ABSTRACT

In today's life, human beings face difficulty to keep in mind the medicines that they are required to take. This application proposes a model of automatic medicine reminder system. This system can relieve unevenness in taking recommended dosage of pills on time prescribed by the doctor and switch from ways primarily reliant with the memory of the human being insignificant regulation, hence people can be freed doing wrong things due to human error like taking pill at different time with incorrect dosage. The proposed application would help people who are under medication mainly for old persons to take the medicine on time without forgetting and inform them to take necessary action. A person's life can be saved by this mobile application. Human effort can also be decreased by this medicine remainder.

This is an innovative application for any client generally focusing on the matured populace as a clinical aid. This application is an integration of many important medical implementations into a single source so it can work within and communicate within the same application. The client or anybody benefit of the client can enter the medication updates, for example, tablet type, name, amount and when it ought to be taken with an update. The update assists the client with investigating the tablets that the individual needs to take.

TABLE OF CONTENTS

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

In modern society, busy life has made people forget many things in day to day life. The elderly people and the people victims of chronicle diseases who need to take the medicines timely without missing are suffering from dementia, which is forgetting things in their daily routine. Considering this situation study has been done in this. Application reviewing the technologies of health care which is used for improving this situation by reminding the schedule of medicine, can be done by prescriber through web.

Most of the time due to number of work for the people as well as regarding age and some disease which leads to forget the basic things among daily routine. If the patient sufferings from the disease where it is compulsory to take medicine at proper time, in this appliaction we have proposed the technology of home-health care system of Medicine Reminder system and some improvement regarding authentication have well focused.The platform used is Android which is an open-source technology that can help in better availability of the environment for the user. It is a completely free and easy to use mobile Application.

1.1 Mobile Application development

Mobile application development is a term used to denote the act or process by which application software is developed for handheld devices, such as personal digital assistants, enterprise digital assistants or mobile phones.

These applications can be pre- installed on phones during manufacturing platforms or delivered as web applications using server-side or User interface design (UI) or user interface engineering is the design of user interfaces for machines and software, such as computers, home appliances, mobile devices, and other electronic devices, with the focus on maximizing the user experience.The goal of user interface design is to make the user's interaction as simple and efficient as possible; in terms of accomplishing user goals (user-centered design).

Good user interface design facilitates finishing the task at hand without drawing unnecessary attention to it. Graphic design and typography are utilized to support its usability, influencing how the user performs certain interactions and improving the aesthetic appeal.

ADVANTAGES:

- Improves Efficiency.
- Offers High Scalability.
- Secures the App Data.
- Easy to Maintain.
- Improves Customer Relationship.
- Facilitates New Client Data Retrieval.
- Provides Real-time Project Access.
- Ease in Project Management.

1.2 Android Studio

Android Studio is the official Integrated Development Environment (IDE) for Android app development, based on IntelliJ IDEA software. It provides the fastest tools for building apps on every type of android device. It is a purpose-built for android to accelerate the development and helps to build the highest-quality apps for every android device.

FEATURES:

- A flexible Gradle-based build system.
- A unified environment where one can develop for all Android devices.
- Apply Changes to push code and resource changes to the running app without restarting the app.
- 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.

1.3 JAVA

Java is an object-oriented programming language created by James Gosling, Mike Sheridan, and Patrick Naughton in 1991.

It is a high-level, class-based language that is designed to have a few implementation dependencies as possible.

It is a general purpose programming language intended to let android developers run the compiled Java code on all platforms that support Java without any need for recompilation.

FEATURES OF JAVA:

- Simple: Java is designed to be easy to learn.
- Secure: With Java's secure feature it enables to develop virus-free, tamper-free systems. Authentication techniques are based on public-key encryption.
- Architecture-neutral: Java compiler generates an architecture-neutral object file format, which makes the compiled code executable on many processors, with the presence of Java runtime system.
- Portable: Being architecture-neutral and having no implementation dependent aspects of the specification makes Java portable. The compiler in Java is written in ANSI C with a clean portability boundary, which is a POSIX subset.
- Robust: Java makes an effort to eliminate error-prone situations by emphasizing mainly on compile time error checking and runtime checking.

1.4 XML

Extensible Markup Language (XML) is a markup language that defines a set of rules for encoding documents in a format that is both human-readable and machine readable. The design goals of XML focus on simplicity, generality, and usability across the Internet. It is a textual data format with strong support via Unicode for different human languages. Although the design of XML focuses on documents, the language is widely used for the representation of arbitrary data structures such as those used in web services.

FEATURES OF XML:

- XML focuses on data rather than how it looks.
- Easy and efficient data sharing.
- Compatibility with other markup language HTML.
- Supports platform transition.
- Allows XML validation.
- Adapts technology advancements.
- XML supports Unicode.

CHAPTER 2

LITERATURE SURVEY

Taking regular medicines is common, and it's not unusual for people to miss an occasional dose or take it outside the regular time window. Forgetting to do something is normal, but in the case of medicines, forgetting to take them at the prescribed time can have negative health effects.

By one estimate, about half the population of people taking regular medicines don't take them as prescribed. Is this a breakdown in communication? A lack of understanding of their importance? Forgetfulness?

Largely, reasons for not taking medicines as prescribed can be organised into two types: intentional and unintentional. Unintentional is when a patient intends to follow the prescribed regimen but doesn't due to factors outside their control, including forgetfulness, difficulties understanding dosing instructions, or cost. But for some, a patient consciously decides not to follow the prescribed regimen. This could be due to side effects, or not believing in the necessity of the medicine.

