FALL
2017

Requirements Analysis Document

STOCK APPLICATION

DEREK BROWN, AYLSSA DROHAN, PHILLIP GIL-PEREA, JARRETT HORTON, ISAIAH JOHNSON, RUSSELL QUAO, ISAAC SILVIOUS

Table of Contents

Functional and Non-Functional Requirements ..... pg 4

Use Case Diagram..... pg 5

Use cases:

Register..... pg 6

Login..... pg 8

Start Application Use Case..... pg 10

Log Out..... pg 11

Reset Password.....pg 12

Calculate Price Alert..... pg 14

View Navigation Slider..... pg 16

View Portfolio Summary..... pg 17

Edit User Settings..... pg 19

Search for Stock..... pg 20

View Trending Stocks..... pg 21

View Infinite Scrolling Banner..... pg 23

Remove from Watchlist..... pg 24

Remove from Stocks Owned List..... pg 25

View Individual Stock Page..... pg 26

Add to Watchlist..... pg 28

Add to Stocks Owned List..... pg 29

Schedule Gantt Chart..... pg 31

Messages to User..... pg 32

Glossary..... pg 33

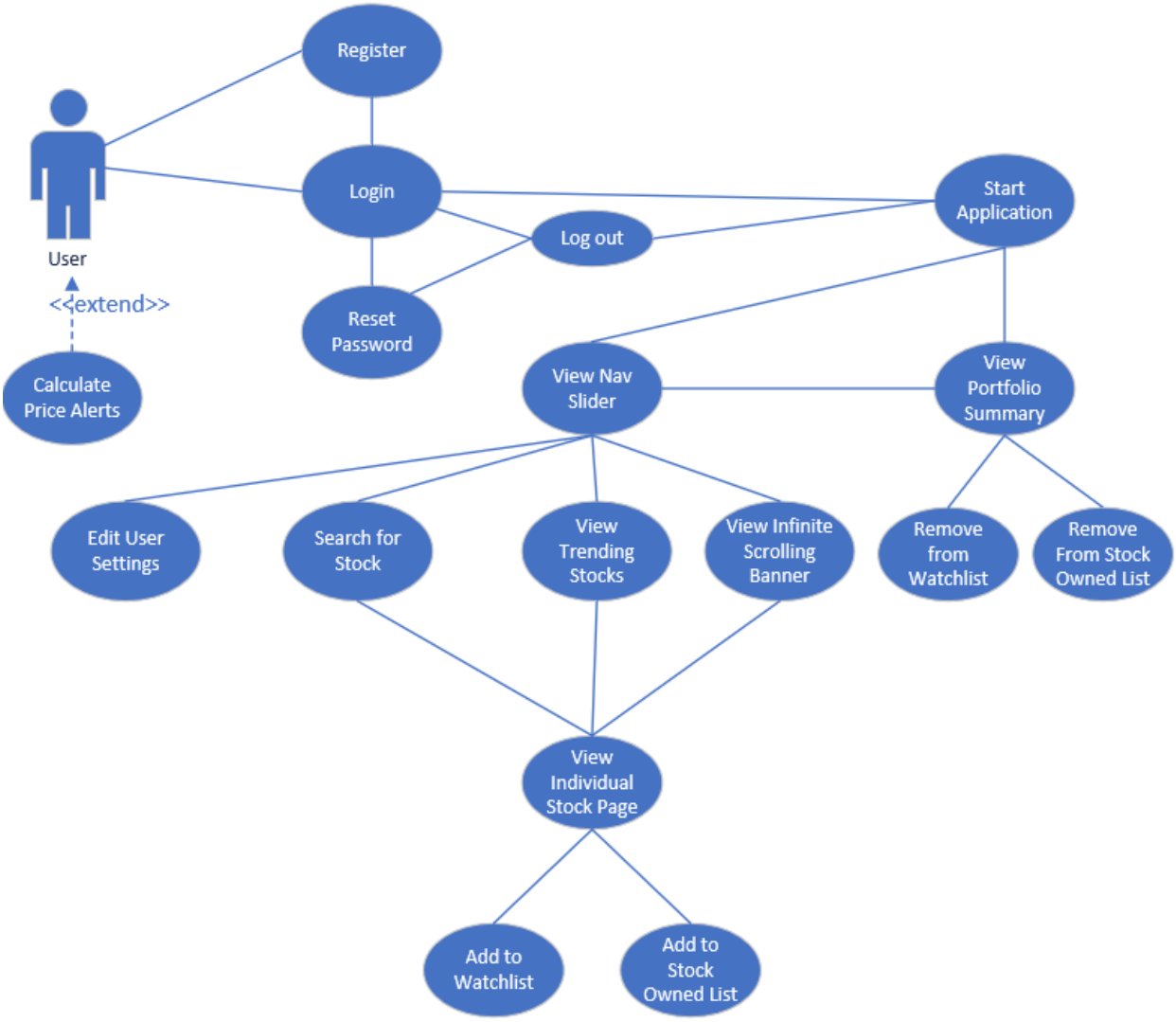
Functional Requirements

1. User can create a secure account with a user name and password.
2. User can Log in and Log out; user can choose to stay logged in through the easy to find logout button at the top of the Navigation Slider.
3. User can update their profile settings which include password, phone number, text alerts, and country of origin.
4. User can add and remove a stock to their watchlist or to their portfolio.
5. User can display calculate profit from bar charts and graphs about the specific stock.
6. User can display stock information for a specific stock.
7. User can search for a specific stock using the company stock symbol.

Non-Functional Requirements

1. The application uses the TextLocal api to send SMS alerts.
2. The application pulls the User's watch list, portfolio, and settings from the Firebase database.
3. The server-side web crawler will pull the trending stocks from websites every 15 mins via cron to upload to the Firebase database.
4. The application uses the IEX Trading API to pull stock information.
5. The application can calculate the profit of a given stock from the user's stock purchase date, purchase price, and quantity.
6. The application creates user information when registering in Firebase database including their password, email address, phone number and currency code.
7. Calculate the exchange rate based off the users chosen currency.

Stock Application Use Case Diagram



Use case: Register

Primary Actor: User

Goal in Context: Allows the user to register for an account.

