StockMock

CS 407 - Software Engineering Senior Project

Design Document

Team 20:

Abid Kaisani

Nicolas James

Donald Bough

Harsh Parakh

Rohan Kashuka

Omar Raza

Table of Contents:

Purpose

There exist countless online stock trading simulators, such as Stockfuse, Stockwars, and Marketsim, however these apps are normally optimized for classrooms or very specific users. Due to this there are none that teach the basics of stock market trading in an entertaining way. It is easy to find expert analyst predictions for any given stock on the market but very difficult to learn about the opinions of your peers and other investors who are in a similar position as you, having almost identical portfolio values. Additionally, learning about different companies before investing in them and educating oneself about the market can be a formidable task by itself. Rather than visiting each company website or a common platform to get updates on news, StockMock provides an in-app news functionality that allows users to simultaneously keep themselves on track with information at the tip of their fingers. By gamifying the learning process and adding leader board rankings, the idea of working with stocks not only becomes more appealing but also a lot more fun.

We will not be designing a new system, we will be adding to an older app that was created in a 408 project. The previous StockMock app had a standard UI interface and several different pages that allowed a few different functions. One page contained a tutorial of how to use the app, another allowed users to use fake money to 'purchase' stocks. This allowed users to learn and track investments without worrying about losing real world money.

StockMock is targeted for a larger audience and we are adding new features to allow us to hit our broad target market of people who want to learn the stock market and interact with others also learning. StockMock is the best tool to safely learn the ins and outs of trading in an entertaining environment. Unfortunately, StockMock currently lacks much needed user to user interaction and available real-world information. We aim to add a forum to allow users to interact and learn from each other and allow real world news to be available to the user. We will also have additional functionalities such as a leader board and profiles to make StockMock more informative and immersive, along with updates to UI and to our bot.

Functional Requirements

As a user, I would like to,

News

1. I would like to access news related to the stock market.
2. I would like to access news from multiple sources.
3. I would like to find news articles about certain companies.
4. I would like to find the latest news articles.
5. I would like to know the source of the news articles.
6. I would like to get all the news articles by a source of my choosing.
7. I would like to find news articles in various languages.
8. I would like to access all news articles from the past thirty days.
9. I would like to be able to access the actual source of the news article.

Forums

10. I would like to be able to view all forums.
11. I would like to access and view a subtopic in an existing forum.
12. I would like to see the topic and the description for a post in the forum.
13. I would like to view the owner of a particular post.
14. I would like to be able to create a topic.
15. I would like to be able to respond to an existing post.
16. I would like to know the time when the post was created.
17. I would like to see the number of replies made to a post.
18. I would like to view all the members using the StockMock forum.
19. I would like to see the number of posts made on the forum.

Profiles

20. I would like to have my own username.
21. I would like to be able to edit my username.
22. I would like to be able to view my profile.
23. I would like my username visible to other users.
24. I would like to be able to edit my profile.
25. I would like to be able to view another user’s profile.
26. I would like other users to not view my private information. (email).

Leaderboard

27. I would like to view a leaderboard of users.
28. I would like to see the username of users with the most money.
29. I would like to see the rank of the users in the leaderboard.
30. I would like to see the amount of money the users have.
31. I would like the leaderboard to be ordered numerically.
32. I would like the leaderboard to be dynamic and change as the user accounts change.
33. I would like to click on a username and be able to see their profile.

Statistics

34. I would like to be able to view number users on StockMock
35. I would like to be able to view total number of shares owned on StockMock
36. I would like to be able to view top shares owned on StockMock
37. I would like to be able to see the number of unique companies traded on StockMock

Bot

38. I would like the bot to observe actions of users on the leaderboard and suggest stock that I could invest in, when I'm purchasing a trending stock
39. I would like the bot to provide suggestions for stocks I could invest in based off stocks I have already invested in, when I'm purchasing a trending stock
40. I would like the bot to become available whenever I go onto purchase stocks
41. I would like the bot to provide suggested stocks in the form of a list
42. I would like the list of suggested stocks, if any, to have a few suggestions so that the presented list is easier to understand and navigate.
43. I would expect the UI of the bot to have same layout regardless of on which page it's presented on
44. I would expect the UI of the bot to have helpful titles with each list that indicate how the list was generated

Tutorial

45. I would like a tutorial that shows how to search stocks, buy and sell stocks, and check my purchase history.
46. there should be a tutorial showing me how to transfer funds from a real stock app, how taxes with stocks work, and basic terminology from the stock market.
47. I would like a tutorial that shows how to search user profiles, check stock news, and use the Stock Mock forum.

Tech Debt

48. As a user, I would like an updated Menu bar with the new pages and with categorized drop downs that display relevant pages.
49. As a user, I would like a navigation bar in the glossary page that allows me to skip to the alphabetic letter of my choice.
50. As a user, I would like the fluctuations in the price of my shares to be dynamically updated in the database.
51. As a user, I would like my total amount of money to be dynamically updated based on the fluctuation of the values of my stock shares at the end of each business day.

Non-Functional Requirements

- Our application will be cross browser compatible.
- Our application will safely store user information.
- Our application will be fully explained with tutorials.
- Our application will be written well to allow fast response times.
- Our application will work in browsers of various device size.
- Our application will have an intuitive and well-designed user interface.
- Our application will have error message pop-ups that will help the users.
- Our application will be coded well so that it can be modified easily in the future and is also testable.

Usability

The user interface should be ameliorated throughout StockMock wherever there is a need for it. The user interface should allow the user to view statistics on stocks. Additionally, it should be capable to navigating within the StockMock forum. It should be able to create a bridge between the core functionality StockMock application and the Forum. When a user goes on to purchase stocks, it's crucial to the new StockMock experience that the stock purchasing workflow of the application is catered towards assisting users with making their investments. Considering that StockMock is a web-based application, it needs cater to countless devices, and so we need to ensure that the user interface is presented correctly on every screen.

Performance

Google released some research indicating that a user will leave a web page if it takes longer than 3 seconds to load. Based on this and person experience we would like to at the least keep our landing page on under a 3 second loading time. By having a fast loading time at the beginning, slower times for pages farther into using our website will be more acceptable since the user has already invested the time in using our website. Some pages, like stock search, have proven to be slower in the past due to issues with the API, which is out of our control. But by prioritizing the landing page to have the fastest landing time, we will improve the user's experience.

Scalability

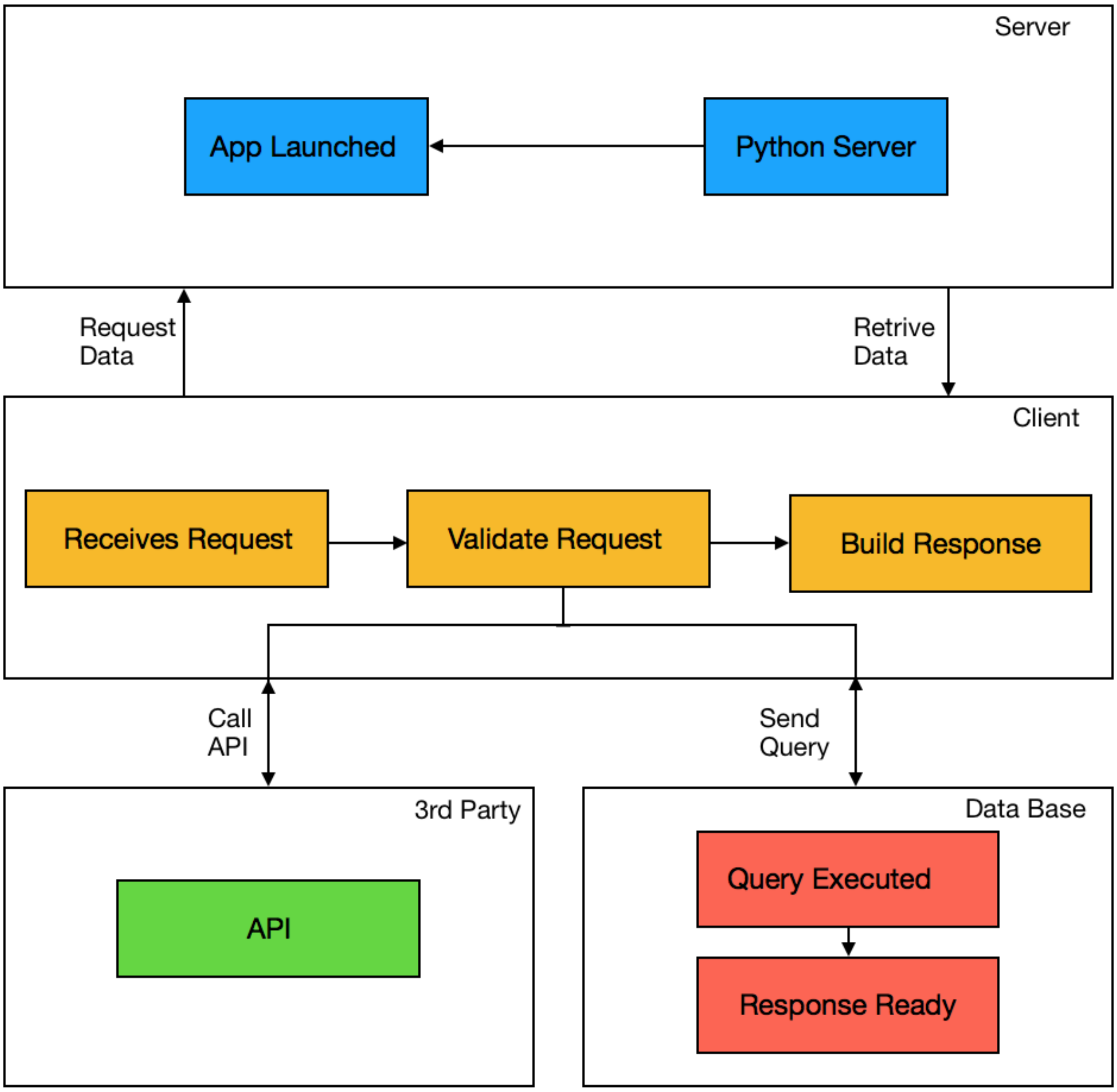
With Firebase as our database, Google has developed a suite of products around it to ensure that scalability comes with the ease of simply purchasing a different plan. Starting with their free version, we would have enough memory and power to support the number of students in CS 407 which is more than enough to test and present our website. This also applies to our news API where scalability is supported on a free, startup, and business level purchase plan. With the free plan on our news api, we can already make 1,000 requests per day which will again allow us to continue with testing and presenting our project.

