PROJECT REPORT

Submitted by

Gerald Alan Raj – RA2211056010147
Sai Suhaas – RA2211056010154
Dheepak – RA2211056010159
Mamtha Sri – RA2211056010161
Pradeep Raj – RA2211031010165

Under the Guidance of

Dr. Siva Kumar A

Assistant Professor, Department of Data Science and Business Systems

In partial satisfaction of the requirements for the degree of

BACHELOR OF TECHNOLOGY
in
COMPUTER SCIENCE ENGINEERING

with specialization in Data Science

SCHOOL OF COMPUTING

COLLEGE OF ENGINEERING AND TECHNOLOGY

SRM INSTITUTE OF SCIENCE AND TECHNOLOGY

KATTANKULATHUR - 603203

MAY 2023

1

SRM INSTITUTION OF SCIENCE AND TECHNOLOGY
KATTANKULATHUR-603203

BONAFIDE CERTIFICATE

Certified that this Project Report titled **"Students Report Card Management System"** is the bonafide work done by *Gerald Alan Raj(RA2211056010147),Sai Suhaas(RA2211056010154), Dheepak(RA2211056010159), Mamtha Sri(RA2211056010161) and Pradeep Raj (RA2211031010165)* who completed the project under my supervision. Certified further, that to the best of my knowledge the work reported herein does not form part of any other work.

SIGNATURE
Dr. Siva Kumar A
OODP – Course Faculty
Assistant Professor
Department of Data Science
and Business Systems
SRMIST

SIGNATURE
Dr. Lakshmi M
Head of the Department
Department of Data Science
and Business Systems
SRMIST

2

TABLE OF CONTENTS

S.No	CONTENTS	PAGE NO
1.	Problem Statement	4
2.	Modules of Project	5
3.	Diagrams	6
	a. Use case Diagram	6
	b. Class Diagram	7
	c. Sequence Diagram	8
	d. State Chart Diagram	9
	e. Activity Diagram	10
	f. Component Diagram	11
	g. Deployment Diagram	12
4.	Code/Output Screenshots	13
5.	Conclusion and Results	34
6.	References	34

3

Problem Statement:

The Student Marksheet Management System is a software application designed to manage and store the academic records of students. The primary objective of this system is to automate the process of managing the marks of the students and generating their marksheet.

The system will have a database of students and their personal information, including their names, roll numbers, contact information, and academic information, including the marks they scored in different subjects. The system will also have features to update and delete student records and generate mark sheets.

The system should provide the following functionalities:
Add new students to the system along with their personal information and academic records.
Update the information of existing students in the system.
Delete the records of students who have left the institution.
Display the records of all the students in the system.
Generate the marksheet of a particular student.
Calculate the total marks and percentage of a student.
Display the ranks of all the students in the system.

The system should be user-friendly and easy to use. It should have a graphical user interface (GUI) to make it easier for the user to navigate and perform different tasks. The programming language used to implement the system is C++.

4

Modules of the Project:

A Student Marksheet Management System typically includes several modules that work together to manage and maintain the student's academic records. Here are some common modules that you may find in such a system:

Student Information Management Module: This module is responsible for managing the personal information of students, including their names, addresses, contact details, etc.

Academic Information Management Module: This module is responsible for managing the academic information of students, including their marks, grades, and progress reports.

User Management Module: This module is responsible for managing user accounts and access levels for different users of the system, such as teachers, administrators, and students.

Marksheet Generation Module: This module is responsible for generating mark sheets and progress reports for students, based on the data stored in the system.

Reporting Module: This module is responsible for generating various reports related to student performance, such as class averages, subject-wise rankings, etc.

Communication Module: This module is responsible for facilitating communication between teachers, administrators, and students, such as sending notifications, messages, or alerts related to academic matters.

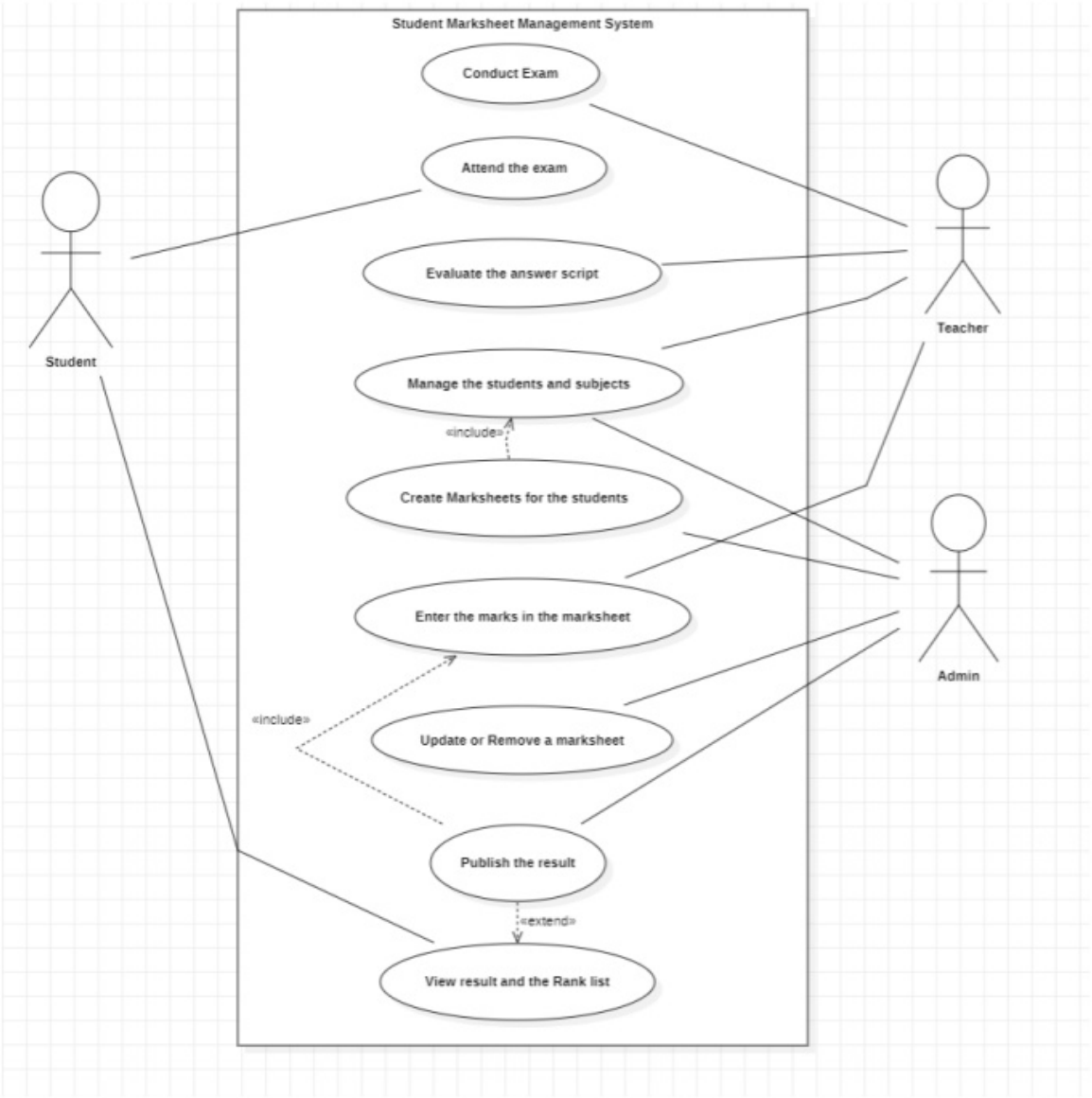
Backup and Recovery Module: This module is responsible for ensuring that the data in the system is backed up regularly and can be restored in case of any data loss or system failure.

These are some of the common modules used in a Student Marksheet Management System. Depending on the specific requirements of your system, you may include additional modules or modify the existing ones to suit your needs.

5

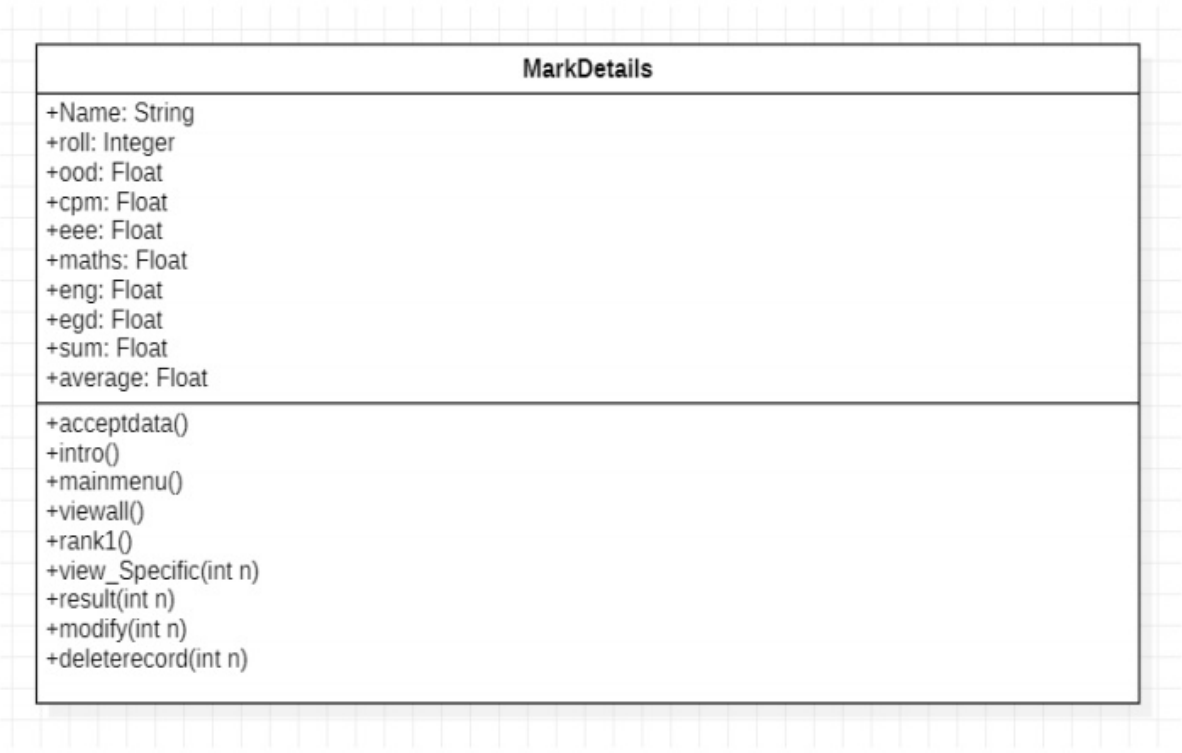
Diagrams:

a. Use case Diagram:

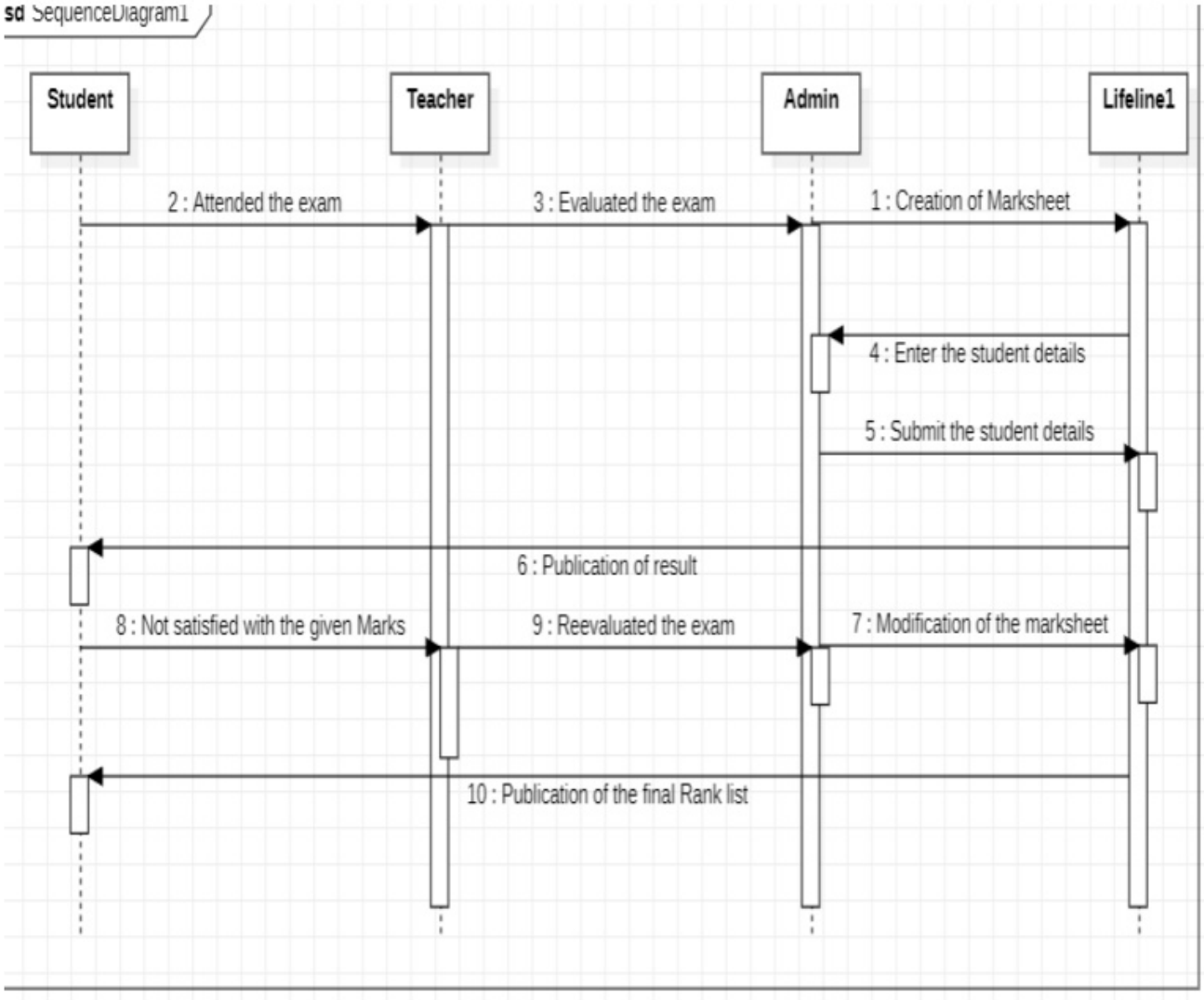


6

b. Class Diagram

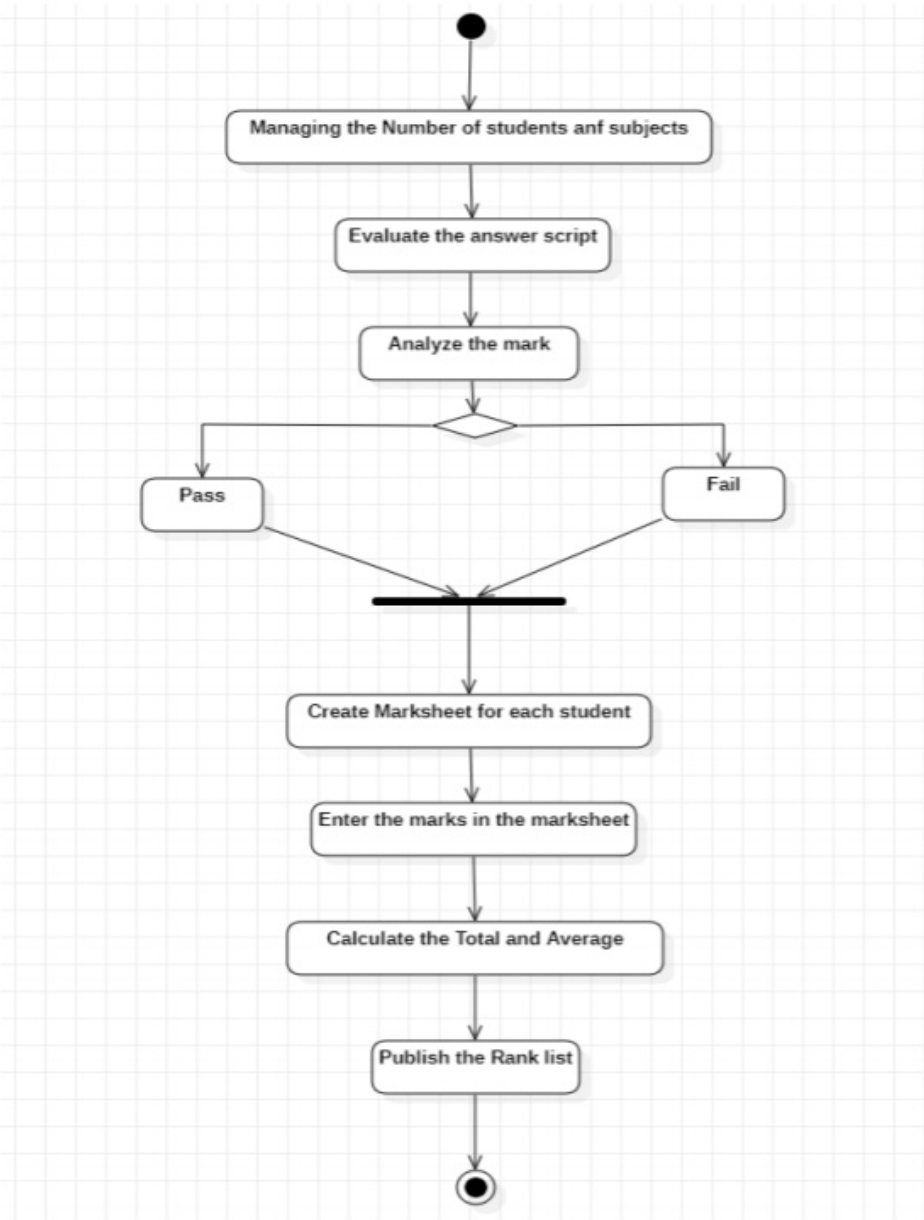


c. Sequence Diagram



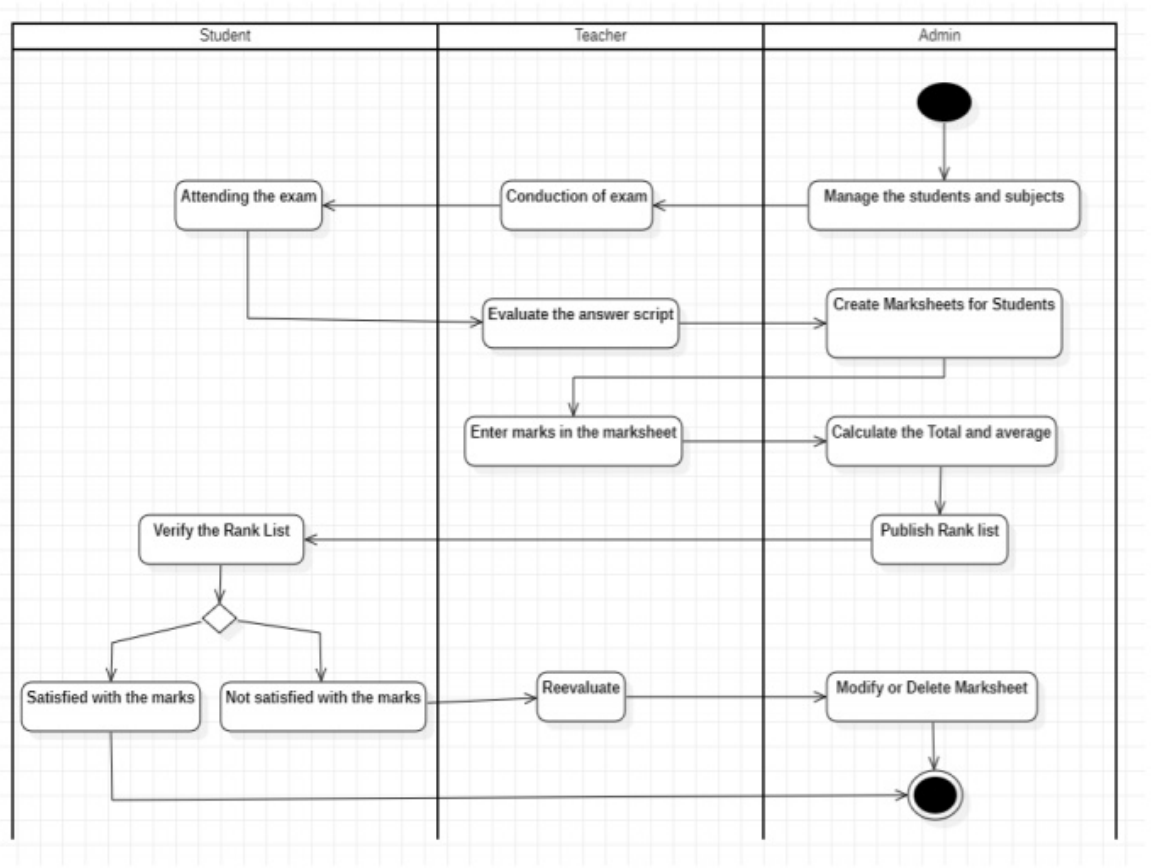
8

e. State Chart Diagram:



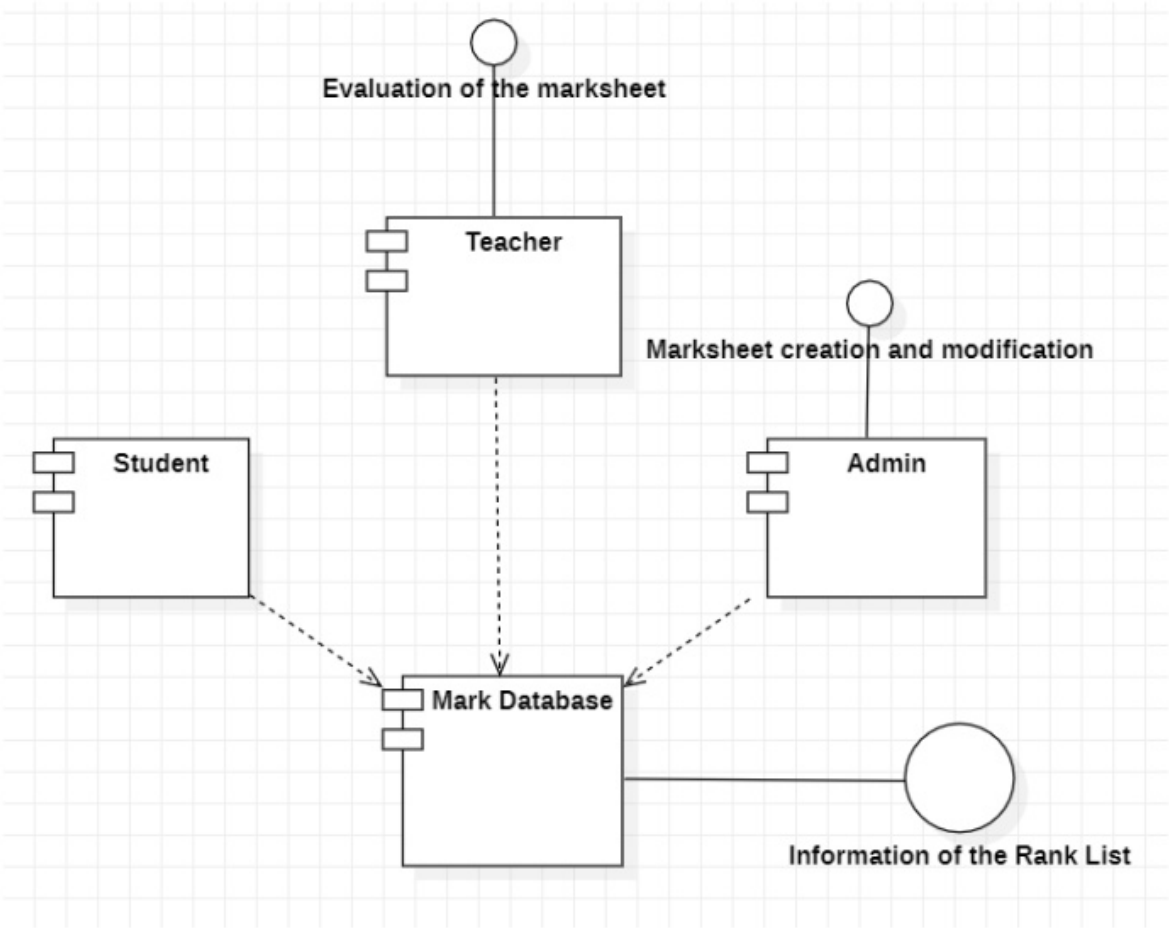
9

f. Activity Diagram



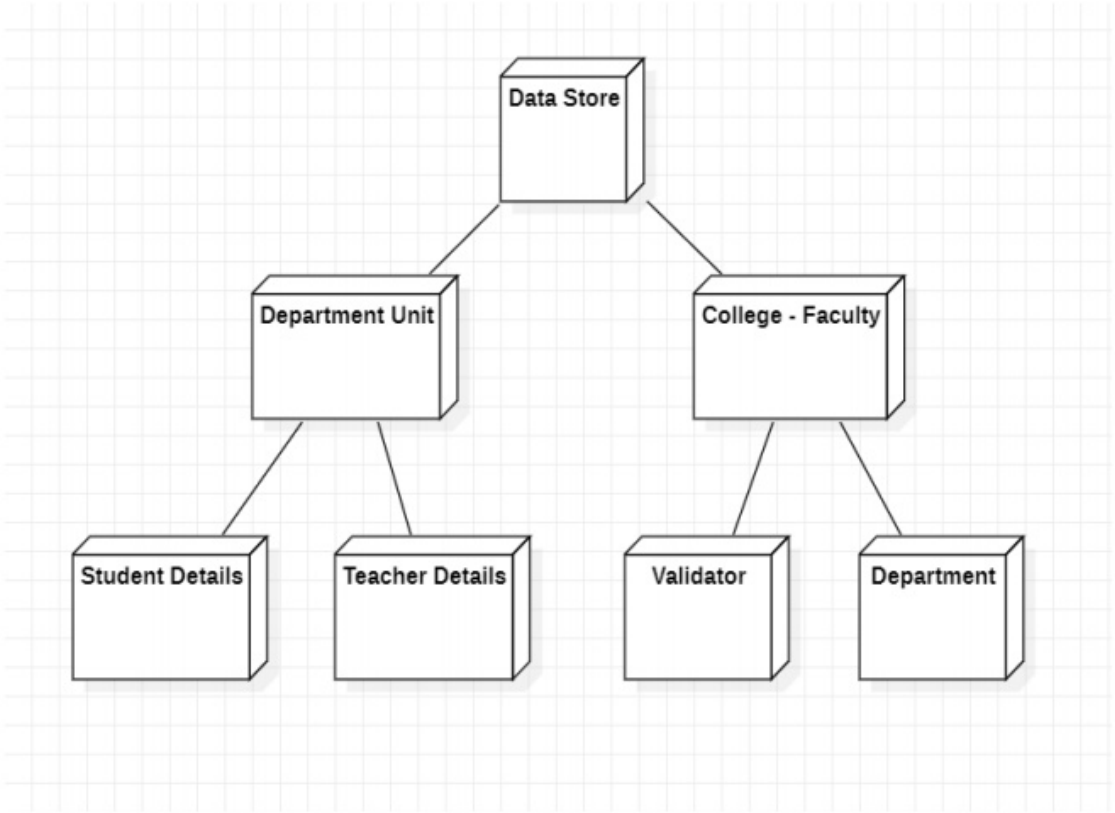
10

h. Component Diagram



11

i. Deployment Diagram



12

Code/Output Screenshots

```
#include <iostream>
#include<fstream>
#include<iomanip>
#include<windows.h>
using namespace std;
struct student
{
   char name[80];
   int roll;
   float ood, cpm, eee, maths, eng, egd, sum, average;
};

class MarkDetails
{

private:
   char name[80];
   int roll;
   float ood, cpm, eee, maths, eng, egd, sum, average;

 public:
   void acceptdata()
   {
      system("cls");
      student s;
      ofstream outfile;
      outfile.open("Report.txt", ios::app | ios::binary);
      if (outfile.fail())
      {
         cout << "THE FILE COULD NOT BE OPEN...press enter key";
         cin.ignore();
         cin.get();
```


