**Billionaire**

**Team 7**

**Members: Jingzhou Wang, Xiaowei Fu, Jay Cha, Rui Pang, Moshi Li**

**Contents**

1. **Purpose** 
   1. **Functional Requirements**
2. **Design Outline** 
   1. **High-Level Overview of the System**
   2. **Flow of Event**
3. **Design Issues** 
   1. **Functional Issues**
   2. **Non-Functional Issues**
4. **Design Detail** 
   1. **Database Class-Level Design / Description**
   2. **General Sequence of Events**
   3. **Search Activity**
   4. **Discussion Activity**
   5. **Study Note Activity**
   6. **User-Profile Activity**
   7. **User-Interface Mockups**

**Purpose**

Billionaire is a program or application that attempts to reproduce or duplicate some or all features of a live stock market on a computer so that a player may practice trading stocks without financial risk. Virtual stock trading is a simulated trading process in which would-be investors can 'practice' investing without committing real money.

This is done by the manipulation of imaginary money and investment positions that behave in a manner similar to the real markets. Before the widespread use of online trading for the general public, paper trading was considered too difficult by many new investors. Now that computers do most of the calculations, new investors can practice making (or losing) fortunes time and time again before actually committing financially. Investors also use paper trading to test new and different investment strategies. Stock market games are often used for educational purposes.

**In this platform, we are going to include following terms:**

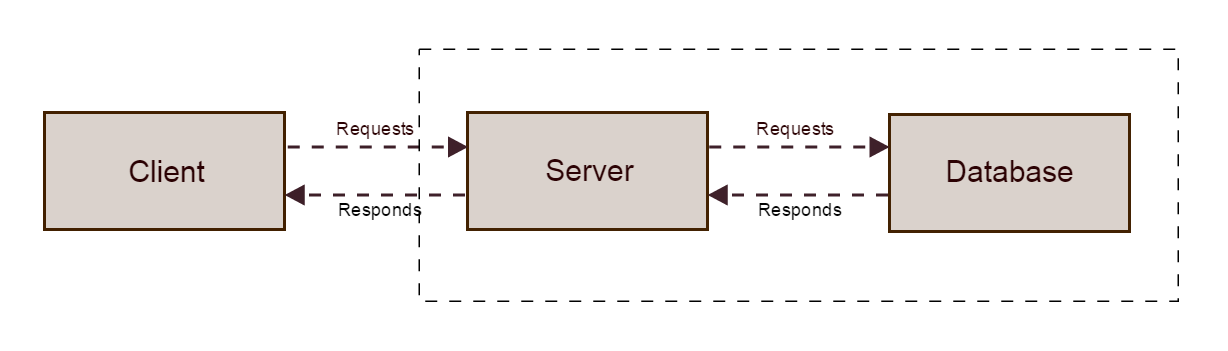
**Functional Requirements:**

1. **Profile Page**
   1. As a user, I would like to create a new account.
   2. As a user, I would like to reset my password via email if I do not remember the original one.
   3. As a user, I would like to modify my password.
   4. As a user, I would like to get free virtual cash once I create an account.
   5. As a user, I would like to gain some free virtual cash (wage) for logging onto the page consecutively for a week.
   6. As a user, I would like to view the total coins that I have.
   7. (If time allows) As a user, I would like to log in with my facebook or any other social media account without creating a new account.
2. **Stock Page**
   1. As a user, I would like to search the stock
   2. As a user, I would like to add my search stock into my watchlist after I login
   3. As a user, I would like to see the stock chart pattern for the stock that I searched
   4. As a user, I would like to check out the related news articles for the stock that I searched
   5. As a user, I would like to jump to the Game trading page to trade my search stock
3. **Game Page**
   1. As a user, I would like to see my total game capital for the game.
   2. As a user, I would like to see the stocks I have bought and how much they worth.
   3. As a user, I would like to search stock information in the game page
   4. As a user, I would like to see the most active stocks by volume list on my game page.
   5. As a user, I would like to see a recent trading history
   6. As a user, I would like to see what stocks I have in my watchlist
   7. As a user, I would like to go directly to the stock page from the watchlist
   8. As a user, I would like to remove the stocks from the watchlist
   9. As a user, I would like to set values in the notification setting for the stocks in my watchlist
   10. As a user, I would like to receive emails if the price of the stocks in my watchlist drops or goes up
   11. As a user, I would like to see my rank among players. (If time allowed)
4. **Trading Page**
   1. As a user, I would like to see all of my trading history
   2. As a user, I would like to place an order on the stock
   3. As a user, I would like to see my account value in the game trading page
   4. As a user, I would like to preview the order I placed
   5. As a user, I would like to know if the order I placed failed or not
5. **Main Page**
   1. As a user, I would like to search a specific stock on the main page.
   2. As a user, I would like to see rolling news summary in current stock market
   3. As a user, I would like to login my account from the main page.
   4. As a user, I would like to logout from my account
   5. As a user, I would like to see pictures of rolling news
   6. As a user, I would like to login with facebook from the main page
6. **Tutorial Page**
   1. As a user, I would like to study basic of stock market
   2. As a user, I would like to know how to buy stock and sell stock
   3. As a user, I would like to see basic stock concepts and strategies
   4. As a user, I would like to know how to play the game
7. **Miscellaneous**
   1. As a user, I would like to know who made the website.
   2. As a user, I would like to fill in the feedback form to developers.
   3. As a user, I would like to know the passing percentage of each course for last semester.
   4. As a user, I would like to see the introduction about the website main purpose on the front page.

