Publication Module
Level 1
UNIT & INTEGRATION TEST PLAN
Version 0.1

Mphahlele, Motsoape, Collins, 12211070
u12211070@tuks.co.za

Ayo, Benedicta, 15378731
benedictaoayo@gmail.com

Matidza, Khathutshelo, Shaun, 11072157
u11072157@tuks.co.za

Wednesday 18th May, 2016

Contents

1 Test Plan Identifier

Publication Module
Level: 1
Version: 0.1

2 References

SOLMS, F., PIETERSE V., OMELEZE S., MICHAEL, A. 2016. *Researcher Support System (RSS) Application Requirements and Design Specications (version 0.1).* Department of Computer Science, University of Pretoria.

3 Introduction

The purpose of the test plan is to test publication module for which was presented by the developers for the research support project. The master specification document version 0.1 released on the 15th march 2016. The objective of the testing the publication is the test the functionality of maintaining information of a publication throughout the circle of a research, in it has an option to be published or abandoned. The functional requirement of the preconditions for this case are:

- Detecting the actual publication.
- Checking the state of which entries captures for the state of publication.
- Checking publication in a database.
- Checking the notification alert for any of the authors.

4 Features And Functions To Test

4.1 Functional Testing

4.1.1 Pre-condition violations

4.1.1.1 addPublication

A new publication is created with its own state trace represented by a time-ordered sequence of state entries and encapsulating the initial state of the publication. A user he/she should be one of the authors of the publication, researches group leader and at least one of the authors is a member of the research group and the user must have the role of research manager to add a publication. The service is tested with the above-mentioned constraints to check if the service meets the specified requirements. The preconditions were not violated.

4.1.1.2 changePublicationState

When changing the publication state the user needs to be the author of the publication, must be a research group leader and the user has manager role. From the test, it showed that this precondition was violated.

4.1.1.3 modifyPublicationType

The user needs to be an administrator to modify the publication type which was not violated

4.1.1.4 diactivatePublicationType

The user needs to be an administrator to deactivate the publication type which was not violated

4.1.1.5 reactivatePublicationType

The user needs to be an administrator to reactivate the publication type which was not violated

4.1.1.6 addPublicationType

The user needs to be an administrator to add the publication type and the publication type with the same name must not exist in the system which was not violated

4.2 Non-Functional Testing

4.2.1 Flexibility

4.2.1.1 addPublication

The function is easily add different access channel of publication.

4.2.1.2 changePublicationState

The function is to be able to make impartation in the state of publication of a research.

4.2.1.3 modifyPublicationType

The function is flexible to adopt change in the publication type

4.2.1.4 diactivatePublicationType

The function is flexible to turn off (disengage) a publication.

4.2.1.5 reactivatePublicationType

The function is flexible to restart a publication

4.2.1.6 addPublicationType

The function is flexible to created and update publications

4.2.2 Maintainability

It should be easy to maintain the system in the future. To this end future, developers should be able to easily understand the system, the technologies chosen for the system can be reasonably expected to be available for a long time, and developers should be able to easily and relatively quickly change aspects of the functionality the system provides and add new functionality to the system.

4.2.2.1 addPublication

The function of maintaining the system to will rely on how the system is managed and cared for. It depends on the developer to

4.2.2.2 changePublicationState

This function totally depend on the maintenance of the system

4.2.2.3 modifyPublicationType

This function totally depend on the maintenance of the system

4.2.2.4 diactivatePublicationType

This function totally depend on the maintenance of the system

4.2.2.5 reactivatePublicationType

This function totally depend on the maintenance of the system

4.2.2.6 addPublicationType

This function totally depend on the maintenance of the system

4.2.3 Scalability

The network round-trip which is outside the control of the System.

4.2.3.1 addPublication

The architecture of the system is easily scaled to add publication research communities of several hundred researchers.

4.2.3.2 changePublicationState

The function of the system is easy to change publication usually within the scale

4.2.4 Performance requirements

It was very hard to measure the performance of the services provided by publication module accurately. The execution time of the code is used at this stage to measure the performance of the provided service of publication module.

