



# Java Code Peer Review Platform

An Application for CSC 380: Software Engineering

By: Jhulendra Bhattarai, Ko Ko Aung, Liam McMahan, Kieran Finnegan, and Kevin Morales

Professors: Bastian Tenbergen and Tamaike Brown

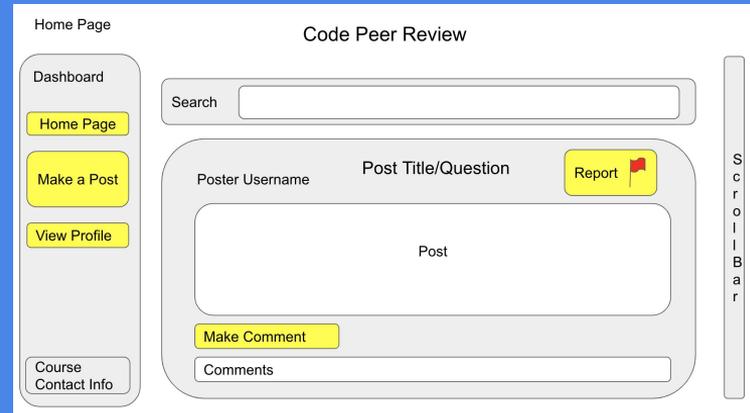


# Project Description

- The goal of the project is to design and implement a web-based application for students to learn and improve code quality based on peer feedback.
- This platform is designed to solve low level problems that help beginner programmers to read, understand, and interpret their code more effectively (i.e., CSC212 students)
- Our system will provide users with a text area, into which they can enter Java code. Additionally, code files are stored in the system and can be shared with other users. These users can then open others' files and provide feedback in terms of code comments.

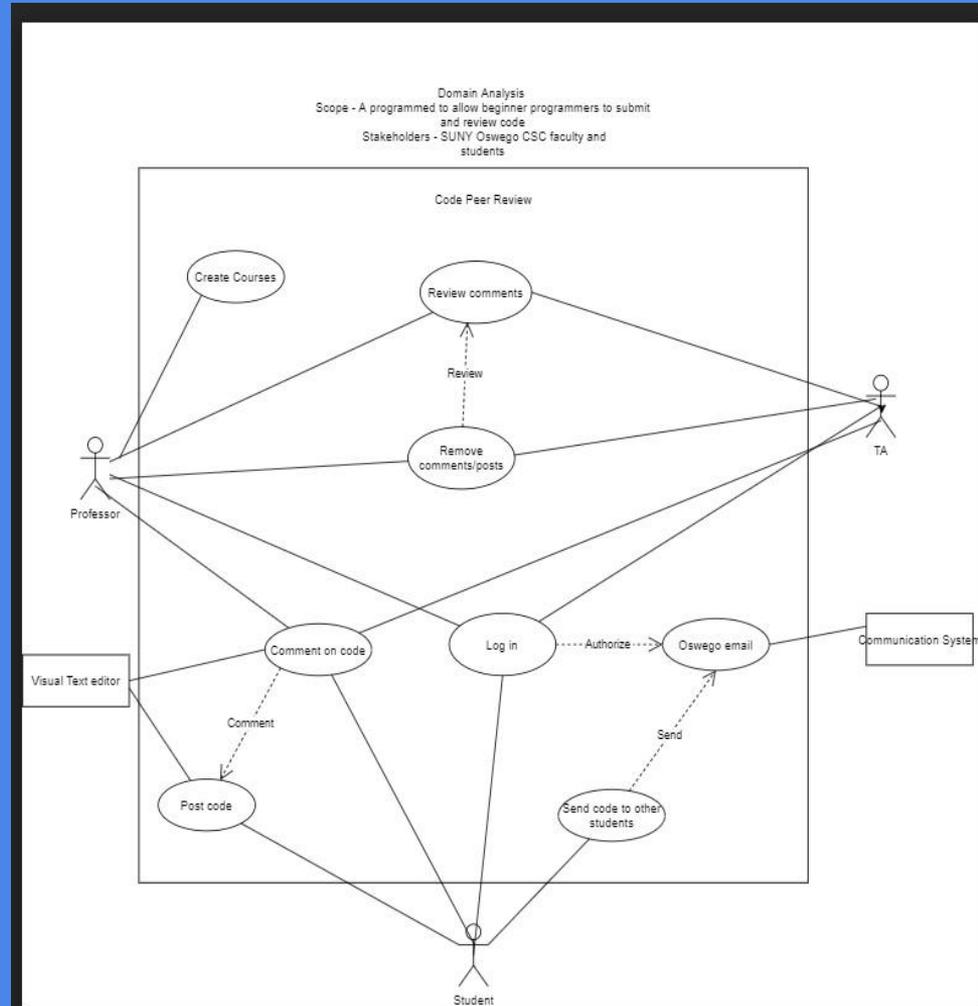
# Requirements

- What profs wanted to do
  - To have clean and understandable design
  - able to post java program files with questions and thoughts.
  - A visual text editor, with java syntax highlighting.
  - Allow students to observe posts, and provide interactive feedback in the form of Java comments,
  - Role classifications for particular users
  
