S-HER
WOMEN SAFETY APPLICATION

BCSF187Z50 - PROJECT WORK PHASE- I

REPORT

Batch 2021-25 (Semester VII)

SUBMITTED BY

PANCHAGNULA SUBRAHMANYA VIKAS

11219A032

RALLAPALLY SAKETH BHARGAVA

11219A038

GUIDED BY

Dr. R. PREMA

ASSISTANT PROFESSOR, CSE

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

SRI CHANDRASEKHARENDRA SARASWATHI VISWA MAHAVIDYALAYA

NOVEMBER 2024

Sri Chandrasekharendra Saraswathi Viswa Mahavidyalaya

Enathur, Kanchipuram – 631 561

BONAFIDE CERTIFICATE

This is to certify that the PROJECT WORK PHASE-I Report entitled **WOMEN SAFETY APPLICATION** is the bonafide work carried out by Mr. **P. Subrahmanya Vikas** Reg.No. **11219A032** and Mr. **R. Saketh Bhargava** Reg.No. **11219A038** during the academic year 2024-2025.

Dr. R. Prema
Assistant Professor,
Department of CSE,
SCSVMV.

Dr. M. Senthil Kumaran,
Head of the Department,
Department of CSE,
SCSVMV.

Submitted for the project work viva - voce examination held on _____

Place: Kanchipuram.
Date:

Examiner 1 ***Examiner 2***

DECLARATION

It is certified that the PROJECT WORK PHASE-I work titled **S-HER WOMEN SAFETY APPLICATION** is originally implemented by me. No ideas, processes, results or words of others have been presented as my own work. Due acknowledgement is given wherever others' work or ideas are utilized.

a. There is no fabrication of data or results which have been compiled /analyzed.

b. There is no falsification by manipulating data or processes, or changing or omitting data or results.

I understand that the project is liable to be rejected at any stage (even at a later date) if it is discovered that the project has been plagiarized, or significant code has been copied. I understand that if such malpractices are found, the project will be disqualified, and the Degree awarded itself will become invalid.

Signature of the student with date

TABLE OF CONTENTS

i. Abstract

Women's safety is a paramount and critical concern in today's society, demanding the development of effective technological solutions. Ensuring the safety and security of women is essential for promoting gender equality, personal well-being, and social progress. Women face various challenges and risks, including harassment, violence, discrimination, and unequal access to resources and opportunities. To address these issues, concerted efforts are being made to raise awareness, advocate for women's rights, and implement effective measures for women's safety. This includes education and awareness campaigns that aim to change societal attitudes and behaviors, promote respectful relationships, and challenge gender stereotypes and biases.

Introducing S-HER, an innovative woman safety application built using Android Studio written in Java, and XML. S-HER is designed to empower women and bolster their security through a comprehensive set of features. Application is directly supported on most android devices and provides a simple easy to go place to ensure security and safety. Location alerts, emergency calls, safety tips, numbers of different helplines and government laws which help woman are available thus making a significant leap towards a safer society. S-HER presents a user-friendly interface and a dependable solution for women's safety. Its development marks a significant stride towards addressing the pressing issue of women's safety in contemporary society, empowering women and fostering a safer environment

Keywords: Application, Equality, Government, Helplines, Location alerts, Safety, Society.

ii. LIST OF FIGURES

iii. LIST OF ABBREVATIONS

S. No	Short Forms	Abbreviation
1.	API	Application Programming Interface
2.	GPS	Global Positioning System
3.	IDE	Integrated Development Environment
4.	LBS	Location Based Services
5.	SDK	Software Development Kit
6.	SOS	Save Our Souls
7.	UI	User Interface
8.	XML	Extensible Markup Language

iv. LIST OF TABLES

Chapter 1

1. Introduction

Android application development has revolutionized the way we interact with technology, bringing a multitude of functionalities and services to our fingertips. Android Studio, the official IDE for Android development, offers a comprehensive set of tools and features that streamline the app creation process. Its intuitive user interface, coupled with a vast array of built-in templates and code snippets, enables developers to design visually appealing and responsive applications. Android Studio also provides robust debugging and testing capabilities, ensuring the reliability and stability of the developed applications. Moreover, its seamless integration with the Android Software Development Kit (SDK) facilitates the utilization of cutting-edge APIs and libraries, enabling developers to harness the full potential of the Android platform.

The safety and security of women remain crucial societal concerns. Women often face various forms of harassment, violence, and discrimination, both in public spaces and private domains. Promoting women's safety is not only a matter of personal well-being but also a fundamental aspect of gender equality. Addressing women's safety requires a multifaceted approach encompassing awareness campaigns, policy reforms, educational initiatives, and the development of technological solutions. By developing a woman safety application, this project aims to contribute to this ongoing effort. The application will provide women with tools to enhance their personal security, access emergency assistance, and foster a sense of empowerment in navigating their daily lives.

1.1. Objectives

Developing a comprehensive women safety application, leveraging Android Studio, Java, and XML, to address the pressing concerns of women's safety and security. Enhance Personal Security: Develop features such as a shake detector and panic button that allow users to trigger emergency alerts and send SOS messages to

registered contacts for immediate assistance. Implement a siren sound to attract attention in critical situations.

1. **Provide Location-Based Assistance:** Integrate location services to transmit the user's last known location to registered contacts, enabling them to locate and assist the user quickly. Enable the addition of multiple emergency contacts to maximize the chances of receiving immediate help.

2. **Access to Emergency Services:** Incorporate features to find nearby police stations and hospitals, providing users with quick access to essential assistance. Include direct calling facilities for national helplines to ensure immediate contact with emergency services.

3. **Empower with Knowledge:** Offer information on women safety laws to educate and empower users about their rights and legal protections. Provide short self-defense videos to equip users with practical techniques and guidance in times of emergency.

4. **User-Friendly Interface:** Develop an intuitive and user-friendly interface that promotes ease of use and accessibility for women of all backgrounds and age groups. Prioritize simplicity, responsiveness, and visual appeal in the application design.

1.2. Scope of the Project

1. Emergency Alerts and Communication:

Develop a shake detector and panic button to trigger SOS alerts.

Send emergency messages with real-time location to registered contacts.

Implement a siren sound feature to attract attention during emergencies.

2. Location-Based Services:

Integrate GPS to share the user's location with contacts.

Provide the ability to add multiple emergency contacts.

Implement functionality to find and show nearby police stations and hospitals on a map.

3. Direct Access to Emergency Services:

Integrate one-click call functionality for police, hospitals, and national helplines.

Develop a feature to display the nearest emergency service locations.

4. Education and Empowerment:

Provide easy access to information on women's safety laws and rights.

Offer short, practical self-defense videos within the app to educate users on protective techniques.

5. User Experience and Accessibility:

Develop an intuitive and user-friendly interface that is accessible to women of all ages. Focus on simplicity, responsiveness, and visual appeal in the design.

Ensure compatibility across various Android devices and screen sizes.

6. Security and Privacy:

Implement secure registration and login for users.

Ensure data privacy by safeguarding users' personal and location information.

7. **Maintenance and Future Scalability:**

Plan for regular updates to include new emergency features, legal updates, and user feedback improvements. Ensure the app can scale by adding more features, like integration with local authorities or wearables, in the future.

1.3. Existing System and Drawbacks

1. **Basic Emergency Alerts:**

   Most apps provide basic SOS buttons or panic alerts but may lack integration with multiple emergency contacts or GPS-based location sharing.

2. **Limited Location Services:**

   Apps typically offer GPS tracking, but many don't update locations in real-time or lack the ability to find nearby emergency services like police stations or hospitals.

3. **Access to Emergency Contacts:**

   Some apps allow users to contact pre-registered emergency numbers, but features like direct calling for helplines and local authorities are not always available.

4. **Awareness and Education:**

   While some apps provide information on safety laws or self-defense techniques, this feature is often minimal or non-interactive.

5. **User Interface:**

   The interfaces of existing systems may not prioritize ease of use, and many apps fail to accommodate diverse user groups, such as older women or those less tech-savvy.

6. **Security Concerns:**

   Many apps don't offer robust data privacy protections, leaving user information vulnerable.

