WIA2007 Mobile Application Development
Semester 1, Session 2022/2023
Practical 6 (View and ViewGroup)

Pre-Task: Plan for Activity

To complete today's practical, you will need the following items from the past tutorial and practical:

- ☑ Practical 3: Open your "My First Application" project used in Practical 3.
- ☑ Tutorial 4: Recall Question 4 in Tutorial 4 which requires you to prepare a UI draft to obtain feedback from the users.

You are implementing your UI draft from Tutorial 4 using Android Studio by utilizing the Project from Practical 3.

Task 1: The Rating Bar, the Button, and their Listeners

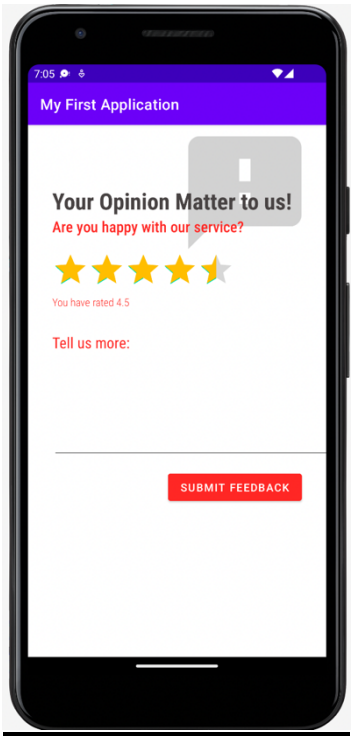

Figure 1: Example UI for Task 1

The naming convention used in this demonstration is:

- RateBarFeedback – to obtain rating from the user.
- TVRating – to display the rating given by the user.
- ETFeedback – to obtain user feedback in text.
- BtnFeedback – to submit feedback to the app.

In this demonstration, we will learn how to use the Rating Bar (RateBarFeedback) and implement its relevant listener – OnRatingBarChangeListener. When the rating bar is clicked (and dragged) by the user, the ratings will be displayed on the Text View (TVRating).

Besides, after the user enters the feedback in the multiline Edit Text (ETFeedback) and clicked the submit button (BtnFeedback), the button OnClickListener will be triggered to display a toast message to the user.

