

CHAPTER 1

INTRODUCTION

1.1 BACKGROUND

Mobile technologies are finding a role in patient monitoring in several different environments: homes, hospitals, and nursing homes. health applications include the use of mobile devices in collecting community and clinical health data, delivery of healthcare information to practitioners, researchers and patients, real-time monitoring of patient vital signs, and direct provision of care. Therefore, usability of software applications is the key to success of any system. This study focused on creating an application for smart phones with android system. The main aim of the proposed system is to help two important categories of the society they are elderly and Alzheimer's patients, these two categories shared in one recipe which is forgetful. This system gives them the ability to have small memory can help them to remember all tasks to life, which may contribute to the prevention of progression of the disease rapidly, and the technology is best care because it is not susceptible to forget or damage. The design of the proposed system presented in this study includes reminding them of the dates of their medications and the amount of medicine. It also reminders food times and some of the important events. The study concludes that the working environment would be improved by supporting with mobile technology and reduces the health expenditures and burden of health care professionals in care facility units.

1.2 OVERVIEW

India has second Most Number of Individual suffering from Dementia While worldwide in 2015, there were between 35 and 47 million people worldwide with AD. It begins over 65 years of age, though there are only 4% to 5% of cases of early-onset Alzheimer's beginning before this. As per a Research it affects about 6% of people 65 years and older. As we can know the Alzheimer patients have short term memory, so they face some common day to day Problems like wandering off, forgetting their diet, forget faces/names of family members are the most common one. Some of the Big Problem that they may face is They Forget the Important Data like Their Address and Doctors treating them etc. To solve all this problem Alzheimer patients, must keep a caretaker with them. The aim of the paper is to focus on Important Aspect of the Diseases which is Affecting a Huge Amount of Population today (i.e.) Alzheimer and building an Android application that could ease the everyday life of a person affected by Alzheimer disease. This app provides various functionalities such as tracking movements of the patient through GPS, providing medicine and food timing notifications, daily routine tracker and quiz/Games to increase cognitive functioning of the patient and Timely Alarm for to provide a Reminder of the App. The main objective of this project is to make people suffering from Alzheimer as independent as possible. These systems use a variety of technologies, such wearables or smartphone apps, to send out timely reminders for daily tasks, appointments, and prescription intake. These systems can also be equipped with alert systems to inform family members or caregivers to crises or departures from normal behaviour. By utilizing technology in this way, caregivers may help Alzheimer's patients by providing timely aid and support, thereby improving their quality of life and safety. They can also receive real-time information and alarms.

1.3 PROBLEM STATEMENT

These days, a lot of people rely heavily on mobile applications, which are utilized in most areas of our lives, particularly in the medical industry. This method talks about a smartphone application that helps elderly and Alzheimer's sufferers remember to aid and enhance the quality of care. Using state-of-the-art technology, an integrated solution for caregiver alerts and reminders is designed as part of a suggested method to address the challenges associated with managing people with Alzheimer's disease. Using wearable sensors, this system would monitor vital signs, activity levels, and location in real time. Caregivers can set up reminders for daily tasks, appointments, and prescription schedules by connecting these devices with a centralized platform accessible via a computer or smartphone. Furthermore, the proposed system would include remote monitoring capabilities that would enable caregivers to check the patient's condition even when they are not physically there. This remote monitoring feature allowed caregivers to maintain a certain level of independence while feeling secure.

1.4 OBJECTIVE

The objective of a medication reminder system or application is to improve medication adherence by ensuring users take their prescribed medications at the correct times . This contributes to better health outcomes, reduces the risk of missed doses or overdoses, and helps manage chronic conditions effectively.

1.5 IMPLICATION

A medication reminder system has significant implications for healthcare and patient well-being. Positively, it can improve medication adherence, leading to better management of chronic conditions, reduced complications, and enhanced health outcomes. Patients are empowered to take control of their treatment, while caregivers can remotely monitor medication compliance, offering peace of mind. Additionally, improved adherence can lower healthcare costs by minimizing hospitalizations and emergency visits.

CHAPTER 2

LITERATURE SURVEY

TITLE : SMART MEDICINE REMINDER BOX

AUTHORS: Sanjay Bhati, Harshad Soni

YEAR : 2023

In day-to-day life most of the people need to take medicines which was not there in past couple of years and the reason behind this is diseases are increasing in large amount. So sooner or later many people meet these diseases. Some diseases are temporary diseases while many are permanent life-threatening diseases. Life threatening diseases gets mixes with the human body in such a way that they can't leave the body ever and they increase in rapid time. Life span of humans became less because of such diseases and to overcome or to live a better life we need to take medicines regularly and in large amount. We need to be in advice of doctor who tells us to take desired pills in desired way so that patients face problems like forgetting pills to take at right time and when Doctor changes the prescription of medicine patients must remember the new schedule of medicine. This problem of forgetting to take pills at right time, taking wrong medicines and accidentally taking of expired medicine causes health issues of patient and this leads to suffer from unhealthy life. Our project is to made Arduino-Uno based Smart medicine box which uses Real time clock. The new awaited feature in our project is our system is sensible that patient has taken medicine or not and thus the patient can't postpone the time on which he needs to take pills. It is compulsory for the patient to take pills from the box at the right time otherwise our systems continue to make large sound until the medicine is taken out from the box. This notification feature adds life years to the patient and thus this thing is not available in any device which is the necessity for present days

