# DR. AMBEDKAR INSTITUTE OF TECHNOLOGY

(An Autonomous Institute, Affiliated to Visvesvaraya Technological University, Belagavi, Accredited by NAAC, with 'A' Grade)

**Near Jnana Bharathi Campus, Bengaluru – 560056**

## DEPARTMENT OF INFORMATION SCIENCE AND ENGINEERING

## PROJECT REPORT ON

# MEDICINE REMINDER

**Submitted in partial fulfillment of the requirement for the award of the Degree of**

## BACHELOR OF ENGINEERING IN INFORMATION SCIENCE

**Submitted by**


**RASHMI MN 1DA19IS030**
**KAVYA V 1DA19IS016**


**For the academic year 2022-23.**

**Under the Guidance of**


MR. Satish Basavaraj Basapur
Asst. Professor, Dept. of ISE
Dr. AIT, Bengaluru

# DR. AMBEDKAR INSTITUTE OF TECHNOLOGY

**Near Jnana Bharathi Campus, Bengaluru-560056**
**(An Autonomous Institution, Aided by Government of Karnataka)**

## CERTIFICATE

This is to certify that the project titled "Medicine reminder" carried out by RASHMI MN (1DA19IS030) and KAVYA V(1DA19IS016) is a bonafide student of **DR. Ambedkar Institute of Technology, Bengaluru,** in partial fulfillment for the award of **Degree in Bachelor of Engineering in Information Science and Engineering** during the academic year 2020-2021.It is certified that all the corrections/suggestions indicated during Internal Assessment have been incorporated in the Project report deposited in the department. The Project report has been approved as it satisfies the academic requirements in respect of Project work prescribed for the said Degree.

| GUIDE | HOD | PRINCIPAL |
|---|---|---|
| SATISH BASAVARAJ BASAPUR | Dr. M V VIJAYA KUMAR | Dr. MEENAKSHI M |
| Asst. Professor | Professor & Head | Principal |
| Dept. of ISE | Dept. of ISE | Dr. AIT, Bangalore. |

**Signature with date**

**Name of the Examiners**

**1)**

**2)**

# ACKNOWLEDGEMENT

The sense of jubilation that accompanies the successful completion of this Project would be incomplete without mentioning and thanking all the people who played a vital role in the completion of this project by providing endless encouragement and support.

I would like to profoundly thank the **Management - PVPWT** of Dr. Ambedkar Institute of Technology, fr providing such a healthy environment to learn and implement new technologies.

I wish to express my sincere thanks to **Dr. MEENAKSHI M,** Principal, DR.AIT, who has always been a great source of inspiration while carrying out this Project work.

I am extremely grateful to **Dr. M V Vijaya Kumar**, Professor and Head, Department of ISE, DR.AIT for providing us constant encouragement and permitting us to utilize the required laboratory facilities and a congenial working environment for the successful completion of this Project work.

I am highly indebted to my guide **Mr Satish Basavaraj Basapur**, Professor, Department of ISE, DR.AIT for constant guidance and support, as well as for providing necessary information regarding the Project work.

I would also like to thank all the teaching and non-teaching staff members of the Department of Information Science and Engineering Department for their support during the course of this Project implementation.

Last but not the least, I would like to thank my parents and friends whose constant encouragement and support was crucial in execution and completion of this Project.

**RASHMI MN (1DA19IS030)**
**KAVYA V (1DA19IS016)**

## ABSTRACT

Good health has been a major concern since the inception of mankind whilst for some people attaining good health requires taking prescribed medicines or pills routinely. However, many patients find it very difficult to keep track of taking their medication in the right time and proportion. This happens especially if it involves taking pills or medication on daily basis due to several reasons such as heavy work load, forgetfulness, old age and alteration in day to day behavior can also have a significant result on whether patient recalls to take prescribed medicine which can be termed as medicine adherence, which is a very serious problem because it may affect the total well-being of the patient, delaying the curing time, raising the total medicine cost of the patient and can be a matter of life and death.

