
Software Requirements Specification

for

SecureFam

Version 1.0 approved

Submitted to
Dr. Vinod K. Bhalla

Submitted by Group 4, COE 1

Priyanka Gupta (101803006)
Abhinandan Chugh (101803003)
Harsimran Kaur (101853015)
Shivam Raj Singh (101853021)

Thapar University, Patiala

23 November, 2020

Table of Contents

Table of Contents	2
1. Introduction.....	3
1.1 Purpose.....	3
1.2 Document Conventions.....	3
1.3 Intended Audience	3
1.4 Project Scope	3
1.5 References.....	3
2. Overall Description.....	4
2.1 Product Perspective.....	4
2.2 Product Features.....	4
2.3 User Classes and Characteristics.....	5
2.4 Operating Environment.....	6
2.5 Design and Implementation Constraints	6
2.6 User Documentation	6
3. System Features	7
3.1 Live Location Tracking.....	7
3.2 Chatting with members	7
3.3 Check-Ins on various locations	7
3.4 Location History	8
3.5 Emergency SOS Messaging.....	8
4. System Models.....	9
5. External Interface Requirements	13
5.1 User Interfaces	13
5.2 Hardware Interfaces	13
5.3 Software Interfaces	13
5.4 Communications Interfaces.....	13
6. Other Non Functional Requiements	13
6.1 Reliability and availability	13
6.2 Responsiveness	14
6.3 Security Requirements	14
6.4 Maintainability	14

1. Introduction

1.1 Purpose

This document outlines the usage and implementation of the SecureFam app, an android application to help people track their family members. More often, parents feel a need to track their children on their way to school, coaching etc. This app is best suited for such purposes, ensuring more safety to children. Moreover, people travelling late at night can share their live location with the family members or notify them of any emergency during the journey. Besides, This app can serve many other purposes like messaging and check-ins and hence keeps the family secure.

1.2 Document Conventions

This Document was created based on the IEEE template for System Requirement Specification Documents.

1.3 Intended Audience

It is made keeping in mind the needs of developers, project managers, marketing staff, testers and documentation writers.

1.4 Project Scope

SecureFam app will be an android application and will be able to locate a phone on a map with the help of GPS. This app's technology will be easy and simple, and it can help you and your loved ones find each other's location in real-time. The system will be designed to help the user to-

- Track their family members
- Share their live location
- Chat with the family members
- View last location history of family members
- Send emergency alerts to family members

1.5 References

<https://www.visual-paradigm.com/guide/uml-unified-modeling-language/what-is-use-case-diagram/>
<https://www.bouml.fr/documentation.html>
<https://www.lucidchart.com/pages/>
<https://www.trackschoolbus.com/blog/best-gps-tracking-apps-for-android-and-ios/>
<https://youtu.be/XFTAJj2N2Lc>
<https://developers.google.com/maps/documentation/android-sdk/map-with-marker>

2. Overall Description

This section of this document provides a general description, including characteristics of the users of this project, the product's hardware, the functional and data requirements constraints and the assumptions made for the development of this project. of the product.

2.1 Product Perspective

SecureFam is a mobile application with a GPS (Global Positioning System) service in order to share their location or track them. The mobile application will work on all mobile Android devices. It will have functions as sharing location, checking in to various places, sending emergency alerts and messaging with users. When users run the application and turn on the GPS, they can see all the location history and members near to them. All information will be kept private to that specific family member and the app will be password protected.

