

FACULTY **Exact sciences**

"Computer Science Department"



PROJECT REPORT

University Academic Platform

STUDENT CONTRIBUTORS

- **DIFALLAH AKRAM**
- **SELLAMI MOHAMED ODAY**

1. Introduction

1.1 *Brief Overview*

This project presents a comprehensive and secure Academic Learning Platform designed to support teaching and learning in a university environment. The system provides a digital interface for two user roles: teachers and students. Inspired by platforms like Moodle, it enables streamlined access to learning materials, interactive assessments, and structured academic workflows.

Core functionalities include secure user registration and login using validated institutional matricules, role-based access control, course creation and distribution, assignment of exercises and practical works, and quiz-based evaluations. Students can view and interact with assigned content, submit responses, and manage their profile, while teachers can manage academic content, monitor participation, and maintain control over learning materials.

The application was developed using Java for the backend, JavaFX (FXML) for the user interface, and MySQL for persistent data storage. Security features such as password hashing and database-driven validation ensure data integrity and controlled access. The result is a modular, extensible learning system ready to support core educational activities and future enhancements.

1.2 *Outline of the Report*

- **Section 2:** Analysis and Specification (Functional and Non-Functional requirements)
- **Section 3:** Software Design (UML diagrams)
- **Section 4:** Implementation Technologies
- **Section 5:** User Guide
- **Section 6:** Project Review and Future Work

2. Analysis and Specifications

2.1 Functional Requirements

The table below outlines key actions supported by each user role in the platform, including students and teachers, and the system's core operations.

Role	Action	Description
Student	View/Submit	Access and submit quizzes, exercises
Student	Edit Profile	Update personal data
Teacher	Manage Courses	Create/edit content, view submissions
Teacher	Profile	Secure view/edit
All Users	Auth/Login	Secure login/logout with hashed credentials

2.2 Non-Functional Requirements

These requirements describe how the system should behave — focusing on security, performance, usability, and its ability to scale and remain reliable.

Category	Requirement Summary
Security	Passwords should be hashed; roles should be enforced
Performance	Operations should complete in < 2s under normal load
Usability	Interface should be intuitive and categorized
Scalability	System should support modular growth
Reliability	Input validation and recovery should be built-in

