

### **COSC 310 Milestone 3- The Learning Layers**

Group members:

- Oluwadabira Omotoso (84518448)
- Afua Frempong (90434176)
- Karim Khalil (38485272)
- Peter Idoko (89385496)
- Atharva Jagtap (39283783)

For this milestone, use the appropriate models to present the details about what your system will look like, what it will do/be composed of, and what your architecture is.

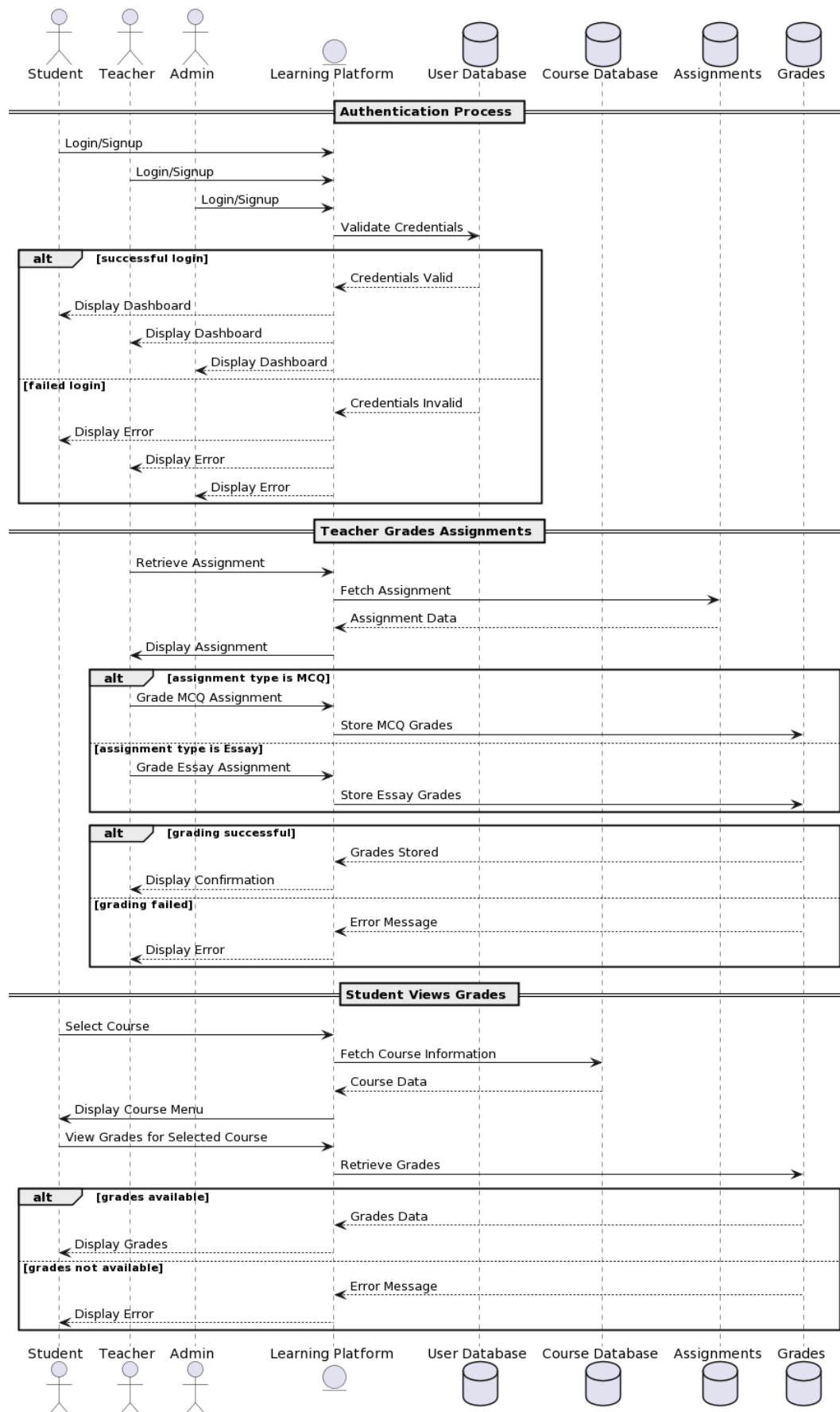
Submit a report containing the following information:

- You will need to model and analyse your use cases and requirements properly using the appropriate models with the appropriate level of granularity. For the projects, at a minimum, you will need to ensure that you have the appropriate **sequence diagrams** (supporting your use cases), and you will also need to include a **class diagram** for the different parts of the system.
- The reader should be able to understand the different entities in the system (class diagram), how they interact with respect to time (sequence), and the data that is moving throughout the system.
- You will need to develop a **test plan** that will detail the requirements for testing.
- Describe at least two **design patterns**. that your team is going to implement for your project. Clearly explain why your team has chosen those design patterns and how they will be implemented.

