

Software Requirement Specification

Complain Box

Course: SE – 505 Software Project Lab – 2

Submitted by

Atiq Ahammed	BSSE0817
Khayrul Islam	BSSE0822

Supervised by

Shah Mostafa Khaled
Associated Professor
Institute of Information Technology
University of Dhaka

Submitted To

SPL – 2 Program Committee
Institute of Information Technology
University of Dhaka

Document Version: 1.0

Date: 20 – 03 – 2018



Institute of Information Technology

University of Dhaka.



Letter of Transmittal

20th March 2018
SPL – 2 Program Committee
Institute of Information Technology
University of Dhaka.

Subject: Submission of software requirement specification of Software Project Lab 2.

Dear Sir,

With due respect, we are pleased to submit the final report on software requirements specifications of Complain Box. Although this report may have short comings we did try our level best to produce an acceptable software requirements specifications. We would be highly obliged if you overlooked our mistakes and accepted our effort we put in this SRS.

Sincerely yours,

Atiq Ahammed
BSSE 0817

Khayrul Islam
BSSE 0822

Shah Mostafa Khaled
Associated Professor
Institute of Information Technology
University of Dhaka

Acknowledgement

At first, we would like to thank almighty for helping us complete this project.

We would like to express our deepest gratitude to all those who provided us the support and encouragement to complete this project. Thanks to our supervisor teacher Shah Mostafa Khaled, Associated Professor, Institute of Information Technology, University of Dhaka, whose continuous suggestions and guidance has been invaluable to us. Without such stimulating suggestions, the project could not have made this much progress.

We are grateful to the Institute of Information Technology for giving us the opportunity to do such a project.

Abstract

This document contains the software requirements and specifications for the Software Project Lab 2. It contains a scenario based model, data based model, class based model, data flow and behavioral model. Using this document as a guide, we are describing the requirements, necessary diagrams, procedures, design for database and working sequence of our project.

Our project is about online based complain management system. Here we will discuss how we will identify the requirements, how to analyze them and how to present a recommended solution for the system.

This will help to make the software according to the demand of the stakeholders.

Contents

Chapter 1: Introduction	1
1.1 Purpose	1
1.2 Intended Audiences	1
1.3 Conclusion	1
Chapter 2: Inception	2
2.1 Introduction	2
2.1.1 Listing down Stakeholders	2
2.1.2 Recognizing Multiple Viewpoints.....	2
Chapter 3: Elicitation.....	4
3.1 Introduction	4
3.2 Eliciting Requirements	4
3.2.1 Collaborative Requirements Gathering	4
3.2.2 Problem in the Scope	4
3.2.3 Quality Function Deployment.....	5
3.2.3.1 Normal Requirements.....	5
3.2.3.2 Expected Requirements.....	5
3.2.3.3 Exciting Features	6
3.2.4 Usage Scenario.....	6
3.2.4.1 Problem Submission	6
3.2.4.1.1 Authentication	6
3.2.4.1.2 Submitting Problem	7
3.2.4.1.3 Viewing User Account.....	7
3.2.4.1.4 Emergency Support.....	8
3.2.4.1.5 Notice Board	8
3.2.4.1.6 Download Application Forms.....	8
3.2.4.2 Problem Supervision	8
3.2.4.2.1 Authentication	9
3.2.4.2.2 View Submitted Problem	9
3.2.4.2.3 Update Problem Information.....	9
3.2.4.2.4 Update Notice, Event and Application Forms.....	10
3.2.4.2.5 Update Emergency Information	10
Chapter 4: Scenario Based Modeling.....	11

4.1 Introduction	11
4.2 Use Case	11
4.3 Activity diagram	11
4.4 Use Case and Activity Diagram	12
4.4.1 Level 0 Use Case Diagram of Complain Box.....	12
4.4.2 Level 1 Use Case Diagram of Complain Box.....	13
4.4.3 Level 1.1 Use Case Diagram of Complain Box.....	14
4.4.4 Level 1.1 Activity Diagram of Complain Box	15
4.4.5 Level 1.1.1 Use Case Diagram of Complain Box.....	16
4.4.6 Level 1.1.1 Activity Diagram of Complain Box	18
4.4.7 Level 1.1.2 Use Case Diagram of Complain Box.....	19
4.4.8 Level 1.1.2 Activity diagram of Complain Box.....	20
4.4.9 Level 1.1.4 Use Case Diagram of Complain Box.....	21
4.4.10 Level 1.1.4 Activity diagram of Complain Box.....	22
4.4.11 Level 1.1.6 Use Case Diagram of Complain Box.....	23
4.4.12 Level 1.1.6 Activity Diagram of Complain Box	24
4.4.13 Level 1.2 Use Case Diagram of Complain Box.....	25
4.4.14 Level 1.2 Activity Diagram of Complain Box	26
4.4.15 Level 1.2.1 Use Case Diagram of Complain Box.....	27
4.4.16 Level 1.2.1 Activity Diagram of Complain Box	28
4.4.17 Level 1.2.2 Use Case Diagram if Complain Box.....	29
4.4.18 Level 1.2.2 Activity Diagram of Complain Box	30
4.4.19 Level 1.2.3 Use Case Diagram of Complain Box.....	30
4.4.20 Level 1.2.3 Activity Diagram of Complain Box	31
4.4.21 Level 1.2.4 Use Case Diagram of Complain Box.....	32
4.4.22 Level 1.2.4 Activity Diagram of Complain Box	33
4.4.23 Level 1.2.5 Use Case Diagram of Complain Box.....	33
4.4.24 Level 1.2.5 Activity Diagram of Complain Box	34
Chapter 5: Data Based Modeling	35
5.1 Introduction	35
5.2 Data Object Selection.....	35
5.2.1 Noun Identification	35
5.2.2 Potential Data Objects	37

5.2.3 Analysis for Finalizing Data Objects	38
5.2.4 Final Data Objects	38
5.3 Data Object Relations	39
5.4 Entity Relationship Diagram.....	40
5.5 Schema Diagram	40
Chapter 6: Class Based Modeling.....	43
6.1 Introduction	43
6.2 Grammatical Parsing and Class Analysis	43
6.2.1 General Classification.....	43
6.2.2 Selection Criteria.....	45
6.2.3 Preliminary Classes	46
6.3 Associated Noun and Verb Identification	47
6.4 Attribute Selection	48
6.5 Method Selection.....	49
6.6 Finalized Classes.....	50
6.6.1 Finalized classes for Problem Submission.....	50
6.6.2 Finalized classes for Problem Supervision	52
6.7 Class Responsibility Collaborator Modeling.....	54
6.7.1 Class Responsibility Collaborator Modeling for Problem Submission	54
6.7.2 Class Responsibility Collaborator Modeling for Problem Supervision.....	56
6.8 Class Diagram.....	58
6.8.1 Class Diagram of Problem Submission.....	58
6.8.1 Class Diagram of Problem Supervision	59
Chapter 7: Data Flow Modeling	60
7.1 Introduction	60
7.2 Data Flow Diagram.....	60
7.2.1 Level 0 Data Flow Diagram of Complain Box	60
7.2.2 Level 1 Data Flow Diagram of Complain Box	61
7.2.2.1 Level 1.1 Data Flow Diagram of Complain Box	62
7.2.2.1.1 Level 1.1.1 Data Flow Diagram of Complain Box	63
7.2.2.1.2 Level 1.1.2 Data Flow Diagram of Complain Box	63
7.2.2.1.3 Level 1.1.3 Data Flow Diagram of Complain Box	64
7.2.2.2 Level 1.2 Data Flow Diagram of Complain Box	65

7.2.2.2.1 Level 1.2.1 Data Flow Diagram of Complain Box	66
7.2.2.2.2 Level 1.2.2 Data Flow Diagram of Complain Box	66
7.2.2.2.3 Level 1.2.3 Data Flow Diagram of Complain Box	67
7.2.2.2.4 Level 1.2.4 Data Flow Diagram of Complain Box	67
7.2.2.2.5 Level 1.2.5 Data Flow Diagram of Complain Box	68
7.2.2.2.6 Level 1.2.6 Data Flow Diagram of Complain Box	68
Chapter 8: Behavioral Modeling	69
8.1 Introduction	69
8.2 State Transition	69
8.2.1 Event Identification	69
8.2.2 State Transition Diagram of Problem Submission Subsystem	71
8.2.3 State Transition Diagram of Problem Submission Subsystem	73
8.3 Sequence Diagram	75
8.3.1 Sequence Diagram of Problem Submission Subsystem	76
8.3.2 Sequence Diagram of Problem Supervision Subsystem	77

Tables

Table 1: Information about level 0 use case diagram	12
Table 2: Information about level 1 use case diagram	13
Table 3: Information about level 1.1 use case diagram	14
Table 4: Information about level 1.1.1 use case diagram	16
Table 5: Information about level 1.1.2 use case diagram	19
Table 6: Information about level 1.1.4 use case diagram	21
Table 7: Information about level 1.1.6 use case diagram	23
Table 8: Information about level 1.2 use case diagram	25
Table 9: Information about level 1.2.1 use case diagram	27
Table 10: Information about 1.2.2 use case diagram	29
Table 11: Information about level 1.2.3 use case diagram	31
Table 12: Information about level 1.2.4 use case diagram	32
Table 13: Information about level 1.2.5 use case	34
Table 14: Noun identification	35
Table 15: Final data objects of complain box	38
Table 16: Schema table of User data object	40
Table 17: Schema table of Problem data object	41
Table 18: Schema table of Location data object	41
Table 19: Schema table of Institute data object	41

Table 20: Schema table of Admin data object	41
Table 21: Schema table of Event data object	42
Table 22: Schema table of Update_Problem data object	42
Table 23: Schema table of Update_Institute data object	42
Table 24: Class identification with general classification	43
Table 25: Class identification with selection criteria	46
Table 26: Associated noun and verb identification	47
Table 27: Attribute selection form selected classes	48
Table 28: Method selection form selected classes	49
Table 29: Class of problem submission (User)	50
Table 30: Class of problem submission (Problem)	50
Table 31: Class of problem submission (ApplicationForm)	51
Table 32: Class of problem submission (Institution)	51
Table 33: Class of problem submission (Notice)	51
Table 34: Class of problem submission (Event)	51
Table 35: Class of problem submission (Location)	52
Table 36: Class of problem supervision (Problem)	52
Table 37: Class of problem supervision (ApplicationForm)	52
Table 38: Class of problem supervision (Admin)	53
Table 39: Class of problem supervision (Institution)	53
Table 40: Class of problem supervision (Notice)	53
Table 41: Class of problem supervision (Event)	53
Table 42: Class of problem supervision (Location)	54
Table 43: Event identification	69

Figures

Figure 1: Level 0 use case diagram of Complain Box	12
Figure 2: Level 1 use case diagram of Complain Box	13
Figure 3: Level 1.1 use case diagram of Complain Box	14
Figure 4: Level 1.1 activity diagram of Complain Box	15
Figure 5: Level 1.1.1 use case diagram of Complain Box	16
Figure 6: Level 1.1.1 activity diagram of Complain Box	18
Figure 7: Level 1.1.2 use case diagram of complain box	19
Figure 8: Level 1.1.2 activity diagram of complain box	20
Figure 9: Level 1.1.4 use case diagram of complain box	21
Figure 10: Level 1.1.4 activity diagram of complain box	22
Figure 11: Level 1.1.6 use case diagram of complain box	23
Figure 12: Level 1.1.6 activity diagram of complain box	24
Figure 13: Level 1.2 use case diagram of complain box	25
Figure 14: Level 1.2 activity diagram of complain box	26
Figure 15: Level 1.2.1 use case diagram of complain box	27

Figure 16: Level 1.2.1 activity diagram of complain box	28
Figure 17: Level 1.2.2 use case diagram of complain box.....	29
Figure 19: Level 1.2.2 activity diagram of complain box	30
Figure 20: Level 1.2.3 use case diagram of complain box.....	30
Figure 21: Level 1.2.3 use case diagram of complain box.....	31
Figure 22: Level 1.2.4 use case diagram of complain box.....	32
Figure 23: Level 1.2.4 activity diagram of complain box	33
Figure 24: Level 1.2.5 use case diagram of complain box.....	33
Figure 25: Level 1.2.5 Activity diagram of complain box	34
Figure 26: Data object relationship of complain box.....	39
Figure 27: Entity relationship diagram of complain box.....	40
Figure 28: CRC diagram of User	54
Figure 29: CRC diagram of Problem	54
Figure 30: CRC diagram of ApplicationForm	55
Figure 31: CRC diagram of Institution	55
Figure 32: CRC diagram of.....	55
Figure 33: CRC diagram of Event.....	55
Figure 34: CRC diagram of Location	55
Figure 35: CRC diagram of Problem	56
Figure 36: CRC diagram of ApplicationForm	56
Figure 37: CRC diagram of Admin	56
Figure 38: CRC diagram of Institution	56
Figure 39: CRC diagram of Notice	56
Figure 40: CRC diagram of Event.....	57
Figure 41: CRC diagram of Location	57
Figure 42: Class diagram of problem submission subsystem of complain box	58
Figure 43: Class diagram of problem supervision subsystem of complain box	59
Figure 44: Level 0 data flow diagram of complain box	60
Figure 45: Level 1 data flow diagram of complain box	61
Figure 46: Level 1.1 data flow diagram of complain box	62
Figure 47: Level 1.1.1 data flow diagram of complain box	63
Figure 48: Level 1.1.2 data flow diagram of complain box	63
Figure 49: Level 1.1.3 data flow diagram of complain box	64
Figure 50: Level 1.2 data flow diagram of complain box	65
Figure 51: Level 1.2.1 data flow diagram of complain box	66
Figure 52: Level 1.2.2 data flow diagram of complain box	66
Figure 53: Level 1.2.3 data flow diagram of complain box	67
Figure 54: Level 1.2.4 data flow diagram of complain box	67
Figure 55: Level 1.2.5 data flow diagram of complain box	68
Figure 56: Level 1.2.6 data flow diagram of complain box	68
Figure 57: State transition diagram of User.....	71
Figure 58: State transition diagram of Problem	71
Figure 59: State transition diagram of Notice.....	72
Figure 60: State transition diagram of Location	72

Figure 61: State transition diagram of Institution	72
Figure 62: State transition diagram of Event	72
Figure 63: State transition diagram of ApplicationForm	72
Figure 64: State transition diagram of Admin.....	73
Figure 65: State transition diagram of ApplicationForm	73
Figure 66: State transition diagram of Event	74
Figure 67: State transition diagram of Institution	74
Figure 68: State transition diagram of Location	74
Figure 69: State transition diagram of Notice.....	75
Figure 70: State transition diagram of Problem	75
Figure 71: Sequence diagram of problem submission subsystem.....	76
Figure 72: Sequence diagram of problem supervision subsystem	77

Chapter 1: Introduction

This chapter is a part of our Software Requirement Specification and Analysis for the project “Complain Box”. In this chapter, the center of our activity is the intended audience for this project.

1.1 Purpose

This document is the simple outline of the Software Requirement Specification and Analysis of our project for Software Project Lab – 2 titled as “Complain Box”. It contains functional, non – functional and supporting requirements and establishes a requirement – baseline for developing the system. The Software Requirement Specification holds the requirements that were collected from Dhaka South City Corporation Authority (DSCCA) and people living in Dhaka South City Corporation (DSCC). The Software Requirement Specification serves as a platform to forward user requirements to us and provides a common reference point for both our team and DSCCA.

1.2 Intended Audiences

Our Software Requirement Specification (SRS) is pinned for several audiences including DSCCA as well as our project supervisor, SPL – 2 coordinators and ours.

- DSCCA will use this SRS to verify that we have developed a product that the required
- Our supervisor will use this SRS to plan milestones and ensure that we are on the right track when developing the system
- We will use this SRS as a basis for creating the system design. We will continually refer back to this SRS to ensure that the system we are designing, will fulfill the requirements of DSCCA and the people living in DSCC
- We will also use this SRS as a basis for developing the system functionality and link the requirements defined in this SRS to the software that we will create to ensure that we have created a software that will fulfill all the requirements

1.3 Conclusion

We wish, this analysis of the audience will help us to focus on the users who will be using our analysis. This document will help each and every person related to this project to perceive the subject matter of the project.

Chapter 2: Inception

In this chapter, we succinctly discuss the Inception part of the SRS for our “Complain Box”.

2.1 Introduction

This is our first aspect of requirement engineering. In this stage, we tried to understand the problem, identified the people living in DSCC and DSCCA as our stakeholders and communicate with them.

For clear perception of the software requirements, we established a groundwork involving the following steps-

1. We listed the DSCCA and people living in DSCC as the stakeholder
2. We collected their multiple viewpoints
3. We worked towards collaboration
4. We broke down the ice and initiated communication with DSCCA and people living in the DSCC

2.1.1 Listing down Stakeholders

To identify stakeholders, we consulted a number of people living in DSCC and DSCCA, asked them the following questions:

- Who will be using the product?
- What kind of features they want in the product?
- Whose work will be affected by this product?

We identified the following stakeholders for our project.

- People living in DSCC
- Dhaka South City Corporation Authority (DSCCA)

2.1.2 Recognizing Multiple Viewpoints

The list of stakeholders will contribute to the input when requirements are elicited. Every stakeholder has different views of the system and achieves different benefits when the system is developed.

Viewpoints of people living in Dhaka City Corporation:

- User can submit problem easily
- User authentication system
- Complaining system to submit problem to DSCCA
- Facility to inform problem description to the authority

- Submitting problem within minimum time
- Easy environment for selecting problem category
- Environment for selecting photos or taking instance photo with phone category
- Viewing information about problem category

Viewpoint of Dhaka city corporation Authority

- Manage submitting problem easily
- Web based system for problem management
- Authentication system for DSCCA
- Environment for viewing submitted problem
- Automatic ward–wise selection system
- Problem ranking system
- Disable problem

Conflicting viewpoints

- User account for submitting problem
- Submitting username with problem
- Account recovery
- Submitting video or audio files for describing problem
- Disable problems

Common viewpoints

- Authentication system for user account
- Using location for specifying problem

Chapter 3: Elicitation

After discussing on the Inception phase, we need to focus on the Elicitation phase. So, this chapter specifies the Elicitation phase.

3.1 Introduction

The second phase of requirements engineering is elicitation. The main task of elicitation is to combine elements of problem solving, elaboration, negotiation and specification. Gathering information from stakeholders regarding the problem was not sufficient to design the software. The problems that arose, were encountered following the principles of elicitation.

3.2 Eliciting Requirements

We worked together with DSCCA and people living in DSCC to identify the problems, propose elements of the solutions, negotiate different approaches and specify an initial set of solution requirements. This approach is sometimes called Facilitated Application Specification Technique (FAST). Elicitation has some sub-phases which are:

1. Collaborative Requirements Gathering
2. Quality Function Deployment
3. Usage Scenario
4. Elicitation Work Products

3.2.1 Collaborative Requirements Gathering

During inception, basic questions and answers established the scope of the problem. However, some problems persisted about the scope as the boundary of the system was ill defined and the people living in DSCC and DSCCA have stated some unnecessary confusing detail. Also, our stakeholders do not have a proper understanding about the abilities of the computing environment which results in further discussion regarding the problem domain and product requirements. The requirements were put under re-evaluation by conducting meeting with stakeholders (DSCCA and people of DSCC) and we went into further investigation about their requirements and expectations. The final requirement list was derived at the end of the meeting.

3.2.2 Problem in the Scope

A number of problems were encountered in the course of preparing the software requirement specification and analysis of “Complain Box”.

Scope

Software is developed for only submitting and managing problems.

Limitation

Internet is required for both submitting and managing problems.

3.2.3 Quality Function Deployment

Quality Function Deployment (QFD) is a quality management technique that translates the needs of the clients into technical requirements for the software. The prime concern of the QFD is customer satisfaction maximization. In order to ensure this, QFD enforces an understanding of what the customers describe as 'valuable' and then deploy these values throughout the engineering process.

QFD defines three types of requirements:

- Normal requirements
- Expected requirements
- Exciting features

3.2.3.1 Normal Requirements

Normal requirements refer to the objectives and the goals that are stated for the product during the meeting with the stakeholders. The presence of these requirements ensures the satisfaction of the customers. The normal requirements for the project are stated below.

- Authentication system for user and authority of City corporation system
- Easy environment for selecting or capturing photo for submitting problem
- Environment for informing the authority about problem detail
- Easy environment for selecting location of problem
- Automatic ward-wise problem categorization for authority
- Ranking system for problem
- Problem list management

3.2.3.2 Expected Requirements

The requirements that are implicit to the system might not be brought up during the meeting because of their fundamental nature. Despite being not explicitly mentioned their presence must be ensured. Otherwise, the product will leave customers dissatisfied. These requirements are called expected requirements and these are stated below.

- Error-free problem submission within minimum time
- More efficient than the existing workflow for submitting and recognizing problem
- Secured Authentication system for both authority and user

- Web based server
- Database for storing problem information

3.2.3.3 Exciting Features

The factors that go beyond the customer's expectations and prove to be satisfying when present are called exciting features. The exciting features are the so called 'wow factor' for our project.

- Viewing own submitted problem
- Viewing problems with continuous updated status

3.2.4 Usage Scenario

Complain Box will be an online based complaint (problem) collection system. Using this system, people living in DSCC will be able to submit those problems that they face in their daily city life to notify DSCCA. On the other hand, DSCCA will be able to view those submitted problems and take necessary steps to solve those problems.

Complain Box has the following subsystem:

1. Problem Submission
2. Problem Supervision

3.2.4.1 Problem Submission

In this subsystem, the people living in DSCC will submit their problems that they face in their daily life and need to inform the DSCCA to solve the problems. This subsystem will work on Android-based smart-phone. This subsystem is divided into:

3.2.4.1.1 Authentication

Initially, the user will create a user account to use this system if she/he do not have an account already. To create a user account, the user will provide her/his name, valid phone number and password. The password should be at least five characters and at most fifteen characters and name should be at least two characters and at most twenty characters long. A unique phone number will be used to create only one account. After creating a user-account the information will be stored in the system and the user will automatically be logged in to the system. Besides a user having a user account, will be able to log in to the system using her/his phone number and password that she/he would have used to create that account. If a user forgets her/his password, the user will request to recover her/his password providing her/his valid phone number that she/he would have used to create the user account. Then, the system will send a message containing the password. A logged in user will be able to log out from the system.

3.2.4.1.2 Submitting Problem

To submit a problem, the user will first select the problem category. The problem category includes:

- Solid Waste Management
- Road, Footpath and Drain
- Drainage
- Repair to Municipal Property
- Road light
- Illegal parking
- Mosquito problem
- Health Department
- Revenue Department
- Social Welfare Department
- Planning Department
- Miscellaneous Complains
- Establishment Sections
- Encroachment
- Factories
- License
- Corruption

Note: problem category was defined by Md. Abu Taiyeb Rokon, System analyst, DSCC.

Then, the user will capture a picture or video, or record an audio file or select one from the storage which will be related to a specific problem, add problem descriptions and select the problem location. The picture, video or audio file describing the problem must be within twenty Megabyte. The system will store the latitude and longitude of the location. Problem description should be maximum five hundred characters. The user will be able to use her/his current location automatically or manually set the problem location, but the location must be within DSCC Area.

3.2.4.1.3 Viewing User Account

The user will be able to view her/his own account information. The user account contains:

- User name
- Phone number
- Reward point
- Problems submitted by the user

The information of the problem that the user will be able to view will contain:

- Problem ID
- Photo/video/audio of the problem
- Description about the problem

- Problem category
- Problem location
- Submission date and time
- Current status of the specific problem
- Problem priority
- Ward number in which the problem would have occurred

The user is unable to change any information of the problem if it once submitted.

3.2.4.1.4 Emergency Support

The user will get necessary information and communication facility of hospitals, fire station, police station within the area of DSCC. The information contains:

- Name of institute
- Location
- Phone number

of hospitals/fire stations/police stations.

3.2.4.1.5 Notice Board

The user will able to view the information of notices and events send by DSCC. The information includes notice content, event name, publishing date, event date and time.

