Mini Project Report

on

“RAKSHANA”

PERSONAL SAFETY NATIVE ANDROID APP

By

Mohnish Keeni (Reg. No.-201900167)

Prashant Kumar (Reg. No.-201900231)

Kesoju Amrithanshu (Reg. No.-201900190)

Group Id - 19

In partial fulfillment of requirements for the award of degree in

Bachelor of Technology in Computer Science and Engineering

(2022)

Under the Project Guidance of

Dr. Subash Harizan

(Assistant Professor II)

Sikkim Manipal Institute of Technology, Majitar

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

SIKKIM MANIPAL INSTITUTE OF TECHNOLOGY

(A constituent college of Sikkim Manipal University)
MAJITAR, RANGPO, EAST SIKKIM – 737136

PROJECT COMPLETION CERTIFICATE

This is to certify that the below mentioned students of Sikkim Manipal Institute of Technology have worked under my supervision and guidance from **17th January 2022 to 21th May 2022** and successfully completed the project entitled **"Personal Safety Native Android App"** in partial fulfillment of the requirements for the award of Bachelor of Technology in Computer Science and Engineering.

University Registration No	Name of Student	Course
201900167	**Mohnish Keeni**	**B.Tech (CSE)**
201900231	**Prashant Kumar**	**B.Tech (CSE)**
201900190	**Kesoju Amrithanshu**	**B.Tech (CSE)**

Dr. Subash Harizan

Assistant Professor II

Department of Computer Science and Engineering

Sikkim Manipal institute of Technology

Majitar, Sikkim – 737136

PROJECT REVIEW CERTIFICATE

This is to certify that the work recorded in this project report entitled **“Personal Safety Native Android App”** has been jointly carried out by **Mohnish Keeni (Reg. 201900167), Prashant Kumar (Reg. 201900231) and Kesoju Amrithanshu (Reg. 201900190)** of Computer Science & Engineering Department of Sikkim Manipal Institute of Technology in partial fulfillment of the requirements for the award of Bachelor of Technology in Computer Science and Engineering. This report has been duly reviewed by the undersigned and recommended for final submission for Mini Project Viva Examination.

Dr. Subash Harizan

Assistant Professor II

Department of Computer Science and Engineering

Sikkim Manipal Institute of Technology

Majitar, Sikkim – 737136

CERTIFICATE OF ACCEPTANCE

This is to certify that the below mentioned students of Computer Science & Engineering Department of Sikkim Manipal Institute of Technology (SMIT) have worked under the supervision of **Dr. Subash Harizan** of Assistant Professor, Department of Computer Science and Engineering from **17th January 2022 to 21th May 2022** on the project entitled **"Personal Safety Native Android App".**

The project is hereby accepted by the Department of Computer Science & Engineering, SMIT in partial fulfillment of the requirements for the award of Bachelor of Technology in Computer Science and Engineering.

University Registration No	Name of Student	Project Venue
201900167	**Mohnish Keeni**	**SMIT**
201900231	**Prashant Kumar**	**SMIT**
201900190	**Kesoju Amrithanshu**	**SMIT**

Prof. (Dr.) Kalpana Sharma

Professor & Head of Department

Computer Science & Engineering Department

Sikkim Manipal Institute of Technology

Majitar, Sikkim – 737136

DECLARATION

We, the undersigned, hereby declare that the work recorded in this project report entitled "**Personal Safety Native Android App**" in partial fulfillment for the requirements of award of B.Tech (CSE) from Sikkim Manipal Institute of Technology (A constituent college of Sikkim Manipal University) is a faithful and bonafide project work carried out at "**SIKKIM MANIPAL INSTITUTE OF TECHNOLOGY**" under the supervision and guidance of **Dr. Subash Harizan,** Assistant Professor, Department of Computer Science and Engineering.

The results of this investigation reported in this project have so far not been reported for any other Degree or any other Technical forum.

The assistance and help received during the course of the investigation have been duly acknowledged.

Mohnish Keeni (Reg. No.-201900167)

Kesoju Amrithanshu (Reg. No.-201900190)

Prashant Kumar (Reg. No.-201900231)

ACKNOWLEDGMENT

We take this opportunity to acknowledge indebtedness and deep sense of gratitude to my guide **Dr. Subash Harizan** whose valuable guidance and kind supervision gave us throughout the course which shaped the present work as it shows.

IWe pay our deep sense of gratitude to **Prof. (Dr.) Kalpana Sharma, H.O.D, Computer Science & Engineering Department, Sikkim Manipal Institute of Technology** for giving us the opportunity to work on this project and provided all support required.

We are obliged to our project coordinators **Dr. Sandeep Gurung and Mr. Biraj Upadhyaya** for elevating, inspiration and supervising in completion of our project.

We would also like to thank any other staff of **Computer Science & Engineering Department, Sikkim Manipal Institute of Technology** for giving us continuous support and guidance that has helped us in completion of our project.

Mohnish Keeni (Reg. No.-201900167)
Kesoju Amrithanshu (Reg. No.-201900190)
Prashant Kumar (Reg. No.-201900231)

DOCUMENT CONTROL SHEET

1	Report No	CSE/Mini Project/Internal/B.Tech/B/19/2022
2	Title of the Report	Personal Safety Native Android App
3	Type of Report	Technical
4	Author	Kesoju Amritanshu, Prashant Kumar, Mohnish Keeni
5	Organizing Unit	Sikkim Manipal Institute of Technology
6	Language of the Document	English
7	Abstract	Brief abstract
8	Security Classification	General
9	Distribution Statement	General

TABLE OF CONTENTS

Chapter	Title	Page No.
	Abstract	
1	Introduction	1
2	Literature Survey	2
3	Problem Definition	6
4	Solution Strategy	7
5	Implementation Details	9
6	Experimental Results	19
7	Testing and Validation	20
8	Conclusion	23
9	Limitations and Future Scope	24
10	Gantt Chart	25
11	References	26

LIST OF FIGURES

Figure No.	Figure Name	Page No.
4.1	Block Diagram	7
4.2	Flowchart	8
5.2.1	Activity Life Cycle Diagram	9
5.3.1	Sign in Page	13
5.3.2	Authentication Page	13
5.3.3	After Pressing “Trouble Signing in?” button	14
5.3.4	Email With Password Reset Link	14
5.3.5	Password reset link opened in browser	15
5.3.6	After entering new password and pressing “SAVE” button	15
5.3.7	Layout of Homepage	16
5.3.8	Layout of Dashboard	16

5.3.9	Adding Guardian Details	17
5.3.10	Viewing Guardian Details	17
5.3.11	Emergency SMS	18
5.3.12	Google Maps Location View	18
10.1	Gantt Chart	25

LIST OF TABLES

Table No.	Title Name	Page no.
2.1	Literature Survey	5
7.1	Testing for Authentication	20
7.2	Testing for entry of guardian details to guardian database	21
7.3	Testing for sending of SMS alerts on pressing “DON’T PANIC” button	22

ABSTRACT

The usage of smart phones equipped with GPS navigation unit have increased rapidly over the last few years. The widespread use of smartphones in today's world coupled with the fact that they are portable and usually carried by people wherever they go makes it an ideal candidate for usage for personal safety or various other protection security. The main aim of our app is that with a single click when the user feels he/she is in danger, the guardians registered by him/her should be notified of the user's location. Location tracking information via SMS helps to find the location of the victim quickly and provide the necessary assistance quickly.

