FEASIBILITY REPORT

-Group 19

COLLEGE REGISTRATION

SUBMITTED BY:

Kshitij Jain(2018BCS-026)
Sakshi Rai(2018BCS-046)
Tanay Saxena(2018BCS-055)

Outline

1. INTRODUCTION:

1.1 OVERVIEW OF THE PROJECT:

The Joint Entrance Examination is one of the largest-scale examinations conducted across the nation, wherein, over 10 lakh students compete. To manage such a large database, an efficient and fool-proof framework is a prerequisite. JoSAA, or Joint Seat Allocation Authority is an agency established by the Ministry of Human Resource Development to manage and regulate the admission to 109 tertiary institutes administered by the national government of India.

Our project is the description of the background functionality of the registration procedure carried out, once the final round of the JoSAA counselling gets commenced. There are two types of "persons" involved in this system, i.e., the students and the staff of the respective institutes. The student class has the data types giving information about the rank and the category of the students. This helps in determining the branch they would opt for, in the institute of their preference. The total number of seats in a specific branch is also displayed to the students. To finally get enrolled, the students have to pay the fee, that is determined according to the category of the students. There is a person from the teaching staff who is the representative for the JoSAA body from that institute who handles the registration procedure for the students.

1.2 OBJECTIVES OF THE PROJECT:

The objectives of this project are to:

1. Develop a central database for the institute registration procedure.
2. Automate the process of institute and branch selection.
3. Provide an efficient online registration and fee determining platform.
4. Make it easy to handle the bulk of students and remove any possibility of manual errors, as the load is really high.

1.3 OVERVIEW OF THE EXISTING SYSTEM AND TECHNOLOGIES:

The existing system of the JoSAA registration procedure is largely dependent on filling a large number of forms, providing signatures, handling the demand drafts for fee payment, getting the medical certificates and providing the students with the allotment forms and letters. At the time of the college registration as well, a lot of papers, certificates, documents, etc. are handled and exchanged. This proves to be a very slow and inefficient process, with a high probability of committing errors.

1.4 THE NEED FOR THE PROJECT:

JEE is one of the most difficult and the most widespread examinations conducted across the whole of India. It is the exam where the largest number of students enrol and compete. Around 13 lakh students participate and around 3 lakh qualify and become eligible for the admission procedure. To handle such a huge number of registrations and the several formalities that come along with them, an efficient and infallible system is mandatory. That system is required to be an online database, otherwise it'll be headache to curb the manual errors that may occur during it. Our project is an attempt to digitalize this registration process, and to handle the large database online, which would definitely make it easy for the respective authorities to manage it. It would also help minimize the errors and loopholes, because the computerized system would ensure the correct handling of all the statistics required and would also provide a backup in case of data loss.

1.5 SCOPE OF THE PROJECT:

Main actors of the system are:

1. Students
2. Institution staff
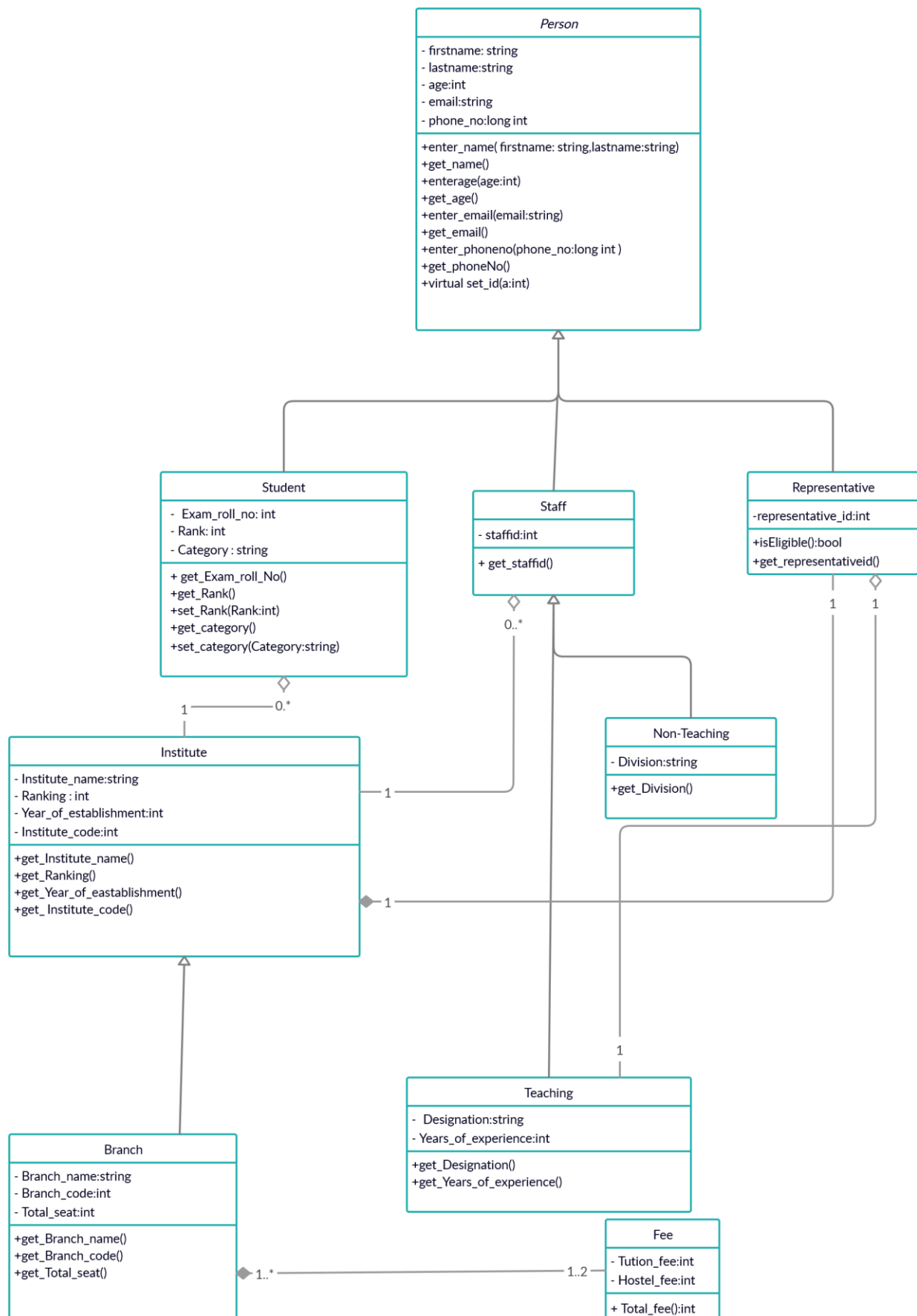3. College representatives for the JoSAA body.

Main use cases associated:

1. Students can:
   - Choose the institute for their own liking out of all the available choices.
   - Select the branch/stream they want to pursue according to their ranks in the JEE.
   - Provide information about their category (general, OBC, SC, ST) to get their fee determined.
   - Take admission into the institute with the help and guidance of the JoSAA representative.

2. Teaching staff can:
   - Get elected as the JoSAA representative in accordance with their years of experience at their respective posts.
   - Guide students in their college registration procedure.

3. Non-teaching staff:
   - Administrative body can overlook the statistical details and handle the paperwork.

1.6 DELIVERABLES:

Object Oriented Programming based college registration system, containing user interface and classes, implemented in the C++ language.

2. CLASS DIAGRAM

Person

- firstname: string
- lastname:string
- age:int
- email:string
- phone_no:long int

+enter_name(firstname: string,lastname:string)
+get_name()
+enterage(age:int)
+get_age()
+enter_email(email:string)
+get_email()
+enter_phoneno(phone_no:long int)
+get_phoneNo()
+virtual set_id(a:int)

Student

- Exam_roll_no: int
- Rank: int
- Category : string

+ get_Exam_roll_No()
+get_Rank()
+set_Rank(Rank:int)
+get_category()
+set_category(Category:string)

Staff

- staffid:int

+ get_staffid()

Representative

-representative_id:int

+isEligible():bool
+get_representativeid()

Non-Teaching

- Division:string

+get_Division()

Institute

- Institute_name:string
- Ranking : int
- Year_of_establishment:int
- Institute_code:int

+get_Institute_name()
+get_Ranking()
+get_Year_of_eastablishment()
+get_ Institute_code()

Teaching

- Designation:string
- Years_of_experience:int

+get_Designation()
+get_Years_of_experience()

Branch

- Branch_name:string
- Branch_code:int
- Total_seat:int

+get_Branch_name()
+get_Branch_code()
+get_Total_seat()

Fee

- Tution_fee:int
- Hostel_fee:int

+ Total_fee():int

3. FUNCTIONAL REQUIREMENT

The project contains the following classes:

1. Person
2. Student
3. Staff
4. Representative
5. Institute
6. Teaching
7. Non-Teaching
8. Branch
9. Fee

Description of classes and its functions:

- **Person**

  Person class contains the general information about a person. It is an *abstract* class and is inherited by 3 other classes.
  Functions:
    i. void EnterName(string firstname, string lastname)
    ii. void get_Name()
    iii. void EnterAge(int age)
    iv. void get_Age()
    v. void EnterEmail(string email)
    vi. void get_Email()
    vii. void EnterPhoneNo(long int)
    viii. void get_PhoneNo()
    ix. virtual set_id(int id)

- **Student**

  Student class will handle the details of a student. This class is inherited from Person class.
  Functions:
    i. void get_Exam_roll_No()
    ii. void get_Rank()
    iii. void set_Rank(int rank)
    iv. void get_Category()
    v. void set_Category(string category)

- **Staff**

  Staff class will contains the information of the staffs working in an institute. This class will inherit from Person class.
  Functions:
  - i. void get_Staffid()

- **Representative**

  This class will contain the information of the representative from each institute for the registration process.
  Functions:
  - ii. void get_Repres_id()
  - iii. bool isEligible() – it returns whether a teaching staff is eligible for becoming a representative or not.

- **Institute**

  The Institute class will contain the information about the institute.
  Functions:
  - i. void get_Institute_name()
  - ii. void get_Ranking()
  - iii. void get_Year_of_establishment()
  - iv. void get_Institute_code()

- **Teaching**

  This class will handle the details of all the staff which are working as a professor/lecturer in the institute. It is inherited from Staff class.
  Functions:
  - i. void get_Designation()
  - ii. void get_Years_of_experience()

- **Non-Teaching**

  Non-Teaching class deals with the details of staff who does not come under the Teaching category(i.e someone who is not a professor). It is inherited from Staff class.
  Functions:
  - i. void get_Division()

- **Branch**

  Branch class contains the information of a branch which is supported by the institute. It is inherited from Institute class.
  Functions:

  ii. void get_Branch_name()

  iii. void get_Branch_code()

  iv. void get_Total_seat()

- **Fee**

  Fee class contains the fess of different institutes for different branches.

  Functions:

    i. int Total_fee()

4. FEASIBILITY STUDY

4.1 ECONOMICAL FEASIBILITY:

Being a software-based system, which will be functional only on the college domain, it would take up minimal cost. Since the system doesn't consist of any multimedia data transfer, bandwidth required for the operation performed by this system would be very low. Besides the associated cost, there will be many benefits for the students and the staff involved. Especially the extra effort that is associated with the document handling and verification will be significantly reduced since the process is entirely digital. Error detection and handling would also get much easier which would save both time and money.

4.2 TECHINICAL FEASIBILITY:

This college registration system is a complete Object Oriented Programming based system, implemented in the C++ language. The main technology and tools associated with the system are:

1. C++
2. Object Oriented Programming
3. Database to maintain student record, that can simply be stored in a computer memory.

Each of the technologies are freely available and the technical skills required are manageable. Time limitations and the ease of implementing using these technologies are synchronized and are negligible, as the functions involved aren't much complex.

From this, it is pretty clear that this college registration system is technically feasible.

4.3 RESOURCE AND TIME FEASIBILITY:

Resources that are required for the proper implementation of the college registration system are:
1. Programming device (laptop, PC, etc.)
2. Hosting space on the local domain (freely available)
3. Programming tools (freely available)
4. Programming individuals.

And the time required to store and process this information is negligible for the modern day processors which work at a very high clock speed.

Thus, it's evident that this system has the required resource and time feasibility.

5. CONSIDERATIONS:

5.1 PERFORMANCE:

Since this project is being implemented in C++ using OOPs on a local system and the g++ compiler, the execution time of the program will be very fast, even when the number of records increase. Since every function would be processed on a computer, the manual errors would be diminished and the system would become fool-proof. Minimal bandwidth and server space is required for the functionality of the college registration system.

5.2 SECURITY:

Since the system is implemented using the encapsulation features of the Object Oriented Programming in C++, with an efficient usage of private and public data members and member functions, the access to the sensitive information will be limited only to the concerned and designated authorities and the students themselves, thus ensuring a proper privacy and security of the data.

5.3 USABILITY AND EASE OF USE:

The interfaces are designed to make it easy for any potential user to get familiar with the system in no time. No additional training is required to use the system, neither to the students, nor to the respective authorities involved in the procedure.

5.4 CAPACITY AND SCALABILITY:

This OOPS project can easily be integrated to any technology or web-based domain and will function equally well in any kind of a system as it doesn't require a strong processor to get implemented.

5.4 MAINTAINABILITY:

This project will be easy to maintain as it is only required for the involved authorities to install the system just once and it will be available to them henceforth, with little or no requirement to ensure its proper functioning, because it is a simple configuration based on OOPs concept using the C++ language.