

Overview

In this brief report, I will discuss my use and implementation of Views, Intents, and Layouts for the creation of a mortgage calculator app for Android. Using these different components, I can streamline a user-friendly interface designed for many different android platforms and utilize different design tools and concepts shown in the course. I was able to take the different classes of views and implement them in a way that allowed me to create an easy-to-use app that does your equated monthly installment (EMI) calculation in 3 simple steps.

I developed this application with the flow of data in mind. The first component of calculating EMI is to understand the number of terms, the interest rate, and the present value of the loan. I created a main activity with 3 fragments. In the first fragment, I prompt the user to get started and in the second fragment I ask the user to enter the loan tenure as a value for the number of years. Since EMI is calculated as a monthly payment, I will convert the number of years into a value in terms of months. In the third fragment I ask the user to enter the value of the loan, and the interest rate that the user will receive. At any time, the user can go back during this process and to do this I have implemented a navigation feature in order to get the most flexibility for the app. The user can do this in the title bar, or from the android system UI button. When the user is satisfied with the values they have entered, they can select the calculate button in the third fragment. Using a bundle, I can pass all the collected data from all 3 fragments to another activity. I use an intent to perform a context switch, and using the entered values, I calculate the monthly mortgage payments for the user. Then the user can go back and use this calculator as many times as they need.

Intents

Intents are a tool for switching activities or contexts. In my application I implement an explicit intent, which tells Android to perform a context switch to another activity part of my application. In my mortgage calculator, I switch the context to calculate the users EMI from data that I pass from the main activity. This is done through a bundle which is passed part of the intent. A bundle lets you store key value pairs which correspond to information past from different views where a user has interacted with the app. In my application, once the user is done and happy with their calculated EMI values, I use another explicit intent in order to switch back to a fresh new instance of the main activity. Intents are a great way to go back to loading screens or switch to brand new forms, activities, and layouts because I don't need to clear or reset any of the content in my views and layouts. In fact, the intent reloads the page as new. While not implemented in this application, there are other types of intents such as an implicit intent. In an implicit intent, I can pass information to the android operating system to search and carry out tasks or other actions. For example, I could use an implicit intent to launch another application or open a web link.

Layouts

Layouts are great for designing vibrant portable applications that run on many devices. There are different types of layouts such as a Linear layout or a constraint layout. These different layouts have unique features that let you build applications to run on different android devices. For example, using a constraint layout, I can target all sorts of android devices and know that my content will fit within the bounds of the constraint and wrap content so that words and important information isn't cut off. A layout is a way of constraining views or other layouts so that data and information can be organized in an easy-to-understand way for the end user. In my mortgage calculator I utilize many constraint layouts along with linear vertical layouts. I do this to let information flow on the screen in a well-spaced out easy to read manor, but I also do this so that if a screen is too small or not wide enough, content will be wrapped and displayed appropriately for the user's device. My app has larger fonts and not every device will be able to render a full line of text, so using a constraint layout allows me to wrap content so that it is compatible with the user's device. I also use a linear layout with the constraint such that information will be spaced out, but relative to the user's screen size to best utilize the devices resolution.

Views

In Android, views are the UI's building blocks that let you organize information, take in user input, and display data. Different views have different methods and abilities, and some view's are even represented in groups. In my application I utilize TextViews, Buttons, and Number Views to display information to the user, and get user input for my mortgage calculator. View groups allow for groups of data to be collected all at once. For instance, a View group of Radio Buttons is very useful on forms for collecting data that the user should only be able to enter 1 choice. Views can be used to create many other components, and can be versatile enough to be constrained with layouts, and contained by many different activities to provide the best user experience.

Conclusion

Views, Layouts, Intents are extremely powerful design tools at the finger tips of developers. I can utilize these tools to make applications a lot simpler than originally thought by combining the versatility of these tools. I can make different Intents for different data flows, I can use layouts to organize data, and I can use views to display and get data.