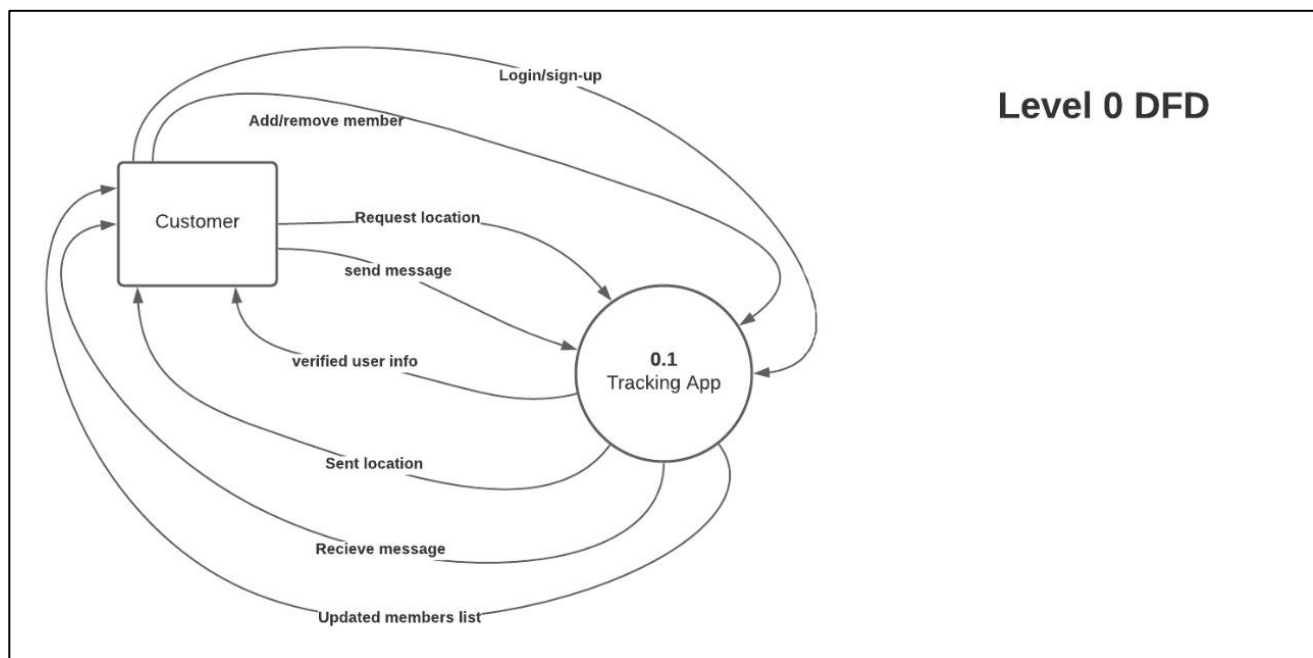
2.2 Product Features

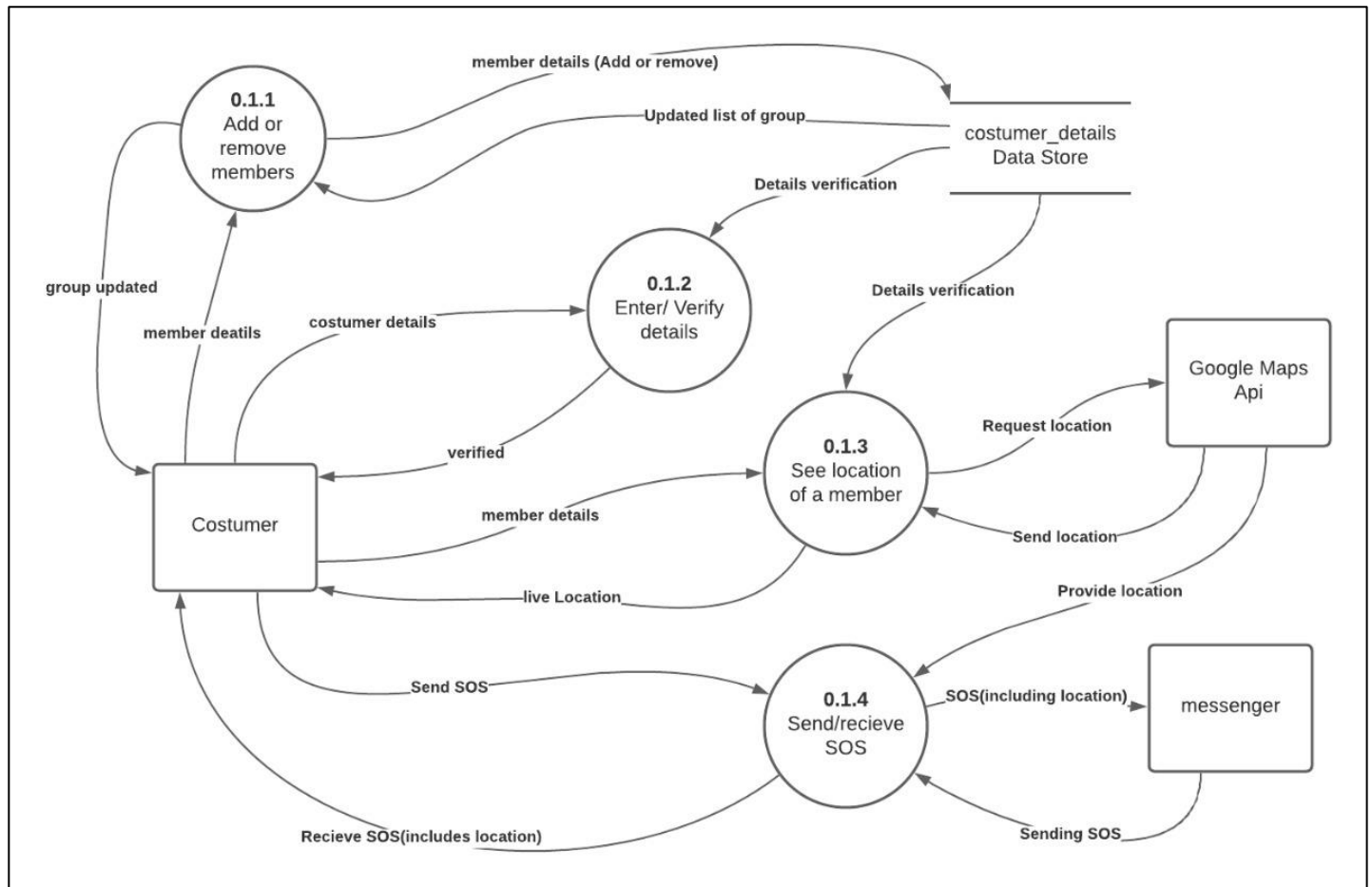
SecureFam consists of these main tasks-

- Cross platform support
- Sign up and Authentication
- Messaging
- GPS service
- Search users and locations

To start with the app, users will have to login with existing account or sign up with username and password. So, they have their own account to manage locations and join their other family members on the app. It will also allow other members to see users' profile to get information about present location and check the last location history of user. It will also notify the user about the members near to him. However, they can turn off their GPS as and when they like.

Data Flow Diagrams



DFD Level 1

2.3 User Classes and Characteristics

- *System Administrators:* User class with domain system privileges which allows them to maintain the entire system, manage enrollment and create virtual classes. Users in this class have complete control and knowledge of the entire system.
- *Administrative End User:* User class with domain class privileges, which allows them to maintain and support the class they are enrolled in, along with the other enrolled users. Users in this class have zero knowledge of the back end of the system; however, they know the front end of the application.
- *Limited End User:* User class with limited class privileges, which allows them to operate the front end of the system with limited read only permissions. Users in this class will only be able to make modifications to their profile section.

2.4 Operating Environment

Operating environment is, as just mentioned above, is an Android OS (v2.3.3 +) mobile device. An android device that can support basic dependencies of the application, for example

- Ability to connect to the Wi-Fi or mobile network.
- Ability to exchange data over the network
- Touch screen for convenience or Keypad (in case touchpad not available)
- Processor with speed of 500 MHz
- Ability to use camera, gallery, microphone and other services of mobile
- Ability to take input from user
- Storage memory required between 50 MB and 75 MB

and support for the base coding language Kotlin, is expected for proper user experience. For backend database and services, a third-party cloud service, Google Firebase is used. This also allows real time updation of user location in the application

2.5 Design and Implementation Constraints

- One important constraint is privacy and security. Users should be accessing only the authenticated data i.e. only their profile and locations of friends.
- The information of all users and their locations must be stored in a database that is accessible by the app.
- The Securefam app will run in background whenever there is an internet connection.
- Users should be able to access it from any android mobile that has internet.
- Users must have been registered in order to access the functionalities.

2.6 User Documentation

Users can take online help by contacting or through email support using the provided contact number and email account on the about section of the app. Registered users will also be provided with the manual to help them get started with the app.

