

WOMEN SAFETY APP FOR IMPROVED PERSONAL SECURITY

A PROJECT REPORT

Submitted by

NAVEEN KUMAR S [711719104055]

NEWTTON HABAKUK S [711719104057]

NIVETHITHA S [711719104061]

POOJA B [711719104063]

in partial fulfillment for the award of the degree

of

BACHELOR OF ENGINEERING

IN

COMPUTER SCIENCE AND ENGINEERING

KGiSL INSTITUTE OF TECHNOLOGY, SARAVANAMPATTI

ANNA UNIVERSITY : CHENNAI 600 025

MAY 2023

ANNA UNIVERSITY : CHENNAI 600 025

BONAFIDE CERTIFICATE

Certified that this project report **“WOMEN SAFETY APP FOR IMPROVED PERSONAL SECURITY”** is the bonafide work of **“NAVEEN KUMAR S, NEWTTON HABAKUK S, NIVETHITHA S, POOJA B”** who carried out the project work under my supervision.

SIGNATURE

Dr. THENMOZHI T

Head Of The Department,

Professor

Computer Science and Engineering,

KGiSL Institute of Technology,

Saravanampatti,

Coimbatore-35

SIGNATURE

Ms. ARUNA T N

Supervisor,

Assistant Professor

Computer Science and Engineering,

KGiSL Institute of Technology,

Saravanampatti,

Coimbatore-35

Submitted for the Anna University Viva-Voce examination held on _____

Internal Examiner

External Examiner

ACKNOWLEDGEMENT

We express our deepest gratitude to our **Chairman and Managing Director Dr. Ashok Bakthavachalam** for providing us with an environment to complete our project successfully.

We are grateful to our CEO, Academic Initiatives **Mr. Aravind Kumar Rajendran**, our beloved Director-Academics **Dr. P. Shankar** and our honorable Secretary **Dr. N. Rajkumar**. We are thankful to our beloved Principal **Dr. M. Selvam** and Vice-Principal **Dr. S. Suresh Kumar** for their support and valuable guidance and blessings.

We would like to thank **Dr. T. Thenmozhi, M.E., M.B.A., Ph. D**, Head of the Department, Department of Computer Science and Engineering for her unwavering support during the entire course of this project work. We express our sincere thanks to **Ms. ARUNA T N, M.E.**, our Project Coordinator, Assistant Professor, Department of Computer Science and Engineering who modelled us both technically and morally for achieving greater success in completing this project work.

We express our sincere and heartfelt thanks to our faculty guide **Ms. ARUNA T N M.E.**, Assistant Professor, Department of Computer Science and Engineering for her constant encouragement and support throughout our course, especially for the useful suggestions given during the course of the project period and being instrumental in the completion of our project with her complete guidance.

We also thank all the **Faculty members** of our department and finally, we take this opportunity to extend our deep appreciation to our **Family and Friends**, for all they meant to us during the crucial times of the completion of our project.

ABSTRACT

The security of women is a critical issue faced by society. Crimes against women such as eve teasing, sexual assaults, domestic violence are increasing in number day by day. When it comes to security concerns, a smart phone can be one the easiest way of gaining help. This project strives to create an android app which can help to protect women in any situation she might face in her day-to-day life. We have created a simple Android application which comprises various safety measures which can be used by women with a few clicks on the screen, to get quick and easy access to help or to avoid and escape a harmful situation. It uses GPS location tracking to provide a simple and fast way for the registered contacts to know that the user is in trouble and for them to reach the user easily. It also provides safety features such as a voice recording which can help a woman or the police for identification or situational evidence, a siren to alert the public of any misbehavior, spy camera detection for detecting hidden cameras, emergency helpline numbers which can be used to directly connect via call to emergency services according to the situation faced by women for their safety.

TABLE OF CONTENTS

CHAPTER NO.	TITLE	PAGENO.
	ABSTRACT	I
	LIST OF FIGURES	i
	LIST OF ABBREVIATIONS	ii
1.	INTRODUCTION	1
	1.1 PROBLEM DEFINITION	1
	1.2 OBJECTIVE OF THE PROJECT	2
	1.3 SIGNIFICANCE OF THE PROJECT	2
	1.4 OUTLINE OF THE PROJECT	3
2.	LITERATURE REVIEW	4
3.	SYSTEM ANALYSIS	5
	3.1 EXISTING SYSTEM	5
	3.2 DRAWBACKS	5
	3.3 PROPOSED SYSTEM	5
	3.4 FEASIBILITY STUDY	6
	3.4.1 Tests of Feasibility	6
	3.4.1.1 Technical Feasibility	7
	3.4.1.2 Operational Feasibility	8
	3.4.1.3 Economical Feasibility	8
4.	SYSTEM SPECIFICATION	9
	4.1 FUNCTIONAL REQUIREMENTS	9
	4.1.1 SOS Alert	9
	4.1.2 Emergency service	9

4.1.3	Detection of hidden camera	9
4.1.4	Login	9
4.1.5	Register	9
4.1.6	Women's NEWS	10
4.1.7	GPS Tracking	10
4.1.8	Panic button	10
4.2	NON-FUNCTIONAL REQUIREMENTS	10
4.2.1	Performance	10
4.2.2	Compatability	10
4.2.3	Security	11
4.2.4	Maintainability	11
4.2.5	Exception handling	11
4.2.6	Ethics	11
4.3	HARDWARE REQUIREMENTS	11
4.4	SOFTWARE REQUIREMENTS	12
5.	SOFTWARE DESCRIPTION	13
5.1	FRONT END	13
5.1.1	XML	13
5.1.1.1	Features	13
5.1.1.2	Advantages	14
5.2	BACK END	15
5.2.1	Firebase	15
5.2.2	Features of Firebase	15
5.2.3	Advantages of Firebase	16
5.2.4	JAVA	16
6.	PROJECT DESCRIPTION	19
6.1	OVERVIEW OF THE PROJECT	19
6.2	MODULE DESCRIPTION	20
6.2.1	UI Design And User Authentication Check	20
6.2.2	Emergency Alert Using Button Click	20

6.2.3 Siren Alert For Distraction	20
6.2.4 SPY Camera Detection	20
6.2.5 Firebase Storage	20
6.3 DATA FLOW DIAGRAM	20
6.3.1 DFD Level 0	21
6.3.2 DFD Level 1	22
6.3.3 DFD Level 2	22
6.4 ARCHITECTURE DIAGRAM	23
6.4.1 Women Safety App	24
6.5 DATABASE DESIGN	25
6.6 INPUT DESIGN	26
6.7 OUTPUT DESIGN	27
7. SYSTEM TESTING	28
7.1 TESTING METHODS	28
7.2 TYPES OF TESTING	28
7.2.1 Unit Testing	28
7.2.2 Integration Testing	28
7.2.3 Functional Testing	29
7.2.4 Stress Testing	29
7.2.5 Acceptance Testing	30
7.2.6 White Box Testing	30
7.2.7 BlackBox Testing	31
7.2.7.1 Methods of Black Box Testing	31
7.3 TESTING STRATEGY	31
8. SYSTEM IMPLEMENTATION	32
8.1 ACCOUNT CREATION	32
8.2 DATABASE INTEGRATION	32
8.3 USER AUTHENTICATION CHECK	32
8.4 SOS EMERGENCY	32
8.5 HIDDEN CAMERA DETECTION	33

	8.6 MAGNETOMETER SENSOR	33
	8.7 SIREN ALERT	33
9.	CONCLUSION & FUTURE ENHANCEMENTS	34
	9.1 CONCLUSION	34
	9.2 FUTURE ENHANCEMENT	34
10.	APPENDIX	36
	10.1. SOURCE CODE	36
	10.2 SCREENSHOTS	51
11.	REFERENCES	57

LIST OF FIGURES

FIG.NO	FIGURE NAME	PAGE NO
6.3	Data Flow diagram	21
6.1	DFD Level 0	21
6.2	DFD Level 1	22
6.3	DFD Level 2	22

LIST OF ABBREVIATIONS

XML	Extensible Markup Language
GPS	Global Positioning System
OS	Operating System

CHAPTER 1

INTRODUCTION

The safety of women has been a major concern in society for a long time. Incidents of harassment, violence, and sexual assault against women are unfortunately common in many parts of the world. To address this issue, a women safety app has been developed to provide improved personal security. This app is designed to empower women with the tools and resources they need to feel safe and secure in their daily lives.

The women safety app offers various features such as emergency alerts, GPS tracking, real-time communication with trusted contacts, spy camera detection, and Siren alert to deviate the person. The app is easy to use and can be accessed from any mobile device. With just a single click, women can send an alert to their emergency contacts, providing them with their location and the nature of the emergency.

In addition to emergency features, the app also includes safety tips and resources for women. It offers information on self-defense techniques, safety precautions, and legal rights. Furthermore, the app provides a community forum where women can connect with other users to share their experiences and seek support.

1.1 PROBLEM DEFINITION

- The problem that the women safety app aims to address is the lack of safety and security for women in their daily lives.
- Women are often targets of harassment, violence, and sexual assault, which can create fear and anxiety and limit their mobility and independence.
- In many cases, women are not aware of the resources available to them or do not have access to them in a timely manner.
- The app provides women with a variety of tools and resources, including emergency alerts, GPS tracking, and real-time communication with trusted contacts.
- Overall, the women safety app aims to address the problem of women feeling unsafe and provide a comprehensive solution that can help reduce incidents of harassment and violence against women.

1.2 OBJECTIVE OF THE PROJECT

The objective of the women safety app is to provide improved personal security for women by empowering them with the tools and resources they need to feel safe and confident in their daily lives. The app aims to achieve this objective by:

- **Providing emergency features:** The app provides features such as emergency alerts, GPS tracking, and real-time communication with trusted contacts. In case of any emergency, women can easily send an alert to their emergency contacts with just a single click
- **Providing safety resources and tips:** The app provides women with safety tips and resources, including information on self-defense techniques, safety precautions, and legal rights.
- **Increasing awareness:** The app aims to increase awareness about the issue of women's safety and the resources available to women. By doing so, it can help reduce incidents of harassment and violence against women.

In this chatbot, our main objectives are as follows.

- Overall, the objective of the women safety app is to empower women with the tools and resources they need to feel safe and confident in their daily lives, thereby reducing incidents of harassment and violence against women and creating a safer environment for all.

1.3 SIGNIFICANCE OF THE PROJECT

- The significance of the women safety app for improved personal security lies in its potential to empower women and reduce incidents of harassment and violence against them.
- Empowering women
- Real-time communication
- Safety resources and tips.

1.4 OUTLINE OF THE PROJECT

Outline of the features that a women safety app for improved personal security could include:

- Emergency alert button that sends a distress signal to pre-set emergency contacts.
- GPS tracking to provide the location of the user in real-time.
- Information on self-defense techniques.
- Safety precautions to be taken in various situations.

CHAPTER 2

LITERATURE REVIEW

The Sexual Harassment of Women at Workplace (Prevention, Prohibition and Redressal) Act, 2013 is a legislative act in India that seeks to protect women from sexual harassment at their place of work. Today women are playing an important role as a president, prime minister, speaker of the Lok Sabha and even in the field of aeronautics, military, IPS, IAS, etc. Even today women have achieved top positions in job and society, yet they are facing problems such as physical harassment and the sexual assault.

