



Safety Pin (SRS)

A service based android application to ensure public safety

Supervised by

Dr. Sumon Ahmed
Associate Professor

Institute Of Information Technology (IIT),
University of Dhaka

Submitted by

Md Mehedi Hasan
BSSE 1037

Fahim Ahmed
BSSE 1039

Institute Of Information Technology (IIT),
University of Dhaka

Software Requirement Specification Document

This chapter is a part of Software Project Lab-2(SPL-2), intended to specify the purpose of this document and the intended audience of it.

1.1 Purpose

This document is the Software Requirements Specification (SRS) for Safety-Pin which is the name of our project for SPL-2. It contains detailed functional, non-functional, and support requirements and establishes a requirements baseline for development of the android app. The requirements contained in the SRS are independent, uniquely numbered, and organized by topic. The SRS serves as the official means of communicating user requirements to the developer and provides a common reference point for both the developer team and stakeholder community.

1.1.2 Intended Audience

This SRS is intended for several audiences, including the customers, as well as the project managers, designers, developers, and testers.

- The customer will use this SRS to verify that the developer team has created a product that is acceptable and useful to the customer.
- The project managers of the developer team will use this SRS to plan milestones and a delivery date, and ensure that the developing team is on track during development of the system.
- The designers will use this SRS as a basis for creating the system's design. The designers will continually refer back to this SRS to ensure that the system they are designing will fulfill the customer's needs.
- The developers will use this SRS as a basis for developing the system's functionality. The developers will link the requirements defined in this SRS to the

software they create to ensure that they have created software that will fulfill all of the customer's documented requirements.

- The testers will use this SRS to derive test plans and test cases for each documented requirement. When portions of the software are complete, the testers will run their tests on that software to ensure that the software fulfills the requirements documented in this SRS. The testers will again run their tests on the entire system when it is complete and ensure that all requirements documented in this SRS have been fulfilled.

1.1.3 Summary

This analysis of the audience helped us to focus on the users who will be using our analysis. This overall document will help each and every person related to this project to have a better idea about the project.

1.2 Inception

In this chapter, the Inception part of the SRS will be discussed briefly

1.2.1 Introduction

Inception is the beginning phase of requirements engineering. It defines how a software project gets started and what is the scope and nature of the problem to be solved. The goal of the inception phase is to identify concurrence needs and conflict requirements among the stakeholders of a software project. To establish the groundwork we have worked with the following factors related to the inception phases:

- Identifying Stakeholders
- Recognizing multiple viewpoints
- Working towards collaboration
- Asking the First Questions

1.2.1.1 Identifying Stakeholders

Stakeholder refers to any person or group who will be affected by the system directly or indirectly. Stakeholders include end-users who interact with the system and everyone else in an organization that may be affected by its installation. At inception, a list of people who will

contribute input as requirements are elicited. The initial list will grow as stakeholders are contacted because every stakeholder will be asked: “whom else do you think I should talk to?”

The following stakeholders were identified for the Safety Pin:.

➤**Customer:** A customer is an individual or business that purchases or consumes the goods or services produced by a business. Attracting customers is the primary goal of most businesses as it is the customer who creates demand for goods or services. In our case the customer is the end user of the mobile application. The demands of the end user will be of top most priority for developing the mobile application.

➤**Software Developer:** A software developer is concerned with facets of the software development process, including the research, design, programming, maintenance and testing of computer software. She will be responsible for the outcomes of the software.

1.2.1.2 Recognizing Multiple Viewpoints

Different stakeholders demand different features from the software. To satisfy the stakeholders, most of these features should be included in the software

Customers' viewpoint

- Easy to use and simple user interface.
- Allow any user to search for various places on the map
- Allow user to find know his/her own current location
- Allow user to share his/her current location
- Allow users to give review on a location about an incident
- Provide users with some emergency help features.

Developer's viewpoint

- Easy to build
- Error free effective software
- No conflicting requirement
- Getting a maximum experience from development

1.3 Elicitation

This chapter specifies the Elicitation phase.

1.3.1 Introduction

Requirements Elicitation is a part of requirements engineering that is the practice of gathering requirements from the users, customers and other stakeholders. Many difficulties were faced, like understanding the problems, making questions for the stakeholders, limited communication with the stakeholders due to a short amount of time and volatility. Though it is not easy to gather requirements within a very short time, these problems have been surpassed in an organized and systematic manner.

1.3.2 Eliciting Requirements

The main task of this phase is to combine the elements of problem solving, elaboration, negotiation and specification. The collaborative working approach of the stakeholders is required to elicit the requirements. The following tasks were done for eliciting requirements:

1. Collaborative Requirements Gathering
2. Quality Function Deployment
3. Usage Scenarios
4. Elicitation work products

1.3.3 Collaborative Requirements Gathering

Many different approaches to collaborative requirements gathering have been proposed. Each makes use of a slightly different scenario. We completed the following steps to do it.

- The meetings were conducted with a few customers, questioned about their requirements and expectations from Safety Pin
- The customers were asked which requirements should be added to the application
- At last we selected our final requirements from the discussions

1.3.4 Quality Function Deployment

Quality Function Deployment (QFD) is a technique that translates the needs of the customer into technical requirements for software. It concentrates on maximizing customer satisfaction from the Software engineering process. With respect to our project the following requirements are identified by a QFD.

1.3.4.1 Normal Requirements

The normal requirements are generally the objectives and goals that are stated for a product or system during meetings with the customer. The presence of these requirements fulfills customers' satisfaction. These are the normal requirements for the project.

1. Android application
2. Accessible via the Internet.
3. Allow any user to search for places.
4. Allow Administrators to check user information(not personal)
5. Allow valid users to login and logout
6. Restrict access to functionality of the system based upon user roles
7. Getting information about preferred locations and search strings
8. User manual and guidelines
9. Storing information regarding danger zones
10. Identifying danger zones
11. Contacting users in times of danger

1.3.4.2 Expected Requirements

These requirements are intrinsic to the product or system and may be so elementary that the customer does not explicitly state them. Their absence will be a cause for significant dissatisfaction. Below the expected requirements for our project are briefly described.

1. Error free software
2. Strong authentication system
3. User friendly
4. Effective system
5. No ambiguous feature
6. Data backup
7. The system shall automatically check the location availability
8. The system shall allow the user to login based upon an assigned phone number
9. Search locations based on search text
10. Share navigation information to other users
11. Alert user before entering marked area

1.3.4.3 Exciting requirements

These requirements are for features that go beyond the customer's expectations and prove to be very satisfying when present

1. Categorized zones based on different types of occurrences.
2. Different color identifiers based on the number of occurrences.
3. Automatic alarm before approaching marked zones.
4. Single tap emergency call.

1.4 Scenario Based Modeling

1.4.1 Usage scenario

Sign Up:

In the android application, there is an authentication part, where it allows the user to access the system. Users of this system are the mass people. To access the system, a user requires an account and for that he/she must fill up a form. Users have to give their Username, password, age, gender and phone number. No user name can contain any number, punctuation mark or any special character and the length of the name should be between 2 characters and 30 characters. There will be duplicity and validity (syntax) checking for email and mobile number. If all the information is correct, the user will be sent a verification code to his/her phone number. The process will require the phone number at first. Later a verification code and the user has to insert the code in the appropriate field within 60 seconds. By inserting the correct verification code, the user will be able to set a password which must contain at least 6 characters.

Sign in:

If any user has an account, he/she can sign in to the system. To sign in, a user has to give his/her mobile number and password. The mobile number and password will be verified. If the verification is successful, the user can sign in to the system successfully. If the password/number is wrong, there is a retry option.

Sign Out:

Users have an option to log out from the system. The phone number associated with the user will be displayed on the login application interface.

Manage Account:

Any user can change their information. To change information, he/she has to sign in then change information. He/she has to confirm the changes and the changes will be confirmed.

1.4.2 User Activities

(a) Sharing Location :

An authorized user can share the locations with some reviews/comments where he/she has faced any crime or harassment. The user can add his/her gender, date & time of the crime and category of crime.

(b) View colorized area zones:

Any user can view the marked area zones but cannot modify others' reviews.

(c) Get Alert :

An user can get warning alarms before approaching those marked-area zones. The notification will be sent around a certain circumference of those zones.

(d) View police station numbers:

Users can view all local police station numbers pre-given in the app. There will be an option for calling on those numbers. They will have a quick dial feature.

(e) Call 999:

A direct call button to call the 999 will be added.

The database will store the user information. Finally, out of the data, the system will inform the citizens about the criminal activities around them.

