

SITUATIONAL AWARENESS SYSTEM



By

Zainab Ijaz

Hamza Malik

Hamna Asher

Submitted to the Faculty of Computer Software Engineering

National University of Sciences and Technology, Islamabad in partial fulfillment

For the requirements of a B.E. Degree in Computer Software Engineering

JUNE 2017

ABSTRACT

Mobility has become a basic premise of communications, thereby requiring a native integration into wifi and 3G/4G networks. Despite the numerous efforts to propose and to standardize effective mobility management models through secure channels, the result is a complex, hack-able, poorly flexible set of mechanisms which, if used by the military personnel for classified data can end up making the operation vulnerable and putting everyone at risk. The natural support for mobility offered by **SAS** makes it a good candidate to define a radically new solution relieving limitations of traditional approaches for navigation in unknown location by the military.

The project aims at developing an android application for the purpose of data collection. The application uses the OSM (Open Street Maps) tiles which are stored in the device. The tiles are loaded once the application starts. The administrator logs in to the system and has the authority to add, delete and edit details of the system. The users log in as guests and can only search for locations and view their location descriptions. The current location is displayed on the map as the application starts.

CERTIFICATE FOR CORRECTNESS AND APPROVAL

Certified that work contained in the thesis – Situational Awareness System carried out by Hamna Asher, Zainab Ijaz and Hamza Malik in supervision of Lt. Col. Adnan Rashdi for partial fulfilment of Degree of Bachelor of Software Engineering is correct and approved.

Approved by

Lt. Col. Adnan Rashdi

CSE DEPARTMENT

MCS

DATED:

DECLARATION

No portion of the work presented in this dissertation has been submitted in support of another award or qualification either at this institution or elsewhere.

DEDICATION

In the name of Allah, the Most Merciful, the Most Beneficent

To our parents, without whose unflinching support and cooperation,

a work of this magnitude would not have been possible.

ACKNOWLEDGEMENTS

We would like to thank Allah Almighty for His incessant blessings which have been bestowed upon us. Whatever we have achieved, we owe it to Him, in totality. We are also thankful to our families for their continuous moral support which makes us what we are.

We are extremely grateful to our project supervisor Lt. Col. Adnan Rashdi from MCS who in addition to providing us with valuable technical help and guidance also provided us moral support and encouraged us throughout the development of the project.

We are highly thankful to all of our teachers and staff of MCS who supported and guided us throughout our course work. Their knowledge, guidance and training enabled us to carry out this whole work.

Finally we are grateful to the faculty of Computer Software Department of the Military College of Signals, NUST.

In the end we would like to acknowledge the support provided by all our friends, colleagues and a long list of well-wishers whose prayers and faith in us propelled us towards our goal.

TABLE OF CONTENTS

1	INTRODUCTION.....	2
1.1	Overview	2
1.2	Problem Statement	2
1.3	Approach	2
1.4	Scope	2
1.5	Aim & Objectives.....	3
1.6	Contributions:.....	3
1.7	Organization.....	3
1.8	Deliverables.....	4
2	LITERATURE REVIEW.....	6
2.1	OSMDroid.....	6
2.2	Tile Source	7
3	OVERALL DESCRIPTION	9
3.1	Introduction	9
3.1.1	Purpose.....	9
3.1.2	Intended Audience and Reading Suggestions.....	9
3.1.3	Intended audience	9
3.1.4	Project Supervisor.....	10
3.1.5	Testers:	10

3.1.6	Documentation writers:.....	10
3.1.7	Project Testers:.....	10
3.1.8	UG Project Evaluation Team	10
3.1.9	Reading suggestions.....	10
3.1.10	Product Scope	11
3.2	Overall Description	11
3.2.1	Product Perspective.....	11
3.2.2	Product Functions	12
3.2.3	User Classes and Characteristics	12
3.2.4	Operating Environment.....	13
3.2.5	Design and Implementation Constraints.....	13
3.2.6	User Documentation	14
3.2.7	Assumptions and Dependencies	14
3.3	External Interface Requirements.....	15
3.3.1	User Interfaces	15
3.3.2	Hardware Interfaces	15
3.4	System Features.....	16
3.4.1	Search.....	16
3.4.2	Navigation to destination	17
3.4.3	Location Description.....	17

3.4.4	Search bar for searching specific locations.....	18
3.4.5	Current Location	19
3.4.6	Access the central database.....	19
3.4.7	Administrator sign in	20
3.4.8	Profile page	21
3.4.9	Storing Tags	21
3.4.10	Functional Requirements	22
3.4.11	Modifying tags/locations	22
3.4.12	Adding tags/locations.....	23
3.5	Other Nonfunctional Requirements	23
3.5.1	Performance Requirements.....	23
3.5.2	Safety Requirements	24
3.5.3	Security Requirements	25
3.5.4	Software Quality Attributes	25
3.6	Business Rules.....	25
4	DESIGN AND DEVELOPMENT	27
4.1	INTRODUCTION:.....	27
4.1.1	Purpose of the document:.....	27
4.1.2	Scope of the Development Project.....	27
4.1.3	Definitions, acronyms, abbreviations	28

4.1.4	Overview of the document.....	28
4.2	SYSTEM ARCHITECTURE DESCRIPTION.....	29
4.2.1	OVERVIEW OF MODULES/COMPONENTS	29
4.3	STRUCTURE AND RELATIONSHIPS.....	31
4.3.1	System Block Diagram	31
4.4	Architectural Style:	32
4.4.1	System Architecture:.....	32
4.4.2	Architecture Diagram:	33
4.5	UML Diagrams:	34
4.5.1	User view (use case diagram).....	34
4.5.2	Use Case Specification:	34
4.5.3	Use Case Description.....	35
4.5.4	Sequence diagram	41
4.5.5	Login success/failure (System administrator).....	42
4.5.6	Add location via GPS.....	43
4.5.7	View and editing admin profile	43
4.5.8	Search tags	44
4.5.9	Edit tags	45
4.5.10	Delete tags.....	45
4.5.11	Logical View:.....	51

4.5.12	State Transition Diagram	51
4.5.13	Implementation View.....	54
4.5.14	Class Diagram.....	54
4.5.15	System Classes Description:	55
4.5.16	Dynamic View:	56
4.5.17	Activity Diagram	56
4.6	User interface	57
4.7	Reuse and Relationship to other products	63
4.8	Design and tradeoffs.....	63
5	SYSTEM IMPLEMENTATION	66
5.1.1	For User Interface	66
5.1.2	For Search Location	66
5.1.3	For Location Description	67
5.1.4	For Add Location	67
5.1.5	For Edit Location	68
5.1.6	For Admin Profile	68
5.1.7	For Synchronization with Central Database	69
5.1.8	For tag current location.....	69
6	Analysis and Evaluation	71
6.1	Introduction	71

6.2	Approach	72
6.3	Features to be tested	73
6.4	Item Pass/Fail Criteria.....	74
6.5	Testing tasks	74
6.6	Test Deliverables.....	74
6.7	Responsibilities	74
6.8	Staffing and training needs.....	75
6.9	Risks and contingencies	75
6.9.1	Schedule Risk:	75
6.9.2	Budget Risk:.....	75
6.9.3	Operational Risks:.....	75
6.9.4	Technical risks:	75
6.9.5	Programmatic Risks:.....	75
6.10	Environmental Needs	76
6.10.1	Hardware.....	76
6.10.2	Software	76
6.11	Risks and contingencies	76
6.12	Test Cases:.....	76
7	FUTURE WORK	88
8	CONCLUSION.....	90

8.1	Overview	90
8.2	Objectives Achieved	90
9	BIBLIOGRAPHY	92
	APPENDIX A	93
	USER MANUAL	93
10	GENERAL INFORMATION	97
10.1	System Overview:	97
10.2	Organization of the manual:	97
11	SYSTEM SUMMARY	99
11.1	System Configuration:	99
11.2	User Access Levels:	99
11.3	Contingencies:	99
12	GETTING STARTED	101
12.1	Installation:	101
12.2	System Menu:	102
12.3	Exit Application:	102
13	USING THE SYSTEM:	104

TABLE OF FIGURES:

Figure 4-1 Component Overview for SAS	30
Figure 4-2 Block Diagram for SAS	32
Figure 4-3 Architecture Diagram for SAS.....	33
Figure 4-4 Use case Diagram for SAS.....	34
Figure 4-5 Sequence Diagram for Login success/failure.....	42
Figure 4-6 Sequence Diagram for Add location via GPS.....	43
Figure 4-7 Sequence Diagram for view and edit admin profile.....	44
Figure 4-8 Sequence Diagram for search.....	44
Figure 4-9 Sequence Diagram for Edit tags.....	45
Figure 4-10 Sequence Diagram for delete tags.....	45
Figure 4-11 Sequence Diagram for route between current and tagged location.....	46
Figure 4-12 Sequence Diagram for route between two searched locations.....	47
Figure 4-13 Sequence Diagram for successful synchroniization.....	48
Figure 4-14 Sequence Diagram for unsuccessful synchroniization.....	48
Figure 4-15 Sequence Diagram for search location(user)	49
Figure 4-16 Sequence Diagram for route between current and tagged location(user)	50
Figure 4-17 Sequence Diagram for route between two searched locations(user)	51
Figure 4-18 State Transition Diagram for user	Error! Bookmark not defined.
Figure 4-19 State Transition Diagram for system administrator	53
Figure 4-20 Class Diagram for SAS	54
Figure 4-21 Activity Diagram for System Administrator.....	56
Figure 4-22 Activity Diagram User	Error! Bookmark not defined.

Figure 4-23 Main menu screen	Error! Bookmark not defined.
Figure 4-24 Map view screen	Error! Bookmark not defined.
Figure 4-25 Drawer Activity screen	60
Figure 4-26 Location description screen.....	61
Figure 4-27 Admin profile screen.....	62

TABLE OF TABLES:

Table 1-1 : Deliverables	4
Table 4-1 System Classes Description.....	55
Table 6-1: Test Cases For Accessing login screen	77
Table 6-2: Test Cases for search location	78
Table 6-3: Test Cases For navigation	78
Table 6-4: Test Cases For add location by long press	79
Table 6-5: Test Cases For add current location	80
Table 6-6: Test Cases For display the additional information	81
Table 6-7: Test Cases For display error message for invalid search	82
Table 6-8: Test Cases For display the current position.....	83
Table 6-9: Test Cases For edit the tagged locations	83
Table 6-10: Test Cases For delete the tagged locations.....	84
Table 6-11: Test Cases For synchronization with central database	85
Table 6-12: Test Cases For edit admin profile.....	86

CHAPTER: 1
INTRODUCTION

1 INTRODUCTION

1.1 Overview

Situational Awareness System is developed to help military personnel with data collection of sensitive locations and their details that are vulnerable to outside influence and therefore need high security. Our application will be in offline mode, using tags from local database. A centralized database will be maintained which will synchronize with the local database through a secure connection only.

1.2 Problem Statement

The purpose of this application is to make it simple and easy for the end user to locate nearby places by using proprietary imagery while providing high security for the military sensitive location. Our system will be implemented on the Android Mobile Phone Platform. The System will provide navigational support to the user to select route to destination.

1.3 Approach

The main focus of the project is to provide a mean to the military corps and personnel through which they will locate key positions of specific areas stored in the database. The database will increase as it will be updated and synchronized over time.

1.4 Scope

This application will be used to navigate around an unknown area using a map and add new locations. Additional information will be added with the tagged place to help users analyze the data and finding appropriate ways for contacting a location.

1.5 Aim & Objectives

The objectives of project include:

Using software engineering techniques for gathering requirements during the development process, designing the software, implementing and testing requirements gathered.

To learn java programming and SDLC

To learn Techniques of Android platform and SQLite

To learn database design and development

To learn data transfer through networking.

1.6 Contributions:

This is a military project, designed and developed for the MCS R&D.

1.7 Organization

The first part of thesis is the abstract which describes the main details of SAS, followed by the introduction section which specifies the problem statement, approach, scope and objectives. The literature review section state the various resources read online before the commencement of the project. They include learning about OSMDroid and Tile Sources. The design and development part illustrate the diagrams which describe the detailed design of the SAS its components, interfaces and data necessary for the implementation phase. The analysis and evaluation part give details of the black box testing, unit testing and system integration testing; actual results against expected results. The future work gives states the enhancements that can be applied to the application.

1.8 Deliverables

Table 1-1 : Deliverables

Deliverable Name	Deliverable Summary Description
Software Requirements Specification(SRS) Document	Complete Description of what the system will do, who will use it. Detailed description of functional and non-functional requirements and the system features.
Design Document	Complete description of how the system will be implemented i.e. the detailed design.
Code	Complete code with the API.
Testing Document	The whole system is tested according to the specification described in the SRS document. Black box, unit and System integration testing is done.
Complete System	Complete working system.

CHAPTER:2
LITERATURE REVIEW

2 LITERATURE REVIEW

For building a GIS application we had to do research on available APIs we can use to display the map and program our features in it. We also needed a downloaded tile source which would be itself the map. After a lot of research we concluded on using OSMDroid APIs for the application.

2.1 OSMDroid

OSMDroid is a (almost) full/free replacement for Android's Map View (v1 API) class. It also includes a modular tile provider system with support for numerous online and offline tile sources and overlay support with built-in overlays for plotting icons, tracking location, and drawing shapes. OSMDroid is a library to interact with Open Street Map data inside an Android application. It offers an almost full/free replacement to Android map objects: Map View, Map Controller, Overlays (Marker, Polyline, Polygon), etc.

It also features an "OSMBonusPack" library for extra features.

This "OSMBonusPack" library complements osmdroid with (very) useful classes:

- Routes and Directions,
- Points of Interests (directory services),
- Marker Clustering,
- Support for KML and GeoJSON content,
- GroundOverlay, similar to its Google Maps equivalent,
- Geocoding and Reverse Geocoding,

2.2 Tile Source

Since our requirements by the R&D were not to use google maps in our application our main issue was which maps to use for our navigation system. Hence, we studied on downloadable maps which can offer us the features we needed to add to our application. We had to find a multi-layer map which provided street level information and an updated road network whilst displaying a high quality map for viewing. Not only this but it also had to be compatible with the OSMDroid APIs we were using to display the maps and add our specific features to it such as tagging.

Our first and obvious choice after our research concluded that we needed to use Mapnik or Mapquest map source for the application as it was the most used. But since it had been banned we resorted to using single layer satellite imagery provided by our institution.

CHAPTER:3
OVERALL DESCRIPTION

3 OVERALL DESCRIPTION

This part of the document contains information about the product, its features, perspective, users' characteristics and constraints.

3.1 Introduction

3.1.1 Purpose

The purpose of this application is to make it simple and easy for the end user to add locations by using proprietary imagery of sensitive areas. Our system will be implemented on the Android Mobile Phone Platform. The System will provide real time navigational support to the user.

This document is aimed to eliminate ambiguities and misunderstandings that may exist and encompasses the requirements for situational awareness system. For the user, the SRS will explain all functions that the software should perform. For the developer, it will be a reference point during software design, implementation and maintenance.

3.1.2 Intended Audience and Reading Suggestions

3.1.3 Intended audience

The intended audiences for the SAS include the project supervisor, base 19 FYP group (developers), UG project evaluation team, and other personnel in MCS CSE Department.

3.1.4 Project Supervisor

It will help the supervisor to supervise the project and guide the team in a better way. This document will be used to check whether all the requirements have been understood and in the end, whether the requirements have been properly implemented or not.

3.1.5 Testers:

The testers of the system can check user requirements from this SRS and develop test scenarios accordingly.

3.1.6 Documentation writers:

The document can serve as a future reference for other versions of the SRS.

3.1.7 Project Testers:

Project testers can use this document as a base for making test cases.

3.1.8 UG Project Evaluation Team

It will help the evaluation team to evaluate the progress of FYP project. The document will provide the evaluators with the scope, requirements and details of the project to be built. It will also be used as basis for the evaluation of the implementation and final project.

3.1.9 Reading suggestions

It would be suggested to the clients to go through the requirement section thoroughly.

For the developers it is suggested that they read and understand the product scope, overall description and system features thoroughly.

Testers should go through the operating environment, constraints, and the non-functional requirements before developing the test scenarios for the system.

3.1.10 Product Scope

The document only covers the requirements specifications for the Situational Awareness System. All the external interfaces and the dependencies are also identified in this document.

For	Military purpose
What	Offline Data Collection Application
The	Situational Awareness System
Is	Software System
That	Provides a secure database for military sensitive locations

This application will be used to navigate around an unknown area using a map and adding nearby untagged locations. User will also be able to navigate between two stored locations. Additional information will be added with the tagged place to help users analyze the data and finding appropriate ways for contacting a location.

3.2 Overall Description

3.2.1 Product Perspective

Situational Awareness System is developed to help provide the user with data collection of military sensitive locations that become vulnerable when collected on a public server or connected to internet. Navigational support will also be provided. Application will be used offline. Offline mode will use saved locations

3.2.2 Product Functions

The main features of the Situational Awareness System are highlighted below:

1. Use of OSM tiles for mapping
2. Manual tagging using known coordinates
3. Location information details, such as contact number, address etc will be added with each tag
4. Route will be shown from source to the destination
5. Two different dashboards will exist; system administrator and the user
6. Tagged coordinates will be stored on server

3.2.3 User Classes and Characteristics

The software has two types of clients; system user and the system administrator. Both the types of users have different access level to the system and its data. Each one can perform functions assigned to their respective roles.

