Bit Me - Bitcoin Conversion Application

developed by Keawa Rozet

**About**

Bit Me is a Bitcoin Conversion application that was issued for me to develop during my interview process at ClassCalc by Daniel Haiem. Bit Me currently contains 20 currencies of which it is able to convert to today’s Bitcoin value. It utilizes asynchronous http calls to retrieve the data, error handling to for stability, and a clean UI to present the user with their information.

**How to use**

Upon first open, the user will be prompted for their name. After typing in their name and hitting submit, they will be brought to a new screen that asks them to select the currency they wish to be converted from Bitcoin. The user selects their currency on the drop-down menu and clicks Convert to proceed. A short animation reveals the conversion of 1 Bitcoin to their selected currency. Clicking Return will allow the user to select a new currency.

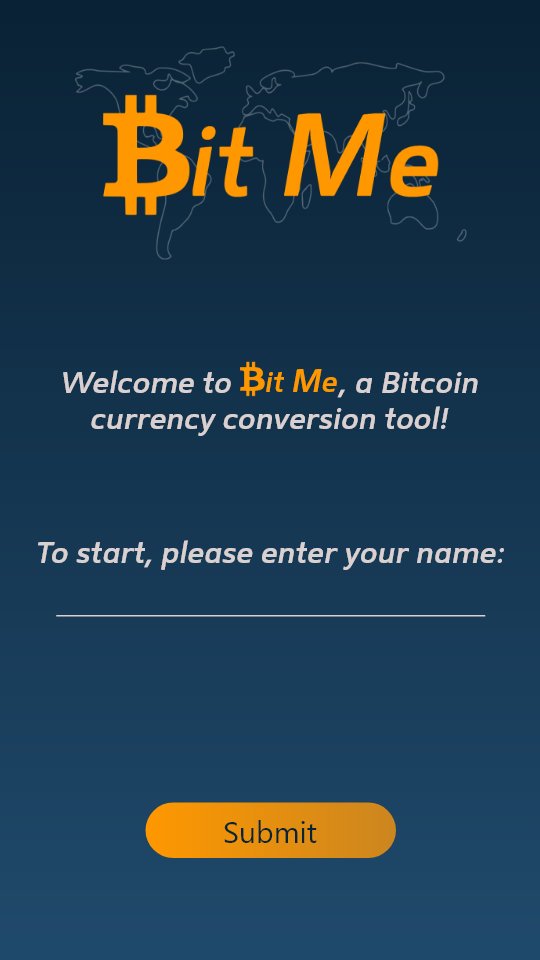
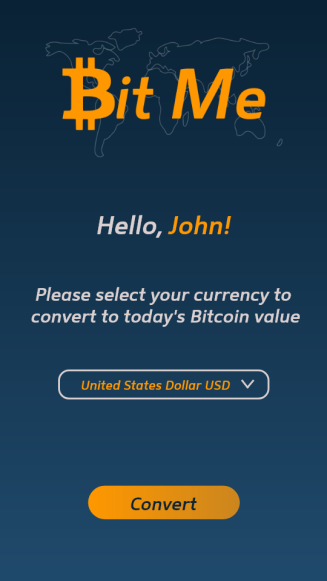
**Development process - Mock Ups and UI**

I always begin my applications with creating a Mock Up design (if one isn’t presented to me). To do this, I use a combination of Adobe Photoshop, Adobe XD, and Zeplin. I always use free assets that have been released under Creative Commons CC0.

