

**“VISVESVARAYA TECHNOLOGICAL UNIVERSITY,
BELAGAVI”**



**B.L.D.E.A's V.P. DR. P.G. HALAKATTI COLLEGE OF
ENGINEERING AND TECHNOLOGY, VIJAYAPUR -- 586103**

**BACHELOR OF ENGINEERING IN COMPUTER SCIENCE &
ENGINEERING**

MOBILE APPLICATION DEVELOPMENT (18CSMP68)

MINI PROJECT REPORT ON:

“Setting Alarm”

UNDER THE GUIDANCE OF

Mr.Pavan Mahendrakar

Mr.Vijay Jatti

SUBMITTED BY:

Nikhil Gugawad (2BL19CS057)

Shreehari Hulyalkar (2BL19CS088)

Shashidhar Vijapur (2BL19CS085)



**B.L.D.E.A.'s V.P. Dr. P.G. HALAKATTI COLLEGE OF ENGINEERING
AND TECHNOLOGY, VIJAYAPURA – 586 103**

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

CERTIFICATE

This is to certify that mini project work of **Mobile Application Development(18CSMP68)** entitled “**Setting Alarm**” is a bonafide work carried out in the sixth semester by **Nikhil Gugawad [2BL19CS057]**, **Shreehari Hulyalkar [2BL19CS088]**, **Shashidhar Vijapur [2BL19CS085]** in partial fulfilment for the award of Bachelor of Engineering in Computer Science and Engineering from **B.L.D.E. A's V. P. Dr. P. G. Halakatti College of Engineering & Technology** during the academic year 2021-2022.

GUIDE

Mr.Pavan Mahendrakar

Mr.Vijay Jatti

HOD

Dr.Pushpa B. Patil

PRINCIPAL

Dr.V.G.Sangam

Examiner 1:

Examiner 2:

ACKNOWLEDGEMENT

Mobile Application Development mini project requires guidance, hard work and co-ordination. It gives great pleasure to acknowledge with thanks to the assistance and contribution of many individuals who had been actively involved at various stages of our project.

I would also like to express my heartfelt gratitude to our beloved Principal Dr. V. G. Sangam and Dr. Pushpa. B. Patil, HOD, Computer Science and Engineering Sir BLDEA's CET whose guidance and support was truly invaluable.

I express our gratitude to our guides Mr.Pavan Mahendrakar and Mr.Vijay Jatti, Asst. Prof., BLDEA's CET for their valuable guidance, instance motivation and constant supervision at all places of study for making this a success.

I'm extremely thankful to our parents and friends for their unfailing enthusiasm, moral boosting and encouragement for us in completion of this.

Finally, a vote of thanks to the Department of Computer Science Engineering, both teaching and non-teaching staff for their co-operation extended.

CONTENTS

Chapter No.	Name	Page No.
1.	INTRODUCTION	1
2.	DESIGN	2
3.	IMPLEMENTATION	3
4.	EXPIREMENT RESULTS	10
5.	CONCLUSION	12
	REFERENCES	13

1.INTRODUCTION

1.1 What is an alarm application?

Alarm application is the need of today's time. Everyone needs an alarm in smartphones. An alarm can be of anything such as a meeting schedule, waking up in the morning, informal meetings. There are various tasks in which people need alarm application to keep their schedule reminded. It is a need of every other people either it is a pupil or office person, a businessman or manager. Time is a power that needs to be the emphasis. Being an android application developer, you can develop an alarm application and take in use to schedule your task.

