A project report on

# Students Information Management System

Submitted by

**Md. Sanaullah Islam Shovon**
ID: 0242310005101082

**Md Abdur Rahman**
ID: 0242310005101413

**Md. Tanvir Hassan**
ID: 0242310005101461

**Ummay Jubaiya Moushi**
ID: 0242310005101887

**Kazi Amir Hamza**
ID: 0242310005101895

Supervised By

**Mr. Md. Aynul Hasan Nahid**
Lecturer
Department of CSE

**DAFFODIL INTERNATIONAL UNIVERSITY**
DHAKA, BANGLADESH
December 2023

# Table of Contents

# CHAPTER 1

# INTRODUCTION

## 1.1 Introduction

In this ever-evolving era of technology, educational institutions need efficient management of student data, which is paramount for ensuring streamlined operations and facilitating informed decision-making.

Implementing a robust Students Information Management System (SIMS) becomes crucial in this context. This project delves into designing and developing a data structure-based Student Information Management System, aiming to address the complexities of organizing and managing student information effectively.

## 1.2 Motivation

The design and implementation of our Students Information Management System (SIMS) are driven by a deep understanding of the complex challenges confronting educational institutions in today's dynamic landscape. The presence of operational inefficiencies, redundant data, and the potential for errors in handling varied student records underscores the necessity for a comprehensive solution. Our motivation arises from a dedicated commitment to confront these challenges proactively, aiming to establish a centralized platform that simplifies the intricacies of data management. The escalating volume and diversity of student data underscore the critical need for upholding data accuracy and integrity, compelling us to develop a system that guarantees reliability in both academic and administrative processes.

## 1.3 Objective

The goals of our Students Information Management System (SIMS) project are diverse, focusing on tackling crucial challenges and optimizing the operational effectiveness of educational institutions. Our foremost aim is to create a centralized hub for student data, offering a holistic remedy to the fragmentation and scattering of information typically encountered in conventional record-keeping systems. This initiative is geared towards eradicating redundancy in data, reducing errors, and establishing a singular, authoritative source for all student-related information.

## 1.3 Expected Outcome

The anticipated successful execution of our Students Information Management System (SIMS) project is poised to deliver numerous impactful results that will markedly improve the efficiency and functionality of educational institutions. Through the integration of functionalities like record addition, deletion, updating, and search capabilities, our objective is to establish a centralized and efficient method for managing student data. This streamlined approach guarantees that pertinent information is consolidated in a single accessible location, mitigating fragmentation and eradicating the inefficiencies linked with dispersed data.

# CHAPTER 2

# BACKGROUND

## 2.1 Related Works

Projects related to the student record system, including Student Records, Student Management, and Student Information Keeping, form an integral part of our initiatives. These projects are designed to optimize the organization and accessibility of student data, providing comprehensive solutions for efficient management and seamless retrieval of essential information in educational settings. Each project plays a crucial role in advancing the overall functionality and effectiveness of student data systems within institutions.

## 2.2 Comparative Studies

Gathering information poses a significant challenge, and this project addresses it by providing educational institutions with a seamless means of storing student information. Student Information Management Systems evaluate factors like user experience, adding new student data, integration capabilities, mobile compatibility, customization options, scalability, analytics, security measures, and cost-effectiveness. The objective is to pinpoint a holistic solution that caters to the unique requirements of educational institutions.

## 2.3 Challenges

The development of our Students Information Management System (SIMS) project comes with several expected challenges. Achieving seamless integration with existing systems and databases is a complex task, demanding careful coordination to ensure compatibility. The ever-changing and dynamic nature of student data presents a challenge in maintaining flexibility within the SIMS to adapt to evolving requirements. Security and privacy concerns add another layer of complexity, necessitating robust measures to safeguard sensitive student information. Striking a delicate balance between accessibility and stringent data protection protocols is crucial. User adoption and training emerge as pivotal challenges, underscoring the need for comprehensive programs to ensure efficient utilization by administrators.

# CHAPTER 3

# METHODOLOGY

## Linked List:

- The code defines linked list structures for Adding records of students, Deleting records, Update and Display records.
- Each structure has fields to store relevant information, such as ID, Name, E-mail, and Phone number.
- Linked lists organize and manage multiple instances of these structures, providing dynamic memory allocation.

## Function:

- The code defines several functions to perform specific tasks, such as adding new records, displaying records, and updating (addNew(), updateStud(), display(), deleteRandom(), deleteAll(), searchByID(), restore(), etc.
- The functions are organized based on their functionality, providing a modular structure to the code.

## Structure:

- Contain one structure for student information.
- These structures help organize and manage information in a structured and logical manner.

## STL:

- C++ STL property Map is used to store particular students enrolled course information. Where course code is the key and course name is the Value.

  **map < course code, course name > courses**

## Console:

- The console is utilized for user interaction, displaying menus and prompting users for input.
- Functions like landing(), returnLanding(), menu(), slowTxt(), etc are responsible for the Command Line User Interface (CLI)).