Medication-taking is complex because each person is unique and the challenges to each person's medication-taking can vary quite significantly. Forgetting to do something is normal, but in the case of medicines, forgetting to take them at the prescribed time can have negative health effects.

The top reason for medication non-adherence is forgetfulness. In the last decade, Express Scripts ran a multi-year pilot study to figure out the main cause of medication non-adherence. found that of the 600,000 patients, 39% simply forgot to take their meds, 20% did not renew scripts on time, and 10% put off refills resulting in multiple missed doses.

2.1 Aim of the Application

The proposed system is based on Android Operating system which will remind the users to take medicines on time through notification and automatic alarm ringing system. .

Input to the system is the information entered by the patient which includes date, time, medicine name, dosage etc.

The output of the system focuses on "Medication Adherence". Medication adherence usually refers to whether patients take their medications as prescribed (e.g. twice daily), as well as whether they continue to take a prescribed medication.

2.2 Scope of the Application

Creating a basic, easy-to-use app so that users don’t forget their medicine schedules can be directed through the app by themselves or their loved ones if necessary. We have used the available technology to send notification on the smart phone using instapush application. After receiving the notification user needs to press the dispenser button which is located at pill dispenser unit. This technology cannot be used by deaf people.

CHAPTER 3

SOFTWARE REQUIREMENT SPECIFICATIONS

System analysis is the process of observing systems for troubleshooting or development purposes where computer-based systems require defined analysis according to their makeup and design. A software requirement definition is an abstract description of the services which the system should provide, and the constraints under which the system must operate.

3.1 SOFTWARE REQUIREMENTS

- Programming language: Java
- Operating system: ANY OS (Recommended: Windows10, Windows 11)
- Digital Storage: 2GB of digital storage minimum, 4GB recommended (500MB for IDE + 1.5GBfor android SDK and emulator system image.)
- Minimum required JDK version: Java Development Kit 8
- Minimum Screen resolution: 1280 x 800

3.2 HARDWARE REQUIREMENTS

- CPU-Intel Core i5+
- RAM - 8GB
- Peripherals - Keyboard, Mouse

3.3 TECHNOLOGIES USED

- Front End - XML
- Backend – JAVA

The various methods used in this project are as follows: -

- Emulator - To perform and display the functionality of the project.
- Android Studio - To create, design, test, debug and run the android project.
- Mouse - To navigate through the emulator.
- Keyboard - To give inputs to the project.

CHAPTER 4

SYSTEM DESIGN

4.1 FLOW CHART

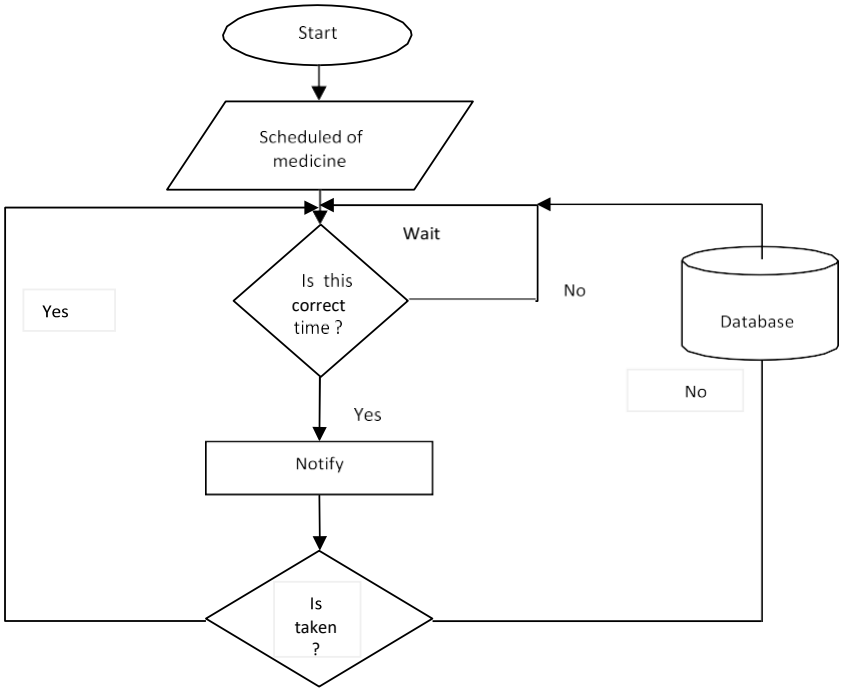

Fig 4.1 : Flow Chart of Medicine Reminder

A flowchart is a picture of the separate steps of a process in sequential order. It is a generic tool that can be adapted for a wide variety of purposes, and can be used to describe various processes. The flow of control in the flow chart is respected to the Texture Package. For any of the program flow chart is compulsory to understand the program. We consider the flow chart for the texture project in which the flow starts from start and proceeds to the main function after which it comes to the initialization of call back functions and further it proceeds to mouse and keyboard functions, input and calculation functions. Finally, it comes to quit, the end of flow chart.

The conceptual working of medicine reminder and monitoring system in flow chart describes the scheduling and the procedure of taking medicine, if schedule is followed by patient or not the data will be stored in the cloud. The stored data will be used to analyze record of patient and further prescription will be give according to it.