**Rubric:**

- Sequence diagrams (5 marks)
- Class diagram (5 marks)
- Testing plan (5 marks)
- Design patterns (5 marks)

## Sequence diagram:



## **Sequence Diagram Description:**

### **1. Authentication Process:**

This process begins when a user initiates a login by entering their credentials.

Login/Signup: The user provides their credentials to the system.

Validate Credentials: The system checks the entered credentials against the user database.

Display Dashboard or Display Error: If the credentials are valid, the user is taken to their dashboard. If not, an error message is shown.

This sequence ensures that only authenticated users can access the system and their respective dashboards.

### **2. Teacher Grades Assignments:**

Here, a teacher interacts with the system to grade assignments.

Retrieve Assignment: The teacher requests an assignment from the system.

Fetch Assignment: The system retrieves the relevant assignment data.

Display Assignment: The assignment is displayed to the teacher for grading.

Grade MCQ/Essay Assignment: Depending on the type of assignment (Multiple Choice Questions or Essay), the teacher grades it appropriately.

Store MCQ/Essay Grades: The grades are then stored in the system.

Display Confirmation or Error: After storing the grades, the system either confirms the successful operation or displays an error if something went wrong.

This sequence demonstrates the process of grading different types of assignments and ensures that grades are correctly stored in the system.

### **3. Student Views Grades:**

This part of the diagram shows how a student can view their grades.

Select Course: The student starts by selecting a course.

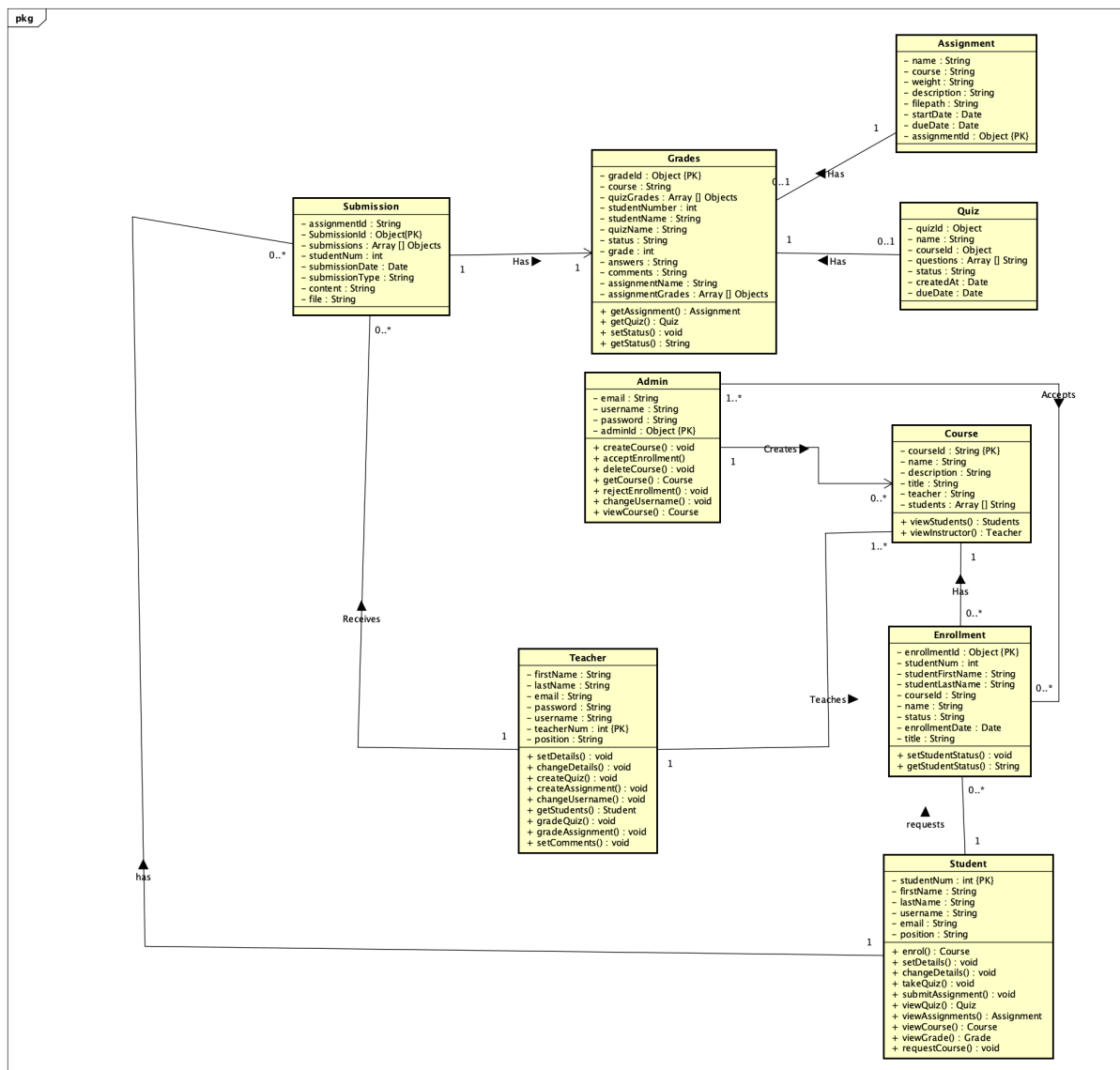
Display Course Menu: The system displays the course menu to the student.

