

# Outline

## **Week1 contents**

Introduction, Reason for choosing, Key Features, Why Android?

## **Week2 contents**

Related Work, Domain Research

## **Week3 contents**

Android Architecture, Android Components Framework

## **Week4 contents**

Requirements for this Project, Requirements Specifications

## **Week5 contents**

System Architecture, Proposed System, Working Flow

## **Week6 contents**

Use Case Diagrams

## **Week7 contents**

Implementation and GUI

## **Week8 contents**

Testing and Future Aspects

# Week1

## Project Introduction:

An Android Application(Shakti) for the Safety of Women and this app can be activated this app by a single click, whenever need arises. A single click on this app identifies the location of place through GPS and sends a message comprising this location URL to the registered contacts and also call on the first registered contact to help the one in dangerous situations. The unique feature of this application is to send the message to the registered contacts continuously for every five minutes until the “stop” button in the application is clicked and user can register complaint online about any harassment by a single click. Continuous location tracking information via SMS helps to find the location of the victim quickly and can be rescued safely.

Whether you are in instant trouble or got separated from friends during night and do not know how to get home, having these apps on your phone can diminish your risk and bring assistance when you require it

# Reason for choosing this Project?

Women are accomplished at mobilizing diverse groups for frequent causes. They often work across racial, sacred, opinionated, and intellectual divides to encourage tranquility. We are aware of importance of women's security, but we must recognize that they should be well secured. As much as we would like to get rid of them, these situations are unavoidable and usually unexpected. The nature and consequences of these situations can vary significantly and in worst cases also be life threatening. Therefore it would be a need to have some mechanism by which we can notify certain people about such circumstances and increase the chances of receiving help as soon as possible. The need for such a mechanism increases even more as in this era of technology, platforms exist to support them. One such platform and a very common one is a Smartphone.

In today's world, people using smart phones have increased rapidly and hence, a smart phone can be used efficiently for personal security or various other protection purposes. The heinous incident that outraged the entire nation have waken us to go for the safety issues and so a host of new apps have been developed to provide security systems to women via their phones.

In the light of previous outrage in Delhi which shook the nation and woke us to the safety issues for our daughters, public are gearing upbeat in different ways to fight back, this is what have moved me to build an app for our backbone of our nation.

# Key features:

- 1) Initially, we have to enter the four contact numbers of police, family members and friends in to the application say and click on “save” button.
- 2) While travelling, run the application and whenever need arises, click “start” button.
- 3) As soon as “start” button pressed, it firsts make a call to the first saved registered contact number and also sends the message containing location URL of the victim to all the contact numbers.
- 4) Unique feature of this app is message with location URL is sent continuously to the registered contact numbers for every five minutes until “stop” button is clicked. So, continuous location tracking of victim is possible with this application.
- 5) Simultaneously we can raise the complaint and it'll be register in nearest police station.

# Why Android Application:

Android is the most widely used mobile OS motorized by Linux kernel. This is the first complimentary platform which is robust and is expected to gain much popularity. It was developed by Google team and allows writing managed code in the Java language. The current system is developed on the basis of android platform. Android utilizes a custom virtual machine that was designed to optimize memory and hardware resources in a smart phone. Android does not differentiate between the phone's core applications and third-party applications. Any application that is built will definitely have equal access to a phone's capabilities providing users with a broad spectrum of applications and services

# Week2

## **Related Work:**

As a part of literature survey, it is investigated some applications of safety that already exist in market. The aim is to observe how these applications work and to see how they can be improved and how are they different. To date it is identified that the following Android Apps of security are good and are offering relatively similar service

### **1. Women's Security**

The app is developed by AppSoftIndia. The key features of the app are: the user has to save some details. These details include: Email address and password of the user, Email address and mobile number of the recipient and a text message. Then, app is loaded as a “widget”, so that when the user touches the app, it alerts the recipient.

## **2. Bsafe- Personal Safety App**

The app is developed by Bipper Inc. On March 6, 2015. The app's motto is "Never Walk Alone". This app helps the user to create a gang of 'Guardians' and SOS message will be received by them when the user is in trouble.

## **3. Safetipin- Complete Safety App**

This is a privacy security app having the troop of features; GPS tracking, emergency, important contact numbers, directions to safe locations, pins displaying unsafe and freeform danger areas and a Safety Score. It drives in advance of exemplary women safety apps, and presents a vast range of features, so that they will help to practically plan and can give a counter attack to those spots in the locality.

## **4. RAKSHA –Women Safety App**

This app is launched by BJP on May 15, 2014. By clicking on this app, it sends location of the user to the contacts registered and the user can also get the details of the location of the contacts. A distress signal just by pressing a single key sends out a loud buzzer to our near and dear ones. We can add multiple contacts to this app and when there is no data connection, this app alerts the contacts by sending SMS.

# Domain Research

As this project is a website as well as an android application. The development of this needs great research work. With research work done properly one can make out the success or failure of the project, as it provides complete exposure of knowledge, business, human networking, better insights and understanding of the required area. Domain research will deal with the whole method of a Website building. Then the developer will include the study of XML, Android, SQL Queries. To make the research better, developer has divided it in the following domains

Need for Mobile Technology

Need for Android Mobile Technology

GSM, GPRS, WIFI

Visibility

Feedback and Constraints



## **GPRS/GSM/Wi-Fi**

The proposed system will provide the services across the countries also as the users often continue to use their mobile phones when they travel to other countries. The proposed system can access the internet by GPRS and the data usage must be less than 56 Kbps so that all users can effectively use the services. The proposed system can access the internet by Wi-Fi and the data usage must be less than 80 Kbps so that all users can effectively use the services