- What we wanted
  - Offer features and components like many popular online forums, such as Stack Overflow
  - Separate roles into Instructor and student
  - Clean and convenient

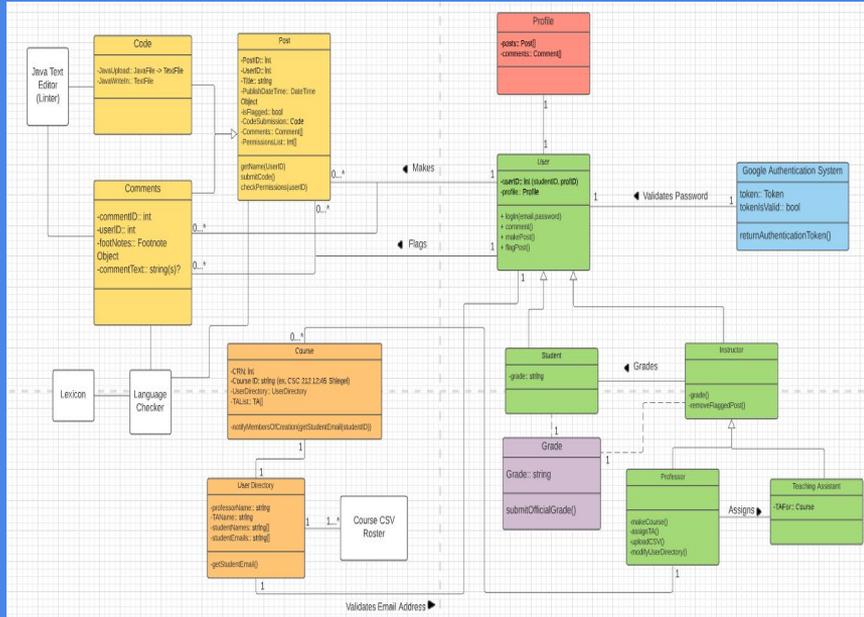


# Use Case Diagram

- To better understand how users will use the program, we created a use case diagram.
- The diagram shows the relationships between the parts of the program when a actor is performing a task
- The diagram is made from the information from a domain analysis document
- The use case diagram is used to help develop the function requirements of a program