**Design Outline**

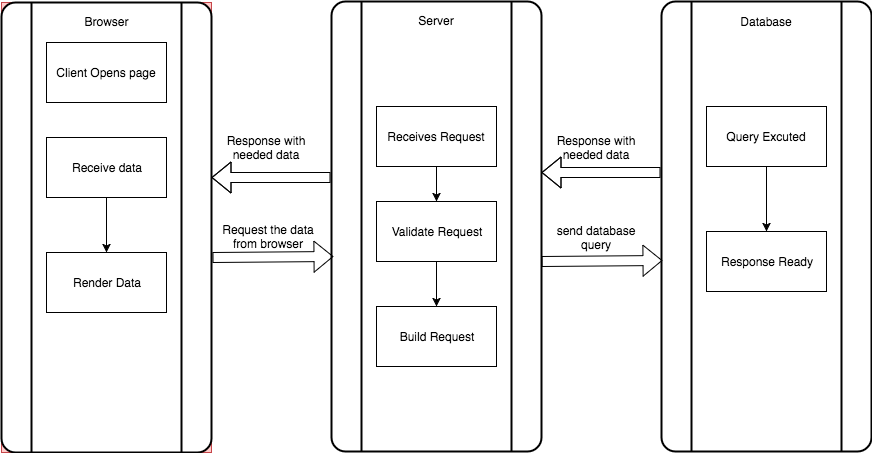
**High-Level Overview of the System**

Client-server model will be used for our project. Server will response to the request made by client and the information sent by client. Database will process the request from the server by providing the information server needs, which are requested by the client. Database will respond to server’s request and record the information sent by the server.



1. Client
   1. Response-data and given-data from database are processed by NodeJS
   2. Client will receive response from the server by AJAX
   3. Response-data and given-data from database are processed by NodeJS
2. Server
   1. Processes all of the data and requests which are sent by users
   2. Checks the correctness of the request and format of data before processing
   3. Generates user’s requested data and information by accessing the database
   4. Regularly requesting the course updated information from the alphavantage API
3. Database
   1. Stock information will be recorded in the database.
   2. All of the user information such as user profiles, their trade history, and stock profits will be recorded in the database.
   3. All of the stock resources are recorded to the database.

**Flow of events**

Our general idea of the events are described as above. Users will make their requests on the browser and the requested the data will be sent to the server by using AJAX that forms Json object. Server will response to the request by interpreting the Json data, and make request to the database, in order to get what client needs. Database will response to server’s request by processing the query and sent the requested information back to the server. At the end, server’s request and information also will be sent by AJAX from the server to the browser for the clients.

**Design Issues**

**Functional Issues**

1. Issue: Do users have to login to use our service, such as news, games, and tutorial?

Option 1: Users have to create an account that is unique to our website to use all the function.

Option 2: Users can login using Facebook or any other social media account to use all the function and join the game.

**Option 3: Users have to create an account to join the stock game and watch the tutorial but they do not have to create an account to use the other services such as search an stock and read stock news, etc**

Decision: We decide that users have to create an account to join the stock game and view the tutorials, but they do not have to create an account to use the other services such as stock information, and market news etc. Since the purpose of our website is to assist stock beginners to have an virtual trading experiences at stock market, and teach them some trading strategies. Only if user has an account, we can record the transactions,and the profits and loss situations of different users.

