Student Management System

Contents

[**Introduction** 2](#_Toc152964694)

[**System Overview** 2](#_Toc152964695)

[**Project Structure** 3](#_Toc152964696)

[**Getting Started** 3](#_Toc152964697)

[**System Features** 4](#_Toc152964698)

[**Screenshots** 5](#_Toc152964699)

[**Conclusion** 7](#_Toc152964700)

[**Appendices** 7](#_Toc152964701)

# **Introduction**

The Student Management System is a comprehensive JavaFX application aimed at streamlining and enhancing the management of educational processes. It leverages the Model-View-Controller (MVC) architecture to maintain a clear separation of data handling, user interface, and control logic. This approach ensures a clean, maintainable, and scalable system.

# **System Overview**

This application is divided into three main components:

* **Models**: Handle all the data-related logic, including interactions with the database for student, course, and university information.
* **Controllers**: Act as intermediaries, connecting the Models and Views. They process user inputs and respond with the appropriate data.
* **Views**: Focused on the graphical user interface, presenting data to users and capturing their inputs. Views are built using FXML to create an intuitive and responsive layout.

The system supports functionalities for different user roles, notably students and administrators, with each having access to role-specific features.

# **Project Structure**

The project is systematically organized into distinct packages:

1. **src/controllers/**:
   * **LoginController.java**, **SignupController.java**: Handle user authentication processes.
   * **UniversitySelectionController.java**, **CourseSelectionController.java**, **StreamSelectionController.java**: Manage the selection process of universities, courses, and streams.
   * **ScholarshipDetailsView.fxml:** Manage the viewing of available Scholarships
   * Other controllers handling functionalities done by admin to add in Universities, Courses and Streams
2. **src/models/**:
   * **Student.java**, **University.java**, **Course.java**, **Stream.java**: Represent the various entities in the system.
   * **Dao.java**: Responsible for database operations.
   * **DBConnect.java**: Manages database connections.
3. **src/views/**:
   * FXML files like **UniversitySelectionView.fxml**, **CourseSelectionView.fxml**, **StreamSelectionView.fxml**, etc., define the UI layout for the system.

# 

# **Getting Started**

To run the application, start **Main.java**. Users will be presented with a login screen. Sample credentials are provided for different user roles.

* **Admin:**
  + Username: admin
  + Password: admin
* **Student:**
  + Username: student
  + Password: student

# **System Features**

* **User Registration and Authentication:** Secure sign-up and login processes.
* **University, Course, and Stream Selection:** Students can navigate through and select universities, courses, and streams.
* **Scholarship Information:** Access to details about various scholarships.
* **Responsive User Interface:** An intuitive and easy-to-use graphical interface.

# **Screenshots**

A screenshot of a computer

Description automatically generatedn

A screenshot of a computer

Description automatically generated

Successful login for an admin

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

Successful login for an student

A screenshot of a computer

Description automatically generated

All 10 records for Scholarships added to database.

A screenshot of a computer

Description automatically generated

A red and blue rectangular box

Description automatically generated with medium confidence

Update performed by the admin for the first record

A white rectangular object with a shadow

Description automatically generated

Deletion of the last record by the admin

A screenshot of a red and white screen

Description automatically generated

Remaining records in a columnar view

A screen shot of a computer

Description automatically generated

Ending user table from the database

# **Conclusion**

The Student Management System is a dynamic and user-friendly JavaFX application designed to manage educational processes efficiently. With its MVC architecture, the system offers clarity in organization and ease of scalability. It provides an effective platform for students and administrators to interact with academic and administrative data.

# **Appendices**

**Class Diagram**

A close-up of several boxes

Description automatically generated