UNIVERSITEIT VAN PRETORIA
UNIVERSITY OF PRETORIA
YUNIBESITHI YA PRETORIA

Denkleiers • Leading Minds • Dikgopolo tša Dihlalefi

FUNCTIONAL REQUIREMENTS
COS 301 GROUP 2 B

Byron Dinkelmann u11057638
Johan van Rooyen u11205131
Mandla Mhlongo u29630135
Ryno Pierce u12003922
Sylvester Mpungane u11241617
Molefe Molefe u12260429
Taariq Ghoord u10132806
Timothy Snayers u13397134

Github

February 2015

Contents

0.1 Introduction

This document outlines the functional requirements of "The Computer Science Education Didactic and Applications Research"(CSEDAR) registered research project called "The use of Online Discussion in Teaching"(TODT). This document describes the "Buzz" system to be developed and gives insight to how to achieve this implementation for those implementing it.

0.2 Vision

The research projects aim is to find ways to enhance and improve the learning of students with the use of online discussion. This online forum "Buzz" will become part of the Computer Science websites modules, creating an online space for discussion for each or where relevant. Where students, teaching assistants and lecturers can engage in activities related to learning the content of the specific module while applying game concepts to motivate students to increase the quality of their participation and consequently experience deeper learning of the course content.

0.3 Background

The reason for this research project is the problem of engaging an extremely large number of first year students within The University of Pretoria, specifically the Computer Science Department students. Currently available tools for discussion forums lead to the problems that are hampering positive engagement of teaching staff and students. These problems consist of inexperienced users, unorganized content and low levels of excitement. To combat these problems the "Buzz" system intends to have the basic functionality of online forums, have automated feedback on common mistakes, have a Game-like presentation and have automated structuring.

The University of Pretoria have the opportunity to teach the COS301 "Software Engineering" students while being able to improve current systems of online discussion which can be extended to benefit all students in the future.

The research project gives the students of COS301 "Software Engineering" an opportunity to learn form the experiences of designing, creating and

developing this system, which include learning to work with colleagues, creating documentation, simplifying and improving the communication of lectures and students through a structured forum, using online systems for group work and the basic process of how software is designed, created and developed in business to name but a few.

0.4 Functional Requirements and application design

0.4.1 Purpose

1. CRUD: To create posts on the buzz system so that users in the system can share relevant information with one another and communicate. To allow registered users and non registered users such as guests to view posts created by users of the buzz system. To allow users to update posts of which they control. To allow users to delete their own posts or to allow users with sufficient status level to delete other posts based on certain factors about that specific post.

2. Message Restrictions: To restrict users to length, content and positioning of posts based on the users status level.

3. Message Tracker: To mark posts for each registered user so that they can keep track of which posts they have read and which users have read certain posts.

4. Check Post For Plagiarism: To use a service provider to check if a post has been plagiarised.

5. Report Post For Plagiarism: To enable users to be able to report posts that they suspect is plagiarised (with a link to the original), because the system couldn't detect it.