2.7 Assumptions and Dependencies

- The user is Indian as it can track friends within India only.
- Mobile phone has an active internet connection on the device.
- GPS facility is present on the device.
- Client/User runs an operating system which is Android (v2.3.3 +) or more.
- Our app will not be violating any Internet Ethnic or Cultural Rules and won't be blocked by the Telecom company

3. System Features

3.1 Live Location Tracking

3.1.1 *Description and Priority*

This allows you to see where your relative is in real-time, for example, you could see at 7:38 pm where they were or as near as possible. Parents can check the real-time location of their child and then write to them in the chat to ask questions, confirm plans, and more

3.1.2 *Stimulus/Response Sequences*

Users need to click on the profile of the user or can directly see the friend on the map along with the other friends. It will show us the location, hence enabling this live tracker.

3.1.3 *Functional Requirements*

1. The user should be registered and has to login to see all the locations.
2. User's friend should have enabled the GPS
3. Both the users should be connected to internet.

3.2 Chatting with members

3.2.1 *Description and Priority*

This feature allows everyone to interact within a group, as well as allowing individuals to communicate with each other one by one. It's a great way to keep everyone in the loop about the day's plans, and kids and parents to check-in.

3.2.2 *Stimulus/Response Sequences*

User can click on the option on the friend's profile to privately message him or can tap on the pointer on the map to interact.

3.2.3 *Functional Requirements*

The only requirement to message is that the user and the user's friend should be connected to internet to exchange messages, besides being registered.

3.3 Check-Ins on various locations

3.3.1 *Description and Priority*

This allows the user to check in on certain locations to ensure that they have arrived safely and to keep track of their location.

3.3.2 *Stimulus/Response Sequences*

Users need to find the location on the map, and click on the check-in option present at the bottom of the page. Doing this, will mark the user at that location on the map along with the time and date. It will be notified to all the user's friends.

3.3.3 *Functional Requirements*

Users need to have their GPS on and active internet connectivity to use check-ins. One should also make sure that the place is visible/globally registered on the maps, and within the country.

3.4 Location History

3.4.1 *Description and Priority*

The family tracker app can also show the 24 hrs history of a relative.

3.4.2 *Stimulus/Response Sequences*

User need to click on the pointer/marker on the map of the friend to view the history or indirectly through the friend's profile.

3.4.3 *Functional Requirements*

The only requirement is that the registered user should have an active internet connection.

3.5 Emergency SOS Messaging

3.5.1 *Description and Priority*

If the user (especially women) is in an emergency situation, then the user can simply press the emergency button such as SOS and the application will automatically send SMS alerts in form of beep or buzzer with functionality of including location address to the store mobile numbers that might be a police station or close relative.

