**Student Management System Documentation**

**Overview**

The **Student Management System** is a Java-based application designed to help users manage student information. It allows users to add new students to a database and view a list of all students. The application provides a simple and intuitive user interface, built using Swing, and stores data persistently using SQLite.

**Features**

1. **Add Student**:
   * Users can input student details such as name, age, and grade.
   * The data is stored persistently in an SQLite database.
2. **View Students**:
   * Displays a list of all students in the database.
   * Includes details such as ID, name, age, and grade.
3. **Aesthetic UI**:
   * A visually pleasing user interface with customized colors, fonts, and layout.

**Technologies Used**

* **Java**: Core programming language for building the application.
* **Swing**: Java framework for creating the graphical user interface (GUI).
* **SQLite**: Lightweight database for storing student data persistently.
* **JDBC**: Java Database Connectivity for database interactions.

**Project Structure**

**Main.java**

* Entry point of the application.
* Initializes and launches the user interface.

**StudentManagementUI.java**

* Handles the graphical user interface (GUI) of the application.
* Provides functionality for adding and viewing students.
* Customizes the UI with colors, fonts, and layout.

**StudentService.java**

* Handles database operations.
* Responsible for creating the database, adding students, and retrieving student data.

**Database Structure**

The application uses an SQLite database with a single table named **students**. The table structure is as follows:

| **Column Name** | **Data Type** | **Description** |
| --- | --- | --- |
| id | INTEGER | Primary key, auto-incremented. |
| name | TEXT | Name of the student. |
| age | INTEGER | Age of the student. |
| grade | TEXT | Grade of the student. |

**How to Run**

1. **Prerequisites**:
   * Ensure you have Java Development Kit (JDK) installed.
   * SQLite database driver included in your project.
2. **Steps**:
   * Compile the code using a Java compiler:
   * javac Main.java StudentManagementUI.java StudentService.java
   * Run the application:
   * java Main
3. **Usage**:
   * Use the GUI to add new students by entering their name, age, and grade, then clicking **Add Student**.
   * View all students by clicking **View Students**.

**Aesthetic Design Details**

1. **Background Colors**:
   * Main panel: Light purple (#E6E6FA)
   * Buttons: Pink (#FFC0CB)
2. **Font Styles**:
   * Labels: Arial, Bold, 14pt
   * Buttons: Comic Sans MS, Bold, 14pt
3. **Alignment**:
   * Labels are centered for better readability.
   * Input fields and buttons are arranged in a grid layout with spacing.

**Code Highlights**

**Adding a Student**

addButton.addActionListener(new ActionListener() {

@Override

public void actionPerformed(ActionEvent e) {

String name = nameField.getText();

int age = Integer.parseInt(ageField.getText());

String grade = gradeField.getText();

studentService.addStudent(name, age, grade);

JOptionPane.showMessageDialog(frame, "🌟 Student added successfully! 🌟");

}

});

**Viewing Students**

viewButton.addActionListener(new ActionListener() {

@Override

public void actionPerformed(ActionEvent e) {

String students = studentService.getAllStudents();

JTextArea textArea = new JTextArea(students);

textArea.setFont(new Font("Monospaced", Font.PLAIN, 12));

textArea.setEditable(false);

JScrollPane scrollPane = new JScrollPane(textArea);

JOptionPane.showMessageDialog(frame, scrollPane, "Student List", JOptionPane.INFORMATION\_MESSAGE);

}

});

**Database Initialization**

public StudentService() {

try (Connection conn = DriverManager.getConnection(DB\_URL)) {

String createTableQuery = "CREATE TABLE IF NOT EXISTS students (" +

"id INTEGER PRIMARY KEY AUTOINCREMENT, " +

"name TEXT NOT NULL, " +

"age INTEGER NOT NULL, " +

"grade TEXT NOT NULL)";

Statement stmt = conn.createStatement();

stmt.execute(createTableQuery);

} catch (SQLException e) {

e.printStackTrace();

}

}

**Conclusion**

The Student Management System is a simple yet powerful tool for managing student information. With its user-friendly interface and reliable database backend, it serves as an excellent starting point for further enhancements and learning in Java application development.