BIRZEIT UNIVERSITY

Faculty of Engineering and Information Technology

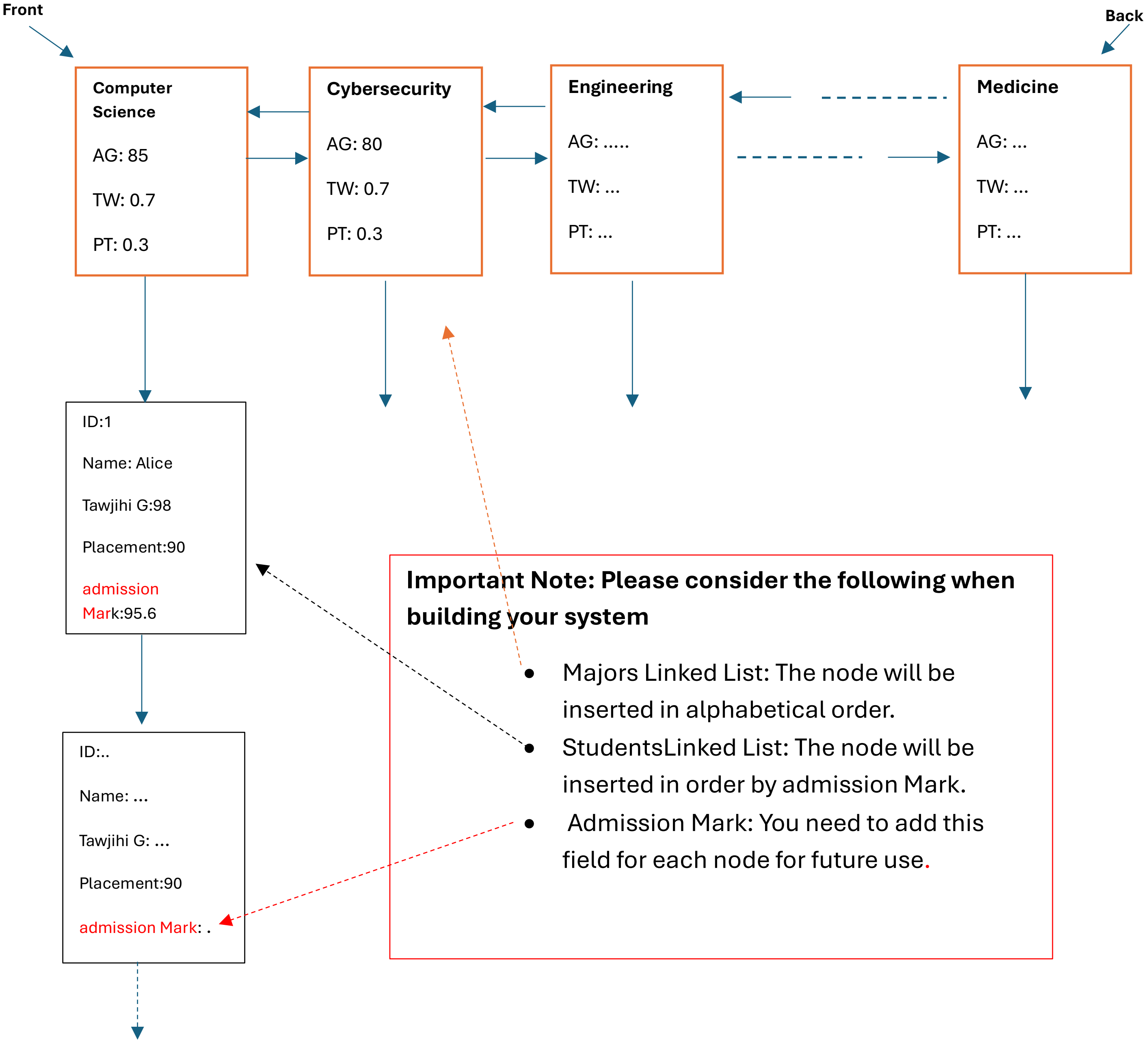
Computer Science Department

COMP242 Project #2

Project: Student Admission Management System

Overview

As a developer in the admissions department, you are responsible for creating a comprehensive Student Admission Management System. This system will utilize Linked Lists to efficiently manage both student and major data, calculate admission marks, and assign students to their respective majors based on predefined criteria.



