UTTARA UNIVERSITY

MEDICATION ALERT – AN ANDROID APPLICATION

BY

FAHAMIN	**ID 2161081067**	**38thB (DAY)**
RATUL ALI	**ID 2161081002**	**38thB (DAY)**
ISMOT ARA LIPI	**ID 2161081085**	**38thB (DAY)**

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
SCHOOL OF SCIENCE AND ENGINEERING
UTTARA UNIVERSITY

FALL 2019

UTTARA UNIVERSITY

MEDICATION ALERT – AN ANDROID APPLICATION]

BY

FAHAMIN	**ID 2161081067**	**38thB (DAY)**
RATUL ALI	**ID 2161081002**	**38thB (DAY)**
ISMOT ARA LIPI	**ID 2161081085**	**38thB (DAY)**

A project submitted in partial fulfillment of the requirement for the degree of
Bachelor of Science in Computer Science and Engineering

Department of Computer Science and Engineering
School of Science and Engineering
Uttara University, Uttara, Dhaka, Bangladesh

Fall 2019

ABSTRACT

This is an android application software that alerts users with an automatic alarming system for medication and dosages. Most of the time patients cannot remember the exact time when they need to take the medicine. This app will alert patients about the time of medication, description of medicine, before-after meal, and so on. The alarm can be set with different types of pictures with the name of the medicines to overcome any kind of difficulty in reading. To develop this project, an incremental development strategy is followed to support customer requirements. Android Studio IDE, Java, and XML tools will be used for the implementation of the project. The system focuses on easy navigation and a good user interface. Many systems have been developed where new hardware and data storage is required but, in our work, we have made an attempt to develop a system that is economical, time-saving and supports medication adherence.

Keywords: *Medication alert, Notification system, Android application, Automatic alarming system, Time schedule alarm, Medication adherence.*

APPROVAL

We certify that we have supervised this project and read this manuscript. In our opinion, it conforms to acceptable standards of scholarly presentation and is fully adequate, in scope and quality, as a report for the degree of BSC in Computer Science and Engineering.

A.H.M. Saifullah Sadi
Supervisor

Md. Mijanur Rahman
Co-Supervisor

I certify that I have read this study. In my opinion, it conforms to acceptable standards of scholarly presentation and is fully adequate, in scope and quality, as a thesis for the degree of BSC in Computer Science and Engineering.

Md. Torikur Rahman
Coordinator/Internal Examiner

This project report was submitted to the Department of Computer Science and Engineering and is accepted as a fulfillment of the requirement for the degree of BSc. in Computer Science and Engineering.

Md. Mijanur Rahman
Chairman, Dept. of CSE

DECLARATION

We hereby declare that this report is the result of our own investigations, except where otherwise stated. We also declare that it has not been previously or concurrently submitted as a whole for any other degrees at Uttara University or any other institution. We also declare that the formatting of the manuscript is the same as the provided template. We also do not have any objections for the further use of the manuscript as Uttara University has all the rights to update, publish, or conduct further research of the submitted work.

Student Names	Student IDs	Signature	Date
Fahamin	2161081067	_____	_____
Ismot Ara Lipi	2161081085	_____	_____
Ratul Ali	2161081002	_____	_____

UTTARA UNIVERSITY

DECLARATION OF COPYRIGHT AND AFFIRMATION OF FAIR USE OF UNPUBLISHED RESEARCH

MEDICATION ALERT – AN ANDROID APPLICATION

We declare that the copyright holders of this dissertation are jointly owned by the students and Uttara University (UU).

1. Any material contained in or derived from this unpublished research may be used by others in their writing with due acknowledgment.

2. UU or its library will have the right to make and transmit copies (print or electronic) for institutional and academic purposes.

3. The UU library will have the right to make, store in a retrieved system and supply copies of this unpublished research if requested by other universities and research libraries.

By signing this form, I acknowledged that I have read and understood the UU Intellectual Property Right and Commercialization policy.

Affirmed by Fahamin

…… ……..
Signature (on behalf of the team) Date

We dedicate this project to
our honorable parents
for their meticulous support, continuous inspiration, and unconditional love
till the very end of this journey.

ACKNOWLEDGMENTS

In performing our project, first of all, deepest gratitude to Allah who gives us the ability to do it. Then, we had to take the help and guideline of some respected persons, who deserve our greatest gratitude. The completion of this project gives us much pleasure. We would like to show our gratitude to Md. Mijanur Rahman Chairman, Dept. of CSE, Assist. Prof. Dr. Md Akhtaruzzaman and Uttara University for giving us a good guideline for this project throughout numerous consultations. We would also like to expand our gratitude to all those who have directly and indirectly guided us in this project.

After that, it is our utmost pleasure to dedicate this work to our dear parents and my family, who granted us the gift of their unwavering belief in our ability to accomplish this goal: thank you for your support and patience.

We wish to express our appreciation and thanks to those who provided their time, effort and support for this project. To the members of our dissertation committee, thank you for sticking with us.

Many people, especially our classmates and team members themselves, have made valuable comment suggestions on this proposal which gave us the inspiration to improve our project.

Finally, a special thanks to ***Associate Prof. Dr. A.H.M. Saifullah Sadi*** for his continuous support, encouragement, and leadership, and for that, we will be forever grateful.

TABLE OF CONTENT

LIST OF TABLES

LIST OF FIGURES

LIST OF SYMBOLS/ACRONYMS

FYP	Final Year Project
ME	Medication Alert
MAS	Medication Alert System
GSM	Global System for Mobile
RTC	Real Time Clock
SMS	Short Message Service
CSE	Computer Science and Engineering
AM	Agile Model
UCD	Use Case Diagram
AA	Android Application

CHAPTER ONE
INTRODUCTION

1.1 OVERVIEW

Most of the times patients may forget to take the medicines at the proper time as per the specified in the prescription which may cause late recovery from the disease/illness. So it is necessary to take proper medicines at the proper time. In this paper, we introduce an Android-based application for the patients. This application will remind their users to take proper medicines in proper time by automatically setting reminders on the mobile. These reminders will be automatically set by the application as per the prescription. This reminder will remind their user-patient that now it's time to take the medicine.

So we are introducing an Android application whose objective is to remind the patients of their dosage timings through the Alarm Ringing system so that they can stay fit and healthy. Through this application, patients can set the alarm by using their Android phone and they can complete the dose at a specific time by using the alarm. As well as they can add a lot of tablets, capsules, etc. to the drug category and can also add photos to them for those who can't read or say. Medication alerts will help in decreasing the wrong dosages. In this paper, we are introducing an Android application for the patients which will remind their users to take proper medicines at the proper time by setting the reminders in the application. In our developing application dependent especially smartphones. Today everyone has a smartphone. So it's more beneficial for us. The application alerts on Android Studio. It is a life-

saving and time-saving application that is easy to use and provides a good user interface.

1.2 PROBLEM STATEMENTS AND ITS SIGNIFICANCE

Proper and timely medical care is one of the major factors for a healthy life. Therefore, the development of the android-based health-care application can assist and facilitates patients to keep themselves healthy. The present time is full of busy times. There is no end to the work here and there is no end to the responsibility and duties. We are getting sick very easy enough to complete these tasks.

Since medicine is very important to stay healthy but in many cases, we forget to take medicines. Especially in the old age, almost everyone has to face this problem. As most computational neuroscientists tend to estimate that human being storage capacity somewhere between 10 terabytes to 100 terabytes.

Though the full spectrum of guesses ranges is from 1 terabyte to 2.5 petabytes. So, if we have absorbed a lot of data from our capacity, then we slowly forget some of the past data. Existing applications lack many important features needed on a daily basis. This is what we address in this work.

- The proper time to take medication
- Patients who are confused about their medicine
- Taking medication on schedule is an issue

1.3 PROJECT OBJECTIVES

Now the basic features for our mobile android application of Medication Alert are Input values, Data storage, and Multimedia support e.g. pictures, etc. We develop a self-help tool in which we integrate this app.

