PROJECT REPORT

ON

Emergency Help and Safety System

(Helping Hands)

B.Tech (CE) Sem-VI

In the Subject of

System Design Practice (CE-621)

Rushi Parmar (CE-084) (17CEUBS112)

Ayush Patel (CE-087) (17CEUOG067)

Prajapati Vatsal (CE-103) (17CEUBS090)

Under the Guidance of

Prof. Brijesh S. Bhatt

Department Of Computer Engineering

Faculty Of Technology,

Dharmsinh Desai University, Nadiad.

DHARMSINH DESAI UNIVERSITY

College Road, NADIAD-387001 (Gujarat)

CERTIFICATE

This is to certify that the term work carried out in the subject of **System Design Practice** and recorded in this report is bonafide work of

Mr. Rushi Parmar (Roll No.: 084, Id: 17CEUBS112) ,

Mr. Ayush Patel (Roll No.: 087, Id: 17CEUOG067)

and

Mr.Vatsal Prajapati (Roll No.: 103, Id: 17CEUBS090)

of **B.Tech Semester 6th** in the branch of Computer Engineering during the academic year 2019-20.

Prof. Brijesh S. Bhatt
(Project Guide and Associate Professor)
Faculty of Technology,
Dharmsinh Desai University,
Nadiad.

Dr. C.K. Bhensdadia
Head of CE Dept.,
Faculty of Technology,
Dharmsinh Desai University,
Nadiad.

Contents

Abstract

Modern societies are confronted with an increasing number of abnormal events, crises, disasters and accident. This type of threats can make a person in emergency.

To overcome these situation, there arise a need for Emergency Help and Safety System. Helping Hands is built for the same reason. Helping Hands is an android application which increases the chance of help in case of emergency by notifying nearby volunteers.

1. Introduction

1.1 Introduction of Emergency Help and Safety System

Increasing crime rate in today's society and unpredictable routine leads to a very insecure daily life, especially for those living alone and away from their family. There must be a handy application so that users can help each other in need. Helping hands does the same work through which users can get help from complete strangers.

Features of Emergency Help and Safety System :-

- Users can contact government emergency services(i.e. police/medical/fire services)
- Users can see available volunteers within the range of current location.
- In case of emergency user can trigger sos alert which will notify user selected contacts and live location of user will be sent to them every 5 minutes.
- User can broadcast help request which will notify nearby volunteers and volunteers can accept the help request.
- User Account Management is also supported by system.

1.2 Technology/Platform/Tools Used

Android Studio :

Android Studio is the official integrated development environment for Google's Android operating system, built on JetBrains' IntelliJ IDEA software and designed specifically for Android development.

Java :

Java is a general-purpose programming language that is class-based, object-oriented, and designed to have as few implementation dependencies as possible.

Xml :

XML is a markup language much like HTML used to describe data. Xml as itself is well readable both by human and machine. It is scalable and simple to develop. In Android XML is used for designing layouts as xml is lightweight language.

Cloud Firestore (by Google):

Cloud Firestore is a flexible, scalable database for mobile, web, and server development from Firebase and Google Cloud Platform. It keeps your data in sync across client apps through realtime listeners and offers offline support for mobile.

Google Maps Platform:

The Google Maps Platform is a set of APIs and SDKs that allows developers to embed Google Maps into mobile apps and web pages, or to retrieve data from Google Maps.

2. Software Requirement Specifications

Emergency Help and Safety System :-

R.1 Managing User Profile

R.1.1 Create User Profile

Description: Users can register them self.

Input: Basic Personal Details

Output: Generated user profile

R.1.2 Deactivate User Profile

Description: Users can deactivate their profile.

Input: User selection

Output: Success message

R.1.3 View User Profile

Description: User can view their profile.

Input: User selection

Output: Appropriate User Profile

R.1.4 Update User Profile

Description: User can update basic profile details.

Input: User Data

Output: Success message and show updated profile

R.2 System forms and Displays GeoLocation Map

R.2.1 System Displays User's Current Location

Description: Show current Location of user in Map

Input: Access Device's Location

Output: Pinned Point Location on Map

R.2.2 System Gets nearby Volunteer's Location

Description: Show Location of Available Volunteer in Map

Input: User Selection

Output: Display Location of Volunteer in Map

R.2.3 System shows nearby Emergency requests of other users

Description: Show Location of nearby incoming request

Input: User Selection

Output: Display Location of requests

R.3 System generates and broadcasts Emergency Help Requests

R.3.1 Generate Help Signals

Description: System Generates Emergency Signals

Input: User selection or Gestures

Output: Emergency Signal data generated on User's Devices

R.3.2 Broadcast Help Signals

R.3.2.1 Broadcast Critical Emergency

Description: System Broadcasts Emergency request to nearby Volunteers and user selected contacts

Input: User Selection

Output: Emergency data sent to nearby volunteers and Emergency Contacts

R.3.2.2 Broadcast Intermediate level Emergency

Description: System Broadcasts Intermediate level Emergency signals and notify user selected contacts

Input: User Selection

Output: Emergency data sent to nearby volunteers and Emergency Contacts

R.3.3 Contact Government Services

Description: System contacts Government Emergency Services

Input: User Selection

Output: Appropriate Government Services are alerted

R.4 System allows Users to respond to Incoming Requests

R.4.1 Receive Emergency Requests

Description: Emergency requests are received and displayed

Input: Incoming emergency request

Output: Display respected request and appropriate message

R.4.2 Accept/Reject incoming Requests

Description: User is given option to accept or reject the request

Input: User selection

Output: Appropriate message is sent

R.4.3 Track Accepted Request

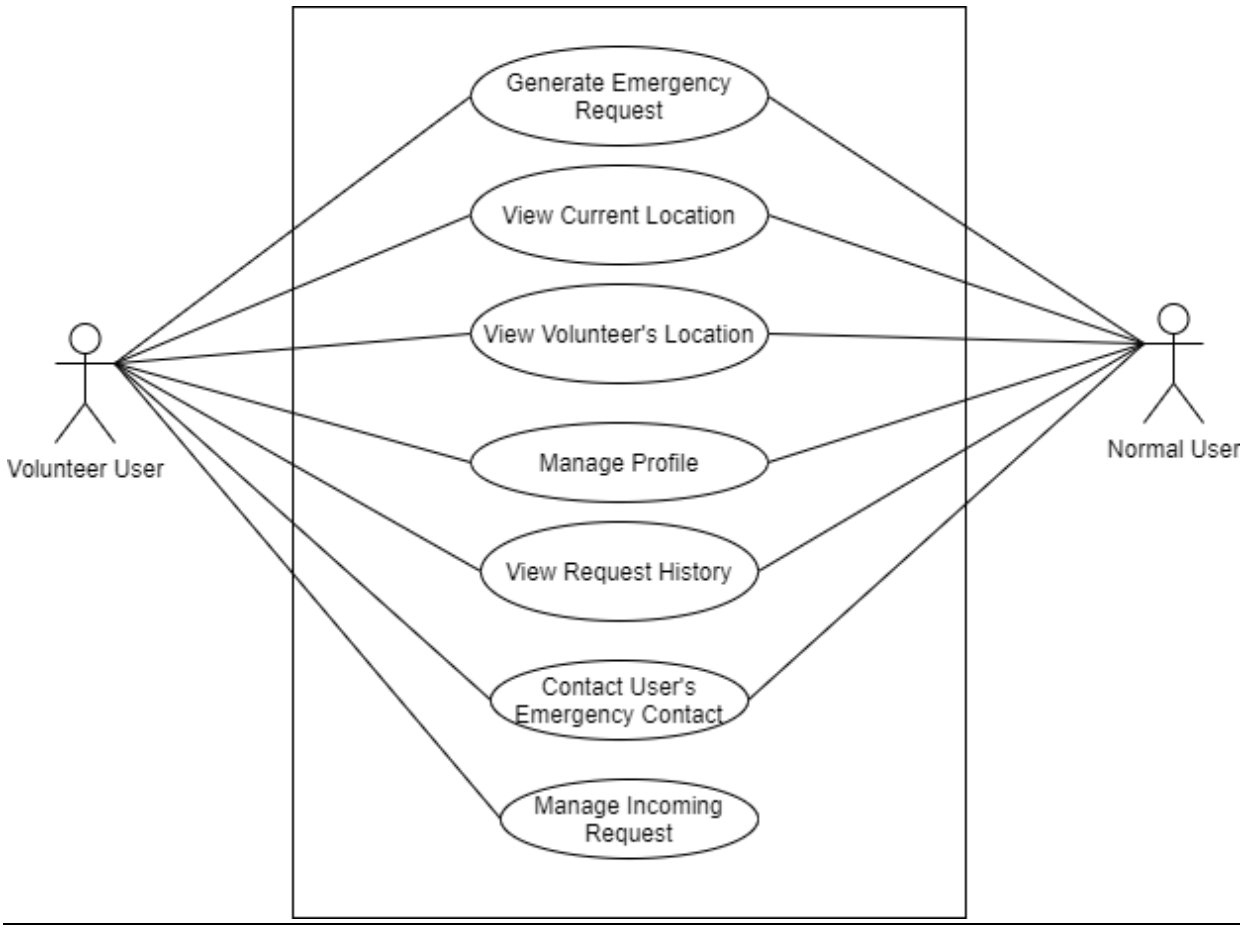
Description: System tracks the requests that are accepted by any volunteer

