CryptoTracker - Product Backlog

Daniel Tracy, Josh Hiatt, Brian Long, Nathan Adduci, Peter Jung

Problem Statement

Cryptocurrency wallets and trading platforms often lack the features of a stock trading platform. Bitcoin, Etherium and many other cryptocurrencies often fluctuate at the same rate and volatility of a stock market, as the trading which takes place is electronic and can yield similar results to the stocks. We intend to make an app mimicking popular stock trading apps, with more in depth analysis of the graphs of cryptocurrencies and a broader selection of trading options normally only supported by the stock market.

Background Information

Trying to trade cryptocurrency can be difficult, and at times like the old days before stock exchanges exist. We are creating a solution to add the trading options and analyses available to stock markets to the cryptocurrency market, in an effort to take out the middle man for a lot of people. This iteration will only be a demonstration model, using an api to get data and keeping mock values to people's wallets.

Environment

We would like our application to be available on any browser with our front end composed mainly of HTML5 and CSS for formatting using Firebase CLI for Web UI. User information is very important to the functionality of CryptoTracker which is why we would like to store all user information using Firebase so that it can be retrieved and updated at any time. Since our application is a web application we are going to have to use something to host the API calls and constant service/backend so we are going to host a server on a raspberry pi. Lastly we need a service provided to us to retrieve the information of each cryptocurrency so we are going to use Nomics.

Functional Requirements

Backlog ID	**Functional Requirement**	**Developer**	**Hours**	**Status**
1	As a user, I would like to create a	DONE	5	Planned for

	CrptoTracker profile DONE			Sprint 1
2	As a user, I would like to log in with my profile DONE	DONE	2	Planned for Sprint 1
3	As a user, I would like a persistent account DONE	DONE	3	Planned for Sprint 1
4	As a user, I would like to see the price of my coins DONE	DONE	3	Planned for Sprint 1
5	As a user, I would like to see a graph of my coins price over time.	DONE	2	Planned for Sprint 2
6	As a user, I would like to see comparison graphs between two coins	DONE	4	Planned for Sprint 2
7	As a user, I would like to buy coins DONE	DONE	2	Planned for Sprint 1
8	As a user, I would like to sell coins DONE	DONE	1	Planned for Sprint 1
9	As a user, I would like to be able to view a portfolio / summary of my wallets. DONE	DONE	4	Planned for Sprint 1
10	As a user, I would like to be able to see value changes in my profile.	DONE	3	Planned for Sprint 1
11	As a user, I would like to be able to see the price of a coin over custom time ranges	DONE	1	Planned for Sprint 2
12	As a user, I would like to have the option to own multiple types of cryptocurrencies DONE	DONE	2	Planned for Sprint 1
13	As a user, I would like to see my trade history DONE	DONE	3	Planned for Sprint 1
14	As a user, I would like to cancel a future trade	DONE	2	Planned for Sprint 1
15	As a user, I would like to be able to recreate past and cancelled trades.	OUT	3	Planned for Sprint 1
16	As a user, I would like to be able to get help on the various trade types and call types.	DONE	6	Planned for Sprint 1

17	As a user, I would like to be able to make a buy limit order	DONE	2	Planned for Sprint 2
18	As a user, I would like to be able to make a sell limit order	DONE	2	Planned for Sprint 2
19	As a user, I would like to be able to make a stop buy order	DONE	2	Planned for Sprint 2
20	As a user, I would like to be able to make a stop sell order	DONE	2	Planned for Sprint 2
21	As a user, I would like to have a leaderboard to compare my trading skills to others	OUT	2	Planned for Sprint 2
22	As a user, I would like to have three choices of a starting amount of USD DONE	DONE	1	Planned for Sprint 1
23	As a user, I would like to have the ability to add friends to a friends list	OUT	2	Planned for Sprint 2
24	As a user, I would like to view a leaderboard of only my friends	OUT	2	Planned for Sprint 2
25	As a user, I would like to have a notifications page		3	Planned for Sprint 1
26	As a user, I would like to be notified of incoming friend requests	OUT	2	Planned for Sprint 2
27	As a user, I would like to be notified of automatic sell orders	DONE	1	Planned for Sprint 2
28	As a user, I would like to be notified of automatic buy orders	DONE	1	Planned for Sprint 2
29	As a user, I would like to be able to see my friends on the leaderboard	OUT	1	Planned for Sprint 2
30	As a user, I would like to be able to friend people from the leaderboard	OUT	2	Planned for Sprint 2

Buy limit order - buy assets at or below specified price
Sell limit order - sell assets at or above specified price
Buy Stop order - buy assets at or above specified price (Safe as to not buy volatile stocks)
Sell Stop order - sell assets at or below specified price (Safe to stop big loses)

Non-Functional Requirements

Architecture

Our application will be delivered in the form of an React / JavaScript web application. Firebase will provide us with a hosting device, as well as our database. The server will therefore be written in React and / or JavaScript. We will use Nomics' free API to query the prices of their 13 supported cryptocurrencies, which in turn will be the 13 cryptocurrencies we support. Our graphs will be generated using Plotly's open source graphing library for JavaScript.

