UNIVERSITEIT VAN PRETORIA
UNIVERSITY OF PRETORIA
YUNIBESITHI YA PRETORIA

Buzz discussion Forum

Requirements and Design Specifications

Compiled By

Nkosinathi Mothoa - u12077420
Nkosenhle Ncube - u13247914
Nathan Ngobale - u15110045
Kamogelo Tsipa - u13010931

Contents

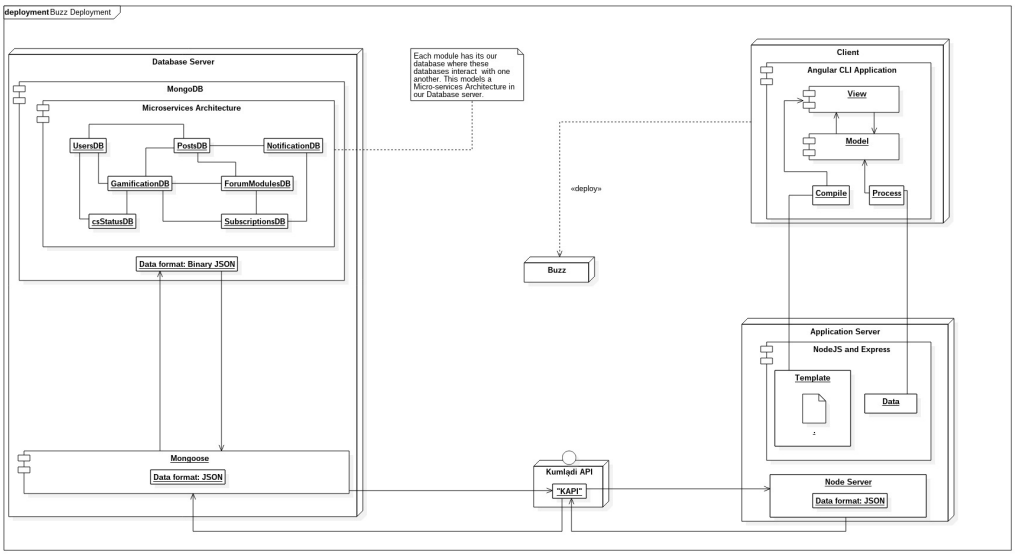
1 Vision and Scope

1.1 Project Vision

The platform is a discussion forum and is aimed at providing a space where users can collaborate on topics of interest. Users can pose questions on a said topic to the community. Through collaboration the quality of questions and answers on the platform will improve and as a result the user benefits from the community and is able to help others. An example of an instance of the platform would be the use of the platform in the field of education where students/learners use this platform to collaborate and seek help on topics that they might be interested in. To promote collaboration and continued interest, features of gamification will be added to the platform in effort to encourage users to contribute and participate in forum activities.

1.2 Project Scope

The platform is at its core a discussion forum, the selling point of the platform is the use of features of gamification to encourage user participation and contribution. Users will be able to interact with other users activity by commenting on their posts, voting posts up/down, and subscribing to posts to receive notifications on activity related to those specific posts and rewards will be accordingly given for participation.



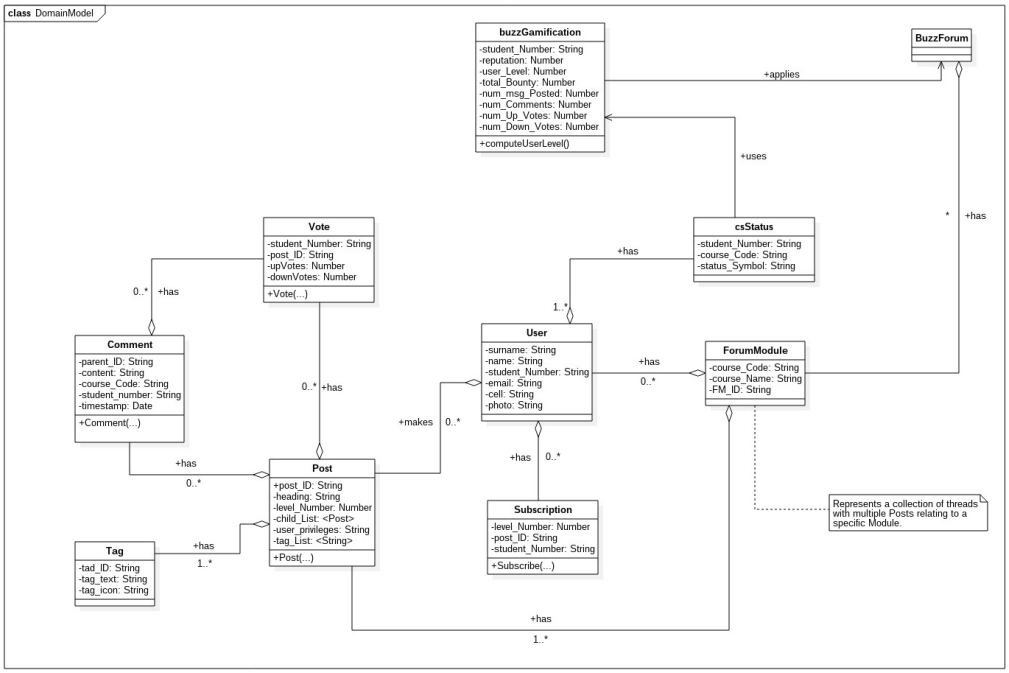


Figure 1: Deployment diagram and domain model.

