



SCHOOL MANAGEMENT SYSTEM

PROJECT NO-18

BY

SHUBHAM KESHARI

2019A8PS0373P

TUSHAR KHANDELWAL

2019A3PS0291P

Github Link for Complete Project- [Link](#)

Drive Link for Complete Project- [Link](#)

CODE EVALUATION, GUI DETAILS AND SAMPLE TEST CASES

1-Concept Used

Here we have used some **basic classes, abstract class and used inheritance relationship between different classes**. For example, we have inherited user class to student, teacher and admin class as they share common features of user.

Various methods like Update_Details, Delete, Add, Modify are used withing appropriate classes in order to execute functionality withing school management system.

● CRITICALLY ANALYSING OUR APPLICATION:-

a) We have **encapsulated** all the private details of User for Admin, Teacher and Student Class. Also we have used School Class as common data which is being used by different users. Depending on priority data will be assessed by different users. For example, Student cant modify his/her marks whereas their teacher and admin can do so.

b) Our application just demanded use of Inheritance so we don't need to **favour composition over inheritance**. Simply by inheriting user class to student, teacher and admin we complete our requirement.

c) Our **Classes are open for extension and closed for modification**, if we needed to expand our application then we can have multiple school system and multiple admin accordingly, we just need to create more objects as per our requirement.

d) **Depend on abstraction, do not depend on concrete classes**, we have used Abstract User Class as we don't this class particular object rather we have inherited this class to other classes like student, teacher and admin and **used Polymorphism**. Therefore we are

dependent on concrete classes rather abstract classes along with polymorphism makes our job easy.

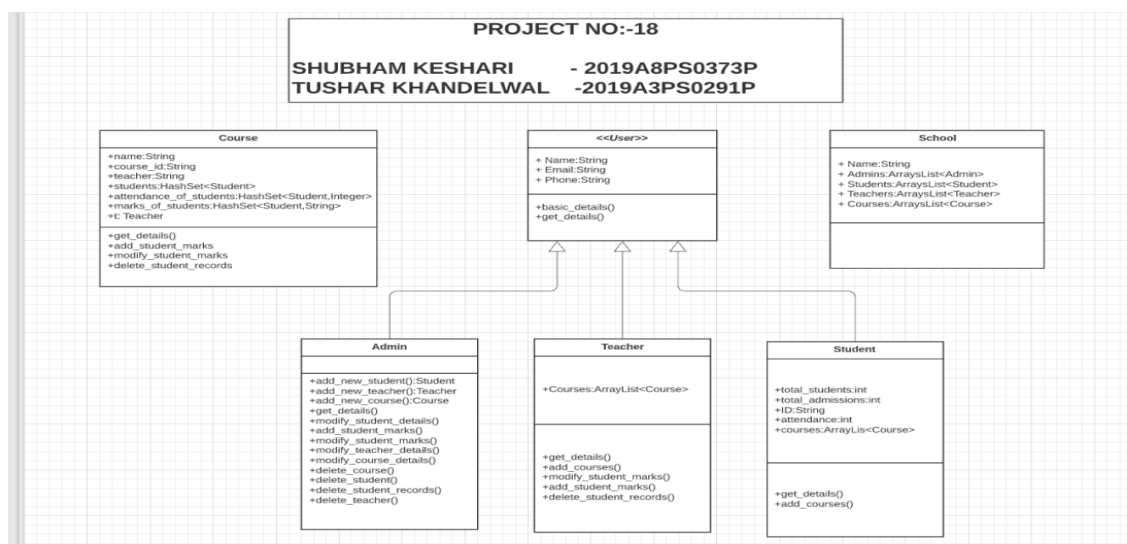
● COMPARING OUR CODE WITH STRATEGY DESIGN PATTERN:-

Strategy Design pattern is needed when instead of implementing single algorithm, we have feature to select algorithm at runtime. This strategy works well when there are number of different behaviour of some user and instead of implementing all the **behaviour independently we program to interface** which is being inherited by different concrete classes and thus separate what is different. Also this **strategy favours composition over inheritance**.