The cases of harassment and rapes on women are increasing hence security issue for such woman is more important. So, it is essential to develop a system to provide security to women. In this he devised a system allows women to protect themselves from attackers. In recent days the attacks on women are increasing and sometimes they are not even able to take their mobile and dial-up to police, this system will help women in such situations to inform about attacks and also in giving their exact location to a nearby police station for necessary action.

In this, the author designed a device, in that, by pressing the button of the device a message along with her location will be transmitted by the system to the police station and her few relatives, so that they will get aware of her current situation. He told that with that message she is also for their defensive purpose they can able to give a shock to the attacker it will be more helpful to women at that critical situation, this system is designed as the defense equipment, it will them to attack the attacker. So, she has some time to rescue herself from that attacker.

The amount of violence against women has increased by many folds due to the greater exposure of women in every field of life. Women were previously restricted to the four walls of the houses and after globalization, they have got the chances. Women are now a day's cab drivers and they are also the CEO of top companies. It is the same mindset that restricts women to go out and work making them as a tool for domestication. This is one of the prime reason violence is increasing in India and women safety is a concern in India.

CHAPTER 3

SYSTEM ANALYSIS

System analysis is a problem solving technique that decay a system into component pieces of purpose of studying how well those component parts work and interact to accomplish their purpose. The following chapter provides a detailed description of the existing system. It also provides an overview of the proposed system and feasibility of women safety app.

3.1 EXISTING SYSTEM

These days women safety app is enabled with emergency SOS and tracking nearby police stations involves the use of mobile applications that enable users to alert authorities in case of an emergency situation. This app has limitations. For instance, the app may not be able to accurately track the user's location in areas with poor network connectivity or GPS signals. It does not have a feature to send messages to 3 members at a same time and to the near police station and a siren sound. These features will be helpful to the women who are in need of help.

3.2 DRAWBACKS

- Women news is not available in previous model of women safety app.
- The usage of the app can only be accessed within the app.
- There is no feature for detecting hidden camera.

3.3 PROPOSED SYSTEM

The proposed system for a women safety app with GPS tracking, detection of hidden cameras using a magnetometer, women's news, emergency SOS for sending distress signals to guardians, control room for emergency situations, a panic button for siren sound to deviate the person in conflict, and some extra features concerning women safety include the following:

- **GPS Tracking:** This feature will enable the app to track the user's location in real time and send the location details to the user's emergency contacts. This feature can be particularly

useful in case of kidnapping or abduction.

- **Detection of Hidden Camera using Magnetometer:** This feature will allow users to detect hidden cameras in their surroundings using their phone's magnetometer. The app will alert the user if it detects any magnetic field anomalies that could indicate the presence of hidden cameras.
- **Women's News:** This feature will provide users with the latest news related to women's safety and rights. It will help users stay informed about the latest developments in the field of women's safety.
- **Emergency SOS for Sending Distress Signals to Guardians:** This feature will enable users to send distress signals to their emergency contacts by pressing a button on their phones. The app will automatically send the distress signal along with the user's location details to their emergency contacts.
- **Emergency SOS for sending a distress signal to the guardian and control room:** In case of an emergency situation, users can send a distress signal to their guardian and the control room, providing them with their current location, details of the situation, and any other relevant information.
- **Panic Button for Siren Sound to Deviate the Person in Conflict:** This feature will allow users to press a button to sound a siren that can be used to distract or deviate the person in conflict. The sound will be loud enough to alert people nearby and can be used as a deterrent against attackers.
- **Extra Features Concerning Women Safety:** This feature will include additional safety features like safety tips, self-defense techniques, emergency contact numbers, and a database of safe location

3.4 FEASIBILITY STUDY

An analysis and evaluation of a proposed project to determine if it is technically feasible, is feasible within the estimated cost, and will be profitable. Feasibility studies are almost always conducted where large sums are at stake. A feasibility study aims to objectively and rationally uncover the strengths and weaknesses of an existing insurance agency applications and threats present in the environment, the resources required to carry through, and ultimately the prospects for women safety app for improved personal security.

3.4.1 Tests of Feasibility

Feasibility study is conducted once the problem is clearly understood. Feasibility study is necessary to determine that the proposed system in women safety app is feasible by considering the technical, operational, and economical factors. By having a detailed feasibility study the management in the will have a clear-cut view of the proposed system of the insurance bot. Feasibility study encompasses the following things:

- Technical Feasibility
- Economical Feasibility
- Operational feasibility

3.4.1.1 Technical Feasibility

A large part of determining resources has to do with assessing technical feasibility. It considers the technical requirements of the proposed project of women safety app. The technical requirements are then compared to the technical capability of women safety app for improved personal security. The systems project is considered technically feasible if the internal technical capability is sufficient to support the women security. The analyst must find out whether current technical resources can be upgraded or added to in a manner that fulfils the request under consideration.

The essential questions that help in testing the operational feasibility of a system include the following:

- Is the project feasible within the limits of current technology?
- Does technology exist at all?
- Is it available within given resource constraints?
- Is it a practical proposition?
- Manpower- programmers, testers & debuggers
- Software and hardware
- Are the current technical resources sufficient for the new system?
- Can they be upgraded to provide the level of technology necessary for the new

3.4.1.2 Operational Feasibility

Operational feasibility is dependent on human resources available for the project and involves projecting whether the system will be used if it is developed and implemented. Operational feasibility is a measure of how well a proposed system in women safety app solves the problems, and takes advantage of the opportunities identified during scope definition and how it satisfies the requirements identified in the requirements analysis phase of women safety app development.

The essential questions that help in testing the operational feasibility of a system include the following:

- Does current mode of operation provide adequate throughput and response time?
- Does current mode provide end users and managers with timely, pertinent, accurate and useful formatted information?
- Does current mode of operation provide cost-effective information services to the business?
- Could there be a reduction in cost and or an increase in benefits?

3.4.1.3 Economical Feasibility

Economic analysis could also be referred to as cost/benefit analysis. It is the most frequently used method for evaluating the effectiveness of a new system of the women safety app. In economic analysis the procedure is to determine the benefits and savings that are expected from a candidate system and compare them with costs.

If benefits outweigh costs, then the decision is made to design and implement the women safety app. An entrepreneur must accurately weigh the cost versus benefits before taking an action.

Possible questions raised in economic analysis are:

- Is the system cost effective?
- Do benefits outweigh costs and system study?

CHAPTER 4

SYSTEM SPECIFICATION

4.1 FUNCTIONAL REQUIREMENTS

4.1.1 SOS Alert

- The app will send an emergency alert to pre-configured contacts such as family or friends, along with the user's current location whenever the user shakes their phone.
- The app shall allow the user to customize the list of contacts that receive the SOS alert.
- The app shall use GPS or similar technology to determine the user's current location accurately.

4.1.2 Emergency service

- The app shall provide easy access to emergency services such as police, ambulance, or fire services in case of an emergency.
- The app shall provide contact details for emergency services.
- The app shall have a feature to call emergency services directly from the app.

4.1.3 Detection of Hidden Camera using Magnetometer

- This feature will allow users to detect hidden cameras in their surroundings using their phone's magnetometer.
- The app will alert the user if it detects any magnetic field anomalies that could indicate the presence of hidden cameras.

4.1.4 Login

The system should maintain a database to store all the login credentials.

4.1.5 Register

- The system should create an account within the app that is linked to the user's email address or phone number.

- Registering allows users to access the app's full range of features and functions, and it is often required in order to use the app.

4.1.6 Women's News

- This feature will provide users with the latest news related to women's safety and rights.
- It will help users stay informed about the latest developments in the field of women's safety

4.1.7 GPS Tracking

- It will enable the app to track the user's location in real time and send the location details to the user's emergency contacts.
- This feature can be particularly useful in case of kidnapping or abduction

4.1.8 Panic button

- It will allow users to press a button to sound a siren that can be used to distract or deviate the person in conflict.
- The sound will be loud enough to alert people nearby and can be used as a deterrent against attacker

4.2 NON-FUNCTIONAL REQUIREMENTS

4.2.1 Performance

- It will respond quickly to user inputs.
- The app shall not consume excessive battery or data usage.

4.2.2 Compatibility

- The app shall be compatible with a wide range of devices and operating systems.
- It work on both mobile data and Wi-Fi networks.

4.2.3 Security

- The app will protect user data and prevent unauthorized access.
- It have a feature to protect the user's data in case of a lost or stolen device.
- It also have a secure login process.

4.2.4 Maintainability

- The system should be easy to maintain.
- There should be a clear separation between the interface and the business logiccode.
- There should be a clear separation between the data access objects that map the database and the business logic code.

4.2.5 Exception handling

Exceptions should be reported effectively to the user if they occur.

4.2.6 Ethics

- The app must ensure the privacy and confidentiality of user data. The app should have clear policies about data collection, storage, and sharing.
- It should also provide users with options to control their data and choose the information they want to share.

4.3 HARDWARE REQUIREMENTS

Processor	: i5 processor
RAM	: 4 GB
Hard Disk	: 250 GB
Monitor	: 16'' Color Monitor
Keyboard	: Standard 110 keys
Pointing Device	: Mouse
Smart Phone	: Any type

4.4 SOFTWARE REQUIREMENTS

Programming Language : Java
Operating System : Windows/Ubuntu/Linux/Mac
Front End : XML
Back End : Firebase, Java
Editor : Android Studio

CHAPTER 5

SOFTWARE DESCRIPTION

A software requirements specification (SRS) is a description of a software system to be developed. It lays out functional and non-functional requirements, and may include a set of use cases that describe user interactions that the software must provide. Software requirements specification establishes the basis for an agreement between users and chat bot on what the software product is to do as well as what it is not expected to do. Software requirements specification permits a rigorous assessment of requirements before design can begin and reduces later redesign. It should also provide a realistic basis for estimating product costs, risks, and schedules.

5.1 FRONT END

The front end is designed using basic app development technologies like XML. It is the collaborative end to end bot platform made by developers for the developers. Here all the bot tools are integrated and it allows automatic detection of entities. It uses Machine Learning. API Connectivity is been done using PHP.

5.1.1 XML

XML (Extensible Markup Language) is a markup language used for storing and transporting data. It is a simple and flexible language that was designed to be both human-readable and machine-readable. XML uses tags to define elements and attributes to provide additional information about those elements. It is similar to HTML in that it uses a hierarchical structure, but unlike HTML, XML does not have predefined tags. XML is often used for data exchange between applications, as well as for storing data in databases. It is also used in web services and as a configuration file format for many software applications..

5.1.1.1 Features

- Self-Describing: XML documents are self-describing, which means that they contain metadata that describes the structure and meaning of the data. This makes it easy for software programs to understand and process the data.

- Platform and Application Independent: XML is platform-independent, meaning that it can be used on any hardware or software platform. It is also application-independent, meaning that it can be used with any software application that can read and process XML.
- Human Readable: XML documents are human-readable, which means that they can be easily understood and edited by humans using a text editor.
- Hierarchical Structure: XML documents have a hierarchical structure, which makes it easy to organize and structure complex data.
- Supports Unicode: XML supports Unicode, which means that it can be used to represent characters from any language or character set.

