Rutgers University

School of Engineering

Department of Electrical and Computer Engineering

StockHop: The Stock Market Fantasy League Game

REPORT #1, Revision 2

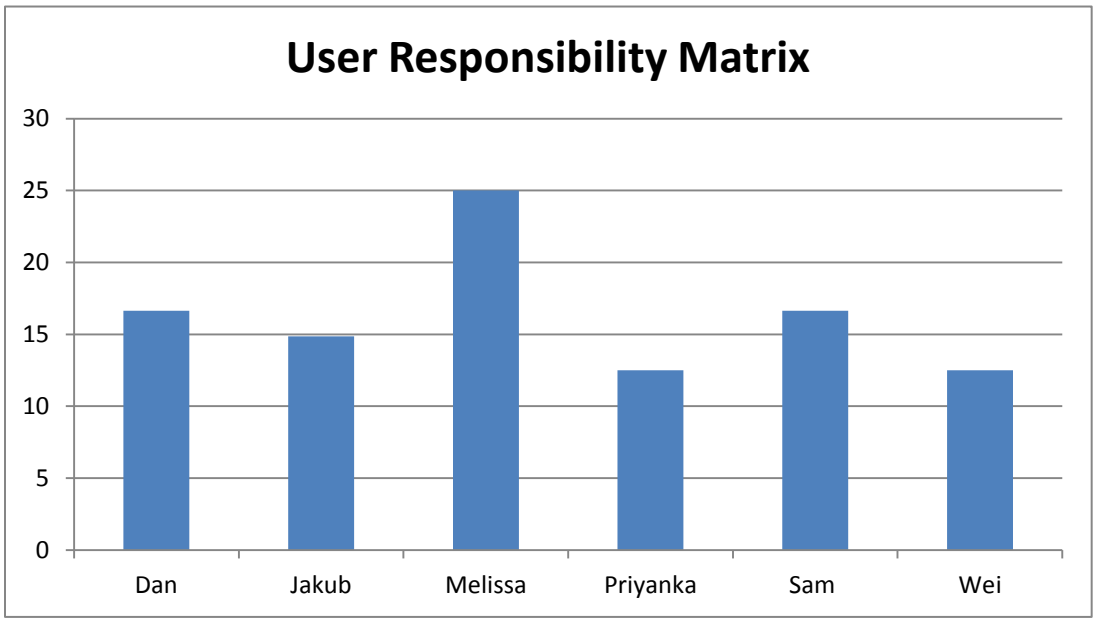
URL: http://www.thestockhop.com

Group #2

Priyanka Kale
JakubKolodziejski
Dan Marzullo
Wei Shen
Sam Ramezanli
Melissa Romanus

Individual Contributions Breakdown

		Team Member					
		Dan	Jakub	Melissa	Priyanka	Sam	Wei
Responsibility Matrix	Project Management (10pts)		40%	60%			
	Customer Statement of Requirements (6pts)			100%			
	Glossary of Terms (4pts)			100%			
	Functional Requirements Specification (37pts)	45%	5%			45%	
	Nonfunctional Requirements (6 pts)			100%			
	Domain Analysis (25 pts)				50%		50%
	User Interface Design (8 pts)		100%				
	Plan of Work (3pts)			100%			
	References (1 pt)		100%				



- *Note: Revision 1 was done by Priyanka (Domain Model), Sam (Use Cases), and Melissa (Statement of Responsibilities, general edits).*

II. Table of Contents

III. Customer Statement of Requirements

To Whom It May Concern:

Bulls and Bills, LLC has been serving the everyday stock market investor for over 50 years. With the recent economic downturn, we have seen a decrease in the number of users on our site. We want to expand our customer base as much as possible and reach people that may have never considered the prospect of stock market trading. In the past, trading in the stock market was thought to be limited to only those with a strong financial educational background. As the internet became more and more popular, it became possible for the everyday person to become involved in stock market trading through the use of online web services. Our customer base lacks novice investors, whom we think may be too intimidated to gamble with their money. We are hoping to change all of that with a stock market investment fantasy game.

The game will simulate the basic functionality and provide some of the same features as the US stock market, but instead of real money, users of the website will be able to invest with virtual money. This virtual money should be in the amount of $100,000. The stocks in which they can invest should be real-world stocks. The site should display current information about these stocks when someone attempts to make a trade with them (we realize this may not be real-time since data from real-time stock market information is expensive and we do not have the licensing to share ours with you). We do not want to provide stock forecasting, since that is something we offer on our real website and costs money.

Each user should be able to have his or her own account. These accounts should be secure. When a user logs in to the site, they should have access to their portfolio. The portfolio should contain all of the stocks that they have purchased. It is also important for users to access their transaction history, so they can keep track of where their money is going. An open transactions page will also be important to show users that have placed stop or limit orders that their transactions are still valid. Open transactions also include those orders placed after the close of the stock market. Please remember the stock market hours are Monday through Friday from 9:30AM to 4:00PM. Remember that it is also closed on Holidays such as Thanksgiving and the day after Christmas. It is a requirement to only let users trade during those times, however, we are flexible if you want to implement Pre-Market Trading or After-Hours Trading. Please notify our office before implementing these in the game.

We would like this to be a website that is as user-friendly and informative as possible. There are many existing stock market simulators on the market today. We want to set ourselves apart with superior tutorials and accessibility. It is important that it is easy for users to both learn about the stock market and easily use the website. It should have simplistic style pages but also equip users with most of the functionality they would be able to get from the real market. Since it is a game, we also want to make this a competition and offer monthly monetary incentives to our customers. Users should be able to view their rank both in the monthly competition and overall in a yearly competition. We hope to attract new customers this way as well as keep our old customers loyal to the company even if they do not currently have the money to invest in the real market.

When a user makes a trade, we would like them to have a single page in which they can buy and sell. Please make sure to implement market, limit, and stop orders. Users should be able to enter the symbol of the company that they want to purchase, the quantity of shares, whether they want to Buy or Sell, and if the transaction should be good for the day or good until cancelled (i.e. for stop/limit orders). If time permits, it will be nice to implement short orders in the game, but since we are on a tight deadline to launch the site, this is another feature that we are flexible on. The system should confirm this amount with the user before executing the order.

We would like to make stock research as easy as possible for the user. They should be able to view the stock's performance over the course of the day, as well as information like the price of the last trade, time of the last trade, change, previous close price, opening price, bid and asking price, the volume, and the market cap. They should also be able to view the stock information for the company they select from the trade page. It would be nice to have an autocomplete field for the symbols that allows the user to actually type the company name and have the stock symbol filled in for them.

We would like the site to offer advertising, so that we can make money from advertisers, and also so that we can advertise our real investment website. The advertising interface will need to have a private login. An advertiser's login should have a clean and simple interface where they can Add a New Ad, Edit an Existing Ad, or Delete an Ad. The ad and edit functions should be able to take uploaded pictures and text. When the advertiser creates a new ad, they should be able to choose a duration and will be billed for this duration. After this duration is up, the ad should be removed from the page. We would like to have any ads that go through this interface allowed to be deleted by a future system administrator at our company, so please allow a manual delete by someone with privileged access. Please use and credit our PayPal account for transactions. We are also okay with incorporating Google AdSense into the site, but we want at least one or two spaces for people to advertise through us.

We did an independent survey for our own investment website and found that people preferred the site to update them on current business news. The approved company source for this news is Reuters Business News. We would like this to be displayed somewhere the game's site. If users want to add their own favorites to this feed, that will be okay.

In order to appeal to a younger market, we also request that you create a mobile website for us that works in the Android and iPhone browsers. We are also hoping to send out text messages or emails to customers that elect to be messaged whenever a trade is processed or if they have set an alert for a stock.

We have also been thinking of the idea of allowing users to buy more virtual cash or pay a small fee to see the portfolios of the top players on the website to get an 'edge'. If you decide to implement this, we were thinking a charge of $5 gets you another $100,000. Please have some way of keeping separate games for the purchased money, so that people cannot just purchase more money in order to win the monthly competitions. These two things are not so much a requirement as a feature that would be nice to have if feasible.

We would like you to have the entire website ready to launch by December 2011. We look forward to working together in the future.

Sincerely,

Frank “Red” Johnson

Requirements from Red’s Letter

System shall be a stock market fantasy game.
System shall have separate user accounts and will allow users to change preferences.
System shall give user $100,000 virtual money.
System shall allow users to virtually buy and sell stock.
System shall provide tutorials that explain **a. How to Use the Websiteb** **b. Basics of Investing**
System shall allow companies and individuals to purchase advertising space on the website.
System shall contain portfolio, transaction history, and pending transactions.
System shall obtain stock data from a given website.

Note that there are more requirements from Red’s Letter but these are the most important.

IV. Glossary of Terms

• Actors – External players of a system.

• After-Hours Trading – Refers to trades made after the market's traditional close time of 4PM EST

• Attributes - properties of concepts, are usually for storage/accounting purposes or state

Information

• Associations – Describes relationship between

• Domain Model – a conceptual model of a system or piece of software

• FURPS –an acronym meaning Functionality, Usability, Reliability, Performance, and Reliability, which are how nonfunctional requirements are

•Gantt Chart- project planning chart that shows time and the duration of tasks

• Google AdSense – A website that allows external advertisements to be placed on a user's personal website

• MySQL – an open-source database system

- Pre-Market Trading – Stock trading that takes place before the 9:30AM opening of the stock market

• Portfolio –The collection of all stocks that a user owns and their current value

• System Administrator –Responsible for maintaining the server, including setup, security, user accounts, etc.

• UML – (Unified Modeling language) Software Engineering modeling language

• Use Case – shows a certain scenario of system and the actors that take place in it

• User - Any individual visiting the website

V. Functional Requirements Specification

- Stakeholders
    a. Bulls and Bills, LLC – The customer and sponsor of this project is interested in this project to attract customers to their company and to make revenue off of the game website.
    b. Advertisers – Other financial institutions may want to advertise on the website, but really any appropriate advertisements are approved. Therefore, potential advertisers can come from a variety of backgrounds (e.g. software, job websites, investment websites, banks, etc).
    c. Users
        - Teachers and Students – Educators can use the website to assist with a business or economics course.
        - Novice Users – Novice users will need the most guidance through the tutorials.
        - Intermediate/Expert Users – These users may be focused in riskier portfolios and also in being at the top of the rankings to increase their chances of winning prizes.
        - Investors – Investors may be interested in how efficiently this website models the real stock market.
    d. Internet game fanatics – Some people love competition and love to play games. These users of the website are interested in winning monthly and yearly games.
    e. Development Team – The website developers are concerned with developing a reliable product in the given amount of time.
    f. System Administrator – The system administrator is concerned with the maintainability and security of this website.
    g. Testers – The site testers, a group that will be able to use the site in demo mode before it goes to full production, will be able to give feedback.
- Actors and Goals
    - **Actor 1: New Player**
        i. **Type:** Initiating actor
        ii. Goals:
            i. To create new account.
            ii. Learn about game.
            iii. Play the game.

- **Actor 2: User**
    i. **Type:** Initiating actor
    ii. Goals:
        i. Access his/her account.
        ii. View cash balance and transaction history at Homepage.
        iii. Buy a specified number of shares of a company's stock.
        iv. Sell a specified number of shares of a company's stock.
        v. View current and past information about the price at Research section.
        vi. View a user's stock account performance relative to other users' performance.
        vii. View Transaction History
        viii. View Portfolio
        ix. Edit Preferences

- **Actor 3: Advertiser**
    i. **Type:** Initiating actor
    ii. Goals:
        i. Access his/her account.
        ii. Manage advertisements on the website.
        iii. Attract users to find out about his/her service.

- **Actor 4: Database**
    i. **Type:** Participating actor
    ii. Goals:
        i. Save information of new users.
        ii. Check its data to prevent duplicating for usernames.
        iii. Save and Keep Transaction History.
        iv. Save and Keep users' portfolio
        v. Save stock market updates.
        vi. Delete information when there is a request from System.

- **Actor 5: Email Server**
    i. **Type:** Participating Actor
    ii. Goals:
        i. Send email to contacts when it receives request from System.

- **Actor 6:Yahoo Finance**
    i. **Type:** Participating

ii. Goals:

i. Provide up to date information regarding Stock market.

ii. Send information to System whenever there is a request.

- **Actor 7: Timer**
  - i. **Type:** Participating Actor
  - **ii. Goals:**
    - i. Prompt system for requesting new Stock market data from Yahoo Finance.

- Use cases
  - i. Casual Description

UC-1:Registration: All visitors to website are able to create free accounts by choosing user name and password and giving some basic information of themselves such as birthday.

UC-2:Sign in: A user is able to sign in to his/her profile from the login page. Once logged in to an account, the user can then access details about the account and make changes.

UC-3: View Home Page: A user is able to view portfolio summary (best/worst stocks), cash balance, overall rank, and other personalized information from profile's homepage.

UC-4: Buy Stocks: Users can search for available stocks as well as select a type of stock order and the number of shares they wish to purchase. If the user has enough cash to cover the cost, the order can be placed, and will then be added to the pending transaction list.

UC-5: Sell Stocks: A user can sell his/her owned shares of stock by selecting the company, number of shares, and type of order. The order will then be added to the pending transaction list.

UC-6: Research Stocks: Users can request and view information about a company's current stock price and price history. By entering a company's stock symbol, the user will be presented with the company's stock information.

UC-7: Send Notifications: The user can receive an E-Mail or SMS message when an important event happens in an account. When a pending buy or sell order is completed or cancelled, a notification will be sent to user.

UC-8: View Rankings: Users can view their ranking in terms of investment performance among other users of the system. The user can view the top ranking players as well as sort the list based on a variety of criteria.

