into the <u>Person</u> and <u>Volunteer</u> tables. The system then confirms that registration was successful.</p>	
<b>POST-CONDITION:</b>	<p>The volunteer has successfully been registered on the system.</p>	
<b>BUSINESS RULES</b>	<ul style="list-style-type: none"> <li>• Only people who have passed the interview and</li> </ul>	

	application process and have been officially hired by TRWLA may register on the system.
<b>IMPLEMENTATION CONSTRAINTS AND SPECIFICATIONS</b>	<ul style="list-style-type: none"><li>• The user must use their email address as their username when they register.</li><li>• Admin must generate a unique code for each volunteer to use for registration.</li></ul>
<b>ASSUMPTIONS:</b>	<ul style="list-style-type: none"><li>• None</li></ul>
<b>OPEN ISSUES:</b>	None

<b>USE CASE NAME:</b>	Search Volunteer	<b>USE CASE TYPE</b>
<b>USE CASE ID:</b>	2.2	Business Requirements: <input type="checkbox"/>
<b>PRIORITY:</b>	High	System Analysis: <input checked="" type="checkbox"/>
<b>SOURCE:</b>	TuksRes Women In Leadership Academy	System Design: <input type="checkbox"/>
<b>PRIMARY BUSINESS ACTOR</b>	User	
<b>PRIMARY SYSTEM ACTOR</b>	None	
<b>OTHER PARTICIPATING ACTORS:</b>	<ul style="list-style-type: none"> <li>None</li> </ul>	
<b>OTHER INTERESTED STAKEHOLDERS:</b>	<ul style="list-style-type: none"> <li>None</li> </ul>	
<b>DESCRIPTION:</b>	<p>This use case describes the event of a user searching for a volunteer. The use case begins when the user has logged onto the system and wants to search a volunteer. The user will search the volunteer and retrieve all the relevant details. The use case concludes when the user is satisfied with the search details and returns to their main menu.</p>	
<b>PRE-CONDITION:</b>	The user is logged into the system.	
<b>TRIGGER:</b>	A user wants to search a volunteer.	
<b>TYPICAL COURSE</b>	Actor Action	System Response
<b>OF EVENTS:</b>	<b>Step 1:</b> A user wants to search a volunteer and proceeds to the main menu.	<b>Step 2:</b> The system prompts the user with a main menu, with an option 'Volunteers'.
	<b>Step 3:</b> The user selects the option 'Volunteers'.	<b>Step 4:</b> The system prompts the user with a list of TRWLA Volunteers, with details such as: Name Surname which have been retrieved from the <b>Person</b> table, and the option to search a specific volunteer by means of entering their details into a field.
	<b>Step 5:</b> The user enters the relevant details into the search field.	<b>Step 6:</b> The system prompts the user with a list of Volunteers' details including: <ul style="list-style-type: none"> <li>Name</li> <li>Surname</li> </ul> from the <b>Person</b> table with details similar to the ones the user entered into the search field.

	<b>Step 7:</b> The user selects the specific Volunteer that they are looking for.	<b>Step 8:</b> The system prompts the user with the volunteers' details, such as Name Surname Date of Birth Email Address Phone Number As retrieved from the <b>Person</b> and <b>Volunteer</b> tables with the option the option to 'Return to Search' section or to 'Return to Main Menu'.
	<b>Step 9:</b> The user selects the option to 'Return to Main Menu'.	<b>Step 10:</b> The system prompts the user with the main menu
<b>ALTERNATE COURSES:</b>	<b>Alt-Step 6:</b> There are no volunteers that match that description, the system prompts the user to try different search details. → Go to <b>Step 5</b>	
<b>CONCLUSION:</b>	The user has successfully searched for the specific volunteer they are looking for, retrieved their relevant details and returned to the main menu.	
<b>POST-CONDITION:</b>	The user has searched for a Volunteer.	
<b>BUSINESS RULES</b>	<ul style="list-style-type: none"> <li>• None</li> </ul>	
<b>IMPLEMENTATION CONSTRAINTS AND SPECIFICATIONS</b>	<ul style="list-style-type: none"> <li>• The volunteer's password must be kept private.</li> </ul>	
<b>ASSUMPTIONS:</b>	<ul style="list-style-type: none"> <li>• None</li> </ul>	
<b>OPEN ISSUES:</b>	None	

<b>USE CASE NAME:</b>	Update Volunteer	<b>USE CASE TYPE</b>
<b>USE CASE ID:</b>	2.3	Business Requirements: <input type="checkbox"/>
<b>PRIORITY:</b>	Medium	System Analysis: <input checked="" type="checkbox"/>
<b>SOURCE:</b>	TuksRes Women In Leadership Academy	System Design: <input type="checkbox"/>
<b>PRIMARY BUSINESS ACTOR</b>	Volunteer	
<b>PRIMARY SYSTEM ACTOR</b>	None	
<b>OTHER PARTICIPATING ACTORS:</b>	<ul style="list-style-type: none"> <li>None</li> </ul>	
<b>OTHER INTERESTED STAKEHOLDERS:</b>	<ul style="list-style-type: none"> <li>None</li> </ul>	
<b>DESCRIPTION:</b>	<p>This use case describes the event of a volunteer updating their details on the TRWLA system. The use case begins when the volunteer has logged onto the system and wants to update their details. They will then go to their profile, update their details and select the option to 'Save' their details. The system must verify that the volunteer has updated their details with complete and valid information, and update the <b>Person</b> and <b>Volunteer</b> tables accordingly. The use case will conclude when the volunteer has successfully updated their details.</p>	
<b>PRE-CONDITION:</b>	The user is logged into the system.	
<b>TRIGGER:</b>	The volunteer wants to update her details.	
<b>TYPICAL COURSE</b>	Actor Action	System Response
<b>OF EVENTS:</b>	<b>Step 1:</b> A volunteer wants to update her details and proceeds to the main menu.	<b>Step 2:</b> The system prompts the volunteer with a main menu, with an option 'My Profile'.
	<b>Step 3:</b> The volunteer selects the option 'My Profile'.	<b>Step 4:</b> The system prompts the volunteer with all their details in an editable format with the options to 'Save' or 'Cancel'.
	<b>Step 5:</b> The volunteer changes any of their details , including Name Surname Date of Birth Email Address Phone Number Race Home Language Volunteer Type	<b>Step 6:</b> The system verifies that all required fields have been filled in, that the fields contain valid details: Name (Maximum 35 characters) Surname (Maximum 35 characters) Date of Birth (Maximum 10 characters in the format CCYY-MM-DD) Email Address (Minimum 7 characters, Maximum 255 characters. Must be a full validated email address)



	in the given fields and selects the 'Save' option.	Phone Number (Max 15 characters (No spaces)) Race(Maximum 35 characters) Home Language (Maximum 35 characters) VolunteerType (Selected from a list of options provided as retrieved from the
		<b>Step 7:</b> The system updates the changed volunteer details into the <b><u>Volunteer</u></b> and <b><u>Person</u></b> tables and confirms with a notification that the update was successful.
<b>ALTERNATE COURSES:</b>	<b>Alt-Step 5:</b> The Volunteer selects the 'Cancel' option. → Go to <b>Step 3</b>	
	<b>Alt-Step 7:</b> The verification of volunteers' details was unsuccessful and the system prompts the user to correct their information. → Go to <b>Step 5</b>	
<b>CONCLUSION:</b>	The volunteer has changed all the details that they wanted to change and the system has updated these details in the <b><u>Person</u></b> and <b><u>Volunteer</u></b> tables. The system confirms that update was successful.	
<b>POST-CONDITION:</b>	The volunteer has successfully updated her details.	
<b>BUSINESS RULES</b>	<ul style="list-style-type: none"> <li>Volunteers can only update their own details.</li> </ul>	
<b>IMPLEMENTATION CONSTRAINTS AND SPECIFICATIONS</b>	<ul style="list-style-type: none"> <li>The volunteer must be registered on the system</li> </ul>	
<b>ASSUMPTIONS:</b>	<ul style="list-style-type: none"> <li>None</li> </ul>	
<b>OPEN ISSUES:</b>	None	

<b>USE CASE NAME:</b>	Delete Volunteer	<b>USE CASE TYPE</b>
<b>USE CASE ID:</b>	2.4	Business Requirements: <input type="checkbox"/>
<b>PRIORITY:</b>	Low	System Analysis: <input checked="" type="checkbox"/>
<b>SOURCE:</b>	TuksRes Women In Leadership Academy	System Design: <input type="checkbox"/>
<b>PRIMARY BUSINESS ACTOR</b>	Admin	
<b>PRIMARY SYSTEM ACTOR</b>	None	
<b>OTHER PARTICIPATING ACTORS:</b>	<ul style="list-style-type: none"> <li>None</li> </ul>	
<b>OTHER INTERESTED STAKEHOLDERS:</b>	<ul style="list-style-type: none"> <li>None</li> </ul>	
<b>DESCRIPTION:</b>	<p>This use case describes the event of admin deleting a volunteer. The use case begins when admin has logged in and wants to delete a volunteer. Admin will search, select and delete the relevant volunteer. The system must display a warning to admin that they are about to permanently delete the volunteer from the system, and the admin must accept and proceed with the deletion. The use case concludes when the volunteer has been successfully deleted and the main menu is displayed.</p>	
<b>PRE-CONDITION:</b>	The admin is logged into the system.	
<b>TRIGGER:</b>	Admin wants to delete a Volunteer.	
<b>TYPICAL COURSE</b>	Actor Action	System Response
<b>OF EVENTS:</b>	<b>Step 1:</b> Admin wants to delete a volunteer and proceeds to the main menu.	<b>Step 2:</b> The system prompts Admin with a main menu, with an option 'Volunteers'
	<b>Step 3:</b> Admin selects the option 'Volunteers'.	<b>Step 4:</b> The system displays to admin a list of volunteers as retrieved from the <u>Person</u> and <u>Volunteer</u> tables, and prompts admin to search a volunteer (Invoke UC 2.2).
		<b>Step 5:</b> The system prompts admin with the Volunteers' details including: <ul style="list-style-type: none"> <li>Name</li> <li>Surname</li> <li>Date of Birth</li> <li>Email Address</li> <li>Phone Number</li> <li>Race</li> <li>Home Language</li> </ul>

		from the <b>Person</b> and <b>Volunteer</b> tables and with the options to 'Delete' or 'Cancel'
	<b>Step 6:</b> Admin selects the 'Delete' option.	<b>Step 7:</b> The system prompts admin with a warning that they are about to permanently delete a volunteer, with the options to 'Confirm Deletion' or to 'Cancel'.
	<b>Step 8:</b> Admin selects the option to 'Confirm Deletion'.	<b>Step 9:</b> The system removes the volunteer from the <b>Person</b> and <b>Volunteer</b> table and confirms with a notification that the volunteer has been successfully removed from the system.
<b>ALTERNATE COURSES:</b>	<b>Alt-Step 8:</b> Admin selects the option to 'Cancel' → Go to <b>Step 5</b>	
<b>CONCLUSION:</b>	Admin has successfully deleted a volunteer and the system has removed all information pertaining to that volunteer from the <b>Volunteer</b> and <b>Person</b> tables. The system confirms that the volunteer has been successfully removed from the system.	
<b>POST-CONDITION:</b>	Admin has successfully deleted a volunteer.	
<b>BUSINESS RULES</b>	<ul style="list-style-type: none"> <li>Only admin can delete a volunteer</li> </ul>	
<b>IMPLEMENTATION CONSTRAINTS AND SPECIFICATIONS</b>	<ul style="list-style-type: none"> <li>Admin must be given administrator login details at the implementation of the system.</li> <li>A delete cascade must be implemented in order to ensure that all information related to the volunteer is deleted across all tables.</li> </ul>	
<b>ASSUMPTIONS:</b>	<ul style="list-style-type: none"> <li>None</li> </ul>	
<b>OPEN ISSUES:</b>	None	

<b>USE CASE NAME:</b>	Add Volunteer Type	<b>USE CASE TYPE</b>
<b>USE CASE ID:</b>	2.5	Business Requirements: <input type="checkbox"/>
<b>PRIORITY:</b>	High	System Analysis: <input checked="" type="checkbox"/>
<b>SOURCE:</b>	TuksRes Women In Leadership Academy	System Design: <input type="checkbox"/>
<b>PRIMARY BUSINESS ACTOR</b>	Admin	
<b>PRIMARY SYSTEM ACTOR</b>	None	
<b>OTHER PARTICIPATING ACTORS:</b>	<ul style="list-style-type: none"> <li>None</li> </ul>	
<b>OTHER INTERESTED STAKEHOLDERS:</b>	<ul style="list-style-type: none"> <li>None</li> </ul>	
<b>DESCRIPTION:</b>	<p>This use case describes the event of admin adding a volunteer type on the system. The use case begins when admin wants to add a new volunteer type on the system. Once admin has entered all the required details of the new volunteer type, the system will have to verify that all the information entered is complete and valid, and will then insert the information into the <b>VolunteerType</b> table. The use case concludes when the system confirms that a new volunteer type has been added to the system.</p>	
<b>PRE-CONDITION:</b>	Admin is logged into the system	
<b>TRIGGER:</b>	Admin wants to add a new volunteer type on the system	
<b>TYPICAL COURSE</b>	Actor Action	System Response
<b>OF EVENTS:</b>	<b>Step 1:</b> Admin wants to add a new volunteer type on the system and proceeds to the main menu.	<b>Step 2:</b> The system prompts admin with a main menu with the option 'Volunteers'
	<b>Step 3:</b> Admin selects the option 'Volunteers'	<b>Step 4:</b> The system prompts admin with the volunteer section, and an option 'Volunteer Types'
	<b>Step 5:</b> Admin selects the option 'Volunteer Types'.	<b>Step 6:</b> The system prompts admin with the volunteer type section, with the option to 'Add Volunteer Type' or to search.
	<b>Step 7:</b> Admin selects the option to 'Add Volunteer Type'	<b>Step 8:</b> The system prompts admin to enter all the relevant details of the new volunteer type with the option to 'Add Volunteer Type'

	<b>Step 9:</b> Admin enters all the relevant details, including: <ul style="list-style-type: none"> <li>• VolunteerTypeName</li> <li>• Description</li> </ul> and selects the option to 'Add Volunteer Type'.	<b>Step 10:</b> The system verifies that all the required fields have been entered with the correct details: Name (Maximum 35 characters) Description (Maximum 300 characters)
		<b>Step 11:</b> The system inserts the relevant data from the fields into the <b><u>VolunteerType</u></b> table and confirms with a notification that the volunteer type has been successfully added to the system.
<b>ALTERNATE COURSES:</b>	<b>Alt-Step 11:</b> The information was not entered correctly and the system prompts admin to re-enter the incorrect information. <b>→ Go to Step 9</b>	
<b>CONCLUSION:</b>	Admin has successfully added a volunteer type which has been inserted into the <b><u>VolunteerType</u></b> table. The system confirms to admin that the volunteer type has been successfully added to the system.	
<b>POST-CONDITION:</b>	A volunteer type has been added to the system.	
<b>BUSINESS RULES</b>	<ul style="list-style-type: none"> <li>• Only admin may add a volunteer type</li> </ul>	
<b>IMPLEMENTATION CONSTRAINTS AND SPECIFICATIONS</b>	<ul style="list-style-type: none"> <li>• Admin must be given administrator login details at the implementation of the system.</li> </ul>	
<b>ASSUMPTIONS:</b>	<ul style="list-style-type: none"> <li>• None</li> </ul>	
<b>OPEN ISSUES:</b>	None	

<b>USE CASE NAME:</b>	Search Volunteer Type	<b>USE CASE TYPE</b>
<b>USE CASE ID:</b>	2.6	Business Requirements: <input type="checkbox"/>
<b>PRIORITY:</b>	High	System Analysis: <input checked="" type="checkbox"/>
<b>SOURCE:</b>	TuksRes Women In Leadership Academy	System Design: <input type="checkbox"/>
<b>PRIMARY BUSINESS ACTOR</b>	Admin	
<b>PRIMARY SYSTEM ACTOR</b>	None	
<b>OTHER PARTICIPATING ACTORS:</b>	<ul style="list-style-type: none"> <li>None</li> </ul>	
<b>OTHER INTERESTED STAKEHOLDERS:</b>	<ul style="list-style-type: none"> <li>None</li> </ul>	
<b>DESCRIPTION:</b>	<p>This use case describes the event of admin searching a volunteer type. The use case begins when admin logs onto the system and wants to search a volunteer type. Admin will search for a volunteer type and gather all the relevant information the she is looking for. The use case concludes when admin has searched the volunteer type and the system redirects her to the main menu.</p>	
<b>PRE-CONDITION:</b>	Admin is logged into the system.	
<b>TRIGGER:</b>	Admin wants to search a volunteer type.	
<b>TYPICAL COURSE</b>	Actor Action	System Response
<b>OF EVENTS:</b>	<b>Step 1:</b> Admin wants to search a volunteer type and proceeds to the main menu.	<b>Step 2:</b> The system prompts Admin with a main menu, with an option 'Volunteers'.
	<b>Step 3:</b> The user selects the option 'Volunteers'.	<b>Step 4:</b> The system prompts admin with the volunteer section and the option 'Volunteer Types'.
	<b>Step 5:</b> Admin selects the option 'Volunteer Types'.	<b>Step 6:</b> The system prompts the user with a list of volunteer types from the <b>VolunteerType</b> table and the option to search a Volunteer type by entering details into the search field.
	<b>Step 7:</b> Admin enters the relevant details of the volunteer type they are searching for into the search field.	<b>Step 8:</b> The system prompts admin with a list of Volunteer types and their details such as: <ul style="list-style-type: none"> <li>Name</li> </ul> from the <b>VolunteerType</b> table that have similar details as the ones entered in the search field by admin.

	<b>Step 9:</b> Admin selects the particular volunteer type that they're looking for.	<b>Step 10:</b> The system prompts admin with the volunteer types details including: <ul style="list-style-type: none"> <li>• Name</li> <li>• Description</li> </ul> from the <b>VolunteerType</b> table of the selected Volunteer type, with the options to 'Save', 'Delete' or 'Cancel'
	<b>Step 11:</b> Admin selects the option to 'Cancel'	<b>Step 12:</b> The system prompts admin with the main menu.
<b>ALTERNATE COURSES:</b>	<b>Alt-Step 8:</b> There are no Volunteer types that match that description, the system prompts admin to try different search details. → Go to <b>Step 7</b>	
<b>CONCLUSION:</b>	Admin has searched the relevant volunteer type and is satisfied with the search results. The system prompts admin with the main menu	
<b>POST-CONDITION:</b>	The admin has searched for a volunteer type.	
<b>BUSINESS RULES</b>	<ul style="list-style-type: none"> <li>• None</li> </ul>	
<b>IMPLEMENTATION CONSTRAINTS AND SPECIFICATIONS</b>	<ul style="list-style-type: none"> <li>• Admin must be given administrator login details at the implementation of the system.</li> </ul>	
<b>ASSUMPTIONS:</b>	<ul style="list-style-type: none"> <li>• None</li> </ul>	
<b>OPEN ISSUES:</b>	None	

<b>USE CASE NAME:</b>	Update Volunteer Type	<b>USE CASE TYPE</b>
<b>USE CASE ID:</b>	2.7	Business Requirements: <input type="checkbox"/>
<b>PRIORITY:</b>	Medium	System Analysis: <input checked="" type="checkbox"/>
<b>SOURCE:</b>	TuksRes Women In Leadership Academy	System Design: <input type="checkbox"/>
<b>PRIMARY BUSINESS ACTOR</b>	Admin	
<b>PRIMARY SYSTEM ACTOR</b>	None	
<b>OTHER PARTICIPATING ACTORS:</b>	<ul style="list-style-type: none"> <li>None</li> </ul>	
<b>OTHER INTERESTED STAKEHOLDERS:</b>	<ul style="list-style-type: none"> <li>None</li> </ul>	
<b>DESCRIPTION:</b>	<p>This use case describes the event of admin updating the details of the volunteer type on the TRWLA system. The use case begins when the admin has logged onto the system and wants to update the volunteer type. They will then search the relevant volunteer type, update the details and then select the option to 'Save'. The system will have to verify that the changed details are complete and valid, and will then update the <u><b>VolunteerType</b></u> table accordingly. The use case will conclude when admin has successfully updated the volunteer type.</p>	
<b>PRE-CONDITION:</b>	The admin is logged into the system.	
<b>TRIGGER:</b>	Admin wants to update a volunteer type.	
<b>TYPICAL COURSE</b>	Actor Action	System Response
<b>OF EVENTS:</b>	<b>Step 1:</b> Admin wants to update a volunteer type and proceeds to the main menu.	<b>Step 2:</b> The system prompts admin with a main menu, with an option 'Volunteers'.
	<b>Step 3:</b> Admin selects the 'Volunteers' option.	<b>Step 4:</b> The system prompts admin with the volunteer section, and prompts the user to search a volunteer type (Invoke UC 2.6).
		<b>Step 8:</b> The system prompts admin with the relevant volunteer type with its details as retrieved from the <u><b>VolunteerType</b></u> table, in editable fields and the option to 'Save', 'Delete' or 'Cancel'.
	<b>Step 9:</b> Admin changes the relevant details such as: <ul style="list-style-type: none"> <li>Name</li> <li>Description</li> </ul>	<b>Step 10:</b> The system verifies that all required fields have been filled in and that the fields contain valid details, namely:



	and selects the 'Save' option.	Name (Maximum 35 characters) Description (Maximum 300 characters)
		<b>Step 11:</b> The system updates the <b><u>VolunteerType</u></b> table with the newly changed details and confirms with a notification that the volunteer type has been successfully updated.
<b>ALTERNATE COURSES:</b>	<b>Alt-Step 9a:</b> Admin selects the 'Cancel' option. → Go to <b>Step 8</b>	
	<b>Alt-Step 9b:</b> Admin selects the delete option. → Invoke UC 2.8	
	<b>Alt-Step 11:</b> The verification of volunteer type details was unsuccessful and the system prompts the admin to correct their information. → Go to <b>Step 9</b>	
<b>CONCLUSION:</b>	Admin has searched and updated a volunteer type. The system has verified that the details updated are complete and valid and updates the <b><u>VolunteerType</u></b> table. The system confirms that the volunteer type has been successfully updated.	
<b>POST-CONDITION:</b>	Admin has updated a volunteer type	
<b>BUSINESS RULES</b>	<ul style="list-style-type: none"> <li>Only admin can update a volunteer type</li> </ul>	
<b>IMPLEMENTATION CONSTRAINTS AND SPECIFICATIONS</b>	<ul style="list-style-type: none"> <li>Admin must be given administrator login details at the implementation of the system.</li> </ul>	
<b>ASSUMPTIONS:</b>	<ul style="list-style-type: none"> <li>None</li> </ul>	
<b>OPEN ISSUES:</b>	None	

<b>USE CASE NAME:</b>	Delete Volunteer Type	<b>USE CASE TYPE</b>
<b>USE CASE ID:</b>	2.8	Business Requirements: <input type="checkbox"/>
<b>PRIORITY:</b>	Low	System Analysis: <input checked="" type="checkbox"/>
<b>SOURCE:</b>	TuksRes Women In Leadership Academy	System Design: <input type="checkbox"/>
<b>PRIMARY BUSINESS ACTOR</b>	Admin	
<b>PRIMARY SYSTEM ACTOR</b>	None	
<b>OTHER PARTICIPATING ACTORS:</b>	<ul style="list-style-type: none"> <li>None</li> </ul>	
<b>OTHER INTERESTED STAKEHOLDERS:</b>	<ul style="list-style-type: none"> <li>None</li> </ul>	
<b>DESCRIPTION:</b>	<p>This use case describes the event of admin deleting a Volunteer type. The use case begins when admin has logged in and wants to delete a Volunteer type. Admin will search, select and delete the relevant Volunteer type. The use case concludes when the Volunteer type has been successfully deleted.</p>	
<b>PRE-CONDITION:</b>	The admin is logged into the system.	
<b>TRIGGER:</b>	Admin wants to delete a Volunteer type.	
<b>TYPICAL COURSE</b>	Actor Action	System Response
<b>OF EVENTS:</b>	<b>Step 1:</b> Admin wants to delete a volunteer type and proceeds to the main menu.	<b>Step 2:</b> The system prompts Admin with a main menu, with an option 'Volunteers'
	<b>Step 3:</b> Admin selects the option 'Volunteers'.	<b>Step 4:</b> The system prompts admin with the volunteer section, and prompts the user to search a volunteer type (Invoke UC 2.6).
		<b>Step 5:</b> The system prompts admin with the relevant volunteer type with its details such as: <ul style="list-style-type: none"> <li>Name</li> <li>Description</li> </ul> in editable fields and the option to 'Save', 'Delete' or 'Cancel'.
	<b>Step 6:</b> Admin selects the option to 'Delete' the volunteer type.	<b>Step 7:</b> The system prompts admin with a warning that they are about to permanently delete a volunteer type, with the options to 'Confirm Deletion' or to 'Cancel'.

	<b>Step 8:</b> Admin selects the option to 'Confirm Deletion'.	<b>Step 9:</b> The system removes the relevant volunteer type from the <b><u>VolunteerType</u></b> table and confirms with a notification that the volunteer type has been successfully removed from the system.
<b>ALTERNATE COURSES:</b>	<b>Alt-Step 8a:</b> Admin selects the 'Cancel' option. → Go to <b>Step 5</b>	
	<b>Alt-Step 8b:</b> Admin selects the 'Save' option. → Invoke UC 2.7	
	<b>Alt-Step 9:</b> The volunteer type that admin is trying to remove is referenced in another table. A delete restrict is applied to the volunteer type. The system cannot delete a volunteer type that is referenced in another table as it will result in orphan records. The system displays a notification to admin that this volunteer type cannot be deleted. → Go to <b>Step 5</b>	
<b>CONCLUSION:</b>	Admin has searched and deleted the volunteer type. Before the system removed the volunteer type from the system, the system warned admin that they were about to permanently delete the volunteer type and prompted admin to confirm the deletion. Once admin confirmed the deletion the system removed the volunteer type. The system confirms that the volunteer type has been successfully removed from the system.	
<b>POST-CONDITION:</b>	Admin has successfully deleted a volunteer type.	
<b>BUSINESS RULES</b>	<ul style="list-style-type: none"> <li>Only admin can delete a Volunteer</li> <li>When a volunteer type is deleted, a delete restrict must be enforced to ensure that no orphan records are left in other tables.</li> </ul>	
<b>IMPLEMENTATION CONSTRAINTS AND SPECIFICATIONS</b>	<ul style="list-style-type: none"> <li>Admin must be given administrator login details at the implementation of the system.</li> </ul>	
<b>ASSUMPTIONS:</b>	<ul style="list-style-type: none"> <li>None</li> </ul>	
<b>OPEN ISSUES:</b>	None	

<b>USE CASE NAME:</b>	Generate Unique Code	<b>USE CASE TYPE</b>
<b>USE CASE ID:</b>	2.9	Business Requirements: <input type="checkbox"/>
<b>PRIORITY:</b>	Medium	<b>System Analysis:</b> <input checked="" type="checkbox"/>
<b>SOURCE:</b>	TuksRes Women In Leadership Academy	System Design: <input type="checkbox"/>
<b>PRIMARY BUSINESS ACTOR</b>	Management	
<b>PRIMARY SYSTEM ACTOR</b>	None	
<b>OTHER PARTICIPATING ACTORS:</b>	<ul style="list-style-type: none"> <li>Volunteer (ERA)</li> </ul>	
<b>OTHER INTERESTED STAKEHOLDERS:</b>	<ul style="list-style-type: none"> <li>None</li> </ul>	
<b>DESCRIPTION:</b>	<p>This use case describes the event management generating a unique code which will be used by a volunteer to register for the system. The use case begins when a new volunteer is officially recognized as a volunteer as the academy. A management member will login to the system, select the 'Volunteer' option and then the option to generate the unique code. The system will then generate a unique code and display it to admin with the option to email it to the volunteer. This unique code will be stored in the <u>UniqueCode</u> table and marked as 'unused'. The use case concludes when management has successfully generated a unique code.</p>	
<b>PRE-CONDITION:</b>	Management is logged into the system	
<b>TRIGGER:</b>	Management wants to generate a unique code on the system	
<b>TYPICAL COURSE</b>	Actor Action	System Response
<b>OF EVENTS:</b>	<b>Step 1:</b> Management wants to generate a unique code on the system and proceeds to the main menu.	<b>Step 2:</b> The system prompts management with a main menu with the option 'Volunteers'
	<b>Step 3:</b> Management selects the option 'Volunteers'	<b>Step 4:</b> The system prompts admin with a list of volunteers, which have been retrieved from the <u>Person</u> and <u>Volunteer</u> table, and an option 'Generate Unique Code'
	<b>Step 5:</b> Management selects the option 'Generate Unique Code'.	<b>Step 6:</b> The system generates a unique code- a 5 digit numeric value- and displays it to management with the option to 'Email to Volunteer' or 'Return to Main Menu'

		<b>Step 7:</b> The system inserts the new Unique Code in the <u>UniqueCode</u> table and marks it as 'unused'.
	<b>Step 8:</b> Management selects the option to 'Email to Volunteer'	<b>Step 9:</b> The system prompts management to enter and email address with the option to 'Send Email'.
	<b>Step 10:</b> Admin enters the relevant email address and selects the option to 'Send Email'.	<b>Step 11:</b> The system verifies that the required field has been entered with the correct details: Email Address (Minimum 7 characters, Maximum 255 characters. Must be a full validated email address.)
		<b>Step 12:</b> The system sends the unique code to the email address and redirects management to the main menu.
<b>ALTERNATE COURSES:</b>	<b>Alt-Step 8:</b> Management selects the option to 'Return to Main Menu' → Go to <b>Step 2</b>	
	<b>Alt-Step 12:</b> The email address is invalid, the system prompts management to enter a valid email address. → Go to <b>Step 10</b>	
<b>CONCLUSION:</b>	Management has successfully generated a unique code which has been inserted into the <u>UniqueCode</u> table and marked as 'unused'. The system has sent the unique code to the volunteer.	
<b>POST-CONDITION:</b>	A unique code has been generated on the system.	
<b>BUSINESS RULES</b>	<ul style="list-style-type: none"> <li>Only management generate a unique code.</li> </ul>	
<b>IMPLEMENTATION CONSTRAINTS AND SPECIFICATIONS</b>	<ul style="list-style-type: none"> <li>None</li> </ul>	
<b>ASSUMPTIONS:</b>	<ul style="list-style-type: none"> <li>None</li> </ul>	
<b>OPEN ISSUES:</b>	None	

### 2.3.3 Student Subsystem Narratives

<b>USE CASE NAME:</b>	Register Student	<b>USE CASE TYPE</b>
<b>USE CASE ID:</b>	3.1	Business Requirements: <input type="checkbox"/>
<b>PRIORITY:</b>	High	System Analysis: <input checked="" type="checkbox"/>
<b>SOURCE:</b>	TuksRes Women In Leadership Academy	System Design: <input type="checkbox"/>
<b>PRIMARY BUSINESS ACTOR</b>	Student	
<b>PRIMARY SYSTEM ACTOR</b>	None	
<b>OTHER PARTICIPATING ACTORS:</b>	<ul style="list-style-type: none"> <li>None</li> </ul>	
<b>OTHER INTERESTED STAKEHOLDERS:</b>	<ul style="list-style-type: none"> <li>Board Members and Management</li> </ul>	
<b>DESCRIPTION:</b>	<p>This use case describes the event of a student registering themselves on the system. The use case begins when the student joins TRWLA and wants to register on the system. The student will go to the login section of the system, select the option to register and then register their required details onto the system. The system will have to verify that the student is registering valid details. The use case concludes when the student has successfully registered on the system.</p>	
<b>PRE-CONDITION:</b>	The student has joined TRWLA and wants to register as a student on the system	
<b>TRIGGER:</b>	The student wants to register on the TRWLA system.	
<b>TYPICAL COURSE</b>	<b>Actor Action</b>	<b>System Response</b>
<b>OF EVENTS:</b>	<b>Step 1:</b> The student wants to register on the system and proceeds to the login section.	<b>Step 2:</b> The system prompts the student to register their login details. (Invoke AUC 1. Register User)
	<b>Step 3:</b> The student selects the Student option.	<b>Step 4:</b> The system prompts the student to enter their personal information.
	<b>Step 5:</b> The student enters their personal details: <ul style="list-style-type: none"> <li>Student Number</li> <li>Name</li> <li>Surname</li> <li>Date of Birth</li> <li>Phone Number</li> <li>Degree</li> <li>Year of Study</li> <li>Race</li> <li>Residence</li> </ul>	<b>Step 6:</b> The system verifies that all required fields have been filled in and that the fields contain valid details: <ul style="list-style-type: none"> <li>Student Number (8 numeric characters)</li> <li>Name (Maximum 35 characters)</li> <li>Surname (Maximum 35 characters)</li> <li>Date of Birth (Maximum 10 characters in the format CCYY-MM-DD)</li> </ul>

	<ul style="list-style-type: none"> <li>Language(s) into the given fields and selects the option to 'Save'.</li> </ul>	<ul style="list-style-type: none"> <li>Phone Number (Max 15 characters (No spaces))</li> <li>Degree (Maximum 35 characters)</li> <li>Year of Study (Maximum 10 characters in the format CCYY-MM-DD)</li> <li>Race (Maximum 35 characters)</li> <li>Home Language (Maximum 35 characters)</li> <li>Residence (Selected from predetermined options as retrieved from the <u>Residence</u> table. The <u>Student</u> and <u>Residence</u> tables are connected via the ResID)</li> <li>Language (Selected from predetermined options as retrieved from the <u>Language</u> table. The <u>Person</u> and <u>Language</u> table are connected via the <u>PersonLanguage</u> association due to the many-to-many relationship, which contains the PersonID and LanguageID)</li> </ul>
		<ul style="list-style-type: none"> <li><b>Step 7:</b> The system inserts the information into the <u>Person</u>, <u>PersonLanguage</u>, <u>Degree</u> and <u>Student</u> tables. The <u>Person</u>, <u>Student</u> and <u>PersonLanguage</u> tables are connected via the PersonID key and the <u>Degree</u> and <u>Student</u> tables are connected via the StudentID key. The system confirms with a notification that registration was successful.</li> </ul>
<b>ALTERNATE COURSES:</b>	<b>Alt-Step 7:</b> The verification of the students' personal details was unsuccessful and the system prompts the student to enter all their information in the correct format. → Go to <b>Step 5</b>	
<b>CONCLUSION:</b>	The system verifies that the student has entered valid and complete user and personal information, and inserts the information into the <u>Person</u>	

	and <b><u>Student</u></b> table. The system then confirms that registration was successful.
<b>POST-CONDITION:</b>	The student has successfully been registered on the system.
<b>BUSINESS RULES</b>	<ul style="list-style-type: none"><li>• Only female students in 1<sup>st</sup> and 2<sup>nd</sup> year and in TuksRes residences may apply to be in the academy.</li></ul>
<b>IMPLEMENTATION CONSTRAINTS AND SPECIFICATIONS</b>	<ul style="list-style-type: none"><li>• The user must use their email address as their username when they register.</li></ul>
<b>ASSUMPTIONS:</b>	<ul style="list-style-type: none"><li>• None</li></ul>
<b>OPEN ISSUES:</b>	None



<b>USE CASE NAME:</b>	Search Student	<b>USE CASE TYPE</b>
<b>USE CASE ID:</b>	3.2	Business Requirements: <input type="checkbox"/>
<b>PRIORITY:</b>	High	System Analysis: <input checked="" type="checkbox"/>
<b>SOURCE:</b>	TuksRes Women In Leadership Academy	System Design: <input type="checkbox"/>
<b>PRIMARY BUSINESS ACTOR</b>	User	
<b>PRIMARY SYSTEM ACTOR</b>	None	
<b>OTHER PARTICIPATING ACTORS:</b>	<ul style="list-style-type: none"> <li>None</li> </ul>	
<b>OTHER INTERESTED STAKEHOLDERS:</b>	<ul style="list-style-type: none"> <li>None</li> </ul>	
<b>DESCRIPTION:</b>	<p>This use case describes the event of a user searching for a student. The use case begins when the user has logged onto the system and wants to search a student. The user will search the student and retrieve all the relevant details. The use case concludes when the user is satisfied with the search details and returns to their main menu.</p>	
<b>PRE-CONDITION:</b>	The user is logged into the system.	
<b>TRIGGER:</b>	A user wants to search a student.	
<b>TYPICAL COURSE OF EVENTS:</b>	<b>Actor Action</b>	<b>System Response</b>
	<b>Step 1:</b> A user wants to search a student and proceeds to their main menu.	<b>Step 2:</b> The system prompts the user with a main menu, with an option 'Students'.
	<b>Step 3:</b> The user selects the option 'Students'.	<b>Step 4:</b> The system prompts the user with a list of students', which have been retrieved from the <u>Student</u> table, and the option to search a specific student by means of entering their details into a field.
	<b>Step 5:</b> The user enters the relevant details into the search field.	<b>Step 6:</b> The system prompts the user with a list of students from the <u>Student</u> table with details similar to the ones the user entered into the search field.
	<b>Step 7:</b> The user selects the specific student that they are looking for.	<b>Step 8:</b> The system prompts the user with the students full set of details, such as: <ul style="list-style-type: none"> <li>Student Number</li> <li>Name</li> <li>Surname</li> <li>Date of Birth</li> <li>Phone Number</li> </ul>

		<ul style="list-style-type: none"> <li>• Degree</li> <li>• Year of Study</li> <li>• Race</li> <li>• Home Language</li> <li>• Residence</li> </ul> <p>Which have been retrieved from the <b>Student</b> and <b>Person</b> tables which are linked via the PersonID key, with the option to 'Return to Search' section or to 'Return to Main Menu'.</p>
	<b>Step 9:</b> The user selects the option to 'Return to Main Menu'.	<b>Step 10:</b> The system displays to the user with the main menu
<b>ALTERNATE COURSES:</b>	<b>Alt-Step 6:</b> There are no students that match that description, the system prompts the user to try different search details. → Go to <b>Step 5</b>	
<b>CONCLUSION:</b>	The user has searched for a particular student on the system and viewed the students' details. Once the user is satisfied, the system prompts the user with the main menu	
<b>POST-CONDITION:</b>	The user has searched for a student.	
<b>BUSINESS RULES</b>	<ul style="list-style-type: none"> <li>• None</li> </ul>	
<b>IMPLEMENTATION CONSTRAINTS AND SPECIFICATIONS</b>	<ul style="list-style-type: none"> <li>• The student's username and password must be kept private.</li> </ul>	
<b>ASSUMPTIONS:</b>	<ul style="list-style-type: none"> <li>• None</li> </ul>	
<b>OPEN ISSUES:</b>	None	

<b>USE CASE NAME:</b>	Update Student	<b>USE CASE TYPE</b>
<b>USE CASE ID:</b>	3.3	Business Requirements: <input type="checkbox"/>
<b>PRIORITY:</b>	Medium	System Analysis: <input checked="" type="checkbox"/>
<b>SOURCE:</b>	TuksRes Women In Leadership Academy	System Design: <input type="checkbox"/>
<b>PRIMARY BUSINESS ACTOR</b>	Student	
<b>PRIMARY SYSTEM ACTOR</b>	None	
<b>OTHER PARTICIPATING ACTORS:</b>	<ul style="list-style-type: none"> <li>None</li> </ul>	
<b>OTHER INTERESTED STAKEHOLDERS:</b>	<ul style="list-style-type: none"> <li>None</li> </ul>	
<b>DESCRIPTION:</b>	<p>This use case describes the event of a student updating their details on the TRWLA system. The use case begins when the student has logged onto the system and wants to update their details. They will then update and save their relevant details, and the use case will conclude when they have successfully updated their details.</p>	
<b>PRE-CONDITION:</b>	The user is logged into the system.	
<b>TRIGGER:</b>	The student wants to update their details.	
<b>TYPICAL COURSE OF EVENTS:</b>	<b>Actor Action</b>	<b>System Response</b>
	<b>Step 1:</b> A student wants to update their details and proceeds to the main menu.	<b>Step 2:</b> The system prompts the student with a main menu, with an option 'My Profile'.
	<b>Step 3:</b> The student selects the option 'My Profile'.	<b>Step 4:</b> The system prompts the student with all their details in an editable format, from the <u>Student</u> and <u>Person</u> tables with the options to 'Save', 'Change Password' or 'Cancel' (Invoke UC 3.2)
	<b>Step 5:</b> The student changes any of their details in the given fields, including <ul style="list-style-type: none"> <li>Student Number</li> <li>Name</li> <li>Surname</li> <li>Date of Birth</li> <li>Phone Number</li> <li>Degree</li> <li>Year of Study</li> <li>Race</li> <li>Home Language</li> </ul>	<b>Step 6:</b> The system verifies that all required fields have been filled in and that the fields contain valid details: <ul style="list-style-type: none"> <li>Student Number (8 numeric characters)</li> <li>Name (Maximum 35 characters)</li> <li>Surname (Maximum 35 characters)</li> <li>Date of Birth (Maximum 10 characters in the format CCYY-MM-DD)</li> <li>Phone Number (Max 15</li> </ul>

	<ul style="list-style-type: none"> <li>Residence and selects the 'Save' option.</li> </ul>	characters (No spaces)) <ul style="list-style-type: none"> <li>Degree (Maximum 35 characters)</li> <li>Year of Study (Maximum 10 characters in the format CCYY-MM-DD)</li> <li>Race (Maximum 35 characters)</li> <li>Home Language (Maximum 35 characters)</li> <li>Residence(Selected from predetermined options)</li> </ul>
		<b>Step 7:</b> The system updates the student details into the <b>Student</b> and <b>Person</b> tables confirms with a notification that update was successful.
<b>ALTERNATE COURSES:</b>	<b>Alt-Step 5:</b> The student selects the 'Cancel' option. → Go to <b>Step 4</b>	
	<b>Alt-Step 7:</b> The verification of students' details was unsuccessful and the system prompts the user to correct their information. → Go to <b>Step 5</b>	
<b>CONCLUSION:</b>	The student has changed all the details in their profile that needed to be changed, and the system verifies that all the changes made were complete and valid. When the verification is successful, the system confirms with a notification that update was successful.	
<b>POST-CONDITION:</b>	The student has successfully updated their details.	
<b>BUSINESS RULES</b>	<ul style="list-style-type: none"> <li>Students can only update their own details.</li> </ul>	
<b>IMPLEMENTATION CONSTRAINTS AND SPECIFICATIONS</b>	<ul style="list-style-type: none"> <li>The student must be registered on the system</li> </ul>	
<b>ASSUMPTIONS:</b>	<ul style="list-style-type: none"> <li>None</li> </ul>	
<b>OPEN ISSUES:</b>	None	

<b>USE CASE NAME:</b>	Delete Student	<b>USE CASE TYPE</b>
<b>USE CASE ID:</b>	3.4	Business Requirements: <input type="checkbox"/>
<b>PRIORITY:</b>	Low	System Analysis: <input checked="" type="checkbox"/>
<b>SOURCE:</b>	TuksRes Women In Leadership Academy	System Design: <input type="checkbox"/>
<b>PRIMARY BUSINESS ACTOR</b>	Admin	
<b>PRIMARY SYSTEM ACTOR</b>	None	
<b>OTHER PARTICIPATING ACTORS:</b>	<ul style="list-style-type: none"> <li>None</li> </ul>	
<b>OTHER INTERESTED STAKEHOLDERS:</b>	<ul style="list-style-type: none"> <li>None</li> </ul>	
<b>DESCRIPTION:</b>	<p>This use case describes the event of admin deleting a student. The use case begins when admin has logged in and wants to delete a student. Admin will search, select and delete the relevant student. Before the student is deleted, the system will warn admin that they are about to permanently delete that student from the system, and admin must confirm the deletion before the use case can conclude. The use case concludes when the student has been successfully deleted and the system confirms with a notification that the student has been deleted.</p>	
<b>PRE-CONDITION:</b>	The admin is logged into the system.	
<b>TRIGGER:</b>	Admin wants to delete a student.	
<b>TYPICAL COURSE OF EVENTS:</b>	<b>Actor Action</b>	<b>System Response</b>
	<b>Step 1:</b> Admin wants to delete a student and proceeds to the main menu section.	<b>Step 2:</b> The system prompts Admin with a main menu, with an option 'Students'
	<b>Step 3:</b> Admin selects the option 'Students'.	<b>Step 4:</b> The system displays to admin a list of students as retrieved from the <u>Person</u> and <u>Student</u> tables, and prompts admin to search a student (Invoke UC 3.2).
		<b>Step 5:</b> The system prompts admin with the particular students details as retrieved from the <u>Student</u> and <u>Person</u> tables with the options to 'Delete' or 'Cancel'
	<b>Step 6:</b> Admin selects the 'Delete' option.	<b>Step 7:</b> The system displays a warning that admin is about to permanently delete a student, with the options to 'Confirm Deletion' or to 'Cancel'.

	<b>Step 8:</b> Admin selects the option to 'Confirm Deletion'.	<b>Step 9:</b> The system deletes the student from the <u>Person</u> and <u>Student</u> tables.
		<b>Step 10:</b> The system confirms with a notification that the student has been successfully removed from the system.
<b>ALTERNATE COURSES:</b>	<b>Alt-Step 8:</b> Admin selects the option to 'Cancel' → Go to <b>Step 5</b>	
<b>CONCLUSION:</b>	The system removes the student from the <u>Student</u> and <u>Person</u> tables, and confirms that the student has been successfully removed from the system.	
<b>POST-CONDITION:</b>	Admin has successfully deleted a student.	
<b>BUSINESS RULES</b>	<ul style="list-style-type: none"> <li>Only admin can delete a student</li> </ul>	
<b>IMPLEMENTATION CONSTRAINTS AND SPECIFICATIONS</b>	<ul style="list-style-type: none"> <li>Admin must be given administrator login details at the implementation of the system.</li> <li>A delete cascade must be implemented to ensure that all details relating to the particular student are deleted across all tables.</li> </ul>	
<b>ASSUMPTIONS:</b>	<ul style="list-style-type: none"> <li>None</li> </ul>	
<b>OPEN ISSUES:</b>	None	

<b>USE CASE NAME:</b>	Generate Graduate List	<b>USE CASE TYPE</b>
<b>USE CASE ID:</b>	3.5	Business Requirements: <input type="checkbox"/>
<b>PRIORITY:</b>	Low	System Analysis: <input checked="" type="checkbox"/>
<b>SOURCE:</b>	TuksRes Women In Leadership Academy	System Design: <input type="checkbox"/>
<b>PRIMARY BUSINESS ACTOR</b>	Admin	
<b>PRIMARY SYSTEM ACTOR</b>	None	
<b>OTHER PARTICIPATING ACTORS:</b>	<ul style="list-style-type: none"> <li>None</li> </ul>	
<b>OTHER INTERESTED STAKEHOLDERS:</b>	<ul style="list-style-type: none"> <li>None</li> </ul>	
<b>DESCRIPTION:</b>	<p>This use case describes the event admin generating the graduate list. The use case begins when admin has logged into the system and wants to generate the graduate list. The system will check student progress and create a list of students that have 100% progress. All students with 100% progress will be inserted into the graduate table. The use case concludes when a graduate list has been successfully generated.</p>	
<b>PRE-CONDITION:</b>	Admin has logged into the system.	
<b>TRIGGER:</b>	Admin wants to generate a graduate list.	
<b>TYPICAL COURSE OF EVENTS:</b>	<b>Actor Action</b>	<b>System Response</b>
	<b>Step 1:</b> Admin wants to generate a graduate list and proceeds to the main menu.	<b>Step 2:</b> The system prompts the user with a main menu, with an option 'Students'.
	<b>Step 3:</b> Admin selects the 'Students' option	<b>Step 4:</b> The system prompts admin with the student section and the option to 'Generate a Graduate List'.
	<b>Step 5:</b> Admin selects the option to 'Generate a Graduate List'	<b>Step 5:</b> The system runs through the <u>Student</u> table and inserts all students with 100% progress into the <u>Graduate</u> table.
		<b>Step 6:</b> The system displays the list of graduates from the <u>Graduate</u> table whose <ul style="list-style-type: none"> <li>GraduationYear</li> </ul> Is the same as the current year, and sends admin the list of graduates with all their details from the <u>Student</u> and <u>Person</u> tables.
<b>ALTERNATE COURSES:</b>	None	

<b>CONCLUSION:</b>	The system displays the list of graduates to admin and emails the graduate list to admins email address.
<b>POST-CONDITION:</b>	A graduate list has been successfully generated.
<b>BUSINESS RULES</b>	<ul style="list-style-type: none"><li>• Students must attend 5/6 lectures, 1 community outreach, 1 function and the gender based violence workshop throughout the year.</li></ul>
<b>IMPLEMENTATION CONSTRAINTS AND SPECIFICATIONS</b>	<ul style="list-style-type: none"><li>• None</li></ul>
<b>ASSUMPTIONS:</b>	<ul style="list-style-type: none"><li>• None</li></ul>
<b>OPEN ISSUES:</b>	None



<b>USE CASE NAME:</b>	Add Student Type	<b>USE CASE TYPE</b>
<b>USE CASE ID:</b>	3.6	Business Requirements: <input type="checkbox"/>
<b>PRIORITY:</b>	High	System Analysis: <input checked="" type="checkbox"/>
<b>SOURCE:</b>	TuksRes Women In Leadership Academy	System Design: <input type="checkbox"/>
<b>PRIMARY BUSINESS ACTOR</b>	Admin	
<b>PRIMARY SYSTEM ACTOR</b>	None	
<b>OTHER PARTICIPATING ACTORS:</b>	<ul style="list-style-type: none"> <li>None</li> </ul>	
<b>OTHER INTERESTED STAKEHOLDERS:</b>	<ul style="list-style-type: none"> <li>None</li> </ul>	
<b>DESCRIPTION:</b>	<p>This use case describes the event of admin adding a student type on the system. The use case begins when admin wants to add a new student type on the system. Once admin has entered all the required details of the new student type, the system will have to verify that all the information entered is complete and valid, and will then insert the information into the <b>StudentType</b> table. The use case concludes when the system confirms that a new student type has been added to the system.</p>	
<b>PRE-CONDITION:</b>	Admin is logged into the system	
<b>TRIGGER:</b>	Admin wants to add a new student type on the system	
<b>TYPICAL COURSE</b>	Actor Action	System Response
<b>OF EVENTS:</b>	<b>Step 1:</b> Admin wants to add a new student type on the system and proceeds to the main menu.	<b>Step 2:</b> The system prompts admin with a main menu with the option 'Students'
	<b>Step 3:</b> Admin selects the option 'Students'	<b>Step 4:</b> The system prompts admin with the student section, and an option 'Student Types'
	<b>Step 5:</b> Admin selects the option 'Student Types'.	<b>Step 6:</b> The system prompts admin with the student type section, with the option to 'Add Student Type' or to search.
	<b>Step 7:</b> Admin selects the option to 'Add Student Type'	<b>Step 8:</b> The system prompts admin to enter all the relevant details of the new student type with the option to 'Add Student Type'

	<b>Step 9:</b> Admin enters all the relevant details, including: <ul style="list-style-type: none"> <li>• StudentTypeName</li> <li>• Description</li> </ul> and selects the option to 'Add Student Type'.	<b>Step 10:</b> The system verifies that all the required fields have been entered with the correct details: Name (Maximum 35 characters) Description (Maximum 300 characters)
		<b>Step 11:</b> The system inserts the relevant data from the fields into the <b>StudentType</b> table and confirms with a notification that the student type has been successfully added to the system.
<b>ALTERNATE COURSES:</b>	<b>Alt-Step 11:</b> The information was not entered correctly and the system prompts admin to re-enter the incorrect information. <b>→ Go to Step 9</b>	
<b>CONCLUSION:</b>	Admin has successfully added a student type which has been inserted into the <b>StudentType</b> table. The system confirms to admin that the student type has been successfully added to the system.	
<b>POST-CONDITION:</b>	A student type has been added to the system.	
<b>BUSINESS RULES</b>	<ul style="list-style-type: none"> <li>• Only admin may add a student type</li> </ul>	
<b>IMPLEMENTATION CONSTRAINTS AND SPECIFICATIONS</b>	<ul style="list-style-type: none"> <li>• Admin must be given administrator login details at the implementation of the system.</li> </ul>	
<b>ASSUMPTIONS:</b>	<ul style="list-style-type: none"> <li>• None</li> </ul>	
<b>OPEN ISSUES:</b>	None	

<b>USE CASE NAME:</b>	Search Student Type	<b>USE CASE TYPE</b>
<b>USE CASE ID:</b>	3.7	Business Requirements: <input type="checkbox"/>
<b>PRIORITY:</b>	High	System Analysis: <input checked="" type="checkbox"/>
<b>SOURCE:</b>	TuksRes Women In Leadership Academy	System Design: <input type="checkbox"/>
<b>PRIMARY BUSINESS ACTOR</b>	Admin	
<b>PRIMARY SYSTEM ACTOR</b>	None	
<b>OTHER PARTICIPATING ACTORS:</b>	<ul style="list-style-type: none"> <li>None</li> </ul>	
<b>OTHER INTERESTED STAKEHOLDERS:</b>	<ul style="list-style-type: none"> <li>None</li> </ul>	
<b>DESCRIPTION:</b>	<p>This use case describes the event of admin searching a student type. The use case begins when admin logs onto the system and wants to search a student type. Admin will search for a student type and gather all the relevant information the she is looking for. The use case concludes when admin has searched the student type and the system redirects her to the main menu.</p>	
<b>PRE-CONDITION:</b>	Admin is logged into the system.	
<b>TRIGGER:</b>	Admin wants to search a student type.	
<b>TYPICAL COURSE</b>	Actor Action	System Response
<b>OF EVENTS:</b>	<b>Step 1:</b> Admin wants to search a student type and proceeds to the main menu.	<b>Step 2:</b> The system prompts Admin with a main menu, with an option 'Students'.
	<b>Step 3:</b> The user selects the option 'Students'.	<b>Step 4:</b> The system prompts admin with the student section and the option 'Student Types'.
	<b>Step 5:</b> Admin selects the option 'Student Types'.	<b>Step 6:</b> The system prompts the user with a list of student types from the <b>StudentType</b> table and the option to search a Student type by entering details into the search field.
	<b>Step 7:</b> Admin enters the relevant details of the student type they are searching for into the search field.	<b>Step 8:</b> The system prompts admin with a list of Student types and their details such as: <ul style="list-style-type: none"> <li>Name</li> </ul> from the <b>StudentType</b> table that have similar details as the ones entered in the search field by admin.

	<b>Step 9:</b> Admin selects the particular student type that they're looking for.	<b>Step 10:</b> The system prompts admin with the student types details including: <ul style="list-style-type: none"> <li>• Name</li> <li>• Description</li> </ul> from the <b>StudentType</b> table of the selected Student type, with the options to 'Save', 'Delete' or 'Cancel'
	<b>Step 11:</b> Admin selects the option to 'Cancel'	<b>Step 12:</b> The system prompts admin with the main menu.
<b>ALTERNATE COURSES:</b>	<b>Alt-Step 8:</b> There are no Student types that match that description, the system prompts admin to try different search details. → Go to <b>Step 7</b>	
<b>CONCLUSION:</b>	Admin has searched the relevant student type and is satisfied with the search results. The system prompts admin with the main menu	
<b>POST-CONDITION:</b>	The admin has searched for a student type.	
<b>BUSINESS RULES</b>	<ul style="list-style-type: none"> <li>• None</li> </ul>	
<b>IMPLEMENTATION CONSTRAINTS AND SPECIFICATIONS</b>	<ul style="list-style-type: none"> <li>• Admin must be given administrator login details at the implementation of the system.</li> </ul>	
<b>ASSUMPTIONS:</b>	<ul style="list-style-type: none"> <li>• None</li> </ul>	
<b>OPEN ISSUES:</b>	None	

<b>USE CASE NAME:</b>	Update Student Type	<b>USE CASE TYPE</b>
<b>USE CASE ID:</b>	3.8	Business Requirements: <input type="checkbox"/>
<b>PRIORITY:</b>	Medium	System Analysis: <input checked="" type="checkbox"/>
<b>SOURCE:</b>	TuksRes Women In Leadership Academy	System Design: <input type="checkbox"/>
<b>PRIMARY BUSINESS ACTOR</b>	Admin	
<b>PRIMARY SYSTEM ACTOR</b>	None	
<b>OTHER PARTICIPATING ACTORS:</b>	<ul style="list-style-type: none"> <li>None</li> </ul>	
<b>OTHER INTERESTED STAKEHOLDERS:</b>	<ul style="list-style-type: none"> <li>None</li> </ul>	
<b>DESCRIPTION:</b>	<p>This use case describes the event of admin updating the details of the student type on the TRWLA system. The use case begins when the admin has logged onto the system and wants to update the student type. They will then search the relevant student type, update the details and then select the option to 'Save'. The system will have to verify that the changed details are complete and valid, and will then update the <u>StudentType</u> table accordingly. The use case will conclude when admin has successfully updated the student type.</p>	
<b>PRE-CONDITION:</b>	The admin is logged into the system.	
<b>TRIGGER:</b>	Admin wants to update a student type.	
<b>TYPICAL COURSE</b>	Actor Action	System Response
<b>OF EVENTS:</b>	<b>Step 1:</b> Admin wants to update a student type and proceeds to the main menu.	<b>Step 2:</b> The system prompts admin with a main menu, with an option 'Students'.
	<b>Step 3:</b> Admin selects the 'Students' option.	<b>Step 4:</b> The system prompts admin with the student section, and prompts the user to search a student type (Invoke UC 2.6).
		<b>Step 8:</b> The system prompts admin with the relevant student type with its details as retrieved from the <u>StudentType</u> table, in editable fields and the option to 'Save', 'Delete' or 'Cancel'.
	<b>Step 9:</b> Admin changes the relevant details such as: <ul style="list-style-type: none"> <li>Name</li> <li>Description</li> </ul>	<b>Step 10:</b> The system verifies that all required fields have been filled in and that the fields contain valid details, namely:

	and selects the 'Save' option.	Name (Maximum 35 characters) Description (Maximum 300 characters)
		<b>Step 11:</b> The system updates the <b><u>StudentType</u></b> table with the newly changed details and confirms with a notification that the student type has been successfully updated.
<b>ALTERNATE COURSES:</b>	<b>Alt-Step 9a:</b> Admin selects the 'Cancel' option. → Go to <b>Step 8</b>	
	<b>Alt-Step 9b:</b> Admin selects the delete option. → Invoke UC 2.8	
	<b>Alt-Step 11:</b> The verification of student type details was unsuccessful and the system prompts the admin to correct their information. → Go to <b>Step 9</b>	
<b>CONCLUSION:</b>	Admin has searched and updated a student type. The system has verified that the details updated are complete and valid and updates the <b><u>StudentType</u></b> table. The system confirms that the student type has been successfully updated.	
<b>POST-CONDITION:</b>	Admin has updated a student type	
<b>BUSINESS RULES</b>	<ul style="list-style-type: none"> <li>• Only admin can update a student type</li> </ul>	
<b>IMPLEMENTATION CONSTRAINTS AND SPECIFICATIONS</b>	<ul style="list-style-type: none"> <li>• Admin must be given administrator login details at the implementation of the system.</li> </ul>	
<b>ASSUMPTIONS:</b>	<ul style="list-style-type: none"> <li>• None</li> </ul>	
<b>OPEN ISSUES:</b>	None	

<b>USE CASE NAME:</b>	Delete Student Type	<b>USE CASE TYPE</b>
<b>USE CASE ID:</b>	3.9	Business Requirements: <input type="checkbox"/>
<b>PRIORITY:</b>	Low	System Analysis: <input checked="" type="checkbox"/>
<b>SOURCE:</b>	TuksRes Women In Leadership Academy	System Design: <input type="checkbox"/>
<b>PRIMARY BUSINESS ACTOR</b>	Admin	
<b>PRIMARY SYSTEM ACTOR</b>	None	
<b>OTHER PARTICIPATING ACTORS:</b>	<ul style="list-style-type: none"> <li>None</li> </ul>	
<b>OTHER INTERESTED STAKEHOLDERS:</b>	<ul style="list-style-type: none"> <li>None</li> </ul>	
<b>DESCRIPTION:</b>	This use case describes the event of admin deleting a Student type. The use case begins when admin has logged in and wants to delete a Student type. Admin will search, select and delete the relevant Student type. The use case concludes when the Student type has been successfully deleted.	
<b>PRE-CONDITION:</b>	The admin is logged into the system.	
<b>TRIGGER:</b>	Admin wants to delete a Student type.	
<b>TYPICAL COURSE</b>	Actor Action	System Response
<b>OF EVENTS:</b>	<b>Step 1:</b> Admin wants to delete a student type and proceeds to the main menu.	<b>Step 2:</b> The system prompts Admin with a main menu, with an option 'Students'
	<b>Step 3:</b> Admin selects the option 'Students'.	<b>Step 4:</b> The system prompts admin with the student section, and prompts the user to search a student type (Invoke UC 2.6).
		<b>Step 5:</b> The system prompts admin with the relevant student type with its details such as: <ul style="list-style-type: none"> <li>Name</li> <li>Description</li> </ul> in editable fields and the option to 'Save', 'Delete' or 'Cancel'.
	<b>Step 6:</b> Admin selects the option to 'Delete' the student type.	<b>Step 7:</b> The system prompts admin with a warning that they are about to permanently delete a student type, with the options to 'Confirm Deletion' or to 'Cancel'.
	<b>Step 8:</b> Admin selects the option to 'Confirm Deletion'.	<b>Step 9:</b> The system removes the relevant student type from the

		<b>StudentType</b> table and confirms with a notification that the student type has been successfully removed from the system.
<b>ALTERNATE COURSES:</b>	<b>Alt-Step 8a:</b> Admin selects the 'Cancel' option. → Go to <b>Step 5</b>	
	<b>Alt-Step 8b:</b> Admin selects the 'Save' option. → Invoke UC 2.7	
	<b>Alt-Step 9:</b> The student type that admin is trying to remove is referenced in another table. A delete restrict is applied to the student type. The system cannot delete a student type that is referenced in another table as it will result in orphan records. The system displays a notification to admin that this student type cannot be deleted. → Go to <b>Step 5</b>	
<b>CONCLUSION:</b>	Admin has searched and deleted the student type. Before the system removed the student type from the system, the system warned admin that they were about to permanently delete the student type and prompted admin to confirm the deletion. Once admin confirmed the deletion the system removed the student type. The system confirms that the student type has been successfully removed from the system.	
<b>POST-CONDITION:</b>	Admin has successfully deleted a student type.	
<b>BUSINESS RULES</b>	<ul style="list-style-type: none"> <li>Only admin can delete a student</li> <li>When a volunteer type is deleted, a delete restrict must be enforced to ensure that no orphan records are left in other tables.</li> </ul>	
<b>IMPLEMENTATION CONSTRAINTS AND SPECIFICATIONS</b>	<ul style="list-style-type: none"> <li>Admin must be given administrator login details at the implementation of the system.</li> </ul>	
<b>ASSUMPTIONS:</b>	<ul style="list-style-type: none"> <li>None</li> </ul>	
<b>OPEN ISSUES:</b>	None	



### 2.3.4 Event Subsystem Narratives

<b>USE CASE NAME:</b>	Add New Venue	<b>USE CASE TYPE</b>
<b>USE CASE ID:</b>	4.1	Business Requirements: <input type="checkbox"/>
<b>PRIORITY:</b>	High	System Analysis: <input checked="" type="checkbox"/>
<b>SOURCE:</b>	TuksRes Women In Leadership Academy	System Design: <input type="checkbox"/>
<b>PRIMARY BUSINESS ACTOR</b>	Volunteer	
<b>PRIMARY SYSTEM ACTOR</b>	None	
<b>OTHER PARTICIPATING ACTORS:</b>	<ul style="list-style-type: none"> <li>User (ERA)</li> </ul>	
<b>OTHER INTERESTED STAKEHOLDERS:</b>	<ul style="list-style-type: none"> <li>Board Members and Management</li> </ul>	
<b>DESCRIPTION:</b>	<p>This use case describes the event of a volunteer adding a new venue to the TRWLA system. The use case begins when the volunteer has logged in and wants to add a new venue to the system. They will enter all the relevant venue details, and if these details are complete and valid the system will add it to the <u>Venue</u> table. The use case concludes when the venue has successfully been added to the system.</p>	
<b>PRE-CONDITION:</b>	The volunteer is logged into the system.	
<b>TRIGGER:</b>	A volunteer wants to add a new venue to the system.	
<b>TYPICAL COURSE</b>	<b>Actor Action</b>	<b>System Response</b>
<b>OF EVENTS:</b>	<b>Step 1:</b> A volunteer wants to add a new venue to the system and proceeds to the main menu.	<b>Step 2:</b> The system prompts the volunteer with a main menu with the option to view 'Events'.
	<b>Step 3:</b> The volunteer selects the option to view 'Events'.	<b>Step 4:</b> The system prompts the volunteer with a list of events as retrieved from the <u>Event</u> table, with the option 'Venues'.

	<p><b>Step 5:</b> The volunteer selects the option to view 'Venues'.</p>	<p><b>Step 6:</b> The system prompts the volunteer with a list of venues details such as:</p> <ul style="list-style-type: none"> <li>• Name</li> </ul> <p>as retrieved from the <u>Venue</u> table, with the option 'Add Venue'.</p>
	<p><b>Step 7:</b> The volunteer selects the option to 'Add Venue'.</p>	<p><b>Step 8:</b> The system prompts the volunteer to enter all the relevant venue details with the options to 'Save' or 'Cancel'.</p>
	<p><b>Step 9:</b> The volunteer enters all the relevant venue details, including:</p> <ul style="list-style-type: none"> <li>• Name</li> <li>• Street Number</li> <li>• Street Name</li> <li>• Suburb</li> <li>• Town/City</li> <li>• Capacity</li> <li>• Accessibility</li> <li>• VenueType</li> </ul> <p>and selects the 'Save' option.</p>	<p><b>Step 10:</b> The system verifies that all required fields have been filled in and that the fields contain valid details:</p> <ul style="list-style-type: none"> <li>• Name (Maximum 35 characters)</li> <li>• Street Number (Max 10 numeric characters)</li> <li>• Street Name (Maximum 35 characters)</li> <li>• Suburb (Maximum 35 characters)</li> <li>• Town/City (Maximum 35 characters)</li> <li>• Capacity (Max 10 numeric characters)</li> <li>• Accessibility</li> <li>• VenueType (Predetermined set of options)</li> </ul>
		<p><b>Step 11:</b> The system inserts the venue details into the <u>Address</u>, <u>Venue</u> and <u>VenueType</u> tables and confirms with a notification that the venue has been successfully added to the system.</p>
<b>ALTERNATE COURSES:</b>	<p><b>Alt-Step 11:</b> The verification of venues details was unsuccessful and the system prompts the volunteer to correct the information.</p> <p>→ Go to <b>Step 9</b></p>	
<b>CONCLUSION:</b>	<p>A volunteer has successfully added a new venue to the TRWLA system. The system has verified that the volunteer entered complete and accurate details, and then inserted those details into the <u>Venue</u> table.</p>	

	The system confirmed with a notification that a new venue has successfully added to the system
<b>POST-CONDITION:</b>	The volunteer has successfully added a new venue to the system.
<b>BUSINESS RULES</b>	<ul style="list-style-type: none"><li>• Only TuksRes approved venues may be used for TRWLA events.</li></ul>
<b>IMPLEMENTATION CONSTRAINTS AND SPECIFICATIONS</b>	<ul style="list-style-type: none"><li>• None</li></ul>
<b>ASSUMPTIONS:</b>	<ul style="list-style-type: none"><li>• None</li></ul>
<b>OPEN ISSUES:</b>	None

<b>USE CASE NAME:</b>	Search Venue	<b>USE CASE TYPE</b>
<b>USE CASE ID:</b>	4.2	Business Requirements: <input type="checkbox"/>
<b>PRIORITY:</b>	High	System Analysis: <input checked="" type="checkbox"/>
<b>SOURCE:</b>	TuksRes Women In Leadership Academy	System Design: <input type="checkbox"/>
<b>PRIMARY BUSINESS ACTOR</b>	User	
<b>PRIMARY SYSTEM ACTOR</b>	None	
<b>OTHER PARTICIPATING ACTORS:</b>	<ul style="list-style-type: none"> <li>None</li> </ul>	
<b>OTHER INTERESTED STAKEHOLDERS:</b>	<ul style="list-style-type: none"> <li>None</li> </ul>	
<b>DESCRIPTION:</b>	<p>This use case describes the event of a user searching for a venue. The use case begins when the user has logged onto the system and wants to search a venue. The user will search the venue and retrieve all the relevant details. The use case concludes when the user is satisfied with the search details and returns to their main menu.</p>	
<b>PRE-CONDITION:</b>	The user is logged into the system.	
<b>TRIGGER:</b>	A user wants to search a venue.	
<b>TYPICAL COURSE</b>	<b>Actor Action</b>	<b>System Response</b>
<b>OF EVENTS:</b>	<b>Step 1:</b> A user wants to search a venue and proceeds to the main menu.	<b>Step 2:</b> The system prompts the user with a main menu, with an option 'Events'.
	<b>Step 3:</b> The user selects the option 'Events'.	<b>Step 4:</b> The system prompts the volunteer with a list of events as retrieved from the <u>Event</u> table, with the option 'Venues'.
	<b>Step 5:</b> The volunteer selects the option to view 'Venues'.	<b>Step 6:</b> The system prompts the user with a list of venues which have been retrieved from the <u>Venue</u> table, and the option to search a specific venue by means of entering its details into a field

	<b>Step 7:</b> The user enters the relevant details into the search field.	<b>Step 8:</b> The system prompts the user with a list of venues from the <u>Venue</u> table with details similar to the ones the user entered into the search field.
	<b>Step 9:</b> The user selects the specific venue that they are looking for.	<b>Step 10:</b> The system prompts the user with the venues full set of details, such as: <ul style="list-style-type: none"> <li>• Name</li> <li>• Street Number</li> <li>• Street Name</li> <li>• Suburb</li> <li>• Town/City</li> <li>• Capacity</li> <li>• Accessibility</li> <li>• VenueType</li> </ul> As retrieved from the <u>Address</u> , <u>Venue</u> and <u>VenueType</u> tables, with the option the option to 'Return to Search' section or to 'Return to Main Menu'.
	<b>Step 11:</b> The user selects the option to 'Return to Main Menu'.	<b>Step 12:</b> The system prompts the user with the main menu
<b>ALTERNATE COURSES:</b>	<b>Alt-Step 8:</b> There are no venues that match that description, the system prompts the user to try different search details. → Go to <b>Step 7</b>	
	<b>Alt-Step 11:</b> The user selects the option to 'Return to Search' → Go to <b>Step 8</b>	
<b>CONCLUSION:</b>	The user has successfully searched for a venue on the TRWLA system. Once the user was satisfied, they returned to the main menu. The system prompts the user with the main menu	
<b>POST-CONDITION:</b>	The user has searched for a venue.	
<b>BUSINESS RULES</b>	<ul style="list-style-type: none"> <li>• None</li> </ul>	
<b>IMPLEMENTATION CONSTRAINTS AND SPECIFICATIONS</b>	<ul style="list-style-type: none"> <li>• None</li> </ul>	

<b>ASSUMPTIONS:</b>	<ul style="list-style-type: none"><li>• None</li></ul>
<b>OPEN ISSUES:</b>	None

<b>USE CASE NAME:</b>	Update Venue	<b>USE CASE TYPE</b>
<b>USE CASE ID:</b>	4.3	Business Requirements: <input type="checkbox"/>
<b>PRIORITY:</b>	Medium	System Analysis: <input checked="" type="checkbox"/>
<b>SOURCE:</b>	TuksRes Women In Leadership Academy	System Design: <input type="checkbox"/>
<b>PRIMARY BUSINESS ACTOR</b>	Volunteer	
<b>PRIMARY SYSTEM ACTOR</b>	None	
<b>OTHER PARTICIPATING ACTORS:</b>	<ul style="list-style-type: none"> <li>None</li> </ul>	
<b>OTHER INTERESTED STAKEHOLDERS:</b>	<ul style="list-style-type: none"> <li>None</li> </ul>	
<b>DESCRIPTION:</b>	<p>This use case describes the event of a volunteer updating a venues details on the TRWLA system. The use case begins when the volunteer has logged onto the system and wants to update the venues' details. They will then search the relevant venue and update and save the relevant details, and the use case will conclude when they have successfully updated the venue.</p>	
<b>PRE-CONDITION:</b>	The volunteer is logged into the system.	
<b>TRIGGER:</b>	The volunteer wants to update a venue.	
<b>TYPICAL COURSE</b>	<b>Actor Action</b>	<b>System Response</b>
<b>OF EVENTS:</b>	<b>Step 1:</b> A user wants to update a venue.	<b>Step 2:</b> The system displays the main menu with the option 'Venues'.
	<b>Step 3:</b> The volunteer selects the option 'Venues'	<b>Step 4:</b> The system displays to admin a list of venues as retrieved from the <u>Venue</u> tables, and prompts the volunteer to search for a venue (Invoke UC 4.2).
		<b>Step 5:</b> The system displays the appropriate venues details in an editable format with the options to 'Save' or 'Cancel'

	<p><b>Step 6:</b> The user updates the appropriate details such as:</p> <ul style="list-style-type: none"> <li>• Name</li> <li>• Street Number</li> <li>• Street Name</li> <li>• Suburb</li> <li>• Town/City</li> <li>• Capacity</li> <li>• Accessibility</li> <li>• VenueType</li> </ul> <p>and chooses the option to 'Save'</p>	<p><b>Step 7:</b> The system verifies that all required fields have been filled in and that the fields contain valid details:</p> <ul style="list-style-type: none"> <li>• Name (Maximum 35 characters)</li> <li>• Street Number (Max 10 numeric characters)</li> <li>• Street Name (Maximum 35 characters)</li> <li>• Suburb (Maximum 35 characters)</li> <li>• Town/City (Maximum 35 characters)</li> <li>• Capacity (Max 10 numeric characters)</li> <li>• Accessibility</li> <li>• VenueType (Predetermined set of options)</li> </ul>
		<p><b>Step 6:</b> The system updates the venue details into the <b>Address</b>, <b>Venue</b> and <b>VenueType</b> tables confirms with a notification that the update was successful.</p>
ALTERNATE COURSES:	<p><b>Alt-Step 6:</b> The user selects the 'Cancel' option.</p> <p>→ Go to <b>Step 5</b></p>	
	<p><b>Alt-Step 6:</b> The verification of venues' details was unsuccessful and the system prompts the user to correct their information.</p> <p>→ Go to <b>Step 6</b></p>	
CONCLUSION:	<p>The user has searched and updated a venue. The system has verified, that the user has entered complete and valid details, and has updated those details in the <b>Address</b>, <b>Venue</b> and <b>VenueType</b> tables. The system confirmed with a notification that the update was successful.</p>	
POST-CONDITION:	<p>The volunteer has updated a venue.</p>	
BUSINESS RULES	<ul style="list-style-type: none"> <li>• None</li> </ul>	
IMPLEMENTATION CONSTRAINTS AND SPECIFICATIONS	<ul style="list-style-type: none"> <li>• The venue must be registered on the system</li> </ul>	



<b>ASSUMPTIONS:</b>	<ul style="list-style-type: none"><li>• None</li></ul>
<b>OPEN ISSUES:</b>	None

<b>USE CASE NAME:</b>	Delete Venue	<b>USE CASE TYPE</b>
<b>USE CASE ID:</b>	4.4	Business Requirements: <input type="checkbox"/>
<b>PRIORITY:</b>	Low	System Analysis: <input checked="" type="checkbox"/>
<b>SOURCE:</b>	TuksRes Women In Leadership Academy	System Design: <input type="checkbox"/>
<b>PRIMARY BUSINESS ACTOR</b>	Volunteer	
<b>PRIMARY SYSTEM ACTOR</b>	None	
<b>OTHER PARTICIPATING ACTORS:</b>	<ul style="list-style-type: none"> <li>None</li> </ul>	
<b>OTHER INTERESTED STAKEHOLDERS:</b>	<ul style="list-style-type: none"> <li>None</li> </ul>	
<b>DESCRIPTION:</b>	<p>This use case describes the event of a volunteer deleting a venue. The use case begins when a volunteer has logged in and wants to delete a venue. The volunteer will search, select and delete the relevant venue. The system will confirm that the volunteer is sure they want to delete the venue before removing it. Once the volunteer has confirmed the deletion, the system delete the venue from the <u>Venue</u> table and all information relating to the venue across different tables. The use case concludes when the venue has been successfully deleted and the main menu is displayed.</p>	
<b>PRE-CONDITION:</b>	The volunteer is logged into the system.	
<b>TRIGGER:</b>	A volunteer wants to delete a venue.	
<b>TYPICAL COURSE</b>	<b>Actor Action</b>	<b>System Response</b>
<b>OF EVENTS:</b>	<b>Step 1:</b> A user wants to delete a venue.	<b>Step 2:</b> The system displays the main menu with the option 'Venues'.
	<b>Step 3:</b> The volunteer selects the option 'Venues'	<b>Step 4:</b> The system displays to admin a list of venues as retrieved from the <u>Venue</u> table, and prompts the volunteer to search for a venue(Invoke UC 4.2).
		<b>Step 5:</b> The system displays the appropriate venues details in an

		editable format with the options to 'Save', 'Delete' or 'Cancel'
	<b>Step 6:</b> The volunteer selects the 'Delete' option.	<b>Step 7:</b> The system prompts the volunteer with a warning that they are about to permanently delete a venue, with the options to 'Confirm Deletion' or to 'Cancel'.
	<b>Step 8:</b> The volunteer selects the option to 'Confirm Deletion'.	<b>Step 9:</b> The system removes the venue from the <b>Venue</b> table as well as any relating to that venue in other tables such as <b>Address</b> , and confirms with a notification that the venue has been successfully removed from the system.
<b>ALTERNATE COURSES:</b>	<b>Alt-Step 6:</b> The volunteer selects the option to 'Cancel'. The system redirects admin to the Venue section. → Go to <b>Step 4</b>	
	<b>Alt-Step 6:</b> The volunteer selects the option to 'Save'. The system updates the venue. → Invoke UC 4.3	
	<b>Alt-Step 6:</b> The volunteer selects the option to 'Cancel'. The system redirects admin to the Venue section. → Go to <b>Step 4</b>	
<b>CONCLUSION:</b>	The volunteer has successfully deleted the venue from the system. The system has confirmed that the volunteer definitely wants to delete the venue, and then deleted the venue from the <b>Venue</b> table. The system confirms with a notification that the venue has been successfully removed from the system.	
<b>POST-CONDITION:</b>	The volunteer has successfully deleted a venue.	
<b>BUSINESS RULES</b>	<ul style="list-style-type: none"> <li>Only a volunteer can delete a venue</li> <li>A delete set null must be enforced to ensure that any other tables referencing that venue, for example - <b>Event</b>, do not reference a venue that does not exist.</li> </ul>	
<b>IMPLEMENTATION CONSTRAINTS AND SPECIFICATIONS</b>	<ul style="list-style-type: none"> <li>None</li> </ul>	

<b>ASSUMPTIONS:</b>	<ul style="list-style-type: none"><li>• None</li></ul>
<b>OPEN ISSUES:</b>	None

<b>USE CASE NAME:</b>	Create Event	<b>USE CASE TYPE</b>
<b>USE CASE ID:</b>	4.5	Business Requirements: <input type="checkbox"/>
<b>PRIORITY:</b>	High	System Analysis: <input checked="" type="checkbox"/>
<b>SOURCE:</b>	TuksRes Women In Leadership Academy	System Design: <input type="checkbox"/>
<b>PRIMARY BUSINESS ACTOR</b>	Volunteer	
<b>PRIMARY SYSTEM ACTOR</b>	None	
<b>OTHER PARTICIPATING ACTORS:</b>	<ul style="list-style-type: none"> <li>• User(ERA)</li> </ul>	
<b>OTHER INTERESTED STAKEHOLDERS:</b>	<ul style="list-style-type: none"> <li>• Students</li> <li>• Facilitators</li> <li>• Guest Speakers</li> </ul>	
<b>DESCRIPTION:</b>	<p>The use case describes how the user will upload the event information onto the system making it visible to all TRWLA members in which the information pertains to. The use case begins once the user has selected a specific event type in which the current event pertains to. The user will input all the information pertaining to the event. Once the user is satisfied with all of the information placed in the event information they will upload the information onto the TRWLA timeline. The user who uploaded the information will have to confirm if the venue for the event is available or not by returning to the event information page before the event commences. Only those who pertain to the event will be able to see the event information. This is based off of the residence in which people stay, the function that is being held, and the community engagement project that the event is aimed towards. The use case concludes when the system confirms that the event has been uploaded onto the TRWLA timeline.</p>	
<b>PRE-CONDITION:</b>	The user has logged into the TRWLA system.	
<b>TRIGGER:</b>	The user wants to create and Event.	
<b>TYPICAL COURSE</b>	<b>Actor Action</b>	<b>System Response</b>
<b>OF EVENTS:</b>	<b>Step 1:</b> The user wants to create and event and proceeds to the main menu.	<b>Step 2:</b> The system prompts the user with the main menu. The option 'Events'

	<b>Step 3:</b> The user selects 'Events'.	<b>Step 4:</b> The system prompts the user with a list of events with the options to 'Create Event', 'Hide Event' or 'Search Event'
	<b>Step 5:</b> The user selects the option to 'Create Event'	<b>Step 6:</b> The system prompts the user to select one of the event types TRWLA has, namely: 'Function', 'Community Engagement' and 'Lecture' with the option to 'Confirm' or 'Return to Events'
	<b>Step 7:</b> The user selects the 'Guest Speaker' option.	<b>Step 8:</b> The system prompts the user to fill in all of the information pertaining to the selected event type, with the options to 'Create Event' or 'Cancel' The user will have to select a venue as retrieved from the <u>Venue</u> table (Invoke UC 4.2) as well as a guest speaker for the event as retrieved from the <u>GuestSpeaker</u> table (Invoke UC 5.2). The user must also specify who must be notified about the new event, this is a list that is drawn from the <u>Residence</u> table.
	<b>Step 9:</b> The user inputs all of the information pertaining to the particular event including selecting a venue and a guest speaker and an 'aimed directed participants'. The user provides the required information: <ul style="list-style-type: none"> <li>• Name of Event</li> <li>• Summary of Event</li> <li>• Detailed Information of Event</li> <li>• Date of Event</li> <li>• Time of Venue</li> <li>• Venue of Event (Max</li> <li>• Guest Speaker Name</li> <li>• Directed Participants.</li> </ul>	<b>Step 10:</b> The system verifies the information provided by the user so that it complies with the following: <ul style="list-style-type: none"> <li>• Name of Event (Max 50 Characters)</li> <li>• Summary of Event (Max 100 Characters)</li> <li>• Detailed Information of Event (Max 300 Characters)</li> <li>• Date of Event (dd/mm/yyyy)</li> <li>• Time of Event (00h00)</li> <li>• Venue of Event (Max 100 Characters)</li> <li>• Guest Speaker Name (Max 60 Characters)</li> <li>• Directed Participants. (Max 50 Characters)</li> </ul>

	and confirms the information for the event by selecting the 'Create Event option..	Invoke 'UC 5.5. Invite Guest Speaker'
		<b>Step 11:</b> The system displays a notification to verify whether the user is sure that the event information is correct with the options to 'Make a Revision', 'Cancel Event Creation' or 'Confirm Event'.
	<b>Step 12:</b> The user selects the option to 'Confirm Event'	<b>Step 13:</b> The system inserts the information into the <b>Function</b> and <b>Event</b> tables and uploads the event information for all TRWLA members to see on the timeline.
<b>ALTERNATE COURSES:</b>	<b>Alt-Step 6:</b> The user does not have the administrative rights to create a specific event based off of its event type. The system will prompt the user to select a relevant event type.  →Go to <b>Step 5</b> .	
	<b>Alt-Step 7a:</b> The user selects to return to the main menu. The system displays the main menu to the user.	
	<b>Alt-Step 7b:</b> The user selects the 'Community Engagement' option.  →Go to <b>Alt-Step 8a</b> .	
	<b>Alt-Step 7c:</b> The user selects the 'Lecture' option.  →Go to <b>Alt-Step 8b</b> .	
	<b>Alt-Step 8a:</b> The system prompts the user to input all information that pertains to the community service event that was selected. The school list is populated from the <b>Venue</b> table. <ul style="list-style-type: none"> <li>● Name of Event</li> <li>● Summary of Event</li> <li>● Detailed Information of Event</li> <li>● Date of Event</li> <li>● Time of Venue</li> <li>● Venue of Event</li> <li>● Community Service Content</li> <li>● Directed Participants.</li> </ul>	

	<p><b>Alt-Step 8b:</b> The system prompts the user to input all information that pertains to the lecture event that was selected. The venue list and residence list are populated based off of the <u>Venue</u> table and <u>Residence</u> table.</p> <ul style="list-style-type: none"> <li>• Name of Event</li> <li>• Summary of Event</li> <li>• Detailed Information of Event</li> <li>• Date of Event</li> <li>• Time of Venue</li> <li>• Venue of Event</li> <li>• Lecture Content</li> <li>• Directed Participants.</li> </ul>
	<p><b>Alt-Step 9:</b> The user sets the directed participants as NA making it available to all users of the TRWLA system.</p>
	<p><b>Alt-Step 10:</b> The user did not provide the information in the correct format.</p> <p>→ Go to <b>Step 8</b>.</p>
	<p><b>Alt-Step 9:</b> The user cancels the event creation.</p> <p>→ Go to <b>Alt-Step 7</b>.</p>
	<p><b>Alt-Step 11a:</b> The user wishes to make amendments to the current event information.</p> <p>→ Go to <b>Step 2, Alt-Step 2a or Alt-Step 2b</b>.</p>
	<p><b>Alt-Step 11b:</b> The user wishes to cancel the event creation.</p>
	<p><b>Alt-Step 13a:</b> Only specific residence members will be able to see the event information displayed on the TRWLA timeline based off of Alt-Step 2b.</p>
<b>CONCLUSION:</b>	<p>The system uploads the event information for all TRWLA members to see on the timeline. The <u>Event</u> table and subsequent tables are updated with the new event information.</p>
<b>POST-CONDITION:</b>	<p>An event has been created.</p>
<b>BUSINESS RULES</b>	<ul style="list-style-type: none"> <li>• Only specific residence students will be able to see information pertaining to lectures at their residence.</li> <li>• A function hosting a guest speaker will be detailed as TBC until the guest speaker RSVPs and the invitation status is updated by the user in UC 5.6</li> </ul>



<b>IMPLEMENTATION CONSTRAINTS AND SPECIFICATIONS</b>	<ul style="list-style-type: none"><li>● Only information pertaining to the current event type will be able to be provided by the user.</li><li>● Only registered venues will be accessible.</li><li>● Only registered residences will be accessible.</li><li>● Only registered guest speakers registered will be able to be seen in the guest speaker list.</li></ul>
<b>ASSUMPTIONS:</b>	<ul style="list-style-type: none"><li>● None</li></ul>
<b>OPEN ISSUES:</b>	None

<b>USE CASE NAME:</b>	Search Event	<b>USE CASE TYPE</b>
<b>USE CASE ID:</b>	4.6	Business Requirements: <input type="checkbox"/>
<b>PRIORITY:</b>	High	System Analysis: <input checked="" type="checkbox"/>
<b>SOURCE:</b>	TuksRes Women In Leadership Academy	System Design: <input type="checkbox"/>
<b>PRIMARY BUSINESS ACTOR</b>	User	
<b>PRIMARY SYSTEM ACTOR</b>	None	
<b>OTHER PARTICIPATING ACTORS:</b>	<ul style="list-style-type: none"> <li>None</li> </ul>	
<b>OTHER INTERESTED STAKEHOLDERS:</b>	<ul style="list-style-type: none"> <li>None</li> </ul>	
<b>DESCRIPTION:</b>	This use case describes the event of a user searching for an Event. The use case begins when the user has logged onto the system and wants to search an Event. The user will search the Event and retrieve all the relevant details. The use case concludes when the user is satisfied with the search details and returns to their main menu.	
<b>PRE-CONDITION:</b>	The user is logged into the system.	
<b>TRIGGER:</b>	A user wants to search an Event.	
<b>TYPICAL COURSE</b>	<b>Actor Action</b>	<b>System Response</b>
<b>OF EVENTS:</b>	<b>Step 1:</b> A user wants to search an Event and proceeds to the main menu.	<b>Step 2:</b> The system prompts the user with a main menu, with an option 'Events'.
	<b>Step 3:</b> The user selects the option 'Events'.	<b>Step 4:</b> The system prompts the volunteer with a list of events as retrieved from the <u>Event</u> table, and the option to search a specific venue by means of entering its details into a field
	<b>Step 7:</b> The user enters the relevant details into the search field.	<b>Step 8:</b> The system prompts the user with a list of Events from the <u>Events</u> table with details similar to the ones the user entered into the search field.

	<p><b>Step 9:</b> The user selects the specific Event that they are looking for.</p>	<p><b>Step 10:</b> The system prompts the user with the venues full set of details, such as:</p> <ul style="list-style-type: none"> <li>• Name of Event</li> <li>• Summary of Event</li> <li>• Detailed Information of Event</li> <li>• Date of Event</li> <li>• Time of Venue</li> <li>• Venue of Event (Max</li> <li>• Guest Speaker Name</li> <li>• Directed Participants.</li> </ul> <p>As retrieved from the <u>GuestSpeaker</u>, <u>Venue</u> and <u>EventType</u> tables, with the option the option to 'Return to Search' section or to 'Return to Main Menu'.</p>
	<p><b>Step 11:</b> The user selects the option to 'Return to Main Menu'.</p>	<p><b>Step 12:</b> The system prompts the user with the main menu</p>
<b>ALTERNATE COURSES:</b>	<p><b>Alt-Step 3:</b> The user selects the TRWLA Timeline option as they are a student and have the student access levels.</p> <p>→ Go to <b>Step 4</b></p>	
	<p><b>Alt-Step 8:</b> There are no events that match the current search specifications.</p> <p>→ Go to <b>Step 4</b></p>	
	<p><b>Alt-Step 10a:</b> The system displays the following information associated with community engagement events:</p> <ul style="list-style-type: none"> <li>• Name of Event</li> <li>• Summary of Event</li> <li>• Detailed Information of Event</li> <li>• Date of Event</li> <li>• Time of Venue</li> <li>• Venue of Event</li> <li>• Community Engagement Content</li> <li>• Directed Participants.</li> </ul>	
	<p><b>Alt-Step 10b:</b> The system displays the following information associated with lecture content events:</p> <ul style="list-style-type: none"> <li>• Name of Event</li> <li>• Summary of Event</li> </ul>	

	<ul style="list-style-type: none"> <li>• Detailed Information of Event</li> <li>• Date of Event</li> <li>• Time of Venue</li> <li>• Venue of Event</li> <li>• Lecture Content</li> <li>• Directed Participants.</li> </ul>
	<b>Alt-Step 11:</b> The user selects the option to 'Return to Search' <b>→ Go to Step 8</b>
<b>CONCLUSION:</b>	The user has successfully searched for an event on the TRWLA system. Once the user was satisfied, they returned to the main menu. The system prompts the user with the main menu
<b>POST-CONDITION:</b>	The user has searched for a venue.
<b>BUSINESS RULES</b>	<ul style="list-style-type: none"> <li>• None</li> </ul>
<b>IMPLEMENTATION CONSTRAINTS AND SPECIFICATIONS</b>	<ul style="list-style-type: none"> <li>• None</li> </ul>
<b>ASSUMPTIONS:</b>	<ul style="list-style-type: none"> <li>• None</li> </ul>
<b>OPEN ISSUES:</b>	None

<b>USE CASE NAME:</b>	RSVP to an Event	<b>USE CASE TYPE</b>
<b>USE CASE ID:</b>	4.7	Business Requirements: <input type="checkbox"/>
<b>PRIORITY:</b>	High	System Analysis: <input checked="" type="checkbox"/>
<b>SOURCE:</b>	TuksRes Women In Leadership Academy	System Design: <input type="checkbox"/>
<b>PRIMARY BUSINESS ACTOR</b>	User	
<b>PRIMARY SYSTEM ACTOR</b>	None	
<b>OTHER PARTICIPATING ACTORS:</b>	<ul style="list-style-type: none"> <li>volunteer</li> </ul>	
<b>OTHER INTERESTED STAKEHOLDERS:</b>	<ul style="list-style-type: none"> <li>Admin</li> </ul>	
<b>DESCRIPTION:</b>	<p>The use case describes how the user RSVP to an event that is displayed on the TRWLA timeline. This use case begins when the user has logged onto the system. The user will be displayed with the main menu. Looking through the upcoming events section, the user will see the timeline of TRWLA and the upcoming events that the organization has. When a user sees an event that they would like to attend they may select on the event to open up the event information. The information is displayed to the user where they can read through everything that was stipulated in UC 4.6. The user may then RSVP to the event by accepting the event at the top of the event or the user may exit the event information and accept the event. The system prompts the user to confirm if they wish to attend the event. Once the user has confirmed, the use case concludes by displaying the TRWLA timeline to the user again prompting them to look at other events.</p>	
<b>PRE-CONDITION:</b>	The user has logged into the TRWLA system.	
<b>TRIGGER:</b>	The user wants to RSVP to a TRWLA upcoming event.	
<b>TYPICAL COURSE</b>	<b>Actor Action</b>	<b>System Response</b>
<b>OF EVENTS:</b>	<b>Step 1:</b> The user wants to RSVP to an upcoming event and proceeds to the main menu.	<b>Step 2:</b> The system prompts the user with the main menu with the option 'Events'

	<b>Step 3:</b> The user selects the option 'Events'	<b>Step 4:</b> The system displays the timeline of TRWLA which details all of the events that will happen in the coming months of the academy as retrieved from the <b>Event</b> table, with the options to search an event, 'Return to Main Menu', 'Cancel Event', 'RSVP Event', 'Hide Event' or 'View Event'. Invoke U.C. 4.6 Search Event.
	<b>Step 5:</b> The user selects the 'RSVP to Event' option.	<b>Step 6:</b> The system prompts the user with a notification to confirm that they wish to RSVP to this event with the option to 'Confirm Attendance' or 'Cancel'.
	<b>Step 7:</b> The user confirms event attendance by selecting the 'Confirm Attendance' option.	<b>Step 8:</b> The system confirms with a notification that the event has been added to the user's timeline. The system updates the <b>RSVP Event</b> table with the information of the student.
<b>ALTERNATE COURSES:</b>	<b>Alt-Step 4:</b> The system displays the logged in user specific events. I.e. A user from Magrietjie will see only events that pertain to Magrietjie and not to events that pertain to Madelief. Events that are not specified will show on every user's timeline.	
	<b>Alt-Step 4:</b> The user immediately RSVPs to a particular event. → Go to <b>Step 5</b> .	
	<b>Alt-Step 6a:</b> The user selects 'Return to Events'. → Go to <b>Step 4</b> .	
	<b>Alt-Step 6b:</b> The user selects 'Return to Main Menu'. The system displays the main menu.	
<b>CONCLUSION:</b>	The system confirms that the event has been added to the user's timeline. The <b>RSVP Event</b> table has been updated with the information of the student.	
<b>POST-CONDITION:</b>	A user has RSVP'd to an event.	
<b>BUSINESS RULES</b>	<ul style="list-style-type: none"> <li>A user may only RSVP to events that they can see on their timeline.</li> </ul>	

<b>IMPLEMENTATION CONSTRAINTS AND SPECIFICATIONS</b>	<ul style="list-style-type: none"><li>● A user may only RSVP to events that are available to their timeline.</li><li>● The system must notify the user when event information changes.</li></ul>
<b>ASSUMPTIONS:</b>	<ul style="list-style-type: none"><li>● None</li></ul>
<b>OPEN ISSUES:</b>	None

<b>USE CASE NAME:</b>	View Event Timeline	<b>USE CASE TYPE</b>
<b>USE CASE ID:</b>	4.8	Business Requirements: <input type="checkbox"/>
<b>PRIORITY:</b>	High	System Analysis: <input checked="" type="checkbox"/>
<b>SOURCE:</b>	TuksRes Women In Leadership Academy	System Design: <input type="checkbox"/>
<b>PRIMARY BUSINESS ACTOR</b>	User	
<b>PRIMARY SYSTEM ACTOR</b>	None	
<b>OTHER PARTICIPATING ACTORS:</b>	<ul style="list-style-type: none"> <li>None</li> </ul>	
<b>OTHER INTERESTED STAKEHOLDERS:</b>	<ul style="list-style-type: none"> <li>None</li> </ul>	
<b>DESCRIPTION:</b>	<p>This use case describes how the user views all of the upcoming events for TRWLA. This use case begins when the user is logged into the system. The user is prompted with the main menu. The user selects 'Events'. The system prompts the user with all events that are upcoming in the form of a timeline. The use case concludes when all of the upcoming events are displayed to the user.</p>	
<b>PRE-CONDITION:</b>	The user has logged into the TRWLA system.	
<b>TRIGGER:</b>	The user wants to view the event timeline.	
<b>TYPICAL COURSE</b>	<b>Actor Action</b>	<b>System Response</b>
<b>OF EVENTS:</b>	<b>Step 1:</b> The user wants to view the event timeline and proceeds to the main menu.	<b>Step 2:</b> The system prompts the user with the main menu. The option 'TRWLA Timeline' is available.
	<b>Step 3:</b> The user selects the option 'TRWLA Timeline'.	<b>Step 4:</b> The system prompts the user with the timeline of the upcoming TRWLA events as retrieves from the <u>Event</u> table with the options to 'Create Event', 'Hide Event', 'Search Event'
<b>ALTERNATE COURSES:</b>	<b>Alt-Step 4:</b> No events have been scheduled yet. No events are displayed.	
<b>CONCLUSION:</b>	The system prompts the user with the timeline of the upcoming TRWLA events which is updated by making use of the <u>Event</u> table.	



<b>POST-CONDITION:</b>	The event timeline has been displayed to the user.
<b>BUSINESS RULES</b>	<ul style="list-style-type: none"><li>• All users may search events.</li></ul>
<b>IMPLEMENTATION CONSTRAINTS AND SPECIFICATIONS</b>	<ul style="list-style-type: none"><li>• Only upcoming events will be shown.</li></ul>
<b>ASSUMPTIONS:</b>	None
<b>OPEN ISSUES:</b>	None

<b>USE CASE NAME:</b>	Update Event Information	<b>USE CASE TYPE</b>
<b>USE CASE ID:</b>	4.9	Business Requirements: <input type="checkbox"/>
<b>PRIORITY:</b>	Medium	System Analysis: <input checked="" type="checkbox"/>
<b>SOURCE:</b>	TuksRes Women In Leadership Academy	System Design: <input type="checkbox"/>
<b>PRIMARY BUSINESS ACTOR</b>	volunteer	
<b>PRIMARY SYSTEM ACTOR</b>	None	
<b>OTHER PARTICIPATING ACTORS:</b>	<ul style="list-style-type: none"> <li>User(ERA)</li> </ul>	
<b>OTHER INTERESTED STAKEHOLDERS:</b>	<ul style="list-style-type: none"> <li>Admin</li> </ul>	
<b>DESCRIPTION:</b>	<p>The use case describes how the user may update the information of an event that has already been uploaded onto the TRWLA timeline. The use case begins when a user logs into the system. If the user has the correct access details when they log into their profile they will see 'Manage Events'. The system will prompt them with the existing events that the current user has created, as well as all other events on the TRWLA system depending on the administrative rights that the current user has. The user selects on the event they want to update where the system will prompt them with the input information that was created during that events initial creation. The user will be able to edit all of these details accordingly. Once the user is satisfied with the updates, the user may prompt the system to upload the new event information onto the system. The use case concludes when the updated information has been added to the TRWLA timeline.</p>	
<b>PRE-CONDITION:</b>	The user has logged into the TRWLA system.	
<b>TRIGGER:</b>	The user wants update event information.	
<b>TYPICAL COURSE</b>	<b>Actor Action</b>	<b>System Response</b>
<b>OF EVENTS:</b>	<b>Step 1:</b> The user wants update event information and proceeds to the main menu.	<b>Step 2:</b> The system prompts the user with the main menu of the TRWLA system with the option 'Events'.
	<b>Step 3:</b> The user selects the option 'Events'.	<b>Step 4:</b> The system displays all of the events that currently exist on the

		TRWLA system as retrieved from the <b>Event</b> table, which are split into events created by the current user and all of the events that exist on the TRWLA system, with the options to 'View Event', 'Edit Event', 'Hide Event', 'Cancel Event', 'Search Event' or 'Return to Main Menu'. Invoke U.C. 4.6 Search Event
	<b>Step 5:</b> The user selects the 'Update Event' option.	<b>Step 6:</b> The system prompts the user with all of the information pertaining to that particular event. The user is able to edit all of this information with the option to 'Update Information'.
	<b>Step 7:</b> The user updates the information that they want to change. This includes: <ul style="list-style-type: none"> <li>• Name of Event</li> <li>• Summary of Event</li> <li>• Detailed Information of Event</li> <li>• Date of Event</li> <li>• Venue of Event</li> <li>• Guest Speaker Name</li> <li>• Directed Participants</li> </ul> and selects the option 'Update Information'	<b>Step 8:</b> The system validates that all the information provided by the user is in the correct format, namely: <ul style="list-style-type: none"> <li>• Name of Event (Max 50 Characters)</li> <li>• Summary of Event (Max 100 Characters)</li> <li>• Detailed Information of Event (Max 300 Characters)</li> <li>• Date of Event (dd/mm/yyyy)</li> <li>• Time of Event (00h00)</li> <li>• Venue of Event (Max 100 Characters)</li> <li>• Guest Speaker Name (Max 60 Characters)</li> <li>• Directed Participants. (Max 50 Characters)</li> </ul>
		<b>Step 9:</b> The system prompts the user to confirm that they wish to update this event.
	<b>Step 10:</b> The user confirms the event update.	<b>Step 11:</b> The system updates the event information in the <b>Event</b> table and respective tables accordingly per <b>EventType</b> table and uploads the new content onto the TRWLA timeline.

		<b>Step 12:</b> The system notifies all users who have already RSVP'd to the event about the updated information. The list of RSVP'd users is taken from the <b><u>RSVP Event</u></b> Table.
<b>ALTERNATE COURSES:</b>	<b>Alt-Step 6a:</b> The information displayed on the page includes the following associated with a community engagement event.	
	<b>Alt-Step 6b:</b> The information displayed on the page includes the following associated with a lecture event.	
	<b>Alt-Step 7:</b> The user cancels the event update. → Go to <b>Step 4</b> .	
	<b>Alt-Step 10:</b> The user cancels the event update. → Go to <b>Step 6</b> .	
<b>CONCLUSION:</b>	The system notifies all users who have already RSVP'd to the event about the updated information. The list of RSVP'd users is taken from the <b><u>RSVP Event</u></b> Table. The information in the <b><u>Event</u></b> table is updated accordingly.	
<b>POST-CONDITION:</b>	Event information has been updated.	
<b>BUSINESS RULES</b>	<ul style="list-style-type: none"> <li>• A user may only update events they have created. This applies to facilitators.</li> <li>• If a user has the respective administrative rights they may update any event. This applies to the management of TRWLA.</li> </ul>	
<b>IMPLEMENTATION CONSTRAINTS AND SPECIFICATIONS</b>	<ul style="list-style-type: none"> <li>• A user must update event details at least 1 hour before the commencement of the event.</li> <li>• The system must notify admin of event changes.</li> <li>• All interested parties must be notified of the event change.</li> </ul>	
<b>ASSUMPTIONS:</b>	<ul style="list-style-type: none"> <li>• None</li> </ul>	
<b>OPEN ISSUES:</b>	None	

<b>USE CASE NAME:</b>	Cancel Event	<b>USE CASE TYPE</b>
<b>USE CASE ID:</b>	4.10	Business Requirements: <input type="checkbox"/>
<b>PRIORITY:</b>	Low	System Analysis: <input checked="" type="checkbox"/>
<b>SOURCE:</b>	TuksRes Women In Leadership Academy	System Design: <input type="checkbox"/>
<b>PRIMARY BUSINESS ACTOR</b>	Volunteer	
<b>PRIMARY SYSTEM ACTOR</b>	None	
<b>OTHER PARTICIPATING ACTORS:</b>	<ul style="list-style-type: none"> <li>User (ERA)</li> </ul>	
<b>OTHER INTERESTED STAKEHOLDERS:</b>	<ul style="list-style-type: none"> <li>Admin</li> </ul>	
<b>DESCRIPTION:</b>	<p>The use case describes how the user may cancel event that has already been uploaded onto the TRWLA timeline. The use case begins when a user logs into the system. If the user has the correct access details when they log into their profile they will see 'Manage Events'. The system will prompt them with the existing events that the current user has created, as well as all other events on the TRWLA system depending on the administrative rights that the current user has. The user selects on the event they want to remove and select 'cancel Event'. The system will prompt the user to confirm that they want to unscheduled the event. The user confirms and the system cancel the event. The use case concludes when the event is removed off of the TRWLA timeline and all users who RSVP'd to the event are notified of the removal.</p>	
<b>PRE-CONDITION:</b>	The user has logged into the TRWLA system.	
<b>TRIGGER:</b>	The user wants to cancel an event.	
<b>TYPICAL COURSE</b>	<b>Actor Action</b>	<b>System Response</b>
<b>OF EVENTS:</b>	<b>Step 1:</b> The user wants cancel an event and proceeds to the main menu.	<b>Step 2:</b> The system prompts the user with the main menu of the TRWLA system. The page displays the option 'Manage Events'.

	<p><b>Step 3:</b> The user selects the option 'Manage Events'.</p>	<p><b>Step 4:</b> The system displays all of the events that currently exist on the TRWLA system as retrieved from the <u>Event</u> table, which is split into events created by the current user and all of the events that exist on the TRWLA system, with the options to 'View Event', 'Edit Event', 'Hide Event', 'Cancel Event', 'Search Event' or 'Return to Main Menu'. Invoke U.C. 4.6 Search Event.</p>
	<p><b>Step 7:</b> The selects 'Cancel Event' on a particular event.</p>	<p><b>Step 8:</b> The system prompts the user to confirm that they wish to cancel this event with the option 'Cancel Event'.</p>
	<p><b>Step 9:</b> The user selects 'Cancel Event'.</p>	<p><b>Step 10:</b> The system removes the event from the TRWLA timeline in the <u>TRWLA Timeline</u> table and updates the <u>Event</u> and <u>RSVP Event</u> tables respectively.</p>
		<p><b>Step 11:</b> The system notifies all users who have already RSVP'd to the event about the removed event. The list of RSVP'd users is taken from the <u>RSVP Event</u> Table</p>
ALTERNATE COURSES:	<p><b>Alt-Step 4:</b> The user immediately selects remove event.</p> <p>→ Go to <b>Step 7</b>.</p>	
	<p><b>Alt-Step 9:</b> The user cancels the event removal.</p> <p>→ Go to <b>Step 4</b>.</p>	
CONCLUSION:	<p>The system notifies all users who have already RSVP'd to the event about the removed event. The list of RSVP'd users is taken from the <u>RSVP Event</u> Table.</p>	
POST-CONDITION:	<p>An event has been cancelled.</p>	
BUSINESS RULES	<ul style="list-style-type: none"> <li>• A user may only cancel events they have created. This applies to facilitators.</li> <li>• If a user has the respective administrative rights they may cancel any event. This applies to the management of TRWLA.</li> </ul>	

<b>IMPLEMENTATION CONSTRAINTS AND SPECIFICATIONS</b>	<ul style="list-style-type: none"><li>• A user may cancel an event at any time.</li><li>• The system must notify admin of event changes.</li><li>• All interested parties must be notified of the event change.</li></ul>
<b>ASSUMPTIONS:</b>	<ul style="list-style-type: none"><li>• None</li></ul>
<b>OPEN ISSUES:</b>	None

<b>USE CASE NAME:</b>	Log Event Attendance	<b>USE CASE TYPE</b>
<b>USE CASE ID:</b>	4.11	Business Requirements: <input type="checkbox"/>
<b>PRIORITY:</b>	High	System Analysis: <input checked="" type="checkbox"/>
<b>SOURCE:</b>	TuksRes Women In Leadership Academy	System Design: <input type="checkbox"/>
<b>PRIMARY BUSINESS ACTOR</b>	Student	
<b>PRIMARY SYSTEM ACTOR</b>	Volunteer	
<b>OTHER PARTICIPATING ACTORS:</b>	<ul style="list-style-type: none"> <li>• None</li> </ul>	
<b>OTHER INTERESTED STAKEHOLDERS:</b>	<ul style="list-style-type: none"> <li>• Admin</li> <li>• Facilitators</li> </ul>	
<b>DESCRIPTION:</b>	<p>This use case describes how the attendance of a student at an event is logged on the system. The use case begins when the student approaches the volunteer to log their attendance to the event. The volunteer is logged into the system. From the main menu, the volunteer can see the option 'Log Attendance'. From here, the volunteer may select the event that they wish to log attendance for. Only events of that day will be available on this page. When the volunteer selects this option, the system will prompt the volunteer with the list of students who have RSVP'd for the event. A student will have to go to the volunteer and ask them to tick their name off of the list which is done by the use searching for the student and physically ticking their name off. If a student did not RSVP to the event but they are attending the event, the volunteer may add the student to the list by searching for their details and then ticking them off of the list accordingly. The use case concludes when the student has successfully logged their attendance to an event.</p>	
<b>PRE-CONDITION:</b>	The volunteer has logged into the TRWLA system.	
<b>TRIGGER:</b>	The student wants to log their attendance at the TRWLA event.	
<b>TYPICAL COURSE</b>	<b>Actor Action</b>	<b>System Response</b>
<b>OF EVENTS:</b>	<b>Step 1:</b> The student wants to log their attendance at the TRWLA event.	<b>Step 2:</b> The system prompts the volunteer with the main menu. The option 'Log Attendance'



		<b>Step 3:</b> The volunteer selects the option 'Log Attendance'.
		<b>Step 4:</b> The system prompts the user with all of the events that are taking place on that particular day. This is drawn from the <b>Event</b> table. The options that are available to the user are: 'Log Attendance' or 'Search Event'
		<b>Step 5:</b> The volunteer selects an event.
		<b>Step 6:</b> The volunteer selects 'Log Event Attendance' option.
		<b>Step 7:</b> The system prompts the user with all of the names of the students who RSVP'd to this particular event as retrieved from the <b>RSVP Event</b> table, with the options to 'Return to Main Menu', 'Add New Student', 'Confirm Attendance' or Search Student via: <ul style="list-style-type: none"> <li>● Student Number</li> <li>● Name/Surname</li> </ul>
		<b>Step 8:</b> The volunteer asks the student if they have RSVP'd to the event.
	<b>Step 9:</b> The student says they have not RSVP'd to the Event.	<b>Step 10:</b> The volunteer selects the 'Add New Student' option.
		<b>Step 12:</b> The system prompts the user with a list of all students who are available to attend the event based off of the directed participants ascertained from the event creation. The options displayed are: <ul style="list-style-type: none"> <li>● Search Student via:               <ul style="list-style-type: none"> <li>○ Student Number</li> <li>○ Name/Surname</li> </ul> </li> <li>● Return to Log Attendance</li> </ul>
		<b>Step 13:</b> The volunteer asks the student for the student's name.

	<b>Step 14:</b> The student provides their name.	<b>Step 15:</b> The volunteer enters the students name into the search bar searching for the student.
		<b>Step 16:</b> The system displays matching persons to that student ascertained from the <b>Person</b> and <b>Student</b> tables.
		<b>Step 17:</b> The volunteer confirms the student's attendance.
		<b>Step 18:</b> The system updates the student's' progress in the <b>StudentProgress</b> table and attendance in the <b>Attendance</b> table and confirms with a notification that the attendance of the student has been logged.
<b>ALTERNATE COURSES:</b>	<b>Alt-Step 9:</b> The student has RSVP'd to the event. <b>→Go to Step 13.</b>	
	<b>Alt-Step 13:</b> The volunteer asks for the student's student number. The student provides their student number. The volunteer inputs the student's student number. <b>→ Go to Step 16.</b>	
<b>CONCLUSION:</b>	The system confirms the event attendance of the student. The student is added to the <b>Attendance</b> table for that event.	
<b>POST-CONDITION:</b>	Student attendance has been logged on the system. The progress of the student has been updated towards them achieving their certification.	
<b>BUSINESS RULES</b>	<ul style="list-style-type: none"> <li>Only students who may attend the specific event will show up on the 'Add New Student' List. I.e. Students of Magrietjie will not show up on the list of a Madelief Lecture Event.</li> <li>A student must present their student card when logging event attendance.</li> <li>The student will receive 12.5 points towards their progress for every event that they attend.</li> <li>The student must attend:               <ol style="list-style-type: none"> <li>5 out of 6 content sessions</li> <li>1 community engagement</li> <li>1 function</li> <li>1 Gender Based Violence Workshop</li> </ol> </li> </ul>	

<b>IMPLEMENTATION CONSTRAINTS AND SPECIFICATIONS</b>	<ul style="list-style-type: none"><li>● Only facilitators and management will be able to log event attendance of a student.</li><li>● A student must present their student card when logging event attendance.</li></ul>
<b>ASSUMPTIONS:</b>	<ul style="list-style-type: none"><li>● None</li></ul>
<b>OPEN ISSUES:</b>	None

<b>USE CASE NAME:</b>	Send Notification	<b>USE CASE TYPE</b>
<b>USE CASE ID:</b>	4.12	Business Requirements: <input type="checkbox"/>
<b>PRIORITY:</b>	High	System Analysis: <input checked="" type="checkbox"/>
<b>SOURCE:</b>	TuksRes Women In Leadership Academy	System Design: <input type="checkbox"/>
<b>PRIMARY BUSINESS ACTOR</b>	Volunteer	
<b>PRIMARY SYSTEM ACTOR</b>	None	
<b>OTHER PARTICIPATING ACTORS:</b>	<ul style="list-style-type: none"> <li>• User (ERA)</li> </ul>	
<b>OTHER INTERESTED STAKEHOLDERS:</b>	<ul style="list-style-type: none"> <li>• Admin</li> <li>• Facilitators</li> </ul>	
<b>DESCRIPTION:</b>	<p>This use case describes how the user send a notification to other users. The use case begins when the user is logged into the system. The system prompts the user with the main menu with the option 'Create Notification'. The user selects this option and the system prompts the user with a list of all of users of the TRWLA system. The user selects a batch of users to send a notification out to. The user types out the notification that they would like to send. The user sends the notification. The use case concludes when the system verifies that the notification has been sent.</p>	
<b>PRE-CONDITION:</b>	The volunteer has logged into the TRWLA system.	
<b>TRIGGER:</b>	The user wants to send a notification.	
<b>TYPICAL COURSE</b>	<b>Actor Action</b>	<b>System Response</b>
<b>OF EVENTS:</b>	<b>Step 1:</b> The user wants to send a notification and proceeds to the main menu.	<b>Step 2:</b> The system prompts the user with the main menu. The option 'Send Notification' is available.
	<b>Step 3:</b> The user selects the option 'Send Notification'.	<b>Step 4:</b> The system prompts the user with a list of all the users on the TRWLA system. This list is populated through the use of the <u>Person</u> table joined to the <u>Student</u> table and the <u>Volunteer</u> table via the use of PersonID Key, with

		the options to 'Search', 'Confirm Recipients' or 'Return to Main Menu'
	<b>Step 5:</b> The user inputs the information of the users to receive the notification in the search bar.	<b>Step 6:</b> The system displays a list of users that pertain to the information inputted by the user.
	<b>Step 7:</b> The user selects the 'Confirm Recipients' option.	<b>Step 8:</b> The system prompts the user to fill in the message that they would like to send to the recipients, with the options to 'Send Message' or 'Return to Main Menu'
	<b>Step 9:</b> The user fills in the message.	<b>Step 10:</b> The system validates that the message is an appropriate length. <ul style="list-style-type: none"> <li>• Content. (Max 500 characters)</li> </ul>
	<b>Step 11:</b> The user selects the 'Send Message' option.	<b>Step 12:</b> The system prompts the user to confirm the message or to return to edit the message.
	<b>Step 13:</b> The user confirms the message.	<b>Step 14:</b> The system sends the message to the recipients of the message whose contact details were obtained from the <b><u>Volunteer</u></b> table and the <b><u>Student</u></b> table.
		<b>Step 15:</b> The system confirms with a notification that the notification has been sent.
<b>ALTERNATE COURSES:</b>	<b>Alt-Step 5:</b> The user immediately selects the 'Confirm Recipients'. → Go to <b>Step 8</b> .	
	<b>Alt-Step 13:</b> The user selects 'Edit Message'. → Go to <b>Step 8</b> .	
<b>CONCLUSION:</b>	The notification has been sent to the users.	
<b>POST-CONDITION:</b>	A notification has been sent.	
<b>BUSINESS RULES</b>	<ul style="list-style-type: none"> <li>• A facilitator may only send notifications to the residence that they are facilitating.</li> <li>• Admin can send a notification to anyone.</li> </ul>	

<b>IMPLEMENTATION CONSTRAINTS AND SPECIFICATIONS</b>	<ul style="list-style-type: none"><li>● A notification cannot be sent to one user alone. It must be sent to at least 2 users.</li><li>● A notification can be sent at any time.</li></ul>
<b>ASSUMPTIONS:</b>	None
<b>OPEN ISSUES:</b>	None

### 2.3.5 Function Subsystem Narratives

<b>USE CASE NAME:</b>	Register Guest Speaker	<b>USE CASE TYPE</b>
<b>USE CASE ID:</b>	5.1	Business Requirements: <input type="checkbox"/>
<b>PRIORITY:</b>	High	System Analysis: <input checked="" type="checkbox"/>
<b>SOURCE:</b>	TuksRes Women In Leadership Academy	System Design: <input type="checkbox"/>
<b>PRIMARY BUSINESS ACTOR</b>	Volunteers	
<b>PRIMARY SYSTEM ACTOR</b>	None	
<b>OTHER PARTICIPATING ACTORS:</b>	<ul style="list-style-type: none"> <li>None</li> </ul>	
<b>OTHER INTERESTED STAKEHOLDERS:</b>	<ul style="list-style-type: none"> <li>Guest Speaker</li> </ul>	
<b>DESCRIPTION:</b>	<p>This use case describes the event of registering a new Guest Speaker on the system. If the organization has found a new guest speaker to attend events and have obtained their details, Volunteers will indicate to 'Register' a new guest speaker on the system. The user will enter the required information about the guest speaker and the system will validate that the information is in line with the necessary standards. Once the system has validated the information the guest speaker will then be stored on the system to invite them to future functions.</p>	
<b>PRE-CONDITION:</b>	The user must be logged onto the system.	
<b>TRIGGER:</b>	The user wants to register a new guest speaker on the system.	
<b>TYPICAL COURSE</b>	<b>Actor Action</b>	<b>System Response</b>
<b>OF EVENTS:</b>	<b>Step 1:</b> The user wants to register a new guest speaker on the system and proceeds to the main menu.	<b>Step 2:</b> The system displays a main menu with different options to choose from including 'Guest Speaker'.
	<b>Step 3:</b> The user selects the 'Guest Speaker' option.	<b>Step 4:</b> The system displays the Guest Speaker section with a populated list of existing guest speaker's names from the

		<b>GuestSpeaker</b> table as well as a 'Register Guest Speaker' option.
	<b>Step 5:</b> The user selects the 'Register Guest Speaker' option.	<b>Step 6:</b> The system displays empty fields for the user to enter the required information about the guest speaker.
	<b>Step 7:</b> The user enters all of the guest speaker details including: <ul style="list-style-type: none"> <li>• Name</li> <li>• Surname</li> <li>• Gender</li> <li>• Date of Birth</li> <li>• Email Address</li> <li>• Phone Number</li> <li>• Theme Topic</li> <li>• Hourly Rate</li> </ul> and selects the option to 'Confirm Guest Speaker' registration.	<b>Step 8:</b> The system validates that all required fields have been filled in and that the fields contain valid details i.e.: <ul style="list-style-type: none"> <li>• Name (Maximum 35 characters)</li> <li>• Surname (Maximum 35 characters)</li> <li>• Date of Birth (Maximum 10 characters in the format CCYY-MM-DD)</li> <li>• Gender</li> <li>• Email Address (Minimum 7 characters, Maximum 255 characters, must be a full validated email address)</li> <li>• Phone Number (Maximum 15 characters, no spaces included)</li> <li>• Theme Topic (Maximum 100 characters)</li> <li>• Hourly Rate (Maximum 10 characters, currency in Rand)</li> </ul>
		<b>Step 9:</b> The system updates the list and inserts the guest speaker details into the <b>GuestSpeaker</b> and <b>SpeakerType</b> tables.
		<b>Step 10:</b> The system confirms with a notification that registration was successful.
<b>ALTERNATE COURSES:</b>	<b>Alt-Step 8a:</b> If all the required information has not been entered, the system prompts the user to enter the required information. → Go to <b>Step 7</b> .	
	<b>Alt-Step 8b:</b> If the guest speaker already exists on the system, the system will prompt the user that the speaker already exists and will not save the information. → Go to <b>Step 7</b> .	



<b>CONCLUSION:</b>	The system confirms with a notification that registration was successful.
<b>POST-CONDITION:</b>	A new guest speaker has been added to the system.
<b>BUSINESS RULES</b>	<ul style="list-style-type: none"><li>• Only Volunteers may register a guest speaker to the system.</li></ul>
<b>IMPLEMENTATION CONSTRAINTS AND SPECIFICATIONS</b>	<ul style="list-style-type: none"><li>• None</li></ul>
<b>ASSUMPTIONS:</b>	<ul style="list-style-type: none"><li>• The user has all the required information of the guest speaker.</li></ul>
<b>OPEN ISSUES:</b>	None

<b>USE CASE NAME:</b>	Search Guest Speaker	<b>USE CASE TYPE</b>
<b>USE CASE ID:</b>	5.2	Business Requirements: <input type="checkbox"/>
<b>PRIORITY:</b>	High	System Analysis: <input checked="" type="checkbox"/>
<b>SOURCE:</b>	TuksRes Women In Leadership Academy	System Design: <input type="checkbox"/>
<b>PRIMARY BUSINESS ACTOR</b>	Users	
<b>PRIMARY SYSTEM ACTOR</b>	None	
<b>OTHER PARTICIPATING ACTORS:</b>	<ul style="list-style-type: none"> <li>None</li> </ul>	
<b>OTHER INTERESTED STAKEHOLDERS:</b>	<ul style="list-style-type: none"> <li>None</li> </ul>	
<b>DESCRIPTION:</b>	<p>This use case describes the event of searching for a Guest Speaker on the system. The use case begins when a user has logged onto the system and wishes to search for a guest speaker to either invite the guest or view their information on the system. A list will be displayed with all the current guest speakers where the user can choose to either select a guest speaker from the list or search for them using a 'Search' option. Once the user has indicated which guest speaker, the user would like to view, the system will display all of the relevant information pertaining to that guest speaker. The user will then have successfully searched for a guest speaker.</p>	
<b>PRE-CONDITION:</b>	The user must be logged onto the system.	
<b>TRIGGER:</b>	The user wants to search for a guest speaker on the system.	
<b>TYPICAL COURSE</b>	<b>Actor Action</b>	<b>System Response</b>
<b>OF EVENTS:</b>	<b>Step 1:</b> The user wants to search for a guest speaker on the system and proceeds to the main menu.	<b>Step 2:</b> The system displays a main menu with different options to choose from including 'Guest Speaker'.
	<b>Step 3:</b> The user selects the 'Guest Speaker' option.	<b>Step 4:</b> The system displays a list of guest speaker names as retrieved from the <u>GuestSpeaker</u> table.

	<b>Step 5:</b> The user searches for the guest speaker via the 'Search' option.	<b>Step 6:</b> The system displays the name/s of the Guest Speakers' based on the search performed by the user. The search results are based on matching information found in the <u><b>GuestSpeaker</b></u> table.
	<b>Step 7:</b> The user selects the guest speaker from the list.	<b>Step 8:</b> The system displays the selected guest speaker information that it has obtained from the <u><b>GuestSpeaker</b></u> table such as; <ul style="list-style-type: none"> <li>• Name</li> <li>• Surname</li> <li>• Gender</li> <li>• Date of Birth</li> <li>• Email Address</li> <li>• Phone Number</li> <li>• Theme Topic</li> <li>• Hourly Rate</li> </ul>
<b>ALTERNATE COURSES:</b>	<b>Alt-Step 5:</b> The user chooses to search through the Guest Speaker list instead of using the 'Search' option. → Go to <b>Step 7</b> .	
	<b>Alt-Step 6a:</b> If the guest speaker is not on the list, the user can register the guest speaker on the system by selecting 'Register Guest Speaker', based on the admin rights of the user. → Invoke UC 5.1.	
	<b>Alt-Step 6b:</b> If the system does not locate a guest speaker based on the user search, → Go to <b>Step 5</b> .	
<b>CONCLUSION:</b>	The system displays the selected guest speaker information that it has obtained from the <u><b>GuestSpeaker</b></u> table such as; <ul style="list-style-type: none"> <li>• Name</li> <li>• Surname</li> <li>• Date of Birth</li> <li>• Email Address</li> <li>• Phone Number</li> <li>• Theme Topic</li> <li>• Hourly Rate</li> </ul>	
<b>POST-CONDITION:</b>	The user has searched for a guest speaker.	

<b>BUSINESS RULES</b>	<ul style="list-style-type: none"><li>• All users may search for a guest speaker.</li><li>• Only volunteers may register/invite a new guest speaker.</li></ul>
<b>IMPLEMENTATION CONSTRAINTS AND SPECIFICATIONS</b>	<ul style="list-style-type: none"><li>• Guest Speaker information will be displayed in editable format if the user is a Volunteer, in order to update the Guest Speaker details if need be.</li><li>• The options to edit Guest Speaker details will be hidden for student users.</li></ul>
<b>ASSUMPTIONS:</b>	<ul style="list-style-type: none"><li>• None</li></ul>
<b>OPEN ISSUES:</b>	None

<b>USE CASE NAME:</b>	Update Guest Speaker	<b>USE CASE TYPE</b>
<b>USE CASE ID:</b>	5.3	Business Requirements: <input type="checkbox"/>
<b>PRIORITY:</b>	Medium	System Analysis: <input checked="" type="checkbox"/>
<b>SOURCE:</b>	TuksRes Women In Leadership Academy	System Design: <input type="checkbox"/>
<b>PRIMARY BUSINESS ACTOR</b>	Volunteers	
<b>PRIMARY SYSTEM ACTOR</b>	None	
<b>OTHER PARTICIPATING ACTORS:</b>	<ul style="list-style-type: none"> <li>None</li> </ul>	
<b>OTHER INTERESTED STAKEHOLDERS:</b>	<ul style="list-style-type: none"> <li>Guest Speaker</li> </ul>	
<b>DESCRIPTION:</b>	<p>This use case describes the event of updating a Guest Speaker's information on the system. If a guest speaker has notified TRWLA that their details have changed, the user will update the details on the system. The user will first search for the guest speaker via the guest speaker list or the user will search via the 'Search' option. Once the user has indicated which guest speaker the user would like to edit, the system will display the selected guest speaker's information. Based on the admin rights, the information will be in editable format so that the user can update it accordingly. Once the user has updated the information, the system will validate if all the necessary required information has been entered and then store the updated information on the system.</p>	
<b>PRE-CONDITION:</b>	The user must be logged onto the system.	
<b>TRIGGER:</b>	The user wants to update a guest speaker's information on the system.	
<b>TYPICAL COURSE</b>	<b>Actor Action</b>	<b>System Response</b>
<b>OF EVENTS:</b>	<b>Step 1:</b> The user wants to update a guest speaker's information on the system and proceeds to the main menu.	<b>Step 2:</b> The system displays a main menu with different options to choose from.
	<b>Step 3:</b> The user selects the 'Guest Speaker' option.	<b>Step 4:</b> The system displays a list of guest speaker names that has been populated with data from the

		<b>GuestSpeaker</b> table. Invoke UC 5.2 Search Guest Speaker.
	<b>Step 5:</b> The user searches and selects a Guest Speaker from the list.	<b>Step 6:</b> The system displays selected guest speaker's information in editable format.
	<b>Step 7:</b> The user edits the relevant information of the guest speaker, such as: <ul style="list-style-type: none"> <li>• Name</li> <li>• Surname</li> <li>• Gender</li> <li>• Date of Birth</li> <li>• Email Address</li> <li>• Phone Number</li> <li>• Theme Topic</li> <li>• Hourly Rate</li> </ul> and saves changes to the guest speaker.	<b>Step 8:</b> The system validates that all required fields have been filled in, that the fields contain valid details i.e.: <ul style="list-style-type: none"> <li>• Name (Maximum 35 characters)</li> <li>• Surname (Maximum 35 characters)</li> <li>• Gender</li> <li>• Date of Birth (Maximum 10 characters in the format CCYY-MM-DD)</li> <li>• Email Address (Minimum 7 characters, Maximum 255 characters, must be a full validated email address)</li> <li>• Phone Number (Maximum 15 characters, no spaces included)</li> <li>• Theme Topic (Maximum 100 characters)</li> <li>• Hourly Rate (Maximum 10 characters, currency in Rand)</li> </ul>
		<b>Step 9:</b> The system displays a warning message that changes are about to be made.
	<b>Step 10:</b> The user confirms the changes made.	<b>Step 11:</b> The system updates the <b>GuestSpeaker</b> and <b>SpeakerType</b> tables with the updated details and displays a confirmation message that the information has been successfully updated.
<b>ALTERNATE COURSES:</b>	<b>Alt-Step 5:</b> If the guest speaker is not on the list, the user can register the guest speaker on the system by selecting 'Invite New Guest speaker', based on the admin rights of the user.  → Invoke UC 5.1.	

