

Learning Management System

Learning Management System

1. Project Metadata

GitHub Repository: <https://github.com/MohQaDev/Software-Engineering-Project/>

Version	Date	Commit Hash	Description
v1.0.0	Dec 18, 2025	7895f3eb	Initial architecture and core UML diagrams.
v2.0.0	Jan 04, 2026	ea819b7d	Refined behavioral models and final report structure.

2. System Description

2.1 Problem Statement

Modern educational institutions require a reliable digital platform to manage teaching and learning efficiently. Without a centralized system, handling course materials, assignments, grading, and communication becomes fragmented and error-prone.

2.2 Stakeholders

- Students
- Instructors
- System Administrators
- University Management
- IT Support Staff

2.3 Main Requirements

- User authentication and role-based access
- Course creation and enrollment
- Assignment submission and grading
- Feedback and performance tracking
- Notification of deadlines and updates

2.4 Assumptions

- Users have internet access
- The system is web-based
- The system is primarily data-driven

3. Context Model

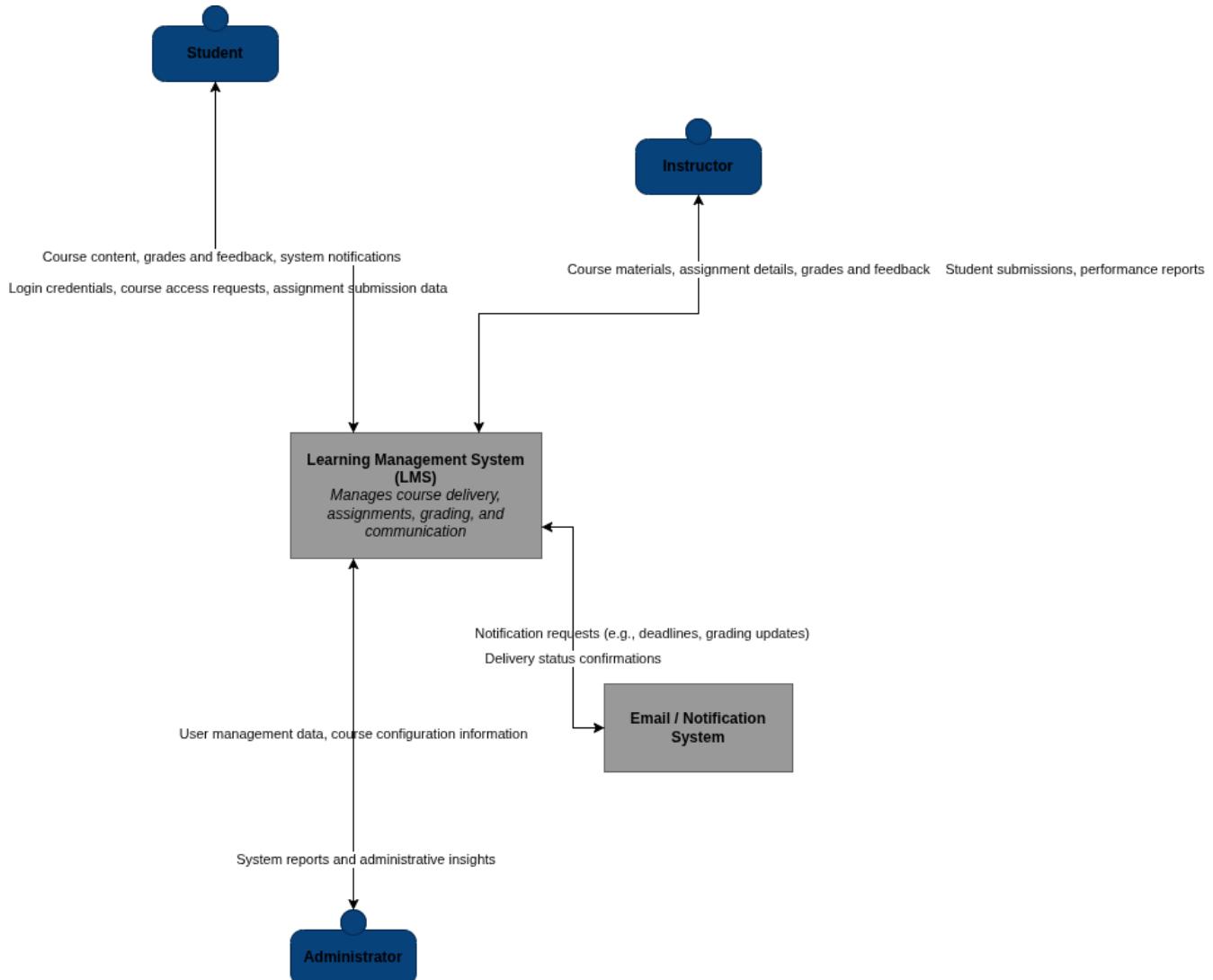


Figure 1: Context Diagram

Explanation: The context diagram shows the LMS as a central system interacting with external actors, including students, instructors, and notification/authentication services. It defines system boundaries and clarifies how the LMS fits in the academic environment.

4. Activity Diagrams

4.1 Assignment Submission

Explanation: This diagram represents the workflow a student follows to submit an assignment, including accessing the course, uploading the submission, and receiving confirmation. It highlights decision points such as submission validation and ensures the process ends with successful recording.

4.2 Course Enrollment

Explanation: This diagram models the course enrollment process. Students browse courses, check prerequisites, select the course, and confirm enrollment. The activity diagram ensures system validations are performed and

