**Student Management System**

**A Java Swing-based Desktop Application**

**1. Introduction**

The Student Management System is a graphical desktop application developed using Java Swing. Its primary goal is to simplify student record handling by offering an intuitive interface where users can register students, assign them to academic courses, input exam results, and generate customized certificates. The system follows object-oriented programming principles and demonstrates practical applications of Java's GUI components and event-driven logic. This project is especially useful for students, educators, and developers who want to understand how academic workflows can be simulated digitally.

**2. Purpose of the Project**

The project aims to create a lightweight yet functional application for managing academic records without relying on web-based systems or databases. It serves as a learning tool for Java developers who want to apply concepts such as encapsulation, modular design, and input validation in real-world scenarios. By using only local storage and Java standard libraries, it maintains simplicity while offering rich features through a friendly user interface.

**3. Functional Modules**

**3.1 Student Registration**

This module collects personal and academic information about the student. The user is prompted to enter the student's full name, roll number, age, and email address. The system uses basic validation to ensure that necessary fields are filled properly, and that roll numbers and ages are valid numeric entries.

**3.2 Course Selection**

Once a student is registered, a course can be selected by entering its name and duration in months. Courses are stored internally and can be reviewed or reused if needed. The course selection module is designed to prevent errors such as assigning a course before student registration.

**3.3 Exam Result Entry**

This module allows the user to enter marks for five fixed subjects. For each subject, the system ensures that the input is valid and numerical. Once all marks are recorded, it automatically calculates the total marks, average percentage, and assigns a grade based on a defined scale. Grades such as "Excellent," "Good," and "Needs Improvement" help summarize student performance clearly.

**3.4 Certificate Generation**

Based on the exam results, the application generates a digital certificate using a formatted HTML string rendered in a Swing dialog box. The certificate includes student details, course name, calculated percentage, grade, and the date of issue. It acts as a proof of academic completion and accomplishment.

**3.5 Student Information Viewer**

At any point, the user can retrieve and view the registered student’s personal details. This feature is useful for confirming entries or displaying summaries during the session.

**4. Technologies Used**

* **Programming Language:** Java
* **Libraries and Tools:** Java Swing, AWT
* **Design Concepts:** Object-Oriented Programming, Event-Driven Programming
* **UI Styling:** Nimbus Look and Feel, custom pastel themes using Color and Font classes
* **Validation:** Input error handling using try-catch blocks and dialog prompts

**5. Project Structure and File Overview**

* MainDashboard.java: Hosts the main window and connects all actions through button-based navigation
* Student.java: Stores student information and includes methods to display it
* Course.java: Manages course creation, duration, and enrollment tracking
* Exam.java: Handles marks input, computes total score, percentage, and assigns grades
* Certificate.java: Formats and displays the final certificate in a dialog window

**6. How to Run the Application**

1. Install Java JDK (version 8 or later).
2. Open the project in Eclipse or any Java-supported IDE.
3. Compile the files and run MainDashboard.java.
4. Use the GUI buttons to navigate between student registration, course selection, exam entry, and certificate generation.

**7. Conclusion**

This Student Management System successfully demonstrates how academic workflows can be represented programmatically using Java. Through modular architecture and a user-friendly interface, the system provides clear functionality without the complexity of backend integration or file handling. It is ideal for anyone aiming to strengthen their understanding of desktop-based application development in Java.

Let me know if you'd like me to generate this as a downloadable PDF or help you style it for printing or presentation.