

# SUPERIOR UNIVERSITY LAHORE



Faculty of Computer Science & IT

## Final Year Project PROJECT REPORT (Part-II)

### [ARvibe]

Project ID: [FYP-BSCS-F18-022]

#### Project Team

Student Name	Student ID	Program	Contact Number	Email Address
Arwa Riaz	BCSM-F15-237	BSCS	0323 8045791	<a href="mailto:arwa.riaz786@gmail.com">arwa.riaz786@gmail.com</a>
Sara Shahid	BCSM-F15-245	BSCS	0320 9470931	<a href="mailto:saarashahid96@gmail.com">saarashahid96@gmail.com</a>

**Project Supervisor: Dr. Arfan Jaffar**  
(Professor)

# Project Report

[ARvibe]

## Change Record

Author(s)	Version	Date	Notes	Supervisor's Signature
Arwa Riaz Sara Shahid	1.0	February	<Original Draft>	
Arwa Riaz Sara Shahid	2.0	May	<Changes Based on Feedback from Supervisor>	

## APPROVAL

---

### PROJECT SUPERVISOR

Comments: \_\_\_\_\_

---

Name: \_\_\_\_\_

Date: \_\_\_\_\_ Signature: \_\_\_\_\_

---

### PROJECT MANAGER

Comments: \_\_\_\_\_

---

Date: \_\_\_\_\_ Signature: \_\_\_\_\_

### HEAD OF THE DEPARTMENT

Comments: \_\_\_\_\_

---

Date: \_\_\_\_\_ Signature: \_\_\_\_\_

## **Dedication**

*This dissertation is dedicated to my most dear parents for their endless support and encouragement and to my teachers for their motivation, inspiration and most valuable time.*

## **Acknowledgements**

**Praise to Allah Almighty, Lord of the worlds, the Merciful and the Beneficent, who gave us strength, thoughts and co-operative people who enabled us to accomplish this goal and fulfill the required functionalities.**

I and my team mate's special gratitude and regards to everyone who helped and supported us during the completion of this project.

A special thanks to **our supervisor Dr Arfan Jaffar** for being very supportive and appreciating to our concepts and motivation for us throughout the time it took to complete the research to make this project. The inspiration for doing this came from our degree program by our CS department, Superior University, which was most informative and made us explore more innovative and latest technology trends. Our special regards to all our teachers for supporting and appreciating us all the time during degree program, as we believe that, appreciation does impact very much.

Moreover thanks to our friends and surroundings who urged us to compete positively and work on an innovative idea.

## Executive Summary

### Project Title: **ARvibe**

Keeping the current needs and trends, ARvibe is introducing featured social media android application merged with navigation services in more useful way.

With ARvibe, post sharing is possible location wise so it become physically interacting with friends and their memories, who once visited and made a post at a specific place.

Moreover, as the usage of technology has become our need and navigations are a big part of that. ARvibe comprises with **augmented maps**, showing the navigation route from starting to ending point on physical location and **maps with more detailed information** (like discounts and service available within in the selected area) . Other than these core functionalities, messages, calls functions are also available in this app.

With all these functions, the core purpose of ARvibe is to help users with more interactive way and **tie the connection with technology not only digitally but physically and intelligently as well.**

## Table of Contents

Dedication.....	iv
Acknowledgements.....	v
Executive Summary .....	vi
Table of Contents.....	vii
List of Tables .....	ix
Chapter 1 .....	1
Introduction .....	1
1.1. Background.....	2
1.2. Motivations and Challenges .....	3
1.3. Goals and Objectives .....	4
1.4. Literature Review/Existing Solutions .....	4
1.5. Gap Analysis.....	4
1.6. Proposed Solution .....	5
1.7. Project Plan.....	5
1.7.1. Work Breakdown Structure .....	5
1.8. Roles & Responsibility Matrix .....	6
1.9. Gantt chart .....	7
1.10. Report Outline .....	8
iv. Literature Review/Existing Solutions .....	8
Chapter 2 .....	9
Software Requirement Specifications .....	9
1.1. Introduction .....	10
1.1.1. Purpose .....	10
1.1.2. Document Conventions .....	10
1.1.3. Intended Audience and Reading Suggestions .....	11
1.1.4. Product Scope.....	11
1.2. Overall Description .....	12
1.2.1. Product Perspective .....	12
1.2.2. Product Functions.....	12
1.2.3. User Classes and Characteristics.....	13
1.2.4. Operating Environment .....	13
1.2.5. Design and Implementation Constraints.....	13
1.2.6. User Documentation.....	13
1.2.7. Assumptions and Dependencies.....	14
1.3. External Interface Requirements .....	14
1.3.1. User Interfaces .....	14
Screenshots .....	15
1.3.2. Hardware Interfaces.....	19
1.3.3. Software Interfaces.....	20
1.3.4. Communications Interfaces .....	20
1.4. System Features .....	20
1.4.1. User Registration.....	20

1.4.1.1.	Description and Priority .....	20
1.4.1.2.	Stimulus/Response Sequences.....	21
1.4.1.3.	Functional Requirements.....	21
1.4.2.	<b>Augmented Maps</b> .....	21
1.4.2.1.	Description and Priority .....	21
1.4.2.2.	Stimulus/Response Sequences.....	21
1.4.2.3.	Functional Requirements.....	22
1.4.3.	<b>Service/ discounts info on navigations</b> .....	22
1.4.3.1.	Description and Priority .....	22
1.4.3.2.	Stimulus/Response Sequences.....	22
1.4.3.3.	Functional Requirements.....	22
1.4.4.	<b>Calls, messages, file sending</b> .....	22
1.4.4.1.	Description and Priority .....	22
1.4.4.2.	Stimulus/Response Sequences.....	23
1.4.4.3.	Functional Requirements.....	23
1.4.5.	<b>Friend/ Unfriend</b> .....	23
1.4.5.1.	Description and Priority .....	23
1.4.5.2.	Stimulus/Response Sequences.....	23
1.4.5.3.	Functional Requirements.....	23
1.4.6.	<b>Search a user</b> .....	24
1.4.6.1.	Description and Priority .....	24
1.4.6.2.	Stimulus/Response Sequences.....	24
1.4.6.3.	Functional Requirements.....	24
1.4.7.	<b>Update account information</b> .....	24
1.4.7.1.	Description and Priority .....	24
1.4.7.2.	Stimulus/Response Sequences.....	24
1.4.7.3.	Functional Requirements.....	24
1.5.	<b>Other Nonfunctional Requirements</b> .....	25
1.5.1.	Performance Requirements .....	25
1.5.2.	Safety Requirements.....	25
1.5.3.	Security Requirements.....	26
1.5.4.	Software Quality Attributes .....	26
1.5.5.	Business Rules .....	27
Chapter 3	.....	28
Use Case Analysis.....		28
3.1.	Use Case Model.....	29
3.2.	Use Case Descriptions.....	30
Chapter 4	.....	38
System Design .....		38
4.1.	Architecture Diagram.....	39
4.2.	Domain Model.....	40
4.3.	Class Diagram.....	41
4.4.	Entity Relationship Diagram with data dictionary .....	42
4.5.	Sequence / Collaboration Diagram .....	43
4.6.	Operation contracts .....	56
4.7.	Activity Diagram .....	61

4.8. State Transition Diagram: .....	64
4.9. Component and diagram:- .....	69
Chapter 5 .....	70
Implementation .....	70
5.1. Important Flow Control/Pseudo codes.....	71
5.2. Components and Libraries.....	72
Database.....	72
Android Studio.....	72
5.3. Tools and Techniques .....	72
5.4. Best Practices / Coding Standards .....	73
5.5. Version Control .....	73
Chapter 6 .....	74
Testing and Evaluation.....	74
6.1. Use Case Testing .....	75
6.2. Equivalence partitioning .....	89
<b>6.2. Boundary value analysis.....</b>	<b>90</b>
6.3. Data flow testing .....	90
6.4. Unit testing .....	90
6.5. Performance testing.....	90
6.6. Stress Testing.....	90
Chapter 7 .....	91
Summary, Conclusion and Future Enhancements .....	91
7.1. Project Summary.....	92
7.2. Achievements and Improvements.....	92
7.3. Critical Review .....	93
7.4. Lessons Learnt .....	93
7.5. Future Enhancements/Recommendations .....	93
Appendices .....	94
Appendix A: User Manual.....	95
Appendix B: Administrator Manual .....	95
Appendix C: Information / Promotional Material .....	95
Reference and Bibliography .....	98

## List of Figures

Figure 1.....	2
Figure 5- Enhanced Maps.....	12

## List of Tables

Table 1-Gap Analysis.....	4
Table 2- Work Breakdown Structure .....	6
Table 3- User Classes and Characteristics .....	13

# Chapter 1

## **Introduction**

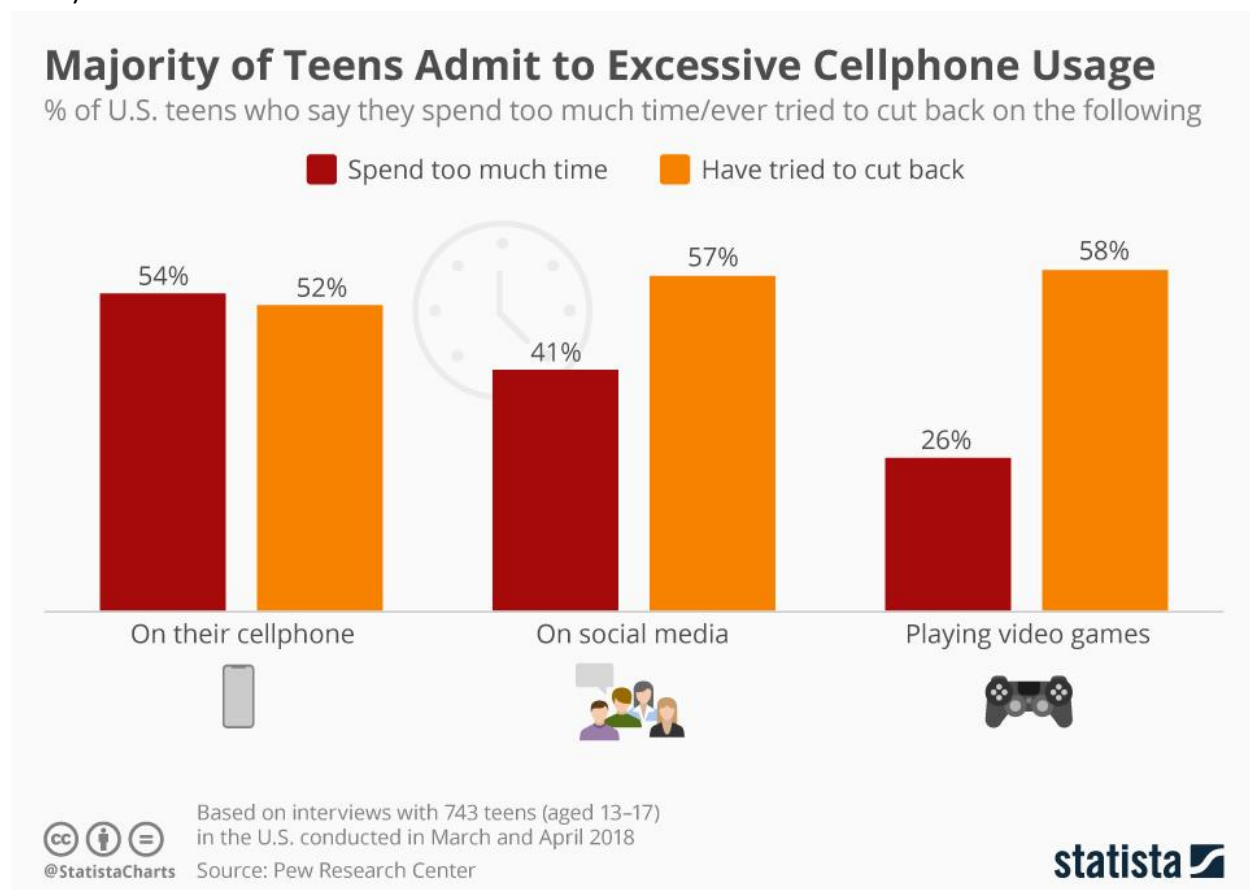
## Chapter 1: Introduction

The idea of ARvibe arrived by observing the massive trend and usage of current social media sites and navigations.

This document is covering all the contents of ARvibe social media app, from problem to detailed solution including work breakdown structure, developing team, system software and hardware requirements, code and testing evaluations and intends for both, developers and common users.

### 1.1. Background

Social media networks have been turned into a society's need that there are billions active users (1.45 billion active users on FB, 200million on instagram and so on, in first quarter of 2018).



**Figure 1**  
**Analysis based on interviews with 743 teens, conducted in March-April 2018**

Keeping the current technological and social trend of people towards digital world and their reliance, this is an android app entertaining and providing navigation services to all android users.

**ARvibe** is a social network to develop physical interaction of users with real world and notifying them and their friends (based on privacy) to see them again either when they go to that remote place again and see their memories at that location or anytime on their phone according to their mood by their facial expressions.

Business places can put offers which would appear on maps and user would also be notified if reached at nearby area. Moreover chat, audio/video, file sending would also be possible.

## 1.2. Motivations and Challenges

### Motivations:

In our daily life almost we all use social media network for our need and for entertainment but in actual we are not getting entertained from current social media network we are just scrolling. Moreover, many and almost all of us use Maps on a daily basis, whether we're driving through the day, navigating public transportation, or walking unfamiliar streets. Taking all this as motivation, **ARvibe is a social media network, merging social media and navigations at one platform and in a very innovative way.**

### Challenges:

- ARvibe will face privacy policy issues at start.
- On Face book and YouTube, the number of monthly users is over one billion, while Twitter, Instagram are in the hundreds of millions. Each day, a staggering amount of content is produced by each of these channels. ARvibe have to beat other social media networks.
- Data management.
- New entrance.

### 1.3. Goals and Objectives

Our main goal is to merge social media and navigations system with innovation. Navigation map is extended with more and synchronized information on daily basis with filtering the info according to the purpose you're navigating.

### 1.4. Literature Review/Existing Solutions

Some of the key players in augmented reality market include Google, WhatsApp, FaceBook, Instagram, Google+, Snapchat, Apple Inc., are investing to improve the quality of software offerings.

These social media networks are **partially working on augmented reality**. Especially Google is now updating their maps by adding augmented reality features. Google showed off new **features for Google Maps** at I/O, including an augmented reality Street View mode to help you follow directions in real time, along with personalized recommendations to help you discover places in your neighborhood. The new AR features combine Google's existing Street View and Maps data with a live feed from your phone's camera to overlay walking directions on top of the real world and help you figure out which way you need to go. Snapchat is playing an essential role spreading the uptake of augmented reality. Snap Chat is also using augmented reality in their filters and games and in 3D Bitmoji which you can use to project a Bitmoji of yourself on to the real world.

### 1.5. Gap Analysis

Table 1-Gap Analysis

Requirements	Existing system	Gap between existing and new system
Longitude/ Latitude wise post sharing	SnapChat , Google, Facebook and other social media networks is not providing this Post sharing (longitude/latitude wise) yet .	User post is saved with location and his/her friend is notified with prior near by location based on privacy.
Augmented Maps	There is no Social media network who provide the Augmented Maps facility to the user	The AR Maps uses the camera on user's smartphone to tell which direction to walk. By using this user can find the shortest way of their destination. User will get so much entertained by using this new and interacting way of navigation.

	but ARvibe will provide this feature.	
Navigation Notifications	Only Google provide this feature but Google is not providing all of these features in single Application .	This feature will provide the facility that user will see the services and discounts on the route they are following user will get many benefits by using this feature. User will be notifying if there is any discount in their route. Only ARvibe provide all these features in single Social media network.

## 1.6. Proposed Solution

**ARvibe** is a social media network to develop physical interaction of users with real world by:

- **Longitude/ Latitude wise post sharing** (dropping their memories in real world and notifying them and their friends (based on privacy))
- **Augmented Maps**
- **Enhanced** (see services and discounts on the route you're following)
- **Messages/ Calls/ File Sending**

## 1.7. Project Plan

### 1.7.1. Work Breakdown Structure

#### 1. Reports / Documentation

- 1.1. Team meeting and Project Start
- 1.2. Final Documentation Introduction
- 1.3. Literature / Markey Survey
- 1.4. Requirements Analysis
- 1.5. System Design
- 1.6. Implementation
- 1.7. Testing & Performance Evaluation
- 1.8. Conclusion & Outlook

- 1.9. End User Documentation
- 1.10. Application Administration Documentation
- 1.11. System Administrator Documentation

## **2. System**

### 2.1. Development Environment

#### 2.1.1. IDE (Android Studio)

#### 2.1.2. Version Control

#### 2.1.3. Server(Local)

#### 2.1.4. Database (firebase )

### 2.2. Presentation Layer

#### 2.2.1. Deliverable 1

#### 2.2.2. Deliverable 2

#### 2.2.3. ...

### 2.3. Business Logic Layer

#### 2.3.1. Deliverable 1

#### 2.3.2. Deliverable 2

#### 2.3.3. ...

### 2.4. Data Management Layer

#### 2.4.1. Deliverable 1

#### 2.4.2. Deliverable 2

#### 2.4.3. ...

### 2.5. Physical Layer

#### 2.5.1. Deliverable 1

#### 2.5.2. Deliverable 2

#### 2.5.3. ...

## **1.8. Roles & Responsibility Matrix**

**Table 2- Work Breakdown Structure**

<b>WBS #</b>	<b>WBS Deliverable</b>	<b>Activity #</b>	<b>Activity Complete to the Deliverable</b>	<b>Duration (# of Days)</b>	<b>Responsible Team Member(s) &amp; Role(s)</b>

1.1	Meeting	1.		1	Arwa, Saara, Supervisor
1.2	Market Survey Statistics	2.	Market survey and analysis	2-3	Arwa, Sara
1.3	Requirements Analysis	3.	System requirements	2	Sara
1.4	System Design	4.	System architecture Design	3	Arwa, Sara
1.5,2.1	Application Development	5.	Coding	30	Arwa, Sara
2.1	Integration	6.	Integrations	20	Arwa, Sara
2.2	Presentation Layer	7.	Front End Development	10	Sara
2.3	Business Logic Layer	8.	Transaction handling	3-4	Arwa
2.4	Database Management	9.	Databases building	5-6	Arwa, Sara
1.10	System Administrator Document	10.	Documentation of admin side working	2-3	Sara
1.2	Documentation Finalization	11.	Finalization of all the documentation	2-3	Arwa

## 1.9. Gantt chart



# Chapter 2

## Software Requirement Specifications

## Chapter 2: Software Requirement Specifications

### 1.1. Introduction

#### 1.1.1. Purpose

Keeping the current technological and social trend of people towards digital world and their reliance, this is an android app entertaining and providing navigation services to all android users.

**ARvibe** is a social network to develop physical interaction of users with real world by dropping their memories in real world and notifying them and their friends (based on privacy) to see them again either when they go to that remote place again and see their memories at that location or anytime on their phone .

Memories will be saved forever to the location and people will get notification every time they visit that specific place.

Augmented Maps are included to add a new and interactive way of using maps.

**Business places can put offers which appear on maps** and user is notified if reached at nearby area. Moreover **chat, audio/video, file sending** would also be possible.

#### 1.1.2. Document Conventions

**Main section:**

Font: Calibri (body)

Face: Bold

Size: 14

Line Spacing: 1.5

Text align: Justified

**Sub section:**

Font: Calibri (body)

Face: Bold

Size: 12

Line Spacing: 1.5

Text align: Justified

**Other text explanation:**

Font: Calibri (body)

Face: Not Bold

Size: 12

Line Spacing: 1.5

Text align: Justified

### **1.1.3. Intended Audience and Reading Suggestions**

The document is intended to be a guide for developers, users, testers and students. The goal of this document is to identify the requirements of ARvibe social networking site.

#### **Developers:**

A developer may directly want to know the requirements by reading the section of USER and SYSTEM REQUIREMENTS and skipping other sections if required.

#### **Users:**

User can use this document to see features and functionalities, and their usage as a guide.

#### **Students:**

Students can use this document for making projects like ARvibe.

### **1.1.4. Product Scope**

The purpose of this app is to renovate the social media posting and giving users more comfort by providing **merger of navigation services providing augmented view maps and plain maps with extended information of services, discounts of business places or any info helping the user during the journey according to their preferences.**

We have a real time database where social media and augmented maps navigations are available to every online user, while extended information on maps are started from covering Lahore, Pakistan and then will go on.

Above all, we hope to provide a very good user experience and service value added convenience.

## 1.2. Overall Description

### 1.2.1. Product Perspective

ARvibe is a self-contained application, introducing the current majorly used services in new way with more convenience and usefulness. Mainly, it is a mobile application running on android smart phones. Users have to register and then login to use the app.

Posting with location, help provide revolutionary change towards social media networks. Navigation like google maps but with more information is more useful through the ARvibe while, info given will be properly authenticated yet, uploaded by the business places themselves.

Moreover, the very core services like messaging, calling, file sharing makes ARvibe a complete social app with many new features.

### 1.2.2. Product Functions

ARvibe is an android app including the features:

- **Augmented navigations**
- **Plain Maps**
- **Services/ Discounts appearance on Navigations according to preferences**



Figure 3- Augmented Maps



Figure 2- Enhanced Maps

### 1.2.3. User Classes and Characteristics

Table 3- User Classes and Characteristics

Class	Characteristics
Admin	Admin class maintains and manages all the navigations updating, user reporting on accounts, accounts management, user queries.
User	User class has access to see the information of every place of map, with the specified attributed it set. Account signup, maintenance, and access to all functionalities available for users.

### 1.2.4. Operating Environment

ARvibe is an android app. As it follows new technology and requiring all the efficiency

- Supports least android version: 4.0 (Ice cream sandwich) and newer
- Language: Java
- Uses Google's maps API for backup support on navigations ( modified for augmented maps and extended information addition)
- A dedicated database server (such as Google firebase) for hosting
- Google vision APIs for interfaces, facial recognition
- Voila Jones Algorithm

### 1.2.5. Design and Implementation Constraints

As app should be efficient and responsive, it requires least memory and battery consumption. For which it is designed in a way being aesthetic and minimalistic. Databases require fully real time, so implementing Firebase is very effective.

Keeping all the functional and non-functional requirements, project timeline is completed in 4 months with no or least compromise on quality, which has to be tackled out later on as soon as possible.