3.5.2 *Stimulus/Response Sequences*

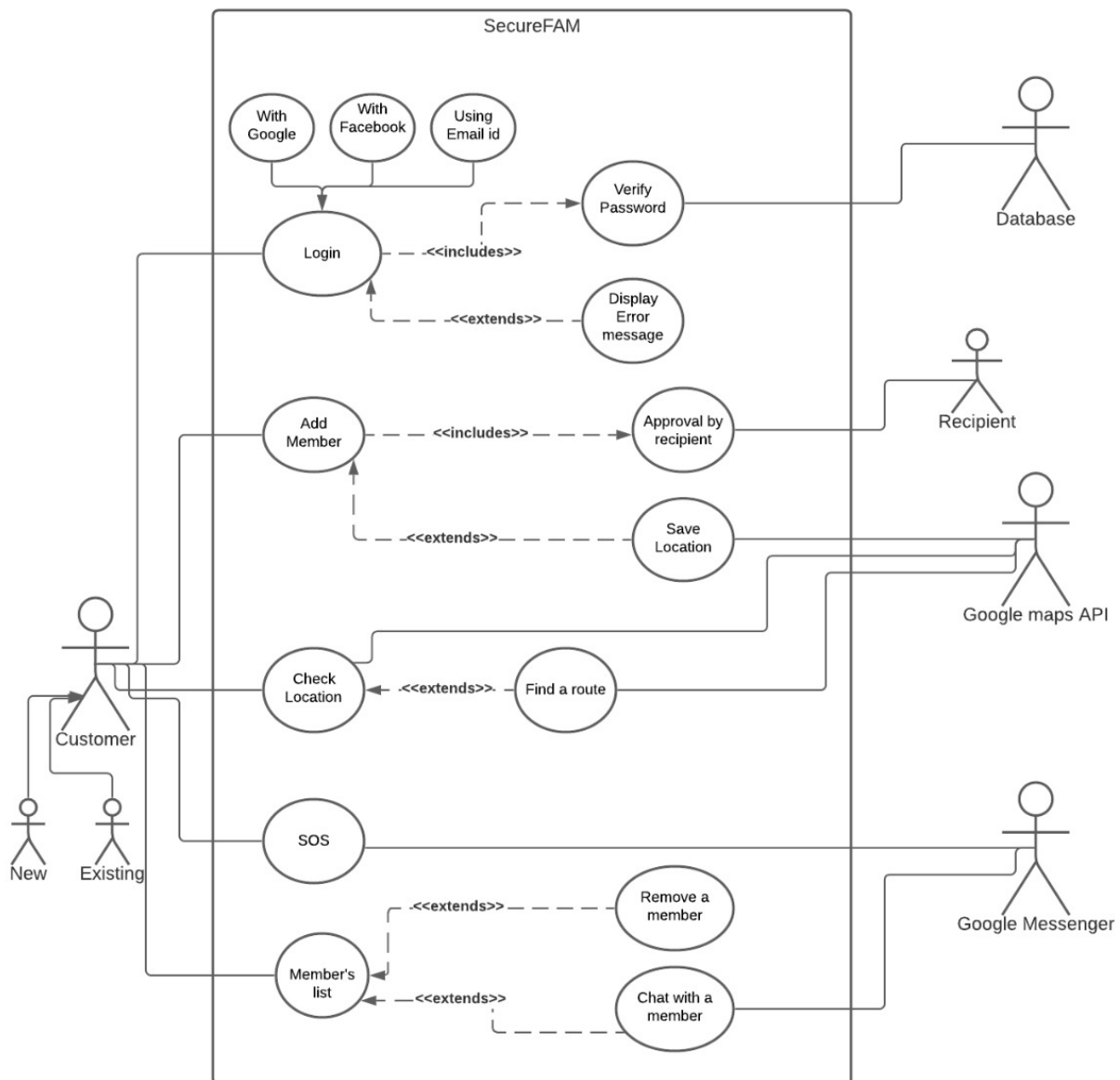
User need to click on the SOS msg button present at the top of the home page of app.