Input: Acceptance of request

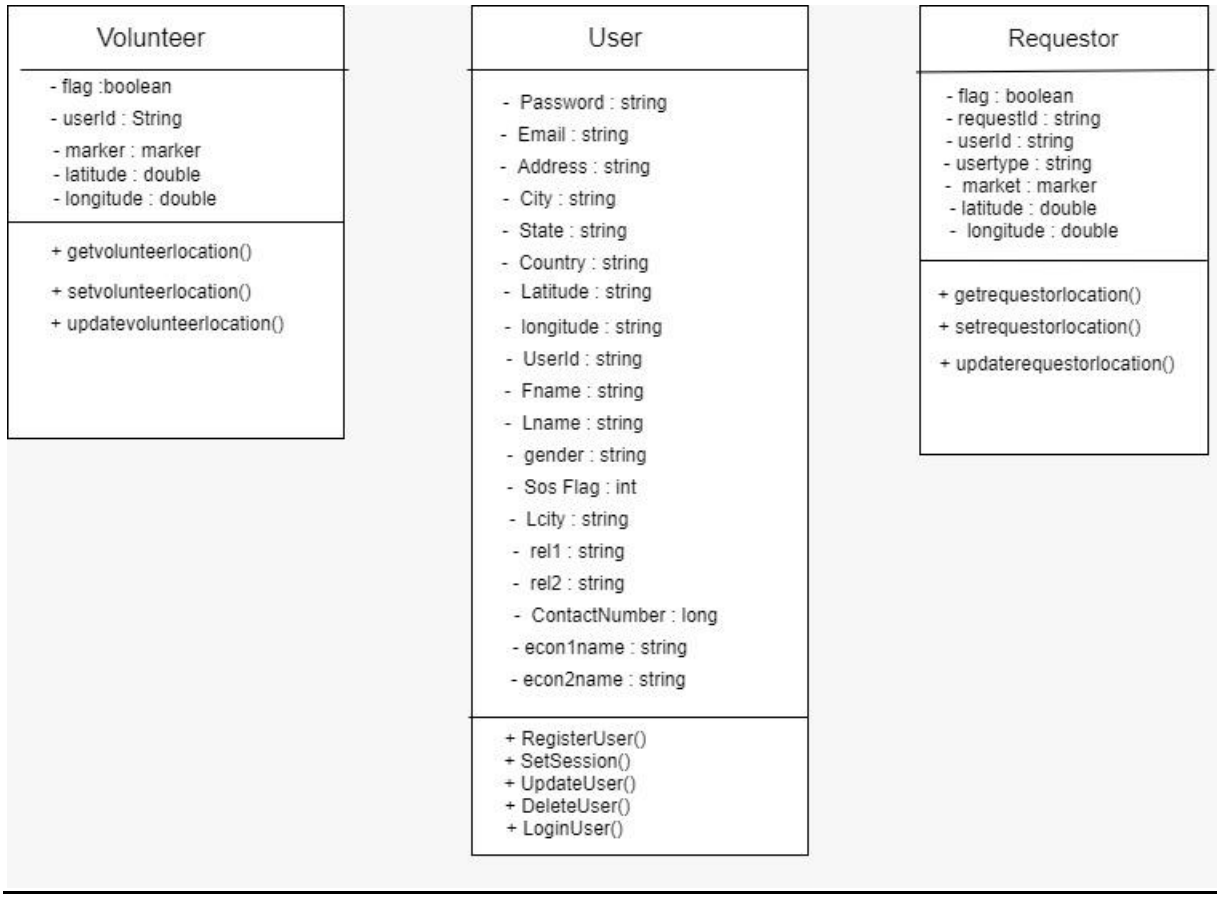
Output: Display track of the volunteer

3. Designs

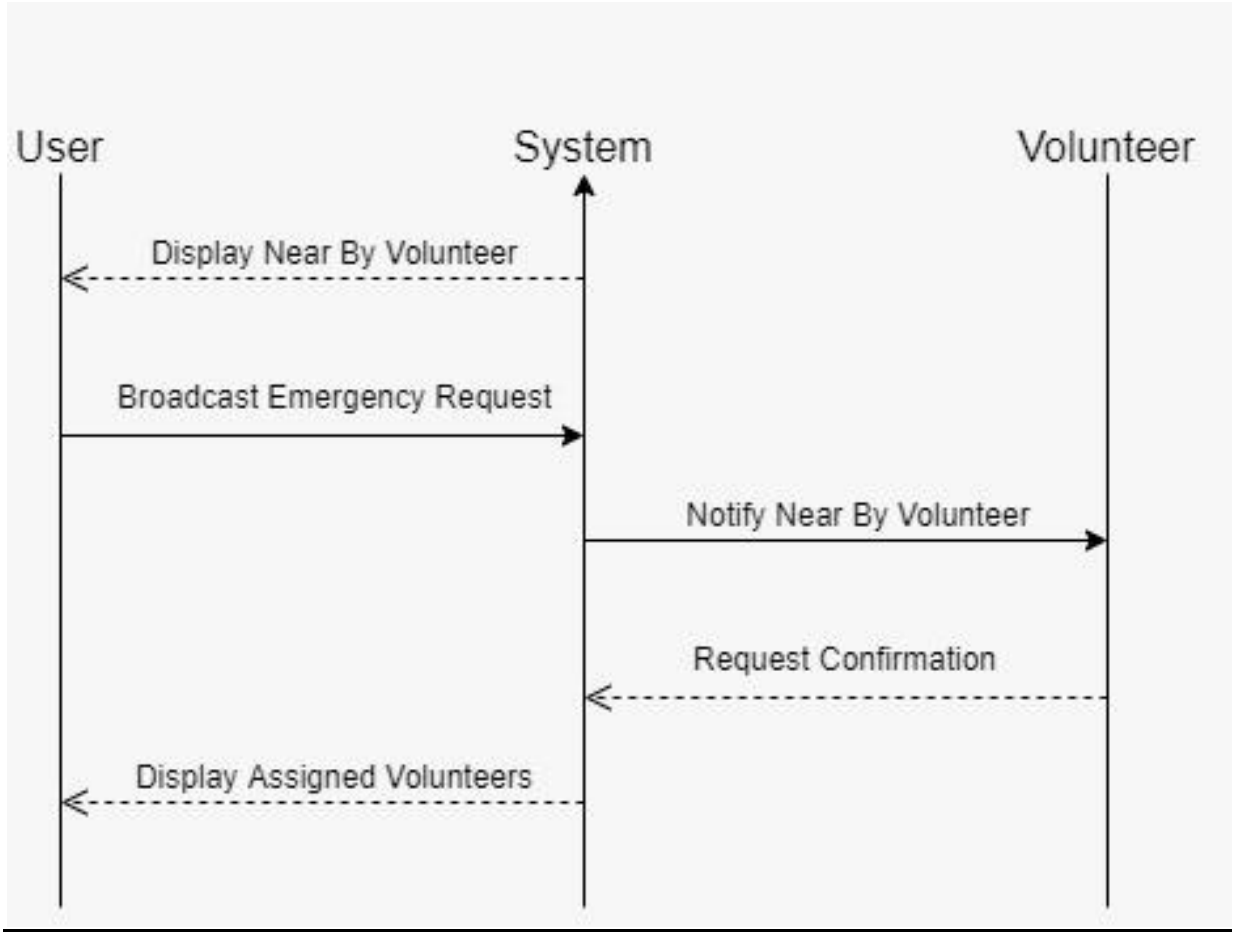
1. Usecase Diagram :-



2. Class Diagram :-

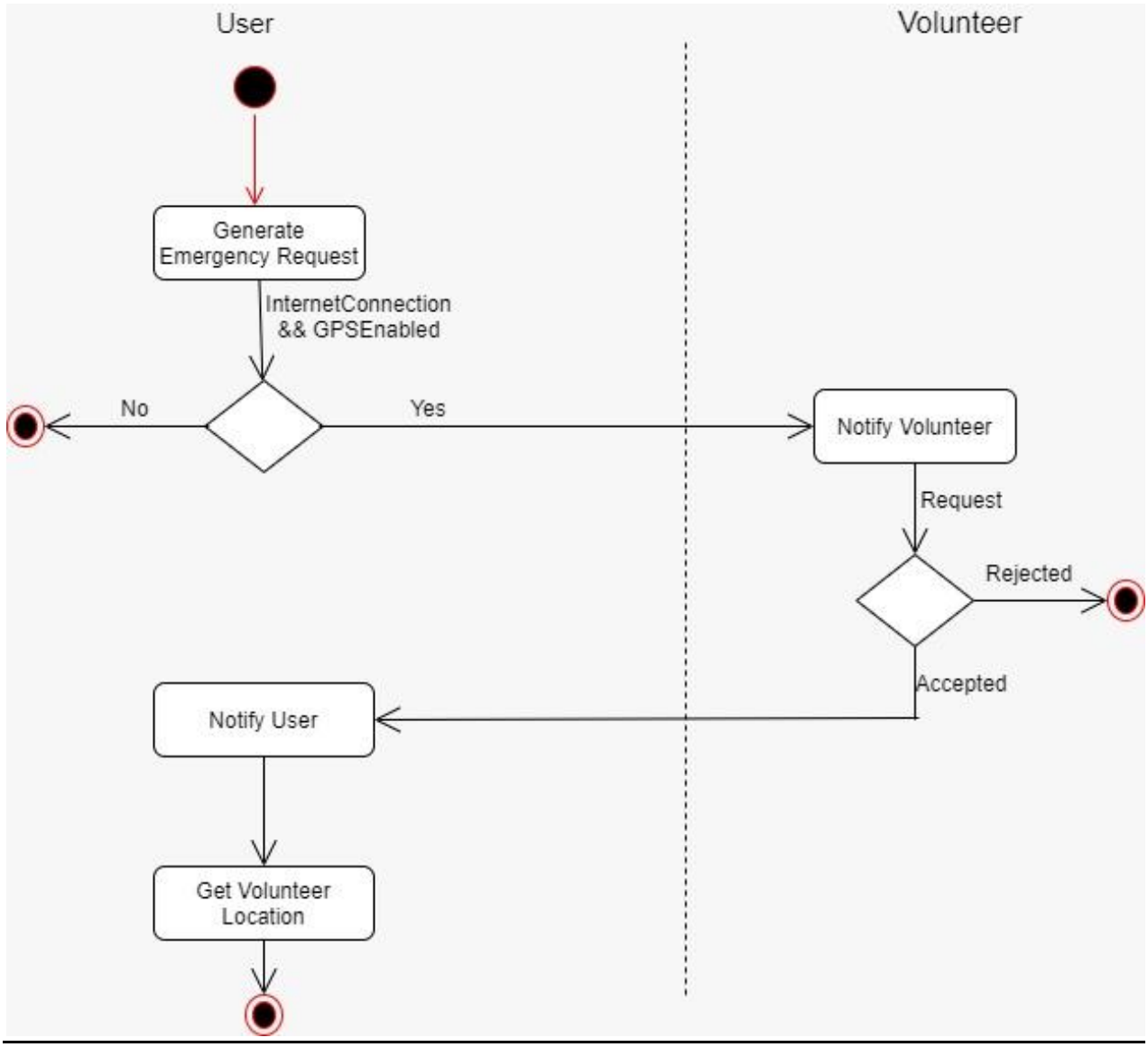


3. Sequence Diagram :-

