Pimpri Chinchwad Education Trust's
Pimpri Chinchwad College of Engineering

Assignment-04

Roll No: 123M1H048
Name of Student: PRATIK RATHOD
Submission Date: 17 / 10 / 24

---

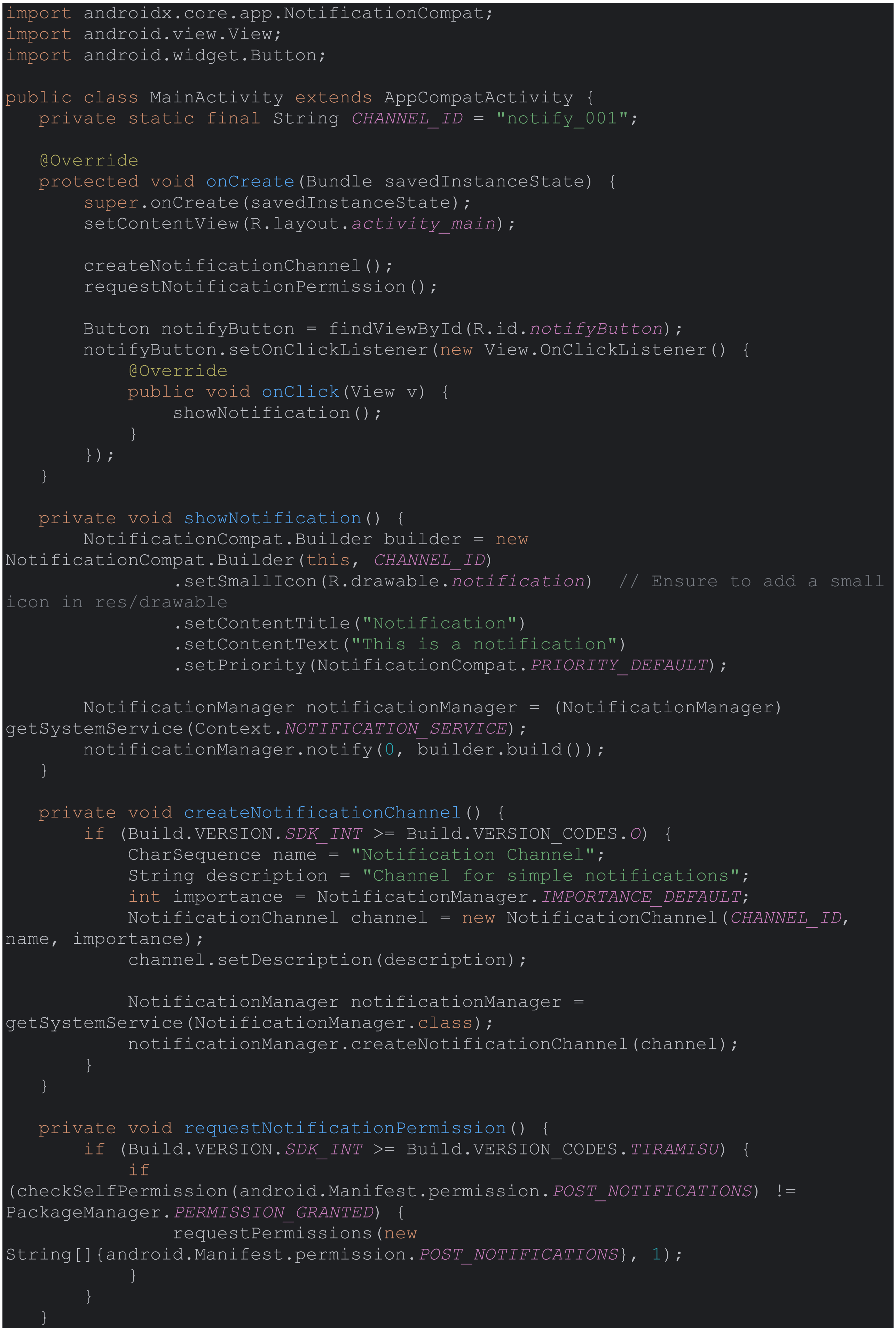
1. Create an Android application that issues a simple notification when a button is clicked. The notification should display a title, message, and small icon. Ensure that the notification appears in the status bar and can be dismissed by the user. Use the NotificationCompat.Builder class to build and issue the notification.

Solution:
xml code:

```
<?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">

   <Button
       android:id="@+id/notifyButton"
       android:layout_width="220dp"
       android:layout_height="89dp"
       android:text="Notification"
       android:textSize="30dp"
       app:layout_constraintBottom_toBottomOf="parent"
       app:layout_constraintEnd_toEndOf="parent"
       app:layout_constraintHorizontal_bias="0.529"
       app:layout_constraintStart_toStartOf="parent"
       app:layout_constraintTop_toTopOf="parent"
       app:layout_constraintVertical_bias="0.822" />
</androidx.constraintlayout.widget.ConstraintLayout>
```

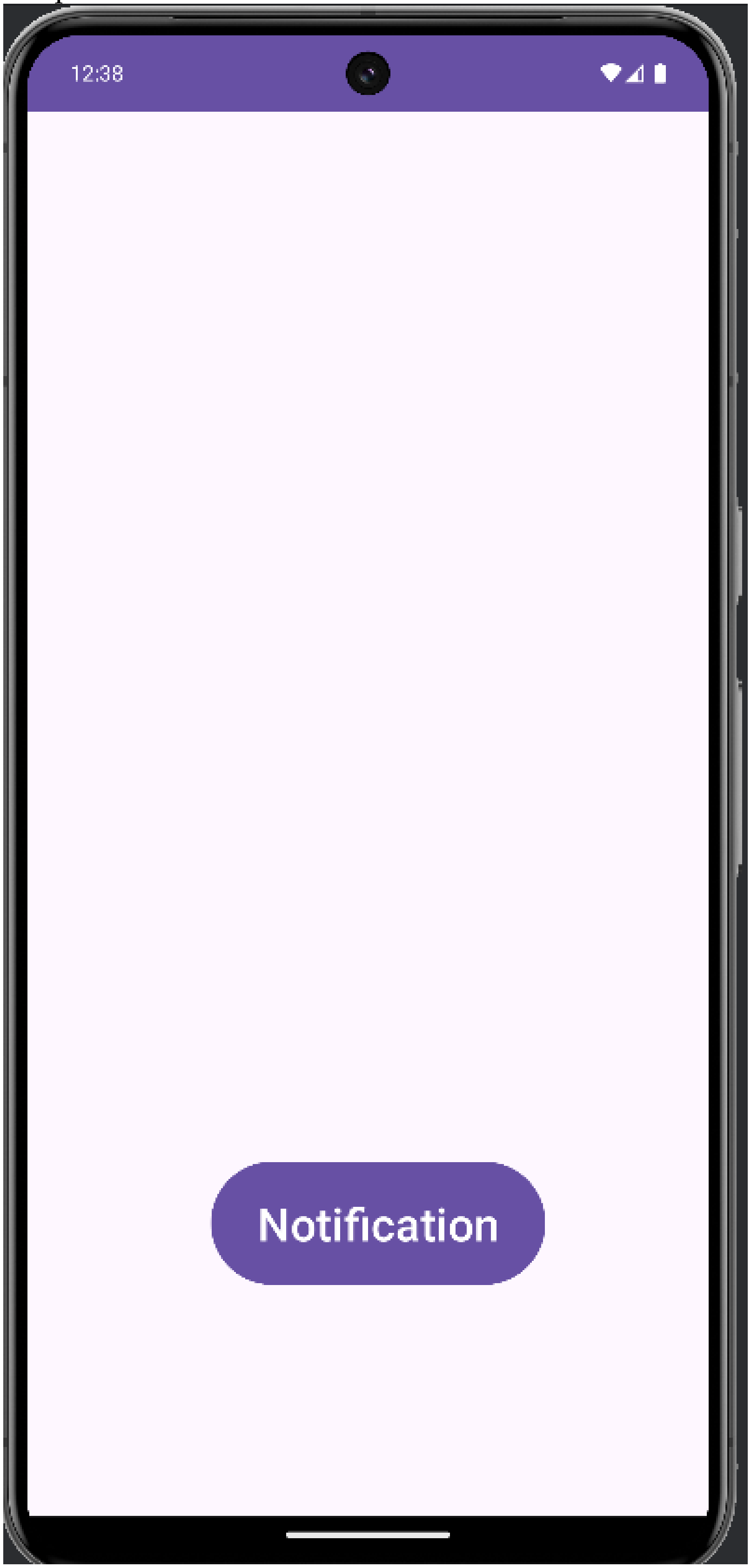
java code:

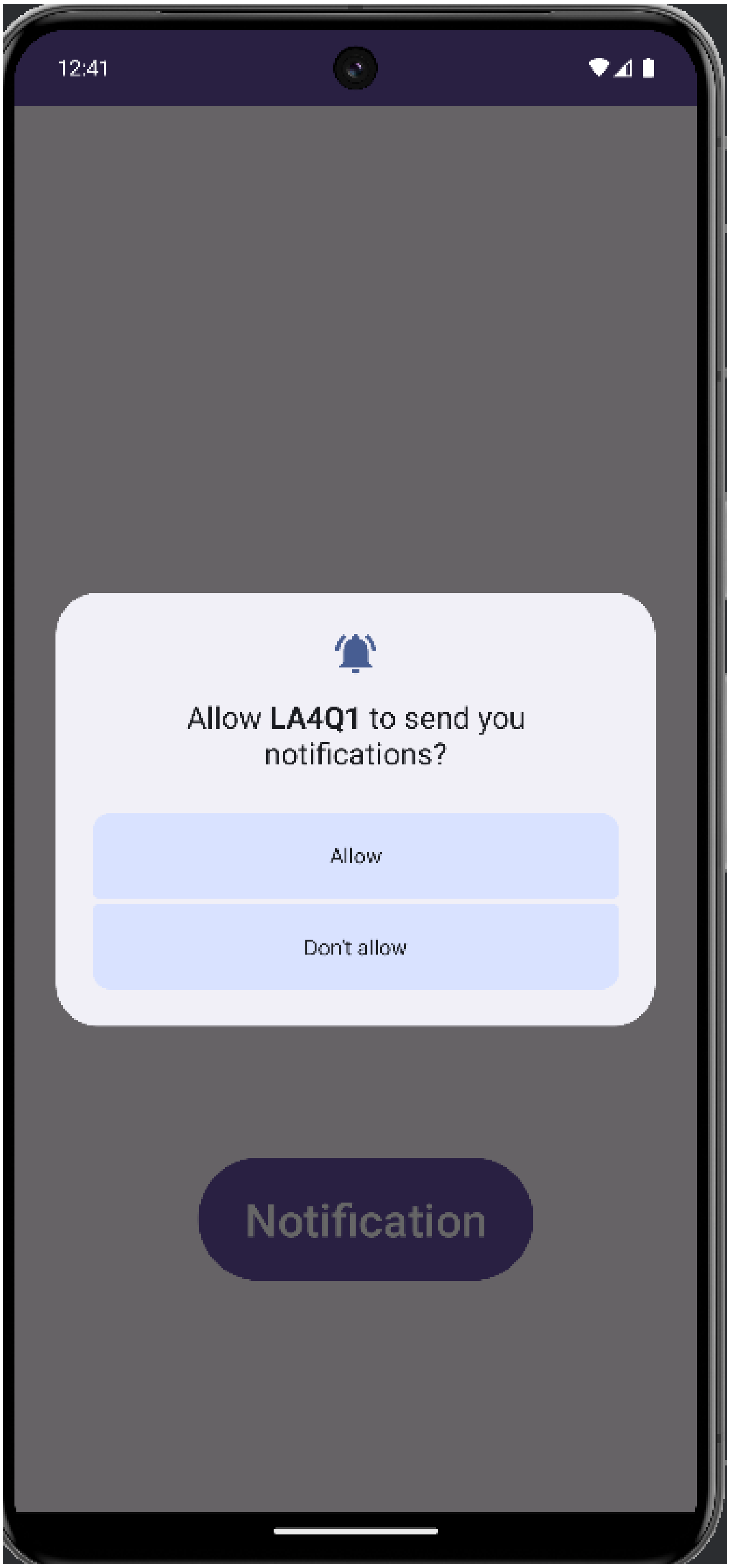
```
package com.example.la4q1;

import android.app.NotificationChannel;
import android.app.NotificationManager;
import android.content.Context;
import android.content.pm.PackageManager;
import android.os.Build;
import android.os.Bundle;
import androidx.appcompat.app.AppCompatActivity;
```



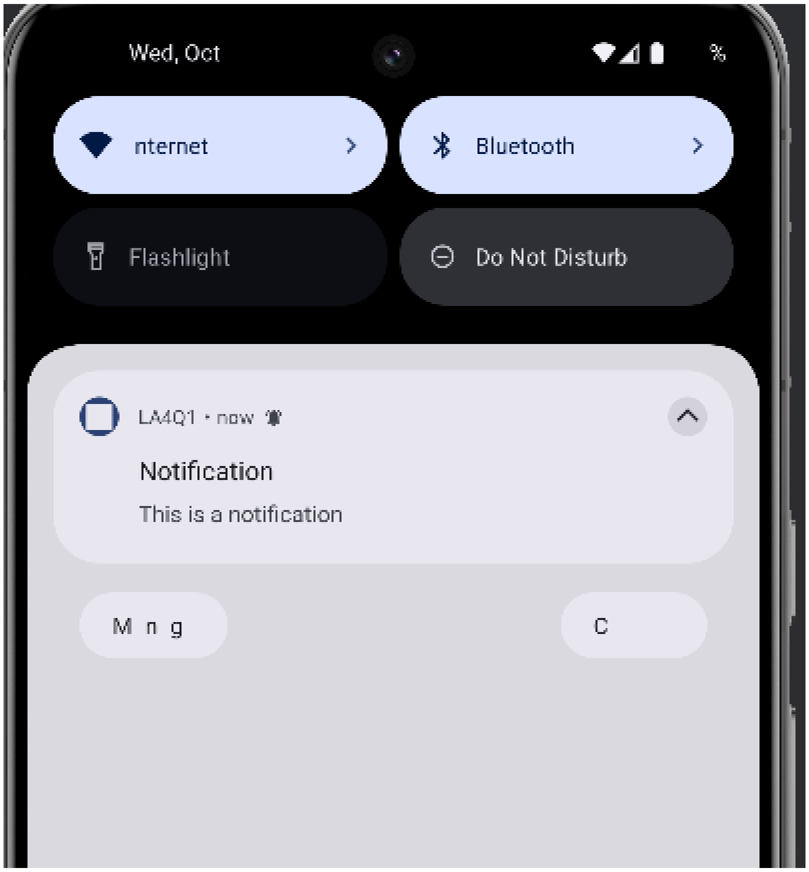
```
import androidx.core.app.NotificationCompat;
import android.view.View;
import android.widget.Button;

public class MainActivity extends AppCompatActivity {
   private static final String CHANNEL_ID = "notify_001";

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

       createNotificationChannel();
       requestNotificationPermission();

       Button notifyButton = findViewById(R.id.notifyButton);
       notifyButton.setOnClickListener(new View.OnClickListener() {
           @Override
           public void onClick(View v) {
               showNotification();
           }
       });
   }

   private void showNotification() {
       NotificationCompat.Builder builder = new
NotificationCompat.Builder(this, CHANNEL_ID)
               .setSmallIcon(R.drawable.notification)  // Ensure to add a small
icon in res/drawable
               .setContentTitle("Notification")
               .setContentText("This is a notification")
               .setPriority(NotificationCompat.PRIORITY_DEFAULT);

       NotificationManager notificationManager = (NotificationManager)
getSystemService(Context.NOTIFICATION_SERVICE);
       notificationManager.notify(0, builder.build());
   }

   private void createNotificationChannel() {
       if (Build.VERSION.SDK_INT >= Build.VERSION_CODES.O) {
           CharSequence name = "Notification Channel";
           String description = "Channel for simple notifications";
           int importance = NotificationManager.IMPORTANCE_DEFAULT;
           NotificationChannel channel = new NotificationChannel(CHANNEL_ID,
name, importance);
           channel.setDescription(description);

           NotificationManager notificationManager =
getSystemService(NotificationManager.class);
           notificationManager.createNotificationChannel(channel);
       }
   }

   private void requestNotificationPermission() {
       if (Build.VERSION.SDK_INT >= Build.VERSION_CODES.TIRAMISU) {
           if
(checkSelfPermission(android.Manifest.permission.POST_NOTIFICATIONS) !=
PackageManager.PERMISSION_GRANTED) {
               requestPermissions(new
String[]{android.Manifest.permission.POST_NOTIFICATIONS}, 1);
           }
       }
   }
}
```

Output:





Allow **LA4Q1** to send you notifications?



Notification

This is a notification

2. Design an app that triggers a basic notification with a clickable action. The notification should have a "View" button that, when clicked, opens a specific activity within the app. Use an Intent to handle the notification action, and display the action's result within the new activity.

Solution:

xml code:

activity_main:

```
<LinearLayout
   xmlns:android="http://schemas.android.com/apk/res/android"
   android:layout_width="match_parent"
   android:layout_height="match_parent"
   android:orientation="vertical"
   android:gravity="center">

   <Button
       android:id="@+id/btn_notify"
       android:layout_width="wrap_content"
       android:layout_height="wrap_content"
       android:text="Trigger Notification" />

</LinearLayout>
```

activity_view:

```
<LinearLayout
   xmlns:android="http://schemas.android.com/apk/res/android"
   android:layout_width="match_parent"
   android:layout_height="match_parent"
```

```
    android:orientation="vertical"
    android:gravity="center">

    <Button
        android:id="@+id/btn_notify"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Trigger Notification" />

</LinearLayout>
```

java code:
MainActivity:

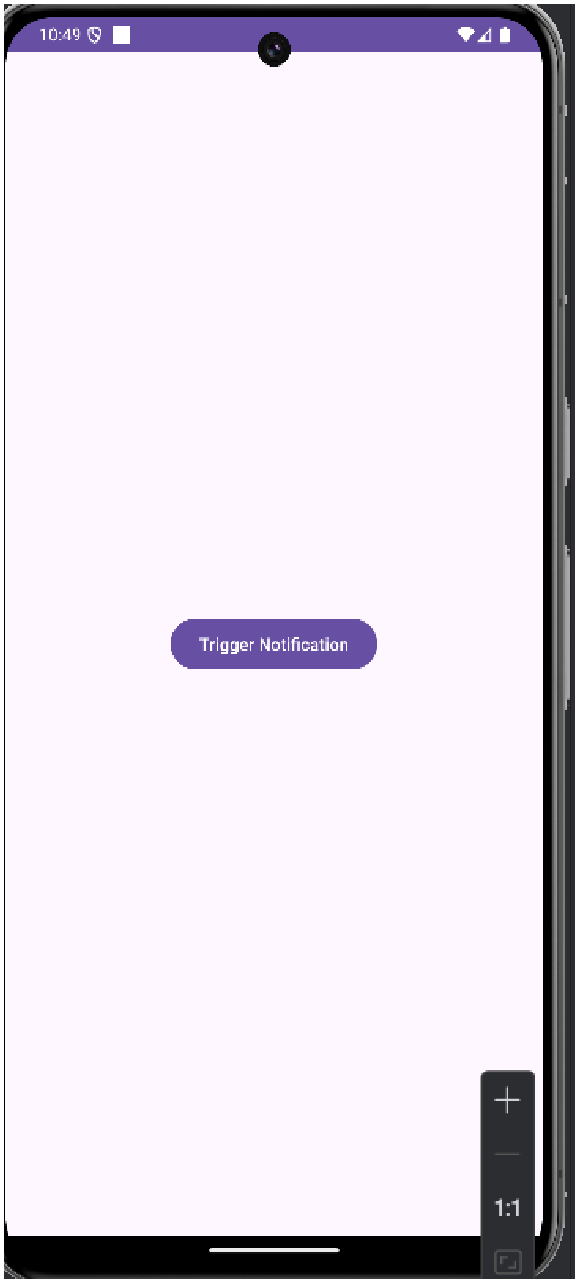
```
package com.example.la4q2;

import android.app.NotificationChannel;
import android.app.NotificationManager;
import android.app.PendingIntent;
import android.content.Context;
import android.content.Intent;
import android.os.Build;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;

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

public class MainActivity extends AppCompatActivity {

    private static final String CHANNEL_ID = "my_channel_id";

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

        Button notifyButton = findViewById(R.id.btn_notify);
        notifyButton.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                triggerNotification();
            }
        });
    }

    private void triggerNotification() {
        Intent intent = new Intent(this, ViewActivity.class);
        intent.putExtra("EXTRA_MESSAGE", "This is the action result from
notification.");

        PendingIntent pendingIntent = PendingIntent.getActivity(
                this, 0, intent, PendingIntent.FLAG_UPDATE_CURRENT |
PendingIntent.FLAG_IMMUTABLE);

        if (Build.VERSION.SDK_INT >= Build.VERSION_CODES.O) {
            NotificationChannel channel = new NotificationChannel(
                    CHANNEL_ID, "My Channel",
NotificationManager.IMPORTANCE_DEFAULT);
            NotificationManager manager =
getSystemService(NotificationManager.class);
```

