PROJECT REPORT

STUDENT MANAGEMENT SYSTEM

To fulfil assignment for the course

Object Oriented And Visual Programming

Lecturer: Williem, M.Sc

Compile by:

Qwyn Celine Djimondo(001202400205)
Tessalonika Angeline Purba(001202400210)

PRESIDENT UNIVERSITY FACULTY OF COMPUTING
NORTH CIKARANG
2025

A. INTRODUCTION

In today’s fast-paced educational environment, managing student information effectively is crucial for institutions to ensure smooth operations and provide high-quality services to students and staff. A **Student Management System (SMS)** is a powerful software solution that centralizes and streamlines the administration of student data, academic records, attendance, grades, and other essential information.

By automating many of the manual tasks traditionally associated with student record-keeping, a Student Management System reduces administrative workload, minimizes errors, and enhances communication between students, faculty, and administrators. The system enables efficient data tracking, easy retrieval of information, and data-driven decision-making, ultimately leading to improved student outcomes and institutional performance.

In this project, we explore the design and implementation of a comprehensive Student Management System that meets the needs of modern educational institutions, ensuring scalability, security, and user-friendliness.

B. OVERVIEW

This project aims to develop a comprehensive Student Management System that streamlines the management of student data, academic records, attendance tracking, grade management, and administrative tasks. The system is designed using Java with NetBeans IDE and Java Swing for the user interface, providing a desktop-based solution tailored to the needs of educational institutions. By automating key processes, the Student Management System reduces manual workload, improves data accuracy, and enhances overall efficiency in managing student information.

C. PROJECT SCOPE AND OBJECTIVES

The Student Management System is designed to serve as a centralized platform for educational institutions to accomplish the following objectives:

• Manage student information.

• Add, update, and track academic records (Grades, Major, Enrollment Date,GPA,)

• Track login and logout activities to ensure secure and accountable system access.

D. SYSTEM ARSITECTURE

Presentation Layer (GUI)

- Login.java
    - Provides the login interface for users to authenticate.
- MainMenu.java
    - Displays the main menu or dashboard after a successful login.
- MajorManagement.java
    - Enables users to manage academic majors (view, add, edit, delete).
- StudentInformation.java
    - Provides an interface for managing student details.
- StudentInformationSystem.java
    - Serves as the main controller for the GUI, initializing and switching between forms.
- txt_add2.java
    - Likely used to add specific text data (like additional student info) through the GUI.
- txt_combo.java
    - Probably used for managing combo box elements or drop-down selections.

Business Logic Layer (Functionality)

- Audit_details.java
    - Handles audit logging for activities like login and logout.
- Emp.java
    - Represents employee or administrator data as a model class.
- MajorManagement.java
    - In addition to UI elements, this file also handles logical operations (like validation and CRUD for majors).
- DbUtils.java
    - Likely a utility/helper class for database interactions and data manipulation (e.g., converting result sets to tables).

Data Layer (Database Connection)