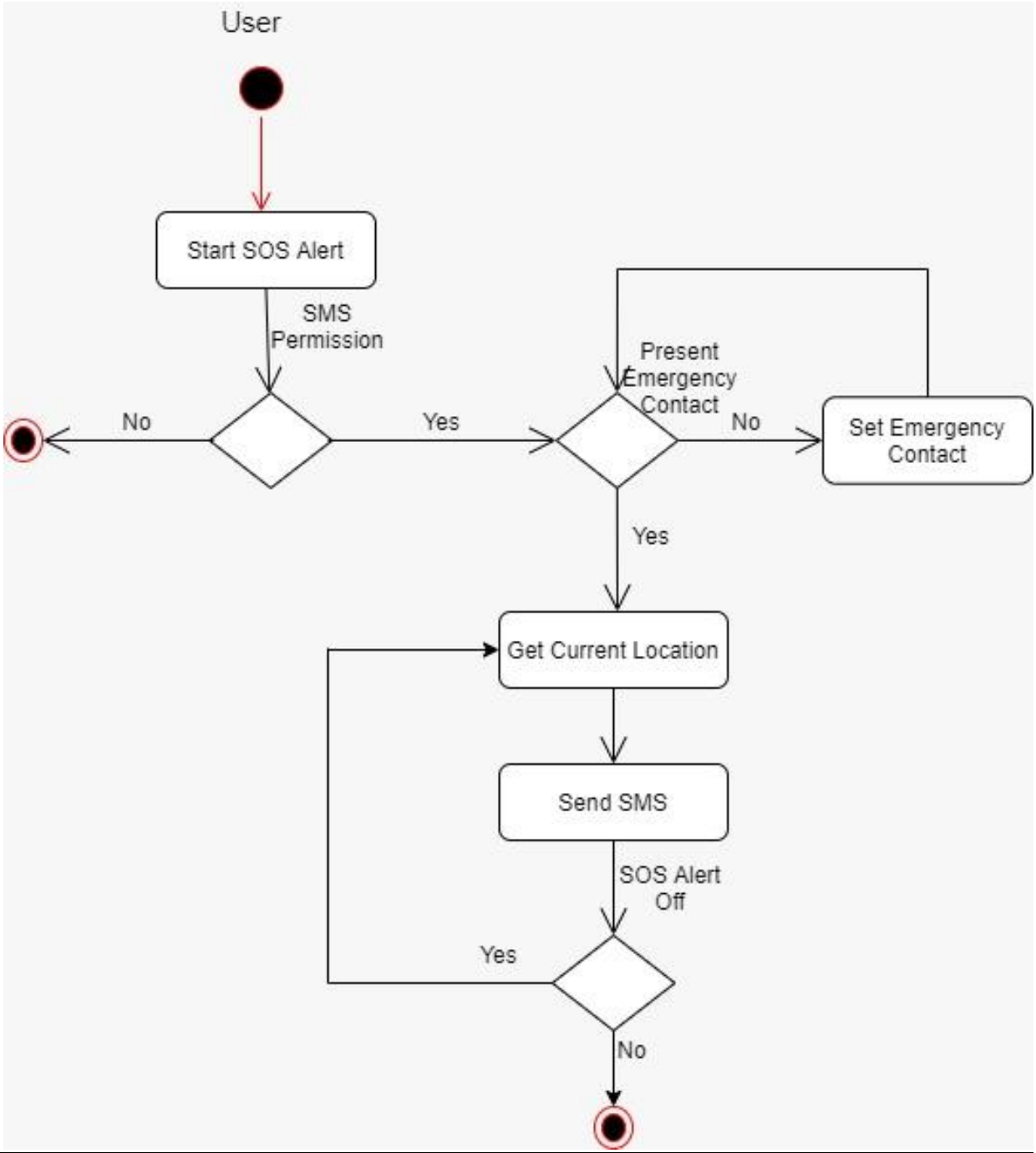


4. Activity Diagram :-

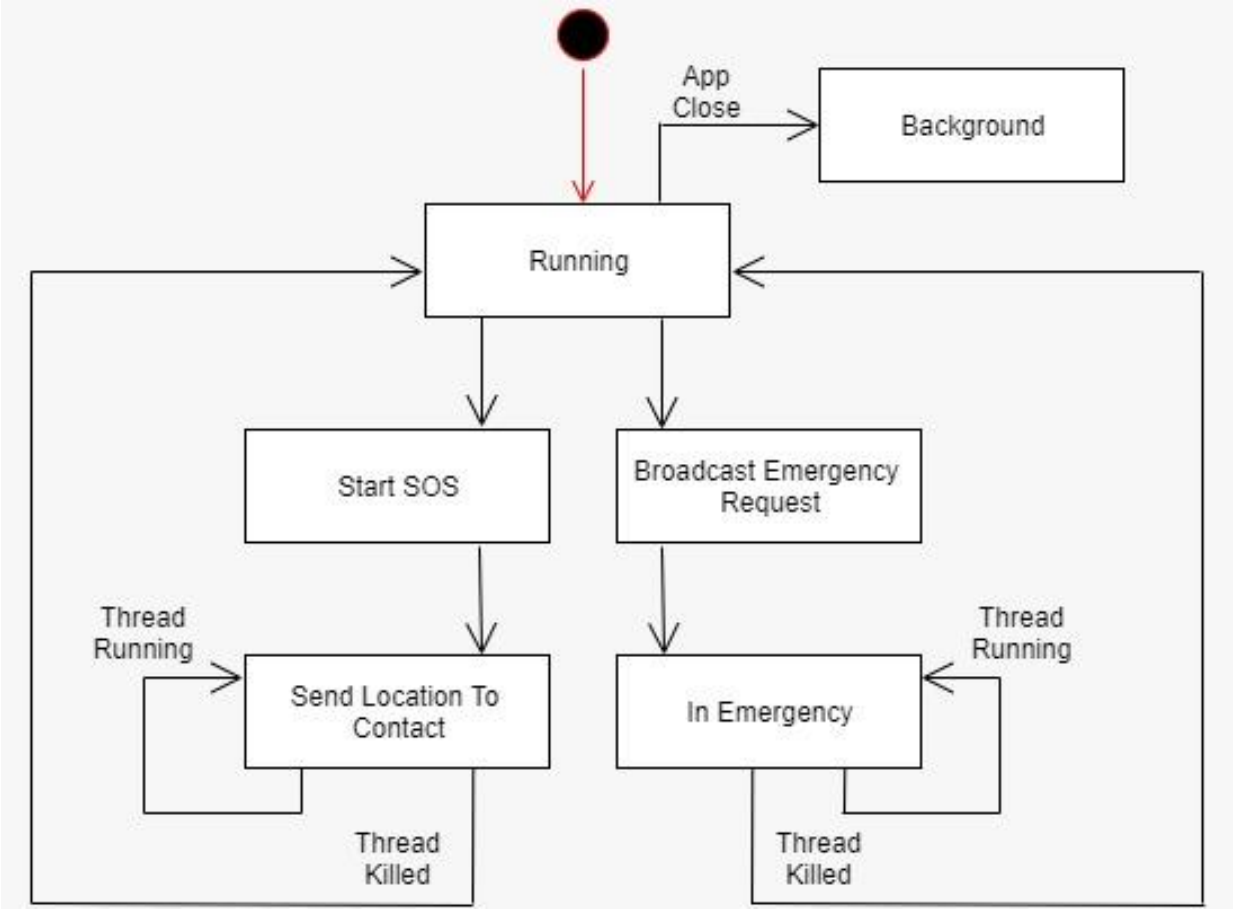
1. Start SOS Alert :-



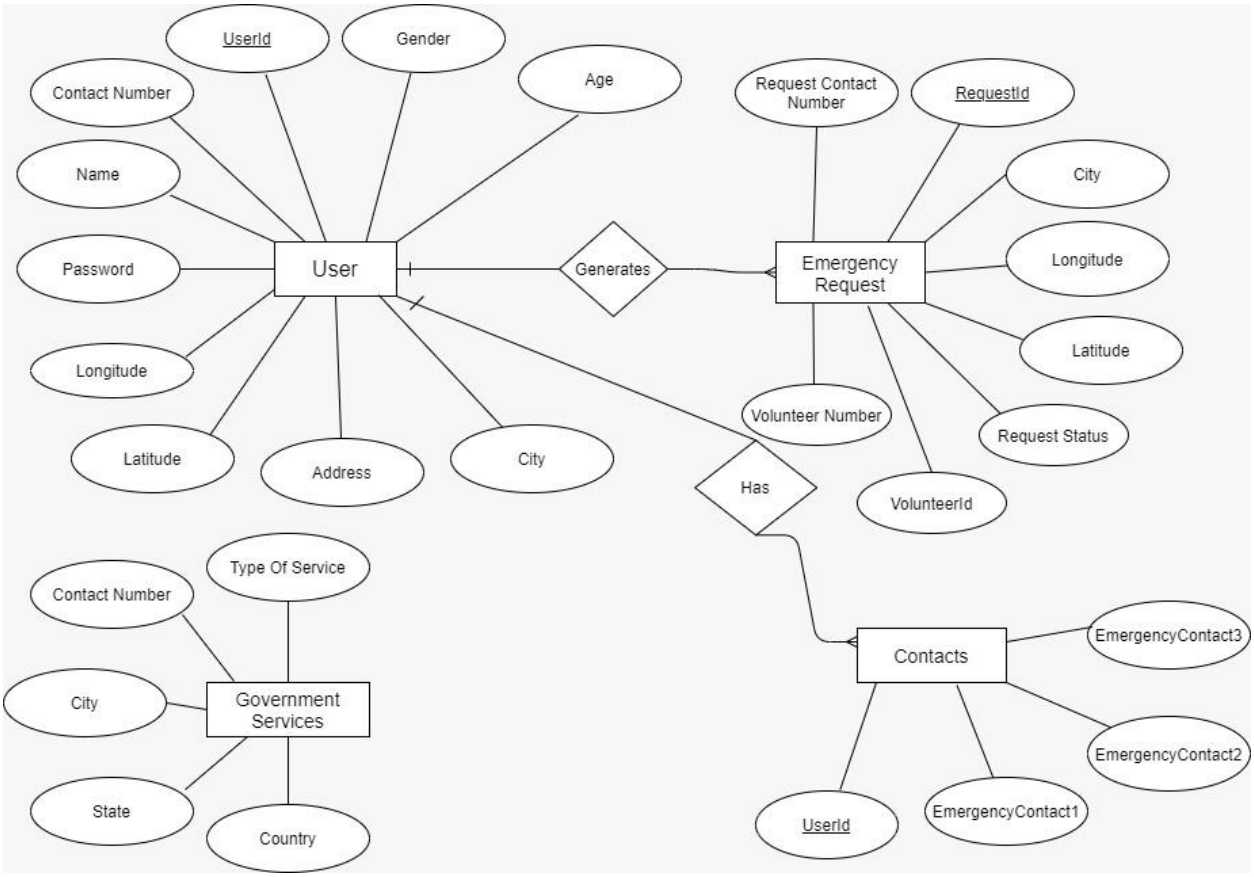
2. Broadcast Emergency Request :-



5. State Diagram :-



6. ER Diagram :-


UserId
Gender
Age
Contact Number
Name
Password
User
Longitude
Latitude
Address
City
Generates
Request Contact Number
RequestId
City
Longitude
Latitude
Request Status
Emergency Request
Volunteer Number
VolunteerId
Has
Contacts
EmergencyContact3
EmergencyContact2
EmergencyContact1
UserId
Type Of Service
Contact Number
City
Government Services
State
Country