Our application provides some modules where it will add a lot of tablets, capsules, etc. to the drug category and can also add photos to them for those who can't read or say. Medication alerts will help in decreasing the wrong dosages. The goal of the proposed project is to design and develop an android application for Medication Alert. The goal of this project can be split into the following objectives:

- To study related systems and analyze the necessary requirements for the initial design of the Medication Alert android application
- To design user-friendly GUI for the application of Medication Alert
- To implement the Medication Alert application through coding and testing

1.4 PROJECT METHODOLOGY

By developing the Medication Alert application system, a full understanding of the current workflow is required. We conducted a mixed-methods study in some Phases. A comparative study of existing mobile applications developed its model and developed the initial version of Medication Alert that was approved.

The Medication Alert framework is designed to simplify the reuse and integration of components. Medication Alert app is built using a mandatory XML manifest file. The manifest file values are bound to the application at compile time. This file provides

essential information to an Android platform for managing the life cycle of an application. Examples of the kinds of information included in a manifest file are descriptions of the app's components among other architectural and configuration properties. Components can be one of the following types: Activities, Services, Broadcast Receivers, and Content Providers.

Medication reminders help in decreasing medication dispensing errors and wrong dosages. The reminder system consists of two parts –setting the Alarm and getting the notification.

For developing the Medication Alert application system here will be followed the agile model.

- Research
- Planning
- Design
- Coding
- Testing
- Deployment

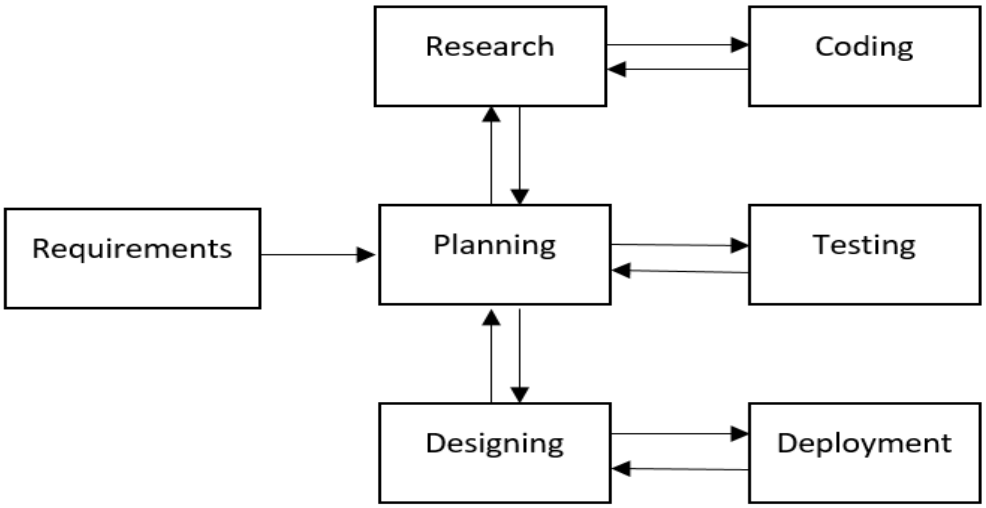

Figure1.1 Incremental Development Model

1.5 GANTT CHART AND PROJECT MILESTONE

By the execution of the Medication Alert app, we started to research it on 15 Jan 2019. After covering some publications and literature review, then we finished Planning, Designing, Coding according to 05 Mar 2019, 01 May 2019, 29 Jul 2019. By completing implementation and execution, we tested it over 100 people. Gathering a positive rating, finally, we deployed a Medication Alert Android Application.

Project Activities			2019											
			Jan	Feb	Mar	Apr	May	Jun	Jul	Aug	Sep	Oct	Nov	Dec
1	Iteration	Research												
2		Planning												
3		Designing												
4		Coding												
5		Testing												
6		Deployment												

Figure1.2 Gantt chart of the proposed project Activity

Milestones	Dates
▼ 1: Completion of Research	15 Jan 2019
▼ 2: Completion of Planning	05 Mar 2019
▼ 3: Completion of Designing	01 May 2019
▼ 4: Completion of Coding	29 Jul 2019
▼ 5: Completion of Testing	01 Nov 2019
▼ 6: Completion of Deployment	02 Dec 2019

Figure 1.3 Project milestones and dates

1.6 SUMMARY AND OUTLINE

The aim of this project is to design and develop a medication alarming system of an android mobile application. Many Medication Reminder Systems have been developed on different platforms. Many of these systems require special hardware devices to remind the patients about the medicine in-take timings. Purchasing new hardware devices becomes costly and more time and money consuming. So in the given work, an attempt has been made to implement a system that is economical, easily accessible and improves medication adherence. Medication non-adherence reduces the effectiveness of treatment and imposes a financial burden on health care systems. We will plan to focus on improving the overall performance of the system.

CHAPTER TWO
BACKGROUND STUDY

2.1 OVERVIEW

The vision of the Medication Alert application is to ensure the healthy life of the patients. By researching it many papers are available on it. By combining all these Medication Alert applications has upgraded with the full system of latest technologies. We proposed a system that includes the latest technologies which are available in it.

2.2 BACKGROUND STUDY

By searching Medication Alert mobile applications and their related articles to identify features of the applications. We analyzed the content of some mobile applications because of the easy installation and availability. In order to assess the validity of the comparative features and attributes, research-made questions were developed on the basis of the selected entities.

Using Android Studio, we developed the initial version of the physical model for Medication Alert mobile application according to the findings of a comparative and conceptual model. The developed application would be installed on the Android operating system version.

2.2.1 Previous Studies

Deepti Ameta, Kalpana Mudaliar, and Palak Patel, (2015) conducted a study on MEDICATION REMINDER AND HEALTHCARE – AN ANDROID APPLICATION. It is an Android-based application in which an automatic alarm ringing system is implemented. It focuses on the doctor and patient interaction. Patients need not remember their medicine dosage timings as they can set an alarm on their dosage timings. The alarm can be set for multiple medicines and timings including date, time and medicine description. A notification will be sent to them through email or message inside the system preferably chosen by the patients. They can search doctor disease wise. The patients will get the contact details of doctors as per their availability.

Shivani Sharma, Katyayni Tyagi, and Pooja Shishodia, (2018) did a study on Salubrity – A medicine reminder application using android. This is a beneficial mobile application that makes people's everyday life easier. This paper focuses on the development of a mobile application to help to provide an effective health care system. In which alarm is used which may be closed by tapping the close alarm button, under the image of the medicine which is to be taken at that particular time. It may even have the contact numbers of the doctors for an emergency. This application will be helping hand for the people who are busy in their day to day life or old age people who forget which medicine is to be taken and when.

Santo K, Chow CK, Thiagalingam A, (2017) conducted a study on MEDication reminder APPs to improve medication adherence in Coronary Heart Disease. This study aims to determine the impact of medication reminder apps on adherence to a

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.

Tabi K, Randhawa AS, Choi F, Mithani Z, Albers F, Schnieder M, Nikoo M, Vigo D, Jang K, Demlova R, Krausz M (2019) conducted a study on Mobile Apps for Medication Management. It provides users with effective mobile health solutions, which can be expected to improve their engagement in the treatment process and long-term well-being. This study also highlighted the need for improved standards for reporting on app stores. Furthermore, it underlined the need for a platform to offer health app users an ongoing evaluation of apps by health professionals in addition to other users and to provide them with tools to easily select an appropriate and trustworthy app.

Ashley Choi*, Annesha White Lovett, Jinhyang Kang, KyungMi Lee, Lydia Choi, (2015) conducted a study on Mobile Applications to Improve Medication Adherence: Existing Apps, Quality of Life and Future Directions. This study provides beneficial information for patients and primary caregivers, as well as pharmacists and other health care providers. Specifically for patients and primary caregivers, current applications contribute to improve patient's adherence with medication by many different features including administration alarms, 68 Mobile Applications to Improve Medication Adherence: Existing Apps, Quality of Life and Future Directions drug information search, and medical records access.

