

FUNCTIONAL REQUIREMENTS SPECIFICATION

FOR

Buzz System

**Mini Project Phase 1
GitHub**

https://github.com/RenaldoV/COS301_Group5_a

Khathutshelo Shaun Matidza 11072157
Renaldo van Dyk 12204359 Andreas du Preez 12207871
Sean Hill 12221458 Kgomotso Sito 12243273
Hlavutelo Maluleke 12318109 Sboniso Masilela 10416260
Semaka Malapane 13081129

Contents

1	Introduction	4
2	Use case prioritization	5
2.1	Critical	5
2.2	Important	5
2.3	Nice-To-Have	5
3	Use case/Services contracts	6
3.1	Pre-Conditions	6
3.2	Post-Conditions	7
3.3	Request and Results Data Structures	7
4	Required functionality	8
5	Process specifications	14
6	Domain Model	22

1 Introduction

This document discusses the application functionality required by users (and other stakeholders).

2 Use case prioritization

2.1 Critical

- 1.1 Delete
- 1.2 Read
- 1.3 Create
- 1.4 Update
- 1.5 Comment
- 1.6 Sign-in
- 1.7 Sign-up

2.2 Important

- 1.1.1 Archive post
- 2.1 Direct messages
- 3.1 The system to have a structure of some sort
- 4.1 Post subject
- 5.1 Unauthorised usage of personal information, e.g Emails.

2.3 Nice-To-Have

- 1.1 View visualized statistical information
- 2.1a The user to choose their own structure
- 2.1b Making use of tags
- 3.1 No sarcasm

3 Use case/Services contracts

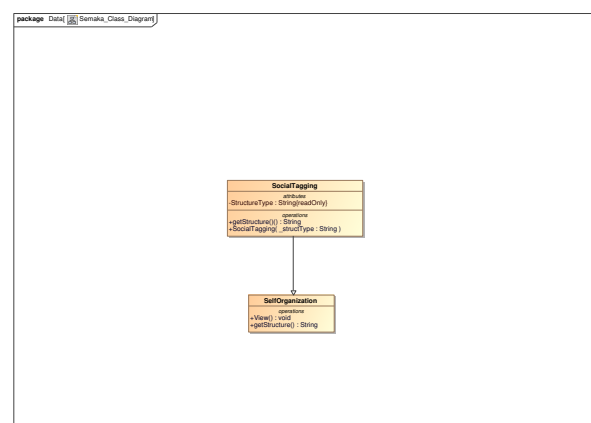
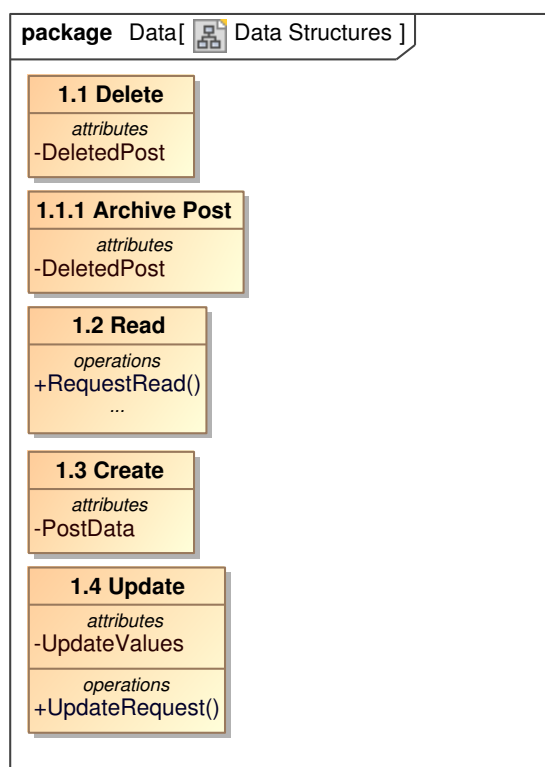
3.1 Pre-Conditions

- 1.1 a) Post must be in database
- 1.1 b) Post must be read from database before it can be deleted
- 1.1 c) User must be logged in to the system to delete a post
- 1.1.1a) Post must be deleted before it can be archived
- 1.2 a) Post must exist in database
- 1.2 b) User must be logged into the system to read values from database
- 1.3 a) User must be logged into the system to create posts to in database
- 1.4 a) User must be logged into the system to update the database
- 1.4 b) Before the database can be updated values must be changed
- 2.1 a) User must have registered account
- 2.1 b) User must have message
- 2.1 c) Message must have (a) recipient(s)
- 3.1 a) User must have registered account
- 3.1 b) User must have been graded
- 4.1 a) Check user status level before posting
- 4.1 b) Check word count of post before posting
- 5.1 a) Check user status level before commenting
- 5.1 b) Check word count of post before commenting
- 6.1 a) Users must be registered to be able to view content/posts according to their own structure
- 7.1 a) User not registered, access denied.
- 7.1 b) User registered but not signed in.
- 7.2 c) For plagiarism detection, if post does not conform to netiquette rules.

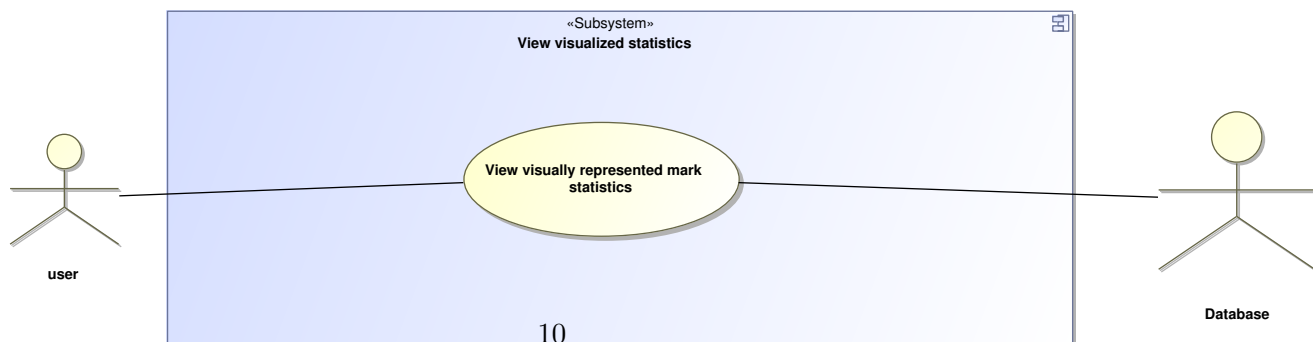
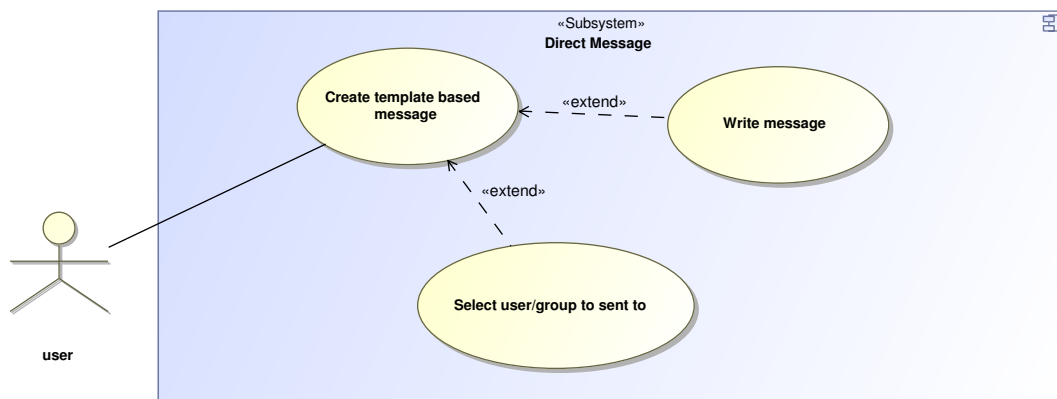
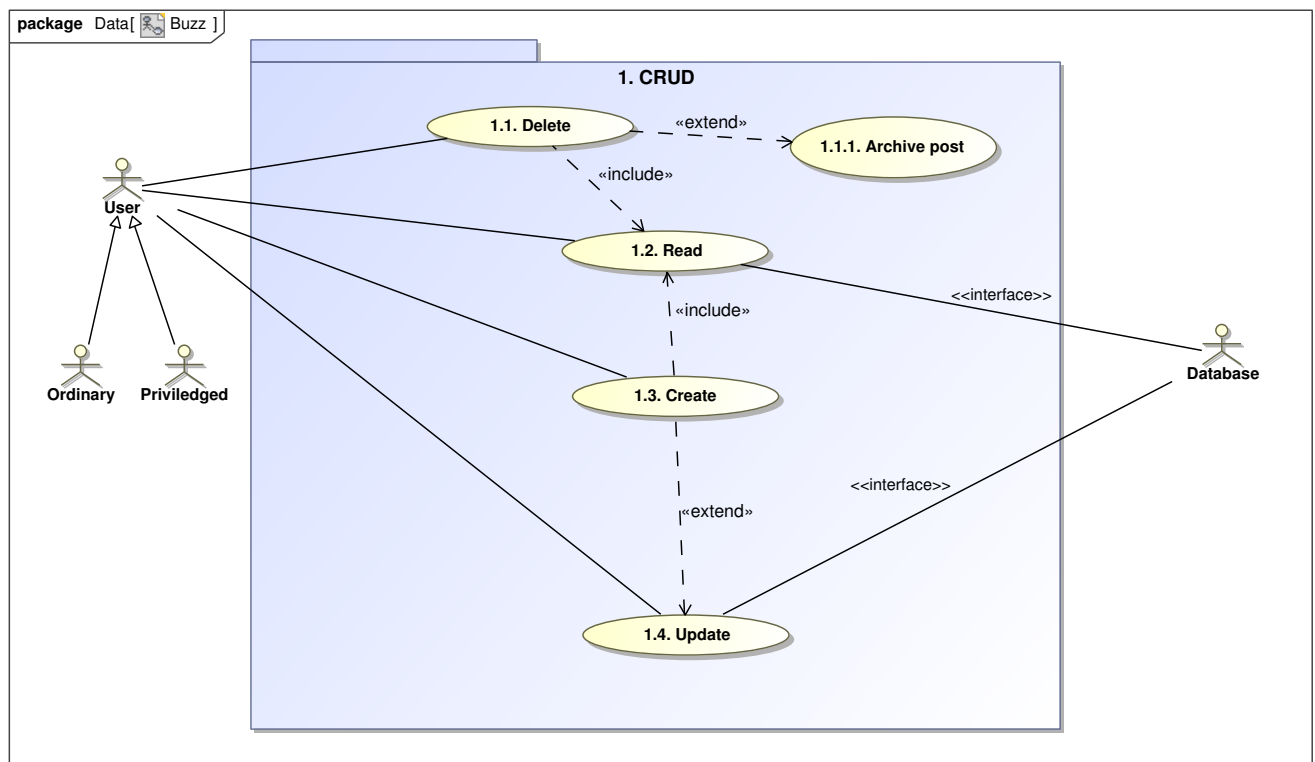
3.2 Post-Conditions

- 1.1 a) Post will not exist in database
- 1.1 b) Post will not be visible to users
- 1.1.1a) Post and log data will reside in an archive
- 1.3 a) Update request will be called on the new data
- 1.3 b) Post will be visible to users who are logged into the system
- 1.4 a) Post data will be inserted into database
- 1.4 b) Post will exist in database
- 2.1 a) A direct message is sent by user and recieved by recipients
- 3.1 a) User views visualized statistical information
- 4.1 a) Comment should be visible to all users of that level
- 5.1 a) Posts should be organised according to tags
- 5.1 b) Registered users should be able to choose the structure in which to view tags
- 6.1 a) Proceed to scan post use case if registered and already signed in.
- 6.2 a) Generate report and reply data after scanning posts.
- 6.3 a) Return report about netiquette rules.

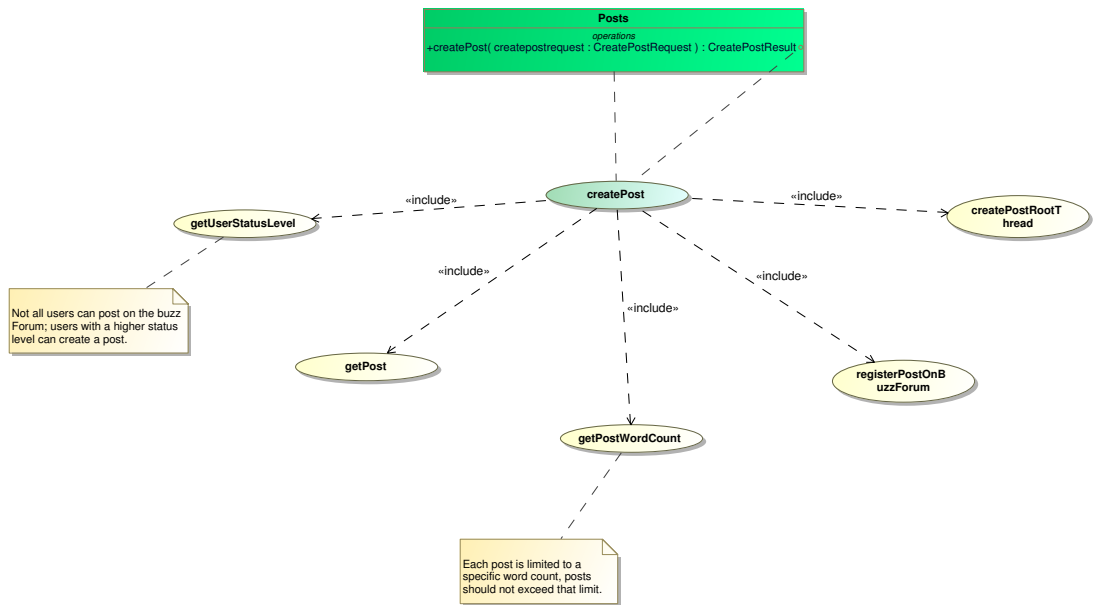
3.3 Request and Results Data Structures

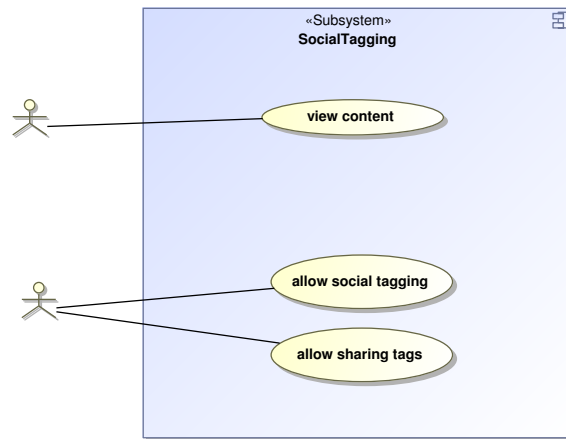


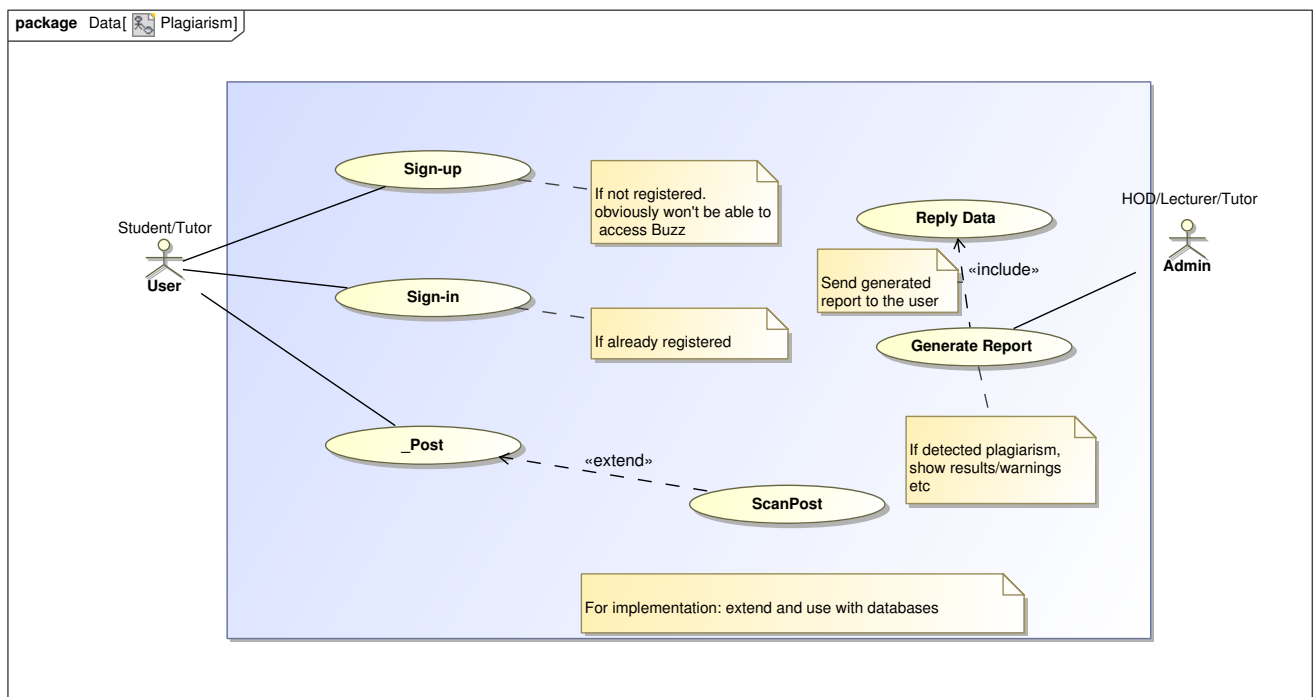
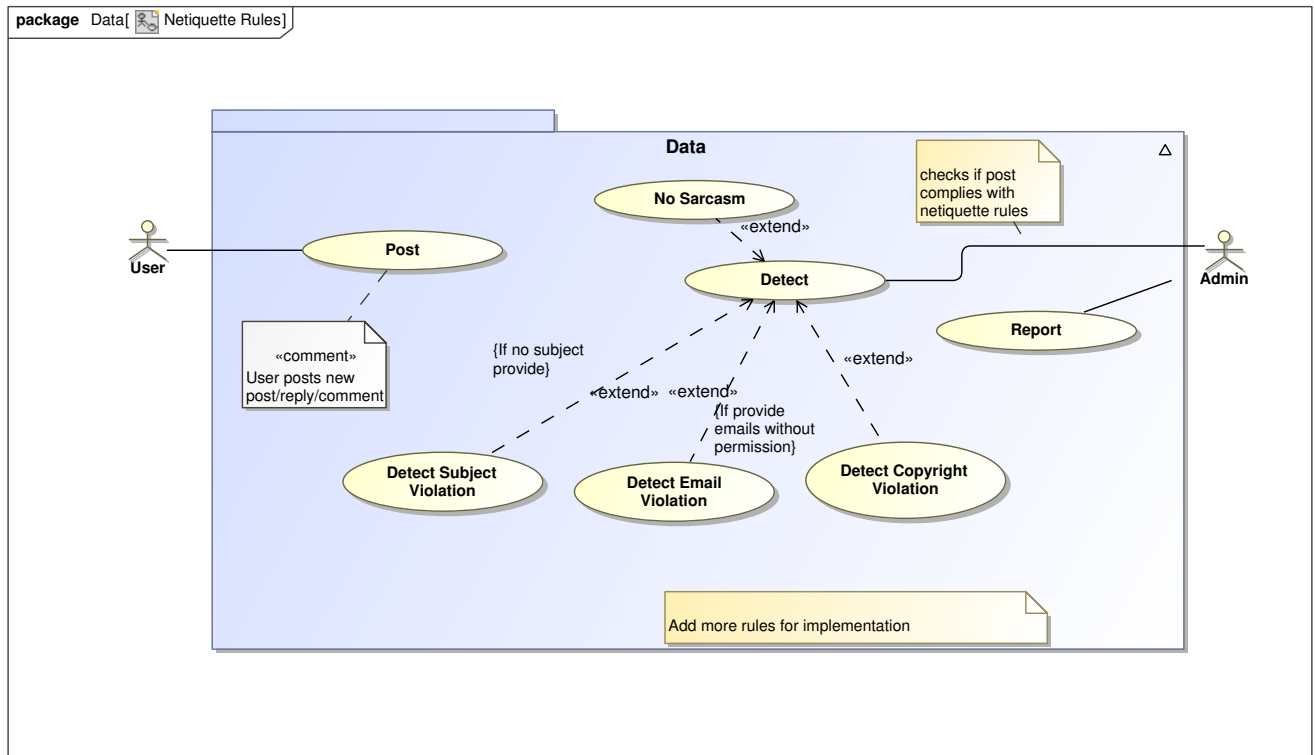
4 Required functionality




use case createPost[functionalRequirements]

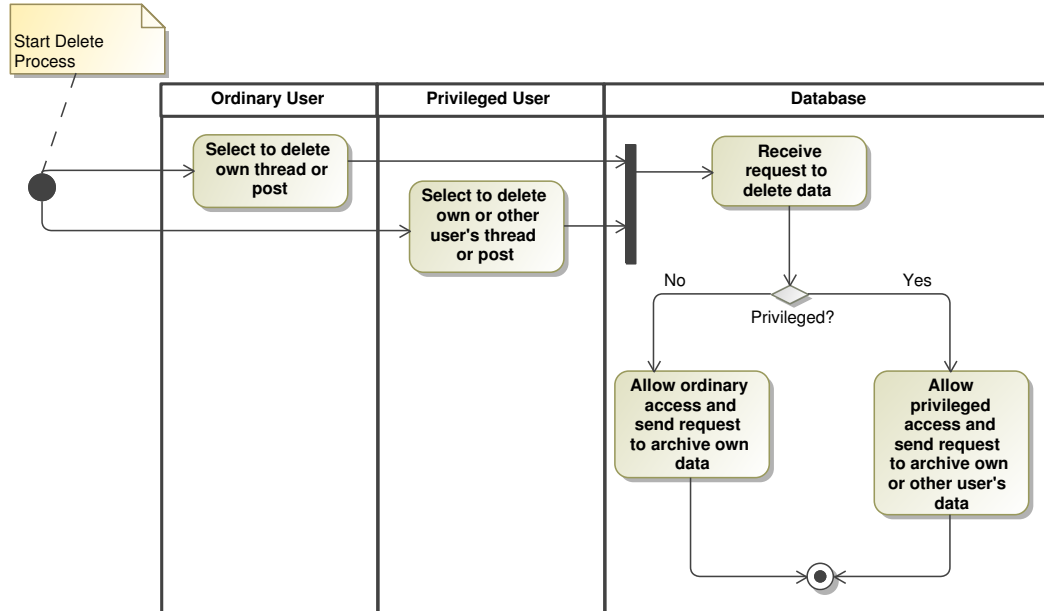





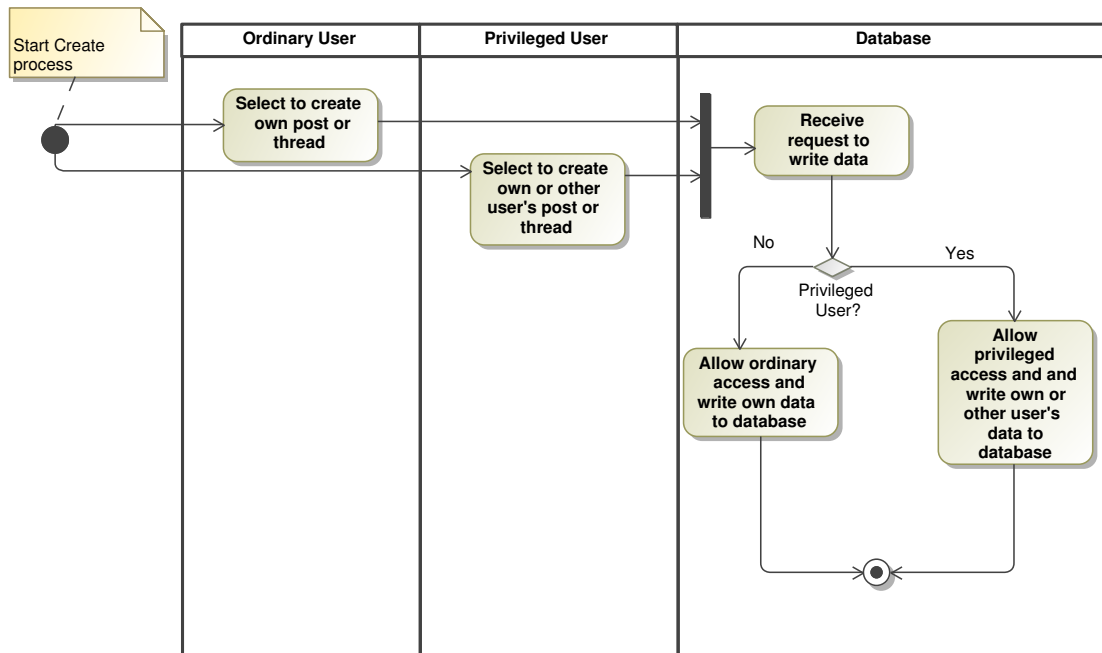


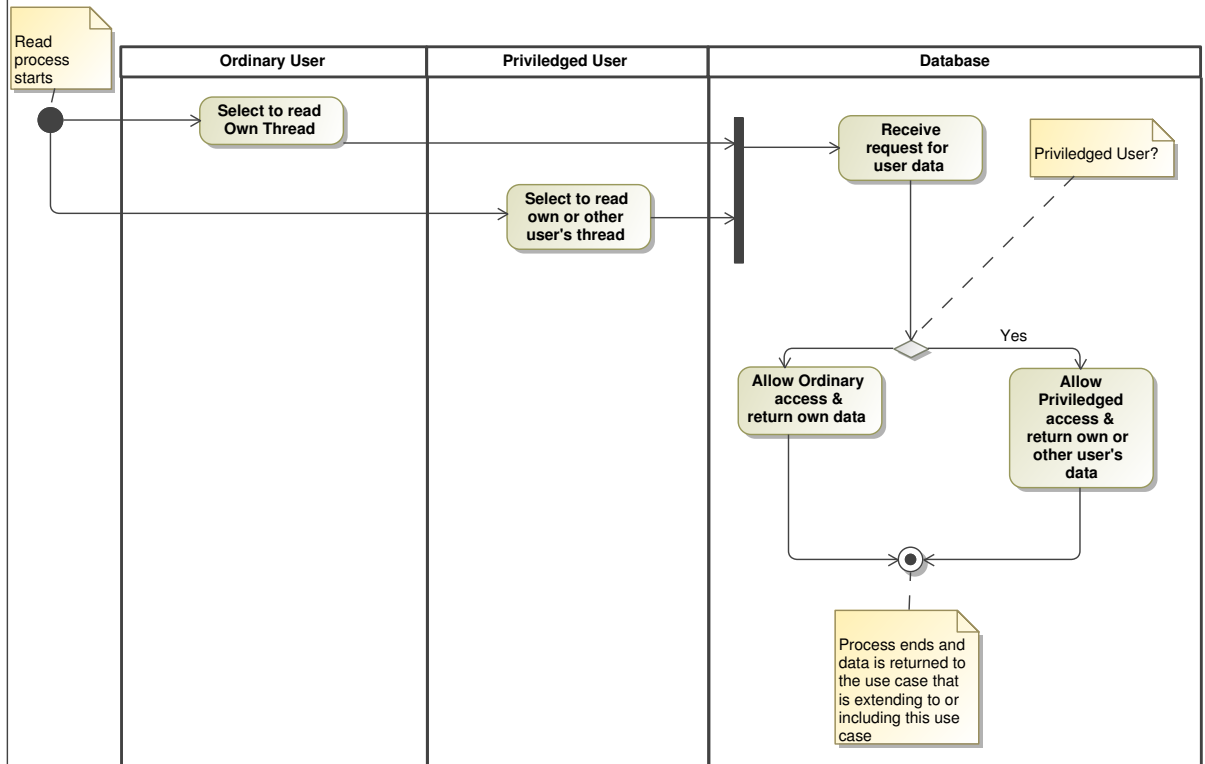
5 Process specifications

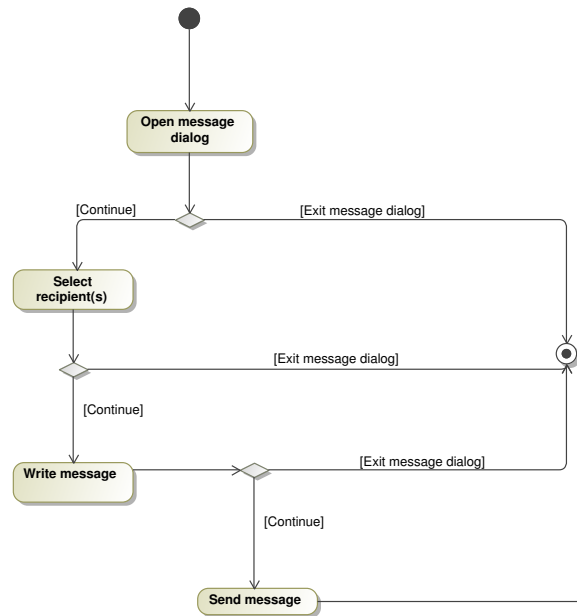
activity Process specification 1.1 [ Process specification 1.1]



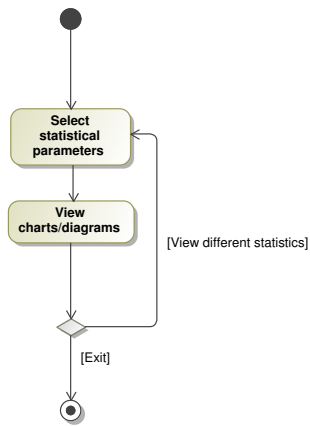
activity Process Specification 1.3 [ Process Specification 1.3]