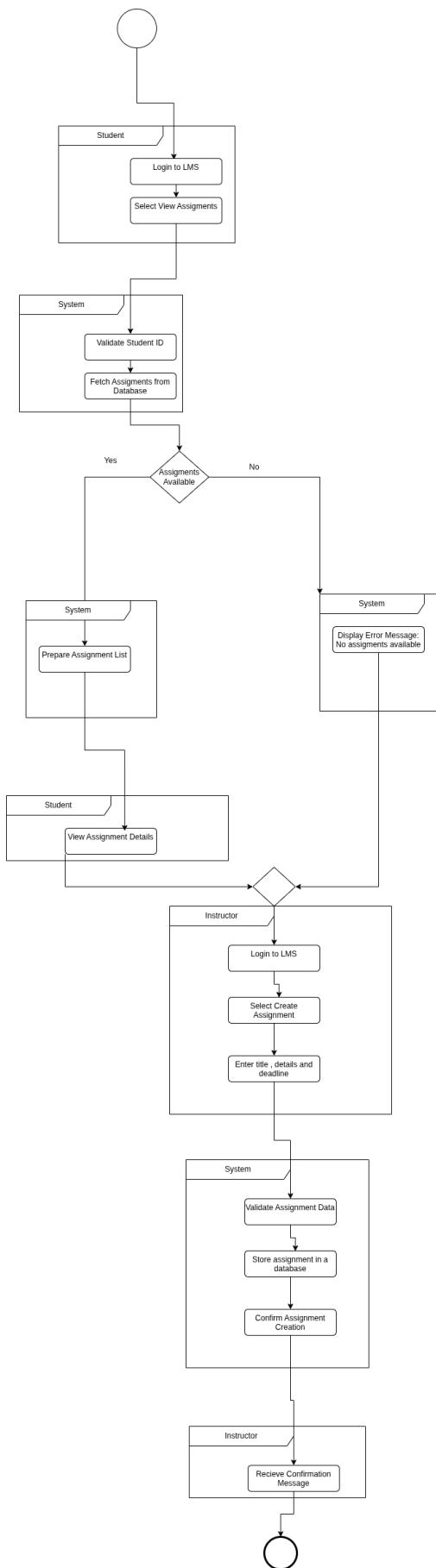


Figure 2: Activity Diagram 1
3