13

```
    }
    cout << "\n\n";
    cout << "\t\t\t\t=======CREATE A REPORT CARD========\n\n";
    cout << "ENTER YOUR FULL NAME :";
    cin.ignore();
    cin.getline(s.name, 80);
    cout << "ENTER YOUR ROLL NUMBER :";
    cin >> s.roll;
    cout << "ENTER YOUR OOD MARK :";
    cin >> s.ood;
    cout << "ENTER YOUR COMPUTATIONAL PHYSICS MARK :";
    cin >> s.cpm;
    cout << "ENTER YOUR EEE MARK :";
    cin >> s.eee;
    cout << "ENTER YOUR MATHS MARK :";
    cin >> s.maths;
    cout << "ENTER YOUR ENGLISH MARK :";
    cin >> s.eng;
    cout << "ENTER YOUR EGD MARK :";
    cin >> s.egd;
    s.sum = s.ood + s.cpm + s.eee + s.maths + s.eng + s.egd;
    s.average = (s.sum / 6);
    outfile.write(reinterpret_cast<char*> (&s), sizeof(student));
    outfile.close();
    cout << endl;
    cout << "\t\t\t\tTHE FILE IS SUCCESSFULLY SAVED" << endl;
    cout << endl;
    cout << "press any key to continue...";
    cin.ignore();
    cin.get();
}

void intro()

{
```


14

```
        cout << "\n\n\n";
        Sleep(300);
        cout << "\t\t\t\t * * *  **** *     **** ***  ***  ****  " << endl;
        Sleep(300);
        cout << "\t\t\t\t * * * *    *    *    *  * * * * *      " << endl;
        Sleep(300);
        cout << "\t\t\t\t * * * ***** *    *    *  * * * * *****   " << endl;
        Sleep(300);
        cout << "\t\t\t\t * * * *    *    *    *  * * * * *      " << endl;
        Sleep(300);
        cout << "\t\t\t\t  ***   **** ***** **** *** * * * ****    " << endl;
        Sleep(300);
        cout << endl;
        cout << "\t\t\t    ================================================" <<
endl;
        Sleep(500);
        cout << "\t\t\t    THIS IS THE STUDENT MARKSHEET MANAGEMENT SOFTWARE"
<< endl;
        Sleep(500);
        cout << "\t\t\t    ================================================" <<
endl;
        Sleep(500);
        cout << "press any key to continue...";
        cin.ignore();
        cin.get();
    }

    void mainmenu()

    {
        system("color c");
        char cc;
        cout << "\t\t\t\t==================MAIN MENU================\n\n" << endl;
        Sleep(300);
```

```
cout << "\t\t\t\t1. CREATE STUDENT REPORT CARD\n\n" << endl;
Sleep(300);
cout << "\t\t\t\t2. VIEW ALL STUDENTs REPORT CARD\n\n" << endl;
Sleep(300);
cout << "\t\t\t\t3. VIEW A SINGLE STUDENT REPORT CARD\n\n" << endl;
Sleep(300);
cout << "\t\t\t\t4. MODIFY REPORT CARD\n\n" << endl;
Sleep(300);
cout << "\t\t\t\t5. RESULT\n\n" << endl;
Sleep(300);
cout << "\t\t\t\t6. RANK LIST\n\n" << endl;
Sleep(300);
cout << "\t\t\t\t7. DELETE RECORD\n\n" << endl;
Sleep(300);
cout << "\t\t\t\t========================================" << endl;
Sleep(300);
cout << "\t\t\t\tENTER YOUR CHOICE...:) <1-6> :";
Sleep(300);
cin >> cc;
cout << endl;
switch (cc)
{
case '1':
{
   acceptdata();
   break;
}
case '2':
{
   viewall();
   break;
}
case '3':
{
   int n;
```


16

```
    cout << "ENTER YOUR ROLL NUMBER :";
    cin >> n;
    view_Specific(n);
    break;
}
case '4':
{
    int n;
    cout << "ENTER YOUR ROLL NUMBER :";
    cin >> n;
    modify(n);
    break;
}

case '5':
{
    int n;
    cout << "ENTER YOUR ROLL NUMBER :";
    cin >> n;
    cout << endl;
    result(n);
    break;
}
case '6':
{
    int n;
    rank1();
    break;
}
case '7':
{
    int n;
    cout << "ENTER YOUR ROLL NUMBER :";
    cin >> n;
    cout << endl;
```


17

```
            deleterecord(n);
            break;
        }
        }
    }

    void viewall()

    {
        system("cls");
        student s;
        ifstream infile;
        bool check = false;
        infile.open("Report.txt", ios::app | ios::binary);
        if (infile.fail())
        {
            cout << "THE FILE COULD NOT BE OPENED.....press enter key...";
            cin.ignore();
            cin.get();
        }
        cout << "\n\n";
        cout << "\t\t\t\tALL STUDENTS REPORT CARDS" << endl;
        cout                                                                                    <<
"=========================================================================
==============" << endl;
        while (infile.read(reinterpret_cast<char*>(&s), sizeof(student)))
        {
            cout << "\t\t\t\tSTUDENT NAME :" << s.name << endl;
            cout << "\t\t\t\tSTUDENT ROLL NUMBER :" << s.roll << endl;
            cout << "\t\t\t\tOOD MARK :" << s.ood << endl;
            cout << "\t\t\t\tCOMPUTATIONAL PHYSICS MARK :" << s.cpm << endl;
            cout << "\t\t\t\tEEE MARK :" << s.eee << endl;
            cout << "\t\t\t\tMATHS MARK :" << s.maths << endl;
            cout << "\t\t\t\tENGLISH MARK :" << s.eng << endl;
            cout << "\t\t\t\tEGD MARK :" << s.egd << endl;
```


18

```
            cout << "\t\t\t\tSUM :" << s.sum << endl << endl;
            cout << "\t\t\t\tAVERAGE :" << s.average << endl << endl;
            cout                                                                    <<
"=============================================================================
===============" << endl;
            check = true;
        }
        infile.close();
        if (check == false)
            cout << "\t\t\t\tNO RECORD FOUND..." << endl << endl;
        cout << "press any key to continue....";
        cin.ignore();
        cin.get();
    }

    void rank1()

    {
        system("cls");
        student s[100];
        ifstream infile;
        bool check = false;
        infile.open("Report.txt", ios::app | ios::binary);
        if (infile.fail())
        {
            cout << "THE FILE COULD NOT BE OPENED.....press enter key...";
            cin.ignore();
            cin.get();
        }
        cout << "\n\n";
        cout << "\t\t\t\t\t\t\t\t\t\tRANK LIST OF STUDENTS" << endl;
        cout                                                                    <<
"=============================================================================
=============================================================================
```


19

```
==========" << endl;
        int n = 0;
        while (infile.read(reinterpret_cast<char*>(&s[n]), sizeof(student)))
        {
           n++;
        }
        infile.close();
        //sorting based on AVERAGE MARKS
        for (int a = 0; a < n; a++)
        {
           for (int b = a + 1; b < n; b++)



           {
              if (s[a].average < s[b].average)
              {
                 student tmp;
                 tmp = s[a];
                 s[a] = s[b];
                 s[b] = tmp;
              }
           }
        }
        cout << "\tRANK  ";
        cout << "\tSTUDENT NAME";
        cout << "\tSTUDENT ROLL NUMBER";
        cout << "\t     OOD MARK";
        cout << "\t COMPUTATIONAL PHYSICS MARK";
        cout << "\tEEE MARK";
        cout << "\tMATHS MARK";
        cout << "\tENGLISH MARK";
        cout << "\tEGD MARK";
        cout << "\tTOTAL ";
        cout << "\t\tAVERAGE " << endl;
```

```
    cout                                                                                  <<
"======================================================================
======================================================================
======================================================================
=========" << endl;
    for (int i = 0; i < n; i++)
    {
      cout << "\t" << i + 1;
      cout << "\t" << setw(10)<< s[i].name;
      cout << "\t\t" << s[i].roll;
      cout << "\t\t\t" << s[i].ood;
      cout << "\t\t\t" << s[i].cpm;
      cout << "\t\t" << s[i].eee;
      cout << "\t\t" << s[i].maths;
      cout << "\t\t" << s[i].eng;
      cout << "\t\t" << s[i].egd;
      cout << "\t\t" << s[i].sum;
      cout << "\t\t" << s[i].average << endl;
      cout                                                                              <<
"======================================================================
======================================================================
======================================================================
=========" << endl;
      check = true;
    }

    if (check == false)
      cout << "\t\t\t\tNO RECORD FOUND..." << endl << endl;
    cout << "press any key to continue....";
    cin.ignore();
    cin.get();
  }
  void view_Specific(int n)
  {
    system("cls");
```

```
    student s;
    ifstream infile;
    infile.open("Report.txt", ios::app | ios::binary);
    if (infile.fail())
    {
       cout << "THE FILE COULD NOT BE OPENED...";

       cin.ignore();
       cin.get();
    }
    bool equality = false;
    cout << "\t\t\t\t==========VIEW A SINGLE STUDENT REPORT==========\n\n";
    while (infile.read(reinterpret_cast<char*>(&s), sizeof(student)))
    {
       if (s.roll == n)
       {
          cout << "\t\t\t\tSTUDENT NAME :" << s.name << endl;
          cout << "\t\t\t\tSTUDENT ROLL NUMBER :" << s.roll << endl;
          cout << "\t\t\t\tOOD MARK :" << s.ood << endl;
          cout << "\t\t\t\tCOMPUTATIONAL PHYSICS MARK :" << s.cpm << endl;
          cout << "\t\t\t\tEEE MARK :" << s.eee << endl;
          cout << "\t\t\t\tMATHS MARK :" << s.maths << endl;
          cout << "\t\t\t\tENGLISH MARK :" << s.eng << endl;
          cout << "\t\t\t\tEGD MARK :" << s.egd << endl;
          cout << "\t\t\t\tTOTAL :" << s.sum << endl;
          cout << "\t\t\t\tAVERAGE :" << s.average << endl;
          cout  <<  "\t\t\t\t================================================="
<< endl;
          equality = true;
       }
    }
    infile.close();
    if (equality == false)
       cout << "\t\t\t\tRECORD NOT FOUND..." << endl;
    cout << endl;
```


