



UNIVERSITEIT VAN PRETORIA
UNIVERSITY OF PRETORIA
YUNIBESITHI YA PRETORIA

DEPARTMENT OF COMPUTER SCIENCE

COS 301 - SOFTWARE ENGINEERING

COS 301 - Mini Project

Author:

Hanrich Potgieter

Chris Cloete

Jason Richard Evans

Kale-ab Tessera

Lelethu Zazaza

Goodness Adegbenro

Herman Willem Keuris

William Seloma

Student number:

u12287343

u13029721

u13032608

u13048423

u13028023

u13046412

u13037618

u10155865

February 27, 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 it 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

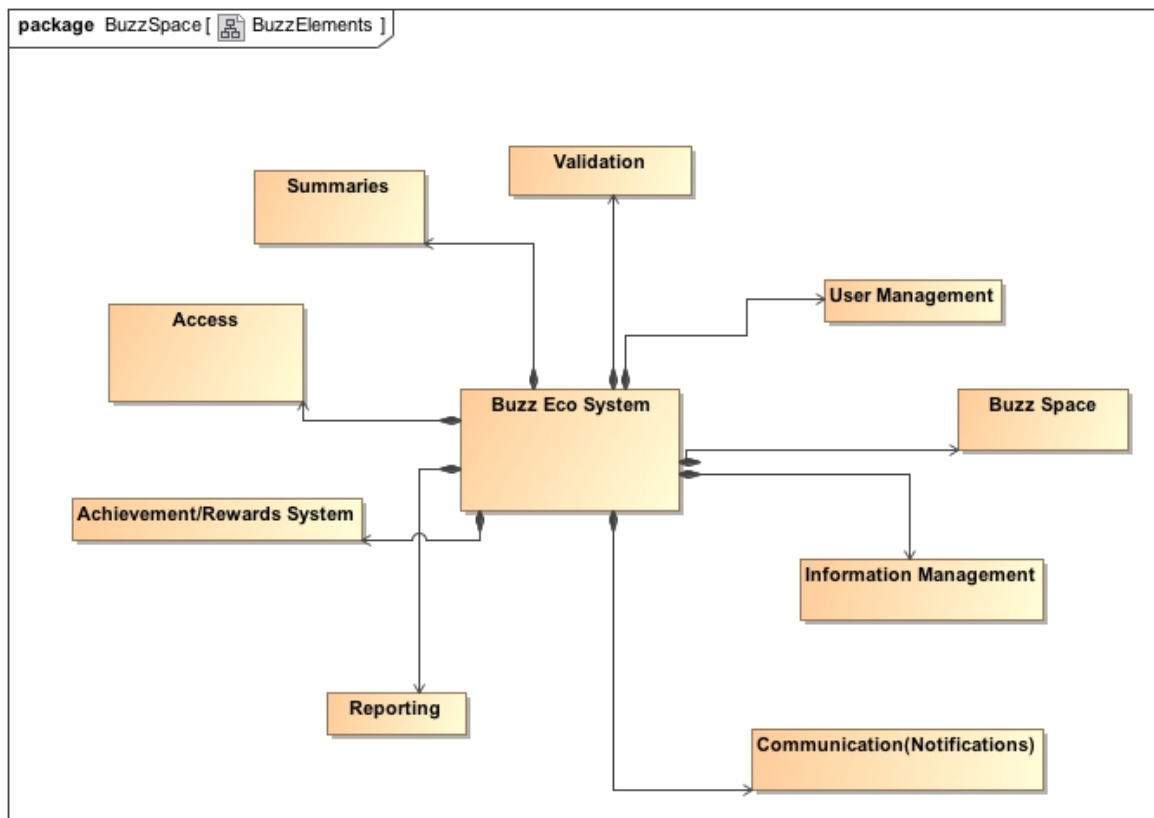
BUZZ SPACE DISCUSSIONS/MINI PROJECT

Version: Version 0.2 Alpha For further references see [gitHub](#). February 27, 2015

Contents

For further references see [gitHub](https://github.com/DieBaber/COS301-GROUP6-A.git) or got to the link <https://github.com/DieBaber/COS301-GROUP6-A.git>

1 Functional requirements



1.1 Introduction

We use this document to give a high level overview of the buzz discussion board. We have identified the various components of our system. The purpose of this document 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 on how we will achieve a system that is both scalable and pluggable. We have

identified the use cases of the various components of the discussion board and helped expand on them.

