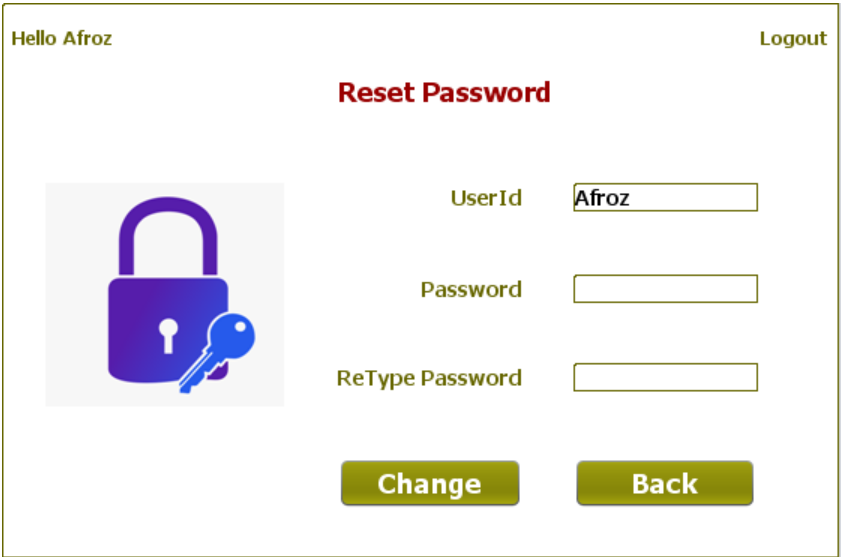
**Designing The ChangePasswordFrame**

****

**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**

**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.**

**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()***

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

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