	<p><b>Alt-Step 7:</b> The user indicates to 'Return' and accidentally changed the guest speaker's details.</p> <p>→ Go to <b>Step 9</b></p>
	<p><b>Alt-Step 8:</b> If the required fields have not been correctly entered, the system will display a message stating that information is missing,</p> <p>→ Go to <b>Step 9.</b></p>
	<p><b>Alt-Step 10:</b> The user does not confirm changes and the updated information is not saved.</p> <p>→ Go to <b>Step 4.</b></p>
<b>CONCLUSION:</b>	The system updates the <u><b>GuestSpeaker</b></u> table with the updated details and displays a confirmation message that the information has been successfully updated.
<b>POST-CONDITION:</b>	A guest speaker has been updated on the system.
<b>BUSINESS RULES</b>	<ul style="list-style-type: none"> <li>Only Volunteers may update a guest speaker on the system.</li> </ul>
<b>IMPLEMENTATION CONSTRAINTS AND SPECIFICATIONS</b>	<ul style="list-style-type: none"> <li>Since all users can search for guest speakers on the system, guest speaker information will only be in editable format for Volunteer users and not for Students.</li> </ul>
<b>ASSUMPTIONS:</b>	<ul style="list-style-type: none"> <li>None</li> </ul>
<b>OPEN ISSUES:</b>	None

<b>USE CASE NAME:</b>	Delete Guest Speaker	<b>USE CASE TYPE</b>
<b>USE CASE ID:</b>	5.4	Business Requirements: <input type="checkbox"/>
<b>PRIORITY:</b>	Low	System Analysis: <input checked="" type="checkbox"/>
<b>SOURCE:</b>	TuksRes Women In Leadership Academy	System Design: <input type="checkbox"/>
<b>PRIMARY BUSINESS ACTOR</b>	Volunteers	
<b>PRIMARY SYSTEM ACTOR</b>	None	
<b>OTHER PARTICIPATING ACTORS:</b>	<ul style="list-style-type: none"> <li>None</li> </ul>	
<b>OTHER INTERESTED STAKEHOLDERS:</b>	<ul style="list-style-type: none"> <li>None</li> </ul>	
<b>DESCRIPTION:</b>	<p>This use case describes the event of deleting a Guest Speaker on the system. The use case begins when management has decided to delete a guest speaker from the system and no longer wishes to use their expertise in the future. The user will first search for the guest speaker via the guest speaker list or the user will search via the 'Search' option. Once the user has indicated which guest speaker the user would like to delete, the system will display the selected guest speaker's information. Based on the admin rights, the information will be in editable format so that the user can delete it accordingly. Once the user has indicated to delete guest speaker from the system, the system will display a warning message that the guest speaker will be deleted permanently. Once the user has confirmed the deletion, the system will delete the guest speaker from the system and update the guest speaker list accordingly.</p>	
<b>PRE-CONDITION:</b>	The user must be logged onto the system.	
<b>TRIGGER:</b>	The user wants to delete a guest speaker from the system.	
<b>TYPICAL COURSE</b>	<b>Actor Action</b>	<b>System Response</b>
<b>OF EVENTS:</b>	<b>Step 1:</b> The user wants to delete a guest speaker from the system.	<b>Step 2:</b> The system displays a main menu with different options to choose from.
	<b>Step 3:</b> The user selects the 'Guest Speaker' option.	<b>Step 4:</b> The system displays a list of guest speaker names as retrieved from



		the <b>GuestSpeaker</b> table. Invoke UC 5.2 Search Guest Speaker.
	<b>Step 5:</b> The user searches and selects a Guest Speaker from the list.	<b>Step 6:</b> The system displays selected guest speaker's information in editable format.
	<b>Step 7:</b> The user selects to 'Delete Guest Speaker'.	<b>Step 8:</b> The system displays a notification which warns the user of permanent deletion, and prompts the user to either 'Confirm Deletion' or 'Cancel'
	<b>Step 9:</b> The user selects the option to 'Confirm Deletion'.	<b>Step 10:</b> The system deletes the guest speaker from the <b>GuestSpeaker</b> table and updates the guest speaker list.
		<b>Step 11:</b> The system displays a notification to the user which confirms the deletion of the guest speaker.
<b>ALTERNATE COURSES:</b>	<b>Alt-Step 5:</b> If the guest speaker is not on the list, the user can register the guest speaker on the system by selecting 'Register guest speaker', based on the admin rights of the user. → Invoke UC 5.1.	
	<b>Alt-Step 9:</b> If the user chooses not to confirm deletion, the system will not delete the guest speaker. → Go to <b>Step 4</b> .	
<b>CONCLUSION:</b>	The system displays a notification to the user which confirms the deletion of the guest speaker.	
<b>POST-CONDITION:</b>	The guest speaker has been deleted from the system.	
<b>BUSINESS RULES</b>	<ul style="list-style-type: none"> <li>Only volunteers may delete a guest speaker on the system.</li> </ul>	
<b>IMPLEMENTATION CONSTRAINTS AND SPECIFICATIONS</b>	<ul style="list-style-type: none"> <li>Since all users can search for guest speakers on the system, guest speaker information will only be in editable format for volunteer users and not for students.</li> </ul>	
<b>ASSUMPTIONS:</b>	<ul style="list-style-type: none"> <li>None</li> </ul>	
<b>OPEN ISSUES:</b>	None	

<b>USE CASE NAME:</b>	Invite Guest Speaker	<b>USE CASE TYPE</b>
<b>USE CASE ID:</b>	5.5	Business Requirements: <input type="checkbox"/>
<b>PRIORITY:</b>	High	System Analysis: <input checked="" type="checkbox"/>
<b>SOURCE:</b>	TuksRes Women In Leadership Academy	System Design: <input type="checkbox"/>
<b>PRIMARY BUSINESS ACTOR</b>	Volunteers	
<b>PRIMARY SYSTEM ACTOR</b>	None	
<b>OTHER PARTICIPATING ACTORS:</b>	<ul style="list-style-type: none"> <li>Guest Speaker</li> </ul>	
<b>OTHER INTERESTED STAKEHOLDERS:</b>	<ul style="list-style-type: none"> <li>None</li> </ul>	
<b>DESCRIPTION:</b>	<p>This use case describes the event of Inviting a Guest Speaker to a function. The use case begins when management creates a function in the UC 4.5 Create event. The guest speakers will be displayed in a list that will then invoke this use case to invite the guest to the function. The user will then pick a guest speaker from this list and their information will be displayed accordingly. The user will then indicate to 'Invite Guest Speaker' and a message will be produced with all the event information obtained from the UC .4.5. After the message is produced, the user can edit the message to their liking and send it to the guest speaker. The system will then open a pending ticket awaiting the guest speaker's response to the invitation. The use case ends when the guest speaker is invited and TRWLA awaits confirmation of attendance.</p>	
<b>PRE-CONDITION:</b>	The user must be logged onto the system.	
<b>TRIGGER:</b>	The user wants to invite a guest speaker to a function.	
<b>TYPICAL COURSE</b>	<b>Actor Action</b>	<b>System Response</b>
<b>OF EVENTS:</b>	<b>Step 1:</b> The user wants to invite a guest speaker to a function and proceeds to the main menu.	<b>Step 2:</b> The system displays a list of guest speakers as retrieved from the <u>GuestSpeaker</u> table and prompts the user to search for the guest speaker they wish to invite (Invoke UC 5.2).

	<b>Step 3:</b> The user searches and selects the guest speaker from the list.	<b>Step 4:</b> The system displays selected guest speaker's information in editable format with an option to select 'Invite Guest Speaker'.
	<b>Step 5:</b> The user selects the 'Invite Guest Speaker' option.	<b>Step 6:</b> The system obtains the event information that it has gathered from the <b>Function</b> table, to send to the guest speaker which includes the theme.
		<b>Step 7:</b> The system produces a message to send to the guest speaker which includes the event information created.
	<b>Step 8:</b> The user edits the message to their liking and selects the 'Send Message' option.	<b>Step 9:</b> The system sends the message to the guest speaker based on the guest speaker information obtained from the <b>GuestSpeaker</b> table.
		<b>Step 10:</b> An open option for an Accepted or Rejected response is created. The <b>FunctionSpeaker</b> table is updated and the UC 5.6 Confirm Guest Speaker is invoked.
<b>ALTERNATE COURSES:</b>	<b>Alt-Step 3:</b> If the guest speaker is not on the list, the user can register the guest speaker on the system by selecting 'Invite New Guest Speaker'. → Invoke UC 5.1	
	<b>Alt-Step 9:</b> The system fails to send the message because of a technical error. The system displays a notification informing the user that the message could not be sent.	
<b>CONCLUSION:</b>	An open option for an Accepted or Rejected response is created. The <b>FunctionSpeaker</b> table is updated and the UC 5.6 Confirm Guest Speaker is invoked.	
<b>POST-CONDITION:</b>	A guest speaker has been invited to attend a function.	
<b>BUSINESS RULES</b>	<ul style="list-style-type: none"> <li>Only volunteers may invite a guest speaker to attend a function.</li> </ul>	
<b>IMPLEMENTATION CONSTRAINTS AND SPECIFICATIONS</b>	<ul style="list-style-type: none"> <li>The information for the event has to be completed while creating the event and converted to an attachment.</li> </ul>	
<b>ASSUMPTIONS:</b>	None	

**OPEN ISSUES:**

None

<b>USE CASE NAME:</b>	Confirm Guest Speaker	<b>USE CASE TYPE</b>
<b>USE CASE ID:</b>	5.6	Business Requirements: <input type="checkbox"/>
<b>PRIORITY:</b>	High	System Analysis: <input checked="" type="checkbox"/>
<b>SOURCE:</b>	TuksRes Women In Leadership Academy	System Design: <input type="checkbox"/>
<b>PRIMARY BUSINESS ACTOR</b>	Guest Speaker	
<b>PRIMARY SYSTEM ACTOR</b>	Volunteers	
<b>OTHER PARTICIPATING ACTORS:</b>	<ul style="list-style-type: none"> <li>None</li> </ul>	
<b>OTHER INTERESTED STAKEHOLDERS:</b>	<ul style="list-style-type: none"> <li>None</li> </ul>	
<b>DESCRIPTION:</b>	<p>This use case describes the event of an invited guest speaker that confirms their attendance at the function. The use case begins when the guest speaker responds to the invitation confirming that they will attend the created function. The user will then update event details, Invoke UC 4.8 Update Event Information. Once the system has retrieved the guest speaker's information, it will display the information accordingly. The user will then indicate on the pending ticket that the guest speaker has 'Accepted' the invitation. Once the ticket has been updated, the system will close the ticket and update the event information in UC 4.8.</p>	
<b>PRE-CONDITION:</b>	The user must be logged onto the system.	
<b>TRIGGER:</b>	The guest speaker has sent a response to the invitation.	
<b>TYPICAL COURSE</b>	<b>Actor Action</b>	<b>System Response</b>
<b>OF EVENTS:</b>	<b>Step 1:</b> The guest speaker has sent a response to the invitation.	<b>Step 2:</b> The system sends a push notification that a response has been received from the guest speaker.
		<b>Step 3:</b> The user opens and reads the response received from the guest speaker.
		<b>Step 4:</b> The user updates the event information according to the guest speaker's response. (Invoke UC 4.9)

		<b>Step 5:</b> The system displays selected guest speaker's information in editable format as well as the pending ticket options, i.e.; 'Accepted' and 'Rejected'.
		<b>Step 6:</b> The user selects the 'Accepted' option by the pending ticket opened previously.
		<b>Step 7:</b> The system confirms the acceptance and closes the pending ticket.
		<b>Step 8:</b> The system updates the event information accordingly as well as the <b>FunctionSpeaker</b> table. Invoke UC 4.9 Update Event Information.
<b>ALTERNATE COURSES:</b>	<b>Alt-Step 6:</b> The user selects 'Rejected' if the guest speaker has rejected attendance for the function. The user will have to invite a different guest speaker to the function. → Invoke UC 5.5 Invite Guest Speaker.	
	<b>Alt-Step 7:</b> The system confirms guest rejection and closes the pending ticket. → Go to <b>Step 8</b> .	
<b>CONCLUSION:</b>	The user has received confirmation that a guest speaker will be speaking at the relevant function. The system updates the event information accordingly as well as the <b>FunctionSpeaker</b> table. Invoke UC 4.9 Update Event Information.	
<b>POST-CONDITION:</b>	The guest speaker has been confirmed for the function.	
<b>BUSINESS RULES</b>	<ul style="list-style-type: none"> <li>Only volunteers may update event information.</li> </ul>	
<b>IMPLEMENTATION CONSTRAINTS AND SPECIFICATIONS</b>	<ul style="list-style-type: none"> <li>None</li> </ul>	
<b>ASSUMPTIONS:</b>	<ul style="list-style-type: none"> <li>None</li> </ul>	
<b>OPEN ISSUES:</b>	None	

### 2.3.6 Community Engagement Subsystem Narratives

<b>USE CASE NAME:</b>	Upload Community Engagement Content	<b>USE CASE TYPE</b>
<b>USE CASE ID:</b>	6.1	Business Requirements: <input type="checkbox"/>
<b>PRIORITY:</b>	High	System Analysis: <input checked="" type="checkbox"/>
<b>SOURCE:</b>	TuksRes Women In Leadership Academy	System Design: <input type="checkbox"/>
<b>PRIMARY BUSINESS ACTOR</b>	Volunteers	
<b>PRIMARY SYSTEM ACTOR</b>	None	
<b>OTHER PARTICIPATING ACTORS:</b>	<ul style="list-style-type: none"> <li>User</li> </ul>	
<b>OTHER INTERESTED STAKEHOLDERS:</b>	<ul style="list-style-type: none"> <li>None</li> </ul>	
<b>DESCRIPTION:</b>	<p>This use case describes the event of uploading community engagement content on the system. The use case begins when a Volunteers wishes to upload new community engagement content so that students can view or download the content for their community engagement events. The user will first create an event whereby they would invoke UC 4.5 Create Event. Once the user has indicated to create a community engagement event, the system will display a list of community engagement events that are currently stored on the system. The user will then indicate to 'upload new community engagement content' whereby the system open up an option to locate the content on the user's computer. Once the user has located the content on their computer, the user will indicate to upload the content. Once the content has been uploaded the user will be required to enter specific information about the content. Once the user has indicated to save the content, the system will automatically lock the content and store it accordingly.</p>	
<b>PRE-CONDITION:</b>	The user must be logged onto the system.	
<b>TRIGGER:</b>	The user wants to upload community engagement content on the system.	
<b>TYPICAL COURSE</b>	<b>Actor Action</b>	<b>System Response</b>

OF EVENTS:	<b>Step 1:</b> The user wants to upload community engagement content on the system and proceeds to the main menu.	<b>Step 2:</b> The system displays a main menu with different options to choose from including 'Events'.
	<b>Step 3:</b> The user selects the 'Events' option.	<b>Step 4:</b> The system prompts the user with a list of events. Invoke UC 4.1 Create Event.
	<b>Step 5:</b> The user creates a Community Engagement event.	<b>Step 6:</b> The system displays a list of community engagement content from the <b>CommContent</b> table.
	<b>Step 7:</b> The user selects 'Upload Community Engagement Content'.	<b>Step 8:</b> The system prompts the user to upload the relevant content.
	<b>Step 9:</b> The user uploads the relevant content.	<b>Step 10:</b> The system prompts the user to name the content.
	<b>Step 11:</b> The user renames the content according to a theme.	<b>Step 12:</b> The system renames the content according to what the user stipulated.
		<b>Step 13:</b> The system automatically locks the content to other users.
		<b>Step 14:</b> The system prompts the user to enter information about the content.
	<b>Step 15:</b> The user enters all the relevant information about the community engagement content including: <ul style="list-style-type: none"> <li>• Theme</li> <li>• Short Summary</li> <li>• Description</li> <li>• Automatic link from upload</li> </ul> and selects 'Confirm Upload' option.	<b>Step 16:</b> The system validates that all the necessary information has been acquired, namely: <ul style="list-style-type: none"> <li>• Theme (Max 50 characters)</li> <li>• Short Summary (Max 100 characters)</li> <li>• Description (Max 255 characters)</li> <li>• Status</li> <li>• Automatic link from upload</li> </ul>
		<b>Step 17:</b> The system stores all the information in the <b>CommContent</b> and



		<b>Content</b> tables and displays a confirmation message notifying the user that the content has been successfully uploaded.
<b>ALTERNATE COURSES:</b>	<b>Alt-Step 12:</b> The content name already exists on the system, the system will warn the user to overwrite the existing file name or change the name. → Go to <b>Step 13</b> if changes the name. → Overwrite the existing file and go to <b>Step 15</b> .	
	<b>Alt-Step 13:</b> The user wishes to unlock the content and allow students to view the content, they will select 'Unlock Content'. → Invoke UC 6.4.	
	<b>Alt-Step 16:</b> The fields are incomplete or do not meet the requirements. → Go to <b>Step 15</b> .	
<b>CONCLUSION:</b>	The system stores all the information in the <b>CommContent</b> and <b>Content</b> tables and displays confirmation message notifying the user that the content has been successfully uploaded.	
<b>POST-CONDITION:</b>	Community engagement content has been uploaded to the system.	
<b>BUSINESS RULES</b>	<ul style="list-style-type: none"> <li>Only volunteers may upload community engagement content.</li> </ul>	
<b>IMPLEMENTATION CONSTRAINTS AND SPECIFICATIONS</b>	<ul style="list-style-type: none"> <li>Link must be accessible to users to download the content if indicated as unlocked.</li> </ul>	
<b>ASSUMPTIONS:</b>	<ul style="list-style-type: none"> <li>None</li> </ul>	
<b>OPEN ISSUES:</b>	None	

<b>USE CASE NAME:</b>	Search Community Engagement Content	<b>USE CASE TYPE</b>
<b>USE CASE ID:</b>	6.2	Business Requirements: <input type="checkbox"/>
<b>PRIORITY:</b>	High	System Analysis: <input checked="" type="checkbox"/>
<b>SOURCE:</b>	TuksRes Women In Leadership Academy	System Design: <input type="checkbox"/>
<b>PRIMARY BUSINESS ACTOR</b>	User	
<b>PRIMARY SYSTEM ACTOR</b>	None	
<b>OTHER PARTICIPATING ACTORS:</b>	<ul style="list-style-type: none"> <li>None</li> </ul>	
<b>OTHER INTERESTED STAKEHOLDERS:</b>	<ul style="list-style-type: none"> <li>None</li> </ul>	
<b>DESCRIPTION:</b>	<p>This use case describes the event of searching for community engagement content on the system. The use case begins when a user wishes to search for community engagement content to view or download, in order to use the content for a community engagement event. The user will first indicate to search for an event where they will search for 'Community Engagement Content'. The system will then display a list of current community engagement events that exists on the system. Once the user has indicated which event they would like to view, the system will display the relevant information pertaining to that event. The user will then select the content link and the system will display the information relating to that content.</p>	
<b>PRE-CONDITION:</b>	The user must be logged onto the system.	
<b>TRIGGER:</b>	The user wants to search for community engagement content on the system.	
<b>TYPICAL COURSE</b>	<b>Actor Action</b>	<b>System Response</b>
<b>OF EVENTS:</b>	<b>Step 1:</b> The user wants to search for community engagement content on the system and proceeds to the main menu.	<b>Step 2:</b> The system displays a main menu with different options to choose from.

	<b>Step 3:</b> The user selects the 'Events' option.	<b>Step 4:</b> The system prompts the user with a list of events with an option to 'Search Event'. Invoke UC 4.6 Search Event.
	<b>Step 5:</b> The user searches for the 'Community Engagement'.	<b>Step 6:</b> The system displays a list of existing community engagement events from the <b><u>CommunityOutreach</u></b> table.
	<b>Step 7:</b> The user selects the necessary community engagement event required from the list.	<b>Step 8:</b> The system displays the selected community engagement event information from the <b><u>CommunityOutreach</u></b> table.
	<b>Step 9:</b> The user selects the content link pertaining to the community engagement event.	<b>Step 10:</b> The system displays the content information such as: <ul style="list-style-type: none"> <li>• Name</li> <li>• Summary</li> <li>• Description</li> <li>• Theme</li> <li>• Status</li> <li>• Link</li> </ul> Obtained from the <b><u>CommContent</u></b> and <b><u>Content</u></b> tables.
<b>ALTERNATE COURSES:</b>	<b>Alt-Step 6a:</b> If the user is a volunteer and the content is not displayed on the list, the volunteer can upload the content. → Invoke UC 6.1 Upload Community Engagement Content.	
<b>CONCLUSION:</b>	The system displays the content information such as: <ul style="list-style-type: none"> <li>• Name</li> <li>• Summary</li> <li>• Description</li> <li>• Theme</li> <li>• Status</li> <li>• Link</li> </ul> Obtained from the <b><u>CommContent</u></b> and <b><u>Content</u></b> tables.	
<b>POST-CONDITION:</b>	The user has searched for community engagement content.	
<b>BUSINESS RULES</b>	<ul style="list-style-type: none"> <li>• All users may search for community engagement content.</li> <li>• Only Volunteers may view locked content.</li> <li>• Student users may only view unlocked content.</li> </ul>	

	<ul style="list-style-type: none"><li>Information will be displayed in editable format if the user is a Volunteer.</li></ul>
<b>IMPLEMENTATION CONSTRAINTS AND SPECIFICATIONS</b>	<ul style="list-style-type: none"><li>Link must be accessible to users to download the content if indicated as unlocked.</li></ul>
<b>ASSUMPTIONS:</b>	<ul style="list-style-type: none"><li>None</li></ul>
<b>OPEN ISSUES:</b>	None

<b>USE CASE NAME:</b>	Delete Community Engagement Content	<b>USE CASE TYPE</b>
<b>USE CASE ID:</b>	6.3	Business Requirements: <input type="checkbox"/>
<b>PRIORITY:</b>	Low	System Analysis: <input checked="" type="checkbox"/>
<b>SOURCE:</b>	TuksRes Women In Leadership Academy	System Design: <input type="checkbox"/>
<b>PRIMARY BUSINESS ACTOR</b>	Volunteers	
<b>PRIMARY SYSTEM ACTOR</b>	None	
<b>OTHER PARTICIPATING ACTORS:</b>	<ul style="list-style-type: none"> <li>None</li> </ul>	
<b>OTHER INTERESTED STAKEHOLDERS:</b>	<ul style="list-style-type: none"> <li>None</li> </ul>	
<b>DESCRIPTION:</b>	<p>This use case describes the event of deleting Community Engagement Content on the system. The use case begins when a user wishes to delete Community Engagement Content from the system that is no longer feasible for TRWLA to make use of. The user will first search for the community engagement content, Invoke UC 6.2 Search Community Engagement Content. The system will then display a list of current community engagement content that is on the system. Once the user has indicated which content they would like to view, the system will display the information in editable format based on their admin rights. Once the information is displayed the user will indicate to delete community engagement content. The system will first warn the user of permanent deletion. Once the user confirms the deletion, the community engagement content will be deleted from the system and the community engagement content list will be updated accordingly.</p>	
<b>PRE-CONDITION:</b>	The user must be logged onto the system.	
<b>TRIGGER:</b>	The user wants to delete Community Engagement Content from the system.	
<b>TYPICAL COURSE</b>	<b>Actor Action</b>	<b>System Response</b>
<b>OF EVENTS:</b>	<b>Step 1:</b> The user wants to delete community engagement content from the	<b>Step 2:</b> The system displays a main menu with different options to choose from.

	system and proceeds to the main menu.	
	<b>Step 3:</b> The user selects the 'Events' option.	<b>Step 4:</b> The system prompts the user with a list of events with an option to 'Search Event'. Invoke UC 6.2 Search Community Engagement Content.
	<b>Step 5:</b> The user searches and selects the necessary 'Community Engagement Content'.	<b>Step 6:</b> The system displays the selected community engagement content information in editable format with the option to 'Delete Content'.
	<b>Step 7:</b> The user selects the 'Delete Content' option.	<b>Step 8:</b> The system displays a message warning the user of permanent deletion and prompts the user to select either 'Confirm Deletion' or 'Cancel'.
	<b>Step 9:</b> The user confirms deletion of content.	<b>Step 10:</b> The system deletes the content from the <u>CommContent</u> and <u>Content</u> tables and updates the content list.
		<b>Step 11:</b> The system confirms with a notification the deletion of content.
<b>ALTERNATE COURSES:</b>	<b>Alt-Step 9:</b> If the user chooses not to permanently delete the content and selects the 'Cancel' option then the system will cancel deletion. → Go to <b>Step 4</b> .	
<b>CONCLUSION:</b>	The user has searched and deleted the relevant community engagement content. The system removes the content from the <u>CommContent</u> and <u>Content</u> tables and confirms deletion of content.	
<b>POST-CONDITION:</b>	Community engagement content has been deleted from the system.	
<b>BUSINESS RULES</b>	<ul style="list-style-type: none"> <li>Only volunteers may delete community engagement content.</li> </ul>	
<b>IMPLEMENTATION CONSTRAINTS AND SPECIFICATIONS</b>	<ul style="list-style-type: none"> <li>Hide all editable information to Student users and other users without the necessary admin rights.</li> </ul>	
<b>ASSUMPTIONS:</b>	<ul style="list-style-type: none"> <li>None</li> </ul>	
<b>OPEN ISSUES:</b>	None	

<b>USE CASE NAME:</b>	Lock Community Engagement Content	<b>USE CASE TYPE</b>
<b>USE CASE ID:</b>	6.4	Business Requirements: <input type="checkbox"/>
<b>PRIORITY:</b>	Medium	System Analysis: <input checked="" type="checkbox"/>
<b>SOURCE:</b>	TuksRes Women In Leadership Academy	System Design: <input type="checkbox"/>
<b>PRIMARY BUSINESS ACTOR</b>	Volunteers	
<b>PRIMARY SYSTEM ACTOR</b>	None	
<b>OTHER PARTICIPATING ACTORS:</b>	<ul style="list-style-type: none"> <li>None</li> </ul>	
<b>OTHER INTERESTED STAKEHOLDERS:</b>	<ul style="list-style-type: none"> <li>None</li> </ul>	
<b>DESCRIPTION:</b>	<p>This use case describes the event of locking community engagement content on the system. The use case begins when a user wishes to lock community engagement content on the system that the user does not want students or other users to view or download. The user will first search for the community engagement content they would like to lock. The system will display a list of community engagement content and the user will indicate which content they would like to lock. The system will then display the information pertaining to the content selected. The user will then indicate to 'Lock content'. The system will then lock the content which would no longer make it available to student users to download or view.</p>	
<b>PRE-CONDITION:</b>	The user must be logged onto the system.	
<b>TRIGGER:</b>	The user wants to lock Community Engagement Content on the system.	
<b>TYPICAL COURSE</b>	<b>Actor Action</b>	<b>System Response</b>
<b>OF EVENTS:</b>	<b>Step 1:</b> The user wants to lock Community Engagement Content on the system.	<b>Step 2:</b> The system displays a main menu with different options to choose from.
	<b>Step 3:</b> The user selects the 'Event' option.	<b>Step 4:</b> The system prompts the user with a list of events with an option to 'Search Event'. (Invoke UC 6.2 Search Community Engagement Content)

	<b>Step 5:</b> The user searches and selects the necessary 'Community Engagement Content'.	<b>Step 6:</b> The system displays the selected community engagement content information in editable format.
	<b>Step 7:</b> The user selects 'Lock Content' option.	<b>Step 8:</b> The system warns the user that the content will be locked and not accessible to all users.
	<b>Step 9:</b> The user confirms lock of content.	<b>Step 10:</b> The system locks the content and updates the content list under the Event section see step 6. The <b><u>CommContent</u></b> table is updated.
		<b>Step 11:</b> The system confirms lock of content.
<b>ALTERNATE COURSES:</b>	<b>Alt-Step 5:</b> If the content is not on the list; → Go to <b>Step 4</b> and create event where the user can upload community engagement content. See UC 6.1 Upload Community Engagement Content.	
	<b>Alt-Step 9:</b> If the user chooses not to lock the content then the system will cancel the lock. → Go to <b>Step 6</b> .	
<b>CONCLUSION:</b>	The system confirms lock of content.	
<b>POST-CONDITION:</b>	Community Engagement content has been locked on the system.	
<b>BUSINESS RULES</b>	<ul style="list-style-type: none"> <li>Only Volunteers may lock community engagement content.</li> <li>Students or unauthorized users cannot view locked content.</li> </ul>	
<b>IMPLEMENTATION CONSTRAINTS AND SPECIFICATIONS</b>	<ul style="list-style-type: none"> <li>None</li> </ul>	
<b>ASSUMPTIONS:</b>	<ul style="list-style-type: none"> <li>None</li> </ul>	
<b>OPEN ISSUES:</b>	None	



<b>USE CASE NAME:</b>	Unlock Community Engagement Content	<b>USE CASE TYPE</b>
<b>USE CASE ID:</b>	6.5	Business Requirements: <input type="checkbox"/>
<b>PRIORITY:</b>	Medium	System Analysis: <input checked="" type="checkbox"/>
<b>SOURCE:</b>	TuksRes Women In Leadership Academy	System Design: <input type="checkbox"/>
<b>PRIMARY BUSINESS ACTOR</b>	Volunteers	
<b>PRIMARY SYSTEM ACTOR</b>	None	
<b>OTHER PARTICIPATING ACTORS:</b>	<ul style="list-style-type: none"> <li>None</li> </ul>	
<b>OTHER INTERESTED STAKEHOLDERS:</b>	<ul style="list-style-type: none"> <li>None</li> </ul>	
<b>DESCRIPTION:</b>	<p>This use case describes the event of unlocking Community Engagement Content on the system. The use case begins when a user wishes to unlock Community Engagement Content on the system that students are now able to access to view or download. The user will first search for the community engagement content they would like to unlock. The system will display a list of community engagement content and the user will indicate which content they would like to unlock. The system will then display the information pertaining to the content selected. The user will then indicate to 'Unlock content'. The system will then unlock the content which will now make it available to student users to download or view.</p>	
<b>PRE-CONDITION:</b>	The user must be logged onto the system.	
<b>TRIGGER:</b>	The user wants to unlock Community Engagement Content on the system.	
<b>TYPICAL COURSE</b>	<b>Actor Action</b>	<b>System Response</b>
<b>OF EVENTS:</b>	<b>Step 1:</b> The user wants to unlock community engagement content on the system.	<b>Step 2:</b> The system displays a main menu with different options to choose from.

	<b>Step 3:</b> The user selects the 'Event' option.	<b>Step 4:</b> The system prompts the user with a list of events and an option to 'Search Events'. (Invoke UC 6.2)
	<b>Step 5:</b> The user searches and selects the necessary 'Community Engagement Content'.	<b>Step 6:</b> The system displays the selected community engagement content information in editable format.
	<b>Step 7:</b> The user selects 'Unlock Content' option.	<b>Step 8:</b> The system warns the user that the content will be unlocked and accessible to all users.
	<b>Step 9:</b> The user confirms unlock of content.	<b>Step 10:</b> The system unlocks the content which allows all users to access and view the content, and updates the content list. The <b><u>CommContent</u></b> table is updated.
		<b>Step 11:</b> The system confirms with a notification that the content has been unlocked.
<b>ALTERNATE COURSES:</b>	<b>Alt-Step 5:</b> The content is not on the list. → Invoke UC 6.1 Upload Community Engagement Content.	
	<b>Alt-Step 9:</b> If the user chooses not to unlock the content then the system will cancel the lock. → Go to <b>Step 6</b> .	
<b>CONCLUSION:</b>	The system confirms with a notification that the content has been unlocked.	
<b>POST-CONDITION:</b>	Community engagement content has been unlocked on the system.	
<b>BUSINESS RULES</b>	<ul style="list-style-type: none"> <li>Only Volunteers may unlock community engagement content.</li> <li>All users can view unlocked content.</li> </ul>	
<b>IMPLEMENTATION CONSTRAINTS AND SPECIFICATIONS</b>	<ul style="list-style-type: none"> <li>None</li> </ul>	
<b>ASSUMPTIONS:</b>	<ul style="list-style-type: none"> <li>None</li> </ul>	
<b>OPEN ISSUES:</b>	None	

### 2.3.7 Lecture Subsystem Narratives

<b>USE CASE NAME:</b>	Upload Lecture Content	<b>USE CASE TYPE</b>
<b>USE CASE ID:</b>	7.1	Business Requirements: <input type="checkbox"/>
<b>PRIORITY:</b>	High	System Analysis: <input checked="" type="checkbox"/>
<b>SOURCE:</b>	TuksRes Women in Leadership Academy	System Design: <input type="checkbox"/>
<b>PRIMARY BUSINESS ACTOR</b>	volunteer	
<b>PRIMARY SYSTEM ACTOR</b>	User	
<b>OTHER PARTICIPATING ACTORS:</b>	None	
<b>OTHER INTERESTED STAKEHOLDERS:</b>	<ul style="list-style-type: none"> <li>Students</li> </ul>	
<b>DESCRIPTION:</b>	<p>This use cases describes the event whereby the volunteer wishes to upload the necessary lecture content onto the system. It is initiated when the volunteer selects the option to upload lecture content onto the system. The content uploaded, will be uploaded per the theme TRWLA have scheduled to cover for that specific point in time. This use case concludes once the system displays another notification notifying the volunteer that their lecture content has been uploaded.</p>	
<b>PRE-CONDITION:</b>	The user needs to be logged onto the system, as well as the relevant event having been created.	
<b>TRIGGER:</b>	Volunteer chooses option to upload content.	
<b>TYPICAL COURSE OF EVENTS:</b>	<b>Actor Action</b>	<b>System Response</b>
	<b>Step 1:</b> The volunteer wants to upload lecture content and proceeds to the main menu.	<b>Step 2:</b> The system prompts the volunteer with a main menu, with an option 'Events'.
	<b>Step 3:</b> The volunteer then selects the 'Events' option.	<b>Step 4:</b> The system displays the events created. Which contain the event types: 'Community Engagement', 'Function' and 'Lecture'. Invoke UC 4.5 Create Event.
	<b>Step 5:</b> The volunteer selects a 'Lecture' event.	<b>Step 6:</b> The system displays the lecture event, with an option to upload lecture content from the <u>Lecture</u> table.
	<b>Step 7:</b> The volunteer then selects the option to upload the lecture content.	<b>Step 8:</b> System prompts the volunteer to upload the lecture content.
	<b>Step 9:</b> The volunteer uploads the lecture content.	<b>Step 10:</b> System displays notification that the lecture content is being uploaded onto the system.

		<b>Step 11:</b> The system prompts the volunteer to enter details about the content with the option to 'Upload'.
	<b>Step 12:</b> The volunteer then proceeds to enter the required details about the lecture content, including: <ul style="list-style-type: none"> <li>• Name</li> <li>• Date</li> <li>• Description</li> <li>• Theme</li> </ul> and then selects the 'Upload' option.	<b>Step 13:</b> The system validates that all the information in fields that are complete and valid, including: <ul style="list-style-type: none"> <li>• Name (Maximum 35 characters)</li> <li>• Date (10 characters in CCYY-MM-DD format)</li> <li>• Description (Maximum 140 characters)</li> <li>• Theme (Maximum 150 characters)</li> </ul>
		<b>Step 14:</b> The system then displays a message asking the volunteer if they are sure about the details entered.
	<b>Step 15:</b> The volunteer then proceeds to select the 'Okay' option to continue.	<b>Step 16:</b> The system inserts the lecture content details into the <u>LectureContent</u> and <u>Content</u> tables and notifies the volunteer that their lecture content has been uploaded.
<b>ALTERNATE COURSES:</b>	<b>Alt step 10:</b> If the file that is uploaded is not the correct file size, format or is corrupted, then the system will display a notification indicating that the file was unable to be uploaded onto the system. → Go to <b>Step 3</b> .	
	<b>Alt step 13a:</b> If the fields that are validated are incorrect, the volunteer will need re-entered the details: <ul style="list-style-type: none"> <li>• Name (Maximum 35 characters)</li> <li>• Date (10 characters in CCYY-MM-DD format)</li> <li>• Description (Maximum 140 characters)</li> </ul> → Go to <b>Step 6</b> .	
<b>CONCLUSION:</b>	This use case concludes once the system displays another notification notifying the volunteer that their lecture content has been uploaded.	
<b>POST-CONDITION:</b>	The lecture content has been uploaded to the system.	
<b>BUSINESS RULES</b>	<ul style="list-style-type: none"> <li>• The volunteer must have the appropriate access rights.</li> <li>• Uploaded content must be in line with the specific theme that is being explored in the lectures.</li> </ul>	
<b>IMPLEMENTATION CONSTRAINTS AND SPECIFICATIONS</b>	<ul style="list-style-type: none"> <li>• The system should indicate to the user the file size restrictions before they upload any content.</li> </ul>	

---

<b>ASSUMPTIONS:</b>	The volunteer is already logged into the system.
<b>OPEN ISSUES:</b>	None

<b>USE CASE NAME:</b>	Search Lecture Content	<b>USE CASE TYPE</b>
<b>USE CASE ID:</b>	7.2	Business Requirements: <input type="checkbox"/>
<b>PRIORITY:</b>	High	System Analysis: <input checked="" type="checkbox"/>
<b>SOURCE:</b>	TuksRes Women in Leadership Academy	System Design: <input type="checkbox"/>
<b>PRIMARY BUSINESS ACTOR</b>	User	
<b>PRIMARY SYSTEM ACTOR</b>	None	
<b>OTHER PARTICIPATING ACTORS:</b>	None	
<b>OTHER INTERESTED STAKEHOLDERS:</b>	<ul style="list-style-type: none"> <li>Students</li> </ul>	
<b>DESCRIPTION:</b>	<p>The use case describes the process of a user searching the system for lecture content. It is initiated when the volunteer selects option to upload lecture content onto the system. After the user, has found the lecture content, they should be able to view it or download it. The use case concludes once the system then displays the search results that the user requested pertaining to the lecture content.</p>	
<b>PRE-CONDITION:</b>	Volunteer has to have uploaded the lecture content, as well as, the user having to be logged into the system.	
<b>TRIGGER:</b>	The user selects the option to search for lecture content.	
<b>TYPICAL COURSE</b>	<b>Actor Action</b>	<b>System Response</b>
<b>OF EVENTS:</b>	<b>Step 1:</b> The use case is initiated when the volunteer wants to search for lecture content on the system and proceeds to the main menu.	<b>Step 2:</b> The system prompts the volunteer with a main menu, with an option 'Events'.
	<b>Step 3:</b> The user then selects the 'Events' option. The user then selects to search via the 'search' option.	<b>Step 4:</b> The system prompts the user with a list of events with an option to 'Search Event'. Invoke UC 4.6 Search Event.
	<b>Step 5:</b> The user searches for the lecture.	<b>Step 6:</b> The system displays a list of existing lecture events from the <b>Lecture</b> table.
	<b>Step 8:</b> The user then proceeds to enter the details of the lecture content.	<b>Step 7:</b> The system prompts the to enter in the search details of the lecture content, including:

		<ul style="list-style-type: none"> <li>• Name (Maximum 35 characters)</li> <li>• Date (10 characters in CCYY-MM-DD format)</li> <li>• Theme (Maximum 150 characters)</li> </ul>
		<b>Step 9:</b> The system then displays the search results that the user requested pertaining to the lecture content. The search results are retrieved from the <u>LectureContent</u> and <u>Content</u> table.
<b>ALTERNATE COURSES:</b>	<b>Alt Step 8a:</b> The system displays a message informing the user that the details that they have entered are incorrect. → Go to <b>Step 5</b> .	
<b>CONCLUSION:</b>	The use case concludes once the system then displays the search results that the user requested pertaining to the lecture content.	
<b>POST-CONDITION:</b>	The user has found the lecture content they were searching for.	
<b>BUSINESS RULES</b>	<ul style="list-style-type: none"> <li>• The content needs to be uploaded first before it can be searched.</li> </ul>	
<b>IMPLEMENTATION CONSTRAINTS AND SPECIFICATIONS</b>	<ul style="list-style-type: none"> <li>• The functionality of the queries to the table need to be optimized to bring efficient and accurate search results.</li> <li>• As well as, characters entered in the search bar to be highlighted to the user on the file name.</li> </ul>	
<b>ASSUMPTIONS:</b>	<ul style="list-style-type: none"> <li>• The user is logged into the system.</li> </ul>	
<b>OPEN ISSUES:</b>	None	

<b>USE CASE NAME:</b>	Delete Lecture Content	<b>USE CASE TYPE</b>
<b>USE CASE ID:</b>	7.3	Business Requirements: <input type="checkbox"/>
<b>PRIORITY:</b>	High	System Analysis: <input checked="" type="checkbox"/>
<b>SOURCE:</b>	TuksRes Women in Leadership Academy	System Design: <input type="checkbox"/>
<b>PRIMARY BUSINESS ACTOR</b>	volunteer	
<b>PRIMARY SYSTEM ACTOR</b>	None	
<b>OTHER PARTICIPATING ACTORS:</b>	User	
<b>OTHER INTERESTED STAKEHOLDERS:</b>	None	
<b>DESCRIPTION:</b>	This use case describes the event whereby the volunteer wishes to delete any lecture content off the system. It is initiated when the volunteer selects the lecture content they wish to delete. Once content is deleted off the system, it is not retrievable. This use case concludes once the system displays a notification confirming that the lecture content has been deleted.	
<b>PRE-CONDITION:</b>	The volunteer has to have uploaded the lecture content, as well as, the user having to be logged into the system.	
<b>TRIGGER:</b>	The volunteer selects the option to delete the lecture content.	
<b>TYPICAL COURSE OF EVENTS:</b>	<b>Actor Action</b>	<b>System Response</b>
	<b>Step 1:</b> The volunteer wishes to delete lecture content off the system and proceeds to the main menu.	<b>Step 2:</b> The system prompts the volunteer with a main menu, with an option 'Events'.
	<b>Step 3:</b> The volunteer then selects the 'Events' option. The volunteer then selects to search via the 'search' option.	<b>Step 4:</b> The system displays the volunteer with a list of events with an option to 'Search lecture Content'. Invoke UC 7.2 Search Lecture Content.
	<b>Step 5:</b> The volunteer then proceeds to search for 'Lecture Content'.	<b>Step 6:</b> The system retrieves a list of the lecture content from the <u>LectureContent</u> table.
	<b>Step 7:</b> The volunteer selects the necessary lecture content required from the list.	<b>Step 8:</b> The system displays the selected Lecture Content with an option to delete lecture content.
	<b>Step 9:</b> The volunteer then selects the option to 'Delete' the selected lecture content.	<b>Step 10:</b> The system then prompts the volunteer to verify their action by displaying a notification asking if they are sure they want to delete the



		selected lecture content, with the options 'Confirm Deletion' or 'Cancel'.
	<b>Step 11:</b> The volunteer then selects 'Confirm Deletion', confirming that they would want to delete the lecture content.	<b>Step 12:</b> The system displays a notification confirming that the lecture content has been deleted. The lecture content that was deleted is also removed from the <u>LectureContent</u> and <u>Content</u> table.
<b>ALTERNATE COURSES:</b>	<b>Alt Step 12:</b> the volunteer does not have the necessary access rights and is unable to delete the selected lecture content. → Go to <b>Step 10</b> .	
<b>CONCLUSION:</b>	This use case concludes once the system displays a notification confirming that the lecture content has been deleted.	
<b>POST-CONDITION:</b>	The selected lecture content has been deleted off the system.	
<b>BUSINESS RULES</b>	<ul style="list-style-type: none"> <li>Only the volunteer with the relevant access rights can delete the lecture content.</li> </ul>	
<b>IMPLEMENTATION CONSTRAINTS AND SPECIFICATIONS</b>	<ul style="list-style-type: none"> <li>The system should also warn the user that the file will be deleted and cannot be retrieved.</li> <li>It should also be easy for users to identify the delete option so that they are not able to delete the content by mistake.</li> </ul>	
<b>ASSUMPTIONS:</b>	<ul style="list-style-type: none"> <li>The volunteer should already be logged into the system.</li> </ul>	
<b>OPEN ISSUES:</b>	None	

<b>USE CASE NAME:</b>	Lock Lecture Content	<b>USE CASE TYPE</b>
<b>USE CASE ID:</b>	7.4	Business Requirements: <input type="checkbox"/>
<b>PRIORITY:</b>	Medium	System Analysis: <input checked="" type="checkbox"/>
<b>SOURCE:</b>	TuksRes Women in Leadership Academy	System Design: <input type="checkbox"/>
<b>PRIMARY BUSINESS ACTOR</b>	volunteer	
<b>PRIMARY SYSTEM ACTOR</b>	None	
<b>OTHER PARTICIPATING ACTORS:</b>	User	
<b>OTHER INTERESTED STAKEHOLDERS:</b>	<ul style="list-style-type: none"> <li>Students</li> </ul>	
<b>DESCRIPTION:</b>	<p>This use case describes the event whereby the volunteer does not want the students to view the lecture content before a specific time. So the lecture content will be locked for a specific time until the volunteer sees fit. The use case concludes once the system then displays a confirmation message confirming that the lecture content has been locked.</p>	
<b>PRE-CONDITION:</b>	Volunteer has to have uploaded the lecture content, as well as, the user having to be logged into the system.	
<b>TRIGGER:</b>	The volunteer wishes to lock lecture content.	
<b>TYPICAL COURSE OF EVENTS:</b>	<b>Actor Action</b>	<b>System Response</b>
	<b>Step 1:</b> The volunteer wishes to lock lecture content on the system and proceeds to the main menu.	<b>Step 2:</b> The system prompts the volunteer with a main menu, with an option 'Events'.
	<b>Step 3:</b> The volunteer then selects the 'Events' option. The volunteer then selects to search via the 'search' option.	<b>Step 4:</b> The system displays the volunteer with a list of events with an option to 'Search lecture Content'. Invoke UC 7.2 Search Lecture Content.
	<b>Step 5:</b> The volunteer then proceeds to search for 'Lecture Content'.	<b>Step 6:</b> The system retrieves a list of the lecture content from the <u>LectureContent</u> table.
	<b>Step 7:</b> The volunteer selects the necessary lecture content required from the list.	<b>Step 8:</b> The system displays the selected Lecture Content with an option to 'lock' lecture content.
	<b>Step 9:</b> The volunteer then proceeds to select the option to lock the lecture content.	<b>Step 10:</b> The system prompts the user to confirm if they are sure they want to lock the lecture content with the options 'Lock Lecture' or 'Cancel'.
	<b>Step 11:</b> The volunteer then	<b>Step 12:</b> The system then displays a

	proceeds to select 'Lock Lecture'.	confirmation message confirming that the lecture content has been locked.
<b>ALTERNATE COURSES:</b>	<b>Alt Step 12:</b> The volunteer does not have the necessary access rights and is unable to lock the selected lecture content. → Go to <b>Step 7</b> .	
<b>CONCLUSION:</b>	The use case concludes once the system then displays a confirmation message confirming that the lecture content has been locked.	
<b>POST-CONDITION:</b>	The lecture content should be hidden from the student users.	
<b>BUSINESS RULES</b>	<ul style="list-style-type: none"> <li>Only volunteers can lock lecture content.</li> </ul>	
<b>IMPLEMENTATION CONSTRAINTS AND SPECIFICATIONS</b>	<ul style="list-style-type: none"> <li>The file should be not downloadable and illustrative of the fact that it is not available for download to the user.</li> </ul>	
<b>ASSUMPTIONS:</b>	<ul style="list-style-type: none"> <li>The volunteer should already be logged into the system.</li> </ul>	
<b>OPEN ISSUES:</b>	None	

<b>USE CASE NAME:</b>	Unlock Lecture Content	<b>USE CASE TYPE</b>
<b>USE CASE ID:</b>	7.5	Business Requirements: <input type="checkbox"/>
<b>PRIORITY:</b>	Medium	System Analysis: <input checked="" type="checkbox"/>
<b>SOURCE:</b>	TuksRes Women in Leadership Academy	System Design: <input type="checkbox"/>
<b>PRIMARY BUSINESS ACTOR</b>	Volunteer	
<b>PRIMARY SYSTEM ACTOR</b>	None	
<b>OTHER PARTICIPATING ACTORS:</b>	User	
<b>OTHER INTERESTED STAKEHOLDERS:</b>	<ul style="list-style-type: none"> <li>Students</li> </ul>	
<b>DESCRIPTION:</b>	<p>The use case describes the event whereby the volunteer has decided to release the lecture content for the students to view. So they unlock on the system and the lecture content is now viewable and downloadable. The use case concludes once the system then displays a confirmation message confirming that the lecture content has been unlocked.</p>	
<b>PRE-CONDITION:</b>	Volunteer has to have uploaded the lecture content, as well as, the user having to be logged into the system.	
<b>TRIGGER:</b>	The volunteer selects the option to unlock lecture content.	
<b>TYPICAL COURSE OF EVENTS:</b>	<b>Actor Action</b>	<b>System Response</b>
	<b>Step 1:</b> The volunteer wishes to unlock lecture content on the system and proceeds to the main menu.	<b>Step 2:</b> The system prompts the volunteer with a main menu, with an option 'Events'.
	<b>Step 3:</b> The volunteer then selects the 'Events' option. The volunteer then selects to search via the 'search' option.	<b>Step 4:</b> The system prompts the volunteer with a list of events with an option to 'Search lecture Content'. Invoke UC 7.2 Search Lecture Content.
	<b>Step 5:</b> The volunteer then proceeds to search for 'Lecture Content'.	<b>Step 6:</b> The system retrieves a list of the lecture content from the <u>LectureContent</u> table.
	<b>Step 7:</b> The volunteer selects the necessary lecture content required from the list.	<b>Step 8:</b> The system displays the selected Lecture Content with an option to 'unlock' lecture content.
	<b>Step 9:</b> The volunteer then proceeds to select the option to unlock the lecture content.	<b>Step 10:</b> The system prompts the user to confirm if they are sure they want to unlock the lecture content with the options 'Lock Lecture' or 'Cancel'.
	<b>Step 11:</b> The volunteer then	<b>Step 12:</b> The system then displays a

	proceeds to select 'Unlock Lecture'.	confirmation message confirming that the lecture content has been Unlocked.
<b>ALTERNATE COURSES:</b>	<b>Alt Step 12:</b> The volunteer does not have the necessary access rights and is unable to unlock the selected lecture content. → Go to <b>Step 7</b> .	
<b>CONCLUSION:</b>	The use case concludes once the system then displays a confirmation message confirming that the lecture content has been unlocked.	
<b>POST-CONDITION:</b>	The specified lecture content should be unlocked and visible to the students.	
<b>BUSINESS RULES</b>	<ul style="list-style-type: none"> <li>Only volunteers can unlock lecture content.</li> </ul>	
<b>IMPLEMENTATION CONSTRAINTS AND SPECIFICATIONS</b>	<ul style="list-style-type: none"> <li>The file should be downloadable and illustrative of the fact that it is available for download to the user.</li> </ul>	
<b>ASSUMPTIONS:</b>	<ul style="list-style-type: none"> <li>The volunteer should already be logged into the system.</li> </ul>	
<b>OPEN ISSUES:</b>	None	

<b>USE CASE NAME:</b>	Content Review	<b>USE CASE TYPE</b>
<b>USE CASE ID:</b>	7.6	Business Requirements: <input type="checkbox"/>
<b>PRIORITY:</b>	Medium	System Analysis: <input checked="" type="checkbox"/>
<b>SOURCE:</b>	TuksRes Women in Leadership Academy	System Design: <input type="checkbox"/>
<b>PRIMARY BUSINESS ACTOR</b>	Student	
<b>PRIMARY SYSTEM ACTOR</b>	None	
<b>OTHER PARTICIPATING ACTORS:</b>	volunteer	
<b>OTHER INTERESTED STAKEHOLDERS:</b>	None	
<b>DESCRIPTION:</b>	This use case describes the event when the student seeks to write a review and rating on the lecture content. After writing it, it will be posted under the lecture content for other students and members to view. The user will give a rating out of five stars for any content. As well as, the creator of the content being notified of reviews left. This use case concludes once the system then sends a notification to the author of the content telling them that a review has been left on their lecture content.	
<b>PRE-CONDITION:</b>	Content is uploaded onto the system.	
<b>TRIGGER:</b>	Student selects the option to write a content review.	
<b>TYPICAL COURSE OF EVENTS:</b>	<b>Actor Action</b>	<b>System Response</b>
	<b>Step 1:</b> The student wants to write a content review and proceeds to the main menu.	<b>Step 2:</b> The system prompts the Student with a main menu, with an option 'Events'.
	<b>Step 3:</b> The volunteer then selects the 'Events' option. The volunteer then selects to search via the 'search' option.	<b>Step 4:</b> The system prompts the volunteer with a list of events with an option to 'Search lecture Content'. Invoke UC 7.2 Search Lecture Content.
	<b>Step 5:</b> The volunteer then proceeds to search for 'Lecture Content'.	<b>Step 6:</b> The system retrieves a list of the lecture content from the <u>LectureContent</u> table.
	<b>Step 7:</b> The student selects the necessary lecture event required from the list.	<b>Step 8:</b> The system then displays lecture content that the student requested with the option to leave a 'Content Review'.
	<b>Step 9:</b> The student proceeds to select content review on the selected lecture content and the field and assign a	<b>Step 10:</b> The system then verifies that the review and rating has adhered to the restrictions: <ul style="list-style-type: none"> <li>• Description (Maximum 140</li> </ul>

	rating to the lecture content.	Characters) <ul style="list-style-type: none"> <li>Rating (5 stars)</li> </ul> and sends the input entered into the <b>Review</b> table
		<b>Step 11:</b> The system displays a message confirming that the student has successfully left a review.
<b>Alternative steps</b>	<b>Alt Step 12a:</b> The system displays a message telling the student that their post was unsuccessful due to their character limit. → Go to <b>Step 11</b> .	
<b>CONCLUSION:</b>	This use case concludes once the system then sends a notification to the author of the content telling them that a review has been left on their lecture content.	
<b>POST-CONDITION:</b>	Content review on a specific piece of lecture content has been published.	
<b>BUSINESS RULES</b>	<ul style="list-style-type: none"> <li>Any user can leave a content review.</li> </ul>	
<b>IMPLEMENTATION CONSTRAINTS AND SPECIFICATIONS</b>	<ul style="list-style-type: none"> <li>The user should be restricted to a predetermined amount of characters to complete their review.</li> <li>Reviews should also have the ability to be flagged for unethical or derogatory comments that do not represent that values of TRWLA.</li> <li>There needs to be a clear indication of the number of ratings a piece of content has received.</li> <li>The users who upload content also need to receive notifications of their content being reviewed.</li> <li>Icons would need to be utilized that are familiar to users that involve the rating of content on other systems.</li> </ul>	
<b>ASSUMPTIONS:</b>	<ul style="list-style-type: none"> <li>The volunteer should already be logged into the system.</li> </ul>	
<b>OPEN ISSUES:</b>	None	

### 2.3.8 Marketing Subsystem Narratives

<b>USE CASE NAME:</b>	Upload Photo	<b>USE CASE TYPE</b>
<b>USE CASE ID:</b>	8.1	Business Requirements: <input type="checkbox"/>
<b>PRIORITY:</b>	High	System Analysis: <input checked="" type="checkbox"/>
<b>SOURCE:</b>	TuksRes Women In Leadership Academy	System Design: <input type="checkbox"/>
<b>PRIMARY BUSINESS ACTOR</b>	User	
<b>PRIMARY SYSTEM</b>	None	

<b>ACTOR</b>		
<b>OTHER PARTICIPATING ACTORS:</b>	<ul style="list-style-type: none"> <li>• Server Provider</li> </ul>	
<b>OTHER INTERESTED STAKEHOLDERS:</b>	<ul style="list-style-type: none"> <li>• Management</li> </ul>	
<b>DESCRIPTION:</b>	<p>This use case is initiated when the user chooses to upload a photo onto the system. The photo would be from an event that TRWLA hosted that a user would like to add to the TRWLA gallery that will visible their website. Once the photo has been uploaded, admin will be notified that a photo is ready for review.</p>	
<b>PRE-CONDITION:</b>	The user must be logged on to the system.	
<b>TRIGGER:</b>	User wants to upload a photo to the system.	
<b>TYPICAL COURSE OF EVENTS:</b>	<b>Actor Action</b>	<b>System Response</b>
	<b>Step 1:</b> A user wants to upload a photo and proceeds to the main menu.	<b>Step 2:</b> The system prompts a user with a main menu, with a 'Gallery' option.
	<b>Step 3:</b> the user selects the 'Gallery' option.	<b>Step 4:</b> The system prompts the user with the 'Gallery', with the option to upload a photo.
	<b>Step 5:</b> The user selects the 'Upload a Photo' option.	<b>Step 6:</b> The system prompts the user with the 'Upload a Photo' section with a field to choose a photo and an 'Upload' option.
	<b>Step 7:</b> The user chooses the photo in the field to choose a photo.	<b>Step 8:</b> The system verifies that a valid photo has been chosen.
	<b>Step 9:</b> The user chooses the 'Upload' option.	<b>Step 10:</b> The system confirms the photo that was chosen.
	<b>Step 11:</b> The user confirms the photo that was chosen.	<b>Step 12:</b> The system displays a confirmation that the photo was successfully uploaded.
		<b>Step 13:</b> The system sends a notification message to admin that a photo has been uploaded and is ready for review and posting.
<b>ALTERNATE COURSES:</b>	<b>Alt-Step 7:</b> If the user cannot find the photo, then they will choose to cancel. → Go to <b>Step 5</b>	
	<b>Alt-Step 8:</b> If the photo that is uploaded is not the correct specification, then the system will notify the user that the photo was not able to be uploaded. → Go to <b>Step 7</b>	



	<b>Alt-Step 11:</b> The user does not confirm the selected photo. → Go to <b>Step 7</b>
<b>CONCLUSION:</b>	The system notifies admin that a photo has been uploaded.
<b>POST-CONDITION:</b>	A photo is uploaded.
<b>BUSINESS RULES</b>	<ul style="list-style-type: none"><li>Any user can upload photos</li></ul>
<b>IMPLEMENTATION CONSTRAINTS AND SPECIFICATIONS</b>	<ul style="list-style-type: none"><li>The system should allow any type of photo to be uploaded.</li></ul>
<b>ASSUMPTIONS:</b>	<ul style="list-style-type: none"><li>None</li></ul>
<b>OPEN ISSUES:</b>	None

<b>USE CASE NAME:</b>	Post Photo	<b>USE CASE TYPE</b>
<b>USE CASE ID:</b>	8.2	Business Requirements: <input type="checkbox"/>
<b>PRIORITY:</b>	Medium	System Analysis: <input checked="" type="checkbox"/>
<b>SOURCE:</b>	TuksRes Women In Leadership Academy	System Design: <input type="checkbox"/>
<b>PRIMARY BUSINESS ACTOR</b>	Admin	
<b>PRIMARY SYSTEM ACTOR</b>	None	
<b>OTHER PARTICIPATING ACTORS:</b>	<ul style="list-style-type: none"> <li>• Server Provider</li> </ul>	
<b>OTHER INTERESTED STAKEHOLDERS:</b>	<ul style="list-style-type: none"> <li>• Management</li> </ul>	
<b>DESCRIPTION:</b>	This use case describes the event of the admin of TRWLA reviewing a photo and posting the photo to the website's gallery. Admin will review the photo to check if it is appropriate for posting, accepts the photo for posting and posts the photo. The use case concludes by posting a photo.	
<b>PRE-CONDITION:</b>	Admin must be logged into the system.	
<b>TRIGGER:</b>	A photo is uploaded to the system for reviewing and posting.	
<b>TYPICAL COURSE</b>	<b>Actor Action</b>	<b>System Response</b>
<b>OF EVENTS:</b>	<b>Step 1:</b> A photo needs to be reviewed and posted.	<b>Step 2:</b> The system prompts admin with a main menu, with a 'Gallery' option.
	<b>Step 3:</b> Admin selects the 'Gallery' option.	<b>Step 4:</b> The system prompts admin with the 'Gallery', with the option to 'Post a Photo'.
	<b>Step 5:</b> Admin selects the 'Post a Photo' option.	<b>Step 6:</b> The system prompts admin with the 'Post a Photo' section.
	<b>Step 7:</b> Admin reviews and selects the photo that they accept for posting.	<b>Step 8:</b> The system confirms the photo that is selected.
	<b>Step 9:</b> Admin confirms the selected photo.	<b>Step 10:</b> The system posts the photo to the 'Gallery'.
		<b>Step 11:</b> The system prompts admin with a confirmation that the photo is posted to the 'Gallery'.
<b>ALTERNATE COURSES:</b>	<b>Alt-Step 6:</b> If there are no uploaded photos, the system will display the 'Post a Photo' section informing the user that there are no new uploaded photos.	
	<b>Alt-Step 7:</b> If the user rejects a photo will be removed from the 'Post a Photo' section.	
	<b>Alt-Step 9:</b> Admin does not confirm the selection.	

	→ Go to <b>Step 8</b> .
<b>CONCLUSION:</b>	The system confirms that the photo is posted to the 'Gallery'.
<b>POST-CONDITION:</b>	A photo is posted to the 'Gallery'.
<b>BUSINESS RULES</b>	<ul style="list-style-type: none"> <li>Only an authorized volunteer will have access to this function.</li> </ul>
<b>IMPLEMENTATION CONSTRAINTS AND SPECIFICATIONS</b>	<ul style="list-style-type: none"> <li>None</li> </ul>
<b>ASSUMPTIONS:</b>	<ul style="list-style-type: none"> <li>Admin will review photos using appropriate standards.</li> </ul>
<b>OPEN ISSUES:</b>	None

<b>USE CASE NAME:</b>	Delete Photo	<b>USE CASE TYPE</b>
<b>USE CASE ID:</b>	8.3	Business Requirements: <input type="checkbox"/>
<b>PRIORITY:</b>	Low	System Analysis: <input checked="" type="checkbox"/>
<b>SOURCE:</b>	TuksRes Women In Leadership Academy	System Design: <input type="checkbox"/>
<b>PRIMARY BUSINESS ACTOR</b>	Admin	
<b>PRIMARY SYSTEM ACTOR</b>	None	
<b>OTHER PARTICIPATING ACTORS:</b>	<ul style="list-style-type: none"> <li>• Server Provider</li> </ul>	
<b>OTHER INTERESTED STAKEHOLDERS:</b>	<ul style="list-style-type: none"> <li>• Management</li> </ul>	
<b>DESCRIPTION:</b>	This use case describes the event of admin wanting to delete a photo from the website's 'Gallery'. Admin will access the 'Gallery', find the photo and delete the photo. This use case concludes by a photo being deleted from the 'Gallery'.	
<b>PRE-CONDITION:</b>	Admin is logged into the system.	
<b>TRIGGER:</b>	Admin wants to delete a photo from the system.	
<b>TYPICAL COURSE</b>	<b>Actor Action</b>	<b>System Response</b>
<b>OF EVENTS:</b>	<b>Step 1:</b> Admin wants to delete a photo from the 'Gallery'.	<b>Step 2:</b> The system prompts admin with a main menu, with a 'Gallery' option.
	<b>Step 3:</b> Admin selects the 'Gallery' option.	<b>Step 4:</b> The system prompts admin with the 'Gallery', with the option to 'Delete a Photo'.
	<b>Step 5:</b> Admin selects a photo to delete.	
	<b>Step 6:</b> Admin deletes the photo by selecting the 'Delete a Photo' option.	<b>Step 7:</b> The system prompts admin to confirm the deletion.
	<b>Step 8:</b> Admin confirms the deletion.	<b>Step 9:</b> The system deletes the photo from 'Gallery'.
		<b>Step 10:</b> The system prompts admin with confirmation that the photo is deleted.
<b>ALTERNATE COURSES:</b>	<b>Alt-Step 6:</b> Admin does not choose to delete the photo. → Go to <b>Step 4</b> .	
	<b>Alt-Step 8:</b> Admin does not confirm deletion. → Go to <b>Step 5</b> .	
<b>CONCLUSION:</b>	The system prompts admin with a confirmation of deletion.	
<b>POST-CONDITION:</b>	A photo is deleted from the 'Gallery'.	

<b>BUSINESS RULES</b>	<ul style="list-style-type: none"><li>• Only an authorized volunteer will have access to this function.</li></ul>
<b>IMPLEMENTATION CONSTRAINTS AND SPECIFICATIONS</b>	<ul style="list-style-type: none"><li>• None</li></ul>
<b>ASSUMPTIONS:</b>	<ul style="list-style-type: none"><li>• All photos are deleted with appropriate reasoning.</li></ul>
<b>OPEN ISSUES:</b>	None

## 2.3.9 Report Subsystem Narratives

<b>USE CASE NAME:</b>	Generate Class Attendance Report	<b>USE CASE TYPE</b>
<b>USE CASE ID:</b>	9.1	Business Requirements: <input type="checkbox"/>
<b>PRIORITY:</b>	High	System Analysis: <input checked="" type="checkbox"/>
<b>SOURCE:</b>	TuksRes Women In Leadership Academy	System Design: <input type="checkbox"/>
<b>PRIMARY BUSINESS ACTOR</b>	Time	
<b>PRIMARY SYSTEM ACTOR</b>	None	
<b>OTHER PARTICIPATING ACTORS:</b>	<ul style="list-style-type: none"> <li>• None</li> </ul>	
<b>OTHER INTERESTED STAKEHOLDERS:</b>	<ul style="list-style-type: none"> <li>• Directors</li> <li>• Management</li> </ul>	
<b>DESCRIPTION:</b>	This use case describes the event of a class attendance report being generated at the end of each year. Information will be retrieved from the Attendance table, Event table and Lecture table. The use case concludes by a report being generated with relevant class attendance information.	
<b>PRE-CONDITION:</b>	Lectures must have taken place during the year.	
<b>TRIGGER:</b>	Time detects it is the end of the TRWLA year.	
<b>TYPICAL COURSE OF EVENTS:</b>	<b>Actor Action</b>	<b>System Response</b>
	<b>Step 1:</b> Time detects it is the end of the TRWLA year.	<b>Step 2:</b> The system accesses the <u>Attendance</u> table, <u>Event</u> table and <u>Lecture</u> table. Invoke use case 4.11
		<b>Step 3:</b> The system retrieves the following information from the <u>Attendance</u> table: <ul style="list-style-type: none"> <li>• AttendanceID</li> </ul> The following information from the <u>Event</u> table: <ul style="list-style-type: none"> <li>• EventDate</li> </ul> The following information from the <u>Lecture</u> table: <ul style="list-style-type: none"> <li>• LectureID</li> <li>• LectureName</li> </ul>
		<b>Step 4:</b> The system uses the information to calculate the total attendance per lecture for each lecture throughout the TRWLA year.
		<b>Step 5:</b> The system creates a report displaying the calculation from the

		previous step with the corresponding information: <ul style="list-style-type: none"> <li>• LectureName from the <b><u>Lecture</u></b> table</li> <li>• EventDate from the <b><u>Event</u></b> table</li> </ul>
		<b>Step 7:</b> The system notifies management of the report generation.
<b>ALTERNATE COURSES:</b>	None	
<b>CONCLUSION:</b>	A notification is sent to management	
<b>POST-CONDITION:</b>	A report is generated.	
<b>BUSINESS RULES</b>	<ul style="list-style-type: none"> <li>• Only authorized management will have access to the report</li> </ul>	
<b>IMPLEMENTATION CONSTRAINTS AND SPECIFICATIONS</b>	<ul style="list-style-type: none"> <li>• The report must be generated during off-peak hours of the system.</li> </ul>	
<b>ASSUMPTIONS:</b>	<ul style="list-style-type: none"> <li>• None</li> </ul>	
<b>OPEN ISSUES:</b>	None	

<b>USE CASE NAME:</b>	Generate Function Attendance Report	<b>USE CASE TYPE</b>
<b>USE CASE ID:</b>	9.2	Business Requirements: <input type="checkbox"/>
<b>PRIORITY:</b>	High	System Analysis: <input checked="" type="checkbox"/>
<b>SOURCE:</b>	TuksRes Women In Leadership Academy	System Design: <input type="checkbox"/>
<b>PRIMARY BUSINESS ACTOR</b>	Time	
<b>PRIMARY SYSTEM ACTOR</b>	None	
<b>OTHER PARTICIPATING ACTORS:</b>	<ul style="list-style-type: none"> <li>None</li> </ul>	
<b>OTHER INTERESTED STAKEHOLDERS:</b>	<ul style="list-style-type: none"> <li>Directors</li> <li>Management</li> </ul>	
<b>DESCRIPTION:</b>	<p>This use case describes the event of a function attendance report being generated at the end of each year. The function attendance information will be retrieved from the Attendance table, Event table, Function table and Guest Speaker table. The use case concludes by a report being generated comparing relevant function attendance information.</p>	
<b>PRE-CONDITION:</b>	A function must have taken place during the year.	
<b>TRIGGER:</b>	Time detects it is the end of the TRWLA year.	
<b>TYPICAL COURSE</b>	<b>Actor Action</b>	<b>System Response</b>
<b>OF EVENTS:</b>	<b>Step 1:</b> Time detects it is the end of the TRWLA year.	<b>Step 2:</b> The system accesses the <u>Attendance</u> table, <u>Event</u> table and <u>Function</u> table, <u>GuestSpeaker</u> table.
		<b>Step 3:</b> The system retrieves the following information from the <u>Attendance</u> table: <ul style="list-style-type: none"> <li>AttendanceID</li> </ul> The following information from the <u>Event</u> table: <ul style="list-style-type: none"> <li>EventDate</li> </ul> The following information from the <u>Function</u> table: <ul style="list-style-type: none"> <li>FunctionID</li> <li>FunctionName</li> </ul> The following information from the <u>GuestSpeaker</u> table: <ul style="list-style-type: none"> <li>GuestSpeakerName</li> </ul>
		<b>Step 4:</b> The system uses the information to calculate the total attendance per function for each function throughout the TRWLA year.



		<b>Step 5:</b> The system creates a report displaying the calculation from the previous step with the corresponding information: <ul style="list-style-type: none"> <li>• FunctionName (from the <b><u>Function</u></b> table)</li> <li>• GuestSpeakerName (from the <b><u>GuestSpeaker</u></b> table)</li> <li>• EventDate (from the <b><u>Event</u></b> table)</li> </ul>
		<b>Step 7:</b> The system notifies management of the report generation.
<b>ALTERNATE COURSES:</b>	None	
<b>CONCLUSION:</b>	A notification is sent to management	
<b>POST-CONDITION:</b>	A report is generated.	
<b>BUSINESS RULES</b>	<ul style="list-style-type: none"> <li>• Only authorized management will have access to the report</li> </ul>	
<b>IMPLEMENTATION CONSTRAINTS AND SPECIFICATIONS</b>	<ul style="list-style-type: none"> <li>• The report must be generated during off-peak hours of the system.</li> </ul>	
<b>ASSUMPTIONS:</b>	<ul style="list-style-type: none"> <li>• None</li> </ul>	
<b>OPEN ISSUES:</b>	None	

<b>USE CASE NAME:</b>	Generate Community Engagement Attendance Report	<b>USE CASE TYPE</b>
<b>USE CASE ID:</b>	9.3	Business Requirements: <input type="checkbox"/>
<b>PRIORITY:</b>	High	System Analysis: <input checked="" type="checkbox"/>
<b>SOURCE:</b>	TuksRes Women In Leadership Academy	System Design: <input type="checkbox"/>
<b>PRIMARY BUSINESS ACTOR</b>	Time	
<b>PRIMARY SYSTEM ACTOR</b>	None	
<b>OTHER PARTICIPATING ACTORS:</b>	<ul style="list-style-type: none"> <li>None</li> </ul>	
<b>OTHER INTERESTED STAKEHOLDERS:</b>	<ul style="list-style-type: none"> <li>Directors</li> <li>Management</li> </ul>	
<b>DESCRIPTION:</b>	<p>This use case describes the event of a community engagement attendance report being generated at the end of each year. The community engagement attendance information will be retrieved from the Attendance table, Community Engagement table and Event table. This report will show information about community engagement throughout the year.</p>	
<b>PRE-CONDITION:</b>	Students must have attended community engagement events throughout the year.	
<b>TRIGGER:</b>	Time detects it is the end of the TRWLA year.	
<b>TYPICAL COURSE OF EVENTS:</b>	<b>Actor Action</b>	<b>System Response</b>
	<b>Step 1:</b> Time detects it is the end of the TRWLA year.	<b>Step 2:</b> The system accesses the <u>Attendance</u> table, <u>Event</u> table and <u>Lecture</u> table.
		<b>Step 3:</b> The system retrieves the following information from the <u>Attendance</u> table: <ul style="list-style-type: none"> <li>AttendanceID</li> </ul> The following information from the <u>Event</u> table: <ul style="list-style-type: none"> <li>EventDate</li> </ul> The following information from the <u>CommunityOutreach</u> table: <ul style="list-style-type: none"> <li>CommID</li> <li>CommunityOutreachName</li> </ul>
		<b>Step 4:</b> The system uses the information to calculate the total attendance per community engagement event for each community

		engagement event throughout the TRWLA year.
		<b>Step 5:</b> The system creates a report displaying the calculation from the previous step with the corresponding information: <ul style="list-style-type: none"> <li>CommunityOutreachName (from the <b><u>CommunityOutreach</u></b> table)</li> <li>Event Date (from the <b><u>Event</u></b> table)</li> </ul>
		<b>Step 7:</b> The system notifies management of the report generation.
<b>ALTERNATE COURSES:</b>	None	
<b>CONCLUSION:</b>	A notification is sent to management	
<b>POST-CONDITION:</b>	A report is generated.	
<b>BUSINESS RULES</b>	<ul style="list-style-type: none"> <li>Only authorized management will have access to the report</li> </ul>	
<b>IMPLEMENTATION CONSTRAINTS AND SPECIFICATIONS</b>	<ul style="list-style-type: none"> <li>The report must be generated during off-peak hours of the system.</li> </ul>	
<b>ASSUMPTIONS:</b>	<ul style="list-style-type: none"> <li>None</li> </ul>	
<b>OPEN ISSUES:</b>	None	

<b>USE CASE NAME:</b>	Generate Demographics Report	<b>USE CASE TYPE</b>
<b>USE CASE ID:</b>	9.4	Business Requirements: <input type="checkbox"/>
<b>PRIORITY:</b>	High	System Analysis: <input checked="" type="checkbox"/>
<b>SOURCE:</b>	TuksRes Women In Leadership Academy	System Design: <input type="checkbox"/>
<b>PRIMARY BUSINESS ACTOR</b>	Time	
<b>PRIMARY SYSTEM ACTOR</b>	None	
<b>OTHER PARTICIPATING ACTORS:</b>	<ul style="list-style-type: none"> <li>• None</li> </ul>	
<b>OTHER INTERESTED STAKEHOLDERS:</b>	<ul style="list-style-type: none"> <li>• Directors</li> <li>• Management</li> </ul>	
<b>DESCRIPTION:</b>	<p>This use case describes the event of a demographics report being generated at the end of each year. The information about volunteers and students will be retrieved from the Person table, Volunteer table and Student table. This report will show information about the different members' part of the academy.</p>	
<b>PRE-CONDITION:</b>	There must be registered students of the academy.	
<b>TRIGGER:</b>	Time detects it is the end of the TRWLA year.	
<b>TYPICAL COURSE OF EVENTS:</b>	<b>Actor Action</b>	<b>System Response</b>
	<b>Step 1:</b> Time detects it is the end of the TRWLA year.	<b>Step 2:</b> The system accesses the <u>Person</u> table, <u>Volunteer</u> table and <u>Student</u> table.
		<b>Step 3:</b> The system retrieves the following information from the <u>Person</u> table: <ul style="list-style-type: none"> <li>• DateOf Birth</li> <li>• Race</li> <li>• HomeLanguage</li> </ul> The following information from the <u>Student</u> table: <ul style="list-style-type: none"> <li>• Residence</li> <li>• YearOfStudy</li> <li>• Degree</li> </ul> The following information from the <u>Volunteer</u> table: <ul style="list-style-type: none"> <li>• VolunteerID</li> </ul>
		<b>Step 4:</b> The system uses the information to calculate the total of the types for each field.
		<b>Step 5:</b> The system creates a report

		displaying the totals calculated in the previous step.
		<b>Step 7:</b> The system notifies management of the report generation.
<b>ALTERNATE COURSES:</b>	None	
<b>CONCLUSION:</b>	A notification is sent to management	
<b>POST-CONDITION:</b>	A report is generated.	
<b>BUSINESS RULES</b>	<ul style="list-style-type: none"> <li>Only authorized management will have access to the report</li> </ul>	
<b>IMPLEMENTATION CONSTRAINTS AND SPECIFICATIONS</b>	<ul style="list-style-type: none"> <li>The report must be generated during off-peak hours of the system.</li> </ul>	
<b>ASSUMPTIONS:</b>	<ul style="list-style-type: none"> <li>None</li> </ul>	
<b>OPEN ISSUES:</b>	None	

<b>USE CASE NAME:</b>	Generate Event Popularity Report	<b>USE CASE TYPE</b>
<b>USE CASE ID:</b>	9.5	Business Requirements: <input type="checkbox"/>
<b>PRIORITY:</b>	High	System Analysis: <input checked="" type="checkbox"/>
<b>SOURCE:</b>	TuksRes Women In Leadership Academy	System Design: <input type="checkbox"/>
<b>PRIMARY BUSINESS ACTOR</b>	Time	
<b>PRIMARY SYSTEM ACTOR</b>	None	
<b>OTHER PARTICIPATING ACTORS:</b>	<ul style="list-style-type: none"> <li>• None</li> </ul>	
<b>OTHER INTERESTED STAKEHOLDERS:</b>	<ul style="list-style-type: none"> <li>• Directors</li> <li>• Management</li> </ul>	
<b>DESCRIPTION:</b>	<p>This use case describes the event of an event popularity report being generated at the end of each year. The information about the types events taken place will be retrieved from the Attendance table, <u>CommunityOutreach</u> table, Function table and Lecture table. This report will show information about the popularity of events throughout the year.</p>	
<b>PRE-CONDITION:</b>	There must be events that have taken place throughout the year.	
<b>TRIGGER:</b>	Time detects it is the end of the TRWLA year.	
<b>TYPICAL COURSE OF EVENTS:</b>	<b>Actor Action</b>	<b>System Response</b>
	<b>Step 1:</b> Time detects it is the end of the TRWLA year.	<b>Step 2:</b> The system accesses the <u>Attendance</u> table, <u>CommunityOutreach</u> table, <u>Function</u> table and <u>Lecture</u> table.
		<b>Step 3:</b> The system retrieves the following information from the <u>Attendance</u> table: <ul style="list-style-type: none"> <li>• AttendanceID</li> </ul> The following information from the <u>CommunityOutreach</u> table: <ul style="list-style-type: none"> <li>• CommID</li> </ul> The following information from the <u>Function</u> table: <ul style="list-style-type: none"> <li>• FunctionID</li> </ul> The following information from the <u>Lecture</u> table: <ul style="list-style-type: none"> <li>• LectureID</li> </ul>
		<b>Step 4:</b> The system uses the information to calculate the total attendance for each event type throughout the TRWLA year.

		<b>Step 5:</b> The system creates a report displaying the calculation from the previous step with the corresponding event type.
		<b>Step 7:</b> The system notifies management of the report generation.
<b>ALTERNATE COURSES:</b>	None	
<b>CONCLUSION:</b>	A notification is sent to management.	
<b>POST-CONDITION:</b>	A report is generated.	
<b>BUSINESS RULES</b>	<ul style="list-style-type: none"> <li>Only authorized management will have access to the report</li> </ul>	
<b>IMPLEMENTATION CONSTRAINTS AND SPECIFICATIONS</b>	<ul style="list-style-type: none"> <li>The report must be generated during off-peak hours of the system.</li> </ul>	
<b>ASSUMPTIONS:</b>	<ul style="list-style-type: none"> <li>None</li> </ul>	
<b>OPEN ISSUES:</b>	None	

<b>USE CASE NAME:</b>	Generate Donations Report	<b>USE CASE TYPE</b>
<b>USE CASE ID:</b>	9.6	Business Requirements: <input type="checkbox"/>
<b>PRIORITY:</b>	High	System Analysis: <input checked="" type="checkbox"/>
<b>SOURCE:</b>	TuksRes Women In Leadership Academy	System Design: <input type="checkbox"/>
<b>PRIMARY BUSINESS ACTOR</b>	Time	
<b>PRIMARY SYSTEM ACTOR</b>	None	
<b>OTHER PARTICIPATING ACTORS:</b>	<ul style="list-style-type: none"> <li>• None</li> </ul>	
<b>OTHER INTERESTED STAKEHOLDERS:</b>	<ul style="list-style-type: none"> <li>• Directors</li> <li>• Management</li> </ul>	
<b>DESCRIPTION:</b>	This use case describes the event of a donations report being generated at the end of each year. Information will be retrieved from the Donor table and Donation table. The use case concludes by a report being generated with relevant donation information.	
<b>PRE-CONDITION:</b>	TRWLA must have received donations throughout the year.	
<b>TRIGGER:</b>	Time detects it is the end of the TRWLA year.	
<b>TYPICAL COURSE</b>	<b>Actor Action</b>	<b>System Response</b>
<b>OF EVENTS:</b>	<b>Step 1:</b> Time detects it is the end of the TRWLA year.	<b>Step 2:</b> The system accesses the <u>DonorLog</u> table and <u>Donation</u> table.
		<b>Step 3:</b> The system retrieves the following information from the <u>DonorLog</u> table: <ul style="list-style-type: none"> <li>• DonorName</li> <li>• DonorSurname</li> <li>• DonorOrganisation</li> </ul> The following information from the <u>Donation</u> table: <ul style="list-style-type: none"> <li>• DonationAmount</li> </ul>
		<b>Step 4:</b> The system uses the information to calculate: <ul style="list-style-type: none"> <li>• The total donations for the TRWLA year</li> <li>• The top 5 donation amounts with the donor names</li> </ul> For the TRWLA year.
		<b>Step 5:</b> The system creates a report displaying the calculations from the previous step.
		<b>Step 7:</b> The system notifies



		management of the report generation.
<b>ALTERNATE COURSES:</b>	None	
<b>CONCLUSION:</b>	A notification is sent to management.	
<b>POST-CONDITION:</b>	A report is generated.	
<b>BUSINESS RULES</b>	<ul style="list-style-type: none"><li>• Only authorized management will have access to the report</li></ul>	
<b>IMPLEMENTATION CONSTRAINTS AND SPECIFICATIONS</b>	<ul style="list-style-type: none"><li>• The report must be generated during off-peak hours of the system.</li></ul>	
<b>ASSUMPTIONS:</b>	<ul style="list-style-type: none"><li>• None</li></ul>	
<b>OPEN ISSUES:</b>	None	

<b>USE CASE NAME:</b>	Generate User Statistics Report	<b>USE CASE TYPE</b>
<b>USE CASE ID:</b>	9.7	Business Requirements: <input type="checkbox"/>
<b>PRIORITY:</b>	High	System Analysis: <input checked="" type="checkbox"/>
<b>SOURCE:</b>	TuksRes Women In Leadership Academy	System Design: <input type="checkbox"/>
<b>PRIMARY BUSINESS ACTOR</b>	Time	
<b>PRIMARY SYSTEM ACTOR</b>	None	
<b>OTHER PARTICIPATING ACTORS:</b>	<ul style="list-style-type: none"> <li>• None</li> </ul>	
<b>OTHER INTERESTED STAKEHOLDERS:</b>	<ul style="list-style-type: none"> <li>• Directors</li> <li>• Management</li> </ul>	
<b>DESCRIPTION:</b>	This use case describes the event of a user statistics report being generated at the end of each year. Information will be retrieved from the AuditLog table, Person table and UserType table. The use case concludes by a report being generated with relevant user information.	
<b>PRE-CONDITION:</b>	Users must have access to the system.	
<b>TRIGGER:</b>	Time detects it is the end of the TRWLA year.	
<b>TYPICAL COURSE</b>	<b>Actor Action</b>	<b>System Response</b>
<b>OF EVENTS:</b>	<b>Step 1:</b> Time detects it is the end of the TRWLA year.	<b>Step 2:</b> The system accesses the <u>Audit Log</u> table and <u>Person</u> table.
		<b>Step 3:</b> The system retrieves the following information from the <u>Audit Log</u> table <ul style="list-style-type: none"> <li>• LoginTime</li> <li>• LoginDate</li> <li>• LoginDuration</li> </ul> The following information from the <u>Person</u> table <ul style="list-style-type: none"> <li>• PersonID</li> </ul> The following information from the <u>User Type</u> table <ul style="list-style-type: none"> <li>• UserID</li> </ul>
		<b>Step 4:</b> The system uses the information to calculate: <ul style="list-style-type: none"> <li>• The average time a user spent on the system</li> <li>• The average time different user types spent on the system and</li> <li>• The total number of times the system was accessed per user</li> </ul>

		type <ul style="list-style-type: none"> <li>The number of users that accessed the system throughout the TRWLA year.</li> </ul>
		<b>Step 5:</b> The system creates a report displaying the calculations from the previous step.
		<b>Step 7:</b> The system notifies management of the report generation.
<b>ALTERNATE COURSES:</b>	None	
<b>CONCLUSION:</b>	A notification is sent to management.	
<b>POST-CONDITION:</b>	A report is generated.	
<b>BUSINESS RULES</b>	<ul style="list-style-type: none"> <li>Only authorized management will have access to the report</li> </ul>	
<b>IMPLEMENTATION CONSTRAINTS AND SPECIFICATIONS</b>	<ul style="list-style-type: none"> <li>The report must be generated during off-peak hours of the system.</li> </ul>	
<b>ASSUMPTIONS:</b>	<ul style="list-style-type: none"> <li>None</li> </ul>	
<b>OPEN ISSUES:</b>	None	

## 2.3.10 Donation Subsystem Narratives

<b>USE CASE NAME:</b>	Add Donor	<b>USE CASE TYPE</b>
<b>USE CASE ID:</b>	10.1	Business Requirements: <input type="checkbox"/>
<b>PRIORITY:</b>	Medium	System Analysis: <input checked="" type="checkbox"/>
<b>SOURCE:</b>	TRWLA System	System Design: <input type="checkbox"/>
<b>PRIMARY BUSINESS ACTOR</b>	Donor	
<b>PRIMARY SYSTEM ACTOR</b>	None	
<b>OTHER PARTICIPATING ACTORS:</b>	<ul style="list-style-type: none"> <li>None</li> </ul>	
<b>OTHER INTERESTED STAKEHOLDERS:</b>	<ul style="list-style-type: none"> <li>Directors</li> </ul>	
<b>DESCRIPTION:</b>	<p>This use case describes the event of adding a donor to the system. A donor wants to add their details to the system and will provide the system with their details which will be saved onto the system for future use. This use case concludes by a donor being added to the system.</p>	
<b>PRE-CONDITION:</b>		
<b>TRIGGER:</b>	A donor wants to add their details to the system.	
<b>TYPICAL COURSE</b>	<b>Actor Action</b>	<b>System Response</b>
<b>OF EVENTS:</b>	<b>Step 1:</b> A donor wants to add their details to the system.	<b>Step 2:</b> The system will display the donations section.
		<b>Step 3:</b> The system will prompt the user to supply the following required information: <ul style="list-style-type: none"> <li>Name</li> <li>Surname</li> <li>Organization Name</li> <li>Phone Number</li> <li>Email Address</li> </ul>
	<b>Step 4:</b> The donor enters the required personal information into the required fields and submits their information.	<b>Step 5:</b> The system verifies that all required fields have been filled in and that the fields contain valid details i.e. <ul style="list-style-type: none"> <li>Name (Maximum 35 characters)</li> <li>Surname (Maximum 35 characters)</li> <li>Email Address (Minimum 7 characters, Maximum 255 characters. Must be a full validated email address)</li> </ul>

		<ul style="list-style-type: none"> <li>Organization Name (Maximum 35 characters)</li> </ul> Phone Number (Max 15 characters (No spaces))
		<b>Step 6:</b> The system confirms all required fields are complete.
		<b>Step 7:</b> The system saves the following information to the <u>Donor</u> table: <ul style="list-style-type: none"> <li>Name</li> <li>Surname</li> <li>Organization Name</li> <li>Phone Number</li> <li>Email Address</li> </ul>
		<b>Step 9:</b> The system notifies the donor that the information was successfully stored.
<b>ALTERNATE COURSES:</b>	<b>Alt-Step 6:</b> The required donor information is incomplete; the system notifies the donor that their information is incomplete. ➔ Go to Step 4.	
<b>CONCLUSION:</b>	A confirmation of donation addition is sent out.	
<b>POST-CONDITION:</b>	A donor is added to the Donor table.	
<b>BUSINESS RULES</b>	<ul style="list-style-type: none"> <li>Only directors will have access to donor information</li> </ul>	
<b>IMPLEMENTATION CONSTRAINTS AND SPECIFICATIONS</b>	<ul style="list-style-type: none"> <li>None</li> </ul>	
<b>ASSUMPTIONS:</b>	<ul style="list-style-type: none"> <li>None</li> </ul>	
<b>OPEN ISSUES:</b>	None	

<b>USE CASE NAME:</b>	Receive Donation	<b>USE CASE TYPE</b>
<b>USE CASE ID:</b>	10.2	Business Requirements: <input type="checkbox"/>
<b>PRIORITY:</b>	Medium	System Analysis: <input checked="" type="checkbox"/>
<b>SOURCE:</b>	TRWLA System	System Design: <input type="checkbox"/>
<b>PRIMARY BUSINESS ACTOR</b>	Donor	
<b>PRIMARY SYSTEM ACTOR</b>	None	
<b>OTHER PARTICIPATING ACTORS:</b>	<ul style="list-style-type: none"> <li>None</li> </ul>	
<b>OTHER INTERESTED STAKEHOLDERS:</b>	<ul style="list-style-type: none"> <li>Directors</li> </ul>	
<b>DESCRIPTION:</b>	<p>This use case describes the event of receiving a donation made to TRWLA. A donor will donate to TRWLA and then provide their details for verification and then provide the donation details. The donation details will then be saved. This use case concludes by TRWLA receiving a donation.</p>	
<b>PRE-CONDITION:</b>	The donor's details must be saved to the system.	
<b>TRIGGER:</b>	A donor donates money to TRWLA.	
<b>TYPICAL COURSE</b>	<b>Actor Action</b>	<b>System Response</b>
<b>OF EVENTS:</b>	<b>Step 1:</b> A wants to donate money.	<b>Step 2:</b> The system will display the donations section.
		<b>Step 3:</b> The system will prompt the user to with an option to choose if they are an 'Existing Donor'.
	<b>Step 4:</b> The donor chooses the 'Existing Donor' option.	<b>Step 5:</b> The system will prompt the user to enter their email address.
	<b>Step 6:</b> The donor enters their information.	<b>Step 7:</b> The system verifies that the email address field is filled in and that the field contains valid details i.e. <ul style="list-style-type: none"> <li>Email Address (Minimum 7 characters, Maximum 255 characters. Must be a full validated email address)</li> </ul>
		<b>Step 8:</b> The system prompts the user to enter the donation amount and submit a proof of payment.
	<b>Step 9:</b> The user enters the donation amount and submits a proof of payment.	<b>Step 10:</b> The system saves the following information <u>Donation</u> table: <ul style="list-style-type: none"> <li>DonationAmount</li> <li>DateSubmitted</li> <li>Proof of payment</li> </ul>

		<b>Step 11:</b> The system notifies the donor that the information was successfully stored.
<b>ALTERNATE COURSES:</b>	<b>Alt-Step 7:</b> The email address provided is incorrect, the system notifies the donor that the information is incorrect. → Go to Step 5.	
<b>CONCLUSION:</b>	A confirmation of donation receipt is sent out.	
<b>POST-CONDITION:</b>	A donation is received by the TRWLA system.	
<b>BUSINESS RULES</b>	<ul style="list-style-type: none"> <li>Only directors will have access to donor information</li> </ul>	
<b>IMPLEMENTATION CONSTRAINTS AND SPECIFICATIONS</b>	<ul style="list-style-type: none"> <li>None</li> </ul>	
<b>ASSUMPTIONS:</b>	<ul style="list-style-type: none"> <li>None</li> </ul>	
<b>OPEN ISSUES:</b>	None	

## 2.4 Conclusion

In this section, the group depicted the functional specifications of the to-be system by means of a use case diagram. The use cases were then further decomposed and explained in the use case narratives. Both of the above provide a clearer explanation to the client as to what their system functionality and processes will be like.



## 3. Process Models

### 3.1 Introduction

3.1.1 The following process models are depicted below; context diagram, fully functional decomposition diagram and high-level, mid-level and primitive-level data flow diagrams. The context diagram depicts how the external actors interact with the system and how the system responds to the external actor. The decomposition diagram displays the hierarchical order that flows throughout the system from the system level, function level, activity level to task level. The data flow diagrams, i.e. High-level, Mid-level and Primitive-Level, are used to describe the process and its data flow from the use case narratives above.