### 1.2.6. User Documentation

ARvibe is following the trend in its new way and designed after considering all the usability requirements and heuristics. So no such need for any other document except the product feature map. Moreover, first time usage after the installation will guide the steps at first.

App itself consists a help menu where the frequently asked question are provided can be asked.

### **1.2.7. Assumptions and Dependencies**

- We assume that system user is literate to the system's minimum software and hardware requirement.
- This system use dependencies of google maps and firebase database also other api's

## **1.3. External Interface Requirements**

ARvibe's user Interface has divided its working in different modules. Each module consists of different working for proper functioning of application. All modules are integrating together to make robust application. These modules provide various features of system.

- Post sharing (dropping memories)
- Augmented maps
- Messaging system
- File sending
- Call function
- Navigation Notifications

### **1.3.1. User Interfaces**

For those who are first time users they will have to fill up registration details. After logging into the system the user see his profile containing the posts shared by user's friends. User is given options to log out , search people, follow people user can also drop memories for their friends user can use Augmented maps and user will have the navigation notifications.

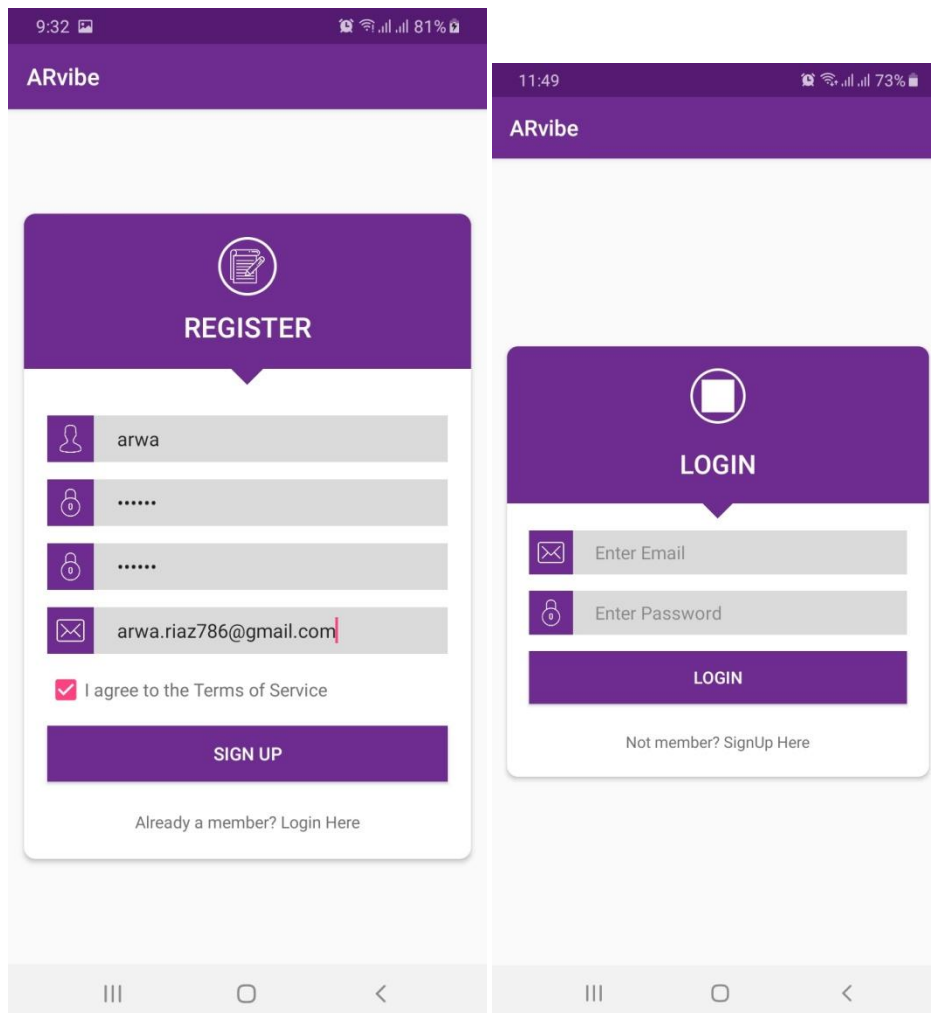
A graphical User interface is available providing following functionalities:

- Drop down menu for function selection
- Post sharing button
- Augmented maps option /button
- Messaging

- Call system
- File sending system
- Scrolling control
- Navigation Notifications

## Screenshots

### 1.3.1.1. Sign Up/ Login



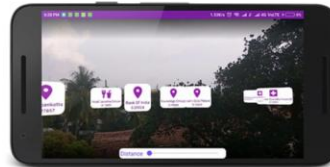
### 1.3.1.2. Augmented Maps

## Welcome

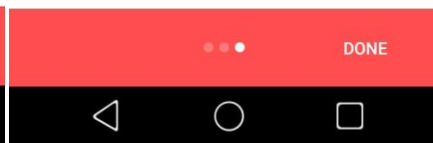
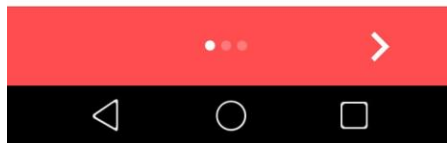


Blink is an Location Based Augmented Reality Navigation Application . Easy to use , Realtime and Fun !

## Augmented Reality Point of Interest Browser



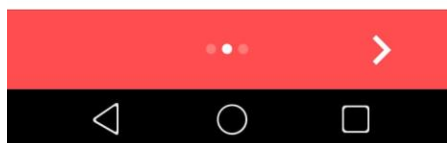
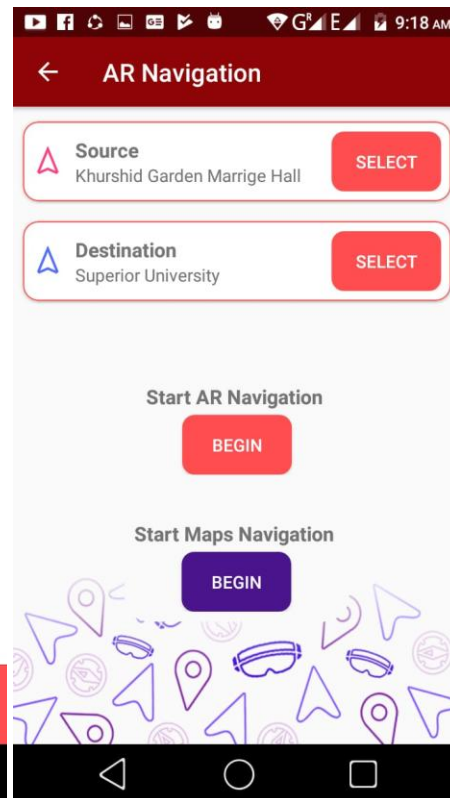
View Places around you in AR and in realtime

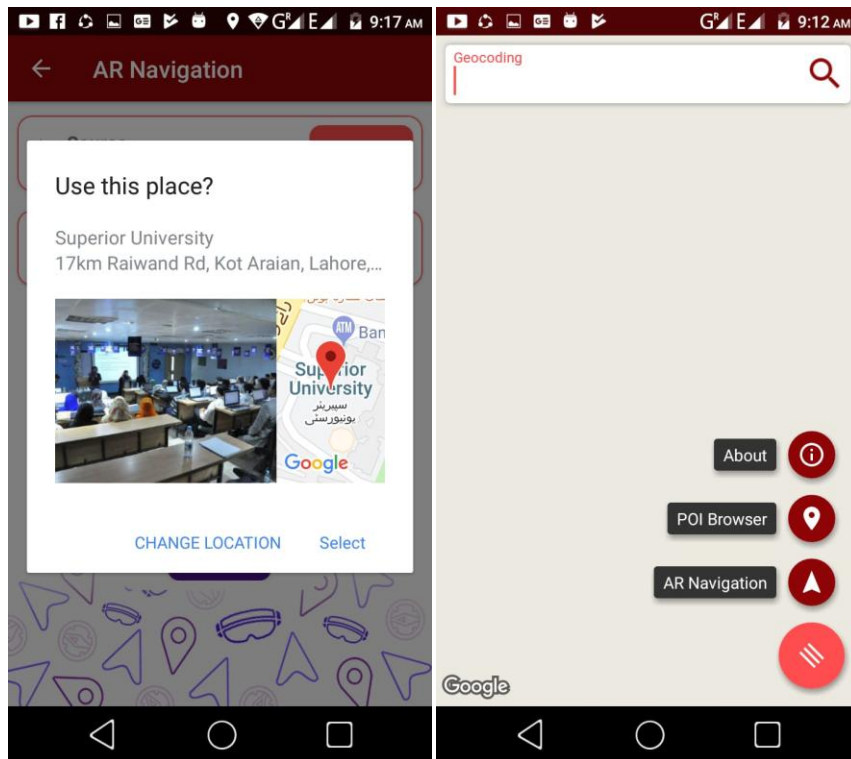


## Augmented Reality Navigation

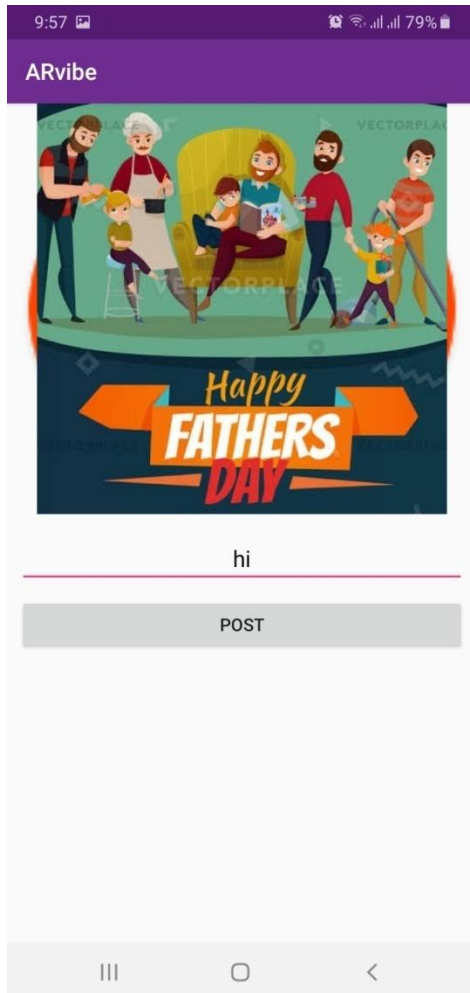


Realtime AR navigation to the destination of your choice . Follow the Dots to reach your destination

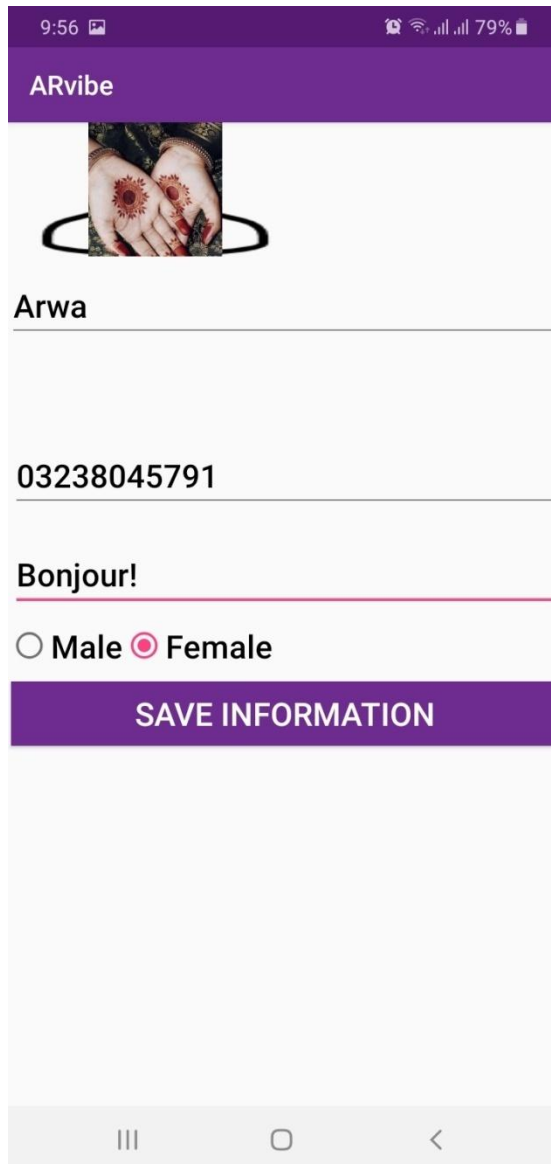




**1.3.1.3. Make a Post( photo, video)**



1.3.1.4.



### 1.3.2. Hardware Interfaces

Device requires

- Android Phone
- Internet modem/ WI FI

### **1.3.3. Software Interfaces**

#### **Operating system**

With minor allowances for the peculiarities of particular operating systems, the application can be modified to act largely platform independent since it has been written in JAVA. The application has however been developed using Windows 10.

#### **Active Appearance Models (AAM)**

AAM will be used for facial recognition algorithms

### **1.3.4. Communications Interfaces**

As the application is social networking application it requires high speed internet for the use of this application. For the suitable use there must be a correct internet connection among the users. Moreover ARvibe is based on HTTP as it is a website and TCP/IP is used for chatting functionality.

## **1.4. System Features**

#### **ARvibe contains:**

- 1- Signup/ Login
- 2- Augmented Maps
- 3- Services/ Discounts info on Navigations
- 4- Plain Maps
- 5- Messages, file sending, calls

### **1.4.1. User Registration**

#### **1.4.1.1. Description and Priority**

##### **1- Sign Up**

Each user of system have interaction with this feature at the first time of app usage.

When application is installed and run for the very first time or user is new to app, the user is presented with an initial registration. This screen prompts user to create an

account. Completing this process will create an account on app, enabling all the attributes provided.

## **2- Login**

If user is registered, he/she can log in from anywhere using the app. User requires enter email and password and then prompts user to front screen, enabling all the attributes.

### **1.4.1.2. Stimulus/Response Sequences**

#### **1- Sign Up**

*1.1. Application launched from the Android home screen*

*1.2. A form gets open to enter the fields*

*1.3. Info goes to server and gets stored in database.*

*1.4. After registration, user is taken to main screen*

#### **2- Login**

*2.1. A form for login gets open*

*2.2. User enters email and password*

*2.3. Info goes to server and gets verified*

*2.4. If user is registered, account gets login*

### **1.4.1.3. Functional Requirements**

**REQ-SF1-1:** *User gets registered and be able to use the app.*

**REQ-SF1-2:** *User gets log in to app and be able to use the app.*

## **1.4.2. Augmented Maps**

### **1.4.2.1. Description and Priority**

Augmented maps have their own priority in the app. They can be used separately or to navigate a post location.

### **1.4.2.2. Stimulus/Response Sequences**

**1-** User sees/ gets notified for a post or searches for a location

**2-** User clicks to navigate the location

3- Selects the augmented map option

4- Augmented map is directed to location (showing some objects at the points)

#### **1.4.2.3. Functional Requirements**

**REQ-SF4-1:** Augmented navigation to a location

**REQ-SF4-2:** Navigate to a post location

### **1.4.3. Service/ discounts info on navigations**

#### **1.4.3.1. Description and Priority**

This feature has the highest priority as it is the main and biggest service being provided and used these days. Navigations have become the prior and need of the day now.

ARvibe presents navigations in extended way by showing some useful info on maps.

User can select the priority, either navigating the path for going to have some dinner or looking for some clinical services provided in a specific area. Map filters and shows all services and discounts provided by eateries, outlets or hospitals etc.

Plain maps without this information can be seen by restricting the filter.

#### **1.4.3.2. Stimulus/Response Sequences**

- 1- User navigates for a location
- 2- Server gets the current location
- 3- Loads the map
- 4- User select/ unselect the info categories
- 5- Map filters out and update the map accordingly

#### **1.4.3.3. Functional Requirements**

**REQ-SF5-1:** Navigation Maps with services and discounts

**REQ-SF5-2:** Navigate a post

### **1.4.4. Calls, messages, file sending**

#### **1.4.4.1. Description and Priority**

User can send messages, files and can call to their friends. This has the core functionality of social media too.

#### **1.4.4.2. Stimulus/Response Sequences**

- 1- User selects the friend
- 2- Friend's chat gets open
- 3- User uploads any type of related content and press send/ press a call button
- 4- Content gets sent or call transfers

#### **1.4.4.3. Functional Requirements**

**REQ-SF6-1:** Messages/ file sending

**REQ-SF6-2:** Make a Call

### **1.4.5. Friend/ Unfriend**

#### **1.4.5.1. Description and Priority**

User can make multiple friends and can unfriend the friends to make a network.

#### **1.4.5.2. Stimulus/Response Sequences**

##### **1- Make Friend**

- i- User search a user
- ii- App returns a list
- iii- User clicks to send request
- iv- Request is being sent to other user
- v- If user accept the request, that user gets added to friend list of both users

##### **2- Unfriend**

- i- User opens his/her friend list
- ii- A list of friends appears
- iii- User selects the user
- iv- Click the menu button and clicks on unfriend
- v- App asks for confirmation
- vi- If user confirms to unfriend, friend is removed from user's list and vice versa

#### **1.4.5.3. Functional Requirements**

**REQ-SF7-1:** Make a user friend

**REQ-SF7-2:** Unfriend a friend

**REQ-SF7-3:** See friends

## **1.4.6. Search a user**

### **1.4.6.1. Description and Priority**

User can search a user either he is friend or a common user.

### **1.4.6.2. Stimulus/Response Sequences**

- 1- User clicks on search bar
- 2- Writes a username, email or phone of user to be searched
- 3- Related search results are provided and exact user is being showed if user is searched through email or phone
- 4- User can make friend, see for info etc

### **1.4.6.3. Functional Requirements**

**REQ-SF8-1:** Search a user

**REQ-SF8-2:** Search a friend

## **1.4.7. Update account information**

### **1.4.7.1. Description and Priority**

It is obvious that information has to be updated time to time. User can update account information.

### **1.4.7.2. Stimulus/Response Sequences**

- 1- User clicks on his profile
- 2- User profile gets open
- 3- User click on information
- 4- User information appears
- 5- User clicks edit button
- 6- User can edit desired information and clicks save
- 7- Information gets saved at server side and user side as well

### **1.4.7.3. Functional Requirements**

**REQ-SF9-1:** See user information

**REQ-SF9-2:** Update information

## 1.5. Other Nonfunctional Requirements

### 1.5.1. Performance Requirements

The system has very perfect performance

#### **Scalability**

System should be able to handle a large number of users. For example handle around thousand users at the same time.

#### **Speed**

The application should be fast. It should not slow down with increase in the number of users. Search functionality should be fast to enable better end-user experience. The system should be quick enough to be able to respond to the user actions within a short period of time. E.g. the search user functionality of the site should perform a quick search among the users on the database.

### 1.5.2. Safety Requirements

ARvibe Social networking site will be a great way to connect with friends. But there are some social networking safety requirements user should always keep in mind.

- **Manage your privacy settings.** Learn about and use the privacy and security settings which will help you control who sees what you post and manage your online experience in a positive way.
- **Protect your device.** Although app does all on your permissions, so permissions should be provided carefully for protection against unauthorized access.
- **Know what action to take.** If someone is harassing or threatening you, remove them from your friends list, block them, and report them to the site administrator.
- **Use strong passwords.** Make sure that your password is at least eight characters long and consists of some combination of letters, numbers, and special characters (for example, +, @, #, or \$).

### 1.5.3. Security Requirements

- During user registration the given email address is validated.
- The password should be at least 8 characters containing at least a small character and one capital a number and a special character.
- Password is stored as a hash value in database.
- We are transferring all data via HTTPS i.e. via SSL so that the data is encrypted during the transit. Thus safeguard the user information.

### 1.5.4. Software Quality Attributes

- **Usability**

User interface should be simple and clear to be understood by end user and must be following usability heuristics.

- **Availability**

The ARvibe should be available at all times. ARvibe should have ensured that there should be minimum or no downtime to ensure better user experience.

- **Reliability**

The ARvibe should be reliable. ARvibe should field correct results if a user performs searches for a person. Also if the user sends message the ARvibe should ensure that the correct message is delivered to the correct destination without any loss or change in content.

- **Testability**

The ARvibe should be testable. A separate test environment should be set up where testers and the quality assurance engineers can test the ARvibe for bugs and incomplete or missed requirements.

- **Maintainability**

The ARvibe should be developed in such a way that ARvibe is extensible. ARvibe should be easy to incorporate new features requirements or accommodate a change in the existing requirements.

### **1.5.5. Business Rules**

This app is not users specific; instead there are only two common users (users, admin). Where user can perform all the functionalities of social media and navigations and admin is responsible for backend databases maintenance and user queries handling.