2.3 Use Case Diagrams

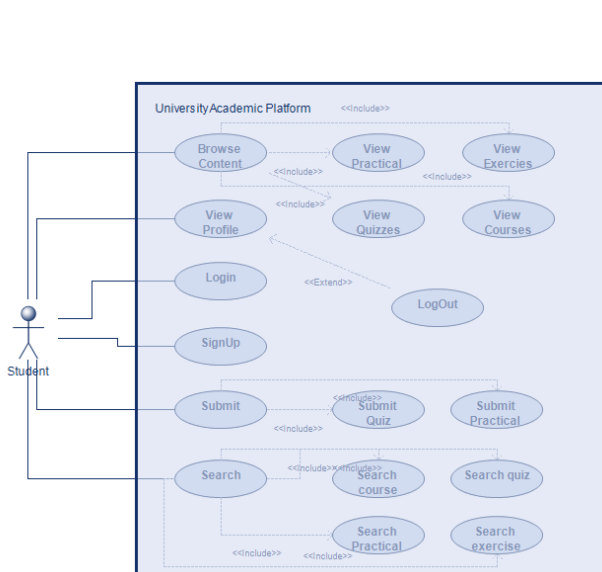


Figure 1: Student use case Diagram

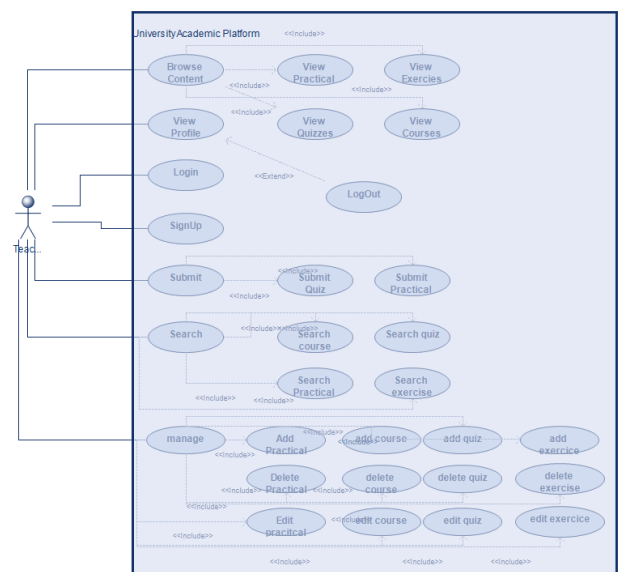


Figure 2: Teacher use case diagram

3. Software Design

3.1 UML Diagrams Overview

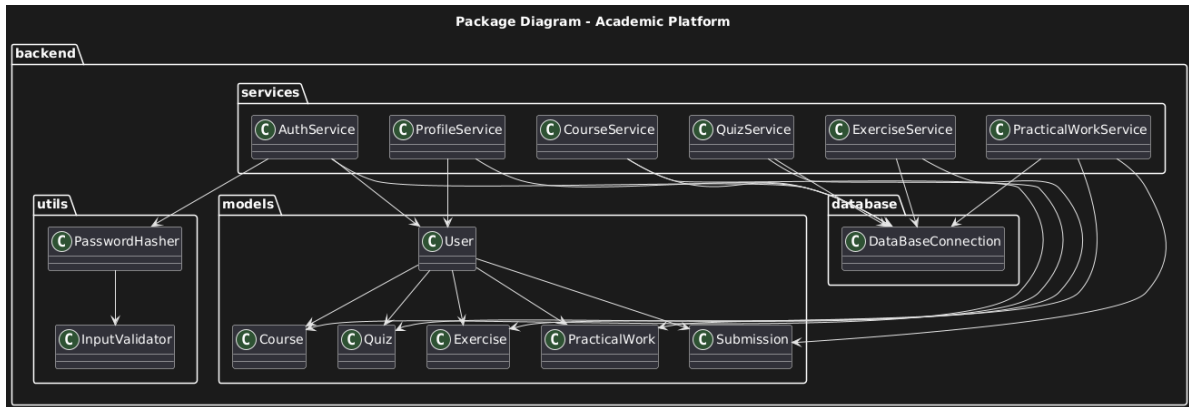


Figure 3: Package diagram for the architecture

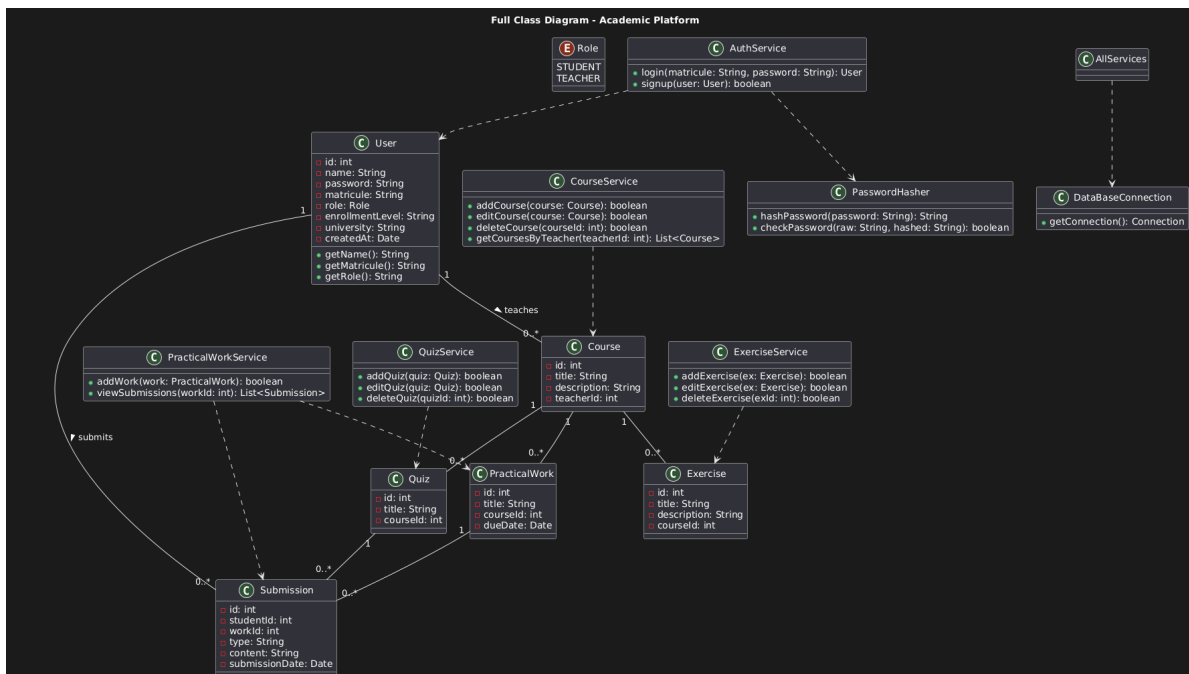