Security

StockMock’s Backend functionality will be implemented on Firebase. Considering that StockMock portrays a user’s interests through their investments, we need to ensure that the user’s data is kept securely. Firebase is hosted on SSL which allows it to provide a secure connection between the host and the server. This ensures that StockMock transfers data to and from the server securely. Additionally, StockMock applies the most effective security rules that Firebase offers, where reading and writing to the database is only allowed if a user is authenticated. Additionally, StockMock should also ensure that when user data is being queried, a user can only query data for the user that is logged in. StockMock’s JSON will be structured and accessed in such a manner to ensure that users do not have access to other user’s data.

Design Outline

Our project is an application that teaches how to trade on the stock market in a fun, risk-free way. Our implementation will use a client-server-3rd party architecture. Our client will make requests and process data from the API and database. The server is simply a HTTP server that will serve html pages to users, immediately sending index.html to start the web application.



Web Client

A. The web client, running in the user's browser, will be the interface for our system.
B. The web client will make calls to the APIs, database, and also request and receive html pages from the server.
C. Data from the API and database is validated, then builds a response to display for the user.

Python Server

A. The server serves files and media from itself to the browser.
B. The server is a prebuilt Python SimpleHTTP server implementation.
C. The server fulfills GET requests for files and media from the client, and returns it back to the client.
D. *(Example)* Once the application starts, and the user visits the website on their browser, index.html is served to the browser from our server. Then, when they login, the home.html file is requested from the server. This is sent back and displayed in the browser.

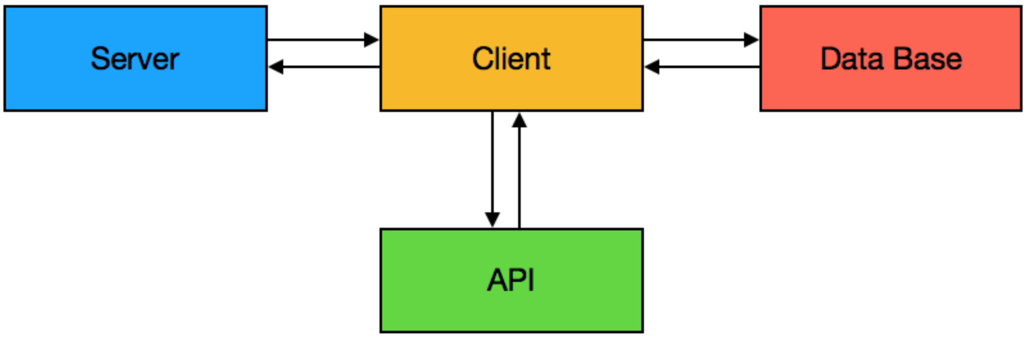
Database

A. The database will store all of the user data used in StockMock, such as user information, purchases, purchase history, forum posts, and forum comments.
B. The database will receive queries and store data from requests that are sent by the client.

3rd Party APIs

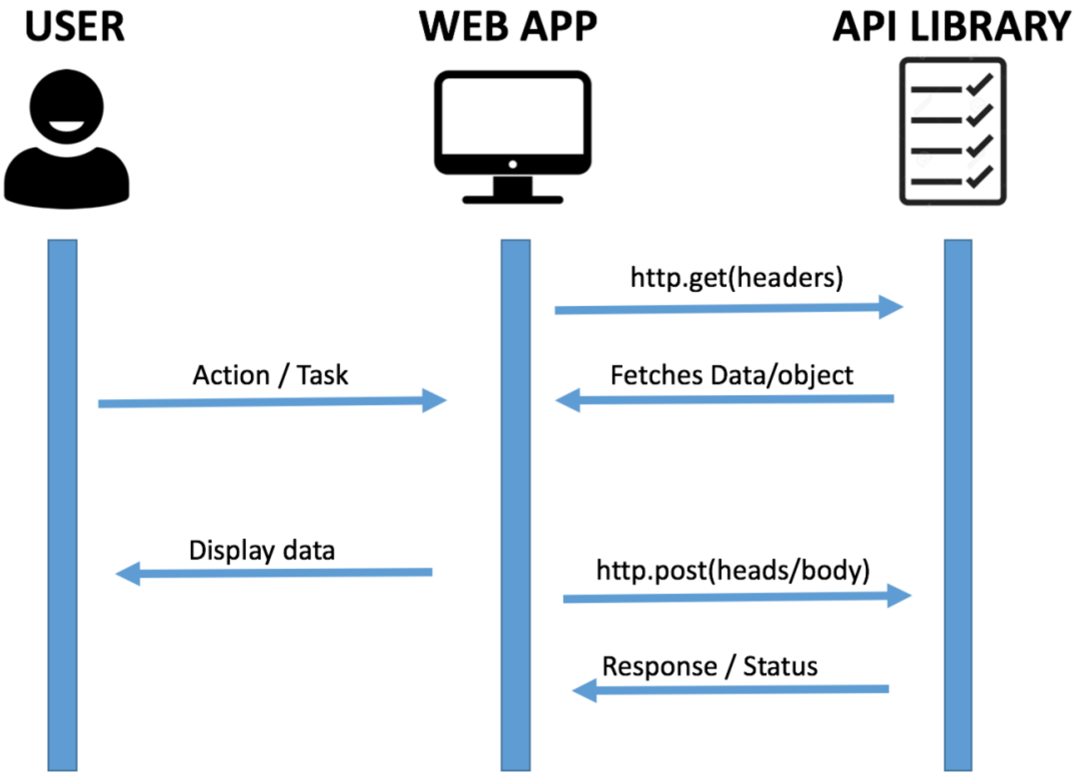
A. *(Financial Times API)* This API will be the source for financial news in our application. Since we are primarily concerned with top headlines in the stock market, we will only be using the top stories from Financial Times API. We also plan to allow users to see the full story with the APIs headline URL.
B. *(Alpha Vantage API)* This API will be the source for stock prices in our application. From the current price, to the past timeline of prices for a stock, Alpha Vantage provides everything we need when analyzing stock prices.

High level Overview



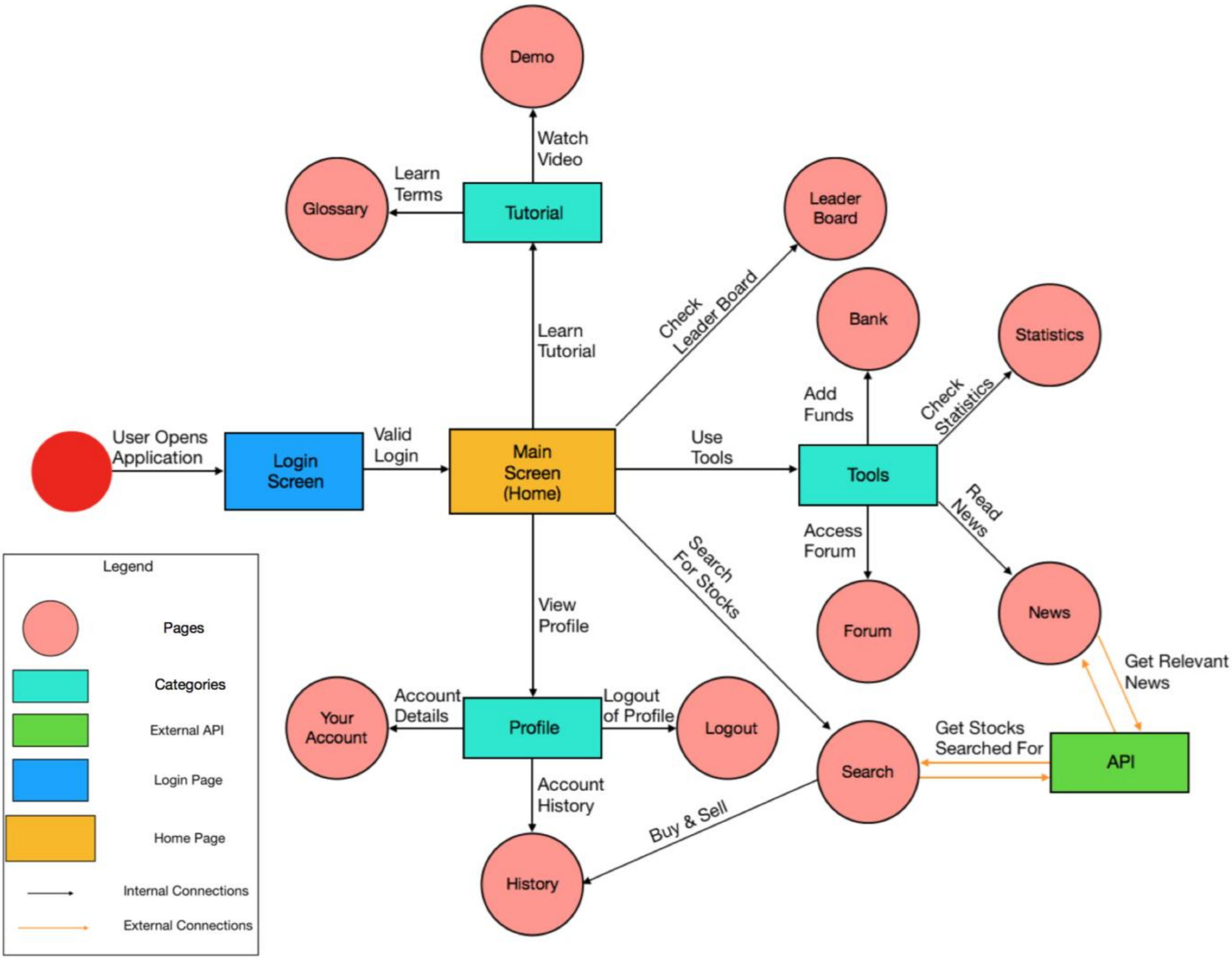
The server will be Python's SimpleHTTP implementation, which is simply serving index.html on application start along with serving other requested html pages. The client is "heavy" because all API and database parsing and requests will be made within it. All user actions are processed by the client. When the user requests data that comes from the database or an API, our client makes the request and then renders the data within our html. This is then served to the user through the server.

API flow diagram



StockMock has a standardized API communication protocol, this starts by the user requesting information. Our web application will then either use POST or GET and return the response status and body. This data is then formatted and displayed to the user.

Activity State Diagram



When the user opens the application, they will be asked to either login or sign up. After they sign in, they will be sent to the Home page which will have a menu bar that will allow them to access all pages. Pages with relevant content have been grouped into Categories in the Menu bar.

In the Tutorial category, the user can navigate to the Demo page which contains a video which demonstrates the multiple features and components that comprise StockMock. The user can also go to the Glossary page which has a list of financial and banking terms that allows them to have an enriched financial vocabulary and a place to have a trusted dictionary.

The user can also go to the Leader board page from the home screen. The Leader board allows the user to see the stocks being traded and the history of the top 10 traders on the StockMock app, allowing users to see what stocks and strategies have worked for the top users.

