**Software Requirements Specification**

**IITI Connect**

1.0

15-01-2016

Akshaya Kumar

Kalyan Garigapati

Shubham Burewar

Tanmay Saraswat

Prepared for

CS 258 Software Engineering

Spring 2016

# Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| **Date** | **Description** | **Author** | **Comments** |
| 15-01-2016 | Version 0.0 | Kalyan Garigapati  Shubham Burewar | Initial draft |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

**Table of Contents**

Revision History II

1. Introduction 1

1.1 Purpose 1

1.2 Scope 1

1.3 Definitions, Acronyms, and Abbreviations 1

1.4 References 2

1.5 Overview 2

2. General Description 2

2.1 Product Perspective 2

2.2 Product Functions 2

2.3 User Characteristics 2

2.4 General Constraints 3

2.5 Assumptions and Dependencies 3

3. Specific Requirements 3

3.1 External Interface Requirements 3

3.1.1 User Interfaces 3

3.1.2 Hardware Interfaces 7

3.1.3 Software Interfaces 7

3.1.4 Communications Interfaces 7

3.2 Functional Requirements 8

3.2.1 Emergency Alert Notifications 8

3.2.2 Chat threads 8

3.2.3 College Schedules 8

3.2.4 College Maps 9

3.2.5 Web Client 9

3.3 Classes / Objects 9

3.4 Non-Functional Requirements 9

3.5 Design Constraints 9

3.6 Logical Database Requirements 9

# 1. Introduction

## 1.1 Purpose

## The purpose of this document is to present a detailed description of the IITI Connect (Emergency Warning System & Local Chatting). It will explain the purpose and features of the system, the interfaces of the system, what the system will do, the constraints under which it must operate and how the system will react to external stimuli. This document is intended for our client, Dr.Abhirup Datta, our mentor, Dr. Abhishek Srivastava and the developers of the Android application & web server.

## 1.2 Scope

This software will be an Android application for the students & faculty of IIT Indore (hereafter referred to as 'the institute'). The system will be designed to provide location-based campus-wide warning system & connectivity in the form of a simplified real-time chat system as well as campus mapping. It will also provide schedule details such as time-table & holiday lists.

More specifically, this application is designed to allow anyone in the institute to communicate easily to reduce the use of e-mail.

## 1.3 Definitions, Acronyms, and Abbreviations

## 

|  |  |
| --- | --- |
| **Term** | **Definition** |
| User | Anyone (students & faculty) who uses the client Android app. |
| Database | Collection of all the information monitored by this system. |
| Server | The main computer of the institute network which will be connected to the database & responsible for user-database & user-user connectivity. |
| Admin | Person who receives articles, sends articles for review, and makes final judgments for publications. |
| Web Client | Webpage which is opened in the security room, which is used to authenticate the warnings sent by the users & to display help requests. |
| Software Requirements Specification | A document that completely describes all of the functions of a proposed system and the constraints under which it must operate. For example, this document. |
| Client | The person with an interest in the project who is not a developer. (Dr. Abhirup Datta) |
| Mentor | The person who provides guidance & support to the developers' group. (Dr. Abhishek Srivastava) |
| API (Application Program Interface) | A set of tools for building software & applications. |

## 1.4 References

*\** Software Engineering : A Practitioner's Approach *seventh edition* by Roger S. Pressman*, 2010*

## 1.5 Overview

The next chapter, the General Description section, of the document provides a brief overview of the application & context for the technical requirements specification in the next chapter.

The third chapter, the Specific Requirements section, of the document is written primarily for the developers & describes the details of the functionality of the product in technical terms.

# 2. General Description

## 2.1 Product Perspective

IITI Connect is a Web-based Android application providing a local chat system along with an emergency warning system for students & faculty of the institute. The app will also facilitate viewing schedules & holidays.

At present, there is no warning or alert system in the institute regarding any kind of mishaps or misdoings, which is a problem considering the huge size of the institute campus.

This application is developed to resolve this very issue. Warnings will be categorized by function.

## 2.2 Product Functions

The IITI Connect application has 4 chief functions.

The Warning system (or S.O.S) is where the user can send a warning along with the location of the sender to the server in case of any emergency.

