



Name	Tanzeel Hussain Muhammad Sohail
ID	38529 38429
Subject	Object-oriented programming (OOP)
Section	E
Submit To	Taha Ali
Project Report	

Polling System Java Application

Introduction

The Polling System Java Application is designed to facilitate a simple and secure voting process for users. The system features user registration, user login, vote casting, and an admin panel for viewing results. The application is built using Java Swing for the graphical user interface (GUI) and utilizes in-memory storage to manage users and votes.

Objectives

The main objectives of the Polling System are:

1. To provide a user-friendly interface for casting votes.
2. To ensure secure login for users and admins.
3. To allow only registered users to vote.
4. To allow the admin to view voting results securely.

Features

- **User Registration:** Allows new users to register with a unique username and password.
- **User Login:** Authenticates registered users to allow them to cast their vote.
- **Vote Casting:** Users can vote for one of the given candidates.
- **Admin Login:** Secure login for the admin to access the admin panel.
- **View Results:** The admin can view the voting results after logging in.

System Design

In-Memory Storage

- **Users:** Stored in a HashMap with the username as the key and the password as the value.
- **Votes:** Stored in a HashMap with the candidate's name as the key and the vote count as the value.
- **Candidates:** A fixed array of candidate names.

Classes and Responsibilities

1. **PollingSystem:** Initializes the application, stores users, votes, and candidate names.
2. **WelcomeFrame:** Main window with buttons for user registration, user login, and admin login.
3. **AdminLoginFrame:** Allows the admin to log in.
4. **AdminPanelFrame:** Displays the admin panel with an option to view results.
5. **UserRegistrationFrame:** Allows new users to register.
6. **UserLoginFrame:** Allows registered users to log in.
7. **VoteFrame:** Allows logged-in users to cast their vote.

8. **ResultFrame**: Displays voting results to the admin.

Class Diagrams

1. PollingSystem

- Attributes:
 - votes: HashMap<String, Integer>
 - users: HashMap<String, String>
 - adminUsername: String
 - adminPassword: String
 - candidates: String[]
- Methods:
 - main(String[] args)

2. WelcomeFrame

- Attributes: None
- Methods: Constructor

3. AdminLoginFrame

- Attributes: None
- Methods: Constructor

4. UserRegistrationFrame

- Attributes: None
- Methods: Constructor

5. UserLoginFrame

- Attributes: None
- Methods: Constructor

6. VoteFrame

- Attributes: votedUsers: HashSet<String>
- Methods: Constructor

7. ResultFrame

- Attributes: None
- Methods: Constructor

User Interface

The application uses Java Swing to create a graphical user interface with the following components:

- **JFrame:** Main window for each feature.
- **JButton:** Buttons for actions like login, register, vote, etc.
- **TextField & JPasswordField:** Text fields for user input.
- **JRadioButton & ButtonGroup:** Radio buttons for selecting candidates.
- **TextArea:** Text area for displaying results.
- **JOptionPane:** Pop-up dialogs for messages and input prompts.

Code Overview

Main Class: PollingSystem

```
java
import javax.swing.*;
import java.awt.*;
import java.util.HashMap;

public class PollingSystem {
    static HashMap<String, Integer> votes = new HashMap<>();
    static HashMap<String, String> users = new HashMap<>();
    static String adminUsername = "admin@gmail.com";
    static String adminPassword = "admin";
    static String[] candidates = {"Tanzeel Hussain", "Muhammad Sohail",
    "Taha Ali"};

    public static void main(String[] args) {
        for (String candidate : candidates) {
            votes.put(candidate, 0);
        }
        SwingUtilities.invokeLater(() -> new WelcomeFrame());
    }
}
```

WelcomeFrame Class

```
java
import javax.swing.*;
import java.awt.*;

class WelcomeFrame extends JFrame {
    WelcomeFrame() {
        setTitle("Polling System");
        setSize(350, 200);
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        setLayout(new FlowLayout());

        JButton adminButton = new JButton("Admin Login");
        adminButton.addActionListener(e -> new AdminLoginFrame());

        JButton userRegisterButton = new JButton("User Registration");
        userRegisterButton.addActionListener(e -> new
        UserRegistrationFrame());

        JButton userLoginButton = new JButton("User Login");
        userLoginButton.addActionListener(e -> new UserLoginFrame());

        add(adminButton);
        add(userRegisterButton);
    }
}
```

```

        add(userLoginButton);

        setLocationRelativeTo(null);
        setVisible(true);
    }
}

