**Designing The ChangePasswordFrame**

****

**STEPS TO BE DONE IN** **ChangePasswordFrame**

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

1. Display **username** on the top left as well as in the **JTextField**

2. Allow the user to **logout**

3. Write code for the "**Update Password**" Button . When this button is clicked it should:

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

**b. Update the password by calling the method changePassword( ) of UserDAO**

**d. If record is successfully added it should display the message "Password Successfully Changed" otherwise it should display the message "Password Not Changed!"**

**e. It should handle all the exceptions also**

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

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

**HOW TO DISPLAY NAME IN JTEXTFIELD**

***public class ChangePasswordFrame extends javax.swing.JFrame {***

***private String userid,password;***

***public ChangePasswordFrame() {***

***initComponents();***

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

***txtUserid.setText(UserProfile.getUSername());***

***}***

**THE TABLES USED IN ChangePasswordFrame**

**1.USERS**

|  |  |  |
| --- | --- | --- |
| **Column Name** | **Data Type** | **Description** |
| **USERID** | **Varchar2(10)** | **Contains Userid of the admin or student** |
| **PASSWORD** | **Varchar2(10)** | **Contains password** |
| **USERTYPE** | **Varchar2(10)** | **Stores "Student" or "Admin"** |

**THE DAO CLASSES USED IN ChangePasswordFrame**

1. The **UserDAO**

**HOW TO CHANGE THE PASSWORD:**

To change the password of a student in the database we need to **update** his/her record in the **USER** table.

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

***public static boolean changePassword(String userid,String password)throws SQLException***

This method will do 2 things:

**a. Try to update the password.**

**b. If updation is successful , then it will return true otherwise it will return false.**

Following is the code:

***public static boolean changePassword(String userid,String password)throws SQLException{***

***String qry="Update users set password=? where userid=?";***

***boolean status=true;***

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

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

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

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

***int ans=ps.executeUpdate();***

***if(ans==0)***

***status=false;***

***return status;***

***}***

**DESIGNING THE validateInputs( ) METHOD:**

This method should do the following:

**1. If any of the inputs is empty , it should return 0;**

**2. if userid or password is of less than 4 characters it should return -1**

**3. If password and retype password don't match**

**it should return -2**

**4.If everything is correct it should return 1**

Based on above points the prototype of **validateInputs( )** is:

***private int validateInputs()***

Following is the code:

***public int validateInputs() {***

***userid=txtUserid.getText();***

***char []pwd=txtPassword.getPassword();***

***char []repwd=txtRetypePassword.getPassword();***

***if(userid.isEmpty()|| pwd.length==0 || repwd.length==0 )***

***return 0;***

***else***

***{***

***password=String.valueOf(pwd);***

***String repassword=String.valueOf(repwd);***

***if(userid.length()<4 || password.length()<4)***

***return -2;***

***if(password.equals(repassword))***

***return 1;***

***else***

***return -1;***

***}***

***}***

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

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

***int response=validateInputs();***

***if(response==0)***

***{***

***JOptionPane.showMessageDialog(null, "Username or Password cannot be left blank!","Error!",JOptionPane.ERROR\_MESSAGE);***

***return;***

***}***

***if(response==-1)***

***{***

***JOptionPane.showMessageDialog(null, "Password and Retype Password do not match!","Error!",JOptionPane.ERROR\_MESSAGE);***

***return;***

***}***

***if(response==-2)***

***{***

***JOptionPane.showMessageDialog(null, "Userid and Password should be atleast 4 characters long!","Error!",JOptionPane.ERROR\_MESSAGE);***

***return;***

***}***

***try***

***{***

***boolean status=UserDAO.changePassword(userid,password);***

***if(status)***

***{***

***JOptionPane.showMessageDialog(null, "Password Successfully Changed!","Userid: "+userid+"\nPassword:"+password,JOptionPane.INFORMATION\_MESSAGE);***

***StudentOptionsFrame optionFrame=new StudentOptionsFrame();***

***optionFrame.setVisible(true);***

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

***}***

***else***

***JOptionPane.showMessageDialog(null, "Password Not Changed!","Password Change Denied!",JOptionPane.ERROR\_MESSAGE);***

***}***

***catch(SQLException ex)***

***{***

***JOptionPane.showMessageDialog(null, "Error while connecting to DB!","Exception!",JOptionPane.ERROR\_MESSAGE);***

***ex.printStackTrace();***

***}***

***}***

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

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

***StudentOptionsFrame optionFrame=new StudentOptionsFrame();***

***optionFrame.setVisible(true);***

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

***}***