1.4.3 Use case Diagrams

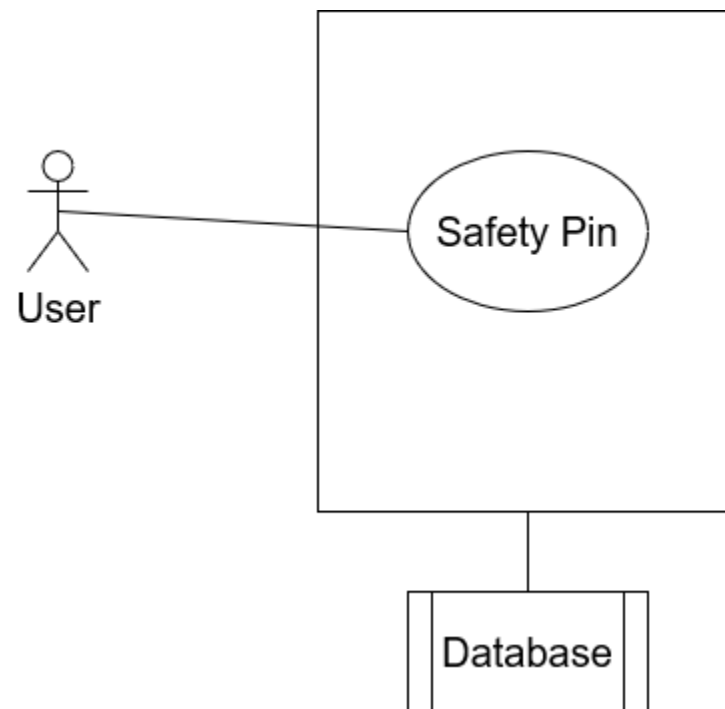


Figure 1.1 Use Case level 0

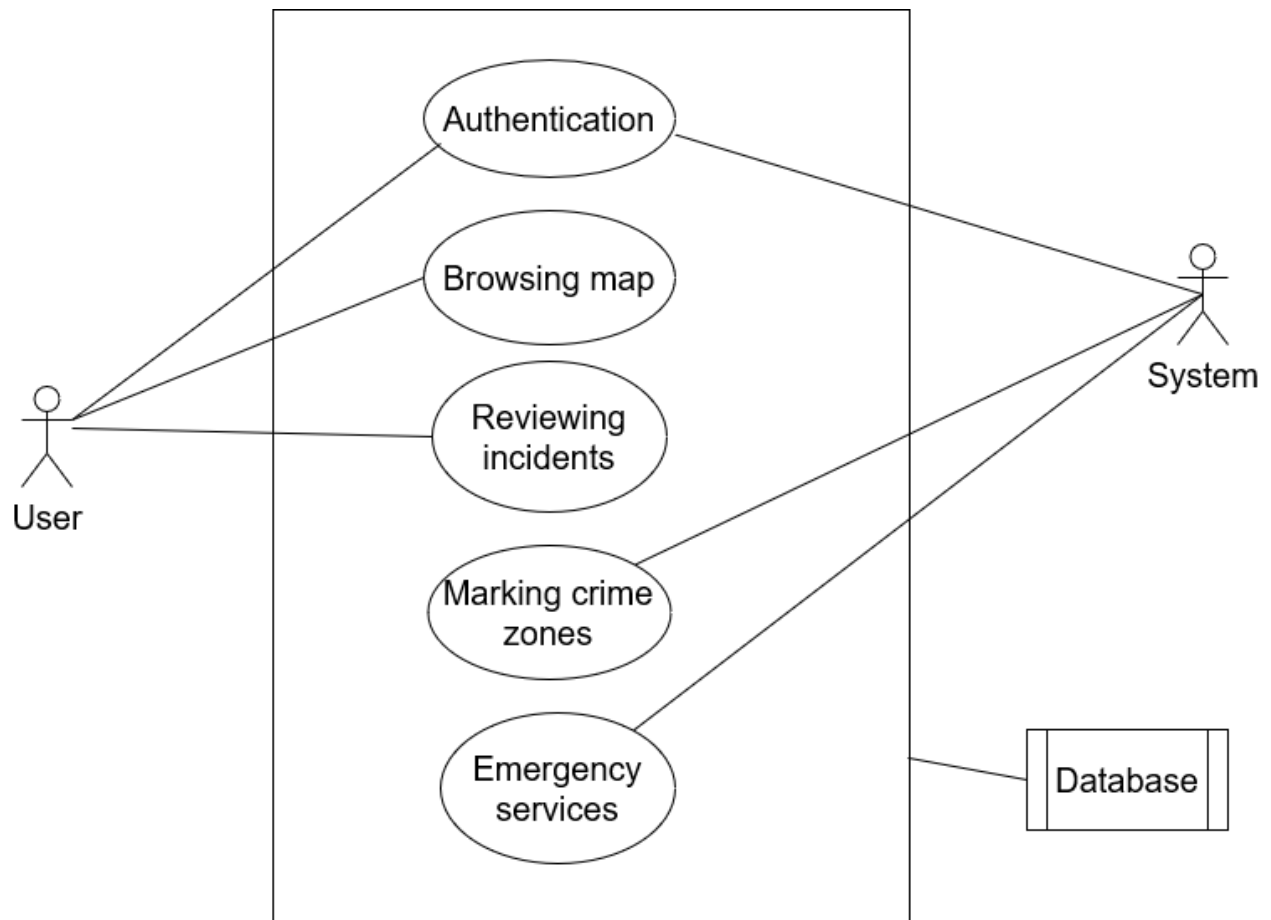


Figure 1.2 Use Case level 1

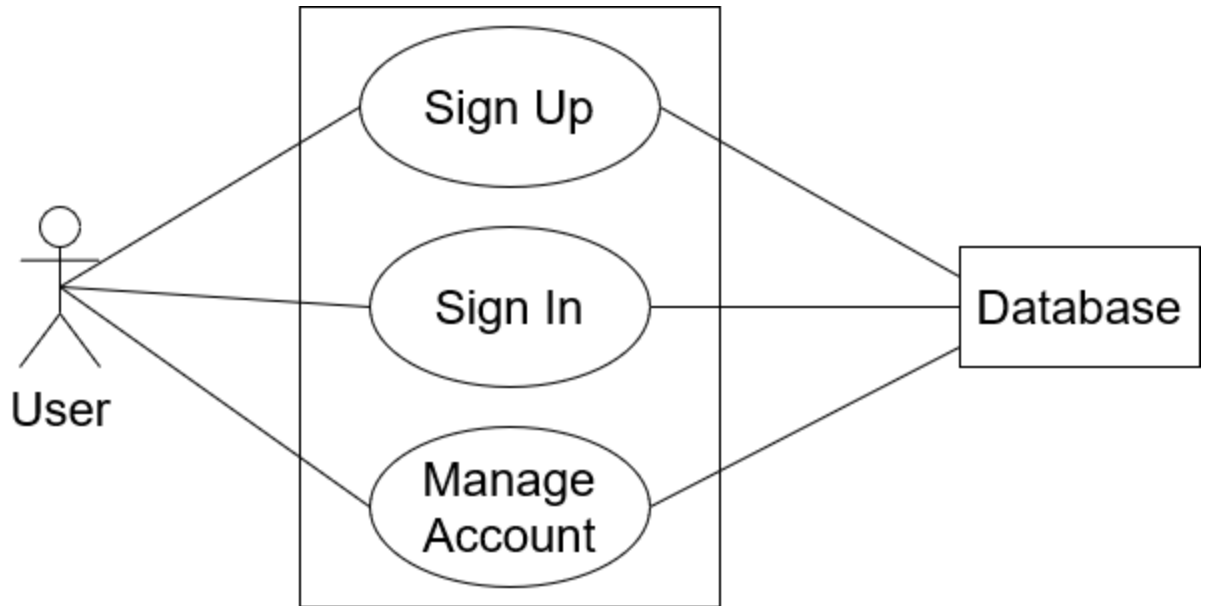


Figure 1.3 Use Case level 1.1

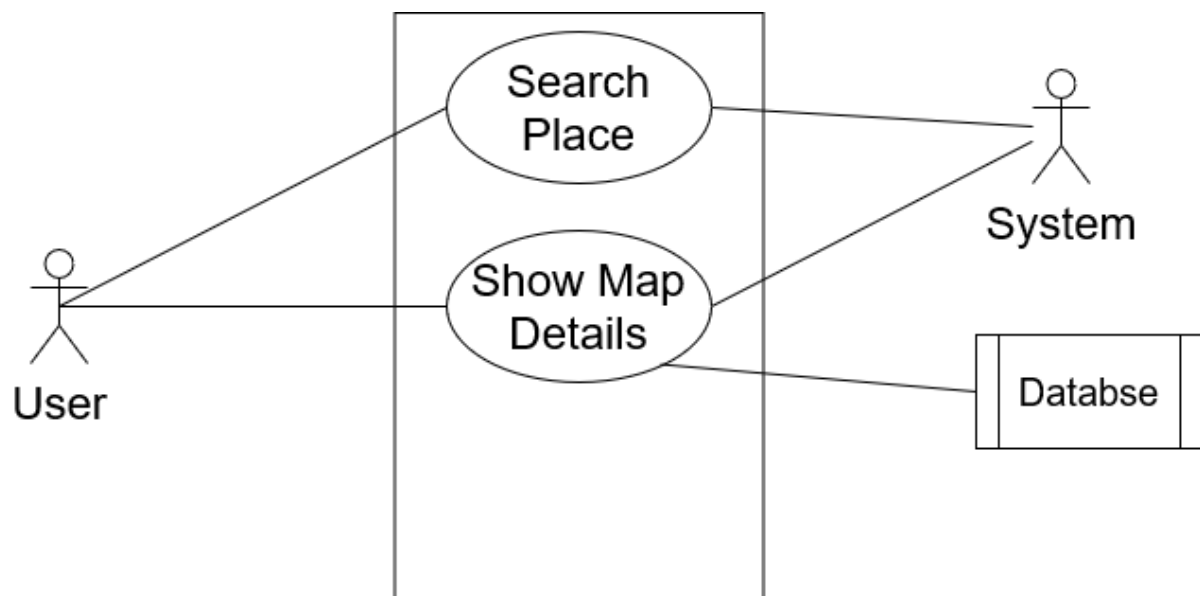


Figure 1.4 Use Case level 1.2

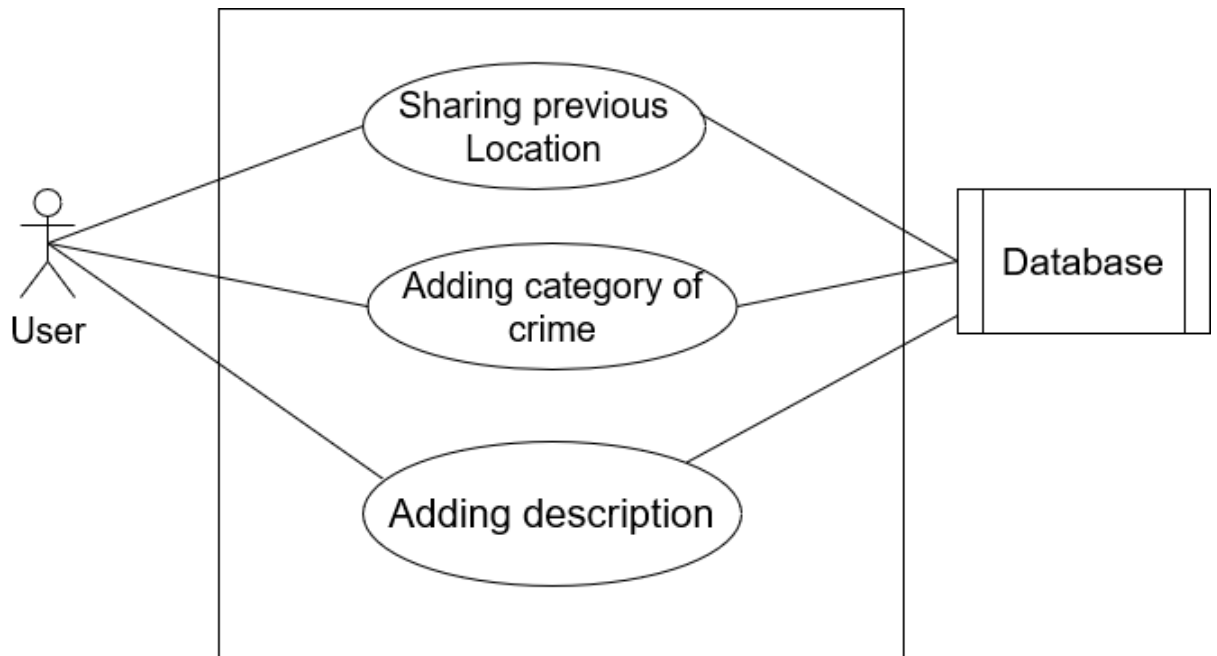


Figure 1.5 Use Case level 1.3

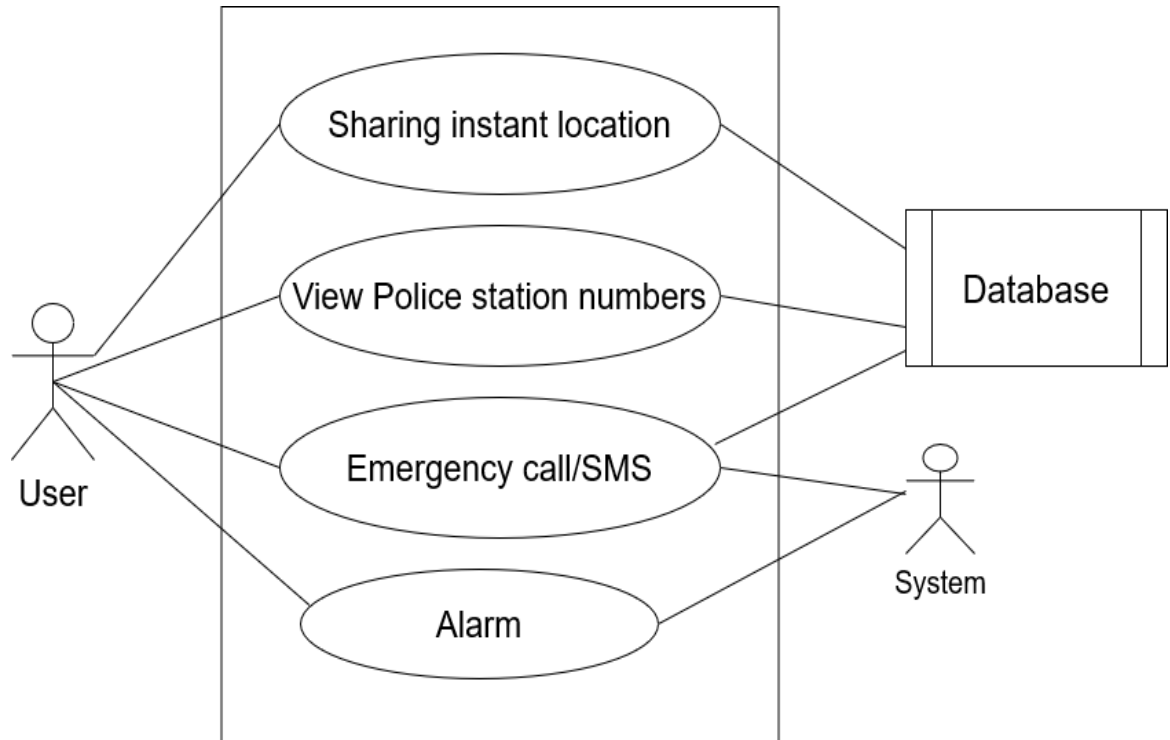


Figure 1.6 Use Case level 1.5

1.4.4 Activity Diagram

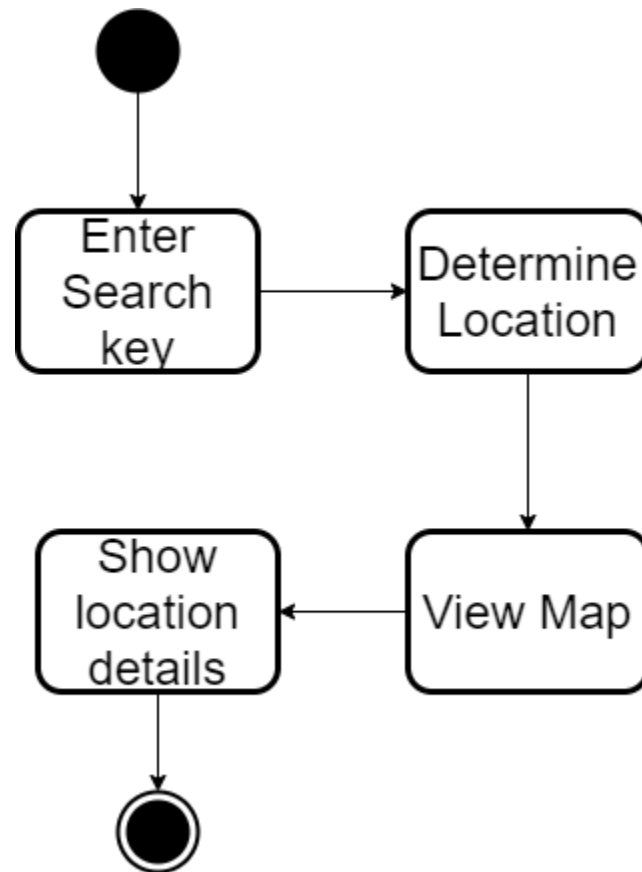


Figure 1.7 Activity Diagram 1

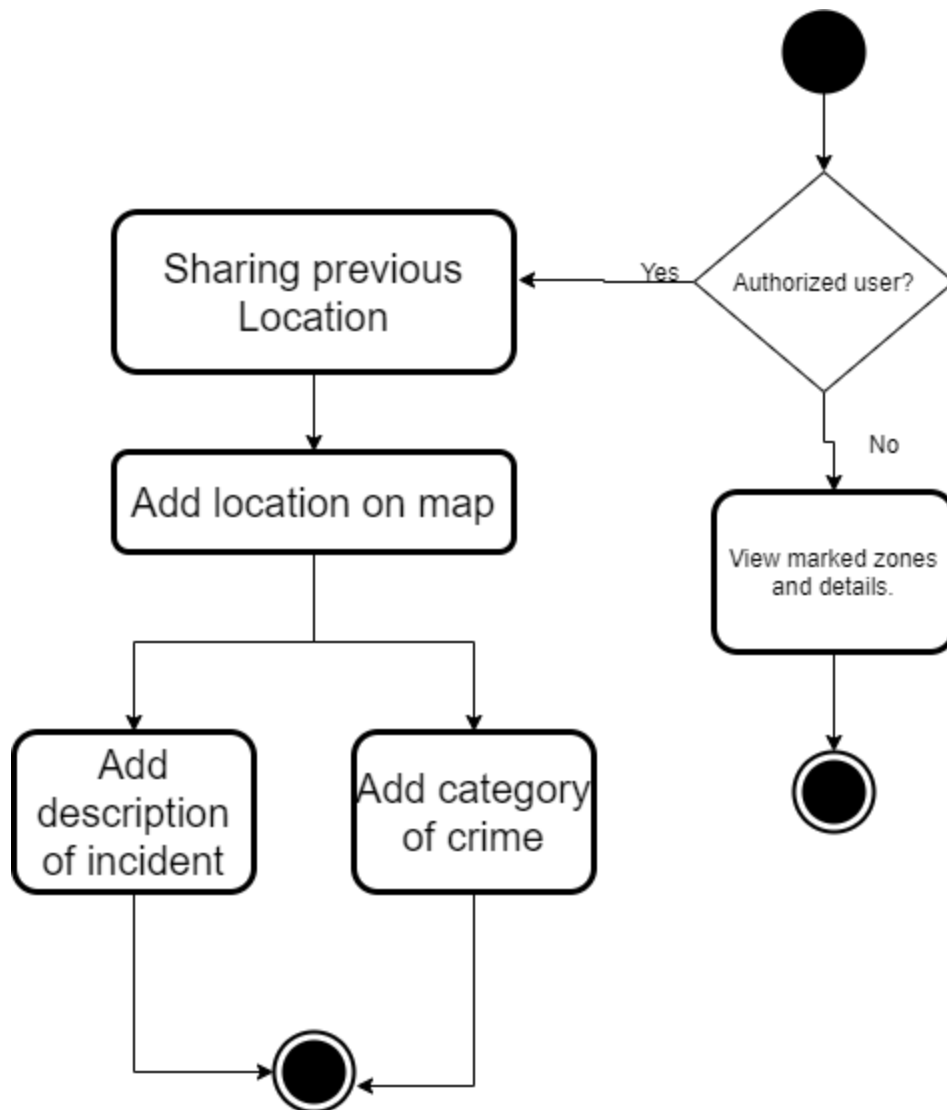


Figure 1.8 Activity Diagram 2

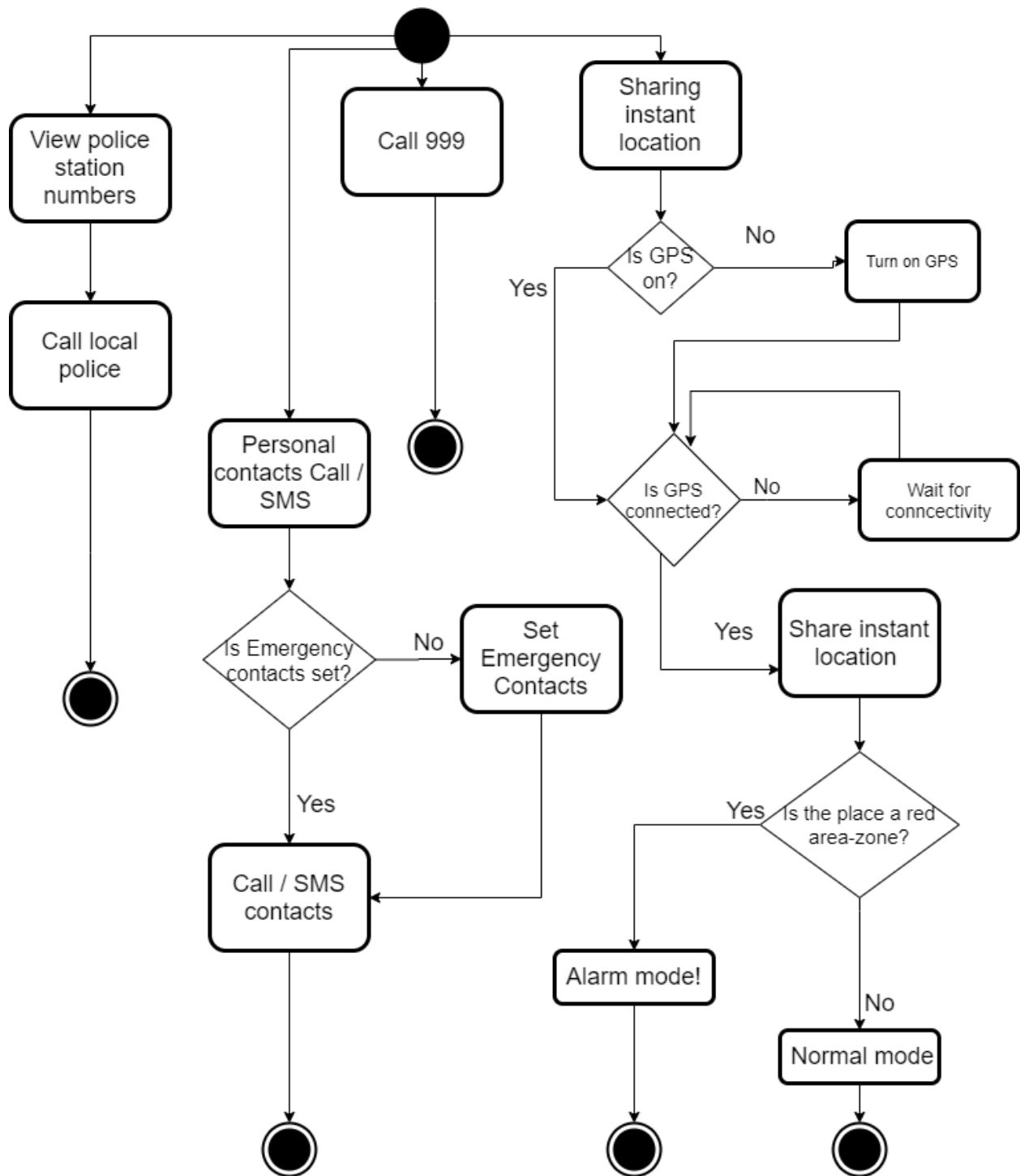


Figure 1.9 Activity Diagram 3

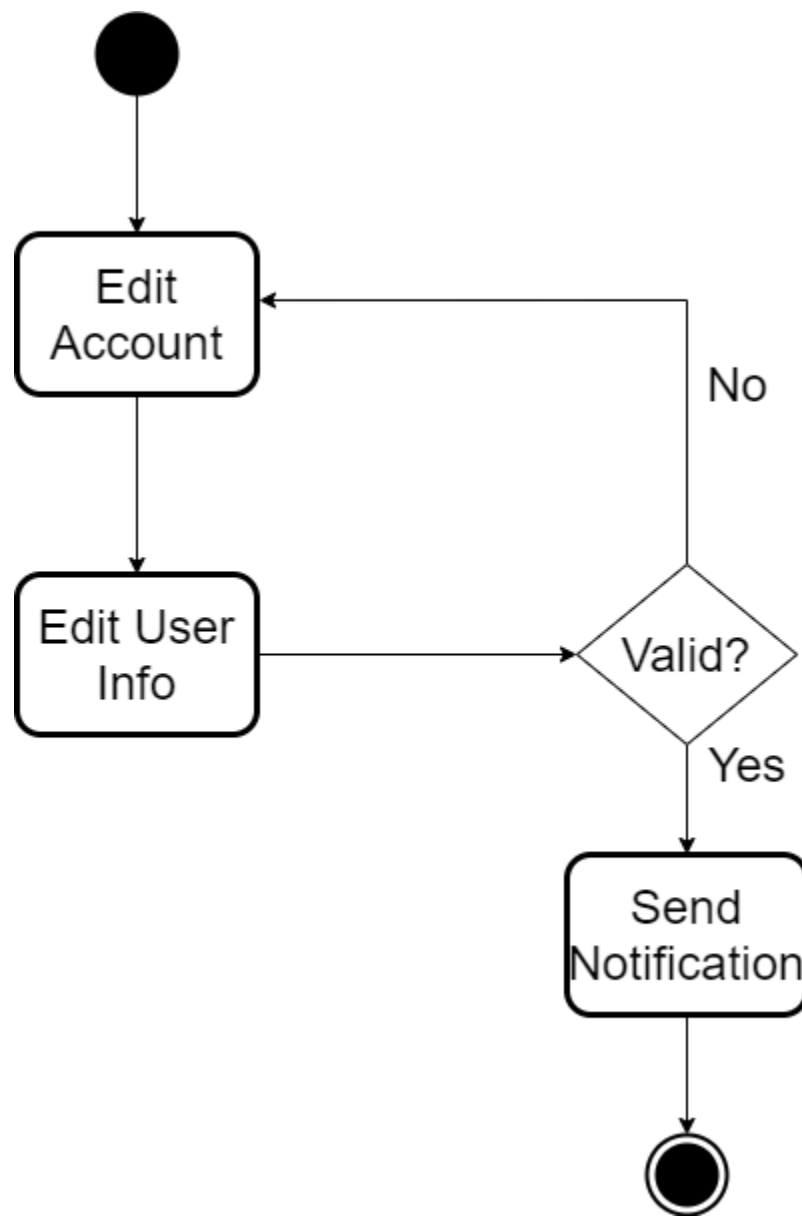


Figure 1.10 Activity Diagram 6

1.4.5 Swimlane Diagram

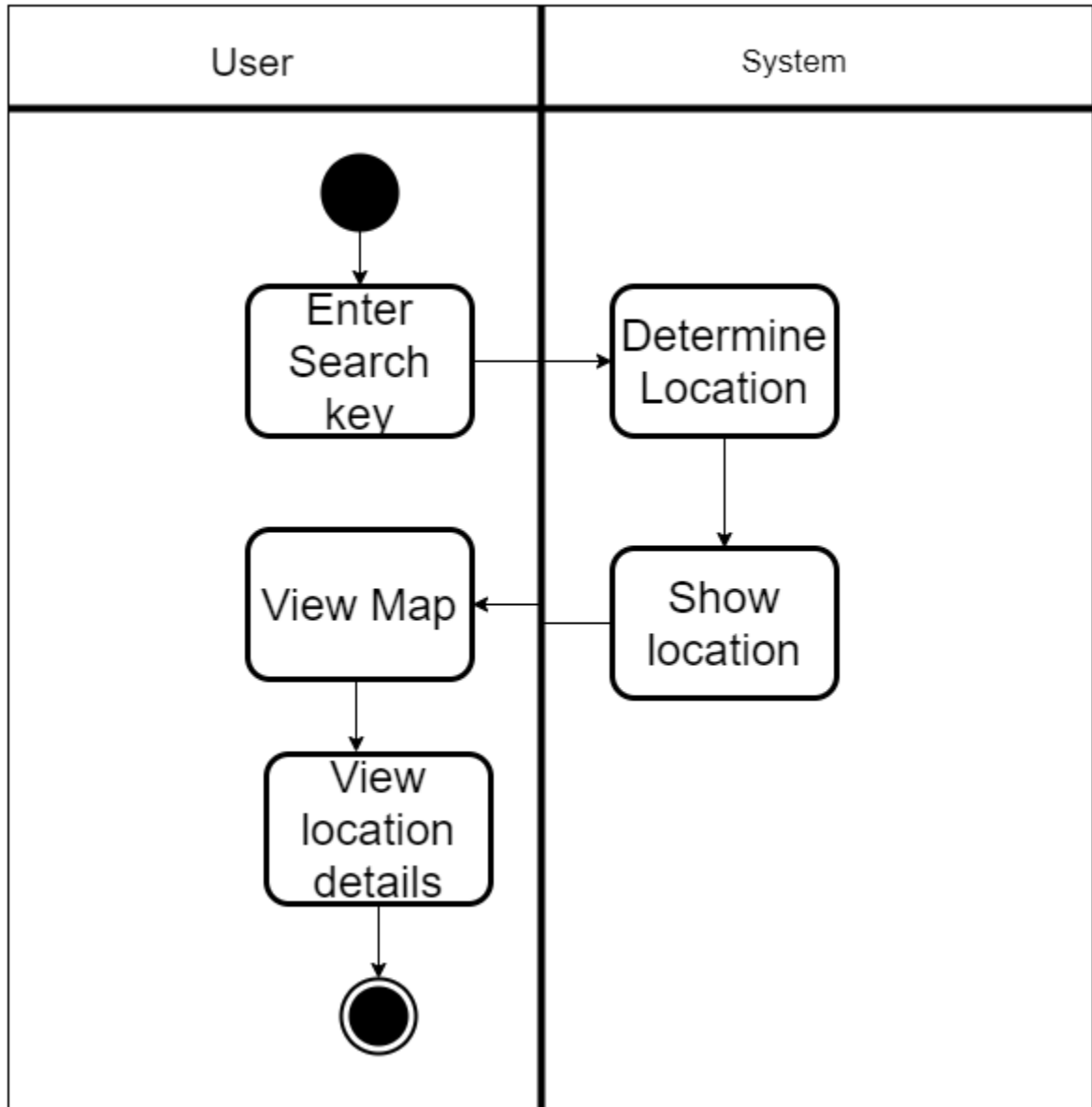


Figure 1.11 Swimlane Diagram 1

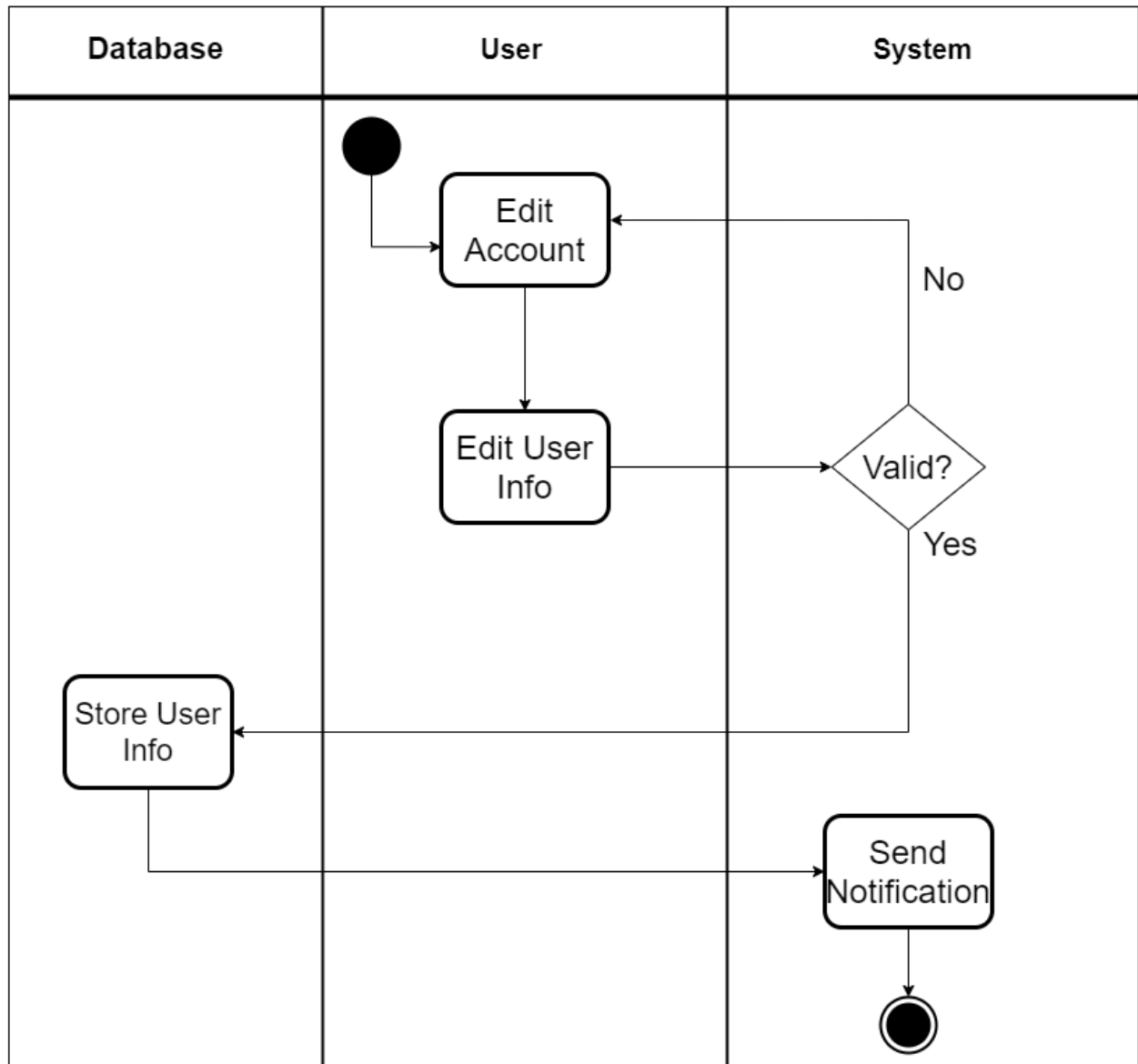


Figure 1.12 Swimlane Diagram 2

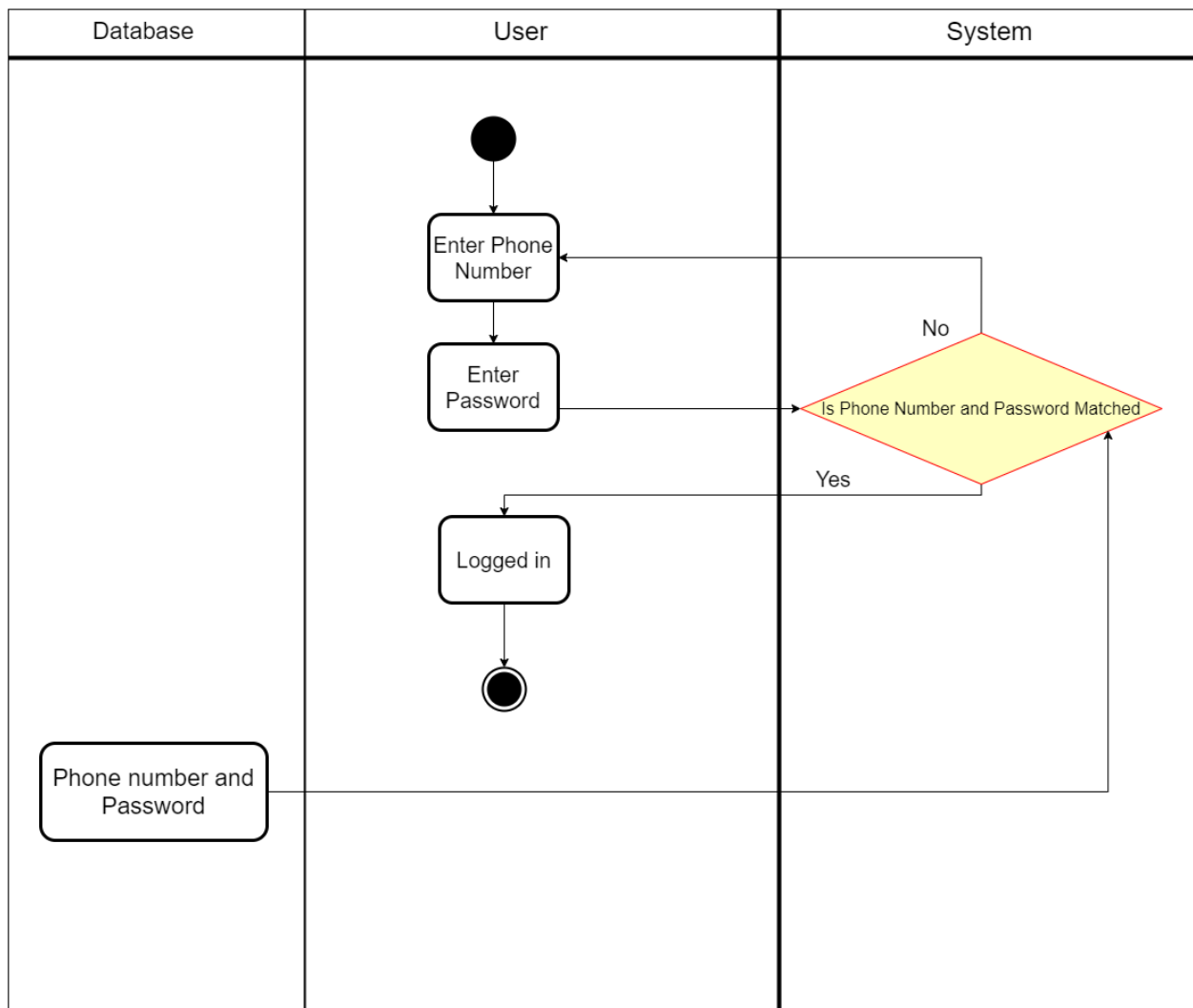


Figure 1.13 Swimlane Diagram 3

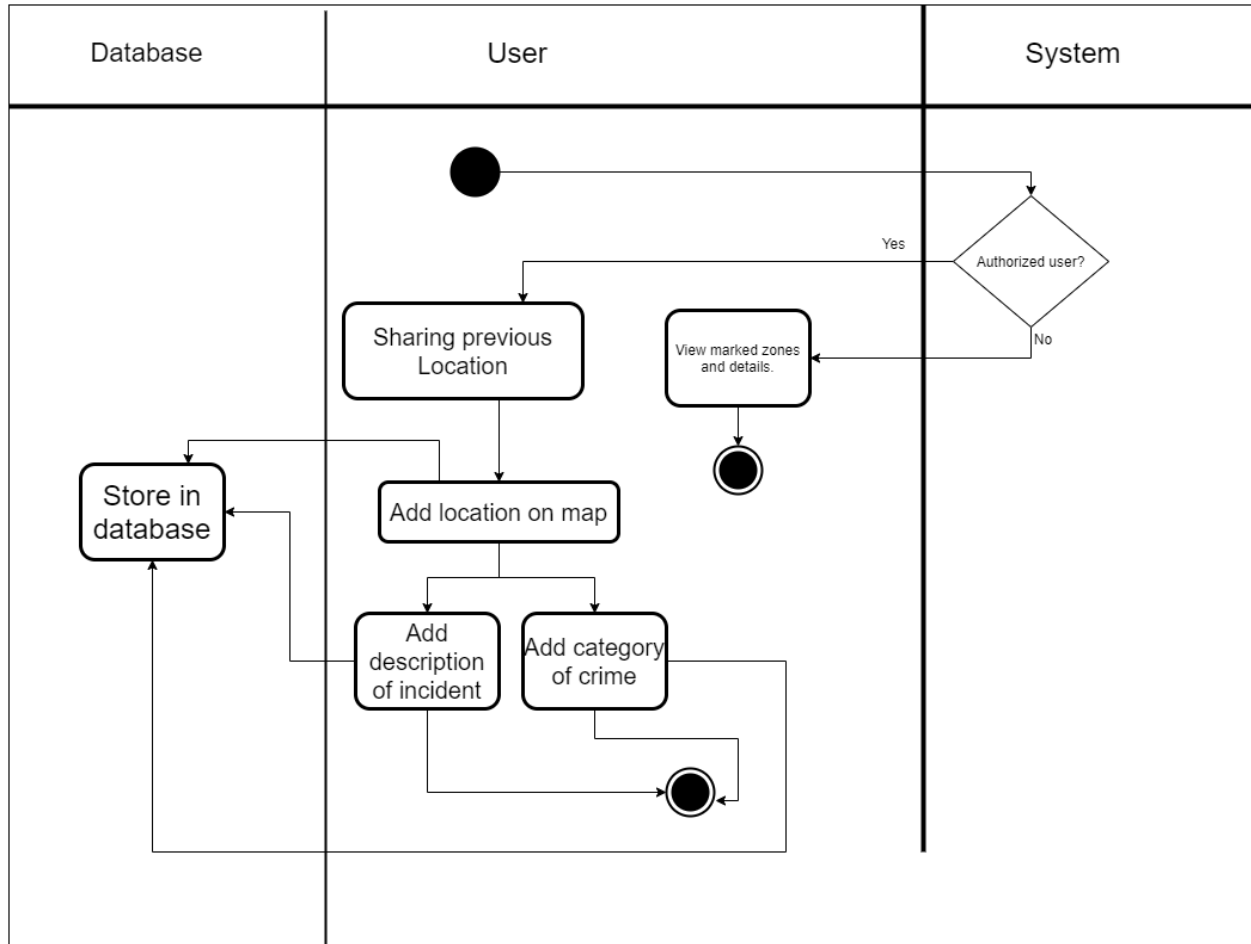


Figure 1.14 Swimlane Diagram 4

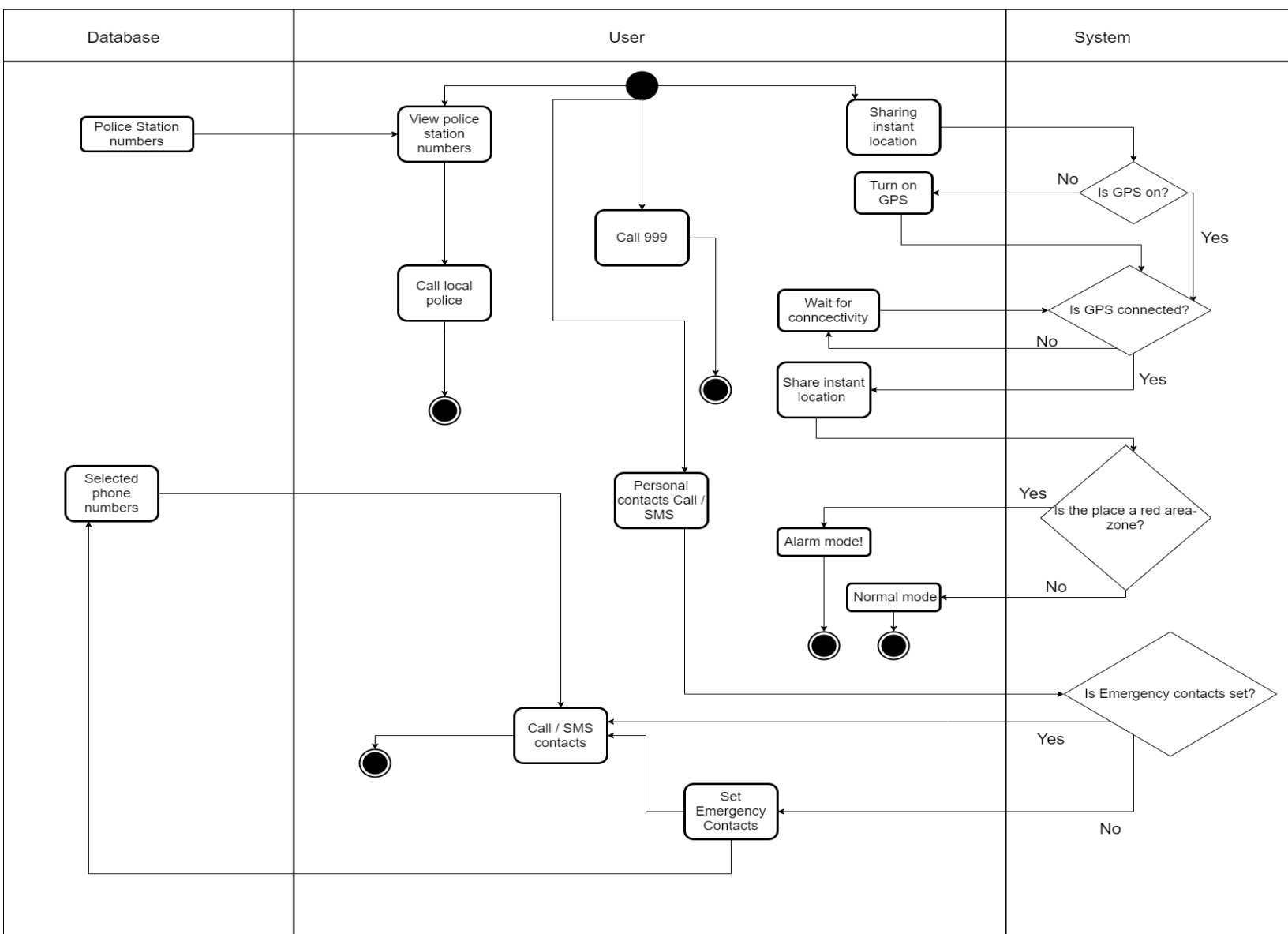


