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*
Ivan Henning *u13008219*
Szymon Ziolkowski *u12007367*
David Breetzke *u12056503*
Keagan Thompson *u13023782*

Version
February 24, 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
23-02-2015	Version 0.2.3	Updated Use Case prioritization and fixed newPage with Clearpage

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

The scope of this project is to create a Buzz Space system to integrate into the Computer Science Department's website at the University of Pretoria. This software solution will provide an online forum that is both organized and interactive to engage the students in their studies. The project can then later also be expanded to multiple universities and institutions.

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

4.1.1 Use Case Prioritization

Critical:

- Users must be able to create, Read, update and delete posts.
- Users must only be able to post on specific threads/levels based on their level.
- Staff should be able to summarize, close/hide threads and move posts around.
- Buzz must integrate seamlessly with any host site.

Important:

- The system should keep track of what posts have been read by a specific user and highlight unread posts.
- The system should allow for social tagging, allowing users to easily locate threads on what they are looking for.

Important-Nice to have:

- The post length should be restricted. Users of higher levels can post and embed Pictures, Videos etc.
- A users status should automatically update based on their participation.
- Post should be able to be up-voted, shown higher in the thread.
- Statistics should be available for each student displaying their marks, and visual reporting of their level.

Nice to have:

- Semi-automated thread summary creation.
- Create an automated template based on Message and other users.
- Provide searching and filtering.
- Enhancement of posts, such as Rich-text-format editor.
- Apply organization of content based on tags, base structure or ownership structure.
- Detect if a post is plagiarised.
- Detect if Netiquette rule are broken.

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. Keep track of who has read what and highlight unread messages for each user.

   (a) Elaboration - This process will allow the student/user to see which thread he/she still has to read by applying some form of emphasis on the thread. It will also allow a lecturer/board moderator to view read statistics of a thread such as how many students have read a thread and who those students are.

   (b) Importance - *

   (c) Dependency level - This function is not critical to the core operation of the system but is a client requested amenity.

   (d) Pre-conditions

      i. A user must be logged in and registered for that particular buzz space.

      ii. A user must be logged in and have elevated privileges, such as lecturer/moderator status, to view read statistics of a post

   (e) Post-conditions

      i. User is presented with formatted view highlighting unread threads

      ii. User(lecturer) is presented with read statistics of a particular thread

   (f) Requester - Client

3. Restrict the length of messages and the type of content allowed in messages based on the level where it is posted as well as on the status of the user posting the message.

   (a) Elaboration - Restrictions on the length of message should be configurable by policy, according to user level and post location. This will allow Buzz Space creators specific control over content type and post length for users of different level/rank.

   (b) Importance - ***

   (c) Dependency level - Relies on the ranking system to be implemented and provide a user's level/rank. Two policies must be set by the Buzz Space creator, one for message length according to level/rank, and one for permitted content types according to level/rank.

   (d) Pre-conditions

      i. User must belong to a Buzz Space.

      ii. User must attempt to post.

      iii. Policies must be acquired.

   (e) Post-conditions

      i. User successfully posted a post that contains only permitted content types.

      ii. User successfully posted a post that contains less than or equal to the permitted character length.