1.2 Use case prioritiation

Critical

- BuzzSpace
- CRUD posts(Creating,Reading; Updating; Deleting).
- System Access
- Information Management

Important

- User Management
- Communication(Notifications)
- Reporting

Nice-To-Have

- Achievement/Rewards System
- Reporting
- Summaries

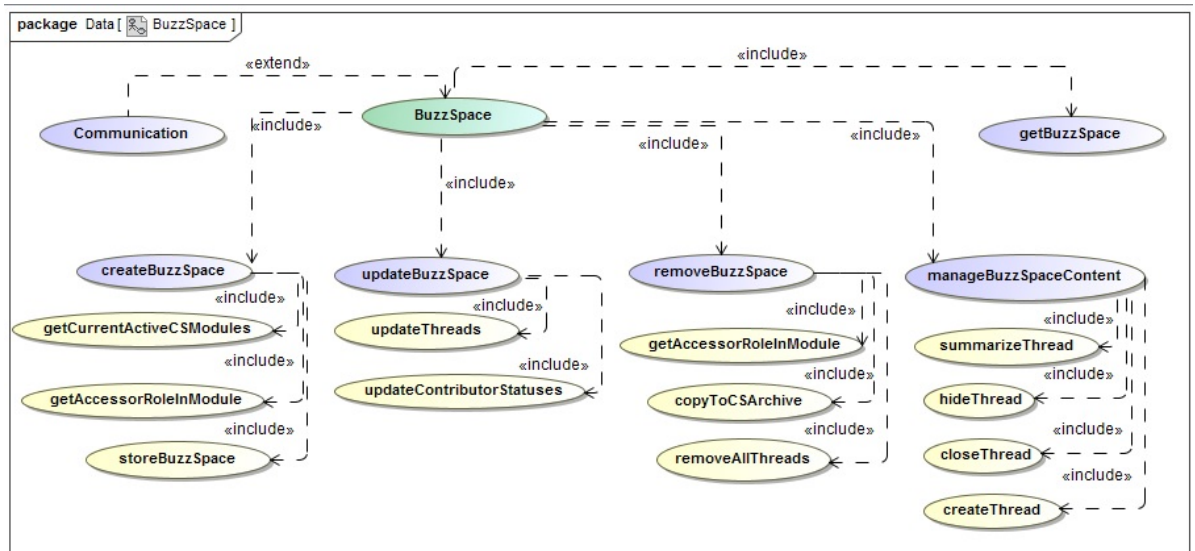
1.3 Use case/Service contracts

Use Case	Pre Condition	Post Condition	Description
BuzzSpace	There must be a valid user	User must still exist	This use case provides an interface that facilitates management of threads
Information Management	User specific information must be well documented (e.g. number of logins) and two or more threads, posts and tags must exist before their relevant search and display functions can be used	Anyone accessing the buzz space can view the profile of any registered user and search and display lists of the threads, posts and tags that have so far been created.	This use case provides an interface that allows for the easy viewing of logically sorted information generated by the buzz space and other users (such as threads, posts and tags). It also allows for a simple display of individual user information (profiles)

Communication	A user needs to be registered in order to have notifications sent to his profile inside the application. For e-mails to be sent, a valid and up-to-date e-mail address is needed on the user database.	A notification should visibally be highlighted in the application with appropriate messages. In some cases, an email is sent out from the system.	This use case specifies all the functions that the Buzz system needs to have in order to communicate important information with the user.
Summaries	Although not strictly required it is very helpful to have a large number of posts on the thread and to already have a few preaviously generated thread summaries. Some user iput is usually required and all posts must make use of relevant tags.	The semi-autonomous thread summary generator will attempt to generater parts of the thread summary (or atleast suggest possible words and phrases to be used)	This use case attempts to generate a thread summary with as little user intervention as possible. However in most cases the user will still be required to contribute some input.
Achievement Rewards System	A user's level requires Achievements to be allocated and/or rewards to be awarded	Achievements are allocated and/or rewards are awarded	This use case provides a system that allocates achievements to users based on their levels and the votes they aquired. it also provides a system that awards rewards to users based on their achievements.
Access	The user will need a browser to view the website.	Threads and posts are displayed in descending order by date.	Details how an end-user will be able to access the Buzz system.
Validation	Post is palgiarised and/or does not follow netiquette	Post is valid against rules	
User Management	If the User is not involved in the course (not a registered student/ tutor/ teaching assistant).	The User is still involved in the course	This is a basic system which manages the User's Login and logout.
Reporting	Data must be available to report on.	Data must not be corrupt.	This use case generate report for all actors