The Tools category allows the user to use tools to help them understand and better use StockMock. The Bank feature allows the user to add or remove funds from their account. The Statistics page allows the user to see aggregate statistics of the entire web app, including amount of stocks traded and most shares in company. The news feature allows a StockMock user to access real-time news related to their companies and general market information from an external API. Users will also have access to an in-site forum which allows them to communicate with others about a variety of topics and subtopics. StockMock also has a built-in search engine that allows you to add shares of stocks from companies by calling an external API which keeps real time stock prices and shares. The stocks added in the search functionality are also reflected in the history page.
The last category is the Profile category. You can access and edit your account details under the Your Account page, delete your account with the Delete Account page, and see stock purchase history under the History Page

Design Issues

Functional Issues:

1. Do users need to login to use our application?
   Option 1: No login required.
   Option 2: Allow the creation of login ID using Facebook, Google+, etc
   Option 3: Create username and password that is unique to our service.

   Decision: For our application, we would like an easy setup and high security. This could be provided by a by a third-party system such as Facebook login, Google+ etc. However, we also want to personalize our app and create user storage in the database system to account for all activities the user makes in the application such as displaying their usernames on the leader board and access their profiles. For each user, a new unique slot is created in our database. Hence, we have decided to go with Option 3.

2. Will users be able to view all information about other users?
   Option 1: Certain private user information will not be shown.
   Option 2: All information will be shown.
   Option 3: Users will be able to choose what information is viewable.

   Decision: Certain information such as email or how many shares a user has purchased needs to private, for privacy and fairness. For example, if the number of shares purchased are viewable it allows other users to calculate how much money was spent on them and so on. Since there will not be many fields within a user's profile we will just have a few select fields permanently private. Additional user functionality is not needed.

3. How do I get to view the news on the application?
   Option 1: The user can view a pop-up version of the article by hovering over the article title.
   Option 2: Each article title will lead to the original page where the user will view the news on the publisher's website.
   Option 3: You only get headlines but no other information.

   Decision: Displaying the entire article would make option very difficult to implement because of the length of some articles. Option 3 would not provide enough information, so we will link the news to the publishers website.

4. How do I add myself to the leader board?

Option 1: You need to earn the rank by being a top 10 performer on StockMock.
Option 2: There exists an option that you choose in order to display your name on the leaderboard.
Option 3: Only the administrators of the app show up on the leaderboards.

Decision: This is the best and fairest way to rank the top users according to the profits they have gained from the share they have purchased.

5. How can a user change their information?
   Option 1: They will not be able to change their info.
   Option 2: They will be able to change their info on their profile page
   Option 3: They will be able to change certain info on an edit profile page.

   Decision: We decided that certain info should be allowed to change and that a separate page should be used for that, so that no accidental changes are made. Creating a page solely for profile updates is less confusing for the user and easier to implement.

6. Is the content you post or already posted on the forum moderated?
   Option 1: No, you can post anything you want.
   Option 2: The content is moderated and users can be banned for posting unsuitable content on the platform.
   Option 3: You cannot create posts and hence there would not be any moderation required.

   Decision: The forum exists so that users can get opinions from other users in similar situations as them. It can be helpful to gain this information when dealing with the market and making transactions, however, one shall not use the forums to discuss other topics that are not related to the market since there exist other sources and channels to do so rather than the StockMock forums.

Non - Functional Issues:

1. What frontend framework would StockMock use?
   Option 1: AngularJS
   Option 2: React
   Option 3: VueJS

   Decision: Between the three options, Angular is easy to scale because of its powerful Command Line interface and design. It also becomes an important factor since the application was previously coded in AngularJS as well and hence, it can involve a lot more work to rewrite everything again using a different library. With that being said, the team of developers working on StockMock also have a lot of expertise with AngularJS and that reduces the learning curve and helps us perform faster.

2. What type of architecture will our application use?
   Option 1: Model View Controller
   Option 2: Client Server
   Option 3: Microservices

   Decision: For the basis of our project, the closest architecture that matches our app is the Client Server model. Although, we will be client heavy. What that means is that we do a lot of the major functionality on the client side such as making requests, parsing them, and displaying the results to the users. All that the server provides for us is loading the necessary web pages of the project. Even the interactions with the database take place through client itself.

3. What APIs will be used for fetching the news?
   Option 1: Financial Times API
   Option 2: XigniteNews

   Decision: Financial Times API is free of charge where as XigniteNews provides us the data with a small fee. Hence, since both the functionalities of the two options were very similar we decided to go with Financial Times API. The only limitation to the free package was the fact that we can only make 1000 requests to the end point however, that is more than enough for the traffic faced by StockMock.

4. What type of database will StockMock use?
   Option 1: SQL
   Option 2: NoSQL
   Option 3: None

   Decision: We have decided to go with a NoSQL type database due to the object-oriented programming that is easy to use. NoSQL also provides superior performance to regular SQL databases and allows scalability. NoSQL is also able to handle large amounts of structured or semi-structured data which is very helpful when handing stock data.

5. What database system will this application implement?
   Option 1: MongoDB
   Option 2: Firebase
   Option 3: Hadoop

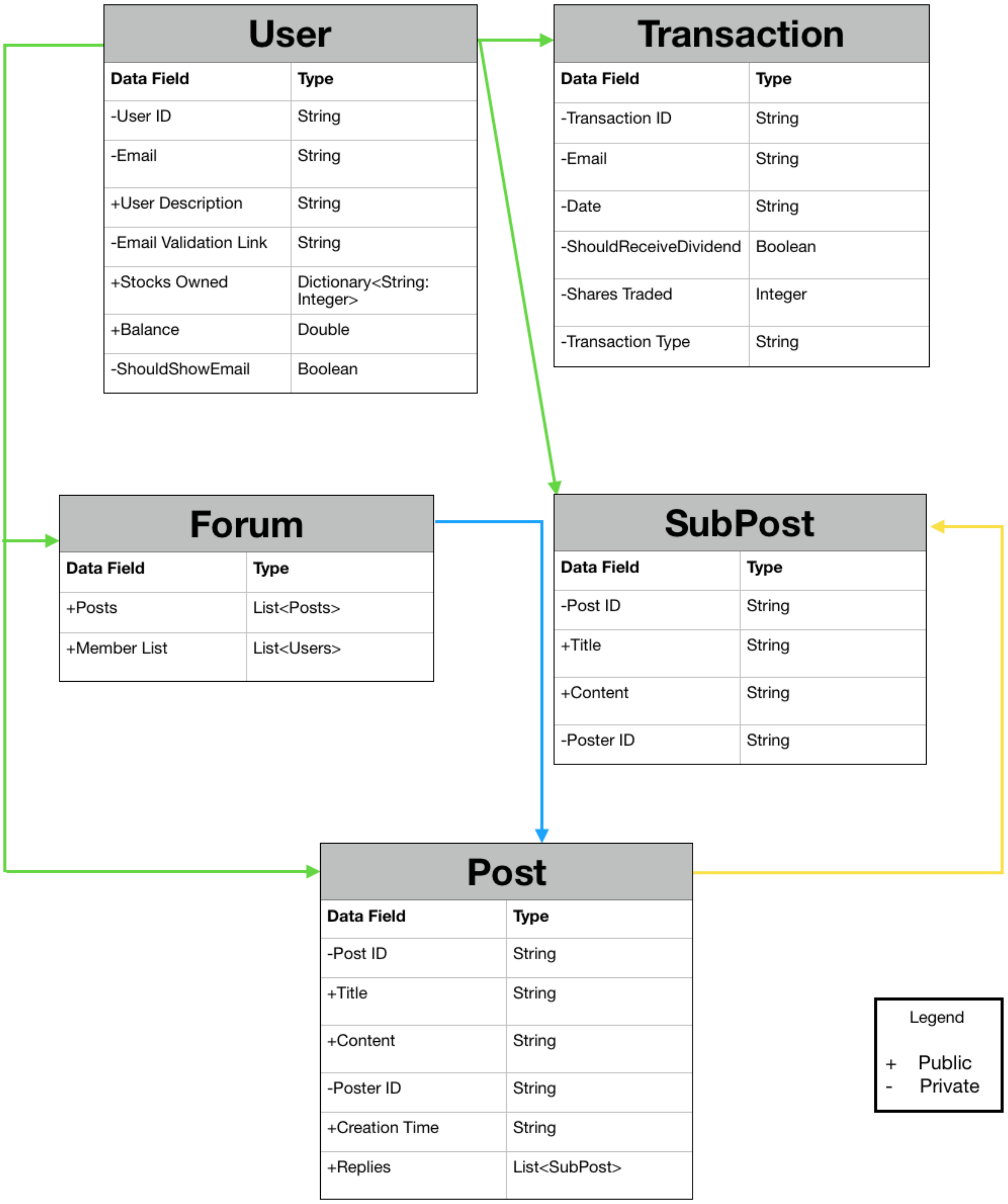
   Decision: It seems that Firebase has a solution for everything. Looking at their site, they have ads, analytics, push notifications, full document storage, and more. Although we might use these features at minimum, it is still very helpful if we wish to add them in the future. Apart from this, some other helpful properties of Firebase include High speed, cross-platform integration if we ever want to release a phone app and real time connection to the back end. Most of the team members have used Firebase in the past and also find it easier to follow the documentation which is laid out very well by Google.

6. What api will we use to pull share/stock prices?
   Option 1: IEX API
   Option 2: Alpha Vantage API
   Option 3: Quandl

   Decision: We have decided to use Alpha Vantage API because it provides real time and historical data in multiple variations. Alpha vantage also contains several other APIs that are useful for acquiring additional information of stocks. IEX while entirely useful is still being worked on and has limited support. Quandl is a simple API and only provides end of day stock prices and not historical.

