Team Members: Gregory Aiello, Alex Curtin, David Mayer,

Douglas Naphas, Jeff Ramspacher

Date: 05/09/2014

Version #3

TEST PROCEDURES

Contents

Overview

This document presents the test procedures for the Random Fruit software project management system.

Abstract

Random Fruit is an open-source project management site for students developing software.

Like existing tools, it allows developers to break work down into tickets, assign the tickets to each other, view and update a page capturing all the information about a ticket, search tickets, and use the information captured in tickets to generate reports. Random Fruit is tailored to the needs of student projects using earned-value reporting, adopting reporting tools likely to be useful in an academic setting and enabling coordination among teams by an instructor.

Random Fruit aims to help students manage their work while demonstrating their progress to an instructor. It tracks time budgets and time incurred for each ticket, as well as hours worked on the project overall. The reporting feature aggregates this information to produce earned-value charts showing the changes in the remaining work volume and highlighting the achievement of milestones. Students can use the system to visualize planned value, earned value, and actual value at any time.

In addition to acting as a superuser with team-member privileges over all the groups, the instructor can use the system to quickly view how the course as a whole is progressing towards its goals. The instructor can compare groups to spot potential problems before they become emergencies.

A dashboard landing page shows users a top-level view of their project's status. Markdown-enabled comments tell the story of a ticket on its view page. Users can filter and sort tickets, and save reports.

For distribution, Random Fruit's installer application places its file structure on the server of a course instructor or department at a university, where it serves a web page to student and instructor users. It is written in PHP, JavaScript, and HTML, and served by a MySQL database. It is accessed through the web.

Finding Tests in Source Code

The code for our tests is in our source code. To view this, download our Git repository from git@github.com:douglasnaphas/RandomFruit.git, or from git@babyhuey.cis.temple.edu:RandomFruit/naphas, and checkout the naphas-docs branch. Unit tests are in RandomFruit/app/tests/ and are run from the command line with RandomFruit/phpunit. Integration tests are in RandomFruit/Selenium/ and are run from Selenium IDE.

Unit Tests

We prepared unit tests for significant methods. View their code at the path described above. We used the PHPUnit framework.

Integration Tests

We used the Selenium application, which tests web-based systems, to run automated integration tests for Random Fruit. The test cases consist of a series of steps carried out in the browser by an application called Selenium IDE. The test cases run through the following use cases and fail if any step cannot be completed.

1. "Incorrect Login Test.html": Supplying an incorrect login. The test tries to login with invalid credentials and expects a rejection message.
2. "Instructor Create and Delete Course.html": Creating and deleting a course. The test fails if the course is not successfully created or if it remains after being deleted.
3. "Instructor Dash Links.html": The test checks whether the links shown on the pages presented to an instructor are correct.
4. "Student Create and Delete Ticket.html": The test checks whether a student can create a ticket and then find it, and whether a ticket remains after being deleted.
5. "Student Login Test.html": The test checks whether a student can login with valid credentials and see the landing page.

Selenium IDE is a Firefox plugin, launched just like Firebug from an icon at the top of Firefox. To run the tests, we launch Selenium IDE and open the test suite "RandomFruit Tests". We then click "Play Entire Test Suite" to run all the tests in the browser. If any fail, the integration testing fails. The total runs and total failures are reported once all the steps are tried.

Acceptance Tests

For acceptance, the system has to pass its automated unit tests, pass its automated integration tests, and pass a series of manual tests. The manual tests are a sequence of steps that cover all the interactions from the use cases. They are a series of actions interspersed with expectations. The manual tests pass if all the expectations are true.

The manual acceptance testing steps are as follows.