The aim of the study is to design and develop an automated mobile application for medicine or pill reminder as prescribed by a doctor to patient using android studio. The recent advancement in technology has provided an enabling technique to solve these types of problems by designing and developing an application that can run on smart phones in which patients will find it easy to carry along. The project has been made on android studio platform using java, basic xml and object-oriented programming concepts by designing various activities for the application.

The medicine reminder application could impact positively on life of the patients in keeping track of their daily intake of pills as remembering the intake of these prescribed medication can be a matter of life and death.

# TABLE OF CONTENTS

# CHAPTER 1: INTRODUCTION

## 1.1 OVERVIEW

The category of patients involves all human beings-teachers, students, businessmen, housewives, children and also all of us have a busy hectic schedule. Today's life is full of responsibilities and stress. So, people are prone to diseases of different types and it is our duty to make ourselves stay fit and healthy. If the patient stays at home, then he or she might get someone to look after him/her but when one is not at home, is out of the city or state away from home then it is hard for the family members to call them and remind them their dosage timings every time. In our developing and technology dependent life we totally rely on gadgets mainly smart phones. Today, everyone has a smart phone. With these comforts, we get an opportunity to use technology in a better way, so that it can be made useful to us. It also plays an important part in our daily life and helps us stay fit in many ways. The remarkable problem is that, patients forget to take the required medicines in right proportion and at right time. Medication adherence, which refers to the degree or extent to which a patient takes the right medication at the right time according to a doctor's prescription, has recently emerged as a serious issue since many studies have reported that non-adherence may critically affect the patient, thereby raising medical costs. Medication non adherence is a common, complex, and costly problem that contributes to poor treatment outcomes and consumes health care resources.

## PROBLEM STATEMENT

Health is one of the most important things for most individuals simply because without a good health nothing seems to go well. In recent times new diseases have emerged which needs to be taken care off by taking medicine or pills routinely. In addition, the working conditions of some individuals is bad and hectic and as such had resulted them in forgetting to take their administered medicine or pills in the appropriate timing or proportion and for some people its old age, most elderly people suffer from dementia i.e. forgetfulness. Not taking the prescribed medicine at the proper dosage or sometimes could result to what is referred to as medicine adherence which can be stated to as the extent at which the medicine is taken at the correct doctor prescribed time and proportion. Medicine adherence is a very serious problem because it may affect the total well-being of the patient, delaying the curing time and also raising the total medical cost of the patient. As such the design and development of an iOS-based medicine or pill reminder could help in curbing out the aforementioned problems by reminding patients to take their medicines as prescribed by the doctor within the stipulated timing and dosage.

## 1.2. OBJECTIVES

- We are introducing an Android application whose objective is to remind the patients of their dosage timings through Alarm Ringing system so that they can stay fit and healthy.

- This application focusses on the people who forget to take medicines on time. It allows users to set an alarm along with the fields of date, time and medicine description which will allow them to set alarm for multiple medicines at different time intervals.
- The notification system will send a notification after setting an alarm. The user can activate or deactivate the notification accordingly.
- It will be sent as email or message as selected by the user. The patients can be aware of the timings of the medicine and would not have to put extra efforts in just remembering the medicine.
- Medication reminders help in decreasing medication dispensing errors and wrong dosages.
- The other objectives of using this application is to make it portable working with any portable devices with ease like smart phones, or tablets that simply outputs the alarms as a reminder to enable intake of medicines.

# CHAPTER 2 : LITERATURE SURVEY

LITERATURE REVIEW or Literature survey or literature review refers to the content getting from the books which is related to the topic or given topic. It should be referred from the some research paper which is related to the topic which is given to the student... any material which is related to the project from the internet and which is valuable for the student and the literature survey help the student to enhance the report status and calculation, analysis and tabulation also strong which majorly reflects in the report.

- There is a popular saying that health is wealth. Health is one of the most important things for most individuals simply because not having good health can lead to a very miserable life (Leonard, 2008).
- In recent times new diseases have emerged which needs to be taken care off by taking medicine or pills routinely (Ohayon, 2002), for many people there are consequences attached to not taking the prescribed medication in proper time and proportion because it can be the difference between life and death.
- In addition, not taking the prescribed medicine at the proper dosage or time can also result to what is referred to as medicine adherence which can be stated to as the extent at which the medicine is taken at the correct doctor prescribed time and proportion (Hughes, 2004).
- Medicine adherence is a very serious problem because it may affect the total well-being of the patient, delaying the curing time and also raising the total medical cost of the patient (Grooves et al., 2013).