3.2.4.1.6 Download Application Forms

The user can download some application forms as pdf file. The application form includes

- Passport Application Form
- E-Income Tax Certificate
- Driving License Form
- Birth Certificate
- Trade License Form
- Tax Form
- Inherit Certificate
- Death Certificate

3.2.4.2 Problem Supervision

This subsystem will work in a web-based system. Only system admin (selected person from DSCCA) will operate the system and take a decision on the submitted problem. This subsystem will be divided into:

3.2.4.2.1 Authentication

The admin of the system will log in to the system providing email address and password that would be defined by the developer. There will be no scope to create another admin account or adding a new admin to the system. The admin will also be able to log out of the system. Admin can request for recover account if she/he forget the password. Then, the system will send the password to the email address that the admin uses to log in to the system from beginning.

3.2.4.2.2 View Submitted Problem

Admin will be able to view all the submitted problem. Admin will also be able to view problems based on the following type:

- Problem id
- Problems in particular category
- Solved problems
- Problems in progress
- Problems under consideration
- Submission date
- Problems in a particular Ward

The admin will be able to view the following information about a particular problem

- Problem id
- Description and photo (if submitted)
- Category
- Location
- Submission date and time
- Status (by default the status will be under consideration)
- Priority (initially priority will be 0)
- Ward number

3.2.4.2.3 Update Problem Information

The admin will be able to edit the problem category if she/he thinks that the user has chosen wrong problem category. The admin will select the priority based on the importance of the submitted problem. She/he will update the status of problems as

- Problem under consideration
- Problem in progress
- Solved

3.2.4.2.4 Update Notice, Event and Application Forms

The admin will upload the notice for the users. The notice will be in pdf. She/he will also be able to create event providing event date, time and description. Besides, she/he will also upload or change application forms for the users.

3.2.4.2.5 Update Emergency Information

The admin will be able to add emergency support information providing name, selecting location and add phone number of the institution. She/he also can update the information of emergency support institutions.

Chapter 4: Scenario Based Modeling

This chapter contains the Scenario Based Model for our project “Complain Box”.

4.1 Introduction

For developing our software, we are giving the highest priority to user satisfaction. To identify the requirements to establish meaningful analysis and design model we determine how users and DSCCA want to interact with the system. Thus, our requirements modeling begins with scenario generation in the form of use cases, activity diagrams.

4.2 Use Case

Use case diagrams are usually referred to as behavior diagrams used to describe a set of actions that some system or sub-systems can perform in collaboration with one or more external users of the system.

The first step in writing a Use Case is to define that set of “actors” that will be involved in the story. Actors are the different people that use the system or product within the context of the function and behavior that is to be described. Actors represent the roles that people play as the system operators. Every user has one or more goals when using the system.

Primary Actor

Primary actors interact directly to achieve required system function and derive the intended benefit from the system. They work directly and frequently with the software. In our system both users (people living in DSCC) and Admin to supervise the system both are primary actor.

Secondary Actor

Secondary actors support the system so that primary actors can do their work. They either produce or consume information. In our system we have no secondary actor.

4.3 Activity diagram

Activity diagrams are graphical representations of workflows of stepwise activities and actions with support for choice, iteration and concurrency. In this chapter we did try to provide each use case and its corresponding activity diagram together.

4.4 Use Case and Activity Diagram

4.4.1 Level 0 Use Case Diagram of Complain Box

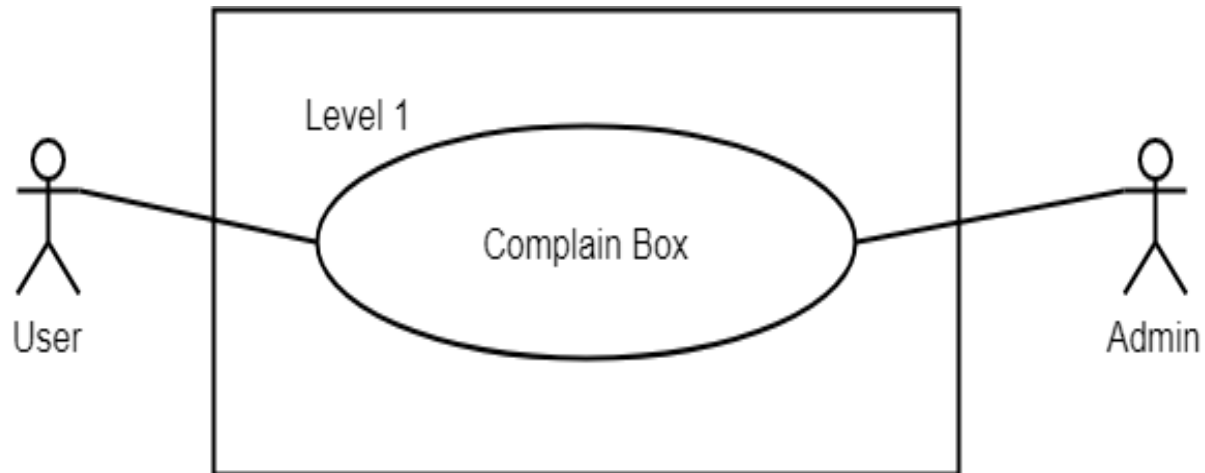


Figure 1: Level 0 use case diagram of Complain Box

Table 1: Information about level 0 use case diagram

Name:	Complain Box
ID:	CB-L-0
Primary Actor:	User, Admin
Secondary Actor:	None

Description of Level 0 Use Case Diagram

After analyzing usage scenario, we found that user and admin interact with our system. Both of them are primary actor.

4.4.2 Level 1 Use Case Diagram of Complain Box

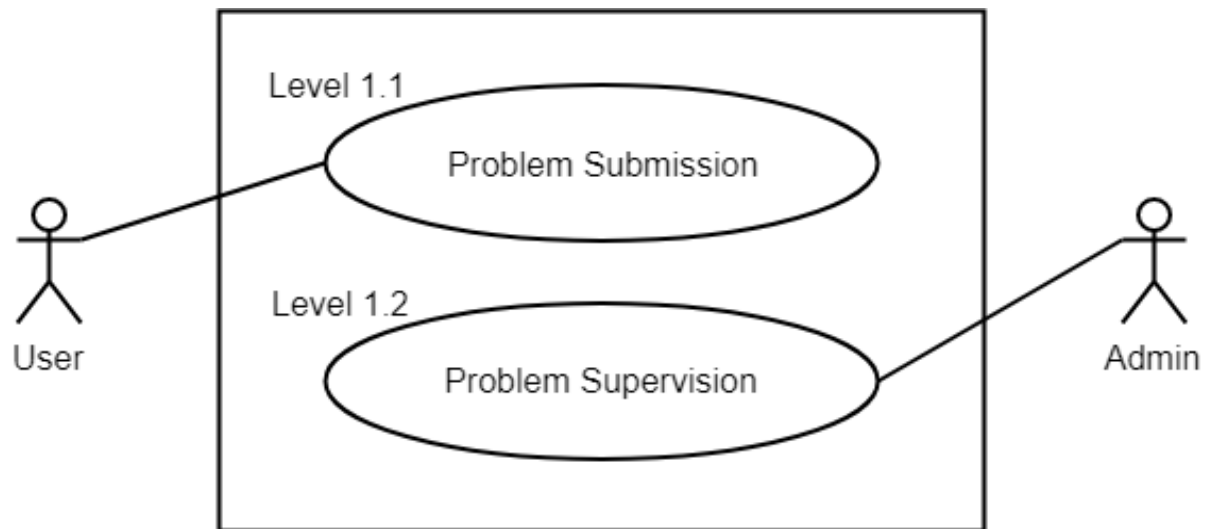


Figure 2: Level 1 use case diagram of Complain Box

Table 2: Information about level 1 use case diagram

Name:	Sub systems of Complain Box
ID:	CB-L-1
Primary Actor:	User, Admin
Secondary Actor:	None

Description of Level 1 Use Case Diagram

In the usage scenario we separated our System into two subsystems named problem submission and problem supervision. The user only interact with the problem submission subsystem and the admin interact with the problem supervision. Here, both of the actors are primary actors.

4.4.3 Level 1.1 Use Case Diagram of Complain Box

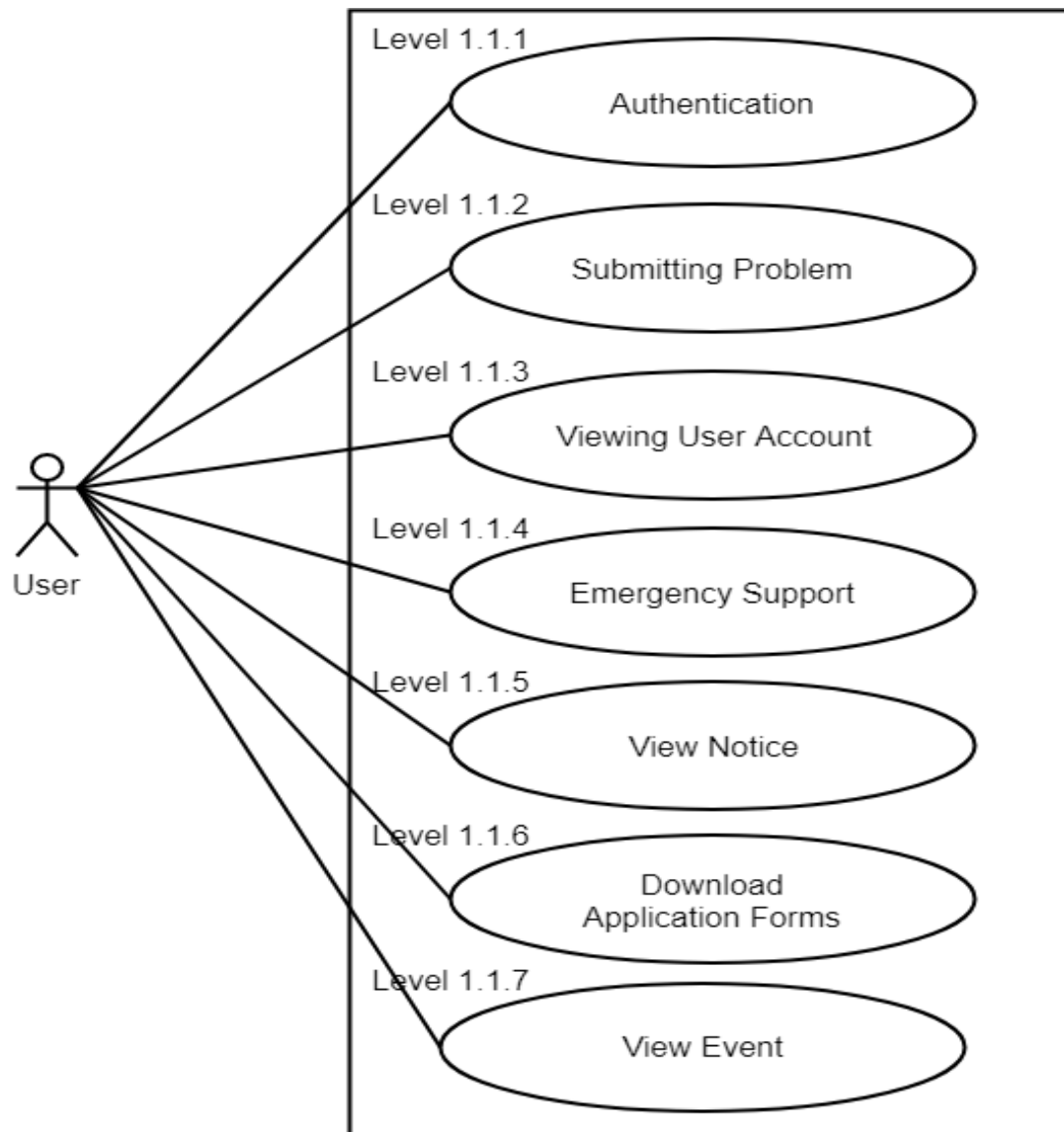


Figure 3: Level 1.1 use case diagram of Complain Box

Table 3: Information about level 1.1 use case diagram

Name:	Problem submission
ID:	CB-L-1.1
Primary Actor:	User
Secondary Actor:	None

Description of Level 1.1 Use Case Diagram

Only user interact with this subsystem. In problem submission subsystem, a user can

- Authenticate to the system
- Submit problem
- View user's own account
- View information of emergency support institute
- View notices
- Download application forms
- View events

4.4.4 Level 1.1 Activity Diagram of Complain Box

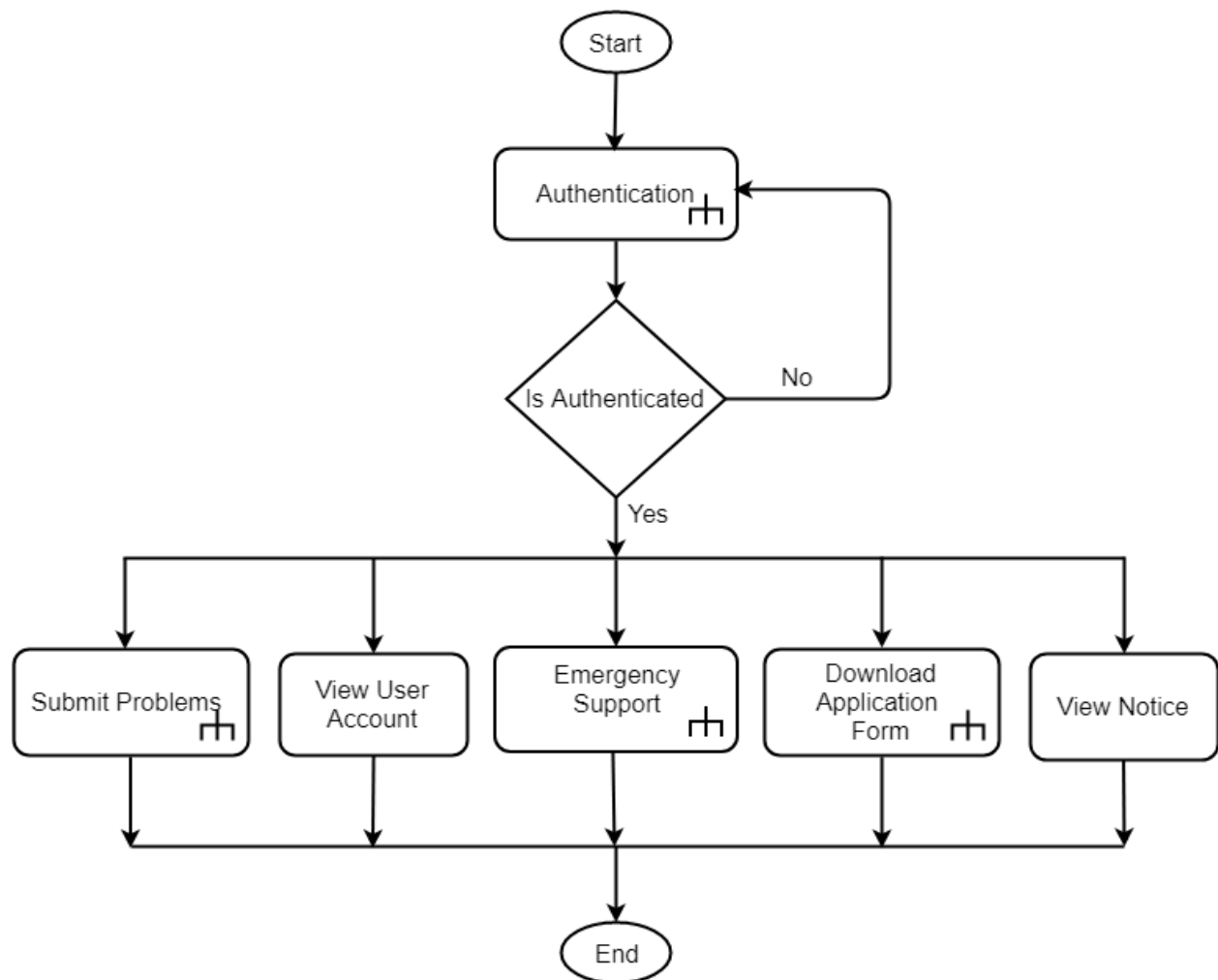


Figure 4: Level 1.1 activity diagram of Complain Box

4.4.5 Level 1.1.1 Use Case Diagram of Complain Box

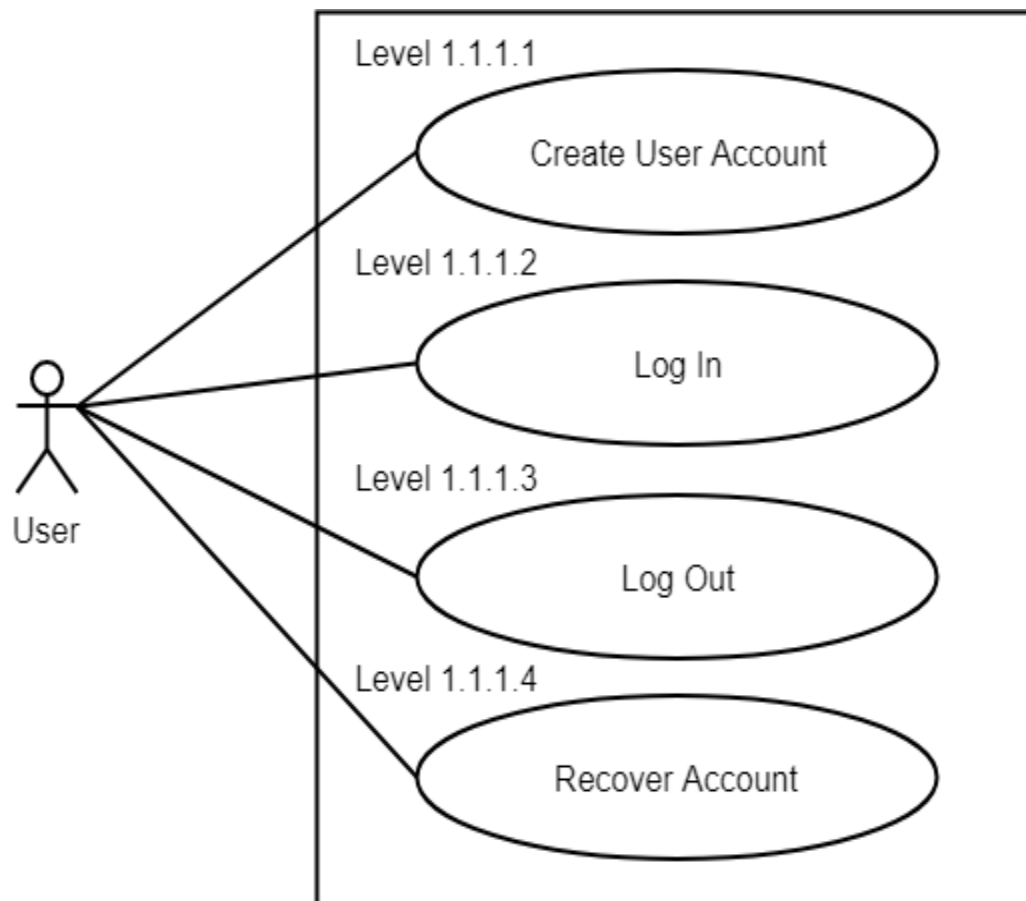


Figure 5: Level 1.1.1 use case diagram of Complain Box

Table 4: Information about level 1.1.1 use case diagram

Name:	User authentication
ID:	CB-L-1.1.1
Primary Actor:	User
Secondary Actor:	None

Description of Level 1.1.1 Use Case Diagram

In this case, the user can create a user account to use this system if she/he do not have an account already, login to the system, logout from the system and recover user account in case of forgetting password.

- To create a user account, the user will provide her/his name, valid phone number and password
- User can login to the system providing phone number and password
- To logout user just has to request to log out from the system
- If the user forgets her/his password, the she can provide her/his phone number which was used to create the user account to the system. Then, the system will send the password to the phone through text message

Action and Reply of Level 1.1.1 Use Case Diagram

- **Action1:** User requests to create an account
- **Reply1:** System provides environment to enter name, phone number and password

- **Action2:** User provides user information and submit for creating user account
- **Reply2:** System stores user information and provided automatic logged in to the system

- **Action3:** User requests to login
- **Reply3:** System provides environment to write phone number and password

- **Action4:** User provides phone number and password
- **Reply4:** System matches provided phone number and password with stored one and provide login permission to the valid user account

- **Action5:** User requests to logout
- **Reply5:** System shows successfully logout message

- **Action6:** User request to recover user account
- **Reply6:** System provides environment to write phone number that was used to create the user account

- **Action7:** User provide phone number to recover user account
- **Reply7:** System match the phone number with the stored one and provide corresponding password through text message

4.4.6 Level 1.1.1 Activity Diagram of Complain Box

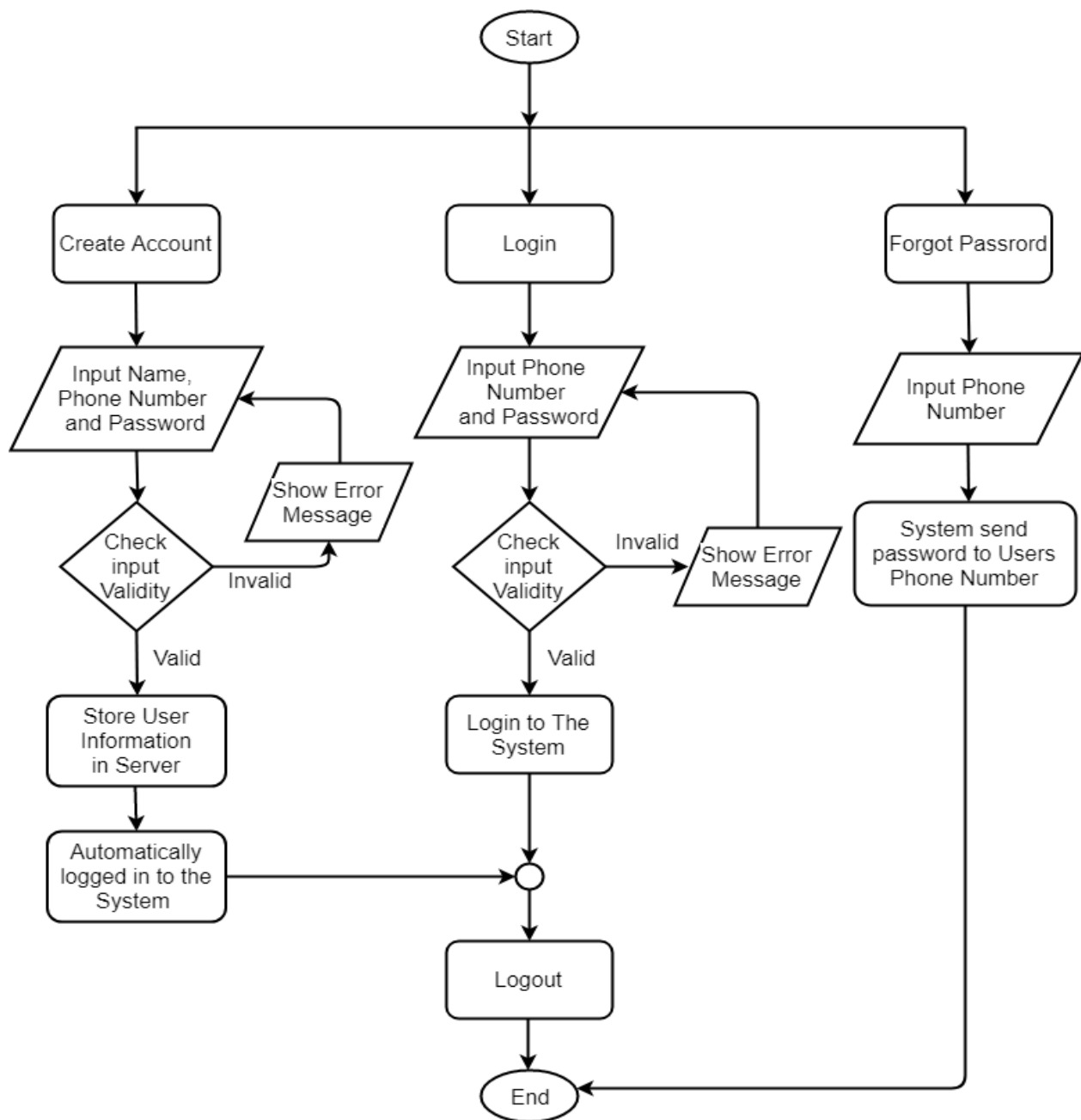


Figure 6: Level 1.1.1 activity diagram of Complain Box

4.4.7 Level 1.1.2 Use Case Diagram of Complain Box

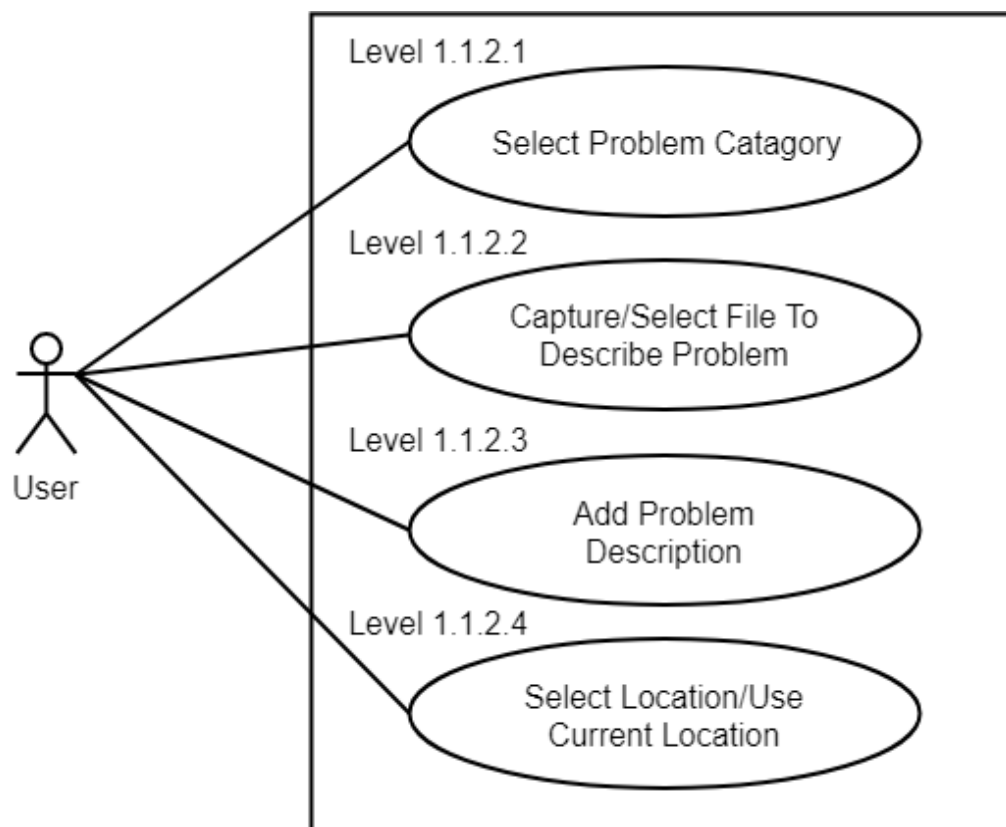


Figure 7: Level 1.1.2 use case diagram of complain box

Table 5: Information about level 1.1.2 use case diagram

Name:	Submit problem
ID:	CB-L-1.1.2
Primary Actor:	User
Secondary Actor:	None

Description of Level 1.1.2 Use Case Diagram

Only user can submit problem. For submitting problem user needs to add problem category, problem describing file, problem description and problem location.