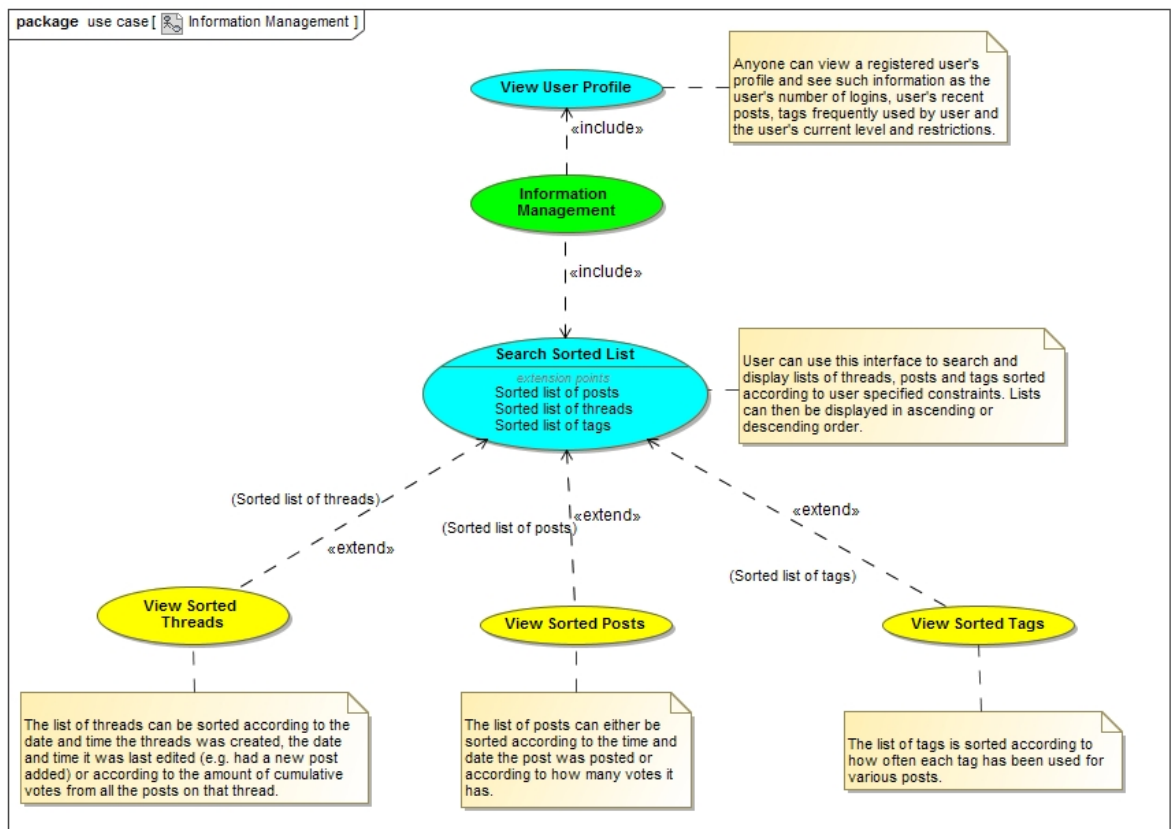
1.4 Required functionality

- **BuzzSpace.** A Buzz Space is a integral component of the Buzz System which facilitates the management of threads added by its users. Buzz Spaces may be created for each active module in the Computer Science Department in order to promote intuitive communication between the Computer Science staff and its students.



- package Data [Validation]
-
- ```
graph TD
 subgraph Validation
 V([Validation
API Plagiarism checker
Print report])
 C1([Check if user status is
valid for post])
 C2([Check post for plagiarism])
 C3([Check netiquette
errors])
 P([Print reports])
 O([Online plagiarism
checker API])
 end
 V -.->|«include»| C1
 V -.->|«include»| C2
 V -.->|«include»| C3
 V -.->|«extend»| P
 P -.->|«extend»| O
 V -.->|«extend»| O
```
- The diagram illustrates the structure of the Validation package. The central use case, **Validation** (red oval), contains the components **API Plagiarism checker** and **Print report**. It includes three other use cases: **Check if user status is valid for post**, **Check post for plagiarism**, and **Check netiquette errors** (all pink ovals). Additionally, the **Validation** use case extends **Print reports** (green oval) with the parameter **(Print report)**. The **Print reports** use case further extends the **Online plagiarism checker API** (green oval) with the parameter **(API Plagiarism checker)**. The **Validation** use case also extends the **Online plagiarism checker API** use case. Two yellow note boxes provide additional context: one notes that the **Check if user status is valid for post** use case ensures users can post certain content, and another notes that the **Check post for plagiarism** use case utilizes the API to detect plagiarized content.