- Yet with all the aforementioned consequences people of different ages still forget to take their prescribed medicines or pills in due time and proportion (Banerjee, 2009).
- However, there are different reasons for been forgetful ranging from busy schedules, old age, cognitive disorders, bad working conditions, Alzheimer disease, loss of memory, dementia, people with emotional problems, stress, anxiety, depression etc. Nevertheless, the recent advancement in technology has provided an enabling technique to solve these types of problems using different methods, one of the methods used is by buying a device designed purposely to remind patient to take their medicines in the prescribed time and proportion, the aforementioned solution seems to be ineffective and costly (Riehemann et al., 2009).
- The study chose to adopt one of the most widely used smartphone OS which is the iOS because they are the top notch in the smartphones business. However, the iOS seems to be very effective in mobile phones according to the top notch engineers
- The proposed system helps a physician in keeping track of his patient medicine intake from his own backend, this aforementioned feature is absent in most medicine reminder applications.
- Hence, the aim of the study is to design, develop and implement an automated iOS based application for medicine or pill reminder as prescribed by a physician to patients using the swift programming language while adopting the Xcode integrated development environment coupled with some iOS API's.

The designed application would help patient to maximize the full benefit of the medicine and abstain from the risk of medicine delay

# CHAPTER 3 : SYSTEM REQUIREMENT SEPCIFICATION

## 3.1 . HARDWARE REQUIREMENTS

- Computer or laptop
- USB cable for emulation
- Mobile for deployment of application

### 3.1.2 HARDWARE DESCRIPTION

a. Computer or laptop

Computer or laptops are vital for coding and deployment of applications. The feasible system which meets software requirements like RAM and supports android studio must be used for the deployment of the application.

b. USB (Universal serial bus):

USB wire is needed to connect the android studio to the phone for the emulation of the application. We can enable the developer option under the settings in the phone and enable USB debugging under the debugging option and connect to the laptop.

We have to select our phone model under new devices in android studio IDE to get the application installed to the phone.

c. ANDROID DEVICE/MOBILE:

Any android phone that supports the deployment of the application can be used. Working of the application while coding in android studio can be

easily done using emulator in android studio and with the aid of the mobile. This also helps us to make any corrections for the applications while working with the same.

## 3.2. SOFTWARE REQUIREMENTS

### 3.2.1 LIST OF SOFTWARE REQUIREMENTS

- Android studio
- JDK
- Emulator

a. Android studio:

Android studio is the best IDE(integrated development environment) for the development of the application in any programming languages like kotlin or java as per our feasibility. Latest version of the android studio needs to be installed in the system for the better frameworks and features. It is specifically built for android development.

We can install android studio through official website https://developer.android.com/studio/install and we can download either the .exe file or .zip folder. Android studio is portable in many laptops and desktops.

b. JDK(Java development kit):

Java development kit is a development environment that enables one to develop applications using java programming language. The JDK includes tools for developing and testing programs written in the Java programming

language and running on the Java platform. JDK can be installed by visiting the official website
www.oracle.com/java/technologies/downloads/
and downloading the latest version as per the system requirements.
JDK is essential for the execution of the application built in java language, but android studio provides the feasibility to install it in its own environment.

c. s Emulator:

Emulator is the software or the hardware device that helps to emulate or deploy our applications on third party devices like mobiles phone before shipping of the applications. The best emulators used are:

- Bluestacks
- LDPlayer
- Android Studio
- Bliss OS
- Game Loop

Fortunately android studio comes with its own emulator namely android virtual device(AVD) with the Android Software Development Kit.