Action and Reply of Level 1.1.2 Use Case Diagram

- **Action1:** User requests to submit problem
- **Reply1:** System provides environment to select problem category, add problem describing files or capture picture, add problem description and problem location sequentially
- **Action2:** User submit problem
- **Reply2:** System stores store problem information

4.4.8 Level 1.1.2 Activity diagram of Complain Box

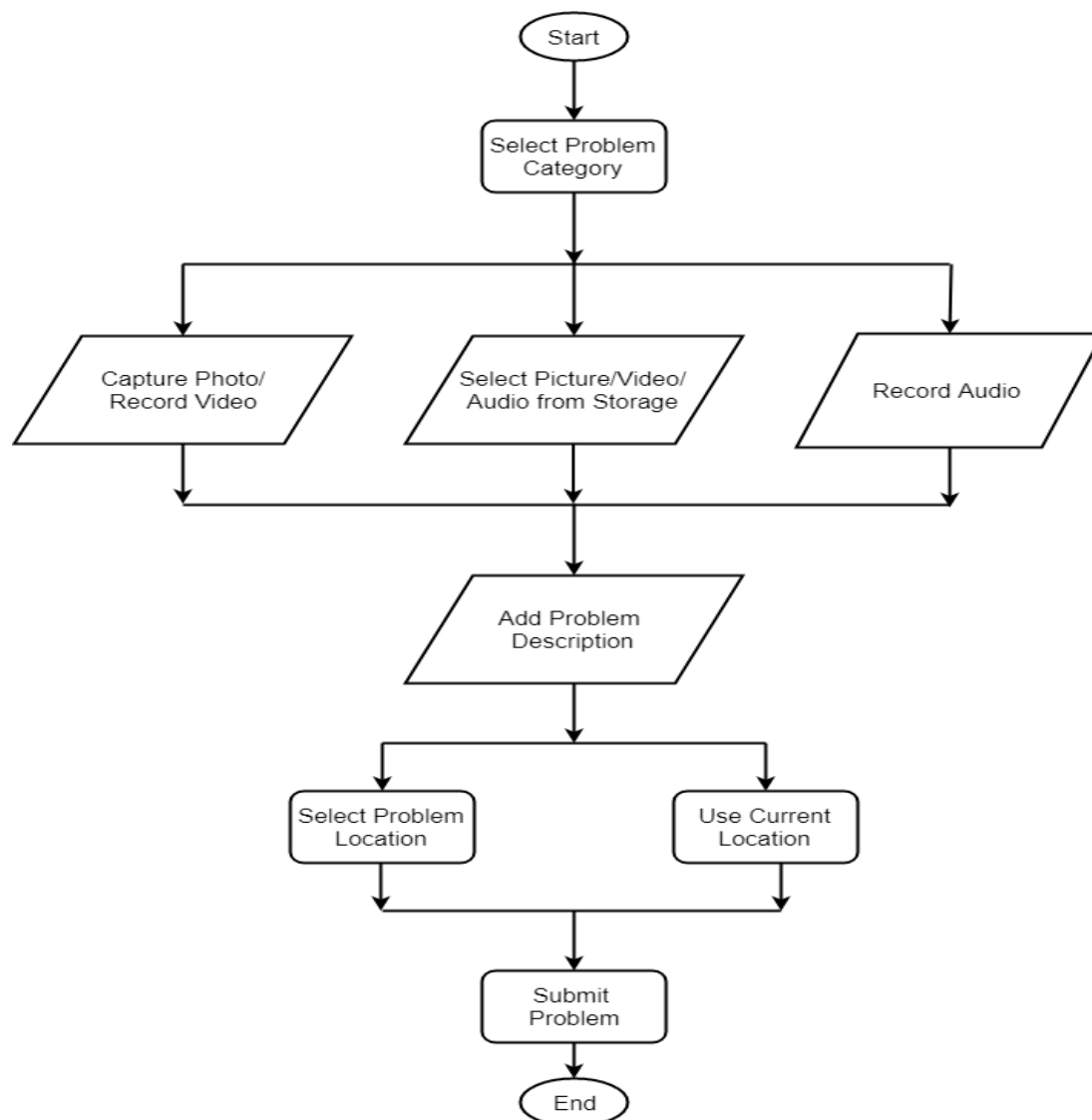


Figure 8: Level 1.1.2 activity diagram of complain box

4.4.9 Level 1.1.4 Use Case Diagram of Complain Box

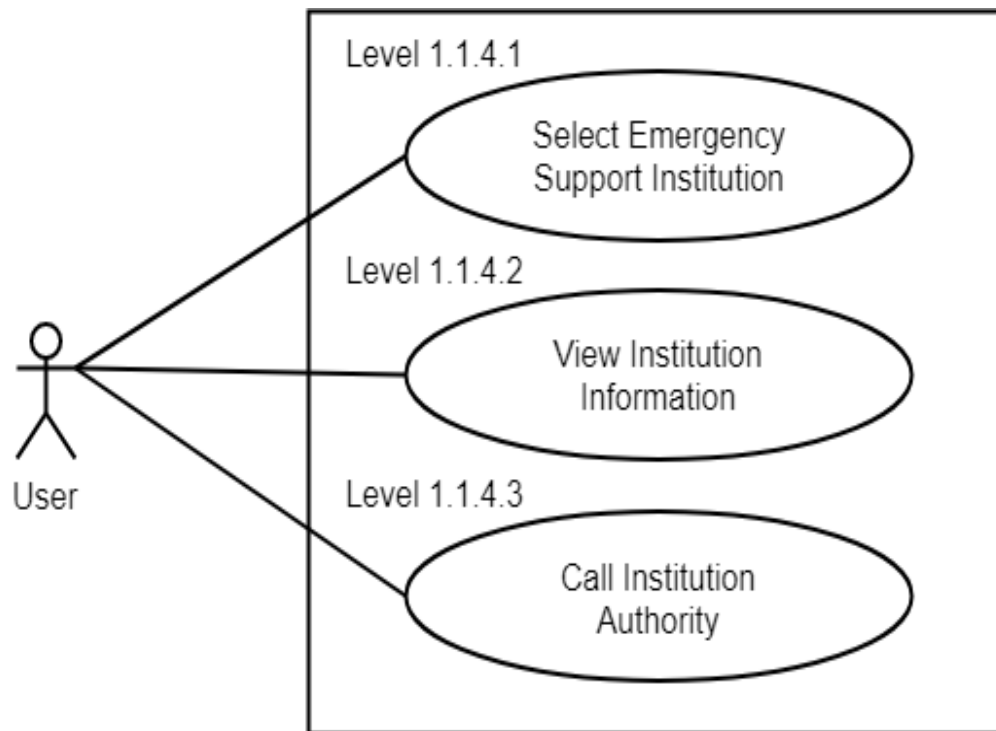


Figure 9: Level 1.1.4 use case diagram of complain box

Table 6: Information about level 1.1.4 use case diagram

Name:	Emergency support
ID:	CB-L-1.1.4
Primary Actor:	User
Secondary Actor:	None

Description of Level 1.1.4 Use Case Diagram

Using emergency user can view emergency support institution's information, location and can make emergency call to a specific institution.

Action and Reply of Level 1.1.4 Use Case Diagram

- **Action1:** User selects a category of emergency support institute
- **Reply1:** System shows all emergency institute of this selected category

- **Action2:** User select an institute
- **Reply2:** System shows the information of that selected institute

- **Action3:** User makes a request to call to the selected institute
- **Reply3:** System provides environment to make a call to that institute

4.4.10 Level 1.1.4 Activity diagram of Complain Box

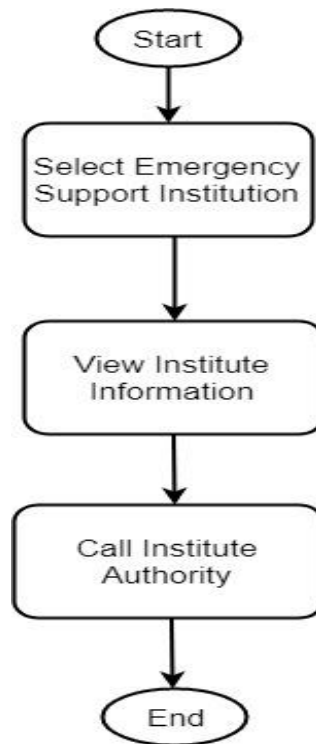


Figure 10: Level 1.1.4 activity diagram of complain box

4.4.11 Level 1.1.6 Use Case Diagram of Complain Box

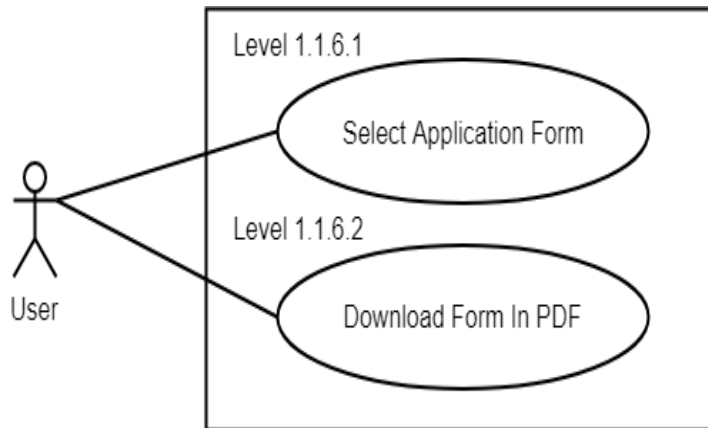


Figure 11: Level 1.1.6 use case diagram of complain box

Table 7: Information about level 1.1.6 use case diagram

Name:	Application form
ID:	CB-L-1.1.6
Primary Actor:	User
Secondary Actor:	None

Description of Level 1.1.6 Use Case Diagram

In this subsystem user can view application forms of DSCC and can download them.

Action and Reply of Level 1.1.6 Use Case Diagram

- **Action1:** User requests to view the information about application form
- **Reply1:** System shows all information of any application

- **Action2:** User requests to download application
- **Reply2:** System provide application form in PDF

4.4.12 Level 1.1.6 Activity Diagram of Complain Box

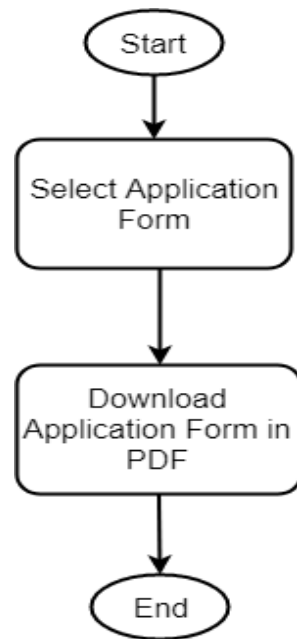


Figure 12: Level 1.1.6 activity diagram of complain box

4.4.13 Level 1.2 Use Case Diagram of Complain Box

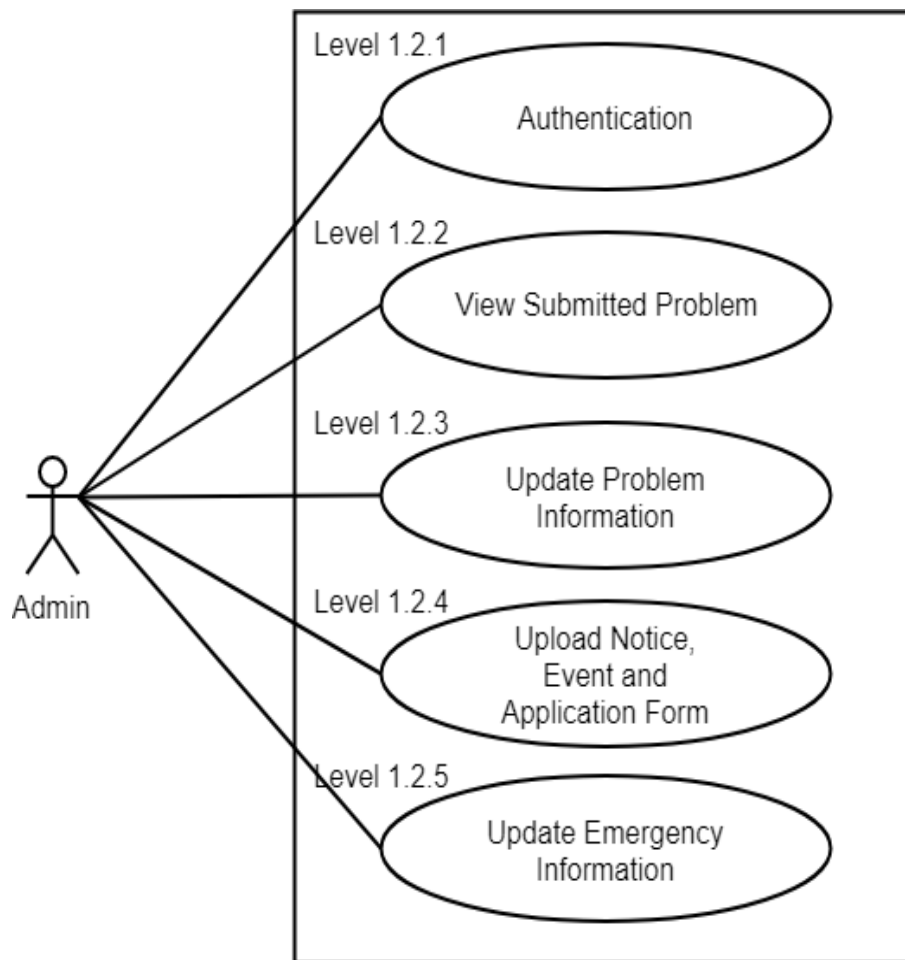


Figure 13: Level 1.2 use case diagram of complain box

Table 8: Information about level 1.2 use case diagram

Name:	Problem supervision
ID:	CB-L-1.2
Primary Actor:	Admin
Secondary Actor:	None

Description of Level 1.2 Use Case Diagram

Only admin can interact with this sub system directly. Admin can

- Authenticate to the system
- View submitted problem
- Update problem information

- Upload event, notice and application form
- Update emergency institute information

4.4.14 Level 1.2 Activity Diagram of Complain Box

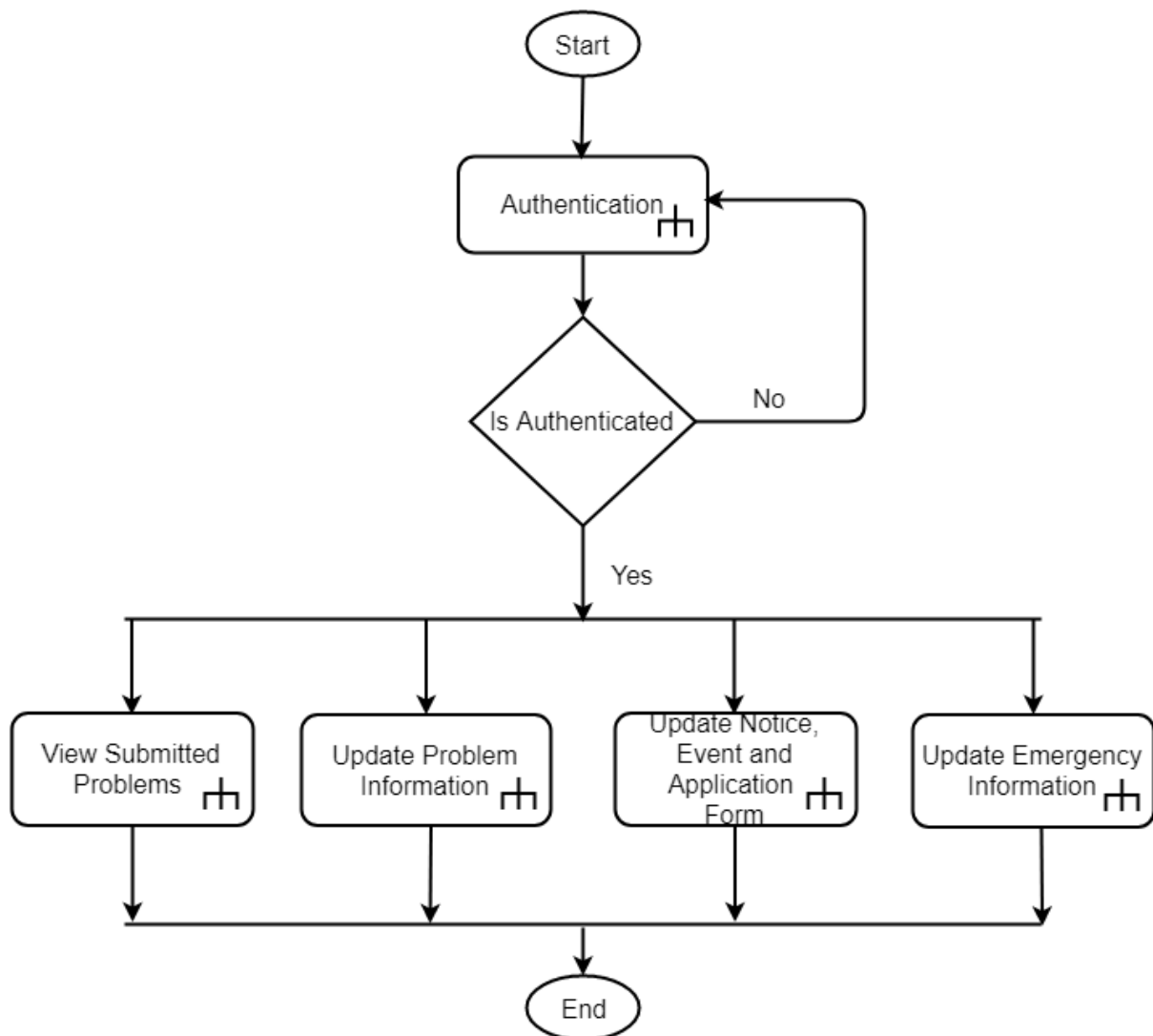


Figure 14: Level 1.2 activity diagram of complain box

4.4.15 Level 1.2.1 Use Case Diagram of Complain Box

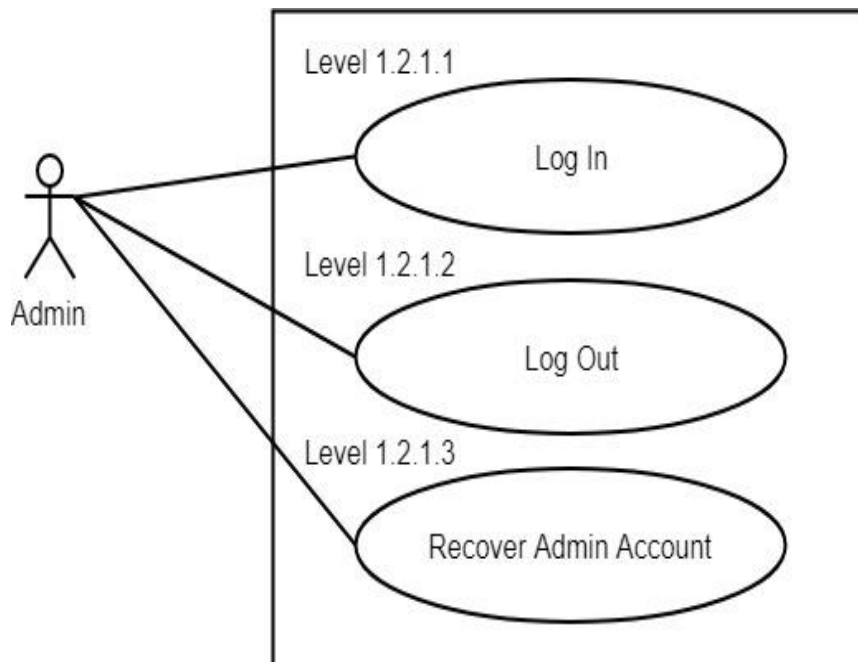


Figure 15: Level 1.2.1 use case diagram of complain box

Table 9: Information about level 1.2.1 use case diagram

Name:	Authentication
ID:	CB-L-1.2.1
Primary Actor:	Admin
Secondary Actor:	None

Description of Level 1.2.1 Use Case Diagram

In this case, admin can login to the system, logout from the system and recover admin account in case of forgetting password.