2.3 SUMMARY

Initial purposes of the improving medication safety Program included. We develop and implement a Medication Alert System (MAS) that uses linked data from patient information to identify and alert for patient medication.

CHAPTER THREE
SYSTEM ANALYSIS & DESIGN

3.1 OVERVIEW

For developing this android application, we will use an incremental development model that is the fundamental strategy of the agile development model. It will be a process that helps teams provide quick and unpredictable responses to the feedback they receive on their projects. It creates opportunities to assess a project's direction during the development cycle.

3.2 ANALYSIS AND DESIGN

People are inventing something new from the primitive age to live well in life. We have now come to the age of information technology from that ancient era. By analyzing this, Medication Alert an android application’s model is the perfect model to cure the patients. The design of this app represents the probability of the easiest way to use it for users.

After studying and analyzing all the above existing popular applications based on Android mobiles, some major findings noticed which is very important. Findings in existing systems:

- Users have to enter the name of the tablet/capsule manually every time.
- Users have to enter the quantity/dose of the tablet/capsule manually every time.

- Users have to enter the reminder about the times of dosage manually i.e. 2 or 3 times in a day.
- Users have to manually select the duration of the remainder.
- They are not facilitating anything regarding the original prescription.
- They are not facilitated to capture the medicine words by using a camera.

Everything needs to be done manually. We need an app that can reduce time. Also, the existing systems have some major drawbacks. Those drawbacks are as follows:

- Reminders cannot be set automatically. There is a need for manual work in setting the reminder.
- A lot of time is consumed in manually setting reminders.
- They don't facilitate storing the original prescription.
- The possibility exists for the existing systems to hang down due to not reducing the time.

