

KK14203 Object Oriented Programming

Section 1

Project 2

**Project Title : Archery Club
Registration**

Lecturer:

Dr. Mohd Shamrie Sainin

Name	Matric No.
Mohd Ikmal Bin Indan	BI19160360

Object Oriented Concept Implementation

1. Object and Class

- a.** There are seven Class in the project which are Global Class, Login Class, MemberPanel Class, AdminPanel Class and the main Class which is ArcheryClubRegistrationApp Class.
- b.** These Class interact with one another to run the code.

2. Encapsulation

- a.** There are five Class in the project which implement Encapsulation which are Login Class, LoginPanel Class, RegisterPanel Class, MemberPanel Class and AdminPanel Class.
- b.** The methods and the data in these Class are declare as private and hidden from other Class. It can be manipulated and accessed only through any member function of own class in which they are declared.

3. Inheritance

- a.** There are four Class which inherit from a parent Class which are LoginPanel Class, RegisterPanel Class, MemberPanel Class and AdminPanel Class.
- b.** These Class inherit from a parent Class JPanel by using keyword extends in order to create a desired GUI for this project.

4. Interface

- a.** The button in the project implement ActionListener interface which call the method public void actionPerformed(ActionEvent evt). The ActionEvent parameter is an Event object that represents an event (a button click). It contains information about the event.

5. Method

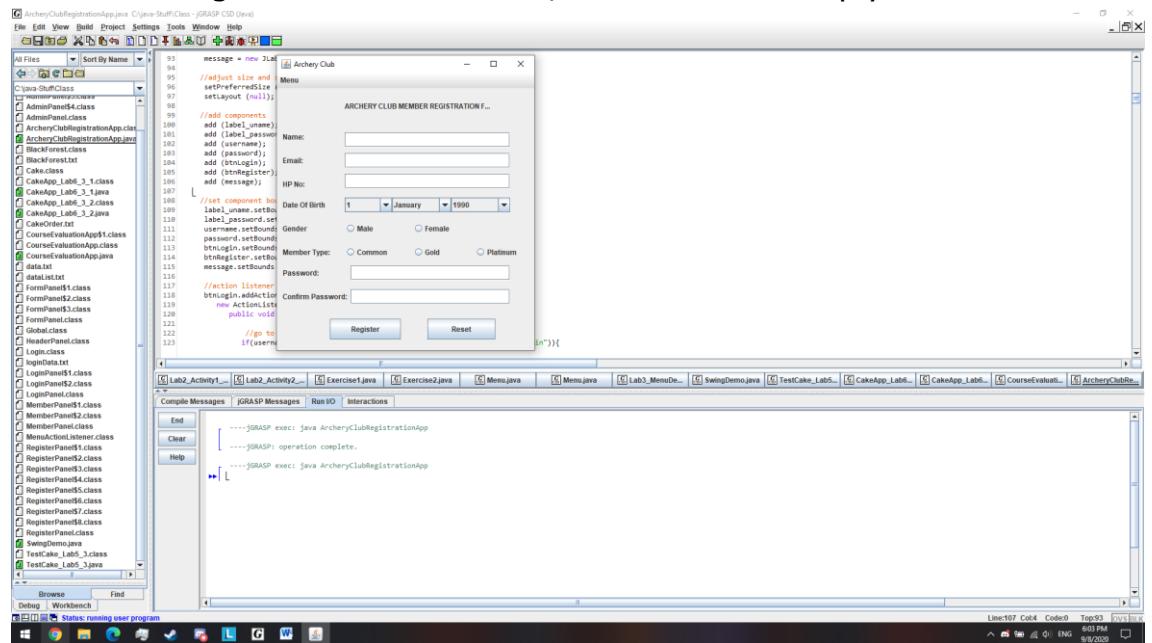
Login Class
Method: checkLogin(); doLogin(String, String);

RegisterPanel Class
Method: writeInput(); writeLoginData(); isNullOrEmpty(String);

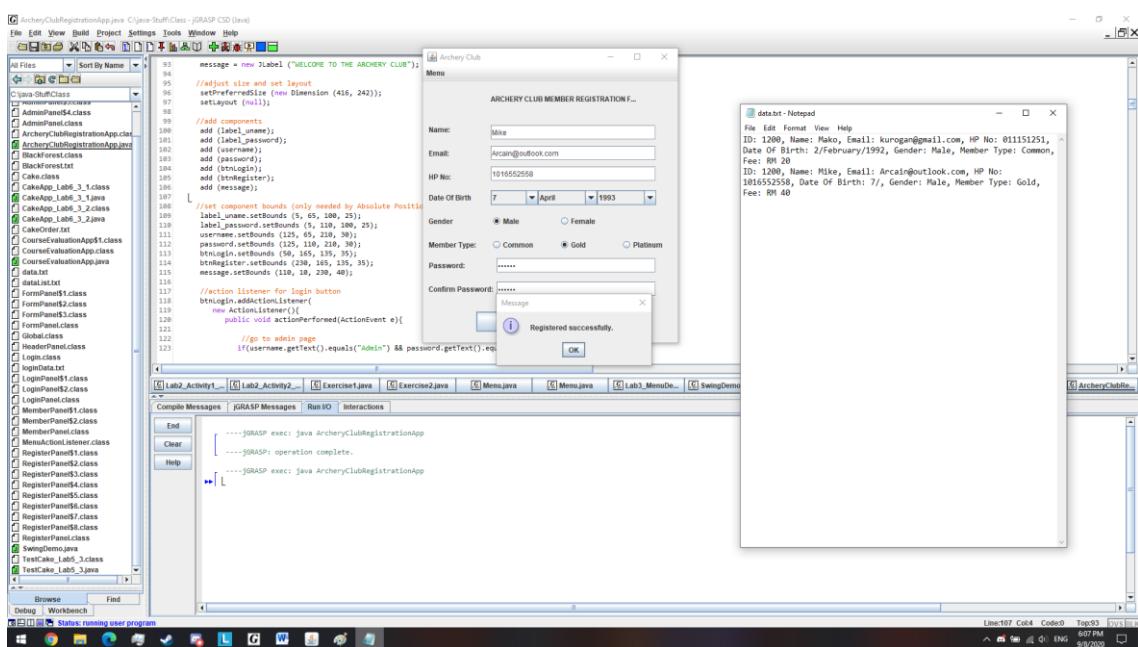
AdminPanel Class
Method: removeUserRecord (String, String); removeUserLogin (String, String);

