UNIVERSITEIT VAN PRETORIA
UNIVERSITY OF PRETORIA
YUNIBESITHI YA PRETORIA

Web Testing

Name of project

Version: 1.0

Mering, Wynand
13230795

Watt, Brenton
14032644

Loosen, Michael
14017254

Masibi, Mothusi
12004589

Tswene, Kamogelo
12163555

Mayimela, Moses
14019702

Hlatshwayo, Nontokozo
14414555

Masemola, Sibusiso
12270467

Dunkley, Nathan
14145759

Langa, Vukile
14035449

April 23, 2016

Contents

1 Introduction

In this document we will be providing our findings on testing the individual web interfaces of teams Alpha and Bravo in regards to the COS 301 mini-project.

2 The functionality that is correctly implemented - Alpha

Due to the back-end of the project never being fully completed and integrated, it is difficult to accurately establish what was and what wasn't correctly implemented as nothing can be thoroughly tested, however, after following the readme that was placed in the repository, the website was indeed running and could be navigated.

With regards to publications, there is a link to publications itself, which comes across as an add publications page. There is a sub-field in which you can change the current life-cycle state of the publication but upon testing it was established that there was only one field to choose from but the functionality was there none-the-less.The ability to change the publication title and envisaged completion date work very well and meet requirements.

When navigating to the research page, the functionality to add, suspend or reactivate a research group exists and it is well set out and easy to navigate. Obviously due to the lack of a back-end there is no way to test if the object is actually being sent. The researcher category addition and modification pages are also implemented and are according to spec.

The overall aesthetic look of the website is good and navigation is simple and easy, nothing seemingly difficult or painful to use.

3 Which aspects of the software architecture specification were adhered to - Alpha

The architecture specifications that were followed are as follows:

- Add publication
- Change Publication state
- Organize publications according to researcher-category
- Organize publications according to research-group
- Templating

- Pre-compilation of templates
- Automatic Model population
- Virtual DOM
- Javascript is used
- integrates well with npm

The sections listed above are the sections that were testable and thus could be classified as parts of the specification that were followed.

4 Software Architecture Specifications Partially Complied To - Alpha

The technologies that were specified in the Architectural Requirements were as follows:

- Bower
- Ember CLI
- Node.js with npm
- Broccolli
- QUnit
- PhantomJS

All of which were involved in the implementation of the Alpha web interface. Thus, nothing can be said in regard to any of the Architecture Specifications only being partially complied to.

5 Which aspects of the software architecture specification were not implemented - Alpha

- Get publication
- Automatic rebuild on file changes
- Caching
- Integration with the rest of the system
- No controllers for MVC, but models and views were implemented

The features that are listed above were not identifiable on the work done. The MVC feature was only missing the controllers in order to fully follow the MVC pattern. The web was also not integrated to the rest of the system.

6 The functionality that is correctly implemented - Bravo

6.1 General Layout

The website is aesthetically pleasing and very simple to use. All available links in the navigation bar work correctly and take the user to the relevant webpages. The "Home" link takes the user to the homepage which is just a welcome page. Under the "Publications" section are two links to either add a new publication or list all current publications and both of these links work correctly. Next is a "People" section which has 3 links to either list all users, add a new user or add a new research group respectively and all these links work. The last section of the website is the "Reports" section which contains the link "Query" which generates a report for the user and this also works correctly.

6.2 Publications

The "List Publications" section correctly displays the publications that have been added to the system, however they appear to have been hard coded because after adding a publication the list remains the same as it was before adding it. For each publication in the list, there is a "View Publication" button and an "Edit Publication" button which both, after having been clicked, provide the user with the means to either view or edit the publication depending on which button they clicked. When viewing the publication, the user can choose to export it either as a CSV file or as a Bibtex file and both of these work correctly. When adding a new publication, the user can enter all the relevant details of that publication or they can choose to import the details from a CSV file which is working correctly.

6.3 People

The "List People" section displays details about the users, including which research field they are part of. For each person in the list, there is a means by which the user can view or edit their information. Both of these buttons work correctly. When editing a person, the user has the option to import the information from a CSV file which is functioning correctly. When adding a person, the user can enter all the relevant details by making use of the various fields or they can opt to once again import a CSV file. The "Adding a Research Group" section correctly provides the user to enter the details of the new research group.

6.4 Reports

This section allows the user to type in the details of the publication that a report must be generated for. The user must enter the publication type and the name of a person or research group that the publication is assigned to. After clicking the "Generate Report" button, the user is taken to a page which displays all of the research groups and the publications that they are part of.

7 Short-Comings of the implemented functionality - Bravo

There was a large problem with the integration of the back- and front-end of the web service to be provided. The web-interface provides basic functionality to move between the different sections. However, first problem encounter is the login does not work. Pre-conditions to login is that the user should be a register member of the system. The credentials provided need to be checked with the database. The post conditions is that the user should be allowed to pass or forbidden on depending on whether they are registered staff.

This problem propagates through the entire system because of the disconnect between the back- and front-end of the system. Thus most of the system falls short of what it was intended to do and contracts which where meant to be realised simply fall short due to not being able to utilize the pre-conditions and persist the post-conditions.

8 Missing Functionality - Bravo

With the back-end of the system not completed, plugged in properly to this interface, We have found that most functionality is hindered somehow, thus not realising if it successfully renders the service or not. Regarding testing for functionality that is not there, there is little to point out, as an Add Publication tab,

9 Aspects of the software architecture specification not implemented - Bravo

Firstly right off the bat, deployability was a very prominent problem. To simply get the software working was a nightmare given that there was no indication of what software frameworks where required. This makes the job of future developers hard given that they need to analyze package files to understand what exactly is needed. Also with no direction from a readme the project is a hassle to get off the ground.

Testability was the next issue. However, this is reasonable given that without proper communication with the backend, there is not much one could have tested that would be worthwhile.

Given the issues above maintainability of the project fall through the floor, if issues are not fixed present a full rewrite of the project would be in order. However, if proper organization can take place then a lot of time will be saved depending on the quality of the code after reorganizing the project.