

Content Testing

COS 301 ASSIGNMENT CONTENT TESTING TEAM

Duran Cole (13329414)
Kyhle Ohlinger (11131952)
Andreas du Preez (12207871)
Aluwani Simetsi (11322935)
Michelle Swanepoel (13066294)
Zander Boshoff (12035671)
Tebogo Seshibe (13181442)
Kgomotso Sito (12243273)
Thabang Letageng ()

April 2015

Github Links

Content Testing
Team A files
Team B files

version: 1.0

Contents

1	Introduction	1
2	Threads Integration	1
2.1	Threads Team A	1
2.1.1	Functional Testing	1
2.1.2	Non-Functional Testing	1
2.2	Threads Team B	2
2.2.1	Functional Testing	2
2.2.2	Non-Functional Testing	2
2.3	Threads Comparison	3
3	Status Integration	4
3.1	Status Team A	4
3.1.1	Functional Testing	4
3.1.2	Non-Functional Testing	4
3.2	Status Team B	5
3.2.1	Functional Testing	5
3.2.2	Non-Functional Testing	5
3.3	Status Comparison	6
4	Resources Integration	7
4.1	Resources Team A	7
4.1.1	Functional Testing	7
4.1.2	Non-Functional Testing	8
4.2	Resources Team B	9
4.2.1	Functional Testing	9
4.2.2	Non-Functional Testing	9
4.3	Resources Comparison	10
5	Reporting Integration	11
5.1	Reporting Team A	11
5.1.1	Functional Testing	11
5.1.2	Non-Functional Testing	12
5.2	Reporting Team B	13
5.2.1	Functional Testing	13
5.2.2	Non-Functional Testing	13

5.3	Reporting Comparison	14
-----	--------------------------------	----

1 Introduction

The purpose of this document is to provide incite into what was produced by both Content Team A (Buzz++) and Content Team B (D3). Groups were to provide an infrastructure to their low level teams and then build integration tests that would then allow the functional teams produced code to be tested and eventually added into the system. Tests were done per sub module (e.g. reporting, resource, threads and status) as to break up the integration that was to be done and provide an easier way to compare the teams work and functional code.

2 Threads Integration

2.1 Threads Team A

2.1.1 Functional Testing

Submit a Post

Expected Test output To Submit a Post the following needs to be integrated:

- User's status points needs to be incremented (Status)
- The total number of posts for that thread needs to be increased (Reporting)
- The total number of active members in a thread needs to be increased (if user is new in thread) (Reporting)
- Should be able to add a resource to a post (Resources)

Actual Test output This test was a failure for the following reasons

- Intergration team didn't use threads functional team's code(only Resources and Reporting are added).

2.1.2 Non-Functional Testing

Problem name 1 identified - e.g. Scalability this is why this is an issue with scalability, blah blah blah

Example of problem this problem is caused because of potatoes.
maybe add picture if required.

Problem name 2 identified - e.g. Security this is why this is an issue
with Security, blah blah blah

Example of problem this problem is caused because of potatoes.
maybe add picture if required.

2.2 Threads Team B

2.2.1 Functional Testing

if success:

Test One Name

Expected Test output Please write what the client expected the test
to do.

Actual Test output This test was a success
if it is not a success:

Test Two Name

Expected Test output Please write what the client expected the test
to do.

Actual Test output This test was a failure for the following reasons

- because the team didn't implement this
- because testing

2.2.2 Non-Functional Testing

Problem name 1 identified - e.g. Scalability this is why this is an
issue with scalability, blah blah blah

Example of problem this problem is caused because of potatoes.
maybe add picture if required.

Problem name 2 identified - e.g. Security this is why this is an issue
with Security, blah blah blah

Example of problem this problem is caused because of potatoes.
maybe add picture if required.

2.3 Threads Comparison

3 Status Integration

3.1 Status Team A

3.1.1 Functional Testing

if success:

Test One Name

Expected Test output Please write what the client expected the test to do.

Actual Test output This test was a success
if it is not a success:

Test Two Name

Expected Test output Please write what the client expected the test to do.

Actual Test output This test was a failure for the following reasons

- because the team didn't implement this
- because testing

3.1.2 Non-Functional Testing

Problem name 1 identified - e.g. Scalability this is why this is an issue with scalability, blah blah blah

Example of problem this problem is caused because of potatoes.
maybe add picture if required.

Problem name 2 identified - e.g. Security this is why this is an issue with Security, blah blah blah

Example of problem this problem is caused because of potatoes.
maybe add picture if required.

3.2 Status Team B

3.2.1 Functional Testing

if success:

Test One Name

Expected Test output Please write what the client expected the test to do.

Actual Test output This test was a success
if it is not a success:

Test Two Name

Expected Test output Please write what the client expected the test to do.

Actual Test output This test was a failure for the following reasons

- because the team didn't implement this
- because testing

3.2.2 Non-Functional Testing

Problem name 1 identified - e.g. Scalability this is why this is an issue with scalability, blah blah blah

Example of problem this problem is caused because of potatoes.
maybe add picture if required.

Problem name 2 identified - e.g. Security this is why this is an issue with Security, blah blah blah

Example of problem this problem is caused because of potatoes.
maybe add picture if required.

