



UNIVERSITEIT VAN PRETORIA  
UNIVERSITY OF PRETORIA  
YUNIBESITHI YA PRETORIA

---

## COS301 MINI PROJECT FUNCTIONAL REQUIREMENTS SPECIFICATION

### Group 4B

Kyhle Ohlinger *u11131952*  
Andrew Parkes *u12189139*  
Sifiso Shabangu *u12081622*  
Matthew Nel *u10126229*  
Maret Stoffberg *u11071762*  
Sphelele Malo *u12247040*  
Tienie Pritchard *u12056741*

Here's a link to Github.

Version 1  
February 23, 2015

# Contents

|          |                                            |           |
|----------|--------------------------------------------|-----------|
| <b>1</b> | <b>Introduction</b>                        | <b>3</b>  |
| <b>2</b> | <b>Vision</b>                              | <b>3</b>  |
| <b>3</b> | <b>Background and System Description</b>   | <b>3</b>  |
| 3.1      | Related Project . . . . .                  | 4         |
| 3.2      | System Environment . . . . .               | 4         |
| <b>4</b> | <b>The Stakeholders</b>                    | <b>4</b>  |
| 4.1      | The Client . . . . .                       | 4         |
| 4.2      | The customers . . . . .                    | 4         |
| 4.3      | Maintenance Users . . . . .                | 4         |
| <b>5</b> | <b>Functional Requirements</b>             | <b>5</b>  |
| 5.1      | Scope and Limitations/Exclusions . . . . . | 5         |
| 5.2      | Required Functionality . . . . .           | 5         |
| 5.3      | Use case Prioritization . . . . .          | 11        |
| 5.4      | Process Specifications . . . . .           | 12        |
| 5.5      | Domain Model . . . . .                     | 13        |
| <b>6</b> | <b>Temporary Space</b>                     | <b>14</b> |
| 6.1      | Andrew . . . . .                           | 14        |
| 6.2      | Sifiso . . . . .                           | 15        |
| 6.2.1    | Point 1: . . . . .                         | 15        |
| 6.2.2    | Point 2: . . . . .                         | 15        |
| 6.2.3    | Point 3: . . . . .                         | 16        |
| 6.2.4    | Point 4: . . . . .                         | 16        |
| 6.3      | Maret . . . . .                            | 18        |
| 6.3.1    | Point 1: . . . . .                         | 18        |
| 6.3.2    | Point 2: . . . . .                         | 18        |
| 6.4      | Matthew . . . . .                          | 19        |
| 6.4.1    | Point 1: . . . . .                         | 19        |
| 6.4.2    | Point 2: . . . . .                         | 19        |
| 6.5      | Sphelele . . . . .                         | 20        |
| 6.5.1    | Point 1: . . . . .                         | 20        |
| 6.5.2    | Point 2: . . . . .                         | 21        |
| 6.6      | Tienie . . . . .                           | 22        |
| 6.6.1    | Point 1: . . . . .                         | 22        |
| 6.6.2    | Point 2: . . . . .                         | 25        |

|     |                |    |
|-----|----------------|----|
| 6.7 | name . . . . . | 29 |
|-----|----------------|----|

## List of Figures

|   |                                                 |    |
|---|-------------------------------------------------|----|
| 1 | CRUD Use case diagram . . . . .                 | 7  |
| 2 | Message Tracking Use case diagram . . . . .     | 8  |
| 3 | Message Restrictions Use case diagram . . . . . | 9  |
| 4 | CRUD Activity diagram . . . . .                 | 12 |
| 5 | Message Tracking Activity diagram . . . . .     | 13 |
| 6 | Message Restrictions Activity diagram . . . . . | 13 |

## 1 Introduction

The purpose of this document is to fully specify and outline the functional requirements of "The use of Online Discussions in Teaching (TODT)" research project, received from the Computer Science Education Didactic and Applications Research (CSEDAR) team, of the Computer Science Department of the University of Pretoria. The document also serves to give the client and developers a clear description and elaboration of the system to be implemented in its totality.

## 2 Vision

The project aims to provide an online space, which will be integrated into the CS website, where students, teaching assistants, and lecturers can engage in activities related to learning the content of specific modules. The system will also apply game-like concepts to motivate students to increase the quality of their participation, and consequently experience a deeper understanding of the course content.

## 3 Background and System Description

This project is due to the Computer Science department of the University of Pretoria having problems with the currently available tools for discussion forums. The following problems are hampering positive engagement of both teaching staff and students:

- Unorganised content,
- User Inexperience, and
- Low levels of excitement.

The System intends to create an online discussion forum that has automated feedback on common mistakes, game-like presentation as well as automated structuring.

The system also provides the COS 301 students with the opportunity to learn about the procedures used for creating, designing and developing projects for businesses, while also providing the University of Pretoria with a potentially new system that may be released as an opensource project, that could possibly be implemented worldwide.

### **3.1 Related Project**

The project is a face lift to the existing discussion forum of the Department of Computer Sciences, and aims to improve the existing one by bringing new features that would encourage students to be more involved in discussing certain modules.

### **3.2 System Environment**

The system will interact with LDAP ,which will handle credentials avoiding the need for a database.

## **4 The Stakeholders**

### **4.1 The Client**

The Client is Ms Vreda Pieterse at the Department of Computer Science.

### **4.2 The customers**

The Customers are Students of the Computer science who are enrolled in the modules, and lectures in the department.

### **4.3 Maintenance Users**

The system will be assigned administrators from the Computer Science department and they will ensure its maintenance.