Now comparing with our code since we discussed in critical analysis of our code that we are using inheritance instead of favouring composition, therefore we are adopting this strategy. Also since **we don't have so many different behaviour of any classes, instead we have fixed roles** to be played by student, teacher and admin. We don't need to separate what is different here, rather **we choose to implement individual class algorithm and use Polymorphism** appropriately wherever needed.

CONCLUSION:- Our code didn't used Strategy Design Pattern.

Below is UML Diagram to show various classes and their functions and how they are related to each other:-



2-Classes Details

● SCHOOL CLASS

Here we are storing basic details of our like school name, list of students, teachers, courses are stored here.

● User Class (Abstract)

Here we store basic details of user that are students, admin and teachers. This class has some functions like basic_details for setting the values of users and get_details where we can get details of our users.

- **Admin extending User Class**

This is one of the main class who has control of all students, teachers and courses. The admin has power to add, delete, modify and view details of any students, teachers and courses.

- **Teacher extending User Class**

This class extends user class and has an arraylist of courses which teacher will be teaching. Functions that are used are basic_details which will display details of teacher along with courses they are teaching, add_courses which will store the courses teacher will be teaching, modify, delete, and add marks, grades, attendance that are attending that course under that teacher.

- **Student extending User Class**

Along with basic details stored in user class ,this class will be storing other parameters like total_students, total_admission in school. We are also generating a unique ID for every student so that different student with same name can exist in school. We are also storing attendance and list of courses that the student is enrolled in. Since student don't have any power to modify or delete any records or marks, so only function student class is having is get details, so that student can view his all details and courses they are enrolled in with marks and grades.

- **Course Class**

Our class course is storing name of the course, course ID, and the teacher under which it is assigned to students. Here we will store list of students, their attendance, marks and grades in HashMap and HashSet. Basic functions that are used here are overridden here from teacher and admin classes like get_details, add, modify and delete students marks and records.

