

**North South University  
Dept. of Electrical & Computer Engineering**

*'Data Structures and Algorithms'*

**A Report on**

**Application with Screenshots**

Submitted To

Dr. Mohammad Rezwanul Huq (MRH1)

Course Instructor: CSE225.08

**‘Group - 01’**

Submitted By

| Group Members (04) | ID |
| --- | --- |
| 01. Mushfiqur Rahman  02. Koushik Ahamed Tonmoy  03. Safayet Ullah Neyam  04. Tanjir Ahmed Nadim | 2121535642  2111023642  2031181642  2121782642 |

Report Submission: September 10, 2022

**A Report On Application with Screenshots**

**Project Title**

Student Information System

**Brief Description of the Project**

The Project ‘Student Information System’ is A One-Ended (Admin/User Restricted) console-based application to perform operations on information regarding students. The main purpose of this project is to store and manipulate students' information widely. This console-based developed application takes data/information from a defined text (.txt) file, manipulates those, and exports the manipulated data to the same defined text or any other file as required (also in PDFs). While manipulating the data/information, it takes required data that needs to be updated, inserted, deleted, or searched from the user end through the console screen. Therefore, it has the ability to **view** the information that has been stored in a file, **insert** new student information, **update** each and every student’s data instances, **delete** required students from the database considering any data occurrence, **search**/**find** data with any possible instances, ascending or descending **sort** by student’s individual instances. It produces PDF or text reports on data that are recently viewed and searched. There is also a short probable feature of exporting PDF reports on the manipulated data/information that are inserted, updated, and deleted recently before the program ends. Importantly, the program does the operations using the Linked List.

**Structure/Class Definition of the Node**

The structure definition of the node for each student is given below.

*struct node {*

*int id, semester;*

*string name, dob, department, email, contact, address, bloodGroup;*

*bool isPaymentDone, isScholarshipCandidate, isGraduated;*

*double completedCredits, cgpa;*

*};*

**Application of The Project (with Screenshots)**

The required description of the program's functionalities is given below.

**Main Menu**

The Main Menu has the Functions to View, Insert, Update, Delete, Search, Sort and Generate Report.