### 3.2 Context Diagram

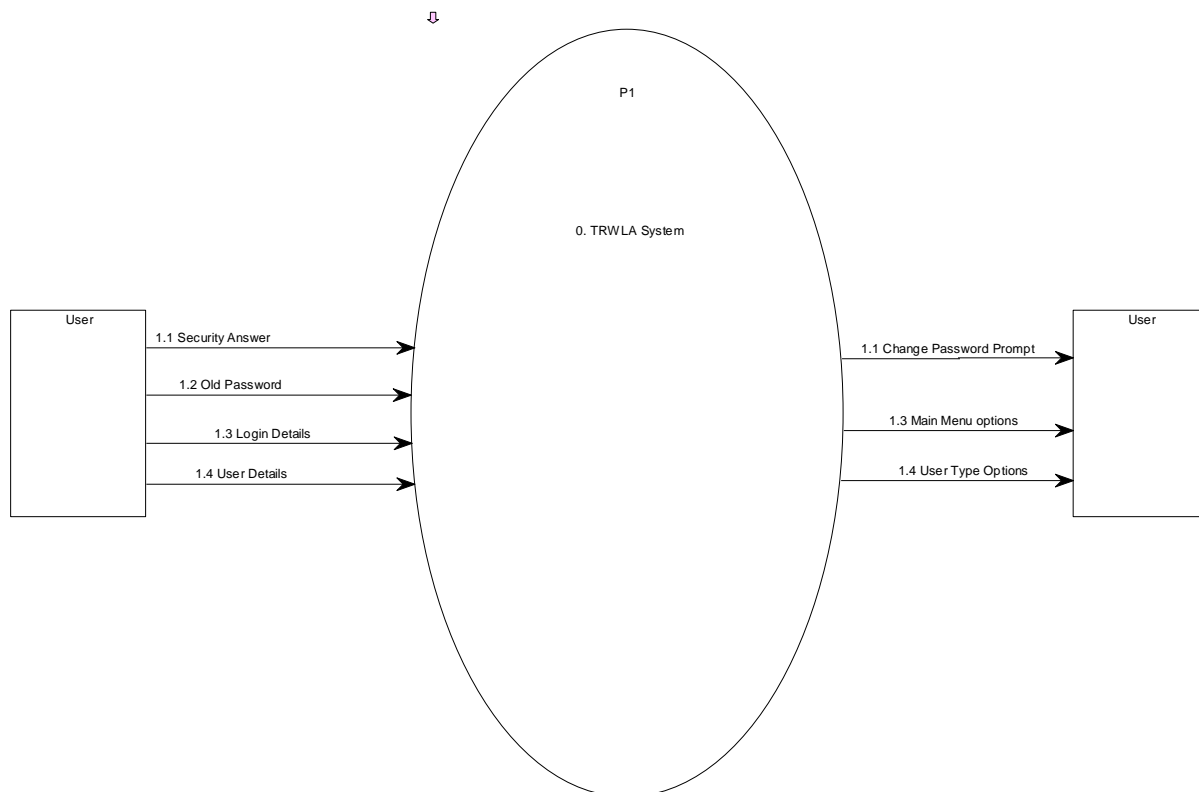


FIGURE 11- CONTEXT DIAGRAM USER SUBSYSTEM

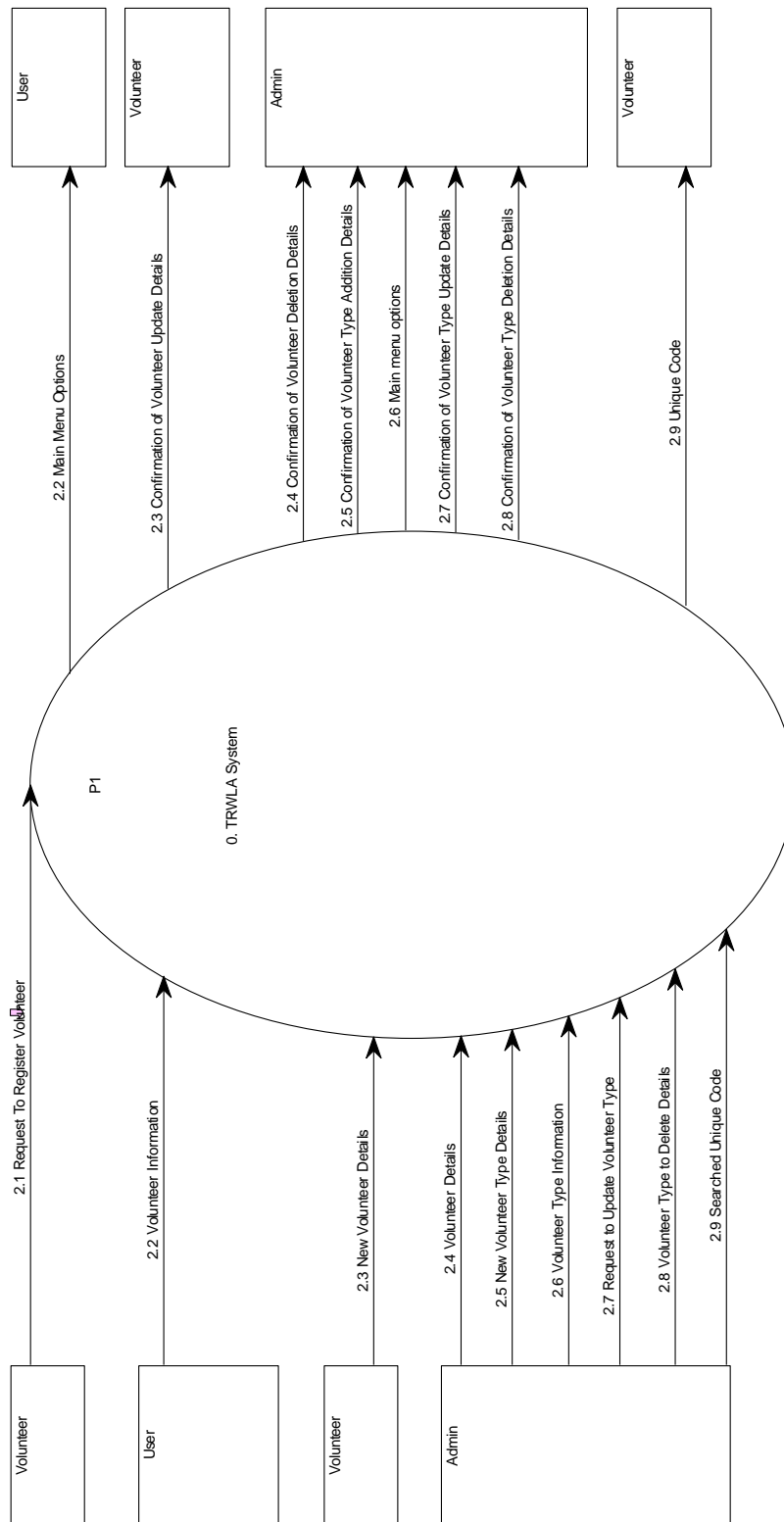


FIGURE 12- CONTEXT DIAGRAM VOLUNTEER SUBSYSTEM

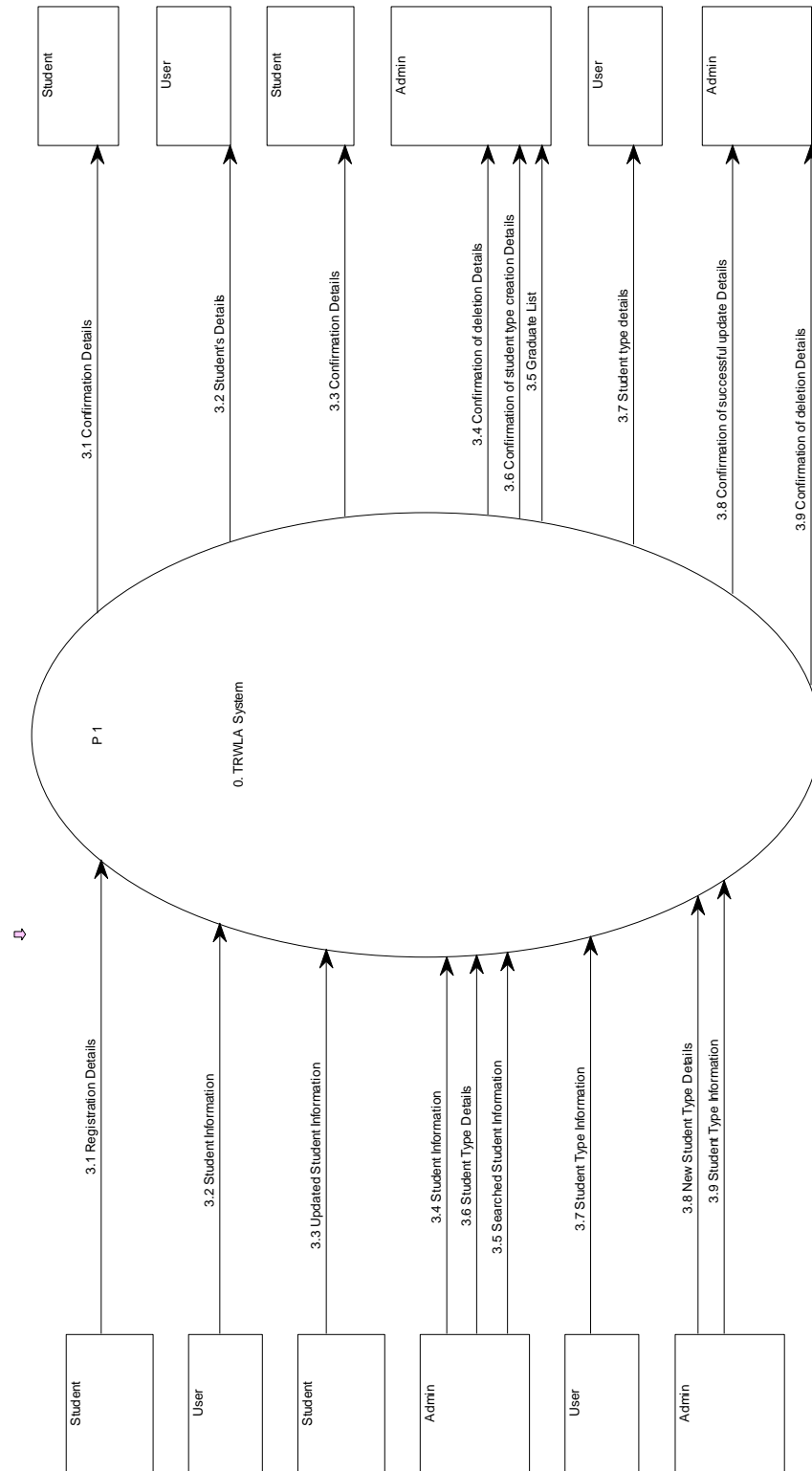


FIGURE 13- CONTEXT DIAGRAM STUDENT SUBSYSTEM

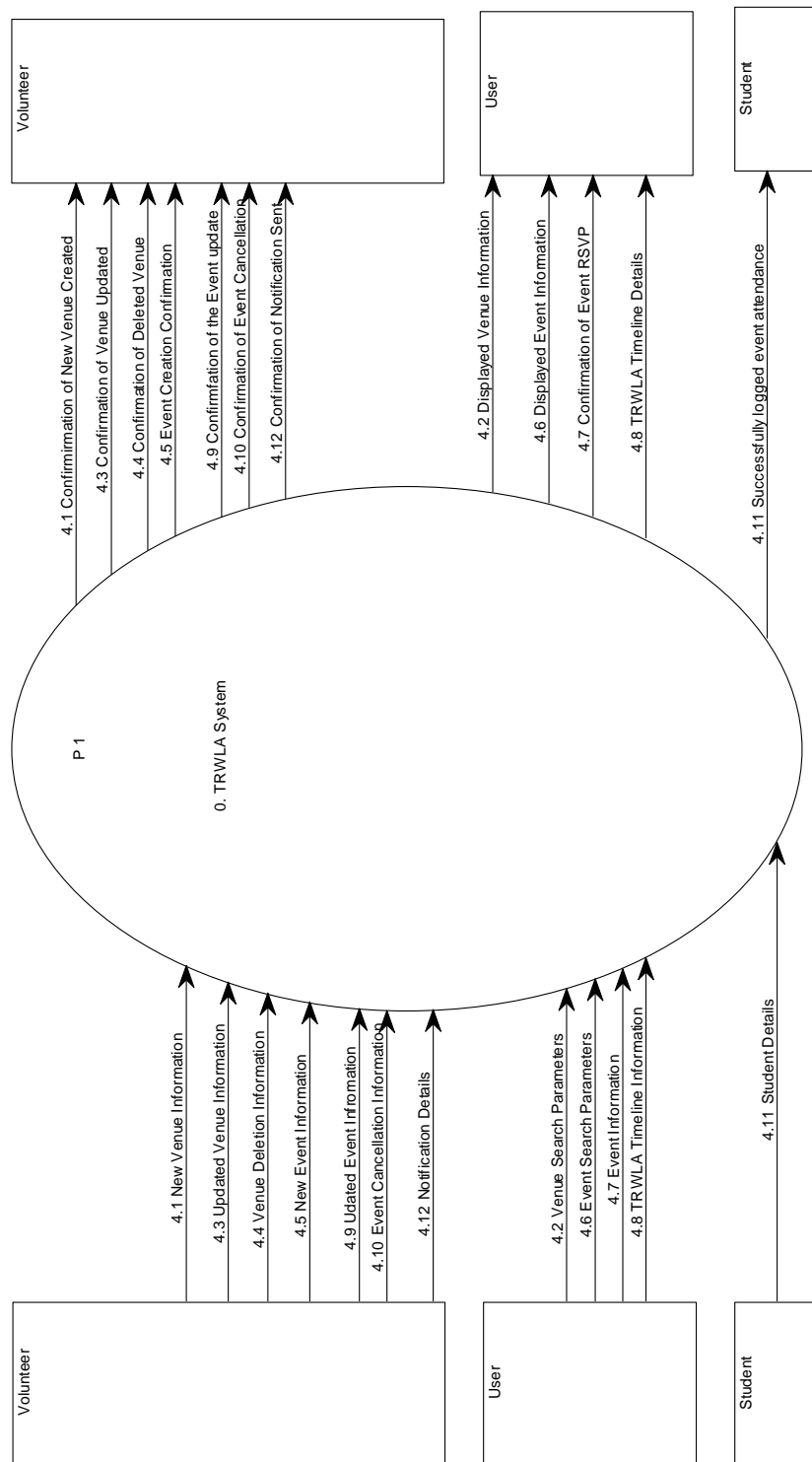


FIGURE 14- CONTEXT DIAGRAM EVENT SUBSYSTEM

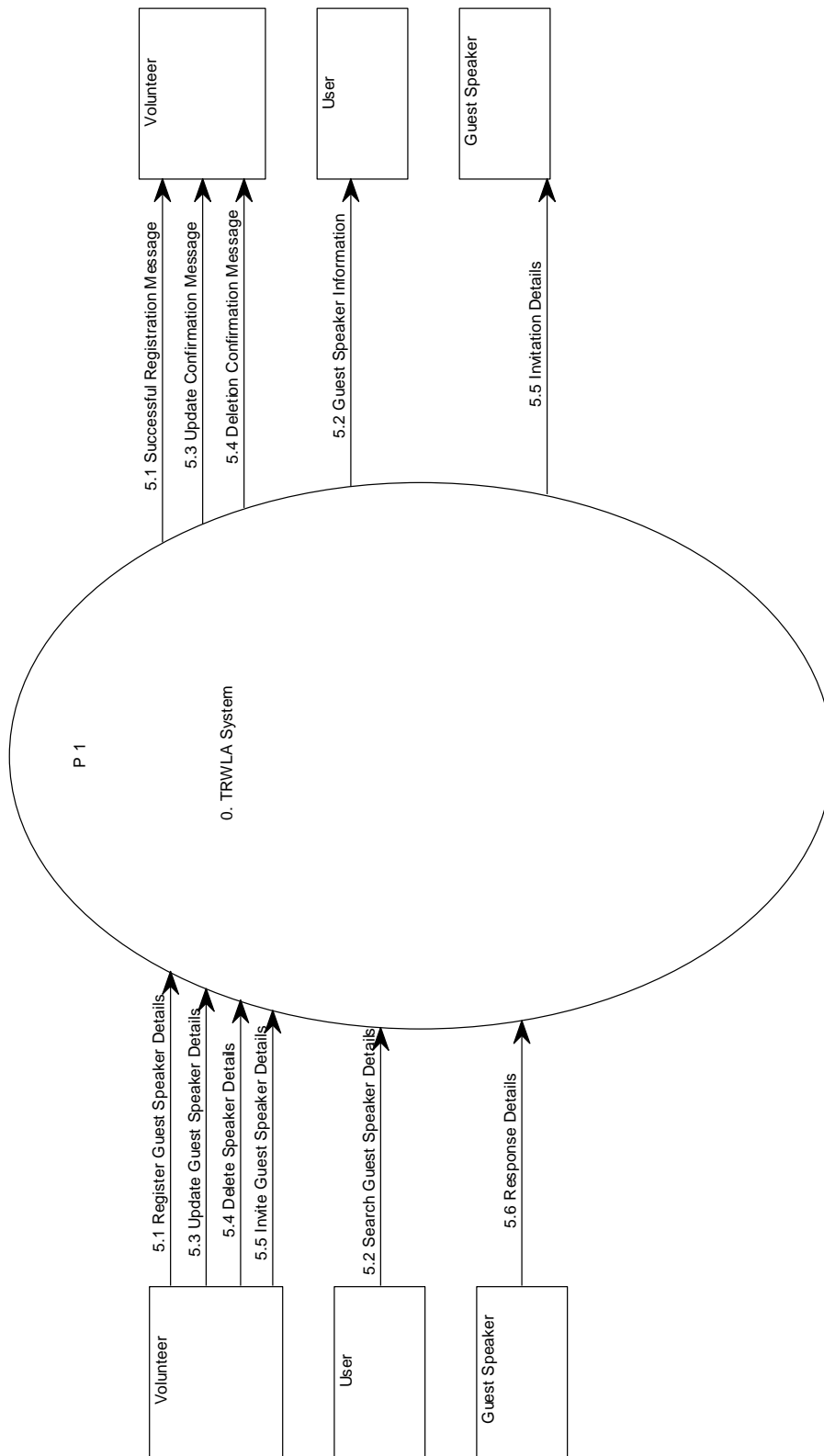


FIGURE 15- CONTEXT DIAGRAM FUNCTION SUBSYSTEM

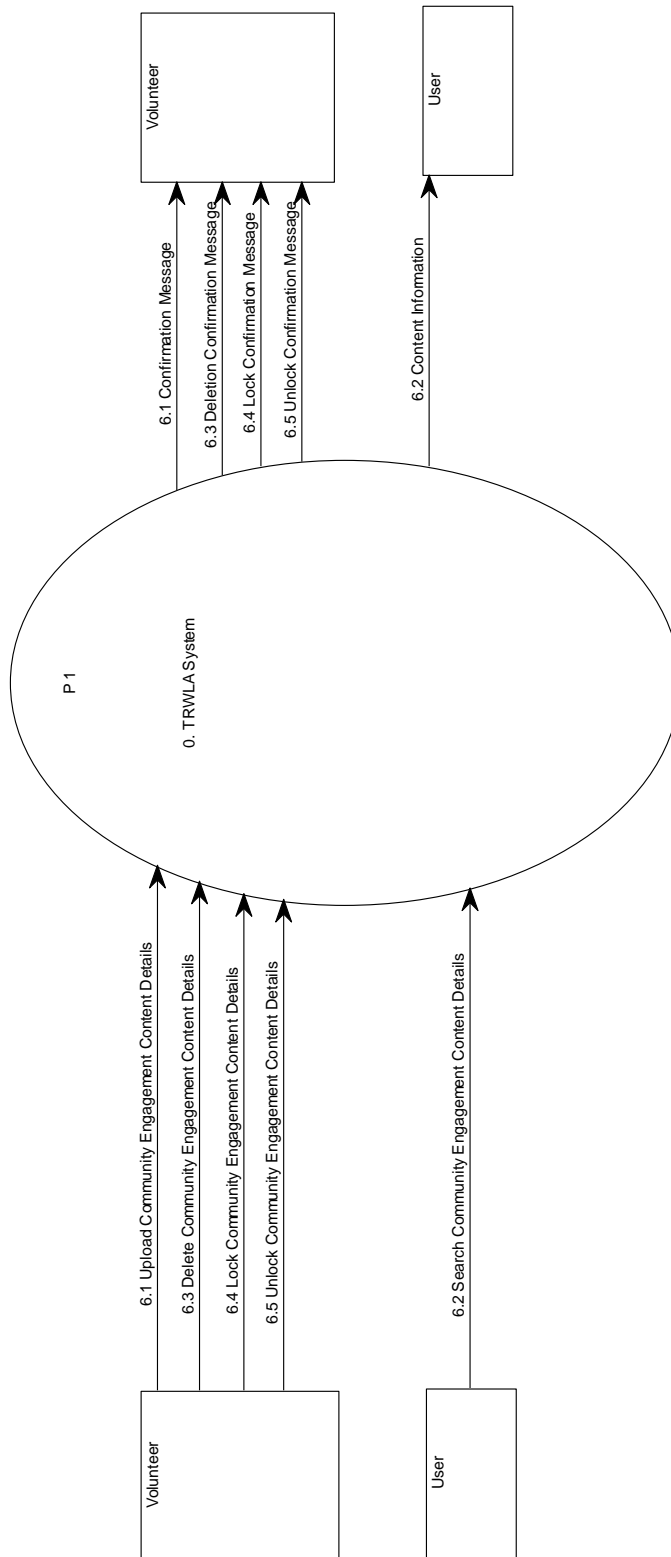


FIGURE 16- CONTEXT DIAGRAM COMMUNITY ENGAGEMENT SUBSYSTEM

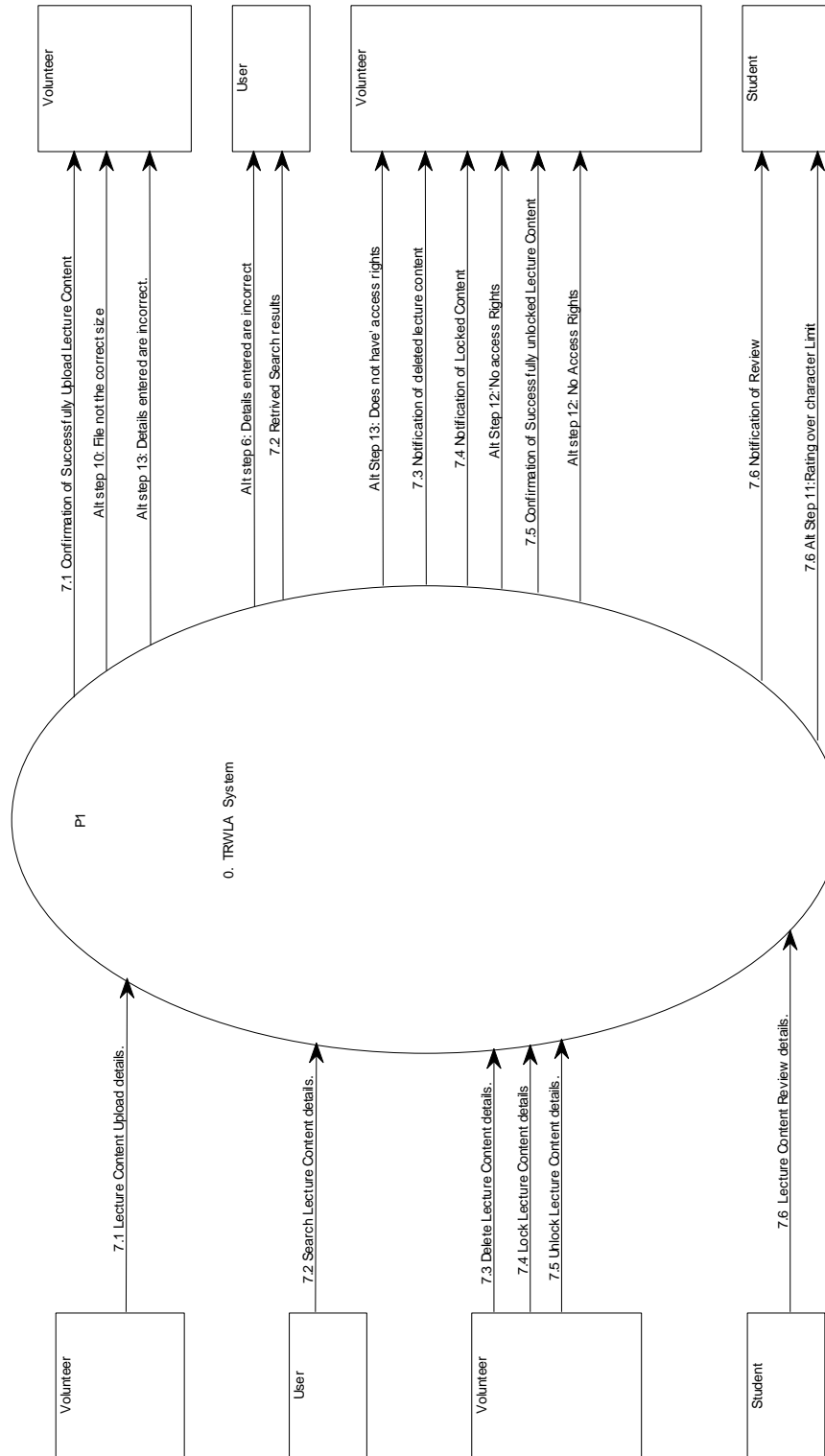


FIGURE 17- CONTEXT DIAGRAM LECTURE SUBSYSTEM

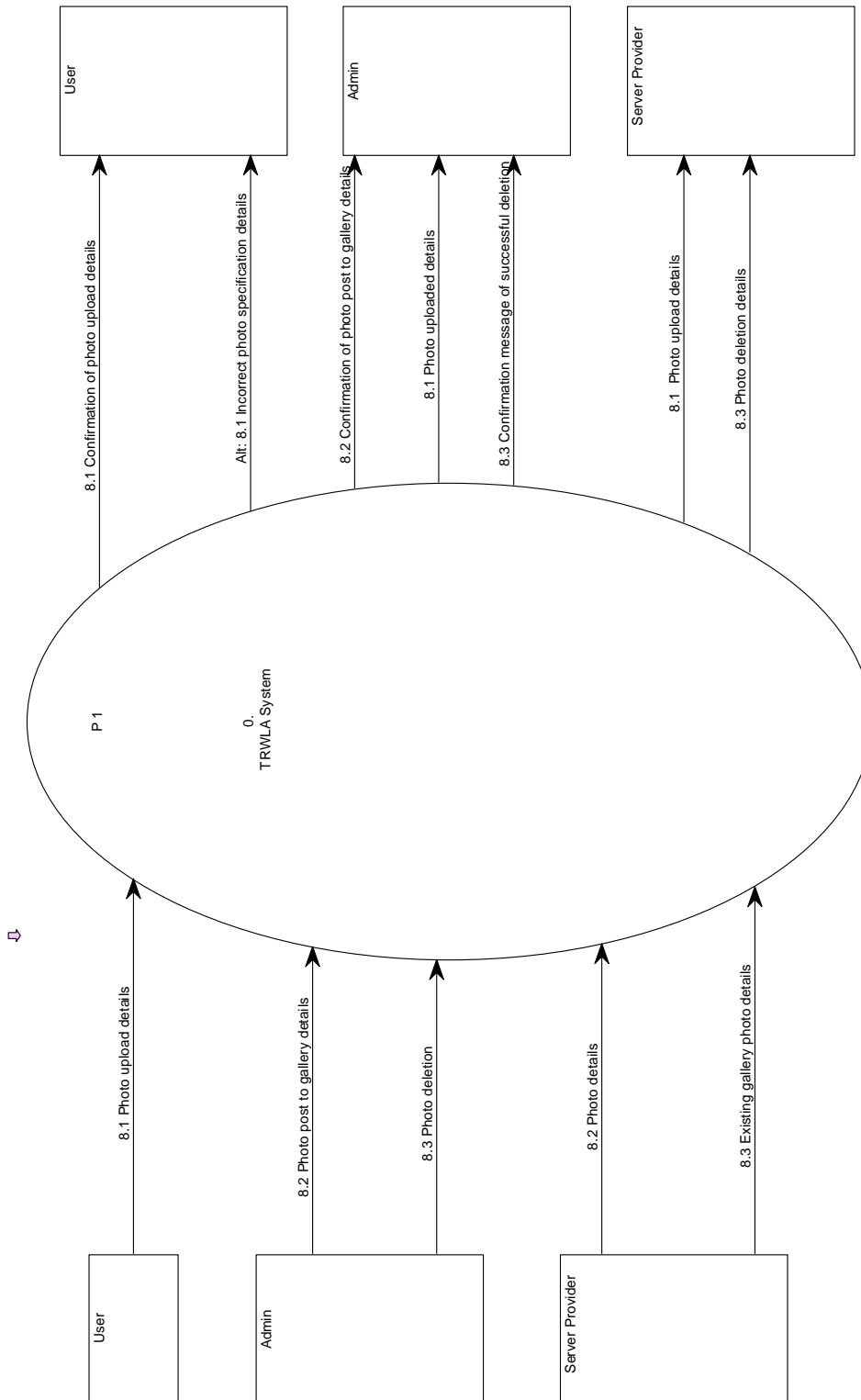


FIGURE 18- CONTEXT DIAGRAM MARKETING SUBSYSTEM



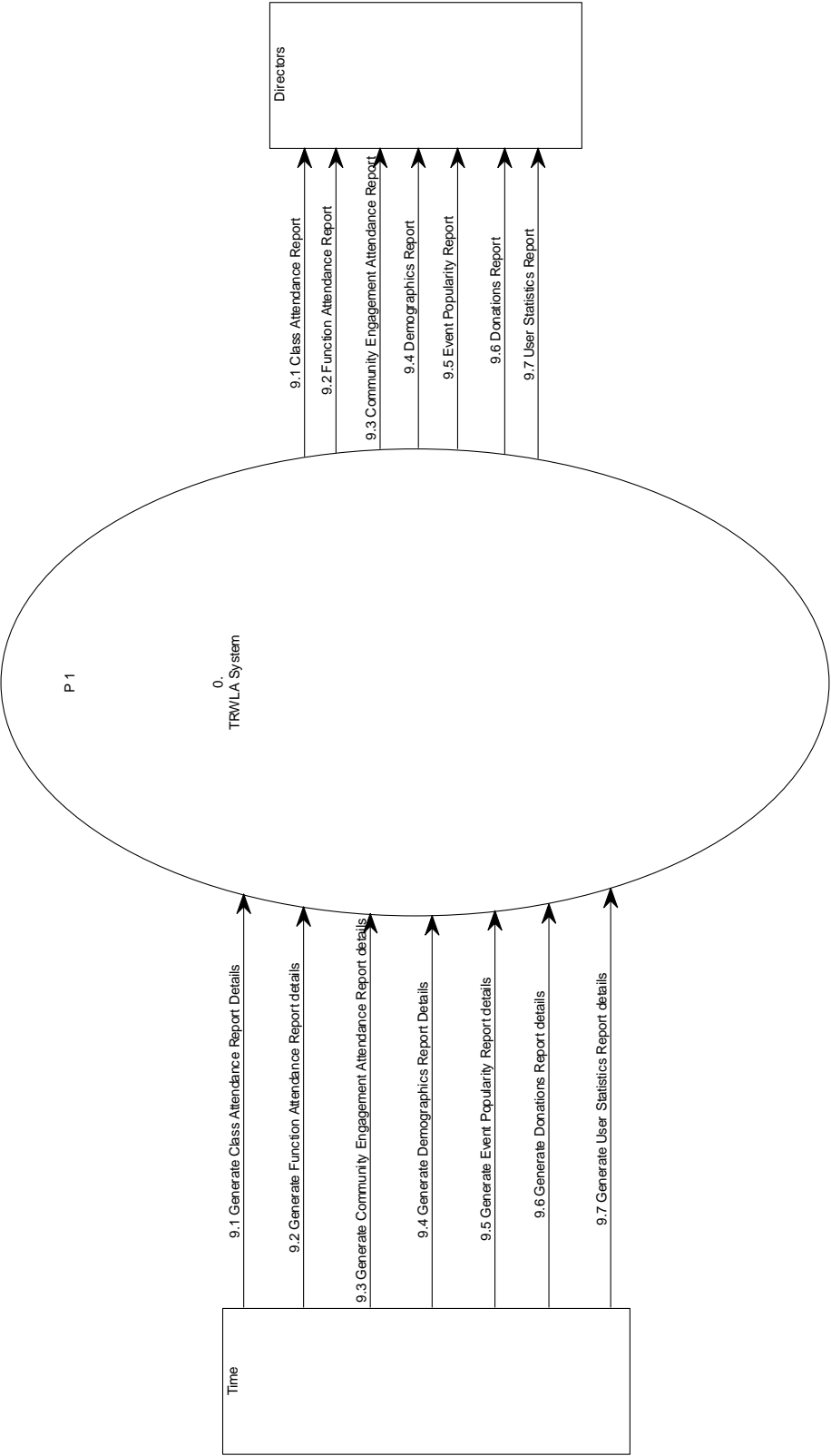


FIGURE 19- CONTEXT DIAGRAM REPORT SUBSYSTEM

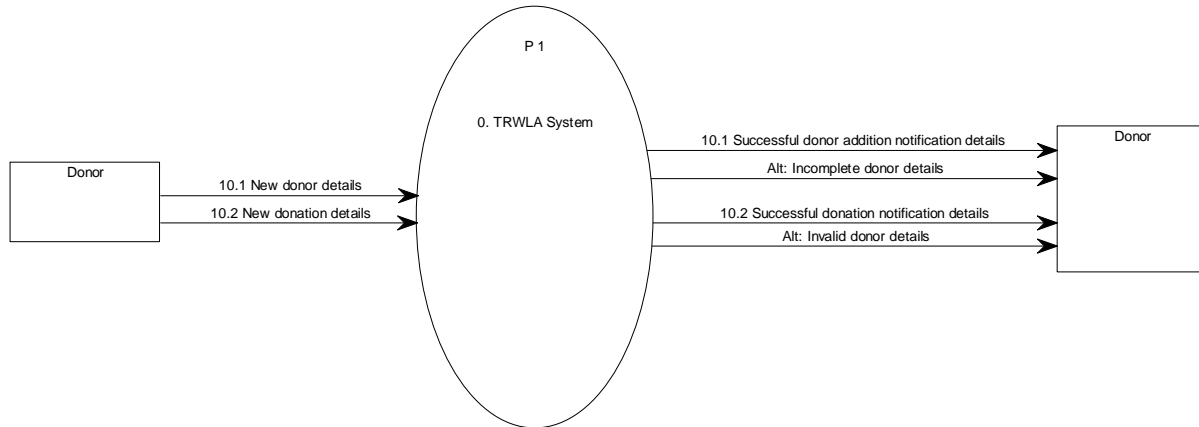


FIGURE 20- CONTEXT DIAGRAM DONATION SUBSYSTEM

### 3.3 Functional Decomposition Diagram

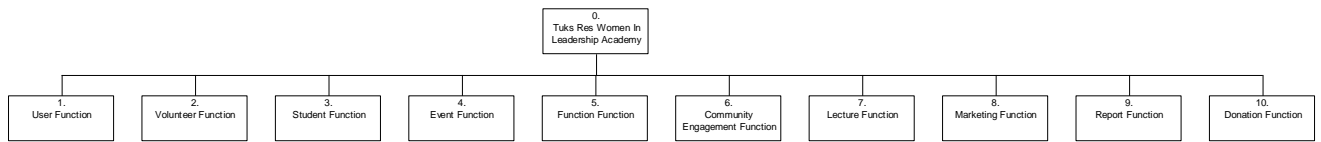


FIGURE 21- DECOMPOSITION DIAGRAM LEVEL 0

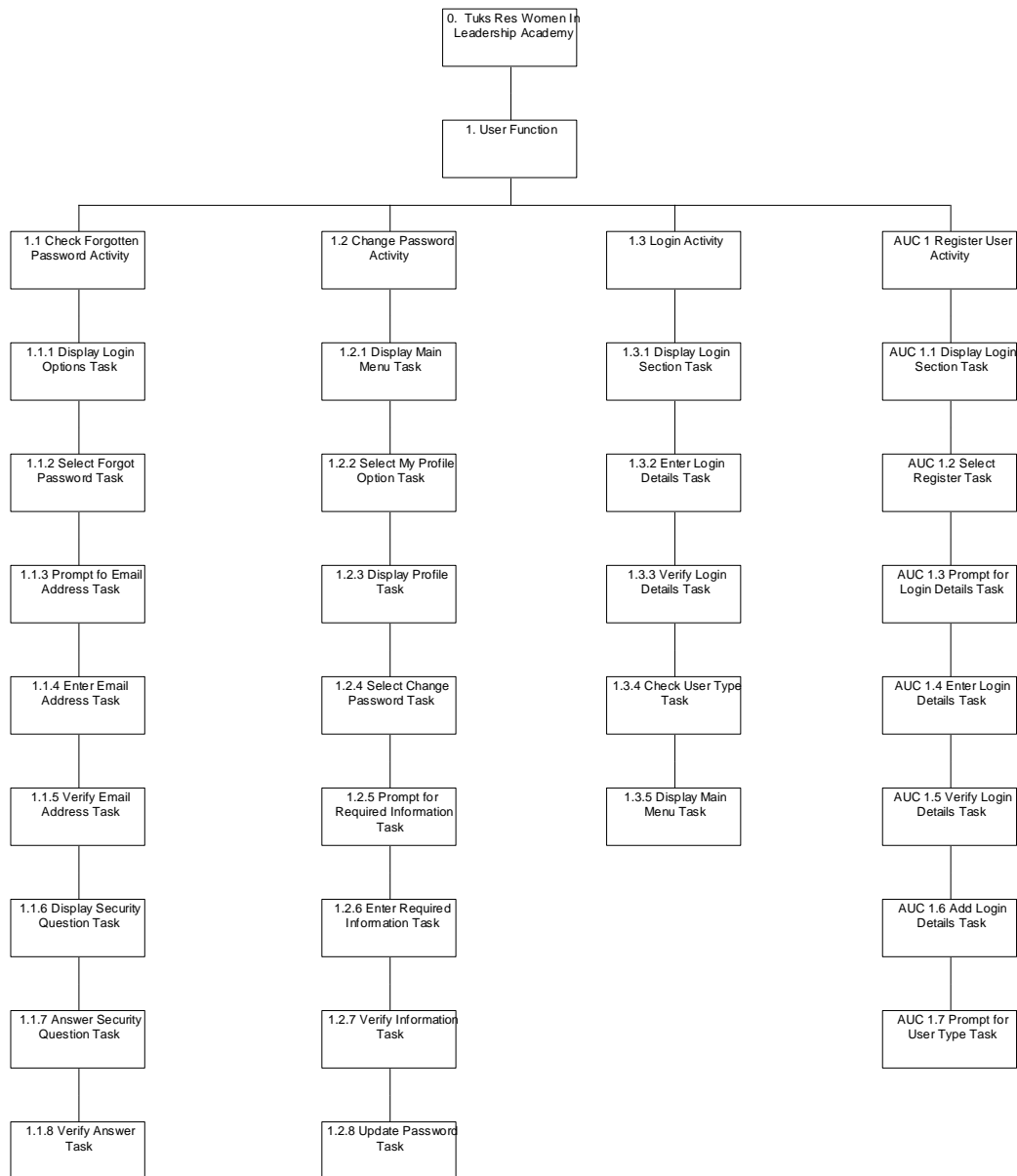


FIGURE 22- DECOMPOSITION DIAGRAM USER FUNCTION 1

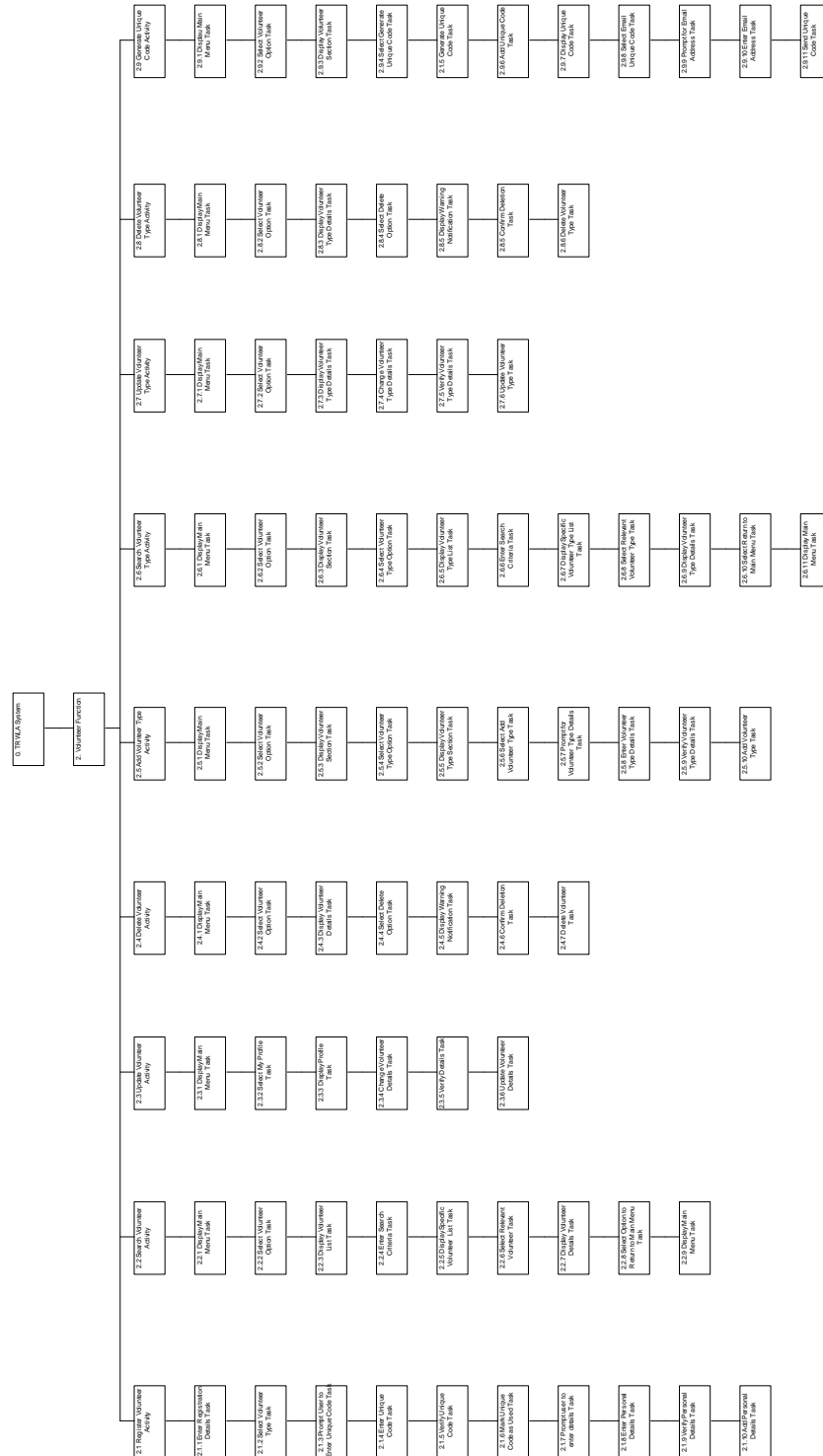


FIGURE 23- DECOMPOSITION DIAGRAM VOLUNTEER FUNCTION 2

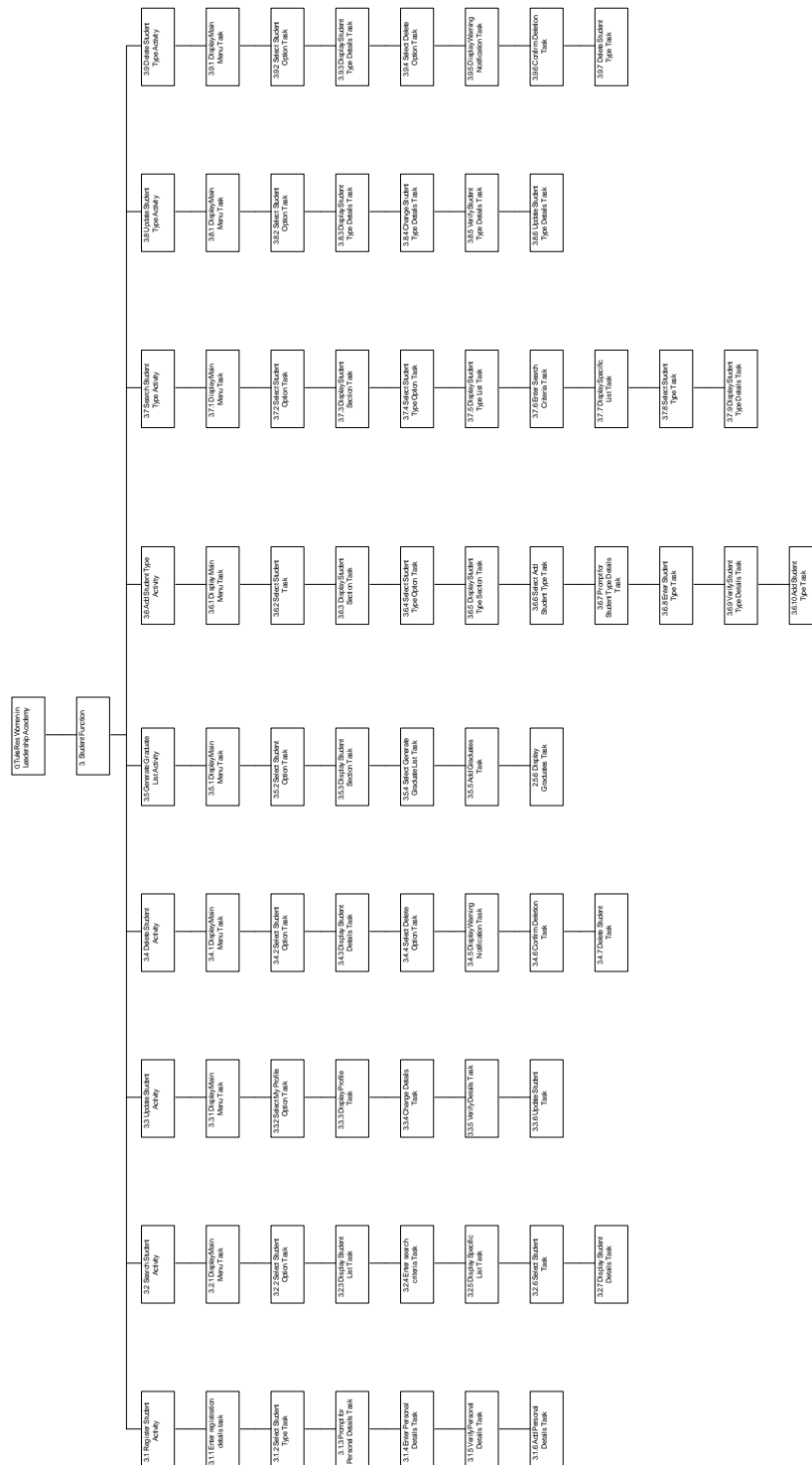


FIGURE 24- DECOMPOSITION DIAGRAM STUDENT FUNCTION 3

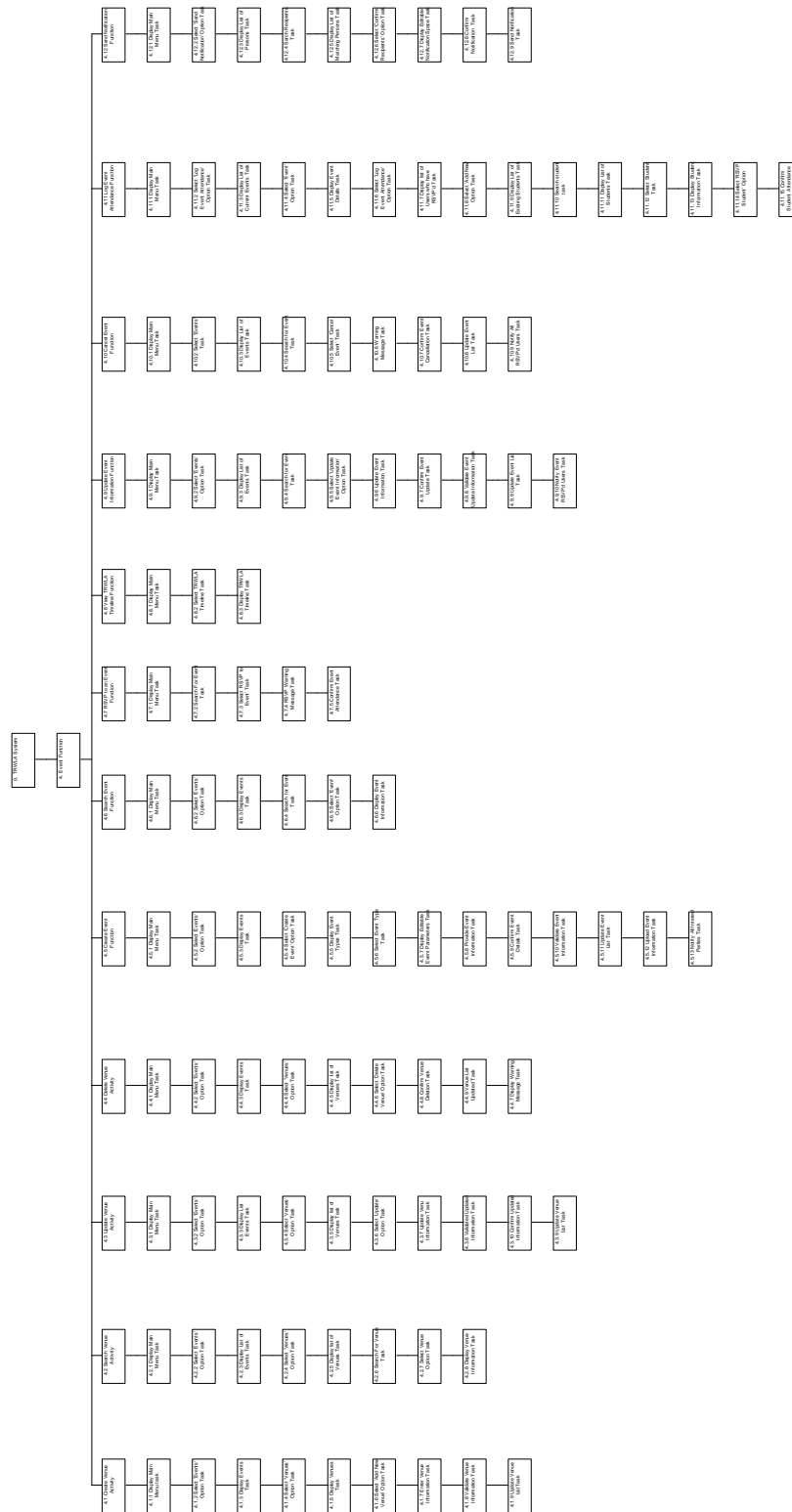


FIGURE 25- DECOMPOSITION DIAGRAM MANAGE EVENTS FUNCTION 4

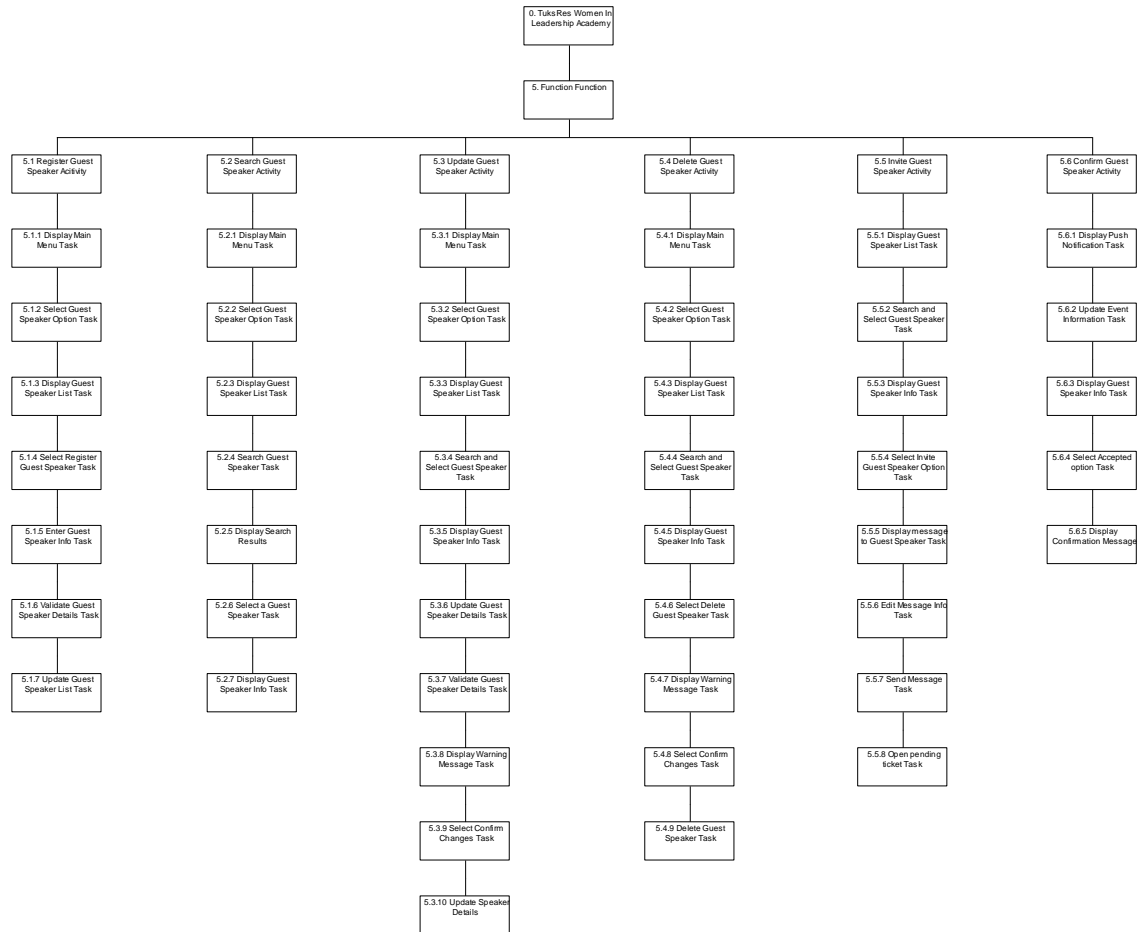


FIGURE 26- DECOMPOSITION DIAGRAM FUNCTION FUNCTION 5

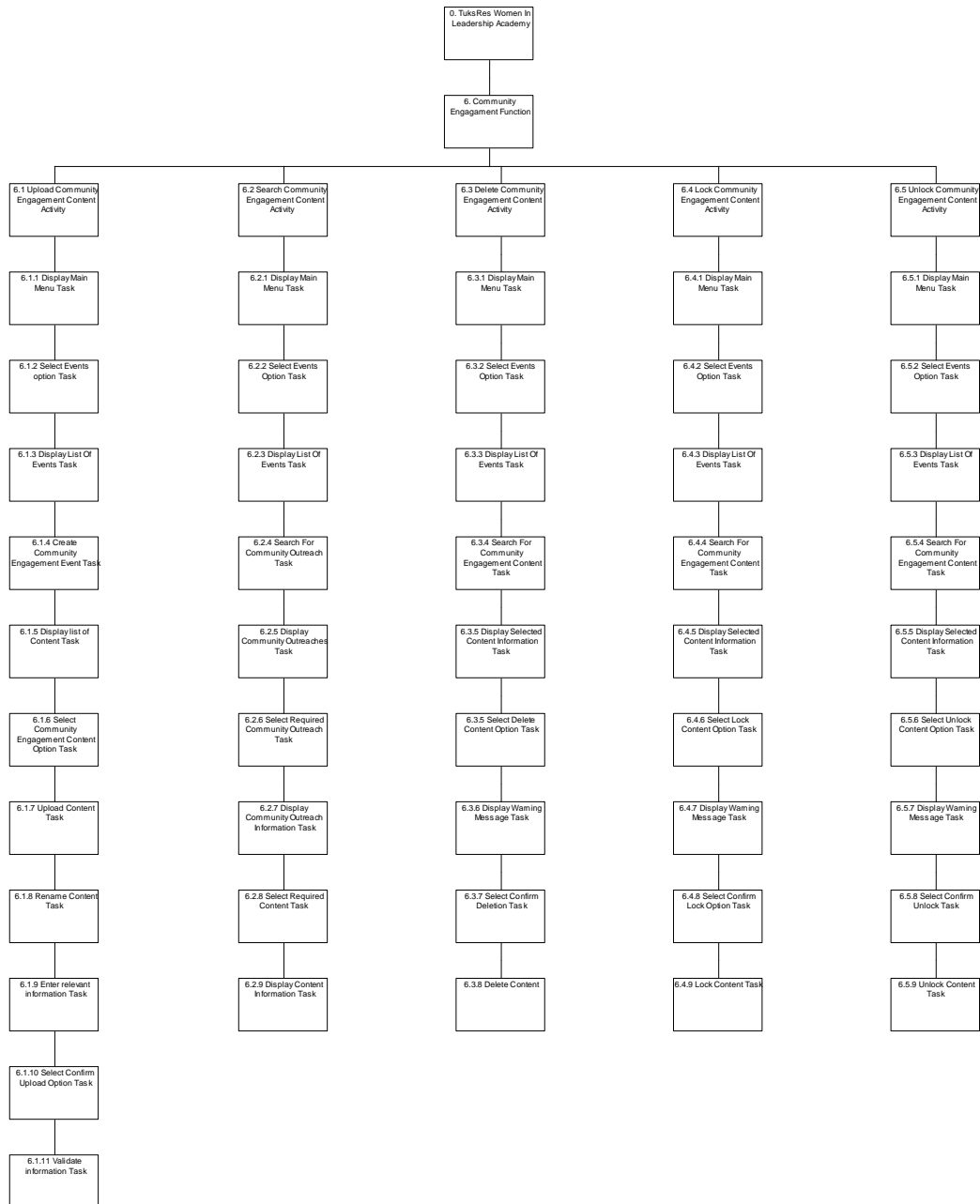


FIGURE 27- DECOMPOSITION DIAGRAM COMMUNITY ENGAGEMENT FUNCTION 6



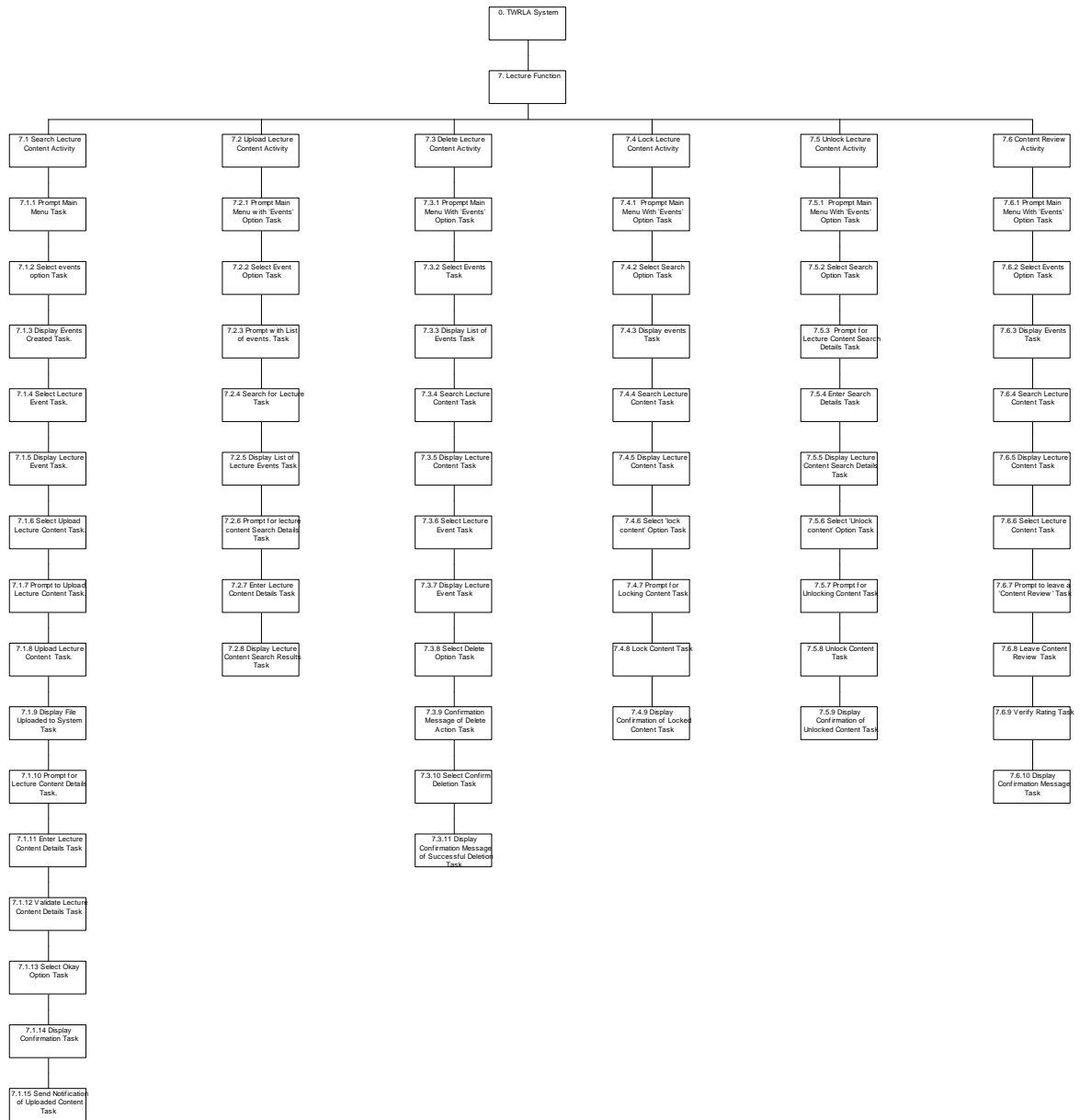


FIGURE 28- DECOMPOSITION DIAGRAM LECTURE FUNCTION 7

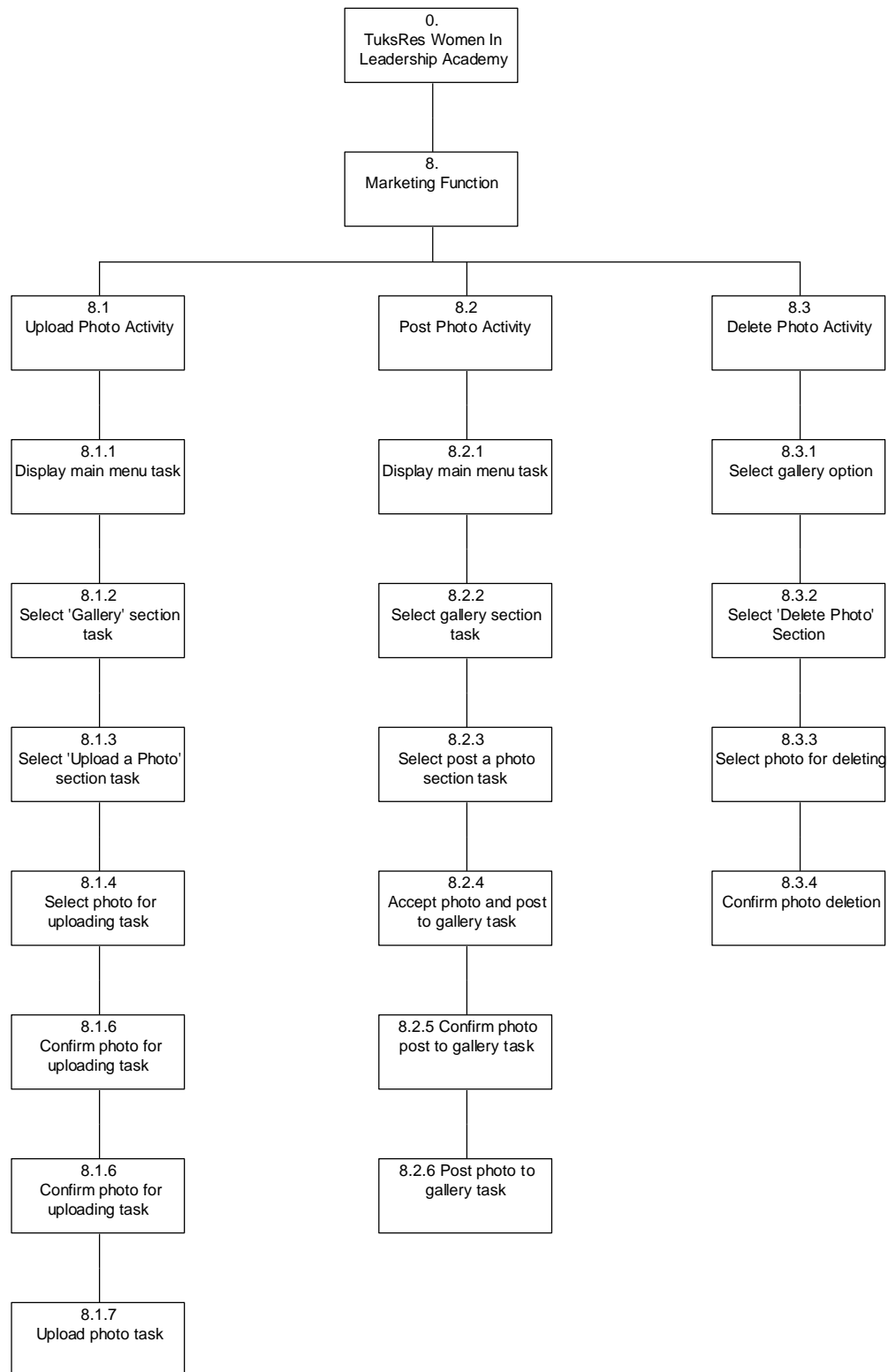


FIGURE 29- DECOMPOSITION DIAGRAM FUNCTION 8

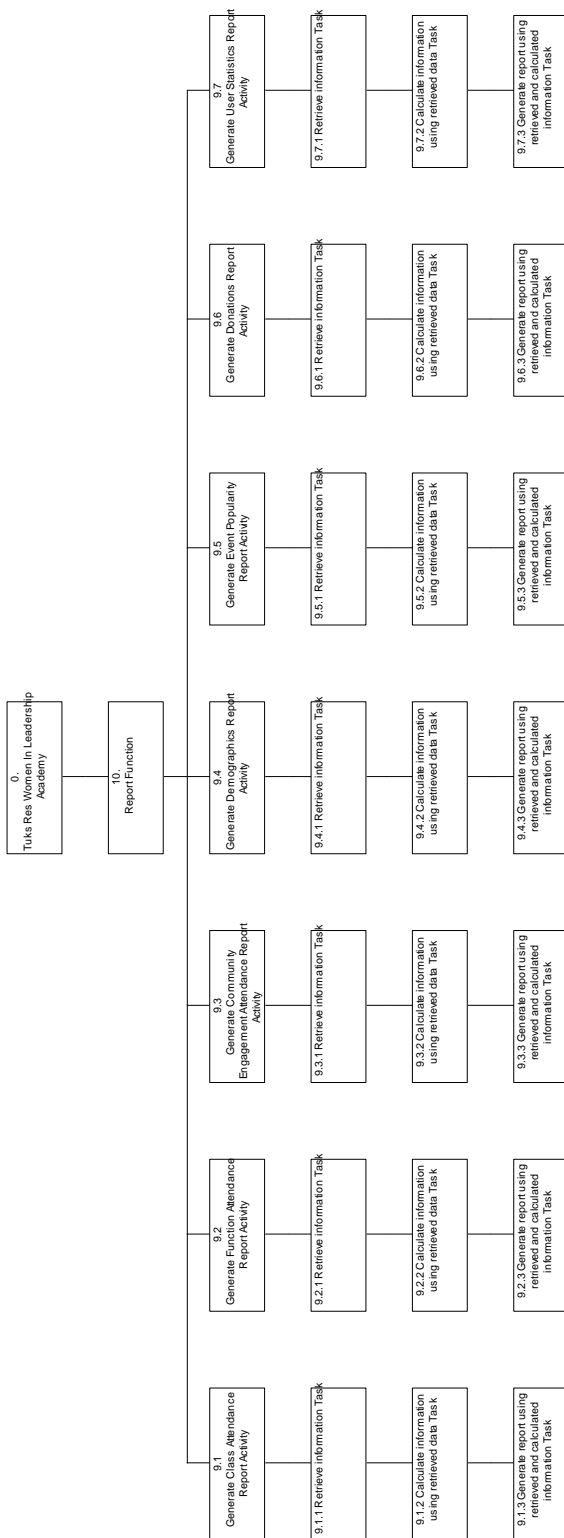


FIGURE 30- DECOMPOSITION DIAGRAM REPORT FUNCTION 9

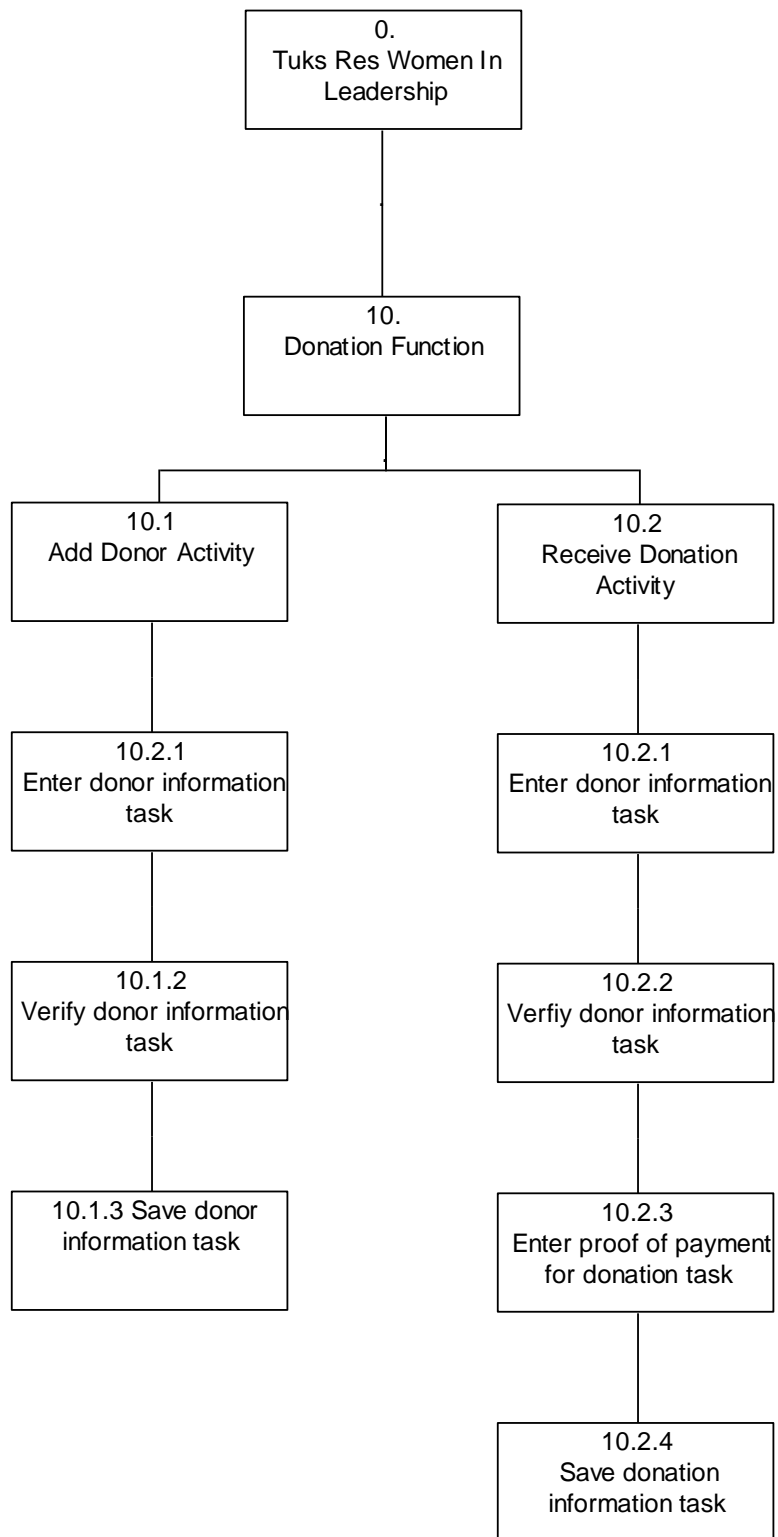


FIGURE 31- DECOMPOSITION DIAGRAM DONATION FUNCTION 10

## 3.4 Data Flow Diagrams

### 3.4.1 High-Level Data Flow Diagrams

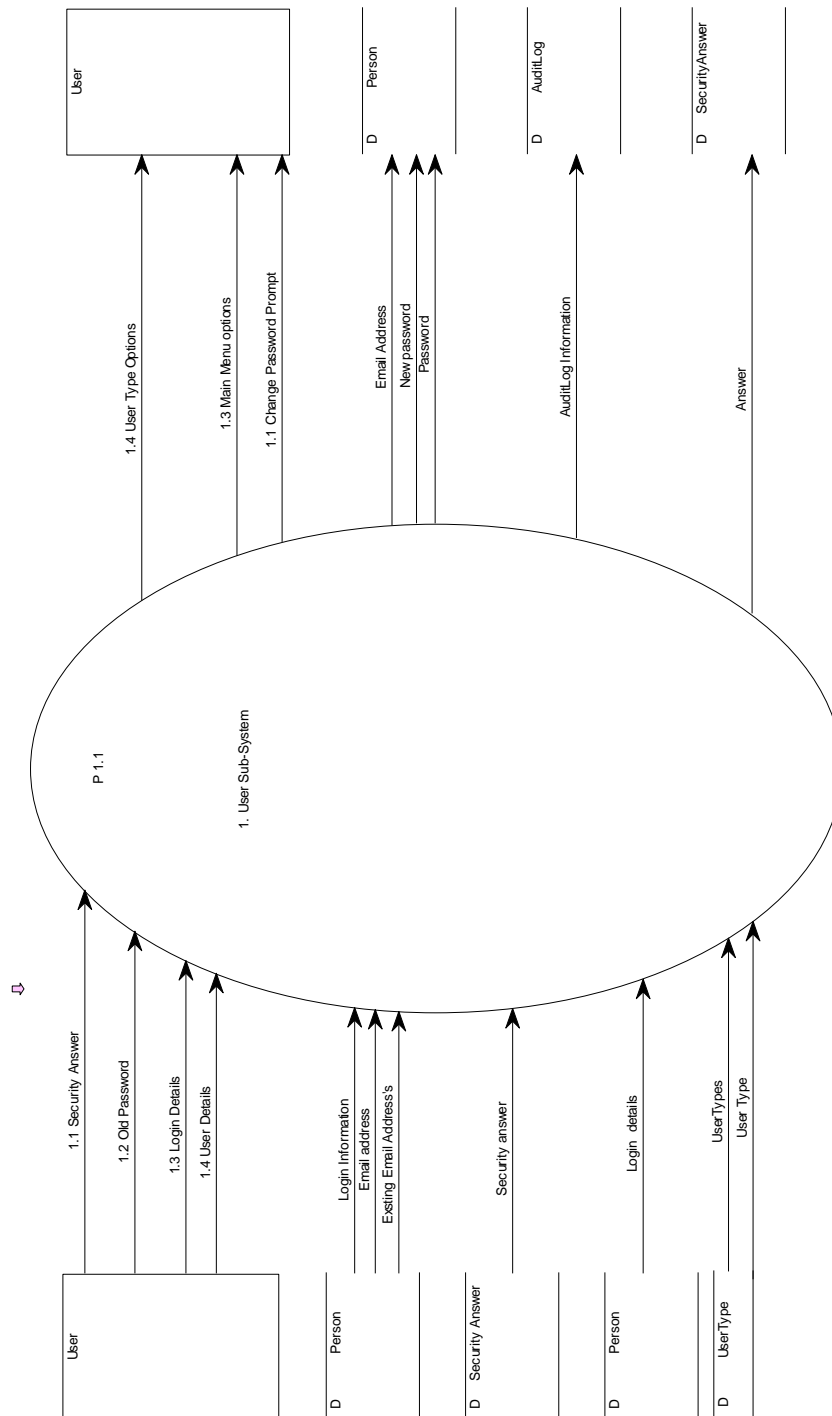


FIGURE 32- HIGH-LEVEL DATA FLOW DIAGRAM 1. USER SUBSYSTEM

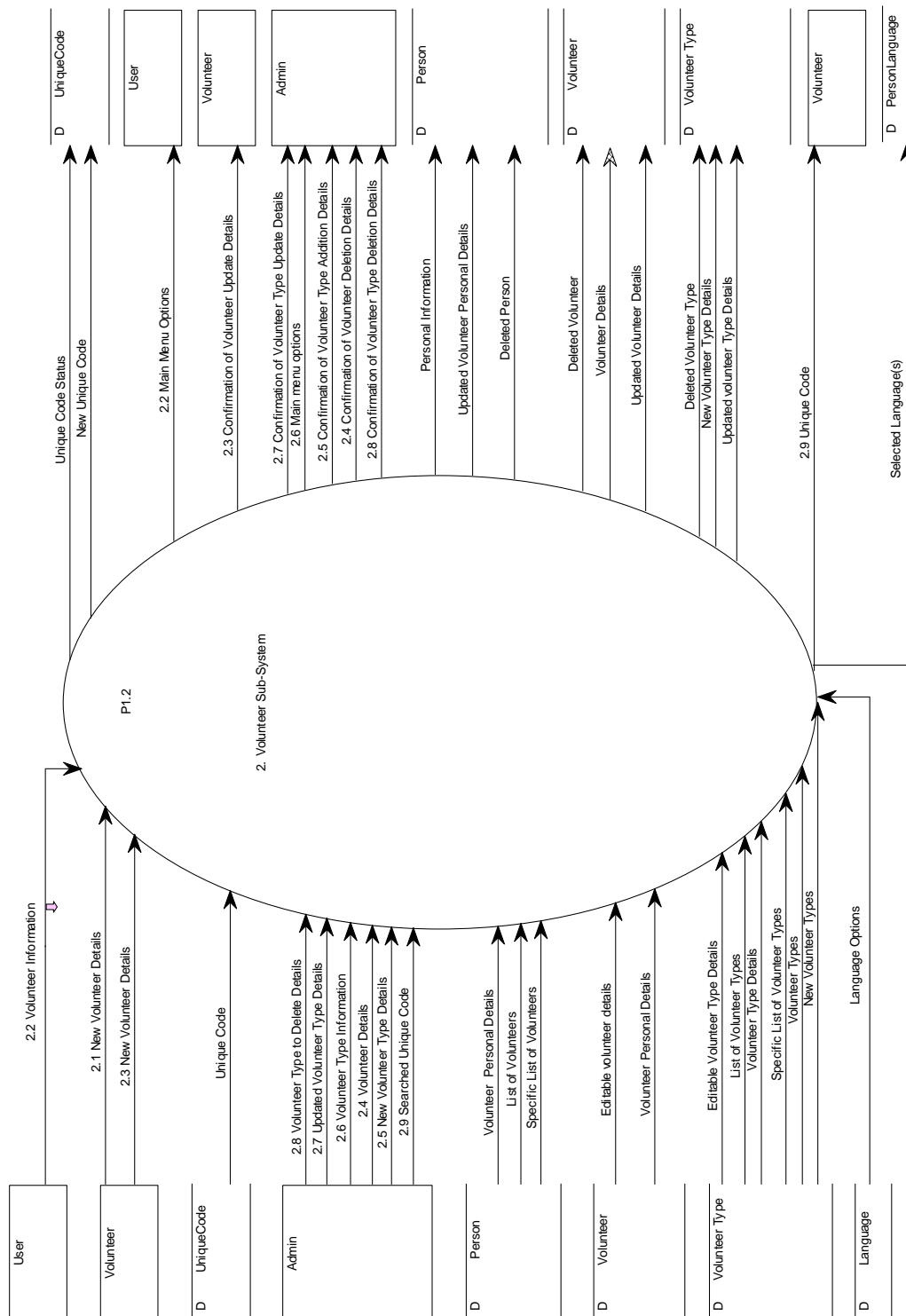


FIGURE 33- HIGH-LEVEL DATA FLOW DIAGRAM 2. VOLUNTEER SUBSYSTEM

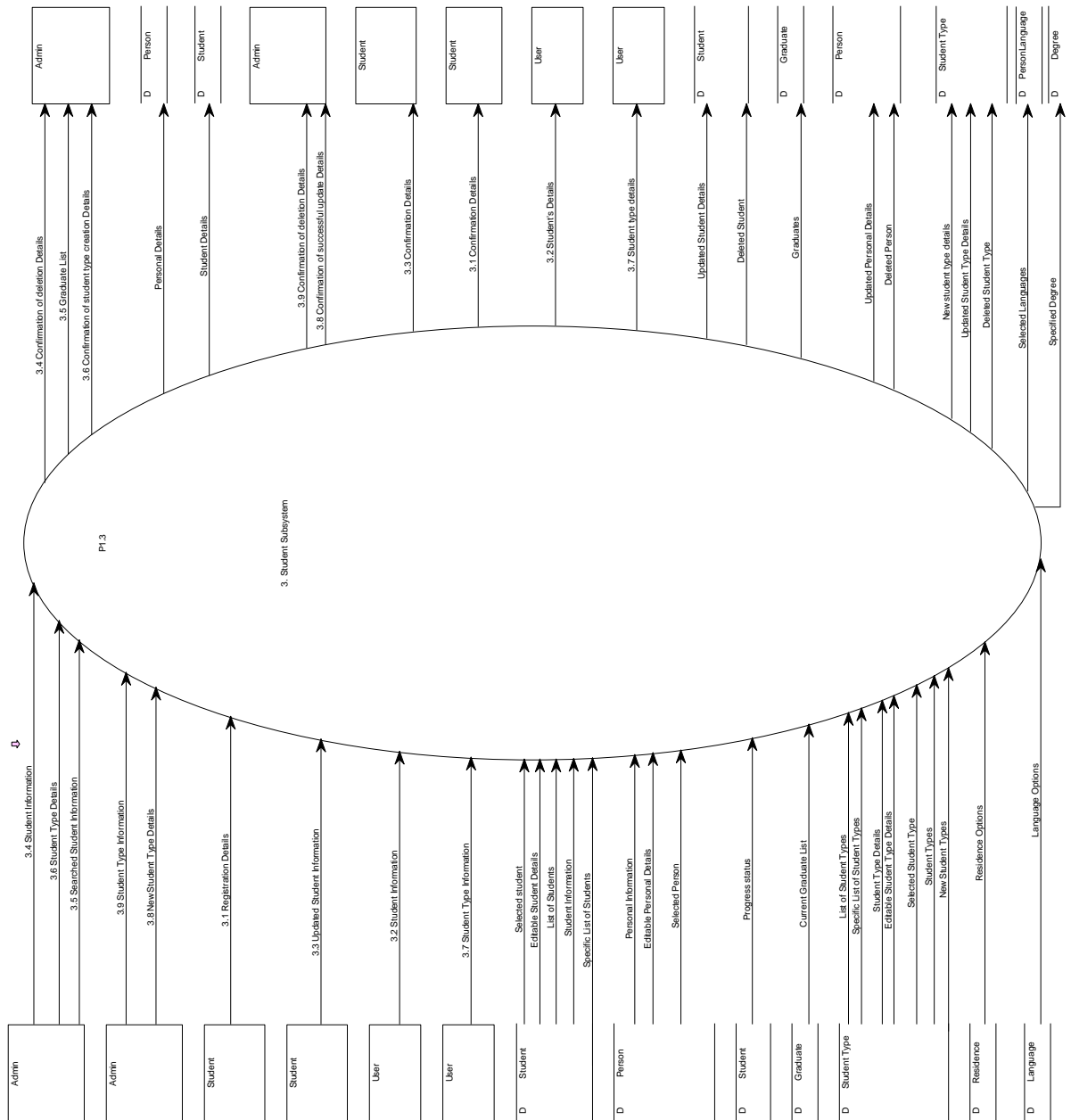


FIGURE 34- HIGH-LEVEL DATA FLOW DIAGRAM 3. STUDENT SUBSYSTEM

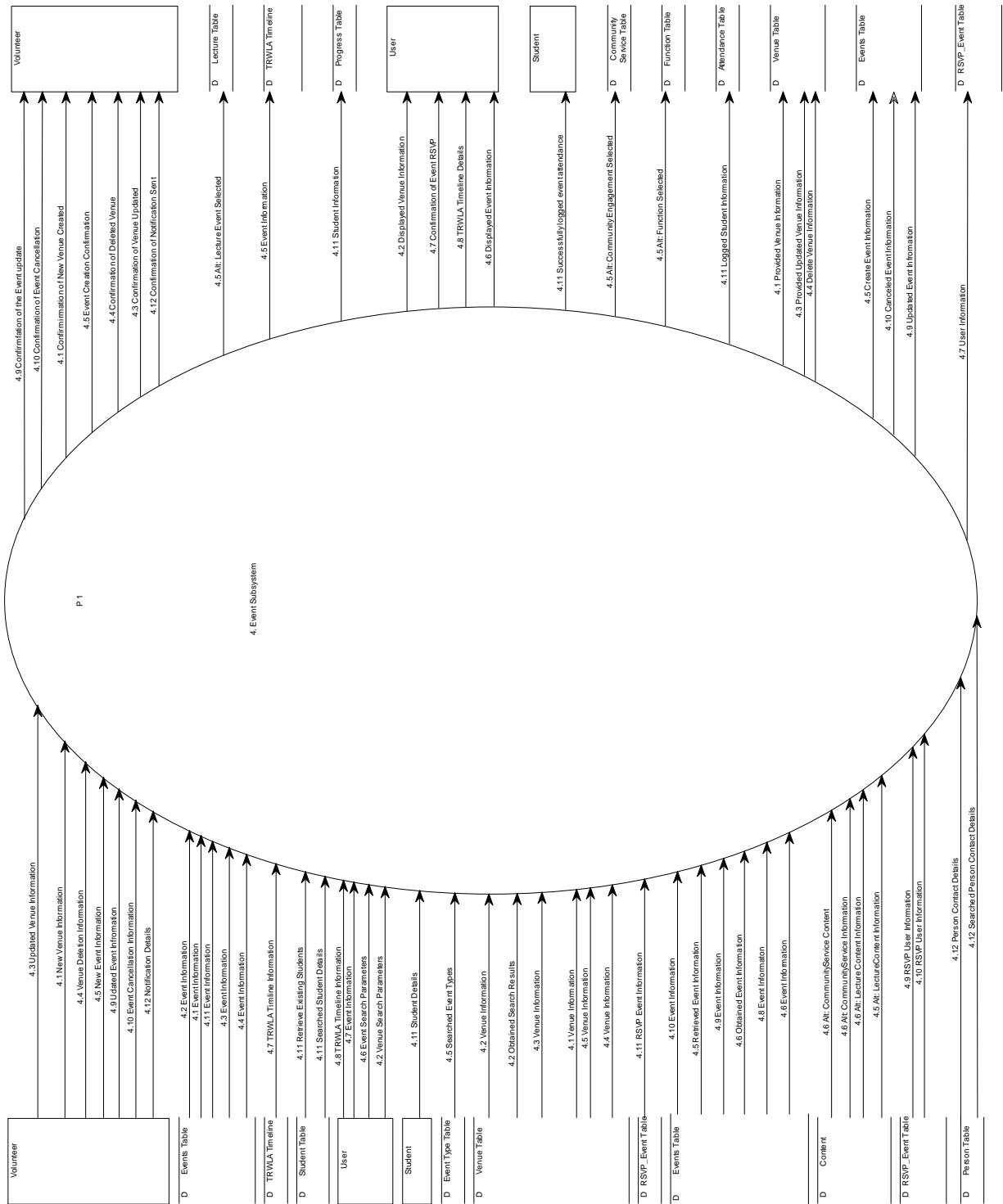


FIGURE 35- HIGH-LEVEL DATA FLOW DIAGRAM 4. EVENT SUBSYSTEM





FIGURE 36- HIGH-LEVEL DATA FLOW DIAGRAM 5. FUNCTION SUBSYSTEM

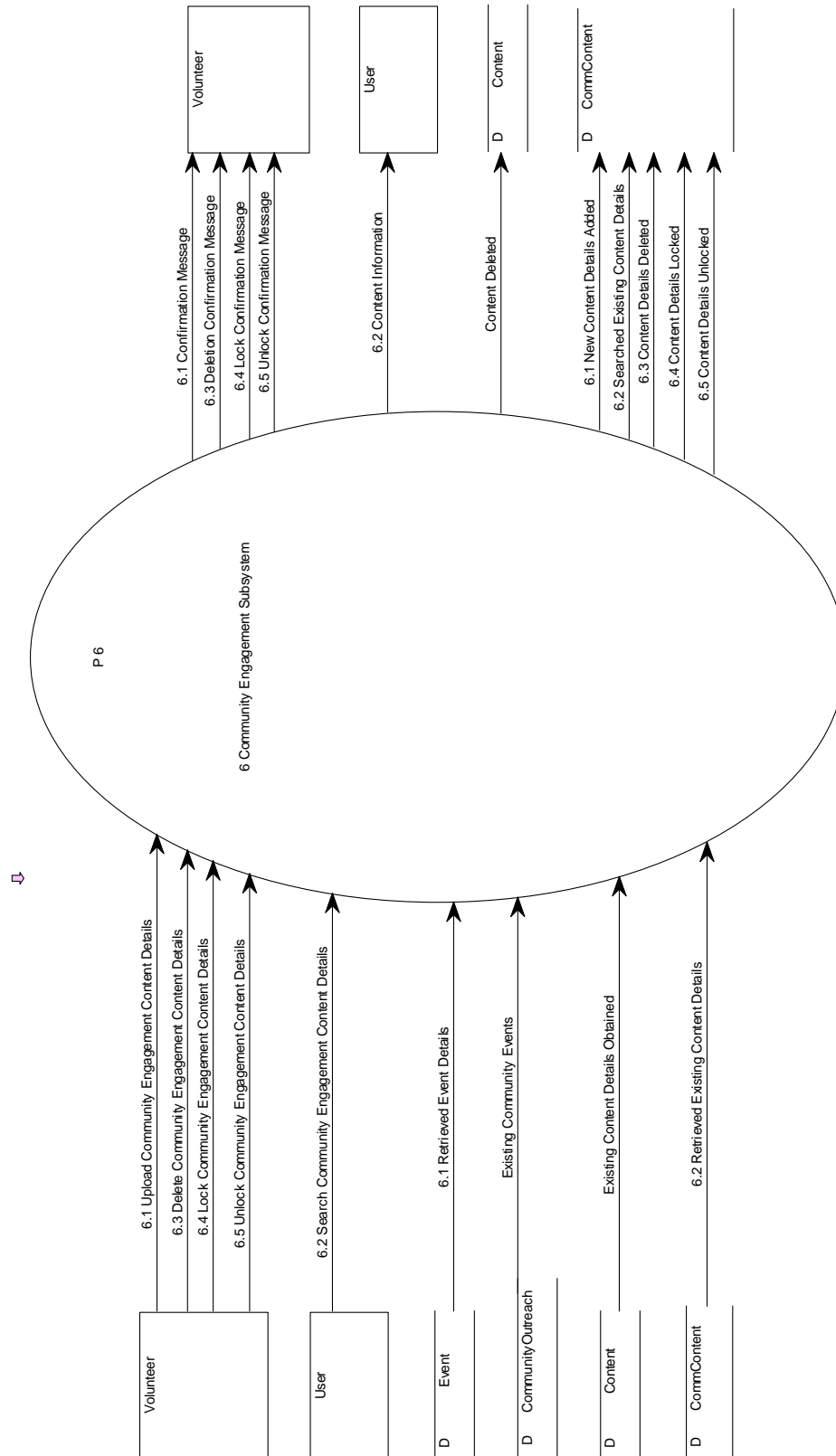


FIGURE 37- HIGH-LEVEL DATA FLOW DIAGRAM 6. COMMUNITY ENGAGEMENT

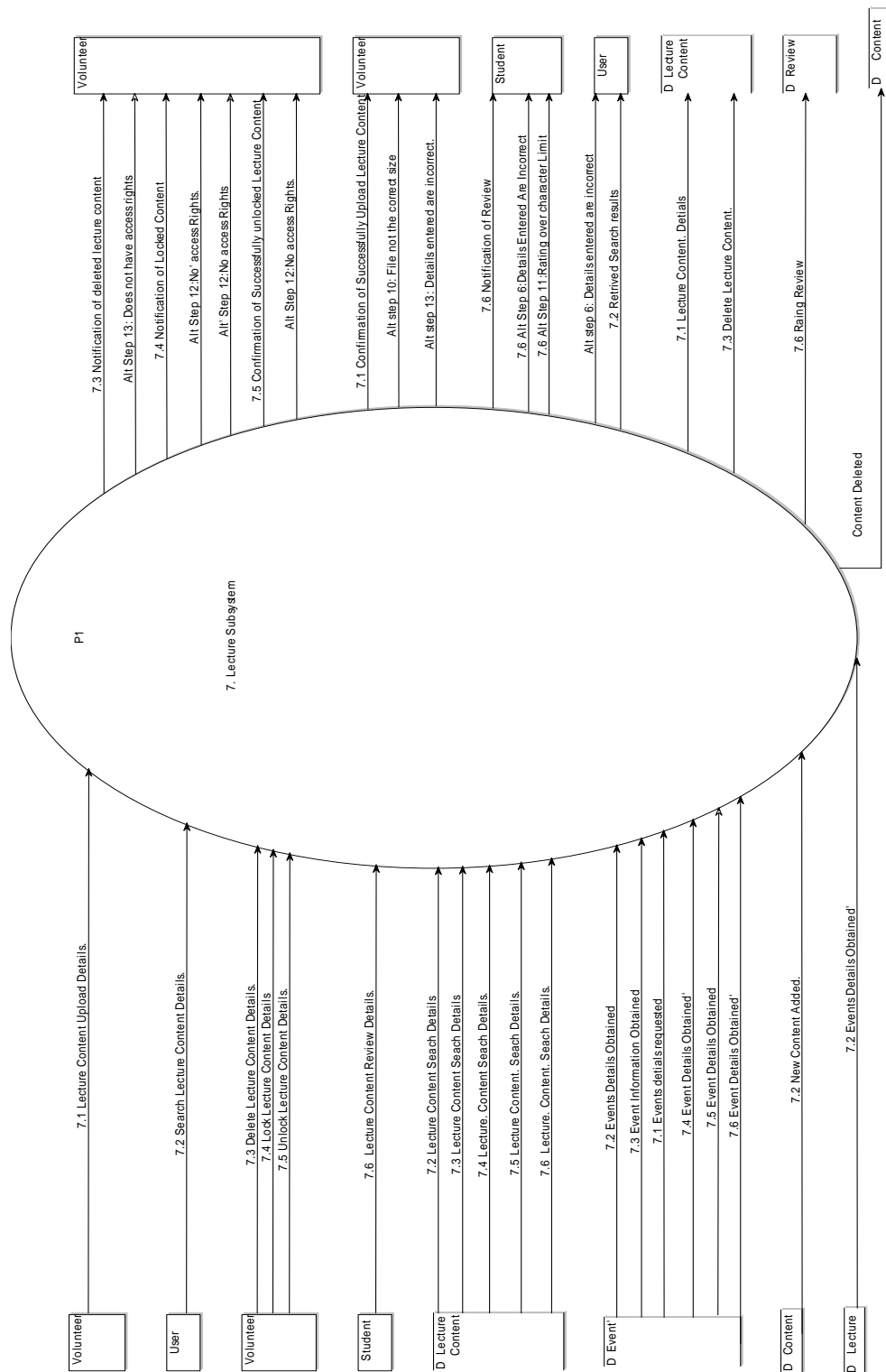


FIGURE 38- HIGH-LEVEL DATA FLOW DIAGRAM 7. LECTURE SUBSYSTEM

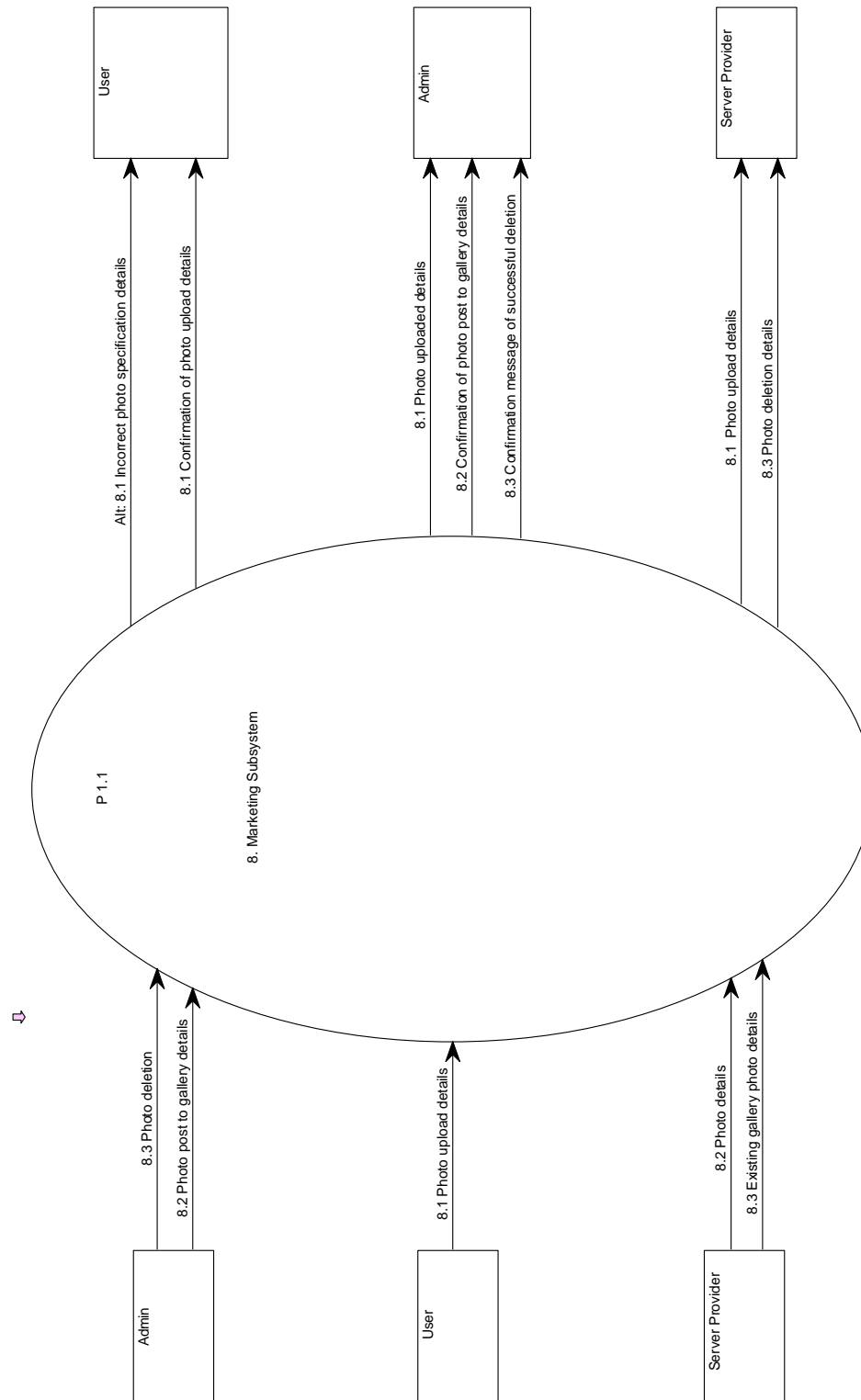


FIGURE 39- HIGH-LEVEL DATA FLOW DIAGRAM 8. MARKETING SUBSYSTEM

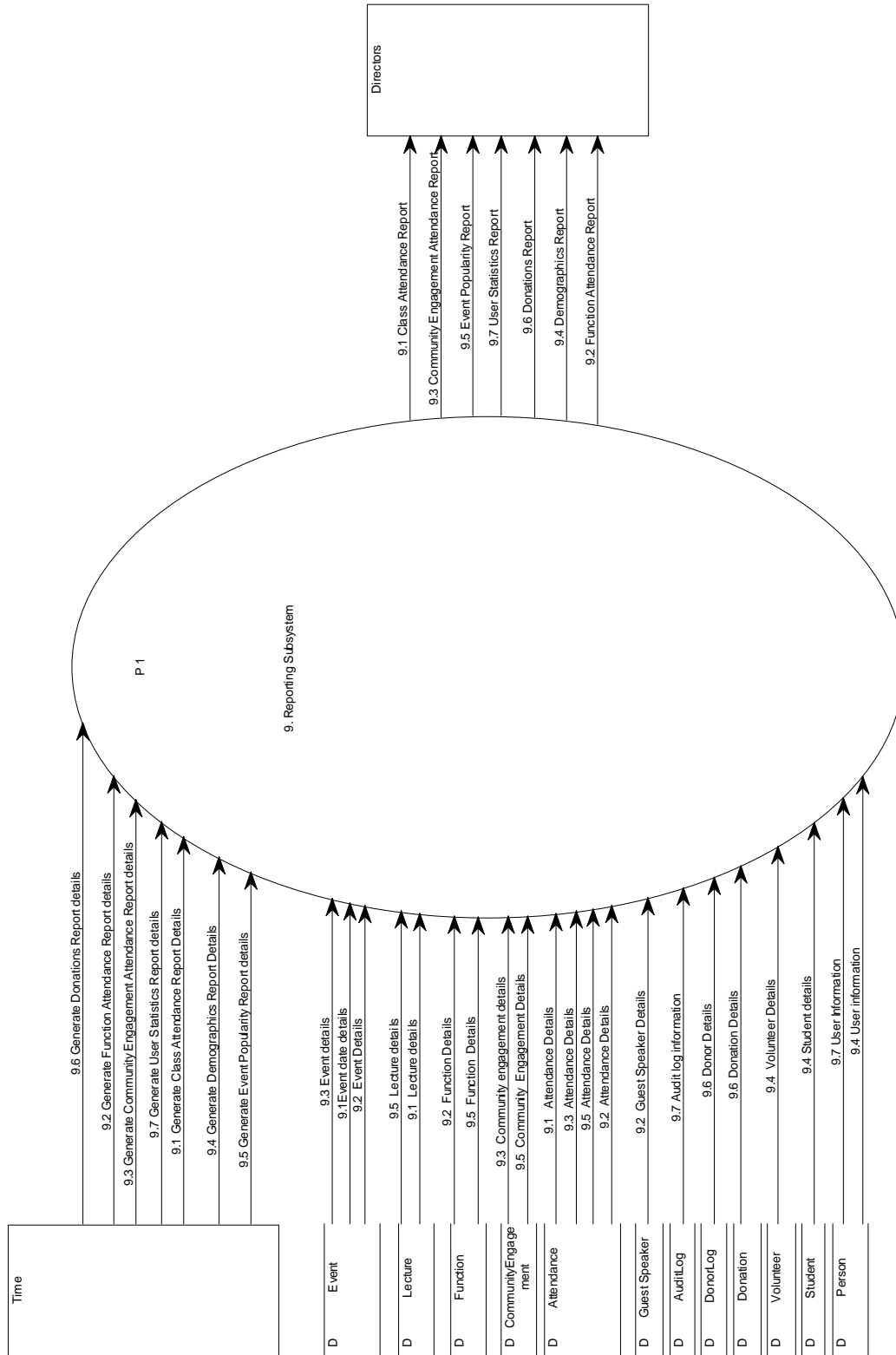


FIGURE 40- HIGH-LEVEL DATA FLOW DIAGRAM 9. REPORTING SUBSYSTEM

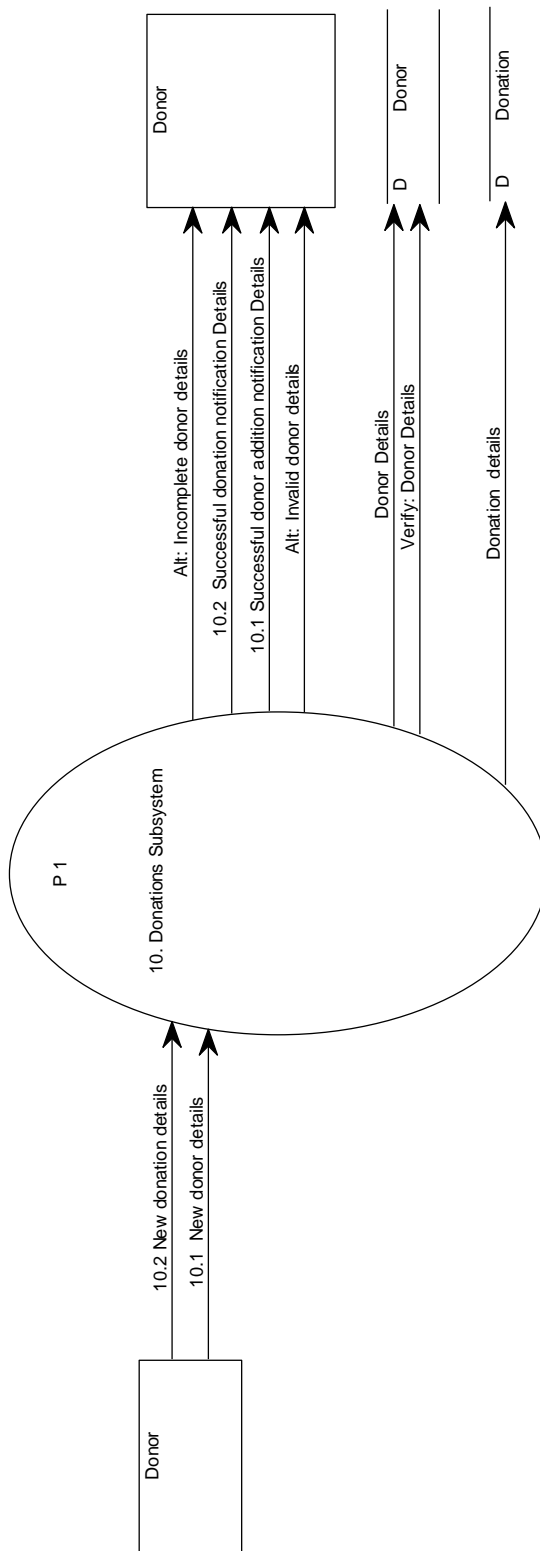


FIGURE 41- HIGH-LEVEL DATA FLOW DIAGRAM 10. DONATION SUBSYSTEM

### 3.4.2 Mid-Level Data Flow Diagrams

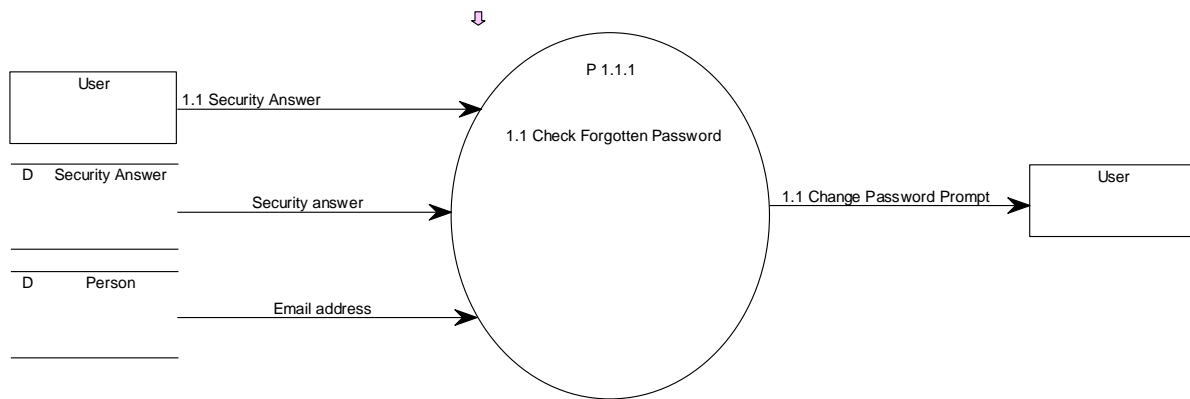


FIGURE 42- MID-LEVEL DATA FLOW DIAGRAM 1.1 CHECK FORGOTTEN PASSWORD

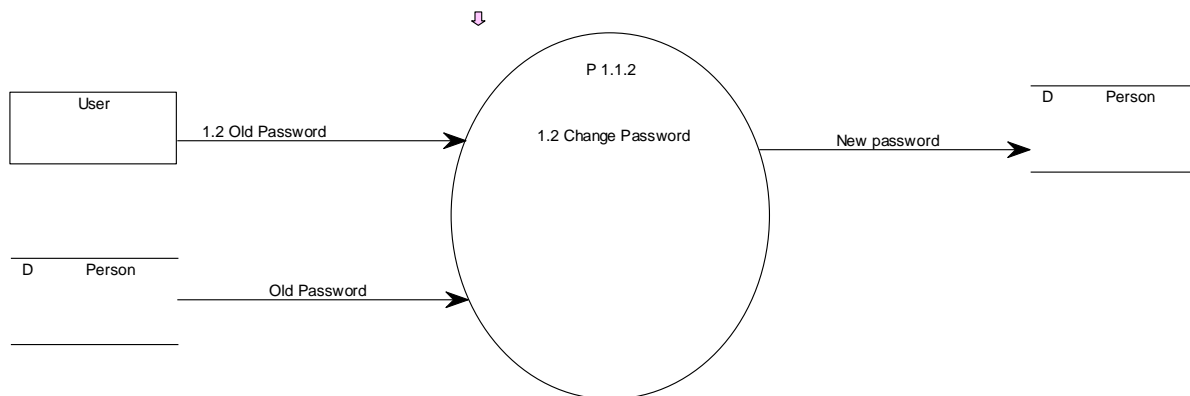


FIGURE 43- MID-LEVEL DATA FLOW DIAGRAM 1.2 FORGOTTEN PASSWORD

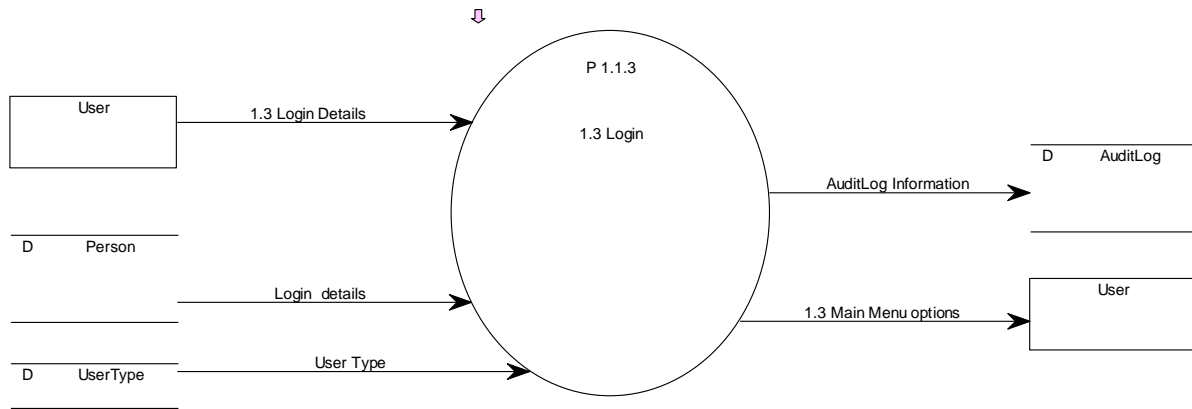


FIGURE 44- MID-LEVEL DATA FLOW DIAGRAM 1.3 LOGIN

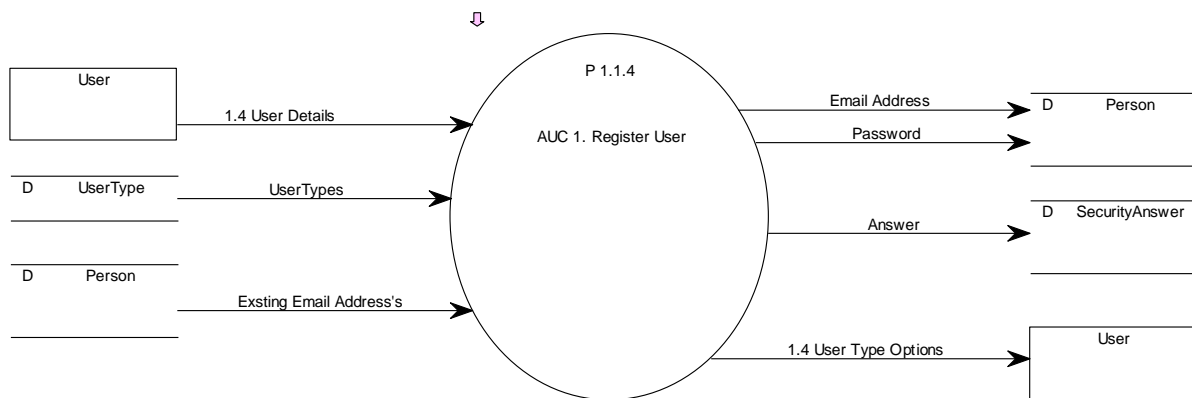


FIGURE 45- MID-LEVEL DATA FLOW DIAGRAM 1.4 ABSTRACT REGISTER USER



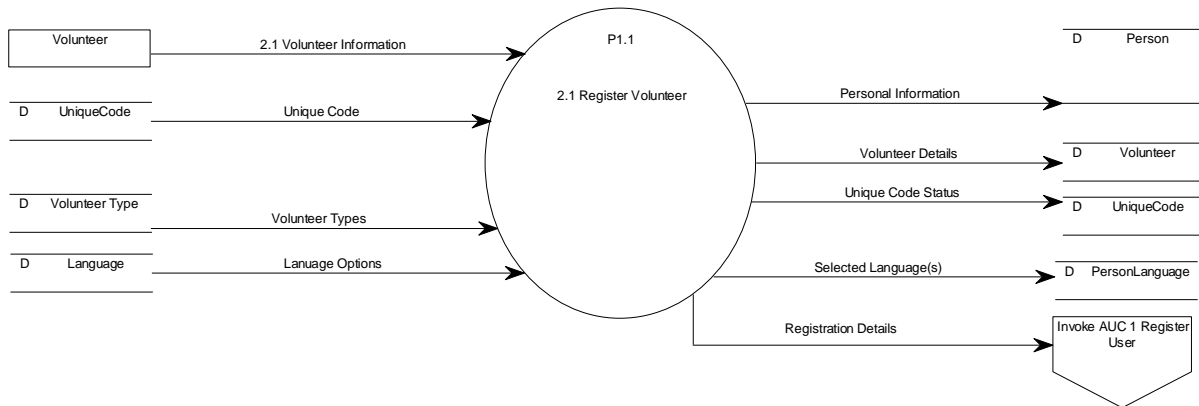


FIGURE 46- MID-LEVEL DATA FLOW DIAGRAM 2.1 REGISTER VOLUNTEER

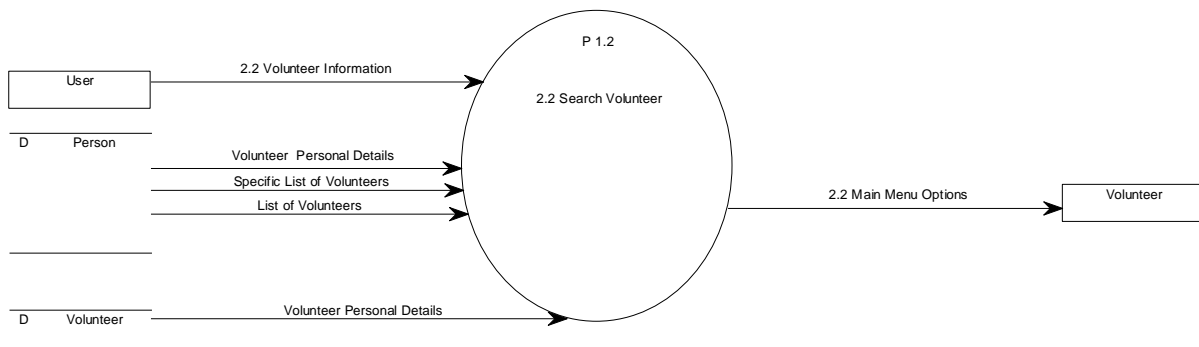


FIGURE 47- MID-LEVEL DATA FLOW DIAGRAM 2.2 SEARCH VOLUNTEER

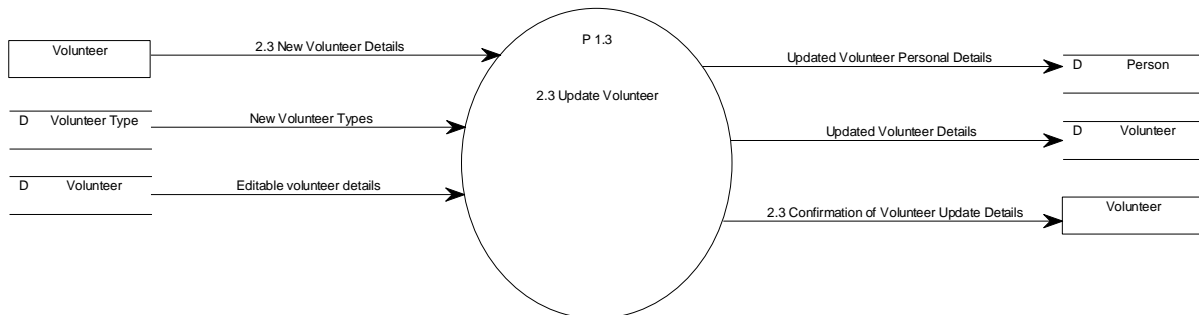


FIGURE 48- MID-LEVEL DATA FLOW DIAGRAM 2.3 UPDATE VOLUNTEER

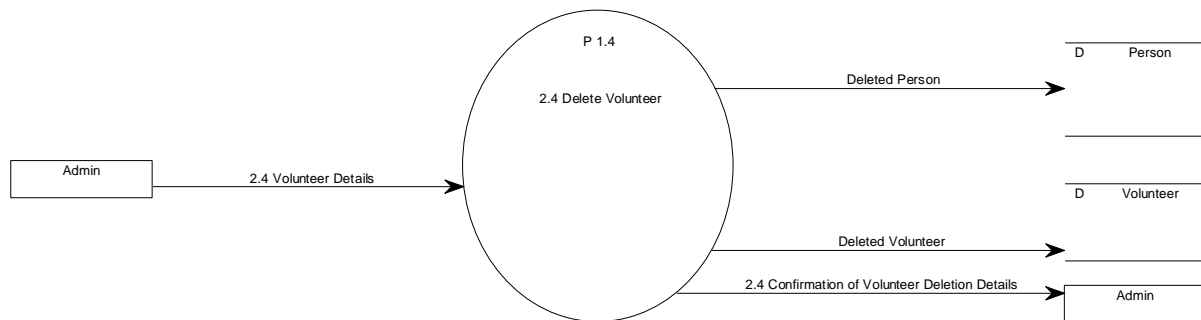


FIGURE 49- MID-LEVEL DATA FLOW DIAGRAM 2.4 DELETE VOLUNTEER

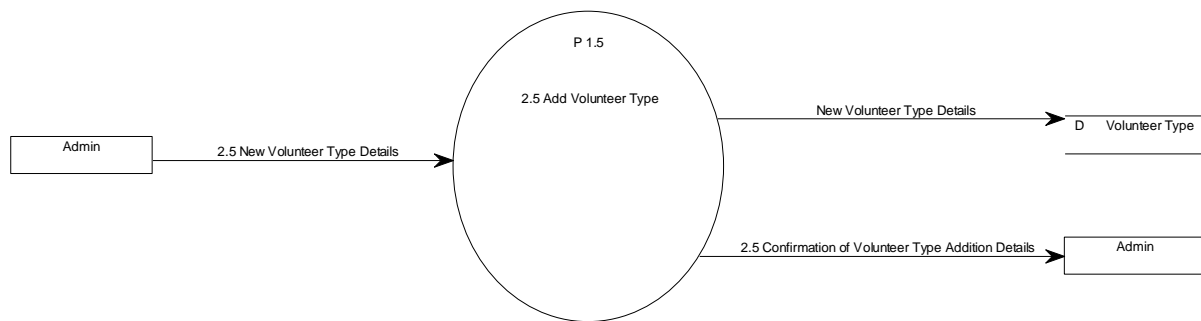


FIGURE 50- MID-LEVEL DATA FLOW DIAGRAM 2.5 ADD VOLUNTEER TYPE

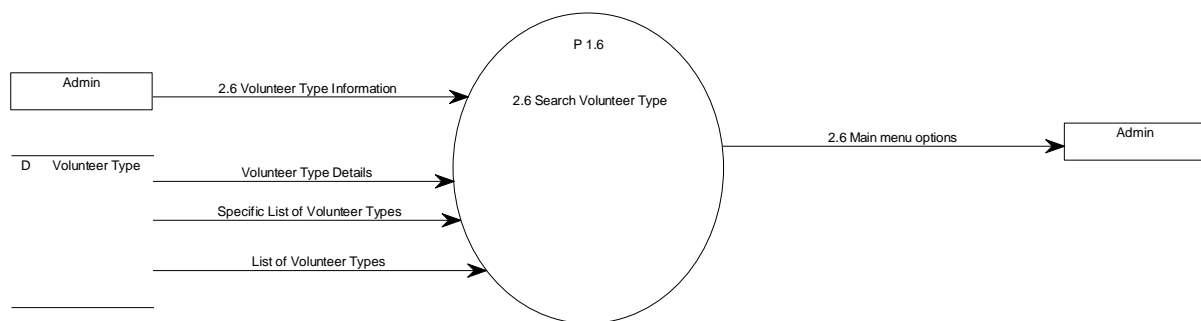


FIGURE 51- MID-LEVEL DATA FLOW DIAGRAM 2.6 SEARCH VOLUNTEER TYPE

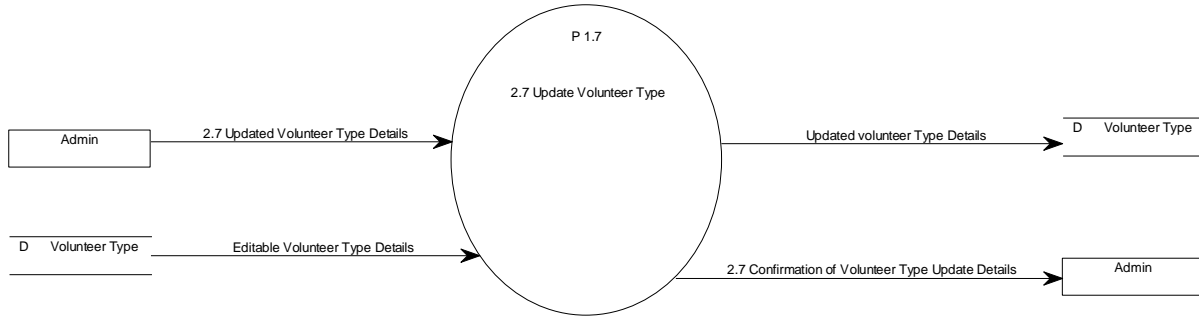


FIGURE 52- MID-LEVEL DATA FLOW DIAGRAM 2.7 UPDATE VOLUNTEER TYPE

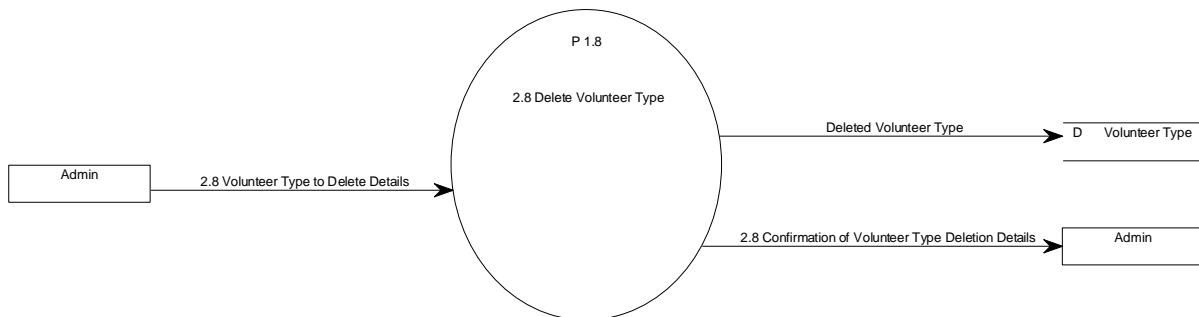


FIGURE 53- MID-LEVEL DATA FLOW DIAGRAM 2.8 DELETE VOLUNTEER TYPE

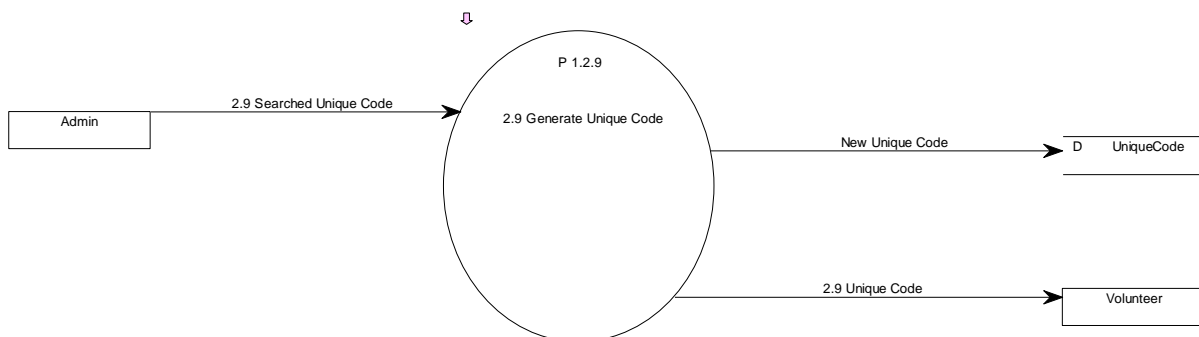


FIGURE 54- MID-LEVEL DATA FLOW DIAGRAM 2.9 GENERATE UNIQUE CODE

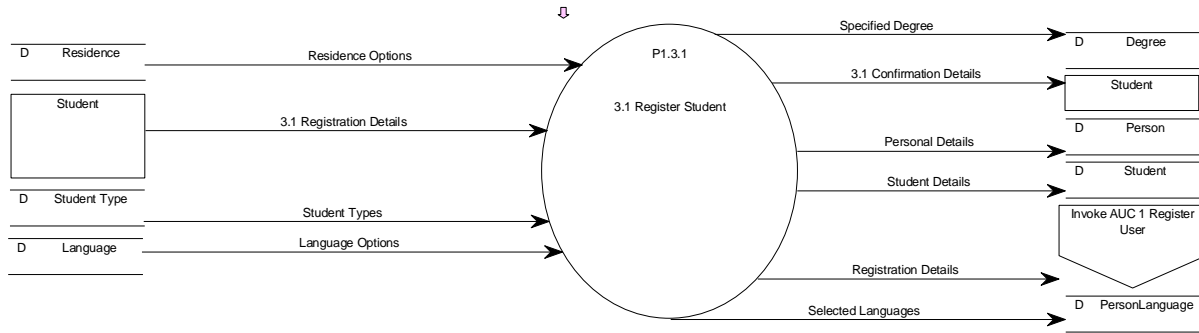


FIGURE 55- MID-LEVEL DATA FLOW DIAGRAM 3.1 REGISTER STUDENT

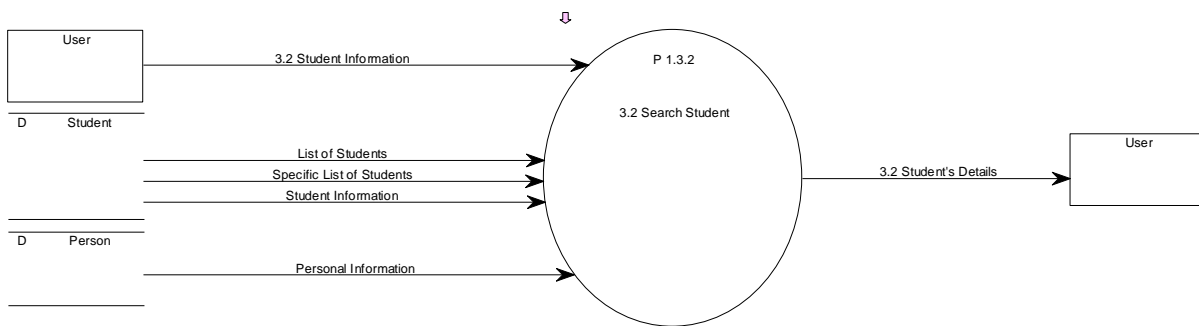


FIGURE 56- MID-LEVEL DATA FLOW DIAGRAM 3.2 SEARCH STUDENT

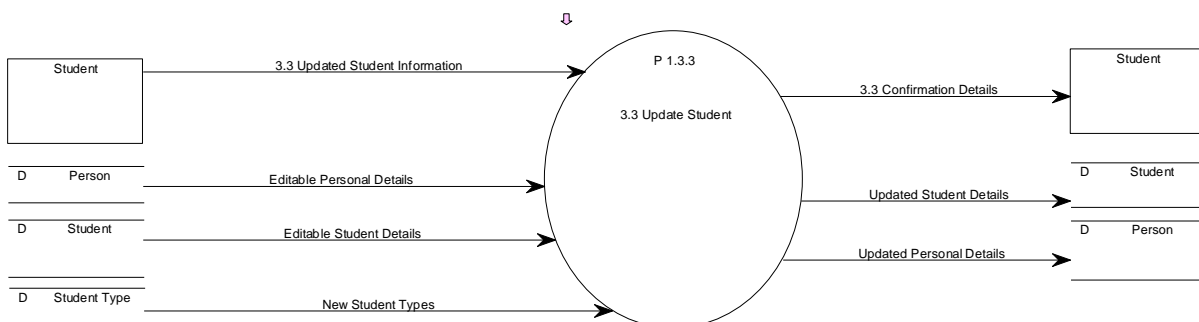


FIGURE 57- MID-LEVEL DATA FLOW DIAGRAM 3.3 UPDATE STUDENT

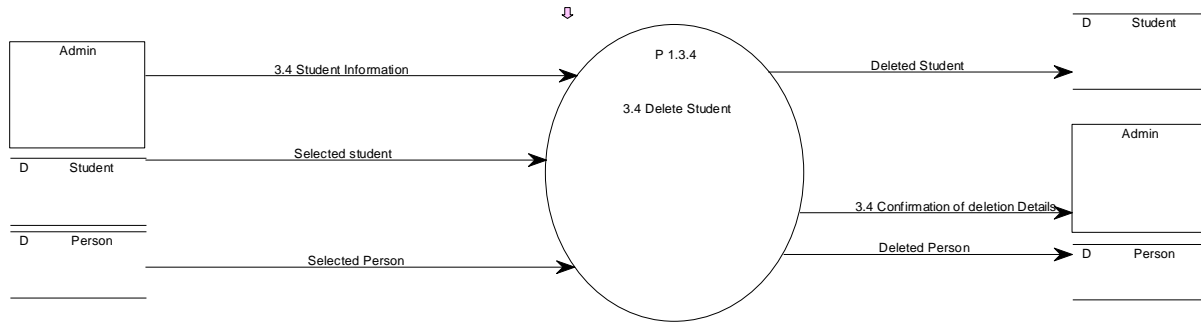


FIGURE 58- MID-LEVEL DATA FLOW DIAGRAM 3.4 DELETE STUDENT

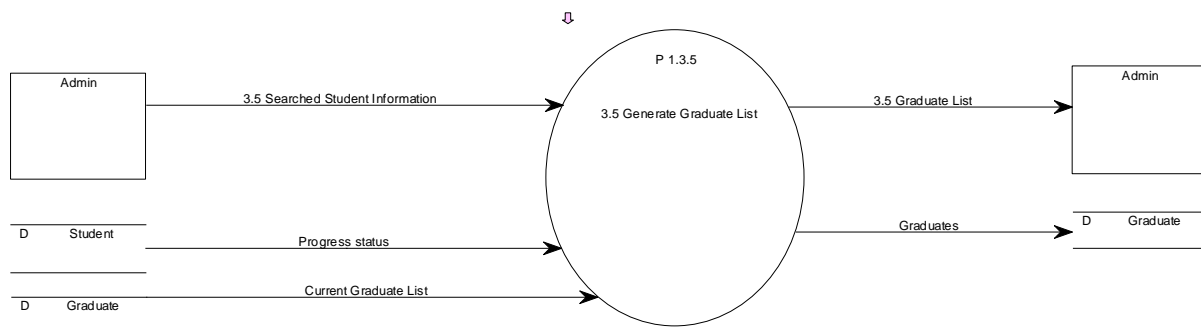


FIGURE 59- MID-LEVEL DATA FLOW DIAGRAM 3.5 GENERATE GRADUATE LIST

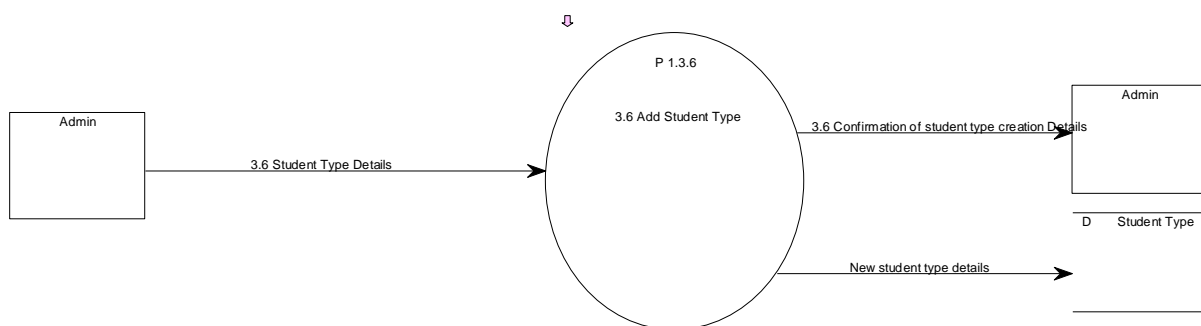


FIGURE 60- MID-LEVEL DATA FLOW DIAGRAM 3.6 ADD STUDENT TYPE

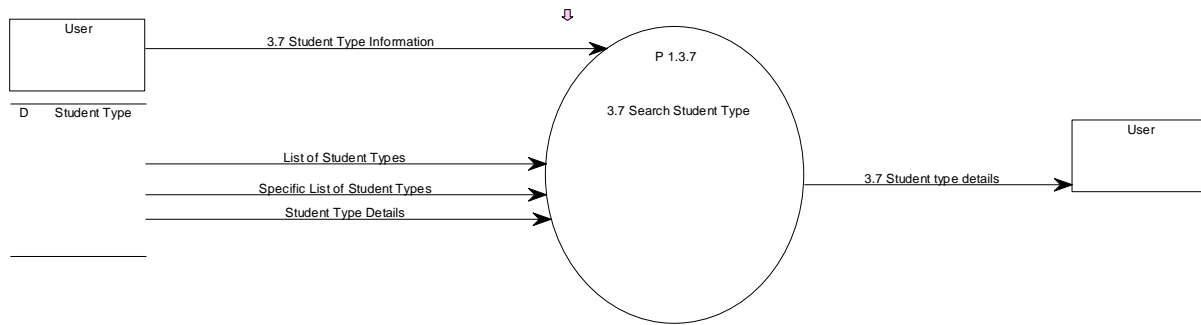


FIGURE 61- MID-LEVEL DATA FLOW DIAGRAM 3.7 SEARCH STUDENT TYPE

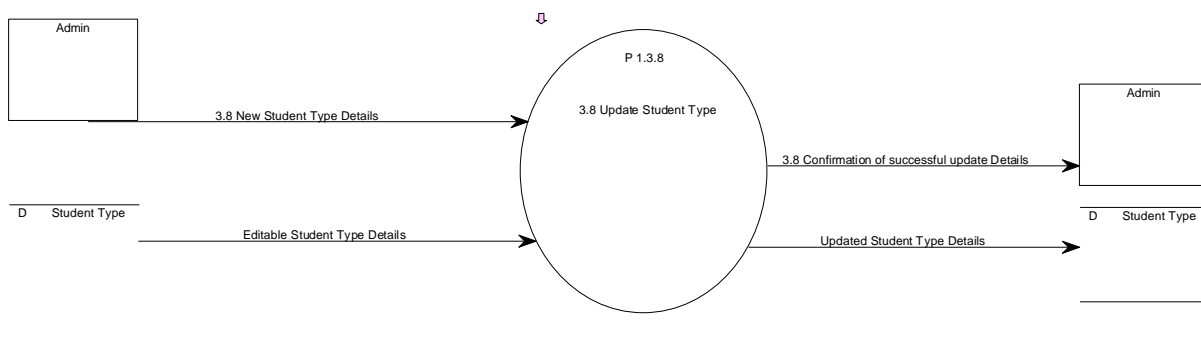


FIGURE 62- MID-LEVEL DATA FLOW DIAGRAM 3.8 UPDATE STUDENT TYPE

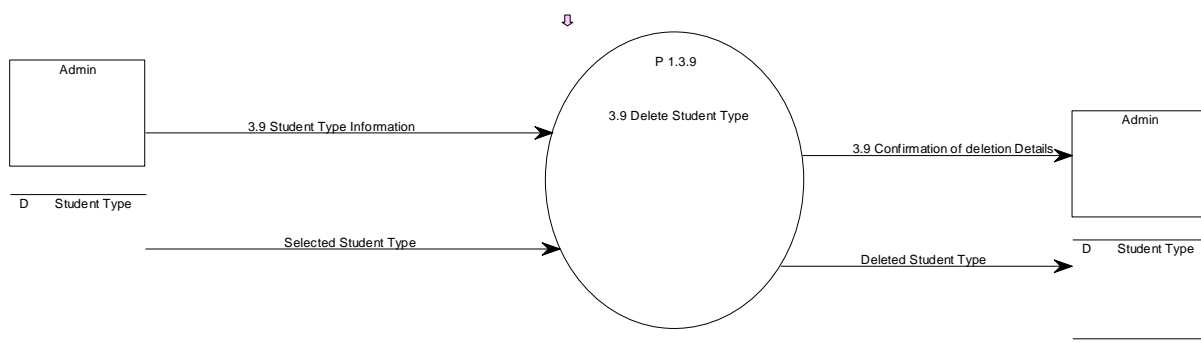


FIGURE 63- MID-LEVEL DATA FLOW DIAGRAM 3.9 DELETE STUDENT TYPE

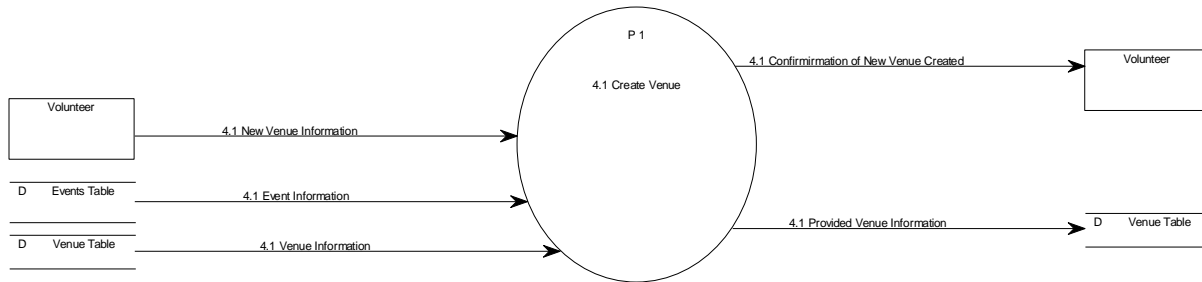


FIGURE 64- MID-LEVEL DATA FLOW DIAGRAM 4.1 CREATE VENUE

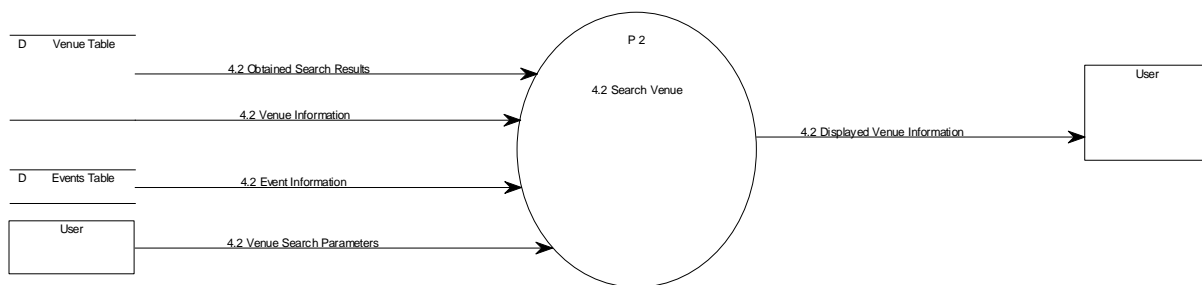


FIGURE 65- MID-LEVEL DATA FLOW DIAGRAM 4.2 SEARCH VENUE

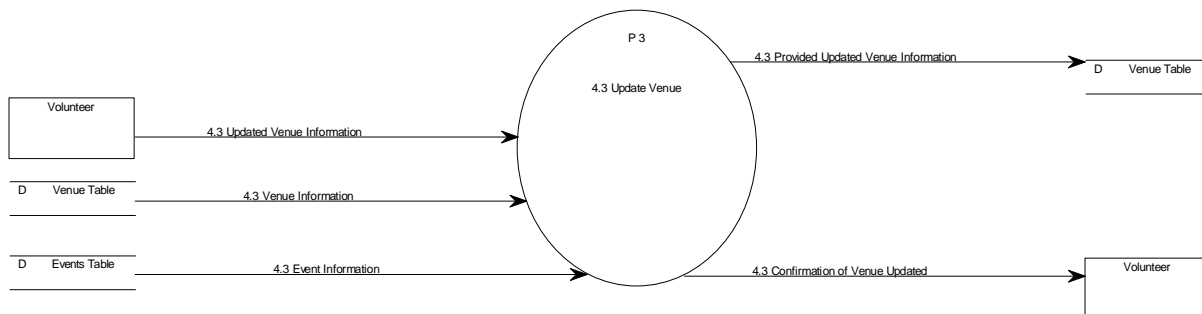


FIGURE 66- MID-LEVEL DATA FLOW DIAGRAM 4.3 UPDATE VENUE

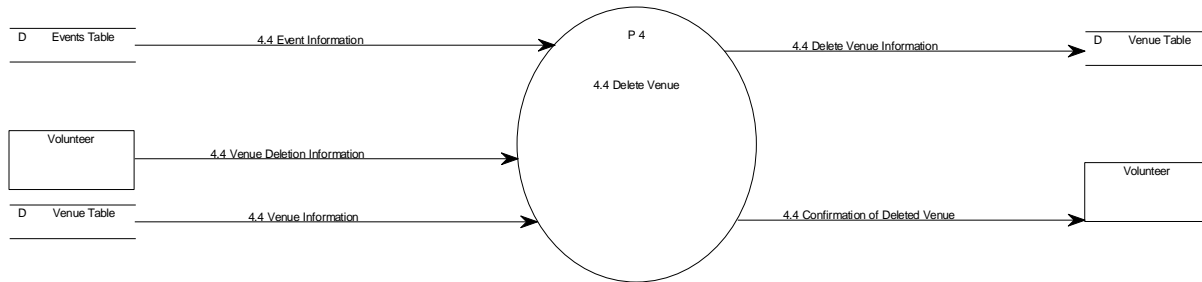


FIGURE 67- MID-LEVEL DATA FLOW DIAGRAM 4.4 DELETE VENUE

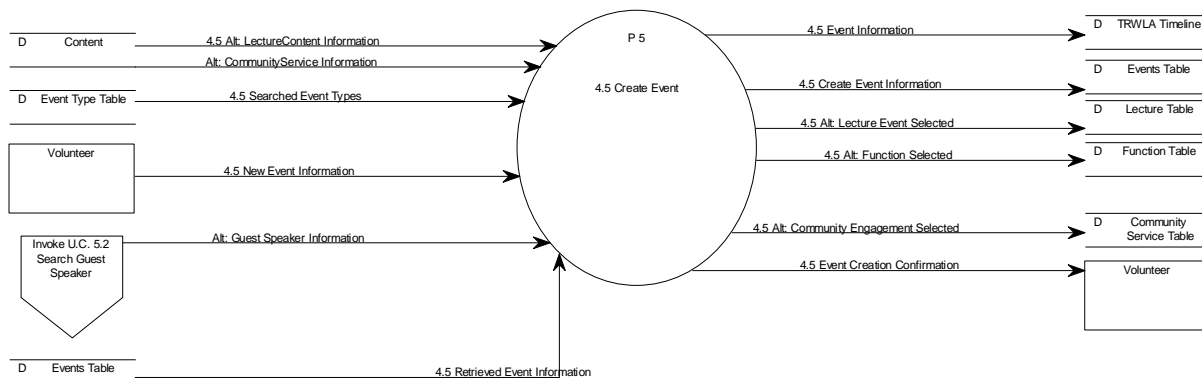


FIGURE 68- MID-LEVEL DATA FLOW DIAGRAM 4.5 CREATE EVENT

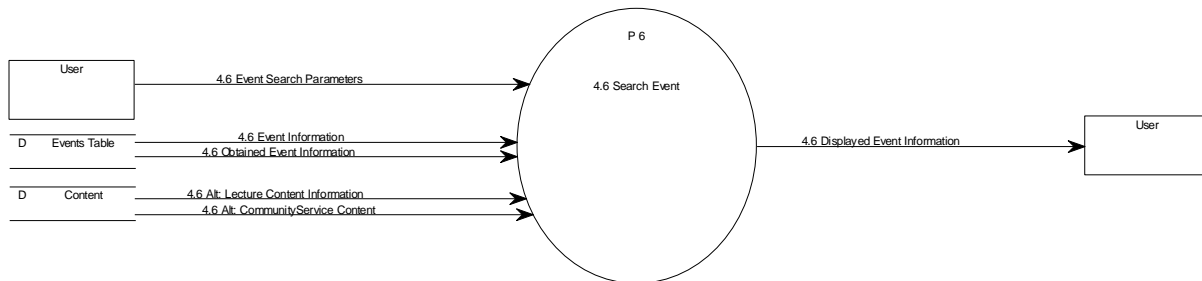