Chapter 2

2.1 Literature Survey

S.No	Authors	Title	Description	Drawbacks	Year
1	Ravi Sekhar Yarrabothula, Bramarambika Thota	ABHAYA: AN ANDROID APP FOR THE SAFETY OF WOMEN	Android app with GPS tracking and emergency alerts; user-friendly interface.	Limited functionality may exclude some users.	2015
2	Alisha Maruti Gawade, Amruta Jadhav, Sachin Shankar Kumbhar	S-ZONE: A SYSTEM FOR WOMEN SAFETY & SECURITY SYSTEM	Comprehensive system with mobile app and hardware (panic buttons); real-time location tracking.	High hardware costs; dependency on network availability.	2016
3	Sagar Khan, Harish Shinde, Ankita Zaroo, Rashmi Koushik	SHIELD: Personal Safety Application	App for emergency messages and safety tips; location data integration.	Usability issues in emergencies; relies on user engagement.	2017
4	Piyush Bhanushali, Rahul Mange, Dama Paras, Prof. Chitra Bhole	Women Safety Android App	Simple app with emergency alerts and voice commands.	Limited features; battery and device accessibility concerns.	2018
5	N. Ramesh Kannan, S. Sujitha, S. Ganapathy Subramanian	Women Safety Mobile App	Safety alerts and emergency contacts; focuses on user awareness and education.	Limited awareness campaigns; may lack offline functionality in poor connectivity areas.	2021

Table 2.1.1: Literature survey

[1]. ABHAYA: AN ANDROID APP FOR THE SAFETY OF WOMEN – This app focuses on enhancing women's safety through features like GPS tracking and emergency alerts. It offers a user-friendly interface for quick access to safety features.

Drawbacks: It has limited functionality beyond emergency alerts and relies heavily on smartphones, which may exclude users without access to them.

[2]. S-ZONE: A SYSTEM FOR WOMEN SAFETY & SECURITY SYSTEM – This system integrates a mobile app with hardware components such as panic buttons. It includes real-time location tracking and alerts to family members.

Drawbacks: The implementation can be costly due to the hardware and is dependent on network availability for proper functionality.

[3]. SHIELD: Personal Safety Application – SHIELD is designed to send emergency messages with location data and includes safety tips and resources.
Drawbacks: The app may have usability issues in critical situations due to complex navigation and relies on regular user engagement.

[4]. Women Safety Android App – This is a straightforward app offering emergency alert features with voice command options for ease of use.

Drawbacks: It offers limited features beyond basic alerts and is dependent on battery life and device accessibility, which can be problematic in emergencies.

[5]. Women Safety Mobile App – The app provides safety alerts and emergency contact options, emphasizing user awareness and engagement through educational content.

Drawbacks: It may have limited awareness campaigns, affecting its uptake, and it could lack offline functionality in areas with poor connectivity.

2.2 Problem Statement

The safety and security of women continue to be a pressing concern in today's society. Women face various risks, including harassment, violence, and discrimination, which hinder their ability to live freely and confidently. Despite ongoing efforts to address these issues, there is still a need for comprehensive and accessible solutions that empower women to protect themselves and seek help when needed. Existing safety measures and resources are often fragmented and lack integration, making it challenging for women to access timely assistance in emergencies. Additionally, the lack of awareness about women's safety laws and self-defense techniques further exacerbates the problem.

Therefore, there is a critical need for a comprehensive women safety application that leverages the power of technology to provide women with a reliable and user-friendly tool for immediate help, information, and empowerment. This project aims to develop such an application, utilizing Android Studio, Java, and XML, to address the multifaceted challenges faced by women in terms of safety and security.

Chapter 3

3.1 Proposed Method

1. **Enhanced Emergency Alerts:**

Implement a shake detector and panic button to send SOS alerts with real-time GPS location to multiple registered contacts. Include a siren sound to draw attention during emergencies.

2. **Real-Time Location Services:**

Continuously track and share the user's live location with emergency contacts. Enable users to locate nearby police stations and hospitals.

3. **Direct Access to Emergency Services:**

Provide one-click calling for local authorities, police stations, hospitals, and national helplines for immediate assistance.

4. **Comprehensive Educational Support:**

Offer detailed information on women's safety laws and rights. Include interactive self-defense videos for quick guidance during emergencies.

5. **Intuitive User Interface:**

Develop a simple, accessible, and responsive interface tailored to women of all ages, making it easy to navigate even under stress.

6. **Robust Security and Privacy:**

Ensure data privacy with strong encryption for user information, GPS location, and emergency contacts. Implement secure authentication methods for user logins.

7. **Integrated System:**

Combine emergency alerts, location services, education, and emergency contact access into one seamless platform, providing comprehensive safety solutions in one app.

3.2 System Architecture

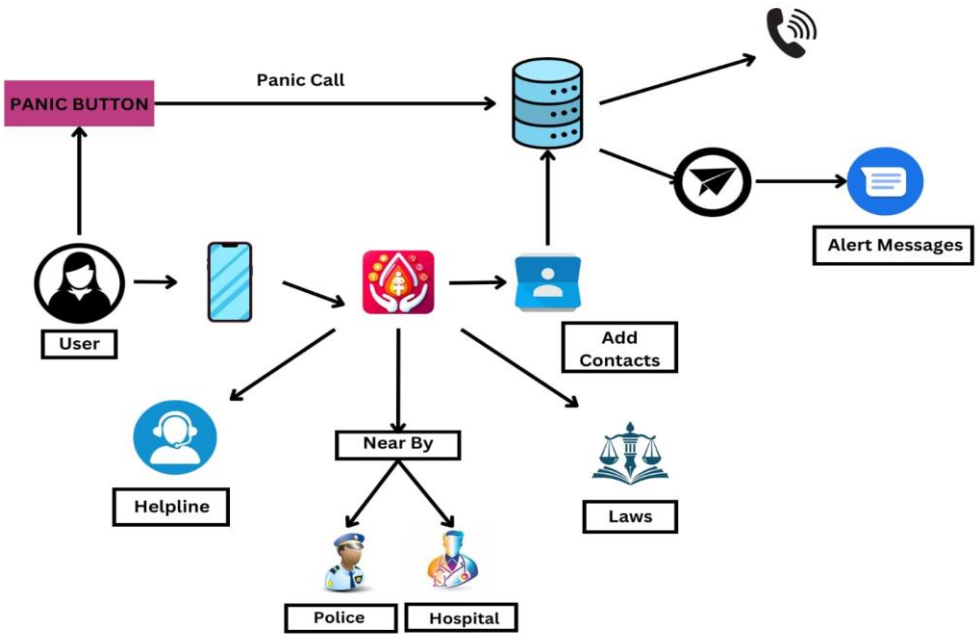

Fig. 3.2.1: System Architecture

1. Client-Side (Android Device)

- **User Interface (UI):**

  The application runs on Android OS devices (e.g., mobile phones) and presents a user-friendly interface with four button cards:

  Contact: Manages emergency contacts.

  SMS Alerts: Sends SOS messages to contacts.

  Women Laws: Displays legal information.

  Self-Defense: Provides access to self-defense resources.

- **Permissions:**

  Location Permission: Used for GPS tracking of the user's location.

  Phone and SMS Permission: Required for sending emergency alerts and making calls.

  Notifications: Allows push notifications for safety alerts.

- **Emergency Contacts Database:**

  Users can add, store, and access emergency contacts in the app's local database.

  Includes a high-speed phone contacts cache for quick retrieval.

2. Backend Services

- **Location and GPS Tracking:**

  Retrieves the user's current location using GPS.

  Uses Map APIs to identify and display nearby police stations and hospitals.

- **Accelerometer Integration:**

  Utilizes the Android device's accelerometer to detect phone shaking. If the phone is shaken five times consecutively, the app automatically makes emergency calls. Sends SMS alerts with location details to registered emergency contacts.

- **National Helplines:**

  Pre-configured with national emergency helpline numbers for Women in Distress, Domestic Abuse, Police, Student/Child Help, and Ambulance.

  Users can directly call these services from the app.

3. External APIs and Communication

- **SMS Alerts:**

  Uses the user's phone service to send messages (charges based on mobile plan).

- **Map API:**

  Searches and displays nearby assistance services (e.g., police stations, hospitals).

4. Event-Driven Emergency Features

- **Shake Detection:**

  Five consecutive shakes trigger the emergency mode, activating SMS alerts and calls.

- **SOS Alerts:**

  Sends an emergency message with the user’s location to all registered contacts, ensuring immediate support.

3.3 Module Description

1. Main Activity Module:

- **Purpose:** Serves as the entry point and home screen of the app.
- **Functionality:**
  Provides navigation to all key features like Contact Management, SMS Alerts, Women's Laws, Self-Defense Videos, and Panic Button. Uses buttons with onClickListeners to navigate to different activities (e.g., ContactActivity, SmsActivity, LawsActivity).
- **Key Components:**
  Buttons for feature selection.
  Activity transitions and intent handling for navigation.

2. Laws Activity Module:

- **Purpose:** Displays information about women's safety laws.
- **Functionality:**
  Fetches and displays a list of women's safety laws. Ensures content is presented in a user-friendly and readable format.
- **Key Components:**
  Static content or dynamically fetched content (if needed).
  Scrollable text view or card view to present the laws in a clean layout.

3. Contact Activity Module:

- **Purpose:** Manages emergency contacts that will be alerted during emergencies.
- **Functionality:**
  Allows users to add, edit, and delete emergency contacts. Stores contacts persistently using SharedPreferences. Displays the list of contacts in a RecyclerView.
- **Key Components:**
  Form input for contact details (name, phone number).
  Shared Preferences for data persistence.
  Recycler View to display the list of contacts dynamically.

