Test Report for Quarters

Team 6
James Anthony (anthonjb)
Wenqiang Chen (chenw25)
Carolyn Chong (chongce)
Kevin Ly (lyk2)

March 21, 2016

Contents

List of Figures

List of Tables

Revision History

Date	**Comments**
March 20, 2016	Created first draft.

1 Introduction

This testing report shows the results of both system tests and non-functional tests on the Quarters application. The system tests are reported based on each individual module. Non-functional tests included tests on usability, performance and robustness.

2 Automated Testing

[Explain use of automated testing, or explain why it was not feasible for this project. —CC] Automatic testing is being used in this project to unit test various parts of the system. The project's components are broken up in to several parts: Client side javascript components, Server side access and security.

The client side javascript is tested every week once a week on an alternate web server using QUnit, a javascript unit-testing framework. Unit tests were written for each component to ensure every method does their intended action. The unit tests are rigorously tested to ensure all exceptions are handled.

Server Side access is tested via a python web crawler. To ensure that ever page is reachable. This is ran once a week similarly to the client side tests. The web crawler also crawls through all available link in a demonstration and checks for broken links. Security also tested within the web crawler by crawling with authentication and without authentication. A predetermined set of pages can only be accessible with out authentication such as the landing page, login and registration.

The unit tests will be updated and modified as more development continues. In practice these unit tests should ensure that updates to the code base and other changes to the server end do not break the system.

3 System Tests

[Specific system tests. All tests should be fully summarized in terms of initial state, input and expected output. Tests should be named. In cases where there are many similar tests just summarize the results. Provide enough info that someone could reproduce your tests. Provide traceability to test plan by referring to test case numbers or modules. —CC]

4 System Tests

[Specific system tests. All tests should be fully summarized in terms of initial state, input and expected output. Tests should be named. In cases where there are many similar tests just summarize the results. Provide enough info that someone could reproduce your tests.

Provide traceability to test plan by referring to test case numbers or modules. —CC] In this section the test cases carried out on each individual module are described. Trivial cases for some modules are not explicitly written out but instead described at a high level. Additional details are provided when necessary.

4.1 Calendar

No.	Test Case	Initial State	Input	Expected Output	Actual Output	Result
3.1-3.4	Add event to Calendar.	Calendar page.	User selects date to add new event, enters information, clicks save.	Modal opens with fields, and closes upon save. Event is updated correctly on Calendar. The same output results if user selects existing event to modify.	As expected.	PASS
3.5	Delete event from Calendar	Calendar page.	User selects event to delete, clicks delete.	Modal opens with fields. Upon clicking delete, the modal closes and the event is removed from the Calendar.	As expected.	PASS

4.2 Maintenance Tracking

No.	Test Case	Initial State	Input	Expected Output	Actual Output	Result
4.1-4.5	Navigating to maintenance page	Quarters application.	User clicks on maintenance tab in the navigation bar	Application navigates to maintenance page, all maintenance tickets relevant to the house are shown	As expected	PASS

4.6	Delete ticket from maintenance page	Maintenance System.	User clicks on ”X” button beside the maintenance ticket, user clicks confirm when confirmation window pops up.	confirmation window will appear. upon deletion conformation, close confirmation window, and ticket is removed from the page.	As expected	PASS
4.7-4.12	Create new maintenance ticket	Maintenance System.	User clicks on ”new request”.	Modal opens with fields, and closes upon save. New ticket is added into the page	As expected	PASS

4.3 Landing Page

No.	**Test Case**	**Initial State**	**Input**	**Expected Output**	**Actual Output**	**Result**
6.1,6.2	Access login or registration.	Not logged in.	Clicks on login.	Modal opens and email and password fields appear. The same output results if user clicks on register.	As expected.	PASS

4.4 Finance

No.	**Test Case**	**Initial State**	**Input**	**Expected Output**	**Actual Output**	**Result**
7.6	Add a new bill to the house	Finance page	User clicks on “+” button, fills in all informations in modal window and click ”save”	Modal window opens with fields, upon save with all fields filled in, a list of tenants that owes the user money will be added to the page	As expected.	PASS

7.7	Mark bill as paid	Finance page	User clicks on “Paid” button beside the bill, clicks ok on the confirmation window	Confirmation window will appear, upon clicking ok, the bill will have a ✓beside it	As expected.	PASS

4.5 Notifications

Test Type: Functional, Dynamic, Manual.
Tools Used: None.
Schedule: Begin testing after the PoC Demo. Complete automated tests by Final Demo April 1.
Team Member Responsible: Wenqiang Chen.
Methodology: The main objective of notification is to remind user of events that has had happened; users should be notified immediate after the event has taken place. The testing involves one user completing different actions which generates notification and have another user related to this event receive notification.