TITLE : ARDUINO BASED MEDICINE REMINDER

AUTHORS: Urvashi Sharma, Chetna Chauhan, Himani Sharma, Anjali Sharma

YEAR : 2020

Many patients fail to comply with their prescribed medication schedules. This can cause disease complications, lower quality of life, and even mortality. To overcome these issues the automatic medicine reminder is used. This system is very simple to operate and update therefore a person of any age group can use it. This system can also be useful in hospital where number of patients is present and sometimes it is difficult to remember the medicine and dosage by the staff. So, this system with some updates can also be used in hospitals. The problem such as 1) maintaining the regularity of prescribed dosage is difficult to be remembered in busy schedule 2) remembering the name of medicine to be taken is difficult 3) due to above two reasons the patient's life can get more complicated. These above problems are arising to everyone due to non-adherence of medications. Therefore, there is a growing need and urgency for in-home healthcare devices and technologies to provide patients with the electronic tools to support medication self-management.

**TITLE : ALARM SYSTEM FOR MEDICINE REMINDER BASED ON
MICROCONTROLLER**

AUTHOR : Ni San Hlaing, San San Naing

YEAR : 2018

Many patients fail to comply with their prescribed medication schedules. This causes disease complications, lower quality of life and even mortality. To overcome these issues the automatic medicine reminder is used. This system is very simple to operate and update therefore a person of any age group uses it. This system is also useful in hospital where number of patients is present and sometimes it is difficult to remember the medicine and dosage by the staff. So, this system with some updates is also used in hospitals. The problem such as maintaining the regularity of prescribed dosage is difficult to be remembered in busy schedule, remembering the name of medicine to be taken is difficult and due to above two reasons the patient's life can get more

complicated. These above problems are arising to everyone due to non-adherence of medications. Therefore, there is a growing need and urgency for in-home healthcare devices and technologies to provide patients with the electronic tools to support medication self-management.

**TITLE : RASBERRY PI MEDICINE REMINDER E-MAIL ALERT
USING IOT**

AUTHOR : CH. Santhosha Kumara. Goutham

YEAR : 2018

In day-to-day life, people face trouble remembering the pills that need to be consumed. Many times, the problem is that the time required to take the medicine is not printed on the packaging of medicine. People also have a habit of forgetting to take pills. Sometimes medicines go beyond their expiry date. Pervasive healthcare is the next generation form of healthcare and distributed, patient-centric and self-managed care. It is an alternative to the traditional hospitalized, staff and professional-managed care. Pervasive healthcare is based on emerging technologies like the Internet-of-Things (IoT). Health informatics is a developing area to advance health problems, mainly deals with the acquisition, transmission, processing, storage, retrieval, and use of different types of information in the health field. The main acquisition technologies of health information are sensing. Most patients with chronic diseases need to take medications over a prolonged period to stabilize their conditions. Ensuring that the patients consume the right medication at the appropriate time becomes crucial. This project deals with the time at which the patient needs to take pills. The timing is set to the system initially and it can be changed by the patient according to his/her requirement. The system will start an alarm at that time. To make the user-friendly system, an LCD and a keypad is connected to the system, this helps to change pill timings. After having pills, the user will have to put the number of pills removed from the box. confusion among pills.

TITLE : DEVELOPMENT OF AUTOMATIC REMINDER SYSTEM FOR
GERIATRIC MEDICIN INTAKE

AUTHOR : Aisyah Rahimi, Hamim Zakri, Azira Khalil

YEAR : 2018

Geriatrics is referred to individuals aged from 50 years old and above that require regard to their healthcare problem. They are also defined as aged individuals receiving special care to improve their lives. This group of individuals is often involved with medicine intake in their daily life. Their medication needs can be due to various diseases, such as hypertension, diabetes mellitus, Alzheimer's, stroke, and others. Statistic from recent studies shows that approximately 80% of elderly in the United States of American have at least one chronic disease, while 77% of them have two chronic diseases per individual. These diseases can cause almost death factors to the geriatrics every year. Meanwhile, diabetes and hypertension are common diseases associated with geriatrics, where 23% of Americans are affected by diabetes, and 90% of elderly Americans are at risk for hypertension. According to, getting older means that we are more exposed to healthcare problems, and only 41% of elderly aged 65 are free from any diseases. Geriatrics need medication to improve their health and help them in their daily life. The functionality of this project is to help geriatrics in taking their medicine at the exact time. This project also allows geriatrics to consume which medicine at a particular time.