1.2 Procedure:

- ❖ **Step 1: Create a New Project**
create a new project in Android Studio and select **Java** as the programming language.
- ❖ **Step 2: Working with the activity_main.xml file**
Navigate the app>res>layout>activity_main.xml. In this file, we have added two items 'TimePicker' and 'ToggleButton'. TimePicker is used to capture the alarm time and ToggleButton is added to set the alarm on or off. Initially, ToggleButton is set to off. It is set on when an alarm is set.
- ❖ **Step 3: Working with the MainActivity.java file**
Go to MainActivity.java Class. In MainActivity.java class onToggleClicked() method is implemented in which the current hour and the minute is set using the calendar. Alarm services are implemented using AlarmManager class. The alarm is set in such a way that it rings and vibrates repeatedly until the toggle button is turned off. Below is the code for the MainActivity.java file. Comments are added inside the code to understand the code in more detail.
- ❖ **Step 4: Working with BroadcastReceiver (AlarmReceiver) class**
Create a new java class named "AlarmReceiver.java" at the same place where MainActivity.java class resides. In this class onReceive() method is implemented. Here we have added vibration functionality and a default ringtone that starts to vibrate and ring when the alarm time is scheduled. Below is the code for the AlarmReceiver.java file. Comments are added inside the code to understand the code in more detail.
- ❖ **Step 5: Adding permission in "AndroidManifest.xml"**
Go to the "AndroidManifest.xml" file. A BroadcastReceiver is registered in AndroidManifest.xml by adding a receiver section after the application section is over. Also, give permission to vibrate using:
<uses-permission android:name="android.permission.VIBRATE"/>

2.DESIGN

2.1 Principal Elements Used in Making Of an Alarm Application:

- ❖ **RelativeLayout:** It is used as a Layout. It helps in aligning the elements in an organized way. You can give the orientation either horizontally and vertically with the help of `android:orientation="vertical/horizontal"`
- ❖ **Button:** The Button is used to perform tasks such as setting or canceling the alarm. In this application, two buttons are used. On click of the “SET ALARM” button, the alarm will be altered.
- ❖ **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.