Read and Write Implementation

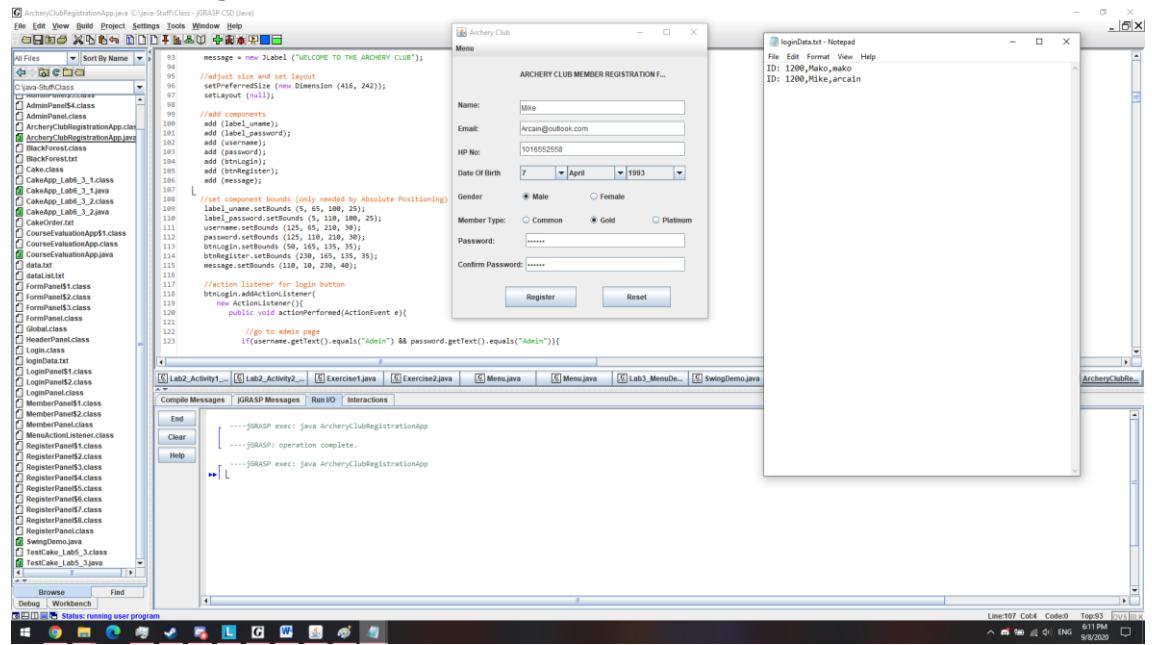
1. Data are stored in a text file named “data.txt” and “loginData.txt”
2. Before user registers their information, the text file are empty



3. Once registration is successful, user information is written into text named “data.txt” to be stored.

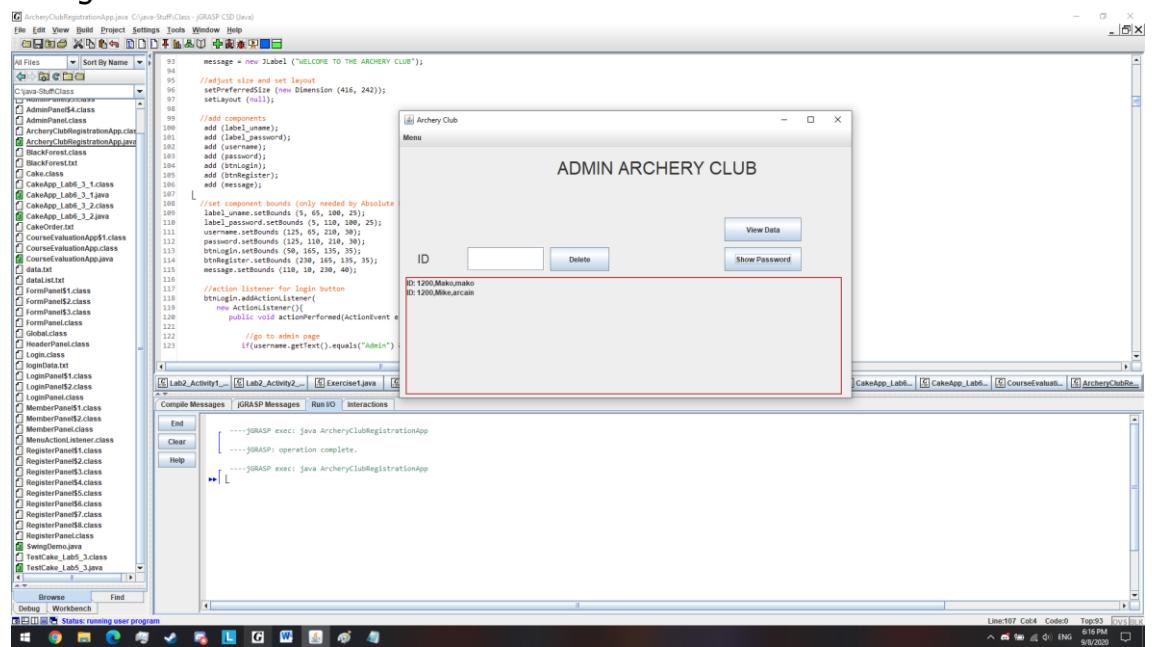


4. User login information is written into text file named "loginData.txt" to be stored once registration is successful.



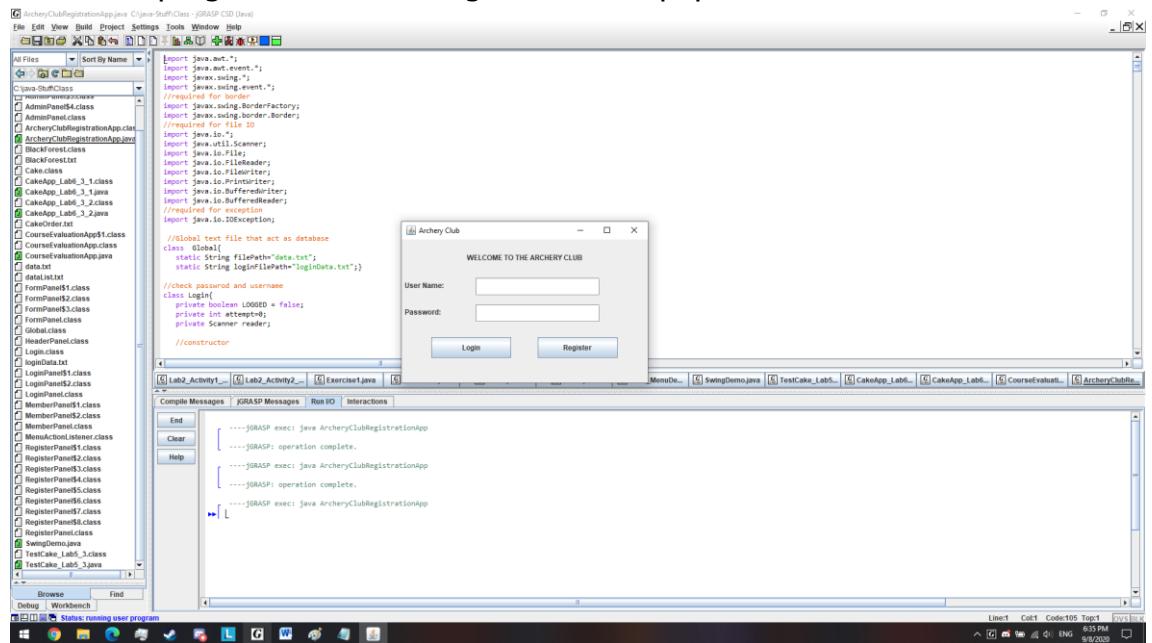
5. "data.txt" is read in Member Panel and Admin Panel to view the information of the registered member.

6. "loginData.txt" is read only in Admin Panel to view the login information of registered member.

