INTRODUCTION

This document details the development of a mobile application that shares a user's real-time location via SMS in emergencies. The app uses Android's GPS and SMS functionalities to alert user-defined contacts. The project aims to ensure personal safetyand quick access to communication tools, particularly during emergencies. The app is designed to be user-friendly and work across various Android devices, ensuring broad usability and effectiveness.

LITERATURE SURVEY

EXISTING SYSTEM

Currently, various safety apps are available for women, but many lack comprehensive features. These apps may provide location tracking or emergency contacts but often miss advanced security measures like fingerprint authentication or customizable contact selection. Users must manually input their details every time they send an alert, and there's often no integration with local police stations or pre-filled emergency information

DISADVANTAGES OF EXISTING SYSTEMS

1. Limited Functionality:

- Many systems provide basic features like sending alerts but lack additional functionalities like real-time location tracking, integration with emergency services, or advanced user profiles.

2. Dependence on Network Connectivity:

- Many apps require an internet connection to function effectively, which can be problematic in emergencies or areas with poor connectivity.

3. Permission Issues:

- Users often face challenges with permission management, such as having to frequently grant access to contacts or SMS, which can be confusing and lead to app abandonment.

4. User Interface:

- Some systems may have complicated or unintuitive user interfaces, making it difficult for users, especially in stressful situations, to navigate quickly.

5. Privacy Concerns:

- Storing sensitive user data (like location and contact information) can raise privacy concerns, especially if the app does not have robust data protection measures.

6. Lack of Integration:

- Many existing systems do not integrate with local emergency services or law enforcement, which can delay response times during critical situations.

7. Customization Limitations:

- Users may have limited options for customizing alerts, messages, or contact lists, reducing the system's effectiveness for individual needs.

8. Accessibility Issues:

- Some systems may not be designed with accessibility in mind, making them difficult for users with disabilities to utilize effectively.

PROPOSED SYSTEM

The Secure Her application enhances women's safety by offering a streamlined, user- friendly experience. It will store essential personal details (name, age, blood group, etc.) once and allow users to select emergency contacts from their phonebook. When indanger, the app will send an alert SMS with the user's location to these contacts. Additionally, the app will feature fingerprint authentication for security, and users can easily update their information or contacts through the profile settings. Integration with the nearest police station will further ensure a timely response in emergencies

ADVANTAGES OF THE PROPOSED SYSTEM

1. Comprehensive Functionality:

- Offers a wide range of features, including SOS alerts, live location sharing, andbiometric authentication, providing users with multiple layers of safety.

2. User-Friendly Interface:

- Designed with an intuitive and accessible interface, enabling users to navigate theapp easily, especially in emergency situations.

3. Reliable Communication:

- Ensures reliable SMS delivery, even in low connectivity areas, enhancing the chancesof alerting contacts or authorities effectively.

4. Customizable Alerts:

- Allows users to customize emergency messages and select specific contacts, tailoringthe system to meet individual needs and preferences.

5. Integration with Emergency Services:

- Provides options to connect directly with local emergency services, enabling quickerresponse times and potentially saving lives.

6. Real-Time Location Tracking:

- Features live location sharing with selected contacts, ensuring that help can reach theuser quickly in emergencies.

SYSTEM REQUIREMENT SPECIFICATIONS

FUNCTIONAL REQUIREMENTS:

1. User Profile Management:
   - Input and save user details (name, age, blood group).
   - Load saved profile data on startup.
2. Emergency Contact Management:
   - Select and display emergency contacts from the phone.
   - Delete the last contact or any contact via a context menu.
3. SMS Sending:
   - Send an SOS message with user details to selected contacts.
4. Permissions:
   - Request and manage permissions for reading contacts and sending SMS.
5. Save and Load Data:
   - Store user profile and contacts in shared preferences.

NON-FUNCTIONAL REQUIREMENTS:

1. Performance:
   - Load profile and contacts within 2 seconds.
2. Security:
   - Handle permissions securely and store data locally.
3. Usability:
   - Provide a simple UI with clear actions and feedback.
4. Error Handling:
   - Gracefully manage permission denials, and no contacts are selected.
5. Compatibility:
   - Support Android API 21 and above, adaptable to various screen sizes.
6. Battery Efficiency:
   - Minimize background processes to conserve battery life.
7. Maintainability:
   - Structure code for easy future enhancements.
8. Reliability:
   - Ensure SMS delivery, even during network issues.

TECHNOLOGY REQUIREMENTS:

SOFTWARE REQUIREMENTS:

- IDE: Android Studio
- Java Development Kit (JDK): Required for Java
- Operating System: Windows 10 or 11

HARDWARE REQUIREMENTS:

- Processor: Intel i5 or above
- RAM: 16 GB minimum
- Storage: 10 GB free (SSD preferred)
- Graphics: Intel XE and Higher

USER HARDWARE AND SOFTWARE REQUIREMENTS OF APPLICATION

SOFTWARE REQUIREMENTS:

- Operating System: Android OS version 7.0 (Nougat)
- Cellular Connectivity
- GPS(Global Positioning System)

HARDWARE REQUIREMENTS:

- Minimum RAM: 2 GB of RAM or higher.
- Storage: At least 50 MB of available storage.