- 8

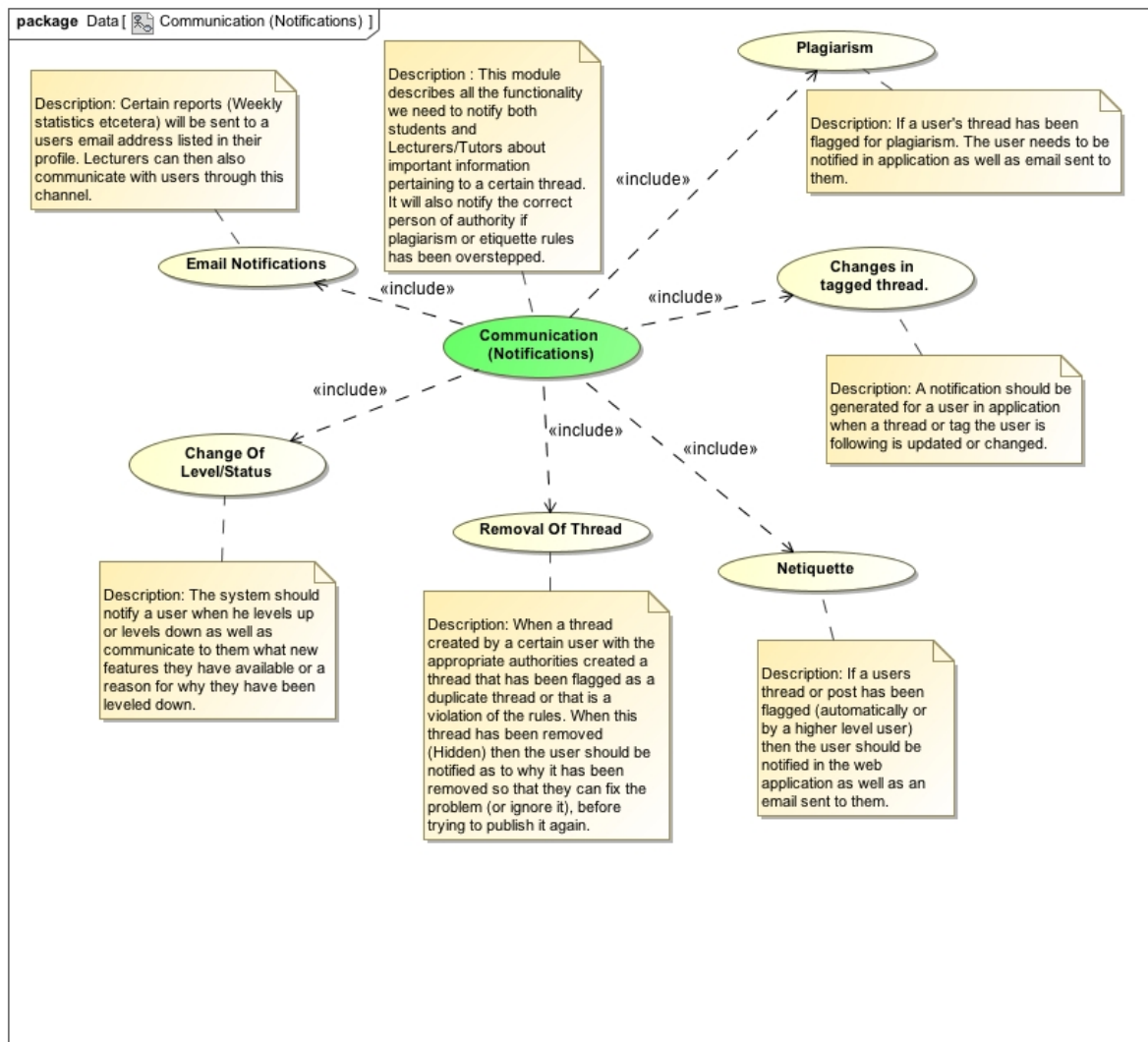


- Reporting.** We will use the reporting module to generate quite a few reports regarding the Buzz Space system. It will be a key player in adding value to lecturers and students. Each student can easily general a report regarding their own contributions towards a Buzz Space. Lecturers will be able to grade student performance and see how much plagiarism has occurred. The system administrators will be able to check for system bugs and see error logs.



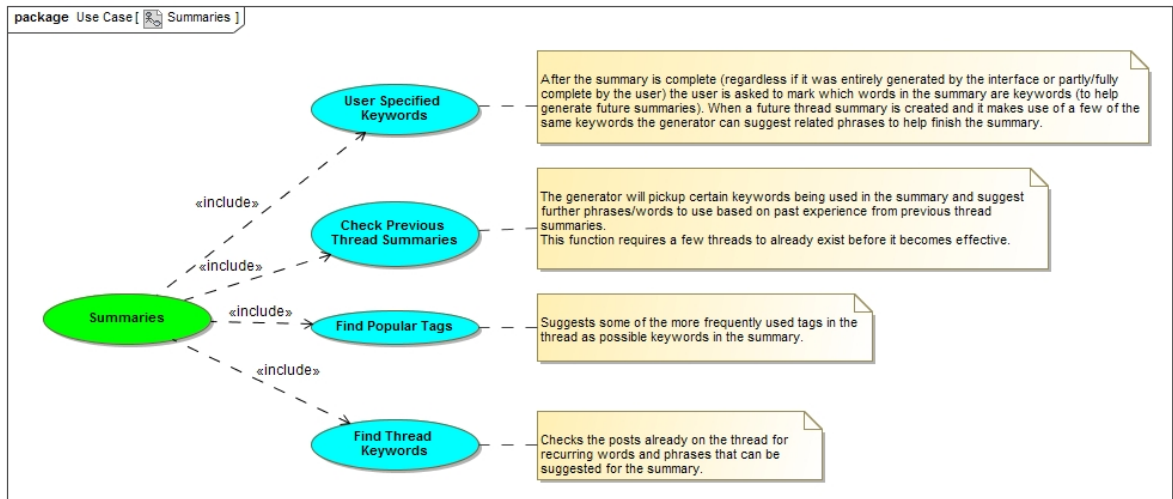
## • Communication (Notifications)

This module describes the way the Buzz System will communicate with its users inside of the application as well as sending information and/or reports from the Buzz system to an external system such as email notifications.



