Mobile App Development Assignment 1

ASHWIN PREM
100805031

SOFE 4640U
CRN 44434
Date: October 2, 2024

Table of Contents

Table of Figures

GitHub Link

https://github.com/ashwinprem/MobileApp_Assignment1

Objectives

Develop a mobile application using Android Studios that calculates Equated Monthly Installments (EMI) based on the amount of mortgage amount, tenure, and interest rate. The app emphasis topics learned thus far in class specifically, layouts, views, and intents along with best practices with mobile application development.

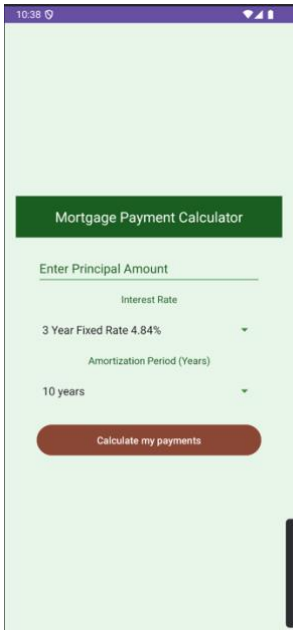

Figure 1: Main Activity/Homepage of EMI App

Activity: Main Activity (Homepage)

This the first of the two activities used in the app. The Main Activity is where the user is able to input in their principal amount, select their interest rate from the available dropdown menu, and finally select their tenure from another dropdown menu. This calculation used a fixed monthly frequency to make matters simple.

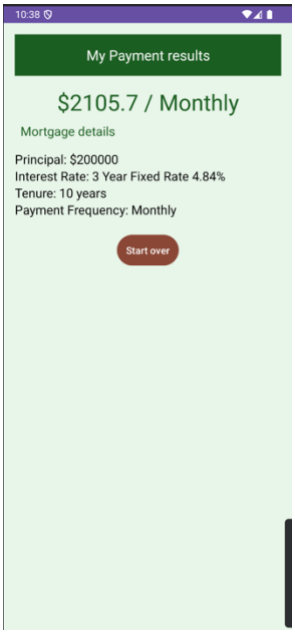

Figure 2: New page to display calculated EMI results

Activity: Result Activity

This activity was created just to showcase the use of intents and how navigation between two activity elements can be done. This also copies the structure of the website that was given to use as reference. This activity also utilizes and intent to navigate back to the main activity to allow the user to input new values.

Layouts and Views

Layouts and views in mobile app development go hand in hand. Layouts define the structure of a view (linearly, items stack on top of each other, split into grid format), and they are a type of view in themselves. Views or specifically basic views represent basic user interface elements like buttons, dropdown lists, text fields.

There are two main views: home page, and result page, I structured the pages largely using the .xml files. I followed a linear layout for both views where the other layout objects are stacked one after another.

TextView was used for any elements that just needed to display text, such as the title of the app in the home page. *EditText* layout is used to allow the user to enter their principle amount. The value input would then be grabbed in the backend java file for the calculation. *Spinner* was used to create dropdown lists, mimicking how the sample websites handle the interest rate and amortization period. Finally, *Button* was used to create the buttons to navigate from one activity view to another. This leads us now to intents.

Intents

Intents are messages, often sent between activities, and in the case of this app, an intent in one activity invoke the other activity. Specifically , in the home page activity, when the user clicks the button to calculate their EMI, the resultant page activity is invoked using an intent. Using the *putExtra* method, data calculated in the first activity is passed onto the result activity to be displayed over there. Refer to Figure 3.

```
// Pass data to ResultActivity
Intent intent = new Intent( packageContext: MainActivity.this, ResultActivity.class);
intent.putExtra( name: "EMI", emiFormatted);
intent.putExtra( name: "PRINCIPAL", principalStr);
intent.putExtra( name: "INTEREST_RATE", selectedInterestRate);
intent.putExtra( name: "TENURE", selectedTenure);
intent.putExtra( name: "PAYMENT_FREQUENCY",  value: "Monthly"); // Hardcoding as Monthly
startActivity(intent);
```

Figure 3: Example of intent code used to invoke the result activity, and also pass data from one activity to another.

A similar approach is used to navigate back to the main activity from the result activity view when the button is clicked. Of course, the intent is triggered by placing an *setOnClickListener* on both buttons.