

**STUDENT DATA MANAGEMENT
PROJECT REPORT**

Submitted by

A. Sree Dharma Sasta Rao (RA2211026010162)

Under the Guidance of

Dr. Prithi Samuel

Assistant Professor, CINTEL

In partial satisfaction of the requirements for the degree of

**BACHELOR OF TECHNOLOGY
in
COMPUTER SCIENCE ENGINEERING**

with specialization in Artificial Intelligence & Machine Learning

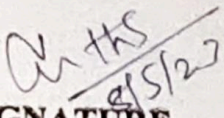


**SCHOOL OF COMPUTING
COLLEGE OF ENGINEERING AND TECHNOLOGY
SRM INSTITUTE OF SCIENCE AND TECHNOLOGY
KATTANKULATHUR - 603203
MAY 2023**

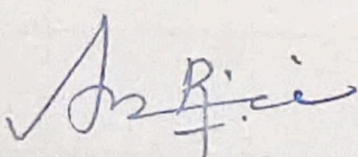
**SRM INSTITUTION OF SCIENCE AND TECHNOLOGY
KATTANKULATHUR-603203**

BONAFIDE CERTIFICATE

Certified that this Project Report titled "Student Data Management System" is the bonafide work done by **A. Sree Dharma Sasta Rao (RA2211026010162)** 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. Prithi Samuel
OODP – Course Faculty
Assistant professor
Department of CINTEL
SRMIST


SIGNATURE

Dr. R. Annle uthra
Head of the Department
Department of CINTEL
School of computing
SRMIST

TABLE OF CONTENTS

| S.No | CONTENTS | PAGE NO |
|-------------|--------------------------|----------------|
| 1. | Problem Statement | 4 |
| 2. | Modules of Project | 4 |
| 3. | Diagrams | 5-12 |
| | a. Use case Diagram | 5 |
| | b. Class Diagram | 6 |
| | c. Sequence Diagram | 7 |
| | d. Collaboration Diagram | 8 |
| | e. State Chart Diagram | 9 |
| | f. Activity Diagram | 10 |
| | g. Package Diagram | 11 |
| | h. Deployment Diagram | 12 |
| 4. | Code/Output Screenshots | 13-22 |
| 5. | Conclusion and Results | 23 |
| 6. | References | 23 |

1.Problem statement:

In educational institutions, managing student data is a crucial and time-consuming task. The current system of managing student data using manual records and spreadsheets is often error-prone and inefficient. This results in data redundancy, inconsistency, and inaccuracies, leading to delayed decision-making and hampered institutional growth.

To address these issues, a student data management system needs to be implemented, which can efficiently manage student data, including personal information, academic records, attendance, and other related information. The system should allow easy access and modification of data by authorized personnel, maintain data accuracy and consistency, and ensure data security and privacy. Additionally, the system should generate reports and analytics to provide insights into student performance and institutional growth.