- Admin can login to the system providing email and password
- To logout admin just has to request to log out from the system
- If admin forgets her/his password, the she can provide her/his email address which was used to create the admin account to the system. Then, the system will send the password to the valid email address

Action and Reply of Level 1.2.1 Use Case Diagram

- **Action1:** Admin requests to login
- **Reply1:** System provides environment to write email and password

- **Action2:** Admin provides email and password
- **Reply2:** System matches provided email and password with stored one and provide login permission
- **Action3:** Admin requests to logout
- **Reply3:** System shows successfully logout message
- **Action4:** Admin request to recover user account
- **Reply4:** System provides environment to write admin email that was used to create the admin account
- **Action5:** Admin provide email address to recover user account
- **Reply5:** System match the email address with the stored one and provide corresponding password through mail

4.4.16 Level 1.2.1 Activity Diagram of Complain Box

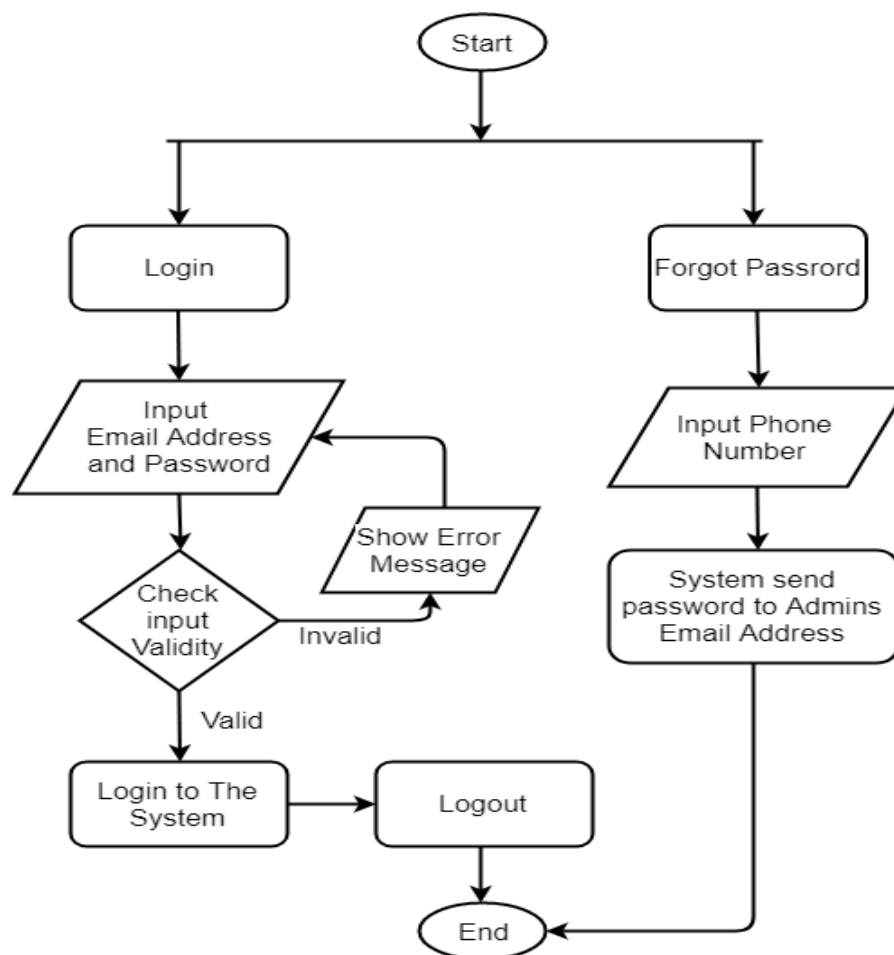


Figure 16: Level 1.2.1 activity diagram of complain box

4.4.17 Level 1.2.2 Use Case Diagram if Complain Box

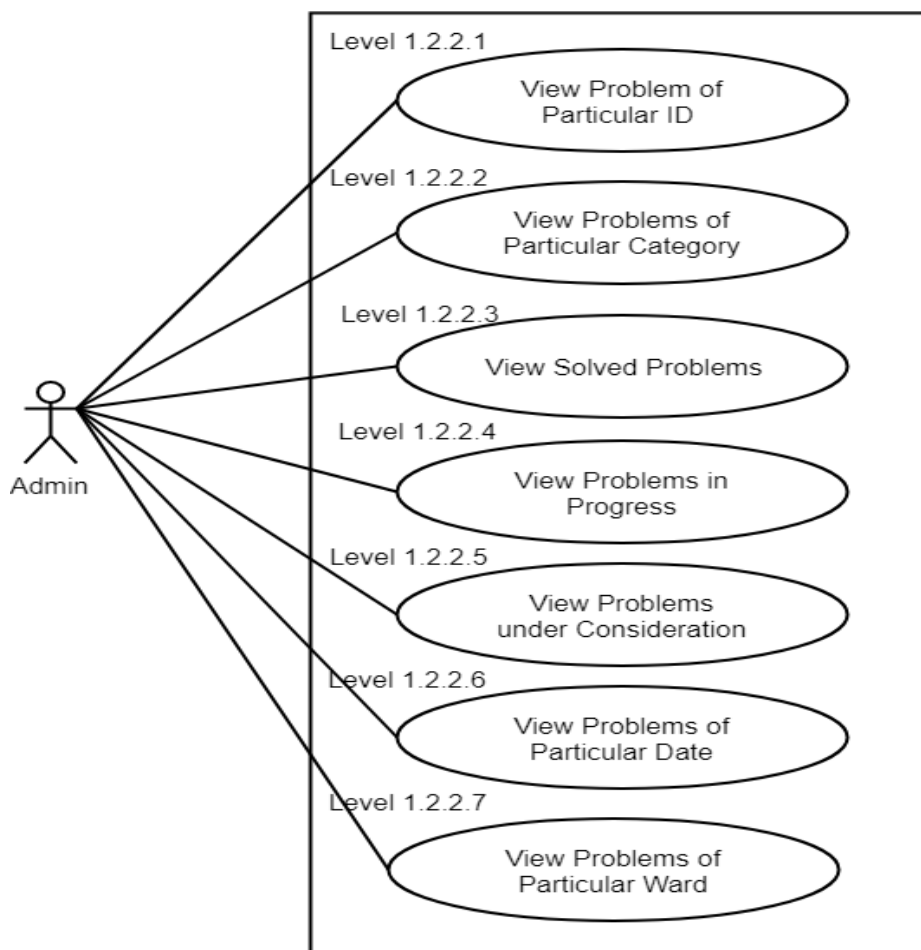


Figure 17: Level 1.2.2 use case diagram of complain box

Table 10: Information about 1.2.2 use case diagram

Name:	View submitted problem
ID:	CB-L-1.2.2
Primary Actor:	Admin
Secondary Actor:	None

Description of Level 1.2.2 Use Case Diagram

In this subsystem, admin can view any type of problems submitted by the users.

Action and Reply of Level 1.2.2 Use Case Diagram

- **Action1:** Admin choose a specific problem type and request to view problem

- **Reply1:** System provides list of required problems and shows to admin

4.4.18 Level 1.2.2 Activity Diagram of Complain Box

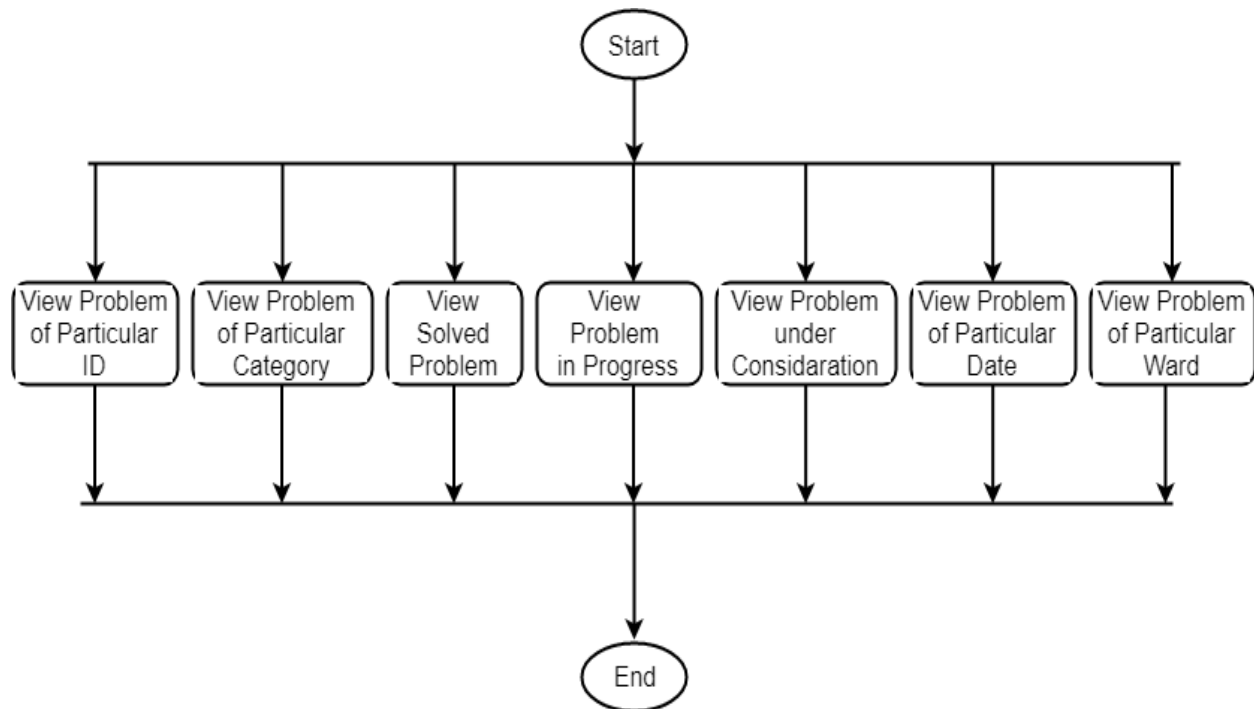


Figure 18: Level 1.2.2 activity diagram of complain box

4.4.19 Level 1.2.3 Use Case Diagram of Complain Box

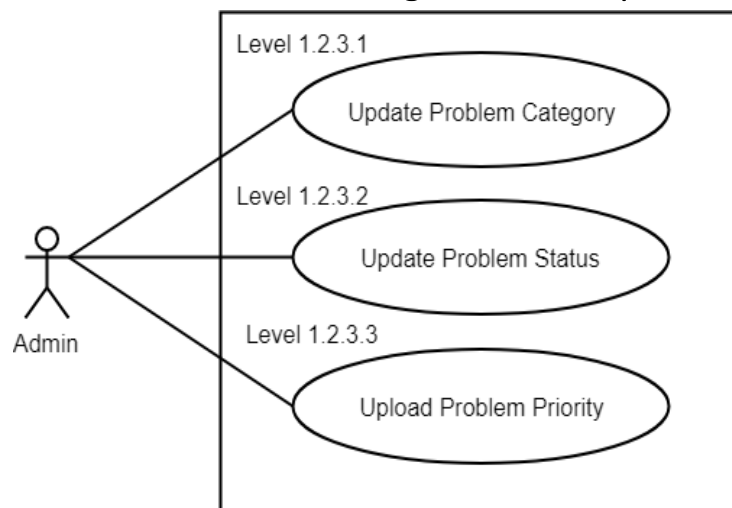


Figure 19: Level 1.2.3 use case diagram of complain box

Table 11: Information about level 1.2.3 use case diagram

Name:	Update problem
ID:	CB-L-1.2.3
Primary Actor:	Admin
Secondary Actor:	None

Description of Level 1.2.3 Use Case Diagram

In this subsystem, admin can change problem category, priority and status.

Action and Reply of Level 1.2.3 Use Case Diagram

- **Action1:** Admin changes problem category
- **Reply1:** System stores updated category

- **Action2:** Admin changes problem priority
- **Reply2:** System stores updated priority

- **Action3:** Admin changes problem status
- **Reply3:** System stores updated status

4.4.20 Level 1.2.3 Activity Diagram of Complain Box

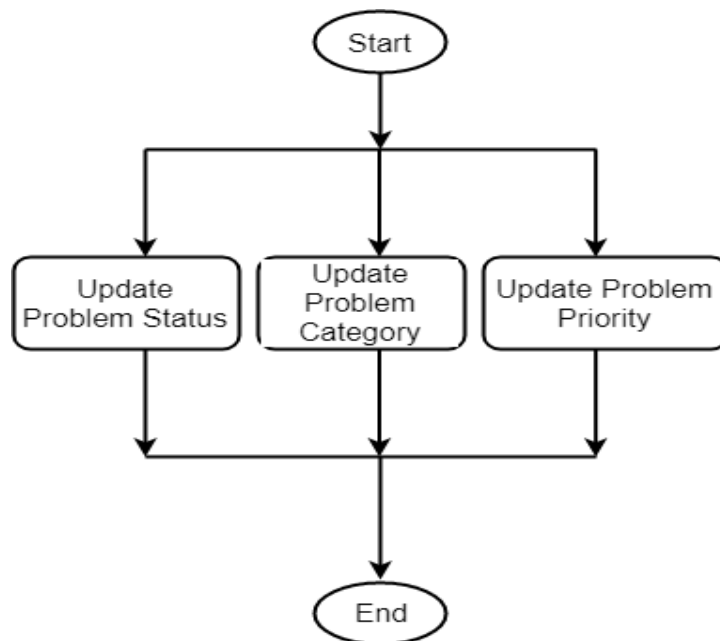


Figure 20: Level 1.2.3 use case diagram of complain box

4.4.21 Level 1.2.4 Use Case Diagram of Complain Box

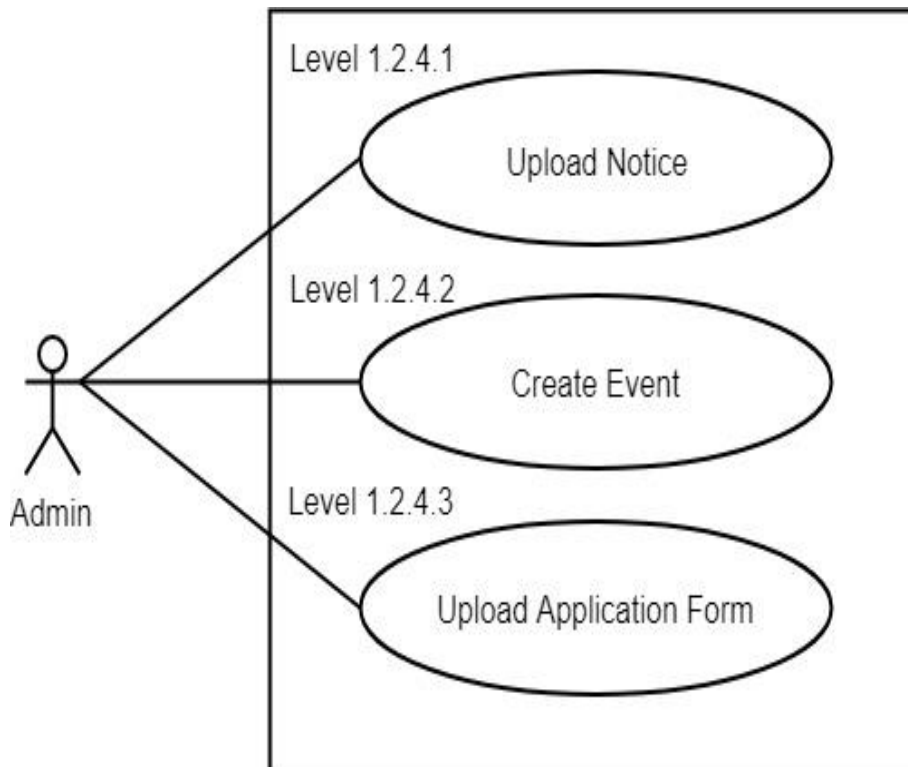


Figure 21: Level 1.2.4 use case diagram of complain box

Table 12: Information about level 1.2.4 use case diagram

Name:	Create notice, event and application form
ID:	CB-L-1.2.4
Primary Actor:	Admin
Secondary Actor:	None

Description of Level 1.2.4 Use Case Diagram

In this subsystem, admin can create notice, event and upload application form.

Action and Reply of Level 1.2.4 Use Case Diagram

- **Action1:** Admin creates notice
- **Reply1:** System stores new notice

- **Action2:** Admin creates event
- **Reply2:** System stores event

- **Action3:** Admin updates application forms
- **Reply3:** System stores updated application forms

4.4.22 Level 1.2.4 Activity Diagram of Complain Box

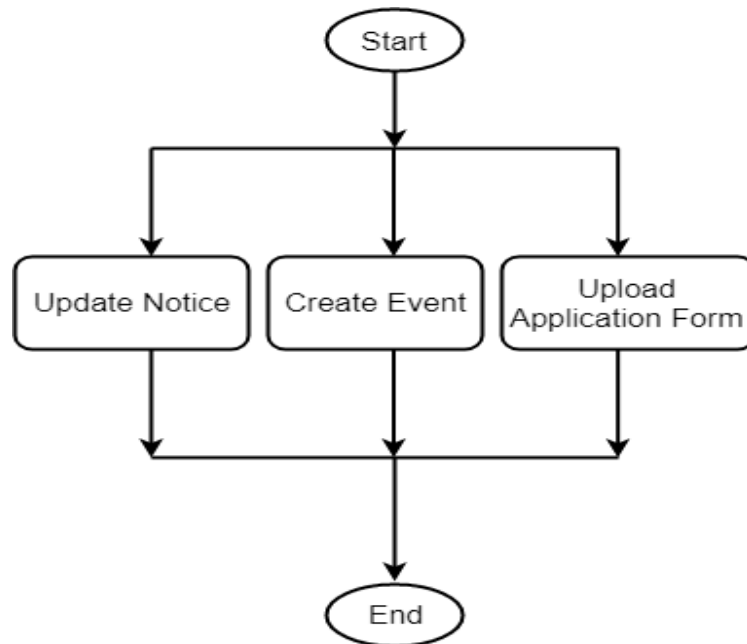


Figure 22: Level 1.2.4 activity diagram of complain box

4.4.23 Level 1.2.5 Use Case Diagram of Complain Box

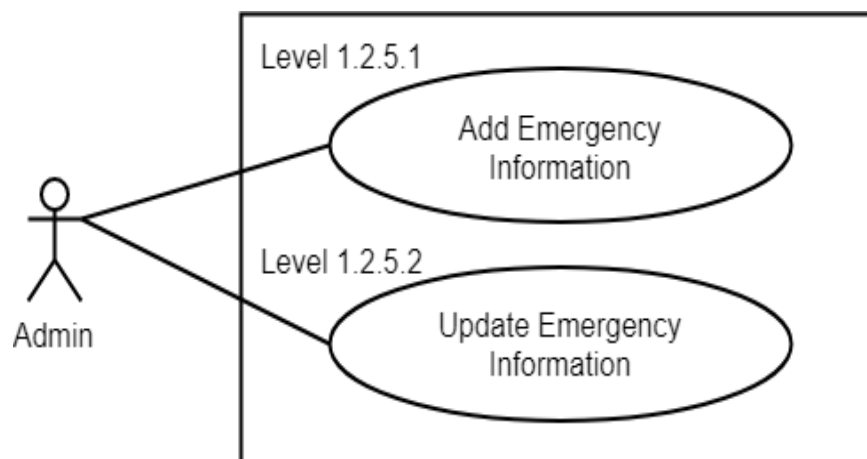


Figure 23: Level 1.2.5 use case diagram of complain box

Table 13: Information about level 1.2.5 use case

Name:	Update institute information
ID:	CB-L-1.2.5
Primary Actor:	Admin
Secondary Actor:	None

Description of Level 1.2.5 Use Case Diagram

In this subsystem, admin can add new emergency institute information and can update them.

Action and Reply of Level 1.2.5 Use Case Diagram

- **Action1:** Admin adds new institute information
- **Reply1:** System stores new institute information

- **Action2:** Admin update institute information
- **Reply2:** System stores new updates

4.4.24 Level 1.2.5 Activity Diagram of Complain Box

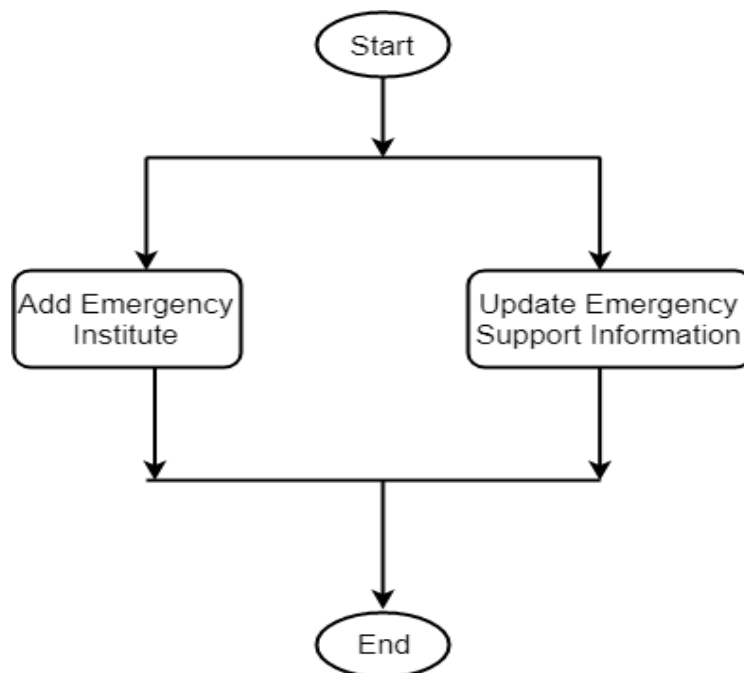


Figure 24: Level 1.2.5 Activity diagram of complain box

Chapter 5: Data Based Modeling

This chapter describes the Data Based Model for our project “Complain Box”.

5.1 Introduction

In our software requirements the creation, extension and interaction with the database is also included. So, we choose to create data models as a part of overall requirements modelling.

5.2 Data Object Selection

The data objects in this chapter represent the information of our “Complain Box” which has different properties or attributes that must be understood by our system, “Complain Box”. Here the table of data object-

5.2.1 Noun Identification

We identified all nouns from our usage scenario whether they are in problem space or solution space.

Table 14: Noun identification

Serial Number	Nouns	Problem/ Solution Space	Attributes
1	Dhaka City Corporation	p	
2	User	s	5, 6, 7, 48
3	Account	s	
4	System	p	
5	Name	s	
6	Phone number	s	
7	Password	s	
8	Five	p	
9	Character	p	
10	Fifteen	p	
11	Subsystem	p	
12	Information	p	
13	Log	p	
14	Message	p	
15	Problem Category	s	
16	Solid Waste	p	

17	Road	p	
18	Drainage	p	
19	Municipal Property	p	
20	Road light	p	
21	Illegal parking	p	
22	Mosquito problem	p	
23	Health Department	p	
24	Revenue Department	p	
25	Social Welfare Department	p	
26	Planning Department	p	
27	Miscellaneous Complains	p	
28	Establishment Sections	p	
29	Encroachment	p	
30	Factories	p	
31	License	p	
32	Corruption	p	
33	Footpath	p	
34	Drain	p	
35	Picture	p	
36	Video	p	
37	Audio	p	
38	File	p	
39	Storage	p	
40	Problem	s	15, 41, 42, 49, 50, 51, 52, 53, 54
41	Description	s	
42	Problem Location	s	43, 44
43	Latitude	s	
44	Longitude	s	
45	Current Location	p	
46	Area	p	
47	User name	s	
48	Reward point	s	
49	Problem ID	s	
50	Submission date	s	
51	Submission time	s	
52	Status	s	
53	Problem priority	s	
54	Ward number	s	
55	Communication facility	p	
56	Hospital	p	
57	Fire station	p	

58	Police station	p	
59	Name of institute	s	
60	Notice	s	61, 63
61	Notice content	s	
62	Event name	s	
63	Publishing date	s	
64	Event date	s	
65	Event time	s	
66	Application form	s	
67	Passport application form	p	
68	E-income tax certificate	p	
69	Driving license form	p	
70	Birth certificate	p	
71	Trade license form	p	
72	Tax form	p	
73	Death certificate	p	
74	Inherit certificate	p	
75	Admin	s	7, 79
76	Person	p	
77	Decision	p	
78	Authentication	p	
79	Email address	s	
80	Developer	p	
81	Scope	p	
82	PDF	p	
83	Event description	s	
84	Institution	s	85, 86, 87
85	Institution name	s	
86	Institution location	s	
87	Institution phone number	s	
88	Event	s	62, 63, 64, 65, 83

5.2.2 Potential Data Objects

- **User:** 5, 6, 7, 48
- **Problem:** 15, 41, 42, 49, 50, 51, 52, 53, 54
- **Location:** 43, 44
- **Admin:** 79, 7
- **Institute:** 85, 86, 87
- **Notice:** 61, 63
- **Event:** 62, 63, 64, 65, 83

5.2.3 Analysis for Finalizing Data Objects

- **User** contains user information who submits problems
- **Problems** holds the information which describes a specific problem
- **Location** contains longitude and latitude of a specific problem or an institution
- **Institute** keeps the information of emergency support institutions
- **Notice** holds the publishing date and the content of notice
- **Event** contains event information

5.2.4 Final Data Objects

Table 15: Final data objects of complain box

No	Entity	Attributes
1	User	name, phone_number, reward_point, password
2	Problem	problem_ID, problem_description, problem_category, submission_date, submission_time, status, problem_priority, ward_number, visibility
3	Location	latitude, longitude, location_ID
4	Admin	email, password
5	Institution	institute_name, institute_location, institute_phone_number, institution_category
6	Notice	notice_content, notice_id, notice_publishing_date
7	Event	event_name, event_id, event_publishing_date, event_date, event_time, event_description

5.3 Data Object Relations

In our “Complain Box” data objects are connected to one another in the ways stated below.