Design Details

Data Class Level - Design



User

Data Field	Type
-User ID	String
-Email	String
+User Description	String
-Email Validation Link	String
+Stocks Owned	Dictionary<String: Integer>
+Balance	Double
-ShouldShowEmail	Boolean

Transaction

Data Field	Type
-Transaction ID	String
-Email	String
-Date	String
-ShouldReceiveDividend	Boolean
-Shares Traded	Integer
-Transaction Type	String

Forum

Data Field	Type
+Posts	List<Posts>
+Member List	List<Users>

SubPost

Data Field	Type
-Post ID	String
+Title	String
+Content	String
-Poster ID	String

Post

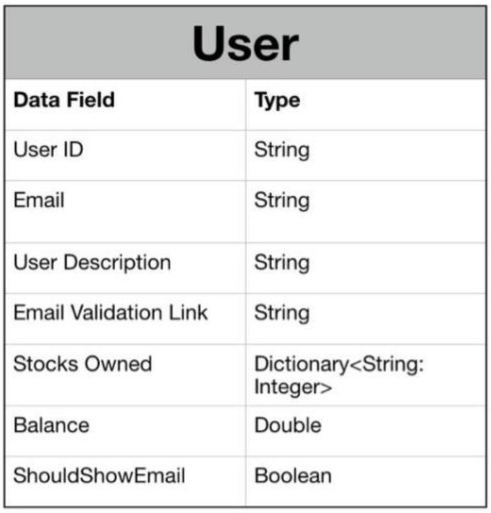
Data Field	Type
-Post ID	String
+Title	String
+Content	String
-Poster ID	String
+Creation Time	String
+Replies	List<SubPost>

Legend

+	Public
-	Private

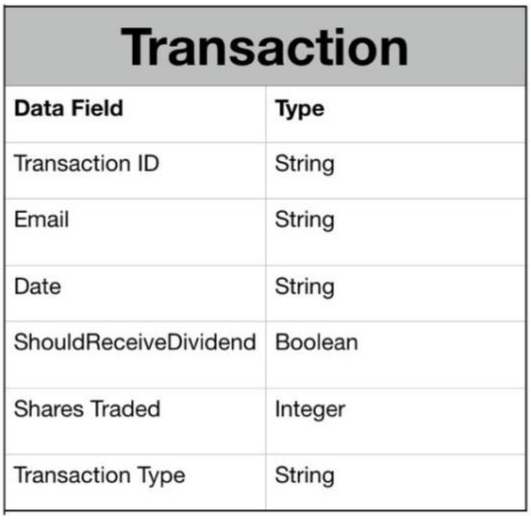
Database Design

NoSQL Database in Firebase will have the following node structures:



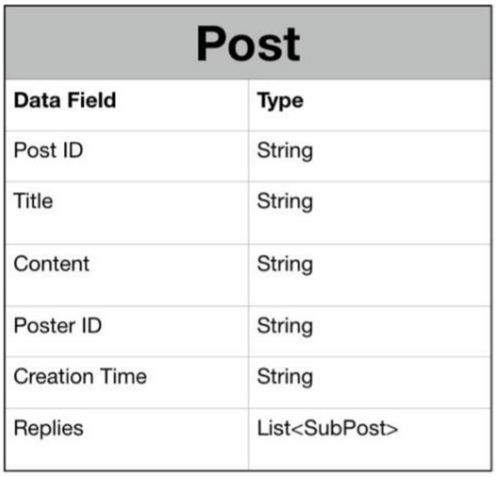
User

Data Field	Type
User ID	String
Email	String
User Description	String
Email Validation Link	String
Stocks Owned	Dictionary<String: Integer>
Balance	Double
ShouldShowEmail	Boolean



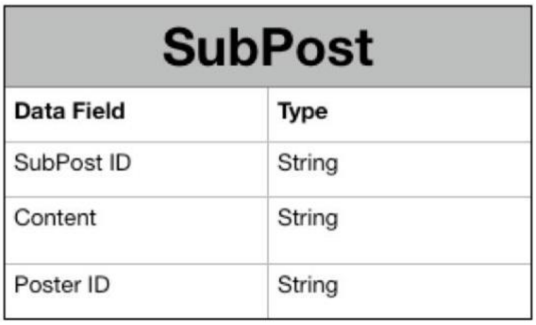
Transaction

Data Field	Type
Transaction ID	String
Email	String
Date	String
ShouldReceiveDividend	Boolean
Shares Traded	Integer
Transaction Type	String



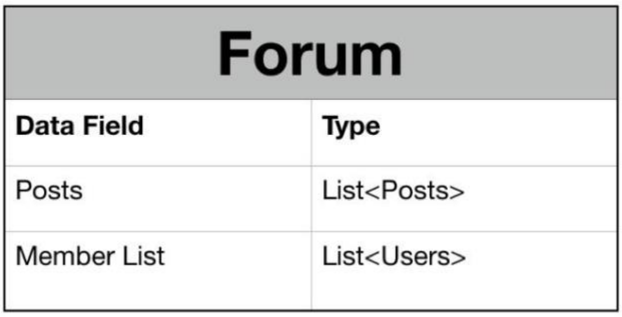
Post

Data Field	Type
Post ID	String
Title	String
Content	String
Poster ID	String
Creation Time	String
Replies	List<SubPost>



SubPost

Data Field	Type
SubPost ID	String
Content	String
Poster ID	String



Forum

Data Field	Type
Posts	List<Posts>
Member List	List<Users>

Description of Database Nodes:

The schematic diagram above describes database nodes and how they are used to represent the user's interactions. The nodes are described below as follows:

a. User
   i. Every StockMock User will be represented using the key values in the User node. These fields enable stock mock to figure out what user actions should be made available for a particular user.
   ii. To uniquely identify a user, every user is provided with a User ID (UID) which is useful when actions are performed that are focused upon one user
   iii. The user's email is stored in the Email Field as a String and is made available if the user decides to show it in their profile for other users to use if they wish to contact them via email.
   iv. Every user is allowed to share a bit about themselves and their interest in the StockMock application, we store their descriptions in the Description field.
   v. The users current standing in terms of the StockMock financials is represented by the Stocks Owned Field and the Balance field indicating what stocks the user owns and their current funds in the StockMock environment respectively.
   vi. The user has the ability to choose whether they want their email to be visible to other StockMock users. Whether they want their email to be visible on their profile stored in the field ShouldShowEmail.
   vii. Finally, to ensure that every on StockMock is a valid user, we store their account validation link sent to their email in a field called Email Validation Link and present it to the user until they have validated their account.

b. Transaction
   i. Every transaction that is made in StockMock will be represented by the fields in this node
   ii. Every transaction will be uniquely identified by a transaction ID to help identify the transaction
   iii. Each transaction will indicate the type of the transaction i.e. whether stocks were bought or sold. Additionally, each transaction would also have a field that indicates how many stocks were traded. Both of these values will be represented by the Transaction Type and Stocks Traded fields respectively.
   iv. This node also has fields ShouldReceiveDividend and Date to indicate whether the user should receive a dividend based of this transaction and when the transaction occurred respectively.

c. Post
    i. Every post has a Post ID associated to it to uniquely identify it and a Poster ID to link it to the user that posted it. This would allow us to retrieve the users information such as their name to associate to the post when it is being displayed in the forum
    ii. To represent the content of the post, each Post would have a Title field and Content field to represent the title and content of the post
    iii. Every post entry or post will have a list of replies to the content of the post which would be represented as subposts.
    iv. Additionally, every post would have a Date field to indicate when it was created.

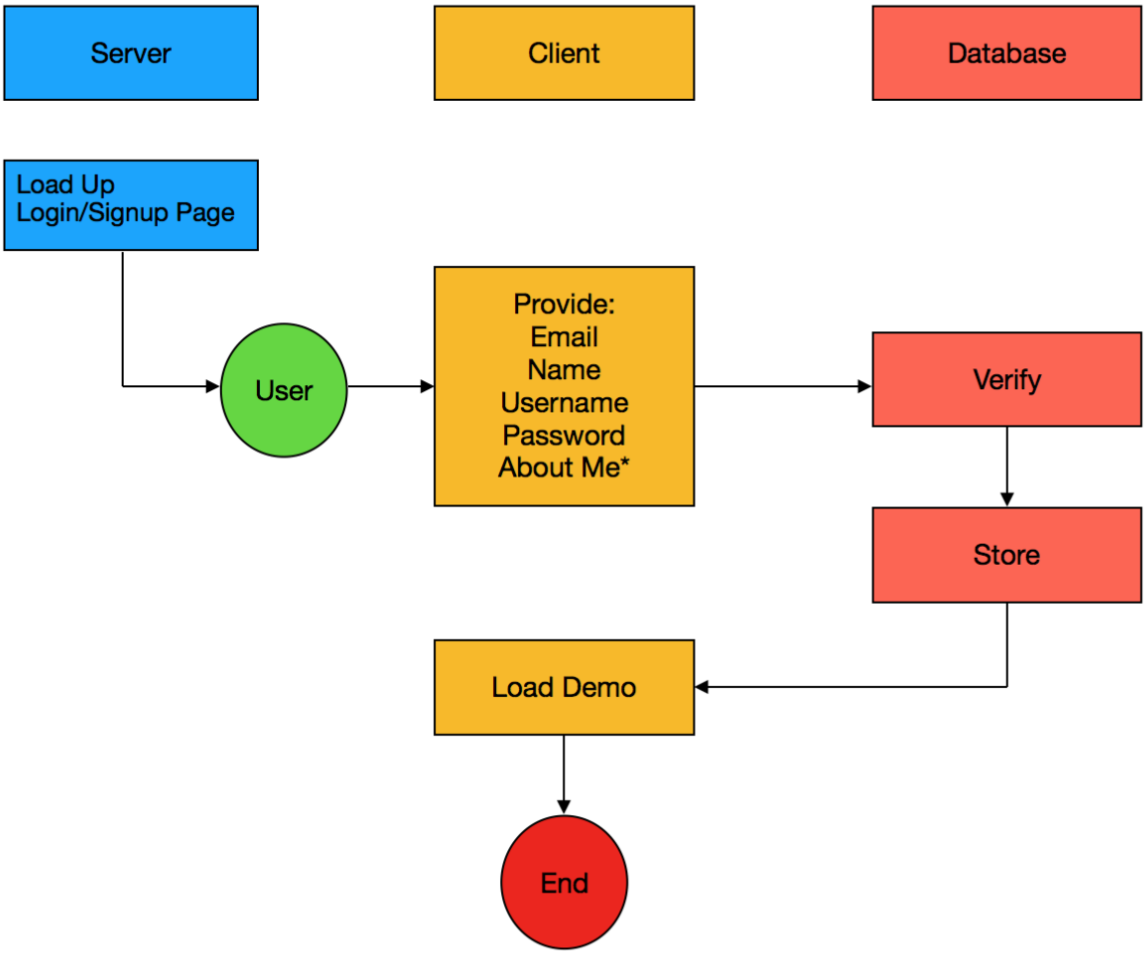
d. SubPost
    i. Sub posts are similar to posts but the lack some of the characteristics of a post such as a Title or a list of replies.
    ii. Every Sub post will have a SubPost ID to easily identify it and a field called Poster ID to associate it to the user that made the reply or sub post. This will allow us to retrieve user information such as the users name in the forum if required.
    iii. Finally, every sub post will some content, represented by the Content Field in the node, to represent the users reply to the original post.

e. Forum
    i. The Forum node represents the entire StockMock forum.
    ii. Just like any other forum the StockMock forum will have a series of posts which will be represented by the Posts field in the node as a list of Posts
    iii. Additionally, the forum will also have a list of members, the Member List Field, to represent all the members in the forum and provide the forum with a member count and other member information if needed.

