Project Proposal:
Bulls and Bears

Software Engineering
14:332:452

Team 1:

David Patrzeba
Eric Jacob
Evan Arbeitman
Christopher Mancuso
David Karivalis
Jesse Ziegler

January 26, 2014

Hyperlinks:

Webapp Link
Project Repository
Reports Repository

Revision History:

Version No.	Date of Revision
v.1	1/26/2014

Contents

1 Team Profile

Team 2 would like to propose "Capital Games", a Stock Market Fantasy League. This will be a variant of Project #5 described on the course website. Further details are described in Chapter 2. This project is intended to be a proof-of-concept, infusing new and exciting innovations in web design and programming into a "classic" web application.

At this time we do not intend to nominate a Team Leader, as we believe we can divide our responsibilities evenly.

We follow with the profiles of our project team members.

1.1 David Patrzeba

David is proficient with the Java, C, and C++ languages, RESTful APIs, SQL, and is highly familiar with iterative software design and object oriented design patterns. David also has experience with Android development, relational database schema, and user experience. David will be acting as a technical lead on the project.

1.2 Eric Jacob

Eric is proficient with the Java, C, and C++ languages, and SQL. Eric also has experience with Bash scripting and Python.

1.3 Evan Arbeitman

Nick is familiar with the C++ and PHP languages, Bash scripting and Python.

1.4 Jesse Ziegler

Jesse has experience with the C and C++ languages.

1.5 David Karivalis

David is proficient with the PHP, Java language. David is familiar with C, JavaScript, and HTML/CSS. David has experience in iOS and Android development, user experience, and photoshop. David will be the UI lead.

1.6 Christopher Mancuso

Chris is familar with the C++ language.

2 Project Proposal

As mentioned previously, Team X would like to work on "Capital Games" (Project #5 on the course website). Our goal is to create a functional stand-alone Stock Market Fantasy League application to serve as a platform for executing stock market simulations. We intend to deploy to Amazon's Elastic Compute Cloud (EC2), a virtual cloud hosting provider. EC2 provides a virtual scalable machine instance on Amazon's servers, which provides a platform to expand infrastructure if necessary.

2.1 Market Models

The fundamental data modeled by stock market fantasy leagues is the stock market. There are many and varied options available to traders, the most fundamental of which are buy, sell, and short. We fully intend to support these trading operations. Supporting more advanced trading features, such as trading on margin, is also a goal.

Group 6 in 2012 implemented an interesting feature: user-defined mutual and hedge funds. They created a model in which a single user could create a portfolio in which other users could invest, and follow the gains and losses of that fund. This exposes an interesting notion, that of following the trades of other users in a given league.

We therefore propose functionality of visit-able trader profile pages for users within a given league. In this model, should be able to track the trades and portfolio performances of their peers, while not being able to execute trades on their behalf. This should promote the competition of the league.

2.2 Social Media Integration

In recent years, some groups, such as Group 3 in 2012 and Group 6 in 2011, have provided Facebook interfaces for their applications. These come in two variants: making Facebook the *de facto* portal for an application; and providing the option of using Facebook as an authentication system. We note that Group 6 failed to achieve the Facebook authentication by their deadline. Therefore, at this time we plan to strike a more conservative path and create a stand-alone website with as few external dependencies as possible.

On the other hand, incorporating modular external features is a definite possibility. We note that Group 3 created a Twitter interface for their trading application, through which users were able to "tweet" trades. Such advanced functionality is an option, but should be considered an additional feature, and as such we can consider adding this module at a later point.

2.3 Transaction Feed

Another feature inspired by Twitter and Facebook is the implementation of a user transaction news feed. A feature which could be integrated into the "social" aspect of the leagues is the ability to have a feed added to certain relevant pages, containing updates about which transactions other players are making. This could offer an interesting study as to how players react to real-time information about player positions.

2.4 Data Source

After conducting market research, we decided to use Yahoo! Finance's HTTP interface for 20-minutes-delayed financial markets data, as have groups in previous years. Though other services exist (e.g. Bloomberg, Financial Content), they are nearly all paid-subscription models. The few services which offer free or "freemium" services (e.g. eoddata) have other unacceptable limitations, such as a lack of live data. Despite its age, Yahoo! Finance API is currently still the only free, versatile finance API. This is despite its unofficial ceiling of daily requests, as we do not expect to receive sufficient volume of traffic as to approach this ceiling.

