RESPONSIVE MEDICATION REMINDER WITH GUARDIAN ALERT

A DESIGN PROJECT REPORT

submitted by

SHANMATHI S

UMA MAHESWARI K

VEDHA VARSHINI V

in partial fulfillment for the award of the degree

of

BACHELOR OF ENGINEERING

in

COMPUTER SCIENCE AND ENGINEERING

K RAMAKRISHNAN COLLEGE OF TECHNOLOGY

(An Autonomous Institution, affiliated to Anna University Chennai, Approved by AICTE, New Delhi)

Samayapuram – 621 112

JUNE 2025

RESPONSIVE MEDICATION REMINDER WITH GUARDIAN ALERT

A DESIGN PROJECT REPORT

submitted by

SHANMATHI S (811722104141)

UMA MAHESWARI K (811722104170)

VEDHA VARSHINI V (811722104178)

in partial fulfillment for the award of the degree

of

BACHELOR OF ENGINEERING

in

COMPUTER SCIENCE AND ENGINEERING

K RAMAKRISHNAN COLLEGE OF TECHNOLOGY

(An Autonomous Institution, affiliated to Anna University Chennai, Approved by AICTE, New Delhi)

Samayapuram – 621 112

JUNE 2025

K RAMAKRISHNAN COLLEGE OF TECHNOLOGY

(AUTONOMOUS)

SAMAYAPURAM – 621 112

BONAFIDE CERTIFICATE

Certified that this project report titled **"RESPONSIVE MEDICATION REMINDER WITH GUARDIAN ALERT"** is Bonafide work of **SHANMATHI S (811722104141), UMA MAHESWARI K (811722104170), VEDHA VARSHINI V (811722104178)** who carried out the project under my supervision. Certified further, that to the best of my knowledge the work reported here in does not form part of any other project report or dissertation on the basis of which a degree or award was conferred on an earlier occasion on this or any other candidate.

SIGNATURE

Dr. A. Delphin Carolina Rani, M.E., Ph.D.,

HEAD OF THE DEPARTMENT

PROFESSOR

Department of CSE

K Ramakrishnan College of Technology

(Autonomous)

Samayapuram – 621 112

SIGNATURE

Mrs. R. Jasmine, M.E.,

SUPERVISOR

ASSISTANT PROFESSOR

Department of CSE

K Ramakrishnan College of Technology

(Autonomous)

Samayapuram – 621 112

Submitted for the viva-voice examination held on ……

INTERNAL EXAMINER **EXTERNAL EXAMINER**

DECLARATION

We jointly declare that the project report on **"RESPONSIVE MEDICATION REMINDER WITH GUARDIAN ALERT"** is the result of original work done by us and best of our knowledge, similar work has not been submitted to **"ANNA UNIVERSITY CHENNAI"** for the requirement of Degree of Bachelor of Engineering. This project report is submitted on the partial fulfillment of the requirement of the awardof Degree of Bachelor of Engineering.

Signature

_____
SHANMATHI S

_____
UMA MAHESWARI K

_____
VEDHA VARSHINI V

Place: Samayapuram

Date:

ACKNOWLEDGEMENT

It is with great pride that we express our gratitude and indebtedness to our institution **"K RAMAKRISHNAN COLLEGE OF TECHNOLOGY"**, for providing us with the opportunity to do this project.

We are glad to credit and praise our honorable and respected chairman sir **Dr. K. RAMAKRISHNAN, B.E.,** for having provided for the facilities during the course of our study in college.

We would like to express our sincere thanks to our beloved Executive Director **Dr. S. KUPPUSAMY, MBA, Ph.D.,** for forwarding our project and offering adequate duration to complete it.

We would like to thank **Dr. N. VASUDEVAN, M.Tech., Ph.D.,** Principal, who gave opportunity to frame the project with full satisfaction.

We heartily thank **Dr. A. DELPHIN CAROLINA RANI, M.E., Ph.D.,** Head of the Department, **COMPUTER SCIENCE AND ENGINEERING** for providing her support to pursue this project.

We express our deep and sincere gratitude and thanks to our project guide **Mrs. R. JASMINE, M.E.,** Department of **COMPUTER SCIENCE AND ENGINEERING,** for her incalculable suggestions, creativity, assistance and patience which motivated us to carry this project.

We render our sincere thanks to Course Coordinator and other staff members for providing valuable information during the course. We wish to express our special thanks to the officials and Lab Technicians of our departments who rendered their help during the period of the work progress.

ABSTRACT

The project titled "Responsive Medication Reminder with Guardian Alert" is a medical-assistance Android application designed to ensure timely medication intake and provide emergency support for individuals with health vulnerabilities. The primary goal of this system is to automate medication reminders while also ensuring that necessary help is reached if a user fails to respond to the alert. The application allows the user to input key information such as their name, alarm time (medication time), and the phone number of a guardian or emergency contact. At the scheduled time, the alarm is triggered with sound and/or vibration. If the user does not respond or dismiss the alarm within a predefined interval, the application automatically initiates a phone call to the registered guardian, ensuring they are promptly informed the system utilizes core Android components such as Alarm Manager, Broadcast Receiver, and Intent services to manage time-based triggers and call functionality. It also incorporates runtime permission handling for CALL_PHONE to ensure smooth execution of emergency calls. This feature is particularly critical for elderly users, patients with memory disorders, or those recovering from medical treatments. By combining automated medication reminders with an emergency alert mechanism, this application offers a practical, life-enhancing tool that supports patient independence while maintaining a safety net through guardian involvement. It promotes health adherence and provides peace of mind for both users and their caregivers.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF ABBREVIATIONS

ABBREVIATION	FULL FORM
UMS	Universal Medication Schedule
CHD	Coronary Heart Disease
VM	Virtual Machine
API	Application Programming Interface
GUI	Graphical User Interface
SDK	Software Development Kit
IDE	Integrated Development Environment
NDK	Native Development Kit
AUD	Android Virtual Device
ADB	Android Debug Bridge

CHAPTER 1

INTRODUCTION

1.1 BACKGROUND

In today's world, managing healthcare-especially for the elderly, chronically ill, or physically challenged individuals-requires reliable systems to ensure timely medication intake and quick response in emergencies. Many such individuals live alone or without constant supervision, making them prone to missing doses or being unable to reach out for help during a health crisis. Missed medications can lead to severe complications, increased hospitalizations, and in extreme cases, life-threatening situations. While traditional methods such as pill organizers or manual logs exist, they rely heavily on user memory and self-discipline. With the widespread use of smartphones and digital tools, there is a great opportunity to provide a reliable, automated solution that not only reminds users to take their medications but also alerts a caregiver if the user is unresponsive. This project aims to address this healthcare gap through a mobile-based reminder and emergency alert system.

Traditional methods such as pill organizers, wall calendars, or manual logs have long been used to manage medication schedules. However, these methods are inherently dependent on the individual's memory and discipline, which can be unreliable—particularly for those suffering from cognitive decline, memory disorders, or mental health conditions. Moreover, these tools do not provide any form of contingency or backup in case the individual fails to act, nor do they notify others in the event of a problem.

In recent years, the proliferation of smartphones and wearable technologies has opened up new avenues in digital health and telemedicine. These technologies offer the potential to create smart, automated systems that combine reminders, health monitoring, and emergency alert functionalities

1.2 OVERVIEW

The Responsive Medication Reminder with Guardian Alert is an Android based mobile application designed to help users adhere to their medication schedule while ensuring that emergency help is notified in case of non-responsiveness. The app allows users to enter essential details such as their name, the time they need to take medication, and a guardian's phone number. When the scheduled time arrives, the app triggers an alarm accompanied by a sound or vibration.

If the user fails to interact with the alert within a specific duration, the application automatically initiates a phone call to the guardian, alerting them of a potential issue. The system uses Android's Alarm Manager to schedule alarms, Broadcast Receiver to handle alarm triggers, and Intent. ACTION_CALL to make emergency calls.

1.3 PROBLEM STATEMENT

Many individuals who require regular medication suffer from forgetfulness, poor routine management, or physical/mental limitations that prevent them from maintaining their health schedule effectively. This issue is particularly prevalent among the elderly and patients with chronic conditions who are often unsupervised. Traditional mobile reminder apps only alert the user without any follow-up mechanism if the user does not respond.

In scenarios where the user is incapacitated or unconscious, these apps fail to notify anyone, increasing the risk of serious medical outcomes. There is an urgent need for a solution that not only reminds users to take their medication but also responds appropriately when the user is unresponsive, thereby ensuring that help can be reached in time. The lack of such dual-functionality systems in current healthcare applications highlights a critical gap in mobile health (mHealth) solutions.

1.4 OBJECTIVE

The main objective of this project is to design and develop a reliable and easy- to use Android application that serves both as a medication reminder and an emergency alert system. The application aims to allow users to input their name, medication time, and an emergency contact number. It will trigger an alarm at the scheduled time and monitor whether the user acknowledges it. If the user does not respond within a predefined timeframe, the system will automatically initiate a phone call to the registered guardian. Additional objectives include ensuring a smooth user interface, secure data storage, and compliance with Android permissions for placing calls.

The app should be able to function reliably in the background and during phone idle states. Future enhancements may include SMS backup alerts, voice-based alarm acknowledgments, and location sharing. The overall goal is to build a system that enhances medication adherence, increases user safety, and offers peace of mind to caregivers.

The core functionality of the application includes allowing users to Enter personal details such as their name, Schedule one or more medication times., Register a guardian or emergency contact number. At the scheduled time, the app will trigger an audible alarm to remind the user to take their medication. If the alarm is not acknowledged within a predefined response window (e.g., one to five minutes), the system will automatically initiate a phone call to the registered emergency contact, ensuring that a caregiver is notified of potential non-responsiveness. This feature aims to provide a critical layer of safety in situations where the user might be incapacitated or unaware.