The Chat system is a basic chatting function which will be in real-time and can be two-way or in a group.

The Schedule system will consist of time-tables, bus schedules available to the user.

The Map system will provide GPS-based view of the campus along with user location (not to be confused with device tracking).

## 2.3 User Characteristics

The users of this application are expected to own an Android smartphone (Android 4.1 or higher) since the application is presently available only on the Android platform.

## 2.4 General Constraints

## The database used by the application will be stored on the local server. Also, we won't expose any user info such as phone numbers.

## 2.5 Assumptions and Dependencies

About 92.6% of the target audience own an Android smartphone (with Android 4.1 or higher) which is a safe assumption since this application is presently available on the Android platform only, which is currently our primary concern.

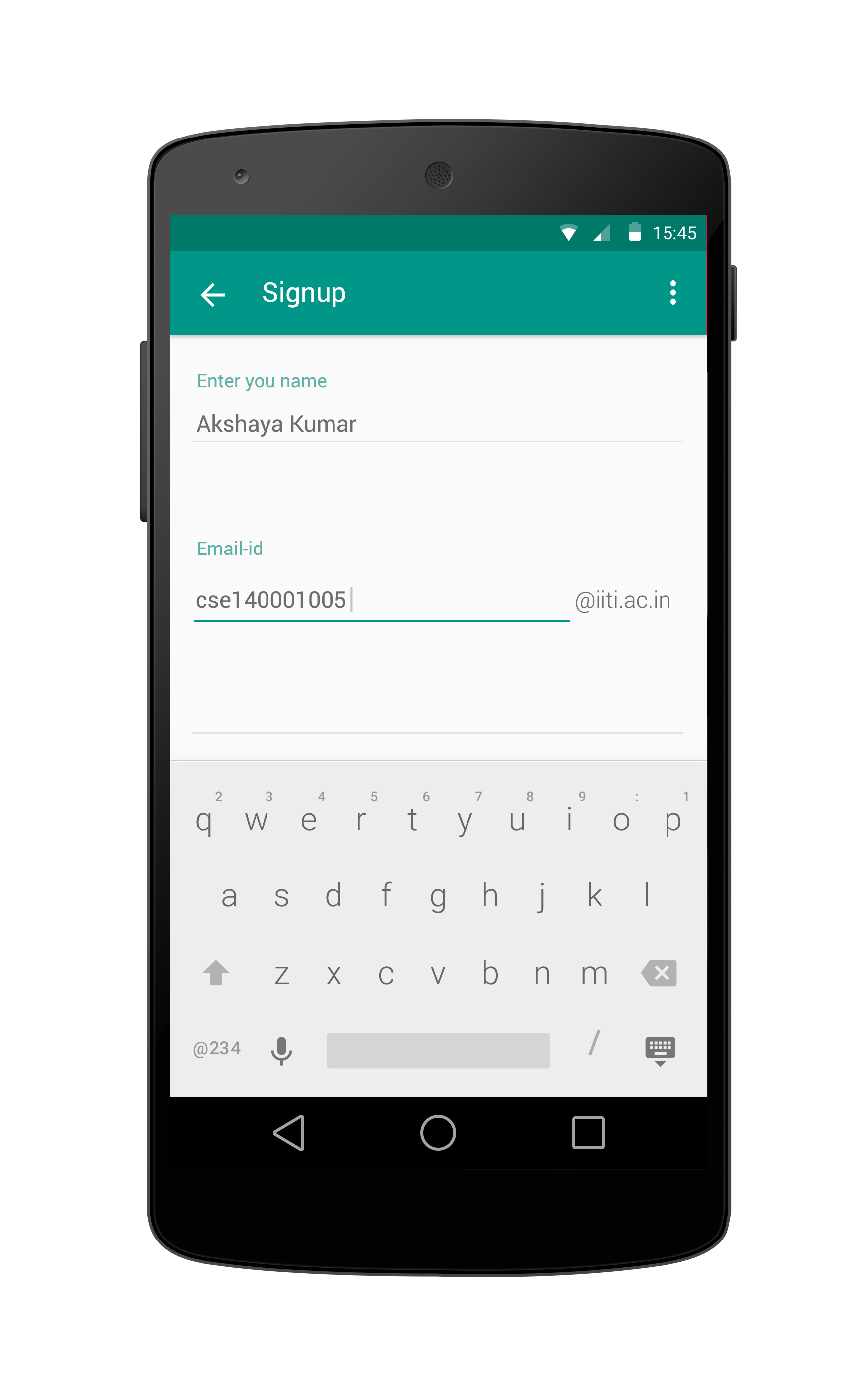
# 3. Specific Requirements

This section contains all of the functional and quality requirements of the system. It gives a detailed description of the system and all its features