Security

When it comes to security we would like to make user information as secure as possible by using existing user logins such as Github and Google so that we know the information stored is safe. We will be using firebase functionality in order to keep the users information not only secure, but also easily accessible.

Usability

The usability of our application should be extremely simple. Any user that knows how to use a basic application on a computer will be able to navigate our easy, fun, tool. Trading fake cryptos in CryptoTracker will be very easy so that every user can have a good and educational experience.

Performance

Server response time < 10s, web client application < 1gb. We will be hosting our website and storing our data in Firebase, using their free options for a web server and database. Our frontend will be React / JavaScript, our backend will be Node.js. These technologies will hopefully allow us to meet our server response time and web client application size specifications. Our web client will be responsive within one second of clicking, our graphs generated will be static.

Use Cases

Case 1: Creating a profile

Action	System Response
1.User clicks "Create Profile" or "Create Account"	2.UI to enter user credentials for account creation appears
3. User enters credentials and clicks "Create"	4.Users information is saved in firebase and

	the account is created leaving the user signed in

Case 2: Log into a profile

Action	System Response
1.Click “Login”	2.UI to enter user login credentials appears
3.User enters login credentials and clicks “Login”	4.The information is checked against the database and responds with a successful login or an incorrect login credentials response

Case 3: Automatic saving of wallet information

Action	System Response
1.After any action IE buy/sell/trade a message is sent to the system to save the users information	2.When the system receives information that the users wallet has been changed in any way the updated wallet is synced to the database saving its current state.

Case 4: Checking the price of coins

Action	System Response
1. User clicks on a coin of their choice	2. The system receives the request to view information on a certain currency and responds with a page showing that information

Case 5: Checking graphs over time

Action	System Response
1. User clicks on a coin of their choice	2. The system receives the request to view information on a certain currency and responds with a page showing that graphs, statistics, etc.

Case 6: Comparing two coins graphs side by side

Action	System Response
1. User clicks on a coin of their choice	2. The system receives the request to view information on a certain currency and responds with a page showing that graphs, statistics, etc.
3. User clicks another coin of choice and chooses “compare” option	4. System responds with a page showing a comparison of the two coins in either one or two graphs.

Case 7: Buying coins

Action	System Response
1. User clicks on a coin of their choice	2. The system receives the request to view information on a certain currency and responds with a page showing that graphs, statistics, etc.
3. User clicks the buy option	4. System displays a dropdown menu showing different buy options
5. User selects option and clicks submit	6. System processes buy order and updates the users wallet

Case 8: Selling coins

Action	System Response
1. User clicks on a coin of their choice	2. The system receives the request to view information on a certain currency and responds with a page showing that graphs, statistics, etc.
3. User clicks the sell option	4. System displays a dropdown menu showing different sell options
5. User selects option and clicks submit	6. System processes sell order and updates the users wallet

Case 9: Portfolio/Summary of wallets

Action	System Response
1. User clicks on profile button	2. The system receives the request and responds with Users page displaying all relevant information, basic user info, wallet balances, user history, etc.

Case 10: Updating a users wallet

Action	System Response
1. User completes any sort of buy, sell, or trade order in order to update wallet	2. The system receives the buy/sell/trade order and proceeds to update the user's wallet, updating it globally for the user in their profile screen and any other instances of the users wallet

Case 11: Monitoring a coins activity over a custom time range

Action	System Response
1. User clicks on a coin of their choice	2. The system receives the request to view information on a certain currency and responds with a page showing that graphs, statistics, etc.
3. User clicks on graph settings to edit to a custom time range	4. A dropdown menu is displayed showing different time ranges for the user to select
5. The user picks a specific time range	6. The server responds with a new graph displaying only the specified range given by the users selection

Case 12: Owning multiple cryptocurrencies

Action	System Response
1. When purchasing coins the user has the option to purchase one of several types of	2. The system receives the buy order and proceeds to update the user's wallet now with

coins.	multiple currencies.

Case 13: Trade history

Action	System Response
1. A signed in user clicks on profile to display their profile information	2. The system responds with the users page displaying all of the relevant user information
3.The user clicks on the “Trade History” tab	4. The system responds with a list format of all previous buy/sell/trade orders that the user has completed in the past along with current buy/sell/trade orders.