## **Visibility**

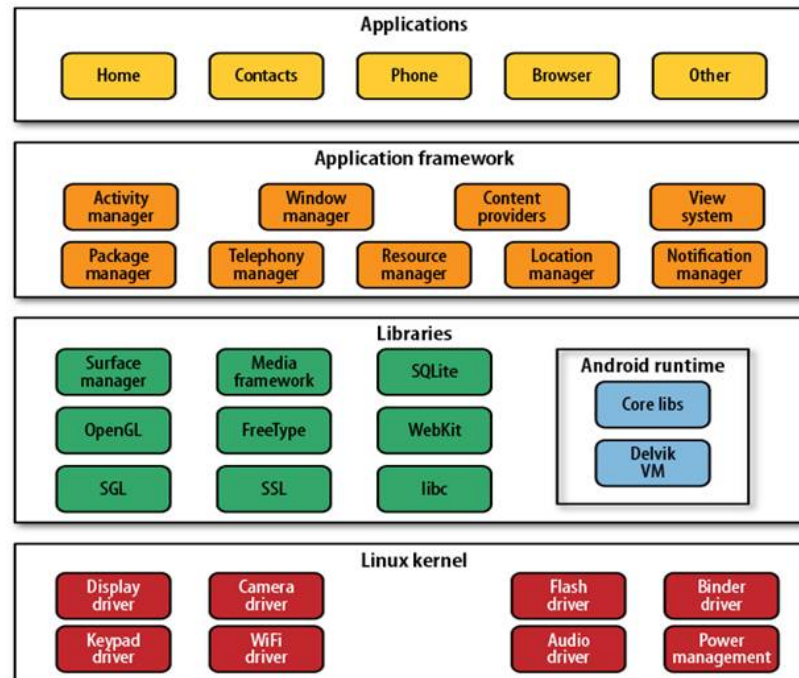
It is one of the most important design principles and what it means is that, as and when the user looks on the system screen he/she may feel the possibility for action. The developer will keep an eye on this principle in order to provide better visibility. 2.6 Feedback and Constraints Feedback is the response to the user of the action performed Constraints are some universally accepted conventions which notify some specific actions.

**Feedback and Constraints** Feedback is the response to the user of the action performed Constraints are some universally accepted conventions which notify some specific actions.

# Week3

## Android Architecture:

The Android operating system is implemented as a stack of different layers of software. The following image depicts these different layers:



**Linux Kernel** – This is the layer at the very bottom of the Android architecture. All other layers run on top of the Linux kernel and rely on this kernel to interact with the hardware. This layer contains all the essential hardware drivers which help to control and communicate with the hardware. It provides the basic functionality like Process Management, Memory Management and Device Management like Camera, Display, Flash etc.

**Libraries** – This is a set of common functions of the application framework that enables the device to handle different types of data. Some of the most important sets of libraries that are included are – Web kit which is the browser engine to display HTML, OpenGL used to render 2- D or 3-D graphics onto the screen, SQLite which is a useful repository for storing and sharing of application data.

**Android Runtime** – The Android runtime mainly consists of the Dalvik Virtual Machine (DVM). DVM is very much like the standard Java Virtual Machine (JVM) except that it is optimized for mobile devices that have low processing power and low memory. DVM generates a .dex file from the .class file at compile time and provides higher efficiency in low resources devices.

**Application Framework**- These are some standard class files that are available to the developer for use. An application can directly interact with them and make use of them. The application framework provides the most basic functionality of the phone like Location Manager, Content Providers etc.

**Applications** – This is the topmost layer in the architecture and the layer where the application that we develop fits in. This layer provides several pre-installed applications that are default for certain things like Contacts Books, Browser etc.

# Android Framework Components

Android applications are written in Java. There are different integrated environments (IDEs) that can be used to develop Android apps. SOS app is developed in eclipse using Android software development kit (SDK). SDK tools create an Android package (.apk) that contain all the necessary resources to install and run the app. Each app runs as a separate process in the underlying Linux kernel and behaves like a separate user. Files within an app can be run only by the specific user id assigned to the app. Each app also has its own instance of the Dalvik Virtual Machine (DVM). In order for the apps to share data with other apps like system services we have to assign permissions to the app during install time. This is done by adding the required permissions in the Manifest file.

**AndroidManifest.xml** The AndroidManifest.xml (Manifest) file provides important information to the Android system to run the app. All the components have to be declared in the Manifest file for the Android system to be able to instantiate them. The Manifest file also contains the various permissions needed by the application, API libraries that the app is linked with like Google maps Android API v2, other hardware and software features that the app uses and also the minimum API Level supported by the app

# Week4

## Requirements for this Project:

Requirements gathering is one of the most important phases of a software development life cycle. It is the phase that tells us what the system is supposed to do and drives the other phases in the life cycle.

Requirements were also collected by looking at other devices like personal locator beacons and satellite messengers that are commercially available. A brief study of the functionality of the devices helped me to refine and narrow down the requirements even further. One important thing to learn for these devices was the simplicity of their design. This helped me to design an effective and simple UI design for my application. The next step for requirements understanding was to look for existing solutions and similar applications in the Android market. A careful study of these applications, adding other important features and removing unnecessary features was done.

# Functional Requirements:

- i. The user of the application should be asked to log in only the first time he uses the application on his device. The user must see the main page of the app (with the buttons to send notifications) for every other time he opens the app.
- ii. The user shall be able to send notifications with the tap of a single button. Separate buttons should be available for sending panic messages, I am OK and making a call to 911.
- iii. An option must be provided to enable/disable these buttons to avoid pressing them by accident.
- iv. The user shall be able to see their current location.
- v. The user shall be able to set the contacts to send text messages from within the application. The user must also be able to set the contents of the messages. Also the user may select these contacts from the contact book or enter them manually.
- vi. They must also be able to see their location history from within the application.

# Non-Functional Requirements:

- i. Providing a simple and elegant UI for the main screen. This is necessary as the user would usually come on to this screen in case of a panic or emergency and hence each button should be clearly visible and easily pressed.
- ii. In case the option to track location is selected and there is no internet connectivity on the device (both wireless and Cellular data), the application should be able to store the locations offline and send them to be stored in the database once the internet connectivity is up again.
- iii. Providing a tab based view to display the different settings for the application and location history for the user.
- iv. Enabling swipe gestures for the tabbed view.
- v. Displaying user friendly dialogs for picking the date, time, entering the contacts to send text and email messages to and to enter the contents of the text and email messages.

# Requirement Specifications

## Software Requirements

These requirements are separated based on whether you are developing the app or running the app on a device.

### **For development:**

Operating System: Windows XP or higher/Mac OS X 15.8 or later/Linux

Platform: Android SDK Framework 10 or higher

Tools: Eclipse SDK 3.5, ADT plug-in for eclipse

Technologies used: Java, SQLite, Android, Google maps v2 API

Debugger: Android Dalvik Debug Monitor Service (DDMS)

Android Emulator: API level 14 or higher

### **For running on a device:**

Operating System: Android 3.0 or higher Cellular capabilities for SMS messages



# Requirement Specifications

## Hardware Requirements

### **For development:**

Processor: Intel Pentium IV or higher

RAM:256MB

Space on disk: 250 MB (at the least)

### **For running on a device:**

Device: Phone or tablet running Android 3.0 or higher Disk space: 6 MB (at the least).