1. INTRODUCTION

Our group members were always inquisitive about security purpose projects so decided that our project will be security based. We also wanted to make something unique and for a social cause which led us to this idea of a Personal Security App. Our app aims to alert the registered emergency contacts of the user via SMS which contains the google maps location link of the user intext. Once the registered emergency contacts of the user receive the SMS, they can travel to the user's location and provide the necessary assistance according to the circumstance due to which the user sent the alert. This is very helpful as the location of the user is sent to contacts instantly which ensures that the user receives the required help more quickly. Our app titled "Rakshana" is a personal safety application for smart phones working over android platform.

2. LITERATURE SURVEY

[1] A Review on Kotlin and Android Studio Java by NiketKeny:- This paper describes how Kotlin differs from Java and various different features of Kotlin and Android Java and concludes which programming language will be perfect fit the developers. You can code using both Kotlin and Java code in the same project as the compiler supports both languages. If a developer wants to migrate their Java project to Kotlin, they can do so converting the Java code to Kotlin one file at a time. Null Pointer Exception is one of the most common reasons why Android applications crash. It takes lot of time to process and fix all of the null point exceptions in your project. In Kotlin null pointer exception is not a problem as by default, Kotlin does not allow us to assign null values to objects (Null Safety)we don't need to worry about null pointer exceptions. Compared to Java, Kotlin requires fewer lines of code to achieve the same task and this translates to projects written in Kotlin having comparatively smaller files while also saving the developer some time. Kotlin is also more readable and easy to understand as it is concise and it removes a lot of unnecessary code that the same project would contain if it was coded in Java. Hence Kotlin is a better option for developers and our app will be developed using Kotlin.

[2] Application of Firebase in Android App Development- A Study by Chunnu Khawas, Pritam Shah :- This paper describes the various features of Firebase in Android App Development. Firebase is a NoSQL database. For android apps, Firebase uses JSON for storing data while the other databases use a tabular format for storing data. Firebase provides services like a real-time database and backend. An API is provided to the application developer which allows application data to be synchronized across clients and stored on Firebase's cloud. The client libraries are provided by the company which enables integration with Android, IOS, and JavaScript applications. Firebase UI Auth supports social login provider like Facebook, Google GitHub, and Twitter. It is a service that can authenticate users using only client-side code. It also includes a user management system whereby developers can enable user authentication with email and password login stored with Firebase. In our project we will use firebase UI Auth to authenticate users using google and email-password options.

[3] Implementation of Model-View-ViewModel (MVVM) Architecture Pattern by Aziz Musthafa, Dihin Muriyatmoko :- This paper describes how MVVM Architetcture pattern is implemented. Model-View-ViewModel (MVVM) is a software architectural pattern that facilitates a separation of development of the graphical user interface from the development of the back-end logic (the data model). Our project will be developed using MVVM architecture.

[4] Android Google Services Part 2-Location and Geocoding by David Drohan:- This paper describes the Android features and the services that let you find, contextualize, and map physical locations. Using Fused Location Provider you can find the device's current location and send notifications when the device's location is 'near' some other location, (via proximity alerts or GeoFencing). Android's Network (Fused) Location Provider determines user location using Cell Towers and Wi-Fi signals. It is less accurate than GPS, but it works indoors and outdoors, it responds faster and it uses lesser battery power. The goal of Fused Location Provider is to lessen the workload of developers who want to interact with location information by Providing a single programmable interface where Google does the hard work in sourcing location, simply feeding it to developers' applications We will use fused location provider in our app to determine user location when alert is triggered.

SL NO	Author	Paper and Publication Details	Findings	Relevance to the project
1	NiketKeny	A Review on Kotlin and Android Studio Java – International Journal of Applied Engineering Research, Volume 14, Number 7, 2019	• Interchangeability with Java. • Easy to Learn. • No NULL Pointer Exception. • Lazy Loading	App will be developed using Kotlin.
2	Chunnu Khawas, Pritam Shah	Application of Firebase in Android App Development- A Study – International Journal of Computer Applications ,Volume 179, Number 46, 2018	• Firebase Auth is a service that can authenticate users using only client-side code. • It also includes a user management system whereby developers can enable user authentication with email and password login stored with Firebase.	Firebase UI Authentication will be used for the developer.

SL NO	Author	Paper and Publication Details	Findings	Relevance to the project
3	Aziz Musthafa, Dihin Muriyatmoko	Implementation of Model-View-ViewModel (MVVM) Architecture Pattern, International Journal of Computer Applications, November 2019	• Model-View-ViewModel is a software architectural pattern that facilitates a separation of development of the graphical user interface from the development of the back-end logic.	• The viewModel is only used in our app so that the user can easily and within no time access the features. • In our app, two fragments need to share data quickly, so this architecture pattern is used.
4	David Drohan	Android Google Services Part 2- Location and Geocoding, Department of Computing & Mathematics Waterford Institute of Technology	• Using Fused Location Provider we can find the device’s current location (GPS, Network Provider) • send notifications when the device’s location is ‘near’ some other location, (via proximity alerts) • provides simple API and optimizes the device’s use of battery power .	• Fused Location Provider is used in our app to retrieve the device's last known location. • It manages the underlying location technology and provides a simple API .

Table 2.1. Literature Survey

3. PROBLEM DEFINITION

People are most likely to get attacked with the intention of robbery, abduction and other threats when they are alone in and in an unsafe and unfamiliar environment. When faced with a dangerous situation when they are alone and in an unsafe and unfamiliar environment, alerting their friends and family of the situation and sending their location can help them receive the necessary assistance quickly. Nowadays most people have smartphones equipped with various powerful tools like accelerometer gyroscope, GPS (Global Positioning System) and ability to communicate which they carry on them wherever they go. These features of smartphones make them an effective tool to help in such situations by alerting others of the dangers so that they can provide the necessary help. Existing applications aim to alert authorities such as police .Our app provides the user with an alternative as it helps the user alert their friends and family with their location when they are faced with dangerous situations like the ones mentioned above when they are alone and in an unfamiliar environment. Existing apps require the user to either pay a subscription fee or display ads which can both be inconvenient in a emergency situation.

4. SOLUTION STRATEGY

The main aim of this app is to provide some security to people who are alone and in need of help by helping them alert their friends and family quickly so they can provide help accordingly. The app will allow the user to register emergency contacts, allow the user to send alerts discreetly with their location to emergency contacts via SMS while having a user interface that is easy to understand and use. Also, unlike existing apps our app will be ad free and free to use.

4.1. DESIGN DIAGRAMS:

1)Block Diagram:

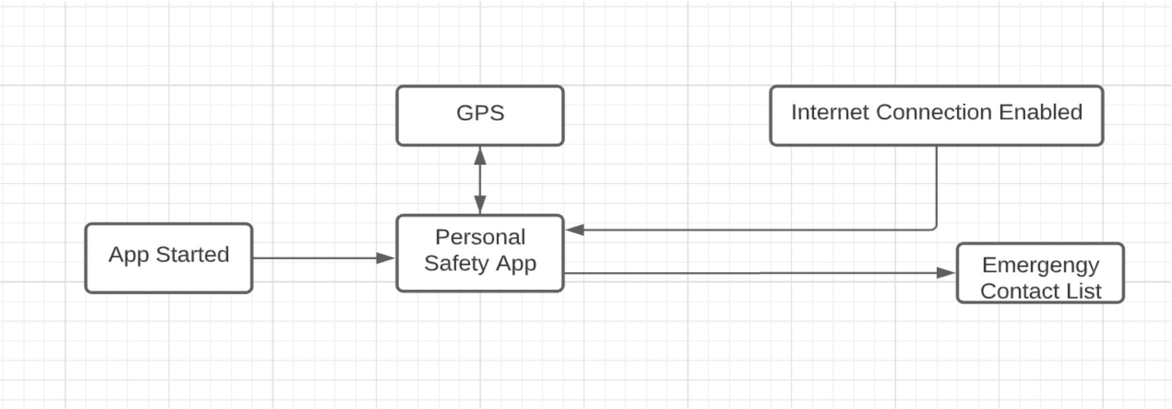


Figure 4.1. Block Diagram

In Figure 4.1 shows the block diagram of our application. Initially, when we start the app, it first checks whether the location settings, data connection settings in the application are on or not. Then, it tracks the location of the victim via GPS and sends these location co-ordinates in the form of URL through message to the registered contacts when the alert is triggered. Here, registered contacts means the contact details that are saved in the app as emergency contact. Now, on the device of the emergency contact , by clicking on the URL in the message, it gives the location of the victim on google maps.

