EDUBRIDGE LEARNING PVT.LTD

PROJECT DOCUMENTATION

ON

STUDENT MANAGEMENT SYSTEM

(DONE BY

G.ASHWINI)

STUDENT MANAGEMENT SYSTEM

ABSTRACT: Student management systems serve academic institutions in a variety of ways, the most important of which is centralized data administration and accessibility. Teachers will be able to input, maintain, and access student data more simply. Parents and guardians have a better understanding of how their children perform in class.

PURPOSE:

A Student Management System is also known as a Student Information System (SIS). These systems work to **coordinate scheduling and communications between faculty regarding students**. This system exists to simplify information tracking for both parents and administrative staff.

MODULES:

* Student Information Management
* Enrollment Management
* Grade and Subject Management
* Grading System
* Fee and Payment Management

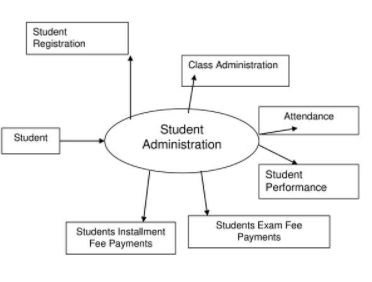
**Student Information Management**: The data that are gathered form the student information must be secured in this system. These data were composed of the student basic information and academic status.

**Enrollment Management**: This is the process where the system takes care of the requirements provided by the student enrollees. They will require the birth cert of the enrollees as well as their academic status that will serve as their basis if the student is new or continuing.

**Grade and Subject Management**: This module designates the instructors per subjects to every student or enrollees.

**Grading System**: The grading system module will sum up and calculate the given output by the students and the result would the academic performance of every student. This will monitor the student performance and behavior throughout the period.

**Fee and Payment Management**: Management of fee and payment is also covered in student management system. The fee and payment information is processed from the enrollment and then the parents or students will be updated

about their accounts to avoid conflicts.

A [use case diagram](https://www.google.com/url?sa=t&rct=j&q=&esrc=s&source=web&cd=&cad=rja&uact=8&ved=2ahUKEwiYy4finOnwAhUbPXAKHeqxDyAQFjACegQICxAD&url=https%3A%2F%2Fwww.smartdraw.com%2Fuse-case-diagram%2F&usg=AOvVaw1paPH3m325jLG07S4AOK-M) is a visual representation of how a user might interact with a program. A use case diagram depicts the system’s numerous use cases and different sorts of users. The circles or ellipses are used to depict the use cases.  
  
By creating the use case of the Student Information System, you must determine first the possible features to identify the flow of the system. After that, you can now create the blueprint or core of the system function.

PROGRAM:

**package** com.Studentmanagementsystem;

**public** **class** Student {

**private** **int** rollno;

**private** String name, email, course;

**private** **int** fee, paid, due;

**private** String address, city, state, country, contactno;

**public** Student() {}

**public** Student(**int** roll no, String name, String email, String course, **int** fee, **int** paid, **int** due, String address,

String city, String state, String country, String contactno) {

**super**();

**this**.rollno = rollno;

**this**.name = name;

**this**.email = email;

**this**.course = course;

**this**.fee = fee;

**this**.paid = paid;

**this**.due = due;

**this**.address = address;

**this**.city = city;

**this**.state = state;

**this**.country = country;

**this**.contactno = contactno;

}

**public** **int** getRollno()

{

**return** rollno;

}

**public** **void** setRollno(**int** rollno)

{

**this**.rollno = rollno;

}

**public** String getName()

{

**return** name;

}

**public** **void** setName(String name)

{

**this**.name = name;

}

**public** String getEmail()

{

**return** email;

}

**public** **void** setEmail(String email)

{

**this**.email = email;

}

**public** String getCourse()

{

**return** course;

}

**public** **void** setCourse(String course)

{

**this**.course = course;

}

**public** **int** getFee()

{

**return** fee;

}

**public** **void** setFee(**int** fee)

{

**this**.fee = fee;

}

**public** **int** getPaid()

{

**return** paid;

}

**public** **void** setPaid(**int** paid)

{

**this**.paid = paid;

}

**public** **int** getDue()

{

**return** due;

}

**public** **void** setDue(**int** due)

{

**this**.due = due;

}

**public** String getAddress()

{

**return** address;

}

**public** **void** setAddress(String address)

{

**this**.address = address;

}

**public** String getCity()

{

**return** city;

}

**public** **void** setCity(String city)

{

**this**.city = city;

}

**public** String getState() {

**return** state;

}

**public** **void** setState(String state)

{

**this**.state = state;

}

**public** String getCountry()

{

**return** country;

}

**public** **void** setCountry(String country)

{

**this**.country = country;

}

**public** String getContactno()

{

**return** contactno;

}

**public** **void** setContactno(String contactno)

{

**this**.contactno = contactno;

}

**package** com.studentdatabaseApp;

**import** java.util.Scanner;

**import** com.Student.Student;

**public** **class** StudentDatabaseApp {

**public** **static** **void** main(String[] args) {

//Ask how many new students we want to add

System.***out***.println("Enter number of new students to enroll: ");

Scanner in = **new** Scanner(System.***in***);

**int** numOfStudents = in.nextInt();

Student[] students = **new** Student[numOfStudents];

//Create n number of new students

**for** (**int** n = 0; n< numOfStudents; n++) {

students[n] = **new** Student();

students[n].enroll();

students[n].paytuition();

}

**for** (**int** n1 = 0; n1 < numOfStudents; n1++) {

System.***out***.println(students[n1].toString());

}

}

}

OUTPUT:

Enter number of new students to enroll:

5

Enter student first name: Bommera

Enter student last name: Pavani

1 - Freshmen

2 - sophmore

3 - junior

4 - senior

Enter student class level 4

Enter course to enroll(Q to quit): English

Enter course to enroll(Q to quit): Physics

Enter course to enroll(Q to quit): Biology

Enter course to enroll(Q to quit): Chemistry

Enter course to enroll(Q to quit): Telugu

Enter course to enroll(Q to quit): Hindi

Enter course to enroll(Q to quit): Q

Your balance is: $3000

Enter your payment: $2000

Thank you for ur payment of $2000

Your balance is: $5000

PROS:

* Improves Communication.
* Improves Student Success.
* Highly Interactive.

CONS:

* Physical Strain.
* Less Interactive.
* Time Consuming.

FEATURES:

* Manage Sessions and Attendance.
* Easy to Learn and Use.
* Responsible and Confident.

CONCLUSION:

Student Management System can be used by educational institutions to maintain their student records easily. Achieving this objective is difficult using the manual system as the information is scattered, can be redundant, and collecting relevant information may be very time-consuming. All these problems are solved by this project.

THANK YOU