SOFTWARE DEVELOPMENT PROCESS MODEL

ARCHITECTURAL DESIGN PLAN

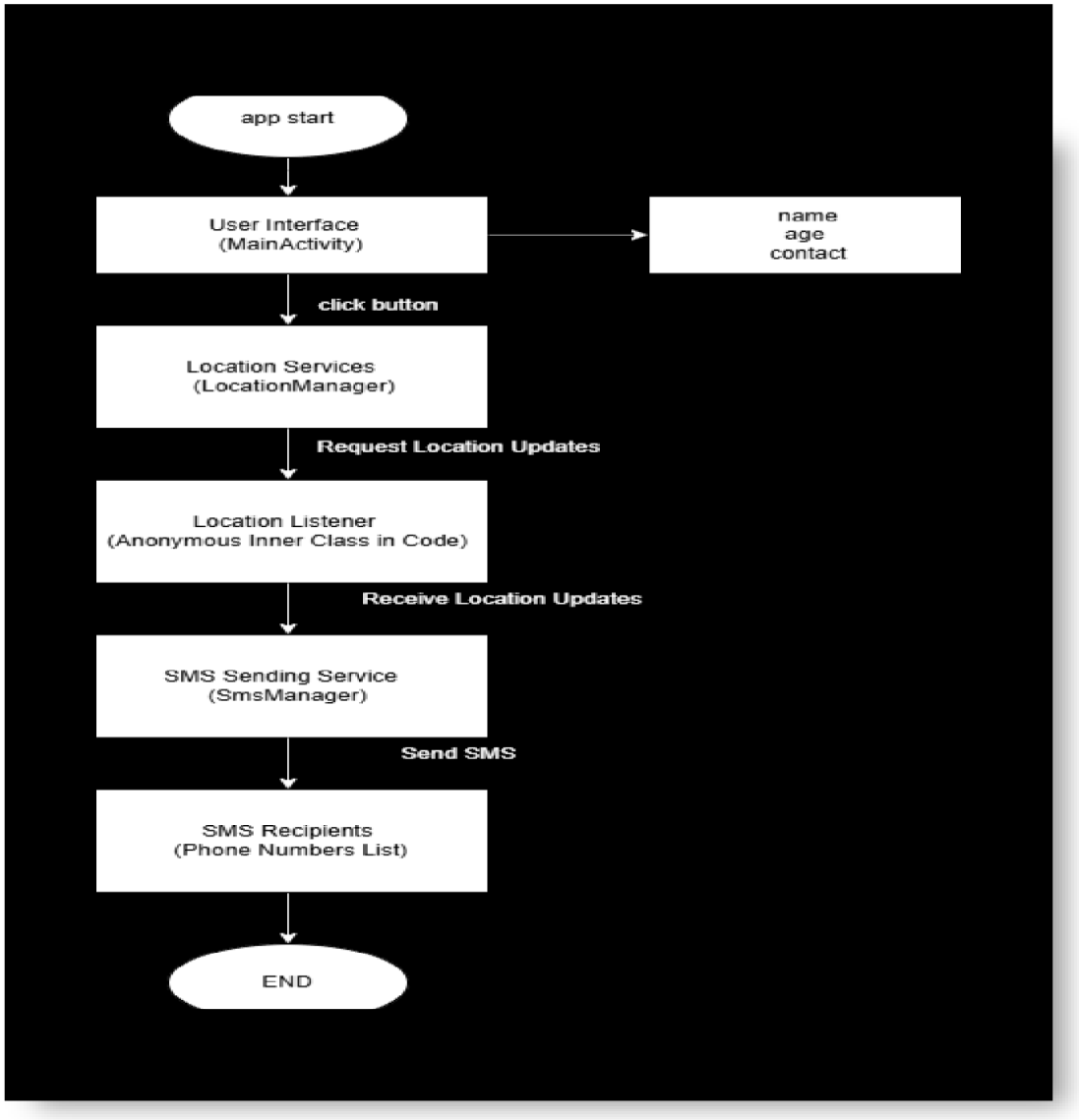


Figure 4.1

DATA FLOW DIAGRAM

LEVEL 0

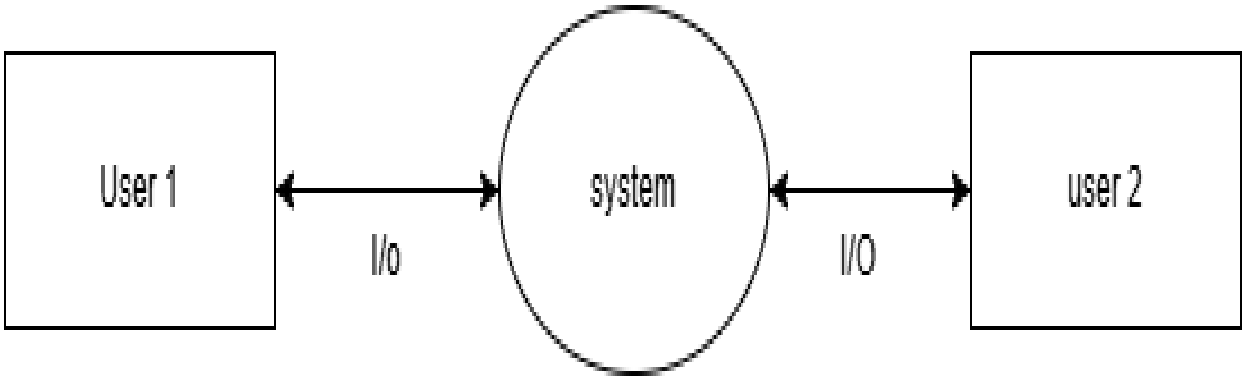


Figure 4.2.1

LEVEL 1

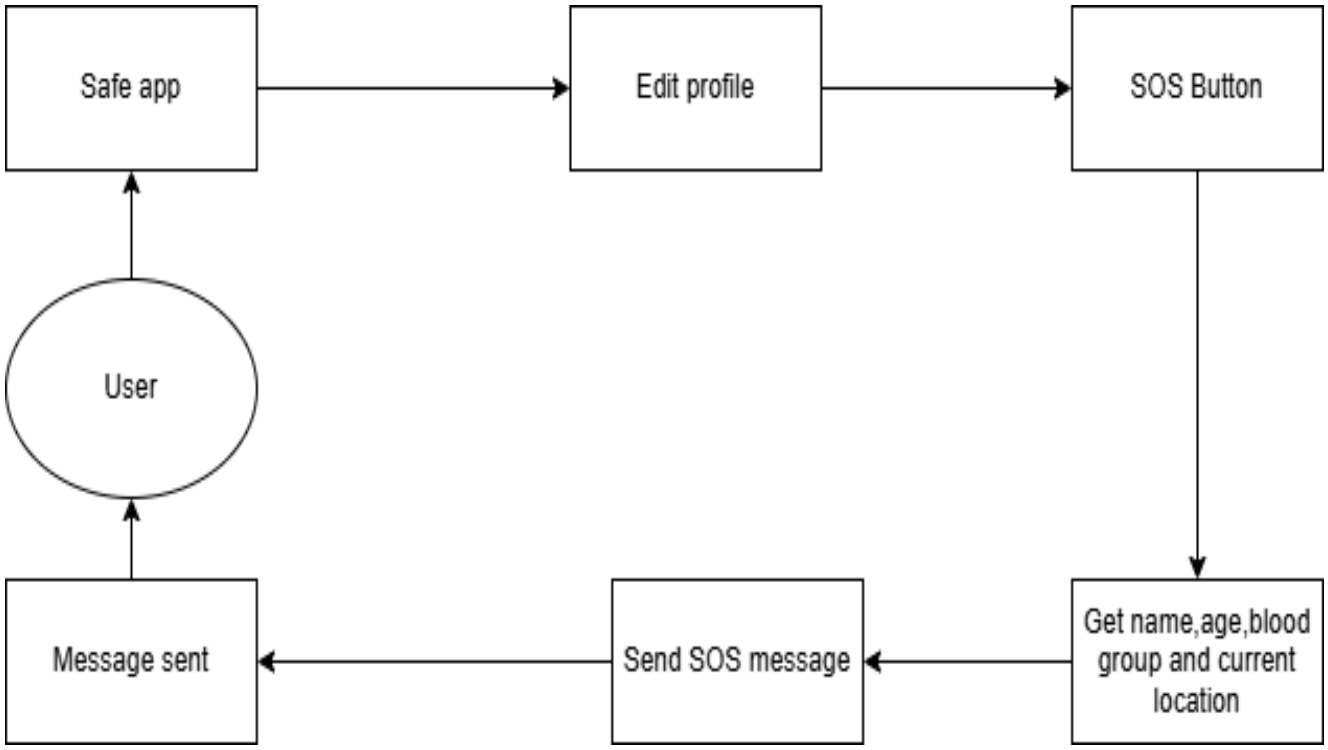
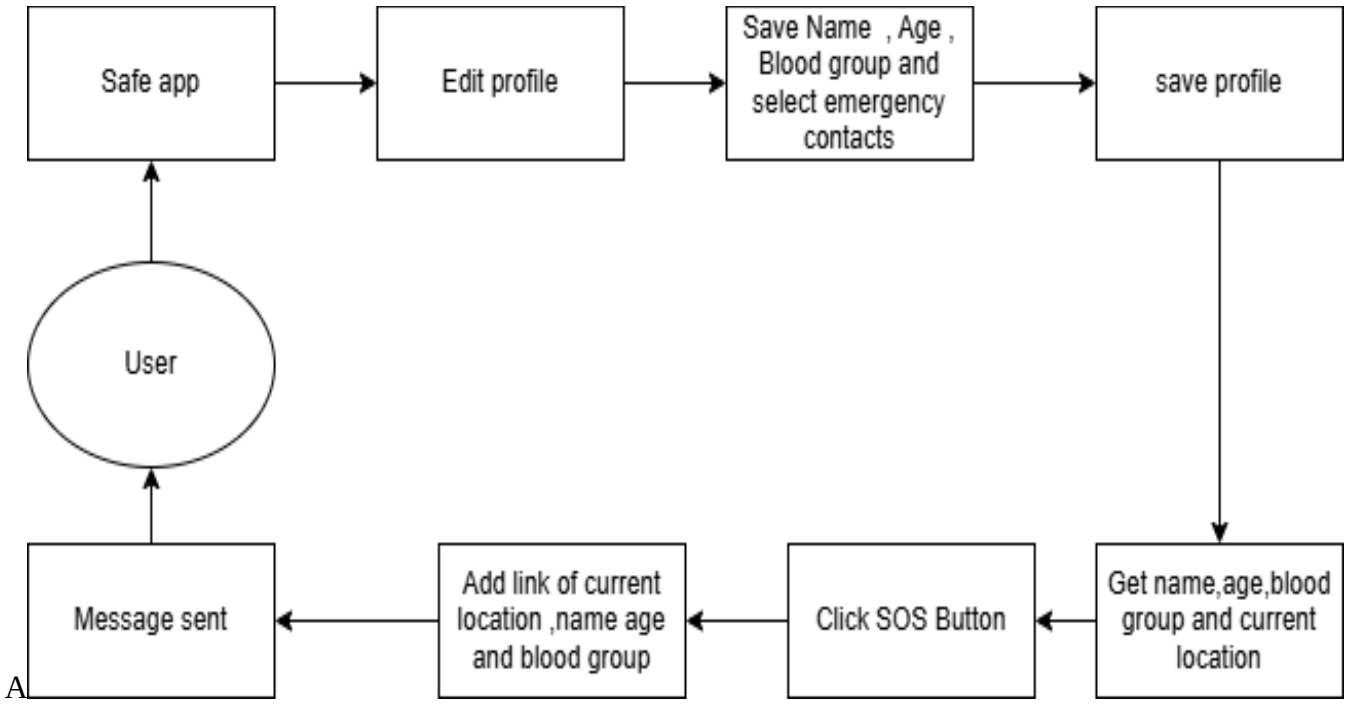