   (f) Requester - System

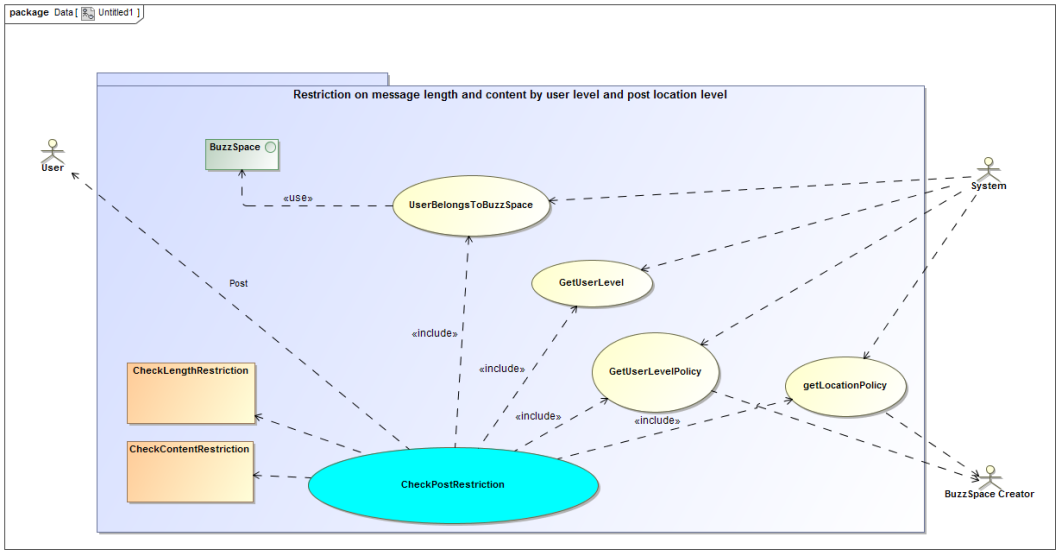


Figure 1: Message Length and Content Restriction

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)