7. Data Dictionary :-

User Details

Name of Field	Datatype	Length
UserId	Varchar2	10
Contact Number	Number	10
Username	Varchar2	20
Password	Varchar2	20
IsVolunteer	Boolean	1
Latitude	Double	10
Longitude	Double	10
LCity	Varchar2	10
Age	Number	3
Gender	Varchar2	6
AddressLine1	Varchar2	50
AddressLine2	Varchar2	50
City	Varchar2	15
State	Varchar2	15
Country	Varchar2	15

Contacts

Name of Field	DataType	Length
UserId	Varchar2	10
EmergencyContact1	Number	10
EmergencyContact2	Number	10
EmergencyContact3	Number	10

EmergencyRequests

Name Of Field	DataType	Length
Service ID	Varchar2	20
Requester Contact Number	Number	10
Type of Emergency	Varchar2	20
Level of Emergency	Varchar2	10
Latitude	Double	10
Longitude	Double	10
Request Status	Varchar2	10
Request Description	Varchar2	20

Government Services

Name Of Field	DataType	Length
Type of Service	Varchar2	20
Contact Number	Number	10
City	Varchar2	15
State	Varchar2	15
Country	Varchar2	15

4. Implementation Details

4.1 Description of Modules :-

1. Register User Module

Basic information of user is taken by system and stored in database.

2. Login User Module

Users are able to login themselves. System logs user in, then and only then user can use other functionalities of system.

3. Map Module

Map Module displays current location of user in the map and it also keeps track of volunteers present within the rage of user location. Also Any emergency request will be displayed on map.

4. Request History Module

All the request that is generated by given user will be displayed with appropriate data in Request history module.

5. Setting Module

Basic settings related to account will be displayed here. This module includes functions like Edit Profile, Change Credentials, Edit Emergency Contacts, Deactivate Account and Logout.

4.2 Function prototype which implements major functionality

Major Functions along with the prototypes according to JAVA language is as shown below:

1. Account Management Functions

User **loginUser**(string Email, string Password);

int **registerUser**(User User);

void **updateUser**(User user, string Email);

void **deactivateUser**(string Email);

2. StartSOSAlert Function

List<String> **getEmergencyContacts**(string UserId);

void **sendSms**(String message,String reciever);

3. Broadcast Emergency Request

Void **generateRequest**(LatLng cur_position, User user);

Void **notifyResult**(Volunteer volunteer,LatLng volunteer_location);

4. Map Functions

LatLng **getCurrentLocation**();

List<Volunteer> **GetVolunteer**(LatLng cur_position);

List<Requestor> **GetRequestor**(LatLng cur_position);

Void **setMarkerPosition**(GoogleMap mMap,Marker marker,LatLng cur_position);

5. Testing

For Testing the System, a mixed approach comprised of Integration Testing and Regression Testing is used.

Every module is tested using Integration Testing where we test the unit parts and then combine those unit parts to form a module.

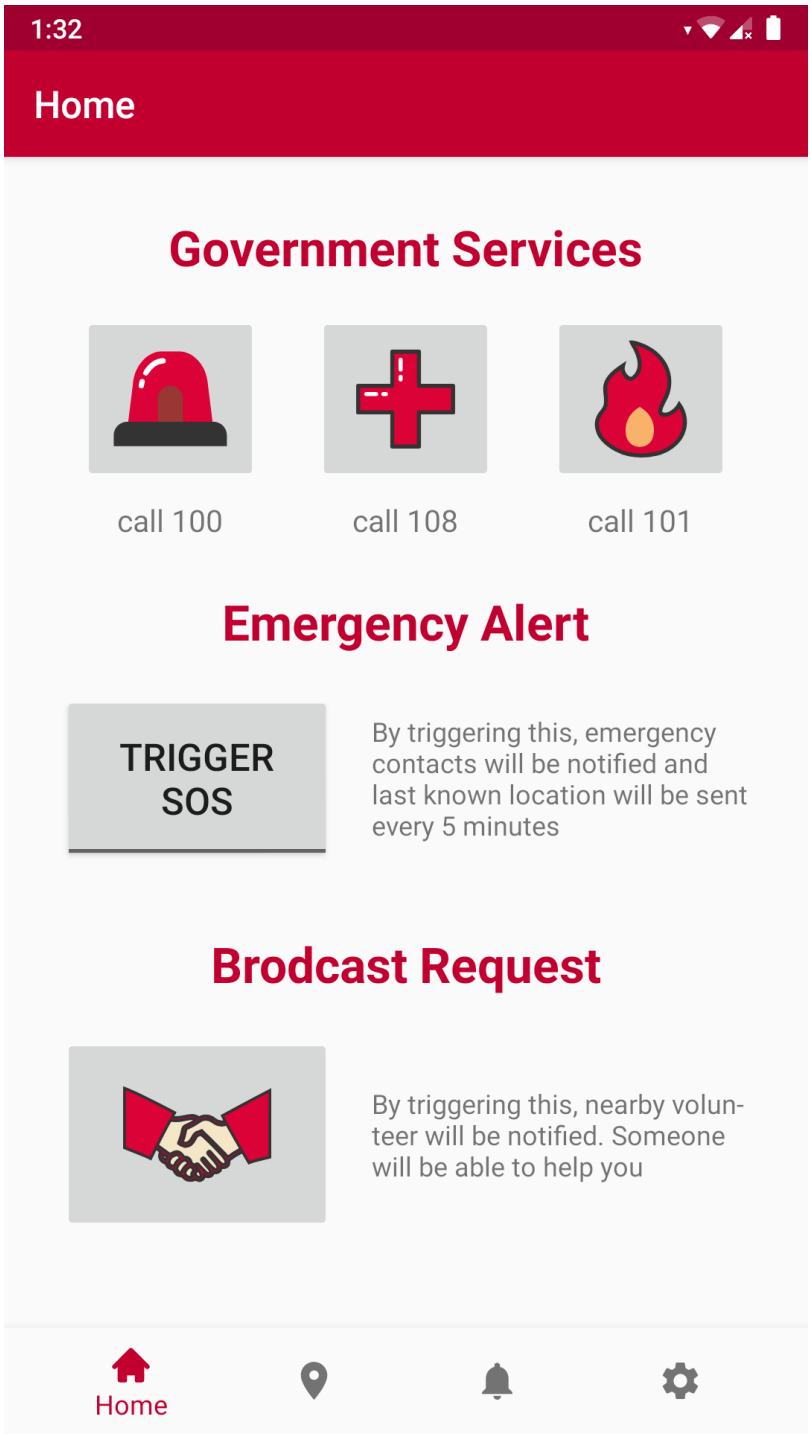
For ex. For creating Map Module, first basic functions like getCurrentLocation, getVolunteers and getRequestor are tested, then the Map Module is Created and Tested if it works properly.

After any module is created, now by the rules of Regression Testing, this module is added to the whole system and then whole system is tested and it is made sure that the whole system works as desired after adding module to system.