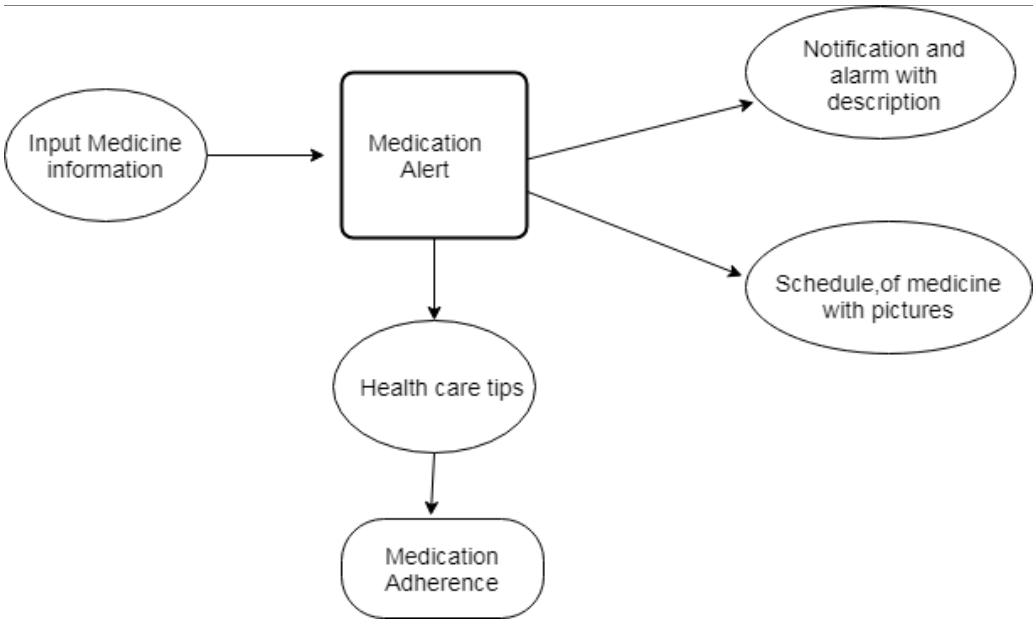

Figure 1.4 Basic designing diagram of Medication Alert Application

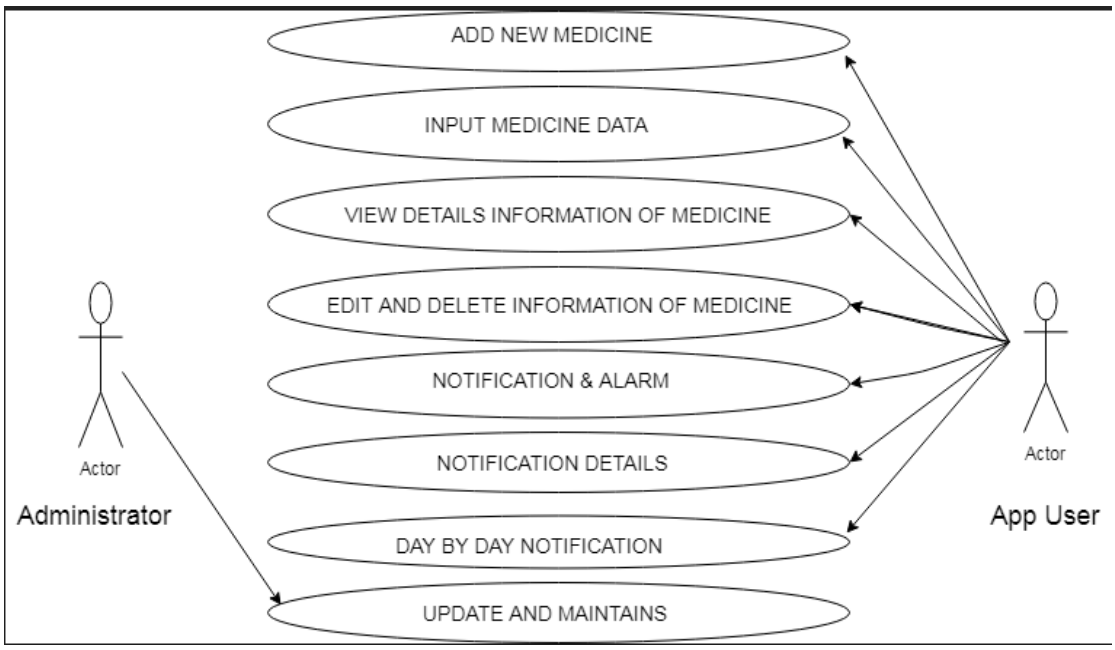

Figure 1.5 Use case diagram

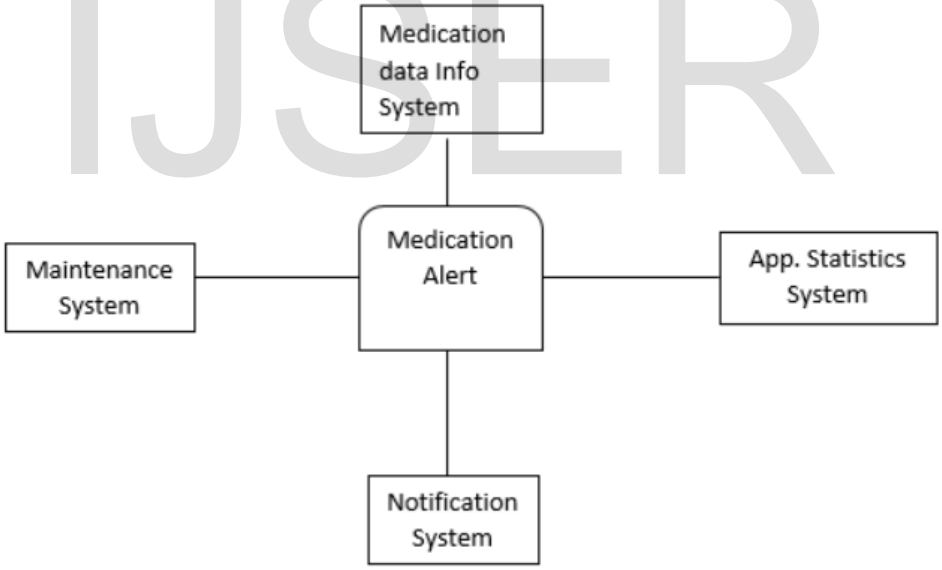

Figure 1.6 The context diagram of Medication Alert

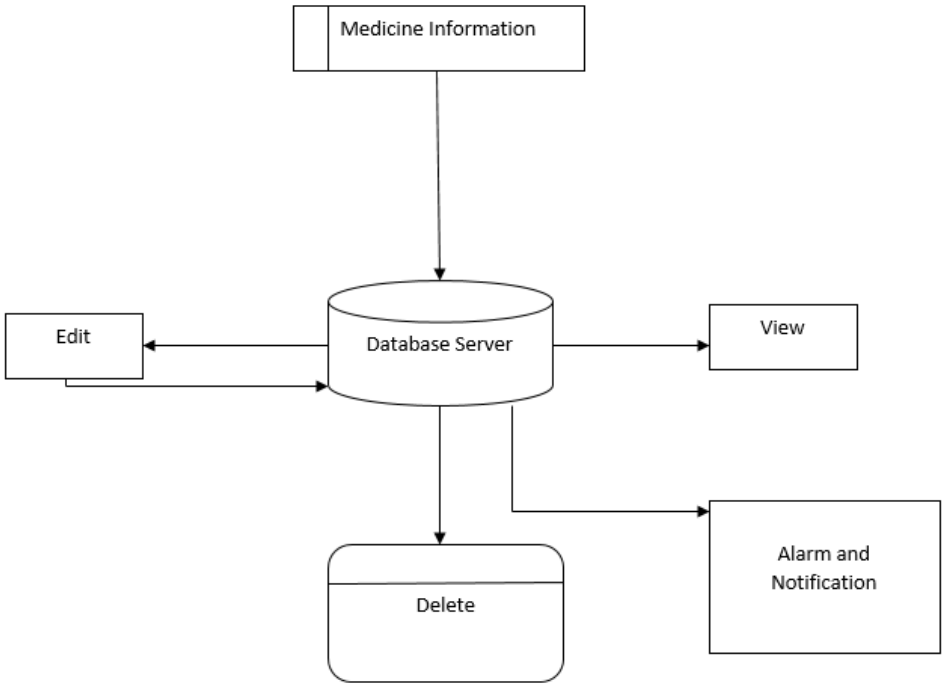

Figure 1.7 Data flow diagram of Medication Alert

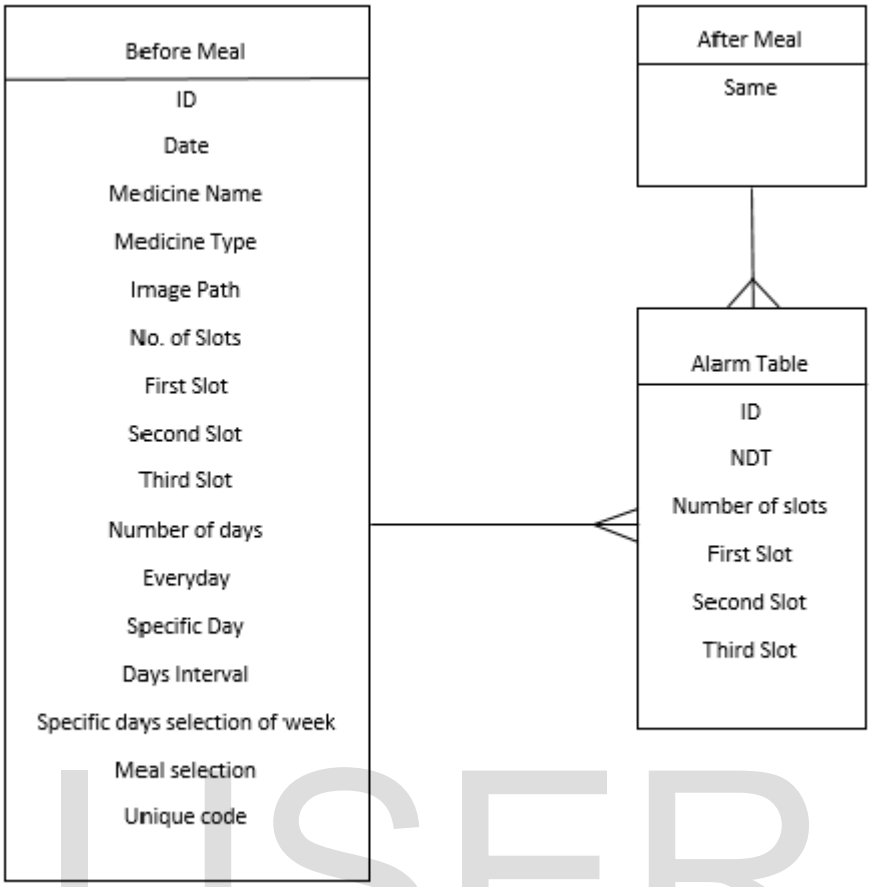

Figure 1.8 ER diagram of Medication Alert

3.2.1 Splash Screen, Startup Pages

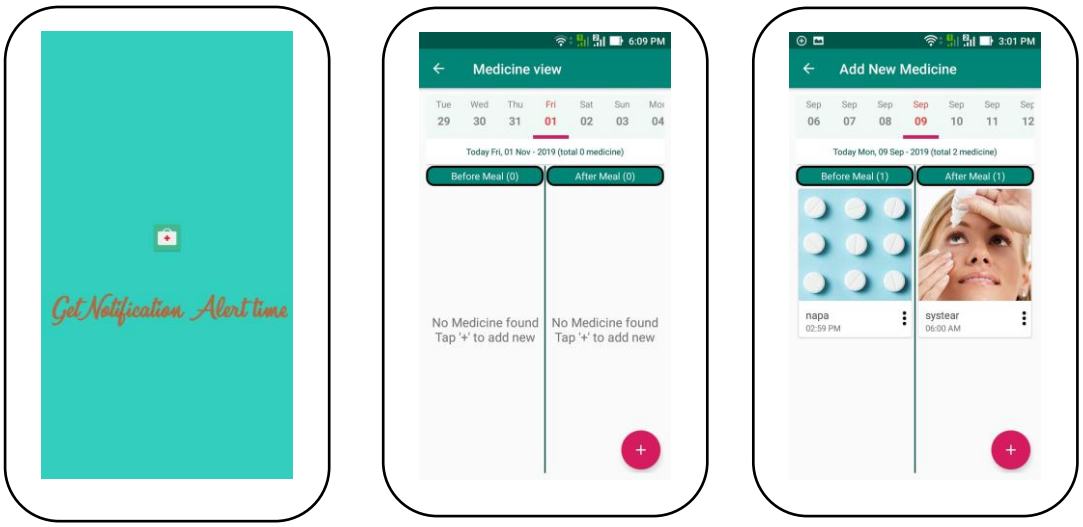

Figure 1.9 Splash Screen, Startup pages of Medication Alert

3.2.2 Time Slots Pages

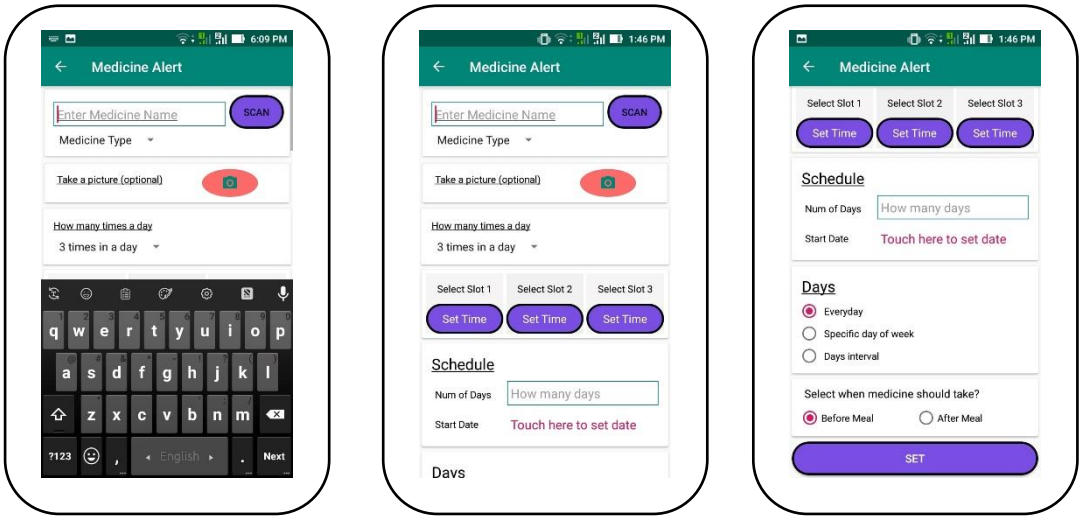

Figure 2.0 Time Slots Pages of Medication Alert

3.2.3 Scheduled Pages

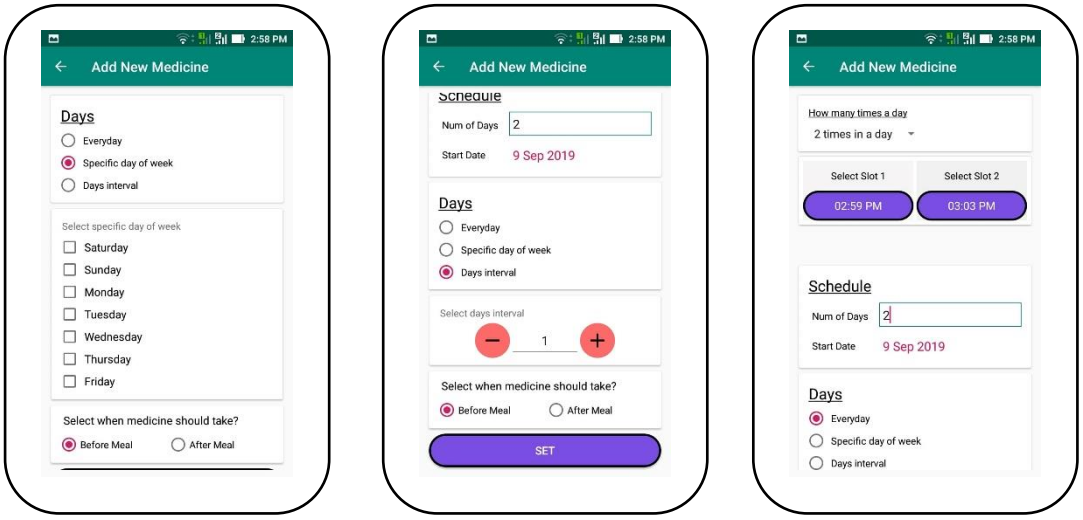

Figure 2.1 Scheduled Pages of Medication Alert

3.2.4 Capturing Words

Figure 2.2 Capturing Words

3.3 SUMMARY

The system helps people who forget to take their medications on time. Aged people usually tend to forget their medication timings. This project is designed to help such people take their medicines on time.

The system has the ways that alert in order to ensure people take the right medicine at the right time. The system alerts the user by sounding for that a medicine to be taken at that particular time and also displays the pictures of medicine to be taken.

In the future, An RTC is interfaced with a microcontroller in order to alert the patient on time. The system can later be improved by interfacing system with a GSM device which will send an SMS alert to the personal mobile phone at the time of taking medicine.

CHAPTER FOUR
SYSTEM SETUP, IMPLEMENTATION, & TESTING

4.1 OVERVIEW

Completing total arrangements of Medication Alert Android Application (AA), an incremental development strategy was chosen to develop this AA. Java testing IDE Android Studio used for generating codes. The final test has taken over 100 people for measuring the importance and usefulness of this application.

4.2 SYSTEM SETUP

The state of the Medical Compliance application that has been continually referred to is maintained by the set of model classes referred to as the model. The model portion of this application is kept completely separate from the view portion, in keeping with the incremental development model design paradigm described in previous sections. This decision allows for easy testing of model classes using conventional Java testing tools/IDE such as Android Studio. In this application, it allows for a much more direct translation of data.

4.3 IMPLEMENTATION

The daily schedule portion of the model is responsible for, maintaining, and updating all compliance events for a given day. When the user chooses to view the daily schedule tab, a list of all of the user's medications is passed to the daily schedule via this model. This method then iterates through this list and determines which compliance schedule objects should be created and added to, or removed from, the day's schedule list. The algorithm first checks to see if the medication has any events

scheduled for the current day. If there are, the algorithm iterates through the list of times and creates a compliance scheduled object containing the proper information. Once a list of new events has been generated, the algorithm checks the current event list for events for medications that no longer exist, and events from previous days. It then checks the new event list for events that have already been scheduled. If an event has not been scheduled, then it is added to the list. Finally, the algorithm sorts the events by time, earliest to latest.