## 5 Functional Requirements

### 5.1 Scope and Limitations/Exclusions

**Scope:**

The scope of the Buzz System project can be encapsulated as a solution that allows the users of the system to:

- Have access to basic the functionalities that are common on all online forums.
- The administrative staff should be able to manage the registration of users to the forum.
- User has to be able to participate in discussions.

The system shall be designed primarily for use by lecturers, teaching assistants, and students for the core purpose of engaging in activities related to learning the content of specific modules, and increasing the participation, and consequently deeper understanding of the modules at hand.

**Limitations/Exclusions:**

### 5.2 Required Functionality

The following system processes describe the functional requirements of the system.

1. CRUD Operations:

- **Purpose:** Users should be able to create, read, update and delete posts.  
**Limitations:** Not all users should be able to use all the functions. Some users may even CRUD other users posts.
- **Importance:** Critical
- **Pre-Conditions:**
  - Buzz space must exist.
  - User must be connected to the buzz system.
  - *Create:*
    - \* Must have necessary permission to create posts.
    - \* Must be registered on the buzz system.

- *Read:*
  - \* Post must exist.
- *Update:*
  - \* Post must exist.
  - \* Must either be owner of the post, or have necessary permissions to update the post.
- *Delete:*
  - \* Post must exist.
  - \* Must either be owner of the post, or have necessary permissions to delete the post
- **Post-Conditions:**
  - *Create:*
    - \* Post will have been created.
    - \* Post may not have been created, due to some error.
  - *Read:*
    - \* If logged in, post will be marked as read for the specific user.
  - *Update:*
    - \* Post will be updated if user has required permissions.
    - \* Post may not have been updated due to some error.
  - *Delete:*
    - \* Post will be marked as deleted, and thus removed from the discussion board.
    - \* Post is not actually removed from the server, it is however hidden from all users.
    - \* Post may not have been deleted due to some error.



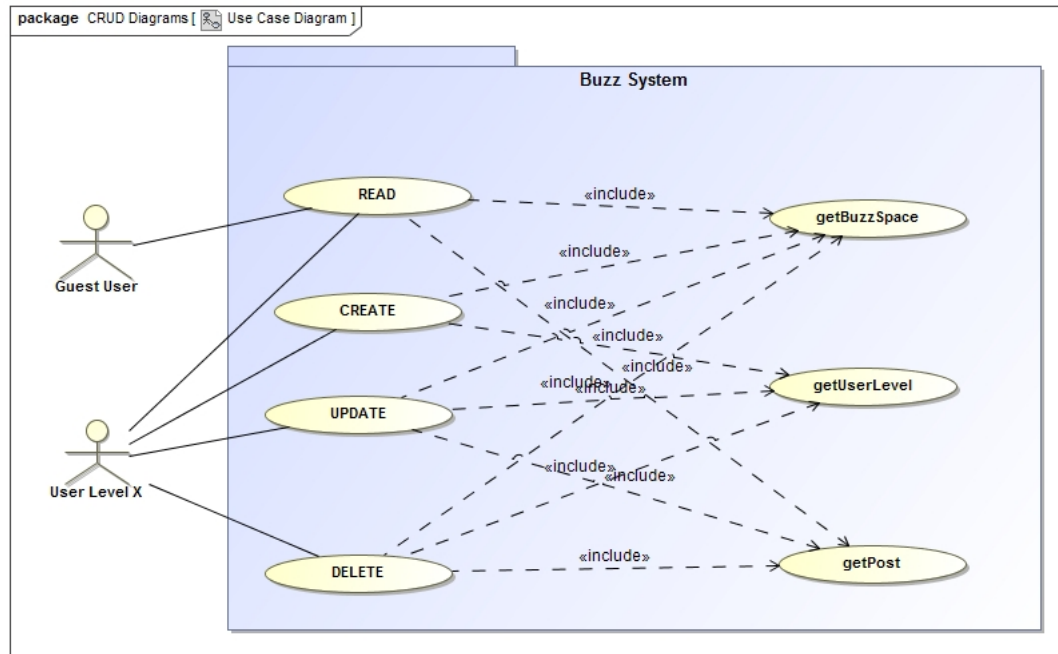


Figure 1: CRUD Use case diagram

2. Tracking read and unread messages:

- **Purpose:** The system must keep track of who has read what, and highlight unread messages for each user.
- **Importance:** Critical
- **Pre-Conditions:**
  - Buzz space must exist.
  - User must be registered to the buzz system.
  - User must be logged into the system while viewing the post.

**Post-Conditions:**

- Post is marked as read.
- Post remains unmarked due to some error.