4.2 PROJECT DIRECTORY

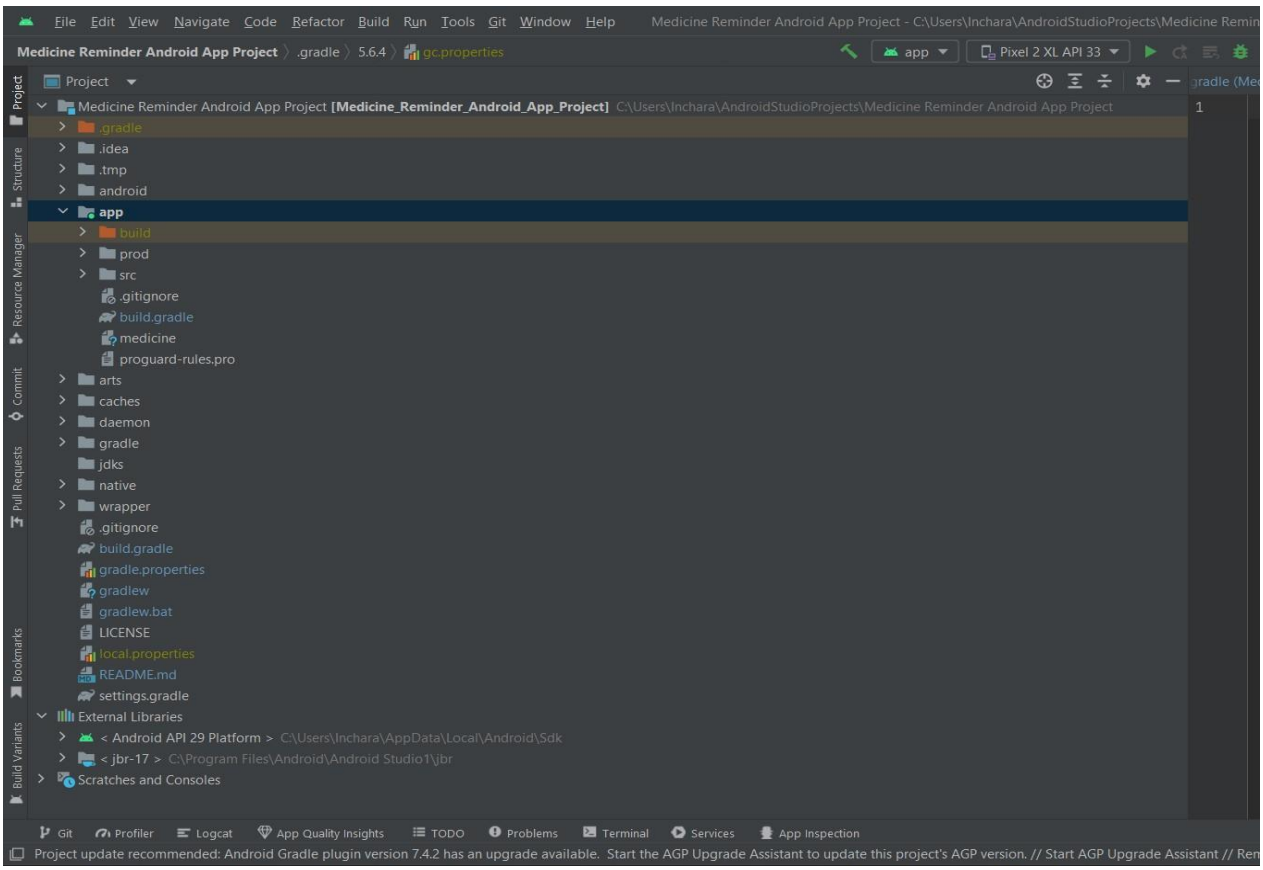

Fig 4.2 : List of all the files and folders used in the project

This picture contains the list of all the files and folders used in the project. We have multiple IDE's, Tools and Technologies like:

- Android Software Development Kit (SDK)
- Android Debug Bridge (ADB)
- Gradle Build
- Android Device Monitor
- SDK Manager
- AVD Manager and many more.

CHAPTER 5

IMPLEMENTATION

5.1 LIST OF CLASSES IMPORTED

- Bundle: Bundles are generally used for passing data from one activity to another activity.
- Intent: Intent is the message that is passed between components such as activities,content providers, broadcast receivers, services etc.
- TextView: A user interface element that displays text to the user.
- EditText: A EditText is an overlay over TextView that configures itself to be editable. It is the predefined subclass of TextView that includes rich editing capabilities.
- Button: Button is a subclass or TextView and CompoundButton is a subclass od Buttonclass. There are different types of buttons in android such as RadioButton, ToggleButton, CompoundButton etc.
- RecyclerView: It is the ViewGroup that contains the views corresponding to the data.
- DrawerLayout: The user can view the navigation drawer when the user swipes a finger from the left edge of the activity.
- LinearLayoutManager: LayoutManager implementations that lays out items in a grid.

5.2 LIST OF IMPORTED LIBRARY FUNCTIONS

- findViewById(): Finds a view that was identified by the id attribute from the XML that was presentedin OnCreate(Bundle).
- getText(): To get text entered in the EditText views. Fun Quiz Application Design and Implementation
- toString(): It’s a method in java’s Object class, which is the superclass of every javaobject. It is meant for returning textual representation of an object.
- getId(): It returns the id value, or the value set via setId().
- getName(): Method class is helpful to get the name of methods, as a String. To get the name of all methods of a class, get all the methods of that class object.
- getInstance(): Provided class is used to return Signature object that implements the specified signaturealgorithm.