FIGURE 69- MID-LEVEL DATA FLOW DIAGRAM 4.6 SEARCH EVENT



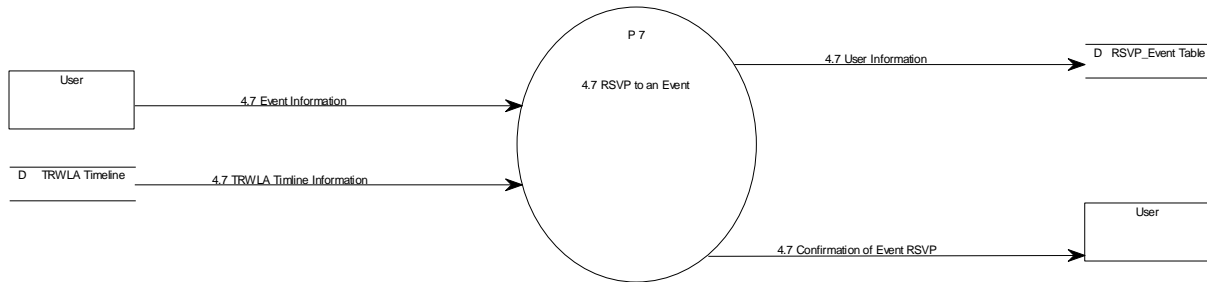


FIGURE 70- MID-LEVEL DATA FLOW DIAGRAM 4.7 RSVP TO AN EVENT

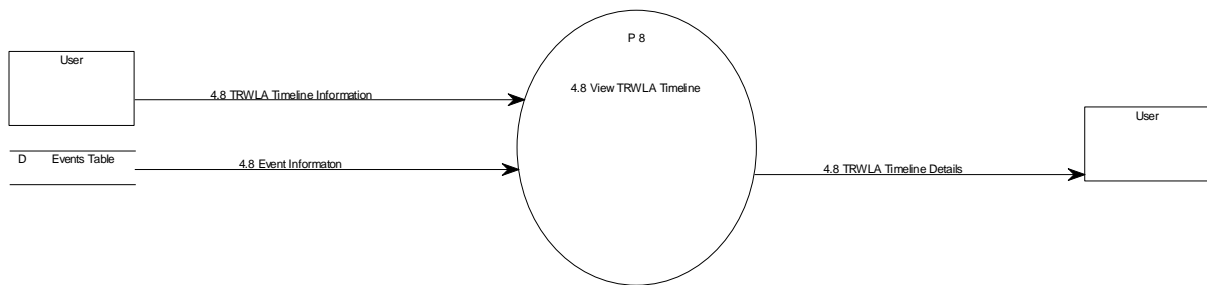


FIGURE 71- MID-LEVEL DATA FLOW DIAGRAM 4.8 VIEW TRWLA TIMELINE

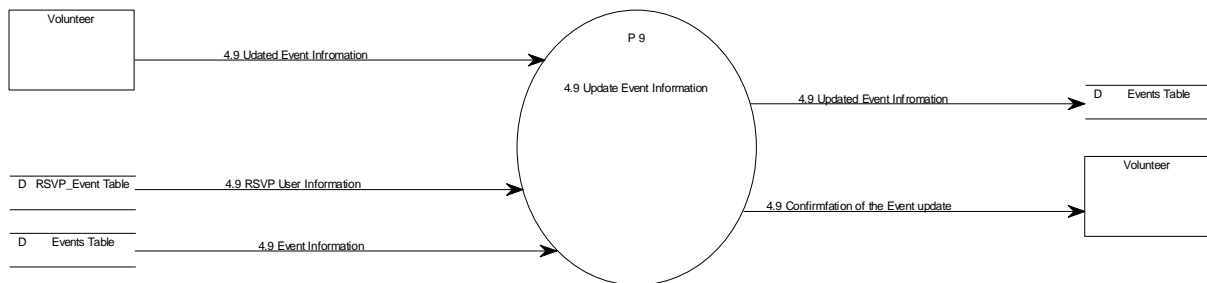


FIGURE 72- MID-LEVEL DATA FLOW DIAGRAM 4.9 UPDATE EVENT INFORMATION

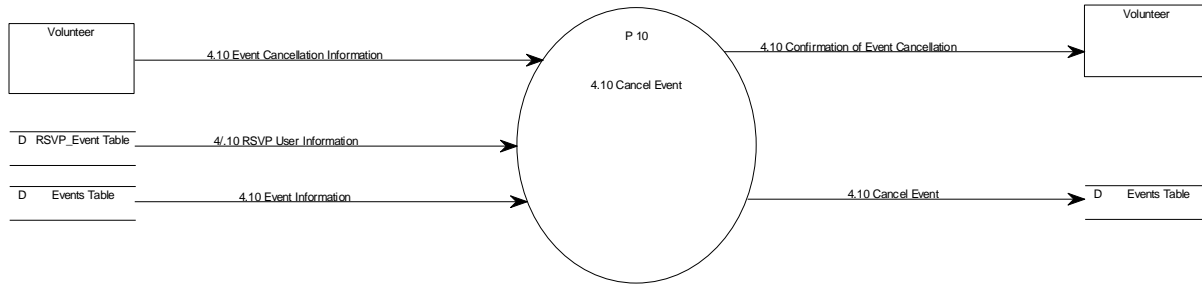


FIGURE 73- MID-LEVEL DATA FLOW DIAGRAM 4.10 CANCEL EVENT

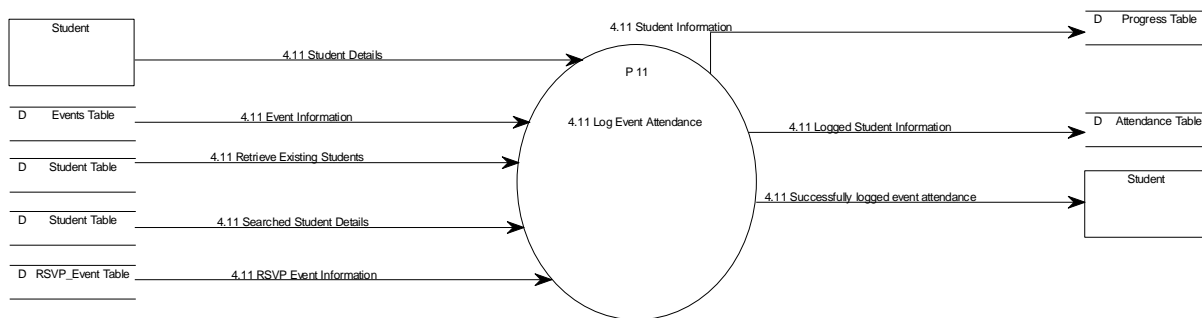


FIGURE 74- MID-LEVEL DATA FLOW DIAGRAM 4.11 LOG EVENT ATTENDANCE

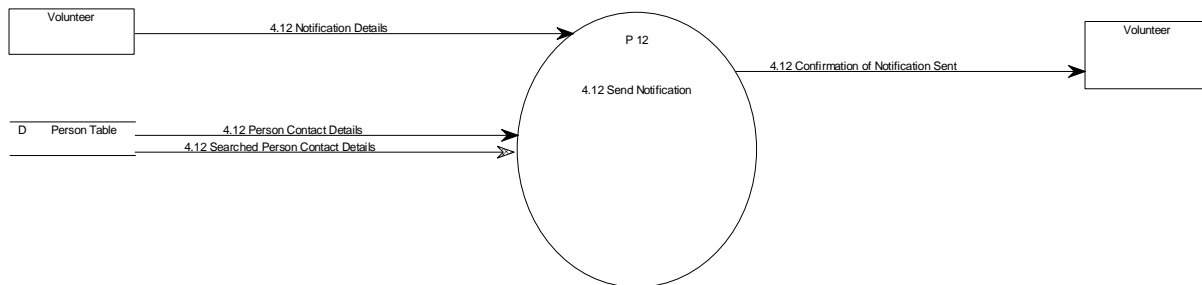


FIGURE 75- MID-LEVEL DATA FLOW DIAGRAM 4.12 SEND NOTIFICATION

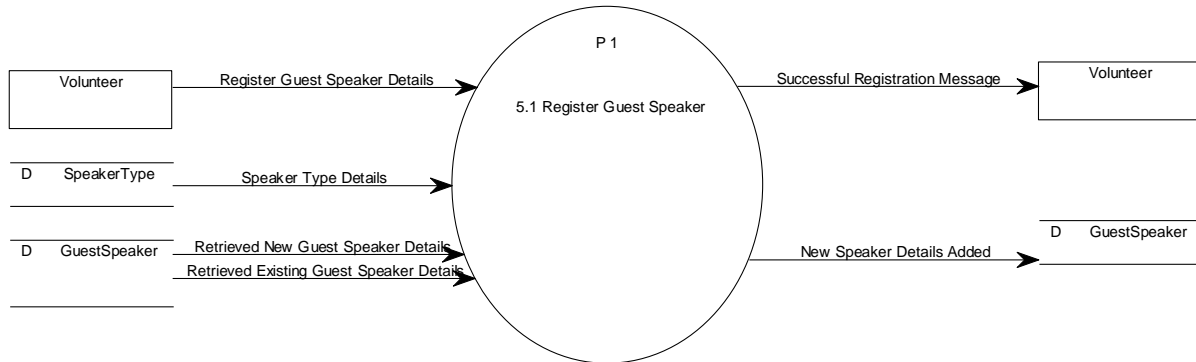


FIGURE 76- MID-LEVEL DATA FLOW DIAGRAM 5.1 REGISTER GUEST SPEAKER

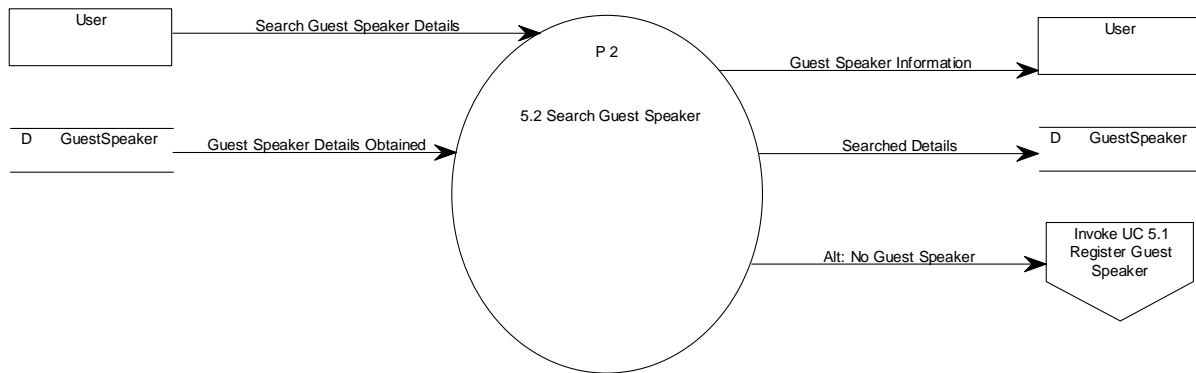


FIGURE 77- MID-LEVEL DATA FLOW DIAGRAM 5.2 SEARCH GUEST SPEAKER

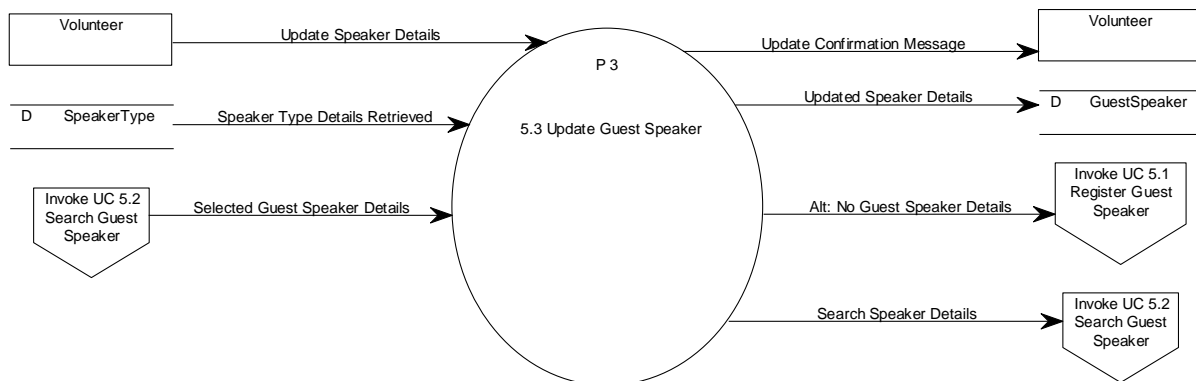


FIGURE 78- MID-LEVEL DATA FLOW DIAGRAM 5.3 UPDATE GUEST SPEAKER

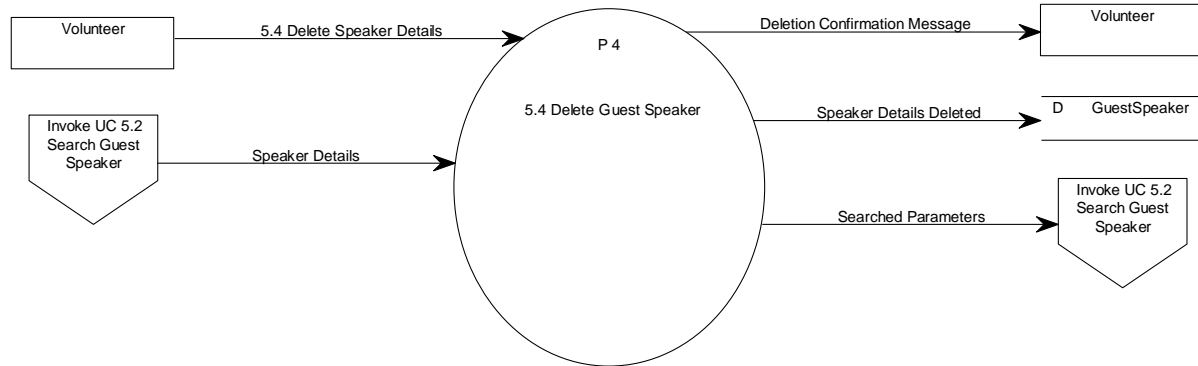


FIGURE 79- MID-LEVEL DATA FLOW DIAGRAM 5.4 DELETE GUEST SPEAKER

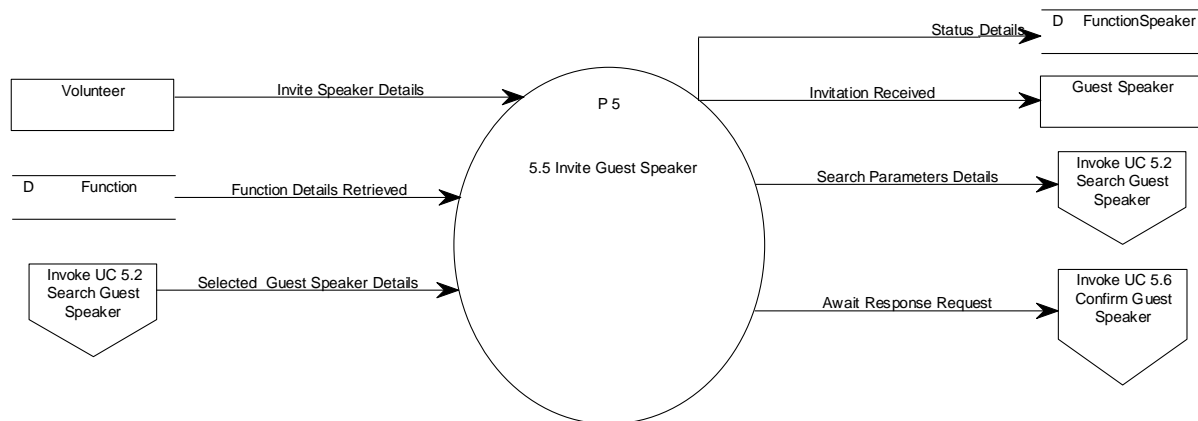


FIGURE 80- MID-LEVEL DATA FLOW DIAGRAM 5.5 INVITE GUEST SPEAKER

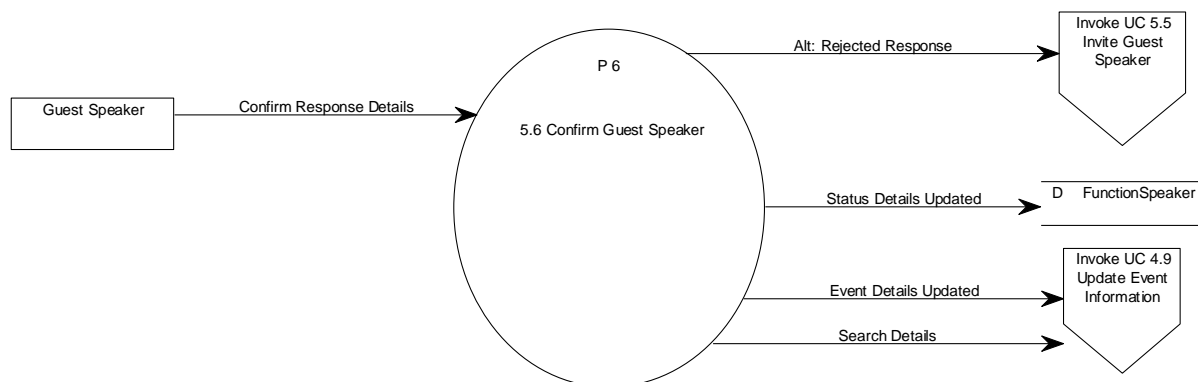


FIGURE 81- MID-LEVEL DATA FLOW DIAGRAM 5.6 CONFIRM GUEST SPEAKER

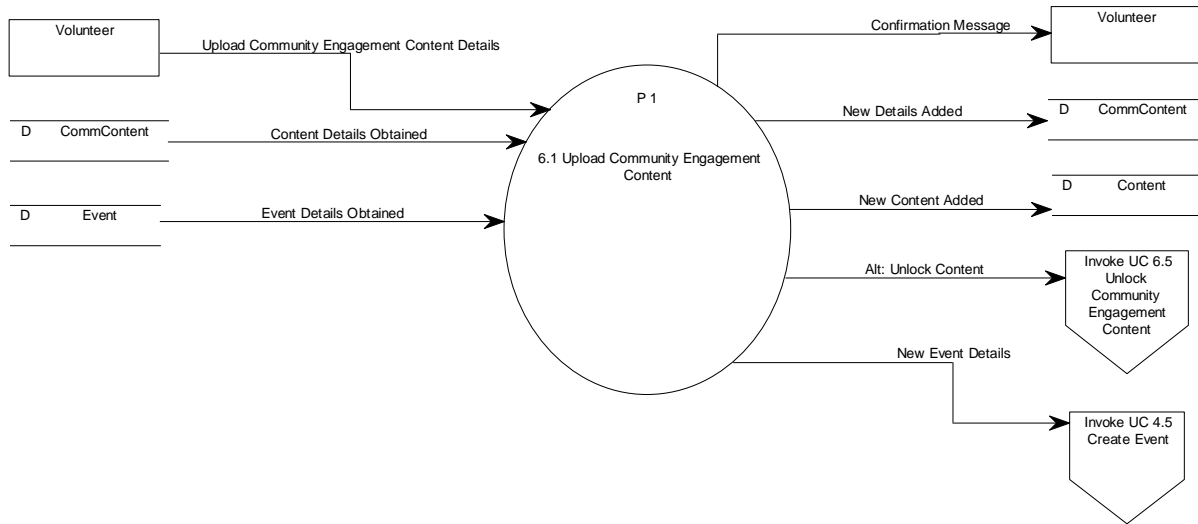


FIGURE 82- MID-LEVEL DATA FLOW DIAGRAM 6.1 UPLOAD COMMUNITY ENGAGEMENT CONTENT

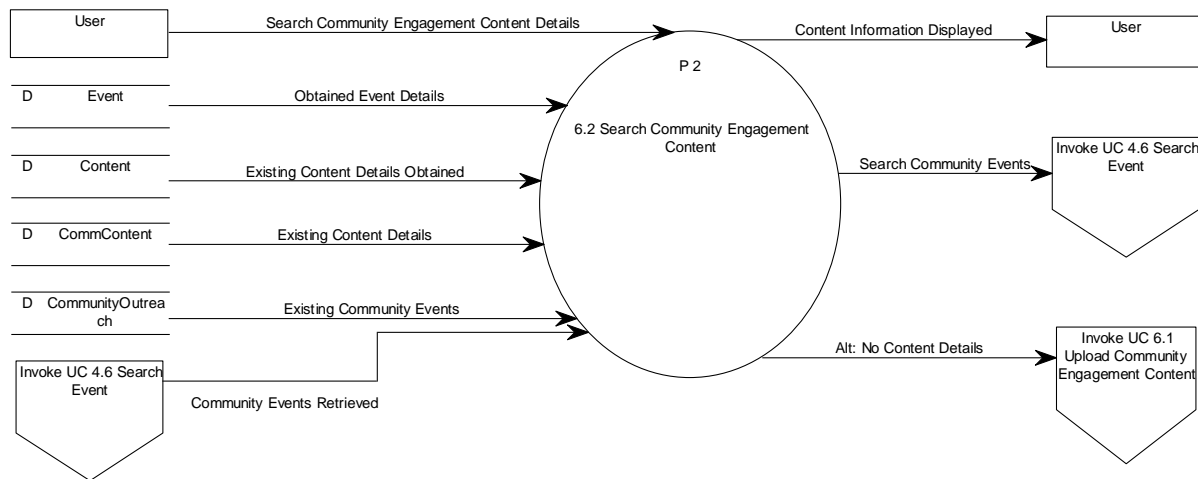


FIGURE 83- MID-LEVEL DATA FLOW DIAGRAM 6.2 SEARCH COMMUNITY ENGAGEMENT CONTENT

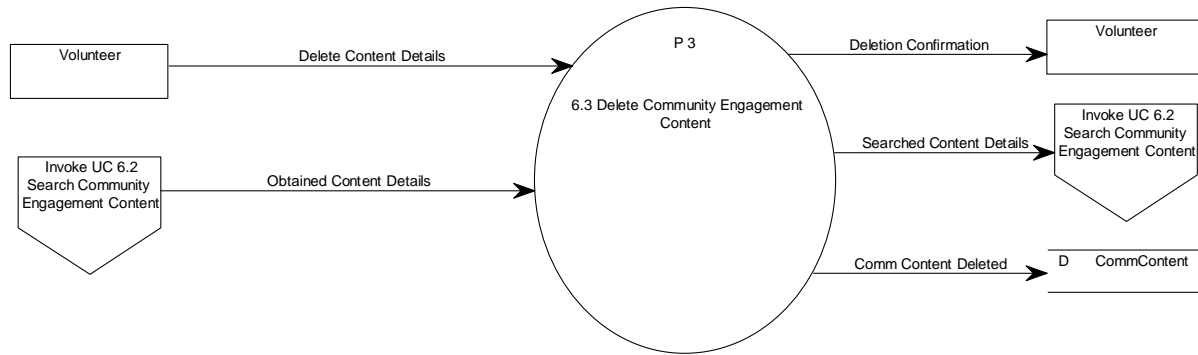


FIGURE 84- MID-LEVEL DATA FLOW DIAGRAM 6.3 DELETE COMMUNITY ENGAGEMENT CONTENT

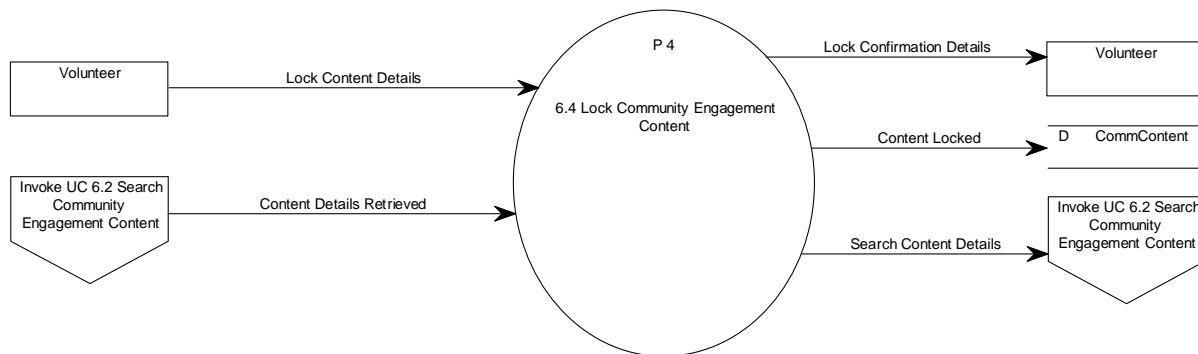


FIGURE 85- MID-LEVEL DATA FLOW DIAGRAM 6.4 LOCK COMMUNITY ENGAGEMENT CONTENT

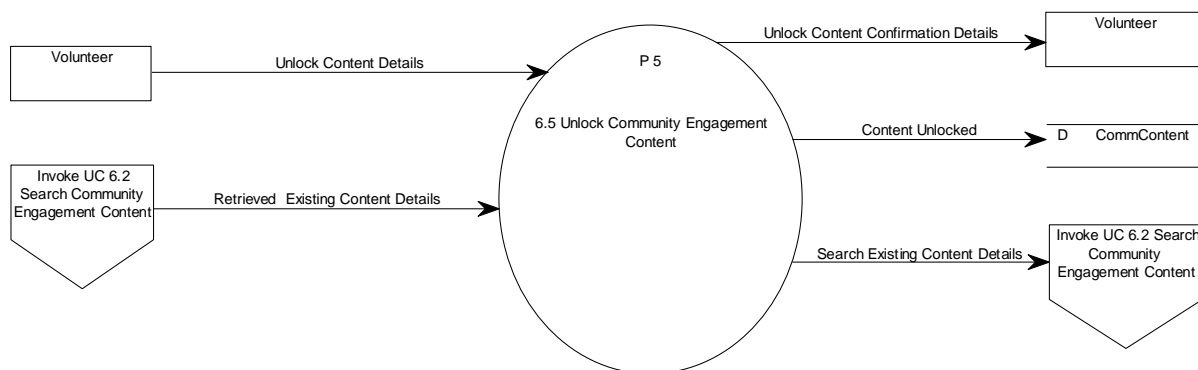


FIGURE 86- MID-LEVEL DATA FLOW DIAGRAM 6.5 UNLOCK COMMUNITY ENGAGEMENT CONTENT

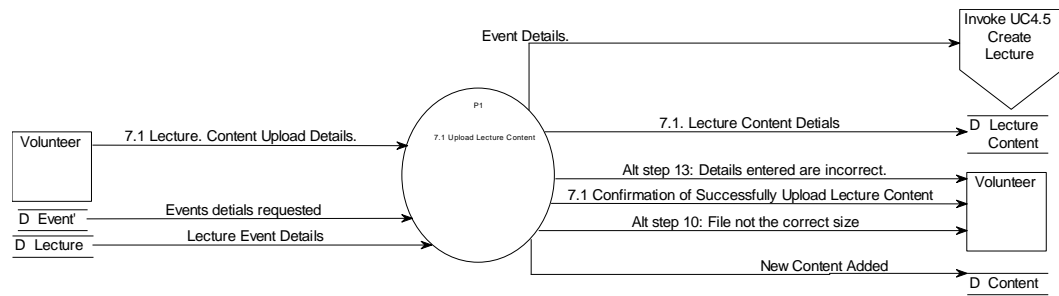


FIGURE 87- MID-LEVEL DATA FLOW DIAGRAM 7.1 UPLOAD LECTURE CONTENT

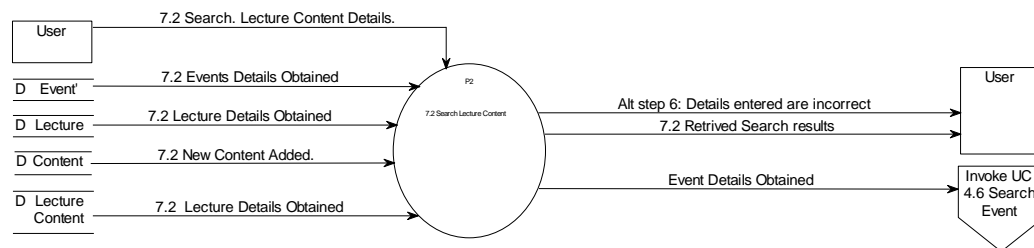


FIGURE 88- MID-LEVEL DATA FLOW DIAGRAM 7.2 SEARCH LECTURE CONTENT

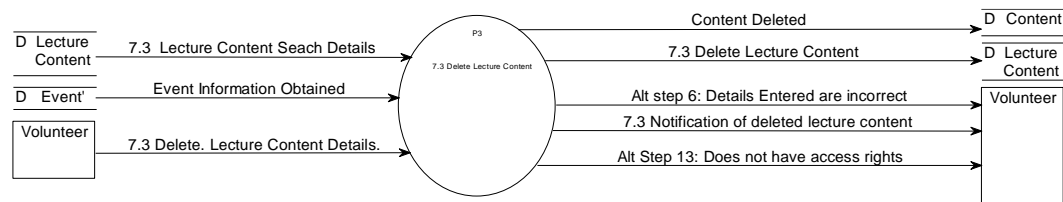


FIGURE 89- MID-LEVEL DATA FLOW DIAGRAM 7.3 DELETE LECTURE CONTENT

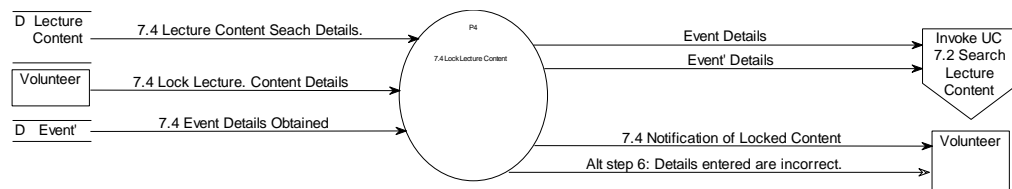


FIGURE 90- MID-LEVEL DATA FLOW DIAGRAM 7.4 LOCK LECTURE CONTENT

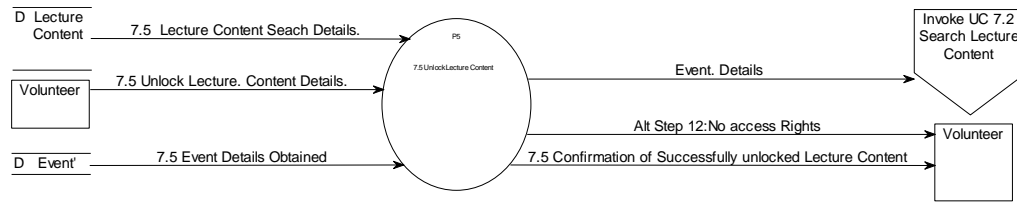


FIGURE 91- MID-LEVEL DATA FLOW DIAGRAM 7.5 UNLOCK LECTURE CONTENT

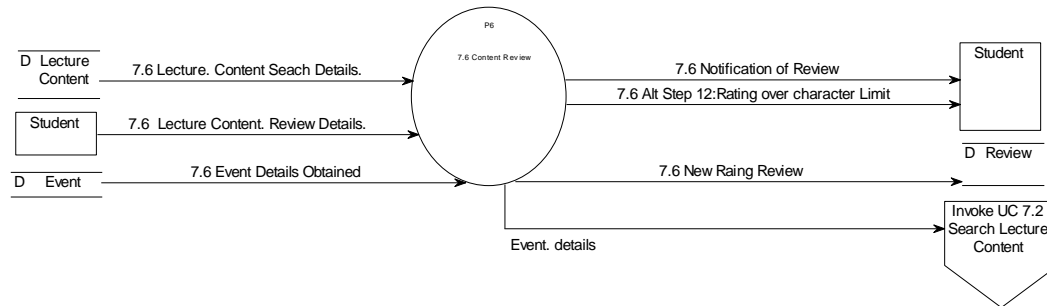


FIGURE 92- MID-LEVEL DATA FLOW DIAGRAM 7.6 CONTENT REVIEW

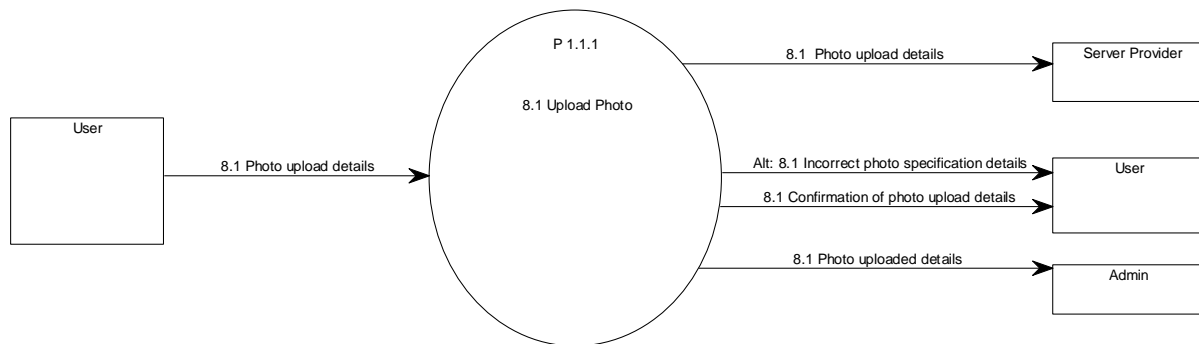


FIGURE 93- MID-LEVEL DATA FLOW DIAGRAM 8.1 UPLOAD PHOTO



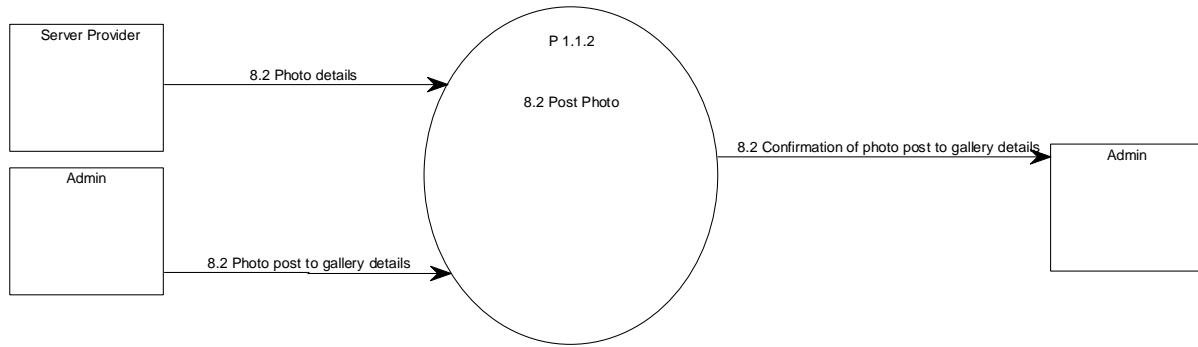


FIGURE 94- MID-LEVEL DATA FLOW DIAGRAM 8.2 POST PHOTO

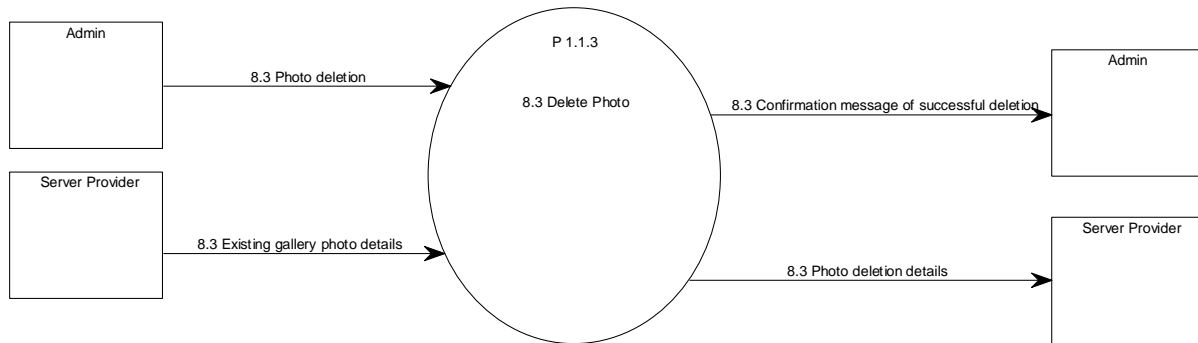


FIGURE 95- MID-LEVEL DATA FLOW DIAGRAM 8.3 DELETE PHOTO

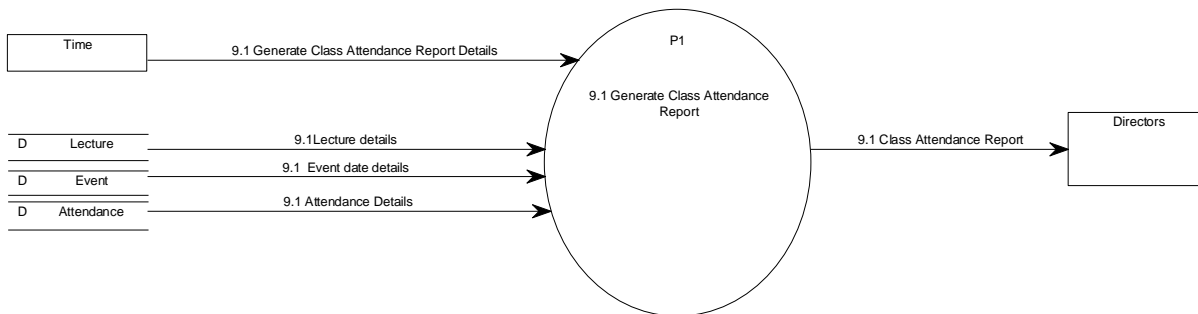


FIGURE 96- MID-LEVEL DATA FLOW DIAGRAM 9.1 GENERATE CLASS ATTENDANCE REPORT

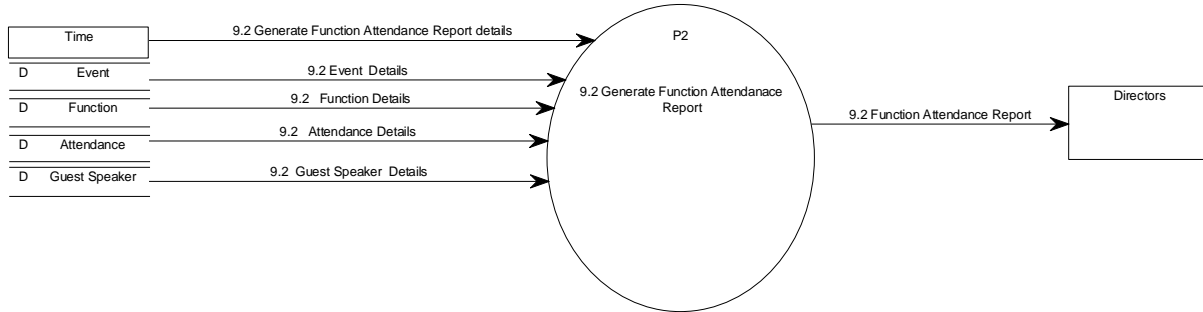


FIGURE 97- MID-LEVEL DATA FLOW DIAGRAM 9.2 GENERATE FUNCTION ATTENDANCE REPORT

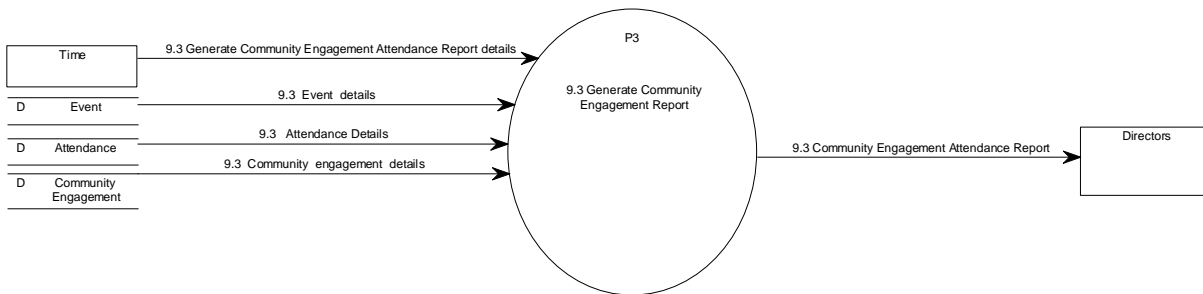


FIGURE 98- MID-LEVEL DATA FLOW DIAGRAM 9.3 GENERATE COMMUNITY ENGAGEMENT REPORT

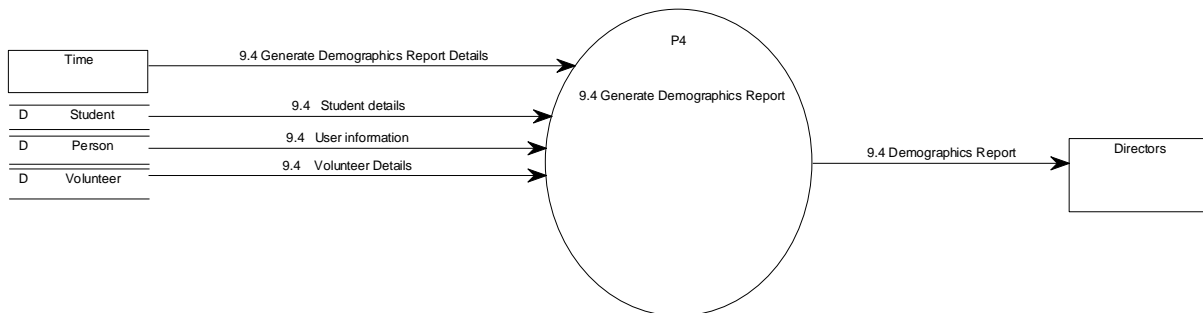


FIGURE 99- MID-LEVEL DATA FLOW DIAGRAM 9.4 GENERATE DEMOGRAPHICS REPORT

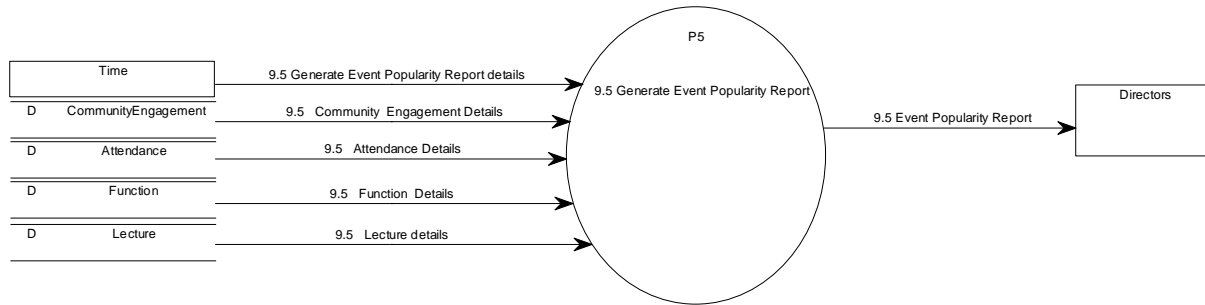


FIGURE 100- MID-LEVEL DATA FLOW DIAGRAM 9.5 GENERATE EVENT POPULARITY REPORT

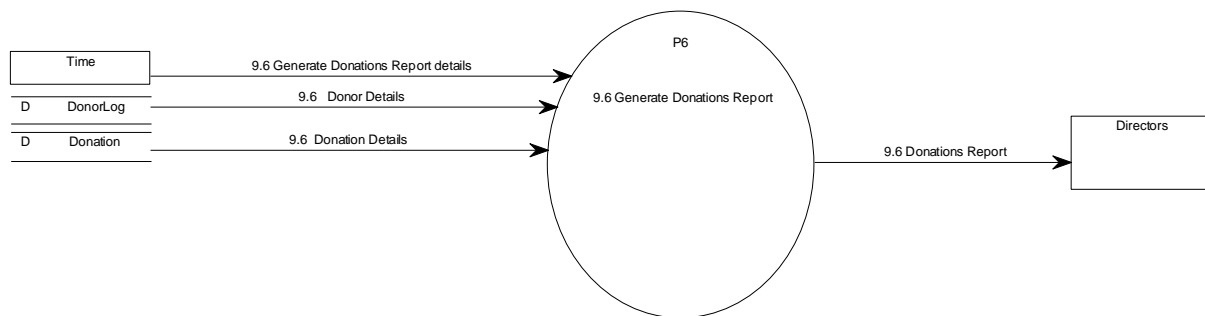


FIGURE 101- MID-LEVEL DATA FLOW DIAGRAM 9.6 GENERATE DONATIONS REPORT

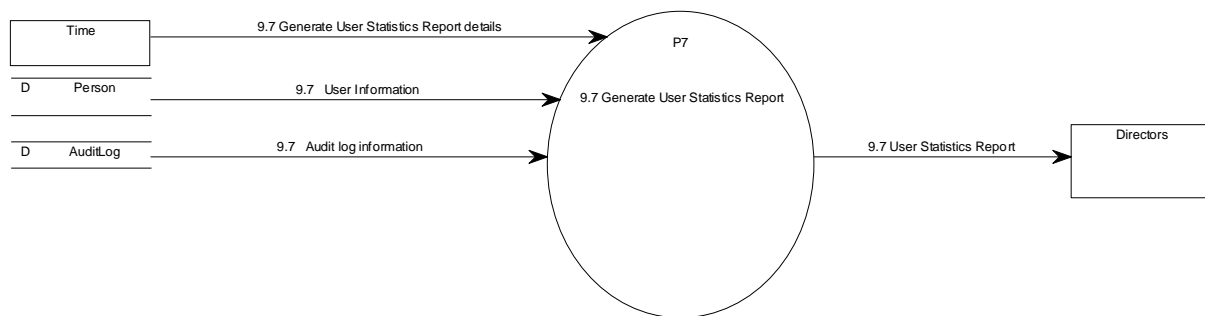


FIGURE 102- MID-LEVEL DATA FLOW DIAGRAM 9.7 GENERATE USER STATISTICS REPORT

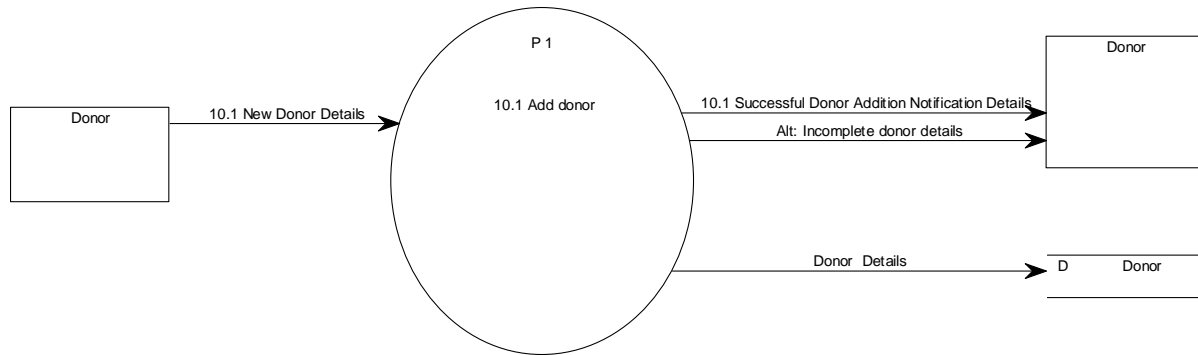


FIGURE 103- MID-LEVEL DATA FLOW DIAGRAM 10.1 ADD DONOR

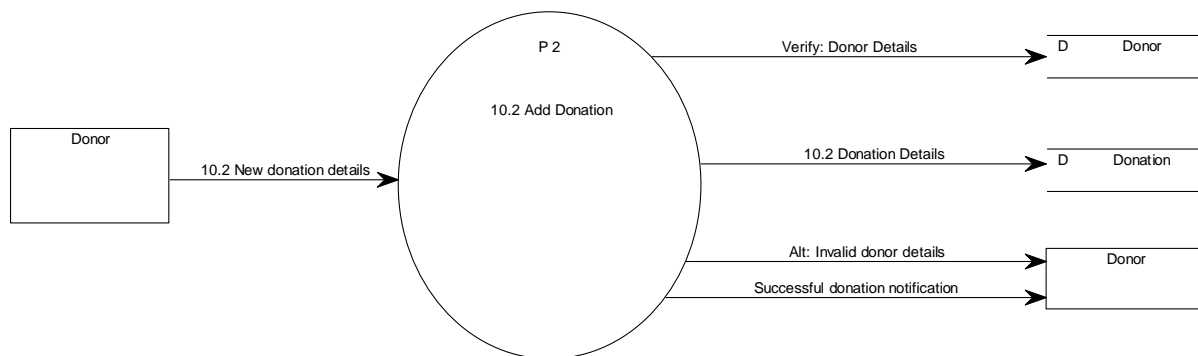


FIGURE 104- MID-LEVEL DATA FLOW DIAGRAM 10.2 ADD DONATION

### 3.4.3 Primitive-Level Data Flow Diagrams

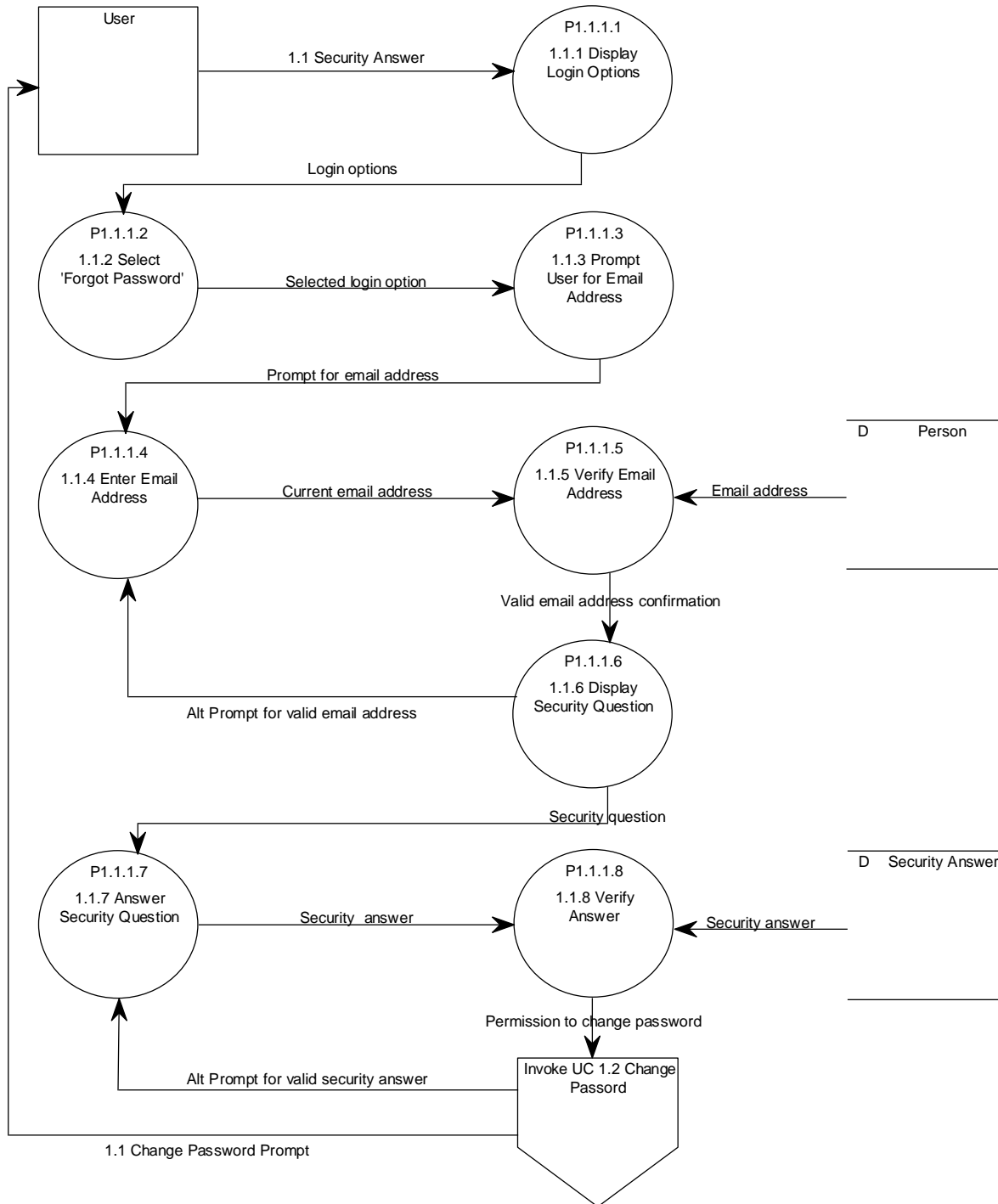


FIGURE 105- PRIMITIVE LEVEL DATA FLOW DIAGRAM 1.1 CHECK FORGOTTEN PASSWORD

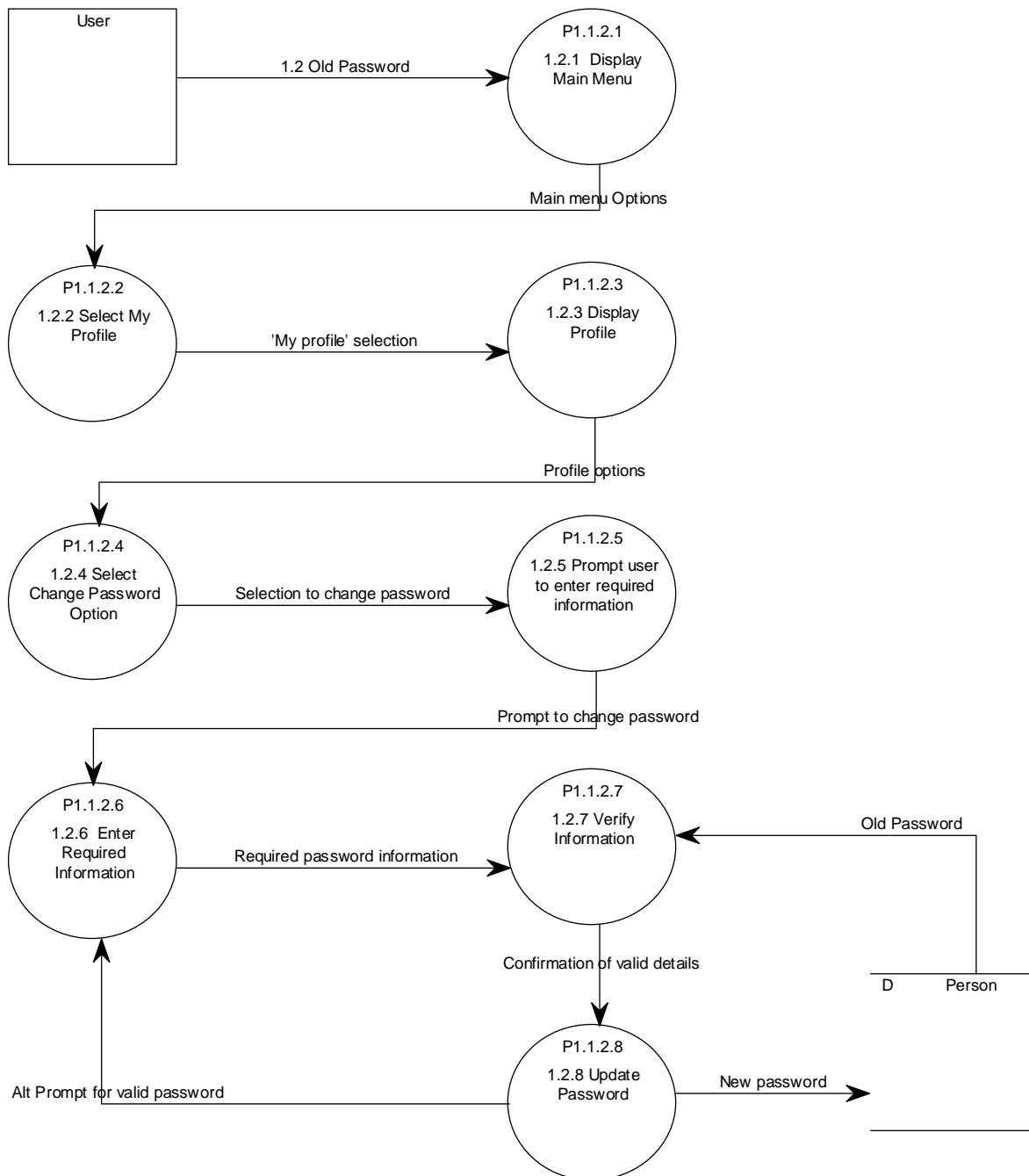


FIGURE 106- PRIMITIVE LEVEL DATA FLOW DIAGRAM 1.2 CHANGE PASSWORD

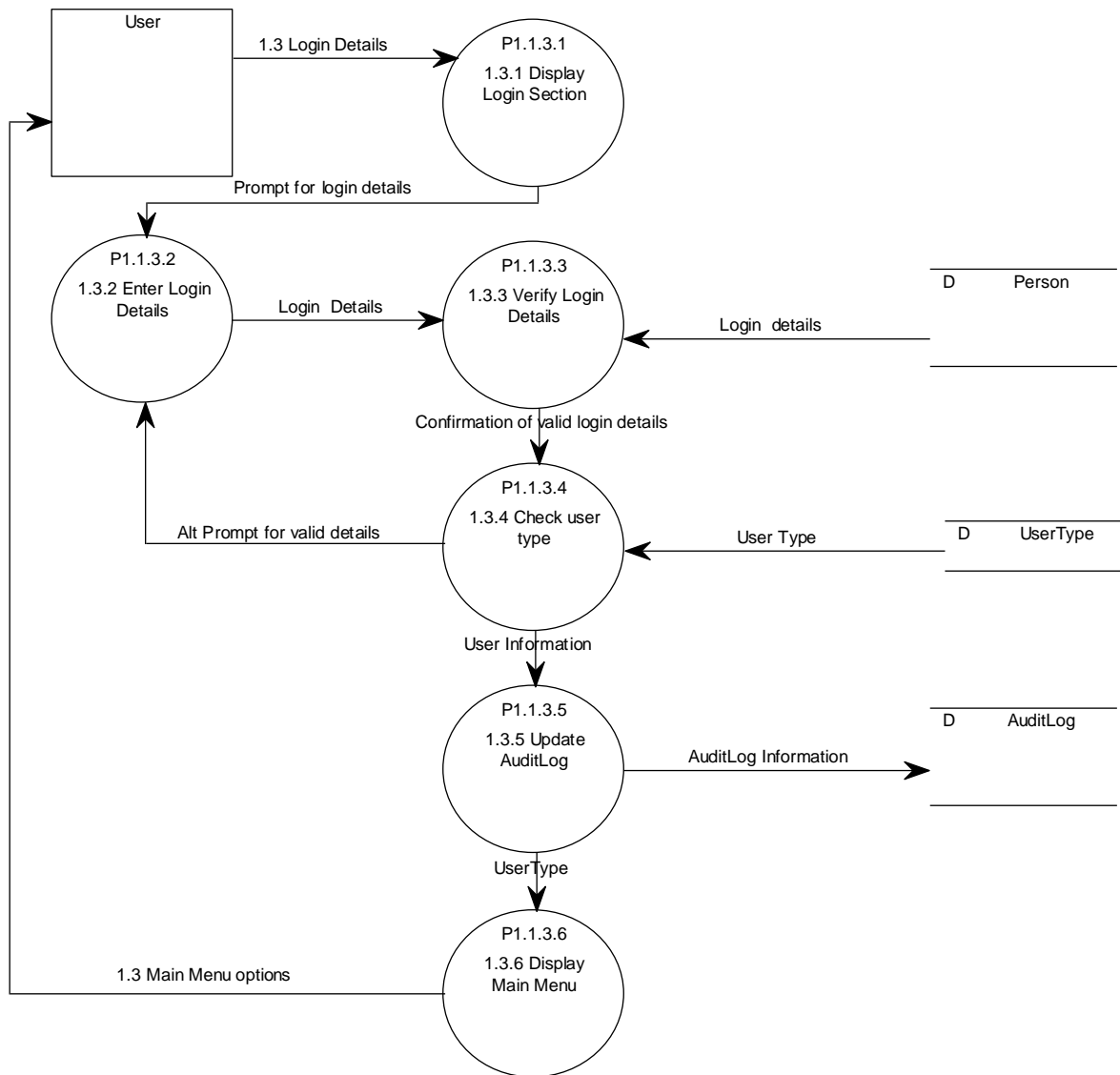


FIGURE 107- PRIMITIVE LEVEL DATA FLOW DIAGRAM 1.3 LOGIN

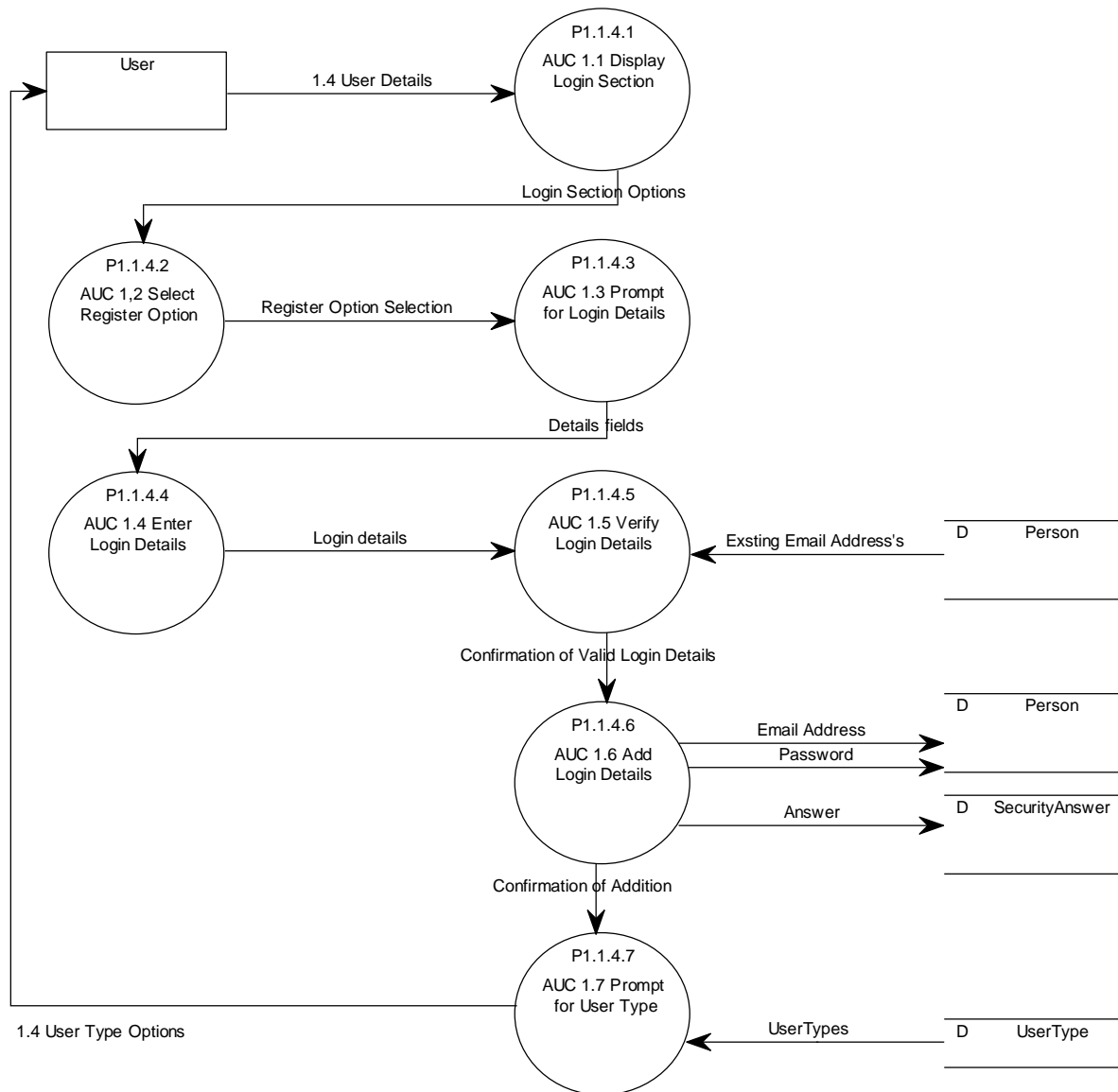


FIGURE 108- PRIMITIVE LEVEL DATA FLOW DIAGRAM AUC 1 REGISTER USER



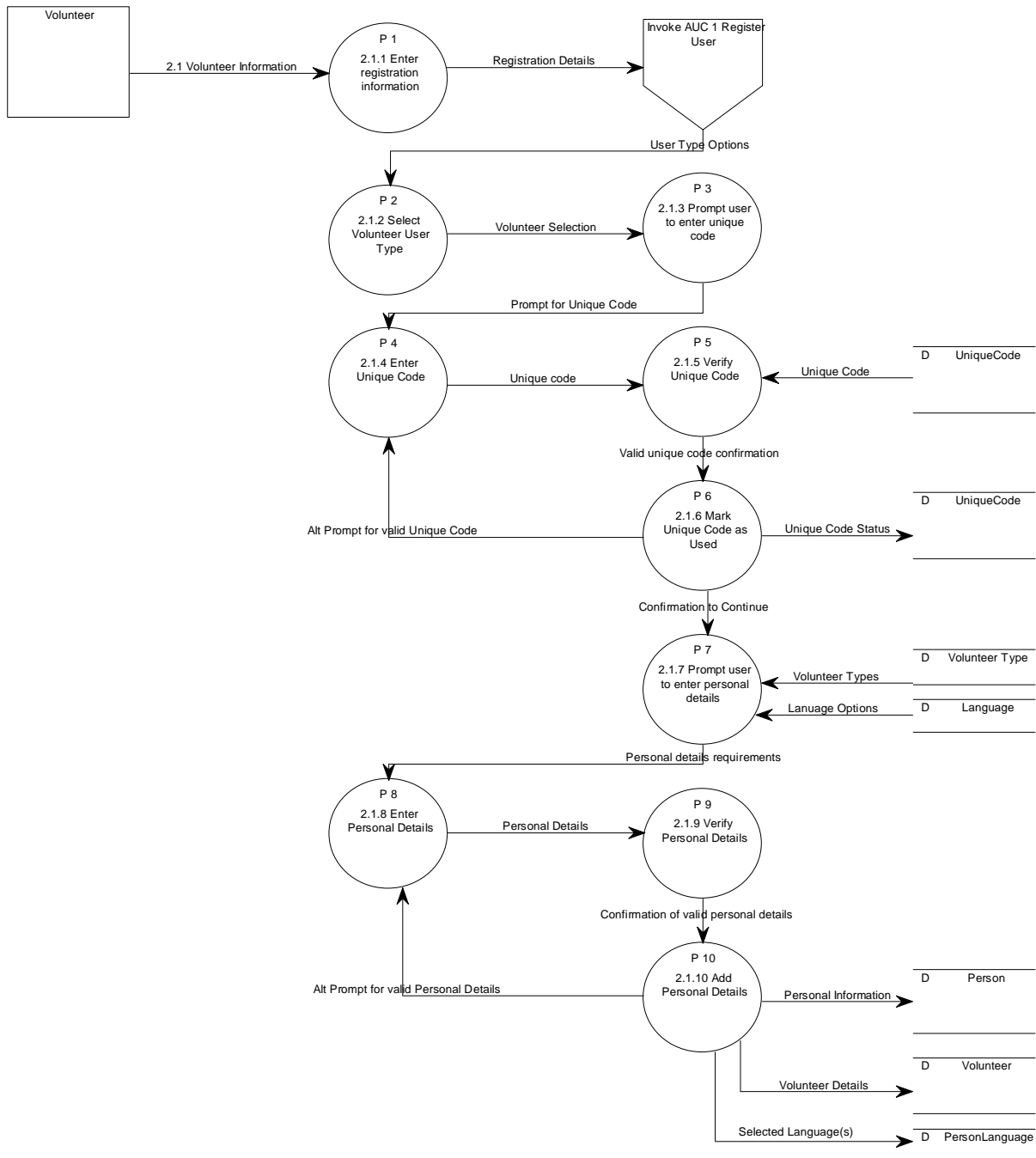


FIGURE 109- PRIMITIVE LEVEL DATA FLOW DIAGRAM 2.1 REGISTER VOLUNTEER

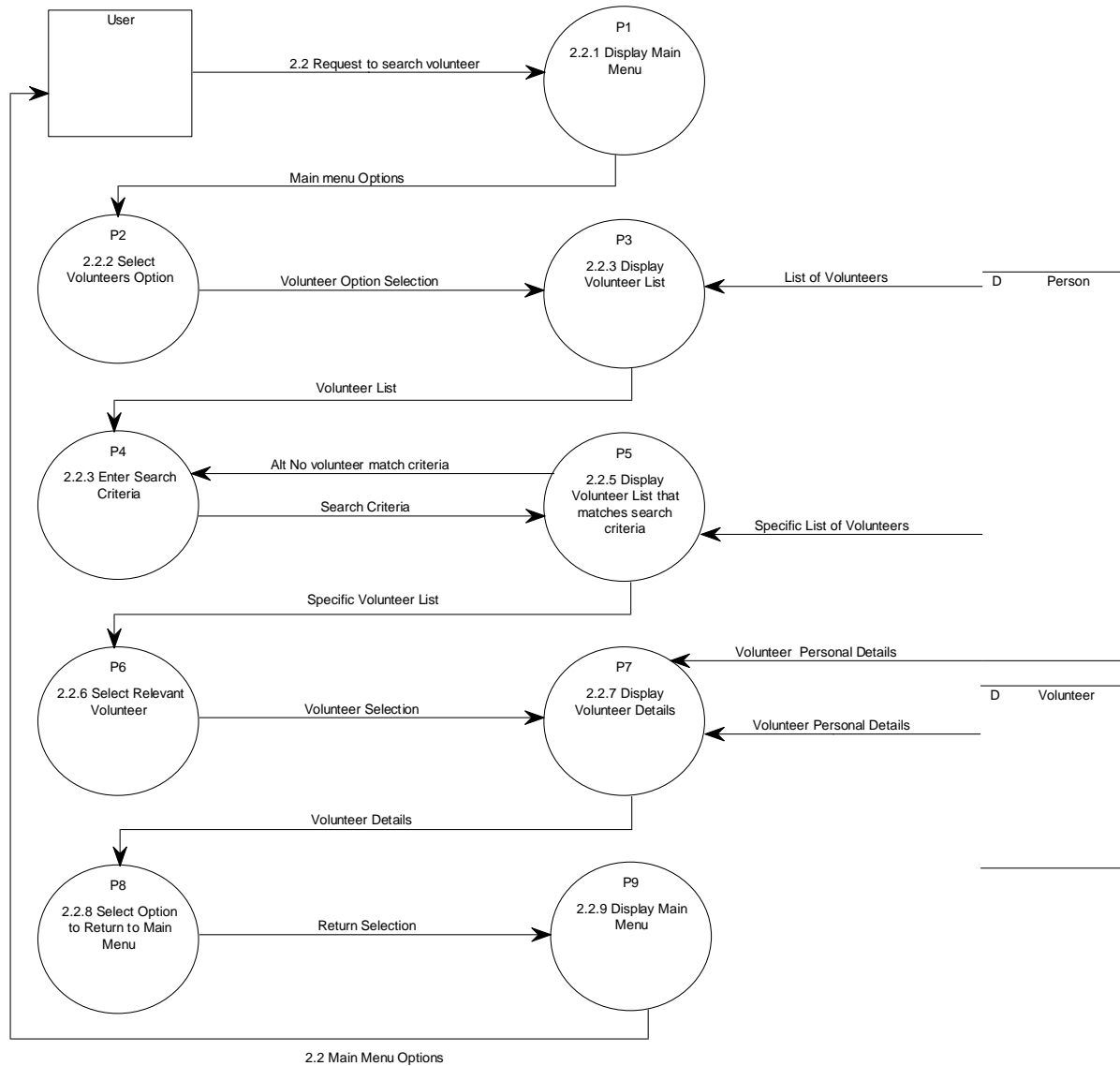


FIGURE 110- PRIMITIVE LEVEL DATA FLOW DIAGRAM 2.2 SEARCH VOLUNTEER

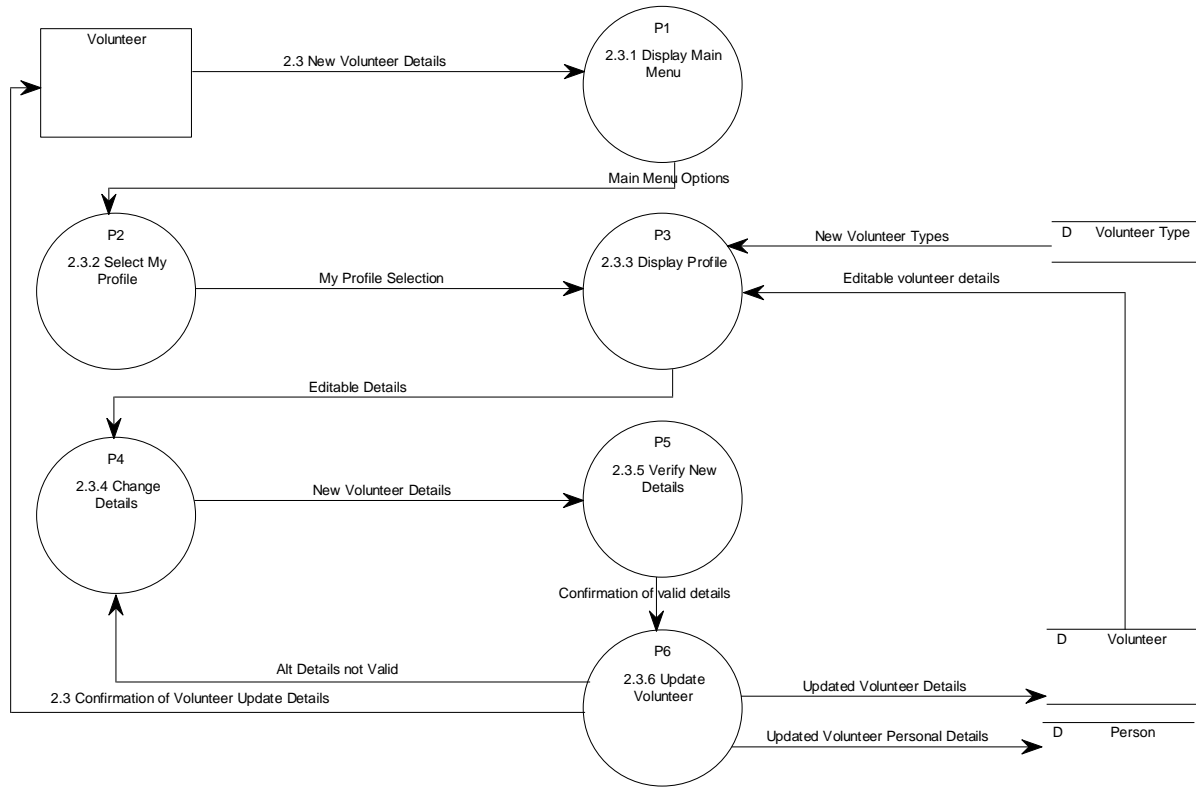


FIGURE 111- PRIMITIVE LEVEL DATA FLOW DIAGRAM 2.3 UPDATE VOLUNTEER

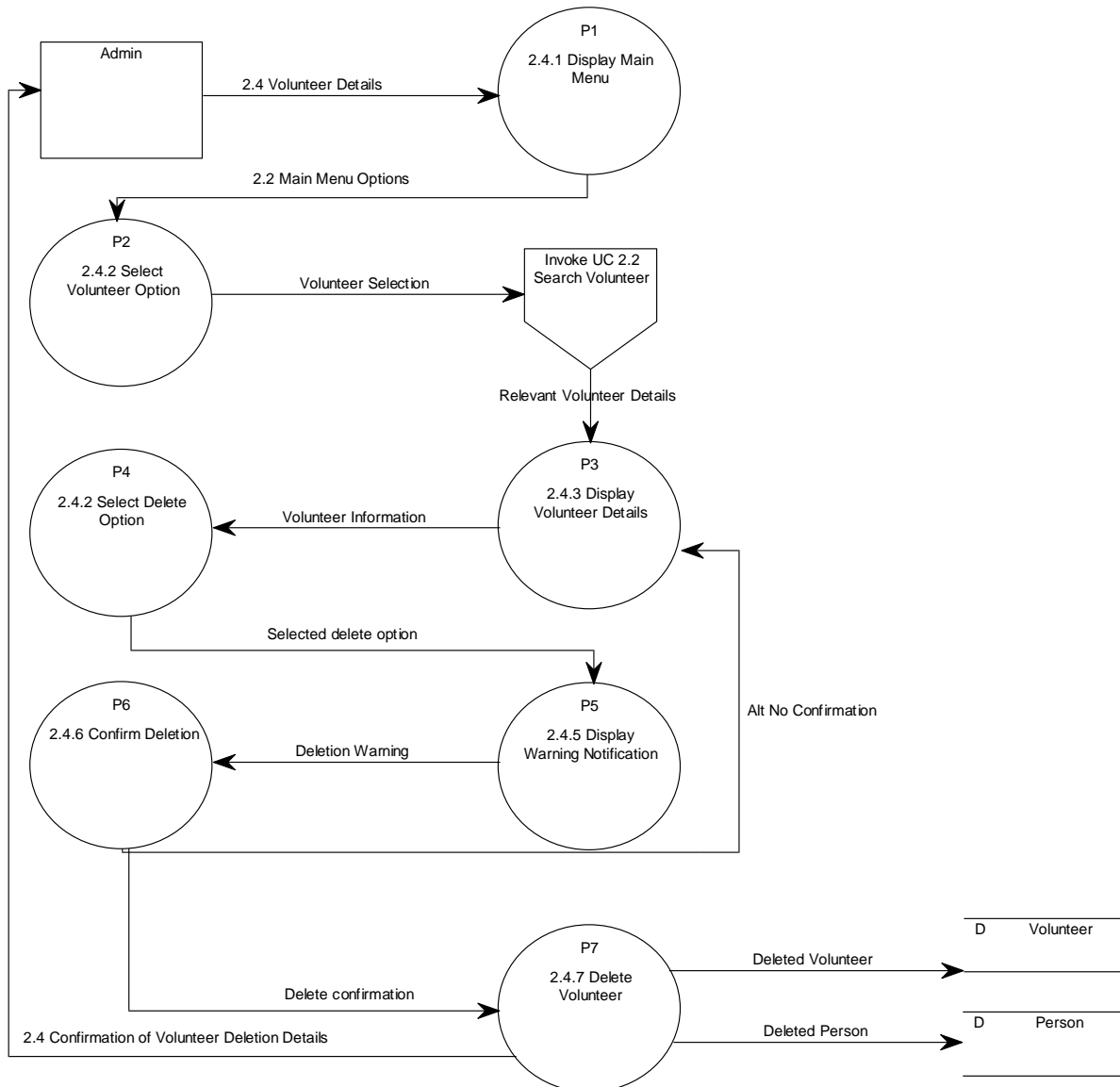


FIGURE 112- PRIMITIVE LEVEL DATA FLOW DIAGRAM 2.4 DELETE VOLUNTEER

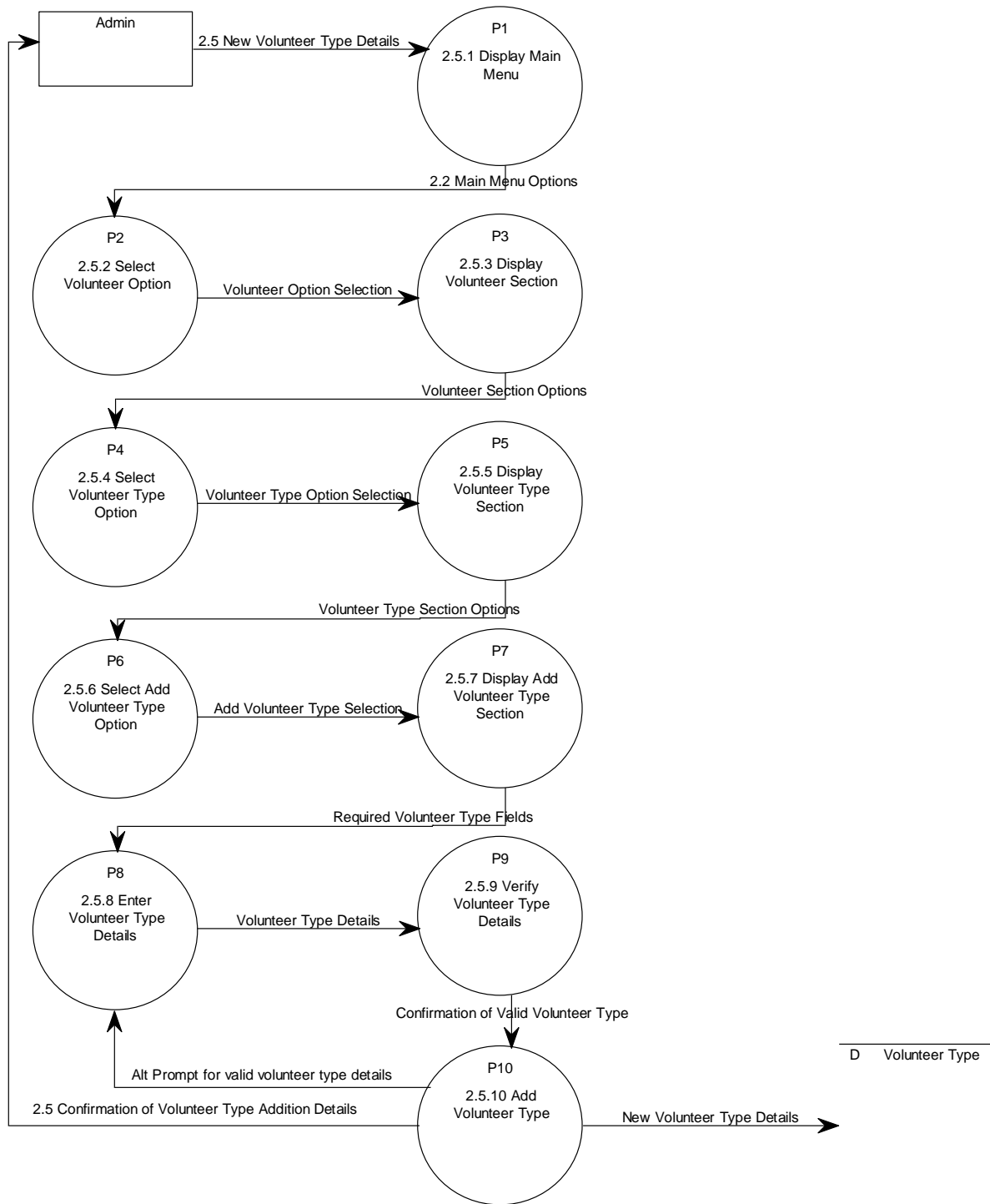


FIGURE 113- PRIMITIVE LEVEL DATA FLOW DIAGRAM 2.5 ADD VOLUNTEER TYPE

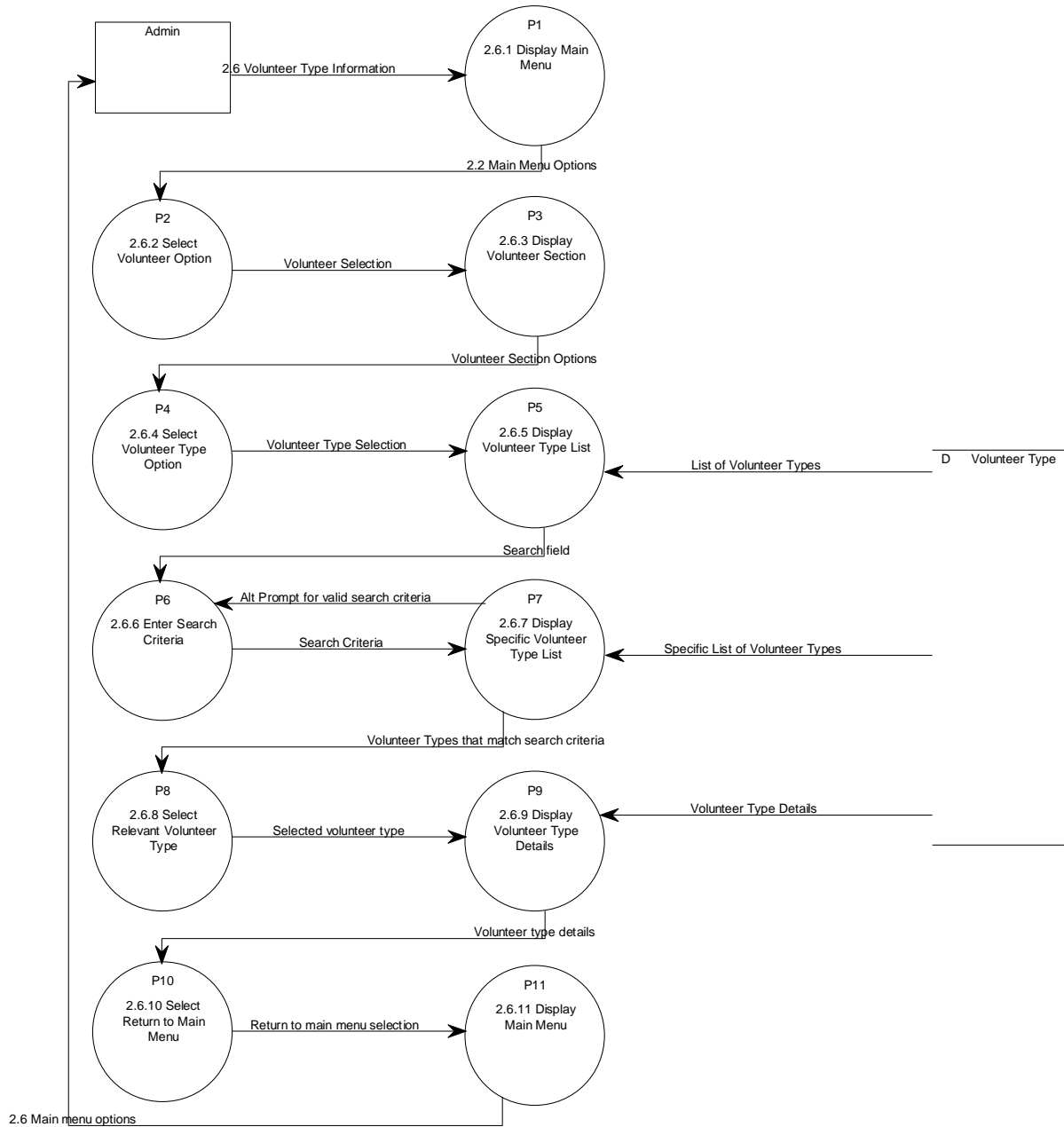


FIGURE 114- PRIMITIVE LEVEL DATA FLOW DIAGRAM 2.6 SEARCH VOLUNTEER TYPE

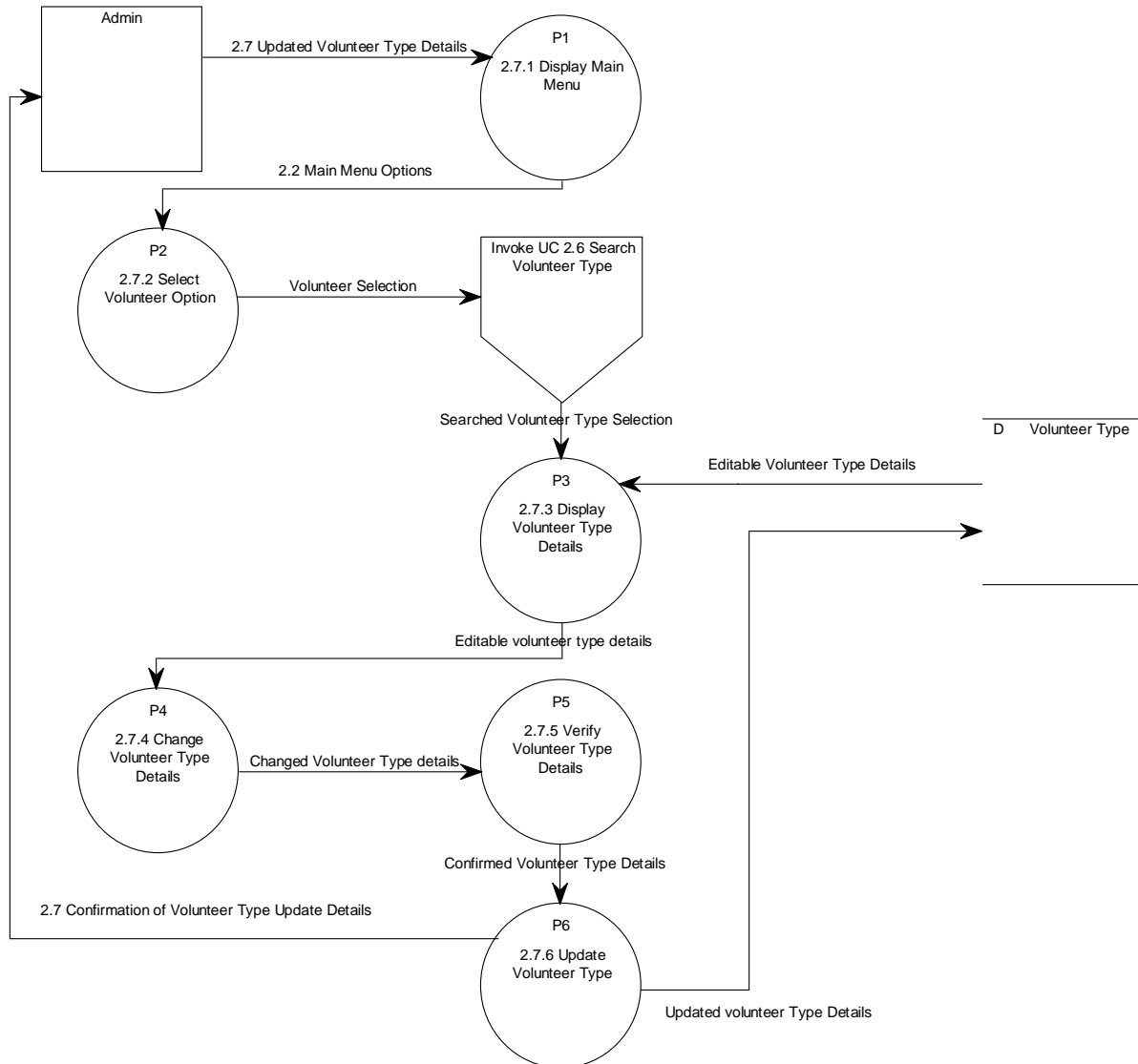


FIGURE 115- PRIMITIVE LEVEL DATA FLOW DIAGRAM 2.7 UPDATE VOLUNTEER TYPE

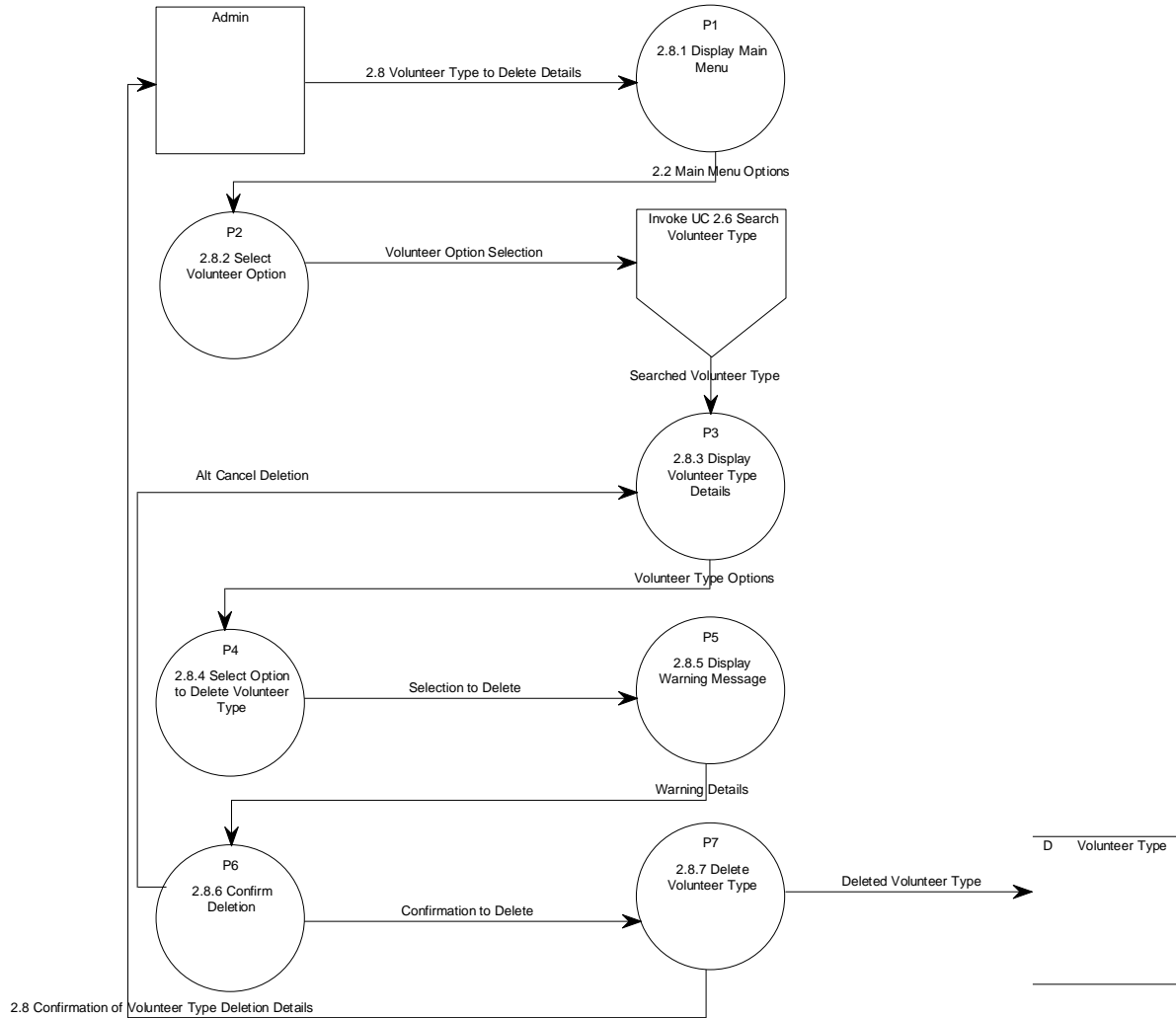


FIGURE 116- PRIMITIVE LEVEL DATA FLOW DIAGRAM 2.8 DELETE VOLUNTEER TYPE



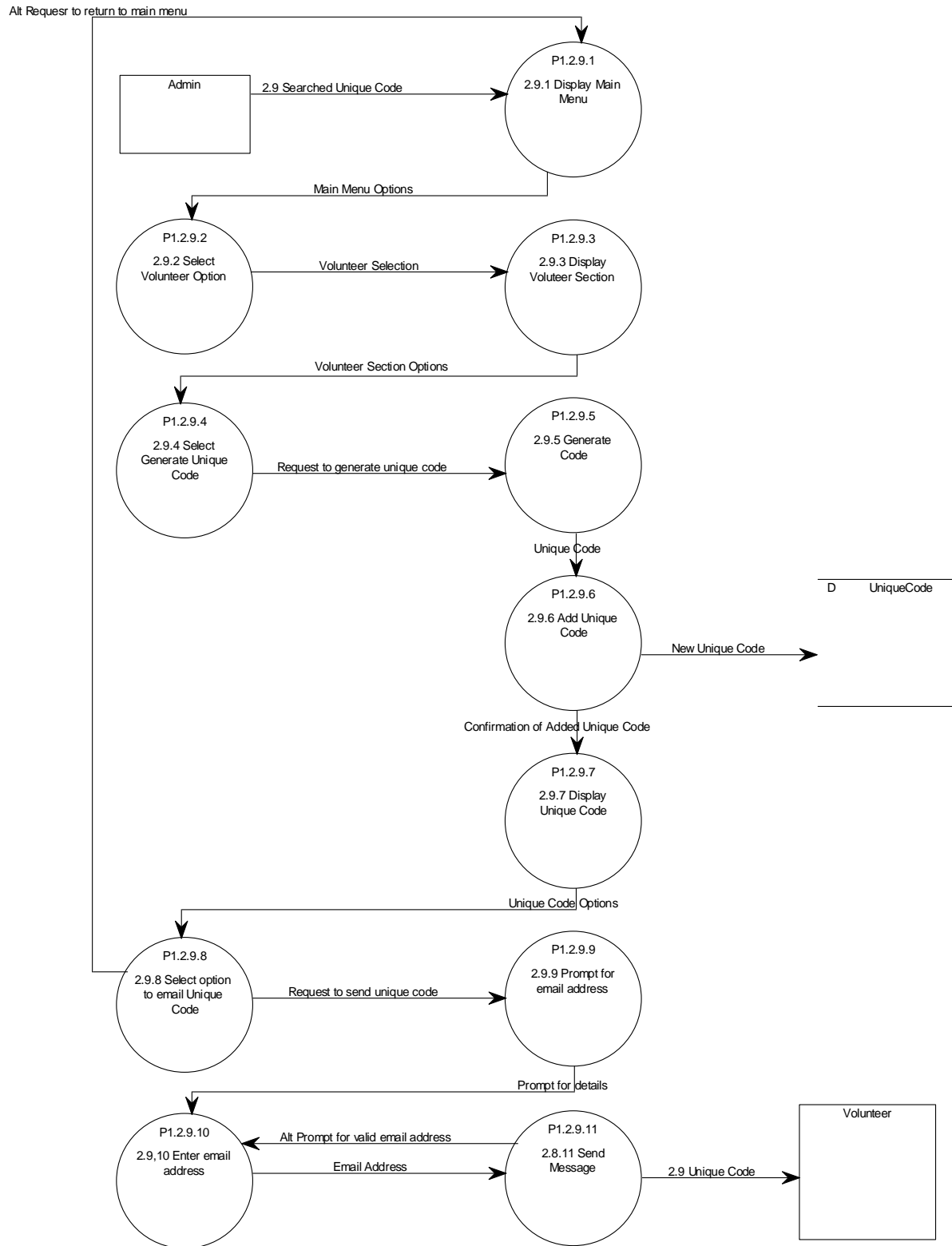


FIGURE 117- PRIMITIVE LEVEL DATA FLOW DIAGRAM 2.9 GENERATE UNIQUE CODE

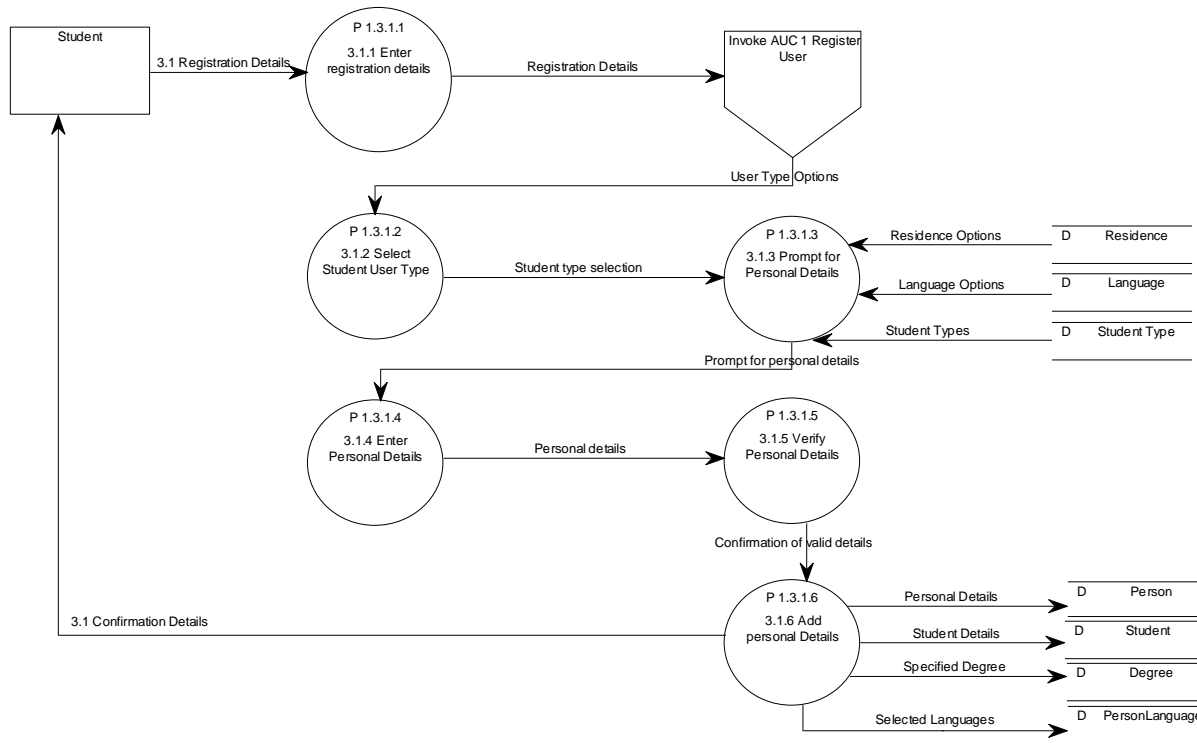


FIGURE 118- PRIMITIVE LEVEL DATA FLOW DIAGRAM 3.1 REGISTER STUDENT

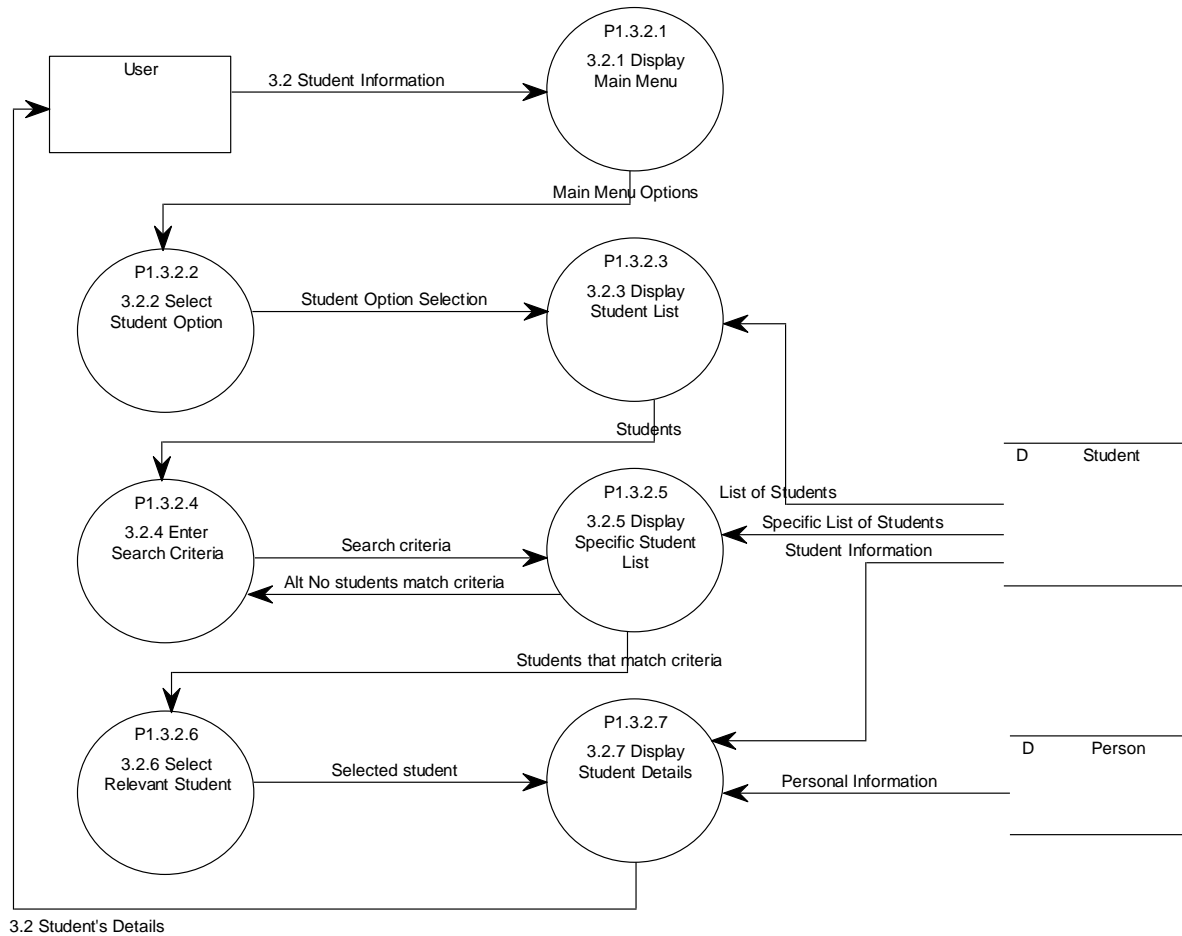


FIGURE 119- PRIMITIVE LEVEL DATA FLOW DIAGRAM 3.2 SEARCH STUDENT

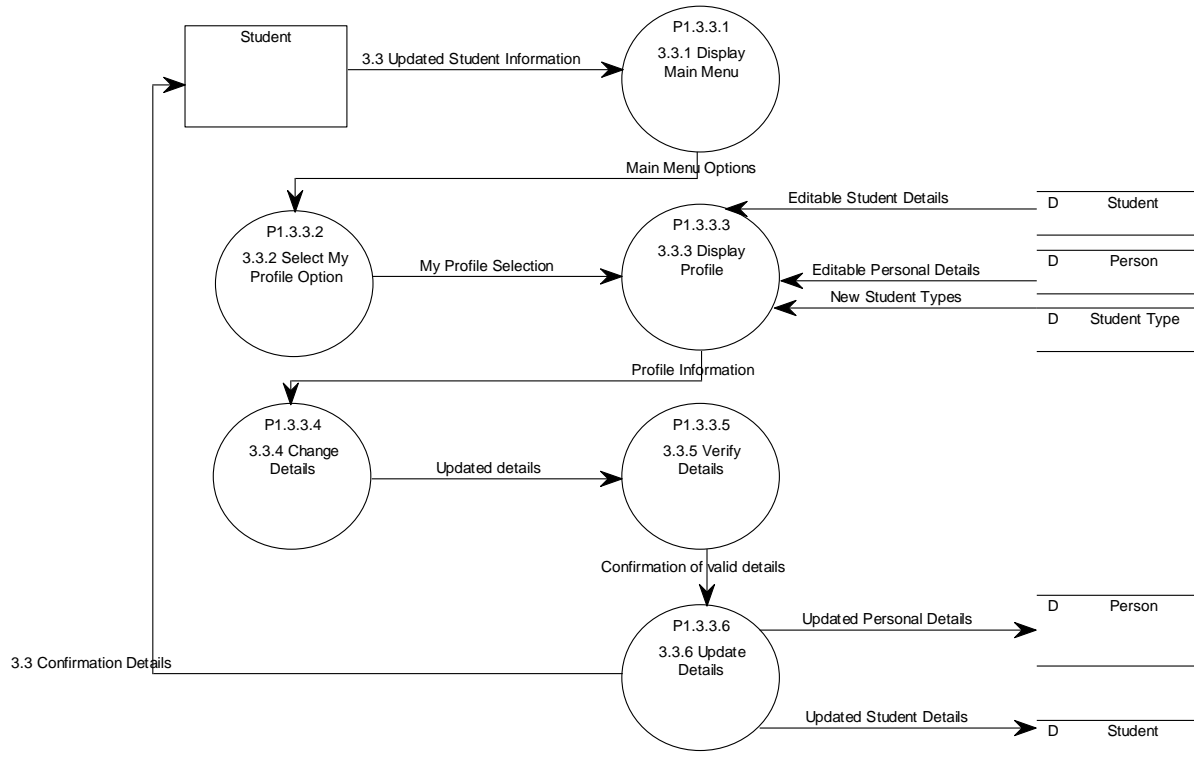


FIGURE 120- PRIMITIVE LEVEL DATA FLOW DIAGRAM 3.3 UPDATE STUDENT

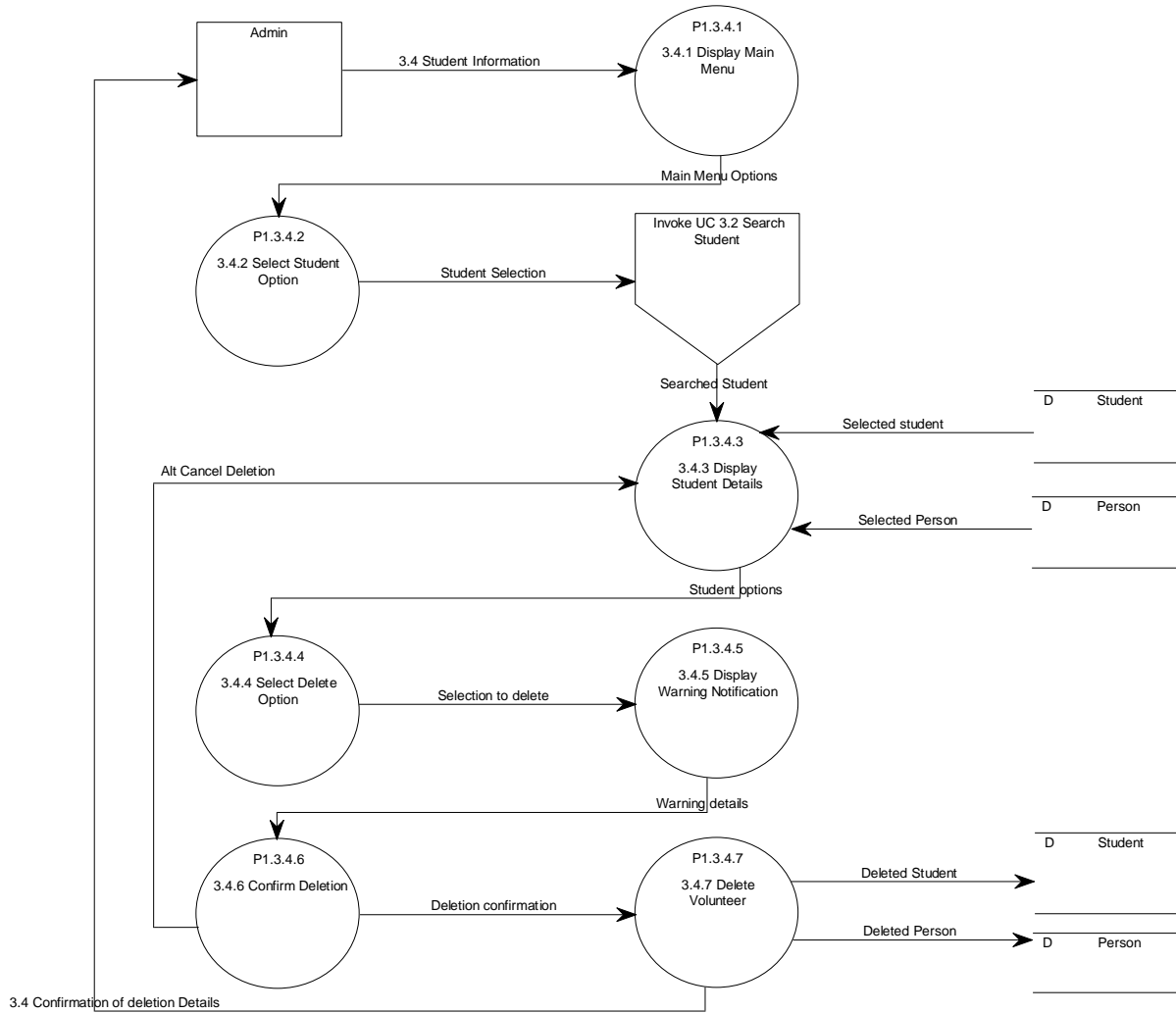


FIGURE 121- PRIMITIVE LEVEL DATA FLOW DIAGRAM 3.4 DELETE VOLUNTEER

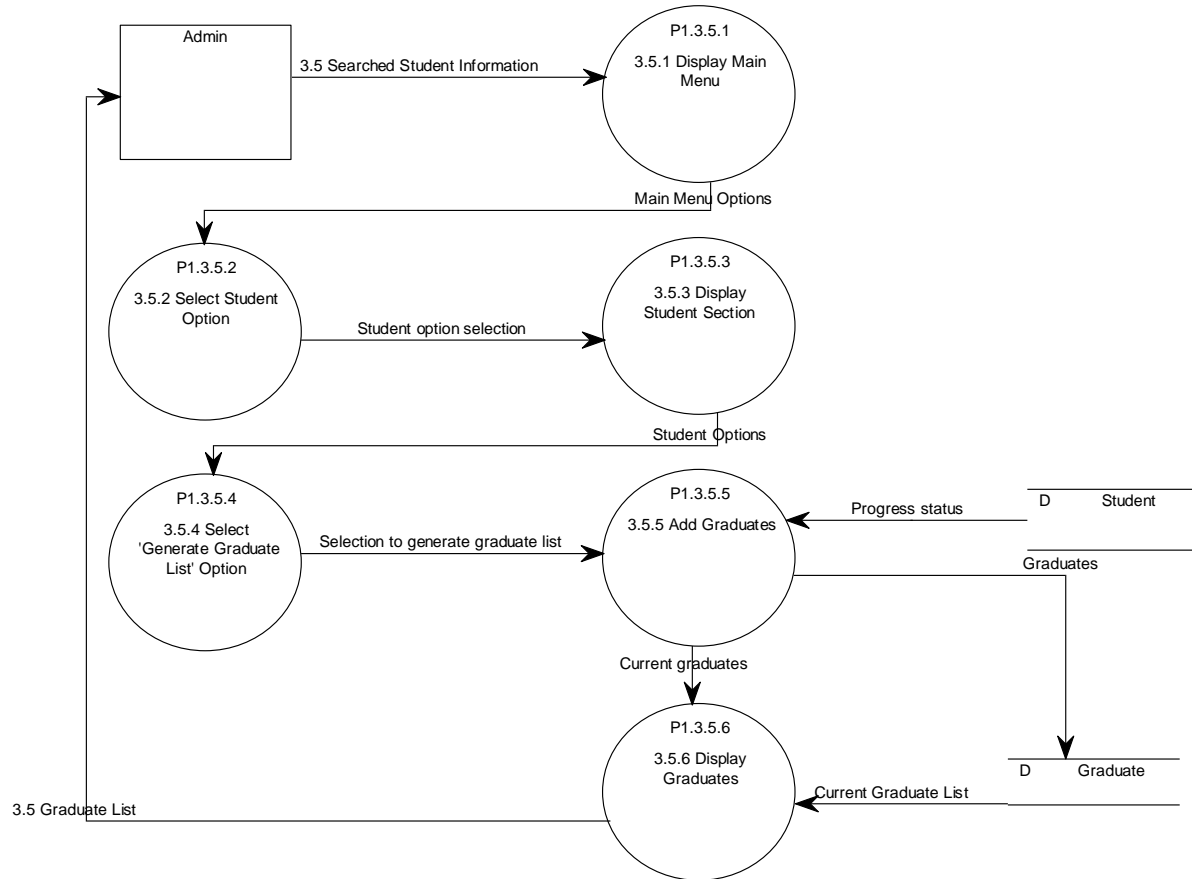


FIGURE 122- PRIMITIVE LEVEL DATA FLOW DIAGRAM 3.5 GENERATE GRADUATE LIST



FIGURE 123- PRIMITIVE LEVEL DATA FLOW DIAGRAM 3.6 ADD STUDENT TYPE