CHAPTER 3

SYSTEM ANALYSIS

3.1 EXISTING SYSTEM

In the absence of advanced technological solutions, caregivers of Alzheimer's patients often rely on manual systems to provide reminders and alerts. These manual systems typically involve setting up physical reminders such as sticky notes or calendars placed strategically around the house to prompt the patient about medication schedules, appointments, or daily tasks. Caregivers also maintain close observation of the patient's behavior and routines, noting any deviations or signs of distress. In case of emergencies or unusual behavior, caregivers communicate through phone calls or in-person contact with family members or healthcare providers. While these manual systems demonstrate the dedication and vigilance of caregivers, they can be prone to human error and may not always provide timely or comprehensive support. Despite their limitations, manual systems remain a fundamental component of caregiving for Alzheimer's patients, highlighting the need for more advanced and efficient technological solutions to enhance patient care and support for caregivers.

DISADVANTAGES

- Manual reminders such as sticky notes or calendars are only effective when the patient is near them.
- Human error is inherent in manual systems. Caregivers may forget to update reminders, misplace important notes, or provide incorrect information, leading to missed medications or appointments.
- Manual systems are not equipped to handle emergencies efficiently.
- In case of sudden medical issues or accidents, there may be delays in alerting caregivers or accessing necessary assistance.

3.2 PROPOSED SYSTEM

Nowadays the mobile applications are very important for many people, these applications used at most field of our life especially at the medical. This system discusses a mobile application that works as a reminder for Elderly and Alzheimer's patients to help and improve the quality of care. An integrated solution for caregiver alerts and reminders is created by utilizing cutting-edge technology as part of a suggested approach to handle the difficulties associated with managing Alzheimer's patients. This system would track location, activity levels, and vital signs in real-time by using wearable sensors. Through syncing these devices with a centralized platform that can be accessed through a computer or smartphone, caregivers can set up reminders for daily tasks, appointments, and prescription schedules. In addition, the suggested system would have remote monitoring features that would let caretakers check the patient's health even when they are not there in person. Caregivers could keep some degree of independence while having peace of mind thanks to this remote monitoring capability.

ADVANTAGES

- The system provides timely reminders for medication schedules, appointments, and daily tasks, reducing the risk of missed doses or appointments.
- The system enhances the safety of Alzheimer's patients by detecting potential hazards and alerting caregivers to take appropriate action.
- The system facilitates communication between caregivers, patients, and healthcare providers, allowing for better coordination of care and support.
- Through data analysis and pattern recognition, the system can adapt to the individual needs and preferences of each Alzheimer's patient, providing personalized reminders and alerts.