2. Issue: How do users access to a specific stock?

Option 1: Navigate through menus

**Option 2: Use search bar**

Decision: We decide to use search bar only. Traversing through different menus is tedious to users while providing users with a search bar to search a specific stock is much more efficient. Since users may already have a targeted stock, there is no need to have separate menus for navigation.

3. Issue: Which part of the functions has the highest priority?

**Option 1: Stock Page**

Option 2: Trade Page

Option 3: Game page

Decision: We believe that stock page has the highest priority. Stock page is able to present the stock from various perspectives and mostly let users have a brief knowledge about the stock information. Besides, Stock provides jump button for Trade page and watch list page.

**Non-Functional Issues**

**1)** Do users need to create an account to get on our website?

* + Option 1: User must have an account to use our website.
  + Option 2: User does not need an account to use our website
  + **Option 3: Users do not need an account to use our website, but users need an account for functions other than searching stocks.**

We decided to go with option 3, because option 1 is unlikely to attract users and option 2 is unlikely to maintain consistent users.

1. What should we use for our database?
   * Option 1: We use a non relational database such as mySQL
   * **Option 2: We use a non relational database such as MongoDB**

It was a hard decision for us, since half of the members are more familiar with MongoDB, and the other half are more familiar with mySQL. In the end, we decide to use MongoDB, as we believe it is easier to get specific data using mongoDB.

1. What front end framework should we use?
   * **Option 1: Angular**
   * Option 2: vue.js
   * Option 3: Ember.js

We decided to use Angular as a front end for multiple reasons. It has client-side MVC and creates a single page application that is responsive. It has excellent data binding (only one with two way data binding which was the largest difference in our other options). It is built for easy testing as it comes bundled with Protractor.

1. What back end framework(s) should we use?
   * Option 1: PHP
   * **Option 2: Node.js with Express.js**
   * Decision: We are using Node.js because it is extremely lightweight and efficient

Express is linked tightly to Node as well, and we will be using that as a “fast, unopinionated, minimalist” web framework with Node. Express abstracts out some low level logic that can could cause unnecessary problems if we went without. Express also helps with file organization for a MVC structure and we will use it for routing. Using Node.js would also allow us to code in JavaScript on the client and server; we wouldn’t have to switch gears. Node.js, however, when compared to PHP, does not integrate with as many older technologies, which is not an issue for us because we don’t plan to be working with any obscure API’s.

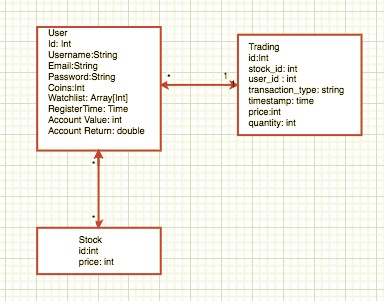
5) Which method of authentication should we use

* Option 1: Cookie based authentication
* Option 2: Token based authentication
* **Option 3: Passport.js**

We chose to use passport.js, a node.js middleware, because it is easiest to implement, since we are already using Express. The other two options have a steeper learning curve and passport would allow us to easily add multiple user login options (Facebook, Twitter, Google, etc)