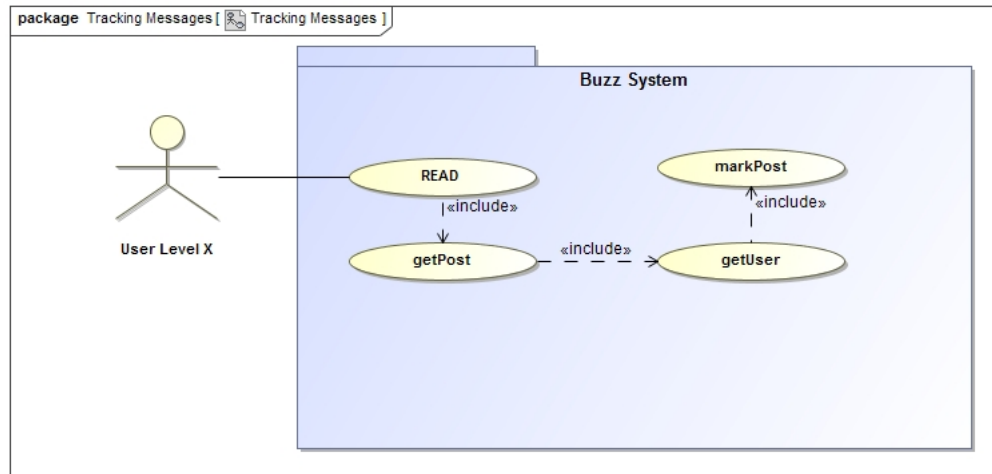


Figure 2: Message Tracking Use case diagram

### 3. Message Restrictions:

- **Purpose:** This deals with the restriction of posting messages.
- **Limitations:** Message length should be restricted. Content type should also be restricted based on level and status of the user posting the message.
- **Importance:** Critical
- **Pre-Conditions:**
  - Buzz space must exist.
  - User must be registered to the buzz system.
  - User must have necessary permissions to create posts of a certain length or content type.
  - Content type and message length must be established by the creator of that specific buzz (configurable).
- **Post-Conditions:**
  - Post is created.
  - Post may not have been created due to some error.

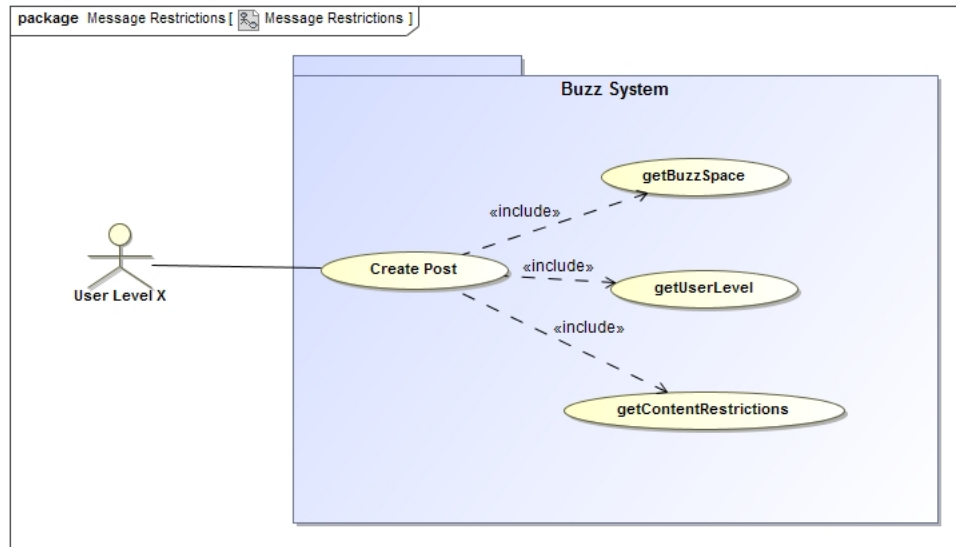


Figure 3: Message Restrictions Use case diagram

#### 4. Social Tagging

- **Purpose:** This deals with social tagging, broad folksnomy type is used in this social tagging

**Limitations:** Not all users will be able to tag a buzz space, Users with higher privileges and lectures will be able to social tag buzz space

- **Importance:** Critical

**Pre-Conditions:**

- Buzz space must exist.
- User must be registered to the buzz system.
- Buzz space must have a rating from users to be tagged .

**Post-Conditions:**

- Buzz space tagged with a keyword.
- Buzz space available at tag box for fast access.

#### 5. Self organise

- **Purpose:** TThis deals with self-organisation based on social tagging and allow the user to view according to the base structure,

owns structure or public structure.

**Limitations:** Users with higher privileges will be able to organise view to their own structure.

- **Importance:** Critical

**Pre-Conditions:**

- Buzz space must exist.
- User must be registered to the buzz system.
- Buzz space must have a rating from users to be tagged .
- Social tagging must be applied to other buzz space

**Post-Conditions:**

- Tagged buzz space with higher rating from users will be organised to be in base structure
- Most accessed tagged buzz space will be included in public structure.
- Users with buzz space can organise their own structure.

## 6. Buzz Tag

- **Purpose:** This deals with an addition of a read later section, which saves buzz space with long comments for a user to read later when logged in and remind the user every time when logged.

**Limitations:** All users will have this feature by default

- **Importance:** Critical

**Pre-Conditions:**

- Buzz space must exist.
- User must be registered to the buzz system.

**Post-Conditions:**

- Read later section will be created and added on the side of the users portal
- Buzz space reference must be saved in a read later section on the users portal is the user clicked a buzz space to read later section

## 7. Read Later Section

- **Purpose:** This deals with buzz space tagging for least privilege users ,this will help social tagging by providing trending buzz space and notification system for users who opened buzz space or

waiting for a reply on a comment.

**Limitations:** All users will have this feature by default

- **Importance:** Critical

**Pre-Conditions:**

- Buzz space must exist.
- User must be registered to the buzz system.
- User optioned for notifications when registering on the buzz system

**Post-Conditions:**

- Menu will be provided to the user with icons similar to Glyphicons to describe the buzz space thread for the user to click on.
- After user click a certain icon , thread will be rated according to the icon and each icon have description of the buzz space thread ranging from lame buzz thread space to must read buzz thread space and also apply to comments on the buzz space thread .
- Notification will be send atomically to users read later section to inform about a new comment on the buzz space user commented

### 5.3 Use case Prioritization

- **Critical:**

- CRUD operations.
- Tracking read and unread messages.
- Message Restrictions.

- **Important:**

- ...