6. Netiquette: To ensure that every post submitted conforms with the netiquette rules.

0.4.2 Use case prioritization

CRUD

Critical

Message Restrictions

Critical

Message Read Tracker

Important

Check Post For Plagiarism

Nice To Have

Report Post For Plagiarism

Nice To Have

Netiquette

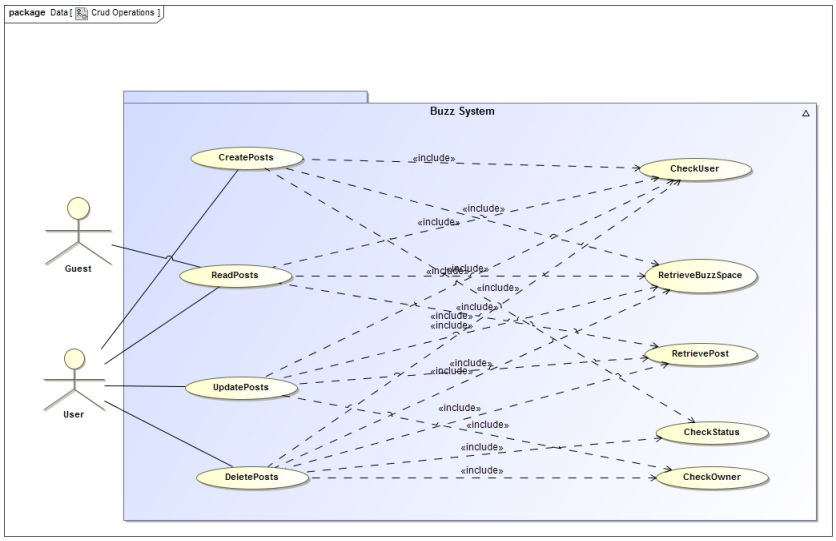
Nice To Have

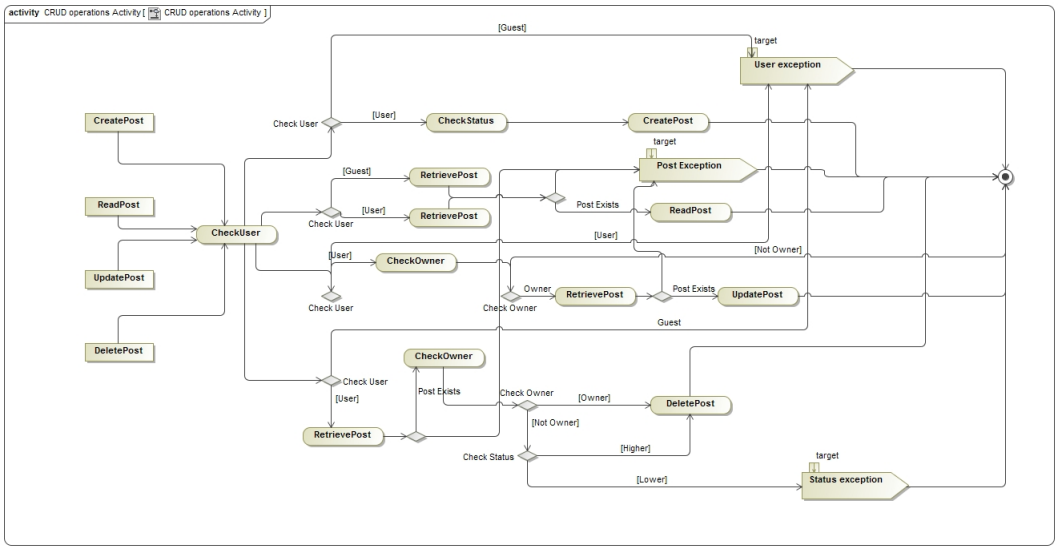
0.4.3 Use case/Service contracts

CRUD

1. Pre-conditions for CRUD: The buzz system needs to be available and a connection to the buzz system possible.

2. Request and Result Data Structures:





CRUD-Create

1. Pre-conditions: For a user to create posts they need to be registered of the buzz system. The user must be logged in to the buzz system.

2. Post-conditions: Posts will succeed to be created and will appear in

the buzz system where posted.

CRUD-Read

1. Pre-conditions: Posts must be present to be read.

2. Post-conditions: Once a post has been viewed by a logged in registered user, the post should be marked as read for that specific user.

CRUD-Update

1. Pre-conditions: For a user to update posts they need to be registered of the buzz system. The user must be logged in to the buzz system. The user must be the owner of the post.

2. Post-conditions: Post will be updated and shown on the buzz system to be updated.

CRUD-Delete

1. Pre-conditions: For a user to delete posts they need to be registered of the buzz system. The user must be logged in to the buzz system. The user must be the owner of the post unless a user with status level "x" views the post to be unfit. The post needs to exist for it to be deleted.