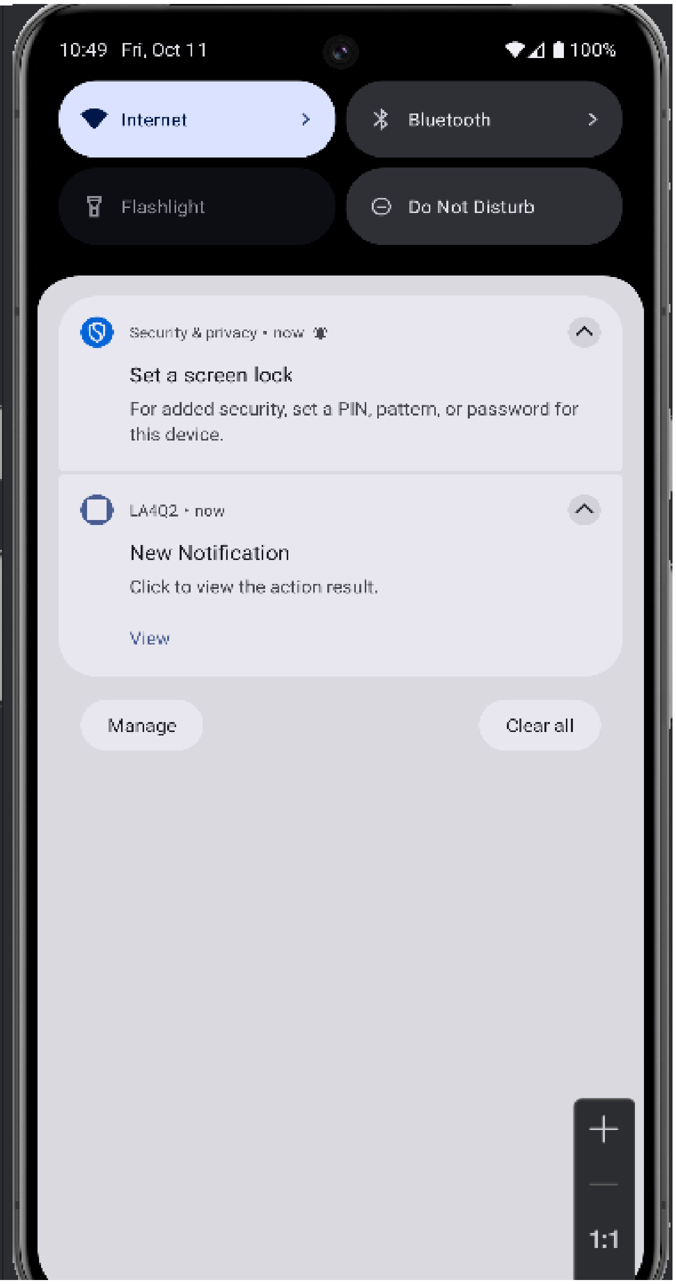
```
            manager.createNotificationChannel(channel);
        }

        NotificationCompat.Builder builder = new
NotificationCompat.Builder(this, CHANNEL_ID)
                .setSmallIcon(R.drawable.noti)
                .setContentTitle("New Notification")
                .setContentText("Click to view the action result.")
                .setPriority(NotificationCompat.PRIORITY_DEFAULT)
                .setContentIntent(pendingIntent)
                .addAction(R.drawable.hackerranklogo, "View", pendingIntent)
                .setAutoCancel(true);


        NotificationManager notificationManager = (NotificationManager)
getSystemService(Context.NOTIFICATION_SERVICE);
        notificationManager.notify(1, builder.build());
    }
}
```

ViewActivity:

```
package com.example.la4q2;

import android.os.Bundle;
import android.widget.TextView;

import androidx.appcompat.app.AppCompatActivity;

public class ViewActivity extends AppCompatActivity {

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_view);

        TextView resultText = findViewById(R.id.tv_result);


        String message = getIntent().getStringExtra("EXTRA_MESSAGE");
        if (message != null) {
            resultText.setText(message);
        }
    }
}
```

Output:





Security & privacy • now

Set a screen lock

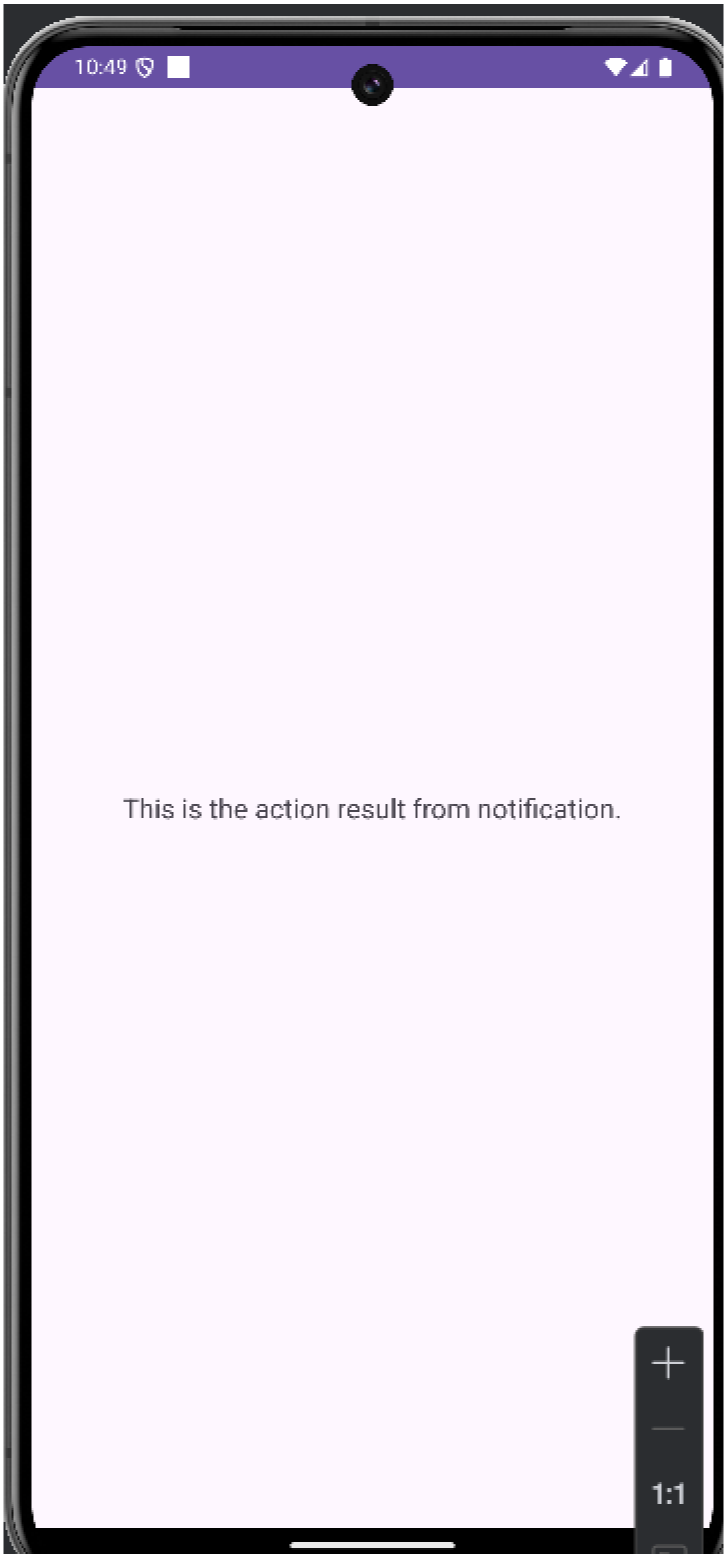
For added security, set a PIN, pattern, or password for this device.

LA4Q2 • now

New Notification

Click to view the action result.

View


This is the action result from notification.

3. Create an Android application that triggers a simple notification when a button is clicked. Use the NotificationCompat.Builder class to build the notification and set its properties, such as title, text, and icon. Ensure that the notification appears in the status bar and can be expanded to show additional content.

Solution:

xml code:

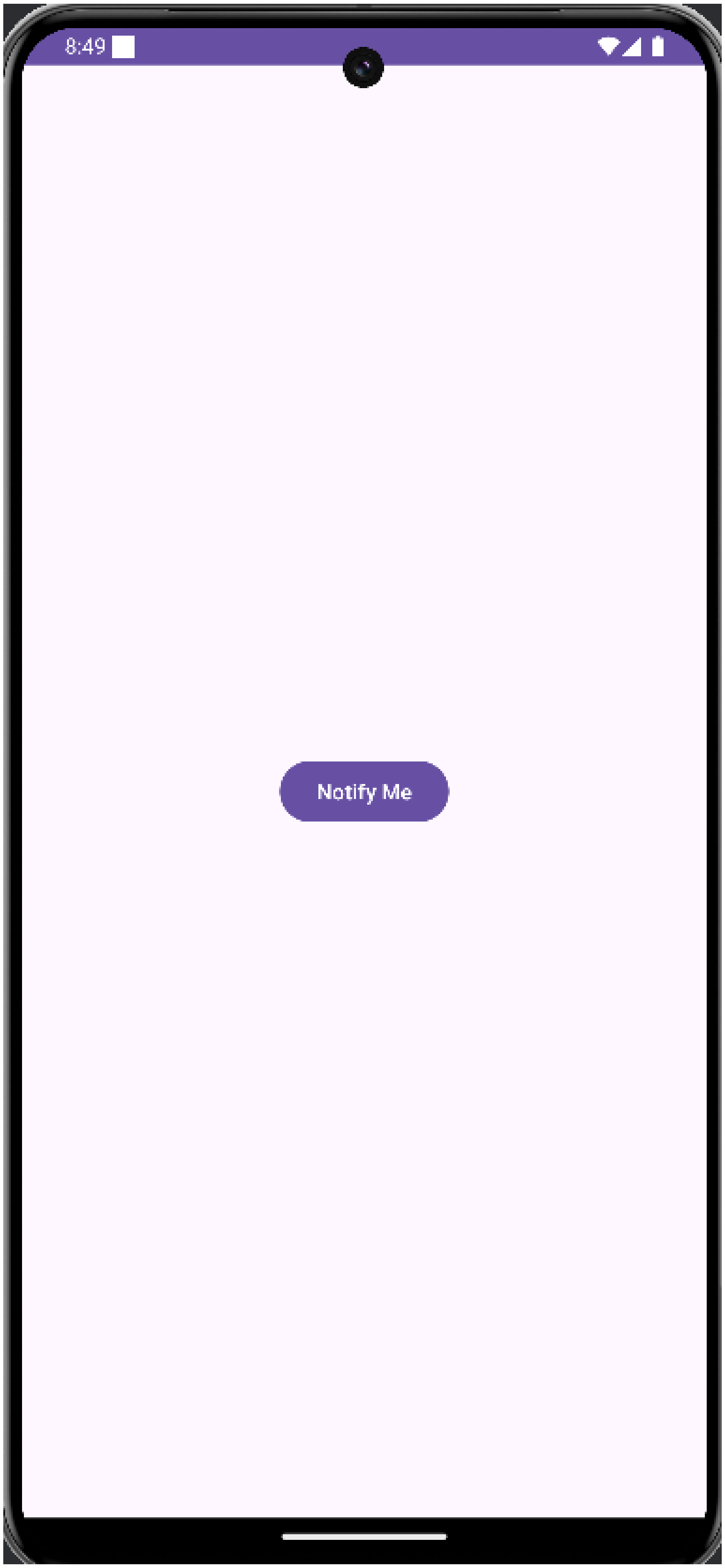
```
<?xml version="1.0" encoding="utf-8"?>
```

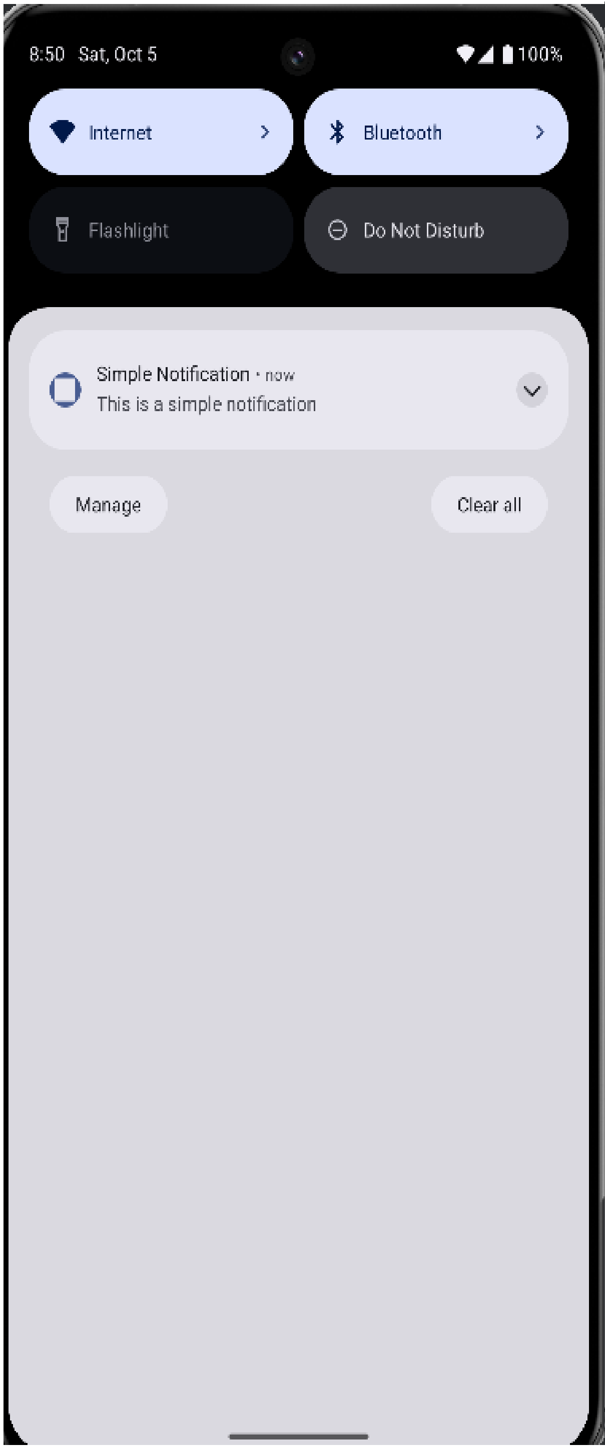
```
<androidx.constraintlayout.widget.ConstraintLayout
xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".MainActivity">

    <Button
        android:id="@+id/notifyButton"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Notify Me"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent" />

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

java code:

```
package com.example.la4q3;

import android.app.NotificationChannel;
import android.app.NotificationManager;
import android.app.PendingIntent;
import android.content.Intent;
import android.content.pm.PackageManager;
import android.os.Build;
import android.os.Bundle;
import androidx.appcompat.app.AppCompatActivity;
import androidx.core.app.ActivityCompat;
import androidx.core.app.NotificationCompat;
import androidx.core.app.NotificationManagerCompat;

import android.view.View;
import android.widget.Button;

public class MainActivity extends AppCompatActivity {

    private final String CHANNEL_ID = "channel_id_example";
    private final int NOTIFICATION_ID = 001;

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

        createNotificationChannel();

        Button notifyButton = findViewById(R.id.notifyButton);
        notifyButton.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                triggerNotification();
            }
        });
    }

    private void triggerNotification() {
        Intent intent = new Intent(this, MainActivity.class);
        intent.setFlags(Intent.FLAG_ACTIVITY_NEW_TASK |
Intent.FLAG_ACTIVITY_CLEAR_TASK);
        PendingIntent pendingIntent = PendingIntent.getActivity(this, 0, intent,
PendingIntent.FLAG_IMMUTABLE);

        NotificationCompat.Builder builder = new
NotificationCompat.Builder(this, CHANNEL_ID)
                .setSmallIcon(R.drawable.notification)
                .setContentTitle("Simple Notification")
                .setContentText("This is a simple notification")
                .setStyle(new NotificationCompat.BigTextStyle()
                        .bigText("This is an expandable notification. Android
programming is the best"))
                .setPriority(NotificationCompat.PRIORITY_DEFAULT)
                .setContentIntent(pendingIntent)
                .setAutoCancel(true);

        NotificationManagerCompat notificationManager =
NotificationManagerCompat.from(this);
        if (ActivityCompat.checkSelfPermission(this,
android.Manifest.permission.POST_NOTIFICATIONS) !=
PackageManager.PERMISSION_GRANTED) {

            return;
        }
        notificationManager.notify(NOTIFICATION_ID, builder.build());
    }

    private void createNotificationChannel() {
        if (Build.VERSION.SDK_INT >= Build.VERSION_CODES.O) {
            CharSequence name = "Example Channel";
            String description = "This is a channel for example notifications";
            int importance = NotificationManager.IMPORTANCE_DEFAULT;
            NotificationChannel channel = new NotificationChannel(CHANNEL_ID,
name, importance);
            channel.setDescription(description);

            NotificationManager notificationManager =
getSystemService(NotificationManager.class);
            notificationManager.createNotificationChannel(channel);
        }
    }
}
```

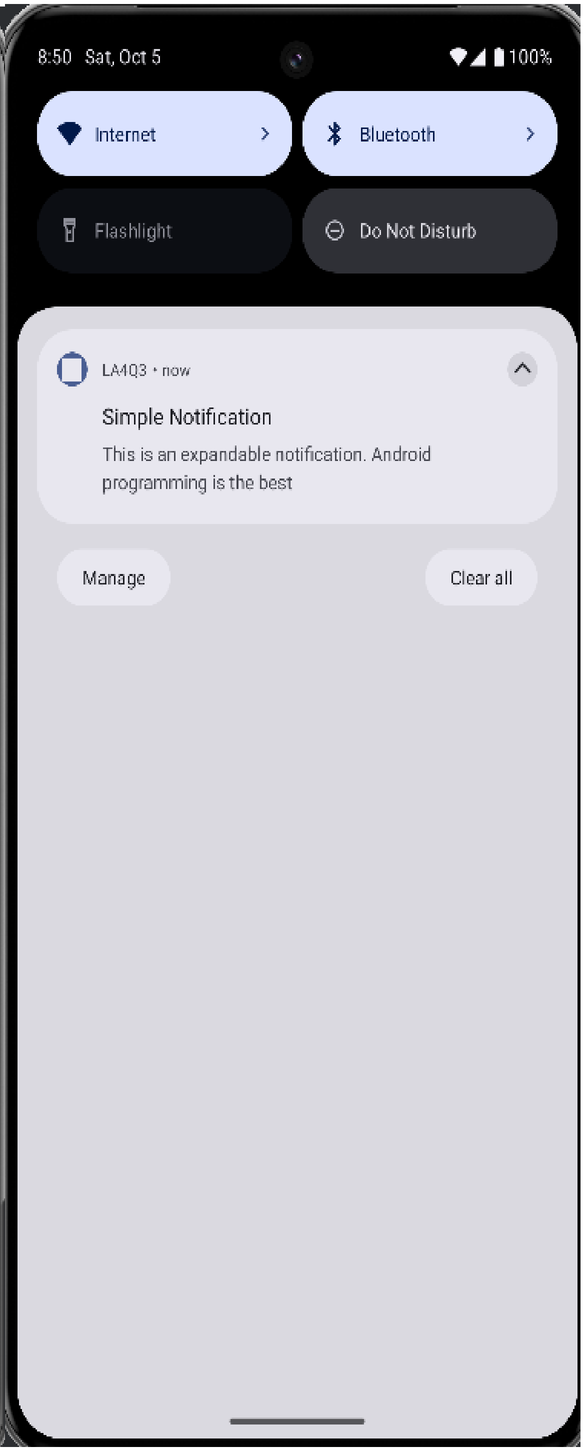
Output:


8:49
Notify Me



Simple Notification • now

This is a simple notification



LA4Q3 • now

Simple Notification

This is an expandable notification. Android programming is the best

4. Build an application that generates a notification with custom properties such as sound, vibration, and LED light color. Use the NotificationCompat.Builder class to set these properties. The app should allow the user to configure these properties through a settings screen and preview the notification with the chosen settings.

Solution:

xml code:

activity_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">

   <Button
       android:id="@+id/openSettingsButton"
       android:layout_width="wrap_content"
       android:layout_height="wrap_content"
       android:text="Open Settings"
       app:layout_constraintBottom_toTopOf="@+id/previewButton"
       app:layout_constraintEnd_toEndOf="parent"
       app:layout_constraintHorizontal_bias="0.498"
       app:layout_constraintStart_toStartOf="parent"
       app:layout_constraintTop_toTopOf="parent"
       app:layout_constraintVertical_bias="0.605" />

   <Button
       android:id="@+id/previewButton"
       android:layout_width="199dp"
       android:layout_height="77dp"
       android:layout_marginBottom="204dp"
       android:text="Preview Notification"
       app:layout_constraintBottom_toBottomOf="parent"
       app:layout_constraintEnd_toEndOf="parent"
       app:layout_constraintHorizontal_bias="0.497"
       app:layout_constraintStart_toStartOf="parent" />
</androidx.constraintlayout.widget.ConstraintLayout>
```

activity_settings:

```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=".SettingsActivity">

   <androidx.appcompat.widget.SwitchCompat
       android:id="@+id/vibrationSwitch"
       android:layout_width="wrap_content"
       android:layout_height="wrap_content"
       android:checked="true"
       android:minHeight="32dp"
       android:text="@string/enable_vibration"
```

```
        app:layout_constraintTop_toTopOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        android:padding="16dp" />


    <Button
        android:id="@+id/chooseSoundButton"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="@string/choose_notification_sound"
        app:layout_constraintTop_toBottomOf="@+id/vibrationSwitch"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        android:padding="16dp"
        app:layout_constraintHorizontal_bias="0.5" />

    <Button
        android:id="@+id/chooseLedColorButton"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="@string/choose_led_color"
        app:layout_constraintTop_toBottomOf="@+id/chooseSoundButton"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        android:padding="16dp"
        app:layout_constraintHorizontal_bias="0.5" />

    <Button
        android:id="@+id/previewNotificationButton"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Preview Notification"
        app:layout_constraintTop_toBottomOf="@+id/chooseLedColorButton"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintEnd_toEndOf="parent"/>


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

java code:
MainActivity:

```
package com.example.la4q4;

import android.app.NotificationChannel;
import android.app.NotificationManager;
import android.app.PendingIntent;
import android.content.Intent;
import android.content.SharedPreferences;
import android.content.pm.PackageManager;
import android.graphics.Color;
import android.media.RingtoneManager;
import android.net.Uri;
import android.os.Build;
import android.os.Bundle;
import androidx.appcompat.app.AppCompatActivity;
import androidx.core.app.ActivityCompat;
import androidx.core.app.NotificationCompat;
import androidx.core.app.NotificationManagerCompat;
import android.view.View;
import android.widget.Button;
```

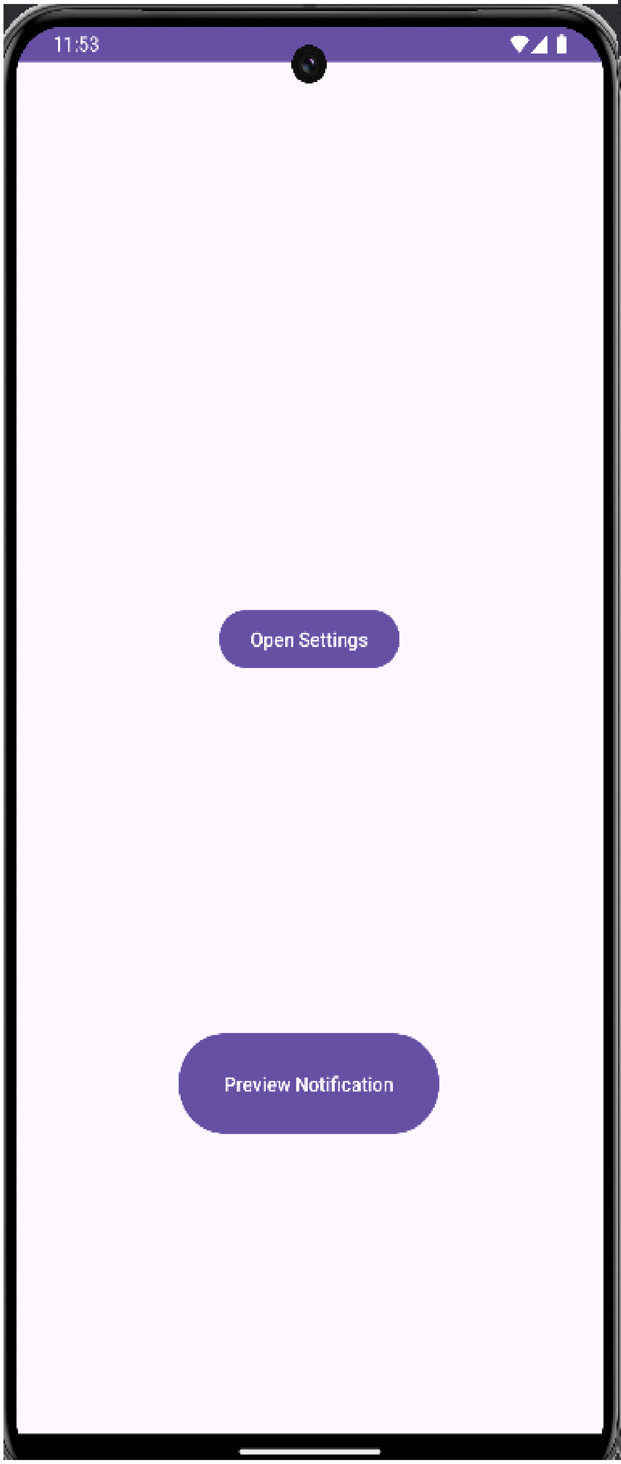
```
public class MainActivity extends AppCompatActivity {

   private final String CHANNEL_ID = "custom_channel_id";
   private final int NOTIFICATION_ID = 100;
   SharedPreferences sharedPreferences;

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

       sharedPreferences = getSharedPreferences("notification_settings",
MODE_PRIVATE);

       Button openSettingsButton = findViewById(R.id.openSettingsButton);
       Button previewButton = findViewById(R.id.previewButton);

       openSettingsButton.setOnClickListener(new View.OnClickListener() {
           @Override
           public void onClick(View v) {
               Intent intent = new Intent(MainActivity.this,
SettingsActivity.class);
               startActivity(intent);
           }
       });

       previewButton.setOnClickListener(new View.OnClickListener() {
           @Override
           public void onClick(View v) {
               triggerNotification();
           }
       });

       createNotificationChannel();
   }

   private void triggerNotification() {
       boolean vibrationEnabled = sharedPreferences.getBoolean("vibration",
true);
       String soundUriString = sharedPreferences.getString("sound",
RingtoneManager.getDefaultUri(RingtoneManager.TYPE_NOTIFICATION).toString());
       int ledColor = sharedPreferences.getInt("led_color", Color.BLUE);

       Uri soundUri = Uri.parse(soundUriString);

       Intent intent = new Intent(this, MainActivity.class);
       PendingIntent pendingIntent = PendingIntent.getActivity(this, 0, intent,
PendingIntent.FLAG_IMMUTABLE);

       NotificationCompat.Builder builder = new
NotificationCompat.Builder(this, CHANNEL_ID)
               .setSmallIcon(R.drawable.notification)
               .setContentTitle("Custom Notification")
               .setContentText("This notification has custom settings")
               .setSound(soundUri)
               .setLights(ledColor, 1000, 1000)
               .setContentIntent(pendingIntent)
               .setAutoCancel(true);

       if (vibrationEnabled) {
           long[] vibrationPattern = {0, 500, 500, 500};
           builder.setVibrate(vibrationPattern);
       }
```

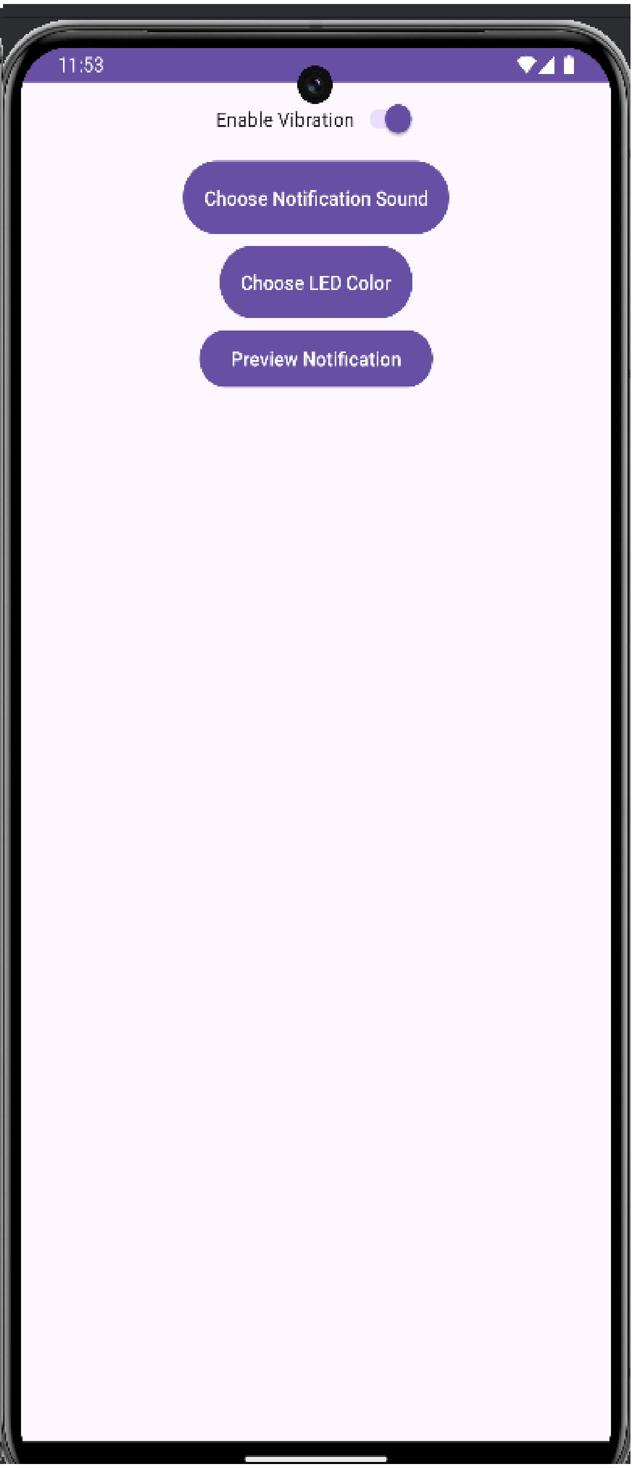
```
        NotificationManagerCompat notificationManager =
NotificationManagerCompat.from(this);
        if (ActivityCompat.checkSelfPermission(this,
android.Manifest.permission.POST_NOTIFICATIONS) !=
PackageManager.PERMISSION_GRANTED) {
            return;
        }
        notificationManager.notify(NOTIFICATION_ID, builder.build());
    }

    private void createNotificationChannel() {
        if (Build.VERSION.SDK_INT >= Build.VERSION_CODES.O) {
            String name = "Custom Channel";
            String description = "This channel is used for custom
notifications";
            int importance = NotificationManager.IMPORTANCE_DEFAULT;
            NotificationChannel channel = new NotificationChannel(CHANNEL_ID,
name, importance);
            channel.setDescription(description);

            NotificationManager notificationManager =
getSystemService(NotificationManager.class);
            notificationManager.createNotificationChannel(channel);
        }
    }
}
```

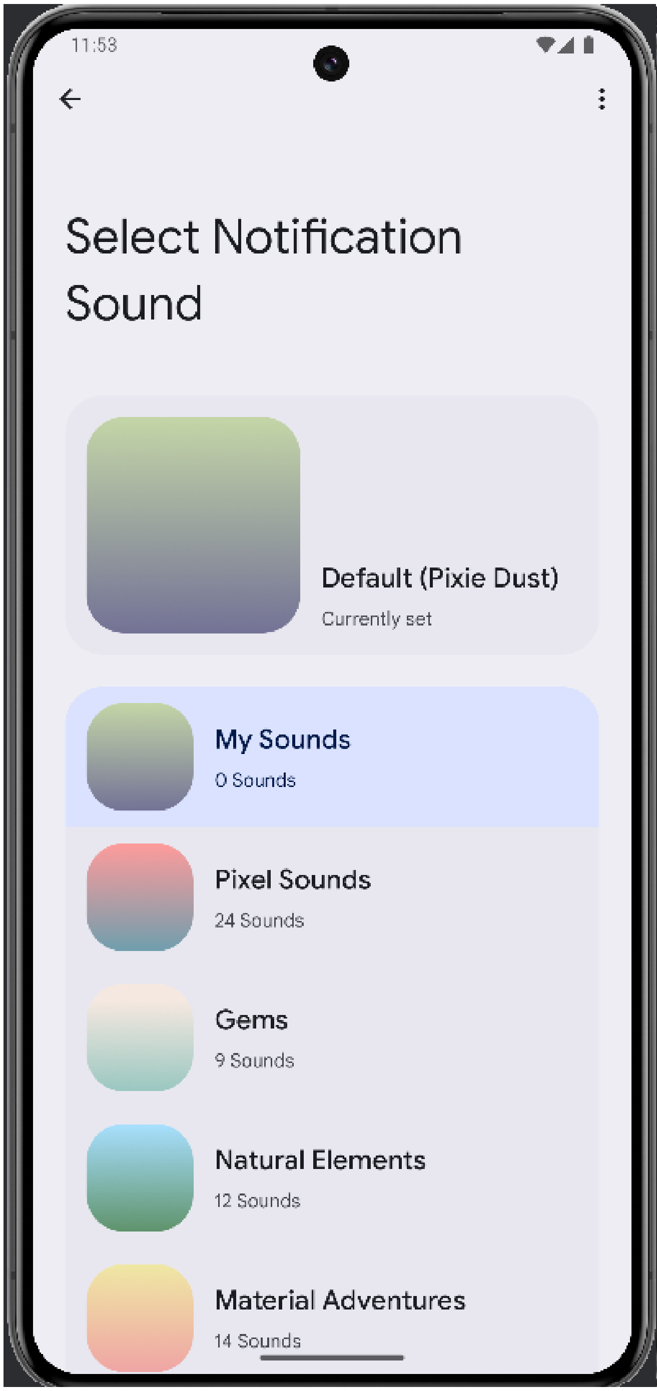
SettingsActivity:

```
package com.example.la4q4;

import android.app.NotificationChannel;
import android.app.NotificationManager;
import android.content.Context;
import android.content.SharedPreferences;
import android.graphics.Color;
import android.media.RingtoneManager;
import android.net.Uri;
import android.os.Build;
import android.os.Bundle;
import android.content.Intent;
import android.widget.Toast;
import androidx.appcompat.app.AppCompatActivity;
import androidx.appcompat.widget.SwitchCompat;
import androidx.core.app.NotificationCompat;
import androidx.activity.result.ActivityResultLauncher;
import androidx.activity.result.contract.ActivityResultContracts;
import android.view.MenuItem;
import android.widget.Button;

import com.skydoves.colorpickerview.ColorEnvelope;
import com.skydoves.colorpickerview.ColorPickerDialog;
import com.skydoves.colorpickerview.listeners.ColorEnvelopeListener;


public class SettingsActivity extends AppCompatActivity {

    private Button selectSoundButton, selectColorButton, previewButton;
    private SwitchCompat vibrationSwitch;
    private Uri selectedSound;
    private int selectedLedColor;
    private boolean vibrationEnabled;
    private ActivityResultLauncher<Intent> ringtonePickerLauncher;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_settings);

        if (getSupportActionBar() != null) {
            getSupportActionBar().setDisplayHomeAsUpEnabled(true);
        }

        selectSoundButton = findViewById(R.id.chooseSoundButton);
        selectColorButton = findViewById(R.id.chooseLedColorButton);
        previewButton = findViewById(R.id.previewNotificationButton);
        vibrationSwitch = findViewById(R.id.vibrationSwitch);

        SharedPreferences preferences =
getSharedPreferences("NotificationPrefs", MODE_PRIVATE);
        selectedSound = Uri.parse(preferences.getString("sound",
RingtoneManager.getDefaultUri(RingtoneManager.TYPE_NOTIFICATION).toString()));
        selectedLedColor = preferences.getInt("ledColor", Color.RED);
        vibrationEnabled = preferences.getBoolean("vibration", true);

        vibrationSwitch.setChecked(vibrationEnabled);

        ringtonePickerLauncher = registerForActivityResult(new
ActivityResultContracts.StartActivityForResult(), result -> {
            if (result.getResultCode() == RESULT_OK && result.getData() != null)
{
                selectedSound =
result.getData().getParcelableExtra(RingtoneManager.EXTRA_RINGTONE_PICKED_URI);
                if (selectedSound != null) {
                    Toast.makeText(SettingsActivity.this, "Sound selected",
Toast.LENGTH_SHORT).show();
                }
            }
        });

        selectSoundButton.setOnClickListener(v -> {
            Intent intent = new Intent(RingtoneManager.ACTION_RINGTONE_PICKER);
            intent.putExtra(RingtoneManager.EXTRA_RINGTONE_TYPE,
RingtoneManager.TYPE_NOTIFICATION);
            intent.putExtra(RingtoneManager.EXTRA_RINGTONE_TITLE, "Select
Notification Sound");
            intent.putExtra(RingtoneManager.EXTRA_RINGTONE_EXISTING_URI,
selectedSound);
            ringtonePickerLauncher.launch(intent);
        });

        selectColorButton.setOnClickListener(v -> {
            new ColorPickerDialog.Builder(this)
                    .setTitle("Pick LED Color")
                    .setPreferenceName("LEDColorPicker")
                    .setPositiveButton("Select", new ColorEnvelopeListener() {
                        @Override
                        public void onColorSelected(ColorEnvelope envelope,
boolean fromUser) {
                            selectedLedColor = envelope.getColor();
                            Toast.makeText(SettingsActivity.this, "Color
selected", Toast.LENGTH_SHORT).show();
                        }
                    })
                    .setNegativeButton("Cancel", (dialogInterface, i) ->
dialogInterface.dismiss())
                    .attachAlphaSlideBar(false)
                    .attachBrightnessSlideBar(true)
                    .show();
        });

        previewButton.setOnClickListener(v -> {
            vibrationEnabled = vibrationSwitch.isChecked();

            SharedPreferences.Editor editor = preferences.edit();
            editor.putString("sound", selectedSound.toString());
            editor.putInt("ledColor", selectedLedColor);
            editor.putBoolean("vibration", vibrationEnabled);
            editor.apply();

            showNotification();
        });
    }

    private void showNotification() {
        NotificationManager notificationManager = (NotificationManager)
getSystemService(Context.NOTIFICATION_SERVICE);

        if (Build.VERSION.SDK_INT >= Build.VERSION_CODES.O) {
            NotificationChannel channel = new
NotificationChannel("default_channel", "Default Channel",
                    NotificationManager.IMPORTANCE_DEFAULT);
            channel.setDescription("Notification Preview");
            notificationManager.createNotificationChannel(channel);
        }

        NotificationCompat.Builder builder = new
NotificationCompat.Builder(this, "default_channel")
                .setSmallIcon(R.drawable.notification)
                .setContentTitle("Notification Preview")
                .setContentText("This is a preview of your custom
notification.")
                .setAutoCancel(true);

        if (vibrationEnabled) {
            long[] vibrationPattern = {0, 500, 1000};
            builder.setVibrate(vibrationPattern);
        }

        builder.setLights(selectedLedColor, 3000, 3000);

        if (selectedSound != null) {
            builder.setSound(selectedSound);
        }