- setOnClickListener(): OnclickListener wires the listener to the button using setOnClickListener. Asa result. The system executes the code you write in onClick after the user presses the button.
- setClickable(): Defines whether the view reacts to the click events

5.3 CODE SNIPPETS

- XML CODE:

Activity_add_medicine.xml

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.drawerlayout.widget.DrawerLayout
xmlns:android="http://schemas.android.com/apk/res/android"
xmlns:app="http://schemas.android.com/apk/res-auto"
xmlns:tools="http://schemas.android.com/tools"
android:id="@+id/drawer_layout"
android:layout_width="match_parent"
android:layout_height="match_parent"
tools:context=".addmedicine.AddMedicineActivity">
<LinearLayout
android:layout_width="match_parent"
android:layout_height="match_parent"
android:orientation="vertical">
<com.google.android.material.appbar.AppBarLayout
android:layout_width="match_parent"
android:layout_height="wrap_content">
<androidx.appcompat.widget.Toolbar
```

```
android:id="@+id/toolbar"
style="@style/ToolbarStyle"
android:layout_width="match_parent"
android:layout_height="?attr/actionBarSize"
app:layout_collapseMode="pin"
app:popupTheme="@style/AppTheme.PopupOverlay" />
</com.google.android.material.appbar.AppBarLayout>
<androidx.coordinatorlayout.widget.CoordinatorLayout
android:id="@+id/coordinatorLayout"
android:layout_width="match_parent"
android:layout_height="match_parent">
<FrameLayout
android:id="@+id/contentFrame"
android:layout_width="match_parent"
android:layout_height="match_parent" />
<com.google.android.material.floatingactionbutton.FloatingActionButton
android:id="@+id/fab_edit_task_done"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:layout_margin="@dimen/sixteen_dp"
app:fabSize="normal"
app:layout_anchor="@id/contentFrame"
app:layout_anchorGravity="bottom|right|end"
app:srcCompat="@drawable/ic_add" />
</androidx.coordinatorlayout.widget.CoordinatorLayout>
```

```
</LinearLayout>
</androidx.drawerlayout.widget.DrawerLayout>
```

Activity_remainder.xml

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.coordinatorlayout.widget.CoordinatorLayout
xmlns:android="http://schemas.android.com/apk/res/android"
xmlns:app="http://schemas.android.com/apk/res-auto"
android:id="@+id/coordinatorLayout"
android:layout_width="match_parent"
android:layout_height="match_parent">
<com.google.android.material.appbar.AppBarLayout
android:layout_width="match_parent"
android:layout_height="180dp">
<androidx.appcompat.widget.Toolbar
android:id="@+id/toolbar"
style="@style/ToolbarStyle"
android:layout_width="match_parent"
android:layout_height="?attr/actionBarSize"
app:layout_collapseMode="pin"
app:navigationIcon="@drawable/ic_clear"
app:popupTheme="@style/AppTheme.PopupOverlay" />
<TextView
android:layout_width="match_parent"
android:layout_height="wrap_content"
android:gravity="center"
android:maxLines="2"
android:paddingLeft="@dimen/sixteen_dp"
android:paddingRight="@dimen/sixteen_dp"
```

```
android:text="@string/reminder_message"
android:textColor="@android:color/white"
android:textSize="30sp" />
</com.google.android.material.appbar.AppBarLayout>
<FrameLayout
android:id="@+id/contentFrame"
android:layout_width="match_parent"
android:layout_height="match_parent"
app:layout_behavior="@string/appbar_scrolling_view_behavior" />
</androidx.coordinatorlayout.widget.CoordinatorLayout>
</FrameLayout>
```

Component_day_view.xml

```
<?xml version="1.0" encoding="utf-8"?>
<FrameLayout xmlns:android="http://schemas.android.com/apk/res/android"
xmlns:tools="http://schemas.android.com/tools"
android:id="@+id/backgroundFrameLayout"
android:layout_width="match_parent"
android:layout_height="match_parent"
android:background="@drawable/background_day_view"
android:orientation="vertical">
<com.inchara.medicinetime.views.RobotoBoldTextView
android:id="@+id/weekDayTextView"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:layout_gravity="center"
android:textSize="@dimen/text_large"/>
```

Activity_medicine.xml

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.coordinatorlayout.widget.CoordinatorLayout
xmlns:android="http://schemas.android.com/apk/res/android"
xmlns:app="http://schemas.android.com/apk/res-auto"
xmlns:tools="http://schemas.android.com/tools"
android:id="@+id/coordinatorLayout"
android:layout_width="match_parent"
android:layout_height="match_parent">
<com.google.android.material.appbar.AppBarLayout
android:id="@+id/app_bar_layout"
android:layout_width="match_parent"
android:layout_height="wrap_content"
android:fitsSystemWindows="true"
android:theme="@style/AppTheme.AppBarOverlay"
app:expanded="false"
app:layout_behavior="com.incharahitha.medicinetime.utils.ScrollingCalendarBehavior">
<com.google.android.material.appbar.CollapsingToolbarLayout
android:id="@+id/collapsingToolbarLayout"
android:layout_width="match_parent"
android:layout_height="match_parent"
android:fitsSystemWindows="true"
android:minHeight="?attr/actionBarSize"
app:contentScrim="?attr/colorPrimary"
app:titleEnabled="false"
app:layout_scrollFlags="scroll|exitUntilCollapsed"
app:statusBarScrim="?attr/colorPrimaryDark">
<LinearLayout
android:id="@+id/compactcalendar_view_container"
android:layout_width="match_parent"
```