Preconditions: Internet connection needed.

Trigger: The user creates an account by pressing the create account button on the login page.

Scenario:

1. User sees a form that includes: first name, last name, email, phone number, password, and confirmation password fields.
2. User enters data into all the specified fields.
3. User presses the enter button.
4. The system validates the email address and confirms that the passwords are the same.
5. The system validates the email has not been registered before by checking each firebase email address against the users specified email.
6. The system creates a new account on the firebase database with password, currency code, phone number and email.
7. The system displays Message 22 and redirects the user to the My Portfolio Summary screen (See My Portfolio Summary Use Case).

Exceptions:

1. User enters wrong confirmation password: Message 4 is displayed underneath the confirmation password field in red.
2. User enters an email that already exists: Message 5 is displayed underneath the email field in red.

3. User does not enter a valid email address: Message 3 is displayed underneath the email field.
4. User enters invalid password.: Message 2 is displayed underneath the email field.

Priority: High

When Available: Week 1

Frequency of Use: Many times per day

Open Issues: No open issues at this time.

Use case: Login

Primary Actor: User

Goal in Context: For the user to be able to log into the app at any time and be able to see their profile.

Preconditions: Internet connection needed.

Trigger: The user decides to log into his account, by clicking on the sign in button on the top right-hand screen.

Scenario:

1. User observes the login page which includes: email address and password fields.
2. User enters username and password correctly in the specified fields.
3. Application verifies credentials by sending that information to the firebase database which compares with all other credentials.
4. Message 23 is displayed on the screen and the user is transferred to the My Portfolio Summary screen (See My Portfolio Summary Use Case).

Exceptions:

1. Please enter valid credentials: Message 6 is displayed if user enters the wrong username or password into the Stock App and an error is displayed asking user to re-enter username or password. If the user gets the username or password wrong five times, they are asked to enter email to reset the password.
2. Email is invalid: Message 1 is displayed if the user enters an invalid email.