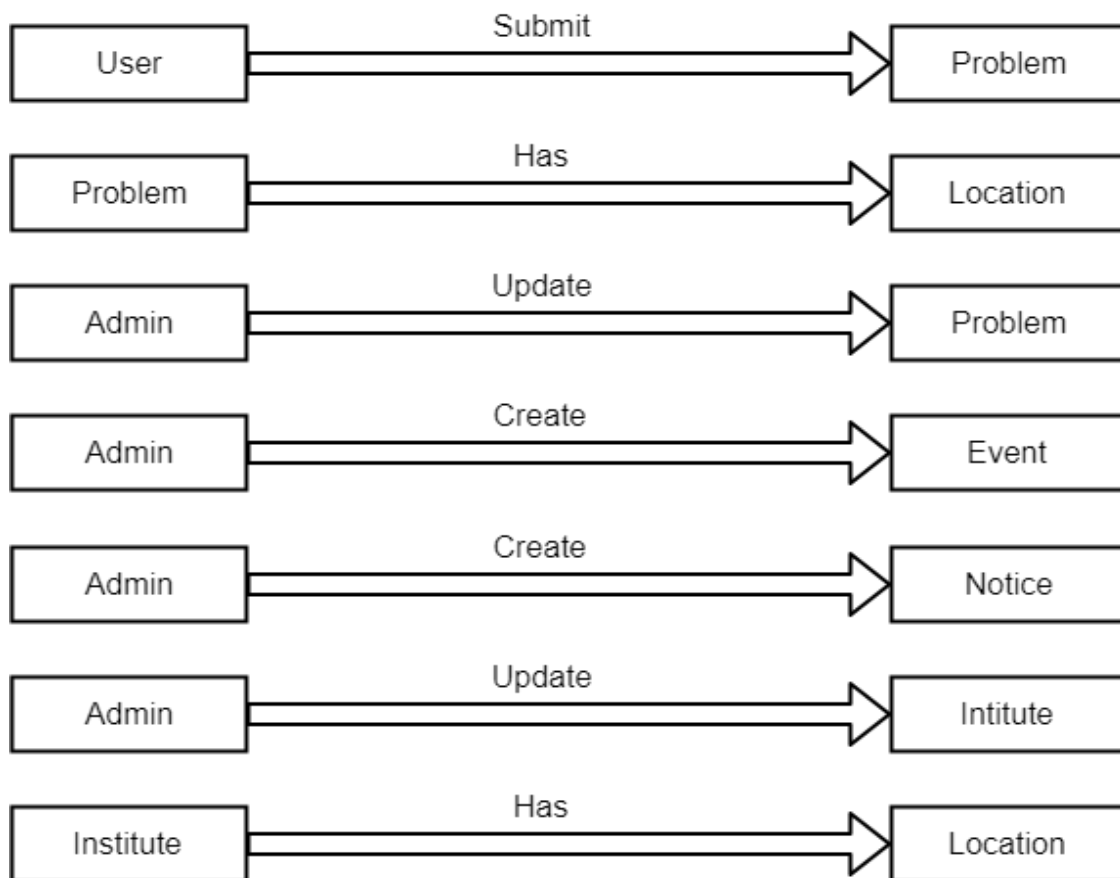


Figure 25: Data object relationship of complain box

5.4 Entity Relationship Diagram

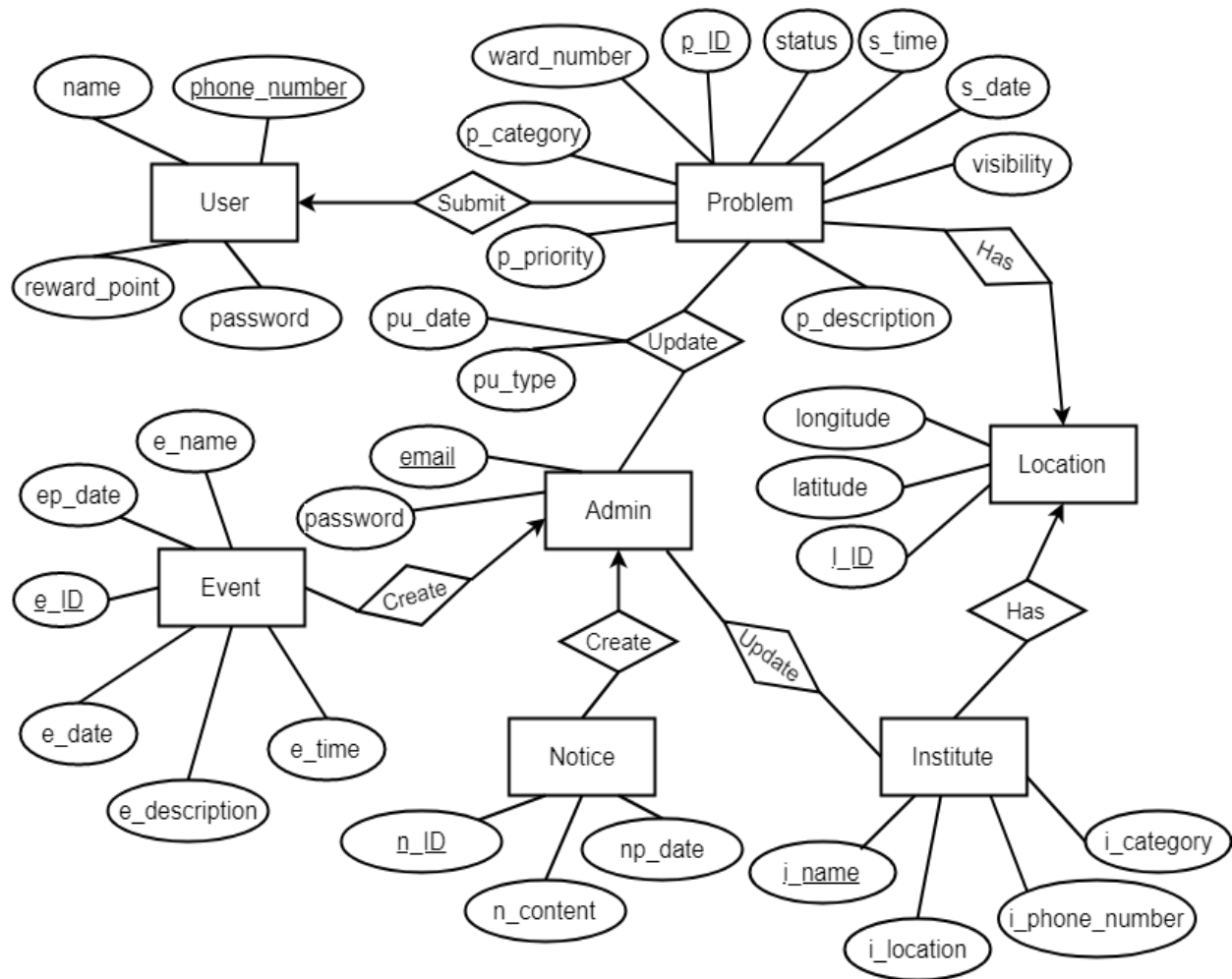


Figure 26: Entity relationship diagram of complain box

5.5 Schema Diagram

Table 16: Schema table of User data object

User		
Attribute	Type	Size
name	varchar2	20
<u>phone_number</u>	varchar2	14
reward_point	number	5
password	varchar2	15

Table 17: Schema table of Problem data object

Problem		
Attribute	Type	Size
p_priority	number	5
p_category	varchar2	20
ward_number	number	5
<u>p_ID</u>	varchar2	20
status	varchar2	20
s_time	varchar2	20
s_date	varchar2	20
visibility	varchar2	20
p_description	varchar2	500
<u>phone_number</u>	varchar2	14
<u>I_ID</u>	varchar2	20

Table 18: Schema table of Location data object

Location		
Attribute	Type	Size
<u>I_ID</u>	varchar2	20
latitude	varchar2	20
longitud	varchar2	20

Table 19: Schema table of Institute data object

Institute		
Attribute	Type	Size
<u>i_name</u>	varchar2	50
i_location	varchar2	100
i_phone_number	varchar2	14
i_category	varchar2	20
<u>I_ID</u>	varchar2	20

Table 20: Schema table of Admin data object

Admin		
Attribute	Type	Size
email	varchar2	20
password	varchar2	20

Table 21: Schema table of Event data object

Event		
Attribute	Type	Size
e_name	varchar2	50
ep_date	varchar2	20
e_date	varchar2	20
e_description	varchar2	500
e_time	varchar2	20
<u>email</u>	varchar2	20
<u>e_ID</u>	varchar2	20

Table 22: Schema table of Update_Problem data object

Update_Problem		
Attribute	Type	Size
<u>p_ID</u>	varchar2	20
<u>email</u>	varchar2	20
pu_date	varchar2	20
pu_type	varchar2	20

Table 23: Schema table of Update_Institute data object

Update_Institute		
Attribute	Type	Size
<u>email</u>	varchar2	20
<u>i_name</u>	varchar2	50

Chapter 6: Class Based Modeling

We intended this chapter to describe class based modeling for our “Complain Box”.

6.1 Introduction

In this chapter, our designed class based model represents the objects that our “Complain Box” will manipulate, the operation that will applied to the objects, relationships between and the collaboration that occur between the classes that are defined.

6.2 Grammatical Parsing and Class Analysis

To analyze class for our system, we first parse all the determined nouns from our usage scenario. Then we categorize them with general classification and selection criteria. We identify potential classes by identifying the nouns from the scenery. Then we compared those with the following criteria whether they match or not. We noted down the number of the fulfilled criteria at the right column.

6.2.1 General Classification

To identify the potential class, we have to first select the nouns from the solution space of the story. These were then characterized in seven general classifications. The seven general characteristics are as follows:

1. External entities
2. Things
3. Events
4. Roles
5. Organizational units
6. Places
7. Structures

Table 24: Class identification with general classification

Number of Nouns	Nouns	Problem/Solution Space	General Classification
1	Dhaka City Corporation	p	
2	User	s	1, 4, 5, 7
3	Account	s	2
4	System	p	
5	Name	s	2

6	Phone number	s	2
7	Password	s	2
8	Five	p	
9	Character	p	
10	Fifteen	p	
11	Subsystem	p	
12	Information	p	
13	Log	p	
14	Message	p	
15	Problem Category	s	2
16	Solid Waste	p	
17	Road	p	
18	Drainage	p	
19	Municipal Property	p	
20	Road light	p	
21	Illegal parking	p	
22	Mosquito problem	p	
23	Health Department	p	
24	Revenue Department	p	
25	Social Welfare Department	p	
26	Planning Department	p	
27	Miscellaneous Complains	p	
28	Establishment Sections	p	
29	Encroachment	p	
30	Factories	p	
31	License	p	
32	Corruption	p	
33	Footpath	p	
34	Drain	p	
35	Picture	p	
36	Video	p	
37	Audio	p	
38	File	p	
39	Storage	p	
40	Problem	s	2, 4, 7
41	Description	s	2
42	Problem Location	s	2
43	Latitude	s	2
44	Longitude	s	2
45	Current Location	p	
46	Area	p	

47	User name	s	2
48	Reward point	s	2
49	Problem ID	s	2
50	Submission date	s	2
51	Submission time	s	2
52	Status	s	2
53	Problem priority	s	2
54	Ward number	s	2
55	Communication facility	p	
56	Hospital	p	
57	Fire station	p	
58	Police station	p	
59	Name of institute	s	2
60	Notice	s	2, 3
61	Notice content	s	2
62	Event name	s	2
63	Publishing date	s	2
64	Event date	s	2
65	Event time	s	2
66	Application form	s	2, 4, 7
67	Passport application form	p	
68	E-income tax certificate	p	
69	Driving license form	p	
70	Birth certificate	p	
71	Trade license form	p	
72	Tax form	p	
73	Death certificate	p	
74	Inherit certificate	p	
75	Admin	s	1, 3, 4, 7
76	Person	p	
77	Decision	p	

6.2.2 Selection Criteria

The nouns having two or more than two were selected from the general classification list. After that step, we compared them with the following criteria list. Those are

1. Retained information
2. Needed services
3. Multiple attributes

4. Common attributes
5. Common operations
6. Essential requirements

Table 25: Class identification with selection criteria

Number of Nouns	Nouns	Selection Classification
1	User	1, 3
2	Problem	1, 2, 3, 4, 5
3	Application form	1, 2, 3
4	Admin	1, 3
5	Institution	1, 2, 3, 4, 5
6	Notice	1, 3, 4, 5
7	Event	1, 3, 4, 5

6.2.3 Preliminary Classes

From above table, we have taken all the noun who passed a single criteria. From further analysis we also got, Location is also need for our system as a class. So, these are the candidate classes who are selected primarily:

1. User
2. Problem
3. Admin
4. Application form
5. Institute
6. Notice
7. Event
8. Location

Besides, we divided our system as two individual subsystem (Problem submission and Problem supervision). Thus, we designed our system two different part working together through a common server. So, our each subsystem holds its own set of classes.

Problem submission is designed with:

1. User
2. Problem
3. Application form
4. Institute
5. Notice
6. Event
7. Location

Problem Supervision is designed with:

1. Problem
2. Admin
3. Application form
4. Institute
5. Notice
6. Event
7. Location

6.3 Associated Noun and Verb Identification

For identifying attributes and methods we choose to parse associate nouns and verbs of a class from our usage scenario.

Table 26: Associated noun and verb identification

Number of Nouns	Potential Classes	Nouns	Verbs
1	User	name, phone number, reward points, password	create, log in, log out, view, send, capture, record, select, contain, recover
2	Problem	problem ID, problem description, problem category, problem priority, submission date, submission time, problem status, ward number, visibility, location	submit, search, view, update, face, change, contain,
3	Application form	application ID, file path	add, view, update, download, include
4	Admin	email, password	log in, log out, send, operate, decide, create, request, view, edit, select, update, upload, change, recover
5	Institution	institute name, institute location, institute phone number, institution category	add, update, view, call
6	Notice	notice ID, notice content, notice publishing date	create, view
7	Location	latitude, longitude	view

8	Event	event ID, event name, event publishing date, event date, event time, event description	create, view
---	-------	----------------------------------------------------------------------------------------	--------------

6.4 Attribute Selection

From our selected nouns associated with individual class we finalize attribute for each class.

Table 27: Attribute selection form selected classes

Number of Nouns	Potential Classes	Attributes
1	User	<ul style="list-style-type: none"> ➤ name ➤ phoneNumber ➤ rewardPoints ➤ password
2	Problem	<ul style="list-style-type: none"> ➤ problemID ➤ problemDescription ➤ problemCategory ➤ problemPriorit ➤ submissionDate ➤ submissionTime ➤ problemStatus ➤ wardNumber ➤ visibility ➤ location
3	Application form	<ul style="list-style-type: none"> ➤ applicationID ➤ filePath
4	Admin	<ul style="list-style-type: none"> ➤ email ➤ password
5	Institution	<ul style="list-style-type: none"> ➤ instituteName ➤ instituteLocation ➤ institutePhoneNumber ➤ institutionCategory
6	Notice	<ul style="list-style-type: none"> ➤ noticeID ➤ noticeContent ➤ noticePublishingDate
7	Location	<ul style="list-style-type: none"> ➤ locationID ➤ latitude ➤ longitude
8	Event	<ul style="list-style-type: none"> ➤ eventID ➤ eventName ➤ eventPublishingDate

		<ul style="list-style-type: none"> ➤ eventDate ➤ eventTime ➤ eventDescription
--	--	----------------------------------------------------------------------------------------------------------------

6.5 Method Selection

From our selected nouns associated with individual class we finalize methods for each class.

Table 28: Method selection form selected classes

Number of Nouns	Potential Classes	Methods
1	User	<ul style="list-style-type: none"> ➤ createAccount() ➤ logIn() ➤ logOut() ➤ viewUserInformation() ➤ recoverPassword()
2	Problem	<ul style="list-style-type: none"> ➤ submitProblem() ➤ serachProblem() ➤ viewProblem() ➤ viewProblemLocation() ➤ updatePriority() ➤ updateStatus() ➤ updateCategory() ➤ capturePicture() ➤ selectProblemDescribingFile() ➤ addProblemDescription() ➤ addProblemLocation() ➤ selectProblemCategory()
3	Application form	<ul style="list-style-type: none"> ➤ addApplicationForm() ➤ viewApplicationForm() ➤ updateApplicationForm() ➤ downoadApplicationForm()
4	Admin	<ul style="list-style-type: none"> ➤ logIn() ➤ logOut() ➤ recoverAccount() ➤ updateSystemInformation()
5	Institution	<ul style="list-style-type: none"> ➤ addInstitute() ➤ updateInstitute() ➤ viewInstituteInsformation() ➤ viewInstituteMap() ➤ callToInstitute()
6	Notice	<ul style="list-style-type: none"> ➤ createNotice()

		➤ viewNotice()
7	Location	➤ viewLocation()
8	Event	➤ createEvent() ➤ viewEvent()

6.6 Finalized Classes

From our identified attributes and methods, we finalized classes for our both sub systems.

6.6.1 Finalized classes for Problem Submission

Table 29: Class of problem submission (User)

User	
Attributes	Methods
<ul style="list-style-type: none"> ➤ name ➤ phoneNumber ➤ rewardPoints ➤ password 	<ul style="list-style-type: none"> ➤ createAccount() ➤ logIn() ➤ logOut() ➤ viewUserInformation() ➤ createProblemInformation() ➤ recoverPassword()

Table 30: Class of problem submission (Problem)

Problem	
Attributes	Methods
<ul style="list-style-type: none"> ➤ problemID ➤ problemDescription ➤ problemCategory ➤ problemPriorit ➤ submissionDate ➤ submissionTime ➤ problemStatus ➤ wardNumber ➤ visibility ➤ location 	<ul style="list-style-type: none"> ➤ submitProblem() ➤ serachProblem() ➤ viewProblem() ➤ capturePicture() ➤ selectProblemDescribingFile() ➤ addProblemDescription() ➤ addProblemLocation() ➤ selectProblemCategory()

Table 31: Class of problem submission (ApplicationForm)

ApplicationForm	
Attributes	Methods
<ul style="list-style-type: none"> ➤ applicationID ➤ filePath 	<ul style="list-style-type: none"> ➤ viewApplicationForm() ➤ downloadApplicationForm()

Table 32: Class of problem submission (Institution)

Institution	
Attributes	Methods
<ul style="list-style-type: none"> ➤ instituteName ➤ instituteLocation ➤ institutePhoneNumber ➤ institutionCategory 	<ul style="list-style-type: none"> ➤ viewInstituteInformation() ➤ callToInstitute()

Table 33: Class of problem submission (Notice)

Notice	
Attributes	Methods
<ul style="list-style-type: none"> ➤ noticeID ➤ noticeContent ➤ noticePublishingDate 	<ul style="list-style-type: none"> ➤ viewNotice()

Table 34: Class of problem submission (Event)

Event	
Attributes	Methods
<ul style="list-style-type: none"> ➤ eventID ➤ eventName ➤ eventPublishingDate ➤ eventDate ➤ eventTime ➤ eventDescription 	<ul style="list-style-type: none"> ➤ viewEvent()

Table 35: Class of problem submission (Location)

Location	
Attributes	Methods
<ul style="list-style-type: none"> ➤ longitude ➤ Latitude ➤ locationID 	<ul style="list-style-type: none"> ➤ viewLocation()

6.6.2 Finalized classes for Problem Supervision

Table 36: Class of problem supervision (Problem)

Problem	
Attributes	Methods
<ul style="list-style-type: none"> ➤ problemID ➤ problemDescription ➤ problemCategory ➤ problemPriorit ➤ submissionDate ➤ submissionTime ➤ problemStatus ➤ wardNumber ➤ visibility ➤ location 	<ul style="list-style-type: none"> ➤ serachProblem() ➤ viewProblem() ➤ updatePriority() ➤ updateStatus() ➤ updateCategory()

Table 37: Class of problem supervision (ApplicationForm)

ApplicationForm	
Attributes	Methods
<ul style="list-style-type: none"> ➤ applicationID ➤ filePath 	<ul style="list-style-type: none"> ➤ addApplicationForm() ➤ viewApplicationForm() ➤ updateApplicationForm() ➤ downloadApplicationForm()

Table 38: Class of problem supervision (Admin)

Admin	
Attributes	Methods
<ul style="list-style-type: none"> ➤ email ➤ password 	<ul style="list-style-type: none"> ➤ logIn() ➤ logOut() ➤ updateSystemInformation() ➤ recoverAccount()

Table 39: Class of problem supervision (Institution)

Institution	
Attributes	Methods
<ul style="list-style-type: none"> ➤ instituteName ➤ instituteLocation ➤ institutePhoneNumber ➤ institutionCategory 	<ul style="list-style-type: none"> ➤ addInstitute() ➤ updateInstitute() ➤ viewInstituteInformation()

Table 40: Class of problem supervision (Notice)

Notice	
Attributes	Methods
<ul style="list-style-type: none"> ➤ noticeID ➤ noticeContent ➤ noticePublishingDate 	<ul style="list-style-type: none"> ➤ createNotice() ➤ viewNotice()

Table 41: Class of problem supervision (Event)

Event	
Attributes	Methods
<ul style="list-style-type: none"> ➤ eventID ➤ eventName ➤ eventPublishingDate ➤ eventDate ➤ eventTime ➤ eventDescription 	<ul style="list-style-type: none"> ➤ createEvent() ➤ viewEvent()

Table 42: Class of problem supervision (Location)

Location	
Attributes	Methods
<ul style="list-style-type: none"> ➤ longitude ➤ Latitude ➤ locationID 	<ul style="list-style-type: none"> ➤ viewLocation()

6.7 Class Responsibility Collaborator Modeling

In this stage, we identify the responsibility and collaborators of each class. It will provide a simple means for identifying and organizing the classes that are relevant to our system requirements. We modeled this for both subsystem of our “Complain Box”.

6.7.1 Class Responsibility Collaborator Modeling for Problem Submission

Figure 27: CRC diagram of User

Class: User	
Responsibilities:	Collaborators:
Stores user information	Problem
Authenticates to the system	
Creates problem information	Problem
Shows user detail information	
Recovers Password	

Figure 28: CRC diagram of Problem

Class: Problem	
Responsibilities:	Collaborators:
Stores problem information	
Shows problem detail information	Location

Figure 29: CRC diagram of ApplicationForm

Class: ApplicationForm	
Responsibilities:	Collaborators:
Downloads application form	
Shows information of application form	

Figure 30: CRC diagram of Institution

Class: Institution	
Responsibilities:	Collaborators:
Shows institute information	Location
Makes a call to institute	

Figure 31: CRC diagram of

Class: Notice	
Responsibilities:	Collaborators:
Shows notice information	

Figure 32: CRC diagram of Event

Class: Event	
Responsibilities:	Collaborators:
Shows event information	

Figure 33: CRC diagram of Location

Class: Location	
Responsibilities:	Collaborators:
Shows location in map	

6.7.2 Class Responsibility Collaborator Modeling for Problem Supervision

Figure 34: CRC diagram of Problem

Class: Problem	
Responsibilities:	Collaborators:
Shows problem detail information	Location
Updates problem Information	

Figure 35: CRC diagram of ApplicationForm

Class: ApplicationForm	
Responsibilities:	Collaborators:
Downloads application form	
Shows information of application form	
Updates application form	

Figure 36: CRC diagram of Admin

Class: Admin	
Responsibilities:	Collaborators:
Authenticates to the system	
Updates system information	Problem, Event, Notice, ApplicationForm, Institution

Figure 37: CRC diagram of Institution

Class: Institution	
Responsibilities:	Collaborators:
Shows institute information	Location
Updates institute information	

Figure 38: CRC diagram of Notice

Class: Notice	
Responsibilities:	Collaborators:
Shows notice information	
Adds notices	

Figure 39: CRC diagram of Event

Class: Event	
Responsibilities:	Collaborators:
Show event information	
Add event	

Figure 40: CRC diagram of Location

Class: Location	
Responsibilities:	Collaborators:
Shows location in map	

6.8 Class Diagram

In this stage we designed class diagram in the Unified Modeling Language. This is a type of static diagram to describe the structure of our system. Here we also designed two individual design for our two subsystems.