```
android:layout_height="250dp"
android:paddingTop="?attr/actionBarSize"
app:layout_collapseMode="parallax"
app:layout_collapseParallaxMultiplier="1.0">
<com.github.sundeepk.compactcalendarview.CompactCalendarView
android:id="@+id/compactcalendar_view"
android:layout_width="match_parent"
android:layout_height="match_parent"
android:paddingLeft="10dp"
android:paddingRight="10dp"
app:compactCalendarBackgroundColor="?attr/colorPrimary"
app:compactCalendarCurrentDayBackgroundColor="#FFC107"
app:compactCalendarCurrentSelectedDayBackgroundColor="#BBDEFB"
app:compactCalendarTextColor="#fff"
app:compactCalendarTextSize="12sp" />
</LinearLayout>
<androidx.appcompat.widget.Toolbar
android:id="@+id/toolbar"
style="@style/ToolbarStyle"
android:layout_width="match_parent"
android:layout_height="?attr/actionBarSize"
app:layout_collapseMode="pin"
app:popupTheme="@style/AppTheme.PopupOverlay">
<RelativeLayout
android:id="@+id/date_picker_button"
android:layout_width="match_parent"
android:layout_height="?attr/actionBarSize"
android:background="?android:selectableItemBackground"
android:gravity="center_vertical"
```

```
android:clickable="true"
android:focusable="true"
android:orientation="vertical">
<TextView
android:id="@+id/date_picker_text_view"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:ellipsize="end"
android:maxLines="1"
android:textStyle="bold"
android:textSize="18sp"
android:textColor="@android:color/white" />
<ImageView
android:id="@+id/date_picker_arrow"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:layout_alignBottom="@id/date_picker_text_view"
android:layout_toRightOf="@id/date_picker_text_view"
app:srcCompat="@drawable/ic_arrow_drop_down"
tools:ignore="ContentDescription,RtlHardcoded" />
</RelativeLayout>
</androidx.appcompat.widget.Toolbar>
</com.google.android.material.appbar.CollapsingToolbarLayout>
</com.google.android.material.appbar.AppBarLayout>
<FrameLayout
android:id="@+id/contentFrame"
android:layout_width="match_parent"
android:layout_height="match_parent"
app:layout_behavior="@string/appbar_scrolling_view_behavior" />
```

```
<com.google.android.material.floatingactionbutton.FloatingActionButton
android:id="@+id/fab_add_task"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:layout_margin="@dimen/sixteen_dp"
app:fabSize="normal"
app:layout_anchor="@id/contentFrame"
app:layout_anchorGravity="bottom|right|end"
app:srcCompat="@drawable/ic_add" />
</androidx.coordinatorlayout.widget.CoordinatorLayout>
```

- JAVA CODE:

Addmedicineactivity.java

```
package com.incharahitha.medicinetime.addmedicine;
import android.os.PersistableBundle;
import androidx.appcompat.app.ActionBar;
import androidx.appcompat.app.AppCompatActivity;
import android.os.Bundle;
import androidx.appcompat.widget.Toolbar;
import com.incharahitha.medicinetime.Injection;
import com.incharahitha.medicinetime.R;
import com.incharahitha.medicinetime.utils.ActivityUtils;
public class AddMedicineActivity extends AppCompatActivity {
public static final int REQUEST_ADD_TASK = 1;
public static final String SHOULD_LOAD_DATA_FROM_REPO_KEY =
"SHOULD_LOAD_DATA_FROM_REPO_KEY";
public static final String EXTRA_TASK_ID = "task_extra_id";
public static final String EXTRA_TASK_NAME = "task_extra_name";
private AddMedicinePresenter mAddMedicinePresenter;
```

```
private ActionBar mActionBar;
@Override
protected void onCreate(Bundle savedInstanceState) {
super.onCreate(savedInstanceState);
setContentView(R.layout.activity_add_medicine);
//Setup toolbar
Toolbar toolbar = (Toolbar) findViewById(R.id.toolbar);
setSupportActionBar(toolbar);
mActionBar = getSupportActionBar();
mActionBar.setDisplayHomeAsUpEnabled(true);
mActionBar.setDisplayShowHomeEnabled(true);
AddMedicineFragment addMedicineFragment = (AddMedicineFragment)
getSupportFragmentManager().findFragmentById(R.id.contentFrame);
int medId =
getIntent().getIntExtra(AddMedicineFragment.ARGUMENT_EDIT_MEDICINE_ID,0);
String medName =
getIntent().getStringExtra(AddMedicineFragment.ARGUMENT_EDIT_MEDICINE_NAME
);
setToolbarTitle(medName);
if (addMedicineFragment == null) {
addMedicineFragment = AddMedicineFragment.newInstance();
if (getIntent().hasExtra(AddMedicineFragment.ARGUMENT_EDIT_MEDICINE_ID)) {
Bundle bundle = new Bundle();
bundle.putInt(AddMedicineFragment.ARGUMENT_EDIT_MEDICINE_ID, medId);
addMedicineFragment.setArguments(bundle);
}
ActivityUtils.addFragmentToActivity(getSupportFragmentManager(),
addMedicineFragment, R.id.contentFrame);
}
```