Test Case	Initial State	Input	Output
8.1	Main page. User(A) logged in.	User(B) sends money request.	User(A) sees notification of pending payment due.
8.2	Main page. User(A) logged in.	User(A) pays user(B).	User(B) sees notification of payment completed.
8.3	Main page. User(A) logged in.	User(A) has late payment.	User(A) sees notification of late payment.
8.4	Main page. User(A) logged in.	User(A) joins a house.	Other users in that house sees notification that user(A) joined the house.
8.5	Main page. User(A)(landlord) logged in.	User(B) sends maintenance ticket(Critical).	User(A) sees notification of unresolved maintenance ticket, receives email, receives text message.
8.6	Main page. User(A)(landlord) logged in.	User(B) sends maintenance ticket(Major.)	User(A) sees notification of unresolved maintenance ticket, receives email.
8.7	Main page. User(A)(landlord) logged in.	User(B) sends maintenance ticket(Minor).	User(A) sees notification of unresolved maintenance ticket.

8.8	Main page. User(A) logged in.	User(B)(Landlord) resolves a maintenance ticket.	User(A) sees notification of resolved maintenance ticket.
8.9	Main page. User(A) logged in.	User(B) sends user(A) a message.	User(A) sees notification of unread message.
8.10	Main page. User(A) logged in.	User(B) makes a post in discussion board.	User(A) sees notification of unread post.
8.11	Main page. User(A) logged in.	User(B) replies to a post made by user(A).	User(A) sees notification of unread reply.
8.12	Main page. User(A) logged in.	User(A) leaves a house.	Other users in that house sees notification that user(A) left the house.
8.13	Main page. User(A) logged in.	User(B) adds event to Calendar.	User(A) sees notification of added post.
8.14	Main page. User(A) logged in.	User(B) deletes event from Calendar.	User(A) sees notification of deleted event.
8.15	Main page. User(A) logged in.	User(A)has event happening on day.	User(A) sees notification of event.
8.16	Main page. User(A) logged in. Notification displayed.	User clicks on Notification icon.	Notification disappears.

4.6 Bulletin Board

Test Type: Functional, Dynamic, Automated.
Tools Used: Custom Scripts.
Schedule: Begin testing after the PoC Demo. Complete automated tests by Final Demo April 1.
Team Member Responsible: James Anthony.
Methodology: A script can be used to test the process of posting on the discussion board, and commenting on existing posts.

Test Case	**Initial State**	**Input**	**Output**

10.1	No posts on bulletin board.	A post with 0 characters	Empty post is disgarded [“discarded” —DS] and not added to bulletin board.
10.2	No posts on bulletin board.	A post with n characters, where $n > 0$.	Bulletin board is updated with the post of n characters.
10.3	p posts on bulletin board, where $p > 0$.	A post with 0 characters	Empty post is disgarded and not added to bulletin board.
10.4	p posts on bulletin board, where $p > 0$.	A post with n characters, where $n > 0$.	Bulletin board is updated with the post of n characters.
10.5	p posts on bulletin board, where $p > 0$.	A comment with 0 characters on an existing post p.	Empty comment is disgarded [discarded —DS] and not added to bulletin board.
10.6	p posts on bulletin board, where $p > 0$.	A comment with n characters where $n > 0$, on an existing post p_i.	Comment is added to the list of comments associated with post p_i.

5 Non-Functional Tests

5.1 Usability

The usability of Quarters was evaluated by asking test participants to complete a pre-defined task, as well as a pre- and post-test questionnaire, as outlined in the Test Plan. The participants' performance was measured by the total time to complete the task. The average time of all participants to complete the task on Quarters was measured. Think-aloud results provided subjective feedback on the user experience of Quarters. The post-questionnaire provided subjective feedback on Quarters itself.

5.1.1 Results

Figure 1 shows the participants. This data was collected during Task 1. The task completion rate was 100% for both tasks 2a and 2b, and the average times were both less than 60 seconds. Therefore the success metric stated in the Test Plan was met for completion rate and completion time, as shown in Figure 2. Figure 3 illustrates the results from Task 3. The average response rating for each question is shown.