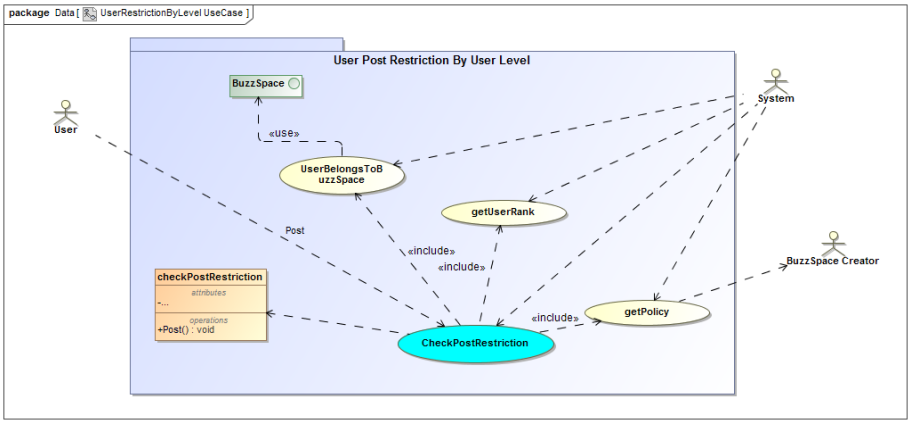


Figure 2: User Post Restriction By User Level Use Case

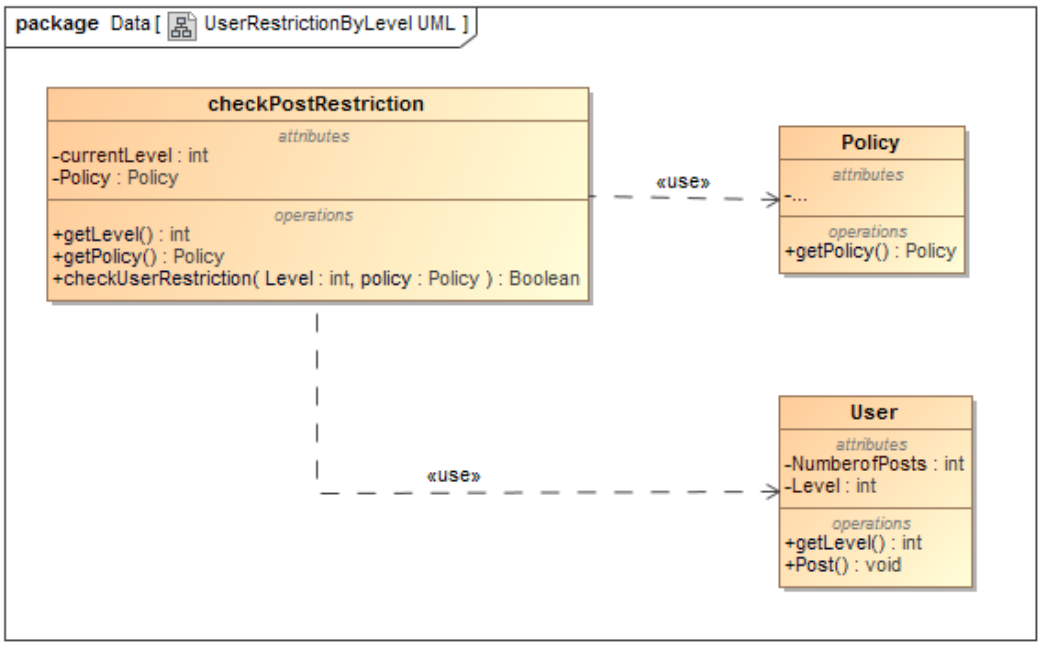


Figure 3: User Post Restriction By User Level UML

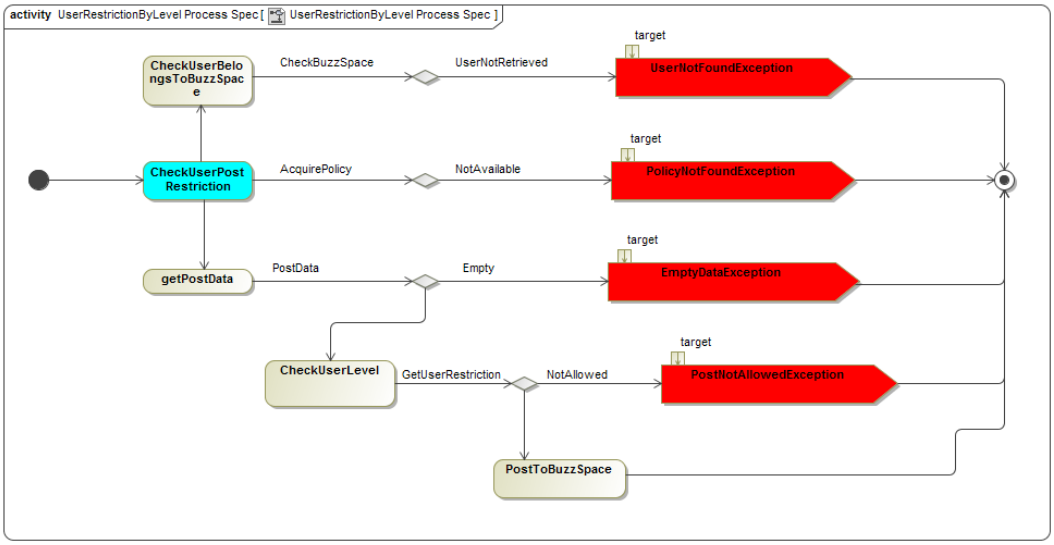


Figure 4: User Post Restriction By User Level Process Specification

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. Create automated template based messages to individual users or specified groups

   (a) Elaboration - This generates a template message about a topic to an individual user or a group.

   (b) Importance - ***

   (c) Dependency level - This depends on the user input; the system will parse through the user input and build a summary accordingly

   (d) Pre-conditions

      i. Condition 1 - Users have to be connected to buzz either logged in or guest account to get automated messages.

      ii. Condition 2 - Group must exist and have at least one member.

   (e) Post-conditions

      i. Condition 1 - Individual user gets a system generated message.

      ii. Condition 2 - Group members get a system generated message.

   (f) Requester - The system.

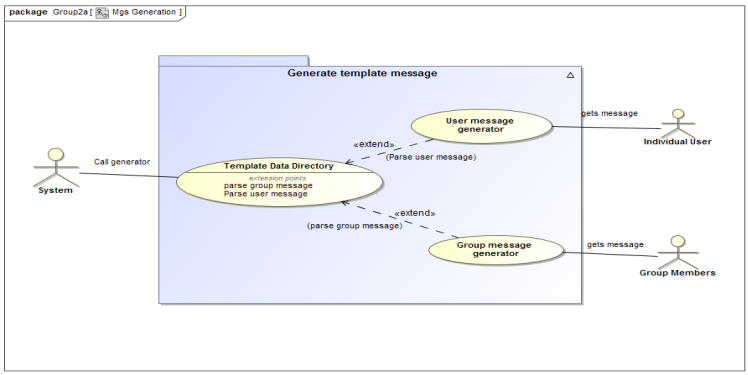


Figure 5: Generate template message for user or group.