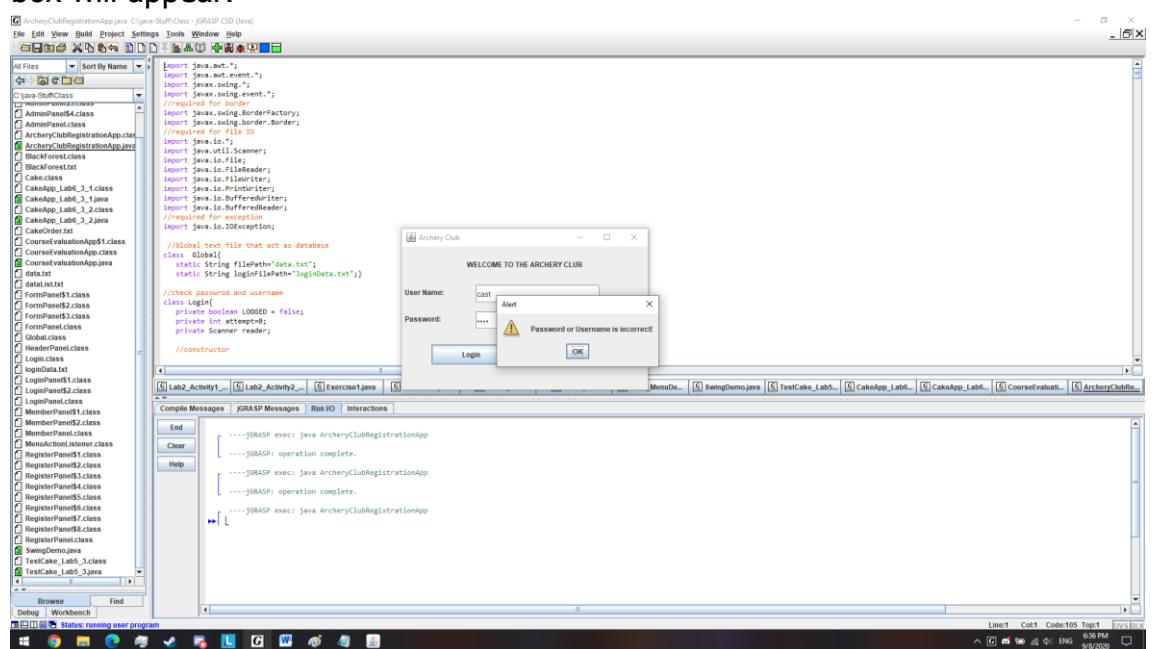


User's Manual:

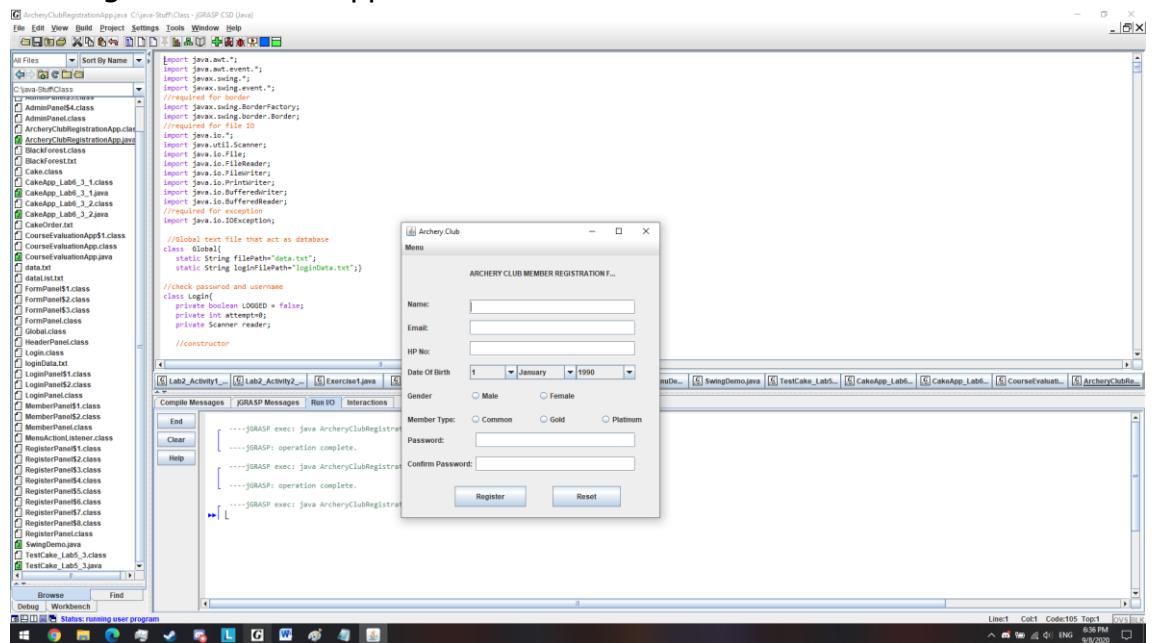
1. When the program is run the Login Panel will pop out.



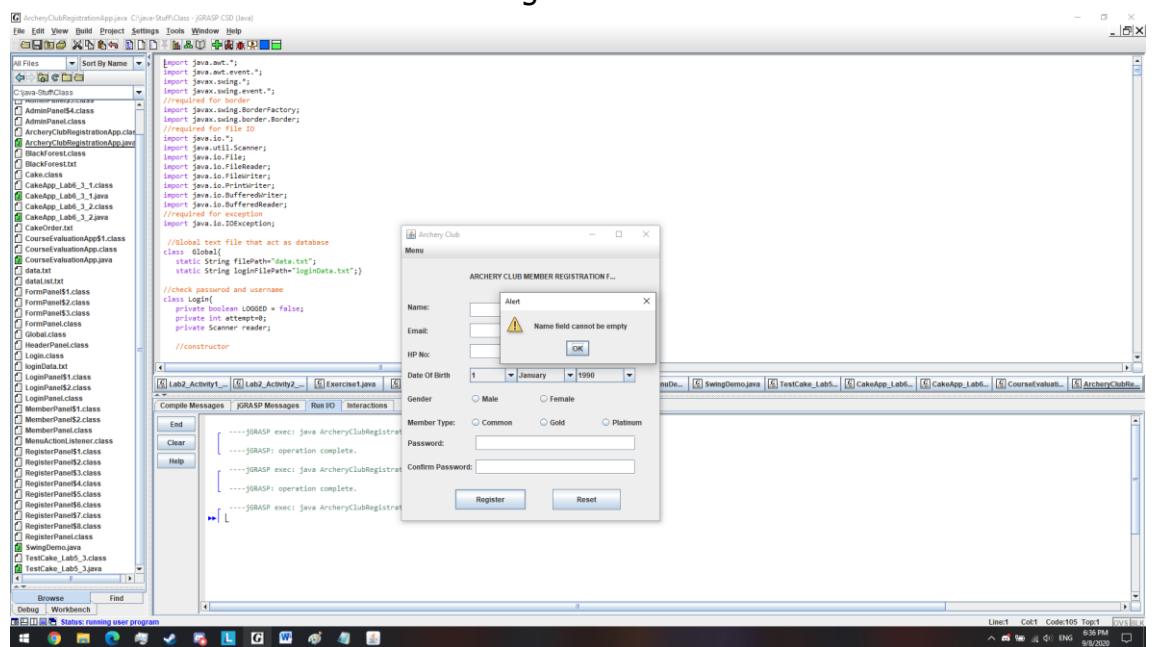
2. If the username & password entered does not exist nor match, a dialog box will appear.