5.1.1.2 Advantages

- Platform independence: XML documents can be created, read, and processed on any hardware or software platform, which makes it an ideal format for data exchange and interoperability.
- Human-readable: XML documents are easy to read and understand, which makes it easy for humans to manually read and edit the contents of the document.
- Self-describing: XML documents are self-describing, meaning that they contain metadata that describes the structure and meaning of the data. This makes it easy for software applications to understand and process the data.
- Flexibility: XML is a flexible language that can be customized to meet the needs of specific applications or industries. Users can define their own elements and attributes to represent data in a way that is specific to their needs.
- Validation: XML documents can be validated against a schema or DTD (Document Type Definition) to ensure that they conform to a specific format and structure.
- Separation of presentation and content: XML separates the presentation of data from the actual content, which allows for greater flexibility in how the data is displayed and presented.
- Standardization: XML is a widely adopted standard, which means that it is supported by a large number of software applications and development tools.
- Compatibility with other data formats: XML can be easily transformed into other data

formats, such as HTML, PDF, or CSV, making it a versatile format for data exchange and storage.

BACK END

The back end is designed using Firebase, whose primary function is to store data securely and retrieve it later, as requested by other software applications.

5.2.1 Firebase

Google Firebase is Google-backed application development software which allows developers to develop Android, IOS, and Web apps. For reporting and fixing app crashes, tracking analytics, creating marketing and product experiments, firebase provides several tools.

Firebase manages real-time data in the database. So, it easily and quickly exchanges the data to and from the database. Hence, for developing mobile apps such as live streaming, chat messaging, etc., we can use Firebase. It allows syncing real-time data across all devices - iOS, Android, and Web - without refreshing the screen. Firebase provides integration to Google Advertising, AdMob, Data Studio, BigQuery, DoubleClick, Play Store, and Slack to develop our apps with efficient and accurate management and maintenance. Everything from databases, analytics to crash reports are included in Firebase. So, the app development team can stay focused on improving the user experience. Firebase applications can be deployed over a secured connection to the firebase server. Firebase offers a simple control dashboard.

5.2.2 Features of Firebase

- **Cloud Messaging** : Firebase allows us to deliver and receive messages in a more reliable way across platforms.
- **Authentication** : Firebase has little friction with acclaimed authentication.
- **Test Lab** : Test in the lab instead on your users.
- **Hosting** : Firebase delivers web content faster.
- **Remote Configuration** : It allows us to customize our app on the go.
- **Dynamic Links** : Dynamic Links are smart URLs which dynamically change behavior for providing the best experience across different platforms.
- **Crash Reporting** : It keeps our app stable.
- **Real-time Database** : It can store and sync app data in real-time.

- **Storage** : We can easily store the file in the database.

5.2.3 Advantages of Firebase

- Fast and Safe hosting
- Reliable and Extensive databases
- Google Analytics
- Free Multi-Platform firebase authentication
- Firebase Testing Service to Improve App Quality
- Free Use of Firebase Dynamic Links

5.2.4 JAVA

Java is a high-level, general-purpose programming language that is widely used for developing software applications. It was first released in 1995 by Sun Microsystems, and later acquired by Oracle Corporation. Java is designed to be platform-independent, meaning that it can run on any hardware or software platform that has a Java Virtual Machine (JVM) installed.

The primary parts of java include,

- **Java Development Kit (JDK):** The JDK is a software development kit that includes everything needed to develop and run Java applications, including the Java compiler, runtime environment, and libraries.
- **Java Virtual Machine (JVM):** The JVM is a runtime environment that interprets Java bytecode and executes Java applications. The JVM is responsible for translating Java bytecode into machine code that can be executed by the underlying hardware.
- **Java Standard Edition (Java SE):** Java SE is a version of Java that includes the core Java APIs and libraries used for developing desktop and server applications.
- **Java Enterprise Edition (Java EE):** Java EE is a version of Java that includes additional APIs and libraries used for developing large-scale, enterprise-level applications.
- **JavaFX:** JavaFX is a platform for building rich, interactive graphical user interfaces (GUIs) for Java applications.

- **Integrated Development Environment (IDE):** An IDE is a software application that provides a comprehensive environment for developing, testing, and debugging Java applications. Some popular Java IDEs include Eclipse, NetBeans, and IntelliJ IDEA

5.2.5 Features of Java

- **Platform-independent:** Java programs can run on any operating system with the help of the Java Virtual Machine (JVM), which interprets the Java bytecode into machine-readable code.
- **Object-oriented:** Java is based on the object-oriented programming (OOP) paradigm, which emphasizes the use of objects, classes, and inheritance to organize code and simplify complex problems.
- **Simple and easy to learn:** Java has a relatively simple syntax and is easy to learn, making it an ideal language for beginners.
- **Robust:** Java is designed to be robust and can handle errors and exceptions gracefully, preventing crashes and improving the overall reliability of the code.
- **Secure:** Java provides a high level of security through its built-in security features, such as its sandbox environment, which prevents untrusted code from accessing system resources.
- **Multithreaded:** Java supports multithreading, which allows multiple threads to run simultaneously, making it possible to write programs that can take advantage of multiple cores and processors.
- **Scalable:** Java is highly scalable and can be used to develop applications of any size, from small desktop applications to large-scale enterprise systems.
- **High performance:** While Java is not the fastest programming language, it is known for its high performance and is capable of running complex applications efficiently.
- **Rich APIs:** Java has a large number of libraries and APIs that make it easy to write code for a wide range of applications, from simple desktop applications to complex web-based systems.
- **Open source:** Java is open source, meaning that its source code is freely available for anyone to use, modify, and distribute. This has helped to make Java a popular choice for developers and has led to the creation of a large and active Java

community.

5.2.6 Advantages Of Java

- Easy to learn.
- Platform-independent.
- Secured
- Multi-Thread

CHAPTER 6

PROJECT DESCRIPTION

This app ensures the safety of women. it helps to identify and call on resources to help the one out of dangerous situations. These reduce risk and bring assistance when we are in danger the help us to send the location to the contacts. The app we designed is to provide security to women is the main purpose of this app to provide the awareness on the time of critical situation for women.

6.1 OVERVIEW OF THE PROJECT

The women safety app offers various features such as emergency alerts, GPS tracking, real-time communication with trusted contacts, spy camera detection, and Siren alert to deviate the person. The app is easy to use and can be accessed from any mobile device. With just a single click, women can send an alert to their emergency contacts, providing them with their location and the nature of the emergency.

The Senora women safety app allows you to:

- Emergency SOS Alert
- GPS tracking
- Alert message with current location of the user
- Emergency call when power button is clicked
- Detection of Hidden Camera using Magnetometer
- Siren Alert
- Every 2hours location Sharing in normal message
- Women's News
- Alert message with call feature activated when phone is locked for 24hours

6.2 MODULE DESCRIPTION

6.2.1 UI Design and User Authentication Check

If the user is new to the application, then he/she want to register it with their email-id and password. If they are a registered user, then they can login to the page using their credentials and we have designed user-friendly UI design for the user's.

6.2.2 Emergency Alert using button click and sending notification with exact location

Emergency alert helps to identify and call on resources to help the one out of dangerous situations. These reduce risk and bring assistance when we are in danger the help us to send the location to the contact by clicking power button for 3 times.

6.2.3 Siren Alert for distraction and Women's news on daily basis

Siren alert is used for distraction and safety purpose, when you are in island you can use this siren alert feature to call the boat/aero plan which are far away from the user and sound is similar to the police siren.

Women's news feature will provide users with the latest news related to women's safety and rights, also help users stay informed about the latest developments in the field of women's safety

6.2.4 Spy Camera Detection and Manual tip for finding Hidden Camera

This feature will allow users to detect hidden cameras in their surroundings using their phone's magnetometer. The app will alert the user if it detects any magnetic field anomalies that could indicate the presence of hidden cameras

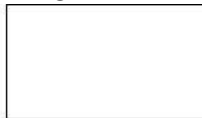
6.2.5 Firebase Storage

Firebase is used to store the user information, which are entered during the registration process. User can register with their email id and password; this information's are stored in the backend firebase storage.

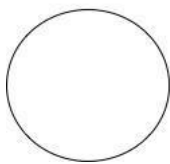
6.3 DATA FLOW DIAGRAM

Data flow diagram is used to describe how the information is processed and stored and identifies how the information flows through the processes. Data flow diagram illustrates how the data is processed by a system in terms of inputs and outputs. The data flow diagram also depicts the flow of the process and it has various levels. The initial level is context level which describes the entire system functionality and the next level describes each and every sub module in the main system as a separate process or describes all the process involved in the system separately.

Data flow diagram are made up of number of symbols,



Square representing external entities, which are sources or destinations of data.



Circle representing processes, which take data as input, do something to it and output it.



Arrows representing the data flows, which can either, be electronic data or physical items.



Parallel lines representing data stores, including electronic stores such as databases or XML files and physical stores

6.3.1 DFD Level 0

The users of the Women Safety App Clicks the Instant Messaging Button the Messaging Function gets initiated and the message is sent to the Recipient and the Instant Call Button the Call is initiated to the Emergency Contact.

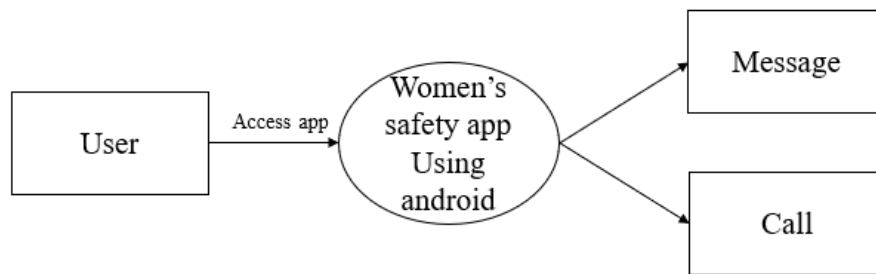


Fig.6.1 DFD Level 0

6.3.2 DFD Level 1

As the User Opens the SAFE Application, the User Login to the Application, after giving the Login credentials, the Firebase Authentication Takes place. After the successful Login the User Id gets added to the Online List and the Latitude and the Longitude of the User is obtained and it is stored in the Firebase. Upon clicking the User's id the Application takes to the Map Activity which Displays the Current Location of their friend and these marker on the map moves as the user moves through.

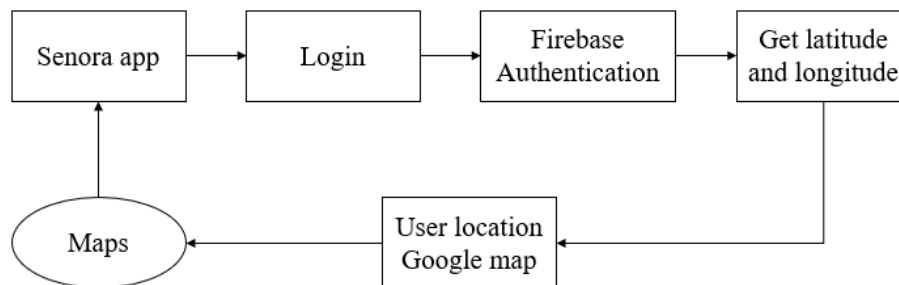


Fig.6.2 DFD Level 1

6.3.3 DFD level 2

Spy camera detection will allow users to detect hidden cameras in their surroundings using their phone's magnetometer. The app will alert the user if it detects any magnetic field anomalies that could indicate the presence of hidden cameras.