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)

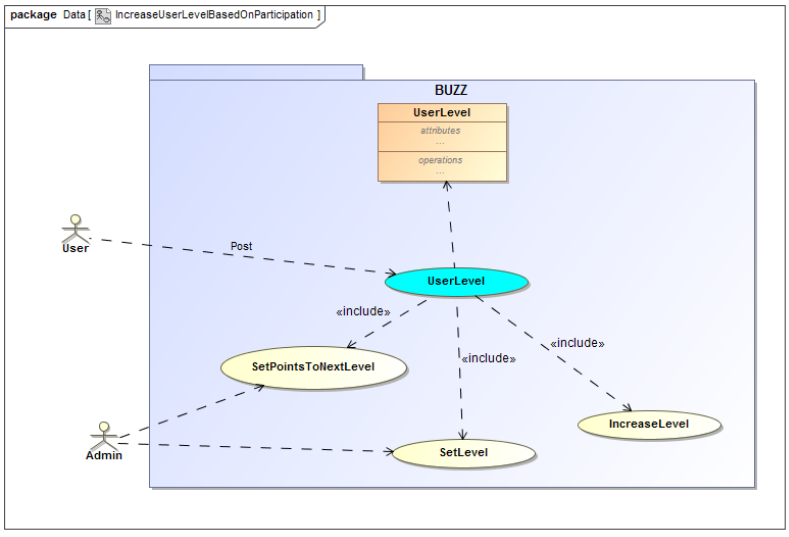


Figure 6: Automatically Change users level based on participation Use Case

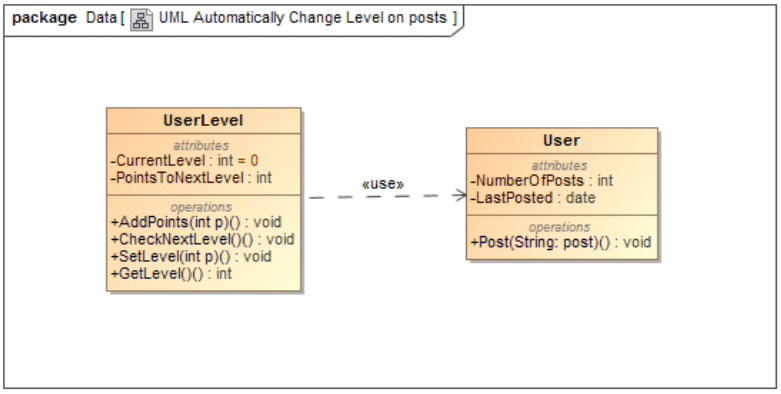


Figure 7: Automatically Change users level based on participation UML

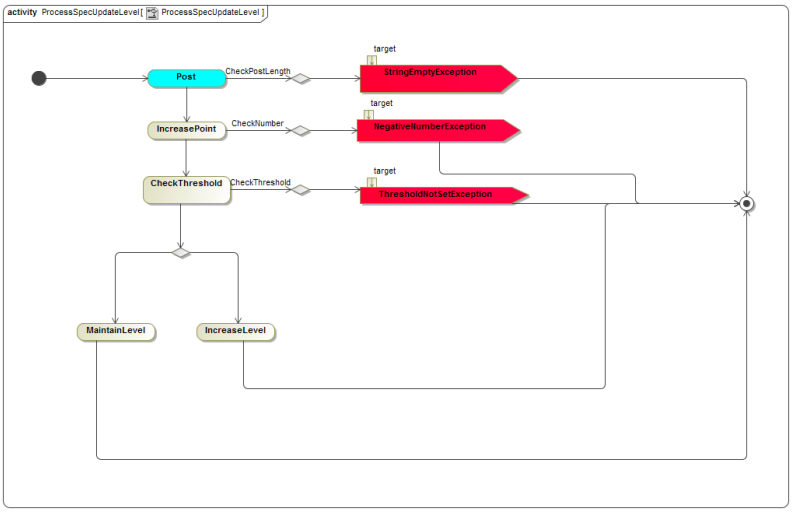


Figure 8: Process Specification for Automatically Update User Level

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

10. Provide functions such as searching and filtering.

    (a) Elaboration - This function will allow users to search for threads by topic, original poster, date etc. and filter posts by the topic such as assignments, practicals, tutorials, general etc.

    (b) Importance - **

    (c) Dependency level - The system can function without this function but it is important because it makes the system easier to use.

    (d) Pre-conditions

        i. User enters a search query
        ii. User chooses filter condition

    (e) Post-conditions

        i. User is presented with results of search query
        ii. User is presented with filtered threads

    (f) Requester - Client

11. Provide functionality to evaluate posts and vote for posts.

    (a) Elaboration - Users must be able to vote a post up or down. Buzz Space creators must be able to evaluate posts. This will result in more relevant posts getting higher ratings and thus standing out.

    (b) Importance - ***

    (c) Dependency level - Depends on the post having been created.