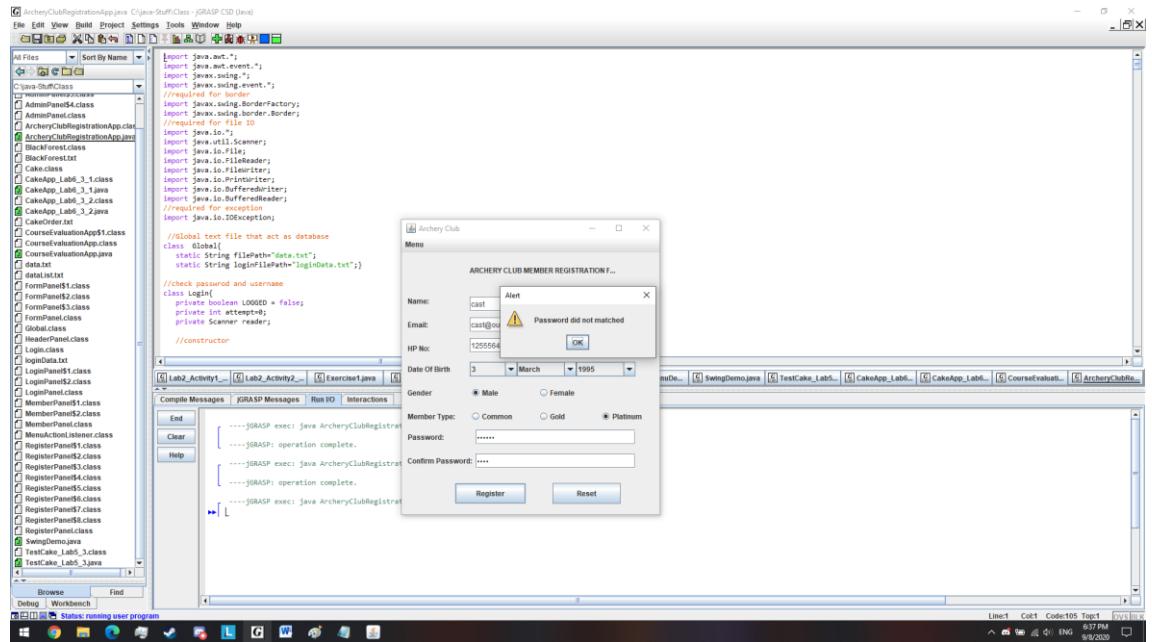
3. In order to register as a new member, click on the Register button, and the Register Panel will appear.



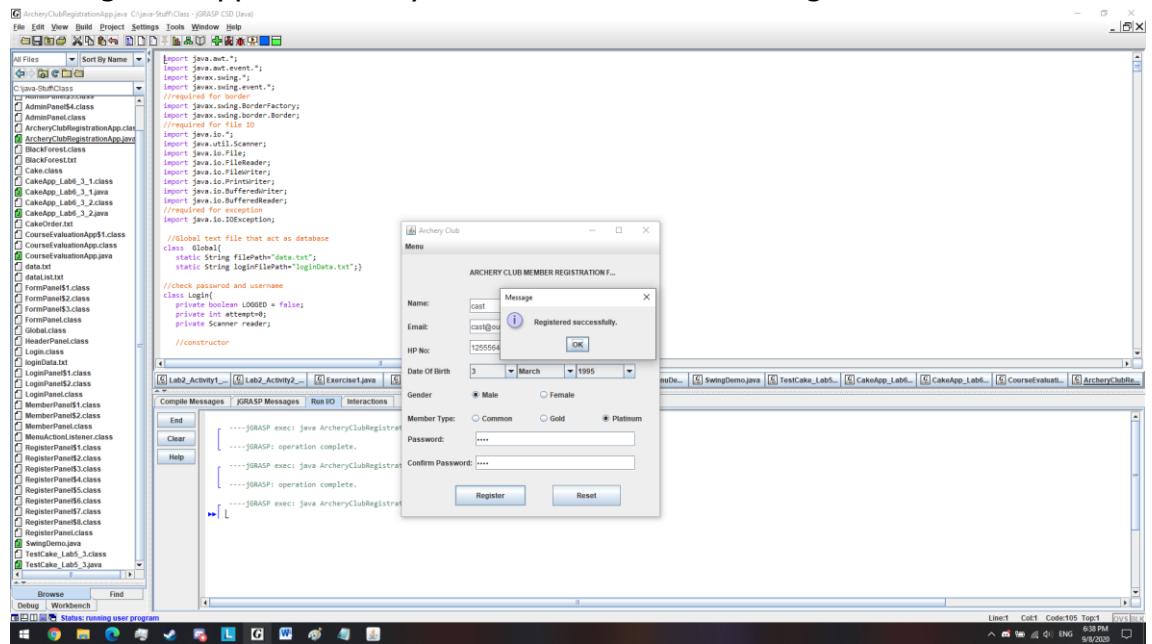
4. If the user leaves a blank field, a message box will appear to notify the user that there is a blank and the registration will be unsuccessful.



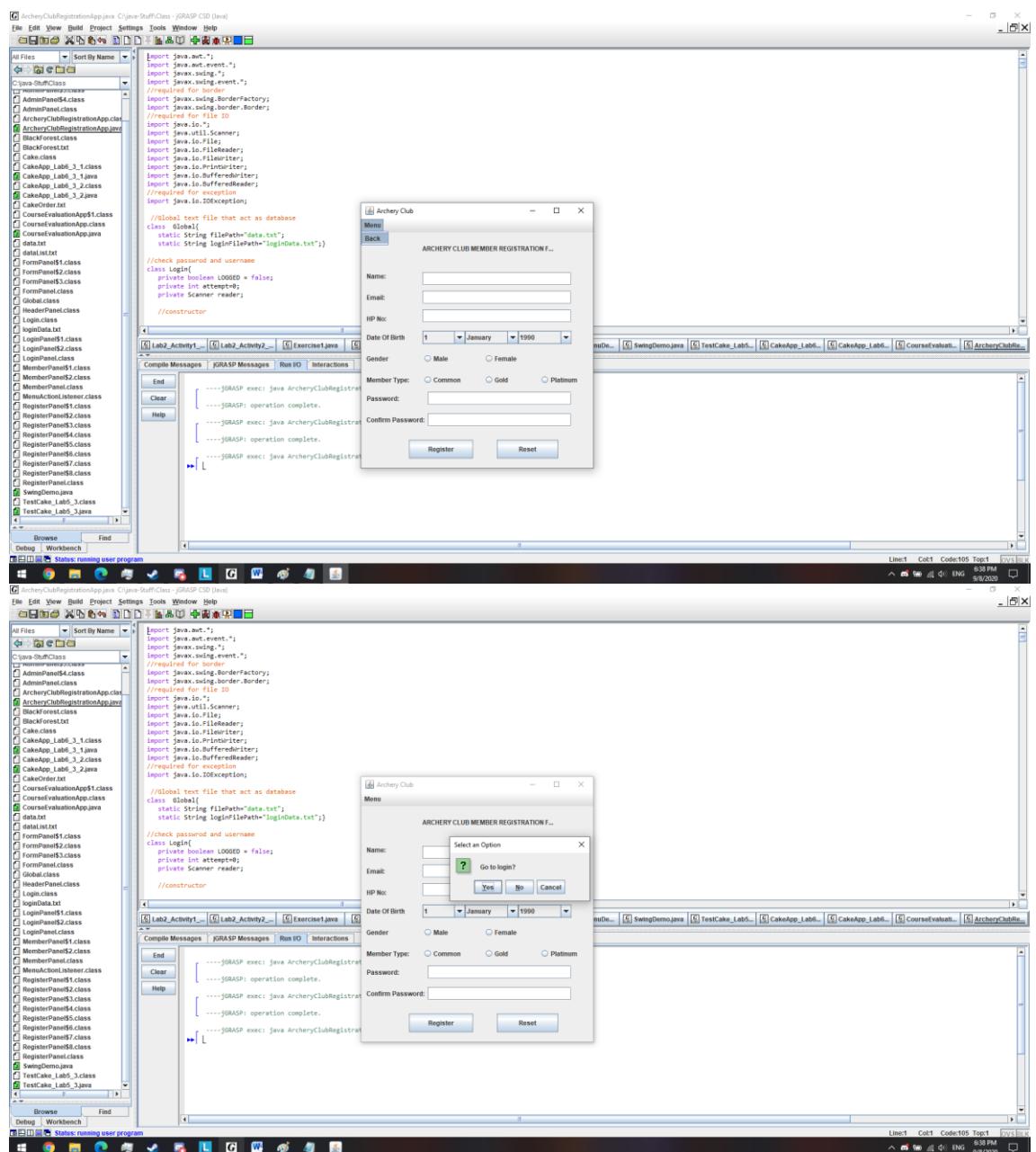
- If the password and confirmation password entered do not matched, an error message will appear.



- The register button will register the users information if the all blanks has been filled and the password and its confirmation matches. A message will appear to notify the user of a successful registration.

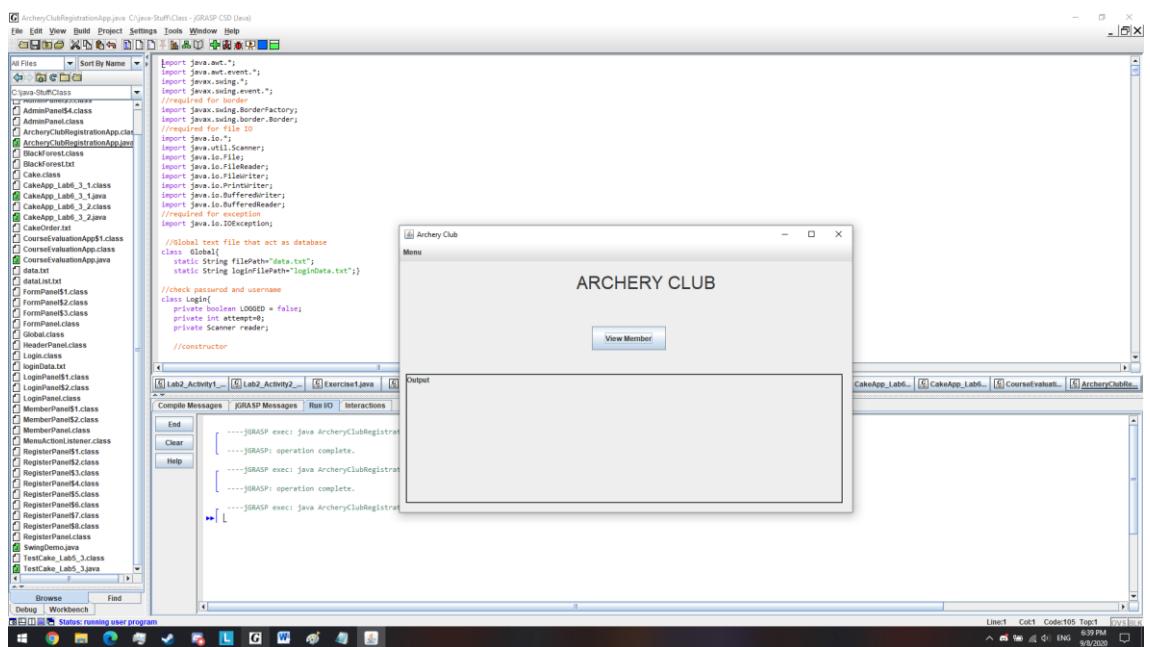


- The Reset button will clear all information type by user.
- Back in the Menu bar will bring the user back to the login panel. When selected, a confirmation message will appear. If user selects Yes, it will bring the user back to login panel.

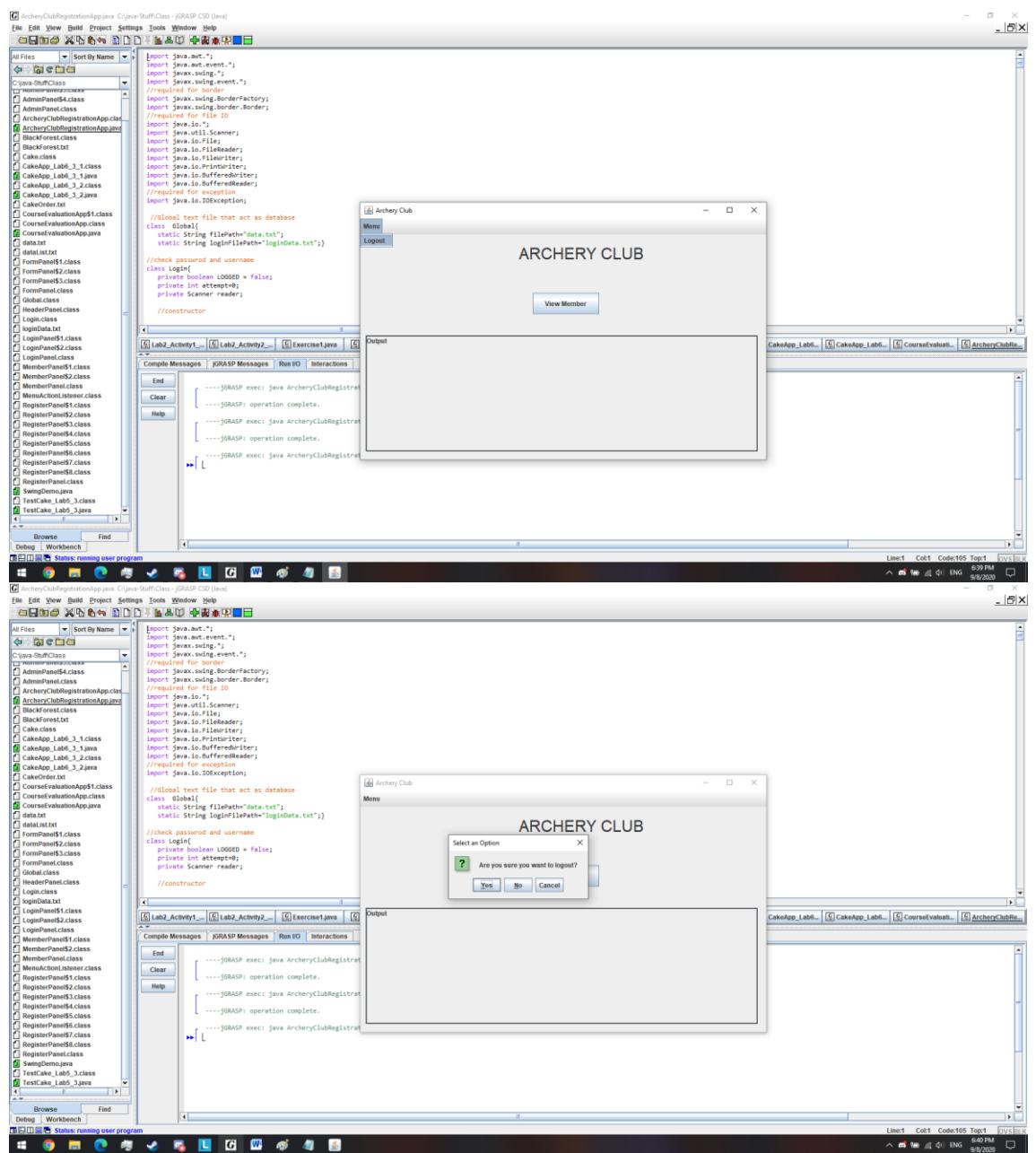


9. Once registered, user now can login by entering their name in the username field and password on the password field.
10. If login is successful, the Member Panel will appear which user can view their information and other registered member by clicking the View

Member button.