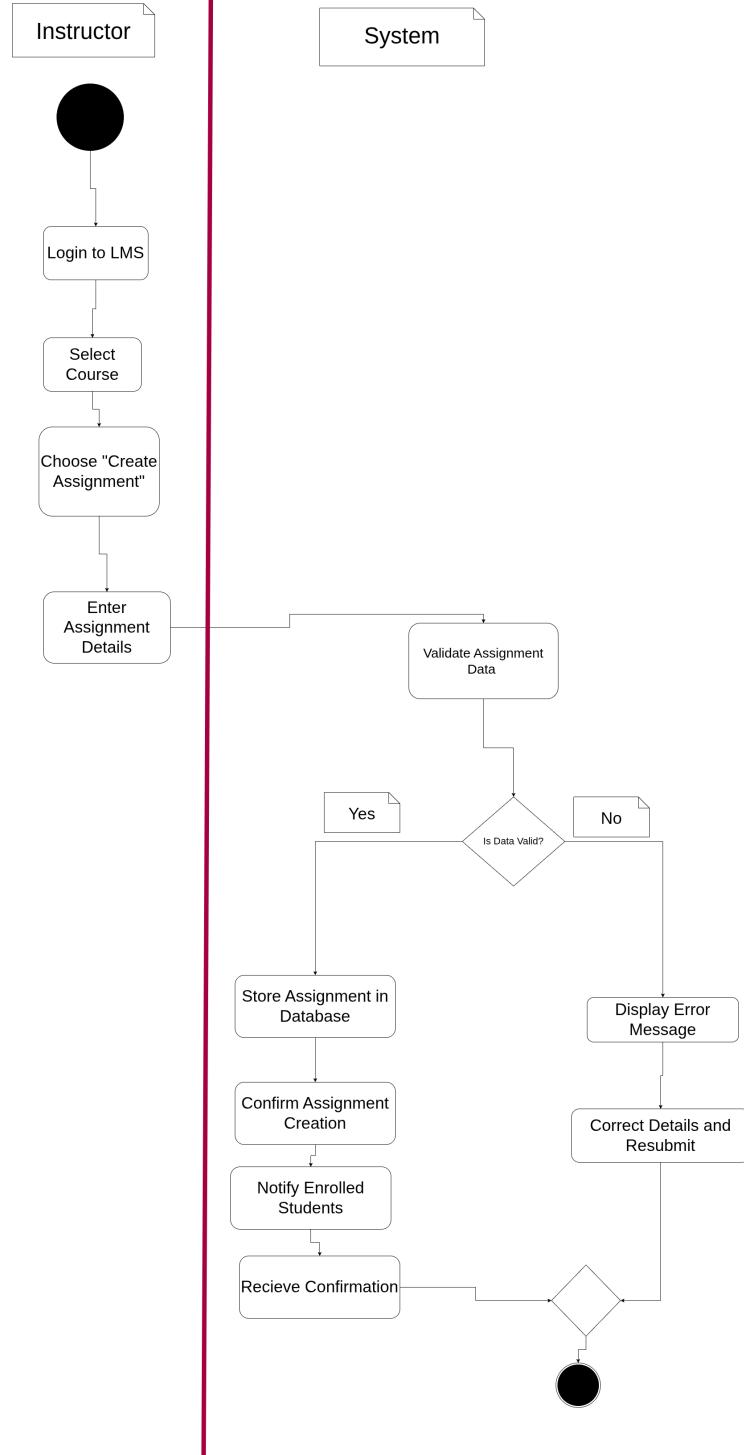


Figure 3: Activity Diagram 2
4

enrollment completes successfully.