# Chapter 3

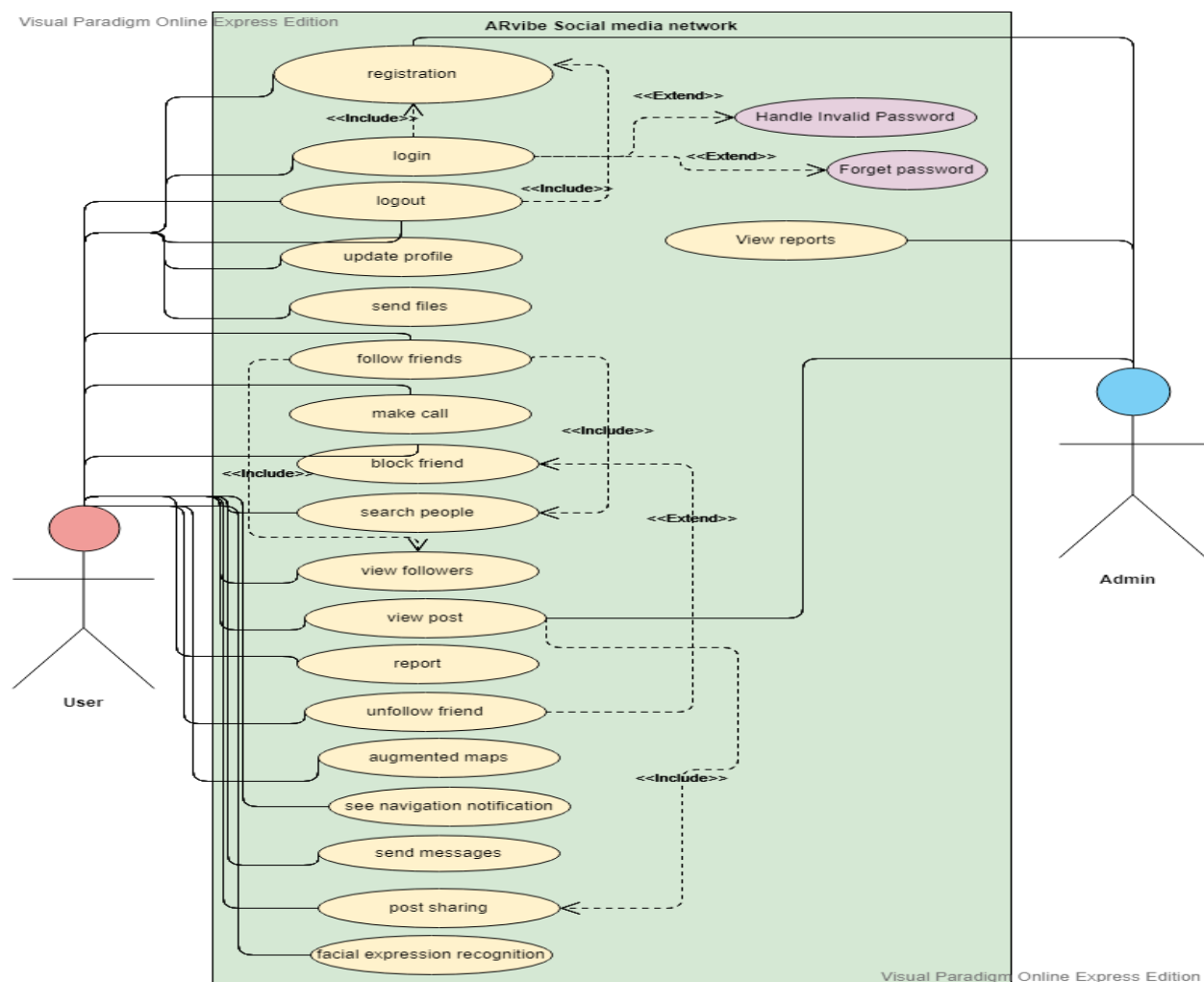
## Use Case Analysis

## Chapter 3: System Analysis

### Usecase Analysis

This chapter includes all the use case diagrams of the functional requirements of the project along with the aggregated use case description. In this chapter we identify the characteristics of the system we are developing and try to define the each use case that define the interaction between the system and the user. In this chapter we define the functionality of the system to identify, clarify, and categorize system requirements. We will discuss and will describe each case separately.

### 3.1. Use Case Model



### 3.2. Use Case Descriptions

<b>Fully dressed use case</b>		
1.	<b>Use case ID</b>	TP_REGISTRATION_01
2.	<b>Use case name</b>	Registration
3.	<b>Purpose</b>	Registration of user account
4.	<b>Description</b>	User will registered and will be able to use the ARvibe
5.	<b>Scope</b>	Open ARvibe social media network
6.	<b>Level</b>	User goal
7.	<b>Primary Actor</b>	Users
8.	<b>Stakeholder</b>	Users and admin
9.	<b>Precondition</b>	Open ARvibe social media network
10.	<b>Post condition</b>	Login
11.	<b>Success Scenario</b>	Successfully login page
12.	<b>Alternative scenario</b>	N/A
13.	<b>Assumptions</b>	User will has good internet access
14.	<b>Special Requirements</b>	Good internet connection

<b>Fully dressed use case</b>		
1.	<b>Use case ID</b>	TP_LOGIN_02
2.	<b>Use case name</b>	Login
3.	<b>Purpose</b>	Login of user account
4.	<b>Description</b>	User will use ARvibe after logging
5.	<b>Scope</b>	Open ARvibe social media network
6.	<b>Level</b>	User goal
7.	<b>Primary Actor</b>	Users
8.	<b>Stakeholder</b>	Users and admin
9.	<b>Precondition</b>	Registration
10.	<b>Post condition</b>	Open user profile
11.	<b>Success Scenario</b>	Successfully open user profile
12.	<b>Alternative Scenario</b>	Already login
13.	<b>Assumptions</b>	User will has good speed of internet and has good internet access
14.	<b>Special Requirements</b>	Good internet access

<b>Fully dressed use case</b>		
1)	<b>Use case ID</b>	TP_LOGOUT_03
2)	<b>Use case name</b>	Logout
3)	<b>Purpose</b>	User will close their account
4)	<b>Scope</b>	Open ARvibe social media network
5)	<b>Level</b>	User goal
6)	<b>Primary Actor</b>	Users
7)	<b>Stakeholder</b>	Users and admin
8)	<b>Precondition</b>	Registration
9)	<b>Post condition</b>	Open login page
10)	<b>Success Scenario</b>	Successfully logout
11)	<b>Alternative scenario</b>	User will direct close the page
12)	<b>Assumptions</b>	User will has good internet access
13)	<b>Special requirements</b>	User will has good internet speed

<b>Fully dressed use case</b>		
1.	<b>Use case ID</b>	TP_UPDATE_PROFILE_04
2.	<b>Use case name</b>	Update profile
3.	<b>Purpose</b>	User will update their profile
4.	<b>Scope</b>	Open User profile
5.	<b>Level</b>	User goal
6.	<b>Primary Actor</b>	Users
7.	<b>Stakeholder</b>	Users and admin
8.	<b>Precondition</b>	Login
9.	<b>Post condition</b>	Profile updated
10.	<b>Success Scenario</b>	Successfully updated
11.	<b>Alternative scenario</b>	Go to profile
12.	<b>Assumptions</b>	User will has high speed internet
13.	<b>Special requirements</b>	User will has good internet

<b>Fully dressed use case</b>		
1.	<b>Use case ID</b>	UC_SENDFILES_5
2.	<b>Use case name</b>	Send files
3.	<b>Purpose</b>	User will have the feature to send files to others
4.	<b>Scope</b>	Open attachment
5.	<b>Level</b>	User goal
6.	<b>Primary Actor</b>	Users
7.	<b>Stakeholder</b>	Users and admin
8.	<b>Precondition</b>	Login
9.	<b>Post condition</b>	Send
10.	<b>Success Scenario</b>	Successfully send
11.	<b>Alternative</b>	N/A
12.	<b>Assumptions</b>	User has good speed internet access
13.	<b>Special requirements</b>	User will have good internet access

<b>Fully dressed use case</b>		
1.	<b>Use case ID</b>	UC_FOLLOWFRIENDS _06
2.	<b>Use case name</b>	Follow friends
3.	<b>Purpose</b>	User will make friends
4.	<b>Scope</b>	Open Friend profile
5.	<b>Level</b>	User goal
6.	<b>Primary Actor</b>	Users
7.	<b>Stakeholder</b>	Users and admin
8.	<b>Precondition</b>	Login
9.	<b>Post condition</b>	Follow
10.	<b>Success Scenario</b>	Successfully followed
11.	<b>Alternative</b>	Just see people profile or don't follow
12.	<b>Assumptions</b>	User has good speed of internet
13.	<b>Special requirements</b>	User will search people to make friends

<b>Fully dressed use case</b>		
1.	<b>Use case ID</b>	UC_BLOCKFOLLOWER _08
2.	<b>Use case name</b>	Block follower
3.	<b>Purpose</b>	User will block friend whom they don't want to see
4.	<b>Scope</b>	Open Friend profile
5.	<b>Level</b>	User goal
6.	<b>Primary Actor</b>	Users
7.	<b>Stakeholder</b>	Users and admin
8.	<b>Precondition</b>	Login
9.	<b>Post condition</b>	Block friend
10.	<b>Success Scenario</b>	Successfully blocked
11.	<b>Alternative</b>	User will un follow friend
12.	<b>Assumptions</b>	User has good internet access
13.	<b>Special requirements</b>	User has good speed

<b>Fully dressed use case</b>		
1.	<b>Use case ID</b>	UC_SEARCHPEOPLE _9
2.	<b>Use case name</b>	Search people
3.	<b>Purpose</b>	User will search friends
4.	<b>Scope</b>	Open search bar
5.	<b>Level</b>	User goal
6.	<b>Primary Actor</b>	Users
7.	<b>Stakeholder</b>	Users and admin
8.	<b>Precondition</b>	Login
9.	<b>Post condition</b>	Friend searched
10.	<b>Success Scenario</b>	Successfully searched
11.	<b>Alternative</b>	N/A
12.	<b>Assumptions</b>	User has good internet access
13.	<b>Special requirements</b>	User will have good speed of internet

<b>Fully dressed use case</b>		
1.	<b>Use case ID</b>	UC_VIEWFOLLOWERS _10
2.	<b>Use case name</b>	View followers
3.	<b>Purpose</b>	User will view friends
4.	<b>Scope</b>	Open followers list
5.	<b>Level</b>	User goal
6.	<b>Primary Actor</b>	Users
7.	<b>Stakeholder</b>	Users and admin
8.	<b>Precondition</b>	Login
9.	<b>Post condition</b>	View Friend list
10.	<b>Success Scenario</b>	Successfully viewed
11.	<b>Alternative</b>	User will search and see friend's profile
12.	<b>Assumptions</b>	User has good internet access
13.	<b>Special requirements</b>	User will have good speed of internet access

<b>Fully dressed use case</b>		
1.	<b>Use case ID</b>	UC_VIEWPOST _11
2.	<b>Use case name</b>	View Post
3.	<b>Purpose</b>	To view the posts
4.	<b>Scope</b>	Open profile
5.	<b>Level</b>	User goal
6.	<b>Primary Actor</b>	Users
7.	<b>Stakeholder</b>	Users and admin
8.	<b>Precondition</b>	Login
9.	<b>Post condition</b>	Post viewed
10.	<b>Success Scenario</b>	Successfully viewed
11.	<b>Alternative</b>	N/A
12.	<b>Assumptions</b>	User has good internet speed
13.	<b>Special requirements</b>	User will have good internet access

<b>Fully dressed use case</b>		
1.	<b>Use case ID</b>	UC_Report _12
2.	<b>Use case name</b>	Report user
3.	<b>Purpose</b>	Report friend
4.	<b>Scope</b>	Open user profile
5.	<b>Level</b>	User goal
6.	<b>Primary Actor</b>	Users
7.	<b>Stakeholder</b>	Users and admin
8.	<b>Precondition</b>	Login
9.	<b>Post condition</b>	Reported
10.	<b>Success Scenario</b>	Successfully report
11.	<b>Alternative</b>	Block friend
12.	<b>Assumptions</b>	User has good internet access
13.	<b>Special requirements</b>	User will have good speed

<b>Fully dressed use case</b>		
1.	<b>Use case ID</b>	UC_AUGMENTEDMAPS _13
2.	<b>Use case name</b>	Augmented maps
3.	<b>Purpose</b>	User will use service of augmented maps
4.	<b>Scope</b>	Open Augmented maps
5.	<b>Level</b>	User goal
6.	<b>Primary Actor</b>	Users
7.	<b>Stakeholder</b>	Users and admin
8.	<b>Precondition</b>	Login
9.	<b>Post condition</b>	View augmented maps
10.	<b>Success Scenario</b>	Successfully viewed
11.	<b>Alternative</b>	Use Enhanced maps
12.	<b>Assumptions</b>	User has good internet access
13.	<b>Special requirements</b>	User will have good speed of pc and internet

<b>Fully dressed use case</b>		
1.	<b>Use case ID</b>	UC_NAVIGATIONNOTIFICATION _14
2.	<b>Use case name</b>	Navigation notification
3.	<b>Purpose</b>	See notification
4.	<b>Scope</b>	Open notification
5.	<b>Level</b>	User goal
6.	<b>Primary Actor</b>	Users
7.	<b>Stakeholder</b>	Users and admin
8.	<b>Precondition</b>	Login
9.	<b>Post condition</b>	View notifications
10.	<b>Success Scenario</b>	Successfully viewed
11.	<b>Alternative</b>	N/A
12.	<b>Assumptions</b>	User has good internet access
13.	<b>Special Requirements</b>	User has good speed internet

<b>Fully dressed use case</b>		
1.	<b>Use case ID</b>	UC_SENDDMESSAGES _15
2.	<b>Use case name</b>	Send messages
3.	<b>Purpose</b>	Send messages to friends or other people
4.	<b>Scope</b>	Open inbox
5.	<b>Level</b>	User goal
6.	<b>Primary Actor</b>	Users
7.	<b>Stakeholder</b>	Users and admin
8.	<b>Precondition</b>	Login
9.	<b>Post condition</b>	Message send
10.	<b>Success Scenario</b>	Successfully sent message
11.	<b>Alternative</b>	Mail
12.	<b>Assumptions</b>	User has good internet access
13.	<b>Special requirements</b>	User will have good speed of internet

<b>Fully dressed use case</b>		
1.	<b>Use case ID</b>	UC_POSTSHARING _16
2.	<b>Use case name</b>	Post sharing
3.	<b>Purpose</b>	User will share any post to any one
4.	<b>Scope</b>	Open profile
5.	<b>Level</b>	User goal
6.	<b>Primary Actor</b>	Users
7.	<b>Stakeholder</b>	Users and admin
8.	<b>Precondition</b>	Login
9.	<b>Post condition</b>	Post shared
10.	<b>Success Scenario</b>	Successfully shared
11.	<b>Alternative</b>	Sent in inbox
12.	<b>Assumptions</b>	User has good speed of internet
13.	<b>Special requirements</b>	User will have good internet access

<b>Fully dressed use case</b>		
1.	<b>Use case ID</b>	UC_UNFOLLOWFRIENDS _18
2.	<b>Use case name</b>	Un Follow friends
3.	<b>Purpose</b>	Un Follow friends whom user don't want to see
4.	<b>Scope</b>	Open Friend profile
5.	<b>Level</b>	User goal
6.	<b>Primary Actor</b>	Users
7.	<b>Stakeholder</b>	Users and admin
8.	<b>Precondition</b>	Login
9.	<b>Post condition</b>	Un Follow
10.	<b>Success Scenario</b>	Successfully un followed
11.	<b>Alternative</b>	Block friend
12.	<b>Assumptions</b>	User has good internet speed
13.	<b>Special requirements</b>	User will has good internet access

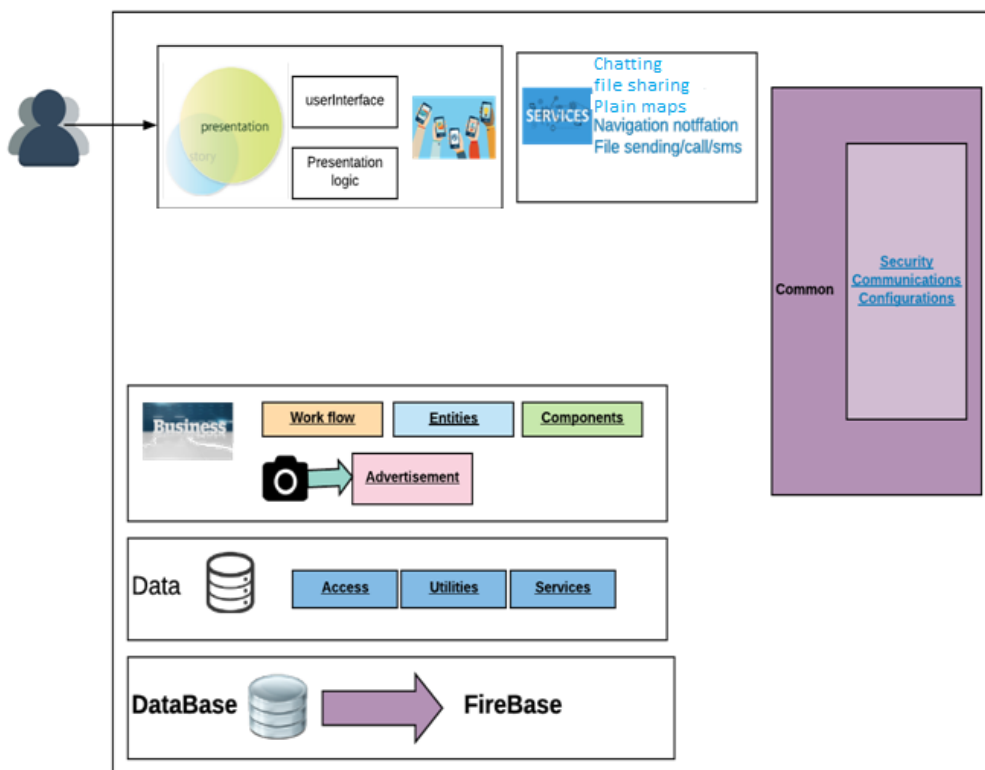
# Chapter 4

## System Design

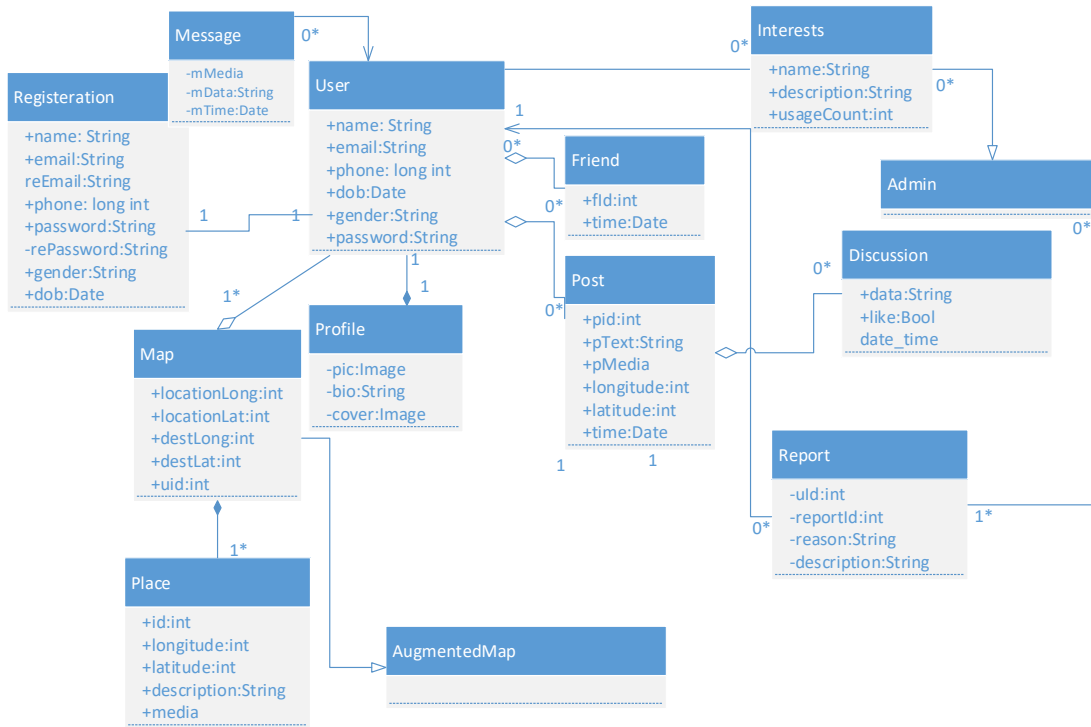
## Chapter 4: System Design

This chapter is covering all the system design. It is a detailed visual implementation of all system functions and its architecture, data flow, core entities and their relationship, system classes to be used and sequence diagrams.