    (d) Pre-conditions

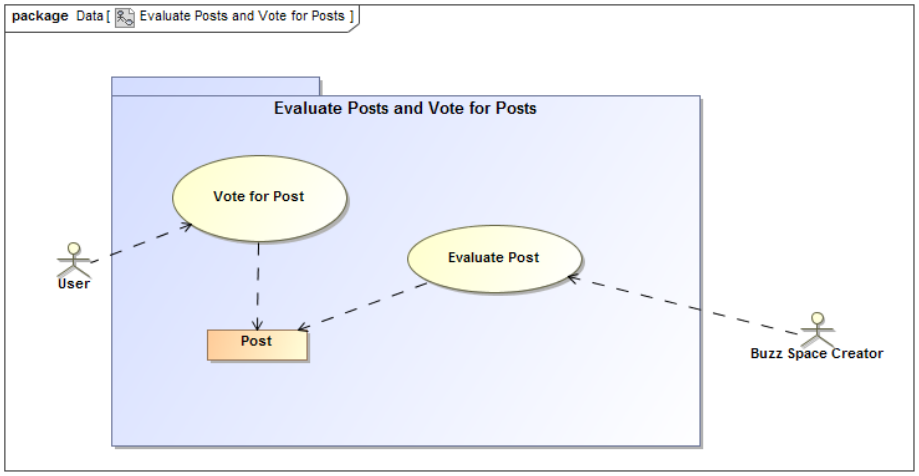
        i. Post must be created

    (e) Post-conditions

        i. Post is voted up or down.

        ii. Post is evaluated.