## 3.1 External Interface Requirements

### 3.1.1 User Interfaces

This section shows the various functions of the app as the user will see them. These previews have been created using the latest visual language - Material Design.

******

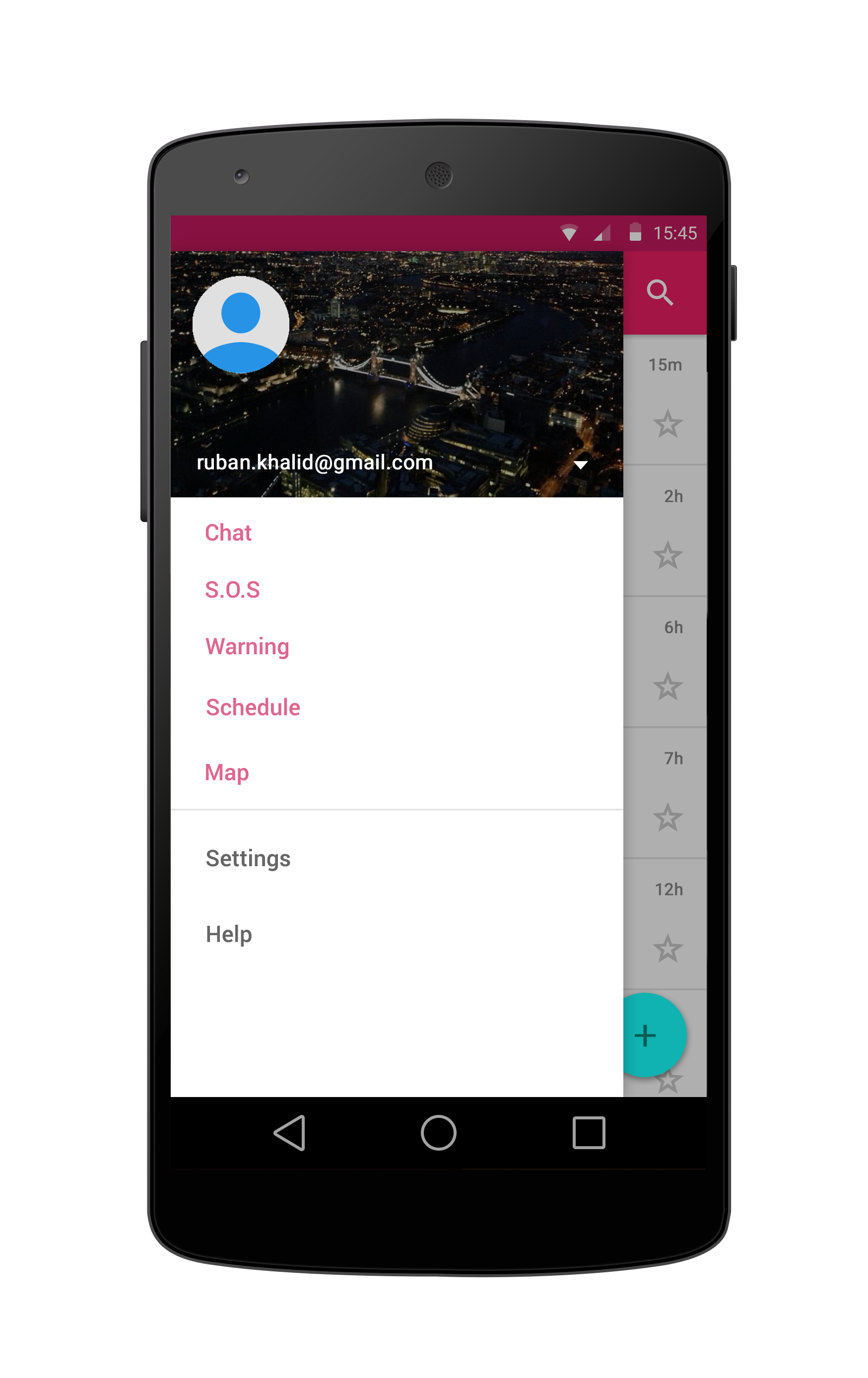
***Signup page***

A first-time user of the mobile application should see the log-in/ Register page when he/she opens the application as shown here