# Week5

## System Architecture:

- i. User – This is the person who installs the application on his Android device. The user provides various inputs like username, password, and contact numbers etc. and triggers various events on the application.
- ii. Front End – This is the part of the application that is visible to the user. A screen presented to the user is usually an Activity, Fragment or a Dialog Box. They contain various elements like text box or buttons to take inputs from and provide outputs to the user.
- iii. Logic – These are the java files that contain the logic of the application. They contain various methods and classes that meet the functional requirements of the application. These files also contain code to communicate with other components in the application.

iv. Services – This is the component of the application that is typically used to perform long background tasks that do not have a user interface. For example – a service is used to track the location of the device at every fixed interval of time.

v. Receivers – This is the component of the application that typically listens for some events or responses from other services. For example – A receiver is used to fetch the location co-ordinates from the location service and then add this location to the database for future references.

vi. Location Manager – It is used to fetch the location of the Android device. The app uses both the GPS provider and the network provider to find the location for the device. GPS provides more accurate data about the location but usually takes sufficient time to start up after the connection is relinquished. Network provider on the other hand are quicker but the accuracy is lesser than GPS.

vii. Contacts Manager – A system service that provides the contact to use so that the user can select a contact that is already present in the contact book. When the user clicks on the number text box to enter a number it opens up the contact book application. If the user selects a contact and if that contact has a number associated, it is sent to the SOS application and displayed in these text boxes.

# Proposed System

To develop a system for android users for keeping track through several applications. This application uses GPS for identifying the location of the person in trouble and the system can be divided into two modules:

First module can be the victim's phone i.e. the root device which uses 3G/2G data connection for tracking the location of the victim through GPS.

Second module can be the mobile phone of registered contacts either police or friends or family members which receives the message containing URL of location of victim that is sent from the root device.

# Working flow of Shakti:

- i. The first time a user opens the app on his device, he must login by entering a username and password. He then remains logged into the application until he explicitly logs out.
- ii. If the user does not have an account, he can register on the login screen.
- iii. The user can also choose the password reset option in case he does not remember his password. A new password is set for the user and a mail containing this new password is sent to the registered email id.

iv. Once logged in, the user is directed to the main screen of the application. This is the screen that would open up when the user opens the application. The user can press the panic button to send text messages and emails to the contacts set up, he can also send an 'I am OK' message to these contacts by clicking on the OK button. The user can also call 100/102/1091 directly from within the application by pressing the police/ambulance/women's helpline button. In order to avoid unnecessary and accidental 7 press of these buttons, the user has the option to enable and disable these buttons.

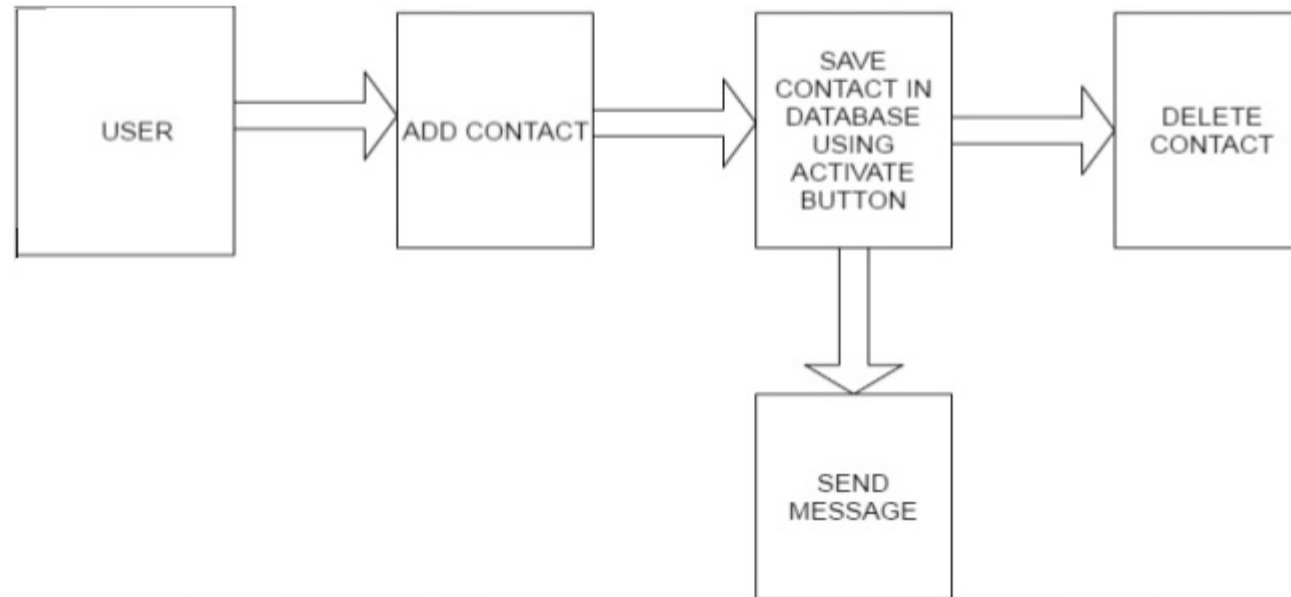
v. The user will also see his current location on the map screen. This way he would know his exact location and refer to it in case he makes a call to 100. This location is also sent as a part of the text..

vi. The user can set the contacts to send the text message within the app. He can either select the contact from the contact book or can enter one manually. He can also set the text message that would be sent.

vii. The user can enable the option to start location tracking. If this option is selected, the application fetches the location of the device (about every 15 minutes) and stores it in an external database.

viii. If the permission to track the location was granted, he can see the latitude and longitude of your present location.