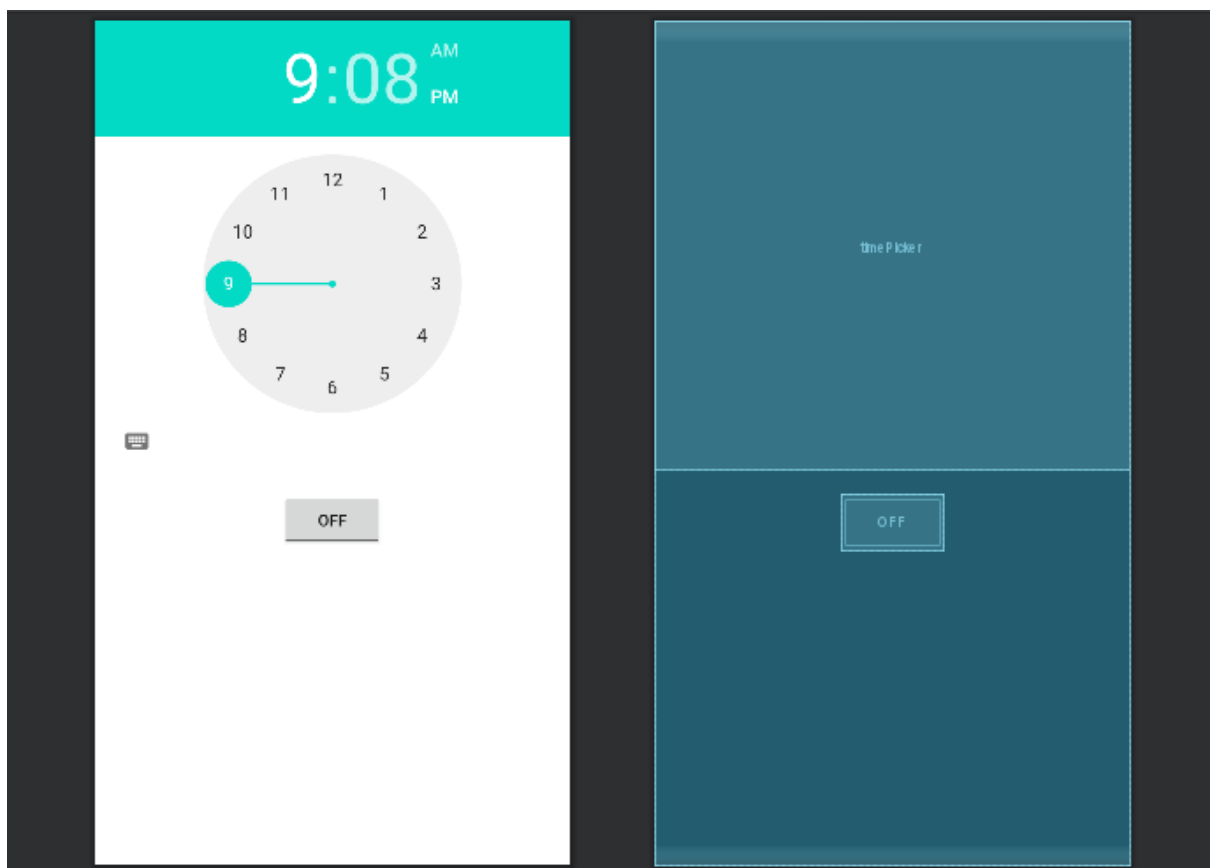


Fig 2.1 Design of alarm application in android studio

3.IMPLEMENTATION

3.1 XML (Extensible Markup Language)

XML (Extensible Markup Language) is a markup language similar to HTML, but without predefined tags to use. Instead, you define your own tags designed specifically for your needs. This is a powerful way to store data in a format that can be stored, searched, and shared. Most importantly, since the fundamental format of XML is standardized, if you share or transmit XML across systems or platforms, either locally or over the internet, the recipient can still parse the data due to the standardized XML syntax.

There are many languages based on XML, including XHTML, MathML, SVG, RSS, and RDF. You can also define your own.

XML Code:

```
<?xml version="1.0" encoding="utf-8"?>

<LinearLayout

    xmlns:android="http://schemas.android.com/apk/res/android"

    android:layout_width="match_parent"

    android:layout_height="match_parent"

    android:orientation="vertical">

    <!--Added Time picker just to pick the alarm time-->

    <!--gravity is aligned to center-->

    <TimePicker

        android:id="@+id/timePicker"

        android:layout_width="wrap_content"

        android:layout_height="wrap_content"

        android:layout_gravity="center" />

    <!--Added Toggle Button to set the alarm on or off-->

    <!--ByDefault toggleButton is set to false-->

    <ToggleButton

        android:id="@+id/toggleButton"

        android:layout_width="wrap_content"
```

```
android:layout_height="wrap_content"  
android:layout_gravity="center"  
android:layout_margin="20dp"  
android:checked="false"  
android:onClick="OnToggleClicked" />
```

<!--"OnToggleClicked" method will be implemented in MainActivity.java -->

</LinearLayout>

3.2 Java-Backend Programming Language:

Java is a high-level, class-based, object-oriented programming language that is designed to have as few implementation dependencies as possible. It is a general-purpose programming language intended to let programmers write once, run anywhere (WORA), meaning that compiled Java code can run on all platforms that support Java without the need to recompile. Java applications are typically compiled to bytecode that can run on any Java virtual machine (JVM) regardless of the underlying computer architecture. The syntax of Java is similar to C and C++, but has fewer low-level facilities than either of them. The Java runtime provides dynamic capabilities (such as reflection and runtime code modification) that are typically not available in traditional compiled languages. As of 2019, Java was one of the most popular programming languages in use according to GitHub, particularly for client–server web applications, with a reported 9 million developers.