2. Modules of project:

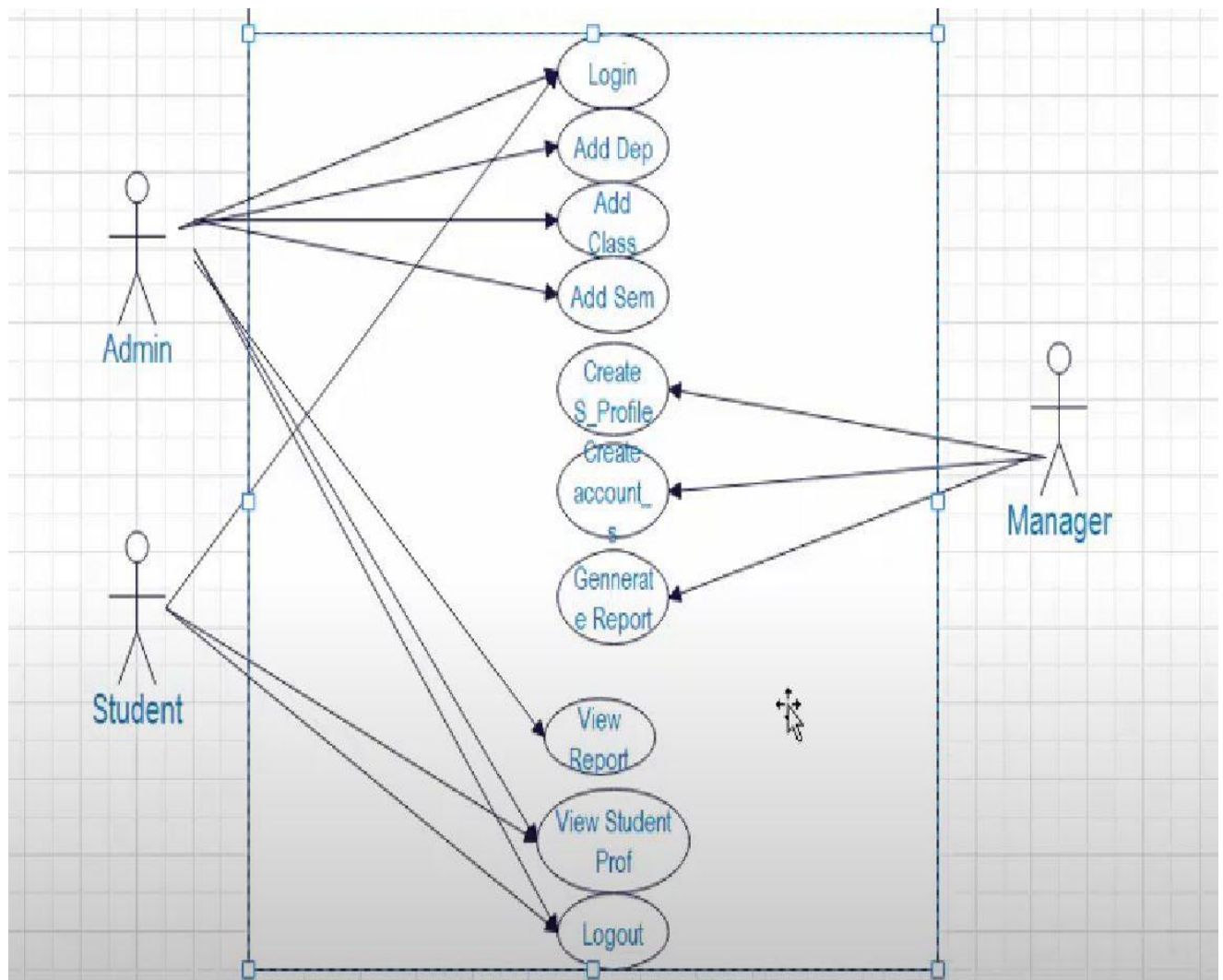
It allows users to add new students, login as a student, faculty member, proctor, or administrator, and view and add marks to the database of a student. The program uses file input/output to store and retrieve data from a text file named "Example.txt". The necessary libraries included in the code are:

- <fstream>: for file input/output operations
- <iostream>: for standard input/output operations
- <stdio.h>: for standard input/output operations