    (f) Requester - User



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

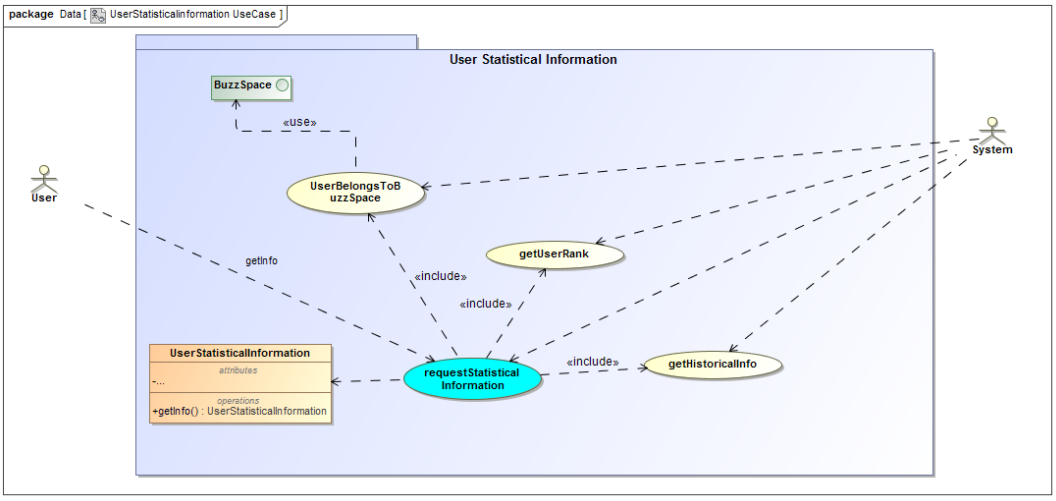


Figure 9: User Statistical Information Use Case

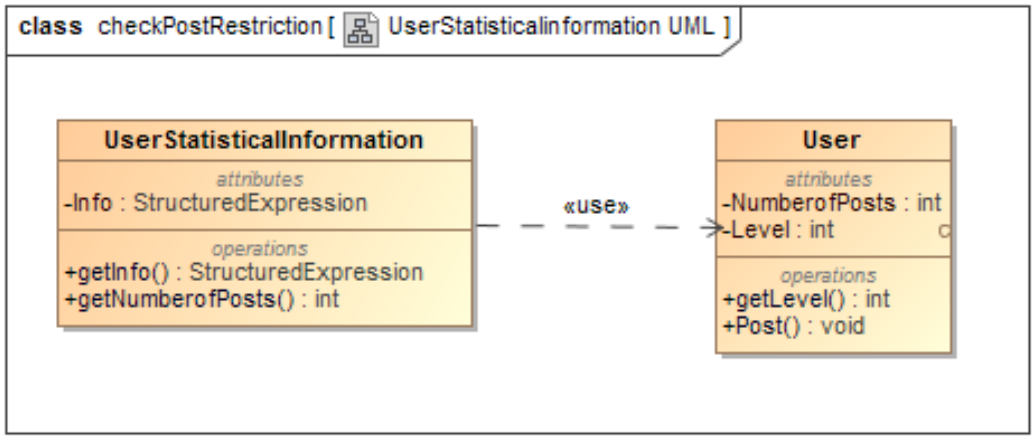


Figure 10: User Statistical Information UML

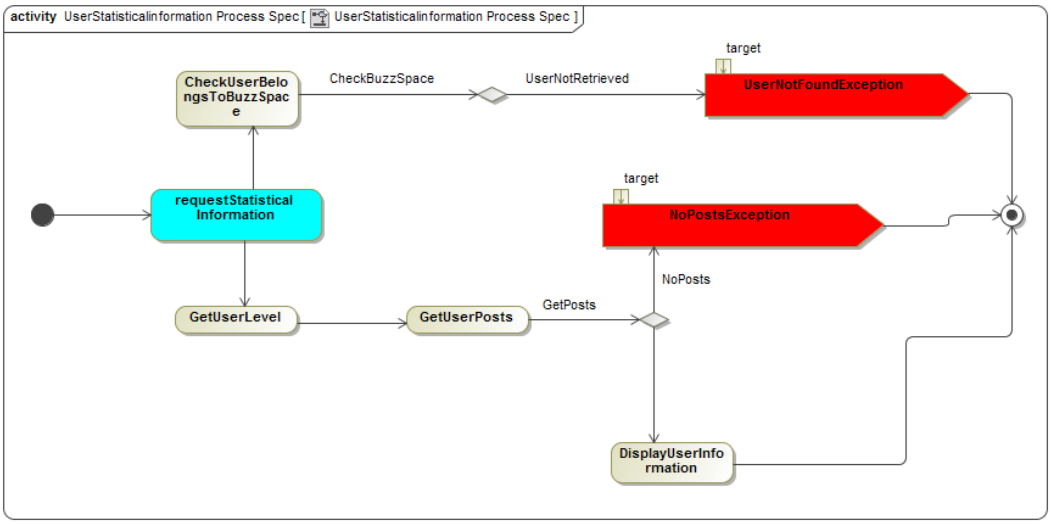


Figure 11: User Statistical Information Process Specification

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

<<<<<<< HEAD

15. =======

16. Apply self-organasation based on social tagging and allow the user to view according to the base structure, own structure or public structure ¿¿¿¿¿¿¿ ba105825c71e2e5fa7df16e61961a8fdf4ecb922

    (a) Elaboration - The user is able to use social tags that tell us more about a post thus is also able to search for a thread according to topics that have those social tags and arrange topic threads accordingly via tags. The user is also able to view posts according to the base structure given by the buzz system or specify their own structure by making use of the social tags there is also a general structure for the public who are not registered users.

    (b) Importance - ***

    (c) Dependency level - This feature depends on the user selecting tags in which to order the base structure of the posts that they see.

    (d) Pre-conditions

        i. Condition - Base structure of posts that is unsorted according to social tags.

    (e) Post-conditions

        i. Condition - Structure that is sorted according to the user's selected organisation of social tags.

    (f) Requester - The user.

¡¡¡¡¡¡¡ HEAD

=======

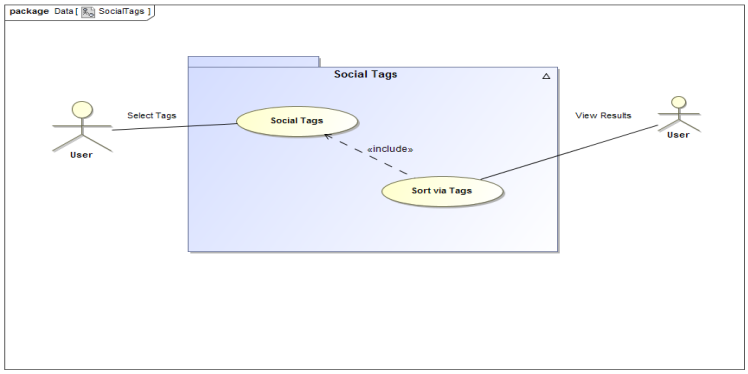


Figure 12: Self-organasation of data via social tags.

