Project Proposal
for
"Capital Games"

Proposal
Software Engineering
14:332:452

Team 2:

Jeff Adler
Eric Cuiffo
Nick Palumbo
Jeff Rabinowitz
Val Red
Dario Rethage

January 21, 2013

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 Jeff Adler

Jeff is proficient with Java, C, C#, C++, Objective-C, VB, Python, Ruby, PHP, Delphi, Pascal, Actionscript, Silverlight, Coldfusion, Eiffel, Bash, Ubercode, BeanShell, and Powershell. He is also experienced with project presentation and user experience design.

1.2 Eric Cuiffo

Eric is proficient with C++, Python, HTML/CSS, and jQuery. He is familiar with Django and Android development as well as web design.

1.3 Nick Palumbo

Nick is proficient with C and Java. He is also familiar with Ruby on Rails, HTML/CSS, Bash, and SQL, and his skills include user experience design.

1.4 Jeff Rabinowitz

Jeff is proficient with C++, and is also familiar with Ruby, Python, C#, and HTML/CSS. He also has experience with organization and management, presentation, and content layout.

1.5 Val Red

Val is proficient with HTML/CSS, PHP, and C++. He is also familiar with Ruby, Python, and Bash. He also possesses strong presentation and design skills.

1.6 Dario Rethage

Dario is proficient with Java, PHP, C, Objective-C, and HTML/CSS. He possesses strong organization and management as well as strong design skills.

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.

3 Subsystem Teams

We have decided to divvy up the group into three principle subsystem teams, with additional modules of functionality also assigned based on relevance. Because of the "MVC" logical abstraction of the web framework we intend to leverage, Ruby on Rails, we intend to section our group accordingly, with each team focusing primarily on one component. Of course teams will communicate and coordinate, as is necessary for such a large scale project. However, this gives each developer a field of expertise and responsibility.

Each team is responsible for unit testing their own projects.

3.1 Models Team

Jeff Adler and Jeff Rabinowitz will be working on the Models Team. In MVC architectures, models are abstractions which logically map database entries to programmatic objects. The Models Team will be responsible for developing, testing, and debugging the implementations of databases, object mapping, and data models.

The Models Team will also be responsible for abstracting the Yahoo! Finance API so as to be accessible to the Views Team, who will use the results to query for live data. (This may or may not include incorporating returned data into the database, to be worked out with the Views Team.) Furthermore, the Models Team will configure the EC2 service, web servers, databases, deployments, and maintenance.

3.2 Views Team

Dario Rethage and Nick Palumbo will be working on the Views Team. In MVC architectures, views are abstractions of static interfaces with dynamically loaded content. The Views Team will be responsible for authoring and formatting the HTML/CSS of our responsive web design, as well as incorporating the models to dynamically load content into a responsive design. This includes building user and league forms, site navigation bars, and logging and trading interfaces.

The Views Team will also be responsible for building class(es) to parse queried Yahoo! Finance data and translate it into the views. This includes the HighStock Javascript graphs, which require data to render, as well as any tables or charts which may contain data textually.

3.3 Controllers Team

Val Red and Eric Cuiffo will be working on the Controllers Team. In MVC architectures, controllers are abstractions of user interactions with the site, called "actions". The Controllers Team will be

responsible for defining the RESTful site design, as well as user authorization and authentication systems. (There are Rails plugins which offer these utilities.) They will also be responsible for implementing the email updates system, although designing email templates and tracking subscriptions will be handled by the Views and Models Teams, respectively.