**Student Management System**

This project is a graphical user interface (GUI) application built using Python's Tkinter library. The system is designed to manage student records, including adding, removing, searching, and displaying student details. It also includes a basic user authentication system.

**Features**

**Student Management**

* **Add Student**: Add a new student with details like ID, name, age, classification, and major.
* **Remove Student**: Remove an existing student by providing their name and ID.
* **Search Student**: Search for a student by their first and/or last name.
* **Display Student Count**: View the total number of students and the number of students by major.

**User Authentication**

* **Login**: Authenticate users with a username and password.
* **Sign Up**: Create new user accounts with password validation.
* **Logout**: Logout from the system and return to the login screen.

**Additional Features**

* **Dynamic Student Types**: Students are categorized by major (e.g., Computer Science, Business, Arts, Engineering), with specialized behavior based on their type.
* **Error Handling**: Validation checks ensure data integrity and user-friendly error messages.

**Technologies Used**

* **Python**: Programming language used to implement the application.
* **Tkinter**: Library used for creating the graphical user interface.
* **ttk**: Used for modern widgets (e.g., dropdown menus).

**How to Run the Application**

**Prerequisites**

Ensure that Python 3 is installed on your system. Tkinter is included with most Python installations by default.

**Steps**

1. Clone the repository or download the project files.
2. Navigate to the project directory in your terminal or IDE.
3. Run the following command to start the application:

python <filename>.py

Replace <filename> with the name of the Python file containing the main() function.

1. The application window will open. Use the login screen to authenticate or create a new user account.

**File Structure**

* **Main Application File**: Contains the LoginSystem class and the entry point for the program.
* **Student Classes**: Define specialized behavior for students based on their major (e.g., ComputerScienceStudent, BusinessStudent).
* **Data Persistence**: Includes methods for saving and loading user and student data.

**Usage**

**Adding a student**

1. Click the "Add Student" button.
2. Fill in the required fields: ID, first name, last name, age, classification, and major.
3. Click "Save Student" to add the student.

**Searching for a Student**

1. Click the "Search Student" button.
2. Enter the first and/or last name of the student.
3. View the search results. If multiple matches are found, select a student from the list to view their details.

**Displaying Student Count**

1. Click the "Display Student Count" button.
2. View the total number of students and the count of students by major.

**Removing a student**

1. Click the "Remove Student" button.
2. Enter the student's name and ID.
3. Confirm the removal.

**Managing Users**

* Use the "Sign Up" button to create a new user account.
* Login with a valid username and password.
* Logout by clicking the "Logout" button.

**Future Enhancements**

* Add data export functionality (e.g., CSV or Excel).
* Implement role-based access control.
* Enhance the UI with themes or a modern library like PyQt.
* Add analytics (e.g., charts for student distribution).
* Improve search performance with optimized data structures.

**Contribution**

Feel free to contribute to this project by submitting issues or pull requests. Ensure that your code follows Python's PEP 8 guidelines and is well-documented.

**Contact**

For any inquiries or feedback, please contact [boatengkwaku1965@gmail.com](mailto:boatengkwaku1965@gmail.com)