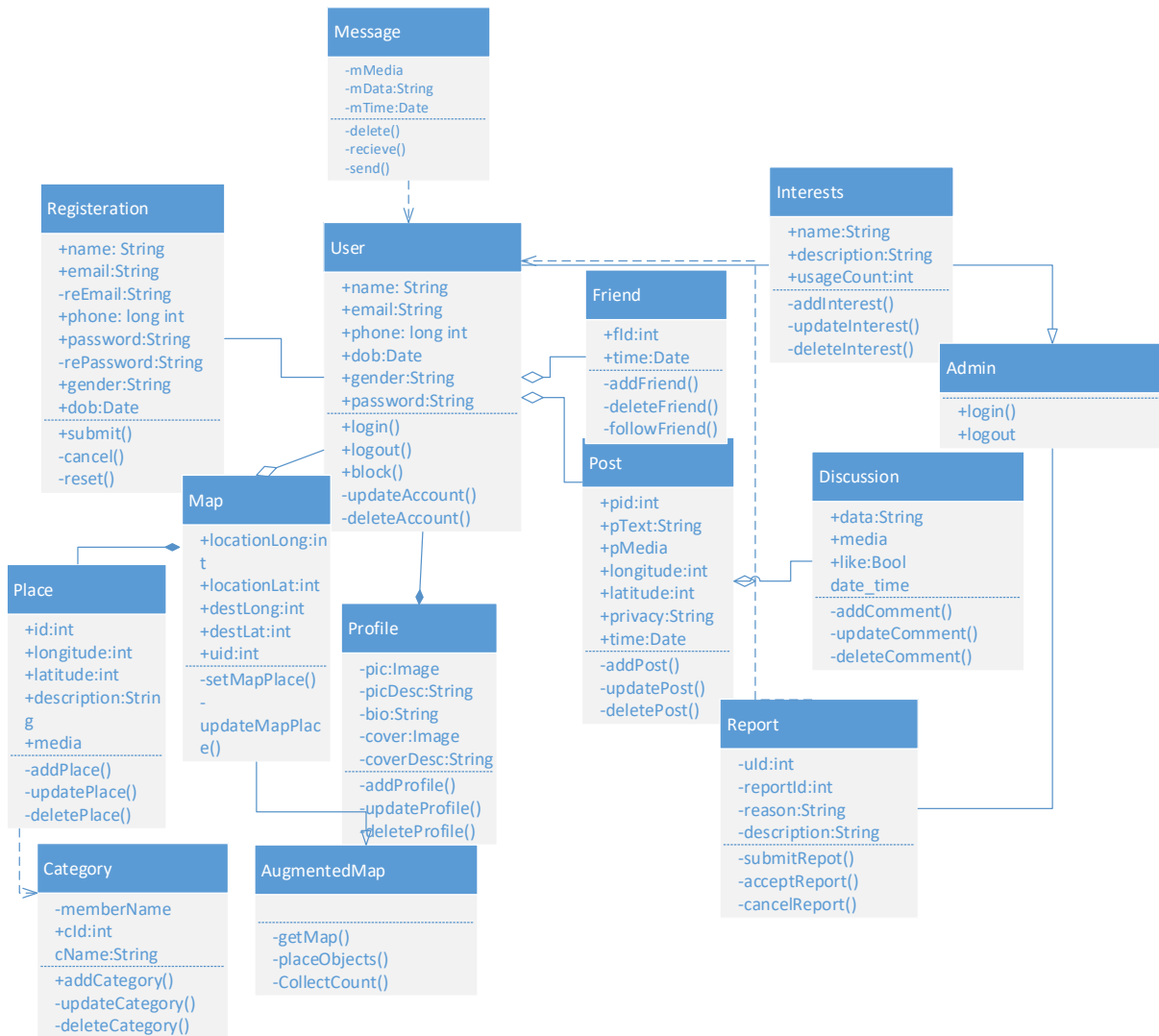
### 4.1. Architecture Diagram



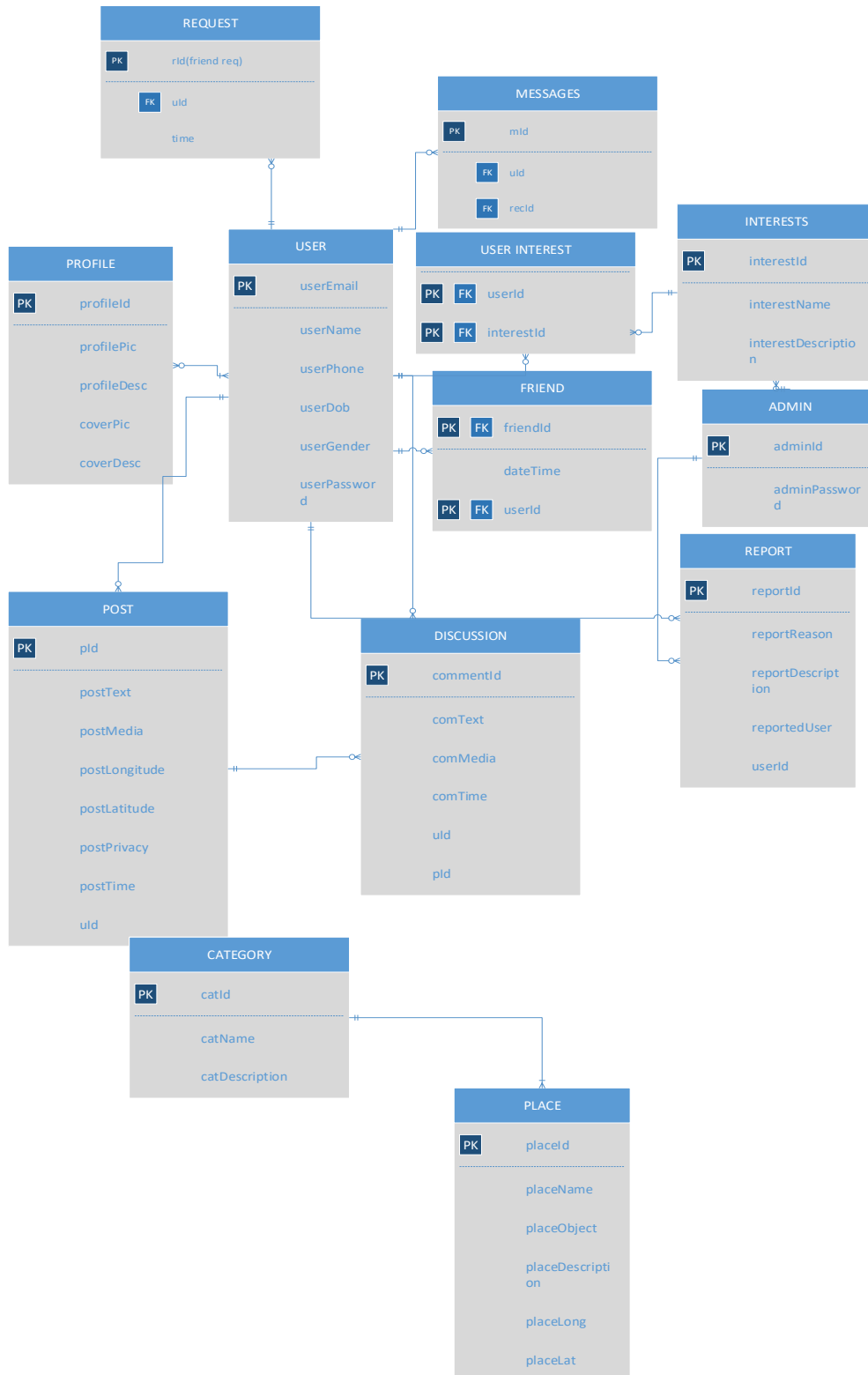
## 4.2. Domain Model



### 4.3. Class Diagram

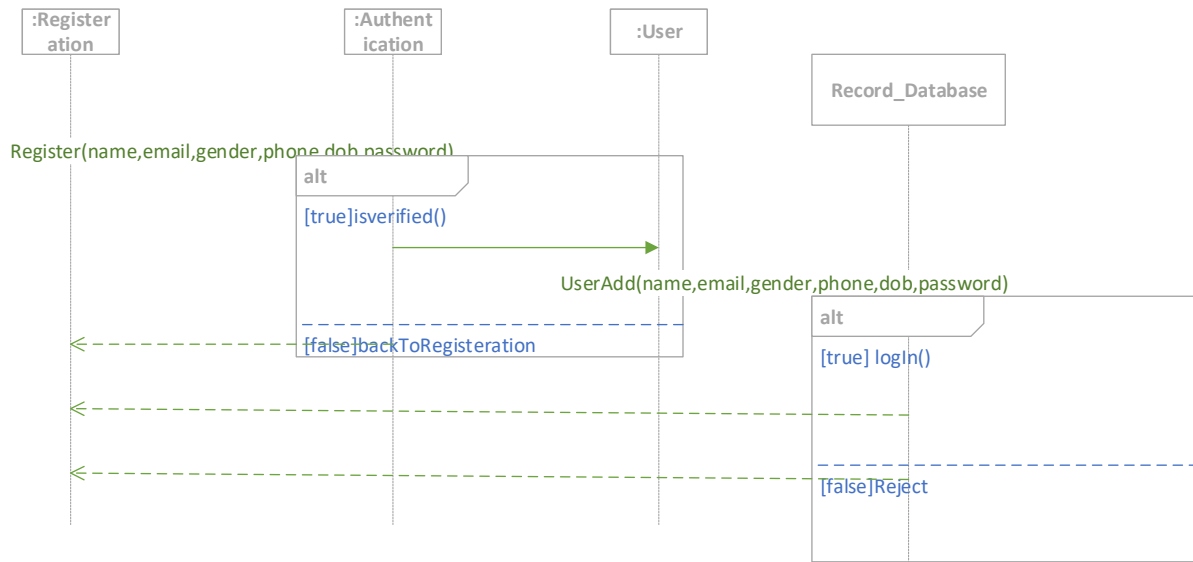


## 4.4. Entity Relationship Diagram with data dictionary

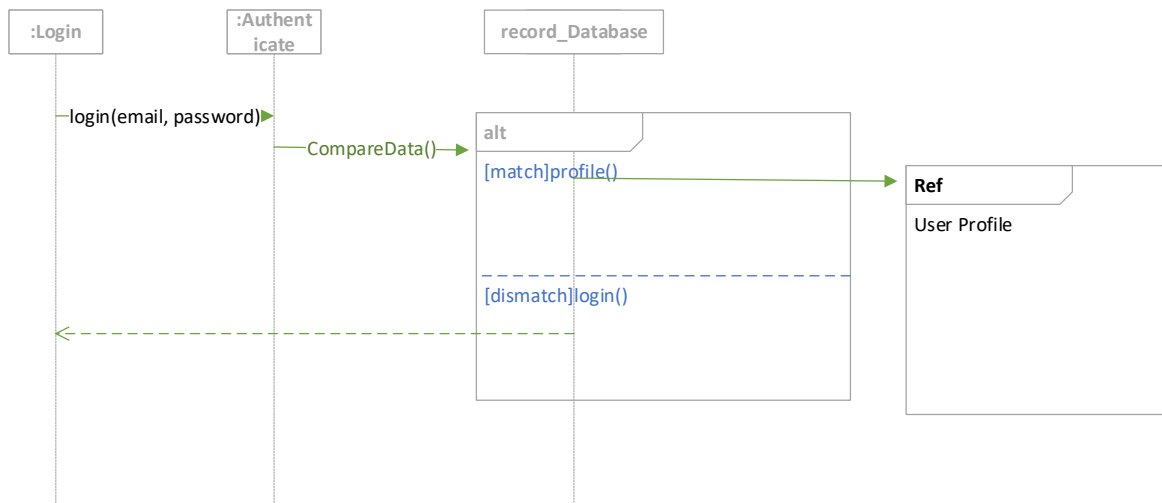


## 4.5. Sequence / Collaboration Diagram

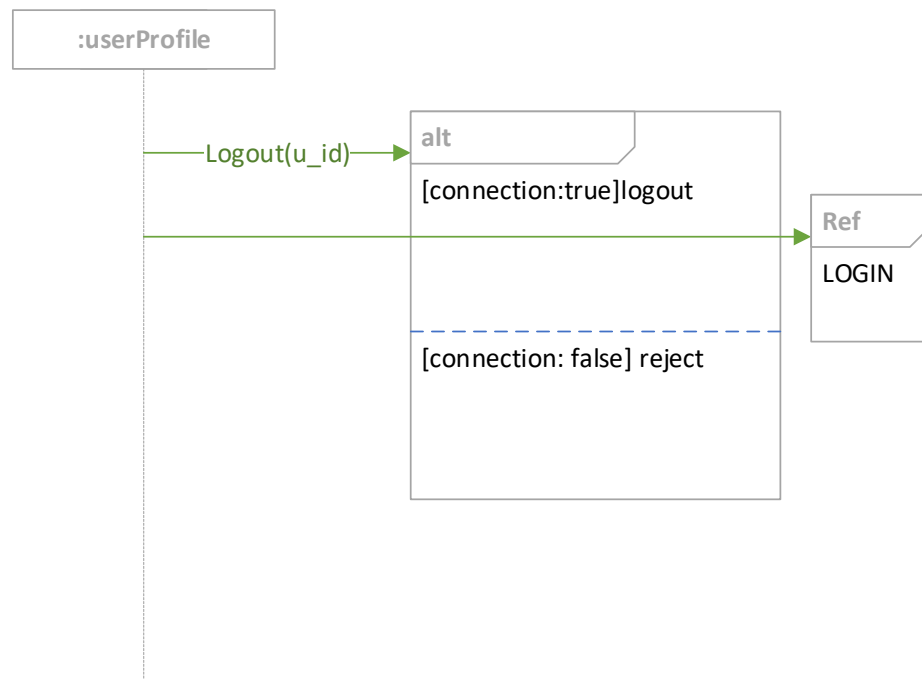
### 1. Registration



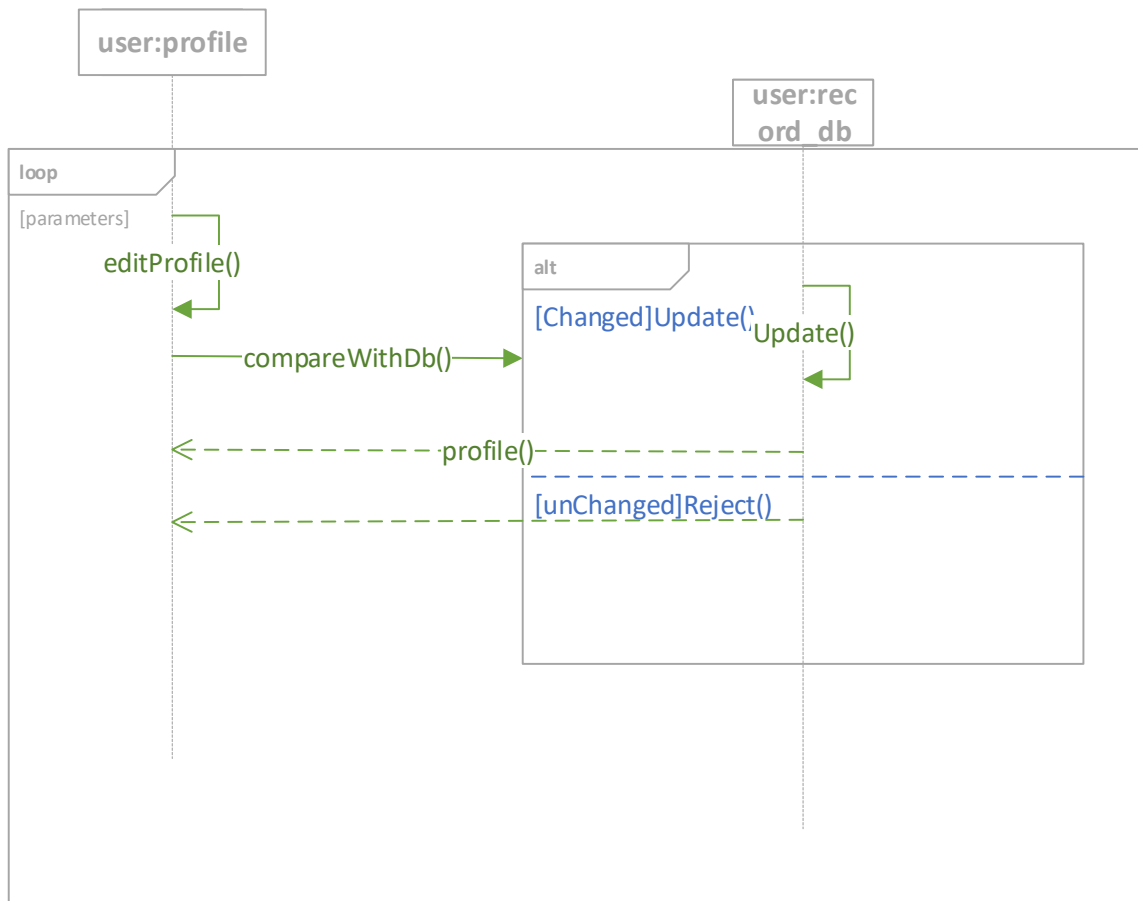
### 2. Login



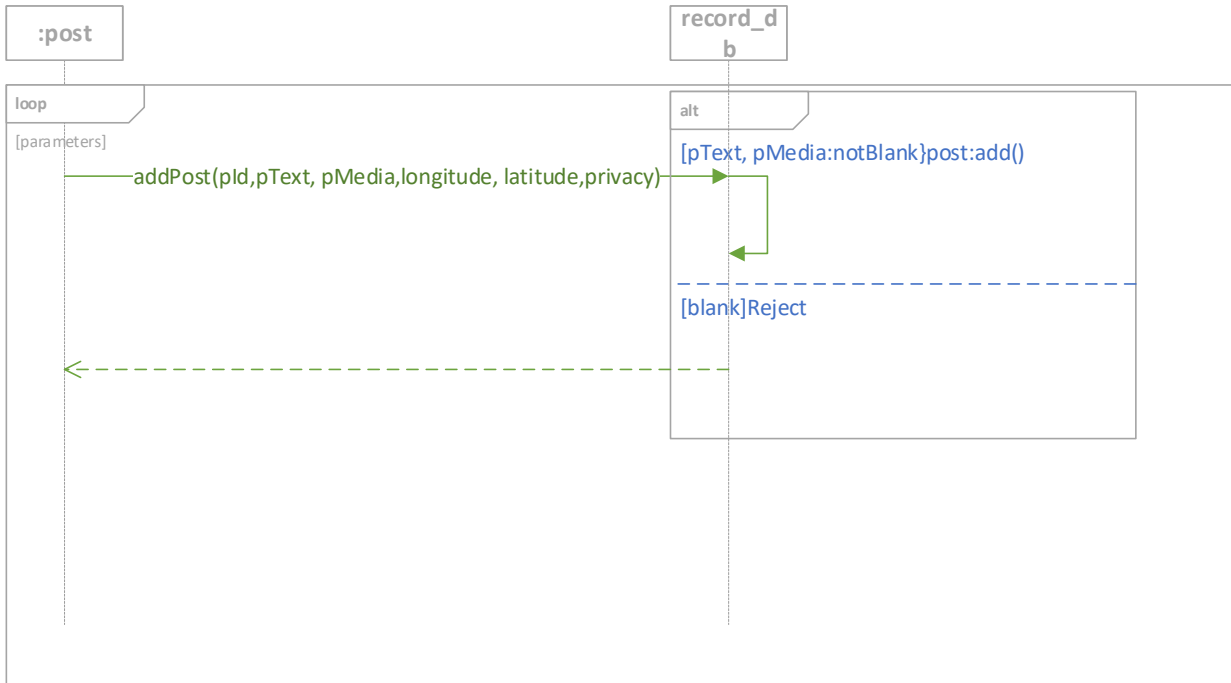
### 3. LOGOUT



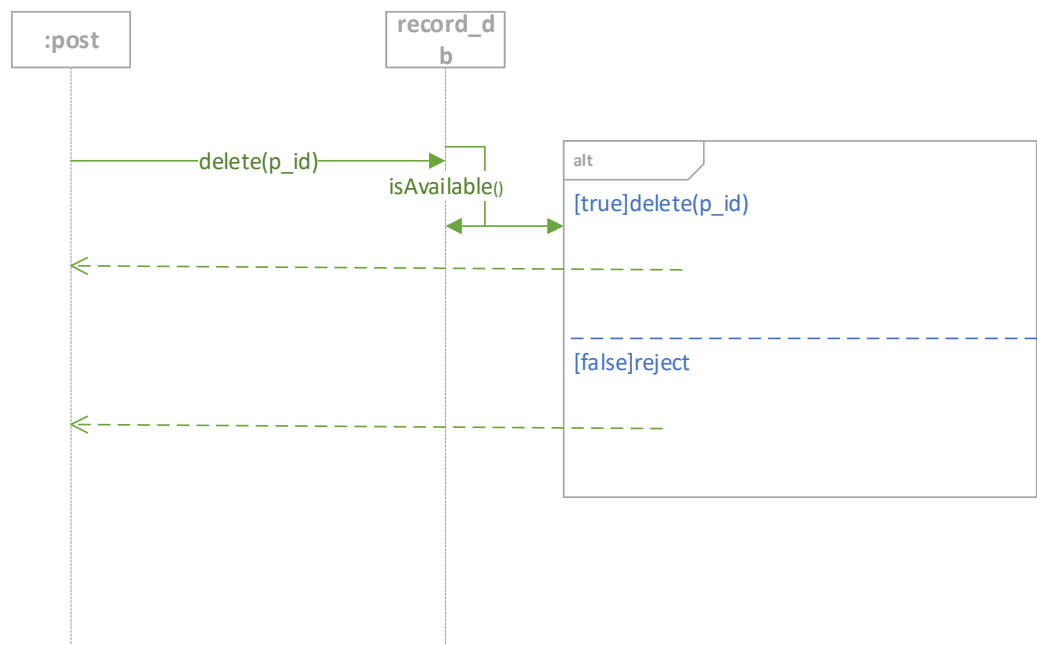
#### 4. EDIT PROFILE



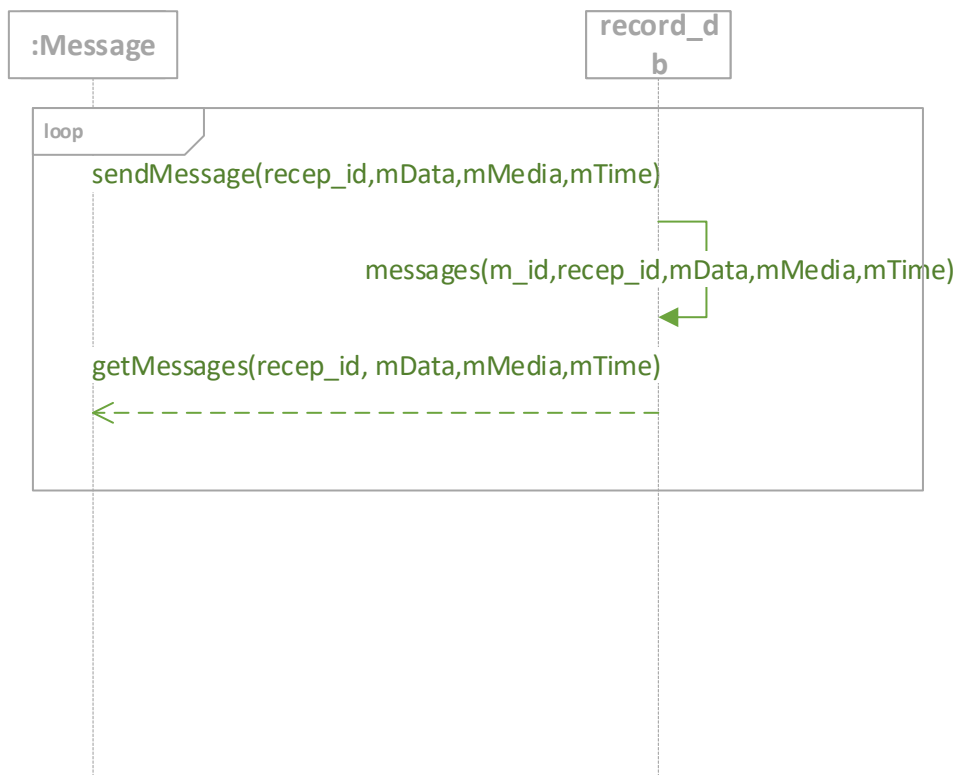
### 5. ADD POST



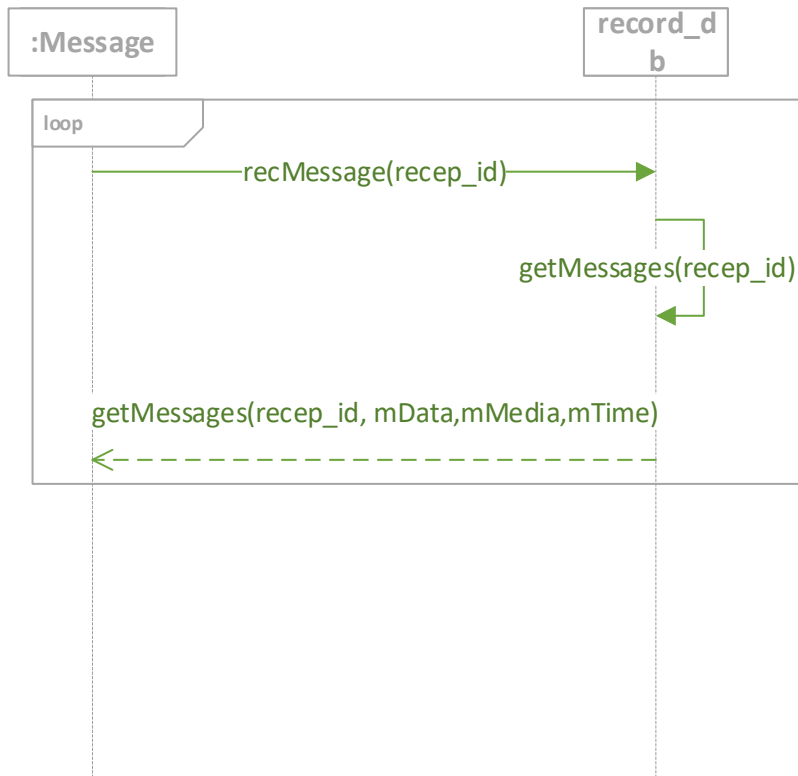
### 7.Delete Post



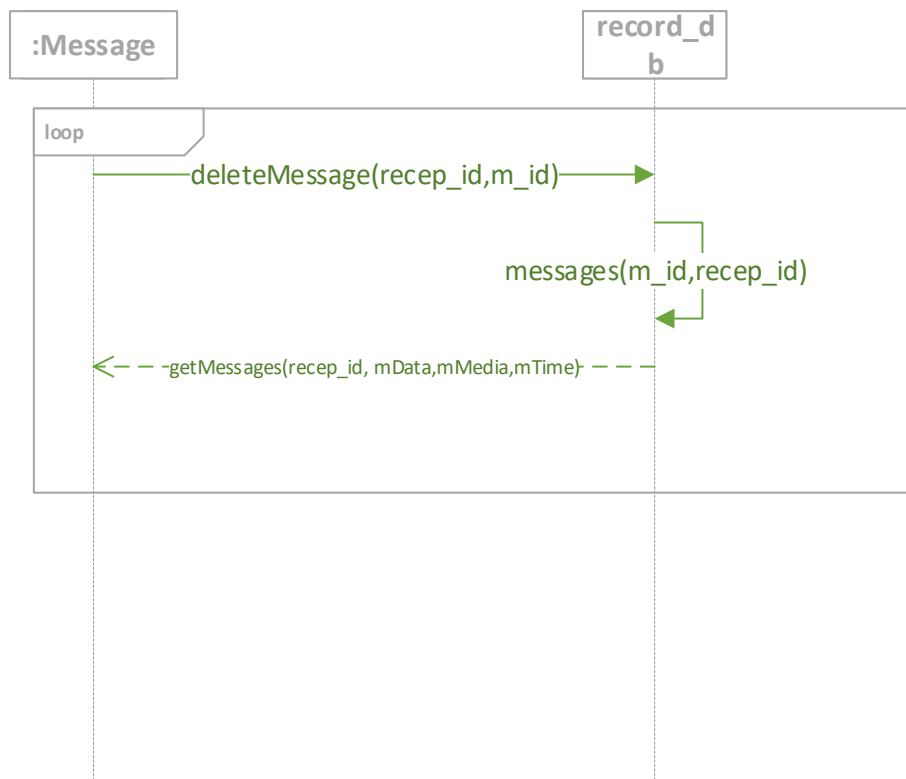
## 8. Send Message



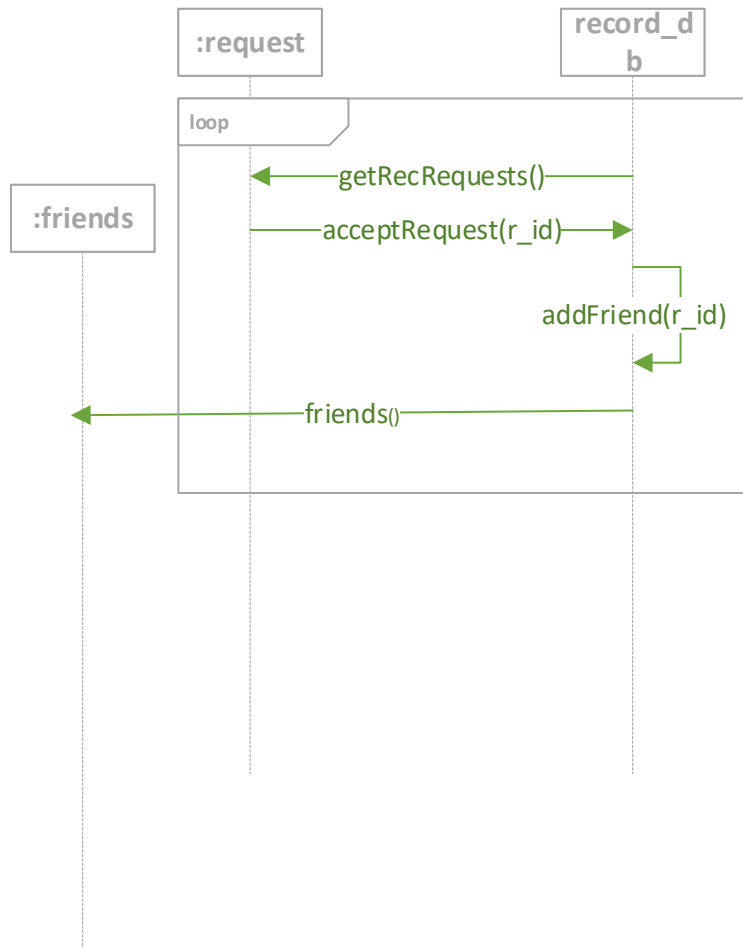
## 9. Recieve Messages



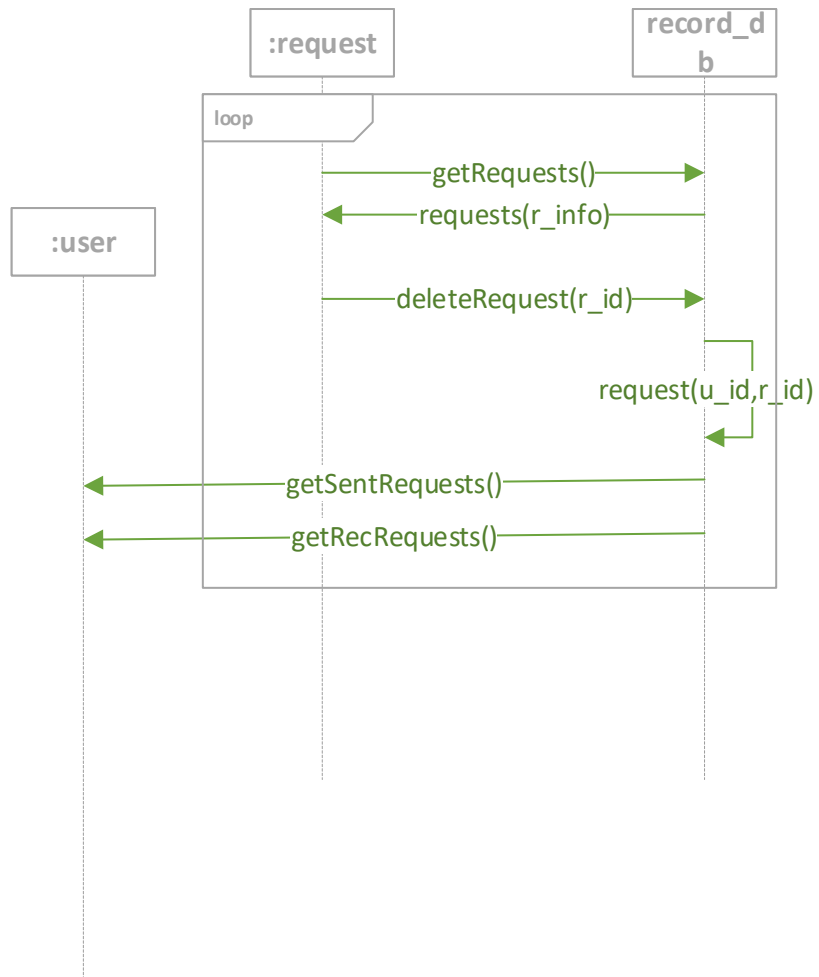
## 10. Delete Message



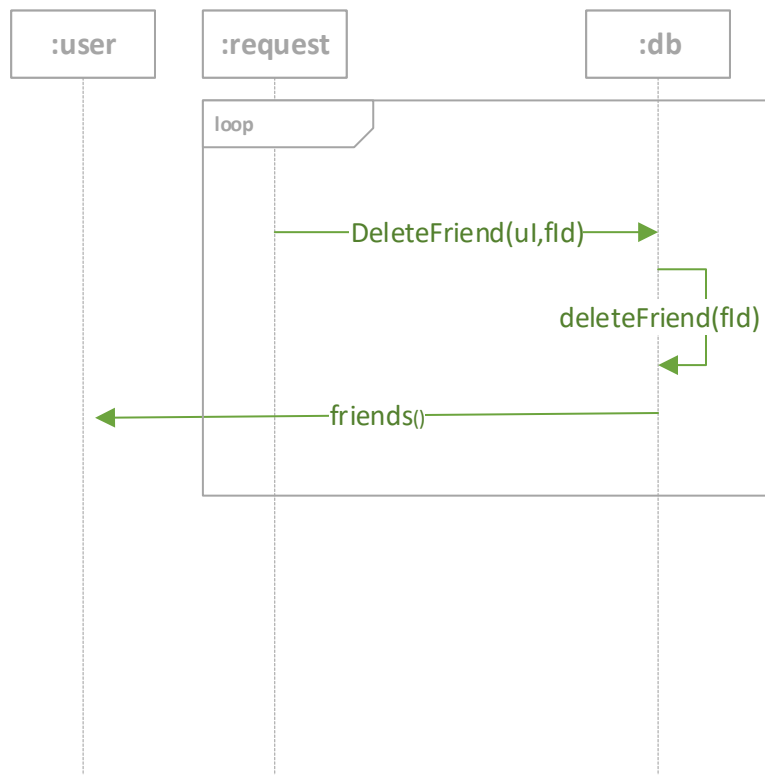
## 12. Accept Request



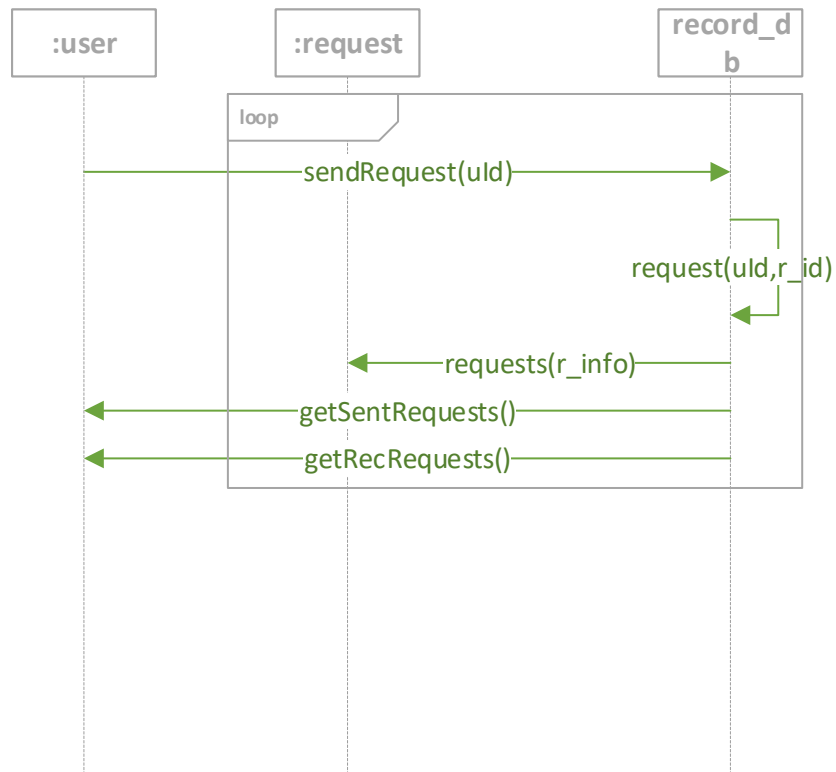
## 13.Delete Reuquest



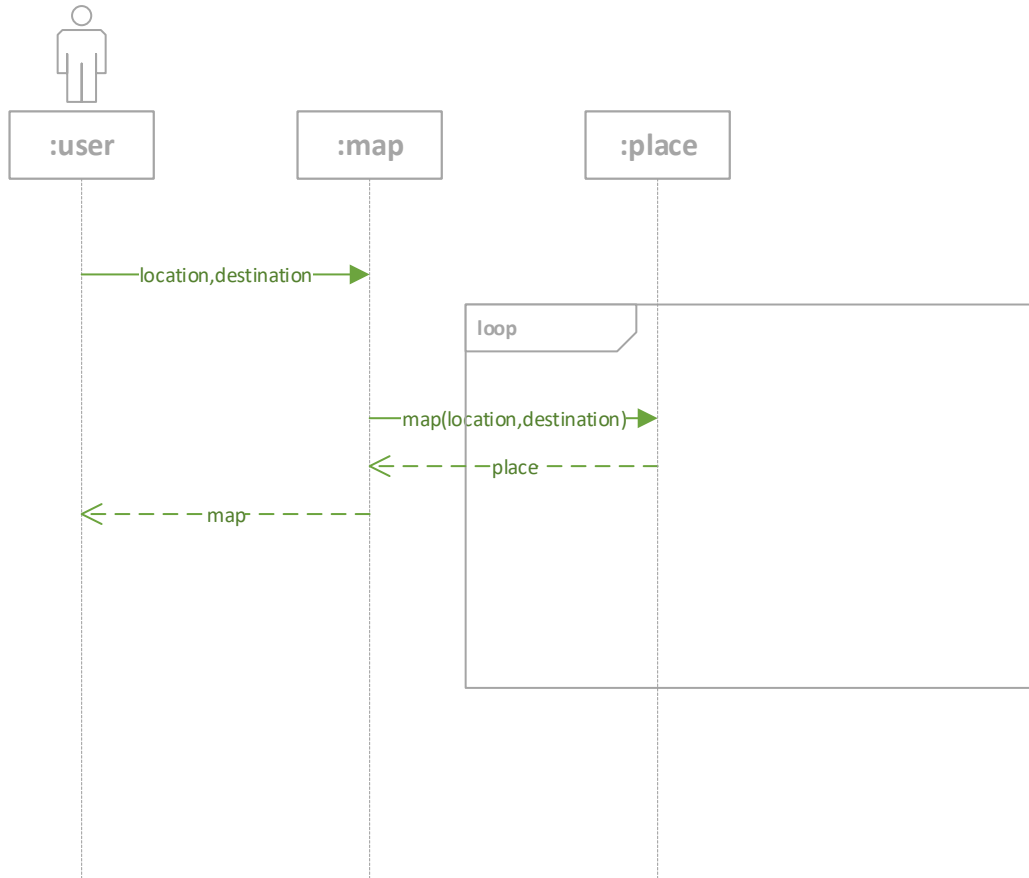
## 14. Delete Friend



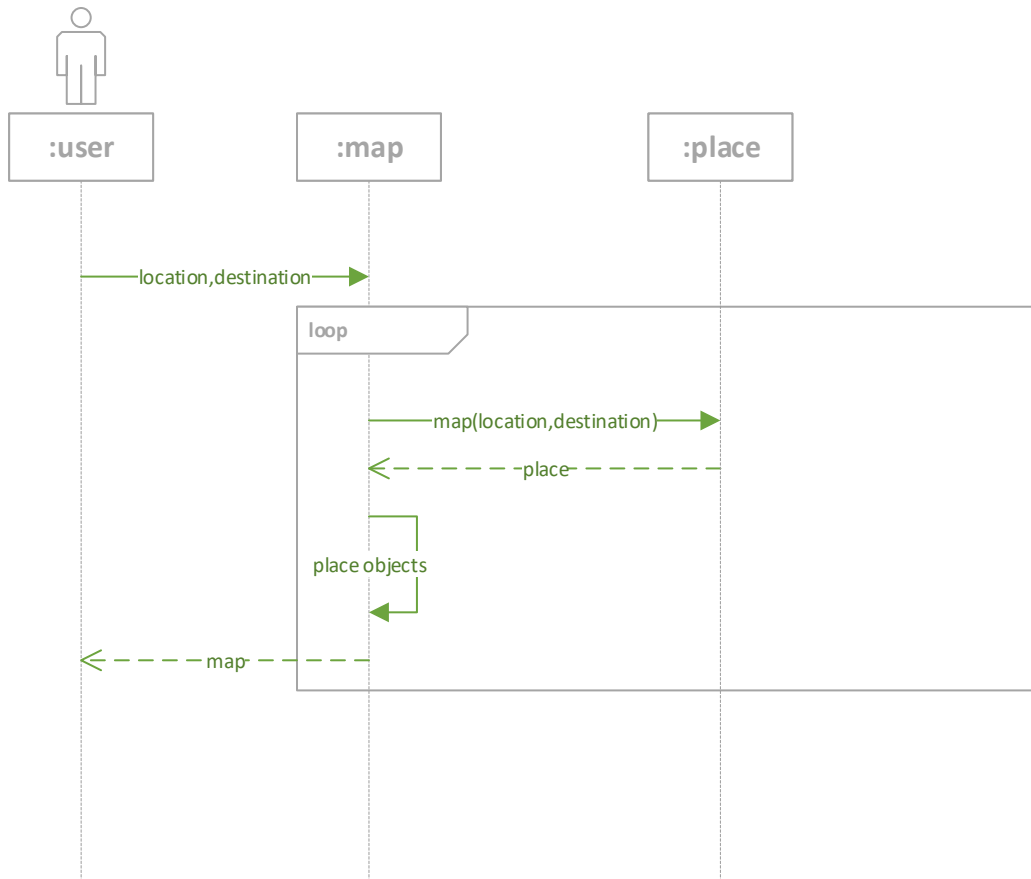
## 11. Send Request



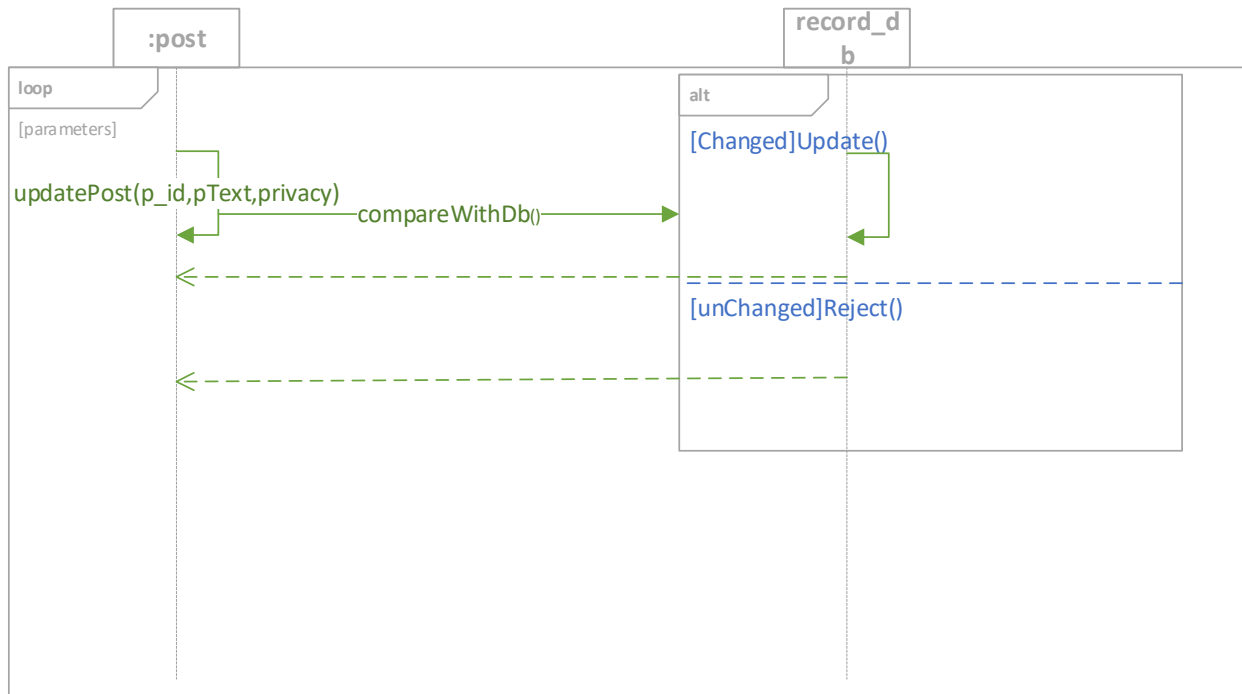
## 16. Map



## 17. Augmented Map



## 6.Update Post



## 4.6. Operation contracts

**Operation name:** Registration (name, email, password, phone, gender, DOB)

**Cross Reference:** Use case Registration

**Pre-condition:** User should be online.

**Post-condition:**

- An instance of user is created with having attributes (uld, name, password, email, phone, address, DOB, gender)
- User is associated with profile
- User is associated with admin

**Operation name:** Login (email, password)

**Cross Reference:** Use case Login

**Pre-condition:** User should be registered

**Post-condition:**

- An instance containing (email, password ) is create
- Login is associated with user profile

**Operation name:** Logout (uld)

**Cross Reference:** Use case Logout

**Pre-condition:** User should be logged in and online

**Post-condition:**

- User associates with session out and gets logout

**Operation name:** Add\_Post (pText, pMedia,privacy, time)

**Cross Reference:** Use case Add Post

**Pre-condition:** User should be logged in

**Post-condition:**

- Instance Post is created with attributes (pId, pText, pMedia, privacy, time)
- Post is associated with user to have user info and with discussion for comments and likes on post

**Operation name:** Delete\_Post (pId)

**Cross Reference:** Use case Delete Post

**Pre-condition:** There should be a post to be deleted

**Post-condition:**

- An instance of Post is created with (pId)
- Post is associated with db and gets deleted

**Operation name:** Update\_Post (pId, pText, pMedia)

**Cross Reference:** Use case Update Post

**Pre-condition:** There should be a post to update

**Post-condition:**

- An Update\_Post instance is created
- Gets associated with Post
- Post gets updated

