



UNIVERSITEIT VAN PRETORIA  
UNIVERSITY OF PRETORIA  
YUNIBESITHI YA PRETORIA

---

DEPARTMENT OF INFORMATION TECHNOLOGY

COS 301 - SOFTWARE ENGINEERING

---

## COS 301 - Mini Project

---

*Author:*

Jason Richard Evans  
Abrie van Aardt  
Anrich van Schalkwyk  
Baruch Molefe  
Jimmy Peleha  
Johannes Coetzee  
Liz Joseph  
Maret Stoffberg  
Sebastian Gerber  
Hugo Greyvenstein

*Student number:*

u13032608  
uXXXXXXXXX  
uXXXXXXXXX  
uXXXXXXXXX  
uXXXXXXXXX  
u10693077  
u10075268  
u11071762  
uXXXXXXXXX  
uXXXXXXXXX

April 22, 2015

# THREADS TESTING REPORT

## BUZZ SPACE MINI PROJECT

Version: Version 0.1 Alpha For further references see [gitHub](https://github.com)  
*<https://github.com/thepickpocket/ThreadsTesting>*  
April 22, 2015

# Contents

1	Company A . . . . .	3
1.1	Functional Requirements. . . . .	3
1.2	Architectural Requirements . . . . .	3
2	Company B . . . . .	3
2.1	Functional Requirements . . . . .	3
2.2	Architectural Requirements . . . . .	3
3	Conclusion . . . . .	4

# 1 Company A

## 1.1 Functional Requirements.

## 1.2 Architectural Requirements

# 2 Company B

## 2.1 Functional Requirements

### SubmitPost

**Test Status:** *Partial Pass.*

**Details:** The submit post use-case was divided into separate smaller functions. One function `create(mUser, mParent, mPostType, mHeading, mContent, mMimeType)` is used to initialize the process of creating a new thread as well as its embedded post object. This function calls `createNewThread(mUser, mParent, mLevel, mPostType, mHeading, mContent, mMimeType, mSubject)` which parses the current thread and post object to JSON strings and then persists them in a remote database.

The problem however is that when child threads are created using the `createThread(...)` function call, the child thread is never persisted to the remote database.

The functional requirements also state clearly that a thread (with its related post) should have been allocated a certain space, yet this functionality is never provided.

### MarkPostAsRead

**Test Status:** *Complete Failure.*

**Details:** The team's approach to the reading events are not that well planned and clearly thought through. It simply sets a flag for the thread and persists that flag to the database. This however will then not be user based, as everyone who then accesses that specific thread from the database will have "read" it.

This function is hence not working as required and thus fails completely. It is however a nice-to-have function and does not break or affect any other part of the thread module.

## 2.2 Architectural Requirements

### 3 Conclusion