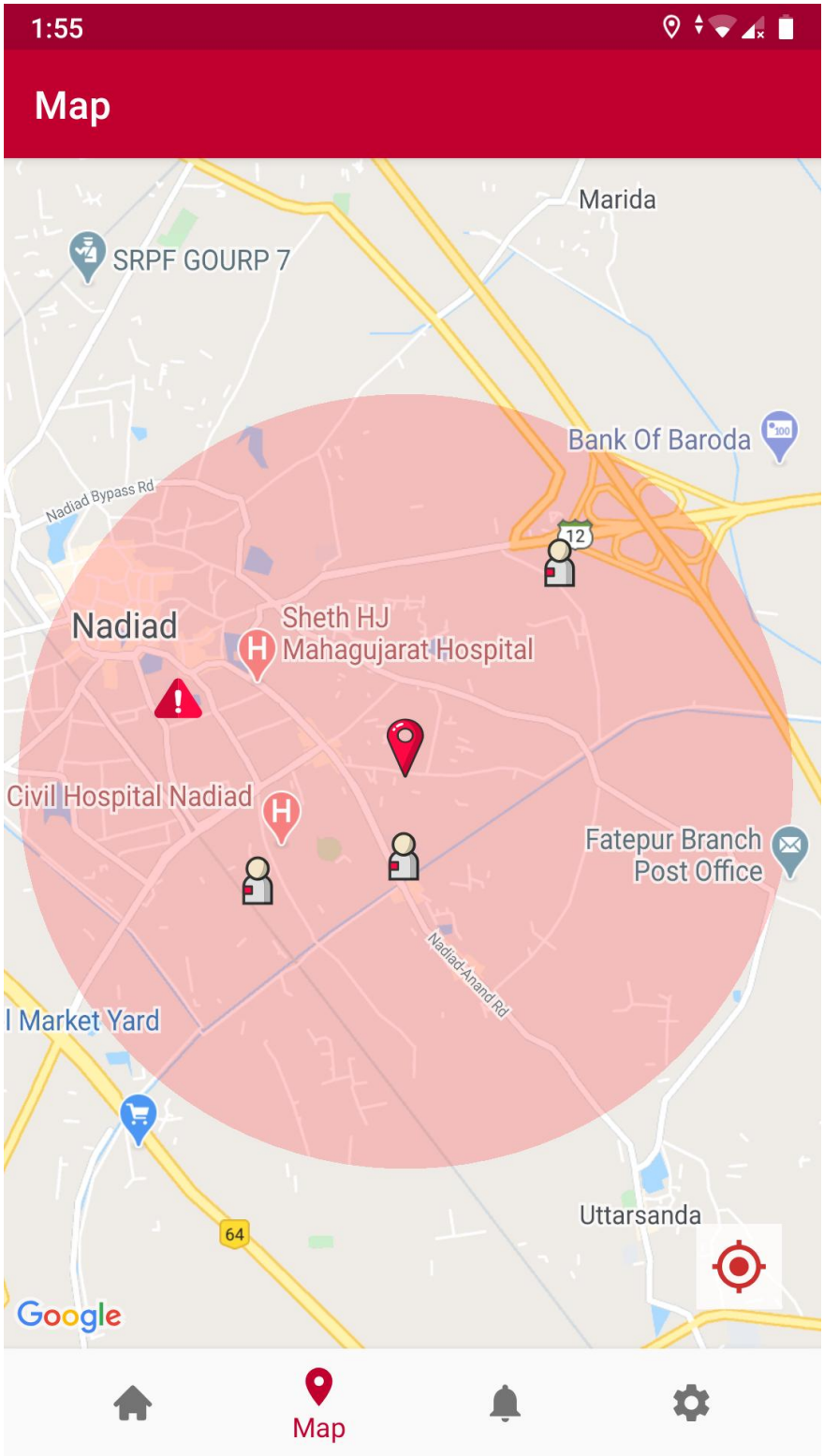
For ex. After successfully creating Map Module, it is added to System and then whole current System is tested to ensure proper working.