1.5 IMPLICATION

The successful implementation of this application can significantly improve the health outcomes of individuals who require constant medication or live with medical risks. By ensuring timely reminders and triggering alerts during emergencies, the application reduces the likelihood of missed medication and unattended health events. It supports independent living for elderly users, reducing the need for constant caregiver supervision while still ensuring emergency readiness. For caregivers and family members, it offers peace of mind knowing they will be notified in case of non-response.

On a broader scale, the application promotes better healthcare management, helps prevent hospitalizations caused by missed medications, and showcases how technology can be effectively used in the healthcare sector. It is scalable for use in homes, hospitals, and elder care facilities. This solution also encourages further integration of mobile applications with smart health monitoring systems and could lay the foundation for future innovations in digital health and emergency support technologies.

CHAPTER 2

LITERATURE SURVEY

1. Interfaces of Medication Reminder Applications: An Analysis Aimed at the Elder Age, Jaqueline Donin Noleto, Vitor Jos´e Costa, Rodrigue, Rhenan Castelo Branco Cirilo, Francisco Ribeiro dos Santos, 2017

Despite how technologies are, nowadays, becoming integrated to several branches of society, there are still many groups with trouble to follow these technological advancements. Among them, the group that most likely suffers from this difficulty are the elders. An area strongly benefited by technology is health, currently counting with multiple mobile applications for smartphones, designed to upkeep the users' health. In this context, the current paper proposes an analysis of interfaces from medication reminder apps focused on the elderly.

The following techniques were implemented: An Usability Test through Exploration (UTE) to evaluate easiness in learning by exploration, the System Usability Scale (SUS) method to ascertain the usability level of said applications and the Heuristic Evaluation of their Interfaces. With those techniques, it was possible to build a solid benchmark to evaluate the level of elder friendliness of an app in the given context. Results from the case study have shown that the apps lack a design focused on the elder public, which might have led to troubles during the tests performed with volunteers. The expert-based results have also found general interface issues on these apps that may need to be improved.

The heuristic analysis of the apps' interfaces was performed by three usability experts, each evaluating all three apps. The application of Nielsen's heuristics occurred through expert-based usability walkthrough of pre-selected system features, from which the evaluators deemed which elements were out of usability standards.

2. Developing the Medication Reminder Mobile Application "Seeb", Asghar Ehteshami, Sakineh Saghaeiannejad, Ebtesam Savari, 2017

Today, the structure of comprehensive health care emphasizes self-care more than therapy. Medication therapy is a strong instrument for therapy received through the health setting, especially in medication area. Error in medication administration has produced different problems and they cost billions of dollars every year.

Regarding mobile phone extensions, we developed a local medication reminder mobile application called "Seeb" as a suitable solution for decreasing medication errors for Iranians. Methods: We conducted a mixed methods study in three Phases: Comparative study of existing mobile applications; developed its object-oriented model; Developed the initial version of "Seeb" that was approved for production. Results: This application was designed for the appropriate medication administration including time and dosages through: recording patient and medication data; scheduling patients' medication; and reporting medication administration on progress.

"Seeb" has been designed in compliance with Iranian health information technologists and pharmacists' requirements. It is expected to reduce medication error and improve patient adherence to medical prescriptions Discharged patients from hospitals need more medication order compliance; creating the essential conditions for reviewing the patient Medi-cation administration orders after dis-charge decreases the hospitalization and mortality, and increases patient quality of life. One of the factors that are known as the medical errors is lack of patient adherence to physician orders. Five percent of hospitals' admissions are because of medication errors that patients themselves commit.

3. Bilingual Medication (Med-Alert) Reminder Application for Patients, William Tichaona Vambe, Rebohele Sehlabo, 2019

Chronic diseases such as diabetes, cancer, HIV/AIDS, and tuberculosis pose significant health challenges. Often, patients have difficulty adhering to prescribed treatment plans due to busy schedules and too much medication leading to forgetting. Intrinsically, to address this problem of forgetting, this research developed a medication reminder application. The application had both English and Sotho language. Thus, it bridges the gap between medication management and technology and offers a promising solution to a pervasive healthcare challenge for those who do not understand English.

Design science research methodology guided the development and implementation of the application. Preliminary findings affirm the feasibility of creating a secure application, incorporating local languages, and accommodating multiple medication reminders. The study lays the foundation for a practical solution at the intersection of medication management and technology, with recommendations emphasizing user education, regular updates, and consultation with healthcare professionals to enhance functionality and safety.

4. An Automated and Online-Based Medicine Reminder and Dispenser, Shayla Sharmin, Md. Ibrahim Khulil Ullah Ratan, Ashraful Haque Piash,2019

It is important to take the right medications at the right times and in the right amounts. Patients, on the other hand, often fail to take their medications at the times specified in their prescriptions, causing disease or illness to develop more slowly, especially in the elderly or those who are too preoccupied with their job. An automated and online-based medicine reminder and dispenser application is introduced in this paper. A three-part package, an LCD on top of the box, a buzzer, and a multi-coloured LED light were all included in this unit.

This interface also reminds the consumer when it is time to take their medicine. An Android application on this device displays some of the results. The input interface and the output interface are the two components of this mobile app interface. Prescriptions are accepted or modified via the input interface.

5. Construction and application of medication reminder system: intelligent generation of universal medication schedule, Hangxing Huang, Lu Zhang, Yongyu Yang, Ling Huang, Xikui Lu, Jingyang Li, Huimin Yu, Shuqiao Cheng, Jian Xiao, 2021

Background Patients with chronic conditions need multiple medications daily to manage their condition. However, most patients have poor compliance, which affects the effectiveness of treatment. To address these challenges, we establish a medication reminder system for the intelligent generation of universal medication schedule (UMS) to remind patients with chronic diseases to take medication accurately and to improve safety of home medication.

Methods To design medication time constraint with one drug (MTCOD) for each drug and medication time constraint with multi-drug (MTCMD) for each two drugs in order to better regulate the interval and time of patients' medication. Establishment of a medication reminder system consisting of a cloud database of drug information, an operator terminal for medical staff and a patient terminal.

6. Medication reminder APPs to improve medication adherence in coronary heart disease (MedApp-CHD) Study: a randomised controlled trial protocol, Karla Santo, Clara K Chow, Aravinda Thiagalingam.,2022

In The growing number of smartphone health applications available in the app stores makes these apps a promising tool to help reduce the global problem of non-adherence to long-term medications. However, to date, there is limited evidence that available medication reminder apps are effective.

This study aims to determine the impact of medication reminder apps on adherence to cardiovascular medication when compared with usual care for people with coronary heart disease (CHD)and to determine whether an advanced app compared with a basic app is associated with higher adherence. Methods and analysis Randomized controlled trial with follow-up at 3 months to evaluate the feasibility and effectiveness of medication reminder apps on medication adherence compared with usual care.

7. A case study of medication reminder system, Mohammad Alhaj, Abdullah Nizar, Shahd Al-Hatem, Athraa Leekha,2022

Medication management is medical treatment system that monitors the medication therapy of a patient to confirm that the patient is complying with a medication regimen. It also ensures that the patient is avoiding potentially dangerous drug interactions and other complications. This is important for patients taking large numbers of medications to address chronic illnesses and multiple diseases.

Taking numerous medications is known as polypharmacy and it is particularly common among older adults, as they are more likely to need medications to manage an array of chronic conditions. There are a number of principles to medication management, all of which are focused on making sure that medications are used appropriately and keeping track of all of the medications currently in use by patients.

8. Medicine reminder application, Kiran borkar, Siddhant kakade, Aayush shahi, Aditya Rathod, 2024

The Medicine Reminder Application is designed to help users manage their medication schedules, medical appointments, and health-related activities efficiently. This mobile application provides an intuitive interface to schedule and track medication intake, set refill alerts, and receive timely reminders for doctor appointments.

With features like customizable reminders, snooze options, and detailed medication logs, the application ensures users never miss a critical dose or appointment. The app uses local notifications, ensuring that reminders are delivered reliably, even when the device is offline.

9. Prevalence and correlates of medication reminder app 'use and use intention' among older adults, Yongjing Ping, Abhijit Visaria b, Sumithra Devi Suppiah,2024

Adherence to prescribed pharmacotherapy regimens is crucial to maintain the clinical effectiveness of prescriptions in chronic conditions. High medication adherence contributes to improved health outcomes, decreased medication and hospitalisation costs, and fewer emergency visits among patients with chronic diseases.

While optimal therapeutic efficacy requires high medication adherence rates, such as 80% and higher in the case of hypertension and cardiovascular diseases, adherence to chronic medications is reported to be only around 50%. Older adults, aged 60 years and above, are especially at risk of medication non-adherence; in Singapore, 60% of older adults with at least one chronic illness had medication non-adherence.

10. GoMed: Daily Medicine Reminder Application, Afshaan Sarguroh, Munaf Shaikh, Kashif Khan, Dr. Zainab Mirza4, 2024

Good Health is a secret of every human being. Since the very beginning, Health is a matter of great concern. Some have to, without a choice, maintain a healthy lifestyle by taking medicines. Many patients find it difficult to take medicines at a proper time due to various reasons such as forgetfulness, busy schedule, old age, etc. This happens most commonly with the people taking medicines daily which results in medical non-adherence. Medical non-adherence is a very serious issue as it can lead to various health-related problems.

The advancement in mobile technology has enabled various techniques to solve these types of problems by designing and developing an application which patient will find it easy to carry along. In this paper, we aim to build an Android-based application, that will cover major features such as Medicine Reminder, Medicine Restocking Alert, Searching Hospitals, Medicals, Doctors in the vicinity, BMICalculator, Notes, Caretaker Information, First Aid/Health Tips, Alarm System, etc.