2)Flowchart:

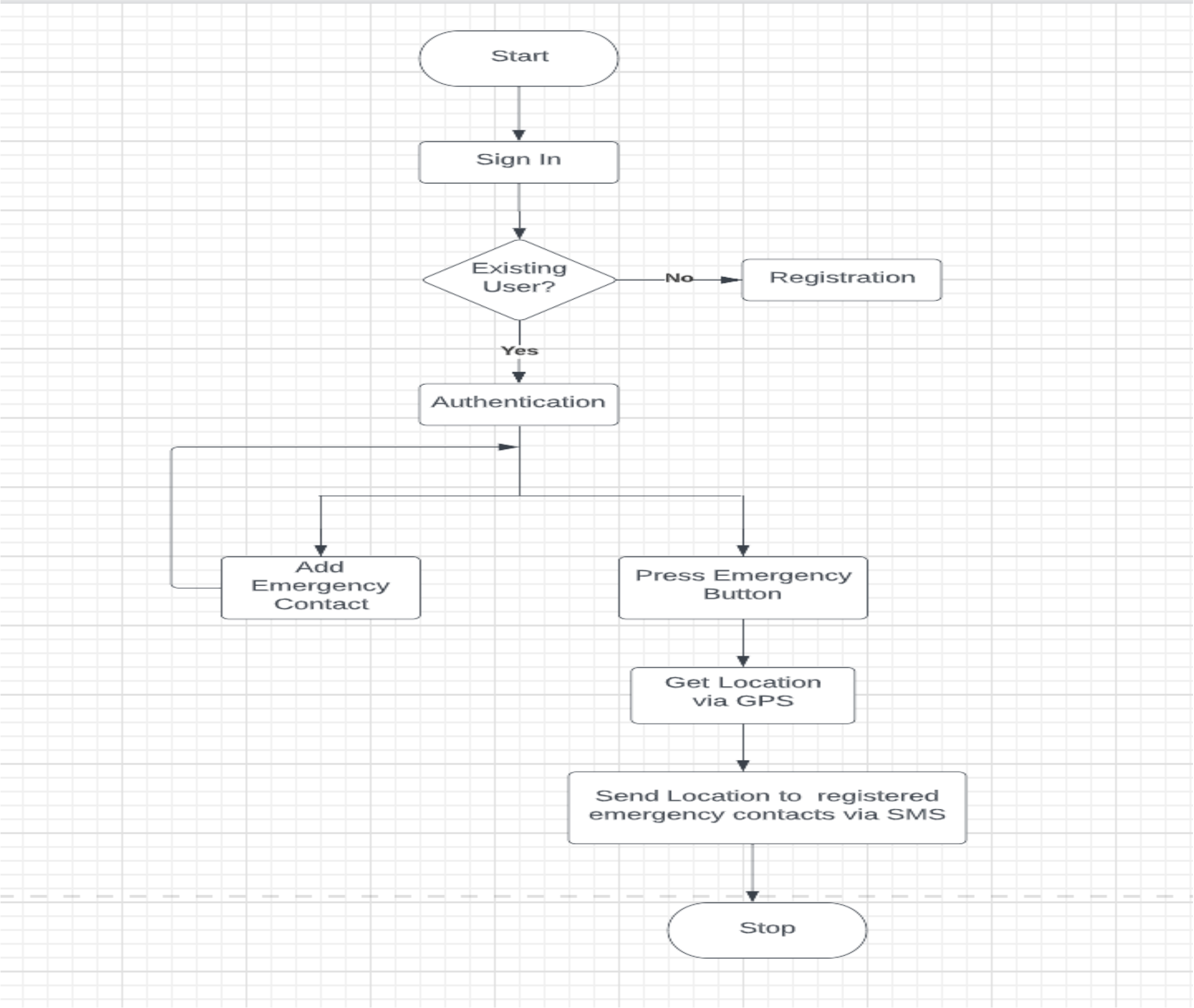


Figure 4.2. Flowchart

Figure 4.2 represents the flowchart of our application. Initially, when we start the app, the user will be asked to sign in. If the user is not an existing user then they are asked to register and if they are existing user they are asked to enter their password for authentication. Once logged in the user can register an emergency contact or press emergency button to trigger the alert. When the emergency button is pressed the location of the user is fetched and then sent to the registered emergency contacts via SMS.

