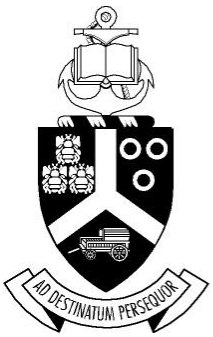

UNIVERSITEIT VAN PRETORIA
UNIVERSITY OF PRETORIA
YUNIBESITHI YA PRETORIA

Department of Computer Science

COS 301 - Software Engineering

COS 301 - Mini Project

Author:	*Student number:*
Hanrich Potgieter	u12287343
Chris Cloete	u13029721
Jason Richard Evans	u13032608
Kale-ab Tessera	u13048423
Lelethu Zazaza	u13028023
Goodness Adegbenro	u13046412
Herman Willem Keuris	u13037618

February 25, 2015

DECLARATION OF ORIGINALITY

UNIVERSITY OF PRETORIA

The University of Pretoria places great emphasis upon integrity and ethical conduct in the preparation of all written work submitted for academic evaluation.

While academic staff teach you about referencing techniques and how to avoid plagiarism, you too have a responsibility in this regard. If you are at any stage uncertain as to what is required, you should speak to your lecturer before any written work is submitted.

You are guilty of plagiarism if you copy something from another author's work (e.g. a book, an article or a website) without acknowledging the source and pass it off as your own. In effect you are stealing something that belongs to someone else. This is not only the case when you copy work word-for-word (verbatim), but also when you submit someone else's work in a slightly altered form (paraphrase) or use a line of argument without acknowledging it. You are not allowed to use work previously produced by another student. You are also not allowed to let anybody copy your work with the intention of passing if off as his/her work.

Students who commit plagiarism will not be given any credit for plagiarised work. The matter may also be referred to the Disciplinary Committee (Students) for a ruling. Plagiarism is regarded as a serious contravention of the University's rules and can lead to expulsion from the University.

The declaration which follows must accompany all written work submitted while you are a student of the University of Pretoria. No written work will be accepted unless the declaration has been completed and attached.

Full names of students: _____

Student numbers: _____

Topic of work: _____

Declaration

1. I understand what plagiarism is and am aware of the University's policy in this regard.
2. I declare that this assignment report is my own original work. Where other people's work has been used (either from a printed source, Internet or any other source), this has been properly acknowledged and referenced in accordance with departmental requirements.
3. I have not used work previously produced by another student or any other person to hand in as my own.
4. I have not allowed, and will not allow, anyone to copy my work with the intention of passing it off as his or her own work.

SIGNATURES: _____ DATE: _____

Software Requirements Specification and Technology Neutral Process Design

Buzz Space Discussions/Mini Project

Version: Version 0.1 Alpha For further references see gitHub. February 25, 2015

Contents

For further references see gitHub.

1 Functional requirements

1.1 Introduction

We use this document to give a high level overview of the buzz space discussion board. We have identified the various components. Our purpose is to create a dynamic and scalable solution. We also want to include an Achievement system that rewards users for using the discussion board. This document will inform you how we well achieve a system that is both scalable and pluggable.

1.2 Use case prioritiation

Critical

- Login System.
- Logout System
- Creation of a Buzz thread.

- CRUD OWN posts(Creating,Reading; Updating; Deleting).

- CRUD Other people's posts

Important

- Content Management (By higher level users and Administrators).
- User Restriction based on level.

- Automatic update of user status.
- Semi-Automatic evaluation of posts.
- Gathering and analization of statistical information.

- Social tagging system on threads (and posts).

- Searching and filtering of threads.

- Format code in an easy to view/edit layout.

Nice-To-Have

- Keeping track of who read what (ie. Message Highlighting).

- Semi-Automatic functionality for generating thread summaries.
- Text formatting functionality based on user level.
- Self organization functionality.
- Automatic plagiarism checking system.
- Semi-Automatic detection of netiquette rule violations.
- Have functionality to vote for and evaluate posts.

1.3 Use case/Service contracts

1. CRUD (Creating, reading, updating, deleting posts) OWN posts

- **Pre-conditions**

  Student is not registered for a course [Unless he/she is a Teaching Assistant/Tutor/Lecturer/Admi

  Student must be a high enough level to participate in the thread. [Unless he/she is given permission by an Admin or Lecturer.

- **Post-conditions**

  Student is still registered for the course.

  Student hasn't been blocked from the thread.

  Student level must still be high enough to participate in the thread.

- **Request and Results Data Structure**

2. CRUD (Creating, reading, updating, deleting posts) OTHER People's posts

- **Pre-conditions**

  Not an adminstrator. [Unless he/she is a Teaching Assistant/Tutor/Lecturer]

  Not a high enough level. [Unless given permission by the System Adminstrator]

- **Post-conditions**

  Individual must still be an Administrator/Teaching Assistant/Tutor/Lecturer.

  Level must still be high enough to allow an individual to CRUD other people's posts.

  Individual mustn't be blocked from that thread.

- **Request and Results Data Structure**

3. Keep track of what has been read and highlight all unread messages in a particular thread

- **Pre-conditions**

  Not a registered student for the course [Unless individual is an Administrator/Teaching Assistant/Tutor/Lecturer.

- **Post-conditions**

  Individual must still be registered for that subject.

  If not a registered student, the individual must still be an Administrator/Teaching Assistant/Tutor/Lecturer.

  Individual mustn't be blocked from that thread.

- **Request and Results Data Structure**

1.4 Required functionality

1.5 Process specification

1.6 Domain Model