1.3 Architecture Design of the System

At the highest level of granularity the system is based on Service Oriented Ar-

chitecture(SOA). Second level of granularity can be visualized as to be based on model-view controller (MVC), which further transcends to micro services in relation to SOA. We apply Model-View-Controller across our Application-Server and Client-application and the Micro-services architecture is applied within our database server (complimented by the Blackboard pattern for inter-connectedness between independent databases).

1.3.1 Architecture Design

- Service oriented architecture patterns (Micro services) on the database server.
- The Blackboard pattern, is applied across the application server and client with our Micro-Services architecture. Independent databases are enabled to communicate with each other through the blackboard pattern by bridging across each database to another.
- Publish and Subscribe pattern will be applied for users to receive notifications based on the content or topic they subscribe to.

1.3.2 Non Functional Requirements

The non-functional requirements of the system will be, but not limited to:

- Security: Users will need to authenticate themselves upon login in order for them to interact in any forum activity. User passwords are not stored in our databases, but a combination of a user name and password is encrypted using RSA encryption before being sent across the server.
- Coupling: The design of all database schema is done in effort to attain low coupling. The architecture applied in our database-server compliments the independence but unity of both our databases and system modules.
- Documentation: The development process includes the proper documentation of each module and/or DB-schema as well as a set of coding standards that must be adhered to by all contributers to the development of the system.

- Integrability: The MEAN Stack is being used to develop the system and integration follows a systematic approach across all tiers of the system.

- Maintainability: The main reason for aiming to produce a system with low coupling among its modules is to make the maintaining of this system as lucid as it can possibly be. It should not be a challenge for a person seeing the implementation for the first time to be able to grasp what a said module does and be able to add and/or alter the functionality successfully without breaking anything. This can be accomplished through high cohesion of our modules/schema and proper documentation.

Technologies

- Database: The platform preserves its data in a collection on mongoDB cloud databases and collections hosted by the mLabs Cloud Hosting Service.

- Data-Sourcing: The user information is sourced from a LDAP server. The platform connects to the LDAP server via an interface that is adaptable and reuseable. Thus one LDAP server can be interchanged for another without the need to change the interface itself.

- Development Stack: The MEAN Stack was used in the development of the platform.

- Deployment: The platform will be deployed on a cloud hosting service (ie: Heroku) initially and be moved to a physical server at a later stage after development has been completed.

2 Application requirements and design

This section introduces and discusses the modules that make the system. The purpose of each module as well as implementation details are discussed and the service contracts for specific use cases are given.

2.1 Users Module

The user module is responsible for maintaining information about registered users of the system.

2.1.1 Scope

The scope of the user module is shown in figure 2. User permissions are defined for different users of the system within the user module. Lectures and other users of higher privilege/reputation should have the ability to manage and oversee user activity within the system for students who are registered for the module that they present/teach.

Users who have logged in, may requests services to persist content from various modules based on their needs. For instance, a user only wants to have access to all the modules they are registered for (those already existing on the platform) instead of having access to platform modules that they will not be using or contributing to.

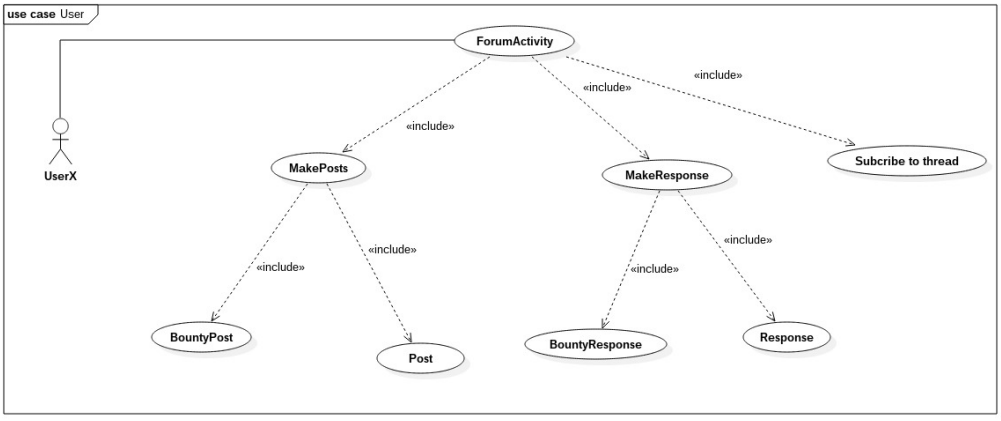


Figure 2: Scope of the Users Module.