5. Use Case Diagrams

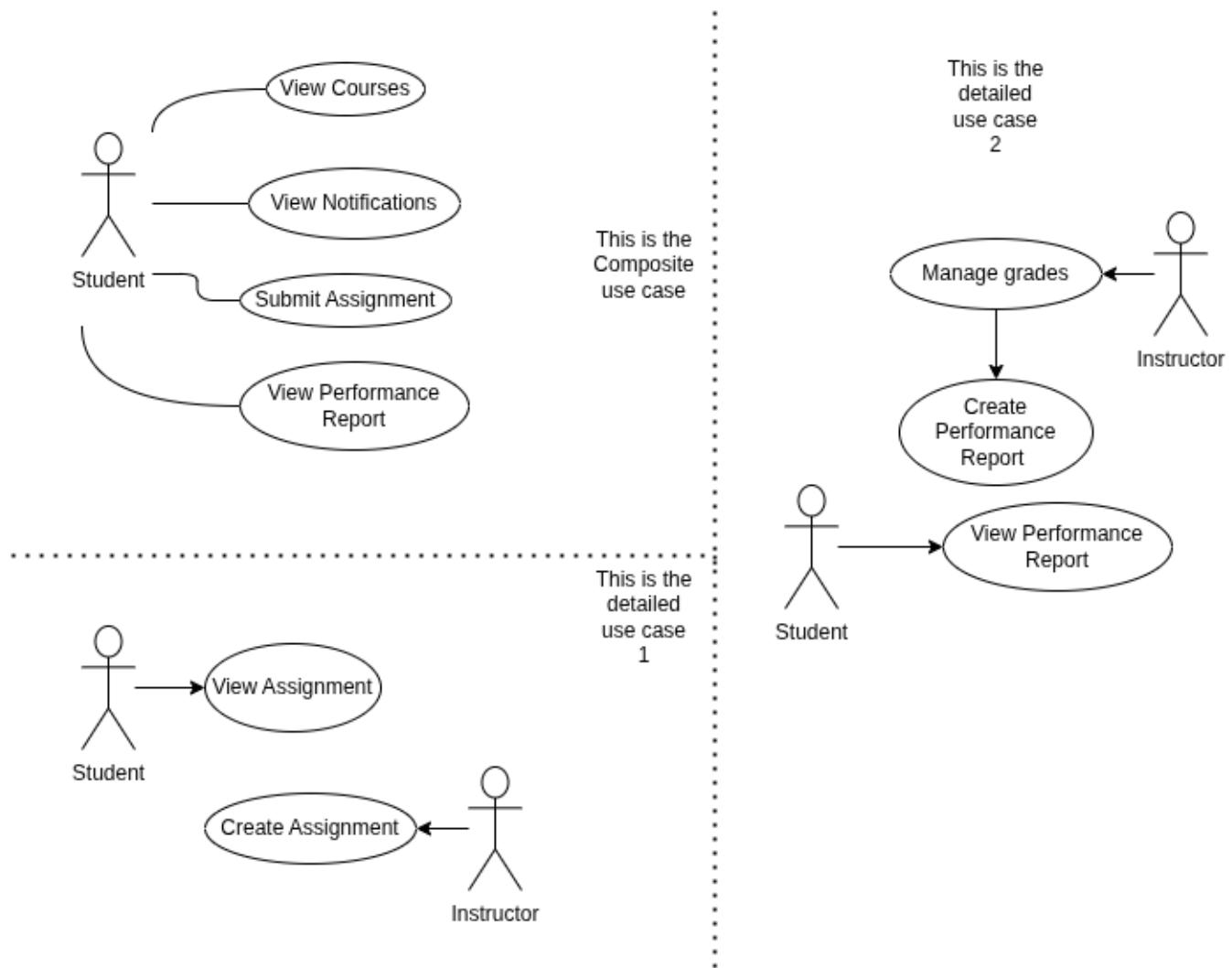


Figure 4: Use Case Diagram & Detailed

Explanation: This figure contains one composite use case diagram and two detailed use case diagrams. The composite diagram provides an overview of all system functionalities for the primary actor, while the two detailed diagrams expand specific use cases, showing additional actors and system responsibilities.

