05/05/2021 **Mohd. Aftab Alam**
02013302717

EXPERIMENT 4

Aim :

Develop a Clock Application for Mobile Phone using Android Studio.

Theory :

Alarm plays a vital role in our day-to-day life. Nowadays alarm has become our wake-up assistant. Every mobile phone is associated with an alarm app. We will create this app using Android Studio. Android Studio provides a great unified environment to build apps for Android phones, tablets, Android Wear, Android TV, and Android Auto because it provides a very large number of app building features and it is also very easy to use. We are going to implement this application using the Java language.

Requirements :

- Android Studio
- Knowledge of XML and JAVA
- Android emulator (or) Android Mobile

Steps for Creating Clock Application :

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 to 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 **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.

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 will add vibration functionality and a default ringtone that starts to vibrate and ring when the alarm time is scheduled.

***Step** 5:* Playing with Colors

Go to the **"values"** folder first then choose the **colors.xml** file. In the colors.xml file, you can keep colors of your choice as many as you want to use in your app. You have to just give the name and put the color code of the respective colors. We have kept the AppBar color as **"#0F9D58"** which we have named as **"colorPrimary".**

Code :

colors.xml file

```
<?xml version="1.0" encoding="utf-8"?>
<resources>
    <color name="colorPrimary">#0F9D58</color>
    <color name="colorPrimaryDark">#0F4C2E</color>
    <color name="colorAccent">#9D0F9B</color>
</resources>
```

***Step** 6:* Changing Theme of the App

Go to the **"values"** folder first then choose the **themes.xml** file. In the themes.xml file, we have used **"Theme.AppCompat.Light.DarkActionBar"** which is a light theme with a dark ActionBar. We can use a light theme with a light action bar using **"Theme.AppCompat.Light.LightActionBar"**, it all depends on our choice and need.

Code :

themes.xml file

```
<resources>

    <!-- Base application theme. -->
    <style name="AppTheme" parent="Theme.AppCompat.Light.DarkActionBar">
        <!-- Customize your theme here. -->
        <item name="colorPrimary">@color/colorPrimary</item>
        <item name="colorPrimaryDark">@color/colorPrimaryDark</item>
        <item name="colorAccent">@color/colorAccent</item>
    </style>

</resources>
```

***Step** 7:* Adding permission in **AndroidManifest.xml** file

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" />
```

Code :

activity_main.xml file

```
<?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">

    <TimePicker
        android:id="@+id/timePicker"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_gravity="center" />

    <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" />

</LinearLayout>
```

Code :

MainActivity.java file

```
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);

    }

    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.set(Calendar.HOUR_OF_DAY, alarmTimePicker.getCurrentHour());
        calendar.set(Calendar.MINUTE, alarmTimePicker.getCurrentMinute());

            Intent intent = new Intent(this, AlarmReceiver.class);

           pendingIntent = PendingIntent.getBroadcast(this, 0, intent, 0);

time = (calendar.getTimeInMillis() - (calendar.getTimeInMillis() %60000));
            if (System.currentTimeMillis() > time) {

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

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

        } else {

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

        }
    }
}
```

Code :

AlarmReceiver.java file

```
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_S
ERVICE);
        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();
    }
}
```

05/05/2021

Mohd. Aftab Alam
02013302717

App Screenshots :