4. Self Defense Activity Module:

- **Purpose:** Provides access to self-defense instructional videos.
- **Functionality:**

  Embeds a WebView to show videos (hosted on platforms like YouTube) that teach self-defense techniques. Ensures a smooth video viewing experience within the app.
- **Key Components:**

  WebView for rendering online video content.

  URL management to link to relevant self-defense resources.

5. SMS Activity Module:

- **Purpose:** Enables SMS-based emergency alert functionality.
- **Functionality:**

  Sends pre-configured SMS alerts to emergency contacts in case of distress. Requests necessary permissions for SMS functionality. Provides a helpline button to send an SOS alert with a single click.
- **Key Components:**

  Permission requests for sending SMS.

  SMS sending functionality through Android's SmsManager.

  Helpline button to quickly send alerts.

6. Panic Button Activity Module:

- **Purpose:** A dedicated panic button that sends an emergency SOS message.
- **Functionality:**

  The panic button sends an SOS message with the user's GPS location to emergency contacts. Checks for necessary permissions (e.g., location, SMS). Uses GPS to get the current location and include it in the message.
- **Key Components:**

  Panic button UI.

  Location services for GPS tracking.

  Integration with the SMS system for sending SOS messages.

7. Shake Detection Module:

- **Purpose:** Detects shake gestures to trigger the panic button functionality.
- **Functionality:**

  Listens for shake gestures (using the accelerometer sensor) to automatically activate the panic button.

  Once triggered, sends SOS messages with the user's location.
- **Key Components:**

  Shake detection logic/library.

  Integration with the PanicButtonActivity for automatic triggering.

8. GPS Module:

- **Purpose:** Tracks and uses the user's location during emergencies.
- **Functionality:**

  Requests location permissions (both foreground and background).

  Fetches the user's current GPS location and passes it to the SmsActivity and PanicButtonActivity.
- **Key Components:**

  LocationManager or FusedLocationProvider to retrieve the user's GPS coordinates.

  Location permissions for access.

3.4 UML Diagrams

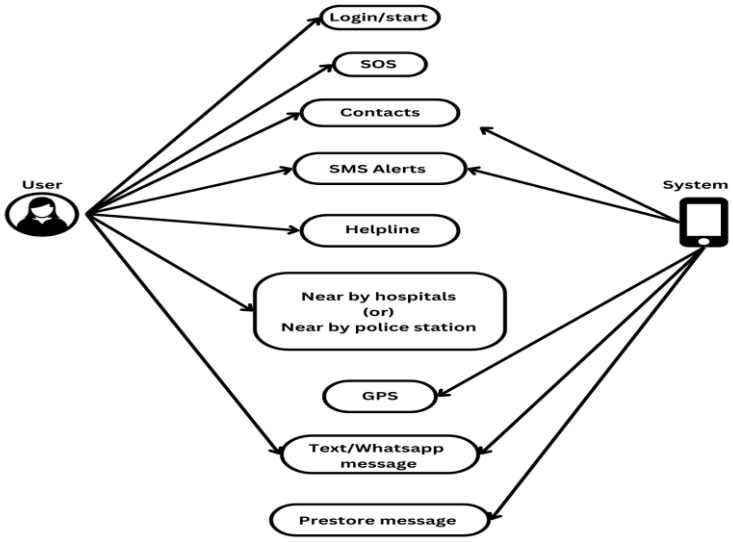

Fig 3.4.1: Use Case Diagram

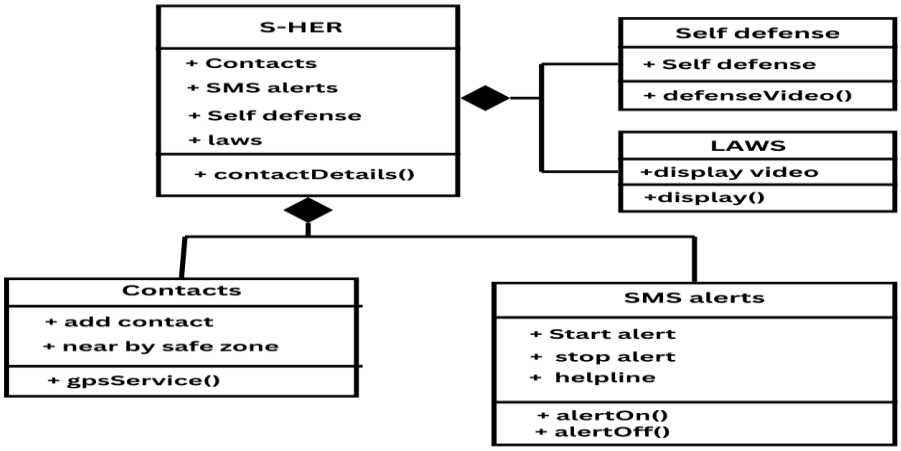

Fig 3.4.2: Class Diagram

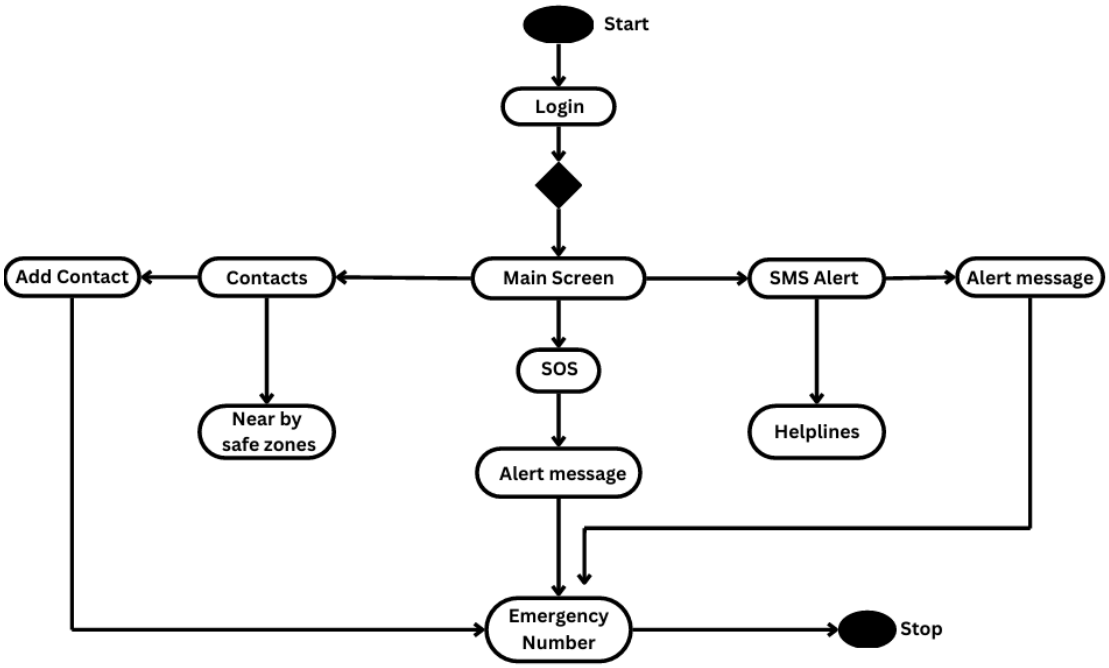

Fig 3.4.3: State Transition Diagram

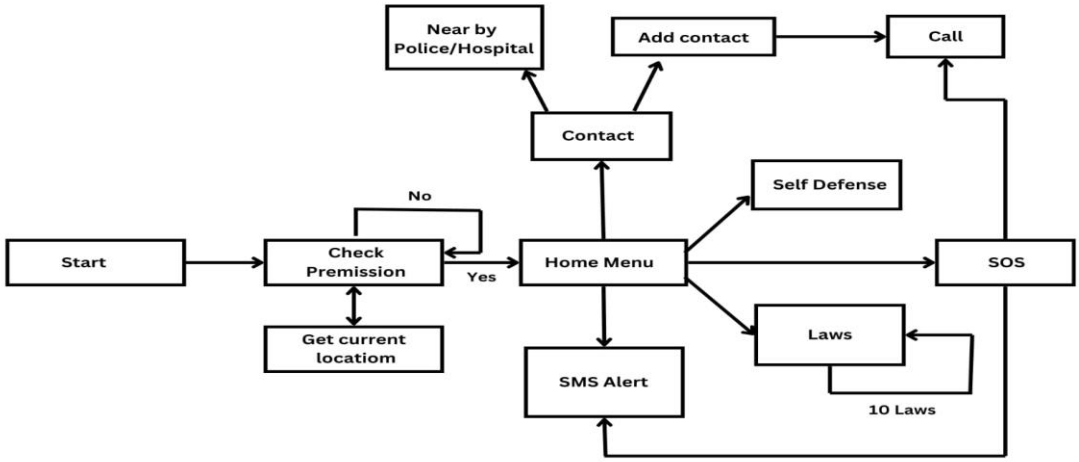

Fig 3.4.3: Activity Diagram

Chapter 4

4.1 Implementation

1. Set Up Development Environment

- Install **Android Studio**.
- Install **JDK** (Java Development Kit) version 8 or newer.
- Ensure the **Android SDK** is up to date

2. Create a New Project

- Open Android Studio and create a new project named **W-Safe**.
- Choose an **Empty Activity** template.