UC-9: User sign out: A user can click the "log out" button to restrict access to their account by unauthorized users. After logging out, a user will need to reenter a valid username and password pair to regain access to the account.

UC-10: View Help: The user can access a variety of documentation about how to use the Stock Hop site as well as basic tutorials on the stock market.

UC-11: View Transaction History: Users can view all of the previous transactions on an account. All completed orders will appear with the details of the transaction.

UC-12: View Pending Transactions: Users can view all of the pending transactions in an account. If an order has yet to be completed or is submitted during off-hours, it will be viewable on the pending transaction page

UC-13: View Portfolio: A user can view the portfolio associated with an account. By navigating to the portfolio page, the user can see the stocks that are owned and details about them such as the number of owned shares and the profits of each.

UC-14: User Preferences: A user can change various settings about his or her account by navigating to the preferences page. There a user can set up E-Mail and SMS messages, change the preferred email contact, change his or her password, or delete the account entirely.

UC-15: Manage Advertisements: The advertiser can log in to the website, add advertisements, and remove their advertisements.

UC-16: Maintain Website: The system administrator can access the backend for periodic maintenance and upgrades. A team of maintenance coders can add or remove features or bugs as needed.

UC-17: Update Stock Prices: The Stock Database will be updated on a timely basis. The data need to use an external source

The login use cases are not usually shown as use cases in software engineering development, because a user does not want (or request) to login. The reason that it is included in this report is that the login functionality does interact with the database, the user's browser, and the system. Additionally, the way StockHop is written, a user only sees a registration page and login option when he or she visits the site (user cannot browse the site without logging in). This makes login a necessary step in being able to view the features of the site, and thus, facilitates its need to be included in the Use Cases.

ii. Fully Dressed Descriptions

The following in depth use cases reflect those features that will be ready by Demo 1.

Use Case UC-1 : Registration

- Initiating Actor: person or advertiser who wishes to make an account in website.
- Participating Actors: Yahoo Finance
- Actor's Goal: Create an account
- Pre-condition: having a computer which is connected to Internet!
- Post-condition: An account is created and user is able to login to the system.
- Flow of Events for main success scenario:
  - → User goes to start page of our website.
  - → System asks for username and password.
  - → User choose arbitrary username and password
  - ← System check database to see whether this username is already taken or not.
  - → User gives additional information such as Birthday and sex**.**
  - ← System save the information in database.
  - ← System show confirmation note for the user.
- Flow of Events for Extensions (Alternate Scenario):
  - → User enter a used username.
    ← System would return to main page and shows message: "This username is taken"

Use Case UC-2 : User Sign In

- Initiating Actor: User
- Participating Actors: Database
- Actor's Goal: To access his/her account
- Pre-condition: User or advertiser has a previously established account.
- Post Condition: The system displays the user's personalized home page.
- Flow of Events for main success scenario:
  - → User visits StockHop webpage.
  - ← System prompts user to enter username and password or to create a new account.
  - → User enters valid username and password pair.
  - ← System authenticates username and password with **database**
  - ← System take User to Home page (UC-3)
- Flow of Events for Extensions (Alternate Scenario):

← System (a) detects invalid username and password with **database** (b) informs **user** of the error (c) prompts the **user** to try entering information again or to create a new account.

Use Case UC-3 : View Homepage

- Initiating Actor: Users and Advertisers
- Participating Actors: Yahoo Finance , Database
- Actor's Goal: To view portfolios, cash balance, transaction history and other personalized information.
- Pre-condition: User is logged in.
- Post-condition: none worth mentioning.
- Flow of Events for main success scenario:
  - → User login to profile.
  - ← System connects to Yahoo Finance and receive new updates and information about stocks.
  - ← System compare the incoming information with User's portfolio
  - ← System updates Database.
  - ← System displays Homepage.
  - → User is informed about recent changes and update in his/her account.

It should be noted that UC-4 Buy Stocks normally includes three different types of orders: market, stop, and limit. It is shown in the system sequence diagrams and fully-dressed descriptions as it applies to market orders, because that is the projected amount of work that will be ready for Demo 1. The reader of this document should keep in mind that this is not a complete description or system sequence diagram for the full functionality of Buy Stocks but instead for the Market Order.

Use Case UC-4 : Buy Stocks

- Initiating Actor: Users
- Participating Actors: Database , Yahoo Finance
- Actor's Goal: To buy a specified number of shares of a company's stock
- Pre-condition: User is logged in.
- Post-condition: (1) The purchased number of shares appears in the user's account. (2) The purchase price is deducted from the user's available cash
- Flow of Events for main success scenario:
  - → User visits the buy section of the StockHop webpage.
  - ← System prompts user to enter stock name/symbol and amount of stocks.
  - → User enters valid stock name/symbol.
  - ← System queries Yahoo Finance for current price per share of the entered stock symbol.

- → Yahoo Finance finds a match for the stock name/symbol and return the price.
- ← System compute total cost and add the commission cost to it.
- ← System verify that total cost plus commission is less than user's cash balance.
- ← System asks user to enter type of price (Market, Limit, Stop).
- → User choose type of price he/she wish to have.
- ← System send the user's request to Yahoo Finance.
- ← System send information to database and update user's portfolio.
- ← System notifies user that the transaction has been completed.
- include::*Send Message* (UC-7)

❖ Flow of events for Extensions (Alternate Scenarios) :

- **User** enters invalid stock symbol
  - **System** queries Yahoo Finance for current price per share of the entered stock symbol
  - Yahoo Finance notifies the **System** that the stock symbol is invalid
  - **System** (a) notifies the **User** that the stock symbol is invalid and (b) prompts for the **User** to enter a different stock symbol.
- System determines that the **User** does not have enough cash to make the desired purchase
  - **System** (a) notifies the **User** that there is not enough cash to make the purchase and (b) prompts the **User** to enter new purchase information

Use Case UC-5 : Sell Stocks

❖ Initiating Actor: Users and Advertisers
❖ Participating Actors: Database, Yahoo Finance, Email server.
❖ Actor's Goal: Sell stocks to investors
❖ Pre-condition: User is logged in, User owns at least one amount of share.
❖ Post-condition: Transaction is done and user's portfolio is updated with stock removed from it .
❖ Flow of Events for main success scenario:

- → User visits the sell section of the StockHop webpage.
- ← System prompts user to enter stock name/symbol and amount of stocks.
- → User enters valid stock name/symbol and amount of stocks he wish to sell.
- ← System request database for user's portfolio information.
- → Database sends information.
- ← System verifies that user own the stock and the amounts of share he wish to sell.
- ← System queries Yahoo Finance for current price per share of the entered stock symbol.
- → Yahoo Finance finds a match for the stock name/symbol and return the price.
- ← System compute total income and subtracts the commission cost from it.
- ← System asks user to enter type of price (Market, Limit, Stop).
- → User choose type of price he/she wish to have.
- ← System sends user request to Yahoo Finance.

- ← System removes sold shares from portfolio and sends information to database to update user's portfolio.
- ← System notifies user that the transaction has been completed.

❖ Flow of Events for Extensions (Alternate Scenario):

1. → User enters wrong amount of share.
   ← System check database and asks user to enter true amount of money.
2. → our share will not be bought by the Stock investors because of high price or type of share.
   ← System sends user email that transaction is not done.

Use Case UC-6 : Research Stocks

❖ Initiating Actor: Users
❖ Participating Actors: Yahoo Finance.
❖ Actor's Goal: To view current and past information about the price of a company's stock.
❖ Pre-condition: (1) User or advertiser has a previously established account (2) User is currently logged into an account
❖ Post-condition: The system will display information about the current and historical prices of a company's stock to the user.
❖ Flow of Events for main success scenario:
- → User visits the research section of the webpage
- ← System prompts user to enter stock symbol
- → User enters valid stock symbol
- ← System queries Yahoo Finance for current price per share of the entered stock symbol as well as any available history about the stock.
- → Yahoo Finance (a) returns requested information to the System
- ← System (a) displays retrieved information to the User (b) provides an area for the User to enter another stock symbol to look up another stock

❖ Flow of Events for Extensions (Alternate Scenario):

1. ← System queries Yahoo Finance for current price per share of the entered stock symbol as well as any available history about the stock
2. Yahoo Finance notifies the System that the stock symbol is invalid
3. System (a) notifies the User that the stock symbol is invalid and (b) prompts for the User to enter a different stock symbol

Use Case UC-7 : Send Notifications

- Initiating Actor: Email server
- Participating Actors: User, Advertiser, Database and Yahoo Finance.
- Actor's Goal: Send notifications in order to inform profile owner about events happened in the profile
- Pre-condition: User has an account, User has activity in his/her account.
- Post-condition: Email would be sent to User successfully.
- Flow of Events for main success scenario:
    - ← System connects to Yahoo Finance and update its information.
    - ← System notice an important new event.
    - ← System signals email server for sending notification to User.
    - → User checks his/her email and will aware of what has happened.
- Flow of Events for Extensions (Alternate Scenario):
    - → User enters email address.
    - ← System sends email to wrong address

Use Case UC-8 : View Rankings

- Initiating Actor: User
- Participating Actors: User System, Database
- Actor's Goal: To view a user's stock account performance relative to other users' performance
- Pre-condition: 1) User or advertiser has a previously established account 2) User is currently logged into an account
- Post-condition: Email The system will display information about the user's ranking based on certain criteria
- Flow of Events for main success scenario:
    - → User visits the rankings section of the webpage.
    - ← System requests account information from the Account Database and sorts users by certain criteria
    - ← System signals a) displays a default view of the User's ranking and (b) provides clickable options to the User to alter the display or view top ranking users.

Use Case UC-9 : User Sign Out

- Initiating Actor: User

- Participating Actors: System, Database.
- Actor's Goal: To protect account from unauthorized access and to delete account.
- Pre-condition: 1) User or advertiser has a previously established account 2) User is currently logged into an account
- Post-condition: The user's account is not immediately accessible from the current computer
- Flow of Events for main success scenario:
  - → User is logged into an account on the webpage.
  - → User clicks the "log out" button on the webpage
  - ← System sends all unsaved data to the Database for storage
  - ← System (a) displays "successfully logged out" message to user and (b) disables immediate continued access of the user's account.

Use Case UC-11 : View Transaction History

- Initiating Actor: User
- Participating Actors: Yahoo Finance, Database, User.
- Actor's Goal: to view all of the previous transactions on an account and all completed and pending orders will appear with the details of the transaction.
- Pre-condition: User should have at least one transaction since he/she has joined the website
- Post-condition: User is aware of history of transactions and their current status.
- Flow of Events for main success scenario:
  - → User Login.
  - → User choose transaction history from the Homepage.
  - ← System request database for user's transaction information
  - → Database sends information (information are already updated in database because of our database base update in UC-17)
  - ← System request information from Yahoo Finance
  - → Yahoo Finance sends current market updates.
  - ← System verify information
  - ← System signal User about his transaction history.

Use Case UC-12 : Help

- Initiating Actor: User
- Participating Actors: Database
- Actor's Goal: To learn about how the system as a whole or a particular function of the

system works

- Pre-condition: (1) User or advertiser has a previously established account (2) User is currently logged into an account
- Post-condition: The user is more informed about how to use the system.
- Flow of Events for main success scenario:
  - → User is logged into an account on the website
  - → User navigates to the "Help" section on the webpage
  - ← System displays all clickable help topics to the User
  - → User clicks to request a help topic from the System
  - ← System sends new data to database.
  - ← System the help content for the requested topic as well as an option to return to the previous list of help topics.

Use Case UC-13 : View Portfolio

- Initiating Actor: User
- Participating Actors: Yahoo Finance, Database, and User.
- Actor's Goal: A user can see the stocks that are owned and details about them such as the number of owned shares, profits of each, current price and recent changes.
- Pre-condition: User should have at least one share since he/she has joined the website
- Post-condition: User is aware of history of his/her stock values and information.
- Flow of Events for main success scenario:
  - → User Login.
  - → User choose View Portfolio from the Homepage.
  - ← System request database for user's transaction information
  - → Database sends information (information are already updated in database because of our database base update in UC-17)
  - ← System request information from Yahoo Finance
  - → Yahoo Finance sends current market updates.
  - ← System verify information
  - ← System signal User about his transaction history.