CHAPTER 3

SYSTEM ANALYSIS

3.1 EXISTING SYSTEM

In the existing system, most medication management solutions are either manual or limited to basic reminder functions. Commonly used tools include physical pillboxes, traditional alarm clocks, or simple mobile reminder applications that notify the user when it's time to take their medication. These systems rely entirely on user acknowledgment and do not take any action if the user fails to respond to the reminder. Mobile apps in this category may send notifications or alarms but lack an emergency handling mechanism.

As a result, if a user is unconscious, unwell, or physically unable to respond, there is no follow-up alert or communication to any caretaker or guardian. This limitation makes these systems unreliable in critical or emergency situations, especially for elderly individuals or patients with chronic conditions who require close monitoring. Furthermore, most existing applications do not offer integration between medication alerts and emergency communication, leading to potential delays in assistance during medical emergencies.

Physical Pill Organizers: Devices such as compartmentalized pillboxes labelled by day and time are widely used. However, they offer no digital integration, do not provide reminders, and completely rely on the user’s memory and discipline.

Traditional Alarm Clocks: Set manually to ring at designated times, these tools serve as basic reminders but lack any contextual awareness or interactivity. They cannot differentiate between a medication reminder and any other alert, and they do not require acknowledgment to confirm that the medication has been taken.

Basic Mobile Reminder Apps: These applications are slightly more advanced, allowing users to set alarms or push notifications for medication times. Some may even provide recurring alarms, snooze options, and scheduling flexibility.

3.2 PROPOSED SYSTEM

The Responsive Medication Reminder with Guardian Alert system aims to overcome the limitations of the existing methods by providing an intelligent and responsive solution. The proposed system not only reminds users to take their medication on time but also ensures that their well-being is monitored by incorporating an automated emergency alert feature.

The application allows users to input their name, schedule medication alarms, and enter a guardian or emergency contact number. When the alarm goes off, the user is prompted to acknowledge it. If the user fails to respond within a predetermined time frame, the app automatically initiates a phone call to the registered guardian, alerting them to a possible emergency situation.

Medication Reminder Setup: Users can input their personal details, including name and medication schedule, by specifying the time(s) they need to be reminded. The system supports flexible scheduling, allowing for multiple alarms per day and recurring reminders.

Alarm with Acknowledgment Prompt: When the scheduled time arrives, the application triggers a prominent alarm notification accompanied by a visual and/or auditory prompt requesting user acknowledgment. This ensures that the reminder captures the user's attention.

Automated Emergency Response: If the user fails to acknowledge the alarm within a predefined time frame (e.g., 1–5 minutes), the system automatically initiates a phone call to a registered guardian or emergency contact. This escalation ensures that assistance can be summoned even if the user is incapacitated or unable to interact with the device.