3. Build Core Features

- **MainActivity**:
  - Create the home screen with buttons for features like Contact, SMS Alerts, Women Laws, Self-Defense, and Panic Button.
  - Set up button clicks to open the corresponding screens.
- **LawsActivity**:
  - Design the screen to display women's safety laws.
  - Fetch and show data in a user-friendly way.
- **ContactActivity**:
  - Create a screen to manage emergency contacts.
  - Use **SharedPreferences** to store contact details.
  - Display contacts using a **RecyclerView**.

- **SelfDefenseActivity**:
  - Use a **WebView** to show self-defense instructional videos.
- **SmsActivity**:
  - Enable SMS functionality and request permissions.
  - Add options for sending SMS alerts and a helpline button.

- **PanicButtonActivity**:
  - Design the panic button screen.
  - Check location permissions and get the user's location.
  - Send an SOS message with the location to emergency contacts.

4. Add Shake Detection

- Include the **Shake Detector** library in your project.
- Set up shake detection to trigger the Panic Button.

5. **Integrate GPS**

- Request location permissions.
- Use GPS to track the user's location for emergencies.

6. **Test the Application**

- Test on emulators created in Android Studio.
- Also test on real Android devices for performance.

7. **Enhance User Interface**

- Make the interface visually appealing and easy to navigate.
- Ensure it works well on different screen sizes.

4.2 Code

MainActivity.java

```
package com.darkness.WSafety;

import androidx.appcompat.app.AppCompatActivity;
import androidx.core.app.ActivityCompat;
import android.Manifest;
import android.content.Intent;
import android.content.SharedPreferences;
import android.content.pm.PackageManager;
import android.net.Uri;
import android.os.Bundle;
import android.telephony.SmsManager;
import android.view.View;
import com.google.android.gms.location.FusedLocationProviderClient;
import com.google.android.gms.location.LocationServices;

import java.util.HashSet;
import java.util.Set;

public class MainActivity extends AppCompatActivity implements
View.OnClickListener {

    FusedLocationProviderClient fusedLocationClient;
    String myLocation = "", numberCall;
    SmsManager manager = SmsManager.getDefault();

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

        fusedLocationClient =
LocationServices.getFusedLocationProviderClient(this);
        findViewById(R.id.panicBtn).setOnClickListener(this);
        findViewById(R.id.fourth).setOnClickListener(this);
        findViewById(R.id.first).setOnClickListener(this);
        findViewById(R.id.second).setOnClickListener(this);
        findViewById(R.id.fifth).setOnClickListener(this);
    }

    @Override
    public void onClick(View view) {
        int id = view.getId();
        if (id == R.id.fourth) {
            startActivity(new Intent(MainActivity.this, LawsActivity.class));
```

```
            MainActivity.this.finish();
        } else if (id == R.id.first) {
            startActivity(new Intent(MainActivity.this, ContactActivity.class));
            MainActivity.this.finish();
        } else if (id == R.id.fifth) {
            startActivity(new Intent(MainActivity.this, SelfDefenseActivity.class));
        } else if (id == R.id.second) {
            startActivity(new Intent(MainActivity.this, SmsActivity.class));
            MainActivity.this.finish();
        } else if (id == R.id.panicBtn) {
            if (id == R.id.panicBtn) {
                if (ActivityCompat.checkSelfPermission(this,
Manifest.permission.ACCESS_FINE_LOCATION) !=
PackageManager.PERMISSION_GRANTED &&
ActivityCompat.checkSelfPermission(this,
Manifest.permission.ACCESS_COARSE_LOCATION) !=
PackageManager.PERMISSION_GRANTED) {
                    return;
                }
                fusedLocationClient.getLastLocation()
                        .addOnSuccessListener(location -> {
                            if (location != null) {
                                location.getAltitude();
                                location.getLongitude();
                                myLocation = "http://maps.google.com/maps?q=loc:" +
location.getLatitude() + "," + location.getLongitude();
                            } else {
                                myLocation = "Unable to Find Location :(";
                            }
                            sendMsg();
                        });

                SharedPreferences sharedPreferences =
getSharedPreferences("MySharedPref", MODE_PRIVATE);
                numberCall = sharedPreferences.getString("firstNumber", "None");
                if (!numberCall.equalsIgnoreCase("None")) {
                    Intent intent = new Intent(Intent.ACTION_CALL);
                    intent.setData(Uri.parse("tel:" + numberCall));
                    startActivity(intent);
                }
            }
        }
    }
    void sendMsg() {
        SharedPreferences sharedPreferences =
getSharedPreferences("MySharedPref", MODE_PRIVATE);
        Set<String> oldNumbers = sharedPreferences.getStringSet("enumbers", new
HashSet<>());
```

```
        if (!oldNumbers.isEmpty()) {
            for (String ENUM : oldNumbers) {
                manager.sendTextMessage(ENUM, null, "Help, I am in Trouble!\nI'm
sharing my current Location :\n" + myLocation, null, null);
            }
        }
    }
}
```

SmsActivity.java

```
package com.darkness.WSafety;

import android.net.Uri; // Import Uri
import java.util.HashSet;
import java.util.Set;
import android.Manifest; // Make sure to import Manifest
import android.content.Context;
import android.content.Intent;
import android.content.SharedPreferences;
import android.content.pm.PackageManager;
import android.location.Location; // Add this import for Location
import android.os.Build;
import android.os.Bundle;
import android.telephony.SmsManager;
import android.provider.Settings;
import android.util.Log;
import android.view.View;
import android.widget.Button;
import android.widget.ToggleButton;
import androidx.appcompat.app.AppCompatActivity;
import androidx.core.app.ActivityCompat;
import androidx.core.content.ContextCompat;
import com.google.android.gms.location.FusedLocationProviderClient;
import com.google.android.gms.location.LocationServices;
import com.google.android.gms.tasks.OnSuccessListener;
import com.google.android.material.snackbar.Snackbar;


public class SmsActivity extends AppCompatActivity {

    private ToggleButton toggleAlert;
    private FusedLocationProviderClient fusedLocationClient;
    private android.os.Handler alertHandler; // Use android.os.Handler
    private Runnable alertRunnable;
    private boolean isAlertActive = false; // Track alert state

    @Override
```

```
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_sms);

        toggleAlert = findViewById(R.id.toggleAlert);
        fusedLocationClient =
LocationServices.getFusedLocationProviderClient(this);
        alertHandler = new android.os.Handler(); // Specify the Handler correctly

        toggleAlert.setOnCheckedChangeListener((buttonView, isChecked) -> {
            if (isChecked) {
                startAlertService();
                sendSmsAlert(); // Send alert immediately when toggled ON
            } else {
                stopAlertService(); // Just stop service when toggled OFF
            }
        });

        // Initialize buttons for starting and stopping services
        Button start, stop, helpline;
        stop = findViewById(R.id.stopService);
        start = findViewById(R.id.startService);
        helpline = findViewById(R.id.btn_helpline);
        start.setOnClickListener(this::startServiceV);
        stop.setOnClickListener(this::stopService);
        helpline.setOnClickListener(this::helplines);
    }

    private void startAlertService() {
        isAlertActive = true; // Set alert state
        Log.d("SmsActivity", "Alert service started.");

        // Set up a runnable to send SMS alerts every 10 minutes
        alertRunnable = new Runnable() {
            @Override
            public void run() {
                if (isAlertActive) {
                    sendSmsAlert(); // Send the alert message
                    alertHandler.postDelayed(this, 10 * 60 * 1000); // Repeat every 10
minutes
                }
            }
        };

        alertHandler.post(alertRunnable); // Start the alert runnable
    }

    private void stopAlertService() {
```

```
        isAlertActive = false; // Clear alert state
        alertHandler.removeCallbacks(alertRunnable); // Stop the runnable
        Log.d("SmsActivity", "Alert service stopped.");
    }

    private void sendSmsAlert() {
        // Check for permissions before accessing the location
        if (ActivityCompat.checkSelfPermission(this,
Manifest.permission.ACCESS_FINE_LOCATION) !=
PackageManager.PERMISSION_GRANTED) {
            // Request permission if not granted (you should handle this in your app)
            return;
        }

        // Get the last known location
        fusedLocationClient.getLastLocation()
                .addOnSuccessListener(this, new OnSuccessListener<Location>() {
                    @Override
                    public void onSuccess(Location location) {
                        if (location != null) {
                            String currentLocation = "http://maps.google.com/maps?q=loc:"
+ location.getLatitude() + "," + location.getLongitude();
                            sendSmsToEmergencyContacts(currentLocation);
                        } else {
                            Log.d("SmsActivity", "Location not found.");
                        }
                    }
                });
    }

    private void sendSmsToEmergencyContacts(String location) {
        SharedPreferences sharedPreferences =
getSharedPreferences("MySharedPref", Context.MODE_PRIVATE);
        Set<String> emergencyNumbers =
sharedPreferences.getStringSet("enumbers", new HashSet<>());
        SmsManager smsManager = SmsManager.getDefault();

        String message = "I am in trouble! Please help! Here is my location: " +
location;
        for (String number : emergencyNumbers) {
            smsManager.sendTextMessage(number, null, message, null, null);
        }
        Log.d("SmsActivity", "Alert sent to emergency contacts with location.");
    }

    @Override
    public void onBackPressed() {
        super.onBackPressed();
```

```
        startActivity(new Intent(SmsActivity.this, MainActivity.class));
    }

    public void helplines(View view) {
        startActivity(new Intent(SmsActivity.this, HelplineCall.class));
    }

    public void stopService(View view) {
        Intent notificationIntent = new Intent(this, ServiceMine.class);
        notificationIntent.setAction("stop");
        if (Build.VERSION.SDK_INT >= Build.VERSION_CODES.O) {
            if (ServiceMine.isRunning) {
                getApplicationContext().startForegroundService(notificationIntent);
                Snackbar.make(findViewById(android.R.id.content), "Service
Stopped!", Snackbar.LENGTH_LONG).show();
            }
        } else {
            if (ServiceMine.isRunning) {
                getApplicationContext().startService(notificationIntent);
                Snackbar.make(findViewById(android.R.id.content), "Service
Stopped!", Snackbar.LENGTH_LONG).show();
            }
        }
    }

    public void startServiceV(View view) {
        if (Build.VERSION.SDK_INT >= Build.VERSION_CODES.O) {
            if (!Settings.canDrawOverlays(this)) {
                Intent intent = new
Intent(Settings.ACTION_MANAGE_OVERLAY_PERMISSION,
                        Uri.parse("package:" + getPackageName()));
                startActivity(intent);
            }
        }

        if (ContextCompat.checkSelfPermission(this,
Manifest.permission.SEND_SMS) ==
PackageManager.PERMISSION_GRANTED
                && ContextCompat.checkSelfPermission(this,
Manifest.permission.ACCESS_COARSE_LOCATION) ==
PackageManager.PERMISSION_GRANTED
                && ContextCompat.checkSelfPermission(this,
Manifest.permission.ACCESS_FINE_LOCATION) ==
PackageManager.PERMISSION_GRANTED) {
            Intent notificationIntent = new Intent(this, ServiceMine.class);
            notificationIntent.setAction("Start");
            if (Build.VERSION.SDK_INT >= Build.VERSION_CODES.O) {
                getApplicationContext().startForegroundService(notificationIntent);
```

```
            Snackbar.make(findViewById(android.R.id.content), "Service
Started!", Snackbar.LENGTH_LONG).show();
        } else {
            getApplicationContext().startService(notificationIntent);
            Snackbar.make(findViewById(android.R.id.content), "Service
Started!", Snackbar.LENGTH_LONG).show();
        }
    }
  }
}
```