3.2.3.1 System User

Users will use the proprietary map imagery to navigate around an area. They will be able to search and use the tags already in the database in offline mode

3.2.3.2 System Administrator

The admin will require to login to the system. They will have the authority to modify (edit, add or delete) the database connected to the server. The admin will also be allowed to add OSM imageries of different locations.

3.2.4 Operating Environment

SAS application will be developed for the Android OS using Android Studio using languages used in the latter namely java and XML. Since a database will be implemented Android SQLite database will also be used.

The SAS application can run on any recently made smartphone (since all of them meet minimum requirements) or smartphones with the following specification:

- At least 512 MB ram
- 100 MB of free space
- Jellybean 4.1.2 or higher
- Working Internet Connection

3.2.5 Design and Implementation Constraints

- The mobile application is constrained by the system interface to the GPS navigation system within the mobile phone. Since there are multiple system and multiple GPS manufacturers, the interface will most likely not be the same for every one of them. In addition, there may be difference between what navigation features each of them provide.
- The Internet connection is also a constraint for the application. Since the application accesses data from the database over the Internet, it is crucial that there is an Internet connection for the application to function.
- Time period provided for the completion of the Final year project may not be sufficient.

- Members may have to familiarize themselves with the tools to be used and lack the pre-requisite knowledge needed for the SAS android application
- Work flow in the SAS application has to be smooth
- Processes implemented in the android application should be simple
- The Graphical User Interface (GUI) should be easy and friendly to use
- System dashboards for both the system administrator and the user should be straightforward and uncomplicated.

3.2.6 User Documentation

Final release will be accompanied with a user guide to inform users how to use Situational Awareness system. User documentation that would be delivered along with the final product

- User manual

3.2.7 Assumptions and Dependencies

- One assumption about the product is that it will always be used on mobile phones that have enough performance. If the phone does not have enough hardware resources available for the application, for example the users might have allocated them with other applications, there may be scenarios where the application does not work as intended or even at all.
- Another assumption is that the GPS components in all phones work in the same way. If the phones have different interfaces to the GPS, the application need to be specifically adjusted to each interface and that would mean the integration with the GPS would have different requirements than what is stated in this specification.

3.3 External Interface Requirements

3.3.1 User Interfaces

The System comprises of an android based application, which shall provide a graphical user interface for user friendly environment. The user will be able to navigate the map using tags. Information of the searched location and their coordinates - longitude and latitude – will also be displayed with each location tags.

The user interface for the android application of the System, shall be compatible to all android devices but for best user experience versions preferable are Jelly beans 4.1.2 and above

The system administrator will have more access/ views including the user's views. System administrator will have a login page and a profile page. They will also be able to add, delete or modify map imagery, the tags and the location information.

3.3.2 Hardware Interfaces

Mobile Phone: SAS android application requires a mobile phone that supports android operating system, GPS and GSM module

Server: The mobile device will be connected to the server through a communication media. The server will have two parts; application server and database server. SAS will send a query to the application server which will fetch the data from the database and send it back to the SAS application.

3.4 System Features

3.4.1 Search

3.4.1.1 *Description and Priority*

Given that the user has opened the mobile application, then the first page that is shown should be the map view along with the search bar. The user should be able to search for a restaurant, hospital, hotel, park etc

3.4.1.2 *Stimulus/Response Sequences*

Input: User will search the name of the location.

Output: All the related search results will be marked on the map

3.4.1.3 *Functional Requirements*

REQ-1: The searched location shall be marked on the map.

3.4.2 Navigation to destination

3.4.2.1 *Description and Priority*

A user should be able to select a pin on a map. When a selection is made, the location of the restaurant should be sent to the mobile phone's GPS-navigation program. The user should then be navigated to the destination. When the destination is reached, a user should be able to go back to the search bar on the mobile application.

3.4.2.2 *Stimulus/Response Sequences*

Input: Selecting the desired tagged location.

Output: Routes from the current location to the destination are highlighted on the map

3.4.2.3 *Functional Requirements*

REQ-1: The system shall provide navigation to the desired destination from point A to point B

3.4.3 Location Description

3.4.3.1 *Description and Priority*

A user should be able to long press on the marked place, which is included on all resulting tagged locations. It will direct the user to all the information regarding the location, which includes the location name, address, description and coordinates etc

3.4.3.2 *Stimulus/Response Sequences*

Input: Long press on the desired tagged location.

3.4.3.3 *Functional Requirements*

REQ-1: The system shall show additional information with each tagged location

3.4.4 Search bar for searching specific locations

3.4.4.1 *Description and Priority*

User will be able to search a location by typing in correct spellings of the name. However, if no match is found the user should be informed but kept on the search bar in order to get the possibility to conduct a new search right away.

3.4.4.2 *Stimulus/Response Sequences*

Input: Name of the tagged location

Output: location will be marked on the screen or else "No match found" text displayed

3.4.4.3 *Functional Requirements*

REQ-1: The system shall display the location being searched on the map view

REQ-2: The system shall show a message of "No match found" in case of invalid search

3.4.5 Current Location

3.4.5.1 Description and Priority

The user should be able to locate his own position at any point. The user can turn on the GPS of the device and find their current position.

3.4.5.2 Stimulus/Response Sequences

Input: Go the GPS and turn it on

Output: The current location is marked on the map in the form of a check point.

3.4.5.3 Functional Requirements

REQ-1: The system shall be capable of finding user's current location via GPS in offline.

3.4.6 Access the central database

3.4.6.1 Description and Priority

The admin should only be able to access the central database in online mode. They can sync the tags/locations saved in the local database using a secure connection

3.4.6.2 Stimulus/Response Sequences

Input: Selecting "sync with central database" button in drawer activity

Output: Sync complete will be notified

3.4.6.3 Functional Requirements

REQ-1: The system shall be able to access the central database in online mode for synchronization with the local database.

3.4.7 Administrator sign in

3.4.7.1 Description and Priority

Given that a user has registered, every time the administrator logs in by re-entering their user-name, password and email, they are given access to the system after the entered information is verified with the database.

3.4.7.2 Stimulus/Response Sequences

Input: Administrator will enter their required information.

Output: It will be indicated that the admin has successfully signed in

3.4.7.3 Functional Requirements

REQ-1: System Administrator shall log in every time, in order to make any changes in the application

3.4.8 Profile page

3.4.8.1 *Description and Priority*

On the mobile application, a system administrator should have a profile page. On the profile page the admin can edit his/her information, which includes the password, e-mail address and user name.

3.4.8.2 *Stimulus/Response Sequences*

Input: Go to the profile page

Output: Personal information is displayed which can be edited

3.4.8.3 *Functional Requirements*

REQ-1: The system shall allow the admin to make any changes to the information they have entered during registration.

3.4.9 Storing Tags

3.4.9.1 *Description and Priority*

The system can save some of the tagged locations in the local database of the device. When the size of the local database reduces, the system can upload the additional tags in the database server through the internet and access them when needed.

3.4.9.2 *Stimulus/Response Sequences*

Input: Storing the tagged locations

Output: The tagged locations are shown in the database with the coordinates and other information.

3.4.10 Functional Requirements

REQ-1: The system shall maintain database for all the tagged locations locally on the device as well as on the server.

3.4.11 Modifying tags/locations

3.4.11.1 Description and Priority

Long press on a location will open a location description screen, given the administrator is logged in they will have the edit and delete location option available. Changes will be updated in the database.

3.4.11.2 Stimulus/Response Sequences

3.4.11.2.1 Edit

Input: Editing any location

Output: Text Fields become editable, save button enabled, which saves the changes made in the database.

3.4.11.2.2 Delete

Input: Deleting any location

Output: Location marker is removed from map view and from the database

3.4.11.3 *Functional Requirements*

REQ-1: The System shall allow the system admin to add, delete and edit the already tagged locations

3.4.12 Adding tags/locations

3.4.12.1 *Description and Priority*

Long press on any position on the map view or clicking on the add current location option in drawer activity, will open a location description screen, given the administrator is logged in, they will have to add the required details and click save. Location will be added in the database.

3.4.12.2 *Stimulus/Response Sequences*

Input: adding any location

Output: Location marker is shown on the map view and added in the database

3.4.12.3 *Functional Requirements*

REQ-1: The System shall allow the system admin to add, delete and edit the already tagged locations

3.5 Other Nonfunctional Requirements

3.5.1 Performance Requirements

- **Response Time:**

Response time of Server on request from Android App must not exceed 30 sec.

Android App should not take more than 5sec in getting User location and send it to Server.

- **Throughput**

Server must process every request within 5 seconds.

- **Reliability**

Failure rate of application/transaction must not exceed 1 in 100.

MTBF (Mean Time Between Failure) for server should be 1 week at minimum within 6 month of deployment. However, after 6 months of deployment MTBF must be 1 month.

MTRS (Mean Time to Restore Service) must be 2 hours max for critical, 1 hour for major and 30 minutes for minor faults.

3.5.2 Safety Requirements

- **Backup & Recovery**

Backups of database shall be maintained automatically every day at least once. In case of server failure or database or application gets corrupted, it should not take more than 1 hour to install the application again and load the backup database.

3.5.3 Security Requirements

- **Data Transfer**

The system shall use secure sockets in all transactions that include any confidential user information.

3.5.4 Software Quality Attributes

3.5.6 The application shall be scalable for maximum possible Users.

3.5.7 Data of locations older than 5 years shall be discarded for better system performance.

3.5.8 System shall be flexible enough to cater future modifications such as prediction of jam, etc.

3.5.9 System shall be modifiable if more features need to be added.

3.6 Business Rules

This product is being developed for all users

Developers have right to keep the modules in the product for later use.

CHAPTER:4
DESIGN AND DEVELOPMENT

4 DESIGN AND DEVELOPMENT

4.1 INTRODUCTION:

This design document captures all our functional requirements and shows how they interact with each other conceptually. The low level design also shows as to how we have been implementing how we are going to implement all of these requirements. This low level design for the time being does not address any non-functional requirements that our system has and that have been mentioned in our SRS.

4.1.1 Purpose of the document:

The purpose of this Software Design Document is to provide a description of the design of our system fully enough to allow for software development to proceed with an understanding of what is to be built and how it is expected to be built. This Software Design Document provides information necessary to get a description of the details for the software and the system to be built. The purpose of this document is to present a design view and detailed description of the Situational Awareness System. It will explain the purpose and features of the system, the interfaces of the system, what the system will do, its entire processes in detail, the constraints under which it must operate and how the system will react to inputs and what will be its outputs. This document is intended for both the stakeholders and the developers of the system.

4.1.2 Scope of the Development Project

The document only covers the design specifications for SAS. The SAS will assist in manually tagging key locations or by using geographical coordinates of the current location. Additional information will be added with the tagged place, for example phone number, address, etc, to help

them analyze the data and to find appropriate ways for contacting the location. Local database will be synchronized with the central database via a secure channel.

4.1.3 Definitions, acronyms, abbreviations

Acronym	Complete
App	Application
GIS	Geographical Information System
GPS	Global Positioning System
UML	Unified Modeling Language
SAS	Situational Awareness System

4.1.4 Overview of the document

This document shows the design and working of Situational Awareness System. It starts from higher level details for a non-technical reader to understand just by seeing the diagrams to the lower level details that aid the developer to code and understand other technical details of the application.

Section 4.2 The System Architecture Description gives a detailed overview of the System.

Section 4.2.1 Overview of Modules/Components shows the main component of the System and their inter-relationships.

Section 4.3 Structure and Relationships shows the higher level details system working by the means of System Block Diagram.

Section 4.4 Architectural Style shows Database Design, Entity Relationship Diagram. Section 4.5 shows UML Diagram Activity, State Transition, and Use Case diagrams, Sequence diagrams and Structure Chart.

Section 4.6 describes how the application is designed to curb the tendency of User Interface Issues and problems during User Interaction.

Section 4.7 shows the Reuse and Relationship to other Products i.e.; information about work done in the same project before and any reuse of the same work. The section also provides a key to reuse this system for further upgrades.

Section 4.8 Design Decisions and Tradeoffs shows the architecture style and design pattern of the application.

4.2 SYSTEM ARCHITECTURE DESCRIPTION

In this section, the overall architecture of the system is discussed, including the introduction of various components and subsystems. It is mainly supported by system Architecture diagram which shows an insider's perspective of the system by describing the high level software components that perform the major functions to make the system operational.

4.2.1 OVERVIEW OF MODULES/COMPONENTS

The system architecture will mainly be in five fundamental modules "User", "Admin", "Mobile Map/SAS/local (Core System)", "Database server" and the "GPS Module". Block diagram provides an overview of the system. Admin accessing the system through the mobile hardware will access the Database using a secure channel to synchronize with the central database. The sub-modules of the Block diagram are further elaborated as below.

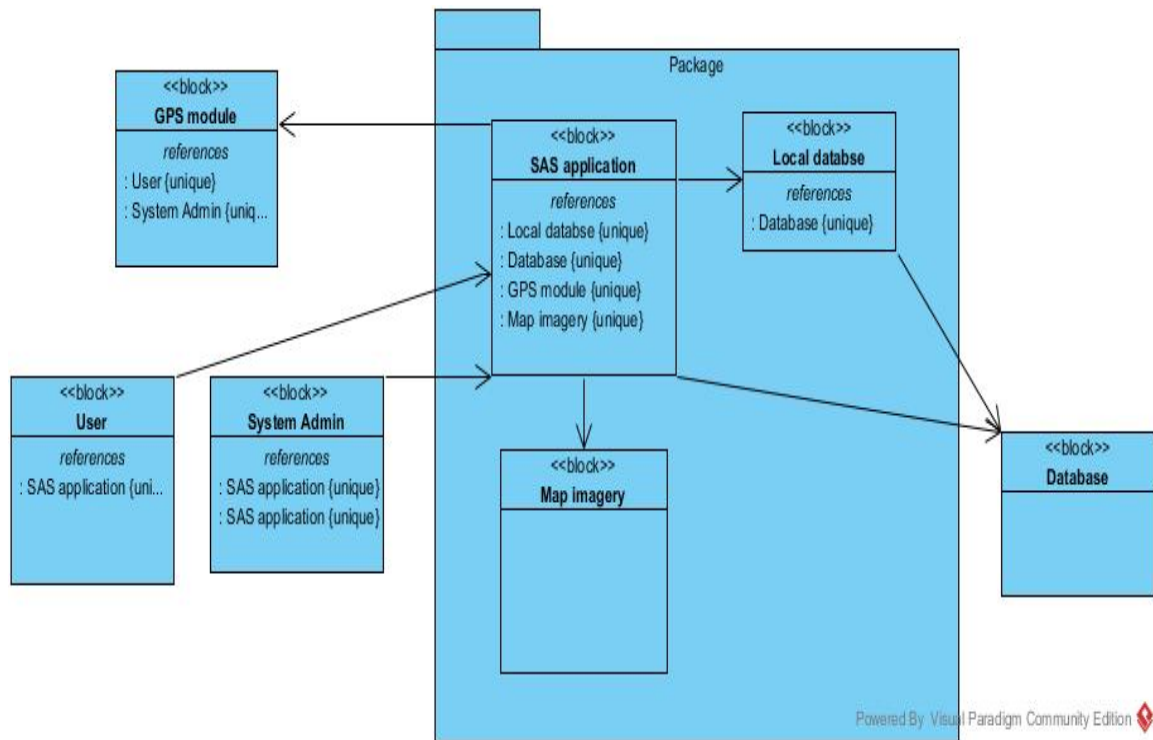


Figure 4-1 Component Overview for SAS

GPS device

SAS android application requires a mobile phone that supports android operating system, GPS and GSM module. The user and the system administrator will use the mobile device to access the application. The inbuilt GPS application will be used to find the user's current location that will be used to find routes from the current location and store new locations.

User

User uses the SAS system to access the Mobile app and then chooses functionality according to their requirement. Users consist of anyone who accesses the application as a guest. User interacts with the mobile app that further accesses the services provided by the app.

System Administrator

System Administrator interacts with the android application and the database for managing admin accounts and by dealing with the whole functionality of the system by handling all the provided services by SAS.

Local Database server

A database stores all the data about the mobile application, admin records, locations and all the related processing also happens there.

SAS application

Mobile app is the platform to access the system and having different functionalities like editing profile, searching locations, accessing location information and finding route between two defined destinations.

Database

Locations and their respective attributes will be synchronized with the local database when a secure connection is available.

Map Imagery

Information regarding Map imagery will be stored here. Any changes made will be updated here.

4.3 STRUCTURE AND RELATIONSHIPS

This section ponders upon the interrelationships and dependencies among various components. It is mainly described by a diagram which is further augmented by explanatory text. UML Class diagram also helps us understanding the system structure.