### 

### *chat screen.pngChat screen*

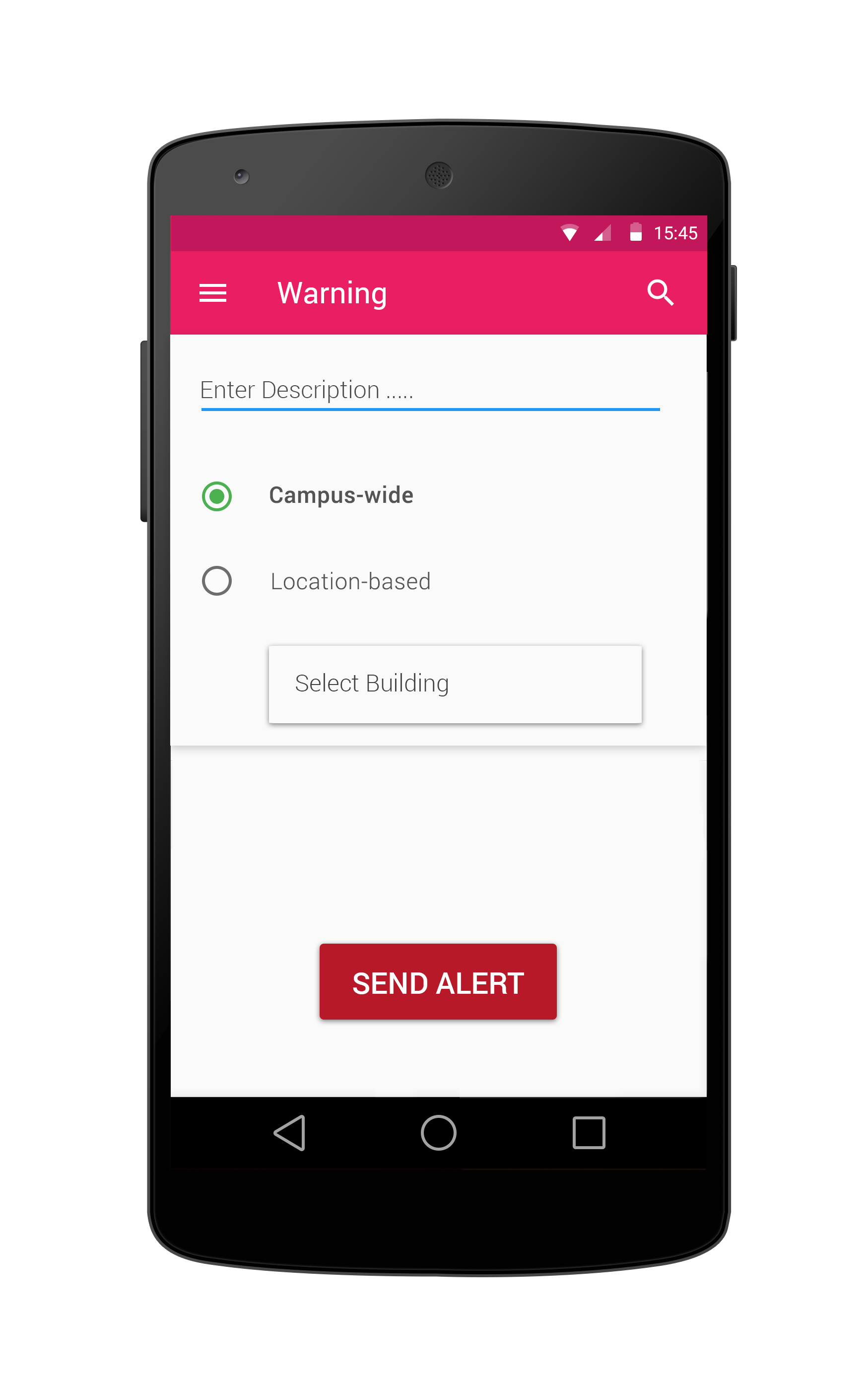
The application will open in the chat screen by default showing the recent chats along with most recent conversation time.