2. Post-conditions: Post successfully deleted will be hidden from the other users and buzz system and archived.Posts will never be physically deleted from the system itself.

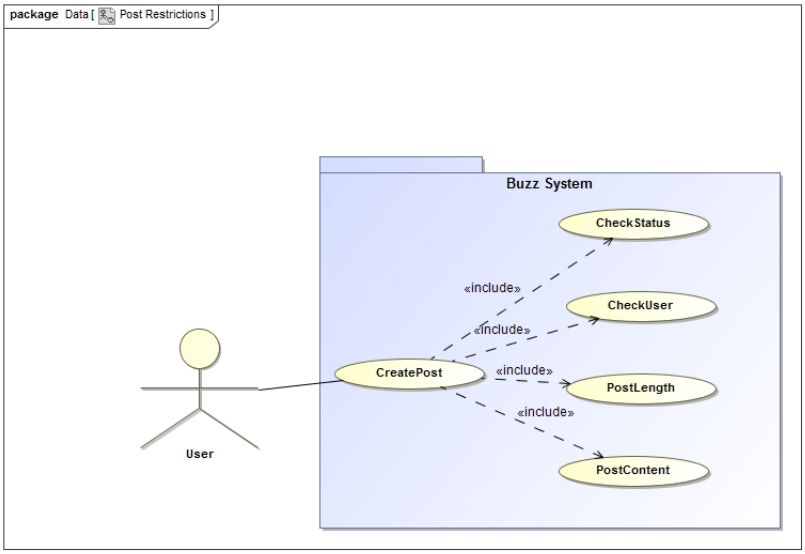
Message Restrictions

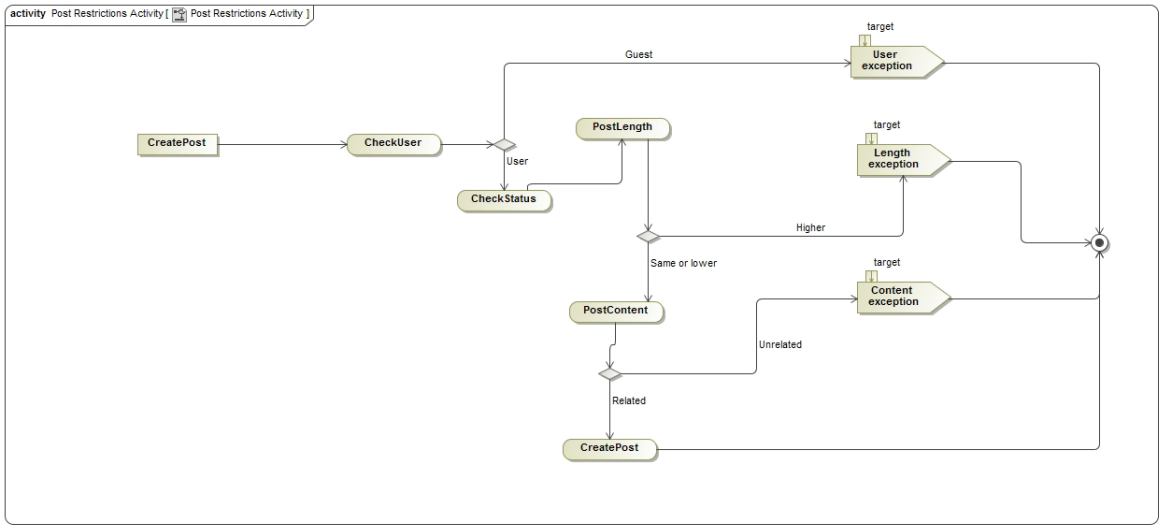
1. Pre-conditions: User must be registered on the buzz system. The buzz system needs to be available and a connection to the buzz system pos-

sible. The user must be logged in to the buzz system. User must be able to create posts. Length and content of posts must be specified by the buzz system of which the user is registered. Must have levels of posts.

