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

Douglas Naphas, Jeff Ramspacher

Date: 05/09/2014

Version #1

TEST REPORT

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.

Test Environments

We tested Random Fruit in multiple environments. Our primary environment was on our local machines. We used a virtualization system called Vagrant (www.vagrantup.com) to synchronize our development environments. We committed our Vagrant configuration file to version control. Vagrant used this file to make sure that our local development environments were always the same as one another's.

To test the web application locally, we reserved port 8888 on localhost for the virtual machine managed by Vagrant. We accessed the application by going to localhost:8888/RandomFruit.

We also deployed Random Fruit to babyhuey.cis.temple.edu/RandomFruit. This is intended to be the production environment.

Finding Test Results in Source Code

Some test output is in our source code repository. 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. Test results are in RandomFruit/app/tests/. unit_test_results_05-09-14 is the full output from the unit tests. This is verbose, since we echoed objects at times in the unit tests during debugging. integration_test_results_05-09-14_1 is the text output from the integration tests. More results are given below.

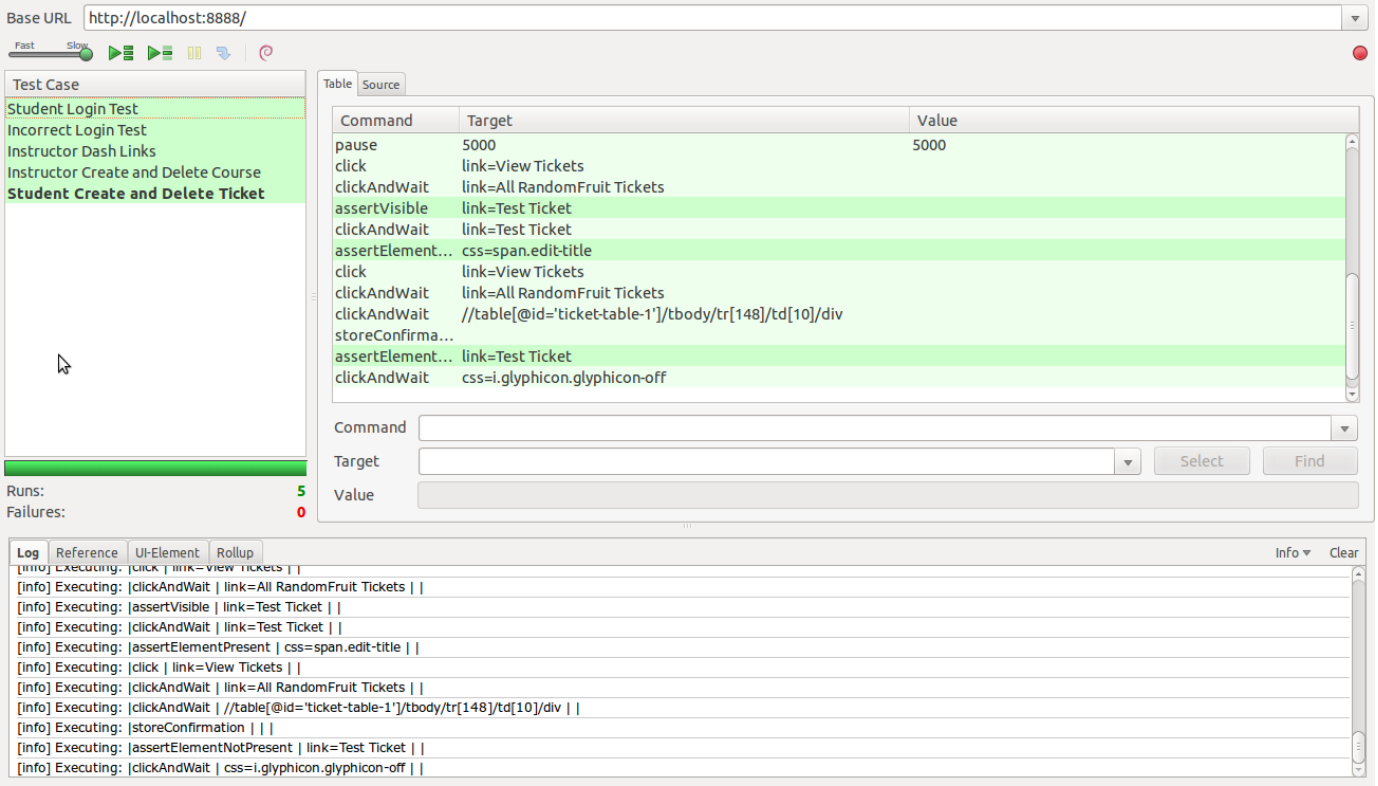
Unit Test Results

Concise output from the unit tests is as follows. "." indicates a test passing; "F" would indicate failure. All unit tests passed. We used the unit tests extensively as we wrote our methods.

```
PHPUnit 4.0.0 by Sebastian Bergmann.

Configuration read from /vagrant/RandomFruit/phpunit.xml

.......................

Time: 48.13 seconds, Memory: 68.50Mb

OK (23 tests, 68 assertions)
```

Integration Tests

The text file referred to above from the integration tests is the debug-level log output. With the logging level set to "Error" (only show errors), there is no log output. All integration tests passed. We used integration testing for regression detection: if you could no longer do something in the browser that you could before, a change broke something. Selenium IDE's GUI output from our tests is below. All integration tests passed.