Index	Type	Description
1	Action	Run unit tests as described in the test procedures.
2	Expectation	Unit tests pass.
3	Action	Run integration tests as described in the test procedures.
4	Expectation	Integration tests pass.
5	Action	Go to the login page.
6	Expectation	A random fruit icon appears.
7	Action	Refresh the page.
8	Expectation	The random fruit icon changes to a different fruit.
9	Action	Enter an invalid username and password, and click "login."
10	Expectation	A rejection message appears. You are still at the login page.
11	Action	Enter the admin username and password, and click "login."
12	Expectation	The overview/dashboard page displays.
13	Expectation	An apple favicon appears in the browser's tab.
14	Expectation	The word "Instructor" displays at the top left.
15	Action	Click the "Create a Course" link on the dashnav/sidebar.
16	Expectation	The "Create a Course" modal appears.
17	Expectation	The modal has text fields for the course code and description, a date picker for the date the first week of the course ends, and a number selector for the number of weeks in the course.
18	Action	Enter "CIS 1210" in the Course Code box and "Combinatorics" in the Description box, and leave the boxes.
19	Expectation	The entered values appear.
20	Action	Click in the Start Date date picker box.
21	Expectation	A date picker appears.
22	Action	Pick the date 9/1/14.
23	Expectation	9/1/14 shows in the date box.
24	Action	Enter 2.5 in the Number of Weeks selector.
25	Expectation	It changes to 3.
26	Action	Enter 13 in the Number of Weeks selector.
27	Action	Click the Number of Weeks selector's plus button 3 times and its minus button once.
28	Expectation	15 shows in the Number of Weeks selector.
29	Action	Click "Create Course."
30	Expectation	The modal disappears.
31	Action	Click "View Courses."
32	Expectation	The View Courses page appears.
33	Expectation	CIS 1210 - Combinatorics shows. Its Active and Planning buttons are blank, and it has no projects or users.
34	Action	Click "Add Project."
35	Expectation	The add-project modal appears.
36	Expectation	It has text boxes for project name and description, and a dropdown menu for the course.
37	Expectation	CIS 1210 is in the dropdown menu.

38	Action	Enter "Rookies" as the project name, and "Verifying rook polynomials" as the description. Pick CIS 1210 as the course.
39	Expectation	The entered values show in the modal's fields.
40	Action	Click "Add Project" in the modal.
41	Expectation	The modal disappears, and the Rookies show up under CIS 1210 back on the View Courses page, with no team members.
42	Action	Add another project in the same manner as the Rookies, this time calling it "Counterspell" and giving it the description "Count the number of ways to rearrange any number of letters in any word"
43	Action	Add another project in the same manner as the Rookies, this time calling it "Hot Rod" and giving it the description "Count the number of ways to color segments of a reversible rod"
44	Action	Click the "x" next to Hot Rod to delete the project.
45	Expectation	A confirmation box appears.
46	Action	Click "OK".
47	Expectation	The page refreshes, and Hot Rod is no longer shown.
48	Action	Click "Create Course." Create a course called CIS 1 - Deprecated Introduction, with a start date of 6/2/14, that is 6 weeks long.
49	Action	Delete CIS 1 by clicking the "x" next to its name once the View Courses page refreshes.
50	Expectation	A confirmation box appears.
51	Action	Click "OK".
52	Expectation	CIS 1 is gone when the View Courses page refreshes.
53	Action	Click "Add User" at the bottom of the View Courses page.
54	Expectation	The add-user modal appears.
55	Expectation	It has a dropdown for the user and for the course.
56	Action	Add the admin to the Rookies and click Add User.
57	Action	Add the admin to Counterspell and click Add User.
58	Action	Click "Create User".
59	Expectation	The create-user modal appears.
60	Expectation	It has fields for username, password, and e-mail.
61	Action	Enter "Logan" as the username, "nagol" as the password, and "logan@temple.edu" as the e-mail.
62	Action	Click "Create User."
63	Action	Create another user with the username "Matt," the password "ttam", and the e-mail "matt@temple.edu."
64	Action	Add Logan to Rookies and Matt to Counterspell.
65	Action	Click the "View Tickets" link on the side dashnav.
66	Expectation	No link for "All Rookies Tickets" or "All Counterspell Tickets" should appear, as they are not in active mode.
67	Action	Click the "Active" and "Planning" boxes to put CIS 1210 into active and planning mode, and refresh the page.
68	Action	Click the "View Tickets" link on the side dashnav.
69	Expectation	Links for "All Rookies Tickets" and "All Counterspell Tickets" should appear.
70	Action	Click the logout icon at the top right.
71	Expectation	The logout page, which is the login page with "User admin has been logged out" displayed, appears.
72	Action	Log back in as Logan.