6. Use Case Descriptions

6.1 Submit Assignment

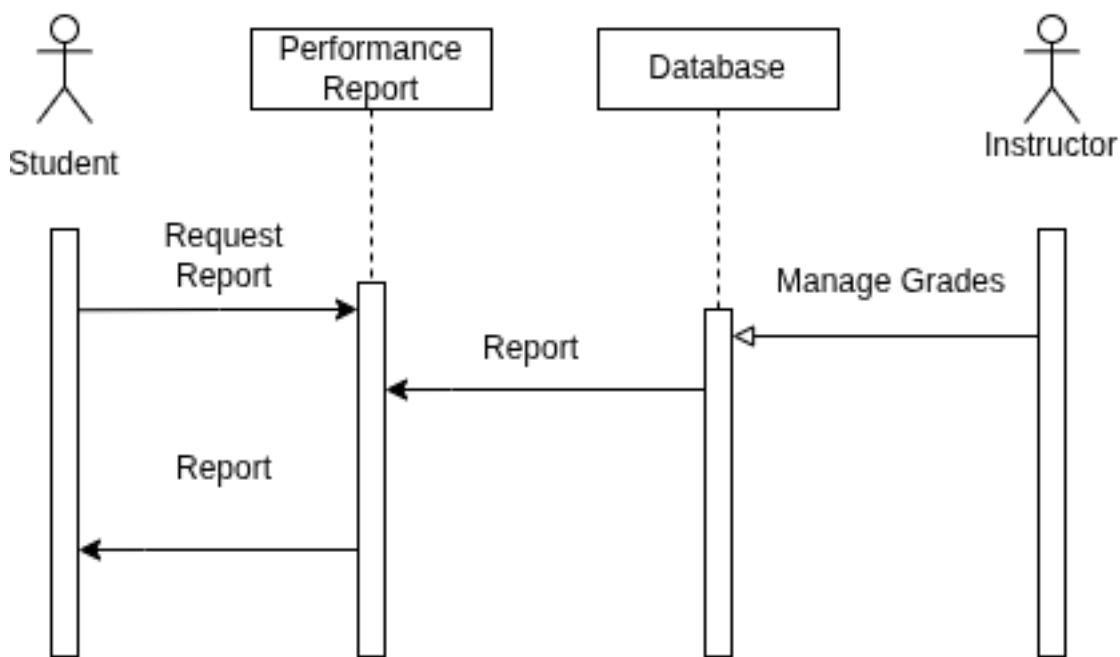
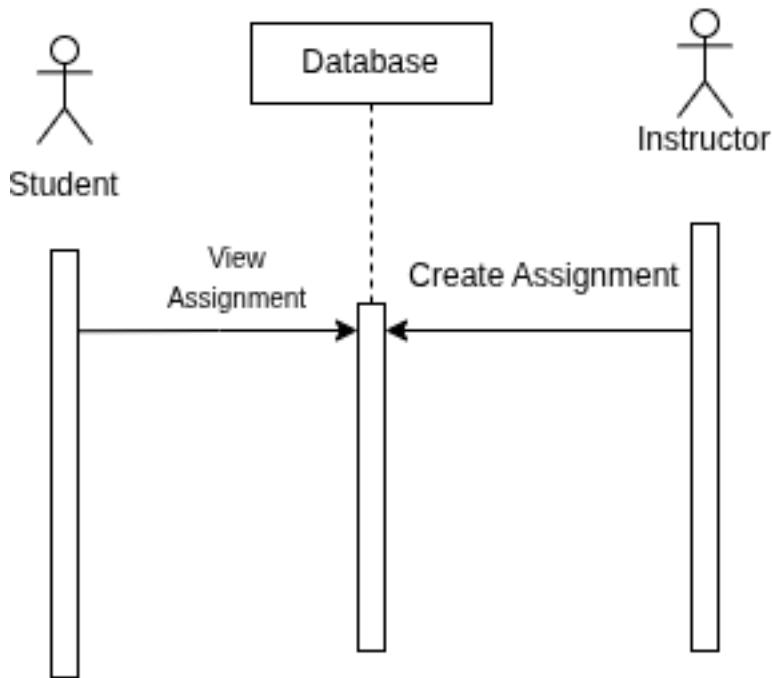
Describes the student submitting an assignment. Preconditions include authentication and course enrollment. Steps: select assignment, upload solution, and receive confirmation. Postconditions: submission is stored and available for instructor review.

6.2 Upload Course Material

Describes the instructor uploading course materials. Steps: select course, upload files, publish content. Postconditions: materials are stored and accessible to students with proper access control.