Acceptance Tests

The following are the acceptance test results. We did manual acceptance tests locally and on babyhuey.

The “Result” column indicates whether each expectation was met. “pass” means the expected behavior happened, so the expectation serves as our record of the observed test results. The manual acceptance tests served as a sample user interaction for the user manual. For further evidence that a manual acceptance test passed, see the screen captures in the corresponding section of the user manual.

All automated and manual acceptance tests passed, except one, as noted in the next section. There is a work-around.

Index	Type	Description	Result
1	Action	Run unit tests as described in the test procedures.	
2	Expectation	Unit tests pass.	pass
3	Action	Run integration tests as described in the test procedures.	
4	Expectation	Integration tests pass.	pass
5	Action	Go to the login page.	
6	Expectation	A random fruit icon appears.	pass
7	Action	Refresh the page.	
8	Expectation	The random fruit icon changes to a different fruit.	pass
9	Action	Enter an invalid username and password, and click "login."	

10	Expectation	A rejection message appears. You are still at the login page.	pass
11	Action	Enter the admin username and password, and click "login."	
12	Expectation	The overview/dashboard page displays.	pass
13	Expectation	An apple favicon appears in the browser's tab.	pass
14	Expectation	The word "Instructor" displays at the top left.	pass
15	Action	Click the "Create a Course" link on the dashnav/sidebar.	
16	Expectation	The "Create a Course" modal appears.	pass
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.	pass
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.	pass
20	Action	Click in the Start Date date picker box.	
21	Expectation	A date picker appears.	pass
22	Action	Pick the date 9/1/14.	
23	Expectation	9/1/14 shows in the date box.	pass
24	Action	Enter 2.5 in the Number of Weeks selector.	
25	Expectation	It changes to 3.	pass
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.	pass
29	Action	Click "Create Course."	
30	Expectation	The modal disappears.	pass
31	Action	Click "View Courses."	
32	Expectation	The View Courses page appears.	pass
33	Expectation	CIS 1210 - Combinatorics shows. Its Active and Planning buttons are blank, and it has no projects or users.	pass
34	Action	Click "Add Project."	
35	Expectation	The add-project modal appears.	pass
36	Expectation	It has text boxes for project name and description, and a dropdown menu for the course.	pass
37	Expectation	CIS 1210 is in the dropdown menu.	pass
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.	pass
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.	pass
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.	pass
46	Action	Click "OK".	
47	Expectation	The page refreshes, and Hot Rod is no longer shown.	pass
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.	pass
51	Action	Click "OK".	
52	Expectation	CIS 1 is gone when the View Courses page refreshes.	pass
53	Action	Click "Add User" at the bottom of the View Courses page.	
54	Expectation	The add-user modal appears.	pass
55	Expectation	It has a dropdown for the user and for the course.	pass
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.	pass
60	Expectation	It has fields for username, password, and e-mail.	pass
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.	pass
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.	pass
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.	pass
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.	pass
74	Expectation	The text "Student" should appear at the top left.	pass
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.	pass
77	Expectation	The project menu should only allow selection of Rookies.	pass
78	Expectation	The assignee menu should only allow selection of admin or Logan.	pass
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.	pass
84	Action	Click on the "Title" heading.	
85	Expectation	The tickets are sorted in ascending order by title.	pass
86	Action	Click on the "Title" heading again.	
87	Expectation	The tickets are sorted in descending order by title.	pass
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.	pass
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.	pass
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.	pass
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.	pass
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.	pass
100	Action	Click to the right of "3.5" beside Planned Hours to edit the planned hours.	
101	Expectation	The field become editable.	pass
102	Action	Enter "4" in the box and press enter.	
103	Expectation	Planned hours should change to 4.	pass
104	Action	Click the Comment button.	
105	Expectation	The Comment modal appears.	pass
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.	pass
108	Action	Click the Log Work button to log work on this ticket.	
109	Expectation	The log work modal appears.	pass
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.	pass
116	Action	Click "Save Rookies Graph."	

117	Expectation	You can save the graph as an image file.	babyhuey: fail, local: pass
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.	pass

Failure Discussion

One of our manual acceptance tests fails on babyhuey. The graphs displayed on the dashboard cannot be saved as files. The test passes locally. As a work-around, users can right-click on the desired graph and click “Save Image As” to save exactly the same image that would be generated by the button.