6.8.1 Class Diagram of Problem Submission

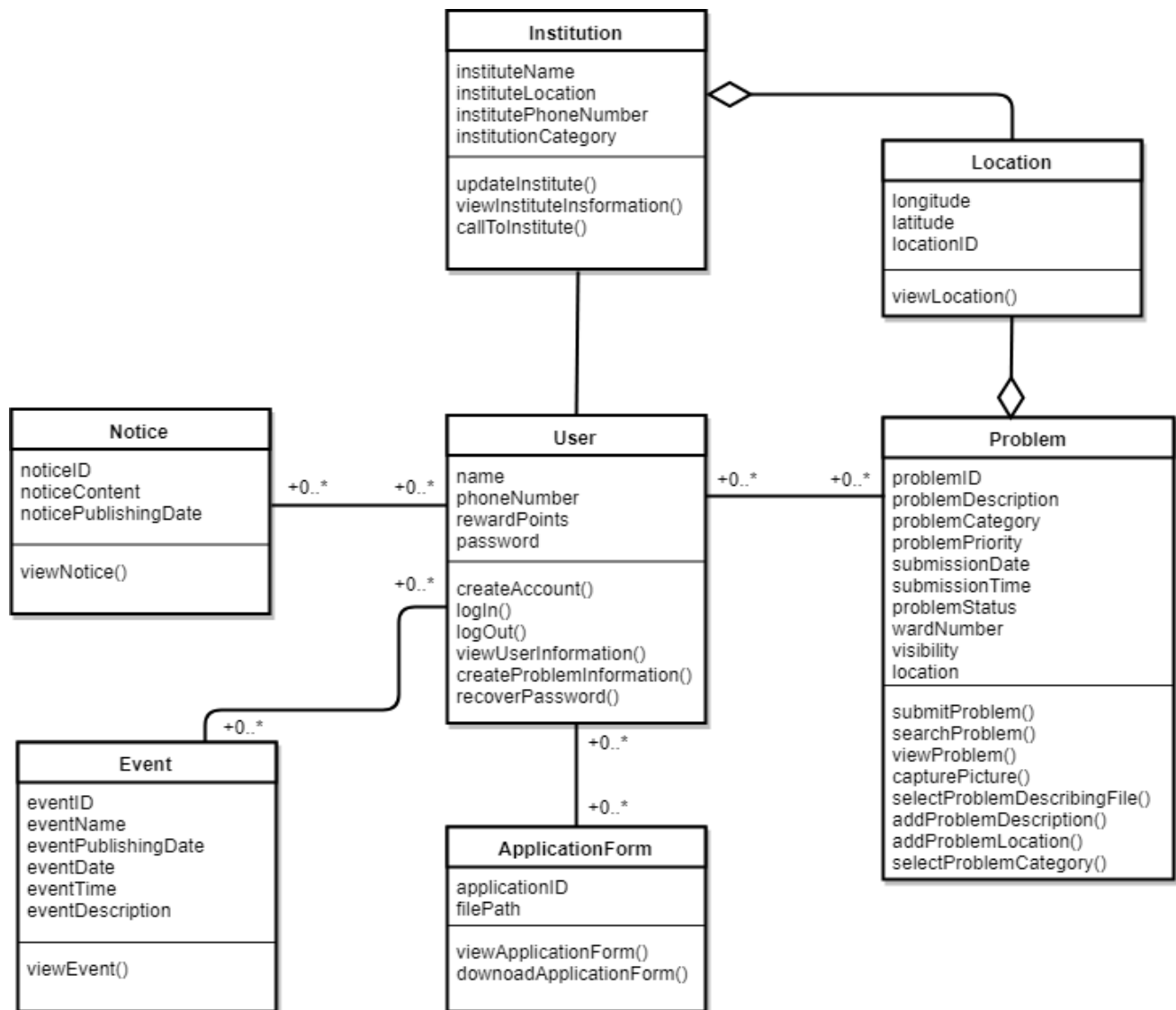


Figure 41: Class diagram of problem submission subsystem of complain box

6.8.1 Class Diagram of Problem Supervision

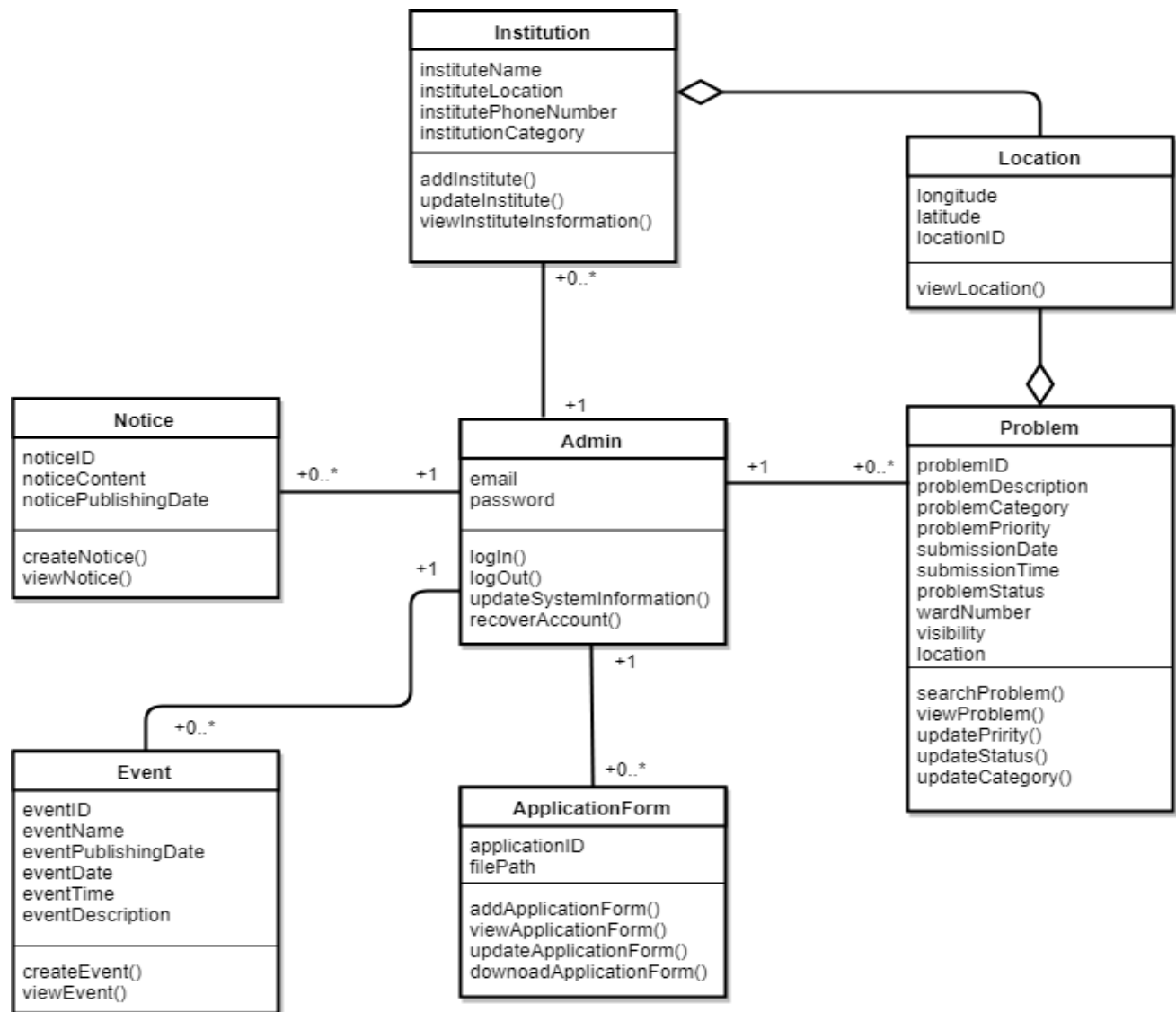


Figure 42: Class diagram of problem supervision subsystem of complain box

Chapter 7: Data Flow Modeling

We intended this chapter to describe data flow modeling for our “Complain Box”.

7.1 Introduction

A data flow diagram is a graphical representation of the flow of data through an information system. We use data flow diagram to diagrammatically represent the flow and exchange of information within our “Complain Box”. As previous chapter, we modeled our data flow diagram based on our two main subsystems.

7.2 Data Flow Diagram

We did try to go initial level to deep level in our system through our data flow diagram.