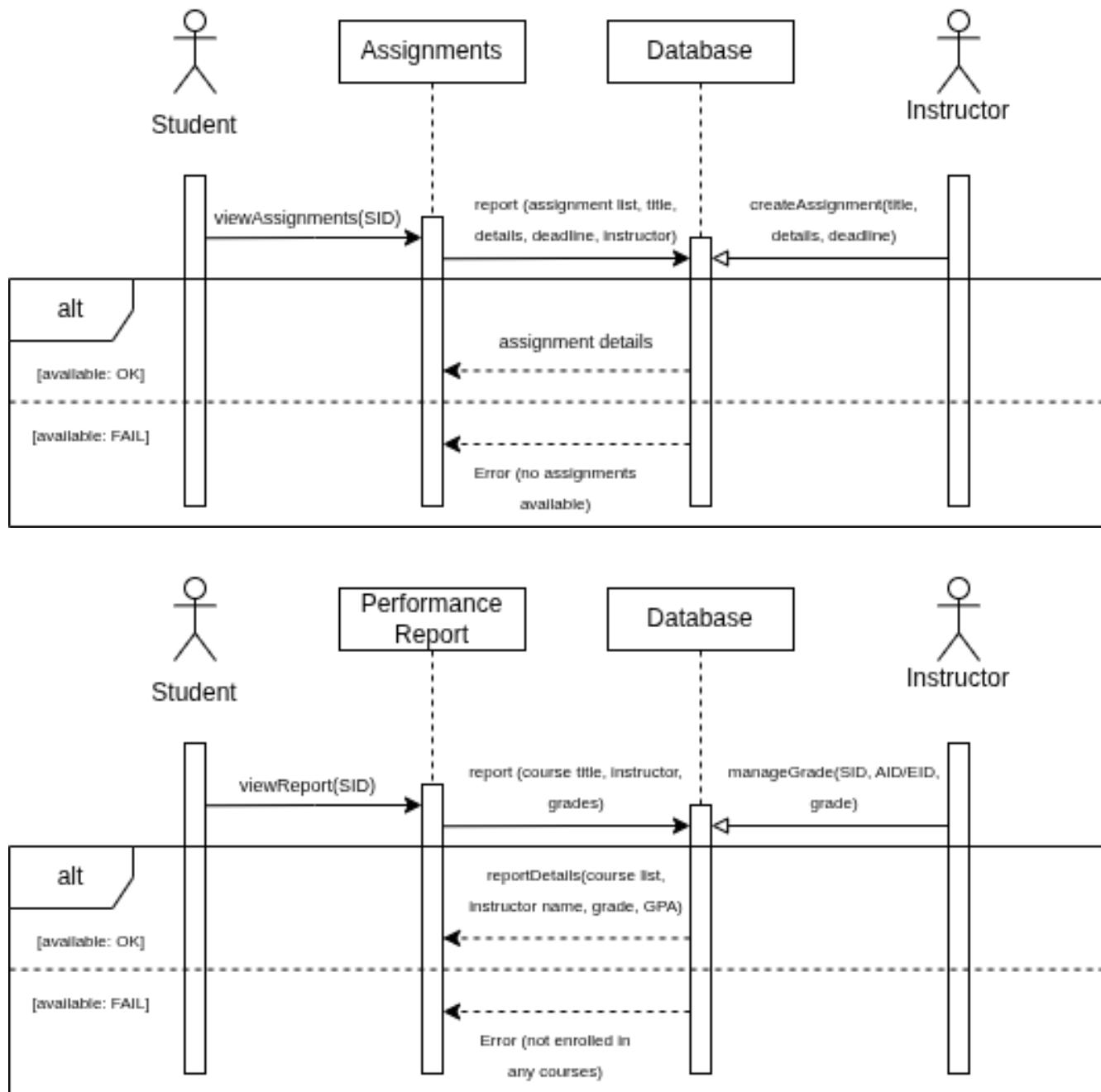
7. Sequence Diagrams

7.1 High-Level Sequence Diagrams



Explanation: These diagrams show simplified actor-system interactions for stakeholders to understand overall system flow without technical details.

7.2 Detailed Sequence Diagrams



Explanation: Detailed diagrams provide technical views, showing object-level communication, method calls, and data exchange needed to implement the selected use cases.

8. Class Diagram

Explanation: The class diagram represents core entities like User, Student, Instructor, Course, Assignment, Submission, and Grade. Attributes, operations, and relationships (inheritance, association, aggregation) illustrate the system structure.

