UNIVERSITEIT VAN PRETORIA
UNIVERSITY OF PRETORIA
YUNIBESITHI YA PRETORIA

COS301 Mini Project Functional Requirements Specification

Group 2a

Matthew Gouws *u11008602*
Neo Thobejane *u11215918*
Roger Tavares *u10167324*
Rendani Dau *u13381467*
Szymon Ziolkowski *u12007367*
David Breetzke *u12056503*
Keagan Thompson *u13023782*
Name Surname *uxxxxxxx*

Version
February 23, 2015

1 History

Date	Version	Description
17-02-2015	Version 0.1	Document Template Created
17-02-2015	Version 0.2	UP Logo Added
18-02-2015	Version 0.2.1	Intro, Purpose, Conventions and Description Added
19-02-2015	Version 0.2.2	Added Skeleton for Required functionality

Contents

List of Figures

2 Introduction

The Computer Science Education Didactic and Application Research (CSEDAR) from the university of Pretoria. Have approached us in building a software platform to create a collaborative community by means of an online discussion. To aid students to excel in problem solving as group. Such a tool already exists for students to use however this tool lacks certain functionality, Lecturer - Student interaction, often students are unaware of who is higher ranked in terms of the course (Teaching Assistant, Tutor, Lecturer & student).

2.1 Purpose

This document serves to present the clients requirements on a functional level by use of use-case diagrams, Domain models, pre- & post conditions as well as possible input-output pairs.

2.2 Document Conventions

A ranking system of importance is used for the functional requirements based on a 'star' system with
***** - Critical
**** - Important
*** - Somewhat important
** - Nice to have
* - Not considered

2.3 Project Scope

Short overview with limitations and restrictions of the project.

2.4 References

Tutorial on Use case diagrams - `http://www.tutorialspoint.com/uml/uml_use_case_diagram.htm`

3 System Description

Buzz will be a complete software unit which is to be integrated seamlessly with existing web servers to be used by courses to encourage the use of online discussion. Users will be able to climb ranks up the online discussion forum and achieve more functionality as they progress. The system will also have certain functionality incorporated into awarding users marks if required so by the teaching staff. Teaching staff will also be able to archive and summarise threads. Users will also only be able to post in threads which pertain to them at that specific time, thus having a thread become 'Ancient' and hence archived no new information may be added to said thread.

4 Functional Requirements

4.1 Use Cases

We can Add all the details from MagicDraw here

4.2 Required Functionality

The Following system processes detail the functional requirements of the individual points.

1. Users must create, Update and delete posts.
    (a) Elaboration -
    (b) Importance - *****
    (c) Dependency level -
    (d) Pre-conditions
        i. Condition 1
        ii. Condition 2
    (e) Post-conditions
        i. Condition 1
        ii. Condition 2
    (f) Requester

2. (a) Elaboration -
   (b) Importance -
   (c) Dependency level -
   (d) Pre-conditions
      i. Condition 1
      ii. Condition 2
   (e) Post-conditions
      i. Condition 1
      ii. Condition 2
   (f) Requester

3. Restrict the length of messages and the type of content allowed in massages based on the level where it is posted as well as on the status of the user posting the message.
    (a) Elaboration -
    (b) Importance -
    (c) Dependency level -
    (d) Pre-conditions
        i. Condition 1
        ii. Condition 2
    (e) Post-conditions
        i. Condition 1
        ii. Condition 2
    (f) Requester

4. Restrict users to post on specified levels based on their status.

   (a) Elaboration - User interaction with the current Buzz space must be configurable by policy to allow users with higher levels to post to the Buzz space on higher levels such as directly below the main post while restricting low level users to only post in lower levels like sub level posts or even sub sub level posts. This allows high level users to post higher up in the Buzz spaces hierarchy while restricting low level users to the bottom.

   (b) Importance - ****

   (c) Dependency level - Relies on the ranking system to be implemented so that it can request user levels. A policy to govern the levels has to be supplied by the creator of the Buzz space.

   (d) Pre-conditions

      i. User must be part of the specific Buzz space.

      ii. Policy acquired.

      iii. User must try to post.

   (e) Post-conditions

      i. User successfully posted to the correct level as specified in the policy.

   (f) Requester - System (This is an automated system requirement)

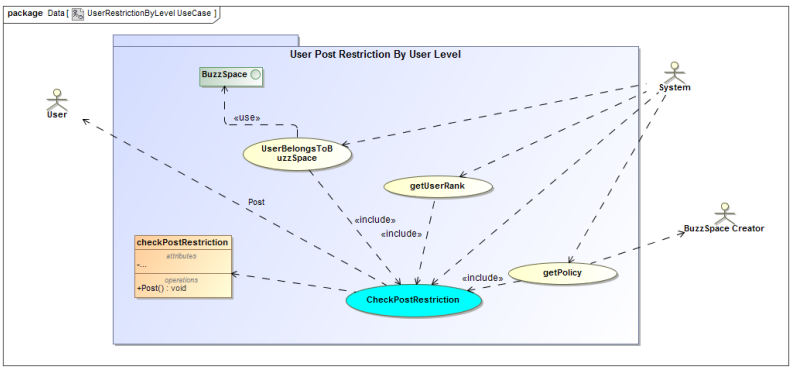

Figure 1: User Post Restriction By User Level Use Case

5. (a) Elaboration -
   (b) Importance -
   (c) Dependency level -
   (d) Pre-conditions
      i. Condition 1
      ii. Condition 2
   (e) Post-conditions
      i. Condition 1
      ii. Condition 2
   (f) Requester

6. (a) Elaboration -
   (b) Importance -
   (c) Dependency level -
   (d) Pre-conditions
      i. Condition 1
      ii. Condition 2
   (e) Post-conditions
      i. Condition 1
      ii. Condition 2
   (f) Requester

7. (a) Elaboration -
   (b) Importance -
   (c) Dependency level -
   (d) Pre-conditions
      i. Condition 1
      ii. Condition 2
   (e) Post-conditions
      i. Condition 1
      ii. Condition 2
   (f) Requester

8. Automatically change the status of a user based on the users participation

   (a) Elaboration - The System should be configured in such a way that when a user participates often the user will progress through the levels of the system, Specific number of points required per level.

   (b) Importance - ****

   (c) Dependency level - Requires the users level to be implemented before a user can participate and increase in level, Users should be able to post on Buzz

   (d) Pre-conditions

      i. User is in level x

      ii. User only needs y amount of points to progress

   (e) Post-conditions

      i. User achieved y amount of point

      ii. User is now in level x+1

   (f) Requester - System (Automatically checks each time a user posts)

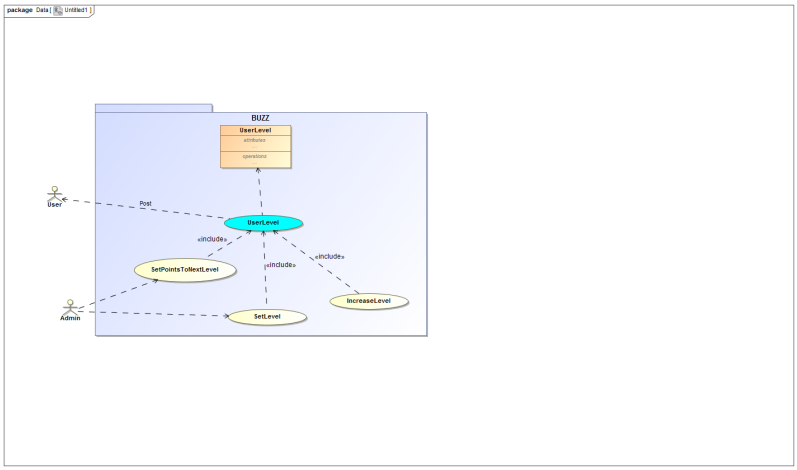

Figure 2: Automatically Change users level based on participation Use Case

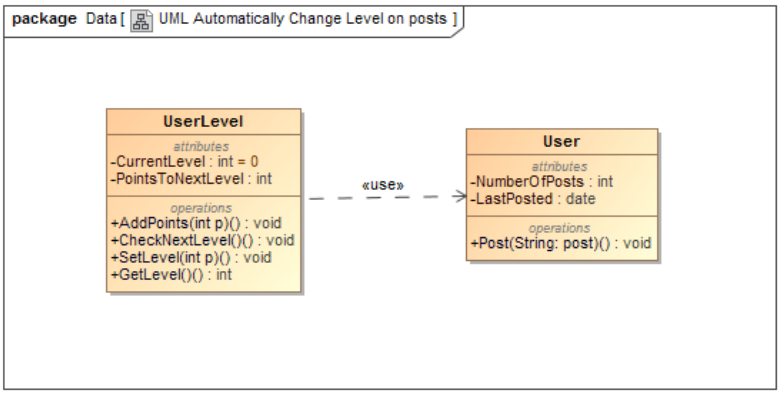

Figure 3: Automatically Change users level based on participation UML

9. (a) Elaboration -
   (b) Importance -
   (c) Dependency level -
   (d) Pre-conditions
      i. Condition 1
      ii. Condition 2
   (e) Post-conditions
      i. Condition 1
      ii. Condition 2
   (f) Requester

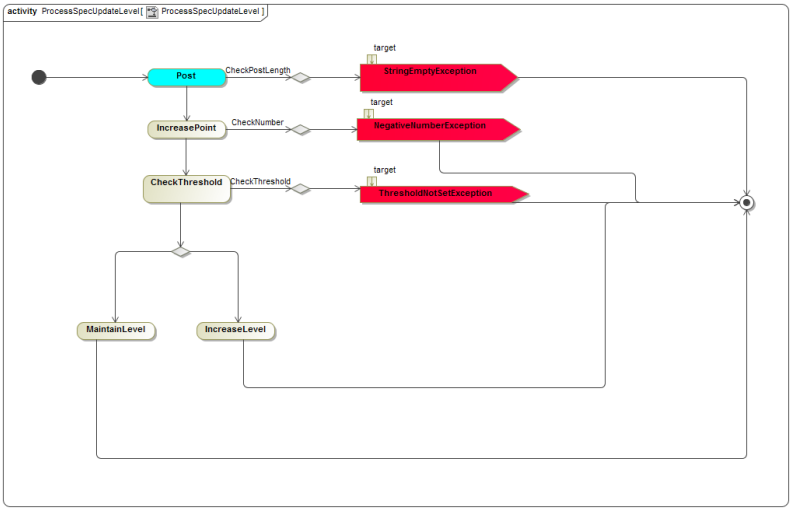

Figure 4: Process Specification for Automatically Update User Level

10. (a) Elaboration -
    (b) Importance -
    (c) Dependency level -
    (d) Pre-conditions
        i. Condition 1
        ii. Condition 2
    (e) Post-conditions
        i. Condition 1
        ii. Condition 2
    (f) Requester

11. Provide functionality to evaluate posts and vote for posts.
    (a) Elaboration -
    (b) Importance -
    (c) Dependency level -
    (d) Pre-conditions
        i. Condition 1
        ii. Condition 2
    (e) Post-conditions
        i. Condition 1
        ii. Condition 2
    (f) Requester

12. Gather statistical information on each user for graphical representation.

    (a) Elaboration - System must continuously gather each users contribution and participation in the Buzz space to allow the user to view a graphical representation of where they stand or rank up amongst their peers. Provide a game like scoreboard to motivate the users.

    (b) Importance - **

    (c) Dependency level - Requires the ranking system to be active in order to pull the stats.

    (d) Pre-conditions

        i. User has participated at least x times to gather historical data

        ii. More than y users have contributed to specific Buzz Space

    (e) Post-conditions

        i. User receives the graphical representation of their participation.