- **Nice-to-Have:**

- Social Tagging
- Self organise
- Buzz Tag
- Read Later Section

## 5.4 Process Specifications

Requirements around the processes which needs to be followed regarding the use cases:

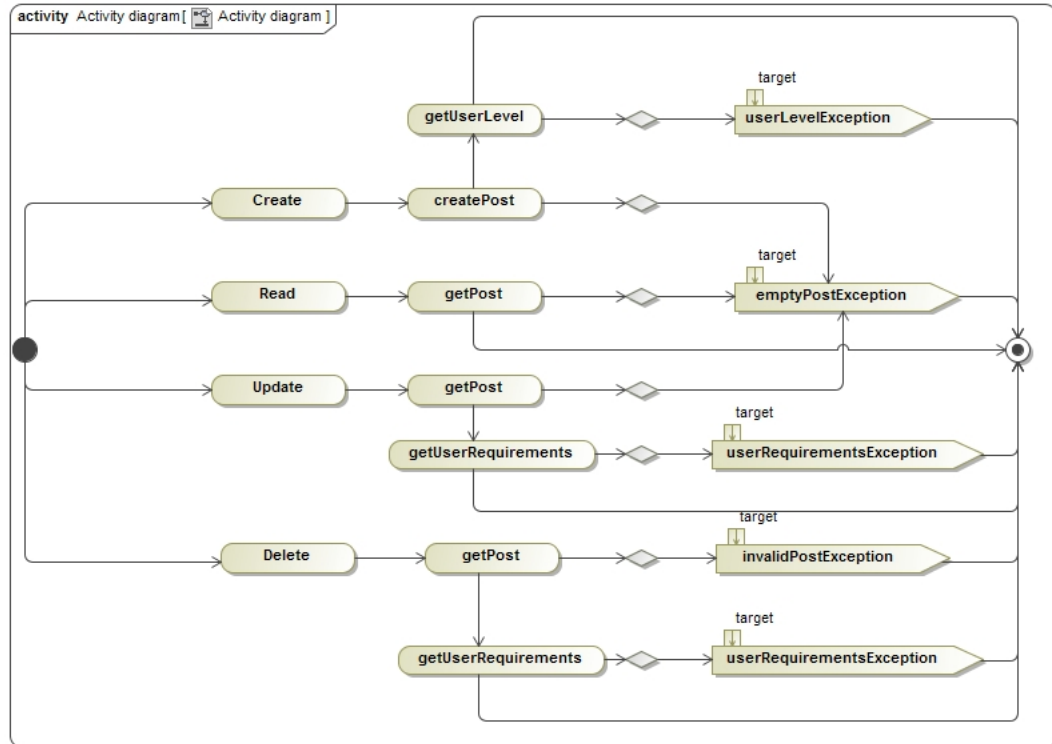


Figure 4: CRUD Activity diagram

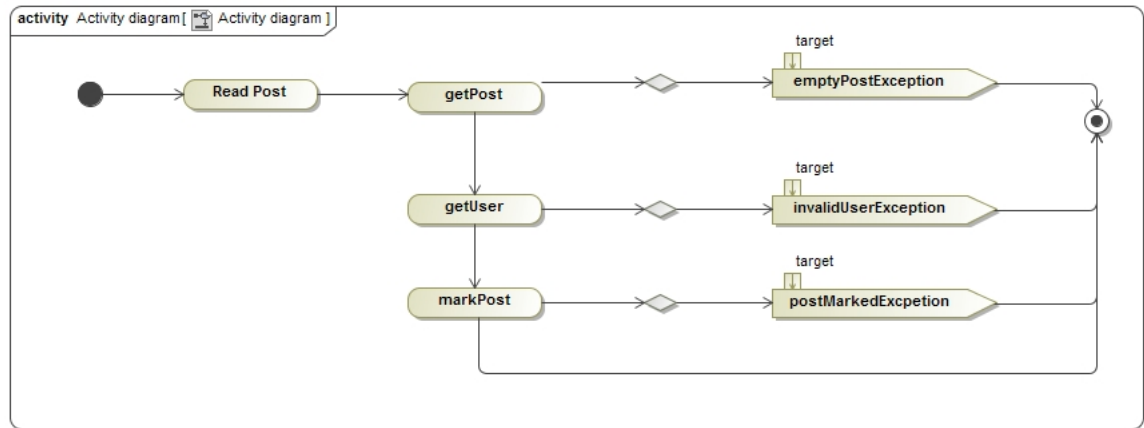


Figure 5: Message Tracking Activity diagram

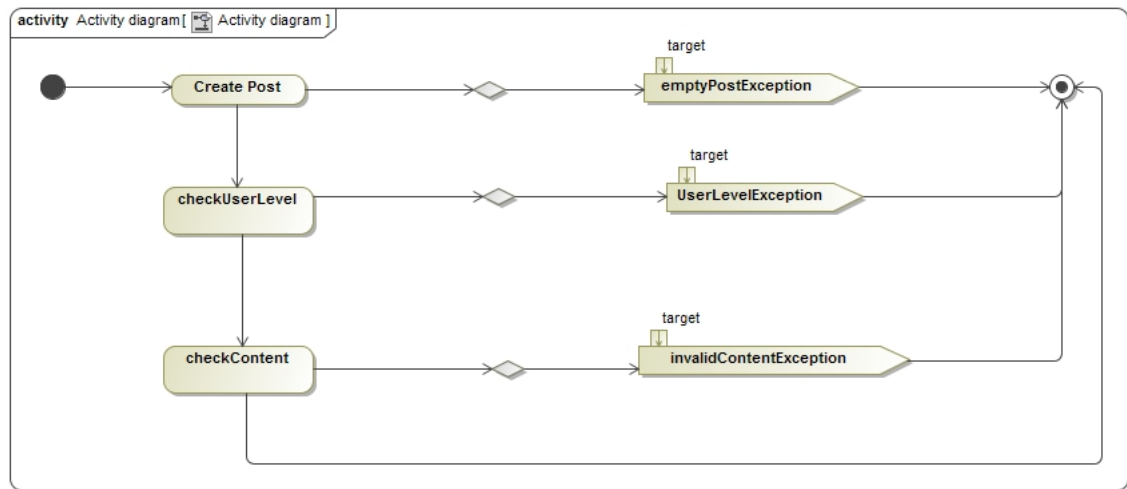


Figure 6: Message Restrictions Activity diagram

## 5.5 Domain Model

Temporary words

## 6 Temporary Space

### 6.1 Andrew



## 6.2 Sifiso

### 6.2.1 Point 1:

1. **Scope:** This deals with social tagging, broad folksnomy type is used in this social tagging  
**Limitations/exclusions:** Not all users will be able to tag a buzz space, Users with higher privileges and lectures will be able to social tag buzz space
2. **Use case Prioritization:** nice to have
3. **Use case/Service Contracts:**  
**Pre-Conditions:**
  - User must be registered to the buzz system.
  - User must have necessary privileges.
  - Buzz space must have a rating from users to be tagged**Post-Conditions:**
  - Buzz space tagged with a keyword.
  - Buzz space available at tag box for fast access.

### 6.2.2 Point 2:

1. **Scope:** This deals with self-organisation based on social tagging and allow the user to view according to the base structure, owns structure or public structure.  
**Limitations/exclusions:** Users with higher privileges will be able to organise view to their own structure.
2. **Use case Prioritization:** nice to have
3. **Use case/Service Contracts:**  
**Pre-Conditions:**
  - User must be registered to the buzz system.
  - User must have necessary privileges.
  - Buzz space must have a rating from users.
  - Social tagging must be applied to other buzz space**Post-Conditions:**

- Tagged buzz space with higher rating from users will be organised to be in base structure
- Most accessed tagged buzz space will be included in public structure.
- Users with buzz space can organise their own structure.

### 6.2.3 Point 3:

1. **Scope:** This deals with an addition of a read later section, which saves buzz space with long comments for a user to read later when logged in and remind the user every time when logged.

**Limitations/exclusions:** All users will have this feature by default

2. **Use case Prioritization:** nice to have

3. **Use case/Service Contracts:**

**Pre-Conditions:**

- User must be registered to the buzz system.

**Post-Conditions:**

- Read later section will be created and added on the side of the users portal
- Buzz space reference must be saved in a read later section on the user's portal is the user clicked a buzz space to read later section

### 6.2.4 Point 4:

1. **Scope:** This deals with buzz space tagging for least privilege users ,this will help social tagging by providing trending buzz space and notification system for users who opened buzz space or waiting for a reply on a comment.

**Limitations/exclusions:** All users will have this feature by default

2. **Use case Prioritization:** nice to have

3. **Use case/Service Contracts:**

**Pre-Conditions:**

- User must be registered to the buzz system.