3.3 BLOCK DIAGRAM OF PROPOSED SYSTEM

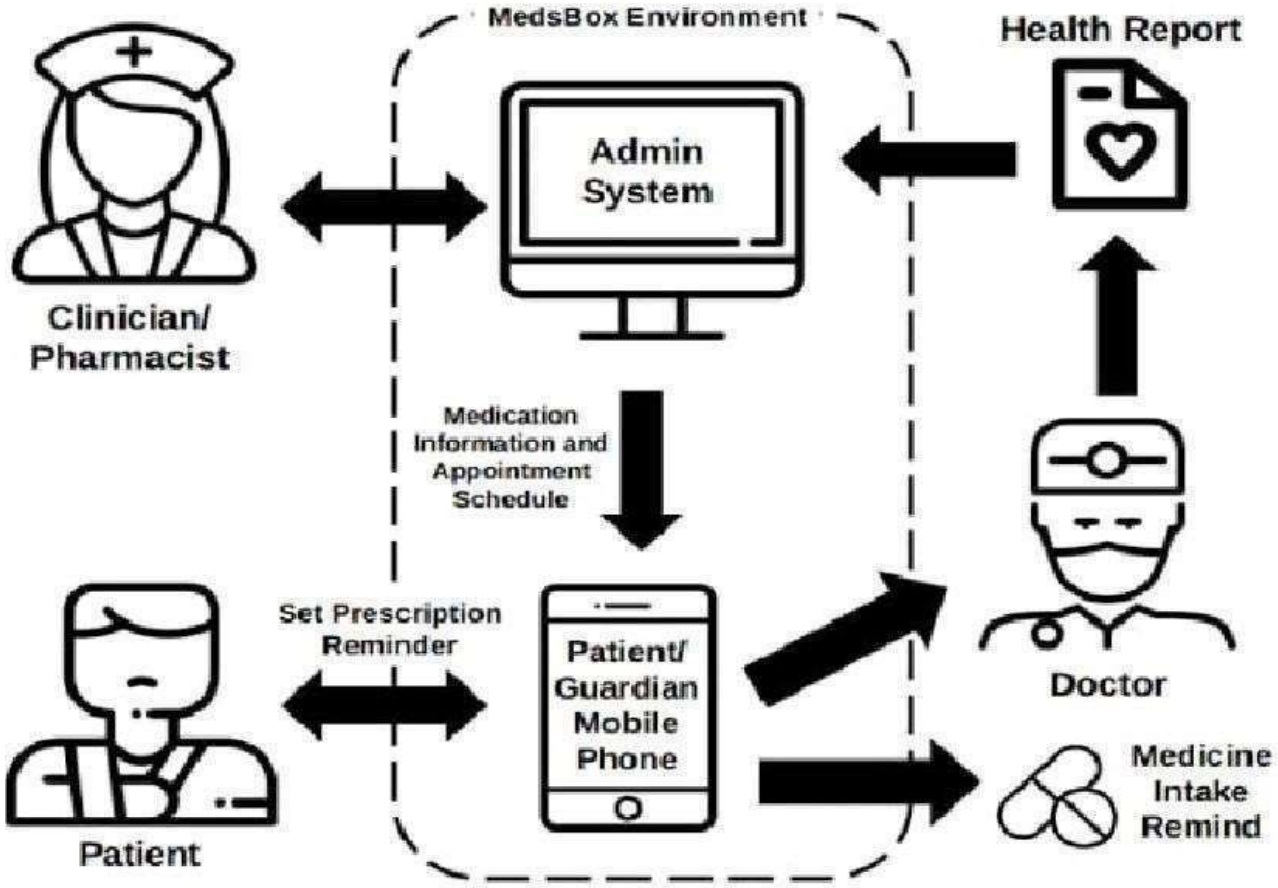

Fig 3.1: Proposed System

3.4 USE CASE DIAGRAM

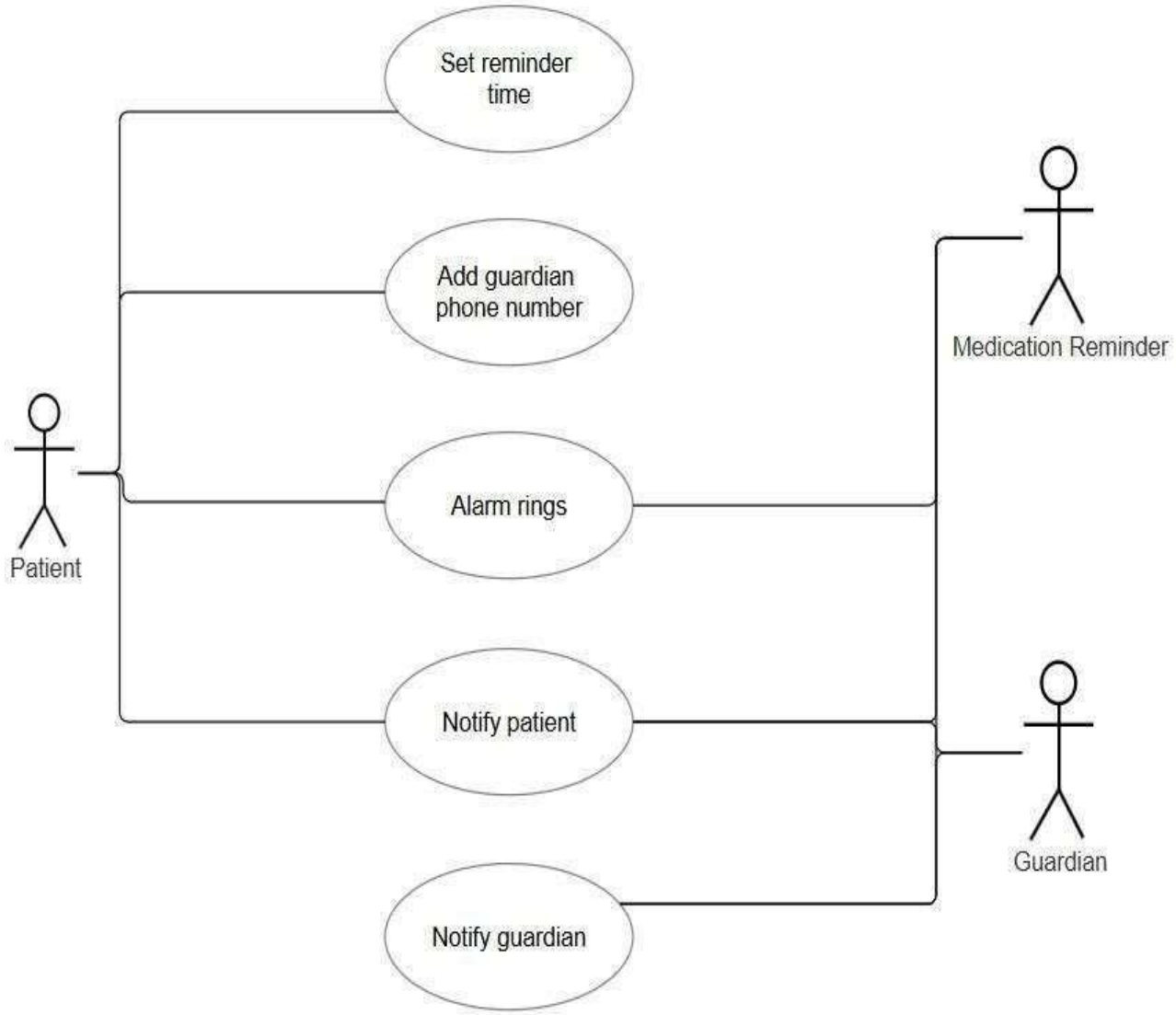

Figure 3.2: Use Case Diagram

3.6 ACTIVITY DIAGRAM

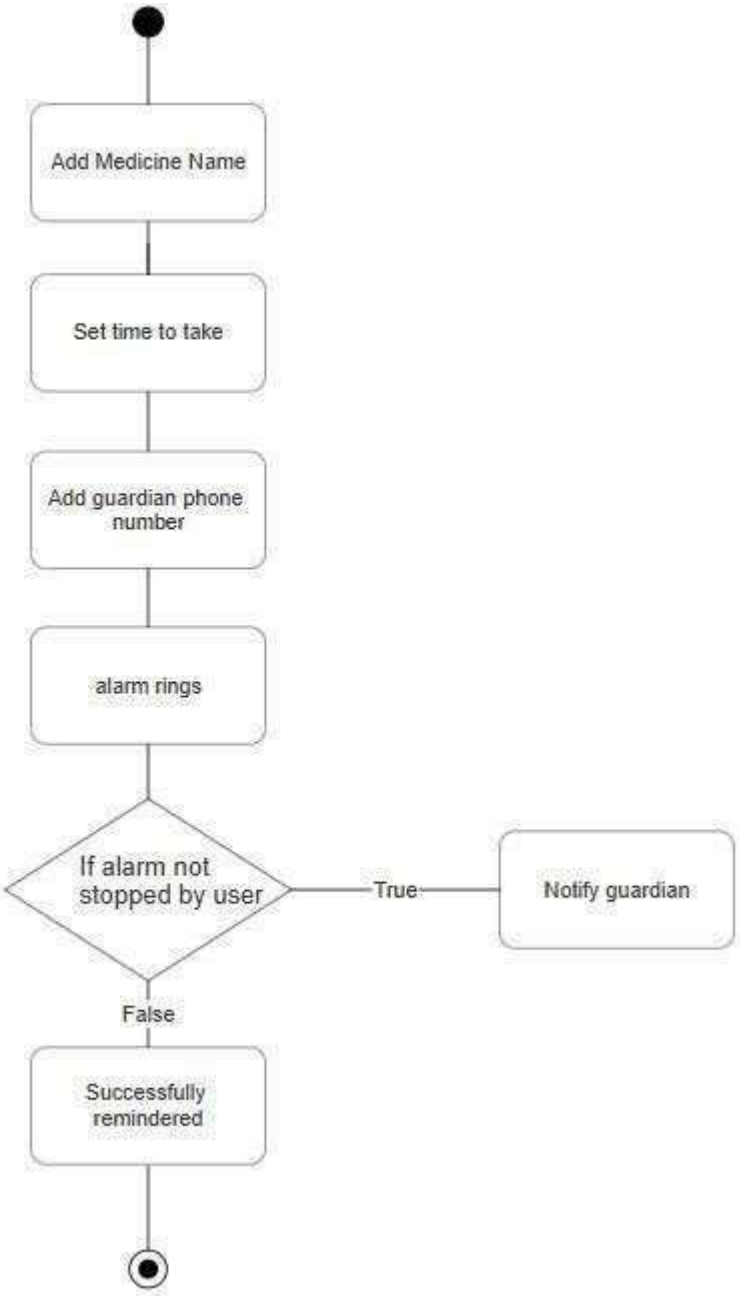

Figure 3.3: Activity Diagram

CHAPTER 4

MODULES

4.1 MODULE DESCRIPTION

- User Registration & Profile Management
- Alarm Configuration & Management
- Alarm Acknowledgment & Detection
- Alternative Contact Notification System
- System Integration & Security

4.1.1 User Registration & Profile Management

The User Registration & Profile Management module is a fundamental component of the “Responsive Medication Reminder with Guardian Alert” system. This module is responsible for capturing and securely storing essential user information required for both personalizing the app and ensuring the effectiveness of the emergency alert feature. When the user first launches the application, they are prompted to complete a simple registration process where they provide their full name, age, and most importantly, an emergency contact number-typically that of a guardian, caregiver, or family member.

This contact information plays a crucial role in the event that the user does not respond to a scheduled medication alarm, as it is used to initiate an automatic emergency call. The registration process is designed to be user-friendly, with a clean and intuitive interface suitable for all users, including elderly individuals who may have limited experience with smartphones. Once registered, the user’s profile is stored locally using Android's SQLite database or secure shared preferences, depending on implementation, ensuring the data is easily accessible by the application during runtime while maintaining privacy and data security.

4.1.2 Alarm Configuration & Management

The Alarm Configuration & Management module serves as the central operational feature of the "Responsive Medication Reminder with Guardian Alert" application. This module enables users to schedule and manage medication reminders in a flexible, user-friendly manner. Upon accessing this section, users can input specific details such as the time of medication, name of the medicine, dosage, and any custom notes. The interface is designed to be intuitive, making it easy for users of all ages-including elderly patients-to set one-time alarms or configure recurring schedules based on daily, weekly, or custom intervals.

Once the alarm is set, the application uses Android's Alarm Manager to trigger alarms accurately, even if the device is idle, in sleep mode, or locked. This ensures the user is always notified on time to take their medication, which is essential for effective treatment adherence and long-term health management.

4.1.3 Alarm Acknowledgment & Detection

The Alarm Acknowledgment & Detection module plays a critical role in ensuring user engagement and safety within the "Responsive Medication Reminder with Guardian Alert" system. After an alarm is triggered by the previously configured schedule, this module takes over by alerting the user through an audible alarm sound, vibration, and an interactive on-screen prompt. The prompt is designed to request immediate acknowledgment from the user, typically through a simple tap or button press indicating they have taken their medication.

A timer is displayed on the screen, usually ranging from 30 to 60 seconds, within which the user is expected to respond. This timeout mechanism serves as a safeguard to determine the user's awareness and physical ability to respond. If the user acknowledges the alarm within the given time frame, the system logs the acknowledgment and dismisses the alert, allowing the user to continue their routine normally. If the user fails to respond, the application interprets this as a potential emergency, initiating the next module responsible for emergency contact notification.

4.1.4 Alternative Contact Notification System

The Alternative Contact Notification System is a vital component of the Responsive Medication Reminder with Guardian Alert application, designed to respond to emergency situations where the user fails to acknowledge a scheduled medication alarm. If the user does not interact with the alarm within a predefined time frame, this module is triggered automatically to initiate emergency protocols.

Using Android's Intent.ACTION_CALL functionality, the application seamlessly places a real-time voice call to the guardian or emergency contact number entered during the user registration process. This ensures that a caregiver or family member is immediately informed of a potential medical issue such as unconsciousness, missed critical medication, or an accidental overdose.

The system operates autonomously without the need for manual input, reducing delays and increasing the chance of timely intervention. In addition to enhancing user safety, this feature provides significant peace of mind to families and caregivers, knowing that they will be notified if the user is unresponsive.

The module is designed with reliability in mind, operating in the background and during idle states of the phone. Future enhancements may include multiple contact escalation, SMS backup alerts, location sharing, and integration with wearable devices to further improve responsiveness. Overall, the Alternative Contact Notification System transforms the application from a simple reminder tool into a proactive emergency response solution, bridging the gap between personal health management and remote caregiver support.

4.1.5 System Integration & Security

The System Integration & Security module forms the backbone of the Responsive Medication Reminder with Guardian Alert application, ensuring that all core functionalities—from alarm scheduling to emergency alerting—operate as a cohesive and reliable system. This module is responsible for managing seamless interaction between various components such as user registration, medication alarm configuration, user acknowledgment tracking, and the alternative contact notification system. It guarantees that each module communicates efficiently and performs its task without conflict or failure. Android’s native tools and APIs, such as Alarm Manager, Broadcast Receiver, Foreground Services, and Content Provider, are strategically used to facilitate background execution, handle asynchronous operations, and manage secure data flow between different parts of the application.

From a security perspective, the module enforces runtime permissions for sensitive operations like placing phone calls and accessing device state, ensuring the application complies with Android's evolving privacy policies. All personal user data, including contact details and alarm schedules, are stored using encrypted shared preferences or secure databases like Room with SQL Cipher for enhanced data protection. Additionally, the system checks for network availability and device status before triggering actions, reducing the risk of errors during critical tasks.

The integration layer also incorporates exception handling and fallback mechanisms to ensure that even in the case of app crashes, missed alarms, or denied permissions, the system can attempt secondary actions (e.g., logging events or sending alternative notifications). The app is optimized to perform reliably in different device states—including Doze Mode or background-restricted scenarios—using foreground services where necessary.