3.3 BLOCK DIAGRAM OF PROPOSED SYSTEM

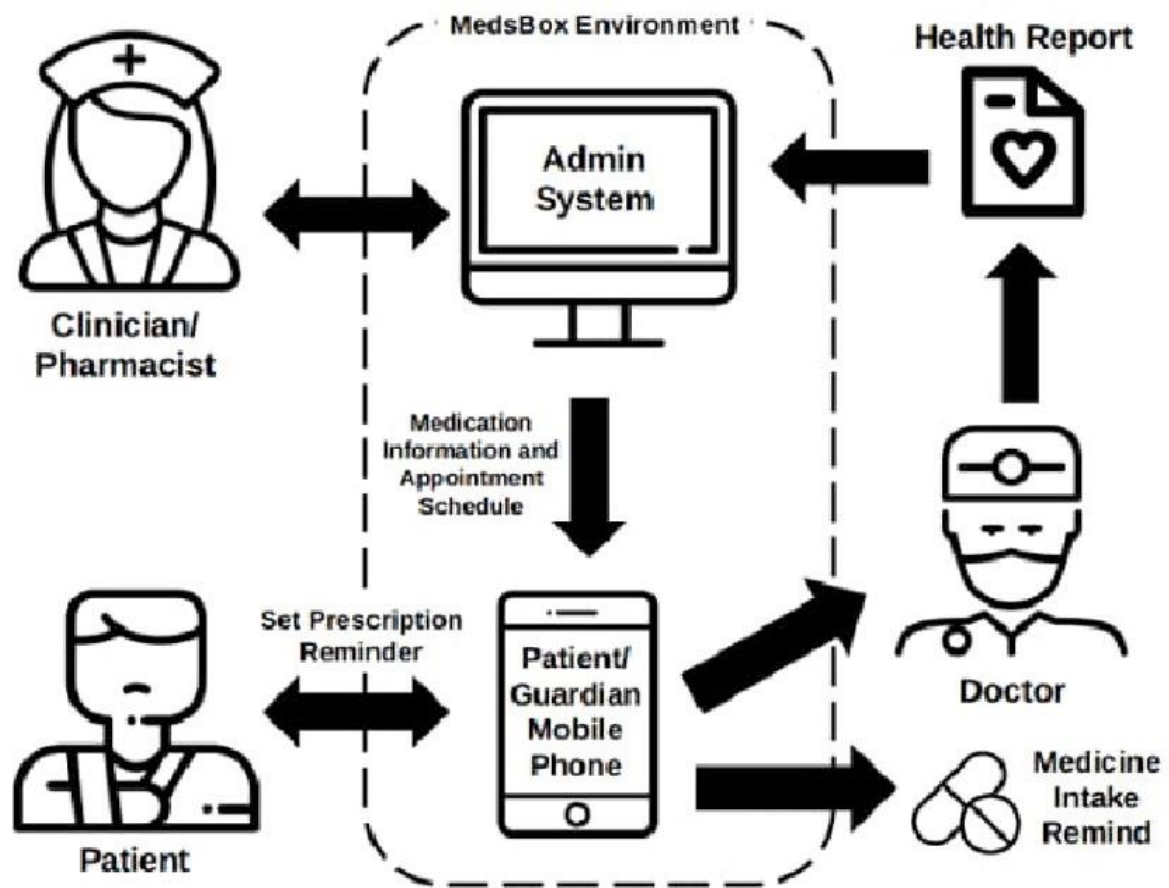


Figure 3.1: proposed system

3.4 USECASE DIAGRAM

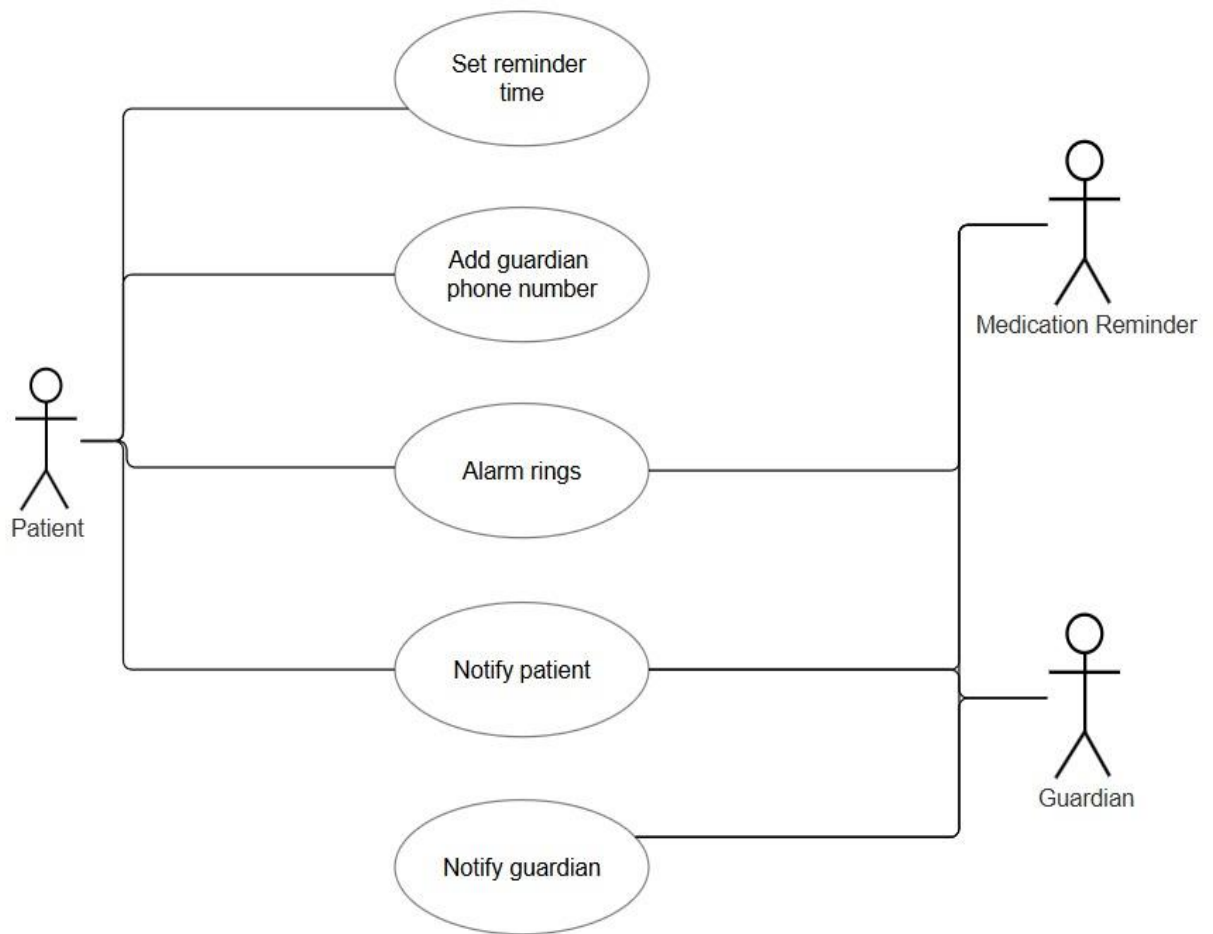


Figure 3.2: Use case Diagram

3.5 FLOWCHART

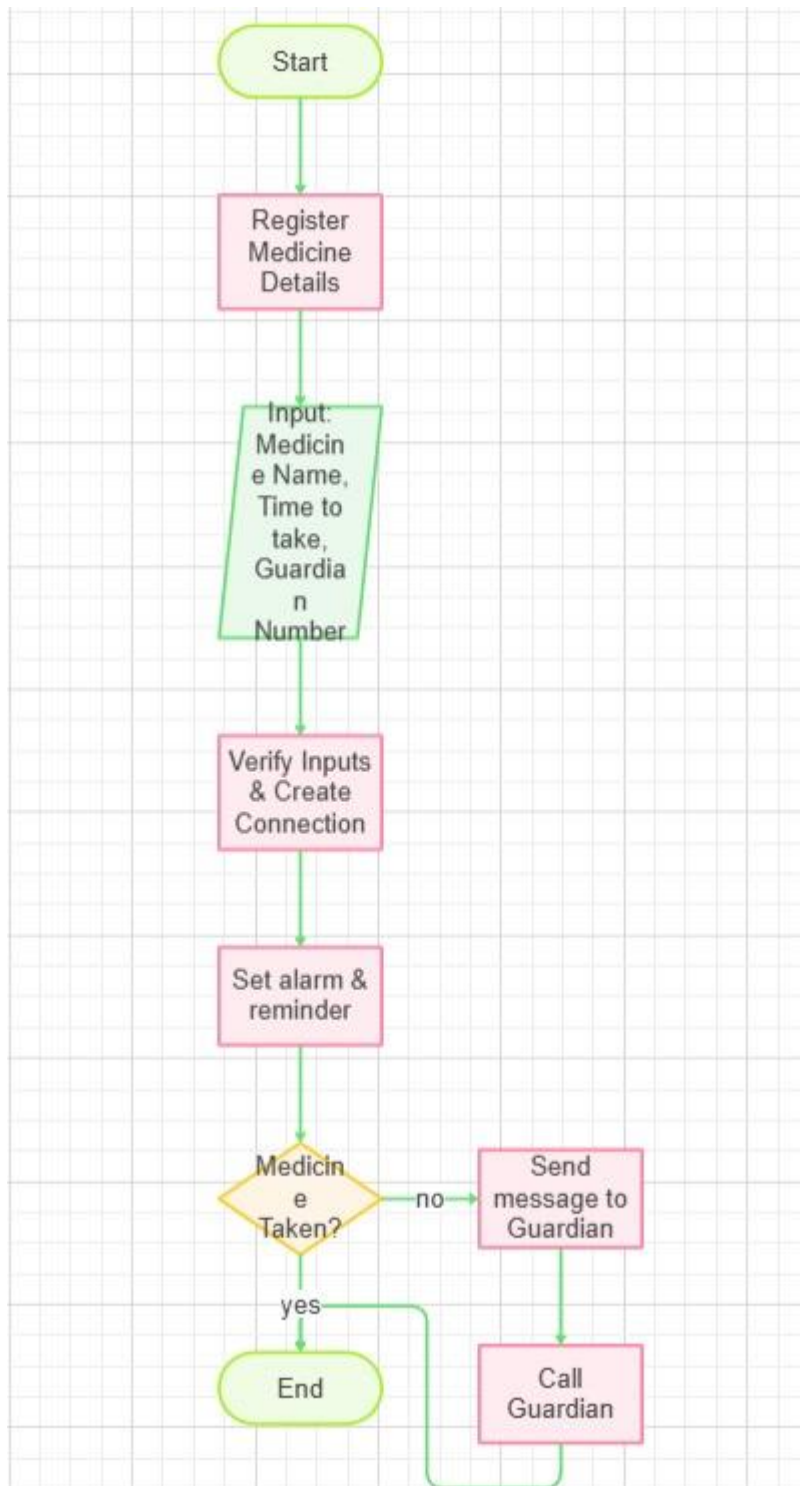


Figure 3.3: flowchart

3.6 ACTIVITY DIAGRAM

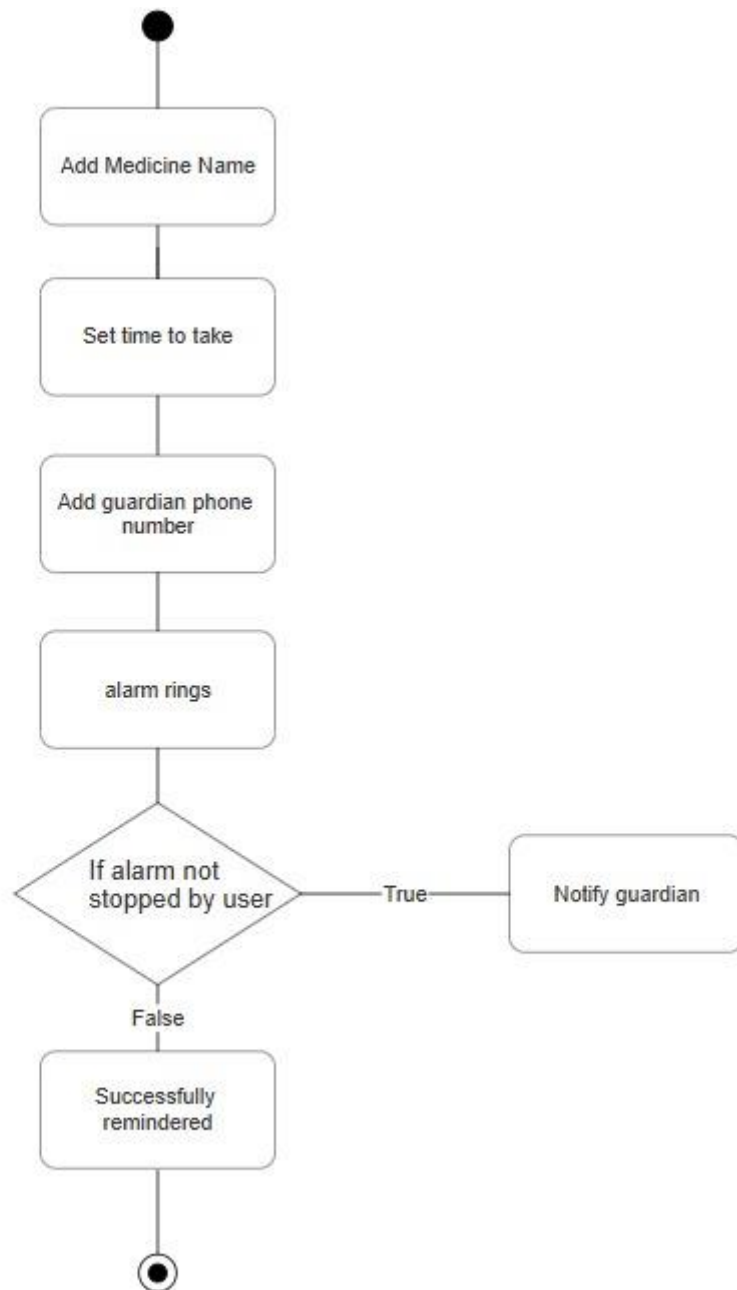


Figure 3.4: Activity diagram

CHAPTER 4

MODULES

4.1 MODULE DESCRIPTION

- Add Medication Reminder
- Alert Received patients
- Alert get caretaker

4.1.1 Add Medication Reminder

The Medicine Reminder Module is a crucial component of healthcare and wellness applications designed to help users manage their medication schedules effectively. This module provides users with features to set up reminders for taking medications, ensuring adherence to prescribed regimens and improving overall health outcomes. The Add Medication Reminder module is designed to help users effectively manage their medication schedules and ensure adherence to treatment plans. It allows users to easily input medication details such as name, dosage, frequency, and time of intake, with support for multiple medications and separate schedules. The module