CU Connect

Part 4

Team Number: 41

Team Members:

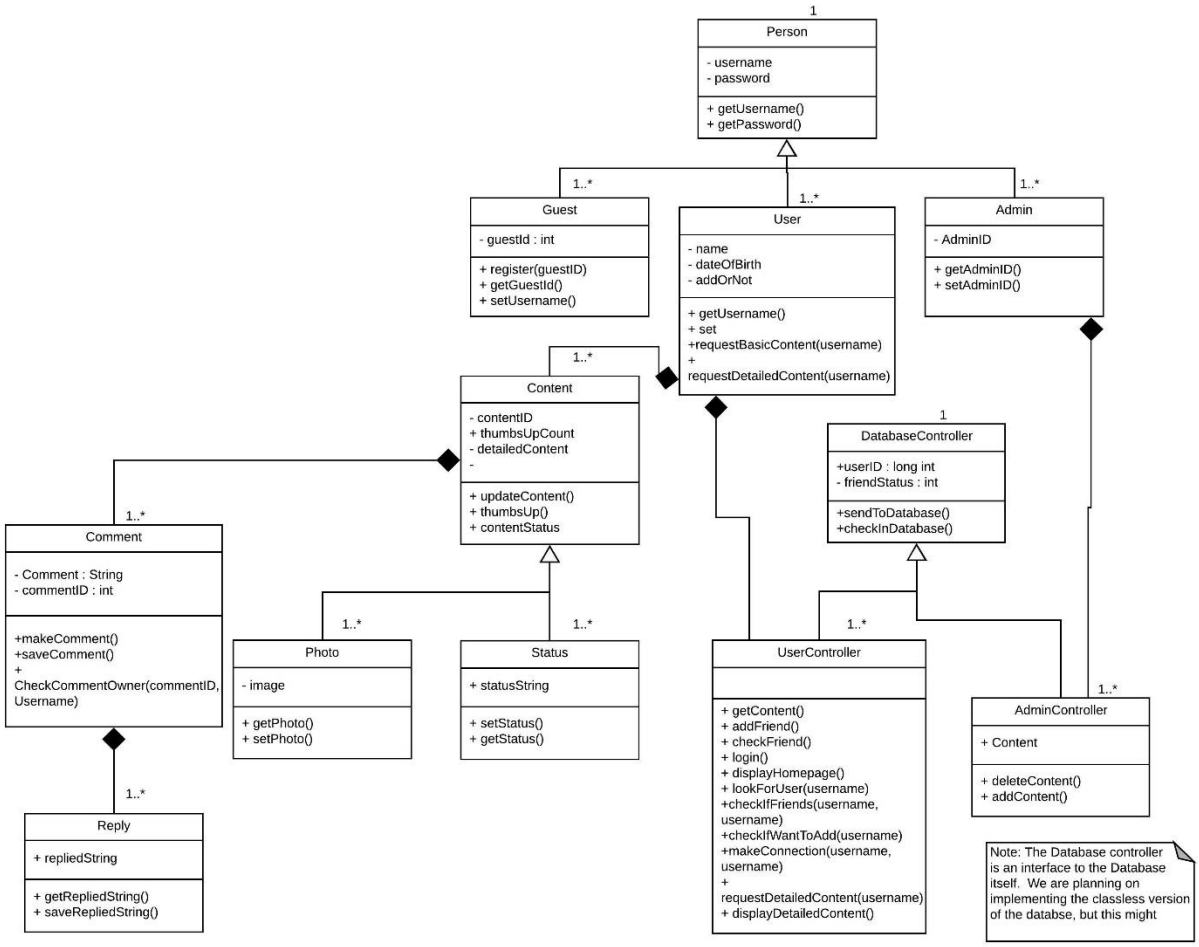
1. Richard Noronha
2. Jayakrishnan HJ

Vision and Description: CUConnect is an online portal for CU students and past pupils to connect with each other. It is a social media prototype for such a portal. In this portal, the users can connect and share content such as photos and status. The other users will have the ability to view these photos and status. They will also be able to make comments and Thumbsup a post.