4.3.1 System Block Diagram

The mobile application will need to communicate to a GPS application within the mobile phone, which in turn communicates with a physical GPS device to find the location of the user. The GPS will provide the mobile application with locations of both the user, the destination and the distance between them, but it will also provide maps and the functionality to display the application's data on the map. The functionality provided by the GPS will be embedded into the application in order for the user to be able to use the functions in the application in a seamlessly

manner. Since this is a data-centric product, it will need somewhere to store the data. For that, a database server will be used

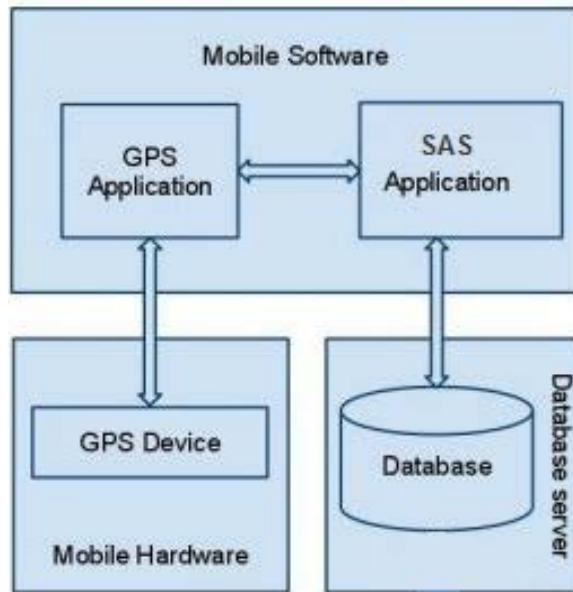


Figure 4-2 Block Diagram for SAS

4.4 Architectural Style:

4.4.1 System Architecture:

Layered Architecture **3-Tiers** will be used to implement SAS. This will be composed of multiple processes, each communicating with the others by passing messages. Conceptually, the services can be seen as components of the overall solution. However, internally, each service is made up of software components, just like any other application, and these components can be logically grouped into presentation, logic, and data layers. Other applications can make use of the services without being aware of the way they are implemented.

4.4.2 Architecture Diagram:

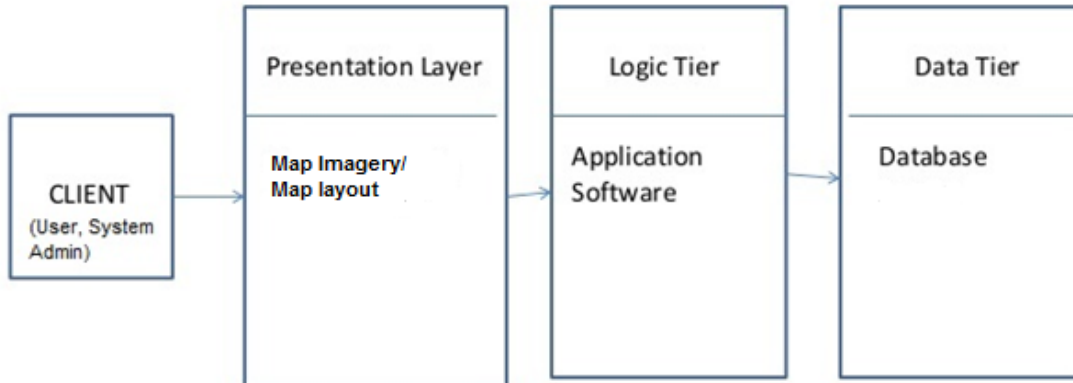


Figure 4-3 Architecture Diagram for SAS

Presentation Layer

It provides a platform for the interaction of the user and the system administrator with the system. It displays data to the user and accepts input from the user. The Presentation layer can only receive requests from, and return responses to the logic layer. It cannot have direct access to Data Access layer.

Logic Layer

When an application must provide services to other applications, as well as implementing features to support clients directly, a common approach is to use a service layer that exposes the business functionality of the application. The services layer effectively provides an alternative view that allows users to use a different channel to access the application.

Data Access Layer

This layer receives request from the Service Layer and sends back data after querying it from the database server.

4.5 UML Diagrams:

4.5.1 User view (use case diagram)

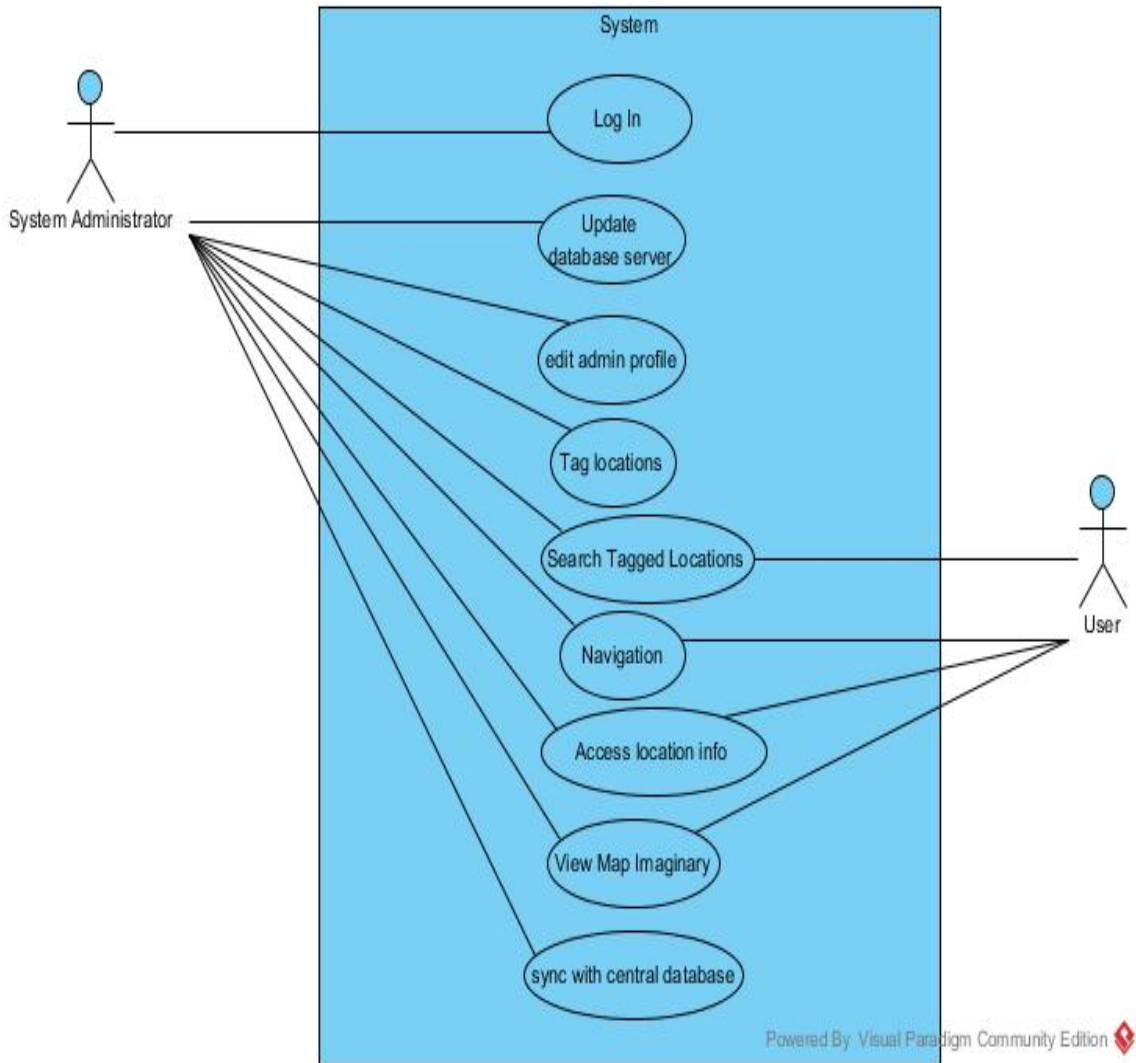


Figure 4-4 Use case Diagram for the System

4.5.2 Use Case Specification:

Actors

System administrator, User

Use Cases

1. Log In
2. Update database server
3. Edit admin profile
4. Tag locations
5. Search tag locations
6. Navigation
7. Access location information
8. View map imagery
9. Synchronization with central database

4.5.3 Use Case Description

Login

Use Case ID:	1		
Use Case Name:	Login		
Actors:	Administrator		
Created By:	Zainab	Last Updated By:	Zainab
Date Created:	8/1/2017	Date Last Updated:	8/1/2017
Description:	The administrator logs in to the system to manage tag locations and handle the users		
Preconditions:	The administrator has to start the app and open the login page first.		
Post conditions:	If the use case was successful, the administrator is now logged into the system. If not the system state remains unchanged.		
Normal Flow (primary scenario):	<p>This use case starts when an administrator wishes to log into the System.</p> <p>The system requests that the administrator enter his/her name and password.</p> <p>The administrator enters his/her name and password.</p>		

	The system verifies the entered name and password from the database and logs the administrator into the system.
Alternative Flows:	If the administrator actor enters an invalid name and/or password, the system displays an error message. The administrator remains on the login page.

Sync with central database

Use Case ID:	2		
Use Case Name:	Sync with central database		
Actors:	Administrator		
Created By:	Zainab	Last Updated By:	Zainab
Date Created:	8/1/2017	Date Last Updated:	8/1/2017
Description:	Admin will be able to synchronize the updated local db with the central database through a secure channel		
Preconditions:	The administrator must be logged in to the system in order to synchronize the local database with central database		
Post conditions:	The updates in local database are saved in the central database		
Normal Flow (primary scenario):	<p>The administrator logs in to the system</p> <p>The administrator clicks the sync option</p> <p>The administrator will be notified if the communication channel is secure</p> <p>The changes are saved in the central database</p>		
Alternative Flows:	Communication channel is not secure, an error message is generated		

Update local database

Use Case ID:	3
Use Case Name:	Update database server
Actors:	Administrator

Created By:	Zainab	Last Updated By:	Zainab
Date Created:	8/1/2017	Date Last Updated:	8/1/2017
Description:	The administrator updates the database server which contains the information of the users, the tagged locations and other tables		
Preconditions:	The administrator logs in to the system and opens the database		
Post conditions:	The information in the database is updated and remains till the next time the administrator access the database to make any changes		
Normal Flow (primary scenario):	<p>The administrator logs in to the system</p> <p>The administrator open the database server</p> <p>The administrator adds, delete , edit the tables in the database</p> <p>The changes are saved in the database</p>		
Alternative Flows:	<p>The administrator updates the information contained in the database but is not able to save it. The working of the database server should be checked in this case</p> <p>The administrator is unable to open the database server due to some network error</p>		

Edit admin profile

Use Case ID:	4		
Use Case Name:	Edit admin profile		
Actors:	Administrator		
Created By:	Zainab	Last Updated By:	Zainab
Date Created:	8/1/2017	Date Last Updated:	8/1/2017
Description:	The administrator tries to view and edit his profile.		
Preconditions:	The administrator has to open the login page first.		
Post conditions:	The System must show the profile and record the changes made therein.		
Normal Flow (primary scenario):	<p>The administrator login credentials and signs in to the system.</p> <p>The administrator clicks profile options.</p> <p>The system enables the administrator to view or modify their profile</p>		

	information. The system records the changes in the database made in the profile.
Alternative Flows:	The system gives error message, mentioning that administrator cannot change the profile information.

Tag locations

Use Case ID:	5		
Use Case Name:	Tag locations		
Actors:	Administrator		
Created By:	Zainab	Last Updated By:	Zainab
Date Created:	8/1/2017	Date Last Updated:	8/1/2017
Description:	The administrator edits, deletes and adds the locations in the database		
Preconditions:	The administrator has to open maps after logging in.		
Post conditions:	The locations have been tagged in the map imagery The locations are shown in the imagery after being tagged The tagged locations are saved in the database		
Normal Flow (primary scenario):	The administrator log ins to the system after entering the password and user name The administrator clicks or opens the map imagery The administrator tags the locations and edit, or delete the tagged locations The changes made therein are saved in the database		
Alternative Flows:	The gps is not active and the administrator is unable to tag the locations The locations are tagged but not saved in the database		

Search tagged locations

Use Case ID:	6
--------------	---

Use Case Name:	Search tagged locations		
Actors:	Administrator, user		
Created By:	Zainab	Last Updated By:	Zainab
Date Created:	8/1/2017	Date Last Updated:	8/1/2017
Description:	The administrator and user searches the locations tagged by the administrator in the map imagery.		
Preconditions:	The locations must be already tagged in the map imagery The actor must have access to the map imagery		
Post conditions:	The information of the location search by the actor is displayed after searching it. The actor can save the tagged locations		
Normal Flow (primary scenario):	The actor logs in to the system The actor opens the map imagery The actor goes in the search bar and types the required location The required location is shown in the imagery		
Alternative Flows:	The actor searches the tagged locations and they are not stored in the db or the gps is not active		

Navigation

Use Case ID:	7		
Use Case Name:	Navigation		
Actors:	Administrator, user		
Created By:	Zainab	Last Updated By:	Zainab
Date Created:	8/1/2017	Date Last Updated:	8/1/2017
Description:	The administrator or user enters the required location		
Preconditions:	The device gps should be turned on The locations must be added in the database		
Post conditions:	A path is displayed on the map imagery		

Normal Flow (primary scenario):	<p>The actor logs in to the system</p> <p>The actor opens the map imagery and locates the current location or enters a starting point</p> <p>The actor then enters the destination in the bar displayed</p> <p>The map imagery displays a route from the starting location to the destination.</p>
Alternative Flows:	<p>The device Gps is not turned on</p> <p>One or both of the entered locations are not tagged in the database by the administrator</p> <p>The system is unable to fetch the destination tagged location from the database due to some error</p>

Access location information

Use Case ID:	8		
Use Case Name:	Access location information		
Actors:	Administrator, user		
Created By:	Zainab	Last Updated By:	Zainab
Date Created:	8/1/2017	Date Last Updated:	8/1/2017
Description:	The administrator or user access the information of the location that is tagged in the db		
Preconditions:	<p>The locations must be tagged in the db</p> <p>The information related to the location must be saved in the database for access</p>		
Post conditions:	<p>The information related to the location is shown with the tagged location</p> <p>The administrator can make changes in the information and update the database</p>		
Normal Flow (primary scenario):	<p>The actor logs in to the system</p> <p>The actor searches the required location and the tag is displayed with the location</p>		

Alternative Flows:	The actor searches for the location information but the information is not displayed due to error in the database system.
--------------------	---

View map imagery

Use Case ID:	9		
Use Case Name:	View map imagery		
Actors:	Administrator, user		
Created By:	Zainab	Last Updated By:	Zainab
Date Created:	8/1/2017	Date Last Updated:	8/1/2017
Description:	The administrator or user views the map imagery		
Preconditions:	The map imagery should be saved in the external file The actor must be using the system		
Post conditions:	The map imagery is shown to the actor		
Normal Flow (primary scenario):	The actor gains access to the system The map imagery is displayed		
Alternative Flows:	The map imagery is not displayed to the actor if the system stops working or the imagery is not saved in the database		

4.5.4 Sequence diagram

Sequence Diagrams of key use cases are mentioned below:

4.5.5 Login success/failure (System administrator)

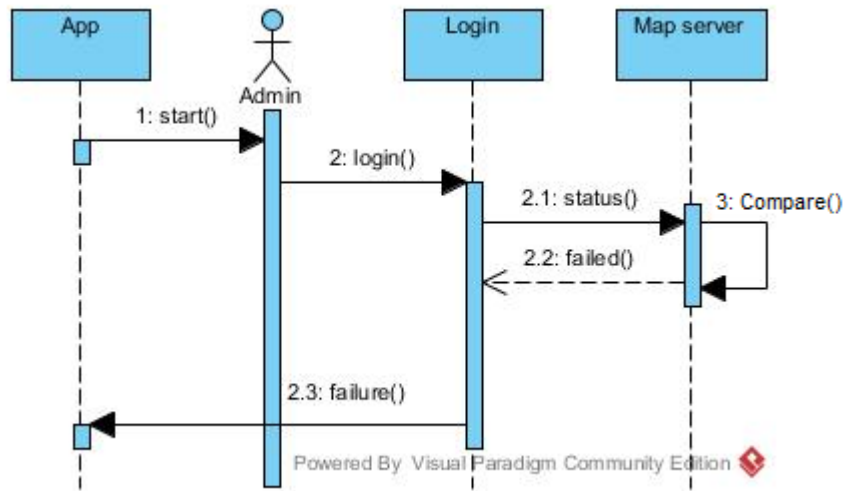
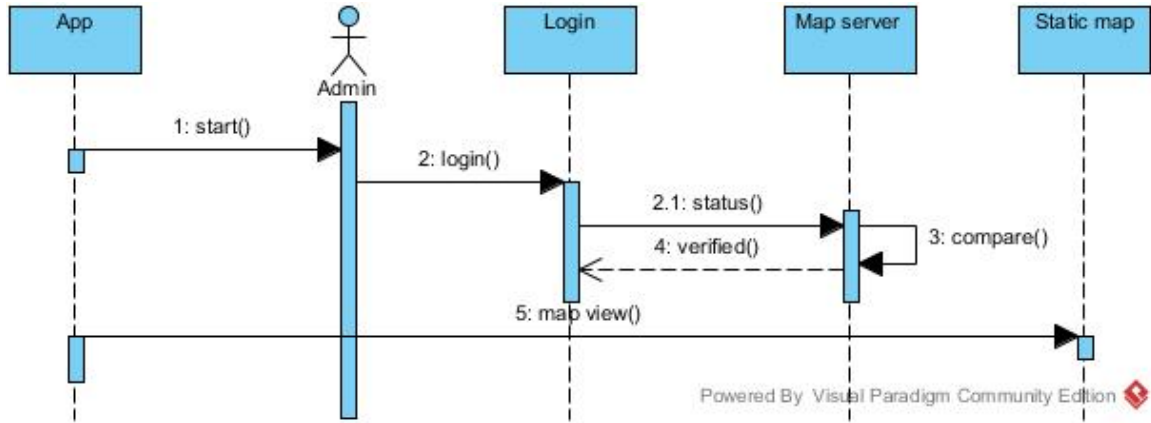