4.2.4.1 addPublication

The function add only text based information so the are not any performance issues.

4.2.4.2 changePublicationState

The function add only text based information so the are not any performance issues.

4.2.4.3 modifyPublicationType

The function add only text based information so the are not any performance issues.

4.2.4.4 diactivatePublicationType

The function it is not suffering from any performance issues as it only manipulates some state flag for deactivation of publication type.

4.2.4.5 reactivatePublicationType

The function it is not suffering from any performance issues as it only manipulates some state flag for reactivation of publication type.

4.2.4.6 addPublicationType

The function add only text based information so the are not any performance issues.

4.2.5 Security

User data in every of the services provided by the publication modules it is not valid to be a legit data or to be in the bounds. Users can perform buffer overflow to try to exploit the vulnerabilities.

5 Features And Functions Not To Test

The reliability quality requirements and scalability quality requirement are not tested in this case.

6 Approach

The test personally uses the IEEE 2008 standard known as the 829 Standard for Software Test Documentation, it uses the publication and the test case specification The approach in this unit testing plan is to test the publication module of the research system for this project. The architectural specification use for this design was the virtual DOM (GitHub), JavaScript and the integration with the pm. The unit testing is executed using test code in the form of their test tools, the unit test requires a set of tools to find the bug. The tools which were use was the Java source code analyzer. And it was also used for security proposes

TEST ACCEPTANCE CRITERIA

1. Approved Functional Specification document, Use case documents must be available prior to the start of Test design phase.
2. Test cases approved and signed-off prior to starting of Test execution
3. Development completed, unit tested with pass status and results shared to Testing team to avoid duplicate defects
4. Test environment with the application installed, configured and ready to use state.

7 Item Pass Or Fail Criteria

Tests executed against the system use the functional requirements, non-functional requirements, and use cases as the factor to determine a pass or fail. The test exhibits a product pass to meet the objectives of any of the functional requirements, non-functional requirements, and the use cases.

Recommendation

- Review test plan, scripts, expected results and provide timely feedback.
- Assist in the validation of results.
- Support the development and testing processes being used to support the project.
- Conduct an investigation into execution discrepancies of the test and
- Assist test executors in a creation of accurate defects.

8 Test Deliverables

The test results from this test plan will be included in the unit/integration test summary report.

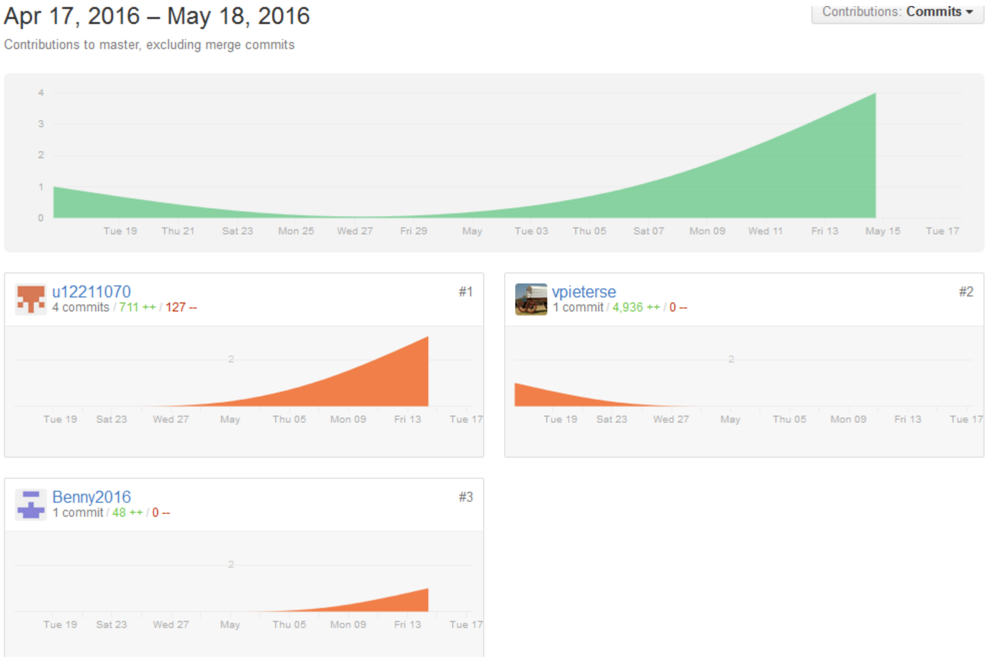
9 Test Environment

The development environment is used for all unit and integration tests.