- User must be allowed to the buzz space.

- User optioned for notifications when registering on the buzz system

**Post-Conditions:**

- Menu will be provided to the user with icons similar to Glyphicons to describe the buzz space thread for the user to click on.
- After user click a certain icon , thread will be rated according to the icon and each icon have description of the buzz space thread ranging from lame buzz thread space to must read buzz thread space and also apply to comments on the buzz space thread .
- Notification will be send atomically to users read later section to inform about a new comment on the buzz space user commented

## 6.3 Maret

### 6.3.1 Point 1:

1. **Scope** The system must send template messages automatically to individual users or specified groups, like a welcoming message or a notification message.

**Limitations/exclusions:**

2. **Use case Prioritization:** Nice-to-have **Use case/Service Contracts:**

**Pre-Conditions:**

- The user must be registered to the buzz system.
- The message template should exist.
- The system must be able to select a certain group based on specific information, to send the group message to.

**Post-Conditions:**

- The user must be alerted of the message.
- The user must not be able to reply to the message.
- The user must be able to delete the messages.
- The user must not be able to see what other users have received the same message via group messaging.

### 6.3.2 Point 2:

1. **Scope** The system must automatically change the status of a user based on his participation.

**Limitations/exclusions:**

2. **Use case Prioritization:** Important **Use case/Service Contracts:**

**Pre-Conditions:**

- The user must be registered to the buzz system.
- The user must be logged in for his status to be affected by his participation.
- The user's current status will be updated, so he must have a current status.

**Post-Conditions:**

- The users privileges change when his status change.
- The user must be able to view his status.
- The users status is public.

## 6.4 Matthew

### 6.4.1 Point 1:

1. **Scope:** Statistical information created from evaluation to capture average mark of every student within a time frame. Visual reporting of a participants evaluation in correlation to the average of the evaluation of all users or certain groups of users for gamification concept.

**Limitations/exclusions:** : Not all users may see statistical information of users. Only higher level users may view statistical information and visual reporting of users.

2. **Use case Prioritization:** Important.

3. **Use case/Service Contracts:**

#### **Pre-Conditions:**

- User must be connected to buzz.
- User must be part of discussion or module to be evaluated.
- User may not be of higher level of the users who monitor the statistical information.

#### **Post-Conditions:**

- Results and statistical information may not be altered.
- No more statistical information may be formulated once time period passes.
- All users may see there statistical information.
- Statistical information will be represented in a visual representation from that time period.

### 6.4.2 Point 2:

1. **Scope:** Improved post editor for example text formatting and automatic pretty printing of code in posts.

**Limitations/exclusions:**

Not all users will be able to use these options. Has to be earned.  
Certain options are available for all users by default.

2. **Use case Prioritization:** Nice-To-Have.

3. **Use case/Service Contracts:**  
**Pre-Conditions:**

- User must be connected to buzz system.
- User must be of a high enough level.
- Users must be using the post editor.

**Post-Conditions:**

- Message will be formatted accordingly and marked up (e.g. font colouring, emoticons, various fonts) to how the user wanted.
- User must send message to be viewable by other users.

## 6.5 Sphelele

### 6.5.1 Point 1:

1. **Scope:** The website will allow users to search the website for topics and buzz spaces, and then filter those search results by various categories.

**Limitation/Exclusions:** Users will be limited to 4 filter categories, namely: topic, date posted/last updated, buzz space name and rating.

2. **Use case Prioritization:** Important

3. **Use case/Service Contracts:**

**Post-Conditions:**

- Search results are returned
- Search filters are successfully executed

4.

5.

6.

### 6.5.2 Point 2:

1. **Scope:** The website will allow users to evaluate/vote for posts on the website.

**Limitations/Exclusions:** Only users that are logged in to the system and have the privilege rights to evaluate/vote on posts will be allowed to evaluate/vote on posts. Higher Privileged users' votes will push posts higher (or lower) than lower privileged users.

2. **Use case Prioritization:** Important

3. **Use Case/Service Contract:**

#### **Pre-Conditions**

- User privilege level allows user to vote for post
- post must exist and not be removed

#### **Post Conditions**

- Post rating is updated.
- post moves up/down accordingly

4.

5.

6.

## 6.6 Tienie

### 6.6.1 Point 1:

1. **Scope:** This deals with the built in plagiarism checker. It describes the functionality of the plagiarism checker as well as its aims and goals.

**Limitations/exclusions:** The plagiarism checker is built into the system, and is thus not an optional feature for users posting. However, the checker may be configurable (i.e. turned on and off) by board administrators.

2. **Use case Prioritization:** Nice-To-Have

3. **Use case/Service Contracts:**

- **Pre-Conditions**

- All posts from within the website will need to be available for traversal in text format.
- A reference collection of documents assumed to be valid will need to be obtained for traversal in text format.
- A similarity threshold will need to be agreed upon.
- Access to the Internet (not just Intranet) is a must.
- A google Search Engine ID and API Key for getting RESTful results to get web search results for comparison in JSON format (Google Custom Search Engine 100 Free Search Queries Per Day)

- **Post-Conditions**

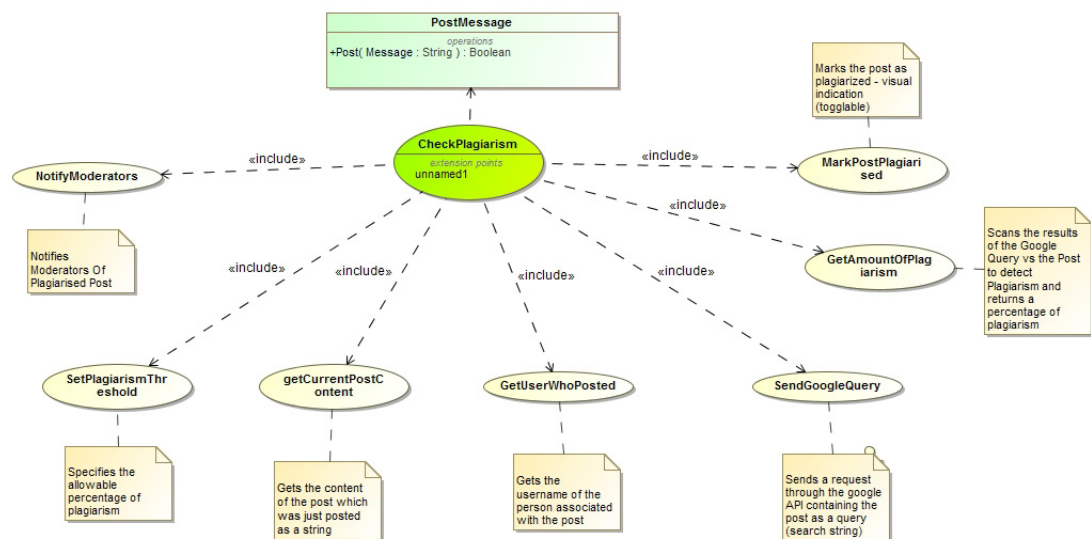
- A moderator must be alerted if the plagiarism percentage surpasses the allowed threshold.
- The percentage of plagiarism detected must be alerted to a moderator if it surpasses the threshold.
- If plagiarism is detected above the allowed threshold, the post must be highlighted as suspicious.
- Unique documents scanned must be added to the repository of reference documents.