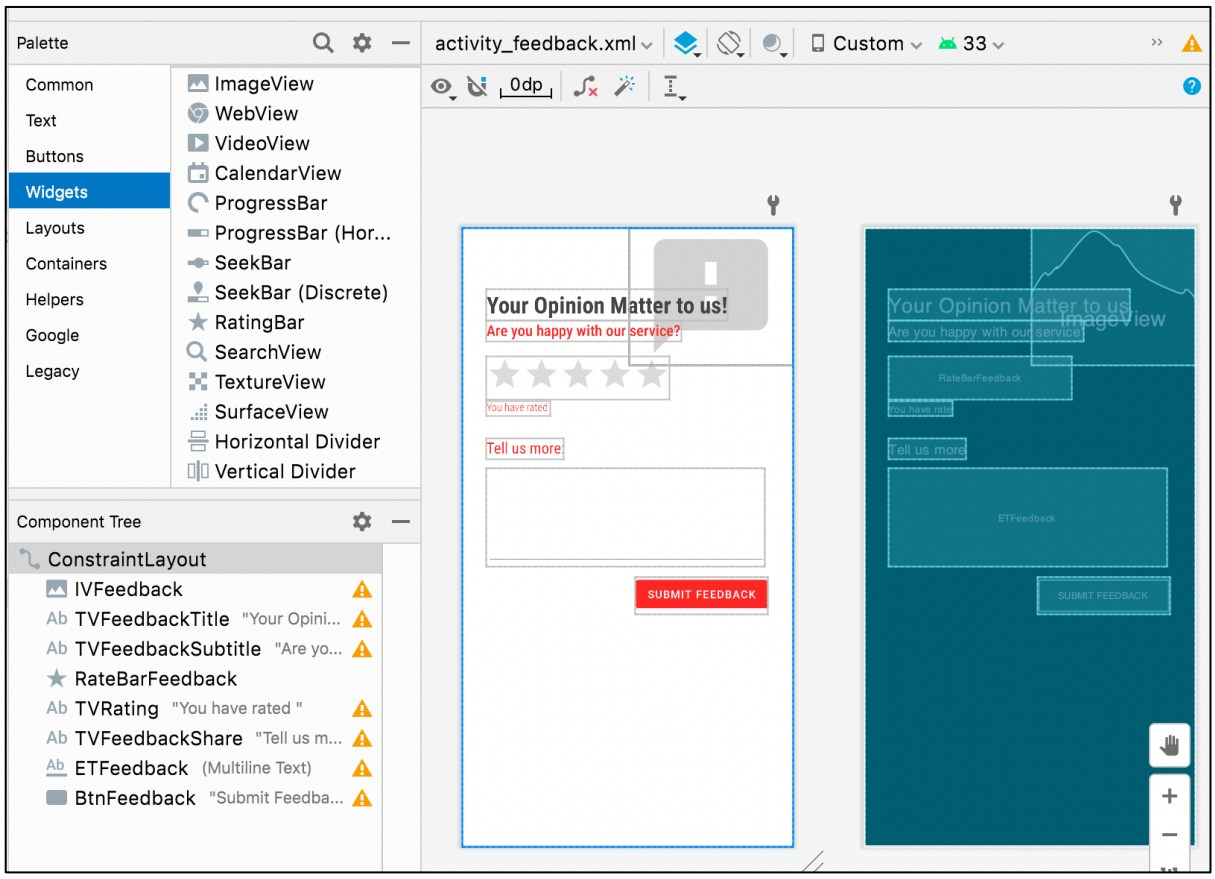

Figure 2: Drag and drop the Rating Bar from the Palette.

First, prepare the required UI widgets / views. You can obtain the Rating Bar view under the "Widgets" category from the Palette Panel. Drag and drop it into your activity. In addition, you can customize the number of stars, the progressTint attribute (color applied when it's clicked by user), etc. depending on your needs.

FeedbackActivity.java

```java
public class FeedbackActivity extends AppCompatActivity {
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_feedback);
        //Connect with UI Widget RateBarFeedback, TVRating, ETFeedback, BtnFeedback
        RatingBar RateBarFeedback = findViewById(R.id.RateBarFeedback);
        TextView TVRating = findViewById(R.id.TVRating);
        EditText ETFeedback = findViewById(R.id.ETFeedback); //needed if you want to collect the feedback later
        Button BtnFeedback = findViewById(R.id.BtnFeedback);

        //The button OnClickListener to toast a message and share the cashback code is user has entered the feedback.
        BtnFeedback.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                String message = "Thank you for your feedback!";
                if (!ETFeedback.getText().toString().isEmpty())
                    message = message + "Please enjoy your RM5 Cashback: ABC123";
                Toast.makeText(getApplicationContext(), message, Toast.LENGTH_LONG).show();
            }
        });

        //The rating bar OnRatingBarChangeListener to change the rating whenever it is used by user.
        RateBarFeedback.setOnRatingBarChangeListener(new RatingBar.OnRatingBarChangeListener() {
            @Override
            public void onRatingChanged(RatingBar ratingBar, float rating, boolean fromUser) {
                TVRating.setText("You have rated " + rating);
            }
        });
    }
}
```

Figure 3: The backend code for actual implementation.

In your .java file, use the codes in Figure 3 to set the OnRatingBarChangeListener and OnClickListener for our desired action.

