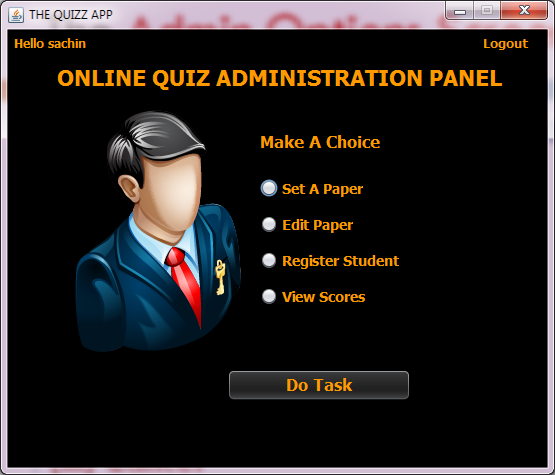
**Designing The AdminOptionsFrame**

****

**STEPS TO BE DONE IN** **AdminOptionsFrame**

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

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

2. Allow the user to **logout**

3. Provide the following options:

**Set Paper**

**Edit Paper**

**Register Student**

**ViewScores**

**Displaying The Username**

We can recall that we saved the **username** and **usertype** in the static fields of the class **UserProfile** . So in the constructor of **AdminOptionsFrame** we will call the **getUsername( )** method of the class **UserProfile** to get the username and display it in the Label.

***public AdminOptionsFrame() {***

***initComponents();***

***this.setLocationRelativeTo(null);***

***lblUsername.setText("Hello "+UserProfile.getUSername());***

***}***

**Allowing The User To Logout**

To allow the user to logout , we will provide a label at the top right corner with the following functionality:

**a. When the user takes the mouse over the label it's font settings will change**

**b. When the user removes the mouse from the label the font settings will be retained**

**c. When the user will click on the label , the user will be logged out i.e. we will dispose the current frame and open the LoginFrame.**

All the three behaviours require handling **MouseEvents**:

**a. mouseEntered ()**

**b. mouseExited ()**

**c. mouseClicked ()**

***private void lblLogoutMouseClicked(java.awt.event.MouseEvent evt) {***

***this.dispose();***

***LoginFrame loginFrame=new LoginFrame();***

***loginFrame.setVisible(true);***

***}***

***private void lblLogoutMouseEntered(java.awt.event.MouseEvent evt) {***

***lblLogout.setForeground(Color.WHITE);***

***Font f=new Font("Tahoma",Font.ITALIC,12);***

***lblLogout.setFont(f);***

***}***

***private void lblLogoutMouseExited(java.awt.event.MouseEvent evt) {***

***lblLogout.setForeground(new Color(255,153,0));***

***Font f=new Font("Tahoma",Font.BOLD,12);***

***lblLogout.setFont(f);***

***}***

**Handling The Options**

The next task is to allow the user to select an options amongst **Set Paper,Edit Paper, Register Student and**

**ViewScores**

To do this we need to handle the "Do Task" Button in the following way:

**a. We will first validate whether any option has been selected or not.**

**b. If not , then we will generate an error message and return.**

**c. Otherwise , it will detect the radio button selected and dispose the current frame and open the next frame.**

***private void btnTaskActionPerformed(java.awt.event.ActionEvent evt) {***

***String task=getAction();***

***if(task==null){***

***JOptionPane.showMessageDialog(null, "Please Choose An Option","Error!",JOptionPane.ERROR\_MESSAGE);***

***return;***

***}***

***JFrame jf=null;***

***if(task.equals("Set A Paper"))***

***jf=new SetPaperFrame();***

***else if(task.equals("Edit Paper"))***

***{***

***jf=new EditPaperFrame();***

***}***

***else if(task.equals("Register Student"))***

***{***

***jf=new RegisterStudentFrame();***

***}***

***else***

***{***

***jf=new ViewScoresFrame();***

***}***

***jf.setVisible(true);***

***this.dispose();***

***}***

***private String getAction() {***

***if(jrbSetpaper.isSelected())***

***return jrbSetpaper.getText();***

***else if(jrbEditpaper.isSelected())***

***return jrbEditpaper.getText();***

***else if(jrbEditpaper.isSelected())***

***return jrbEditpaper.getText();***

***else if(jrbRegister.isSelected())***

***return jrbRegister.getText();***

***else if(jrbViewscore.isSelected())***

***return jrbViewscore.getText();***

***return null;***

***}***