22

```
    cout << "press any key to continue...";
    cin.ignore();
    cin.get();
}

void result(int n)
{
    system("cls");
    student s;
    ifstream infile;
    infile.open("Report.txt", ios::app | ios::binary);
    if (infile.fail())
    {
        cout << "THE FILE COULD NOT BE OPENED..." << endl;
        cin.ignore();
        cin.get();
    }
    bool eq = false;
    cout << "\t\t\t==========VIEW A SINGLE STUDENT RESULT==========\n\n";
    while (infile.read(reinterpret_cast<char*>(&s), sizeof(student)))
    {
        if (s.roll == n)
        {
            cout << "\t\t\tSTUDENT NAME :" << s.name << endl;
            cout << "\t\t\tSTUDENT ROLL NUMBER :" << s.roll << endl;
            cout << "\t\t\tOOD MARK :" << s.ood << endl;
            cout << "\t\t\tCOMPUTATIONAL PHYSICS MARK :" << s.cpm << endl;
            cout << "\t\t\tEEE MARK :" << s.eee << endl;
            cout << "\t\t\tMATHS MARK :" << s.maths << endl;
            cout << "\t\t\tENGLISH MARK :" << s.eng << endl;
            cout << "\t\t\tEGD MARK :" << s.egd << endl;
            cout << "\t\t\tTOTAL :" << s.sum << endl << endl;
            cout << "\t\t\tAVERAGE :" << s.average << endl << endl;
            cout                                                                    <<
"\t\t\t================================================" << endl;
```


23

```
            eq = true;
        }
    }
    infile.close();
    if (eq == false)
        cout << "\t\t\t\tRECORD NOT FOUND..." << endl;
    cout << endl;
    cout << "press any key to continue...";
    cin.ignore();
    cin.get();
}

void modify(int n)
{
    system("cls");
    student s;
    fstream infile;
    infile.open("Report.txt", ios::binary | ios::in | ios::out);
    if (infile.fail())
    {
        cout << "THE FILE COULD NOT BE OPENED..." << endl;
        cin.ignore();
        cin.get();
    }
    bool checker = false;
    cout << "\t\t\t==========MODIFY A REPORT CARD==========\n\n";
    while (!infile.eof() && checker == false)
    {
        infile.read(reinterpret_cast<char*>(&s), sizeof(student));
        {
            if (s.roll == n)
            {
                cout << "\t\t\t\tSTUDENT NAME :" << s.name << endl;
                cout << "\t\t\t\tSTUDENT ROLL NUMBER :" << s.roll << endl;
                cout << "\t\t\t\tOOD MARK :" << s.ood << endl;
```


24

```
        cout << "\t\t\t\tCOMPUTATIONAL PHYSICS MARK :" << s.cpm << endl;
        cout << "\t\t\t\tEEE MARK :" << s.eee << endl;
        cout << "\t\t\t\tMATHS MARK :" << s.maths << endl;
        cout << "\t\t\t\tENGLISH MARK :" << s.eng << endl;
        cout << "\t\t\t\tEGD MARK :" << s.egd << endl;
        cout << "==========================================" << endl;
        cout << "\t\tENTER THE NEW INFORMATION" << endl;
        cout << "==========================================" << endl;
        cout << "ENTER YOUR FULL NAME :";
        cin.ignore();
        cin.getline(s.name, 80);
        cout << "ENTER YOUR ROLL NUMBER :";
        cin >> s.roll;
        cout << "ENTER YOUR OOD MARK :";
        cin >> s.ood;
        cout << "ENTER YOUR COMPUTATIONAL PHYSICS MARK :";
        cin >> s.cpm;
        cout << "ENTER YOUR EEE MARK :";
        cin >> s.eee;
        cout << "ENTER YOUR MATHS MARK :";
        cin >> s.maths;
        cout << "ENTER YOUR ENGLISH MARK :";
        cin >> s.eng;
        cout << "ENTER YOUR EGD MARK :";
        cin >> s.egd;
        s.sum = s.ood + s.cpm + s.eee + s.maths + s.eng + s.egd;
        s.average = (s.sum / 6);
        int pos = (-1) * static_cast<int>(sizeof(student));
        infile.seekp(pos, ios::cur);
        infile.write(reinterpret_cast<char*> (&s), sizeof(student));
        cout << endl;
        cout << "\t\t\t\tTHE FILE IS SUCCESSFULLY updated" << endl;
        checker = true;
    }
}
```


25

```
    }
    infile.close();
    if (checker == false)
        cout << "\t\t\t\tRECORD NOT FOUND" << endl;
    cout << endl;
    cout << "press any key to continue...";
    cin.ignore();
    cin.get();
}

void deleterecord(int n)
{
    system("cls");
    student s;
    ifstream infile;
    infile.open("Report.txt", ios::binary);
    if (!infile)
    {
        cout << "THE FILE COULD NOT BE OPENED..." << endl;
        cin.ignore();
        cin.get();
    }
    ofstream outfile;
    outfile.open("Record2.txt", ios::binary);
    infile.seekg(0, ios::beg);
    cout << "\t\t\t\t==========DELETE A REPORT CARD==========\n\n";
    while (infile.read(reinterpret_cast<char*>(&s), sizeof(student)))
    {
        if (s.roll != n)
        {
            outfile.write(reinterpret_cast<char*>(&s), sizeof(student));
        }
    }
    infile.close();
    outfile.close();
```