SplashActivity.java

```
package com.darkness.WSafety;

import androidx.appcompat.app.AppCompatActivity;
import androidx.core.app.ActivityCompat;

import android.Manifest;
import android.content.Intent;
import android.content.pm.PackageManager;
import android.net.Uri; // Import the Uri class
import android.os.Bundle;
import android.telephony.SmsManager;
import android.util.Log;
import android.widget.Toast;

import com.google.android.gms.common.api.ResolvableApiException;
import com.google.android.gms.location.LocationCallback;
import com.google.android.gms.location.LocationRequest;
import com.google.android.gms.location.LocationResult;
import com.google.android.gms.location.LocationServices;
import com.google.android.gms.location.LocationSettingsRequest;
import com.google.android.gms.location.LocationSettingsResponse;
import com.google.android.gms.location.SettingsClient;
import com.google.android.gms.tasks.Task;
import com.karumi.dexter.Dexter;
import com.karumi.dexter.MultiplePermissionsReport;
import com.karumi.dexter.PermissionToken;
import com.karumi.dexter.listener.PermissionRequest;
import com.karumi.dexter.listener.multi.MultiplePermissionsListener;

import java.util.List;

public class SplashActivity extends AppCompatActivity {
    boolean isAllPermissionsGranted = false;
    LocationCallback mLocationCallback;
```

```
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_splash);
        requestPermission();
        setupLocationCallback();

        findViewById(R.id.btnGetStarted).setOnClickListener(view -> {
            if (isAllPermissionsGranted) {
                startActivity(new Intent(SplashActivity.this, MainActivity.class));
                SplashActivity.this.finish();
            } else {
                Toast.makeText(this, "Please grant required permissions!",
Toast.LENGTH_SHORT).show();
                requestPermission();
            }
        });
    }

    private void requestPermission() {
        Dexter.withContext(this)
                .withPermissions(
                        Manifest.permission.ACCESS_FINE_LOCATION,
                        Manifest.permission.ACCESS_COARSE_LOCATION,
                        Manifest.permission.CALL_PHONE,
                        Manifest.permission.SEND_SMS
                ).withListener(new MultiplePermissionsListener() {
            @Override
            public void onPermissionsChecked(MultiplePermissionsReport report) {
                if (report.areAllPermissionsGranted()) {
                    isAllPermissionsGranted = true;
                    requestLocation();
                } else {
                    isAllPermissionsGranted = false;
                    Toast.makeText(SplashActivity.this, "Permissions denied.",
Toast.LENGTH_SHORT).show();
                }
            }

            @Override
            public void
onPermissionRationaleShouldBeShown(List<PermissionRequest> permissions,
PermissionToken token) {
                token.continuePermissionRequest();
            }
        }).check();
    }
```

```
    private void setupLocationCallback() {
        mLocationCallback = new LocationCallback() {
            @Override
            public void onLocationResult(LocationResult locationResult) {
                if (locationResult != null && locationResult.getLastLocation() != null)
{
                    Log.d("LocationUpdate", "Location: " +
locationResult.getLastLocation().toString());
                }
            }
        };
    }

    void requestLocation() {
        LocationRequest mLocationRequest = LocationRequest.create();
        mLocationRequest.setInterval(60000); // 1 minute interval
        mLocationRequest.setFastestInterval(5000); // 5 seconds interval

mLocationRequest.setPriority(LocationRequest.PRIORITY_HIGH_ACCURAC
Y);

        LocationSettingsRequest.Builder builder = new
LocationSettingsRequest.Builder()
                .addLocationRequest(mLocationRequest);

        SettingsClient client = LocationServices.getSettingsClient(this);
        Task<LocationSettingsResponse> task =
client.checkLocationSettings(builder.build());

        task.addOnSuccessListener(locationSettingsResponse -> {
            // All location settings are satisfied, start location updates
            if (ActivityCompat.checkSelfPermission(this,
Manifest.permission.ACCESS_FINE_LOCATION) !=
PackageManager.PERMISSION_GRANTED
                    && ActivityCompat.checkSelfPermission(this,
Manifest.permission.ACCESS_COARSE_LOCATION) !=
PackageManager.PERMISSION_GRANTED) {
                return;
            }

LocationServices.getFusedLocationProviderClient(SplashActivity.this).requestLo
cationUpdates(mLocationRequest, mLocationCallback, null);
        });

        task.addOnFailureListener(e -> {
            if (e instanceof ResolvableApiException) {
                try {
```

```
                // Show the dialog by calling startResolutionForResult(), and check
the result in onActivityResult().
                ResolvableApiException resolvable = (ResolvableApiException) e;
                resolvable.startResolutionForResult(SplashActivity.this, 100);
            } catch (Exception ex) {
                ex.printStackTrace();
            }
        }
    });
  }

  @Override
  protected void onActivityResult(int requestCode, int resultCode, Intent data) {
    super.onActivityResult(requestCode, resultCode, data);
    if (requestCode == 100) {
      if (resultCode == RESULT_OK) {
        // User agreed to enable location services, request location updates
again
        requestLocation();
      } else {
        Toast.makeText(this, "Please enable location services",
Toast.LENGTH_SHORT).show();
      }
    }
  }

  @Override
  protected void onDestroy() {
    super.onDestroy();
    // Remove location updates to save battery
    if (mLocationCallback != null) {

LocationServices.getFusedLocationProviderClient(this).removeLocationUpdates(
mLocationCallback);
    }
  }

  // Method to make a call
  private void makePhoneCall(String phoneNumber) {
    if (ActivityCompat.checkSelfPermission(this,
Manifest.permission.CALL_PHONE) !=
PackageManager.PERMISSION_GRANTED) {
      return; // Permission not granted
    }
    Intent callIntent = new Intent(Intent.ACTION_CALL);
    callIntent.setData(Uri.parse("tel:" + phoneNumber));
    startActivity(callIntent);
  }
```

```
    // Method to send an SMS
    private void sendSMS(String phoneNumber, String message) {
        if (ActivityCompat.checkSelfPermission(this,
Manifest.permission.SEND_SMS) !=
PackageManager.PERMISSION_GRANTED) {
            return; // Permission not granted
        }
        SmsManager smsManager = SmsManager.getDefault();
        smsManager.sendTextMessage(phoneNumber, null, message, null, null);
        Toast.makeText(this, "SMS Sent!", Toast.LENGTH_SHORT).show();
    }
}
```