3.6.3 *Functional Requirements*

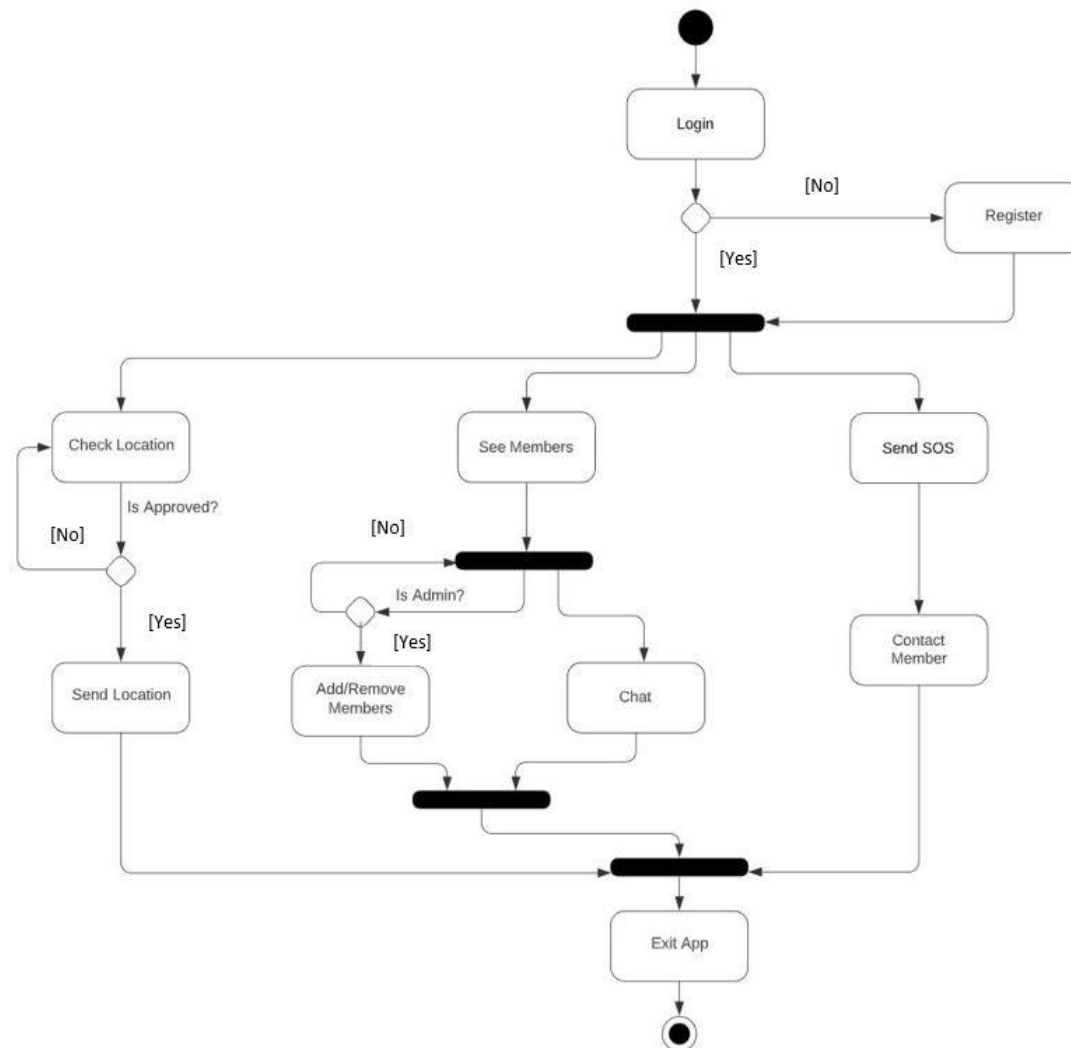
User needs to have mobile phone with no active internet connection required. It will send the SOS message to special friends immediately.