******

***Menu drawer***

All the functions offered by the app can be seen in the menu drawer which includes Chat, SOS, Warning, Schedule & Maps.

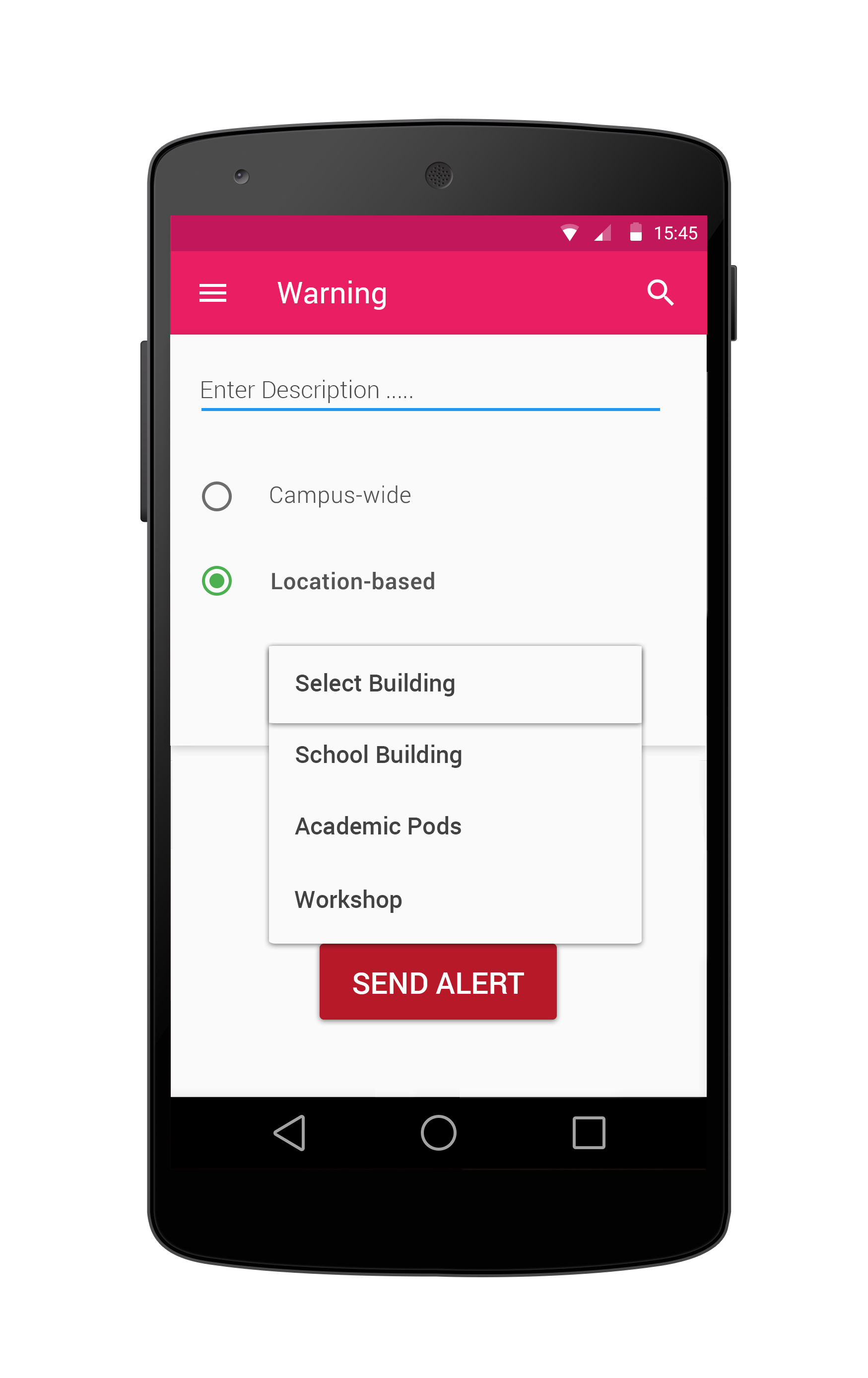
You can also see Settings & access Help.



