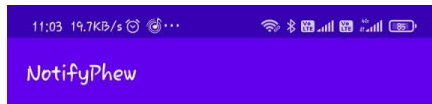


Name : Shivan Tyagi

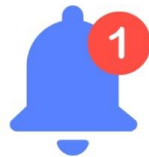
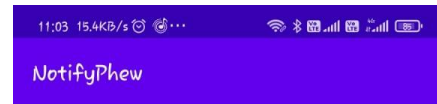
Roll No. : 2047122

Course : Mobile Applications LAB13

1.



2.



from

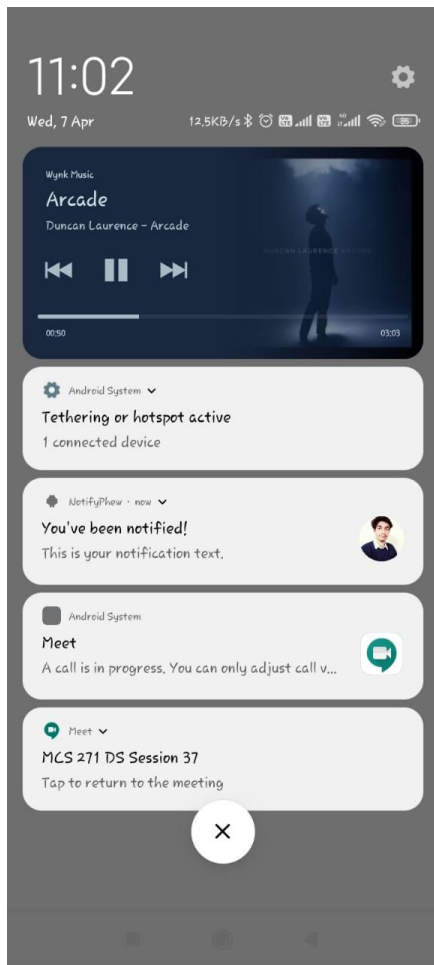
FACEBOOK

NOTIFY ME!

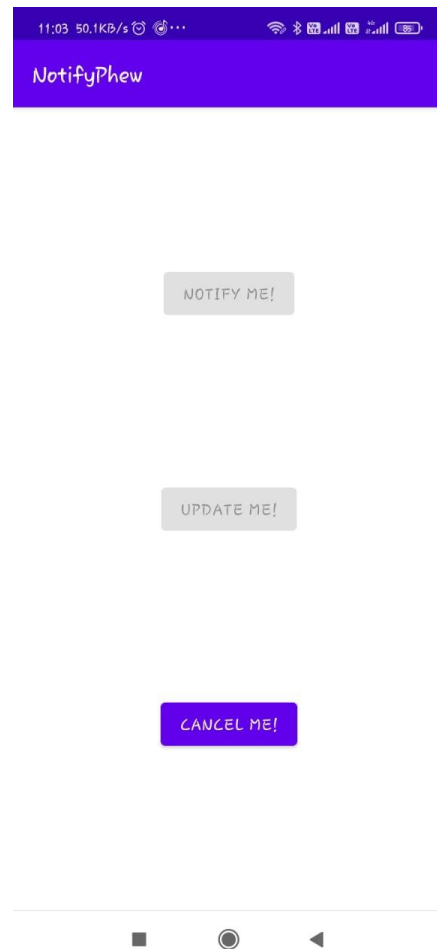
UPDATE ME!

CANCEL ME!

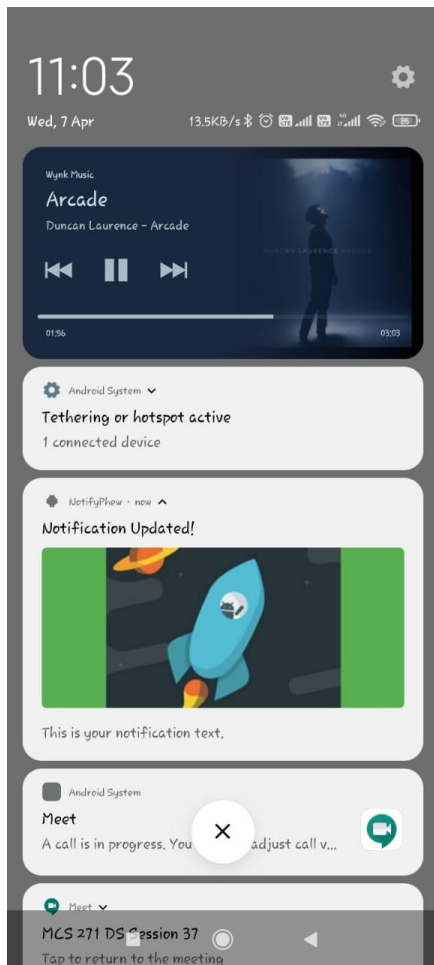
3.