        notificationManager.notify(1, builder.build());
    }

    @Override
    public boolean onOptionsItemSelected(MenuItem item) {
        if (item.getItemId() == android.R.id.home) {
            finish();
            return true;
        }
        return super.onOptionsItemSelected(item);
    }
}
```

Output:


11:53
Open Settings
Preview Notification


11:53
Enable Vibration
Choose Notification Sound
Choose LED Color
Preview Notification



11:53

Select Notification Sound

Default (Pixie Dust)
Currently set

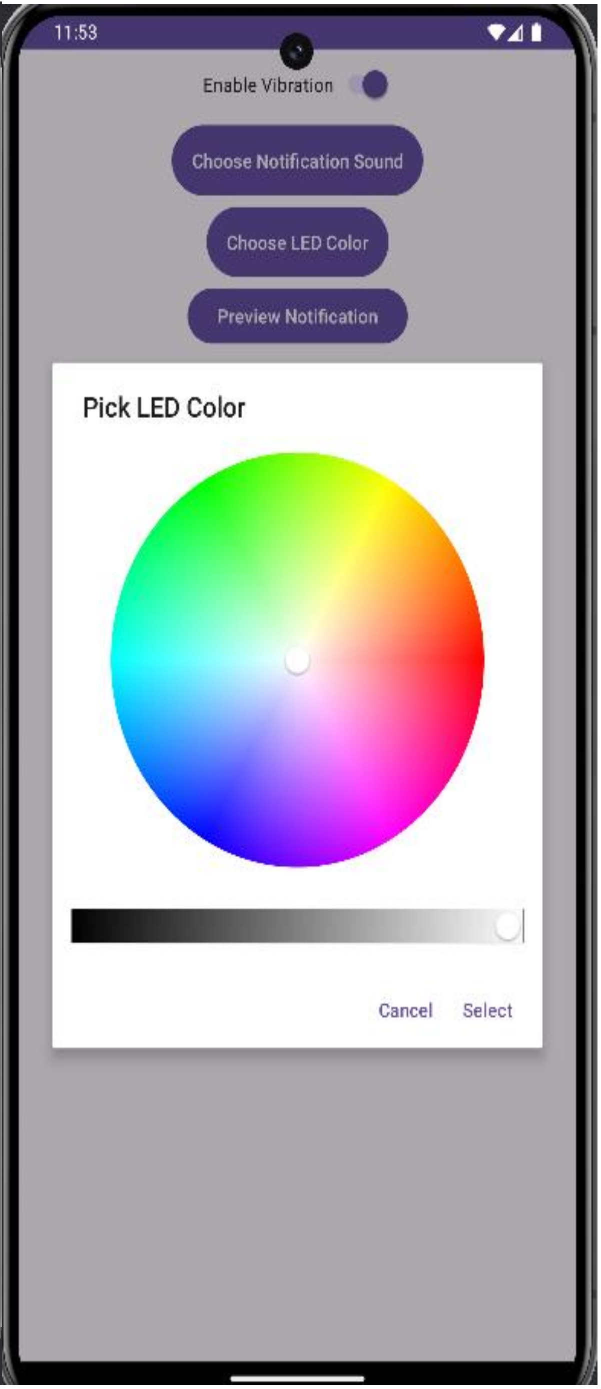
My Sounds
0 Sounds

Pixel Sounds
24 Sounds

Gems
9 Sounds

Natural Elements
12 Sounds

Material Adventures
14 Sounds



11:53

Enable Vibration

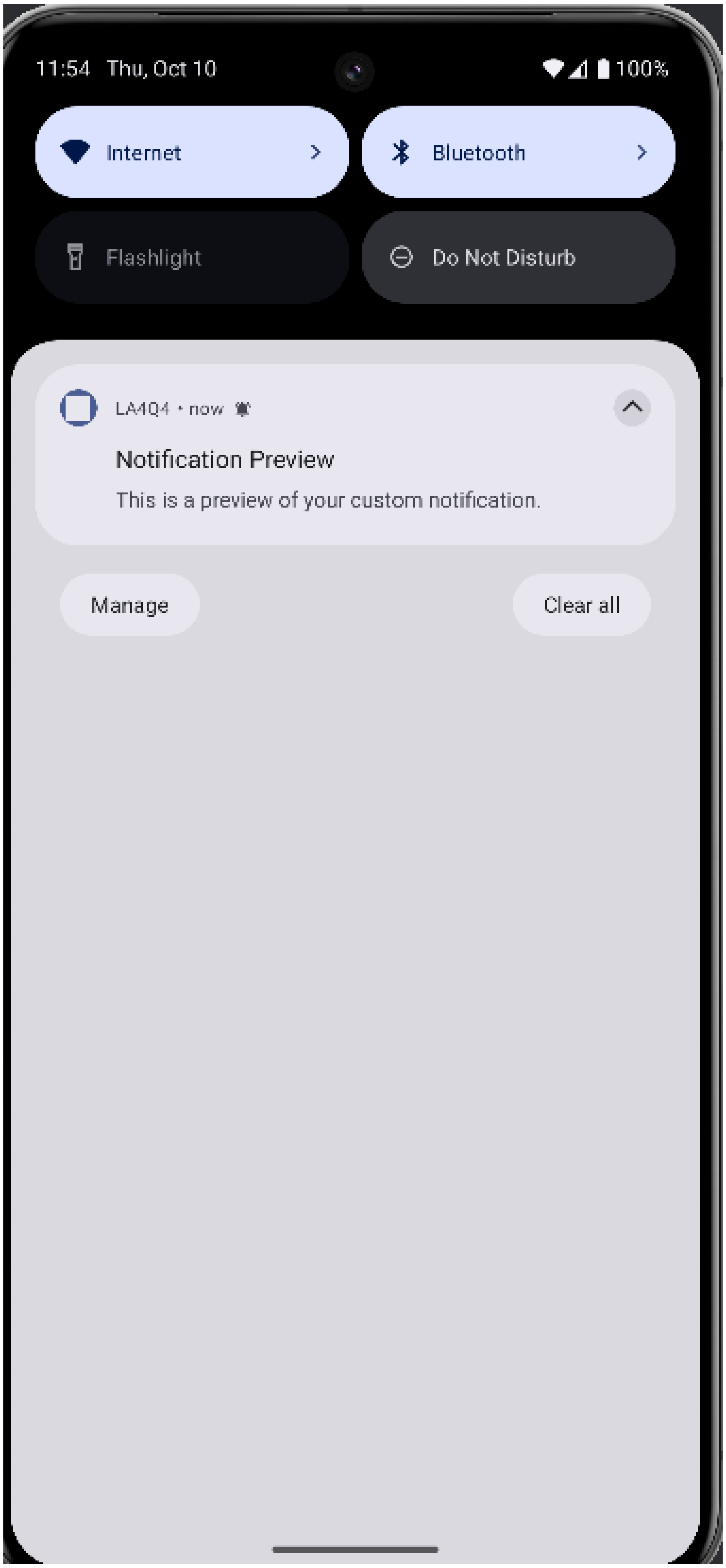
Choose Notification Sound

Choose LED Color

Preview Notification

Pick LED Color

Cancel Select



Notification Preview

This is a preview of your custom notification.

5. Create a notification that includes action buttons. For example, build a media player notification with "Play", and "Stop" buttons. Use the "Pause", NotificationCompat.Builder class to attach these actions and handle the corresponding intents when the user interacts with the notification.

Solution:

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"
   android:gravity="center">

   <Button
       android:id="@+id/showNotificationButton"
       android:layout_width="wrap_content"
       android:layout_height="wrap_content"
       android:text="Show Notification" />
</LinearLayout>
```

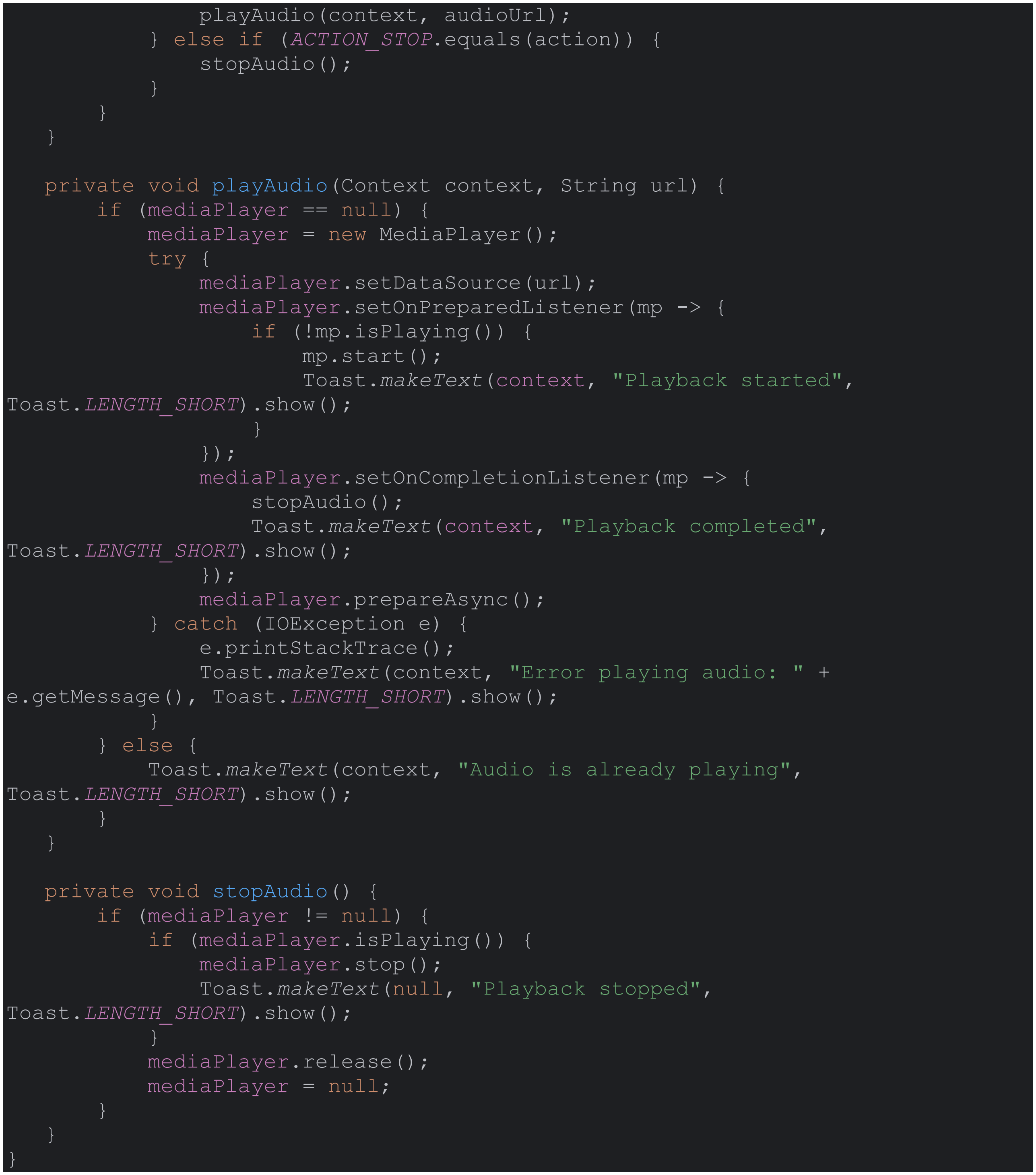
java code:

ActivityMain:

```
package com.example.la4q5;

import android.app.NotificationChannel;
import android.app.NotificationManager;
import android.app.PendingIntent;
import android.content.Context;
import android.content.Intent;
import android.os.Build;
import android.os.Bundle;
import androidx.appcompat.app.AppCompatActivity;
import androidx.core.app.NotificationCompat;
import android.view.View;
import android.widget.Button;

public class MainActivity extends AppCompatActivity {

   private static final String CHANNEL_ID = "media_channel";

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

       Button showNotificationButton =
findViewById(R.id.showNotificationButton);
       showNotificationButton.setOnClickListener(new View.OnClickListener() {
           @Override
           public void onClick(View v) {
               showNotification(MainActivity.this);
           }
       });
   }

   public void showNotification(Context context) {
       Intent playIntent = new Intent(context, MediaPlayerReceiver.class);
       playIntent.setAction(MediaPlayerReceiver.ACTION_PLAY);
       PendingIntent playPendingIntent = PendingIntent.getBroadcast(context, 0,
playIntent, PendingIntent.FLAG_UPDATE_CURRENT | PendingIntent.FLAG_IMMUTABLE);


       Intent stopIntent = new Intent(context, MediaPlayerReceiver.class);
       stopIntent.setAction(MediaPlayerReceiver.ACTION_STOP);
       PendingIntent stopPendingIntent = PendingIntent.getBroadcast(context, 1,
stopIntent, PendingIntent.FLAG_UPDATE_CURRENT | PendingIntent.FLAG_IMMUTABLE);


       if (Build.VERSION.SDK_INT >= Build.VERSION_CODES.O) {
           NotificationChannel channel = new NotificationChannel(
                   CHANNEL_ID, "Media Player",
                   NotificationManager.IMPORTANCE_LOW
           );
           NotificationManager notificationManager = (NotificationManager)
context.getSystemService(Context.NOTIFICATION_SERVICE);
           notificationManager.createNotificationChannel(channel);
       }

       NotificationCompat.Builder builder = new
NotificationCompat.Builder(context, CHANNEL_ID)
               .setSmallIcon(R.drawable.media)
               .setContentTitle("Media Player")
               .setContentText("Now playing music")
               .setPriority(NotificationCompat.PRIORITY_LOW)
               .addAction(R.drawable.play, "Play", playPendingIntent)
               .addAction(R.drawable.stop, "Stop", stopPendingIntent)
               .setAutoCancel(true);

       NotificationManager notificationManager = (NotificationManager)
context.getSystemService(Context.NOTIFICATION_SERVICE);
       notificationManager.notify(1, builder.build());
   }
}
```

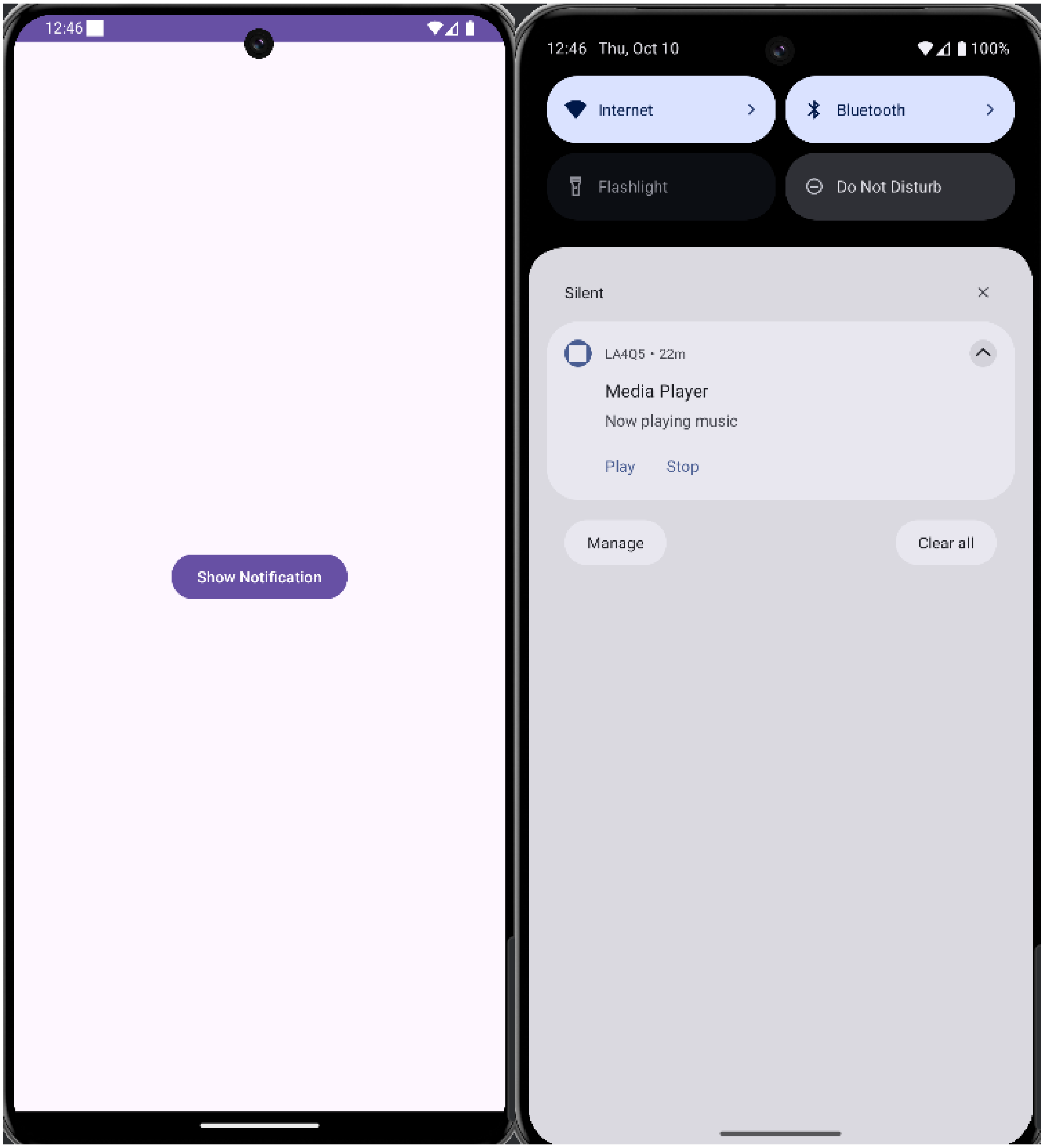
MediaPlayerReceiver:

```
package com.example.la4q5;

import android.content.BroadcastReceiver;
import android.content.Context;
import android.content.Intent;
import android.media.MediaPlayer;
import android.widget.Toast;
import java.io.IOException;

public class MediaPlayerReceiver extends BroadcastReceiver {
    public static final String ACTION_PLAY =
"com.example.mediaplayernotification.ACTION_PLAY";
    public static final String ACTION_STOP =
"com.example.mediaplayernotification.ACTION_STOP";

    private MediaPlayer mediaPlayer;

    @Override
    public void onReceive(Context context, Intent intent) {
        if (intent != null) {
            String action = intent.getAction();
            if (ACTION_PLAY.equals(action)) {
                String audioUrl =
"https://drive.google.com/uc?id=1l2QjyGJrunx_32vu243JdtO5tQxiZvMy";
```



```
                playAudio(context, audioUrl);
            } else if (ACTION_STOP.equals(action)) {
                stopAudio();
            }
        }
    }

    private void playAudio(Context context, String url) {
        if (mediaPlayer == null) {
            mediaPlayer = new MediaPlayer();
            try {
                mediaPlayer.setDataSource(url);
                mediaPlayer.setOnPreparedListener(mp -> {
                    if (!mp.isPlaying()) {
                        mp.start();
                        Toast.makeText(context, "Playback started",
Toast.LENGTH_SHORT).show();
                    }
                });
                mediaPlayer.setOnCompletionListener(mp -> {
                    stopAudio();
                    Toast.makeText(context, "Playback completed",
Toast.LENGTH_SHORT).show();
                });
                mediaPlayer.prepareAsync();
            } catch (IOException e) {
                e.printStackTrace();
                Toast.makeText(context, "Error playing audio: " +
e.getMessage(), Toast.LENGTH_SHORT).show();
            }
        } else {
            Toast.makeText(context, "Audio is already playing",
Toast.LENGTH_SHORT).show();
        }
    }

    private void stopAudio() {
        if (mediaPlayer != null) {
            if (mediaPlayer.isPlaying()) {
                mediaPlayer.stop();
                Toast.makeText(null, "Playback stopped",
Toast.LENGTH_SHORT).show();
            }
            mediaPlayer.release();
            mediaPlayer = null;
        }
    }
}
```

Output:



Media Player

Now playing music

6. Develop an app that triggers should display a "Big large Picture Style" image notification. The notification expanded. Use when NotificationCompat. BigPictureStyle to implement the expanded notification and ensure it includes both a title and a summary text when collapsed.

Solution:

xml code:

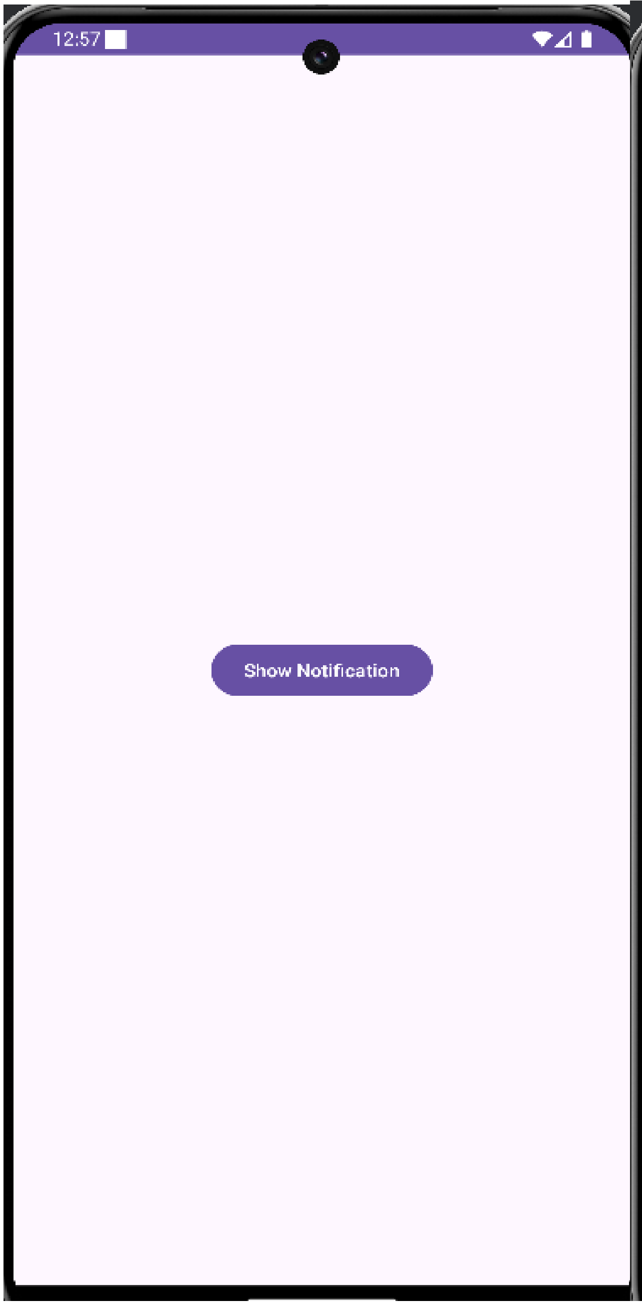
```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
   android:layout_width="match_parent"
   android:layout_height="match_parent">
```