1. Show Student List

2. Add Student Information

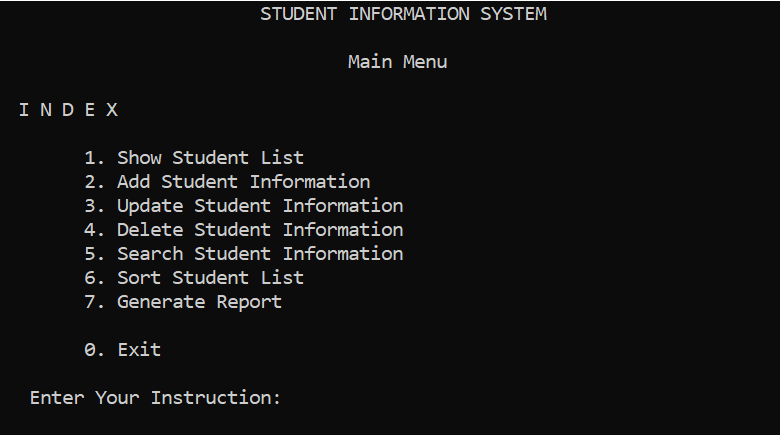
3. Update Student Information

4. Delete Student Information

5. Search Student Information

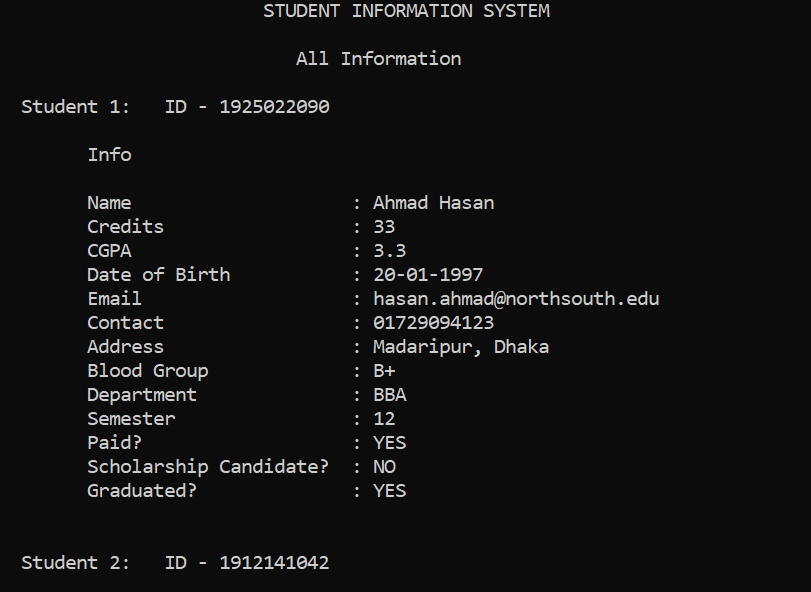
6. Sort Student List

7. Manipulate Report



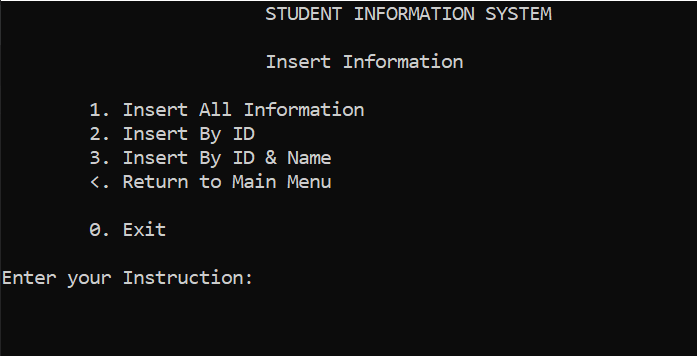
* **Show Student List**

This function has no sub-category to get any option. It directly shows all the information by traversing the whole linked list.



* **Add Student Information**

This function inserts student information as a new student structure by taking All the 14 instances, by taking ID, and by Taking both ID and Name.



* **Insert By All Information**

This sub-menu for adding student information (new student structure) takes all the 14 instances from the user through the console.

* **Insert By ID**

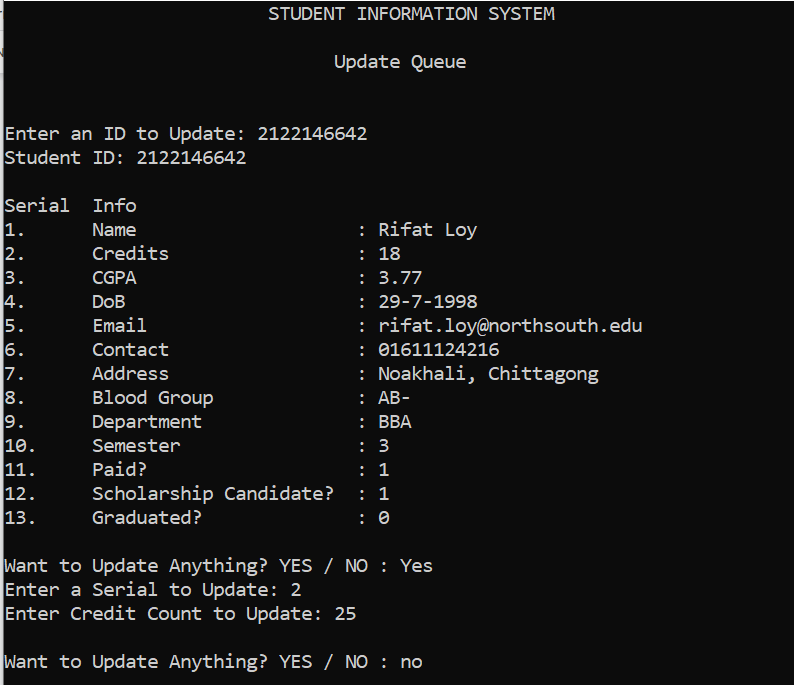
This sub-menu for adding student information (new student structure) takes just the ID from the user through the console.

* **Insert By ID & Name**

This sub-menu for adding student information (new student structure) takes just the ID and Name from the user through the console.

* **Update Student Information**

This function updates existing student information by passing specific parameters and/or by passing every instance.