**Design Details**

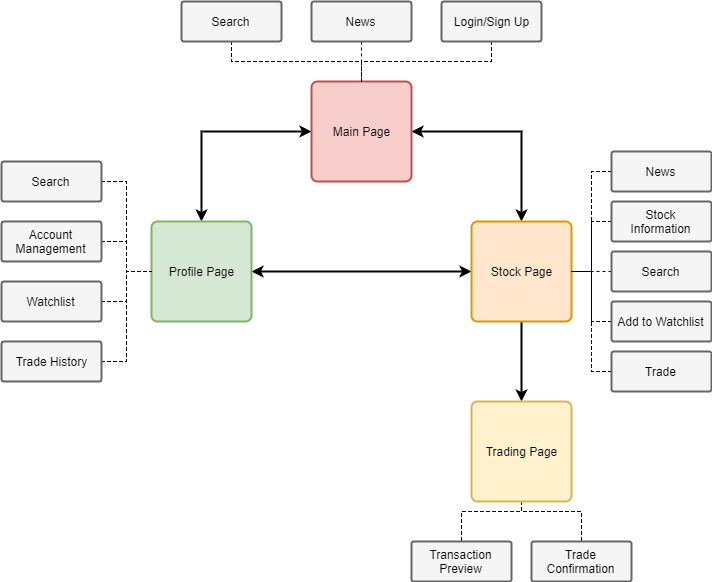
**Database Class Level-Design**

****

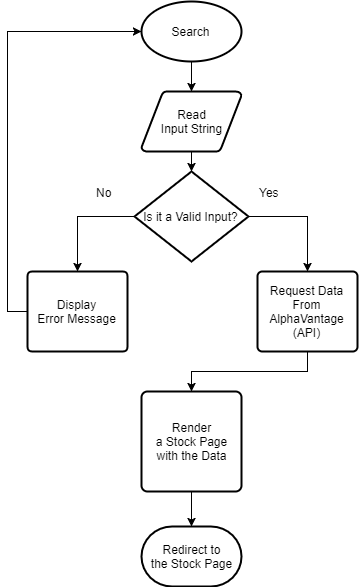
We will be using 2 tables in our database

* **User** 
  + Every user is in this class. The subjects are including students,and adminitives.
  + This class contains attributes of users such as id (username), email, account details, register Time, and their account return.
  + Users can add stock to their watchlist, trading with their money, and watch various tutorial.
* **Stock** 
  + Each stock page has specific stock name and related news about this stock.
  + Stock information will be provided by the stock class, such as passing price, stock name and price history.
  + Stock can also be added to the watchlist and can be trade by the user after user search the stock.
* **Trading** 
  + Each trading has specific stock information and user information.
  + Trading table will contains user\_ id , stock\_ id, trading amount, trading time, price and trading type.
  + User can view their trading history by time, price amount, and type.

**General Sequence of Event**

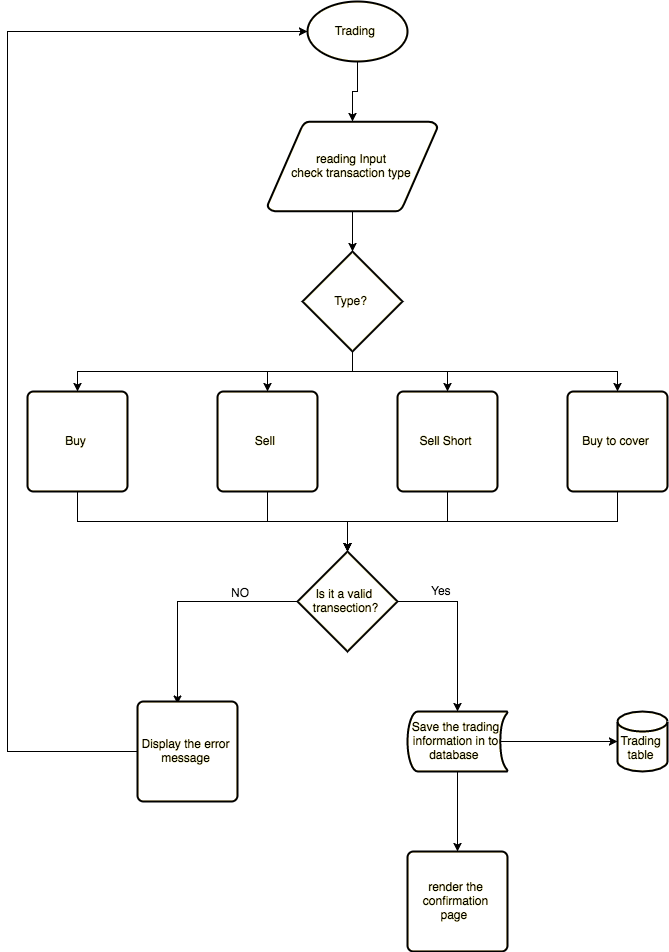
****User first accesses the website and will see headline news articles and search bar for the stock that the user wants to search. If the user clicks the news article, the user will be redirected to the corresponding news article website. From the main page, the user can either sign up and log in to go to the profile page or type in the stock symbol or the name of the company to get on the stock page. On the stock page, the user can click trade button to be redirected to the trading page, and make a transaction on the page.