Figure 4.1.2

LEVEL 2



A

Figure 4.1.3

ACTIVITY DIAGRAM:

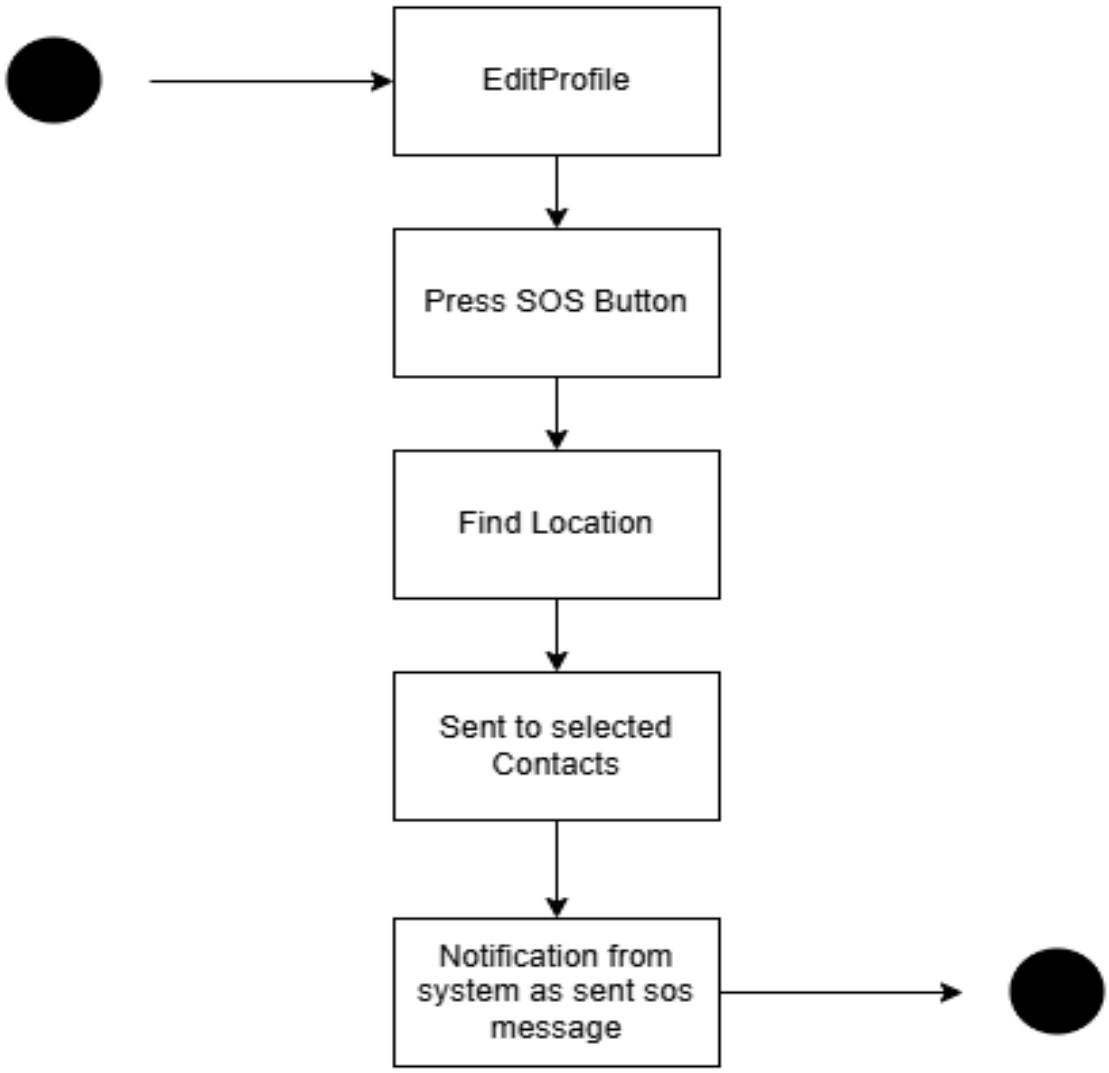


Figure 4.3

USE CASE DIGRAM

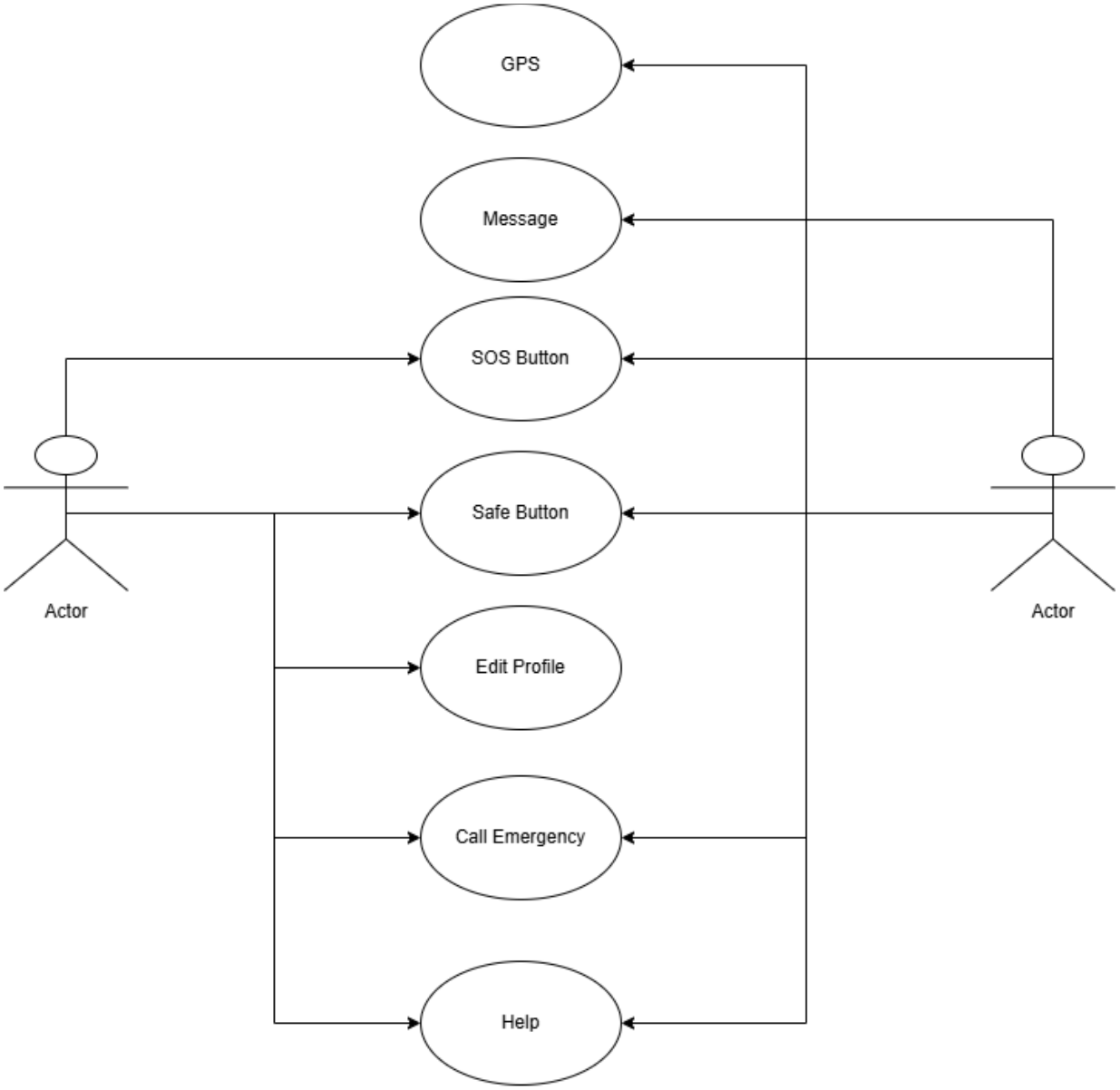


Figure 4.4

SEQUENCE DIAGRAM

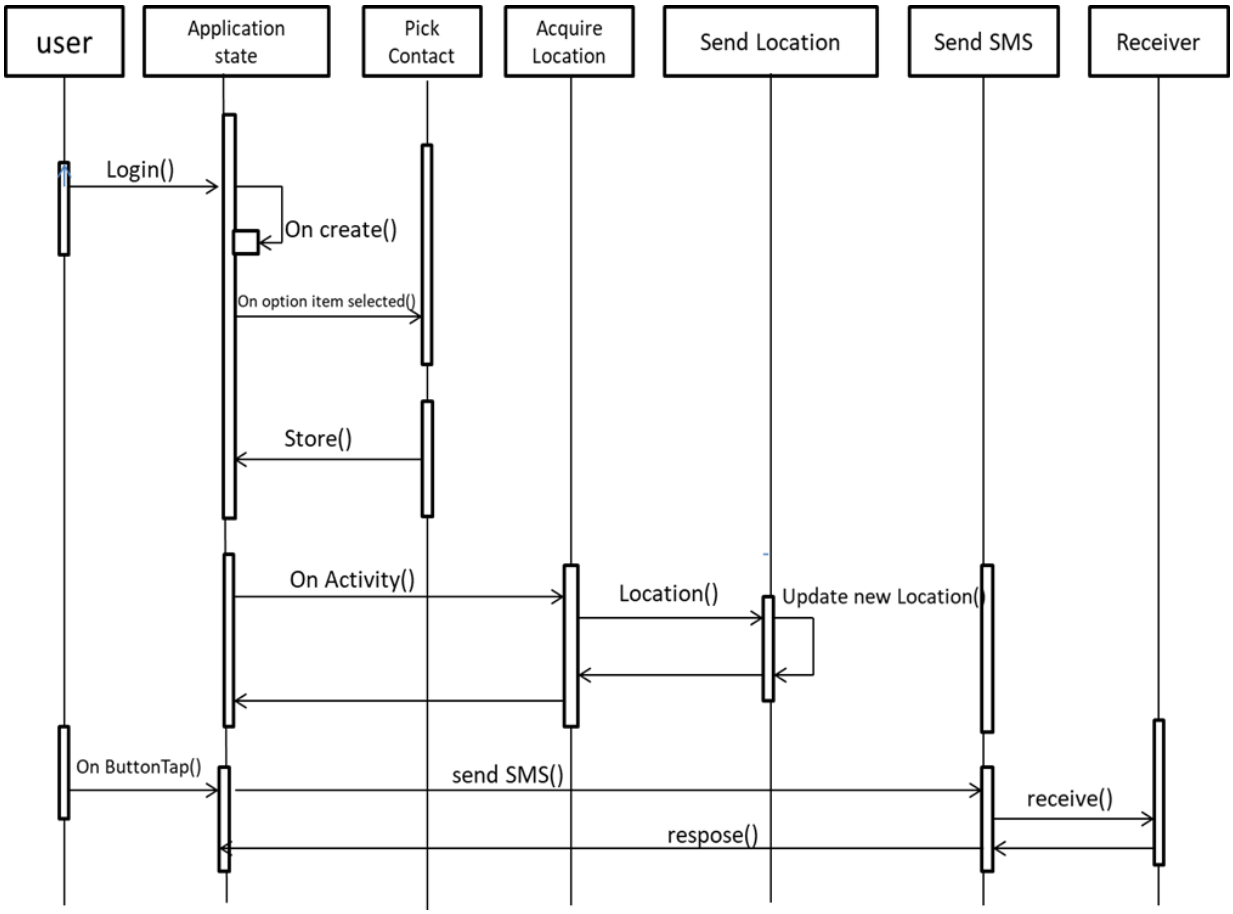


Figure 4.5

SOURCE CODE

XML CODE

```
<ScrollView xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:padding="16dp"
    android:background="#040101">
    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:orientation="vertical"
        android:padding="16dp">
        <!-- About Button (Question Mark) -->
        <!-- Greeting TextView -->
        <ImageButton
            android:id="@+id/buttonAbout"
            android:textColor="#B31111"
            android:layout_width="50dp"
            android:layout_height="50dp"
            android:layout_gravity="end"
            android:layout_marginTop="8dp"
            android:layout_marginEnd="8dp"
            android:background="#BAE2E1"
        android:contentDescription="About"
        android:src="@android:drawable/ic_menu_help" />
        <!-- User Name TextView -->
        <!-- Edit Profile Button -->
        <TextView
```

```
        android:id="@+id/greetingTextView"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
    android:layout_marginTop="2dp"
        android:text="User"
        android:textColor="#FFFFFF"
        android:textSize="40sp"
        android:textStyle="italic" />
      <Button
        android:id="@+id/buttonProfile"
        android:layout_width="200dp"
        android:layout_height="50dp"
        android:backgroundTint="#FF8000"
        android:text="Edit Profile"
        android:textColor="#FFFFFF"
        android:textSize="18sp"
        android:textStyle="bold"
        android:layout_gravity="center_horizontal"
        android:layout_marginTop="32dp"
      <!-- Call Emergency Button -->
      <com.google.android.material.button.MaterialButton
        android:layout_width="wrap_content"
        android:layout_height="70dp"
        android:backgroundTint="#CC2B52"
        android:text="Call Emergency (100)"
        android:textColor="#FFFFFF"
        android:layout_gravity="center_horizontal
   </LinearLayout>
</ScrollVi>
```