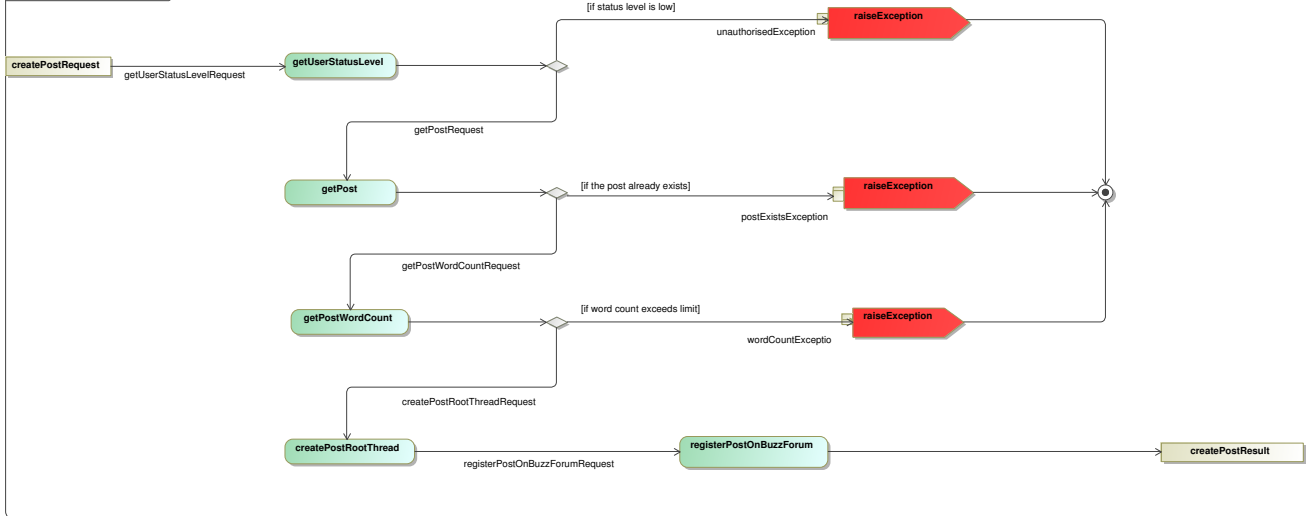


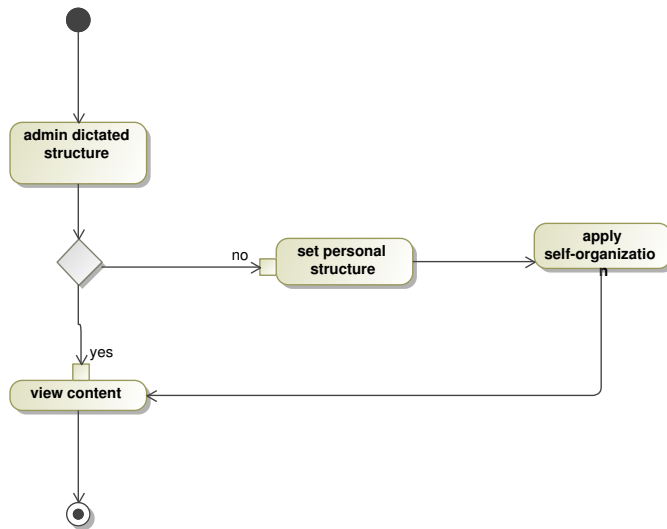


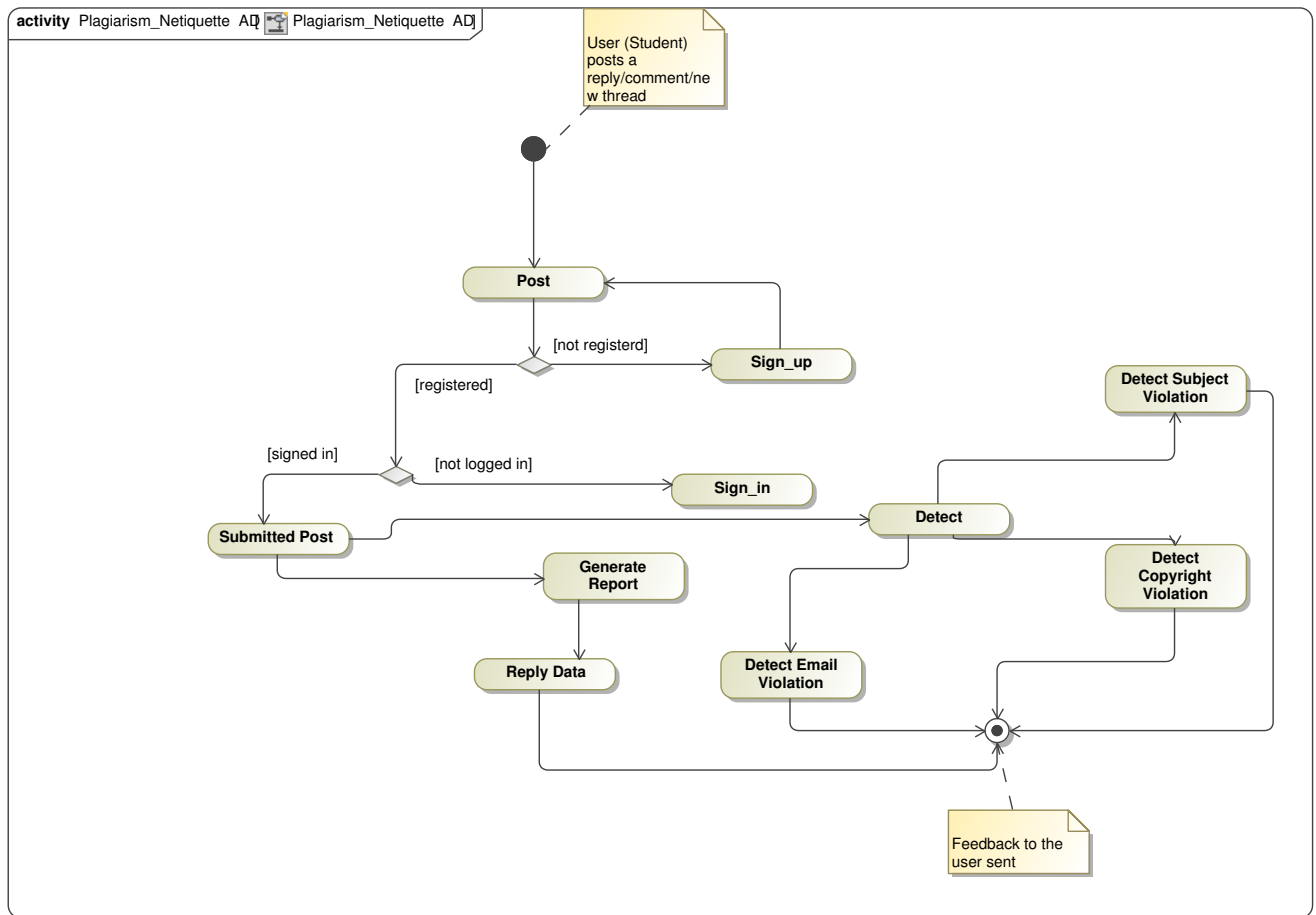
activity View visually represented statistics [View visually represented statistics]