AlertReceiver.java

```
package com.darkness.WSafety;

import android.Manifest;
import android.app.PendingIntent;
import android.content.BroadcastReceiver;
import android.content.Context;
import android.content.Intent;
import android.content.SharedPreferences;
import android.location.Location;
import android.telephony.SmsManager;
import android.util.Log;
import android.content.pm.PackageManager; // Import PackageManager

import androidx.core.app.ActivityCompat;
import com.google.android.gms.location.FusedLocationProviderClient;
import com.google.android.gms.location.LocationServices;
import com.google.android.gms.tasks.OnSuccessListener;

import java.util.HashSet;
import java.util.Set;

public class AlertReceiver extends BroadcastReceiver {
    private FusedLocationProviderClient fusedLocationClient;

    @Override
    public void onReceive(Context context, Intent intent) {
        Log.d("AlertReceiver", "Periodic alert triggered.");
        sendPeriodicAlert(context);
    }

    private void sendPeriodicAlert(Context context) {
```

```
        fusedLocationClient =
LocationServices.getFusedLocationProviderClient(context);

        // Check location permission
        if (ActivityCompat.checkSelfPermission(context,
Manifest.permission.ACCESS_FINE_LOCATION) !=
PackageManager.PERMISSION_GRANTED) {
            Log.d("AlertReceiver", "Location permission not granted.");
            return; // Handle permission request in your activity
        }

        fusedLocationClient.getLastLocation()
                .addOnSuccessListener(location -> {
                    if (location != null) {
                        String currentLocation = "http://maps.google.com/maps?q=loc:" +
location.getLatitude() + "," + location.getLongitude();
                        sendSmsToEmergencyContacts(context, currentLocation);
                    } else {
                        Log.d("AlertReceiver", "Location not found.");
                    }
                });
    }

    private void sendSmsToEmergencyContacts(Context context, String location)
{
        SharedPreferences sharedPreferences =
context.getSharedPreferences("MySharedPref", Context.MODE_PRIVATE);
        Set<String> emergencyNumbers =
sharedPreferences.getStringSet("enumbers", new HashSet<>());
        SmsManager smsManager = SmsManager.getDefault();

        if (!emergencyNumbers.isEmpty()) {
            for (String number : emergencyNumbers) {
                String message = "I am in trouble! Here is my location: " + location;
                try {
                    smsManager.sendTextMessage(number, null, message, null, null);
                    Log.d("AlertReceiver", "Alert sent to: " + number);
                } catch (Exception e) {
                    Log.e("AlertReceiver", "Failed to send SMS to: " + number + "
Error: " + e.getMessage());
                }
            }
        } else {
            Log.d("AlertReceiver", "No emergency contacts found.");
        }
    }
}
```

MainActivity.xml

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:background="#F0FFF0"
    tools:context=".MainActivity">

    <View
        android:id="@+id/topleftoval"
        android:layout_width="150dp"
        android:layout_height="150dp"
        android:background="@drawable/top_left_corner_oval"
        app:layout_constraintTop_toTopOf="parent"
        app:layout_constraintStart_toStartOf="parent" />
    <TextView
        android:id="@+id/title_view"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_centerHorizontal="true"
        android:layout_marginStart="12dp"
        android:layout_marginTop="120dp"
        android:layout_marginEnd="12dp"
        android:layout_marginBottom="-5dp"
        android:text="S-HER"
        android:textColor="@color/purple_700"
        android:textSize="40sp"
        android:textStyle="bold" />

    <GridLayout
        android:id="@+id/gridlayout"
        android:layout_width="400dp"
        android:layout_height="400dp"
        android:layout_below="@id/title_view"
        android:layout_marginStart="20dp"
        android:layout_marginTop="40dp"
        android:layout_marginEnd="20dp"
        android:layout_marginBottom="0dp"
        android:layout_weight="7"
        android:columnCount="2"
        android:rowCount="2">


        <androidx.cardview.widget.CardView
```

```
    android:layout_width="wrap_content"
    android:layout_height="58dp"
    android:layout_row="0"
    android:layout_rowWeight="1"
    android:layout_column="0"
    android:layout_columnWeight="1"
    android:layout_gravity="fill"
    android:id="@+id/first"
    app:cardCornerRadius="12dp"
    app:cardElevation="14dp"
    app:cardUseCompatPadding="true">

    <LinearLayout
      android:layout_width="wrap_content"
      android:layout_height="wrap_content"
      android:layout_gravity="center_vertical|center_horizontal"
      android:gravity="center"
      android:orientation="vertical">

      <ImageView
        android:layout_width="50dp"
        android:layout_height="50dp"
        android:src="@drawable/contact" />

      <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="CONTACT"
        android:textAlignment="center"
        android:textStyle="bold" />
    </LinearLayout>
  </androidx.cardview.widget.CardView>

  <androidx.cardview.widget.CardView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_row="0"
    android:layout_rowWeight="1"
    android:layout_column="1"
    android:layout_columnWeight="1"
    android:layout_gravity="fill"
    android:id="@+id/second"
    app:cardCornerRadius="12dp"
    app:cardElevation="14dp"
    app:cardUseCompatPadding="true">

    <LinearLayout
      android:layout_width="wrap_content"
```

```
        android:layout_height="wrap_content"
        android:layout_gravity="center_vertical|center_horizontal"
        android:gravity="center"
        android:orientation="vertical">

        <ImageView
            android:layout_width="50dp"
            android:layout_height="50dp"
            android:src="@drawable/ic_baseline_message_24" />

        <TextView
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="SMS ALERT"
            android:textAlignment="center"
            android:textStyle="bold" />
    </LinearLayout>
</androidx.cardview.widget.CardView>

<androidx.cardview.widget.CardView
    android:id="@+id/fourth"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_row="1"
    android:layout_rowWeight="1"
    android:layout_column="0"
    android:layout_columnWeight="1"
    android:layout_gravity="fill"
    app:cardCornerRadius="12dp"
    app:cardElevation="14dp"
    app:cardUseCompatPadding="true">

    <LinearLayout
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_gravity="center_vertical|center_horizontal"
        android:gravity="center"
        android:orientation="vertical">

        <ImageView
            android:layout_width="50dp"
            android:layout_height="50dp"
            android:src="@drawable/law_image" />

        <TextView
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="WOMEN LAWS"
```

```
            android:textAlignment="center"
            android:textStyle="bold" />
        </LinearLayout>
    </androidx.cardview.widget.CardView>

    <androidx.cardview.widget.CardView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_row="1"
        android:layout_rowWeight="1"
        android:layout_column="1"
        android:layout_columnWeight="1"
        android:id="@+id/fifth"
        android:layout_gravity="fill"
        app:cardCornerRadius="12dp"
        app:cardElevation="14dp"
        app:cardUseCompatPadding="true">

        <LinearLayout
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_gravity="center_vertical|center_horizontal"
            android:gravity="center"
            android:orientation="vertical">

            <ImageView
                android:layout_width="50dp"
                android:layout_height="50dp"
                android:src="@drawable/self_defence" />

            <TextView
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:text="SELF DEFENSE"
                android:textAlignment="center"
                android:textStyle="bold" />
        </LinearLayout>
    </androidx.cardview.widget.CardView>
</GridLayout>

<com.google.android.material.button.MaterialButton
    android:id="@+id/panicBtn"
    android:layout_width="160dp"
    android:layout_height="60dp"
    android:layout_below="@id/gridlayout"
    android:layout_centerHorizontal="true"
    android:layout_marginStart="20dp"
    android:layout_marginTop="48dp"
```

```
        android:layout_marginEnd="20dp"
        android:layout_marginBottom="20dp"
        android:layout_weight="3"
        android:backgroundTint="#FF69B4"
        android:text="sos"
        android:textSize="26sp"
        app:cornerRadius="100dp" />


