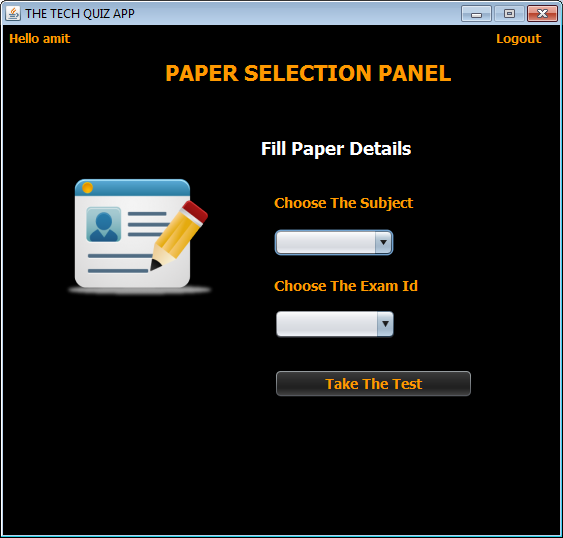
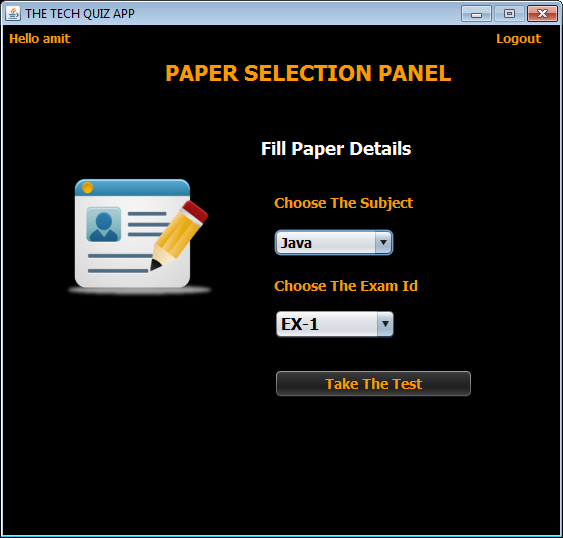
**Designing The ChoosePaperFrame**

****

****

**STEPS TO BE DONE IN** **ChoosePaperFrame**

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

1. Display **username** on the top left

2. Allow the user to **logout**

3. Write code for selecting all **Exam Id** of the **selected subject** from the database ***as soon as subject is selected.***

Since subject names are displayed in the **JComboBox** , we need to handle it's **actionPerformed( )** method as it is the event that gets called as soon as the user selects an item in the **JComboBox**. We need to perform following steps in the **actionPerformed( )** method of the **JComboBox**:

**a. Verify whether the user has selected any subject or not. If not it should print the Error Message and return.**

**b. Check whether the subject selected by the user has any Paper set or not. If not paper has been set for the selected subject then the code should display the message Sorry! No exam set for subject XXX**

**c. This will be done by calling the method isPaperSet() of ExamDAO which accepts a subject name as argument and returns true if any paper has been set for that subject otherwise it returns false.**

**d.Then we will call getExamIdBySubject( ) method of ExamDAO which accepts 2 arguments , userid and subject name and returns the list of those Exam-Id of the given subject for which student has not appeared yet. If returned list is empty the code should display the message You have appeared for all the exams of this subject otherwise it should load all the examids in the combo box.**

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

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

**b. If validation passes , it should fetch total number of questions of this exam from the database by calling the method getQuestionCountByExam( ) of the ExamDAO class.**

**c. Finally it should create an Exam object , fill the** **Exam** **details**(**Exam Id**,**subject**,**total no of questions**) **and pass this object to the** **TakeTestFrame**

**d. Dispose the current frame**

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

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

**THE TABLES USED IN ChoosePaperFrame**

**1. EXAM**

|  |  |  |
| --- | --- | --- |
| **Column Name** | **Data Type** | **Description** |
| **EXAMID** | **Varchar2(10)** | **Contains id of the paper** |
| **LANGUAGE** | **Varchar2(10)** | **Contains the programming language name** |
| **TOTAL\_QUESTION** | **Number(4)** | **Contains total number of question present** |

**THE POJO CLASSES USED IN ChoosePaperFrame**

1. The **Exam** POJO

**THE DAO CLASSES USED IN ChoosePaperFrame**

1. The **ExamDAO**

**A.** The first method called will be **isPaperPresent( )** of **ExamDAO** to check whether the subject chosen has any paper set or not.

The protoype of the method is:

***public static boolean isPaperSet(String subject)throws SQLException***

***Following is the code:***

***public static boolean isPaperSet(String subject)throws SQLException{***

***String qry="Select examid from Exam where language=? ";***

***Connection conn=DBConnection.getConnection();***

***PreparedStatement ps=conn.prepareStatement(qry);***

***ps.setString(1,subject);***

***ResultSet rs=ps.executeQuery();***

***return rs.next();***

***}***

**B**. The next method defined will be **getExamIdBySubject( )** of **ExamDAO** to get all **Exam-Id** of the subject name passed as argument for which the user has not appeared yet.

The protoype of the method is:

***public static ArrayList<String> getExamIdBySubject(String userid,String subject)throws SQLException***

***Following is the code:***

***public static ArrayList<String> getExamIdBySubject(String userid,String subject)throws SQLException{***

***String qry="Select examid from Exam where language=? minus Select examid from performance where userid=?";***

***ArrayList<String> examList=new ArrayList<>();***

***Connection conn=DBConnection.getConnection();***

***PreparedStatement ps=conn.prepareStatement(qry);***

***ps.setString(1,subject);***

***ps.setString(2,userid);***

***ResultSet rs=ps.executeQuery();***

***while(rs.next()){***

***examList.add(rs.getString(1));***

***}***

***return examList;***

***}***

**C.** Now we have to create the **getQuestionCountByExam( )** method in the **ExamDAO** to get the total number of questions of a particular exam from the **EXAM** table.

The prototype of the method is:

***public static int getQuestionCountByExam(String examId)throws SQLException***

Following is the code:

***public static int getQuestionCountByExam(String examId)throws SQLException{***

***String qry="select total\_question from Exam where examid=?";***

***Connection conn=DBConnection.getConnection();***

***PreparedStatement ps=conn.prepareStatement(qry);***

***ps.setString(1,examId);***

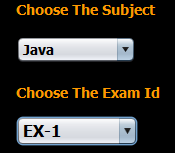
***ResultSet rs=ps.executeQuery();***

***rs.next();***

***int rowCount=rs.getInt(1);***

***return rowCount;***

***}***

**WRITING THE CODE FOR JCOMBOBOX  IN ChoosePaperFrame**

***public boolean validateInput(){***

***int selectedIndex=jcSubject.getSelectedIndex();***

***if(selectedIndex==0)***

***return false;***

***subjectName=jcSubject.getSelectedItem().toString();***

***return true;***

***}***

**WRITING THE CODE FOR BUTTON Button.png IN ChoosePaperFrame**