Use Case UC-14 : Preferences

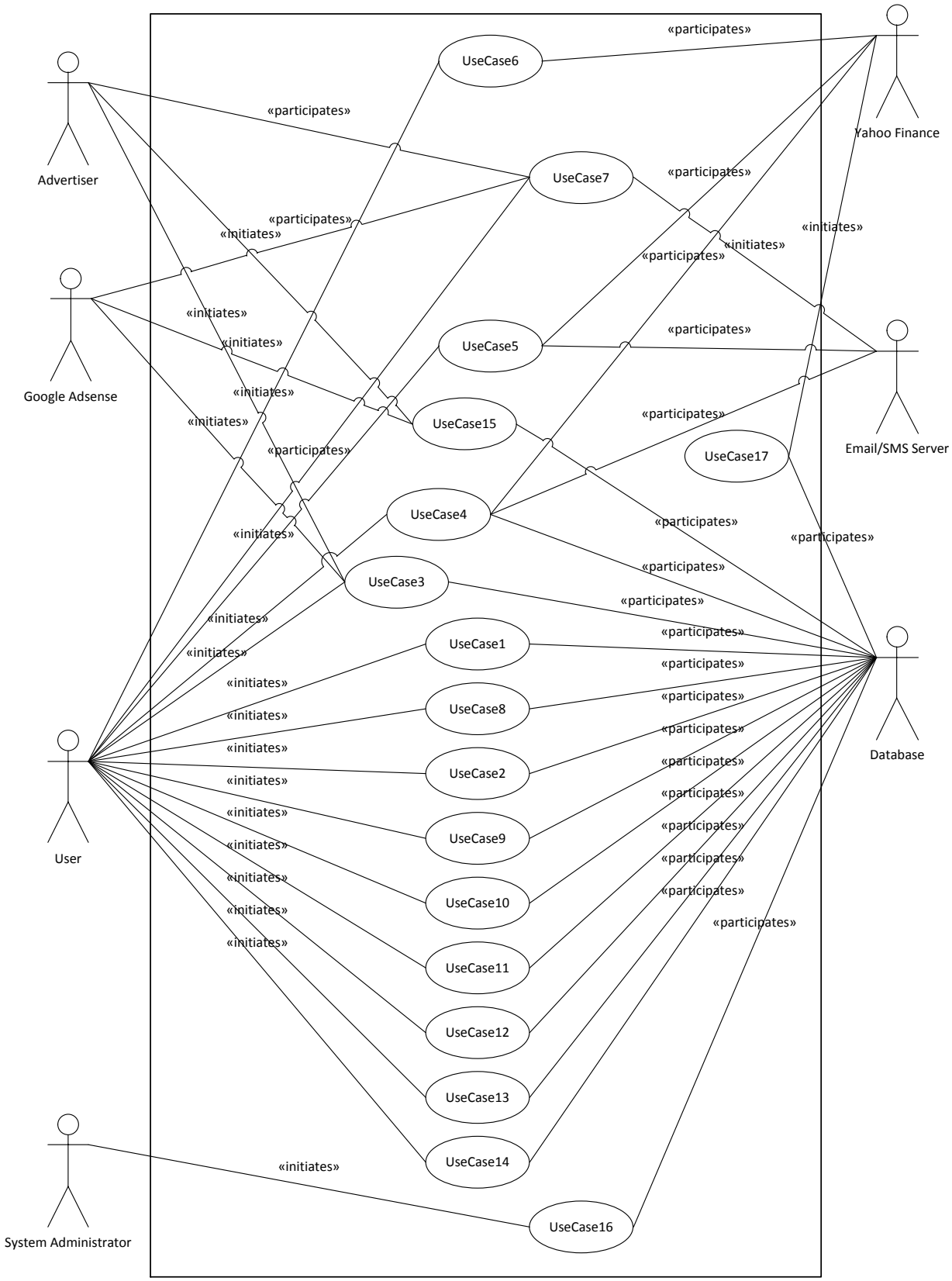
- Initiating Actor: User
- Participating Actors: Database, Email Server
- Actor's Goal: Delete account, change email, change password.
- Pre-condition: User has an account.

- Post-condition: Profile preference will be changed.
- Flow of Events for main success scenario:
  - → User goes to preference page and select the setting he/she wish to change.
  - ← System asks for new data.
  - → User enter new data.
  - ← System sends new data to database.
  - ← System prompt email server to notify the user of successful operation.
  - → Email server sends user notification about operation.

Use Case UC-15 : Manage Advertisements

- Initiating Actor: Advertiser
- Participating Actors: Database, email server, Users.
- Actor's Goal: display and manage advertisements on the website.
- Pre-condition: Advertiser is logged in. Advertiser has the right from webpage instructors to post his advertisements on user profiles.
- Post-condition: Advertisements will be posted on Users profiles. Every time any user clicks on one of the links, amount of money would be sent to advertiser's bank account.
- Flow of Events for main success scenario:
  - → Advertiser log in to his/her profile.
  - → Advertiser choose a picture (or motion picture) for his product.
  - ← System add the picture to database and post the picture on advertisement section on the website.
- Flow of Events for Extensions (Alternate Scenario):
  1. → agreement between user and web instructor is expired.
     ← System would delete the pictures from the website.
  2. → Advertiser post inappropriate picture on the website!
     ← System (web instructor) would delete the advertisement from the website.

iii. Use Case Diagram


Advertiser
Google Adsense
User
System Administrator
Yahoo Finance
Email/SMS Server
Database
UseCase1
UseCase2
UseCase3
UseCase4
UseCase5
UseCase6
UseCase7
UseCase8
UseCase9
UseCase10
UseCase11
UseCase12
UseCase13
UseCase14
UseCase15
UseCase16
UseCase17
«participates»
«initiates»

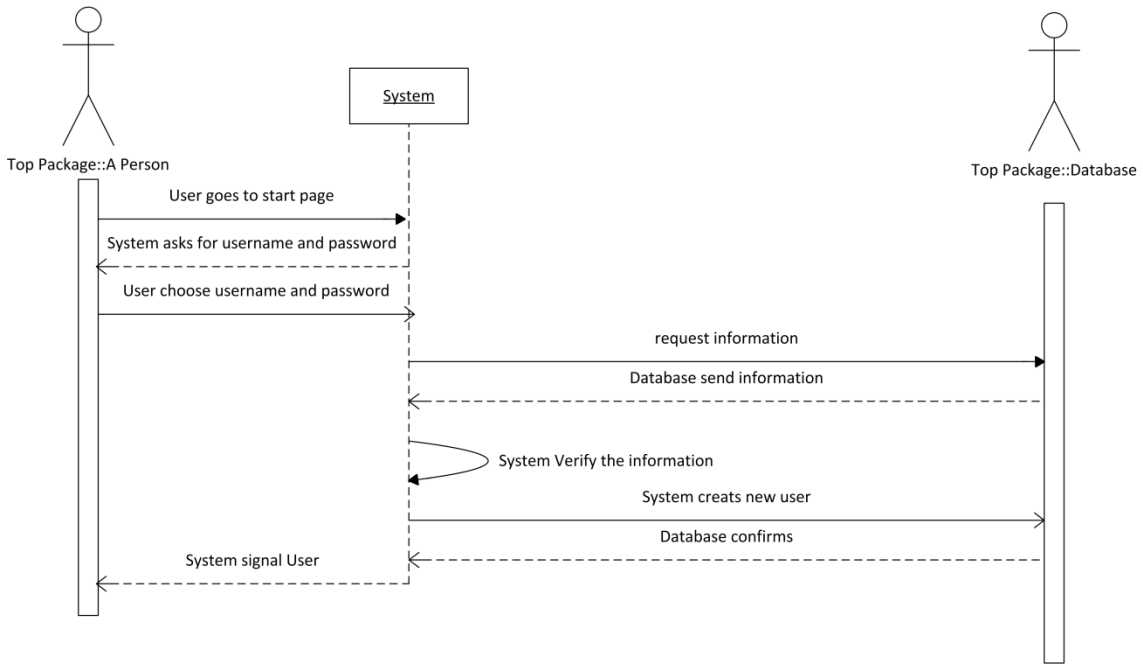
iv. System Requirements – Use Case Traceability Matrix

Requirement Number	Requirement	Use Case Traceability
1	**System shall be a stock market fantasy game.**	All Use Cases
2	**System shall have separate user accounts and will allow users to change preferences.**	1,2,9,14
3	**System shall give user $100,000 virtual money.**	1,3,4,5
4	**System shall allow users to virtually buy and sell stock.**	4,5
5	**System shall provide tutorials that explain** **How to Use the Website** **Basics of Investing**	10
6	**System shall allow companies and individuals to purchase advertising space on the website.**	15
7	**System shall contain portfolio, transaction history, and pending transactions.**	13,12,11,10
8	**System shall obtain stock data from a given website.**	17

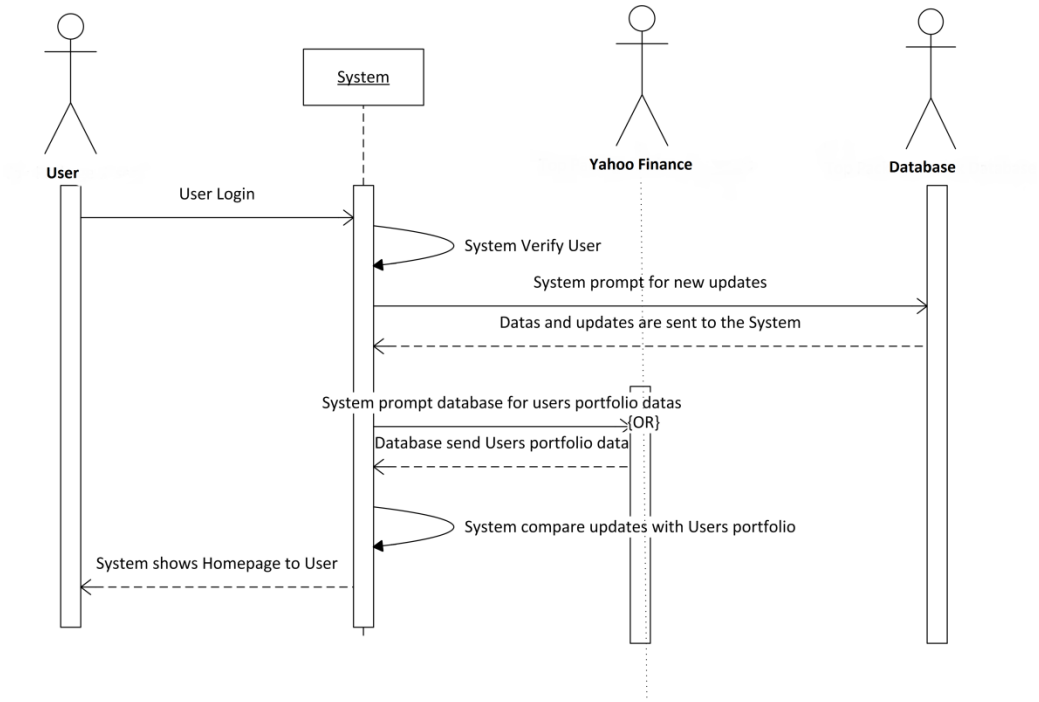
d. System Sequence Diagrams

These diagrams are explained in the above Use Case descriptions. Please see the above descriptions for in depth explanations of each diagram. They are organized by Use Case, where UC-1 corresponds to Use Case 1, etc. This was done in the interest of not being repetitive.

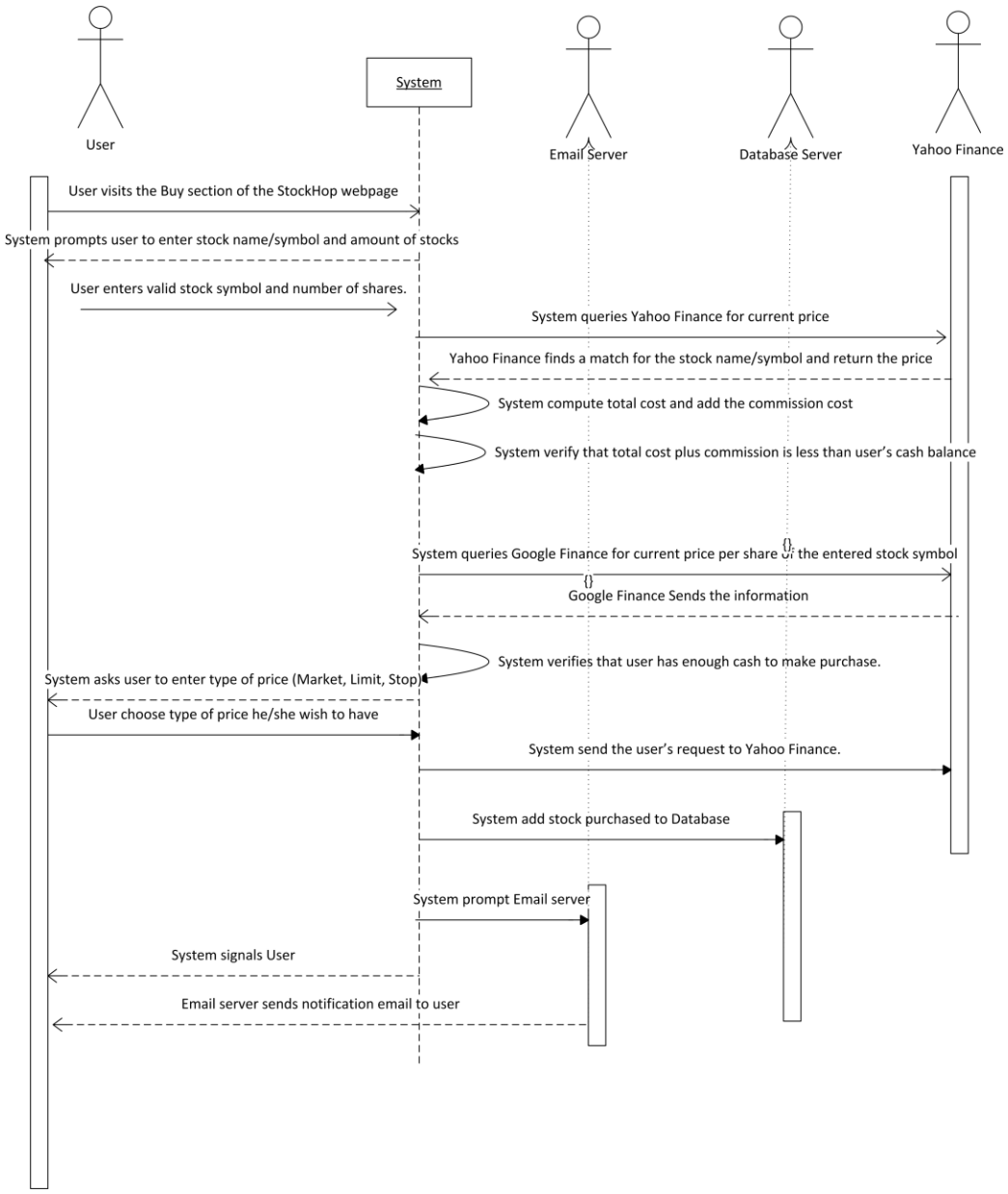
UC-1 Registration:



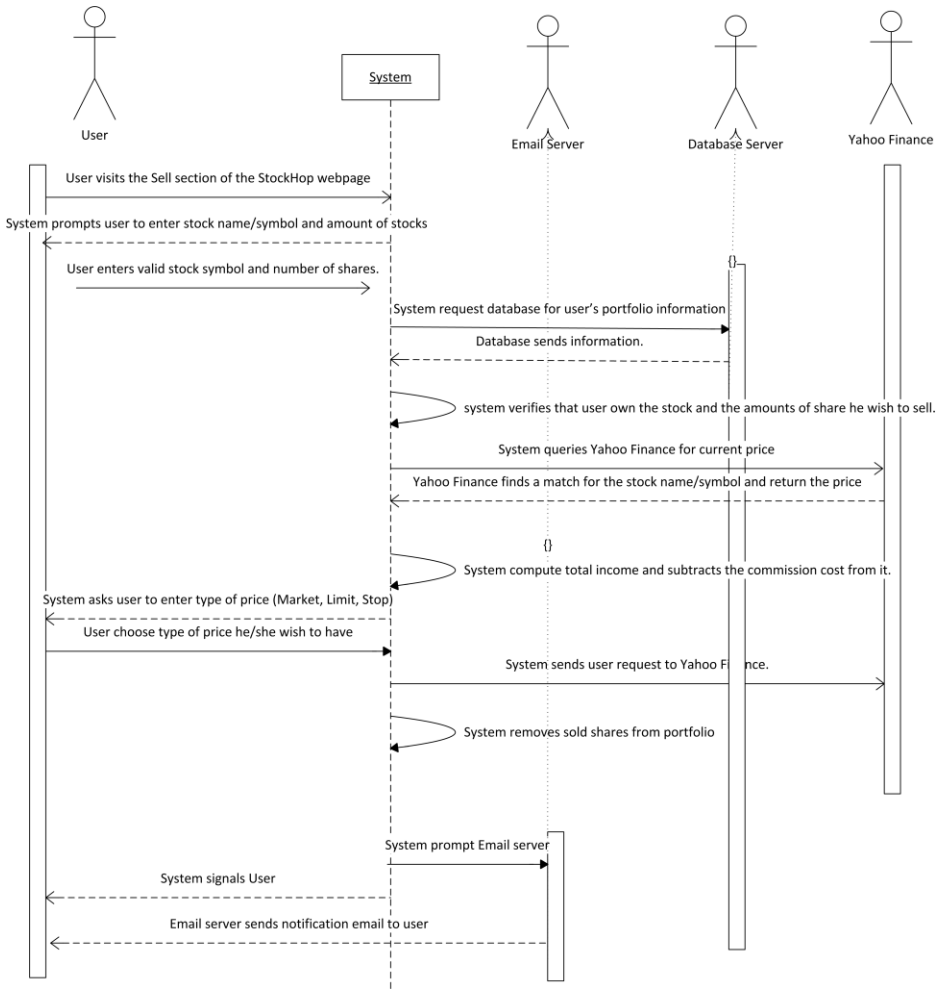
UC-3 View Homepage:



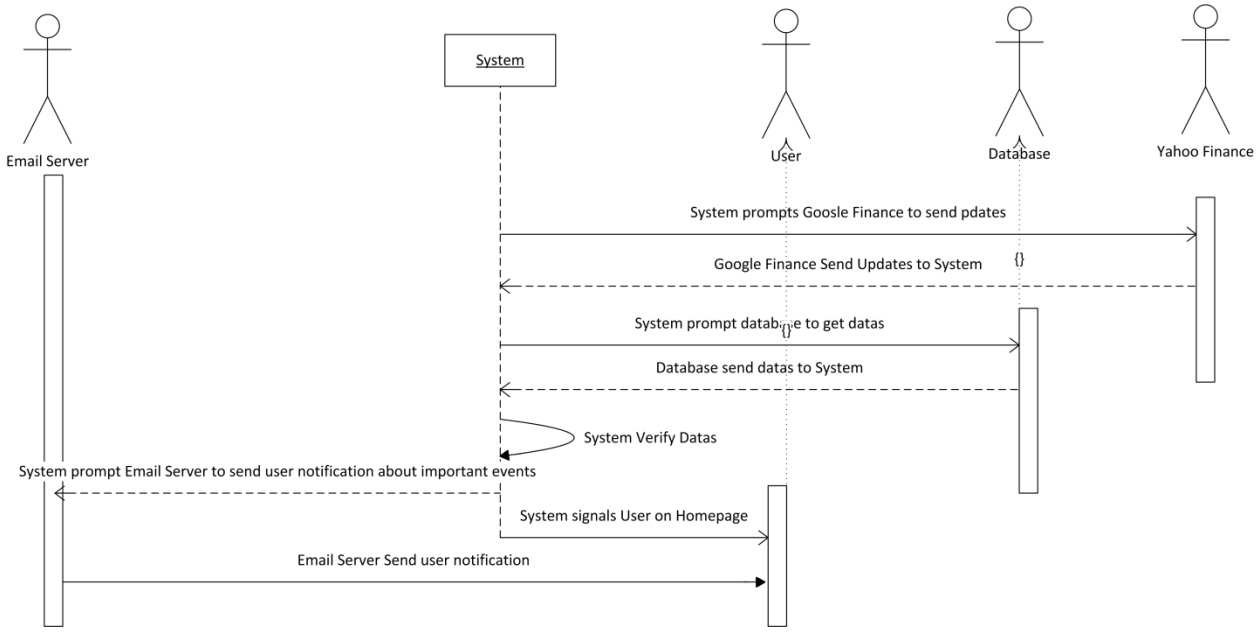
UC-4 Buy Stock:



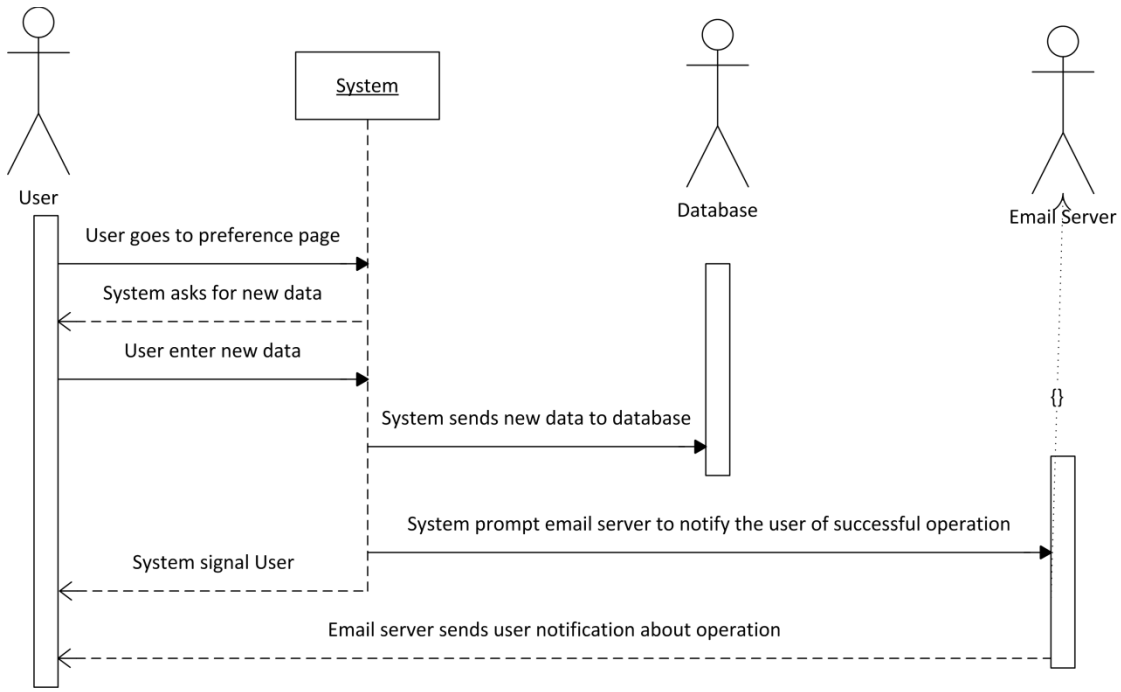
UC-5 Sell Stocks:



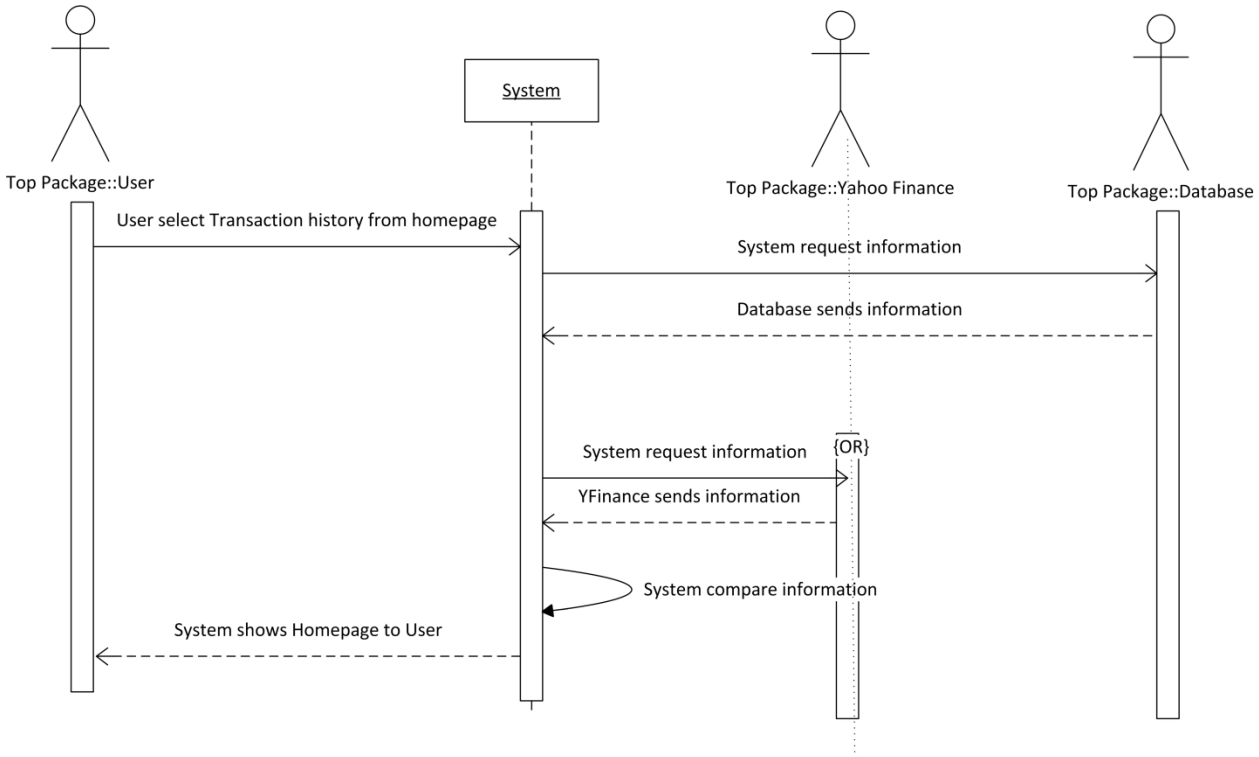
UC-7 Send Notifications:



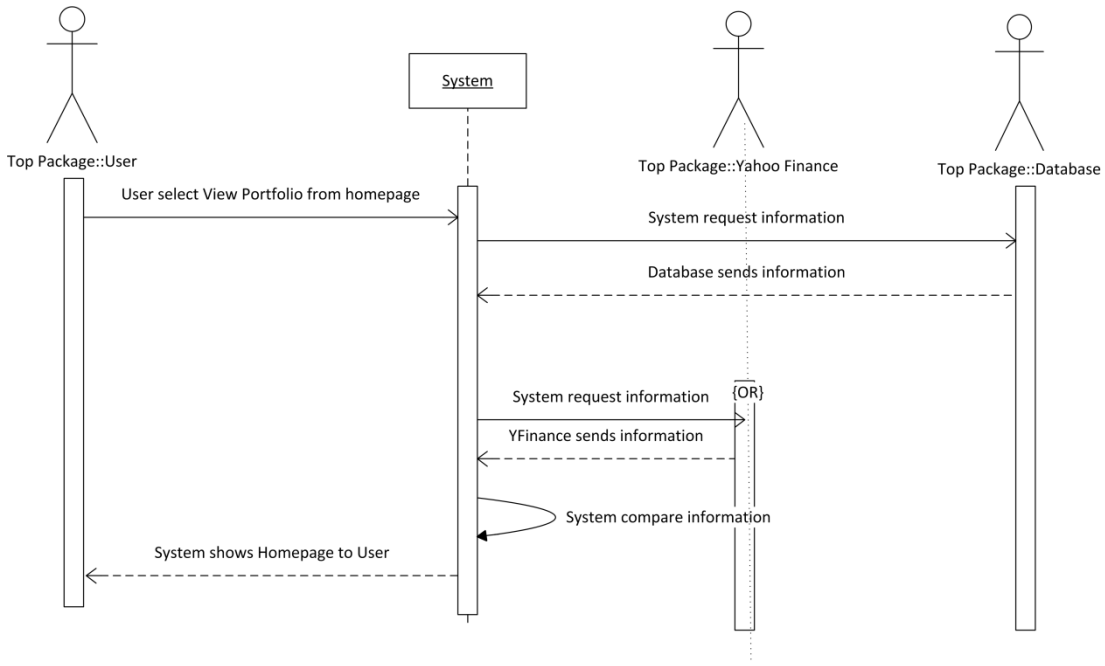
UC-11 Preferences:



UC-11 View Transactions History:



UC-13 View Portfolio:



The team had many discussions on how to implement the stock queries from Yahoo Finance. Rather than periodically dumping all information of all possible stocks to a database, the team decided that the queries to get information from Yahoo Finance would be done at the time that

the user is querying. The group decided that stop and limit orders would spawn server-side processes that periodically check Yahoo Finance and create an event if any of its criteria is met.

