SOFE4640 Mobile Application Development Assignment 1 - Luke Hruda - 100654666

https://github.com/LukeHruda/MortgageCalculator

Introduction

This report will go over both the general breakdown of how the assignment was completed and then cover how the requirements of using Intents Views and Layouts were used. The formula used to calculate the required payment by month is as follows:

$$M = P[i(1 + i)\hat{}n]/[(1 + i)\hat{}n{-}1]$$

Monthly Payment = Price*(Monthly interest(Monthly interest + 1)^months) / ((1+Monthly interest)^(months)-1)

General Overview

The way I approached this assignment was to use the preset basic activity with Android studio and then removed the second fragment and navigation activities, until I was left with the Main Activity and First Fragment only. Thus starting the app showed one page, with the “Next Button”, however clicking this button would do nothing. I added a relative layout to the first fragment and 3 number inputs to read the Mortgage Price, Interest Rate and Payment period. For Mortgage price, it is an integer. Interest Rate is a double, in which i asked the user to enter it as a percent rather than a decimal value so 6 rather than 0.06. Payment Period is also an integer, and it is in months. I chose to ask for input in months rather than years as this would allow the user to be more specific in their choice for the remaining term if they wanted to do 38 months 3.16 years etc. Upon entering the values and pressing the “Calculate” button, the following operations occur in this order. The program performs error checking to ensure all three values are entered, as null values would cause the program to crash. It does this by parsing the textbox value to a string, and then checking if the value is empty (or null), if this is detected, an alert box pops up to notify the user that a textbox is empty and halts the code execution. Prompting the user to acknowledge this to clear the alert box from the screen. If all the checks pass, the code reads the numbers and passes them to my calculate function, which uses the formula from the introduction to calculate the monthly payment amount. This is then passed to the Intent which calls my DisplayCalc class. The DisplayCalc class takes the calculated amount and its initial components and formats them into the textview in my popup.xml file. Once there, it would grab the device display manager and popup the popup layout with the calculated amount. Once the user is satisfied with this, they can hit the “Confirm” button, which would call finish on the intent and return the user to the main page.

Intents

The intent that I use in my project is to change the display to show the value of the calculated monthly payment. This is used in conjunction with layouts, so I will talk about this more in that section as well. The main objective of the intent is to pass the data using a bundle. Particularly I pass the user entered values of Mortgage Price, Interest Rate and Length along with the calculated monthly payment amount. Once the DisplayCalc class receives this information. It formats it into the Popup.xml layout file, and displays it. The intent is cleared when the user hits the "Confirm" button on the popup layout, which calls the intent finish method, thus returning the user back to the First Fragment where they can enter in new information if desired.

Layouts

The mortgage calculator app utilizes two main layouts. The first of these being the first fragment layout. This utilizes a relative layout container, which in it holds four textviews, three edit texts and a button. The button labelled "Calculate" is constrained to the bottom of the page, while the title text view is constrained to the top. The remaining three text views are stacked vertically and have a corresponding edit text field set to numerical values only to their right, forming a three deep two across grid pattern. This layout does not change via the program and is only used as a main page to handle user input.

The second layout file is the popup.xml file. This contains only a TextView and a button. This layout is called within my DisplayCalc class once the intent is submitted. This layout is dynamically formatted in the following manner. The content view is set to the popup layout. The code grabs the text view and writes in the monthly payment amount. A display metrics variable is created that grabs the screen size of the device using the app in width and height. The window is then set to be 80% of the width and 60% of the height, thus appearing as a popup.

Views

Views are used throughout the app to perform changing of the display information. Two of the notable examples of this are firstly, mentioned above, where the DisplayCalc class called from the intent is used to shift the view from the First Fragment to the popup view so that the results can be displayed to the user. The second area is in my error checking of the user inputs in the first fragment. I do this by referencing the current view to create an Alert Dialog box that features an "Ok" button used to cancel the alert and revert back to the current view. These popup boxes appear whenever the user tries to perform the calculation when one of the input boxes is empty.

Conclusion

In conclusion I utilize views, intents and layouts to create a mortgage calculator that shows the correct amount per monthly payment when given the Mortgage price, Interest rate and Payment period length.