CHAPTER 5

SOFTWARE DESCRIPTION

5.1 FRONT END Android

Overview:

Android (stylized as android) is a mobile operating system developed by Google, based on the Linux kernel and designed primarily for touch screen mobile devices such as smart phones and tablets. Android's user interface is mainly based on direct manipulation, using touch gestures that loosely correspond to real-world actions, such as swiping, tapping and pinching, to manipulate on-screen objects, along with a virtual keyboard for text input. In addition to touch screen devices, Google has further developed Android TV for televisions, Android Auto for cars, and Android Wear for wrist watches, each with a specialized user interface. Variants of Android are also used on notebooks, game consoles, digital cameras, and other electronics.

Android is a powerful Operating System supporting many applications in Smart Phones. These applications make life more comfortable and advanced for the users. Hardware's that support Android are mainly based on ARM architecture platform. Android comes with an Android market which is an online software store. It was developed by Google. It allows Android users to select, and download applications developed by third party developers and use them. There are around 2.0 lack+ games, application and widgets available on the market for users.

Android applications are written in java programming language. Android is available as open source for developers to develop applications which can be further used for selling in android market. There are around 200000 applications developed for android with over 3billion+ downloads. Android relies on Linux version 2.6 for core system services such as security, memory management, process management, network stack, and driver model. For software development, Android provides Android SDK (Software development kit)

5.2 BACKEND

Java Platform:

A platform is the hardware or software environment in which a program runs. The Java platform differs from most other platforms in that it's a software-only platform that runs on top of other, hardware-based platforms. Most other platforms are described as a combination of hardware and operating system. The Java platform has two components: The Java Virtual Machine (Java VM) and The Java Application Programming Interface (Java API) Java VM is the base for the Java platform and is ported onto various hardware-based platforms.

The Java API is a large collection of ready-made software components that provide many useful capabilities, such as graphical user interface (GUI) widgets. The Java API is grouped into libraries (packages) of related components. The following figure of Java Structure depicts a Java program, such as an application or applet, that's running on the Java platform. As the figure shows, the Java API and Virtual Machine insulates the Java program from hardware dependencies.

The Java platform stands out from traditional platforms because it is software-based and operates independently of underlying hardware. Unlike conventional platforms that tie applications to specific operating systems or hardware architectures, Java applications are compiled into bytecode, which is executed by the Java Virtual Machine (JVM). This architecture allows the same code to run on any device that has a JVM installed, ensuring high portability and flexibility.

5.3 TOOL

Android SDK –API 23

Android is an operating system based on Linux with a Java programming interface. The Android Software Development Kit (Android SDK) provides all necessary tools to develop Android applications. This includes a compiler, debugger and a device emulator, as well as its own virtual machine to run Android programs. Android is primarily developed by Google.

Android allows background processing, provides a rich user interface library, supports 2-Dand 3-D graphics using the OpenGL libraries, access to the file system and provides an embedded SQLite database. Android application consists of different components and can reuse components of other applications. This leads to the concept of a task in Android; an application a reuse other Android component to archive a task.

Android Development Tools

Android is a widely anticipated open-source operating system for mobile devices that provides a base operating system, an application middleware layer, a Java software development kit (SDK), and a collection of system applications. Android mobile application development is based on Java language codes, as it allows developers to write codes in the Java language as illustrated in the below architecture figure of Android structure.

The Android SDK supplies the essential APIs and tools needed to build Android apps, including libraries for UI components, location services, data storage, and multimedia support. Tools like the Android Emulator and Android Virtual Devices (AVDs) enable developers to simulate different devices, screen sizes, and Android versions, making testing more comprehensive and efficient. For debugging and performance monitoring, developers rely on Logcat, a built-in logging tool, and ADB (Android Debug Bridge), which facilitates communication between development machines and devices.

Android Studio 13.2

Android Studio is the official Integrated Development Environment (IDE) for Android app development, based on IntelliJ IDEA. On top of IntelliJ's powerful code editor and developer tools, Android Studio offers even more features that enhance your productivity when building Android apps, such as: A flexible Gradle-based build system.

A fast and feature-rich emulator− A unified environment where you can develop for all Android devices− Instant Run to push changes to your running app without building a new APK− Code templates and GitHub integration to help you build common app features and import sample code− Extensive testing tools and frameworks Lint tools to catch performance, usability, version compatibility, and other problems C++ support Built-in support for Google Cloud Platform, making it easy to integrate Google Cloud Messaging and App Engine.

CHAPTER 6
TEST RESULT AND ANALYSIS

6.1 TESTING

The testing phase of the “Responsive Medication Reminder with Guardian Alert” application was conducted systematically to ensure the reliability, functionality, and robustness of all its core modules. Each feature was tested on multiple Android devices running various API levels from Android 8.0 (Oreo) to Android 13 to ensure cross-version compatibility. The alarm configuration module was evaluated for accuracy in scheduling one-time and recurring alarms, and it consistently triggered alerts at the correct time without delay, even when the device was in sleep mode or locked. The alarm acknowledgment system functioned as intended, allowing the user to acknowledge alerts via button press, with a timeout feature correctly identifying non-responses. In scenarios where no response was detected, the emergency call module activated successfully, automatically initiating a phone call to the saved guardian number.

This key safety feature was verified in both online and offline conditions to simulate real emergency situations. The user registration and profile management system securely stored and retrieved user data, and profile updates were reflected in real time without system lag or crashes.

From a performance standpoint, the application maintained low CPU and battery usage, making it suitable for continuous background operation without negatively impacting overall device performance. Security measures were tested by attempting to access sensitive data without permission, and the application correctly denied access and handled permission prompts in compliance with Android’s runtime permission policies. Edge cases such as device reboots, forced stops, and alarm rescheduling were also tested, and the system resumed functionality correctly upon relaunch, confirming effective use of persistent storage and services. User experience testing showed that the interface was intuitive and accessible, even for elderly users, with clear labels, large buttons, and a straightforward workflow. Based on this comprehensive testing and analysis.

6.2 TEST OBJECTIVES

The primary objective of testing the “Responsive Medication Reminder with Guardian Alert” application is to ensure that the system performs its core functionalities reliably, accurately, and securely in real-world medical use cases. The testing process aims to validate that alarms are triggered precisely at the scheduled time and that users are notified through effective alert mechanisms. A critical goal is to verify that the system correctly detects whether or not the user acknowledges the alarm within the defined timeout period.

If a response is not detected, the system must seamlessly transition to its emergency notification functionality by initiating a phone call to the registered guardian or emergency contact. Another key objective is to test the app’s stability under various conditions such as device reboots, power-saving modes, and background operation.

Ensuring secure handling of user data, proper permission usage, and protection against unauthorized access are also major testing priorities. Additionally, the testing seeks to confirm that the application’s user interface is intuitive, responsive, and accessible, especially for elderly or non-technical users. By achieving these objectives, the testing process guarantees that the application can be safely and effectively used as a medical assistance tool for improving medication adherence and ensuring timely emergency responses.

CHAPTER 7
RESULT AND DISCUSSION

7.1 RESULT

The “Responsive Medication Reminder with Guardian Alert” application successfully achieved its intended functionality by providing users with timely reminders to take their medications and initiating emergency contact protocols in the event of non-responsiveness. The system reliably triggered alarms at the scheduled times, and users were able to acknowledge these alerts through the app’s interactive interface. In scenarios where the user failed to respond within the given timeout period, the application automatically initiated a phone call to the registered guardian, thereby ensuring immediate attention. The user registration, profile management, and alarm configuration modules functioned smoothly and retained data effectively across sessions.

Testing across various Android devices and operating system versions confirmed that the application performed consistently without crashes or significant delays. The app also demonstrated strong performance in terms of background service handling, even during device sleep or low-power conditions. Overall, the project delivered a stable, user-friendly, and reliable solution that meets the critical healthcare needs of users who require medication adherence and emergency support, especially elderly individuals or those with chronic health conditions.

7.2 CONCLUSION

The “Responsive Medication Reminder with Guardian Alert” application effectively addresses a critical need in the healthcare domain by combining medication management with real-time emergency support. Through the integration of alarm scheduling, user acknowledgment detection, and automatic guardian notification, the application provides a reliable solution for individuals who require regular medication and close monitoring. The system ensures that users are reminded of their medication at the correct time and that their well-being is monitored through an intelligent acknowledgment mechanism. In situations where the user fails to respond, the automated emergency call feature acts as a safety net, immediately alerting a caregiver or family member, thereby reducing the risk of medical emergencies going unnoticed.

The app has been designed with a user-friendly interface and tested for reliability, responsiveness, and security across various Android platforms. It demonstrates the potential of mobile technology to enhance personal healthcare, particularly for elderly users, patients with chronic conditions, or individuals living alone. By combining simplicity with critical functionality, this project not only improves medication adherence but also offers peace of mind to users and their families. The successful implementation of this system lays the foundation for future enhancements such as SMS alerts, GPS tracking, and cloud integration, making it a scalable and impactful tool in digital health support.

7.3 FUTURE ENHANCEMENT

While the “Responsive Medication Reminder with Guardian Alert” application fulfills its core purpose effectively, there are several areas for future enhancement that can further improve its functionality, usability, and reach. One potential upgrade is the integration of SMS or WhatsApp alerts alongside the emergency call feature, ensuring that guardians receive messages even if they are unable to answer a phone call. Additionally, GPS tracking can be incorporated to share the user’s real-time location during an emergency, which would be especially helpful for caregivers or medical responders in locating the individual quickly. Another valuable addition could be voice-based interaction or speech recognition, allowing elderly users or users with disabilities to interact with the app more easily without the need for typing or touching the screen.

Cloud integration is another major future improvement. By synchronizing user profiles, alarm data, and history logs to a cloud database, users can access their data from multiple devices, and caregivers can remotely monitor medication adherence. The app can also be enhanced with AI-based analytics that track user behavior over time and provide predictive alerts or health insights.