2.1.2 Domain model

The domain model for the user module is represented in the figure below, there are two types of users: a student user and a user of elevated privilege/reputation. Users with higher reputation range from Teaching Assistants to Lecturers and course coordinators. The stock user is able to build up their reputation on the platform, the details for this are left to the discussion on the Gamification module.

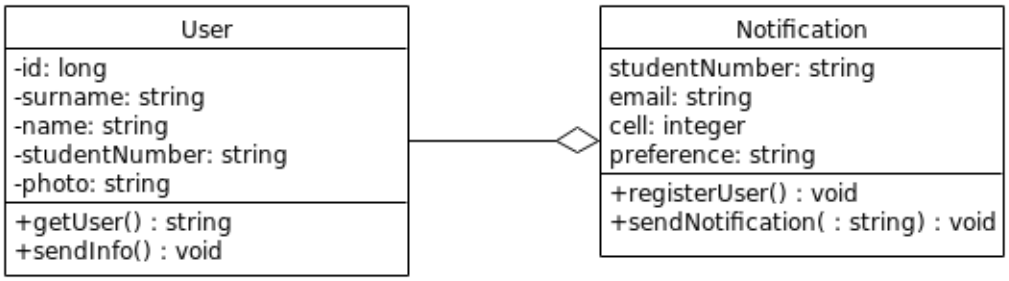


Figure 3: User domain model.

2.1.3 Use cases

Login (AuthN)

This will be a functionality that authenticates each users, through managing and controlling access to the system, by requesting the user to enter their credentials. A session will be created for the user after a successful login. Figure 4 describes the process of login and authentication of a user by the platform using LDAP.

Precondition: The user is a authenticated on LDAP.
Exception: If user is not authenticated, return an empty User Object.
Postcondition: No change
Return: a User Object with user details.

Get user information

This will be the functionality that allows lectures to retrieve the information of a specific user within the system. Only users who are enrolled in the

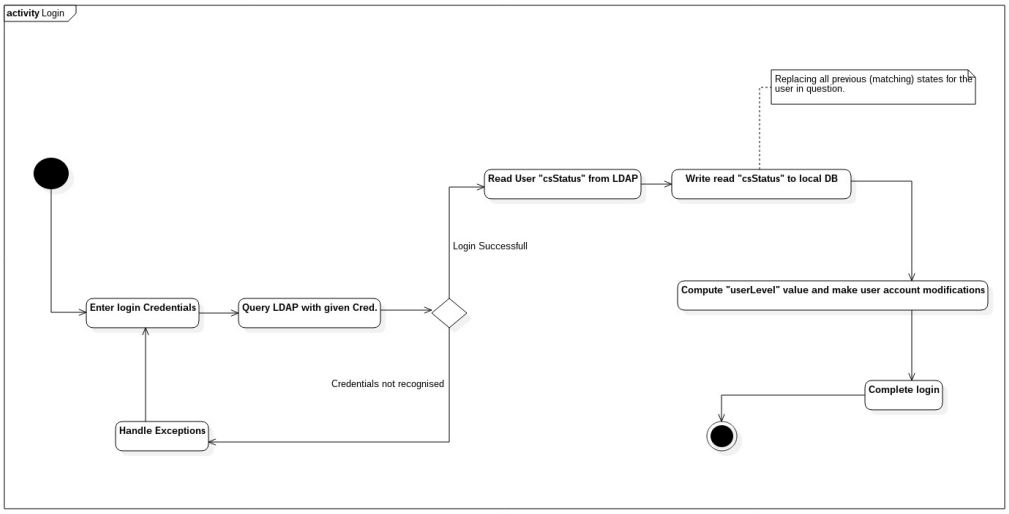


Figure 4: Login/AuthN Sequence diagram.

module which the lecture presents will be returned.

2.1.4 Security

Role based access control has been applied to restrict system access to authorized users. Users of the system have different roles within the system, which are then assigned different permissions.

The primary rules of Role-Based Access Control as applied in the system.

1. Role assignment:
    - Students have been assigned permission to access the forum modules, within the system, which they are enrolled to.
    - Admin users have been assigned permission to access the forum modules, within the system, which they present.
2. Role authorization:
    - Student authorization is done by verifying their status on LDAP, which is then validated through their status defined on the system's database.

- Admin user authorization is done by verifying the lectures credentials on LDAP, then retrieving all the modules which they present.

3. Permission authorization:

    - Once validated, students have access to forum modules which they have been verified to use. Making them unable to access modules they are not enrolled in.
    - As admin users, lectures have permission to access to the student's information, for the module which they present

A table to describe the design of the role based access control.