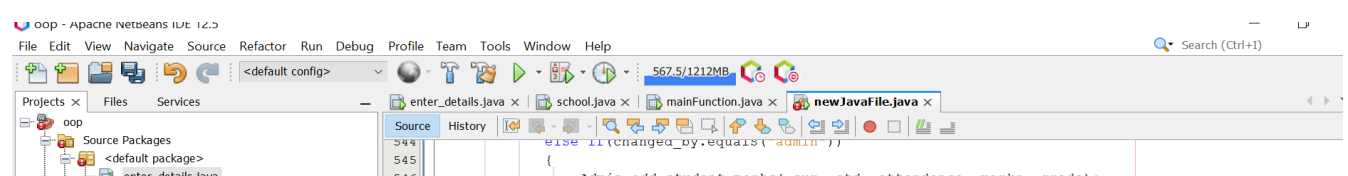
- **Main Class**

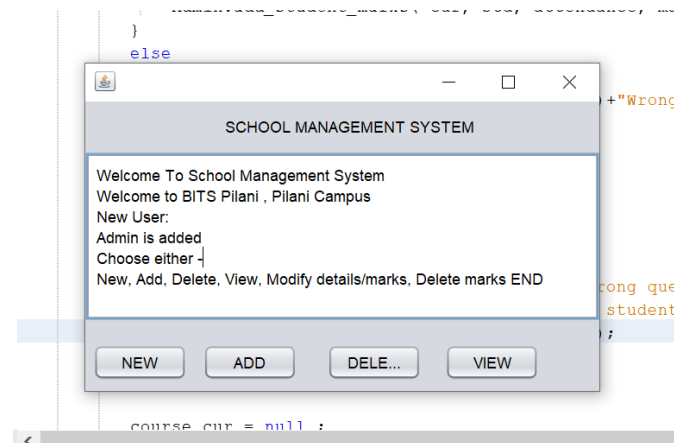
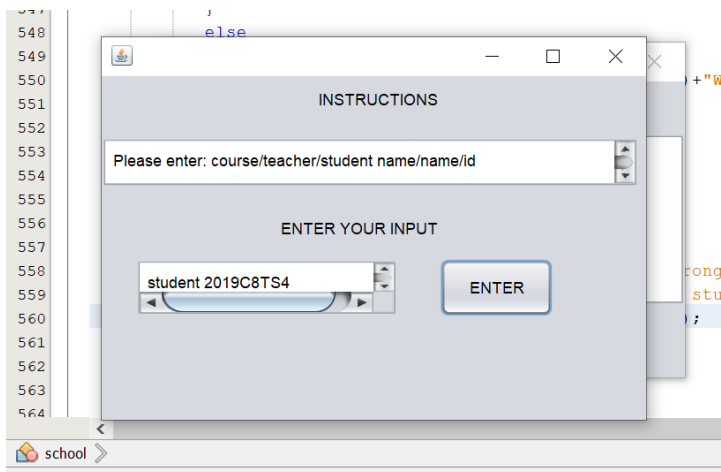
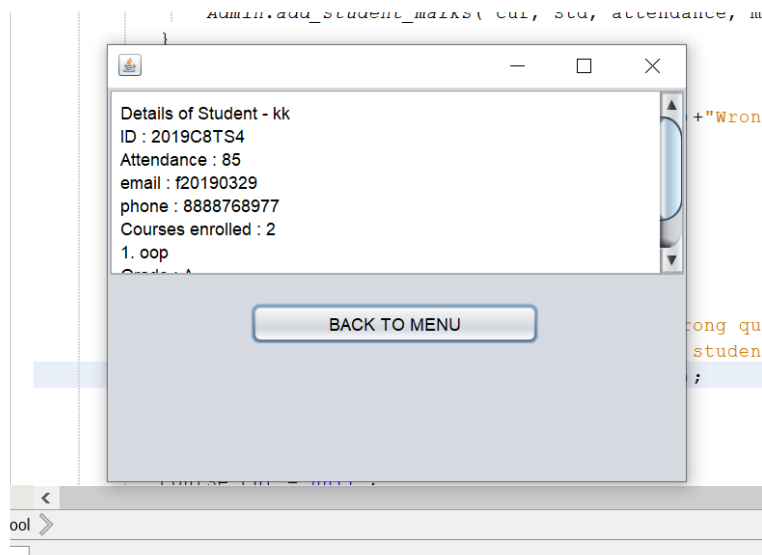
This is our main class where we will be taking input from the operator and displaying appropriate output accordingly. Here we have created object of school and admin, since we can have only one admin, no further object of admin will be created. We will continue taking input from user untill they type END. Here we will have different options like add, delete, view, modify details of students, teachers and courses.

2-GUI Explanation

For the GUI Part we have used NetBeans Editor and created our GUI using Java Swing. The main code of our school management system was edited according to GUI need.