Priority: High

When Available: Week 2

Frequency of Use: Many times per day

Open Issues: No open issues at this time.

Use Case Name: Start Application use case

Primary Actor: User

Goal Context: The start application use case will determine if the user has already logged onto the website previously.

Preconditions: The user must be connected to the internet and have access to the firebase database

Trigger: The user enters the website's URL into a browser of their choice.

Scenario:

1. The application will check the browser for the user's credentials through the browser's cookies.
2. The application loads the View Navigation Slider. (See View Navigation Slider use case).
3. The application loads View Portfolio Summary. (See View Portfolio Summary use case).
4. The application loads Search for Stocks. (See Search for stocks use case).
5. The application loads Infinite Scrolling Banner. (See Infinite Scrolling Banner use case).

Exceptions:

1. There are no exceptions at this time.

Priority: High

When available: Week 3

Frequency of use: This feature will be used frequently, when the User loads a new page or logs in for the first time.

Open Issues: No open issues at this time.

Use case: Log out

Primary Actor: User

Goal in Context: Allows user to log out and takes them back to the login use case

Preconditions: Internet connection needed.

Trigger: User clicks the logout button to exit the application is located in the Navigation Slider.

Scenario:

1. User observes logout button on the View Navigation Slider use case.
2. The application will display a Confirmation box to the user. Message 18 will be displayed.
3. The user will select yes.
4. The user is redirected to login page.

Exceptions:

1. If the user selects ‘no’, they will be returned to the Start Application use case.

Priority: High

When Available: Week 2

Frequency of Use: Every time the User wants to log out.

Open Issues: No open issues at this time.

Use case: Reset Password

Primary Actor: User

Goal in Context: Allows the user to reset the password by providing an email address.

Preconditions: Internet connection needed.

Trigger: User clicks on forgot password button (see login use case).

Scenario:

1. User observes the reset password page which includes the email address field.
2. User enters email address.
3. User selects ‘Reset Password’ button.
4. Password Server sends user email with reset password link.
5. User reads the email and clicks the link contained in the email.
6. User observes change password fields.
7. User enters a new password twice and presses ‘save’ button.
8. Both passwords are checked to make sure they are the same.
9. Passwords are updated into the Database.

Exceptions:

1. User does not enter the same password twice: Message 4 is displayed to the user underneath the password field.
2. User does not enter a valid email address: Message 1 is displayed to the user underneath the email field.
3. User enters a password that does not meet all of the requirements: Message 3 is displayed to the user underneath the password field.

Priority: High

When Available: Week 2

Frequency of Use: not regularly

Open Issues: No open issues at this time.

Use Case: Calculate Price Alert

Primary Actor: User

Goal in Context: To calculate how much the User's stock price fluctuated and send the User an automated alert.

Trigger: When the user's stock on their watch list or a stock on their stock owned list goes up/down a certain percentage.

Scenario:

1. The software pulls information on the original stock price from the Firebase Database and compares it to the current listed stock price.
2. The current stock price is then pushed to the database as the new current price.
3. The application notices a percent increase/decrease in all stocks in the watchlist/stocks owned page. The percentage will be determined by the User in Settings by a upper and lower bound according to the user's preference.
4. The computer then sends an SMS message to the User via the telephone number provided in account registration.

Exceptions:

1. The stock in the user's watchlist has increased: See message 8 is displayed.
2. The stock in the user's owned list has increased: See message 10 is displayed.
3. The stock in the user's watchlist has decreased: See message 9 is displayed.
4. The stock in the user's owned list has increased: See message 11 is displayed.

Priority: Low

When Available: Week 13

Frequency of use: Dependent on User

Open Issues: No open issues at this time.

Use Case: View Navigation slider

Primary Actor: User

Goal Context: The view Navigation slider is a side panel and top bar which houses different panes including the search bar, and a link to the trending stocks page.

Preconditions: Internet connection needed.

Trigger: The start application use case will have been triggered for this to occur for the first time.