**Search Flow Map**



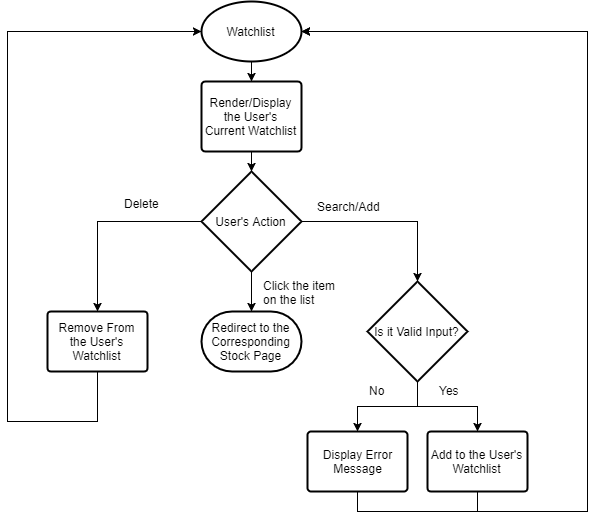
When the user specify either a stock symbol or a company name, the search engine will access our database for the stock id. It will then decide whether the entered string is valid or invalid. If the string is invalid, it will display error message and the user will need to search again. If the string is valid, the search engine will query the related stock data and render the stock page and then the user will be taken to the corresponding stock page.

**Trading Flow map**

****

When user enter their trading page and type in their trading information such as stock symbol, transaction type, quantities, market price, price limit ex. After user entered all the information and choose the trading type. The system will check if the transaction is valid based theri account information. If the transaction is valid, the confirmation page will be render and the trading transaction information will be saved into database. If the transaction is not valid, error message will be pop out and render back to trading page.