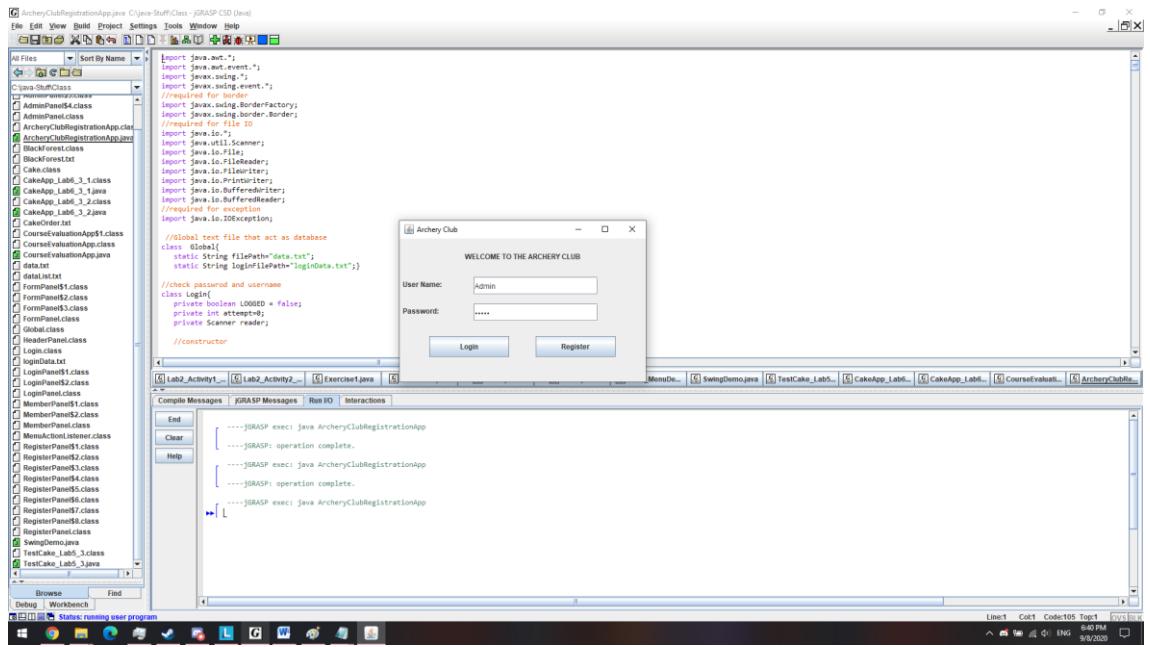


11. Logout on the Menu bar will bring back user to the login panel. A message box will appear asking for confirmation when user selects Logout. If user selects on Yes, user will be brought back to the login panel.

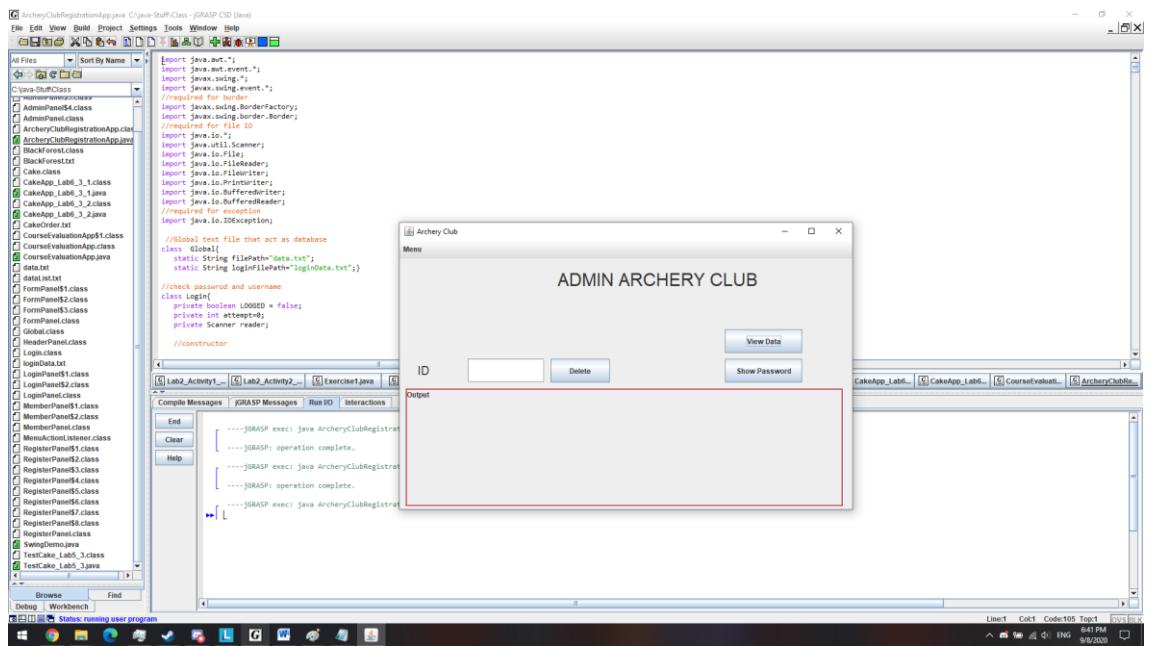


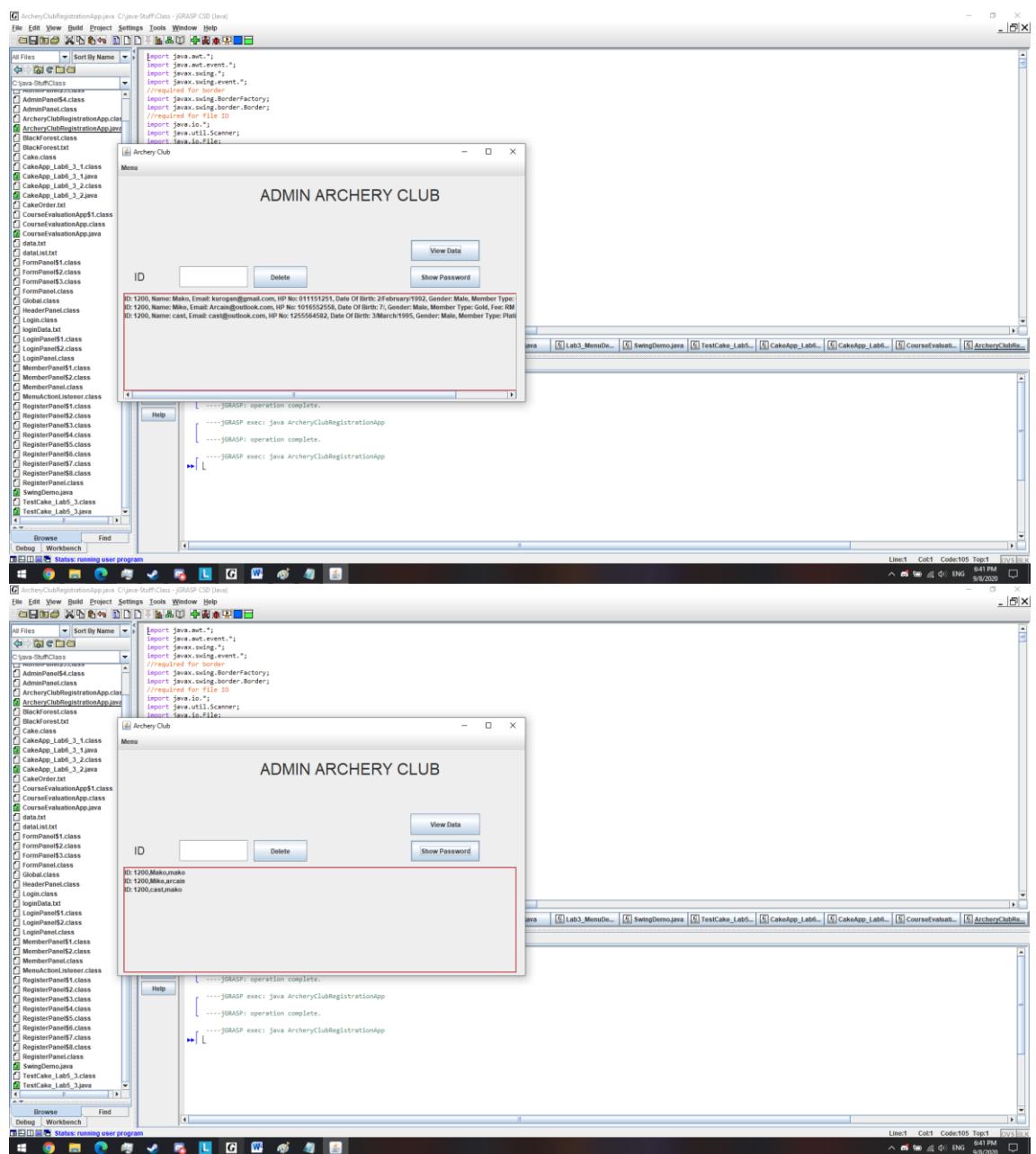
12. More members can be registered.