Scenario:

1. The user views the navigation slider page which includes the following clickable buttons:
    a. Search for Stock button in top right-hand corner (See Search for Stock use case).
    b. View Trending Stocks button on the left-hand side of the panel (See View Trending stocks use case).
    c. View Logout button on the top right-hand side of the panel (See Logout use case).

Exceptions:

1. There are no exceptions at this time.

Priority: High

When Available: Week 4

Frequency of Use: Many times per day

Open Issues: No open issues at this time.

Use case: View Portfolio Summary

Primary Actor: User

Goal in Context: The user will be able to see their net worth, portfolio growth, and stock diversity, along with the price of individual stocks from the user's portfolio list and watchlist.

Preconditions: The application is connected to the internet and the IEX Trading server.

Trigger: The Start Application is launched.

Scenario:

1. The user observes net worth chart, portfolio growth chart, stock diversity chart, and their stocks owned list. The stocks owned list is a list of stocks where each stock symbol, name, price, and quantity is listed. Each stock in the stock owned list is clickable and will transfer to the user to the Individual Stock. (See View Individual Stock). Each stock also has a button on the right to remove the stock from the list.
2. The user's portfolio data is pulled from the Firebase database and stock information is pulled from the Iex Trading server.
3. The user's net worth is calculated and displayed in a bar chart, portfolio growth in a bar chart, and stock diversity in a pie chart.
4. Below the stock owned list is the watchlist which houses a list of stocks that is in the users watchlist.

Exceptions:

1. The user has no stocks in their portfolio list: Message 13 is displayed in gray text under the portfolio list.
2. The user has no stocks in their watchlist: Message 12 is displayed in gray text under the watchlist.

Priority: High

When Available: Week 5

Frequency of Use: Common usage

Open Issues: No open Issues at this time.

Use case: Edit User Settings

Primary Actor: User

Goal in Context: Allows the user to set their currency code, upper and lower bounds of SMS alerts, and phone number.

Preconditions: User is connected to internet

Trigger: User clicks on settings button to view and change settings which is located in the top right next to the search bar located within the Navigation Slider.

Scenario:

1. User observes all the different settings: currency code, SMS alerts, and the user's phone number.
2. User clicks on the text field in which setting they want to change.
3. User clicks on save button in the window to save the settings.
4. The application will send the updated information to the Firebase database to be overwritten.
5. User is returned to main/home screen with changes applied

Exceptions:

1. User does not want to change their settings.
2. User clicks on cancel button that redirects them back to the previous page

Priority: High

When Available: Week 4

Frequency of Use: Probably infrequent

Open Issues: None at this time.

Use Case: Search for Stock

Primary Actor: User

Goal in Context: The goal of this use case is to search for stocks in the search bar. Once the user has searched for a stock, the user will be transferred to the view individual stock use case to view information about that given stock.

Precondition: The application is connected to the internet and the IEX Trading server.

Trigger: User hovers the cursor over the search bar in the top right-hand corner of the page

Scenario:

1. The user observes a text field that allows the user to input text within the Navigation Slider.
2. User proceeds to enter the name of the desired stock symbol in the search bar.
3. User clicks the search button.
4. The system connects to the IEX server and displays the information on the specified stock (See View Individual Stock use case).

Exceptions:

1. String Value entered in the search bar is not recognized as an actual stock and Message 7 is displayed in red font in the Search Bar.

Priority: High.

When Available: Week 5

Frequency of use: Many times

Open issues: No open issues at this time.

Use case: View Trending Stocks

Primary Actor: User

Goal in Context: The User will be able to view information about trending stocks that will be pulled from numerous sites like CNN Money, Motley Fool, and Yahoo Money.

Preconditions: The User will have to be connected to the firebase database, and be connected to the internet.

Trigger: The User decides to click on the Trending stocks page from the Navigation Slider.

Scenario:

1. The user observes the trending stocks:
    a. Stock Symbol with price
    b. Percentage growth that particular day
    c. News regarding that stocks
2. The user clicks on one of the stocks to view individual information about that given stock (See Individual Stock use case).