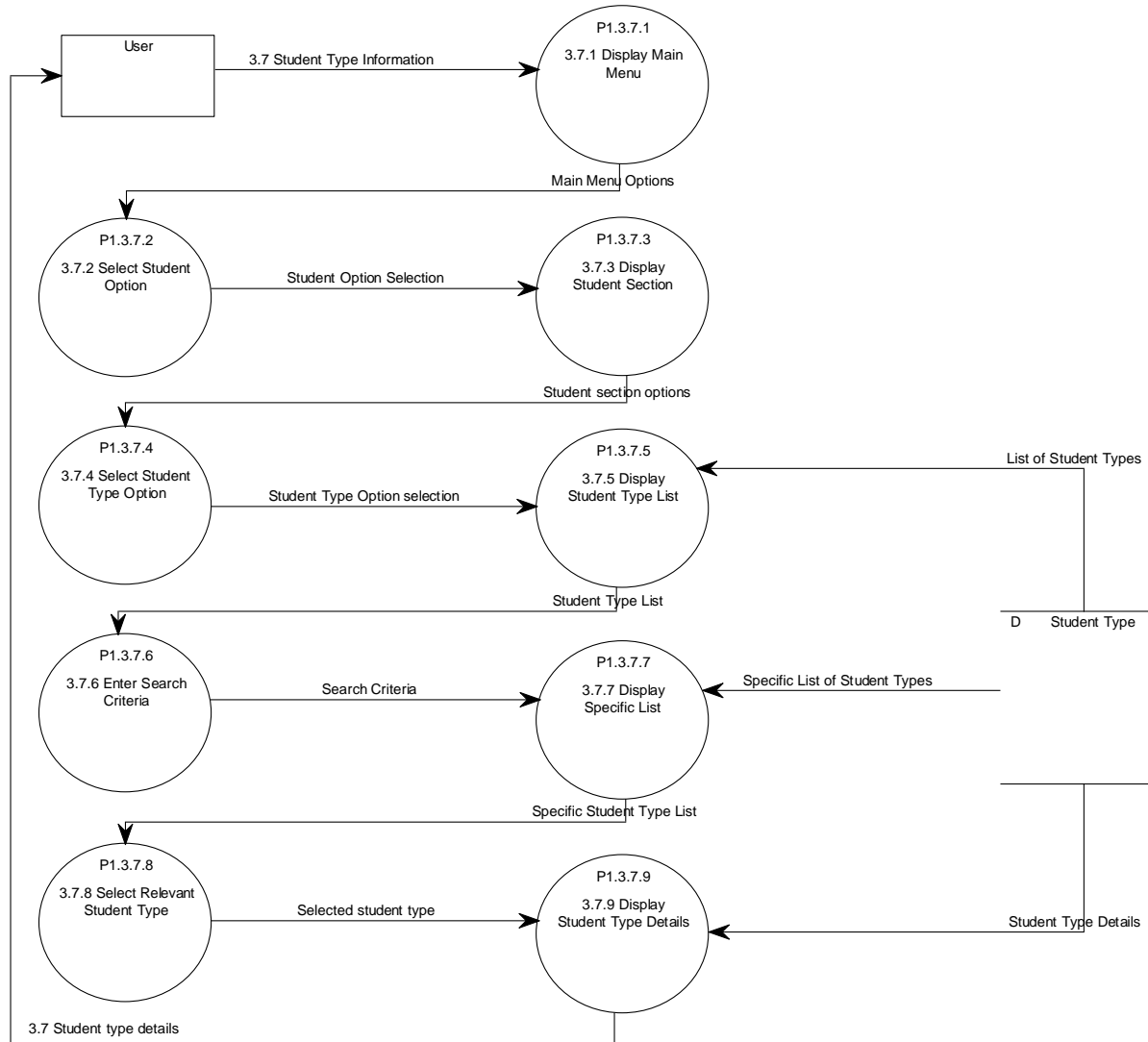


FIGURE 124- PRIMITIVE LEVEL DATA FLOW DIAGRAM 3.7 SEARCH STUDENT TYPE



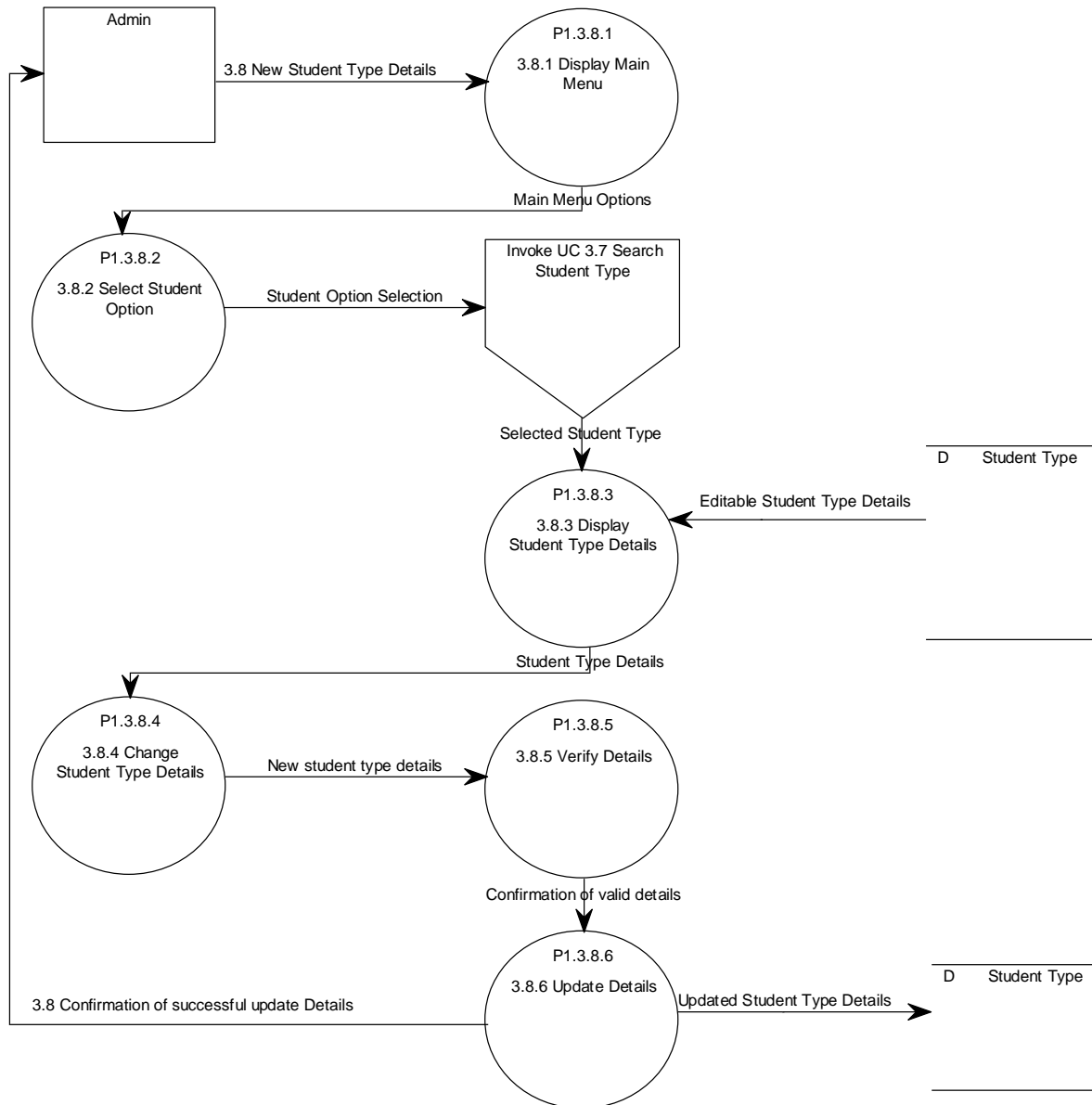


FIGURE 125- PRIMITIVE LEVEL DATA FLOW DIAGRAM 3.8 UPDATE STUDENT TYPE

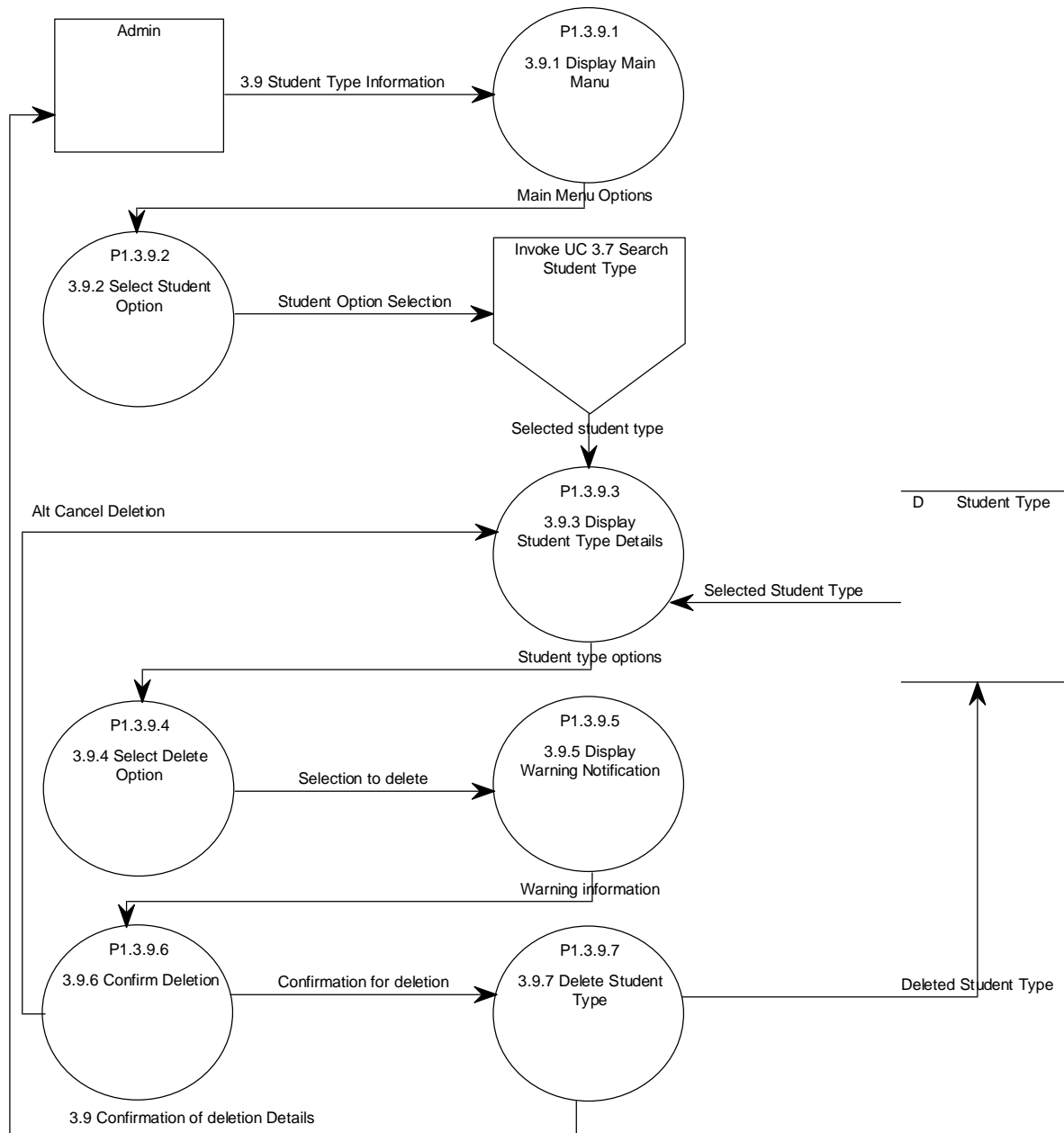


FIGURE 126- PRIMITIVE LEVEL DATA FLOW DIAGRAM 3.9 DELETE STUDENT TYPE

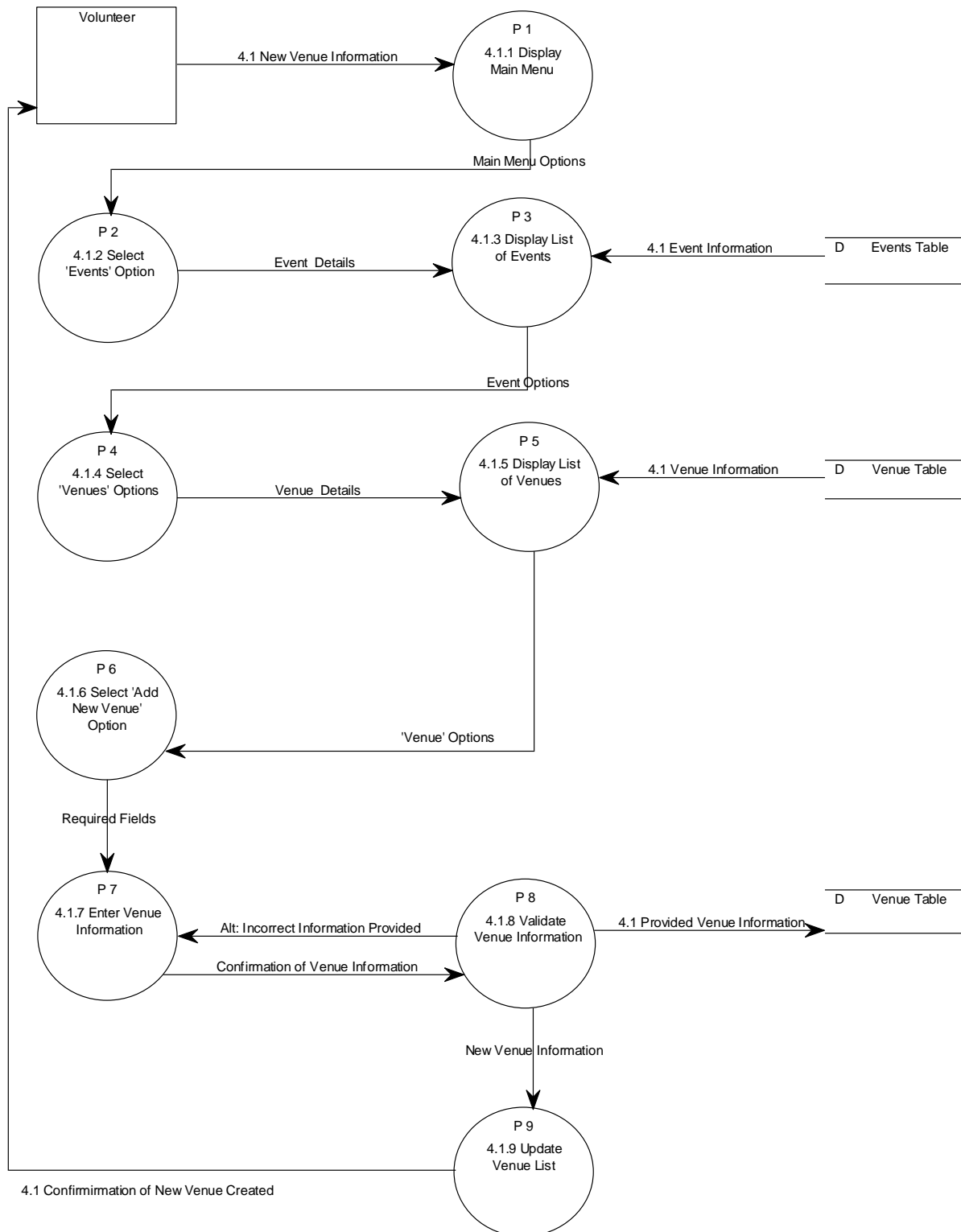


FIGURE 127- PRIMITIVE LEVEL DATA FLOW DIAGRAM 4.1 ADD NEW VENUE

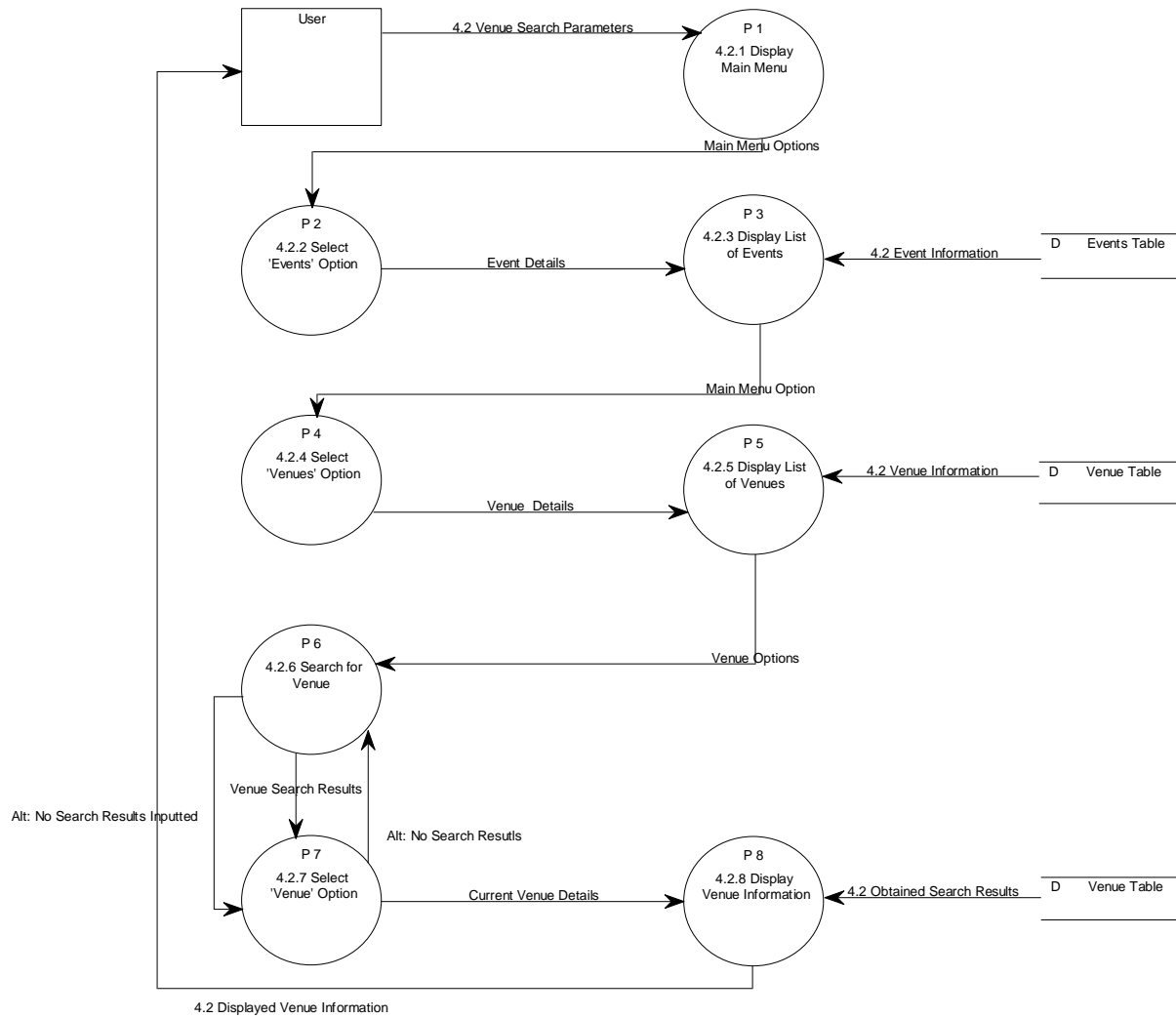


FIGURE 128- PRIMITIVE LEVEL DATA FLOW DIAGRAM 4.2 SEARCH VENUE

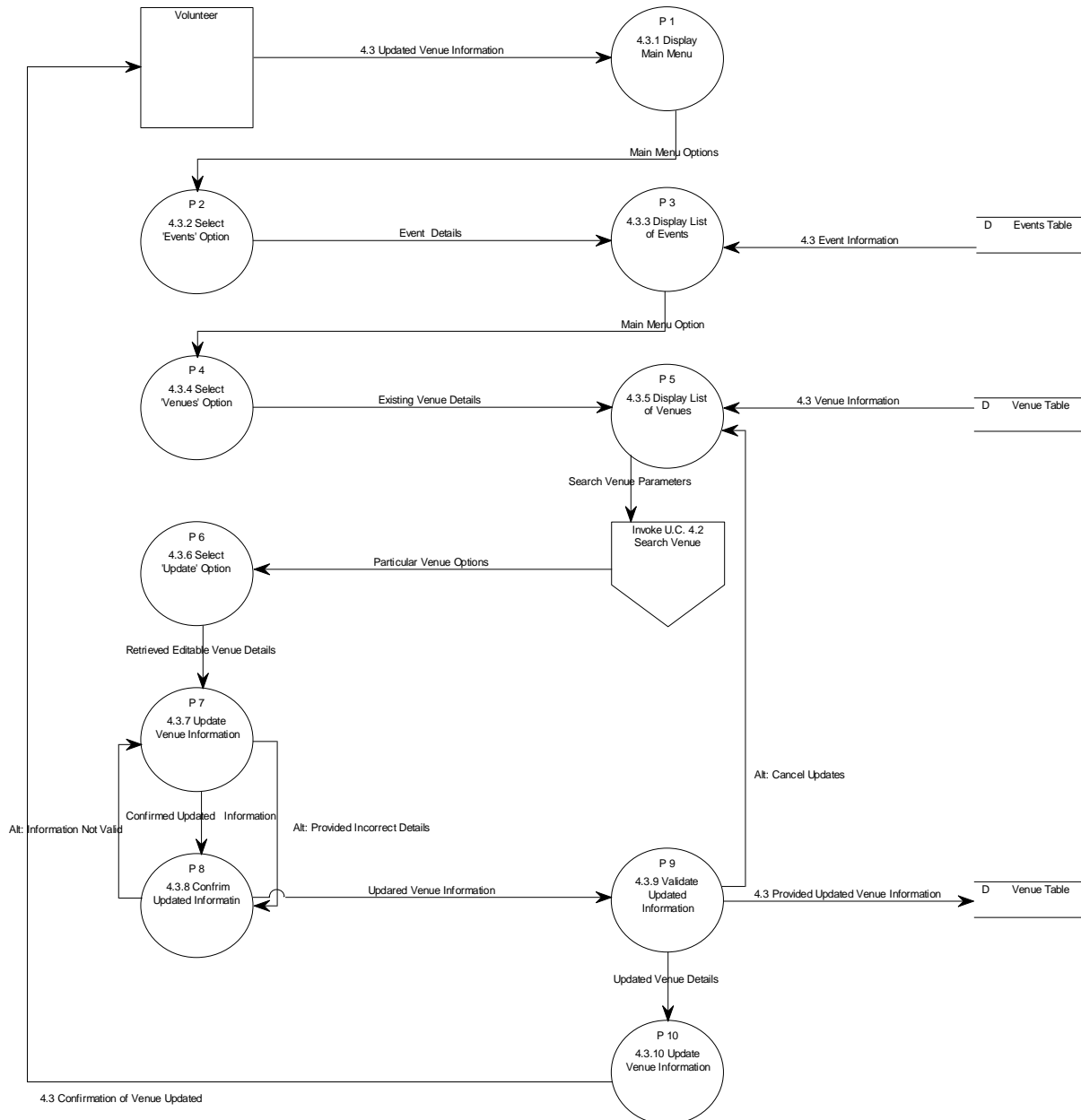


FIGURE 129- PRIMITIVE LEVEL DATA FLOW DIAGRAM 4.3 UPDATE VENUE

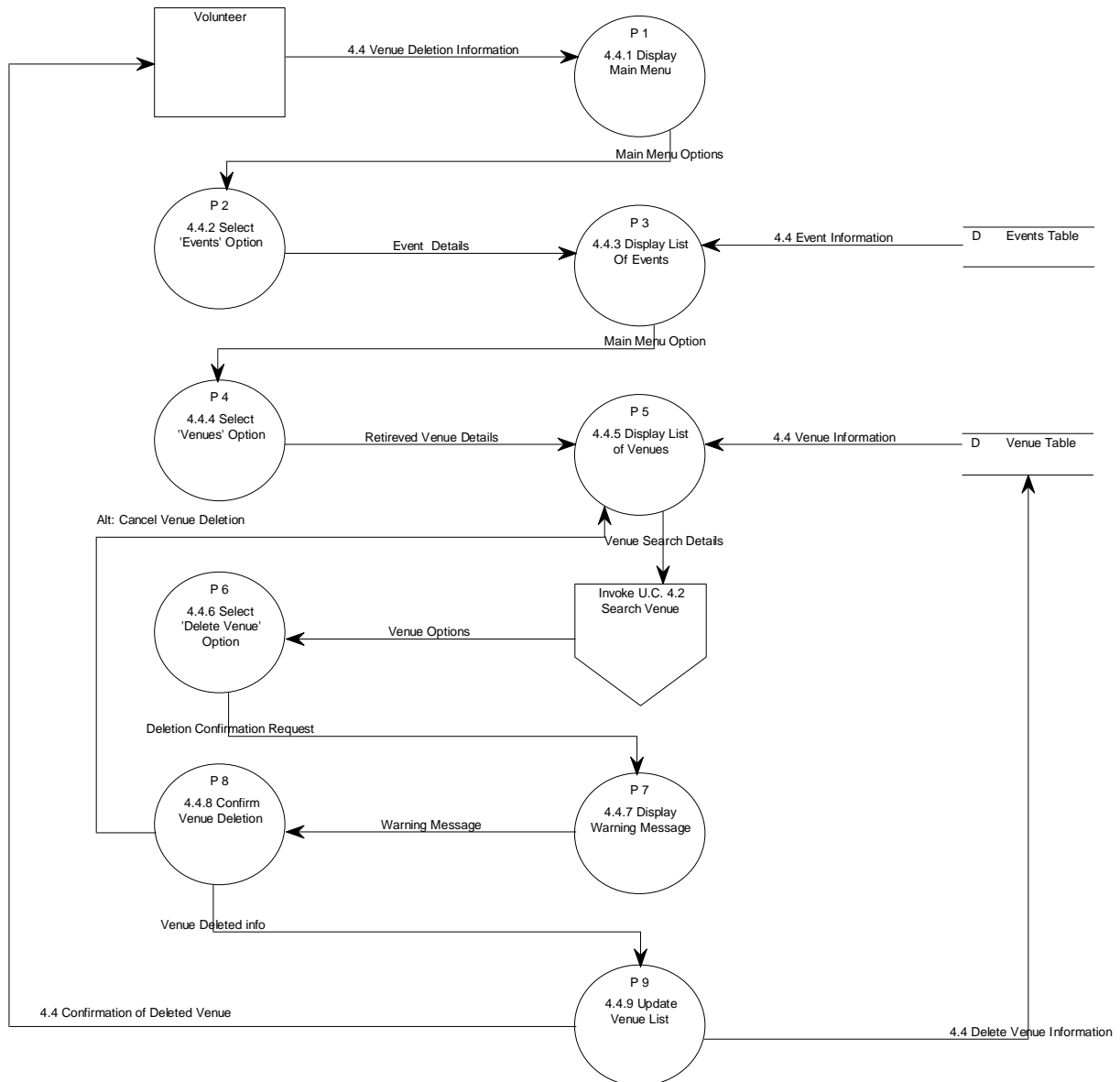


FIGURE 130- PRIMITIVE LEVEL DATA FLOW DIAGRAM 4.4 DELETE VENUE

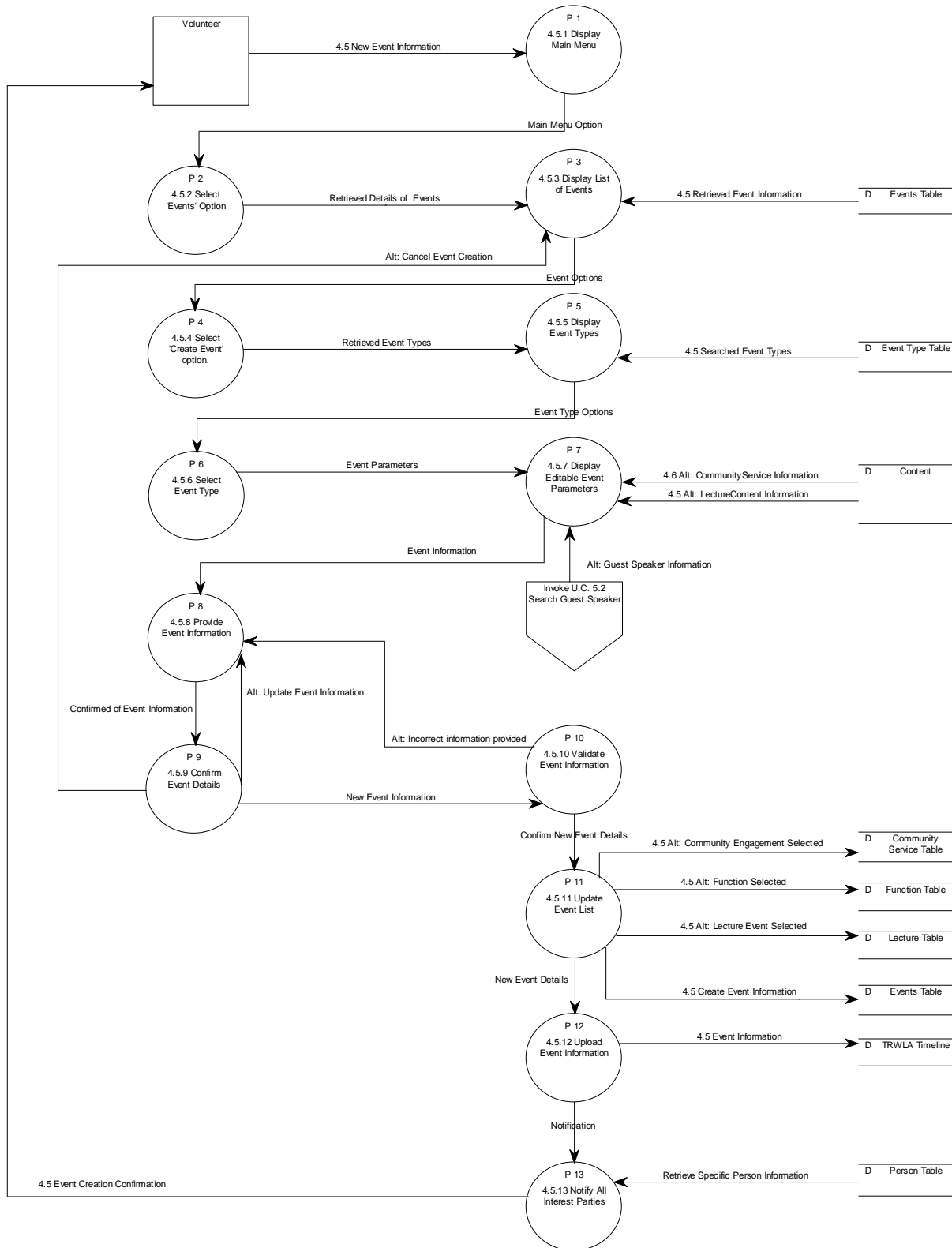


FIGURE 131- PRIMITIVE LEVEL DATA FLOW DIAGRAM 4.5 CREATE NEW EVENT

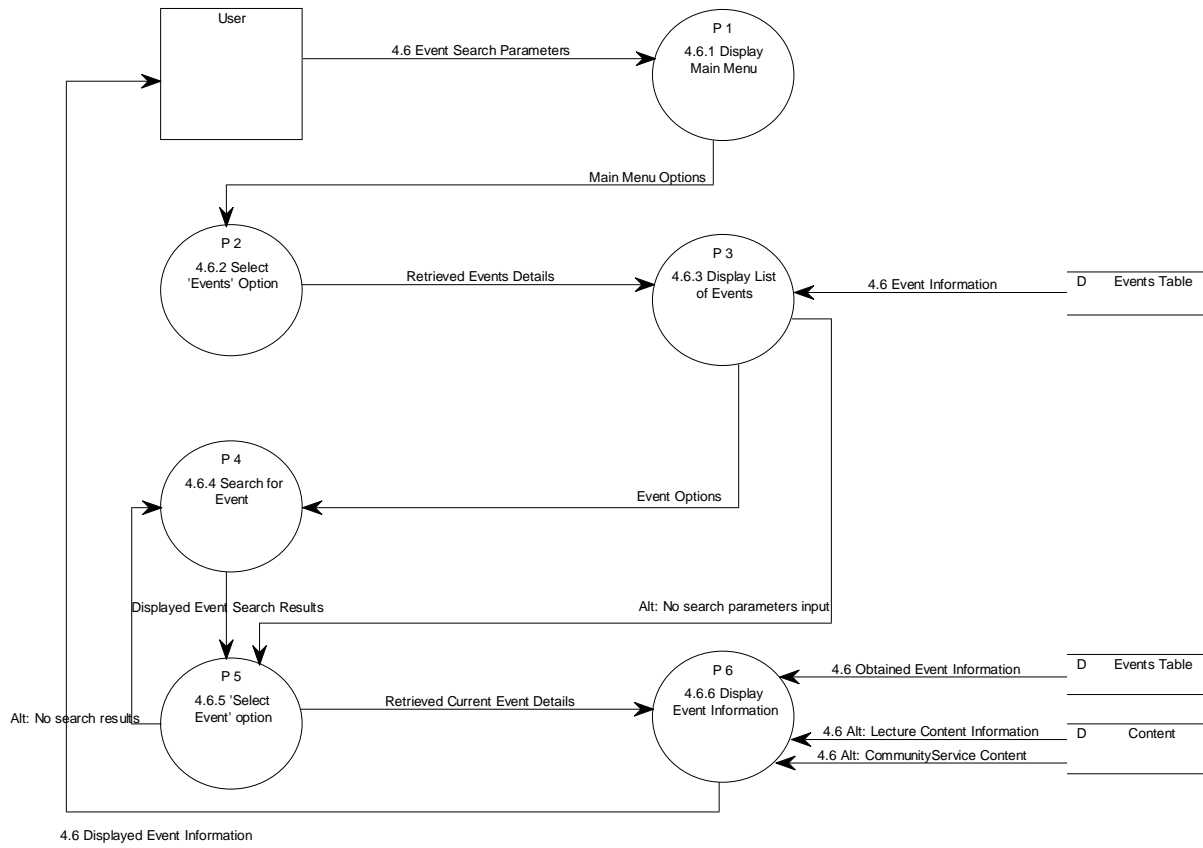


FIGURE 132- PRIMITIVE LEVEL DATA FLOW DIAGRAM 4.6 SEARCH EVENT



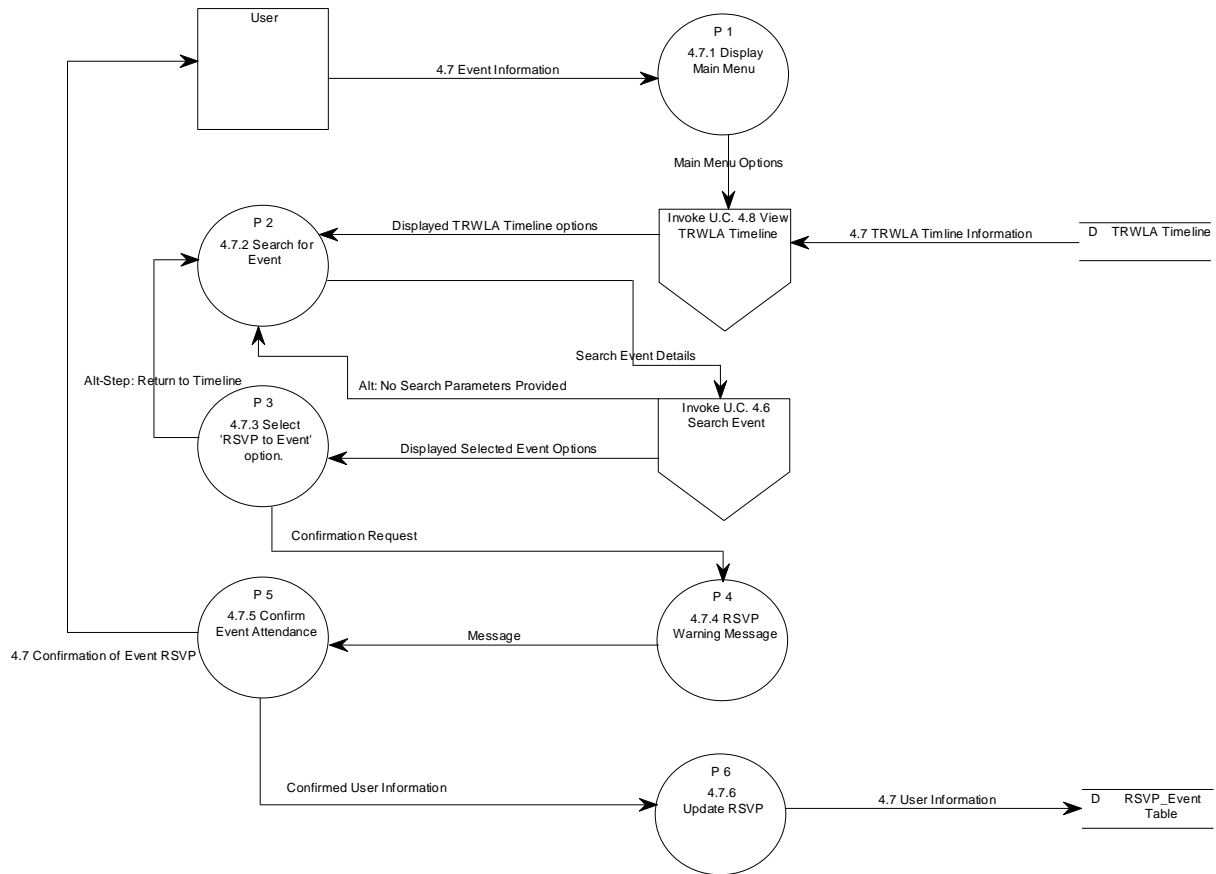


FIGURE 133- PRIMITIVE LEVEL DATA FLOW DIAGRAM 4.7 RSVP TO AN EVENT

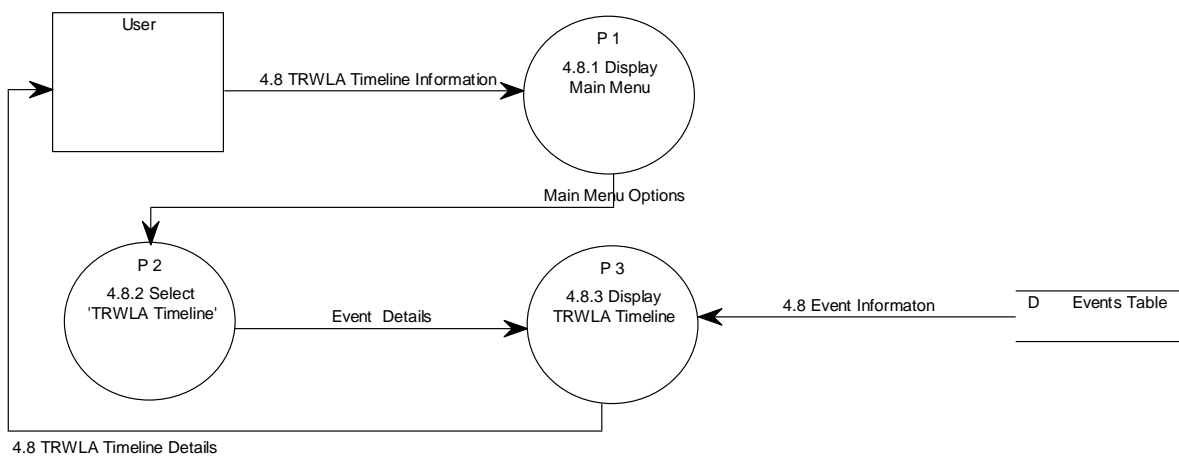


FIGURE 134- PRIMITIVE LEVEL DATA FLOW DIAGRAM 4.8 VIEW TRWLA TIMELINE

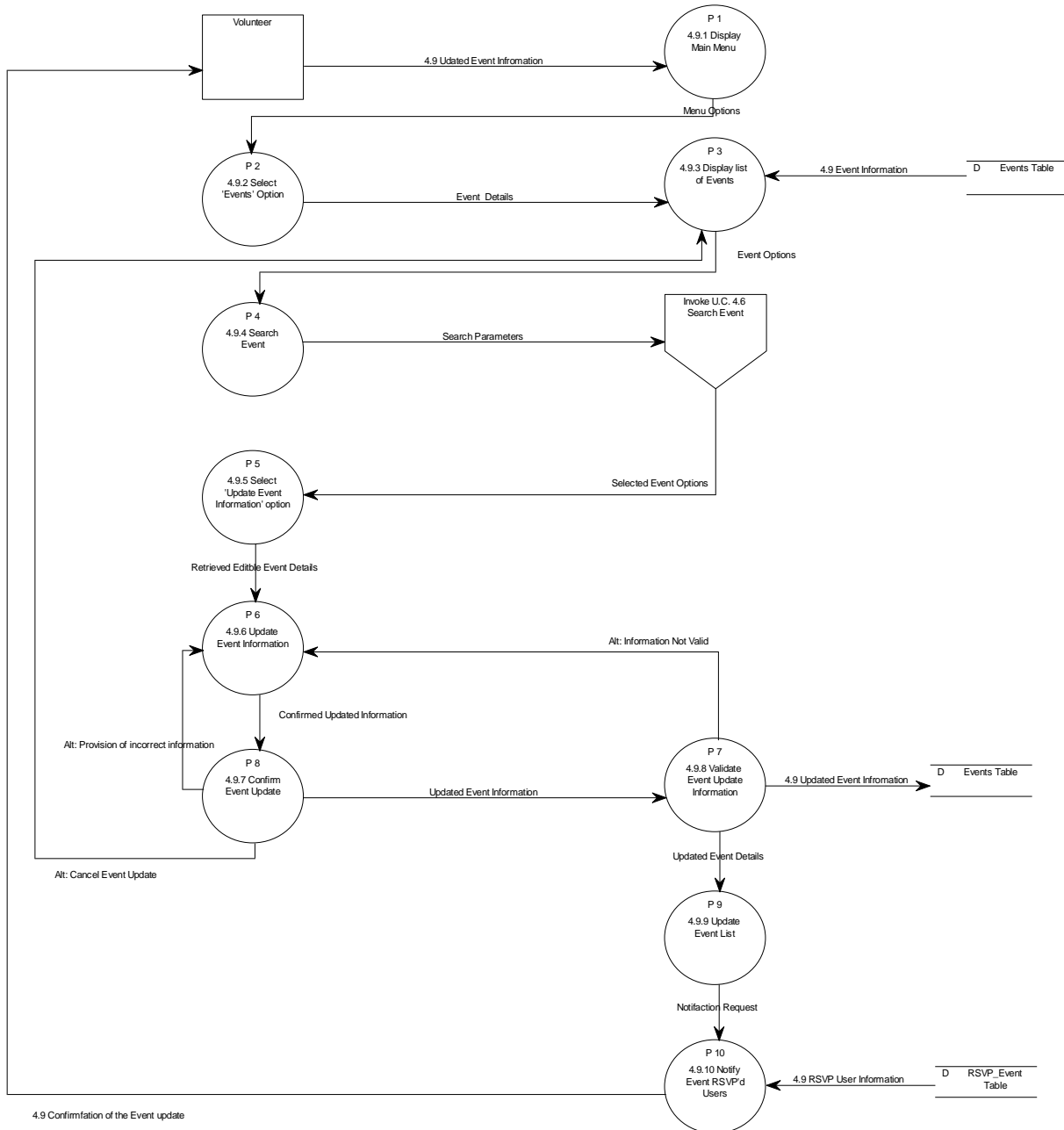


FIGURE 135- PRIMITIVE LEVEL DATA FLOW DIAGRAM 4.9 UPDATE EVENT INFORMATION

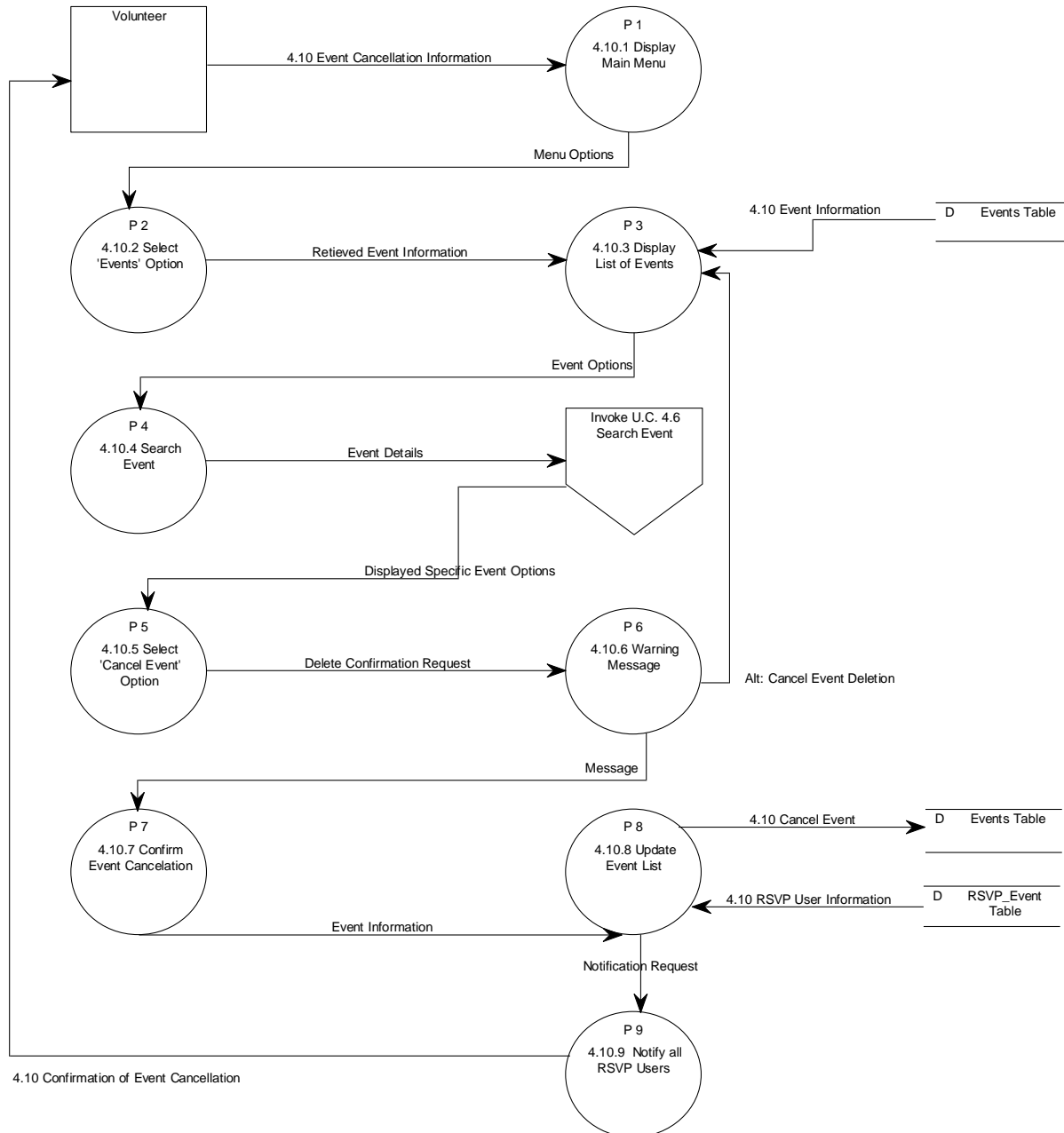


FIGURE 136- PRIMITIVE LEVEL DATA FLOW DIAGRAM 4.10 CANCEL EVENT

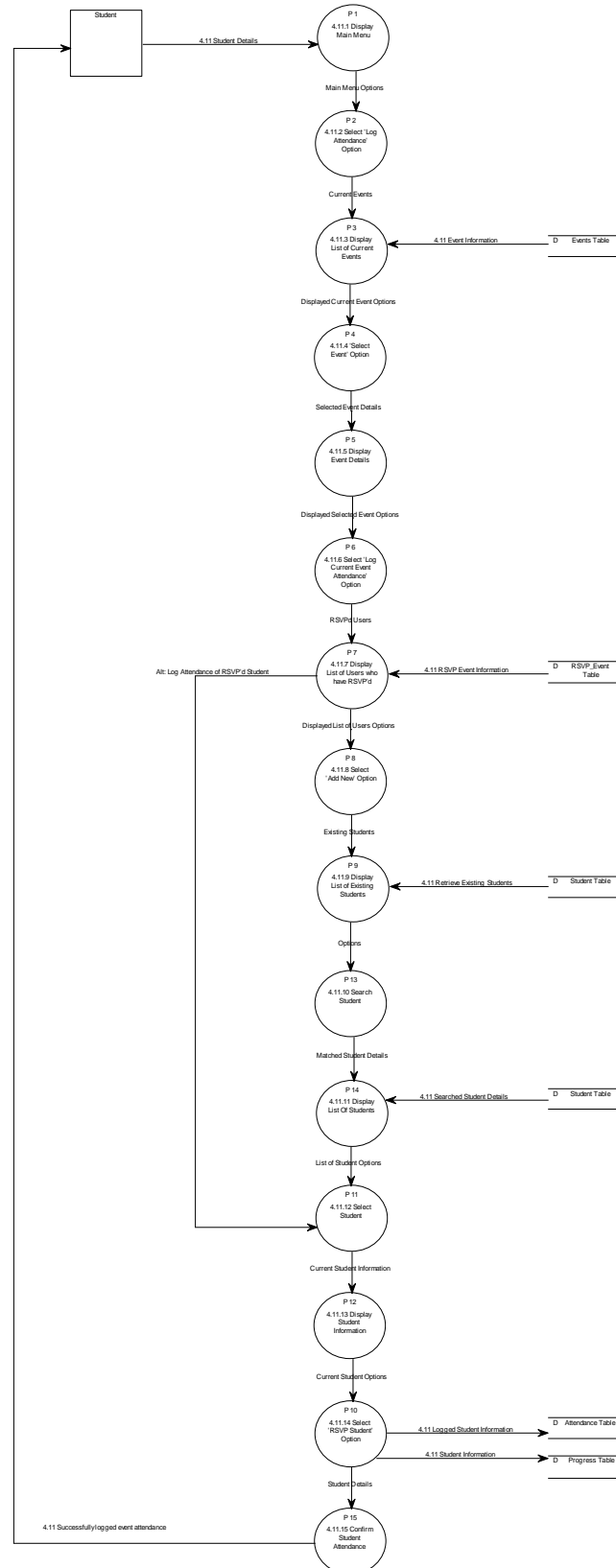


FIGURE 137- PRIMITIVE LEVEL DATA FLOW DIAGRAM 4.11 LOG EVENT ATTENDANCE

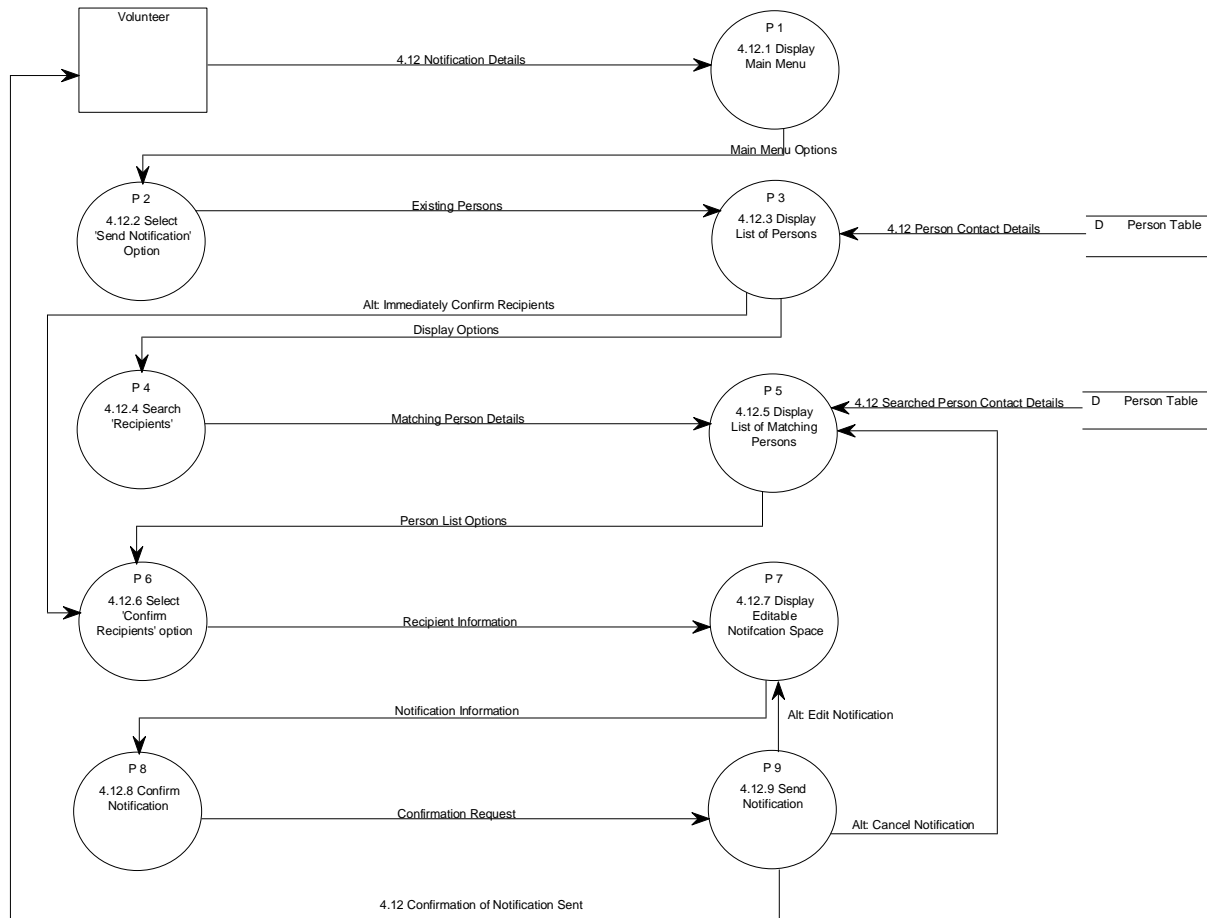


FIGURE 138- PRIMITIVE LEVEL DATA FLOW DIAGRAM 4.12 SEND NOTIFICATION

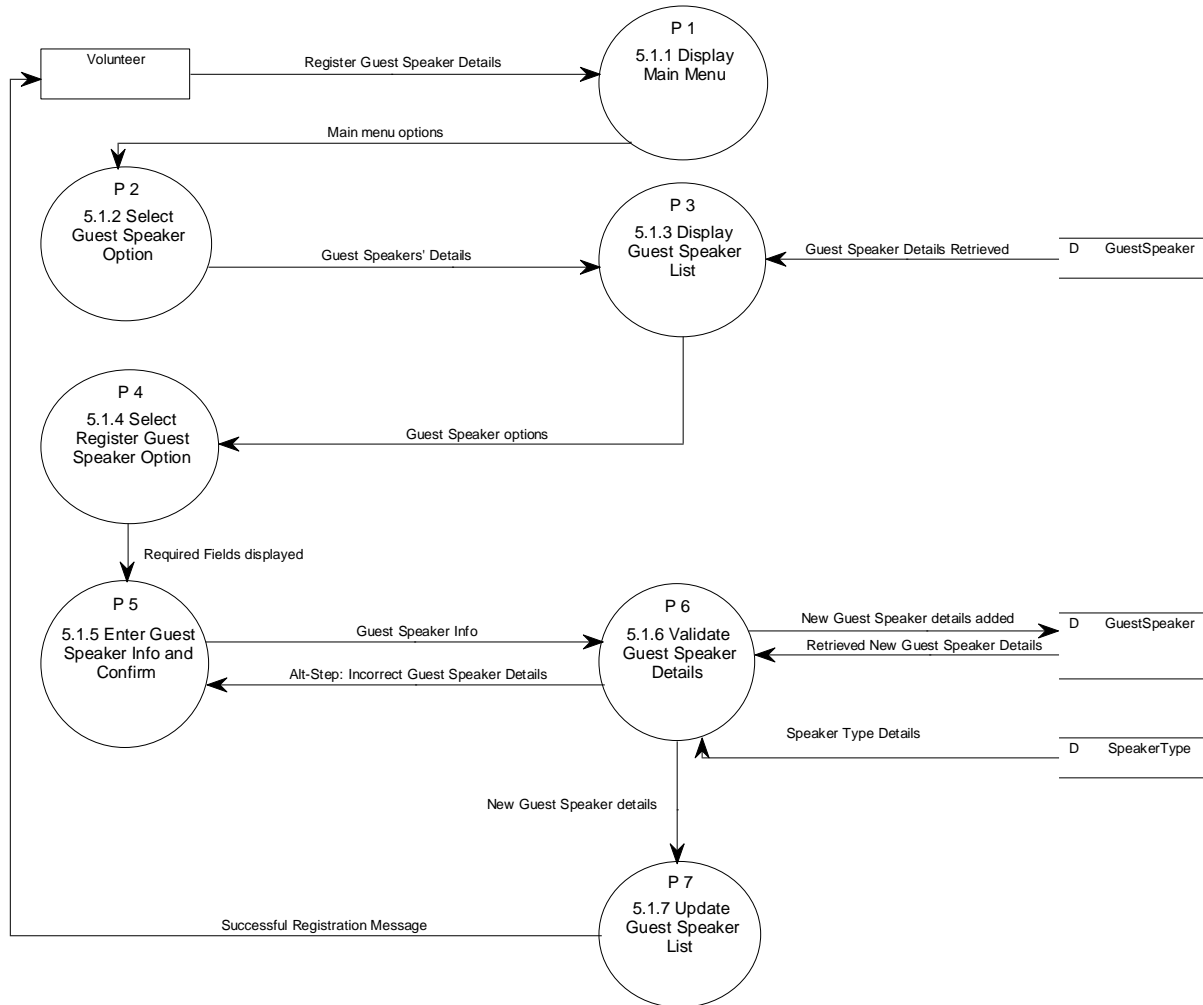


FIGURE 139- PRIMITIVE LEVEL DATA FLOW DIAGRAM 5.1 REGISTER GUEST SPEAKER

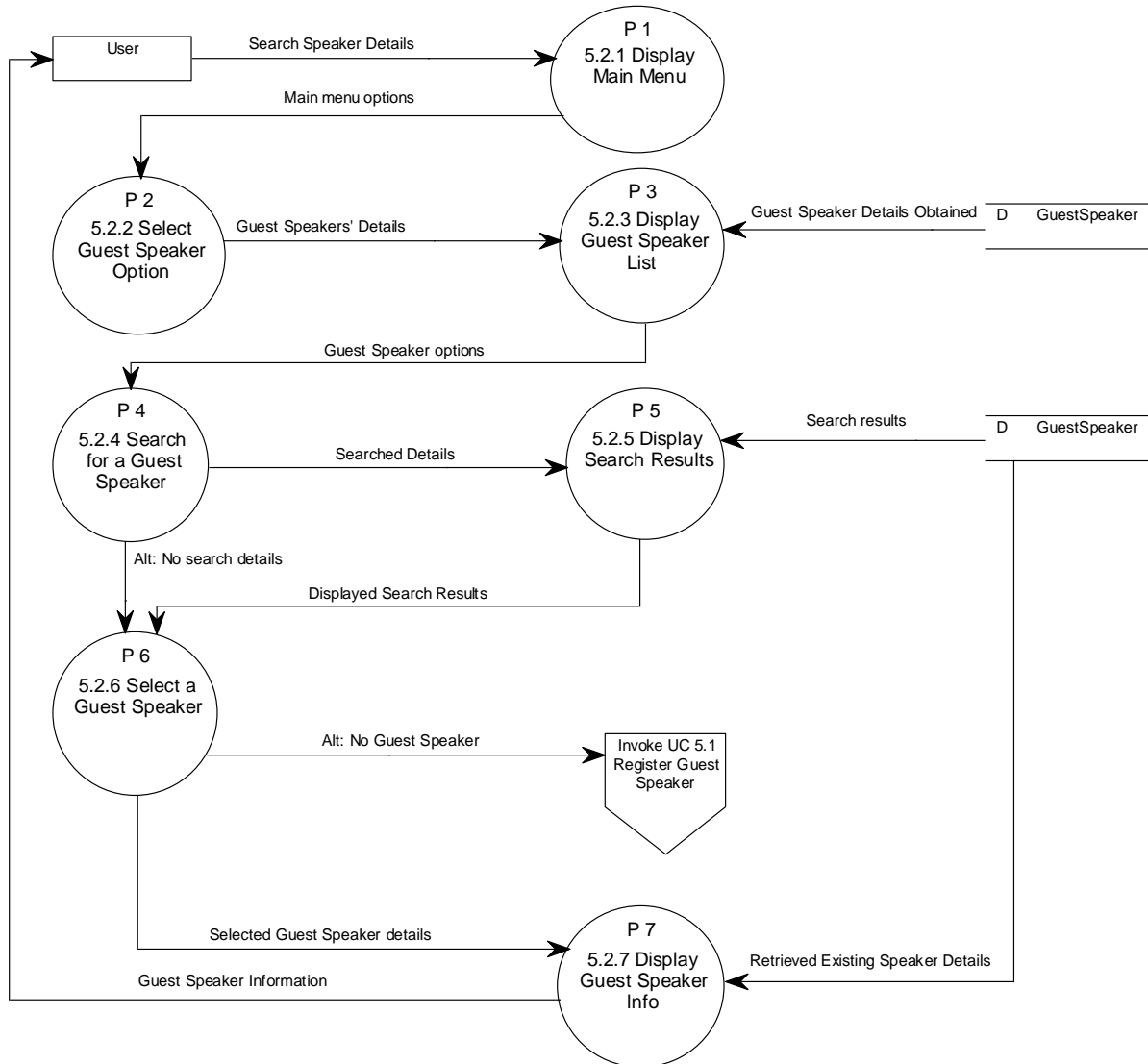


FIGURE 140- PRIMITIVE LEVEL DATA FLOW DIAGRAM 5.2 SEARCH GUEST SPEAKER

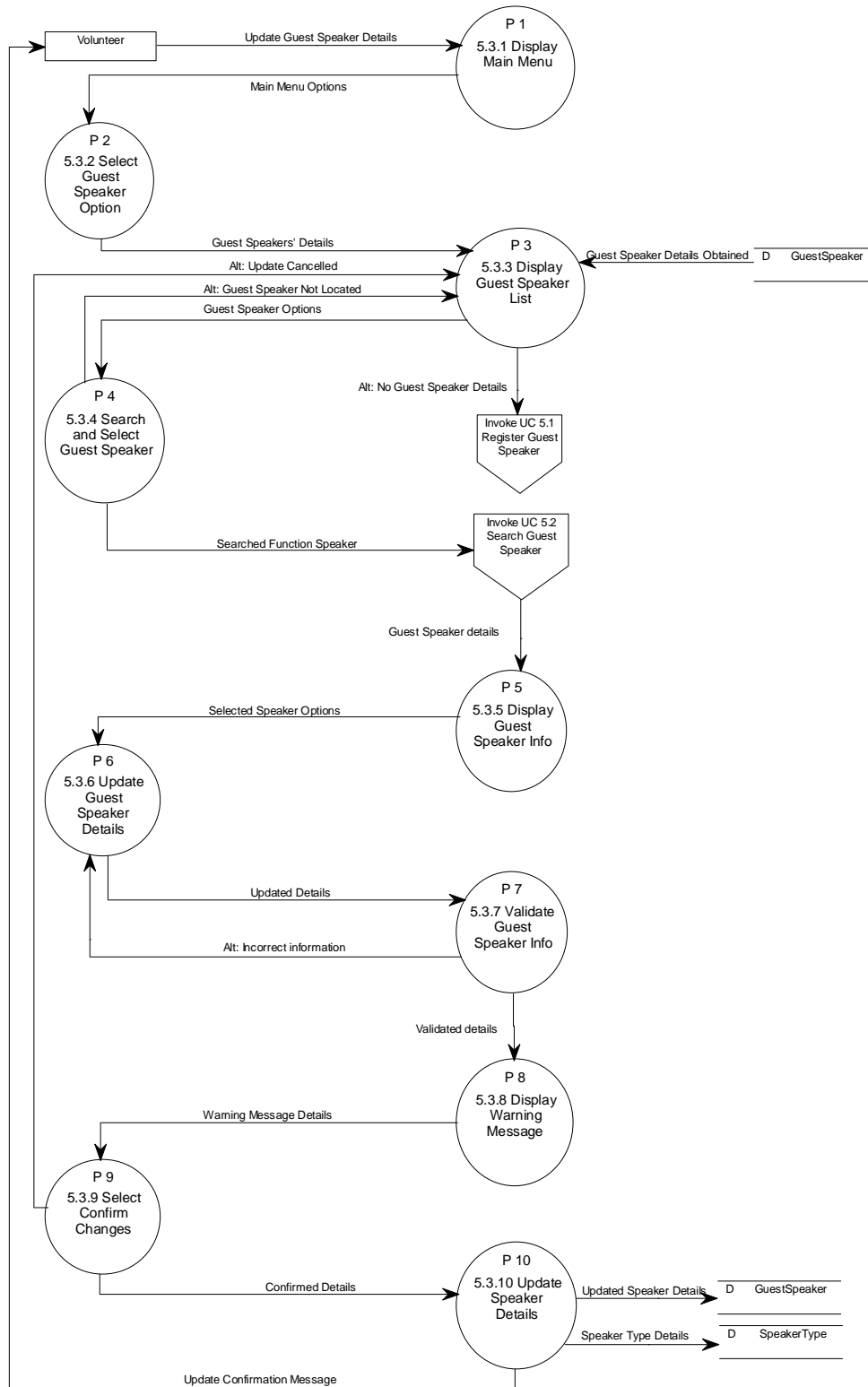


FIGURE 141- PRIMITIVE LEVEL DATA FLOW DIAGRAM 5.3 UPDATE GUEST SPEAKER



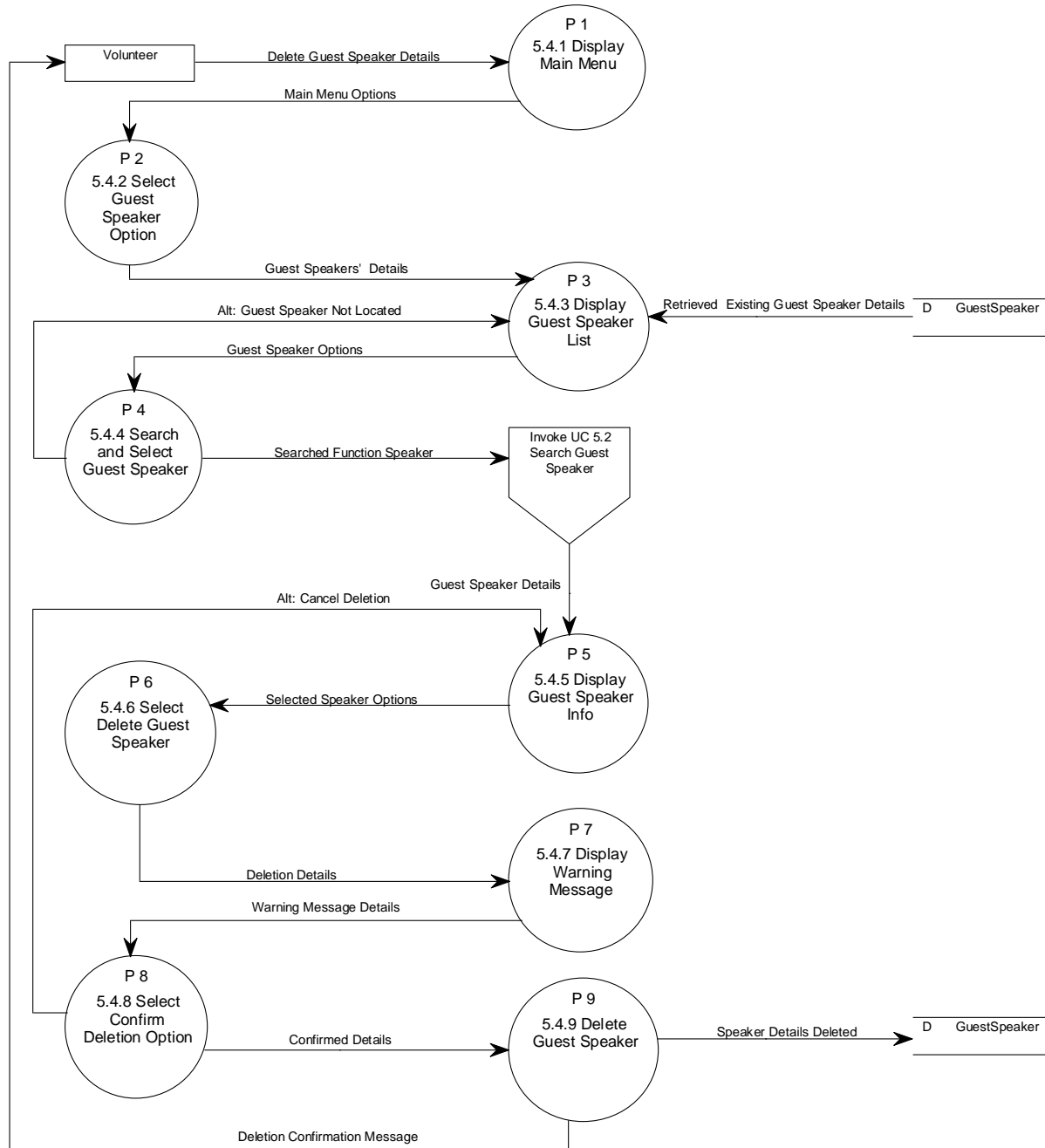


FIGURE 142- PRIMITIVE LEVEL DATA FLOW DIAGRAM 5.4 DELETE GUEST SPEAKER

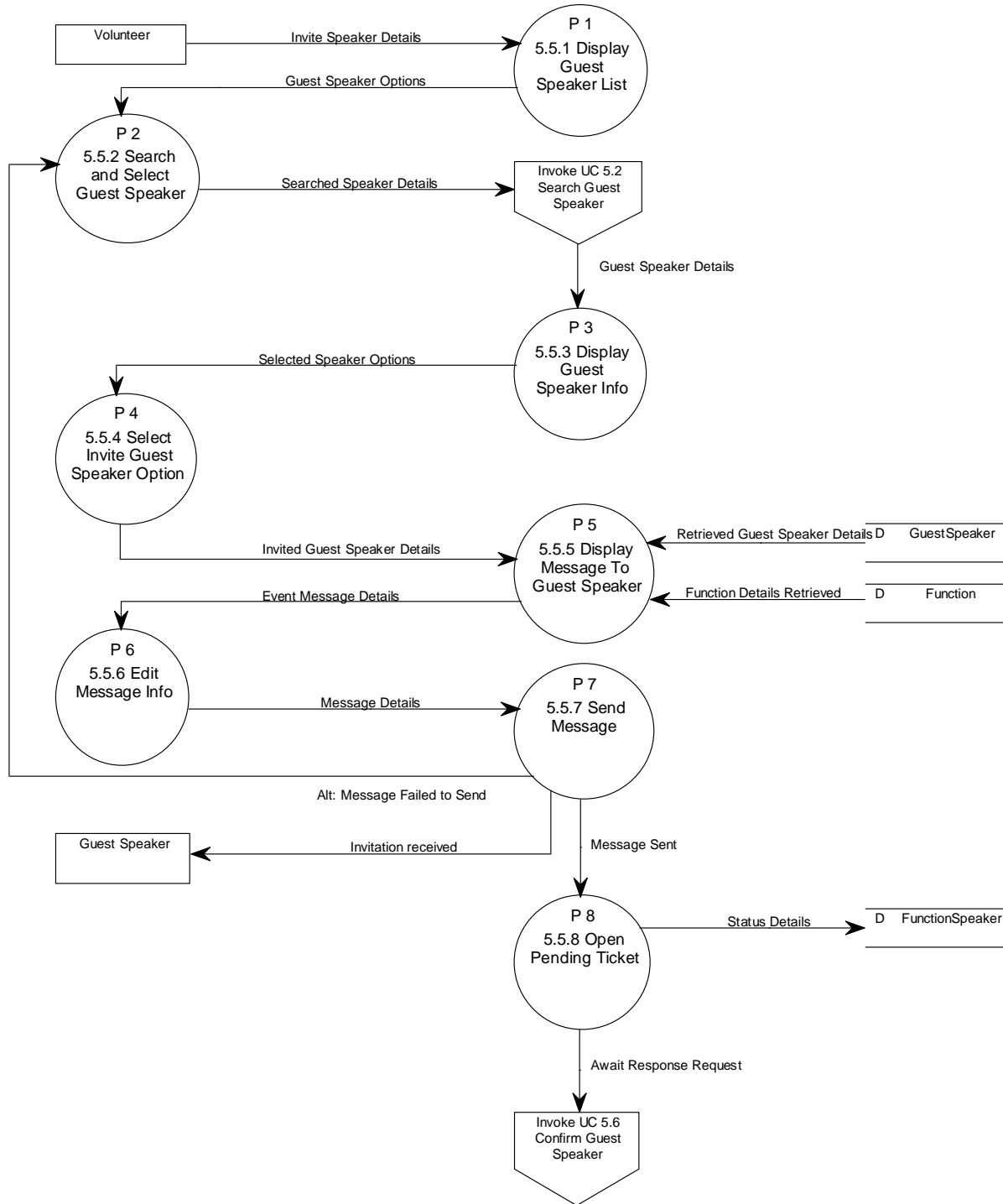


FIGURE 143- PRIMITIVE LEVEL DATA FLOW DIAGRAM 5.5 INVITE GUEST SPEAKER

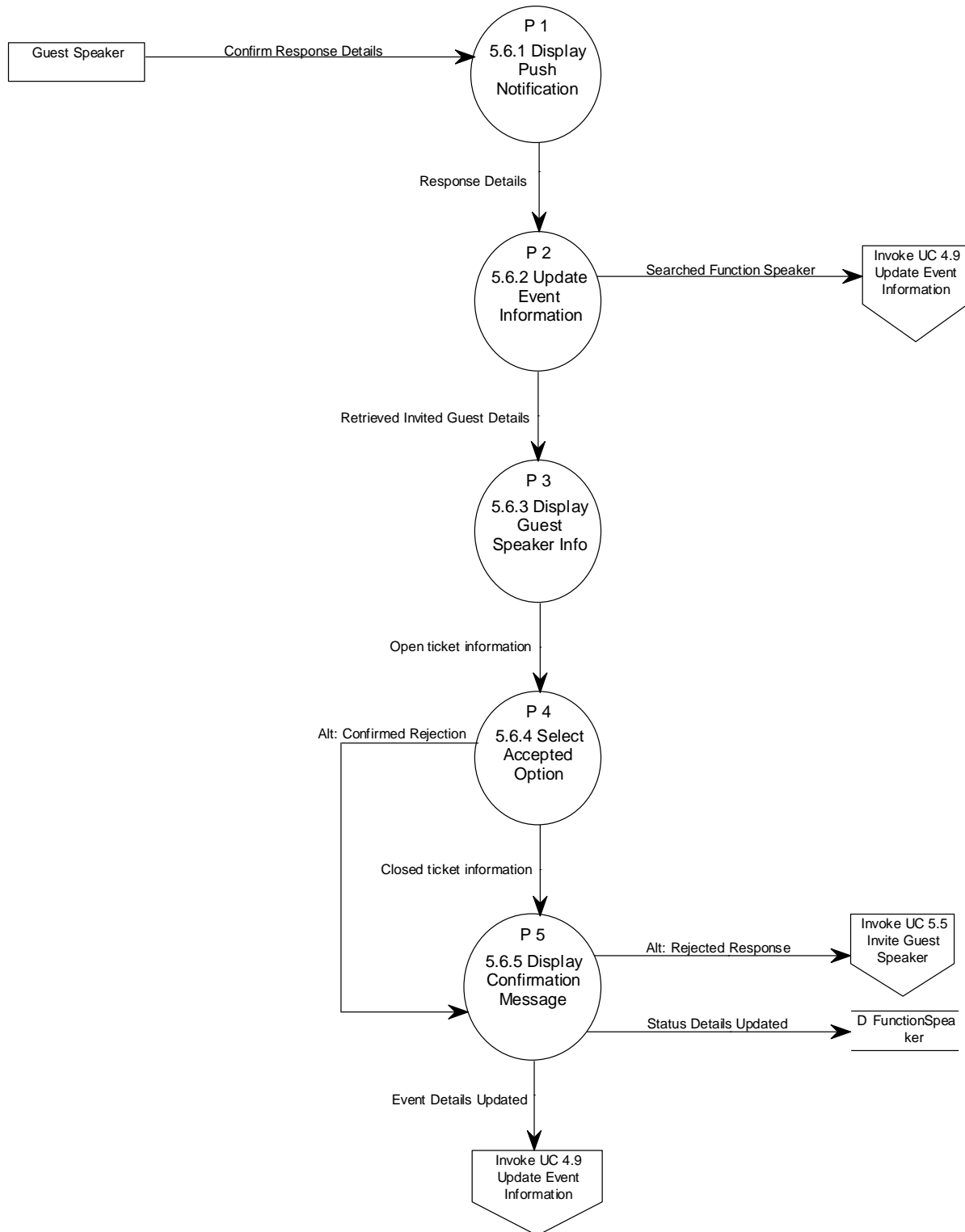


FIGURE 144- PRIMITIVE LEVEL DATA FLOW DIAGRAM 5.6 CONFIRM GUEST SPEAKER

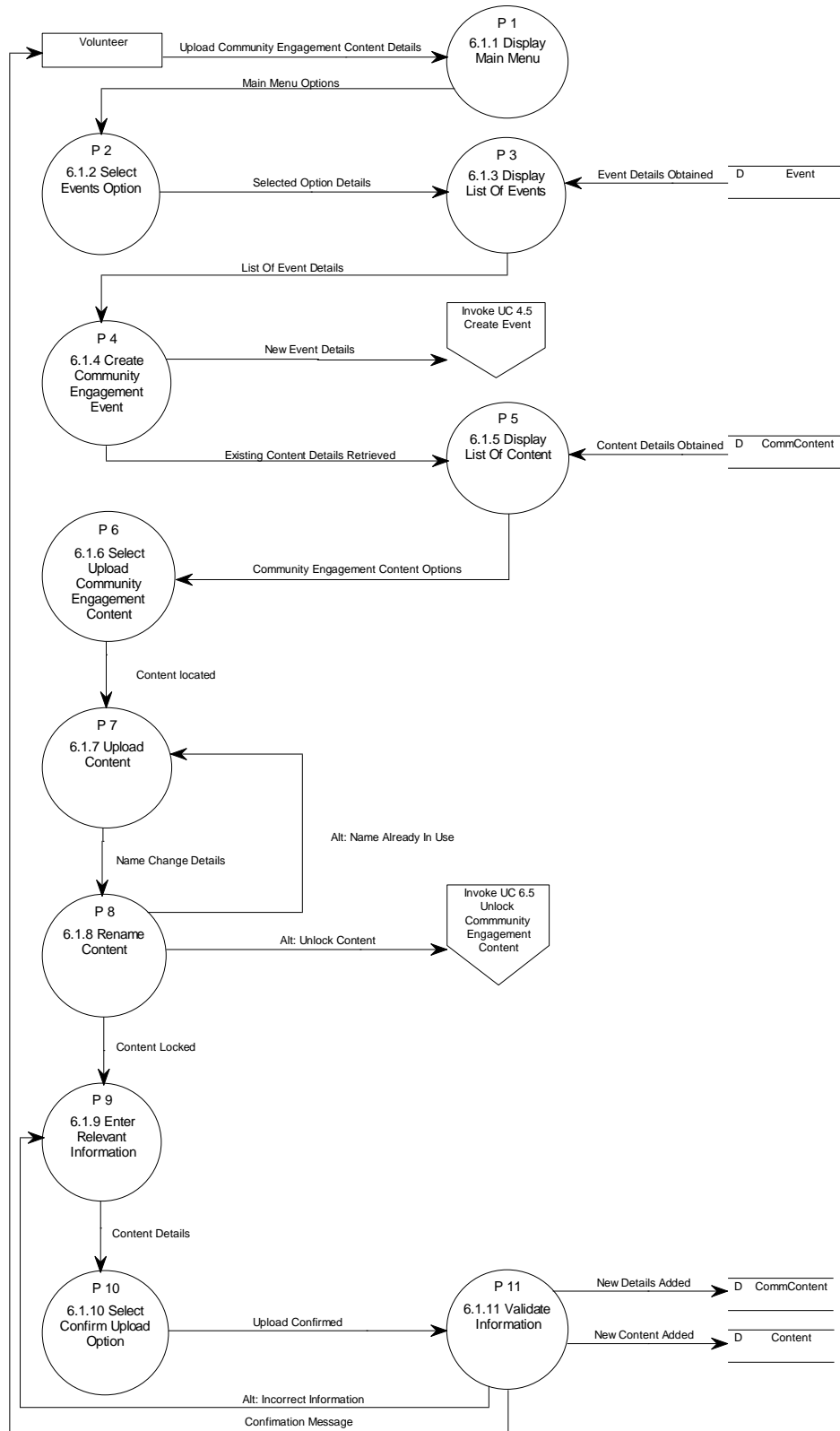


FIGURE 145- PRIMITIVE LEVEL DATA FLOW DIAGRAM 6.1 UPLOAD COMMUNITY ENGAGEMENT CONTENT

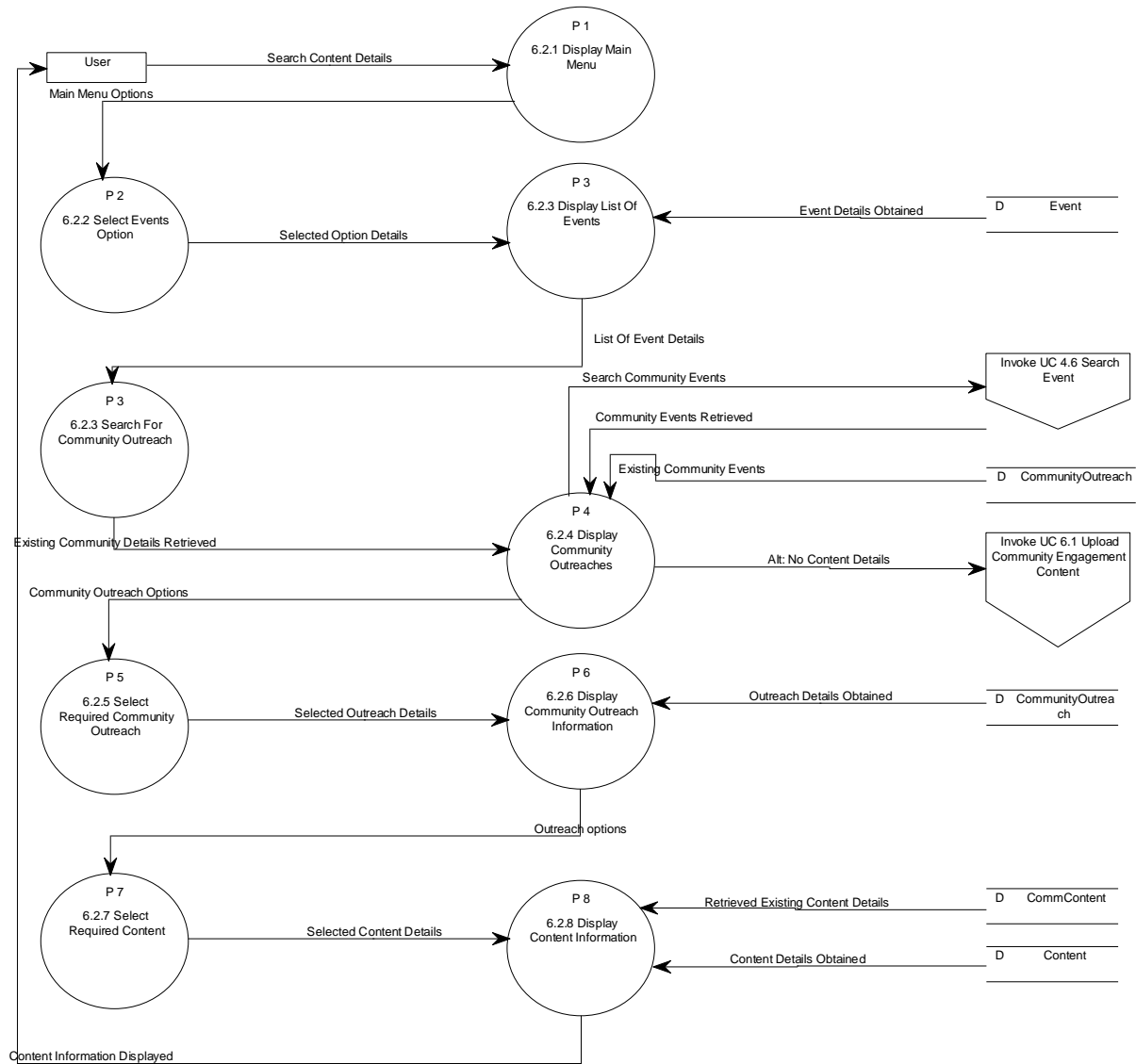


FIGURE 146- PRIMITIVE LEVEL DATA FLOW DIAGRAM 6.2 SEARCH COMMUNITY ENGAGEMENT CONTENT

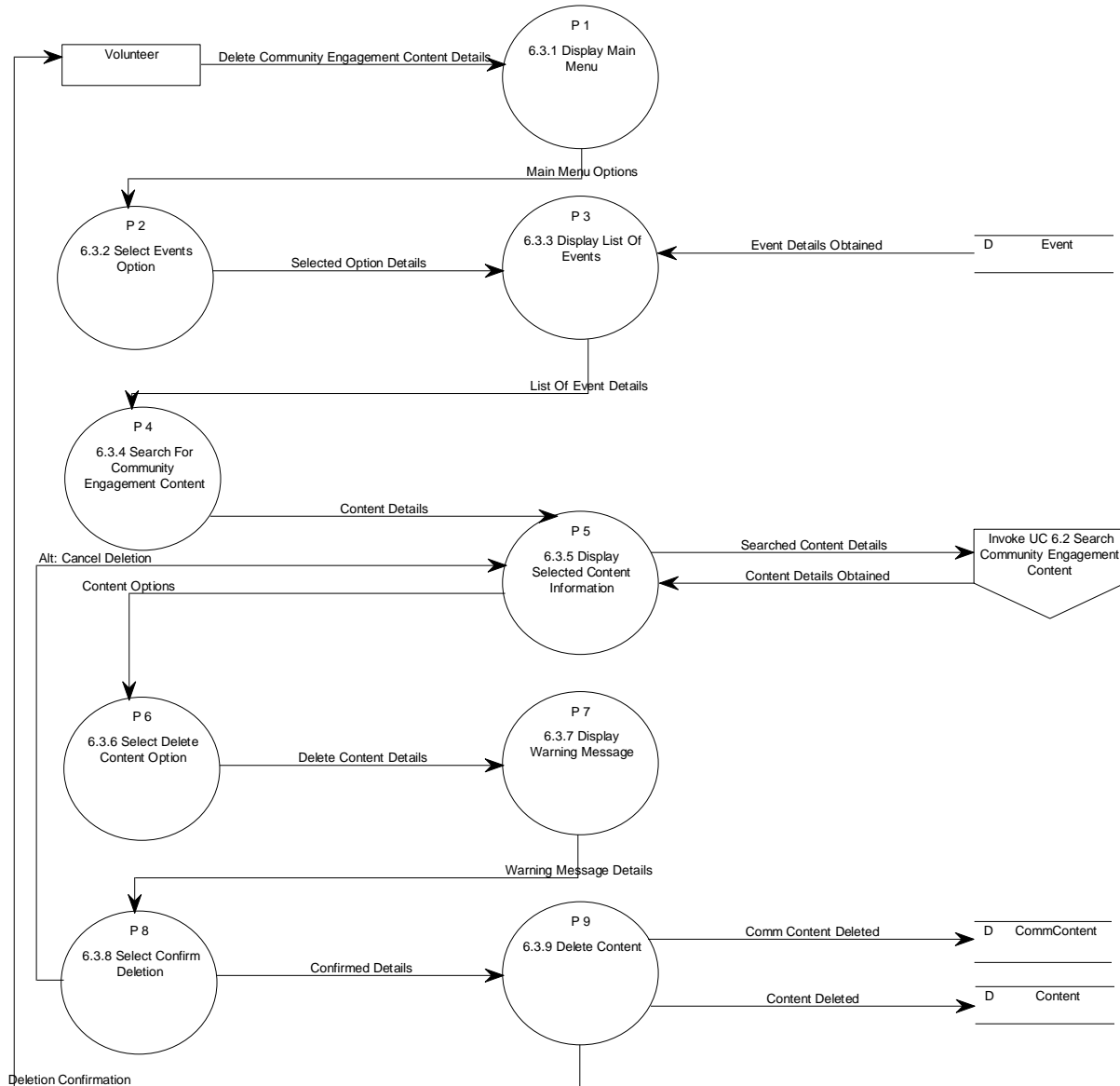


FIGURE 147- PRIMITIVE LEVEL DATA FLOW DIAGRAM 6.3 DELETE COMMUNITY ENGAGEMENT CONTENT

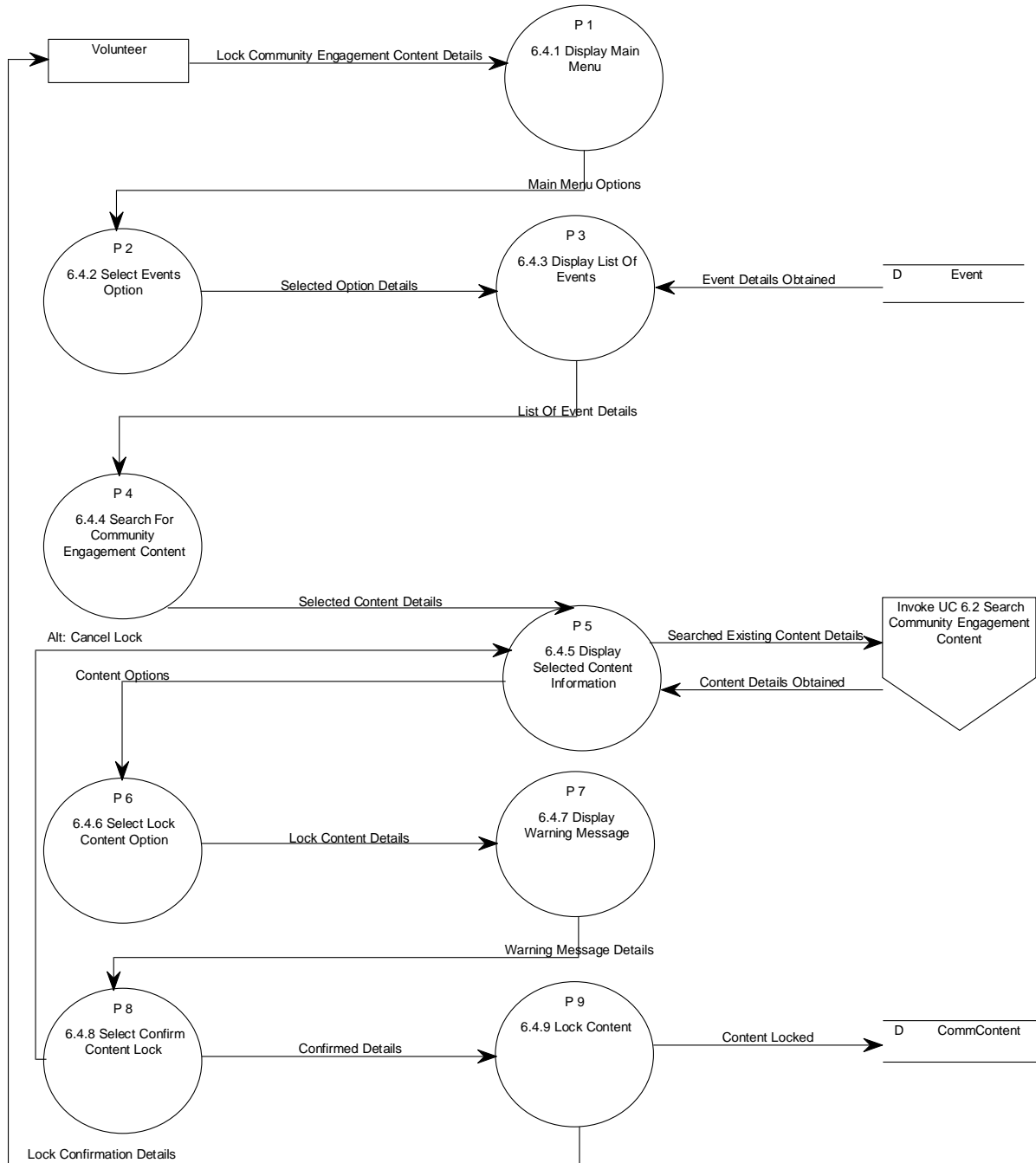


FIGURE 148- PRIMITIVE LEVEL DATA FLOW DIAGRAM 6.4 LOCK COMMUNITY ENGAGEMENT CONTENT



FIGURE 149- PRIMITIVE LEVEL DATA FLOW DIAGRAM 6.5 UNLOCK COMMUNITY ENGAGEMENT CONTENT



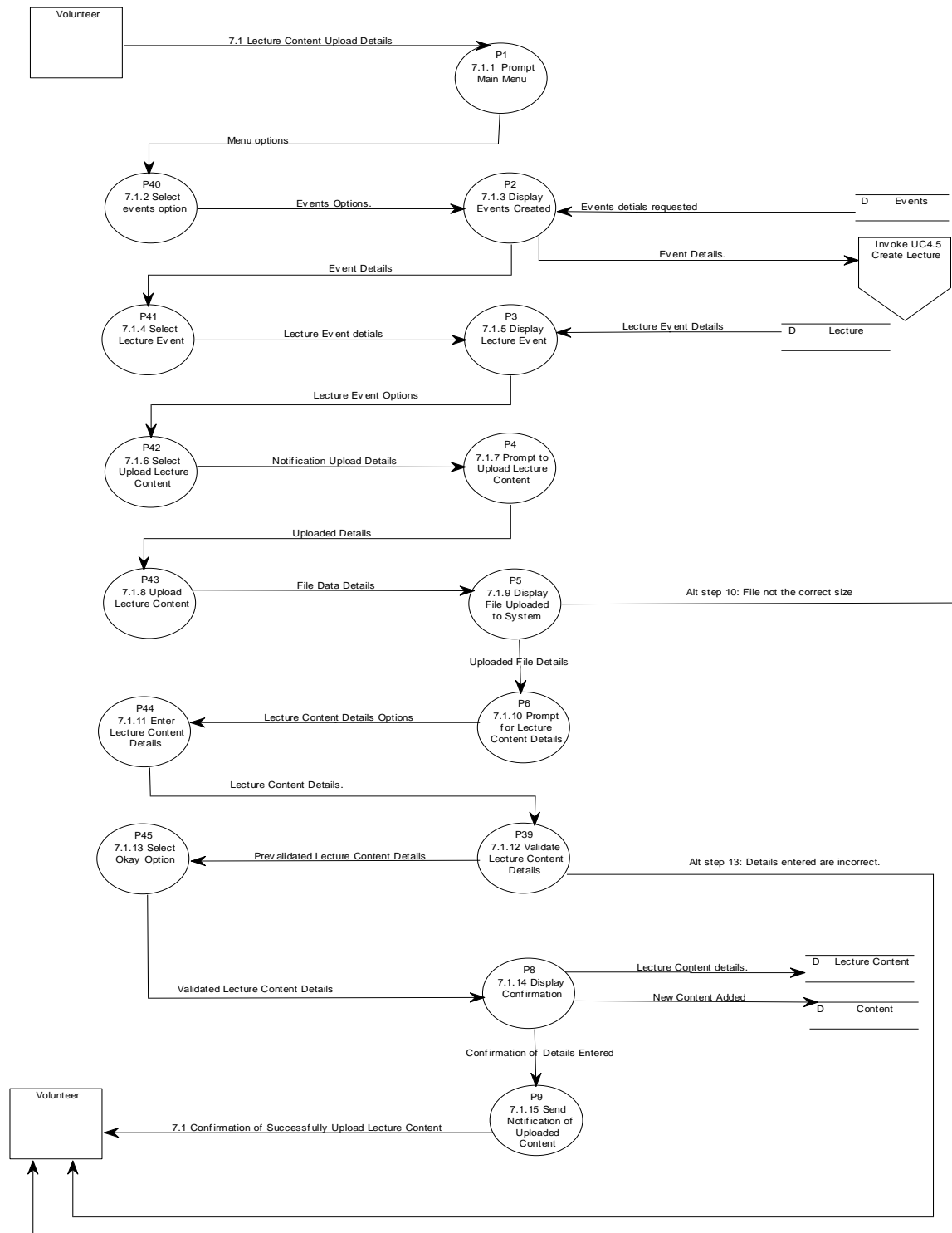


FIGURE 150- PRIMITIVE LEVEL DATA FLOW DIAGRAM 7.1 UPLOAD LECTURE CONTENT

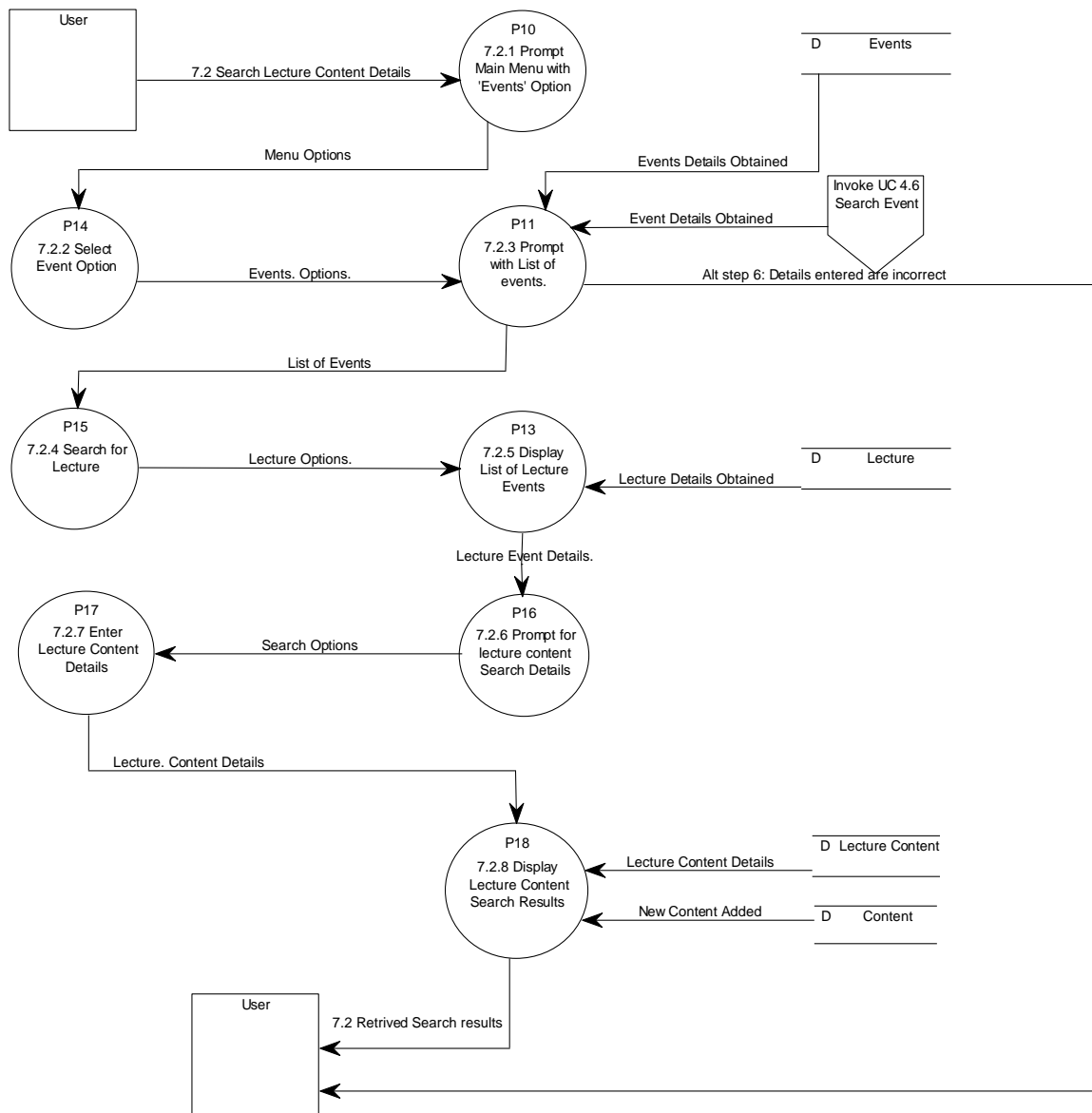


FIGURE 151- PRIMITIVE LEVEL DATA FLOW DIAGRAM 7.2 UPLOAD LECTURE CONTENT

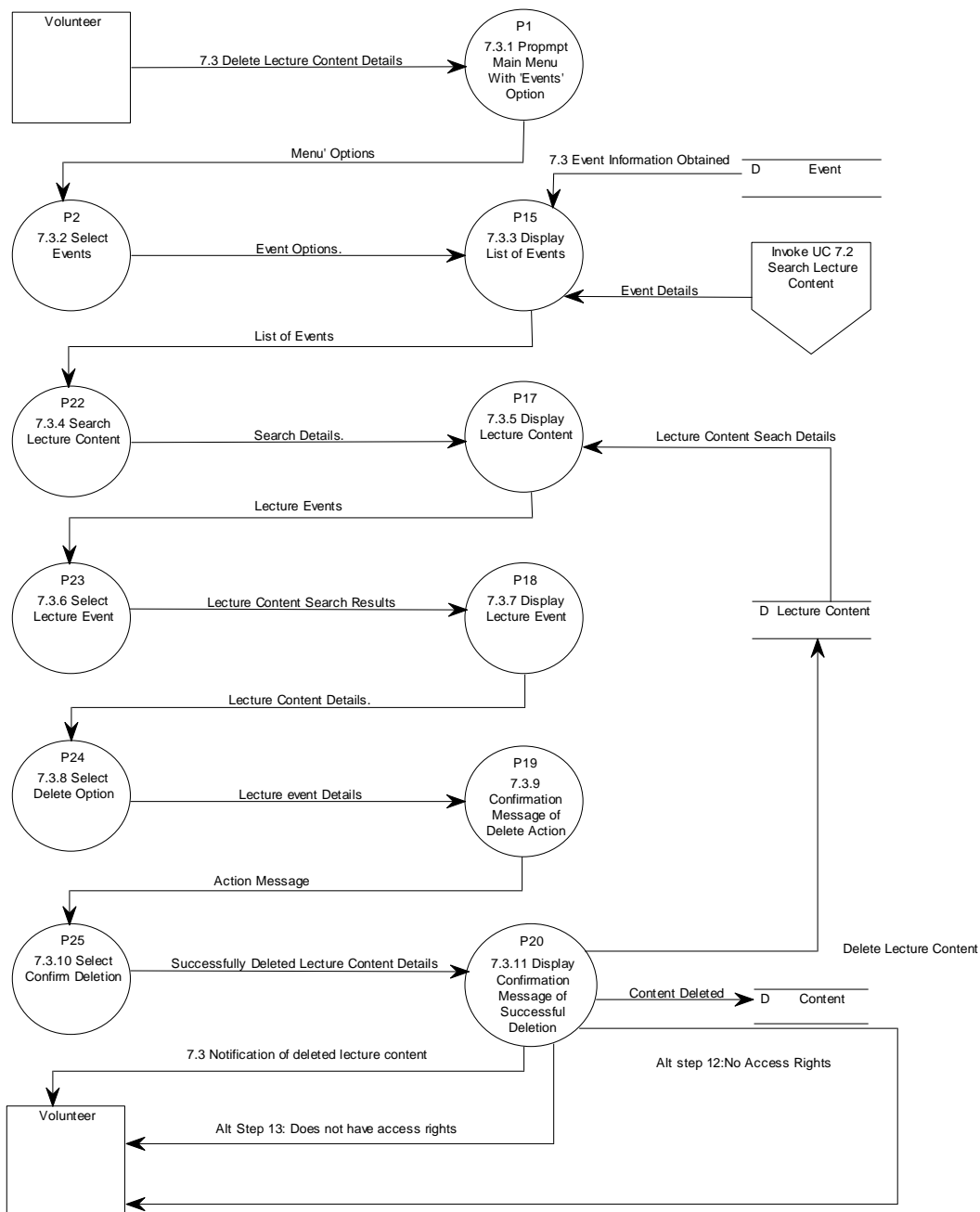


FIGURE 152- PRIMITIVE LEVEL DATA FLOW DIAGRAM 7.3 DELETE LECTURE CONTENT

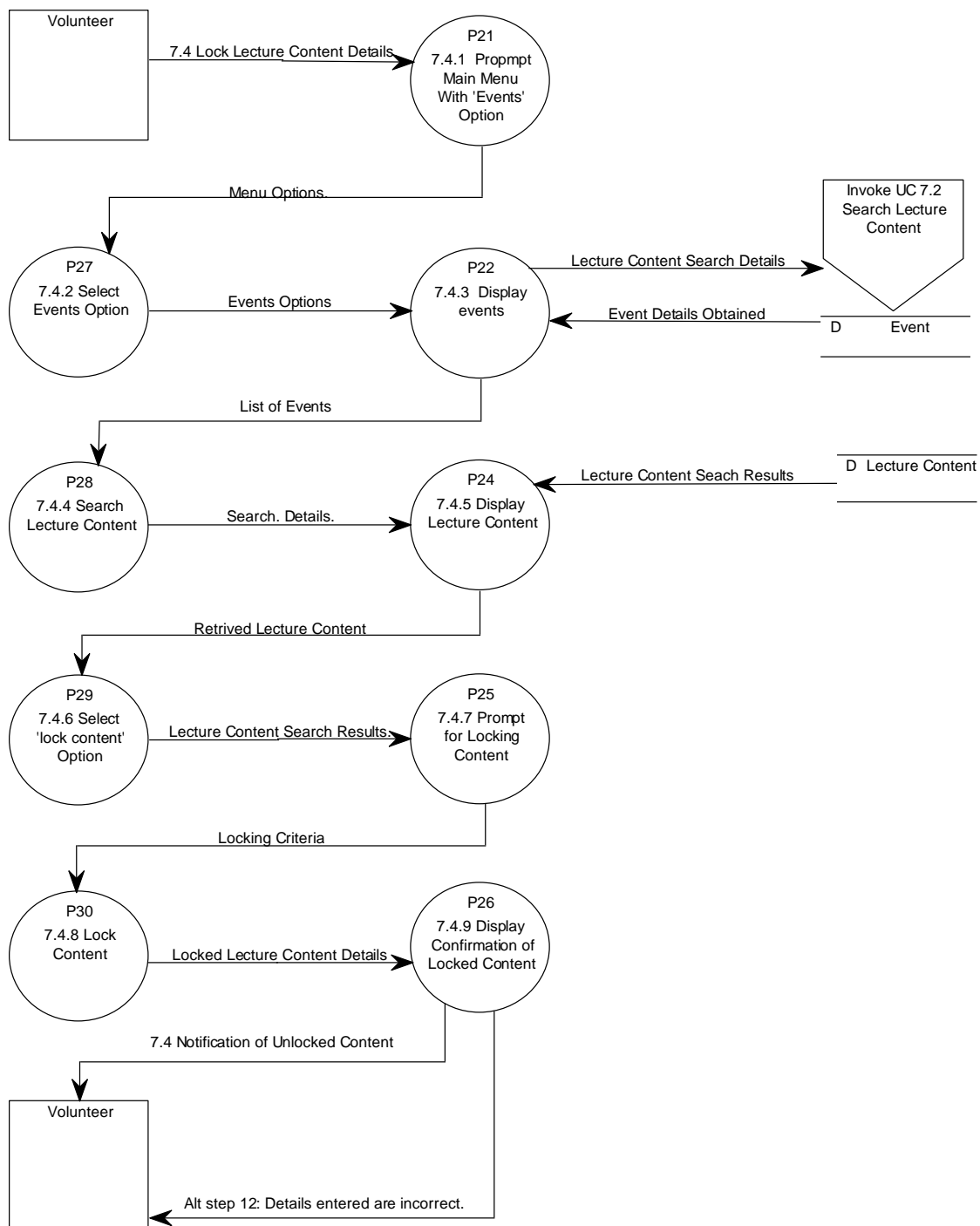


FIGURE 153- PRIMITIVE LEVEL DATA FLOW DIAGRAM 7.4 LOCK LECTURE CONTENT

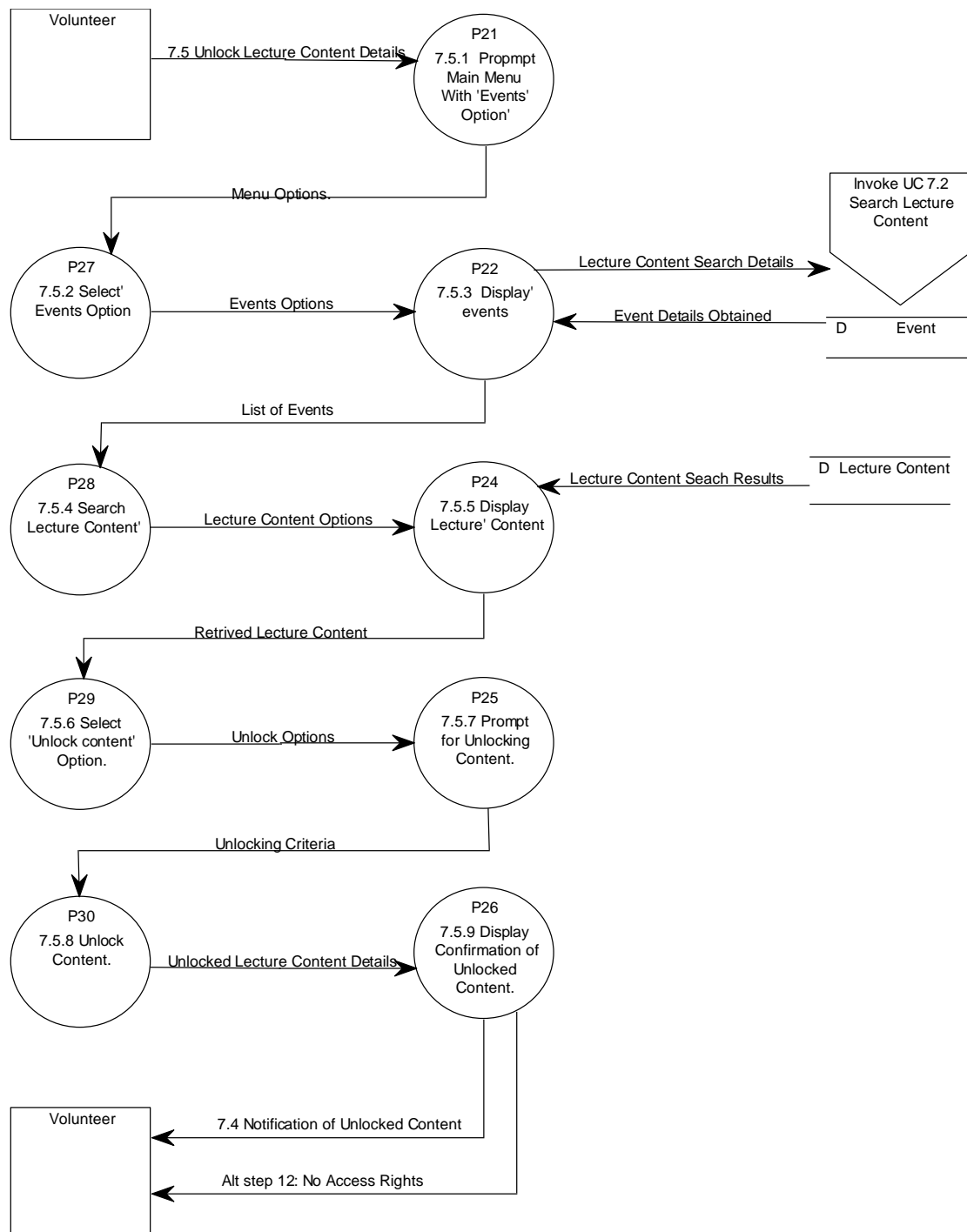


FIGURE 154- PRIMITIVE LEVEL DATA FLOW DIAGRAM 7.5 UNLOCK LECTURE CONTENT

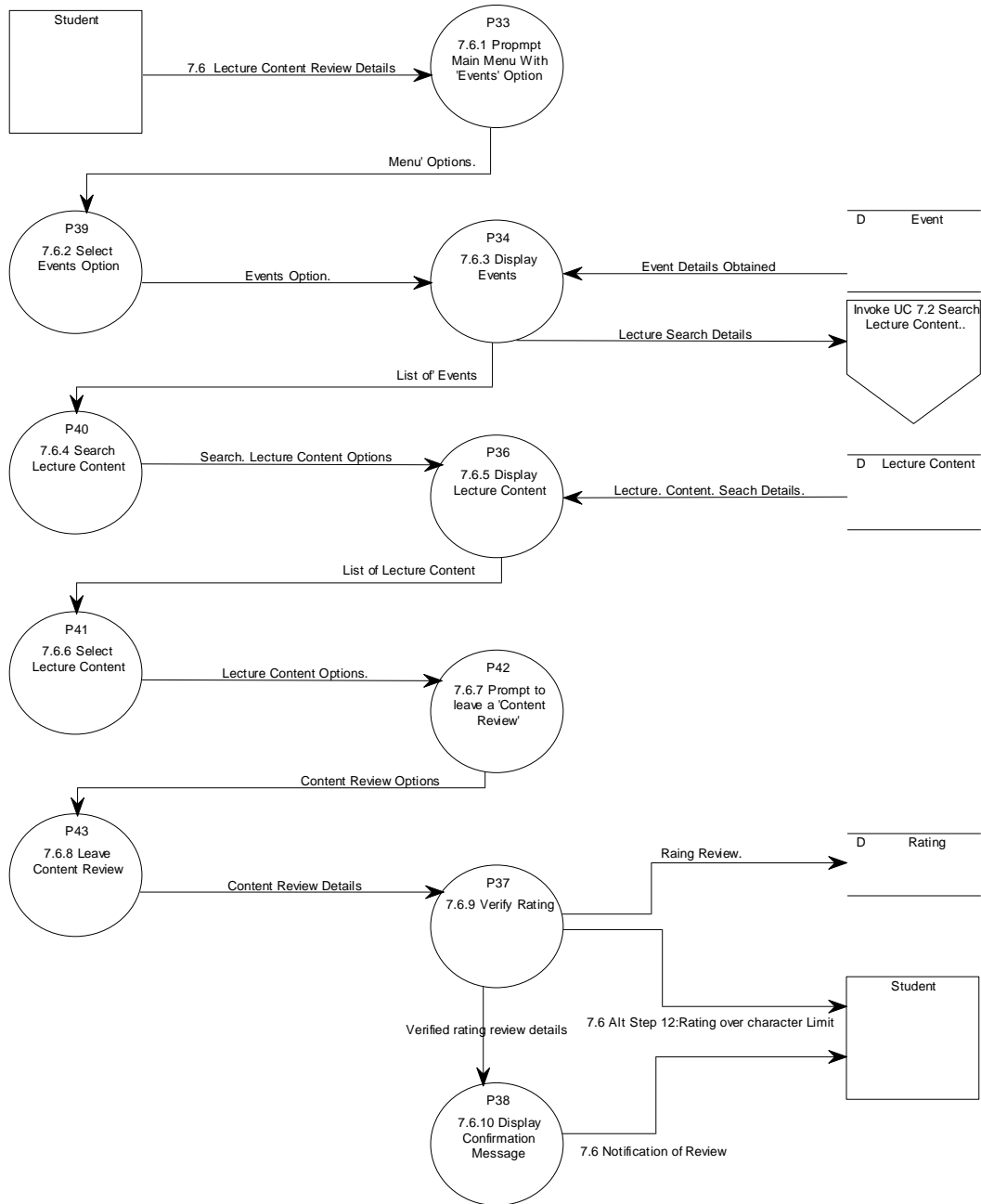


FIGURE 155- PRIMITIVE LEVEL DATA FLOW DIAGRAM 7.6 CONTENT REVIEW

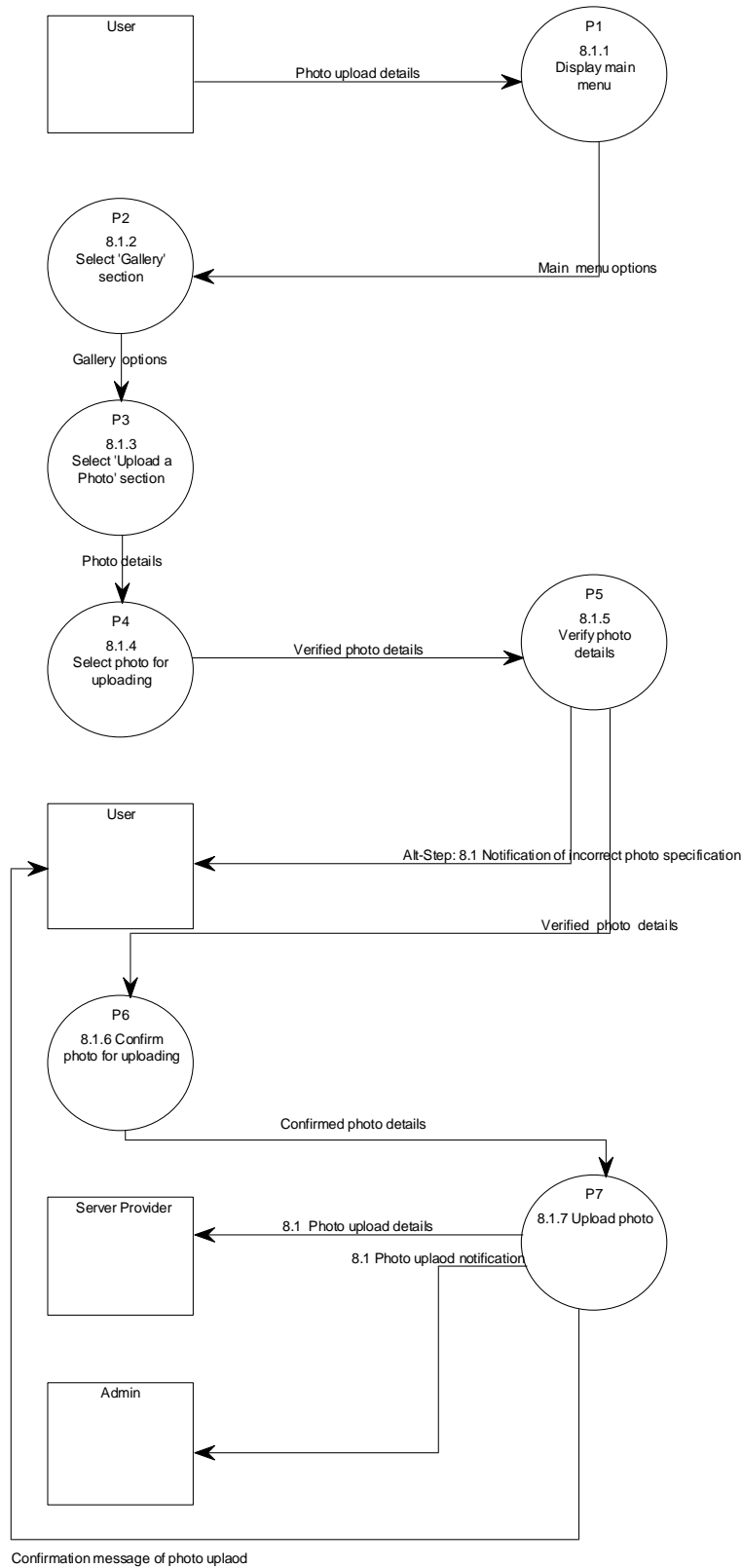


FIGURE 156- PRIMITIVE LEVEL DATA FLOW DIAGRAM 8.1 UPLOAD PHOTO

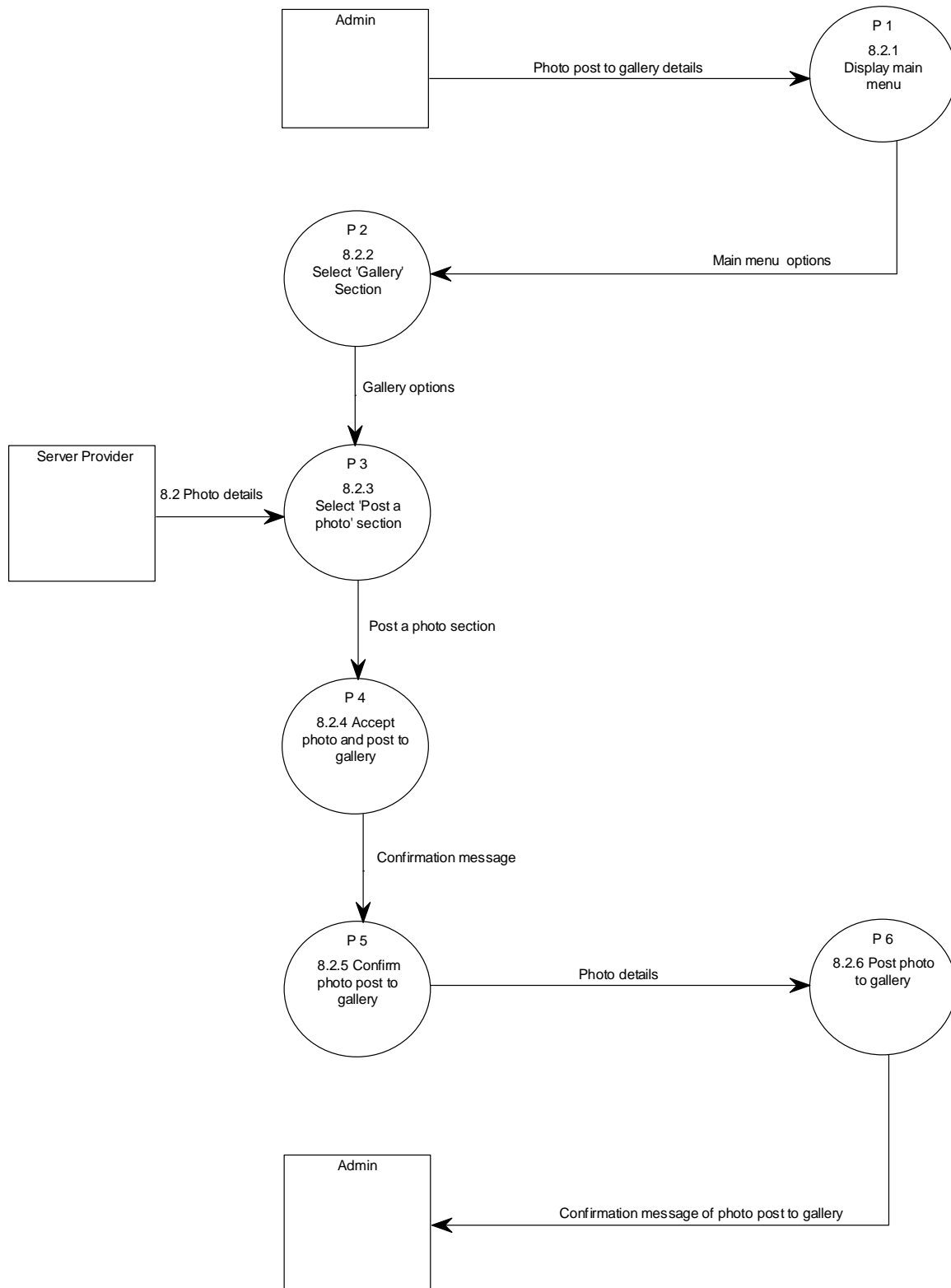


FIGURE 157- PRIMITIVE LEVEL DATA FLOW DIAGRAM 8.2 POST PHOTO



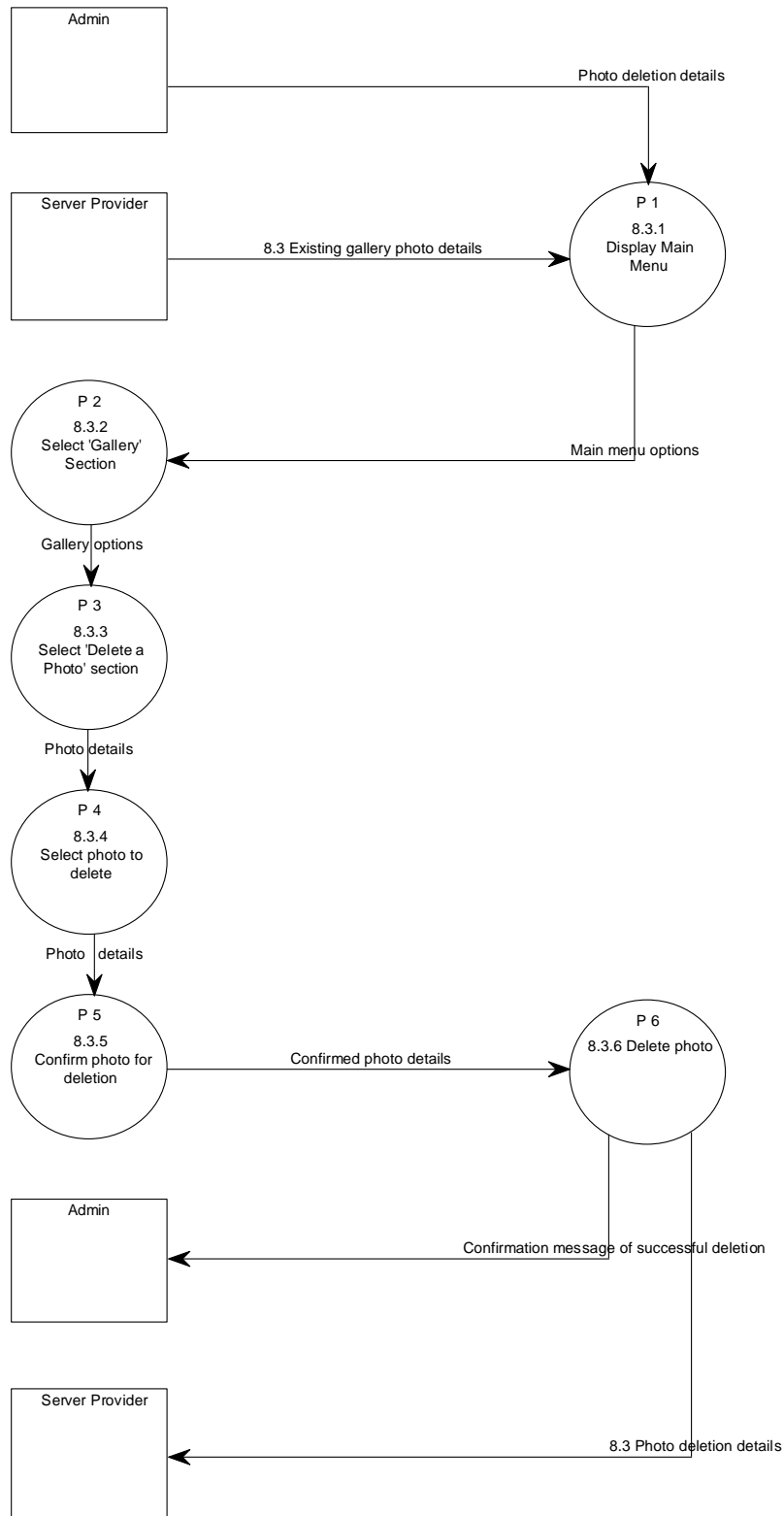


FIGURE 158- PRIMITIVE LEVEL DATA FLOW DIAGRAM 8.3 DELETE PHOTO

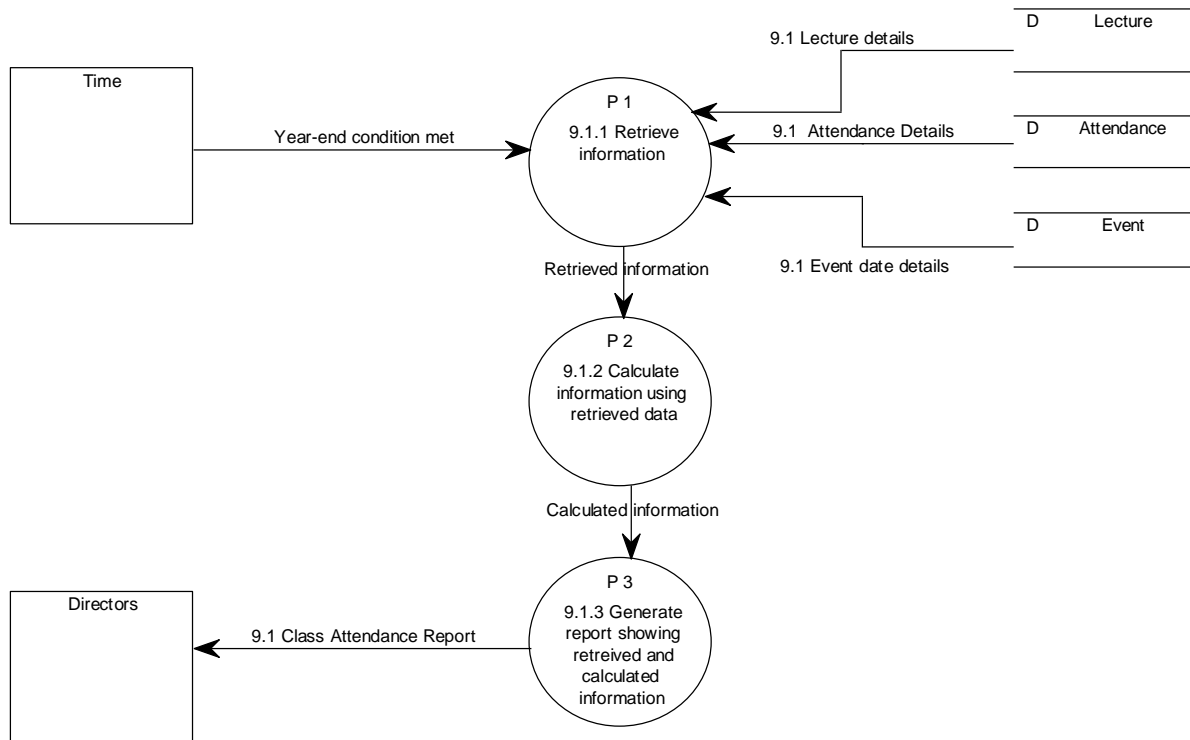


FIGURE 159- PRIMITIVE LEVEL DATA FLOW DIAGRAM 9.1 GENERATE CLASS ATTENDANCE REPORT

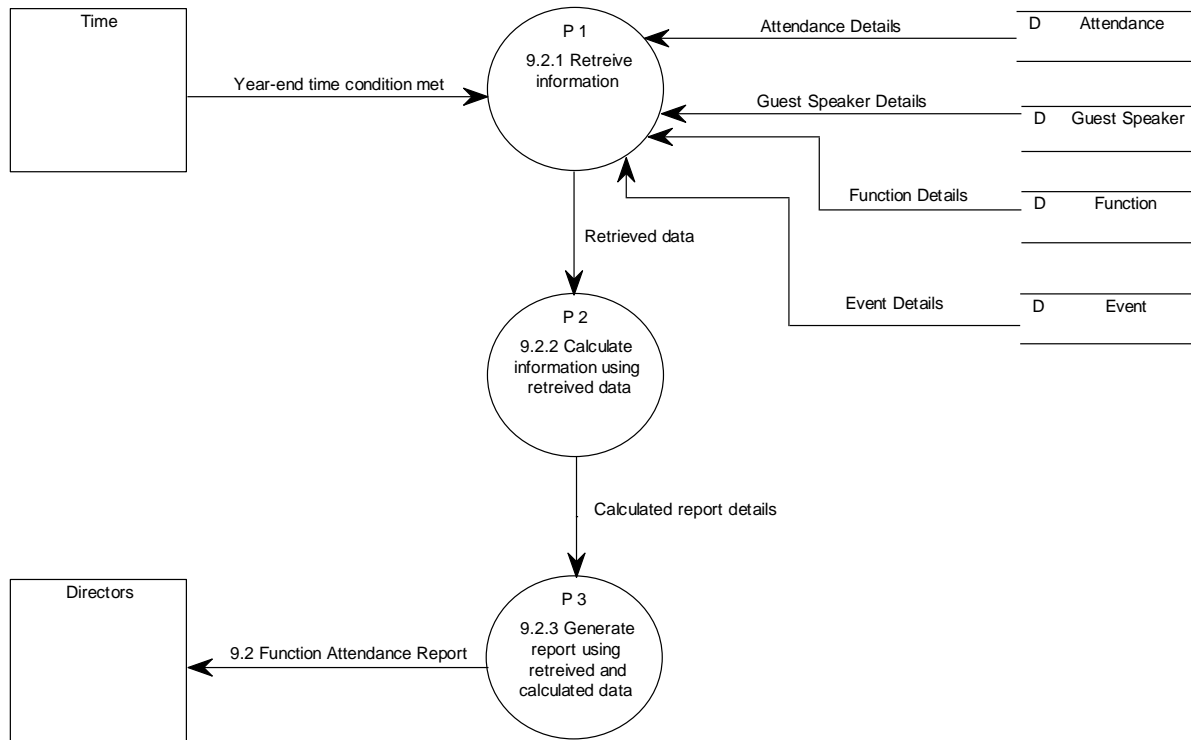


FIGURE 160- PRIMITIVE LEVEL DATA FLOW DIAGRAM 9.2 GENERATE FUNCTION ATTENDANCE REPORT

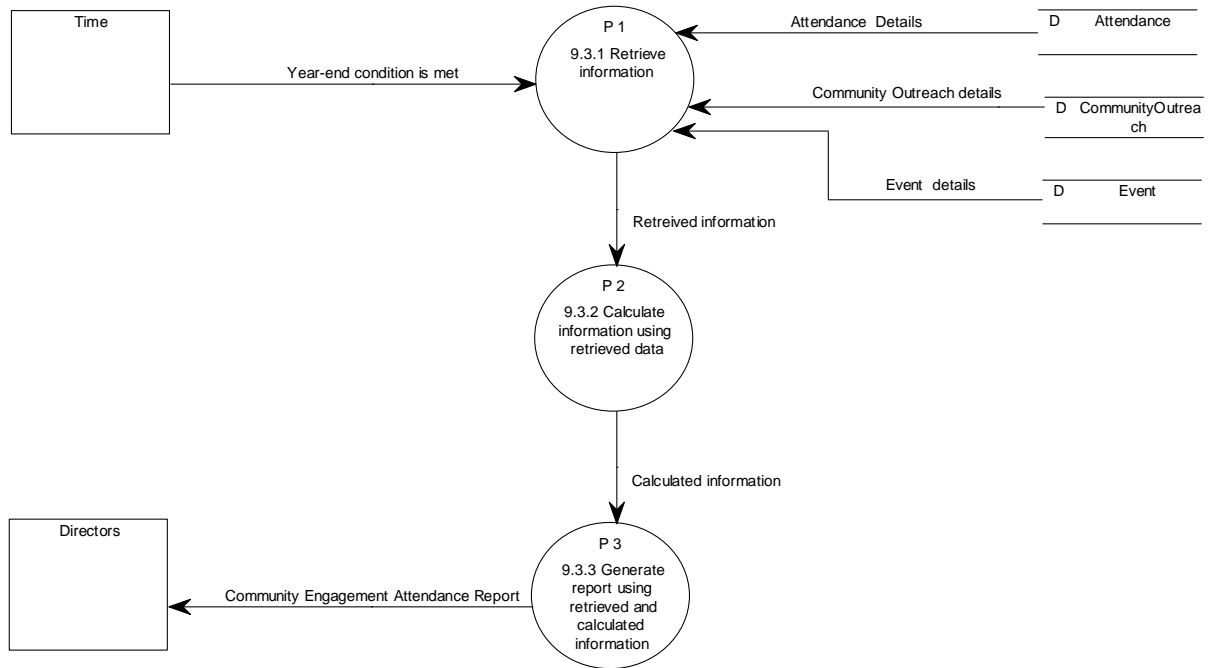


FIGURE 161- PRIMITIVE LEVEL DATA FLOW DIAGRAM 9.3 GENERATE COMMENGAGEMENT ATTENDANCE REPORT

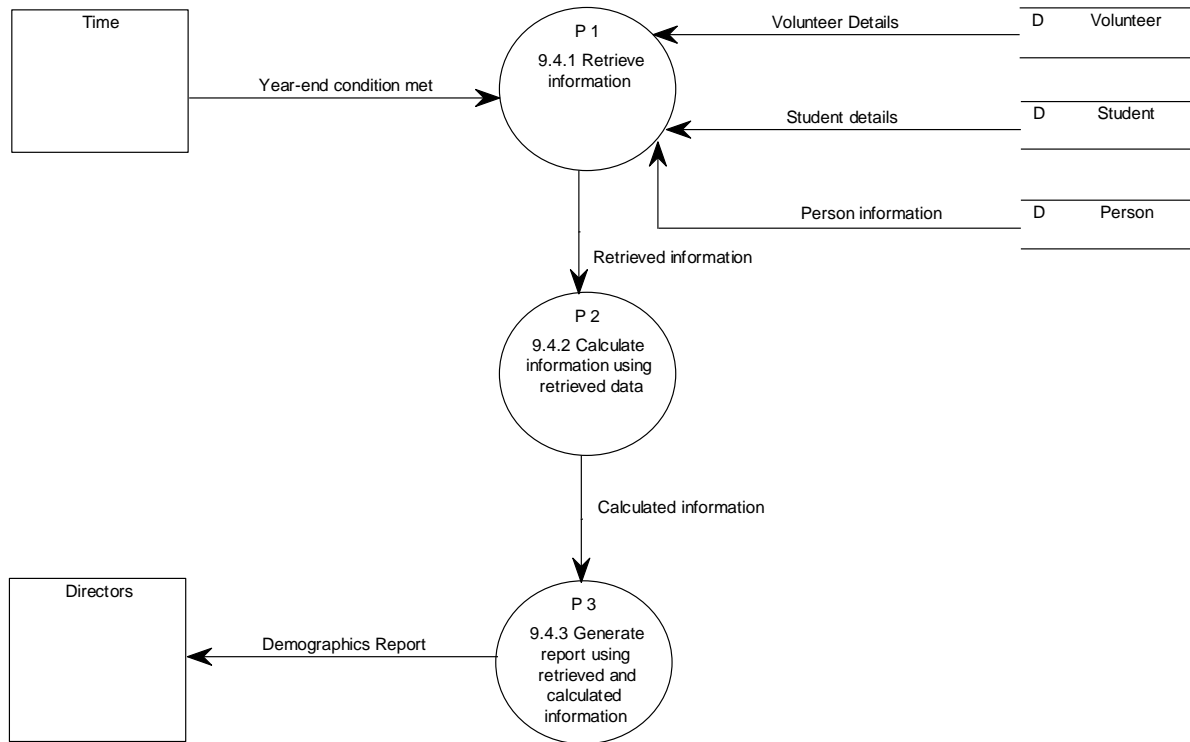


FIGURE 162- PRIMITIVE LEVEL DATA FLOW DIAGRAM 9.4 GENERATE DEMOGRAPHIC REPORT

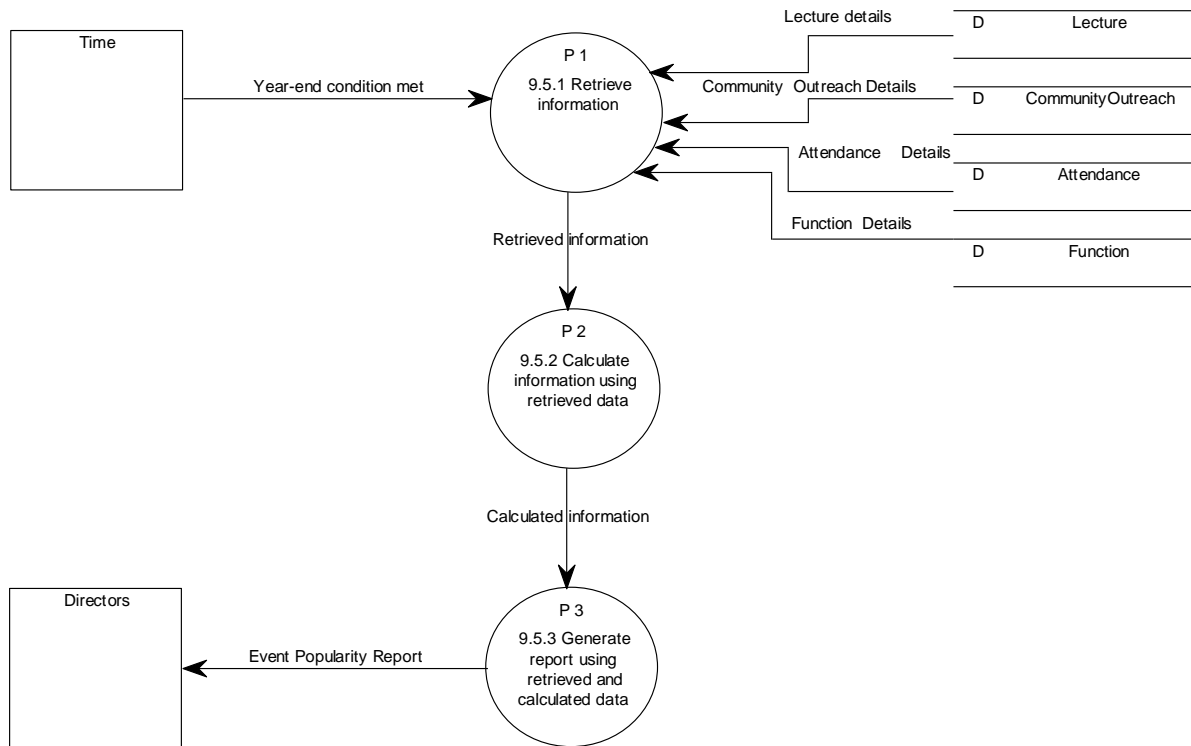


FIGURE 163- PRIMITIVE LEVEL DATA FLOW DIAGRAM 9.5 GENERATE EVENT POPULARITY REPORT

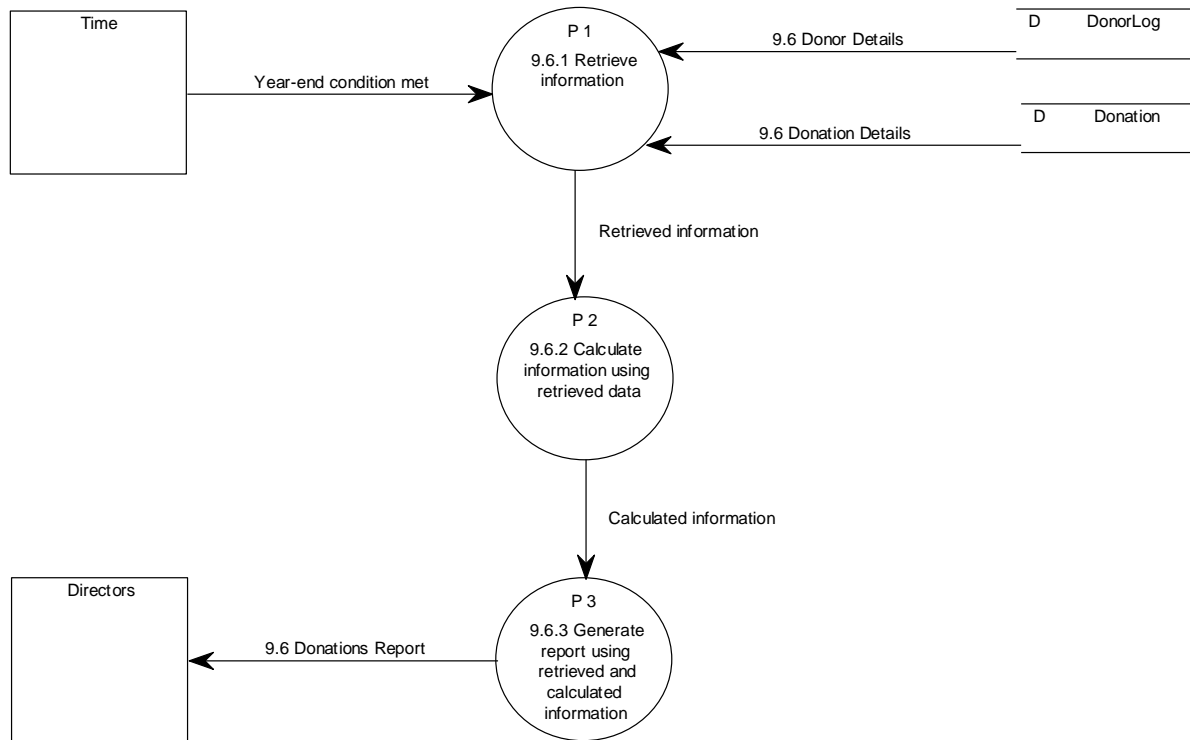


FIGURE 164- PRIMITIVE LEVEL DATA FLOW DIAGRAM 9.6 GENERATE DONATION REPORT

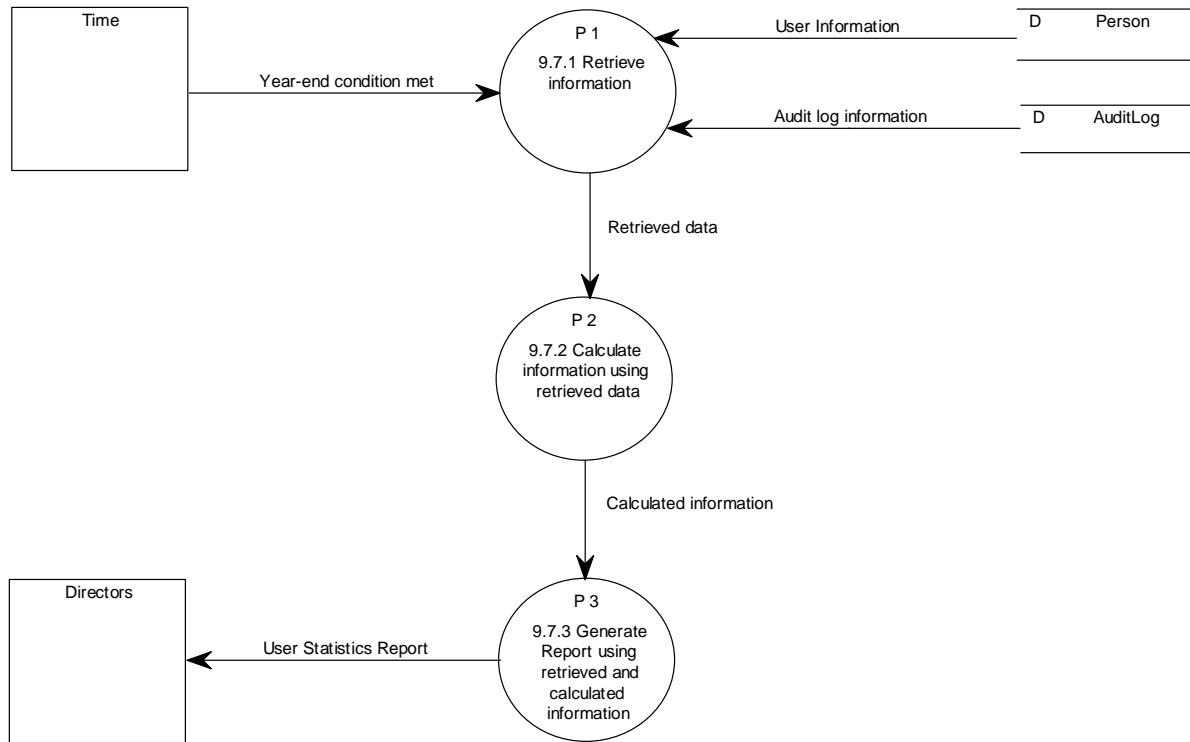


FIGURE 165- PRIMITIVE LEVEL DATA FLOW DIAGRAM 9.7 GENERATE USER STATISTICS REPORT



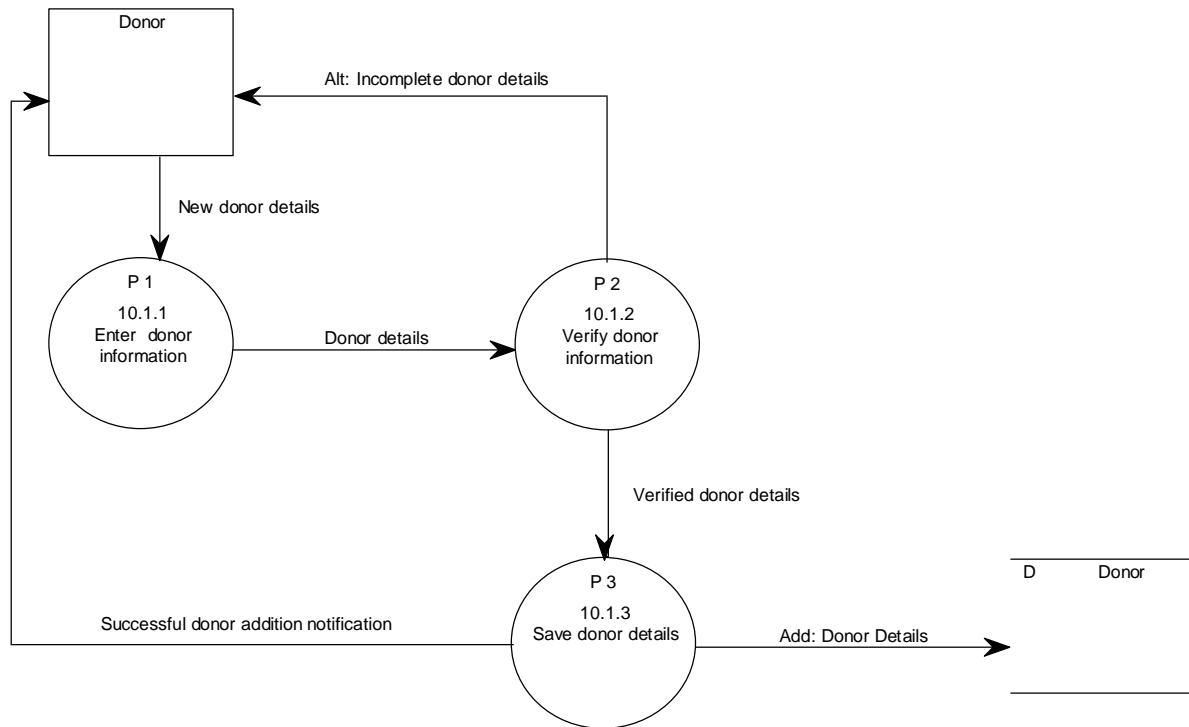


FIGURE 166- PRIMITIVE LEVEL DATA FLOW DIAGRAM 10.1 ADD DONOR

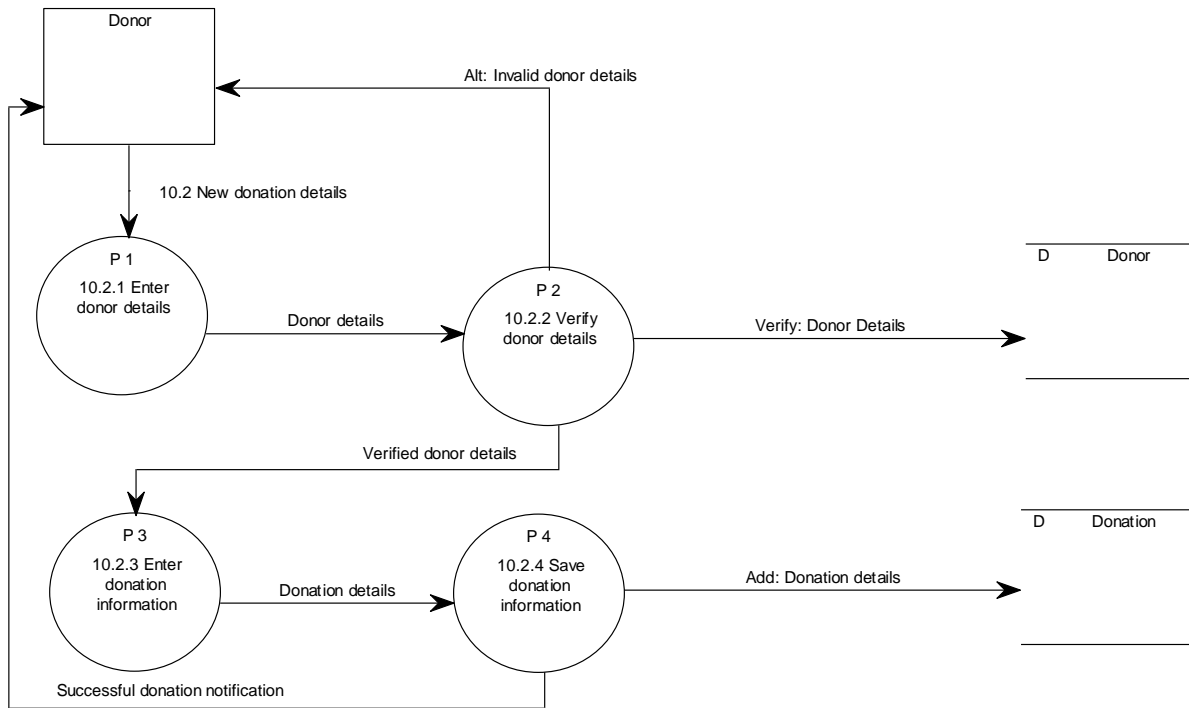


FIGURE 167- PRIMITIVE LEVEL DATA FLOW DIAGRAM 10.2 RECEIVE DONATION

## 3.5 Conclusion

3.5.1 The process models of the TRWLA system were illustrated above. This included the context diagram which described how the external actors interacted directly with the TRWLA system. The decomposition diagram describes the hierarchical order that flows throughout the system from the TRWLA system level, function level, activity level to task level. The data flow diagrams were then illustrated at the end which described every process within the TRWLA system.

## 4. UML Modelling

### 4.1 Introduction

4.1.1 The next section illustrates UML modelling by the use of activity diagrams. The activity diagrams depict the flow of activities with regards to each use case illustrated above. The activities relate to the primitive-level data flow diagrams but describe when an action takes place and when it ends. It also states where an external actor receives any physical output from the system, such as reports.