## Flow Diagram:

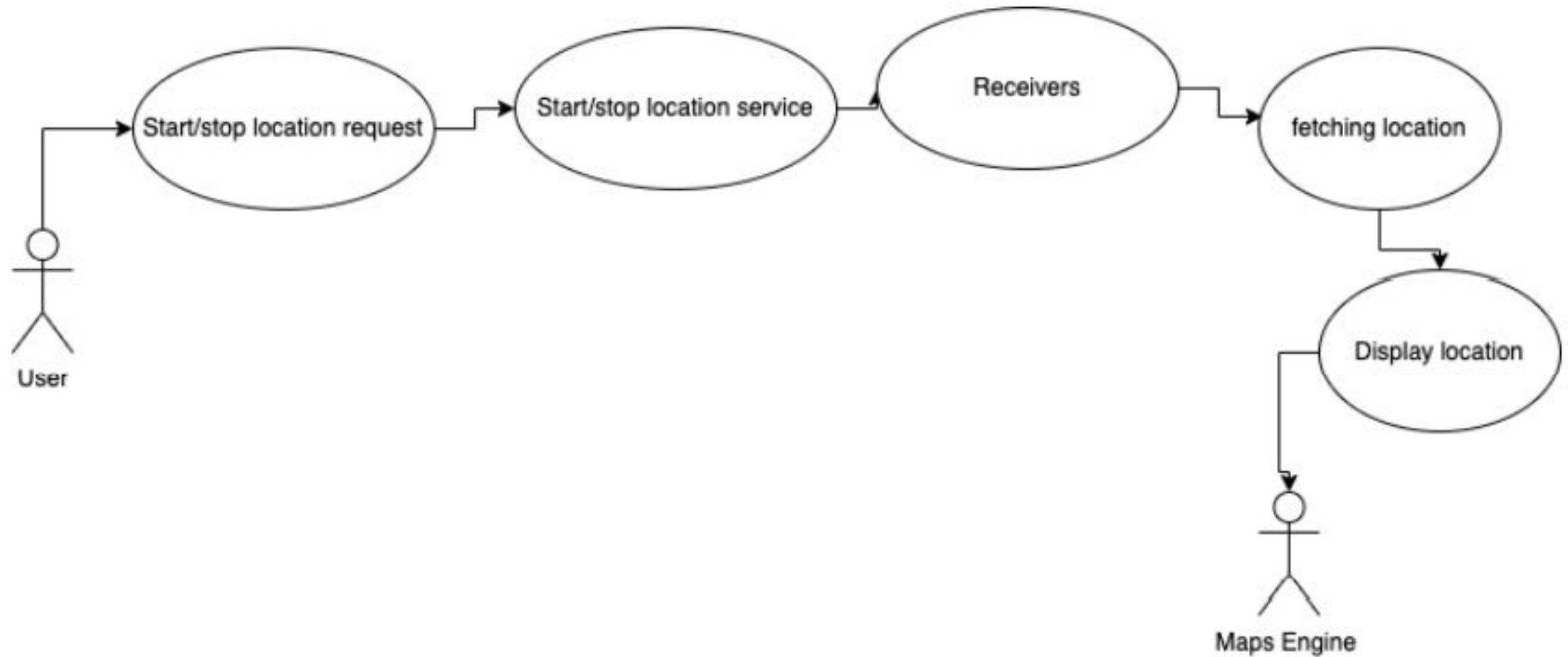


**Week6**

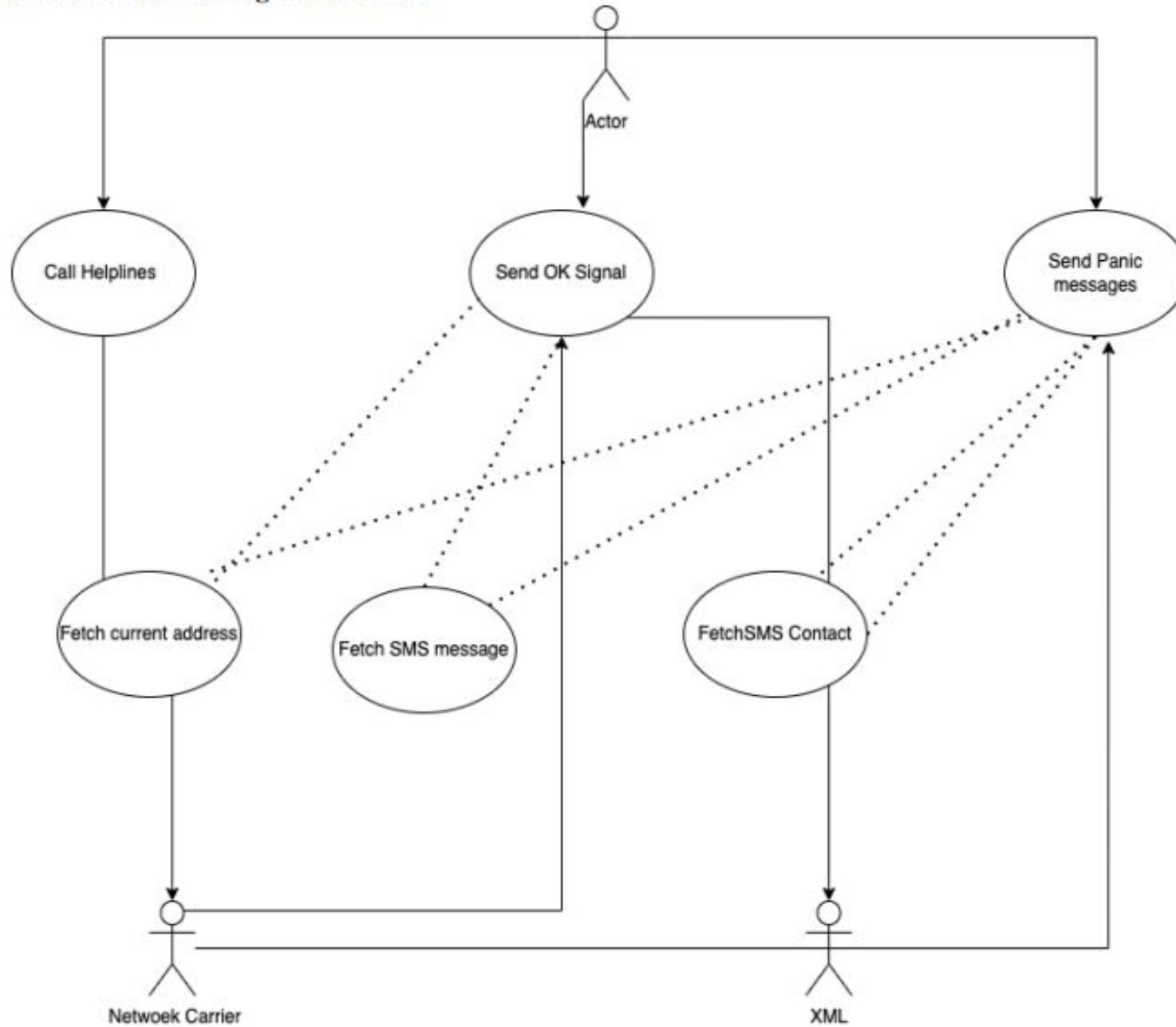
# **USE CASE DIAGRAMS**



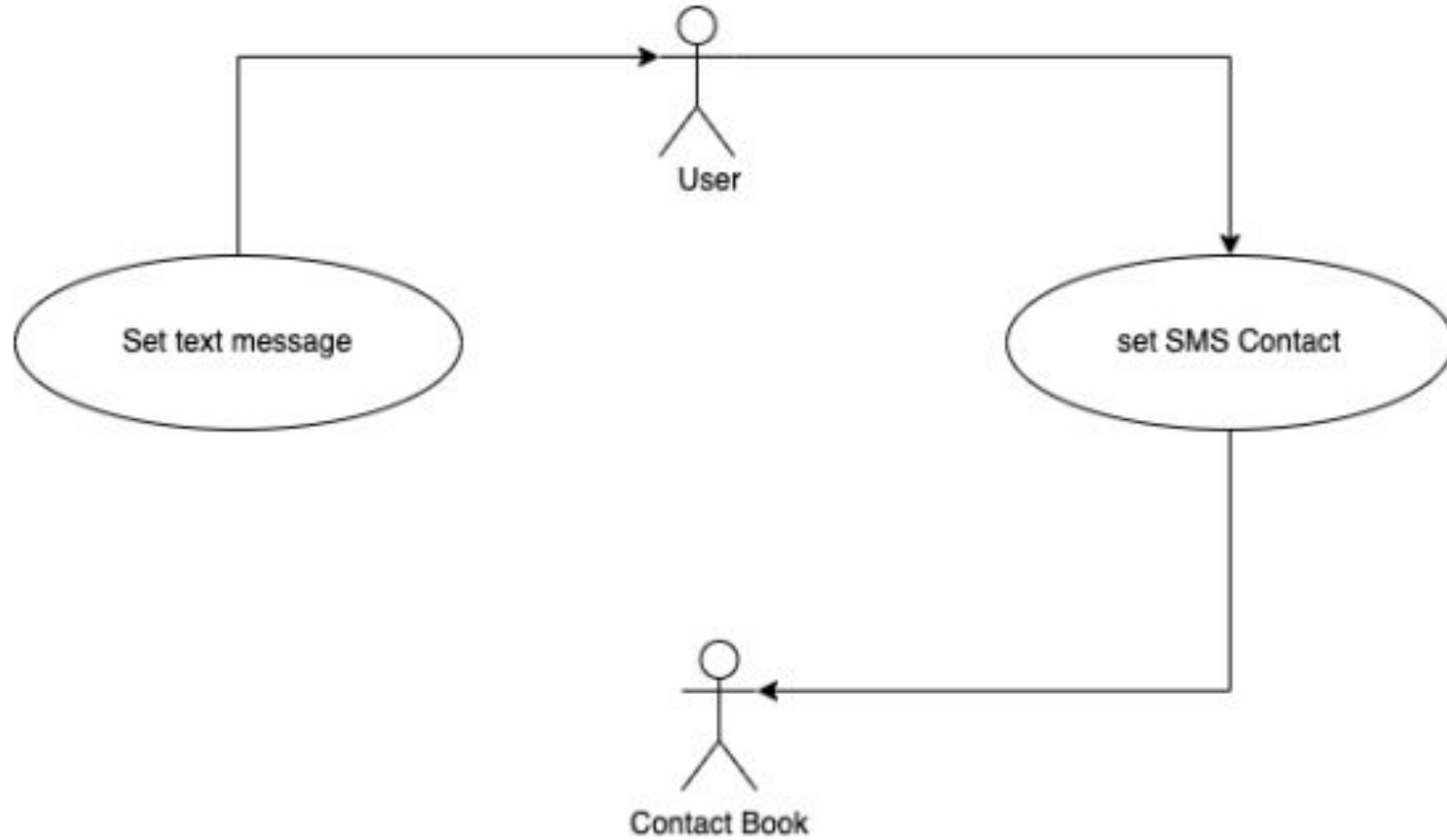
