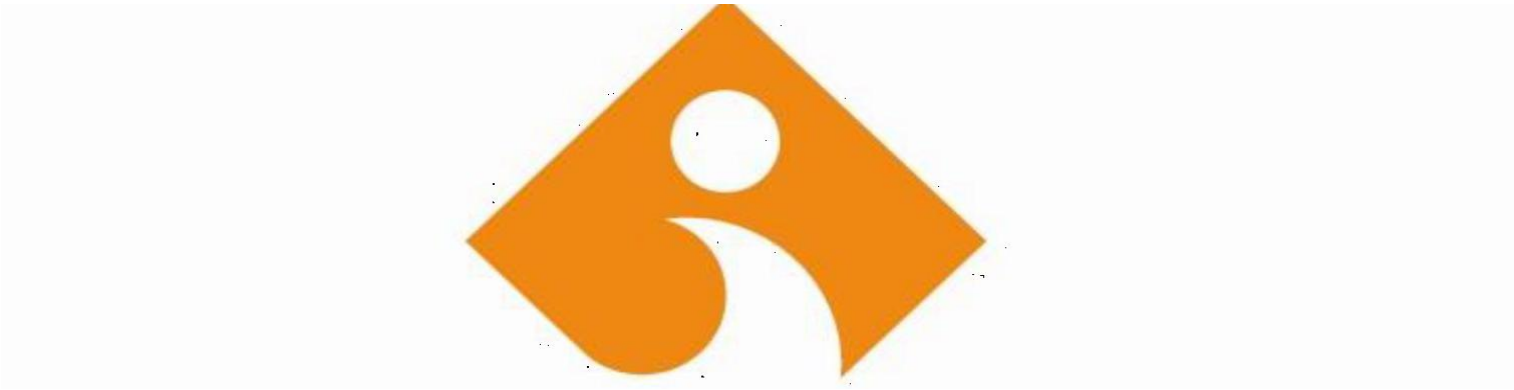


IZMIR UNIVERSITY OF ECONOMICS FACULTY OF ENGINEERING & COMPUTER SCIENCE

FINAL REPORT(2017)

SE 116 – INTRODUCTION TO PROGRAMMING II

Submitted By:
Dilara GEBEŞ
Mehmet Emin ÇEŞİTLİ
Mert KESİMLİ
Naz TEKINALP

Submission Date: 20.05.2017

Lecturer: Ilker KORKMAZ

1. OBJECTIVE OF THE SYSTEM

DAMMNSIS Student Information System (IS)

As our project, we decided to develop a simulation of a Student Information System. The system is based on the following:

1. Storing students, lecturer, lecture, grade information and printing whenever the user wants:

- Students', lecturers' names, ID numbers, grades for each lecture etc.

2.Lecturers' accessing the information of students and editing them:

- Printing students' information
- Grading and storing grades

3.Management of the system

- A manager section to provide the opportunity of adding new students, lecturers, lectures.

4. Login section for students, manager and lecturers:

- The students, the lecturers and the manager have their own login IDs' and passwords'.
- Students are able to print their personal information and grades.
- Lecturers are able to print their personal information and grade their students.
- Manager is able to add new students, lecturers and lectures.

Usage of the system

At the beginning there will be a menu which lists the possible options for users to choose. The user will be able to sign in as a student, lecturer or manager. Firstly, the system will ask for user type (student, manager, or lecturer). After selecting the user type, the system will request an ID and a password from the user. In case the user enters wrong password or ID, the system will automatically return to the login menu. //Also in the system, there is print option for each student and lecturer.

In the system there are some student, lecturer and lecture instances which are already stored, however the manager is always able to add new instances of those, as well.

In the student section, students are able to print their personal information and grades.

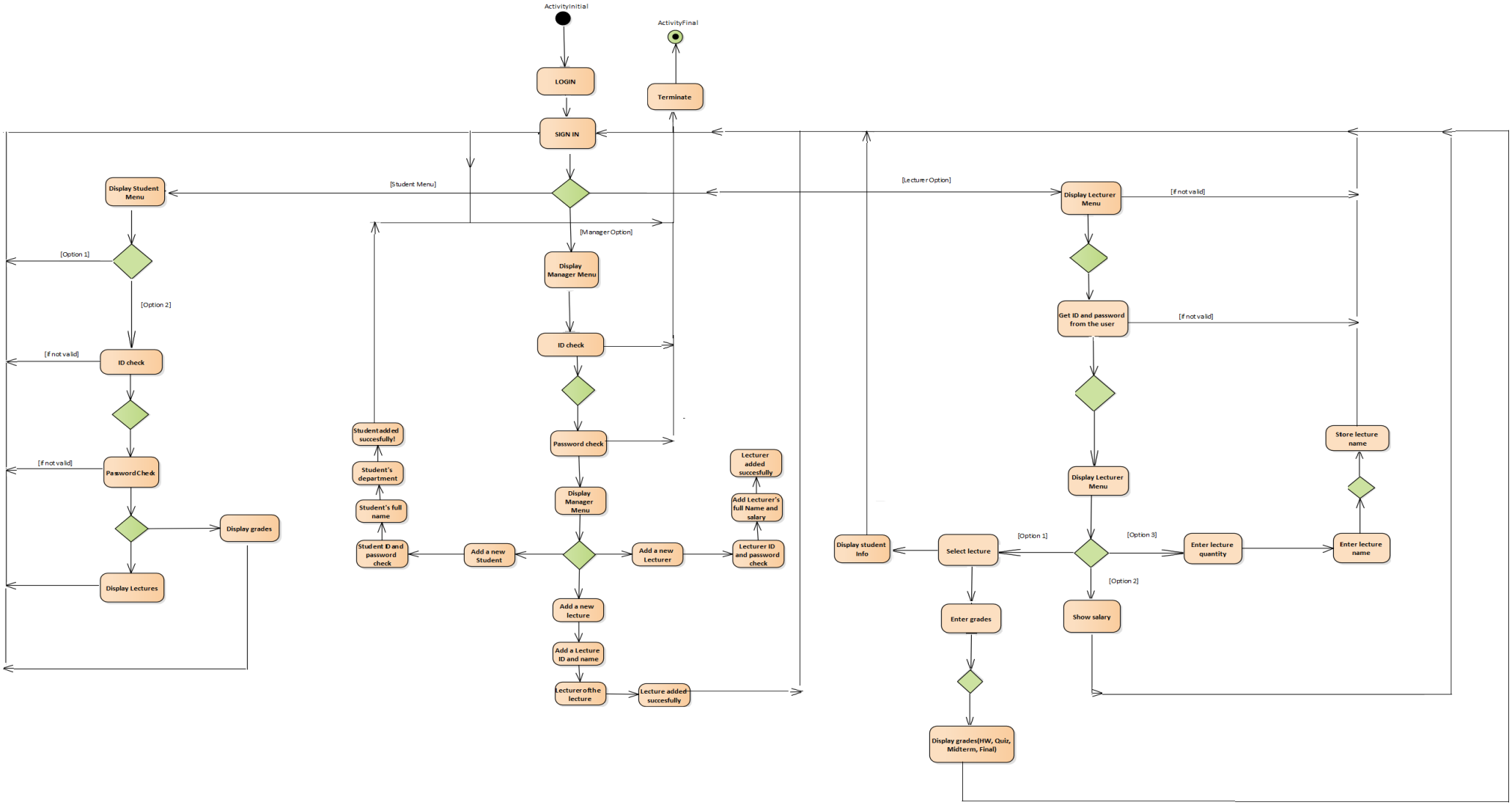
In the manager section, three options for the manager are listed:

Adding a student, adding a lecture and adding a lecturer. Only the manager can add those instances to be stored in the system. After storing the data, the manager can return to the login menu.

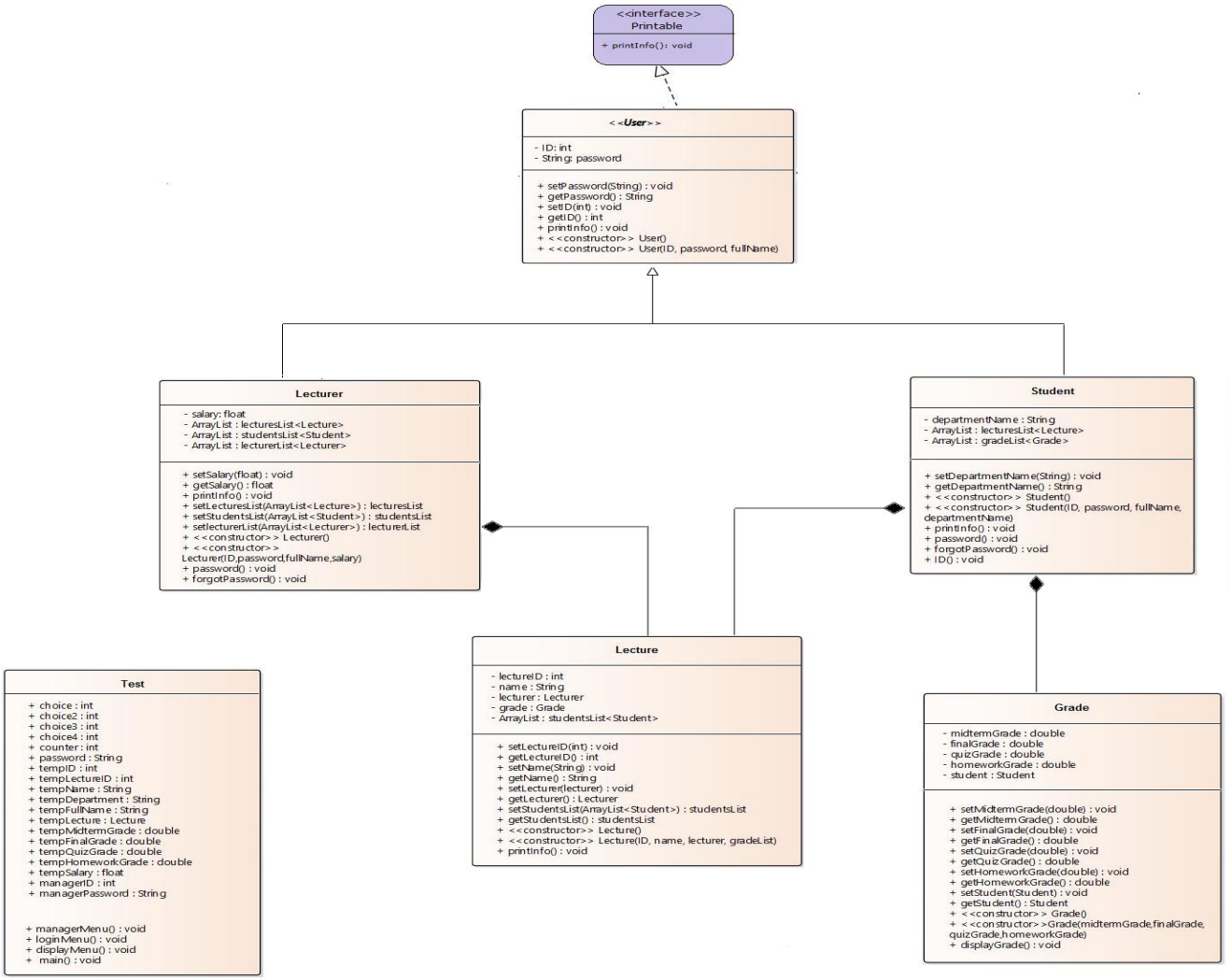
In the lecturer section, after entering ID and password, the personal information of the lecturer will be printed. After that, by entering the ID number of the lecture and student, lecturers will be able to grade their students.

Grades include midterm grade, homework, final grade and quiz grade. Lecturer can enter these grades for their lectures. Students can display the entered grades by lecturers in students menu.