- <string.h>: for string operations

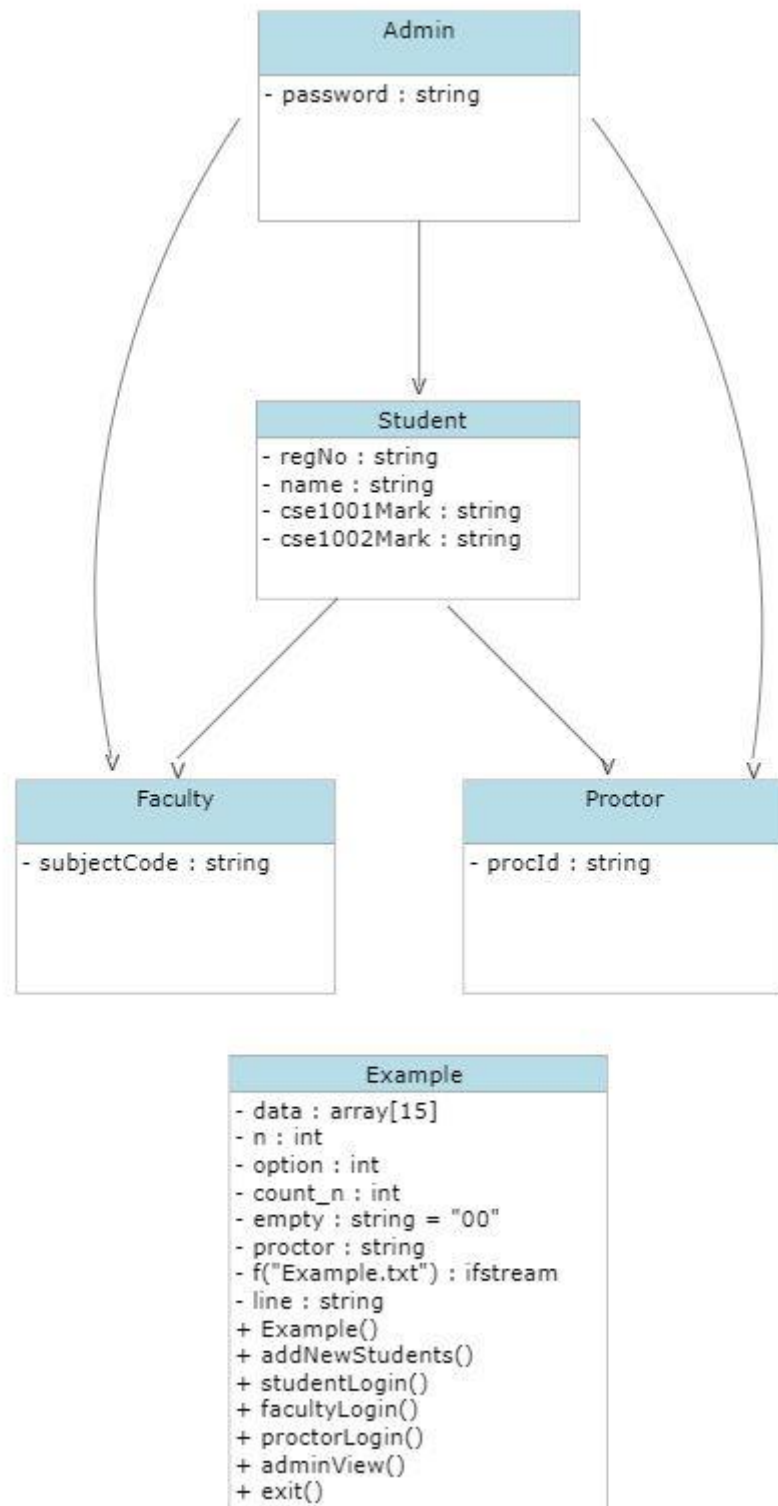
The program reads and writes data to the text file in a structured manner, using tabs and newlines to separate different fields of data. The program also uses a few variables to keep track of various things, such as the total number of lines in the file (which corresponds to the number of students in the database), the initial mark allotted to a subject, and the proctor ID of each student.