Exceptions:

1. The application can not grab the stock information due to the Iex Trading API is down: See Message 20.
2. The application can not scrape stock symbol from CNN Money because the web server is down: See Message 21
3. The application can not scrape stock symbol from Motley Fool because the web sever is down: See Message 22.
4. The application can not scrape stock symbol from Yahoo Money because the web server is down: See Message 23.

Priority: Medium

When Available: Week 11

Frequency of Use: Many times per day

Open Issues: No open issues at this time.

Use case: View Infinite Scrolling Banner

Primary Actor: User

Goal in Context: To be able to view stocks that are trending, on the bottom of their screen.

Preconditions: Need access to the IEX Trading API and the Fire database.

Trigger: The scrolling bar is automatically loaded with trending stocks that the user gets to choose from.

Scenario:

1. The user observes an infinite banner that is scrolling to the left at the bottom of the screen (see View Portfolio Summary use case).
2. User clicks on the stock in which they want to observe.
3. The user is taken to the profile of the stock (see Individual Stock use case).

Exceptions:

1. Bar doesn't scroll: The scroll bar is unmoving, but still displays stocks.
2. Bar does nothing when clicked: The stock is chosen, but when clicked it doesn't take you to the stocks page so that you may get more information.
3. Trending stocks are not loaded onto scrolling banner: See message 19.

Priority: Low

When Available: week 14

Frequency of Use: Many times per day

Open Issues: No open issues at this time.

Use case: Remove from Watchlist

Primary Actor: User

Goal in Context: The goal of this use case is to remove a stock from the User's watchlist.

Preconditions: The User will have to be connected to the firebase database, and be connected to the internet.

Trigger: The User is currently in the watchlist section of the View Portfolio Summary Page.

Scenario:

1. User observes the option to remove the stock from the watchlist.
2. The user clicks the remove watchlist button.
3. The stock is removed by passing a query to the Firebase database.
4. User is given a prompt displaying message 16.

Exceptions:

1. If the watchlist is currently empty: See Message 12.

Priority: Medium

When Available: Week 9

Frequency of Use: low usage

Open Issues: No open issues at this time.

Use Case: Remove from Stocks Owned List

Primary Actor: User

Goal Context: The goal of this use case is to remove a stock from their owned list. There will be a button on the portfolio summary page to remove the stock.

Preconditions: The User will have to be connected to the firebase database, and be connected to the internet.

Trigger: User selects the remove from Stocks Owned List button from the View Portfolio Summary.

Scenario:

1. User observes the option to remove the stock from the stocks owned list.
2. The user clicks the remove stock from stock owned list button.
3. The stock is removed by passing a query to the Firebase database.
4. User is given a prompt once removed. See Message 17.

Exceptions:

1. If the stock owned list is currently empty: See Message 13.

Priority: Medium

When available: Week 8

Frequency of use: Depends on the customer and if they want to remove stocks frequently. Assumed will be used very frequently

Open Issues: No open Issues at this time.

Use case: View Individual Stock Page

Primary Actor: User

Goal in Context: To create a page that holds information about one given stock that includes information like view stock growth over months, view current stock price, current stock symbol, upcoming news, dividends, and peers.

Preconditions: The User will have to be connected to the internet. The User will need access to the IEX trading server.

Trigger: The User either searches for a given stock or finds one of interest from the portfolio summary page or the trending stock page.

Scenario:

1. The system connects to the IEX server and retrieves the following information:
    a. Stock growth
    b. Current stock price
    c. Current stock symbol
    d. Current News
    e. Summary
        i. Volume
        ii. Dividends
        iii. Earnings per share
        iv. Previous Close
        v. P/E Ratio
    f. Peers

2. The user observes the page that will display the view stock growth in the middle of the pane. The stock symbol will rest above the stock growth chart but to the left of the current stock price. Company information will reside to the left of the stock growth chart and run parallel to pane. The News about that given stock will sit below the company and will display the last three news items. The summary will include the items listed above with their respective numbers and will be directly under the chart and to the right of the news section. The last section will be the peers, which display to the right of the summary section and will list different companies that share the same field.