VI. Nonfunctional Requirements (FURPS)

- **F**unctionality

  Feature set
  - System shall be able to display stock information on website automatically updates every minute (user does not have to refresh to page)
  - System shall be able to send both email and text message alerts on transaction completion and individual stock alerts
  - System shall be able to display a business News RSS Feed
  - System shall be user-friendly and minimize time to completion of orders
  - System shall display scrolling stock ticker at the top of trading page

  Capabilities
  - System shall be able to model stock trading -- market orders, limit orders, stop orders, possible short orders
  - System shall be able to run on multiple browsers, including mobile web

  Security

  Logins to the website will be password protected so that users cannot view other users' portfolios. Any charging for extra features will be done through external secure services such as PayPal. The login forms will take extra security measures to prevent against MySQL injection attacks (i.e. an attack where the user enters malicious code in a PHP form that accesses a MySQL database to try to take over the server or tables where sensitive information is stored). The MySQL database will store passwords in an encrypted format.

- **U**sability

  The website should be easy to navigate and operate in a realistic manner to the actual stock market. Each page on the website will have the same top bar, navigation, and color scheme so that users feel a consistency as they visit all of the pages within the site. In addition, the website should include easy to use navigability and minimization of the time it would take a user to complete transactions. The website should also offer easy-to-understand tutorials about how to use the website.

- **R**eliability

  The website shall be as accurate as possible by taking information from Yahoo Finance. These stock prices are delayed by 15 minutes, but should still be consistent with the information that would be displayed on the original websites (i.e. stock prices will not be made up).

The website host server requires a Verizon FIOS internet connection to work and runs on PSE&G power. If this connection were to go out due to problems from either company, this would cause the website to be affected. These external factors are not controllable, but the server itself is attached to a UBS backup power unit to reduce its downtime in the event of a power failure.

Additionally, the system shall periodically backup the user data and MySQL database tables in order to reduce the time to get the website back up and running should it become compromised in any way. This also reduces the time to recover if the primary server storage fails.

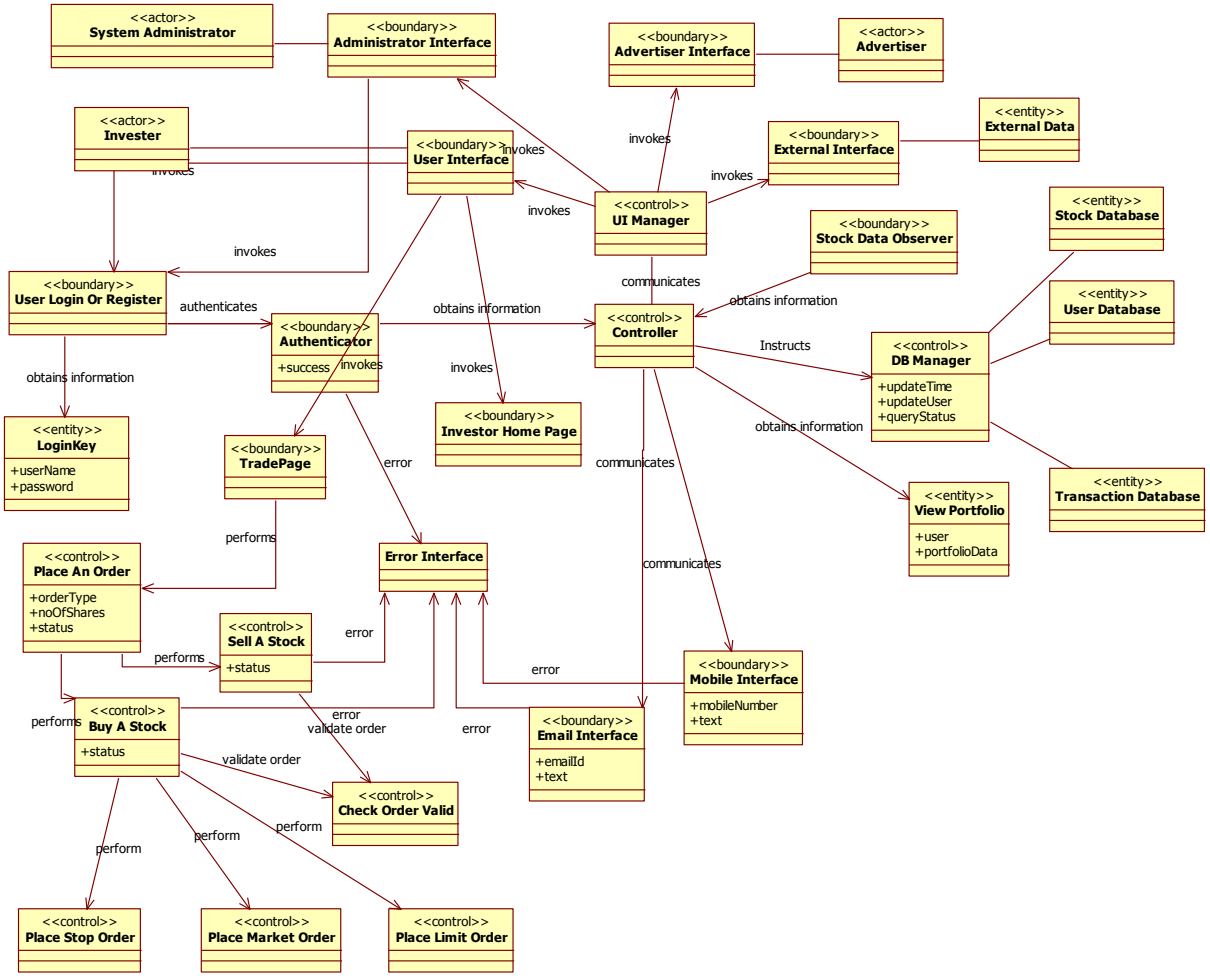
- **P**erformance

  The user should be not experience slow-down when the server is gathering data from Yahoo Finance or when other users are trying to make transactions. The actual server hardware is lightweight but is easily upgradable. The website and server-side applications will be written in a way that is not intensive to the server hardware and also that provides users with the quickest possible response time.

- **S**upportability
  The website shall be written to be as modular as possible to ease any code maintenance that needs to take place in the future and to account for additional features to be added. Code shall be well-commented for future maintainability. It should be written such that it is expandable in the future to run on upgraded server hardware. Since the users of the site are so diverse, the overall site aims to have compatibility with as many different browsers as possible, including Google Chrome, Mozilla Firefox, Internet Explorer, Safari, as well as mobile browsers.

VII. Domain Analysis

a. Domain Model



The following sections explain the Domain Model diagram.

i. Concept Definitions

Concept Name	Type	Concept Description
Administrator Interface	D	Interface for system administrator actor to perform various operations like maintaining user accounts, configuration changes, data manipulation etc
UserLoginOrRegister	D	Displays functions for users to log in into the system or for new users

		to register themselves.
Login Key	K	Stores the username and password information.
Place An Order	K	Stores information about a transaction initiated by an investor.
Investor Home Page	D	Displays information after users log in.
Trade Page	D	Provides interface for user to enter data to buy and sell stocks. Once a symbol is selected, interacts with Yahoo finance to display the stock information.
Buy A Stock	D	Stores data for the order placed to buy a stock. Uses functions provided by Check Order Valid to verify the validity of the order. Adds data to open orders database table. For valid orders based on the order type invokes appropriate transaction i.e. market, limit, stop.
Sell A Stock	D	Stores data for the order placed to buy a stock. Uses functions provided by Check Order Valid to verify the validity of the order. For valid orders based on the order type invokes appropriate transaction i.e. market, limit, stop.
Check Order Valid	D	Provides functionality to check the validity of the order placed. Checks if the symbol is valid, user has enough cash to perform the transaction etc.
Place Market Order	D	Provides functionality to complete a market order transaction. Checks to see if 20 minutes has passed.

		Checks if user has enough cash to perform the transaction. If yes, execute order. Update information in database to reflect purchase (decrement user's cash, update transaction history and portfolio, etc). Notify user if user preferences are set.
Place Limit Order	D	Provides functionality to complete a limit order transaction. Gets the stock prices and volumes for all open orders. Monitors data from yahoo to check if the limit price has reached. If yes, further processes the order by purchasing or selling the shares at limit price and following the same path as the market order.
Place Stop Order	D	Provides functionality to complete a stop order transaction. Gets the stock prices and volumes for all open orders. Monitors data from yahoo to check if the price has reached. If yes, further processes the order by transitioning the order from limit order to market order.
View Portfolio	K	Provides interface for user to view his portfolio. Interacts with database to obtain the portfolio information.
Stock Data Observer	D	Keeps observing the up to date stock prices and information. Gets data from the controller through the external interface
UI Manager	D	Manages the various user interfaces. Interacts with the controller for information from database, to obtain

		external data etc.
Authenticator	D	Responsible for authenticating the user based on the login information provided.
Error Interface	D	Displays appropriate error messages to the users.
Advertiser Interface	D	Interface for advertiser actor.
Controller	D	Coordinates information between concepts and events based on use cases. It instructs the database, obtains information from the database and gives it to the different interfaces.
External Interface	D	Interacts with the external systems like Yahoo to get the data.
External Database	K	External data from Yahoo or any other external system
Email Interface	D	System Interface for email system
Mobile Interface	D	System Interface for SMS system
DB Manager	D	Interface to database used to retrieve and store data
Stock Market Database	K	Stores stock market information
Transaction Database	K	Stores information about transactions
User Database	K	Stores information about the users

Type D: Doing, Type K: Knowing

ii. Association Definitions

Concept	**Relation [direction: ->]**	**Concept**
Controller	Communicates	UI Manager
Controller	Communicates	Email Interface
Controller	Communicates	Mobile Interface
Controller	obtains information	Portfolio
Controller	Instructs	DB Manager
UI Manager	Invokes	User Interface
UI Manager	Invokes	Admin Interface
UI Manager	Invokes	Advertiser Interface
UI Manager	Invokes	External Interface
UI Manager	Communicates	Controller
Stock Data Observer	obtains information	Controller
Authenticator	obtains information	Controller
User Interface	invokes	Login Page
User Interface	invokes	Buy/Sell Page
User Interface	invokes	Home Page
Admin Interface	invokes	Login Page
DatabaseManager	forwardsRequest	RetrieveStockData
DatabaseManager	forwardsRequest	RetrieveStockChar
Login Page	obtains information	Login
Login Page	authenticates using	Authenticator
Trade Page	Performs	Place An Order
Place An Order	Performs	Buy A Stock
Place An Order	Performs	Sell A Stock
Buy A Stock	Validates	Is Order Valid
Sell A Stock	Validates	Is Order Valid
Buy A Stock	Performs	Place Market Order
Buy A Stock	Performs	Place Limit Order
Buy A Stock	Performs	Place Stop Order
Sell A Stock	Performs	Place Market Order
Sell A Stock	Performs	Place Limit Order
Sell A Stock	Performs	Place Stop Order
Authenticator	Error	Error Interface
Buy A Stock	Error	Error Interface
Sell A Stock	Error	Error Interface
Email Interface	Error	Error Interface
Mobile Interface	Error	Error Interface

iii. Attribute Definitions

Concept	Attribute/ Definition
Administrator Interface	

UserLogin	
LoginKey	1. username – Username provided by the users 2. password – Password provided by the user
Invester Home page	
Trade Page	
PlaceOrder	1. orderType – Type of order that is either buy or sell 2. noOfShares – number of the shares to be bought or sold 3. status - Indicates whether the buy/sell was successful
BuyAStock	1 status – indicates whether buy was successful (the same is communicated in upward direction)
SellAStock	1 status – indicates whether sell was successful(the same is communicated in upward direction)
ViewPortfolio	1. user – user for whom the portfolio is created 2. portfolioData - other data pertaining to portfolio
Check Order Valid	
Place Market Order	
Place Limit Order	
Place Stop Order	
UIManager	
Authenticator	1. success – set to 1 if the authentication succeeded
Error Interface	1. error – holds the error message
Advertiser Interface	
Controller	
External Interface	
Email Interface	1. emailId – email id of the user 2. text – the text to be emailed
Mobile interface	1. mobileNumber – mobile number 2. text – text to be messaged

DB Manager	1. updateTime – time when the update was performed 2. updateUser – user who performed the update 3. updateStatus – Indicated if the update was successful

b. System Operation Contracts

Operation:	Create New Account
Preconditions:	➢ The account name does not exist
Postconditions:	➢ A new user account created ➢ An amount of virtual money is added to the new account

Operation:	Buy Stocks
Preconditions:	➢ An investor is already logged in ➢ The investor has sufficient funds
Postconditions:	➢ Update the investor's portfolio

Operation:	Sell Stocks
Preconditions:	➢ An investor is already logged in ➢ The investor has sufficient number of shares of the stock
Postconditions:	➢ Update the investor's portfolio

Operation:	View Portfolio
Preconditions:	➢ An investor is already logged in
Postconditions:	➢ Display investor's portfolio information

Operation:	View Stock Detail Information
Preconditions:	➢ An investor is already logged in ➢ The entered content matches a stock name in database
Postconditions:	➢ Display the desired stock information

Operation:	Manage User Accounts
Preconditions:	➢ A system administrator is already logged in ➢ The desired account information exists
Postconditions:	➢ The account reflects the change based on the administrator's operation