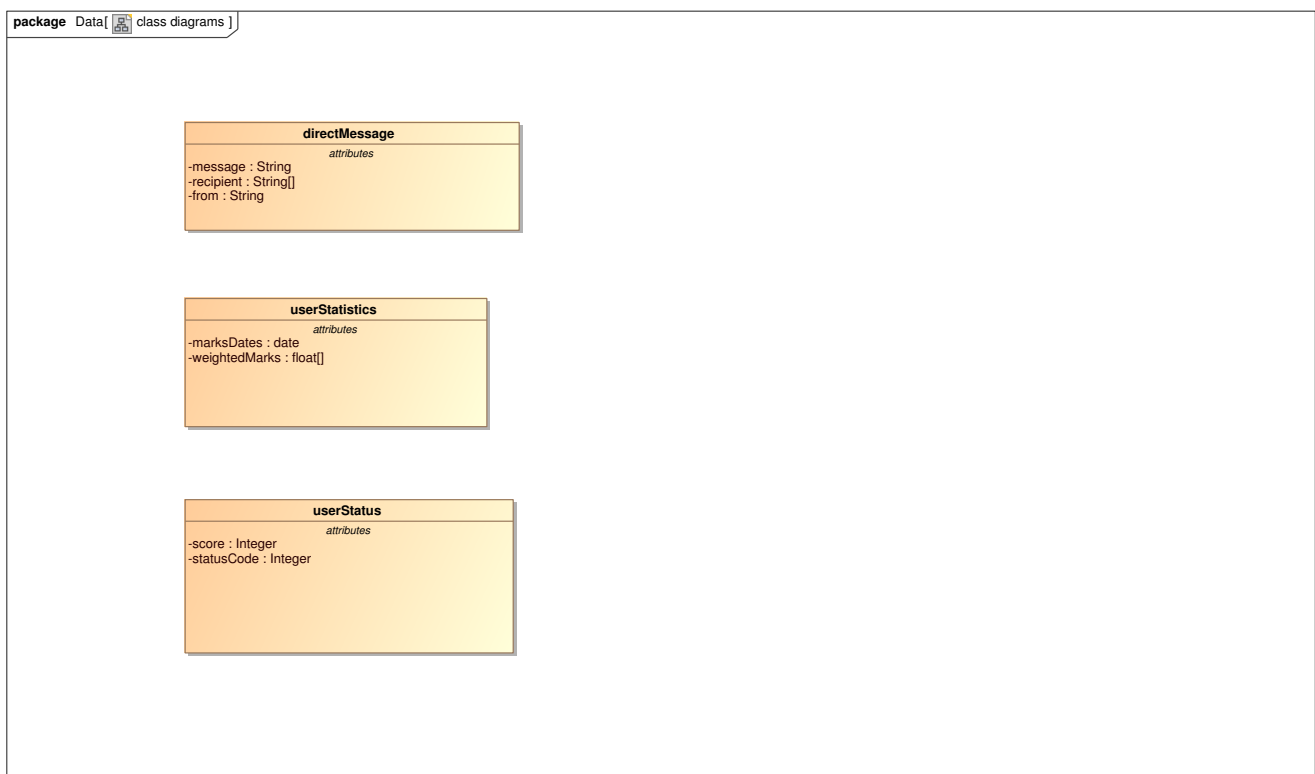
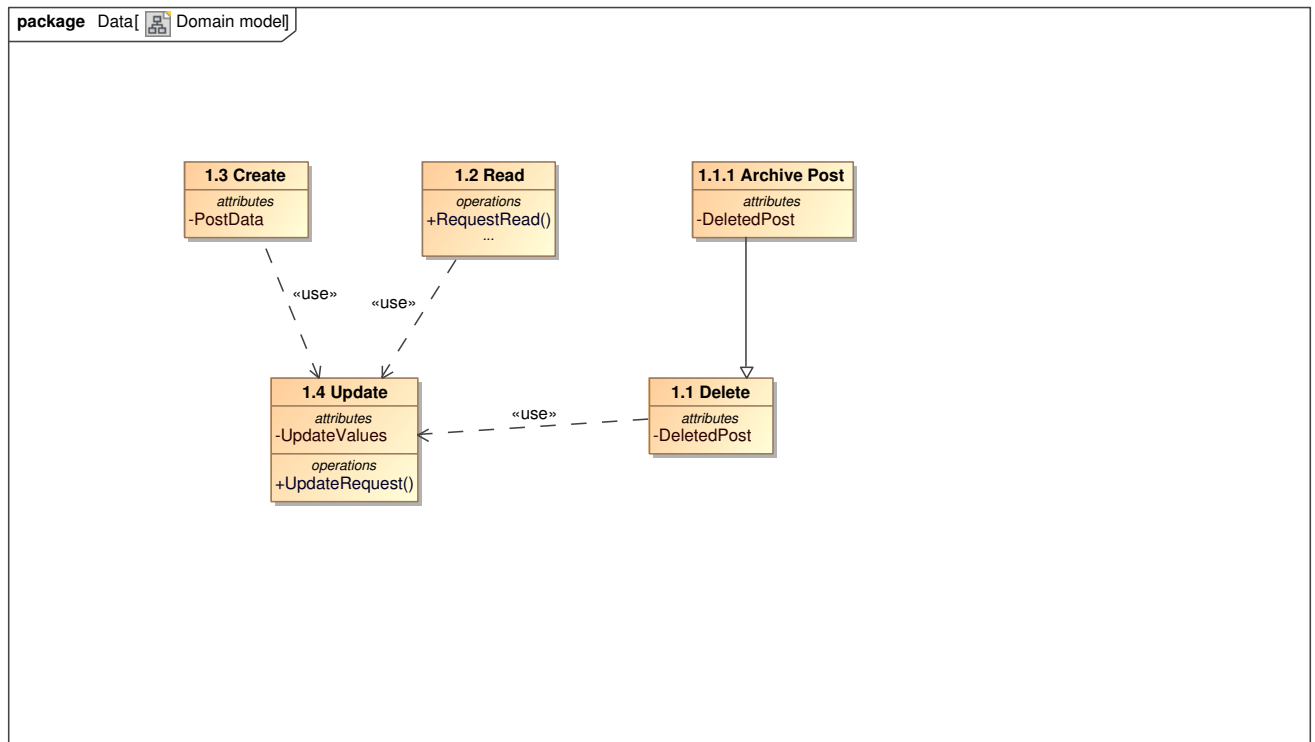
activity createPost[createPost]



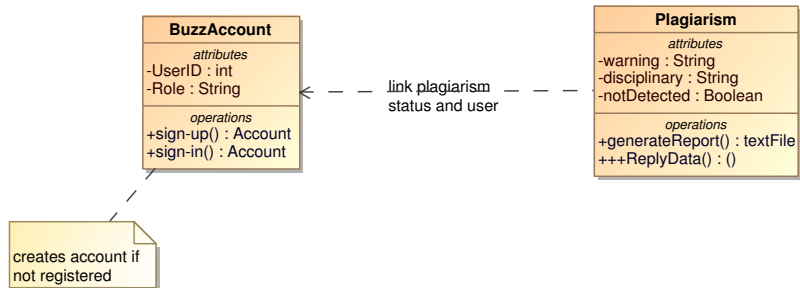




6 Domain Model



package Data[Plagiarism CD]



package Data[Netiquette CD]