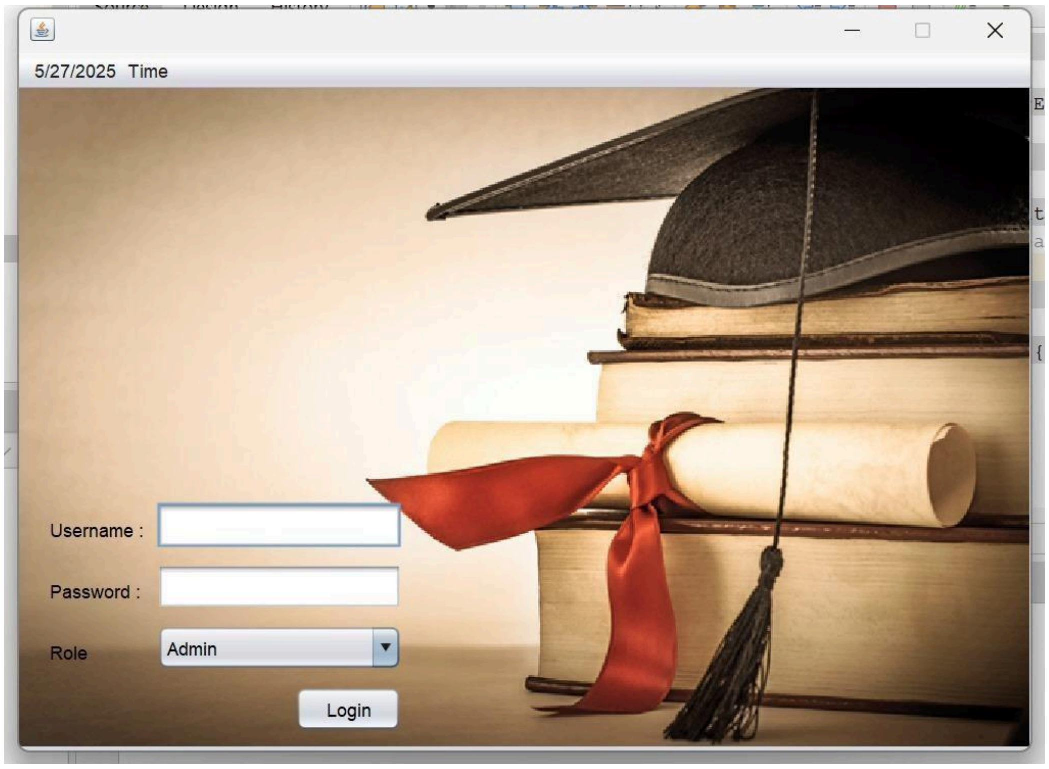
- db.java
    - Manages database connection setup, typically using JDBC to connect to a MySQL database.

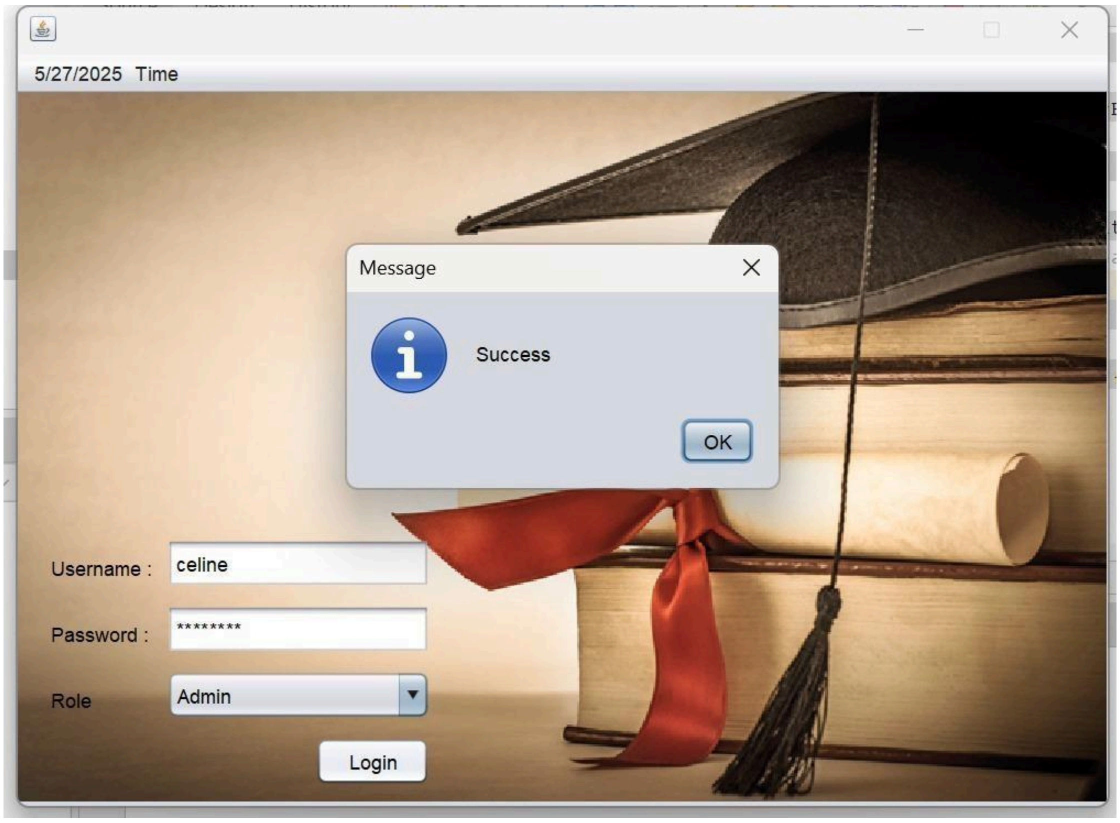
- student.information.system.images
    - This is a resource folder containing images and icons to support the GUI (not actual database content).

E. SYSTEM FEATURES

Login and Authentication

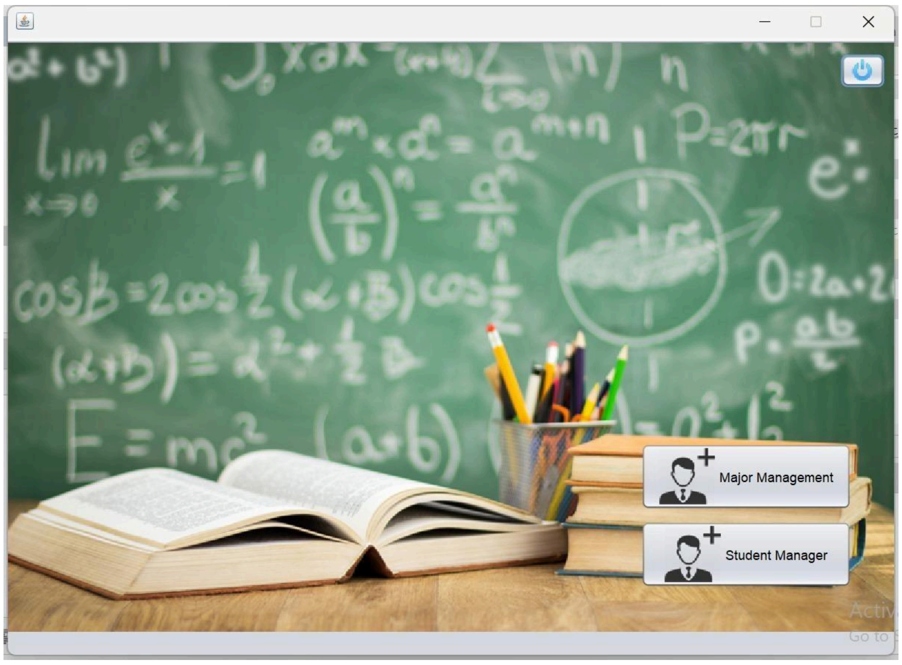
- File: Login.java
- Description:
  Acts as the secure access point for users, featuring a clean and academic-themed login form with fields for Username, Password, and Role.
    - o The Role field enables selection of either "Admin" or "User", ensuring proper access control.
    - o Upon successful login, an informative message box ("Success") is displayed to notify the user of a successful authentication.
    - o The background features a scholarly atmosphere with books, a graduation cap, and a diploma scroll, reinforcing the academic identity of the system.





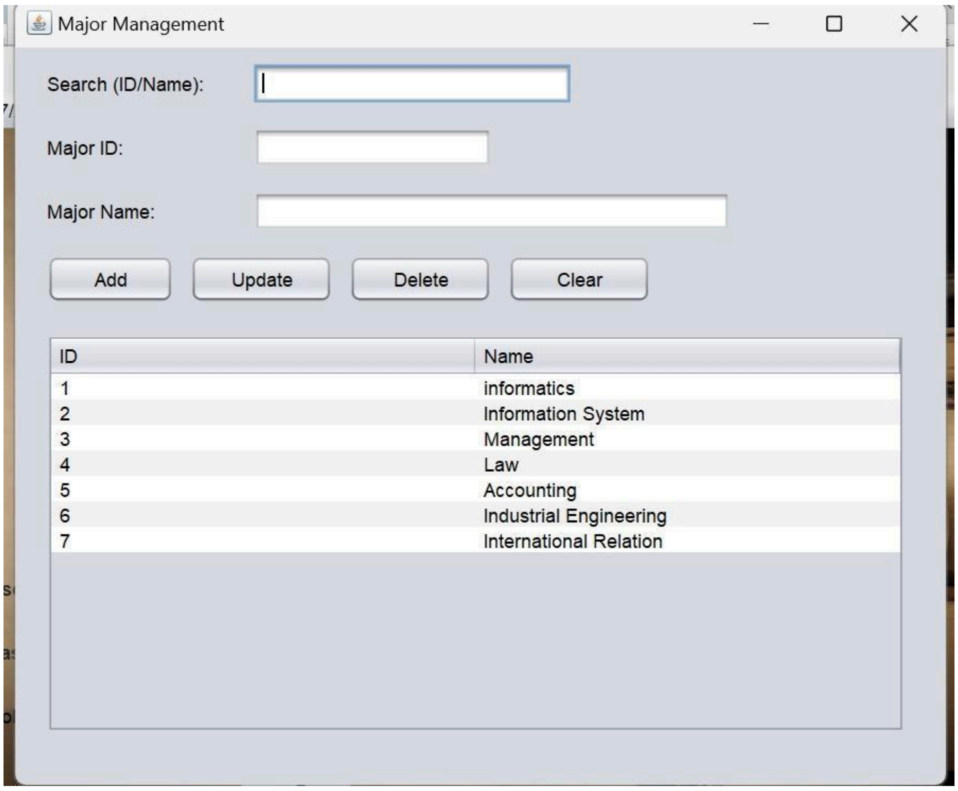
Navigation and Dashboard

- File: StudentInformationSystem.java
- Description:
  Serves as the main entry point of the application, featuring a visually engaging dashboard that resembles a classroom chalkboard with mathematical formulas. The interface includes two primary clickable buttons:
  - Major Management: Directs users to the MajorManagement.java form, where they can manage and update academic majors.
  - Student Manager: Directs users to the StudentInformation.java form, providing access to comprehensive student information management**.**



Major Management

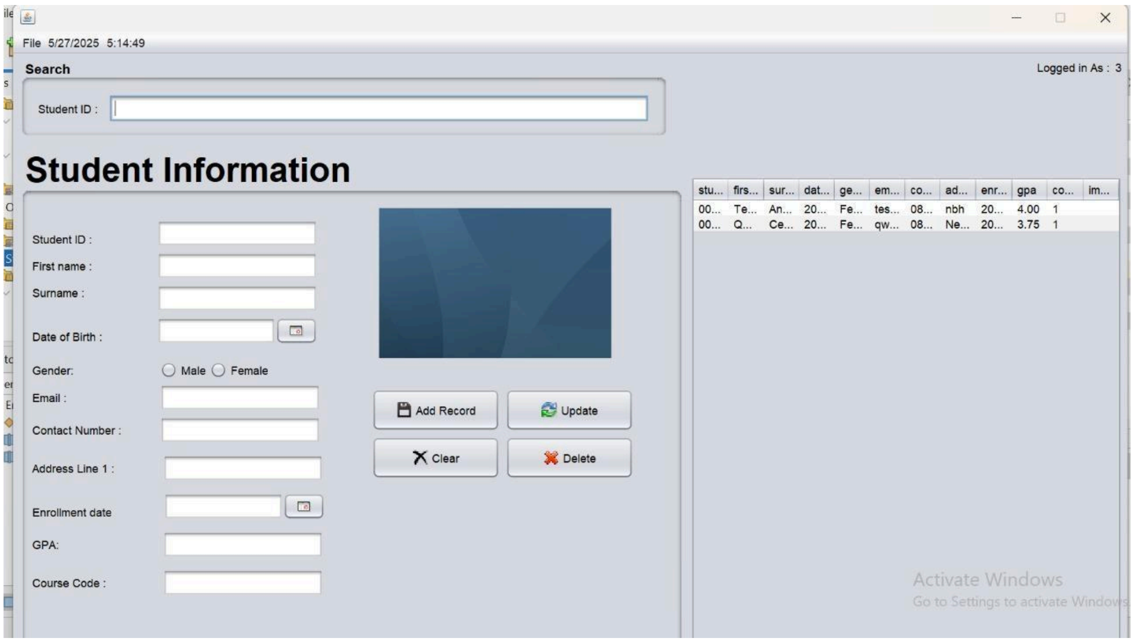
- **File:** MajorForm.java
- **Description:**
  Provides a dedicated form for managing academic majors.
  - Includes search functionality based on **ID** or **Name**, making it easy to locate specific majors.
  - Equipped with input fields for **Major ID** and **Major Name** to add or update major information.
  - Features four action buttons: **Add**, **Update**, **Delete**, and **Clear**, enabling complete CRUD operations on major data.
  - Displays a table view listing the **ID** and **Name** of all available majors for better visibility and management.
  - The intuitive interface supports easy navigation and efficient major management, with a minimalistic, clear layout.