## 4.2 Activity Diagrams

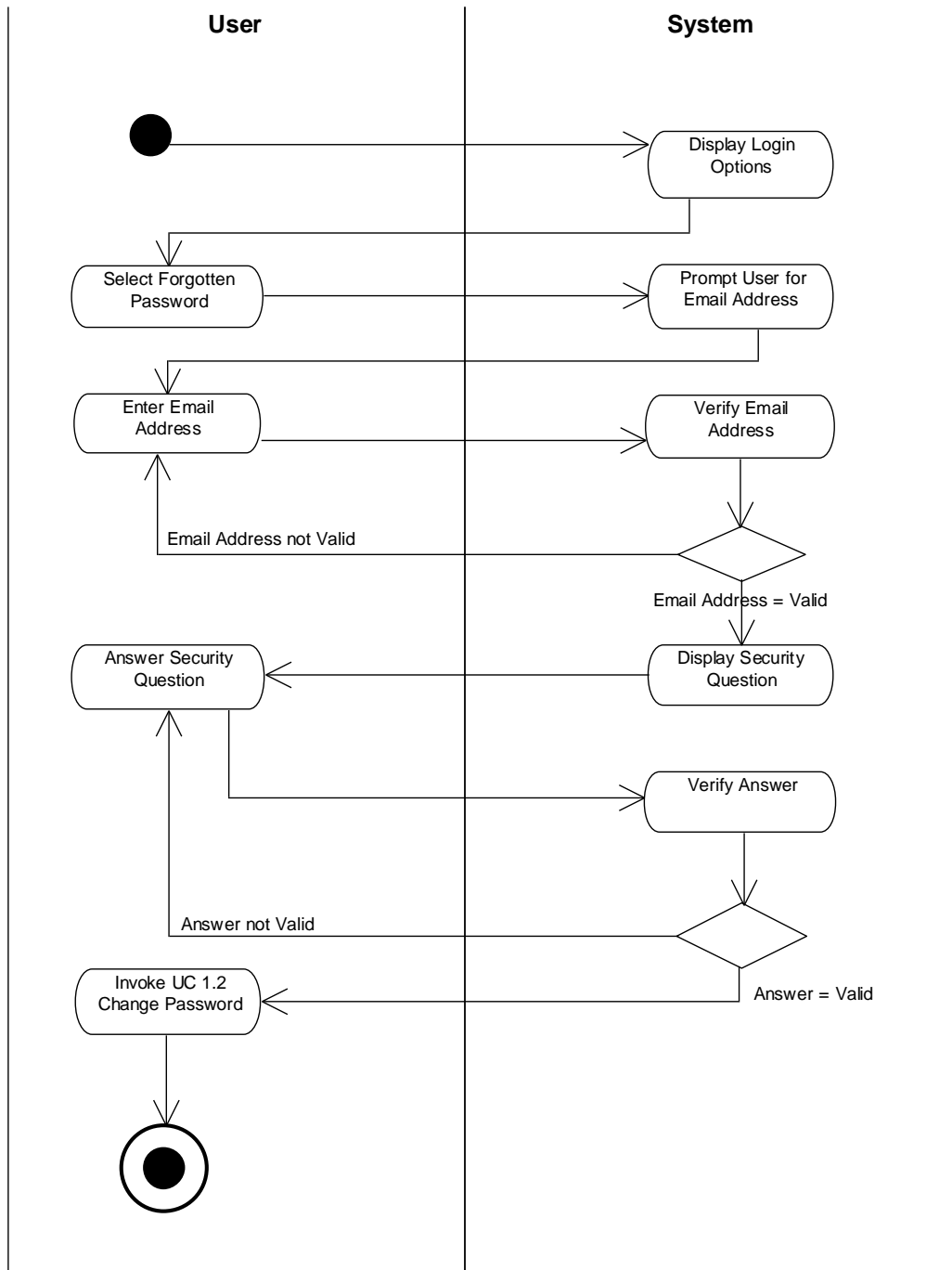


FIGURE 168- ACTIVITY DIAGRAM 1.1 CHECK FORGOTTEN PASSWORD

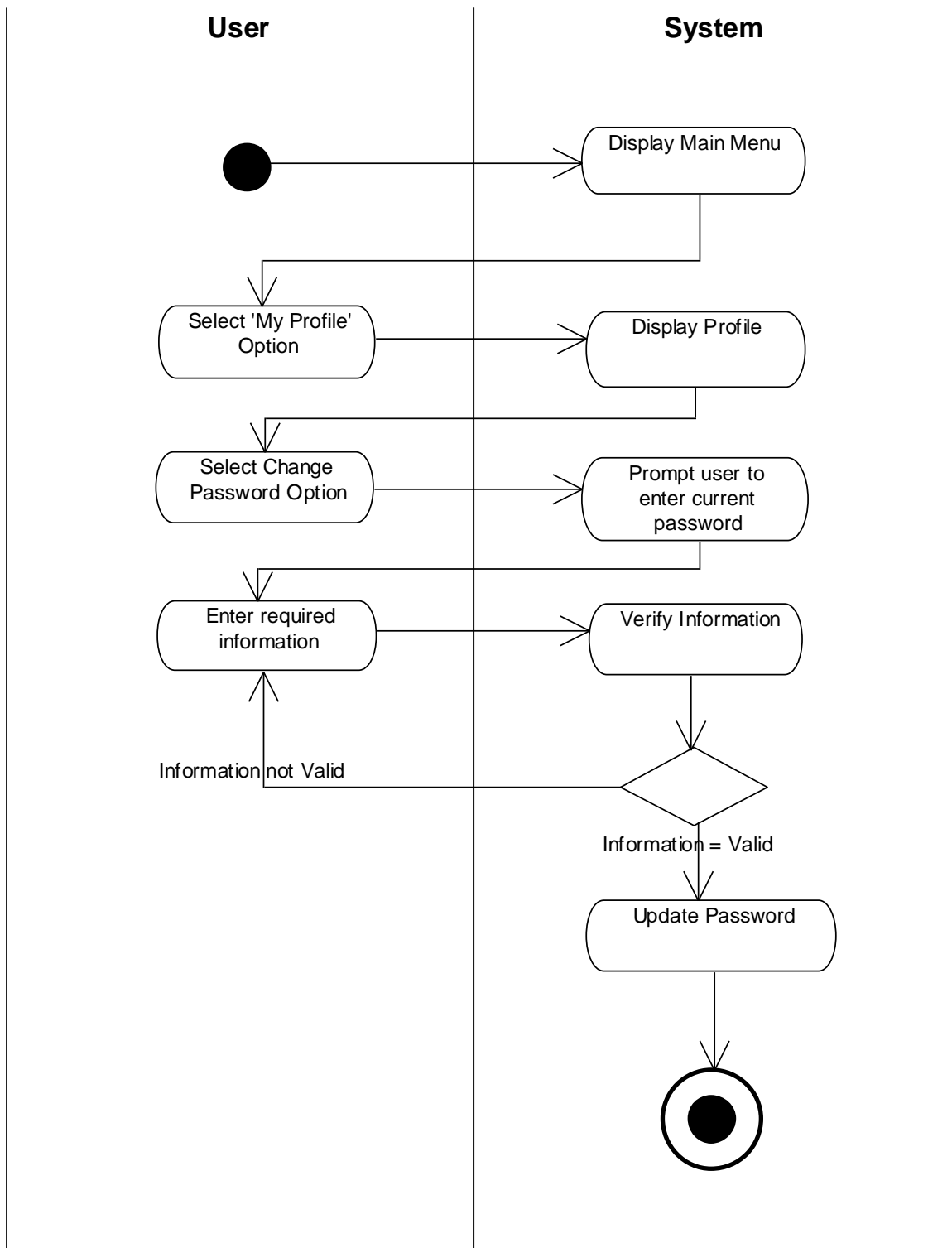


FIGURE 169- ACTIVITY DIAGRAM 1.2 CHANGE PASSWORD

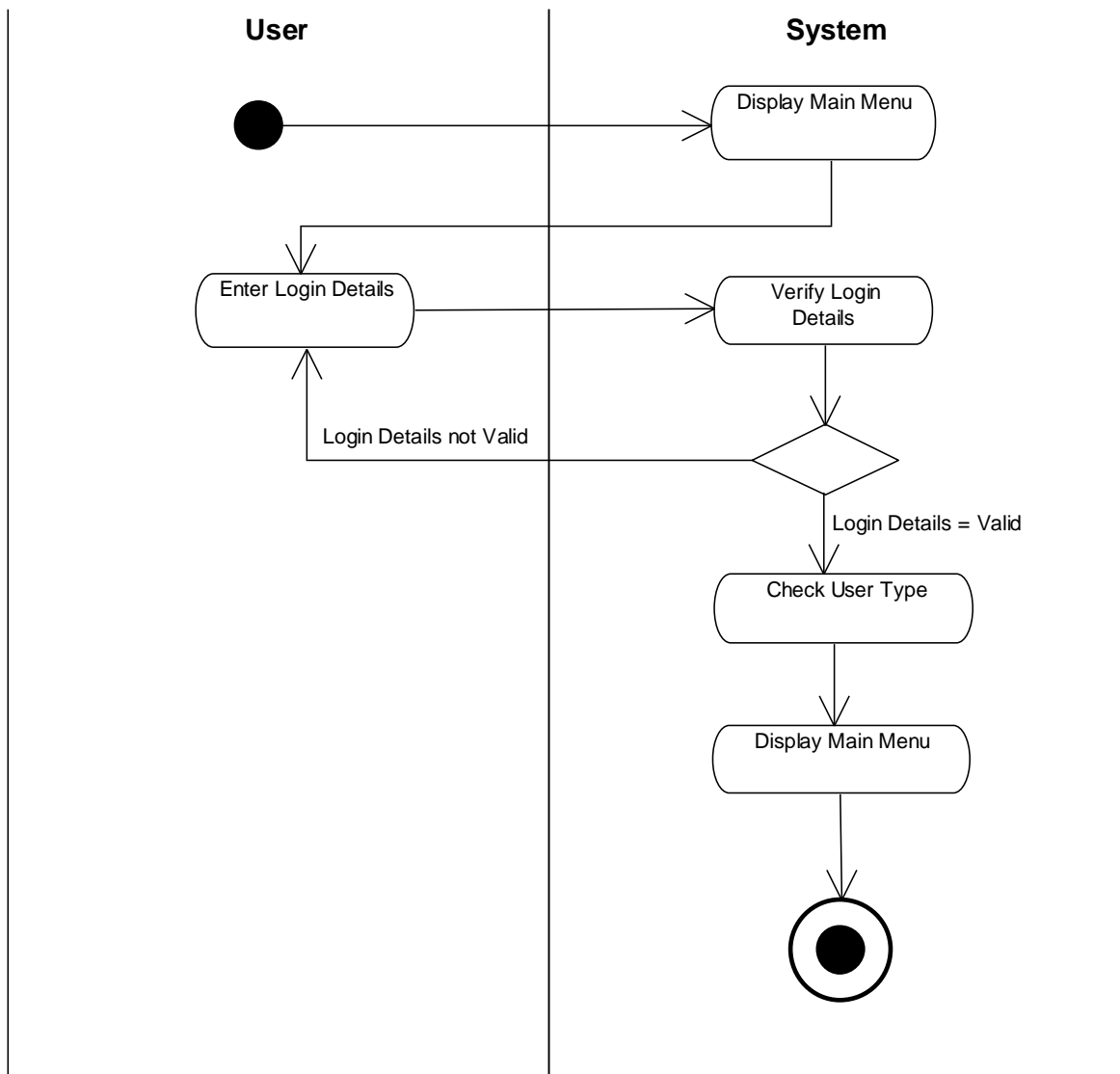


FIGURE 170- ACTIVITY DIAGRAM 1.3 LOGIN

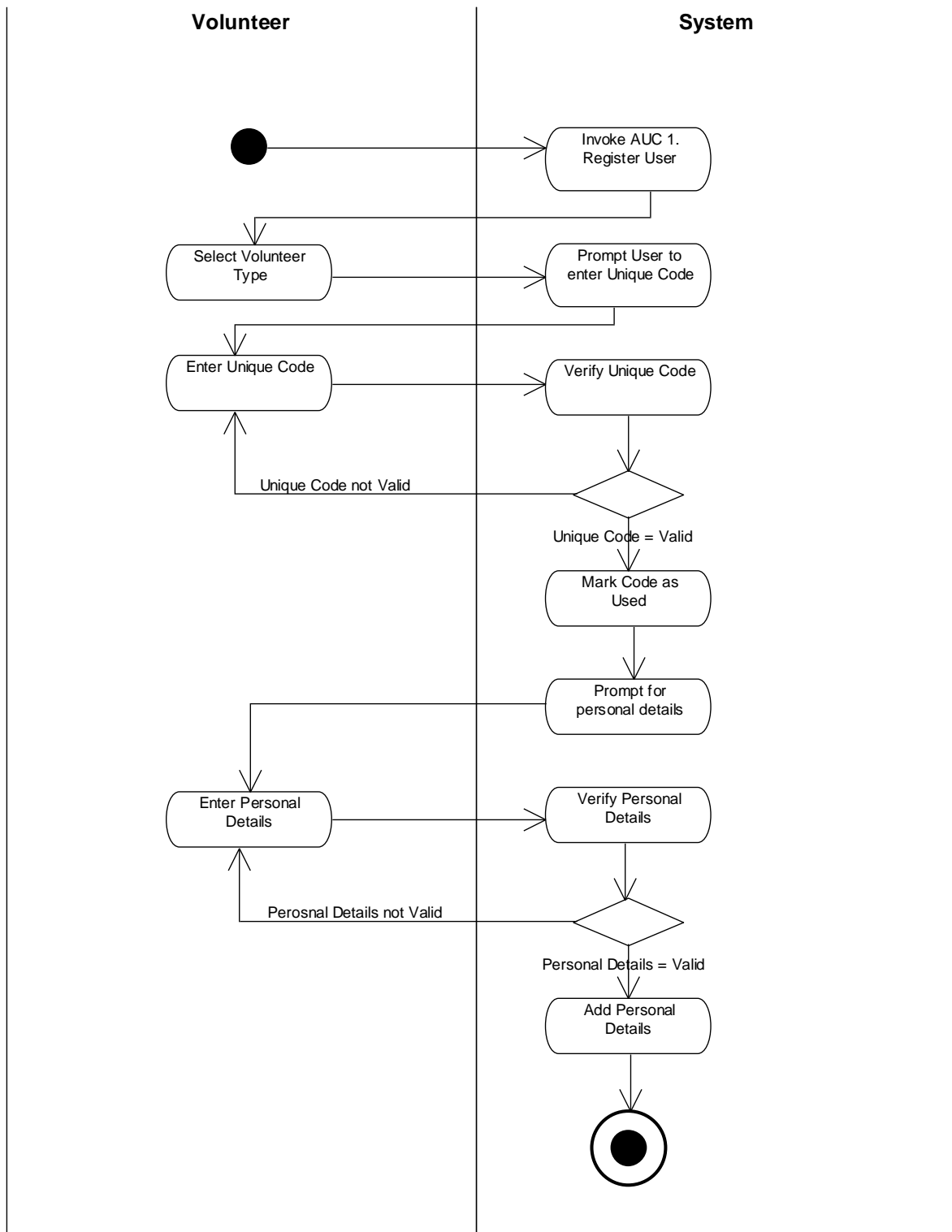


FIGURE 171- ACTIVITY DIAGRAM 2.1 REGISTER VOLUNTEER

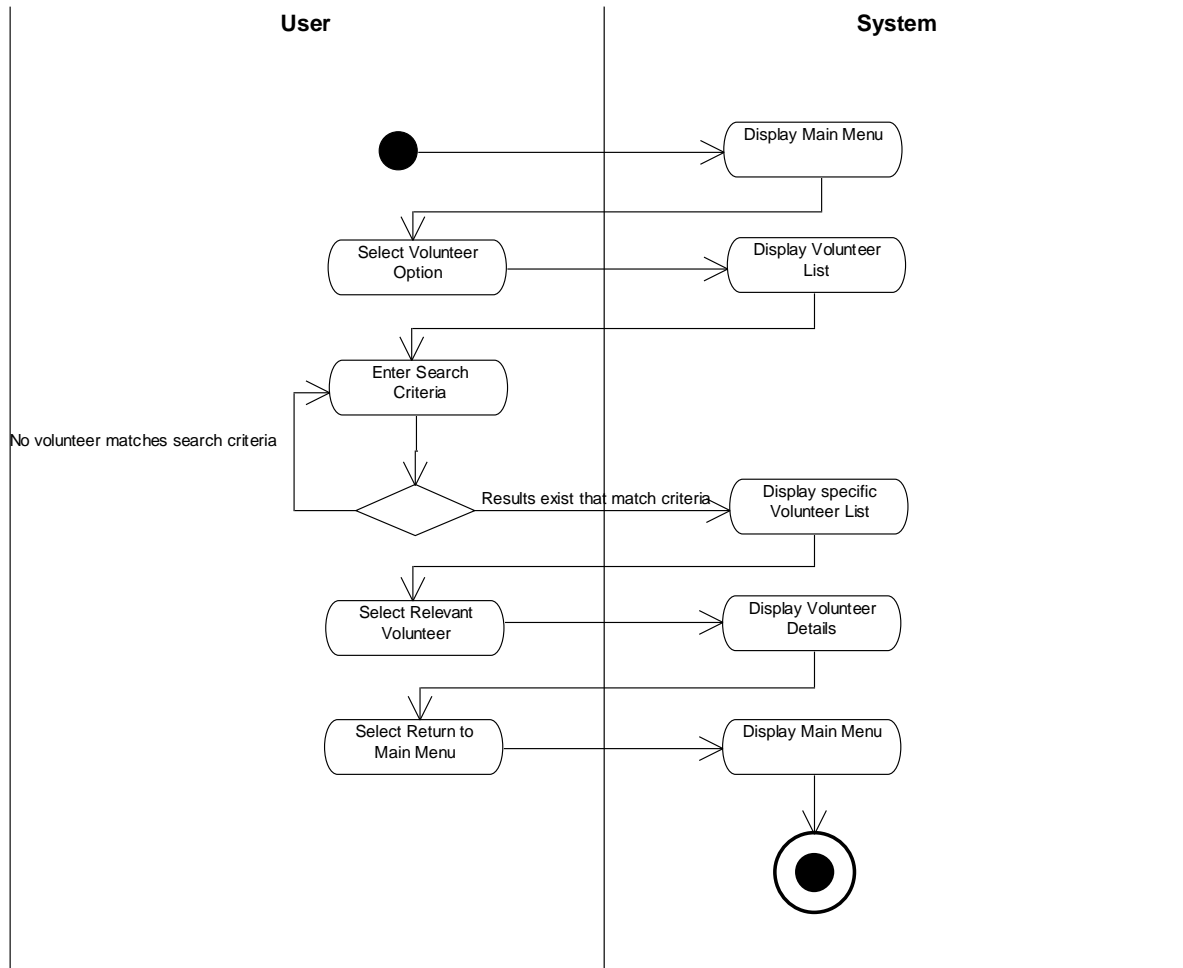


FIGURE 172- ACTIVITY DIAGRAM 2.2 SEARCH VOLUNTEER



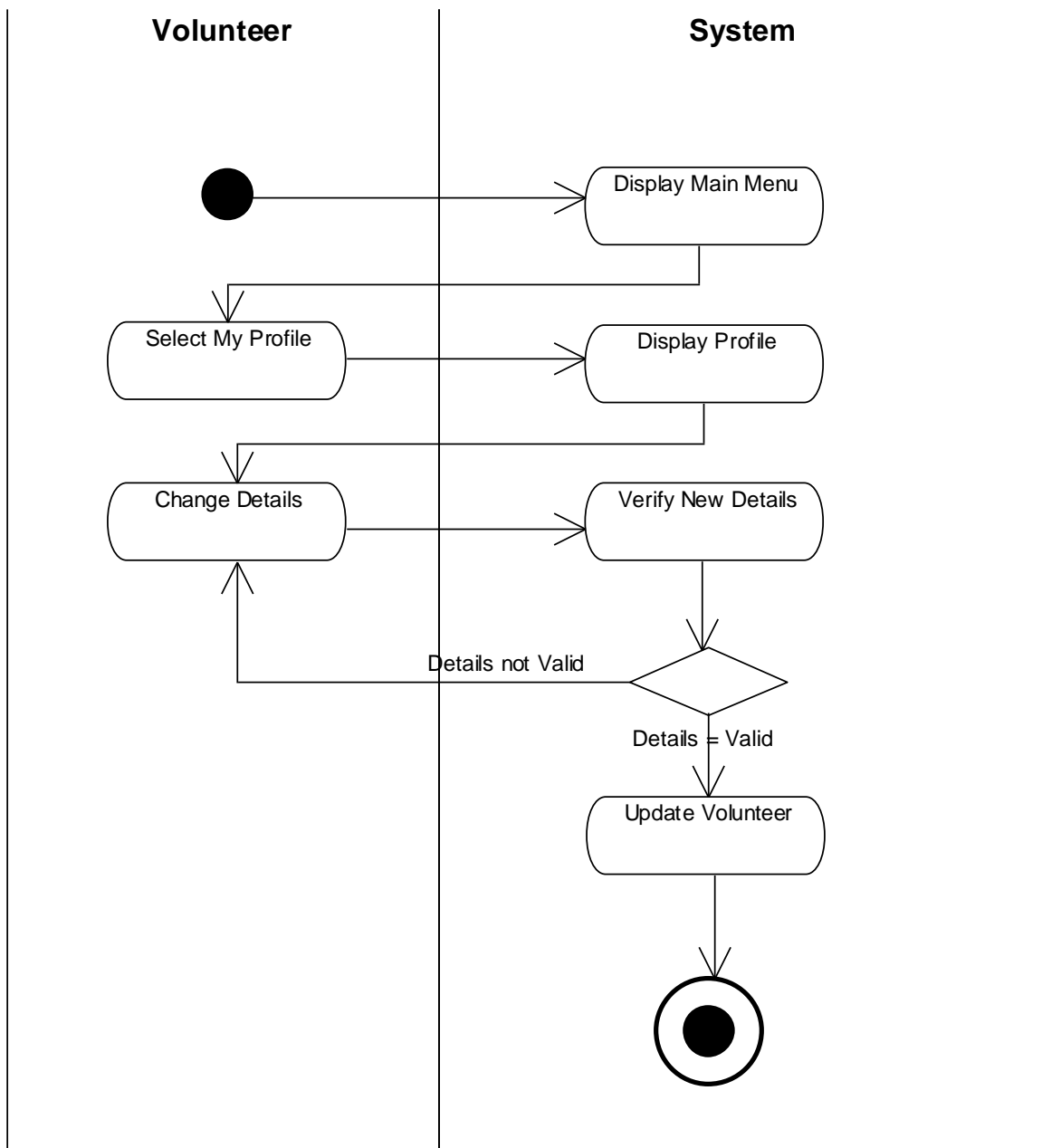


FIGURE 173- ACTIVITY DIAGRAM 2.3 UPDATE VOLUNTEER

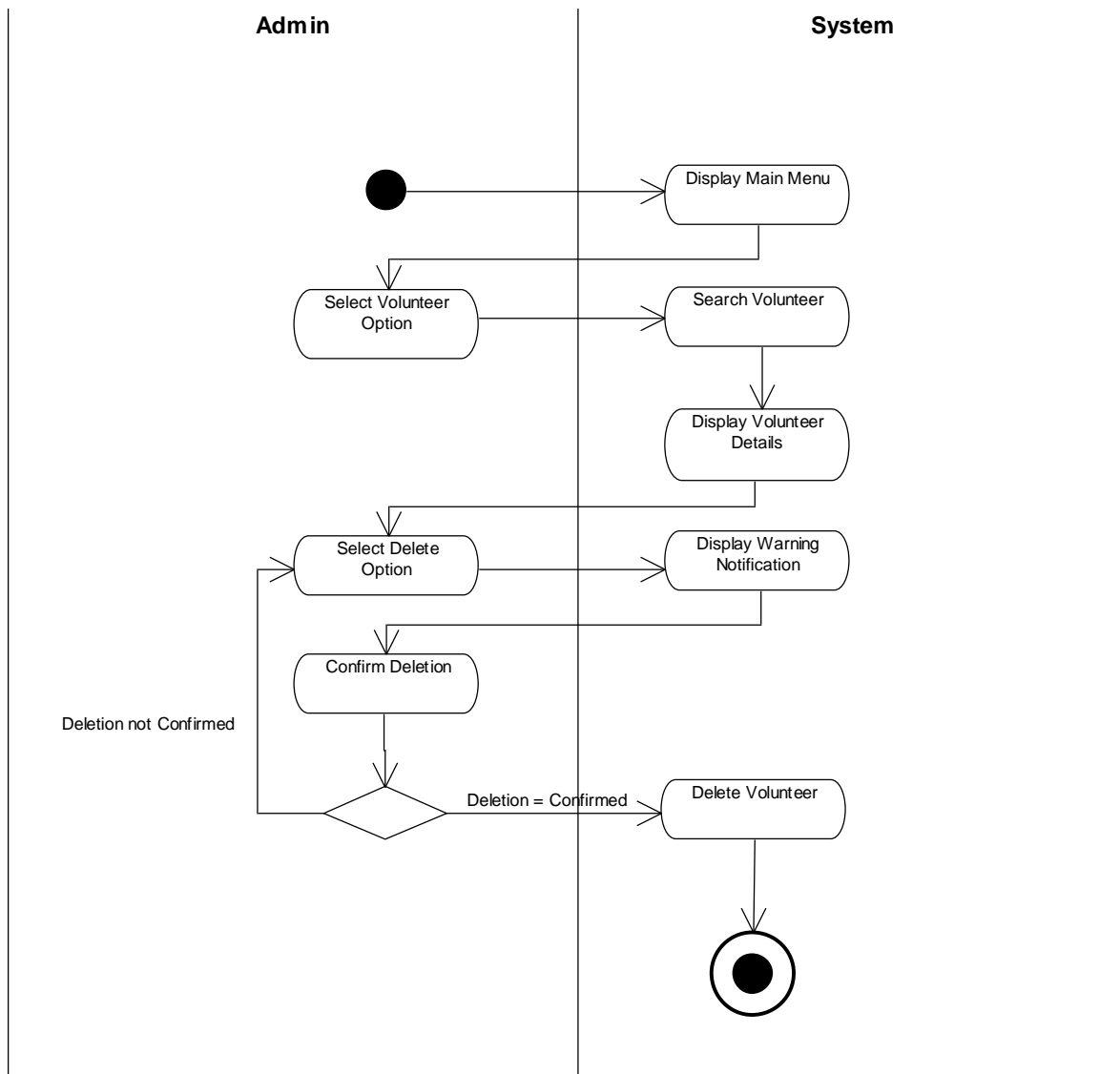


FIGURE 174- ACTIVITY DIAGRAM 2.4 DELETE VOLUNTEER

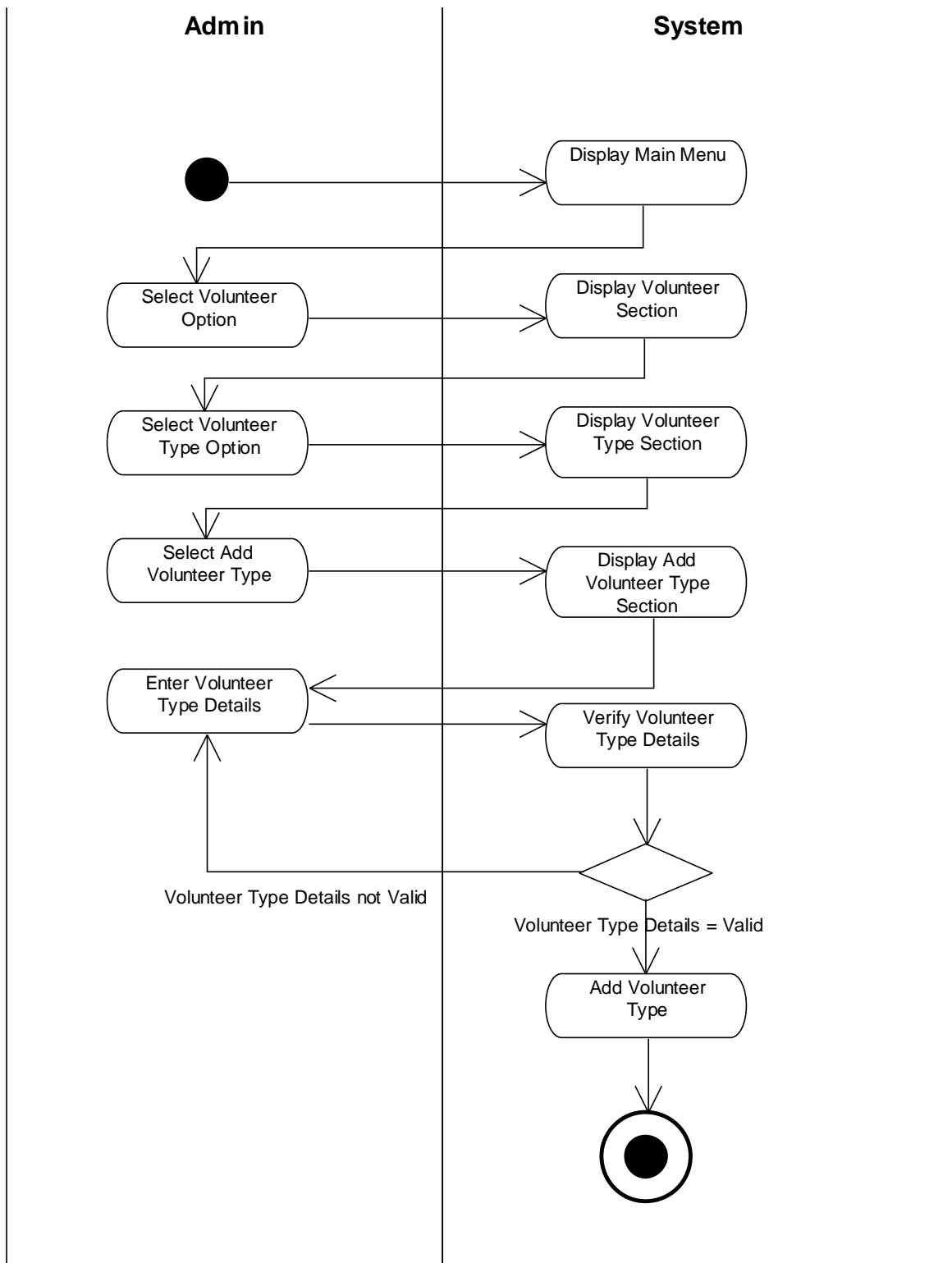


FIGURE 175- ACTIVITY DIAGRAM 2.5 ADD VOLUNTEER TYPE

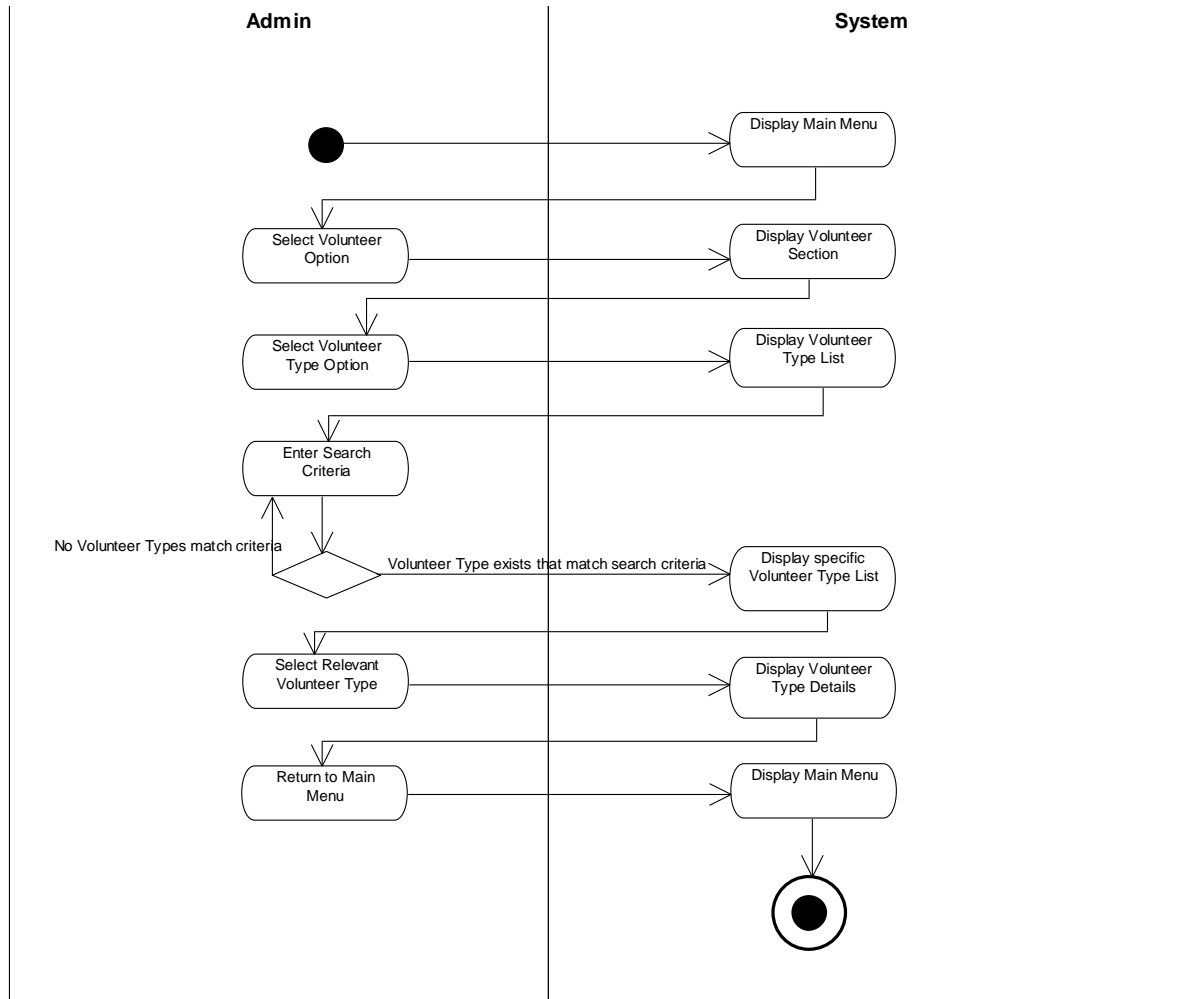


FIGURE 176- ACTIVITY DIAGRAM 2.6 SEARCH VOLUNTEER TYPE

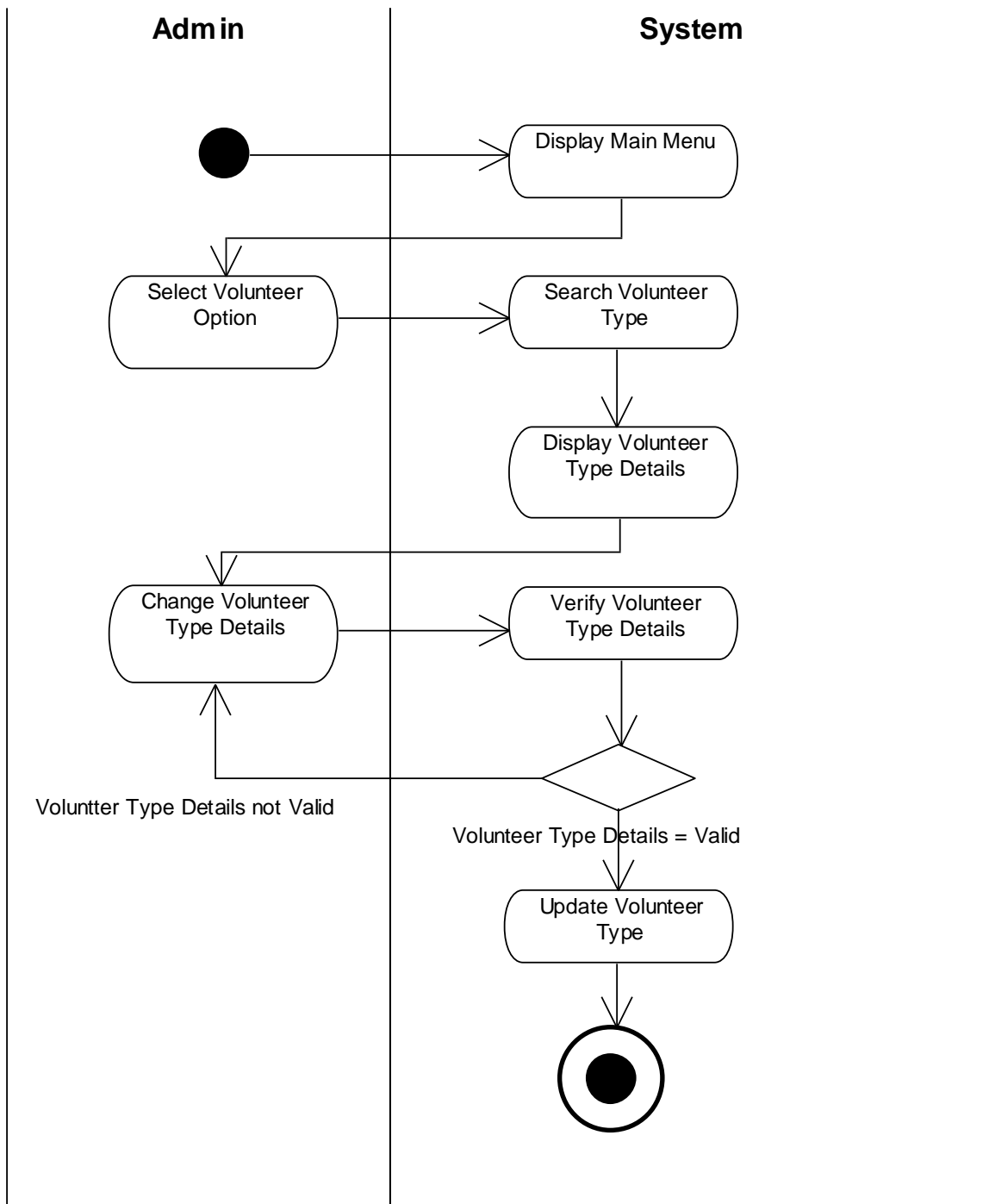


FIGURE 177- ACTIVITY DIAGRAM 2.7 UPDATE VOLUNTEER TYPE

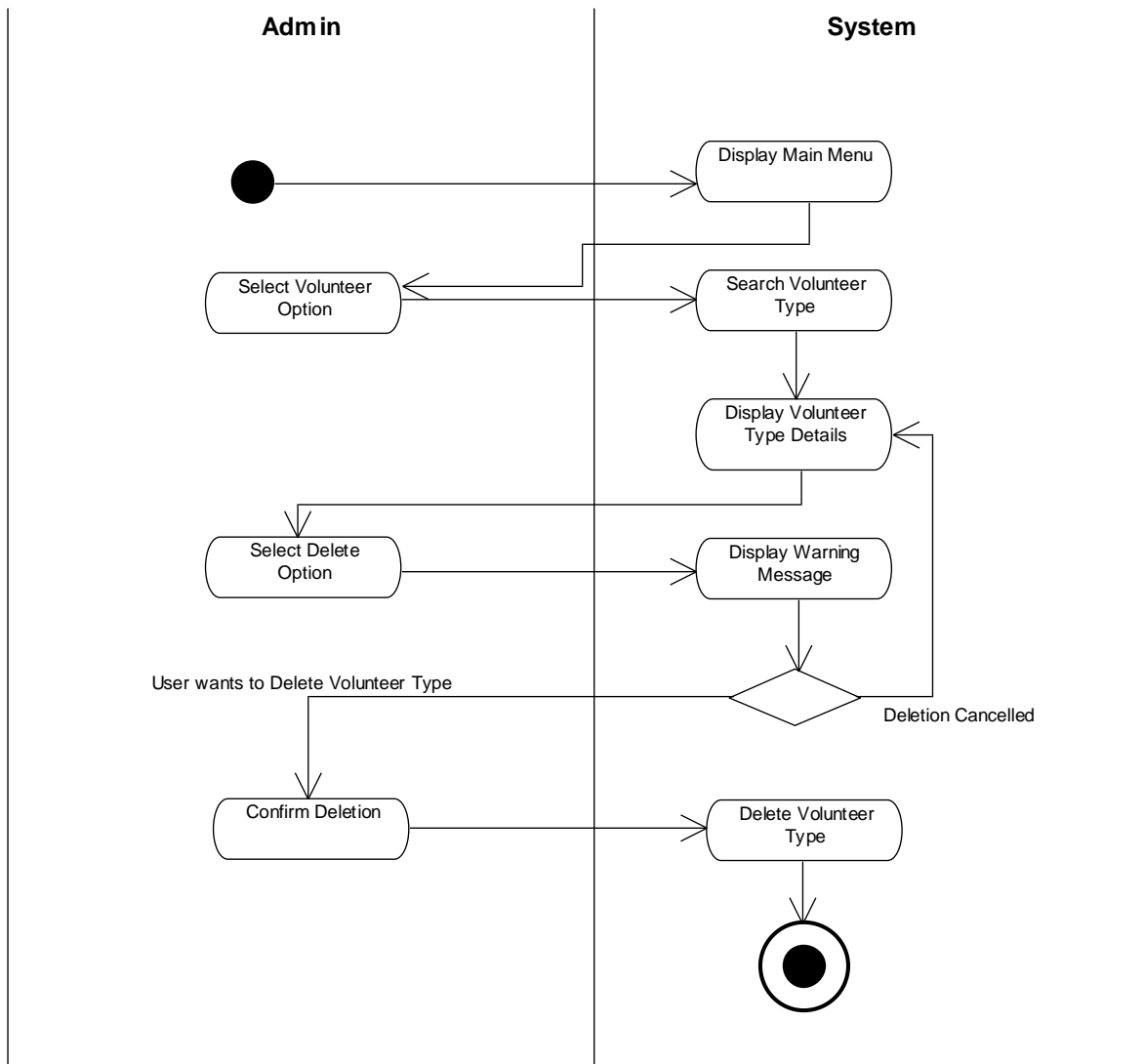


FIGURE 178- ACTIVITY DIAGRAM 2.8 DELETE VOLUNTEER TYPE

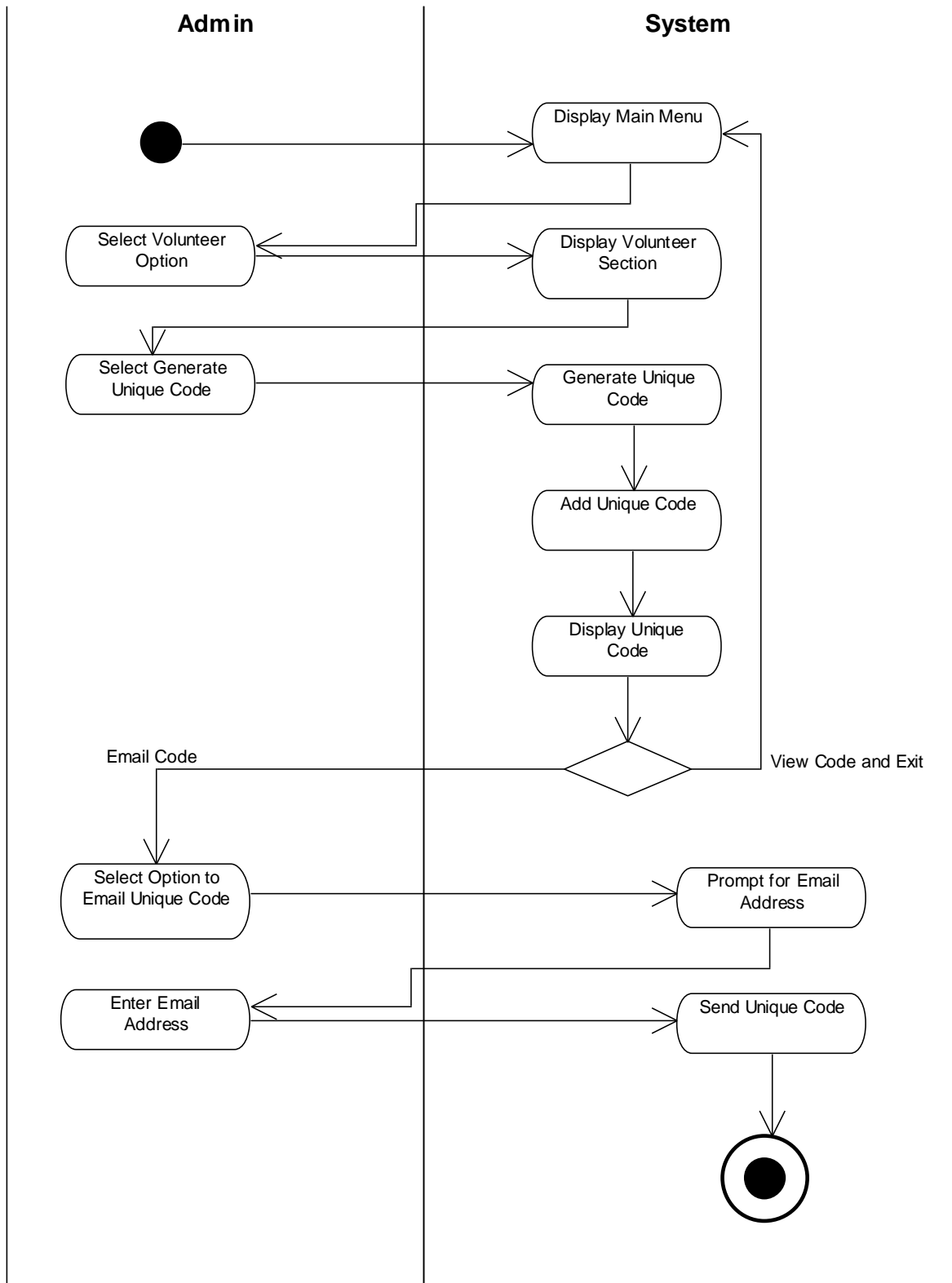


FIGURE 179- ACTIVITY DIAGRAM 2.9 GENERATE UNIQUE CODE

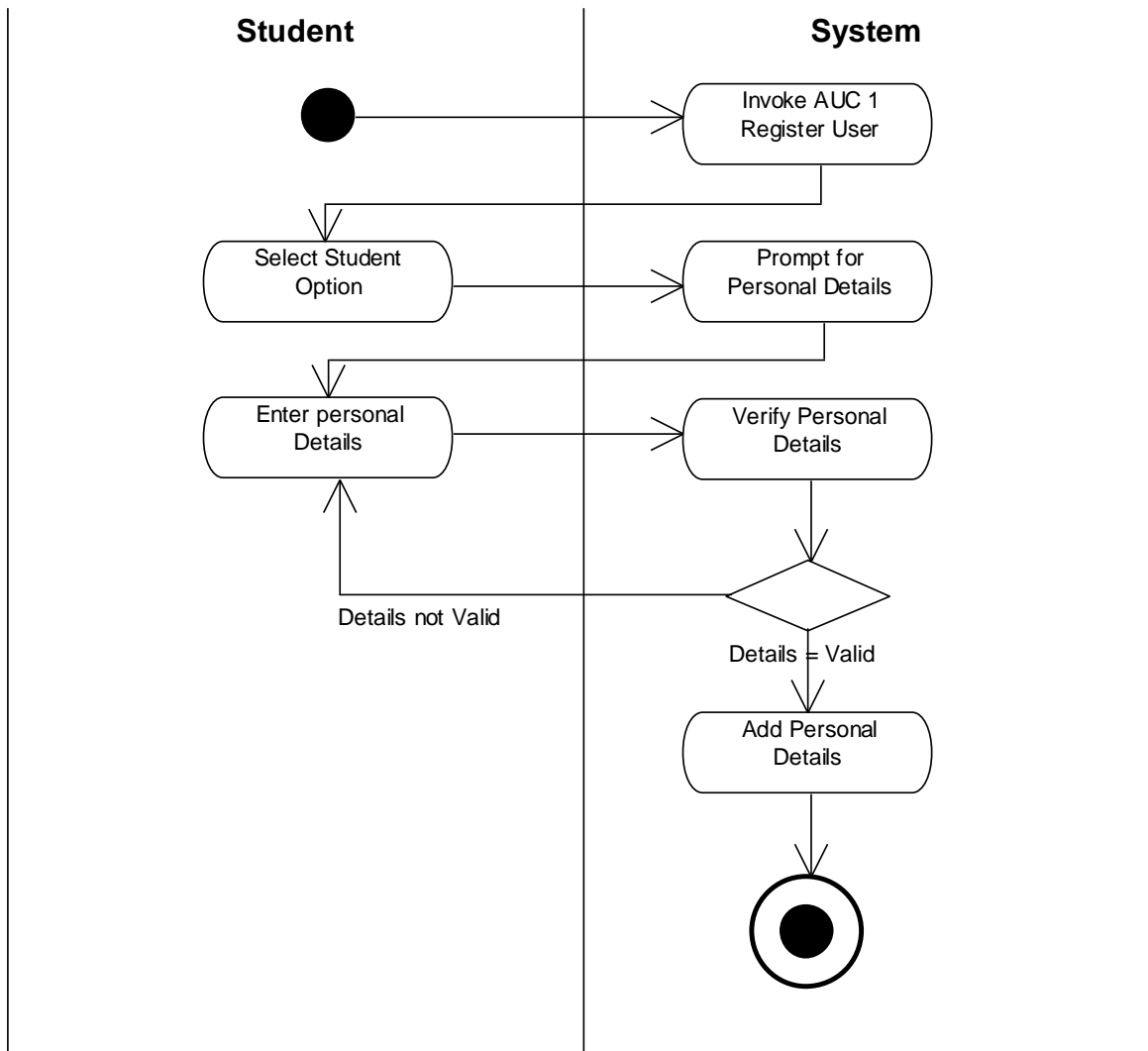


FIGURE 180- ACTIVITY DIAGRAM 3.1 REGISTER STUDENT



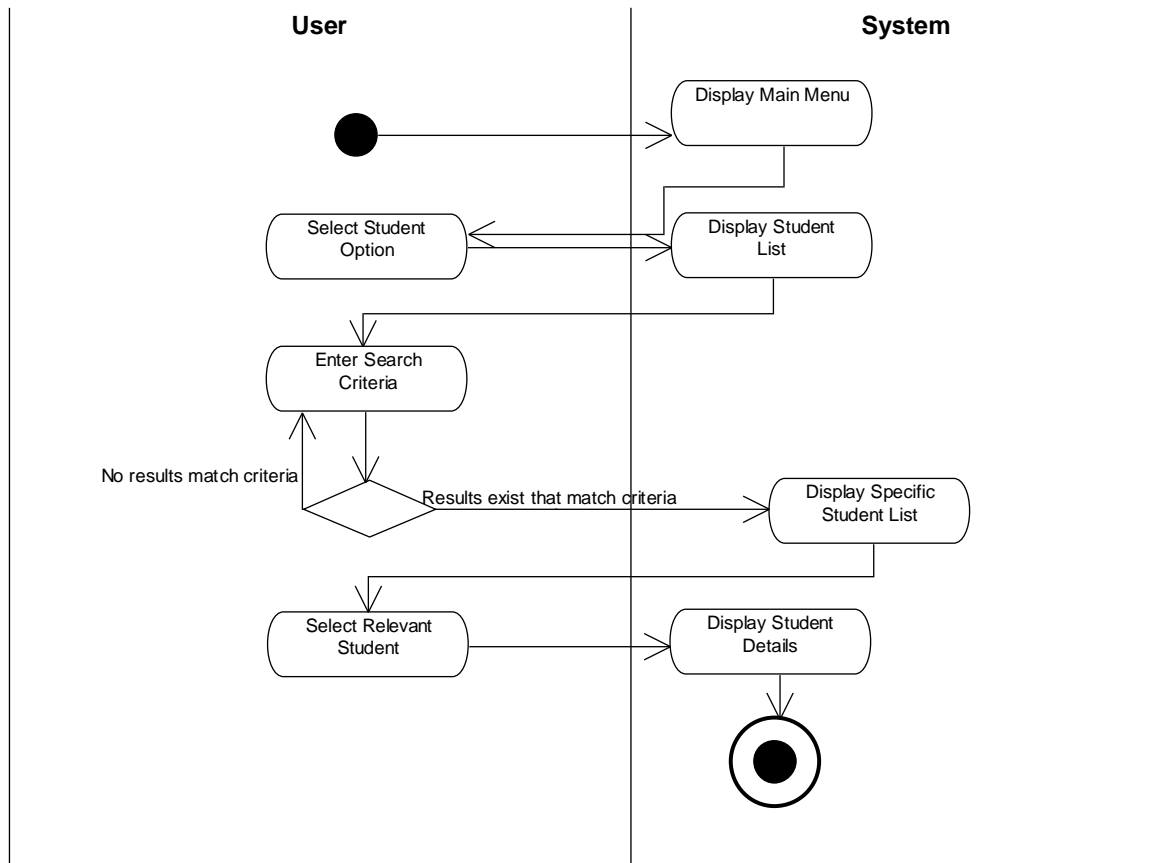


FIGURE 181- ACTIVITY DIAGRAM 3.2 SEARCH STUDENT

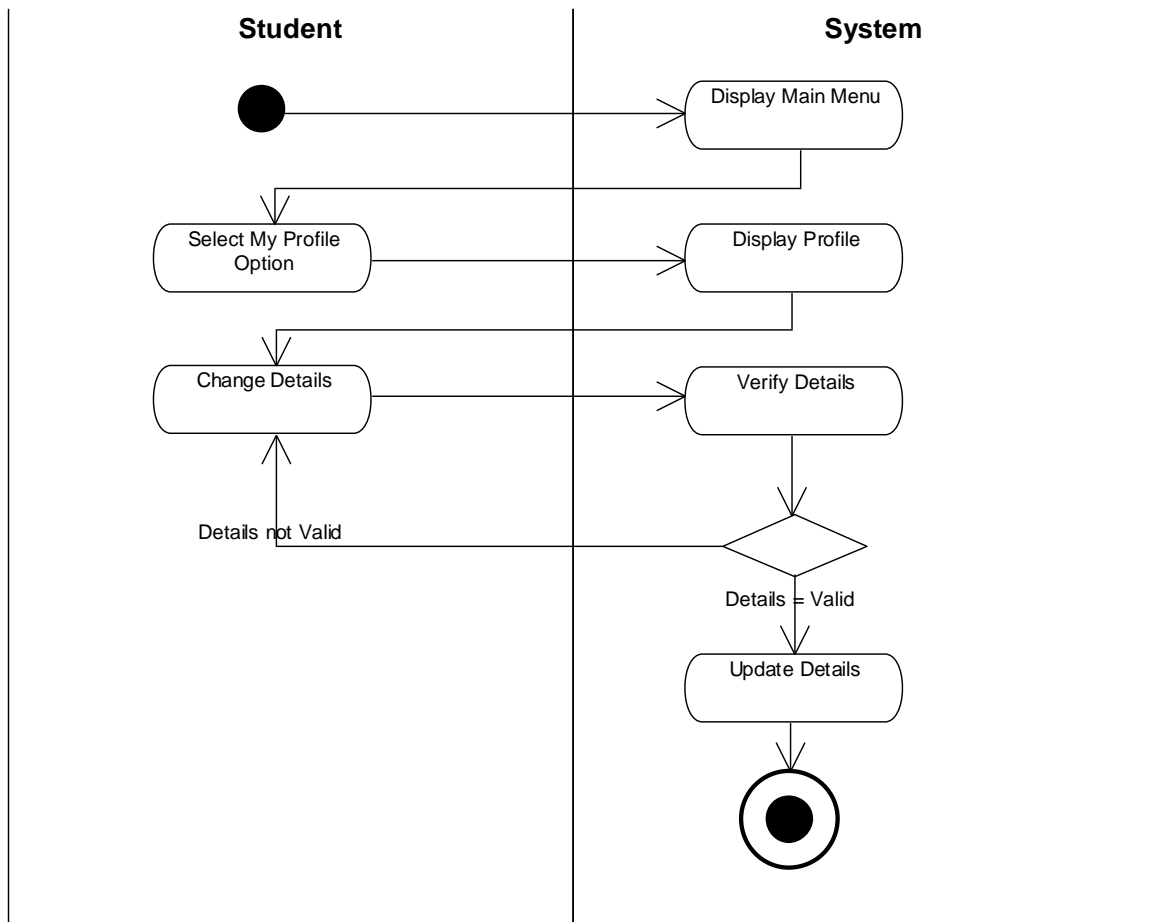


FIGURE 182- ACTIVITY DIAGRAM 3.3 UPDATE STUDENT

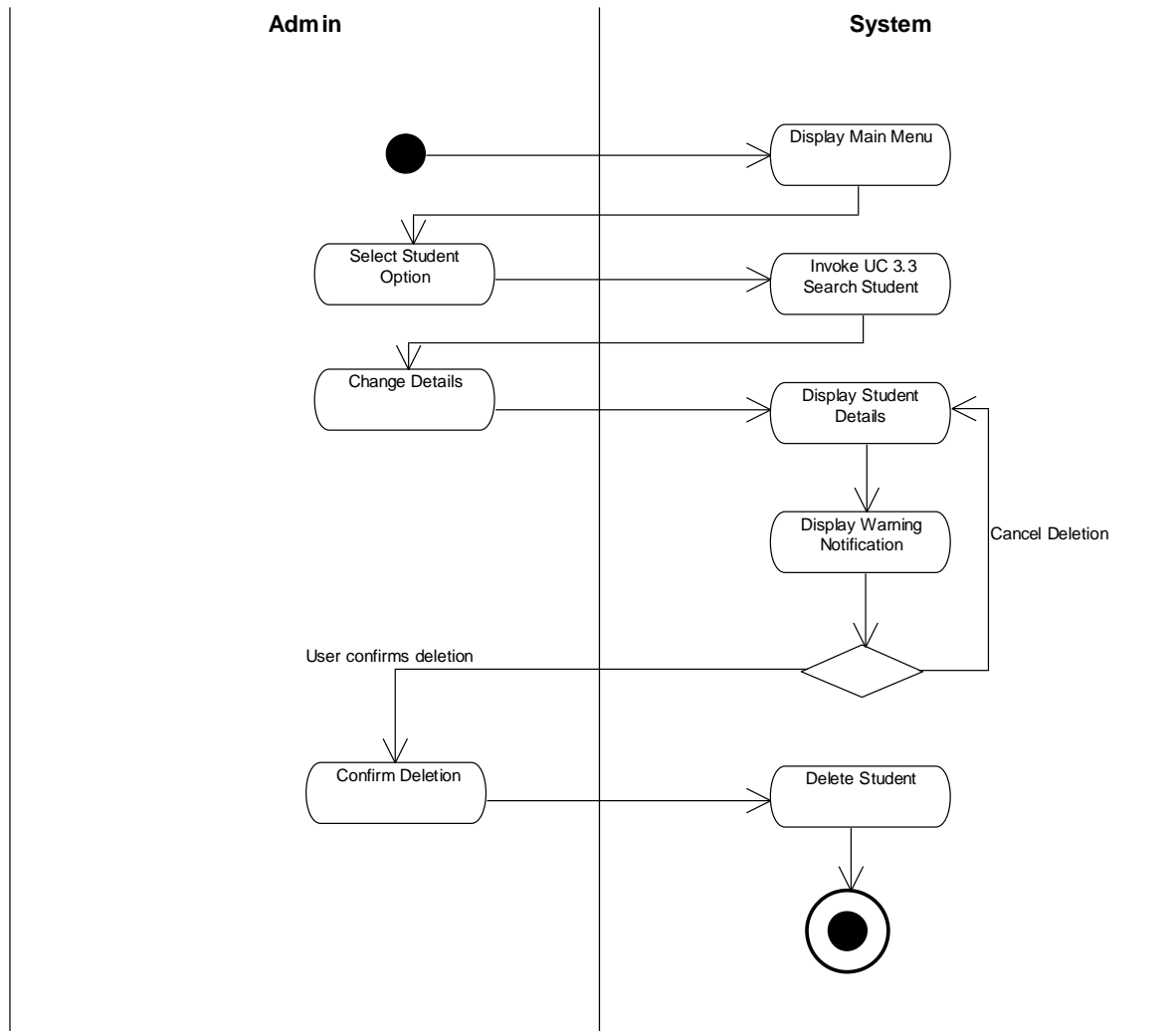


FIGURE 183- ACTIVITY DIAGRAM 3.4 DELETE STUDENT

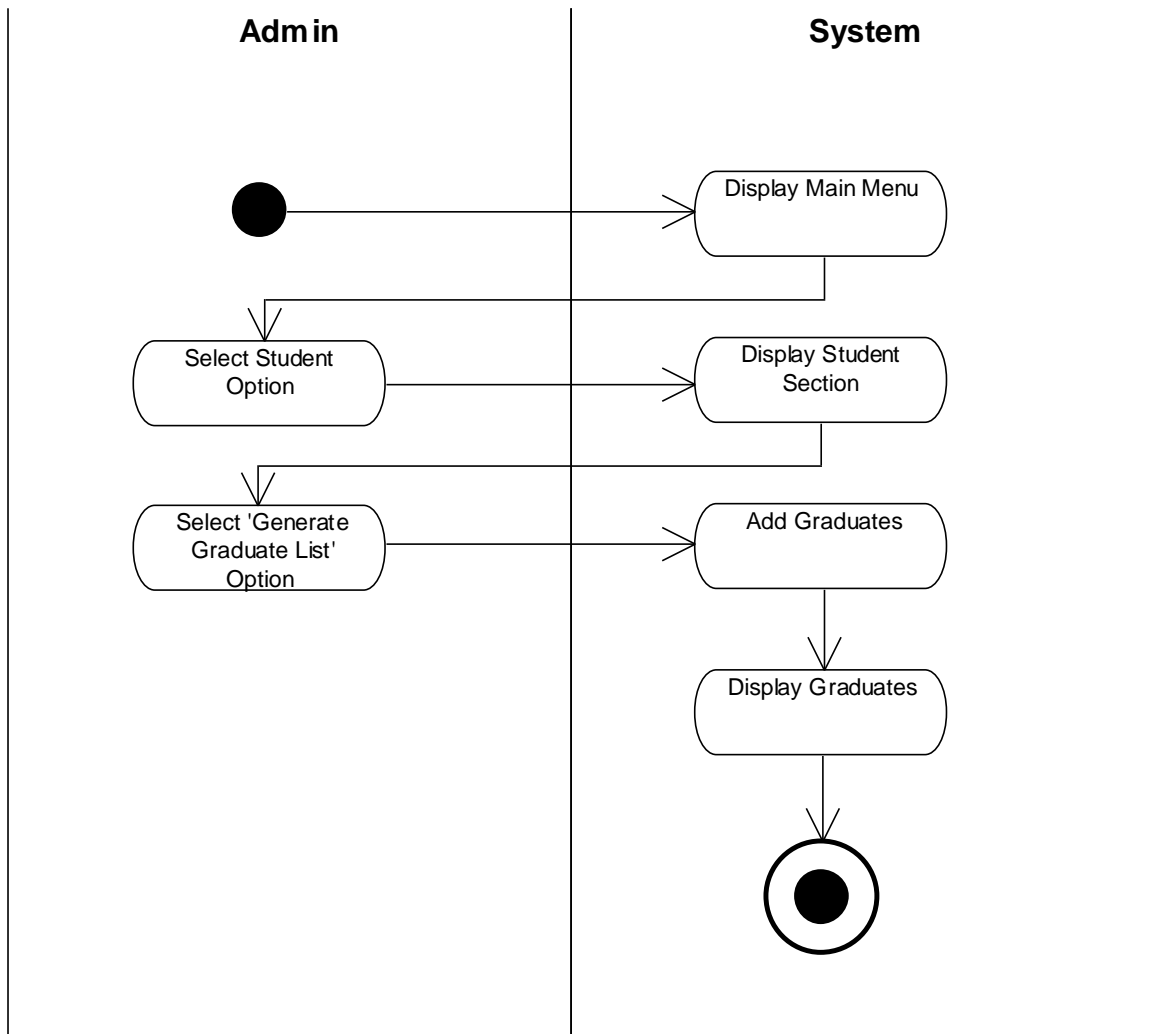


FIGURE 184- ACTIVITY DIAGRAM 3.5 GENERATE GRADUATE LIST

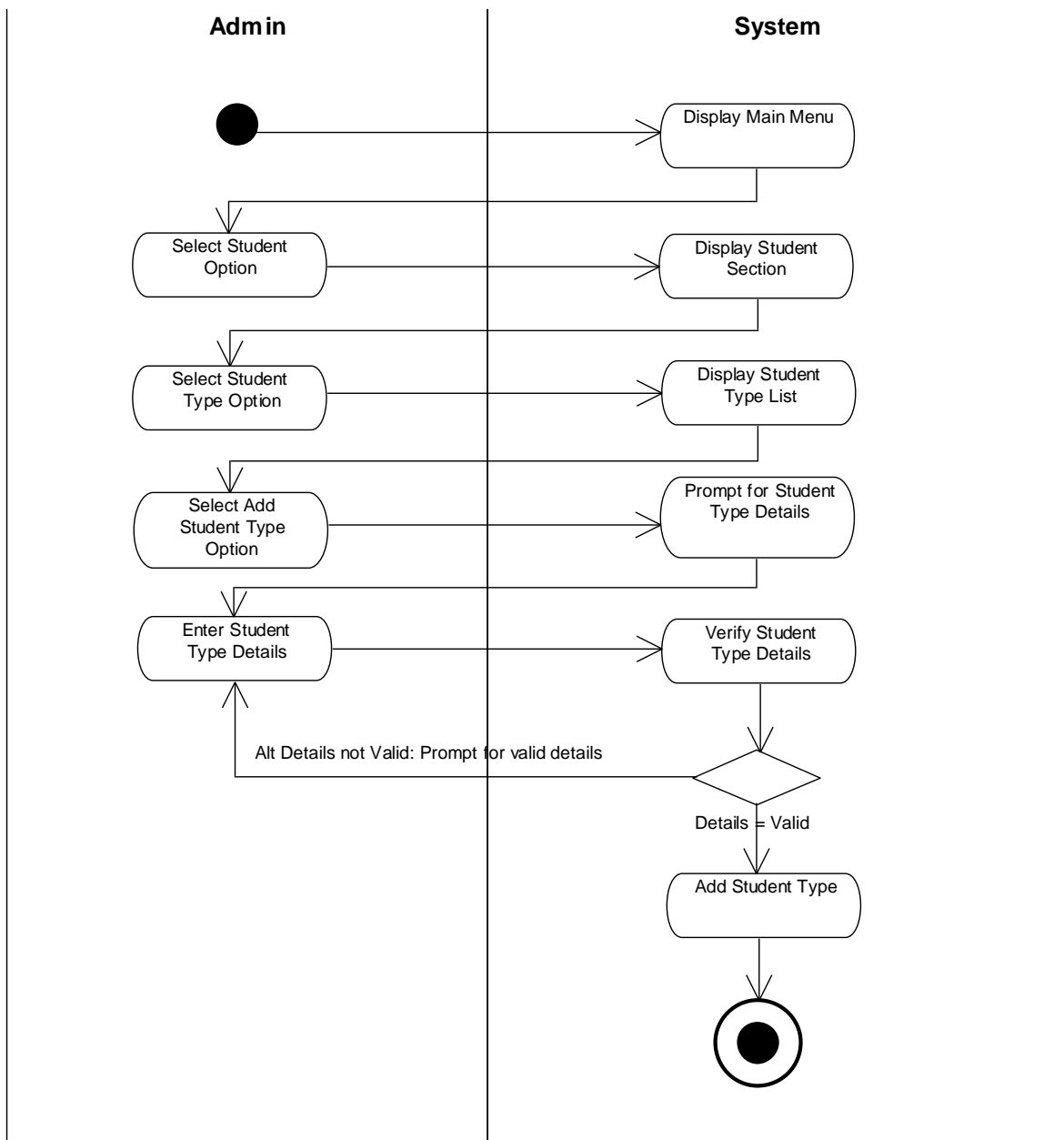


FIGURE 185- ACTIVITY DIAGRAM 3.6 ADD STUDENT TYPE

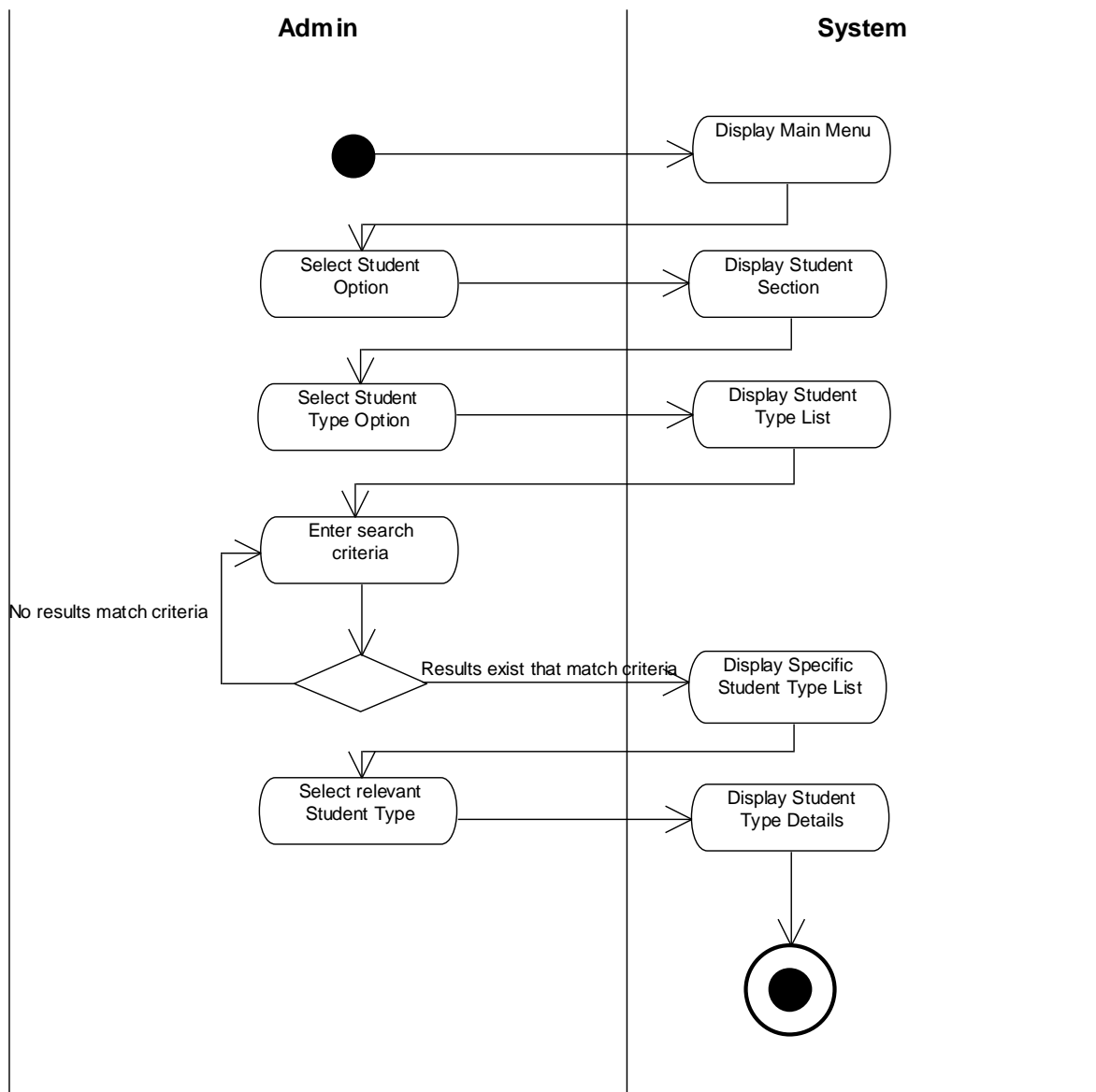


FIGURE 186- ACTIVITY DIAGRAM 3.7 SEARCH STUDENT TYPE

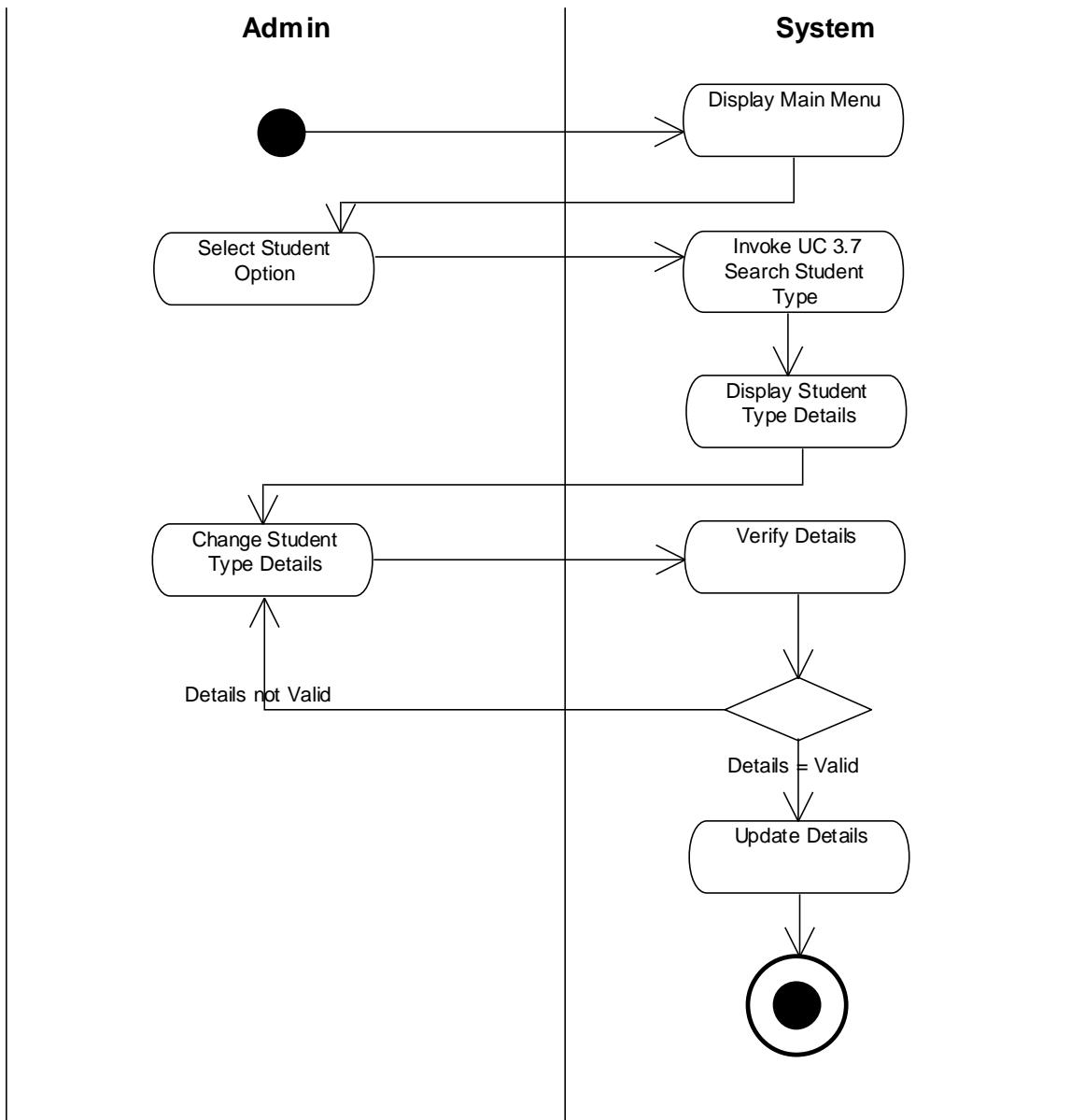


FIGURE 187- ACTIVITY DIAGRAM 3.8 UPDATE STUDENT TYPE

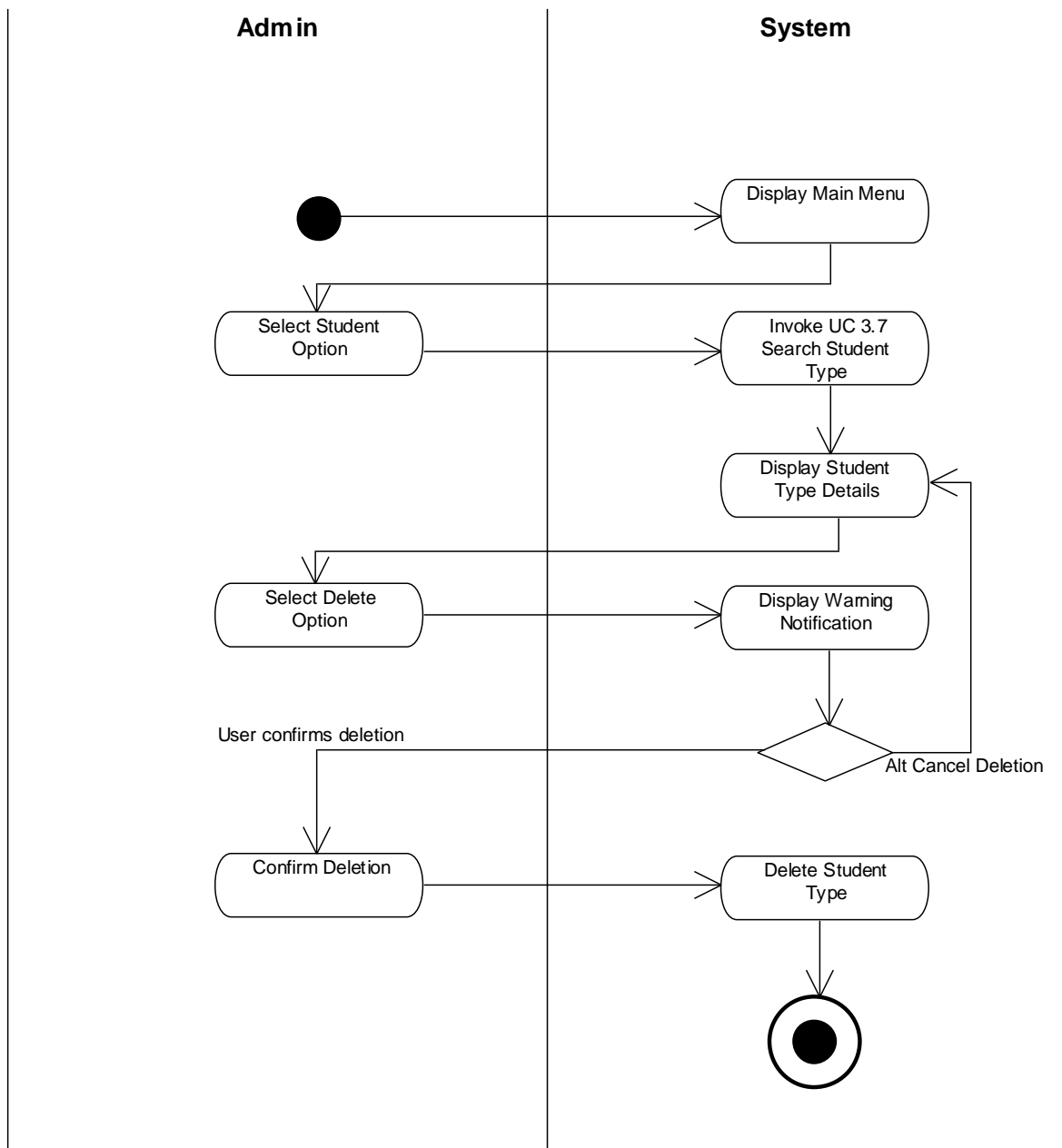


FIGURE 188- ACTIVITY DIAGRAM 3.9 DELETE STUDENT TYPE



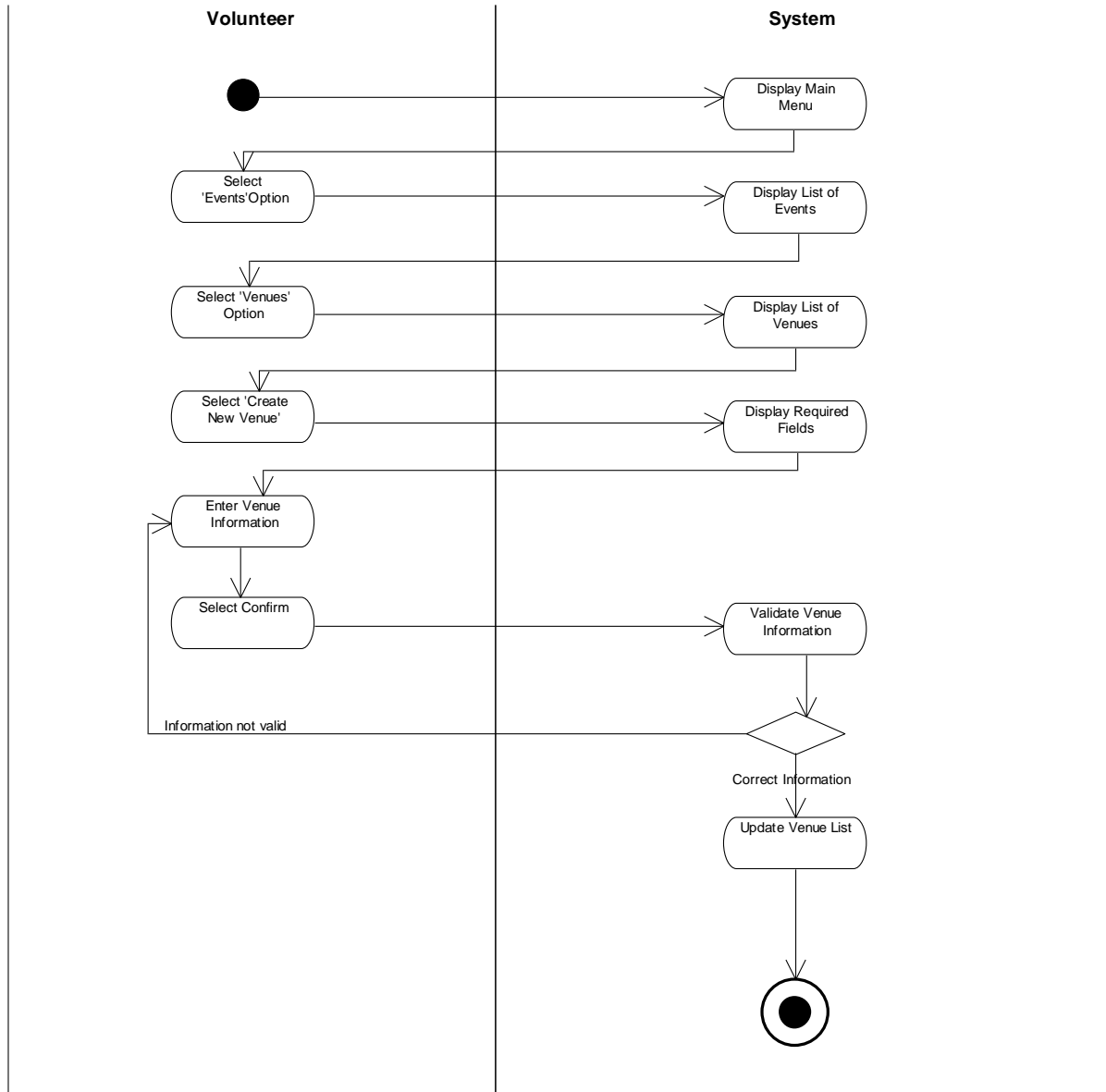


FIGURE 189- ACTIVITY DIAGRAM 4.1 ADD NEW VENUE

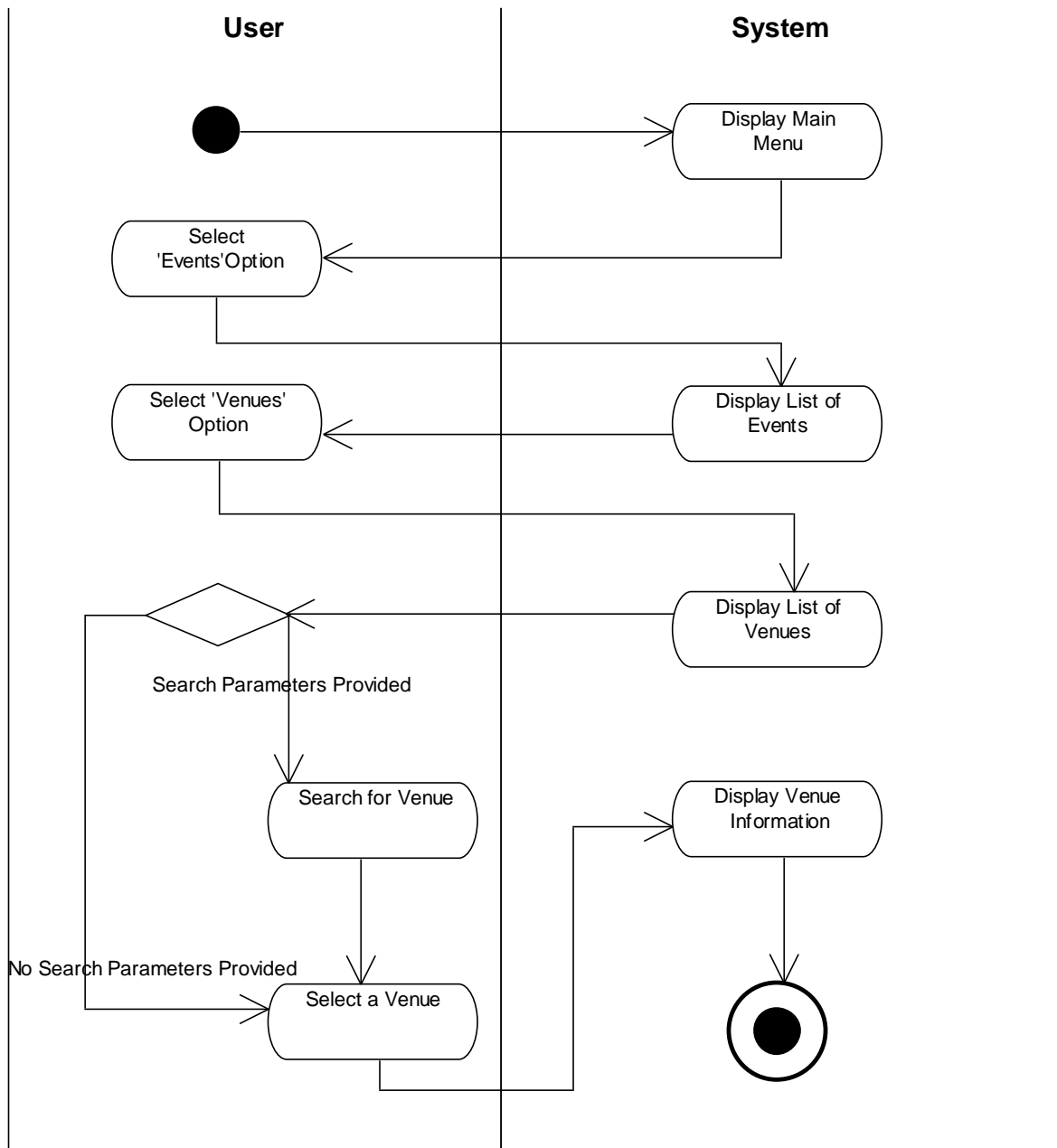


FIGURE 190- ACTIVITY DIAGRAM 4.2 SEARCH VENUE

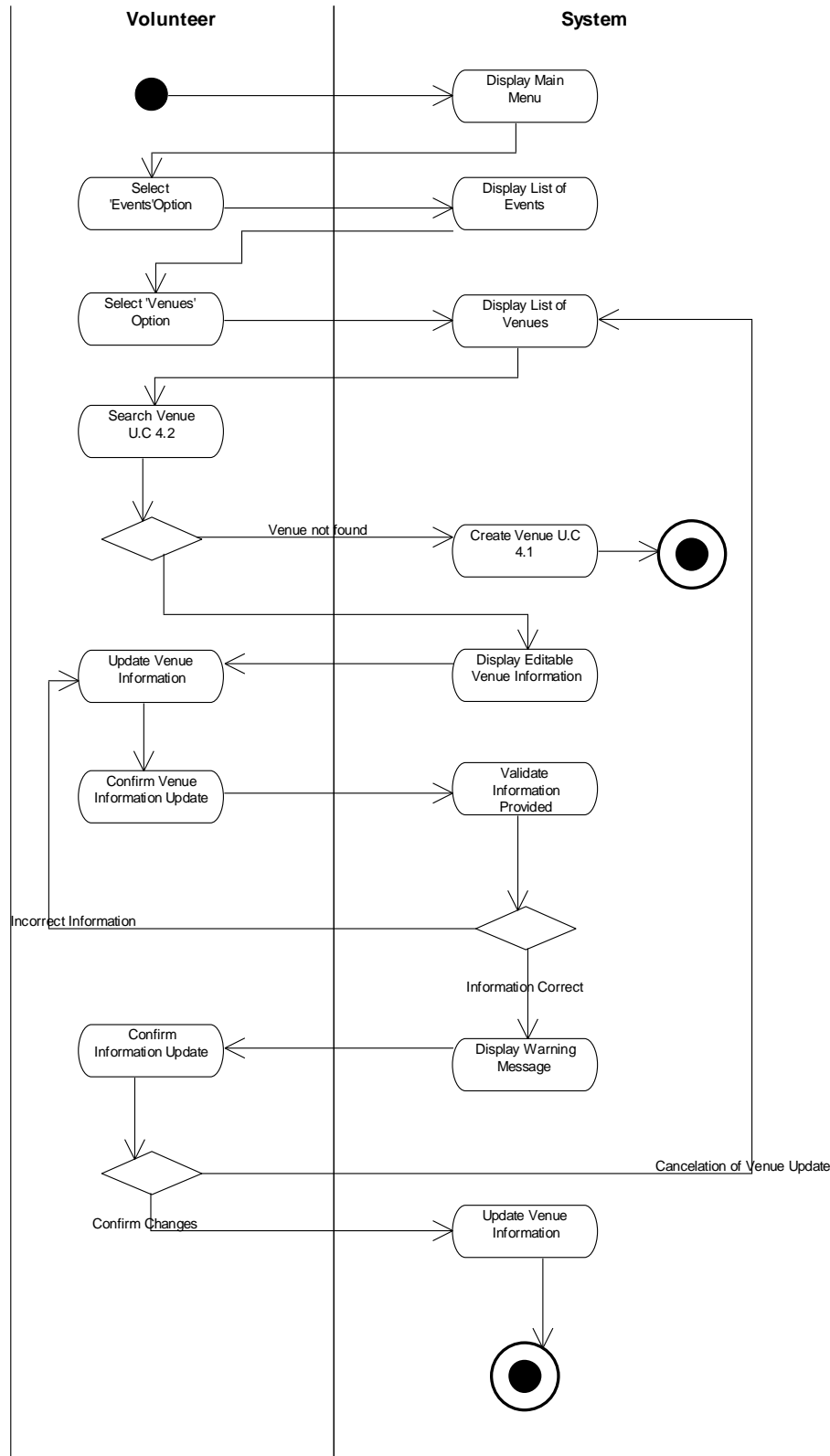


FIGURE 191- ACTIVITY DIAGRAM 4.3 UPDATE VENUE

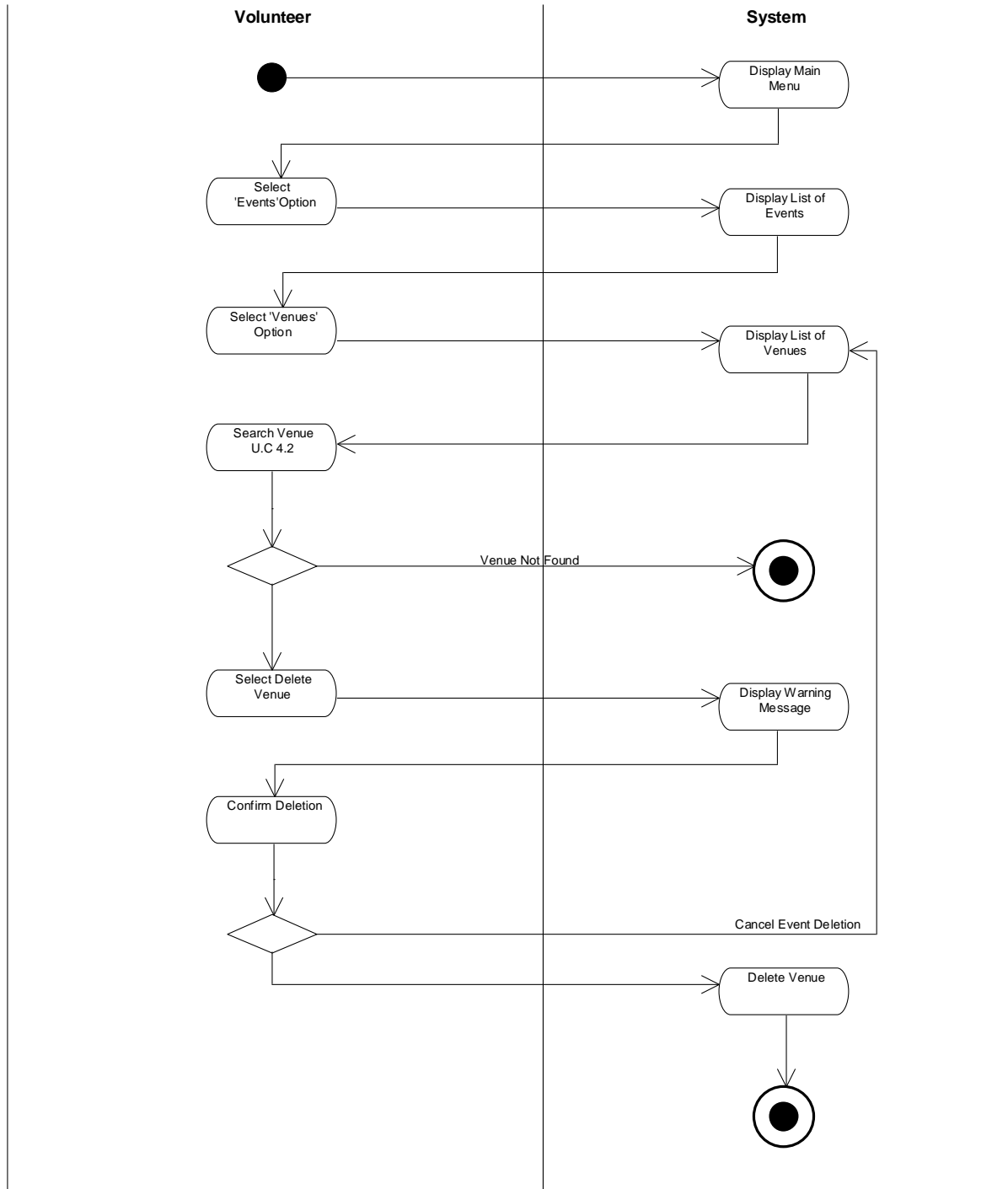


FIGURE 192- ACTIVITY DIAGRAM 4.4 DELETE VENUE

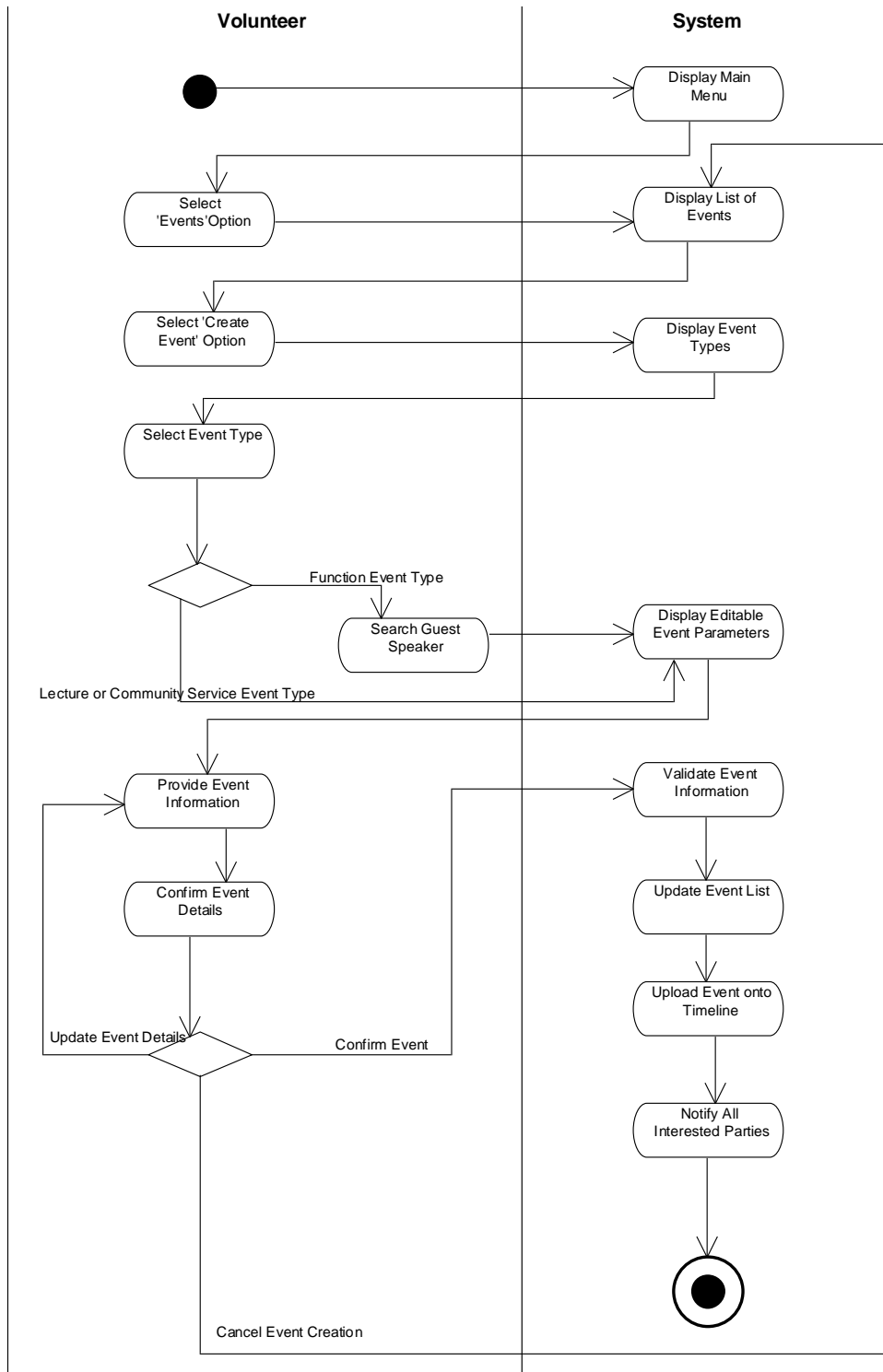


FIGURE 193- ACTIVITY DIAGRAM 4.5 CREATE NEW EVENT

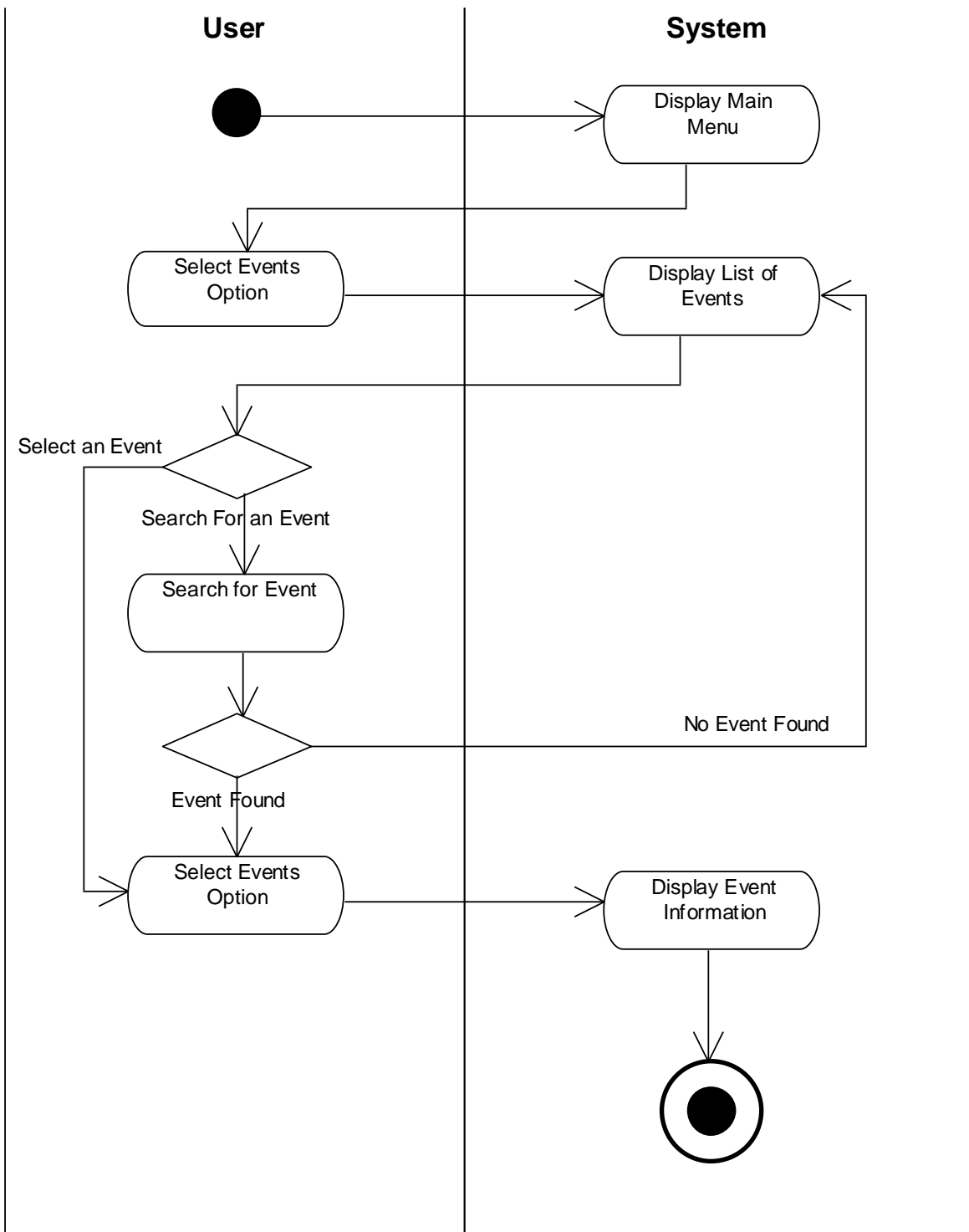


FIGURE 194- ACTIVITY DIAGRAM 4.6 SEARCH EVENT

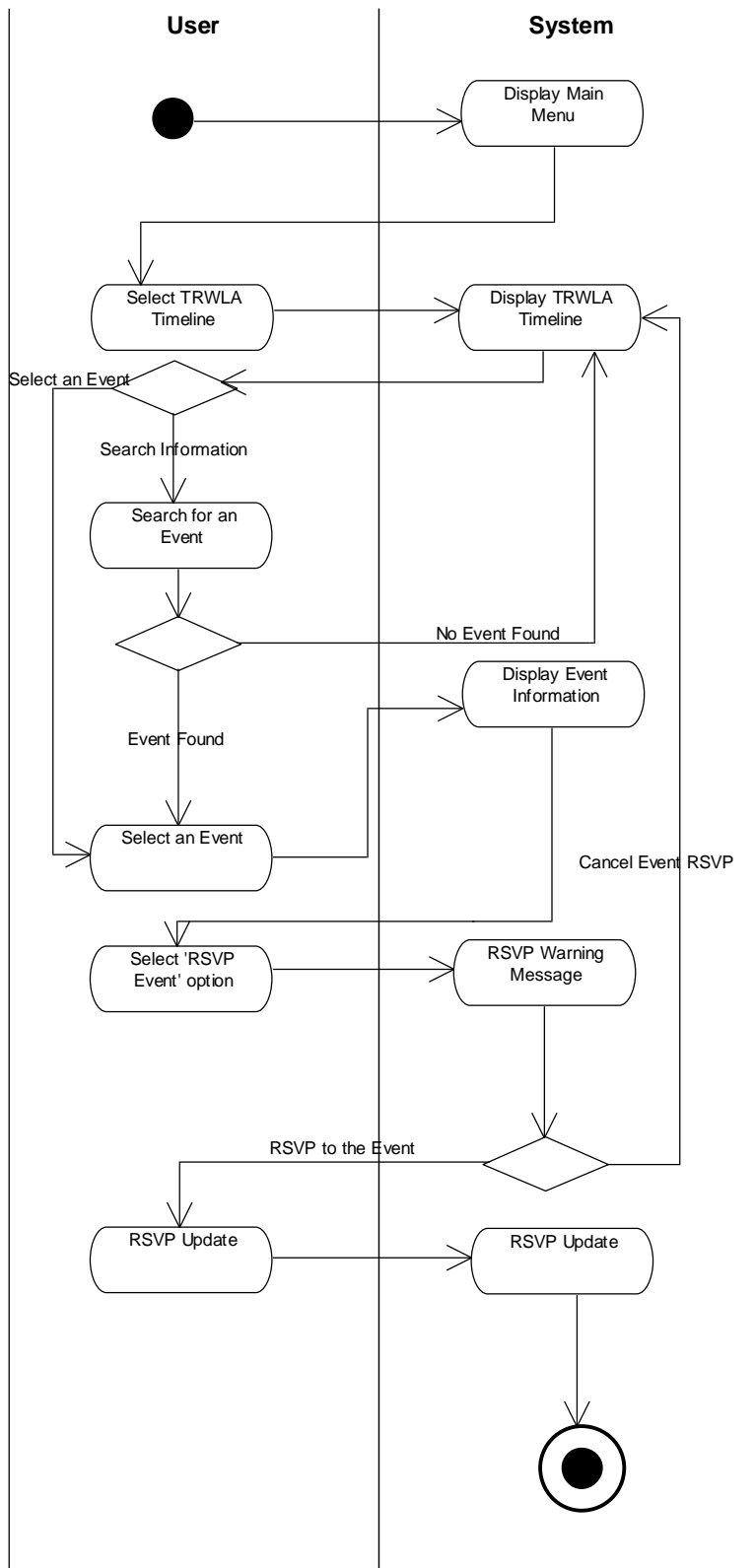


FIGURE 195- ACTIVITY DIAGRAM 4.7 RSVP TO AN EVENT

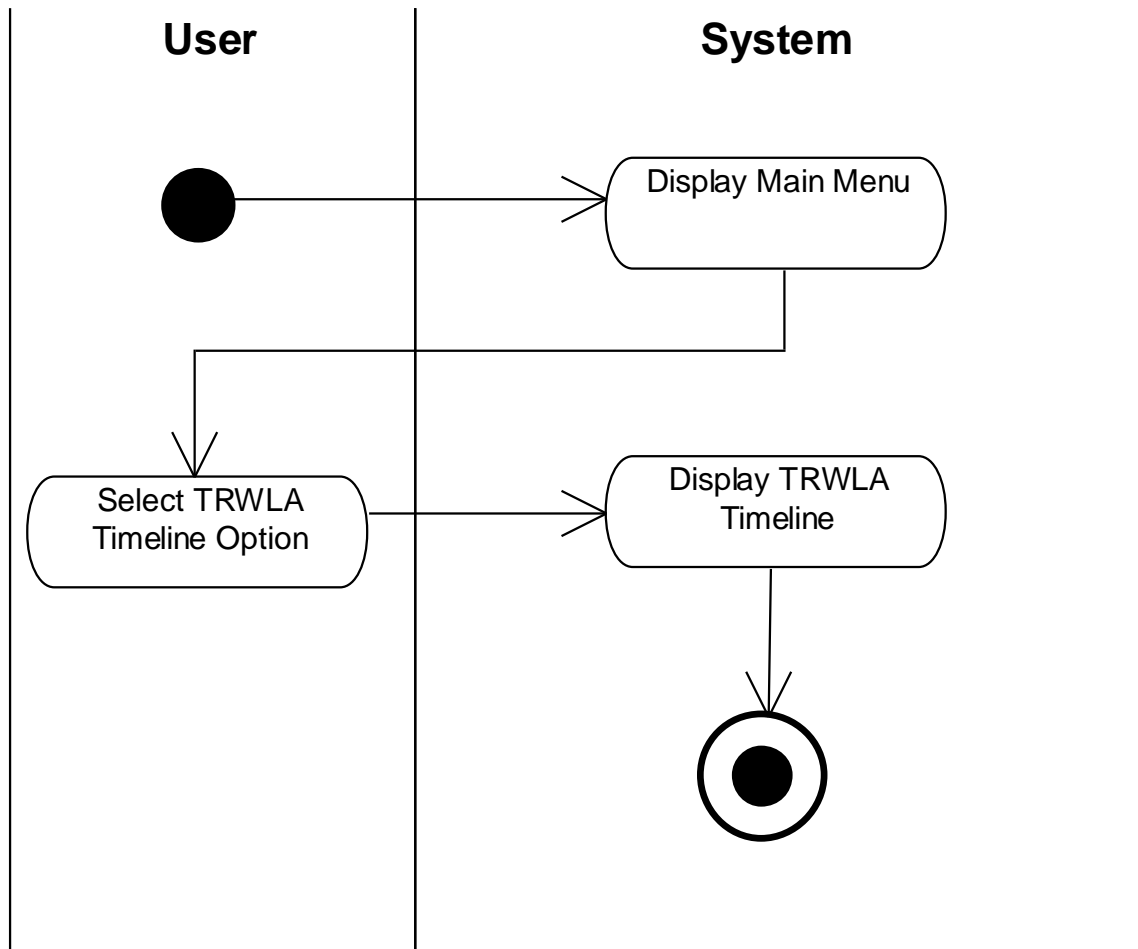


FIGURE 196- ACTIVITY DIAGRAM 4.8 VIEW TRWLA TIMELINE



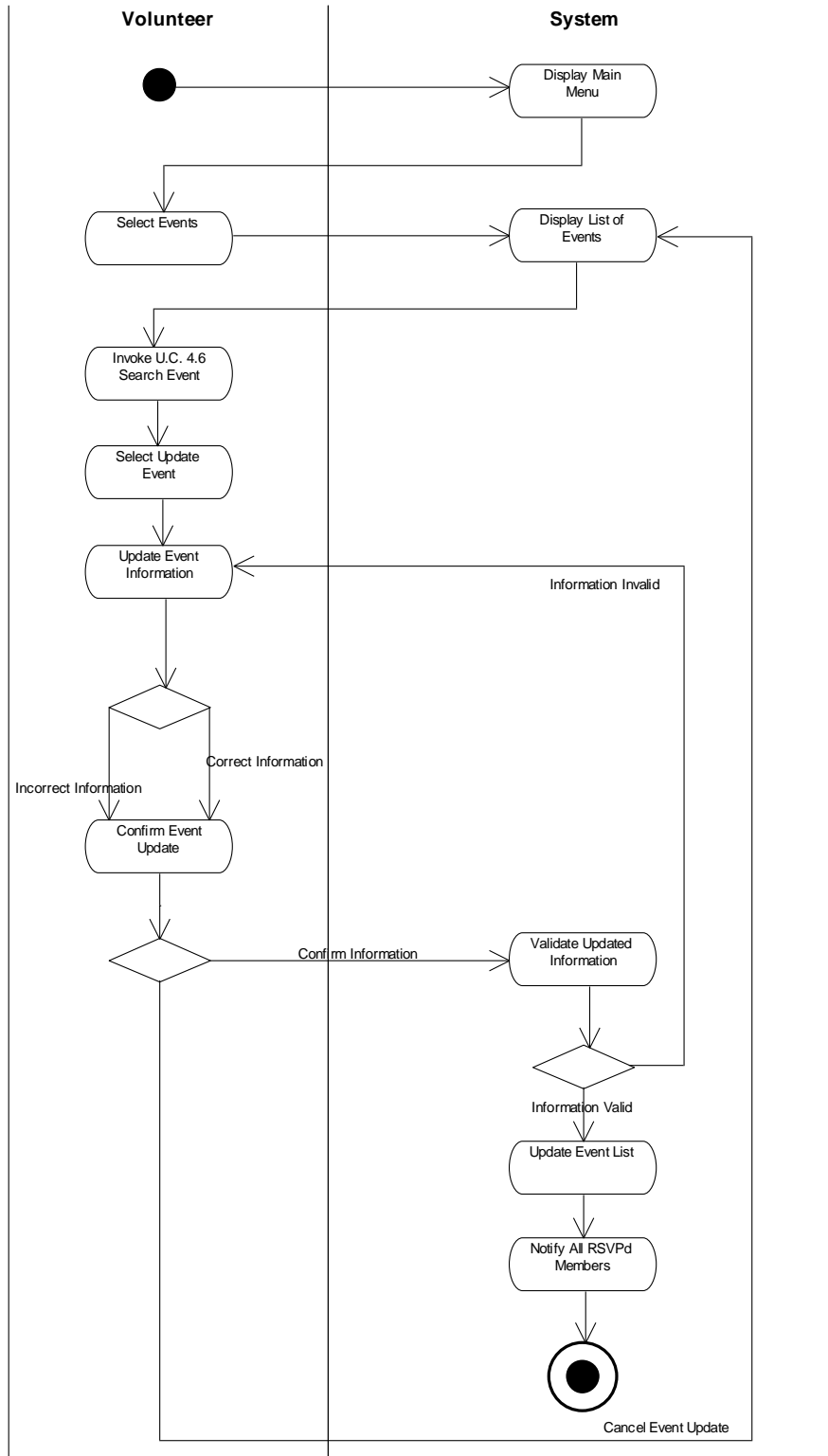


FIGURE 197- ACTIVITY DIAGRAM 4.9 UPDATE EVENT INFORMATION

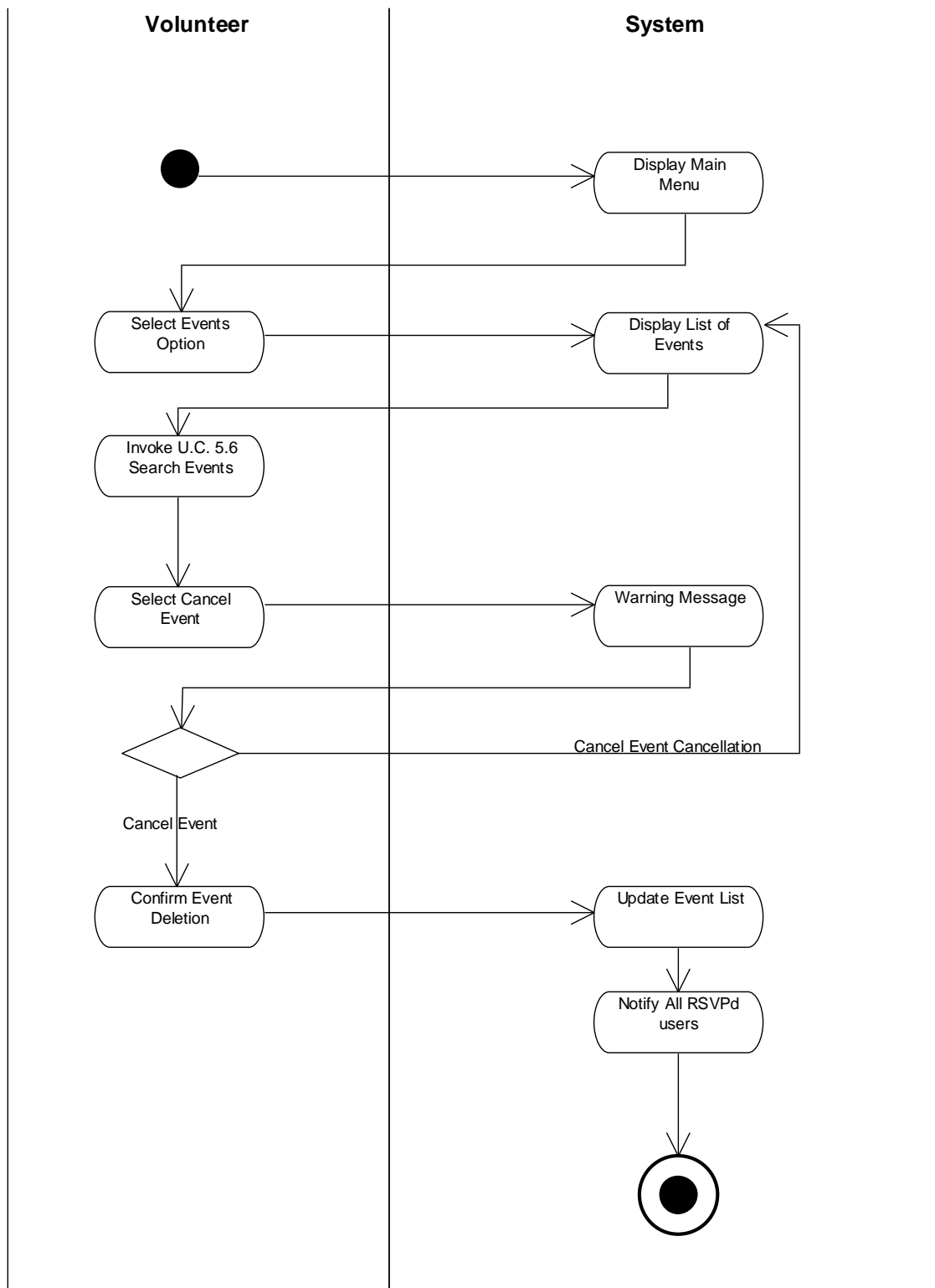


FIGURE 198- ACTIVITY DIAGRAM 4.10 CANCEL EVENT

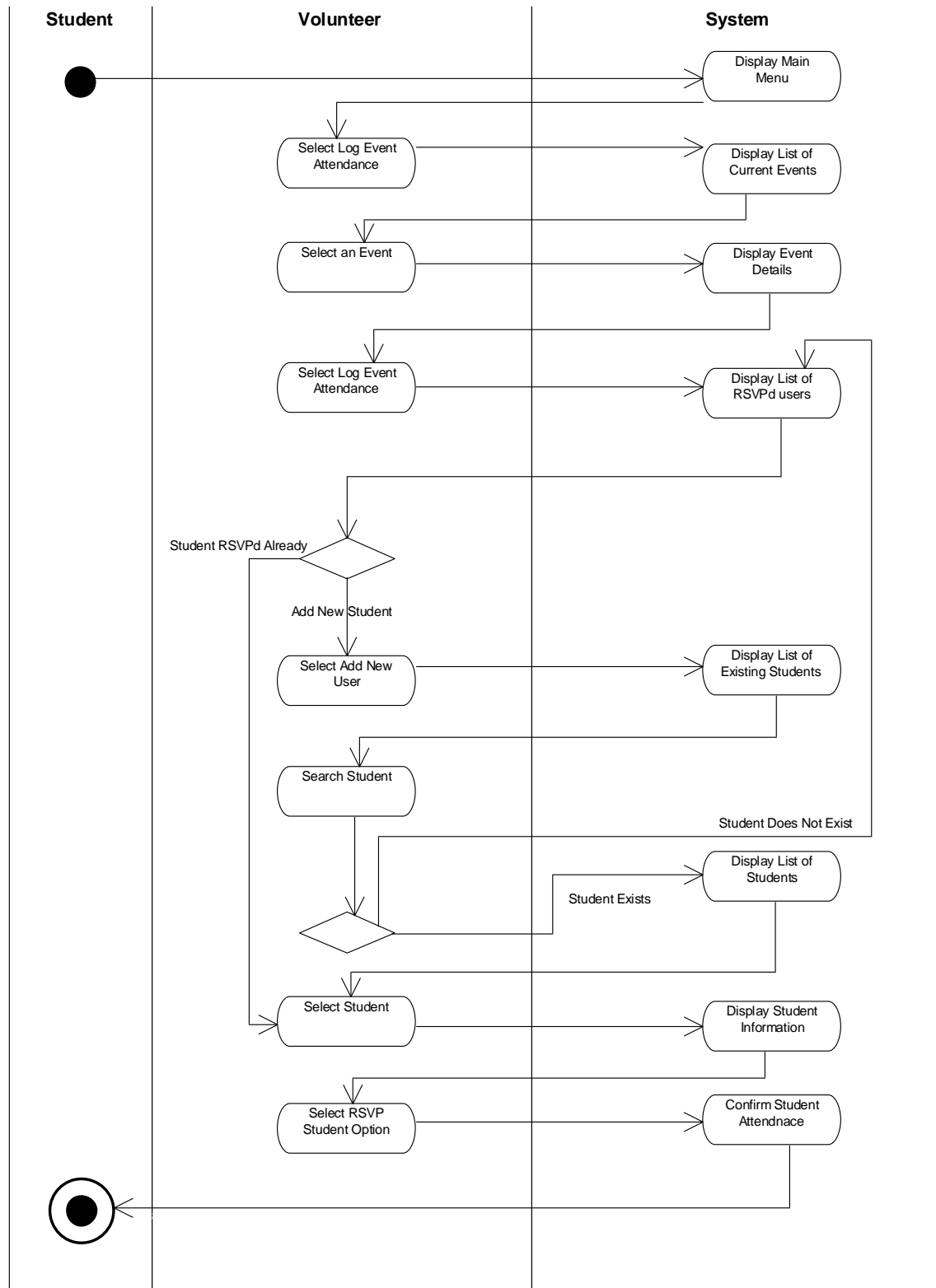


FIGURE 199- ACTIVITY DIAGRAM 4.11 LOG EVENT ATTENDANCE

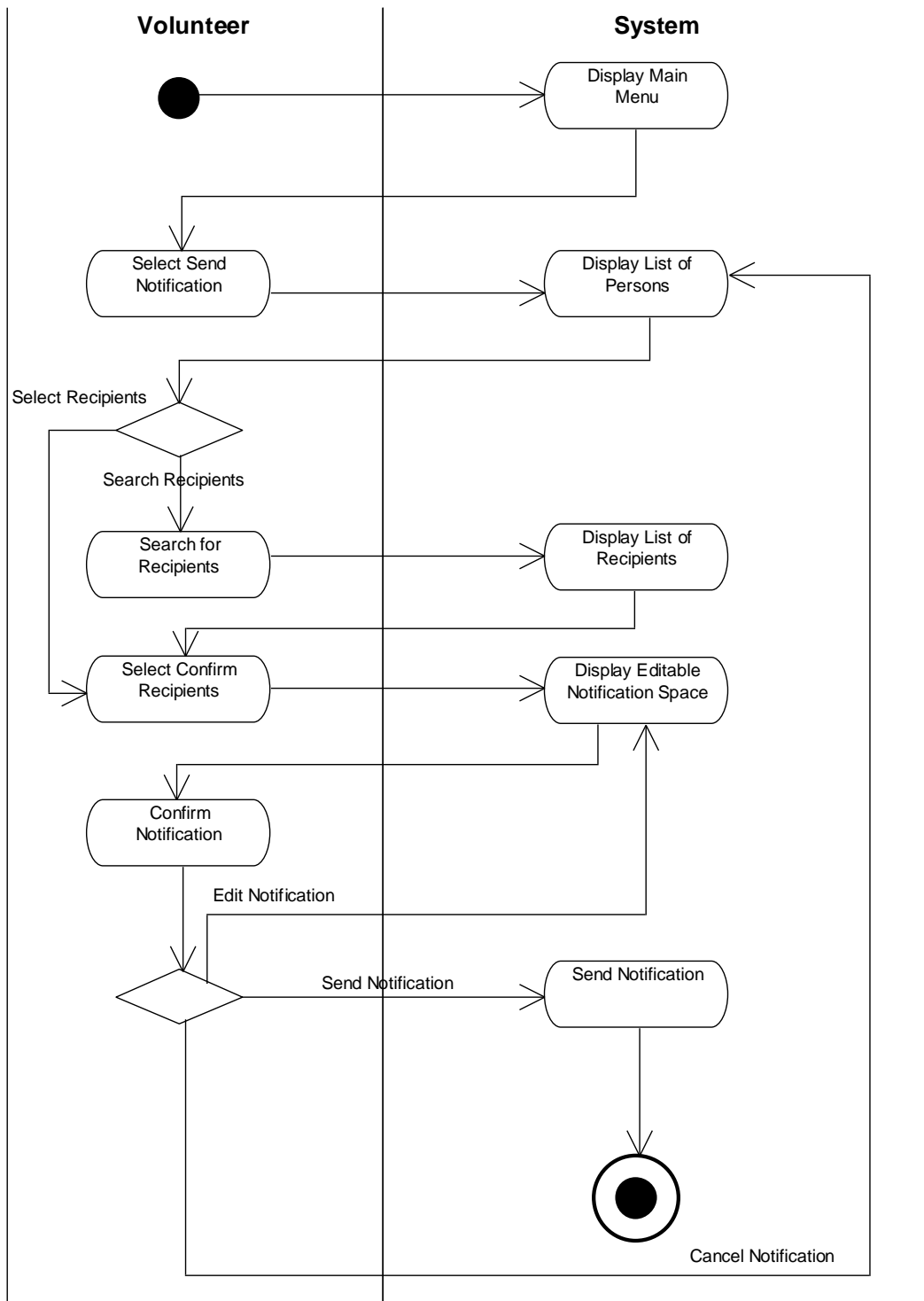


FIGURE 200- ACTIVITY DIAGRAM 4.12 SEND NOTIFICATION

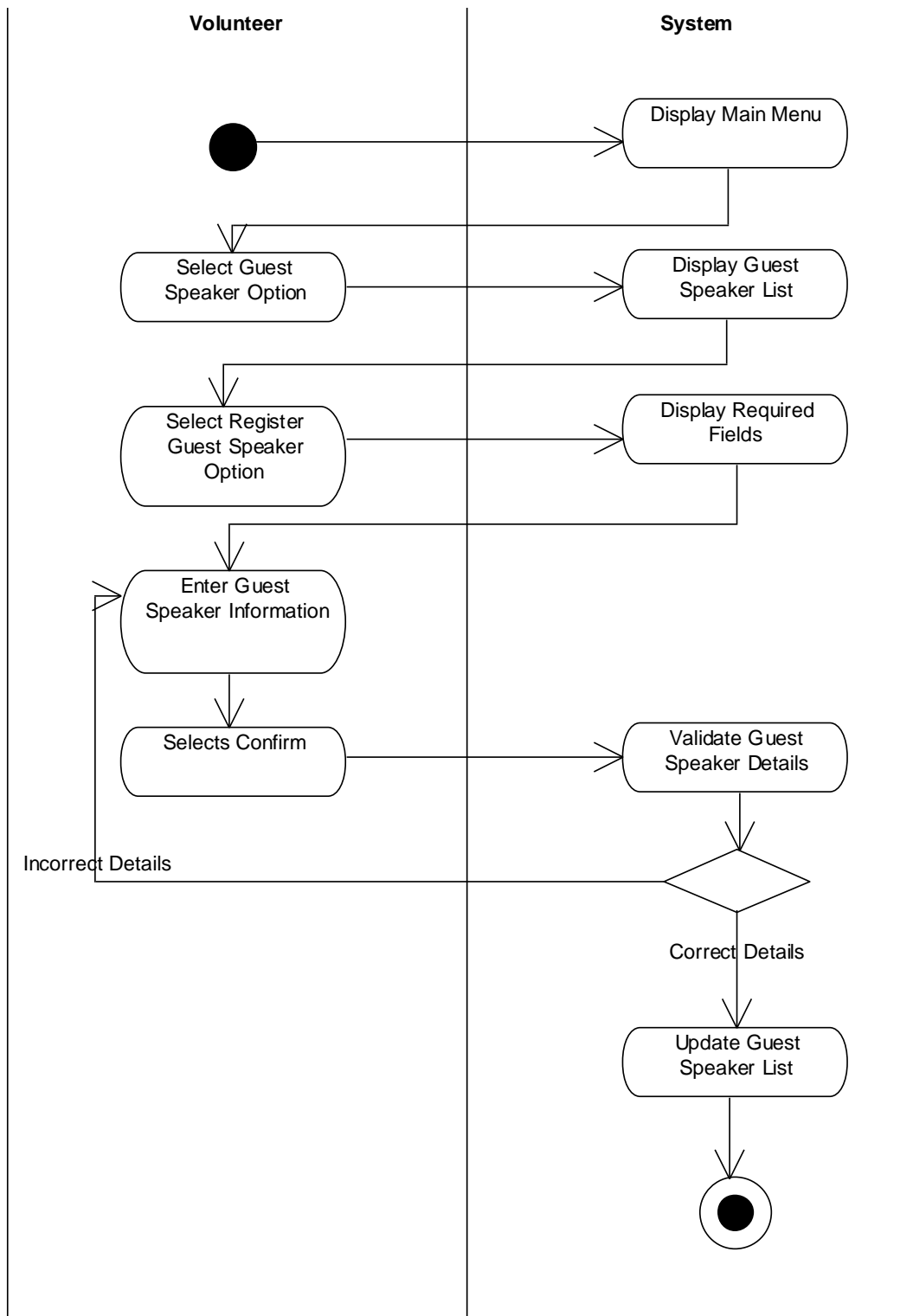


FIGURE 201- ACTIVITY DIAGRAM 5.1 REGISTER GUEST SPEAKER

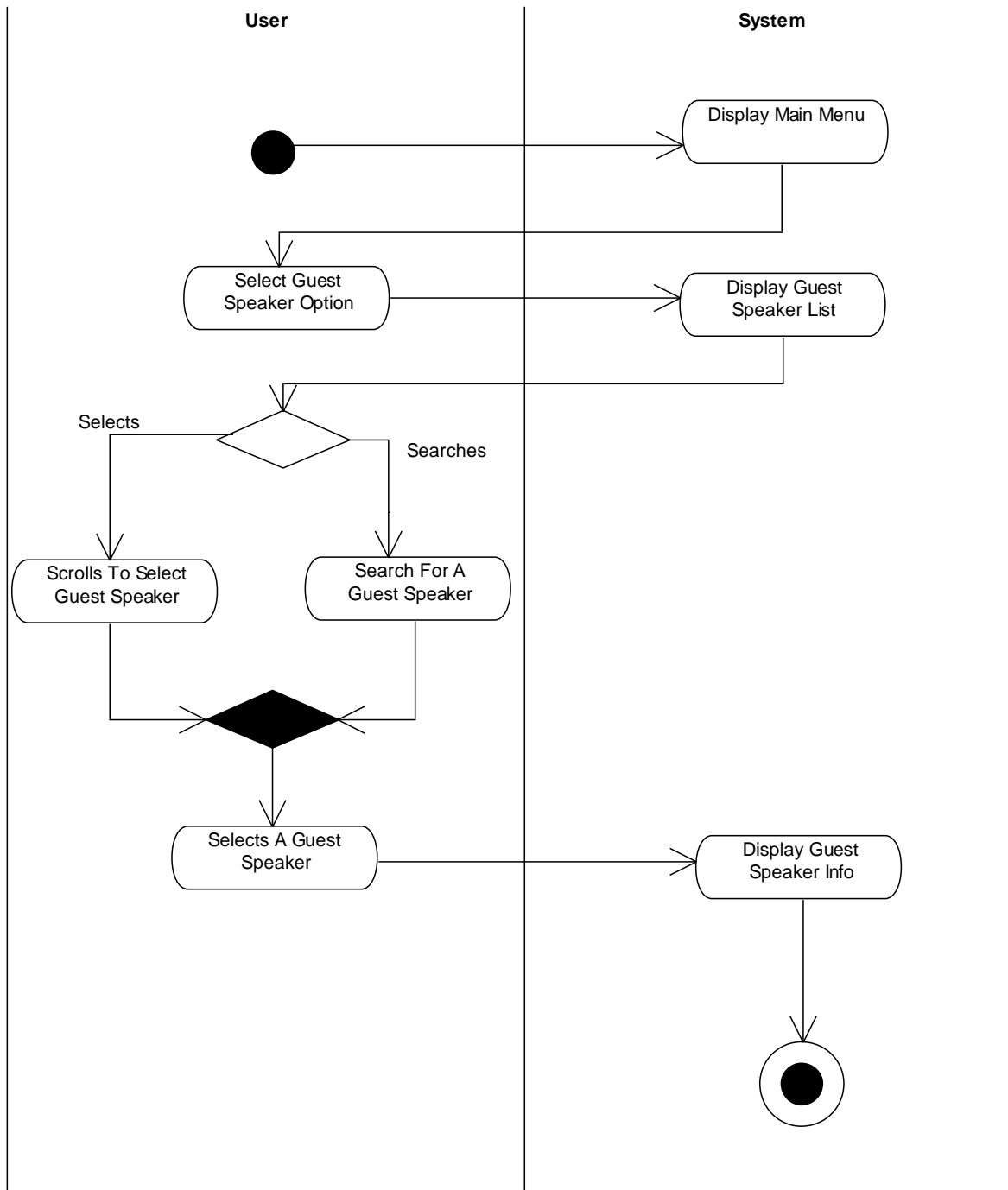


FIGURE 202- ACTIVITY DIAGRAM 5.2 SEARCH GUEST SPEAKER

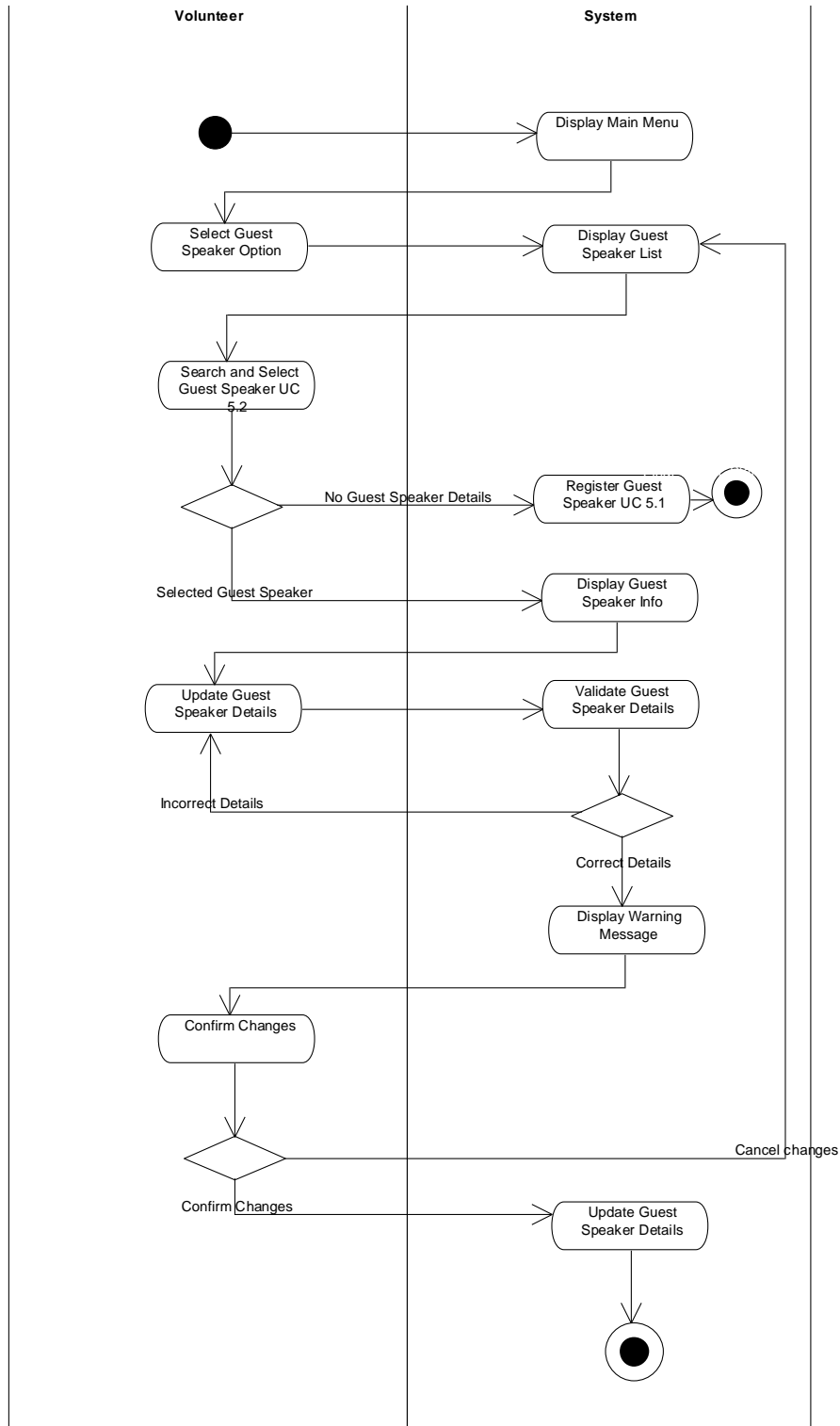


FIGURE 203- ACTIVITY DIAGRAM 5.3 UPDATE GUEST SPEAKER

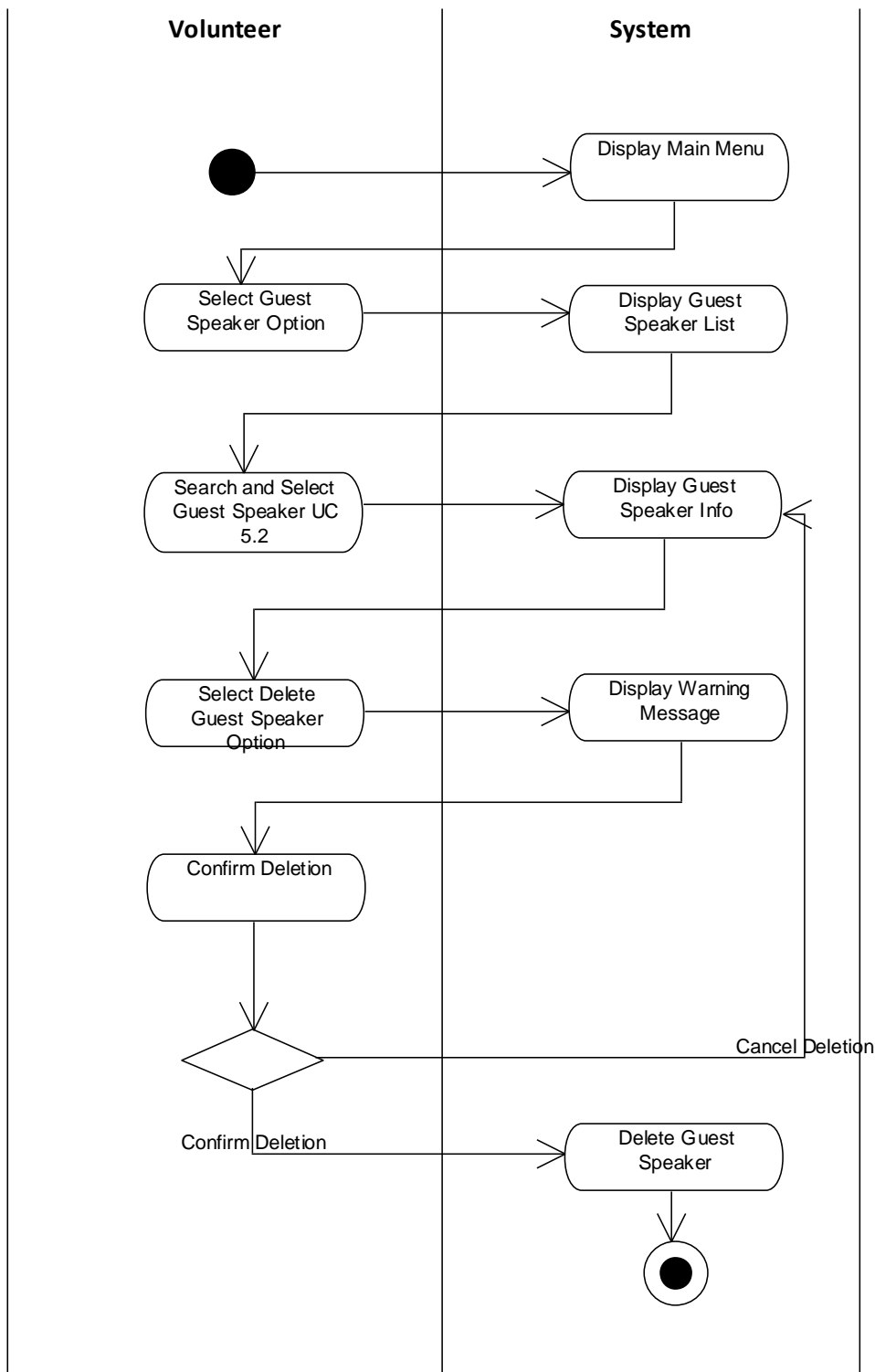


FIGURE 204- ACTIVITY DIAGRAM 5.4 DELETE GUEST SPEAKER



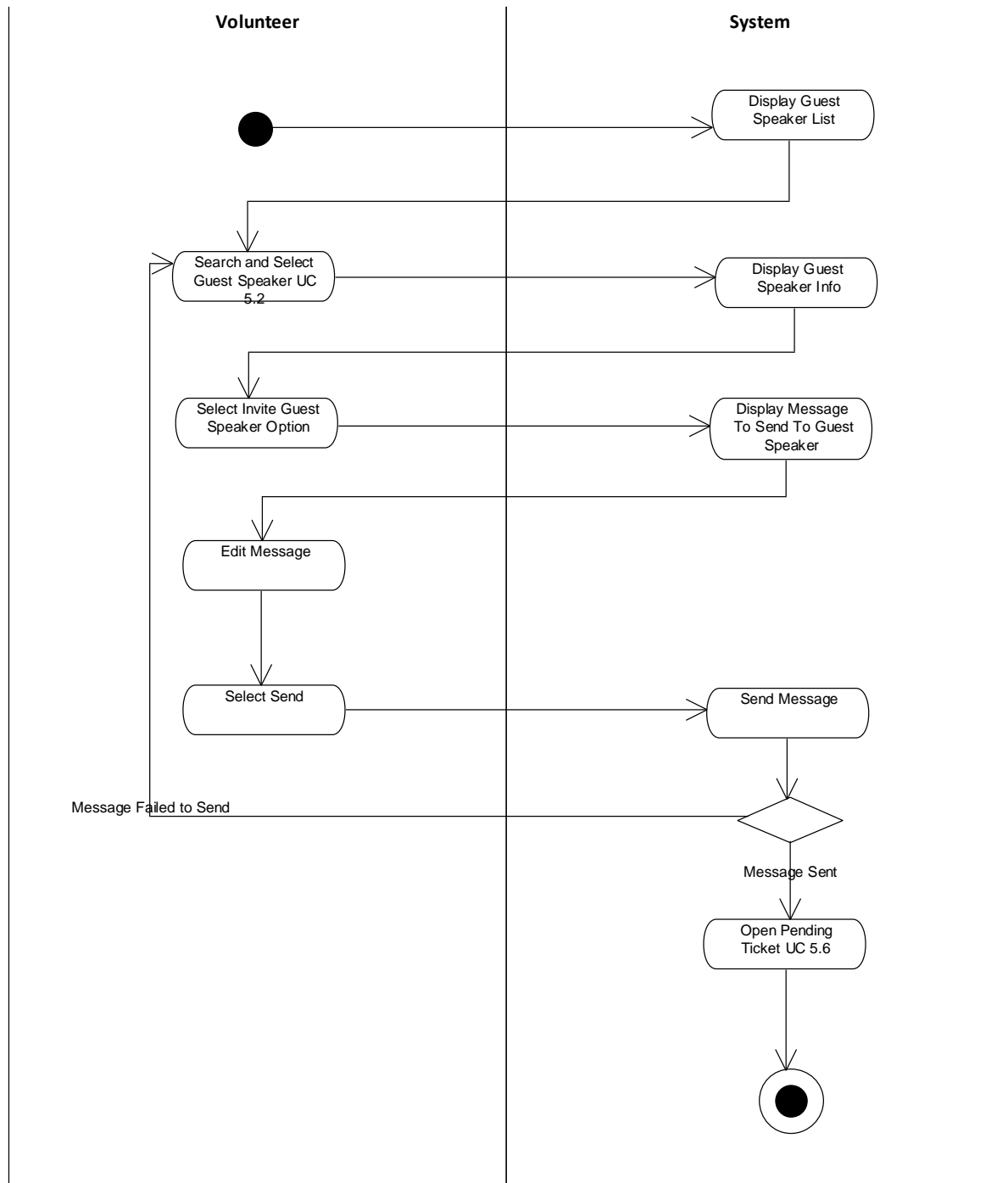


FIGURE 205- ACTIVITY DIAGRAM 5.5 INVITE GUEST SPEAKER

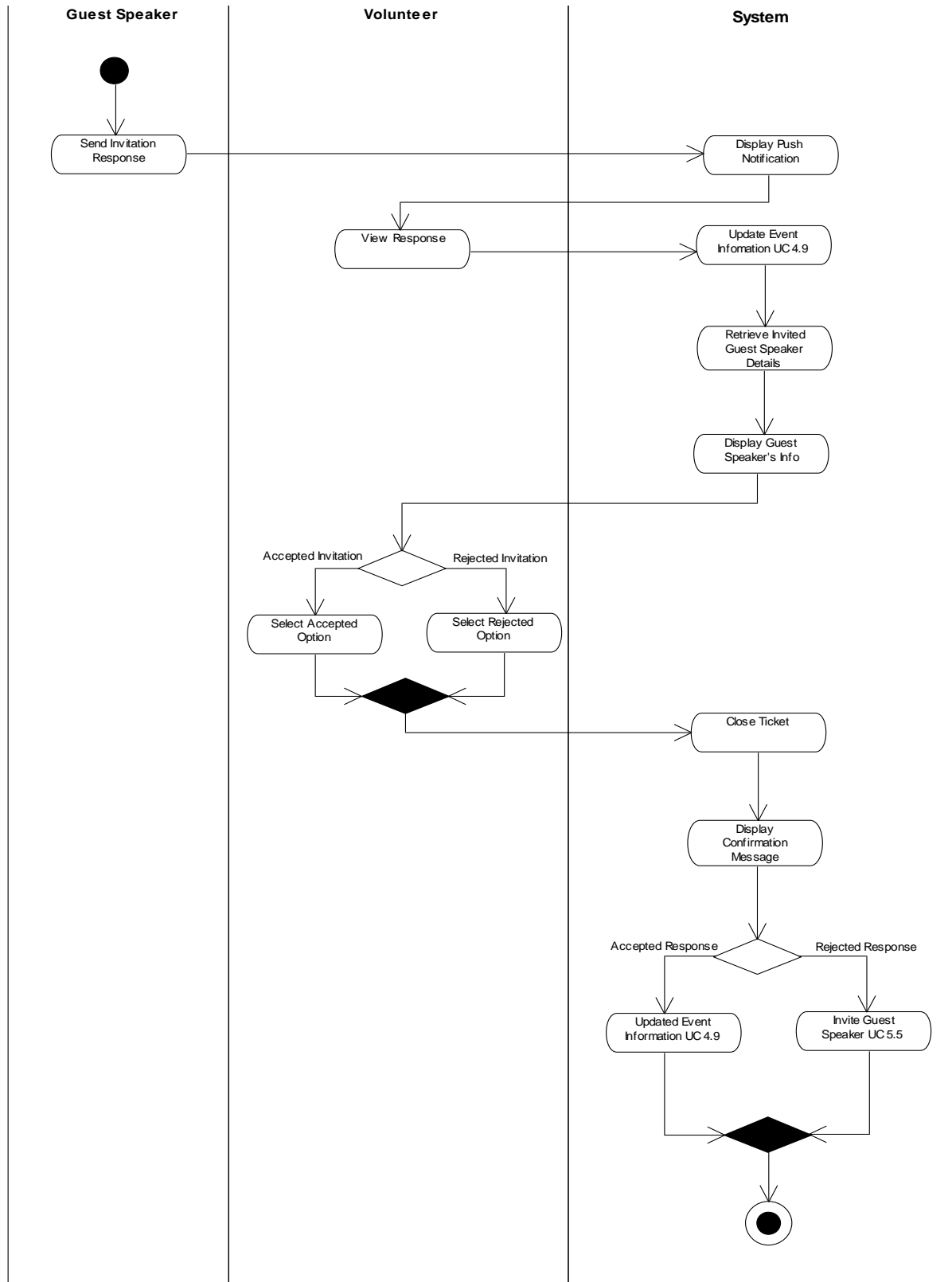


FIGURE 206- ACTIVITY DIAGRAM 5.6 CONFIRM GUEST SPEAKER

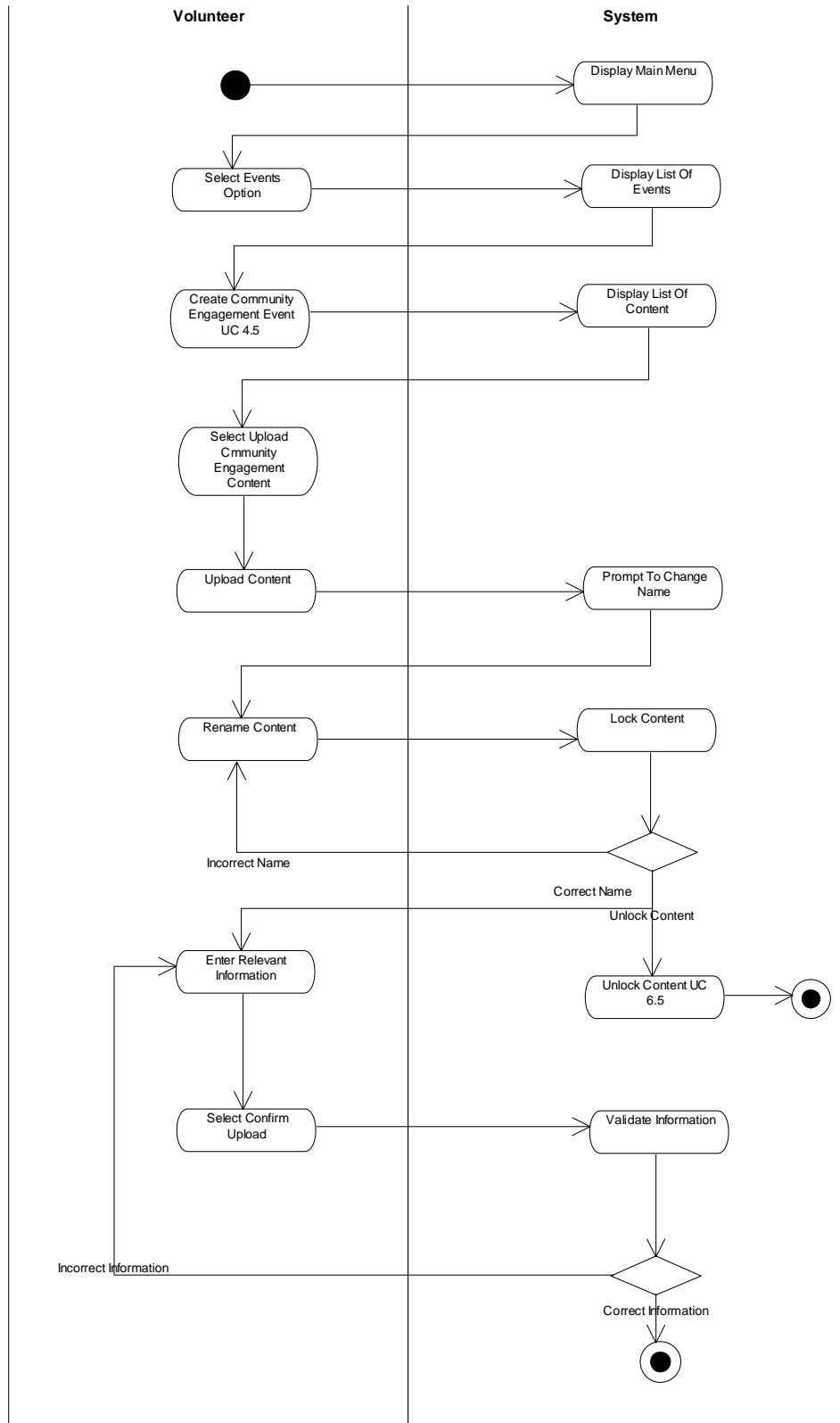


FIGURE 207- ACTIVITY DIAGRAM 6.1 UPLOAD COMMUNITY ENGAGEMENT CONTENT

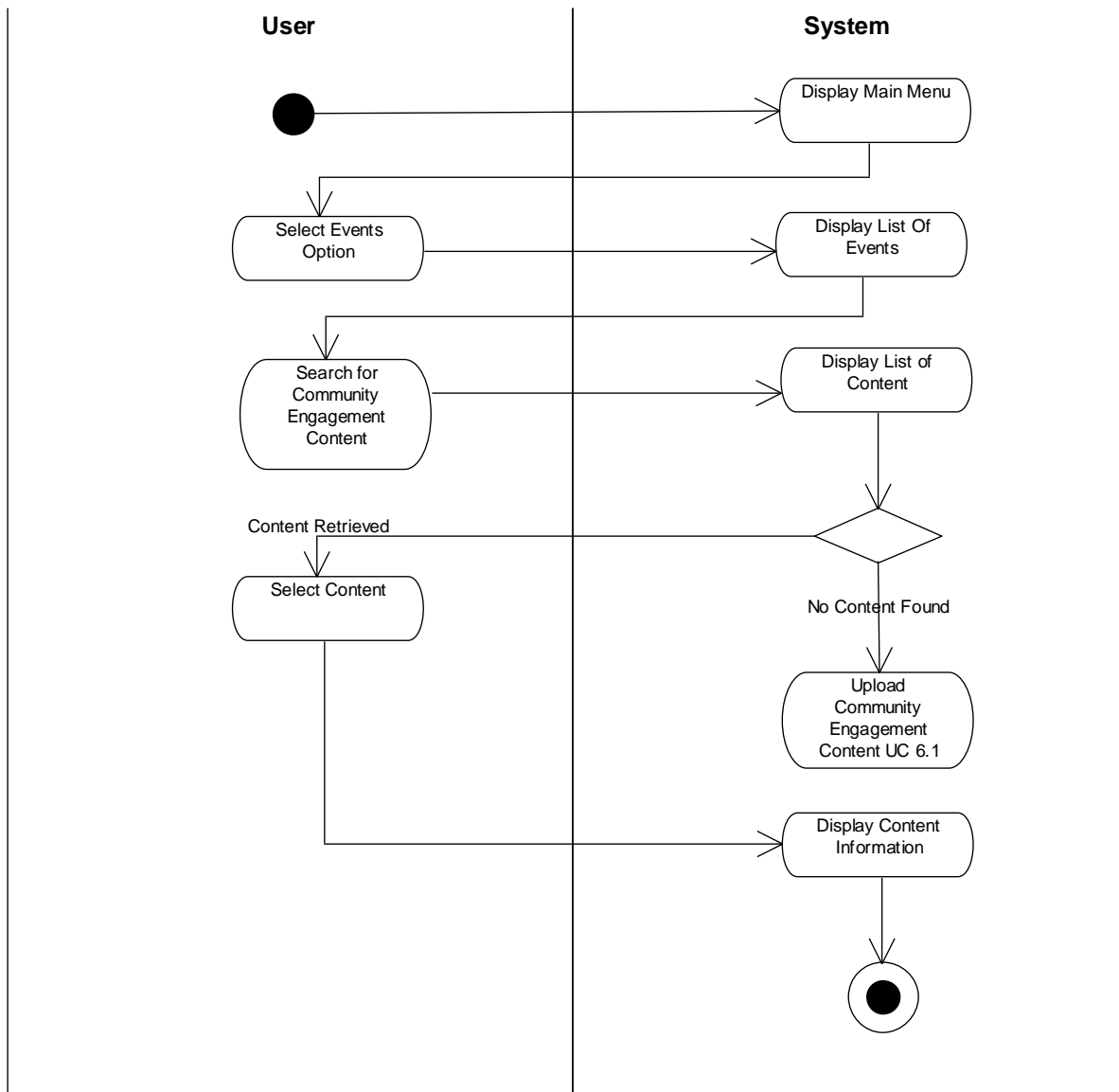


FIGURE 208- ACTIVITY DIAGRAM 6.2 SEARCH COMMUNITY ENGAGEMENT CONTENT

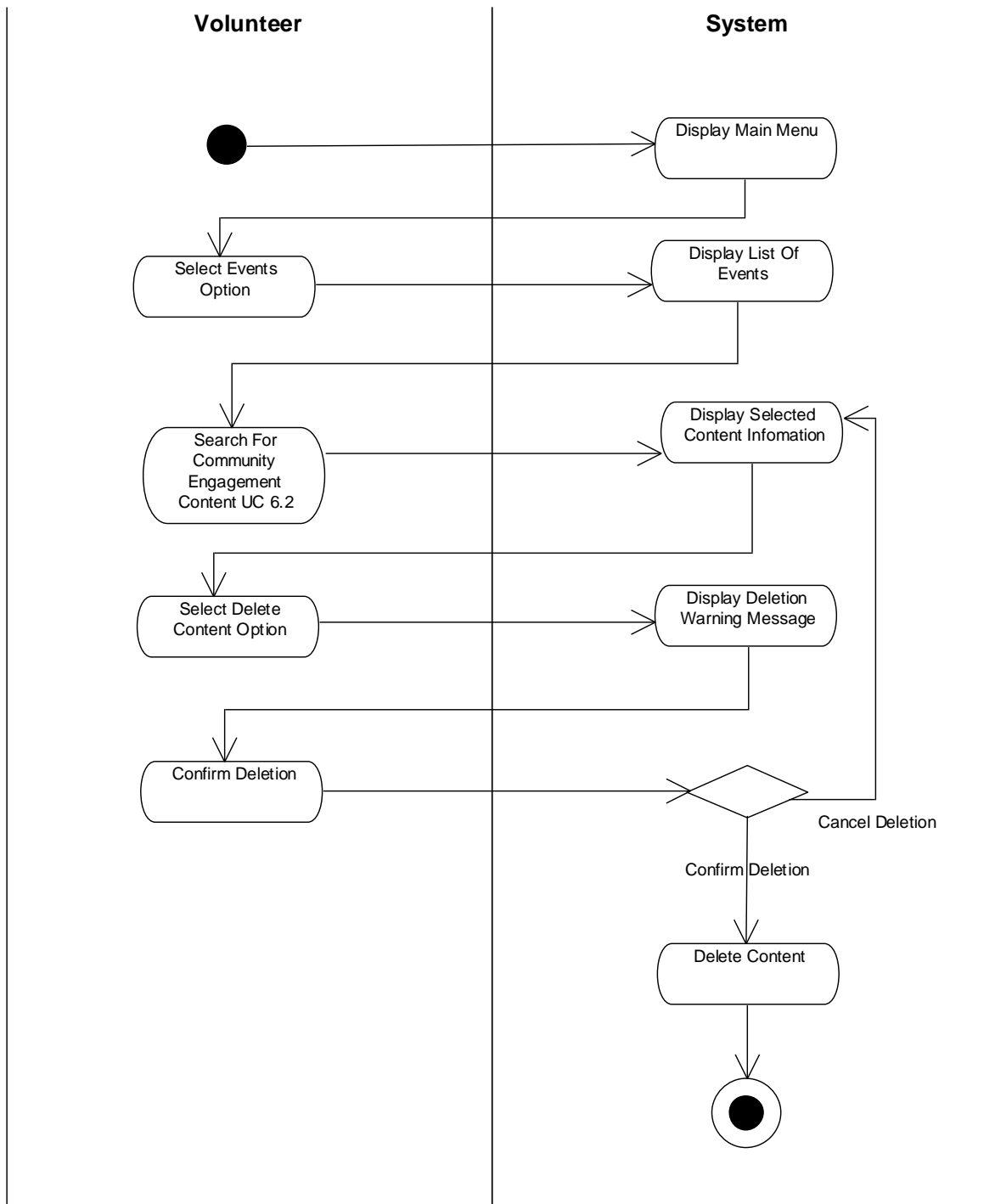


FIGURE 209- ACTIVITY DIAGRAM 6.3 DELETE COMMUNITY ENGAGEMENT CONTENT

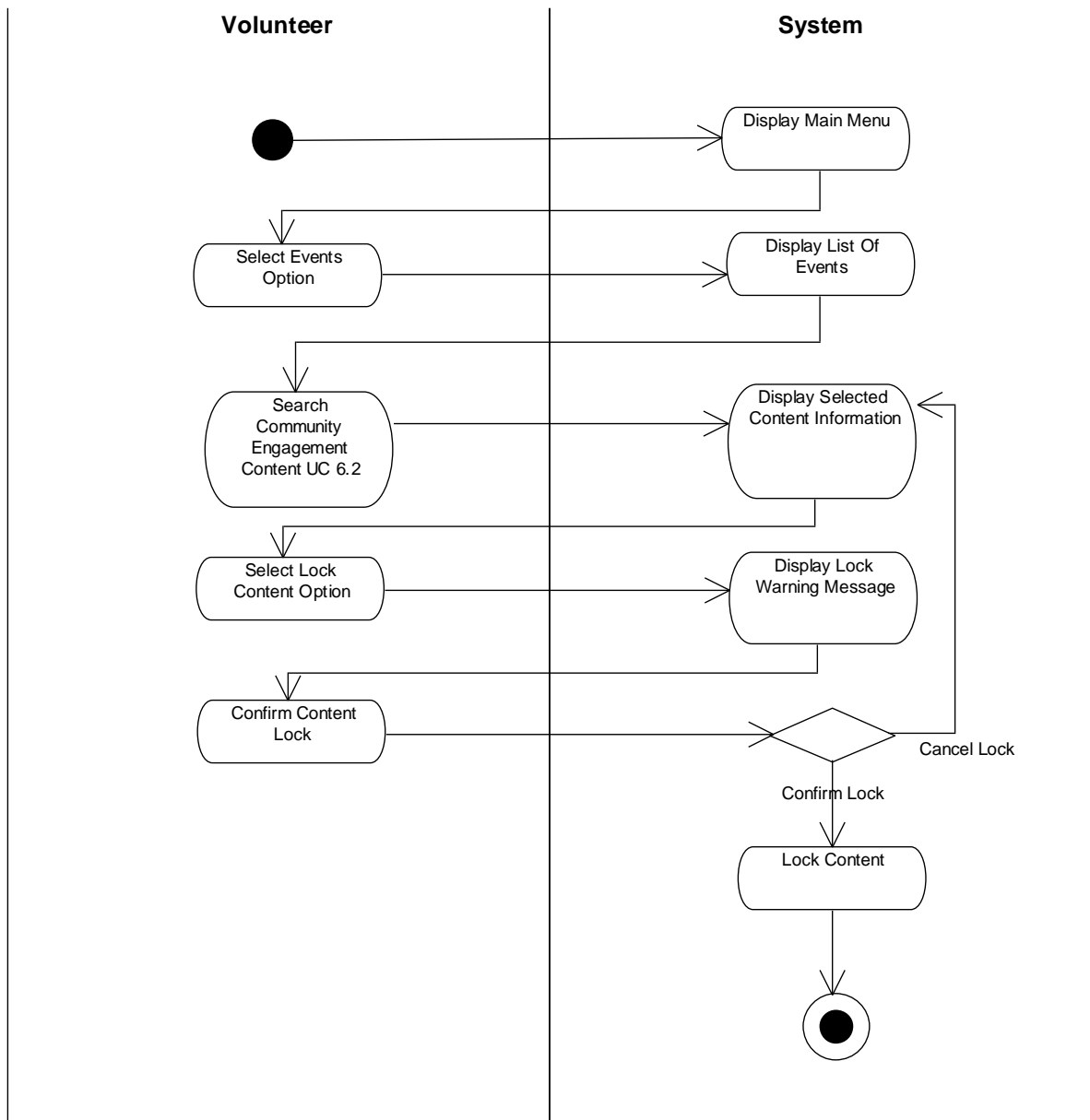


FIGURE 210- ACTIVITY DIAGRAM 6.4 LOCK COMMUNITY ENGAGEMENT CONTENT

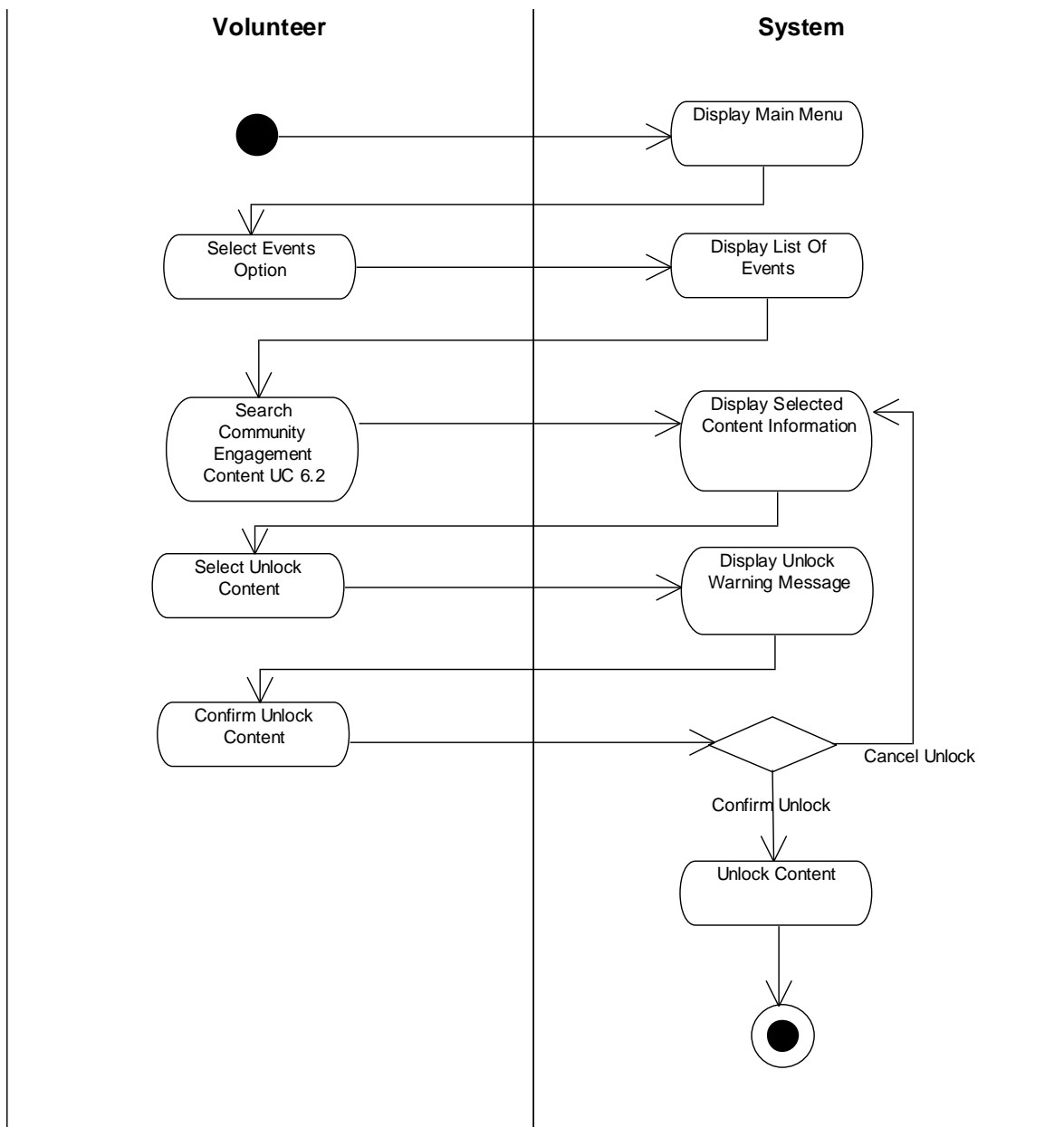


FIGURE 211- ACTIVITY DIAGRAM 6.5 UNLOCK COMMUNITY ENGAGEMENT CONTENT

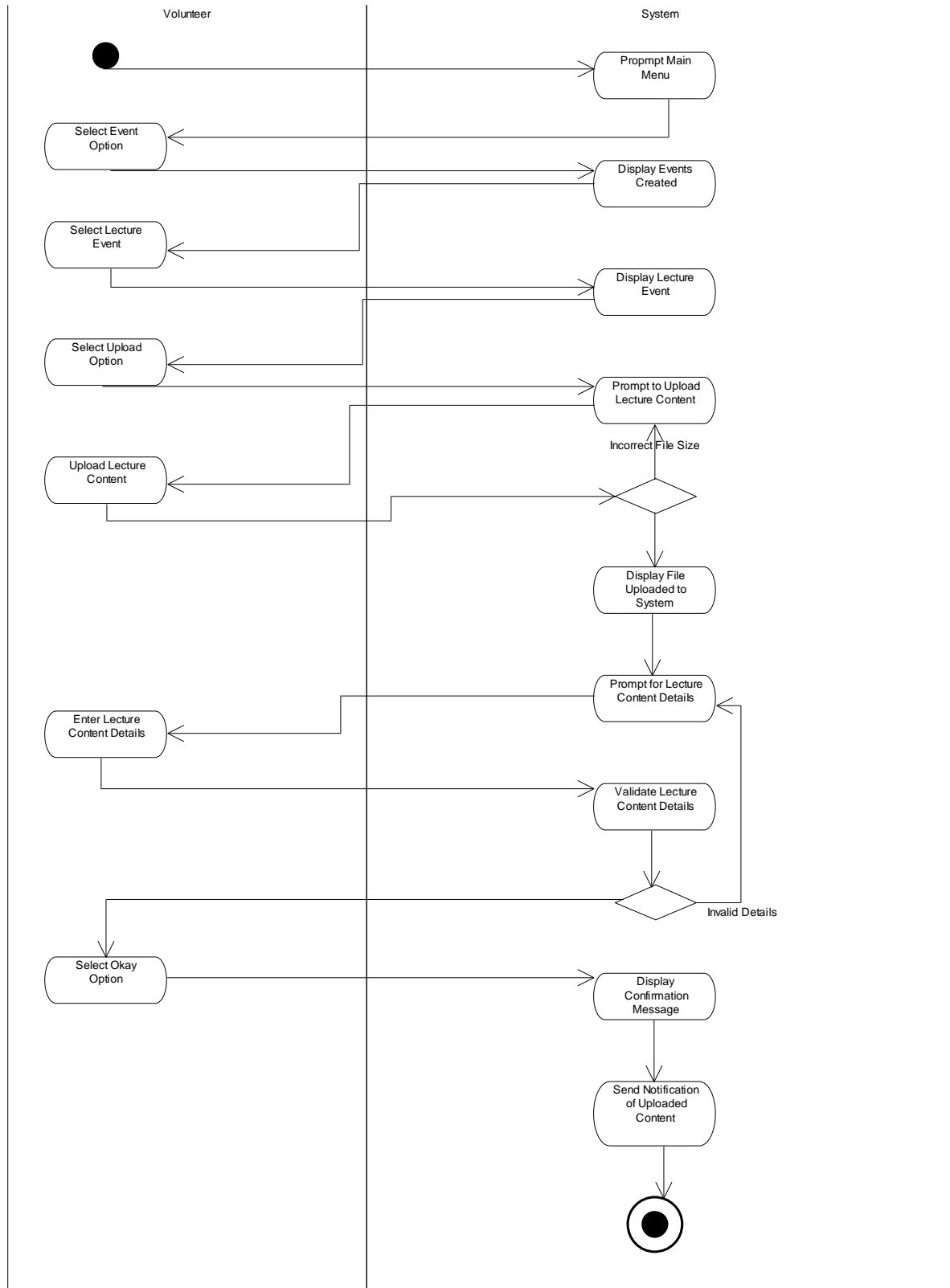


FIGURE 212- ACTIVITY DIAGRAM 7.1 UPLOAD LECTURE CONTENT



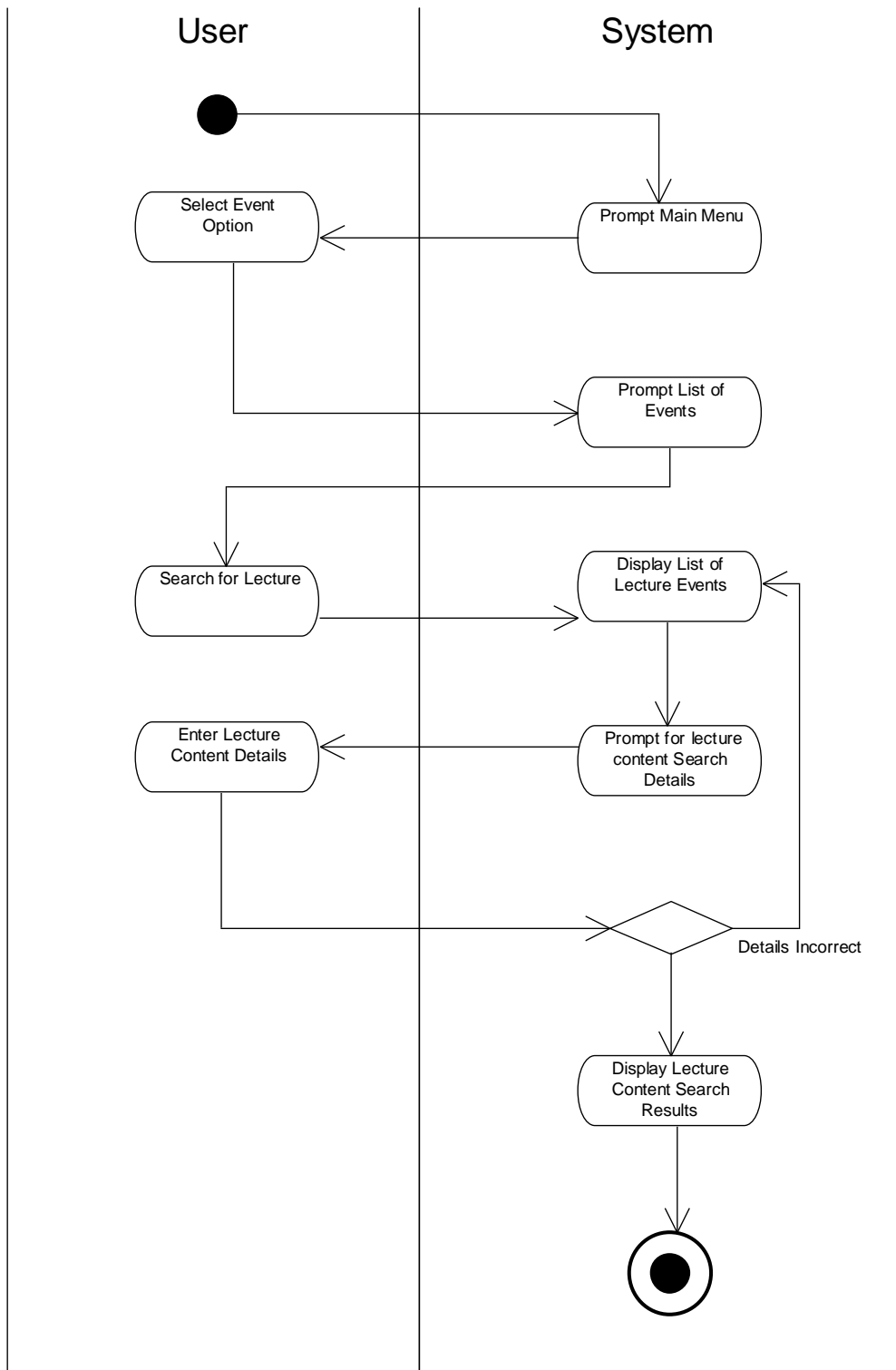


FIGURE 213- ACTIVITY DIAGRAM 7.2 SEARCH LECTURE CONTENT

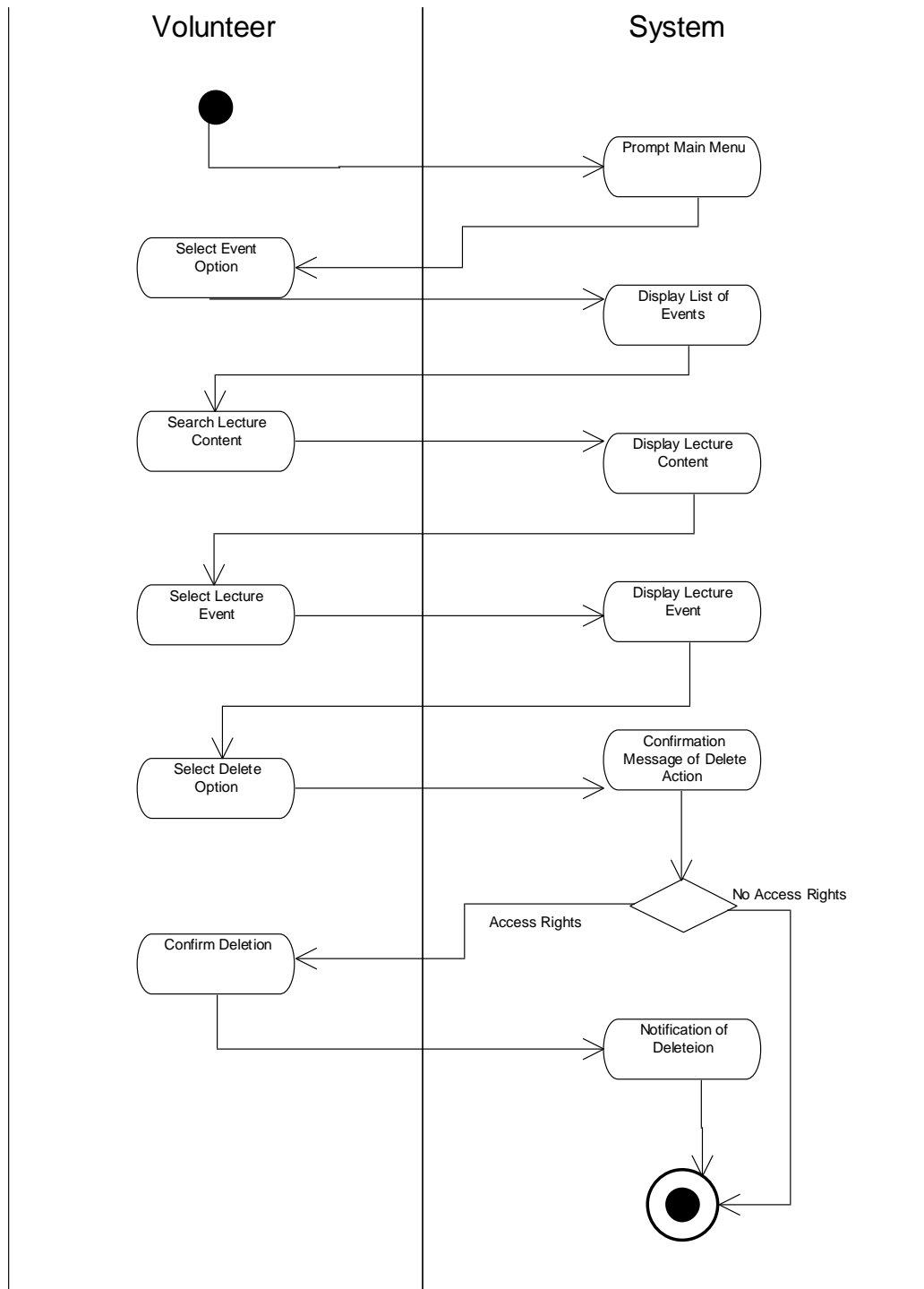


FIGURE 214- ACTIVITY DIAGRAM 7.3 DELETE LECTURE CONTENT

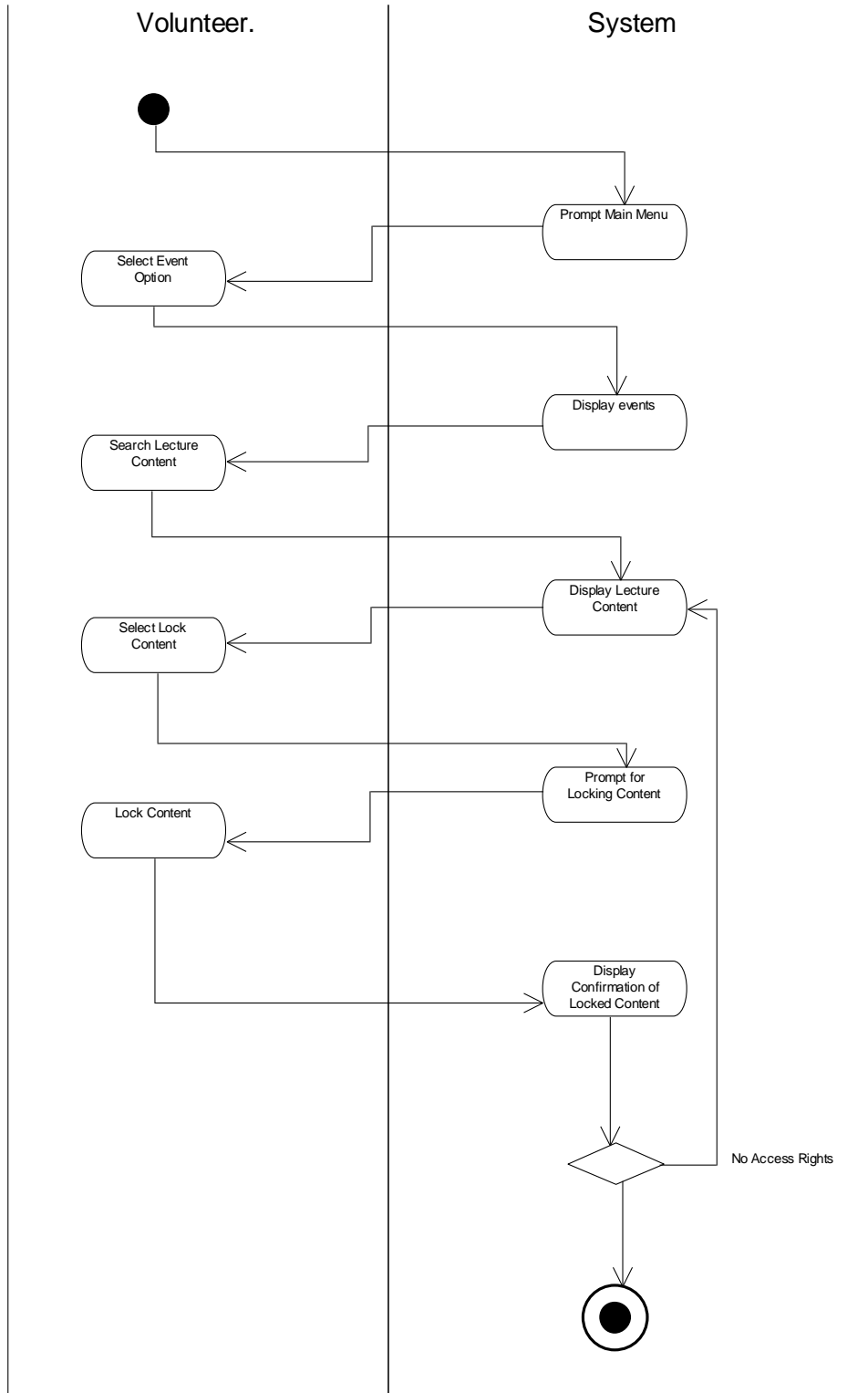


FIGURE 215- ACTIVITY DIAGRAM 7.4 LOCK LECTURE CONTENT

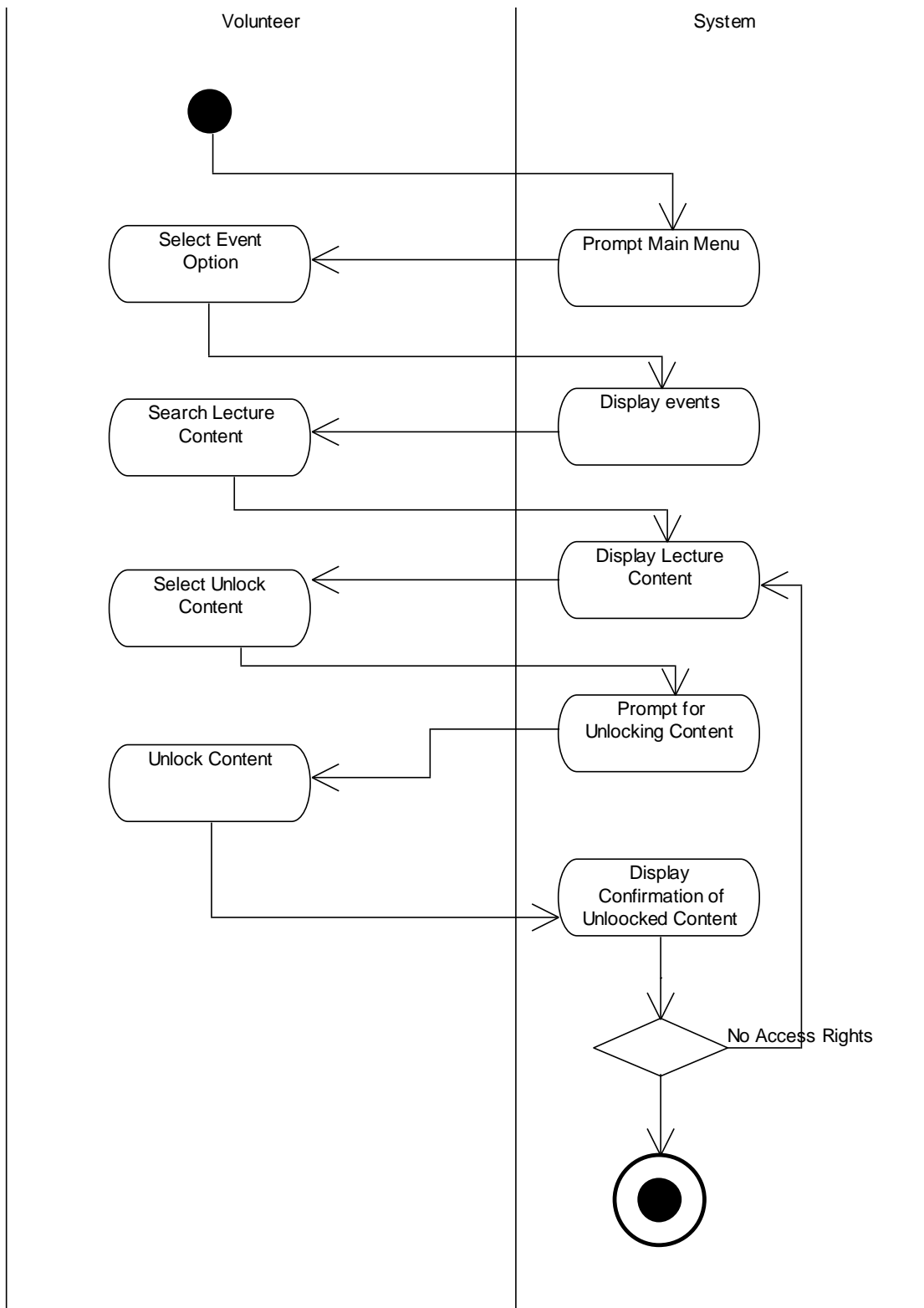


FIGURE 216- ACTIVITY DIAGRAM 7.5 UNLOCK LECTURE CONTENT

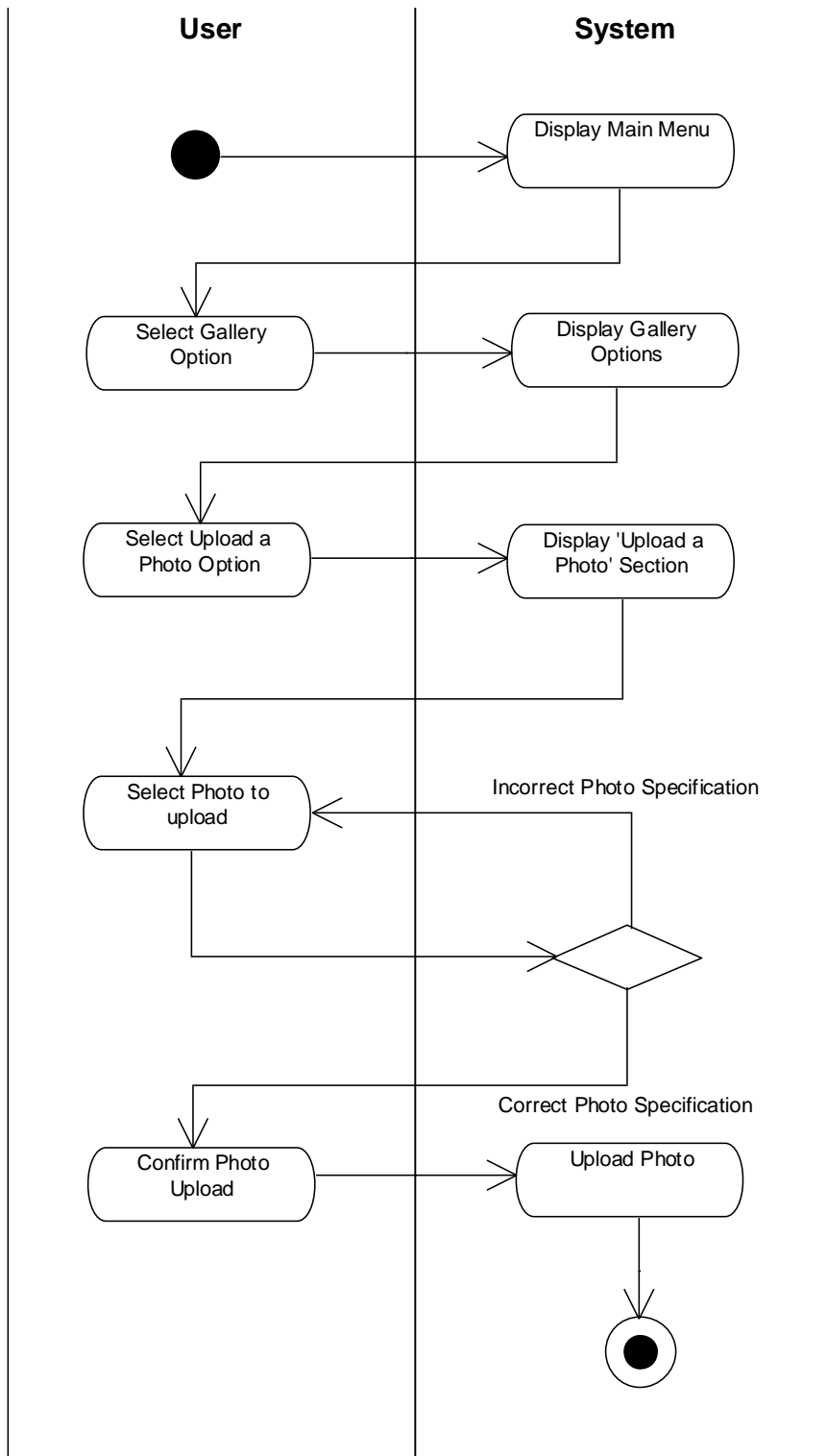


FIGURE 217- ACTIVITY DIAGRAM 8.1 UPLOAD PHOTO

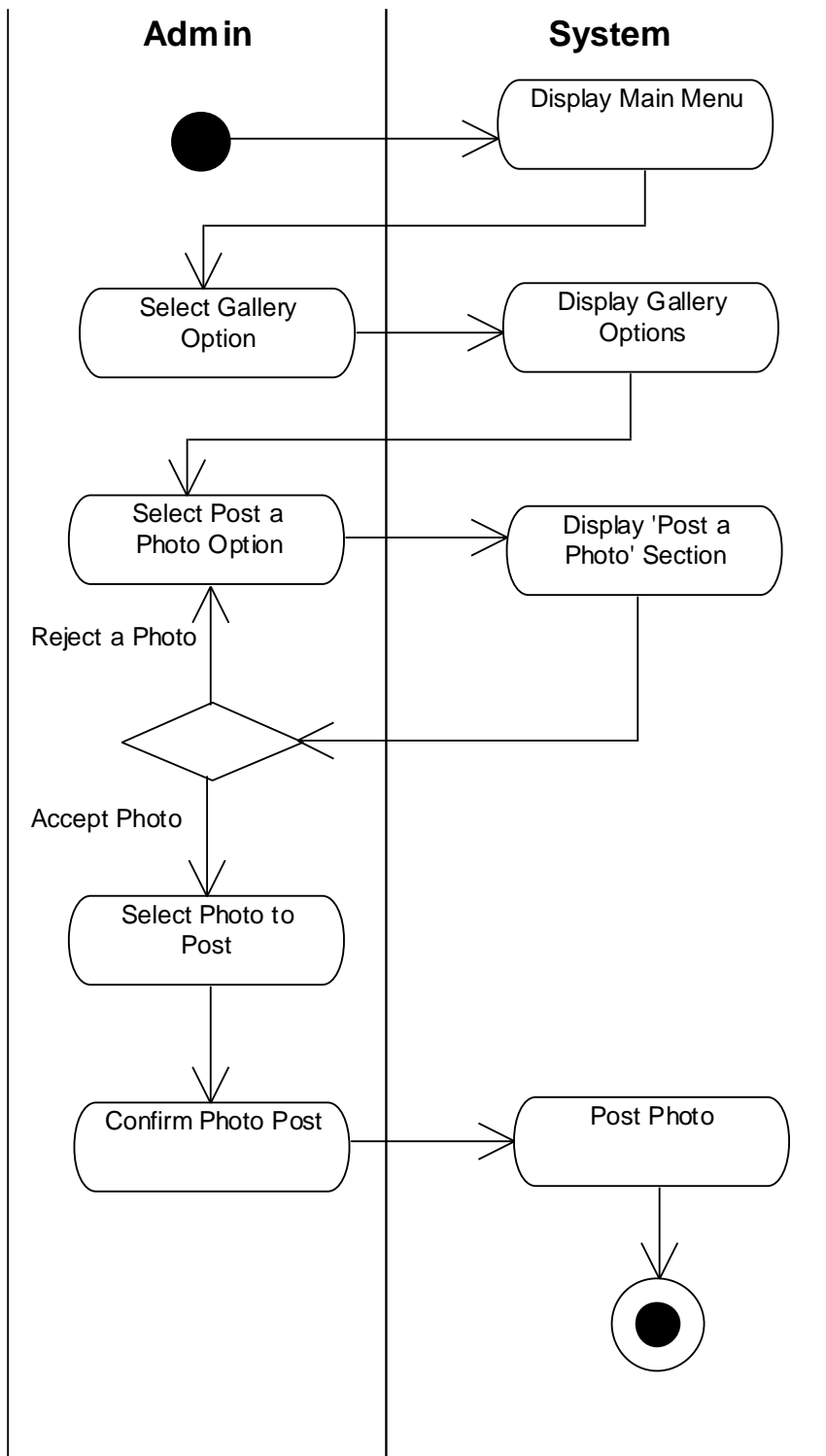


FIGURE 218- ACTIVITY DIAGRAM 8.2 POST PHOTO

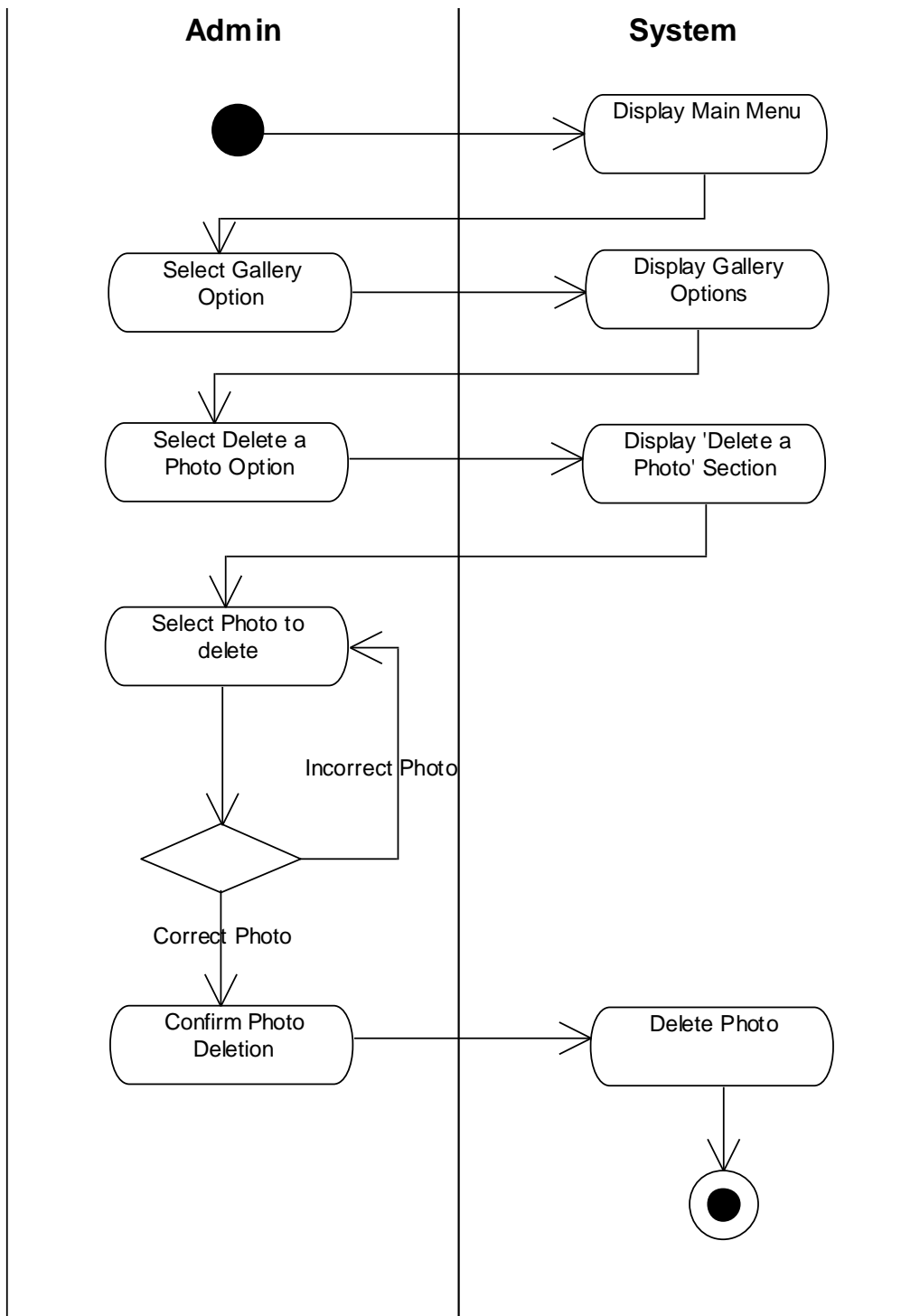


FIGURE 219- ACTIVITY DIAGRAM 8.3 DELETE PHOTO

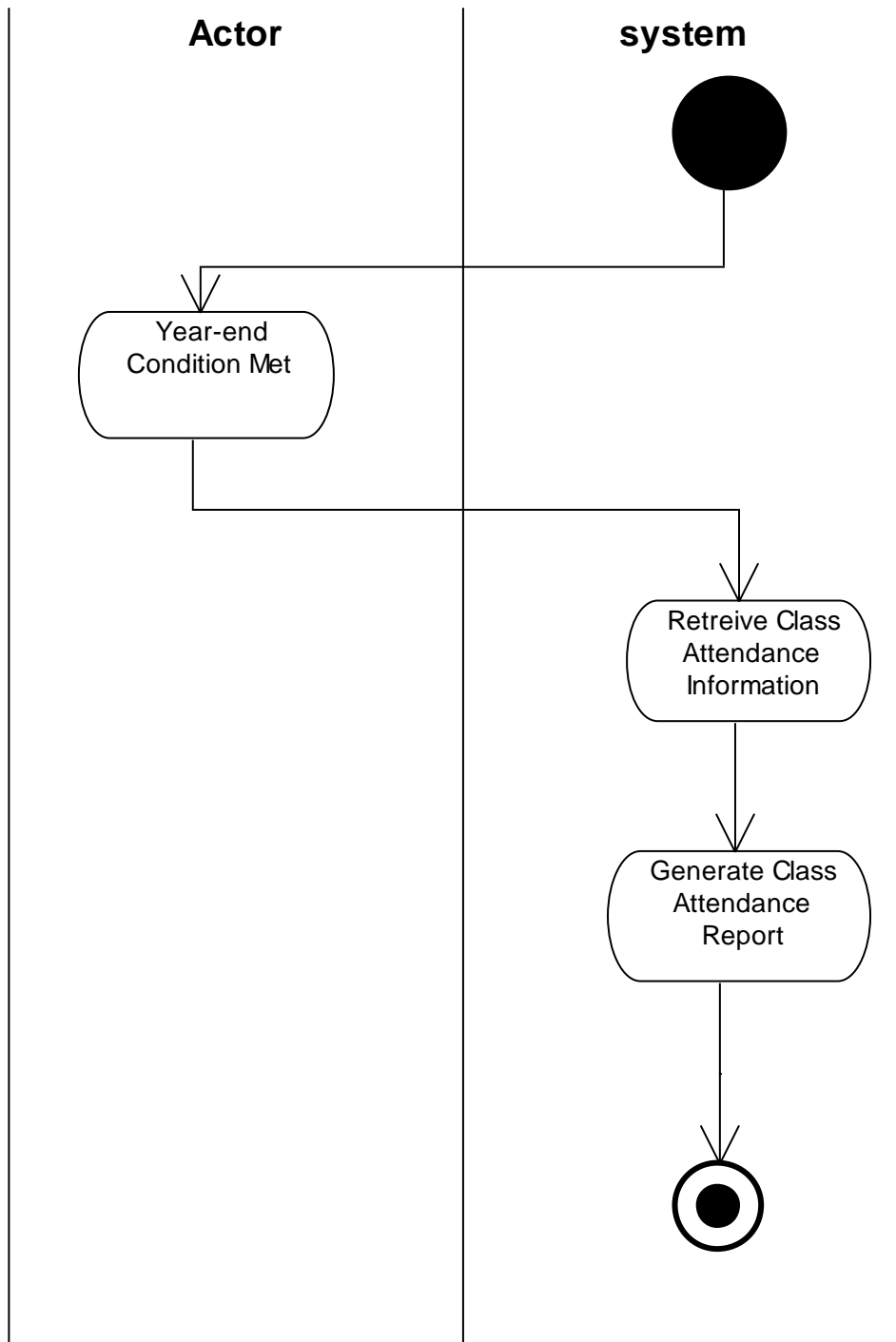


FIGURE 220- ACTIVITY DIAGRAM 9.1 GENERATE CLASS ATTENDANCE REPORT



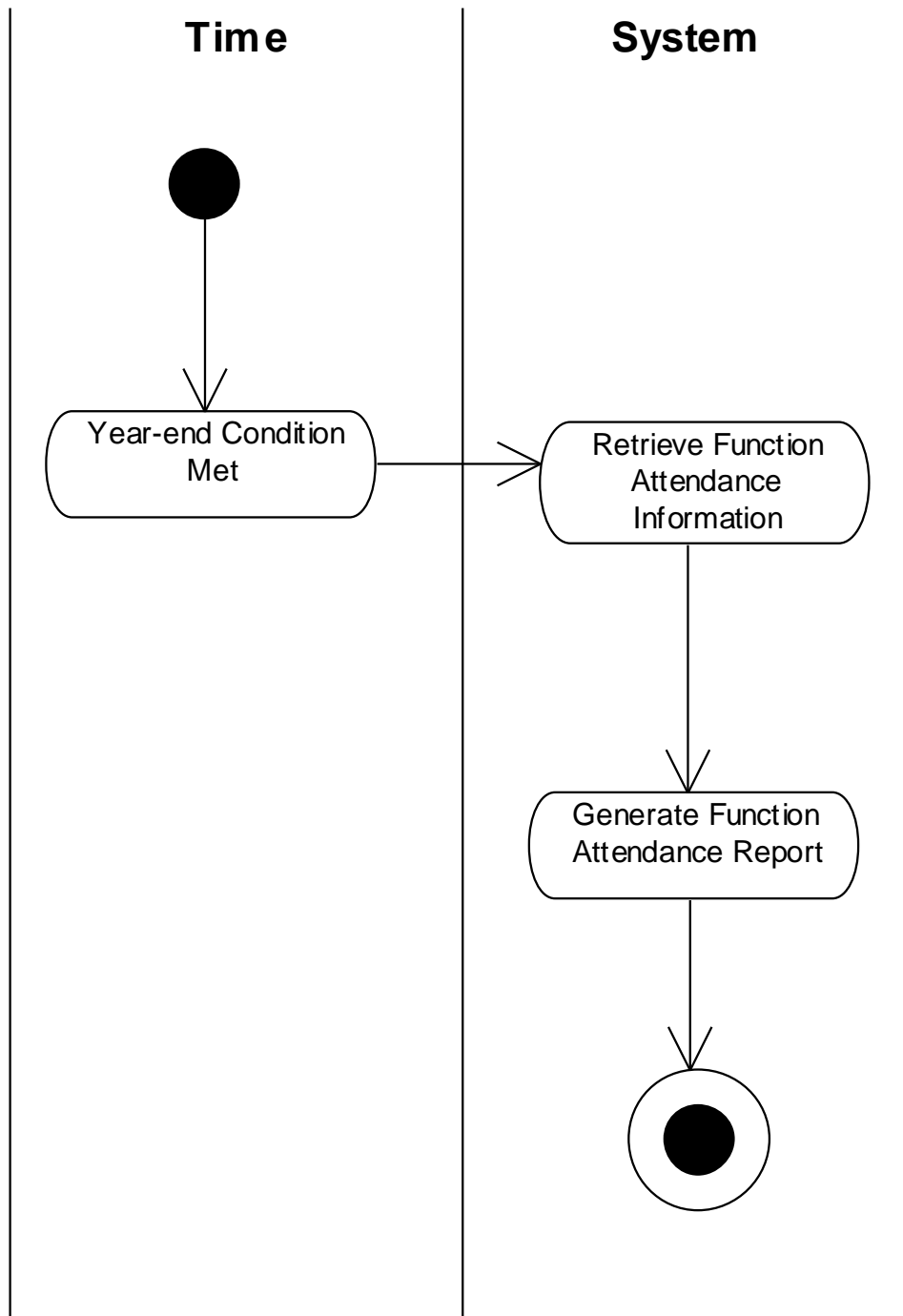


FIGURE 221- ACTIVITY DIAGRAM 9.2 GENERATE FUNCTION ATTENDANCE REPORT

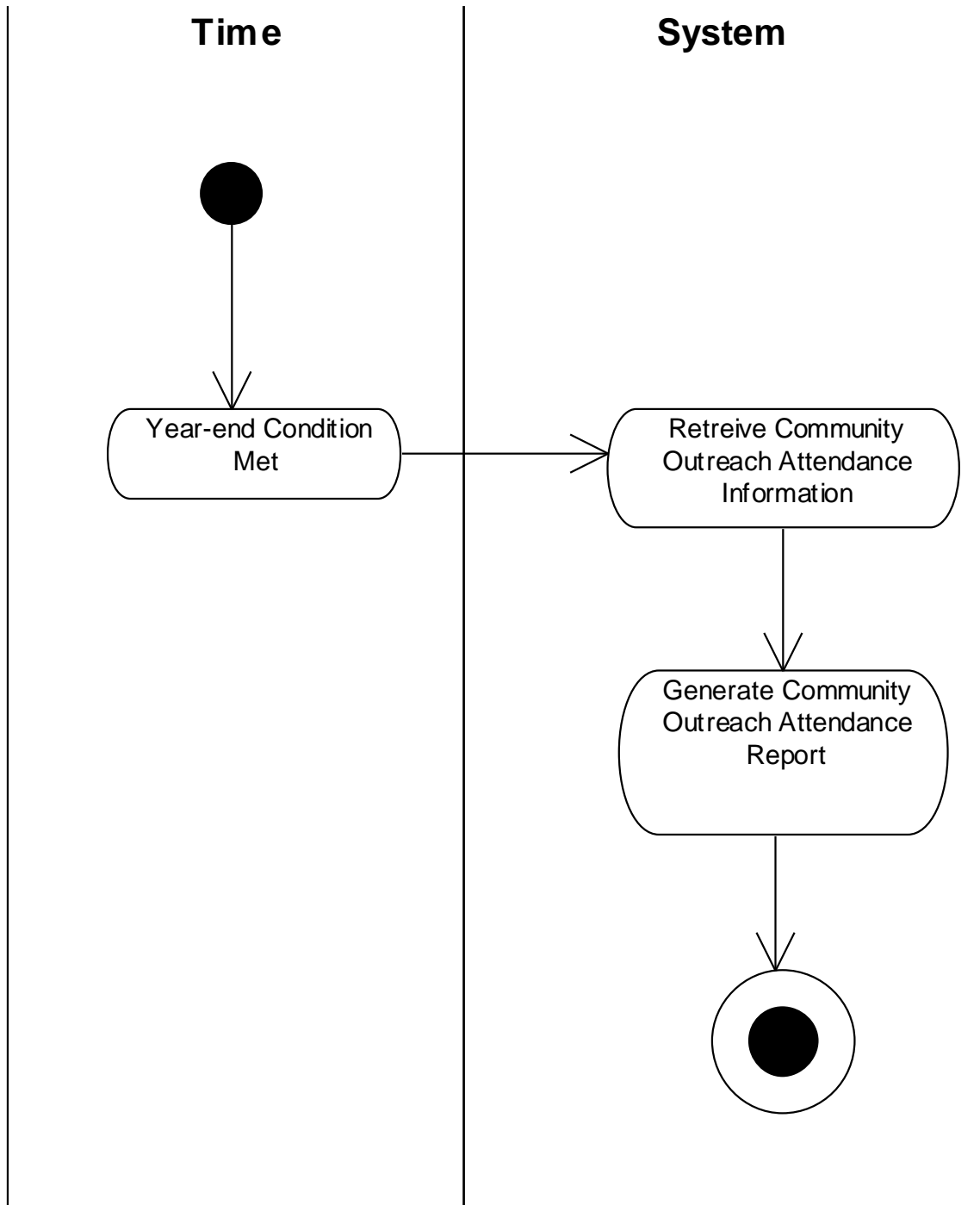


FIGURE 222- ACTIVITY DIAGRAM 9.3 GENERATE COMMUNITY OUTREACH ATTENDANCE REPORT

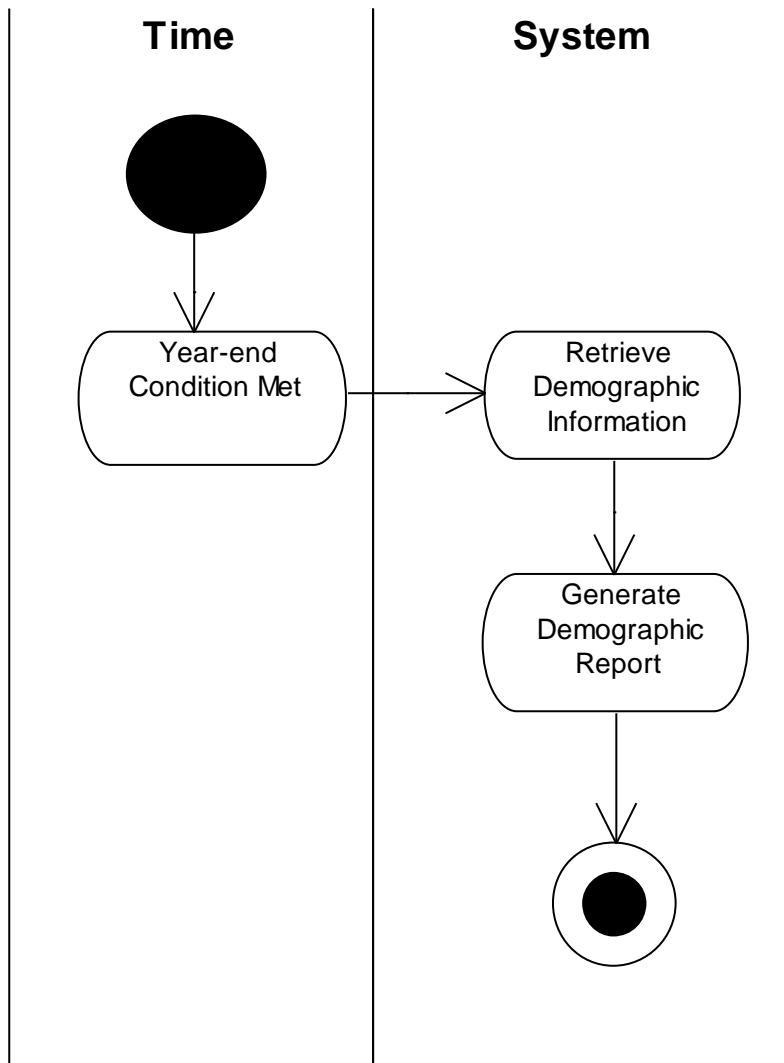


FIGURE 223- ACTIVITY DIAGRAM 9.4 GENERATE DEMOGRAPHIC REPORT

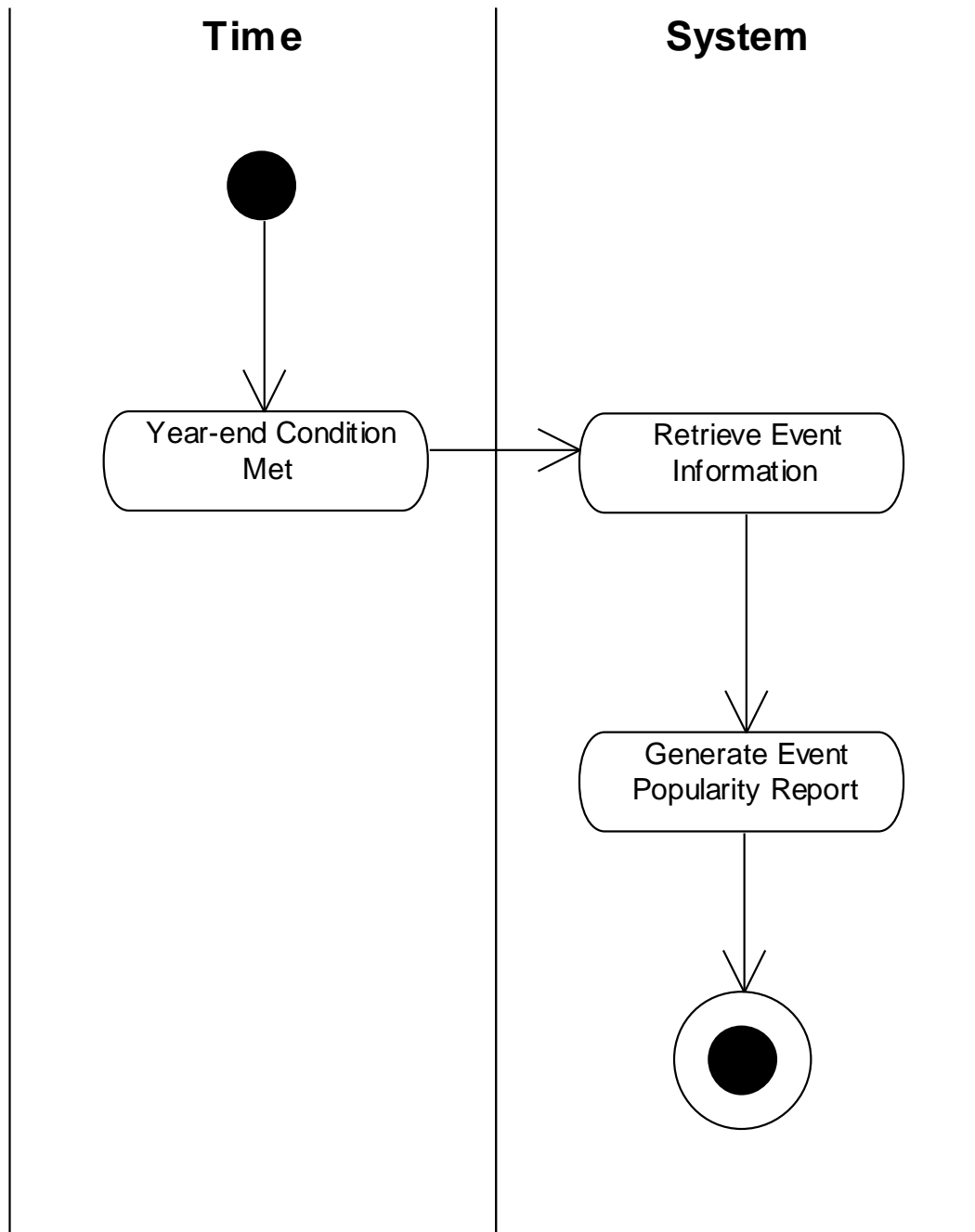


FIGURE 224- ACTIVITY DIAGRAM 9.5 GENERATE EVENT POPULARITY

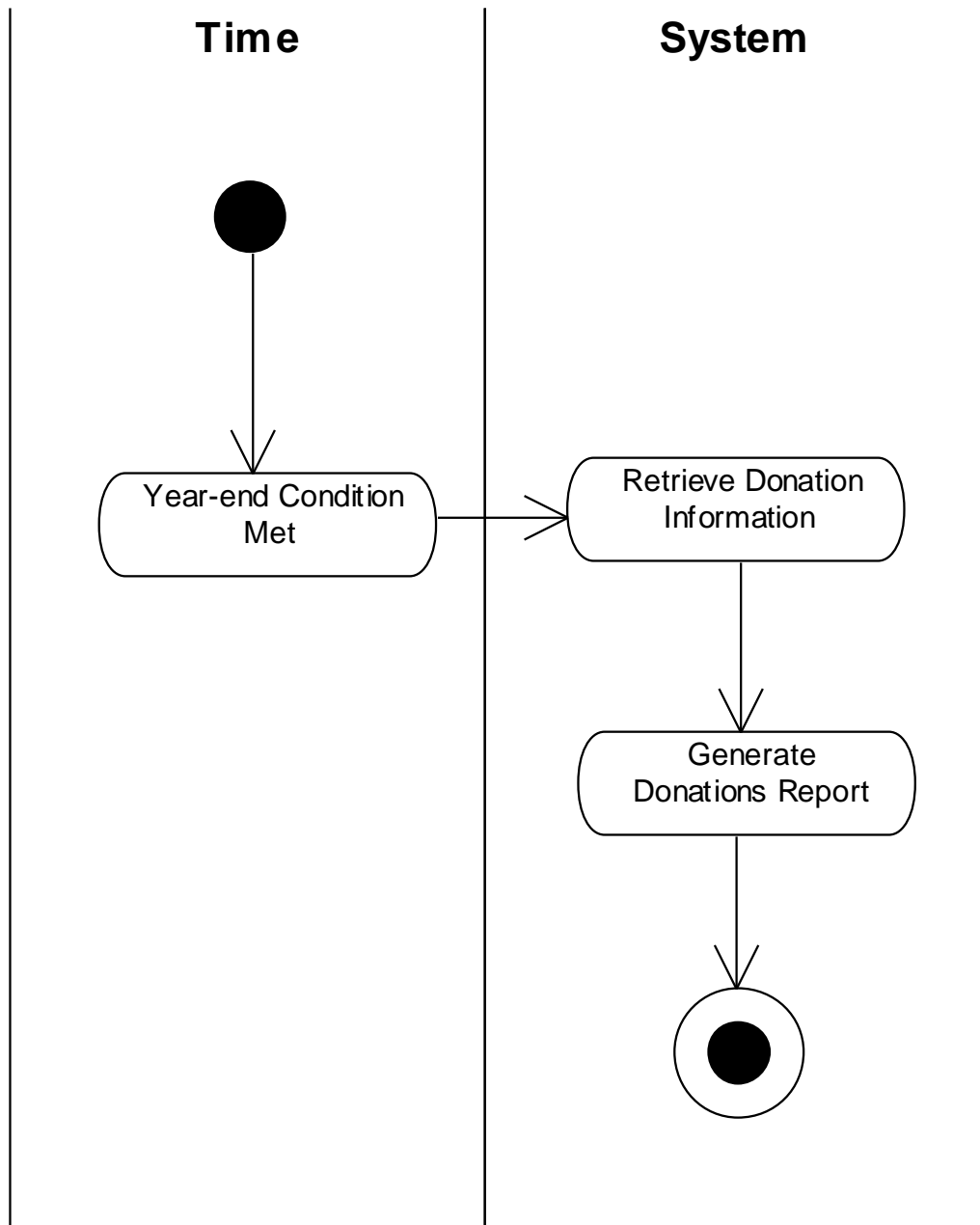


FIGURE 225- ACTIVITY DIAGRAM 9.6 GENERATE DONATIONS REPORT

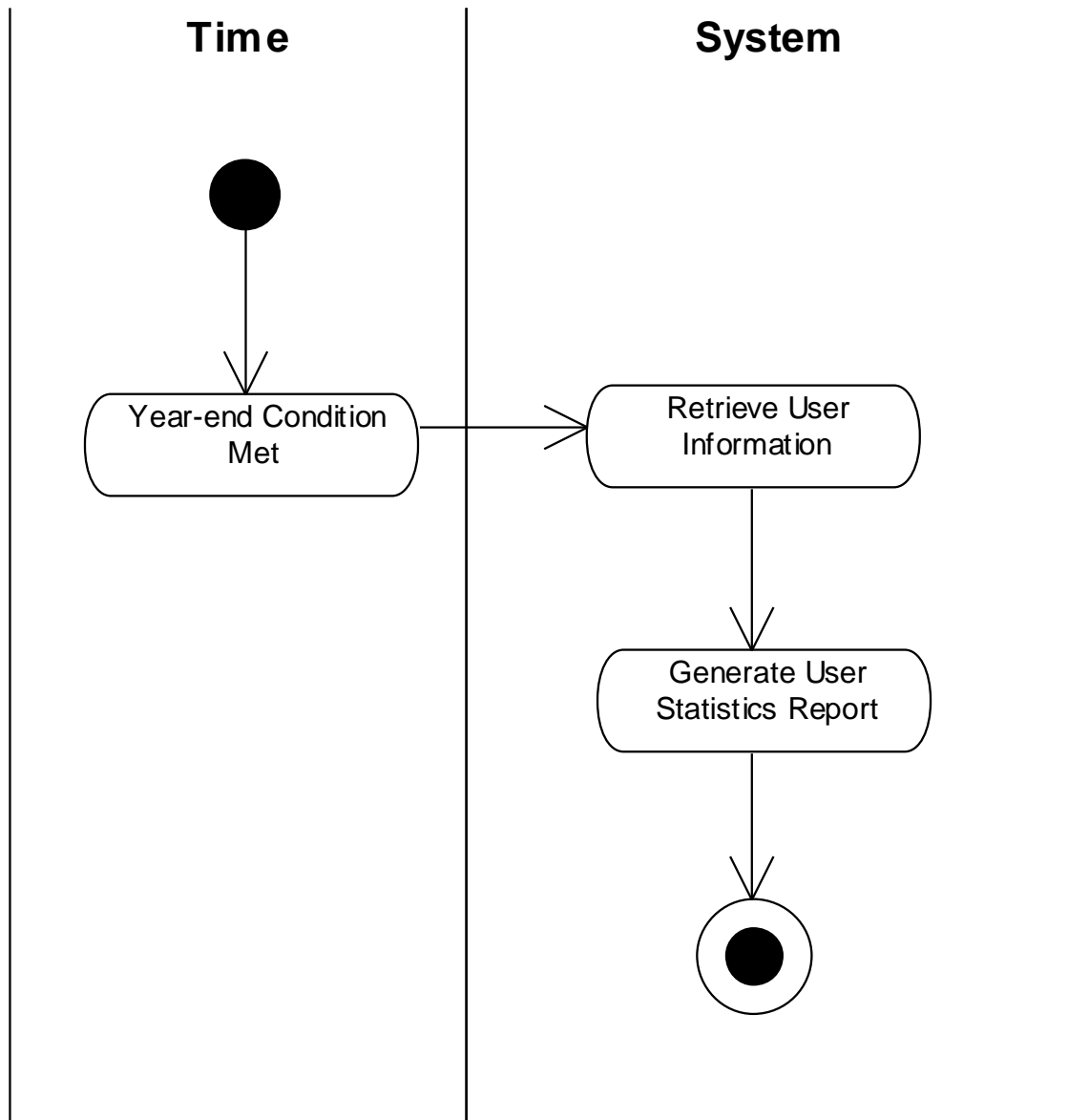


FIGURE 226- ACTIVITY DIAGRAM 9.7 GENERATE USER STATISTICS REPORT

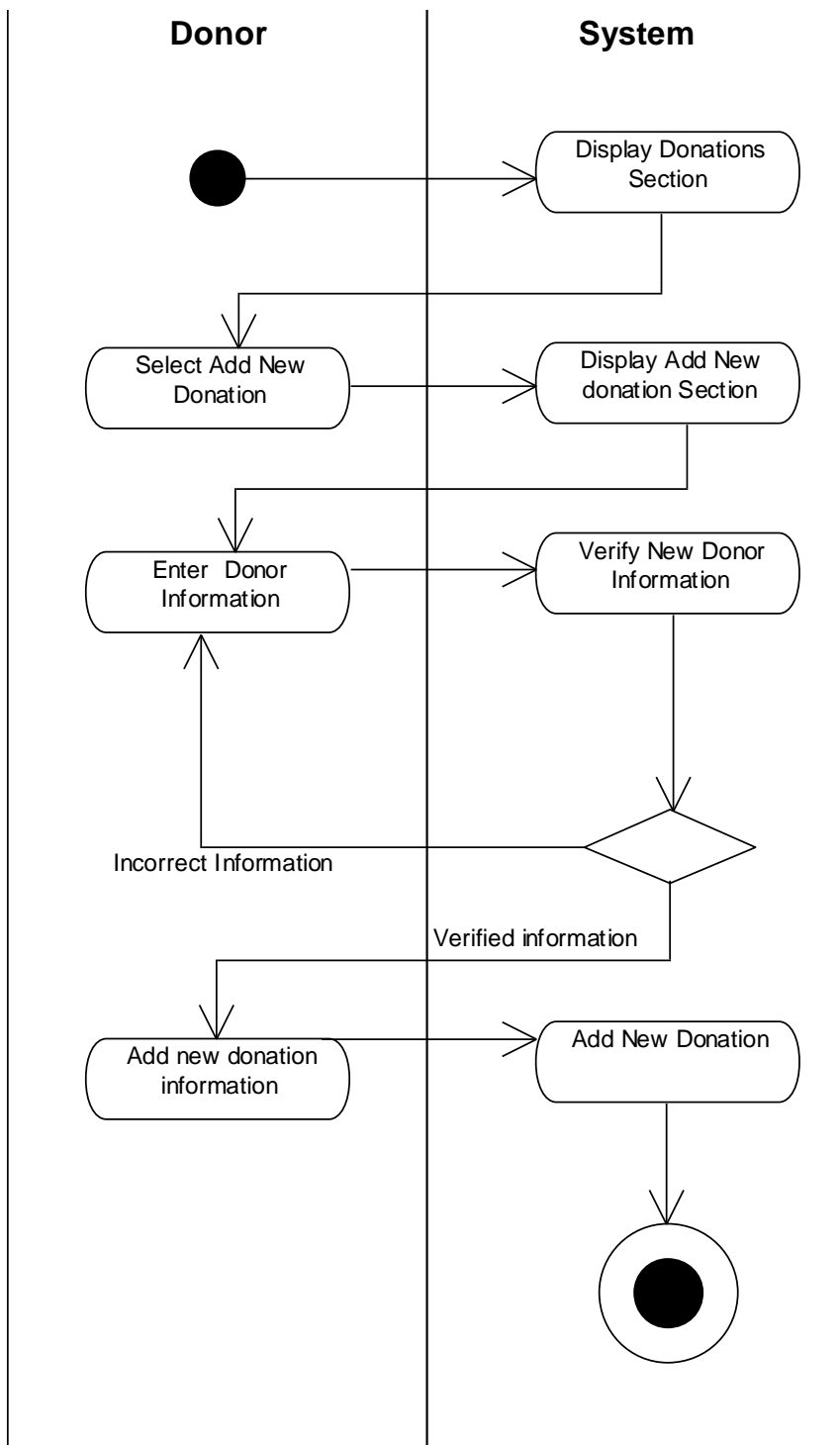


FIGURE 227- ACTIVITY DIAGRAM 10.1 ADD DONOR

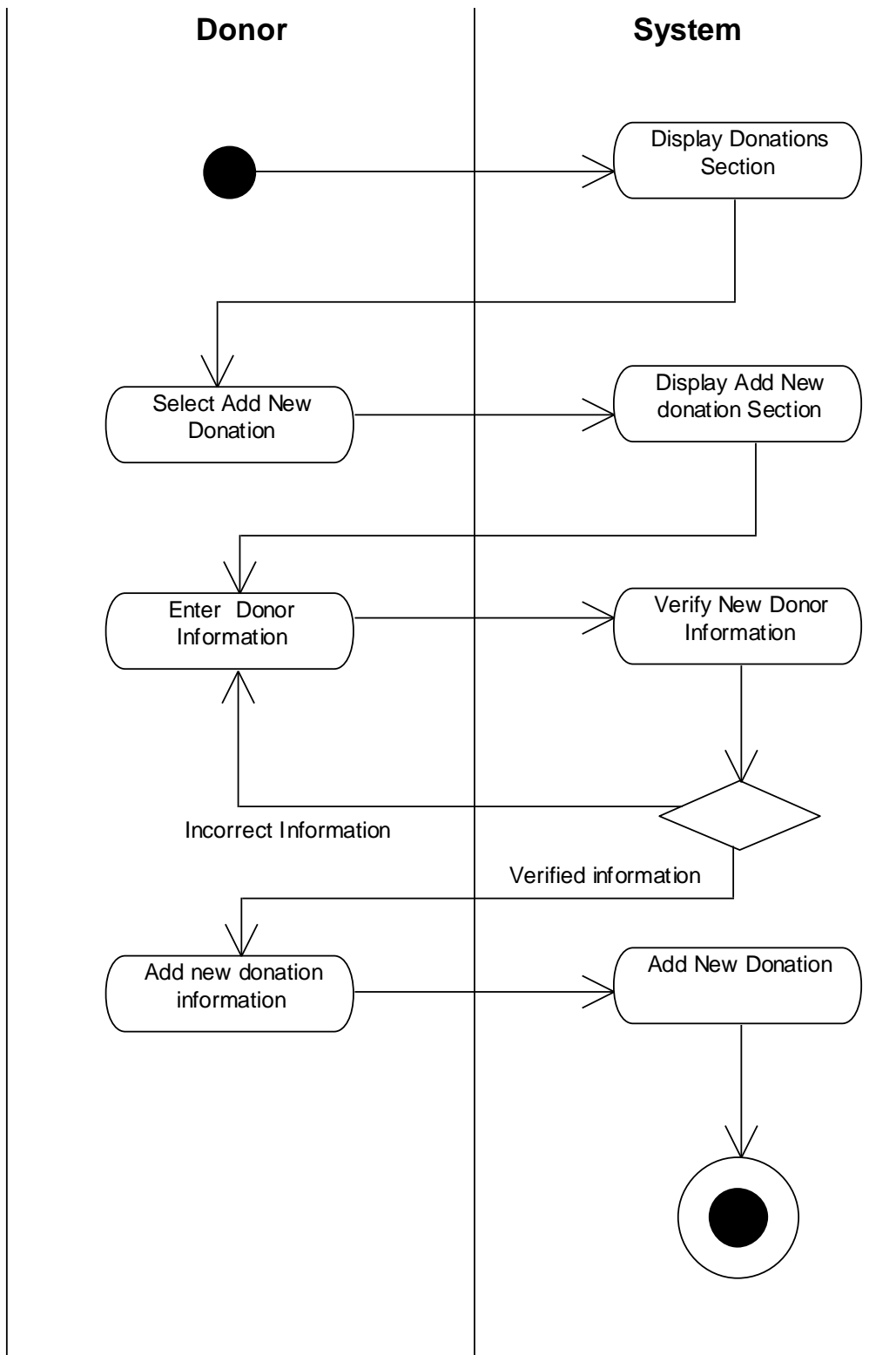


FIGURE 228- ACTIVITY DIAGRAM 10.2 RECEIVE DONATION



## 4.3 Conclusion

4.3.1 The UML modelling of the TRWLA system was stated above. The activity diagrams separated the actions of the actor and the system. It also described when an activity starts and ends within the system. The activity diagrams overall depicted the flow of activities for each use case in the TRWLA system.



## 5. Data Model

### 5.1 Introduction

5.1.1 This section encompasses the database design and data modelling of a functional data model. The group designed a normalized entity relationship diagram to represent the database that would need to be implemented to create the proposed system.

## 5.2 Entity Relationship Diagram

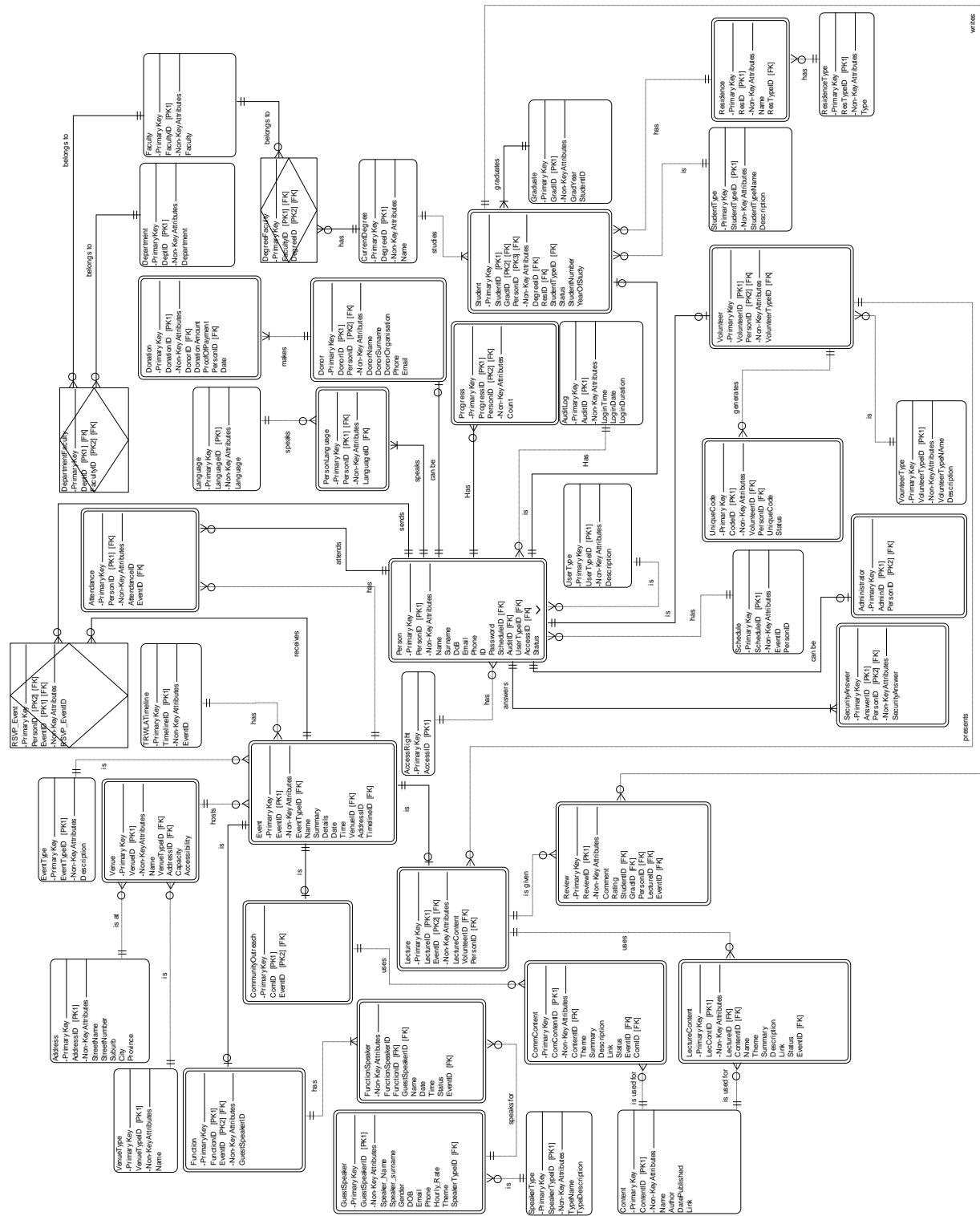


FIGURE 229- ENTITY RELATIONSHIP DIAGRAM

## 5.3 Conclusion

5.3.1 This section encompassed the database design and data modelling of a functional data model. The group designed a normalized entity relationship diagram to represent the database that would need to be implemented to create the proposed system.



---

## 6. Interfaces and Other Inputs

### 6.1 Introduction

6.1.1 The following section gives a complete description of the interface and other inputs for each requirement of the system with a description and purpose of the input, when the input will be used, the entities and attributes involved as well as the logical layout of the input.

## 6.2 Input Interface Description

### 6.2.1 User Input Interface Description

Number	Description	Purpose	When will it be Used	Entities and Attributes	Logical Layout
<b>1.1 Check Forgotten Password</b>	Request to check forgotten password.	To Check for a forgotten password.	When the user forgets their password	<b><u>Person</u></b> <ul style="list-style-type: none"> <li>• PersonID</li> <li>• Email</li> <li>• Password</li> </ul> <b><u>SecurityAnswer</u></b> <ul style="list-style-type: none"> <li>• AnswerID</li> <li>• Answer</li> </ul>	Checked by the user.
<b>1.2 Change Password</b>	Request to change password.	To change the users password.	When the user seeks to change their password to a new one.	<b><u>Person</u></b> <ul style="list-style-type: none"> <li>• PersonID</li> <li>• Email</li> <li>• Password</li> </ul>	Changed by the user.
<b>1.3 Login</b>	Request to Login.	To allow the user to log into the system.	When the user wants to use their password to log into the system.	<b><u>Person</u></b> <ul style="list-style-type: none"> <li>• PersonID</li> <li>• Email</li> <li>• Password</li> </ul> <b><u>UserType</u></b> <ul style="list-style-type: none"> <li>• UserTypeID</li> </ul> <b><u>AuditLog</u></b> <ul style="list-style-type: none"> <li>• AuditID</li> <li>• LogInTime</li> <li>• LogInDate</li> <li>• LogInDuration</li> </ul>	Logged in by the user.
<b>AUC 1 Register User</b>	User Details	To register a new user onto the system	When a student or volunteer wants to register on the system	<b><u>Person</u></b> <ul style="list-style-type: none"> <li>• PersonID</li> <li>• Email</li> <li>• Password</li> </ul> <b><u>SecurityAnswer</u></b> <ul style="list-style-type: none"> <li>• AnswerID</li> <li>• Answer</li> </ul> <b><u>UserType</u></b> <ul style="list-style-type: none"> <li>• UserTypeID</li> </ul>	



## 6.2.2 Volunteer Input Interface Description

Use Case	Description	Purpose	When is it Used	Entities and Attributes	Logical Layout
<b>2.1 Register Volunteer</b>	New Volunteer Details	To register a new volunteer on the TRWLA system.	When a volunteer has been accepted into TRWLA and wants to register.	<b><u>UniqueCode</u></b> <ul style="list-style-type: none"> <li>• CodeID</li> <li>• UniqueCode</li> <li>• Status</li> </ul> <b><u>Person</u></b> <ul style="list-style-type: none"> <li>• PersonID</li> <li>• Name</li> <li>• Surname</li> <li>• DoB</li> <li>• Phone</li> <li>• Race</li> </ul> <b><u>Volunteer</u></b> <ul style="list-style-type: none"> <li>• VolunteerID</li> <li>• VolunteerTypeID</li> </ul> <b><u>Language</u></b> <ul style="list-style-type: none"> <li>• LanguageID</li> <li>• Language</li> </ul> <b><u>PersonLanguage</u></b> <ul style="list-style-type: none"> <li>• PersonID</li> <li>• LanguageID</li> </ul>	The volunteer provides their personal information.
<b>2.2 Search Volunteer</b>	Volunteer Information	To provide the available information of the volunteer that a user wants to search.	Ad hoc. Whenever a user wants to search a volunteer.	<b><u>Person</u></b> <ul style="list-style-type: none"> <li>• Name</li> <li>• Surname</li> <li>• DoB</li> <li>• Email</li> <li>• Phone</li> </ul>	The user provides the search criteria.
<b>2.3 Update Volunteer</b>	New Volunteer Details	To update a volunteers old details to current details.	Ad hoc. Whenever a volunteer wants to make changes to their own details.	<b><u>Person</u></b> <ul style="list-style-type: none"> <li>• PersonID</li> <li>• Name</li> <li>• Surname</li> <li>• DoB</li> <li>• Email</li> <li>• Phone</li> <li>• Race</li> </ul> <b><u>Volunteer</u></b> <ul style="list-style-type: none"> <li>• VolunteerID</li> <li>• VolunteerTypeID</li> </ul>	The volunteer provides their own current details.

Use Case	Description	Purpose	When is it Used	Entities and Attributes	Logical Layout
				<u>Language</u> <ul style="list-style-type: none"> <li>LanguageID</li> <li>Language</li> </ul> <u>PersonLanguage</u> <ul style="list-style-type: none"> <li>PersonID</li> <li>LanguageID</li> </ul>	
<b>2.4 Delete Volunteer</b>	Volunteer Details	To delete a volunteer from the system.	Ad hoc. Whenever admin wants to permanently remove the volunteer from the system.	<u>Person</u> <ul style="list-style-type: none"> <li>PersonID</li> <li>Name</li> <li>Surname</li> <li>DoB</li> <li>Email</li> <li>Phone</li> <li>Race</li> </ul> <u>Volunteer</u> <ul style="list-style-type: none"> <li>VolunteerID</li> <li>VolunteerTypeID</li> </ul> <u>Language</u> <ul style="list-style-type: none"> <li>LanguageID</li> <li>Language</li> </ul> <u>PersonLanguage</u> <ul style="list-style-type: none"> <li>PersonID</li> <li>LanguageID</li> </ul>	Admin provides the information of the volunteer to be deleted from the system.
<b>2.5 Add Volunteer Type</b>	New Volunteer Type Details	To add a new volunteer type to the system.	Ad hoc. Whenever admin wants to add a new volunteer type.	<u>VolunteerType</u> <ul style="list-style-type: none"> <li>VolunteerTypeID</li> <li>VolunteerTypeName</li> <li>Description</li> </ul>	Admin provides the details of the new volunteer type to be added.
<b>2.6 Search Volunteer Type</b>	Volunteer Type Information	To provide the search criteria of a volunteer type to be searched on the system.	Ad hoc. Whenever admin wants to search a volunteer type.	<u>VolunteerType</u> <ul style="list-style-type: none"> <li>VolunteerTypeID</li> <li>VolunteerTypeName</li> <li>Description</li> </ul>	Admin provides the search criteria for the volunteer type to be searched.