Java Code (MainActivity.java)

```
package com.example.alarmclock;

import android.app.AlarmManager;
import android.app.PendingIntent;
import android.content.Intent;
import android.os.Bundle;
import android.view.View;
import android.widget.TimePicker;
import android.widget.Toast;
import android.widget.ToggleButton;

import androidx.appcompat.app.AppCompatActivity;

import java.util.Calendar;

public class MainActivity extends AppCompatActivity {

    TimePicker alarmTimePicker;

    PendingIntent pendingIntent;

    AlarmManager alarmManager;

    @Override
```

```

protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_main);

    alarmTimePicker = (TimePicker) findViewById(R.id.timePicker);
    alarmManager = (AlarmManager) getSystemService(ALARM_SERVICE);

}

// OnToggleClicked() method is implemented the time functionality
public void OnToggleClicked(View view) {
    long time;
    if (((ToggleButton) view).isChecked()) {
        Toast.makeText(MainActivity.this, "ALARM ON",
        Toast.LENGTH_SHORT).show();

        Calendar calendar = Calendar.getInstance();

        // calendar is called to get current time in hour and minute
        calendar.set(Calendar.HOUR_OF_DAY, alarmTimePicker.getCurrentHour());
        calendar.set(Calendar.MINUTE, alarmTimePicker.getCurrentMinute());

        // using intent i have class AlarmReceiver class which inherits
        // BroadcastReceiver
        Intent intent = new Intent(this, AlarmReceiver.class);

        // we call broadcast using pendingIntent
        pendingIntent = PendingIntent.getBroadcast(this, 0, intent, 0);

        time = (calendar.getTimeInMillis() - (calendar.getTimeInMillis() % 60000));
        if (System.currentTimeMillis() > time) {
            // setting time as AM and PM

```

```
        if (calendar.AM_PM == 0)
            time = time + (1000 * 60 * 60 * 12);
        else
            time = time + (1000 * 60 * 60 * 24);
    }

    // Alarm rings continuously until toggle button is turned off

    alarmManager.setRepeating(AlarmManager.RTC_WAKEUP, time, 10000,
pendingIntent);

    // alarmManager.set(AlarmManager.RTC_WAKEUP, System.currentTimeMillis() +
(time * 1000), pendingIntent);

    } else {

        alarmManager.cancel(pendingIntent);

        Toast.makeText(MainActivity.this, "ALARM OFF",
Toast.LENGTH_SHORT).show();

    }

}

}
```

Java Code(AlarmReceiver.java)

```
package com.example.alarmclock;

import android.content.BroadcastReceiver;
import android.content.Context;
import android.content.Intent;
import android.media.Ringtone;
import android.media.RingtoneManager;
import android.net.Uri;
import android.os.Build;
import android.os.Vibrator;
import android.widget.Toast;

import androidx.annotation.RequiresApi;

public class AlarmReceiver extends BroadcastReceiver {
    @RequiresApi(api = Build.VERSION_CODES.Q)
    @Override
    // implement onReceive() method
    public void onReceive(Context context, Intent intent) {

        // we will use vibrator first

        Vibrator vibrator = (Vibrator)
context.getSystemService(context.VIBRATOR_SERVICE);
        vibrator.vibrate(4000);

        Toast.makeText(context, "Alarm! Wake up! Wake up!",
Toast.LENGTH_LONG).show();

        Uri alarmUri = RingtoneManager.getDefaultUri(RingtoneManager.TYPE_ALARM);
        if (alarmUri == null) {
```

```
        alarmUri =  
RingtoneManager.getDefaultUri(RingtoneManager.TYPE_NOTIFICATION);  
    }  
  
    // setting default ringtone  
    Ringtone ringtone = RingtoneManager.getRingtone(context, alarmUri);  
  
    // play ringtone  
    ringtone.play();  
}  
}
```