ID	Name
1	informatics
2	Information System
3	Management
4	Law
5	Accounting
6	Industrial Engineering
7	International Relation

Student Manager

- **File:** StudentForm.java
- **Description:**
  Provides a comprehensive form to manage detailed student records:
  - **Search Bar** at the top enables easy lookup by **Student ID**.
  - Form includes fields for **Student ID, First Name, Surname, Date of Birth, Gender, Email, Contact Number, Address, Enrollment Date, GPA,** and **Course Code** for thorough data entry and updates.
  - Features an **image display** for student photos, adding a visual identity to records.
  - Includes four action buttons: **Add Record, Update, Clear,** and **Delete**, supporting full CRUD operations on student data.
  - Displays a detailed **table view** listing all students and their data, enhancing data management and searchability.
  - The interface is designed for ease of use, with a professional and clear layout to manage student information efficiently.

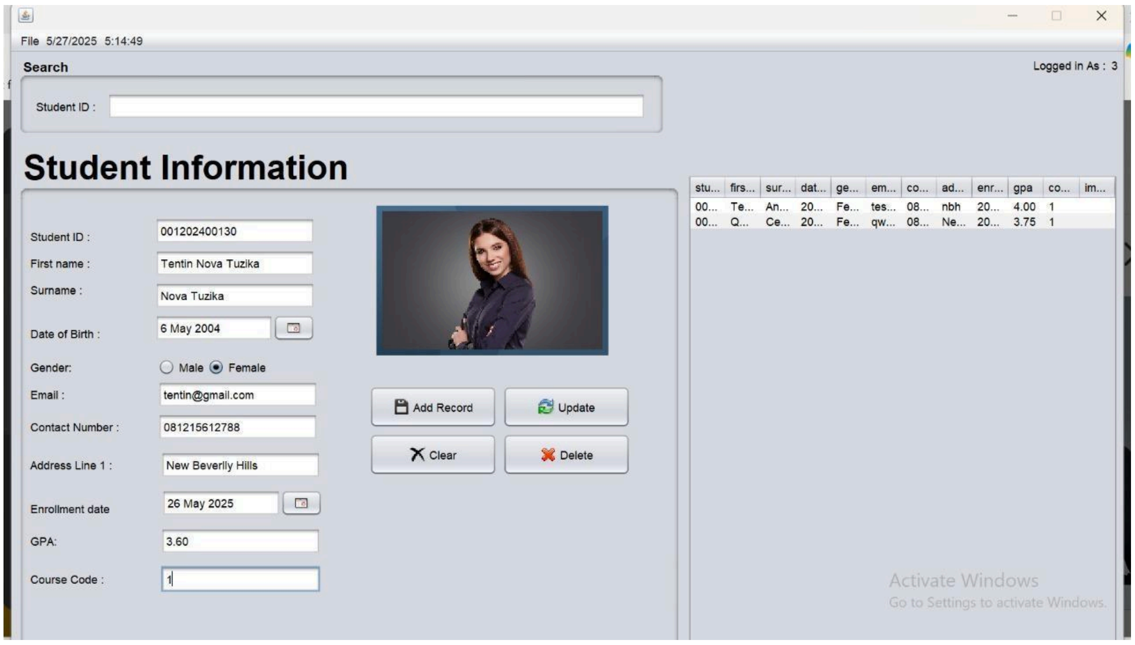


Student ID :

Student Information

Student ID :	
First name :	
Surname :	
Date of Birth :	
Gender:	Male Female
Email :	
Contact Number :	
Address Line 1 :	
Enrollment date	
GPA:	
Course Code :	

stu...	firs...	sur...	dat...	ge...	em...	co...	ad...	enr...	gpa	co...	im...
00...	Te...	An...	20...	Fe...	tes...	08...	nbh	20...	4.00	1	
00...	Q...	Ce...	20...	Fe...	qw...	08...	Ne...	20...	3.75	1	



Student ID :

Student Information

Student ID :	001202400130
First name :	Tentin Nova Tuzika
Surname :	Nova Tuzika
Date of Birth :	6 May 2004
Gender:	Male [✓ Female]
Email :	tentin@gmail.com
Contact Number :	081215612788
Address Line 1 :	New Beverlly Hills
Enrollment date	26 May 2025
GPA:	3.60
Course Code :	1

stu...	firs...	sur...	dat...	ge...	em...	co...	ad...	enr...	gpa	co...	im...
00...	Te...	An...	20...	Fe...	tes...	08...	nbh	20...	4.00	1	
00...	Q...	Ce...	20...	Fe...	qw...	08...	Ne...	20...	3.75	1	

F. CRUD System Overview

1. Major Management

Module Create (Add)

- Input fields: Major ID and Major Name.
- The **“Add”** button allows users to insert a new major record into the database.