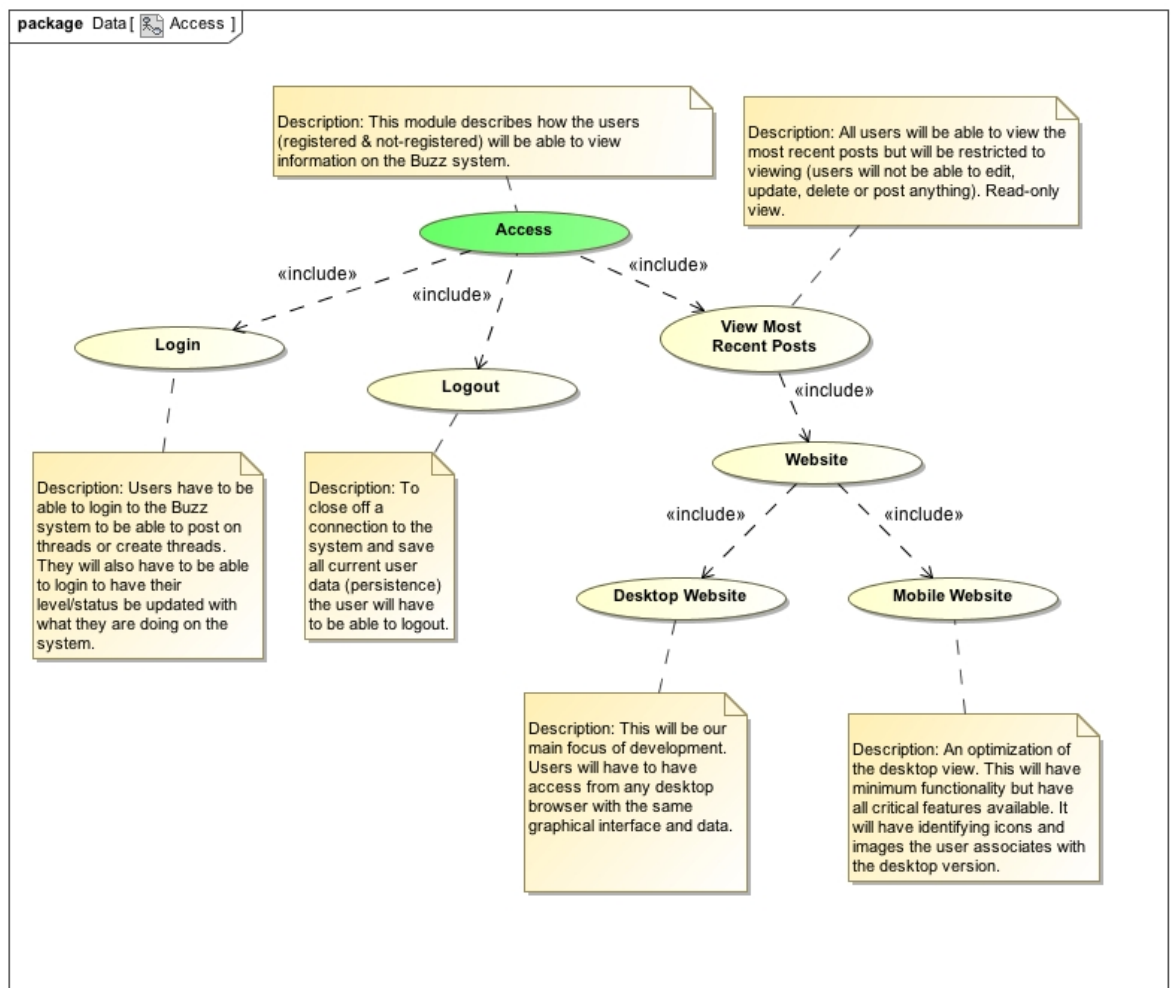
## • Summaries

This module uses a semi-autonomous thread summary generator to help the user create thread summaries by checking the posts in the thread for frequently used words and phrases which it can then suggest to the user to be used in the summary. It also suggests tags that are used often within the thread as possible keywords and checks previously generated thread summaries for any keywords and phrases which might be relevant to the creation of the new thread summary and suggests them to the user. After the completion of the thread summary the user is asked to mark which words in the summary are keywords; this makes generating all subsequent thread summaries easier as it allows for quicker comparisons between summaries (and better word/phrase suggestions by the generator).