4. System Models

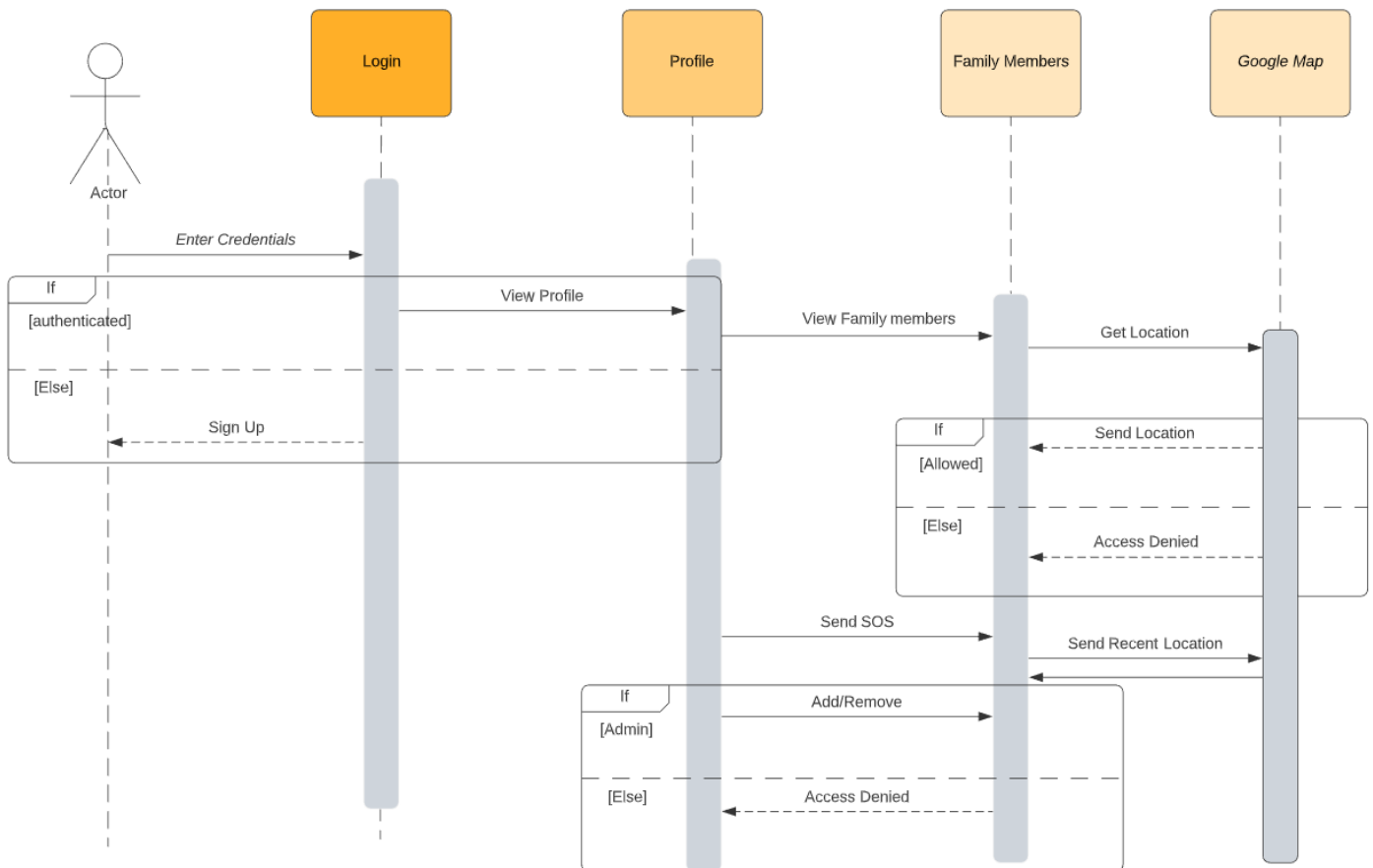
1. Use Case Diagram



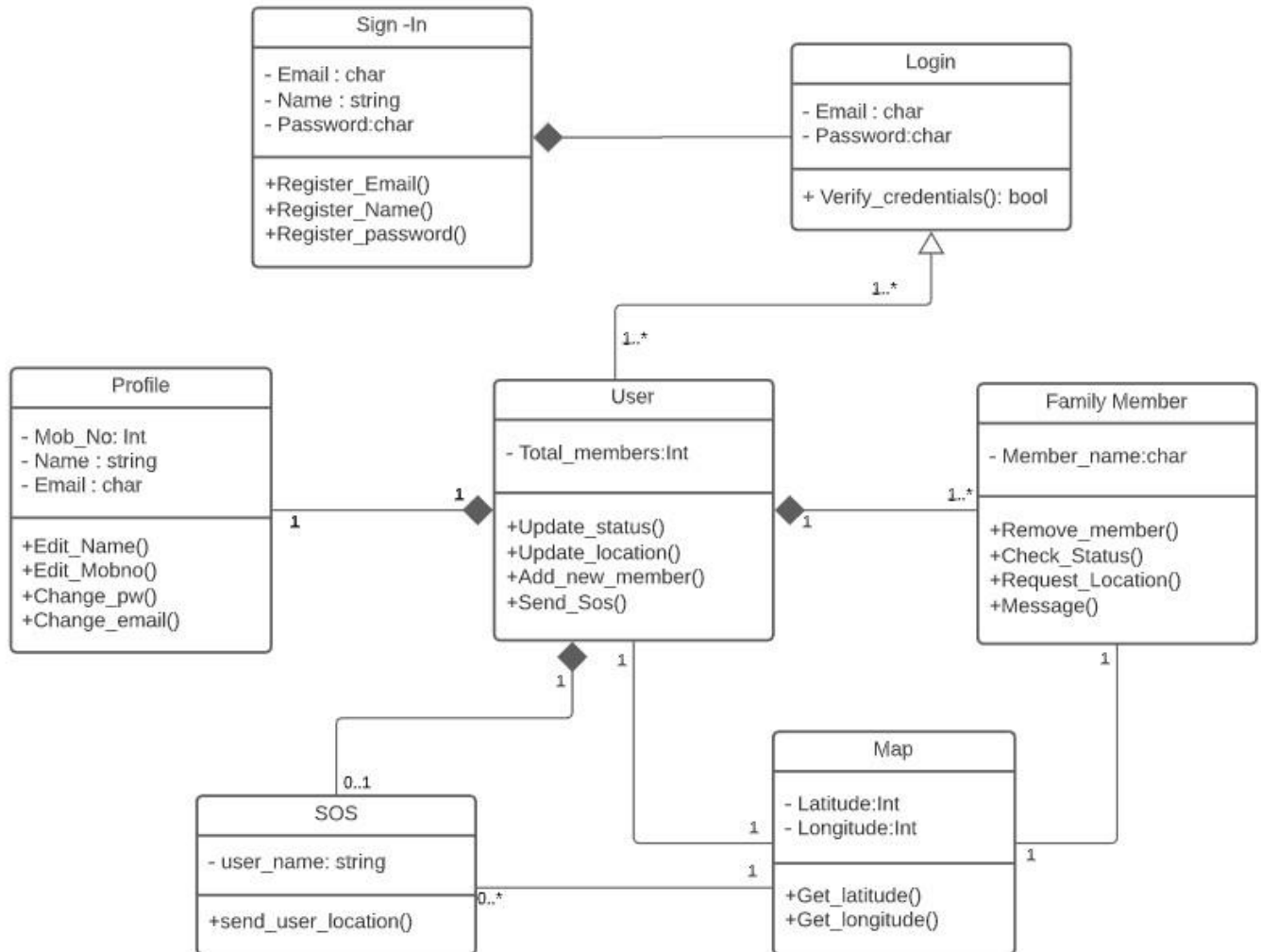
2. Activity Diagram



3. Sequence Diagram



4. Class Diagram



5. External Interface Requirements

5.1 User Interfaces

User interface of our application will be easy to use and understandable. The user is expected to know how to use Android mobile devices and to be able to write and read messages and use buttons. The user interfaces will include login/signup interface, location tracking interface and messaging interface.

5.2 Hardware Interfaces

A mobile phone which has basic additional functionality like touchscreen, GPS receiver and a camera for optional image message exchange.

5.3 Software Interfaces

The mobile application communicates with the GPS application in order to get geographical information about where the user is located.

- Operating system - Android operating system is chosen for its best support and userfriendliness.
- Android Studio - To implement the project we will use Android Studio, with Kotlin language.
- Google Maps API – To implement the google maps based GPS tracker in the app.
- Database - We have chosen Google Firebase for database work.

5.4 Communications Interfaces

This application solely uses Wi-Fi Direct to communicate Android devices. Recommended data transfer rate is 1.5 MBPS Download speed (for any activities) and upload speed of minimum 128 Kbps (messaging).

6. Other Non-Functional Requirements

6.1 Reliability and availability

The capability to maintain the specified level of performance is what meant by reliability. This application will run on any android phone. The application will run 24 X 7 if internet connection is available.

6.2 Responsiveness