    (f) Requester - User

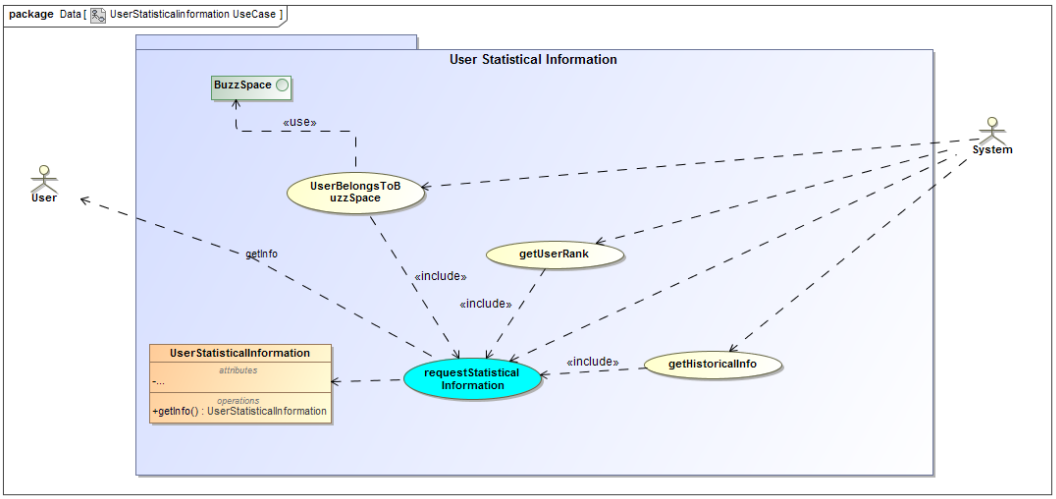

Figure 5: User Statistical Information

13. (a) Elaboration -
    (b) Importance -
    (c) Dependency level -
    (d) Pre-conditions
        i. Condition 1
        ii. Condition 2
    (e) Post-conditions
        i. Condition 1
        ii. Condition 2
    (f) Requester

14. (a) Elaboration -
    (b) Importance -
    (c) Dependency level -
    (d) Pre-conditions
        i. Condition 1
        ii. Condition 2
    (e) Post-conditions
        i. Condition 1
        ii. Condition 2
    (f) Requester

15. (a) Elaboration -
    (b) Importance -
    (c) Dependency level -
    (d) Pre-conditions
        i. Condition 1
        ii. Condition 2
    (e) Post-conditions
        i. Condition 1
        ii. Condition 2
    (f) Requester

16. Detect if a post is plagiarised

    (a) Elaboration - The entire post will be checked to see if it has been copied directly from another source, a full post quote will be open searched in a search engine, if any hits are found the post will be marked as possibly plagiarised and send to administrator

    (b) Importance - **

    (c) Dependency level - User must be able to post to Buzz

    (d) Pre-conditions

        i. User posts a post

    (e) Post-conditions

        i. Post is marked as Plagiarised - Added to Buzz(Invisible, Message sent to user and Administrator

        ii. Post is marked as not Plagiarised - Posted to Buzz

    (f) Requester

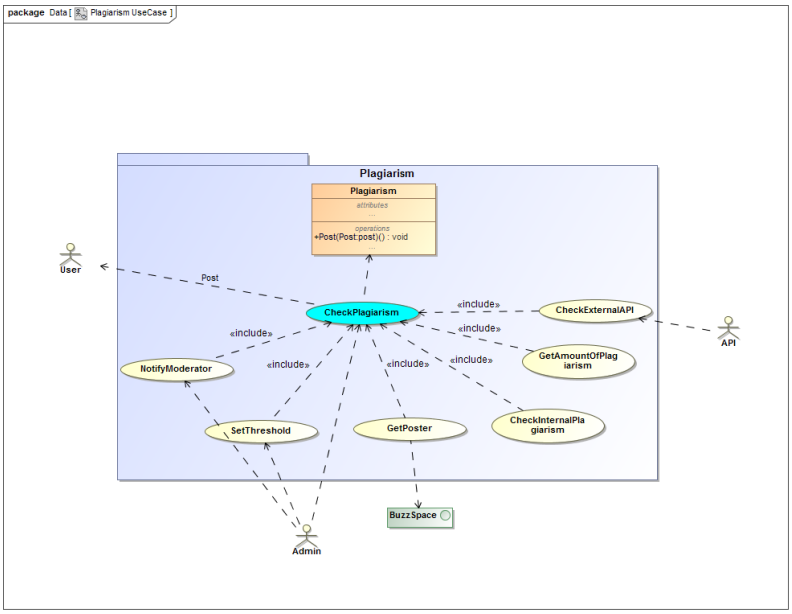

Figure 6: Plagiarism Check Use Case

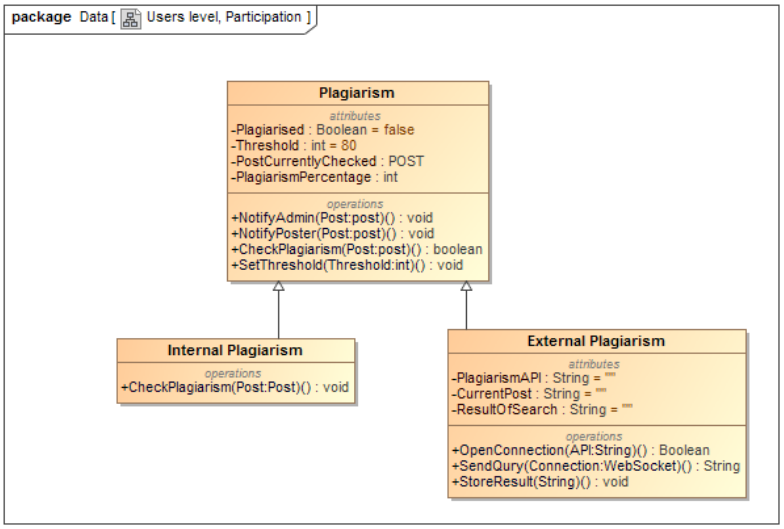

Figure 7: Plagiarism Check UML

17. (a) Elaboration -
    (b) Importance -
    (c) Dependency level -
    (d) Pre-conditions
        i. Condition 1
        ii. Condition 2
    (e) Post-conditions
        i. Condition 1
        ii. Condition 2
    (f) Requester

    ...

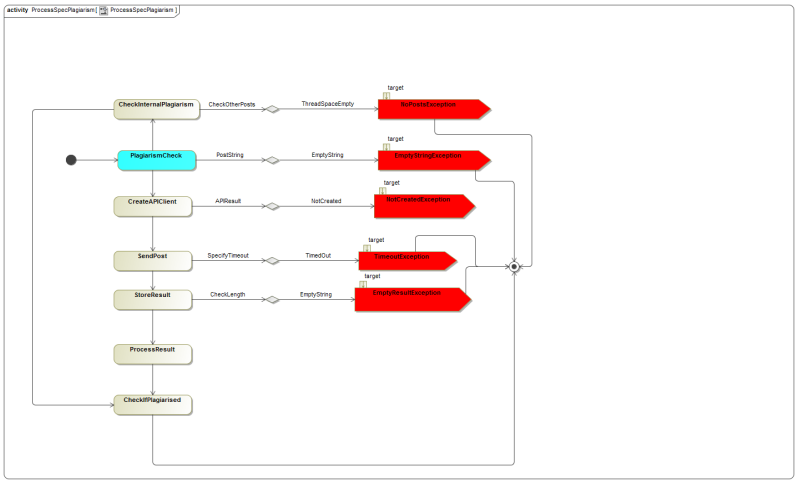

Figure 8: Process Specification for Checking Plagiarism API and Internal Checks

4.3 Entity Relationship Diagrams

4.4 Data Dictionary

Description of entities.

5 External Interface Requirements

Requirements for user interfaces.

6 Technical Requirements

List of non-functional requirements.

7 Requirement Matrices

8 Open Issues

9 Glossary