```java
package view;

import java.awt.event.ActionEvent;

/**
 * Class ResetPasswordInterface to construct interface for user to reset their password.
 * @author Daria Vekic (Student ID: 586661)
 *
 */
public class ResetPasswordInterface  extends JFrame {

    private static final long serialVersionUID = 1L;
    // Instance fields
    private JPanel contentPane;
    private JTextField usernameTextField;
    private JPasswordField oldPasswordTextField;
    private JPasswordField newPasswordTextField;
    private JButton resetPWBtn;
    //Controller instance for validation and data flow control
    private LogInController loginController = new LogInController();

    /**
     * Constructor method to create user interface that allows for resetting of password.
     * @param employees the List of Employees needed to create LogInInterface if reset successful.
     * @param cases the List of Cases needed to create LogInInterface if reset successful.
     * @param clients the List of Clients needed to create LogInInterface if reset successful.
     */
    public ResetPasswordInterface(ArrayList<Employee> employees,
            ArrayList<Case> cases, ArrayList<Client> clients) {
        constructResetPasswordInterface(employees, cases, clients);
    } //end constructor method

    /**
     * Method to create the Reset Password interface.
     * @param employees the List of Employees needed to create LogInInterface if reset successful.
     * @param cases the List of Cases needed to create LogInInterface if reset successful.
     * @param clients the List of Clients needed to create LogInInterface if reset successful.
     */
    private void constructResetPasswordInterface(ArrayList<Employee> employees,
            ArrayList<Case> cases, ArrayList<Client> clients) {
        setTitle("Set Password");
        setDefaultCloseOperation(JFrame.DISPOSE_ON_CLOSE);
        setBounds(100, 100, 339, 264);
        contentPane = new JPanel();
        contentPane.setBorder(new EmptyBorder(5, 5, 5, 5));

        setContentPane(contentPane);
        contentPane.setLayout(null);

        JLabel usernameLabel = new JLabel("Enter username:");
        usernameLabel.setBounds(36, 53, 107, 14);
        contentPane.add(usernameLabel);

        JLabel oldPasswordLabel = new JLabel("Old password:");
```

```
70          oldPasswordLabel.setBounds(36, 95, 88, 14);
71          contentPane.add(oldPasswordLabel);
72
73          JLabel newPasswordLabel = new JLabel("New password:");
74          newPasswordLabel.setBounds(36, 135, 88, 14);
75          contentPane.add(newPasswordLabel);
76
77          usernameTextField = new JTextField();
78          usernameTextField.setBounds(151, 50, 128, 20);
79          contentPane.add(usernameTextField);
80          usernameTextField.setColumns(10);
81
82          oldPasswordTextField = new JPasswordField();
83          oldPasswordTextField.setColumns(10);
84          oldPasswordTextField.setBounds(151, 92, 128, 20);
85          contentPane.add(oldPasswordTextField);
86
87          newPasswordTextField = new JPasswordField();
88          newPasswordTextField.setColumns(10);
89          newPasswordTextField.setBounds(151, 132, 128, 20);
90          contentPane.add(newPasswordTextField);
91
92          resetPWBtn = new JButton("Reset Password");
93          resetPWBtn.addActionListener(new ActionListener() {
94          public void actionPerformed(ActionEvent event) {
95              resetPWBtnActionPerformed(event, employees, cases, clients);
96          } //end method actionPerformed
97          } //end method ActionListener
98          ); //end call to resetPWBtnActionPerformed ActionListener
99          resetPWBtn.setBounds(104, 182, 153, 23);
100         contentPane.add(resetPWBtn);
101     } // end method constructResetPasswordInterface
102
103     /**
104      * Routine to be followed if Reset Password button is pressed.
105      * @param event the Reset Password button is pressed.
106      * @param employees the List of Employees needed to create LogInInterface if reset successful.
107      * @param cases the List of Cases needed to create LogInInterface if reset successful.
108      * @param clients the List of Clients needed to create LogInInterface if reset successful.
109      */
110     private void resetPWBtnActionPerformed(ActionEvent event, ArrayList<Employee> employees,
111             ArrayList<Case> cases, ArrayList<Client> clients) {
112         if(loginController.resetPassword(usernameTextField, oldPasswordTextField, newPasswordTextField)) {
113             this.dispose();
114             LogInInterface login = new LogInInterface(employees, clients, cases);
115             login.setVisible(true);
116         } //endif
117     } //end method resetPWBtnActionPerformed
118 } //end class ResetPasswordInterface
```