The Application should be responsive to the user Input or to any external interrupt which is of highest priority and return back to same state. When app gets interrupted by call, then app should be able to save state and return to same state/ page which was there before it got interrupted. If the user got disconnected to internet, the page will be reloaded and saved as and when it gains access.

6.3 Security Requirements

All the entered information will be handled and stored by our system in an online repository (database) therefore user's data is safe in database and also passwords are encrypted. All the app data should be secured and be encrypted with minimum needs so that it's protected from outside environment also from internal attack. The app will be password protected and to gain somebody's live location information, it will need that person's permission.

6.4 Maintainability

Maintenance is one form of change that typically is done after the software development has been completed. As the time change, so do the needs. It revolves around the understanding of the existing s/w and the effects of the change. This application needs a timely updation of information table of the database by the admin. Any other feature as per the requirement can be added any time by the admin. We have 75 percent maintainability for 24 hours, this means that there's a 75 percent chance the component can be fixed in 24 hours.

6.5 Performance Requirements

Performance of the software highly depends on type and version of android installed in the system i.e. it should be compatible. Under normal conditions, all data would process in record time depending on internet speed. Any person in need of an app adaptable to their requirements can make their amends just by registering to the app. Moreover, the user can also make an extra effort to contact experts for a great advice, who can guide them well under supervision, for better performance of the app.