****

**STUDENT RECOMMENDER SYSTEM**

|  |  |
| --- | --- |
| **Name:** | Mohit Sharma |
| **Branch:** | CSE |
| **Section:** | A2 |
| **Roll No.:** | 1711/17 |
| **Session:** | 2018-19 |
| **Subject:** | COM-404 |
| **Programming Language Used:** | JAVA |
| **Submitted To:** | Mrs. Sheerin Zadoo |

**Contents:**

1. **Requirement Specifications:** List of USE Cases.
2. **Design Specification:** Class Diagram and Sequence Diagram.
3. **Program Specification:** Important Algorithms/ Function Logic.
4. **Implementation:** GitHub Link
5. **Testing:** Unit Testing Specifications and Execution Logs.

**1. Requirement Specification**

List of Use Cases

**Use Case 1:** Uploading Details – In this the details is the details of the student is uploaded including their name and backlog count.

**Use Case 2:** Sorting – In this step the sorting of the student with 0 backlogs is done and are recommended for the further placement drive.

**2. Design Specification**

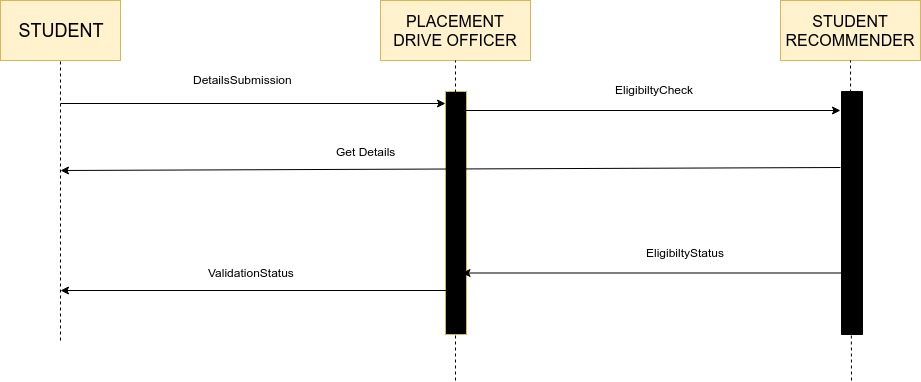
Class Diagram



Class diagram show the function that are included in the program that we are making. This class diagram consist of mainly three tables:

1. Student Table: which is used for representation of four things in it. It consists of Name and backlog count of the student.
2. Student Recommender: This is the main table in which the students are sorted on the basis of their backlog count and are further recommended.

Sequence Diagram



Sequence diagram show the sequence in which the program works. And in this sequence diagram there are mainly three parts:

1. Student: who submit all his details for the placement drive to the placement drive officer.
2. Placement Drive Officer: which further makes use of the student recommender system for the sorting of the students.
3. Student Recommender: which checks the eligibility of the student whether the student is eligible to sit for the placement or not.

After that the Student Recommender get the details of the student and pass the eligibility result to the placement drive officer who further pass on the enrollment status to the student for the placement drive.

**3. Program Specification**

Important Algorithms/ Function Logic

This programme is a student recommender program which is used to recommend students which are eligible to sit in the placement drive on the basis of the marks obtained by the individual. This program works for the sorting of the students which have more than 60% or having 0 backlogs in any of the semesters.

This is a basic input output program which is used to take the input from the user about the marks and the name of the student and print out whether the user is valid to sit for the placement drive or not.

The main logic of code for sorting of students with more than 60% is as follows:

public static void Recomender(double Percentage, int Backlog, String Name)

{ int a;

if (Percentage>=60 && Backlog==0)

{

System.out.println("Mr."+Name+" "+"we're recommending you for the placements");

}

else

{

System.out.println("Sorry Mr."+Name+" "+"you are not eligible for the placement criteria");

Then the program also run for searching of the details if the student have any backlog or not. The main logic code for this is as:

for (int j = 0; j<5 ; j++)

{

if (s.arr[j]<40)

{

back++;

}

}

pr = (sum/500)\*100;

Recomender(pr,back,n);

}

}

This is the main code for the program created which would help in recommending the students with 0 backlog and more than 60% of marks for the placement drive.

**4. Implementation**

GitHub Link

This is the link for the GitHub repository for the Student Recommender Program:

<https://github.com/Mohit-Sharma1/Java-Programs/tree/master/Micro_Project/Micro_Project/Micro_Project/Student_Rcmndr-master>

**5. Testing**

Unit Testing Specifications and Execution Logs

1. **While Entering the Name:**

While entering the name if we had entered the input name as numeric input or some special character input, it should have given an error but instead of giving an error the program has accepted the error input.

1. **While Entering the Marks:**

While entering the marks if we had entered the input marks as alphabetic input or some special character input, it should have given an error but instead of giving an the program has accepted the error input.