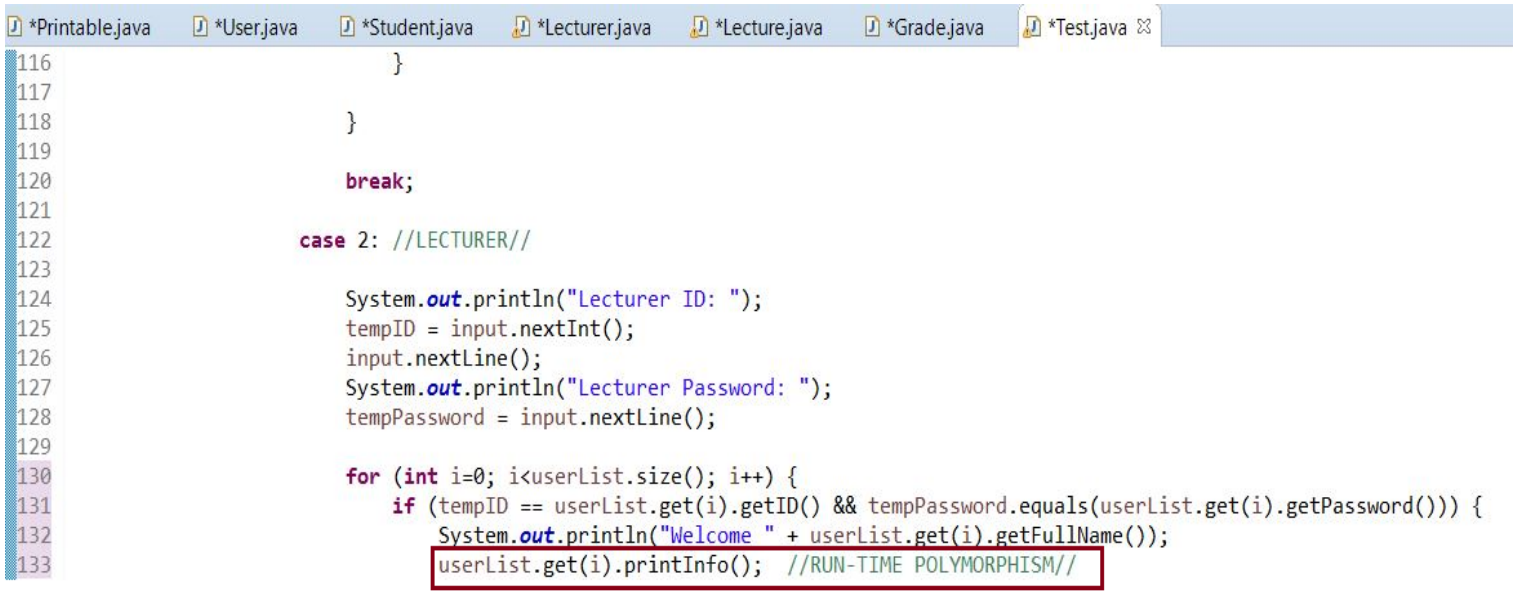
2. UML ACTIVITY DIAGRAM



3. UML CLASS DIAGRAM



4. RUN-TIME POLYMORPHISM




```
116                             }
117
118                         }
119
120                         break;
121
122                     case 2: //LECTURER//
123
124                         System.out.println("Lecturer ID: ");
125                         tempID = input.nextInt();
126                         input.nextLine();
127                         System.out.println("Lecturer Password: ");
128                         tempPassword = input.nextLine();
129
130                         for (int i=0; i<userList.size(); i++) {
131                             if (tempID == userList.get(i).getID() && tempPassword.equals(userList.get(i).getPassword())) {
132                                 System.out.println("Welcome " + userList.get(i).getFullName());
133                                 userList.get(i).printInfo();  //RUN-TIME POLYMORPHISM//
```

Test Class – Line 133

- "User" class is the base class of classes "Student" and "Lecturer". (inheritance hierarchy - "is-a relationship")
- printInfo() method is overriden. (overriding)
- Userlist which has the reference variable of parent class"User" refers to the object of child class "Lecturer".

5. SOURCE CODES

#1 Interface – PRINTABLE

```
public interface Printable {

void printInfo();

}
```

#2 Abstract Class –USER

```
public abstract class User implements Printable  {

protected int ID;
protected String password;

public String getPassword() {
return password;
    }

public void setPassword(String password) {
this.password = password;
    }

protected String fullName;


public int getID() {
return ID;
    }

public void setID(int ID) {
if (ID <= 0)
            ID = 0;
else
            this.ID = ID;
    }

public String getFullName() {
return fullName;
    }

public void setFullName(String fullName) {
this.fullName = fullName;
    }

public User(){
```

```
        setID(0);
        setFullName("No name.");
    }

public User(int ID, String password,String fullName){
        setID(ID);
        setPassword(password);
        setFullName(fullName);
    }


}
```

#3 Concrete Class – STUDENT

```
import java.util.ArrayList;
import java.util.Scanner;

public class Student extends User {
    Scanner input = new Scanner(System.in);

protected String departmentName;
protected ArrayList<Lecture> lecturesList= new ArrayList<Lecture>();
protected ArrayList<Grade> gradeList =  new ArrayList<Grade>();


public void setDepartmentName(String departmentName) {

this.departmentName = departmentName;
    }

public String getDepartmentName() {
return departmentName;
    }

public ArrayList<Lecture> getLecturesList() {
return lecturesList;
    }

public void setLecturesList(ArrayList<Lecture> lecturesList) {
this.lecturesList = lecturesList;
    }


public Student(int ID, String password,String fullName, String departmentName) {
this.ID = ID;
this.password = password;
this.fullName = fullName;
this.departmentName = departmentName;

    }
```

```
public Student(){
super();
        setDepartmentName("No department.");
    }

public Student(int ID, String password,String fullName, String departmentName,ArrayList<Lecture> lecturesList,
ArrayList<Grade> gradeList){
super(ID,password,fullName);
        setDepartmentName(departmentName);
        lecturesList= new ArrayList<Lecture>();
        gradeList =  new ArrayList<Grade>();

    }

    @Override


public void printInfo(){
        System.out.println("Student ID number: "+getID());
        System.out.println("Full name: "+getFullName());
        System.out.println("Department: "+getDepartmentName());

for (int j = 0; j < gradeList.size(); j++) {
            gradeList.get(j).displayGrade();
        }

    }


}
```

#4 Concrete Class - LECTURER

```
import java.util.ArrayList;
import java.util.Scanner;

public class Lecturer extends User {

    Scanner input = new Scanner(System.in);
protected ArrayList<Lecture> lecturesList= new ArrayList<Lecture>();
protected ArrayList<Student> studentsList = new ArrayList<Student>();
protected ArrayList <Lecturer> lecturerList = new ArrayList<Lecturer>();
protected float salary;


public float getSalary() {
return salary;
    }

public void setSalary(float salary) {
if(salary <0)
            salary =0;
else
            this.salary = salary;
    }



public ArrayList<Lecture> getLecturesList() {

return lecturesList;
    }

public void setLecturesList(ArrayList<Lecture> lecturesList) {

this.lecturesList = lecturesList;
    }
```

```
public ArrayList<Student> getStudentsList() {
return studentsList;
    }

public void setStudentsList(ArrayList<Student> studentsList) {
this.studentsList = studentsList;
    }

public Lecturer(){

super();
    }

public Lecturer(int ID, String password, String fullName, float salary){

super(ID,password,fullName);
this.salary = salary;
    }



    @Override
public void printInfo(){
        System.out.println("Lecturer ID: "+getID());
        System.out.println("Name: "+getFullName());
        System.out.println("Salary: "+getSalary());
        System.out.println();



    }

}
```