Operation:	Manage Webpage
Preconditions:	➢ A system administrator is already logged in
Postconditions:	➢ The webpage reflects the change based on the administrator's operation

Operation:	Email Notification
Preconditions:	➢ An operation or an event is triggered
Postconditions:	➢ An email notification is sent to the corresponding user based on an event or operation

Operation:	Mobile Device Notification
Preconditions:	➢ An operation or an event is triggered
Postconditions:	➢ A text message is sent to the corresponding user's mobile device based on an event or operation

Operation:	View Leaderboard
Preconditions:	➢ An investor is already logged in
Postconditions:	➢ Display the leaderboard

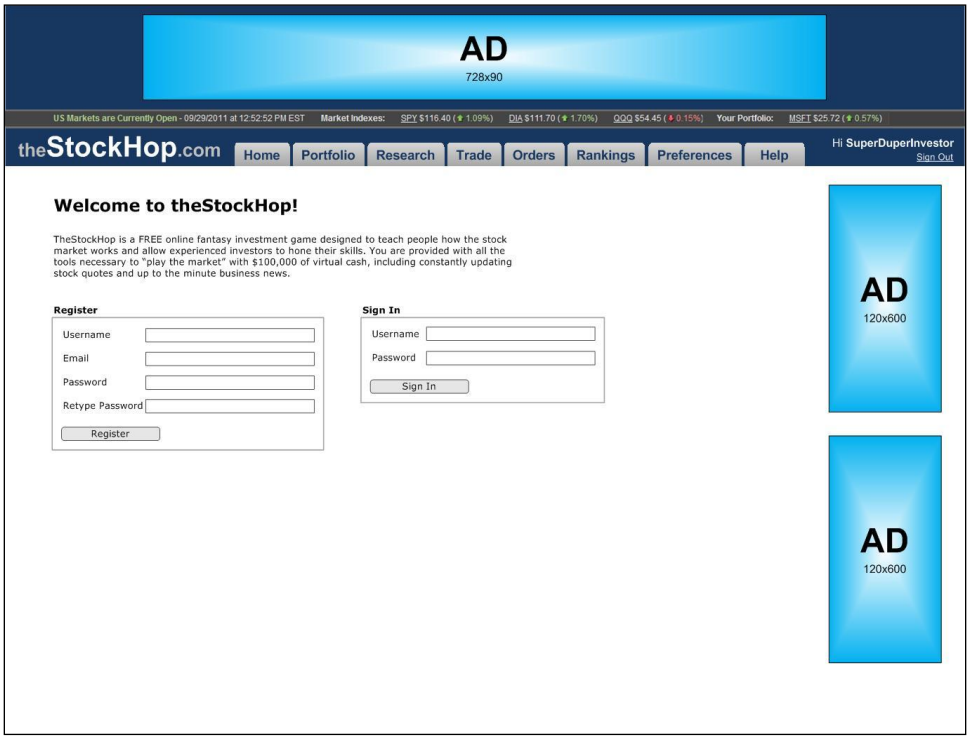
VII. User Interface Design

a. Preliminary User Interface Design

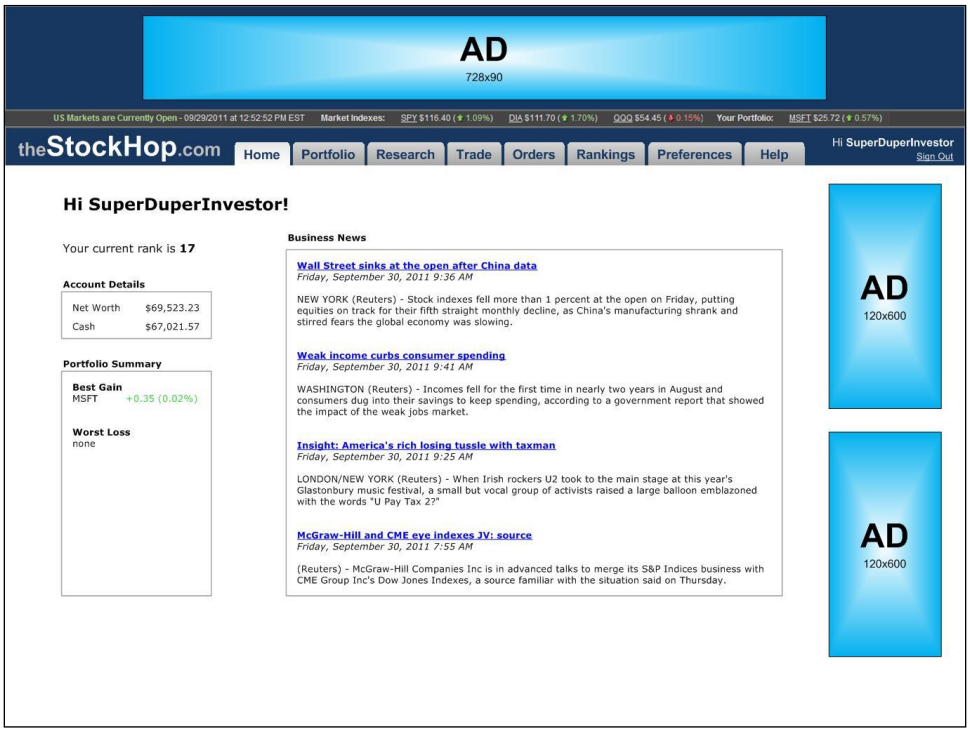
The site has been designed with simplicity and intuitiveness in mind. To that effect, we have attempted to minimize the number of mouse-clicks necessary to complete any desired action and also organized all relevant information in a manner which avoids the user needing to "flip" back and forth between different sections of the website.

The site is comprised of the following pages:

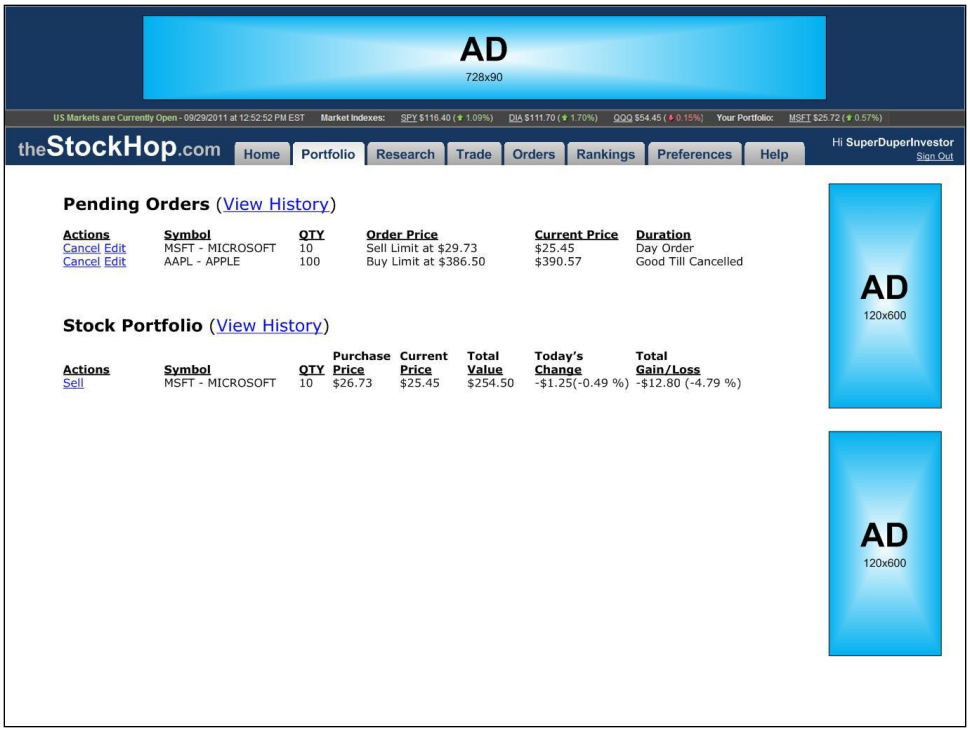
1. Welcome – default page seen when a new or not yet logged in existing user accesses the website at http://www.stockhop.com. Here a new user may register or an existing user may sign in.



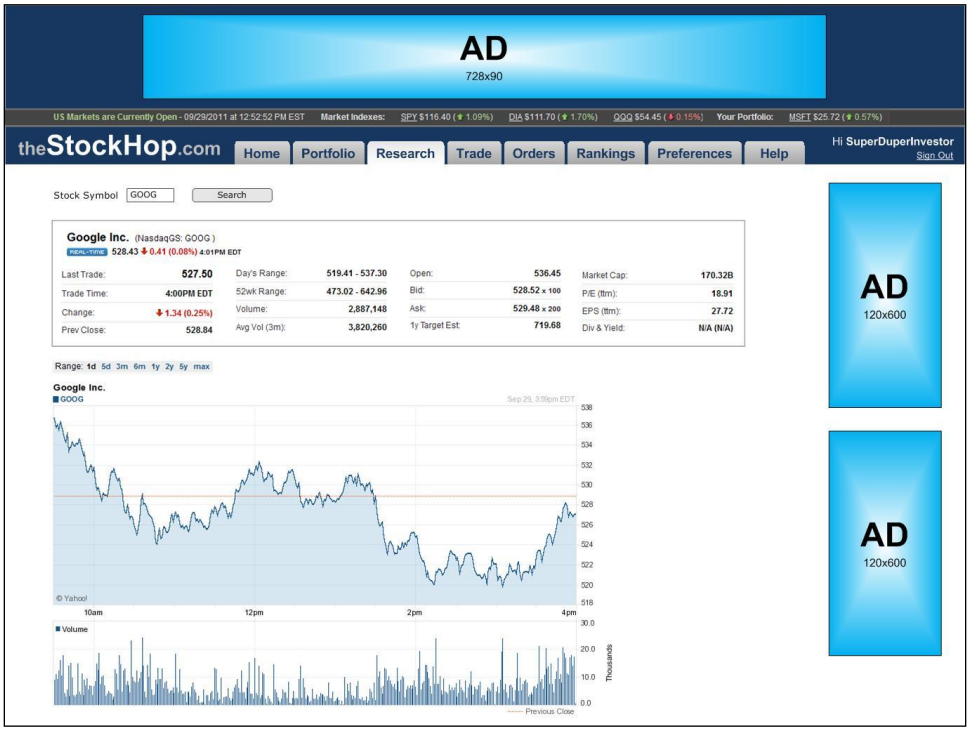
2. Home – individual user's home screen containing the user's current rank, a brief performance summary of their portfolio showing the best/worst stocks they currently own, and a news feed showing current business news. Clicking on a news feed title will take the user to a third-party website that has the full article.



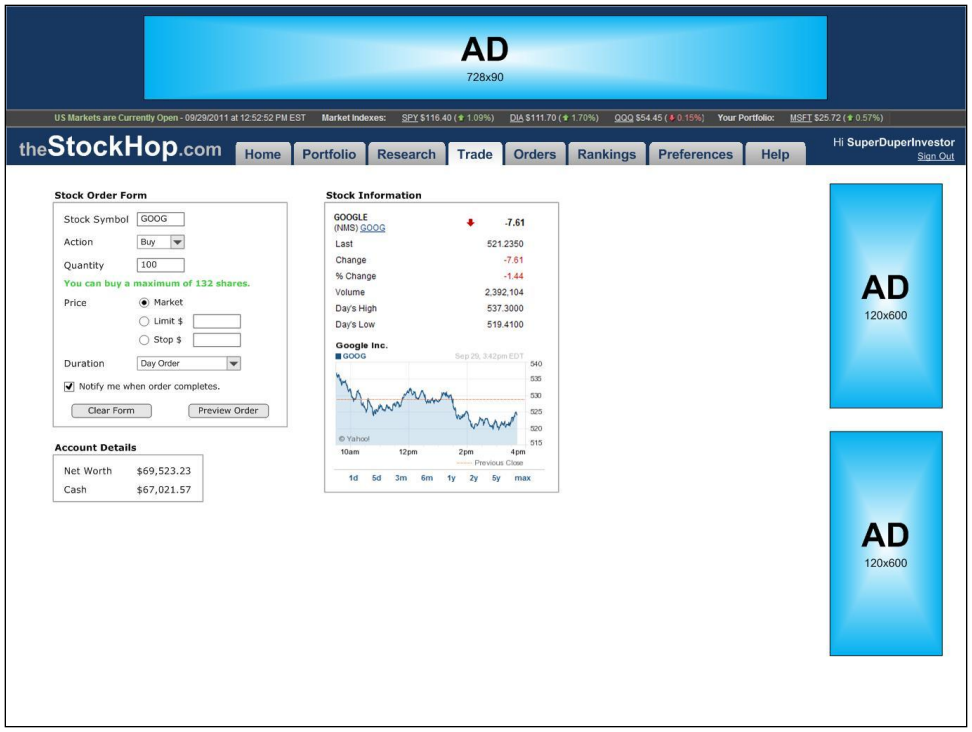
3. Portfolio – shows the pending transactions and the stocks the user currently owns. The user can cancel or edit individual pending transactions from this screen. The user can also click "Sell" next to a currently owned stock to automatically fill out a sell order.