If a user frequently misses medication, the system could notify a doctor or suggest schedule adjustments. Support for multilingual interfaces would make the app more accessible to non-English speaking users, while integration with wearable devices like smartwatches could further simplify alarm notifications and acknowledgment.

APPENDIX – 1

SOURCE CODE

MainActivity.java

```
package com.example.medicationreminder;
import android.app.AlarmManager;
import android.app.PendingIntent;
import android.content.Context;
import android.content.Intent;
import android.media.MediaPlayer;
import android.os.Build;
import android.os.Bundle;
import android.widget.ArrayAdapter;
import android.widget.Button;
import android.widget.EditText;
import android.widget.ListView;
import android.widget.Spinner;
import android.widget.Toast;

import androidx.appcompat.app.AppCompatActivity;
import java.text.SimpleDateFormat;
import java.util.ArrayList;
import  java.util.Calendar;
import java.util.Locale;
public class MainActivity extends AppCompatActivity {
  private EditText medicationName, reminderTime, guardianPhone;
    private Spinner repeatOption;
    private Button addReminderButton, stopAlarmButton;
    private ListView remindersListView;
    private ArrayList<String> remindersList;
```

```
private ArrayAdapter<String> adapter;
private MediaPlayer mediaPlayer;
@Override
   protected void onCreate(Bundle savedInstanceState) {
      super.onCreate(savedInstanceState);
      setContentView(R.layout.activity_main);
      medicationName = findViewById(R.id.medicationName);
      reminderTime = findViewById(R.id.reminderTime);
      guardianPhone = findViewById(R.id.guardianPhone);
      repeatOption = findViewById(R.id.repeatOption);
      addReminderButton = findViewById(R.id.addReminderButton);
      stopAlarmButton = findViewById(R.id.stopAlarmButton);
      remindersListView = findViewById(R.id.remindersListView);
      remindersList = new ArrayList<>();
      adapter = new ArrayAdapter<>(this, android.R.layout.simple_list_item_1,
remindersList);
      remindersListView.setAdapter(adapter);
      ArrayAdapter<CharSequence> spinnerAdapter =
ArrayAdapter.createFromResource(this,
            R.array.repeat_options, android.R.layout.simple_spinner_item);
spinnerAdapter.setDropDownViewResource(android.R.layout.simple_spinner_dropdo
wn_item);
      repeatOption.setAdapter(spinnerAdapter);
      addReminderButton.setOnClickListener(v -> addReminder());
      stopAlarmButton.setOnClickListener(v -> stopAlarm());
   }

   private void addReminder() {
      String name = medicationName.getText().toString();
      String time = reminderTime.getText().toString();
```

```
        String guardian = guardianPhone.getText().toString();
        String repeat = repeatOption.getSelectedItem().toString();
        if (name.isEmpty() || time.isEmpty() || guardian.isEmpty()) {
            Toast.makeText(this, "Please fill in all fields",
Toast.LENGTH_SHORT).show();
            return;
        }
        try {
            SimpleDateFormat sdf = new SimpleDateFormat("HH:mm",
Locale.getDefault());
            Calendar calendar = Calendar.getInstance();
            // Parse the user-provided time
            String[] timeParts = time.split(":");
            int hour = Integer.parseInt(timeParts[0]);
            int minute = Integer.parseInt(timeParts[1]);

            // Set the reminder time
            calendar.set(Calendar.HOUR_OF_DAY, hour);
            calendar.set(Calendar.MINUTE, minute);
            calendar.set(Calendar.SECOND, 0);
            // If the time is in the past, schedule it for the next day
            if (calendar.getTimeInMillis() <= System.currentTimeMillis()) {
                calendar.add(Calendar.DAY_OF_YEAR, 1);
            }
            // Log the scheduled time for debugging
            System.out.println("Scheduled reminder for: " + calendar.getTime());
            // Schedule the alarm
            scheduleAlarm(name, guardian, calendar, repeat);
            remindersList.add(name + " at " + new SimpleDateFormat("HH:mm",
Locale.getDefault()).format(calendar.getTime()));
```

```
            adapter.notifyDataSetChanged();
            medicationName.setText("");
            reminderTime.setText("");
            guardianPhone.setText("");

            Toast.makeText(this, "Reminder added", Toast.LENGTH_SHORT).show();
        } catch (Exception e) {
            Toast.makeText(this, "Invalid time format", Toast.LENGTH_SHORT).show();
        }
    }
    private void scheduleAlarm(String medicationName, String guardianPhone,
Calendar calendar, String repeat) {
        AlarmManager alarmManager = (AlarmManager)
getSystemService(Context.ALARM_SERVICE);
        Intent intent = new Intent(this, ReminderReceiver.class);
        intent.putExtra("medicationName", medicationName);
        intent.putExtra("guardianPhone", guardianPhone);
        PendingIntent pendingIntent = PendingIntent.getBroadcast(this, 0, intent,
PendingIntent.FLAG_UPDATE_CURRENT);
        try {
            if (Build.VERSION.SDK_INT >= Build.VERSION_CODES.M) {
              alarmManager.setExactAndAllowWhileIdle(AlarmManager.RTC_WAKEUP,
calendar.getTimeInMillis(), pendingIntent);
            } else {
               alarmManager.setExact(AlarmManager.RTC_WAKEUP,
calendar.getTimeInMillis(), pendingIntent);
            }
            Toast.makeText(this, "Alarm scheduled for: " + calendar.getTime().toString(),
Toast.LENGTH_SHORT).show();
        } catch (SecurityException e) {
```

```
            Toast.makeText(this, "Failed to schedule alarm: " + e.getMessage(),
Toast.LENGTH_LONG).show();
        }
        // Handle repeating alarms
        long interval = 0;
        switch (repeat) {
            case "Daily":
                interval = AlarmManager.INTERVAL_DAY;
                break;
            case "Weekly":
                interval = AlarmManager.INTERVAL_DAY * 7;
                break;
            case "Monthly":
                interval = AlarmManager.INTERVAL_DAY * 30;
                break;
        }
        if (interval > 0) {
            alarmManager.setRepeating(AlarmManager.RTC_WAKEUP,
calendar.getTimeInMillis(), interval, pendingIntent);
        }
    }
    private void stopAlarm() {
        AlarmManager alarmManager = (AlarmManager)
getSystemService(Context.ALARM_SERVICE);
        Intent intent = new Intent(this, ReminderReceiver.class);
        PendingIntent pendingIntent = PendingIntent.getBroadcast(this, 0, intent,
PendingIntent.FLAG_UPDATE_CURRENT);
        if (mediaPlayer != null && mediaPlayer.isPlaying()) {
            mediaPlayer.stop();
            mediaPlayer.release();
```

```
            mediaPlayer = null;
        }
        alarmManager.cancel(pendingIntent);
        pendingIntent.cancel();
        ReminderReceiver.stopAlarmSound();
        Toast.makeText(this, "Alarm stopped", Toast.LENGTH_SHORT).show();
    }
    @Override
    protected void onDestroy() {
        super.onDestroy();
        if (mediaPlayer != null) {
            mediaPlayer.release();
        }
    }
}
```

activity_main.xml

```
<LinearLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    android:padding="16dp">
    <EditText
        android:id="@+id/medicationName"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:autofillHints=""
        android:hint="@string/Add_medication_name"
        android:inputType="text" />
    <EditText
```

```
    android:id="@+id/reminderTime"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:autofillHints=""
    android:hint="@string/Add_Reminder_Time"
    android:inputType="time" />
<EditText
    android:id="@+id/guardianPhone"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:autofillHints=""
    android:hint="@string/Add_Guardian_Phone_Number"
    android:inputType="phone" />
<Spinner
    android:id="@+id/repeatOption"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_marginTop="8dp" />
<Button
    android:id="@+id/addReminderButton"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:text="@string/Add_Reminder" />
<Button
    android:id="@+id/stopAlarmButton"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:text="@string/Stop_alarm" />

<ListView
```

```
        android:id="@+id/remindersListView"
        android:layout_width="match_parent"
        android:layout_height="0dp"
        android:layout_weight="1"
        android:layout_marginTop="16dp"
        android:divider="@android:color/darker_gray"
        android:dividerHeight="1dp" />
</LinearLayout>
```

Strings.xml

```
<resources>
    <string name="app_name">MedicationReminder</string>
    <string name="action_settings">Settings</string>
    <!-- Strings used for fragments for navigation -->
    <string name="Add_medication_name">Enter Medication Name</string>
    <string name="Add_Reminder_Time">Enter Reminder Time (HH:mm)</string>
    <string name="Add_Guardian_Phone_Number">Enter Guardian Phone
Number</string>
    <string name="Add_Reminder">Add Reminder</string>
    <string name="Stop_alarm">Stop Alarm</string>
    <string name="first_fragment_label">First Fragment</string>
    <string name="second_fragment_label">Second Fragment</string>
    <string name="next">Next</string>
    <string name="previous">Previous</string>
    <string-array name="repeat_options">
        <item>Once</item>
        <item>Daily</item>
        <item>Weekly</item>
        <item>Monthly</item>
    </string-array>
    <string name="lorem_ipsum">
```

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nam in scelerisque sem. Mauris volutpat, dolor id interdum ullamcorper, risus dolor egestas lectus, sit amet mattis purus

dui nec risus. Maecenas non sodales nisi, vel dictum dolor. Class aptent taciti sociosqu ad

litora torquent per conubia nostra, per inceptos himenaeos. Suspendisse blandit eleifend

diam, vel rutrum tellus vulputate quis. Aliquam eget libero aliquet, imperdiet nisl a,

ornare ex. Sed rhoncus est ut libero porta lobortis. Fusce in dictum tellus.\n\n

Suspendisse interdum ornare ante. Aliquam nec cursus lorem. Morbi id magna felis. Vivamus

egestas, est a condimentum egestas, turpis nisl iaculis ipsum, in dictum tellus dolor sed

neque. Morbi tellus erat, dapibus ut sem a, iaculis tincidunt dui. Interdum et malesuada

fames ac ante ipsum primis in faucibus. Curabitur et eros porttitor, ultricies urna vitae,

molestie nibh. Phasellus at commodo eros, non aliquet metus. Sed maximus nisl nec dolor

bibendum, vel congue leo egestas.\n\n

Sed interdum tortor nibh, in sagittis risus mollis quis. Curabitur mi odio, condimentum sit

amet auctor at, mollis non turpis. Nullam pretium libero vestibulum, finibus orci vel,

molestie quam. Fusce blandit tincidunt nulla, quis sollicitudin libero facilisis et. Integer

interdum nunc ligula, et fermentum metus hendrerit id. Vestibulum lectus felis, dictum at

lacinia sit amet, tristique id quam. Cras eu consequat dui. Suspendisse sodales

nunc ligula,

in lobortis sem porta sed. Integer id ultrices magna, in luctus elit. Sed a pellentesque

est.\n\n

Aenean nunc velit, lacinia sed dolor sed, ultrices viverra nulla. Etiam a venenatis nibh.

