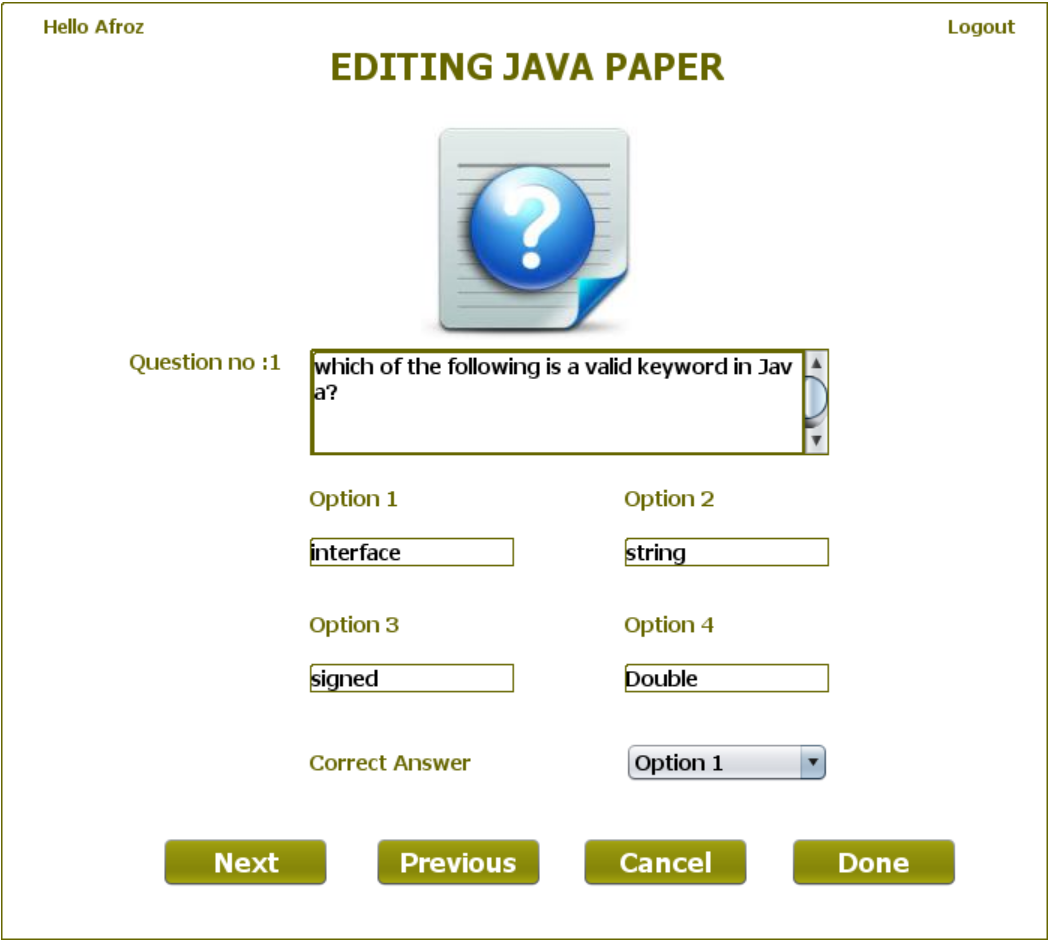
**Designing The EditQuestionsFrame**

****

**STEPS TO BE DONE IN** **EditQuestionsFrame**

In the **EditQuestionsFrame** we need to do following steps:

1. Handle the constructor for proper initialization of class members

2.In the constructor we also have to load the questions from the Database and display the first question as well as it's options

3. Allow the user to **logout**

4. Write code for the "**Next**" Button. When this button is clicked it should:

**a. Validate the inputs. If validation fails it should print the Error Message and return**

**b. Create a new object of Question POJO and initialize it with the details about the question.**

**c. Replace the Question in qstore at the current position with the newly edited Question**

**d. If all the questions have been updated, then display the message asking the user to send them to the database**

**e. Load and show the next question**

5. Write code for the "**Previous**" Button. When this button is clicked it should:

**a. Validate the inputs. If validation fails it should print the Error Message and return**

**b. Create a new object of Question POJO and initialize it with the details about the question.**

**c. Replace the Question in qstore at the current position with the newly edited Question**

**d. If all the questions have been updated, then display the message asking the user to send them to the database**

**e. Load and show the next question**

6. Write code for the "**Done**" Button. When this button is clicked it should:

**a. Send all the Questions to the method updateQuestions( ) of QuestionDAO for getting updated in the database of QUESTIONS table.**

**b. Dispose the current frame and open the EditPaperFrame**

7. Write code for the "**Back**" Button. When this button is clicked it should:

a**. Dispose the current frame and open the OptionsFrame**

**THE TABLES USED IN EditQuestionsFrame**

**1.QUESTIONS**

|  |  |  |
| --- | --- | --- |
| **Column Name** | **Data Type** | **Description** |
| **EXAMID** | **Varchar2(10)** | **Contains id of the paper** |
| **QNO** | **Number(5)** | **Represents the Question No.** |
| **QUESTION** | **Varchar2(100)** | **Represents Question Description** |
| **ANSWER1** | **Varchar2(30)** | **Option 1** |
| **ANSWER2** | **Varchar2(30)** | **Option 2** |
| **ANSWER3** | **Varchar2(30)** | **Option 3** |
| **ANSWER4** | **Varchar2(30)** | **Option 4** |
| **CORRECT\_ANSWER** | **Varchar2(10)** | **The actual correct answer** |
| **LANGUAGE** | **Varchar2(10)** | **Programming language name** |

**THE POJO CLASSES USED IN EditQuestionsFrame**

1. The **Exam** POJO

2. The **Question** POJO

**THE DAO CLASSES USED IN EditQuestionsFrame**

1. The **QuestionDAO**

**HOW TO LOAD AND SHOW THE QUESTIONS:**

The first step we need to do in the **constructor** of **EditQuestionsFrame** is to load all the questions from the database and show the first question.

To do this we will create a method called **getQuestionsByExamId( )** in the **QuestionDAO**. The prototype of the method is:

***public static ArrayList <Question> getQuestionsByExamId(String examId)throws SQLException***

To call this method from **EditQuestionsFrame** we will create a method called **loadQuestions( )** in the **EditQuestionsFrame**

We will also create a method called **showQuestions( )** in the **EditQuestionsFrame** for showing the loaded question

**Handling the constructor:**

The constructor should do the following:

a. Display the username

b. Create a **QuestionStore** object

c. Create a **HashMap** object

d. Set the name of the Language in a label

e. Load all the questions from the database and show the first question.