JAVA CODE:

```
package com.example.womensafety;
import android.Manifest;
import android.app.NotificationChannel;
import android.app.NotificationManager;
import android.content.Intent;
import android.content.IntentSender;.TextView;
import android.widget.Toast;
import androidx.annotation.NonNull;
import androidx.appcompat.app.AppCompatActivity;;
import com.google.android.gms.location.SettingsClient;
import com.google.gson.Gson;
import com.google.gson.reflect.TypeToken;
import java.io.IOException;
import java.lang.reflect.Type;
import java.util.List;
import java.util.Locale;
public class MainActivity extends AppCompatActivity {
    private Button buttonSOS, buttonProfile, buttonCall;
    private ImageButton buttonAbout;
    private FusedLocationProviderClient fusedLocationClient;
    private TextView greetingTextView, locationTextView;
    private SoundPool soundPool;
    private int soundID;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);

    private void callEmergency() {
```

```
        String emergencyNumber = "7411411500";
        Intent callIntent = new Intent(Intent.ACTION_DIAL, Uri.parse("tel:" + emergencyNumber));
        startActivity(callIntent);
    }
NotificationManager.IMPORTANCE_HIGH);
            channel.setDescription("This is the default notification channel.");
            the documentation
            // for ActivityCompat#requestPermissions for more details.
            return;
        }
        notificationManagerCompat.notify(0, builder.build());
    }
    private void callEmergency() {
        String emergencyNumber = "7411411500";
        Intent callIntent = new Intent(Intent.ACTION_DIAL, Uri.parse("tel:" + emergencyNumber));
        startActivity(callIntent);
    }
    @Override
    protected void onActivityResult(int requestCode, int resultCode, Intent data) {
        super.onActivityResult(requestCode, resultCode, data);
        if (requestCode == 1) {
            if (resultCode == RESULT_OK) {
                // Location is now enabled, proceed to fetch location
                getCurrentLocation();
            } else {
                Toast.makeText(this, "Location is required for SOS functionality",
Toast.LENGTH_SHORT).show();
            }
```

SCREENSHOT

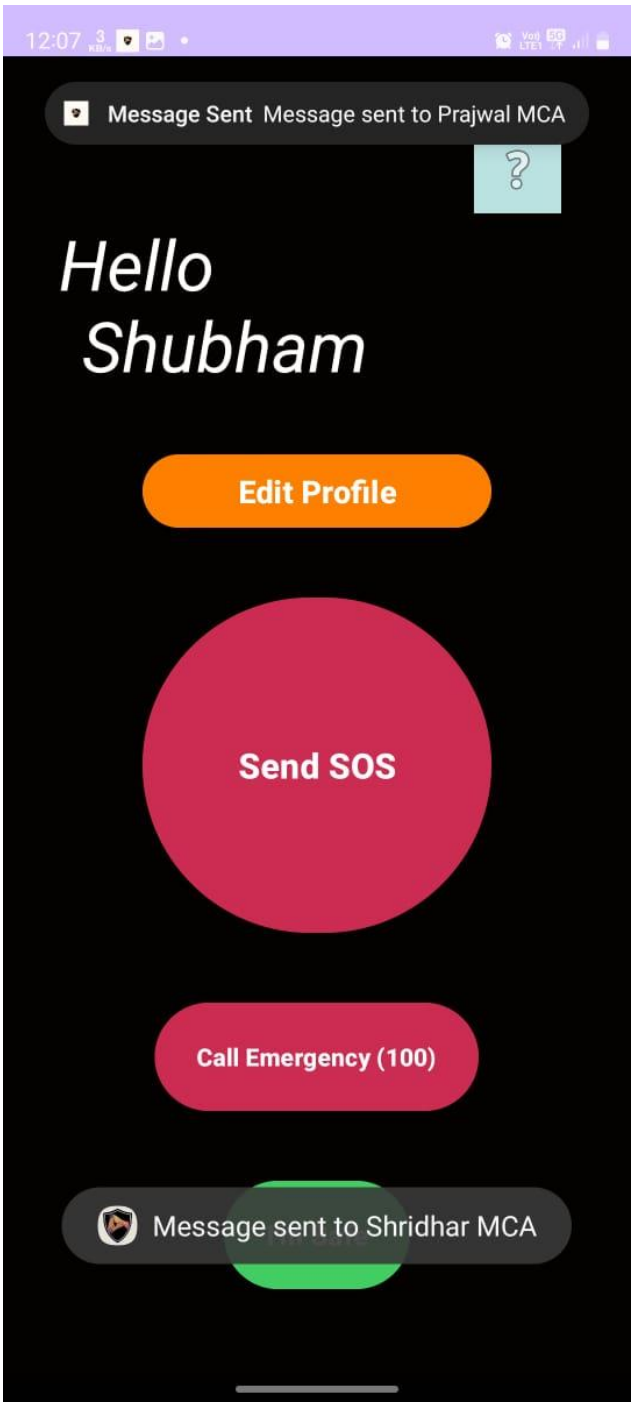


Figure 5.1

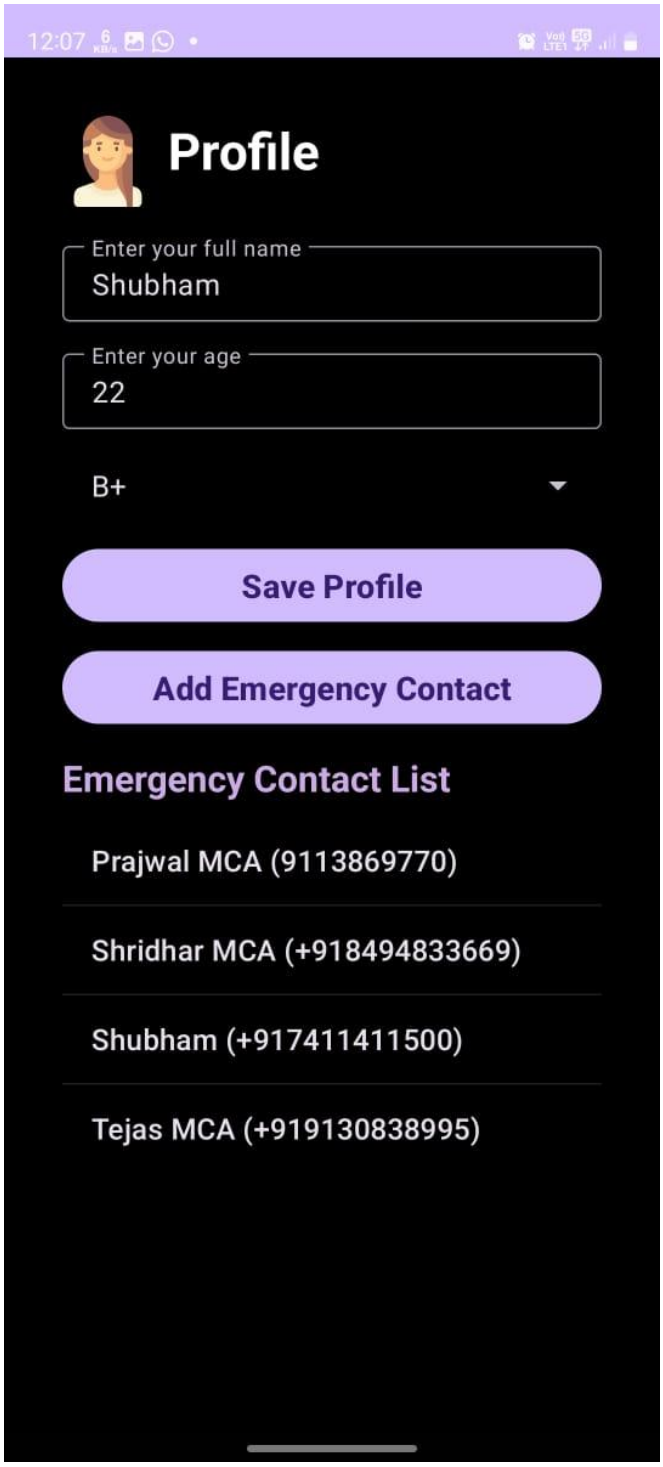

12:07
Profile
Enter your full name
Shubham
Enter your age
22
B+
Save Profile
Add Emergency Contact
Emergency Contact List
Prajwal MCA (9113869770)
Shridhar MCA (+918494833669)
Shubham (+917411411500)
Tejas MCA (+919130838995)