**Watchlist Flow Map**

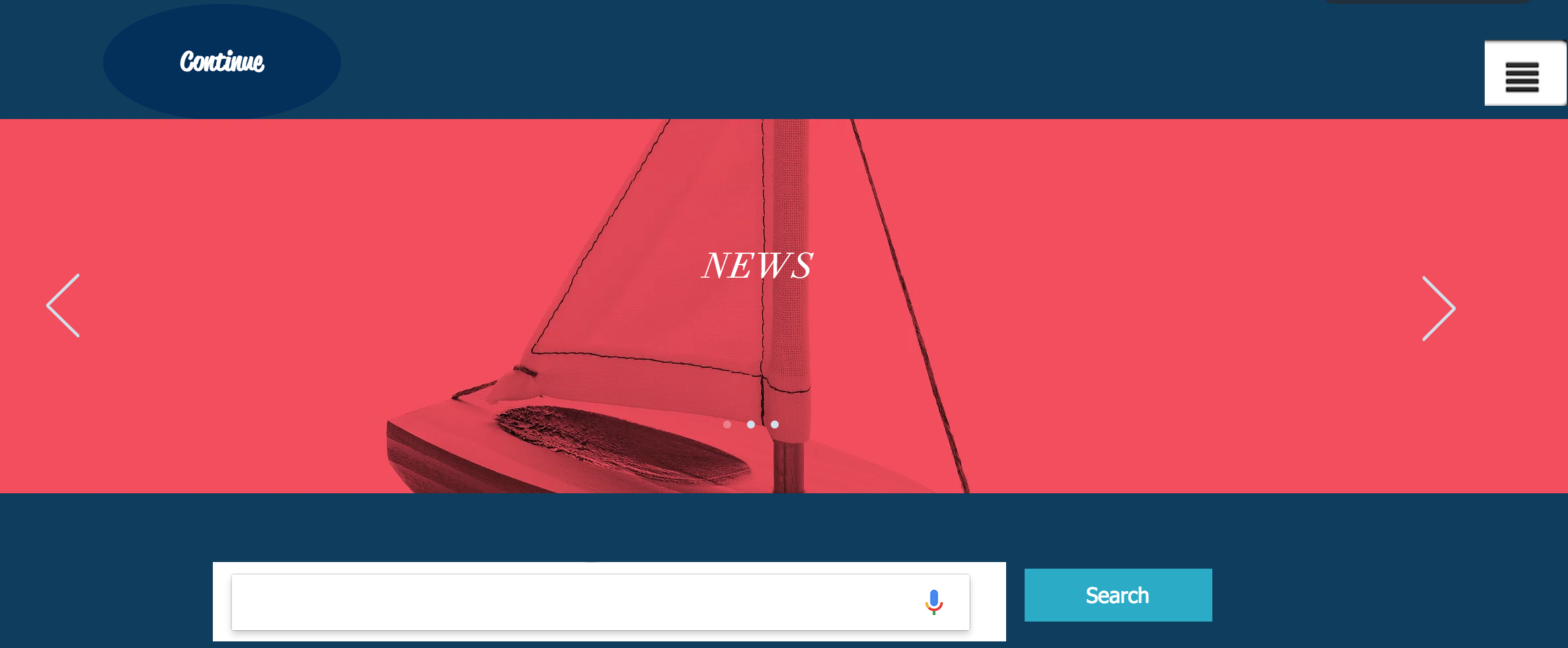
****

When the user accesses the watchlist, the page will first be rendered and will display the user’s current watchlist. On this page, the user can search/add the stock to his or her watchlist, delete the stock from the watchlist, or click the item to navigate to the corresponding stock page.