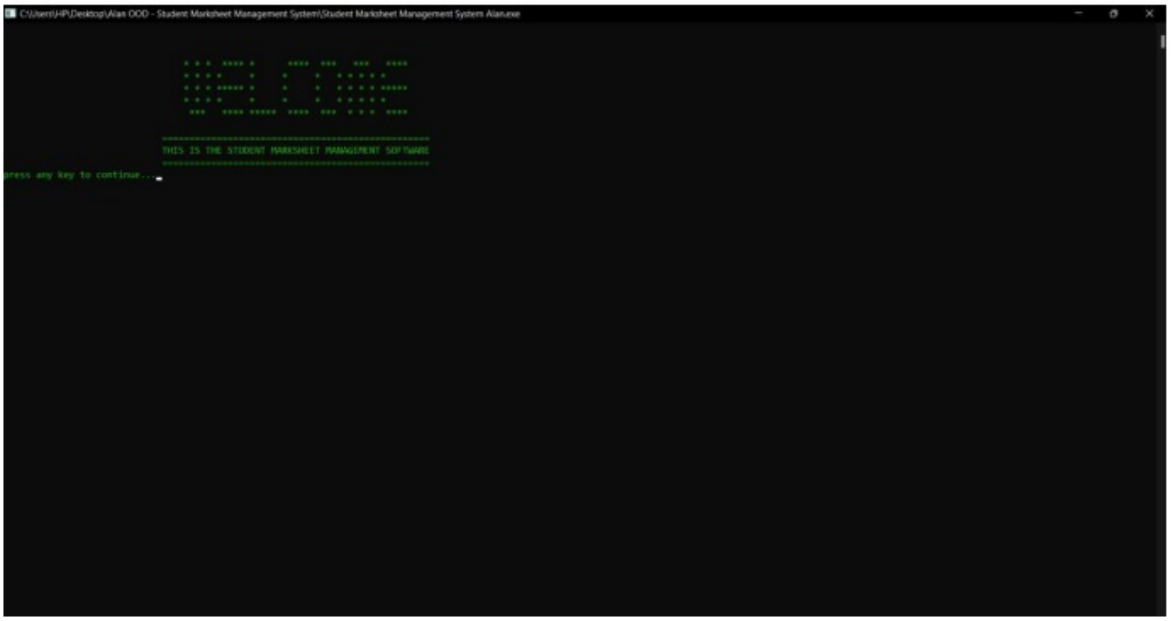

26

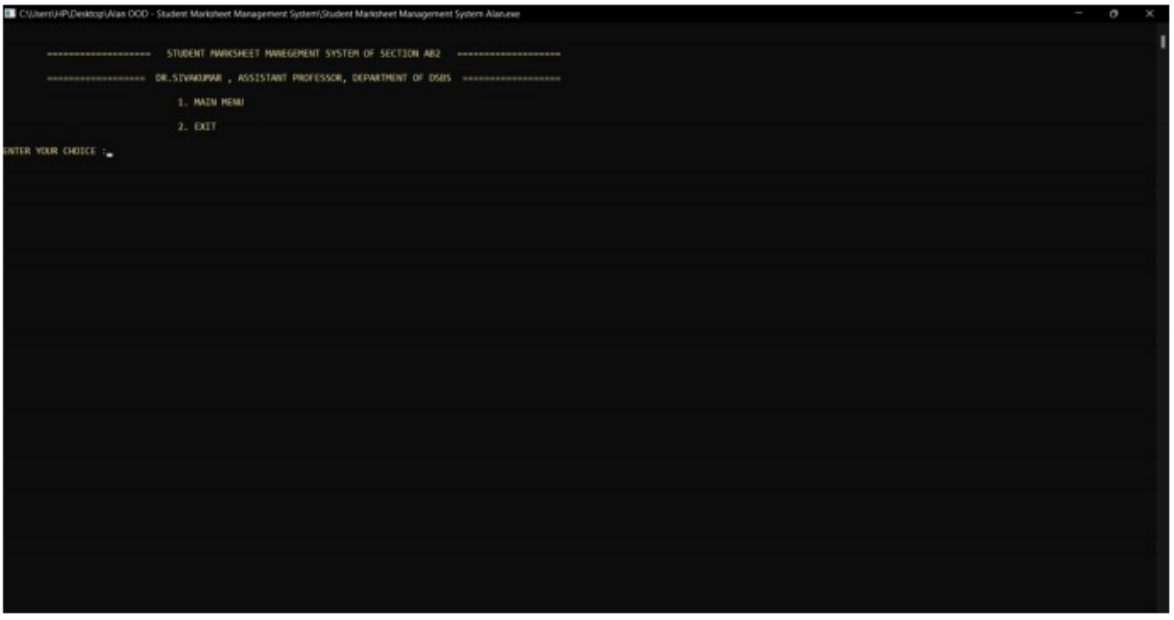
```
        remove("Report.txt");
        rename("Record2.txt", "Report.txt");
        cout << endl;
        cout << "\t\t\t\tRECORD SUCCESSFULLY DELETED" << endl;
        cout << "press any key to continue...";
        cin.ignore();
        cin.get();
    }
};

int main()
{
    system("color a");
    char c;
    MarkDetails m;
    system("cls");
    m.intro();

    do {
        system("cls");
        system("color e");
        cout << "\n\n";
        cout << "\t==========     STUDENT MARKSHEET MANEGEMENT
SYSTEM OF SECTION AB2  ==========" << endl;
        cout << endl;
        cout << "\t==========  DR.SIVAKUMAR , ASSISTANT PROFESSOR,
DEPARTMENT OF DSBS  ==========" << endl;
        cout << endl;
        cout << "\t\t\t\t1. MAIN MENU\n\n";
        cout << "\t\t\t\t2. EXIT\n\n";
        cout << "ENTER YOUR CHOICE :";
        cin >> c;
        system("cls");
        switch (c)
        {
```


27

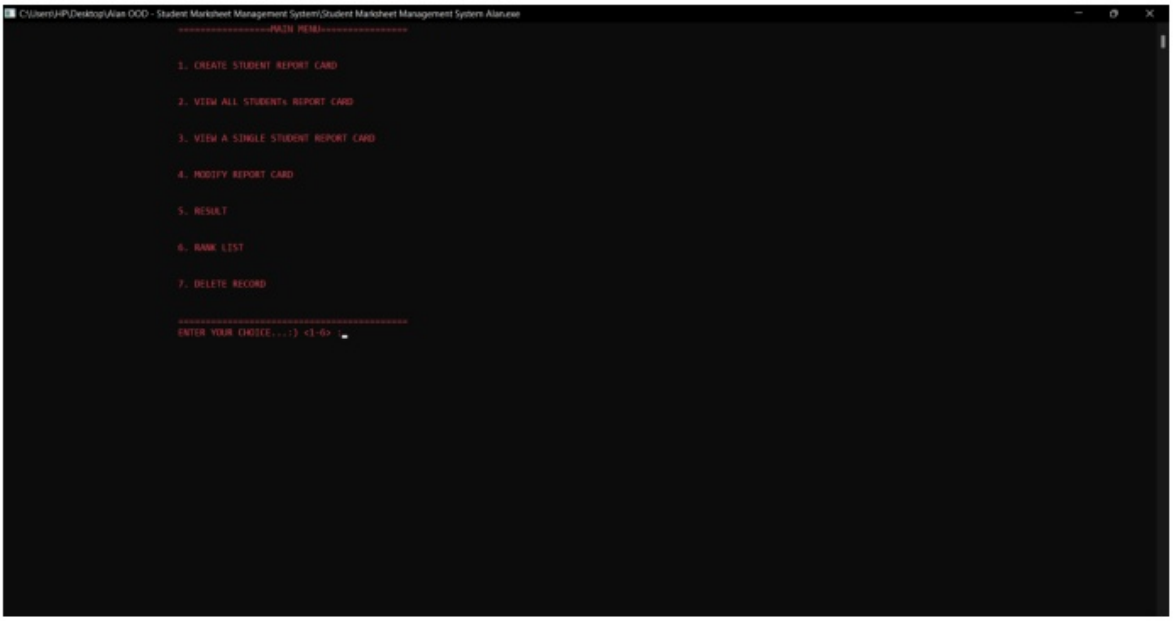
```
        case '1':
        {
           m.mainmenu();
           break;
        }
        case '2':
        {         cout << "\t\t     THANK YOU FOR USING THE SOFTWARE" << endl;
        cout << "\n\n";
        cout << "\t   GROUP MEMBERS(DEVELOPERS)";
        cout << "\n\n";
        cout << "\t   1. GERALD ALAN RAJ\n\n";
        cout << "\t   2. SAI SUHAAS \n\n";
        cout << "\t   3. DHEEPAK\n\n";
        cout << "\t   4. MAMTHA SRI\n\n";
        cout << "\t   5. PRADEEP RAJ\n\n";
        cout << "\n\n";
        }
        }
    } while (c != '2');
    return 0;
}
```

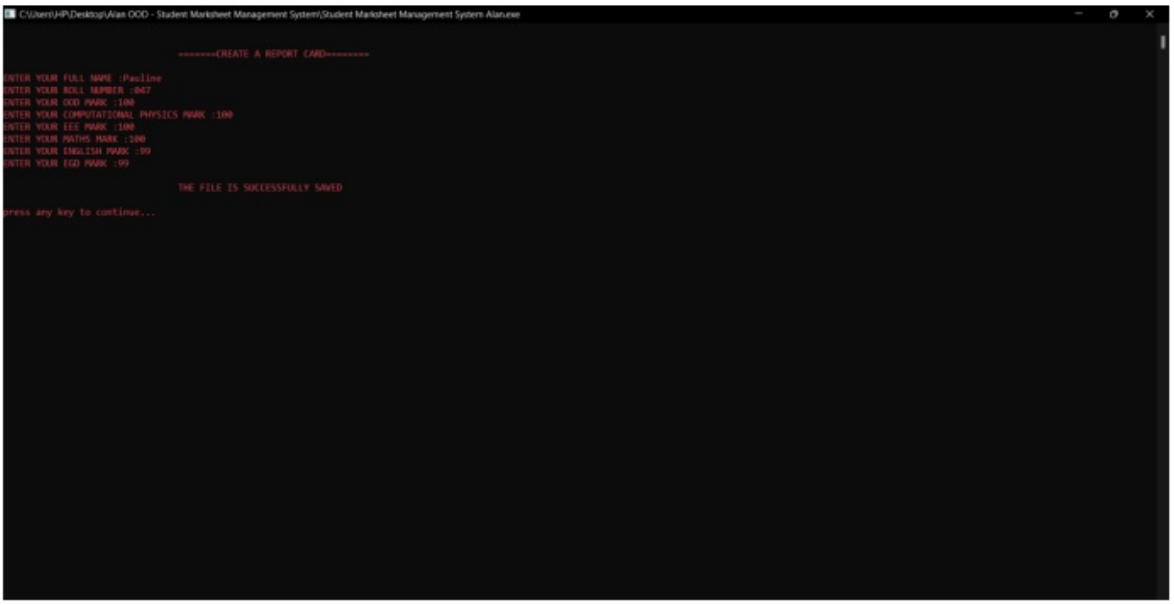

28


C:\Users\HP\Desktop\Alan OOD - Student Marksheet Management System\Student Marksheet Management System Alan.exe
THIS IS THE STUDENT MARKSHEET MANAGEMENT SOFTWARE
press any key to continue...

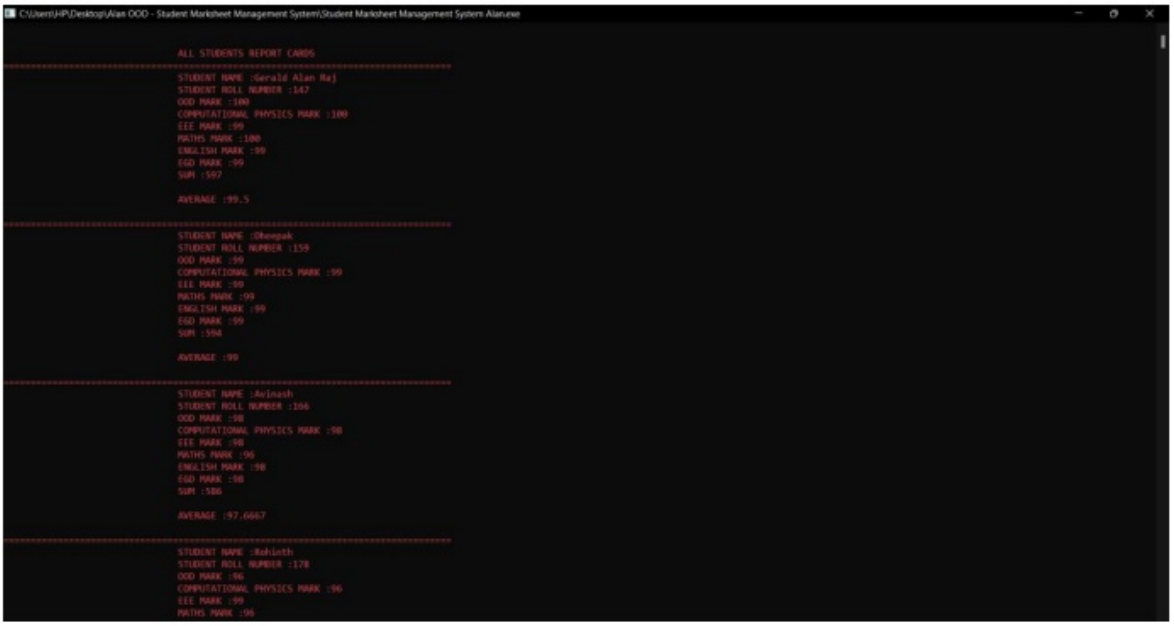

C:\Users\HP\Desktop\Alan OOD - Student Marksheet Management System\Student Marksheet Management System Alan.exe
STUDENT MARKSHEET MANEGEMENT SYSTEM OF SECTION AB2
DR.SIVAKUMAR , ASSISTANT PROFESSOR, DEPARTMENT OF DSBS
1. MAIN MENU
2. EXIT
ENTER YOUR CHOICE :