## 3.3 CONNECTION TO THE EMULATOR

Connection to the emulator is done using USB or we can connect our phones through the virtual connection using WIFI. By default, if we cannot install the Android Software Development Kit, we can move along with this option.
The setup is quite simple, plug the USB cable to the USB port in the

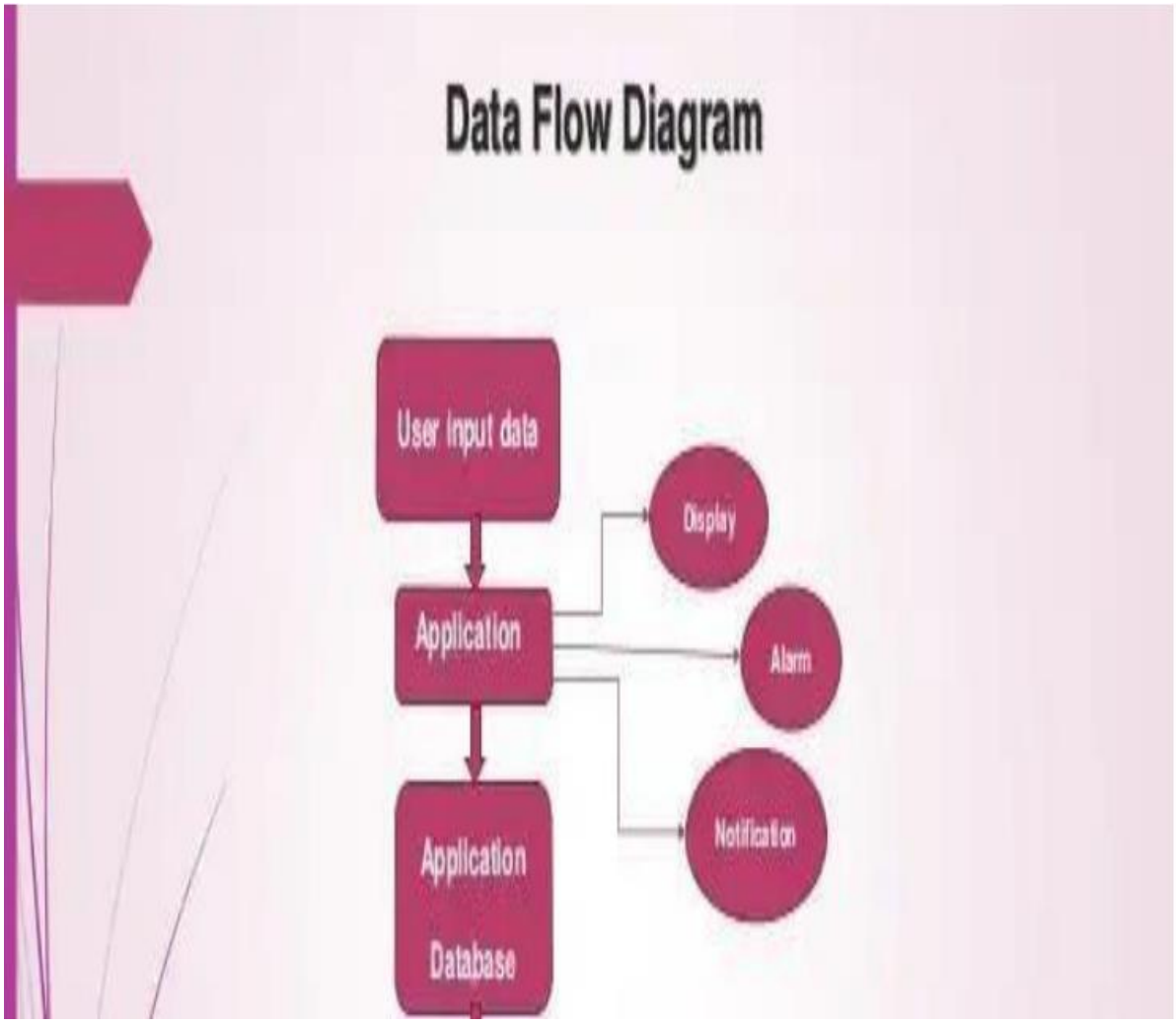
laptop or system.

Then, enable the developer options in the settings of the phone and also enable the USB debugging under the debugging.

