UNIVERSITEIT VAN PRETORIA
UNIVERSITY OF PRETORIA
YUNIBESITHI YA PRETORIA

COS301 Mini Project Testing Infrastructure

Matthew Gouws *u11008602*
Andrew Parkes *u12189139*
Axel Ind *u12063178*
Patience Mtsweni *u11116774*
Khathutshelo Shaun Matidza *u11072157*
Matthew Nel *u10126229*
Ephiphania Munava *u10624610*
uxxxxxxx

Here's a link to Github.
`https://github.com/MatthewGouws/COS301_Testing_infrastructure`

Version 0.1-alpha
April 24, 2015

1 History

Date	Version	Description
21-04-2015	Version 0.1	Document Template Created
22-04-2015	Version 0.1.1	Added Authorization for B
22-04-2015	Version 0.1.2	Added Authorization for A
22-04-2015	Version 0.1.3	Added Notification Table
22-04-2015	Version 0.1.4	Added introduction
22-04-2015	Version 0.1.5	Added uses cases for Buzz B
22-04-2015	Version 0.1.6	Added uses cases for Buzz 1
22-04-2015	Version 0.1.7	Fixed Authorization formatting
23-04-2015	Version 0.1.8	Added Space use cases
23-04-2015	Version 0.1.9	Added CSDS use cases
24-04-2015	Version 0.2.0	Added availability and security non-functional requirements

Contents

2 Introduction

This document contains: Part 1 the functional testing phase for each mid level parts Buzz A and Buzz B. Each section will show the success or the failure of each part. This contains all violations of the contract requirements. Pre- and post- conditions should be tested for all the violations and the data structure requirements. For all the testing, an analysis report of the percentage cases will be given that will depict the amount of work done and the successfulness of the sections in the implementation.

Part 2 the non-functional testing phase. This part contains the performance, scalability, maintainability, reliability, usability of the application and problems associated with the system.

3 Purpose

The purpose of this task was to test functionality provided by mid-level integration for infrastructure, which consisted of Notification, Authorization, Spaces and CSDS.

4 Project Scope

The scope of the integration for infrastructure was to combine all functional teams code in a manner which could be used by top level integration. From what has been discovered and explained further in this document it shows that both teams A and B have failed to do so. Team A was very difficult to try and decipher. With missing dependencies, while Team B only had mock functionality.

5 Functional

5.1 use cases and results

5.1.1 Authorization

Use Case(s)	Buzz A	Buzz B
addAuthorizationRest - Adds an authorization restriction for a user role in a particular buzz space.	Could not NPM start as stated in README due to missing dependencies, as a result could not run	Only Mock functionality, does not run
updateAuthorizationRest - Facilitates editing of authorization restrictions.	Could not NPM start as stated in README due to missing dependencies, as a result could not run	Only Mock functionality, does not run
removeAuthorizationRest - Removes an authorization restriction for a user role from a buzz space.	Could not NPM start as stated in README due to missing dependencies, as a result could not run	Only Mock functionality, does not run
getAuthorizationRest - Retrieves the authorization restriction to enable users to select a restriction to update.	Could not NPM start as stated in README due to missing dependencies, as a result could not run	Only Mock functionality, does not run
isAuthorized - Queries the services a user may access in order to customize the UI for the user.	Could not NPM start as stated in README due to missing dependencies, as a result could not run	Only Mock functionality, does not run

5.1.2 Notification

Use Case(s)	Buzz A	Buzz B
Daily Email - Sends Daily Email.	Could not npm install, Missing dependencies, thus would not run	Only Mock functionality, does not run
Delete Notification - Checks if the user should receive a notification	Could not npm install, Missing dependencies, thus would not run	Only Mock functionality, does not run
Edit Notification Settings - Edits the notifications	Could not npm install, Missing dependencies, thus would not run	Only Mock functionality, does not run
Web Notification - returns a list of notifications for the specified user	Could not npm install, Missing dependencies, thus would not run	Only Mock functionality, does not run
Register For Notification - Allows a user to register for notifications on a thread, to specified users	Could not npm install, Missing dependencies, thus would not run	Only Mock functionality, does not run
Standard Notification - When a user adds a new thread it sends notifications to a list of registered users	Could not npm install, Missing dependencies, thus would not run	Only Mock functionality, does not run

5.1.3 Spaces

Use Case(s)	Buzz A	Buzz B
Create Buzz Space - Creates and adds the buzz space to the activated list of buzz spaces.	Could not npm install, Missing dependencies, thus would not run	
Close Buzz Space - Receives buzz to close and then removes the buzz space from the list of activated buzz spaces.	Could not npm install, Missing dependencies, thus would not run	
Assign Administrator - Gets the user to be assigned to be administrator then checks if it is administrator and adds the user to the list of administrators.	Could not npm install, Missing dependencies, thus would not run	
Remove Administrator - Receives the user to be removed then removes the user from the list of admin.	Could not npm install, Missing dependencies, thus would not run	
Is Administrator - Receives the user to be checked then searches the admin list for the user.	Could not npm install, Missing dependencies, thus would not run	
Get User Profile - Searches for the user that is queried and returns the user searched for.	Could not npm install, Missing dependencies, thus would not run	
Register On Buzz Space - Registers the user on buzz spaces and stores the user in the database.	Could not npm install, Missing dependencies, thus would not run	

5.1.4 The Buzz-Data-Sources Module

Use Case(s)	Buzz A	Buzz B
Login and administrative user - The system should authenticate against the CS DataSources within which authentication credentials are currently stored.	Gets connection to DataSource, Authenticate user against the DataSource and userID is returned.	Gets connection to DataSource, Authenticate user against the DataSource and userID is returned.
getUsersRolesForModule - This function is to query the user roles for a particular user.	Has local copy to the userID, ModuleID and roles array. It returns user roles for module request.	Has local copy to the userID, ModuleID and roles array. It *does not* returns user roles for module request.
getUsersWithRole - It retrieves all the users that have a particular role like a teachingAssistant role for a particular module.	Has local copy to the userID, ModuleID and roles array. It *does not* returns user roles for module request.	Has local copy to the userID, ModuleID and roles array. It *does not* returns user roles for module request

6 Non-functional

6.1 Performance

As stated for the B system, A would run similarly in performance due to the use of NodeJS and MongoDB, With the code very difficult to track down and run, the true performance could not be tested for the system, but given a decent entry level server would be expected to handle multiple client connections at once.

6.2 Scalability

Stub - This will be added in the future

6.3 Maintainability

Due to the fact that the code is very difficult to access in the repository for A it is highly unlikely it can be called maintainable. However unit tests seem to have been completed and thus would be easy to add specific extra cases to the code while ensuring it does not break the known working code.

6.4 Reliability

Stub - This will be added in the future

6.5 Usability

Due to the fact that the code is barely traceable to where it is located, it is very difficult to use this package. Insight into the development would be required to allow a user to be able to use the code correctly and how to use the code before production could begin.

6.6 Availability

Considering the fact that the system was going to be hosted online on successful completion, its availability would've been archived since it would be in an operable and committable state at any random time. However this is not the case with the current state of the system.

6.7 Manageability

Stub - This will be added in the future

6.8 Security

Authorization is the module that was supposed to provide the degree of security in this system. Both authorization A and B failed to run due to missing dependencies and only mock functionality provided for the other. Considering this, resistance to, or protection from, harm failed.

6.9 Monitorability and Auditability

Stub - This will be added in the future

6.10 Integrability

Stub - This will be added in the future

7 References

Stub - This will be added in the future