2. Post-conditions: Post will be created and shown on the buzz system. Correct length and content of posts if buzz system is configured correctly.

3. Request and Result Data Structures:



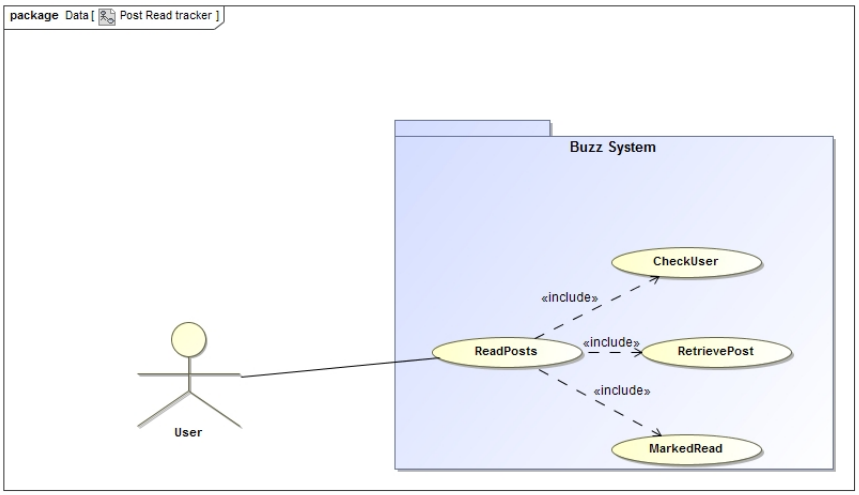


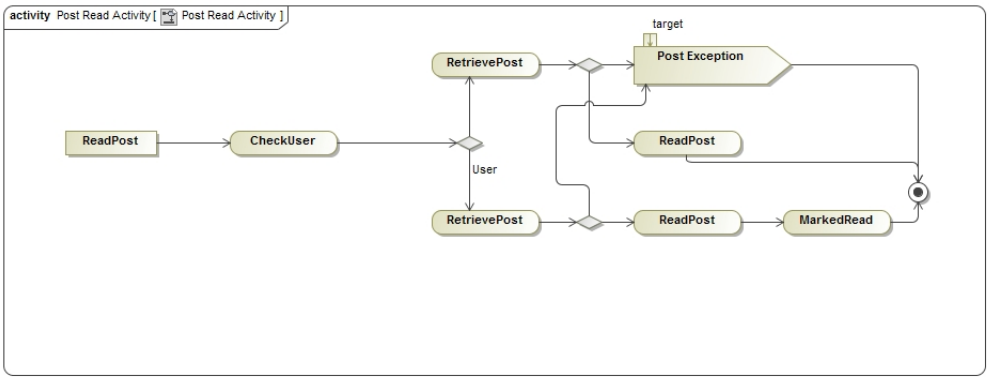
Message Read Tracker

1. Pre-conditions: User must be registered on the buzz system. The user must be logged in to the buzz system. Messages must exist to be read, marked as unread and must be seen as read by other users.