5. IMPLEMENTATION DETAILS

5.1. Software Details:

1. Android SDK 11.0 R – API 30 Level

2. Android Studio Version – 4.2

3. Kotlin version: 211-1.6.10-release-923-AS7442.40

4. Firebase UI Auth version- 5.0.0

5. Layouts: XML Schema 1.0

6. Run on mobile (Xiaomi Redmi Y2) via USB cable

5.2. Implementation Details:

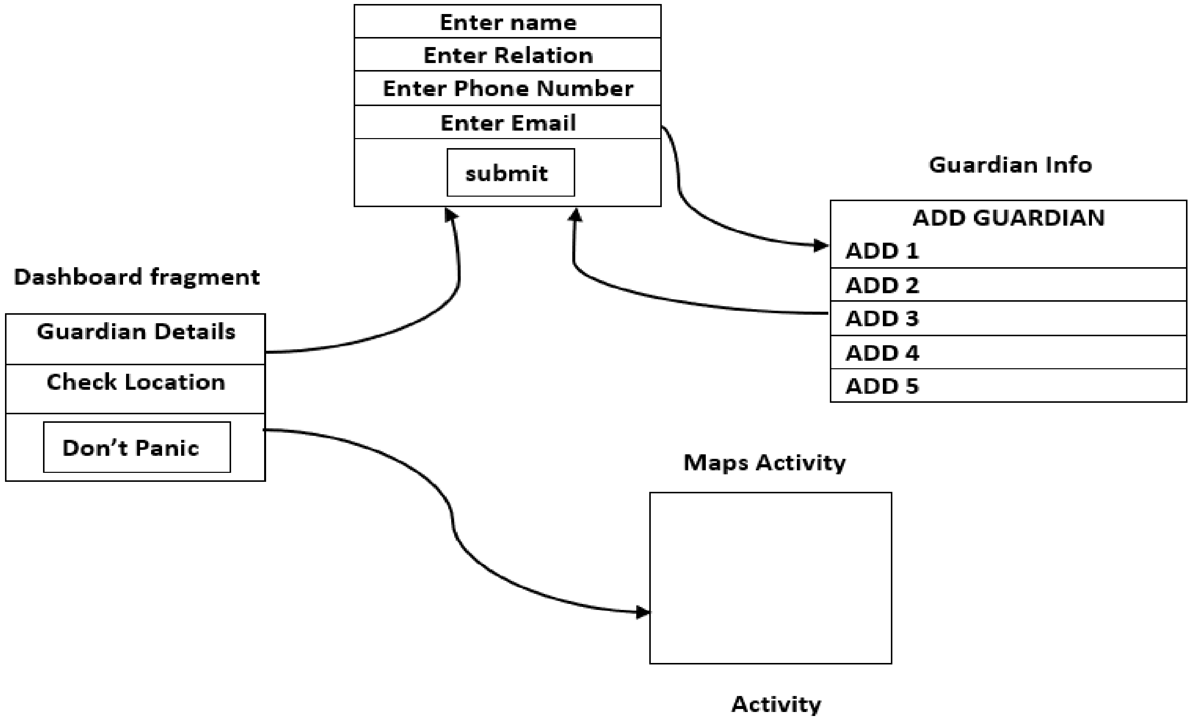


Figure 5.2.1. Activity Life Cycle Diagram

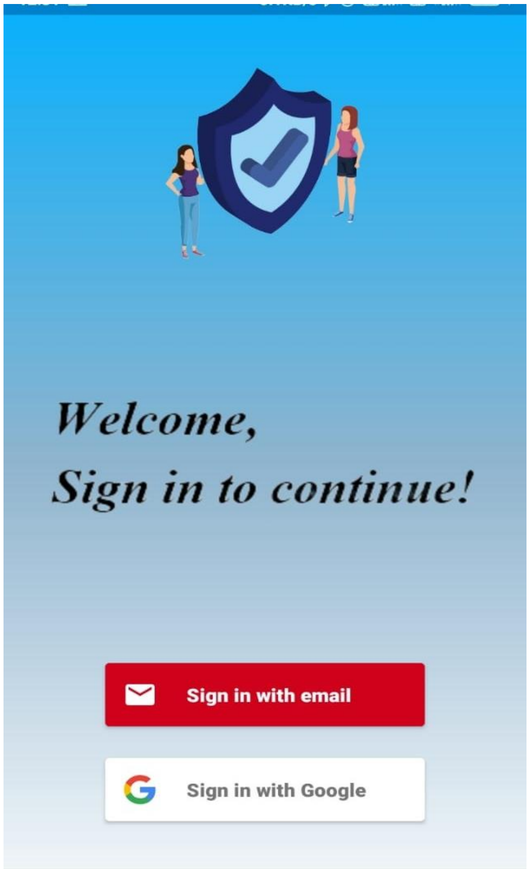
Pseudocode for DashboardFragment.kt File

```
Step 1: Import firebase modules, gms modules, widget modules, fragment modules
Step2: Create a class named  DashBoardFragment : Fragment()
Step3: Declare private lateinit var variables  binding, fusedLocationClient , lastLocation:
Step4: Declare private var variables  Latitude: String = "" and Longitude: String = ""
Step5: Intialise uiScope which is a CourtineScope variable (which simplifies the execution and run faster).
Step6: In the function companion object
        Step6.1:Declare const val variables used in the app ;TAG= "DashBoardFragment",SIGN_IN_RESULT_CODE = 1001, PERMISSION_SEND_SMS = 1
Step7: Create a onCreateView function with its parameters
Step8: Create usedLocationClient = LocationServices.getFusedLocationProviderClient(activity!!) to access the location services of the app.
Step9:  Inflate the layout for this fragment where making a connection between XML layout and logic of Kotlin
Step10:  Call getLocation()
Step11: Once Guardian Details Button is clicked on , it calls findNavController().navigate(DashBoardFragmentDirections.actionDashBoardFragmentToGuardianInfo()) // to access and view details of guardian
Step12: The Don’t Panic Button is pressed,
        Step12.1: Call getLocation()
        Step12.2: if  Longitude.isNullOrBlank() || Longitude.isNullOrEmpty()
          // Display a toast message to give as error .
                Step12.2.1: Toast.makeText(activity!!,"Click on Location button and try again").show()
        Step12.3:else if ActivityCompat.checkSelfPermission(activity!!, Manifest.permission.SEND_SMS)!=PackageManager.PERMISSION_GRANTED)
             Step12.3.1: Send SMS to the Registerred Guardian
        Step12.4: return binding.root
Step13 Create onViewCreated stae and call observeAuthenticationState()
```

```
Step14: Create onActivityResult(requestCode: Int, resultCode: Int, data: Intent?)
    Step14.1: if  requestCode == SIGN_IN_RESULT_CODE
        Step14.1.1: Declare val response = IdpResponse.fromResultIntent(data)
        Step14.1.2: if resultCode == Activity.RESULT_OK
            //User successfully signed in
        Step14.1.3: else
            // Sign in failed. If response is null, the user canceled the
            // sign-in flow using the back button. Otherwise, check
            // the error code and handle the error.
                Step 14.1.3.1: Display error
Step15: In the function observeAuthenticationState() {
    Step15.1: Observe authentication sate when (authenticationState)
    Step15.2: In LoginViewModel is authenticated and display "You are stronger than you think!!”, and your name on main activity
     Step15.3: Call launchSignInFlow()
Step16: In the function launchSignInFlow()
    // Give users the option to sign in / register with their email
    // If users choose to register with their email and they will need to create a password
    Step16.1: Create an object providers which create AUTHUI of EmailBuilder and GoogleBuilder.
    // Create and launch sign-in intent and listen to the response of this activity with the
Step17: In the function startActivityForResult(
        Step17.1: Intialise a sign intent to login  on navigation header with logo and build.
Step 18: Create function onRequestPermissionsResult(requestCode: Int, permissions: Array<out String>, grantResults: IntArray)
// which accepts the requests of SMS and also check user permission of sending sms is present while using app.
        Step18.1: Call emergencyFun
Step19: In the function private fun getLocation()  // Retrieves the location of finding latitude and longitude.
    Step19.1: fusedLocationClient listens and access the last locatiob of user
    Step19.2: location ->if  location != null
        Step19.2.1: then lastLocation = location
        Step19.2.2: Intialise Latitude = (location.latitude).toString() and
                Longitude = (location.longitude).toString()
Step20: In the function private fun emergencyFun() {
    Step20.1: Create val variable  db which stores data of the guardian details in the Guardian Database Dao and also access the email of guardian
    // The subject of the mail "From Personal Safety App”
    // Send the coordinates of the user in form of longitude  and latitiude.
Step21: In the function phoneList.forEach()  it access the correct phone number of guardian and sends sms to the mobile.
    Step21.1: Create a shareIntent which creates a Action Send  //making the different available social media sites.
      Step21.2: Intialise startActivity(Intent.createChooser(shareIntent, "Send mail using..")) // intent gives extra message passage options.
```

5.3. Screenshots of App:

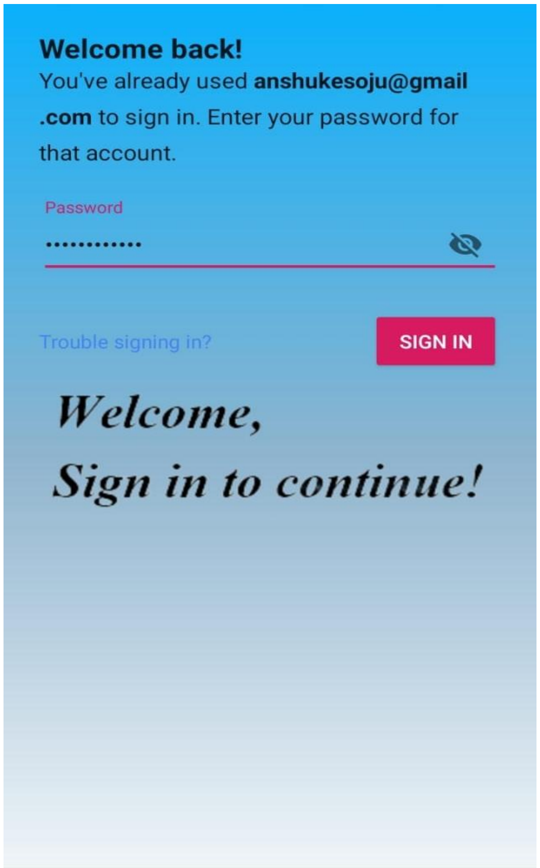
1> **User Authentication via email/password and google account added with Firebase UI Auth.**



Welcome,

Sign in to continue!