Times for compliance events are stored in the application data by using the simple time object. This decision was made in order to store an entire java calendar or similar object in memory for these events and to simplify the process by which the time objects can be manipulated.

Features of this system are:

- Showing the list of medicines with their dosage at the prescribed time - The user gets a notification in his device at the time of taking a medicine, and that notification contains the list of medicines to be taken along with their respective dosage
- Prescription Duration - The duration for which the patient has to take the medicine
- Number of Medicines - Field is mandatory to generate further options
- Names of the Medicines - To keep track of what medicines are taken by the user and to remind the user later of the medicine to take

- Dosage Time - The time at which to remind the user, whether in morning, evening, night or a combination of the three
- Dosage Quantity - The quantity of dosage to be taken for each medicine
- Capturing Medication Words - By using the camera it will capture the words and will fill the name of the medicine

4.4 TESTING

This medicine reminder system serves reliable reminders, has a good and easy to use user interface and supports a lot of features adhering to medicines. The details are not at all confusing and can be easily understood by the user. The best part of the application is that the details only have to be entered one time. On submitting the details once, the data is synced on all the user's devices. This allows for easy reminders no matter what kind of Android device the user is using. The reviews on the system are overall positive.

Table 1.1 Testing Table for Medication Alert

Testing	Errors	Bug	Console Ok
Debug	No	No	Yes
Release	No	No	Yes
Alpha	No	No	
Beta	No	No	
Production	No	No	

4.5 SUMMARY

An application that leverages mobile technology to address the issues and costs attributed to patient noncompliance has been developed and presented. The smartphone application is the first step in a solution distributable to individuals on a large scale. This application provided an intuitive means of use despite the complicated nature of prescriptions and medication regimens. The application was developed to require minimal user input with features like the camera scanning for capturing words.

CHAPTER FIVE
RESULT ANALYSIS & BENCHMARKING

5.1 OVERVIEW

Gathering results analyzing of the collecting data rates proves the positiveness then benchmarking process comparison move together with some of the tops ranked android applications.

5.2 RESULT PRESENTATION

We did a survey on approximately 100 people and asked them to download and use the app for their medicinal needs and see if it helps them. According to the gathered report, 80% of the users said that the app actually was helpful in reminding them about their medications and would love to use the app. 20% said that they would like further improvements in the app which would be very helpful to them. Thus we intend to improve the app and support as many as we can.