* **Update By ID**

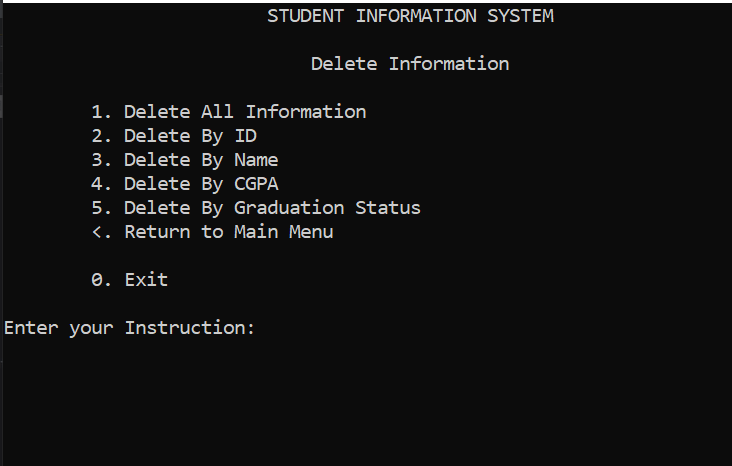
This sub-menu for updating student information from the existing student list takes the ID from the user through the console. It then passes the ID and gives the user to choose whatever data s/he wants to update.

* **Update Incomplete Student List**

This sub-menu for updating student information from the existing student list iterates/traverses the whole linked list and checks if there is any incomplete data or blocks/instances that is not given/set yet.

* **Delete Student Information**

This function deletes student information based on some parameter or some instruction.



* **Delete Whole List**

This sub-menu for deleting student information from the existing student list deletes all the information from the whole linked list.

* **Delete By ID**

This sub-menu for deleting student information from the existing student list deletes a single entry of a particular/unique ID.

* **Delete By Name**

This sub-menu for deleting student information from the existing student list deletes a single entry of a particular/unique student.

* **Delete By CGPA**

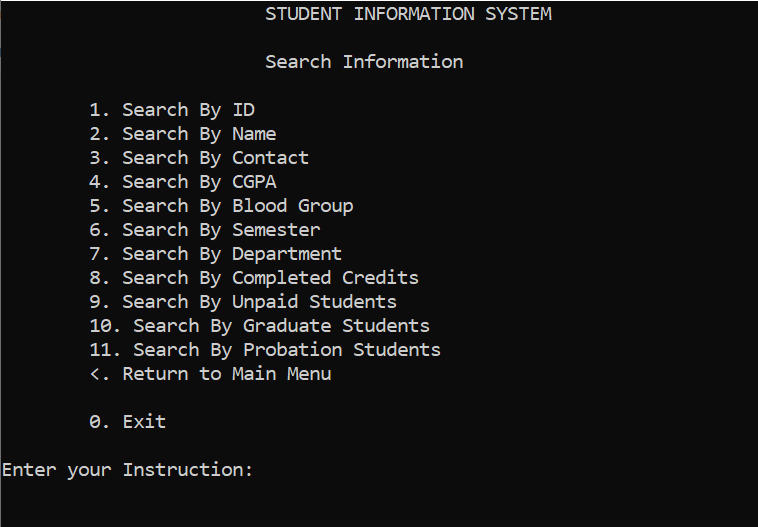
This sub-menu for deleting student information from the existing student list deletes multiple entries of CGPA of 05 functionalities. Those are equal ‘=’, greater than ‘>’, less than ‘<’, greater than or equal ‘>=’, less than or equal ‘<=’.

* **Delete By Graduation Status**

This sub-menu for updating student information from the existing student list deletes multiple entries of students who have graduated.

* **Search Student Information**

This function searches student information based on some parameter or some instruction.



* **Search By ID**

This sub-menu for searching student information from the existing student list searches for a particular student by its ID by traversing the whole linked list.

* **Search By Name**

This sub-menu for searching student information from the existing student list searches for a particular student by its Name by traversing the whole linked list.

* **Search By Contact**

This sub-menu for searching student information from the existing student list searches for a particular student by its Contact/Phone Number by traversing the whole linked list.

* **Search By CGPA**

This sub-menu for searching student information from the existing student list searches multiple students by its CGPA by traversing the whole linked list. There are some instructions to search. Those are equal ‘=’, greater than ‘>’, less than ‘<’, greater than or equal ‘>=’, less than or equal ‘<=’.

* **Search By Blood Group**

This sub-menu for searching student information from the existing student list searches multiple students by its Blood Group by traversing the whole linked list.

* **Search By Semester**

This sub-menu for searching student information from the existing student list searches multiple students by its Semester by traversing the whole linked list.

* **Search By Department**

This sub-menu for searching student information from the existing student list searches multiple students by its Department by traversing the whole linked list.