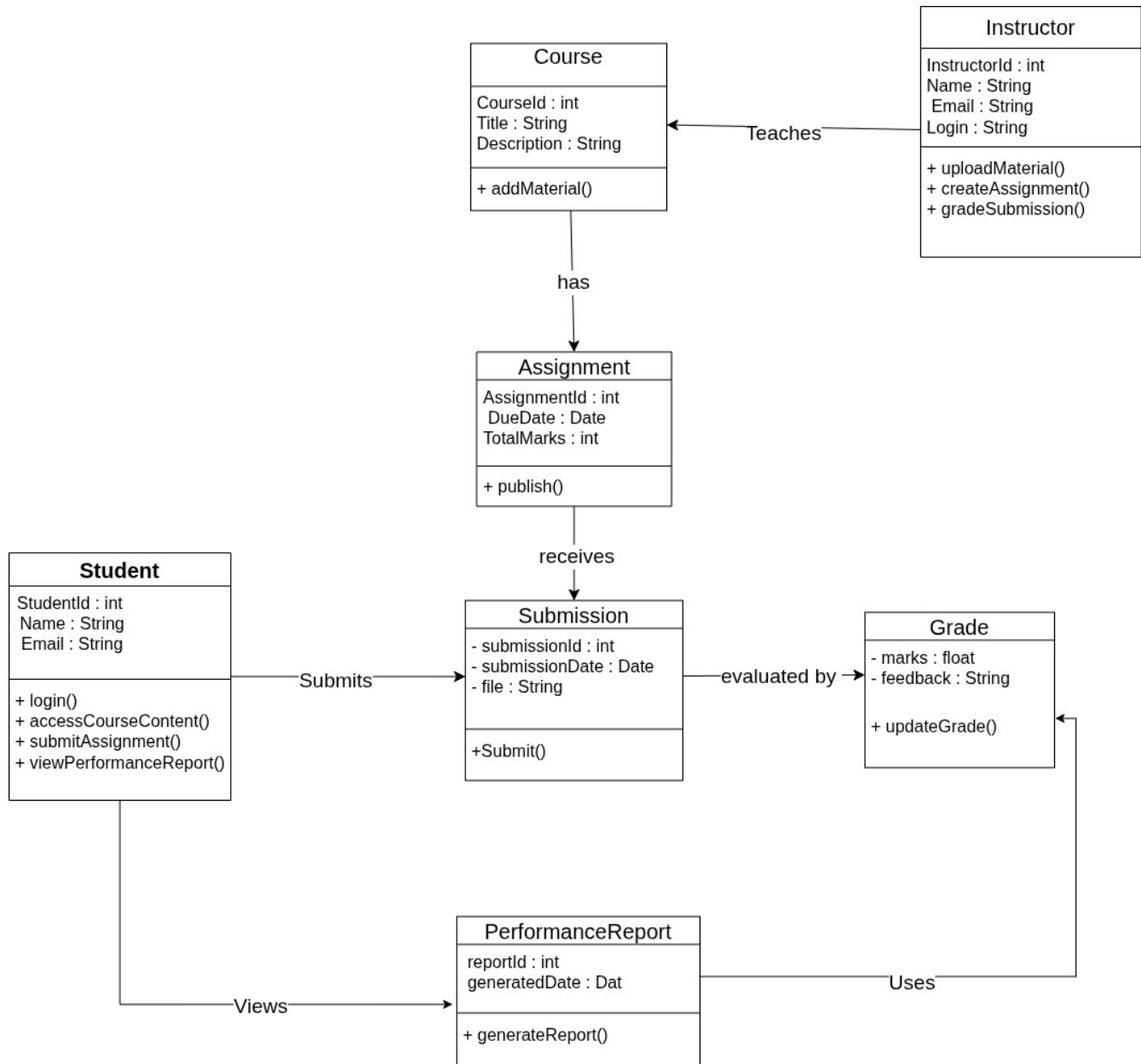


Figure 5: Class Diagram

9. Behavioral Modeling

9.1 State Diagram

Explanation: Models the lifecycle of an assignment submission. States include created, submitted, graded, and finalized. It shows how the system transitions between states in response to events.

9.2 State Stimulus Table

The table defines triggering events and resulting state changes, ensuring consistent system behavior and complementing the state diagram.

