**Case Study (60 Marks)**

**Mail from Project Manager**

Dear Team,

The dedication you all are showing is really wonderful.

Here are the updates from our Client. As the number of students registering for the university has increased exponentially they are facing few problems during the registration process.

They want us to create an exception handling system so that there is a possibility to intimate the student about the problem that has occurred during resignation process.

Kindly go through the enhancements that are required for the existing registration system. It’s mandatory to follow Java Coding Standards. Unit Testing should be done. All front end and back end the validations should be considered.

Update me the status stage by stage.

Regards,

SLC Project Manager

@SLC.

**Objective**

Our aim is to make u understand the concepts of Exception Handling

**CASE -1**

**Problem Definition:**

* Developing a university Registration System where students register for admission.
* For registration process students should contact the university registrar.
* Registrar will perform preliminary validations and pass the student details to the validator for validation and return the control back to the registrar
  + **Validator validates the student details and throws AgeException if the student age is >35 or <23**
* Eligibility exam is conducted for registered students, under the supervision of Examiner
* Examiner uploads the question paper for the student
* Student submits the filled paper after the exam
* Evaluator evaluates the paper and the result is intimated back to the student
* Student gets admission in the university

**Assumptions:**

* All the students applied will get admitted.
* **Age of the student must not exceed 35 and must not be less than 23**
* **University Exception must be caught in the registerStudent() Method**

**Suggestions**

**RegistrationForm Class**

* This class consists of all the student details fields as its member variables
* Consists of a parameterized constructor having all the student details as its parameters. Assign the current student details to the member variables

Parameters of RegistrationForm Class are : Name, Marital Status, Age, Sex, Date of Birth, Address, Primary Email id, Secondary Email id, Phone Number, Interested Subject, Highest Education Qualification, Nationality

**Student Class**

This class contains a parameterized constructor having all the student details as its parameters. The registration Form constructor gets called for the student class constructor.

* **Student Object should contain**

String admissionId –it’s a unique id for each student which gets generated after successful validation of student details

String result – The result of the examination

The student constructor to assign the member variables.

* **Each student has a registrar object**
* **Methods in Student class**
* **registerStudent()throws UniversityException–** Contact the registrar and register this student with the registrar(i.e. University)
* **registerForExam() -** Contact the Examination registrar and register this student for the exam which the Examination Registrar conducts. The student gets an uploaded exam paper.
* **appearForExam –** Get the paper from the exam object and appear for the exam and submit it after completing the examination

**Registrar Class**

* Consists of a Private constructor ie implementing singleton pattern, hence there will be only one registrar throughout the application
* **Methods in Registrar class**
* **getRegistrar()** –This method returns a registrar
* **registerStudent(Student student) throws RegistrationException** – This method has a student parameter object (containing all the current student details) – This method contacts the Validator for validating all the student details. After successful validation the Registrar generates a unique admission Id and registers this student with the university

**AgeException**

* it must extend ValidatorException class.
* In its constructor it must pass String to the base class constructor that is describing about the exception.

**ValidatorException**

* It must extend RegistrationException class.
* In must have parameterized constructor which will take string parameter and passes it to the base class constructor.

**RegistrationException**

* It must extend UniversityException class.
* In must have parameterized constructor which will take string parameter and passes it to the base class constructor.

**UniversityException**

* It must extend Exception class.
* In must have parameterized constructor which will take string parameter and passes it to the base class constructor.

Create a Test class to test the above functionality so that students can register and appear for the exam. And if any of the exception comes, throw the exception.