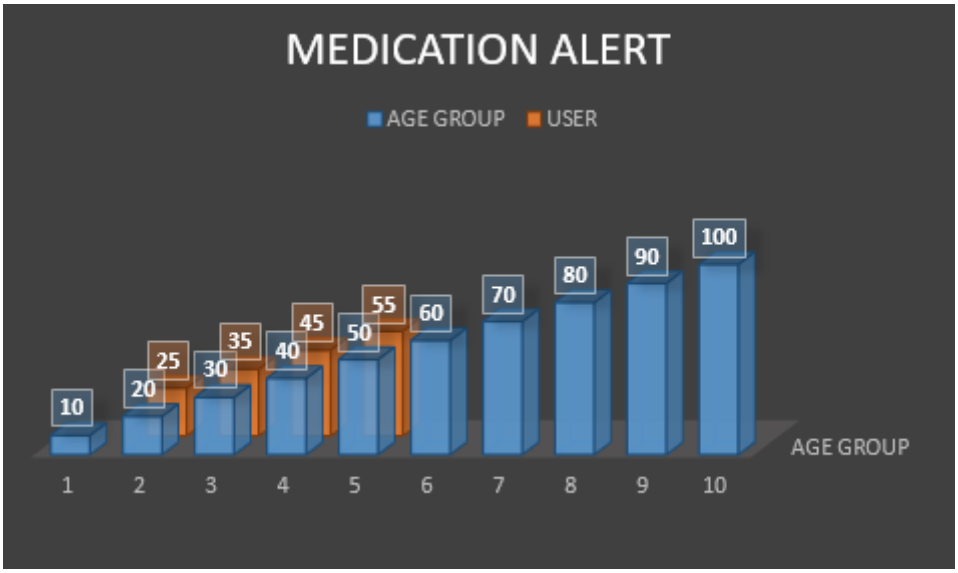

Figure 2.3 Result Representation diagram for Medication Alert

5.3 DISCUSSIONS

Observing critical analysis of all data that remarks on the necessity of the usefulness of medical adherence. Necessary comparisons among the results build up the confidence for making the benchmark.

5.4 BENCHMARKING

Gathering results analyzing of the collecting data rates proves the positiveness then benchmarking process comparison move together with some of the tops ranked android applications.

Table 1.2 Comparison of the App with three existing Apps

Name	Medication Reminder & Pill Tracker	Pill Reminder and Medication Tracker by Medisafe
File Size	18 MB	34 MB
Product	Free	Included products carry the cost
Requires Android	4.4 and up	Varies with device

5.5 SUMMARY

The Medication Alert reminder system is a useful resource for those who need technological help in completing or need help in working through day-to-day tasks and taking care of their health. It is a smart and organized system that is designed with helping the elderly people in our homes, but we have not put any restrictions that stop an everyday user from using the system. Anyone can need medical attention and normal people forget taking their prescriptions as well. The Medication Alert features will help them out in regulating their medications. It can also help a working person with a busy schedule by reminding him of the device they use a full day. Thus there is no restriction on the user base for our system.

CHAPTER SIX
CONCLUSION& RECOMMENDATIONS

6.1 PROJECT OUTCOMES

Execution of Medication Alert android application provides the necessary outcomes that prove the necessity of this application is as follows:

- Medication after a meal or before a meal and specific time
- Addition of proper information with medicine alert
- Accordingly, day by day days Medication alert notification

6.2 LIMITATIONS OF THE PROJECT

Hard work doesn't make the overall perfect all time that why it has also some limitations that expand in brief.

- Drug-to-drug interaction checker
- Get reminders for all your meds, times and displays your notes such as "take with food"
- Send your friends and family notifications if you want help staying on track if you miss a dose
- Track dozens of measurements - such as blood pressure, weight, and glucose all in one place

- See daily and monthly medication progress reports that you can send to a doctor or nurse ahead of an appointment
- Choose from a library of free medication reminder sounds
- Know you when multiple drugs you've entered may have drug-to-drug interactions to discuss with your caregiver

6.3 RECOMMENDATIONS

Improves medication adherence supports the patients in a long way to stay healthy. Here is the improves medication adherence is as follows:

- The attempts of that project have been made to implement a system that is cost-saving, easily accessible and improves medication adherence.
- In the future, at the end of the dosages, it will suggest the schedule to the users.

REFERENCES

Deepti Ameta, Kalpana Mudaliar, and Palak Patel, (2015). MEDICATION REMINDER AND HEALTHCARE – AN ANDROID APPLICATION. International Journal of Managing Public Sector Information and Communication Technologies (IJMPICT) Vol. 6, No. 2, June 2015.

Shivani Sharma, Katyayni Tyagi, and Pooja Shishodia, (2018). Salubrity - A medicine reminder application using android. International Journal of Advanced Research, Ideas and Innovations in Technology. ISSN: 2454-132X Impact factor: 4.295 (Volume 4, Issue 2) Available online at: www.ijariit.com.

Santo K, Chow CK, Thiagalingam A, et al. MEDication reminder APPs to improve medication adherence in Coronary Heart Disease (MedApp-CHD) Study: a randomized controlled trial protocol. BMJ Open 2017;7:e017540. doi:10.1136/ bmjopen-2017-017540.

Tabi K, Randhawa AS, Choi F, Mithani Z, Albers F, Schnieder M, Nikoo M, Vigo D, Jang K, Demlova R, Krausz M. Mobile Apps for Medication Management: Review and Analysis JMIR Mhealth Uhealth 2019;7(9):e13608 URL: https://mhealth.jmir.org/2019/9/e13608 DOI: 10.2196/13608.

Ashley Choi*, Annesha White Lovett, Jinhyang Kang, KyungMi Lee, Lydia Choi, (2015). Mobile Applications to Improve Medication Adherence: Existing Apps, Quality of Life and Future Directions. College of Pharmacy, Mercer University, United States. Advances in Pharmacology and Pharmacy 3(3): 64-74, 2015 http://www.hrpub.org DOI: 10.13189/app.2015.030302.

APPENDIX A
EXAMPLE CODES

CODES FOR MEDICINE VIEW FRAGMENT

```
package app.cave.medicinalertapp.fragment;


import android.Manifest;

import android.content.Context;

import android.content.Intent;

import android.content.pm.PackageManager;

import android.content.res.Resources;

import android.graphics.Color;

import android.net.Uri;

import android.os.Build;

import android.os.Bundle;

import android.support.design.widget.FloatingActionButton;

import android.support.v4.app.Fragment;


import android.support.v4.content.ContextCompat;

import android.support.v7.widget.DefaultItemAnimator;

import android.support.v7.widget.GridLayoutManager;

import android.support.v7.widget.RecyclerView;

import android.util.TypedValue;

import android.view.LayoutInflater;
```

```
import android.view.View;

import android.view.ViewGroup;

import android.widget.ImageView;

import android.widget.TextView;

import android.widget.Toast;



import java.text.DateFormat;

import java.text.SimpleDateFormat;

import java.util.ArrayList;

import java.util.Calendar;

import java.util.Date;

import java.util.HashMap;

import java.util.List;

import java.util.Locale;

import java.util.Map;



import app.cave.medicinalertapp.StaticVariables;

import app.cave.medicinalertapp.activity.AddMedicineActivity;

import app.cave.medicinalertapp.activity.MainActivity;

import app.cave.medicinalertapp.adapter.MedicineAdapter;

import app.cave.medicinalertapp.classfile.DateCalculations;

import app.cave.medicinalertapp.classfile.GridSpacingItemDecoration;

import app.cave.medicinalertapp.database.MedicineDatabase;
```

```
import app.cave.medicinalertapp.model.MedicineModel;
import app.cave.medicinalertapp.R;
import devs.mulham.horizontalcalendar.HorizontalCalendar;
import devs.mulham.horizontalcalendar.HorizontalCalendarView;
import devs.mulham.horizontalcalendar.utils.HorizontalCalendarListener;


public class Medichine_View_fragment extends Fragment {

    View view;

    FloatingActionButton fab;

    //InterstitialAd mInterstitialAd;

    boolean allPermission;
    RecyclerView recyclerViewBeforeMeal, recyclerViewAfterMeal;
    List<MedicineModel> medicineModelListBeforeMeal,
medicineModelListAfterMeal;
    MedicineAdapter adapter;
    MedicineDatabase dbHelper;

    Calendar startDate;
    Calendar endDate;
```

```
    TextView beforeMessageTV, afterMessageTV;



    HorizontalCalendar horizontalCalendar;
    ImageView leftIV, rightIV;
    TextView dateTV, beforeTV, afterTV;
    boolean firstStart = true;
    int position = 5;


    public Medichine_View_fragment() {
    }


    @Override
    public View onCreateView(LayoutInflater inflater, ViewGroup container, Bundle
savedInstanceState) {

        view = inflater.inflate(R.layout.fragment_medichine_view, container, false);

        if (!allPermission){

            if (Build.VERSION.SDK_INT >= 23) {
                checkMultiplePermissions();
```

```
        }
    }

    setupCalender();
    init();

    return view;
}

private void init() {

    Date currentDate = Calendar.getInstance().getTime();
    DateFormat df = DateFormat.getDateInstance(DateFormat.MEDIUM,
Locale.UK);
    String searchQuery = df.format(currentDate);

    fab = (FloatingActionButton) view.findViewById(R.id.fab_add);
```