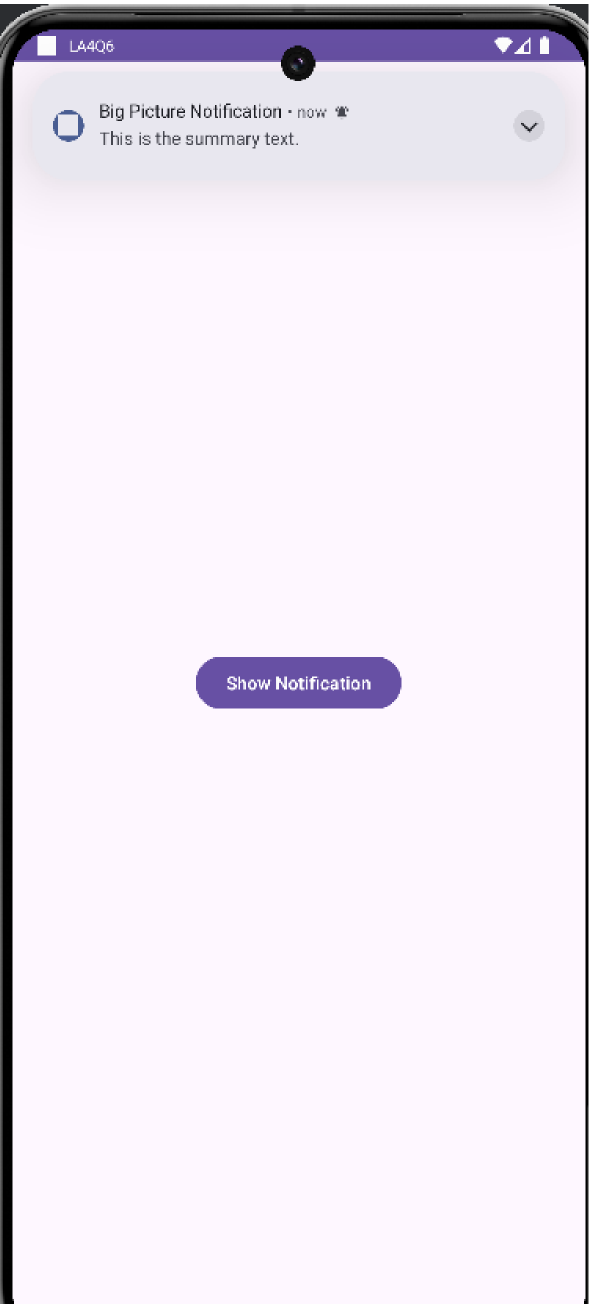
```
    <Button
        android:id="@+id/showNotificationButton"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Show Notification"
        android:layout_centerInParent="true" />
</RelativeLayout>
```

java code:

```
package com.example.la4q6;

import android.app.NotificationChannel;
import android.app.NotificationManager;
import android.content.Context;
import android.graphics.BitmapFactory;
import android.os.Build;
import android.os.Bundle;
import android.widget.Button;
import androidx.appcompat.app.AppCompatActivity;
import androidx.core.app.NotificationCompat;

public class MainActivity extends AppCompatActivity {

    private static final String CHANNEL_ID = "big_picture_channel";

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

        Button showNotificationButton =
findViewById(R.id.showNotificationButton);
        showNotificationButton.setOnClickListener(v ->
showBigPictureNotification());

        createNotificationChannel();
    }

    private void createNotificationChannel() {
        if (Build.VERSION.SDK_INT >= Build.VERSION_CODES.O) {
            NotificationChannel channel = new NotificationChannel(CHANNEL_ID,
                    "Big Picture Channel",
                    NotificationManager.IMPORTANCE_HIGH);
            channel.setDescription("Channel for big picture notifications");
            NotificationManager notificationManager =
getSystemService(NotificationManager.class);
            if (notificationManager != null) {
                notificationManager.createNotificationChannel(channel);
            }
        }
    }

    private void showBigPictureNotification() {

        int imageResource = R.drawable.gfglogo;

        NotificationCompat.Builder builder = new
NotificationCompat.Builder(this, CHANNEL_ID)
                .setSmallIcon(R.drawable.notification)
                .setContentTitle("Big Picture Notification")
                .setContentText("This is the summary text.")
                .setPriority(NotificationCompat.PRIORITY_HIGH)
                .setStyle(new NotificationCompat.BigPictureStyle()
                        .bigPicture(BitmapFactory.decodeResource(getResources(),
imageResource))
                        .setBigContentTitle("Expanded Title")
                        .setSummaryText("This is the summary text when
expanded."))
                .setAutoCancel(true);

        NotificationManager notificationManager = (NotificationManager)
getSystemService(Context.NOTIFICATION_SERVICE);
        notificationManager.notify(1, builder.build());
    }
}
```

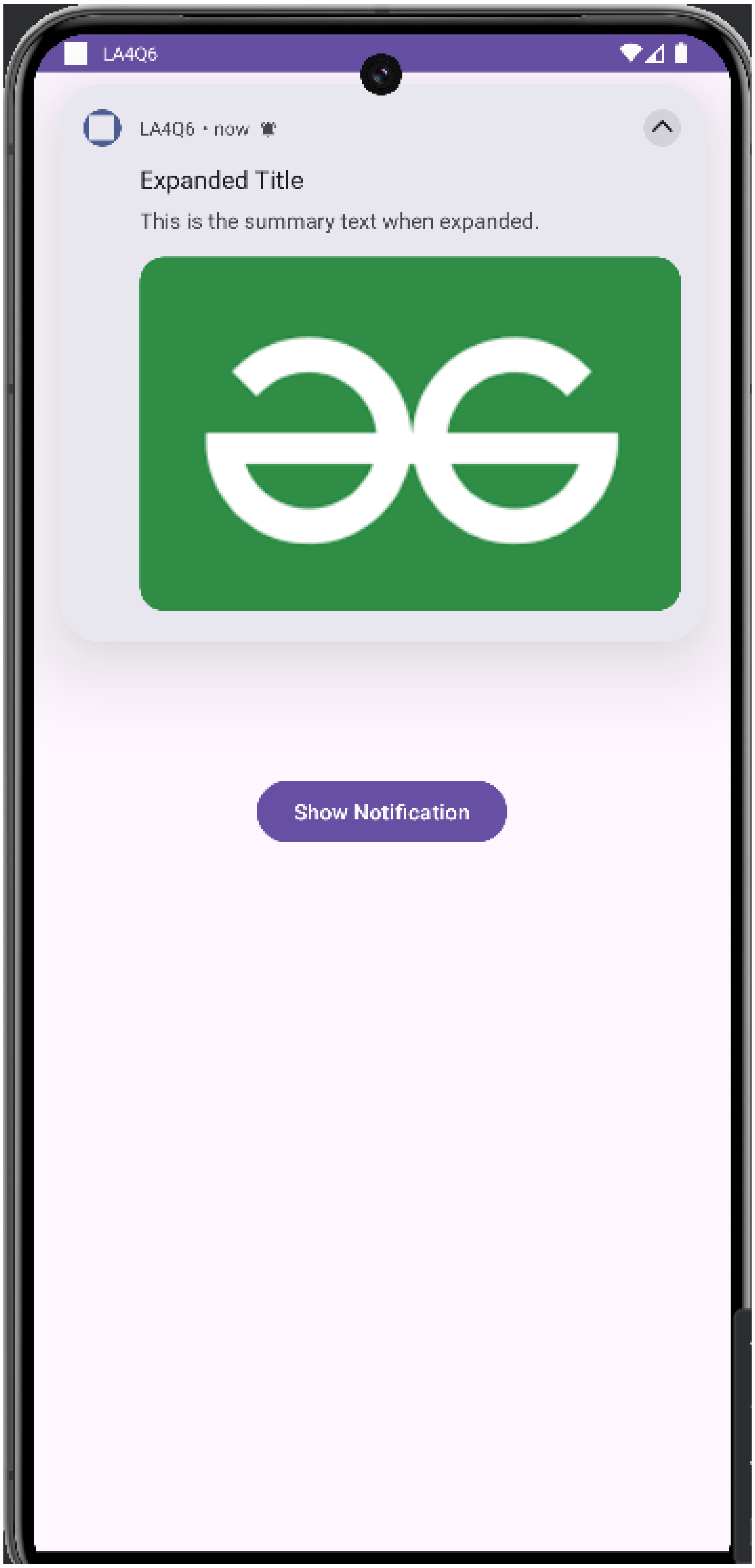
Output:





Big Picture Notification • now

This is the summary text.



Expanded Title

This is the summary text when expanded.

7. Build an app that generates a heads-up notification (high-priority notification that pops up as an overlay). Set up the notification to appear when an urgent event occurs, such as receiving an important message or a time-sensitive alert. Customize the notification to include an action, such as "Dismiss" or "Snooze".

Solution:

xml code:

```xml
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
   android:layout_width="match_parent"
   android:layout_height="match_parent">

   <Button
       android:id="@+id/showNotificationButton"
       android:layout_width="wrap_content"
       android:layout_height="wrap_content"
       android:text="Show Heads-Up Notification"
       android:layout_centerInParent="true" />
</RelativeLayout>
```

java code:

MainActivity:

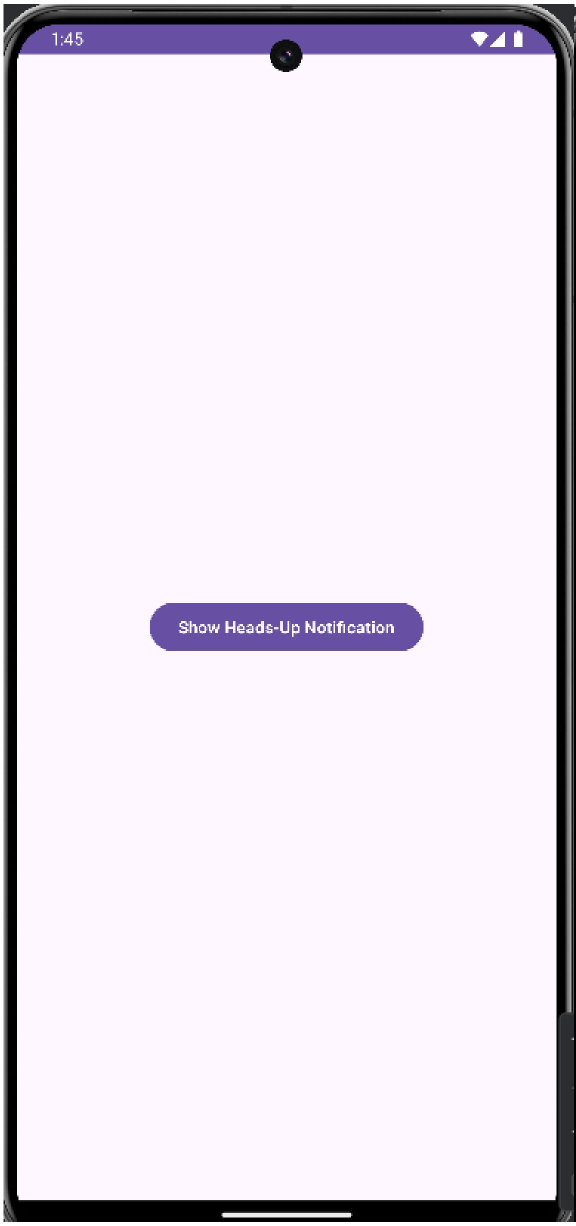
```java
package com.example.la4q7;

import android.app.NotificationChannel;
import android.app.NotificationManager;
import android.app.PendingIntent;
import android.content.Context;
import android.content.Intent;
import android.os.Build;
import android.os.Bundle;
import android.widget.Button;
import androidx.appcompat.app.AppCompatActivity;
import androidx.core.app.NotificationCompat;

public class MainActivity extends AppCompatActivity {

   private static final String CHANNEL_ID = "heads_up_channel";

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

       Button showNotificationButton =
findViewById(R.id.showNotificationButton);
       showNotificationButton.setOnClickListener(v ->
showHeadsUpNotification());

       createNotificationChannel();
   }

   private void createNotificationChannel() {
       if (Build.VERSION.SDK_INT >= Build.VERSION_CODES.O) {
           NotificationChannel channel = new NotificationChannel(CHANNEL_ID,
                   "Heads-Up Channel",
                   NotificationManager.IMPORTANCE_HIGH);
           channel.setDescription("Channel for heads-up notifications");
```

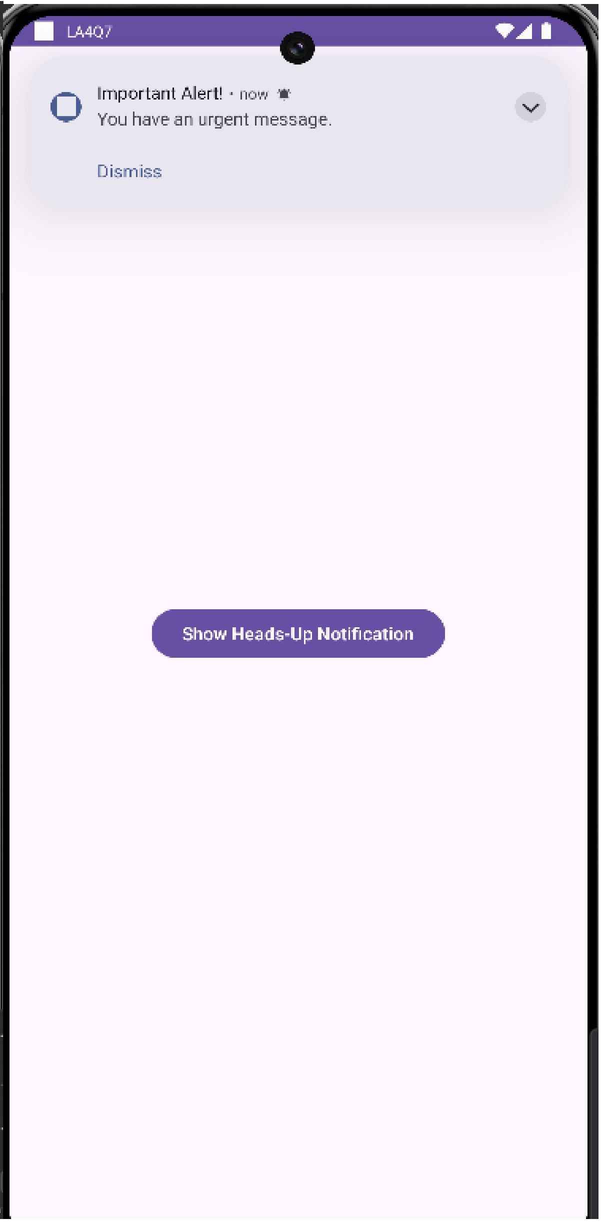
```
            NotificationManager notificationManager =
getSystemService(NotificationManager.class);
            if (notificationManager != null) {
                notificationManager.createNotificationChannel(channel);
            }
        }
    }

    private void showHeadsUpNotification() {
        Intent dismissIntent = new Intent(this, NotificationReceiver.class);
        dismissIntent.setAction(NotificationReceiver.ACTION_DISMISS);

        PendingIntent dismissPendingIntent = PendingIntent.getBroadcast(
                this,
                0,
                dismissIntent,
                PendingIntent.FLAG_UPDATE_CURRENT | PendingIntent.FLAG_IMMUTABLE
        );

        NotificationCompat.Builder builder = new
NotificationCompat.Builder(this, CHANNEL_ID)
                .setSmallIcon(R.drawable.noti)
                .setContentTitle("Important Alert!")
                .setContentText("You have an urgent message.")
                .setPriority(NotificationCompat.PRIORITY_HIGH)
                .setAutoCancel(true)
                .addAction(R.drawable.dismiss, "Dismiss", dismissPendingIntent);

        NotificationManager notificationManager = (NotificationManager)
getSystemService(Context.NOTIFICATION_SERVICE);
        notificationManager.notify(1, builder.build());
    }

}
```

NotificationReceiver:

```
package com.example.la4q7;

import android.content.BroadcastReceiver;
import android.content.Context;
import android.content.Intent;
import android.widget.Toast;

public class NotificationReceiver extends BroadcastReceiver {

    public static final String ACTION_DISMISS =
"com.example.headsupnotification.ACTION_DISMISS";

    @Override
    public void onReceive(Context context, Intent intent) {
        if (intent != null) {
            String action = intent.getAction();
            if (ACTION_DISMISS.equals(action)) {
                Toast.makeText(context, "Notification Dismissed",
Toast.LENGTH_SHORT).show();
            }
        }
    }
}
```

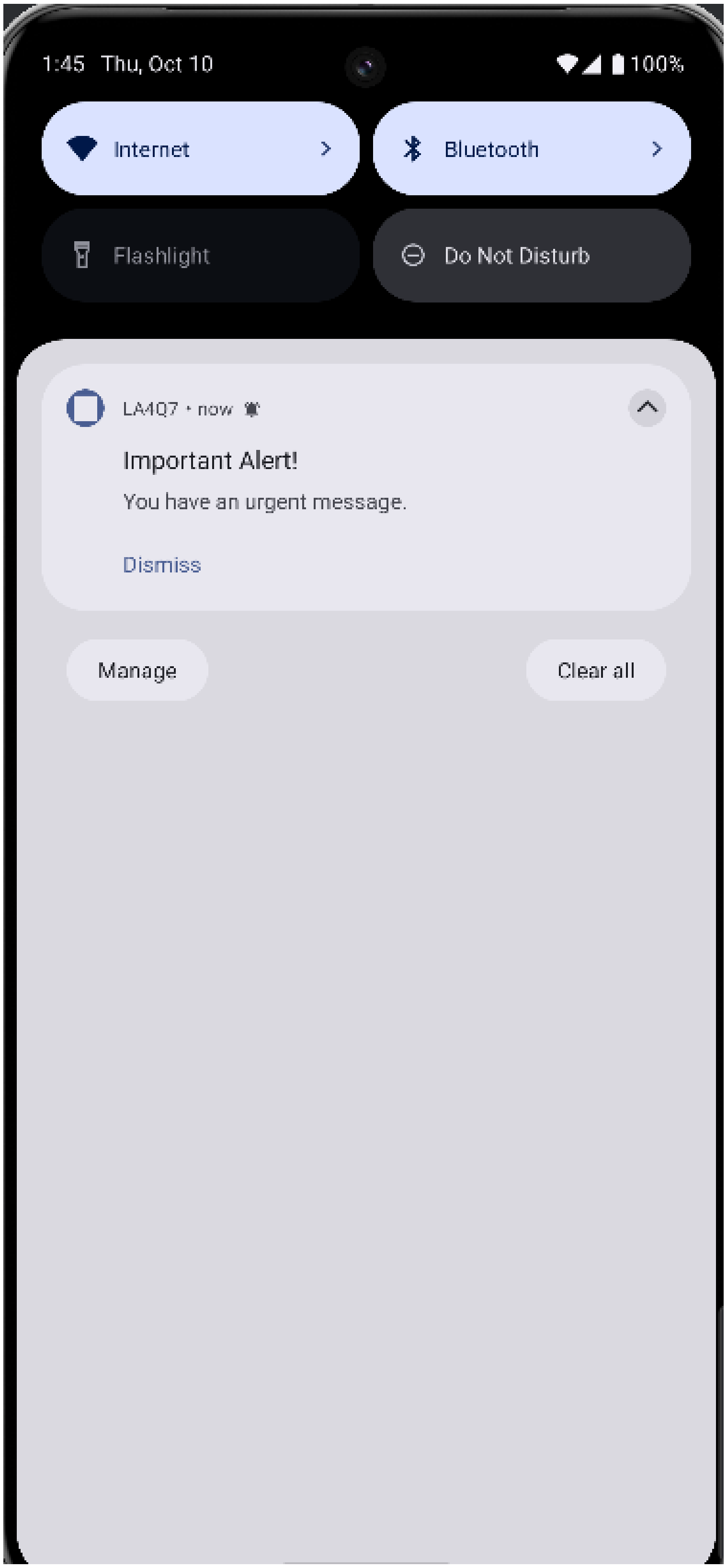
Output:





Important Alert! • now

You have an urgent message.



LA4Q7 • now

Important Alert!

You have an urgent message.

Dismiss

8. Develop an Android application that creates notification channels for different categories of notifications (e.g., "Messages", "Alerts", "Promotions"). Use the NotificationChannel class to define channel properties like importance, sound, and vibration. Ensure notifications are issued under the appropriate channel, and allow the user to customize channel settings.

