

**STUDENT RECOMMENDER SYSTEM**

|  |  |
| --- | --- |
| **Name:** | Sumit Gupta |
| **Branch:** | CSE |
| **Section:** | A1 |
| **Roll No.:** | 1635/17 |
| **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.
6. **Source Code:** Main Program Code for Student recommender system

# Requirement Specification

## List of Use Cases

**Use Case 1:** Input Details – details like name,roll no,marks and backlog count is being given for recommend.

**Use Case 2:** Processing and sorting – In this step processing is done on the basis of given details and sorting of the student with 0 backlogs is done and are recommended for the further placement drive.

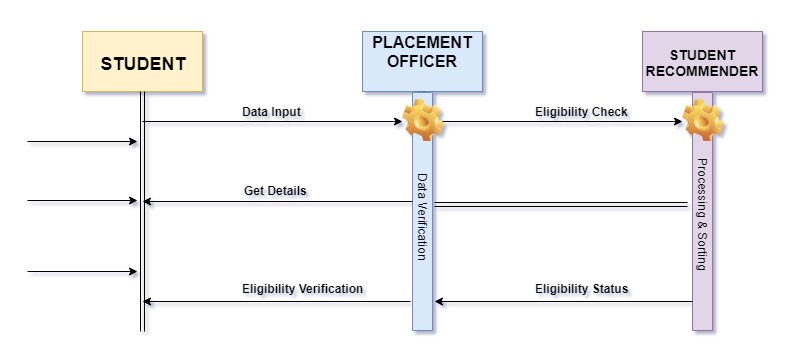
# Design Specification

## Class Diagram

Class diagram show the function that are included in the program which we are implementing. This class diagram consist of mainly two tables:

1. Student Table: It is used for representation of four things in it. It consists of Name,roll no., marks 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.

# Program Specification

## Important Algorithms/ Function Logic

This programme is a student recommender program which is used to recommend students who are eligible to sit in the placement drive on the basis of the marks obtained by them and backlog count. This program works for the sorting of the students who 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 void per()

{

int add= d.e+java+s.p+ppl+mth+comm.;

res=add/6;

System.out.println("the result of the student is ="+res+"%");

}

public void reco()

{

if (mth>=40 && d.e>=40 && java>=40 && s.p>=40 && ppl>40 &&comm>40 && back==0 && res>=60)

{

System.out.println (name+" is recommended to the company");

}

else

{

System.out.println (name+" is recommended not to the company");

}

}

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:

public class Main

{

public static void main (String[]args)

{

Student s1 = new Student();

s1.getInfo();

s1.per();

s1.reco();

}

}

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.

# Implementation (GitHub Link)

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

https://github.com/sumitgupta2213/student-recommander.git

# Testing

## Unit Testing Specifications and Execution Logs

### 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.

### 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 the error input.

**6. Program Code**

import java.util.\*;

class Student

{

int roll;

String name;

int mth;

int d.e;

int s.p;

int ppl;

int java;

int comm;

int back;

Scanner in = new Scanner(System.in);

public void getInfo()

{

System.out.println("enter the name and roll no.:");

name = in.next();

roll = in.nextInt();

System.out.println("enter the marks of mth, d.e., s.p, ppl, comm, java:");

mth = in.nextInt();

d.e = in.nextInt();

s.p = in.nextInt();

ppl = in.nextInt();

comm = in.nextInt();

java = in.nextInt();

System.out.println("Enter the no. of the backlogs left ");

back = in.nextInt();

}

public void per()

{

int add= mth+d.e+s.p+ppl+java+comm.;

res=add/6;

System.out.println("the result of the student is ="+res+"%");

}

public void reco()

{

if (mth>=40 && d.e>=40 && s.p>=40 && ppl>=40&&java>40&&comm.>40 && back==0 && res>=60)

{

System.out.println (name+" is recommended to the company");

}

else

{

System.out.println (name+" is not recommended to the company");

}

}

}

public class Main

{

public static void main (String[]args)

{

Student s1 = new Student();

s1.getInfo();

s1.per();

s1.reco();

}

}

*Sumit Gupta*

*1635/17*

*CSE ‘A1’*