2.5 Mobile and Tablet Interfaces

One area in which we intend to differ from teams of previous years is with a unified mobile and desktop website experience. Due to recent advances in web site design, known as "responsive design", it is possible to design a single site which is accessible from mobile, tablet, and traditional web browsers. Various CSS libraries provide these designs styles, including Twitter Bootstrap and Skeleton. Additionally, enhancements in mobile browser capabilities enable the use of Javascript, which is now supported universally by all modern mobile browsers. These changes opens up availability of our team to dedicate more resources towards core site functionality.

2.6 Interactive Portfolio Graphs

Another area in which we intend to offer improvements over features provided in previous years is that of interactive portfolio graphs. Highcharts JS provides the HighStock dynamic interactive graphing API. This particular library is designed with financial modeling in mind, as it includes dynamic tooltips, time scrolling and time zooming. Thus we can present a user's entire portfolio's performance in a single object, for concise control and trend analytics. Other options include graphing propsective investments and graphing multiple investments on a single graph for comparison.

2.7 Periodic Portfolio Email Updates

A feature often presented to commercial investors is periodic performance update emails. Like Group 3 in 2011, we would like to implement an email system which will periodically inform users of their portfolio performance in various leagues, as well as any gains or losses they may have incurred since the last update. We note that Group 3 failed to model performance of their portfolios, instead simply regurgitating the day's-end values. An important improvement will be offering the ability to compare to previous trading data.

2.8 Educational Tutorial

There are two target demographics for this application: students and novice investors. As a result, we find it important to the success of our project that we include an interactive tutorial to educate our users in both the basic and more advanced aspects of investing and finance. This will greatly broaden our appeal to users and make a marked improvement over previous interactions by combining social, educational, and entertaining elements to create a rich and captivating experience.

3 Product Ownership

We have decided to divvy up the group in such a way that each function is claimed by a particular team member. In this way, we allow for easy traceability for the future reference and to determine accountability. When the customer is in need of finding the "go to" person for a specific function of the project, this will be the person to consult. Of course, this is not to say that each team member works in a vacuum. As each person's particular skills vary, one may contract out some part of his own function to another team member. In fact, as many functions will likely interface with one another, it will almost certainly be required for some cross-functional cooperation. However, it is important for the person with ownership to be very particular about the design and understanding of his function.

Each person is responsible for unit testing his own projects.

3.1 Portfolio Functionality

Nick Palumbo will be taking responsibility for the portfolio functionality of this project. One main aspect of this is the way a user is able to enact trades within their own portfolio, including basic transactions such as buy and sell as well as more advanced actions like short/cover and stop/limit. In addition, Nick will be responsible for the structure of leagues and the way in which users create and join leagues.

3.2 Social Functionality

Dario Rethage will be taking responsibility for the social functionality of this project. As this application will involve grouping users into leagues and competition, it is important for users to be able to interact with one another in the form of viewing one another's portfolios, leaderboards, and the live activity stream. Dario will be in charge of implementing these functions in a way that is intuitive and clean as well implementing a commenting/messaging system between users.

3.3 Interactive and Graphical Functionality

Val Red will be taking responsibility for the interactive and graphical functionality of this project. This includes being able to view one's own portfolio as well as the finances of his league mates in an easily maneuverable and manipulatable way. In addition, Val will be in charge of the way users follow the trends of a company's stock performance and view any other relevant data pertaining to a company, industry, or user.

3.4 Account Functionality

Eric Cuiffo will be taking responsibility for the account functionality of this project. He will be heading functions such as registration, Facebook integration, and user profile settings. Also, he will implement account management operations such as changing passwords or account deletion. In addition, he will create ways in which users can receive information about their accounts, such as downloading personal portfolio reports or the e-mail updates described above.

3.5 League Functionality

Jeff Rabinowitz will be taking responsibility for the league functionality of this project. Pertinent functions here will be the powers and actions available to the league manager of a given league, such as league settings and player management. Jeff will implement the different league settings available to a manager, as well the ability to create league-wide announcements and moderating comments within the league.

3.6 Administrative Functionality

Jeff Adler will be taking responsibility for the administrative functionality of this project. This essentially covers all the power available to a site administrator. He will be responsible for the ways in which an administrator can manage leagues, users, and user interactions. He will also generate the way in which site administrators can view statistics about the website, such as user count, active leagues, transactions, new accounts, popular stocks, and other relevant data that will dictate product decisions down the road.