Sequence of Events Diagram

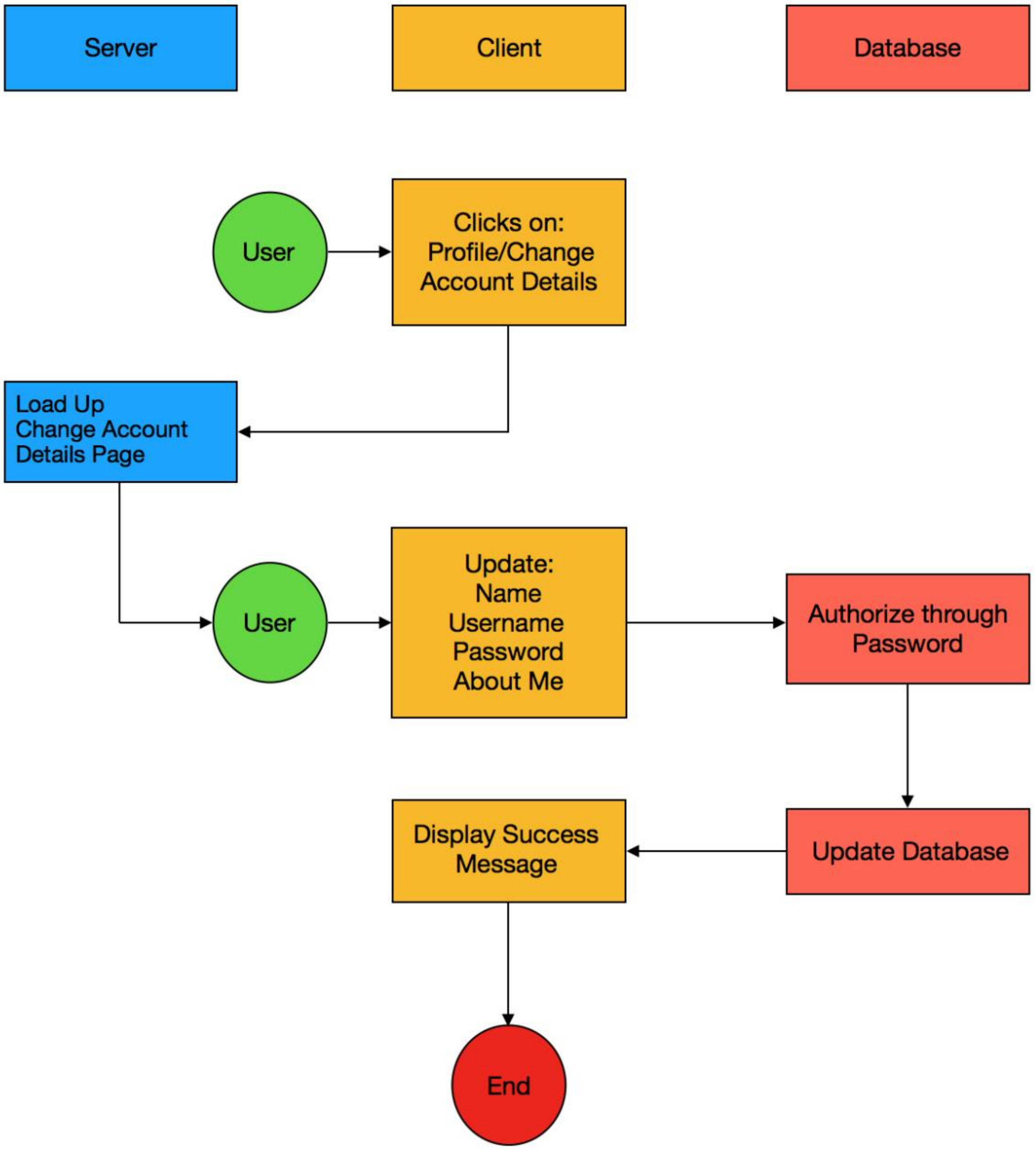
New User Creation:

A user registers a new account by filling in the email, name, username, password, and about me sections on the Login/Signup page. The data is then sent into the database where it is verified as a new account, then stored in the firebase database. The database then returns a success message which allows the client to log in to the newly created account and load a demo showing the full functionality of StockMock.


Server
Client
Database
Load Up Login/Signup Page
User
Provide: Email Name Username Password About Me*
Verify
Store
Load Demo
End

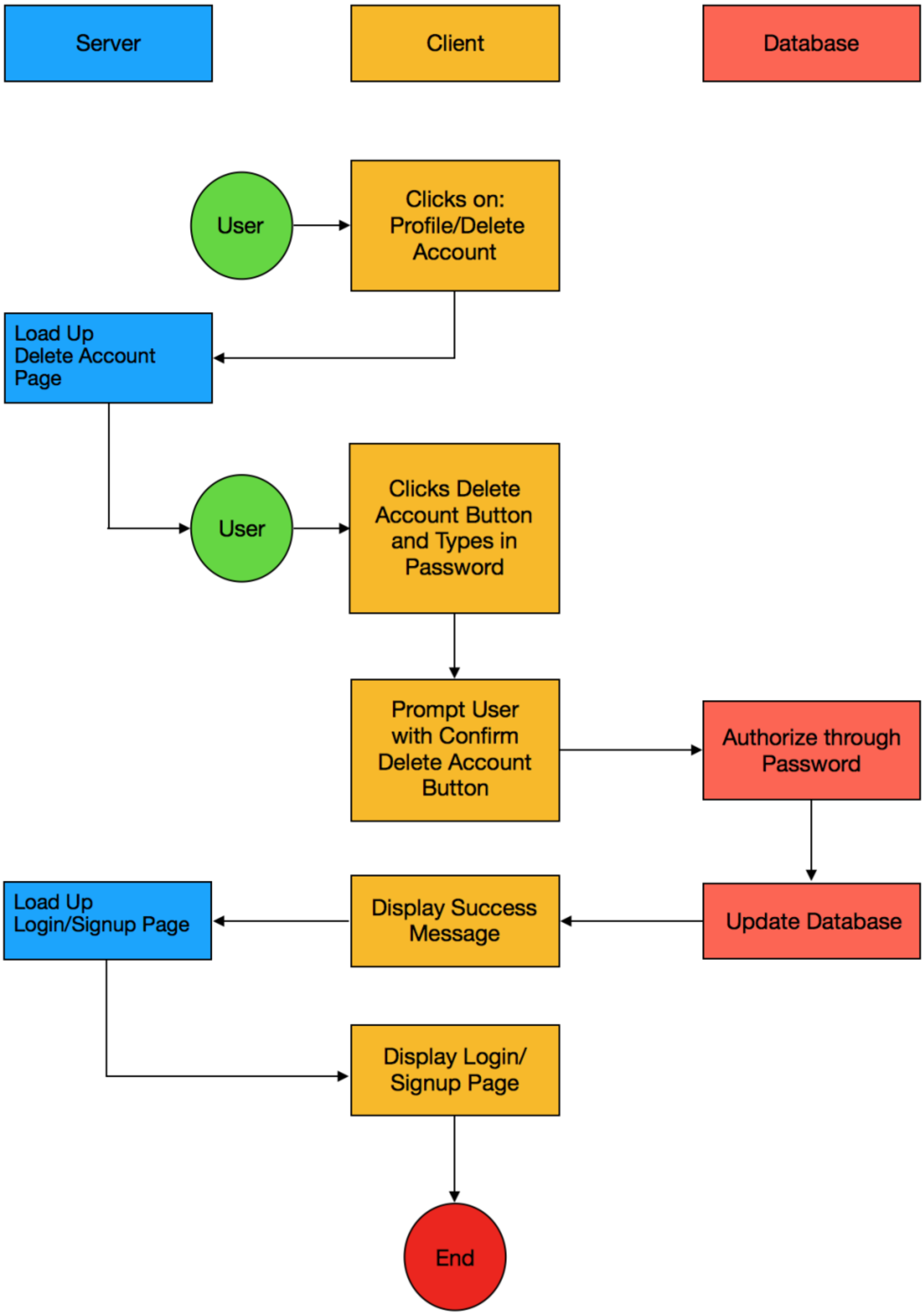
Change Account Details:

The user clicks the“Change Account Details” subtab under the profile tab on the menu bar that sends a request to the server to load up the Change Account Details Page. The user then updates any information they would like, except for the Email, which is immutable data, and then clicks the submit button. They will then be prompted to provide the password to authorize, which is then sent to the database. The database authorizes the request through the password and then updated. The database then sends a confirmation message to the user.


Server
Client
Database
User
Clicks on: Profile/Change Account Details
Load Up Change Account Details Page
User
Update: Name Username Password About Me
Authorize through Password
Update Database
Display Success Message
End

Delete Account:

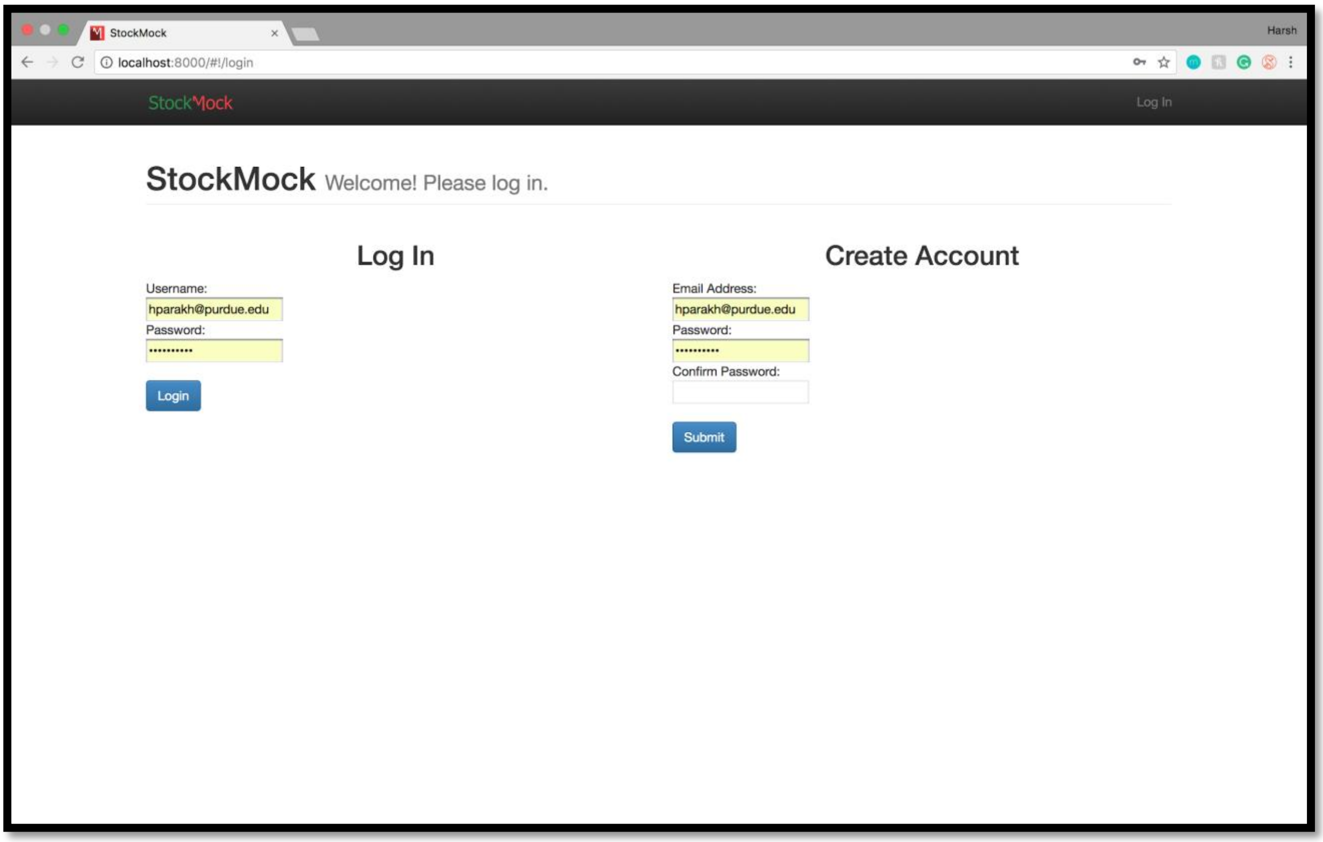
The user clicks the “Delete Account” subtab under the profile tab on the menu bar that sends a request to the server to load up the Delete Account Page. The user then confirms that they would like to delete their account by typing in their password and clicking delete. StockMock then shows a confirmation box that prompts then again to confirm, then sends the request. The Databases authorizes the request through the password, then updates the database, removing their information, sending back a success message. The Client then displays the message and sends a request to the server to display the login/signup page, which then sends the files and is loaded by the client.