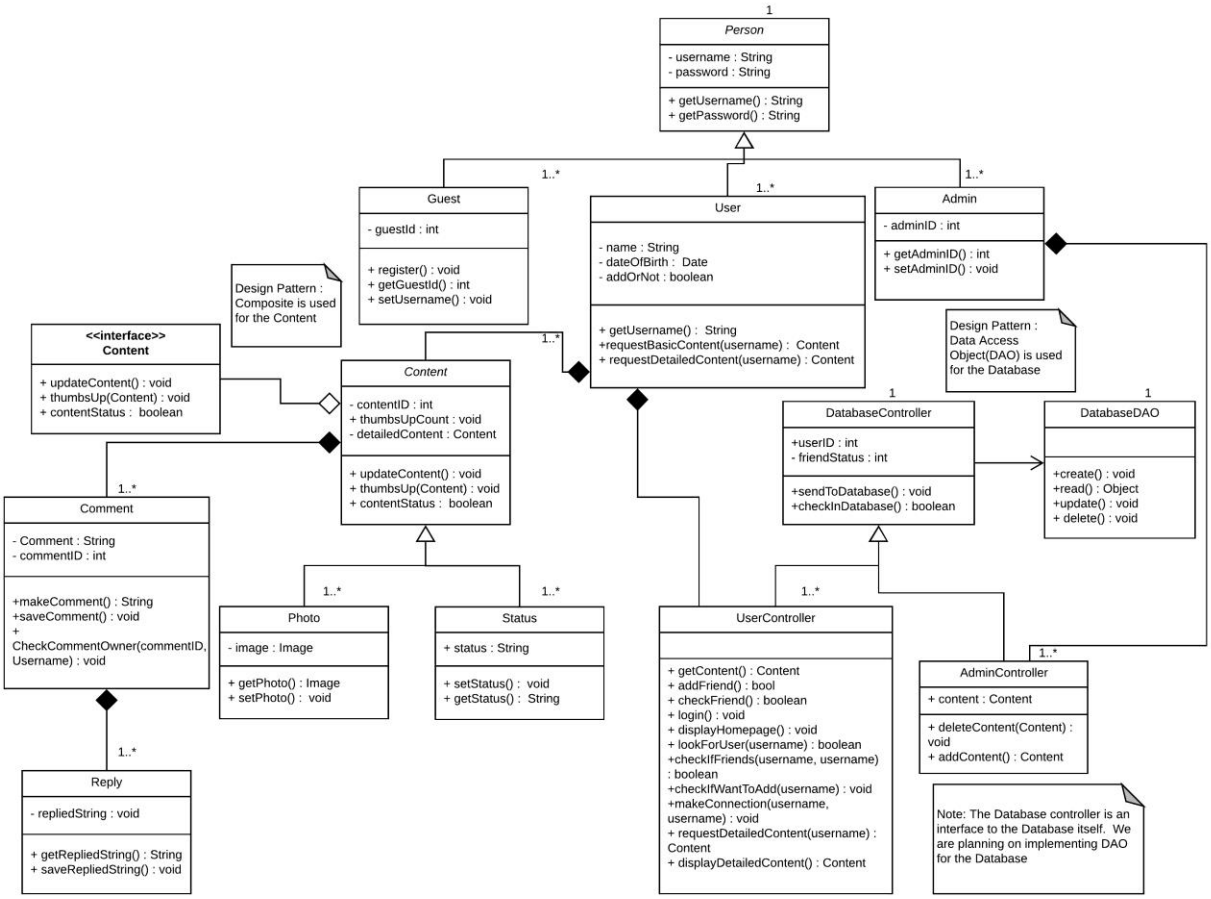
Actors: User, Admin

The backend of the system will be implemented using Java. We have tried to be as detailed as we can foresee as of now. As and when modifications are needed, the documents and diagrams will be updated. A database will be implemented too

Part 2 Class Diagram:



Implemented Class Diagram:



Summary of the work:

Summary of the work:

1. Modifying the content class to a composite design pattern to utilize its tree structure and provide uniform access to all objects.