#5 Concrete Class – LECTURE

```
import java.util.Scanner;
import java.util.ArrayList;
public class Lecture {

    Scanner input = new Scanner (System.in);

protected int lectureID;
protected String name;
protected Lecturer lecturer;
protected User user; //BU DA YENİ //
protected ArrayList<Student> studentsList;
protected Grade grade;


public User getUser() {
return user;
    }

public void setUser(User user) {
this.user = user;
    }

public int getLectureID() {
return lectureID;
    }

public void setLectureID(int lectureID) {
if (lectureID <= 0)  //DEĞİŞTİ //
lectureID = 0; //DEĞİŞTİ//
else
            this.lectureID = lectureID;
    }

public String getName() {
return name;
```

```
    }

public void setName(String name) {
this.name = name;
    }

public Lecturer getLecturer() {
return lecturer;
    }

public void setLecturer(Lecturer lecturer) {
this.lecturer = lecturer;
    }


public Lecture(int ID, String name, Lecturer lecturer, ArrayList<Grade> gradeList ) {
        setLectureID(ID);
        setName(name);
        setLecturer(lecturer);

    }
public Lecture(int ID, User user) { //BU CONSTRUCTOR GEREKLİYDİ //
setLectureID(ID);
        setUser(user); //BUNU YENİ KOYDUM //
ArrayList<Grade> gradeList = new ArrayList<Grade>();



    }

public Lecture(int ID, String name, Lecturer lecturer) {
        setLectureID(ID);
        setName(name);
this.lecturer = lecturer;

    }


public Lecture(int ID, String name) {
```

```
    }


public void printInfo(){
        System.out.println("Lecture ID: "+getLectureID());
        System.out.println("Lecture name: "+getName());
        System.out.println("Lecturer of this lecture: "+getLecturer().getFullName());


    }

}
```

#6 Concrete Class – GRADE

```
import java.util.ArrayList;


public class Grade {
protected double midtermGrade;
protected double finalGrade;
protected double quizGrade;
protected double homeworkGrade;
protected Student student;
protected Lecture lecture;

public double getMidtermGrade() {
return midtermGrade;
    }

public void setMidtermGrade(double midtermGrade) {
this.midtermGrade = midtermGrade;
    }

public double getFinalGrade() {
return finalGrade;
    }

public void setFinalGrade(double finalGrade) {
this.finalGrade = finalGrade;
    }

public double getQuizGrade() {
return quizGrade;
    }

public void setQuizGrade(double quizGrade) {
this.quizGrade = quizGrade;
    }

public double getHomeworkGrade() {
```

```
return homeworkGrade;
    }

public void setHomeworkGrade(double homeworkGrade) {
this.homeworkGrade = homeworkGrade;
    }

public Grade() {

    }

public void calculateAverage(){
double avg =
((getMidtermGrade()*30/100)+(getFinalGrade()*40/100)+(getQuizGrade()*10/100)+(getHomeworkGrade()*20/100));
        System.out.println("Average grade: "+avg);
    }


public Grade(double midtermGrade, double finalGrade, double quizGrade, double homeworkGrade, Lecture lecture){
this.midtermGrade = midtermGrade;
this.finalGrade = finalGrade;
this.quizGrade = quizGrade;
this.homeworkGrade = homeworkGrade;
this.lecture = lecture;
    }

public void displayGrade() {
        System.out.println();
        System.out.println("Grades for " + lecture.getName());
        System.out.println("Homework Grade: " +getHomeworkGrade());
        System.out.println("Quiz Grade: " +getQuizGrade());
        System.out.println("Midterm Grade: " +getMidtermGrade());
        System.out.println("Final Grade: " +getFinalGrade());
        calculateAverage();
        System.out.println();
    }


}
```