Figure 1.15 Swimlane Diagram 5

Chapter 1.5 Data Modeling

1.5.1 Data modeling concept

If software requirements include the necessity to create, extend or interact with a database or complex data structures need to be constructed and manipulated, then the software team chooses to create data models as part of overall requirements modeling. The entity-relationship diagram (ERD) defines all data objects that are processed within the system, the relationships between the data objects and the information about how the data objects are entered, stored, transformed and produced within the system.

1.5.2 Data objects

A data object is a representation of composite information that must be understood by the software. Here, composite information means an information that has a number of different properties or attributes. A data object can be an external entity, a thing, an occurrence, a role, an organizational unit, a place or a structure.

Noun Identification:

First, we will identify the nouns from the story.

| # | Noun | P/S | Attributes |
|----|-------------------|-----|-----------------------|
| 1 | Android | S | |
| 2 | User | P | 7,8,10,12,13,29,32,33 |
| 3 | Incident | P | |
| 4 | Location | S | 12,13,14 |
| 5 | System | P | |
| 6 | Account | P | |
| 7 | Username | S | |
| 8 | Mobile number | S | |
| 9 | Verification code | S | |
| 10 | Password | S | |

| | | | |
|----|-----------------|---|-------------|
| 11 | Marked zone | P | 12,13,14,30 |
| 12 | Latitude | S | |
| 13 | Longitude | S | |
| 14 | Location_id | S | |
| 15 | Search key | S | |
| 16 | Database | P | |
| 17 | Auto-suggestion | S | |
| 18 | Review | P | 20,19,50 |
| 19 | Category | P | 32,33 |
| 20 | Date | S | |
| 21 | Time | S | |
| 22 | GPS | S | |

| | | | |
|----|---------------------|---|---------|
| 23 | Category | S | |
| 24 | Alarm | S | 35 |
| 25 | Radius | S | |
| 26 | Circumference | S | |
| 27 | Area | S | |
| 28 | Police station | S | 8,14,51 |
| 29 | Emergency contact | S | |
| 30 | Number of incidents | S | |
| 31 | Colour | S | |
| 32 | Gender | S | |
| 33 | Age | S | |
| 34 | Emergency Services | P | 8,36 |

| | | | |
|----|----------------|---|---|
| 35 | Alarm mode | S | |
| 36 | Type | S | |
| 37 | Emergency call | P | 8 |
| 38 | Emergency SMS | P | 8 |
| 39 | Application | | |