3. Diagrams

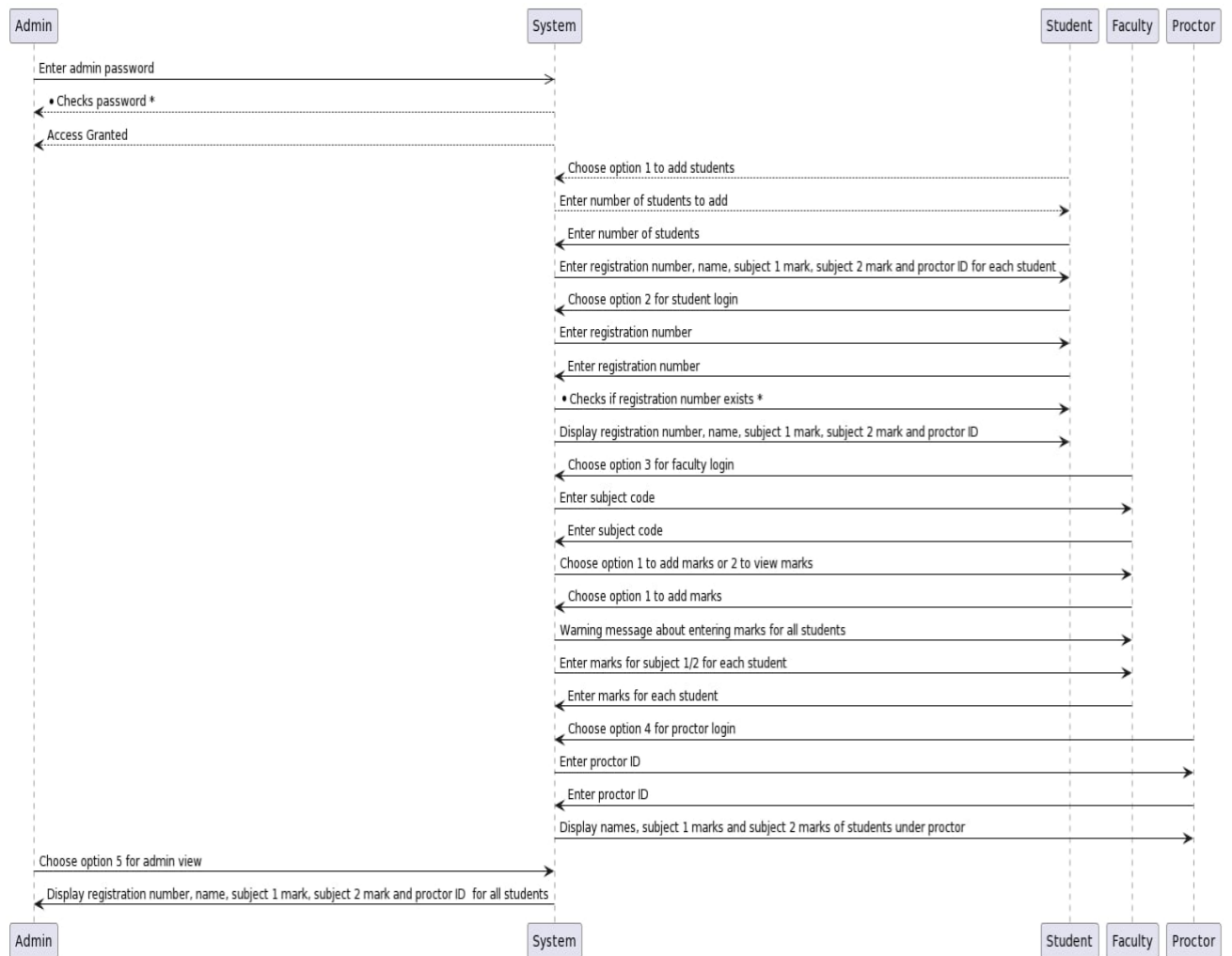
a. Use Case Diagram



b. Class diagram

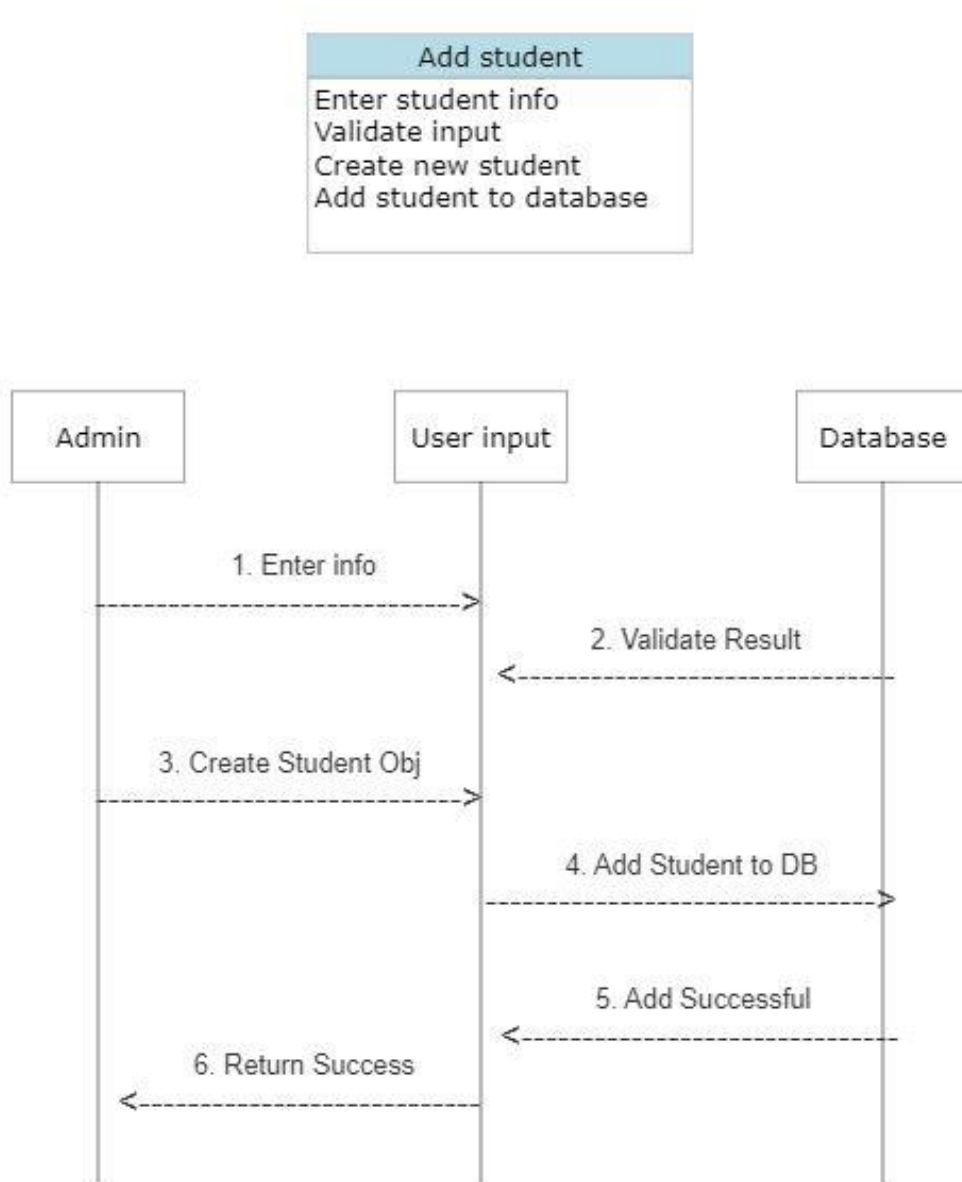


c. Sequence diagram



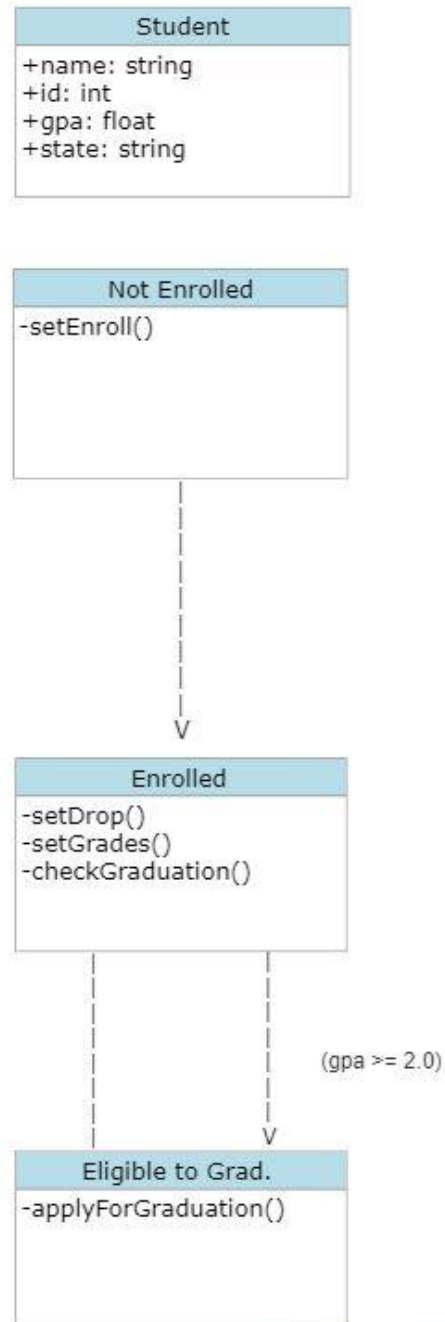
d. Collaboration Diagram

Collaboration Diagram

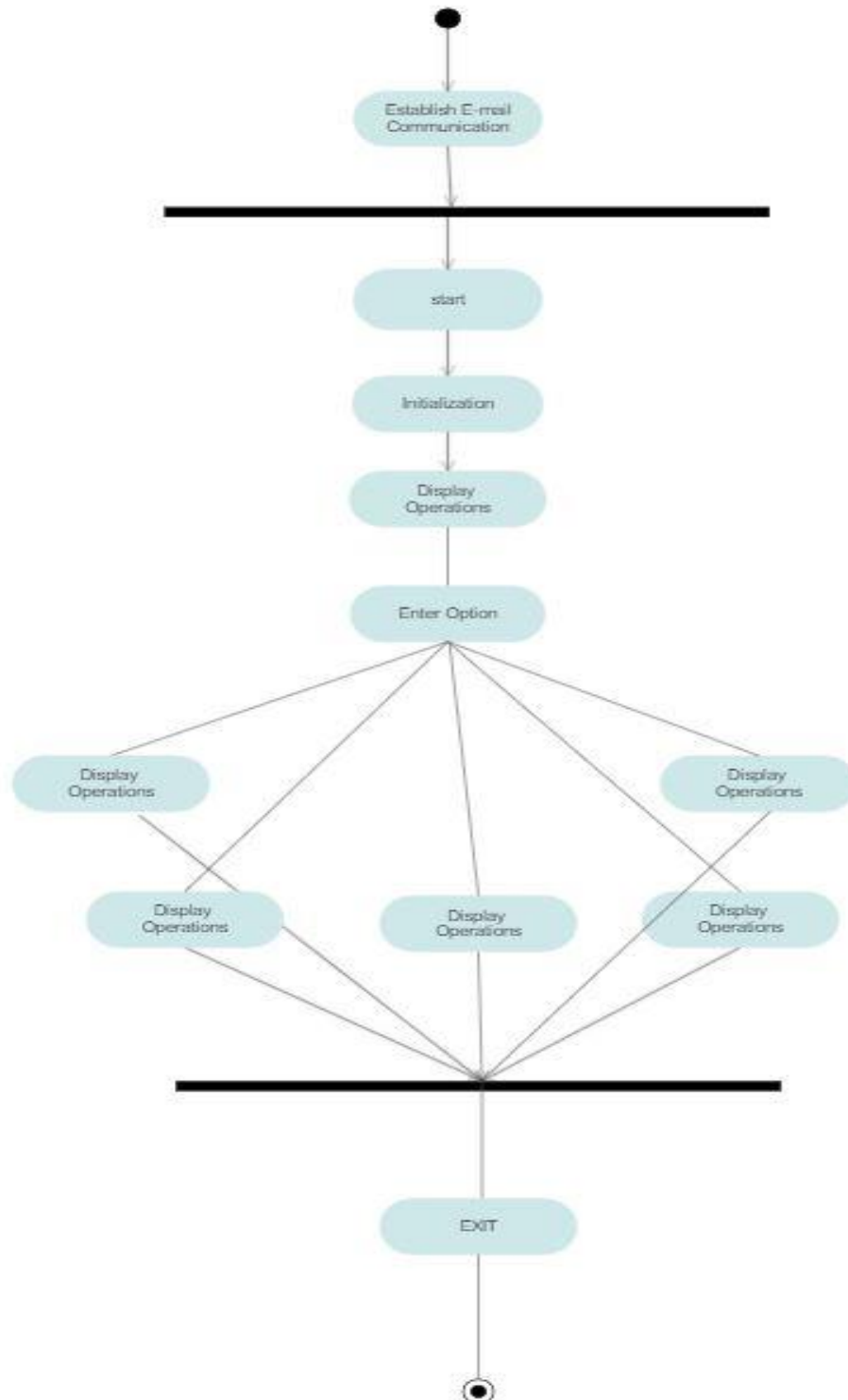


e. State Chart Diagram

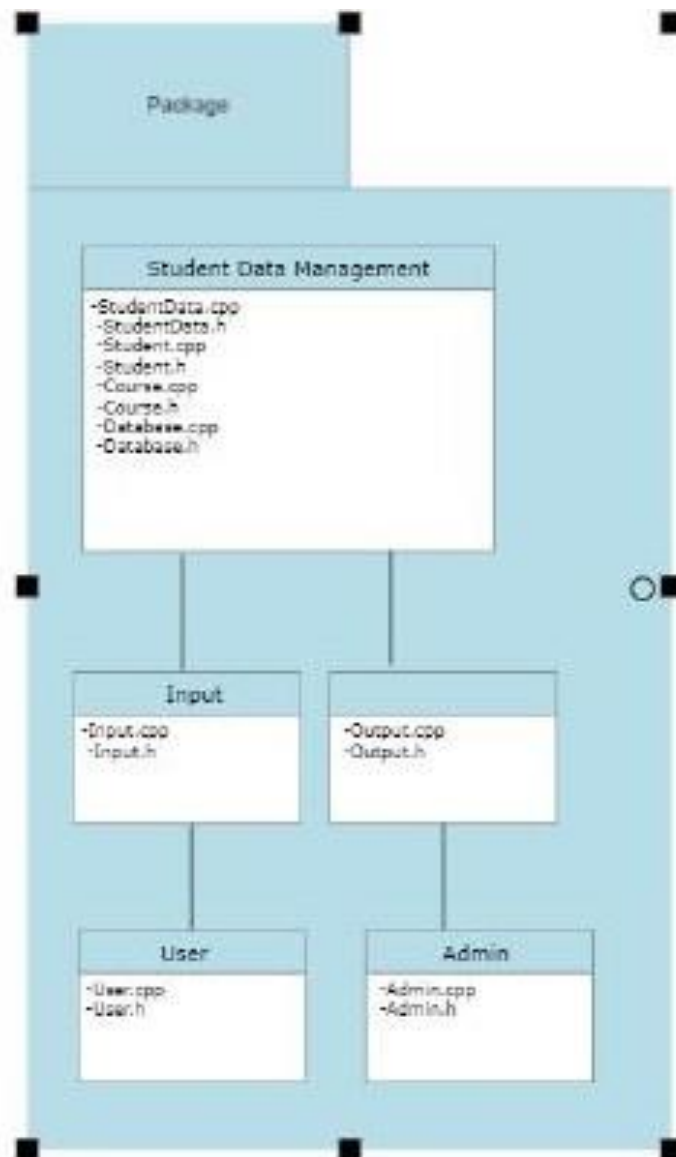
State chart Diagram



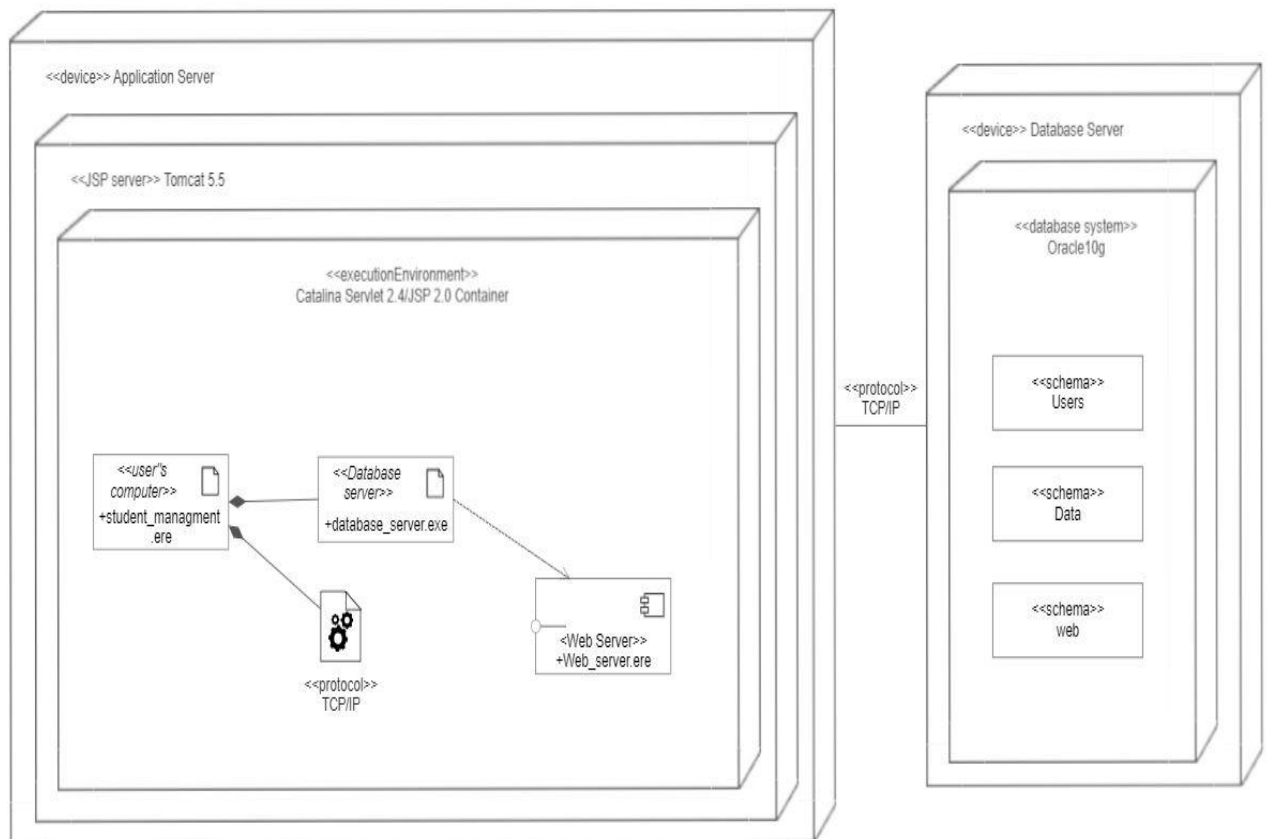
f. Activity Diagram



g. Package Diagram



h. Deployment Diagram



4.Code/Output:

Code:

```
// Include all the necessary libraries.
#include <fstream>
#include <iostream>
#include <stdio.h>
#include <string.h>
using namespace std;
int main()
{