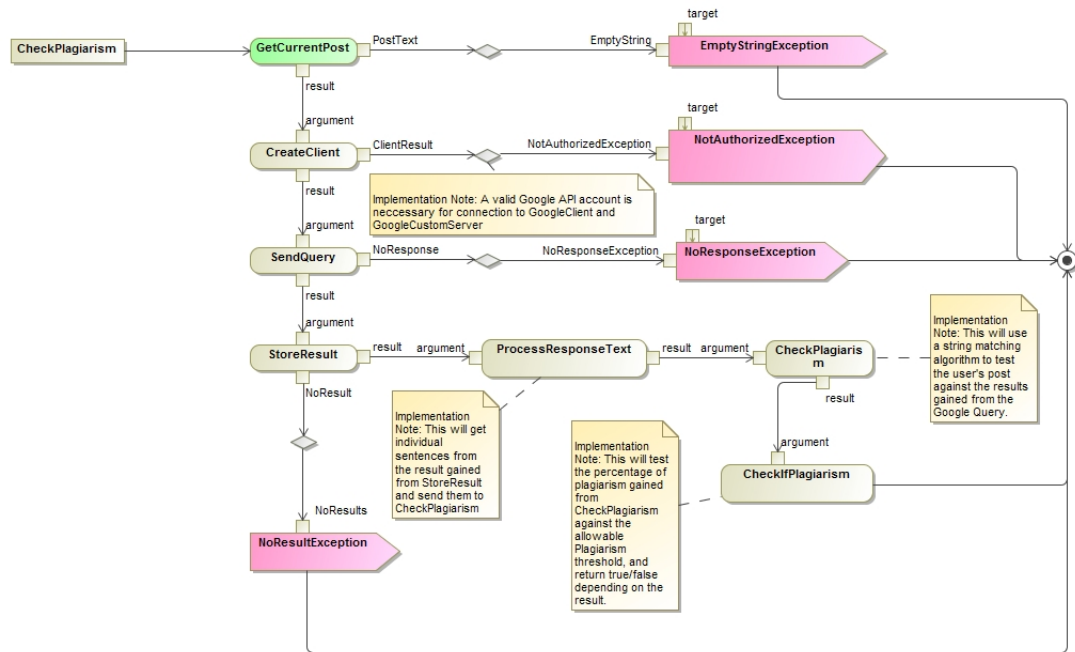
## • Request and Results Data Structures



## 4. Required Functionality



## 5. Process Specifications



### 6.6.2 Point 2:

1. **Scope:** This deals with the built in Netiquette checker. It describes the functionality of the Netiquette checker as well as its aims and goals.

**Limitations/exclusions:** The netiquette checker is built into the system, and is thus not an optional feature for users posting. However, the checker may be configurable (i.e. turned on and off) by board administrators.

2. **Use case Prioritization:** Nice-To-Have

3. **Use case/Service Contracts:**

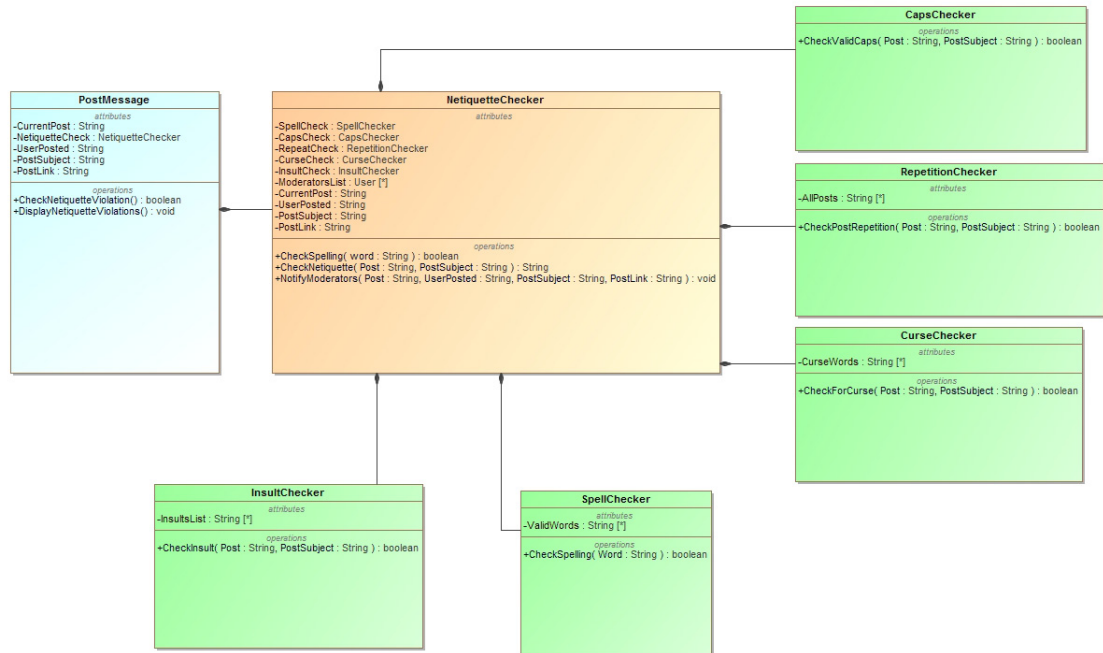
- **Pre-Conditions**

- All posts from within the website will need to be available for traversal in text format
- The netiquette checker will only check the following netiquette rules:
  - \* Proper Spelling
  - \* Repetition of Asked Questions
  - \* Checking all caps
  - \* Curse word checking
  - \* Checking for general insults
- We will need a text file containing a large list of commonly used words for the spell checker.
- We will need a text file containing a large list of commonly used curse words.
- We will need a text file containing a large list of words commonly used with the intention of insulting someone.