7.2.1 Level 0 Data Flow Diagram of Complain Box

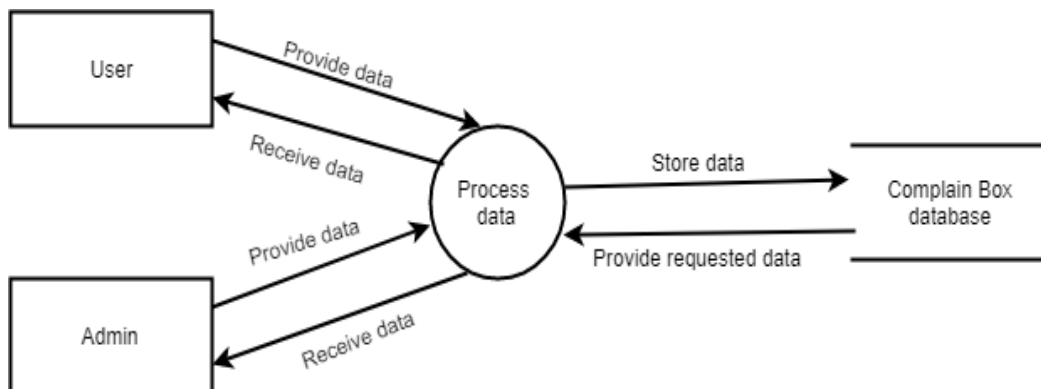


Figure 43: Level 0 data flow diagram of complain box

7.2.2 Level 1 Data Flow Diagram of Complain Box

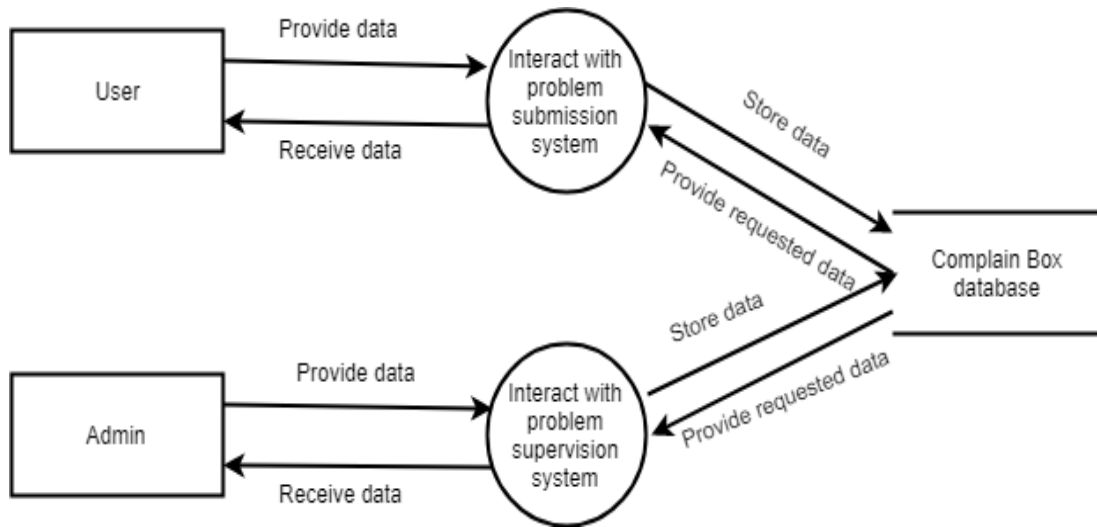


Figure 44: Level 1 data flow diagram of complain box

7.2.2.1 Level 1.1 Data Flow Diagram of Complain Box

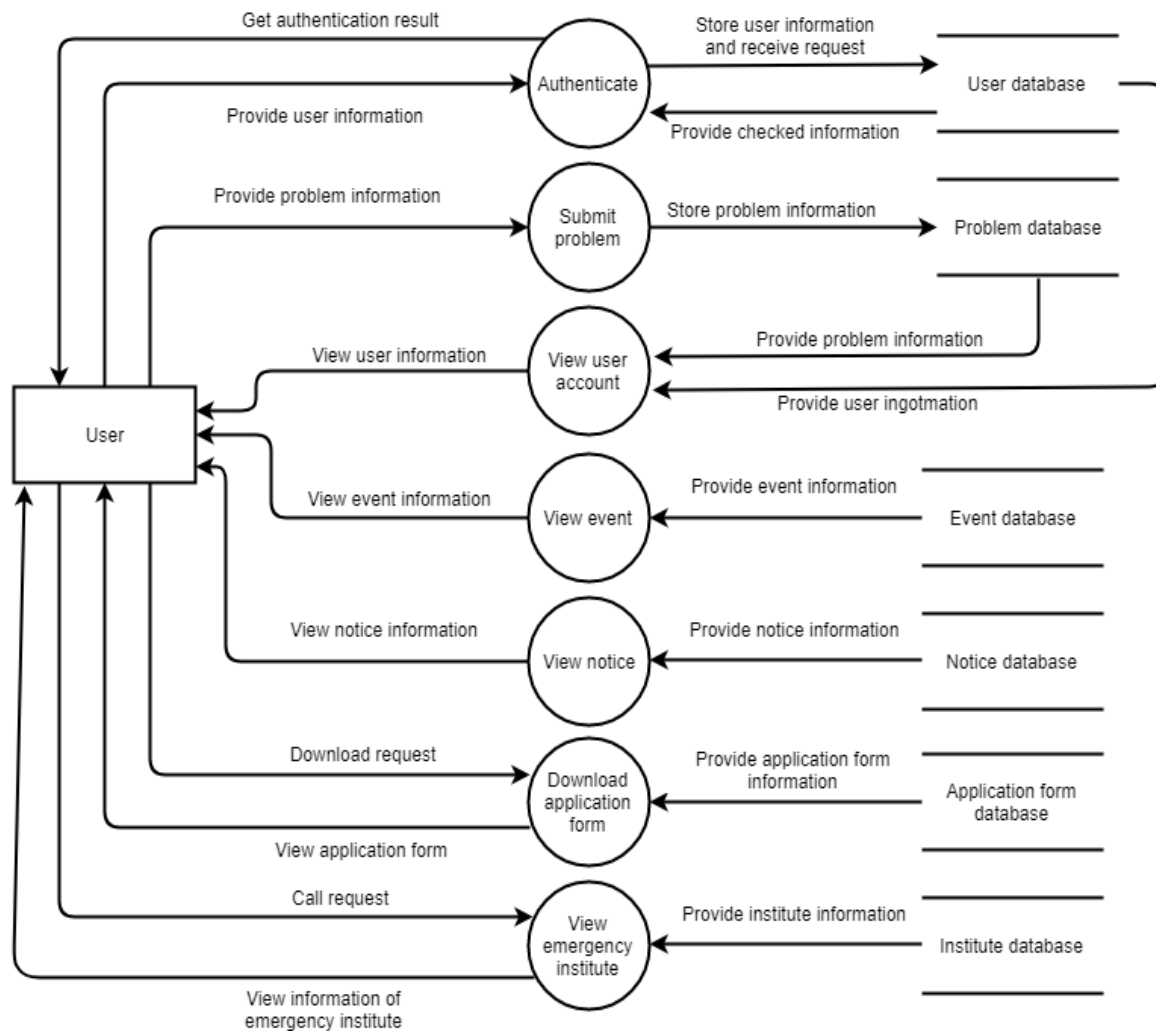


Figure 45: Level 1.1 data flow diagram of complain box

7.2.2.1.1 Level 1.1.1 Data Flow Diagram of Complain Box

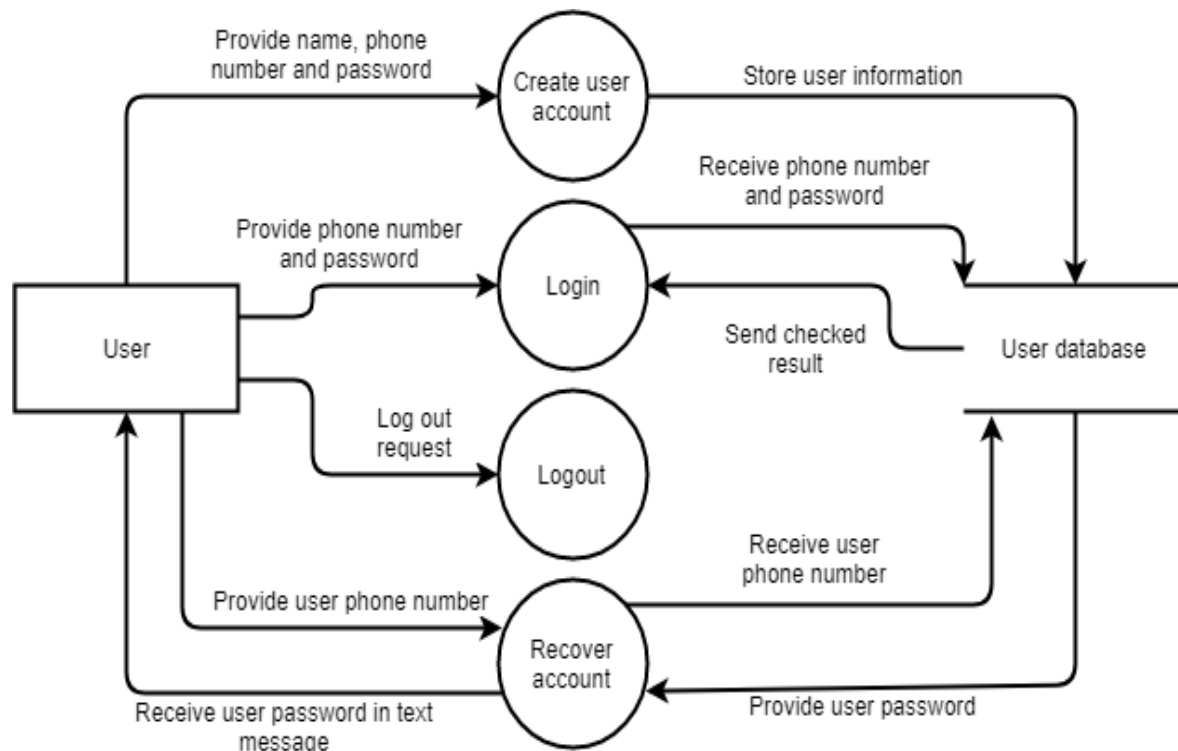


Figure 46: Level 1.1.1 data flow diagram of complain box

7.2.2.1.2 Level 1.1.2 Data Flow Diagram of Complain Box

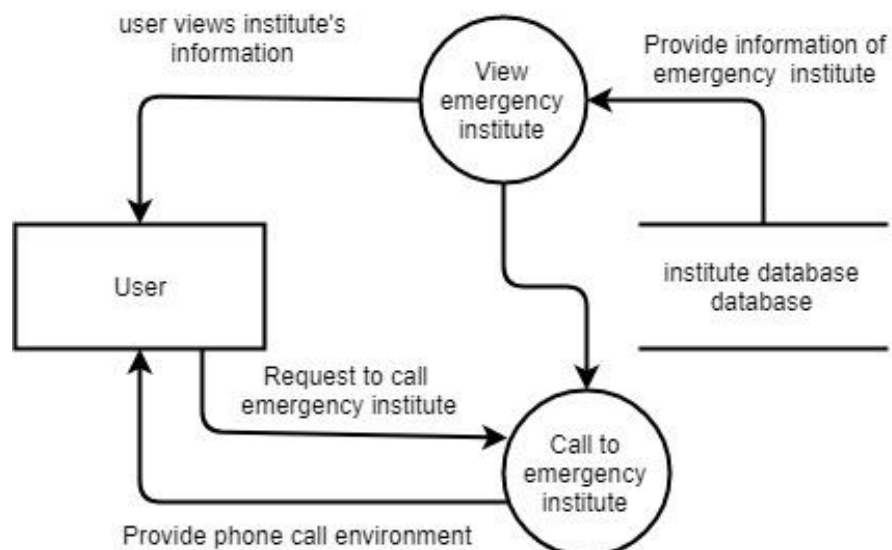


Figure 47: Level 1.1.2 data flow diagram of complain box

7.2.2.1.3 Level 1.1.3 Data Flow Diagram of Complain Box

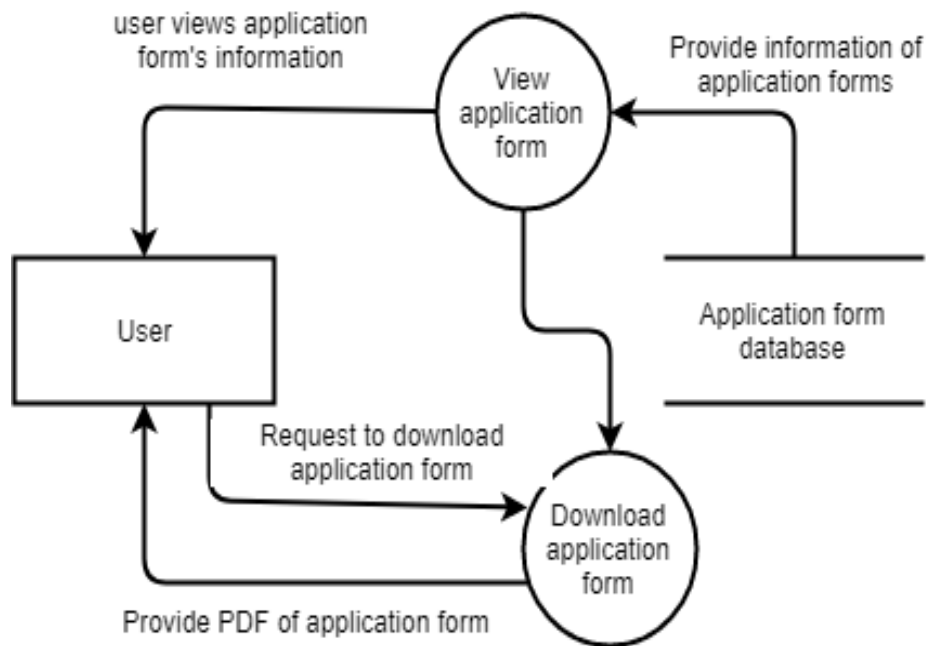


Figure 48: Level 1.1.3 data flow diagram of complain box

7.2.2.2 Level 1.2 Data Flow Diagram of Complain Box

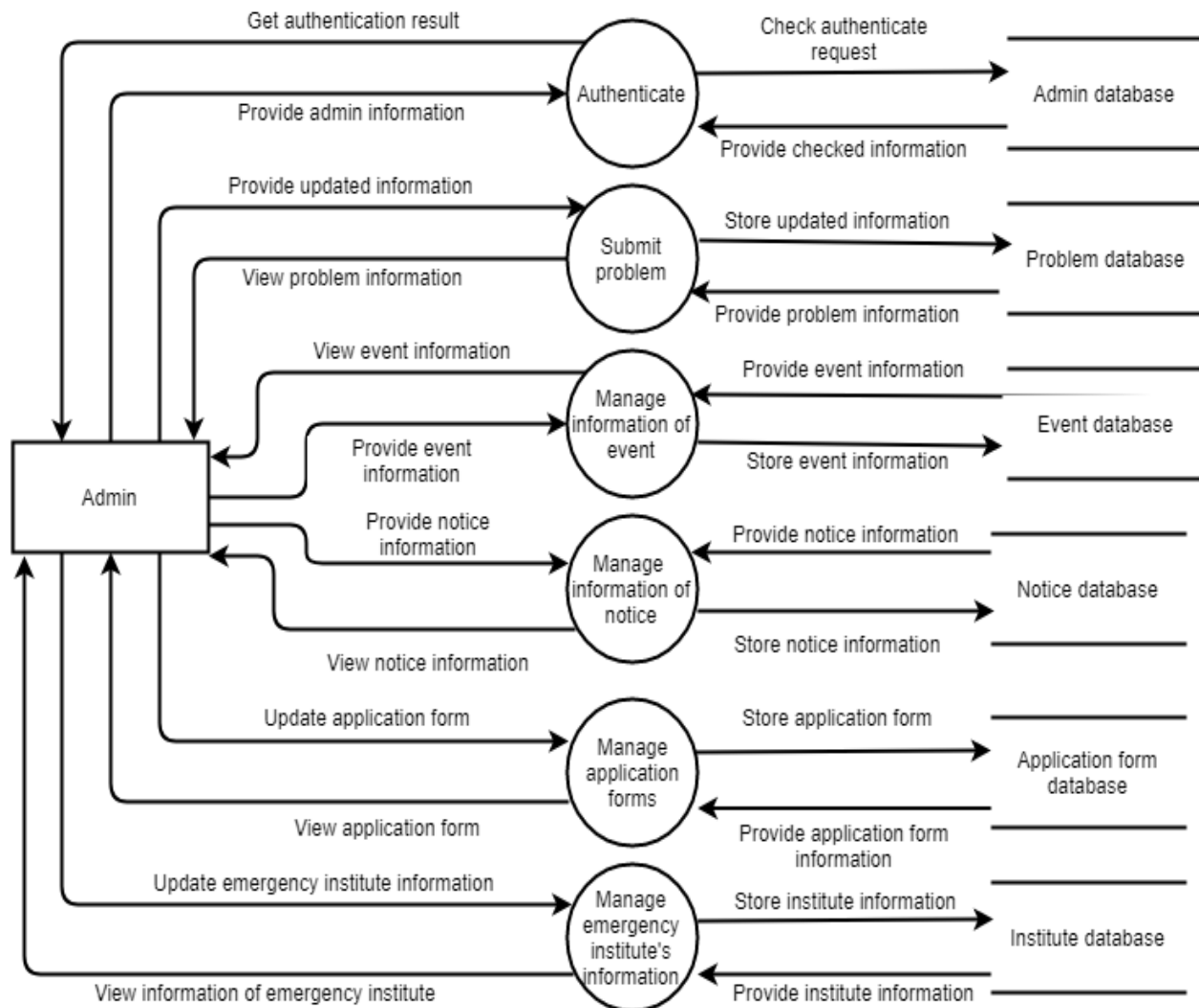


Figure 49: Level 1.2 data flow diagram of complain box

7.2.2.2.1 Level 1.2.1 Data Flow Diagram of Complain Box

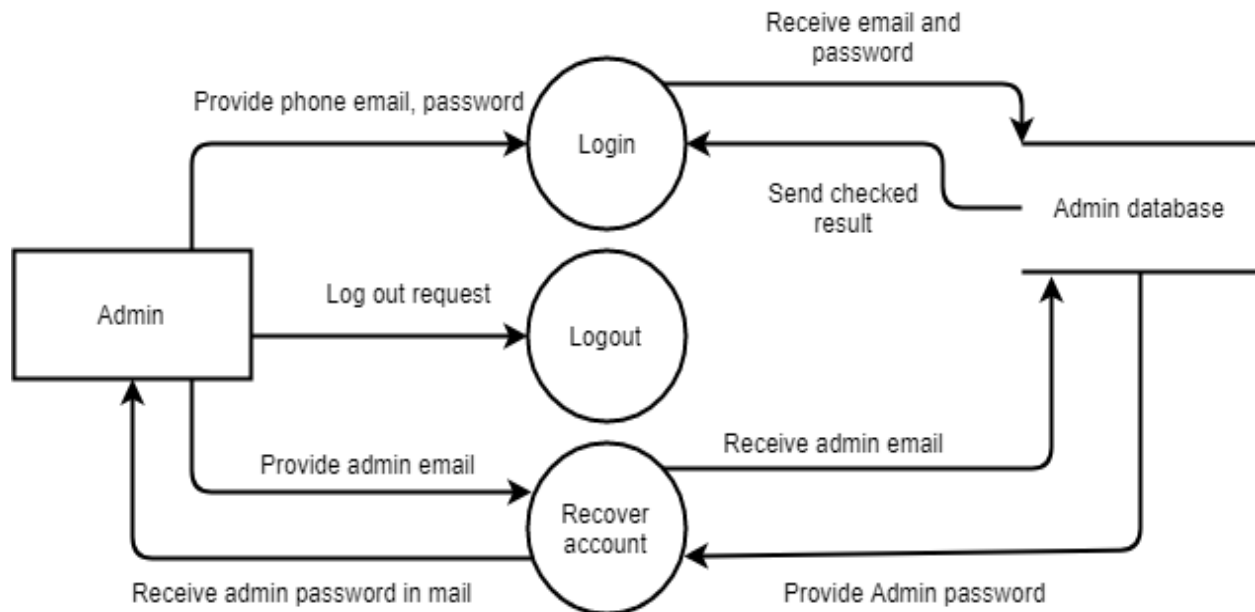


Figure 50: Level 1.2.1 data flow diagram of complain box

7.2.2.2.2 Level 1.2.2 Data Flow Diagram of Complain Box

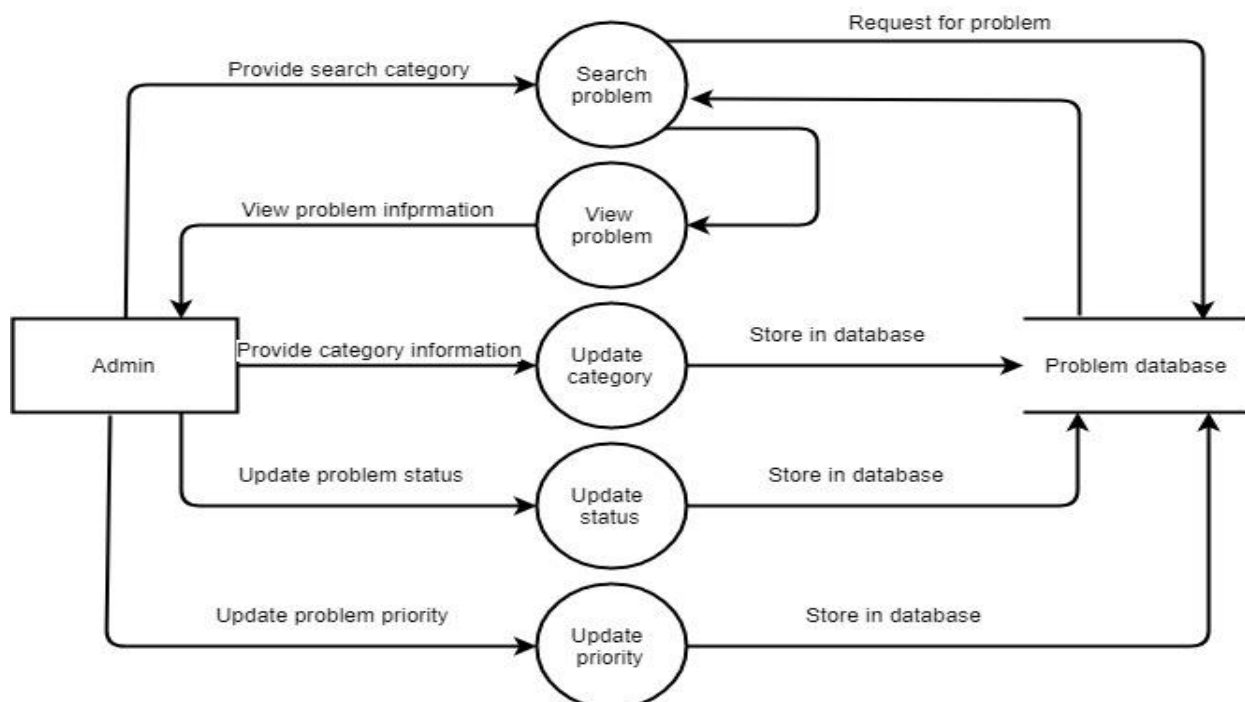


Figure 51: Level 1.2.2 data flow diagram of complain box

7.2.2.2.3 Level 1.2.3 Data Flow Diagram of Complain Box

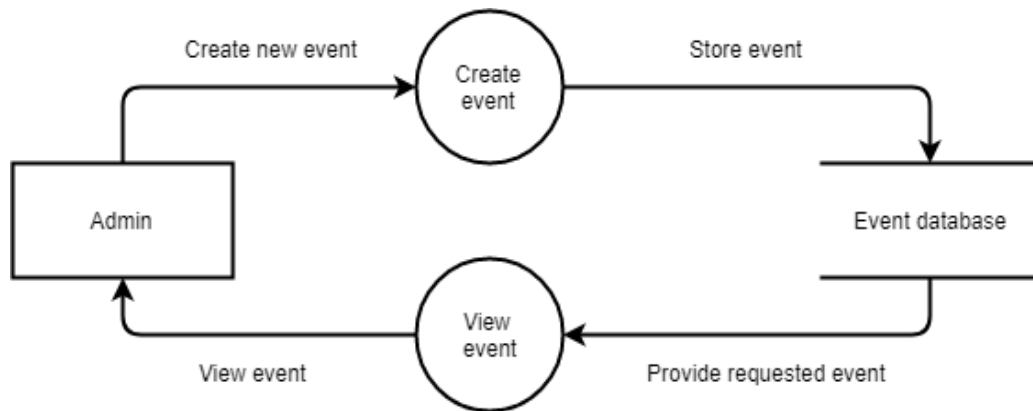


Figure 52: Level 1.2.3 data flow diagram of complain box

7.2.2.2.4 Level 1.2.4 Data Flow Diagram of Complain Box

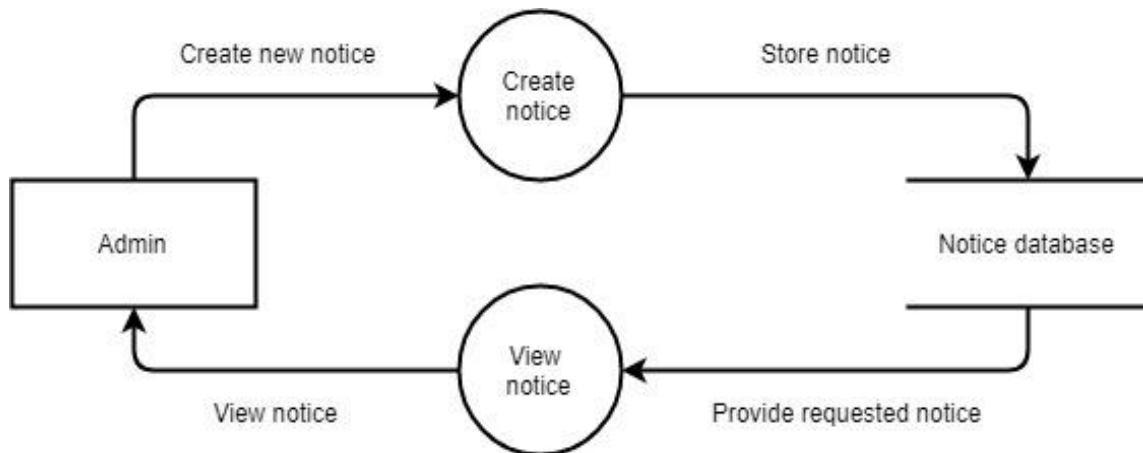


Figure 53: Level 1.2.4 data flow diagram of complain box

7.2.2.2.5 Level 1.2.5 Data Flow Diagram of Complain Box

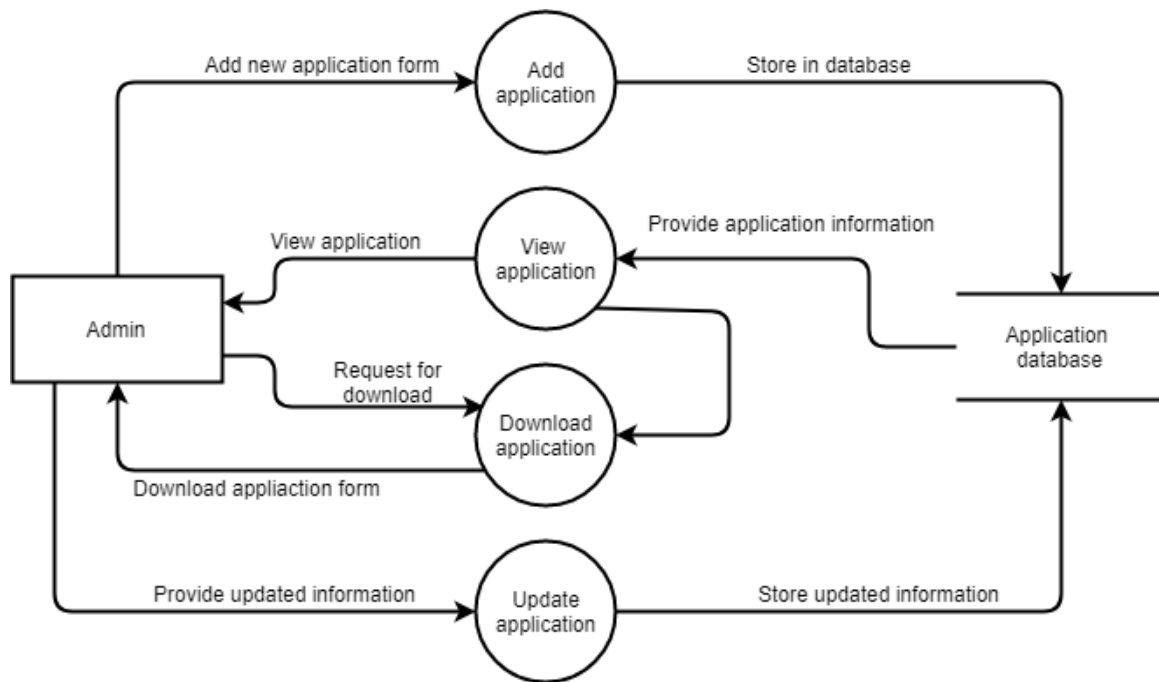


Figure 54: Level 1.2.5 data flow diagram of complain box

7.2.2.2.6 Level 1.2.6 Data Flow Diagram of Complain Box

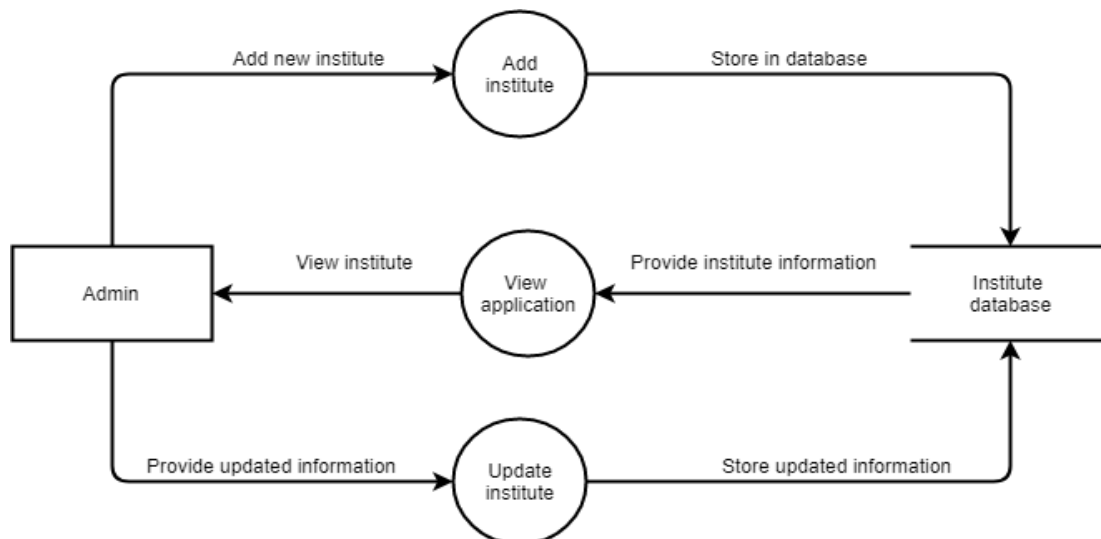


Figure 55: Level 1.2.6 data flow diagram of complain box

Chapter 8: Behavioral Modeling

We intended this chapter to describe behavioral modeling for our “Complain Box”.

8.1 Introduction

The behavioral model indicates how software will respond to external events or stimuli. This chapter throws light on the ways “Complain Box” interacts.

8.2 State Transition

In the context of behavioral modelling to different characterization of states must be considered and these are:

- The state of each class as the system performs its functions
- The state of the system observed from the outside as the system performs its functions

8.2.1 Event Identification

State diagram represents active states for each class the events (triggers). For this we identified all events, their initiators and collaborators.

Table 43: Event identification

Event	Primary Object	Collaborator	Methods
Create an account	User		createAccount()
Store user information	User		createAccount()
Login	User		login()
Recover password	User		recoverPassword()
Create problem Information	User	Problem	createProblemInformation()
Select problem category	Problem		selectProblemCategory()
Capture a picture	Problem		capturePicture()
Submit problem	Problem		submitProblem()
Select problem describing file	Problem		selectProblemDescribingFile()
Add description	Problem		addProblemDescription()
Set problem location	Problem	Location	addProblemLocation()
View user account	User	Problem	viewUserInformation()
View emergency support information	Institution	Location	viewInstituteInformation()
Communicate to institute	Institute		callToInstitute()
View notice	Notice		viewNotice()

View event	Event		viewEvent()
Download application form	ApplicationForm		downloadApplicationForm()
Add application form	ApplicationForm		addApplicationForm()
View application form's information	ApplicationForm		viewApplicationForm()
Update Application form	ApplicationForm		updateApplicationForm()
Log out	User		logout()
View submitted problem	Problem		viewProblem()
Log in to system	Admin		login()
Recover Admin account	Admin		recoverAccount()
Update system information	Admin	Problem	updateSystemInformation()
Log out	Admin		logout()
Edit problem category	Problem		updateCategory()
Update priority	Problem		updatePriority()
Search problem	Problem	Location	searchProblem()
Update status	Problem		updateStatus()
Create notice	Notice		createNotice()
Create event	Event		createEvent()
Add emergency support information	Institution	Location	addInstitute()
Update emergency institute information	Institute	Location	updateInstitute()
View location	Location		viewLocation()