</RelativeLayout>
```

SmsActivity.xml

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".SmsActivity">

    <View
        android:layout_width="160dp"
        android:layout_height="148dp"
        android:background="@drawable/top_left_corner_oval" />

    <TextView
        android:layout_width="400dp"
        android:layout_height="90dp"
        android:layout_marginLeft="70dp"
        android:layout_marginTop="150dp"
        android:layout_marginRight="70dp"
        android:text="Send SOS in\nEmergency"
        android:textAlignment="center"
        android:textSize="35sp" />

    <LinearLayout
        android:id="@+id/lin"
        android:layout_width="300dp"
        android:layout_height="wrap_content"
        android:layout_marginTop="300dp"
        android:layout_marginLeft="110dp"
        android:layout_marginRight="110dp"
        android:gravity="center"
        android:orientation="vertical">

        <Button
```

```
        android:id="@+id/startService"
        android:layout_width="290dp"
        android:layout_height="60dp"
        android:textSize="7pt"
        android:shadowColor="@color/purple_500"
        android:text="Start SMS Alerts" />

    <Button
        android:id="@+id/stopService"
        android:layout_width="290dp"
        android:layout_height="60dp"
        android:textSize="7pt"
        android:shadowColor="@color/purple_500"
        android:text="Stop SMS Alerts" />

    <ToggleButton
        android:id="@+id/toggleAlert"
        android:layout_width="149dp"
        android:layout_height="wrap_content"
        android:layout_marginLeft="100dp"
        android:layout_marginTop="10dp"
        android:layout_marginRight="100dp"
        android:background="@drawable/toggle_button_states"
        android:textOff="Alert OFF"
        android:textOn="Alert ON" />

</LinearLayout>

<LinearLayout
    android:id="@+id/li"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:gravity="center"
    android:orientation="vertical">

    <TextView
        android:layout_width="316dp"
        android:layout_height="wrap_content"
        android:layout_marginLeft="100dp"
        android:layout_marginTop="500dp"
        android:layout_marginRight="100dp"
        android:text="Helpline Numbers"
        android:textAlignment="center"
        android:textSize="35sp"

android:textColorHighlight="@color/common_google_signin_btn_text_dark_pressed
"/>
```

```
        <Button
            android:id="@+id/btn_helpline"
            android:layout_width="149dp"
            android:layout_height="wrap_content"
            android:layout_marginLeft="100dp"
            android:layout_marginTop="10dp"
            android:layout_marginRight="100dp"
            android:text="Helplines" />
    </LinearLayout>

</RelativeLayout>
```

SplashActivity.xml

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".SplashActivity">


    <View
        android:id="@+id/topleftoval"
        android:layout_width="172dp"
        android:layout_height="159dp"
        android:layout_marginStart="-20dp"
        android:layout_marginTop="-30dp"
        android:background="@drawable/top_left_corner_oval" />

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="  S-HER "
        android:textSize="50sp"
        android:layout_centerHorizontal="true"
        android:textColor="#803A99"
        android:textStyle="bold"
        android:layout_above="@id/sparkImg"
        android:id="@+id/sparkText"/>

    <ImageView
        android:layout_width="700px"
        android:layout_height="700px"
        android:layout_centerVertical="true"
        android:paddingStart="30dp"
        android:src="@drawable/girl"
```

```
        android:id="@+id/sparkImg"
        android:layout_centerHorizontal="true"
        />

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:id="@+id/slang"
        android:textSize="28sp"
        android:layout_below="@id/sparkImg"
        android:text="Living Without Fear"
        android:textColor="#803A99"
        android:layout_centerHorizontal="true"/>

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_above="@id/sparkImg"
        android:layout_centerHorizontal="true"
        android:layout_marginStart="-316dp"
        android:layout_marginTop="8dp"
        android:layout_marginEnd="-316dp"
        android:layout_marginBottom="-405dp"
        android:text=" Safety. It's in your hands "
        android:textColor="@color/black"
        android:textSize="18sp"
        android:textStyle="bold" />

    <Button
        android:id="@+id/btnGetStarted"
        android:layout_width="200dp"
        android:layout_height="50dp"
        android:layout_below="@id/slang"
        android:layout_alignStart="@+id/sparkImg"
        android:layout_alignEnd="@+id/sparkImg"
        android:layout_centerHorizontal="true"
        android:layout_marginStart="38dp"
        android:layout_marginTop="12dp"
        android:layout_marginEnd="34dp"
        android:layout_marginBottom="15dp"
        android:text="START" />