CODES FOR ADD MEDICINE FRAGMENT

```
package app.cave.medicinalertapp.fragment;


import android.Manifest;

import android.app.Activity;

import android.app.AlarmManager;

import android.app.AlertDialog;

import android.app.DatePickerDialog;

import android.app.Dialog;

import android.app.DialogFragment;

import android.app.PendingIntent;

import android.app.TimePickerDialog;

import android.content.Context;

import android.content.Intent;

import android.content.SharedPreferences;

import android.content.pm.PackageManager;

import android.graphics.Bitmap;

import android.graphics.drawable.BitmapDrawable;

import android.os.Build;

import android.os.Bundle;

import android.provider.MediaStore;

import android.support.annotation.RequiresApi;

import android.support.v4.app.Fragment;

import android.support.v7.widget.CardView;
```

```
import android.util.Log;

import android.view.LayoutInflater;

import android.view.View;

import android.view.ViewGroup;

import android.widget.AdapterView;

import android.widget.Button;

import android.widget.CheckBox;

import android.widget.CompoundButton;

import android.widget.DatePicker;

import android.widget.EditText;

import android.widget.ImageView;

import android.widget.LinearLayout;

import android.widget.RadioButton;

import android.widget.Spinner;

import android.widget.TextView;

import android.widget.TimePicker;

import android.widget.Toast;


import java.text.DateFormat;

import java.text.ParseException;

import java.text.SimpleDateFormat;

import java.util.ArrayList;

import java.util.Calendar;

import java.util.Date;
```

```
import java.util.HashMap;

import java.util.List;

import java.util.Locale;

import java.util.Map;


import app.cave.medicinalertapp.R;

import app.cave.medicinalertapp.StaticVariables;

import app.cave.medicinalertapp.activity.MainActivity;

import app.cave.medicinalertapp.recicver.AlarmReceiver;

import app.cave.medicinalertapp.classfile.DateCalculations;

import app.cave.medicinalertapp.database.AlarmDatabase;

import app.cave.medicinalertapp.database.MedicineDatabase;

import app.cave.medicinalertapp.model.AlarmModel;

import app.cave.medicinalertapp.model.MedicineModel;

import app.cave.medicinalertapp.server.ImageSaver;


import static android.content.Context.MODE_PRIVATE;



public class Add_Medichine_fragment extends Fragment {

    public SharedPreferences sharedPreferences;


    EditText medNameET, noOfDaysET;

    TextView firstSlotTV, secondSlotTV, thirdSlotTV, startDateTV;
```

```
    Spinner noOfTimesSP, medicineTypeSP;
    RadioButton everyDayRB, specificDayRB, daysIntervalRB, beforeMealRB,
afterMealRB;
    CheckBox cbSaturday, cbSunday, cbMonday, cbTuesday, cbWednesday,
cbThursday, cbFriday;
    LinearLayout firstSlotLAYOUT, secondSlotLAYOUT, thirdSlotLAYOUT;
    EditText etDaysInterval;
    ImageView plusIV, mynasIV, takeSnapIV, medicineIV;
    CardView cvSpecificDayOfWeek, cvDaysInterval, cvMedicineImage;
    Button setBTN, retakeBTN, cancelBTN;

    String formattedTime;
    Calendar myCalender;

    int id, numberOfSlot, noOfDays, daysInterval;
    String medName, imagePath, firstSlotTime, secondSlotTime, thirdSlotTime,
startDate, daysNameOfWeek, status, calculatedDate,
            newStartDate, medicineMeal, medicineType, finalDate;
    boolean isEveryday, isSpecificDaysOfWeek, isDaysInterval;
    boolean sat, sun, mon, tue, wed, thu, fri;
    boolean allPermission;


    String tableName = "";
```

```
    int requestCode = 1;
    int flag = 0;
    int uniqueCode = 0;
    int firstRequestCode, secondRequestCode, thirdRequestCode;

    MedicineDatabase dbHelper;

    View view;


    public Add_Medichine_fragment() {
    }

    @Override
    public View onCreateView(LayoutInflater inflater, ViewGroup container, Bundle
savedInstanceState) {


        view = inflater.inflate(R.layout.fragment_add__medichine, container, false);

        sharedPreferences = getActivity().getSharedPreferences("alarmRequestCode",
MODE_PRIVATE);
        requestCode = sharedPreferences.getInt("requestCodeValue", 1);
        flag = sharedPreferences.getInt("flagValue", 0);
```

CODES FOR ALARM DATABASE

```
package app.cave.medicinalertapp.database;


import android.content.ContentValues;

import android.content.Context;

import android.database.Cursor;

import android.database.sqlite.SQLiteDatabase;

import android.database.sqlite.SQLiteOpenHelper;


import java.util.ArrayList;

import java.util.List;

import app.cave.medicinalertapp.model.AlarmModel;


public class AlarmDatabase extends SQLiteOpenHelper {

   Context context;

   private static final String DATABASE_NAME = "alarm_manager";

   private static final int DATABASE_VERSION = 1;


   private static final String TABLE_NAME = "alarm_table";


   private static final String COLUMN_1 = "ID";

   private static final String COLUMN_2 = "NDT";

   private static final String COLUMN_3 = "NUMBER_OF_SLOT";
```

```
private static final String COLUMN_4 = "FIRST_SLOT_TIME";

private static final String COLUMN_5 = "SECOND_SLOT_TIME";

private static final String COLUMN_6 = "THIRD_SLOT_TIME";

private static final String COLUMN_7 = "FIRST_SLOT_RC";

private static final String COLUMN_8 = "SECOND_SLOT_RC";

private static final String COLUMN_9 = "THIRD_SLOT_RC";


public AlarmDatabase(Context context) {

   super(context, DATABASE_NAME, null, DATABASE_VERSION);

   this.context = context;

}
```



```
@Override

public void onCreate(SQLiteDatabase db) {

   String CREATE_TABLE_QUERY = "CREATE TABLE "

         + TABLE_NAME + "("

         + COLUMN_1 + " INTEGER PRIMARY KEY AUTOINCREMENT ,"

         + COLUMN_2 + " TEXT,"

         + COLUMN_3 + " INTEGER,"

         + COLUMN_4 + " TEXT,"

         + COLUMN_5 + " TEXT,"

         + COLUMN_6 + " TEXT,"

         + COLUMN_7 + " INTEGER,"
```

```
        + COLUMN_8 + " INTEGER,"
        + COLUMN_9 + " INTEGER" + ")";

    db.execSQL(CREATE_TABLE_QUERY);

}

@Override
public void onUpgrade(SQLiteDatabase db, int oldVersion, int newVersion) {
    db.execSQL("DROP TABLE IF EXISTS " + TABLE_NAME);
    onCreate(db);
}

public void insertAlarn(AlarmModel alarmModel) {
    SQLiteDatabase db = this.getWritableDatabase();
    ContentValues values = new ContentValues();

    values.put(COLUMN_2, alarmModel.getNdt());
    values.put(COLUMN_3, alarmModel.getNumberOfSlot());
    values.put(COLUMN_4, alarmModel.getFirstSlotTime());
    values.put(COLUMN_5, alarmModel.getSecondSlotTime());
    values.put(COLUMN_6, alarmModel.getThirdSlotTime());
    values.put(COLUMN_7, alarmModel.getFirstSlotRequestCode());
```

```
    values.put(COLUMN_8, alarmModel.getSecondSlotRequestCode());
    values.put(COLUMN_9, alarmModel.getThirdSlotRequestCode());

    db.insert(TABLE_NAME, null, values);
    db.close();
}

public int updateAlarm(AlarmModel alarmModel) {
    SQLiteDatabase db = this.getWritableDatabase();
    ContentValues values = new ContentValues();

    values.put(COLUMN_2, alarmModel.getNdt());
    values.put(COLUMN_3, alarmModel.getNumberOfSlot());
    values.put(COLUMN_4, alarmModel.getFirstSlotTime());
    values.put(COLUMN_5, alarmModel.getSecondSlotTime());
    values.put(COLUMN_6, alarmModel.getThirdSlotTime());
    values.put(COLUMN_7, alarmModel.getFirstSlotRequestCode());
    values.put(COLUMN_8, alarmModel.getSecondSlotRequestCode());
    values.put(COLUMN_9, alarmModel.getThirdSlotRequestCode());

    return db.update(TABLE_NAME, values, COLUMN_1 + " = ?",
          new String[]{String.valueOf(alarmModel.getId())});

}
```

```
public List<AlarmModel> getAllAlarmList() {
    List<AlarmModel> alarmModels = new ArrayList<>();
    String seletQuery = "Select * FROM " + TABLE_NAME;

    SQLiteDatabase database = this.getReadableDatabase();
    Cursor cursor = database.rawQuery(seletQuery, null);

    if (cursor.moveToFirst()) {
      do {
        AlarmModel alarmModel = new AlarmModel();
        alarmModel.setId(Integer.parseInt(cursor.getString(1)));
        alarmModel.setNdt(cursor.getColumnName(2));
        alarmModel.setNumberOfSlot(Integer.parseInt(cursor.getString(3)));
        alarmModel.setFirstSlotTime(cursor.getString(4));
        alarmModel.setSecondSlotTime(cursor.getString(5));
        alarmModel.setThirdSlotTime(cursor.getString(6));
        alarmModel.setFirstSlotRequestCode(Integer.parseInt(cursor.getString(7)));

alarmModel.setSecondSlotRequestCode(Integer.parseInt(cursor.getString(8)));

alarmModel.setThirdSlotRequestCode(Integer.parseInt(cursor.getString(9)));

        alarmModels.add(alarmModel);
```

