How to Use this Template

1. Make a copy [File → Make a copy...]
2. Rename this file: "**Capstone_Stage1**"
3. Replace the text in green

Submission Instructions

1. After you've completed all the sections, download this document as a PDF [File → Download as PDF]
2. Create a new GitHub repo for the capstone. Name it "**Capstone Project**"
3. Add this document to your repo. Make sure it's named "**Capstone_Stage1.pdf**"

GitHub Username: SingularityFuture

BlockWatch

Description

BlockWatch tells the time on your phone and/or Android Wear device by transforming the latest transaction hashes from the BitCoin blockchain in a visually compelling way, powered by the Blockchain.info API.

Intended User

The app will appeal to all of those who really grok everything Bitcoin and blockchain related (like me!) and want an accessory that is connected to real-time Bitcoin data.

Features

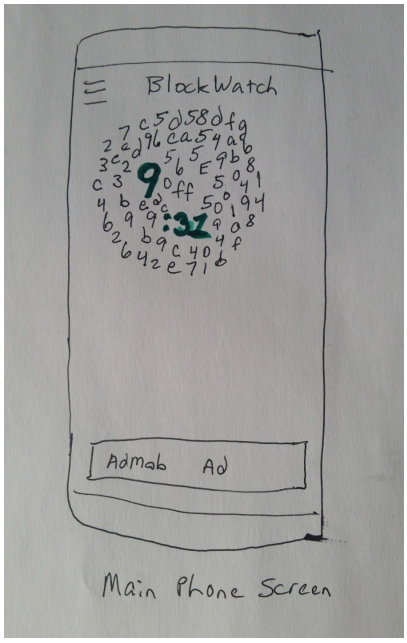
List the main features of your app. For example:

- Retrieves real-time transaction information from the blockchain using an API
- Converts a transaction hash from the blockchain to a visual compelling watchface
- Will post watch notifications for Bitcoin price alerts (possibly in a future version)

User Interface Mocks

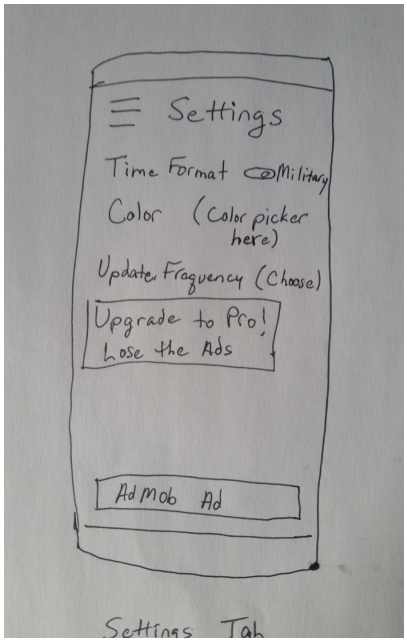
These can be created by hand (take a photo of your drawings and insert them in this flow), or using a program like Photoshop or Balsamiq.

Screen 1



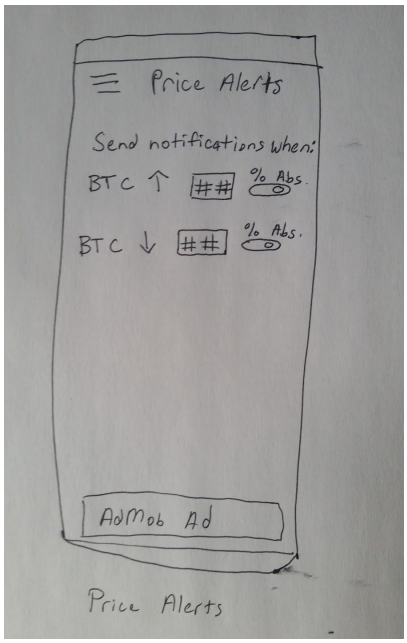
This is the Main Screen, where users can see a phone version of the ‘watchface’, which is the entire hash of a recent transaction on the blockchain, with particular numbers emphasized that correspond to the current time. There will be an ad at the bottom for the free version and a hamburger or other type of menu for Settings at the top. Obviously, one of the hardest things about this project will be displaying the hash in a visually compelling way and programming that!

Screen 2



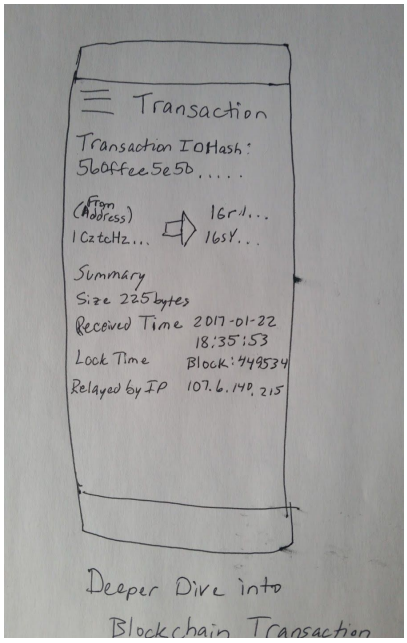
Here is a prototype for the settings menu. Here, they can change the time format to military time, perhaps pick the colors they'd like on the watch (maybe only in the paid version), change the update frequency (which will only change how often the watch retrieves transaction information, not the time frequency itself) in order to save on battery, and give them an option to upgrade to the pro version for more features and to remove the ads.