features customizable alerts through sound, text, or push notifications, with options for recurring reminders based on daily or specific-day schedules. Users can snooze or reschedule missed reminders and integrate notifications with calendars or smart devices for added convenience.

The interface is user-friendly, catering to individuals of all ages, and includes a visual dashboard that displays upcoming medications and compliance history. Additional features include the ability to add medication details such as food restrictions, special instructions, or notes, and even upload photos of medications for easy identification. For caregivers or families, the module supports multi-user functionality, enabling shared reminders or alerts for dependents. To enhance health management, it also tracks adherence over time, providing insights into medication compliance and syncing with health monitoring apps for a more holistic approach. This module is ideal for

individuals managing chronic conditions, caregivers assisting others, or anyone seeking a reliable way to maintain their health routines.

4.1.2 Alert Received Patients

Receiving an alert about a patient module related to Alzheimer's likely the patient update medicine details taken or ignore. The Alert Received Patients module is designed to notify patients promptly when important alerts or updates are available. This feature ensures effective communication between healthcare providers and patients, enhancing patient engagement and response to critical information. Alerts can include reminders for upcoming appointments, medication refills, lab results, health warnings, or other time-sensitive notifications. The module supports various communication channels such as in-app notifications, SMS, email, or voice calls, ensuring messages reach patients in their preferred format.

The interface is intuitive, allowing patients to view, acknowledge, and act on alerts quickly. Customization options enable patients to prioritize alerts or set preferences for the types of notifications they wish to receive. Additionally, the module can track whether alerts have been read or acknowledged, providing healthcare providers with insights into patient responsiveness. This feature is ideal for maintaining continuous communication, improving patient compliance, and fostering proactive health management.