¿¿¿¿¿¿¿ ba105825c71e2e5fa7df16e61961a8fdf4ecb922

17. 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 - System, Automatically checks to see if the post is plagiarised.

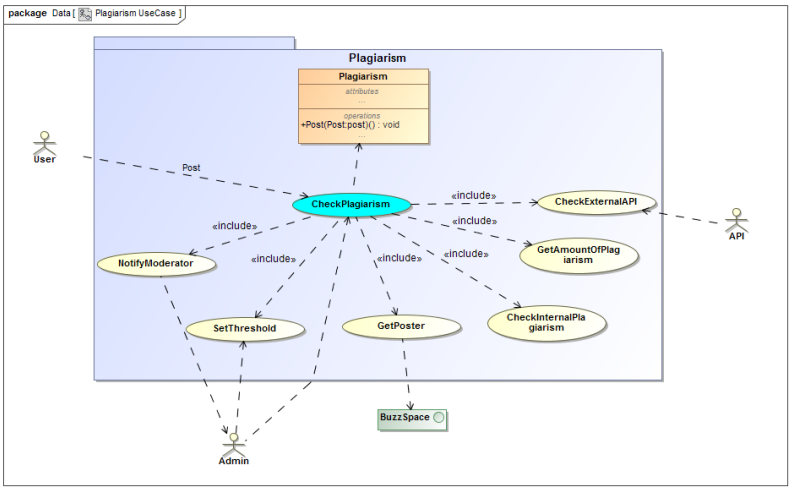


Figure 13: Plagiarism Check Use Case

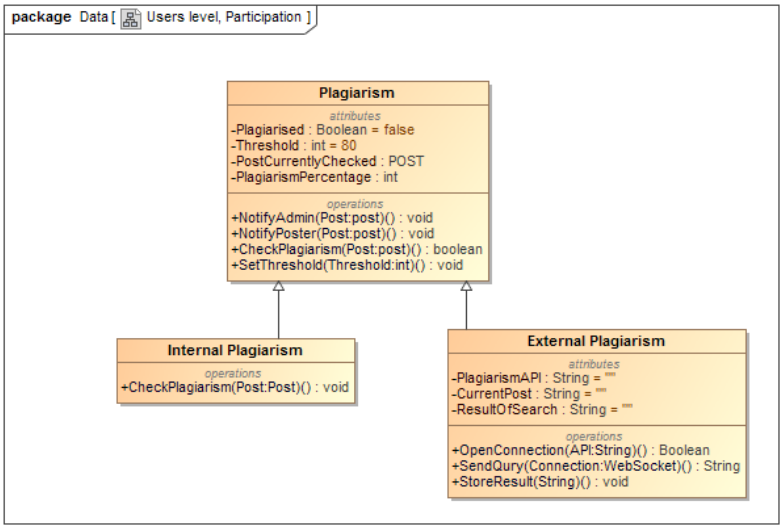


Figure 14: Plagiarism Check UML

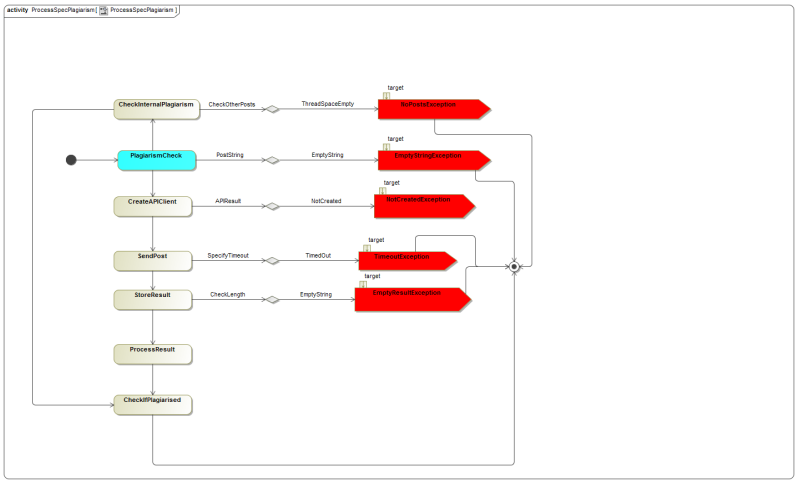


Figure 15: Process Specification for Checking Plagiarism API and Internal Checks

18. (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

    ...

5 Open Issues

6 Glossary