Figure 5.3.1. Sign in Page



Welcome back!

You've already used **anshukesoju@gmail.com** to sign in. Enter your password for that account.

Password

••••••••••••

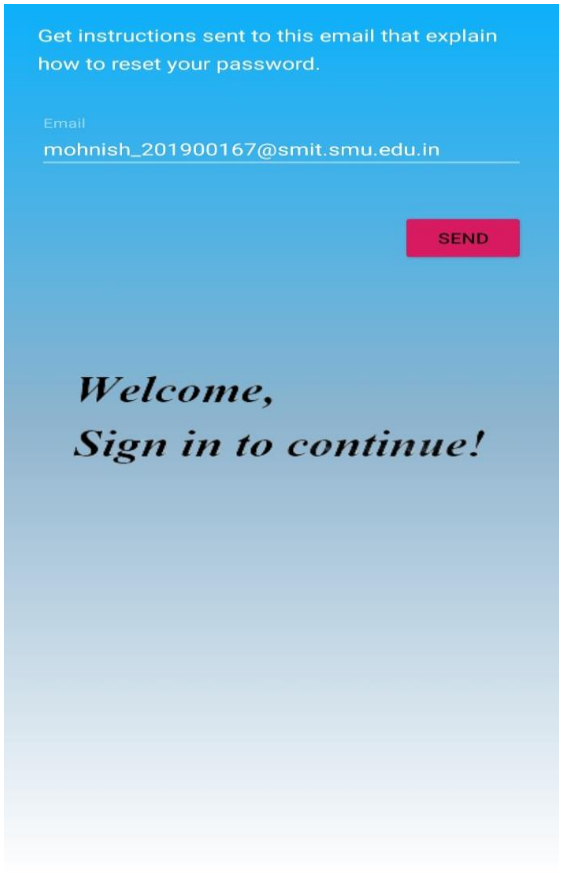
Welcome,

Sign in to continue!

Figure 5.3.2. Authentication Page

User can login using their google account or through email/password login(refer figure 5.3.1). If the user tries to login via email and they are not registered already, they are taken to registration page and asked to register. If a user forgets their password then they can have a password reset link sent to their email with the Trouble signing in? option so that they can log in.

2>Trouble Signing in Option to recover password.

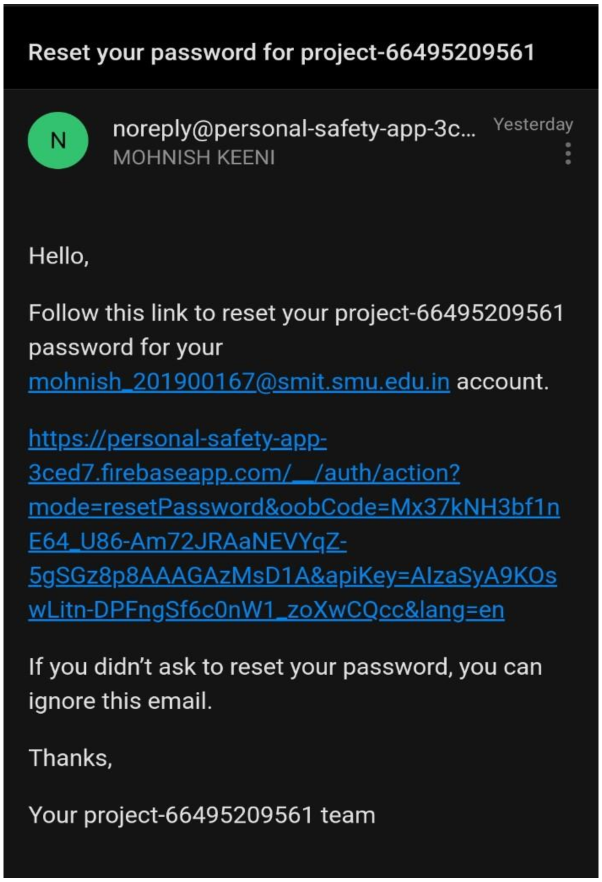


Get instructions sent to this email that explain how to reset your password.

Email
mohnish_201900167@smit.smu.edu.in

Welcome,
Sign in to continue!

Figure 5.3.3. After Pressing "Trouble Signing in?" button



Reset your password for project-66495209561

noreply@personal-safety-app-3c... Yesterday
MOHNISH KEENI

Hello,

Follow this link to reset your project-66495209561 password for your mohnish_201900167@smit.smu.edu.in account.

https://personal-safety-app-3ced7.firebaseapp.com/__/auth/action?mode=resetPassword&oobCode=Mx37kNH3bf1nE64_U86-Am72JRAaNEVYqZ-5gSGz8p8AAAGAzMsD1A&apiKey=AIzaSyA9KOswLitn-DPFngSf6c0nW1_zoXwCQcc&lang=en

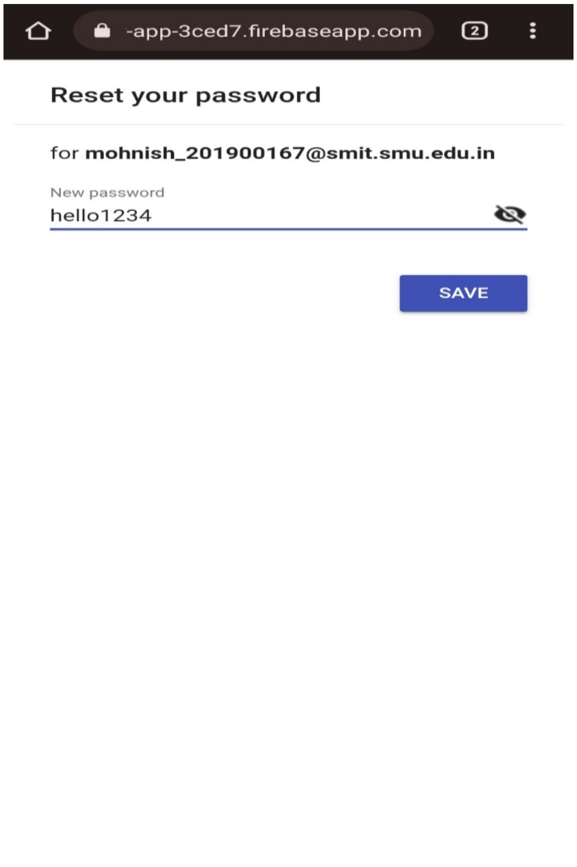
If you didn't ask to reset your password, you can ignore this email.