</RelativeLayout>
```

4.3 Result

Fig 4.3.1: Start Page

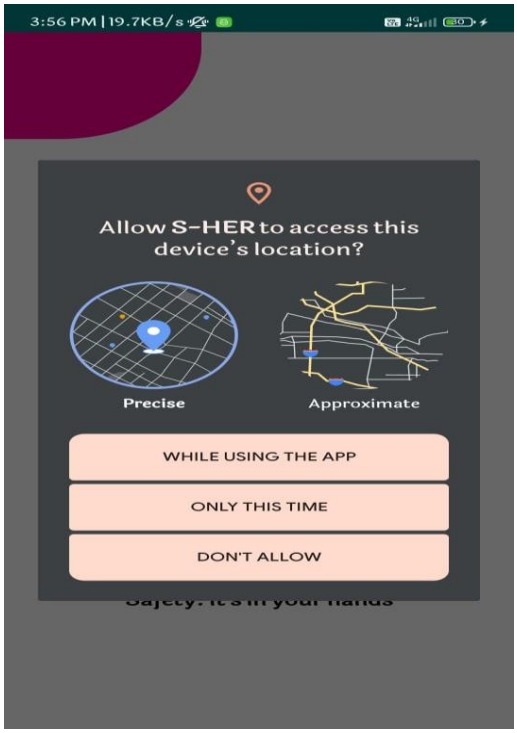

Fig 4.3.2: GPS Permissions

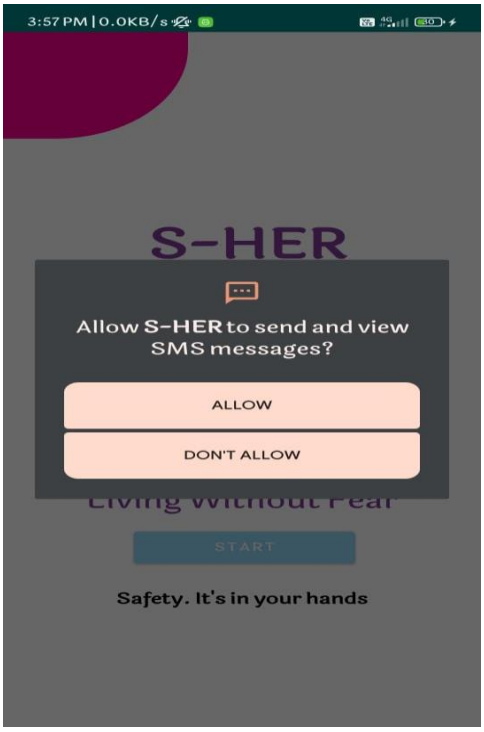

Fig 4.3.3: SMS Permissions

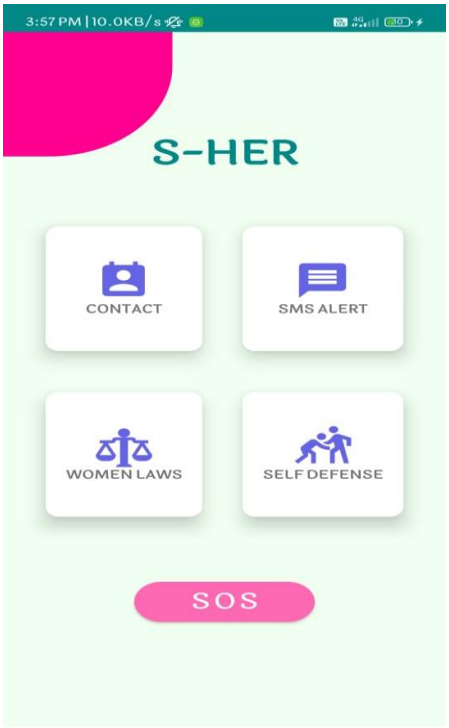

Fig 4.3.4: Home Screen

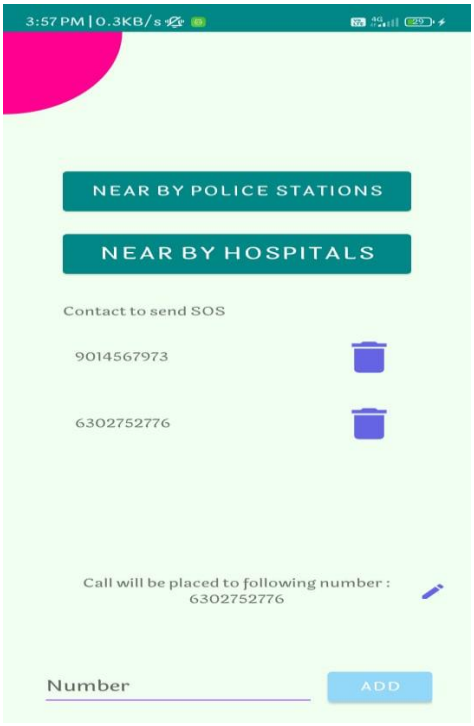

Fig 4.3.5: Contacts

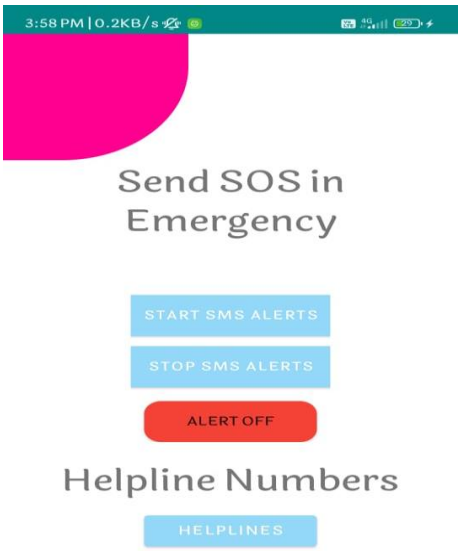

Fig 4.3.6: SMS Alerts

Fig 4.3.7: Helplines

Fig 4.3.8: Laws

Fig 4.3.9: Self Defense Video

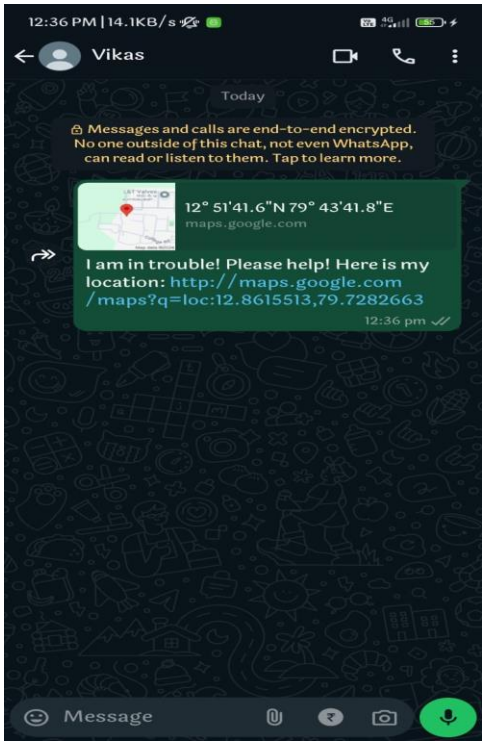

Fig 4.3.10: Alert in WhatsApp

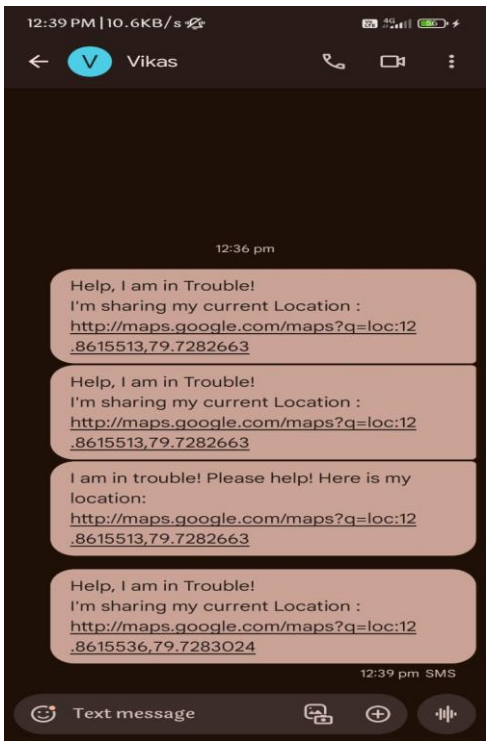

Fig 4.3.11: Alert in Text

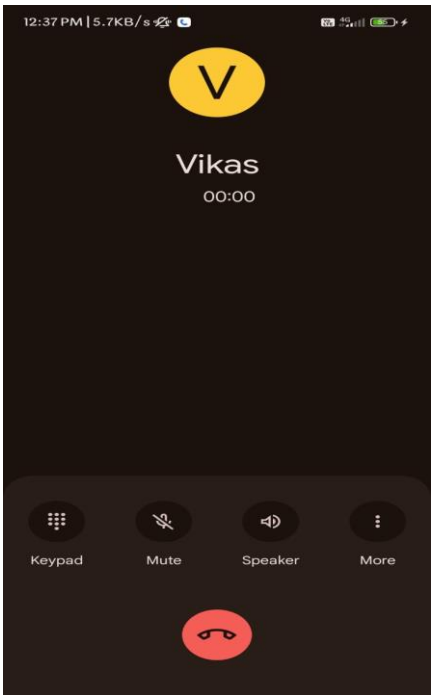

Fig 4.3.12: Calling Emergency Number

Chapter 5

Conclusion

5.1 Conclusion

The development of the S-HER application has addressed the critical need for women's safety in today's society. By leveraging advanced technologies such as Android Studio, Java, and XML, S-HER offers a comprehensive solution to empower women and enhance their personal security and safety. The application incorporates features like a shake detector, panic button, SOS alerts, and location tracking to ensure prompt assistance during emergencies. Additionally, the integration of nearby police stations and hospitals, women safety laws, self-defense videos, and national helpline numbers provides users with valuable resources and information. The user-friendly interface and seamless functionalities make S-HER accessible and easy to navigate, further enhancing its effectiveness. The successful implementation of the project showcases the potential of technology in promoting women's safety and creating a safer environment. S-HER serves as a significant step forward in safeguarding women and raising awareness about their rights and well-being.

5.2 References

[1]. Ravi Sekhar Yarrabothula Bramarambika Thota, “ABHAYA: AN ANDROID APP FOR THE SAFETY OF WOMEN,” IEEE, 1 December 2015.

[2]. Alisha Maruti Gawade, Amruta Jadhav and Sachin Shankar Kumbhar, “S-ZONE: A SYSTEM FOR WOMEN SAFETY & SECURITY SYSTEM,” Journal of Information, Knowledge and Research In Electronics And Communication Engineering ISSN: 0975 – 6779| Nov 16 to Oct 17 | Volume – 04, Issue – 02.

[3]. Sagar Khan, Harish Shinde, Ankita Zaroo, Rashmi Koushik , F. S. Ghodichor, “SHIELD: Personal Safety Application,” IRJET Volume: 04 Issue: 05 , May -2017.

[4]. Piyush Bhanushali, Rahul Mange, Dama Paras, Prof. Chitra Bhole, “Women Safety Android App,” IRJET Journal - Volume 5 Issue4, April 04 , 2018.

[5]. N. Ramesh Kannan , S. Sujitha, S. Ganapathy Subramanian, “Women Safety Mobile App,” International Journal on Cybernetics & Informatics (IJCI) Vol. 10, No.1/2, May 2021.

[6]. Sen, Trisha, Arpita Dutta, Shubham Singh, and Vaegae Nveen Kumar. "ProTecht–Implementation of an IoT based 3–Way Women Safety Device." In 2019 3rd International conference on Electronics, Communication and Aerospace Technology (ICECA), pp. 1377-1384. IEEE, 2019.

[7]. Tejonidhi, M. R., Chaithra KS Aishwarya, M. K. Dayana, and H. Nagamma. "IoT based smart security gadget for women’s safety." In 2019 1st international conference on advances in information technology (ICAIT). 2019

[8]. Kabir, AZM Tahmidul, and Tasnuva Tasneem. "Safety Solution for women using Smart band and CWS App." In 2020 17th International Conference on Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology (ECTI-CON), pp. 566-569. IEEE, 2020.

[9]. Khan, Rubaiat, Nagib Mahfuz, and Nadia Nowshin. "A Novel Approach of Women Safety Assistant Device with Biometric Verification in Real Scenario." In 2020 IEEE International Women in Engineering (WIE) Conference on Electrical and Computer Engineering (WIECON-ECE), pp. 426-431. IEEE, 2020.

[10]. Sunehra, Dhiraj, V. Sai Sreshta, V. Shashank, and B. Uday Kumar Goud. "Raspberry Pi Based Smart Wearable Device for Women Safety using GPS and GSM Technology." In 2020 IEEE International Conference for Innovation in Technology (INOCON), pp. 1-5. IEEE, 2020.

[11]. Hyndavi, V., N. Sai Nikhita, and S. Rakesh. "Smart wearable device for women safety using IoT." In 2020 5th International Conference on Communication and Electronics Systems (ICCES), pp. 459-463. IEEE, 2020.

[12]. Tejesh, B. S. S., Yarabarla Mohan, Ch Anil Kumar, T. Peter Paul, R. Sai Rishitha, and B. Purvaja Durga. "A Smart Women protection system using Internet of Things and Open-Source Technology." In 2020 International Conference on Emerging Trends in Information Technology and Engineering (ic-ETITE),pp. 1-4. IEEE, 2020.

[13].Tunggadewi, Elsyea, Eva Inaiyah, and Yunardi Riky Tri. "A smart wearable device based on internet of things for the safety of children in online transportation." Indonesian Journal of Electrical Engineering and Computer Science 9 (2021): 708.

[14].Raganna, A., K. Nithesh, B. Neha, Omchandra V. Shrivastav, and Praveen T. Musaguppi. "Iot Based Night Patrolling Robot for Women Safety." International Journal of Modern Agriculture 10, no. 2 (2021): 3886-3894.