Participant	A	B	C	D	E	F	G	H	I	J
Type	Tenant	Tenant	Tenant	Tenant	Tenant	Tenant	Tenant	Tenant	Landlord	Landlord
Age	22	18	22	21	20	18	22	22	47	58
Gender	Male	Female	Female	Male	Male	Female	Male	Female	Male	Male
Device	Computer	Computer	Computer	Computer	Mobile	Mobile	Mobile	Mobile	Computer	Mobile
Browser	Firefox	Opera	Safari	Explorer	Chrome	Chrome	Safari	Safari	Chrome	Safari
Pre Survey	Weekly	Never	Daily	Weekly	Daily	Never	Weekly	Weekly	Daily	Never
Post Survey	Weekly	Never	Daily	Weekly	Daily	Monthly	Weekly	Weekly	Daily	Monthly

Figure 1: Task 1 Pre-Questionnaire Responses.

Task	2a	2b
Completion Rate	100%	100%
Avg. Time (s)	59.02	38.81

Figure 2: Average time for Task 2.

5.1.2 Discussion

The usability evaluation proved there were many positive aspects of the Quarters user interface. Every participant was able to complete their task, and in an efficient time, regardless of the browser or the device. The straightforward navigation of the application allowed the participants to navigate easily across the web application and communicate quickly, which is a high-level goal of the software. The questionnaire results showed that participants agreed that Quarters was easy and intuitive to use. Based on these usability results, one could infer that the design and implementation support Normans Design Principles, as discussed in the Detailed Design document. The participants of the usability test unanimously strongly agreed that they would use the Maintenance Ticketing, File Upload and Notifications features. After testing Quarters, in response to how frequently they would use Quarters, participants either did not change their mind, or said they would use it more frequently relative to what they had stated prior to testing Quarters. Several participants noted during the talk-aloud that they could see Quarters solving a lot of issues they experience in their current households. These positive test results prove that Quarters has marketability.

Quarters was not without its weaknesses though. Not every participant saw the value in using Quarters on a daily basis and not every participant would recommend Quarters to a friend. Additionally, Quarters performed poorly on questions 5 and 6, which tested the usability of the Chat feature and the Finances feature, respectively. Participants noted during the talk-aloud that they could not see a use for the Chat feature when the Bulletin Board allowed them the same functionality. Additionally, they noted that the purpose of the Finances feature was not initially clear. One landlord noted that they saw value in the File

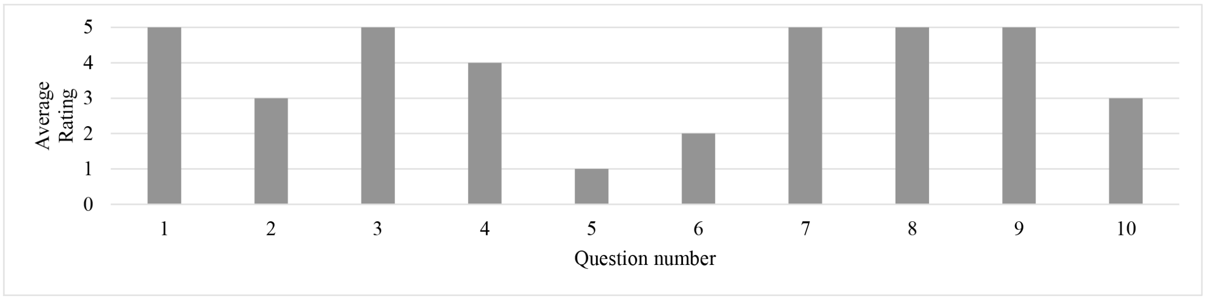

Figure 3: Task 3 Post-Questionnaire Responses.

Upload feature, but not so much in the other features. Lastly, some users with a keen eye for design noted some glitches or flaws in our interface.

Moving forward, there is room for improvement with regard to the non-functional tests. Removing the Chat feature is something to consider to ensure all of our features collectively integrate well into Quarters. Redesigning the Finances feature or adding more functionality to it may help users understand its purpose more intuitively. Devoting more time and focus to styling would help resolve any design concerns and give the interface a more polished and professional appearance. Hopefully, with these changes, more participants would consider using Quarters more frequently and recommending the application to a friend. The results of the usability test have low external validity; in future usability tests, it would be worthwhile to seek a more diverse testing population outside of a school setting, with more landlords participating. Furthermore, a more complex set of tasks for test participants could give a more accurate reading of the effectiveness and efficiency of our application.

5.2 Performance

To test the server, we will do a load testing to make sure the server can handle 100 simultaneous requests.

[How? —DS]

5.3 Robustness

To test the security of the system, including file access, failed password attempts, SQL injections, and expired sessions, we will do manual testing.

[Be more descriptive. —DS]

6 Summary of Changes

[Summarize changes made in response to testing. —CC]