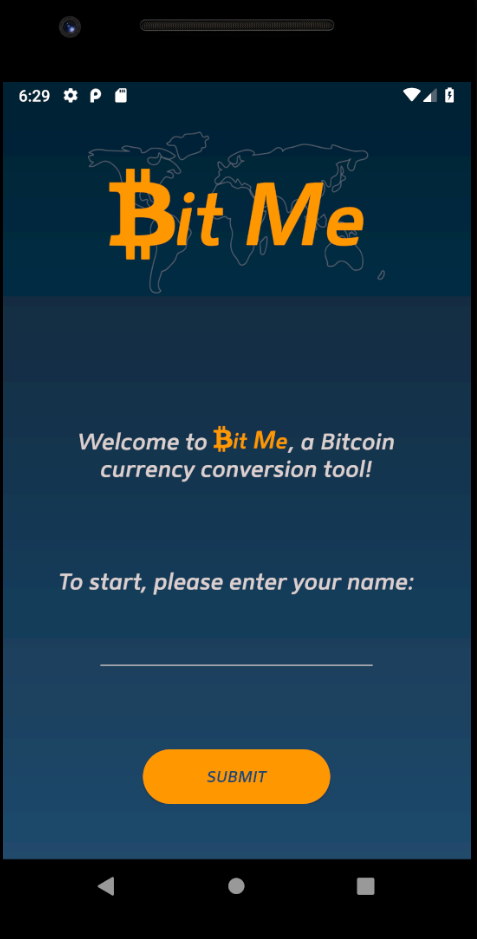
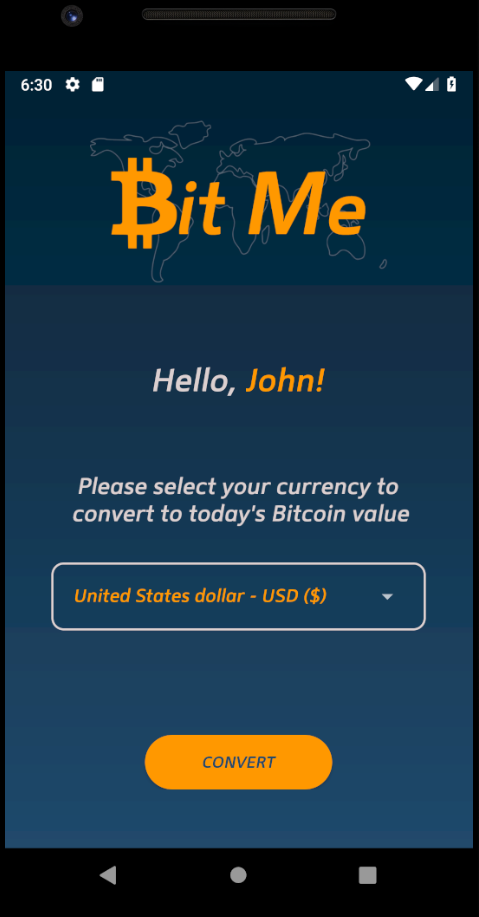
Next, I begin with creating a barebones version of the UI based on the mock ups created. Scalability is a high priority and I design my UI’s to accommodate for this. Upon creating all the Activities I plan on using, I design my UI in such a way that allows me take more creative liberties that doesn’t involve very much backtracking over my code.

Below you can see a comparison of my Mock Ups and Implementation to see that I follow my Mock Ups very closely, but deviate on small matter to improve the design.

*Mock Ups*

**

*Implementation*

**

**Checklist for Requirements**

Since this was issued to me to complete, I have included the guidelines for the project presented to me as well as where to find the location for completing those guidelines in the code.

1. Build an app that when you open it for the first time, it asks for  
   your name and persists it on the phone without using a database.

RequestCurrency.java - line 102

InputName.java - line 42  
2. After that, it goes to a screen that says "Hello, !"  
and asks you which currency you'd like to use to convert from today's  
Bitcoin value (you can choose three or four predefined currencies options  
to make things simpler and easier, or offer a search bar/scroll view with  
10-20 types).

CurrencyConversion.java - line 56

RequestCurrency.java - line 82  
3. After confirming the desired currency, it takes you to another  
screen showing how much is 1 BTC converted to that chosen currency.

RequestCurrency.java - line 91  
4. After you minimize or close the app and open it again, it should  
not ask you for your name again, but instead take you directly to the  
second screen where it says "Hello, !" and asks about the  
desired currency.

RequestCurrency.java - line 183  
5. Library to use: [http://loopj.com/android-async-http/](http://loopj.com/android-async-http/" \t "_blank) (android),  
[https://github.com/Alamofire/Alamofire(iOS)](https://github.com/Alamofire/Alamofire(iOS)" \t "_blank)

RequestCurrency.java - line 91

**Notable code**

As you will see, all my code is sufficiently documented and upholds strong coding practices that makes it easy to read. That said, I will touch on a few pieces of code that might need more explanation.

*InputName.java, onClick(), line 38*

Notibile sections of this code include validateName(), line 56, and the usage of sharedPreferences. SharedPreferences is what is used to allow the user’s name to persist even when the application is closed. The name is first validated by validateName() and then is committed with the id of userName.

validateName() checks if the name entered is empty, contains symbols, and is over19 chars. If any of these are true, then the user must enter a new name to proceed.

*RequestCurrency.java, checkPreviouslyStarted(), line 102*

This is what checks if the user is starting up the application for the first time. It checks SharedPreferences if “isFirstStart” is true, and if it is, it opens the InputName activity. Once the user inputs their name, the result code is checked (line 123) and “isFirstStart” is set to false.

*CurrencyConversion.java, retrieveConvertedCurrency(), line 56*

Here I use the requested AsynchHttpClient class to make a call to free.currencyconverterapi.com to receive the converted currency. I first take in the user’s selected currency and use it to make the appropriate query in the url. I then check the data received and process it if it is valid data.