2. We also incorporated the Data Access Object(DAO) pattern to abstract all the access to the database.

3. Did more research into an ideal IDE, Spring MVC, servers (glassfish etc.) and general web app development.

Tasks:

1. Content class refactoring: Jayakrishnan

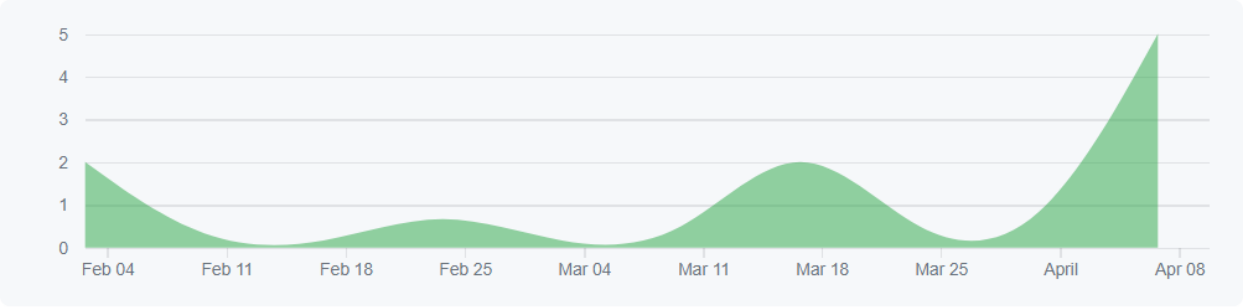
2. Database refactoring: Richard

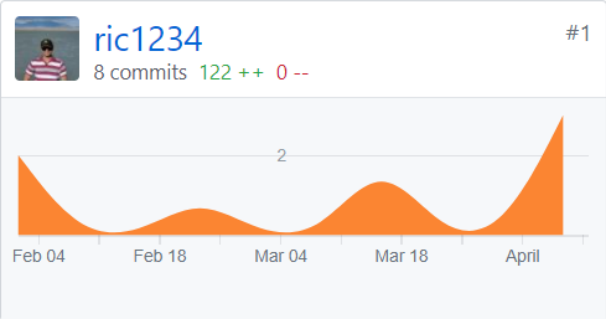
3. IDE, framework, servers, web app development: Jayakrishnan, Richard

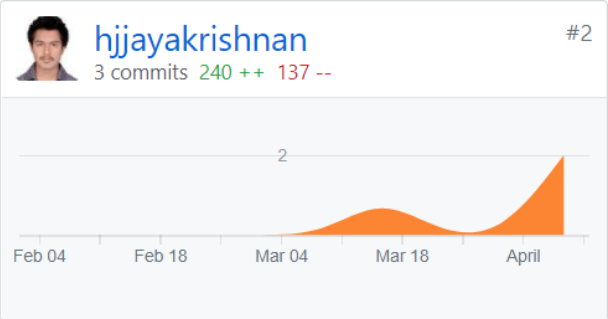
Remaining Effort

There is plenty of work left to be done. The database integration, implementing a framework, and work on the frontend if necessary are still pending.