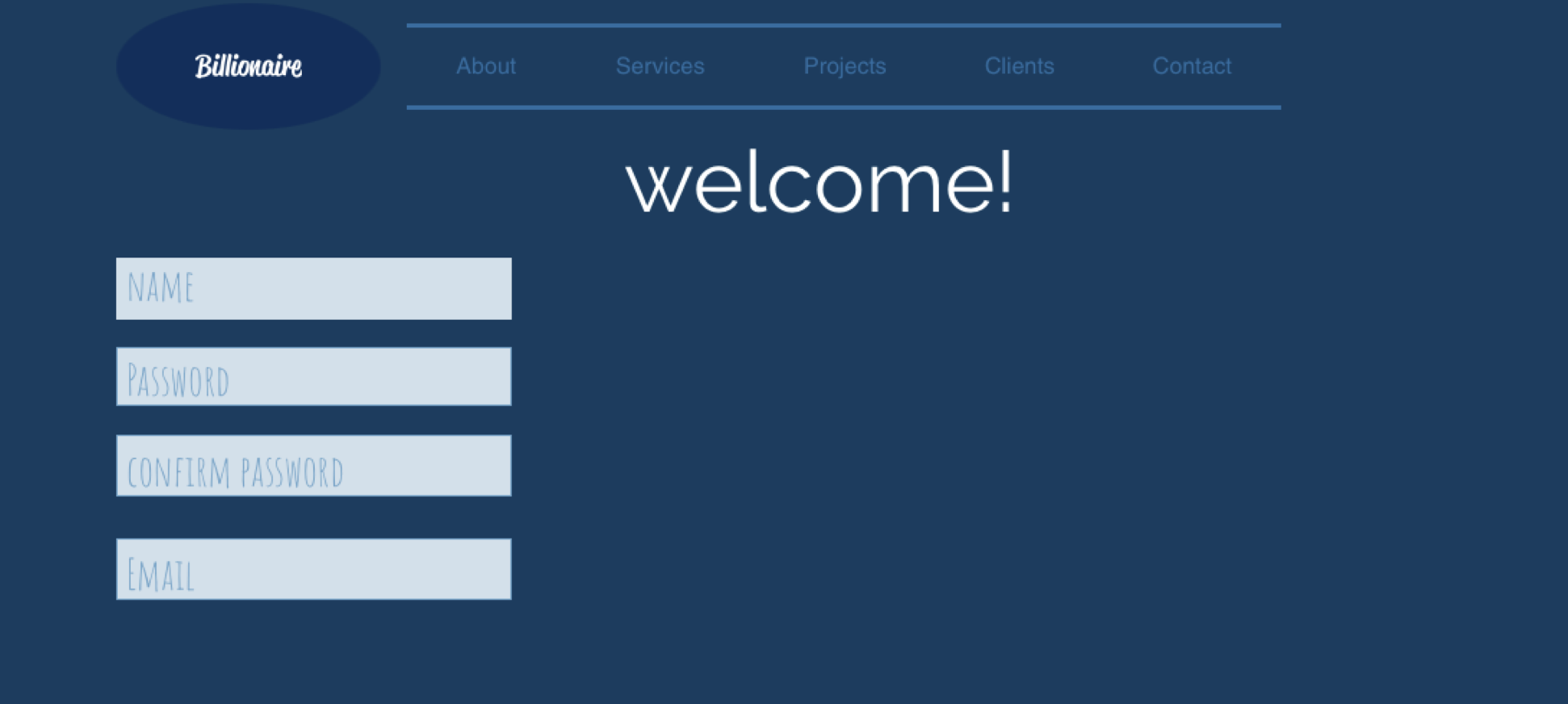
**User-Interface Mock-ups**

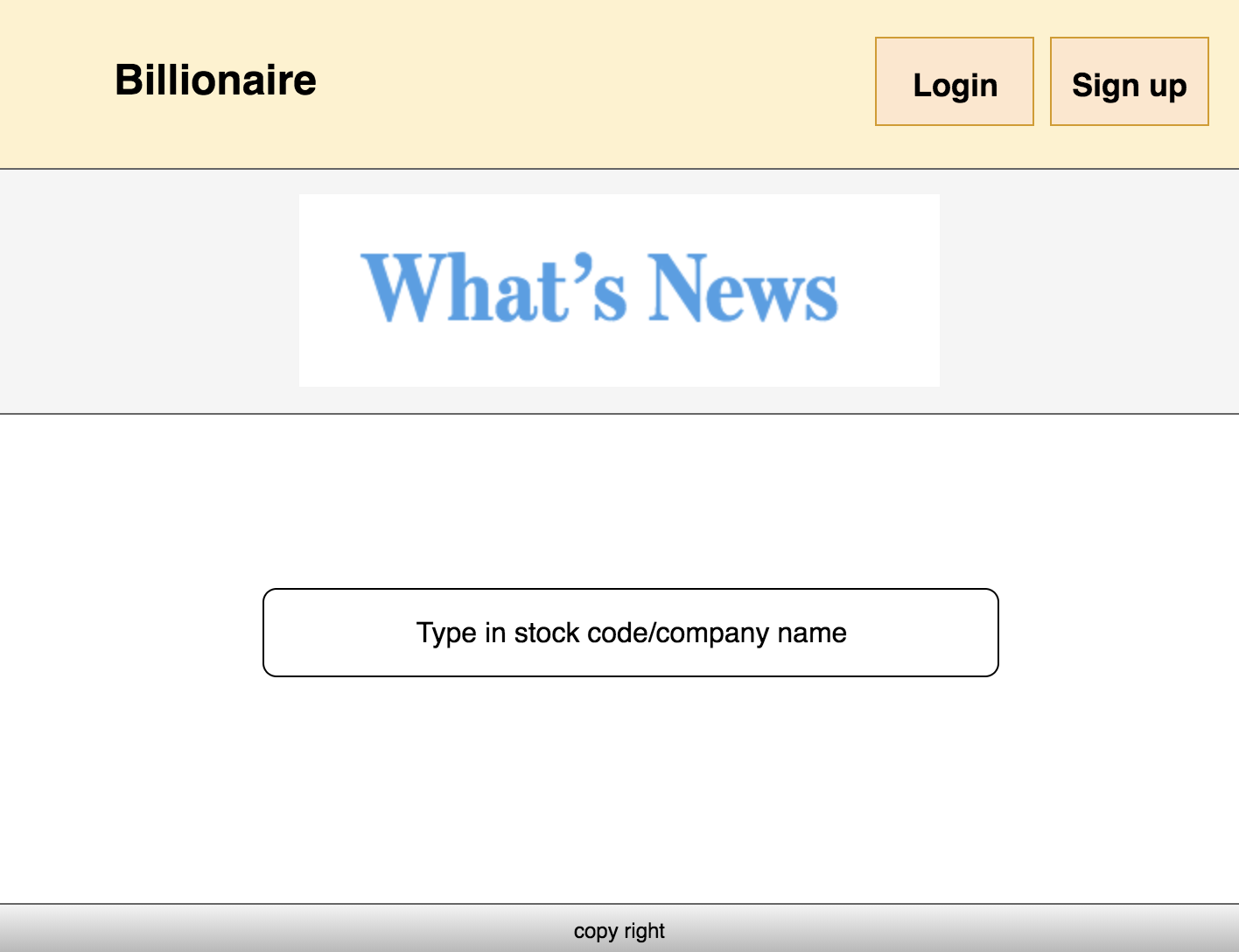
The main purpose of the UI design is to let user search for the intended course from various places in the website. Our top navigation bar will have the search function implemented. Account management, such as login and signup, will be accessible from the navigation bar as well. Our intention of designing the stock page is not to bombard the user with the confusing numbers, but to provide them with the core information that is ordered with sophisticated frontend.