Accept to connect to the laptop and the android studio runs to automatically deploy the application to the phone.

## SYSTEM DESIGN

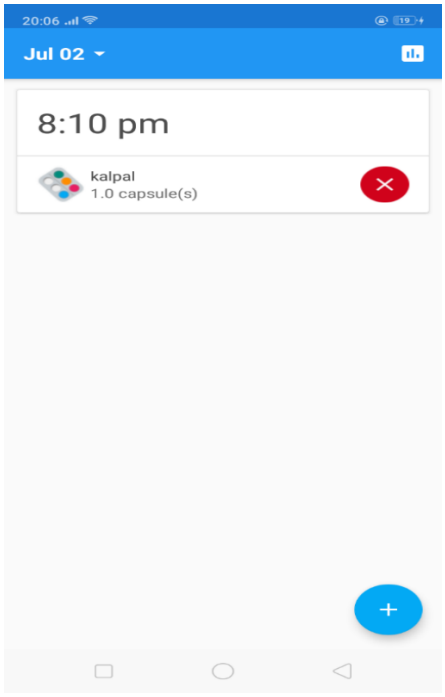
## DATAFLOW DIAGRAM



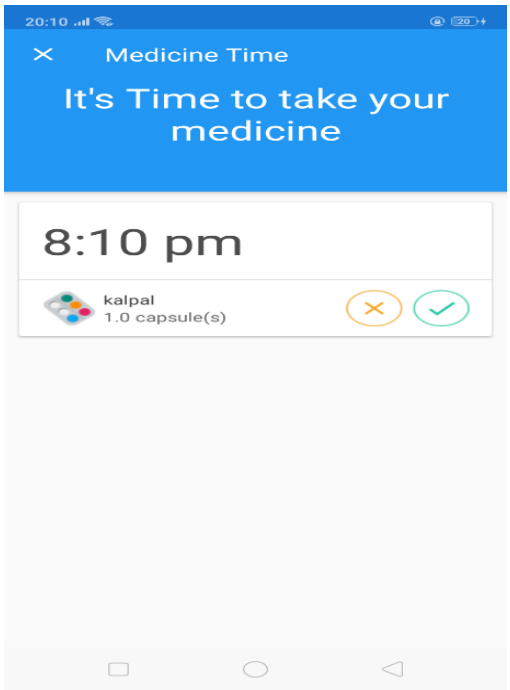
# CHAPTER 4: SCREENSHOTS

The application developed using android studio environment is as shown below.


Once the application is deployed to the phone, we can set the medicine name to be reminded, with number of days and also the timings respectively.




The above screenshot shows the medicine named as kalpal and the timing as 8:10pm on Saturday. The alarm rings on the set time. There is also '+' icon at the end which enables the users to add more medicines that they require to remind them.




When the alarm starts to ring, the above framework is displayed to remind the intake of the medicine.

## CHAPTER 5.1: EXISTING SYSTEM

- The current project signifies the deployment of application on the phone of the medicine reminder. The project is coded using java.
- It follows a simplistic mode of development of application that can be easily used by anyone who tend to forget to have medicines at the right time, which can also save the lives of many.
- Multiple methods to connect to the Android studio emulator is possible but the simplest one is using USB. We can also use Blue stacks, or pair the devices with the connected WIFI to emulate.

## CHAPTER 5.2: DEVELOPED SYSTEM

- Simplistic system which is easily usable and feasible.
- Use of android studio IDE and java as a programming language helps to deploy the application easier.
- The application aims to be portable to any phone and can be carried anywhere as their lifesaver.
- All the software and hardware are easily available to develop the application for the small-scale industries.