```

AdminLoginFrame Class

```

java
import javax.swing.*;
import java.awt.*;

class AdminLoginFrame extends JFrame {
    AdminLoginFrame() {
        setTitle("Admin Login");
        setSize(300, 150);
        setLayout(new GridLayout(3, 2));

        JTextField usernameField = new JTextField();
        JPasswordField passwordField = new JPasswordField();

        JButton loginButton = new JButton("Login");
        loginButton.addActionListener(e -> {
            String username = usernameField.getText();
            String password = new String(passwordField.getPassword());

            if (PollingSystem.adminUsername.equals(username) &&
                PollingSystem.adminPassword.equals(password)) {
                new ResultFrame();
                dispose();
            } else {
                JOptionPane.showMessageDialog(this, "Invalid
credentials!");
            }
        });

        add(new JLabel("Username:"));
        add(usernameField);
        add(new JLabel("Password:"));
        add(passwordField);
        add(loginButton);

        setLocationRelativeTo(null);
        setVisible(true);
    }
}

```

UserRegistrationFrame Class

```

java
import javax.swing.*;
import java.awt.*;

class UserRegistrationFrame extends JFrame {
    UserRegistrationFrame() {
        setTitle("User Registration");
        setSize(300, 150);
        setLayout(new GridLayout(3, 2));
    }
}

```

```

JTextField usernameField = new JTextField();
JPasswordField passwordField = new JPasswordField();

JButton registerButton = new JButton("Register");
registerButton.addActionListener(e -> {
    String username = usernameField.getText();
    String password = new String(passwordField.getPassword());

    if (!username.isEmpty() && !password.isEmpty()) {
        if (!PollingSystem.users.containsKey(username)) {
            PollingSystem.users.put(username, password);
            JOptionPane.showMessageDialog(this, "User Registered
Successfully!");
            dispose();
        } else {
            JOptionPane.showMessageDialog(this, "Username already
exists!");
        }
    } else {
        JOptionPane.showMessageDialog(this, "Please fill in all
fields!");
    }
});

add(new JLabel("Username:"));
add(usernameField);
add(new JLabel("Password:"));
add(passwordField);
add(registerButton);

setLocationRelativeTo(null);
setVisible(true);
}
}

```

UserLoginFrame Class

```

java
import javax.swing.*;
import java.awt.*;

class UserLoginFrame extends JFrame {
    UserLoginFrame() {
        setTitle("User Login");
        setSize(300, 150);
        setLayout(new GridLayout(3, 2));

        JTextField usernameField = new JTextField();
        JPasswordField passwordField = new JPasswordField();

        JButton loginButton = new JButton("Login");
        loginButton.addActionListener(e -> {
            String username = usernameField.getText();
            String password = new String(passwordField.getPassword());

            if (PollingSystem.users.containsKey(username) &&
PollingSystem.users.get(username).equals(password)) {
                JOptionPane.showMessageDialog(this, "Login Successful!");
                new VoteFrame(username);
            }
        });
    }
}

```

```

        dispose();
    } else {
        JOptionPane.showMessageDialog(this, "Invalid
credentials!");
    }
});

add(new JLabel("Username:"));
add(usernameField);
add(new JLabel("Password:"));
add(passwordField);
add(loginButton);

setLocationRelativeTo(null);
setVisible(true);
}
}

```

VoteFrame Class

```

java
import javax.swing.*;
import java.awt.*;
import java.util.HashSet;

class VoteFrame extends JFrame {
    static HashSet<String> votedUsers = new HashSet<>();

    VoteFrame(String username) {
        setTitle("Vote");
        setSize(300, 200);
        setLayout(new GridLayout(0, 1));

        if (votedUsers.contains(username)) {
            JOptionPane.showMessageDialog(this, "You have already voted!");
            dispose();
            return;
        }

        ButtonGroup group = new ButtonGroup();

        JRadioButton[] options = new
JRadioButton[PollingSystem.candidates.length];
        for (int i = 0; i < PollingSystem.candidates.length; i++) {
            options[i] = new JRadioButton(PollingSystem.candidates[i]);
            group.add(options[i]);
            add(options[i]);
        }

        JButton submitButton = new JButton("Submit Vote");
        submitButton.addActionListener(e -> {
            for (JRadioButton option : options) {
                if (option.isSelected()) {
                    PollingSystem.votes.put(option.getText(),
PollingSystem.votes.get(option.getText()) + 1);
                    votedUsers.add(username);
                    JOptionPane.showMessageDialog(this, "Vote Submitted!");
                    dispose();
                    return;
                }
            }
        });
    }
}

```