Server
Client
Database
User
Clicks on: Profile/Delete Account
Load Up Delete Account Page
User
Clicks Delete Account Button and Types in Password
Prompt User with Confirm Delete Account Button
Authorize through Password
Update Database
Display Success Message
Load Up Login/Signup Page
Display Login/ Signup Page
End

UI Mock-ups

Login Page

This screen will appear when you first try to use the application. It allows users to create an account or login to an existing account. For security reasons, if a user gets either the username or password incorrect, we do not tell them which went wrong. If you tell an attacker the email address is wrong, they'll try a different one. If you tell them the password is wrong, then an attacker knows that the username is correct, and can go on to try a bunch of passwords for that username until they hit the right one. Just to consider this measure, our site won't tell you which one is wrong, to try and avoid the information disclosure.



StockMock Welcome! Please log in.

Log In

Username: hparakh@purdue.edu

Password: ••••••••••

Create Account

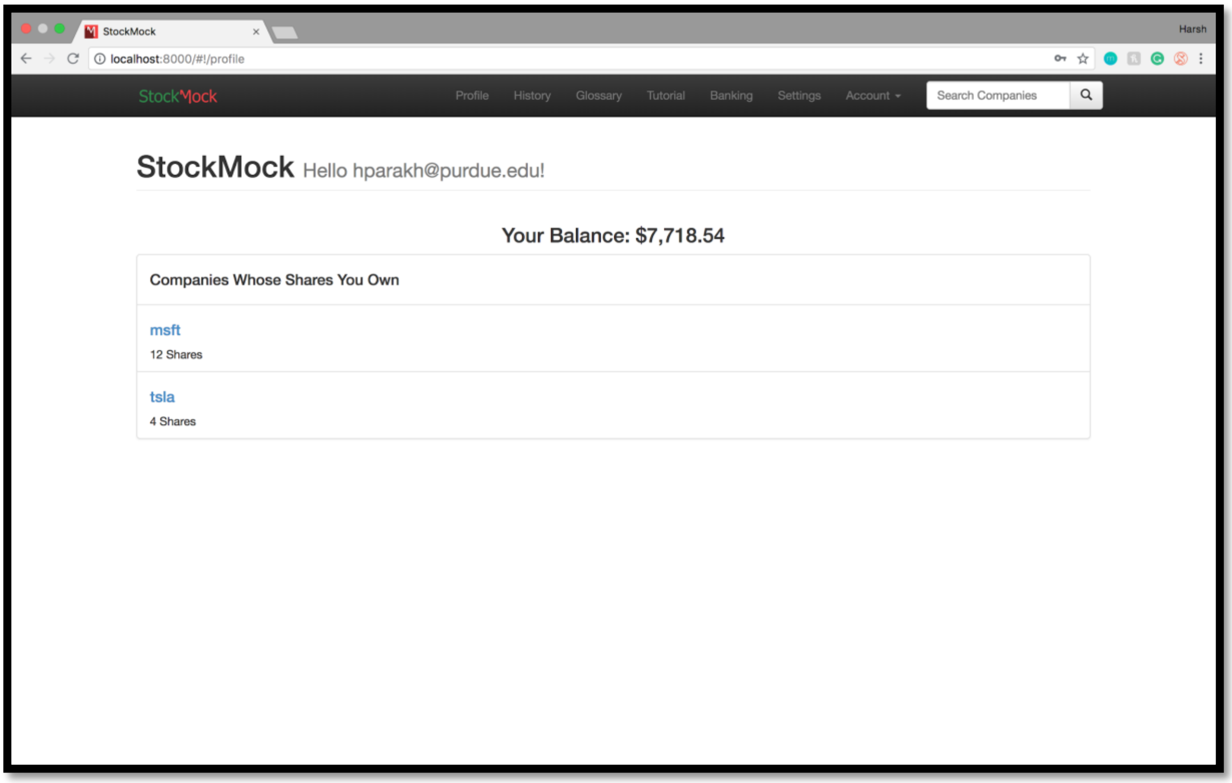
Email Address: hparakh@purdue.edu

Password: ••••••••••

Confirm Password:

Main Screen / Homepage

This is the main screen the user will see when not engaging in any of the additional activities that our application provides. This screen shows a general overview of important stocks information on the users account. It also contains links to other activities within our application. There is also a tab bar that allows the user to move about our site.



StockMock Hello hparakh@purdue.edu!

Your Balance: $7,718.54

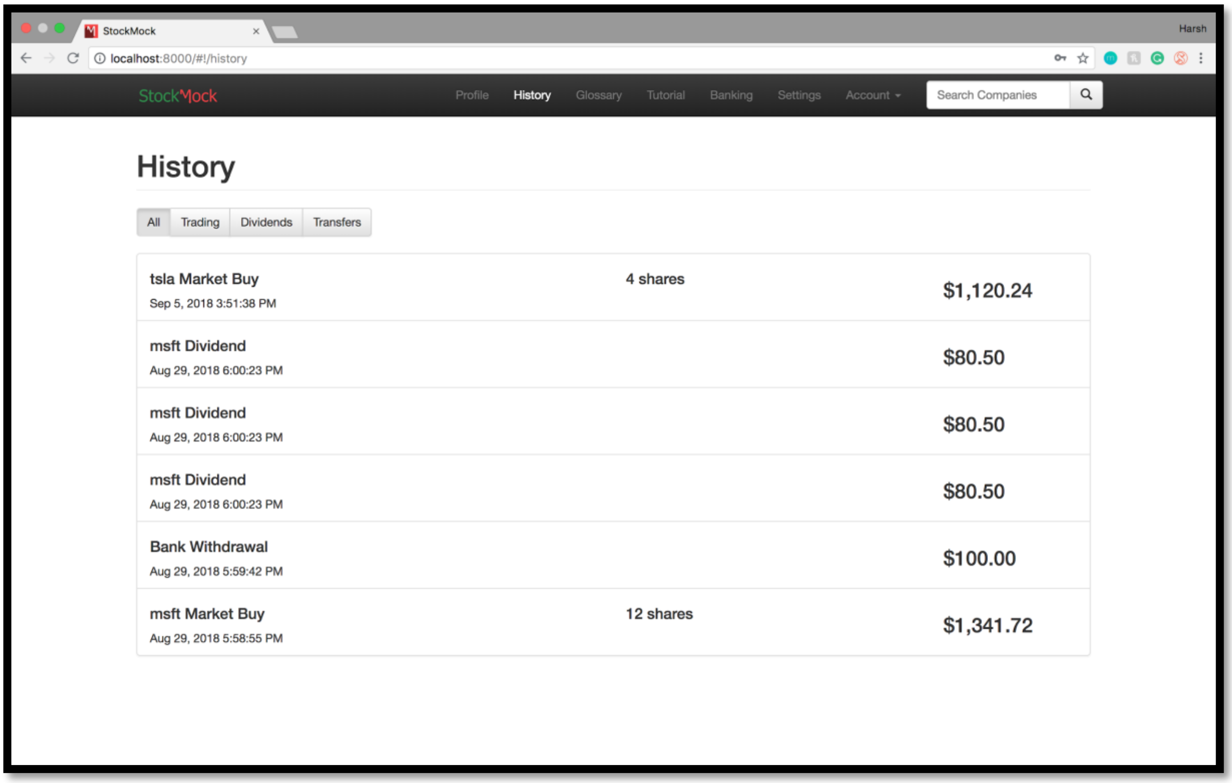
Companies Whose Shares You Own

msft
12 Shares

tsla
4 Shares

History

This page contains a history of stocks that have bought or sold by the user since they first created an account. The user is able to filter between different views of their history. There is also a tab bar that allows the user to move about our site. The trading filer shows buy and sell transactions whereas the dividend tab shows if the company has distributed money to its shareholders. Lastly, the transfers tab shows if you added or withdrawn money from StockMock.