```
boolean shouldLoadDataFromRepo = true;
// Prevent the presenter from loading data from the repository if this is a config change.
if (savedInstanceState != null) {
// Data might not have loaded when the config change happen, so we saved the state.
shouldLoadDataFromRepo =
savedInstanceState.getBoolean(SHOULD_LOAD_DATA_FROM_REPO_KEY);
}
// // Create the presenter
mAddMedicinePresenter = new AddMedicinePresenter(
medId,
Injection.provideMedicineRepository(getApplicationContext()),
addMedicineFragment,
shouldLoadDataFromRepo);
}
public void setToolbarTitle(String medicineName) {
if (medicineName == null) {
mActionBar.setTitle(getString(R.string.new_medicine));
} else {
mActionBar.setTitle(medicineName);
}
}
@Override
public void onSaveInstanceState(Bundle outState, PersistableBundle outPersistentState) {
outState.putBoolean(SHOULD_LOAD_DATA_FROM_REPO_KEY,
mAddMedicinePresenter.isDataMissing());
super.onSaveInstanceState(outState, outPersistentState);
}
@Override
public boolean onSupportNavigateUp() {
```

```
onBackPressed();
return true;
}
}
```

RemainderActivity.java

```
package com.incharahitha.medicinetime.alarm;
import android.content.Intent;
import android.os.Bundle;
import androidx.appcompat.app.ActionBar;
import androidx.appcompat.app.AppCompatActivity;
import androidx.appcompat.widget.Toolbar;
import android.view.MenuItem;
import com.incharahitha.medicinetime.Injection;
import com.incharahitha.medicinetime.R;
import com.incharahitha.medicinetime.utils.ActivityUtils;
import butterknife.BindView;
import butterknife.ButterKnife;
public class ReminderActivity extends AppCompatActivity {
@BindView(R.id.toolbar)
Toolbar toolbar;
ReminderPresenter mReminderPresenter;
@Override
protected void onCreate(Bundle savedInstanceState) {
super.onCreate(savedInstanceState);
setContentView(R.layout.activity_reminder_actvity);
ButterKnife.bind(this);
setSupportActionBar(toolbar);
ActionBar actionBar = getSupportActionBar();
```

```
if (actionBar != null) {
actionBar.setDisplayHomeAsUpEnabled(true);
}
Intent intent = getIntent();
if (!intent.hasExtra(ReminderFragment.EXTRA_ID)) {
finish();
return;
}
long id = intent.getLongExtra(ReminderFragment.EXTRA_ID, 0);
ReminderFragment reminderFragment = (ReminderFragment)
getSupportFragmentManager().findFragmentById(R.id.contentFrame);
if (reminderFragment == null) {
reminderFragment = ReminderFragment.newInstance(id);
ActivityUtils.addFragmentToActivity(getSupportFragmentManager(), reminderFragment,
R.id.contentFrame);
}
//Create MedicinePresenter
mReminderPresenter = new
ReminderPresenter(Injection.provideMedicineRepository(ReminderActivity.this),
reminderFragment);
}
@Override
public boolean onOptionsItemSelected(MenuItem item) {
if (item.getItemId() == android.R.id.home) {
if (mReminderPresenter != null) {
mReminderPresenter.finishActivity();
}
}
return super.onOptionsItemSelected(item);
```

```
}
@Override
public void onBackPressed() {
if (mReminderPresenter != null) {
mReminderPresenter.finishActivity();
}
}
}
```

MedicineReportActivity.java

```
package com.incharahitha.medicinetime.report;
import android.os.Bundle;
import androidx.appcompat.app.ActionBar;
import androidx.appcompat.app.AppCompatActivity;
import androidx.appcompat.widget.Toolbar;
import android.view.MenuItem;
import com.incharahitha.medicinetime.Injection;
import com.incharahitha.medicinetime.R;
import com.incharahitha.medicinetime.utils.ActivityUtils;
import butterknife.BindView;
import butterknife.ButterKnife;
public class MonthlyReportActivity extends AppCompatActivity {
private static final String CURRENT_FILTERING_TYPE = "current_filtering_type";
@BindView(R.id.toolbar)
Toolbar toolbar;
private MonthlyReportPresenter presenter;
@Override
protected void onCreate(Bundle savedInstanceState) {
super.onCreate(savedInstanceState);
```

```
setContentView(R.layout.activity_monthly_report);
ButterKnife.bind(this);
setSupportActionBar(toolbar);
ActionBar actionBar = getSupportActionBar();
if (actionBar != null) {
actionBar.setHomeAsUpIndicator(R.drawable.ic_clear);
actionBar.setDisplayHomeAsUpEnabled(true);
}
//Create Fragment
MonthlyReportFragment monthlyReportFragment = (MonthlyReportFragment)
getSupportFragmentManager().findFragmentById(R.id.contentFrame);
if (monthlyReportFragment == null) {
monthlyReportFragment = MonthlyReportFragment.newInstance();
ActivityUtils.addFragmentToActivity(getSupportFragmentManager(),
monthlyReportFragment,
R.id.contentFrame);
}
//Create TaskPresenter
presenter = new
MonthlyReportPresenter(Injection.provideMedicineRepository(MonthlyReportActivity.this),
monthlyReportFragment);
//Load previous saved Instance
if (savedInstanceState != null) {
FilterType taskFilterType = (FilterType)
savedInstanceState.getSerializable(CURRENT_FILTERING_TYPE);
presenter.setFiltering(taskFilterType);
}
}
@Override
```