2. Post-conditions: Post is marked as unread.Post is marked when read and seen to be read on the buzz system.

3. Request and Result Data Structures:



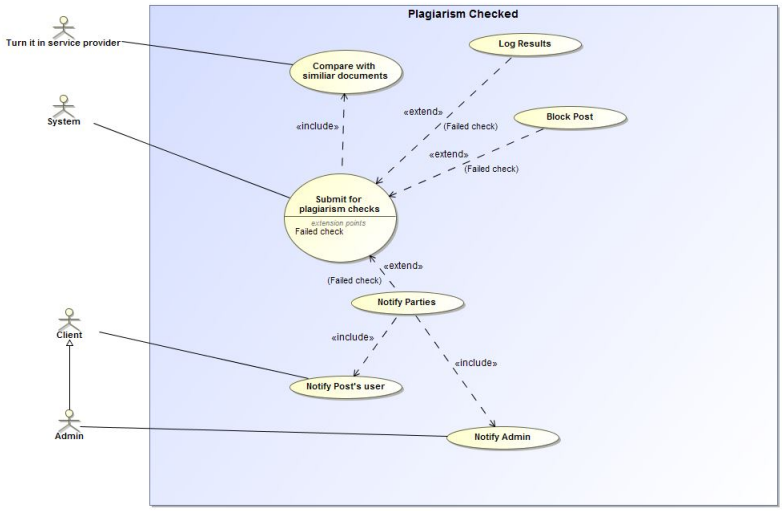


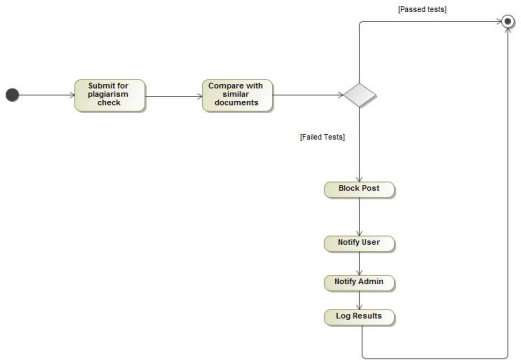
Check Post For Plagiarism

1. Pre-conditions: A post must have been submit by a registered user.

2. Post-conditions: If the post passes the plagiarism test, the new post should be registered on the system. If the post has failed the plagiarism tests the post will be blocked, then the post's user and the module's administrator will be notified. Afterwords a log will be stored.

3. Request and Result Data Structures:





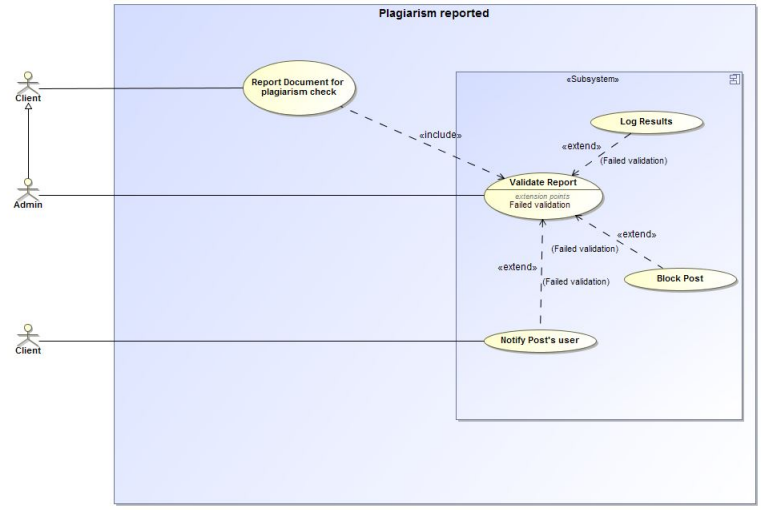
Report Post For Plagiarism

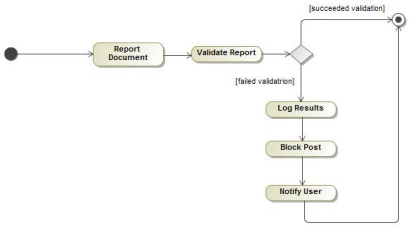
1. Pre-conditions: A post must have been reported by a registered user that has a level X account.

2. Post-conditions: The post is sent to an administrator to follow up on the report. If the administrator finds the post not plagiarised, the post will be registered on the system. If it is found to be plagiarised the post will be blocked, then the post's user and the module's administrator

will be notified. Afterwords a log will be stored.

3. Request and Result Data Structures:





Netiquette

1. Pre-conditions: A post must have been submit by a registered user.

2. If the post passes the netiquette test, the new post should be registered on the system. If the post has failed the netiquette tests the post will be blocked, then the post's user and the module's administrator will

be notified. Afterwords a log will be stored.

3. Request and Result Data Structures: