University ERP System

Project Report

Team Members:
Pranshu Prakash: 2024426
Dewang Patel: 2024410

November 2025

---

1 Project Overview

1.1 Introduction

The University ERP System is a comprehensive desktop application designed to streamline the core academic operations of a university. Built using a robust Java Swing framework, it provides a secure, role-based environment for Students, Instructors, and Administrators. The system facilitates critical tasks such as course catalog management, student registration, scheduling, and grade processing, emphasizing data integrity and a clean user experience.

1.2 Key Objectives

- **Role-Based Access Control:** Enforces strict separation of duties between Students, Instructors, and Admins to prevent unauthorized access.
- **Data Security:** Implements industry-standard password hashing and utilizes separate database schemas for authentication credentials and academic records.
- **Academic Integrity:** Automates grading calculations based on predefined weights and prevents unauthorized modifications to historical academic records.
- **System Stability:** Ensures robust error handling and includes a "Maintenance Mode" to safely lock the system during updates.

2 System Architecture

2.1 Technology Stack

The application leverages a modern, industry-standard Java stack to ensure reliability and maintainability:

- **Language:** Java
- **User Interface:** Java Swing with **FlatLaf** (Light Flat Look and Feel) for a modern, cross-platform aesthetic.
- **Database:** MySQL, utilizing two distinct databases (auth_db and erp_db) for security isolation.
- **Connectivity:** JDBC with **HikariCP** for high-performance database connection pooling.
- **Security: jBCrypt** for secure password hashing.
- **Export: OpenCSV** for generating downloadable student transcripts.

2.2 Architectural Pattern

The project follows a strict **Layered Architecture** to ensure Separation of Concerns (SoC). The code is organized into distinct packages, ensuring that the User Interface never communicates directly with the Database.

1. **Presentation Layer (edu.univ.erp.ui):** Contains all JFrame and JPanel classes. It handles user inputs and displays data. It communicates *only* with the Service Layer.
2. **Service Layer (edu.univ.erp.service):** Acts as the "Brain" of the application. It contains business logic, enforces access rules (e.g., checking Maintenance Mode), and validates inputs before passing them to the Data Layer.
3. **Data Access Layer (edu.univ.erp.data):** Acts as the Data Access Object (DAO). It is the only layer containing SQL queries. It handles CRUD operations for both the auth_db and erp_db.
4. **Domain Layer (edu.univ.erp.domain):** Contains simple POJOs (Plain Old Java Objects) used to transfer data between layers (e.g., StudentProfile, GradebookRow).

2.3 Database Design & Security

The system implements a "UNIX shadow" style security model by physically separating credentials from user profiles:

- **auth_db:** Stores only authentication data (username, role, password_hash, failed_attempts). No plain-text passwords are ever stored; only bcrypt hashes are used.
- **erp_db:** Stores academic data (students, courses, enrollments, grades).
- **Linking:** The two databases are linked logically via a shared user_id, ensuring strict data segregation.

3 Key Features & Logic

3.1 Access Control & Maintenance

The system enforces strict access rules through the AccessChecker and Service classes.

- **Role Enforcement:** The UI dynamically adjusts to show only relevant tabs (e.g., "Course Management" is hidden for Students). The Service layer double-checks these permissions before any database write.
- **Maintenance Mode:** A global flag in the database controls system availability. When an Admin toggles Maintenance Mode ON:
  - A red banner appears on all Student and Instructor dashboards.
  - All "write" operations (Register, Drop, Save Grade) are immediately blocked at the Service level.

3.2 Grading Logic

The system enables Instructors to maintain a digital gradebook. The grading logic is automated within the InstructorService class.

Weighting Formula:
The final grade is calculated based on five distinct assessment components. The system accepts raw scores (out of 100) for each component and applies the following weighted formula:

- **Quiz 1:** 10%
- **Quiz 2:** 10%
- **Project:** 20%
- **Mid-Sem Exam:** 25%
- **End-Sem Exam:** 35%

Calculation Process:

$$(Q1 * 0.10) + (Q2 * 0.10) + (Project * 0.20) + (Mid * 0.25) + (End * 0.35)$$

Grading Scale:
The numeric Final Score is automatically converted into a Letter Grade:

- **A+**: 90-100
- **A-**: 80-89.9
- **B**: 65-79.9

- **B-**: 50-64.9
- **C**: 40-49.9
- **D**: 30-39.9
- **F**: Below 30

4 User Walkthrough

4.1 Student Flow

Students log in to a personalized dashboard where they can view the course catalog, manage their schedule, and track their academic progress.

- **Registration:** Students can browse the "Monsoon 2025" catalog and register for sections. The system checks capacity limits and duplicates.
- **Timetable:** The "My Timetable" tab displays a sorted, 5-day view of their schedule.
- **Grades:** The "My Grades" tab presents a clean, card-based layout showing breakdown scores.

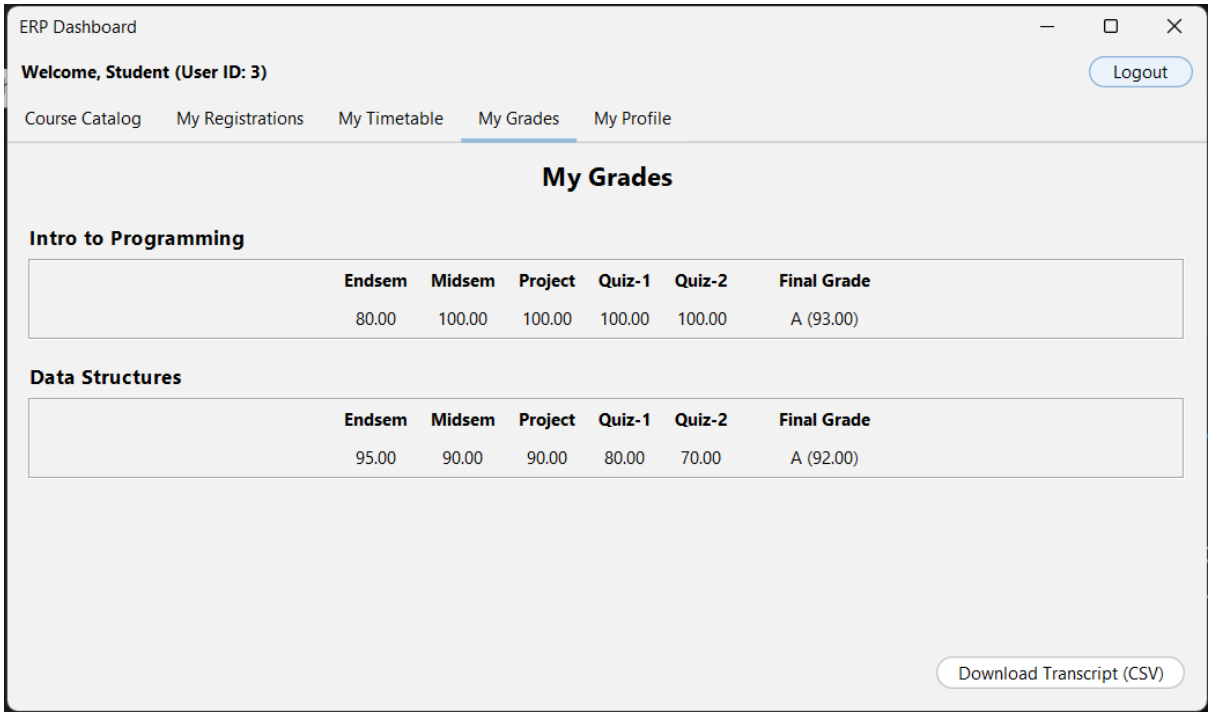

Figure 1: Student's "My Grades" view.

4.2 Instructor Flow

Instructors have access to a dedicated "My Sections" tab.

- **Section Management:** Instructors can only view sections they are assigned to teach.
- **Gradebook:** An editable table allows entry of raw scores. The "Calculate Final Grades" button computes the weighted average and saves it to the database.

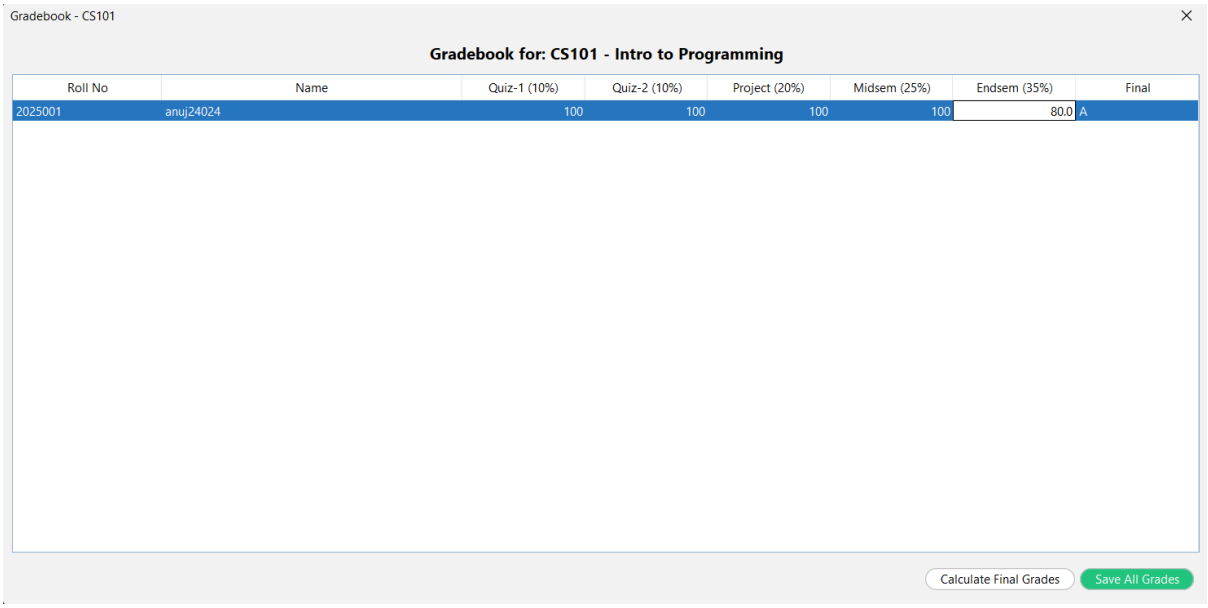

Figure 2: Instructor's gradebook with editable cells.

4.3 Admin Flow

Administrators have full control over the system configuration.

- **Course Management:** Admins create courses and schedule sections using a dynamic day/time builder.
- **User Management:** Admins create new Student and Instructor accounts, automatically populating both Auth and ERP databases.

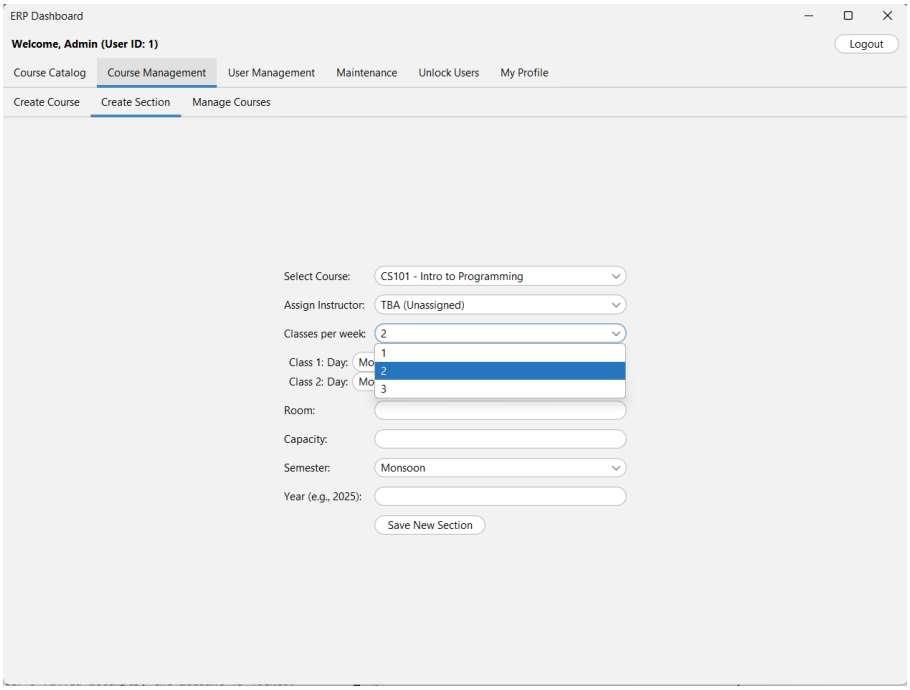

Figure 3: Admin panel for creating new sections.

5 Bonus Features Implemented

We implemented several advanced features beyond the core requirements:

5.1 CSV Transcript Export

Students can download an official transcript directly from their dashboard. The TranscriptService generates a formatted CSV file containing their course list, component scores, and final grades.

5.2 Account Security & Lockout

To prevent brute-force attacks, the AuthService tracks failed login attempts.

- **Warning:** Every failed attempt, the user receives a warning.
- **Lockout:** After 5 failed attempts, the account is locked.
- **Unlock:** Admins have a dedicated "Unlock Users" panel to reset these accounts.

5.3 Self-Service Password Change

A "My Profile" tab allows all users to securely change their password. The system verifies the old password and enforces a minimum length requirement before updating the hash.

6 Testing Summary

The system has undergone rigorous testing to ensure stability and security.

- **Functional Testing:** Verified all core flows (Login, Register, Grade, Admin Create).
- **Security Testing:**
  - Verified that students cannot register when Maintenance Mode is ON.
  - Verified that instructors cannot save grades for sections they do not own.
  - Verified that SQL injection is prevented by using Prepared Statements. (preventing attacks like, ' OR '1'='1).
- **Data Integrity Testing:** Confirmed that deleting a course is blocked if active sections exist, preserving database referential integrity.

7 How to run?

Before running the application, ensure the following are installed:

- Java Development Kit (JDK) 11 or higher.
- MySQL Server (running on port 3306).

Database Setup

The application requires two databases (auth_db and erp_db) to be initialised before launch.

1. Open your MySQL terminal or Workbench.
2. Run the provided SQL seed scripts included in the submission folder.
3. Verify that the default users (admin1, etc.) have been created in the users_auth table.

Running the Application

The application is packaged as an executable JAR file. Open your terminal, navigate to the project directory, and execute:

```
cd path/to/project
java -jar target/UniversityERP-1.0-SNAPSHOT.jar
```

User Credentials

The system is pre-seeded with the following accounts for testing:

Role	**Username**	**Password**
Student	anuj24024	anuj24024
Student	rohit24567	rohit24567
Student	arun24100	arun24100
Instructor	pankaj_	pankaj__
Instructor	tillo_	tillo___
Instructor	subhajit_	subhajit_
Admin	admin1	iamadmin
Admin	admin2	iamadmin

Table 1: Default Login Credentials

8 Entity Relationship Diagram (ERD)

The ERD below illustrates the separation between the Authentication Database (auth_db) and the ERP Database (erp_db), linked logically by the user_id.

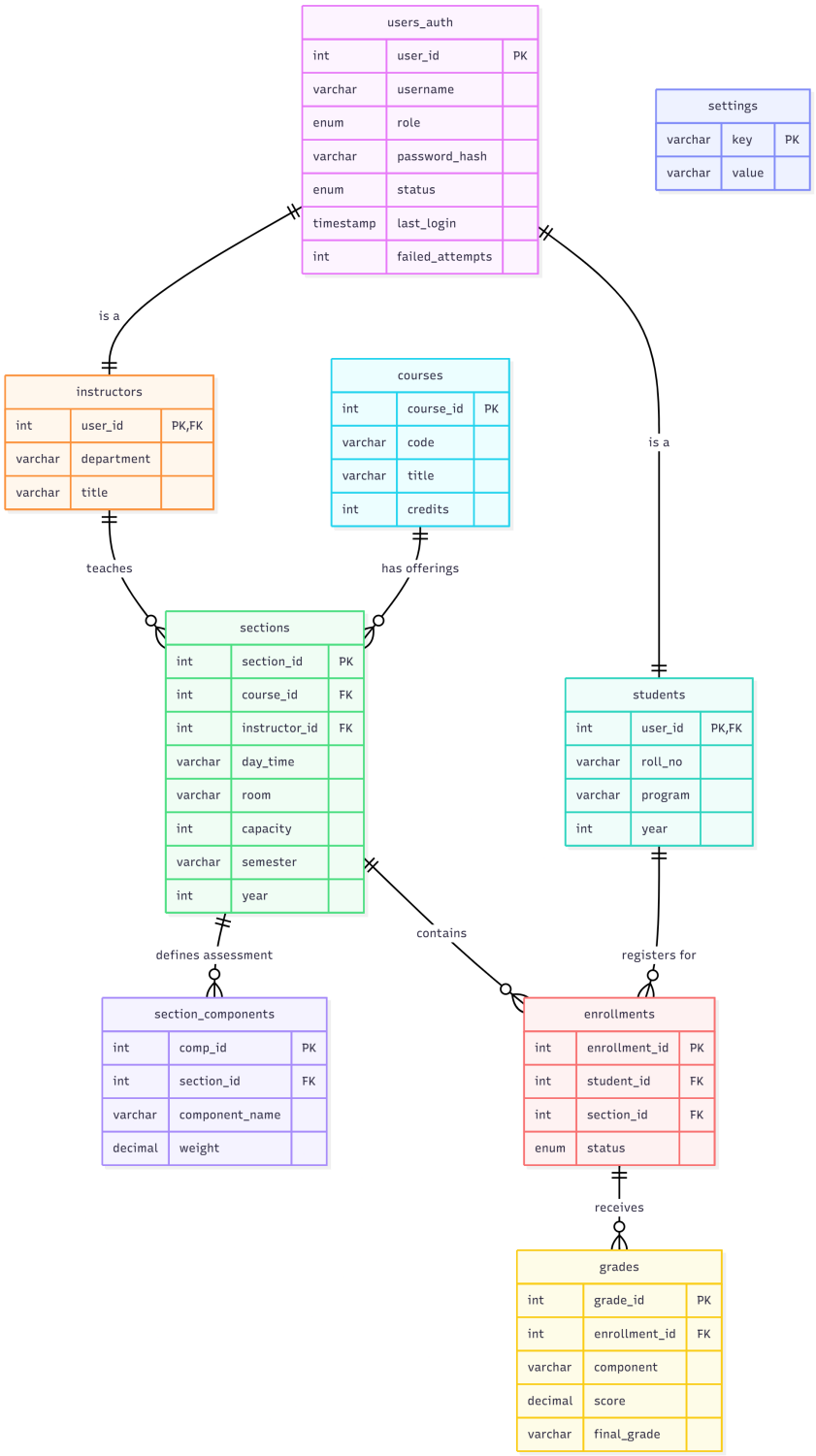

Figure 4: Entity Relationship Diagram showing database separation.

9 Flowcharts

9.1 Student Registration

This flowchart details the logic enforced by the StudentService when a user attempts to register for a course. Note the security checks for Maintenance Mode and Capacity.

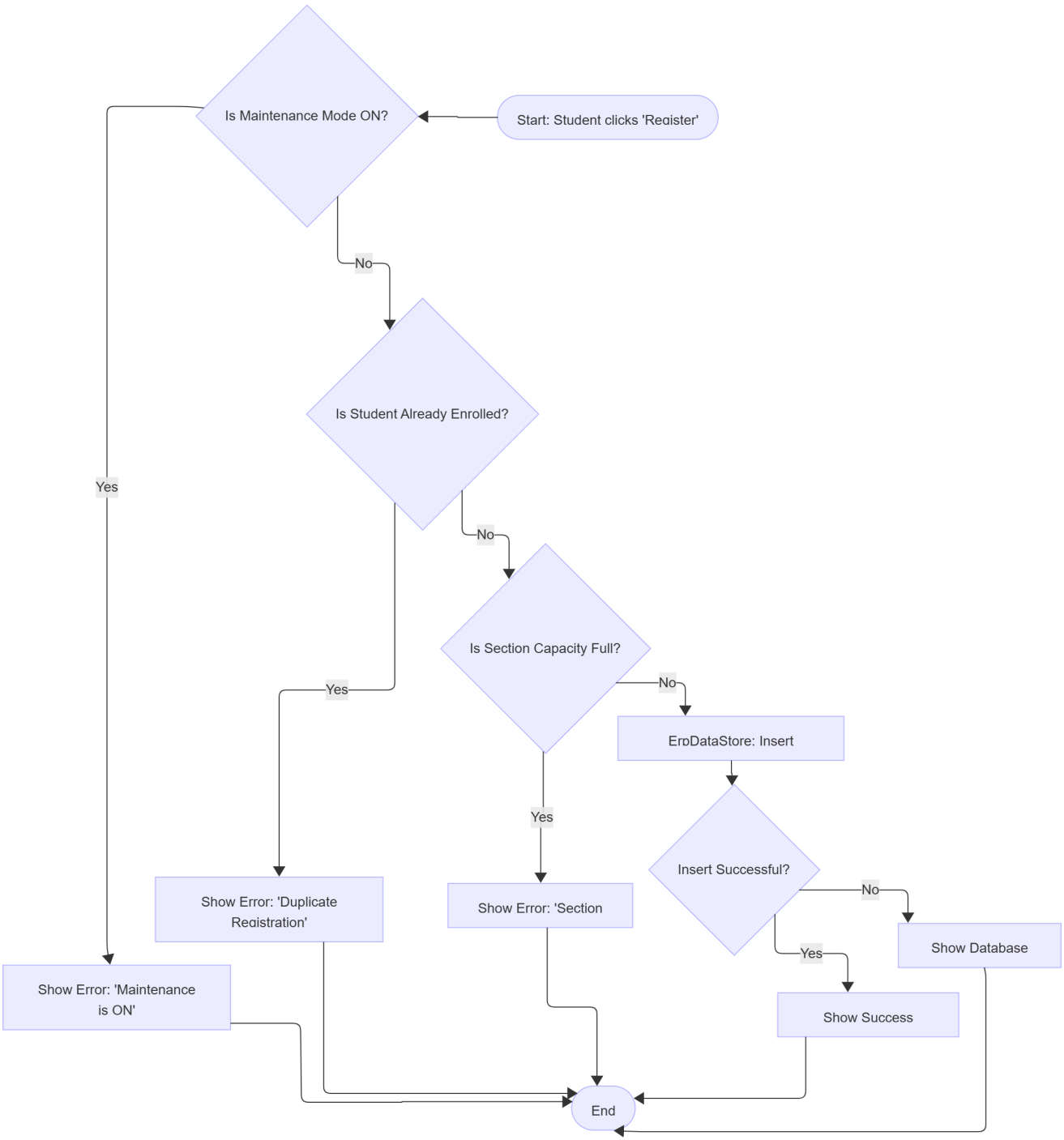

Figure 5: Logic flow for Student Course Registration.

9.2 Instructor Grading

This flowchart illustrates how the Instructor can grade the students of their section.

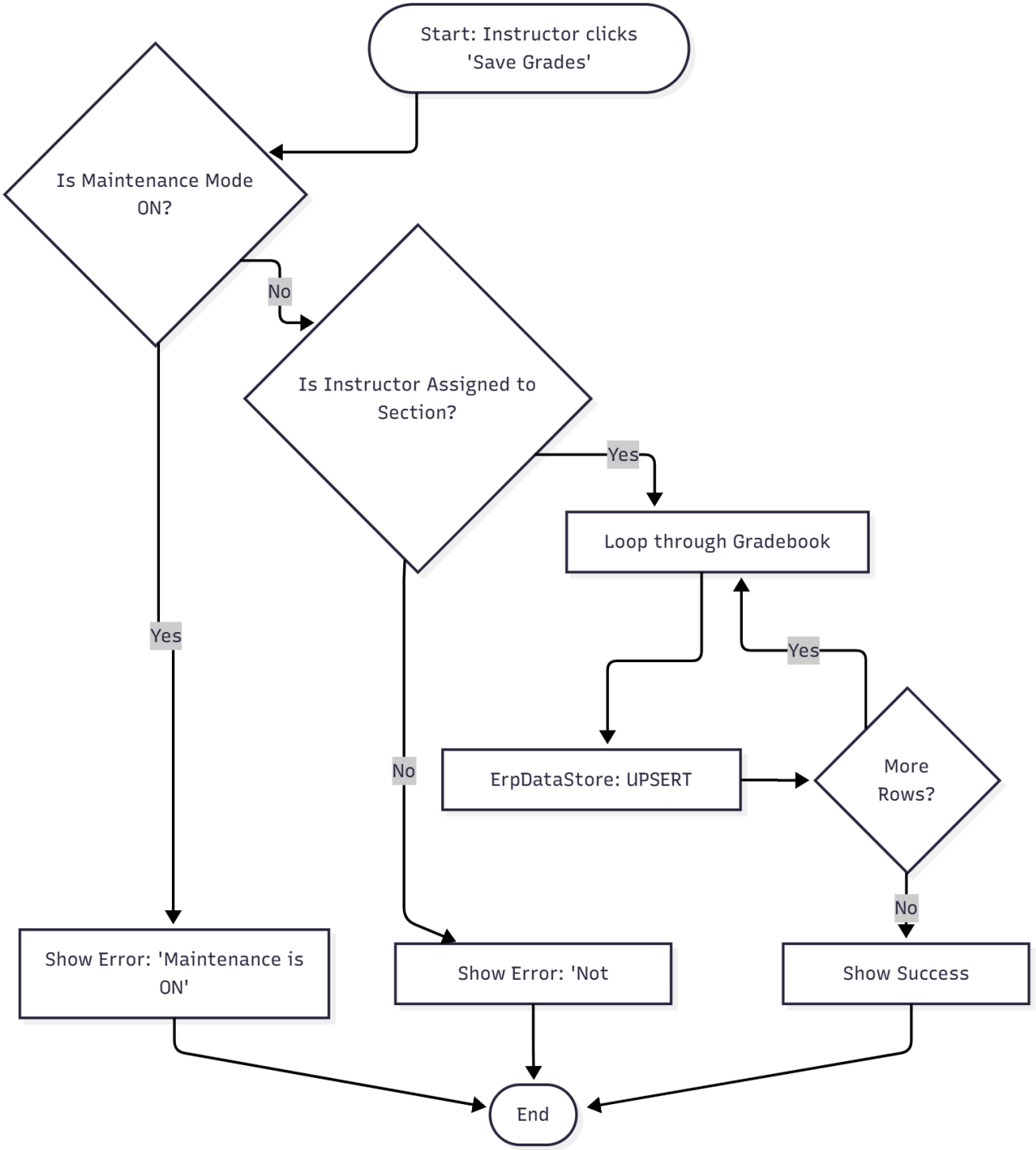

Figure 6: System-wide Maintenance Mode enforcement flow.

9.3 Maintenance Mode Toggle

This flowchart illustrates how the Administrator can lock the system globally. The AccessChecker reads this flag to block subsequent write operations.

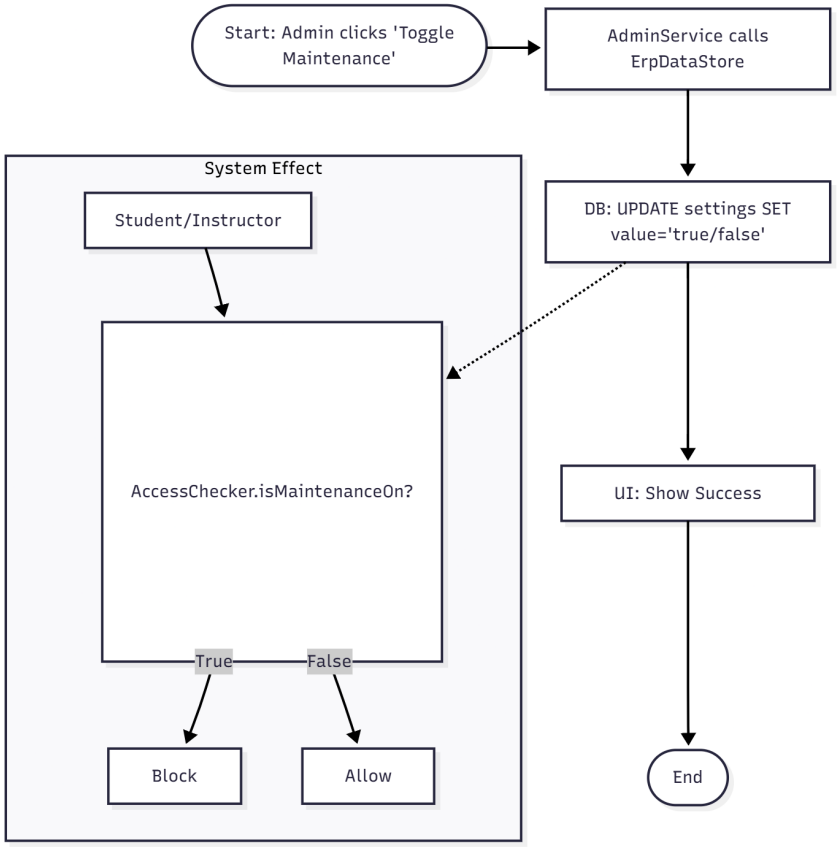

Figure 7: System-wide Maintenance Mode enforcement flow.

10 Use Case List

User	Use Cases
Student	Login; Change Password; View Course Catalog; Register for Section; View Timetable; View Grades; Drop Section; Download Transcript
Instructor	Login; Change Password; View My Sections; Enter Scores; Calculate Final Grade; View Statistics; Drop Student
Admin	Login; Change Password; Create User; Create Course & Section; Toggle Maintenance Mode; Unlock User Account

Table 2: Users and Their Use Cases in the University ERP System

11 Testing Plan

The following test cases were executed to verify system functionality, security, and role-based access control.

Role	Test Scenario	Expected Result	Status
Stu	Student attempts to run Admin dashboard directly from terminal (UI Bypass)	Access denied; Admin tabs not loaded.	**Pass**
Inst	Instructor attempts to delete a user via direct service call	Access denied; Backend service blocks action.	**Pass**
Stu	Student tries to register/drop while Maintenance Mode is ON	Operation blocked with "Maintenance ON" error.	**Pass**
Inst	Instructor tries to grade a section they are not assigned to	Blocked by Access Checker; "Not Authorized" error.	**Pass**
User	Account Lockout: User enters wrong password 5 times	Account status changes to LOCKED; login blocked.	**Pass**
Adm	Admin creates a new Student user	User created in Auth DB and Profile created in ERP DB.	**Pass**
Adm	Admin creates a new Course and Section	New section appears in Course Catalog immediately.	**Pass**
Adm	Admin toggles Maintenance Mode ON	Red banner appears on Student/Instructor dashboards.	**Pass**
Adm	Admin resets password for locked user	User account unlocked and password updated.	**Pass**
Stu	Student registers for a valid section	Enrollment successful; appears in My Timetable.	**Pass**
Stu	Student tries to drop a course after grades are entered	Drop failed; error message shown to protect data integrity.	**Pass**
Stu	Student downloads transcript	CSV file generated with correct grades.	**Pass**
Inst	Instructor enters raw scores and clicks Calculate	Final Grade calculated (e.g., 88.5 $\rightarrow$ B) and saved.	**Pass**
Inst	Instructor drops a student	Student removed from section and gradebook.	**Pass**

Table 3: Comprehensive Test Plan and Execution Results