**Student Management System (SMS) Project**

**Abstract**

The Student Management System (SMS) is a Java-based console application developed to efficiently manage student records. It offers a range of functionalities, including adding new student records, updating existing records, deleting student entries, and retrieving detailed student data. The system utilizes an ArrayList for data storage, ensuring dynamic and efficient data handling. Designed with a Model-View-Controller (MVC) architecture, the project ensures modularity, scalability, and a clear separation of concerns. This structured approach enhances maintainability and facilitates future upgrades. The SMS is ideal for educational institutions, providing a streamlined method for managing student information with a user-friendly console interface.

**Introduction**

In educational institutions, managing student data manually is inefficient and error-prone. This project aims to create a **simple yet effective student management system** that allows users to perform CRUD (Create, Read, Update, Delete) operations on student records. The system leverages Java's **Object-Oriented Programming (OOP)** principles and follows structured coding practices for scalability.

**Objectives**

* To develop a console-based student management system using Java.
* To implement **ArrayList** as a data storage mechanism.
* To provide CRUD functionalities for managing student records.
* To follow the **MVC architecture** for maintainability and modularity.

**Technologies Used**

* **Programming Language**: Java
* **Data Storage**: ArrayList
* **Design Pattern**: MVC (Model-View-Controller)
* **Development Tools**: JDK, IDE (Eclipse/IntelliJ IDEA/VS Code)

**System Architecture**

**1. Model (Student.java)**

This class represents the **Student entity**, containing attributes like ID, name, email, password, date of birth, gender, relationship status, and mobile number.

**2. Controller (StudentController.java)**

This class acts as the intermediary between the view and model. It contains business logic for storing and retrieving student data using an **ArrayList**.

**3. View (StudentView.java)**

This class provides the **user interface** to interact with the system. It accepts user input, processes it, and displays results accordingly.

**Functionalities**

**1. Create Student**

Users can input student details, which are stored in an **ArrayList**.

**2. Find Student by ID**

Users can retrieve a specific student record by entering the student ID.

**3. Update Student by ID**

Users can update an existing student's details by providing the student ID and new information.

**4. Delete Student by ID**

Users can remove a student record from the system.

**5. Find All Students**

Displays all student records stored in the **ArrayList**.

**6. Exit**

Terminates the program gracefully.

**Testing**

* The system was tested with different input cases to ensure robustness.
* The **CRUD operations** were verified by adding, updating, deleting, and retrieving student records.
* Input validations were conducted to prevent errors.

**Future Enhancements**

* Implement a **Graphical User Interface (GUI)** for better user experience.
* Use **Database (MySQL, PostgreSQL)** for permanent data storage.
* Introduce authentication and role-based access control.

**Conclusion**

The **Student Management System (SMS)** is an efficient tool for managing student records using **Java and ArrayList**. It follows an **MVC pattern**, ensuring **scalability and maintainability**. Future enhancements will include **database integration** and **GUI support** for an improved user experience.