6. Screenshots

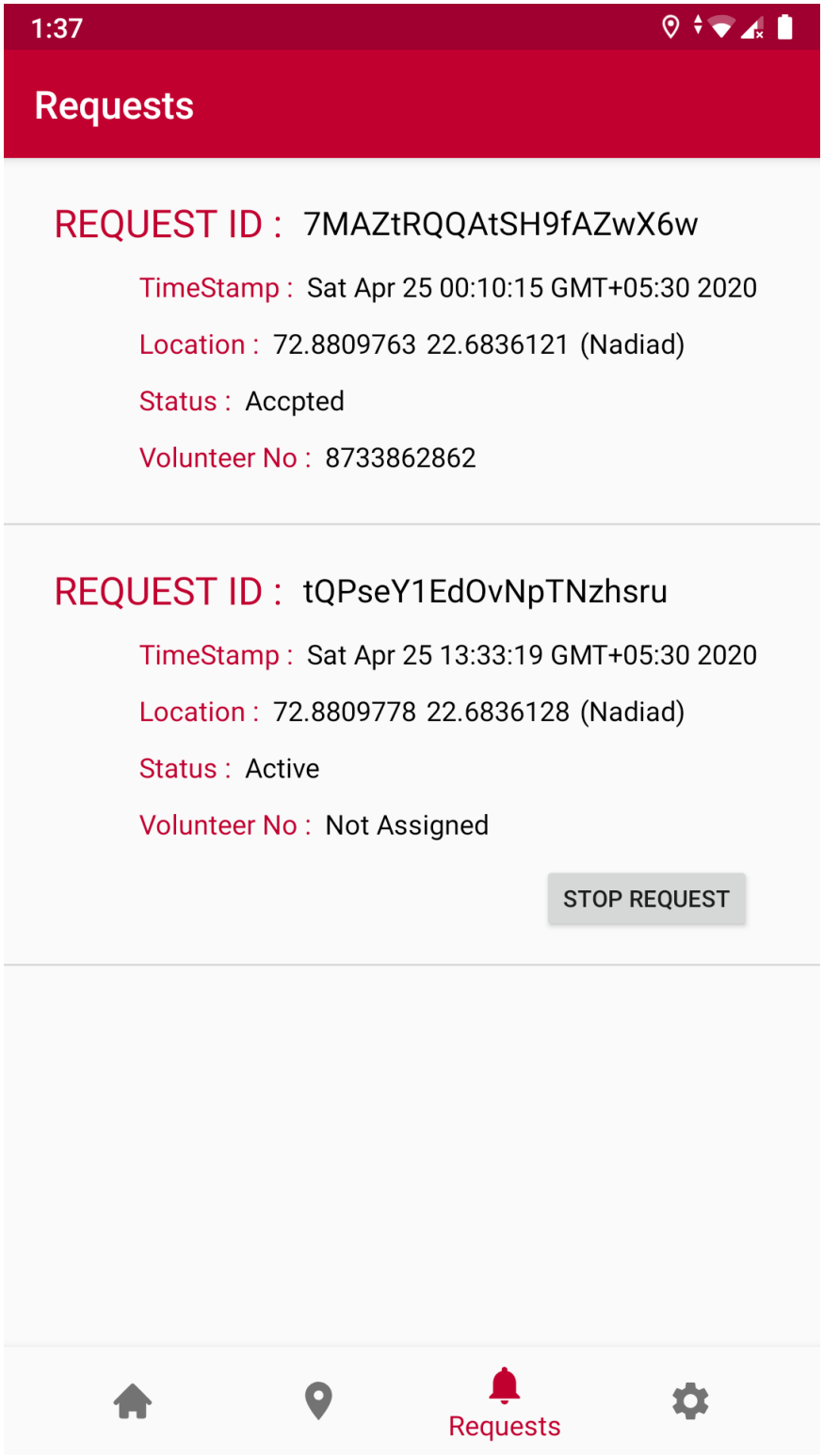
1. Home Panel :-



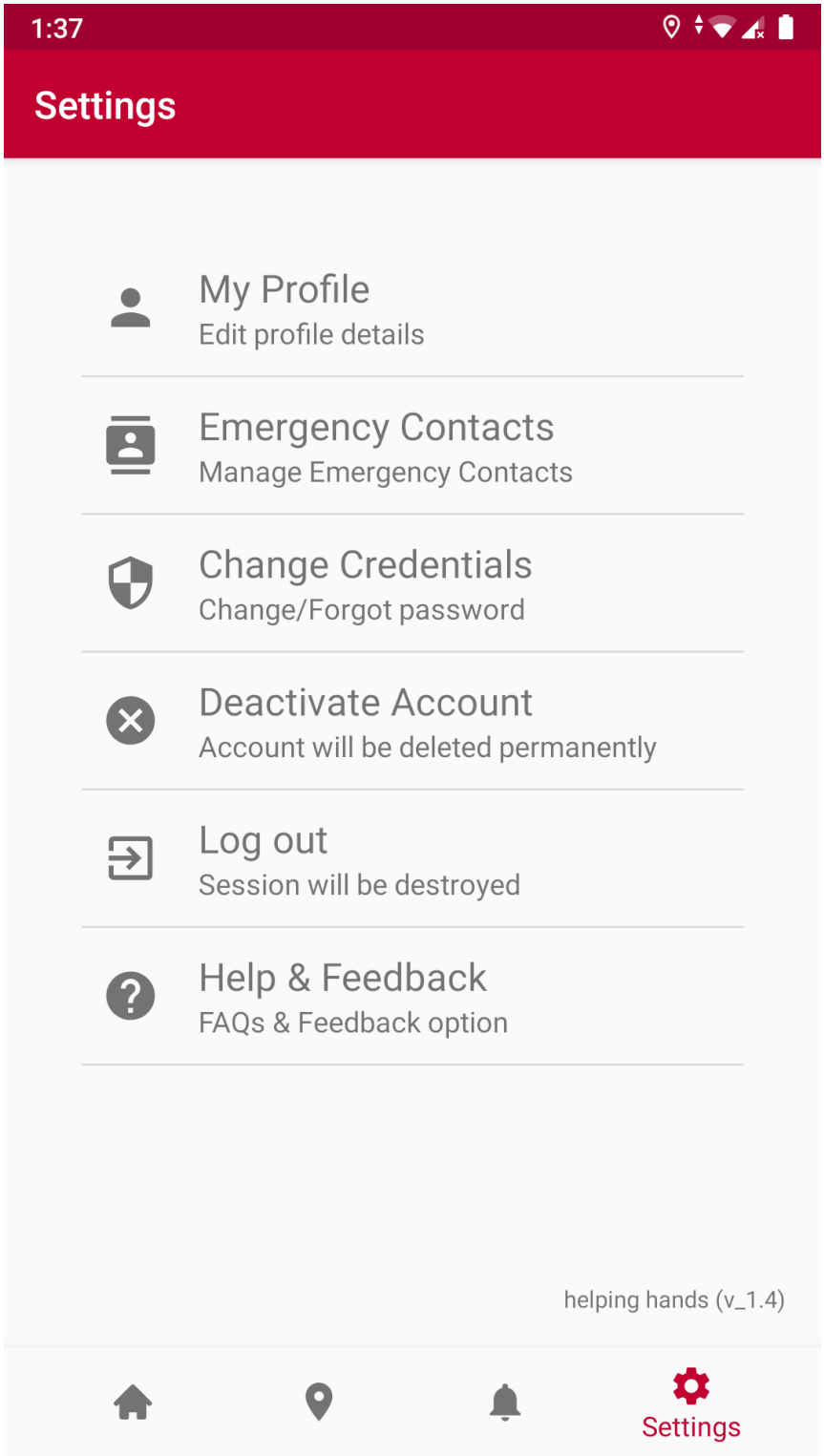
2. Map Panel :-


1:55
Map
Marida
SRPF GOURP 7
Bank Of Baroda
Nadiad Bypass Rd
12
Nadiad
Sheth HJ
Mahagujarat Hospital
Civil Hospital Nadiad
Fatepur Branch
Post Office
Nadiad-Anand Rd
l Market Yard
64
Uttarsanda
Google
Map