    // Considering the max length of data entered (name) to
    // be 15.
    char data[15];
    int n = 0, option = 0, count_n = 0;
    // This is the initial mark allotted to a subject.
    string empty = "00";
    string proctor = "";
    // Name of the file in which DB is stored.
    ifstream f("Example.txt");
    string line;

    // The following for loop counts the total number of
    // lines in the file.
    for (int i = 0; std::getline(f, line); ++i)
    {
        count_n++;
    }

    while (option != 6)
    {
```



```

// This prints out all the available options in the
// DB
cout << "\nAvailable operations: \n1. Add New "
    "Students\n2."
    << " Student Login\n3. Faculty Login\n4. "
    "Proctor Login\n5. Admin View\n"
    << "6. Exit\nEnter option: ";
cin >> option;

if (option == 1)
{
    cout << "Enter the number of students: ";
    cin >> n;

    count_n = count_n + n;

    for (int i = 0; i < n; i++)
    {
        ofstream outfile;
        outfile.open("Example.txt", ios::app);
        // The entire data of a single student is
        // stored line-by-line.
        cout << "Enter your registration number: ";
        cin >> data;
        outfile << data << "\t";

        cout << "Enter your name: ";
        cin >> data;
        int len = strlen(data);

        while (len < 15)
        {
            data[len] = ' ';

```

```

        len = len + 1;
    }
    outfile << data << "\t";
    // Inserting empty data initially into the
    // file
    outfile << empty << "\t";
    outfile << empty << "\t";

    cout << "Enter your proctor ID: ";
    cin >> proctor;

    outfile << proctor << endl;
}
}

else if (option == 2)
{
    char regno[9];
    cout << "Enter your registration number: ";
    cin >> regno;
    ifstream infile;
    int check = 0;
    infile.open("Example.txt", ios::in);

    // This loop prints out the data according to
    // the registration number specified.
    while (infile >> data)
    {
        if (strcmp(data, regno) == 0)
        {
            cout
                << "\nRegistration Number: " << data
                << endl;