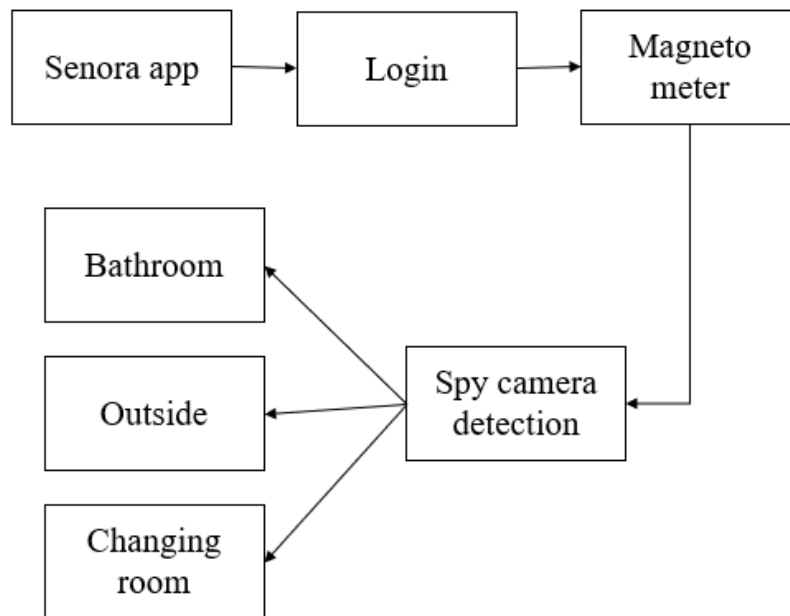


Fig.6.3 DFD Level 2

6.4 ARCHITECTURE DIAGRAM

The architecture diagram for a women safety app typically outlines the various components of the app and how they interact with each other. Here is an explanation of each component in the proposed system:

1. **User Interface:** The user interface is the front-end component of the app that users interact with. It includes screens for various features such as GPS tracking, hidden camera detection, women's news, emergency SOS, panic button, and other extra features.
2. **App Logic:** This component is responsible for handling user inputs and processing them to provide appropriate outputs. It includes algorithms for GPS tracking, hidden camera detection, panic button, and other features.
3. **Database:** The database component stores user information, such as emergency contacts, safe zones, and preferences.

4. Hidden Camera Detection Module: This module uses the magnetometer sensor on the user's smartphone to detect any hidden cameras in their surroundings and alert them.
5. GPS Tracking Module: This module tracks the user's location in real-time and alerts their trusted contacts if they leave a predefined safe zone.
6. Panic Button Module: This module is responsible for producing a loud siren sound to deviate the person in conflict and attract attention from others.
7. Emergency SOS Module: This module allows users to send distress signals to their guardians or a control room in case of an emergency.

6.4.1 Women Safety App

In our project app, ensures the safety of women. it helps to identify and call on resources to help the one out of dangerous situations. These reduce risk and bring assistance when we are in danger the help us to send the location to the contacts

All the information's are stored and retrieved from the firebase storage database. In login screen we collect the basic details such as Email id, password in login page from the user.

The Diagram is being given for all the processes such as the process flow is been specified.

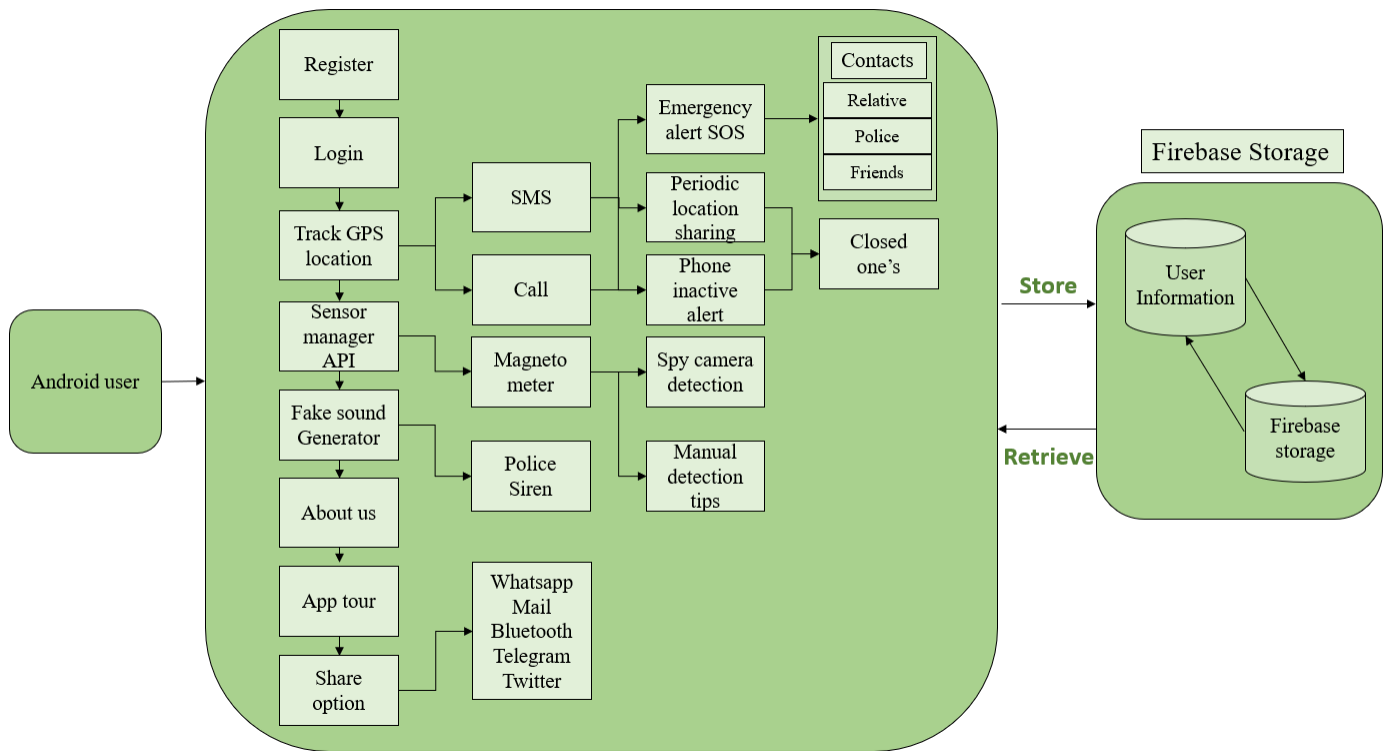


Fig.6.4 ARCHITECTURE DIAGRAM

6.5 DATABASE DESIGN

Database design is the process of producing a detailed data model of database. This data model contains all the needed logical and physical design choices and physical storage parameters needed to generate a design in a data definition language, which can then be used to create a database. A fully attributed data model contains detailed attributes for each entity.

The term database design can be used to describe many different parts of the design of an overall database system. Principally, and most correctly, it can be thought of as the logical design of the base data structures used to store the data. In the relational model these are the tables and views. In an object database the entities and relationships map directly to object classes and named relationships. However, the term database design could also be used to apply to the overall process of designing, not just the base data structures, but also the forms and queries used as part of the overall database application within the database management system (DBMS).

TABLE 6.6.1 User login detail

S.NO	FIELD NAME	DATATYPE
1	Email id	Varchar
2	passsword	Varchar

TABLE 6.6.2 User's Registration Details

S.NO	FIELD NAME	FIELD TYPE
1	Email id	Varchar
3	password	Varchar

TABLE 6.6.3 User's Details

S.NO	FIELD NAME	DATATYPE
1	Email id	Varchar
2	passsword	Varchar

6.6 INPUT DESIGN

The input design is the link between the information system and the user. It comprises the developing specification and procedures for data preparation and those steps are necessary to put transaction data in to a usable form for processing can be achieved by inspecting the computer to read data from a written or printed document or it can occur by having people keying the data directly into the system. The design of input focuses on controlling the amount of input required, controlling the errors, avoiding delay, avoiding extra steps and keeping the process simple. The input is designed in such a way so that it provides security and ease of use with retaining the privacy. Input Design considered the following things:

- What data should be given as input?
- How the data should be arranged or coded?
- The dialog to guide the operating personnel in providing input.
- Methods for preparing input validations and steps to follow when error occur.

Input Design is the process of converting a user-oriented description of the input into a computer-based system. This design is important to avoid errors in the data input process and show the correct direction to the management for getting correct information from the computerized system.

It is achieved by creating user-friendly screens for the data entry to handle large volume of data. The goal of designing input is to make data entry easier and to be free from errors. The data entry screen is designed in such a way that all the data manipulates can be performed. It also provides record viewing facilities.

When the data is entered it will check for its validity. Data can be entered with the help of screens. Appropriate messages are provided as when needed so that the user will not be in maize of instant. Thus the objective of input design is to create an input layout that is easy to follow.

6.7 OUTPUT DESIGN

A quality output is one, which meets the requirements of the end user and presents the information clearly. In any system results of processing are communicated to the users and to other system through outputs. In output design it is determined how the information is to be displaced for immediate need and also the hard copy output. It is the most important and direct source information to the user. Efficient and intelligent output design improves the system's relationship to help user decision-making.

CHAPTER 7

SYSTEM TESTING

System Testing is a level of the software testing where complete and integrated software is tested. The purpose of this test is to evaluate the system's compliance with the specified requirements. By definition of ISTQB system testing is the process of testing an integrated system to verify that it meets specified.

7.1 TESTING METHODS

Software Testing Type is a classification of different testing activities into categories, each having, a defined test objective, test strategy, and test deliverables. The goal of having a testing type is to validate the Application under Test for the defined TestObjective.

For instance, the goal of Accessibility testing is to validate the AUT to be accessible by disabled people. So, if your Software solution must be disabled friendly, you check it against Accessibility Test Cases.

7.2 TYPES OF TESTING

7.2.1 Unit Testing

In computer programming, unit testing is a software testing method by which individual units of source code, sets of one or more computer program modules together with associated control data, usage procedures, and operating procedures, are tested to determine whether they are fit for use.

In this women safety app, every units of code is been tested and the correctness of every module is been ensured.

7.2.2 Integration Testing

Integration testing (sometimes called integration and testing, abbreviated I&T) is the phase in software testing in which individual software modules are combined and tested as a group. It occurs after unit testing and before validation testing. Integration testing takes as its input modules that have been unit tested, groups them in larger aggregates,

applies tests defined in an integration test plan to those aggregates, and delivers as its output the integrated system ready for system testing. In this women safety app, the units are been tested as a whole and the testing was successful.

7.2.3 Functional Testing

Functional testing is a quality assurance (QA) process and a type of black-box testing that bases its test cases on the specifications of the software component under test. Functions are tested by feeding them input and examining the output, and internal program structure is rarely considered (unlike white-box testing). Functional testing usually describes what the system does. Functional testing does not imply that you are testing a function (method) of your module or class. Functional testing tests a slice of functionality of the whole system.

Functional testing has many types:

- Smoke testing
- Sanity testing
- Regression testing
- Usability testing

7.2.4 Stress Testing

Stress testing a Non-Functional testing technique that is performed as part of performance testing. During stress testing, the system is monitored after subjecting the system to overload to ensure that the system can sustain the stress.