Solution:

xml code:

activity_main:

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
   android:layout_width="match_parent"
   android:layout_height="match_parent"
   android:padding="16dp">

   <Button
       android:id="@+id/button_message"
       android:layout_width="wrap_content"
       android:layout_height="wrap_content"
       android:text="Send Message Notification" />

   <Button
       android:id="@+id/button_alert"
       android:layout_width="wrap_content"
       android:layout_height="wrap_content"
       android:layout_below="@id/button_message"
       android:layout_marginTop="39dp"
       android:text="Send Alert Notification" />

   <Button
       android:id="@+id/button_promotion"
       android:layout_width="wrap_content"
       android:layout_height="wrap_content"
       android:layout_below="@id/button_alert"
       android:layout_marginTop="34dp"
       android:text="Send Promotion Notification" />

   <Button
       android:id="@+id/button_settings"
       android:layout_width="wrap_content"
       android:layout_height="wrap_content"
       android:layout_below="@id/button_promotion"
       android:layout_marginTop="175dp"
       android:text="Settings" />

</RelativeLayout>
```

activity_settings:

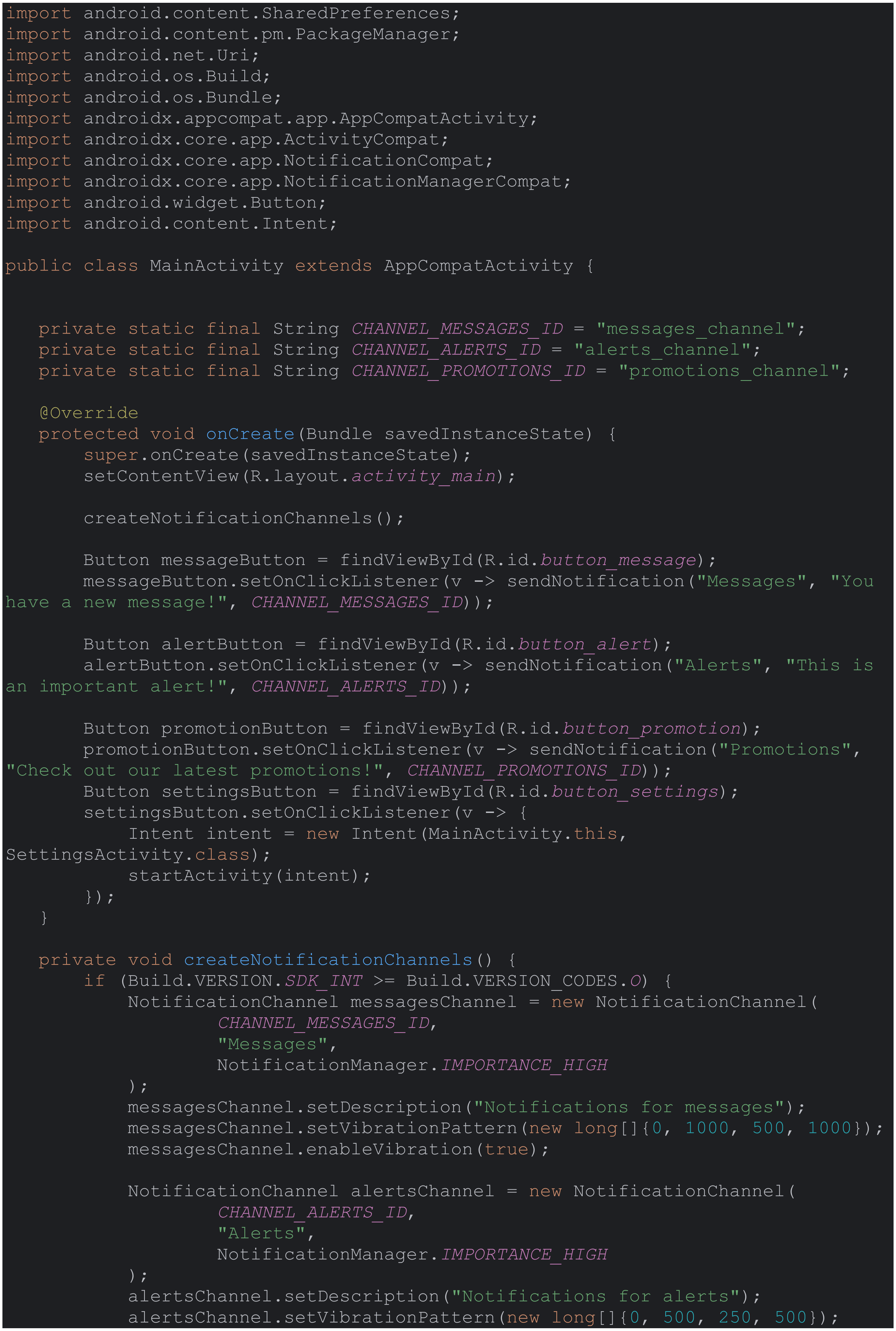
```
<androidx.coordinatorlayout.widget.CoordinatorLayout
xmlns:android="http://schemas.android.com/apk/res/android"
   xmlns:app="http://schemas.android.com/apk/res-auto"
   xmlns:tools="http://schemas.android.com/tools"
   android:layout_width="match_parent"
   android:layout_height="match_parent"
   tools:context=".SettingsActivity">

   <androidx.appcompat.widget.Toolbar
```

```
        android:id="@+id/toolbar"
        android:layout_width="match_parent"
        android:layout_height="?attr/actionBarSize"
        android:background="?attr/colorPrimary"
        app:popupTheme="@style/ThemeOverlay.AppCompat.Light" />

    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:orientation="vertical"
        android:padding="16dp"
        android:layout_marginTop="?attr/actionBarSize">
        <TextView
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="Notification Settings"
            android:textSize="24sp"
            android:layout_marginBottom="16dp" />

        <Switch
            android:id="@+id/notificationsSwitch"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="Enable Notifications" />

        <TextView
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="Notification Importance" />

        <Spinner
            android:id="@+id/importanceSpinner"
            android:layout_width="match_parent"
            android:layout_height="wrap_content" />

        <TextView
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="Notification Sound" />

        <Button
            android:id="@+id/selectSoundButton"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="Select Sound" />

        <Button
            android:id="@+id/saveButton"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_marginTop="24dp"
            android:text="Save Settings" />
    </LinearLayout>

</androidx.coordinatorlayout.widget.CoordinatorLayout>
```

java code:

MainActivity:

```
package com.example.la4q8;

import android.app.NotificationChannel;
import android.app.NotificationManager;
```

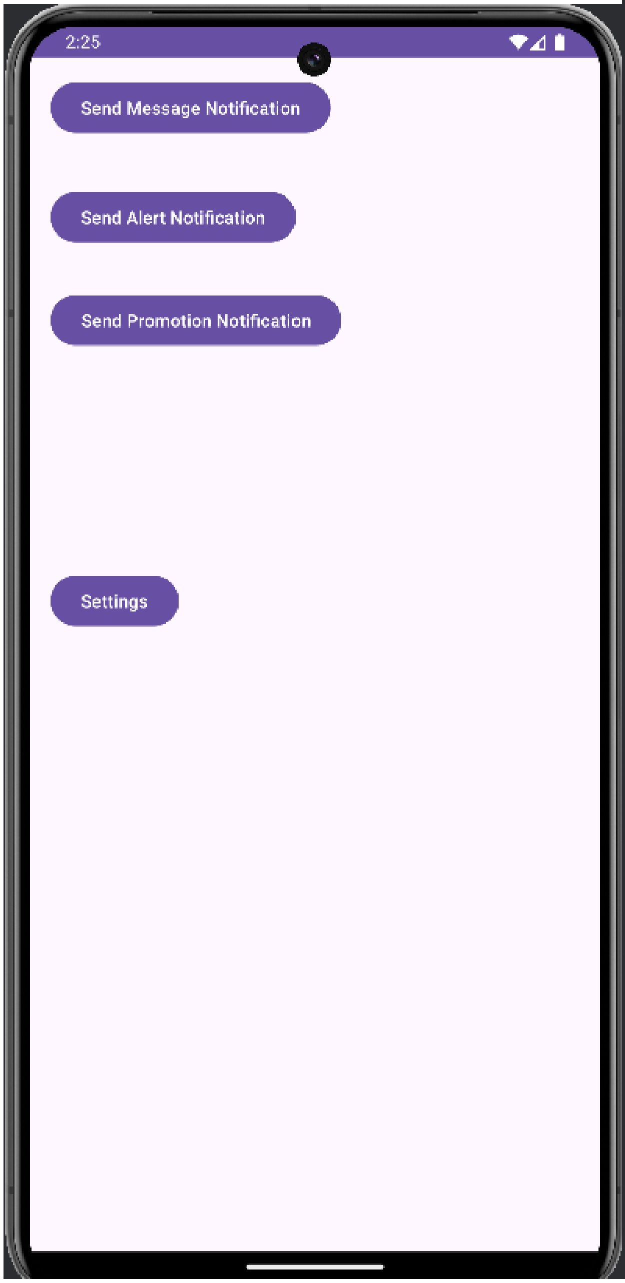


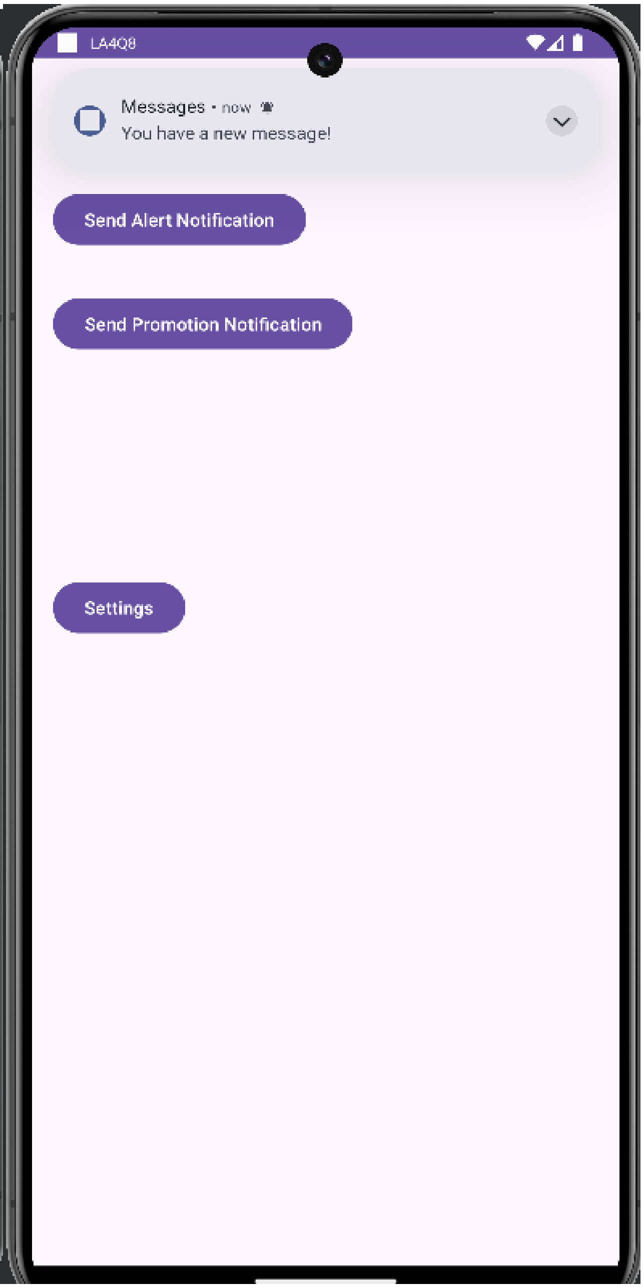
```
import android.content.SharedPreferences;
import android.content.pm.PackageManager;
import android.net.Uri;
import android.os.Build;
import android.os.Bundle;
import androidx.appcompat.app.AppCompatActivity;
import androidx.core.app.ActivityCompat;
import androidx.core.app.NotificationCompat;
import androidx.core.app.NotificationManagerCompat;
import android.widget.Button;
import android.content.Intent;

public class MainActivity extends AppCompatActivity {


    private static final String CHANNEL_MESSAGES_ID = "messages_channel";
    private static final String CHANNEL_ALERTS_ID = "alerts_channel";
    private static final String CHANNEL_PROMOTIONS_ID = "promotions_channel";

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

        createNotificationChannels();

        Button messageButton = findViewById(R.id.button_message);
        messageButton.setOnClickListener(v -> sendNotification("Messages", "You
have a new message!", CHANNEL_MESSAGES_ID));

        Button alertButton = findViewById(R.id.button_alert);
        alertButton.setOnClickListener(v -> sendNotification("Alerts", "This is
an important alert!", CHANNEL_ALERTS_ID));

        Button promotionButton = findViewById(R.id.button_promotion);
        promotionButton.setOnClickListener(v -> sendNotification("Promotions",
"Check out our latest promotions!", CHANNEL_PROMOTIONS_ID));
        Button settingsButton = findViewById(R.id.button_settings);
        settingsButton.setOnClickListener(v -> {
            Intent intent = new Intent(MainActivity.this,
SettingsActivity.class);
            startActivity(intent);
        });
    }

    private void createNotificationChannels() {
        if (Build.VERSION.SDK_INT >= Build.VERSION_CODES.O) {
            NotificationChannel messagesChannel = new NotificationChannel(
                    CHANNEL_MESSAGES_ID,
                    "Messages",
                    NotificationManager.IMPORTANCE_HIGH
            );
            messagesChannel.setDescription("Notifications for messages");
            messagesChannel.setVibrationPattern(new long[]{0, 1000, 500, 1000});
            messagesChannel.enableVibration(true);

            NotificationChannel alertsChannel = new NotificationChannel(
                    CHANNEL_ALERTS_ID,
                    "Alerts",
                    NotificationManager.IMPORTANCE_HIGH
            );
            alertsChannel.setDescription("Notifications for alerts");
            alertsChannel.setVibrationPattern(new long[]{0, 500, 250, 500});
            alertsChannel.enableVibration(true);

            NotificationChannel promotionsChannel = new NotificationChannel(
                    CHANNEL_PROMOTIONS_ID,
                    "Promotions",
                    NotificationManager.IMPORTANCE_LOW
            );
            promotionsChannel.setDescription("Notifications for promotions");
            promotionsChannel.setVibrationPattern(new long[]{0, 300, 100, 300});
            promotionsChannel.enableVibration(false);

            NotificationManager notificationManager =
getSystemService(NotificationManager.class);
            if (notificationManager != null) {
                notificationManager.createNotificationChannel(messagesChannel);
                notificationManager.createNotificationChannel(alertsChannel);

notificationManager.createNotificationChannel(promotionsChannel);
            }
        }
    }

    private void sendNotification(String title, String message, String
channelId) {
        NotificationCompat.Builder builder = new
NotificationCompat.Builder(this, channelId)
                .setSmallIcon(R.drawable.noti)
                .setContentTitle(title)
                .setContentText(message)
                .setPriority(NotificationCompat.PRIORITY_HIGH)
                .setAutoCancel(true);

        NotificationManagerCompat notificationManager =
NotificationManagerCompat.from(this);
        if (ActivityCompat.checkSelfPermission(this,
android.Manifest.permission.POST_NOTIFICATIONS) !=
PackageManager.PERMISSION_GRANTED) {
            return;
        }
        notificationManager.notify((int) System.currentTimeMillis(),
builder.build());
    }
    private void createNotificationChannel() {
        SharedPreferences preferences = getSharedPreferences("AppPreferences",
MODE_PRIVATE);
        boolean notificationsEnabled =
preferences.getBoolean("notifications_enabled", true);
        String importanceLevel =
preferences.getString("notification_importance", "Default");
        String soundUriString = preferences.getString("notification_sound",
null);
        Uri soundUri = soundUriString != null ? Uri.parse(soundUriString) :
null;

        if (notificationsEnabled) {
            int importance = NotificationManager.IMPORTANCE_DEFAULT;
            switch (importanceLevel) {
                case "High":
                    importance = NotificationManager.IMPORTANCE_HIGH;
                    break;
                case "Low":
                    importance = NotificationManager.IMPORTANCE_LOW;
                    break;
                case "Max":
                    importance = NotificationManager.IMPORTANCE_MAX;
                    break;
                case "Min":
                    importance = NotificationManager.IMPORTANCE_MIN;
                    break;
            }

            NotificationChannel channel = null;
            if (Build.VERSION.SDK_INT >= Build.VERSION_CODES.O) {
                channel = new NotificationChannel("YourChannelId", "Your Channel
Name", importance);
            }
            if (soundUri != null) {
                if (Build.VERSION.SDK_INT >= Build.VERSION_CODES.O) {
                    channel.setSound(soundUri, null);
                }
            }
            NotificationManager notificationManager =
getSystemService(NotificationManager.class);
            if (Build.VERSION.SDK_INT >= Build.VERSION_CODES.O) {
                notificationManager.createNotificationChannel(channel);
            }
        }
    }

}
```

SettingsActivity:

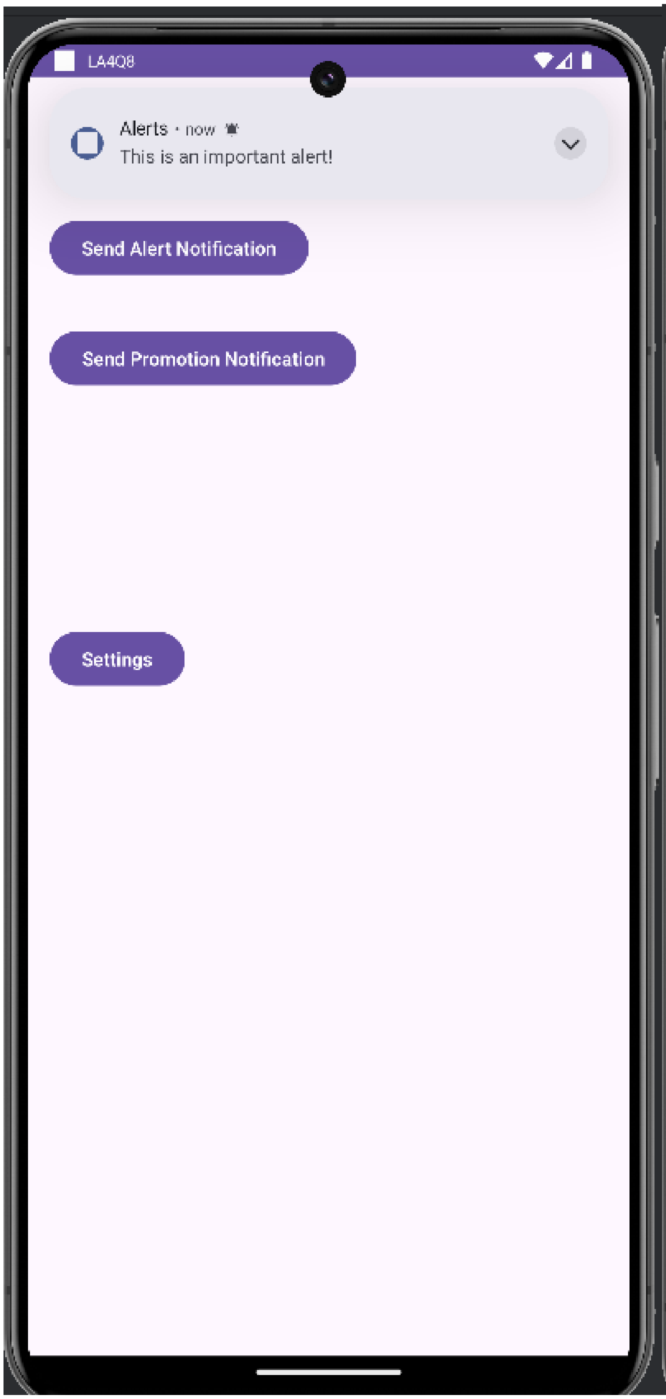
```
package com.example.la4q8;

import android.content.Intent;
import android.content.SharedPreferences;
import android.media.RingtoneManager;
import android.net.Uri;
import android.os.Bundle;
import android.widget.ArrayAdapter;
import android.widget.Button;
import android.widget.Spinner;
import android.widget.Switch;
import androidx.appcompat.app.AppCompatActivity;
import androidx.appcompat.widget.Toolbar;

public class SettingsActivity extends AppCompatActivity {

    private Switch notificationsSwitch;
    private Spinner importanceSpinner;
    private Button selectSoundButton;
    private Button saveButton;
    private Uri notificationSound;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_settings);

        Toolbar toolbar = findViewById(R.id.toolbar);
        setSupportActionBar(toolbar);
        getSupportActionBar().setDisplayHomeAsUpEnabled(true);

        notificationsSwitch = findViewById(R.id.notificationsSwitch);
        importanceSpinner = findViewById(R.id.importanceSpinner);
        selectSoundButton = findViewById(R.id.selectSoundButton);
        saveButton = findViewById(R.id.saveButton);

        setupImportanceSpinner();

        selectSoundButton.setOnClickListener(v -> selectNotificationSound());

        saveButton.setOnClickListener(v -> saveSettings());
    }

    private void setupImportanceSpinner() {
        ArrayAdapter<CharSequence> adapter =
ArrayAdapter.createFromResource(this,
                R.array.importance_levels,
android.R.layout.simple_spinner_item);

adapter.setDropDownViewResource(android.R.layout.simple_spinner_dropdown_item);
        importanceSpinner.setAdapter(adapter);
    }

    private void selectNotificationSound() {
        Intent intent = new Intent(RingtoneManager.ACTION_RINGTONE_PICKER);
        intent.putExtra(RingtoneManager.EXTRA_RINGTONE_TYPE,
RingtoneManager.TYPE_NOTIFICATION);
        intent.putExtra(RingtoneManager.EXTRA_RINGTONE_TITLE, "Select
Notification Sound");
        intent.putExtra(RingtoneManager.EXTRA_RINGTONE_EXISTING_URI,
notificationSound);
        startActivityForResult(intent, 1);
    }