Figure 4-5 Sequence Diagram for Login success/failure

4.5.6 Add location via GPS

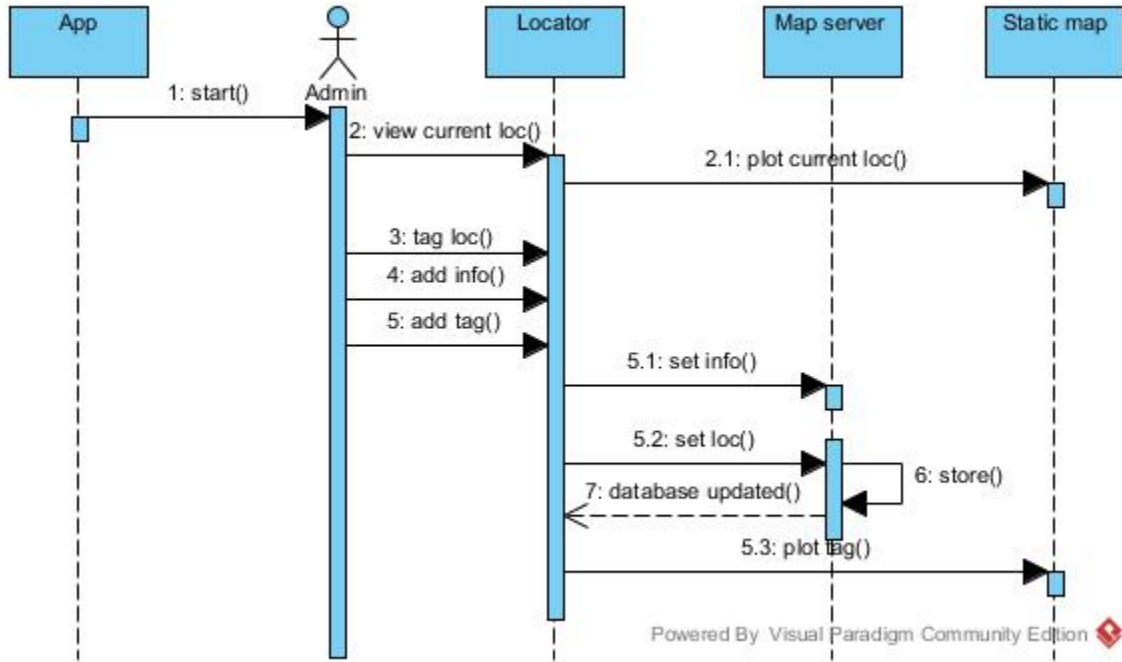
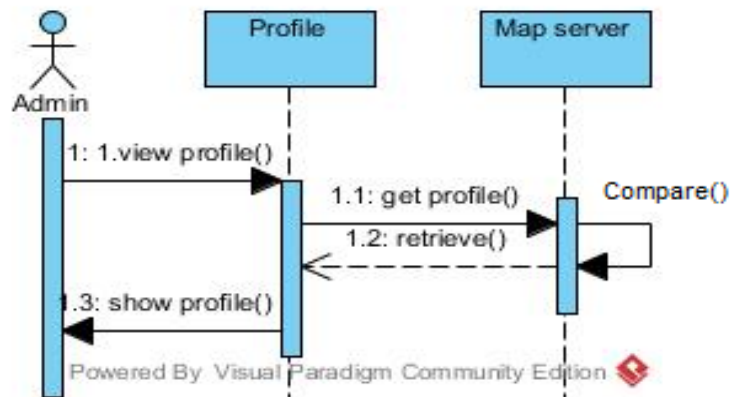


Figure 4-6 Sequence Diagram for add location via GPS

4.5.7 View and editing admin profile



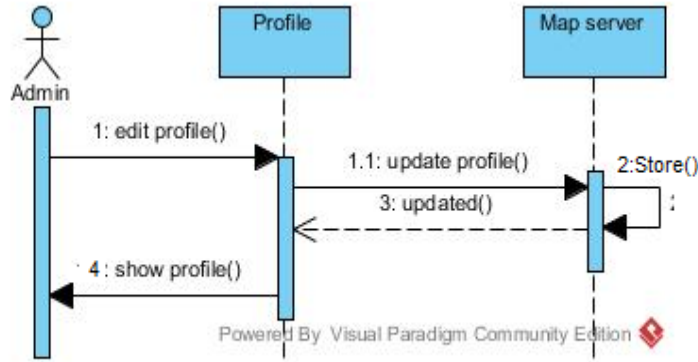


Figure 4-7 Sequence Diagram for view and edit admin profile

4.5.8 Search tags

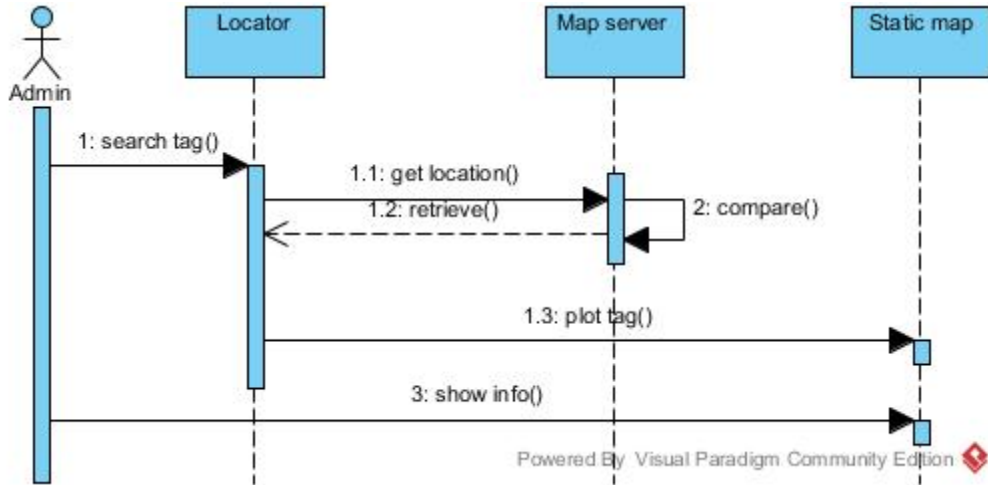


Figure 4-8 Sequence Diagram for search tags

4.5.9 Edit tags

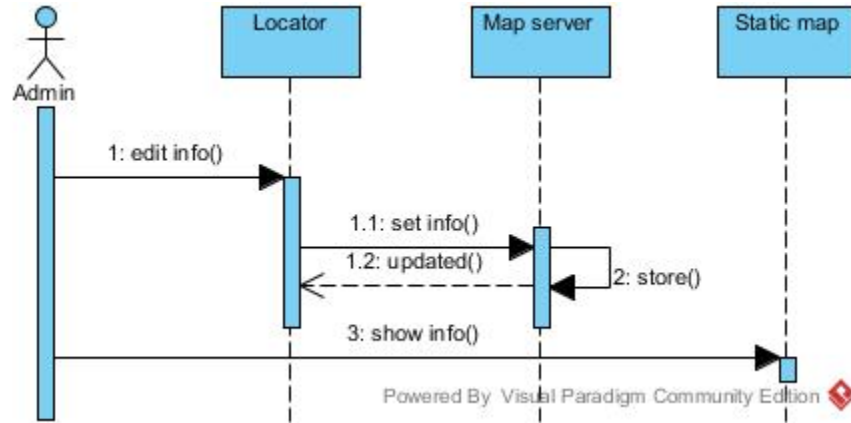


Figure 4-9 Sequence Diagram for edit tags

4.5.10 Delete tags

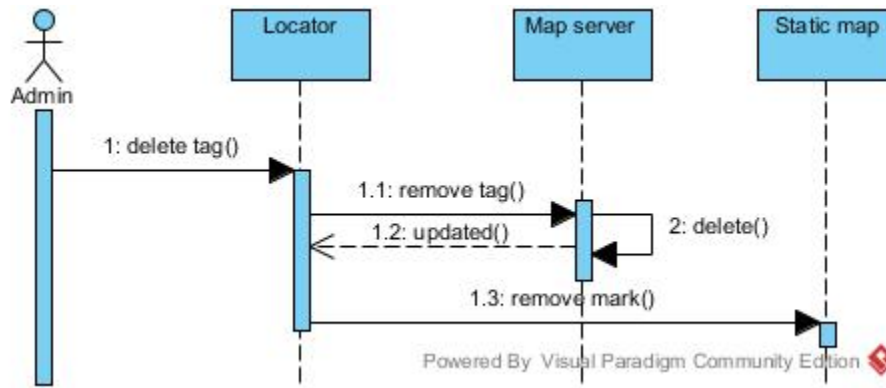


Figure 4-10 Sequence Diagram for delete tags

4.5.11 Route between current location and tagged location

The diagram below defines the sequence of actions that happens when the admin searches for a route between the current location and a searched tagged location

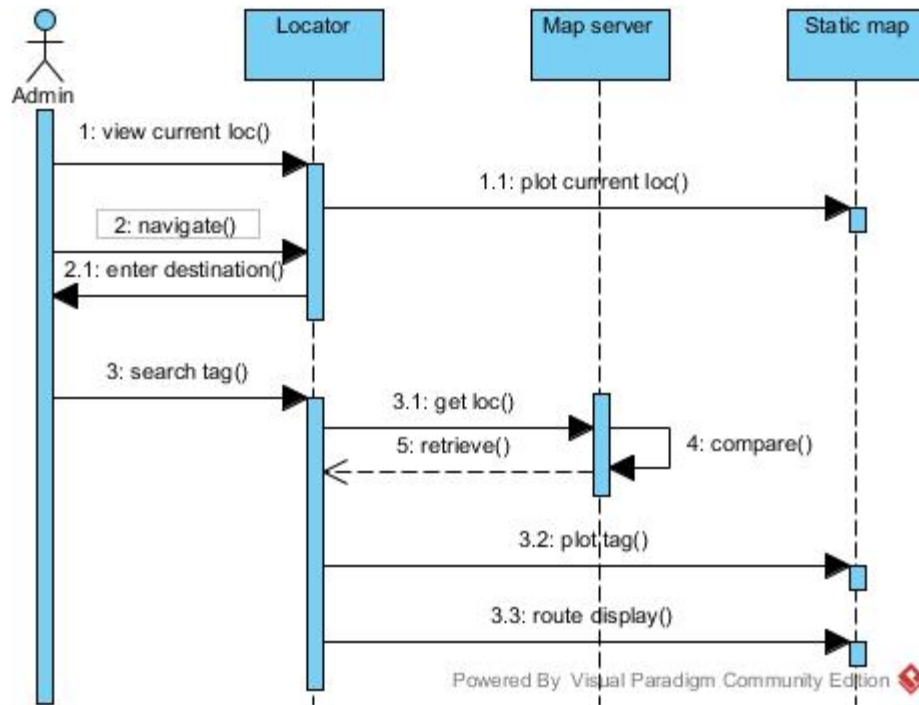


Figure 4-11 Sequence Diagram for route between current and tagged location

4.5.12 Route between two searched tagged location

The diagram below defines the sequence of actions that happens when the admin searches for a route between two searched tagged location

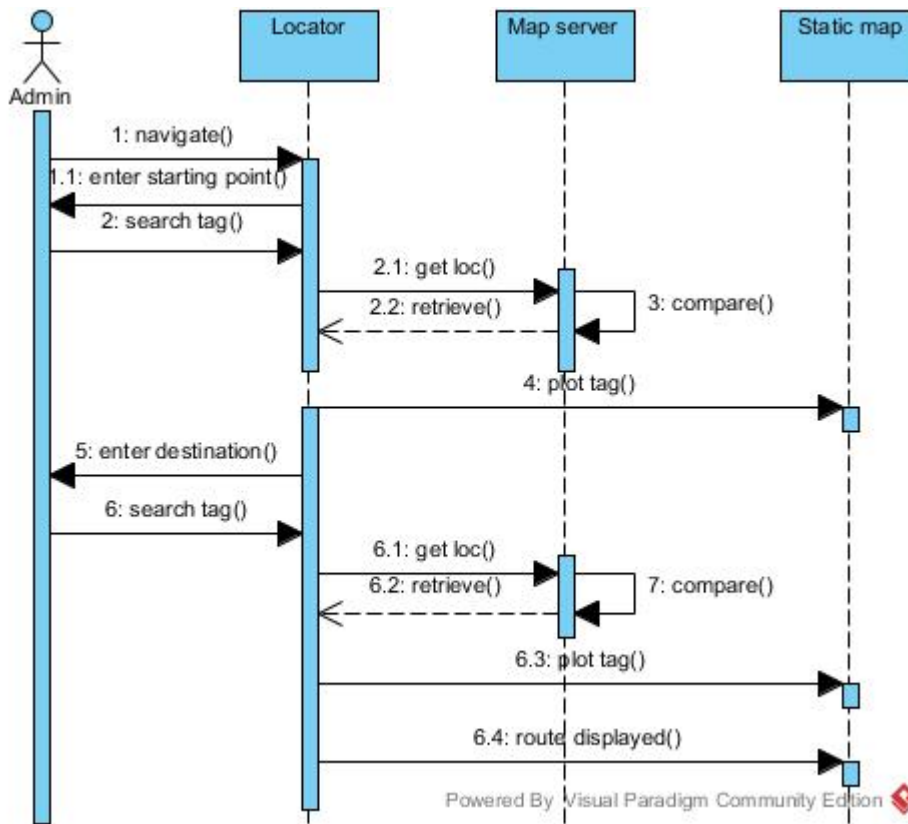


Figure 4-12 Sequence Diagram for route between two searched tagged location

4.5.13 Successful Synchronization

The diagram below defines the sequence of actions that happens when the admin synchronizes the local database with the central database for secure connection.

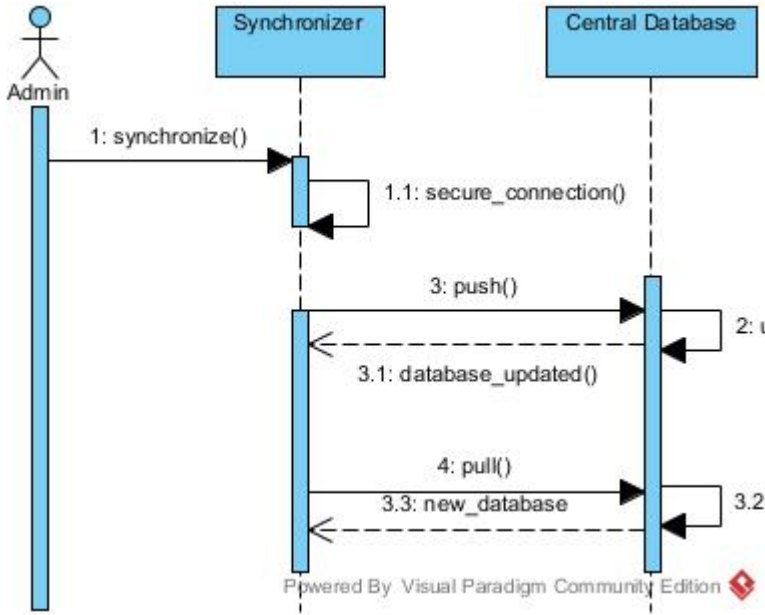


Figure 4-13 Sequence Diagram for successful synchronization

4.5.14 Unsuccessful Synchronization

The diagram below defines the sequence of actions that happens when the admin fails to synchronize the local database with the central database.

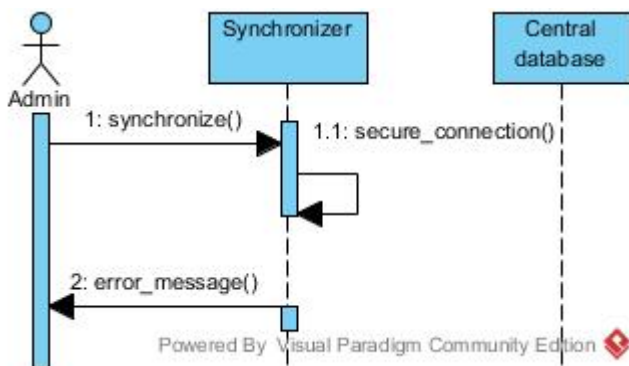


Figure 4-14 Sequence Diagram for unsuccessful synchronization

4.5.15 Search location (User)

The diagram below defines the sequence of actions that happens when the user searches for a tagged location.

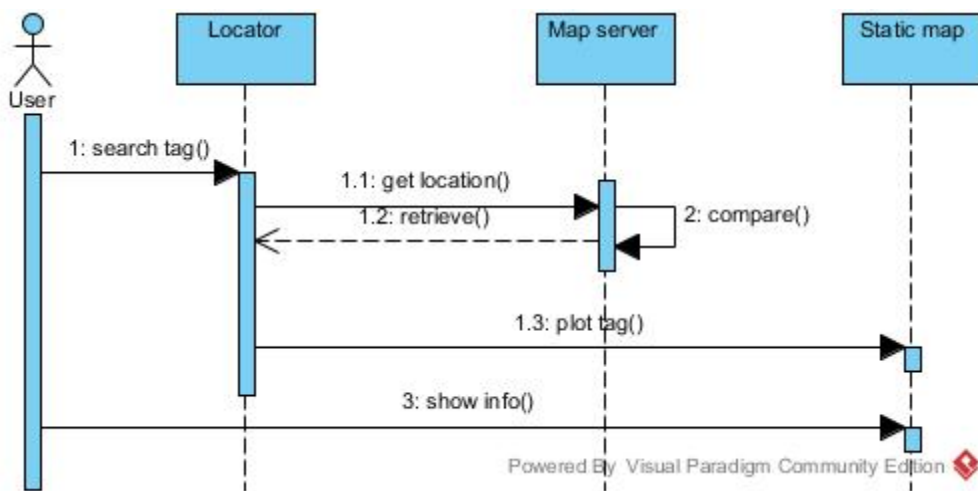


Figure 4-15 Sequence Diagram for search location(user)

4.5.16 Route between current and tagged location(User)

The diagram below defines the sequence of actions that happens when the user wants to find the route between current and tagged location.