Important Note: Please consider the following when building your system

- Majors Linked List: The node will be inserted in alphabetical order.
- StudentsLinked List: The node will be inserted in order by admission Mark.
- Admission Mark: You need to add this field for each node for future use.

Key Features

1. **File Management**:
   - The program will read two input files:
     - **File 1**: Contains student data, including `unique student IDs, names, Tawjihi grades`, and `Placement Test, Chosen Major`.
     - **File 2**: Contains the admission criteria, which includes `weightings for Tawjihi` and `Placement Test grades`, as well as the `acceptance grade` for each major.
   - Implement a file chooser to allow users to easily select and read these input files.
   - The program will handle various scenarios, including file read errors and data format inconsistencies, ensuring robust exception handling throughout.

2. **Student Data Management**:
   - The system will support the following operations on student records:
     - **Insert**: Add new students to the linked list, capturing their unique student ID, name, Tawjihi grade, and Placement Test grade.
     - **Delete**: Remove student records based on a unique student ID, with confirmation messages to prevent accidental deletions.
     - **Update**: Modify existing student information, including Tawjihi grades, Placement Test, name, with validation checks to ensure data integrity.
     - **Search**: Efficiently locate student records by ID, providing users with immediate feedback on whether the student exists in the system.

3. **Major Data Management**:
   - The system will support operations on major records:
     - **Insert**: Add new majors to the linked list, capturing the major name and its corresponding acceptance grade, as well as individual weightings for Tawjihi and Placement Test grades.

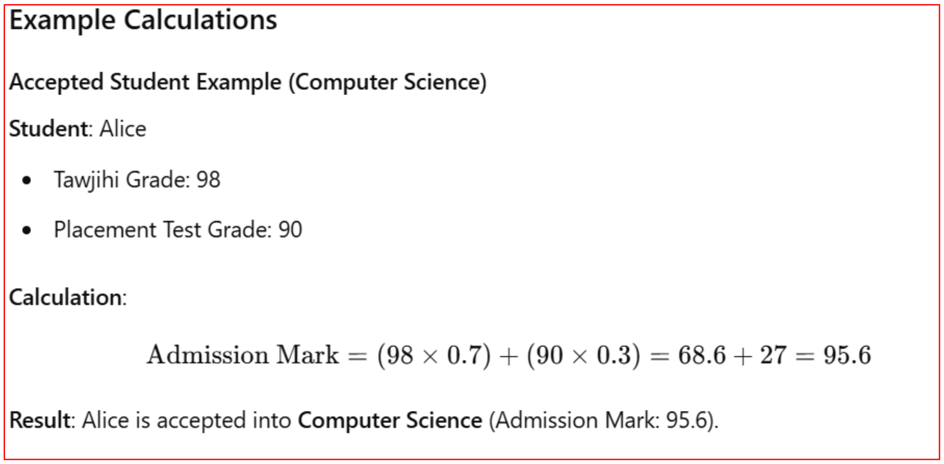
        - **Delete**: Remove major records based on the major name, with confirmation messages to prevent accidental deletions.
        - **Update**: Modify existing major information, including the name, acceptance grade, and weightings for Tawjihi and Placement Test grades, with validation checks to ensure accuracy.
        - **Search**: Efficiently locate major records by name, providing users with immediate feedback on whether the major exists in the system.

4. **Admission Criteria**:
    - The program will read from an input file detailing admission criteria, which includes:

      {Tawjihi Weight, Placement Test Weight, Acceptance Grade}

    - Admission marks will be calculated using the following formula:

      Admission Mark=(Tawjihi Grade×Tawjihi Weight)+(Placement Test Grade×Placement Test Weight)



Example Calculations

Accepted Student Example (Computer Science)

Student: Alice

- Tawjihi Grade: 98
- Placement Test Grade: 90

Calculation:

$$\text{Admission Mark} = (98 \times 0.7) + (90 \times 0.3) = 68.6 + 27 = 95.6$$

Result: Alice is accepted into **Computer Science** (Admission Mark: 95.6).