**Operation name:** Send\_Messages (recp\_id, data, media, date\_time)

**Cross Reference:** Send Messages

**Pre-condition:** Must be logged in and online

**Post-condition:**

- Send\_Messages instantiate Messages with attributes ( mld, recpld, mData, mMedia, time)
- Messages is associated with user

**Operation name:** Receive\_Messages (recpld,mData, mMedia, time)

**Cross Reference:** Use case Receive Messages

**Pre-condition:** Must be logged in and online

**Post-condition:**

- Receive\_Messages instance is created with (recpld)
- Associate with messages
- Messages get received and aligned to time

**Operation name:** Delete\_Messages (mld, recpld)

**Cross Reference:** Use case Delete Messages

**Pre-condition:** Must be logged in and online

**Post-condition:**

- Delete\_Messages instance is created with (mld, recpld)
- Gets associated with Messages
- Message gets deleted

**Operation name:** View\_Messages (recpld)

**Cross Reference:** View Messages use case

**Pre-condition:** Must be logged in and online

**Post-condition:**

- View\_Messages instance is created with attribute (recpld)
- Associated with Messages

**Operation name:** Update\_Profile (name, email, phone, gender, bio, DOB,cover,pic)

**Cross Reference:** Update Profile use case

**Pre-condition:** Must be registered and logged in

**Post-condition:**

- Update\_Profile instance with attributes(name, email, phone, gender, bio, Dob, cover, pic)
- Associates with Profile

**Operation name:** Send\_Request (uld,reqId)**Cross Reference:** Send request use case**Pre-condition:**

- Must be registered and logged in
- To whom request is being sent, should not be friend

**Post-condition:**

- Sent\_Requests object instantiate with attributes (rec\_id)
- Associates with Requests

**Operation name:** accept\_request ()**Cross Reference:** Accept request use case**Pre-condition:** Must be registered and logged in**Post-condition:**

- Accept\_request operation instantiates Friend with attributes (f\_id)
- Associate with Friends

**Operation name:** Delete\_request (reqId)**Cross Reference:** Delete request use case**Pre-condition:** There should be a request to be deleted**Post-condition:**

- Delete\_request instantiates with attribute(reqId)
- Associates with Requests and deletes the request

**Operation name:** delete\_friend (fld)**Cross Reference:** Delete friend use case**Pre-condition:** There should be a friend to be deleted**Post-condition:**

- Delete\_friend object is created with attribute(fld)
- Associate with Friend
- Friend gets deleted

**Operation name:** Map (location, destination)

**Cross Reference:** extended info map

**Pre-condition:** location and destination should be provided

**Post-condition:**

- Map(location, destination) object instantiate
- Map should show places with all the current updates

**Operation name:** Augmented\_Map (location, destination)

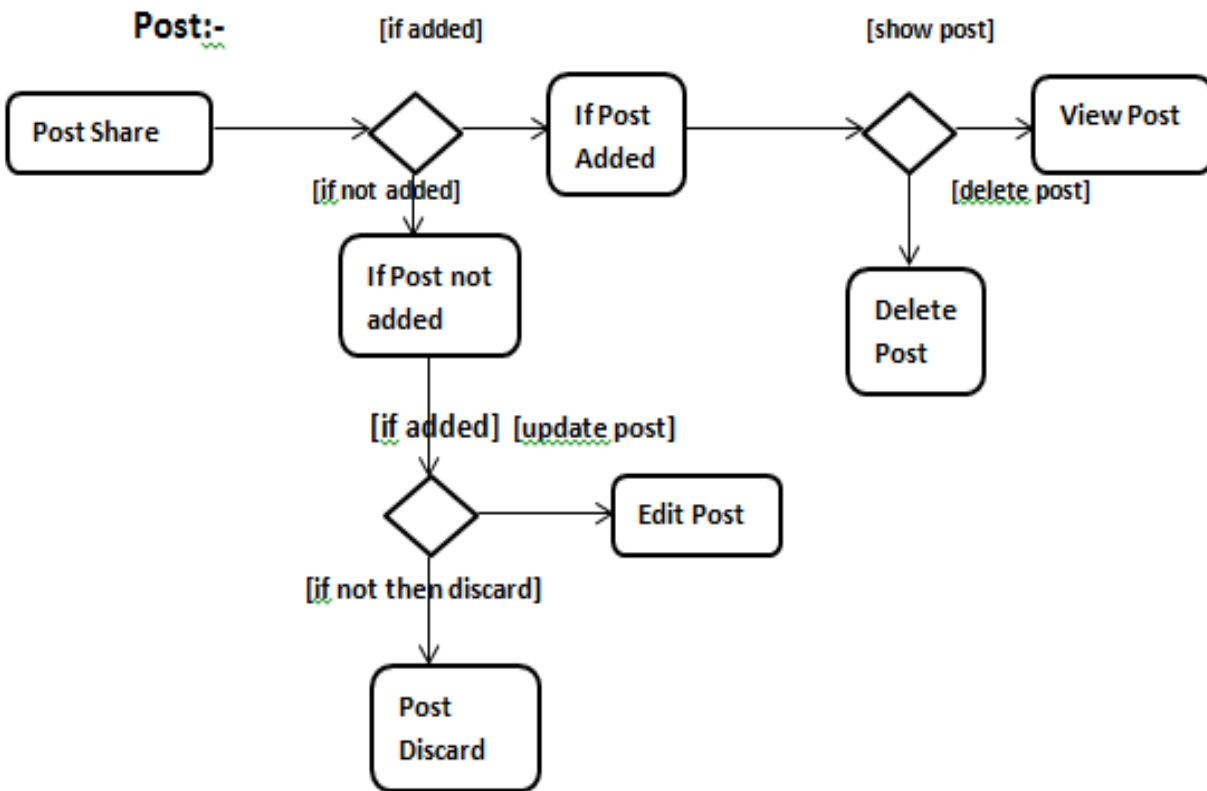
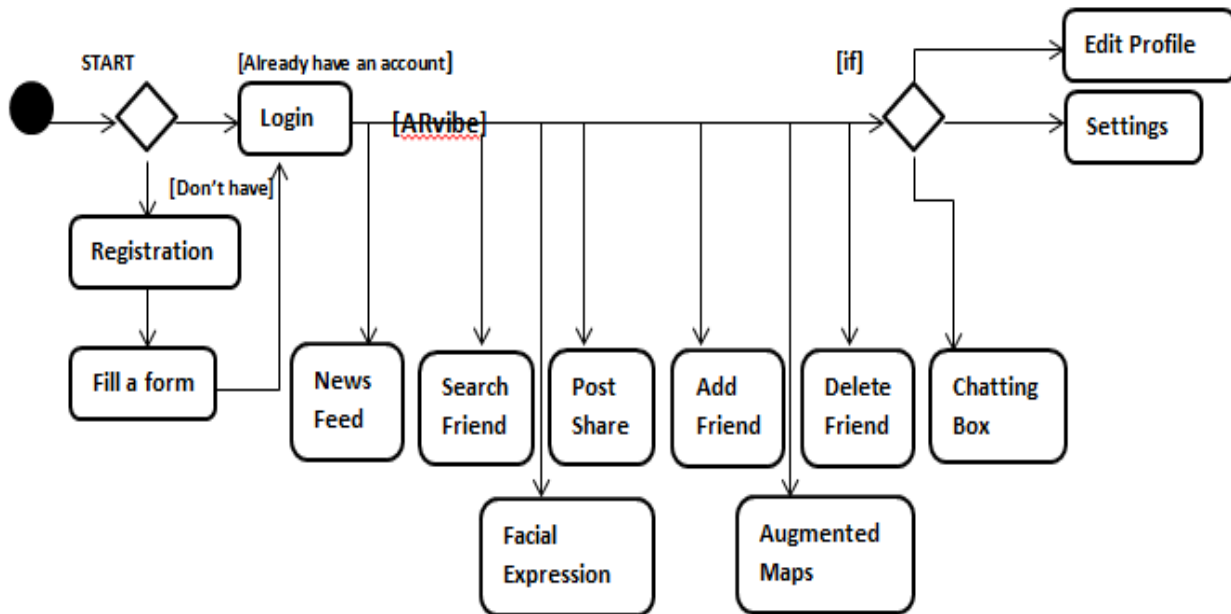
**Cross Reference:** augmented map

**Pre-condition:** location and destination should be provided

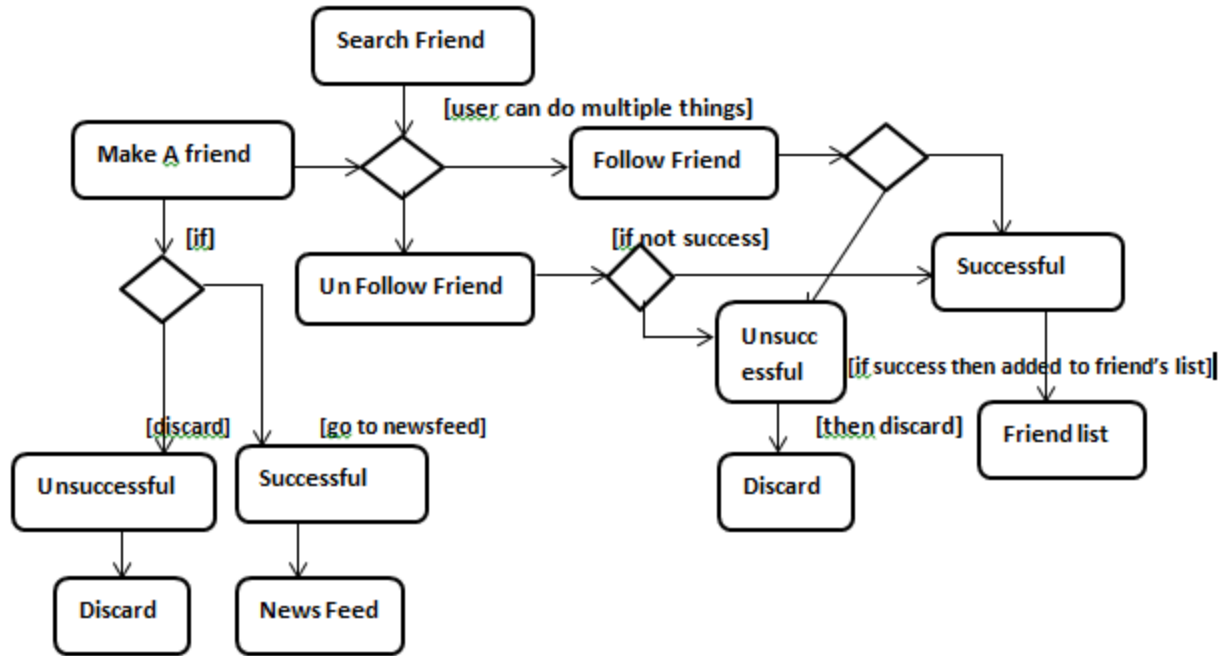
**Post-condition:**

- Augmented\_Map and Map objects with attributes (location, destination) gets created
- Map should show places augmentedly thrown objects on real location, seen by camera