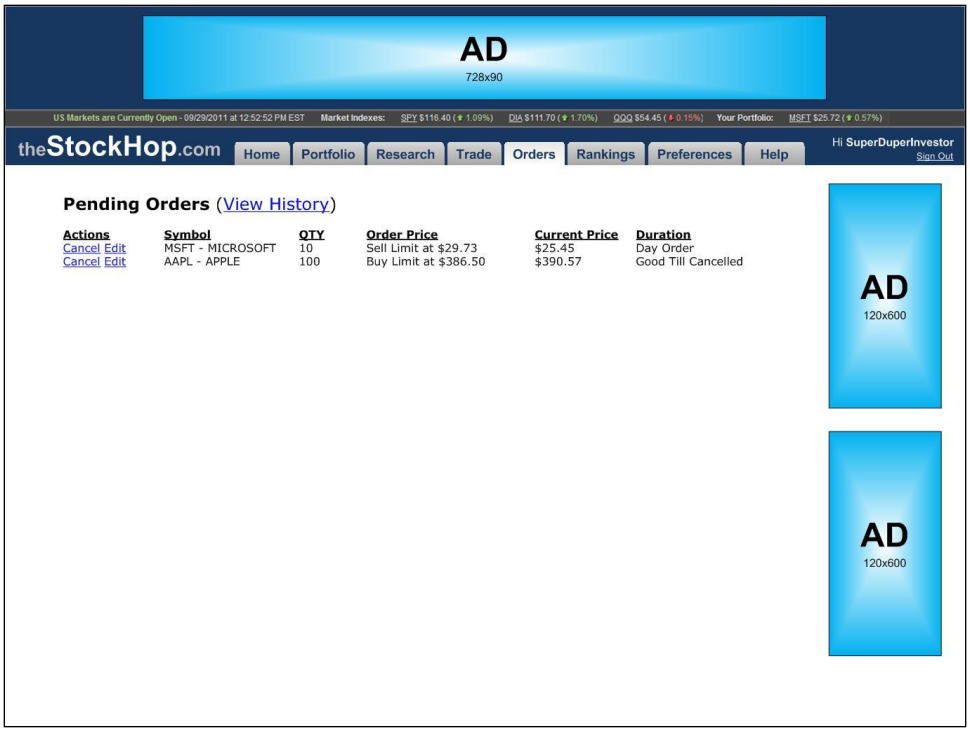
4. Research – used for researching and viewing detailed information about a particular stock. Stock information is searched by stock symbol.



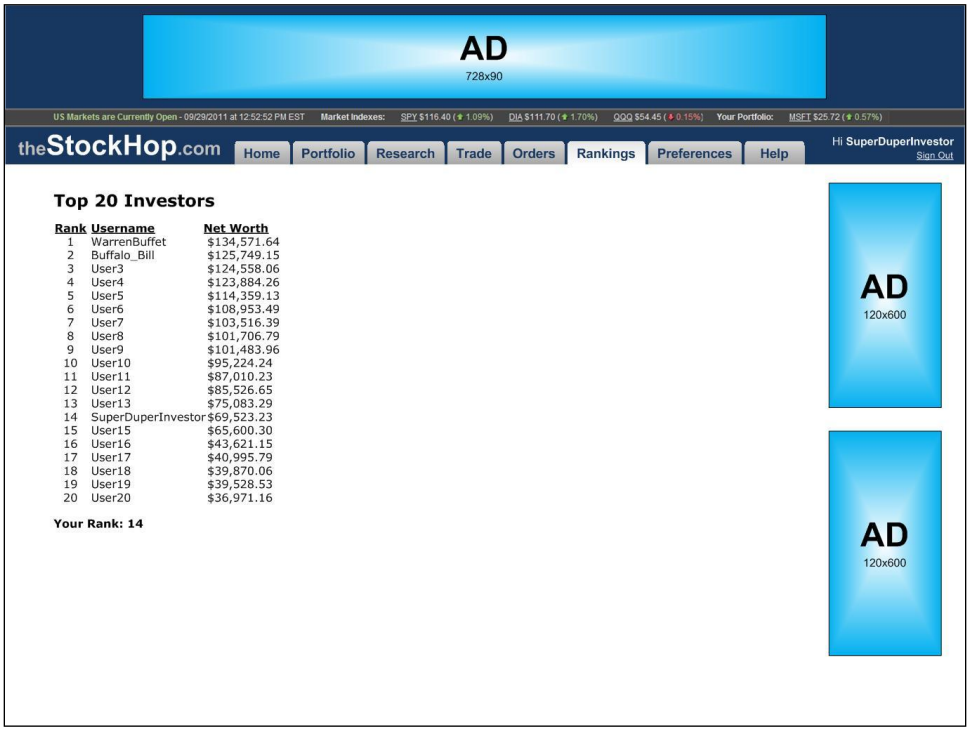
5. Trade – buy/sell orders are made from this screen. They are then previewed and confirmed via a popup dialog.



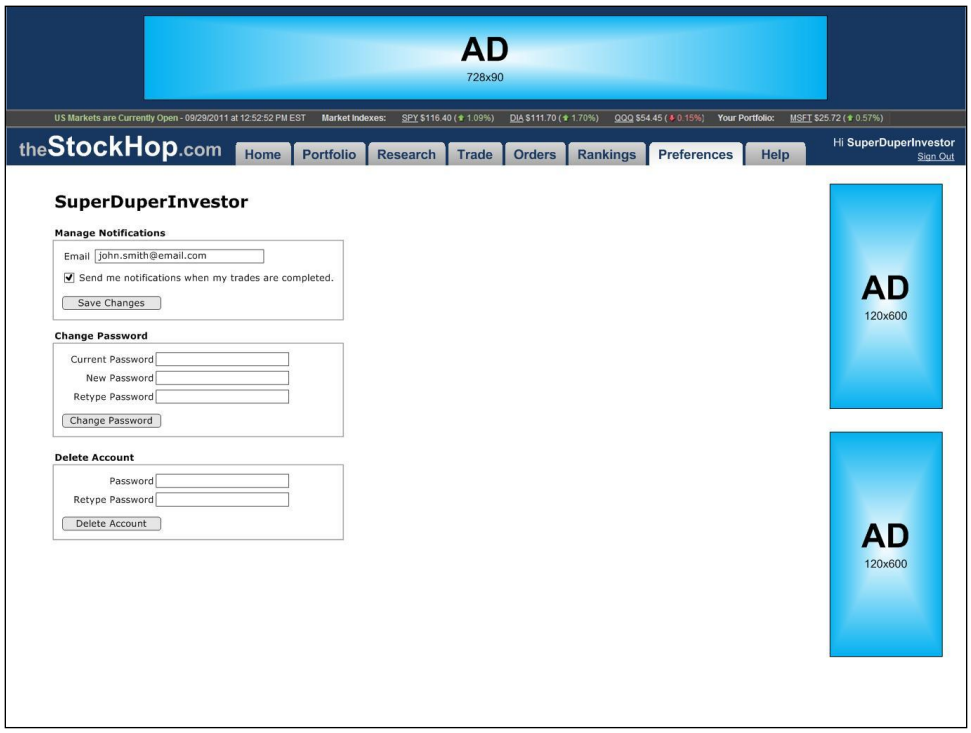
6. Orders – shows the currently pending buy/sell orders that have not yet been executed. The user may cancel an order from this screen.



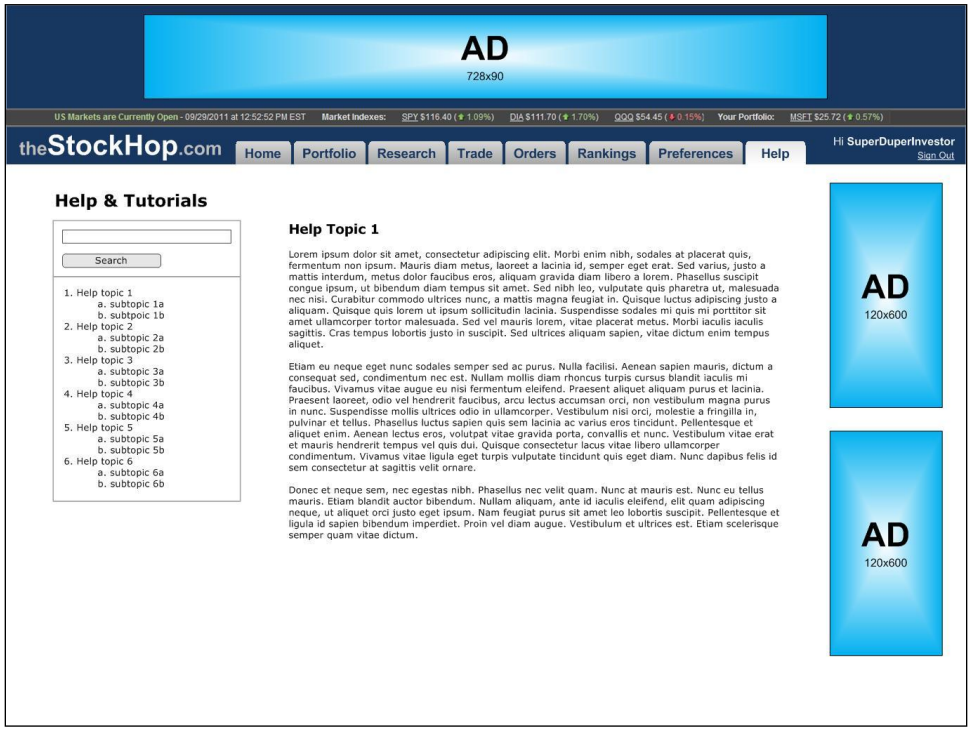
7. Rankings – shows the top 20 users based on net worth (cash + portfolio value) and the current user's rank.



8. Preferences – used to edit the notification email address, enable/disable notifications about stock order completions, change the user's password, and permanently delete the user's account.

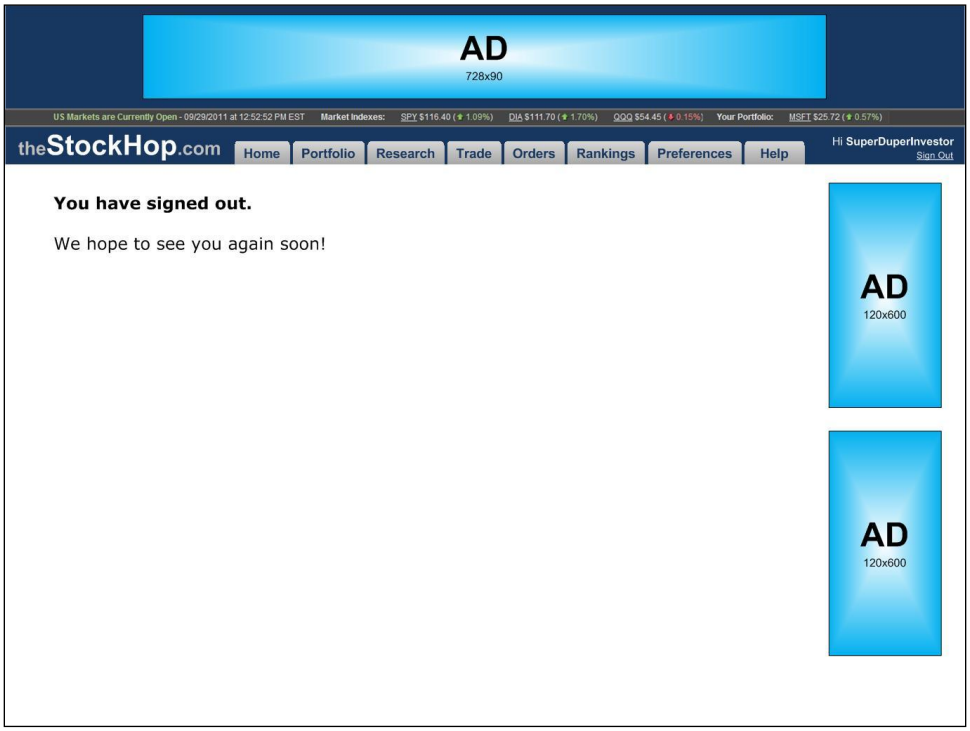


9. Help – contains help and tutorial information on stock market basics and how to use theStockHop website



10. History – shows stock order transaction history

11. Sign Out – displayed to confirm the user has logged out successfully.



All the navigational paths below assume the user has already reached the website at http://www.thestockhop.com. Whether by typing out the address in their browser's URL address bar, clicking on a link on another website or search engine, or clicking on a shortcut/bookmark they may have on their computer.

Welcome → Register → Home
Welcome → Sign In→ Home

If the user was still signed in from a previous session, he/she is taken directly to their Home screen.
Once signed in, the user can access the following sections from any page on the site.

… → Home
… → Portfolio
… → Portfolio → History
… → Portfolio → Trade
… → Research
… → Trade
… → Orders → History
… → Rankings
… → Preferences
… → Help
… → Sign Out

All the pages are also designed to show stock market relevant advertising in visible yet not obtrusive places as indicated by the "Ad" placeholders in the preceding mockup screenshots. There is also a stock ticker showing the current market indices as well as the stock prices of stocks currently in the user's portfolio.

b. User Effort Estimation

Here are a few key usage scenarios presented in GOMS (Goals, Operators, Methods, and Selection rules) format to analyze the type and amount of actions a user has to perform in order to achieve particular goals. These are only a small sample of the many possible scenarios, but they give a good idea of the overall level of interactivity.

All methods below assume the user has already reached the main page of the website at http://www.thestockhop.com. Whether by typing out the address in their browser's URL address bar, clicking on a link on another website or search engine, or clicking on a shortcut/bookmark they may have on their computer.

Note: Precise keystroke counts are omitted because the number of keystrokes depends on the length of the data the user inputs (i.e. username, password, number of stocks to buy, etc.).