Below are some of the screenshots of GUI Window that was executed:-





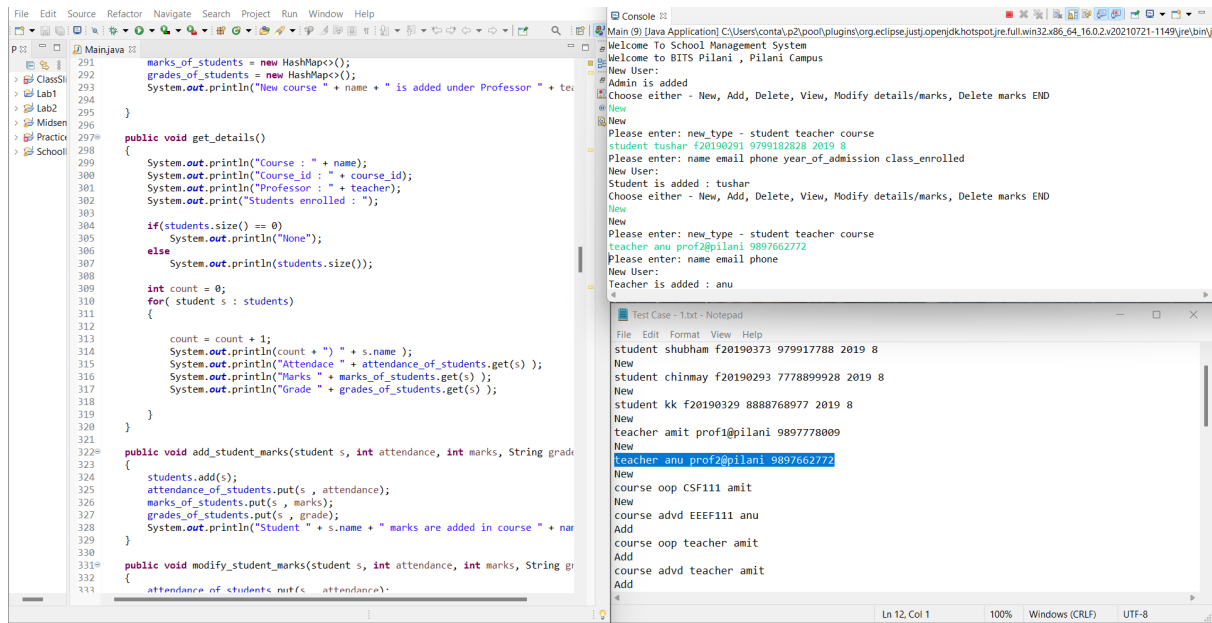
4-Future Scope and Limitations

Currently this School Management System works fine if we are managing a single school where we can add , modify or delete any records easily, for this we have also shown you some sample test cases and also recorded proper explanation of our code, test case and our GUI. but what if we want to manage multiple schools in the same code, that's why we have designed our code in such a way that we can create objects of multiple school where every information can be saved separately, but we have designed our main class in such a way that it works fine for the management of the one school object that we have created. So what we can do is while changing records of any object we can ask for the school name also and change accordingly.

We have also tried to make our school management system as efficient as possible, so to keep records of every student we have given them unique ID according to their year of admission, admission number and class enrolled like BITS ID which is unique for every student so that multiple student with same name can exist , similarly we can implement this for every teacher if it is required in future.

5-SAMPLE TEST CASES AND ITS OUTPUT

Below is our some sample test cases with corresponding output to give you the walkthrough how our code will run:-



