BAHWAN CYBERTEK

Fresher Batch –5

Project

Online Test Management System

Team Members

Ashish Vishwakarma

Under Guidance

Sachin Anil (Technical Trainer)

Department of

APPLICATION DEVELOPMENT SERVICES –INDIA

Problem Statement:

To develop an Online Test Management System with various defined roles for user as well as admin.

Description:

This project is aimed at developing Online Test Management System for user. This is a web based application that can be accessed throughout the web. This system can be used to add Test, add Questions, assign Test to user & user can give test . This is an integrated system that contains both the user component and the administration component.

Scope:

Using the Online Test Management System, a user 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.

Out of Scope:

The System only involves Multiple Choice Questions and no other type of tests like coding and theory tests. The system does not support a single user giving or being assigned to multiple tests at any given point of time.

Assumptions:

1. It is assumed that the test is only a Multiple Choice Questions(MCQs) test.
2. It is assumed that every question has only four choices and only one of them is correct.
3. A single user can be assigned to a single test at a given point of time.

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

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 negative or null.
2) userName must not be null and the first character should be capitalized.
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) testDuration value should be less than difference of endTime and startTime.
5) startTime should not be after endTime.
6) endTime should not be in the past.