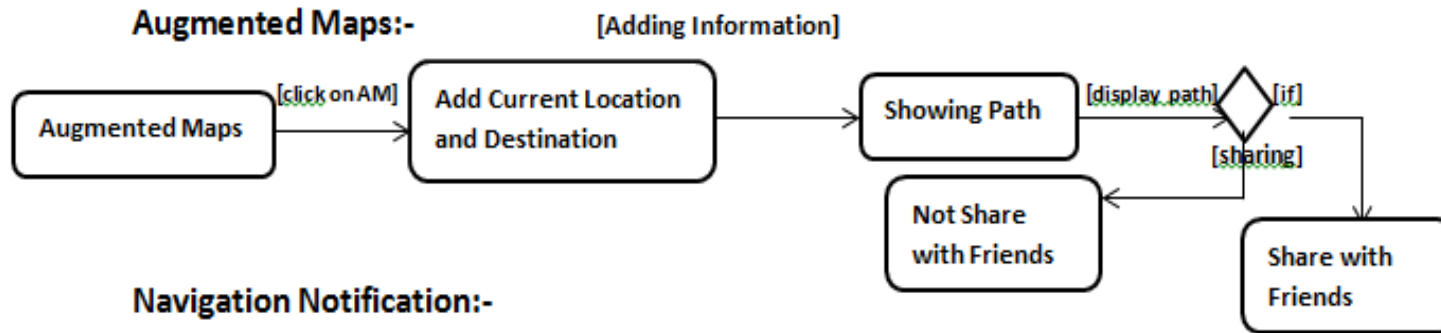
### 4.7. Activity Diagram



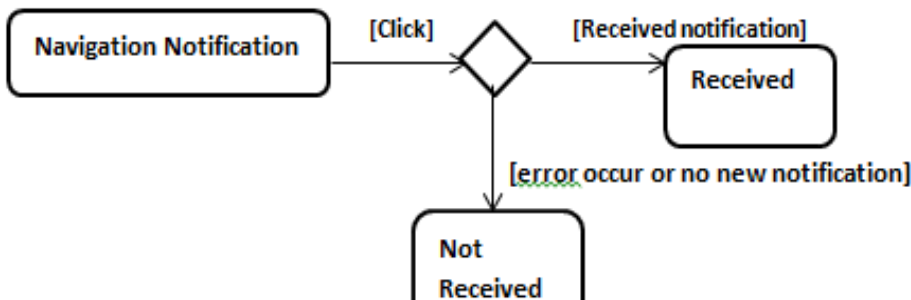
**Search Friend:-**



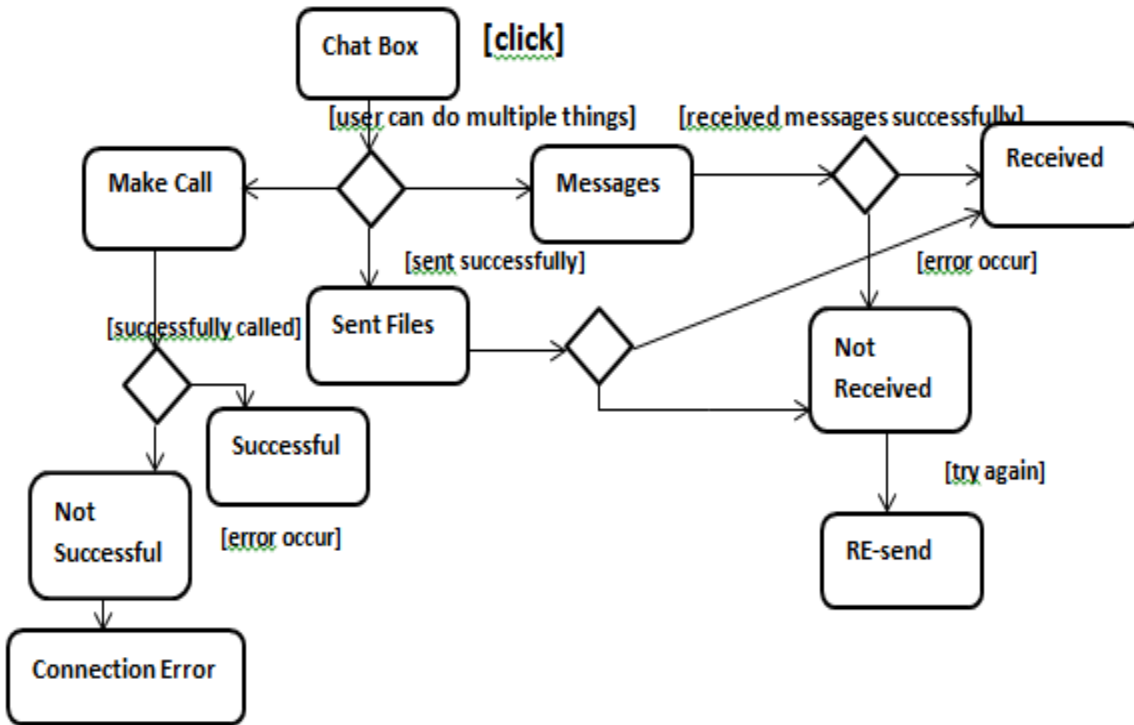
**Augmented Maps:-**



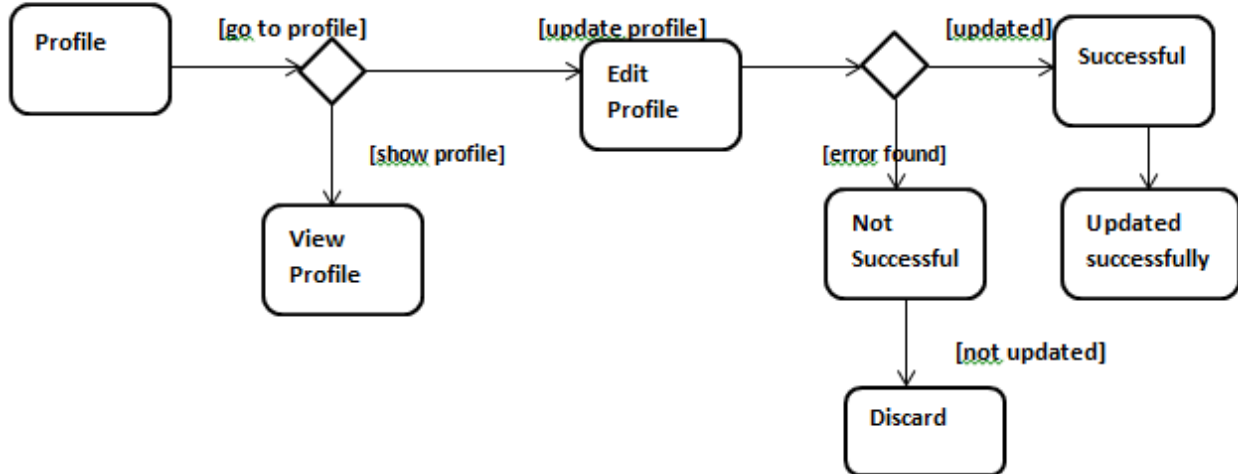
**Navigation Notification:-**



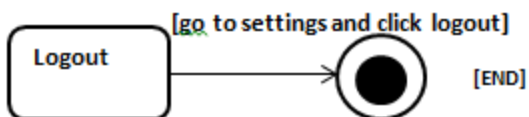
**Chat box:-**



**Edit Profile:-**

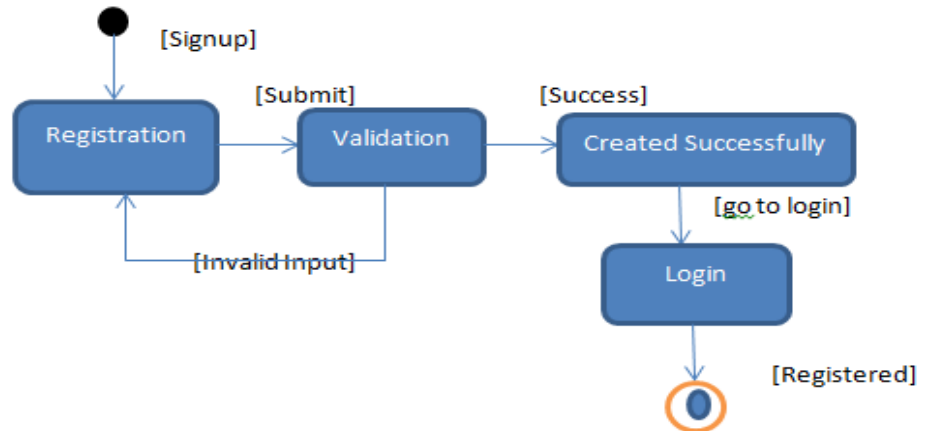


**Logout:-**

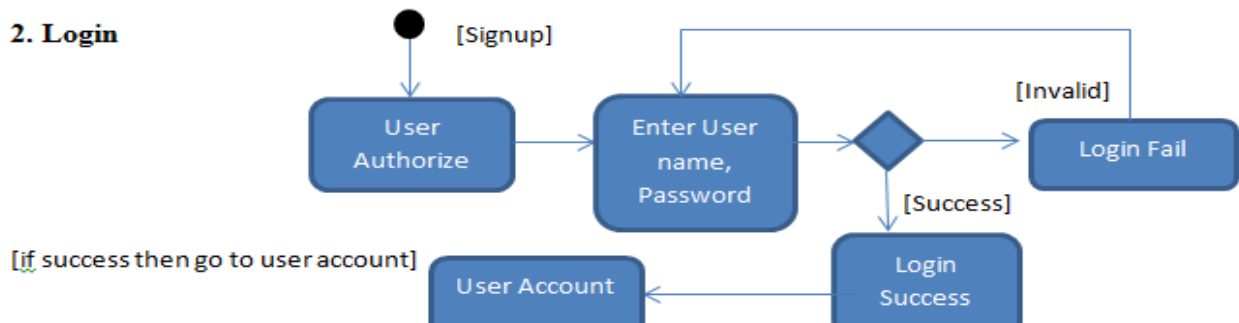


## 4.8. State Transition Diagram:

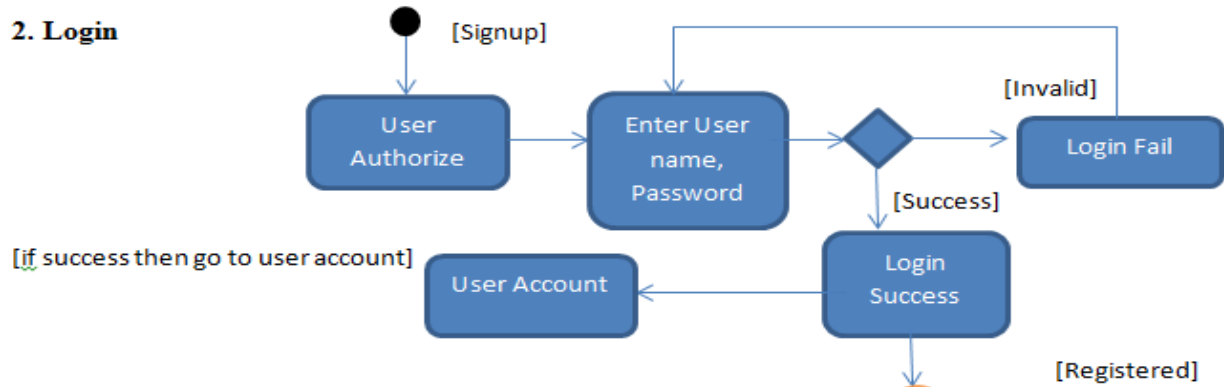
### 1. Registration



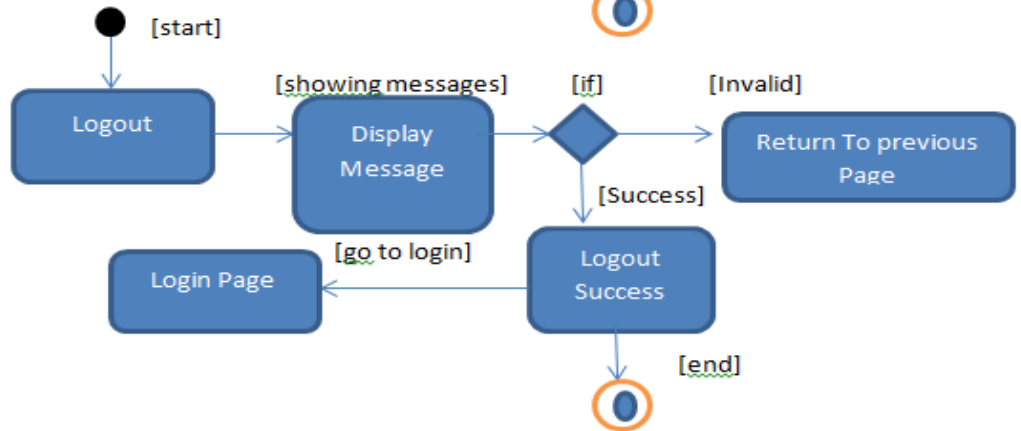
### 2. Login



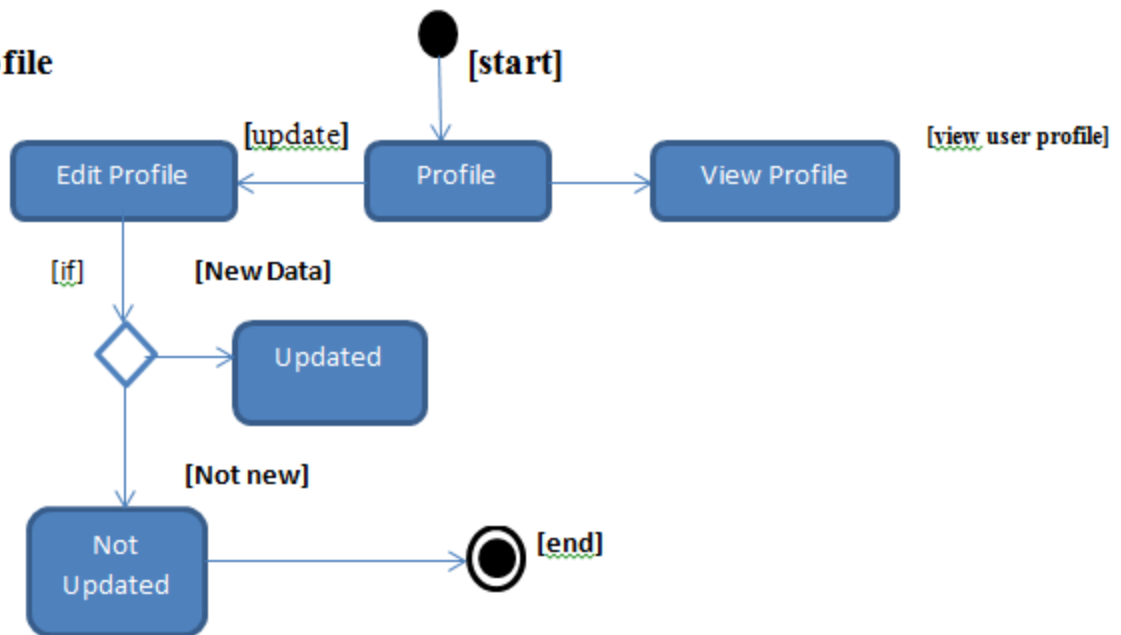
**2. Login**



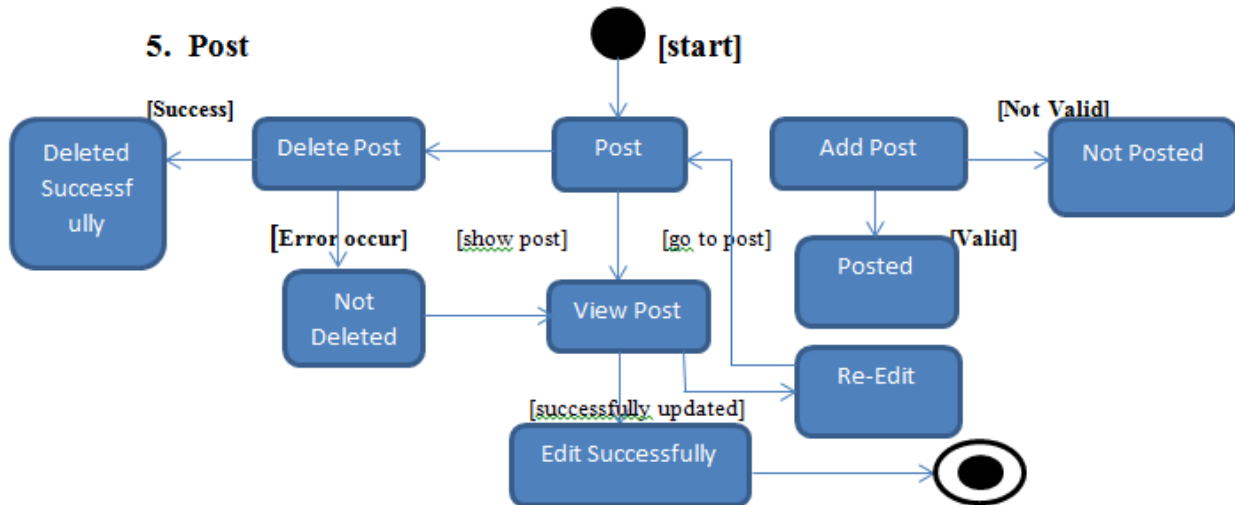
**3. Log Out**



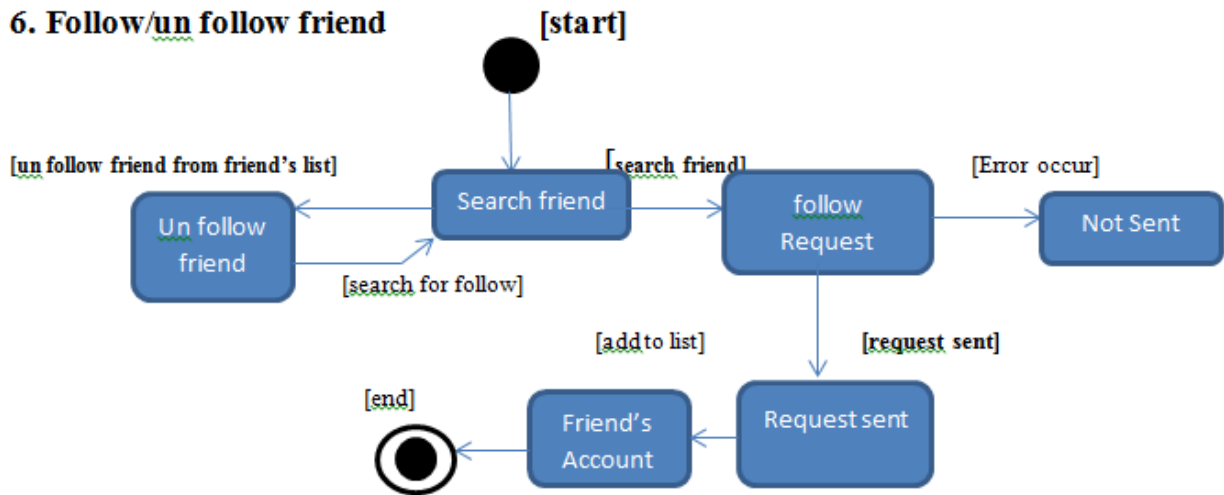
**4. Profile**



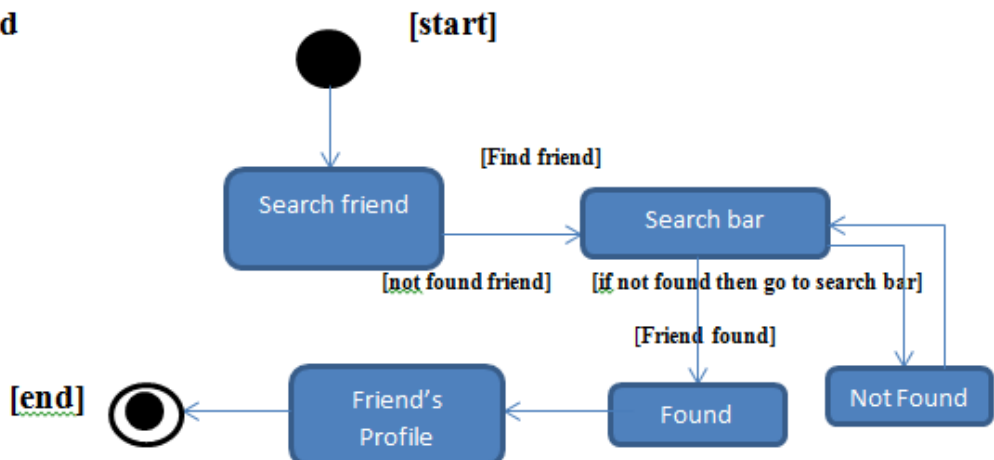
### 5. Post



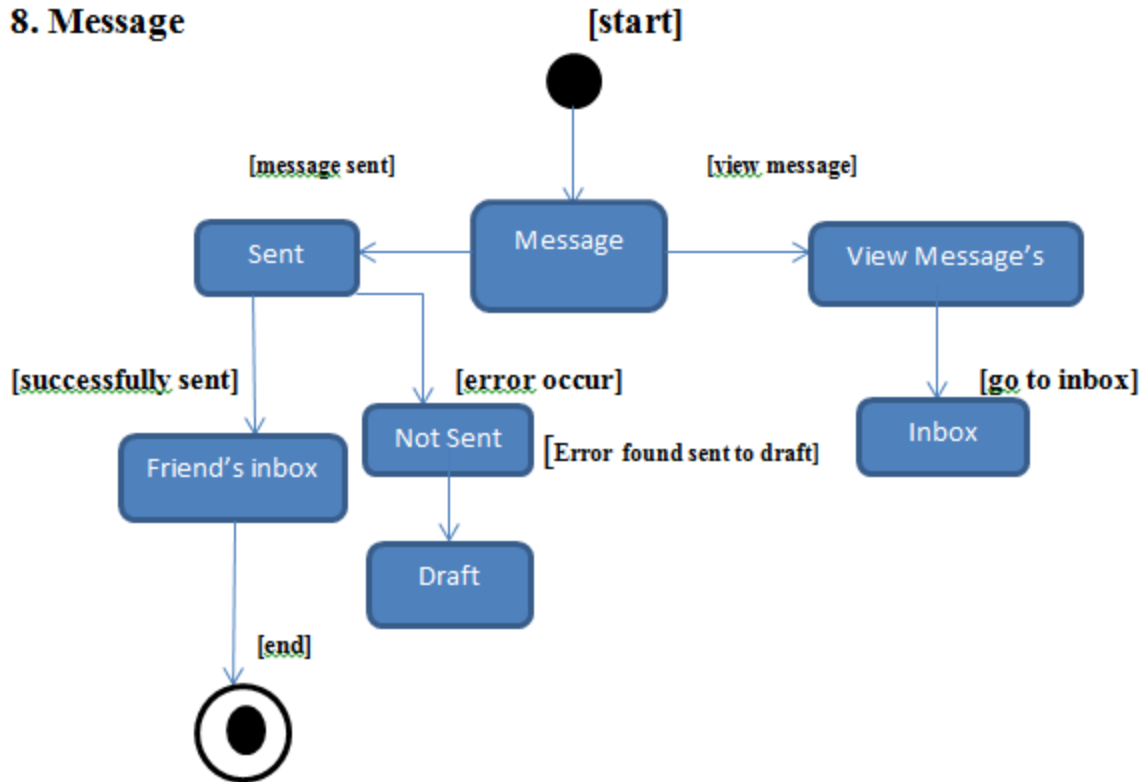
### 6. Follow/un follow friend



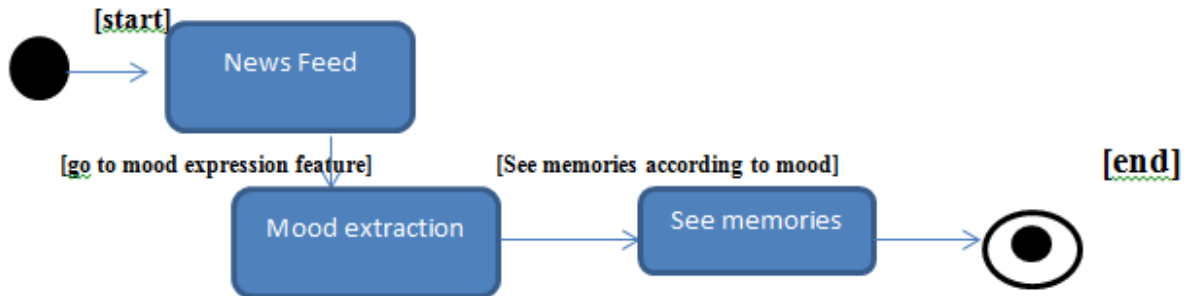
### 7. Search Friend



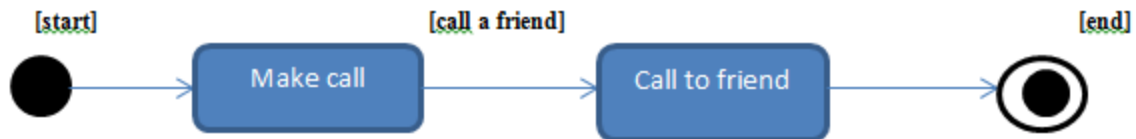
## 8. Message



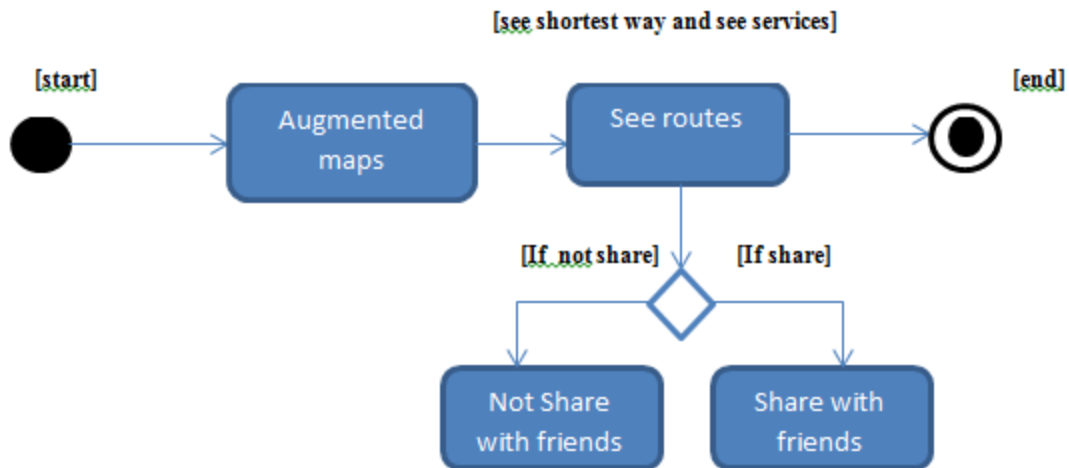
## 9. Facial Expression



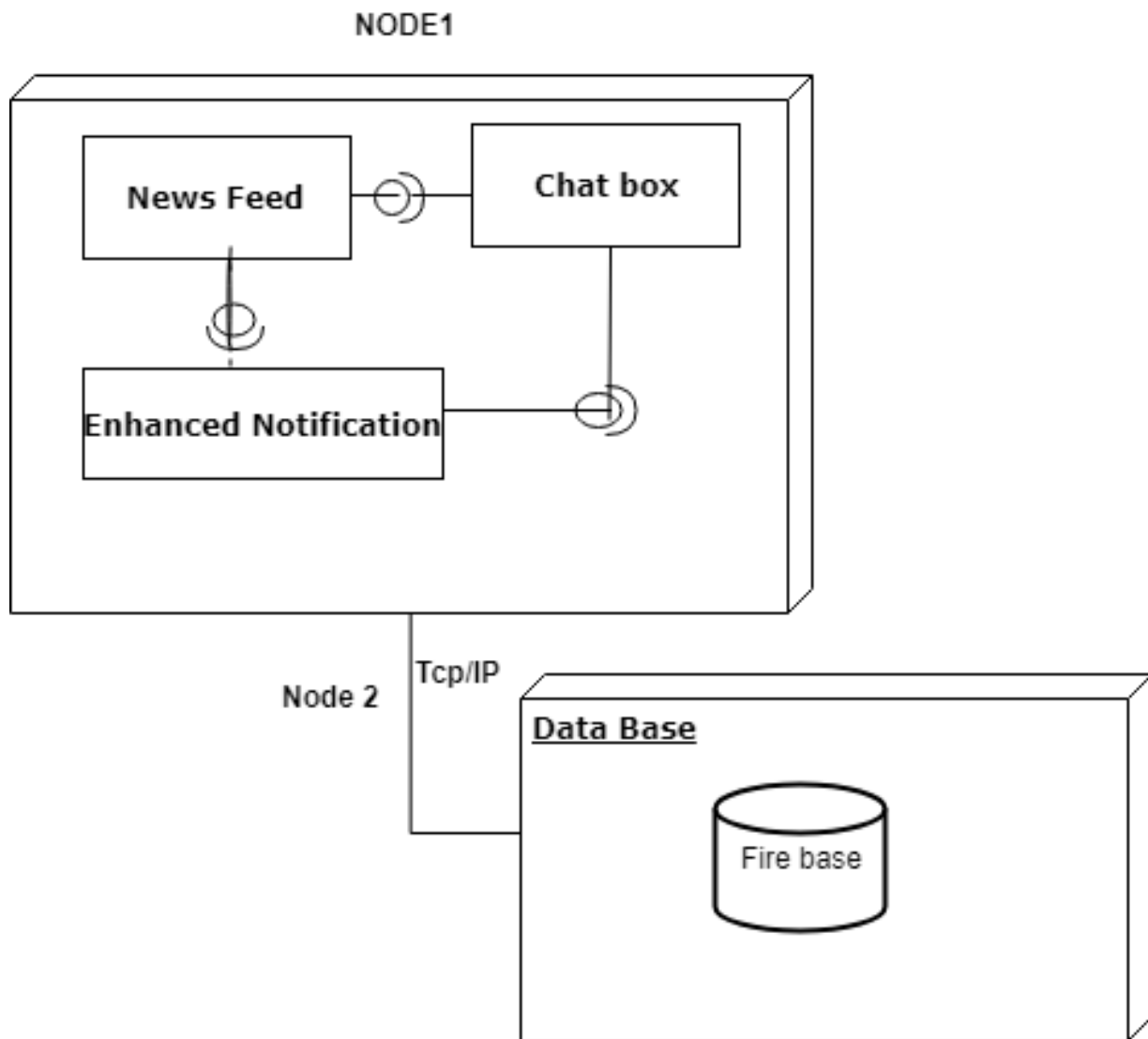
## 10. Make call



## 11. Augmented maps



#### 4.9. Component and diagram:-



# Chapter 5

## Implementation

## Chapter 5: Implementation

In this chapter, Some important flows of app, its components, Libraries used, tools and techniques being followed, coding standards and version control decisions are described which will help see application implementation details.

### 5.1. Important Flow Control/Pseudo codes

As ARvibe is mainly focusing on

#### 1. Post Saving

1. User makes a post
2. User location is detected
3. Post mood is classified using content classification and picture objects and types
4. Post is saved and showed up in news feed according to functionality defined

#### 2. Augmented Maps

1. User selects current location and destination
2. A path is extracted from Google maps
3. Cam gets open
4. Points are placed

#### 3. Enhanced Map

1. User adds up destination
2. If wants, can add further filters on information
3. Directions appear with offers and discounts
4. User can share details and path through.

## 5.2. Components and Libraries

### Components

#### Database

Firebase is use to share real-time data synchronization.

#### Android Studio

Android studio is used for development of application including application design.

### Libraries

```
com.google.maps.android:android-maps-utils:0.5+
```

```
implementation 'com.google.firebase:firebase-auth:17.0.0'  
implementation 'com.google.firebase:firebase-database:17.0.0'  
implementation 'com.google.firebase:firebase-core:16.0.1'  
implementation 'com.google.firebase:firebase-storage:17.0.0'  
implementation 'com.firebaseui:firebase-ui-database:5.0.0'  
implementation 'com.alexzh:circleimageview:1.1.2@aar'  
implementation 'com.github.bumptech.glide:glide:3.7.0'  
implementation project(':firebase')  
implementation project(':chat')  
implementation project(':blink')  
api 'com.theartofdev.edmodo:android-image-cropper:2.7.0'
```

## 5.3. Tools and Techniques

### Tools:

Android studio

### Techniques:

- Google maps
- Google Api's
- Firebase
- Java
- Xml
- Human computer interaction standards

## 5.4. Best Practices / Coding Standards

We will follow the naming conventions. It helps people to understand code better. We will avoid unnecessary objects. Avoiding memory leaks by simple tricks or avoiding Deadlocks in Java. For each public class and method (maybe setters and getters could be omitted). Avoid irrelevant and obvious comments. Explain the class purpose and its behavior. Always Predefine the Scope.

**Packages:** Names should be in **lowercase**.

**Classes:** Names should be in **Camel Case**. We will try to use nouns because a class is normally representing something in the real world.

**Interfaces:** Names should be in **Camel Case**. They tend to have a name that describes an operation that a class can do.

**Variables:** Names should be in **mixed case**.

**Constants:** Names should be in **uppercase**.

## 5.5. Version Control

### v1.0

- Currently, “Posts mood (content posted)” is content based [posts mood is detected through the picture and text it contain], while further in next version, we’re about to implement it more cognitively intelligent.
- And in Augmented Maps currently we are showing random objects, in next version user will be able to collect coins and will get services or discounts.
- In enhanced maps currently we are covering data entered by providing outlets with services/ discounts entrance form, in future data would be collected automatically from websites.