10 Conclusion

The code was running and there was no any complication in running the code for the publication module.

11 Github



Apr 17, 2016 – May 18, 2016

Contributions to master, excluding merge commits

u12211070 #1
4 commits / 711 ++ / 127 --

vpieterse #2
1 commit / 4,936 ++ / 0 --

Benny2016 #3
1 commit / 48 ++ / 0 --

Figure 1: Contributions tab of the github profile page

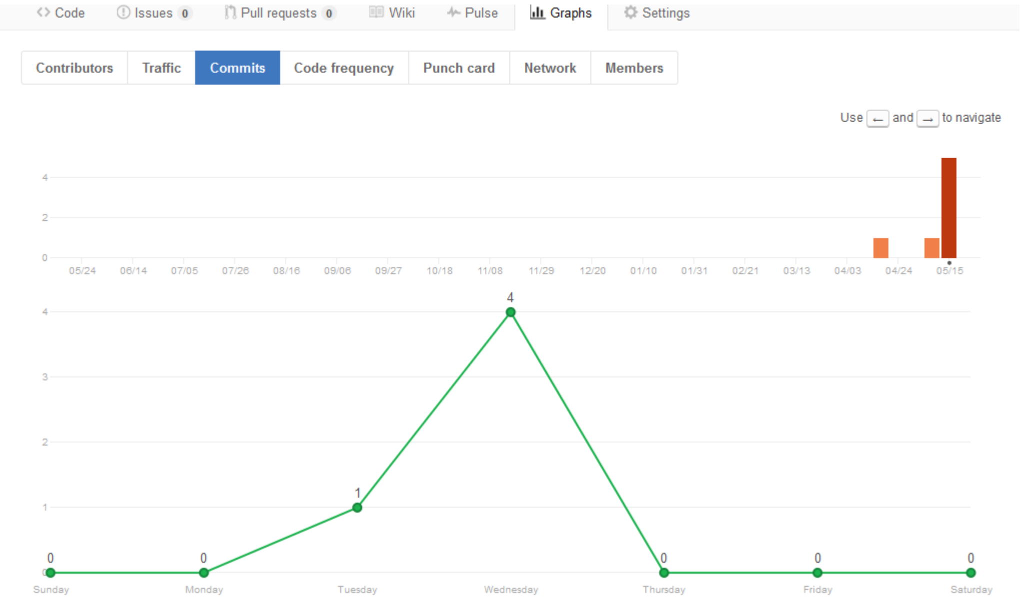
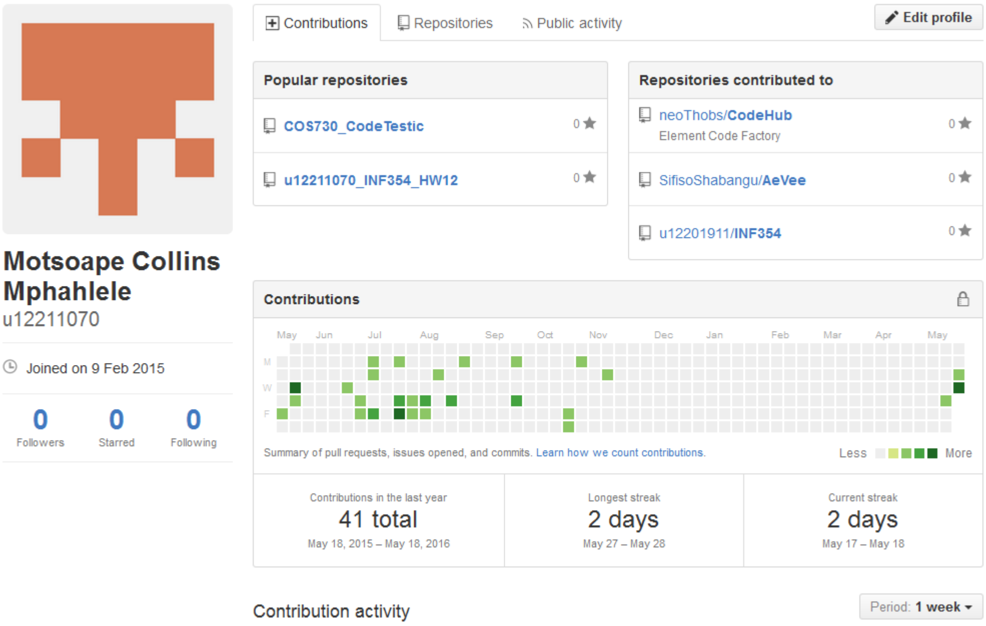