## Use case for location tracking and fetching location history



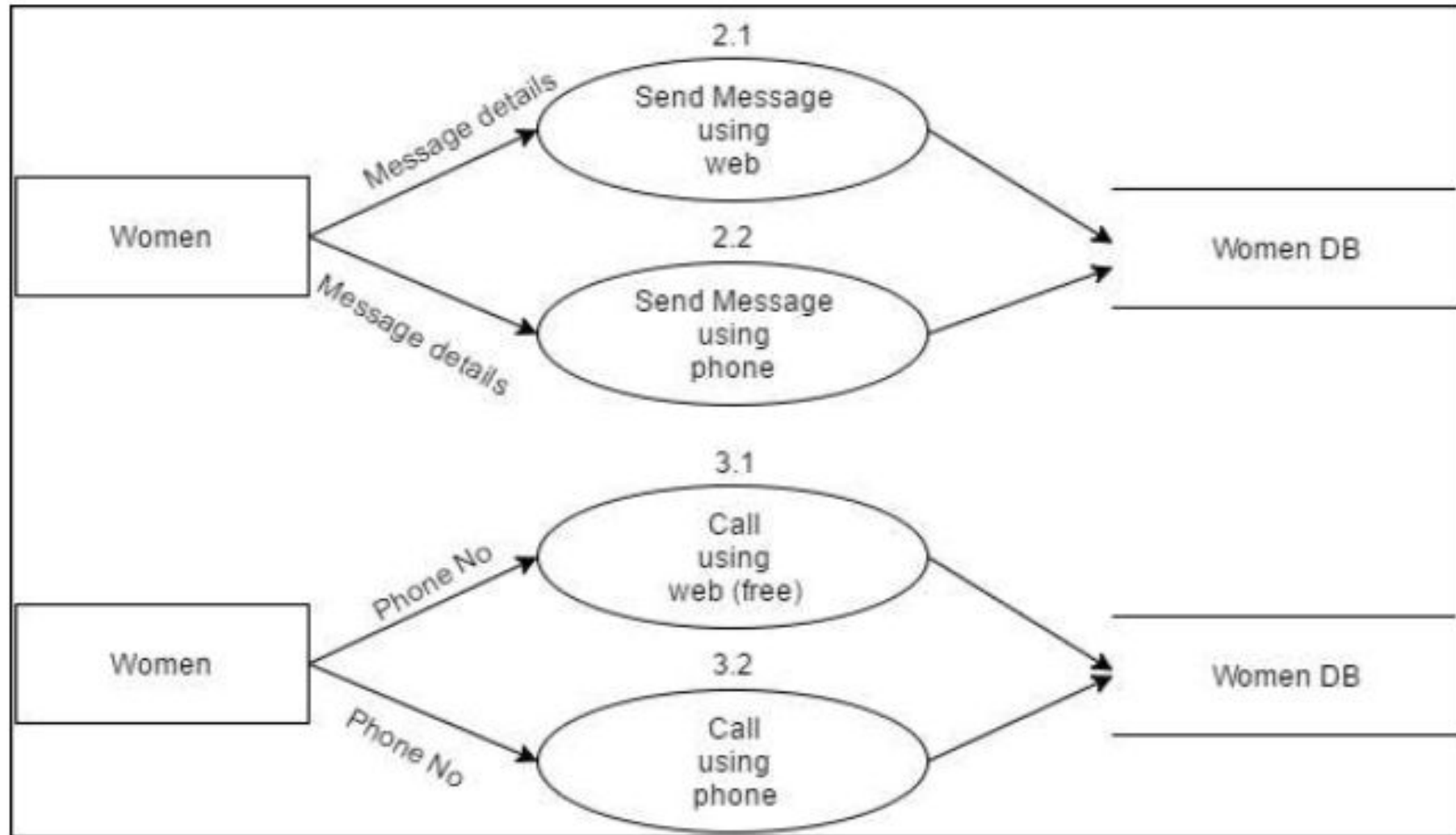
## 2. Use case for sending notifications