* **Search By Completed Credits**

This sub-menu for searching student information from the existing student list searches multiple students by its Blood Group by traversing the whole linked list.

* **Search By Unpaid Students**

This sub-menu for searching student information from the existing student list searches multiple students by their status of payment by traversing the whole linked list.

* **Search By Graduation Student**

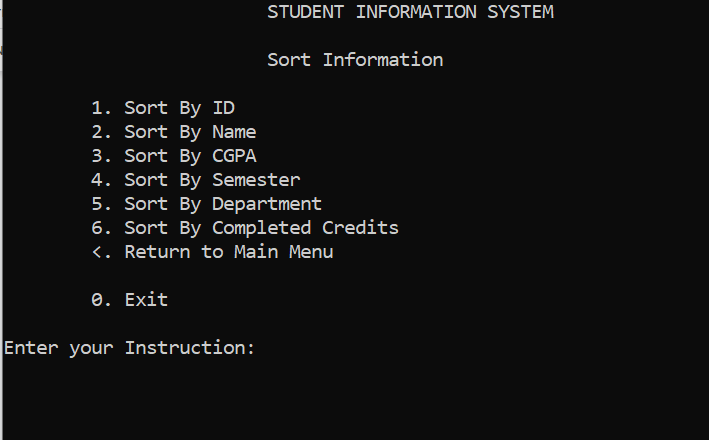
This sub-menu for searching student information from the existing student list searches multiple students by their status of graduation by traversing the whole linked list.

* **Search By Probation Students**

This sub-menu for searching student information from the existing student list searches multiple students by their CGPA status of CGPA by traversing the whole linked list.

* **Sort Student Information**

This function sorts student information based on some instructions in ascending order.



* **Sort By ID**

This sub-menu for sorting student information from the existing student list sorts or making orders based on the ID.

* **Sort By Name**

This sub-menu for sorting student information from the existing student list sorts or making orders based on the Name.

* **Sort By Semester**

This sub-menu for sorting student information from the existing student list sorts or making orders based on the Semester.

* **Sort By Department**

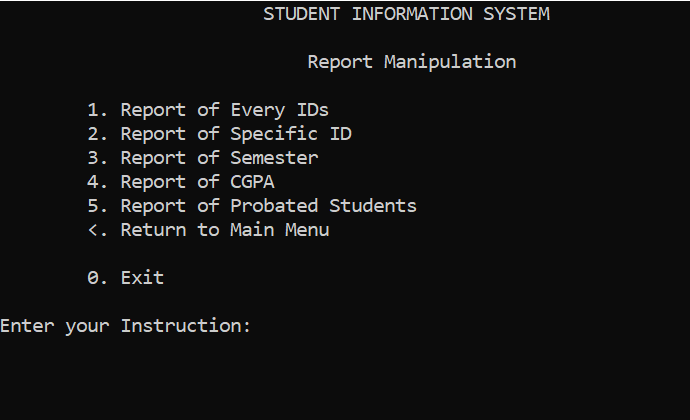
This sub-menu for sorting student information from the existing student list sorts or making orders based on the Department.

* **Search By Completed Credits**

This sub-menu for sorting student information from the existing student list sorts or making orders based on the Completed Credits.

* **Report Generation**

This function generates text (.txt) reports based on user preference. There are some sub-menus to do the stuff.



* **Report By ID**

This sub-menu for generating reports from the existing student list generates reports for a specific ID.

* **Report By Every ID**

This sub-menu for generating reports from the existing student list generates reports for all the students from the whole linked list.

* **Report By Semester**

This sub-menu for generating reports from the existing student list generates reports for all the students from the whole linked list based on a specific Semester.

* **Report By CGPA**

This sub-menu for generating reports from the existing student list generates reports for all the students from the whole linked list based on a required cgpa and a direction of 05 instructions. Those are equal ‘=’, greater than ‘>’, less than ‘<’, greater than or equal ‘>=’, less than or equal ‘<=’.

* **Report By Probation Status**

This sub-menu for generating reports from the existing student list generates reports for all the students from the whole linked list based on a cgpa lower than 2.0.

**Conclusion**

The Student Information System helps the user to access and view information more widely in an ordered manner. This program handles information to be removed and updated. It also helps by sorting entities as per the user's instruction and by showing calculated information to avoid probable issues and data collision. The feature of searching with exporting reports helps the user to collect the required information. It is believed that the project has the capability of helping the admin or the user to perform manipulation of data most conveniently.