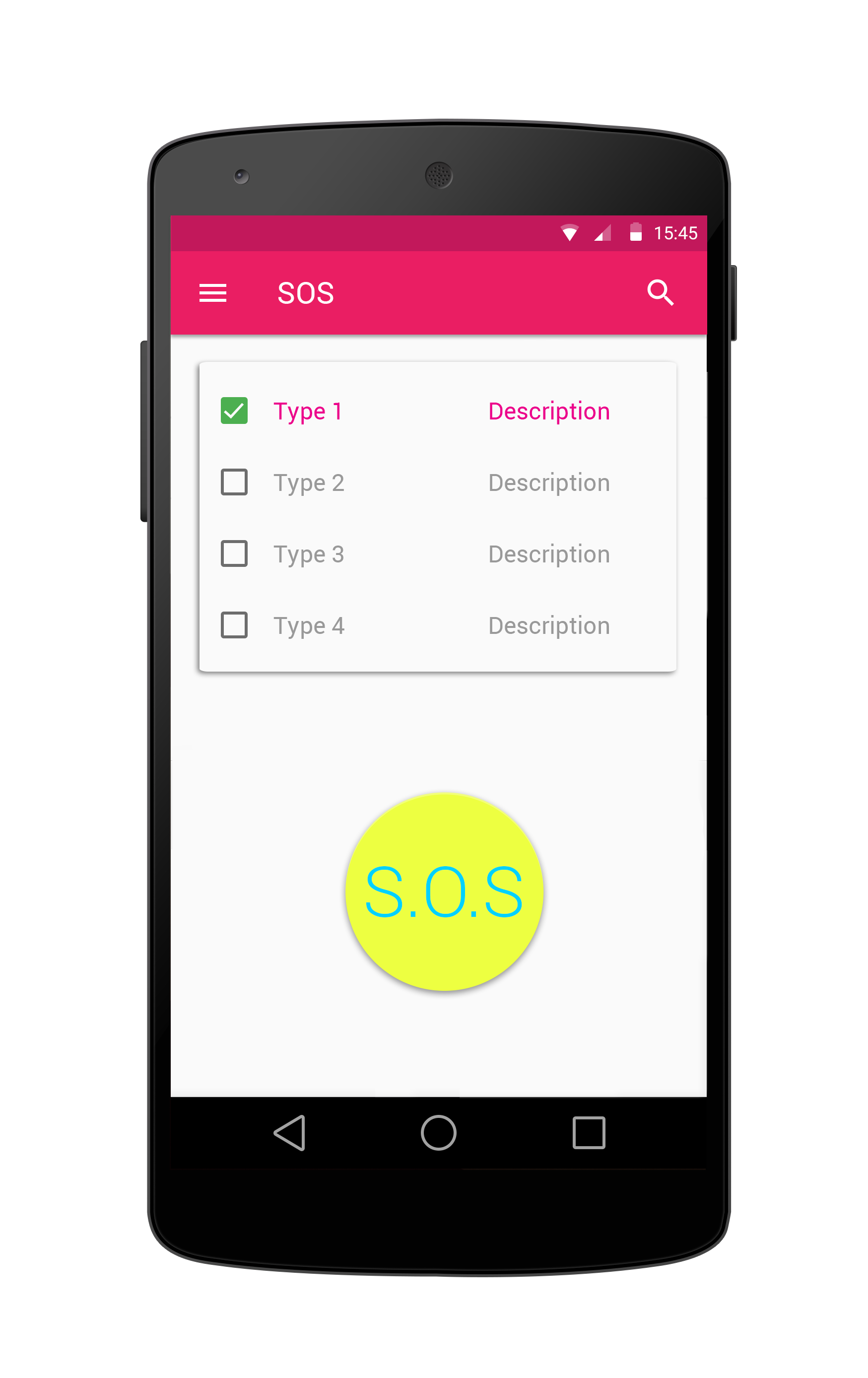
### *Warning screen*

The user can also broadcast warning alert to every user on the campus, or broadcast it based on a location such as a particular building.