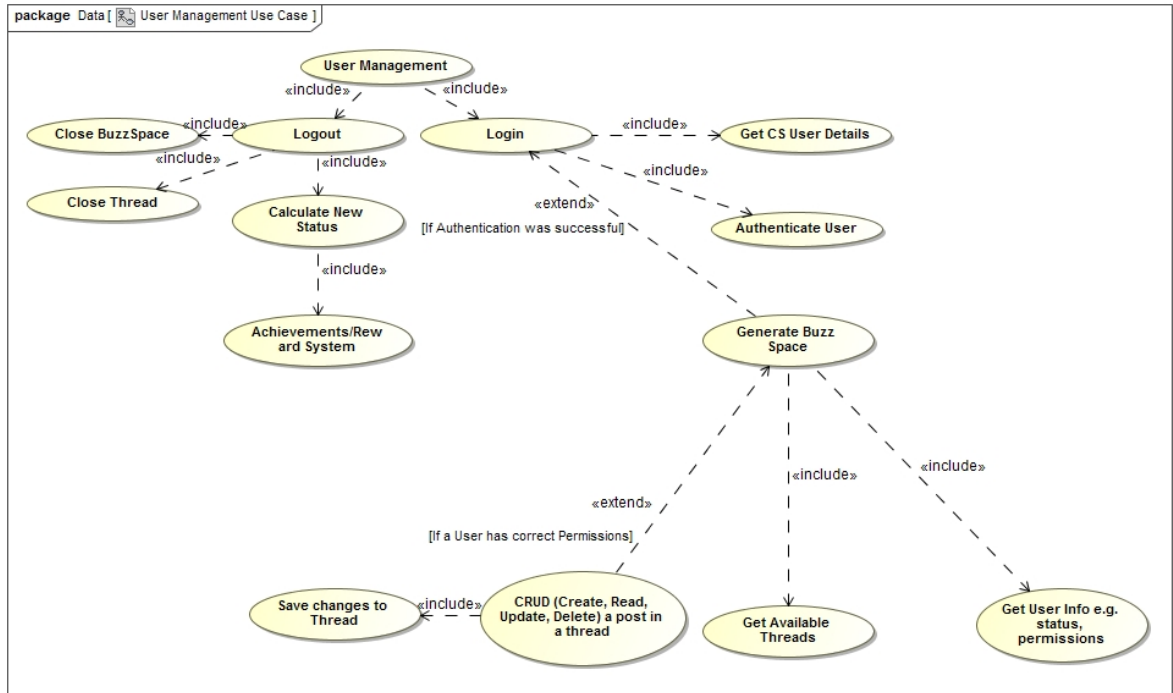
## • Access

The use case below shows how a end-user will be able to access the Buzz system. Although a user may not be registered, they should still be able to view and read threads and posts.



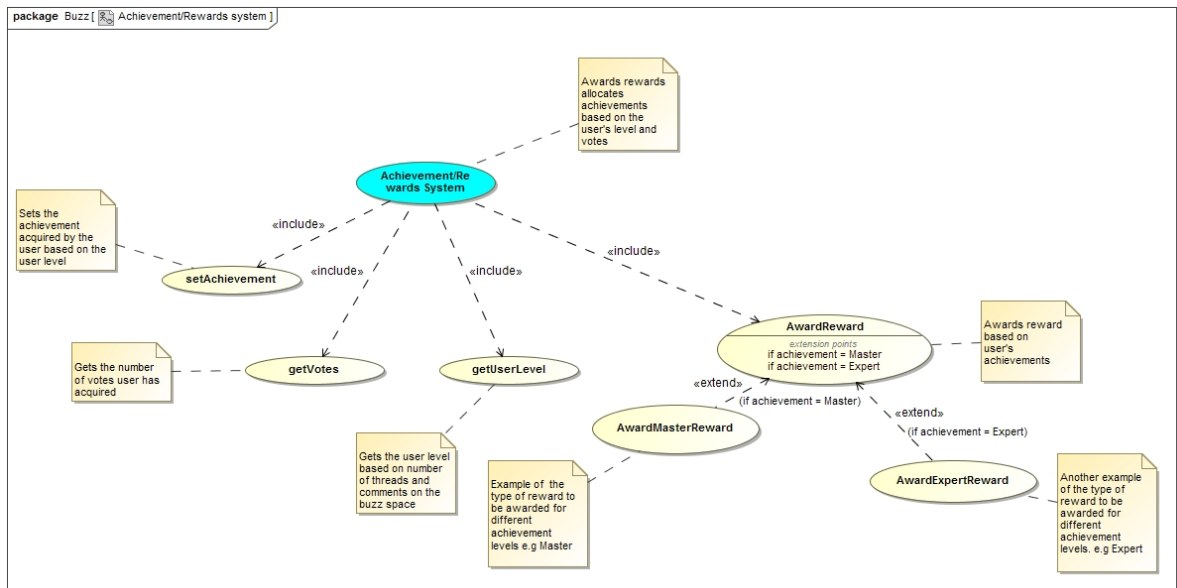
- **User Management**

This use case specifies how the user's themselves will be managed (not their data). This refers to who will be allowed to firstly login, access a buzzspace and then a specific thread and eventually logout. This displays what services help acheive User Management and also which servies are subsets of other services.



