**Interview Analysis**

**Umang Agarwal**

**Problem Statement:**

To develop an Interview Analysis system which will analyze the performance of candidates in quizes. There are two entities Admin and Candidate.

* Candidate /Admin can login through the app.
* Admin should schedule interview for candidates.
* Create multiple questions on different topics with answer.
* Create at least 3 sections for online test.
* Candidate should be provided with sets of MCQ questions.
* Candidate should be able to answer the questions.
* Once done, all the answers should be submitted.
* The admin should be able to view how candidate performed in each section.
* Provide varies graph to show the result (like graph on each section, graph on topics, etc.)

**Scope:**

Using the Interview Analysis application, a candidate can give only one test at a time and see questions whereas the admin can add and update test and also add questions to a specific test. The admin can also delete tests and the questions in them and assign the tests to a specific user.

**Classes:**

**1.User**

**2.Test**

**3.Questions**

**Attributes:**

**1.User**

a.userId : Long

b.userName : String

c.userTest : Test

d.isAdmin : Boolean

e.userPassword : String

**2.Test**

a.testId : BigInteger

b.testTitle : String

c.testDuration : LocalTime

d.testQuestions : Set<Question>

e.testTotalMarks : BigDecimal

f.testMarksScored : BigDecimal

g.currentQuestion: Question

h.startTime: LocalDateTime

i.endTime: LocalDateTime

**3.Questions**

a.questionId : BigInteger

b.questionOptions : Array<String>

c.questionTitle : String

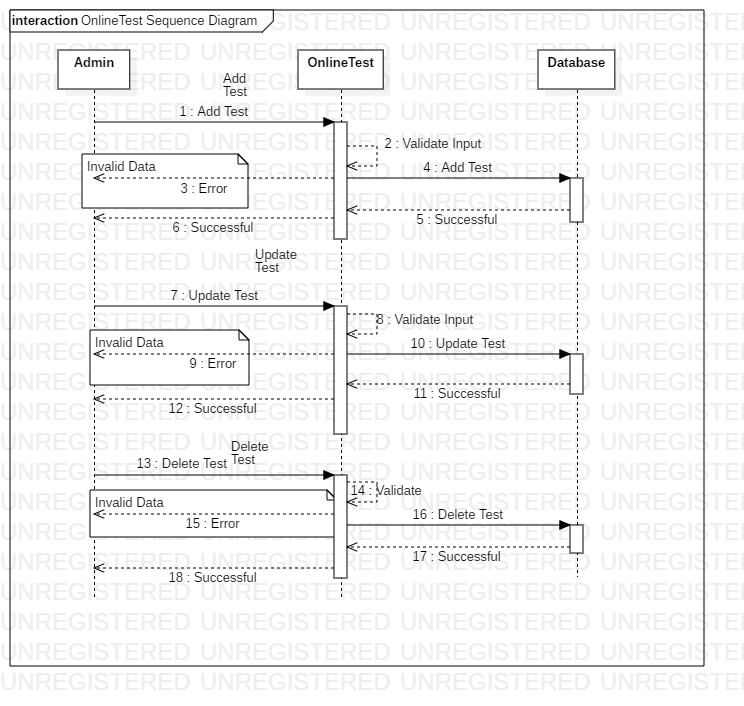
d.questionAnswer : Integer

e.questionMarks : BigDecimal

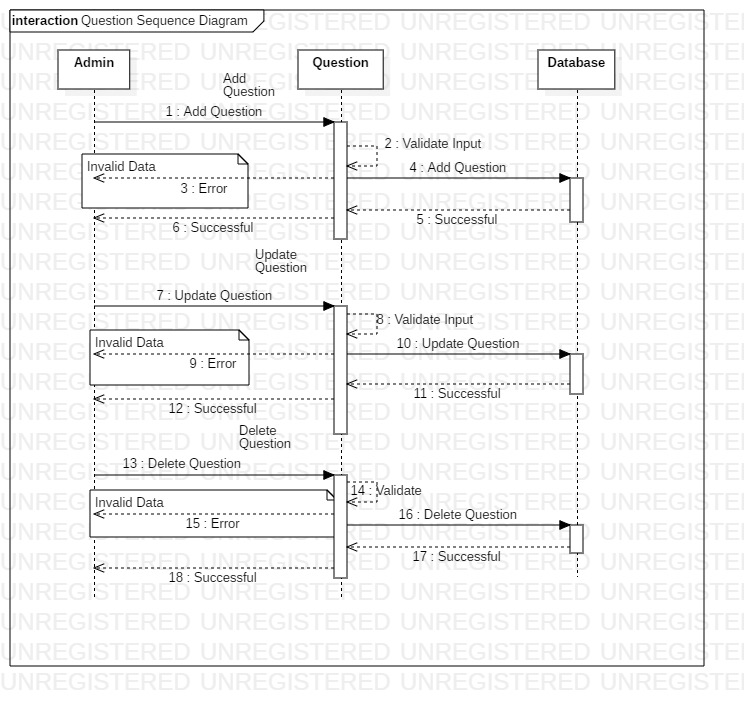
f.chosenAnswer : Integer

g.marksScored : BigDecimal

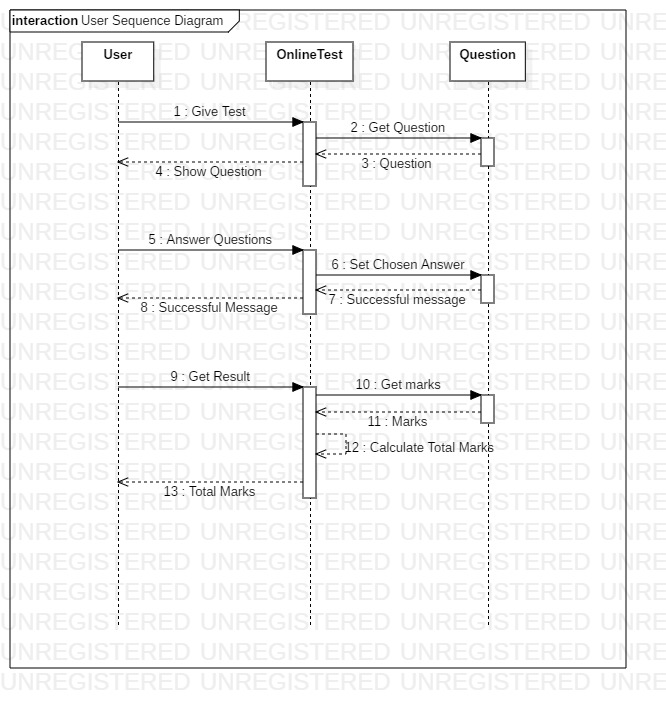
**Diagrams:**

****

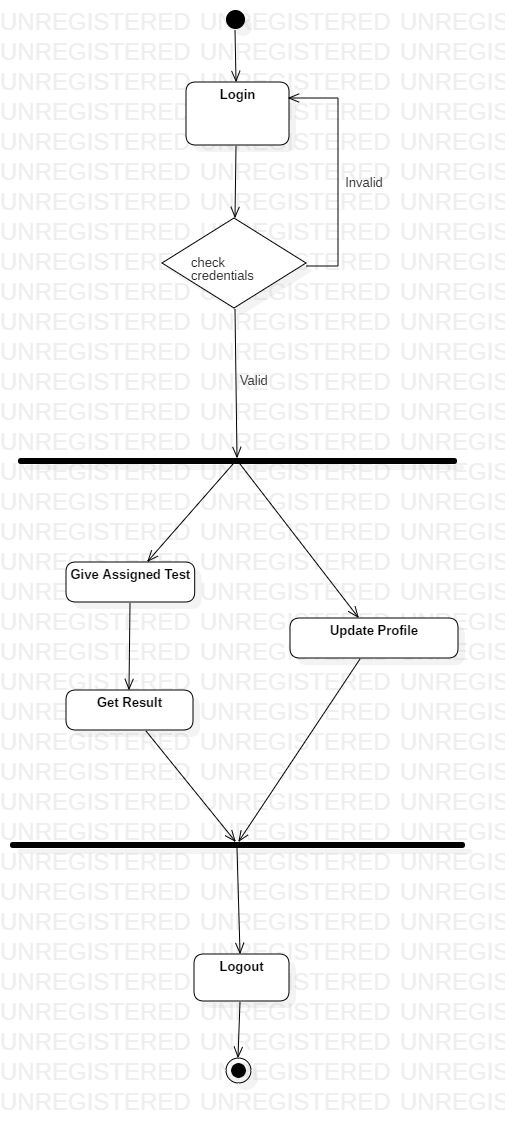
Sequence Diagram for Test

****

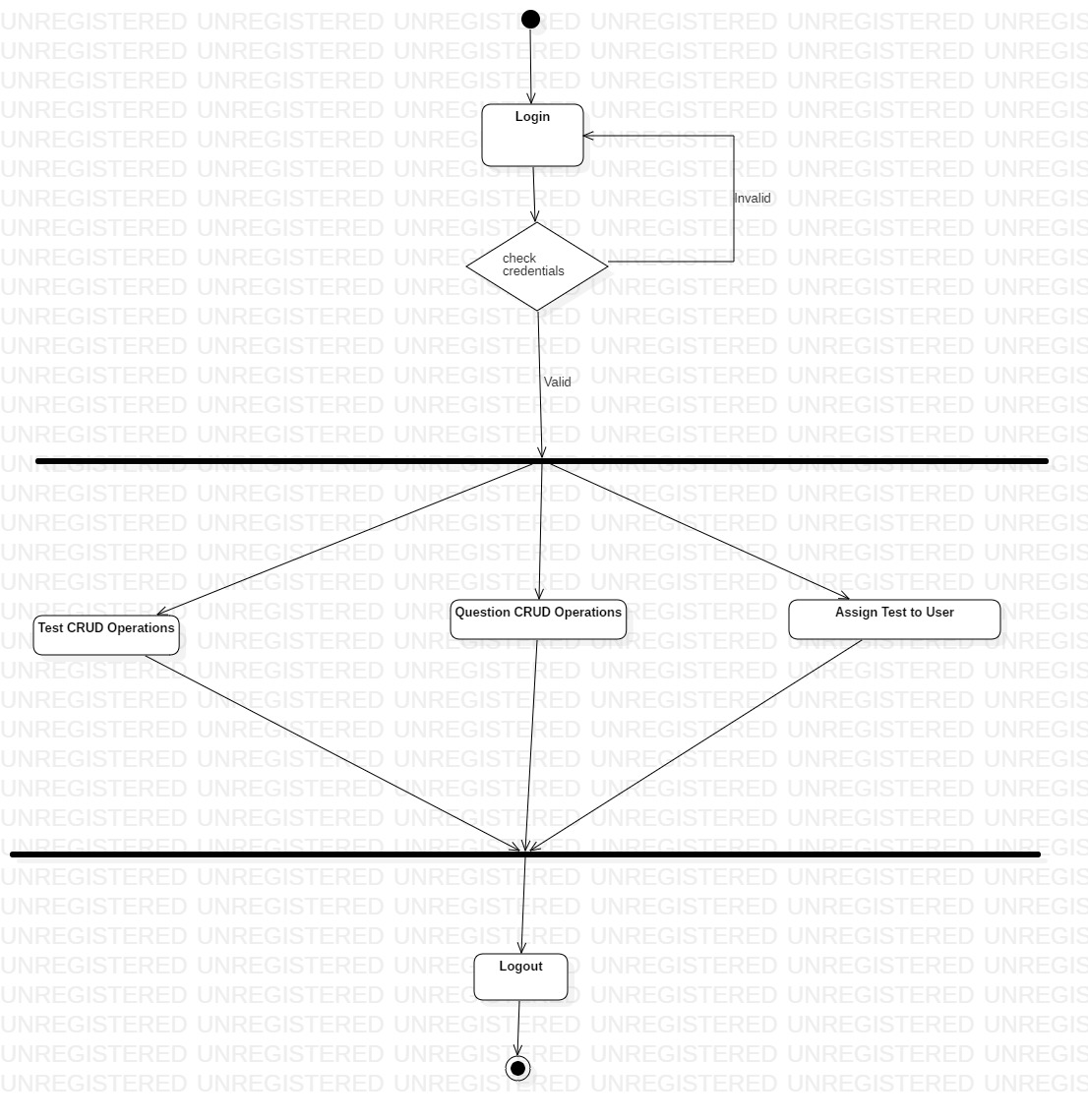
Sequence Diagram for Question

****

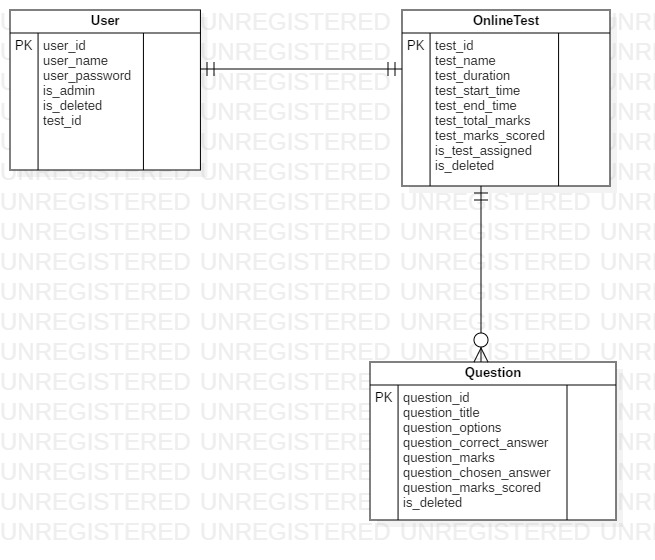
Sequence Diagram for User

****

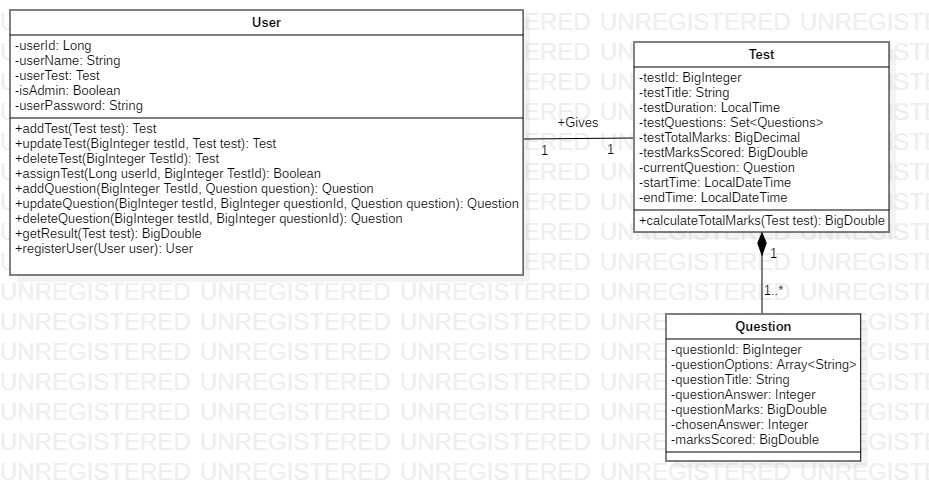
Activity Diagram for User

****

Activity Diagram for Admin

****

ER Diagram

****

Class Diagram

**Methods:**

**1.User**

1.1. addTest(Test test) : Test

If the user is an admin, then only he can add the test. There are two ways to add test : Add through excel or add one by one using the interface.

