Jaypee Institute of Information Technology, Noida

DEPARTMENT OF COMPUTER SCIENCE &
ENGINEERING AND INFORMATION TECHNOLOGY

JAVA Project Synopsis

Project Title : School Management System

Submitted to: Ms. Preeti Mittal

S.No	Name	Enroll No.	Batch
1.	Teesha Kakkar	22103098	B2
2.	Avlokita Singh	22103034	B2
3.	Mehul Bansal	22103048	B2

Program: BTech CSE

Semester and Year: 5th and 3rd Year

Project Overview:-

The **School Management System (SMS)** is a software application designed to handle the administration and management of a school's daily operations. It simplifies tasks such as managing teachers, students, fees, salaries, and the school's finances. The system allows school administrators to track teacher and student details, handle fee payments, salary distribution, and manage the overall financial state of the institution.

The system is implemented using Java, and it makes use of object-oriented principles to design classes for **Teacher**, **Student**, and **School** entities, with methods for handling actions such as paying fees, receiving salaries, and updating the total money earned and spent by the school.

Project Structure:-

This project consists of the following classes:

- **Main Class**: Entry points to the application where objects of `Teacher`, `Student`, and `School` are created, and the main logic of the system is executed.
- **Teacher Class**: Handles the teacher's details such as name, salary, and the salary they receive. It also tracks the total salary earned by the teacher.
- **Student Class**: Handles the student's details such as name, grade, total fees, and fees paid. It also tracks the payments made by students.
- **School Class**: Manages the school's finances and the teacher and student lists. It keeps track of the total money earned and spent by the school.

Class Details:-

❖ Teacher Class:-

The `Teacher` class represents the teachers in the school and contains the following attributes:

- `id`: Unique identifier for each teacher.
- `name`: Name of the teacher.
- `salary`: The salary of the teacher.
- `salaryEarned`: Total salary received by the teacher.

Methods in Teacher Class:

- **getId()**: Returns the teacher's ID.
- **getName()**: Returns the teacher's name.
- **getSalary()**: Returns the teacher's salary.
- **receiveSalary(int salary)**: Adds the salary to the teacher's `salaryEarned` and updates the total money spent by the school.

Teacher Class Implementation:-

```
public class Teacher {

    private int id;
    private String name;
    private int salary;
    private int salaryEarned;
```

❖ Student Class:-

The `Student` class represents the students in the school and contains the following attributes:

- `id`: Unique identifier for each student.
- `name`: Name of the student.
- `grade`: Grade of the student.
- `feesPaid`: Total fees paid by the student.
- `feesTotal`: Total fees the student is supposed to pay.

Methods in Student Class:

- **getId()**: Returns the student's ID.
- **getName()**: Returns the student's name.
- **getGrade()**: Returns the student's grade.
- **payFees(int fees)**: Adds the paid fees to the student's `feesPaid` and updates the total money earned by the school.

Student Class Implementation:-

```
public class Student {

    private int id;
    private String name;
    private int grade;
    private int feesPaid;
    private int feesTotal;
```

❖ School Class:-

The `School` class manages the teachers and students and tracks the school's finances. It contains the following attributes:

- `teachers`: A list of teachers employed by the school.
- `students`: A list of students enrolled in the school.
- `totalMoneyEarned`: Total money earned by the school through student fee payments.
- `totalMoneySpent`: Total money spent by the school, primarily on teacher salaries.

Methods in School Class:

- **getTeachers()**: Returns the list of teachers in the school.
- **addTeacher(Teacher teacher)**: Adds a teacher to the school.
- **getStudents()**: Returns the list of students in the school.
- **addStudent(Student student)**: Adds a student to the school.
- **getTotalMoneyEarned()**: Returns the total money earned by the school.
- **updateTotalMoneyEarned(int moneyEarned)**: Updates the total money earned by the school.
- **getTotalMoneySpent()**: Returns the total money spent by the school.
- **updateTotalMoneySpent(int moneySpent)**: Updates the total money spent by the school.

School Class Implementation:-

```
public class School {

    private List<Teacher> teachers;
    private List<Student> students;
    private static int  totalMoneyEarned;
    private static int totalMoneySpent;
```

❖ Main Class:-

The `Main` class is where the application is executed. It contains:

- Creation of `Teacher` and `Student` objects.
- Adding teachers and students to the `School` object.
- Managing fee payments and salary disbursements.
- Displaying the financial status of the school.

Main class Implementation:-

```
package school.management.system;

import java.util.ArrayList;
import java.util.List;
```



```
public class Main {

    public static void main(String[] args) {
        System.out.println("Starting the School Management
System...");
        Teacher lizzy = new Teacher(1,"Lizzy",500);
        Teacher mellisa = new Teacher(2,"Mellisa",700);
        Teacher vanderhorn = new Teacher(3,"Vanderhorn",600);
        Teacher mehul = new Teacher(4,"Lizzy",1000);

        List<Teacher> teacherList = new ArrayList<Teacher>();
        teacherList.add(lizzy);
        teacherList.add(mellisa);
        teacherList.add(vanderhorn);
        teacherList.add(mehul);


        Student tamasha = new Student(1,"Tamasha",4);
        Student rakshith = new Student(2,"Rakshith Vasudev",12);
        Student rabbi = new Student(3,"Rabbi",5);
        List<Student> studentList = new ArrayList<Student>();
```

Key Features:-

- **Teacher Management**: The system allows the creation of teacher records and tracks their salaries. Teachers can be added dynamically to the school at any time.
- **Student Management**: The system manages student records, including grades and fees. Students can pay fees, and the system keeps track of their payments.
- **Fee Management**: The system tracks the total fees collected from students and updates the school's total earnings.
- **Salary Management**: The system tracks teacher salaries and updates the school's total money spent whenever a teacher's salary is paid.

Conclusion:-

The **School Management System** project effectively manages the administration of a school by keeping track of teachers, students, fee payments, and salary distribution. By employing object-oriented principles, the system is easy to extend and modify. It can be further improved by adding more features like attendance management, report cards, or a user interface for better user interaction.

References:-

1) https://www.java.com/en/
2) https://www.w3schools.com/java/
3) https://www.javatpoint.com/java-tutorial
4) https://www.geeksforgeeks.org/java/
5) https://en.wikipedia.org/wiki/Java_(programming_language)