

Assignment-2(Analog Clock)

Name: Chetan Umale

Student No:118221161

Introduction:

The application displays an analog clock which can be customized by changing the colors of its hands and marks. It also has an alarm feature.

Application Screenshots:

1. The first page displays an analog clock with 2 buttons ie. Settings and Alarm.



Figure 1

2. When the user clicks on the settings button, the user is redirected to the settings page. It has 5 spinners which can be used to change the colors of the hands, marks and background.

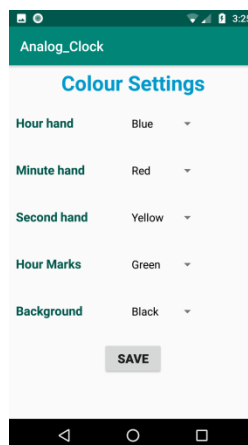


Figure 2

3. When the user clicks on the alarm button, a time picker is opened which can be used to set the alarm.

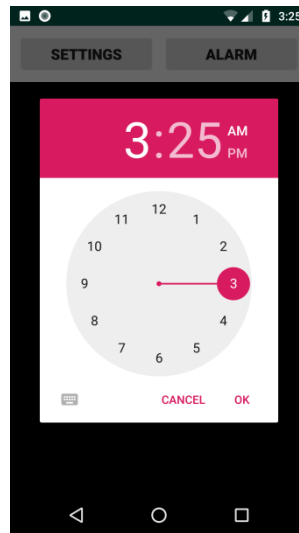


Figure 3

4. When the user sets the alarm, the time picker is closed and a toast is displayed.

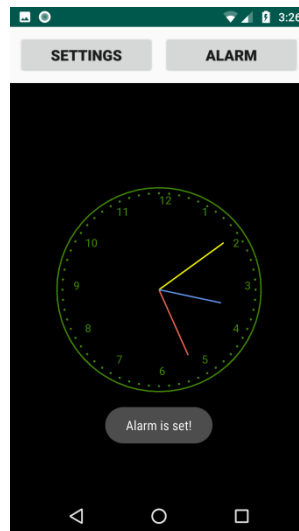


Figure 4

5. When the alarm rings, a toast is displayed and the phone vibrates.

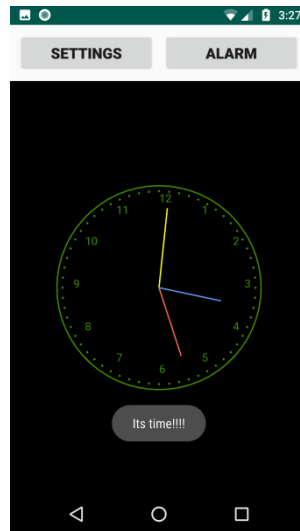


Figure 5

Technical Contribution:

1. The application uses shared preferences file to store the values of the colors which are set by the user. These settings are saved even when the user closes the application. When the application is launched again, the same color settings will be applied.

```
if(pref.getInt( S: "mark_color", E: -1) == -1 || pref.getInt( S: "hour_color", E: -1) == -1
||pref.getInt( S: "minute_color", E: -1) == -1
|| pref.getInt( S: "second_color", E: -1) == -1)
{
    p1.setColor(getResources().getColor(R.color.green));
    p2.setColor(getResources().getColor(R.color.blue));
    p3.setColor(getResources().getColor(R.color.yellow));
    p4.setColor(getResources().getColor(R.color.red));
    p5.setColor(getResources().getColor(R.color.green));
    editor.putInt( S: "hour_color", R.color.blue);
    editor.putInt( S: "minute_color", R.color.red);
    editor.putInt( S: "second_color", R.color.yellow);
    editor.putInt( S: "mark_color", R.color.green);
    editor.putInt( S: "background_color", R.color.black);
    editor.commit();
}
```

Figure 6

2. The application uses AlarmManager class to manage alarms.

```
public void setAlarm(View view)
{
    Calendar cal2 = Calendar.getInstance();
    TimePickerDialog timePickerDialog = new TimePickerDialog( context: this, new TimePickerDialog.OnTimeSetListener() {
        @Override
        public void onTimeSet(TimePicker timePicker, int i, int il) {
            hour = i;
            minute = il;
            alarm_ring();
        }
    }, cal2.get(Calendar.HOUR_OF_DAY), cal2.get(Calendar.MINUTE), is24HourView: false);

    timePickerDialog.show();
}

public void alarm_ring()
{
    Intent intent1 = new Intent( packageContext: this, MyBroadcastReceiver.class);

    PendingIntent pendingIntent = PendingIntent.getBroadcast(this, getApplicationContext(), requestCode: 23433, intent1, flags: 0);

    AlarmManager alarmManager = (AlarmManager) getSystemService(ALARM_SERVICE);
    Calendar cal_alarm = Calendar.getInstance();
    cal_alarm.set(Calendar.HOUR_OF_DAY, hour);
    cal_alarm.set(Calendar.MINUTE, minute);
    cal_alarm.set(Calendar.SECOND, 0);

    alarmManager.setExact(AlarmManager.RTC_WAKEUP, cal_alarm.getTimeInMillis(), pendingIntent);
    Toast.makeText( context: this, text: "Alarm is set!", Toast.LENGTH_SHORT).show();
}
```

Figure 7

BroadcastReceiver class is used for triggering the alarm.

```
public class MyBroadcastReceiver extends BroadcastReceiver {

    MediaPlayer mMediaPlayer;
    @Override
    public void onReceive(Context context, Intent intent) {
        Toast.makeText(context, text: "Its time!!!!", Toast.LENGTH_SHORT).show();

        Vibrator vibrator = (Vibrator) context.getSystemService(Context.VIBRATOR_SERVICE);
        mMediaPlayer = MediaPlayer.create(context, R.raw.alarm_classic);

        mMediaPlayer.start();

        vibrator.vibrate( milliseconds: 3000);
    }
}
```

Figure 8

3. The following permissions are used so that the alarm can use vibration and the alarm is triggered even when the application is closed.

```
<uses-permission android:name="android.permission.WAKE_LOCK" />
<uses-permission android:name="android.permission.VIBRATE" />
```

Figure 9

Design Contribution:

1. The application pops up a time picker instead of using a different activity.

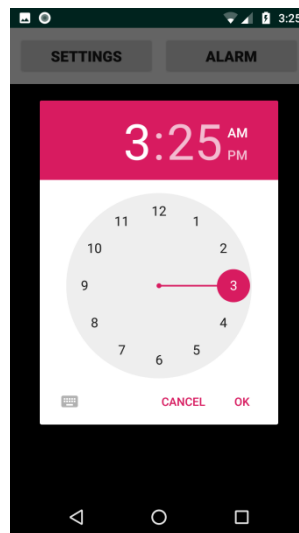


Figure 10

2. The user can customize the colors of the clock.