### 3. Use case for setting contacts



## Module Frontend & Backend Diagram



# Week7

## Implementation:

The app is a collection of Activities and Fragments that are presented to the user. These Activities and Fragments have associated XML files (Layouts) declared in the layout folder which determine the graphical interface for these components. The SOS app also contains other Service and Broadcast Receivers along with the declarations and necessary permissions in the Manifest file in the root directory of the project.

### Activities

An Activity is the component of an Android app that is presented to the user and is responsible for interacting with them. The Activity may cover the entire screen of the device or may only cover a part of the screen displaying on top of another Activity. An Android app is a collection of loosely coupled activities along with other resources where one Activity can call another Activity at any point usually using Intents.

An Activity implements a number of callback methods that are invoked by different events during the lifecycle of an Activity. The following diagram shows the various callback methods for an Activity:

# Manifest File

```
<uses-permission android:name="android.permission.INTERNET" />
<uses-permission android:name="android.permission.SEND_SMS" />
<uses-permission android:name="android.permission.RECEIVE_SMS" />
<uses-permission android:name="android.permission.CALL_PHONE" />
<uses-permission android:name="android.permission.ACCESS_FINE_LOCATION" />
<uses-permission android:name="android.permission.ACCESS_COARSE_LOCATION" />
<uses-permission android:name="android.permission.ACCESS_BACKGROUND_LOCATION" />
```

## <application

```
    android:allowBackup="true"
    android:icon="@mipmap/ic_launcher"
    android:label="Shuraksha"
    android:roundIcon="@mipmap/ic_launcher_round"
    android:supportsRtl="true"
    android:theme="@style/Theme.Shuraksha">
```

## <activity

```
    android:name=".helpline"
    android:exported="false" />
```

## <activity

```
    android:name=".maps_page_2"
    android:exported="false" />
```

## <activity

```
    android:name=".MapsPage"
    android:exported="false" />
```

## <activity

```
    android:name=".criminate"
    android:exported="false" />
```

## <activity

```
    android:name=".messages"
    android:exported="false" />
```

## <activity android:name=".register" />

## <activity android:name=".honeact" />

## <activity android:name=".login">

## <intent-filter>

```
    <action android:name="android.intent.action.MAIN" />
```

```
    <category android:name="android.intent.category.LAUNCHER" />
```

## </intent-filter>

## </activity>

## <meta-data

```
    android:name="com.google.android.gms.version"
    android:value="12451000" />
```

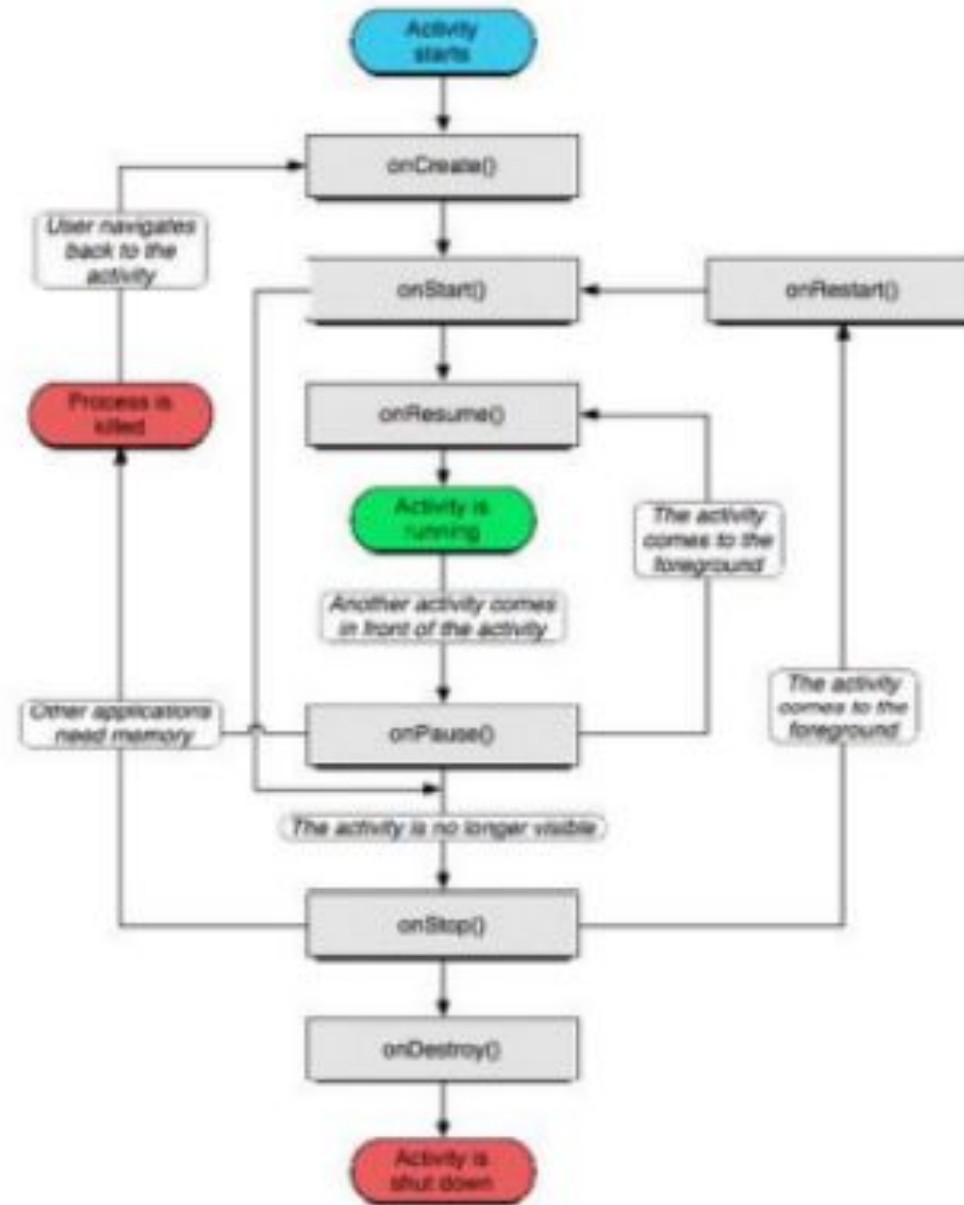
## <meta-data

```
    android:name="com.google.android.geo.API_KEY"
    android:value="AIzaSy8hm_51C8X0rulfCY10sFaq0o5NFTyD10Q" />
```