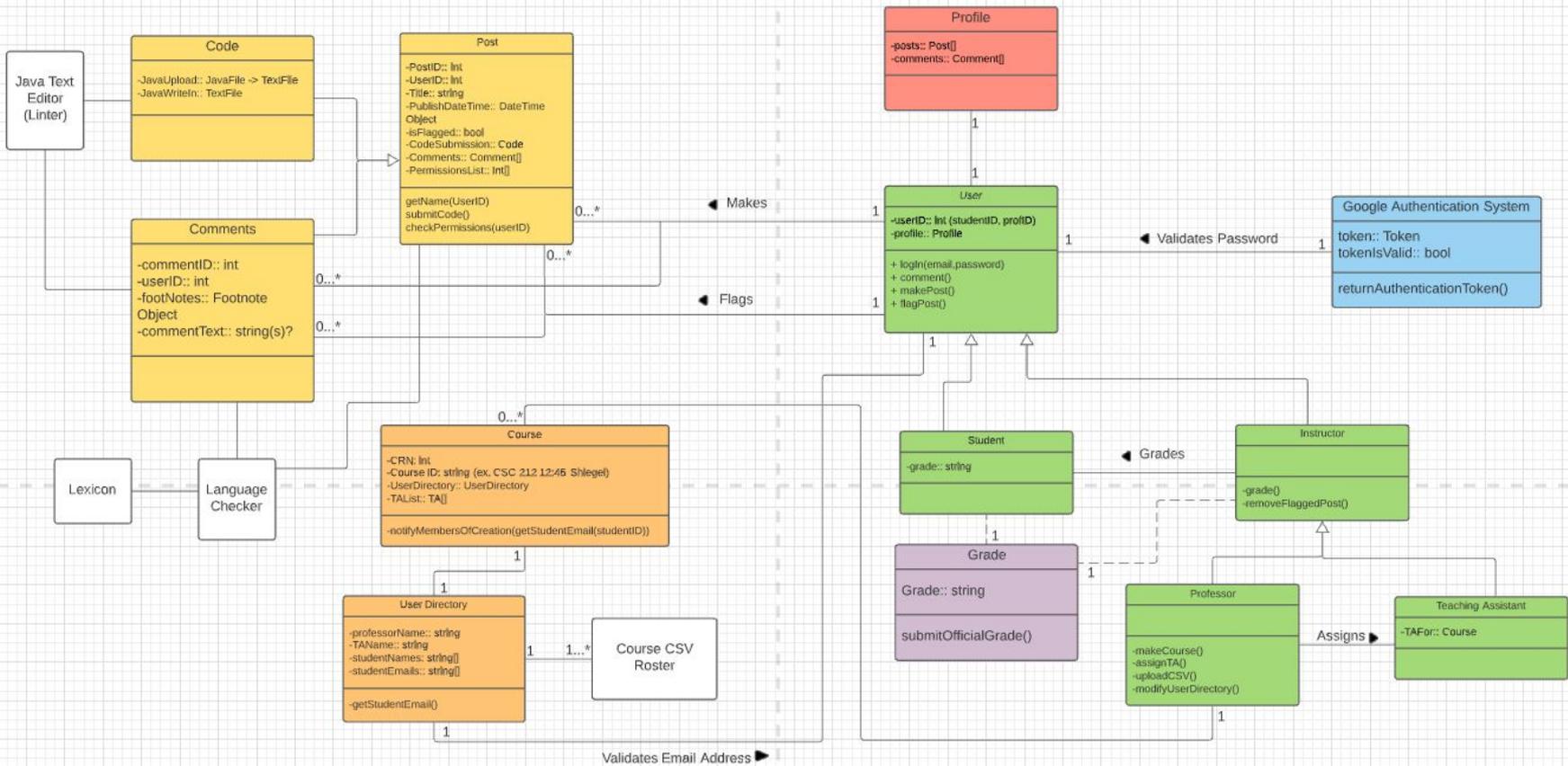
# System Architecture



In order to outline the varying components of our platform, our next objective would be to make what is known as a UML Class Diagram.

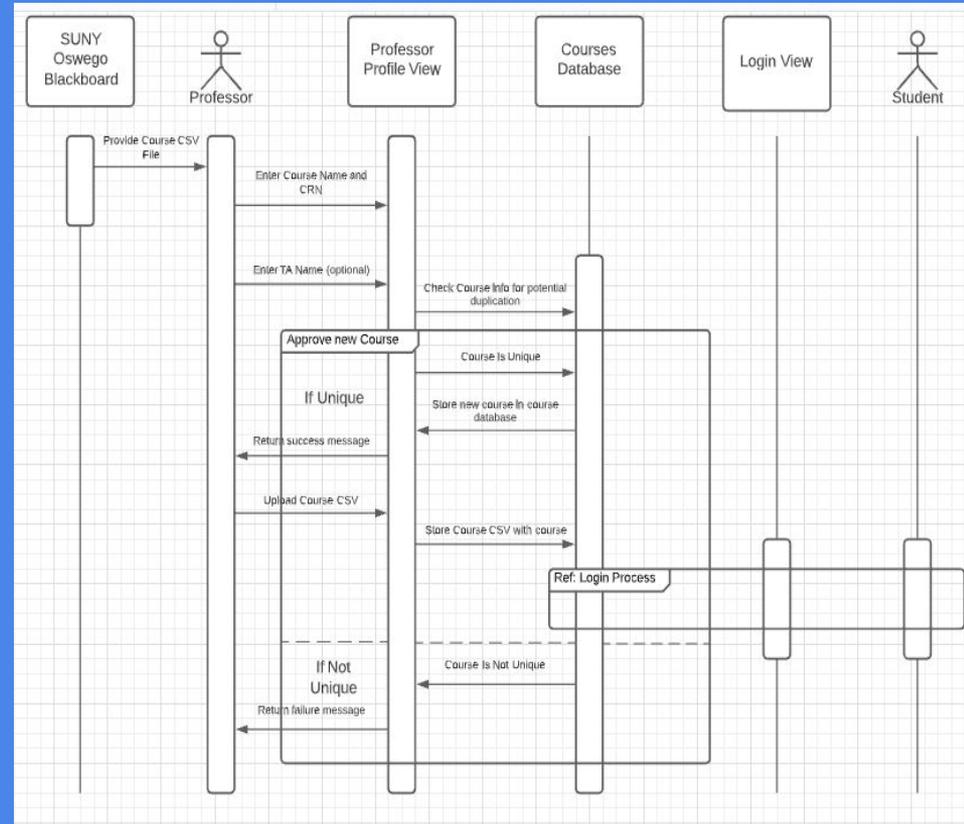
Our UML Class Diagram consists of elements such as:

- Classes: Components of the program that will be implemented to carry out the goals of the user
- Associations: Connections between classes that are accompanied by particular labels and arrow heads to specify their relation
- Multiplicities: To indicate how many instances of a particular class are involved in an association.



# Detailed Design

- We are using Model/View/Controller Pattern because it is commonly used to build websites or related system like college registration systems, canvas, etc.
- Model View Controller Pattern system needs to access to retrieve data as well as display it to the user. That's why our platform fits best with Model View pattern.
- Our Course Creation Diagram:





# Implementation

We will implement our program using various programming languages and frameworks. Our project will be split between a user interface (frontend) and the application logic (backend). The frontend will be the point of interaction for our users, while the backend will handle all the requests that those users will make.

Below is a list of our tools for implementing our peer review platform.

Backend:

\*C# programming language

\*ASP.NET framework

\*Java programming language

\*Spring Boot framework

MongoDB database

Frontend:

HTML/CSS

React Javascript framework



# Closing remarks

Thank you for your time!