# CHAPTER 6: TECHNOLOGIES USED IN DETAIL

## ANDROID STUDIO :

IDE Android Studio is the official integrated development environment (IDE) for Google's Android operating system, built on JetBrains' IntelliJ IDEA software and designed specifically for Android development. It is available for download on Windows, macOS and Linux based operating systems or as a subscription-based service in 2020. It is a replacement for the Eclipse Android Development Tools (E-ADT) as the primary IDE for native Android application development. Android Studio was announced on May 16, 2013 at the Google I/O conference. It was inearly access preview stage starting from version 0.1 in May 2013, then entered beta stage starting from version 0.8 which was released in June 2014. The first stable build was released in December 2014, starting from version 1.0. On May 7, 2019, Kotlin replaced Java as Google's preferred language for Android app development. Java is still supported, as is C++. Feature:

A specific feature of the Android Studio is an absence of the possibility to switch autosave feature off. The following features are provided in the current stable version:

• Gradle-based build support
• Android-specific refactoring and quick fixes
• Lint tools to catch performance, usability, version compatibility and other problems
• ProGuard integration and app-signing capabilities
• Template-based wizards to create common Android designs and components
• A rich layout editor that allows users to drag-and-drop UI components,

option to preview layouts on multiple screen configurations Support for building Android Wear apps

• Built-in support for Google Cloud Platform, enabling integration with Firebase Cloud Messaging (Earlier 'Google Cloud Messaging') and Google App Engine

• Android Virtual Device (Emulator) to run and debug apps in the Android studio.

**JAVA PROGRAMMING**

Android applications are developed using the Java language. As of now, that's really your only option for native applications. Java is a very popular programming language developed by Sun Microsystems (now owned by Oracle). Developed long after C and C++, Java incorporates many of the powerful features of those powerful languages while addressing some of their drawbacks. Still, programming languages are only as powerfulas their libraries. These libraries exist to help developers build applications. Some of the Java's important core features are:

• It's easy to learn and understand

• It's designed to be platform-independent and secure, using

• virtual machines • It's object-oriented Android relies heavily on these Java fundamentals. The Android SDK includes many standard Java libraries (data structure libraries, math libraries, graphics libraries, networking libraries and everything else you could want) as well as special Android libraries that will help you develop awesome Android applications.

## XML BASICS :

Android layouts are written in eXtensible Markup Language, also known as XML. Much like HTML (or HyperText Markup Language), XML is also a markup language. It was created as a standard way to encode data in internet-based applications. However, unlike HTML, XML is case-sensitive, requires each tag is closed properly, and preserves whitespace. Much like creating an HTML layout and later altering it with jQuery, as we've done in previous courses, we can create XML layouts in Android, and later alter them using Java logic. Android XML layouts are also part of a larger umbrella of Android files and components called resources. Resources are the additional files and static content an application needs, such as animations, color schemes, layouts, menu layouts.

## OBJECT ORIENTED PROGRAMMING LANGUAGE

OOP is a programming style or technique that relies upon the definition of data structures called objects. For those new to OOP, an object can be thought of much like a custom data type. For example, you might have a Dog object, which represents the blueprint for a generic dog, with a name, breed, and gender. You could then create different instances of the Dog object to represent specific dogs. Each Dog object must be created by calling its constructor (a method that has the same name as the object itself, and may or may not have parameters for setting initial values). For example, the following Dog objects use a constructor with three parameters (name, breed, gender): Dog dog1 = new Dog("Lassie", collie,

female); Dog dog2 = new Dog("Fifi", poodle, female); Dog dog3 = new Dog("Asta", foxterrier, male); So where is this Dog object defined? Well, here we need to begin defining some of the fundamental building blocks of the Java programming language. A class provides a definition for an object. Therefore, there is a Dog class somewhere—either defined by you or in some library somewhere. Generally speaking, a class will be defined in its own file, with the filename matching the class name (e.g. Dog.java). There are exceptions to this rule, such as classes defined within other classes (when a class is declared within a class, it is generally defined for use within the parent class only as a helper class, and referred to as an inner class). When you want to reference an object from within another class, you need to include an import statement in the top of your class file, much like you would use a #include statement in a compiled language like C. A class typically describes the data and behaviour of an object. The behaviour is defined using class methods. Method is the common term for a subroutine in an OOP language. Many common object classes are defined in shared class libraries like software development kits (SDKs), whereas others are defined by you, the developer, for your own purposes. Software is then built up by using and manipulating object instances in different ways.