Figure 5.2

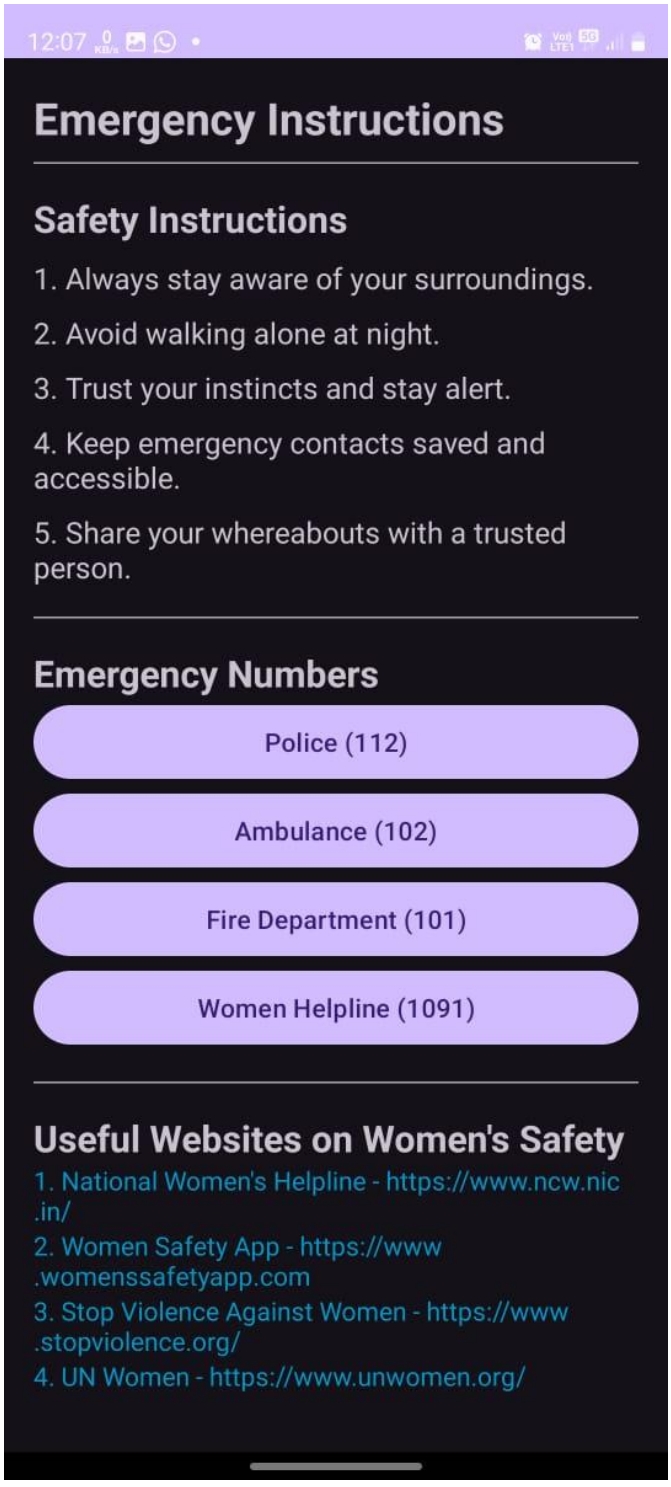

12:07
Emergency Instructions
Safety Instructions
1. Always stay aware of your surroundings.
2. Avoid walking alone at night.
3. Trust your instincts and stay alert.
4. Keep emergency contacts saved and accessible.
5. Share your whereabouts with a trusted person.
Emergency Numbers
Police (112)
Ambulance (102)
Fire Department (101)
Women Helpline (1091)
Useful Websites on Women's Safety
1. National Women's Helpline - https://www.ncw.nic.in/
2. Women Safety App - https://www.womenssafetyapp.com
3. Stop Violence Against Women - https://www.stopviolence.org/
4. UN Women - https://www.unwomen.org/