```

```

infile >> data;

cout << "Name: " << data << endl;


infile >> data;

cout << "CSE1001 mark: " << data
    << endl;


infile >> data;

cout << "CSE1002 mark: " << data
    << endl;


infile >> data;

cout << "Proctor ID: " << data << endl;


infile.close();

check = 1;
}
}

if (check == 0)
{
    cout << "No such registration number found!"
        << endl;
}
}

// This loop is used to view and add marks to the
// database of a student.
else if (option == 3)
{
    char subcode[7];
    cout << "Enter your subject code: ";
    cin >> subcode;

```

```

string code1 = "CSE1001", code2 = "CSE1002",
    mark = "";
ifstream infile;
int check = 0;

cout << "\nAvailable operations: \n1. Add data "
    "about marks\n"
    << "2. View data\nEnter option: ";
cin >> option;

if (option == 1)
{
    cout
        << "Warning! You would need to add mark"
        << "details for all the students!"
        << endl;
    for (int i = 0; i < count_n; i++)
    {
        fstream file("Example.txt");

        // The seek in file has been done
        // according to the length
        // of the data being inserted. It needs
        // to adjusted accordingly for different
        // lengths of data.

        if (strcmp(subcode, code1.c_str()) == 0)
        {
            file.seekp(26 + 37 * i,
                std::ios_base::beg);
            cout << "Enter the mark of student#"
                << (i + 1) << " : ";
            cin >> mark;

```

```

        file.write(mark.c_str(), 2);
    }

    if (strcmp(subcode, code2.c_str()) == 0)
    {
        file.seekp(29 + 37 * i,
            std::ios_base::beg);
        cout << "Enter the mark of student#"
            << (i + 1) << " : ";
        cin >> mark;
        file.write(mark.c_str(), 2);
    }
}

// This loop is used to view marks of a student.
// The extra infile commands have been used to
// get a specific mark only since the data has
// been separated by a tabspace.

else if (option == 2)
{
    infile.open("Example.txt", ios::in);
    if (strcmp(subcode, code1.c_str()) == 0)
    {
        cout << "Registration number - Marks\n"
            << endl;
        while (infile >> data)
        {
            cout << data;
            infile >> data;
            infile >> data;
            cout << " - " << data << endl;
        }
    }
}

```



```

        infile >> data;
        infile >> data;
        check = 1;
    }
}

infile.close();
infile.open("Example.txt", ios::in);

if (strcmp(subcode, code2.c_str()) == 0)
{
    cout << "Registration number - Marks\n"
        << endl;
    while (infile >> data)
    {
        cout << data;
        infile >> data;
        infile >> data;
        infile >> data;
        cout << " - " << data << endl;
        infile >> data;
        check = 1;
    }
}

infile.close();

if (check == 0)
{
    cout << "No such subject code found!"
        << endl;
}

```

```

}

// This loop displays all the details of students
// under the same proctor ID.

else if (option == 4)
{
    char procid[7];
    cout << "Enter your proctor ID: ";
    cin >> procid;
    int check = 0;
    char temp1[100], temp2[100], temp3[100];
    char temp4[100], id[100];
    ifstream infile;
    infile.open("Example.txt", ios::in);

    while (infile >> temp1)
    {
        infile >> temp2;
        infile >> temp3;
        infile >> temp4;
        infile >> id;

        if (strcmp(id, procid) == 0)
        {
            cout << "\nRegistration Number: "
                << temp1 << endl;
            cout << "Name: " << temp2 << endl;
            cout << "CSE1001 Mark: " << temp3
                << endl;
            cout << "CSE1002 Mark: " << temp4
                << endl;
            check = 1;

```

```

    }
}

if (check == 0)
{
    cout << "No such proctor ID found!" << endl;
}
}

// This loop acts as an admin view to see all the
// data in the file.

else if (option == 5)
{
    char password[25];
    cout << "Enter the admin password: ";
    cin >> password;

    // This variable value can be changed according
    // to your requirement of the administrator
    // password.

    string admin_pass = "admin";

    if (strcmp(password, admin_pass.c_str()) == 0)
    {
        cout << "Reg No.  "
            "\tName\tCSE1001\tCSE1002\tProctor "
            "ID"
            << endl;

        ifstream infile;
        infile.open("Example.txt", ios::in);
        char data[20];

```


5.Conclusion:

Therefore, the objective of this project is to develop an automated student data management system that streamlines and simplifies the management of student data, making it more accurate, secure, and accessible.

6.References:

<https://cplusplus.com/>