Method for goal: Create a New Account (2 mouse clicks, variable keystrokes)

1. Move cursor to Username text field in Register area on page and click mouse
2. Enter username in text field using keyboard
3. Press “Tab” to move to Email field
4. Enter email address in text field using keyboard
5. Press “Tab” to move to Password field
6. Enter password in text field using keyboard
7. Press “Tab” to move to Retype Password field
8. Enter password in text field using keyboard
9. Move cursor to Register button in Register area on page and click mouse
10. Return with goal accomplished

Method for goal: Sign In (2 mouse clicks, variable keystrokes)

1. Move cursor to Username text field in Sign In area on page and click mouse
2. Enter username in text field using keyboard
3. Press “Tab” to move to Password field
4. Enter password in text field using keyboard
5. Move cursor to Sign In button in Sign In area on page and click mouse
6. Return with goal accomplished

Method for goal: Sign Out (1 mouse click)

1. Move cursor to Sign Out link on page and click mouse
2. Return with goal accomplished

Method for goal: Buy Stock at market price with day order (5 mouse clicks, variable keystrokes)

1. Accomplish goal: Sign In
2. Move cursor to Trade tab on page and click mouse
3. Move cursor to Stock Symbol text field on page and click mouse
4. Enter stock symbol in text field using keyboard
5. Move cursor to Quantity text field on page and click mouse
6. Enter stock quantity in text field using keyboard
7. Move cursor to Preview Order button on page and click mouse
8. Move cursor to Confirm Order button on popup dialog and click mouse
9. Return with goal accomplished

Method for goal: Research Stock (2 mouse clicks, variable keystrokes)

1. Accomplish goal: Sign In
2. Move cursor to Research tab on page and click mouse
3. Move cursor to Stock Symbol text field on page and click mouse
4. Enter stock symbol in text field using keyboard
5. Move cursor to Search button on page and click mouse
6. Return with goal accomplished

Method for goal: Sell all shares of an owned stock at market price (4 mouse clicks)

1. Accomplish goal: Sign In
2. Move cursor to Portfolio tab on page and click mouse
3. Move cursor to appropriate Sell link on page and click mouse
4. Move cursor to Preview Order button on page and click mouse
5. Move cursor to Confirm Order button on popup dialog and click mouse
6. Return with goal accomplished

It is clear that a great deal of emphasis has been placed on the efficiency of the user interface. Many goals can be achieved in less than 5 mouse clicks. This is especially useful in situations that require immediate action, such as selling off a stock when the price is dropping steeply. An additional effect of such minimized navigational interaction is that most of the time spent by the user is on clerical data entry, such as entering a stock symbol or the number of shares via keyboard. The result is that the user can quickly get to the section of the site that he or she needs, thereby maximizing the "profitable" time spent on market related activity and minimizing the "wasted" time spent navigating the site.

IX. Plan of Work

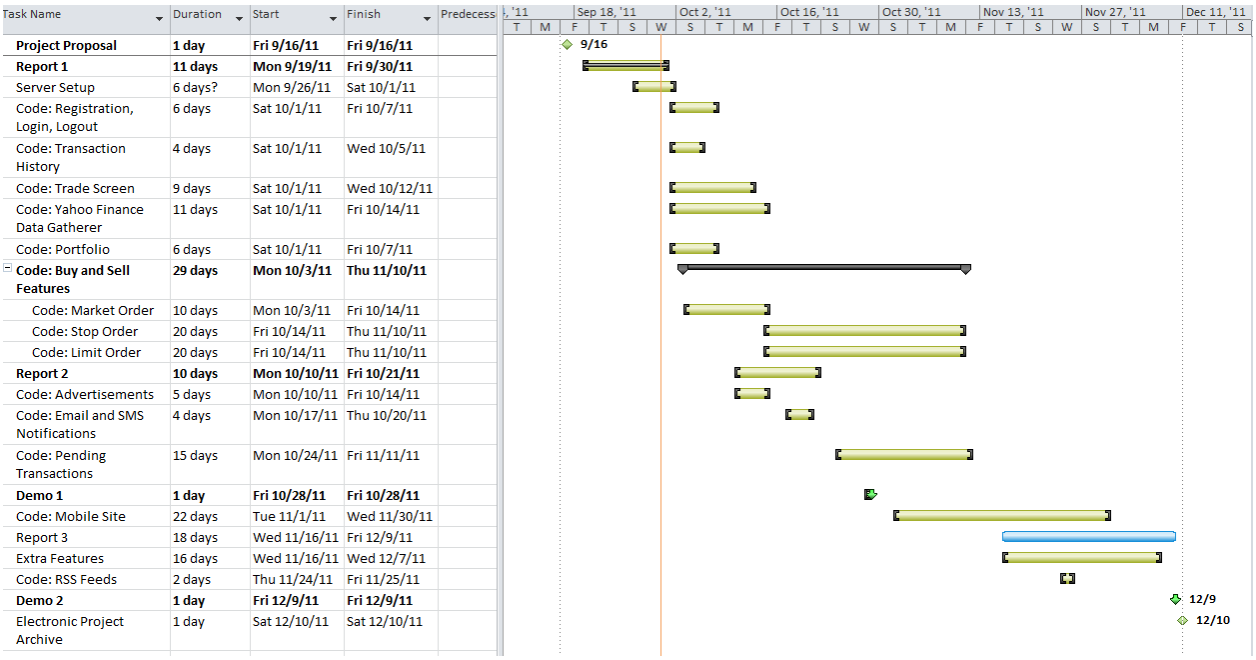
The following table shows a list of tasks and their duration, start, and finish dates.

Task Name	Duration	Start	Finish
Project Proposal	**1 day**	**Fri 9/16/11**	**Fri 9/16/11**
Report 1	**11 days**	**Mon 9/19/11**	**Fri 9/30/11**
Server Setup	6 days?	Mon 9/26/11	Sat 10/1/11
Code: Registration, Login, Logout	6 days	Sat 10/1/11	Fri 10/7/11
Code: Transaction History	4 days	Sat 10/1/11	Wed 10/5/11
Code: Trade Screen	9 days	Sat 10/1/11	Wed 10/12/11
Code: Yahoo Finance Data Gatherer	11 days	Sat 10/1/11	Fri 10/14/11
Code: Portfolio	6 days	Sat 10/1/11	Fri 10/7/11
Code: Buy and Sell Features	**29 days**	**Mon 10/3/11**	**Thu 11/10/11**

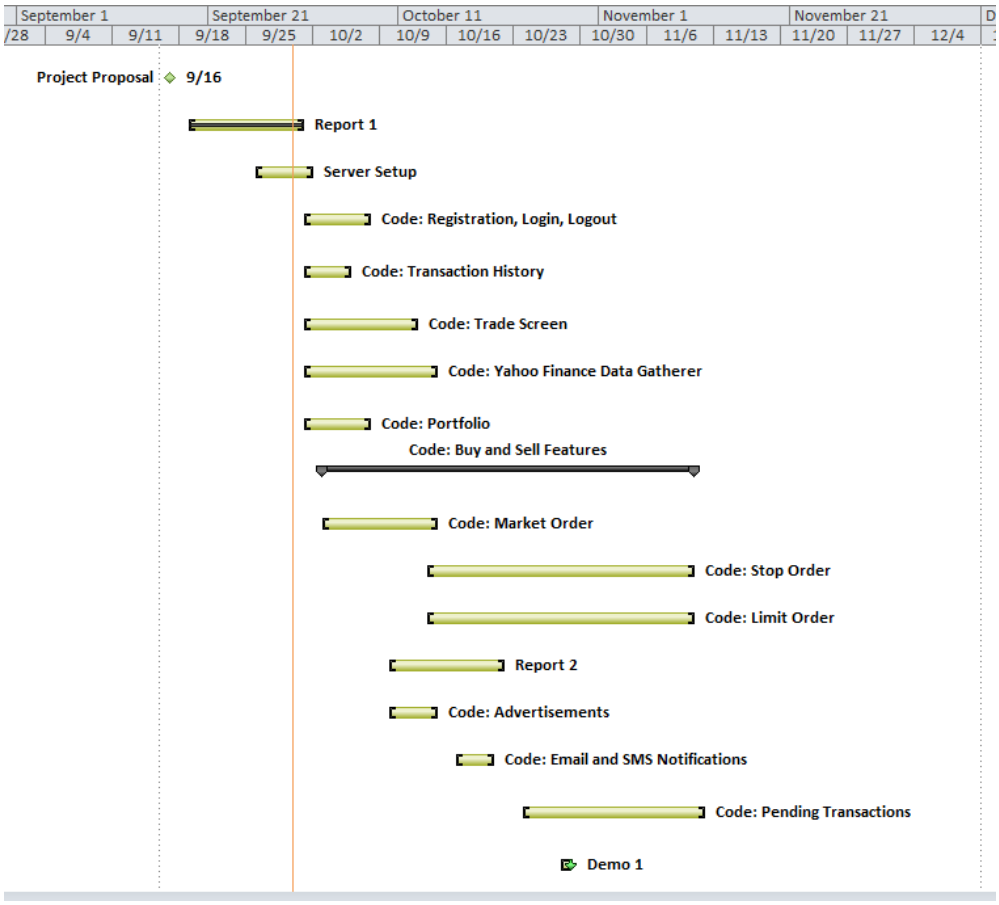
Code: Market Order	10 days	Mon 10/3/11	Fri 10/14/11
Code: Stop Order	20 days	Fri 10/14/11	Thu 11/10/11
Code: Limit Order	20 days	Fri 10/14/11	Thu 11/10/11
Report 2	**10 days**	**Mon 10/10/11**	**Fri 10/21/11**
Code: Advertisements	5 days	Mon 10/10/11	Fri 10/14/11
Code: Email and SMS Notifications	4 days	Mon 10/17/11	Thu 10/20/11
Code: Pending Transactions	15 days	Mon 10/24/11	Fri 11/11/11
Demo 1	**1 day**	**Fri 10/28/11**	**Fri 10/28/11**
Code: Mobile Site	22 days	Tue 11/1/11	Wed 11/30/11
Report 3	18 days	Wed 11/16/11	Fri 12/9/11
Extra Features	16 days	Wed 11/16/11	Wed 12/7/11
Code: RSS Feeds	2 days	Thu 11/24/11	Fri 11/25/11
Demo 2	**1 day**	**Fri 12/9/11**	**Fri 12/9/11**
Electronic Project Archive	1 day	Sat 12/10/11	Sat 12/10/11

The chart shows a lot of code that has to occur for Demo 1. Demo 1 will implement all the basic functionality, while the more difficult functionality will be implemented as time permits on the end of the project.

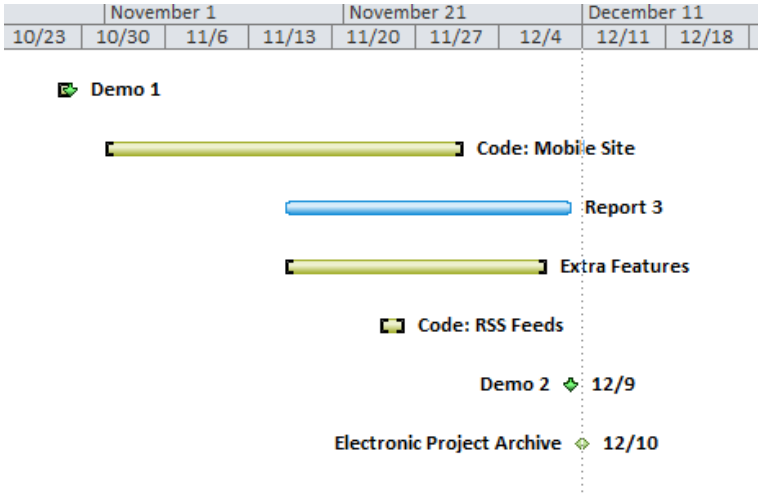
The Gantt chart was created in Microsoft Project, however, the Gantt chart image makes it difficult to view all of the details, so it was decided to take the main image and highlight the important parts. Below is the full Gantt chart, followed by two images that show the details in a clearer fashion.



Project Plan up to Demo 1:



Project Plan After Demo 1:



Breakdown of Responsibilities

Past and Current:

- Organizing Meetings: Melissa
- Project Proposal: Melissa and Jakub
- Report #1 : All Members
    - Use Cases: Sam and Dan
    - Domain Analysis: Priyanka and Wei
    - User Interface Design: Jakub
    - Other Sections: Melissa
- Server Implementation: Melissa
    - Server running off home network
    - Necessary packages installed
    - User accounts created for developers

Future:

The team split up to two groups of three. These groups were broken up based on current skillset with the languages that are to be used to code the project. Jakub, Priyanka, and Melissa will focus on the front end design. The team considers the front end design to encompass all functionality that can be done by the browser and the browser's interactions with the database or YahooFinance). Dan, Sam, and Wei will focus on the back end design. The back end design encompasses the program(s) that will run on the site host server.

- Front End Design
    - HTML Template: Jakub
    - Registration, Login, Logout: Melissa
    - Portfolio Page Functionality: Melissa
    - Trade Screen Functionality: Priyanka
    - Transaction History Functionality: Jakub
    - Advertisement Functionality: TBD
- Back End Design
    - Retrieve Data: Sam
    - Parse Data: Wei
    - Limit/Stop Order criteria checks: Wei
    - Market Order Functionality: Dan
    - Database Updates: Sam

Weekly team meetings and frequent email correspondence have been ongoing since the beginning of the course and are expected to continue. Future report responsibilities will be shared amongst the team members.

X. References

"Wall Street Survivor." *Best Stock Market Game and Investing Game for Paper Trading. Try Our Stock Simulator Stock Game and Win CASH.* Wall Street Survivor.Web. 25 Sept. 2011. <http://www.wallstreetsurvivor.com/>.

"Investopedia."*Investopedia.com - Your Source For Investing Education*. Web. 30 Sept. 2011. <http://www.investopedia.com>.

Google.com AdSense. Web. 30 Sept. 2011. <www.google.com/adsense>.

Wikipedia.org, Web. 30 Sept. 2011. <www.wikipedia.org>