**File:**

- The code includes the header file “fstream” to handle the file.
- files save Student's information.
- Files save information only when the display() function is called.

# CHAPTER 4

## DESIGN SPECIFICATION

Here are the console screenshots captured after executing the code.

### Main Menu

```
=============================================

    Student Information Management System

=============================================

                  Main Menu

[1] Add a new Student.
[2] Search a Student by ID.
[3] View all Student's Info.
[4] Update any Students Info.
[5] Delete any Student's Info.
[6] Delete All Student's Info.
[7] Restore deleted Info.
[9] Return to Home.
[0] Exit The program.
=============================================
Enter your Choise:
```

### Adding Student

```
=============================================

    Student Information Management System

=============================================


You wanted to add a new student.
Please enter his/her detailed information

Student ID: 1461
Student Name: Md Tanvir Hasan
The Phone Number: 01724757410
E-mail Address: hasan23105101461@gmail.com
Enter The Number of Courses: 1
Course-1:
Course Code: CSE-124
Course Name: Ds lab

Student added successfully!

To return Home[H]
To return to Main Menu[M]
To Close the Programme[0]
Enter your choice:
```

## Viewing Student

```
=============================================

    Student Information Management System

=============================================


Viewing All the Student Information:

                Student-01
=============================================
Student ID: 1461
Name: Md Tanvir Hasan
Phone: 01724757410
E-mail: hasan23105101461@gmail.com
Enrolled courses: 1
        Course-01:
                Course Name: Ds lab
                Course Code: CSE-124
=============================================

To return Home[H]
To return to Main Menu[M]
To Close the Programme[0]
Enter your choice:
```

## Search Student by ID Number

```
=============================================

    Student Information Management System

=============================================


To search a student, please Enter following
information.

Stident ID: 1461
Name: Md Tanvir Hasan
Phone: 01724757410
E-mail: hasan23105101461@gmail.com
Enrolled courses: 1
        Course-01:
                Course Name: Ds lab
                Course Code: CSE-124

To return Home[H]
To return to Main Menu[M]
To Close the Programme[0]
Enter your choice:
```

## Delete Student

```
=============================================

    Student Information Management System

=============================================


You are about to delete an students
information. Please enter his/her ID.


Enter Student ID: 1461

Do you really want to delete 1461's info?

[1] Confirm Deletion.
[0] Cancel Deletion.
Your choice: 1

Information Moved to recycle bin.

To return Home[H]
To return to Main Menu[M]
To Close the Programme[0]
Enter your choice:
```

## Recycle Bin

```
=============================================

    Student Information Management System

=============================================

                    Home

[1] Main Menu
[2] Open recycle bin.
[3] Meet The Developer's
[0] Exit the Program.
=============================================
Enter your Choise: 2

Recycle Bin: 1461

To return Home[H]
To return to Main Menu[M]
To Close the Programme[0]
Enter your choice:
```

# CHAPTER 5

# IMPLEMENTATION AND TESTING

Here are some screenshots of the source code.

main.cpp - Code::Blocks 20.03

File Edit View Search Project Build Debug Fortran wxSmith Tools Tools+ Plugins DoxyBlocks Settings Help

<global>

Start here | main.cpp

```cpp
#include <bits/stdc++.h>
#include <windows.h>
#include<fstream>

using namespace std;

//Number of element that recycle bin can hold.
#define bin_size 10

//Structure of Students
typedef struct students{
    string ID, name, email, phone;
    int courseNum;
    map<string,string> course; //to store all his courses(Course code and name)
    struct students *next;
}students;

//List start
students *start = NULL;

students *queStart = NULL, *queEnd = NULL; //stores deleted students information to restore
int queEle = 0;

//Function Prototypes
void landing();
void menu();
void display();
void exitProgram();
void addNew();
bool exists(string SID, string type, string ID);
void SearchByID();
void returnlanding();
void slowTxt(string s);
```

G:\Data Structure\Student Info Management System\Student Info Manageme... C/C++ Windows (CR+LF) WINDOWS-1252 Line 1, Col 1, Pos 0 Insert Read/Write default

main.cpp - Code::Blocks 20.03

File Edit View Search Project Build Debug Fortran wxSmith Tools Tools+ Plugins DoxyBlocks Settings Help

<global>

Start here | main.cpp

```cpp

signed main(){
    while(true){
        system("CLS"); //Clear's output screen
        landing();
        int ope;
        cin>>ope;
        switch(ope){
            case 0:
                exitProgram();
                break;
            case 1:
                menu();
                break;

            case 2:
                queDisplay();
                break;

            case 3:
                viewDevelopers();
                break;

            default:
                string invalid = "\n\tInvalid Choice...";
                slowTxt(invalid);
                Sleep(300);
                break;
        }
    }
}

void landing(){
```

G:\Data Structure\Student Info Management System\Student Info Manageme... C/C++ Windows (CR+LF) WINDOWS-1252 Line 1, Col 1, Pos 0 Insert Read/Write default

Some more screenshots of the source code.

main.cpp - Code::Blocks 20.03

File Edit View Search Project Build Debug Fortran wxSmith Tools Tools+ Plugins DoxyBlocks Settings Help