1.2. updateTest(in BigInteger testId, in Test test): Test

If the user is an admin, then only he can update the test. He can update or delete the test using the interface and can add test using the interface as well as excel.

1.3. deleteTest(in BigInteger TestId): Test

If the user is an admin, then only he can delete the test. This method will delete the test along with all the questions inside it as well.

1.4. assignTest(in Long userId, in BigInteger TestId): Boolean

If the user is an admin, then he can assign a particular test to a user.

1.5. addQuestions(BigInteger TestId,Question question):

Question

This method will be called whenever the admin wants to add a question into a test.

1.6. updateQuestions(BigInteger TestId,Question question) :

Question

This method is called whenever the admin wants to update a question in the given test. It will return null if the question doesn’t exist in the test.

1.7. deleteQuestions(BigInteger TestId,Question question) :

Question

This method is called whenever the admin wants to delete a question in the given test. It will return null if the question doesn’t exist in the test.

1.8. getResult(Test test): BigDecimal

If the user is a student, he can use this method to get the total marks scored by them in the test assigned to them.

**2.Test**

2.1. calculateTotalMarks(): BigDecimal

This method is called whenever the user completes a test. This method will calculate the sum of the marks obtained by the student in each question and store them in the testTotalMarks attribute.

**Relationship:**

Association exists between the User and Test classes as the user can give a test. The relationship is one - to - one relationship as only one user can give a test and one test is assigned to a single user.

Composition exists between Test and Question classes as it represents a whole-part relationship where the part(question) has no individual existence. The relationship is one - to - many relationship as a test can have many questions and many questions can be a part of one test.

**Validations -**

1. userId, testId, questionId should not be null.
2. userName must not be null and its length must be between 5 and 20 characters long.
3. userPassword should have at least one upper case character, one lower case character, one numeric character and one special character and the length of password should be minimum 8 characters.
4. startTime should not be after endTime.
5. endTime should not be in the past.