**Homepage：**

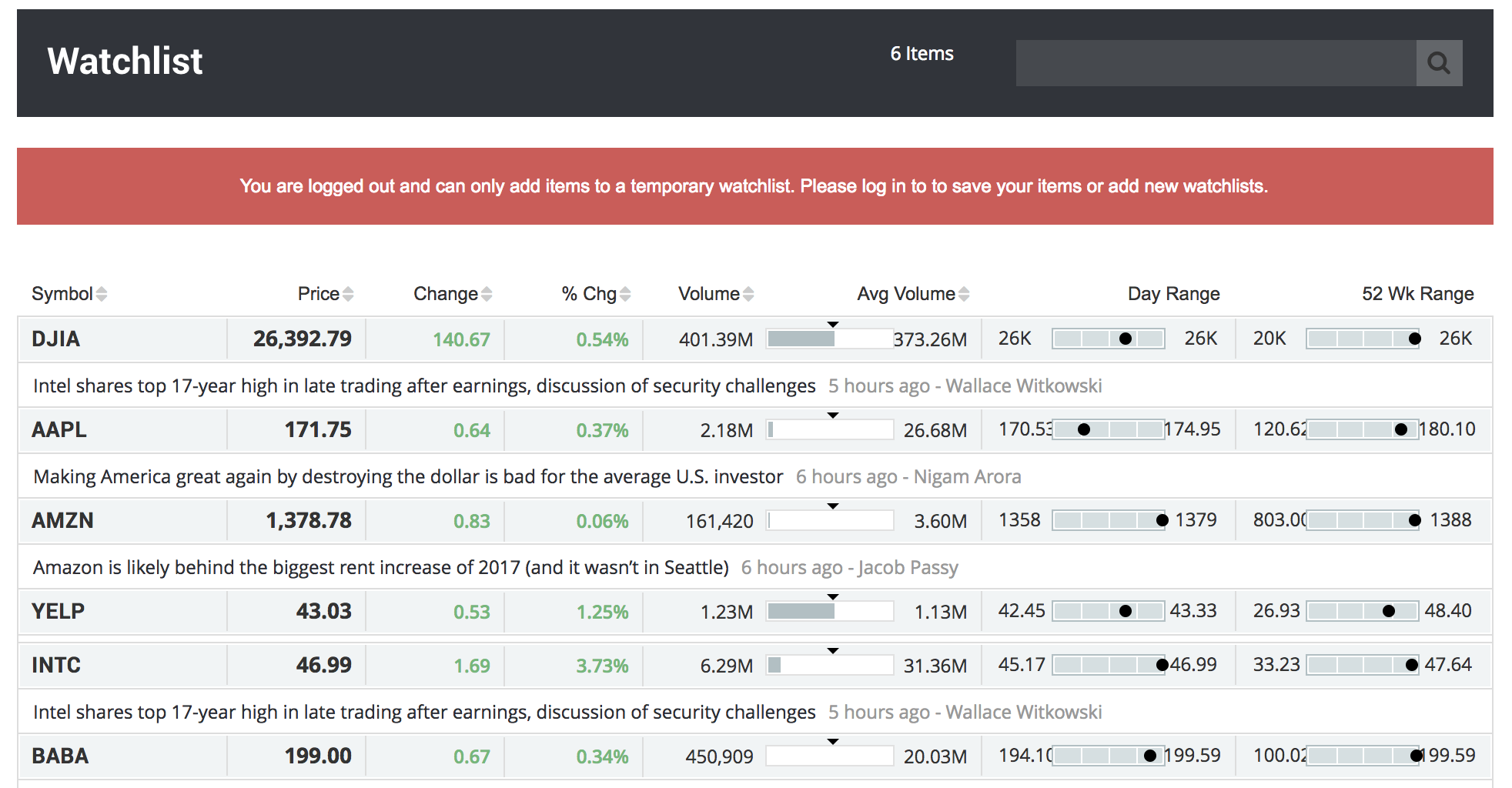
****

**SignUp:**

****



**Watchlist:**



**Stock page:**



**Trading Page: (Account Detail)**