Thanks,

Your project-66495209561 team

Figure 5.3.4. Email With Password Reset Link

If a user who has created account using email/password option, forgets his/her password while logging in they can use the Trouble Signing In option on the login screen to have a password reset link sent to their email as shown in Figure 5.3.4. On opening the link the following page(refer Figure 5.3.5) opens:

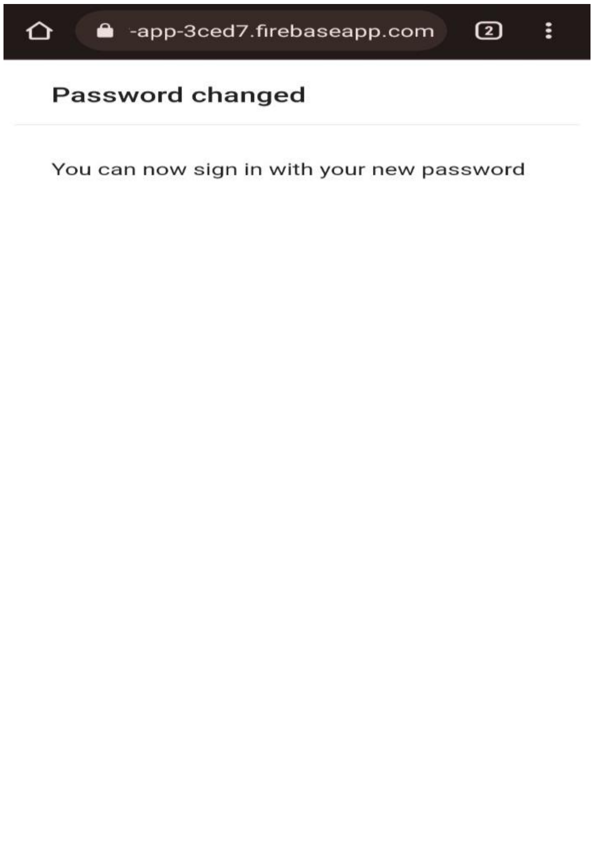


Reset your password

for **mohnish_201900167@smit.smu.edu.in**

New password
hello1234

Figure 5.3.5. Password reset link opened in browser



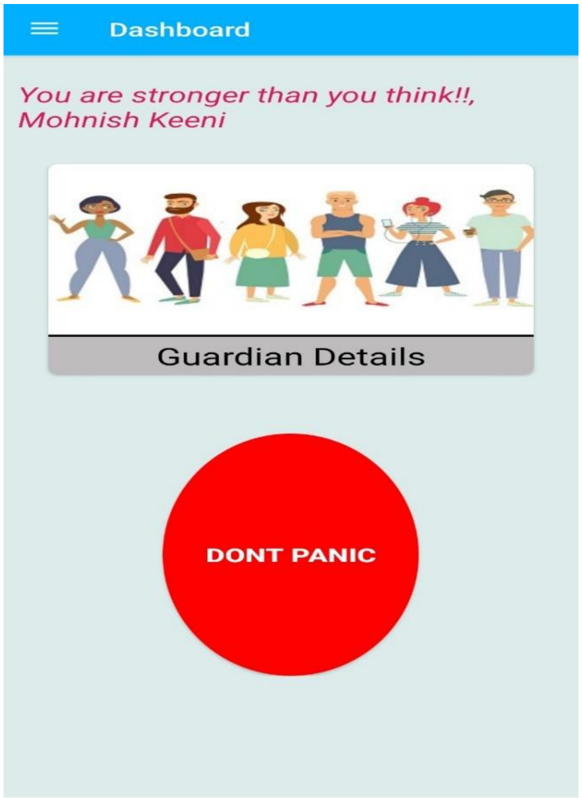
Password changed

You can now sign in with your new password

Figure 5.3.6. After entering new password and pressing "SAVE" button

On opening the password reset link, the user has to create a new password for their account as shown in Figure 5.3.5, when ever he forgets and press the "SAVE" button. Once done, the user can then login in the app with their new password.

3>Designed Layout for Homepage and Dashboard



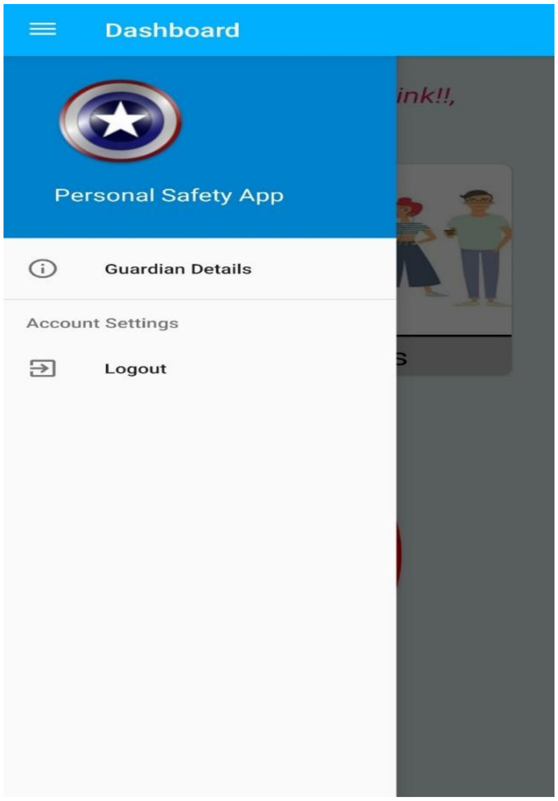


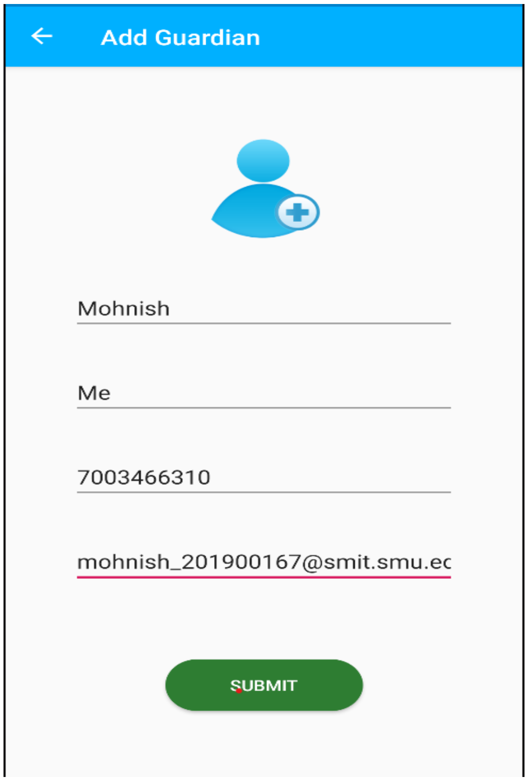
Figure 5.3.7. Layout of Homepage

Figure 5.3.8. Layout of Dashboard

Designed the layout for homepage (refer Figure 5.3.7) with buttons for adding guardian details, updating location manually and a button to trigger the alert which send SMS with location to registered emergency contacts.

Dashboard (refer Figure 5.3.8) with logout button and buttons for checking guardian details and updating location manually has been designed too.

4>Guardian details fragment added to allow the user to register emergency contacts and view details of already registered emergency contacts.