# Chapter 6

## Testing and Evaluation

## Chapter 6: Testing and Evaluation

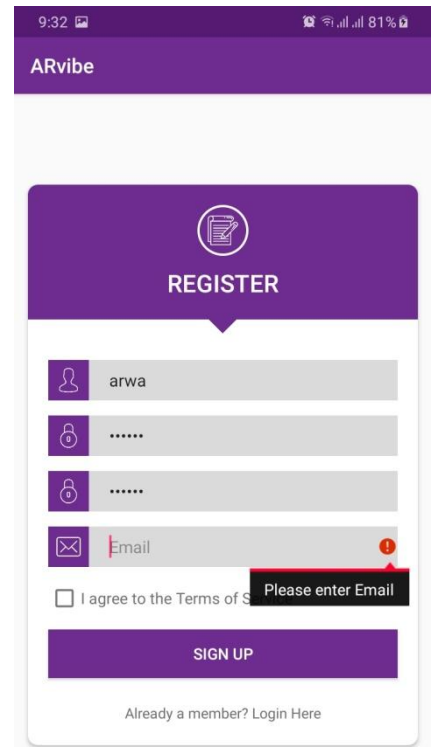
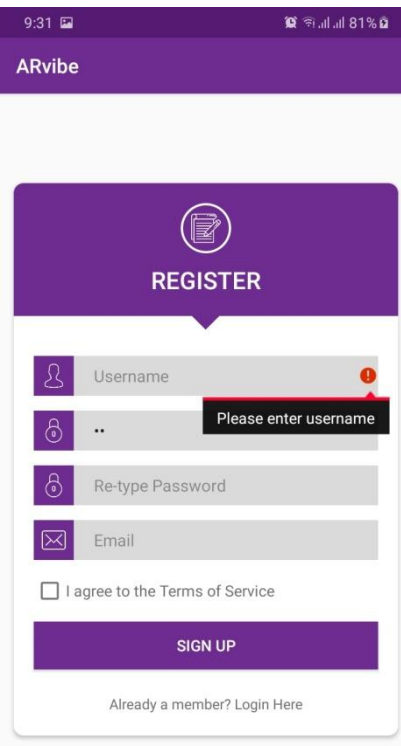
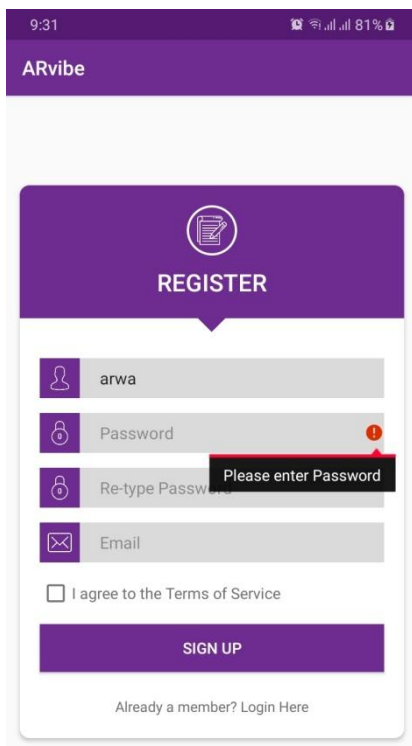
This chapter is all about testing of ArVibe to ensure the application quality. In this chapter we will describe the motive of finding errors in software. Testing will help us to enhance the quality of our application while reducing costs and saving development time. We will also describe the types of testing, equivalence partitioning and boundary value analysis.

### 6.1. Use Case Testing

**Register:-**

Test Suite ID	TS001
Test Case ID	TC001
Test Case Summary	To verify that by clicking Register button username, password, email will be store in firebase.
Related Requirement	RSoo1: User can Register/make an account on ArVibe.
Prerequisites	No
Test Procedure	<ol style="list-style-type: none"> <li>1. Select fields in Register form.</li> <li>2. Enter user data in fields.</li> <li>3. Click Register button.</li> </ol>
Test Data	Valid username: Saara, SAARA Invalid username: 12 saara ./saara ,saara %^3 Valid password: 123Abc@5, Abc567\$%9 Invalid password: 1_2saara, _saara12 Valid email: ali@gmail.com, 12ali@yahoo.com Invalid email: ali.com, iqra@yahoo Valid phone# :03123456578 Invalid phone: @56rfgf7999, 2wstyyA
Expected Result	<ol style="list-style-type: none"> <li>1. If username, password and email are valid then store user data by clicking register button.</li> <li>2. If given inputs are invalid then display error message.</li> <li>3. If fields are empty then show warning message.</li> </ol>

Actual Result	<ol style="list-style-type: none"> <li>1. If name is valid, the result is as expected.</li> <li>2. If name is not valid then invalid message displayed.</li> <li>3. If fields are empty then warning message displayed.</li> </ol>
Status	Pass
Remarks	This test case is simple and easy.
Created By	Sara Shahid
Date of Creation	04/01/19
Executed By	Sara Shahid
Date of Execution	04/01/19
Text Environment	OS: Android Studio Version 3.1



9:32 81%

ARvibe

**REGISTER**

arwa

.....

.....

arwa.riaz786@gmail.com

I agree to the Terms of Service

**SIGN UP**

Already a member? [Login Here](#)

**Login:-**

Test Suite ID	TS001
Test Case ID	TC002
Test Case Summary	To verify Authentication or Login.
Related Requirement	RS002: User should able to Login.
Prerequisites	User should Registered first.
Test Procedure	<ol style="list-style-type: none"> <li>1. Select Username field and enter username.</li> <li>2. Select Password field and enter password.</li> <li>3. Click Login button.</li> </ol>
Test Data	Valid username: Sara,sara Invalid username: 1sara, ./sara , sara%^3 Valid password: 123Aabc@, abMc567\$% Invalid password: 1_2sara, _sara12
Expected Result	<ol style="list-style-type: none"> <li>1. If username and password are valid then login successfully.</li> <li>2. If username and password are invalid message display.</li> </ol>

	3. If fields are empty then display warning message.
Actual Result	1. If fields are valid, the result is as expected. 2. If fields are not valid then invalid message displayed. 3. If fields are empty then warning message displayed.
Status	Pass
Remarks	This test case is simple and easy.
Created By	Sara Shahid
Date of Creation	04/01/19
Executed By	Sara Shahid
Date of Execution	04/01/19
Text Environment	OS: Android Studio Version 3.1

**Logout:-**

Test Suite ID	TS001
Test Case ID	TC003
Test Case Summary	To verify user is logged out correctly after clicked on logout button.
Related Requirement	RS003: the user can logout their account.
Prerequisites	The user should be login.
Test Procedure	User should go to settings and then click on logout button
Test Data	04/01/19
Expected Result	1. user profile will be logout when user clicked in logout button. 2. If user not logout then show connection error.
Actual Result	User profile will be logout when user clicked in logout button.

Status	Pass
Remarks	This test case is simple and easy.
Created By	Sara Shahid
Date of Creation	04/01/19
Executed By	Sara Shahid
Date of Execution	04/01/19
Text Environment	OS: Android Studio Version 3.1

**Update profile:-**

Test Suite ID	TS002
Test Case ID	TC004
Test Case Summary	To verify that user profile is updated or not in the firebase.
Related Requirement	RS004: User can update profile.
Prerequisites	User should be login.
Test Procedure	<ol style="list-style-type: none"> <li>1. Select fields.</li> <li>2. Edit fields.</li> <li>3. Click on save button.</li> </ol>
Expected Result	<ol style="list-style-type: none"> <li>1. Profile will be updated</li> <li>2. If Profile will not updated error message display</li> </ol>
Actual Result	<ol style="list-style-type: none"> <li>1. If action is valid, the result is as expected.</li> <li>2. Profile will be updated</li> </ol>
Status	Pass
Remarks	This test case is simple and easy.
Created By	Sara Shahid

Date of Creation	04/01/19
Executed By	Sara Shahid
Date of Execution	04/01/19
Text Environment	OS: Android Studio Version 3.1

**Send Files:-**

Test Suite ID	TS002
Test Case ID	TC005
Test Case Summary	To verify that files is send to destination by clicking send button.
Related Requirement	RS005: User should be able to send files.
Prerequisites	User should be login.
Test Procedure	<ol style="list-style-type: none"> <li>1. Open friend chat box.</li> <li>2. select attachments.</li> <li>3. Click on send button.</li> </ol>
Expected Result	<ol style="list-style-type: none"> <li>1. File will be send</li> <li>2. Show error message "large file"</li> </ol>
Actual Result	<ol style="list-style-type: none"> <li>1. File will be send.</li> <li>2. Show error message</li> </ol>
Status	Pass
Remarks	This test case is simple and easy.
Created By	Sara Shahid
Date of Creation	04/01/19
Executed By	Sara Shahid

Date of Execution	04/01/19
Text Environment	OS: Android Studio Version 3.1

**Add friend:-**

Test Suite ID	TS002
Test Case ID	TC006
Test Case Summary	To verify Friend is added in firebase.
Related Requirement	RS006: User can add friends.
Prerequisites	User should be login.
Test Procedure	1. Enter name and number in phone or in ArVibe. 2. Click on add/save button.
Expected Result	Friend will be added.
Actual Result	1. Friend will be added 2. Not added show message "connection error"
Status	Pass
Remarks	This test case is simple and easy.
Created By	Sara Shahid
Date of Creation	04/01/19
Executed By	Sara Shahid
Date of Execution	04/01/19
Text Environment	OS: Android Studio Version 3.1

**Block friend:-**

Test Suite ID	TS002
Test Case ID	TC007
Test Case Summary	To verify that friend is block.
Related Requirement	RS006: User should be able to block friend.
Prerequisites	User should be login.
Test Procedure	1. Search friend. 2. Click on block button.
Expected Result	1. Friend will be blocked. 2. If connection error then display error message.
Actual Result	1. If friend blocked successfully, the result is as expected. 2. If friend not blocked then error message displayed.
Status	Pass
Remarks	This test case is simple and easy.
Created By	Sara Shahid
Date of Creation	04/01/19
Executed By	Sara Shahid
Date of Execution	04/01/19
Text Environment	OS: Android Studio Version 3.1

**Search friend:-**

Test Suite ID	TS002
Test Case ID	TC008
Test Case Summary	To verify that friend is searched by clicking search button.
Related Requirement	RS005: User should be able to search friend.

Prerequisites	User should be login.
Test Procedure	<ol style="list-style-type: none"> <li>1. Select search field.</li> <li>2. Enter friend name in fields.</li> <li>3. Click search button.</li> </ol>
Expected Result	<ol style="list-style-type: none"> <li>1. If name is valid then show detail.</li> <li>2. If name is invalid then display error message.</li> <li>3. If field is empty then display warning message.</li> </ol>
Actual Result	<ol style="list-style-type: none"> <li>1. If name is valid, the result is as expected.</li> <li>2. If name is not valid then invalid message displayed.</li> <li>3. If fields are empty then warning message displayed.</li> </ol>
Status	Pass
Remarks	This test case is simple and easy.
Created By	Sara Shahid
Date of Creation	04/01/19
Executed By	Sara Shahid
Date of Execution	04/01/19
Text Environment	OS: Android Studio Version 3.1

**View Friends:-**

Test Suite ID	TS002
Test Case ID	TC009
Test Case Summary	To verify that friend can be view by clicking friend's button.
Related Requirement	RS005: User should be able to view friend.
Prerequisites	User should be login.
Test Procedure	<ol style="list-style-type: none"> <li>1. Select friends.</li> </ol>

	2. see friend's list
Expected Result	Friends will be view
Actual Result	Friends will be view
Status	Pass
Remarks	This test case is simple and easy.
Created By	Sara Shahid
Date of Creation	04/01/19
Executed By	Sara Shahid
Date of Execution	04/01/19
Text Environment	OS: Android Studio Version 3.1

**View Post:-**

Test Suite ID	TS002
Test Case ID	TC0010
Test Case Summary	To verify that Post can be view by clicking post button.
Related Requirement	RS005: User should be able to view posts.
Prerequisites	User should be login.
Test Procedure	1. Select post. 2. View post.
Expected Result	Post will be view
Actual Result	Post will be view
Status	Pass
Remarks	This test case is simple and easy.

Created By	Sara Shahid
Date of Creation	04/01/19
Executed By	Sara Shahid
Date of Execution	04/01/19
Text Environment	OS: Android Studio Version 3.1

**Un follow/delete friend:-**

Test Suite ID	TS002
Test Case ID	TC0011
Test Case Summary	To verify that friend is delete.
Related Requirement	RS006: User should be able to delete friend.
Prerequisites	User should be login.
Test Procedure	1. Search friend. 2. Click on delete button.
Expected Result	1. Friend will be delete. 2. If connection error then display error message.
Actual Result	1. If friend deleted successfully, the result is as expected. 2. If friend not deleted then error message displayed.
Status	Pass
Remarks	This test case is simple and easy.
Created By	Sara Shahid
Date of Creation	04/01/19
Executed By	Sara Shahid
Date of Execution	04/01/19
Text Environment	OS: Android Studio Version 3.1

**Augmented maps:-**

Test Suite ID	TS002
Test Case ID	TC0012
Test Case Summary	To check augmented maps working correctly when user click on augmented maps.
Related Requirement	RS006: User should be able to use augmented maps.
Prerequisites	User should be login.
Test Procedure	1. Go to feature option button then choose augmented maps 2. Click on augmented maps button.
Expected Result	1. Augmented maps will be open. 2. If connection error then display error message.
Actual Result	1. If augmented maps open successfully, the result is as expected. 2. If Augmented maps not open then error message displayed.
Status	Pass
Remarks	This test case is simple and easy.
Created By	Sara Shahid
Date of Creation	04/01/19
Executed By	Sara Shahid
Date of Execution	04/01/19
Text Environment	OS: Android Studio Version 3.1

**Enhanced maps:-**

Test Suite ID	TS002
Test Case ID	TC0013
Test Case Summary	To check enhanced maps working correctly when user click on enhanced maps.
Related Requirement	RS006: User should be able to use enhanced maps.
Prerequisites	User should be login.
Test Procedure	1. Go to feature option button then choose enhanced maps 2. Click on enhanced maps button.
Expected Result	1. Enhanced maps will be open. 2. If connection error then display error message.
Actual Result	1. If enhanced maps open successfully, the result is as expected. 2. If enhanced maps not open then error message displayed.
Status	Pass
Remarks	This test case is simple and easy.
Created By	Sara Shahid
Date of Creation	04/01/19
Executed By	Sara Shahid
Date of Execution	04/01/19
Text Environment	OS: Android Studio Version 3.1

**Send messages:-**

Test Suite ID	TS002
Test Case ID	TC0014
Test Case Summary	To verify message is sent or not to other user.

Related Requirement	RS006: User should be able to send messages.
Prerequisites	User should be login.
Test Procedure	1. Go to chat option button then choose friend from friends 2. Click on send button.
Expected Result	1. message will be sent. 2. If connection error then display error message.
Actual Result	1. If message sent successfully, the result is as expected. 2. If message not sent then error message displayed.
Status	Pass
Remarks	This test case is simple and easy.
Created By	Sara Shahid
Date of Creation	04/01/19
Executed By	Sara Shahid
Date of Execution	04/01/19
Text Environment	OS: Android Studio Version 3.1

**Post sharing:-**

Test Suite ID	TS002
Test Case ID	TC0015
Test Case Summary	To verify post is shared or not.
Related Requirement	RS006: User should be able to share post.
Prerequisites	User should be login.
Test Procedure	1. Go to post option button then choose/select data 2. Click on post button.

Expected Result	1. Post will be shared. 2. If connection error then display error message.
Actual Result	1. If post shared successfully, the result is as expected. 2. If post not shared then error message displayed.
Status	Pass
Remarks	This test case is simple and easy.
Created By	Sara Shahid
Date of Creation	04/01/19
Executed By	Sara Shahid
Date of Execution	04/01/19
Text Environment	OS: Android Studio Version 3.1

## 6.2 Equivalence partitioning

User name is alphabetic	<b>Valid</b>
User name is not alphabetic	<b>Invalid</b>
username should be in uppercase letter	<b>Invalid</b>
Password can be 2 letters	<b>Invalid</b>
password can be of 8 characters	<b>Valid</b>
Email without "@"	<b>Invalid</b>
Email without ".com"	<b>Invalid</b>
view post	<b>Valid</b>
view friends	<b>Valid</b>
Email with "@" and ".com"	<b>Valid</b>
Logout without login	<b>Invalid</b>
Login first time without registration	<b>Invalid</b>
Email can be start from special character	<b>Invalid</b>
User phone number can be 11 digits	<b>Valid</b>
Search friend by their name	<b>Valid</b>

## 6.2. Boundary value analysis

Sr.		Partition 1	Partition 2	Partition 3
1.	<b>Password</b>	Less than 8 character	1 – 8 character	9 – 12
2.	<b>Phone</b>	<=0	1 - 11	9 – 12

## 6.3. Data flow testing

We used Data flow testing to verify that ArVibe flow works well under a variety of conditions. For example to test ArVibe login flow, we first create a new test user account with ArVibe. Log into ArVibe with test user credentials then test user enters the email and password after registration if test user successfully login then login is working properly. If we test data flow of chat feature we go to message box after login test user then we will check the chat feature by sending messages to other user if message is sent then this feature is working well.

## 6.4. Unit testing

We test ArVibe smallest and testable parts individually and independently for proper testing of units. We do unit testing to verify that every unit of ArVibe are fit for use or not. We do unit testing for verification and validation. We test ArVibe units after and during development. We do unit testing to check functional correctness and completeness, for error handling, to check input values, to see correctness of output data and for optimizing and performance.

## 6.5. Performance testing

We do performance testing to ensure that ArVibe performed well under their expected workload. We do performance testing to check the speed of ArVibe. We verify the performance of every feature of Arvibe and all the feature working speedily and properly.

## 6.6. Stress Testing

We do stress Testing to verify the stability and reliability of the ArVibe and to see how ArVibe handle heavy load of users. We do stress testing to verify that ArVibe saved the data in firebase before some error or crashing application, to verify if unexpected failures do not cause security issues.

For example we have monitored our application behavior when maximum number of users logged in at same time , all users performing the critical operations at the same time and all the user accessing the same file at the same file we see that our database (firebase) server not down and our application is also not crashed.

# Chapter 7

## Summary, Conclusion and Future Enhancements

## Chapter 7: Summary, Conclusion & Future Enhancements

### 7.1. Project Summary

In summarized context, ARvibe is a social media android application which basically has 4 new functionalities.

- **Social Media with maps navigation**

As maps has become our daily and basic use during going to somewhere or to know about a location. Social media is also a must thing one do in mean time. So why not to merge both.

- **Augmented Maps**

Augmented Maps are an interacting and interesting way to use maps during following the maps to some location

- **Enhanced Maps**

Enhanced Maps are providing some additional services to user, when he/she wants to filter out additional and current services and discounts available at some specific place.

(Currently we are collecting data of outlets and service providers by ourselves)

- **Longitude Latitude wise Posting**

Posting in ARvibe has a new feature of saving the location of user when posting. So that when a user goes to somewhere he is able to see the newsfeed with priority of his/her current location.

All other functionalities of add friend, chatting is also included.

### 7.2. Achievements and Improvements

The major achievement is that we learned and applied many techniques and skills through this project. What we learned in four years of graduation has helped us alot in this and made us sure to do the learnings practically. We learnt software architecture design techniques, UML modeling, project management, testing and much more, and were able to apply it all in this project.

The next big achievement is the things we learnt during this project. New languages, frameworks, libraries, different software's for diagrams, Database. All that will be useful for us in our future along with creative and critical thinking.

### **7.3. Critical Review**

The critical element of our project is that we've merged two most used systems with additional advanced features while providing interactive user interface and fulfilling all the user requirements at a single platform.

### **7.4. Lessons Learnt**

Lessons learnt from our Final Year Project of Graduation not include only technical aspects. Besides technical aspects we've gained many moral and professional terms.

In technical overview, we gone through many problems in interface, java and IDE and got to know how to tackle with a technical problem without having any learning about it.

Then we got to know that team management, time management, strategy planning are all core tasks to do a project, and all these thing are very helpful for our future and professional life.

### **7.5. Future Enhancements/Recommendations**

- We will add intelligent newsfeed feature
- Advancement is security purpose, as due to handling functional requirements we didn't get enough time for non-functional requirements.
- Enhancement in Augmented maps to give top-ups and discounts, for the core purpose of user interaction and application usage is intended.

# Appendices

## **AppendixA: User Manual**

In this appendix section we describe the different phases of user interface and also describe how user can use our project.

### **A.1. SignUp**

### **A.2. Login**

### **A.3. User Item Selection Panel**

## **Appendix B: Administrator Manual**

Currently we are directly managing database which is Firebase.

## **Appendix C: Information / Promotional Material**

### **C.1. Broacher**



**ARvibe**

**Basic Social Media**

**Enhanced Maps**

This feature will provide the facility that user will see the services and discounts on the route they are following user will get many benefits by using this feature. User will be notifying if there is any discount in their route. Only ARvibe will provide all these features in single Social media network.

**Augmented Maps**

The AR Maps uses the camera on your smartphone to tell you which direction to walk. By using this user can find the shortest way of their destination. User will be get so much entertain by using this feature user will be see augmented world by using Augmented maps. ARvibe will provide this feature.

**Plain Maps**

## C.2. Standee



### C.3. Banner

**Introduction**

Keeping the current technological and social trend of people towards digital world and their reliance, this is an android app entertaining and providing navigation services to all android users.

ARvibe is a social network to develop physical interaction of users with real world and notifying them and their friends (based on privacy) to see them again either when they go to that remote place again and see their memories at that location or anytime on their phone according to their mood by their facial expressions.

Business places can put offers which would appear on maps and user would also be notified if reached at nearby area. Moreover chat, audio/video, file sending would also be possible.

The ARvibe logo is displayed in a large, white, serif font on the right side of the banner. The background of the banner is dark purple with white diagonal lines on the left and right sides.

# Reference and Bibliography

## Reference and Bibliography

[1]s

**Author** View &rarr;  
**Article title:** Class Diagram Relationships in UML Explained with Examples  
**Website title:** Creately Blog  
**URL:** <https://creately.com/blog/diagrams/class-diagram-relationships/>

[2]

**Article title:** A List Of Emotions And Facial Expressions  
**Website title:** Thought Catalog  
**URL:** <https://thoughtcatalog.com/january-nelson/2018/06/list-of-emotions/>

[3]

**URL:** <https://developer.android.com/guide/topics/>