- **Achievement/Rewards system**

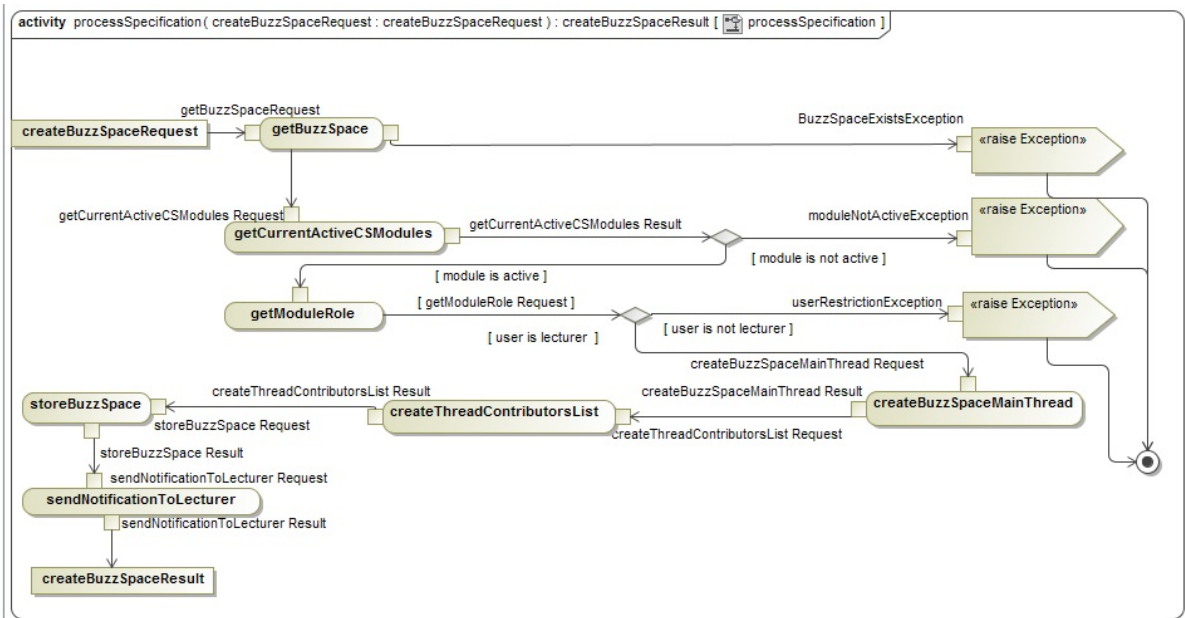
The Achievement/Rewards system use case component shows how the Buzz System generates and awards rewards to users, based on their different achievements. The achievement is derived from each user's level of participation on the Buzz Space as well as the number of votes they acquire. The Achievement/Reward system incorporates the gamification functionality of the Buzz System. Therefore, forcing the users of the system to participate more often as there will be rewards for this.