CODES FOR MEDICINE DATABASE

```
package app.cave.medicinalertapp.database;


import android.content.ContentValues;

import android.content.Context;

import android.database.Cursor;

import android.database.sqlite.SQLiteDatabase;

import android.database.sqlite.SQLiteOpenHelper;

import android.widget.Toast;


import java.util.ArrayList;

import java.util.List;


import app.cave.medicinalertapp.database.ConstansDatabase;

import app.cave.medicinalertapp.model.MedicineModel;


public class MedicineDatabase extends SQLiteOpenHelper {


   Context context;


   private static final String DATAABASE_NAME = "medicine_alerts";

   private static final int DATABASE_VERSION = 1;


   private static final String BEFORE_MEAL_TABLE = "before_table";
```

```
private static final String AFTER_MEAL_TABLE = "after_table";


private static final String COLUMN_0 = "ID";
private static final String COLUMN_1 = "DATE";
private static final String COLUMN_2 = "MEDICINE_NAME";
private static final String COLUMN_3 = "MEDICINE_TYPE";
private static final String COLUMN_4 = "IMAGE_PATH";
private static final String COLUMN_5 = "NO_OF_SLOT";
private static final String COLUMN_6 = "FIRST_SLOT";
private static final String COLUMN_7 = "SECOND_SLOT";
private static final String COLUMN_8 = "THIRD_SLOT";
private static final String COLUMN_9 = "NUMBER_OF_DAYS";
private static final String COLUMN_10 = "IS_EVERYDAY";
private static final String COLUMN_11 = "IS_SPECIFIC_DAYS_OF_WEEK";
private static final String COLUMN_12 = "IS_DAYS_INTERVAL";
private static final String COLUMN_13 = "DAYS_NAME_OF_WEEK";
private static final String COLUMN_14 = "DAYS_iNTERVAL";
private static final String COLUMN_15 = "START_DATE";
private static final String COLUMN_16 = "STATUS";
private static final String COLUMN_17 = "MEDICINE_MEAL";
private static final String COLUMN_18 = "UNIQUE_CODE";


public MedicineDatabase(Context context) {
   super(context, DATAABASE_NAME, null, DATABASE_VERSION);
```

```
    this.context = context;
}

@Override
public void onCreate(SQLiteDatabase db) {

    String BEFORE_MEAL_QUERY =
        "CREATE TABLE "
            + BEFORE_MEAL_TABLE + "("
            + COLUMN_0 + " INTEGER PRIMARY KEY AUTOINCREMENT ,"
            + COLUMN_1 + " TEXT,"
            + COLUMN_2 + " TEXT,"
            + COLUMN_3 + " TEXT,"
            + COLUMN_4 + " TEXT,"
            + COLUMN_5 + " INTEGER,"
            + COLUMN_6 + " TEXT,"
            + COLUMN_7 + " TEXT,"
            + COLUMN_8 + " TEXT,"
            + COLUMN_9 + " INTEGER,"
            + COLUMN_10 + " BOOLEAN,"
            + COLUMN_11 + " BOOLEAN,"
            + COLUMN_12 + " BOOLEAN,"
            + COLUMN_13 + " TEXT,"
            + COLUMN_14 + " INTEGER,"
```

```
        + COLUMN_15 + " TEXT,"

        + COLUMN_16 + " TEXT,"

        + COLUMN_17 + " TEXT,"

        + COLUMN_18 + " INTEGER" + ")";


String AFTER_MEAL_QUERY = "CREATE TABLE "

    + AFTER_MEAL_TABLE + "("

    + COLUMN_0 + " INTEGER PRIMARY KEY AUTOINCREMENT ,"

    + COLUMN_1 + " TEXT,"

    + COLUMN_2 + " TEXT,"

    + COLUMN_3 + " TEXT,"

    + COLUMN_4 + " TEXT,"

    + COLUMN_5 + " INTEGER,"

    + COLUMN_6 + " TEXT,"

    + COLUMN_7 + " TEXT,"

    + COLUMN_8 + " TEXT,"

    + COLUMN_9 + " INTEGER,"

    + COLUMN_10 + " BOOLEAN,"

    + COLUMN_11 + " BOOLEAN,"

    + COLUMN_12 + " BOOLEAN,"

    + COLUMN_13 + " TEXT,"

    + COLUMN_14 + " INTEGER,"

    + COLUMN_15 + " TEXT,"

    + COLUMN_16 + " TEXT,"
```

```
        + COLUMN_17 + " TEXT,"
        + COLUMN_18 + " INTEGER" + ")";

    db.execSQL(BEFORE_MEAL_QUERY);
    db.execSQL(AFTER_MEAL_QUERY);

}

@Override
public void onUpgrade(SQLiteDatabase db, int oldVersion, int newVersion) {

    db.execSQL("DROP TABLE IF EXISTS " + BEFORE_MEAL_TABLE);
    db.execSQL("DROP TABLE IF EXISTS " + AFTER_MEAL_TABLE);
    onCreate(db);
}

public void insertData(MedicineModel medicineModel, String tableName) {
    SQLiteDatabase db = this.getWritableDatabase();
    ContentValues values = new ContentValues();

    values.put(COLUMN_1, medicineModel.getDate());
    values.put(COLUMN_2, medicineModel.getMedicineName());
    values.put(COLUMN_3, medicineModel.getMedicineType());
    values.put(COLUMN_4, medicineModel.getImagePath());
```