<global>

Start here | main.cpp

```cpp
void landing(){
    system("CLS");
    cout<<"\n\n";
    cout<<"\t==============================================\n"; Sleep(10);
    cout<<"\n\t    Student Information Management System"<<endl<<endl; Sleep(10);
    cout<<"\t==============================================\n"; Sleep(10);
    cout<<"\n\t\t\t   Home"<<endl; Sleep(10);
    cout<<"\n\t[1] Main Menu"<<endl; Sleep(10);
    cout<<"\t[2] Open Recycle Bin"<<endl; Sleep(10);
    cout<<"\t[3] Meet The Developer's"<<endl; Sleep(10);
    cout<<"\t[0] Exit the Program."<<endl; Sleep(10);
    cout<<"\t==============================================\n"; Sleep(10);
    cout<<"\tEnter your Choice: ";
}

void menu(){
    system("CLS");
    cout<<"\n\n";
    cout<<"\t==============================================\n"; Sleep(10);
    cout<<"\n\t    Student Information Management System"<<endl<<endl; Sleep(10);
    cout<<"\t==============================================\n"; Sleep(10);
    cout<<"\t\t\t  Main Menu"<<endl; Sleep(10);
    cout<<"\n\t[1] Add a new Student."<<endl; Sleep(10);
    cout<<"\t[2] Search a Student by ID."<<endl; Sleep(10);
    cout<<"\t[3] View all Student's Info."<<endl; Sleep(10);
    cout<<"\t[4] Update any Student's Info."<<endl; Sleep(10);
    cout<<"\t[5] Delete any Student's Info."<<endl; Sleep(10);
    cout<<"\t[6] Delete All Student's Info."<<endl; Sleep(10);
    cout<<"\t[7] Restore deleted Info."<<endl; Sleep(10);
    cout<<"\t[9] Return to Home."<<endl; Sleep(10);
    cout<<"\t[0] Exit The program."<<endl; Sleep(10);
    cout<<"\t==============================================\n"; Sleep(10);
    cout<<"\tEnter your Choice: ";
    int ope;
    cin>>ope;
    switch(ope){
```

G:\Data Structure\Student Info Management System\Student Info Manageme... | C/C++ | Windows (CR+LF) | WINDOWS-1252 | Line 1, Col 1, Pos 0 | Insert | Read/Write | default

main.cpp - Code::Blocks 20.03

File Edit View Search Project Build Debug Fortran wxSmith Tools Tools+ Plugins DoxyBlocks Settings Help

<global>

Start here | main.cpp

```cpp

void addNew(){
    cout<<"\n\n"<<endl;
    string s = "You wanted to add a new student. \n\tPlease enter his/her detailed information";
    slowTxt(s);

    //Input information from User
    string ID = scanID();
    string name = scanName();
    string phone = scanPhone();
    string email = scanEmail();

    students *student = new students();
    student->ID = ID;
    student->name = name;
    student->email = email;
    student->phone = phone;
    student->next = NULL;

    cout<<"\tEnter The Number of Courses: ";
    cin>>student->courseNum;

    for(int i = 1;i<=student->courseNum;i++){
        cout<<"\tCourse-"<<i<<": \n";
        string cCode = scanCCode();
        string cName = scanCName();

        student->course[cCode] = cName; //assigning map value - Course code is key and name is value
    }

    if(start == NULL){
        start = student;
    }
    else{
        students *p = start;
        while(p->next!=NULL){
```

G:\Data Structure\Student Info Management System\Student Info Manageme... | C/C++ | Windows (CR+LF) | WINDOWS-1252 | Line 1, Col 1, Pos 0 | Insert | Read/Write | default

# CHAPTER 6

# CONCLUSION AND FUTURE SCOPE

## 6.1 Conclusion

In summary, our Student Information Management System (SIMS) project aims to make handling student information in schools easier. We expect challenges, like fitting the new system with the existing ones and keeping student data safe. Despite these challenges, we're working on solutions to make the system user-friendly and adaptable. Overall, the project aims to bring positive changes to how schools manage student records, making it more efficient and straightforward.

## 6.2 Future Scope

In the future, we are willing to develop this project as a Web Application. We are trying to create a database and connect with our service. Also working on admin and student-based interface. They will be able to log in to their profile and use the service. Also, we will try to add more features to this application according to demand. In future development, we will provide device service.

## 6.3 Limitation

Everything has its limitations and we are not exceptional, we have some limitations too. We tried as much as we could to avoid limitations. It could be more dynamic and user-friendly. Unauthorized users or anyone can make misuse of this platform with fake documentation. Besides, we were unable to implement a sorting algorithm because our data was stored in the file. Also, we have some bugs in input buffer systems.

**Reference:**


- https://www.geeksforgeeks.org/data-structures/
- https://www.javatpoint.com/data-structure-tutorial
- https://www.w3schools.com/cpp/cpp_structs.asp
- https://www.youtube.com/watch?v=uPTlVsIZr5o (Idea)