Reasons can include:

- to determine breaking points or safe usage limits
- to confirm mathematical model is accurate enough in predicting breaking points or safe usage limits
- to confirm intended specifications are being met
- to determine modes of failure (how exactly a system fails)
- to test stable operation of a part or system outside standard usage

The recovery of the system from such phase (after stress) is very critical as it is highly likely to happen in production environment.

In this women safety app, whole of the modules are been tested and it has the safe usagemasures.

7.2.5 Acceptance Testing

Acceptance Testing is a level of the software testing where a system is tested for acceptability. The purpose of this test is to evaluate the system's compliance with the business requirements and assess whether it is acceptable for delivery.

Formal testing with respect to user needs, requirements, and business processes conducted to determine whether or not a system satisfies the acceptance criteria and to enable the user, customers or other authorized entity to determine whether or not to accept the system.

In this women safety app, the customer's acceptance is been monitored and it is been put into usage.

7.2.6 White Box Testing

White Box Testing is the testing of a software solution's internal coding and infrastructure. It focuses primarily on strengthening security, the flow of inputs and outputs through the application, and improving design and usability. White box testing is also known as Clear Box testing, Open Box testing, Structural testing, Transparent Box testing, Code-Based testing, and Glass Box are testing. It is one of two parts of the "box testing" approach of software testing. Its counter-part, black box testing, involves testing from an external or end-user type perspective. On the other hand, White box testing is based on the inner workings of an application and revolves around internal testing.

The term "white box" was used because of the see-through box concept. The clear box or white box name symbolizes the ability to see through the software's outer shell (or "box") into its inner workings. Likewise, the "black box" in "black box testing" symbolizes not being able to see the inner workings of the software so that only the end- user experience can be tested.

In this women safety app, all the inner functionality is been tested and it is been correctly implemented.

7.2.7 BlackBox Testing

Black box testing is a software testing techniques in which functionality of the software under test (SUT) is tested without looking at the internal code structure, implementation details and knowledge of internal paths of the software. This type of testing is based entirely on the software requirements and specifications.

In this women safety app, the implementation part is been checked for its correctness.

7.2.7.1 Methods of Black Box Testing

There are many types of Black Box Testing but following are the prominent ones -

- Functional testing - This black box testing type is related to functional requirements of a system; it is done by software testers.
- Non-functional testing - This type of black box testing is not related to testing of a specific functionality, but non-functional requirements such as performance, scalability, usability.
- Regression testing - Regression testing is done after code fixes, upgrades or any other system maintenance to check the new code has not affected the existing code.

7.3 TESTING STRATEGY

Test Strategy is also known as test approach defines how testing would be carried out. Test approach has two techniques:

- Proactive - An approach in which the test design process is initiated as early as possible in order to find and fix the defects before the build is created.
- Reactive - An approach in which the testing is not started until after design and coding are completed.

Test strategy calls for implementing two entirely different methodologies for testingthis project. The women safety app includes a fair amount of manual UI-based testing.

CHAPTER 8

SYSTEM IMPLEMENTATION

The system is been implemented as follows:

8.1 ACCOUNT CREATION

It is handy to create an account in the recast framework in which it prompts for a simple login with the mail id and the password credentials, on verification of the mail id, it allows the user to create account on the framework.

8.2 DATABASE INTEGRATION

To create a women safety app, you would need to identify the specific requirements and functionalities of the app and it is to store the credentials of the user. We need to give user profile, location data, emergency contacts, trusted contacts, safety tips, SOS alerts and incidents, app usage analytics, feedback and reviews. The things that are needed can be selected by the user and finally account is created.

8.3 USER AUTHENTICATION USING EMAIL AND PASSWORD

To implement user authentication using email and password for women safety app the first step is to give the user registration and next to give the password hashing and then user database then the user should give user login, password verification.

8.4 SOS EMERGENCY

Implementing an SOS emergency feature in a women safety app is crucial for providing immediate assistance in critical situation. The SOS emergency is then made for the women to ensure their reliability and effectiveness. SOS emergency building forms the main task in creating the emergency which allows creating triggers, setting the corresponding requirements and performs necessary actions like alert notification, emergency contact number. This is main motive in performing the building task.

8.4.1 HIDDEN CAMERA DETECTION

Implementing hidden camera detection in a women safety app can help the users to identify potential privacy threats and ensure their safety. However, its important to note that detecting hidden cameras is a complex task and may not be foolproof. It's important to note that hidden camera detection is a challenging task, and the effectiveness of the feature can vary depending on the device's hardware and limitations. It is to ensure that the implementation of such features compiles with local laws and regulations concerning privacy and surveillance.

8.4.2 MAGNETOMETER SENSOR

The magnetometer sensor can be utilized in a women safety app to provide additional functionality and enhance user safety. Factors like electromagnetic interference or the presence of large metal objects can affect the sensor's performance. It is to ensure that the app provides appropriate instructions to users for optimal usage and educates them about any limitations associated with the magnetometer sensor. Additionally, take into consideration any legal or privacy concerns when implementing features that involve location tracking or detection of hidden objects.

8.4.3 SIREN ALERT

Implementing a siren alert feature in a women safety app can help draw attention and deter potential threats in emergency situations. The proper usage of the siren alert feature, including when and how to use it responsibly. It is to ensure that the implementation of the siren alert feature complies with local regulations and does not cause unnecessary panic or disturbance.

CHAPTER 9

CONCLUSION & FUTURE ENHANCEMENTS

9.1 CONCLUSION

In conclusion, a women safety app for improved personal security can be an effective solution for reducing incidents of harassment and violence against women. By providing emergency features, safety resources and tips, community support, and additional features, the app can empower women and provide them with the necessary tools to stay safe. The literature review highlights the importance of user-friendliness, accessibility, and ongoing development to ensure the effectiveness of such apps. It is essential to design the app in a way that is easy to use, with features that can be easily accessed in case of an emergency. In addition to providing immediate support and assistance during emergencies, a women safety app can also create a sense of community among women, allowing them to share their experiences and provide support to one another. Overall, a comprehensive women safety app can be an effective tool in creating a safer environment for women. By promoting awareness, providing support and resources, and facilitating communication, the app can help women feel more secure and confident in their daily lives.

9.2 FUTURE ENHANCEMENT

This project is focused on women safety that easily interact with the user. One possible future enhancement could be the integration of AI, which can help the app learn the user's behavior, preferences, and patterns to provide personalized safety recommendations and alerts.

- **Wearable Technology Integration:** Another possible future enhancement is the integration of wearable technology like smartwatches or fitness bands to provide an additional layer of safety and monitoring.
- **Voice-activated Features:** Voice-activated features could be another potential enhancement, which will allow users to activate safety features using their voice commands.

- Community-based Safety: Future development could also incorporate a community-based safety feature that allows users to share their location with their trusted contacts, who can then monitor their safety in real time.
- Machine Learning: Machine learning can be applied to analyze and identify patterns in data collected by the app, which can be used to provide more personalized safety recommendations and alerts.

CHAPTER 10

APPENDIX

10.1. SOURCE CODE

AndroidManifest.xml :

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.teamDroiders.ladybuddy">
    <uses-feature
        android:name="android.hardware.telephony"
        android:required="true" />
    <uses-permission android:name="android.permission.ACCESS_COARSE_LOCATION" />
    <uses-permission android:name="android.permission.ACCESS_FINE_LOCATION" />
    <uses-permission android:name="android.permission.SEND_SMS" />
    <uses-permission android:name="android.permission.INTERNET" />
    <application
        android:allowBackup="true"
        android:icon="@mipmap/ic_launcher"
        android:label="@string/app_name"
        android:roundIcon="@mipmap/ic_launcher"
        android:supportsRtl="true"
        android:theme="@style/AppTheme">
        <activity
            android:name=".Register"
            android:exported="false" />
        <activity
            android:name=".Login"
            android:exported="false" />
        <activity
            android:name=".AboutUs"
            android:exported="true"
            android:label="AboutUs"
            android:parentActivityName=".MainActivity"
            android:theme="@style/Theme.New" />
            android:exported="true"
            android:label="Manual Detection"
            android:parentActivityName=".ChoosenActivity"
            android:theme="@style/Theme.New" />
        <activity
            android:name=".SmsActivity"
            android:label="Emergency Service"
            android:parentActivityName=".MainActivity"
            android:theme="@style/Theme.New" />
        <activity android:name=".CurrentLocation" />
        <activity android:name=".Flashing" />
```

```

<activity android:name=".SplashActivity"
    android:exported="true">
    <intent-filter>

<receiver android:name=".ScreenOnOffReceiver"
    android:exported="true">
    <intent-filter>
        <action android:name="android.intent.action.SCREEN_OFF" />
        <action android:name="android.intent.action.SCREEN_ON" />
        <action android:name="number.action.string" />
    </intent-filter>
</receiver>

<service
    android:name=".ScreenOnOffBackgroundService"
    android:enabled="true"
    android:exported="false" />
</application>

</manifest>

```

activity_sms.xml :

```

<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout
    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="@drawable/bgapps"
    tools:context=".SmsActivity">
    <LinearLayout
        <com.google.android.material.textfield.TextInputLayout
            android:id="@+id/textInputLayout1"
            android:layout_width="match_parent"
            android:layout_height="match_parent"
            android:padding="8dp"
            android:theme="@style/EditTextThemeOverlay"
            app:layout_constraintBottom_toTopOf="@+id/textInputLayout2"
            app:layout_constraintEnd_toEndOf="parent"
            app:layout_constraintStart_toStartOf="parent"
            android:textColorHint="@color/design_default_color_on_secondary"
            app:layout_constraintTop_toTopOf="parent">
        <com.google.android.material.textfield.TextInputEditText
            android:id="@+id/txt_phone_number1"
            android:layout_width="match_parent"
            android:layout_height="match_parent"
            android:layout_marginStart="10dp"

```

```

        android:layout_marginLeft="10dp"
        android:textColor="@color/design_default_color_on_secondary" />
    </com.google.android.material.textfield.TextInputLayout>
</androidx.cardview.widget.CardView>

```

```

<androidx.cardview.widget.CardView
    android:layout_width="match_parent"

```

```

    <com.google.android.material.textfield.TextInputLayout
        android:id="@+id/textInputLayout2"
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:padding="8dp"
        android:textColorHint="@color/design_default_color_on_secondary"
        android:theme="@style/EditTextThemeOverlay"

```

```

">

```

```

    <com.google.android.material.textfield.TextInputEditText
        android:id="@+id/txt_phone_number2"
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:layout_marginLeft="10dp"
        android:layout_marginRight="10dp"
        android:textColor="@color/design_default_color_on_secondary" />
    </com.google.android.material.textfield.TextInputLayout>
</androidx.cardview.widget.CardView>

```

```

    <com.google.android.material.textfield.TextInputLayout
        android:id="@+id/textInputLayout3"
        android:layout_width="match_parent"
        android:layout_height="match_parent"

```

```

    <com.google.android.material.textfield.TextInputLayout
        android:id="@+id/textInputLayout4"
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:padding="8dp"
        android:textColorHint="@color/design_default_color_on_secondary"
        android:theme="@style/EditTextThemeOverlay"
    </androidx.cardview.widget.CardView>

```

```

<androidx.cardview.widget.CardView
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:layout_margin="12dp"
    android:elevation="5dp"
    app:cardBackgroundColor="@color/white"
    app:cardCornerRadius="50dp">
    </com.google.android.material.textfield.TextInputLayout>

```