Figure 2: Commits for github



Motsoape Collins Mphahlele
u12211070

Joined on 9 Feb 2015

0 Followers 0 Starred 0 Following

Popular repositories

- COS730_CodeTestic 0 ★
- u12211070_INF354_HW12 0 ★

Repositories contributed to

- neoThobs/CodeHub 0 ★ — Element Code Factory
- SifisoShabangu/AeVee 0 ★
- u12201911/INF354 0 ★

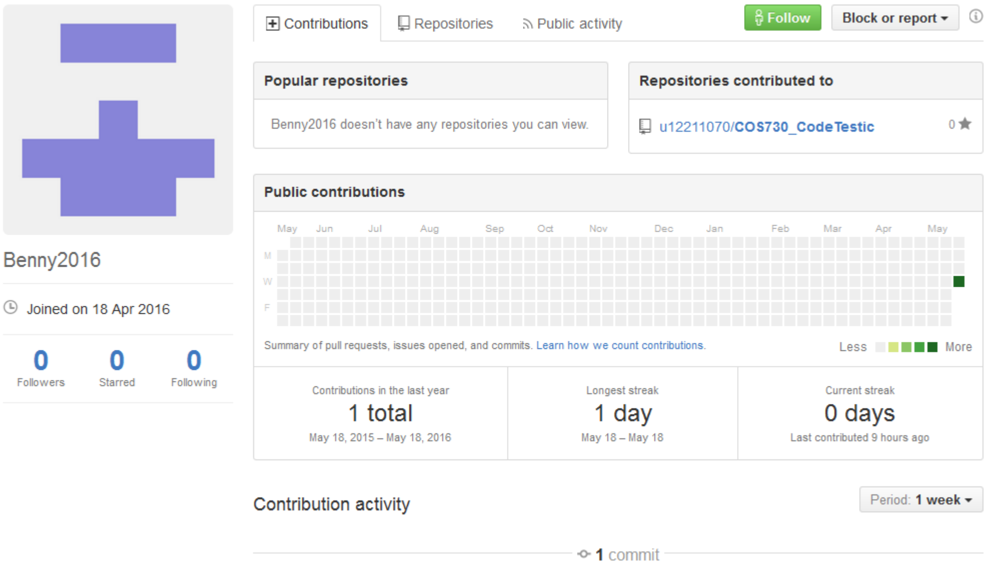
Contributions

Summary of pull requests, issues opened, and commits. Learn how we count contributions.

Contributions in the last year	Longest streak	Current streak
41 total	2 days	2 days
May 18, 2015 – May 18, 2016	May 27 – May 28	May 17 – May 18

Contribution activity

Figure 3: Mphahlele Collins Contributions



Benny2016

Joined on 18 Apr 2016

0 Followers 0 Starred 0 Following

Popular repositories

Benny2016 doesn't have any repositories you can view.

Repositories contributed to

- u12211070/COS730_CodeTestic 0 ★

Public contributions

Summary of pull requests, issues opened, and commits. Learn how we count contributions.

Contributions in the last year	Longest streak	Current streak
1 total	1 day	0 days
May 18, 2015 – May 18, 2016	May 18 – May 18	Last contributed 9 hours ago

Contribution activity

1 commit

Figure 4: Ayo Benny Contributions