# CHAPTER 7: IMPLEMENTATION

## SAMPLE SOURCE CODE

### Manifest.xml

```
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
package="com.gautam.medicinetime">
<uses-permission android:name="android.permission.VIBRATE" />
<application
android:allowBackup="true"
android:icon="@mipmap/ic_launcher"
android:label="@string/app_name"
android:roundIcon="@mipmap/ic_launcher_round"
android:supportsRtl="true"
android:theme="@style/AppTheme.NoActionBar">
<activity android:name=".medicine.MedicineActivity">
<intent-filter>
<action android:name="android.intent.action.MAIN" />
<category android:name="android.intent.category.LAUNCHER" />
</intent-filter>
</activity>
<!-- <activity android:name=".MainActivity" /> -->
<activity android:name=".addmedicine.AddMedicineActivity" />
<activity android:name=".alarm.ReminderActivity" />
<activity android:name=".report.MonthlyReportActivity"/>
</application>
</manifest>
```

### Build.graddle

```
apply plugin: 'com.android.application'
android {
signingConfigs {
config {
keyAlias 'medicine'
keyPassword 'test123'
storeFile file('/Users/gautam/Documents/Projects/MedicineTime/app/medicine')
storePassword 'test123'
}
}
compileSdkVersion 29
buildToolsVersion '29.0.3'
```

```
defaultConfig {
applicationId "com.gautam.medicinetime"
minSdkVersion rootProject.ext.minSdkVersion
targetSdkVersion rootProject.ext.targetSdkVersion
versionCode 1
versionName "1.0"
testInstrumentationRunner "androidx.test.runner.AndroidJUnitRunner"
vectorDrawables.useSupportLibrary = true
signingConfig signingConfigs.config
}
compileOptions {
sourceCompatibility JavaVersion.VERSION_1_8
targetCompatibility JavaVersion.VERSION_1_8
}
buildTypes {
release {
minifyEnabled true
proguardFiles getDefaultProguardFile('proguard-android.txt'), 'proguard-
rules.pro'
signingConfig signingConfigs.config
pseudoLocalesEnabled true
proguardFile '/Users/Documents/Projects/MedicineTime/app/proguardrules.pro
}
}
flavorDimensions "default"
// If you need to add more flavors, consider using flavor dimensions.
productFlavors {
mock {
applicationIdSuffix = ".mock"
dimension "default"
}
prod {
dimension "default"
signingConfig signingConfigs.config
}
}
// Remove mockRelease as it's not needed.
android.variantFilter { variant ->
if (variant.buildType.name == 'release'
&& variant.getFlavors().get(0).name == 'mock') {
variant.setIgnore(true);
}
}
```

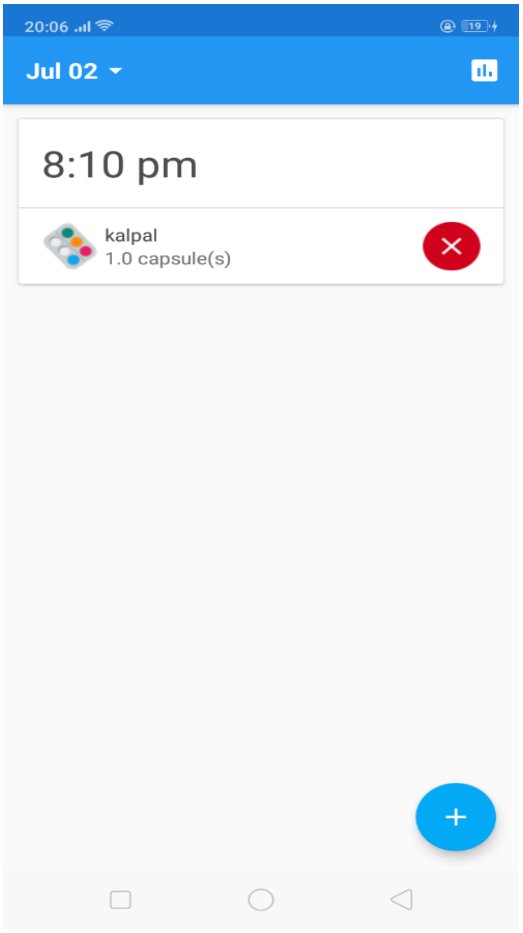
```
}
dependencies {
implementation fileTree(include: ['*.jar'], dir: 'libs')
androidTestImplementation('androidx.test.espresso:espresso-core:3.1.0', {
exclude group: 'com.android.support', module: 'support-annotations'
})
androidTestImplementation "junit:junit:$rootProject.ext.junitVersion"
implementation 'androidx.appcompat:appcompat:1.1.0'
implementation 'androidx.cardview:cardview:1.0.0'
implementation 'com.google.android.material:material:1.1.0'
implementation 'androidx.recyclerview:recyclerview:1.1.0'
implementation 'androidx.legacy:legacy-support-v4:1.0.0'
implementation "com.github.sundeepk:compact-
calendarview:$rootProject.compactCalendarViewVersion"
implementation 'com.jakewharton:butterknife:10.2.1'
annotationProcessor 'com.jakewharton:butterknife-compiler:10.2.1'
}
```