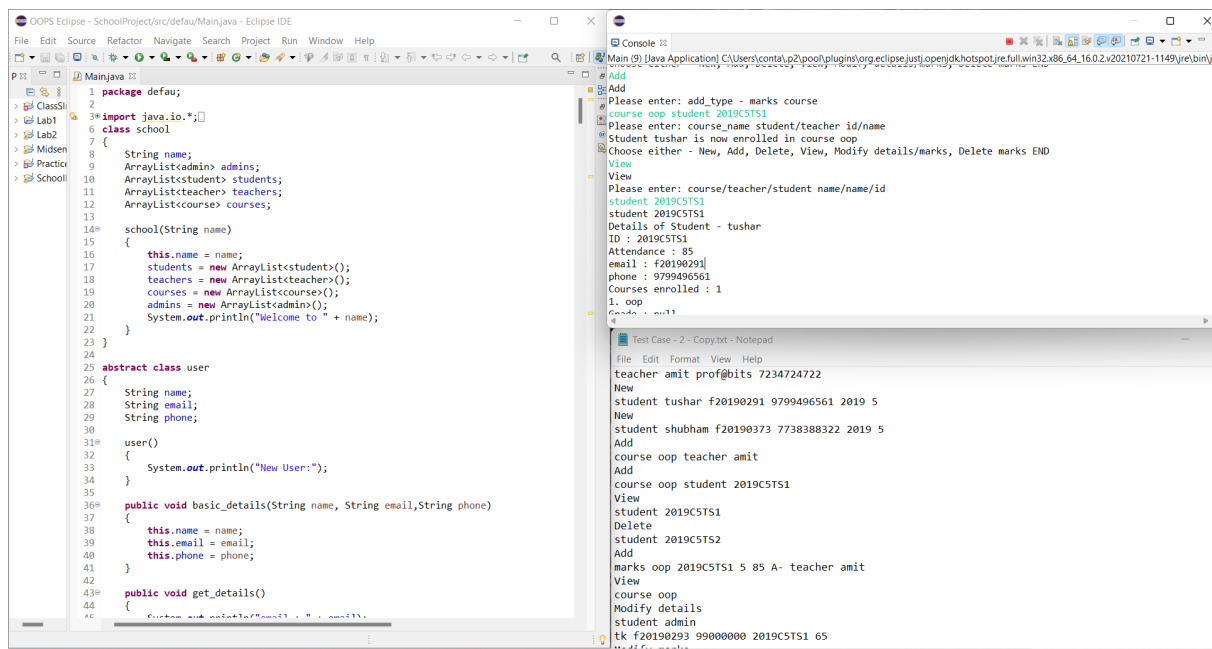
The screenshot shows the Eclipse IDE with a Java project named 'School'. The main class is 'Main.java'. The code includes methods for adding a new course, getting details of a course, adding student marks, and modifying student marks. The console output shows the execution of the program, including prompts for course details, student details, and marks.

```
File Edit Source Refactor Navigate Search Project Run Window Help
Main.java
291 marks_of_students = new HashMap<>();
292 grades_of_students = new HashMap<>();
293 System.out.println("New course " + name + " is added under Professor " + teacher);
294 }
295
296 public void get_details()
297 {
298     System.out.println("Course : " + name);
299     System.out.println("Course_id : " + course_id);
300     System.out.println("Professor : " + teacher);
301     System.out.println("Students enrolled : ");
302
303     if(students.size() == 0)
304         System.out.println("None");
305     else
306         System.out.println(students.size());
307
308     int count = 0;
309     for(student s : students)
310     {
311         count = count + 1;
312         System.out.println(count + " " + s.name);
313         System.out.println("Attendance " + attendance_of_students.get(s));
314         System.out.println("Marks " + marks_of_students.get(s));
315         System.out.println("Grade " + grades_of_students.get(s));
316     }
317 }
318
319 public void add_student_marks(student s, int attendance, int marks, String grade)
320 {
321     students.add(s);
322     attendance_of_students.put(s, attendance);
323     marks_of_students.put(s, marks);
324     grades_of_students.put(s, grade);
325     System.out.println("Student " + s.name + " marks are added in course " + name);
326 }
327
328 public void modify_student_marks(student s, int attendance, int marks, String grade)
329 {
330     attendance_of_students.put(s, attendance);
331 }
332
333
```

Console Output:

```
Main (9) [Java Application] C:\Users\contal.p2\pool\plugins\org.eclipse.justi.openjdk.hotspot.jre.full.win32.x86_64.16.0.2.v20210721-1149\jre\bin\java.exe
Welcome To School Management System
Welcome to BITS Pilani, Pilani Campus
New User:
Admin is added
Choose either - New, Add, Delete, View, Modify details/marks, Delete marks END
New
Please enter: new_type - student teacher course
student tushar f20190291 9799182828 2019 8
Please enter: name email phone year_of_admission class_enrolled
New User:
Student is added : tushar
Choose either - New, Add, Delete, View, Modify details/marks, Delete marks END
New
Please enter: new_type - student teacher course
teacher anu prof2@pilani 9897662772
Please enter: name email phone
New User:
Teacher is added : anu

Test Case - 1.txt - Notepad
File Edit Format View Help
student shubham f20190373 979917788 2019 8
New
student chinmay f20190293 7778899928 2019 8
New
student kk f20190329 8888768977 2019 8
New
teacher amit prof1@pilani 9897778009
New
teacher anu prof2@pilani 9897662772
New
course oop CSF111 amit
New
course advd EEEF111 anu
Add
course oop teacher amit
Add
course advd teacher amit
Add
course
```