Exceptions:

1. The IEX trading API is down: See Message 20.

Priority: Medium

When Available: Week 12

Frequency of Use: Many times per day

Open Issues: No open issues at this time.

Use case: Add to Watchlist

Primary Actor: User

Goal Context: The goal of this use case is to add a stock to the User's watchlist.

Preconditions: The application is connected to the internet, and the IEX Trading server and the Firebase database are not down.

Trigger: The User clicks a button on the View Individual Stock use case

Scenario:

1. The user observes the option to select the add to watchlist button that is synced through the individual stock page and their watchlist within the View Portfolio Summary.
2. The user selects the 'add to watchlist' button.
3. The application will send a query to the Firebase database.
4. The application will display message 14 to the user.

Exceptions:

1. There are no exceptions at this time.

Priority: Medium

When Available: Week 10

Frequency of Use: Low usage

Open Issues: No open issues at this time.

Use Case: Add to Stock Owned List

Primary Actor: User

Goal Context: The Add to Stock Owned List feature will allow the user to be able to add individual stocks to their owned list with relational data in regards to the date of purchase of stock, purchase price of stock, and the number of shares purchased.

Preconditions: The User will have to be connected to the firebase database, and be connected to the internet.

Trigger: User selects the Add to Stock Owned List button that will be viewed with an Individual stock page which will add that particular stock to their owned list.

Scenario

1. The user observes the option to select the ‘Add to owned list’ button that is synced through the individual stock page and their stock owned list.
2. The user selects the add button
3. The user observes the dialog with the following options:
    a. Purchase price
    b. Purchase date
    c. Quantity of stock
4. The application will send a query to the Firebase database.
5. The application will display message 15 to the user.

Exceptions:

1. There are no exceptions at this time

Priority: Medium

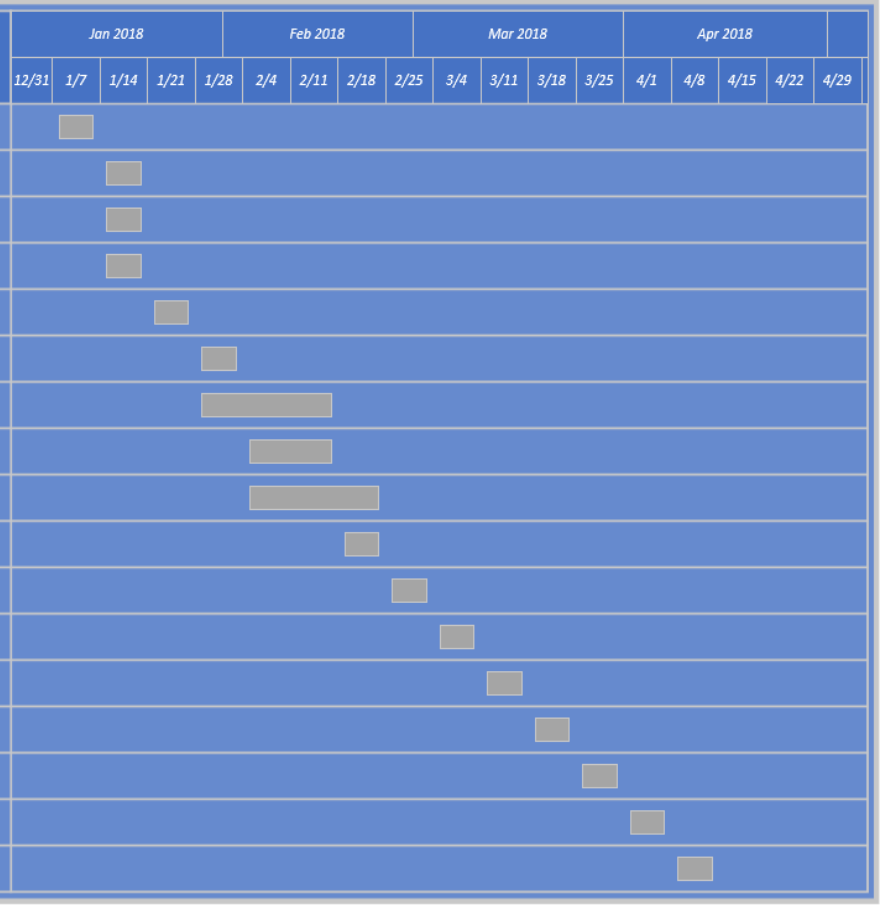
When available: Week 7

Frequency of use: Depends on the customer and if they want to add stocks frequently. Assumed will be used very frequently

Open Issues: No open issues at this time.