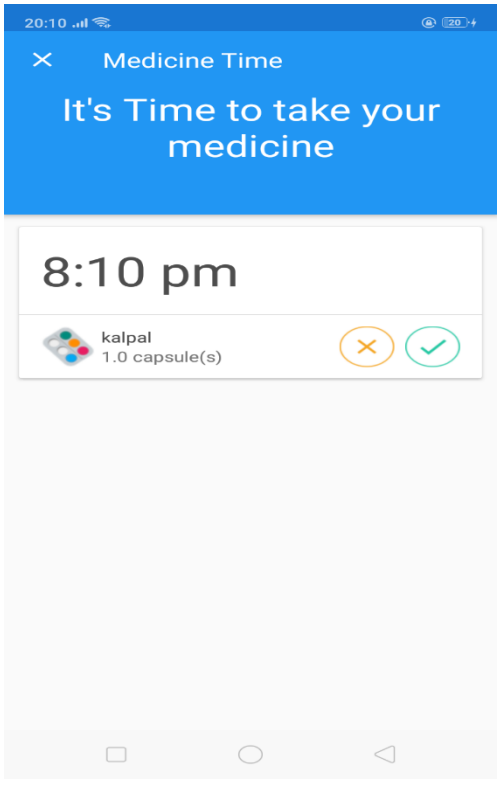
Mainactivity.java

```
package com.rk.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.rk.medicinetime.Injection;
import com.rk.medicinetime.R;
import com.rk.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);
}
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_KE
Y);
}
// // 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)
```

# CHAPTER 8: OUTPUTS

# CHAPTER 9 : CONCLUSION

From the developed system we can finally conclude that:

- We are automating the old ways of remembering the medicine taking and therefore making it easier for the users to keep track of their medication schedule and taking the medicine on time .
- This project will be making the system cheaper than its pre-existing solutions and have an easy-to-use interface which will make it accessible for every user and will not discriminate.
- The medicine reminder will prove to be even more efficient in the upcoming future with some further enhancements as people tend to get busier and busier in the developing world and, in this fastly moving world they surely will need someone who stand by their side and not let them forget their important medicinal routine and make them life a stressful life.
- The existing project can also be easily deployed into any platforms like tablets, phones and also is portable to use.

# CHAPTER 10: FUTURE SCOPE

The developed system meets the current user requirements, but enhancements can be made in the system to run the process more smoothly. Some of the proposed future enhancements can be as follows :

• Making it more user friendly.

• Letting doctors track the patients.

• Letting patients find nearest available doctors.

• Letting users know nearest medical shops in case of emergency.

• Letting users connect their smart watches to the application.

• Providing alarm system.

- The major scope of the existing project is to enable the application to work in various languages to reach to the larger set of people which could be more portable than with the existing one.

# CHAPTER 11: BIBLIOGRAPHY

1) http://www.google.com

2) http://www.w3school.com

3) http://www.tutorialspoint.com

4) http://www.wikipidea.com

5) MEDICATION REMINDER AND HEALTHCARE – AN ANDROID APPLICATION | Kalpana Mudaliar and Deepti Ameta - Academia.edu