A Project Report
On

Student Information System (SIS) - Student Record Management Software

Submitted to

KISCIT UNIVERSITY, KAHUTA

KISCIT

UNIVERSITY

In partial fulfillment of the requirements for the award of the degree of

BACHELOR OF TECHNOLOGY In SOFTWARE ENGINEERING

Submitted by

Naila Shehzadi (222101008)
Laiba Pervaiz (222101002)
Iqra Matloob (222101004)

Under the guidance of

SIR Sohail Ahmed

DEPARTMENT OF SOFTWARE ENGINEERING
KISCIT UNIVERSITY DR. A. Q. KHAN INSTITUTE OF
COMPUTER SCIENCE INFORMATION TECHNOLOGY

Q1 – Analyze the complete model of Student Information System (SIS) and its development?

1. Attendance Management
2. Admission Management
3. Fee Management
4. Document Management

Q2 – Analyze the C++ programming by develop the Student Information System (SIS) based on given requirements:

1. Current Status of Students (In-studies, Left, Pass out, etc.)
2. Admission Management
3. Managing Record of admitted Students
4. Fee Management Record
5. Attendance Management
6. Students Results and Performance Record
7. Students Health Fitness and Disability Record
8. Students Participation and Extra Curricular Activities Record

Project:

Student Information System (SIS) - Student Record Management Software:

Introduction: The Student Information System (SIS) or Student Record Management Software is designed to automate and streamline various administrative tasks in educational institutions. It provides efficient management of student-related information, including attendance, admission, fees, documents, results, health records, and extracurricular activities. This project aims to develop a Student Information System using C++ programming language, incorporating the specified functionalities.

Q1 – Analysis of the complete model of Student Information System (SIS) and its development:

The Student Information System (SIS) is a comprehensive software solution that simplifies administrative tasks and improves the efficiency of educational institutions. It comprises several modules that handle different aspects of student management. The development of an SIS involves designing and implementing these modules to provide a seamless experience for administrators, teachers, students, and parents. Here is an analysis of the key modules:

Attendance Management: This module allows teachers to mark and view attendance for students. It also enables sharing attendance status with parents. It includes features like generating attendance reports, tracking staff attendance, and integrating with biometric systems if available.

Admission Management: The admission management module automates the entire admission process. It facilitates online form submission, fee payment, batch allotment, and document submission. It ensures quick and accurate collection of student information during the admission process.

Managing Record of Admitted Students: This module focuses on maintaining a comprehensive record of admitted students. It includes their personal details, contact information, parents' details, academic records, and any other relevant information. It ensures easy retrieval and updating of student records.

Fee Management Record: The fee management module handles tasks related to fee collection, record-keeping, receipt generation, dues management, and refunds. It allows administrators to define fee cycles and provides reports on fee-related information.

Attendance Management: Similar to the teacher attendance management, this module allows administrators to track the attendance of staff members. It includes features like leave management, attendance reports, and integration with payroll systems.

Students Results and Performance Record: This module focuses on recording and managing students' academic performance and results. It allows teachers to enter and update marks,

generate report cards, and provide performance insights. It may include features like grade calculation and graphical representation of results.

Students Health Fitness and Disability Record: This module handles the storage and management of students' health-related information, including medical history, allergies, disabilities, and fitness records. It ensures easy access to health information for timely interventions and supports.

Students Participation and Extra Curricular Activities Record: This module enables tracking and documenting students' participation in extracurricular activities such as sports, clubs, events, and competitions. It helps in recognizing achievements, promoting talent, and organizing related activities.

Q2 – Development of Student Information System (SIS) using C++ programming language:

To develop the Student Information System (SIS) using C++, you need to implement the following functionalities:

Current Status of Students: Implement a feature to track the current status of students, such as whether they are in-studies, left the institution, or graduated.

Admission Management: Design a module that handles the admission process, including online form submission, fee payment, batch allotment, and document submission.

Managing Record of Admitted Students: Create a data structure or database to store and manage the records of admitted students. Include fields for personal information, contact details, parents' information, academic records, etc.

Fee Management Record: Develop a module that manages fee-related tasks, such as fee collection, record-keeping, receipt generation, dues management, and refunds. Consider implementing a fee cycle management system.

Attendance Management: Implement features for teachers to mark and view attendance for students. Provide options to share attendance status with parents. Also, develop the functionality to track staff attendance.

Students Results and Performance Record: Create a system to record and manage students' academic results and performance. Allow teachers to enter and update marks, generate report cards, and provide performance insights.

Students Health Fitness and Disability Record: Design a module to store and manage students' health-related information, including medical history, allergies, disabilities, and fitness records. Ensure proper data handling and privacy measures.

Students Participation and Extra Curricular Activities Record: Develop a feature that allows tracking and documenting students' participation in extracurricular activities. Provide options to record their achievements, manage clubs or teams, and organize related activities.

```cpp
#include <iostream>
#include <cstring>

const int MAX_STUDENTS = 2;

struct Student {
    int id;
    string name;
    string status;

    string admissionDate;
    string batch;
    string documentSubmission;

    double totalFees;
    double feesPaid;
    double feesDue;

    int attendance;

    double averageScore;

    string healthStatus;
    string disabilityStatus;

    string participationDetails;
};

void addStudent(Student students[], int& numStudents) {
    if (numStudents >= MAX_STUDENTS) {
        std::cout << "Maximum number of students reached." << std::endl;
        return;
```

```
}

Student newStudent;
newStudent.id = numStudents + 1;

cout << "Enter student name: ";
cin.ignore();
getline(std::cin, newStudent.name);

cout << "Enter student status (In-studies, Left, Pass out, etc.): ";
getline(std::cin, newStudent.status);


cout << "Enter admission date: ";
getline(std::cin, newStudent.admissionDate);

cout << "Enter batch: ";
getline(std::cin, newStudent.batch);

cout << "Enter document submission status: ";
getline(std::cin, newStudent.documentSubmission);


cout << "Enter total fees: ";
cin >> newStudent.totalFees;

cout << "Enter fees paid: ";
cin >> newStudent.feesPaid;

newStudent.feesDue = newStudent.totalFees - newStudent.feesPaid;
```

```
    cout << "Enter attendance: ";
    cin >> newStudent.attendance;


    cout << "Enter average score: ";
    cin >> newStudent.averageScore;

    cout << "Enter health status: ";
    cin.ignore();
  getline(std::cin, newStudent.healthStatus);

    cout << "Enter disability status: ";
    getline(std::cin, newStudent.disabilityStatus);


    cout << "Enter participation details: ";
    getline(std::cin, newStudent.participationDetails);

    students[numStudents] = newStudent;
    numStudents++;

    cout << "Student added successfully." << std::endl;
}

void displayStudents(const Student students[], int numStudents) {
    if (numStudents == 0) {
        cout << "No students found." << std::endl;
        return;
    }

   cout << "ID\tName\t\tStatus" << std::endl;
    for (int i = 0; i < numStudents; i++) {
```

```
void searchStudent(const Student students[], int numStudents, int studentId) {
    for (int i = 0; i < numStudents; i++) {
        if (students[i].id == studentId) {
            cout << "ID: " << students[i].id << std::endl;
            cout << "Name: " << students[i].name << std::endl;
            cout << "Status: " << students[i].status << std::endl;

            cout << "Admission Date: " << students[i].admissionDate << std::endl;
            cout << "Batch: " << students[i].batch << std::endl;
            std::cout << "Document Submission: " << students[i].documentSubmission << std::endl;
            std::cout << "Total Fees: " << students[i].totalFees << std::endl;
            std::cout << "Fees Paid: " << students[i].feesPaid << std::endl;
            std::cout << "Fees Due: " << students[i].feesDue << std::endl;
            std::cout << "Attendance: " << students[i].attendance << std::endl;
            std::cout << "Average Score: " << students[i].averageScore << std::endl;
            std::cout << "Health Status: " << students[i].healthStatus << std::endl;
            std::cout << "Disability Status: " << students[i].disabilityStatus << std::endl;
            std::cout << "Participation Details: " << students[i].participationDetails << std::endl;
            return;
        }
    }
 cout << "Student with ID " << studentId << " not found." << std::endl;
}

int main() {
    Student students[MAX_STUDENTS];
    int numStudents = 0;

    int choice;
```

```
    int studentId;

    do {
        std::cout << "Student Information System" << std::endl;
        std::cout << "1. Add Student" << std::endl;
        std::cout << "2. Display Students" << std::endl;
        std::cout << "3. Search Student" << std::endl;
        std::cout << "4. Quit" << std::endl;
        std::cout << "Enter your choice: ";
        std::cin >> choice;

        switch (choice) {
            case 1:
                addStudent(students, numStudents);
                break;
            case 2:
                displayStudents(students, numStudents);
                break;
            case 3:
                std::cout << "Enter student ID to search: ";
                std::cin >> studentId;
                searchStudent(students, numStudents, studentId);
                break;
            case 4:
                std::cout << "Exiting program." << std::endl;
                break;
            default:
                std::cout << "Invalid choice. Please try again." << std::endl;
        }

        std::cout << std::endl;
    } while (choice != 4);


    return 0;
}
```

OUTPUT:

```
Student Information System
1. Add Student
2. Display Students
3. Search Student
4. Quit
```

```
Enter your choice: 4
```

```
Enter your choice: 1
Enter student name: naila
Enter student status (In-studies, Left, Pass out, etc.): in studeis
Enter admission date: 2-5-22
Enter batch: 2022
Enter document submission status: Submitted
Enter total fees: 20000
Enter fees paid: 15000
Enter attendance: 90
Enter average score: 85
Enter health status: Satisfied
Enter disability status: Nill
Enter participation details: Sports
Student added successfully.
```

```
Enter your choice: 2
ID      Name            Status
1       naila           in studeis
```

```
Enter your choice: 3
Enter student ID to search: 1
ID: 1
Name: naila
Status: in studeis
Admission Date: 2-5-22
Batch: 2022
Document Submission: Submitted
Total Fees: 20000
Fees Paid: 15000
Fees Due: 5000
Attendance: 90
Average Score: 85
Health Status: Satisfied
Disability Status: Nill
Participation Details: Sports
```