Stock Application Time Line

ID	Use Case	Start	Finish	Duration
1	Register	1/8/2018	1/12/2018	1w
2	Log In	1/15/2018	1/19/2018	1w
3	Log Out	1/15/2018	1/19/2018	1w
4	Reset Password	1/15/2018	1/19/2018	1w
5	Start Application	1/22/2018	1/26/2018	1w
6	Edit User Settings	1/29/2018	2/2/2018	1w
7	View Navigation Slider	1/29/2018	2/16/2018	3w
8	View Portfolio Summary	2/5/2018	2/16/2018	2w
9	Search for Stocks	2/5/2018	2/23/2018	3w
10	Add to Stock Owned List	2/19/2018	2/23/2018	1w
11	Add to Watchlist	2/26/2018	3/2/2018	1w
12	Remove from Stock Owned List	3/5/2018	3/9/2018	1w
13	Remove from Watchlist	3/12/2018	3/16/2018	1w
14	View Trending Stocks	3/19/2018	3/23/2018	1w
15	View Individual Stock Page	3/26/2018	3/30/2018	1w
16	Calculate Price Alerts	4/2/2018	4/6/2018	1w
17	View Infinite Scrolling Banner	4/9/2018	4/13/2018	1w



Messages to the User

1. Successfully logged in. Transferring to homepage.
2. Password does not meet all requirements! Please enter a valid password that is 8 characters long, with at least 1 uppercase and 1 lowercase character.
3. Please enter a valid email address.
4. Please enter valid credentials.
5. Please enter a valid email address. Email already exists!
6. Stock in user's Watchlist has Increased.
7. Stock in user's Watchlist has decreased.
8. Stock in user's stock owned list has Increased.
9. Stock in user's stock owned list has decreased.
10. Watchlist is currently empty.
11. Stock Owned List is currently empty.
12. Added stock to the watchlist.
13. Added stock to the Stock Owned List.
14. Removed stock from the watchlist.
15. Removed stock from the Stock Owned List.
16. Are you sure you want to logout? Yes or No..
17. Failed to load trending stocks.
18. IEX Trading API is down.
19. CNN Money Web Server is down.
20. Motley Fool's Web Server is down.
21. Yahoo Money Web Server is down.
22. Successful Register. Transferring to homepage…
23. Successful login. Transferring to homepage…

Glossary:

1. Dividends: a sum of money regularly paid (typically quarterly) by a company to its shareholders out of its profits (or reserves).
2. Stock symbol: unique series of letters assigned to security for trading purposes.
3. Peers: a group of individuals or entities who share similar characteristics and interests.
4. IEX trading server: is the API we are using to pull information for the stock information.
5. Stock Diversity: The differences in a stock page compared to other stocks
6. Trending stocks: Stock that is doing well and being looked at by many people
7. Net Worth of portfolio: Book value or shareholders' equity
8. Pane: is the page in which the navigation slider use case encloses.
9. Firebase Database: Defines who has access to the data and how they can access it
10. User: A shareholder
11. Stock Portfolio: a group of financial assets such as stocks
12. Stock Watchlist: Stocks that are being added to a list so they can be tracked for the users interest.
13. View Calculated Profit: Being able to see the profit of each stock on your profile if the user were to sell at the point in time.
14. View stock growth: Showing the growth in your stock over time.
15. Portfolio Summary: is the main menu of the stock app so that you can see any stocks at the same time
16. Peers: group of individuals or entities who share similar characteristics and interests.