4.1.3 Alert Get Caretaker

The Alert Send Module is a vital component of communication systems and applications, facilitating the efficient dissemination of alerts, notifications, and emergency messages to designated recipients or user family member. If the user is somewhere else, the family member receives an alert message. The Get Alert Caretaker module is designed to ensure that caretakers stay informed about critical updates related to the individuals under their care. This module enables real-time notifications for events such as missed medications, upcoming appointments, health emergencies, or significant changes in the patient's

condition. Alerts are delivered through multiple channels, including in-app notifications, SMS, email, or voice calls, ensuring caretakers receive timely information regardless of their location.

Caretakers can customize the types of alerts they receive and prioritize notifications based on urgency. The module includes an intuitive dashboard for tracking and managing alerts, with options to acknowledge and act on them promptly. It can also integrate with patient profiles to provide context, such as the patient's medical history or specific care requirements. By facilitating immediate communication and enabling swift action, this module supports proactive caregiving, improves coordination, and ensures better health outcomes for those under care.

CHAPTER 5

SYSTEM SPECIFICATION

5.1 SOFTWARE REQUIREMENTS

- Operating system : Windows XP, Windows 7
- Front End : Android
- Tool : Android Studio

5.2 HARDWARE REQUIREMENTS

- Processor : Dual core processor 2.6.0 GHZ
- RAM : 4GB
- Hard disk : 320 GB
- Compact Disk : 650 Mb
- Keyboard : Standard keyboard

5.1.1 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)

5.1.2 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.

5.1.3 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-D and 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 can reuse other Android component to achieve 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.

Android Studio 1,3.2

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++ and NDK support– Built-in support for Google Cloud Platform, making it easy to integrate Google Cloud Messaging and App Engine.

CHAPTER 6

CONCLUSION AND FUTURE ENHANCEMENT

6.1 CONCLUSION

In summary, the healthcare app is a shining example of innovation in the field of medical technology, providing a thorough answer to a wide range of issues that both patients and healthcare professional's encounter. Its user-friendly design and wide range of functions, such as health tracking and prescription reminders, promise to completely change how people interact with their health. The software has the potential to greatly enhance patient outcomes and healthcare accessibility by enabling remote consultations and monitoring, especially for marginalized populations or those with restricted mobility. Furthermore, it instills confidence in patients and healthcare professionals by adhering to strict security policies and regulatory regulations that guarantee the confidentiality and integrity of user data. The app is critical step towards a future where technology seamlessly integrates with healthcare, enabling people to take charge of their well-being and promoting a healthier society overall, even though it acknowledges potential obstacles like barriers to digital literacy and the need for constant updates.

6.2 FUTURE ENHANCEMENT

Medication reminders aim to make it more intuitive, user-friendly, and impactful in promoting medication adherence. AI-powered insights could analyse patterns, predict side effects, and flag potential interactions, while integration with voice assistants like Alexa or Siri would allow hands-free reminders. Syncing with smart devices, such as wearables or smart pill dispensers, would enable real-time tracking and notifications. Geolocation-based reminders could trigger alerts based on the user's location, like when they arrive home or near a pharmacy. Enhanced features might include automated refill alerts and pharmacy integration for seamless medication reordering.

Caregiver connectivity could allow family members or healthcare providers to monitor adherence remotely and receive missed-dose notifications. The module could also integrate with health tracking systems to combine medication data with vital signs, offering detailed reports for healthcare providers. Multilingual support would enhance accessibility, while gamification features like rewards or progress tracking could motivate consistent use. Additional upgrades might include emergency alerts for critical missed doses, offline functionality to ensure reminders work without internet, and AI-driven personalized schedules based on the user's routine. These enhancements aim to create a comprehensive, efficient, and reliable system for managing medications.

APPENDIX – A

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_dropd
own_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\nPhasellus 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 – B

SCREENSHOTS

Sample Output

The screenshot shows a mobile application interface for adding a medication reminder. The status bar at the top displays the time 10:39, signal strength, 4G network, and 72% battery. The app interface has a light pink background. It features three input fields with placeholder text: "Enter Medication Name", "Enter Reminder Time (HH:mm)", and "Enter Guardian Phone Number". Below these fields is a dropdown menu currently set to "Once". At the bottom of the form are two purple buttons: "Add Reminder" and "Stop Alarm". The Android navigation bar is visible at the very bottom.