#7 Concrete Class – TEST

```
import java.util.ArrayList;
import java.util.Scanner;

public class Test {

public static void managerMenu() {
        System.out.println("****MANAGER MENU****");
        System.out.println("Which one do you want to add to the system?");
        System.out.println("ENTER 1 TO ADD A NEW STUDENT");
        System.out.println("ENTER 2 TO ADD A NEW LECTURER");
        System.out.println("ENTER 3 TO ADD A NEW LECTURE");
        System.out.println("ENTER 0 TO GO BACK");
        System.out.print("Choice? :");

    }

public static void loginMenu() {
        System.out.println("***Welcome to our Student Information System!***");
        System.out.println("ENTER ANY NUMBER TO LOGIN");
        System.out.println("ENTER 0 TO TERMINATE THE SYSTEM");
        System.out.print("Choice? :");


    }

public static void displayMenu() {
        System.out.println("***SIGN IN*** ");
        System.out.println("ENTER 1 TO SIGN IN AS STUDENT ");
        System.out.println("ENTER 2 TO SIGN IN AS LECTURER ");
        System.out.println("ENTER 3 TO SIGN IN AS MANAGER ");
        System.out.println("ENTER 0 TO GO BACK TO THE MAIN MENU");
        System.out.print("Choice? :");

    }
```

```
public static void main(String[] args) {
        Scanner input = new Scanner(System.in);
int choice;
int choice2;
int choice3;
int counter;
int tempID;
int tempLectureID;
        String tempName;
        String tempPassword;
        String tempDepartment;
        String tempFullName;
double tempMidtermGrade;
double tempFinalGrade;
double tempQuizGrade;
double tempHomeworkGrade;
float tempSalary;
        Lecturer lecturerRef = new Lecturer();

int managerID = 123;
        String managerPassword = "admin";
        ArrayList<Student> studentsList = new ArrayList<Student>();
        ArrayList<User> userList = new ArrayList();
        ArrayList<Lecture> lecturesList = new ArrayList<Lecture>();
        ArrayList<Grade>  gradeList = new ArrayList<Grade>();

        Student s1 = new Student(123, "naz", "Naz Tekinalp", "Computer Engineering");
        Student s2 = new Student(345, "meert", "Mert Kesimli", "Computer Engineering ");
        Student s3 = new Student(678, "mehmet", "Mehmet Emin Cesitli", "Computer Engineering");
        Student s4 = new Student (567, "dilara", "Dilara Gebes", "Software Engineering");
        Lecturer lr1 = new Lecturer(123, "ilker",  "Ilker Korkmaz" , 5000);
        Lecturer lr2 = new Lecturer(3459, "nazan", "Nazan Gurkan", 3000);
        Lecturer lr3 = new Lecturer (4789, "sevin" , "Sevin Gumgum", 6000);
        Lecturer lr4 = new Lecturer(1123, "nimer" , "Nimet Kardes Sever", 2000);
        Lecture l1 = new Lecture (456, "SE116", lr1);
        Lecture l2 = new Lecture(111, "GER102", lr2);
        Lecture l3 = new Lecture(222, "MATH154", lr3);
        Lecture l4 = new Lecture (999, "PHYS101", lr4);
```

```
        studentsList.add(s1);
        studentsList.add(s2);
        studentsList.add(s3);
        studentsList.add(s4);
        userList.add(lr1);
        userList.add(lr2);
        userList.add(lr3);
        userList.add(lr4);
        lecturesList.add(l1);
        lecturesList.add(l2);
        lecturesList.add(l3);
        lecturesList.add(l4);
```

```
while (true) {   //LOGIN//
loginMenu();
            choice = input.nextInt();
if(choice == 0) {
                System.exit(0);
            }
            displayMenu();
            choice2 = input.nextInt();
            input.nextLine();
while (choice2 != 0){
switch (choice2) { //2//
case 1: //STUDENT//
System.out.println("Student ID: ");
                        tempID = input.nextInt();
                        input.nextLine();
                        System.out.println("Student Password: ");
                        tempPassword = input.nextLine();

for (int i=0; i<studentsList.size(); i++) {
if (tempID == studentsList.get(i).getID() && tempPassword.equals(studentsList.get(i).getPassword())) {
                                System.out.println("Welcome " + studentsList.get(i).getFullName());
                                studentsList.get(i).printInfo();
                            }
```

```
                        }

break;

case 2: //LECTURER//

System.out.println("Lecturer ID: ");
                        tempID = input.nextInt();
                        input.nextLine();
                        System.out.println("Lecturer Password: ");
                        tempPassword = input.nextLine();

for (int i=0; i<userList.size(); i++) {
if (tempID == userList.get(i).getID() && tempPassword.equals(userList.get(i).getPassword())) {
                                System.out.println("Welcome " + userList.get(i).getFullName());
                                userList.get(i).printInfo();  //RUN-TIME POLYMORPHISM//

System.out.println("***GRADE MENU****");
                                System.out.println("Plese enter the ID of your lecture: ");
                                tempLectureID = input.nextInt();
                                input.nextLine();

for( i=0; i< lecturesList.size(); i++) {
                                    Lecture lec1 = new Lecture(tempLectureID, userList.get(i));
if (tempLectureID == lecturesList.get(i).getLectureID()) {
                                        lecturesList.get(i).printInfo();
                                        System.out.println("Please enter the ID of your student: ");
                                        tempID = input.nextInt();
for (Student st : studentsList) {
if (tempID == st.getID()) {
                                                System.out.println("Student : " + st.getFullName());
                                                System.out.println("Please enter the number of grades: ");
                                                counter = input.nextInt();
                                                input.nextLine();

for (int j = 1; j <= counter; j++) {
                                                    System.out.println("HOMEWORK GRADE " + j + ":");
                                                    tempHomeworkGrade = input.nextDouble();
                                                    input.nextLine();
```

```
                                                System.out.println("QUIZ GRADE " + j + ":");
                                                tempQuizGrade = input.nextDouble();
                                                input.nextLine();
                                                System.out.println("MIDTERM GRADE " + j + ":");
                                                tempMidtermGrade = input.nextDouble();
                                                input.nextLine();
                                                System.out.println("FINAL GRADE " + j + ":");
                                                tempFinalGrade = input.nextDouble();
                                                input.nextLine();

                                                Grade gradeRef = new Grade(tempMidtermGrade, tempFinalGrade,
tempQuizGrade, tempHomeworkGrade, lecturesList.get(i));

                                                st.gradeList.add(gradeRef);
                                                gradeRef.displayGrade();


                                            }


                                        }
                                    }
                                }
                            }
                        }

                    }



break;


case 3: //MANAGER//
System.out.println("Manager ID: ");
                        tempID = input.nextInt();
                        input.nextLine();
                        System.out.println("Manager Password: ");
                        tempPassword = input.nextLine();
if (tempID == managerID && tempPassword.equals(managerPassword)) {
```