13. To access the admin panel, enter “Admin” as username and password.

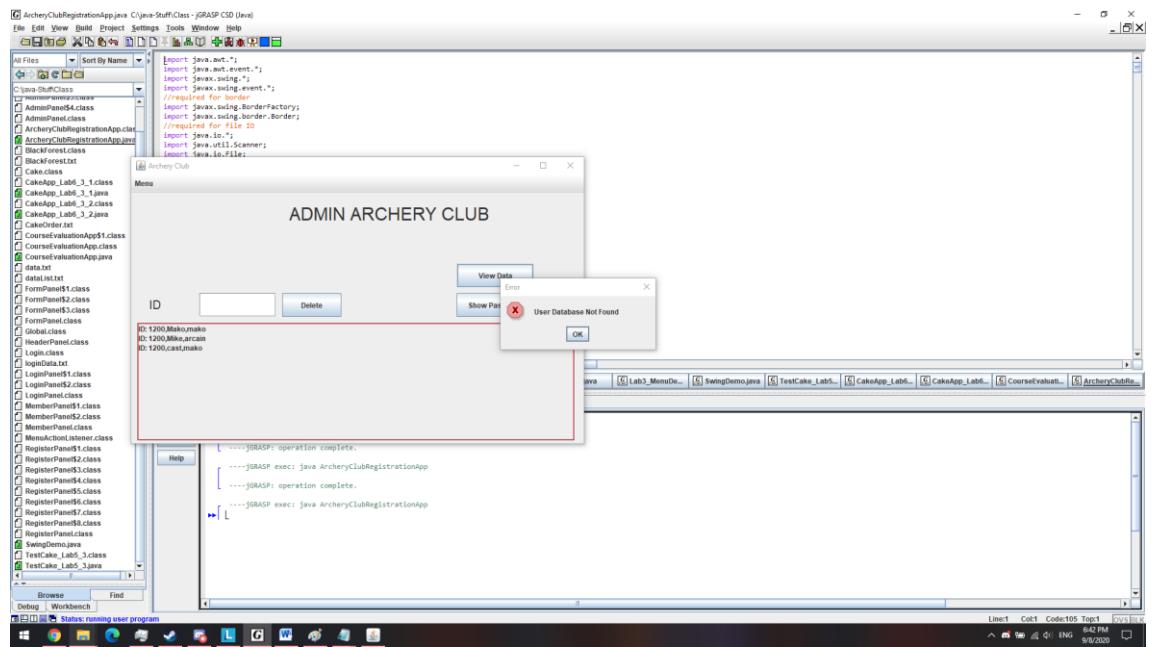


14. Admin Panel will appear where the user can view the registered member or their login information by selecting on View Data and Show Password button.