Figure 4: Class Diagram

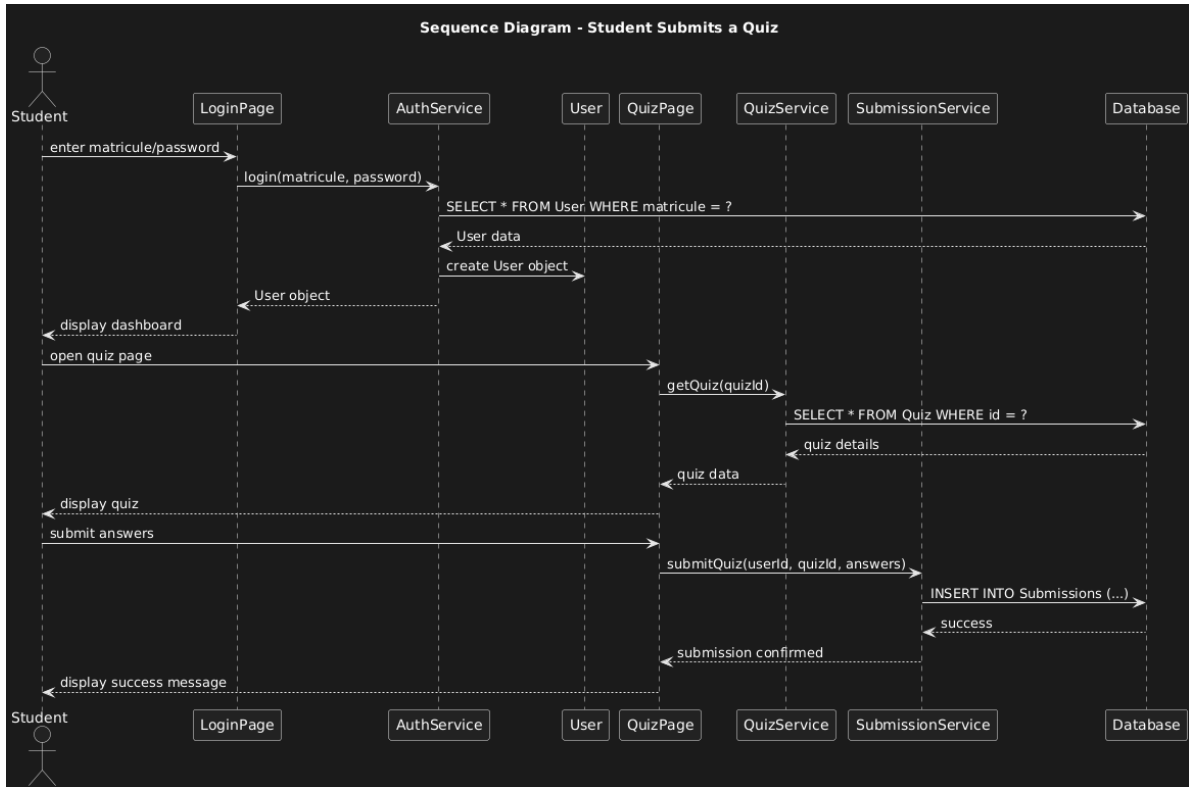


Figure 5: Sequence Diagram - Student Submits a Quiz

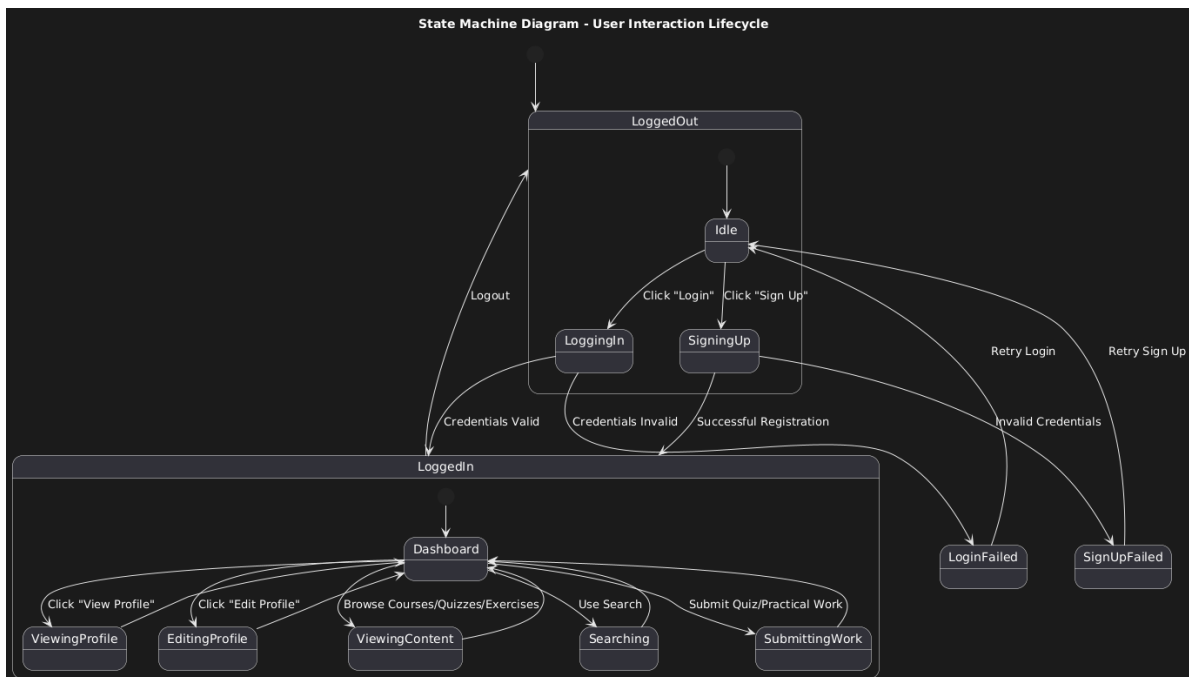


Figure 6: State machine Diagram - User Interaction Lifecycle

4. Implementation

4.1 Technologies Used

Layer	Tool	Description
Frontend	JavaFX/FXML	GUI structure and rendering
Backend	Java 17	Application logic and services
Database	MySQL	User and course storage
Security	BCrypt	Password hashing
Dev Tools	Git, SceneBuilder	Versioning, UI prototyping

4.2 Architecture

The application follows an MVC-like architecture:

- **Model:** POJO-based representations
- **View:** JavaFX interfaces (FXML)
- **Controller:** AuthService, UserService, etc.

