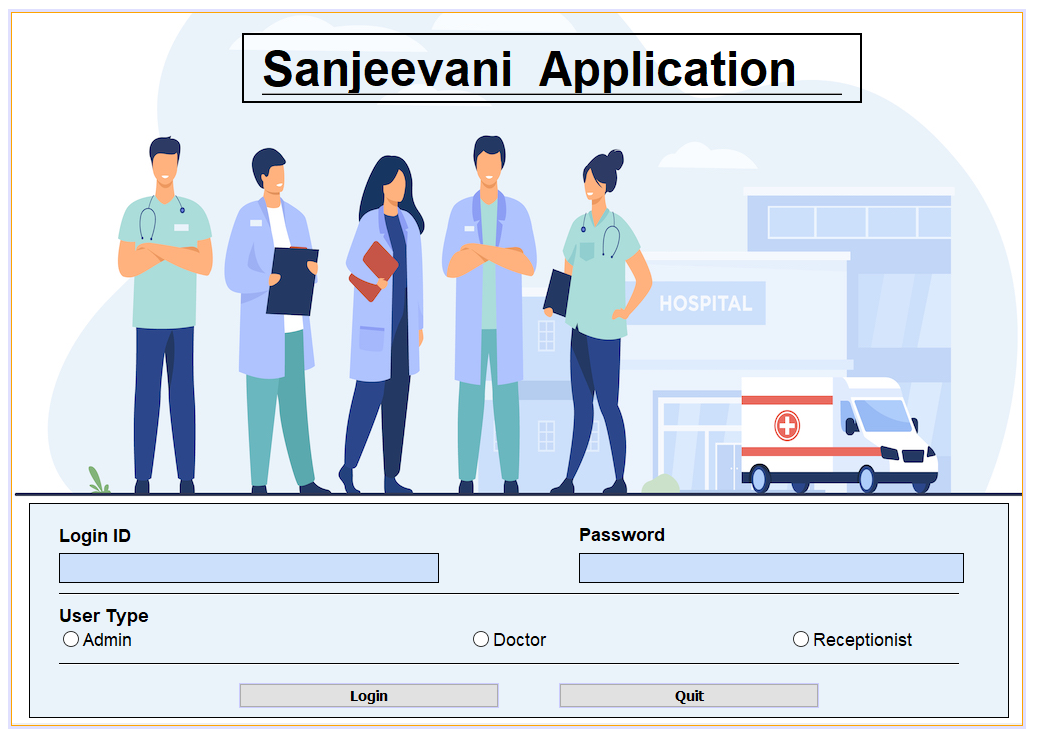
Java SE Project

Hospital Management System Application

***Login Frame Documentation***

1. **LoginFrame -**



**2. STEPS TO BE DONE IN** **LoginFrame**

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

**A. Coding for Login Button**

**B. Coding for Quit Button**

1. **Validating all the inputs.**
2. **Concept for getting selected user type**
3. **Concept for Login Button :**

We have to first **validate** all the **inputs** for that we will have the **validateInput()** and **getUserType() method** which we will discuss later in this section. After **validating** the user, we will check the given **loginId** and **password** in the **database**. If the user does not exist then we will generate the dialog box with the error **"Login Denied"** followed by the error message **"Invalid UserId/Password"** or else we will store the user details (userid, password, usertype) in the UserProfile POJO and then we will check the user type and open the frame according to the user type and dispose this frame.