Figure 6.B.1: Add medication Reminder



Figure 6.B.2: Reminding Caretaker



Figure 6.B.3: Alert caretaker via phone call

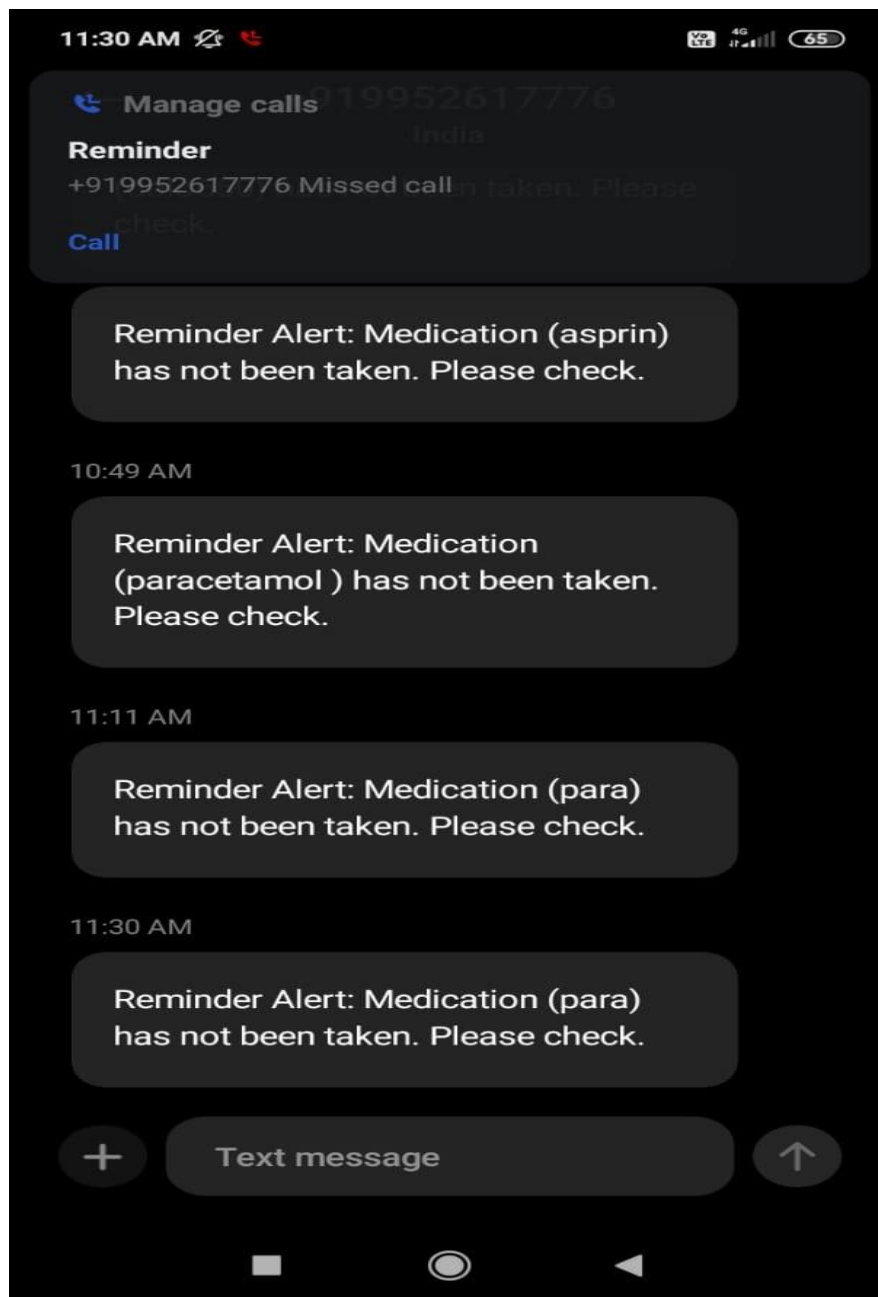


Figure 6.B.4:Alert caretaker via SMS

REFERENCES

- [1] Park, Kee Hyun & Lim, Seung Hyeon, (2012) “Construction of a Medication Reminder Synchronization System based on Data Synchronization”, International Journal of Bioscience and Biotechnology, Vol.4, No. 4, pp1-10.
- [2] "Smartphone medication adherence apps: Potential benefits to patients and providers", available at:
<http://www.ncbi.nlm.nih.gov/pmc/articles/PMC3919626/>
- [3] Slagle, J.M., Gordon, J.S., Harris, C.E., Davison, C.L., Culpepper, D.K., Scott P. and Johnson, K.B., (2011) “MediaMath – Designing a next generation system for child-cantered medication management”, Journal of Biomedical Informatics, Vol. 43, No. 5, pp. 27-31.
- [4] Becker, E., Metis, V., Arora, R., Vinum, J.K., Xu, Y. and Macedon, F. (2009) “Smart Drawer: RFID- Based smart medicine drawer for assistive environments”, Proc. of Pervasive technologies related to assistive environment, June, pp 1-8.
- [5] Amours, S. and Bilodeau, G.A. (2008) “Face and hands detection and tracking applied to the monitoring of medication intake”, Proc. of Canadian Conf. on Computer and Robot Vision, May, pp. 147-154.
- [6] Batz, D., Batz, M., Lobo, N.D.V. and Shah, M. (2005) “A computer vision system for monitoring medication intake”, Canadian Conf. on Computer and Robot Vision, May, pp. 362-369.