| 40 | city | | |
| 41 | features | | |
| 42 | map | | |
| 43 | area | | |
| 44 | Search bar | | |
| 45 | shop | | |
| 46 | harassment | | |

| | | | |
|----|-----------------|---|--|
| 47 | Authorized user | P | |
| 48 | citizen | P | |
| 49 | Text | S | |
| 50 | Review id | S | |
| 51 | Station id | S | |

1.5.2.1 Potential Data Objects:

User: 7,8,10,12,13,29,32,33

Location: 12,13,14

Marked zone: 12,13,14,30

Review: 20,19,50

Category: 32,33

Police station: 8,14,51

Emergency Services: 8,36

1.5.2.2 Analyzing for finalizing Data objects:

- i. Police stations and emergency services can be eliminated.
- ii. Location can be used as an attribute only. And it is also in the review section. And review is also a part of the Users. So this potential data object is also negotiable.

1.5.2.3 Final Data Objects:

| | |
|---|--|
| 1 | User: <u>Mobile number</u> , User name, Password, Latitude_longitude, Review |
| 2 | Marked Zone: <u>Location id</u> , Latitude_longitude, Number of incidents, |

1.5.3 Entity Relationship Diagram:

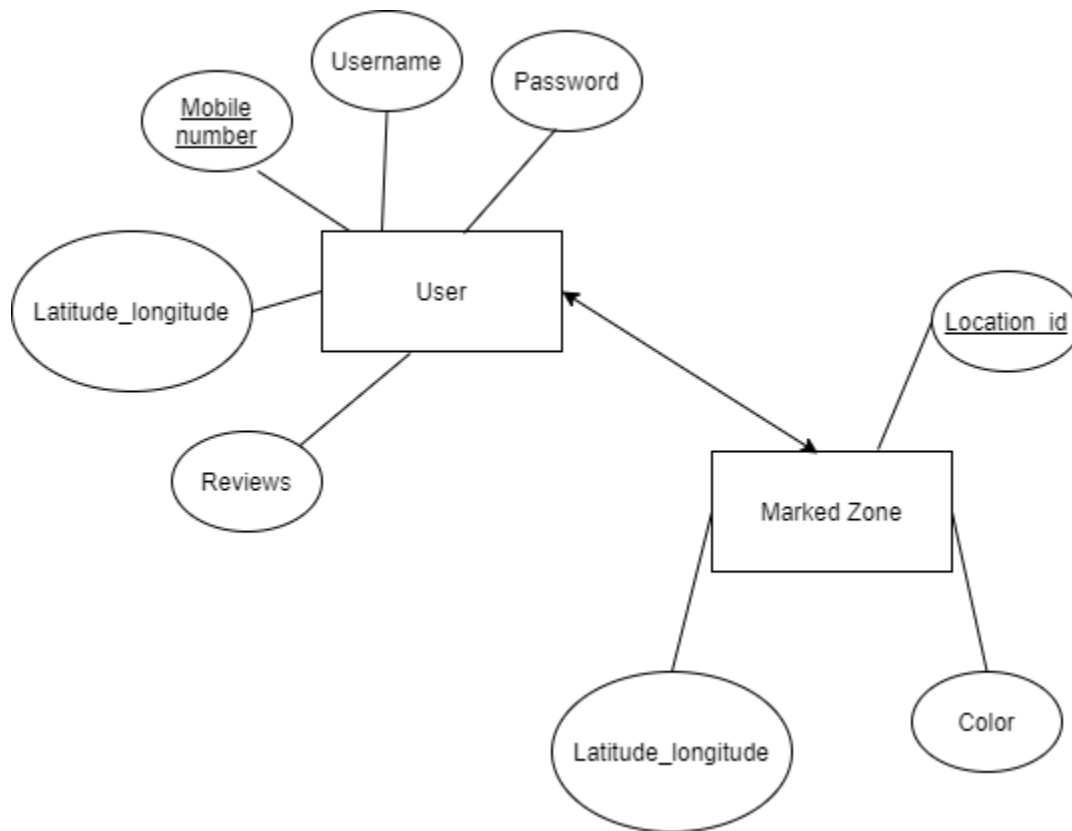


Figure 1.16 Entity Relationship Diagram

1.5.4 Schema Diagram:

| USER | | |
|----------------------|---------|------|
| Attribute | Type | Size |
| Username | Varchar | 50 |
| Password | Varchar | 40 |
| <u>Mobile Number</u> | Varchar | 14 |
| Latitude_longitude | Number | |

| | | |
|--------|--------|--|
| Review | Review | |
|--------|--------|--|

| Review | | |
|--------------------------------|---------|------|
| Attribute | Type | Size |
| Text | Varchar | 512 |
| Date | Date | |
| | | |
| Mobile Number (foreign key) | Varchar | 14 |
| Category id (foreign key) | Number | |

| | | |
|------------------------------|--------|--|
| Location id (foreign key) | Number | |
|------------------------------|--------|--|

| Marked Zone | | |
|---------------------|--------|------|
| Attribute | Type | Size |
| <u>Location_id</u> | Number | |
| Latitude_longitude | Number | |
| Number of Incidents | Number | |

Chapter 1.6 Class Based Modeling

1.6.1 Class Based Modeling Concept

Class-based modeling represents the objects that the system will manipulate, the operations that will be applied to the objects, relationships between the objects and the collaborations that occur between the classes that are defined.