Morbi laoreet, tortor sed facilisis varius, nibh orci rhoncus nulla, id elementum leo dui

non lorem. Nam mollis ipsum quis auctor varius. Quisque elementum eu libero sed commodo. In

eros nisl, imperdiet vel imperdiet et, scelerisque a mauris. Pellentesque varius ex nunc,

quis imperdiet eros placerat ac. Duis finibus orci et est auctor tincidunt. Sed non viverra

ipsum. Nunc quis augue egestas, cursus lorem at, molestie sem. Morbi a consectetur ipsum, a

placerat diam. Etiam vulputate dignissim convallis. Integer faucibus mauris sit amet finibus

convallis.\n\n

Phasellus in aliquet mi. Pellentesque habitant morbi tristique senectus et netus et

malesuada fames ac turpis egestas. In volutpat arcu ut felis sagittis, in finibus massa

gravida. Pellentesque id tellus orci. Integer dictum, lorem sed efficitur ullamcorper,

libero justo consectetur ipsum, in mollis nisl ex sed nisl. Donec maximus ullamcorper

sodales. Praesent bibendum rhoncus tellus nec feugiat. In a ornare nulla. Donec rhoncus

libero vel nunc consequat, quis tincidunt nisl eleifend. Cras bibendum enim a justo luctus

vestibulum. Fusce dictum libero quis erat maximus, vitae volutpat diam dignissim.

</string>

</resources>

AndroidManifest.xml

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools">
    <uses-permission android:name="android.permission.SEND_SMS" />
    <uses-permission android:name="android.permission.CALL_PHONE" />
    <uses-feature android:name="android.hardware.telephony" required="false"
        tools:ignore="MissingPrefix" />
    <uses-permission
android:name="android.permission.SCHEDULE_EXACT_ALARM"
android:permissionGroup="android.permission-group.SCHEDULE" />
    <application
        android:allowBackup="true"
        android:dataExtractionRules="@xml/data_extraction_rules"
        android:fullBackupContent="@xml/backup_rules"
        android:icon="@mipmap/ic_launcher"
        android:label="@string/app_name"
        android:roundIcon="@mipmap/ic_launcher_round"
        android:supportsRtl="true"
        android:theme="@style/Theme.MedicationReminder"
        tools:targetApi="31">
        <activity
            android:name=".MainActivity"
            android:exported="true"
            android:theme="@style/Theme.MedicationReminder">
            <intent-filter>
```

```
            <action android:name="android.intent.action.MAIN" />
            <category android:name="android.intent.category.LAUNCHER" />
        </intent-filter>
    </activity>
    <receiver android:name=".ReminderReceiver" />
  </application>
</manifest>
```

ReminderReceiver.java

```
package com.example.medicationreminder;
import  android.content.BroadcastReceiver;
import android.content.Context;
import android.content.Intent;
import android.media.MediaPlayer;
import android.media.Ringtone;
import android.media.RingtoneManager;
import android.net.Uri;
import android.os.Handler;
import android.telephony.SmsManager;
import android.widget.Toast;

public class ReminderReceiver extends BroadcastReceiver {

  private static MediaPlayer mediaPlayer; // Use static to persist state across
broadcasts
  private static Ringtone ringtone;      // Static to manage a single instance
  private static Handler handler = new Handler(); // For delayed actions
  private static Runnable stopAlarmRunnable;      // Runnable for delayed stop

  @Override
  public void onReceive(Context context, Intent intent) {
```

```
        String action = intent.getAction();

        // Handle stop alarm action
        if ("STOP_ALARM".equals(action)) {
            stopAlarmSound();
            return;  // Exit immediately if it's a stop alarm action
        }

        // Get extras from the intent
        String medicationName = intent.getStringExtra("medicationName");
        String guardianPhone = intent.getStringExtra("guardianPhone");

        // Notify the user
        Toast.makeText(context, "Time to take: " + medicationName,
Toast.LENGTH_LONG).show();

        // Play ringtone
        Uri alarmUri =
RingtoneManager.getDefaultUri(RingtoneManager.TYPE_ALARM);
        if (alarmUri == null) {
            alarmUri =
RingtoneManager.getDefaultUri(RingtoneManager.TYPE_NOTIFICATION);
        }
        ringtone = RingtoneManager.getRingtone(context, alarmUri);
        ringtone.play();

        // Play additional alarm sound using MediaPlayer
        playAlarmSound(context);

        // Schedule notification to guardian after 1 minute
```

```
    stopAlarmRunnable = () -> {
        stopAlarmSound(); // Stop the alarm sound
        notifyGuardian(context, guardianPhone, medicationName);
    };
    handler.postDelayed(stopAlarmRunnable, 60 * 1000); // 1-minute delay
}

private void playAlarmSound(Context context) {
    if (mediaPlayer == null) {
        mediaPlayer = MediaPlayer.create(context, R.raw.music_05);
        mediaPlayer.setLooping(true);
        mediaPlayer.start();
    }
}
public static void stopAlarmSound() {
    // Stop MediaPlayer if it's playing
    if (mediaPlayer != null && mediaPlayer.isPlaying()) {
        mediaPlayer.stop();
        mediaPlayer.release();
        mediaPlayer = null;
    }
    // Stop Ringtone if it's playing
    if (ringtone != null && ringtone.isPlaying()) {
        ringtone.stop();
        ringtone = null;
    }
    // Cancel any pending stop alarm runnable
    if (handler != null && stopAlarmRunnable != null) {
        handler.removeCallbacks(stopAlarmRunnable);
    }
```

```
    }
    private void notifyGuardian(Context context, String phoneNumber, String
medicationName) {
        sendSMS(context, phoneNumber, medicationName);
        makeCall(context, phoneNumber);
    }

    private void sendSMS(Context context, String phoneNumber, String
medicationName) {
        try {
            SmsManager smsManager = SmsManager.getDefault();
            String message = "Reminder Alert: Medication (" + medicationName + ") has
not been taken. Please check.";
            smsManager.sendTextMessage(phoneNumber, null, message, null, null);
            Toast.makeText(context, "SMS sent to guardian: " + phoneNumber,
Toast.LENGTH_LONG).show();
        } catch (Exception e) {
            Toast.makeText(context, "Failed to send SMS to guardian.",
Toast.LENGTH_LONG).show();
            e.printStackTrace();
        }
    }

    private void makeCall(Context context, String phoneNumber) {
        try {
            Intent callIntent = new Intent(Intent.ACTION_CALL);
            callIntent.setData(Uri.parse("tel:" + phoneNumber));
            callIntent.setFlags(Intent.FLAG_ACTIVITY_NEW_TASK);
            context.startActivity(callIntent);
            Toast.makeText(context, "Calling guardian: " + phoneNumber,
```