<b>2.7 Update Volunteer Type</b>	Updated Volunteer Type Details	To provide the current details of	Ad hoc. Whenever admin wants	<u>VolunteerType</u> <ul style="list-style-type: none"> <li>VolunteerTypeID</li> <li>VolunteerTypeName</li> </ul>	Admin provides the updated details of

Use Case	Description	Purpose	When is it Used	Entities and Attributes	Logical Layout
		the volunteer type to be updated, and update old details to current details.	to update a volunteer type.	<ul style="list-style-type: none"> <li>Description</li> <li>Description</li> </ul>	the volunteer type to be updated.
<b>2.8 Delete Volunteer Type</b>	Volunteer Type to Delete Details	To provide the information of a volunteer that admin wants to delete.	Ad hoc. Whenever admin wants to permanently remove a volunteer type	<u><b>VolunteerType</b></u> <ul style="list-style-type: none"> <li>VolunteerTypeID</li> <li>VolunteerTypeName</li> <li>Description</li> </ul>	Admin provides the information of the volunteer type to be deleted.
<b>2.9 Generate Unique Code</b>	Searched Unique Code	To generate a unique code on the system.	Ad hoc. Whenever a new volunteer has been accepted into TRWLA and admin wants to generate a unique code to allow them to register.	<u><b>UniqueCode</b></u> <ul style="list-style-type: none"> <li>CodeID</li> <li>Code</li> <li>Status</li> </ul>	The system generates a unique code and displays it in a notification, as well as in a message to the volunteer.

### 6.2.3 Student Input Interface Description

Use Case	Description	Purpose	When is it Used	Entities and Attributes	Logical Layout
<b>3.1 Register Student</b>	New Student Details	To register a new student on the TRWLA system.	When a student has joined TRWLA and wants to register.	<b><u>Person</u></b> <ul style="list-style-type: none"> <li>• PersonID</li> <li>• Name</li> <li>• Surname</li> <li>• DoB</li> <li>• Phone</li> <li>• Race</li> </ul> <b><u>Student</u></b> <ul style="list-style-type: none"> <li>• StudentID</li> <li>• StudentNumber</li> <li>• StudentTypeID</li> <li>• DegreeID</li> <li>• YearOfStudy</li> </ul> <b><u>StudentType</u></b> <ul style="list-style-type: none"> <li>• StudentTypeID</li> <li>• Name</li> <li>• Description</li> </ul> <b><u>Language</u></b> <ul style="list-style-type: none"> <li>• LanguageID</li> <li>• Language</li> </ul> <b><u>PersonLanguage</u></b> <ul style="list-style-type: none"> <li>• PersonID</li> <li>• LanguageID</li> </ul> <b><u>CurrentDegree</u></b> <ul style="list-style-type: none"> <li>• DegreeID</li> <li>• Name</li> </ul>	The student provides their personal information.
<b>3.3 Search Student</b>	Student Information	To provide the available information of the student that a user wants to search.	Ad hoc. Whenever a user wants to search a student.	<b><u>Person</u></b> <ul style="list-style-type: none"> <li>• PersonID</li> <li>• Name</li> <li>• Surname</li> <li>• DoB</li> <li>• Phone</li> <li>• Race</li> </ul> <b><u>Student</u></b> <ul style="list-style-type: none"> <li>• StudentID</li> <li>• StudentNumber</li> <li>• StudentTypeID</li> <li>• DegreeID</li> <li>• YearOfStudy</li> </ul> <b><u>StudentType</u></b> <ul style="list-style-type: none"> <li>• StudentTypeID</li> </ul>	The user provides the search criteria.

Use Case	Description	Purpose	When is it Used	Entities and Attributes	Logical Layout
				<ul style="list-style-type: none"> <li>Name</li> <li>Description</li> </ul> <u>Language</u> <ul style="list-style-type: none"> <li>LanguageID</li> <li>Language</li> </ul> <u>PersonLanguage</u> <ul style="list-style-type: none"> <li>PersonID</li> <li>LanguageID</li> </ul> <u>CurrentDegree</u> <ul style="list-style-type: none"> <li>DegreeID</li> <li>Name</li> </ul>	
<b>3.3 Update Student</b>	New Student Details	To update a student's old details to current details.	Ad hoc. Whenever a student wants to make changes to their own details.	<u>Person</u> <ul style="list-style-type: none"> <li>PersonID</li> <li>Name</li> <li>Surname</li> <li>DoB</li> <li>Phone</li> <li>Race</li> </ul> <u>Student</u> <ul style="list-style-type: none"> <li>StudentID</li> <li>StudentNumber</li> <li>StudentTypeID</li> <li>DegreeID</li> <li>YearOfStudy</li> </ul> <u>StudentType</u> <ul style="list-style-type: none"> <li>StudentTypeID</li> <li>Name</li> <li>Description</li> </ul> <u>Language</u> <ul style="list-style-type: none"> <li>LanguageID</li> <li>Language</li> </ul> <u>PersonLanguage</u> <ul style="list-style-type: none"> <li>PersonID</li> <li>LanguageID</li> </ul> <u>CurrentDegree</u> <ul style="list-style-type: none"> <li>DegreeID</li> <li>Name</li> </ul>	The student provides their own current details.
<b>3.4 Delete Student</b>	Student Details	To delete a student from the system.	Ad hoc. Whenever admin wants to permanently remove the	<u>Person</u> <ul style="list-style-type: none"> <li>PersonID</li> <li>Name</li> <li>Surname</li> <li>DoB</li> <li>Phone</li> </ul>	Admin provides the information of the student to be deleted

Use Case	Description	Purpose	When is it Used	Entities and Attributes	Logical Layout
			student from the system.	<ul style="list-style-type: none"> <li>Race</li> </ul> <b><u>Student</u></b> <ul style="list-style-type: none"> <li>StudentID</li> <li>StudentNumber</li> <li>StudentTypeID</li> <li>DegreeID</li> <li>YearOfStudy</li> </ul> <b><u>StudentType</u></b> <ul style="list-style-type: none"> <li>StudentTypeID</li> <li>Name</li> <li>Description</li> </ul> <b><u>Language</u></b> <ul style="list-style-type: none"> <li>LanguageID</li> <li>Language</li> </ul> <b><u>PersonLanguage</u></b> <ul style="list-style-type: none"> <li>PersonID</li> <li>LanguageID</li> </ul> <b><u>CurrentDegree</u></b> <ul style="list-style-type: none"> <li>DegreeID</li> <li>Name</li> </ul>	from the system.
<b>3.5 Generate Graduate List</b>	Searched Student Information	To generate a list of students who are eligible to graduate from TRWLA.	At the end of the TRWLA year.	<b><u>Student:</u></b> <ul style="list-style-type: none"> <li>StudentID</li> </ul> <b><u>Graduate:</u></b> <ul style="list-style-type: none"> <li>GraduationID</li> <li>StudentID</li> <li>GradYear</li> </ul> <b><u>Progress</u></b> <ul style="list-style-type: none"> <li>ProgressID</li> </ul>	A list of student Information who qualify to graduate.
<b>3.6 Add Student Type</b>	New Student Type Details	To add a new student type to the system.	Ad hoc. Whenever admin wants to add a new student type.	<b><u>StudentType</u></b> <ul style="list-style-type: none"> <li>StudentTypeID</li> <li>StudentTypeName</li> <li>Description</li> </ul>	Admin provides the details of the new student type to be added.
<b>3.7 Search Student Type</b>	Student Type Information	To provide the search criteria of a student type to be searched on the system.	Ad hoc. Whenever admin wants to search a student type.	<b><u>StudentType</u></b> <ul style="list-style-type: none"> <li>StudentTypeID</li> <li>StudentTypeName</li> <li>Description</li> </ul>	Admin provides the search criteria for the student type to be searched.

Use Case	Description	Purpose	When is it Used	Entities and Attributes	Logical Layout
<b>3.8 Update Student Type</b>	Updated Student Type Details	To provide the current details of the student type to be updated, and update old details to current details.	Ad hoc. Whenever admin wants to update a student type.	<u><b>StudentType</b></u> <ul style="list-style-type: none"> <li>StudentTypeID</li> <li>StudentTypeName</li> <li>Description</li> </ul>	Admin provides the updated details of the student type to be updated.
<b>3.9 Delete Student Type</b>	Student Type to Delete Details	To provide the information of a student that admin wants to delete.	Ad hoc. Whenever admin wants to permanently remove a student type	<u><b>StudentType</b></u> <ul style="list-style-type: none"> <li>StudentTypeID</li> <li>StudentTypeName</li> <li>Description</li> </ul>	Admin provides the information of the student type to be deleted.

## 6.2.4 Events Input Interface Description

Use Case	Description	Purpose	When will it be Used	Entities and Attributes	Logical Layout
<b>4.1 Create Venue</b>	New Venue Information	To add a new venue	Volunteer wants to add a new venue for an event.	<b>Venue:</b> <ul style="list-style-type: none"> <li>• VenueID</li> <li>• Name</li> <li>• Capacity</li> <li>• Accessibility</li> </ul> <b>VenueType:</b> <ul style="list-style-type: none"> <li>• VenueTypeID</li> <li>• Name</li> </ul> <b>Address:</b> <ul style="list-style-type: none"> <li>• AddressID</li> <li>• StreetName</li> <li>• StreetNumber</li> <li>• Suburb</li> <li>• City</li> <li>• Province</li> </ul>	Added by Volunteer who provided the correct information.
<b>4.2 Search Venue</b>	Updated Venue Information	To view a venue's information.	User wants to search for a venue or they are creating an event.	<b>Venue:</b> <ul style="list-style-type: none"> <li>• VenueID</li> <li>• Name</li> <li>• Capacity</li> <li>• Accessibility</li> </ul> <b>VenueType:</b> <ul style="list-style-type: none"> <li>• VenueTypeID</li> <li>• Name</li> </ul> <b>Address:</b> <ul style="list-style-type: none"> <li>• AddressID</li> <li>• StreetName</li> <li>• StreetNumber</li> <li>• Suburb</li> <li>• City</li> <li>• Province</li> </ul>	Selected by User.
<b>4.3 Update Venue</b>	Updated Venue Information	To update a venue's details.	A venue's information was entered incorrectly or the details of the venue have changed.	<b>Venue:</b> <ul style="list-style-type: none"> <li>• VenueID</li> <li>• Name</li> <li>• Capacity</li> <li>• Accessibility</li> </ul> <b>VenueType:</b> <ul style="list-style-type: none"> <li>• VenueTypeID</li> <li>• Name</li> </ul> <b>Address:</b> <ul style="list-style-type: none"> <li>• AddressID</li> <li>• StreetName</li> </ul>	Updated by Volunteer who provided the appropriate information.



Use Case	Description	Purpose	When will it be Used	Entities and Attributes	Logical Layout
				<ul style="list-style-type: none"> <li>StreetNumber</li> <li>Suburb</li> <li>City</li> <li>Province</li> </ul>	
<b>4.4 Delete Venue</b>	Venue Deletion Information	To delete a venue.	The organisation no longer wishes to use the venue.	<b>Venue:</b> <ul style="list-style-type: none"> <li>VenueID</li> <li>Name</li> <li>Capacity</li> <li>Accessibility</li> </ul> <b>VenueType:</b> <ul style="list-style-type: none"> <li>VenueTypeID</li> <li>Name</li> </ul> <b>Address:</b> <ul style="list-style-type: none"> <li>AddressID</li> <li>StreetName</li> <li>StreetNumber</li> <li>Suburb</li> <li>City</li> <li>Province</li> </ul>	Deleted by Volunteer who provided the appropriate information.
<b>4.5 Create Event</b>	New Event Information	To create an event that will take place in the near future on the TRWLA Timeline.	When an event is due to be created that is upcoming in the TRWLA timeline.	<b>Events:</b> <ul style="list-style-type: none"> <li>EventID</li> <li>EventTypeID</li> <li>Name</li> <li>Summary</li> <li>Details</li> <li>Time</li> <li>Date</li> <li>VenueID</li> </ul> <b>Function:</b> <ul style="list-style-type: none"> <li>FunctionID</li> <li>GuestSpeakerID</li> </ul> <b>Lecture:</b> <ul style="list-style-type: none"> <li>LectureID</li> </ul> <b>CommunityOutreach:</b> <ul style="list-style-type: none"> <li>ComID</li> </ul> <b>Residence:</b> <ul style="list-style-type: none"> <li>ResidenceID</li> <li>Name</li> </ul> <b>Content:</b> <ul style="list-style-type: none"> <li>ContentID</li> <li>Name</li> <li>Author</li> <li>DatePublished</li> <li>Link</li> </ul>	Created by Volunteer who provided the appropriate information.

Use Case	Description	Purpose	When will it be Used	Entities and Attributes	Logical Layout
<b>4.6 Search Event</b>	Event Search Parameters	To view an event's information.	User wants to search for an event or they are creating an event.	<b>Events:</b> <ul style="list-style-type: none"> <li>EventID</li> <li>EventTypeID</li> <li>Name</li> <li>Summary</li> <li>Details</li> <li>Time</li> <li>Date</li> <li>VenueID</li> </ul> <b>Function:</b> <ul style="list-style-type: none"> <li>FunctionID</li> <li>GuestSpeakerID</li> </ul> <b>Lecture:</b> <ul style="list-style-type: none"> <li>LectureID</li> </ul> <b>CommunityOutreach:</b> <ul style="list-style-type: none"> <li>ComID</li> </ul> <b>Residence:</b> <ul style="list-style-type: none"> <li>ResidenceID</li> <li>Name</li> </ul> <b>Content:</b> <ul style="list-style-type: none"> <li>ContentID</li> <li>Name</li> <li>Author</li> <li>DatePublished</li> <li>Link</li> </ul> <b>TRWLATimeline</b> <ul style="list-style-type: none"> <li>TimelineID</li> </ul>	Searched by the User.
<b>4.7 RSVP To an Event</b>	Event Information	To RSVP to an event.	When a user wishes to RSVP to an event.	<b>Events:</b> <ul style="list-style-type: none"> <li>EventID</li> <li>EventTypeID</li> <li>Name</li> <li>Summary</li> <li>Details</li> <li>Time</li> <li>Date</li> <li>VenueID</li> </ul> <b>RSVP Event:</b> <ul style="list-style-type: none"> <li>RSVP_EventID</li> <li>EventID</li> <li>PersonID</li> </ul> <b>Person:</b> <ul style="list-style-type: none"> <li>PersonID</li> <li>Name</li> <li>Surname</li> </ul>	Done by the user.