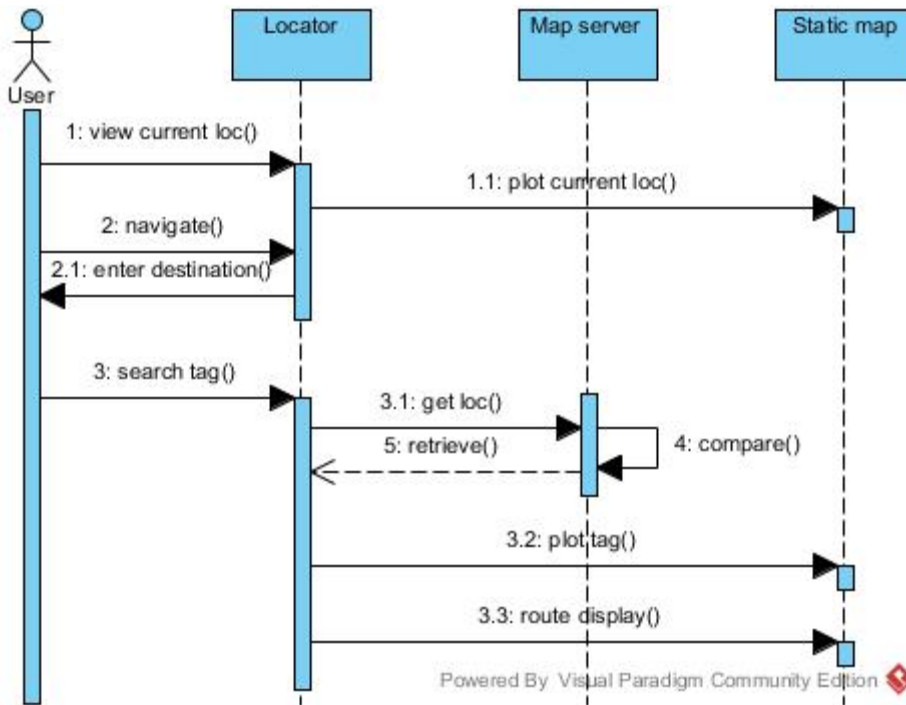


Figure 4-16 Sequence Diagram for route between current and tagged location(user)

4.5.17 Route between two tagged locations(User)

The diagram below defines the sequence of actions that happens when the user wants to find the route between two tagged locations.

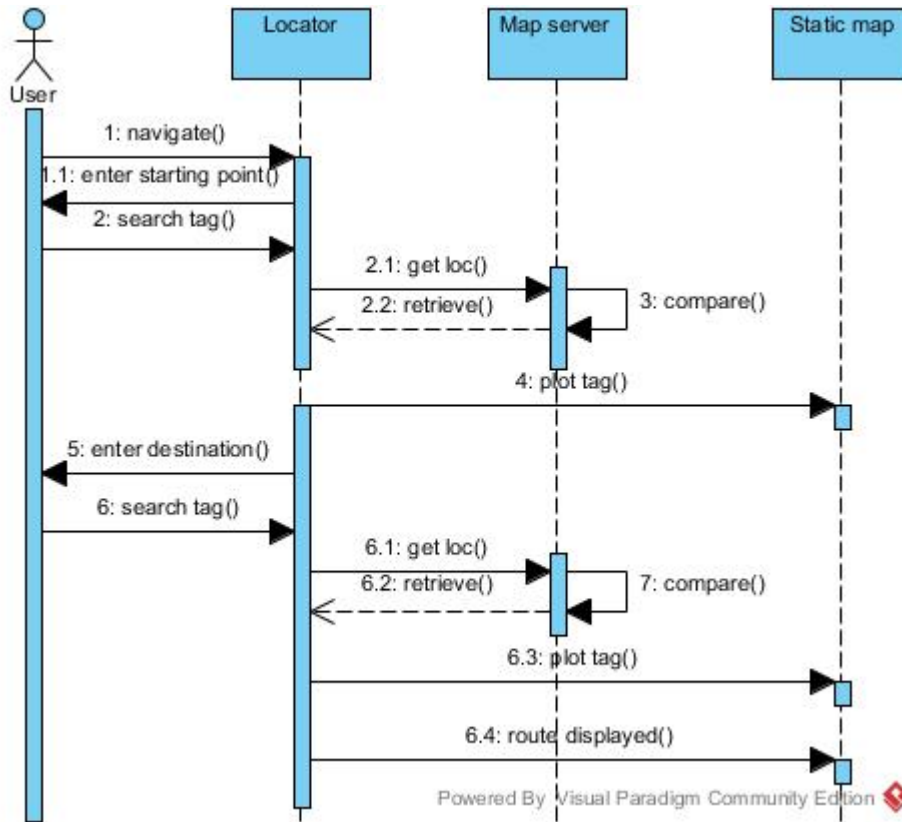


Figure 4-17 Sequence Diagram for route between two tagged locations(user)

4.5.11 Logical View:

4.5.12 State Transition Diagram

The State Transitions occurring in the application are shown in fig. below:

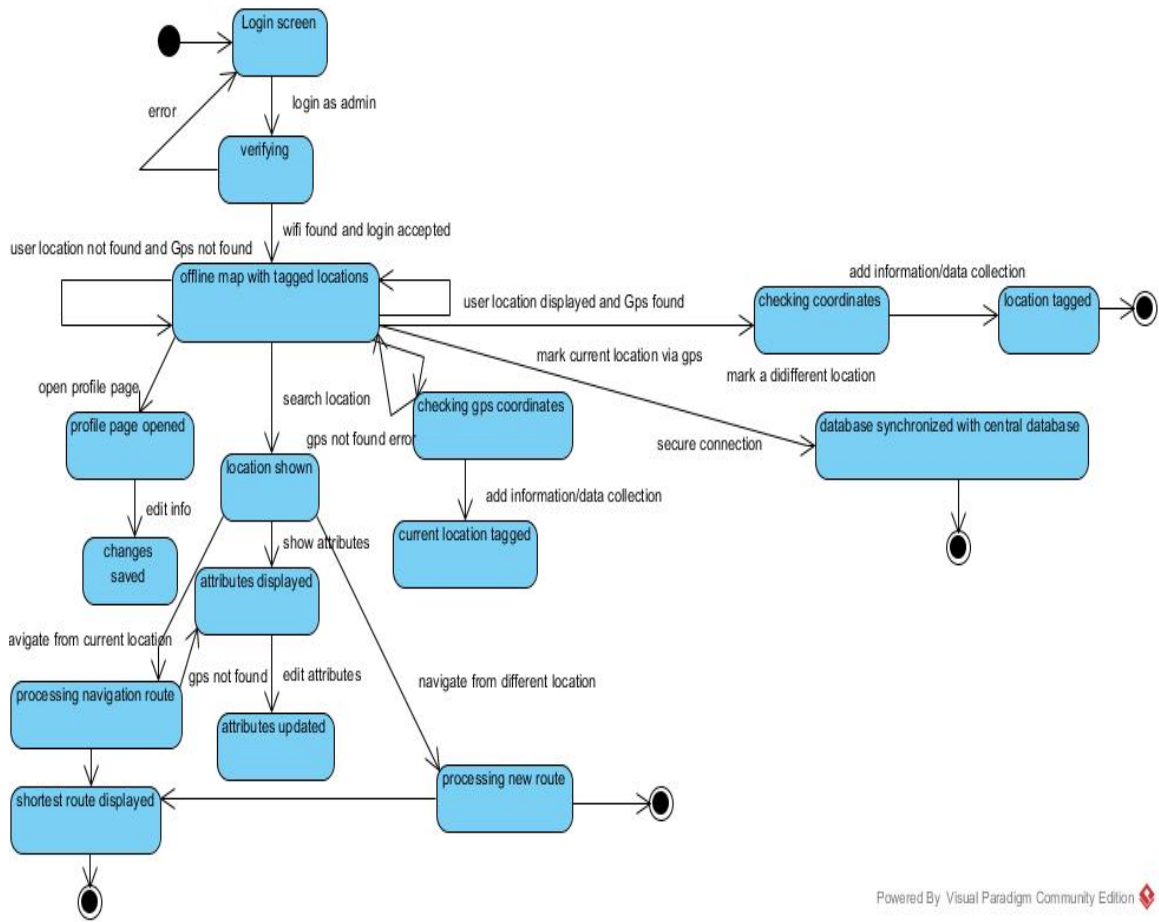
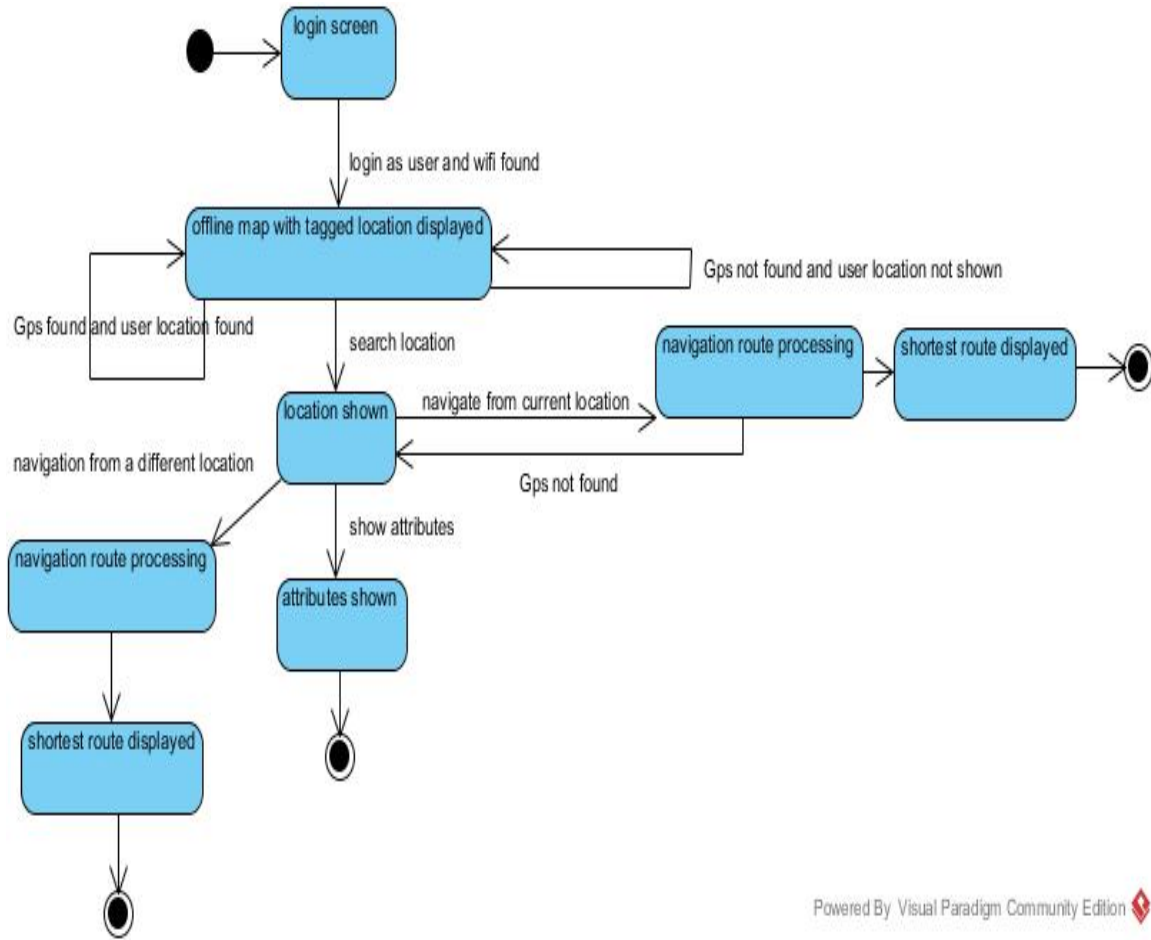


Figure 4-18 State transition diagram for User



Powered By Visual Paradigm Community Edition

Figure 4-19 State Transition Diagram for System Administrator

4.5.13 Implementation View

4.5.14 Class Diagram

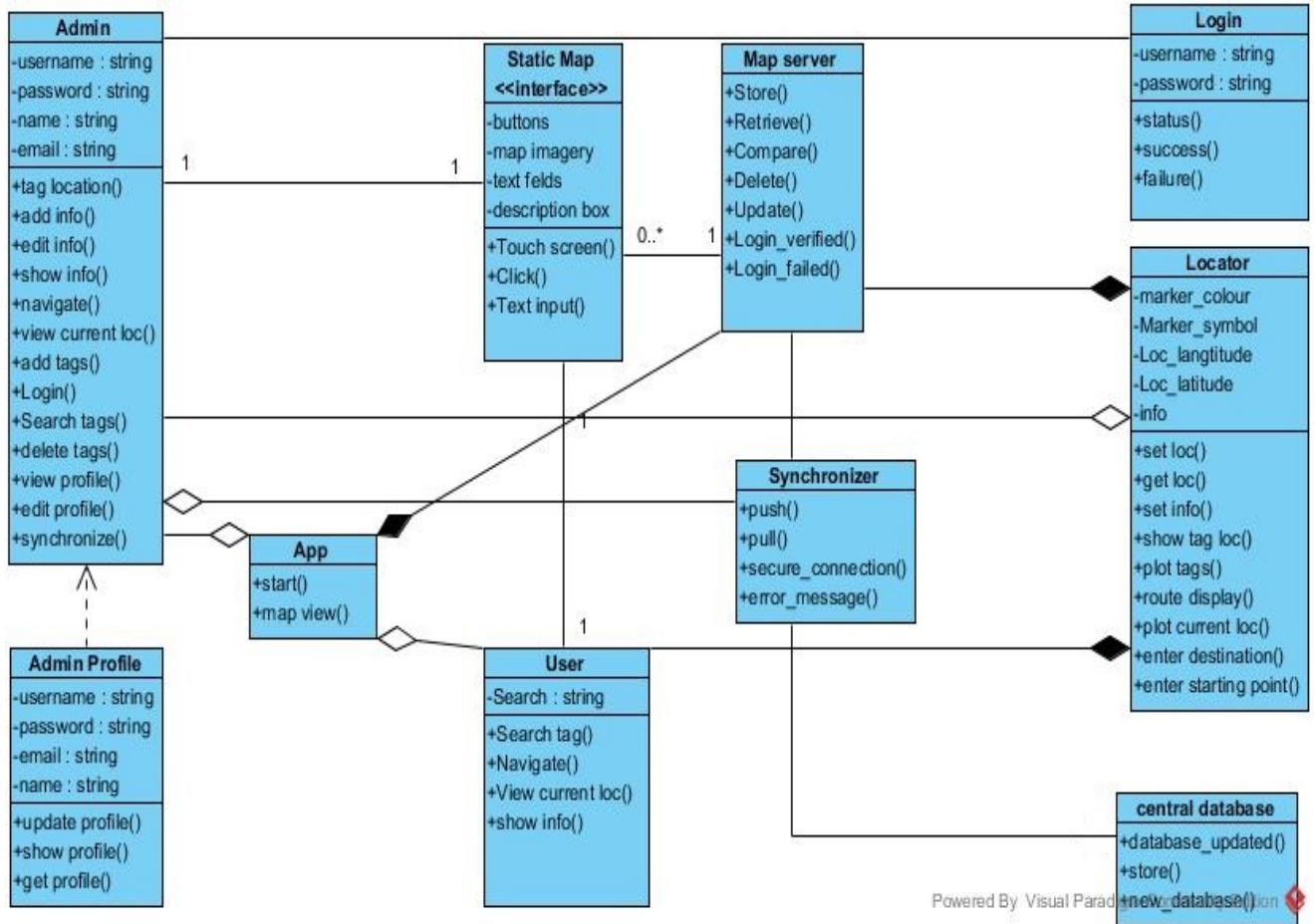


Figure 4-20 Class Diagram for SAS

4.5.15 System Classes Description:

Table 4-1 System Classes Description

Class Name	Description
App	App class contains all the information that SAS has to perform. It is the main class, which will be acting as a gateway to all the other classes.
User	User class contains all the activities a user that is signed in as a guest, can perform.
System admin	Admin class contains all the activities a user that is signed in as a guest, can perform.
Admin Profile	The class handles all the functions that an administrator can perform and maintains the rights of an administrator. An admin also has to login into the system to get all-access privileges.
Static Map (Interface)	It contains all the information to enable user to interact with SAS. It has links to all the functions of different classes that on selection lead to different actions.
Map server	This class contains the database objects and server information. It is the local database that will connect the application to the central database for synchronization via a secure channel. It will contain to all the database objects and save the data in the local database.
Login	This class controls the access to the SAS for the admin. It handles the login system for SAS.
Locator	This class handles the location mapping system of SAS. It is a major class and handles most of the functions. It plots all the tags, shows routes between destinations and helps admin to add new tags etc
Synchronizer	This class handles the synchronization of the local database to the central database. It checks for a secure connection when called. If the connection is secure and databases are synchronized or else an error message is generated.
Central database	It represents a backup database which will contain elements from the local database and will be synched using push and pull methods from synchronizer class.

