



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*

*uxxxxxxxxx*

*uxxxxxxxxx*

*uxxxxxxxxx*

Here's a link to Github.

[https://github.com/MatthewGouws/COS301\\_Testing\\_infrastructure](https://github.com/MatthewGouws/COS301_Testing_infrastructure)

Version 0.1-alpha  
April 22, 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

# Contents

<b>1</b>	<b>History</b>	<b>1</b>
<b>2</b>	<b>Introduction</b>	<b>3</b>
<b>3</b>	<b>Purpose</b>	<b>3</b>
<b>4</b>	<b>Project Scope</b>	<b>3</b>
<b>5</b>	<b>Functional</b>	<b>4</b>
5.1	use cases and results . . . . .	4
5.1.1	Authorization . . . . .	4
5.1.2	Notification . . . . .	5
<b>6</b>	<b>Non-functional</b>	<b>5</b>
6.1	Performance . . . . .	5
6.2	Scalability . . . . .	5
6.3	Maintainability . . . . .	6
6.4	Reliability . . . . .	6
6.5	Usability . . . . .	6
6.6	Availability . . . . .	6
6.7	Manageability . . . . .	6
6.8	Security . . . . .	6
6.9	Monitorability and Auditability . . . . .	6
6.10	Integrability . . . . .	6
<b>7</b>	<b>References</b>	<b>6</b>

## **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 violation 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.	Only mock functionality, but can not run	Only Mock functionality, does not run
updateAuthorizationRest - Facilitates editing of authorization restrictions.	Only mock functionality, but does not run	Only Mock functionality, does not run
removeAuthorizationRest - Removes an authorization restriction for a user role from a buzz space.	Only mock functionality, but can not run	Only Mock functionality, does not run
getAuthorizationRest - Retrieves the authorization restriction to enable users to select a restriction to update.	Only mock functionality, but can 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.	Only mock functionality, but can 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

## 6 Non-functional

### 6.1 Performance

Stub - This will be added in the future

### 6.2 Scalability

Stub - This will be added in the future

### **6.3 Maintainability**

Stub - This will be added in the future

### **6.4 Reliability**

Stub - This will be added in the future

### **6.5 Usability**

Stub - This will be added in the future

### **6.6 Availability**

Stub - This will be added in the future

### **6.7 Manageability**

Stub - This will be added in the future

### **6.8 Security**

Stub - This will be added in the future

### **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