10. Conclusion

This report describes the analysis and design of the Learning Management System using UML modeling. Diagrams and explanations collectively define the system's structure, behavior, and interactions, fulfilling project requirements and demonstrating proper software engineering practices.

Assignment Lifecycle State Diagram

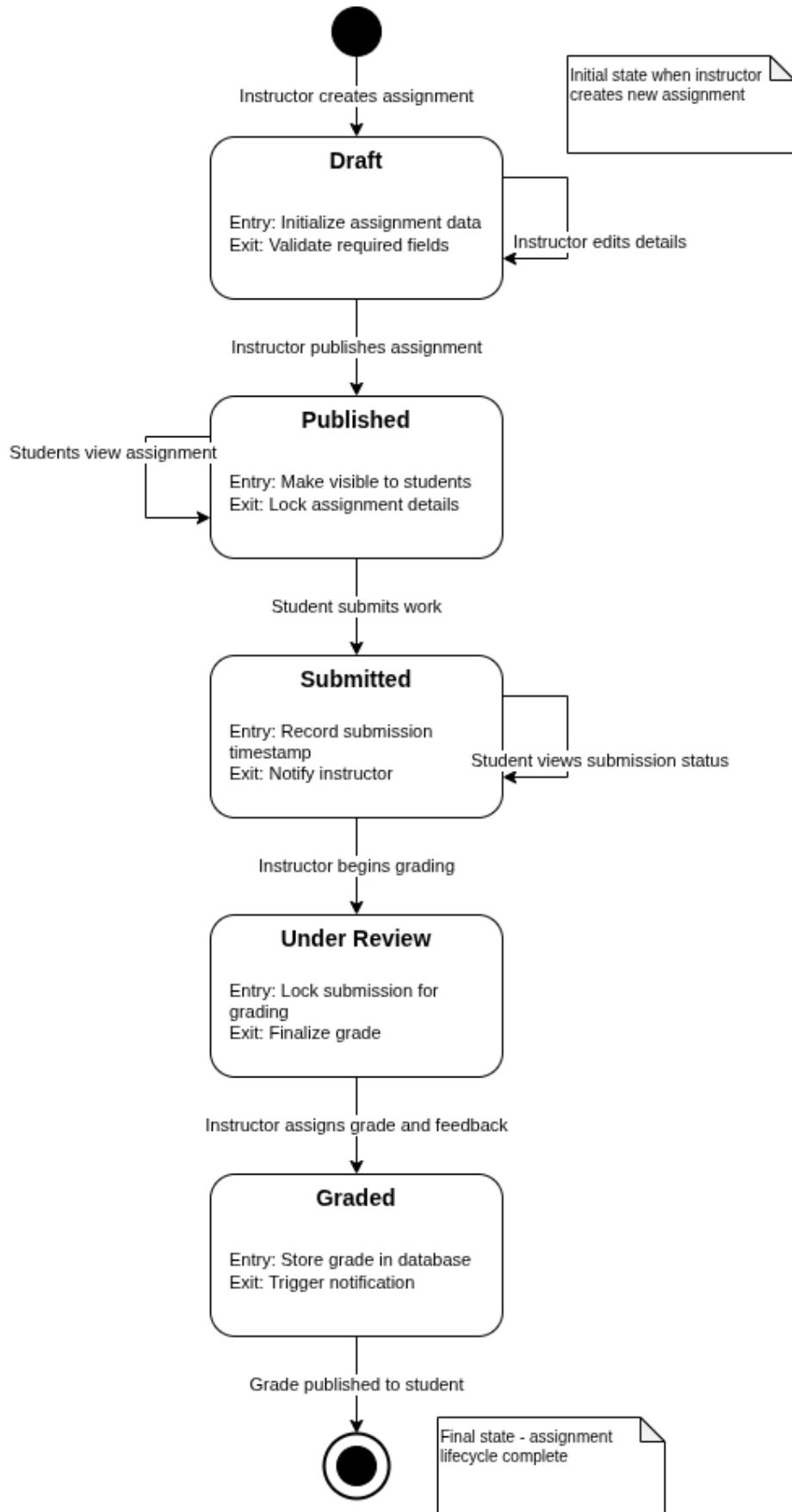


Figure 6: State Diagram
10