4.RESULTS

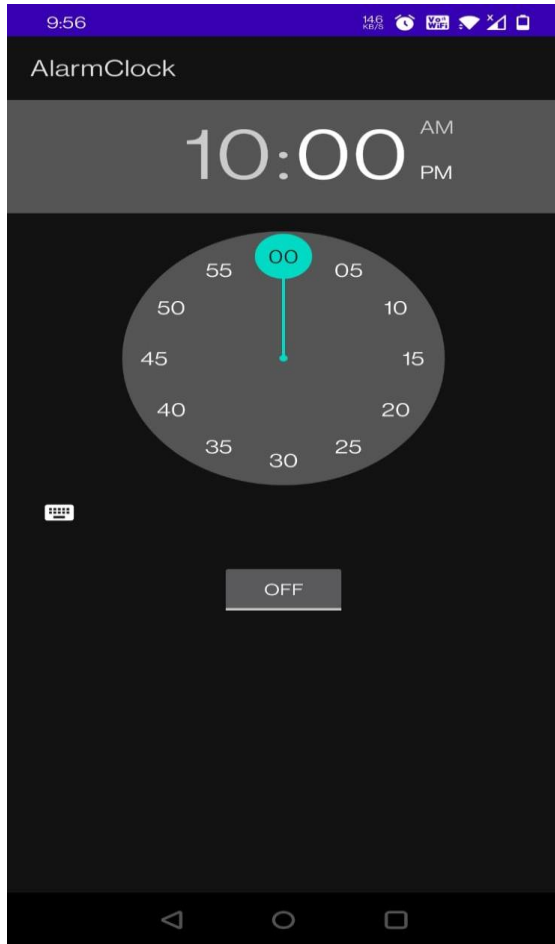


Fig 4.1 alarm is set in analog clock mode

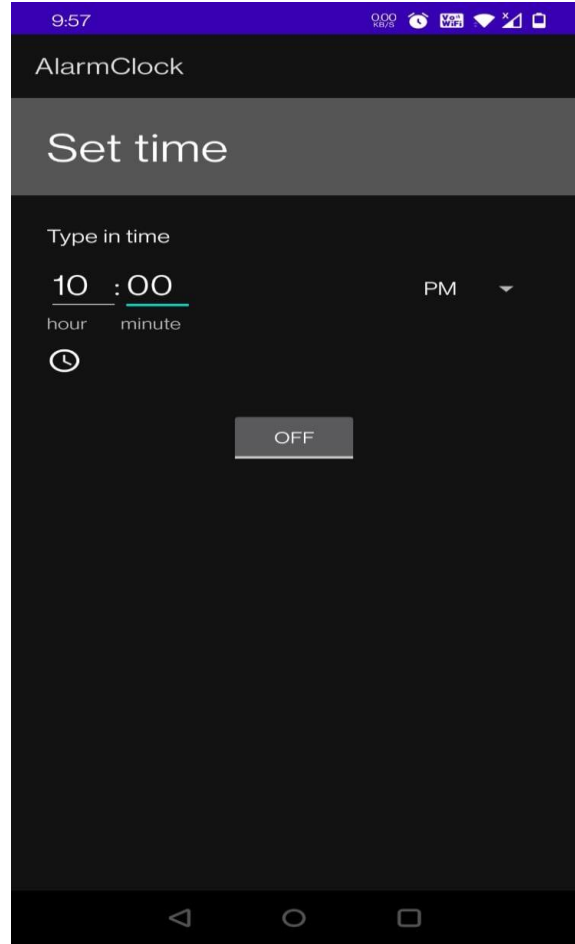


Fig 4.2 alarm is set using keypad

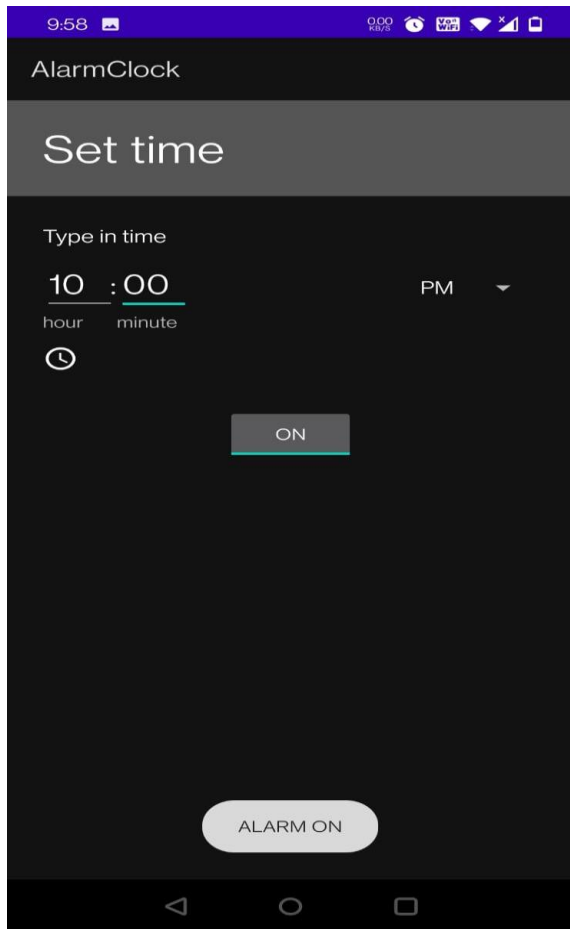


Fig 4.3 The button “on” is clicked

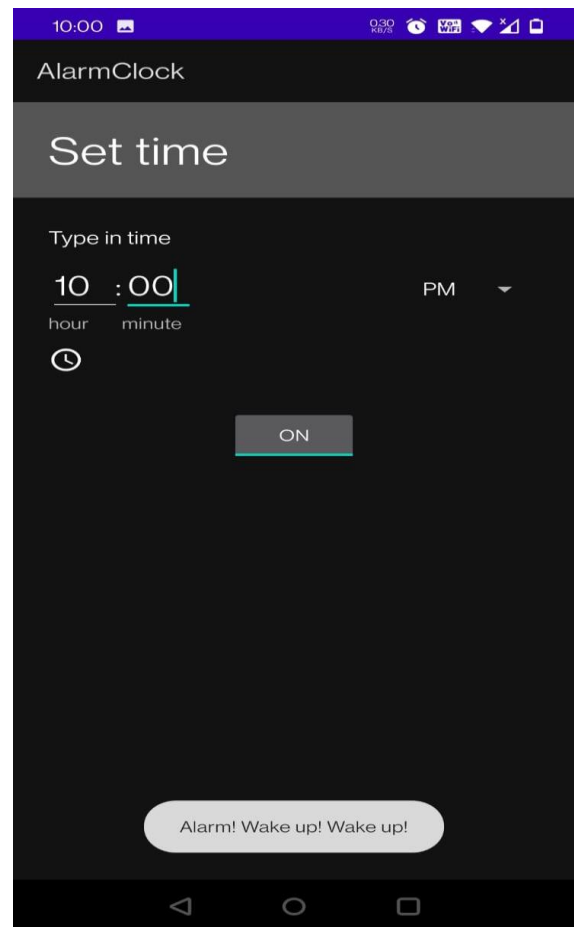


Fig 4.4 Alarm rings

5.CONCLUSION

This project proposed with simple alarm clock app that is simple No fuss, no features that you never use – straight up the best alarm clock apps to get you where you have to be or to remind you of what is coming next. All you need is an alarm that you set when you need it, and then turn it off when you're done. So, if you are one of the heavy sleeper kind of people, then this one is for you. If you're having difficulties getting up on time, then there are various ways that your Android can be helpful in this kind of situation. An alarm clock is the only thing we get mad at when it does its job – and also when it doesn't! However, no matter if you are the heaviest sleeper there is, just want to pump up your alarm clock or just need a little reminder of your future events.

REFERENCES

[1]<https://www.youtube.com/watch?v=LsKjw-IJQpI>

[2]<https://www.wikipedia.org/>

[3]<https://tutorialslink.com/Articles/How-to-create-an-Alarm-Application-in-Android-Android-Tutorial/2028>

[4]<https://www.geeksforgeeks.org/how-to-build-a-simple-alarm-setter-app-in-android/>