```
</androidx.cardview.widget.CardView>
</LinearLayout>
```

```
<Button
    android:id="@+id/button"
    android:layout_width="0dp"
    android:layout_height="wrap_content"
    android:layout_marginStart="9dp"
    app:layout_constraintStart_toEndOf="@+id/Save_btn"
    app:layout_constraintTop_toBottomOf="@+id/linearLayout" />
```

```
</androidx.constraintlayout.widget.ConstraintLayout>
```

activity_register.xml :

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout 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:background="@drawable/bgapps"
    android:orientation="vertical"
    android:padding="15dp"
    android:gravity="center"
    android:layout_height="match_parent"
    tools:context=".Register">

    <TextView
        android:text="@string/register"
        android:textSize="25sp"
        android:textColor="@color/white"
        android:layout_height="wrap_content" />

    <com.google.android.material.textfield.TextInputLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content">
    </com.google.android.material.textfield.TextInputLayout>

        <com.google.android.material.textfield.TextInputEditText
            android:id="@+id/password"
            android:layout_height="wrap_content"/>
    </com.google.android.material.textfield.TextInputLayout>

    <ProgressBar
        android:id="@+id/progressBar"
        android:visibility="gone"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"/>
```

```

<TextView
    android:textStyle="bold"
    android:textSize="20sp"
    android:gravity="center"
    android:layout_marginTop="20dp"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"/>

```

```

</LinearLayout>

```

MainActivity.java :

```

package com.teamDroiders.ladybuddy;

import androidx.appcompat.app.AlertDialog;
import androidx.appcompat.app.AppCompatActivity;
import androidx.cardview.widget.CardView;
import com.google.firebase.auth.FirebaseAuth;
import com.google.firebase.auth.FirebaseUser;

public class MainActivity extends AppCompatActivity {

    FirebaseAuth auth;
    Button buttonlogout;
    TextView textView;
    FirebaseUser user;
    CardView siren, location, Settings, currentlocation, community, news, aboutUs, shareBtn;
    @SuppressWarnings("MissingInflatedId")
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate( savedInstanceState );
        setContentView( R.layout.activity_main );

        Intent backgroundService = new Intent( getApplicationContext(), ScreenOnOffBackgroundService.class
    );
        this.startService( backgroundService );
        Log.d( ScreenOnOffReceiver.SCREEN_TOGGLE_TAG, "Activity onCreate" );
        int permissionCheck = ContextCompat.checkSelfPermission (MainActivity.this,
Manifest.permission.SEND_SMS);
        if (permissionCheck != PackageManager.PERMISSION_GRANTED &&
ContextCompat.checkSelfPermission (MainActivity.this,

        final AlertDialog.Builder alert = new AlertDialog.Builder(MainActivity.this);
        View mView = getLayoutInflater().inflate(R.layout.custom_dialog_mainactivity,null);

        Button btn_okay = (Button)mView.findViewById(R.id.btn_okay);
        TextView heading=mView.findViewById (R.id.heading);
        heading.setText("LadyBuddy needs access to");

```

```

        TextView sms=mView.findViewById (R.id.sms);
        locationText.setText("Messaging embedded with live location needs Location permission");
        TextView call=mView.findViewById (R.id.call);
        call.setText("Phone Call:-");
        declaration.setText("Declaration");
        TextView declaratioText=mView.findViewById (R.id.textDeclaration);
        declaratioText.setText("The app is solely developed by INDIAN Developers and all data related to
this app is stored locally in your phone.");

        checkbox.setText("");
        checkBoxtext.setText(Html.fromHtml("I accept the " +
        "<a href='https://www.websitepolicies.com/policies/view/IaK4RjyB'>PRIVACY
POLICY</a>"+" of the app"));
        checkBoxtext.setClickable(true);
        checkBoxtext.setMovementMethod(LinkMovementMethod.getInstance());
        public void onClick(View v) {
            if(checkbox.isChecked () {

                alertDialog.dismiss ();
            });
            alertDialog.show();

        }

        siren = findViewById( R.id.Siren );
        location = findViewById( R.id.send_Location );
        Settings = findViewById( R.id.Settings );
        // community=findViewById (R.id.community);
        siren.setOnClickListener( new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                startActivity( new Intent( getApplicationContext(), Flashing.class ) );
            }
        } );
        news.setOnClickListener( new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                startActivity( new Intent( getApplicationContext(), NewsActivity.class ) );
            }
        } );

        shareBtn = findViewById(R.id.ShareBtn);
        shareBtn.setOnClickListener(new View.OnClickListener() {

            @Override
            public void onClick(View v) {

                try {
                    Intent shareIntent = new Intent(Intent.ACTION_SEND);
                    shareIntent.setType("text/plain");

```

```

        shareIntent.putExtra(Intent.EXTRA_SUBJECT, "LadyBuddy");

        shareMessage = shareMessage + "https://play.google.com/store/apps/details?id=" +
BuildConfig.APPLICATION_ID ;
    });

    auth = FirebaseAuth.getInstance();
    buttonlogout = findViewById( R.id.btn_logout);
    textView = findViewById( R.id.user_details );
    user = auth.getCurrentUser();
    if (user == null){
        Intent intent = new Intent(getApplicationContext(), Login.class);
        startActivity(intent);
        finish();
    }
    else {
        textView.setText(user.getEmail());
    }

    buttonlogout.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View v) {
            FirebaseAuth.getInstance().signOut();
            Intent intent = new Intent(getApplicationContext(), Login.class);
            startActivity(intent);
            finish();
        }
    }
}

```

Login.java :

```

package com.teamDroiders.ladybuddy;

import androidx.annotation.NonNull;
import androidx.appcompat.app.AppCompatActivity;
import com.google.firebase.auth.AuthResult;
import com.google.firebase.auth.FirebaseAuth;
import com.google.firebase.auth.FirebaseUser;

public class Login extends AppCompatActivity {

    TextInputEditText editTextEmail, editTextPassword;
    Button buttonLogin;
    FirebaseAuth mAuth;
    ProgressBar progressBar;
    TextView textView;
    @Override
    public void onStart() {
        super.onStart();
        FirebaseUser currentUser = mAuth.getCurrentUser();
        if(currentUser != null){
            Intent intent = new Intent(getApplicationContext(), MainActivity.class);
            startActivity(intent);
        }
    }
}

```

```

        finish();
    }
}

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

    mAuth = FirebaseAuth.getInstance();
    editTextEmail = findViewById(R.id.email);
    textView.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View v) {
            Intent intent = new Intent(getApplicationContext(), Register.class);
            startActivity(intent);
            finish();
        }
    });

    buttonLogin.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View v) {
            progressBar.setVisibility(View.VISIBLE);
            String email, password;
            if (TextUtils.isEmpty(email)){
                Toast.makeText(Login.this, "Enter email", Toast.LENGTH_SHORT).show();
                return;
            }

            Toast.makeText(getApplicationContext(), "Login Successful",
Toast.LENGTH_SHORT).show();
            Intent intent = new Intent(getApplicationContext(), MainActivity.class);
            startActivity(intent);
            finish();

        } else {
            Toast.makeText(Login.this, "Authentication failed.",
                Toast.LENGTH_SHORT).show();
        }
    }
}

```

SmsActivity.java :

```

package com.teamDroids.ladybuddy;

import android.Manifest;
import android.content.BroadcastReceiver;
import com.google.android.gms.location.FusedLocationProviderClient;

```

```

import com.google.android.gms.location.LocationServices;
import com.google.android.gms.tasks.OnCompleteListener;

public class SmsActivity extends AppCompatActivity {

    TextInputEditText txt_pnumber1, txt_msg, txt_pnumber2, txt_pnumber3, txt_pnumber4;
    Button Save;

    FusedLocationProviderClient fusedLocationProviderClient;
    String prevStarted = "yesSms";
    @Override
    protected void onResume() {
        super.onResume();
        SharedPreferences sharedPreferences = getSharedPreferences(getString(R.string.app_name),
Context.MODE_PRIVATE);
        if (!sharedPreferences.getBoolean(prevStarted, false)) {
            SharedPreferences.Editor editor = sharedPreferences.edit();
            editor.putBoolean(prevStarted, Boolean.TRUE);
            editor.apply();

            final AlertDialog.Builder alert = new AlertDialog.Builder(SmsActivity.this);
            View mView = getLayoutInflater().inflate(R.layout.custom_dialog,null);

            Button btn_okay = (Button)mView.findViewById(R.id.btn_okay);
            alert.setView(mView);
            @Override
            public void onClick(View v) {
                alertDialog.dismiss();
            }
        });
        alertDialog.show();
    }
}
@RequiresApi(api = Build.VERSION_CODES.KITKAT)
@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate (savedInstanceState);
    setContentView (R.layout.activity_sms);

    txt_msg = findViewById (R.id.txt_sms);

    Save.setOnClickListener (new View.OnClickListener () {
        @Override
        public void onClick(View v) {
            editor.putString ("phone2", phone2);
            txt_pnumber1.setText (phone1);
            txt_pnumber2.setText (phone2);
            txt_msg.setText (msg);
        }
    });
}

```

```

        if (!txt_pnumber1.getText ().toString ().equals ("") && !txt_pnumber2.getText ().toString ().equals
        ("") && !txt_pnumber3.getText ().toString ().equals ("") && !txt_pnumber4.getText ().toString ().equals
        ("")) {
            Toast.makeText (SmsActivity.this, "Saved...", Toast.LENGTH_SHORT).show ();

        } else
            Toast.makeText (SmsActivity.this, "Please enter the numbers first...",
            Toast.LENGTH_SHORT).show ();

    }
});
* instance pointing to the file that contains the values of preferences.
String Value = getShared.getString ("msg", "I am in danger, please come fast...");
txt_msg.setText (Value);
}
} else {
    Toast.makeText (this, "Please enter first number...", Toast.LENGTH_LONG);
}
int permissionCheck = ContextCompat.checkSelfPermission (this, Manifest.permission.SEND_SMS);
if (permissionCheck == PackageManager.PERMISSION_GRANTED) {

} else {

}

private void SendLocationMessage() {

    if (ActivityCompat.checkSelfPermission (this, Manifest.permission.ACCESS_FINE_LOCATION) !=
    PackageManager.PERMISSION_GRANTED && ActivityCompat.checkSelfPermission (this,
    Manifest.permission.ACCESS_COARSE_LOCATION) != PackageManager.PERMISSION_
    )
        fusedLocationProviderClient.getLastLocation ().addOnCompleteListener (new
    OnCompleteListener<Location> () {
        } else {
            String str="Software was not able to retrieve live location due to some internal errors..";
            Message += str;
        }
        String phoneNumber1 = txt_pnumber1.getText ().toString ().trim ();
        Toast.makeText (SmsActivity.this, "Please give the phone number first...",
        Toast.LENGTH_SHORT).show ();
    }

@Override
public void onRequestPermissionsResult(int requestCode, @NonNull String[] permissions, @NonNull
int[] grantResults) {

```

```

super.onRequestPermissionsResult (requestCode, permissions, grantResults);
switch (requestCode){
    case 0:

        break;
    }
}
static BroadcastReceiver broadcastReceiver = new BroadcastReceiver () {
    @Override
    public void onReceive(Context context, Intent intent) {
        // internet lost alert dialog method call from here...
    }
});
alertDialog.show();
default:
    return false;
}