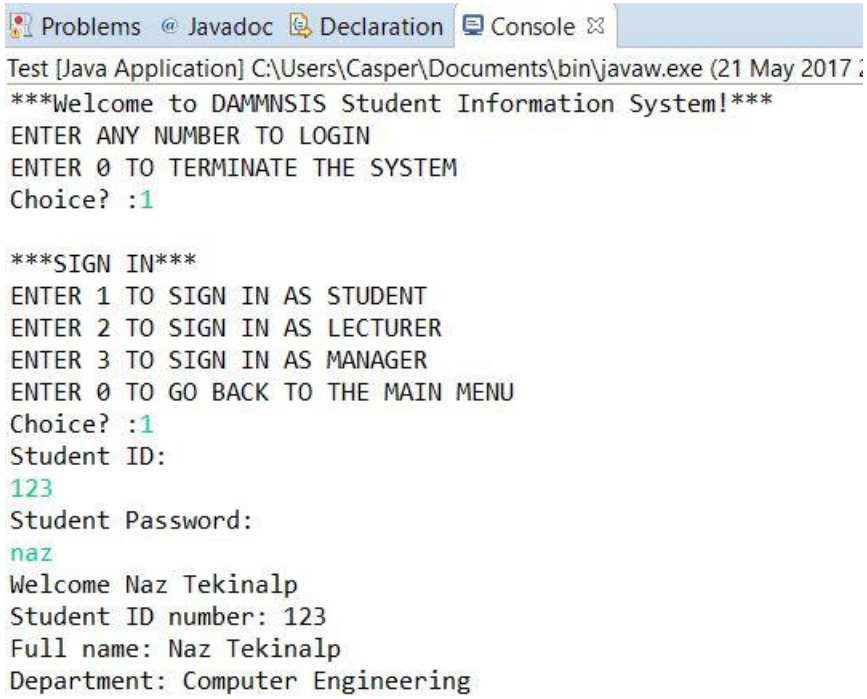
```
                        System.out.println("You have signed in as manager.");
                        managerMenu();

                        choice3 = input.nextInt();
                        input.nextLine();
while (choice3 != 0){
switch (choice3) { //3//

case 1:
                                        System.out.println("You have chosen to add a new STUDENT!");
                                        System.out.println("Student ID: ");
                                        tempID = input.nextInt();
                                        input.nextLine();
                                        System.out.println("Student password: ");
                                        tempPassword=input.nextLine();
                                        System.out.println("Student's Full name: ");
                                        tempFullName = input.nextLine();

                                        System.out.println("Student's Department: ");
                                        tempDepartment = input.nextLine();

                                        Student st1 = new Student(tempID, tempPassword,tempFullName,
tempDepartment);
                                        studentsList.add(st1);
                                        System.out.println("Student successfully added!");

break;


case 2:
                                        System.out.println("You have chosen to add a new LECTURER!");
                                        System.out.println("Lecturer ID: " );
                                        tempID = input.nextInt();
                                        input.nextLine();
                                        System.out.println("Lecturer password: ");
                                        tempPassword=input.nextLine();
                                        System.out.println("Lecturer Name: ");
                                        tempFullName = input.nextLine();
                                        System.out.println("Salary: ");
                                        tempSalary = input.nextFloat();
```

```
                                        input.nextLine();

                                        Lecturer le1 = new Lecturer(tempID, tempPassword, tempFullName, tempSalary);
                                        userList.add(le1);
                                        System.out.println("Lecturer successfully added!");
break;

case 3:
                                        System.out.println("You have chosen to add a new LECTURE!");
                                        System.out.println("Lecture ID: ");
                                        tempID = input.nextInt();
                                        input.nextLine();
                                        System.out.println("Lecture Name: ");
                                        tempName = input.nextLine();
                                        System.out.println("Enter the id of the lecturer of this lecture: "  );
                                        tempID= input.nextInt();
                                        input.nextLine();
for (int i=0; i< userList.size(); i++){
if (tempID==userList.get(i).getID()){
                                                Lecture lec1 = new Lecture(tempID,tempName,lecturerRef);
                                                lecturesList.add(lec1);
                                                System.out.println("Lecture successfully added!");
                                            }


                                        }
break;
default:
                                        System.out.println("Wrong choice. Please try again.");
break;


                                    }
                                    managerMenu();
                                    choice3= input.nextInt();
                                    input.nextLine();
                                }
                            }
```

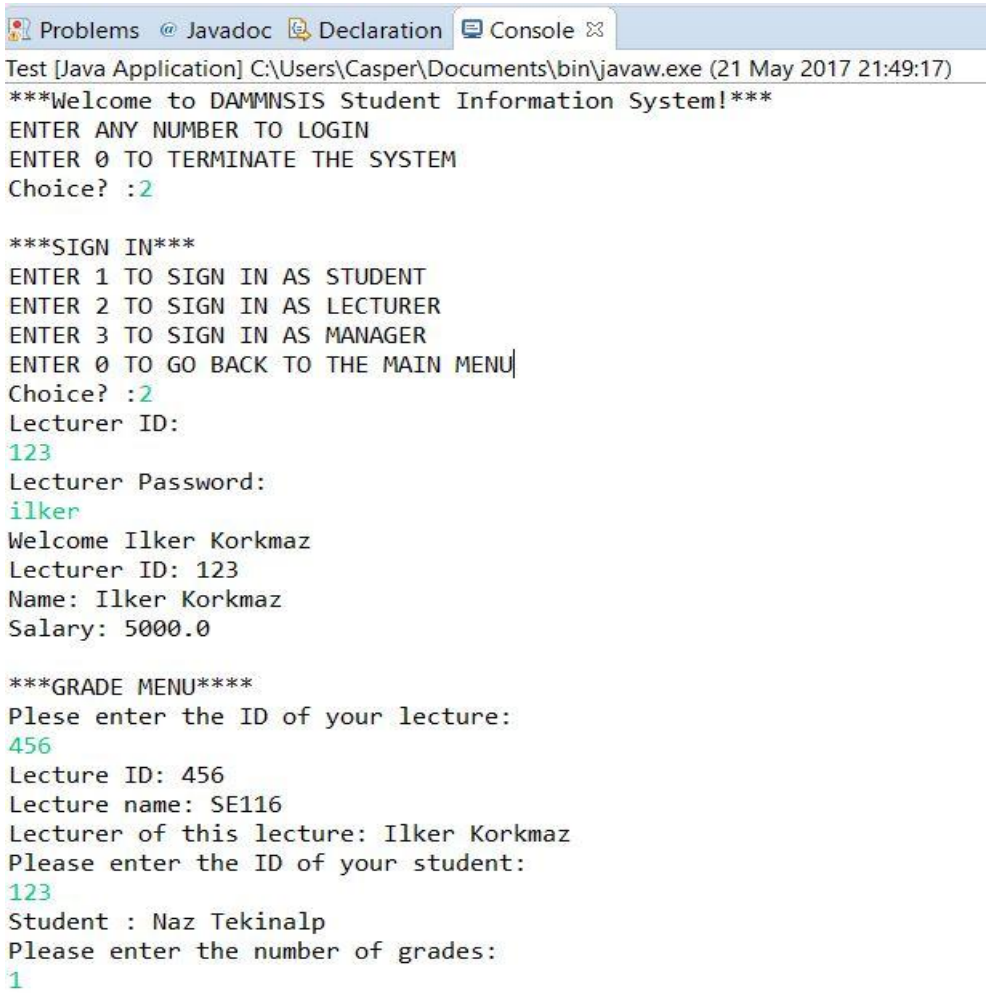
```
break;

default:

break;

                }
                displayMenu();
                choice2= input.nextInt();
                input.nextLine();
            }




        }

    }
}
```