Subject:	Student	Higher level User
Role:	Low level user	Admin
Resources:	• Posts • Reports • Group • User profile	• Posts • User profile • Reports • Audit log
Operations:	• Create and Read posts • View reports of user contribution • Create and Delete groups • Edit user profile details	• Create, Read, Update and Delete posts • View profiles of users enrolled in their module • Generate reports of user activity within the forum module, for statistical analysis. • Generate an audit log to view and maintain users activity within the system.

User credentials, userID and password, are encrypted using RSA encryption during transmission and are decrypted during the query to LDAP. User information retrieved from LDAP is then encrypted before transmission back to the system and then decrypted before being stored on the systems database.User passwords are not stored within the systems database, users are authorized through LDAP for access to the system.

User level names

Below is a description of the kinds of users the forum should have, to add an element of gamification to the user CS-Status module a user will be assigned a user-status name which will indicate the user-level of the user (CS-Status). These user-status names should be as follows:

- Variable: All stock-users (students) are assigned the user-level-name "variable" by default.
- Function: All tutors are assigned the user-level-name "function" by default.
- Class: All Tutorial-Assistants are assigned the user-level-name "class" by default.
- Library: All Lectures are assigned the user-level-name "library" by default.

Note: a user will have a user-level-name for each of the course modules they are registered for that are supported by the forum. Meaning that a user can be a "variable" in one module and a "class" in another module. A unique colour will be associated with each user-level-name enabling the user status of a user to be quickly distinguishable.

2.2 CS-Status Module

Each user on the forum has a user status termed CS-Status. A users status (CS-Status) is determined by the role(s) he/she plays in one or more course modules. Thus from a persons CS-Status we are able to filter out what privileges the person should have on the forum and for which course modules.

2.2.1 Scope

The CS-Status of a user will be used for authorization purposes on the forum. Users of higher statuses will be able to perform additional actions on the forum that stock-users will not be able to perform.

2.2.2 Use cases

Create CS Status

Upon login, the users CS-Status will be queried from LDAP and the local copy of the state of the user will be overwritten (for safety and efficiency). This ensures that any changes that may have taken place (updates to the users state) are persisted to our local databases.

Get Status Symbol

When the user information is fetched from LDAP, the state of the user (for each of their course modules) must be transfered to an instance of the CS-Status module.

2.2.3 Domain model

The figure of the CS-Status domain model is in figure 5.

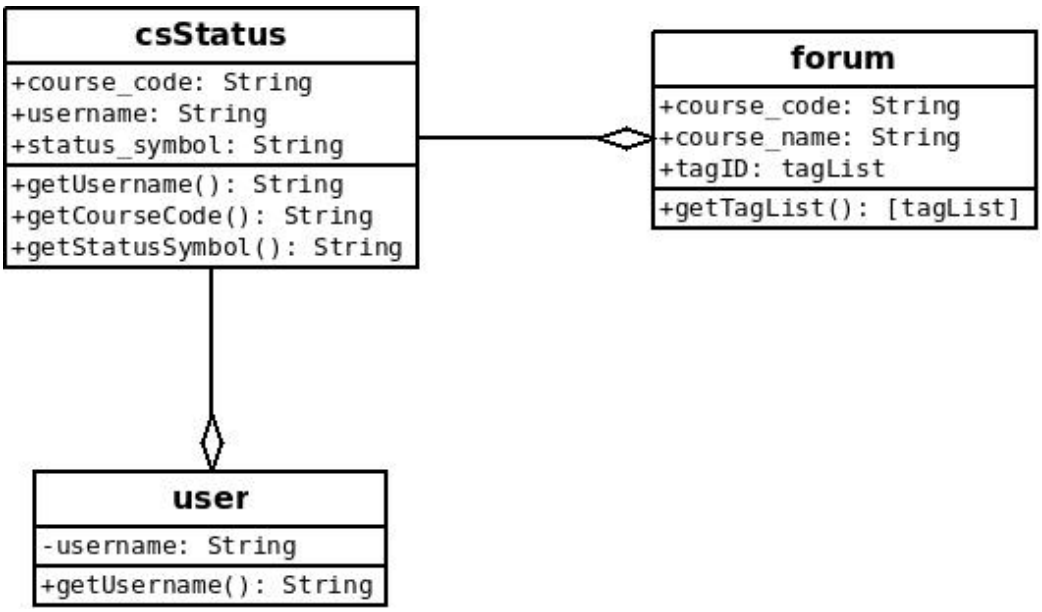


Figure 5: Domain model representing the CS status model

2.3 Post Module

The post model will be a service used to post messages to the forum. The messages will be in the form of either posts and comments.

2.3.1 Requirements

Functional

- Handle the posting and accessing of forum content as well as structuring forum threads in an orderly manner.
- Three types of posts can be made. The creation of a thread (level-zero post), the response to a post and the creation of a bounty post.