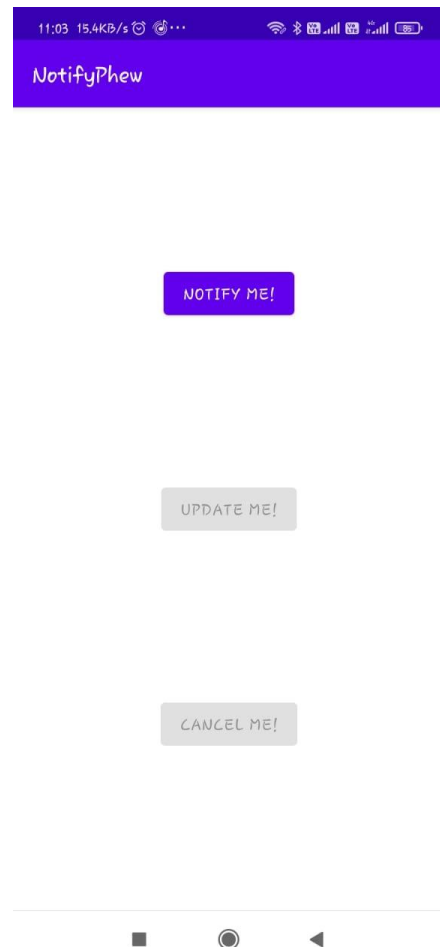
4.



5.



6.



XML CODE:

1.Acitivity_Main.xml

```

<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout
    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:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".MainActivity">

    <com.airbnb.lottie.LottieAnimationView
        android:id="@+id/animation_view"
        android:layout_width="347dp"
        android:layout_height="455dp"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintHorizontal_bias="0.6"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent"
        app:layout_constraintVertical_bias="0.248"
        app:lottie_autoPlay="true"
        app:lottie_fileName="notification.json"
        app:lottie_loop="true"
        tools:ignore="MissingConstraints" />

    <TextView
        android:id="@+id/textView"
        android:layout_width="206dp"
        android:layout_height="50dp"
        android:gravity="center"
        android:text="from "

        android:textSize="20sp"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/animation_view" />

    <TextView
        android:id="@+id/textView2"
        android:layout_width="206dp"
        android:layout_height="50dp"
        android:gravity="center"
        android:text="FACEBOOK "
        android:textColor="#000"
        android:textStyle="bold"
        android:textSize="20sp"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintHorizontal_bias="0.502"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/textView"

```

```
app:layout_constraintVertical_bias="0.0" />  
</androidx.constraintlayout.widget.ConstraintLayout>
```

2. Acitivty_Main2.xml

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout
    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:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".MainActivity2">

    <Button
        android:id="@+id/notify"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="@string/notify_me"
        app:layout_constraintBottom_toTopOf="@+id/update"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent" />

    <Button
        android:id="@+id/update"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="@string/update_me"
        app:layout_constraintBottom_toTopOf="@+id/cancel"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/notify" />

    <Button
        android:id="@+id/cancel"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="@string/cancel_me"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/update" />

</androidx.constraintlayout.widget.ConstraintLayout>
```

JAVA CODE:

1. Main1

```
package com.example.notifyphew;

import androidx.appcompat.app.AppCompatActivity;

import android.content.Intent;
import android.os.Bundle;

import java.util.Timer;
import java.util.TimerTask;

public class MainActivity extends AppCompatActivity {

    Timer timer;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        timer=new Timer();
        timer.schedule(new TimerTask() {
            @Override
            public void run() {
                Intent intent =new Intent(getApplicationContext(),MainActivity2.class);
                startActivity(intent);
            }
        },5000);
    }
}
```

2. Main2