View Grades for Selected Course: The student opts to view grades for the course they have selected.

Retrieve Grades: The system retrieves the grades from the database.

Display Grades or Error: If grades are available, they are displayed to the student. If there are no grades to display or there is an issue, an error message is shown.

## Class diagram:



## Testing Plan:

- **Unit Testing:** Using the jest framework we can perform unit tests on our code to ensure that they are functioning as intended. We will be unit-testing all branches before integrating them into the main branch to ensure that changes made to the branch do not contain errors that may affect the main branch. For testing to be successful, all unit tests would need to be passed.
- **Integration testing:** Similarly, the Jest framework provides us with automated coverage tests which we can use for integration testing. This is to be done during the merge between the main and created branches. This ensures that all added and changed codes work fluidly and as intended. Coverage tests would need to achieve a score of at least 90% to be accepted.
  - We will utilise Github actions with Jest to test our code whenever pull requests are made to ensure they are functioning as expected
- **Acceptance testing:** We will utilise manual testing methods to ensure the program is as efficient and effective as possible and also to ensure that the pages are aesthetically to the standards we have set. Additionally, the Jest framework provides us with mock environments that would test all aspects of our code in a simulated environment to ensure they are functioning effectively.

### **Design Pattern:**

1. **Singleton pattern:** This is because for authentication systems like the login and protection of user information, we can manage the system in a centralised manner.

#### **Rationale:**

- Authentication:
  - Centralising the authentication system is crucial for ensuring a secure and consistent login experience for users across the entire Canvas-based e-learning platform.
  - A singleton instance can manage user sessions, handle authentication requests, and maintain user-related information in a centralised manner, promoting uniformity and security.
- **Implementation:**
  - Create a AuthenticationManager singleton class responsible for managing user authentication. Test to make sure the class cannot be instantiated more than once.
  - Class holds the authenticated user's information, manages sessions, and provides methods for login, logout, and user-related operations.
  - Other components in the system interact with the AuthenticationManager to check user authentication status or perform authentication-related tasks.
  -
- **Benefits:**
  - Ensures there is only one instance of the authentication manager, preventing multiple instances that could lead to inconsistencies.

- Simplifies access to authentication functionalities throughout the system.
- 
- 2. **Observer pattern:** we can use this to notify students of assignments, discussion posts, or changes made to course material.

**Rationale:**

- Real-time Notifications:
  - The e-learning platform should provide students with real-time notifications for assignments, discussion posts, and changes to course materials.
  - The Observer pattern facilitates a publish-subscribe mechanism, where students (observers) can receive notifications about specific events.
- **Implementation:**
  - Create an EventDispatcher class responsible for managing events and notifying registered observers.
  - Components such as the AssignmentManager, **DiscussionForum**, and CourseContentManager act as subjects that generate events when changes occur.
  - *Students subscribe to specific events of interest (e.g., new assignments, discussion posts) and receive notifications when these events occur.*
- **Benefits:**
  - We are able to maintain consistency in our work
  - Loosely coupling objects, makes the system more flexible and maintainable