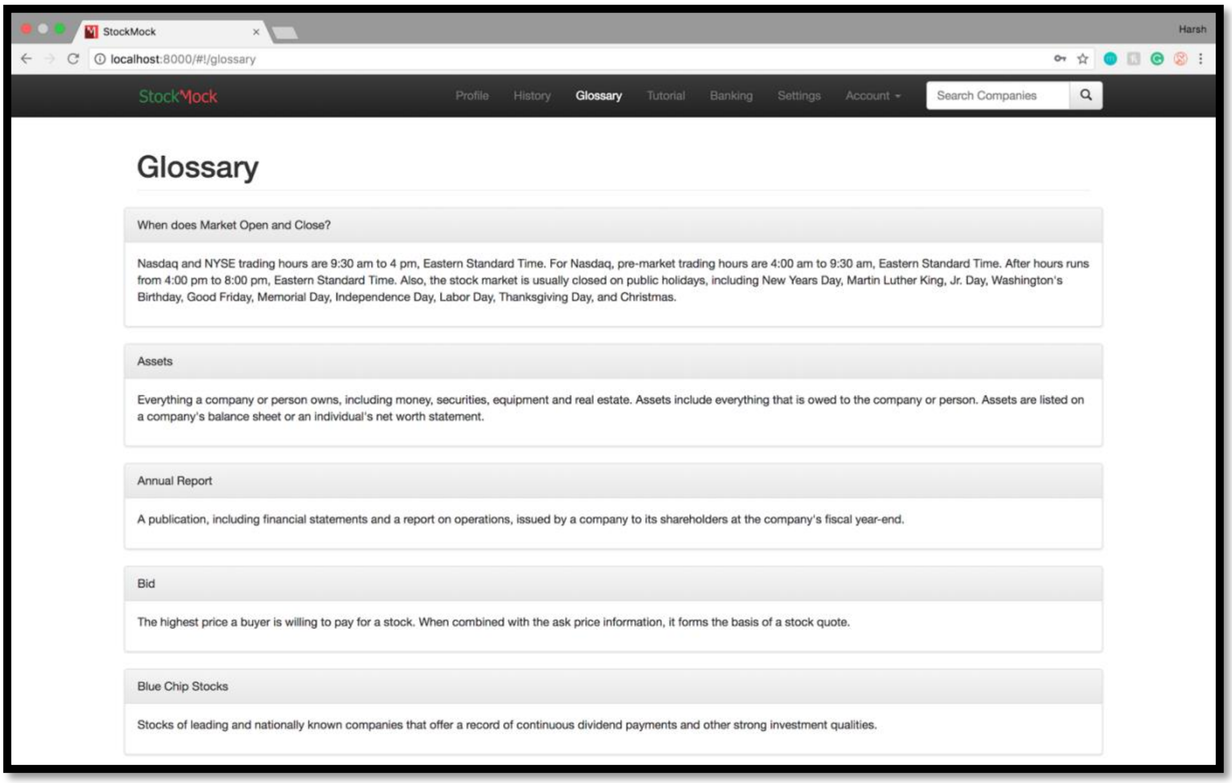


History

tsla Market Buy Sep 5, 2018 3:51:38 PM	4 shares	$1,120.24
msft Dividend Aug 29, 2018 6:00:23 PM		$80.50
msft Dividend Aug 29, 2018 6:00:23 PM		$80.50
msft Dividend Aug 29, 2018 6:00:23 PM		$80.50
Bank Withdrawal Aug 29, 2018 5:59:42 PM		$100.00
msft Market Buy Aug 29, 2018 5:58:55 PM	12 shares	$1,341.72

Glossary

This page contains a list of explanations related to stocks and financial information. This page is helpful for newcomers since they do not know a lot about the market and you need to know a few of these terms when trading.



Glossary

When does Market Open and Close?

Nasdaq and NYSE trading hours are 9:30 am to 4 pm, Eastern Standard Time. For Nasdaq, pre-market trading hours are 4:00 am to 9:30 am, Eastern Standard Time. After hours runs from 4:00 pm to 8:00 pm, Eastern Standard Time. Also, the stock market is usually closed on public holidays, including New Years Day, Martin Luther King, Jr. Day, Washington's Birthday, Good Friday, Memorial Day, Independence Day, Labor Day, Thanksgiving Day, and Christmas.

Assets

Everything a company or person owns, including money, securities, equipment and real estate. Assets include everything that is owed to the company or person. Assets are listed on a company's balance sheet or an individual's net worth statement.

Annual Report

A publication, including financial statements and a report on operations, issued by a company to its shareholders at the company's fiscal year-end.

Bid

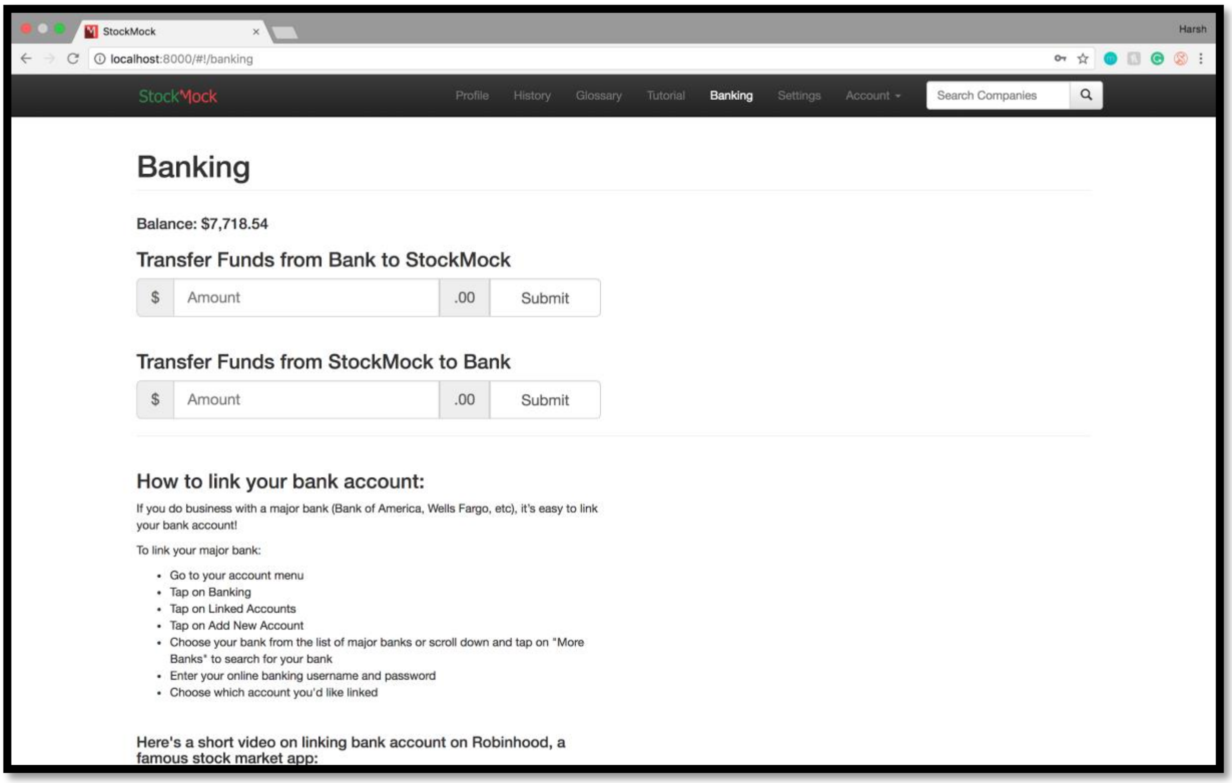
The highest price a buyer is willing to pay for a stock. When combined with the ask price information, it forms the basis of a stock quote.

Blue Chip Stocks

Stocks of leading and nationally known companies that offer a record of continuous dividend payments and other strong investment qualities.

Banking

This page allows users to transfer fake amounts of money to their account to allow them to continue purchasing stocks in the event they run out or would like to make a larger purchase. They are also able to remove funds from their account. The user does not link their bank account to our application, but we do explain how that process would be done in a real-world scenario. There is also a tab bar that allows the user to move about our site.



Banking

Balance: $7,718.54

Transfer Funds from Bank to StockMock

Transfer Funds from StockMock to Bank

How to link your bank account:

If you do business with a major bank (Bank of America, Wells Fargo, etc), it's easy to link your bank account!

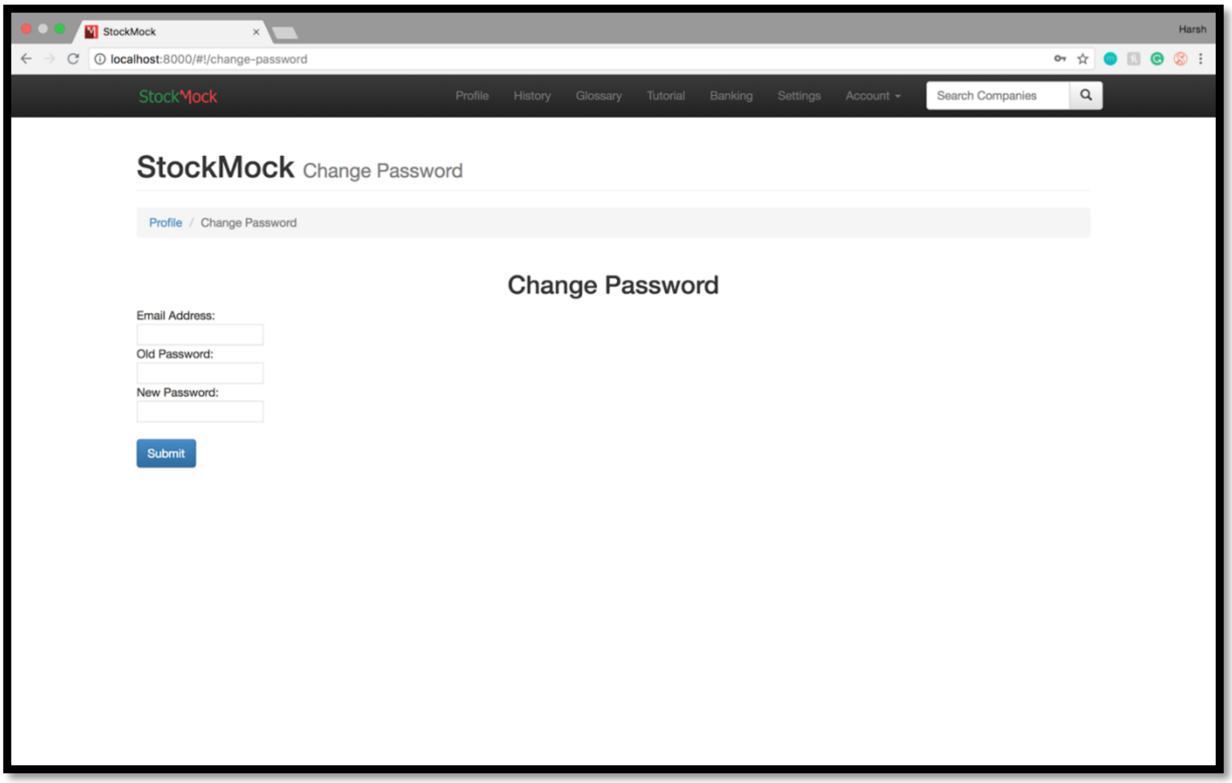
To link your major bank:

- Go to your account menu
- Tap on Banking
- Tap on Linked Accounts
- Tap on Add New Account
- Choose your bank from the list of major banks or scroll down and tap on "More Banks" to search for your bank
- Enter your online banking username and password
- Choose which account you'd like linked

Here's a short video on linking bank account on Robinhood, a famous stock market app:

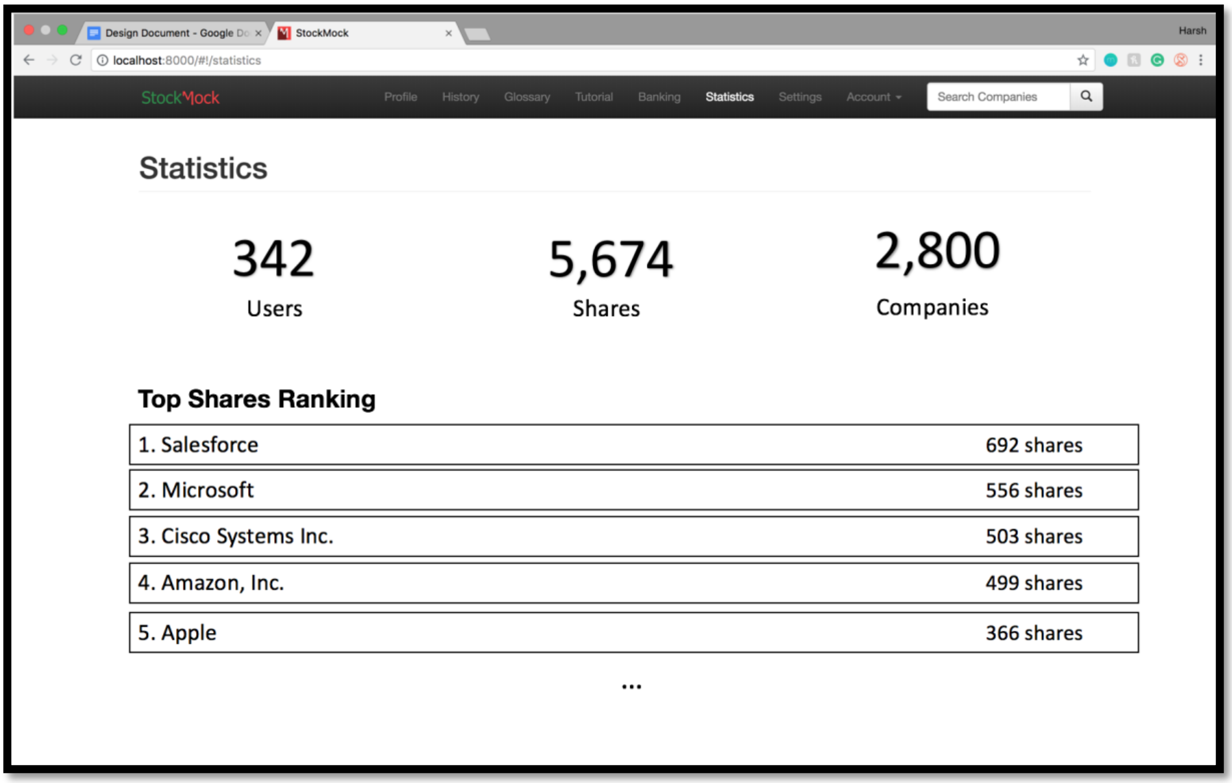
Change Password

This page is used by users who would like to update or change their password to a new one. The users must already be logged in to change their password. There is also a tab bar that allows the user to move about our site.



Statistics

This page shows the most top stocks purchased by users of our application and ranks them. It also provides some related informational stats such as number of users, total shares purchased and number of different companies users bought from within our site. There is also a tab bar that allows the user to move about our site.



Statistics

342
Users

5,674
Shares

2,800
Companies

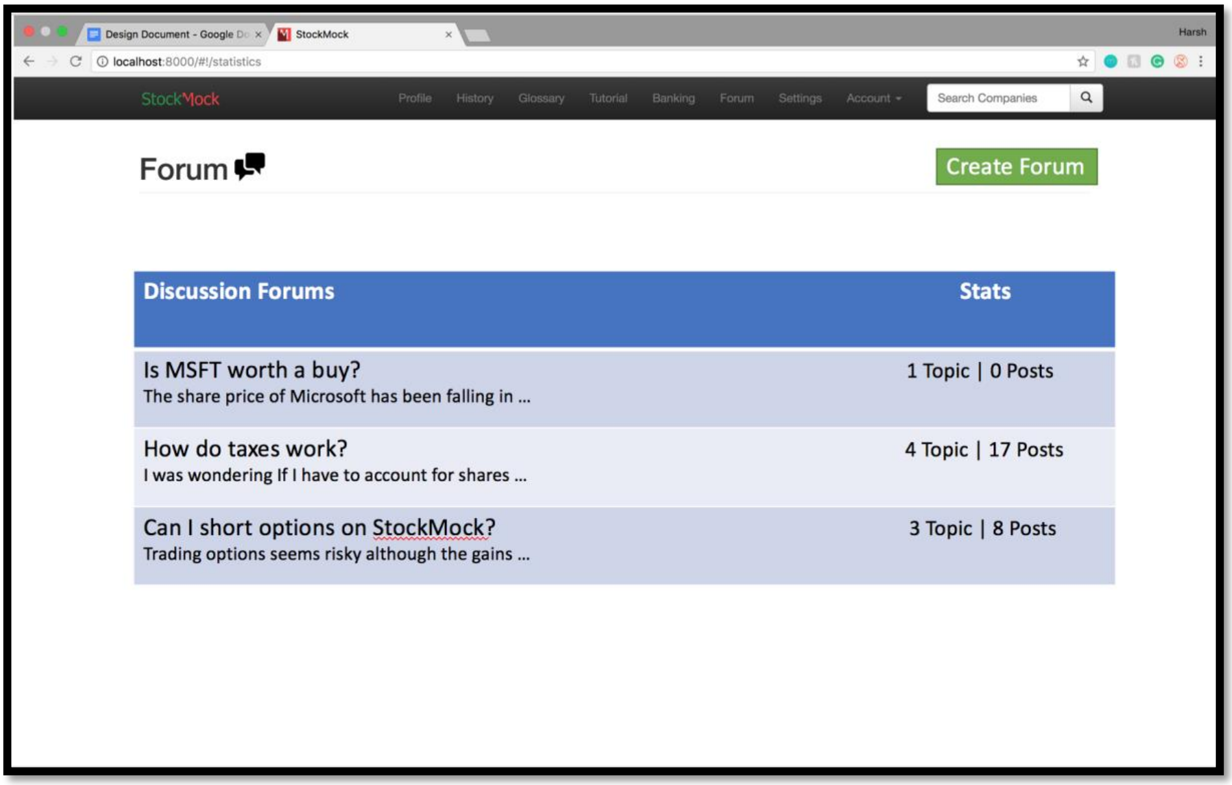
Top Shares Ranking

1. Salesforce	692 shares
2. Microsoft	556 shares
3. Cisco Systems Inc.	503 shares
4. Amazon, Inc.	499 shares
5. Apple	366 shares

...

Forum

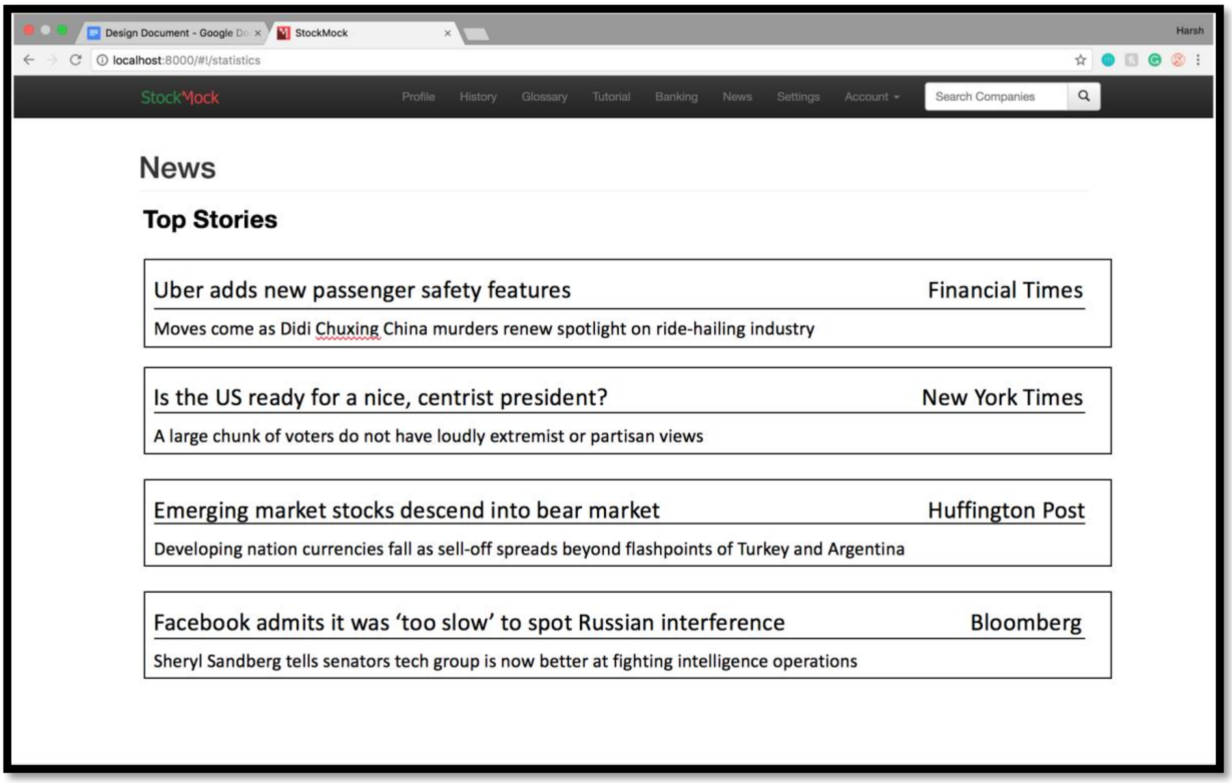
This screen is where the user is able to create forums or view forums created by other users. These discussions can include informational posts that users can read to further learn the ins and outs to stock trading. Users can also create discussions to ask a question that other users can then respond to and answer.



Discussion Forums	Stats
Is MSFT worth a buy? The share price of Microsoft has been falling in ...	1 Topic | 0 Posts
How do taxes work? I was wondering If I have to account for shares ...	4 Topic | 17 Posts
Can I short options on StockMock? Trading options seems risky although the gains ...	3 Topic | 8 Posts

News

This would be another new addition to the existing platform where we provide the users with daily news from different sources in regard to the financial world. The user can click on any link and get redirected to the original article and gain more information.



News

Top Stories

Uber adds new passenger safety features — Financial Times

Moves come as Didi Chuxing China murders renew spotlight on ride-hailing industry

Is the US ready for a nice, centrist president? — New York Times

A large chunk of voters do not have loudly extremist or partisan views

Emerging market stocks descend into bear market — Huffington Post

Developing nation currencies fall as sell-off spreads beyond flashpoints of Turkey and Argentina

Facebook admits it was 'too slow' to spot Russian interference — Bloomberg

Sheryl Sandberg tells senators tech group is now better at fighting intelligence operations