8.2.2 State Transition Diagram of Problem Submission Subsystem

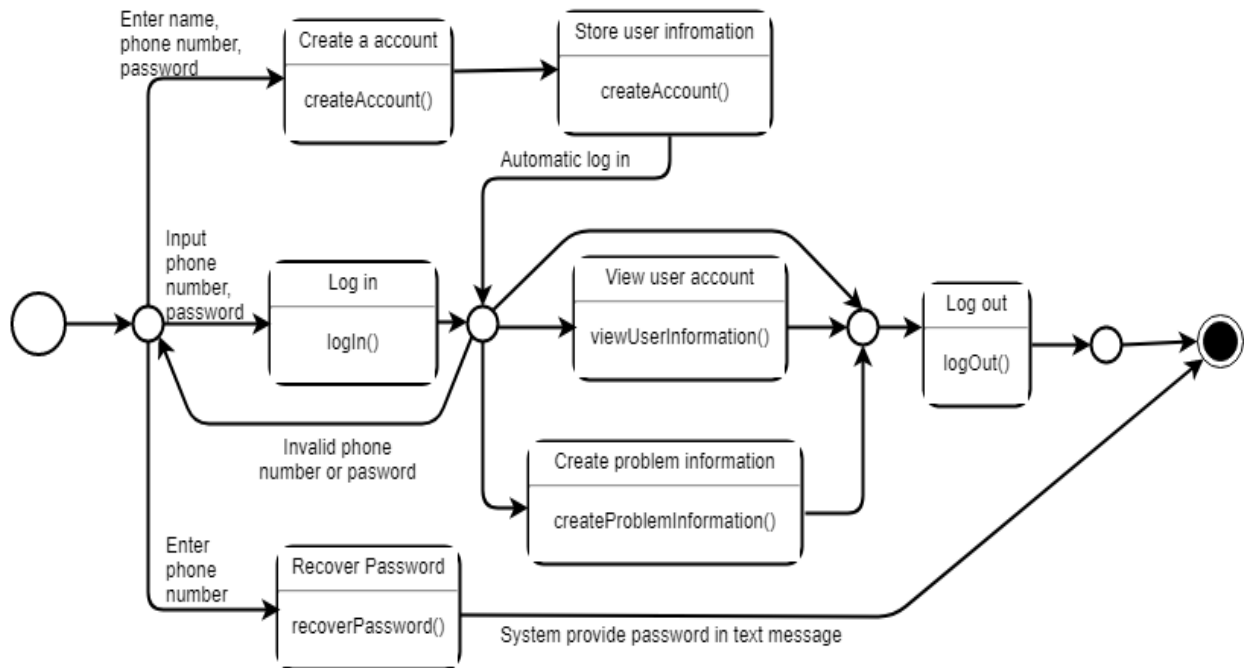


Figure 56: State transition diagram of User

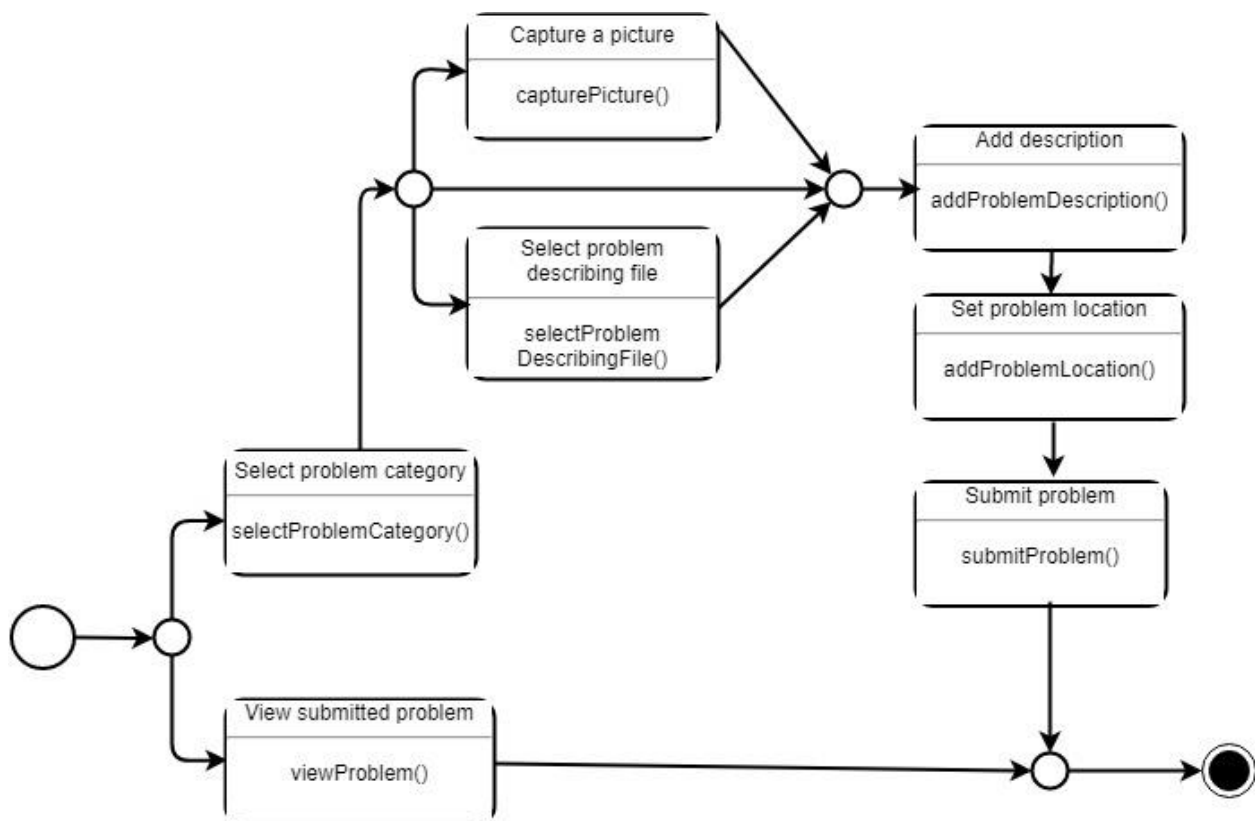


Figure 57: State transition diagram of Problem

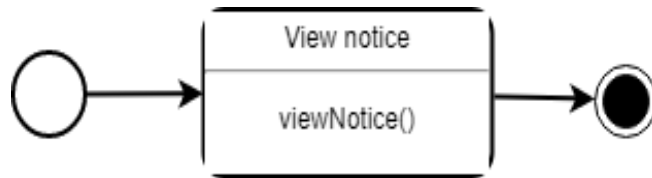


Figure 58: State transition diagram of Notice

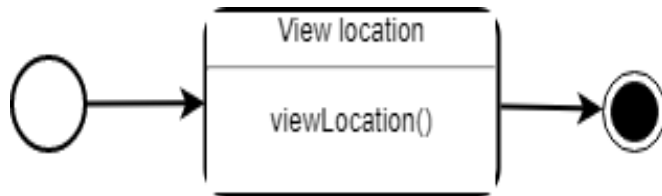


Figure 59: State transition diagram of Location

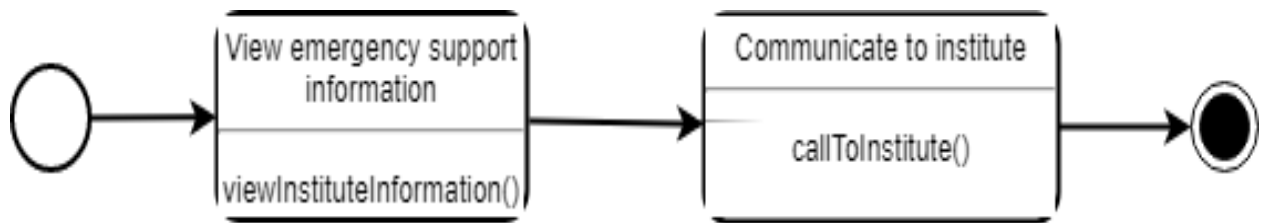


Figure 60: State transition diagram of Institution

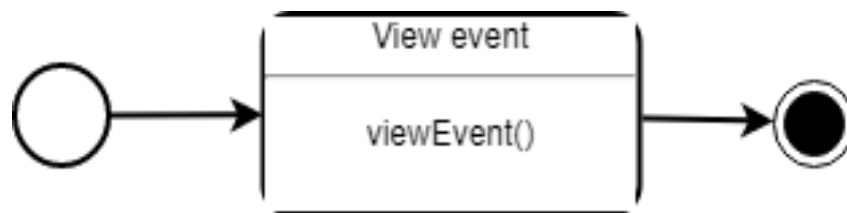


Figure 61: State transition diagram of Event

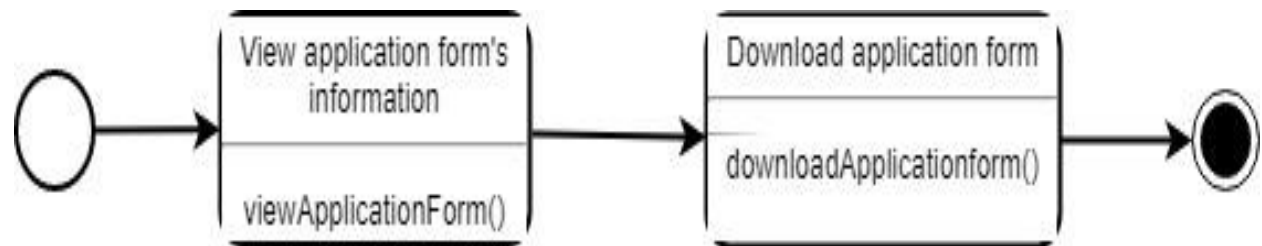


Figure 62: State transition diagram of ApplicationForm

8.2.3 State Transition Diagram of Problem Submission Subsystem

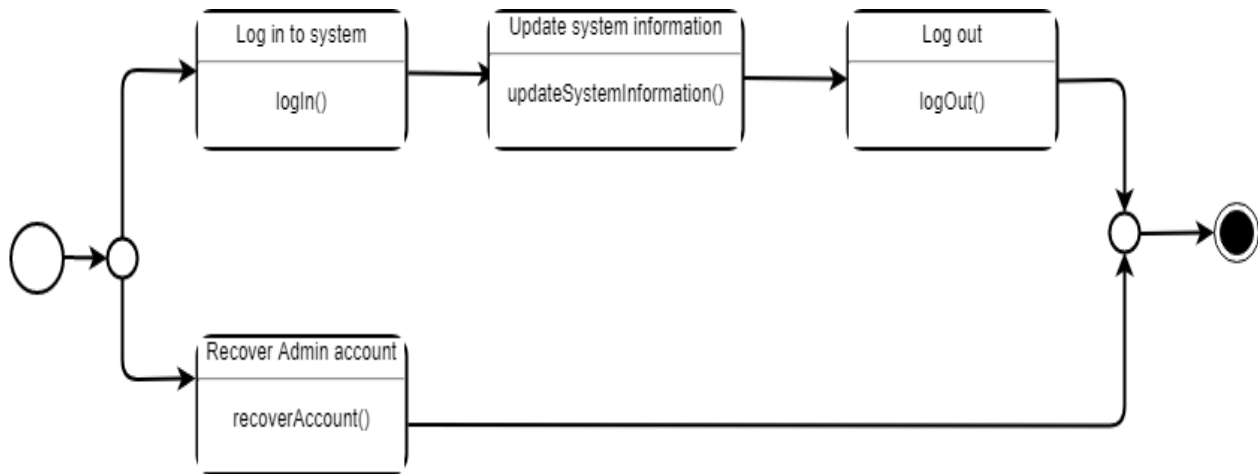


Figure 63: State transition diagram of Admin

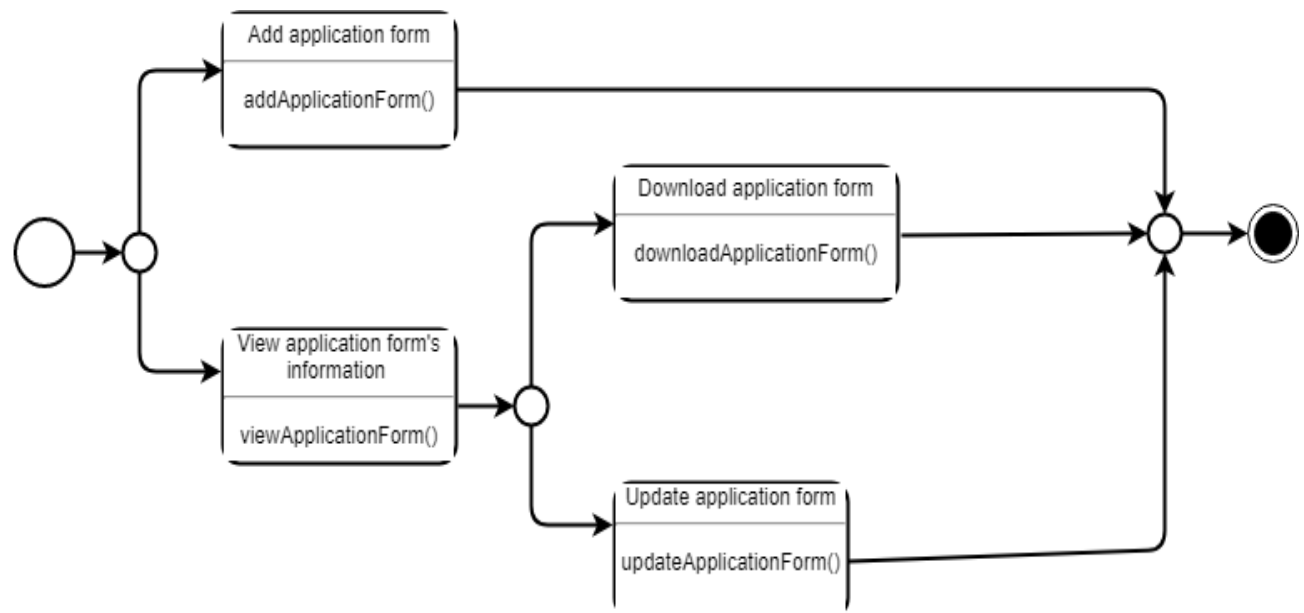


Figure 64: State transition diagram of ApplicationForm

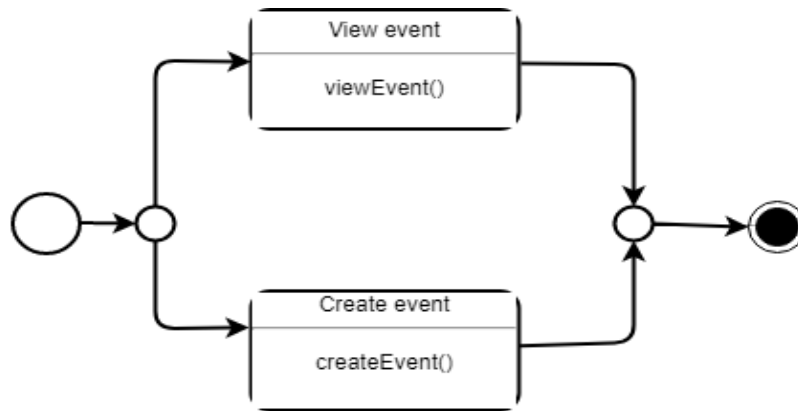


Figure 65: State transition diagram of Event

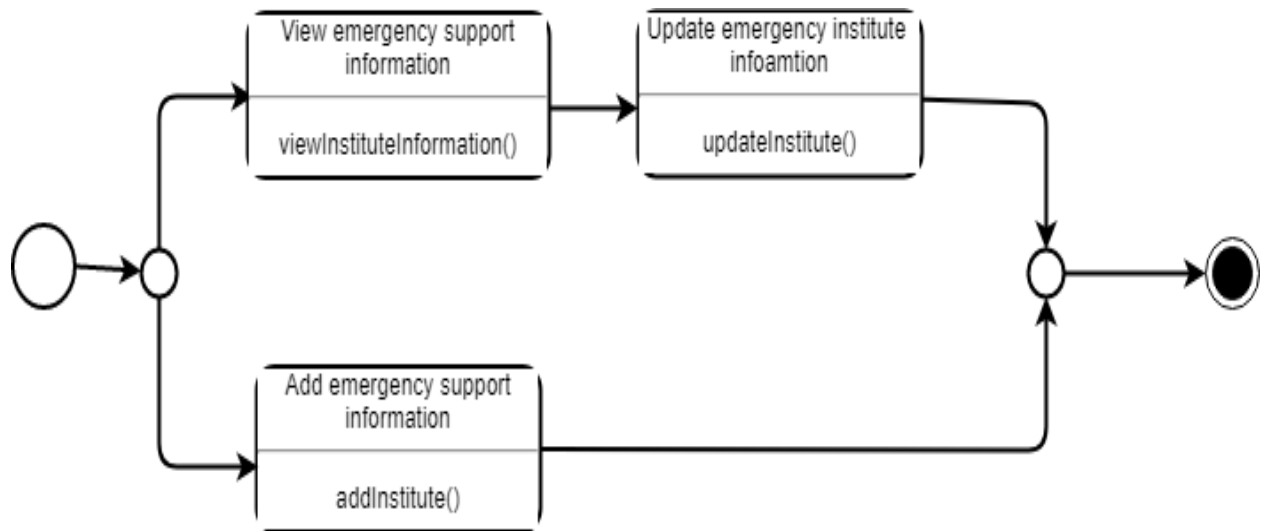


Figure 66: State transition diagram of Institution

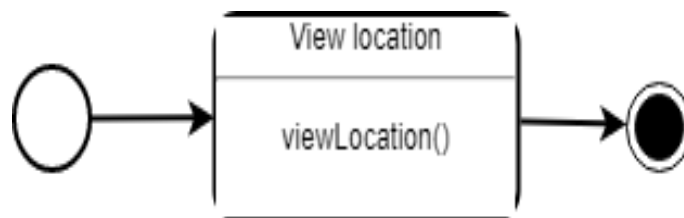


Figure 67: State transition diagram of Location

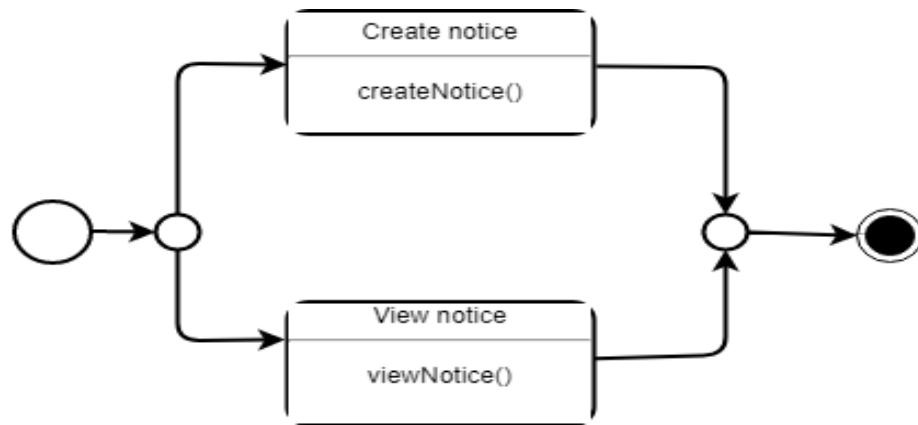


Figure 68: State transition diagram of Notice

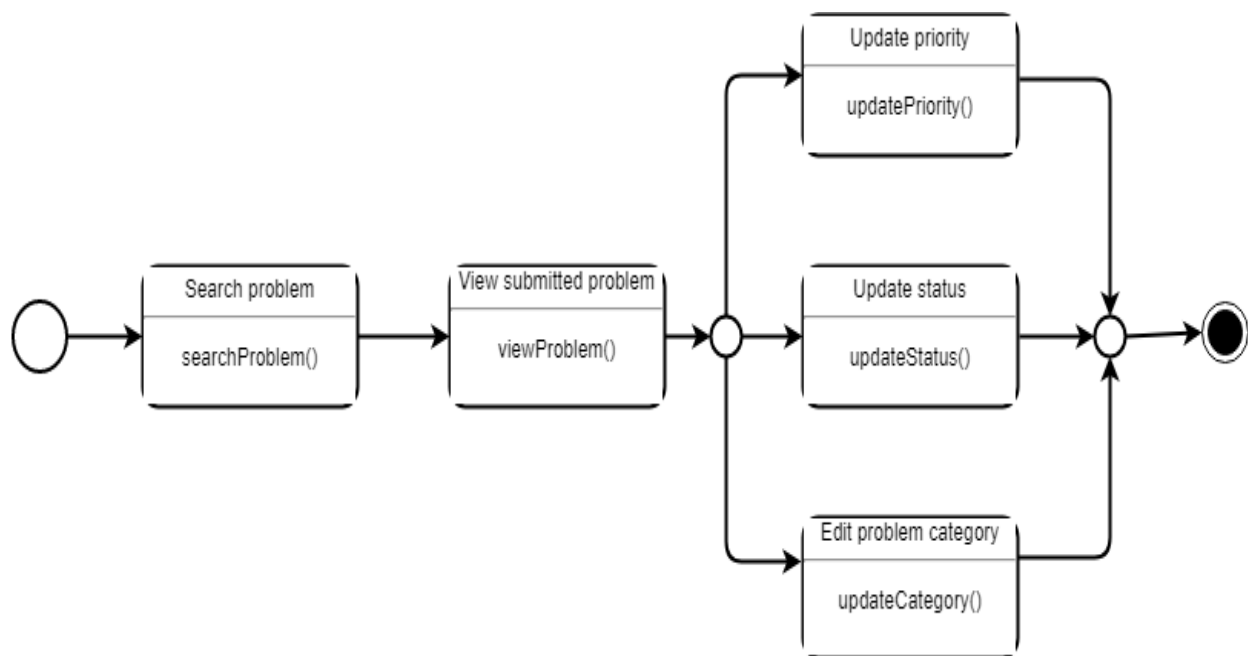


Figure 69: State transition diagram of Problem

8.3 Sequence Diagram

We designed two different sequence diagrams for our two subsystems.

8.3.1 Sequence Diagram of Problem Submission Subsystem

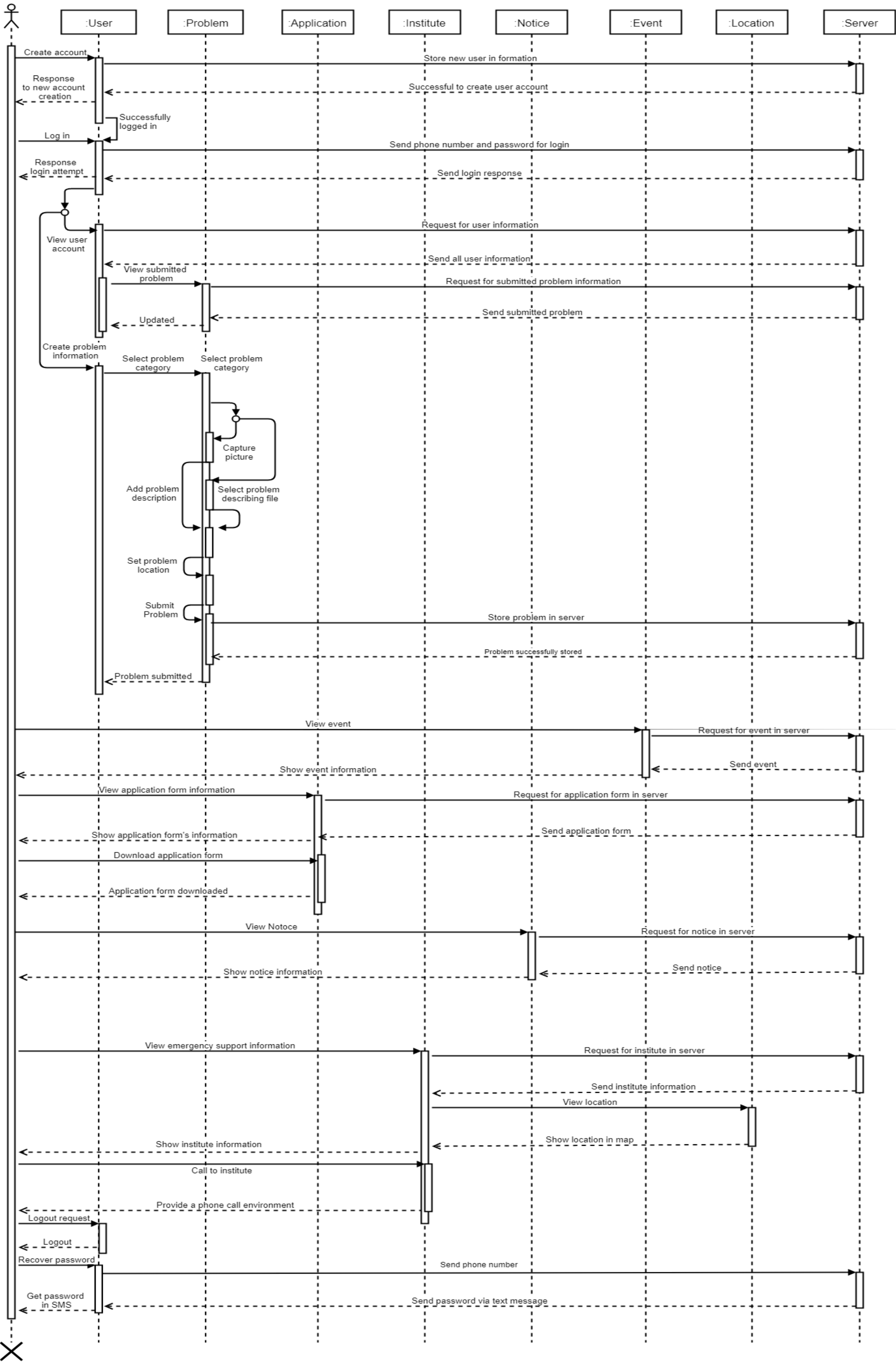


Figure 70: Sequence diagram of problem submission subsystem

8.3.2 Sequence Diagram of Problem Supervision Subsystem

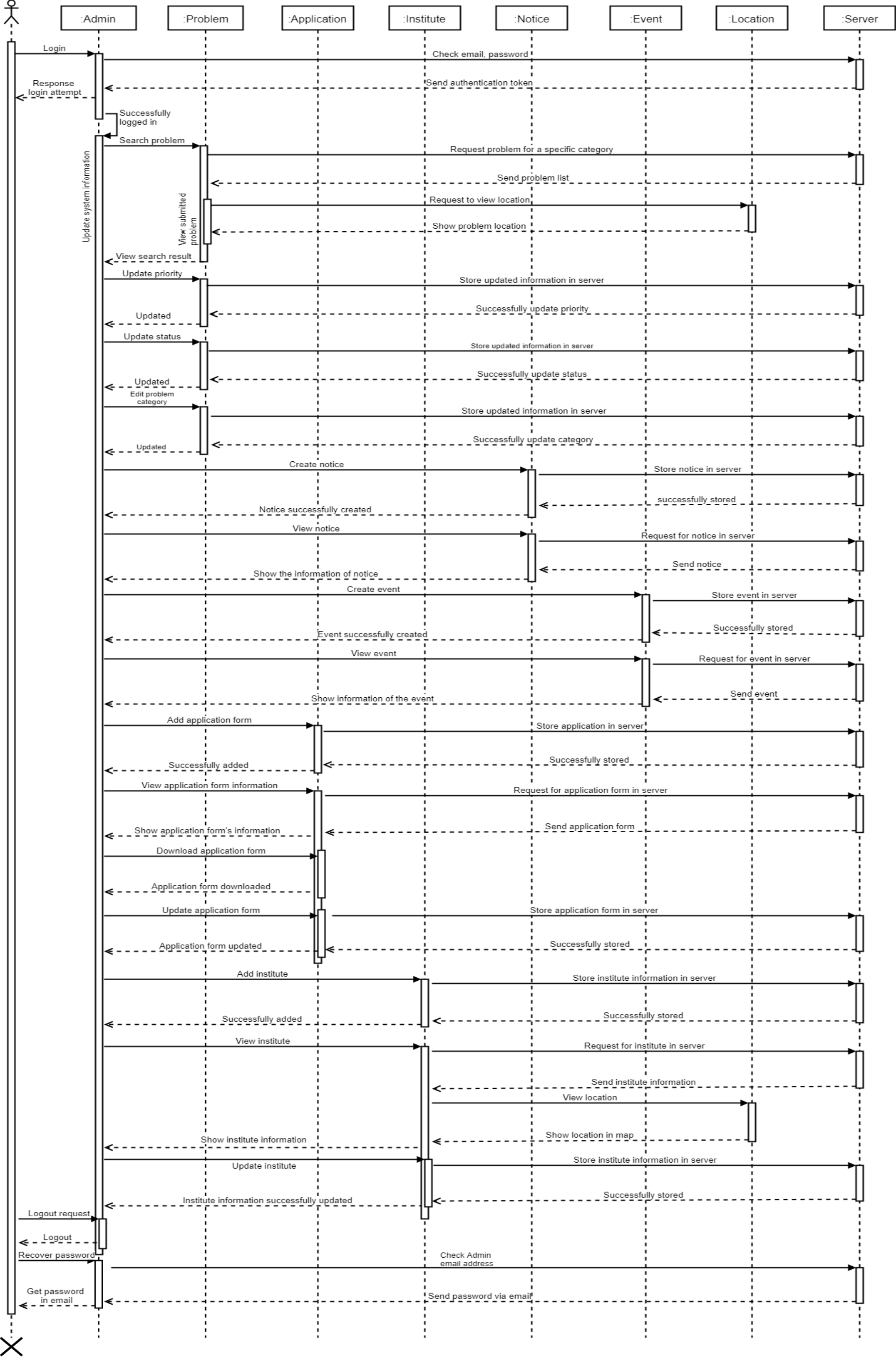


Figure 71: Sequence diagram of problem supervision subsystem