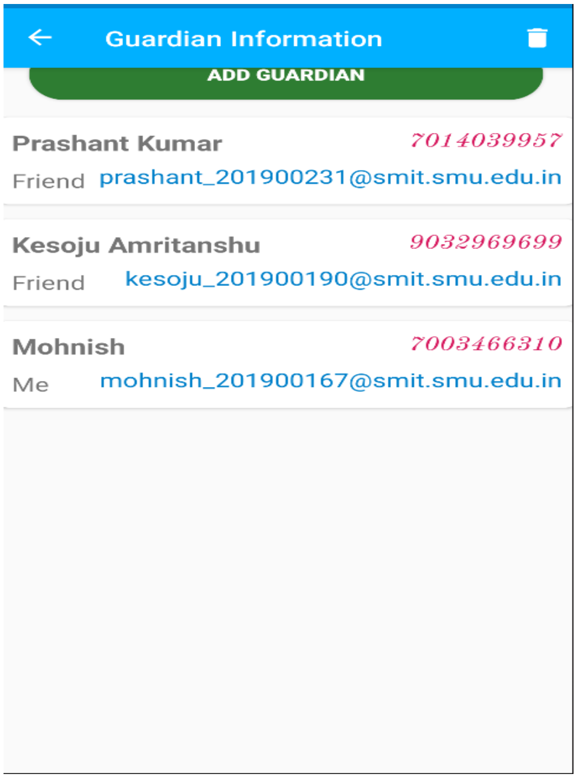
Mohnish

Me

7003466310

mohnish_201900167@smit.smu.ec

Figure 5.3.9. Adding Guardian Details



Prashant Kumar		7014039957
Friend	prashant_201900231@smit.smu.edu.in	
Kesoju Amritanshu		9032969699
Friend	kesoju_201900190@smit.smu.edu.in	
Mohnish		7003466310
Me	mohnish_201900167@smit.smu.edu.in	

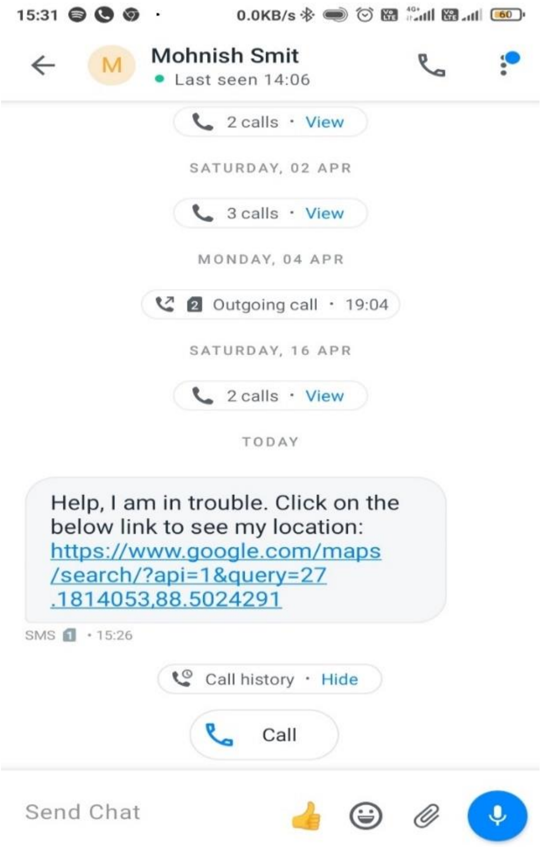
Figure 5.3.10. Viewing Guardian Details

Guardian details fragment as a shown in Figure 5.3.9 and Figure 5.3.10 allows the user to register a new guardian by entering their name, relation, phone number and email. It also allows the user to view the details of guardians who have already been registered.

5>Location fragment added to fetch user location when alert is triggered or check location button is pressed.

Location fragment fetches the longitude and latitude of the user when alert is triggered or user presses the check location button.

6>Emergency fragment has added to send sms with location to registered emergency contacts.


15:31 0.0KB/s

Mohnish Smit

Last seen 14:06

2 calls · View

SATURDAY, 02 APR

3 calls · View

MONDAY, 04 APR

Outgoing call · 19:04

SATURDAY, 16 APR

2 calls · View

TODAY

Help, I am in trouble. Click on the below link to see my location: https://www.google.com/maps/search/?api=1&query=27.1814053,88.5024291

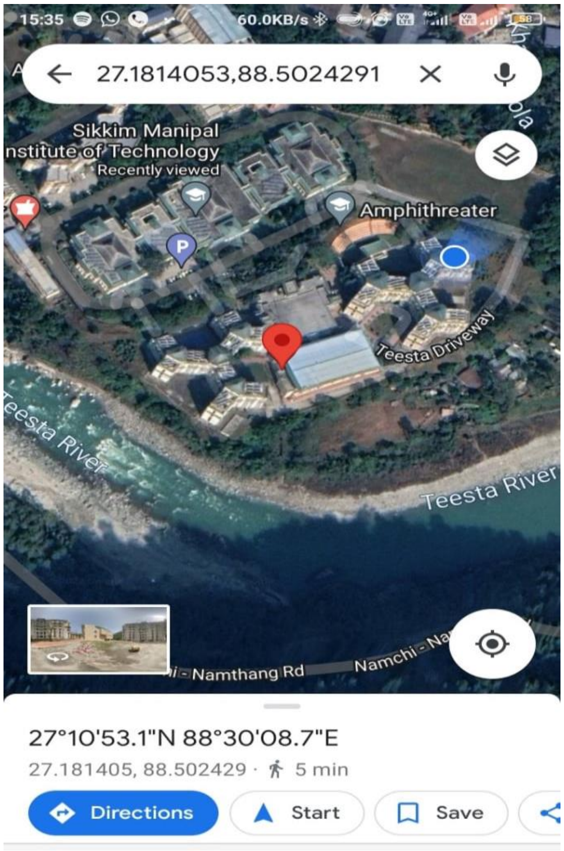
SMS · 15:26

Call history · Hide

Call

Send Chat

Figure 5.3.11. Emergency SMS


15:35 60.0KB/s

27.1814053,88.5024291

27°10'53.1"N 88°30'08.7"E

27.181405, 88.502429 · 5 min

Figure 5.3.12. Google Maps Location View

Emergency fragment sends sms as shown in Figure 5.3.11 with google maps location(refer to Figure 5.3.12) of the user to the registered guardians when Don’t Panic button is pressed.

6. EXPERIMENTAL RESULTS

On launching our app, the user will be asked to login. There are two login options: via email/password and via google account. If the user does not have an existing account they will be asked to create one. Once logged in, the user can register guardian details by clicking on guardian details button on the homepage or in the dashboard. Once the user has registered the required guardians, he/she can trigger the alert by pressing on “Don’t Panic” button on the homepage of the app. Once the button is pressed, the coordinates of the user (longitude and latitude) are fetched using FusedLocationProvider API which makes use of GPS sensor, nearby wifi signals and cell tower signals) to provide a more accurate location. These coordinates are then used to generate google maps link with the location of the user which is then sent to the registered guardians via SMS. The link can then be opened by the guardian which informs them of the user’s location so that they can provide the required assistance.