4.5.16 Dynamic View:

4.5.17 Activity Diagram

System Administrator

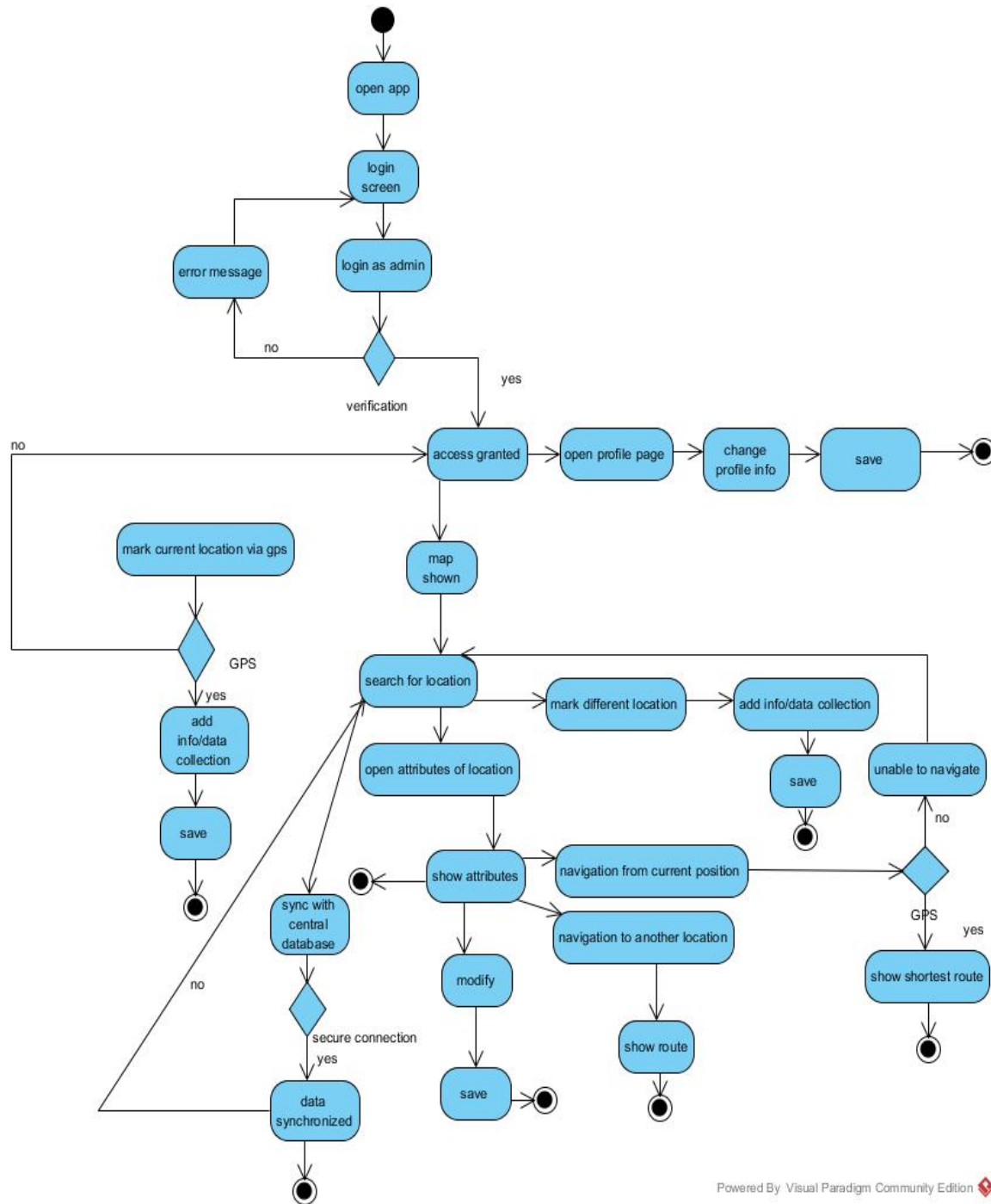


Figure 4-21 Activity Diagram for System Administrator

User

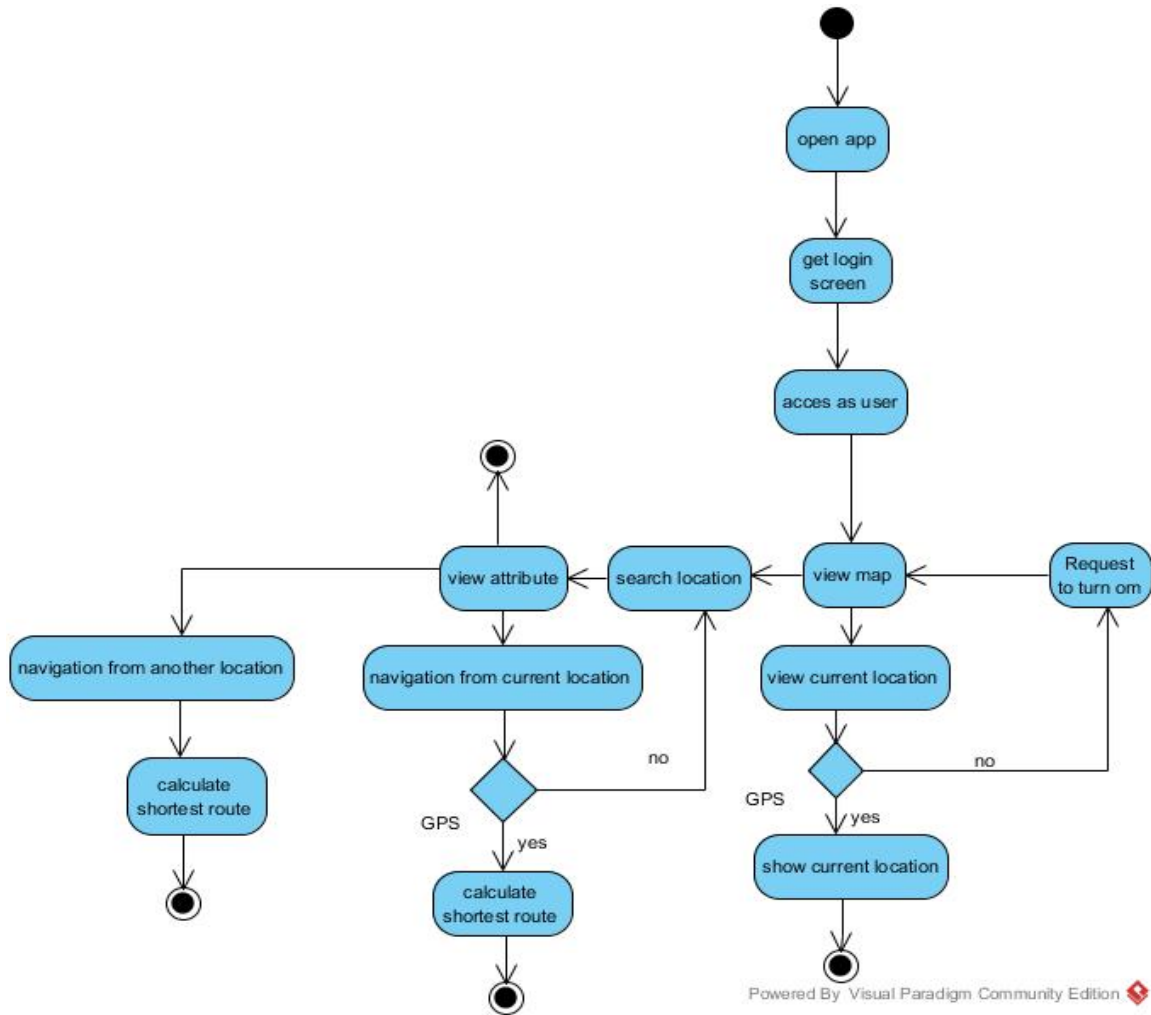


Figure 4-22 Activity Diagram for User

4.6 User interface

This is the log in screen of the GUI. This will start when user opens the app. The administrator will have to log in to the system and a user can enter as guest.

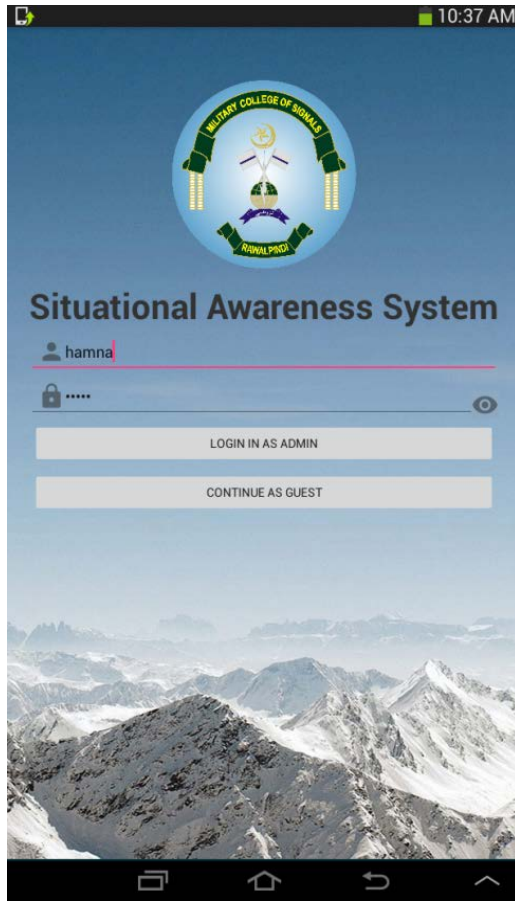


Figure 4-23 Main Menu Screen

Administrator or the user will be directed to the map view screen which will display all the tagged locations that are in the database

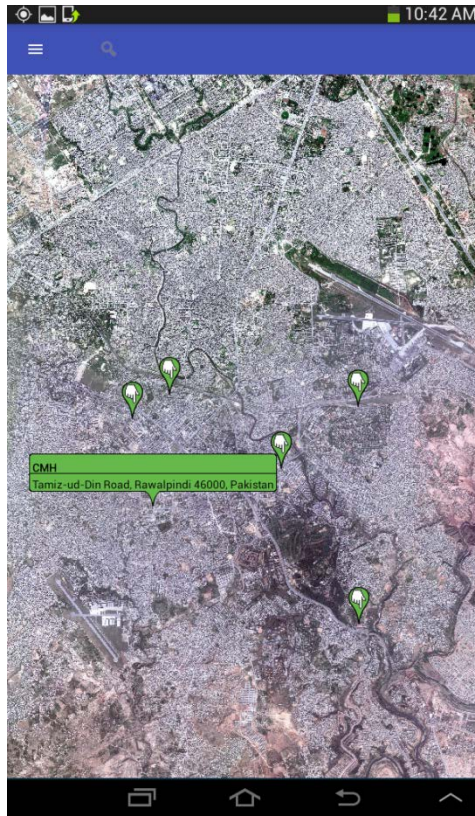


Figure 4-24 Map View Screen

This is the drawer activity screen which consists of different options for the administrator. This screen will not be visible to user who log in as a guest

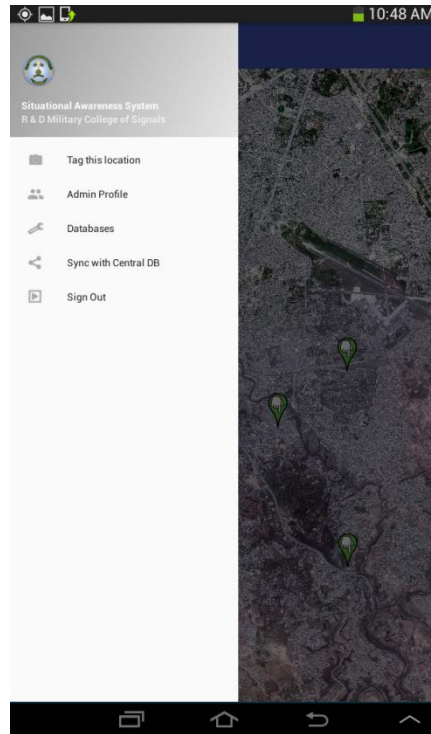


Figure 4-25 Drawer Activity Screen

This is the location description screen which shows the details of the selected location. The administrator can edit or delete a location, clicking the edit button will enable the save button. Edit and delete buttons will not be available for the users signed in as a guest.

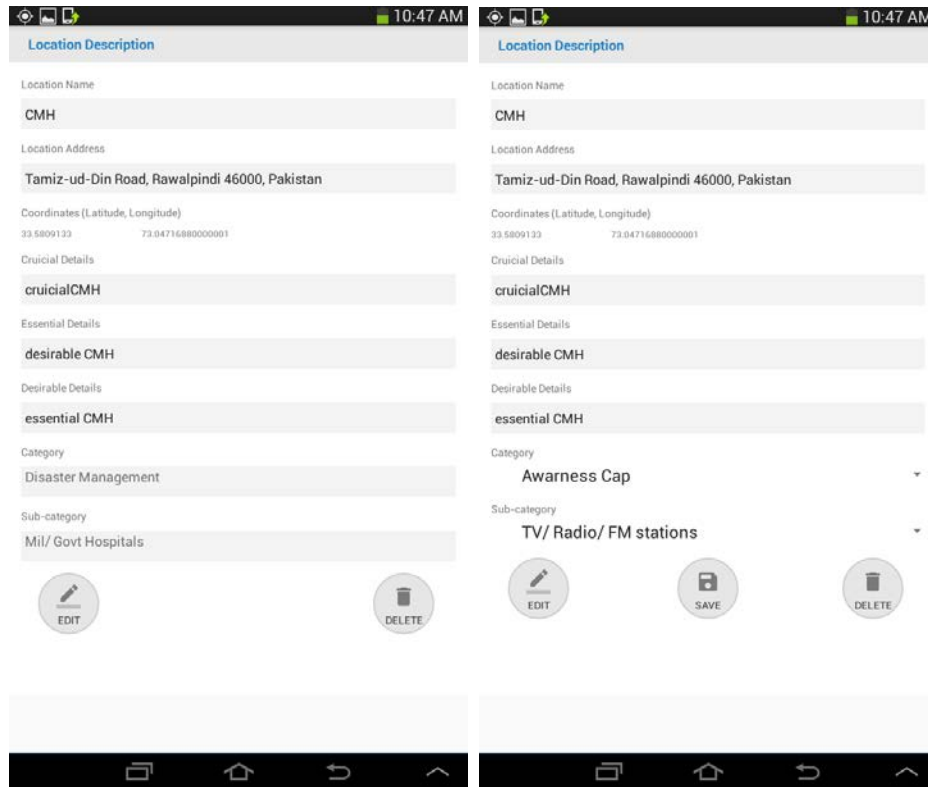


Figure 4-26 Location Description screen

This is the administrator profile screen. The administrator can edit their username and password by clicking the edit button.

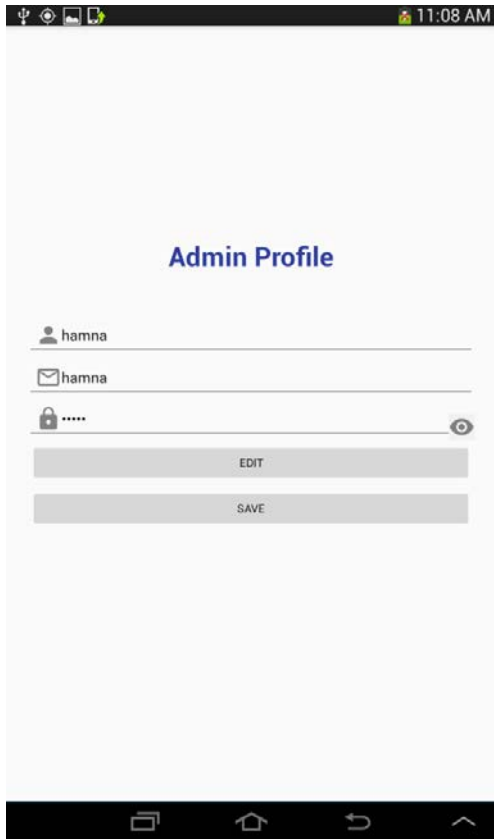


Figure 4-27 Admin profile screen

4.7 Reuse and Relationship to other products

Although there has been many applications developed that provide navigational support and some of them are very popular and used worldwide for example Google Maps. However, our application provides proprietary map imagery for military use. It is much safer to use with respect to security as we maintain the database in our own database servers.

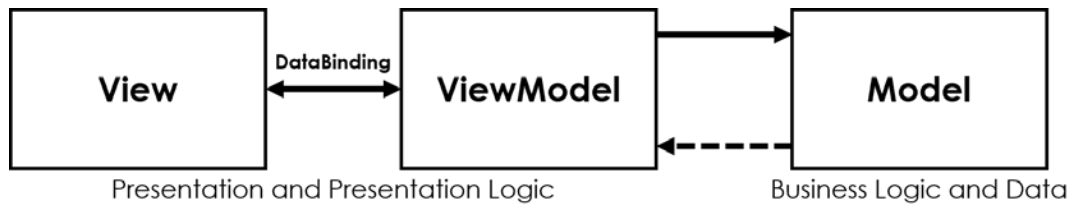
Our application can later be modified for public use. It can be used solely by the civilians who can save their tags for example by adding map imagery of other countries and developing it for tourists who can use it without internet.

4.8 Design and tradeoffs

We have kept the user interface simple and friendly so even a user with only basic knowledge of android applications can use it effectively.

