



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
u12213749
uXXXXXXXXX

April 23, 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 23, 2015

Contents

1	Company A	3
1.1	Functional Requirements.	3
1.2	Architectural Requirements	4
2	Company B	4
2.1	Functional Requirements	4
2.2	Architectural Requirements	4
3	Conclusion	5

1 Company A

1.1 Functional Requirements.

SubmitPost

Test Status: *Partial Pass.*

Details: The purpose of SubmitPost is to allow a user to submit a post to a thread, creating a new child thread.

The implemented solution passes only partially, because while the new post is submitted, it does not successfully create it's own child thread.

It also has a mimeType, post content and dateTime fields, which were required.

As this function is classified as critical and only partially works, the system does not meet all the requirements it should.

MarkPostAsRead

Test Status: *Complete Failure.*

Details: The purpose of MarkPostAsRead is to store a reading event which contains the information that a user has read a particular post at a particular time.

The implemented solution fails completely, posts are not marked as read and as this is the only requirement it fails.

As this function is classified as nice to have, it's failure does not prevent the rest of the system from working.

MoveThread

Test Status: *Partial Pass.*

Details: The main objective of the MoveThread function is to detach a sub-tree of thread nodes from one thread and add it to another thread.

The services contract MoveThreadRequest requires only one pre-condition, that the user is a administrator. The services contract was successfully implemented. This function only managed to pass partially. The MoveThread function depends on the functionality of the HideThread and UnHideThread functions. If these functions was not required during testing the thread could successfully be moved to another thread. One of the main requirements for all functions is that the function can be used on threads in the database. This is not true for this function as it only works locally. There is no functionality to retrieve the required threads from the database.

HideThread

Test Status: *Complete Failure.*

Details: The main objective of the HideThread function is that selected thread nodes and all its descendant nodes will be marked as hidden and user interfaces are meant not to render them.

The HideThread function was partially implemented. The services contract HideThreadRequest requires only one pre-condition, that the user is an administrator. The services contract was successfully implemented. The functionality of the HideThread function itself was not implemented. The priority of the HideThread function is stated as important thus the system will still function as it is not a critical function within the Thread module.

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