**Coding for User Pojo :**

**public class User {**

**private String userId;**

**private String password;**

**private String userType;**

**public String getUserId() {**

**return userId;**

**}**

**public void setUserId(String userId) {**

**this.userId = userId;**

**}**

**public String getPassword() {**

**return password;**

**}**

**public void setPassword(String password) {**

**this.password = password;**

**}**

**public String getUserType() {**

**return userType;**

**}**

**public void setUserType(String userType) {**

**this.userType = userType;**

**}**

**}**

**Coding for validateUser :**

**public class UserDao{**

**public static String validateUser(User user) throws SQLException{**

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

**String qry = "Select user\_name from Users where login\_id=? and password=? and user\_type=?";**

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

**ps.setString(1,user.getUserId());**

**ps.setString(2,user.getPassword());**

**ps.setString(3,user.getUserType());**

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

**String username =null;**

**if(rs.next()){**

**username=rs.getString(1);**

**}**

**return username;**

**}**

**}**

**Coding for Login Button :**

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

**boolean isValidInput=validateInput();**

**if(isValidInput==false){**

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

**return;**

**}**

**String userType=getUserType();**

**if(userType==null){**

**JOptionPane.showMessageDialog(null,"Please Choose User type","Error",JOptionPane.ERROR\_MESSAGE);**

**return;**

**}**

**try{**

**User user=new User();**

**user.setUserId(userId);**

**String pwd=new String(PasswordEncryption.getEncryptedPassword(password));**

**user.setPassword(pwd);**

**System.out.println(pwd);**

**user.setUserType(userType);**

**System.out.println("username :"+userId+" password :"+password+" usertype :"+userType);**

**String username=UserDao.validateUser(user);**

**if(username!= null)**

**{**

**JOptionPane.showMessageDialog(null,"Login Accepted","Welcome "+username,JOptionPane.INFORMATION\_MESSAGE);**

**UserProfile.setUsername(username);**

**UserProfile.setUserType(userType);**

**UserProfile.setUserId(userId);**

**if(userType.equalsIgnoreCase("admin")){**

**AdminOptionsFrame adminOptionsFrame = new AdminOptionsFrame();**

**adminOptionsFrame.setVisible(true);**

**}**

**else if(userType.equalsIgnoreCase("receptionist")){**

**ReceptionistOptionsFrame r=new ReceptionistOptionsFrame();**

**r.setVisible(true);**

**}**

**else**

**{**

**DoctorsOptionsFrame d = new DoctorsOptionsFrame();**

**d.setVisible(true);**

**}**

**this.dispose();**

**}**

**else**

**JOptionPane.showMessageDialog(null,"Invalid UserId/Password","Login Denied",JOptionPane.ERROR\_MESSAGE);**

**}**

**catch(Exception ex){**

**JOptionPane.showMessageDialog(null,"No UserId/Password Found! Try Again."+ex,"Exception",JOptionPane.ERROR\_MESSAGE);**

**ex.printStackTrace();**

**}**

**}**

1. **Concept for Quit Button :**

On the click of quit button we will show a **confirmDialog** if the user click the yes button we will **close** the **connection** from the **database and end the program by System.exit(0)** or else we will do nothing.

**Coding for Quit Button :**

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

**int ans;**

**ans=JOptionPane.showConfirmDialog(null,"Are You Sure?","Quitting!",JOptionPane.YES\_NO\_OPTION,JOptionPane.QUESTION\_MESSAGE);**

**if (ans==JOptionPane.YES\_OPTION){**

**DBConnection.closeConnection();**

**System.exit(0);**

**}**

**}**

1. **Concept for validating all the inputs fields :**

Check whether all the **inputs fields** are **filled/selected** or not, If they are not then this method will **return false** or else we will **store** the **userId** and **password** in the **data member** of this **class**.

**Coding for validating all the inputs fields :**

**private boolean validateInput(){**

**userId=txtUserId.getText();**

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

**if(userId.isEmpty()|| pwd.length==0)**

**return false;**

**else**

**{**

**password=new String(pwd);**

**return true;**

**}**

**}**

1. **Concept for getting selected user type :**

Jradiobutton has a method called **isSelected()** which will tell you which **Jradiobutton** is **currently selected**. **Jradiobutton** has a method called **getText()** which will give you the **text** of the **selected Jradiobutton** which will be **return** by this **method** or else this **method** will **return null.**

**Coding for getting selected user type :**

**private String getUserType(){**

**if(jrbAdmin3.isSelected())**

**return jrbAdmin3.getText().toUpperCase();**

**else if(jrbReceptionist3.isSelected())**

**return jrbReceptionist3.getText().toUpperCase();**

**else if(jrbDoctors.isSelected()){**

**return jrbDoctors.getText().toUpperCase();**

**}**

**else**

**return null;**

**}**