```
public boolean onOptionsItemSelected(MenuItem item) {
if (item.getItemId() == android.R.id.home){
onBackPressed();
}
return super.onOptionsItemSelected(item);
}
@Override
public void onBackPressed() {
super.onBackPressed();
}
@Override
protected void onSaveInstanceState(Bundle outState) {
outState.putSerializable(CURRENT_FILTERING_TYPE, presenter.getFilterType());
super.onSaveInstanceState(outState);
}
```

CHAPTER 6

SNAPSHOTS

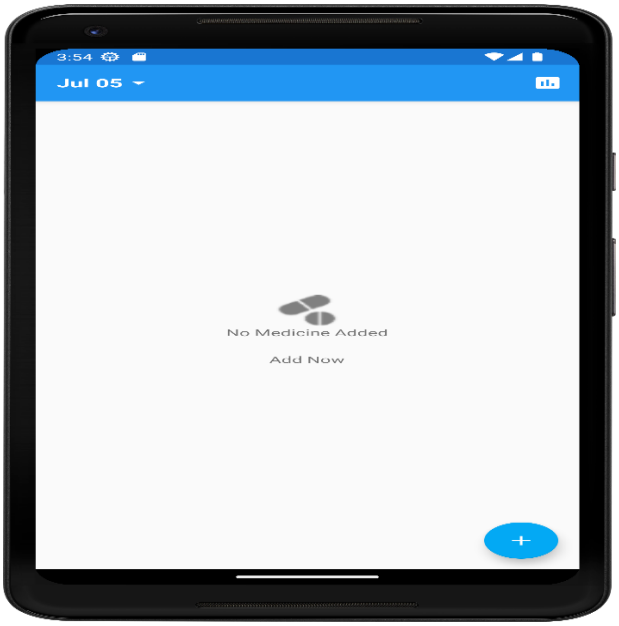

Fig 6.1 : First display picture

The above figure is the first page that appears as soon as the project is launched. It contains a text “No Medicine Added” and on clicking “+” we can add new medicine, time and dosage.

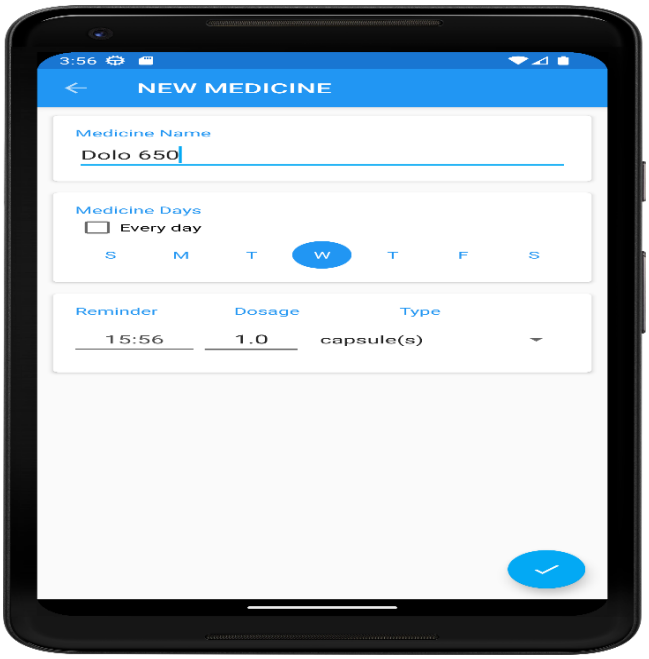

Fig 6.2 : Adding new medicine details

The picture above contains the details of the medicine that we have to give as input i.e., medicine name ,medicine details , time ,dosage and type of medicine.

Fig 6.3 : Setting reminder time

In the above pictire we are setting the time of when the reminder has be set. At that particular time the reminder will be sent to user's phone.

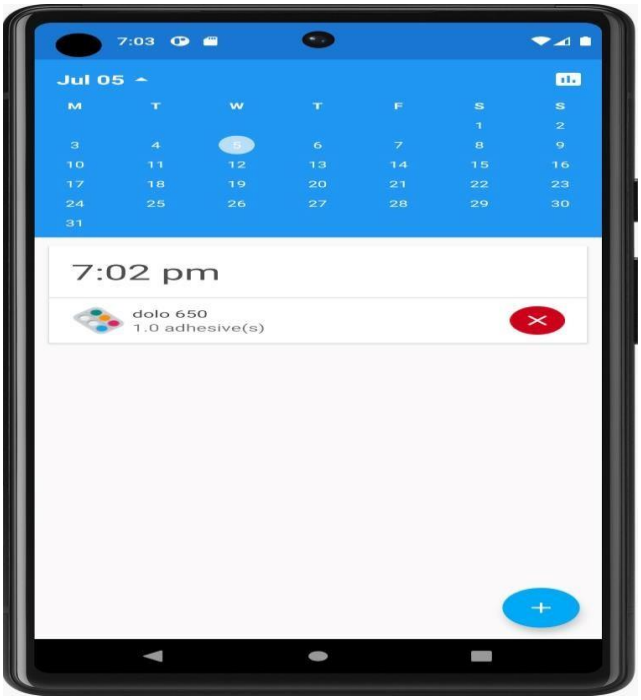

Fig 6.4 : Updated calendar

The above picture contains the details of all the reminders in that particular month. It alsocontains the details of medicine and its description.

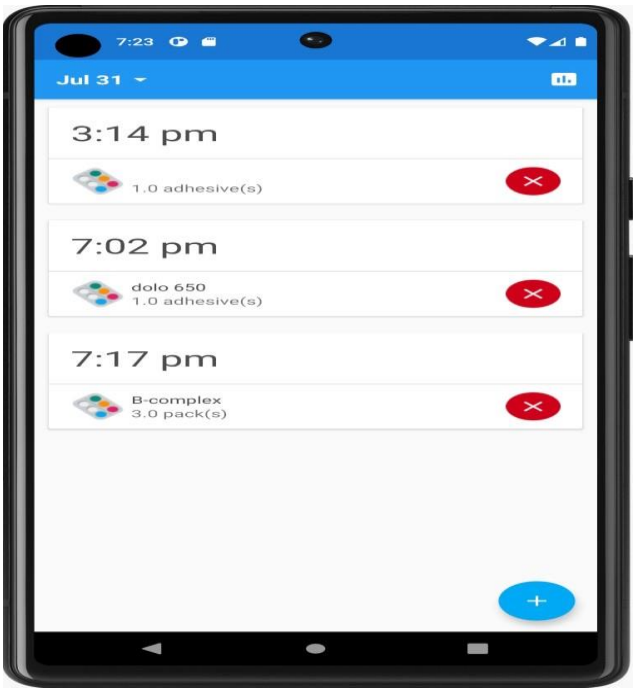

Fig 6.5 : All medicines list

In the above picture we can find all the medicine and their description about the medicines whose reminder has been set.

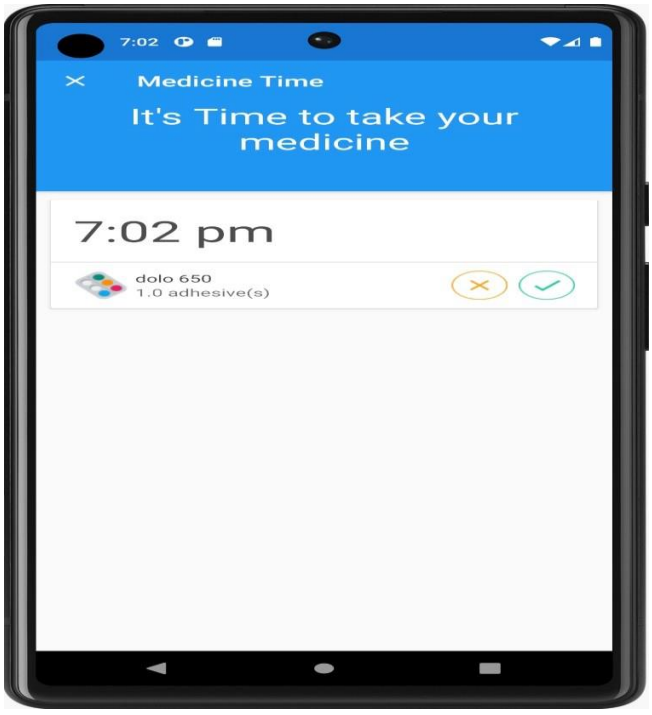

Fig 6.5 : Alarm notification

The above picture is displayed at the time which has been set by the user . It is the alarm notification page which says “IT’S TIME TO TAKE YOUR MEDICINE”

CONCLUSION AND FUTURE ENHANCEMENTS

In conclusion, the medicine reminder Android app is an essential tool that can greatly benefit individuals who need to manage their medication schedules effectively. It provides a convenient and user-friendly way to set reminders, track medication intake, and stay organized with medication routines. The app's features and functionality have the potential to improve medication adherence, reduce the risk of missed doses or overdosing, and enhance overall health outcomes.

Future Enhancements:

- Customizable Reminders: Consider adding more flexibility to the reminder settings, allowing users to customize the frequency and timing of reminders based on their specific needs. This can include options for multiple daily reminders, advanced scheduling options, and personalized reminders for different medications.
- Medication Interaction Warnings: Enhance the app by integrating a medication interaction database or an API that can provide warnings or alerts about potential drug interactions. This feature would be particularly useful for users taking multiple medications simultaneously.
- Integration with Health Trackers: Explore integration with popular health tracking devices or applications such as fitness trackers, smartwatches, or electronic health records (EHRs). This integration would allow the app to gather additional data, such as heart rate, activity levels, or vital signs, which can help personalize medication reminders and provide more comprehensive health monitoring..
- Medication Information and Education: Include a comprehensive medication database or integrate with reliable sources of medication information. This can provide users with detailed information about their medications, including side effects, dosage instructions, and precautions. Additionally, offering educational resources about various health conditions and treatments can empower users to make informed decisions about their healthcare.

By implementing these future enhancements, the medicine reminder Android app can become an even more powerful tool for medication management, fostering improved medication adherence, better health outcomes, and enhanced overall well-being for its users.

REFERENCES

1. Google Developer Training, Google "Android Developer Fundamentals Course-Concept Reference", Google Developer Training Team,2017.
2. Erik Hellman, "Android Programming - Pushing the Limits", 1" Edition, Wiley India Pvt Ltd, 2014. ISBN-13: 978-8126547197
3. Dawn Griffiths and David Griffiths, "Head First Android Development", 1st Edition, O'Reilly SPD Publishers, 2015. ISBN-13: 978-9352131341
4. Bill Phillips, Chris Stewart and Kristin Marsicano, "Android Programming: The Big Nerd Ranch Guide", 3rd Edition, Big Nerd Ranch Guides, 2017. ISBN-13: 978-0134706054
5. https://android-developers.googleblog.com/2023/06/pixel-fold-and-pixel-tablet-developers-guide.html
6. https://developer.android.com/studio/projects/create-project