Github Graph:







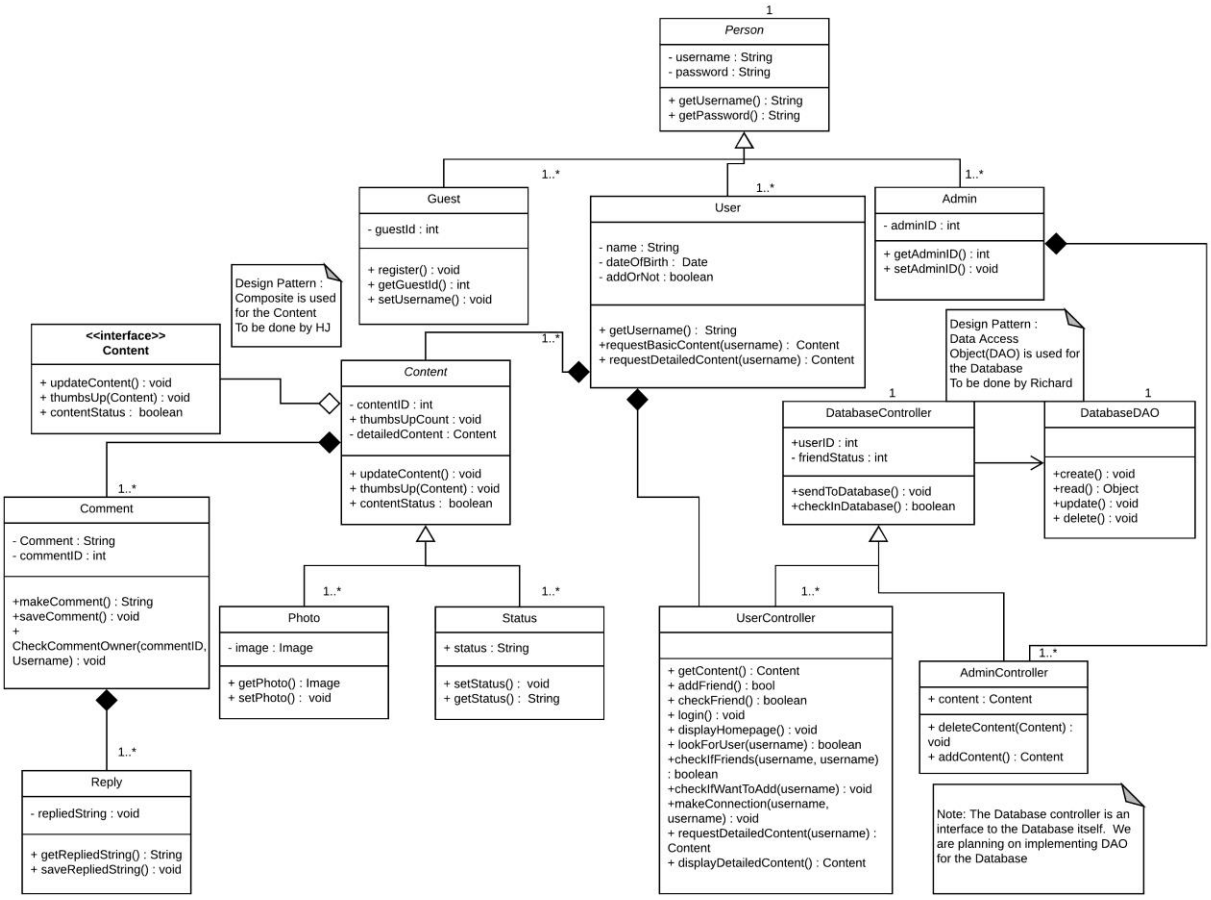
Design Pattern Description:

1. Now that we got a better grip over design patterns, we explored different patterns and researched how it can be effectively applied for our application. For instance, the content class can be created using the composite design pattern to utilize its tree structure and to provide a uniform access across all objects.

2. We also researched patterns for data access and decided to go with the 'Data Access Object' pattern. A Data Access Object (DAO) is used to abstract and encapsulate all access to the data source. The DAO manages the connection with the data source to obtain and store data. For the data source, we are planning to use a persistent source like an RDBMS.

3. We will be using Hibernate and Spring MVC for the Database management

Design Patterns in class diagram:



Final Iteration:

In the coming weeks we will implement the classes. We will be using Hibernate for the Database related operations and the Spring MVC framework as well. This implementation is still pending.

We will implement the most important sections of the project on a priority basis.