Security: Authentication uses hashed credentials with matricule validation.

Modularity: Components organized by services, models, utilities, and database layers.

5. User Guide

5.1 Screenshots

This is how a student can view their courses on the platform.

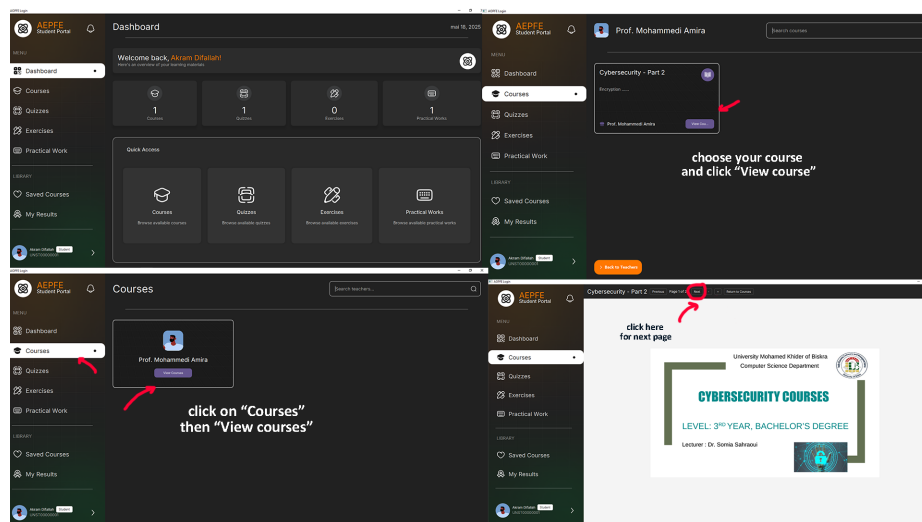


Figure 7: Student view course guide

This is a teacher's guide to adding a new course.

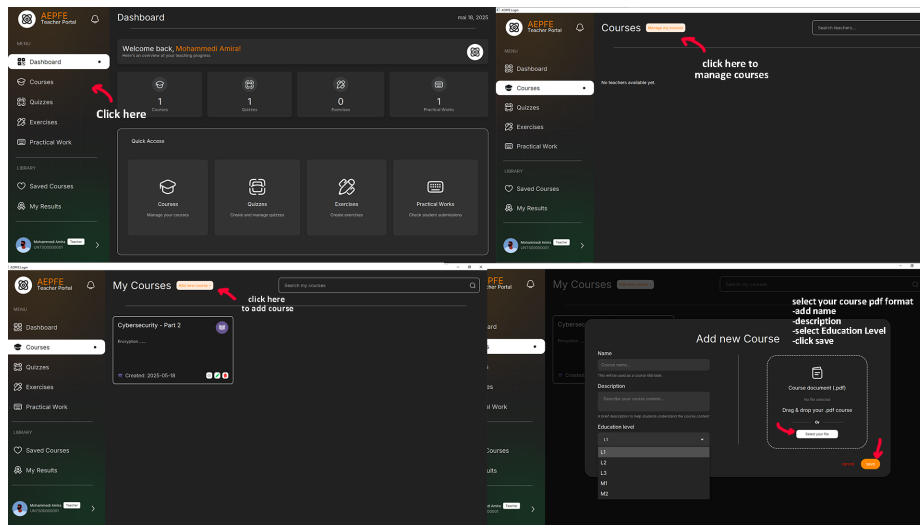


Figure 8: Teacher add course guide

6. Conclusion and Future Work

This Academic Platform achieves a robust, modular, and secure learning management system. It fulfills key requirements for user authentication, content delivery, and profile control.

Future enhancements will include:

- Integration of a frontend notification system (already supported at the database level)
- A graduation module allowing teachers to submit grades, with automatic calculation and display of students' average grade points
- Module assignment features to allow course distribution and management per teacher