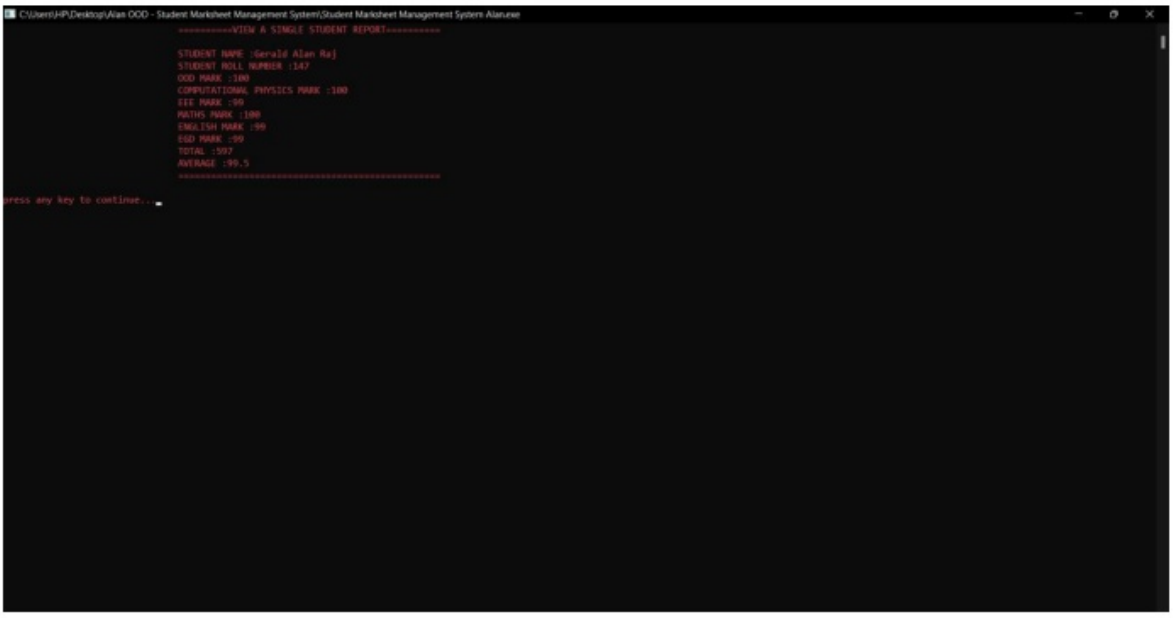
29


C:\Users\HP\Desktop\Aian OOD - Student Marksheet Management System\Student Marksheet Management System Aian.exe
==========MAIN MENU==========
1. CREATE STUDENT REPORT CARD
2. VIEW ALL STUDENTs REPORT CARD
3. VIEW A SINGLE STUDENT REPORT CARD
4. MODIFY REPORT CARD
5. RESULT
6. RANK LIST
7. DELETE RECORD
==========
ENTER YOUR CHOICE...:) <1-6> :

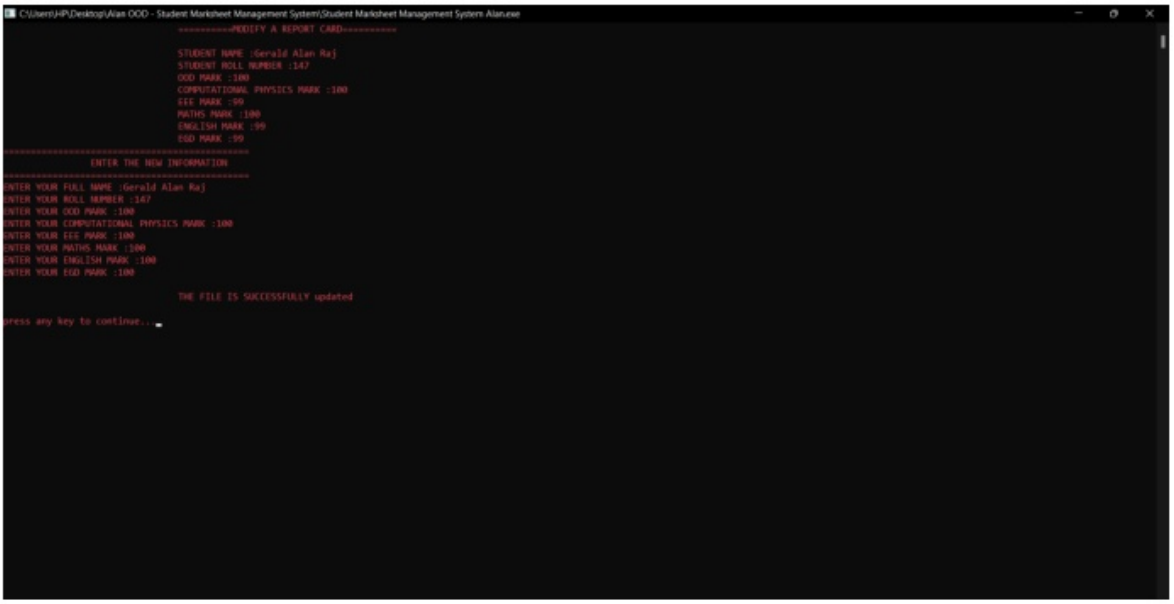

C:\Users\HP\Desktop\Aian OOD - Student Marksheet Management System\Student Marksheet Management System Aian.exe
=======CREATE A REPORT CARD=======
ENTER YOUR FULL NAME :Pauline
ENTER YOUR ROLL NUMBER :047
ENTER YOUR OOD MARK :100
ENTER YOUR COMPUTATIONAL PHYSICS MARK :100
ENTER YOUR EEE MARK :100
ENTER YOUR MATHS MARK :100
ENTER YOUR ENGLISH MARK :99
ENTER YOUR EGD MARK :99
THE FILE IS SUCCESSFULLY SAVED
press any key to continue...

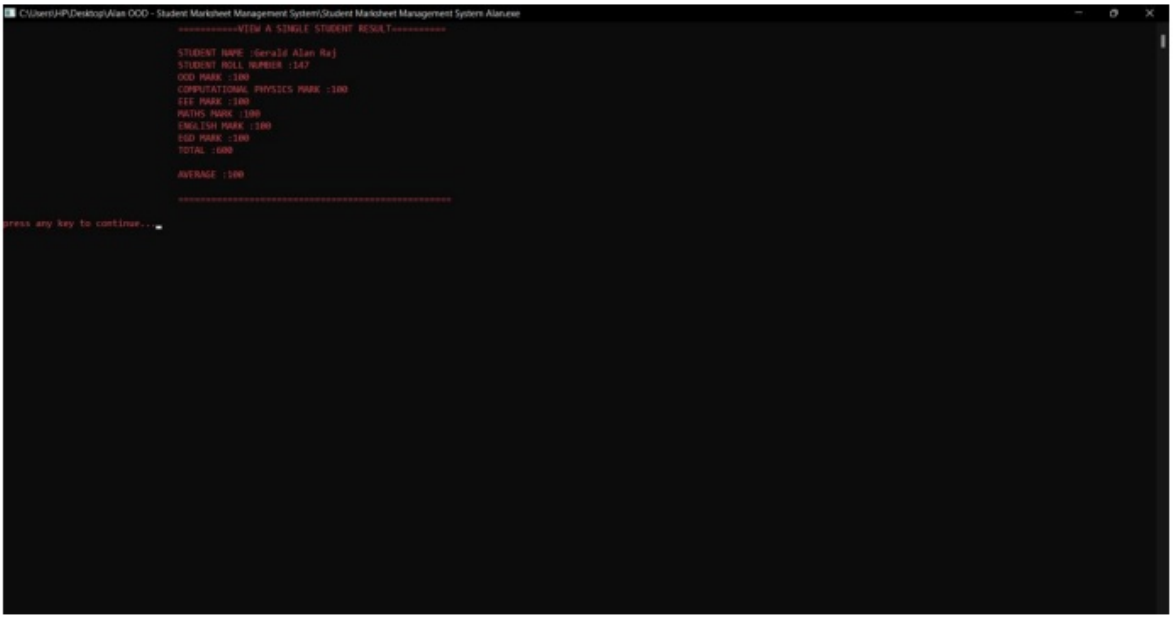
30


C:\Users\HP\Desktop\Alan OOD - Student Marksheet Management System\Student Marksheet Management System Alan.exe
ALL STUDENTS REPORT CARDS
STUDENT NAME :Gerald Alan Raj
STUDENT ROLL NUMBER :147
OOD MARK :100
COMPUTATIONAL PHYSICS MARK :100
EEE MARK :99
MATHS MARK :100
ENGLISH MARK :99
EGD MARK :99
SUM :597
AVERAGE :99.5
STUDENT NAME :Dhevpak
STUDENT ROLL NUMBER :159
OOD MARK :99
COMPUTATIONAL PHYSICS MARK :99
EEE MARK :99
MATHS MARK :99
ENGLISH MARK :99
EGD MARK :99
SUM :594
AVERAGE :99
STUDENT NAME :Avinash
STUDENT ROLL NUMBER :166
OOD MARK :98
COMPUTATIONAL PHYSICS MARK :98
EEE MARK :98
MATHS MARK :96
ENGLISH MARK :98
EGD MARK :98
SUM :586
AVERAGE :97.6667
STUDENT NAME :Rohinth
STUDENT ROLL NUMBER :178
OOD MARK :96
COMPUTATIONAL PHYSICS MARK :96
EEE MARK :99
MATHS MARK :96