Read (Search & Display)

- A search bar is available for finding records based on Major ID or Major Name.

- A table view at the bottom displays all current majors (showing ID and Name) for easy browsing and selection.

Update

- Users can select a record in the table; the data will populate the input fields above.
- Make edits (e.g., changing the Major Name).
- The **“Update”** button saves any changes made to the selected record.

Delete

- Select a record in the table.
- The **“Delete”** button removes the selected major from the database.

Clear

- The **“Clear”** button resets the input fields, preparing the form for new entries or searches.

2. **Student Manager**

Module Create (Add

Record)

- Input fields: Student ID, First Name, Surname, Date of Birth, Gender, Email, Contact Number, Address, Enrollment Date, GPA, Course Code.
- The **“Add Record”** button saves a new student record to the database.

Read (Search & Display)

- A search bar for searching by Student ID.
- A table to the side displays all student records, including their ID, Name, Contact Details, Enrollment, GPA, and Course Code.

Update

- Click a row in the table to display its data in the form fields.
- Edit the fields as needed (e.g., changing address or GPA).
- Click **“Update”** to save the updated information.

Delete

- Select a record from the table and press **“Delete”** to remove the student’s record permanently.

Clear

- The **“Clear”** button resets the input form for fresh data entry.

G. CONCLUSION

The Student Management System is a comprehensive desktop-based application designed to streamline and automate student and academic program data management. Developed using Java Swing, it features an intuitive user interface and robust functionalities that ensure
seamless data operations.

The system architecture is carefully structured in a layered approach:

- The **Presentation Layer** provides user-friendly forms and navigational interfaces.
- The **Business Logic Layer** manages the core operations and data validation.
- The **Data Layer** handles secure database connectivity and interaction with MySQL.