

STUDENT INFORMATION SYSTEM

Team members:

Kwaku Boateng

G00451270

Obed Osei Kwame Nti

G00451618

PROJECT SUMMARY

This project is designed to create a Python-based Student Information System for efficient management of student data. The system implements object-oriented programming using classes and subclasses representing various categories of students. There is a superclass "Student" that contains the common attributes, such as name and ID, and has a class variable to keep track of the total number of students. The subclasses, like "Computer Science Student," "Engineering Student," inherit from the superclass and further define additional major-specific attributes and methods. For the persistence of data, the system writes down the student's information in a file comprising their names, ID, and major. Design a user-friendly GUI interface with login functionality, allowing users to create new accounts and access the system. Login credentials should be securely stored in a file. By combining object-oriented programming, file I/O, and GUI design, this project saves immense time in managing the student information effectively.

STUDENT INFORMATION SYSTEM

PROCESS OVERVIEW

Algorithm

Step 1: Start

1. Display the **Login Screen**.

Step 2: User Login

2. Prompt the user to select one of the following:
 - **New User?**
 - **Existing User?**

Step 3: New User Registration

3. If the user selects **New User**:
 - Prompt for username and password.
 - Save the credentials to a **Login File**.
 - Redirect the user back to the **Login Screen**.

Step 4: Existing User Login

4. If the user selects **Existing User**:
 - Validate the entered username and password against the **Login File**.
 - If credentials are **invalid**, display an error and return to **Step 2**.
 - If credentials are **valid**, proceed to the **Main Menu**.

Step 5: Main Menu

5. Display the **Main Menu** with the following options:
 - **Add Student**
 - **View Students**
 - **Save Data**
 - **Logout**

Step 6: Add Student

6. If the user selects **Add Student**:
 - Prompt the user to **Select a Major**:

STUDENT INFORMATION SYSTEM

- **Computer Science**
- **Engineering**
- **Business**
- **Arts**
- Input the student's details (e.g., Name, ID).
- Create a **Student Object** based on the selected major.
- Add the object to the respective **List of Students**.
- Return to the **Main Menu**.

Step 7: View Students

7. If the user selects **View Students**:
 - Display the details of all students stored in the **List of Students**.
 - Return to the **Main Menu**.

Step 8: Save Data

8. If the user selects **Save Data**:
 - Write the student details (e.g., Name, ID, Major) to a **Student File**.
 - Return to the **Main Menu**.

Step 9: Logout

9. If the user selects **Logout**:
 - Exit the system.

Step 10: End

10. Terminate the program.

STUDENT INFORMATION SYSTEM

FLOW DIAGRAM

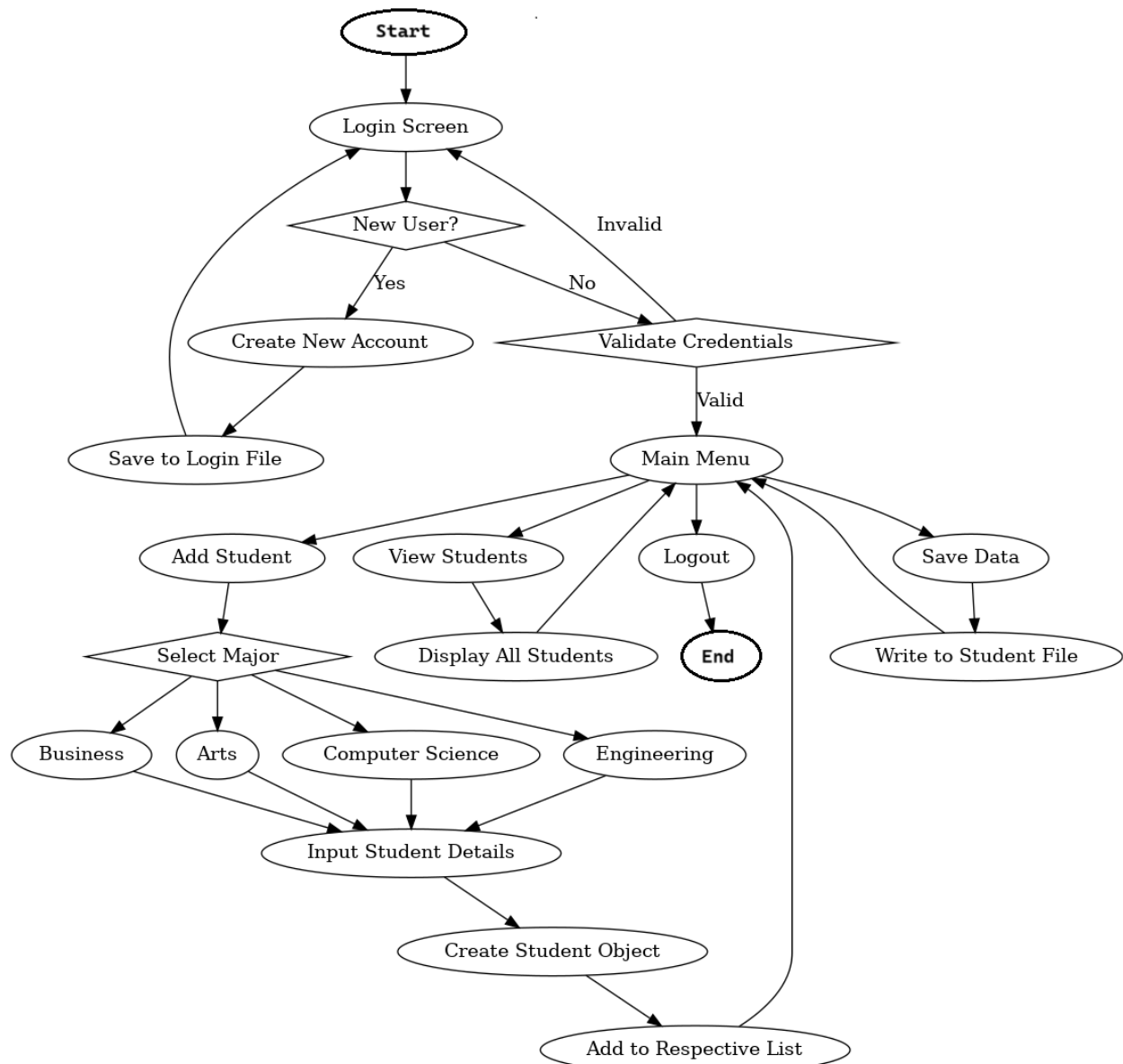


Figure 1. Flow diagram