Figure 5.3

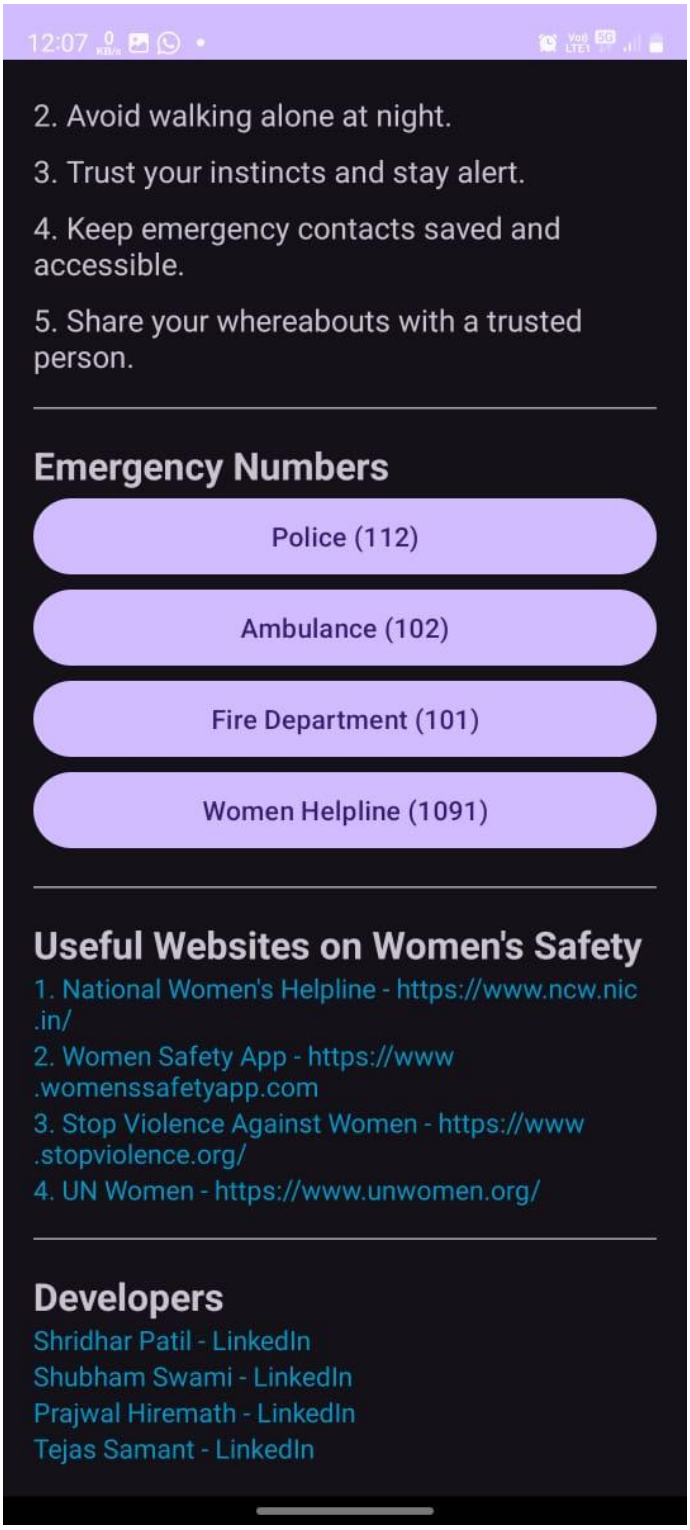

12:07
2. Avoid walking alone at night.
3. Trust your instincts and stay alert.
4. Keep emergency contacts saved and accessible.
5. Share your whereabouts with a trusted person.
Emergency Numbers
Police (112)
Ambulance (102)
Fire Department (101)
Women Helpline (1091)
Useful Websites on Women's Safety
1. National Women's Helpline - https://www.ncw.nic.in/
2. Women Safety App - https://www.womenssafetyapp.com
3. Stop Violence Against Women - https://www.stopviolence.org/
4. UN Women - https://www.unwomen.org/
Developers
Shridhar Patil - LinkedIn
Shubham Swami - LinkedIn
Prajwal Hiremath - LinkedIn
Tejas Samant - LinkedIn

Figure 5.4

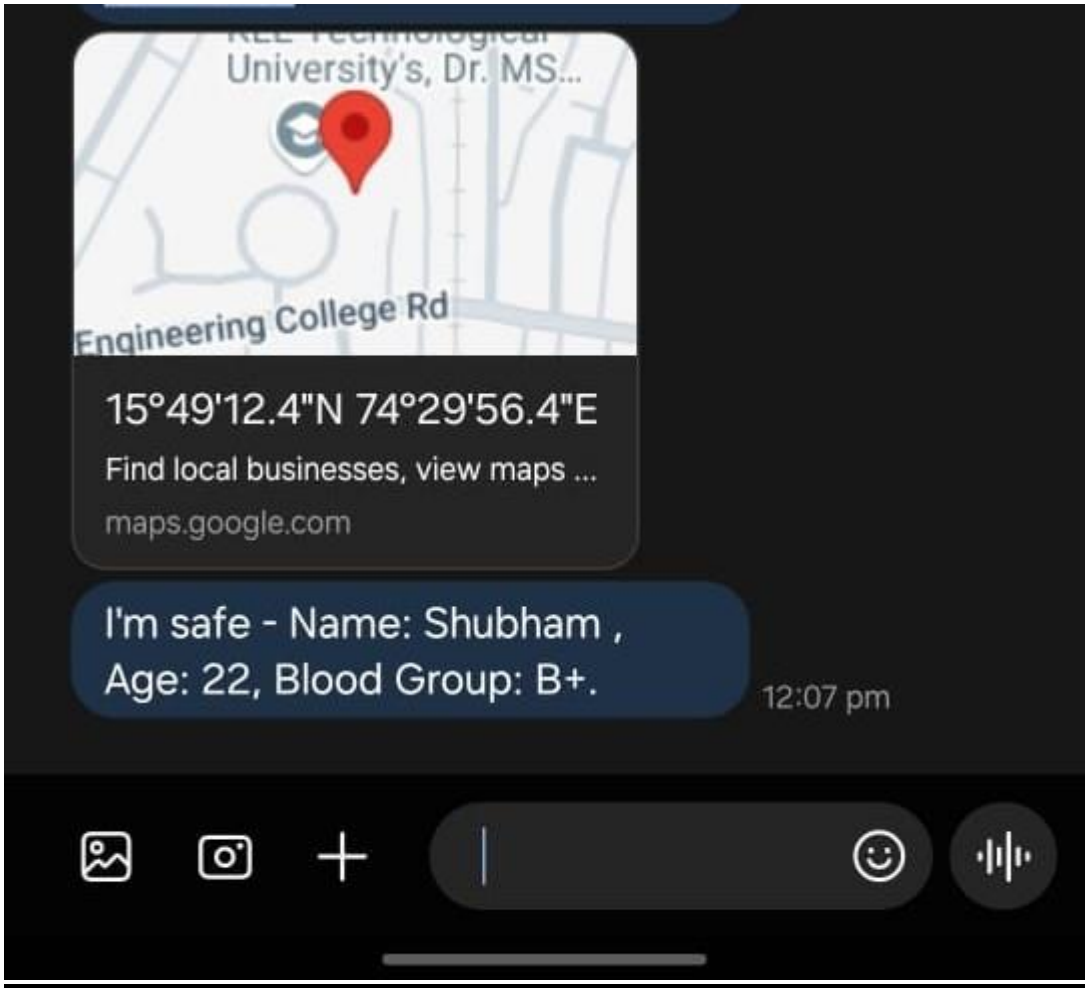

University's, Dr. MS...
Engineering College Rd
15°49'12.4"N 74°29'56.4"E
Find local businesses, view maps ...
maps.google.com
I'm safe - Name: Shubham ,
Age: 22, Blood Group: B+.
12:07 pm