```
Toast.LENGTH_LONG).show();
        } catch (SecurityException e) {
            Toast.makeText(context, "Unable to make the call. Missing permission.",
Toast.LENGTH_LONG).show();
            e.printStackTrace();
        }
    }
```

APPENDIX – 2

SCREENSHOTS

Sample Output

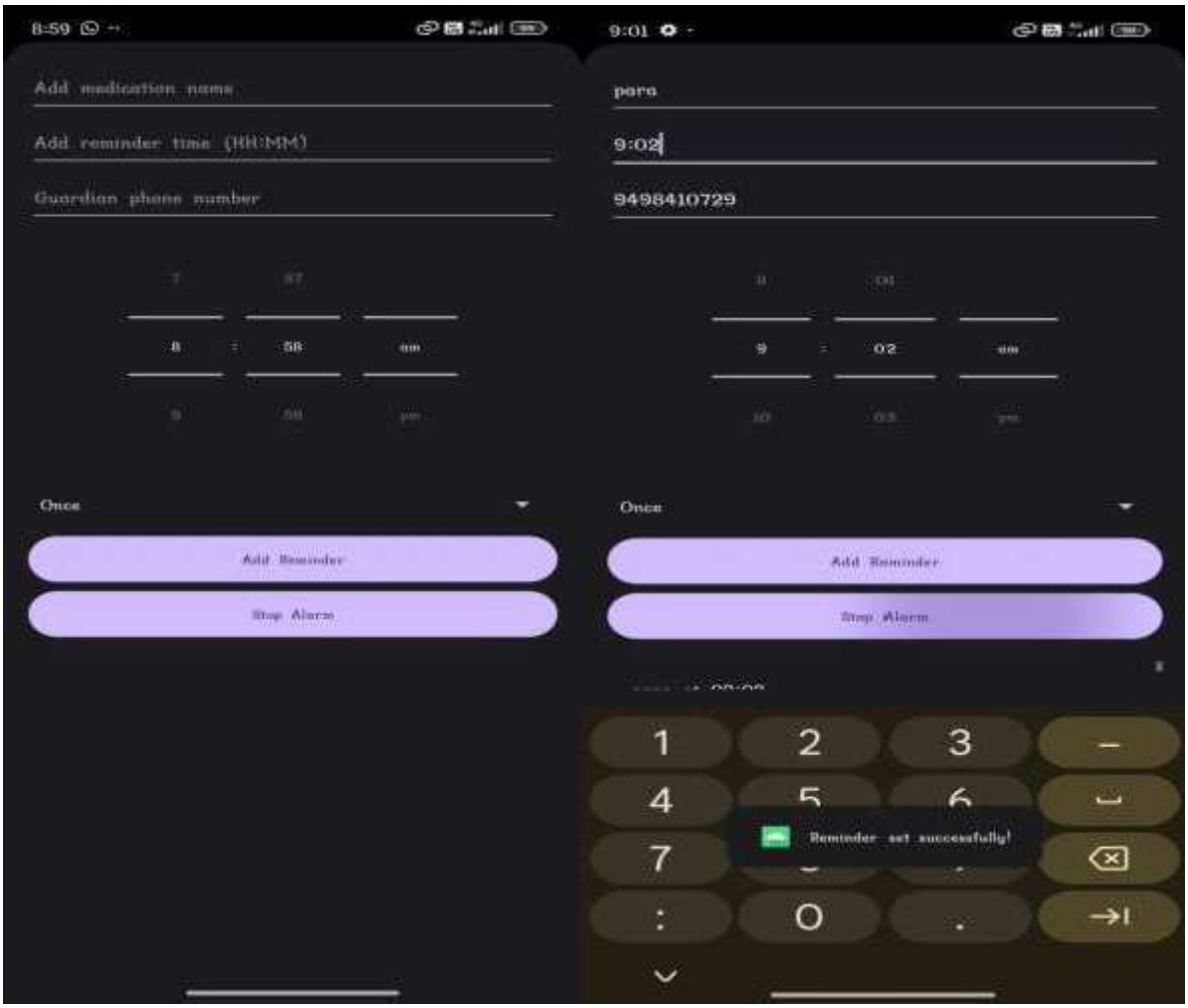

Figure 6.1: Add medication Reminder

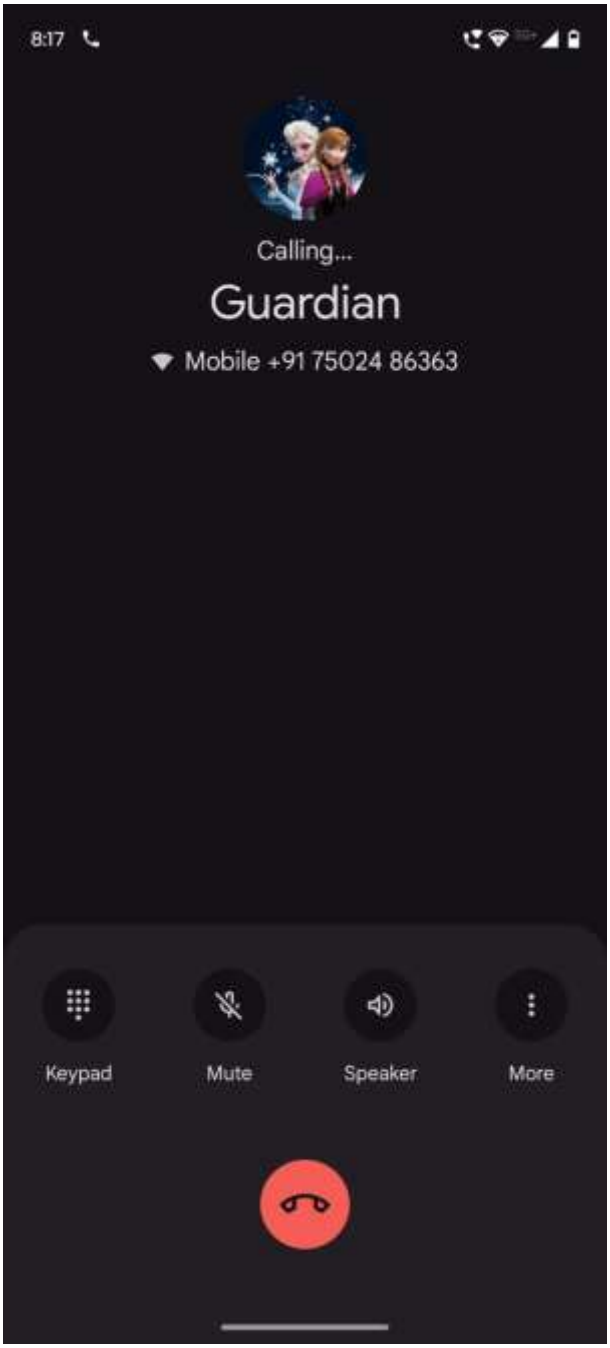

Figure 6.2: Reminding Caretaker

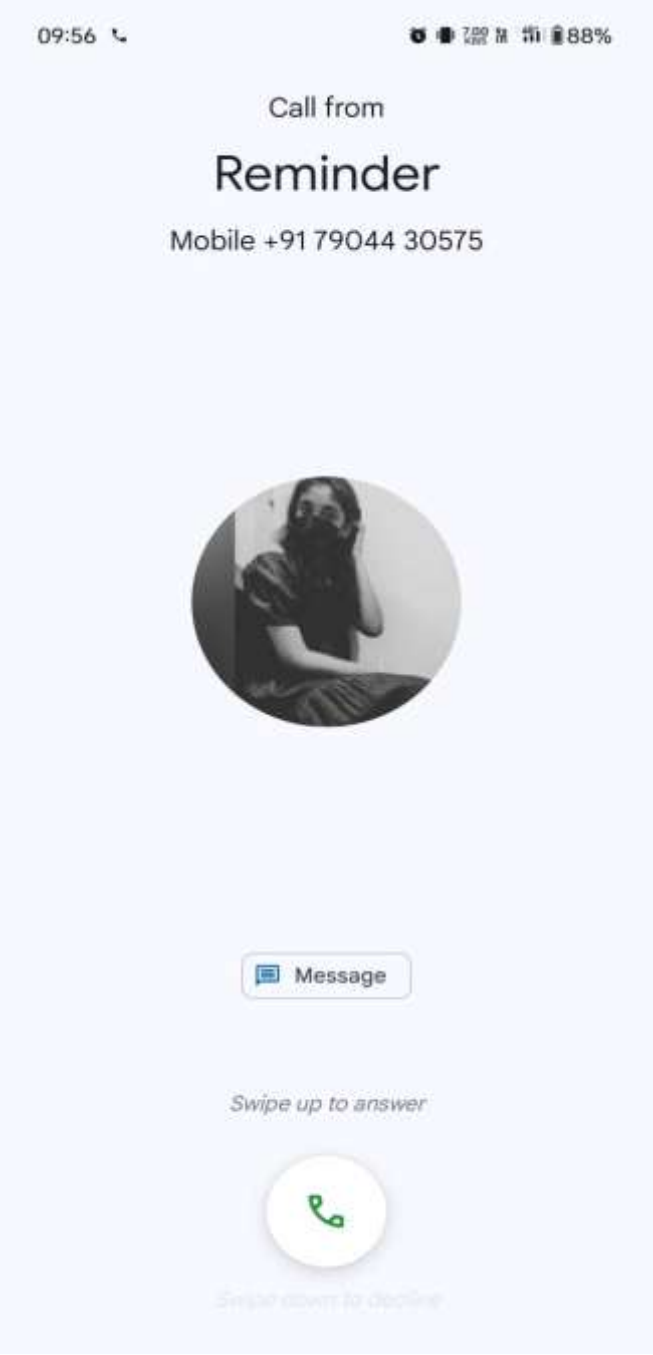

Figure 6.3: Alert caretaker via phone call

REFERENCES

1. Android Developers. (n.d.). *Alarm Manager | Android Developers*. Retrieved from:
2. World Health Organization. (2003). *Adherence to Long-Term Therapies: Evidence for Action*. Retrieved from:
3. Lee, J., & Kim, J. (2020). *Development of a Mobile-Based Medication Reminder System for Chronic Patients*. International Journal of Medical Informatics, 141, 104226. https://doi.org/10.1016/j.ijmedinf.2020.104226
4. Google. (n.d.). *Permissions Overview | Android Developers*. Retrieved from:
5. Kumar, P., & Tripathi, S. (2018). *An Android-based Medicine Reminder System with Call Feature for Elderly People*. International Journal of Advanced Research in Computer Science, 9(3), 42–46.
6. Puri, R., & Sharma, A. (2019). *Smart Healthcare Monitoring and Alert System using IoT*. In Proceedings of the 2nd International Conference on Intelligent Computing and Control Systems (ICICCS). IEEE.
7. Appleby, J. (2017). *The Role of Mobile Applications in Supporting Medication Adherence*. BMJ Innovations, 3(2), 64–68.
8. Android Developers. (n.d.). *Broadcast Receiver | Android Developers*. Retrieved from:
9. Lin, L., & Wu, Y. (2016). *Design and Implementation of an Android-Based Emergency Calling System for the Elderly*. Journal of Medical Systems, 40, 187.
10. Goyal, N., & Arora, A. (2021). *Review on Healthcare Applications Based on Android Platform*. International Journal of Engineering Research & Technology (IJERT), 10(2), 45–48.