A PRELIMINARY REPORT ON "School Management System"

SUBMITTED TO THE EDUBRIDGE INDIA PRIVATE LIMITED

SUBMITTED BY

Miss. Dhanushya.G Batch No: EON-5755

Under The Guidance of

Amruta Deore

DEPARTMENT OF S PRAYAS CERTIFIED JAVA FULL STACK DEVELOPER

Chennai

EDUBRIDGE INDIA PRIVATE LIMITED

2021-2022.

ACKNOWLEDGMENT

It gives all of us great pleasure in presenting the preliminary project report on “School Management System”. With due respect and gratitude we would like to take this opportunity to thank internal guide of our project Mrs.Amruta Deore for giving us all the help and guidance we needed. We are really grateful for his kind support. He has always encouraged us and given us the motivation to move ahead. He has put in a lot of time and effort in this project along with us and given us a lot of confidence. Also we wish to thank all the other people who have helped us in the successful completion of this project.

Miss. Dhanushya.G

ABSTRACT

The school management system is an application developed for schools. It is an application developed in Java which is used to store all the school-related records. It stores information related to students and teachers. The objective of developing such a system was to reduce the errors that creep in the manual system where it was very difficult to store the records. It also provides the facility to calculate the fees paid of the student. The whole system is handled by the administrator who has all the rights to edit or modify any school member information. This system was developed to provide a secure, easy to use a reliable system. This was created to handle all the school-related information and save it in records.

Chapter 1

1.INTRODUCTION

1.1 EXISTING SYSTEM

In the earlier system, it was difficult to store all the student and teacher records which required a lot of paperwork and time consumption. It also leads to errors and mistakes. Everything has to be marked manually. These systems also caused redundancy of data. If any records need to be updated then it has to be changed in every file so making it difficult to manage all the data. There was no security as anyone can view those records. So there was the need for developing an automated system which could handle all these problems making the system easier to use.

1.2 SYSTEM REQUIREMENTS

Software Requirements

1. Operating System - Windows 10.
2. Platform – Eclipse IDE.
3. Language – Java.

Hardware Requirements

1. Hard Disk – 2 GB.
2. RAM required – 1 GB (minimum)
3. Processor – Dual Core or Above.

1.3 Source Code:

Student:

```
package schoolmanagementsystem;

public class student {
private int id;
private String  name;
private int grade;
private int feesPaid;
private int feesTotal;
public student (int id,String name,int grade) {
		this.feesPaid=0;
		this.feesTotal=30000;
		this.id=id;
		this.name=name;
		this.grade=grade;
}
public void Setgrade(int grade) {
		this.grade=grade;
}
public void payfees(int fees) {
		feesPaid+=fees;
		school.updateTotalMoneyEarned(feesPaid);
}
public int getid() {
		return id;
}
public String getname() {
		return name;
}
public int getgrade() {
		return grade;
}
public int getfeesPaid() {
		return feesPaid;
}
public int getfeesTotal() {
    return feesTotal;
}
public int getRemainingfees() {
     return feesTotal-feesPaid;
}
@ Override
public String toString() {
    return "student's name:" + name +
                "\nTotal fees paid so far $" + feesPaid;
}
}
```

Teacher:

```
package schoolmanagementsystem;

public class Teacher {
private int id;
private String name;
private int salary;
private int salaryEarned;
public Teacher(int id,String name,int salary) {
    this.id=id;
    this.name=name;
    this.salary=salary;
    this.salaryEarned=0;
}
public int getid() {
    return id;
}
public String getname() {
    return name;
}
public int getsalary() {
    return salary;
}
public void getsalary(int salary) {
    this.salary=salary;
}
public void receivesalary(int salary) {
    salaryEarned+=salary;
    school.updateTotalMoneySpent(salary);
}
@Override
public String toString() {
    return "Name of the Teacher:" + name+"\nTotal salary earned so far $" + salaryEarned;
}
}
```

School:

```
package schoolmanagementsystem;
import java.util.List;
public class school {
    private List<Teacher> teachers;
    private List<student> students;
    private static int totalMoneyEarned;
    private static int totalMoneySpent;
    public school(List<Teacher> teachers, List<student> students) {
        this.teachers=teachers;
        this.students=students;
        totalMoneyEarned=0;
        totalMoneySpent=0;
    }
    public List<Teacher> getTeachers(){
        return teachers;
    }
    public void addTeacher(Teacher teacher) {
        teachers.add(teacher);
    }
    public List<student> getStudents(){
        return students;
    }
    public void addStudent(student student) {
        students.add(student);
}
    public int getTotalMoneyEarned() {
        return totalMoneyEarned;
    }
    public static void updateTotalMoneyEarned(int MoneyEarned) {
        totalMoneyEarned+=MoneyEarned;
    }
    public int getTotalMoneySpent() {
        return totalMoneySpent;
    }
    public static void updateTotalMoneySpent(int MoneySpent) {
        totalMoneyEarned-=MoneySpent;
    }
}
```

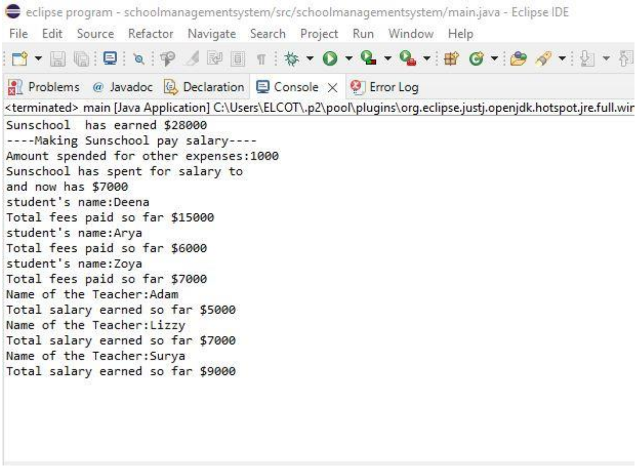
Main:

```
package schoolmanagementsystem;
import java.util.ArrayList;
import java.util.List;
public class main {

    public static void main(String[] args )
    {
        Teacher Adam  =  new Teacher( 1,"Adam",5000);
        Teacher Lizzy  = new Teacher(2,"Lizzy",7000);
        Teacher Surya  = new Teacher(3,"Surya",9000);
        List<Teacher> teacherList = new ArrayList<>();
        teacherList.add(Adam);
        teacherList.add(Lizzy);
        teacherList.add(Surya);

          student Deena = new student(1,"Deena",7);
          student Zoya = new student(2,"Zoya",9);
          student Arya = new student(3,"Arya",11);
    List<student> studentList = new ArrayList<>();
    studentList.add(Deena);
    studentList.add(Zoya);
    studentList.add(Arya);


    school Sunschool= new school(teacherList,studentList);

    Deena.payfees(15000);
    Arya.payfees(6000);
    Zoya.payfees(7000);
    System.out.println("Sunschool  has earned $"  + Sunschool.getTotalMoneyEarned());
    System.out.println("----Making Sunschool pay salary----");
    System.out.println("Amount spended for other expenses:1000");
    Adam.receivesalary(Adam.getsalary());
    Lizzy.receivesalary(Lizzy.getsalary());
    Surya.receivesalary(Surya.getsalary());

    System.out.println("Sunschool has spent for salary to"
    + "\nand now has $"
                    + Sunschool.getTotalMoneyEarned());
    System.out.println(Deena);
    System.out.println(Arya);
    System.out.println(Zoya);
    System.out.println(Adam);
    System.out.println(Lizzy);
    System.out.println(Surya);
    }
    }
```

Chapter 2

2.Project Implementation

2.1 Screen Shot:



eclipse program - schoolmanagementsystem/src/schoolmanagementsystem/main.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help
Problems @ Javadoc Declaration Console Error Log

```
<terminated> main [Java Application] C:\Users\ELCOT\.p2\pool\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.wir
Sunschool  has earned $28000
----Making Sunschool pay salary----
Amount spended for other expenses:1000
Sunschool has spent for salary to
and now has $7000
student's name:Deena
Total fees paid so far $15000
student's name:Arya
Total fees paid so far $6000
student's name:Zoya
Total fees paid so far $7000
Name of the Teacher:Adam
Total salary earned so far $5000
Name of the Teacher:Lizzy
Total salary earned so far $7000
Name of the Teacher:Surya
Total salary earned so far $9000
```

2.2 Conclusion:

Now it is developed to solve all the problems that occurred in the earlier systems. It provides a user-friendly interface where there is an administrator which only has the privilege to access the data. This makes system secure and reliable. This system allows to keep track of the student's details and maintain all the staff information. So everything becomes easy thus reducing the risks that come with an earlier manual records.