Use Case	Description	Purpose	When will it be Used	Entities and Attributes	Logical Layout
				<ul style="list-style-type: none"> <li>Email</li> <li>Phone</li> </ul>	
<b>4.8 View TRWLA Timeline</b>	TRWLA Timeline Information	To view the TRWLA timeline.	When a user wants to view the upcoming events on the TRWLA timeline.	<b>Events:</b> <ul style="list-style-type: none"> <li>EventID</li> <li>EventTypeID</li> <li>Name</li> <li>Summary</li> <li>Details</li> <li>Time</li> <li>Date</li> <li>VenueID</li> </ul> <b>Function:</b> <ul style="list-style-type: none"> <li>FunctionID</li> <li>GuestSpeakerID</li> </ul> <b>Lecture:</b> <ul style="list-style-type: none"> <li>LectureID</li> <li>LectureContent</li> </ul> <b>CommunityOutreach:</b> <ul style="list-style-type: none"> <li>ComID</li> <li>EventID</li> </ul> <b>Residence:</b> <ul style="list-style-type: none"> <li>ResidenceID</li> <li>ResidenceName</li> </ul>	Done by a user.
<b>4.9 Update Event Information</b>	Updated Event Information	To update an event on the system.	Information is incorrectly provided or details about the event have changed.	<b>Events:</b> <ul style="list-style-type: none"> <li>EventID</li> <li>EventTypeID</li> <li>Name</li> <li>Summary</li> <li>Details</li> <li>Time</li> <li>Date</li> <li>VenueID</li> </ul> <b>Function:</b> <ul style="list-style-type: none"> <li>FunctionID</li> <li>GuestSpeakerID</li> </ul> <b>Lecture:</b> <ul style="list-style-type: none"> <li>LectureID</li> </ul> <b>CommunityOutreach:</b> <ul style="list-style-type: none"> <li>ComID</li> </ul> <b>Residence:</b> <ul style="list-style-type: none"> <li>ResidenceID</li> <li>Name</li> </ul> <b>Content:</b> <ul style="list-style-type: none"> <li>ContentID</li> </ul>	Updated by Volunteer who provided the appropriate information.

Use Case	Description	Purpose	When will it be Used	Entities and Attributes	Logical Layout
				<ul style="list-style-type: none"> <li>Name</li> <li>Author</li> <li>DatePublished</li> <li>Link</li> </ul>	
<b>4.10 Cancel Event</b>	Event Cancellation Information	To cancel an event off of the system.	The event is no longer going to take place and thus it will be deleted off of the system.	<b>Events:</b> <ul style="list-style-type: none"> <li>EventID</li> <li>EventTypeID</li> <li>Name</li> <li>Summary</li> <li>Details</li> <li>Time</li> <li>Date</li> <li>VenueID</li> </ul> <b>Function:</b> <ul style="list-style-type: none"> <li>FunctionID</li> <li>GuestSpeakerID</li> </ul> <b>Lecture:</b> <ul style="list-style-type: none"> <li>LectureID</li> </ul> <b>CommunityOutreach:</b> <ul style="list-style-type: none"> <li>ComID</li> </ul> <b>Residence:</b> <ul style="list-style-type: none"> <li>ResidenceID</li> <li>Name</li> </ul> <b>Content:</b> <ul style="list-style-type: none"> <li>ContentID</li> <li>Name</li> <li>Author</li> <li>DatePublished</li> <li>Link</li> </ul>	Cancelled by Volunteer.
<b>4.11 Log Event Attendance</b>	Student Details	To log the attendance of a student at an event.	When a student is attending an event towards their certification.	<b>Person:</b> <ul style="list-style-type: none"> <li>PersonID</li> <li>Name</li> <li>Surname</li> </ul> <b>Student:</b> <ul style="list-style-type: none"> <li>StudentID</li> <li>StudentNumber</li> </ul> <b>RSVP Event:</b> <ul style="list-style-type: none"> <li>RSVP_EventID</li> <li>EventID</li> <li>PersonID</li> </ul> <b>Attendance:</b> <ul style="list-style-type: none"> <li>Attendance_ID</li> <li>EventID</li> <li>PersonID</li> </ul>	Requested by Student, completed by Volunteer.

Use Case	Description	Purpose	When will it be Used	Entities and Attributes	Logical Layout
				<b><u>Progress:</u></b> <ul style="list-style-type: none"> <li>ProgressID</li> <li>ProgressCount</li> </ul>	
<b>4.12 Send Notification</b>	Notification Details	To send a notification to a certain selected group of users.	When information needs to be sent to users.	<b><u>Person:</u></b> <ul style="list-style-type: none"> <li>PersonID</li> <li>Name</li> <li>Surname</li> <li>Email</li> <li>Phone</li> </ul>	Sent by the volunteer who provided the recipients and message.

## 6.2.5 Function Input Interface Description

Use Case	Description	Purpose	When will it be Used	Entities and Attributes	Logical Layout
<b>5.1 Register Guest Speaker</b>	Register Guest Speaker Details	To register a new guest speaker on the system.	Ad hoc basis	<b><u>GuestSpeaker</u></b> <ul style="list-style-type: none"> <li>• GuestSpeakerID</li> <li>• Speaker_Name</li> <li>• Speaker_Surname</li> <li>• Gender</li> <li>• DOB</li> <li>• Email</li> <li>• Phone</li> <li>• Hourly_Rate</li> <li>• Theme</li> </ul> <b><u>SpeakerType</u></b> <ul style="list-style-type: none"> <li>• SpeakerTypeID</li> <li>• TypeName</li> <li>• TypeDescription</li> </ul>	Volunteer provides the following details:  Name  Surname  Gender  Date of Birth  Email Address  Phone Number  Hourly Rate  Theme
<b>5.2 Search Guest Speaker</b>	Search Guest Speaker Details	To view a guest speaker's information.	Ad hoc basis	<b><u>GuestSpeaker</u></b> <ul style="list-style-type: none"> <li>• GuestSpeakerID</li> <li>• Speaker_Name</li> <li>• Speaker_Surname</li> <li>• Gender</li> <li>• DOB</li> <li>• Email</li> <li>• Phone</li> <li>• Hourly_Rate</li> <li>• Theme</li> </ul>	System displays information such as:  Name  Surname  Gender  Date of Birth  Email Address  Phone Number  Hourly Rate  Theme

Use Case	Description	Purpose	When will it be Used	Entities and Attributes	Logical Layout
<b>5.3 Update Guest Speaker</b>	Update Guest Speaker Details	To update a guest speaker's details.	Ad hoc basis.	<b><u>GuestSpeaker</u></b> <ul style="list-style-type: none"> <li>GuestSpeakerID</li> <li>Speaker_Name</li> <li>Speaker_Surname</li> <li>Gender</li> <li>DOB</li> <li>Email</li> <li>Phone</li> <li>Hourly_Rate</li> <li>Theme</li> </ul> <b><u>SpeakerType</u></b> <ul style="list-style-type: none"> <li>SpeakerTypeID</li> <li>TypeName</li> <li>TypeDescription</li> </ul>	Volunteer provides the following details:  Name  Surname  Gender  Date of Birth  Email Address  Phone Number  Hourly Rate  Theme
<b>5.4 Delete Guest Speaker</b>	Delete Speaker Details	To delete a guest speaker.	Ad hoc basis	<b><u>GuestSpeaker</u></b> <ul style="list-style-type: none"> <li>GuestSpeakerID</li> <li>Speaker_Name</li> <li>Speaker_Surname</li> <li>Gender</li> <li>DOB</li> <li>Email</li> <li>Phone</li> <li>Hourly_Rate</li> <li>Theme</li> </ul>	Deleted by Volunteer by selecting Delete Speaker.
<b>5.5 Invite Guest Speaker</b>	Invite Guest Speaker Details	To invite a guest speaker to a function.	When a function is created.	<b><u>GuestSpeaker</u></b> <ul style="list-style-type: none"> <li>GuestSpeakerID</li> <li>Speaker_Name</li> <li>Speaker_Surname</li> <li>Gender</li> <li>DOB</li> <li>Email</li> <li>Phone</li> <li>Hourly_Rate</li> <li>Theme</li> </ul>	Invited by Volunteer by use of an Email.  Details on Email are as follows:  Name of Event

Use Case	Description	Purpose	When will it be Used	Entities and Attributes	Logical Layout
				<b>FunctionSpeaker:</b> <ul style="list-style-type: none"> <li>FunctionSpeakerID</li> <li>GuestSpeakerID</li> <li>FunctionID</li> <li>Name</li> <li>Date</li> <li>Time</li> <li>Status</li> </ul> <b>Events:</b> <ul style="list-style-type: none"> <li>EventID</li> <li>EventTypeID</li> <li>Name</li> <li>Summary</li> <li>Details</li> <li>Time</li> <li>Date</li> <li>VenueID</li> </ul>	Theme of Event Summary Description Date Time Venue
<b>5.6 Confirm Guest Speaker</b>	Response Details	To update the event information in order to finalise event.	As a result of creating a function event and sending an email invitation.	<b>GuestSpeaker</b> <ul style="list-style-type: none"> <li>GuestSpeakerID</li> <li>Speaker_Name</li> <li>Speaker_Surname</li> <li>Gender</li> <li>DOB</li> <li>Email</li> <li>Phone</li> <li>Hourly_Rate</li> <li>Theme</li> </ul> <b>FunctionSpeaker:</b> <ul style="list-style-type: none"> <li>FunctionSpeakerID</li> <li>GuestSpeakerID</li> <li>FunctionID</li> <li>Name</li> <li>Date</li> <li>Time</li> <li>Status</li> </ul>	Provided by Guest Speaker via Email. Details include: Response of Acceptance or Rejection Terms



## 6.2.6 Community Engagement Input Interface Description

Use Case	Description	Purpose	When will it be Used	Entities and Attributes	Logical Layout
<b>6.1 Upload Community Engagement Content</b>	Upload Community Engagement Content Details	To upload new community engagement content.	As a result of creating a community outreach event.	<b><u>CommContent:</u></b> <ul style="list-style-type: none"> <li>CommContentID</li> <li>ComID</li> <li>Theme</li> <li>Summary</li> <li>Description</li> <li>Link</li> <li>Status</li> </ul> <b><u>Content:</u></b> <ul style="list-style-type: none"> <li>ContentID</li> <li>Name</li> <li>Author</li> <li>DatePublished</li> <li>Link</li> </ul>	Document uploaded by Volunteer with details:  Name  Theme  Summary  Description  Status  Author  Date  Link
<b>6.2 Search Community Engagement Content</b>	Search Community Engagement Content Details	To search for existing community content.	Ad hoc basis	<b><u>CommContent:</u></b> <ul style="list-style-type: none"> <li>CommContentID</li> <li>ComID</li> <li>Theme</li> <li>Summary</li> <li>Description</li> <li>Link</li> <li>Status</li> </ul> <b><u>Content:</u></b> <ul style="list-style-type: none"> <li>ContentID</li> <li>Name</li> <li>Author</li> <li>DatePublished</li> <li>Link</li> </ul>	System displays content information details such as:  Name  Theme  Summary  Description  Link  Status (based on admin rights)  Date Published

Use Case	Description	Purpose	When will it be Used	Entities and Attributes	Logical Layout
					(Based on admin rights)  Author
<b>6.3 Delete Community Engagement Content</b>	Delete Community Engagement Content Details	To delete existing content.	Ad hoc basis	<b><u>CommContent:</u></b> <ul style="list-style-type: none"> <li>• CommContentID</li> <li>• ComID</li> <li>• Theme</li> <li>• Summary</li> <li>• Description</li> <li>• Link</li> <li>• Status</li> </ul> <b><u>Content:</u></b> <ul style="list-style-type: none"> <li>• ContentID</li> <li>• Name</li> <li>• Author</li> <li>• DatePublished</li> <li>• Link</li> </ul>	Deleted by Volunteer by selecting Delete Content option.
<b>6.4 Lock Community Engagement Content</b>	Lock Community Engagement Content Details	To lock existing content.	Ad hoc basis	<b><u>CommContent:</u></b> <ul style="list-style-type: none"> <li>• CommContentID</li> <li>• ComID</li> <li>• Theme</li> <li>• Summary</li> <li>• Description</li> <li>• Link</li> <li>• Status</li> </ul> <b><u>Content:</u></b> <ul style="list-style-type: none"> <li>• ContentID</li> <li>• Name</li> <li>• Author</li> <li>• DatePublished</li> <li>• Link</li> </ul>	Locked by Volunteer where status is changed.
<b>6.5 Unlock Community Engagement</b>	Unlock Community Engagement Details	To unlock existing content.	Ad hoc basis	<b><u>CommContent:</u></b> <ul style="list-style-type: none"> <li>• CommContentID</li> <li>• ComID</li> <li>• Theme</li> <li>• Summary</li> </ul>	Unlocked by Volunteer where status is changed.

Use Case	Description	Purpose	When will it be Used	Entities and Attributes	Logical Layout
ent Content				<ul style="list-style-type: none"><li>• Description</li><li>• Link</li><li>• Status</li></ul> <b><u>Content:</u></b> <ul style="list-style-type: none"><li>• ContentID</li><li>• Name</li><li>• Author</li><li>• DatePublished</li><li>• Link</li></ul>	

## 6.2.7 Lecture Input Interface Description

Use Case	Description	Purpose	When will it be Used	Entities and Attributes	Logical Layout
<b>7.1 Upload Lecture Content</b>	Lecture Content Upload details.	To upload new lecture content	When the volunteer wants to upload lecture content	<b><u>LectureContent:</u></b> <ul style="list-style-type: none"> <li>• LecContID</li> <li>• Theme</li> <li>• Summary</li> <li>• Description</li> <li>• Link</li> <li>• Status</li> </ul> <b><u>Content:</u></b> <ul style="list-style-type: none"> <li>• ContentID</li> <li>• Name</li> <li>• Author</li> <li>• Date_Published</li> <li>• Link</li> </ul> <b><u>Lecture:</u></b> <ul style="list-style-type: none"> <li>• LectureID</li> </ul>	Uploaded by the Volunteer
<b>7.2 Search Lecture Content</b>	Search Lecture Content details.	To search for lecture Content	When the user seeks to search for lecture content	<b><u>LectureContent:</u></b> <ul style="list-style-type: none"> <li>• LecContID</li> <li>• Theme</li> <li>• Summary</li> <li>• Description</li> <li>• Link</li> <li>• Status</li> </ul> <b><u>Content:</u></b> <ul style="list-style-type: none"> <li>• ContentID</li> <li>• Name</li> <li>• Author</li> <li>• Date_Published</li> <li>• Link</li> </ul> <b><u>Lecture:</u></b> <ul style="list-style-type: none"> <li>• LectureID</li> </ul>	Searched by user
<b>7.3 Delete Lecture Content</b>	Delete Lecture Content details.	To delete Lecture Content	When the volunteer wants to delete lecture content	<b><u>LectureContent:</u></b> <ul style="list-style-type: none"> <li>• LecContID</li> <li>• Theme</li> <li>• Summary</li> <li>• Description</li> <li>• Link</li> <li>• Status</li> </ul> <b><u>Content:</u></b>	Searched & deleted by volunteer

				<ul style="list-style-type: none"> <li>ContentID</li> <li>Name</li> <li>Author</li> <li>Date_Published</li> <li>Link</li> </ul> <p><b>Lecture:</b></p> <ul style="list-style-type: none"> <li>LectureID</li> </ul>	
<b>7.4 Lock Lecture Content</b>	Lock Lecture Content details.	To lock uploaded lecture content	When the volunteer seeks to lock content	<p><b>LectureContent:</b></p> <ul style="list-style-type: none"> <li>LecContID</li> <li>Theme</li> <li>Summary</li> <li>Description</li> <li>Link</li> <li>Status</li> </ul> <p><b>Content:</b></p> <ul style="list-style-type: none"> <li>ContentID</li> <li>Name</li> <li>Author</li> <li>Date_Published</li> <li>Link</li> </ul> <p><b>Lecture:</b></p> <ul style="list-style-type: none"> <li>LectureID</li> </ul>	Searched and locked by volunteer
<b>7.5 Unlock Lecture Content</b>	Unlock Lecture Content details.	To unlock uploaded lecture content	When the volunteer seeks to unlock content	<p><b>LectureContent:</b></p> <ul style="list-style-type: none"> <li>LecContID</li> <li>Theme</li> <li>Summary</li> <li>Description</li> <li>Link</li> <li>Status</li> </ul> <p><b>Content:</b></p> <ul style="list-style-type: none"> <li>ContentID</li> <li>Name</li> <li>Author</li> <li>Date_Published</li> <li>Link</li> </ul> <p><b>Lecture:</b></p> <ul style="list-style-type: none"> <li>LectureID</li> </ul>	Searched and unlocked by volunteer
<b>7.6 Content Review</b>	Lecture Content Review details.	To leave a comment and rating on any lecture Content.	When the user wants to leave a review	<p><b>Review:</b></p> <ul style="list-style-type: none"> <li>ReviewID</li> <li>Comment</li> <li>Rating</li> </ul> <p><b>LectureContent:</b></p>	Searched and reviewed by user

				<ul style="list-style-type: none"><li>• LecContID</li><li>• Theme</li><li>• Summary</li><li>• Description</li><li>• Link</li><li>• Status</li></ul>	
--	--	--	--	---	--

## 6.2.8 Marketing Input Interface Description

Use Case	Description	Purpose	When will it be Used	Entities and Attributes	Logical Layout
<b>8.1 Upload Photo</b>	Photo upload details	To upload a photo.	A user wants to upload a photo for the gallery.	None.	Uploaded by user.
<b>8.2 Post Photo</b>	Photo post to gallery details	To post a photo to the gallery.	Admin needs to review and post photos to the gallery.	None.	Accepted and posted by admin.
	Photo details	To get the details of the photo to post to the gallery	Admin wants to post a photo.	None.	Provided by server provider.
<b>8.3 Delete Photo</b>	Photo deletion details	To delete a photo from the gallery.	Admin wants to delete a photo from a gallery.	None.	Deleted by user.
	Existing gallery photo details	To delete a photo from the gallery.	Admin wants to delete a photo.	None.	Provided to server provider.

## 6.2.9 Reports Input Interface Description

Use Case	Description	Purpose	When will it be Used	Entities and Attributes	Logical Layout
<b>9.1 Generate Class Attendance Report</b>	Generate Class Attendance Report Details.	To generate a class attendance report.	A report needs to be generated on class attendance.	<u><b>Lecture</b></u> <ul style="list-style-type: none"> <li>LectureID</li> </ul> <u><b>Attendance</b></u> <ul style="list-style-type: none"> <li>AttendanceID</li> </ul> <u><b>Event</b></u> <ul style="list-style-type: none"> <li>Date</li> <li>Name</li> </ul>	Created by time event.
<b>9.2 Generate Function Attendance report</b>	Generate Function Attendance Report Details.	To generate a function attendance report.	A report needs to be generated on function attendance.	<u><b>Function</b></u> <ul style="list-style-type: none"> <li>FunctionID</li> </ul> <u><b>Guest Speaker</b></u> <ul style="list-style-type: none"> <li>Speaker_Name</li> </ul> <u><b>Attendance</b></u> <ul style="list-style-type: none"> <li>AttendanceID</li> </ul> <u><b>Event</b></u> <ul style="list-style-type: none"> <li>Date</li> <li>Name</li> </ul>	Created by time event.
<b>9.3 Generate Community Engagement Report</b>	Generate Community Attendance Report Details.	To generate a community engagement attendance report.	A report needs to be generated on community engagement attendance.	<u><b>Community Outreach</b></u> <ul style="list-style-type: none"> <li>ComID</li> </ul> <u><b>Attendance</b></u> <ul style="list-style-type: none"> <li>AttendanceID</li> </ul> <u><b>Event</b></u> <ul style="list-style-type: none"> <li>Date</li> <li>Name</li> </ul>	Created by time event.
<b>9.4 Generate Demographics Report</b>	Generate Demographics Report Details.	To generate a demographic reports.	A report needs to be generated on demographics.	<u><b>Person</b></u> <ul style="list-style-type: none"> <li>DoB</li> <li>DegreeID</li> <li>YearofStudy</li> <li>Race</li> </ul> <u><b>Volunteer</b></u> <ul style="list-style-type: none"> <li>VolunteerID</li> </ul> <u><b>Student</b></u> <ul style="list-style-type: none"> <li>StudentID</li> </ul>	Created by time event.



Use Case	Description	Purpose	When will it be Used	Entities and Attributes	Logical Layout
				<ul style="list-style-type: none"> <li>ResID</li> </ul> <b><u>Language:</u></b> <ul style="list-style-type: none"> <li>LanguageID</li> <li>Language</li> </ul>	
<b>9.5 Generate Event Popularity Report</b>	Generate Event Popularity Report Details.	To generate an event popularity report.	A report needs to be generated on event popularity.	<b><u>Function</u></b> <ul style="list-style-type: none"> <li>FunctionID</li> </ul> <b><u>CommunityOutreach</u></b> <ul style="list-style-type: none"> <li>ComID</li> </ul> <b><u>Lecture</u></b> <ul style="list-style-type: none"> <li>LectureID</li> </ul> <b><u>Attendance</u></b> <ul style="list-style-type: none"> <li>AttendanceID</li> </ul>	Created by time event.
<b>9.6 Generate Donations Report</b>	Generate Donations Attendance Report Details.	To generate a donations report.	A report needs to be generated on donations.	<b><u>Donor</u></b> <ul style="list-style-type: none"> <li>DonorName</li> <li>DonorSurname</li> <li>DonorOrganisation</li> </ul> <b><u>Donations</u></b> <ul style="list-style-type: none"> <li>DonationAmount</li> </ul>	Created by time event.
<b>9.7 Generate User Statistics Report</b>	Generate User Statistics Report Details	To generate a user statistics report.	A report needs to be generated on user statistics.	<b><u>Audit Log</u></b> <ul style="list-style-type: none"> <li>LogInTime</li> <li>LogInDate</li> <li>LogInDuration</li> </ul> <b><u>Person</u></b> <ul style="list-style-type: none"> <li>PersonID</li> </ul> <b><u>User Type</u></b> <ul style="list-style-type: none"> <li>UserTypeID</li> </ul>	Created by time event.

## 6.2.10 Donation Input Interface Description

Use Case	Description	Purpose	When will it be Used	Entities and Attributes	Logical Layout
<b>10.1 Add Donor</b>	New donor details	To add a new donor.	A donor wants to register their details.	<u><b>Donor</b></u> <ul style="list-style-type: none"> <li>• Name</li> <li>• Surname</li> <li>• Organisation Name</li> <li>• Phone Number</li> <li>• Email Address</li> </ul>	Created by donor.
<b>10.2 Receive Donation</b>	New donation details	To receive donation from a donor.	When a donor donates.	<u><b>Donor</b></u> <ul style="list-style-type: none"> <li>• Email Address</li> </ul> <u><b>Donation</b></u> <ul style="list-style-type: none"> <li>• ProofOfPayment</li> <li>• DonationAmount</li> </ul>	Submitted by donor.

## 6.3 Conclusion

6.3.1 The above table describes the inputs that flowed in the context diagram. The input table described what flows went into the context diagram and described who initiated them and why those processes were performed.





## 7. Reports and Other Outputs

### 7.1 Introduction

7.1.1 The following section gives a complete description of the reports and outputs for each requirement of the system with a description of the purpose of each output, when it is used, the entities and attributes involved as well as a logical layout of the input.

### 7.2 Output Description

#### 7.2.1 User Output Description

Use Case	Description	Purpose	When is it Used	Entities and Attributes	Logical Layout
<b>1.1 Check Forgotten Password</b>	Change Password Prompt	To Check for a forgotten password.	When the user forgets their password	<u>Person</u> <ul style="list-style-type: none"> <li>PersonID</li> <li>Email</li> <li>Password</li> </ul> <u>SecurityAnswer</u> <ul style="list-style-type: none"> <li>AnswerID</li> <li>Answer</li> </ul>	A message that allows access to change password.
<b>1.2 Change Password</b>	None.	To change the users password.	When the user seeks to change their password to a new one.	<u>Person</u> <ul style="list-style-type: none"> <li>PersonID</li> <li>Email</li> <li>Password</li> </ul>	A message that displays that a new password has been created
<b>1.3 Login</b>	Main Menu options	To allow the user to log into the system.	When the user wants to use their password to log into the system.	<u>Person</u> <ul style="list-style-type: none"> <li>PersonID</li> <li>Email</li> <li>Password</li> </ul> <u>UserType</u> <ul style="list-style-type: none"> <li>UserTypeID</li> </ul> <u>AuditLog</u>	A message that shows the login was successful.

				<ul style="list-style-type: none"> <li>• AuditID</li> <li>• LogInTime</li> <li>• LogInDate</li> <li>• LogInDuration</li> </ul>	
<b>AUC 1 Register User</b>	User Type Options	To register a new user onto the system	When a student or volunteer wants to register on the system	<b><u>Person</u></b> <ul style="list-style-type: none"> <li>• PersonID</li> <li>• Email</li> <li>• Password</li> </ul> <b><u>SecurityAnswer</u></b> <ul style="list-style-type: none"> <li>• AnswerID</li> <li>• Answer</li> </ul> <b><u>UserType</u></b> <ul style="list-style-type: none"> <li>• UserTypeID</li> </ul>	A prompt to register according to the user's user type.

## 7.2.2 Volunteer Output Description

Use Case	Description	Purpose	When is it Used	Entities and Attributes	Logical Layout
<b>2.2 Search Volunteer</b>	Main Menu Options	Once the user has successfully searched the volunteer the main menu is displayed so that the user can continue using the system as desired.	Whenever a user searches a volunteer on the system and has obtained the information they wanted to search for.	<u>Person</u> <ul style="list-style-type: none"> <li>Name</li> <li>Surname</li> <li>DoB</li> <li>Email</li> <li>Phone</li> <li></li> </ul>	A menu consisting of different options to choose from.
<b>2.3 Update Volunteer</b>	Confirmation of Volunteer Update Details	To provide the volunteer with the details of the successful update of their profile.	Whenever a volunteer has successfully updated details on their profile.	<u>Person</u> <ul style="list-style-type: none"> <li>PersonID</li> <li>Name</li> <li>Surname</li> <li>DoB</li> <li>Email</li> <li>Phone</li> <li>Race</li> </ul> <u>Volunteer</u> <ul style="list-style-type: none"> <li>VolunteerID</li> <li>VolunteerTypeID</li> </ul> <u>Language</u> <ul style="list-style-type: none"> <li>LanguageID</li> <li>Language</li> </ul> <u>PersonLanguage</u> <ul style="list-style-type: none"> <li>PersonID</li> </ul> LanguageID	A notification is displayed confirming the successful update of a volunteer.
<b>2.4 Delete Volunteer</b>	Confirmation of Volunteer Deletion Details	To provide the admin with the details of the	Whenever admin has successfully	<u>Person</u> <ul style="list-style-type: none"> <li>PersonID</li> <li>Name</li> <li>Surname</li> <li>DoB</li> </ul>	A notification is displayed confirming

		successful deletion of a volunteer.	deleted a volunteer.	<ul style="list-style-type: none"> <li>Email</li> <li>Phone</li> <li>Race</li> </ul> <b><u>Volunteer</u></b> <ul style="list-style-type: none"> <li>VolunteerID</li> <li>VolunteerTypeID</li> </ul> <b><u>Language</u></b> <ul style="list-style-type: none"> <li>LanguageID</li> <li>Language</li> </ul> <b><u>PersonLanguage</u></b> <ul style="list-style-type: none"> <li>PersonID</li> <li>LanguageID</li> </ul>	the successful deletion of a volunteer.
<b>2.5 Add Volunteer Type</b>	Confirmation of Volunteer Type Addition Details	To provide the admin with the details of the successful addition of a volunteer type.	Whenever admin has successfully added a volunteer type.	<b><u>VolunteerType</u></b> <ul style="list-style-type: none"> <li>VolunteerTypeID</li> <li>VolunteerTypeName</li> <li>Description</li> </ul>	A notification is displayed confirming the successful creation of a volunteer type.
<b>2.6 Search Volunteer Type</b>	Main menu options	Once the user has successfully searched the volunteer type the main menu is displayed so that the user can continue using the system as desired.	Whenever admin searches a volunteer type on the system and has obtained the information they wanted to search for.	<b><u>VolunteerType</u></b> <ul style="list-style-type: none"> <li>VolunteerTypeID</li> <li>VolunteerTypeName</li> <li>Description</li> </ul>	A menu consisting of different options to choose from.
<b>2.7 Update Volunteer Type</b>	Confirmation of Volunteer Type Update Details	To provide the admin with the details of	Whenever admin has successfully updated a	<b><u>VolunteerType</u></b> <ul style="list-style-type: none"> <li>VolunteerTypeID</li> <li>VolunteerTypeName</li> <li>Description</li> </ul>	A notification is displayed



		the successful update of a volunteer type.	volunteer type.		confirming the successful update of a volunteer type.
<b>2.8 Delete Volunteer Type</b>	Confirmation of Volunteer Type Deletion Details	To provide the admin with the details of the successful deletion of a volunteer type.	Whenever admin has successfully deleted a volunteer type.	<b><u>VolunteerType</u></b> <ul style="list-style-type: none"> <li>• VolunteerTypeID</li> <li>• VolunteerTypeName</li> <li>• Description</li> </ul>	A notification is displayed confirming the successful creation of a volunteer type.
<b>2.9 Generate Unique Code</b>	Unique Code	To provide the volunteer with a unique code which they can use to register on the system.	Whenever admin has generated a unique code and has selected the option to send the code to a volunteer	<b><u>UniqueCode</u></b> <ul style="list-style-type: none"> <li>• CodeID</li> <li>• Code</li> <li>• Status</li> </ul>	A message is sent to a volunteer containing the unique code.

## 7.2.3 Student Output Description

Use Case	Description	Purpose	When is it Used	Entities and Attributes	Logical Layout
<b>3.1 Register Student</b>	Confirmation Details	To provide the student with the details of their successful profile registration.	Whenever a student joins TRWLA and has successfully registered on the system.	<b>Person</b> <ul style="list-style-type: none"> <li>PersonID</li> <li>Name</li> <li>Surname</li> <li>DoB</li> <li>Phone</li> <li>Race</li> </ul> <b>Student</b> <ul style="list-style-type: none"> <li>StudentID</li> <li>StudentNumber</li> <li>StudentTypeID</li> <li>DegreeID</li> <li>YearOfStudy</li> </ul> <b>StudentType</b> <ul style="list-style-type: none"> <li>StudentTypeID</li> <li>Name</li> <li>Description</li> </ul> <b>Language</b> <ul style="list-style-type: none"> <li>LanguageID</li> <li>Language</li> </ul> <b>PersonLanguage</b> <ul style="list-style-type: none"> <li>PersonID</li> <li>LanguageID</li> </ul> <b>CurrentDegree</b> <ul style="list-style-type: none"> <li>DegreeID</li> <li>Name</li> </ul>	A notification is displayed confirming the successful registration of the students profile.
<b>3.2 Search Student</b>	Student's Details	Once the user has successfully searched the student the student's details are displayed to provide the user with the information	Whenever a user searches a student on the system.	<b>Person</b> <ul style="list-style-type: none"> <li>PersonID</li> <li>Name</li> <li>Surname</li> <li>DoB</li> <li>Phone</li> <li>Race</li> </ul> <b>Student</b> <ul style="list-style-type: none"> <li>StudentID</li> <li>StudentNumber</li> <li>StudentTypeID</li> <li>DegreeID</li> <li>YearOfStudy</li> </ul> <b>StudentType</b>	The students details including Name, Surname, Date Of Birth, Email Address and Phone Number are displayed to the user.

Use Case	Description	Purpose	When is it Used	Entities and Attributes	Logical Layout
		they were looking for.		<ul style="list-style-type: none"> <li>StudentTypeID</li> <li>Name</li> <li>Description</li> </ul> <u>Language</u> <ul style="list-style-type: none"> <li>LanguageID</li> <li>Language</li> </ul> <u>PersonLanguage</u> <ul style="list-style-type: none"> <li>PersonID</li> <li>LanguageID</li> </ul> <u>CurrentDegree</u> <ul style="list-style-type: none"> <li>DegreeID</li> <li>Name</li> </ul>	
<b>3.3 Update Student</b>	Confirmation Details	To provide the student with the details of the successful update of their profile.	Whenever a student has successfully updated details on their profile.	<u>Person</u> <ul style="list-style-type: none"> <li>PersonID</li> <li>Name</li> <li>Surname</li> <li>DoB</li> <li>Phone</li> <li>Race</li> </ul> <u>Student</u> <ul style="list-style-type: none"> <li>StudentID</li> <li>StudentNumber</li> <li>StudentTypeID</li> <li>DegreeID</li> <li>YearOfStudy</li> </ul> <u>StudentType</u> <ul style="list-style-type: none"> <li>StudentTypeID</li> <li>Name</li> <li>Description</li> </ul> <u>Language</u> <ul style="list-style-type: none"> <li>LanguageID</li> <li>Language</li> </ul> <u>PersonLanguage</u> <ul style="list-style-type: none"> <li>PersonID</li> <li>LanguageID</li> </ul> <u>CurrentDegree</u> <ul style="list-style-type: none"> <li>DegreeID</li> <li>Name</li> </ul>	A notification is displayed confirming the successful update of a student.
<b>3.4 Delete Student</b>	Confirmation of deletion Details	To provide the admin with the details of the	Whenever admin has successfully	<u>Person</u> <ul style="list-style-type: none"> <li>PersonID</li> <li>Name</li> <li>Surname</li> <li>DoB</li> </ul>	A notification is displayed confirming the

Use Case	Description	Purpose	When is it Used	Entities and Attributes	Logical Layout
		successful deletion of a student.	deleted a student.	<ul style="list-style-type: none"> <li>Phone</li> <li>Race</li> </ul> <b><u>Student</u></b> <ul style="list-style-type: none"> <li>StudentID</li> <li>StudentNumber</li> <li>StudentTypeID</li> <li>DegreeID</li> <li>YearOfStudy</li> </ul> <b><u>StudentType</u></b> <ul style="list-style-type: none"> <li>StudentTypeID</li> <li>Name</li> <li>Description</li> </ul> <b><u>Language</u></b> <ul style="list-style-type: none"> <li>LanguageID</li> <li>Language</li> </ul> <b><u>PersonLanguage</u></b> <ul style="list-style-type: none"> <li>PersonID</li> <li>LanguageID</li> </ul> <b><u>CurrentDegree</u></b> <ul style="list-style-type: none"> <li>DegreeID</li> <li>Name</li> </ul>	successful deletion of a student.
<b>3.5 Generate Graduate List</b>	Graduate List	To provide admin and management with a list of students who qualify to graduate from TRWLA.	At the end of the TRWLA year.	<b><u>Student:</u></b> <ul style="list-style-type: none"> <li>StudentID</li> </ul> <b><u>Graduate:</u></b> <ul style="list-style-type: none"> <li>GraduationID</li> <li>StudentID</li> <li>GradYear</li> </ul> <b><u>Progress</u></b> <ul style="list-style-type: none"> <li>ProgressID</li> </ul>	A list of graduates is displayed and sent in a message to admin.
<b>3.6 Add Student Type</b>	Confirmation of student type creation Details	To provide the admin with the details of the successful addition of a student type.	Whenever admin has successfully added a student type.	<b><u>StudentType</u></b> <ul style="list-style-type: none"> <li>StudentTypeID</li> <li>StudentTypeName</li> <li>Description</li> </ul>	A notification is displayed confirming the successful creation of a student type.
<b>3.7 Search</b>	Student type details	Once the user has	Whenever admin	<b><u>StudentType</u></b> <ul style="list-style-type: none"> <li>StudentTypeID</li> </ul>	A menu consisting

Use Case	Description	Purpose	When is it Used	Entities and Attributes	Logical Layout
<b>Student Type</b>		successfully searched the student type the main menu is displayed so that the user can continue using the system as desired.	searches a student type on the system and has obtained the information they wanted to search for.	<ul style="list-style-type: none"> <li>• StudentTypeName</li> <li>• Description</li> </ul>	of different options to choose from.
<b>3.8 Update Student Type</b>	Confirmation of successful update Details	To provide the admin with the details of the successful update of a student type.	Whenever admin has successfully updated a student type.	<u>StudentType</u> <ul style="list-style-type: none"> <li>• StudentTypeID</li> <li>• StudentTypeName</li> <li>• Description</li> </ul>	A notification is displayed confirming the successful update of a student type.
<b>3.9 Delete Student Type</b>	Confirmation of deletion Details	To provide the admin with the details of the successful deletion of a student type.	Whenever admin has successfully deleted a student type.	<u>StudentType</u> <ul style="list-style-type: none"> <li>• StudentTypeID</li> <li>• StudentTypeName</li> <li>• Description</li> </ul>	A notification is displayed confirming the successful creation of a student type.

## 7.2.4 Events Output Description

Use Case	Description	Purpose	When is it Used	Entities and Attributes	Logical Layout
<b>4.1 Create Venue</b>	Confirmation of New Venue Created	To add a new venue	When a new venue needs to be created	<b>Venue:</b> <ul style="list-style-type: none"> <li>VenueID</li> <li>Name</li> <li>Capacity</li> <li>Accessibility</li> </ul> <b>VenueType:</b> <ul style="list-style-type: none"> <li>VenueTypeID</li> <li>Name</li> </ul> <b>Address:</b> <ul style="list-style-type: none"> <li>AddressID</li> <li>StreetName</li> <li>StreetNumber</li> <li>Suburb</li> <li>City</li> <li>Province</li> </ul>	A message is displayed showing the user that a venue has been created
<b>4.2 Search Venue</b>	Displayed Venue Information	To search a venue's details	When a user wishes to view the venue's details	<b>Venue:</b> <ul style="list-style-type: none"> <li>VenueID</li> <li>Name</li> <li>Capacity</li> <li>Accessibility</li> </ul> <b>VenueType:</b> <ul style="list-style-type: none"> <li>VenueTypeID</li> <li>Name</li> </ul> <b>Address:</b> <ul style="list-style-type: none"> <li>AddressID</li> <li>StreetName</li> <li>StreetNumber</li> <li>Suburb</li> <li>City</li> <li>Province</li> </ul>	The venue information is displayed to the user
<b>4.3 Update Venue</b>	Confirmation of Venue Updated	To update venue information	To update the venue information	<b>Venue:</b> <ul style="list-style-type: none"> <li>VenueID</li> <li>Name</li> </ul>	A message is displayed showing the

Use Case	Description	Purpose	When is it Used	Entities and Attributes	Logical Layout
			if entered incorrectly or if the venue information has changed.	<ul style="list-style-type: none"> <li>Capacity</li> <li>Accessibility</li> </ul> <b>VenueType:</b> <ul style="list-style-type: none"> <li>VenueTypeID</li> <li>Name</li> </ul> <b>Address:</b> <ul style="list-style-type: none"> <li>AddressID</li> <li>StreetName</li> <li>StreetNumber</li> <li>Suburb</li> <li>City</li> <li>Province</li> </ul>	volunteer that the venue has been updated.
<b>4.4 Delete Venue</b>	Confirmation of Deleted Venue	To delete a venue off of the TRWLA system	If the venue is no longer needed then it will be deleted off of the system	<b>Venue:</b> <ul style="list-style-type: none"> <li>VenueID</li> <li>Name</li> <li>Capacity</li> <li>Accessibility</li> </ul> <b>VenueType:</b> <ul style="list-style-type: none"> <li>VenueTypeID</li> <li>Name</li> </ul> <b>Address:</b> <ul style="list-style-type: none"> <li>AddressID</li> <li>StreetName</li> <li>StreetNumber</li> <li>Suburb</li> <li>City</li> <li>Province</li> </ul>	A message is displayed showing the volunteer that the venue has been deleted.
<b>4.5 Create Event</b>	Event Creation Confirmation	To create new Event	When a new event needs to be created as it will be in the TRWLA system, it can be added here.	<b>Events:</b> <ul style="list-style-type: none"> <li>EventID</li> <li>EventTypeID</li> <li>Name</li> <li>Summary</li> <li>Details</li> <li>Time</li> <li>Date</li> </ul>	A message is displayed showing the user that an event has been created

Use Case	Description	Purpose	When is it Used	Entities and Attributes	Logical Layout
				<ul style="list-style-type: none"> <li>VenueID</li> </ul> <p><b>Function:</b></p> <ul style="list-style-type: none"> <li>FunctionID</li> <li>GuestSpeakerID</li> </ul> <p><b>Lecture:</b></p> <ul style="list-style-type: none"> <li>LectureID</li> </ul> <p><b>CommunityOutreach:</b></p> <ul style="list-style-type: none"> <li>ComID</li> </ul> <p><b>Residence:</b></p> <ul style="list-style-type: none"> <li>ResidenceID</li> <li>Name</li> </ul> <p><b>Content:</b></p> <ul style="list-style-type: none"> <li>ContentID</li> <li>Name</li> <li>Author</li> <li>DatePublished</li> <li>Link</li> </ul>	
<b>4.6 Search Event</b>	Displayed Event Information	To display the particular events information	User wants to search for an event or they are creating an event.	<p><b>Events:</b></p> <ul style="list-style-type: none"> <li>EventID</li> <li>EventTypeID</li> <li>Name</li> <li>Summary</li> <li>Details</li> <li>Time</li> <li>Date</li> <li>VenueID</li> </ul> <p><b>Function:</b></p> <ul style="list-style-type: none"> <li>FunctionID</li> <li>GuestSpeakerID</li> </ul> <p><b>Lecture:</b></p> <ul style="list-style-type: none"> <li>LectureID</li> </ul> <p><b>CommunityOutreach:</b></p> <ul style="list-style-type: none"> <li>ComID</li> </ul> <p><b>Residence:</b></p>	The event information is displayed to the user



Use Case	Description	Purpose	When is it Used	Entities and Attributes	Logical Layout
				<ul style="list-style-type: none"> <li>ResidenceID</li> <li>Name</li> </ul> <p><b>Content:</b></p> <ul style="list-style-type: none"> <li>ContentID</li> <li>Name</li> <li>Author</li> <li>DatePublished</li> <li>Link</li> </ul> <p><b>TRWLATimeline</b></p> <ul style="list-style-type: none"> <li>TimelineID</li> </ul>	
<b>4.7 RSVP To an Event</b>	Confirmation of Event RSVP	To RSVP to an upcoming event on the sytem	When a user wishes to RSVP to an event.	<p><b>Events:</b></p> <ul style="list-style-type: none"> <li>EventID</li> <li>EventTypeID</li> <li>Name</li> <li>Summary</li> <li>Details</li> <li>Time</li> <li>Date</li> <li>VenueID</li> </ul> <p><b>RSVP Event:</b></p> <ul style="list-style-type: none"> <li>RSVP_EventID</li> <li>EventID</li> <li>PersonID</li> </ul> <p><b>Person:</b></p> <ul style="list-style-type: none"> <li>PersonID</li> <li>Name</li> <li>Surname</li> <li>Email</li> <li>Phone</li> </ul>	The user receives a message that they have RSVP'd to the event.
<b>4.8 View TRWLA Timeline</b>	TRWLA Timeline Details	To view the upcoming events and activities within TRWLA	When a user wants to view the upcoming events on	<p><b>Events:</b></p> <ul style="list-style-type: none"> <li>EventID</li> <li>EventTypeID</li> <li>Name</li> <li>Summary</li> <li>Details</li> </ul>	The user is prompted with all of the upcoming events and

Use Case	Description	Purpose	When is it Used	Entities and Attributes	Logical Layout
			the TRWLA timeline.	<ul style="list-style-type: none"> <li>Time</li> <li>Date</li> <li>VenueID</li> </ul> <p><b>Function:</b></p> <ul style="list-style-type: none"> <li>FunctionID</li> <li>GuestSpeakerID</li> </ul> <p><b>Lecture:</b></p> <ul style="list-style-type: none"> <li>LectureID</li> <li>LectureContent</li> </ul> <p><b>CommunityOutreach:</b></p> <ul style="list-style-type: none"> <li>ComID</li> <li>EventID</li> </ul> <p><b>Residence:</b></p> <ul style="list-style-type: none"> <li>ResidenceID</li> <li>ResidenceName</li> </ul>	activities on the TRWLA system.
<b>4.9 Update Event Information</b>	Confirmation of the Event Update	To update the details of an event that have changed	Information is incorrectly provided or details about the event have changed.	<p><b>Events:</b></p> <ul style="list-style-type: none"> <li>EventID</li> <li>EventTypeID</li> <li>Name</li> <li>Summary</li> <li>Details</li> <li>Time</li> <li>Date</li> <li>VenueID</li> </ul> <p><b>Function:</b></p> <ul style="list-style-type: none"> <li>FunctionID</li> <li>GuestSpeakerID</li> </ul> <p><b>Lecture:</b></p> <ul style="list-style-type: none"> <li>LectureID</li> </ul> <p><b>CommunityOutreach:</b></p> <ul style="list-style-type: none"> <li>ComID</li> </ul> <p><b>Residence:</b></p> <ul style="list-style-type: none"> <li>ResidenceID</li> <li>Name</li> </ul>	A message is displayed showing the volunteer that the event has been updated.

Use Case	Description	Purpose	When is it Used	Entities and Attributes	Logical Layout
				<b>Content:</b> <ul style="list-style-type: none"> <li>ContentID</li> <li>Name</li> <li>Author</li> <li>DatePublished</li> <li>Link</li> </ul>	
<b>4.10 Cancel Event</b>	Confirmation of Event Cancellation	To delete the event information off of the system	The event is no longer going to take place and thus it will be deleted off of the system.	<b>Events:</b> <ul style="list-style-type: none"> <li>EventID</li> <li>EventTypeID</li> <li>Name</li> <li>Summary</li> <li>Details</li> <li>Time</li> <li>Date</li> <li>VenueID</li> </ul> <b>Function:</b> <ul style="list-style-type: none"> <li>FunctionID</li> <li>GuestSpeakerID</li> </ul> <b>Lecture:</b> <ul style="list-style-type: none"> <li>LectureID</li> </ul> <b>CommunityOutreach:</b> <ul style="list-style-type: none"> <li>ComID</li> </ul> <b>Residence:</b> <ul style="list-style-type: none"> <li>ResidenceID</li> <li>Name</li> </ul> <b>Content:</b> <ul style="list-style-type: none"> <li>ContentID</li> <li>Name</li> <li>Author</li> <li>DatePublished</li> <li>Link</li> </ul>	A message is displayed showing the volunteer that the event has been cancelled.
<b>4.11 Log Event Attendance</b>	Successfully logged Event Attendance	To log the event attendance of a student	When a student is attending an event	<b>Person:</b> <ul style="list-style-type: none"> <li>PersonID</li> <li>Name</li> </ul>	The volunteer receives confirmation that the

Use Case	Description	Purpose	When is it Used	Entities and Attributes	Logical Layout
		towards their TRWLA certification	towards their certification.	<ul style="list-style-type: none"> <li>Surname</li> </ul> <b><u>Student:</u></b> <ul style="list-style-type: none"> <li>StudentID</li> <li>StudentNumber</li> </ul> <b><u>RSVP_Event:</u></b> <ul style="list-style-type: none"> <li>RSVP_EventID</li> <li>EventID</li> <li>PersonID</li> </ul> <b><u>Attendance:</u></b> <ul style="list-style-type: none"> <li>Attendance_ID</li> <li>EventID</li> <li>PersonID</li> </ul> <b><u>Progress:</u></b> <ul style="list-style-type: none"> <li>ProgressID</li> <li>ProgressCount</li> </ul>	student has been RSVP'd to the current event.
<b>4.12 Send Notification</b>	Confirmation of Notification Sent	To send a notification to users	When information needs to be sent to users.	<b><u>Person:</u></b> <ul style="list-style-type: none"> <li>PersonID</li> <li>Name</li> <li>Surname</li> <li>Email</li> <li>Phone</li> </ul>	The volunteer is prompted with a message that the notification has been sent to the selected recipients.

## 7.2.5 Function Output Description

Use Case	Description	Purpose	When is it Used	Entities and Attributes	Logical Layout
<b>5.1 Register Guest Speaker</b>	Successful Registration Message	To register a new guest speaker.	As a result of a new guest speaker registered.	<b><u>GuestSpeaker</u></b> <ul style="list-style-type: none"> <li>• GuestSpeakerID</li> <li>• Speaker_Name</li> <li>• Speaker_Surname</li> <li>• Gender</li> <li>• DOB</li> <li>• Email</li> <li>• Phone</li> <li>• Hourly_Rate</li> <li>• Theme</li> </ul> <b><u>SpeakerType</u></b> <ul style="list-style-type: none"> <li>• SpeakerTypeID</li> <li>• TypeName</li> <li>• TypeDescription</li> </ul>	A message that displays that the registration of the guest speaker was successful.
<b>5.2 Search Guest Speaker</b>	Guest Speaker Information	Display guest speaker information	As a result of a guest speaker search.	<b><u>GuestSpeaker</u></b> <ul style="list-style-type: none"> <li>• GuestSpeakerID</li> <li>• Speaker_Name</li> <li>• Speaker_Surname</li> <li>• Gender</li> <li>• DOB</li> <li>• Email</li> <li>• Phone</li> <li>• Hourly_Rate</li> <li>• Theme</li> </ul>	A message that displays the selected guest speaker's information to the user such as:  Name  Surname  Gender  Date of Birth  Email Address  Phone Number

Use Case	Description	Purpose	When is it Used	Entities and Attributes	Logical Layout
					Hourly Rate (based on admin rights)  Theme
<b>5.3 Update Guest Speaker</b>	Update Confirmation Message	To update a guest speaker's details.	As a result of updated details to guest speaker.	<b><u>GuestSpeaker</u></b> <ul style="list-style-type: none"> <li>• GuestSpeakerID</li> <li>• Speaker_Name</li> <li>• Speaker_Surname</li> <li>• Gender</li> <li>• DOB</li> <li>• Email</li> <li>• Phone</li> <li>• Hourly_Rate</li> <li>• Theme</li> </ul> <b><u>SpeakerType</u></b> <ul style="list-style-type: none"> <li>• SpeakerTypeID</li> <li>• TypeName</li> <li>• TypeDescription</li> </ul>	A message that displays that the guest speaker's details have been updated successfully.
<b>5.4 Delete Guest Speaker</b>	Deletion Confirmation Message	To delete a guest speaker's details.	As a result of deletion.	<b><u>GuestSpeaker</u></b> <ul style="list-style-type: none"> <li>• GuestSpeakerID</li> <li>• Speaker_Name</li> <li>• Speaker_Surname</li> <li>• Gender</li> <li>• DOB</li> <li>• Email</li> <li>• Phone</li> <li>• Hourly_Rate</li> <li>• Theme</li> </ul>	A message that displays that the guest speaker's details have been deleted successfully.
<b>5.5 Invite Guest Speaker</b>	Invitation Details	To invite a guest speaker to a function.	As a result of a function created and	<b><u>GuestSpeaker</u></b> <ul style="list-style-type: none"> <li>• GuestSpeakerID</li> <li>• Speaker_Name</li> <li>• Speaker_Surname</li> </ul>	A message that displays that the message

Use Case	Description	Purpose	When is it Used	Entities and Attributes	Logical Layout
			email sent.	<ul style="list-style-type: none"> <li>Gender</li> <li>DOB</li> <li>Email</li> <li>Phone</li> <li>Hourly_Rate</li> <li>Theme</li> </ul> <b>FunctionSpeaker:</b> <ul style="list-style-type: none"> <li>FunctionSpeakerID</li> <li>GuestSpeakerID</li> <li>FunctionID</li> <li>Name</li> <li>Date</li> <li>Time</li> <li>Status</li> </ul> <b>Events:</b> <ul style="list-style-type: none"> <li>EventID</li> <li>EventTypeID</li> <li>Name</li> <li>Summary</li> <li>Details</li> <li>Time</li> <li>Date</li> <li>VenueID</li> </ul>	has been sent to the guest speaker containing the invitation to the function.





## 7.2.6 Community Engagement Output Description

Use Case	Description	Purpose	When is it Used	Entities and Attributes	Logical Layout
<b>6.1 Upload Community Engagement Content</b>	Confirmation Message	To upload new community engagement content.	As a result of an upload.	<b><u>CommContent:</u></b> <ul style="list-style-type: none"> <li>• CommContentID</li> <li>• ComID</li> <li>• Theme</li> <li>• Summary</li> <li>• Description</li> <li>• Link</li> <li>• Status</li> </ul> <b><u>Content:</u></b> <ul style="list-style-type: none"> <li>• ContentID</li> <li>• Name</li> <li>• Author</li> <li>• DatePublished</li> <li>• Link</li> </ul>	A message that displays that the content has been uploaded successfully.
<b>6.2 Search Community Engagement Content</b>	Content Information	Display selected content information.	As a result of a search.	<b><u>CommContent:</u></b> <ul style="list-style-type: none"> <li>• CommContentID</li> <li>• ComID</li> <li>• Theme</li> <li>• Summary</li> <li>• Description</li> <li>• Link</li> <li>• Status</li> </ul> <b><u>Content:</u></b> <ul style="list-style-type: none"> <li>• ContentID</li> <li>• Name</li> <li>• Author</li> <li>• DatePublished</li> <li>• Link</li> </ul>	A message that displays the selected content to the user.
<b>6.3 Delete Community Engagement Content</b>	Deletion Confirmation Message	Delete Content.	As a result of content deletion.	<b><u>CommContent:</u></b> <ul style="list-style-type: none"> <li>• CommContentID</li> <li>• ComID</li> <li>• Theme</li> <li>• Summary</li> <li>• Description</li> <li>• Link</li> </ul>	A message that displays that the content has been

Use Case	Description	Purpose	When is it Used	Entities and Attributes	Logical Layout
Content				<ul style="list-style-type: none"> <li>Status</li> </ul> <b><u>Content:</u></b> <ul style="list-style-type: none"> <li>ContentID</li> <li>Name</li> <li>Author</li> <li>DatePublished</li> <li>Link</li> </ul>	successfully deleted.
6.4 Lock Community Engagement Content	Lock Confirmation Message	Lock Existing Content.	As a result of a content lock	<b><u>CommContent:</u></b> <ul style="list-style-type: none"> <li>CommContentID</li> <li>ComID</li> <li>Theme</li> <li>Summary</li> <li>Description</li> <li>Link</li> <li>Status</li> </ul> <b><u>Content:</u></b> <ul style="list-style-type: none"> <li>ContentID</li> <li>Name</li> <li>Author</li> <li>DatePublished</li> <li>Link</li> </ul>	A message that displays that the content has been successfully locked to other users.
6.5 Unlock Community Engagement Content	Unlock Confirmation Message	Unlock Existing Content.	As a result of a content unlock.	<b><u>CommContent:</u></b> <ul style="list-style-type: none"> <li>CommContentID</li> <li>ComID</li> <li>Theme</li> <li>Summary</li> <li>Description</li> <li>Link</li> <li>Status</li> </ul> <b><u>Content:</u></b> <ul style="list-style-type: none"> <li>ContentID</li> <li>Name</li> <li>Author</li> <li>DatePublished</li> <li>Link</li> </ul>	A message that displays that the content has been successfully unlocked and is accessible to other users.



## 7.2.7 Lecture Output Description

Use Case	Description	Purpose	When is it Used	Entities and Attributes	Logical Layout
<b>7.1 Upload Lecture Content</b>	Confirmation of Successfully Uploaded Lecture Content	To upload new lecture content	When the volunteer wants to upload lecture content	<p><b><u>LectureContent:</u></b></p> <ul style="list-style-type: none"> <li>• LecContID</li> <li>• Theme</li> <li>• Summary</li> <li>• Description</li> <li>• Link</li> <li>• Status</li> </ul> <p><b><u>Content:</u></b></p> <ul style="list-style-type: none"> <li>• ContentID</li> <li>• Name</li> <li>• Author</li> <li>• Date_Published</li> <li>• Link</li> </ul> <p><b><u>Lecture:</u></b></p> <ul style="list-style-type: none"> <li>• LectureID</li> </ul>	A message that shows the content was uploaded successfully.
<b>7.2 Search Lecture Content</b>	Retrieved Search Results	To search for lecture Content	When the user seeks to search for lecture content	<p><b><u>LectureContent:</u></b></p> <ul style="list-style-type: none"> <li>• LecContID</li> <li>• Theme</li> <li>• Summary</li> <li>• Description</li> <li>• Link</li> <li>• Status</li> </ul> <p><b><u>Content:</u></b></p> <ul style="list-style-type: none"> <li>• ContentID</li> <li>• Name</li> <li>• Author</li> <li>• Date_Published</li> <li>• Link</li> </ul> <p><b><u>Lecture:</u></b></p> <ul style="list-style-type: none"> <li>• LectureID</li> </ul>	A message that shows the retrieved search results.

Use Case	Description	Purpose	When is it Used	Entities and Attributes	Logical Layout
<b>7.3 Delete Lecture Content</b>	Notification of Successfully Deleted Lecture Content	To delete Lecture Content	When the volunteer wants to delete lecture content	<p><b><u>LectureContent:</u></b></p> <ul style="list-style-type: none"> <li>• LecContID</li> <li>• Theme</li> <li>• Summary</li> <li>• Description</li> <li>• Link</li> <li>• Status</li> </ul> <p><b><u>Content:</u></b></p> <ul style="list-style-type: none"> <li>• ContentID</li> <li>• Name</li> <li>• Author</li> <li>• Date_Published</li> <li>• Link</li> </ul> <p><b><u>Lecture:</u></b></p> <ul style="list-style-type: none"> <li>• LectureID</li> </ul>	A message that shows notification of successfully deleted lecture content.
<b>7.4 Lock Lecture Content</b>	Notification of Successfully locked Lecture Content	To lock uploaded lecture content	When the volunteer seeks to lock content	<p><b><u>LectureContent:</u></b></p> <ul style="list-style-type: none"> <li>• LecContID</li> <li>• Theme</li> <li>• Summary</li> <li>• Description</li> <li>• Link</li> <li>• Status</li> </ul> <p><b><u>Content:</u></b></p> <ul style="list-style-type: none"> <li>• ContentID</li> <li>• Name</li> <li>• Author</li> <li>• Date_Published</li> <li>• Link</li> </ul> <p><b><u>Lecture:</u></b></p> <ul style="list-style-type: none"> <li>• LectureID</li> </ul>	A message that shows notification of successfully locked lecture content.
<b>7.5 Unlock Lecture Content</b>	Notification of Successfully unlocked	To unlock uploaded lecture content	When the volunteer seeks to	<p><b><u>LectureContent:</u></b></p> <ul style="list-style-type: none"> <li>• LecContID</li> <li>• Theme</li> <li>• Summary</li> </ul>	A message that shows notification of

Use Case	Description	Purpose	When is it Used	Entities and Attributes	Logical Layout
	Lecture Content		unlock content	<ul style="list-style-type: none"> <li>Description</li> <li>Link</li> <li>Status</li> </ul> <u>Content:</u> <ul style="list-style-type: none"> <li>ContentID</li> <li>Name</li> <li>Author</li> <li>Date_Published</li> <li>Link</li> </ul> <u>Lecture:</u> <ul style="list-style-type: none"> <li>LectureID</li> </ul>	successfully unlocked lecture content.
<b>7.6 Content Review</b>	Notification of Review	To leave a comment and rating on any lecture Content.	When the user wants to leave a review	<u>Review:</u> <ul style="list-style-type: none"> <li>ReviewID</li> <li>Comment</li> <li>Rating</li> </ul> <u>LectureContent:</u> <ul style="list-style-type: none"> <li>LecContID</li> <li>Theme</li> <li>Summary</li> <li>Description</li> <li>Link</li> <li>Status</li> </ul>	A message that displays a content review has been successfully made.

## 7.2.8 Marketing Output Description

Use Case	Description	Purpose	When is it Used	Entities and Attributes	Logical Layout
<b>8.1 Upload Photo</b>	Confirmation of photo upload details.	To upload a new photo for admin to review and post.	When a user successfully uploads a photo.	None.	A message that informs the user that the photo upload was successful.
	Photo uploaded details	To notify admin that a new photo has been uploaded and is ready for review.	When a user successfully uploads a photo.	None.	A notification that informs admin that a photo uploaded.
	Photo upload details	To notify the server provider a photo has been uploaded	When a user successfully uploads a photo.	None.	This informs the server provider that a photo has been uploaded.
<b>8.2 Post Photo</b>	Confirmation of photo post to gallery details.	To review and post a photo to the gallery.	When admin successfully adds a photo to the gallery.	None.	A message that informs admin that the photo was successfully added to the gallery.
<b>8.3 Delete Photo</b>	Confirmation on successful deletion.	To delete a photo from the gallery.	When admin successfully deletes a	None.	A message that informs admin that the photo was successfully

Use Case	Description	Purpose	When is it Used	Entities and Attributes	Logical Layout
<b>8.1 Upload Photo</b>	Confirmation of photo upload details.	To upload a new photo for admin to review and post.	When a user successfully uploads a photo.	None.	A message that informs the user that the photo upload was successful.
	Photo uploaded details	To notify admin that a new photo has been uploaded and is ready for review.	When a user successfully uploads a photo.	None.	A notification that informs admin that a photo uploaded.
	Photo upload details	To notify the server provider a photo has been uploaded	When a user successfully uploads a photo.	None.	This informs the server provider that a photo has been uploaded.
			photo from the gallery.		deleted from the gallery.
	Photo deletion details	To delete a photo from the gallery	To delete the photo details.		To notify the server provider to delete the photo.



## 7.2.9 Reports Output Description

Use Case	Description	Purpose	When is it Used	Entities and Attributes	Logical Layout
<b>9.1 Generate Class Attendance Report</b>	Class Attendance Report	To generate a report on class attendance.	A class attendance report is successfully generated at year-end.	<u><b>Lecture</b></u> <ul style="list-style-type: none"> <li>LectureID</li> </ul> <u><b>Attendance</b></u> <ul style="list-style-type: none"> <li>AttendanceID</li> </ul> <u><b>Event</b></u> <ul style="list-style-type: none"> <li>Date</li> <li>Name</li> </ul>	A report that shows relevant class attendance information .
<b>9.2 Generate Function Attendance Report</b>	Function Attendance Report	To generate a function attendance report.	A function attendance report is successfully generated at year-end.	<u><b>Function</b></u> <ul style="list-style-type: none"> <li>FunctionID</li> </ul> <u><b>Guest Speaker</b></u> <ul style="list-style-type: none"> <li>Speaker_Name</li> </ul> <u><b>Attendance</b></u> <ul style="list-style-type: none"> <li>AttendanceID</li> </ul> <u><b>Event</b></u> <ul style="list-style-type: none"> <li>Date</li> <li>Name</li> </ul>	A report that shows relevant function attendance information .
<b>9.3 Generate Community Attendance Report</b>	Community Engagement Attendance Report	To generate a community engagement attendance report.	A community engagement attendance report is successfully generated at year-end.	<u><b>Community Outreach</b></u> <ul style="list-style-type: none"> <li>ComID</li> </ul> <u><b>Attendance</b></u> <ul style="list-style-type: none"> <li>AttendanceID</li> </ul> <u><b>Event</b></u> <ul style="list-style-type: none"> <li>Date</li> <li>Name</li> </ul>	A report that shows relevant community engagement attendance information .
<b>9.4 Generate Demographics Report</b>	Demographics Report	To generate a demographics reports.	A demographics report is successfully generated at year-end.	<u><b>Person</b></u> <ul style="list-style-type: none"> <li>DoB</li> <li>DegreeID</li> <li>YearofStudy</li> <li>Race</li> </ul> <u><b>Volunteer</b></u> <ul style="list-style-type: none"> <li>VolunteerID</li> </ul>	A report that shows relevant demographics information .

Use Case	Description	Purpose	When is it Used	Entities and Attributes	Logical Layout
				<u><b>Student</b></u> <ul style="list-style-type: none"> <li>StudentID</li> <li>ResID</li> </ul> <u><b>Language:</b></u> <ul style="list-style-type: none"> <li>LanguageID</li> <li>Language</li> </ul>	
<b>9.5 Generate Event Popularity Report</b>	Event Popularity Report	To generate an event popularity report.	An event popularity report is successfully generated at year-end.	<u><b>Function</b></u> <ul style="list-style-type: none"> <li>FunctionID</li> </ul> <u><b>CommunityOutreach</b></u> <ul style="list-style-type: none"> <li>ComID</li> </ul> <u><b>Lecture</b></u> <ul style="list-style-type: none"> <li>LectureID</li> </ul> <u><b>Attendance</b></u> <ul style="list-style-type: none"> <li>AttendanceID</li> </ul>	A report that shows relevant event popularity information .
<b>9.6 Generate Donations Report</b>	Donations Report	To generate a donations report.	A donations report is successfully generated at year-end.	<u><b>Donor</b></u> <ul style="list-style-type: none"> <li>DonorName</li> <li>DonorSurname</li> <li>DonorOrganisation</li> </ul> <u><b>Donations</b></u> <ul style="list-style-type: none"> <li>DonationAmount</li> </ul>	A report that shows relevant donation information .
<b>9.7 Generate User Statistics Report</b>	User Statistics Report	To generate a user statistics report.	A user statistics report is successfully generated at year-end.	<u><b>Audit_Log</b></u> <ul style="list-style-type: none"> <li>LogInTime</li> <li>LogInDate</li> <li>LogInDuration</li> </ul> <u><b>Person</b></u> <ul style="list-style-type: none"> <li>PersonID</li> </ul> <u><b>User_Type</b></u> <ul style="list-style-type: none"> <li>UserTypeID</li> </ul>	A report that shows relevant user information .

### 7.2.10 Donation Output Description

Use Case	Description	Purpose	When is it Used	Entities and Attributes	Logical Layout
<b>10.1 Add Donor</b>	Successful donor addition notification.	To add a new donor.	Donor information has been successfully added to the system.	<u>Donor</u> <ul style="list-style-type: none"> <li>Name</li> <li>Surname</li> <li>Organisation Name</li> <li>Phone Number</li> <li>Email Address</li> </ul>	A message that informs a donor their details are successfully added to the system.
<b>10.2 Receive Donation</b>	Successful donation notification.	To receive donation from a donor.	Donation information has been successfully added to the system.	<u>Donor</u> <ul style="list-style-type: none"> <li>Email Address</li> </ul> <u>Donation</u> <ul style="list-style-type: none"> <li>ProofOfPayment</li> <li>DonationAmount</li> </ul>	A message that informs a donor that their donation submission is successful.

## 7.3 Conclusion

The above table describe outputs that flow out of the context diagram. The output table described all the output flows of the context diagram, as well as the output medium and who received the information.



## 8. Validation

### 8.1 Introduction

The purpose of the validation is to ensure that the functional specifications detailed above are in line with all the requirements from the project proposal deliverable. It will display the subsystem, the requirement from the subsystem, the use case that pertains to that subsystem, each process in primitive level that pertain to that use case as well as the entities and attributes that are affected by that Use Case.

### 8.2 Validation against Requirements

#### 8.2.1 User Subsystem

Subsystem	Requirement	Use Case	Primitive Level	Entities and Attributes
1. User Subsystem	1.1 Check Forgotten Password	1.1 Check Forgotten Password	1.1.1 Display Login Options 1.1.2 Select 'Forgot Password' 1.1.3 Prompt User for Email Address 1.1.4 Enter Email Address 1.1.5 Verify Email Address 1.1.6 Display Security Question 1.1.7 Answer Security Question 1.1.8 Verify Answer	<b><u>Person</u></b> <ul style="list-style-type: none"> <li>• PersonID</li> <li>• Email</li> <li>• Password</li> </ul> <b><u>SecurityAnswer</u></b> <ul style="list-style-type: none"> <li>• AnswerID</li> <li>• Answer</li> </ul>
1. User Subsystem	1.2 Change Password	1.2 Change Password	1.2.1 Display Main Menu	<b><u>Person</u></b> <ul style="list-style-type: none"> <li>• PersonID</li> <li>• Email</li> </ul>

Subsystem	Requirement	Use Case	Primitive Level	Entities and Attributes
			1.2.2 Select My Profile 1.2.3 Display Profile 1.2.4 Select Change Password Option 1.2.5 Prompt user to enter required information 1.2.6 Enter Required Information 1.2.7 Verify Information 1.2.8 Update Password	<ul style="list-style-type: none"> <li>Password</li> </ul>
1. User Subsystem	1.3 Login	1.3 Login	1.3.1 Display Login Section 1.3.2 Enter Login Details 1.3.3 Verify Login Details 1.3.4 Check user type 1.3.5 Display Main Menu	<b><u>Person</u></b> <ul style="list-style-type: none"> <li>PersonID</li> <li>Email</li> <li>Password</li> </ul> <b><u>UserType</u></b> <ul style="list-style-type: none"> <li>UserTypeID</li> </ul> <b><u>AuditLog</u></b> <ul style="list-style-type: none"> <li>AuditID</li> <li>LogInTime</li> <li>LogInDate</li> <li>LogInDuration</li> </ul>
1. User Subsystem		AUC 1 Register User	AUC 1.1 Display Login Section AUC 1.2 Select Register Option AUC 1.3 Prompt for Login details	<b><u>Person</u></b> <ul style="list-style-type: none"> <li>PersonID</li> <li>Email</li> <li>Password</li> </ul> <b><u>SecurityAnswer</u></b> <ul style="list-style-type: none"> <li>AnswerID</li> <li>Answer</li> </ul> <b><u>UserType</u></b> <ul style="list-style-type: none"> <li>UserTypeID</li> </ul>

Subsystem	Requirement	Use Case	Primitive Level	Entities and Attributes
			AUC 1.4 Enter Login details AUC 1.5 Verify Login details AUC 1.6 Add login details AUC 1.7 Prompt for user type	

## 8.2.2 Volunteer Subsystem

Subsystem	Requirement	Use Case	Primitive Level	Entities and Attributes
2. Volunteer Subsystem	2.1 Register Volunteer	2.1 Register Volunteer	2.1.1 Select Volunteer Type 2.1.2 Prompt for unique code 2.1.3 Enter unique code 2.1.4 Verify unique code 2.1.5 Mark code as unused 2.1.6 Prompt for personal details 2.1.7 Enter personal details 2.1.8 Verify personal details 2.1.9 Add personal details	<b><u>UniqueCode</u></b> <ul style="list-style-type: none"> <li>• CodeID</li> <li>• UniqueCode</li> <li>• Status</li> </ul> <b><u>Person</u></b> <ul style="list-style-type: none"> <li>• PersonID</li> <li>• Name</li> <li>• Surname</li> <li>• DoB</li> <li>• Phone</li> <li>• Race</li> </ul> <b><u>Volunteer</u></b> <ul style="list-style-type: none"> <li>• VolunteerID</li> <li>• VolunteerTypeID</li> </ul> <b><u>Language</u></b> <ul style="list-style-type: none"> <li>• LanguageID</li> <li>• Language</li> </ul> <b><u>PersonLanguage</u></b> <ul style="list-style-type: none"> <li>• PersonID</li> <li>• LanguageID</li> </ul>
2. Volunteer Subsystem	2.2 Search Volunteer	2.2 Search Volunteer	2.2.1 Display main menu 2.2.2 Select volunteer option 2.2.3 Display list of volunteers 2.2.4 Enter search criteria 2.2.5 Display specific volunteer list	<b><u>Person</u></b> <ul style="list-style-type: none"> <li>• Name</li> <li>• Surname</li> <li>• DoB</li> <li>• Email</li> <li>• Phone</li> </ul>



Subsystem	Requirement	Use Case	Primitive Level	Entities and Attributes
			2.2.6 Select relevant volunteer 2.2.7 Display volunteer details 2.2.8 Select option to return to main menu 2.2.9 Display main menu	
2. Volunteer Subsystem	2.3 Update Volunteer	2.3 Update Volunteer	2.3.1 Display main menu 2.3.2 Select 'My Profile' option 2.3.3 Display profile 2.3.4 Change volunteer details 2.3.5 Verify details 2.3.6 Update details	<b><u>Person</u></b> <ul style="list-style-type: none"> <li>• PersonID</li> <li>• Name</li> <li>• Surname</li> <li>• DoB</li> <li>• Email</li> <li>• Phone</li> <li>• Race</li> </ul> <b><u>Volunteer</u></b> <ul style="list-style-type: none"> <li>• VolunteerID</li> <li>• VolunteerTypeID</li> </ul> <b><u>Language</u></b> <ul style="list-style-type: none"> <li>• LanguageID</li> <li>• Language</li> </ul> <b><u>PersonLanguage</u></b> <ul style="list-style-type: none"> <li>• PersonID</li> <li>• LanguageID</li> </ul>
2. Volunteer Subsystem	2.4 Delete Volunteer	2.4 Delete Volunteer	2.4.1 Display main menu 2.4.2 Select volunteer option (Invoke UC 2.2 Search Volunteer) 2.4.3 Display volunteer details 2.4.4 Select delete option	<b><u>Person</u></b> <ul style="list-style-type: none"> <li>• PersonID</li> <li>• Name</li> <li>• Surname</li> <li>• DoB</li> <li>• Email</li> <li>• Phone</li> <li>• Race</li> </ul> <b><u>Volunteer</u></b> <ul style="list-style-type: none"> <li>• VolunteerID</li> <li>• VolunteerTypeID</li> </ul> <b><u>Language</u></b> <ul style="list-style-type: none"> <li>• LanguageID</li> <li>• Language</li> </ul>

Subsystem	Requirement	Use Case	Primitive Level	Entities and Attributes
			2.4.5 Display warning notification 2.4.6 Confirm deletion 2.4.7 Delete volunteer	<u><b>PersonLanguage</b></u> <ul style="list-style-type: none"> <li>PersonID</li> <li>LanguageID</li> </ul>
2. Volunteer Subsystem	2.5 Add Volunteer Type	2.5 Add Volunteer Type	2.5.1 Display main menu 2.5.2 Select volunteer option 2.5.3 Display volunteer section 2.5.4 Select volunteer type option 2.5.5 Display volunteer type section 2.5.6 Select add option 2.5.7 Prompt for volunteer type details 2.5.8 Enter volunteer type details 2.5.9 Verify volunteer type details	<u><b>VolunteerType</b></u> <ul style="list-style-type: none"> <li>VolunteerTypeID</li> <li>VolunteerTypeName</li> <li>Description</li> </ul>

Subsystem	Requirement	Use Case	Primitive Level	Entities and Attributes
			2.5.10 Add volunteer type	
2. Volunteer Subsystem	2.6 Search Volunteer Type	2.6 Search Volunteer Type	2.6.1 Display main menu 2.6.2 Select volunteer option 2.6.3 Display volunteer section 2.6.4 Select volunteer type option 2.6.5 Display volunteer type list 2.6.6 Enter search criteria 2.6.7 Display specific volunteer type list 2.6.8 Select relevant volunteer type 2.6.9 Display volunteer type details 2.6.10 Select option to return to main menu 2.6.11 Display main menu	<b><u>VolunteerType</u></b> <ul style="list-style-type: none"> <li>• VolunteerTypeID</li> <li>• VolunteerTypeName</li> <li>• Description</li> </ul>
2. Volunteer Subsystem	2.7 Update Volunteer Type	2.7 Update Volunteer Type	2.7.1 Display main menu	<b><u>VolunteerType</u></b> <ul style="list-style-type: none"> <li>• VolunteerTypeID</li> <li>• VolunteerTypeName</li> <li>• Description Description</li> </ul>

Subsystem	Requirement	Use Case	Primitive Level	Entities and Attributes
			2.7.2 Select volunteer option (Invoke UC 2.6 Search Volunteer Type) 2.7.3 Display volunteer type details 2.7.4 Change details 2.7.5 Verify details 2.7.6 Update volunteer type	
2. Volunteer Subsystem	2.8 Delete Volunteer Type	2.8 Delete Volunteer Type	2.8.1 Display main menu 2.8.2 Select volunteer option (Invoke UC 2.6 Search Volunteer Type) 2.8.3 Display volunteer type details 2.8.4 Select delete option 2.8.5 Display warning notification 2.8.6 Confirm deletion 2.8.7 Delete volunteer type	<b><u>VolunteerType</u></b> <ul style="list-style-type: none"> <li>• VolunteerTypeID</li> <li>• VolunteerTypeName</li> <li>• Description</li> </ul>

Subsystem	Requirement	Use Case	Primitive Level	Entities and Attributes
2. Volunteer Subsystem	2.9 Generate Unique Code	2.9 Generate Unique Code	2.9.1 Display main menu 2.9.2 Select volunteer option 2.9.3 Display volunteer section 2.9.4 Select generate unique code 2.9.5 Generate unique code 2.9.6 Add unique code 2.9.7 Display unique code 2.9.8 Select option to email unique code 2.9.9 Prompt for email address 2.9.10 Enter email address 2.9.11 Send unique code	<u><b>UniqueCode</b></u> <ul style="list-style-type: none"> <li>• CodeID</li> <li>• Code</li> <li>• Status</li> </ul>

## 8.2.3 Student Subsystem

Subsystem	Requirement	Use Case	Primitive Level	Entities and Attributes
3. Student Subsystem	3.1 Register Student	3.1 Register Student	3.1.1 Select student type 3.1.2 Prompt for personal details 3.1.3 Enter personal details 3.1.4 Verify personal details 3.1.5 Add personal details	<b><u>Person</u></b> <ul style="list-style-type: none"> <li>PersonID</li> <li>Name</li> <li>Surname</li> <li>DoB</li> <li>Phone</li> <li>Race</li> </ul> <b><u>Student</u></b> <ul style="list-style-type: none"> <li>StudentID</li> <li>StudentNumber</li> <li>StudentTypeID</li> <li>DegreeID</li> <li>YearOfStudy</li> </ul> <b><u>StudentType</u></b> <ul style="list-style-type: none"> <li>StudentTypeID</li> <li>Name</li> <li>Description</li> </ul> <b><u>Language</u></b> <ul style="list-style-type: none"> <li>LanguageID</li> <li>Language</li> </ul> <b><u>PersonLanguage</u></b> <ul style="list-style-type: none"> <li>PersonID</li> <li>LanguageID</li> </ul> <b><u>CurrentDegree</u></b> <ul style="list-style-type: none"> <li>DegreeID</li> <li>Name</li> </ul>
3. Student Subsystem	3.3 Search Student	3.3 Search Student	3.2.1 Display main menu 3.2.2 Select student option 3.2.3 Display student list 3.2.4 Enter search criteria 3.2.5 Display specific student list 3.2.6 Select relevant student	<b><u>Person</u></b> <ul style="list-style-type: none"> <li>PersonID</li> <li>Name</li> <li>Surname</li> <li>DoB</li> <li>Phone</li> <li>Race</li> </ul> <b><u>Student</u></b> <ul style="list-style-type: none"> <li>StudentID</li> <li>StudentNumber</li> <li>StudentTypeID</li> <li>DegreeID</li> <li>YearOfStudy</li> </ul> <b><u>StudentType</u></b> <ul style="list-style-type: none"> <li>StudentTypeID</li> <li>Name</li> <li>Description</li> </ul>

Subsystem	Requirement	Use Case	Primitive Level	Entities and Attributes
			3.2.7 Display student details	<u>Language</u> <ul style="list-style-type: none"> <li>LanguageID</li> <li>Language</li> </ul> <u>PersonLanguage</u> <ul style="list-style-type: none"> <li>PersonID</li> <li>LanguageID</li> </ul> <u>CurrentDegree</u> <ul style="list-style-type: none"> <li>DegreeID</li> <li>Name</li> </ul>
3. Student Subsystem	3.3 Update Student	3.3 Update Student	3.3.1 Display main menu 3.3.2 Select 'My Profile' option 3.3.3 Display profile 3.3.4 Change details 3.3.5 Verify details 3.3.6 Update student	<u>Person</u> <ul style="list-style-type: none"> <li>PersonID</li> <li>Name</li> <li>Surname</li> <li>DoB</li> <li>Phone</li> <li>Race</li> </ul> <u>Student</u> <ul style="list-style-type: none"> <li>StudentID</li> <li>StudentNumber</li> <li>StudentTypeID</li> <li>DegreeID</li> <li>YearOfStudy</li> </ul> <u>StudentType</u> <ul style="list-style-type: none"> <li>StudentTypeID</li> <li>Name</li> <li>Description</li> </ul> <u>Language</u> <ul style="list-style-type: none"> <li>LanguageID</li> <li>Language</li> </ul> <u>PersonLanguage</u> <ul style="list-style-type: none"> <li>PersonID</li> <li>LanguageID</li> </ul> <u>CurrentDegree</u> <ul style="list-style-type: none"> <li>DegreeID</li> <li>Name</li> </ul>
3. Student Subsystem	3.4 Delete Student	3.4 Delete Student	3.4.1 Display main menu 3.4.2 Select student option (Invoke UC 3.2 Search Student)	<u>Person</u> <ul style="list-style-type: none"> <li>PersonID</li> <li>Name</li> <li>Surname</li> <li>DoB</li> <li>Phone</li> <li>Race</li> </ul> <u>Student</u> <ul style="list-style-type: none"> <li>StudentID</li> </ul>

Subsystem	Requirement	Use Case	Primitive Level	Entities and Attributes
			3.4.3 Display student details 3.4.4 Select delete option 3.4.5 Display warning notification 3.4.6 Confirm deletion 3.4.7 Delete student	<ul style="list-style-type: none"> <li>• StudentNumber</li> <li>• StudentTypeID</li> <li>• DegreeID</li> <li>• YearOfStudy</li> </ul> <b><u>StudentType</u></b> <ul style="list-style-type: none"> <li>• StudentTypeID</li> <li>• Name</li> <li>• Description</li> </ul> <b><u>Language</u></b> <ul style="list-style-type: none"> <li>• LanguageID</li> <li>• Language</li> </ul> <b><u>PersonLanguage</u></b> <ul style="list-style-type: none"> <li>• PersonID</li> <li>• LanguageID</li> </ul> <b><u>CurrentDegree</u></b> <ul style="list-style-type: none"> <li>• DegreeID</li> <li>• Name</li> </ul>
3. Student Subsystem	3.5 Generate Graduate List	3.5 Generate Graduate List	3.5.1 Display main menu 3.5.2 Select student option 3.5.3 Display student section 3.5.4 Select generate graduate list option 3.5.5 Add graduates 3.5.6 Display graduate list	<b><u>Student:</u></b> <ul style="list-style-type: none"> <li>• StudentID</li> </ul> <b><u>Graduate:</u></b> <ul style="list-style-type: none"> <li>• GraduationID</li> <li>• StudentID</li> <li>• GradYear</li> </ul> <b><u>Progress</u></b> <ul style="list-style-type: none"> <li>• ProgressID</li> </ul>
3. Student Subsystem	3.6 Add Student Type	3.6 Add Student Type	3.6.1 Display main menu 3.6.1 Display main menu 3.6.2 Select student option 3.6.3 Display student section	<b><u>StudentType</u></b> <ul style="list-style-type: none"> <li>• StudentTypeID</li> <li>• StudentTypeName</li> <li>• Description</li> </ul>



Subsystem	Requirement	Use Case	Primitive Level	Entities and Attributes
			3.6.4 Select student type option 3.6.5 Display student type section 3.6.6 Select option to add student type 3.6.7 Prompt for student type details 3.6.8 Enter student type details 3.6.9 Verify student type details 3.6.10 Add student type	
3. Student Subsystem	3.7 Search Student Type	3.7 Search Student Type	3.7.1 Display main menu 3.7.2 Select student option 3.6.3 Display student section 3.7.4 Select student type option 3.7.5 Display student type list 3.7.6 Enter search criteria 3.7.7 Display specific student type list 3.7.8 Select relevant student type	<b><u>StudentType</u></b> <ul style="list-style-type: none"> <li>• StudentTypeID</li> <li>• StudentTypeName</li> <li>• Description</li> </ul>

Subsystem	Requirement	Use Case	Primitive Level	Entities and Attributes
			3.7.9 Display student type details	
3. Student Subsystem	3.8 Update Student Type	3.8 Update Student Type	3.8.1 Display main menu 3.8.2 Select student option (Invoke UC 2.7 Search student type) 2.8.3 Display student type details 2.8.4 Change student type details 2.8.5 Verify student type details 2.8.6 Update student type details	<b><u>StudentType</u></b> <ul style="list-style-type: none"> <li>• StudentTypeID</li> <li>• StudentTypeName</li> <li>• Description</li> </ul>
3. Student Subsystem	3.9 Delete Student Type	3.9 Delete Student Type	3.9.1 Display main menu 3.9.2 Select student option (Invoke UC 2.7 Search student type) 2.9.3 Display student type details 2.9.4 Select delete option 2.9.5 Display warning notification 2.9.6 Confirm deletion	<b><u>StudentType</u></b> <ul style="list-style-type: none"> <li>• StudentTypeID</li> <li>• StudentTypeName</li> <li>• Description</li> </ul>

Subsystem	Requirement	Use Case	Primitive Level	Entities and Attributes
			2.9.7 Delete student type	

## 8.2.4 Event Subsystem

Subsystem	Requirement	Use Case	Primitive Level	Entities and Attributes
4. Event Subsystem	4.1 Add New Venue	4.1 Add New Venue	4.1.1 Display Main Menu 4.1.2 Select 'Events' Option 4.1.3 Display list of events 4.1.4 Select 'Venues' Option 4.1.5 Display list of existing venues 4.1.6 Select 'Add New Venue' option 4.1.7 Enter Venue Information 4.1.8 Validate Venue Information 4.1.9 Update Venue List	<b><u>Venue:</u></b> <ul style="list-style-type: none"> <li>VenueID</li> <li>Name</li> <li>Capacity</li> <li>Accessibility</li> </ul> <b><u>VenueType:</u></b> <ul style="list-style-type: none"> <li>VenueTypeID</li> <li>Name</li> </ul> <b><u>Address:</u></b> <ul style="list-style-type: none"> <li>AddressID</li> <li>StreetName</li> <li>StreetNumber</li> <li>Suburb</li> <li>City</li> <li>Province</li> </ul>
4. Event Subsystem	4.2 Search Venue	4.2 Search Venue	4.2.1 Display Main Menu 4.2.2 Select 'Events' Option 4.2.3 Display list of events 4.2.4 Select 'Venues' Option 4.2.5 Display list of existing venues	<b><u>Venue:</u></b> <ul style="list-style-type: none"> <li>VenueID</li> <li>Name</li> <li>Capacity</li> <li>Accessibility</li> </ul> <b><u>VenueType:</u></b> <ul style="list-style-type: none"> <li>VenueTypeID</li> <li>Name</li> </ul> <b><u>Address:</u></b> <ul style="list-style-type: none"> <li>AddressID</li> <li>StreetName</li> <li>StreetNumber</li> <li>Suburb</li> <li>City</li> <li>Province</li> </ul>

Subsystem	Requirement	Use Case	Primitive Level	Entities and Attributes
			4.2.6 Search for Venue 4.2.7 Select Venue Option 4.2.8 Display Venue Information	
4. Event Subsystem	4.3 Update Venue Information	4.3 Update Venue Information	4.3.1 Display Main Menu 4.3.2 Select 'Events' Option 4.3.3 Display list of events 4.3.4 Select 'Venues' Option 4.3.5 Display list of existing venues 4.3.6 Select Update Venue Option 4.3.7 Update Venue Information 4.3.8 Confirm Update Information 4.3.9 Validate Updated Information 4.3.10 Update Venue Information	<b>Venue:</b> <ul style="list-style-type: none"> <li>VenueID</li> <li>Name</li> <li>Capacity</li> <li>Accessibility</li> </ul> <b>VenueType:</b> <ul style="list-style-type: none"> <li>VenueTypeID</li> <li>Name</li> </ul> <b>Address:</b> <ul style="list-style-type: none"> <li>AddressID</li> <li>StreetName</li> <li>StreetNumber</li> <li>Suburb</li> <li>City</li> <li>Province</li> </ul>
4. Event Subsystem	4.4 Delete Venue	4.4 Delete Venue	4.4.1 Display Main Menu 4.4.2 Select 'Events' Option	<b>Venue:</b> <ul style="list-style-type: none"> <li>VenueID</li> <li>Name</li> <li>Capacity</li> <li>Accessibility</li> </ul> <b>VenueType:</b>

Subsystem	Requirement	Use Case	Primitive Level	Entities and Attributes
			4.4.3 Display list of events 4.4.4 Select 'Venues' Option 4.4.5 Display list of existing venues 4.4.6 Select Delete venue option 4.4.7 Display Warning Message 4.4.8 Confirm Venue Deletion 4.4.9 Update Venue List	<ul style="list-style-type: none"> <li>VenueTypeID</li> <li>Name</li> </ul> <b><u>Address:</u></b> <ul style="list-style-type: none"> <li>AddressID</li> <li>StreetName</li> <li>StreetNumber</li> <li>Suburb</li> <li>City</li> <li>Province</li> </ul>
4. Event Subsystem	4.5. Create Event	4.5 Create Event	4.5.1 Display Main Menu 4.5.2 Select 'Events' Option 4.5.3 Display list of events 4.5.4 Select 'Create Event' Option 4.5.5 Select Event Type 4.5.6 Provide Event Information 4.5.7 Validate Event Information 4.5.8 Update Event List	<b><u>Events:</u></b> <ul style="list-style-type: none"> <li>EventID</li> <li>EventTypeID</li> <li>Name</li> <li>Summary</li> <li>Details</li> <li>Time</li> <li>Date</li> <li>VenueID</li> </ul> <b><u>Function:</u></b> <ul style="list-style-type: none"> <li>FunctionID</li> <li>GuestSpeakerID</li> </ul> <b><u>Lecture:</u></b> <ul style="list-style-type: none"> <li>LectureID</li> </ul> <b><u>CommunityOutreach:</u></b> <ul style="list-style-type: none"> <li>ComID</li> </ul> <b><u>Residence:</u></b> <ul style="list-style-type: none"> <li>ResidenceID</li> <li>Name</li> </ul> <b><u>Content:</u></b> <ul style="list-style-type: none"> <li>ContentID</li> <li>Name</li> <li>Author</li> <li>DatePublished</li> </ul>

Subsystem	Requirement	Use Case	Primitive Level	Entities and Attributes
			4.5.9 Confirm New Event Creation 4.5.10 Upload Event Information 4.5.11 Notify All interest parties.	<ul style="list-style-type: none"> <li>Link</li> </ul>
4. Event Subsystem	4.6. Search Event	4.6 Search Event	4.6.1 Display Main Menu 4.6.2 Select 'Events' Option 4.6.3 Display list of events 4.6.4 Search for Event 4.6.5 Select Event Option 4.6.6 Display Event Information	<b>Events:</b> <ul style="list-style-type: none"> <li>EventID</li> <li>EventTypeID</li> <li>Name</li> <li>Summary</li> <li>Details</li> <li>Time</li> <li>Date</li> <li>VenueID</li> </ul> <b>Function:</b> <ul style="list-style-type: none"> <li>FunctionID</li> <li>GuestSpeakerID</li> </ul> <b>Lecture:</b> <ul style="list-style-type: none"> <li>LectureID</li> </ul> <b>CommunityOutreach:</b> <ul style="list-style-type: none"> <li>ComID</li> </ul> <b>Residence:</b> <ul style="list-style-type: none"> <li>ResidenceID</li> <li>Name</li> </ul> <b>Content:</b> <ul style="list-style-type: none"> <li>ContentID</li> <li>Name</li> <li>Author</li> <li>DatePublished</li> <li>Link</li> </ul> <b>TRWLATimeline</b> <ul style="list-style-type: none"> <li>TimelineID</li> </ul>
4. Event Subsystem	4.7 RSVP to an Event	4.7 RSVP to an Event	4.7.1 Display Main Menu 4.7.2 Search for an Event	<b>Events:</b> <ul style="list-style-type: none"> <li>EventID</li> <li>EventTypeID</li> <li>Name</li> <li>Summary</li> <li>Details</li> <li>Time</li> </ul>

Subsystem	Requirement	Use Case	Primitive Level	Entities and Attributes
			4.7.3 Select 'RSVP to an Event' option 4.7.4 RSVP Warning Message 4.7.5 Confirm Event Attendance 4.7.6 Update RSVP	<ul style="list-style-type: none"> <li>Date</li> <li>VenueID</li> </ul> <b>RSVP Event:</b> <ul style="list-style-type: none"> <li>RSVP_EventID</li> <li>EventID</li> <li>PersonID</li> </ul> <b>Person:</b> <ul style="list-style-type: none"> <li>PersonID</li> <li>Name</li> <li>Surname</li> <li>Email</li> <li>Phone</li> </ul>
4. Event Subsystem	4.8 View TRWLA Timeline	4.8 View TRWLA Timeline	4.8.1 Display Main Menu 4.8.2 Select 'TRWLA Timeline' option 4.8.3 Display TRWLA Timeline	<b>Events:</b> <ul style="list-style-type: none"> <li>EventID</li> <li>EventTypeID</li> <li>Name</li> <li>Summary</li> <li>Details</li> <li>Time</li> <li>Date</li> <li>VenueID</li> </ul> <b>Function:</b> <ul style="list-style-type: none"> <li>FunctionID</li> <li>GuestSpeakerID</li> </ul> <b>Lecture:</b> <ul style="list-style-type: none"> <li>LectureID</li> <li>LectureContent</li> </ul> <b>CommunityOutreach:</b> <ul style="list-style-type: none"> <li>ComID</li> <li>EventID</li> </ul> <b>Residence:</b> <ul style="list-style-type: none"> <li>ResidenceID</li> <li>ResidenceName</li> </ul>
4. Event Subsystem	4.9 Update Event Information	4.9 Update Event Information	4.9.1 Display Main Menu 4.9.2 Select 'Events' Option 4.9.3 Display list of events 4.9.4 Search Event	<b>Events:</b> <ul style="list-style-type: none"> <li>EventID</li> <li>EventTypeID</li> <li>Name</li> <li>Summary</li> <li>Details</li> <li>Time</li> <li>Date</li> <li>VenueID</li> </ul> <b>Function:</b> <ul style="list-style-type: none"> <li>FunctionID</li> </ul>



Subsystem	Requirement	Use Case	Primitive Level	Entities and Attributes
			4.9.5 Select 'Update Event Information' option 4.9.6 Update Event Information 4.9.7 Confirm Event Update 4.9.8 Validate Event Update Information 4.9.9 Update Event List 4.9.10 Notify Event RSVP'd Users	<ul style="list-style-type: none"> <li>• GuestSpeakerID</li> </ul> <b>Lecture:</b> <ul style="list-style-type: none"> <li>• LectureID</li> </ul> <b>CommunityOutreach:</b> <ul style="list-style-type: none"> <li>• ComID</li> </ul> <b>Residence:</b> <ul style="list-style-type: none"> <li>• ResidenceID</li> <li>• Name</li> </ul> <b>Content:</b> <ul style="list-style-type: none"> <li>• ContentID</li> <li>• Name</li> <li>• Author</li> <li>• DatePublished</li> <li>• Link</li> </ul>
4. Event Subsystem	4.10 Cancel Event	4.10 Cancel Event	4.10.1 Display Main Menu 4.10.2 Select 'Events' Option 4.10.3 Display list of events 4.10.4 Search Event 4.10.5 Select 'Cancel Event' option 4.10.6 Warning Message 4.10.7 Confirm Event Cancellation 4.10.8 Update Event List 4.10.9 Notify All RSVP'd Members	<b>Events:</b> <ul style="list-style-type: none"> <li>• EventID</li> <li>• EventTypeID</li> <li>• Name</li> <li>• Summary</li> <li>• Details</li> <li>• Time</li> <li>• Date</li> <li>• VenueID</li> </ul> <b>Function:</b> <ul style="list-style-type: none"> <li>• FunctionID</li> <li>• GuestSpeakerID</li> </ul> <b>Lecture:</b> <ul style="list-style-type: none"> <li>• LectureID</li> </ul> <b>CommunityOutreach:</b> <ul style="list-style-type: none"> <li>• ComID</li> </ul> <b>Residence:</b> <ul style="list-style-type: none"> <li>• ResidenceID</li> <li>• Name</li> </ul> <b>Content:</b> <ul style="list-style-type: none"> <li>• ContentID</li> <li>• Name</li> <li>• Author</li> <li>• DatePublished</li> <li>• Link</li> </ul>

Subsystem	Requirement	Use Case	Primitive Level	Entities and Attributes
4. Event Subsystem	4.11 Log Event Attendance	4.11 Log Event Attendance	4.11.1 Display Main Menu 4.11.2 Select 'Log Attendance' option 4.11.3 Display List of Current Events 4.11.4 Select Event Option 4.11.5 Display Event Details 4.11.6 Select 'Log Current Event Attendance' Option 4.11.7 Display List of users who have RSVPd 4.11.8 Select 'Add New' Option 4.11.9 Display list of existing Students 4.11.10 Search Student 4.11.11 Display List of Students 4.11.12 Select Student 4.11.13 Display Student Information 4.11.14 Select 'RSVP Student' Option	<b><u>Person:</u></b> <ul style="list-style-type: none"> <li>PersonID</li> <li>Name</li> <li>Surname</li> </ul> <b><u>Student:</u></b> <ul style="list-style-type: none"> <li>StudentID</li> <li>StudentNumber</li> </ul> <b><u>RSVP Event:</u></b> <ul style="list-style-type: none"> <li>RSVP_EventID</li> <li>EventID</li> <li>PersonID</li> </ul> <b><u>Attendance:</u></b> <ul style="list-style-type: none"> <li>Attendance_ID</li> <li>EventID</li> <li>PersonID</li> </ul> <b><u>Progress:</u></b> <ul style="list-style-type: none"> <li>ProgressID</li> <li>ProgressCount</li> </ul>

Subsystem	Requirement	Use Case	Primitive Level	Entities and Attributes
			4.11.15 Confirm Student Attendance	
4. Event Subsystem	4.12 Send Notification	4.12 Send Notification	4.12.1 Display Main Menu 4.12.2 Select 'Send Notification' Option 4.12.3 Display List of Persons 4.12.4 Search Recipients 4.12.5 Display List of Matching Persons 4.12.6 Select 'Confirm Recipients' Option 4.12.7 Display Editable Notification Space 4.12.8 Confirm Notification 4.12.9 Send Notification	<b><u>Person:</u></b> <ul style="list-style-type: none"> <li>• PersonID</li> <li>• Name</li> <li>• Surname</li> <li>• Email</li> <li>• Phone</li> </ul>

## 8.2.5 Function Subsystem

Subsystem	Requirement	Use Case	Primitive Level	Entities and Attributes
5. Function Subsystem	5.1. Register Guest Speaker	5.1 Register Guest Speaker	5.1.1 Display Main Menu  5.1.2 Select Guest Speaker Option  5.1.3 Display Guest Speaker List  5.1.4 Select Register Guest Speaker Option  5.1.5 Enter Guest Speaker Info and Confirm  5.1.6 Validate Guest Speaker Details  5.1.7 Update Guest Speaker List	<b><u>GuestSpeaker</u></b> <ul style="list-style-type: none"> <li>• GuestSpeakerID</li> <li>• Speaker_Name</li> <li>• Speaker_Surname</li> <li>• Gender</li> <li>• DOB</li> <li>• Email</li> <li>• Phone</li> <li>• Hourly_Rate</li> <li>• Theme</li> </ul> <b><u>SpeakerType</u></b> <ul style="list-style-type: none"> <li>• SpeakerTypeID</li> <li>• TypeName</li> <li>• TypeDescription</li> </ul>
5. Function Subsystem	5.2. View Guest Speaker	5.2 View Guest Speaker	5.2.1 Display Main Menu  5.2.2 Select Guest Speaker Option  5.2.3 Display Guest Speaker List  5.2.4 Search Guest Speaker	<b><u>GuestSpeaker</u></b> <ul style="list-style-type: none"> <li>• GuestSpeakerID</li> <li>• Speaker_Name</li> <li>• Speaker_Surname</li> <li>• Gender</li> <li>• DOB</li> <li>• Email</li> <li>• Phone</li> <li>• Hourly_Rate</li> <li>• Theme</li> </ul>

Subsystem	Requirement	Use Case	Primitive Level	Entities and Attributes
			5.2.5 Display Search Results  5.2.6 Select Guest Speaker  5.2.7 Display Guest Speaker Info	
5. Function Subsystem	5.3. Update Guest Speaker	5.3 Update Guest Speaker	5.3.1 Display Main Menu  5.3.2 Select Guest Speaker Option  5.3.3 Display Guest Speaker List  5.3.4 Search and Select Guest Speaker  5.3.5 Display Guest Speaker Info  5.3.6 Update Guest Speaker Details  5.3.7 Validate Guest Speaker Info  5.3.8 Display Warning Message  5.3.9 Select Confirm Changes	<b><u>GuestSpeaker</u></b> <ul style="list-style-type: none"> <li>• GuestSpeakerID</li> <li>• Speaker_Name</li> <li>• Speaker_Surname</li> <li>• Gender</li> <li>• DOB</li> <li>• Email</li> <li>• Phone</li> <li>• Hourly_Rate</li> <li>• Theme</li> </ul> <b><u>SpeakerType</u></b> <ul style="list-style-type: none"> <li>• SpeakerTypeID</li> <li>• TypeName</li> <li>• TypeDescription</li> </ul>

Subsystem	Requirement	Use Case	Primitive Level	Entities and Attributes
			5.3.10 Update Speaker Details	
5. Function Subsystem	5.4. Delete Guest Speaker	5.4 Delete Guest Speaker	5.4.1 Display Main Menu  5.4.2 Select Guest Speaker Option  5.4.3 Display Guest Speaker List  5.4.4 Search and Select Guest Speaker  5.4.5 Display Guest Speaker Info  5.4.6 Select Delete Guest Speaker  5.4.7 Display Warning Message  5.4.7 Select Confirm Changes  5.4.8 Delete Guest Speaker	<b><u>GuestSpeaker</u></b> <ul style="list-style-type: none"> <li>• GuestSpeakerID</li> <li>• Speaker_Name</li> <li>• Speaker_Surname</li> <li>• Gender</li> <li>• DOB</li> <li>• Email</li> <li>• Phone</li> <li>• Hourly_Rate</li> <li>• Theme</li> </ul>
5. Function Subsystem	5.5. Invite Guest Speaker	5.5 Invite Guest Speaker	5.5.1 Display Guest Speaker List  5.5.2 Search and Select Guest Speaker	<b><u>GuestSpeaker</u></b> <ul style="list-style-type: none"> <li>• GuestSpeakerID</li> <li>• Speaker_Name</li> <li>• Speaker_Surname</li> <li>• Gender</li> <li>• DOB</li> <li>• Email</li> <li>• Phone</li> </ul>

Subsystem	Requirement	Use Case	Primitive Level	Entities and Attributes
			5.5.3 Display Guest Speaker Info  5.5.4 Select Invite Guest Speaker Option  5.5.5 Display Message to Guest Speaker  5.5.6 Edit Message Info  5.5.7 Send Message  5.5.8 Open Pending Ticket	<ul style="list-style-type: none"> <li>Hourly_Rate</li> <li>Theme</li> </ul> <b>FunctionSpeaker:</b> <ul style="list-style-type: none"> <li>FunctionSpeakerID</li> <li>GuestSpeakerID</li> <li>FunctionID</li> <li>Name</li> <li>Date</li> <li>Time</li> <li>Status</li> </ul> <b>Events:</b> <ul style="list-style-type: none"> <li>EventID</li> <li>EventTypeID</li> <li>Name</li> <li>Summary</li> <li>Details</li> <li>Time</li> <li>Date</li> <li>VenueID</li> </ul>
5. Function Subsystem	5.6. Confirm Guest Speaker	5.6 Confirm Guest Speaker	5.6.1 Display Push Notification  5.6.2 Update Event Information  5.6.3 Display Guest Speaker Info  5.6.4 Select Accepted Option  5.6.5 Display Confirmation Message	<b>GuestSpeaker</b> <ul style="list-style-type: none"> <li>GuestSpeakerID</li> <li>Speaker_Name</li> <li>Speaker_Surname</li> <li>Gender</li> <li>DOB</li> <li>Email</li> <li>Phone</li> <li>Hourly_Rate</li> <li>Theme</li> </ul> <b>FunctionSpeaker:</b> <ul style="list-style-type: none"> <li>FunctionSpeakerID</li> <li>GuestSpeakerID</li> <li>FunctionID</li> <li>Name</li> <li>Date</li> <li>Time</li> <li>Status</li> </ul>

## 8.2.6 Community Engagement Subsystem

Subsystem	Requirement	Use Case	Primitive Level	Entities and Attributes
6. Community Engagement Content Subsystem	6.1. Upload Community Engagement Content	6.1. Upload Community Engagement Content	6.1.1 Display Main Menu  6.1.2 Select Events Option  6.1.3 Display List Of Events  6.1.4 Create Community Engagement Event  6.1.5 Display List Of Content  6.1.6 Select Upload Community Engagement Content  6.1.7 Upload Content  6.1.8 Rename Content  6.1.9 Enter Relevant Information  6.1.10 Select Confirm Upload Option  6.1.11 Validate Information	<b><u>CommContent:</u></b> <ul style="list-style-type: none"> <li>• CommContentID</li> <li>• ComID</li> <li>• Theme</li> <li>• Summary</li> <li>• Description</li> <li>• Link</li> <li>• Status</li> </ul> <b><u>Content:</u></b> <ul style="list-style-type: none"> <li>• ContentID</li> <li>• Name</li> <li>• Author</li> <li>• DatePublished</li> <li>• Link</li> </ul>



Subsystem	Requirement	Use Case	Primitive Level	Entities and Attributes
6. Community Engagement Subsystem	6.2. Search Community Engagement Content	6.2 Search Community Engagement Content	6.2.1 Display Main Menu  6.2.2 Select Events Option  6.2.3 Display List Of Events  6.2.4 Search Community Outreach  6.2.5 Display Community Outreaches  6.2.6 Select Required Community Outreach  6.2.7 Display Community Outreach Information  6.2.8 Select Required Content  6.2.9 Display Content Information	<b><u>CommContent:</u></b> <ul style="list-style-type: none"> <li>• CommContentID</li> <li>• ComID</li> <li>• Theme</li> <li>• Summary</li> <li>• Description</li> <li>• Link</li> <li>• Status</li> </ul> <b><u>Content:</u></b> <ul style="list-style-type: none"> <li>• ContentID</li> <li>• Name</li> <li>• Author</li> <li>• DatePublished</li> <li>• Link</li> </ul>
6. Community Engagement Subsystem	6.3. Delete Community Engagement Content	6.3 Delete Community Engagement Content	6.3.1 Display Main Menu  6.3.2 Select Events Option	<b><u>CommContent:</u></b> <ul style="list-style-type: none"> <li>• CommContentID</li> <li>• ComID</li> <li>• Theme</li> <li>• Summary</li> <li>• Description</li> <li>• Link</li> <li>• Status</li> </ul>

Subsystem	Requirement	Use Case	Primitive Level	Entities and Attributes
			6.3.3 Display List Of Events  6.3.3 Search Community Engagement Event  6.3.4 Display Selected Content Information  6.3.5 Select Delete Content Option  6.3.6 Display Warning Message  6.3.7 Select Confirm Deletion  6.3.8 Delete Content	<b><u>Content:</u></b> <ul style="list-style-type: none"> <li>ContentID</li> <li>Name</li> <li>Author</li> <li>DatePublished</li> <li>Link</li> </ul>
6. Community Engagement Subsystem	6.4. Lock Community Engagement Content	6.4. Lock Community Engagement Content	6.4.1 Display Main Menu  6.4.2 Select Events Option  6.4.3 Display List Of Events  6.4.4 Search Community Engagement Event	<b><u>CommContent:</u></b> <ul style="list-style-type: none"> <li>CommContentID</li> <li>ComID</li> <li>Theme</li> <li>Summary</li> <li>Description</li> <li>Link</li> <li>Status</li> </ul> <b><u>Content:</u></b> <ul style="list-style-type: none"> <li>ContentID</li> <li>Name</li> <li>Author</li> <li>DatePublished</li> <li>Link</li> </ul>

Subsystem	Requirement	Use Case	Primitive Level	Entities and Attributes
			6.4.5 Display Selected Content Information  6.4.6 Select Lock Content Option  6.4.7 Display Warning Message  6.4.8 Select Confirm Content Lock  6.4.9 Lock Content Task	
6. Community Engagement Subsystem	6.5. Unlock Community Engagement Content	6.5 Unlock Community Engagement Content	6.5.1 Display Main Menu  6.5.2 Select Events Option  6.5.3 Display List Of Events  6.5.4 Search Community Engagement Event  6.5.5 Display Selected Content Information  6.5.6 Select Unlock Content Option	<b><u>CommContent:</u></b> <ul style="list-style-type: none"> <li>• CommContentID</li> <li>• ComID</li> <li>• Theme</li> <li>• Summary</li> <li>• Description</li> <li>• Link</li> <li>• Status</li> </ul> <b><u>Content:</u></b> <ul style="list-style-type: none"> <li>• ContentID</li> <li>• Name</li> <li>• Author</li> <li>• DatePublished</li> <li>• Link</li> </ul>

Subsystem	Requirement	Use Case	Primitive Level	Entities and Attributes
			6.5.6 Display Warning Message  6.5.7 Select Confirm Content Unlock  6.5.8 Unlock Content	

## 8.2.7 Lecture Content Subsystem

Subsystem	Requirement	Use Case	Primitive Level	Entities and Attributes
7. Lecture Content Subsystem	7.1 Upload Lecture Content	7.1 Upload Lecture Content	7.1.1 Prompt Main Menu 7.1.2 Select events option 7.1.3 Display Events Created 7.1.4 Select Lecture Event 7.1.5 Display Lecture Event 7.1.6 Select Upload Lecture Content 7.1.7 Prompt to Upload Lecture Content 7.1.8 Upload Lecture Content 7.1.9 Display File Uploaded to System 7.1.10 Prompt for Lecture Content Details 7.1.11 Enter Lecture Content Details 7.1.12 Validate Lecture Content Details 7.1.13 Select Okay Option 7.1.14 Display Confirmation 7.1.15 Send Notification of Uploaded Content	<b><u>LectureContent:</u></b> <ul style="list-style-type: none"> <li>LecContID</li> <li>Theme</li> <li>Summary</li> <li>Description</li> <li>Link</li> <li>Status</li> </ul> <b><u>Content:</u></b> <ul style="list-style-type: none"> <li>ContentID</li> <li>Name</li> <li>Author</li> <li>Date_Published</li> <li>Link</li> </ul> <b><u>Lecture:</u></b> <ul style="list-style-type: none"> <li>LectureID</li> </ul>
7. Lecture Content Subsystem	7.2 Search Lecture Content	7.2 Search Lecture Content	7.2.1 Prompt Main Menu with 'Events' Option 7.2.2 Select Event Option 7.2.3 Prompt with List of events.	<b><u>LectureContent:</u></b> <ul style="list-style-type: none"> <li>LecContID</li> <li>Theme</li> <li>Summary</li> <li>Description</li> <li>Link</li> <li>Status</li> </ul>

Subsystem	Requirement	Use Case	Primitive Level	Entities and Attributes
			7.2.4 Search for Lecture 7.2.5 Display List of Lecture Events 7.2.6 Prompt for lecture content Search Details 7.2.7 Enter Lecture Content Details 7.2.8 Display Lecture Content Search Results	<b><u>Content:</u></b> <ul style="list-style-type: none"> <li>ContentID</li> <li>Name</li> <li>Author</li> <li>Date_Published</li> <li>Link</li> </ul> <b><u>Lecture:</u></b> <ul style="list-style-type: none"> <li>LectureID</li> </ul>
7. Lecture Content Subsystem	7.3 Delete Lecture Content	7.3 Delete Lecture Content	7.3.1 Prompt Main Menu With 'Events' Option 7.3.2 Select Events 7.3.3 Display List of Events 7.3.4 Search Lecture Content 7.3.5 Display Lecture Content 7.3.6 Select Lecture Event 7.3.7 Display Lecture Event 7.3.8 Select Delete Option 7.3.9 Confirmation Message of Delete Action 7.3.10 Select Confirm Deletion 7.3.11 Display Confirmation Message of Successful Deletion	<b><u>LectureContent:</u></b> <ul style="list-style-type: none"> <li>LecContID</li> <li>Theme</li> <li>Summary</li> <li>Description</li> <li>Link</li> <li>Status</li> </ul> <b><u>Content:</u></b> <ul style="list-style-type: none"> <li>ContentID</li> <li>Name</li> <li>Author</li> <li>Date_Published</li> <li>Link</li> </ul> <b><u>Lecture:</u></b> <ul style="list-style-type: none"> <li>LectureID</li> </ul>
7. Lecture Content Subsystem	7.4 Lock Lecture Content	7.4 Lock Lecture Content	7.4.1 Prompt Main Menu with 'Events' Option 7.4.2 Select Events Option 7.4.3 Display events	<b><u>LectureContent:</u></b> <ul style="list-style-type: none"> <li>LecContID</li> <li>Theme</li> <li>Summary</li> <li>Description</li> </ul>

Subsystem	Requirement	Use Case	Primitive Level	Entities and Attributes
			7.4.4 Search Lecture Content 7.4.5 Display Lecture Content 7.4.6 Select 'lock content' Option 7.4.7 Prompt for Locking Content 7.4.8 Lock Content 7.4.9 Display Confirmation of Locked Content	<ul style="list-style-type: none"> <li>• Link</li> <li>• Status</li> </ul> <b><u>Content:</u></b> <ul style="list-style-type: none"> <li>• ContentID</li> <li>• Name</li> <li>• Author</li> <li>• Date_Published</li> <li>• Link</li> </ul> <b><u>Lecture:</u></b> <ul style="list-style-type: none"> <li>• LectureID</li> </ul>
7. Lecture Content Subsystem	7.5. Unlock Lecture Content	7.5 Unlock Lecture Content	7.5.1 Prompt Main Menu With 'Events' Option 7.5.2 Select Search Option 7.5.3 Prompt for Lecture Content Search Details 7.5.4 Enter Search Details 7.5.5 Display Lecture Content Search Details 7.5.6 Select 'Unlock content' Option 7.5.7 Prompt for Unlocking Content 7.5.8 Unlock Content 7.5.9 Display Confirmation of Unlocked Content	<b><u>LectureContent:</u></b> <ul style="list-style-type: none"> <li>• LecContID</li> <li>• Theme</li> <li>• Summary</li> <li>• Description</li> <li>• Link</li> <li>• Status</li> </ul> <b><u>Content:</u></b> <ul style="list-style-type: none"> <li>• ContentID</li> <li>• Name</li> <li>• Author</li> <li>• Date_Published</li> <li>• Link</li> </ul> <b><u>Lecture:</u></b> <ul style="list-style-type: none"> <li>• LectureID</li> </ul>

Subsystem	Requirement	Use Case	Primitive Level	Entities and Attributes
7. Lecture Content Subsystem	7.6. Content Review	7.6 Content Review	7.6.1 Prompt Main Menu with 'Events' Option 7.6.2 Select Events Option 7.6.3 Display Events 7.6.4 Search Lecture Content 7.6.5 Display Lecture Content 7.6.6 Select Lecture Content 7.6.7 Prompt to leave a 'Content Review' 7.6.8 Leave Content Review 7.6.9 Verify Rating 7.6.10 Display Confirmation Message	<b><u>Review:</u></b> <ul style="list-style-type: none"> <li>• ReviewID</li> <li>• Comment</li> <li>• Rating</li> </ul> <b><u>LectureContent:</u></b> <ul style="list-style-type: none"> <li>• LecContID</li> <li>• Theme</li> <li>• Summary</li> <li>• Description</li> <li>• Link</li> <li>• Status</li> </ul>



## 8.2.8 Marketing Subsystem

Subsystem	Requirement	Use case	Primitive Level	ERD
8. Marketing Subsystem	8.1 Upload Photo	8.1 Upload Photo	8.1.1 Display main menu 8.1.2 Select 'Gallery' section 8.1.3 Select 'Upload a Photo' section 8.1.4 Select photo for uploading 8.1.5 Confirm photo for uploading 8.1.6 Upload photo	None.
8. Marketing Subsystem	8.2 Post Photo	8.2 Post Photo	8.2.1 Display main menu 8.2.2 Select 'Gallery' Section 8.2.3 Select 'Post a photo' section 8.2.4 Accept photo and post to gallery 8.2.5 Confirm photo post to gallery	None.

Subsystem	Requirement	Use case	Primitive Level	ERD
			8.2.6 Post photo to gallery	
8. Marketing Subsystem	8.3 Delete Photo	8.3 Delete Photo	8.3.1 Display Main Menu 8.3.2 Select 'Gallery' Section 8.3.3 Select 'Delete a Photo' section 8.3.4 Select photo to delete 8.3.5 Confirm photo for deletion 8.3.6 Delete photo	None.

## 8.2.9 Report Subsystem

Subsystem	Requirement	Use case	Primitive Level	Entities and Attributes
9. Report Subsystem	9.1 Generate Class Attendance Report	9.1 Generate Class Attendance Report	9.1.1 Retrieve information 9.1.2 Calculate information using retrieved data 9.1.3 Generate report showing retrieved and calculated information	<u><b>Lecture</b></u> <ul style="list-style-type: none"> <li>LectureID</li> </ul> <u><b>Attendance</b></u> <ul style="list-style-type: none"> <li>AttendanceID</li> </ul> <u><b>Event</b></u> <ul style="list-style-type: none"> <li>Date</li> <li>Name</li> </ul>
9. Report Subsystem	9.2 Generate Function Attendance Report	9.2 Generate Function Attendance Report	9.2.1 Retrieve information 9.2.2 Calculate information using retrieved data 9.2.3 Generate report using retrieved and calculated data	<u><b>Function</b></u> <ul style="list-style-type: none"> <li>FunctionID</li> </ul> <u><b>Guest Speaker</b></u> <ul style="list-style-type: none"> <li>Speaker_Name</li> </ul> <u><b>Attendance</b></u> <ul style="list-style-type: none"> <li>AttendanceID</li> </ul> <u><b>Event</b></u> <ul style="list-style-type: none"> <li>Date</li> <li>Name</li> </ul>
9. Report Subsystem	9.3 Generate Community Engagement Attendance Report	9.3 Generate Community Engagement Attendance Report	9.3.1 Retrieve information 9.3.2 Calculate information using retrieved data	<u><b>Community Outreach</b></u> <ul style="list-style-type: none"> <li>ComID</li> </ul> <u><b>Attendance</b></u> <ul style="list-style-type: none"> <li>AttendanceID</li> </ul> <u><b>Event</b></u> <ul style="list-style-type: none"> <li>Date</li> <li>Name</li> </ul>

Subsystem	Requirement	Use case	Primitive Level	Entities and Attributes
			9.3.3 Generate report using retrieved and calculated information	
9. Report Subsystem	9.4 Generate Demographics Report	9.4 Generate Demographics Report	9.4.1 Retrieve information 9.4.2 Calculate information using retrieved data 9.4.3 Generate report using retrieved and calculated information	<b><u>Person</u></b> <ul style="list-style-type: none"> <li>• DoB</li> <li>• DegreeID</li> <li>• YearofStudy</li> <li>• Race</li> </ul> <b><u>Volunteer</u></b> <ul style="list-style-type: none"> <li>• VolunteerID</li> </ul> <b><u>Student</u></b> <ul style="list-style-type: none"> <li>• StudentID</li> <li>• ResID</li> </ul> <b><u>Language:</u></b> <ul style="list-style-type: none"> <li>• LanguageID</li> <li>• Language</li> </ul>
9. Report Subsystem	9.5 Generate Event Popularity Report	9.5 Generate Event Popularity Report	9.5.1 Retrieve information 9.5.2 Calculate information using retrieved data 9.5.3 Generate report using retrieved and calculated data	<b><u>Function</u></b> <ul style="list-style-type: none"> <li>• FunctionID</li> </ul> <b><u>CommunityOutreach</u></b> <ul style="list-style-type: none"> <li>• ComID</li> </ul> <b><u>Lecture</u></b> <ul style="list-style-type: none"> <li>• LectureID</li> </ul> <b><u>Attendance</u></b> <ul style="list-style-type: none"> <li>• AttendanceID</li> </ul>
9. Report Subsystem	9.6 Generate Donations Report	9.6 Generate Donations Report	9.6.1 Retrieve information 9.6.2 Calculate information	<b><u>Donor</u></b> <ul style="list-style-type: none"> <li>• DonorName</li> <li>• DonorSurname</li> <li>• DonorOrganisation</li> </ul> <b><u>Donations</u></b> <ul style="list-style-type: none"> <li>• DonationAmount</li> </ul>

Subsystem	Requirement	Use case	Primitive Level	Entities and Attributes
			using retrieved data 9.6.3 Generate report using retrieved and calculated information	
9. Report Subsystem	9.7 Generate User Statistics Report	9.7 Generate User Statistics Report	9.7.1 Retrieve information 9.7.2 Calculate information using retrieved data 9.7.3 Generate Report using retrieved and calculated information	<b><u>Audit Log</u></b> <ul style="list-style-type: none"> <li>• LogInTime</li> <li>• LogInDate</li> <li>• LogInDuration</li> </ul> <b><u>Person</u></b> <ul style="list-style-type: none"> <li>• PersonID</li> </ul> <b><u>User Type</u></b> <ul style="list-style-type: none"> <li>• UserTypeID</li> </ul>

## 8.2.10 Donation Subsystem

Subsystem	Requirement	Use case	Primitive Level	Entities and Attributes
10. Donation Subsystem	10.1 Add Donor	10.1 Add Donor	10.1.1 Enter donor information 10.1.2 Verify donor information 10.1.3 Save donor details	<u><b>Donor</b></u> <ul style="list-style-type: none"> <li>• Name</li> <li>• Surname</li> <li>• Organisation Name</li> <li>• Phone Number</li> <li>• Email Address</li> </ul>
10. Donation Subsystem	10.2 Receive Donation	10.2 Receive Donation	10.2.1 Enter donor details 10.2.2 Verify donor details 10.2.3 Enter donation information 10.2.4 Save donation information	<u><b>Donor</b></u> <ul style="list-style-type: none"> <li>• Email Address</li> </ul> <u><b>Donation</b></u> <ul style="list-style-type: none"> <li>• ProofOfPayment</li> <li>• DonationAmount</li> </ul>

## 8.3 Conclusion

The purpose of the validation was to ensure that the functional specifications detailed above are in line with all the requirements from the project proposal deliverable. It displayed the subsystem, the requirement from the subsystem, the use case that pertains to that subsystem, each process in the primitive level that pertain to that use case as well as all the entities and attributes that are affected by that Use Case.

## 9. Conclusion

9.1 In this document, the group discussed the functional specifications with regards to the TRWLA system. The use case diagrams were illustrated and the following subsystems were identified in the system; User, Volunteer, Student, Event, Function, Community Engagement, Lecture, Marketing, Report and Donation subsystem. After the use case diagrams were illustrated, the steps explaining each use case in the use case diagram was depicted in the use case narrative. The process models were then illustrated where the data flowing throughout the system were described and identified in the decomposition, context, high-level, mid-level and primitive-level diagrams. After the process models were identified, the UML models were illustrated by the use of activity diagrams. The activity diagrams described how each activity in the process of the system starts and ends. The data model entity-relationship diagram was then illustrated, where the diagram explained how the database for the system will be produced. The inputs and outputs regarding the data flows in the context diagram were then explained in tables with each of their descriptions. Lastly, the validations regarding each data flow in the primitive-level diagrams were explained in a table with each of their descriptions.





## 10. Complexity

Topic	Level		Marks	M
<b>1. Special GUI</b>	Appropriate MDS and SDI form design of the system	*	3	Yes
	Appropriate use of grids		3	Yes
	Appropriate use of tabs		3	Yes
	Use of graphs in an appropriate business context		3	Yes
	The storage and display of graphical information, like photos with a good business reason		3	Yes
	Working e-mail automatically generated from the database in an appropriate business context		3	Yes
	SMS messages automatically generated from the system in an appropriate business context		3	Yes
	Extensive user-friendly search facility		3	Yes
	At least one use of a tree to display data		3	Yes
	At least one use of a calendar view of data		3	Yes

Topic	Level		Marks	M
	Uploading a file into the system with appropriate business reason		3	Yes
	The use of multimedia in an appropriate business context		3	Yes
	At least one use of a timer in an appropriate business context		3	Yes
2. Database access	At least 30 tables used (4 member groups) or 40 tables used (5 member groups)	*	6	Yes
	Full referential integrity on all tables	*	6	Yes
	At least one use of master-detail table relationships	*	3	Yes
3. Reports	At least 5 simple list reports in a reporting tool (no control breaks, no graphs, single table)	*	6	Yes
	At least 1 transactional report with 2 or more control breaks (with heading and total lines, multiple tables)		3	No
	At least 1 management report using a graph		3	Yes
4. Flexibility	All data that can change in future should not be hard coded but maintained in a		6	Yes

Topic	Level		Marks	M
	sub-module of the system (e.g. Lookup tables)			
	Some business rules are not hard coded, but maintained in a sub-module of the system.		6	Yes
5. Error handling	All system-generated errors are trapped and consistent, user-friendly error messages are displayed		6	Yes
	Appropriate data validation on all input fields		6	Yes
6. Help	At least one menu item or other control that opens up a complete help document (HTML, PDF, Help-file)		3	Yes
	Extensive context-sensitive help. E.g. calling Help on a specific screen/function will automatically open the specific help for that screen/function.		6	No
	Search Facility on Help		3	Yes
	Extensive use of hints		3	Yes
7. Security	Logon screen with user ID and password and fixed user profiles		3	Yes

Topic	Level		Marks	M
	Encrypted passwords in database		3	Yes
	Flexible user profiles (i.e. you can dynamically add user profiles that will enable/disable access to certain parts of the system)		6	Yes
8. Audit Trail	An audit trail of all transactions in the system showing at least date, time, user, transaction type, critical data (such as amount and quantity of transaction)		6	Yes
	Able to search the audit trail on any of the following: date, user, transaction type		3	Yes
9. Installation	Fully functional installation disks that take care of application installation requirements (install and uninstall)		6	No
	Fully functional installation disks that take care of database installation requirements (including database settings)		6	No
10. Backup and Restore	A backup and restore subsystem exists that backup/restore all data (system may exit during restore)		6	Yes
	OLE: Opens Word or Excel and automatically places data in it based on		6	Yes

Topic	Level		Marks	M
11. Import/Export Data	the selected data in the calling screen (with good business reason)			
	Text File: At least 1 text file for Importing or Exporting of data (with good business reason)		3	Yes
	XML: At least 1 XML file for Importing or Exporting of data (with good business reason)		3	Yes
12. External INPUT device	Simple Link to an external INPUT device using Windows plug-and-play technology. (This could include a swipe card reader, bar code reader, etc.)		3	No
	Loose Link to an external INPUT device using device specific software. Data or images must seamlessly be stored in the database but device specific software is visible to the user. (This could include a digital camera, scanner, voice recording device, thump print reader, etc.)		6	No
	Tight Link to an external INPUT device using device specific software. Data or images must seamlessly be stored in the database but device specific software is <b>not</b> visible to the user. (This could include a digital camera, scanner, voice		9	No

Topic	Level		Marks	M
	recording device, thumb print reader, etc.)			
<b>13. External APPLICATION / Services</b>	Integrate an existing web service into your application (with good business reason)		3	<b>Yes</b>
	A fully functional link to an installed external application system exists and the interface must be shown to work on the external system. Note that this excludes Microsoft Office Applications		6	<b>No</b>
<b>14. Web processing</b>	At least one appropriate business use of static Web-pages (e.g. Help files or an advertisement for your client)		3	<b>Yes</b>
	Substantial Web-server processing – Display data from a database on the browser		3	<b>Yes</b>
	Substantial Web-server processing – add data from a browser into the system		6	<b>Yes</b>
	Substantial Web-server processing – uploading a file from a local PC to the web-server (integrated into the system)		6	<b>Yes</b>

Topic	Level		Marks	M
	Substantial mobile device processing integrated into the system (e.g. Smartphone, Tablet)		9	Yes
15. Programming Principals	The system consists of three distinct tiers: data; business; and presentation. Each of the levels consists of a separate application object.		6	Yes
	Basic interfacing to the Windows system registry for appropriate application data and settings		3	No
	Comprehensive use of stored procedures and/or triggers		3	Yes





## 11. Sign-off by Client

I \_\_\_\_\_ hereby agree to the contents of the above document.

Signed on this the \_\_\_\_ of May 2017 at \_\_\_\_\_.

\_\_\_\_\_

**Bes Liebenberg**

## 12. Sign-off by Team

The team hereby agrees that each member has contributed towards and agrees with the contents of the above document.

Signed on this the \_\_\_\_ of May 2017 at \_\_\_\_\_.

\_\_\_\_\_  
**Cailin Smith**

\_\_\_\_\_  
**Christopher Oakes**

\_\_\_\_\_  
**Amogelang Moloko**

\_\_\_\_\_  
**Achal Seechoonparsad**

\_\_\_\_\_  
**Jacquiline Lawler**