Interface of the system is distinct from the application logic. Layered architecture is used to isolate application logic from the user interface. It can be modeled using multitier layered architecture consisting of three layers i.e.; presentation, application logic and Database. Presentation layer corresponds to elements of the user interface such as text, checkbox item etc., and application logic layer controls the communication of data between the presentation and the Database layer, and is the part where the main logic, user actions and working of the system is defined. In general, it controls the complete behavior of the system, while the Database layer is responsible for handling, fetching and storage

As per the interface and business logic goes, SAS applications follow Model View Model (MVVM) design pattern, which is largely based on model-view-controller (MVC) pattern. It is a specific implementation targeted at UI development platforms which support event-driven programming on the Android platforms using XML and Java.



CHAPTER: 5
SYSTEM IMPLEMENTATION

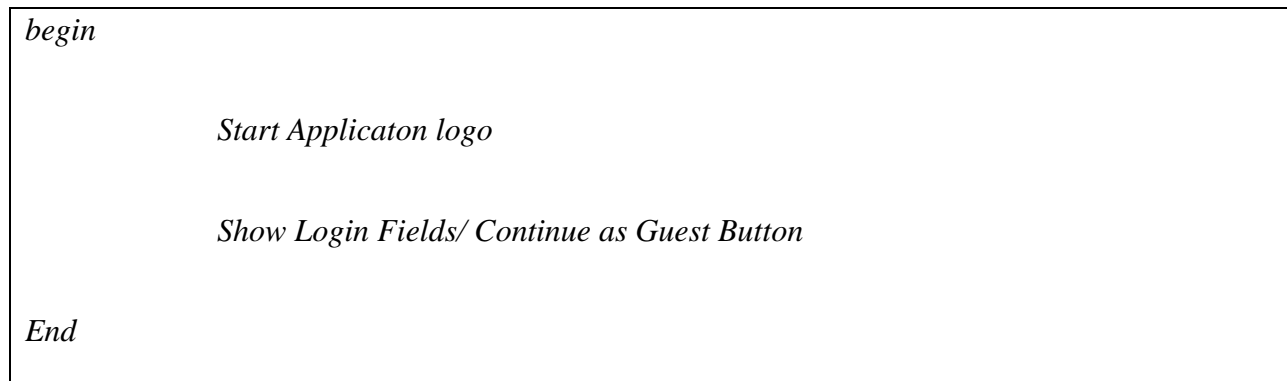
5 SYSTEM IMPLEMENTATION

In the application, an admin can save edit and delete tags while both the admin and the client have the option to search for tags and navigation.

The application works by accessing database all the tagged locations which are saved manually or tagged by the admin by long press. The locations are displayed on the maps with the current location when the application is opened.

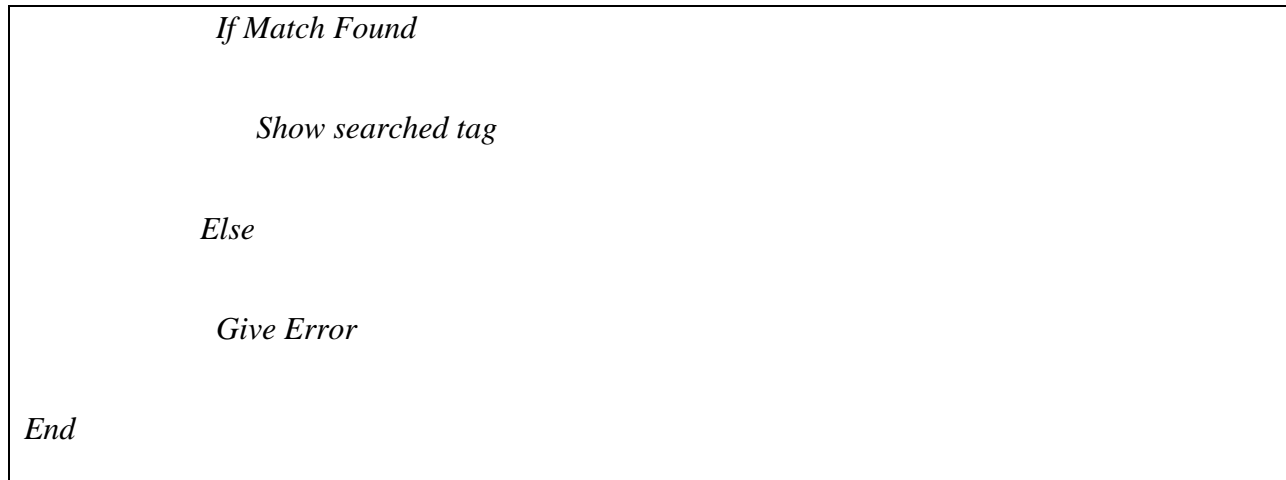
5.1 Pseudo code for components

5.1.1 For User Interface

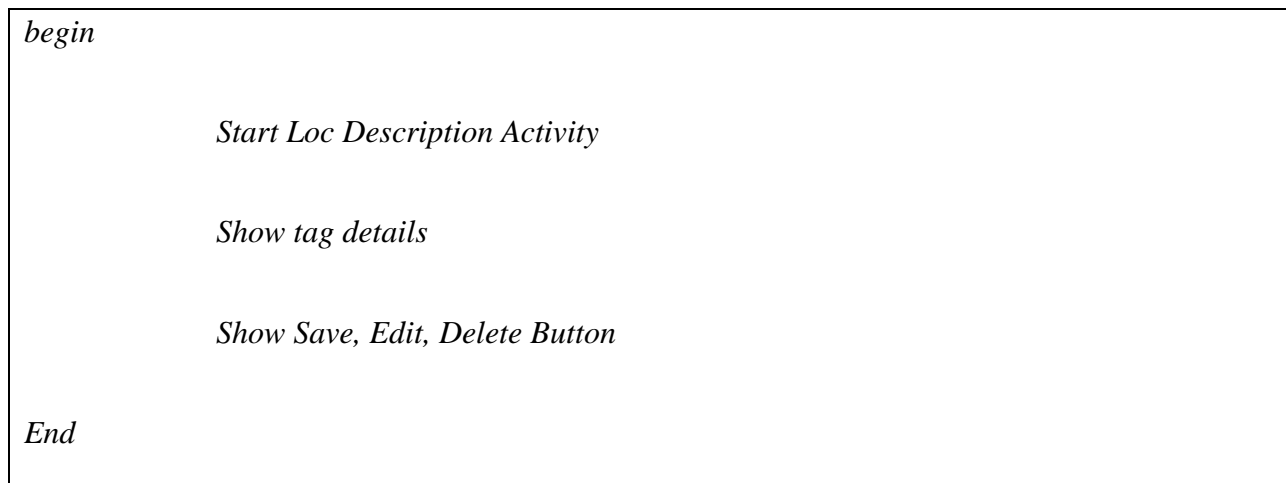


5.1.2 For Search Location



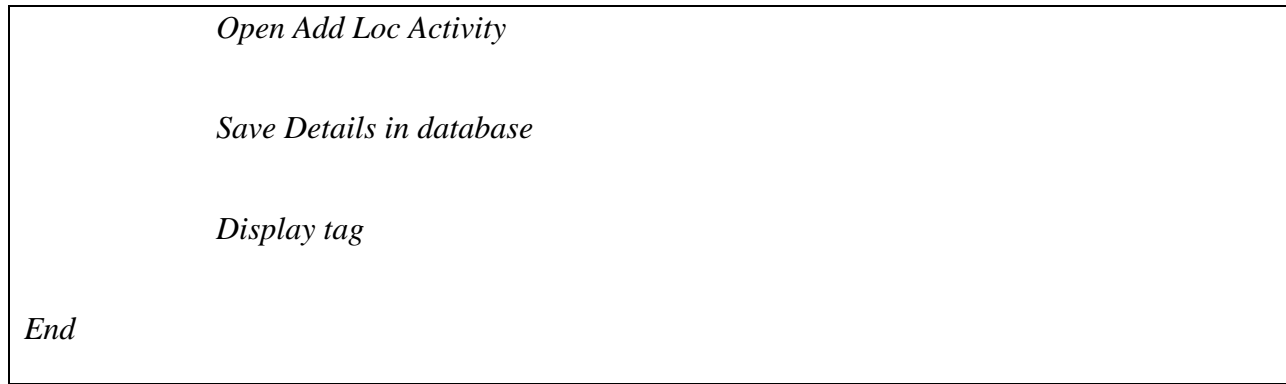


5.1.3 For Location Description

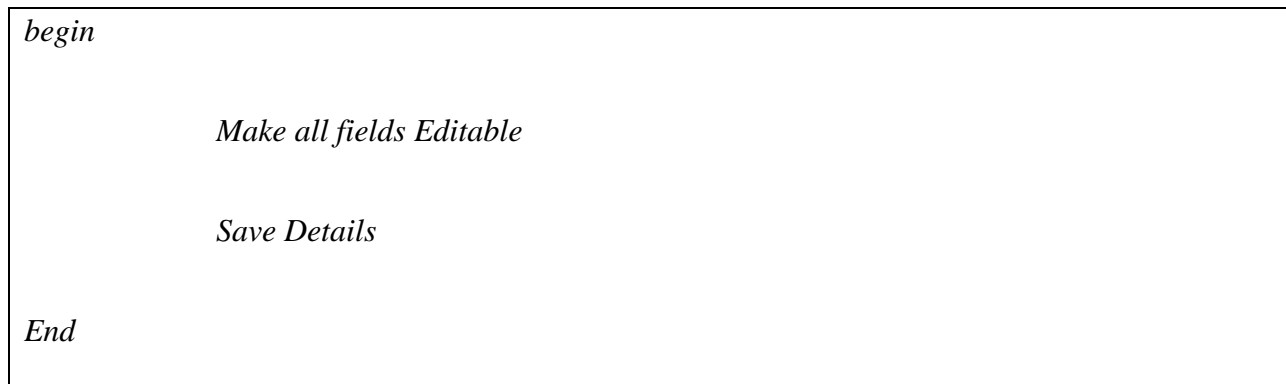


5.1.4 For Add Location

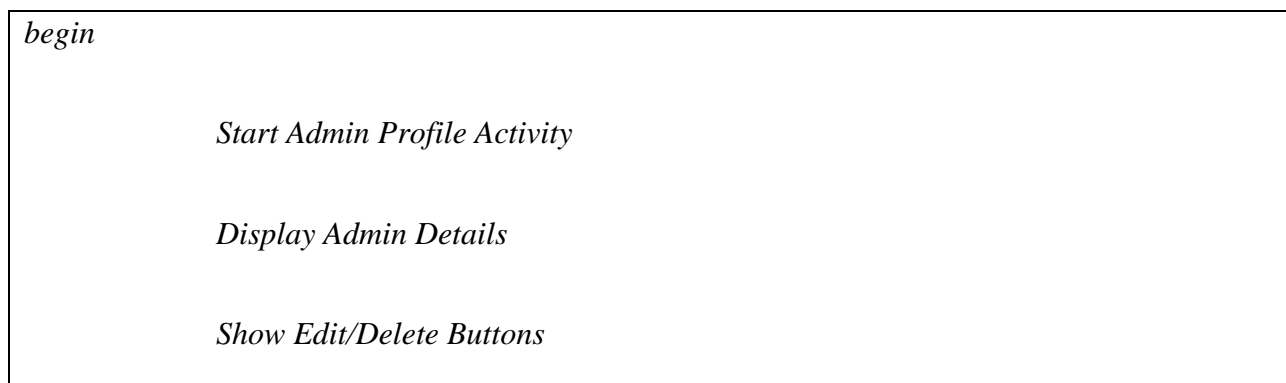




5.1.5 For Edit Location



5.1.6 For Admin Profile



End

5.1.7 For Synchronization with Central Database

begin

Connect with Central Server

IF Connection Successful

Export new entries in Database

ELSE

Display Error

End

5.1.8 For tag current location

begin

Get GPS Coordinates

Start New Loc Activity

Save tag

Display tag

End

CHAPTER:6
ANALYSIS AND EVALUATION

6 Analysis and Evaluation

6.1 Introduction

This test plan document describes the appropriate strategies, process and methodologies used to plan, execute and manage testing of the SAS (Situational Awareness System) Android application project. The test plan will ensure that SAS meets the customer requirements at an accredited level.

Manual Testing will be followed which includes testing a software manually, i.e., without using any automated tool or any script. In this type, the tester takes over the role of an end-user and tests the software to identify any unexpected behavior or bug. Each Unit will be tested separately and then will be integrated with other units; therefore Unit Testing and Integration testing will be followed. For each unit Black box Testing is done and for combined units Acceptance Testing is done.

The test scope includes the Testing of all functional, application performance and use cases requirements listed in the *requirement document*

Software testing, depending on the testing method employed, can be implemented at any time in the development process. However, most of the test effort occurs after the requirements have been defined and the coding process has been completed.

This document includes the plan, scope, approach and procedure of SAS test. The pass/fail criteria of the test items are also defined. The Test Plan document documents and tracks the necessary information required to effectively define the approach to be used in the testing of the product.

6.2 Approach

Acceptance test will be executed based on this acceptance test plan. And after all test cases are executed, a test report will be summarized to show the quality of SAS. Following test approaches will be used in test execution:

- **Unit test.** Developers are responsible for unit test as white-box testing. The implementation of each module and individual component will be verified separately.
- **Integration test.** After the unit test is passed above the defined quality threshold, testers will execute the integration test cases. After all the modules are integrated, it's crucial to test the product as a black-box. End-to-end scenarios will be tested to ensure the communication functionality.
- **Regression test.** After developers fix the bug in one feature, regression test will be executed by testers to ensure that the other functions are not affected.
- **Field test.** Firstly, untrained end users recreate one or more existing (but narrow) mass observation events in the SAS Android Application. A number of observers will be invited to help with evaluation. After that, post event questionnaires will be used to collect quantitative usage data as well as qualitative data and further improvement will be taken into consideration.
- **Positive and negative testing design technique.** This approach will be combined with unit test

and integration test. Test cases are designed in obvious scenarios, which ensure that all functional requirements are satisfied. What's more, different test cases will also be covered to show how the system reacts with invalid operations.

6.3 Features to be tested

Following Features are tested:

- a) Ability to search location from the OSM tiles.
- b) Ability to login as admin.
- c) Ability to edit the admin profile.
- d) Ability to find a route from source to destination.
- e) Ability to add locations by long press on the map.
- f) Ability to add locations by current location on the map.
- g) Ability to display the additional information with the tagged location.
- h) Ability to display error message in case of invalid search.
- i) Ability to display the current location using GPS.
- j) Ability to edit the tagged location information.
- k) Ability to delete a tagged location.
- l) Ability to synchronize with the central database.

6.4 Item Pass/Fail Criteria

Details of the test cases are specified in section Test Deliverables. Following the principles outlined below, a test item would be judged as pass or fail.

Preconditions are met

Inputs are carried out as specified

The result works as what specified in output => Pass

The system doesn't work or not the same as output specification => Fail

6.5 Testing tasks

Develop Test Cases.

Execute tests on the basis of the test cases developed

Report defects during tests if any.

Complete the test report.

Manage the changes made after testing.

6.6 Test Deliverables

Test cases

Output from tools

6.7 Responsibilities

All developers of the project are responsible for the completion of all components testing and integration testing tasks.

6.8 Staffing and training needs

Basics knowledge of testing strategies and techniques is needed for the testing of the project.

All the developers will be testing each other's work and will be actively participating in the development and testing of the project simultaneously.

6.9 Risks and contingencies

6.9.1 Schedule Risk:

The project might get behind schedule so in order to complete the project in time we will need to increase the hours/day that the project is being worked on.

6.9.2 Budget Risk:

The budget will be compensated by using less costly alternatives to fit the budget requirements.

6.9.3 Operational Risks:

Operational risks will be eliminated by Scheduling daily meetings and regular deadlines to meet the goals of the project as well as provide proper communication within the group.

6.9.4 Technical risks:

Technical risks will be eliminated by keeping the once defined requirements constant.

6.9.5 Programmatic Risks:

In case of a programmatic risk the scope of the project will be limited in order to stay inside the constraints of the project.

6.10 Environmental Needs

6.10.1 Hardware

- Mobile with Android platform

6.10.2 Software

- Mobile Platform: Android 6.0.1
- Android Studio (2.2.2)
- Net beans 8.1
- My SQL work bench

6.11 Risks and contingencies

Efforts have been made to remove all and every chance of failure but there are certain unpredictable factors such as network issues, corrupt input data, or system failure that may lead to some issues. Error handling will be applied more deeply to cover all these issues but unforeseen circumstances may happen.

6.12 Test Cases:

Test Case Name	Accessing login screen
Test Case No	1
Description	Testing login screen
Testing Technique Used	Unit Testing
Preconditions	Application should be installed in Android Operating System
Input Values	User name and password

Valid Inputs	Valid user name and password
Steps	Select the SAS android application installed in Android Operating System Fill the desired fields Click “login as admin”
Expected Output	Admin should have access to the SAS application and should see the map view screen
Actual Output	Admin will access to the SAS application and see the map view screen

Table 6-1: Test Cases For Accessing login screen

Test Case Name	Search Location
Test Case No	2
Description	Testing the search location feature
Testing Technique Used	Unit Testing
Preconditions	Database should contain the locations to be searched.
Input Values	Enter the location name
Valid Inputs	Enter the location name with proper spellings
Steps	First select the SAS android application installed in Android Operating System Click on the search bar at the top of the map view screen Search the name of the desired location
Expected Output	The database of the android mobile application should be accessed by SAS and should display the desired location on the map with the help of a marker.