Screen 3



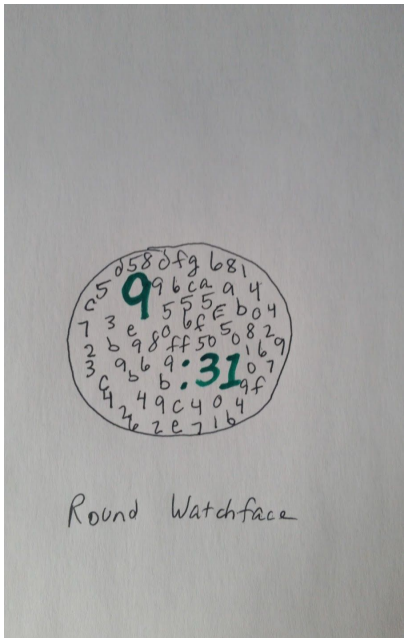
Here is an example of how the Price Alerts section could look. It will allow them to set up notifications if the price of Bitcoin goes above or below a certain amount (or moves a certain percentage).

Screen 4



For fun, I will allow the user to dive deeper into a particular transaction of a time. For example, if they press on the watchface on the wearable, it could trigger an activity to open on the phone showing the details of the transaction whose hash was showing on the watch at that time. These details include things like the from address of the transaction, the to addresses, the IP address that relayed the transaction, whether any scripts were associated with it, etc. For an example, you can look at: https://blockchain.info/tx/f65368100a7a93be33ca131ef6b0cb58102c70c009ce1bbcd485ba85aa0a2a56

Screen 5



Here is an example of a watchface layout. I'm mainly thinking it will be a kind of chaotic scattered presentation, with the time numbers standing out in emphasis. Of course, in a paid version, I could create several new types of layouts and styles.

Key Considerations

How will your app handle data persistence?

I will use Firebase as the authorization system and to store user preferences in a Firebase database. I will use a Loader to load the data from the Firebase system into the app layout.

Describe any corner cases in the UX.

The main app itself should be fairly straightforward at first. The main things they can change will be settings, like color, layout, etc., so it will be a back and forth between those kind of settings (probably from a slide-out menu) and the main screen which will show the time. I don't think there will be any 'corner cases' per se. However, there is a case in which a particular hash on the blockchain does not contain the digits necessary to create the current time. In this case, I will have to figure out a compelling way to still show the time while indicating that certain digits are outside the current hash.

Describe any libraries you'll be using and share your reasoning for including them.

I will be using the Blockchain.info API, which provides an endpoint that sends the data about the latest blocks being published on the blockchain and the transaction hashes inside that block. I will then transform each hash that I retrieve from the API into a cool, visually compelling watchface!

Describe how you will implement Google Play Services.

I will be using Admob to display ads on the free version of the phone side of the app. I will create a paid version that turns these ads off.

I will also use Google Analytics to start analyzing user behavior throughout the app. This will be very useful in the future as well - I would like to extend this app to more functionality in the future. The watchface will be a novel draw that gets people in, and then I can make their usage sticky with added functionality.

Next Steps: Required Tasks

Task 1: Project Setup

I will explore the Blockchain.info api to determine which endpoints will be necessary to get the relevant data to create my project. I will do the preliminary steps to setup the API calls and appropriate URI structures, etc. I will then start setting up my Firebase structure and exploring how to integrate that with the rest of the code.

Task 2: Implement UI for Each Activity and Fragment

- Build UI for MainActivity - this will be the initial login screen.
- Build UI for Settings - this will allow the user to enter in preferences for how they want the watchface to appear (colors, general layout, military time, etc.) There are a lot of other neat features in Blockchain's api that I can integrate onto the watchface too over time - the user will be able to configure which features they want to show in the Settings as well (e.g., like pushing a notification or simple icon to the watch every time a new block is verified on the blockchain).
- Build the layout for the watchface and a phone widget counterpart. This will involve setting up a few layouts that look cool - the hash is a long string of digits and characters and I will have to decide how to emphasize the numbers inside the hash that correspond to the time.

Task 3: Configure the API data to represent the time.

I will have to filter the API data with an algorithm that compares the digits in the current hash to the current time - it will find numbers that match the hour and minute of the current time. This will be a relatively straightforward search algorithm. If no numbers are found in a particular hash that match the time, I can figure out a cool way to still represent the time while visually indicating that a minute or hour number lies 'outside' the hash.

Task 4: Integrate Google AdMob and Analytics

- I will integrate AdMob into the free version of the app. This will show a simple banner ad at the bottom of the phone part of the app if they choose the free version.
- I will integrate Google Analytics into the platform as well in order to understand behavior across the app.

Task 5: Clean it up and publish

- After making sure everything works as planned, I will publish the app.

Submission Instructions

1. After you've completed all the sections, download this document as a PDF [File → Download as PDF]
2. Create a new GitHub repo for the capstone. Name it "**Capstone Project**"
3. Add this document to your repo. Make sure it's named "**Capstone_Stage1.pdf**"