    - The system will assign students to majors based on their calculated admission marks:
    - Students not meeting the requirements for any major will be marked as "Not Accepted".

5. **Major Suggestion Functionality**: (You may use menu with two items to handle this functionality)

    - **Initial Major Suggestion**: When a student inputs their details, the system will suggest a suitable major based on their calculated admission mark.

        Example: The system will provide a list of majors that the student can choose from. The student will select one major from the list and then press the 'Confirm' button to register.

    - **Alternative Major Recommendation**: If a *student is not accepted into* their first-choice major (chosen major), the system will provide recommendations for other suitable majors based on their admission marks.

        Example: The system will provide a list of alternative majors that the student can choose from. The student will select one major from the list and then press the 'Confirm' button to register.

6. **Statistics and Reporting**: (You may use menus to handle this functionality)

    - The system will track and report various statistics, including:

        - Total number of students accepted into each major.

        - Total number of students rejected for each major, providing insights into the reasons for rejection (e.g., Tawjihi grade too low, Placement Test grade too low).

        - Overall Acceptance Rates for All Majors:

            - Total number of accepted students:

$$\text{Total Accepted} = \sum(\text{Accepted Students for Each Major})$$

            - Total number of students evaluated:

$$\text{Total Evaluated} = \text{Total Accepted} + \text{Total Rejected}$$

            - Acceptance Rate Formula:

$$\text{Acceptance Rate} = \left(\frac{\text{Total Accepted}}{\text{Total Evaluated}}\right) \times 100\%$$

- Total number of students accepted across all majors.
- Total number of students rejected across all majors.
- Top N accepted students into specific major (without duplicates).

  Example: Select a major from a list, enter the number of students to display, and click "Get button" to see the top N accepted students.

These statistics will be displayed in a user-friendly manner through the GUI, allowing administrators to easily access and analyze the admission data.

7. **User Interface**:
   - The graphical user interface (GUI) will enhance user interaction with features such as:
     - A table view to display student and major data clearly.
     - Navigation buttons (Next and Previous) for easy record browsing.
     - Confirmation messages for critical actions, such as deletions and updates, to enhance usability.
     - A statistics section (see point 6 above)

8. **Data Management**
   - Consider the following:
     - The program shall be able to display the statistics report on the user interface or save it to a file.
     - The program shall be able to save the updated lists back to the files.

9. **Performance Considerations**:
   - The implementation will focus on efficiency, with consideration of time complexity in operations such as insertion, deletion, and searching within the linked lists. This will ensure the system remains responsive, even with a large volume of students and major records.

Sample Input Files

File 1: Students Data (students.txt) (this is an example; you need to extend this file)

Student ID	Name	Tawjihi Grade	Placement Test Grade	Chosen Major
1	Alice	98	90	Computer Science
2	Bob	85	80	Cybersecurity
…	…	…	…	…

File 2: Acceptance Criteria (criteria.txt)

Number of Majors: 4 (this is an example; you need to extend this file)

Major	Acceptance grade	Tawjihi Weight	Placement Test Weight
Medicine	97	0.8	0.2
Engineering	90	0.6	0.4
Computer Science	85	0.7	0.3
Cybersecurity	80	0.7	0.3
…	…	…	…

Please note the Followings:

I. Your application should have all functionalities working properly.

II. There must be adequate documentation and comments in the code (e.g., functions, loops, etc.).

III. Your code should follow coding conventions (e.g., spacing, indentation, etc.) and guidelines (**Remember COMP2311**).

IV. This is an **individual Project**. Disciplinary action will be taken against those who **cheat**. Additionally, the use of **AI tools** for generating solutions or **copying from websites** is strictly prohibited. Students found in violation of these policies will face severe consequences. It is crucial to ensure that all work submitted is **your own** and adheres to the **guidelines provided for this project**.

V. Please submit your Java files (java) and corresponding test text files (txt) via the ITC by Tuesday, 29/ 10/2024, at 11:00 PM. Late submissions will not be accepted under any circumstances.

All the Best :)