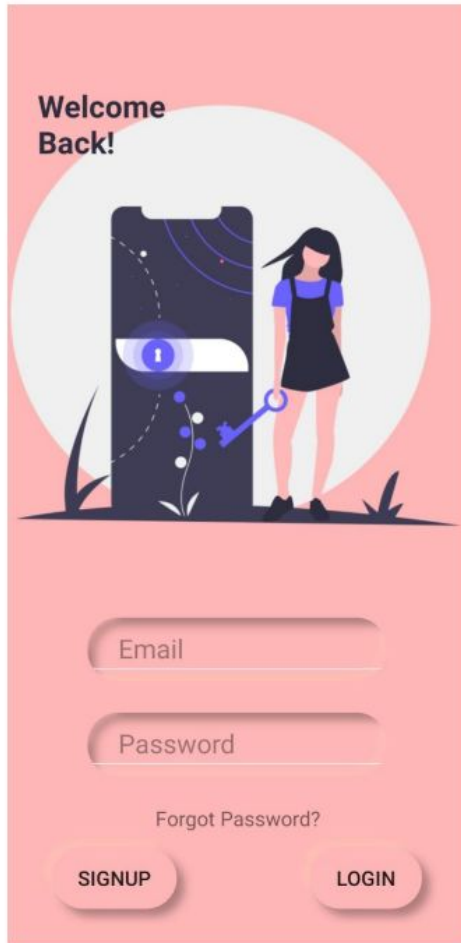
## </application>

## </manifest>

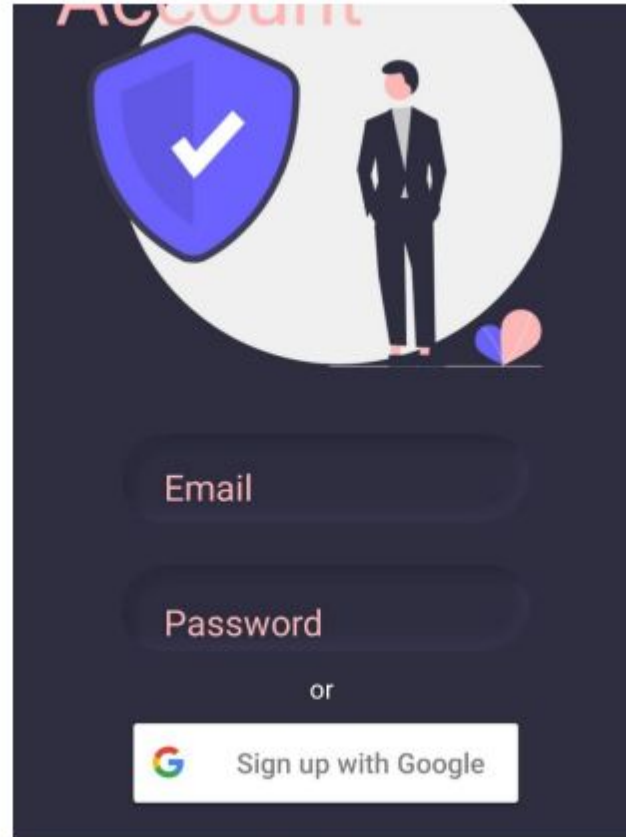
# Activity Cycle



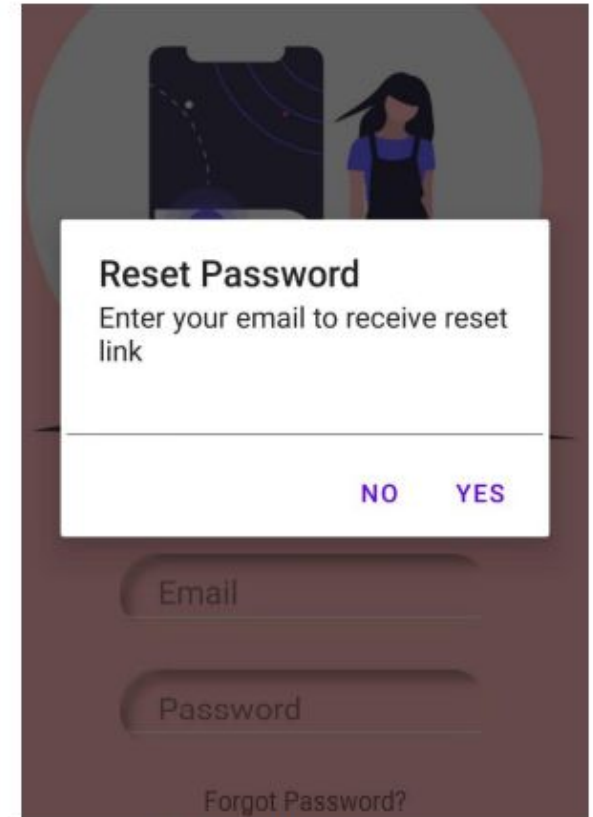
# Graphical User Interface



Login Screen



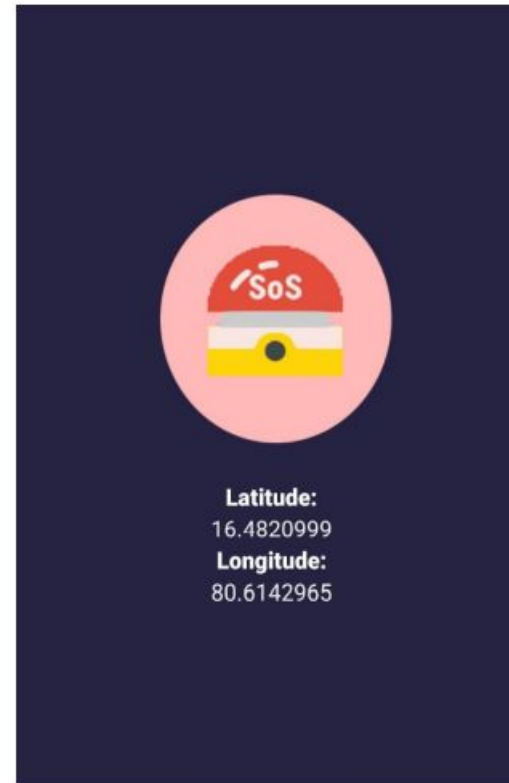
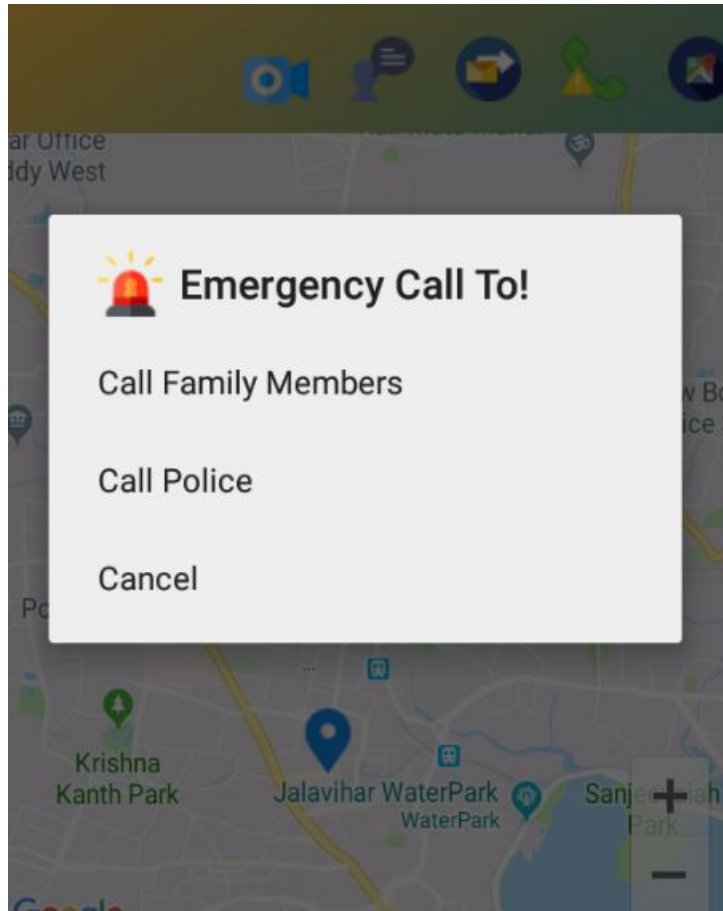
Register screen



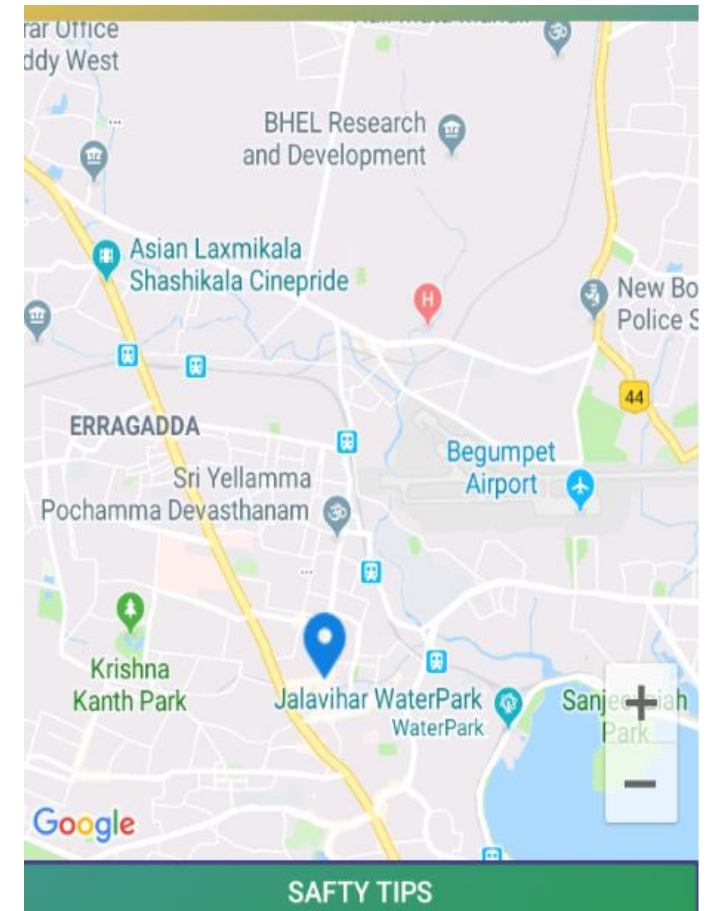
Password reset screen



## Graphical User Interface



Location records screen



# Week8

## Testing Software

Testing is an essential phase in the development life cycle of an application. Testing ensures that the developed system meets its functional and non-functional requirements. Two important terms in software testing are Verification and Validation. Verification is the process of evaluating work-products like requirement specs, design specs and test cases etc. of different development phases to make sure that they meet the requirements for that phase. It ensures that the system is built in the right way. Whereas Validation is the process of evaluating the software at the end of the development phase to make sure that it meets the business requirements. It is used to make sure that the product fulfills its intended use and that the end product is built right. In this chapter we mainly validate the app to make sure it meets the requirements set initially.

## **Unit Testing**

It is a strategy in software testing where individual components in a software are tested for correctness. In the SOS apps, these components are the Activities that are presented to the user as screens on the Android device, Fragments, Services and Receivers.