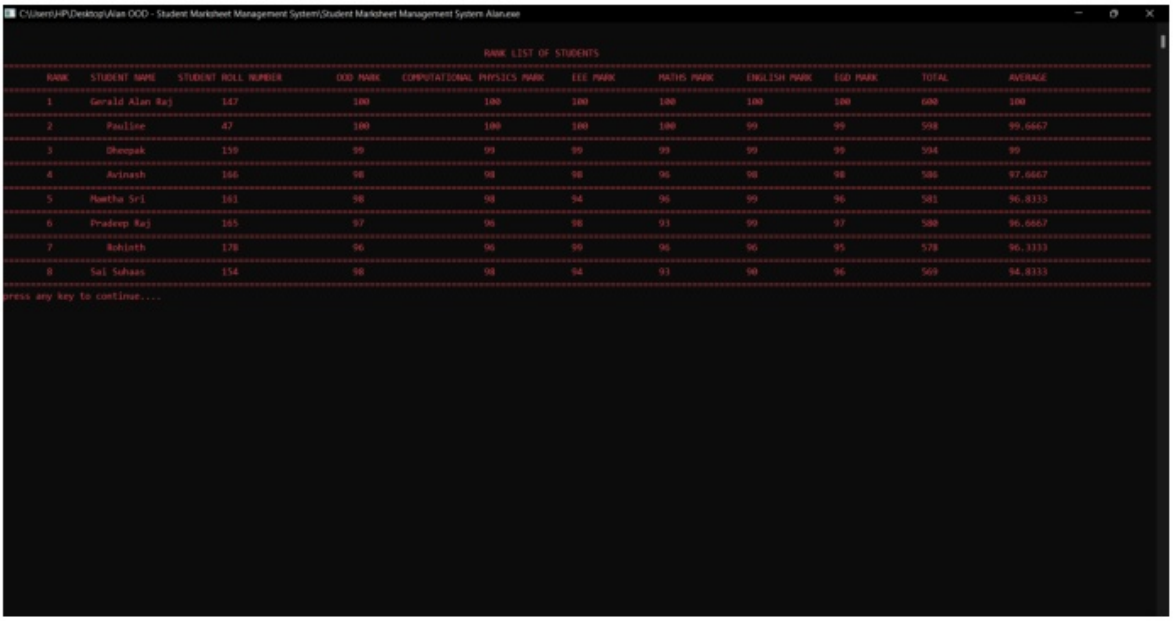

C:\Users\HP\Desktop\Alan OOD - Student Marksheet Management System\Student Marksheet Management System Alan.exe
==========VIEW A SINGLE STUDENT REPORT==========
STUDENT NAME :Gerald Alan Raj
STUDENT ROLL NUMBER :147
OOD MARK :100
COMPUTATIONAL PHYSICS MARK :100
EEE MARK :99
MATHS MARK :100
ENGLISH MARK :99
EGD MARK :99
TOTAL :597
AVERAGE :99.5
press any key to continue...

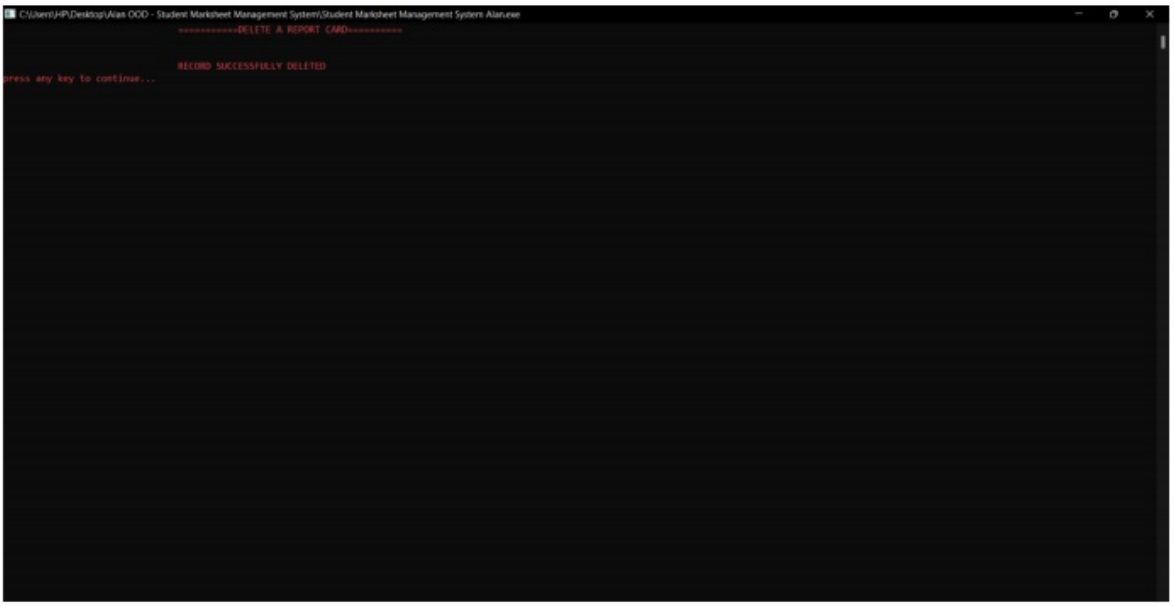
31


C:\Users\HP\Desktop\Alan OOD - Student Marksheet Management System\Student Marksheet Management System Alan.exe
==========MODIFY A REPORT CARD==========
STUDENT NAME :Gerald Alan Raj
STUDENT ROLL NUMBER :147
OOD MARK :100
COMPUTATIONAL PHYSICS MARK :100
EEE MARK :99
MATHS MARK :100
ENGLISH MARK :99
EGD MARK :99
ENTER THE NEW INFORMATION
ENTER YOUR FULL NAME :Gerald Alan Raj
ENTER YOUR ROLL NUMBER :147
ENTER YOUR OOD MARK :100
ENTER YOUR COMPUTATIONAL PHYSICS MARK :100
ENTER YOUR EEE MARK :100
ENTER YOUR MATHS MARK :100
ENTER YOUR ENGLISH MARK :100
ENTER YOUR EGD MARK :100
THE FILE IS SUCCESSFULLY updated
press any key to continue...


C:\Users\HP\Desktop\Alan OOD - Student Marksheet Management System\Student Marksheet Management System Alan.exe
==========VIEW A SINGLE STUDENT RESULT==========
STUDENT NAME :Gerald Alan Raj
STUDENT ROLL NUMBER :147
OOD MARK :100
COMPUTATIONAL PHYSICS MARK :100
EEE MARK :100
MATHS MARK :100
ENGLISH MARK :100
EGD MARK :100
TOTAL :600
AVERAGE :100
press any key to continue...

32


C:\Users\HP\Desktop\Aian OOD - Student Marksheet Management System\Student Marksheet Management System Aian.exe
RANK LIST OF STUDENTS
RANK STUDENT NAME STUDENT ROLL NUMBER OOD MARK COMPUTATIONAL PHYSICS MARK EEE MARK MATHS MARK ENGLISH MARK EGD MARK TOTAL AVERAGE
1 Gerald Alan Raj 147 100 100 100 100 100 100 600 100
2 Pauline 47 100 100 100 100 99 99 598 99.6667
3 Dheepak 159 99 99 99 99 99 99 594 99
4 Avinash 166 98 98 98 96 98 98 586 97.6667
5 Mamtha Sri 161 98 98 94 96 99 96 581 96.8333
6 Pradeep Raj 165 97 98 98 93 99 97 580 96.6667
7 Rohinth 178 96 96 99 96 96 95 578 96.3333
8 Sai Suhaas 154 98 98 94 93 90 96 569 94.8333
press any key to continue....


C:\Users\HP\Desktop\Aian OOD - Student Marksheet Management System\Student Marksheet Management System Aian.exe
==========DELETE A REPORT CARD==========
RECORD SUCCESSFULLY DELETED
press any key to continue...

33

Conclusion and Result:

The Student Marksheet Management System is an efficient and user-friendly software application designed to manage and store the academic records of students. The primary objective of this system is to automate the process of managing the marks of the students and generating their marksheet. The system provides several functionalities such as managing student information, academic records, mark sheet generation.The system is implemented in C++ programming language, which makes it easily maintainable and extensible.

The implementation of the Student Marksheet Management System has successfully met the objectives and requirements of the project. The system provides a simple and effective way of managing and maintaining student records. It enables the users to store and retrieve the student's academic data accurately and efficiently. The system generates mark sheets and progress reports for the students, based on the data stored in the system.

The system also provides several reporting functionalities, such as class averages, student's rankings, etc. that help the teachers and administrators in analyzing the academic performance of students.

Overall, the Student Marksheet Management System is a reliable and robust solution for managing the academic records of students. It reduces the manual workload of teachers and administrators, making the entire process more efficient and error-free.

References:

https://www.geeksforgeeks.org/c-plus-plus/
https://docs.staruml.io/

34

35