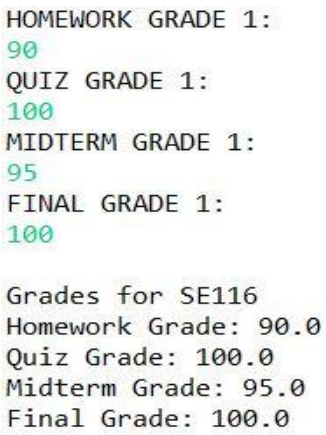
5. SAMPLE OUTPUTS

Example 1 : Sign in as a Student



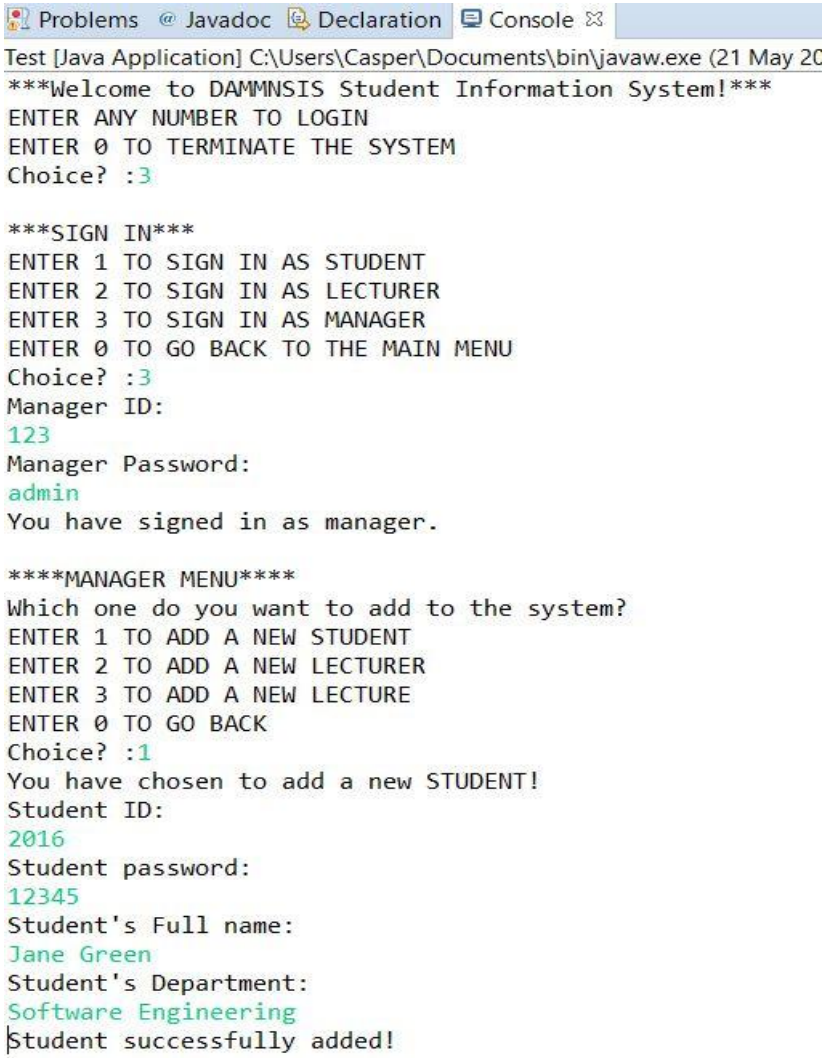
Example 2 : Sign in as a Lecturer



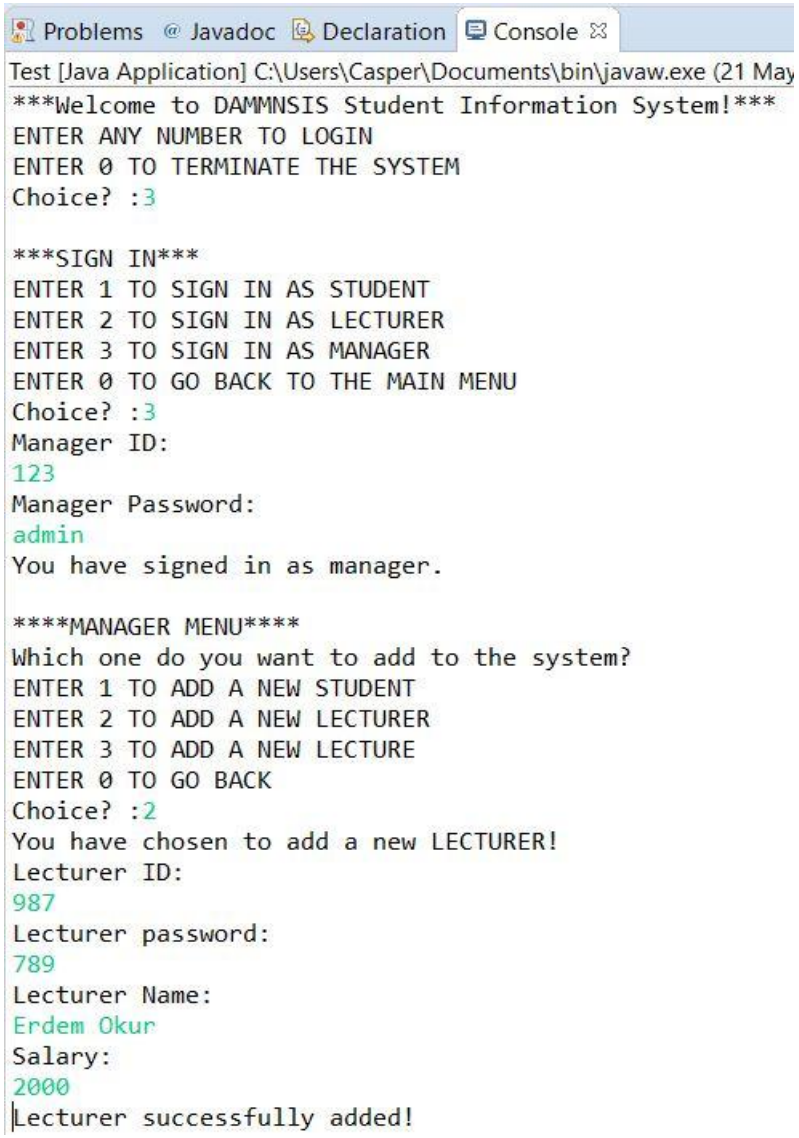


Example 3 : Sign in as a Manager

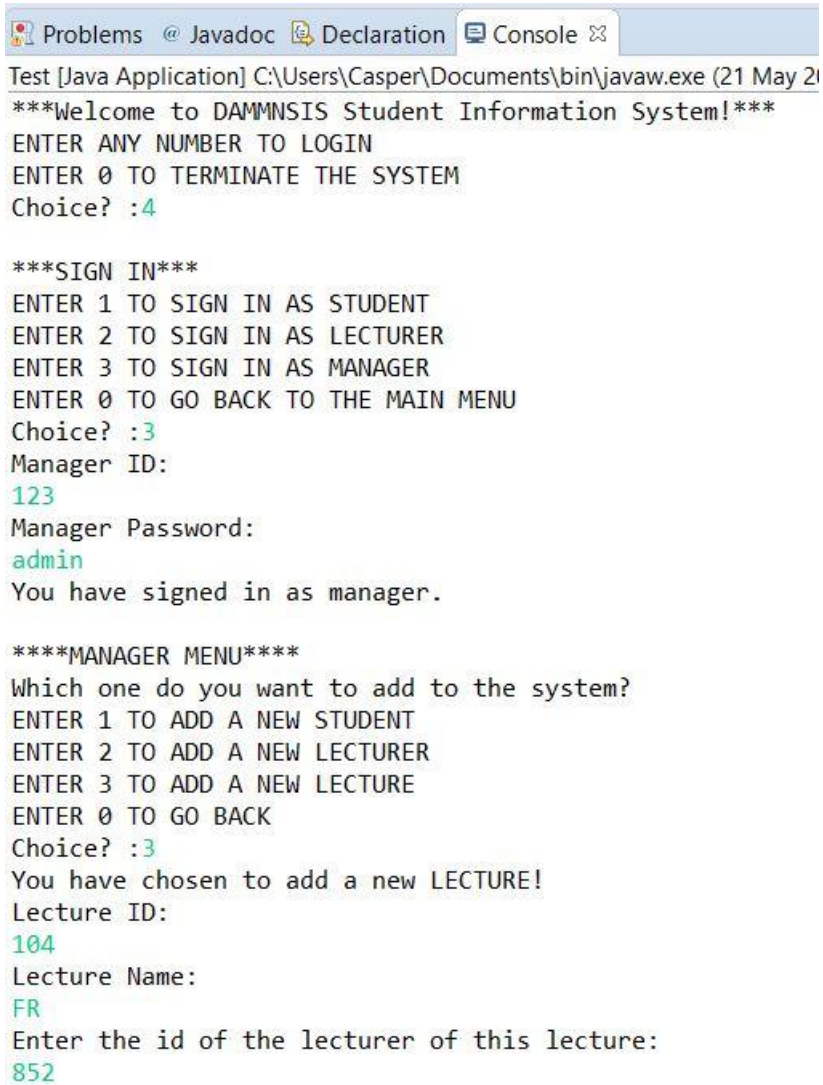
3.1 Add a new Student



3.2 Add a new Lecturer

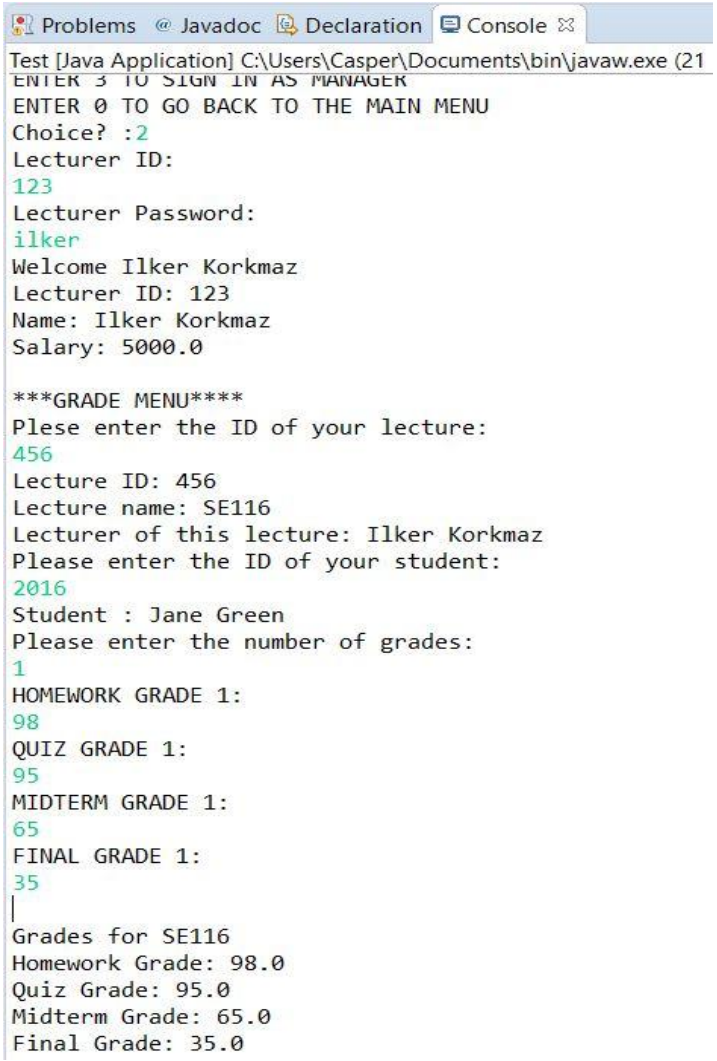


3.3 Add a new Lecture



Example 4 : Display Info Created by The Manager

4.1 Display grade created by the Lecturer



4.2 Display grade for the Student

```
***SIGN IN***
ENTER 1 TO SIGN IN AS STUDENT
ENTER 2 TO SIGN IN AS LECTURER
ENTER 3 TO SIGN IN AS MANAGER
ENTER 0 TO GO BACK TO THE MAIN MENU
Choice? :1
Student ID:
2016
Student Password:
12345
Welcome Jane Green
Student ID number: 2016
Full name: Jane Green
Department: Software Engineering

Grades for SE116
Homework Grade: 98.0
Quiz Grade: 95.0
Midterm Grade: 65.0
Final Grade: 35.0
```

4.3 Display lecture created by the Manager from Lecturer Menu