JAVA PROJECT REPORT
ON
STUDENT ATTENDANCE MANAGEMENT SYSTEM

Partial fulfilment of

***B. Tech 2nd year* (Computer Science & Engineering)**

Submitted To:

Dr.Shaloo Dadheech
(Associate Professor)

Submitted by:

Prem Kumar (B3)
Roll No. 22EJICS118

Acknowledgment

I would like to acknowledge the contributions of the following people without whose help and guidance this project would not have been completed.

I am thankful to Mrs. Mamta Garg, HOD(Mentor) and Mrs. Anju Jangid, HOD(Academic) of Computer Science and Engineering Department, Jodhpur Institute of Engineering and Technology, for her constant encouragement, valuable suggestions and moral support and blessings.

I respectfully thank Dr. Shaloo Dadheech (Associate Professor) of Computer Science and Engineering Department, Jodhpur Institute of Engineering and Technology, for providing me an opportunity to do this project work and giving me all support and guidance, which made me complete the project up to very extent.

Although it is not possible to name individually, we shall ever remain indebted to the faculty members of Jodhpur Institute of Engineering and Technology, for their persistent support and cooperation extended during his work.

This acknowledgement will remain incomplete if I fail to express my deep sense of obligation to my parents and God for their consistent blessings and Encouragement.

Table of Contents

Introduction

The Attendance Management System (AMS) is a Java-based console application designed to manage and track the attendance of students. It provides functionality to add students, mark their attendance, and generate attendance reports.

Technology Used In Project

1. Computer System
2. Java Programming language

Requirements

1. System must have windows 7 or above.

Features

1. Student Management

The system allows the addition of students with their names and unique roll numbers.

2. Attendance Tracking

The system keeps track of each student's attendance status (present or absent).

3. User Interface

The user interacts with the system through a simple console-based menu.

4. Attendance Report

The system generates a report displaying the names, roll numbers, and attendance status of all students.

Project Code

```
import java.util.*;

class Student {
    private String name;
    private int rollNo;
    private boolean present;

    public Student(String name, int rollNo) {
        this.name = name;
        this.rollNo = rollNo;
        this.present = false;
    }

    public String getName() {
        return name;
    }

    public int getRollNo() {
        return rollNo;
    }
```

```
    public boolean isPresent() {
        return present;
    }

    public void markAttendance() {
        this.present = true;
    }
}

class AttendanceManager {
    private List<Student> students;

    public AttendanceManager() {
        students = new ArrayList<>();
    }

    public void addStudent(Student student) {
        students.add(student);
    }

    public void markStudentPresent(int rollNo) {
        for (Student student : students) {
            if (student.getRollNo() == rollNo) {
                student.markAttendance();
                System.out.println(student.getName() + " is marked present.");
                return;
            }
        }
        System.out.println("Student with roll number " + rollNo + " not
found.");
    }

    public void displayAttendance() {
        System.out.println("Attendance Report:");
        for (Student student : students) {
            System.out.println("Name: " + student.getName() + ", Roll No: " +
student.getRollNo() + ", Present: " + (student.isPresent() ? "Yes" : "No"));
        }
    }
}

public class AttendanceManagementSystem {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        AttendanceManager attendanceManager = new AttendanceManager();

        // Adding some sample students
        attendanceManager.addStudent(new Student("Prem kumar", 101));
        attendanceManager.addStudent(new Student("Ayush kr.", 102));
        attendanceManager.addStudent(new Student("Aryan", 103));
```

```
        boolean exit = false;

        while (!exit) {
            System.out.println("\nMenu:");
            System.out.println("1. Mark Attendance");
            System.out.println("2. Display Attendance");
            System.out.println("3. Exit");
            System.out.print("Enter your choice: ");

            int choice = scanner.nextInt();

            switch (choice) {
                case 1:
                    System.out.print("Enter the roll number of the student to
mark present: ");
                    int rollNo = scanner.nextInt();
                    attendanceManager.markStudentPresent(rollNo);
                    break;
                case 2:
                    attendanceManager.displayAttendance();
                    break;
                case 3:
                    exit = true;
                    break;
                default:
                    System.out.println("Invalid choice. Please try again.");
            }
        }

        scanner.close();
    }
}
```

Example Usage

```
Menu:
1. Mark Attendance
2. Display Attendance
3. Exit
Enter your choice: 1
Enter the roll number of the student to mark present: 101
John Doe is marked present.

Menu:
1. Mark Attendance
2. Display Attendance
3. Exit
```

```
Enter your choice: 2
Attendance Report:
Name: John Doe, Roll No: 101, Present: Yes
Name: Jane Doe, Roll No: 102, Present: No
Name: Bob Smith, Roll No: 103, Present: No

Menu:
1. Mark Attendance
2. Display Attendance
3. Exit
Enter your choice: 3
```

Implementation details

1. Student Class

Represents a student with attributes like name, roll number, and attendance status.

2. AttendanceManager Class

Manages the list of students and provides methods for adding students, marking attendance, and displaying the attendance report.

3. Main Class (AttendanceManagementSystem)

Provides the main method for user interaction through a console interface.

Future improvement

1. **User Authentication**:
   - Implement a login system to authenticate users before accessing the system. This adds a layer of security and ensures that only authorized individuals can manage attendance.
2. **Database Integration**:
   - Integrate a database (e.g., MySQL, PostgreSQL) to store student information and attendance records. This allows for persistent data storage and retrieval.
3. **Graphical User Interface (GUI)**:
   - Create a user-friendly GUI using a library like JavaFX or Swing. This provides a more intuitive and visually appealing interface for users.
4. **Date and Time Stamps**:
   - Record the date and time when attendance is marked. This allows for more detailed tracking and reporting, especially in scenarios where multiple sessions occur in a day.
5. **Attendance Reports by Date Range**:

- Allow users to generate attendance reports for specific date ranges. This enables administrators to track attendance trends over time.

6. **Export to PDF/Excel**:
   - Add functionality to export attendance reports to popular file formats like PDF or Excel for easy sharing and printing.
7. **Edit and Delete Student Records**:
   - Provide options to edit or delete student records. This allows for the correction of mistakes or the removal of outdated information.
8. **Search and Filter Options**:
   - Implement search and filter functionalities to quickly locate specific students or attendance records within a large dataset.
9. **Email Notifications**:
   - Set up email notifications to notify students, parents, or administrators of attendance-related events (e.g., low attendance, special events).
10. **Student Profile Information**:
    - Expand the student profile to include additional information like contact details, parent/guardian information, and academic performance.
11. **Multi-Class Support**:
    - Extend the system to support multiple classes or groups, each with its own set of students and attendance records.
12. **Data Analytics and Visualizations**:
    - Provide graphical representations and analytics tools to visualize attendance data trends, such as attendance percentages, patterns, and comparisons.

Conclusion

The Attendance Management System provides a simple yet effective way to manage student attendance. While this example is a console-based application, a real-world implementation would incorporate a database for data persistence and possibly a graphical user interface for improved user experience.

Please note that this report provides an overview of the Attendance Management System. In a real-world scenario, you would likely include additional sections such as system architecture, data flow diagrams, and more detailed explanations of the classes and methods.