3.3 Status Comparison

4 Resources Integration

4.1 Resources Team A

4.1.1 Functional Testing

uploadResources

Expected Test output Add the uploaded resource to the list of resources, and display the list

Actual Test output Option to upload available, but not functioning (does not add resource to the list)

- Because the team didn't integrate this, implemented the function

Test removeResource

Expected Test output Remove resource and display list of resources with resource of interest removed

Actual Test output Removed resource and displayed list of resources with resource of interest removed

Test addConstraint

Expected Test output Add constraint with specified details and display list of constraints with the constraint of added

Actual Test output Added constraint with specified details and display list of constraints with the constraints of added included

Test removeConstraint

Expected Test output Remove constraint and display list of constraints with constraint of interest removed

Actual Test output Removed constraint and displayed list of constraints with constraint of interest removed

Test updateConstraint

Expected Test output Change constraint details to those specified in the field (size) and display list of constraints with the updated constraint included

Actual Test output Changed constraint details to those specified in the field (size) and displayed list of constraints with the updated constraint included

4.1.2 Non-Functional Testing

Security Data can be manipulated by unauthorized parties, that is removal of resources and/or constraints

Example of problem Lack of authentication and authorization, to confirm if a party is authorized to make changes to resources

Responsive / Performance Takes time to load pages, and response to most queries also take time

Example of problem Long server response time, lack of browser caching

Usability Users get impatient (website takes long to respond), and understanding overview of site is difficult (especially when working with resources)

Example of problem Feedback is not provided when users must wait, and website value and purpose was not provided

Quality Error messages are not provided and/or error are not fixed

Example of problem Lack of Identification and rectification of faults

4.2 Resources Team B

4.2.1 Functional Testing

if success:

Test One Name

Expected Test output Please write what the client expected the test to do.

Actual Test output This test was a success
if it is not a success:

Test Two Name

Expected Test output Please write what the client expected the test to do.

Actual Test output This test was a failure for the following reasons

- because the team didn't implement this
- because testing

4.2.2 Non-Functional Testing

Problem name 1 identified - e.g. Scalability this is why this is an issue with scalability, blah blah blah

Example of problem this problem is caused because of potatoes.
maybe add picture if required.

Problem name 2 identified - e.g. Security this is why this is an issue with Security, blah blah blah

Example of problem this problem is caused because of potatoes.
maybe add picture if required.

4.3 Resources Comparison

5 Reporting Integration

5.1 Reporting Team A

5.1.1 Functional Testing

If the use case implementation was successful:

Requirement 1

Expected functionality 1. It provides statistical information that can be used in the interface to display facts about the average user and how the logged-in user compares with the average.

Actual Test output This test was a failure, due to the fact that the files that were meant to be uploaded for Group A reporting were not added to the final github commit, thus there are no files to test.

Requirement 2

Expected functionality 2. It provides ways to gather data that is in the persistent store and present it in a format that is usable by other modules that are external to Buzz.

Actual Test output This test was a failure, due to the fact that the files that were meant to be uploaded for Group A reporting were not added to the final github commit, thus there are no files to test.

Requirement 3

Expected functionality 3. It provides functionality to alter record sets in a Buzz space by uploading the relevant information that is stored in a csv file.

Actual Test output This test was a failure, due to the fact that the files that were meant to be uploaded for Group A reporting were not added to the final github commit, thus there are no files to test.

Required Use-Cases

Get Thread Statistics Meant to provide a versatile way to get statistical information of subsets of posts complying with specified restrictions.

Get Thread Appraisal Meant to provide a versatile way to get detailed or statistical information of subsets of posts complying with specified restrictions and their associated appraisals assigned by specified members.

Export Thread Appraisal Meant to realise an off-line facility to apply a manual appraisal. It creates the dataset to be used that can be edited offline and allow updates to be inserted through the `importThreadAppraisal` function.

Import Thread Appraisal Meant to realise an offline-facility to apply a manual appraisal. It is dependent on the `exportThreadAppraisal` function

Export Thread Meant to provide means to back up the content of a thread or subset of a thread in a serialised text file.

Import Thread Meant to provide means to restore the content of a thread or subset of a thread that was stored using the `exportThread` function

Test Coverage Analysis This test was a failure, due to the fact that the files that were meant to be uploaded for Group A reporting were not added to the final github commit, thus there are no files to test. Thus there was a 0 percent completion rate for this phase of the mini project.

5.1.2 Non-Functional Testing

Problems The files that were meant to be uploaded for Group A reporting were not added to the final github commit, thus there are no files to test.

Performance Due to this, there would be a problem with performance.

Scalability Due to this, there would be a problem with Scalability.

Maintainability Due to this, there would be a problem with Maintainability.

Reliability Due to this, there would be a problem with Reliability.

Usability Due to this, there would be a problem with Usability.

5.2 Reporting Team B

5.2.1 Functional Testing

If the use case implementation was successful:

Requirement 1

Expected functionality 1. It provides statistical information that can be used in the interface to display facts about the average user and how the logged-in user compares with the average.

Actual Test output This test was successful, however it provides minimal statistical information which is only limited to threads and does not include statistics related users.

Requirement 2

Expected functionality 2. It provides ways to gather data that is in the persistent store and present it in a format that is usable by other modules that are external to Buzz.

Actual Test output This test was successful but does not present data in a format that is usable by other modules. The only functionality provided is the ability to store data.

Requirement 3

Expected functionality 3. It provides functionality to alter record sets in a Buzz space by uploading the relevant information that is stored in a csv file.

Actual Test output This test was succesfull. One is able to import that alters records and also export data.

5.2.2 Non-Functional Testing

Problems The reporting module was not integrated with the overall system. Due to this reason, the following were not able to be tested for.

Performance Module was not integrated and therefore there is no way of testing for perfomance.

Scalability Module was not integrated and therefore there is no way of testing for Scalability.

Maintainability Module was not integrated and therefore there is no way of testing for Maintainability.

Reliability Module was not integrated and therefore there is no way of testing for Reliability.

Usability Module was not integrated and therefore there is no way of testing for Usability.

5.3 Reporting Comparison