Demonstration Specification
for
"Capital Games"

Demonstration 1
Software Engineering
14:332:452

Team 2:

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

April 7, 2013
Version: 1

Contents

		Names					
Category	Points	Jeff A	Eric C	Nick P	Jeff R	Val R	Dario R
Coding	30 Points	15%	20%	25%	20%	20%	20%
Unit Testing	10 Points	0%	20%	20%	30%	10%	20%
Integration Testing	10 Points	10%	10%	20%	10%	20%	30%
Web Design	5 Points	0%	50%	0%	0%	0%	50%
Debugging	5 Points	20%	20%	20%	20%	20%	20%
Documentation	5 Points	0%	0%	0%	50%	0%	50%
Data Collection	8 Points	0%	0%	50%	0%	50%	0%
Database Design	5 Points	25%	0%	25%	25%	0%	25%
Flyers	2 Points	50%	0%	0%	50%	0%	0%
Slides	3 Points	0%	0%	0%	0%	0%	0%
Activity Coordination	3 Points	0%	0%	50%	50%	100%	0%
Meeting Organization	3 Points	0%	0%	50%	50%	0%	0%
Website Administration	3 Points	100%	0%	0%	0%	0%	0%
Production Database Administration	3 Points	100%	0%	0%	0%	0%	0%

1 Project Program Code Writing

Most programming was shared between all members of the group. This was coordinated during group sessions held during the week of 22-26 March 2013. Specific areas of focus that particular members were involved in included the Financial API (yfinadaptor.rb) reworked mostly by Val Red and integrated into the Rails' application code by Nick Palumbo. Most of the controllers, routing, and chart generation were shared between Nick Palumbo and Jeff Rabinowitz. Dario Rethage headed orders development with Eric Cuiffo.

Yahoo! Finance API (lib/assets/yfinadaptor.rb)

While Val A. Red did much of the development on the Yahoo! Finance API, he mainly worked with an open-source ruby iteration that received stock quotes from Yahoo! Finance as a hash and returned strings containing the respective data. Much of the work that had to be done for the purposes of Capital Games was the rewriting of the code to repurpose the string back into a hash that could be invoked by our rails application. In addition, values had to be returned in the hash converted to cents and returned as an integer to be compatible with our monetary value parser. Percentages and ratios, however, were left as float values. Essentially, all financial data drawn by Capital Games was due to this code being called upon by other parts of the rails application.

2 Unit & Integration Testing

Testing is a critical part of software engineering – just as important, if not more so, than the actual programming. Without tests, how can one know that anything works as intended?

There are two types of testing we intend to perform on our project: unit and integration testing. Both live within the "spec" folder of our directory.

2.1 Unit Testing

Because we programmed our application using Ruby, we programmed our unit tests using the RSpec programming suite. RSpec provides a natural and domain-specific language for testing individual suites of functionality, both within generic Ruby, and especially within Ruby on Rails web frameworks such as ours. Our tests check the responsiveness of both individual attributes of different modules, and even that they behave in controlled manners.

2.2 Integration Testing

Integration testing is always more difficult than unit testing, because of the need to test as many interactions as possible between units. To simplify this, we will leverage the Capybara testing suite, which utilizes the Selenium Web Driver to simulate end-user interactions and test the expected output against the actual results of the interactions. Due to time constraints, we were unable to author integration tests for the application, which is something we intend to remediate promptly.

3 Webpage Design

Our web page design adapated bootstrap as well as the TEMPLATE NAME. Eric CUiffo worked tirelessly in order to integrate both to make our website as responsive as possible.

4 Requirements Testing & Debugging

Most testing was done in-house as a shared responsbility by all group members. Particularly, debugging had to be done with the Yahoo! Finance Adaptor, the controller, as well as the server. Jeff Adler predominantly dealt with the inital server debugging of Capital Games.

5 Program Documentation

6 Database Design & Maintenance

7 Data Collection

8 Brochures & Slides Preparation

All members contributed to the writing, publishing and distribution of brochures. A copy is attached in the following pages.

Slides were prepared by Val A. Red on Microsoft Powerpoint based on input by Jeff Rabinowitz. These slides are also attached in the following pages. Images were obtained over the internet and photoshpped to make the presentation look professional; an image of Professor Marsic included among them in the slide explaining who "Capital Games" is for in order to stress how universally approachable our web application is supposed to be. Essentially, for the powerpoint, we focused on the educational and responsive aspects of Capital Games to make our audience more familiar with what makes our web application a unique and attractive option.

9 Project Management

9.1 Activity Coordination

Most group activity coordination was done either electronically, online, through a Facebook Group including all members of the group or by SMS text via this Smartphone Application called "GroupMe." Meeting times and responsibilities were designated or confirmed through either of these means. Both proved very effective.

9.2 Organizing Meetings

Meetings were predominantly organized on Mondays. We met mostly at the Hill Center's iLab's. Much of our coding and report work was facilitated during these meetings, and the meetings typically resulted in our most productive cycles of work.

9.3 Web Server Administration

9.4 Production Database Administration