```

Magnetometer.java :

```

package com.teamDroiders.ladybuddy;

import android.content.Context;
import android.content.Intent;
import android.content.SharedPreferences;
public class Magnetometer extends AppCompatActivity implements SensorEventListener {

    private TextView magR, show_conditions, x_cor, y_cor, z_cor ;
    SpeedometerView Speed;

    MediaPlayer mediaPlayer;
    private double magD;
    private Sensor magnetometer ;
    private SensorManager sensorManager; // read about SensorManger just by ttapping on where it is
    used.
    String prevStarted = "yesMagnet";
    @Override
    protected void onResume() {
        super.onResume();
        SharedPreferences sharedPreferences = getSharedPreferences(getString(R.string.app_name),
Context.MODE_PRIVATE);
        if (!sharedPreferences.getBoolean(prevStarted, false)) {
            SharedPreferences.Editor editor = sharedPreferences.edit();
            editor.putBoolean(prevStarted, Boolean.TRUE);
            editor.apply();
            alert.setView(mView);

```



```

        @Override
        public void onClick(View v) {
            alertDialog.dismiss();
        }
    });
    alertDialog.show();
}
}

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

    FloatingActionButton floatingActionButton=findViewById(R.id.magnetoInst);
    floatingActionButton.setOnClickListener (new View.OnClickListener () {
        @Override
        public void onClick(View v) {
            startActivity( new Intent ( getApplicationContext(), MagBtnInst.class ) );
        }
    });

    // configure value range and ticks
    Speed.setMaxSpeed(100);
    Speed.setMajorTickStep(10);
    Speed.setMinorTicks(0);

    magR = (TextView) findViewById(R.id.value);
    show_conditions = findViewById( R.id.show_conditions );
    x_cor = findViewById( R.id.x_cor );
    sensorManager = (SensorManager) getSystemService(Context.SENSOR_SERVICE);
);
    if (magnetometer != null){

sensorManager.registerListener(Magnetometer.this,magnetometer,SensorManager.SENSOR_DELAY_NORMAL);
    }else {
        magR.setText("Magnetometer not Supported");
    }

    @Override
    public void onSensorChanged(SensorEvent event) {

        Sensor sensor = event.sensor;

```

```

if (sensor.getType() == Sensor.TYPE_MAGNETIC_FIELD){
    double x;
    if(newx>0 && newx<100){
        Speed.setSpeed(newx, 1, 1);

    }else if(newx>=100){
        Speed.setSpeed (100,1,1);
    }

    BigDecimal bdx = new BigDecimal(event.values[0]).
        mediaPlayer = MediaPlayer.create(this, R.raw.beep);
        mediaPlayer.start();

@Override
public void onAccuracyChanged(Sensor sensor, int accuracy) {

}

}

```

AboutUs.java :

```

package com.teamDroiders.ladybuddy;

import androidx.appcompat.app.AppCompatActivity;
import android.content.ActivityNotFoundException;
import android.content.Intent;
import android.net.Uri;

public class AboutUs extends AppCompatActivity {

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate (savedInstanceState);
        setContentView (R.layout.activity_about_us);
        Button button=findViewById (R.id.contactUs);
        button.setOnClickListener (new View.OnClickListener () {
            @Override
            public void onClick(View v) {
                composeEmail();
            }
        });
    }

    public void composeEmail() {
        String subject="Contacting for LadyBuddy";
    }
}

```

```

String mailto = "mailto:newttonhabakuk.s@gmail.com" +
    "?cc="+ "nivethithashanmuganathan5@gmail.com"+
    "?cc="+ "naveenkmr.0402@gmail.com"+
    "&bcc="+ "bpooja37202@gmail.com"+
    "&subject=" + Uri.encode(subject);

Intent emailIntent = new Intent(Intent.ACTION_SENDTO);
emailIntent.setData(Uri.parse(mailto));

try {
    startActivity(emailIntent);
} catch (ActivityNotFoundException e) {
    //TODO: Handle case where no email app is available
}

}
}

```

CurrentLocation.java :

```

package com.teamDroiders.ladybuddy;

import android.Manifest;
import android.content.Intent;
import android.content.pm.PackageManager;
import android.location.Location;
import com.google.android.gms.location.LocationServices;

import java.util.Locale;

public class CurrentLocation extends AppCompatActivity {

    protected Location mLastLocation;

    private String mLatitudeLabel;
    private String mLongitudeLabel;

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

        mLatitudeLabel = "latitude";
        mLongitudeLabel = "longitude";
        mLatitudeText = (TextView) findViewById ((R.id.latitude_text));
        mLongitudeText = (TextView) findViewById ((R.id.longitude_text));

        mFusedLocationClient = LocationServices.getFusedLocationProviderClient (this);
    }
}

```

```

    }

    @Override
    public void onStart() {
        super.onStart ();

        if (!checkPermissions ()) {
            requestPermissions ();
        } else {
            getLastLocation ();
        }
    }

    @Override
    public void onRequestPermissionsResult(int requestCode, String[] permissions,
                                           int[] grantResults) {
        super.onRequestPermissionsResult (requestCode, permissions, grantResults);
        Log.i (TAG, "onRequestPermissionsResult");
        if (requestCode == REQUEST_PERMISSIONS_REQUEST_CODE) {
            Log.i (TAG, "User interaction was cancelled.");
        }
    }

    private boolean checkPermissions() {
        int permissionState = ActivityCompat.checkSelfPermission (this,
            Manifest.permission.ACCESS_FINE_LOCATION);
        return permissionState == PackageManager.PERMISSION_GRANTED;
    }

    private void requestPermissions() {
        new View.OnClickListener () {
            @Override
            public void onClick(View view) {
                // Request permission
            }
        };
    }

    private void getLastLocation() {
        if (ActivityCompat.checkSelfPermission (this, Manifest.permission.ACCESS_FINE_LOCATION) !=
            PackageManager.PERMISSION_GRANTED && ActivityCompat.checkSelfPermission (this,
            Manifest.permission.ACCESS_FINE_LOCATION) != PackageManager.PERMISSION_GRANTED) {
            return;
        }
        mFusedLocationClient.getLastLocation ()
            .addOnCompleteListener (this, new OnCompleteListener<Location> () {
                @Override
                public void onComplete( Task<Location> task) {
                    if (task.isSuccessful () && task.getResult () != null) {
                        mLatitudeText.setText (String.format (Locale.ENGLISH, "%s: %f",
                            mLatitudeLabel,
                            mLastLocation.getLatitude ()));
                    }
                }
            });
    }
}