The screenshot shows the Eclipse IDE with a Java project named 'School'. The main class is 'Main.java'. The code includes methods for adding a new course, getting details of a course, adding student marks, and modifying student marks. The console output shows the execution of the program, including prompts for course details, student details, and marks.

```
File Edit Source Refactor Navigate Search Project Run Window Help
Main.java
1 package default;
2
3 import java.io.*;
4
5 class school
6 {
7     String name;
8     ArrayList<admin> admins;
9     ArrayList<student> students;
10    ArrayList<teacher> teachers;
11    ArrayList<course> courses;
12
13    school(String name)
14    {
15        this.name = name;
16        students = new ArrayList<student>();
17        teachers = new ArrayList<teacher>();
18        courses = new ArrayList<course>();
19        admins = new ArrayList<admin>();
20        System.out.println("Welcome to " + name);
21    }
22
23
24
25    abstract class user
26    {
27        String name;
28        String email;
29        String phone;
30
31        user()
32        {
33            System.out.println("New User:");
34        }
35
36        public void basic_details(String name, String email, String phone)
37        {
38            this.name = name;
39            this.email = email;
40            this.phone = phone;
41        }
42
43        public void get_details()
44        {
45            System.out.println("Name: " + name);
46        }
47    }
48
49
```

Console Output:

```
Main (9) [Java Application] C:\Users\contal.p2\pool\plugins\org.eclipse.justi.openjdk.hotspot.jre.full.win32.x86_64.16.0.2.v20210721-1149\jre\bin\java.exe
Add
Please enter: add_type - marks course
course oop student 2019C5T51
Please enter: course_name student/teacher id/name
student tushar is now enrolled in course oop
Choose either - New, Add, Delete, View, Modify details/marks, Delete marks END
View
Please enter: course/teacher/student name/name/id
student 2019C5T51
student 2019C5T51
Details of Student - tushar
ID : 2019C5T51
Attendance : 85
email : f20190291
phone : 9799496561
Courses enrolled : 1
1. oop
Grade : null
New
teacher amit prof@bits 7234724722
New
student tushar f20190291 9799496561 2019 5
New
student shubham f20190373 7738388322 2019 5
Add
course oop teacher amit
Add
course oop student 2019C5T51
View
student 2019C5T51
Delete
student 2019C5T52
Add
marks oop 2019C5T51 5 85 A- teacher amit
View
course oop
Modify details
student admin
tk f20190293 99000000 2019C5T51 65
```

Sample Test Case 1:

New
course oop csf111 amit dua
New
teacher amit prof@bits 7234724722
New
student tushar f20190291 9799496561 2019 5
Add
course oop teacher amit
Add
course oop student 2019C5TS1
View
student 2019C5TS1
Delete
student 2019C5TS2
Add
marks oop 2019C5TS1 5 85 A- teacher amit
View
course oop
Modify details
student admin
tk f20190293 99000000 2019C5TS1 65
Modify marks
oop 2019C5TS1 45 100 A teacher
View
student 2019C5TS1
Delete marks
oop 2019C5TS1 admin
View
course oop
END

Sample Output Case 1:

Welcome To School Management System
Welcome to BITS Pilani , Pilani Campus
New User:
Admin is added
Choose either - New, Add, Delete, View, Modify details/marks, Delete marks END
New
Please enter: new_type - student teacher course
Please enter: name course_id teacher
New course oop is added under Professor amit
Choose either - New, Add, Delete, View, Modify details/marks, Delete marks END
New
Please enter: new_type - student teacher course
Please enter: name email phone
New User:
Teacher is added : amit
Choose either - New, Add, Delete, View, Modify details/marks, Delete marks END
New
Please enter: new_type - student teacher course
Please enter: name email phone year_of_admission class_enrolled
New User:
Student is added : tushar
Choose either - New, Add, Delete, View, Modify details/marks, Delete marks END
New
Please enter: new_type - student teacher course
Please enter: name email phone year_of_admission class_enrolled