### Campus-wide warning will alert every user in the campus connected to the server.



Location-based warning will alert all the users in a particular location such as a building or the workshop in case of an accident or chemical leakage.



### *S.O.S Screen*

### The user can send a help request to the server who will inform the security. The SOS is categorized which can be selected by the user, according to the situation as shown here.

### schedule screen.png

### *Schedule screen*

### Any kinds of schedule ranging from time-tables to holiday lists can be viewed by the user in the Schedule section as shown.

### map screen.png*Map screen*

### The map interface is yet to be decided. It may be created using Google Maps API or made from scratch. We may or may not make the final layout as shown in the layout . This is due to constraints presented by the proxy server.

### 3.1.2 Hardware Interfaces

Since neither the mobile application nor the web portal have any designated hardware, it does not have any direct hardware interfaces. This Android application requires ‘current location’ of the user where the user needs to send an SOS. The physical GPS is managed by the GPS sensor in the mobile phone and the hardware connection to the database server is managed by the underlying Android operating system on the mobile phone and the web server hosted at IIT Indore servers.

### 3.1.3 Software Interfaces

The mobile application communicates with the GPS application in order to get geographical information about where the user is located and the visual representation of it, and with the server and database in order to send data for chatting and SOS messages. The communication between the database and the mobile application consists of reading as well as writing options.

### 3.1.4 Communications Interfaces

This mobile application shall use TCP/IP protocol for communication through intranet (Local connectivity) . It may use internet through HTTP protocol for real time pushing which is useful in firing the alarm in all the users devices.

**MySQL Database**

It’s an open source SQL database to store all data which communicates with the application through the server. i.e., server will be always connected to the database

## 

## 3.2 Functional Requirements

### 3.2.1 Emergency Alert Notifications

3.2.1.1 Purpose

To send/receive emergency or warning notifications to other users

3.2.1.2 Input

User can send the alert notifications to others by pushing button of desired emergency level which will be provided on the SOS layout of user interface.

3.2.1.3 Processing

The message from the user will be sent to the local server where it has get authenticated depending on the emergency level and then it will be forwarded to other users.

3.2.1.4 Outputs

The message from the server will be sent to all users in form of Android Notification with an alarm sound.

### 3.2.2 Chat Threads

3.2.2.1 Purpose

To provide communication and offer a real time transmission of text messages from sender to receiver.

3.2.2.2 Input

A text message (String) sent from user along with some information like senders Id, receivers Id, sent timestamp will be delivered to server.

3.2.1.3 Processing

The server on receiving the message from the sender, will forward the data to actual receiver.

3.2.2.4Output

Finally the message will be delivered to the receiver addressed by the sender at the time of sending.

### 3.2.3 College Schedules

3.2.3.1Purpose

To help the users (Students and Staff) in comfortably look into their timetable.

3.2.3.2 Input

Does not require any input. The data will be fetched from the server.

3.2.3.3Processing

The app will download the data when connected to the College server.

3.2.3.4 Output

After downloading the data app will display the neatly parsed data onto the screen.

### 3.2.4 College Maps

3.2.4.1 Purpose

Purpose of this map is to help the users in moving across the campus.

3.2.4.2 Input

Does not require any input. The data will be fetched from the server.

3.2.4.3 Processing

The app will download the data when connected to the College server.

3.2.4.4 Output

After downloading the data app will display the neatly parsed data onto the screen.

### 3.2.5Web Client

3.2.5.1 Purpose

To authenticate and process security and warning alerts.

3.2.5.2 Input

Alert from client devices.

3.2.5.3 Processing

Server will receive alert generated by client application and forward it to web client where a notification will be generated. A person in charge will validate the request.

3.2.5.4 Output

According to the validation a notification will be generated to all client applications or in case of help request, action will be taken in the most appropriate way.

## 3.3 Classes / Objects

* *Code not written yet*

## 3.4 Non-Functional Requirements

* *Code optimisation not done since the it is not written yet.*

## 

## 3.5 Design Constraints.

App runs on Android version above 4.1 which would be compatible with 92.6% of android devices in the world.

## 3.6 Logical Database Requirements

MySQL database is used for storing data of Schedules, chat threads which were not delivered and warning/help requests generated.