Types of posts

- Level-zero post: The first post for a specified topic is termed a level-zero post. A topic will only have one level-zero post which can be seen as a the root in a hierarchy of related posts

- Child-post: Responses to posts are termed child-posts. Child posts become parent posts to other posts when responses are made to that child post.
- Sibling-Post: Posts at the same level within the same topic are termed sibling posts.
- Bounty-post: Users are able to make posts in the form of questions with an attached reward to be awarded to the person who gives a correct answer to this question. We call such a post a bounty-post. This feature forms part of the gamification module and will be further more discussed in the Gamification module.

2.3.2 Use cases

Create Post

This will be a functionality which will allow users to create new posts for the course module they belong to. If they are adding on to an already existing post then it will fall under that post. The posts structure will follow a tree structure. Upon creating a post, tags will also be created that accompany that post.

Edit Post

This will be a service for users with higher privileges. It will allow them to edit various posts. The main edits will be on tags of a post.

Remove Post

This will be a service for users with higher privileges. It will allow them to remove various posts, meaning that they will no longer be visible to other users. When a post is removed, all the posts and comments that accompany and fall under it will also be removed.

2.3.3 Domain Model

- A new post will be on level 0, making its heading the name of the thread.

- The level of a new post, under another post, will be an increment of the post that it falls under.
- A post cannot be created by a user who does not belong to the module.

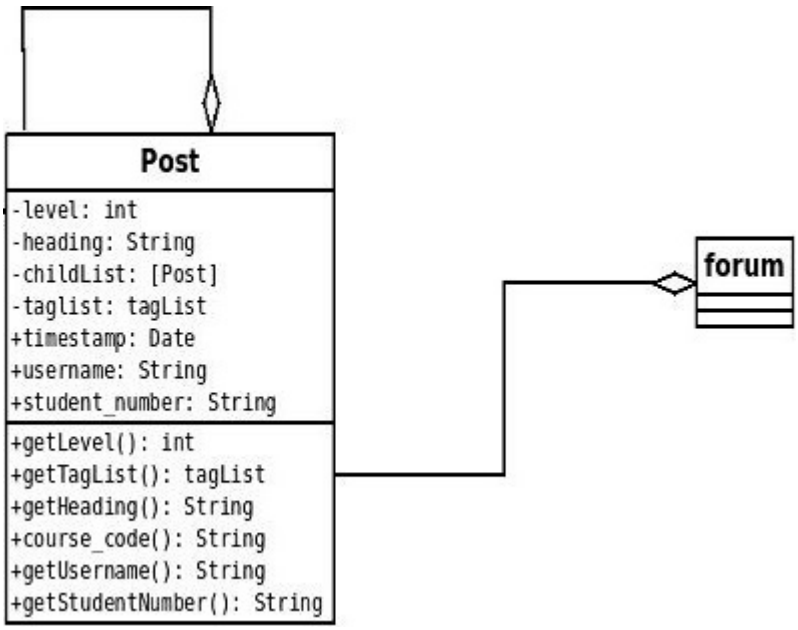


Figure 6: Domain model representing the message model

2.4 Notifications Module

This module will be responsible for the handling of User notifications.

2.4.1 Scope

Users will have the option of subscribing to Forum content and receiving notifications on this specific content via email.

2.4.2 Use cases

Add notification

The user will be able to make additions to the list of content he/she wants notifications on. This will be handled in the form of subscriptions.

Remove notification

This involves the removal of subscriptions when a user does not want to receive anymore notifications on the said content or posts.

2.4.3 Service Contracts

The notification requests will need to be of a specific type and properly validated before being processed. The request will fail if it does not meet its specified prerequisites. In order for a notification request to be passed on to the server, the user needs to already exist on the database. If this requirement is not met, a notification request will fail. Once a user has subscribed to receive notification on a said topic the following will be accomplished:

- Notify user of new posts for the subscribed thread.
- Notify user of new responses made for the subscribed thread.
- Notify user of new bounty-posts made for the subscribed thread.

2.4.4 Technologies

- The linking of the notification with the thread or post it is associated with will be done using JavaScript.This will allow for dynamic posting

of notifications, without the administrator having to manually create every notification.

- Google Email is an API with a capacity to service a number of email within a limit. The API will be ideal to provide messaging capacity for the system without having to implement email servers internally.

2.4.5 Domain Model

The domain model for the notifications module is represented in figure 6 below:

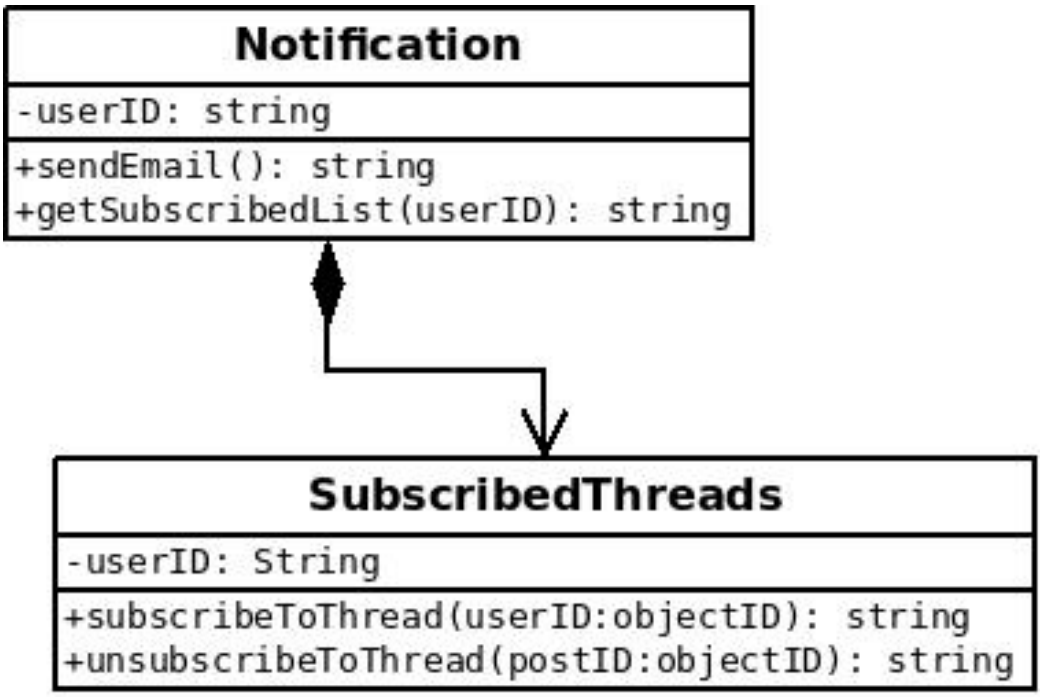


Figure 7: Domain model of Notifications

Figure 8: Domain model of Notifications

2.5 Reports Module

The Reports module is used to provide statistical information that can be used by the lecture to observe overall usage of the forum and award rewards to users that participate constructively to the forum. Management can also use the data collected to make updates and maintain the system. The statistics module can be separated into the following layers:

Personal Statistics

All users are able to view their own statistics and also stats related to the course modules they are participating in.

User Statistics

Admin users will be able to view the statistics of individual users. These stats include the number of posts, likes, dislikes and overall participation.

Course statistics

Admin users will be able to view statistics for a specified course module. These stats include the activity within a said period of time (ie: a week or month).

Forum statistics

The statistics relating to overall forum activity is meant more for the platform administrators than the course module admin.

2.5.1 Use cases

The reporting module provides services to gather statistical information for each user and export a report back to the lecture, who can assess the user's overall usage of the forum.

getTotalPosts

The system will store the assessed user profile to generate a total for the number of posts a user has made. This statistical information will be saved

and exported to the lecture/management.

ComputeAverages

An effective way to present statistics is by representing them as averages over a given sample size. Factors like averagePostsPerWeek will be represented by averages across a collection of users for the said course module.

2.5.2 Domain Model

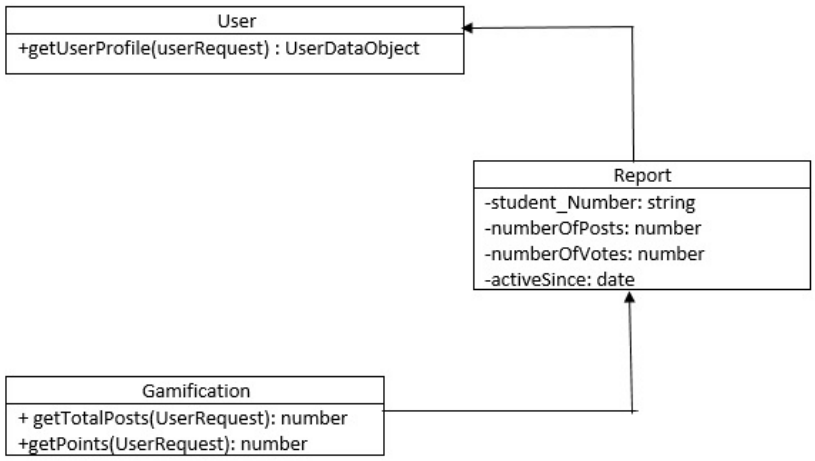


Figure 9: Domain model of the Buzz-Report module.

2.6 Authorization Module (AuthZ)

This module will be used to authorize a user, therefore allowing or denying a user from performing a specified task or operation within the forum.

2.6.1 Scope

As explained, the forum will have different types of users and each user will be limited as to what operations they can perform within the forum. Upon requesting to perform the specified operation, a user will be verified as authorized or notAuthorized by this module.

2.6.2 Use cases

authorizeUser

Check that a user is authorized to perform a specified operation. If user is successfully authorized the operation is triggered, else the operation is not performed. The criteria used to determine whether a user is allowed to perform a said operation is the user-level of the user which is dependent on the users forum status. However, with the use of gamification and the rewards system, a user is able to climb up the ranks and achieve the ability to perform non-stock operations.