    @Override
    protected void onActivityResult(int requestCode, int resultCode, Intent
data) {
        if (requestCode == 1 && resultCode == RESULT_OK) {
            notificationSound =
data.getParcelableExtra(RingtoneManager.EXTRA_RINGTONE_PICKED_URI);
        }
        super.onActivityResult(requestCode, resultCode, data);
    }

    private void saveSettings() {
        SharedPreferences preferences = getSharedPreferences("AppPreferences",
MODE_PRIVATE);
        SharedPreferences.Editor editor = preferences.edit();

        editor.putBoolean("notifications_enabled",
notificationsSwitch.isChecked());
        editor.putString("notification_importance",
importanceSpinner.getSelectedItem().toString());
        editor.putString("notification_sound", notificationSound != null ?
notificationSound.toString() : null);

        editor.apply();
        finish();
    }
}
```

Output:



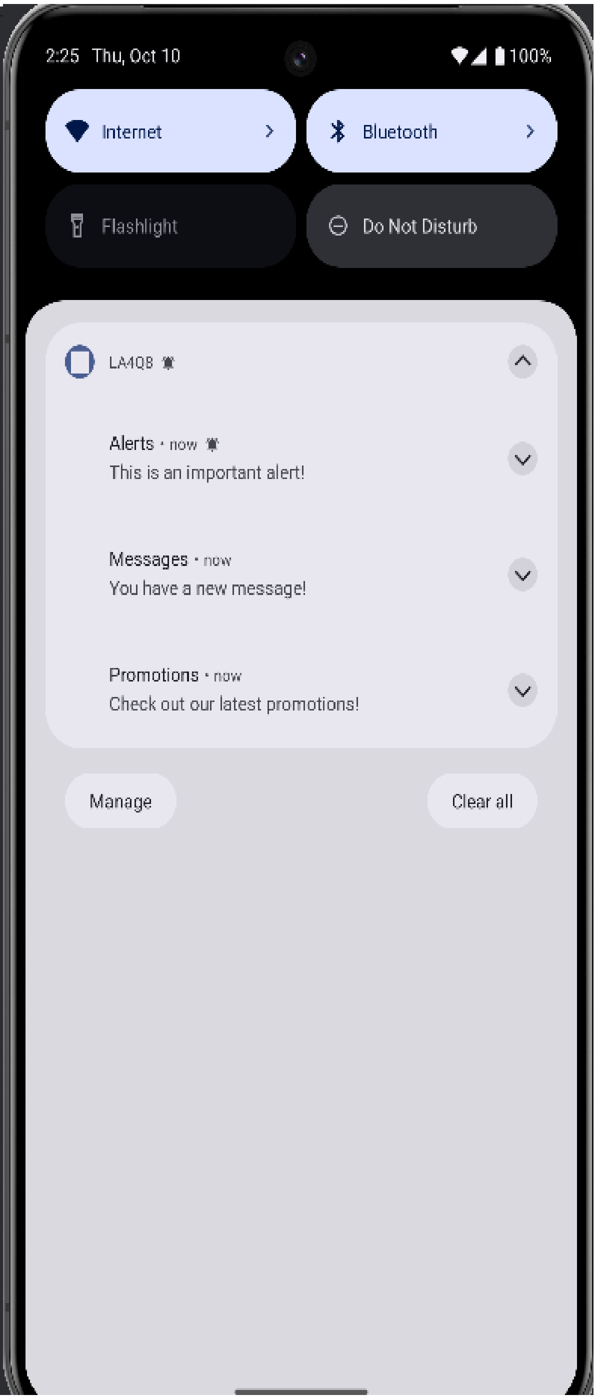


Messages • now
You have a new message!



Alerts · now

This is an important alert!



Alerts · now

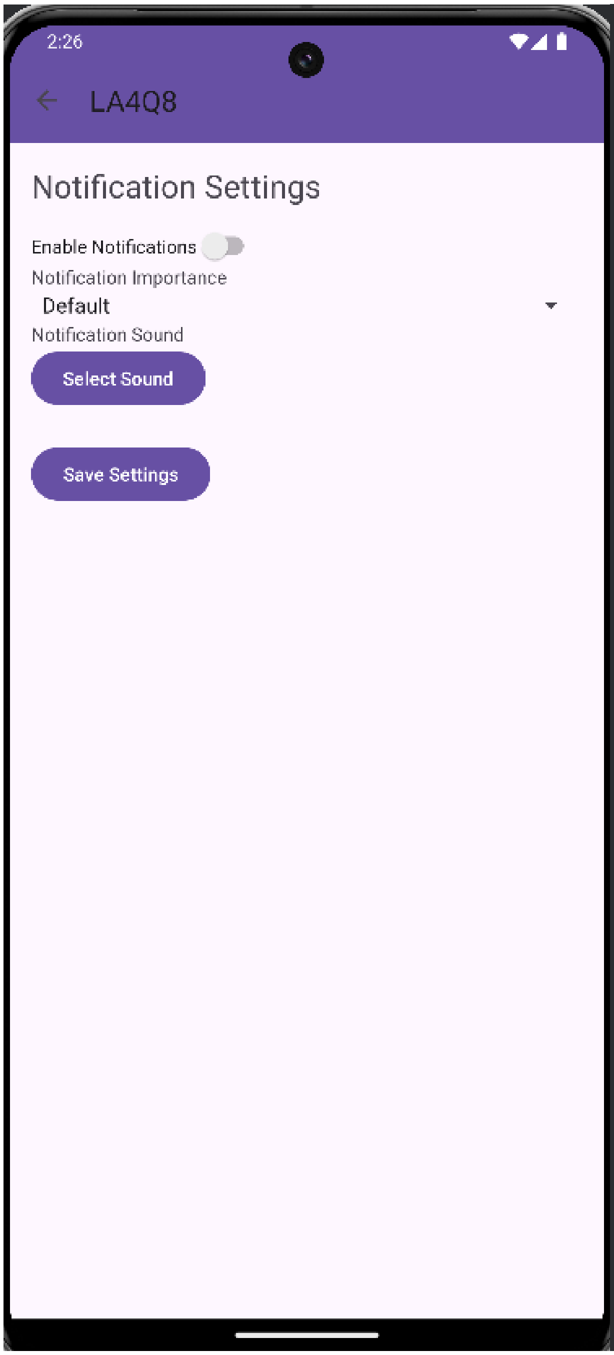
This is an important alert!

Messages · now

You have a new message!

Promotions · now

Check out our latest promotions!



2:26

LA4Q8

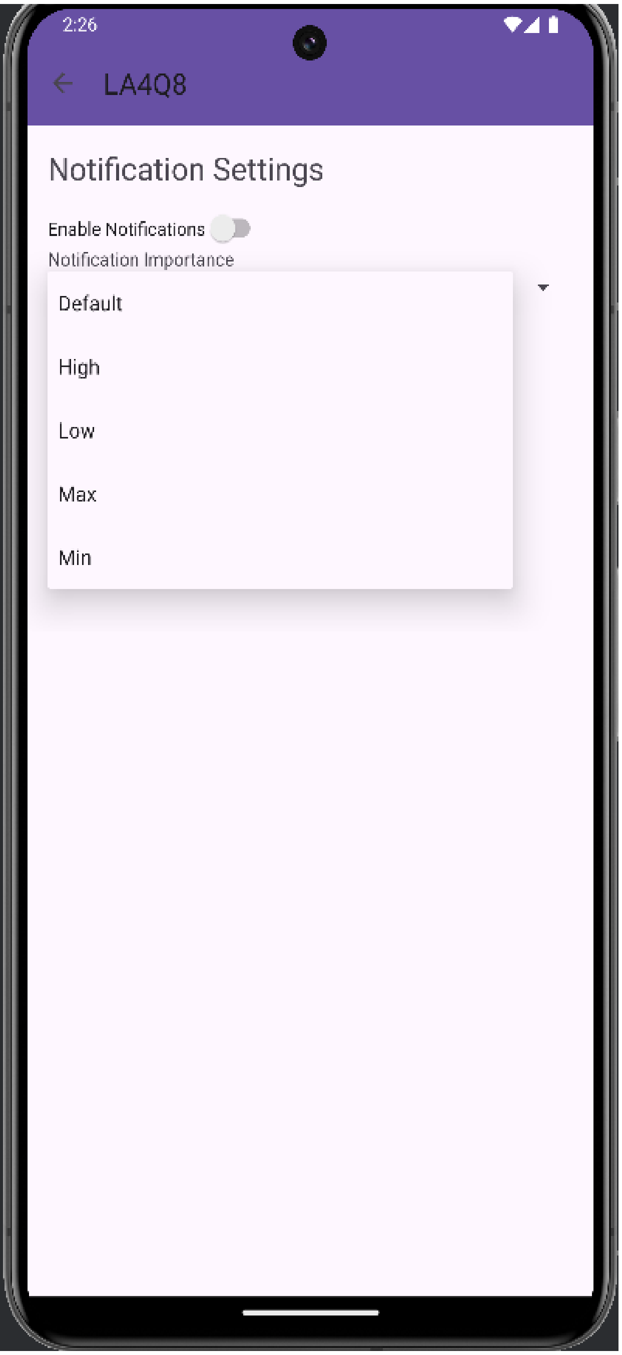
Notification Settings

Enable Notifications

Notification Importance

Default

Notification Sound



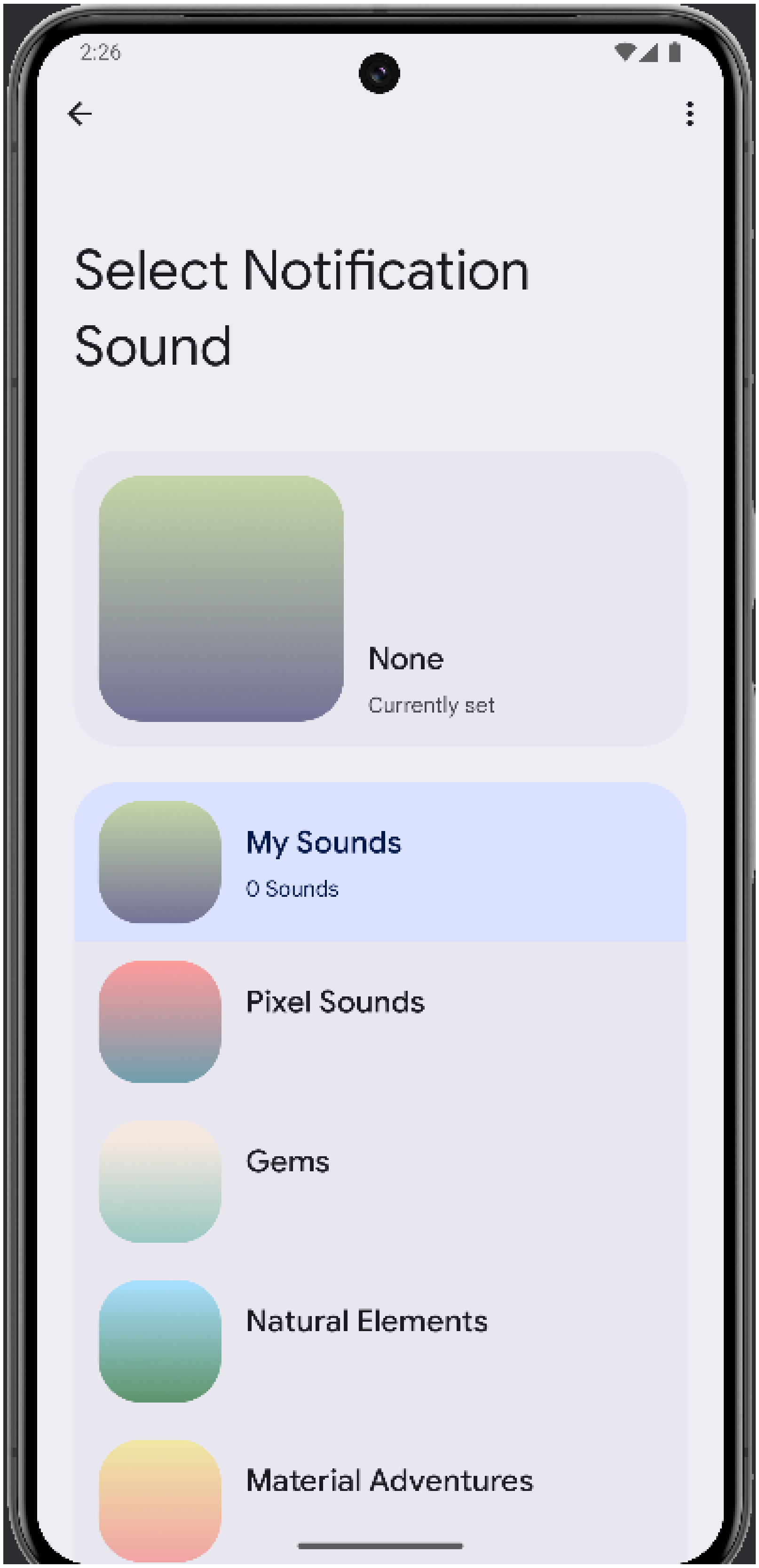
2:26

LA4Q8

Notification Settings

Enable Notifications

Notification Importance



2:26

Select Notification Sound

None

Currently set

My Sounds

0 Sounds

Pixel Sounds

Gems

Natural Elements

Material Adventures

9. Create an application that issues multiple notifications and groups them into a single expandable notification. Use NotificationCompat.Builder and NotificationCompat.InboxStyle to group notifications, such as showing a list of recent messages in a messaging app. Implement functionality to expand and collapse the group.

Solution:

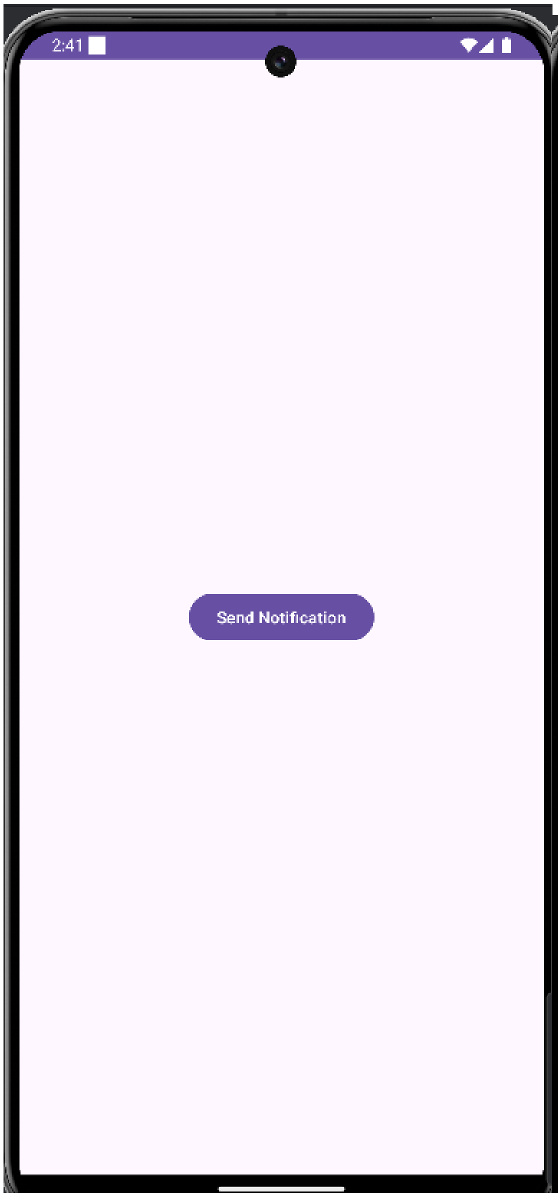
xml code:

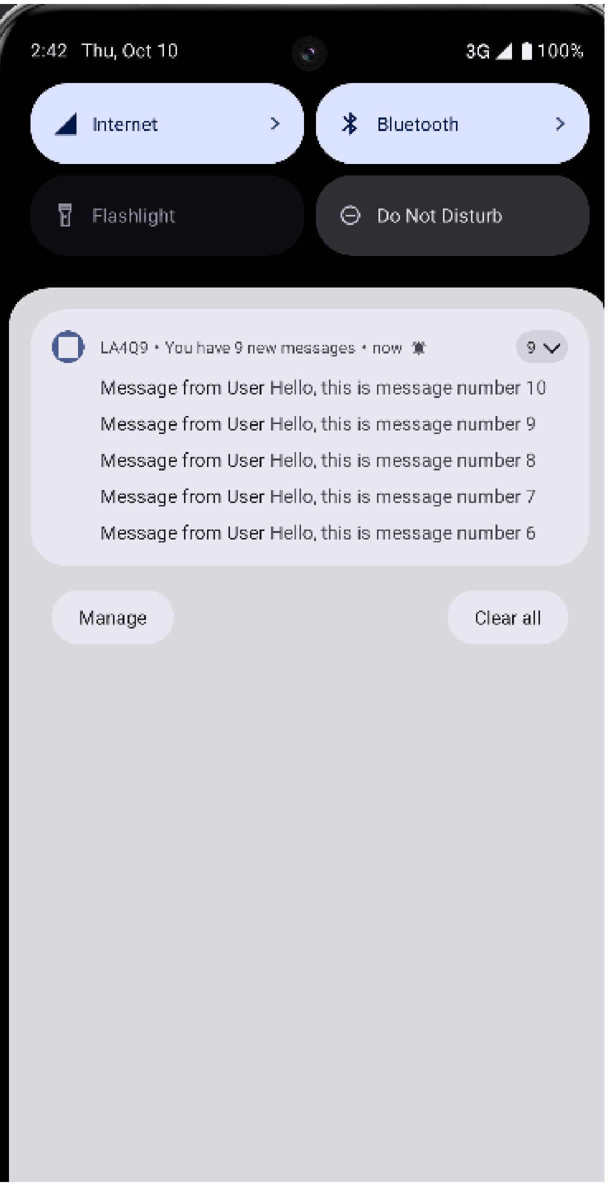
```xml
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
   android:layout_width="match_parent"
   android:layout_height="match_parent">

   <Button
       android:id="@+id/send_button"
       android:layout_width="wrap_content"
       android:layout_height="wrap_content"
       android:text="Send Notification"
       android:layout_centerInParent="true"/>
</RelativeLayout>
```

java code:

```java
package com.example.la4q9;

import android.app.NotificationChannel;
import android.app.NotificationManager;
import android.app.PendingIntent;
import android.content.Intent;
import android.os.Build;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import androidx.appcompat.app.AppCompatActivity;
import androidx.core.app.NotificationCompat;

public class MainActivity extends AppCompatActivity {

   private static final String CHANNEL_ID = "message_channel";
   private int notificationCount = 0;

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

       Button sendButton = findViewById(R.id.send_button);
       createNotificationChannel();

       sendButton.setOnClickListener(new View.OnClickListener() {
           @Override
           public void onClick(View v) {
               sendGroupedNotification("Message from User", "Hello, this is message number " + (notificationCount + 1));
               notificationCount++;
           }
       });
   }

   private void sendGroupedNotification(String title, String message) {
       NotificationManager notificationManager = (NotificationManager)
getSystemService(NOTIFICATION_SERVICE);

       Intent intent = new Intent(this, MainActivity.class);
       intent.setFlags(Intent.FLAG_ACTIVITY_NEW_TASK |
Intent.FLAG_ACTIVITY_CLEAR_TASK);

       PendingIntent pendingIntent = PendingIntent.getActivity(this, 0, intent,
PendingIntent.FLAG_UPDATE_CURRENT | PendingIntent.FLAG_IMMUTABLE);

       NotificationCompat.Builder builder = new
NotificationCompat.Builder(this, CHANNEL_ID)
               .setSmallIcon(R.drawable.noti)
               .setContentTitle(title)
               .setContentText(message)
               .setContentIntent(pendingIntent)
               .setAutoCancel(true)
               .setGroup("messages_group")
               .setGroupSummary(false);


       notificationManager.notify(notificationCount, builder.build());


       NotificationCompat.Builder summaryBuilder = new
NotificationCompat.Builder(this, CHANNEL_ID)
               .setContentTitle("New Messages")
               .setContentText("You have " + notificationCount + " new
messages")
               .setSmallIcon(R.drawable.noti)
               .setStyle(new NotificationCompat.InboxStyle()
                       .addLine("Message from User: Hello, this is message
number " + notificationCount)
                       .setBigContentTitle("New Messages")
                       .setSummaryText("You have " + notificationCount + " new
messages"))
               .setGroup("messages_group")
               .setGroupSummary(true)
               .setContentIntent(pendingIntent);