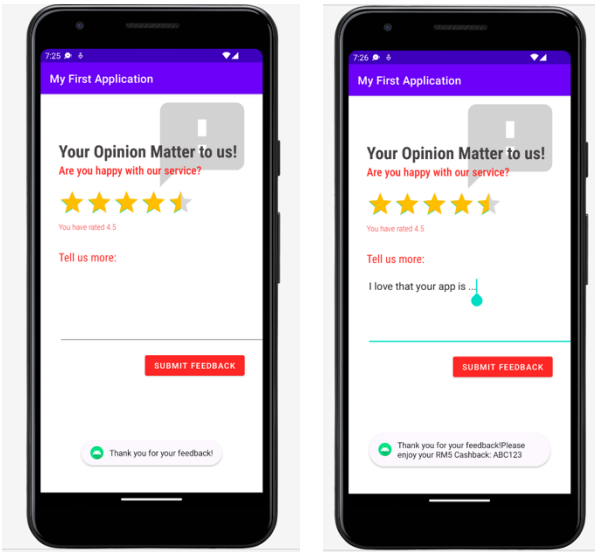

Figure 4: The toast messages.

For the button OnClickListener, a toast message with a cashback code will be displayed if the user has entered his/her additional comments to the ETFeedback. No cashback code is given if there is no additional comment entered.

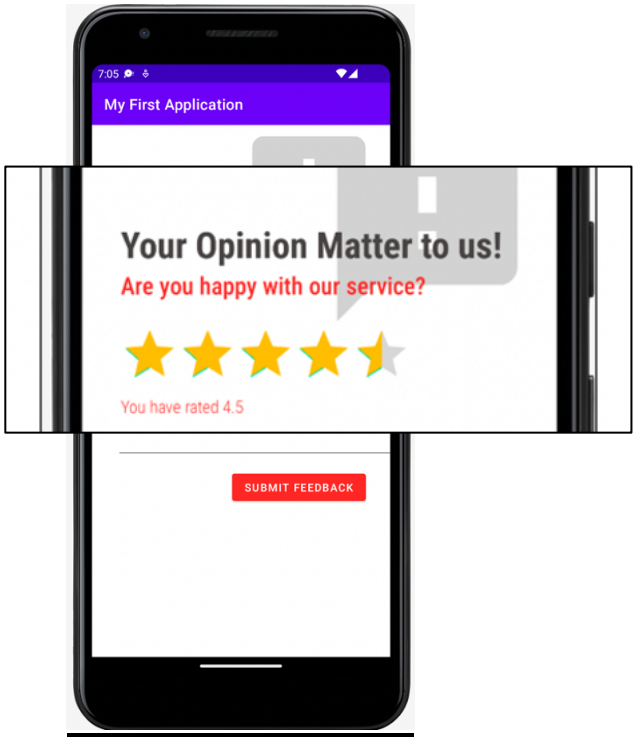

Figure 5: The text view will reflect the stars selected by the user.

As for the Rating Bar, when the user clicked (or dragged) the rating bar, it will be instantly reflected in the text view by using OnRatingBarChangeListener.

Task 2: Implement your Feedback Activity

Now, modify your UI draft from Tutorial 4 to incorporate with Rating Bar.

Then, create a new activity in "My First Application" (from Practical 3) named "FeedbackActivity", and implement the frontend and backend of your modified UI draft (after adding the Rating Bar).

FeedbackActivity.java AndroidManifest.xml

```xml
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    package="com.example.myfirstapplication" >

    <application
        android:allowBackup="true"
        android:dataExtractionRules="@xml/data_extraction_rules"
        android:fullBackupContent="@xml/backup_rules"
        android:icon="@mipmap/ic_launcher"
        android:label="My First Application"
        android:roundIcon="@mipmap/ic_launcher_round"
        android:supportsRtl="true"
        android:theme="@style/Theme.MyFirstApplication"
        tools:targetApi="31" >
        <activity
            android:name=".FeedbackActivity"
            android:exported="true" >
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />
                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>
        <activity
            android:name=".MainActivity"
            android:exported="true" >
        </activity>
    </application>
</manifest>
```

Additional information: How to change the first activity to run when the mobile application launches?

In your AndroidManifest.xml file, place the <intent-filter> … </intent-filter> tag (which consists of the action.MAIN and category.LAUNCHER) on the activity that will be started first. In this example, I have moved the <intent-filter> from MainActivity to FeedbackActivity.

Submission

Submit the Android Studio Project to Spectrum.