2.7 Gamification Module

This module will be used to add another dimension to the discussion chat platform by including a gamifying factor which will help make the system as a whole interactive as one can measure their progress, not only does it make it exciting but also promotes participation.

2.7.1 Scope

This module will be the backbone of the gamification attribute to the system, to which the administrator has the ability to change/assign titles as well as bounty requirements.

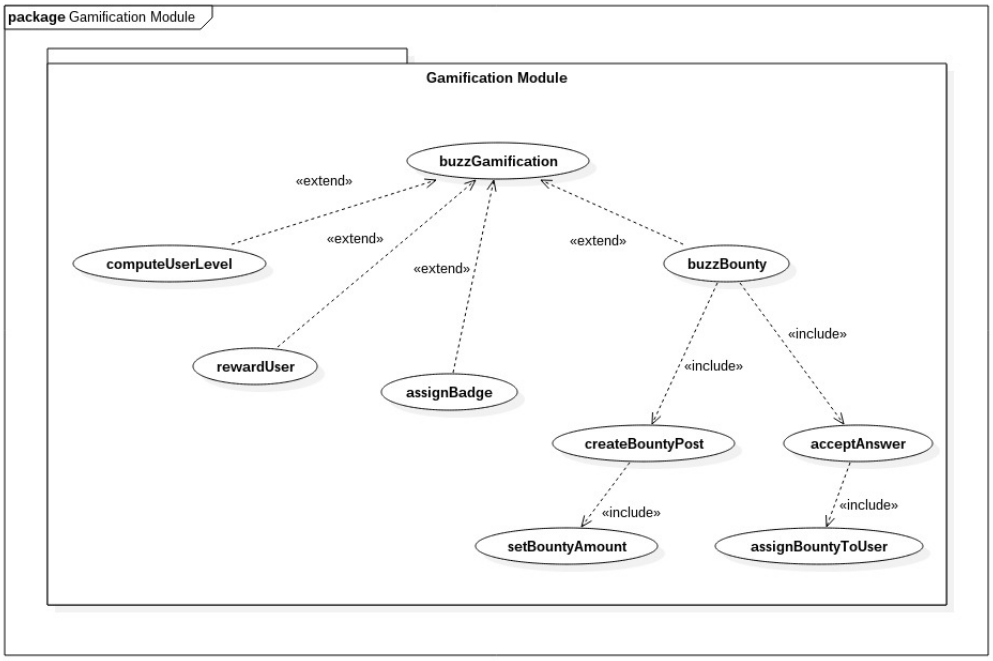


Figure 10: Scope of the Gamification Module.

2.7.2 Use Cases

Status promotion

This is in the event that a user's status is to be promoted.

Status demotion

This is in the event that a user's status is to be demoted.

Increase bounty

This is in the event that a user has done something to have their points increase.

Decrease bounty

This is in the event that something has occurred that must decrease the

bounty points of the user.

Create Group

This pertains to the event of a new group being created by a user.

Assign titles

This is an administrative feature that enables the administrator to further adjust aspects of the game.

2.7.3 Service Contracts

Calculate Status

This is used to determine what is the current status of a user, using a formula that is based on the current level their currently on

Status promotion

A person is given a new status due to them accumulating enough points to deserve a promotion, after their status is calculated.

Status demotion

A person is given a prior status, after their status is calculated, due to them loosing enough points to deserve a demotion.

Increase bounty

A person has their bounty points increased due to them performing a positive act of participation on the site, such as posting, or having another user do something in relation to their action such as like their post.

Create Group

This is the event in which someone spends their currency to create a group, to which other members are able to join in and interact on their on private exclusive thread.

Assign titles

The administrator is able to set the name and number of different titles (levels) the users can obtain as well the number of points required to achieve each.

2.7.4 Domain Model

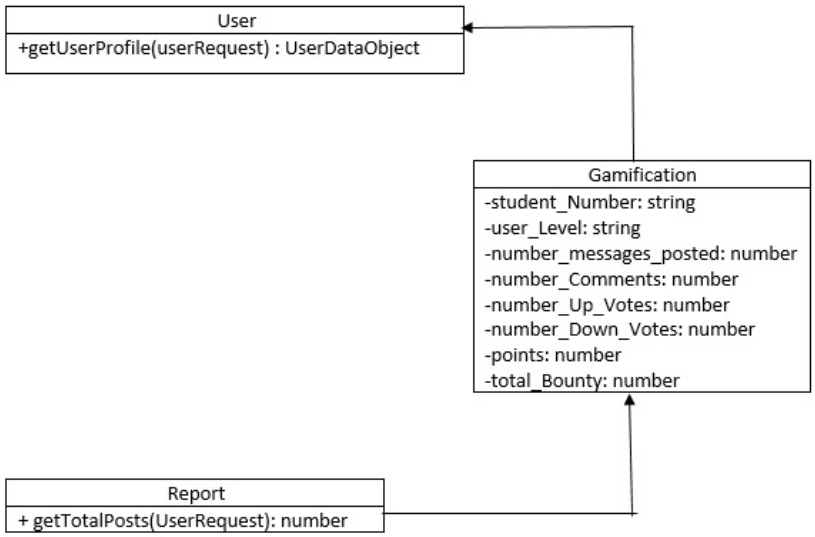


Figure 11: Domain model of the Gamification module.