Case 14: Cancelling future trades

Action	System Response
1. A user makes a buy/sell/trade order of any type.	2. The system accepts with buy/sell/trade order and creates the order in the system
3. At any time the user clicks the cancel option on the buy/sell/trade order	4. The system removes the order and cancels and future trading that would have happened after the user clicked cancel

Case 15: Recreating past and cancelled trades

Action	System Response
1. A signed in user clicks on the profile tab	2. The system responds with the users page displaying all of the relevant user information
3. The user clicks on the “Trade History” tab where there past and cancelled trades are displayed	4. The system responds with a list format of all previous buy/sell/trade orders that the user has completed in the past along with current buy/sell/trade orders
5. User clicks option to create buy/sell/trade order based off of a previous listing	6.System receives the buy/sell/trade order and initiates it

Case 16: Help Buttons

Action	System Response
1. A user selects the trade menu	2. System serves the trade menu
3. User clicks the help button	4. Information on the currently selected trade (or how to select a trade from the dropdown) is displayed in a popup.

Case 17: Buy Limit Order

Action	System Response
1. A user selects the trade menu	2. System serves the trade menu
3. User selects buy limit order from drop down	
4. User inputs value of trade in number of coins or USD	
5. User places order	6. Order is logged in database
	7. Every 15 minutes the server checks if the trade can execute
	8. Trade is executed or cancelled
	9. Update user’s wallet to reflect purchase

Case 18: Sell Limit Order

Action	System Response
1. A user selects the trade menu	2. System serves the trade menu
3. User selects sell limit order from drop down	
4. User inputs value of trade in number of coins or USD	
5. User places order	6. Order is logged in database

	7. Every 15 minutes the server checks if the trade can execute
	8. Trade is executed or cancelled
	9. Update user’s wallet to reflect purchase

Case 19: Stop Buy Order

Action	System Response
1. A user selects the trade menu	2. System serves the trade menu
3. User selects stop buy order from drop down	
4. User inputs value of trade in number of coins or USD	
5. User places order	6. Order is logged in database
	7. Every 15 minutes the server checks if the trade can execute
	8. Trade is executed or cancelled
	9. Update user’s wallet to reflect purchase

Case 20: Stop Sell Order

Action	System Response
1. A user selects the trade menu	2. System serves the trade menu
3. User selects stop sell order from drop down	
4. User inputs value of trade in number of coins or USD	
5. User places order	6. Order is logged in database
	7. Every 15 minutes the server checks if the trade can execute
	8. Trade is executed or cancelled

	9. Update user’s wallet to reflect sale

Case 21: Portfolio Leaderboards

Action	System Response
1. User navigates to leaderboard	2. Serve leaderboard
3. User reviews leaderboard	

Case 22: Choice of Starting Portfolio Value

*subject to change

Action	System Response
1. User begins to create a portfolio	2. Log changes as they occur
	3. Offer three tiers of value to start portfolio at
4. User selects a portfolio starting option in USD and submits new account form	5. Log new account in database, including starting value

Case 23: Adding Friends

Action	System Response
1. User searches for target friend account	2. Serve list of matching accounts
3. User selects target account	4. Serve account profile
5. User clicks “Add Friend”	6. Update database
7. Reflect changes in UI	

Case 24: Friends Only Leaderboard

Action	System Response
1. User navigates to leaderboard page	2. Serve leaderboard
3. User selects “Friends Only” option	4. Filter and serve leaderboard

Case 25: Notifications Page

Action	System Response
	1. Crypto order goes through, user is notified
2. User clicks bell icon near profile in banner	3. Serve notifications page

Case 26: Friend Request Notifications

Action	System Response
1. Friend adds user as friend	2. Log request
	3. Send notification to target user
4. User accepts / rejects request	5. Log request answer
	6. Notify original Friend

Case 27: Sell Order Notifications

Action	System Response
	1. Sell order price is reached, stock sold
	2. Notify user
3. Click notification	4. Serve trade overview screen for sale

Case 28: Buy Order Notifications

Action	System Response
	1. Buy order price is reached, coin purchased
	2. Notify user
3. Click notification	4. Serve trade overview screen for purchase

Case 29: Highlight Friends on Global Leaderboards

Action	System Response
1. User navigates to leaderboard	2. Render leaderboard table
	3. Add coloration tags to any users who are friends with the original user
	4. Serve modified leaderboard

Case 30: Send Friend Requests from the Leaderboard

Action	System Response
1. User navigates to leaderboard	2. Serve leaderboard table
3. User clicks “+” button next to desired friend	4. Log friend request
	5. Notify target leader