New User:
Student is added : shubham
Choose either - New, Add, Delete, View, Modify details/marks, Delete marks END
Add
Please enter: add_type - marks course
Please enter: course_name student/teacher id/name
Course oop is now under Professor amit
Choose either - New, Add, Delete, View, Modify details/marks, Delete marks END
Add
Please enter: add_type - marks course
Please enter: course_name student/teacher id/name
Student tushar is now enrolled in course oop
Choose either - New, Add, Delete, View, Modify details/marks, Delete marks END
View
Please enter: course/teacher/student name/name/id
student 2019C5TS1
Details of Student - tushar
ID : 2019C5TS1
Attendance : 85
email : f20190291
phone : 9799496561
Courses enrolled : 1
1. oop
Grade : null
Marks : null
Choose either - New, Add, Delete, View, Modify details/marks, Delete marks END
Delete
Please enter: course/teacher/student name/name/id
Student - shubham is deleted
Choose either - New, Add, Delete, View, Modify details/marks, Delete marks END
Add
Please enter: add_type - marks course
Please enter: course_name student_id attendance marks grade changed_by name
Student tushar marks are added in course oop
Choose either - New, Add, Delete, View, Modify details/marks, Delete marks END
View
Please enter: course/teacher/student name/name/id
course oop
Course : oop
Course_id : csf111
Professor : amit
Students enrolled : 1
1) tushar
Attendance 5
Marks 85
Grade A-
Choose either - New, Add, Delete, View, Modify details/marks, Delete marks END
Modify details
Please enter: course/teacher/student modified_by
name email phone ID attendance
Details of Student - tushar are modified by admin
Choose either - New, Add, Delete, View, Modify details/marks, Delete marks END
Modify marks
Please enter: course_name student_id attendance marks grade modified_by
Student tk marks are modified in course oop
Choose either - New, Add, Delete, View, Modify details/marks, Delete marks END
View
Please enter: course/teacher/student name/name/id
student 2019C5TS1
Details of Student - tk

ID : 2019C5TS1
Attendance : 65
email : f20190293
phone : 99000000
Courses enrolled : 1
1. oop
Grade : A
Marks : 100
Choose either - New, Add, Delete, View, Modify details/marks, Delete marks END
Delete marks
Please enter: course_name student_id modified_by
Student tk is removed from course oop
Choose either - New, Add, Delete, View, Modify details/marks, Delete marks END
View
Please enter: course/teacher/student name/name/id
course oop
Course : oop
Course_id : csf111
Professor : amit
Students enrolled : None
Choose either - New, Add, Delete, View, Modify details/marks, Delete marks END
END

Sample Test Case 2:

New
student tushar f20190291 9799182828 2019 8
New
student shubham f20190373 979917788 2019 8
New
student chinmay f20190293 7778899928 2019 8
New
student kk f20190329 8888768977 2019 8
New
teacher amit prof1@pilani 9897778009
New
teacher anu prof2@pilani 9897662772
New
course oop CSF111 amit
New
course advd EEEF111 anu
Add
course oop teacher amit
Add
course advd teacher amit
Add
course oop student 2019C8TS1
Add
course oop student 2019C8TS4
Add
course oop student 2019C8TS2
Add
course advd student 2019C8TS4
Add
course advd student 2019C8TS3
Add
marks oop 2019C8TS1 87 96 A teacher amit
Add
marks oop 2019C8TS2 82 98 A teacher amit
Add

marks oop 2019C8TS4 91 89 A- teacher amit
Add
marks advd 2019C8TS3 77 87 B teacher amit
Add
marks advd 2019C8TS4 82 85 B teacher amit
View
course oop
View
course advd
View
student 2019C8TS4
Delete
teacher anu
View
course advd
View
student 2019C8TS4
Modify details
student admin
kartikaya f20190329 847493933 2019C8TS4 76
View
student 2019C8TS4
Modify marks
oop 2019C8TS2 100 100 A teacher
View
student 2019C8TS2
Delete marks
oop 2019C8TS1 teacher
View
teacher amit
END