## **Performance testing**

It is a type of non-functional testing performed to determine how fast the system can perform under certain workload. In the SOS app performance testing is done to make sure that there are no significant lags in the user interface while using the application due to background tasks etc. Android SDK provide a graphical tool called Trace view to profile the performance of the application.

## **Integration testing**

It is a strategy in software testing where different modules are combined and test to make sure they work together correctly. It is done when the components are unit test and the main objective is to test the interfaces between different components

# Future Work

The current work on the Shakti app has a lot of essential features that would be used in case of an emergency situation like sending text messages , and making calls to 100/102/1091 from within the app on tap of a single button. An app for such a purpose has a lot of scope for enhancement. In the future the app may include features like –

- i. A home screen widget that can be used as a triggering point to send panic notifications. A user would then not have to open the app to send these panic notifications.
- ii. ii. Initiating a call to a number set from within the application when the user presses the panic button.
- iii. iii. The app can also listen to incoming messages from the set contacts. If these message have a pre-defined text like “UPDATE LOCATION” the app can reply with a text message containing the current location or for some other text like “AUDIO” in which case the app can record a short audio and send it as an email to the person. This is very helpful as you may have already pressed the panic button and may be in some trouble where you cannot reply. This way the person can track you constantly and also understand something about the nature of the emergency from the audio clip.
- iv. Setting up a password to stop the application