1.6.2 General Classification

To identify the potential classes, nouns were selected 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

Following are the specifications of the nouns according to the general classifications.

| # | Noun | GC |
|----|-------------------|---------|
| 1 | Android | 1 |
| 2 | User | 1,4,5 |
| 3 | Incident | 3 |
| 4 | Location | 2,6 |
| 5 | System | 1,2,4,7 |
| 6 | Account | 2,4 |
| 7 | Username | |
| 8 | Mobile number | |
| 9 | Verification code | |
| 10 | Password | |
| 11 | Marked zone | 2,6 |

| | | |
|----|-----------------|---------|
| 12 | Latitude | |
| 13 | Longitude | |
| 14 | Location_id | |
| 15 | Search key | |
| 16 | Database | 1,2,4,7 |
| 17 | Auto-suggestion | |
| 18 | Review | 2,3 |
| 19 | Category | 2,7 |
| 20 | Date | |
| 21 | Time | |
| 22 | GPS | |
| | | |

| | | |
|----|---------------------|-----|
| 24 | Alarm | 2,3 |
| 25 | Radius | |
| 26 | Circumference | |
| 27 | Area | 6 |
| 28 | Police station | 6,7 |
| 29 | Emergency contact | 2 |
| 30 | Number of incidents | |
| 31 | Colour | |
| 32 | Gender | |
| 33 | Age | |
| 34 | Emergency Services | 2,4 |
| 35 | Alarm mode | |

| | | |
|----|-----------------|---------|
| 36 | Type | |
| 37 | Emergency call | 2,3 |
| 38 | Emergency SMS | 2,3 |
| 39 | Application | 1,2,4,7 |
| 40 | city | 6 |
| 41 | features | 1,2,7 |
| 42 | map | 1,2 |
| 43 | area | 6 |
| 44 | Search bar | |
| 45 | shop | |
| 46 | harassment | |
| 47 | Authorized user | 1,4,5 |

| | | |
|----|------------|---|
| 48 | citizen | 5 |
| 49 | Text | |
| 50 | Review id | |
| 51 | Station id | |

1.6.3 Selection Criteria

The potential classes were then selected as classes by six Selection Criteria. A potential class becomes a class when it fulfills all six characteristics.

1. Retained Information
2. Needed Services
3. Multiple Attributes
4. Common attributes
5. Common operations
6. Essential requirements

| # | Noun | SC |
|---|-------------|-------------|
| 1 | User | 1,2,3,4,5,6 |
| 2 | Location | 1,2,3,4 |
| 3 | System | 1,2,3,5,6 |
| 4 | Account | 1,2,3,4,5,6 |
| 5 | Marked zone | 1,2,3,4,5,6 |
| 6 | Database | 1,3,4,6 |

| | | |
|----|--------------------|-------------|
| 7 | Review | 1,2,3,4,6 |
| 8 | Category | 1,3 |
| 9 | Alarm | 1,2,3,4,6 |
| 10 | Police station | 1,3,6 |
| 11 | Emergency services | 1,2,3,4,5,6 |
| 12 | Emergency call | 1,2,3,4,5,6 |
| 13 | Emergency SMS | 1,2,3,4,5,6 |
| 14 | Application | 1,2,3,5,6 |
| 15 | Features | 1,2,3,4,6 |
| 16 | Authorized user | 1,2,3,4,5,6 |

1.6.4 Associate Noun and Verb Identification

The nouns and the verbs associated with the potential classes are identified to find out the attributes and methods of each class.

| # | Potential Class | Noun | Verb |
|---|--------------------|---|--|
| 1 | User | Mobile number,username,password, latitude_longitude, emergency contacts | create, log in, provide (username, mobile number, password), log out, reset, view, share(location, current location) |
| 2 | System | Database, users, marked zone,services | Verify, calculate, inform, search, update database |
| 3 | Database | Table name, key | Store, retrieve, delete, update, insert |
| 4 | Marked zone | Location id, latitude, longitude, number of incidents | Display, swipe, zoom in, zoom out |
| 5 | Review | Text, date, review id, category | Add, provide, edit |
| 6 | Emergency services | Type, mode | Set emergency contacts, send alarm |

| | | | |
|----|----------------|---|---|
| 7 | Police station | Station id, location id, mobile number | Call, contain |
| 8 | Location | Location id, latitude, longitude | Display, swipe, zoom in, zoom out |
| 9 | Account | Mobile number,username,password, latitude_longitude, emergency contacts | create, log in, provide (username, mobile number, password), log out, reset, view, share(location, current location), get (alarm) |
| 10 | Emergency call | Type | Set emergency contacts |
| 11 | Alarm | Mode | send alarm |

1.6.5 Attribute Selection

| # | Class | Noun |
|---|-----------------------|---|
| 1 | User | Mobile number, Username, Password, Latitude_longitude, emergency contacts |
| 2 | System | Database, Users, marked zone, services |
| 3 | Database | Table name, key |
| 4 | Marked zone | Location id, Latitude_longitude, number of incidents |
| 5 | Review | Text, Date, review id, category |
| 6 | Emergency services | Type, mode |

| | | |
|---|----------------|--|
| 7 | Police station | Station id, location id, mobile number |
|---|----------------|--|

3.6.6 Method Identification:

| # | Potential Class | Methods |
|---|-----------------|---|
| 1 | User | <ul style="list-style-type: none"> • getName() • setName() • getPhoneNumber() • setPhoneNumber() • getPassword() • setPassword() • logIn() • logOut() • reset() • share() • view() |
| 2 | System | <ul style="list-style-type: none"> • initializeDatabase() • verifyInformation() • calculateDistance() • searchLocation() • updateDatabase() • sendNotification() |

| | | |
|---|--------------------|--|
| 3 | Database | <ul style="list-style-type: none"> • insert() • delete() • update() • retrieve() |
| 4 | Marked Zone | <ul style="list-style-type: none"> • DisplayMap() • getUserPosition() • setUserPosition() • getMarkers() • setMarkers() • getRadius() • setRadius() • calculateArea() • swipe() • getPlaceData() • changeView() |
| 5 | Emergency Services | <ul style="list-style-type: none"> • alarm() • getEmergencyContact() • setEmergencyContact() • getUserPostion() • setUserPosition() |

| | | |
|----|----------------|--|
| | | <ul style="list-style-type: none"> • <code>call()</code> • <code>SMS()</code> |
| 6 | Review | <ul style="list-style-type: none"> • <code>getDescription()</code> • <code>setDescription()</code> • <code>edit()</code> |
| 7 | Police Station | <ul style="list-style-type: none"> • <code>getPoliceStationNumber()</code> • <code>setPoliceStationNumber()</code> • <code>call999()</code> • <code>callPoliceStation()</code> |
| 8 | Notification | <ul style="list-style-type: none"> • <code>generateMessage()</code> • <code>parseMessage()</code> • <code>sendNotification()</code> |
| 9 | Alarm | <ul style="list-style-type: none"> • <code>getAlarm()</code> • <code>setAlarm()</code> |
| 10 | Emergency Call | <ul style="list-style-type: none"> • <code>getEmergencyContact()</code> • <code>setEmergencyContact()</code> |

| | | |
|----|---------|---|
| 11 | Account | <ul style="list-style-type: none"> • getName() • setName() • getPhoneNumber() • setPhoneNumber() • getPassword() • setPassword() • logIn() • logOut() • reset() • share() • view() |
|----|---------|---|

1.6.7 Finalizing Classes

To identify the final classes, it was required to check if there can be any hierarchies, merges, additional attributes, methods or classes. These identifications are given below:

1. No separate class for parent class (Emergency service) and child (Emergency call, alarm) is required as the functionalities are similar and the difference can be compared using a tag. So we can merge Emergency calls and alarms as Emergency service.
2. User and account class have almost the same attributes and methods. As we have only one type of user account, there is no need to keep user and account as separate classes. User class will hold all necessary account information.
3. We won't keep location as a separate class for storing all users' current location value. Location information of marked zones will be stored only. Users' current location will be stored as an attribute of the user.

1.6.8 Class Cards

| User | |
|---|---|
| Attributes | Methods |
| <ul style="list-style-type: none">• Mobile number,• Username,• Password,• Latitude_longitude,• Emergency contacts | <ul style="list-style-type: none">• getName()• setName()• getPhoneNumber()• setPhoneNumber()• getPassword()• setPassword()• logIn()• logOut()• reset()• share()• view() |
| Responsibility | Collaborator |
| <ul style="list-style-type: none">• Updating user information(full name, phone number, password)• Getting user's location | <ul style="list-style-type: none">• System• Database• Marked Zone |

| | |
|---|--|
| <ul style="list-style-type: none"> • Log in • Log out | |
|---|--|

| | |
|---|--|
| System | |
| Attributes | Methods |
| <ul style="list-style-type: none"> • Database, • Users, • Marked zone, • Services | <ul style="list-style-type: none"> • initializeDatabase() • verifyInformation() • calculateDistance() • searchLocation() • updateDatabase() • sendNotification() |
| Responsibility | Collaborator |
| <ul style="list-style-type: none"> • Sign up • Sign in • Update database • Search place | <ul style="list-style-type: none"> • User • Database • Marked Zone |

| | |
|---|--|
| Database | |
| Attributes | Methods |
| <ul style="list-style-type: none"> • Table name, • key | <ul style="list-style-type: none"> • insert() • delete() • update() • retrieve() |
| Responsibility | Collaborator |
| <ul style="list-style-type: none"> • Insert data • Update data • Delete data | <ul style="list-style-type: none"> • Review • User • Marked Zones • Police Station |

| | |
|--|--|
| Marked Zone | |
| Attributes | Methods |
| <ul style="list-style-type: none"> • Location id, • Latitude_longitude, • number of incidents | <ul style="list-style-type: none"> • DisplayMap() • getUserPosition() • setUserPosition() • getMarkers() • setMarkers() • getRadius() • setRadius() • calculateArea() • swipe() • getPlaceData() • changeView() |
| Responsibility | Collaborator |

| | |
|--|--|
| <ul style="list-style-type: none"> • Browse map • Get location information • Track user • Show user position • Load place data • Crime intensity range | <ul style="list-style-type: none"> • System • Review • Database |
|--|--|

| | |
|--|---|
| Emergency Services | |
| Attributes | Methods |
| <ul style="list-style-type: none"> • Type, • Mode • Latitude_longitude • Location id • EmergencyContact | <ul style="list-style-type: none"> • alarm() • getEmergencyContact() • setEmergencyContact() • getUserPostion() • setUserPosition() • call() • SMS() |
| Responsibility | Collaborator |

| | |
|--|--|
| <ul style="list-style-type: none"> • Find user's location • Alarm user • Provide emergency contacts | <ul style="list-style-type: none"> • User • System |
|--|--|

| | |
|--|--|
| Review | |
| Attributes | Methods |
| <ul style="list-style-type: none"> • Latitude • Longitude • Text • Date • review id • category | <ul style="list-style-type: none"> • getDescription() • setDescription() • edit() |

| Responsibility | Collaborator |
|---|---|
| <ul style="list-style-type: none"> • input information about incidents • identify the category of crime | <ul style="list-style-type: none"> • Marked Zone • Database • User |

| Police Station | |
|---|--|
| Attributes | Methods |
| <ul style="list-style-type: none"> • Station id, • location id, • mobile number | <ul style="list-style-type: none"> • getPoliceStationNumber() • setPoliceStationNumber() • call999() • callPoliceStation() |
| Responsibility | Collaborator |
| <ul style="list-style-type: none"> • Giving police station numbers • Quick dial option to 999 | <ul style="list-style-type: none"> • Database |