```
package com.example.notifyphew;

import androidx.appcompat.app.AppCompatActivity;
import androidx.core.app.NotificationCompat;

import android.app.NotificationChannel;
import android.app.NotificationManager;
import android.app.PendingIntent;
import android.content.BroadcastReceiver;
import android.content.Context;
import android.content.Intent;
import android.content.IntentFilter;
import android.graphics.Bitmap;
import android.graphics.BitmapFactory;
```

```

import android.graphics.Color;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;

public class MainActivity2 extends AppCompatActivity {

    // @Override
    // protected void onCreate(Bundle savedInstanceState) {
    //     super.onCreate(savedInstanceState);
    //     setContentView(R.layout.activity_main2);
    // }
    // Constants for the notification actions buttons.
    private static final String ACTION_UPDATE_NOTIFICATION =
        "com.android.example.notifyphew.ACTION_UPDATE_NOTIFICATION";
    // Notification channel ID.
    private static final String PRIMARY_CHANNEL_ID =
        "primary_notification_channel";
    // Notification ID.
    private static final int NOTIFICATION_ID = 0;

    private Button button_notify;
    private Button button_cancel;
    private Button button_update;

    private NotificationManager mNotifyManager;

    final NotificationReceiver mReceiver = new NotificationReceiver();

    /**
     * Initializes the activity.
     *
     * @param savedInstanceState The current state data.
     */
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main2);

        // Create the notification channel.
        createNotificationChannel();

        // Register the broadcast receiver to receive the update action from
        // the notification.
        registerReceiver(mReceiver,
            new IntentFilter(ACTION_UPDATE_NOTIFICATION));

        // Add onClick handlers to all the buttons.
        button_notify = findViewById(R.id.notify);
        button_notify.setOnClickListener(view -> {
            // Send the notification
            sendNotification();
        });

        button_update = (Button) findViewById(R.id.update);
        button_update.setOnClickListener(new View.OnClickListener() {

```



```

        @Override
        public void onClick(View view) {
            // Update the notification.
            updateNotification();
        }
    });

    button_cancel = (Button) findViewById(R.id.cancel);
    button_cancel.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View view) {
            // Cancel the notification.
            cancelNotification();
        }
    });

    // Reset the button states. Enable only Notify button and disable
    // update and cancel buttons.
    setNotificationButtonState(true, false, false);
}

/**
 * Unregisters the receiver when the app is being destroyed.
 */
@Override
protected void onDestroy() {
    unregisterReceiver(mReceiver);
    super.onDestroy();
}

/**
 * Creates a Notification channel, for OREO and higher.
 */
public void createNotificationChannel() {

    // Create a notification manager object.
    mNotifyManager =
        (NotificationManager) getSystemService(NOTIFICATION_SERVICE);

    // Notification channels are only available in OREO and higher.
    // So, add a check on SDK version.
    if (android.os.Build.VERSION.SDK_INT >=
        android.os.Build.VERSION_CODES.O) {

        // Create the NotificationChannel with all the parameters.
        NotificationChannel notificationChannel = new NotificationChannel
            (PRIMARY_CHANNEL_ID,
             getString(R.string.notification_channel_name),
             NotificationManager.IMPORTANCE_HIGH);

        notificationChannel.enableLights(true);
        notificationChannel.setLightColor(Color.RED);
        notificationChannel.enableVibration(true);
        notificationChannel.setDescription
            (getString(R.string.notification_channel_description));
    }
}

```

```

        mNotifyManager.createNotificationChannel(notificationChannel);
    }
}

/**
 * onClick method for the "Notify Me!" button.
 * Creates and delivers a simple notification.
 */
public void sendNotification() {

    // Sets up the pending intent to update the notification.
    // Corresponds to a press of the Update Me! button.
    Intent updateIntent = new Intent(ACTION_UPDATE_NOTIFICATION);
    PendingIntent updatePendingIntent = PendingIntent.getBroadcast(this,
        NOTIFICATION_ID, updateIntent, PendingIntent.FLAG_ONE_SHOT);