7. TESTING AND VALIDATION

7.1 Testing for Authentication:

Test Case #	Test Case Description	Test Data	Expected Result	Actual Result	Pass /Fail
1	Check Response when correct email id and password is entered.	Email id: mohnish_201900167@smit.smu.edu.in Password: password	Login successful.	Login successful.	Pass
2	Check Response when correct email id and incorrect password is entered.	Email id: mohnish_201900167@smit.smu.edu.in Password: hello1234	Login unsuccessful.	Login unsuccessful.	Pass
3	Check Response when unregistered email id is entered.	Email id: abc@mail.com	User must be directed to create an account.	User is directed to create an account.	Pass
4	Check response when user chooses "Trouble logging in?" option	NA	Password reset link must be sent to the user's email id.	Password reset link is sent to the user's email Id.	Pass
5	Check response when user presses logout button in dashboard	NA	User must be logged out from the app.	User is logged out from the app.	Pass

Table 7.1. Testing for authentication

7.2 Testing for entry of guardian details to guardian database:

Test Case #	Test Case Description	Test Data	Expected Result	Actual Result	Pass/Fail
1	Check Response when guardian's name, relation to the user, phone number and email are entered in the add guardians form before pressing submit.	Name: Abc Relation: Friend Phone no: 9876540323 Email: abc@gmail.com	Guardian details must be successfully added to the database.	Guardian details are successfully added to the database.	Pass
2	Check response when one or more fields are left empty in add guardians form before pressing submit.	Name: Relation: Phone no: Email:	Guardian details must not be added to the database and this field cannot be empty message must be displayed for each empty field.	Guardian details were not added to the database and "This field cannot be empty" message was displayed for each empty field.	Pass

Table 7.2. Testing for entry of guardian details to guardian database

7.3 Testing for sending of SMS alerts on pressing “DON’T PANIC” button:

Test Case #	Test Case Description	Test Data	Expected Result	Actual Result	Pass/Fail
1	Check Response when guardian’s name, relation to the user, phone number and email are entered in the add guardians form before pressing submit.	NA	Alerts containing the user’s location in a google maps link must be sent to the registered guardians via SMS.	Alerts containing the user’s location in a google maps link were sent to the registered guardians via SMS.	Pass

Table 7.3. Testing for sending of SMS alerts on pressing “DON’T PANIC” button

8. CONCLUSION

Our main aim was to develop a low cost solution to help the user alert their close contacts in case of a emergency along with their location. In our implementation, the user can alert the registered emergency contacts with location via a google maps location link which is sent intext via an SMS via the user's primary sim just by clicking a button. Our app is free of cost can be installed and used by anyone with an android phone running android version 5.0 and above. The app has user friendly UI which is easy to navigate and use and the user can easily trigger the alert in an emergency situation. Hence, our main objective has been accomplished via the completion of this project.

9. LIMITATIONS AND FUTURE SCOPE

The main limitation of our app in its current state are:

- It requires the user to have an active internet connection in order use GPS for fetching the location of the user so in case the user is an a poor network area then it would be difficult to successfully send the alerts.
- Our app is currently on the android platform and can be installed on android devices with android version 5.0 and above. There is no compatibility for Iphones or android devices having version older than android 5.0.
- Also the user will have to open the app and manually press the "Don't Panic" button in order to send alerts to the registered emergency contacts which might not be ideal in an emergency situation.

The future scope of our project is:

- Adding gesture based (such as shaking the phone to trigger alert , pressing the power button a certain number of times, or drawing a gesture on the screen) or voice recognition based(user can say a certain phrase which will trigger the alert) triggering of alert that would make the process of alerting guardians much easier to access in an emergency.
- Adding a feature to automatically dial the nearest police station according to the users location by clicking a separate button or using an alternate alert triggering mechanism.

10. GANTT CHART

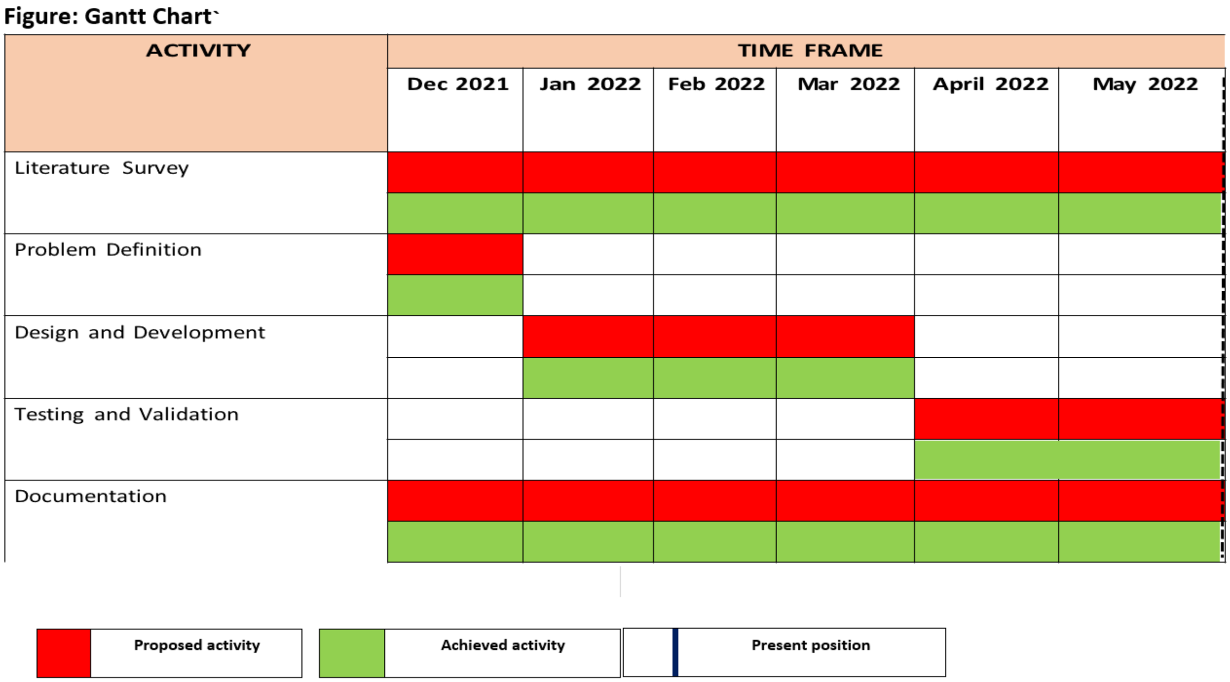


Figure: Gantt Chart`

ACTIVITY	TIME FRAME					
	Dec 2021	Jan 2022	Feb 2022	Mar 2022	April 2022	May 2022
Literature Survey	🟥	🟥	🟥	🟥	🟥	🟥
	🟩	🟩	🟩	🟩	🟩	🟩
Problem Definition	🟥					
	🟩					
Design and Development		🟥	🟥	🟥		
		🟩	🟩	🟩		
Testing and Validation					🟥	🟥
					🟩	🟩
Documentation	🟥	🟥	🟥	🟥	🟥	🟥
	🟩	🟩	🟩	🟩	🟩	🟩

🟥 Proposed activity 🟩 Achieved activity | Present position

Figure 7.1. Gantt Chart

11. REFERENCES

[1] NiketKeny. "A Review on Kotlin and Android Studio Java" International Journal of Applied Engineering Research ISSN 0973-4562 Volume 14, Number 7, 2019 (Special Issue)

[2] Khawas, Chunnu, and Pritam Shah. "Application of firebase in android app   development-a study." International Journal of Computer Applications 179.46 (2018): 49-53.

[3]Arif, Muhammad Shaker, Aziz Musthafa, and Dihin Muriyatmoko. Implementation of Model-View-ViewModel (MVVM) Architecture Pattern in the Sistem Informasi Akademik UNIDA Gontor Mobile Application. Diss. Universitas Darussalam Gontor, 2019.

[4]David Drohan. "Mobile Application Development" Department of Computing & Mathematics,Waterford Institute of Technology
https://ddrohan.github.io/msc-mad/topic04-google-services/talk-2-google-2/01.location.pdf
[Accessed on 26th February,2022]