Chapter 1.7 Behavioral Modeling:

The behavioral model indicates how software will respond to external events.

| Serial No. | Event | State name | Initiator | Collaborator |
|------------|---------------------------------|----------------------|-----------|-------------------------------|
| 1. | create an account | Sign Up | User | System, Database |
| 2. | verify the mobile number | Account Verification | System | User, Database |
| 3. | can login | Login | User | System, Database |
| 4. | will be a logout option | Logout | User | |
| 5 | Forgotten password can be reset | Reset Password | User | System, Database |
| 6. | Can only view | View | User | System, Database, Marked Zone |
| 7. | Require user approval | Approval | System | User |
| 8. | Swipe the map | Swipe | User | System, Marked Zone |

| | | | | |
|-----|--|--------------------|--------|-------------------------------------|
| 9. | Can zoom in and zoom out | Zoom in Zoom out | User | Marked Zone, System |
| 10. | Search for location | Search | User | Marked Zone, Database, System |
| 11. | Auto suggestion | Suggestion | System | Database, User, Marked Zone |
| 12. | Set Emergency contact | Emergency Contact | User | Emergency Service, Database, System |
| 13. | Share Location with some reviews | Share Review | User | Review, Database, Marked Zone |
| 14. | Can add gender, date, time and category of crime | Review description | User | Review, Database |
| 15. | Can edit review information | Edit Review | User | Review |
| 16. | Update information | Update | User | Review, Database, Marked Zone |
| 17. | Calculate the area zones | Calculate Zone | System | Database, Review, Marked Zone |

| | | | | |
|-----|--|----------------|-------------------|----------------------------------|
| 18. | Get alarm | Alarm | Emergency Service | User, System, Marked Zone |
| 19. | Can share that instant location | Share Location | User | System, Marked Zone, Database |
| 20. | View all local Police Station Numbers | Police Number | System | Police Station, Database |
| 21. | Call Police Station | Call Police | User | Police Station, Database, System |
| 22. | Direct call button to 999 | Call 999 | User | System, Police Station |
| 23. | The database will store information | Store Data | System | Database |
| 24. | The database will insert information | Insert Data | System | Database |
| 25. | The database will update information | Update Data | System | Database |
| 26. | The database will delete information | Delete Data | System | Database |
| 27. | The database will retrieve information | Retrieve Data | System | Database |

1.7.1 State Transition Diagram

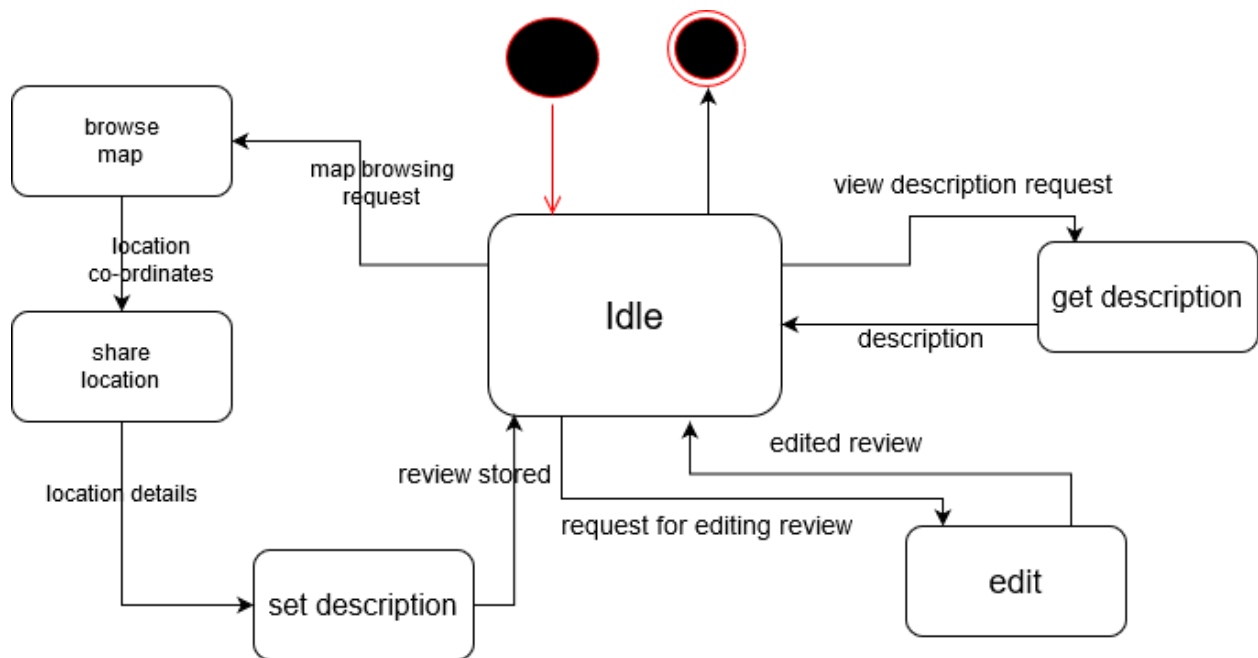


Figure 1.17: State transition for .Review class

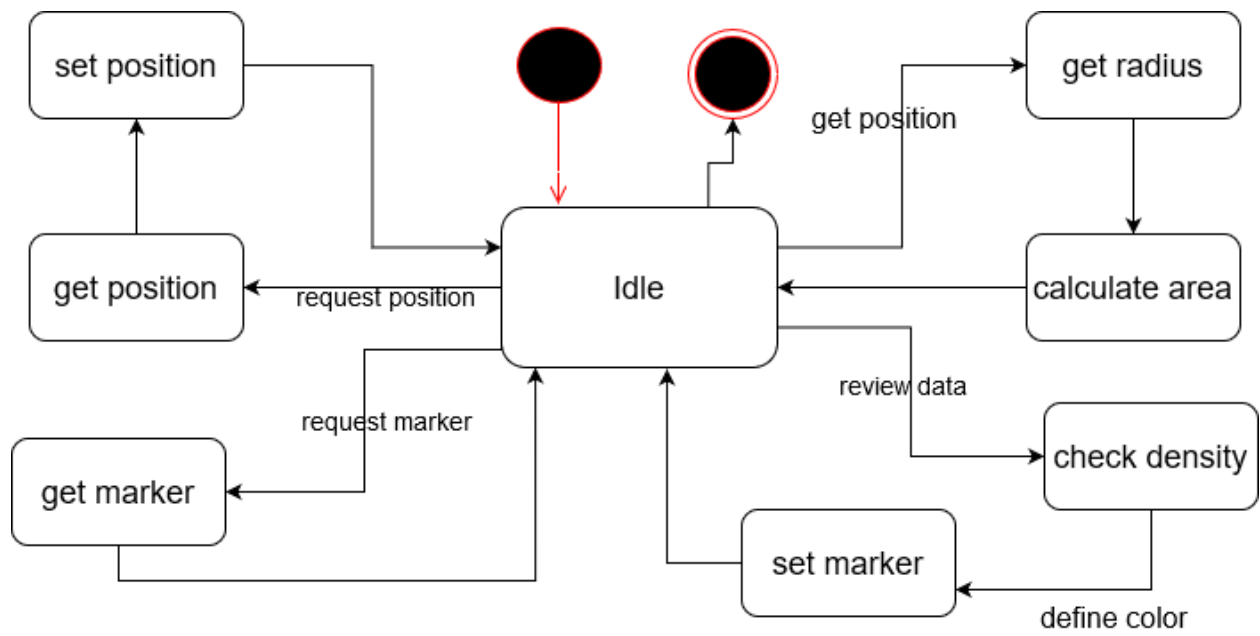


Figure 1.18 State transition for Marked Zone class.

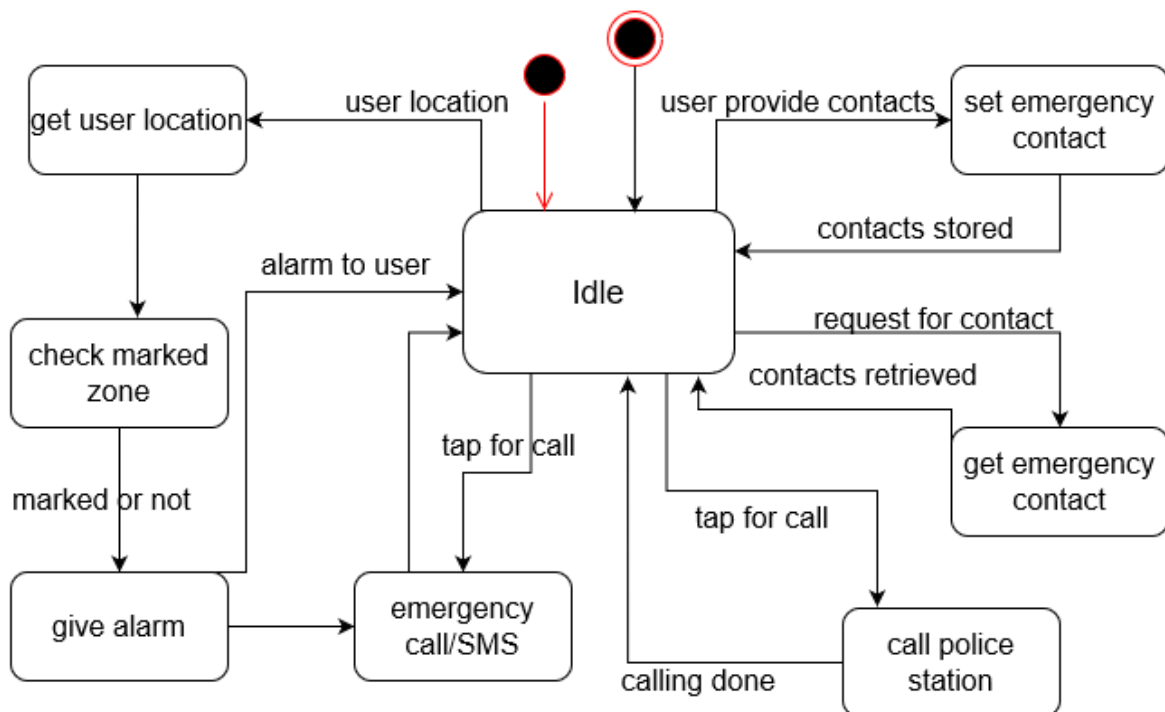


Figure 1.19 State transition for Emergency Service class.

1.7.2 Sequence Diagram