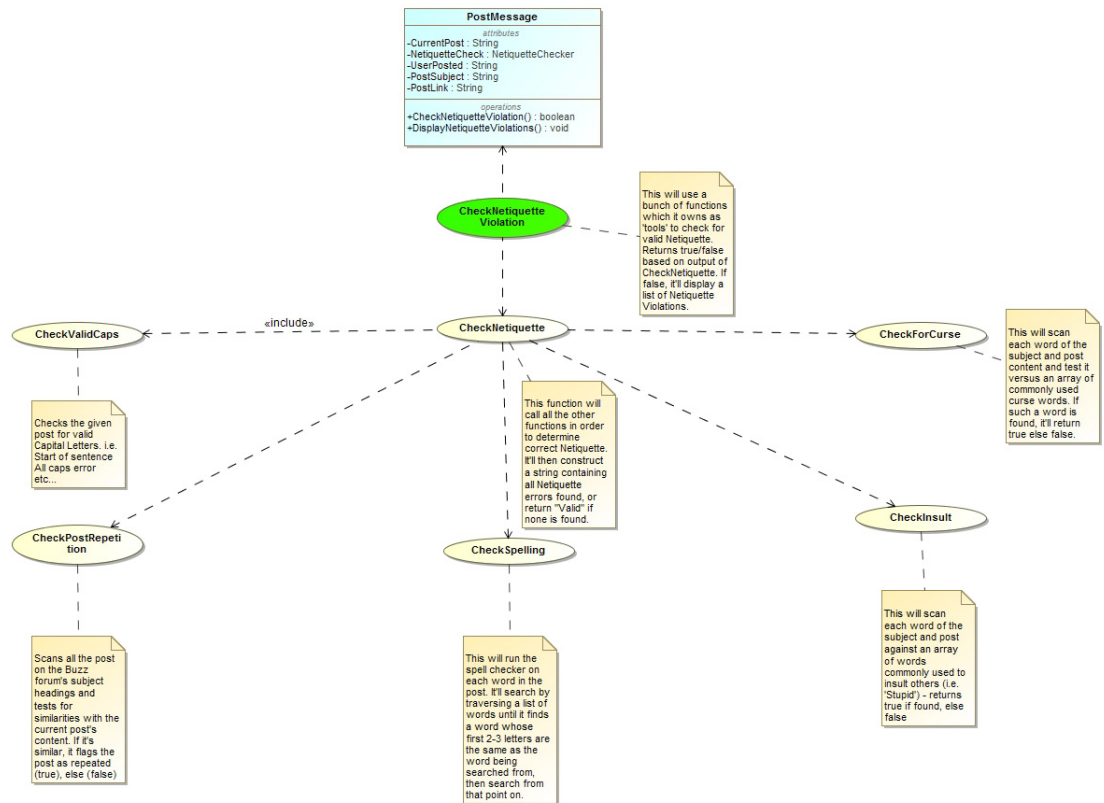
- **Post-Conditions**

- A moderator must be alerted if a post is detected as violating netiquette.
- The post must be highlighted to show that it violates netiquette rules.
- A moderator should be able to hide a post which violates netiquette.
- The system itself must alert the user of what netiquette rules they have violated as well as give them the opportunity to edit the post and fix them.

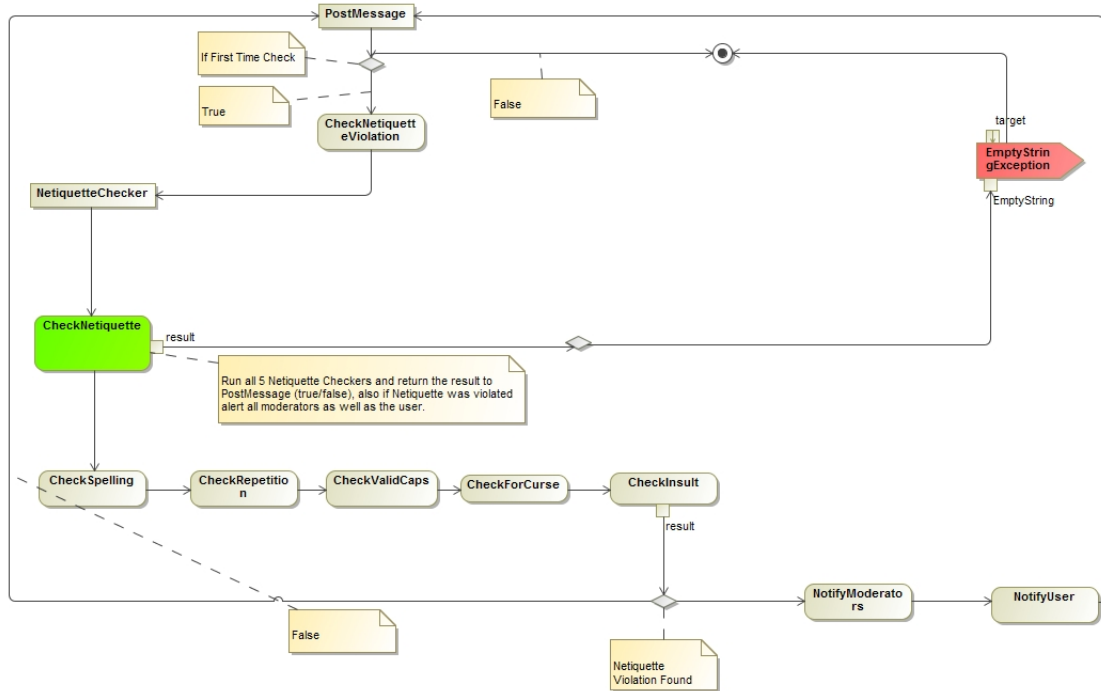
## • Request and Results Data Structures



## 4. Required Functionality



## 5. Process Specifications



## 6.7 name

- 1.
- 2.
- 3.
- 4.
- 5.
- 6.