    // Build the notification with all of the parameters using helper
    // method.
    NotificationCompat.Builder notifyBuilder = getNotificationBuilder();

    // Add the action button using the pending intent.
    notifyBuilder.addAction(R.drawable.ic_update,
        getString(R.string.update), updatePendingIntent);

    // Deliver the notification.
    mNotifyManager.notify(NOTIFICATION_ID, notifyBuilder.build());

    // Enable the update and cancel buttons but disables the "Notify
    // Me!" button.
    setNotificationButtonState(false, true, true);
}

/**
 * Helper method that builds the notification.
 *
 * @return NotificationCompat.Builder: notification build with all the
 * parameters.
 */
private NotificationCompat.Builder getNotificationBuilder() {

    // Set up the pending intent that is delivered when the notification
    // is clicked.
    Intent notificationIntent = new Intent(this, MainActivity.class);
    PendingIntent notificationPendingIntent = PendingIntent.getActivity(
        this, NOTIFICATION_ID, notificationIntent,
        PendingIntent.FLAG_UPDATE_CURRENT);

    // Build the notification with all of the parameters.
    NotificationCompat.Builder notifyBuilder = new NotificationCompat
        .Builder(this, PRIMARY_CHANNEL_ID)
        .setContentTitle(getString(R.string.notification_title))
        .setContentText(getString(R.string.notification_text))
        .setSmallIcon(R.drawable.android)
        .setAutoCancel(true).setContentIntent(notificationPendingIntent)
        .setLargeIcon(BitmapFactory.decodeResource(getResources(),
            R.mipmap.circle_cropped))

```

```

        .setPriority(NotificationCompat.PRIORITY_HIGH)
        .setDefaults(NotificationCompat.DEFAULT_ALL);
    return notifyBuilder;
}

/**
 * OnClick method for the "Update Me!" button. Updates the existing
 * notification to show a picture.
 */
public void updateNotification() {

    // Load the drawable resource into the a bitmap image.
    Bitmap androidImage = BitmapFactory
        .decodeResource(getResources(), R.drawable.mascot_1);

    // Build the notification with all of the parameters using helper
    // method.
    NotificationCompat.Builder notifyBuilder = getNotificationBuilder();

    // Update the notification style to BigPictureStyle.
    notifyBuilder.setStyle(new NotificationCompat.BigPictureStyle()
        .bigPicture(androidImage)
        .setBigContentTitle(getString(R.string.notification_updated)));

    // Deliver the notification.
    mNotifyManager.notify(NOTIFICATION_ID, notifyBuilder.build());

    // Disable the update button, leaving only the cancel button enabled.
    setNotificationButtonState(false, false, true);
}

/**
 * OnClick method for the "Cancel Me!" button. Cancels the notification.
 */
public void cancelNotification() {
    // Cancel the notification.
    mNotifyManager.cancel(NOTIFICATION_ID);

    // Reset the buttons.
    setNotificationButtonState(true, false, false);
}

/**
 * Helper method to enable/disable the buttons.
 *
 * @param isNotifyEnabled, boolean: true if notify button enabled
 * @param isUpdateEnabled, boolean: true if update button enabled
 * @param isCancelEnabled, boolean: true if cancel button enabled
 */
void setNotificationButtonState(Boolean isNotifyEnabled, Boolean
    isUpdateEnabled, Boolean isCancelEnabled) {
    button_notify.setEnabled(isNotifyEnabled);
    button_update.setEnabled(isUpdateEnabled);
    button_cancel.setEnabled(isCancelEnabled);
}

```

```
/**
 * The broadcast receiver class for notifications.
 * Responds to the update notification pending intent action.
 */
public class NotificationReceiver extends BroadcastReceiver {

    public NotificationReceiver() {

    }

    /**
     * Receives the incoming broadcasts and responds accordingly.
     *
     * @param context Context of the app when the broadcast is received.
     * @param intent The broadcast intent containing the action.
     */
    @Override
    public void onReceive(Context context, Intent intent) {
        // Update the notification.
        updateNotification();
    }
}
```