       notificationManager.notify(0, summaryBuilder.build());
   }

   private void createNotificationChannel() {
       if (Build.VERSION.SDK_INT >= Build.VERSION_CODES.O) {
           NotificationChannel channel = new NotificationChannel(
                   CHANNEL_ID,
                   "Message Notifications",
                   NotificationManager.IMPORTANCE_DEFAULT);
           NotificationManager manager =
getSystemService(NotificationManager.class);
           manager.createNotificationChannel(channel);
       }
   }
}
```

Output:





LA4Q9 • You have 9 new messages • now

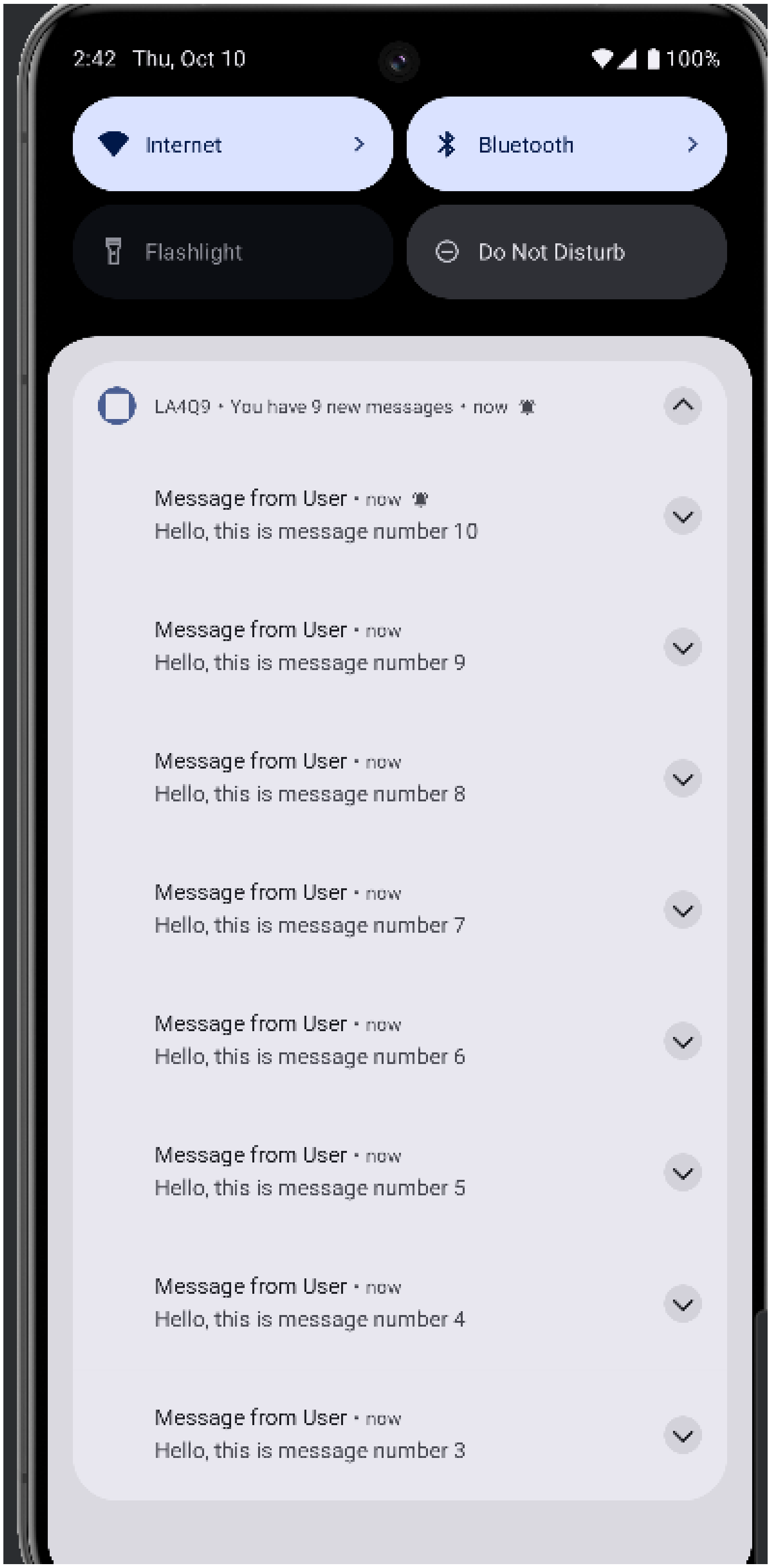
Message from User Hello, this is message number 10

Message from User Hello, this is message number 9

Message from User Hello, this is message number 8

Message from User Hello, this is message number 7

Message from User Hello, this is message number 6



LA4Q9 • You have 9 new messages • now

Message from User • now
Hello, this is message number 10

Message from User • now
Hello, this is message number 9

Message from User • now
Hello, this is message number 8

Message from User • now
Hello, this is message number 7

Message from User • now
Hello, this is message number 6

Message from User • now
Hello, this is message number 5

Message from User • now
Hello, this is message number 4

Message from User • now
Hello, this is message number 3

10. Design an application that schedules and triggers notifications at a specific time or interval (e.g., daily reminders). Use AlarmManager or WorkManager to schedule the notifications, and issue them using NotificationCompat.Builder. Ensure that notifications are triggered even when the app is in the background or closed.

Solution:

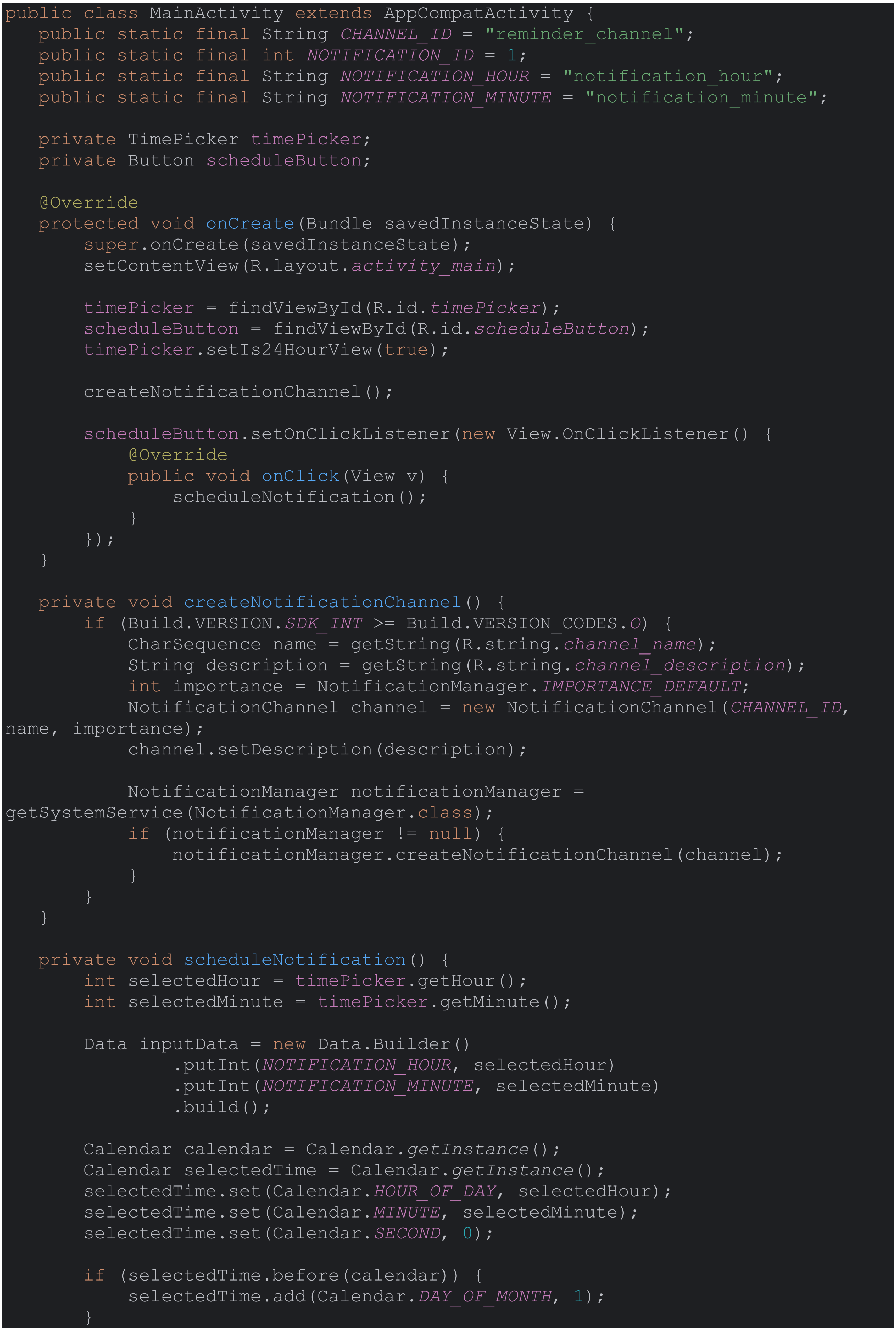
xml code:

```
<?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"
   android:padding="16dp">

   <TextView
       android:layout_width="wrap_content"
       android:layout_height="wrap_content"
       android:text="@string/select_time"
       android:textSize="18sp"
       android:layout_marginBottom="16dp"/>

   <TimePicker
       android:id="@+id/timePicker"
       android:layout_width="match_parent"
       android:layout_height="wrap_content"
       android:timePickerMode="spinner"/>

   <Button
       android:id="@+id/scheduleButton"
       android:layout_width="match_parent"
       android:layout_height="wrap_content"
       android:text="@string/schedule_notification"
       android:layout_marginTop="16dp"/>

</LinearLayout>
```

java code:

MainActivity:

```
package com.example.la4q10;

import android.app.NotificationChannel;
import android.app.NotificationManager;
import android.os.Build;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.TimePicker;
import android.widget.Toast;

import androidx.appcompat.app.AppCompatActivity;
import androidx.work.Data;
import androidx.work.ExistingPeriodicWorkPolicy;
import androidx.work.PeriodicWorkRequest;
import androidx.work.WorkManager;

import java.util.Calendar;
import java.util.concurrent.TimeUnit;
```



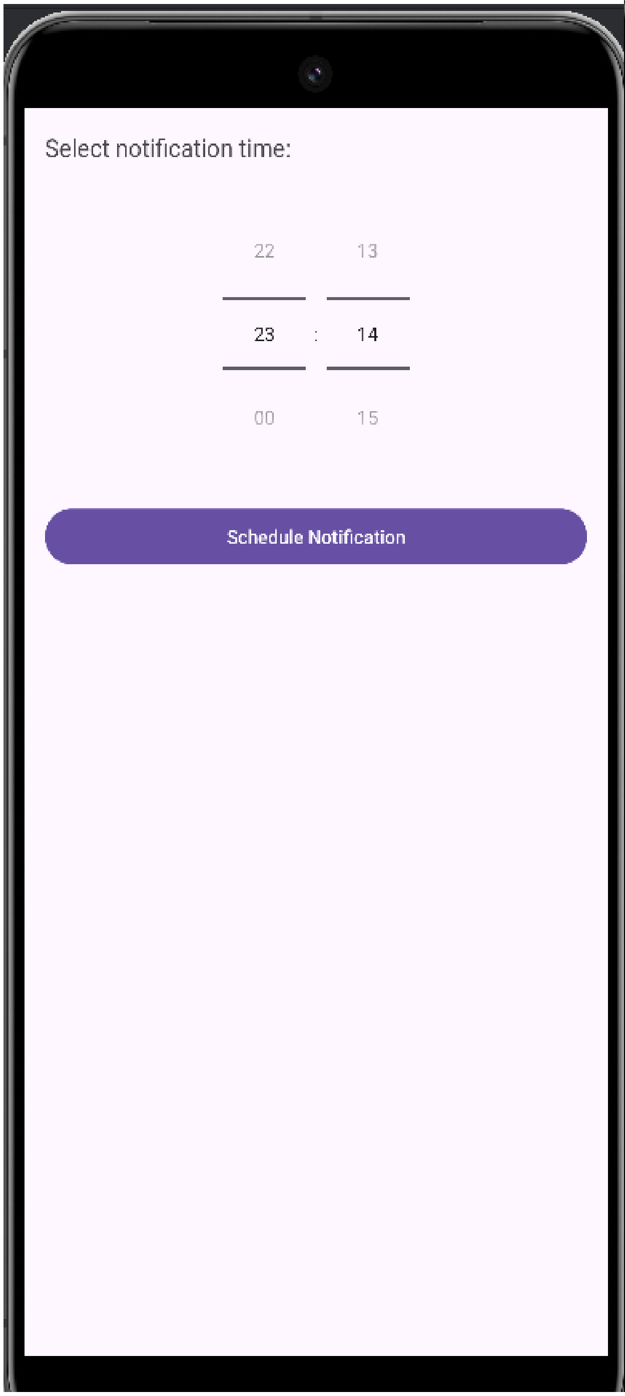
```
public class MainActivity extends AppCompatActivity {
   public static final String CHANNEL_ID = "reminder_channel";
   public static final int NOTIFICATION_ID = 1;
   public static final String NOTIFICATION_HOUR = "notification_hour";
   public static final String NOTIFICATION_MINUTE = "notification_minute";

   private TimePicker timePicker;
   private Button scheduleButton;

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

       timePicker = findViewById(R.id.timePicker);
       scheduleButton = findViewById(R.id.scheduleButton);
       timePicker.setIs24HourView(true);

       createNotificationChannel();

       scheduleButton.setOnClickListener(new View.OnClickListener() {
           @Override
           public void onClick(View v) {
               scheduleNotification();
           }
       });
   }

   private void createNotificationChannel() {
       if (Build.VERSION.SDK_INT >= Build.VERSION_CODES.O) {
           CharSequence name = getString(R.string.channel_name);
           String description = getString(R.string.channel_description);
           int importance = NotificationManager.IMPORTANCE_DEFAULT;
           NotificationChannel channel = new NotificationChannel(CHANNEL_ID,
name, importance);
           channel.setDescription(description);

           NotificationManager notificationManager =
getSystemService(NotificationManager.class);
           if (notificationManager != null) {
               notificationManager.createNotificationChannel(channel);
           }
       }
   }

   private void scheduleNotification() {
       int selectedHour = timePicker.getHour();
       int selectedMinute = timePicker.getMinute();

       Data inputData = new Data.Builder()
               .putInt(NOTIFICATION_HOUR, selectedHour)
               .putInt(NOTIFICATION_MINUTE, selectedMinute)
               .build();

       Calendar calendar = Calendar.getInstance();
       Calendar selectedTime = Calendar.getInstance();
       selectedTime.set(Calendar.HOUR_OF_DAY, selectedHour);
       selectedTime.set(Calendar.MINUTE, selectedMinute);
       selectedTime.set(Calendar.SECOND, 0);

       if (selectedTime.before(calendar)) {
           selectedTime.add(Calendar.DAY_OF_MONTH, 1);
       }
```

```
        long initialDelay = selectedTime.getTimeInMillis() -
calendar.getTimeInMillis();

        PeriodicWorkRequest notificationWorkRequest =
                new PeriodicWorkRequest.Builder(NotificationWorker.class, 24,
TimeUnit.HOURS)
                        .setInputData(inputData)
                        .setInitialDelay(initialDelay, TimeUnit.MILLISECONDS)
                        .build();


WorkManager.getInstance(getApplicationContext()).enqueueUniquePeriodicWork(
                "daily_reminder",
                ExistingPeriodicWorkPolicy.REPLACE,
                notificationWorkRequest
        );

        String message = String.format("Notification scheduled for %02d:%02d
daily",
                selectedHour, selectedMinute);
        Toast.makeText(this, message, Toast.LENGTH_SHORT).show();
    }
}
```

NotificationWorker:

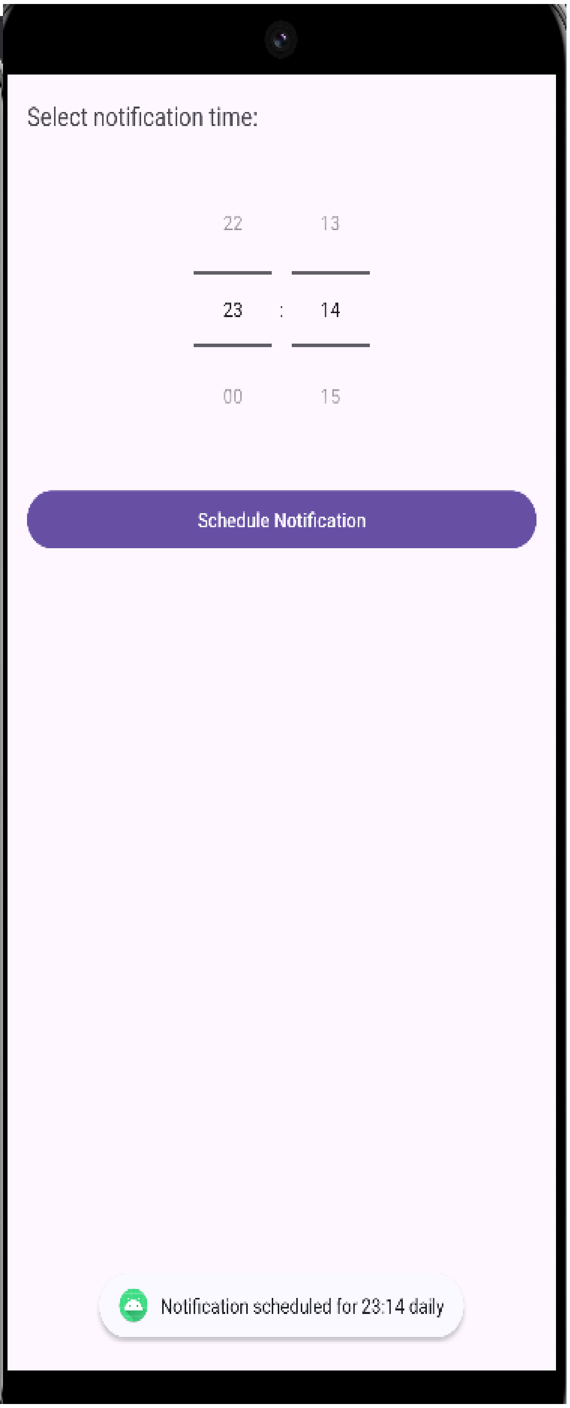
```
package com.example.la4q10;

import android.app.NotificationManager;
import android.content.Context;

import androidx.annotation.NonNull;
import androidx.core.app.NotificationCompat;
import androidx.work.Worker;
import androidx.work.WorkerParameters;
import androidx.work.Data;


public class NotificationWorker extends Worker {
    public NotificationWorker(@NonNull Context context, @NonNull
WorkerParameters params) {
        super(context, params);
    }

    @NonNull
    @Override
    public Result doWork() {
        Data inputData = getInputData();
        int hour = inputData.getInt(MainActivity.NOTIFICATION_HOUR, -1);
        int minute = inputData.getInt(MainActivity.NOTIFICATION_MINUTE, -1);

        if (hour != -1 && minute != -1) {
            showNotification(hour, minute);
        }

        return Result.success();
    }

    private void showNotification(int hour, int minute) {
        Context context = getApplicationContext();
        if (context == null) return;
        String notificationText = String.format("It's %02d:%02d - Time for your
daily reminder!",hour, minute);

        NotificationCompat.Builder builder = new
NotificationCompat.Builder(context, MainActivity.CHANNEL_ID)
                .setSmallIcon(R.drawable.noti)
                .setContentTitle("Daily Reminder")
                .setContentText(notificationText)
                .setPriority(NotificationCompat.PRIORITY_DEFAULT)
                .setAutoCancel(true);

        NotificationManager notificationManager =
                (NotificationManager)
context.getSystemService(Context.NOTIFICATION_SERVICE);
        if (notificationManager != null) {
            notificationManager.notify(MainActivity.NOTIFICATION_ID,
builder.build());
        }
    }
}
```

Output:



Select notification time:

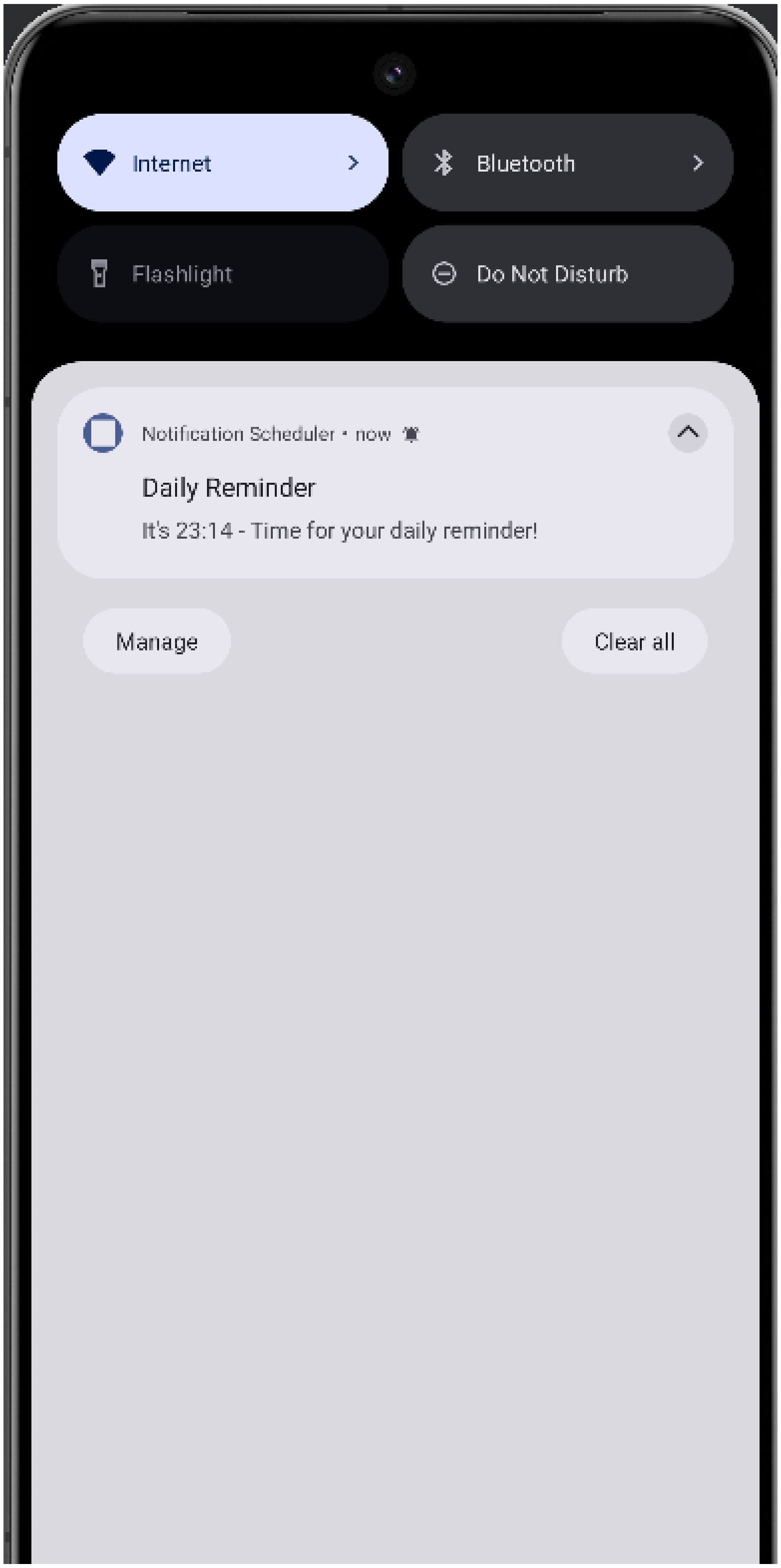
```
22   13
23 : 14
00   15
```



Select notification time:

```
22   13
23 : 14
00   15
```

Notification scheduled for 23:14 daily



Notification Scheduler • now

Daily Reminder

It's 23:14 - Time for your daily reminder!