```

10.2 SCREENSHOTS

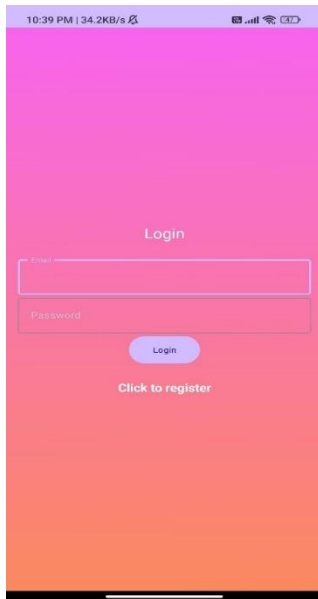


Figure 10.2.1 Login Screen

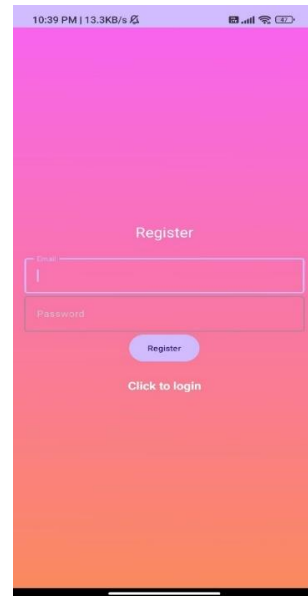


Figure 10.2.2 Registration Screen



Figure 10.2.3 Home Screen

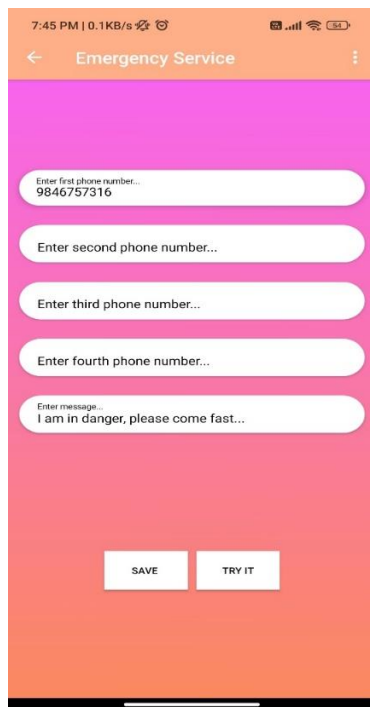


Figure 10.2.4 Emergency Service Screen



Figure 10.2.5 Women's News Screen

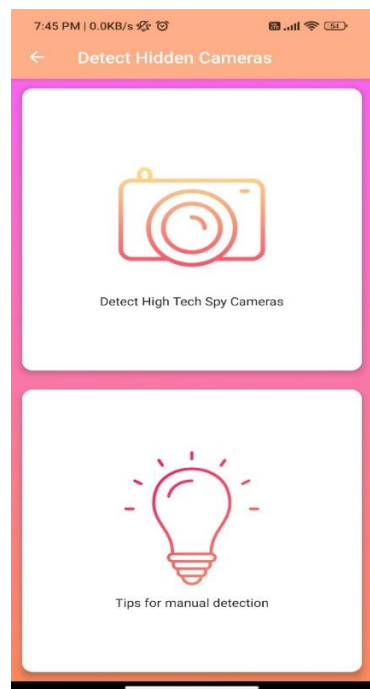


Figure 10.2.6 Hidden Camera Detection Home Screen

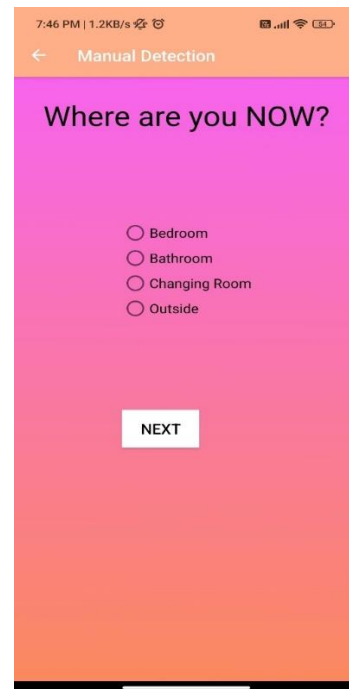


Figure 10.2.7 Manual Detection Choose Screen

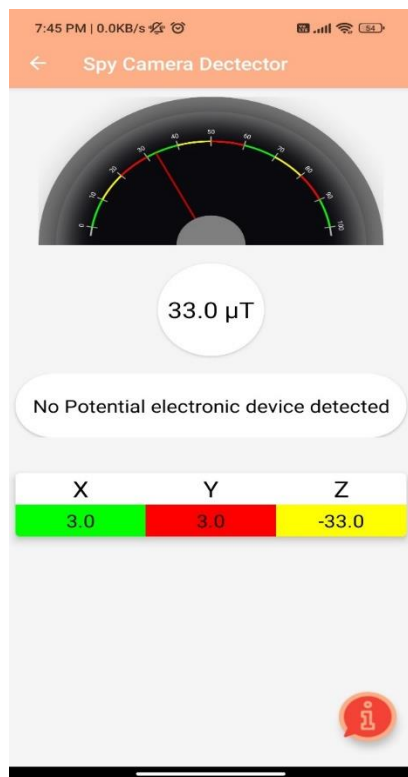


Figure 10.2.8 Hidden Camera Detection Screen

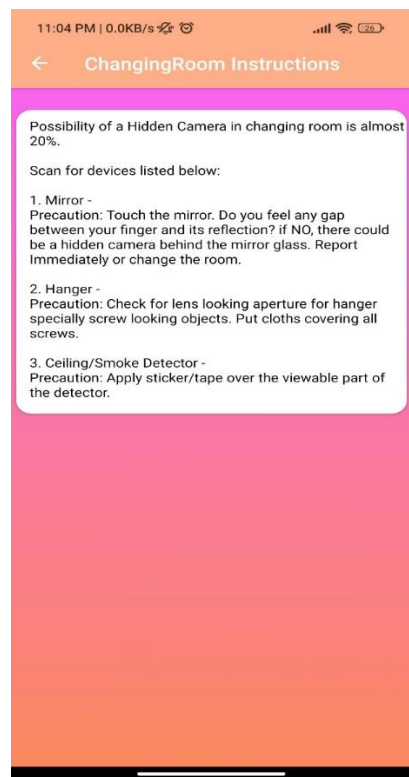


Figure 10.2.9 Manual Detection Instruction Screen



Figure 10.2.10 Emergency Siren Alert Screen

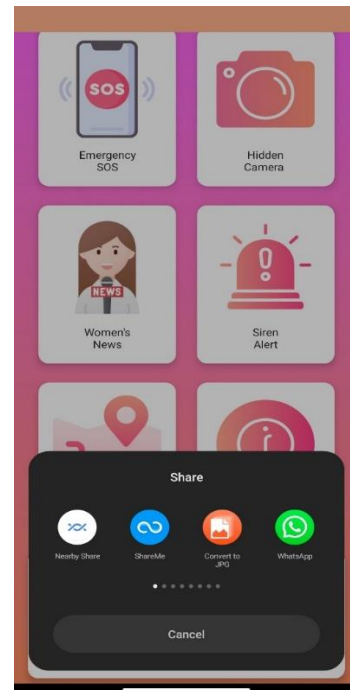


Figure 10.2.11 App Share Screen

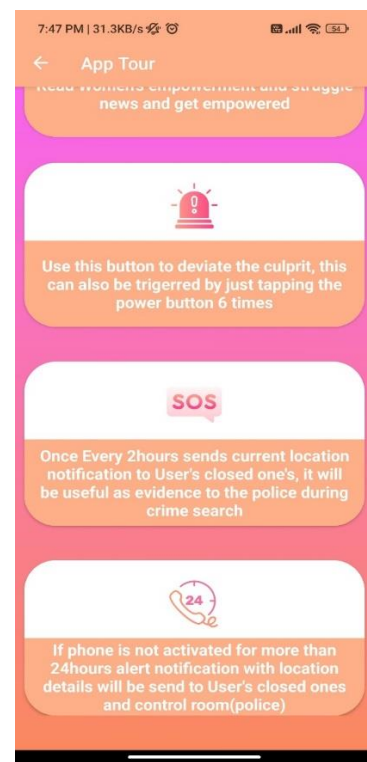


Figure 10.2.12 App Tour Screen

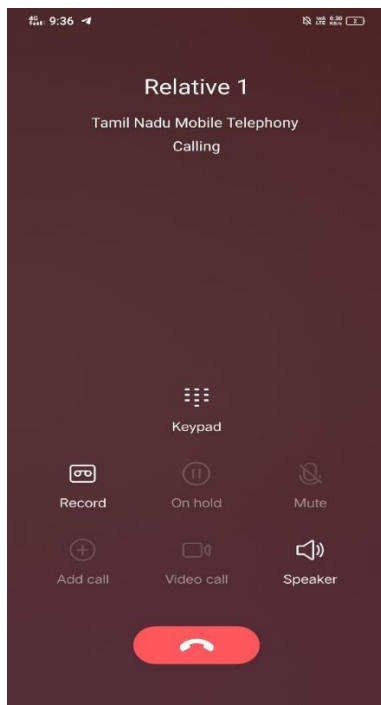


Figure 10.2.13 Emergency SOS Calling

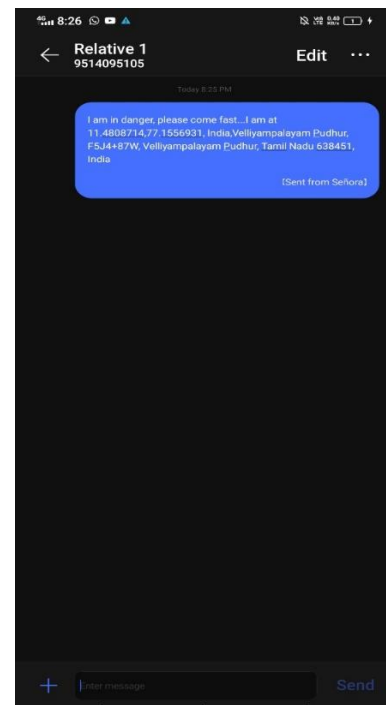
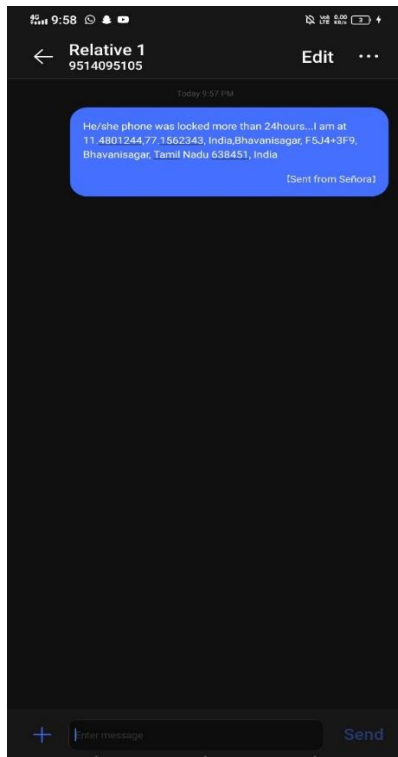
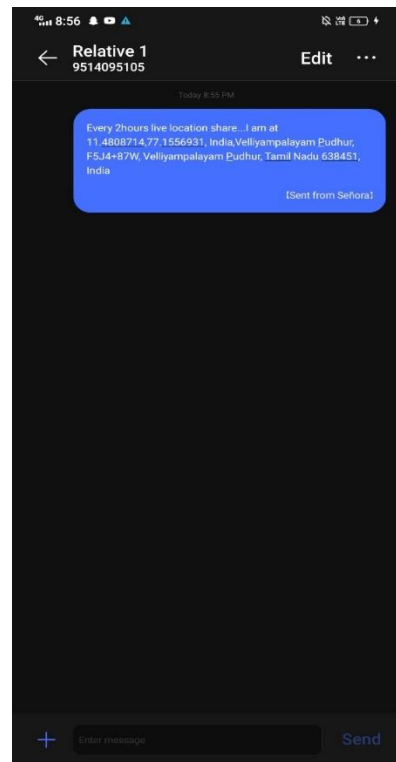


Figure 10.2.14 Emergency SOS Message and location Share



**Figure 10.2.15 24 Hours Phone locked
Alert Message with location**



**Figure 10.2.16 Every 2 hours location
details sharing**

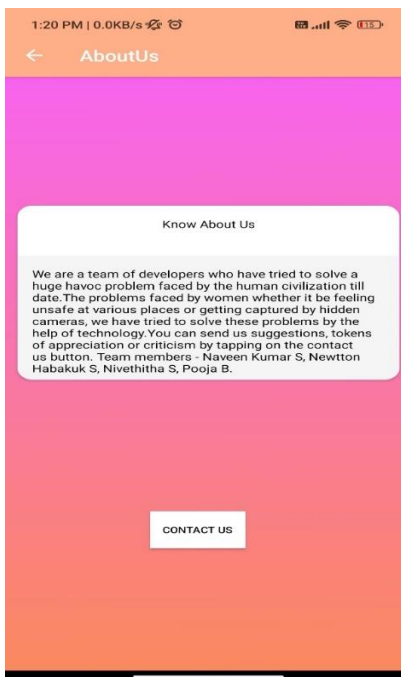


Figure 10.2.17 About Us Screen

Authentication

[Users](#) [Sign-in method](#) [Templates](#) [Usage](#) [Settings](#) | [Extensions](#) NEW

Search by email address, phone number, or user UID					Add user	↺	⋮
Identifier	Providers	Created ↓	Signed In	User UID			
pooja@gmail.com	✉	May 14, 2023		9cl67h6LHdYpWNiczjeEFVKqgNG3			
newtton@gmail.com	✉	May 12, 2023	May 12, 2023	5YWYGSOhd8TVUGluoxUIQexcNg...			
newtesting@gmail.com	✉	May 12, 2023	May 12, 2023	ALwReCmCY6SEf2cKyuTrPwzalzI3			
naveen@gmail.com	✉	May 12, 2023	May 12, 2023	mHkBmmAI05h5ucOLO8qdxinxuE...			
nivethithanivethitha360@g...	✉	May 12, 2023	May 12, 2023	wTcA1kmiPuMwzis2a5IWB3UcTf...			

Figure 10.2.18. Database for user details

CHAPTER 11

REFERENCES

- [1] S. Vahini, N. Vijaykumar, Efficient tracking for women safety and security using IoT, International Journal of Advanced Research in Computer Science, Volume 8, No.,9, November-December 2018.
- [2] S. Sangeetha, P. Radhika, "Application for Women Safety", IOSR Journal of Computer Engineering (IOSRJCE), Volume 17, Issue 3, Ver. IV (May – Jun. 2016), PP01-04.
- [3] TruptiRajendraShimpi, "Tracking and Security System for Women's using GPS & GSM", International Research Journal of Engineering and Technology (IRJET), Volume: 04 Issue:07 | July-2017.
- [4] RashmiDeodhe, "Woman Security System By UsingGps & Gsm", International journal for engineering applications and Technology, ISSN:2321-8134.
- [5] Sharma, S., Ayaz, F., Sharma, R., Jain, D., & Student, B. E. (2019). IoT Based Women Safety Device using ARM7.
- [6] K. Madhavi and S. Suresh Kumar. (2020) "Women Safety App: A Review of Existing Approaches and Future Directions"
- [7] M. R. Abitha and S. V. Shilpa. (2021)"Design and Development of a Mobile App for Women Safety Using IoT"
- [8] N. Rai and N. Rastogi. (2020)"A Review of Women's Safety Mobile Applications"
- [9] A. Jain, A. Khampariya, and N. Jain. (2021) "Smartphone-based Women Safety Application using Machine Learning Algorithm"
- [10] S. Kumari and S. Sivaramakrishnan.(2019)"Enhancing Women's Safety in Public Spaces with a Mobile Application: A Case Study from New Delhi, India"