73	Expectation	A blank earned value chart, with date lables 9/1/14 (a week before the end of the first week of the course) through 12/15/14, and an empty table of owned tickets, should appear.
74	Expectation	The text "Student" should appear at the top left.
75	Action	Click the pad-and-pencil icon at the top right to create a ticket.
76	Expectation	The create-ticket modal appears, with fields for project, assignee, week, title, description, and planned value.
77	Expectation	The project menu should only allow selection of Rookies.
78	Expectation	The assignee menu should only allow selection of admin or Logan.
79	Action	Create a task: project Rookies, assignee Logan, week 1, title "Project abstract", description "Decide on a project and its scope", 3.5 hours planned value.
80	Action	Create a task: project Rookies, assignee Logan, week 2, title "Evaluate existing systems", no description, 5 hours planned value.
81	Action	Create a task: project Rookies, assignee Logan, week 3, title "Mock-up", no description, 11.5 hours planned value.
82	Action	Click View Tickets > All Rookies Tickets.
83	Expectation	The three created tickets should appear.
84	Action	Click on the "Title" heading.
85	Expectation	The tickets are sorted in ascending order by title.
86	Action	Click on the "Title" heading again.
87	Expectation	The tickets are sorted in descending order by title.
88	Action	Click on the filter icon at the top right of the table.
89	Expectation	Filter headings appear at the top of each column.
90	Action	Click on the Week Due filter menu, and select week 2.
91	Expectation	Ticket number 2 becomes the only ticket in the table.
92	Action	Clear the filter by selecting "All" from the Week Due filter menu.
93	Action	Enter "abstract" in the Title filter box and press enter.
94	Expectation	Ticket number 1 becomes the only ticket in the table.
95	Action	Clear the filter by clearing the Title filter box and pressing enter.
96	Action	Go back to the Overview page.
97	Expectation	An earned value graph with a planned value line sloping up to 20 hours at the 4th point, and flat thereafter, appears.
98	Action	Back on the View Tickets page, click on the Project abstract ticket.
99	Expectation	The individual ticket display page for Ticket #1 appears.
100	Action	Click to the right of "3.5" beside Planned Hours to edit the planned hours.
101	Expectation	The field become editable.
102	Action	Enter "4" in the box and press enter.
103	Expectation	Planned hours should change to 4.
104	Action	Click the Comment button.
105	Expectation	The Comment modal appears.
106	Action	Enter the comment "I have some preliminary research, shared on the project wiki", and click "Create Comment".
107	Expectation	The comment appears on the ticket.
108	Action	Click the Log Work button to log work on this ticket.
109	Expectation	The log work modal appears.
110	Action	Enter 9 hours of work taking place in Week 1 and click "Save Changes."
111	Action	Enter another work log event of 1 hour on Ticket #1 in Week 2.
112	Action	Change the week completed to 2 and click OK.

113	Action	Through the View Tickets link, navigate to Ticket #2, log 4 hours against it in Week 2 and 1 hour in Week 3, and change its week completed to 3, and log 8 hours of work in week 3 against Ticket #3. Change plan hours for Ticket #3 to 17.5.
114	Action	Go back to the Overview page.
115	Expectation	A graph with planned, earned, and actual value lines at the levels computed from the values entered should appear. The blue line should cross the green line from above, and the red line should be beneath both of them.
116	Action	Click "Save Rookies Graph."
117	Expectation	You can save the graph as an image file.
118	Action	Log out and back in as admin.
119	Action	Uncheck the box for CIS 1210's planning mode.
120	Action	Log out and back in as Logan.
121	Action	Try to change the planned hours for a ticket.
122	Expectation	Planned hours should not be editable.