2.8 Subscriptions Module

The Subscriptions module handles content subscriptions made by the user. These subscriptions include subscriptions to level-zero posts.

2.8.1 Scope

The scope of the Subscriptions Module is modeled by Figure 25. Users have the option of subscribing to particular Posts or topics. The User will then have the option to receive Notifications via email on the content of their subscriptions.

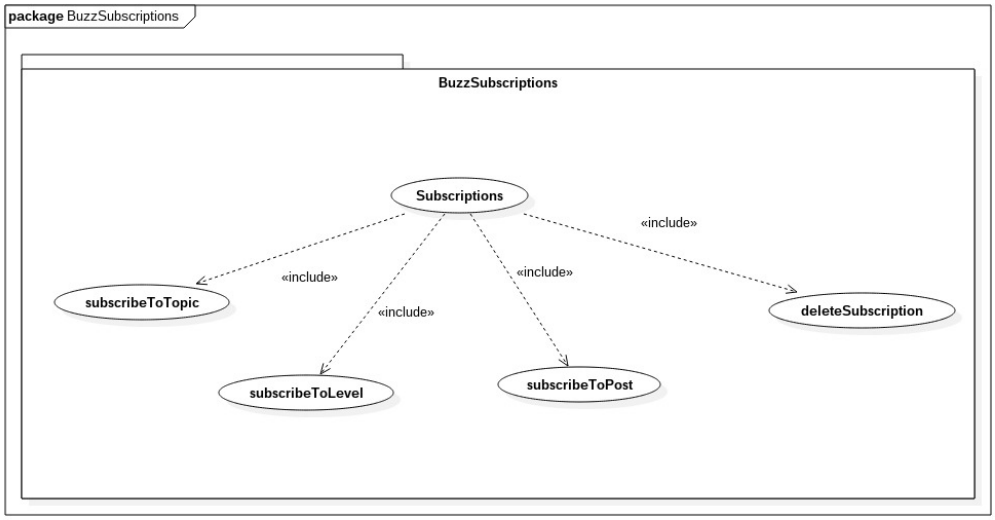


Figure 12: Scope of the Subscriptions Module.

2.8.2 Use Cases

Subscribe to topic

User will be notified on all content for a specified topic within a Forum-Module. This includes all created child posts into infinity.

Subscribe to level

User will be notified on only the activity for a given post and its immediate child posts.

Subscribe to post

User will be notified on all activity related to this post and all its child post.

Delete subscription

Removal of a subscription to a users subscription list. Resulting in the user not receiving notifications on the topic or post in the future.

2.8.3 Service Contracts

Subscriptions

A user can only subscribe to a topic or post to which they have registered for in the current academic term.

Notifications

A user has the option of specifying to which email account notifications should be sent to.

3 Glossary

- **Software Architecture** - Structured solution that meets all of the technical and operational requirements of the system.
- **Service oriented software** - Style of software design where services are provided to the other components by application components, through a communication protocol over a network.
- **Coupling** - a measure of how closely connected two routines or modules are.
- **Model-View Controller** - Software architectural pattern for implementing user interfaces on computers by dividing a given application into three interconnected parts.
- **RSA encryption** - A public key cryptography algorithm used in the encrypting and decrypting of messages.
- **DB-Schema** - Database Schema is a structure used to describe the organization of data within the database.
- **csStatus** - The role of the user for a specific module i.e. lecture, student, tutor
- **Netiquette** - The correct or acceptable way of communicating on the Internet.
- **Level-Zero post** - The first post created in a topic for a specified module.
- **Child post** - A reply to a post.
- **Sibling post** - Posts on the same level are called siblings. Thus posts with the same parent are termed siblings.
- **Parent post** - See level-zero post.
- **Forum-Module** - A module/subject that is registered on the forum and is presented by the institution in question.

- **Forum-Module** - A module/subject that is registered on the forum and is presented by the institution in question.
- **LDAP** - "(Lightweight Directory Access Protocol) is a software protocol for enabling anyone to locate organizations, individuals, and other resources such as files and devices in a network, whether on the public Internet or on a corporate intranet." - Wikipedia
- **Stock user** - a basic user with no added privileges on the forum. A stock user can only perform the basic operations on the forum.
- **AuthN** - The process of verifying that a user is "who they say they are" after this user has identified themself.
- **AuthZ** - The process of verifying that a said user has the right to access system assets/ objects or perform certain operations within the platform/system.