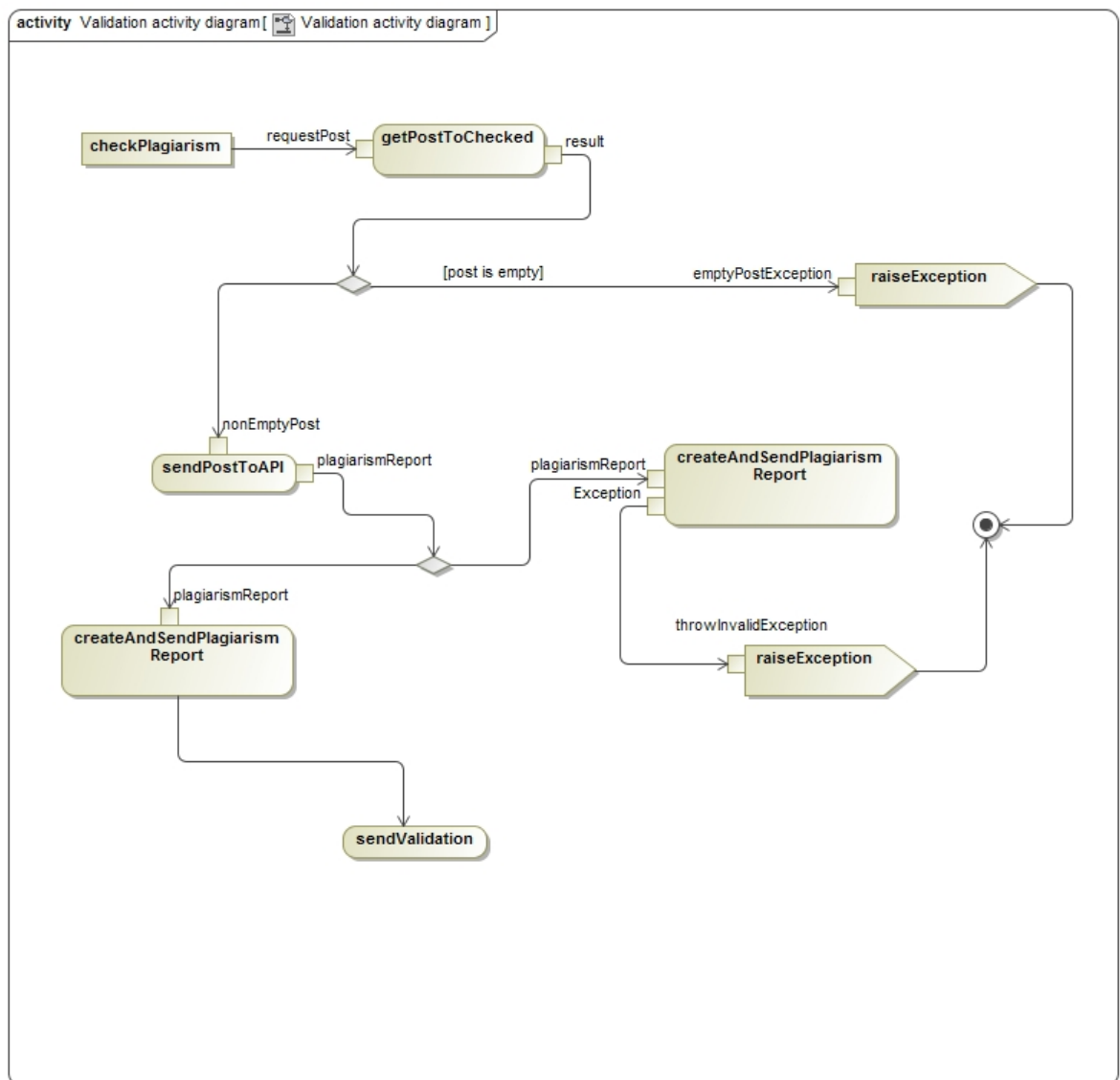
## 1.5 Process specification

We want to show various important process specification of our recommendation.

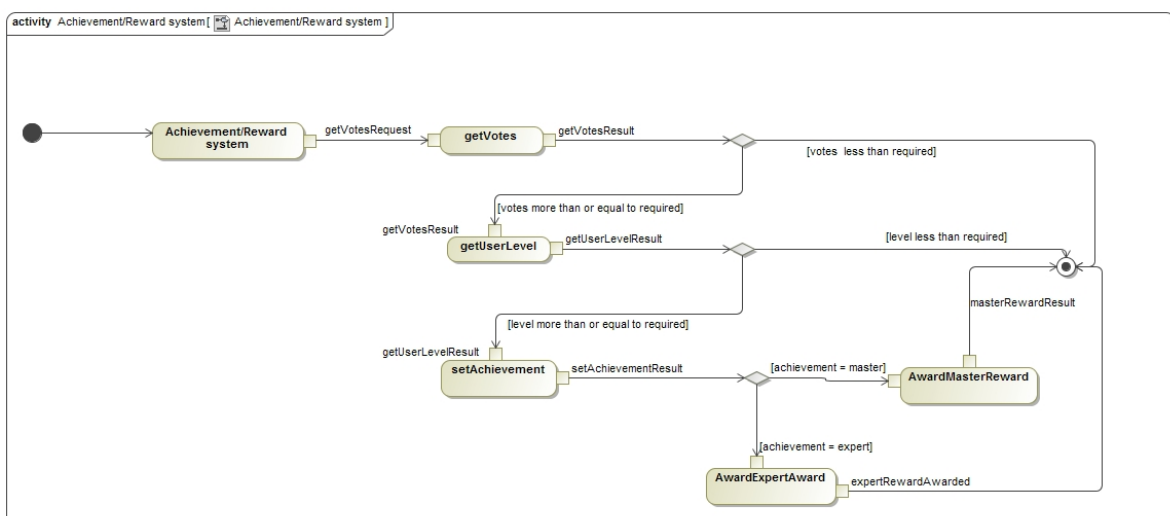
- CreateBuzzSpace



- Validation



- Achievement/Rewards system





## 1.6 Domain Model