Actual Output	The database of the android mobile application will be accessed by SAS and the desired location is displayed on the map with the help of a marker.
---------------	--

Table 6-2: Test Cases for search location

Test Case Name	Navigation
Test Case No	3
Description	Testing finding a route from source to destination
Testing Technique Used	Unit Testing
Preconditions	The coordinates of the source and destination should be known by the SAS application.
Input Values	Enter source and destination location
Valid Inputs	Enter source and destination location stored in the database
Steps	First select the SAS android application installed in Android Operating System Go to the navigation option Enter the source and destination in the bar at the top, and select navigate.
Expected Output	The route from source to destination should be displayed on the maps after accessing the coordinates of the locations from the database.
Actual Output	--

Table 6-3: Test Cases For navigation

Test Case Name	Add location by long press
Test Case No	4
Description	Testing add location feature on the map view
Testing Technique Used	Integration Testing
Preconditions	The application must be installed on the android operating system The admin should be logged in
Input Values	Longitude and latitude should be known
Valid Inputs	The longitude and latitude of the desired place should be known
Steps	First select the SAS android application installed in Android Operating System Login as admin Long press anywhere on the map view Add location activity should open Fill all the fields Click save
Expected Output	The location should be added on the map and a marker should be displayed.
Actual Output	The location will be added on the map and a marker is displayed

Table 6-4: Test Cases For add location by long press

Test Case Name	Add current location
Test Case No	5
Description	Testing adding current locations using GPS.
Testing Technique	Integration Testing

Used	
Preconditions	<p>The application must be installed on the android operating system.</p> <p>The admin must be logged in to the application</p> <p>GPS should be turned on</p>
Input Values	Longitude and latitude of the current position
Valid Inputs	The longitude and latitude of the current position
Steps	<p>First select the SAS android application installed in Android Operating System</p> <p>Login as admin</p> <p>Open the drawer activity</p> <p>Click on “tag this location” option</p> <p>Fill the add location activity form</p> <p>Click save</p>
Expected Output	The location should be tagged on the map and a marker should be displayed
Actual Output	The location will be tagged on the map and a marker is displayed

Table 6-5: Test Cases For add current location

Test Case Name	Display the additional information
Test Case No	6
Description	Testing display the additional information for the tagged locations
Testing Technique	Integration Testing
Used	

Preconditions	Location must be marked on the map
Input Values	Clicking on the marker
Valid Inputs	Long press on the marker
Steps	First select the SAS android application installed in Android Operating System Click on the tagged locations displayed on the map view or the marker displayed after the user searches for a particular location Long press on the desired location
Expected Output	The additional information should be displayed with the tagged location
Actual Output	The additional information will be displayed with the tagged location

Table 6-6: Test Cases For display the additional information

Test Case Name	Display error message for invalid search
Test Case No	7
Description	Testing display error message in case of invalid search
Testing Technique Used	Unit Testing
Preconditions	The application must be installed on the android operating system. The location should not be in the database
Input Values	Wrong spellings Location which is not present in the database

Valid Inputs	Wrong spellings Location which is not present in the database
Steps	First select the SAS android application installed in Android Operating System Search the desired location in the search bar at the top
Expected Output	An error message should be displayed showing that the search is invalid.
Actual Output	An error message will be displayed showing that the search is invalid.

Table 6-7: Test Cases For display error message for invalid search

Test Case Name	Display the current location
Test Case No	8
Description	Testing display the current location by GPS
Testing Technique Used	Unit Testing
Preconditions	The application must be installed on the android operating system. The GPS is turned on.
Input Values	Coordinates from GPS
Valid Inputs	The application takes the values of longitude and latitude through GPS of the current position.
Steps	First select the SAS android application installed in Android Operating System Turn on the GPS

	The marker is displayed on the screen showing the current location.
Expected Output	The current location should be displayed on the map view
Actual Output	The current location will be displayed on the map view

Table 6-8: Test Cases For display the current position

Test Case Name	Edit the tagged locations
Test Case No	9
Description	Testing tagged location editing
Testing Technique Used	Unit Testing, Integration Testing
Preconditions	Location must be marked on the map view
Input Values	Information to be edited
Valid Inputs	Desired information to be edited
Steps	First select the SAS android application installed in Android Operating System Long press on the marked locations then Press edit button in the location description activity Click save when done
Expected Output	The edited information of the marked locations should be saved in the database
Actual Output	The edited information of the tagged location will be saved in the database

Table 6-9: Test Cases For edit the tagged locations

Test Case Name	Delete the tagged locations
----------------	-----------------------------

Test Case No	10
Description	Testing marked location deletion
Testing Technique Used	Unit Testing, Integration Testing
Preconditions	Location must be marked on the map view
Input Values	Location to be deleted
Valid Inputs	Desired location to be deleted
Steps	First select the SAS android application installed in Android Operating System Long press on the marked locations then Press the delete button in the location description activity
Expected Output	The deleted location should be removed from the database
Actual Output	The deleted location will be removed from the database

Table 6-10: Test Cases For delete the tagged locations

Test Case Name	Synchronization with central database
Test Case No	11
Description	Testing the connection of the application with the central database.
Testing Technique Used	Unit Testing, White box testing
Preconditions	The application is installed on the android operating system Secure wifi connection available The admin must be logged in to the system

Input Values	Select the synchronization option
Valid Inputs	Select the synchronization option
Steps	First open the SAS application Admin should login to the system Open the drawer activity Click on sync with central database option
Expected Output	The entries in the local database should be uploaded and updated every time in the central database
Actual Output	--

Table 6-11: Test Cases For synchronization with central database

Test Case Name	Edit admin profile
Test Case No	12
Description	Testing admin profile editing
Testing Technique Used	Unit Testing
Preconditions	Admin must be registered in the database
Input Values	Information to be edited
Valid Inputs	Desired information to be edited
Steps	First select the SAS android application installed in Android Operating System Admin should login to the system Open the drawer activity Select admin profile Click on edit button Change desired fields Click save when done
Expected Output	The edited information of the admin profile should be saved in

	the database
Actual Output	The edited information of the admin profile will be saved in the database

Table 6-12: Test Cases For edit admin profile

CHAPTER:7
FUTURE WORK

7 FUTURE WORK

A system of this potential always needs more and more work to evolve. The application developed can be reused in the future to proceed with evolution. More features for improving the user's experience can be added in the application. There are a lot of possible changes and additions that can be integrated into the system to improve its performance and functionalities. The system has been made in a modular fashion which enables integrating new features very easy.

Some of the key features which current version doesn't have are: map imagery for other cities of Pakistan, real-time traffic data including oad blocks and sharing your current location. These functions can be added over the time to increase further usability of the system and provide better results with more proficiency.

Another key development that has been talked about a lot is making the system available for public use. This will require a lot of work and separate servers/ database but the end results could be amazing. As we offer completely offline maps, people will not need an internet connection andd will end up saving money on roaming charges, offline search feature will find everything need on the map, auto-suggestion feature will make searching much easier. Bookmarks will be made available to add your favorite locations or travel destinations to the bookmarks list.

CHAPTER:8
CONCLUSION

8 CONCLUSION

8.1 Overview

The main purpose of this project is the development of a system that would allow military personnel to add key locations in sensitive areas for other personnel in the database and to add helpful details regarding each location. Navigation will also be provided between two points. The system will show the current position of the user as soon as they log in to the system. The users can also search for different locations on the maps. The locations are then displayed on the map view screen.

The locations have been stored in the database and a central database is also synchronized with the system to keep a backup of all the data stored in the local dataset. i.e database of the device.. The information of the locations are stored in a Database. Black box testing has been applied on the application together with unit and integration testing.

8.2 Objectives Achieved

The Project developed with R&D MCS as target audience helped to achieve the objectives of learning software development process/cycle, Android studio, SQL, netbeans and integration of databases. It also helped us understand what are the problems we need to face when developing a project in the industry.

CHAPTER:9
BIBLIOGRAPHY

9 BIBLIOGRAPHY

1. <http://gisgeography.com/what-gis-geographic-information-systems>
2. <https://github.com/osmdroid/osmdroid>
3. <https://www.nationalgeographic.org/encyclopedia/geographic-information-system-gis>
4. <http://www.esri.com/what-is-gis>
5. <https://androidapps.com/best-navigation-apps-for-android/>
6. <http://stackoverflow.com/questions/24531391/is-it-possible-to-create-turn-by-turn-gps-navigation-app-on-android-using-google>
7. <https://developers.google.com/maps/documentation/android-api/intents>
8. <https://help.openstreetmap.org/questions/43792/can-i-download-osm-tiles-for-offline-use>
9. <http://osm2vectortiles.org/>
10. <http://www.crowdcompass.com/mobile-conference-app-development.shtml>
11. <http://enc2017.eu/site/>

APPENDIX A

USER MANUAL

Table of Contents

APPENDIX A	93
USER MANUAL	93
10 GENERAL INFORMATION	97
10.1 System Overview:	97
10.2 Organization of the manual:	97
11 SYSTEM SUMMARY	99
11.1 System Configuration:	99
11.2 User Access Levels:	99
11.3 Contingencies:	99
12 GETTING STARTED	101
12.1 Installation:	101
12.2 System Menu:	102
12.3 Exit Application:	102
13 USING THE SYSTEM:	104

1.0 General Information

10 GENERAL INFORMATION

This section explains in general terms the system **SAS** and the purpose for which it is intended.

10.1 System Overview:

SAS is a system that provides a secure way for the military personnel to collect data of key military locations in sensitive area, which could be vulnerable in online mode. Navigation will be provided between tagged key locations. They will be able to search their desired location and get acquainted with it by reading the location description. System admin will have more authority than system user. Central database will be updated whenever a secure connection is found.

10.2 Organization of the manual:

The user's manual consists of five sections: General Information, System Summary, Getting Started, Using the System

1. **General Information** section explains in general terms the system and the purpose for which it is intended.
2. **System Summary** section provides a general overview of the system. The summary outlines the uses of the system's hardware and software requirements, system's configuration, user access levels and system's behavior in case of any contingencies.
3. **Getting Started** section explains how to setup the system and configure it for the first time. The section presents briefly system's settings.
4. **Using the System** section provides a detailed description of system functions.

2.0 System Summary

11 SYSTEM SUMMARY

System Summary section provides a general overview of the system. The summary outlines the uses of the system's hardware and software requirements, system's configuration, user access levels and system's behavior in case of any contingencies.

11.1 System Configuration:

SAS requires android platform to be installed on the mobile. Minimum version required is jellybean 4.1.2 or higher. First time configuration of database is also needed for the system.

11.2 User Access Levels:

The software has two types of clients; system user and the system administrator. Both the types of users have different access level to the system and its data. Each one can perform functions assigned to their respective roles.

Users will use the proprietary map imagery to navigate around an area. They will be able to search and use the tags already in the database in offline mode

The admin will require to login to the system. In addition to the user features, they will have the authority to modify (edit, add or delete locations) the database. Edit their profile and request synchronization with central database.

11.3 Contingencies:

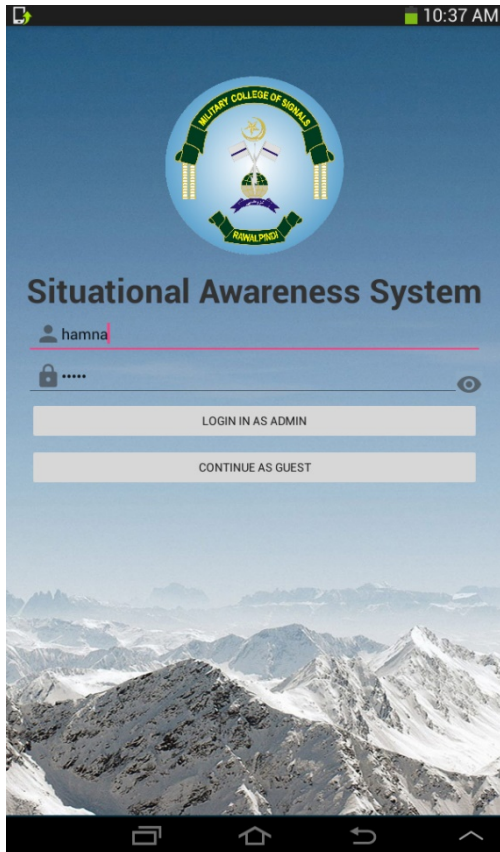
In case of any errors or system crashes, the database will not be affected and tagged locations will remain safe. The currently running session will not be saved hence it will have to be started again. User Settings will also remain same.

3.0 Getting Started

12 GETTING STARTED

Getting Started section explains how to configure the system and install it for the first time use.

The section also presents briefly the system's menu.



12.1 Installation:

Since this is a military application, certain measures are taken to ensure secrecy hence military personnel would install the application on your phone. Also, making sure the phone can connect to the secure army server when needed.

1. The application will already have a built in already set up database with all tag details previously stored in the secure server.
2. For using the synchronization feature, it must be connected with the secure server.
3. The phone must have a working and turned on GPS so the application can be correctly used.

12.2 System Menu:

To use the application, you can either login as an admin or continue as a guest. The main screen of the application will display the high-res maps in both cases. But admin has information edit delete privileges. As an admin, you can either

1. Tag a new location
2. Tag current location
3. Edit details of a stored tag
4. Delete a location
5. Edit admin profile
6. Search for a location
7. Navigate

12.3 Exit Application:

The application can be exited at any time as the database is updated on real-time basis and does not need a save button.

4.0 Using the System

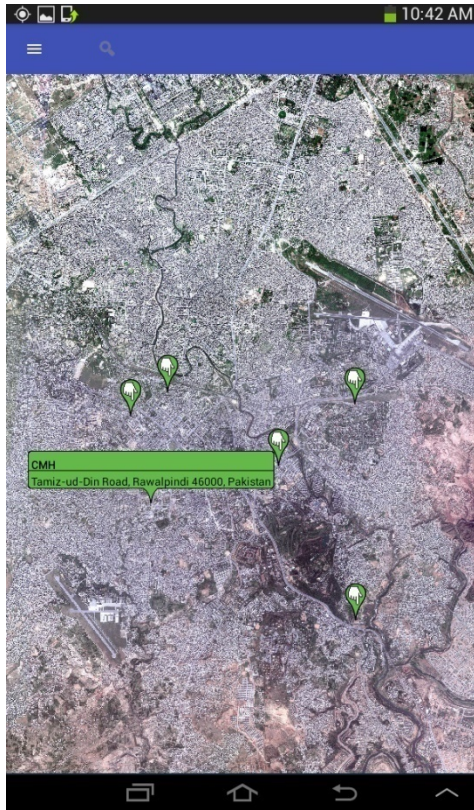
13 USING THE SYSTEM:

This section provides a description of system functions and features. (Written for admin)

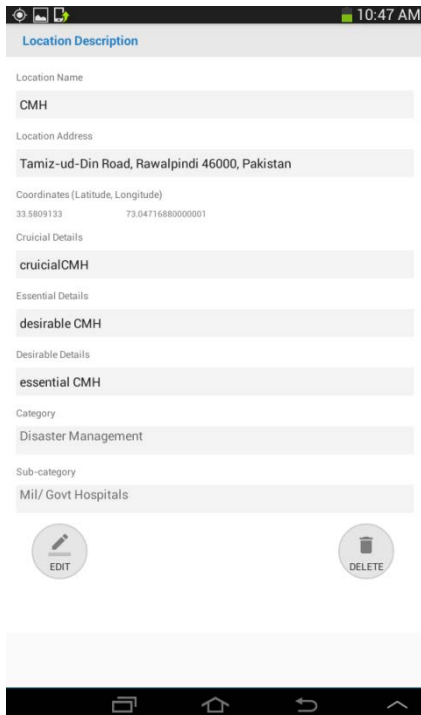
1. Logging into the application will display the high res maps. The user can view this map (currently only for Rwp and Isb city)



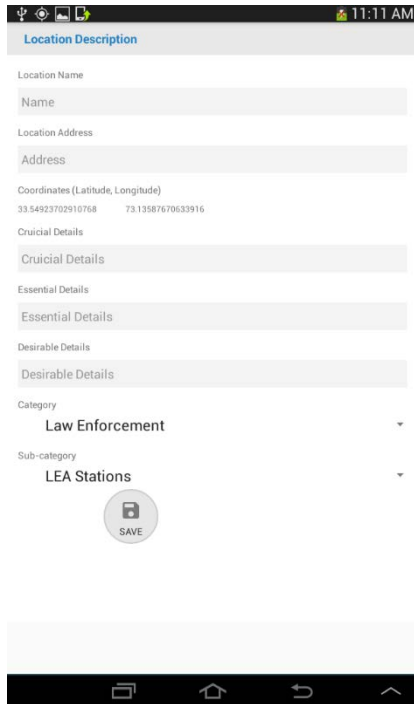
2. User has the option to tap on already tagged locations and view short details.



3. Long pressing the same tag will open detailed information of this tagged location.



4. The user can also edit these details or delete this tag altogether.
5. A new tag can also be placed on a location either by long pressing the screen showing the location or using your GPS to tag the location.



6. Details can then be entered into this tag and it will automatically be then displayed on the screen with the database updated.
7. The admin can also make changes in his profile by opening admin profile from the side menu.
8. Both the admin and the guest have the option to search for a tag. They can search by typing the location name on the search bar. If a match is found the application will display the tagged location or display an error message that the location was not found.
9. The admin and user can navigate from their GPS location to any tagged location in the database.
10. Using the sign out button in the nav bar will sign out the admin from the application.