Figure 5.4

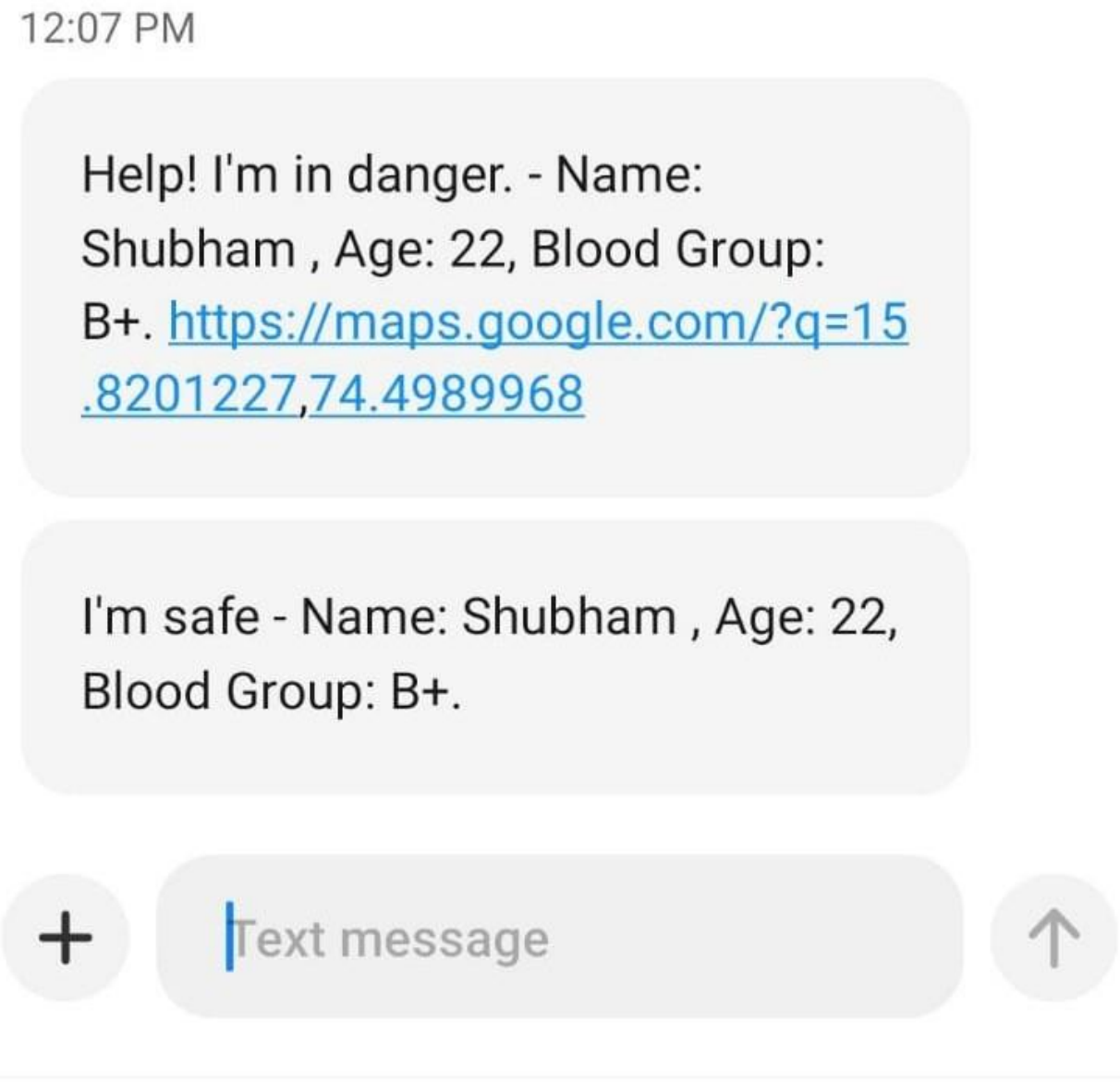

12:07 PM
Help! I'm in danger. - Name: Shubham , Age: 22, Blood Group: B+. https://maps.google.com/?q=15.8201227,74.4989968
I'm safe - Name: Shubham , Age: 22, Blood Group: B+.
Text message

Figure 5.5

TESTING

This testing is black-box testing because it focuses on verifying the app's external functionalities without examining its internal code or implementation. Testers interact with the app's user interface to validate outputs based on given inputs and scenarios. The test cases check how the app behaves in different situations, such as sending SOS messages, saving profile data, and handling permissions. It also tests edge cases like lack of network or location services. Internal workings of the app, such as algorithms or code structure, are not analyzed. The emphasis is solely on whether the app meets its functional requirements.

TEST CASES

Test Case ID	Test Case Description	Preconditions	Test Steps	Expected Result	Status
TC001	Verify SOS button sends the correct information	The app is installed, and user profile information is filled	1. Open the app. 2. Tap the SOS button.	An SOS message is sent with the user's name, age, blood group, and link to the current location.	Pass
TC002	Verify "I'm Safe" button sends the appropriate SMS	The app is installed, contacts are saved	1. Open the app. 2. Tap the "I'm Safe" button.	SMS is sent with the message "I'm safe" along with the current time.	Pass
TC003	Verify Profile Save button saves user info	App is installed	1. Open the Profile section. 2. Enter details (name, age, blood group, etc.). 3. Tap the Save button.	User information is successfully saved and displayed on the Profile screen.	Pass
TC004	Verify Add Contact button adds a new contact	The app is installed, permission to access contacts	1. Open the Profile section. 2. Tap the Add Contact button. 3.	The selected contact is added to the list of emergency contacts.	Pass

Test Case ID	Test Case Description	Preconditions	Test Steps	Expected Result	Status
TC005	Verify deletion of contacts	The app is installed, contacts are saved	1. Open the Profile section. 2. Select a contact. 3. Tap Delete.	Contact is successfully removed from the list of emergency contacts.	Pass
TC006	Verify location link updates dynamically	App has internet and location access	1. Open the app. 2. Tap the SOS button multiple times in different locations.	Each SOS message includes a unique link that corresponds to the current location at the time of sending.	Pass
TC007	Verify data persistence after app restart	App is installed, user profile and contacts saved	1. Enter user info and add contacts. 2. Restart the app. 3. Check Profile and contacts.	All user data and contacts remain saved after restarting the app.	Pass
TC008	Verify biometric authentication for app access	Device supports biometrics, app configured for it	1. Open the app. 2. Authenticate using fingerprint/face recognition.	User is granted access to the app after successful biometric authentication.	Pass
TC009	Verify SMS permission request	App installed on new device	1. Open the app on a new device. 2. Attempt to send an SOS message.	App prompts the user for SMS permissions before sending an SOS message.	Pass

Test Case ID	Test Case Description	Preconditions	Test Steps	Expected Result	Status
TC010	Verify handling of denied location access	User denies location permission on prompt	1. Open the app for the first time. 2. Deny location permission. 3. Tap the SOS button.	App sends SOS message without location and shows alert that location access is needed for full functionality.	Pass
TC011	Verify SOS message fails to send without network coverage	App is installed, no network coverage	1. Disable network (e.g., switch to airplane mode). 2. Open the app. 3. Tap the SOS button.	SOS message fails to send, and an error message appears indicating no network is available.	Pass
TC012	Verify app handles location services failure gracefully	Location services disabled	1. Turn off location services on the device. 2. Open the app. 3. Tap the SOS button.	SOS message sends without the location link, and an alert message appears stating that location services are disabled.	Pass

CONCLUSION

Your women's safety app effectively combines essential features like SOS alerts with location sharing, emergency contact management, and biometric security, making it a practical and secure tool for personal safety. To enhance it further, consider adding real-time location updates, cloud backup, push notifications, and multilingual support, as well as improving the UI layout and data encryption. These improvements would make the app even more user-friendly, accessible, and robust for safeguarding its users.

FURTHER ENHANCEMENTS

1. Location Accuracy and Updates:
    - You could improve the live location feature by implementing continuous location updates, ensuring that the SOS feature always sends the most current location.
    - Consider adding geofencing to detect when a user enters or exits a specific safe zone, triggering alerts or location updates automatically.
2. User Interface Improvements:
    - Since you prefer not to use colors in your UI, consider enhancing the layout with more structured and intuitive components like icons, spacing, and font changes to improve the user experience without relying on colors.
    - Add an onboarding screen to guide new users through setting up their profile and contacts.
3. Enhanced Security:
    - Implement a two-factor authentication process in addition to biometric authentication, like a PIN or OTP, for added security.
    - Ensure the sensitive data like emergency contacts are stored encrypted for better privacy protection.
4. Cloud Backup and Synchronization:
    - Allow users to back up their data to the cloud so that they can restore it on a new device. This would make the app more robust and reliable in case the user loses their phone or needs to transfer data.
5. Push Notifications:
    - Implement push notifications to keep users informed, such as reminders to update their safety profile or alerts when an emergency contact responds to an SOS alert.

6. Multilingual Support:
    - Add multilingual support to make the app accessible to users in different regions, enhancing its reach and usability.
7. Testing and User Feedback:
    - Conduct extensive testing with real-world users to identify pain points and further refine the app's functionality, performance, and user experience.

BIBLIOGRAPHY

- WOMEN'S SECURITY", Android App developed by App Soft India,December 17, 2013. https://play. Google. com/store
- Head First Android Development: A Brain-Friendly Guide by DawnGriffiths and David Griffiths
- Android Programming: The Big Nerd Ranch Guide by Bill Phillips, ChrisStewart, and Kristin Marsicano
- Burnette, E. (2008). Hello, Android: Introducing Google's Mobile Development Platform. Hardy, B., & Phillips, B. (2017).
- Android Programming: The Big Nerd Ranch Guide. Smith, D. (2011).
- Meike, G. B., Dornin, L., Nakamura, M., &Mednieks, Z. R. (2011) Android Programming