Sample Output Case 2:

Welcome To School Management System
Welcome to BITS Pilani , Pilani Campus
New User:
Admin is added
Choose either - New, Add, Delete, View, Modify details/marks, Delete marks END
New
Please enter: new_type - student teacher course
Please enter: name course_id teacher
New course oop is added under Professor amit
Choose either - New, Add, Delete, View, Modify details/marks, Delete marks END
New
Please enter: new_type - student teacher course
Please enter: name email phone
New User:
Teacher is added : amit
Choose either - New, Add, Delete, View, Modify details/marks, Delete marks END
New
Please enter: new_type - student teacher course
Please enter: name email phone year_of_admission class_enrolled
New User:
Student is added : tushar
Choose either - New, Add, Delete, View, Modify details/marks, Delete marks END
New
Please enter: new_type - student teacher course
Please enter: name email phone year_of_admission class_enrolled
New User:
Student is added : shubham
Choose either - New, Add, Delete, View, Modify details/marks, Delete marks END
Add

Please enter: add_type - marks course
Please enter: course_name student/teacher id/name
Course oop is now under Professor amit
Choose either - New, Add, Delete, View, Modify details/marks, Delete marks END
Add
Please enter: add_type - marks course
Please enter: course_name student/teacher id/name
Student tushar is now enrolled in course oop
Choose either - New, Add, Delete, View, Modify details/marks, Delete marks END
View
Please enter: course/teacher/student name/name/id
student 2019C5TS1
Details of Student - tushar
ID : 2019C5TS1
Attendance : 85
email : f20190291
phone : 9799496561
Courses enrolled : 1
1. oop
Grade : null
Marks : null
Choose either - New, Add, Delete, View, Modify details/marks, Delete marks END
Delete
Please enter: course/teacher/student name/name/id
Student - shubham is deleted
Choose either - New, Add, Delete, View, Modify details/marks, Delete marks END
Add
Please enter: add_type - marks course
Please enter: course_name student_id attendance marks grade changed_by name
Student tushar marks are added in course oop
Choose either - New, Add, Delete, View, Modify details/marks, Delete marks END
View
Please enter: course/teacher/student name/name/id
course oop
Course : oop
Course_id : csf111
Professor : amit
Students enrolled : 1
1) tushar
Attendace 5
Marks 85
Grade A-
Choose either - New, Add, Delete, View, Modify details/marks, Delete marks END
Modify details
Please enter: course/teacher/student modified_by
name email phone ID attendance
Details of Student - tushar are modified by admin
Choose either - New, Add, Delete, View, Modify details/marks, Delete marks END
Modify marks
Please enter: course_name student_id attendance marks grade modified_by
Student tk marks are modified in course oop
Choose either - New, Add, Delete, View, Modify details/marks, Delete marks END
View
Please enter: course/teacher/student name/name/id
student 2019C5TS1
Details of Student - tk
ID : 2019C5TS1
Attendance : 65
email : f20190293
phone : 99000000
Courses enrolled : 1
1. oop
Grade : A
Marks : 100
Choose either - New, Add, Delete, View, Modify details/marks, Delete marks END
Delete marks
Please enter: course_name student_id modified_by

Student tk is removed from course oop

Choose either - New, Add, Delete, View, Modify details/marks, Delete marks END

View

Please enter: course/teacher/student name/name/id

course oop

Course : oop

Course_id : csf111

Professor : amit

Students enrolled : None

Choose either - New, Add, Delete, View, Modify details/marks, Delete marks END