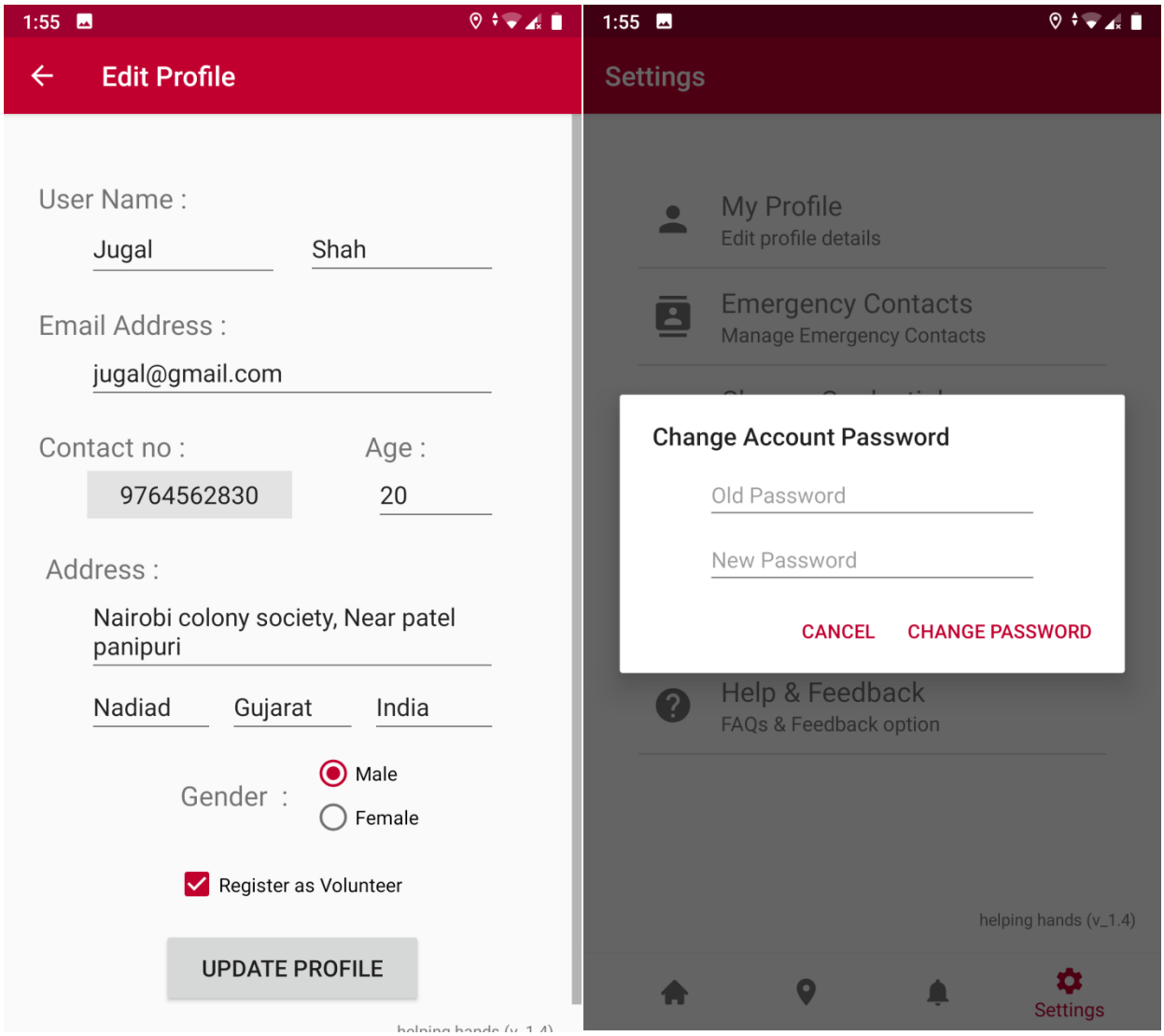
3. Request history :-



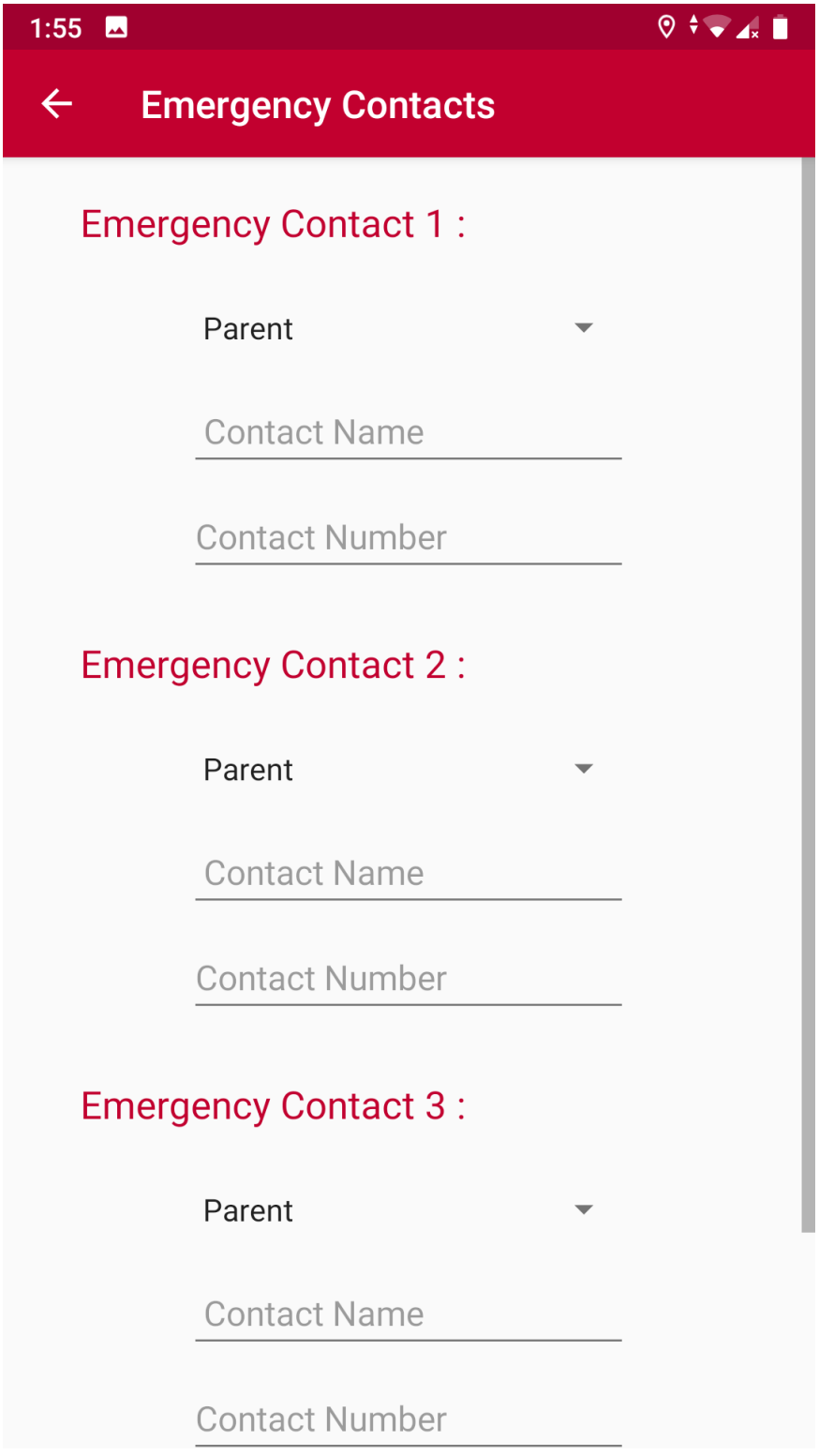
4. Setting Panel :-



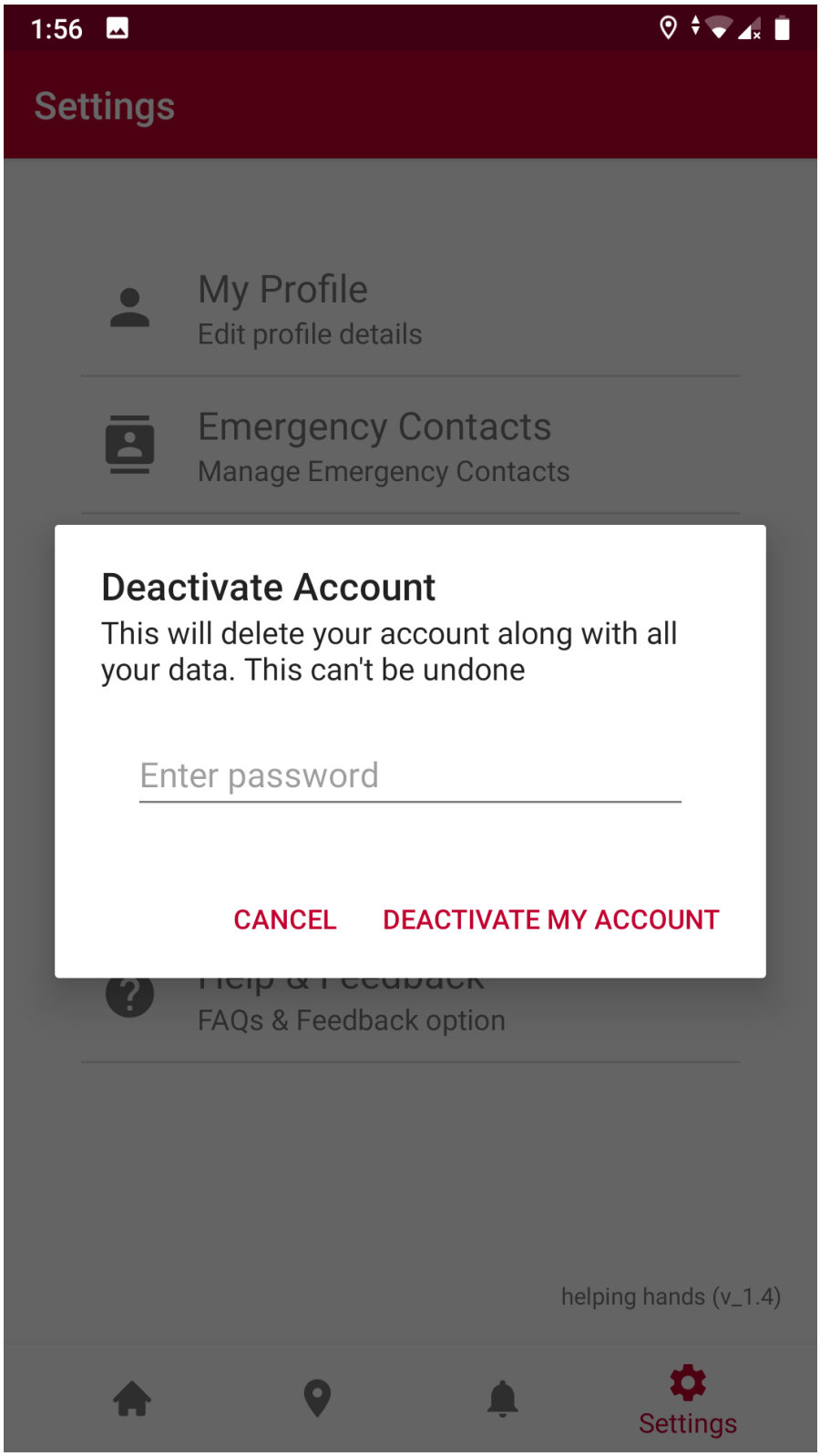
5. Manage Account :-



6. Add/Edit Emergency contacts :-



7. Deactivate Account :-



7. Conclusion

By the successful deployment of Helping Hands, team of volunteers will be ready to help those in need. At the time of emergency, users will be able to broadcast emergency signals, application would send information to the selected contacts of user to get help and the local government services could also be informed. Hence saving many lives and helping the countless, helping hands make sure you get help on time.

8. Limitation and future extension

Limitation:

Background processes can be handled more efficiently. Process load due to concurrent thread execution can be reduced.

Future Extension:

User interface will be improved to provide better interaction with system. Background processes and Thread execution will be efficient and process load will be reduced.

9. Bibliography

Book :

Head First Android Development: by DAVID GRIFFITHS and Dawn Griffiths

Websites :

https://developer.android.com/docs

https://www.tutorialspoint.com/android/android_studio.htm

Search Engine: www.google.com