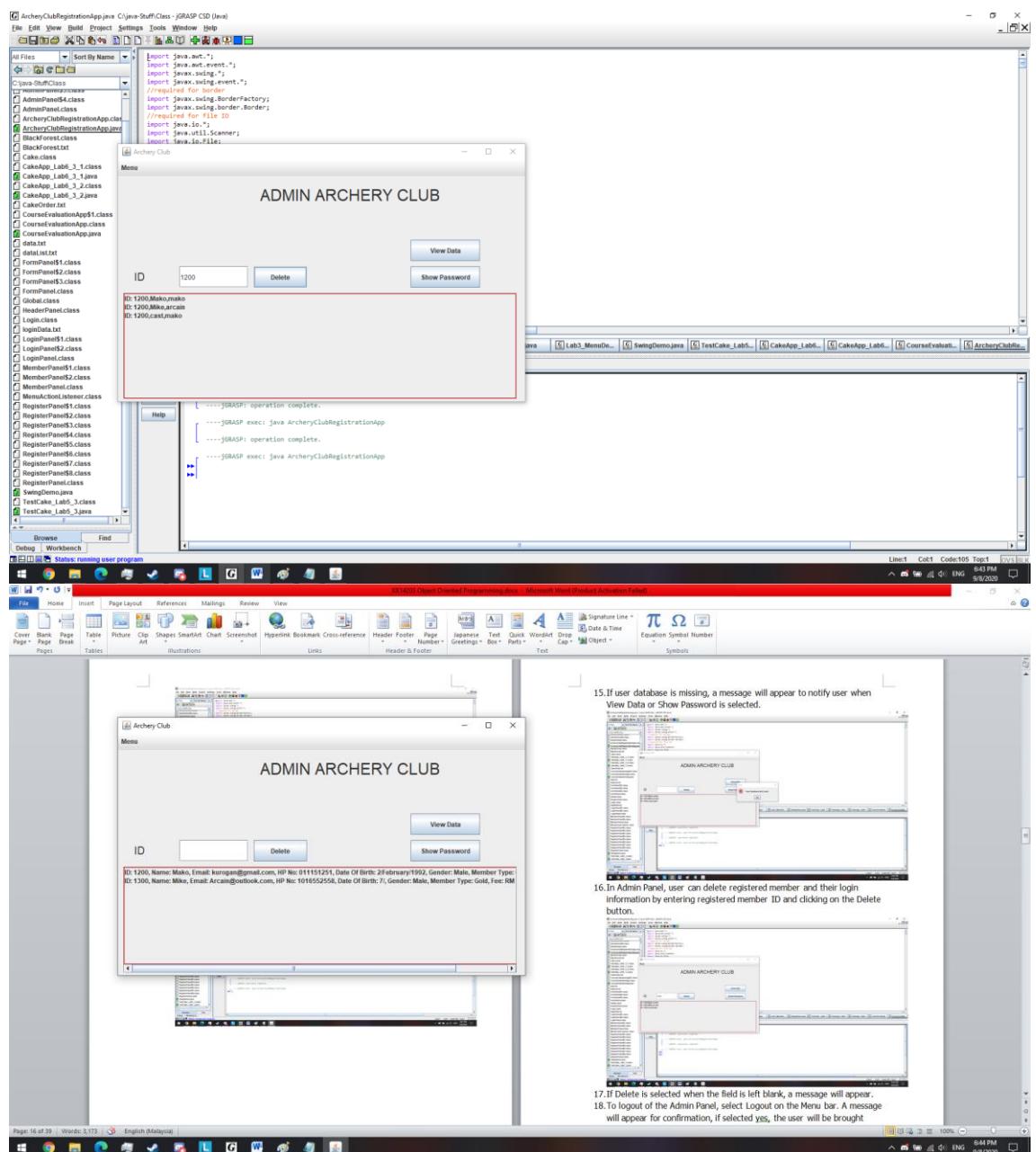




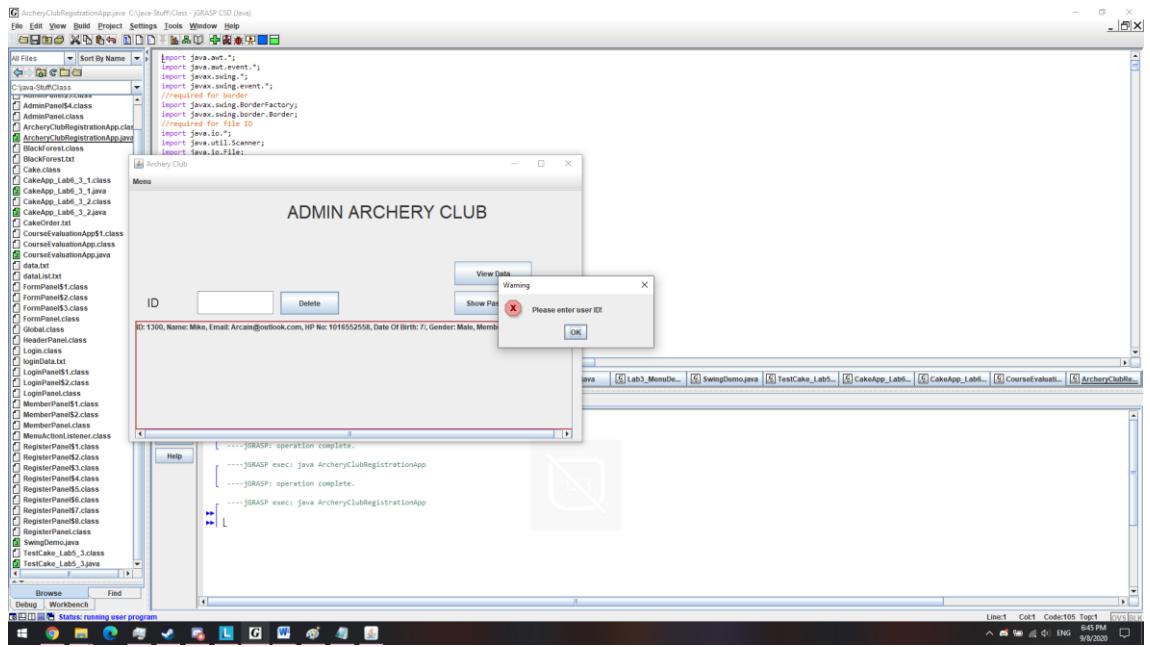
15. If user database is missing, a message will appear to notify user when View Data or Show Password is selected.



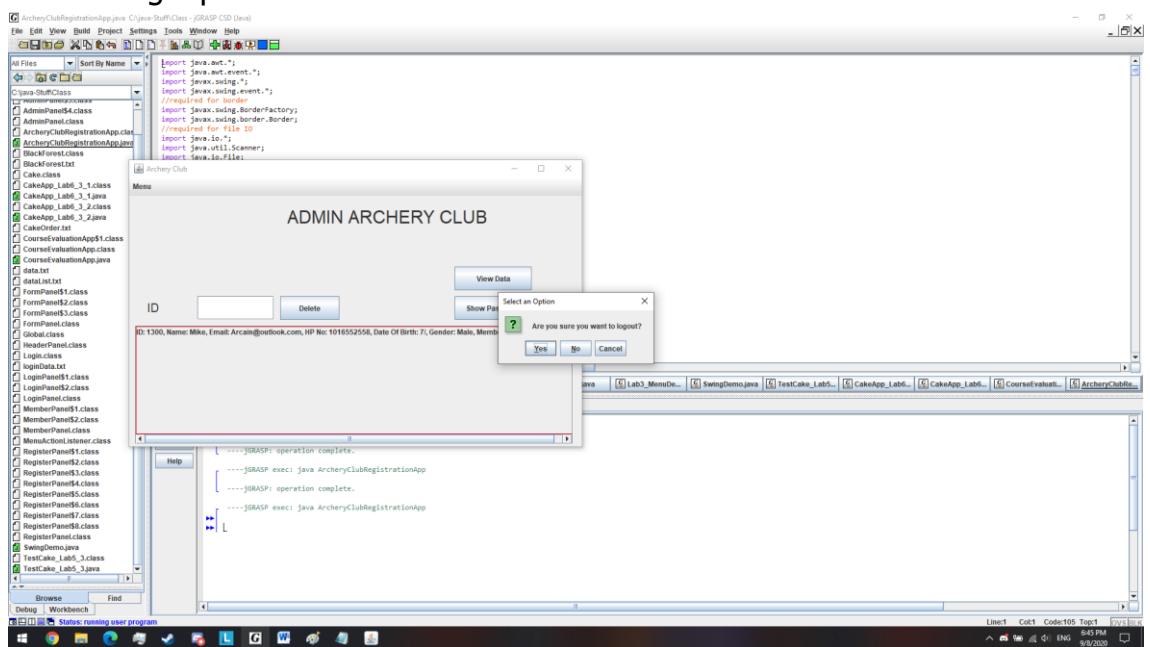
16. In Admin Panel, user can delete registered member and their login information by entering registered member ID and clicking on the Delete button.



17. If Delete is selected when the field is left blank, a message will appear.



18. To logout of the Admin Panel, select Logout on the Menu bar. A message will appear for confirmation, if selected yes, the user will be brought back to login panel.



19. The deleted member will no longer be able to login to the Member Panel.

Source Code:

```
import java.awt.*;
import java.awt.event.*;
import javax.swing.*;
import javax.swing.event.*;
//required for border
import javax.swing.BorderFactory;
import javax.swing.border.Border;
//required for file IO
import java.io.*;
import java.util.Scanner;
import java.io.File;
import java.io.FileReader;
import java.io.FileWriter;
import java.io.PrintWriter;
import java.io.BufferedWriter;
import java.io.BufferedReader;
//required for exception
import java.io.IOException;

//Global text file that act as database
class Global{
    static String filePath="data.txt";
    static String loginFilePath="logData.txt";}

//check password and username
class Login{
    private boolean LOGGED = false;
    private int attempt=0;
    private Scanner reader;

    //constructor
    public Login(){
    }

    public boolean checkLogin(){
        if(LOGGED){
            return true;
        }
        return false;
    }
}
```

```

public boolean doLogin(String username, String
password) {
    boolean name_found = false;
    boolean pass_found = false;
    boolean found = false;
    String ID = "";
    String tempusername = "";
    String temppassword = "";
    File file = new File(Global.loginFilePath);
    try{
        Scanner reader = new Scanner(file);
        reader.useDelimiter("[,\n]");

        while(reader.hasNext() && !found) {
            ID = reader.next();
            tempusername = reader.next();
            temppassword = reader.next();
            if(tempusername.trim().equals(username.trim())
&& temppassword.trim().equals(password.trim())) {
                found = true;
            }
        }
    }
    catch (Exception e) {
        JOptionPane.showMessageDialog(null,"User Database
Not Found", "Error", JOptionPane.ERROR_MESSAGE);
        return found;
    }
}

//Login panel to login
class LoginPanel extends JPanel {
    private JLabel label_uname;
    private JLabel label_password;
    private JTextField username;
    private JPasswordField password;
    private JButton btnLogin;
    private JButton btnRegister;
    private JLabel message;
    private JFrame frame;
    private JFrame dialog;
}

```

```
private Scanner reader;
private Login log;

public LoginPanel(JFrame frame, Login login) {
    //construct components
    this.frame = frame;
    this.log = login;
    label_uname = new JLabel ("User Name:");
    label_password = new JLabel ("Password:");
    username = new JTextField (5);
    password = new JPasswordField (5);
    btnLogin = new JButton ("Login");
    btnRegister = new JButton ("Register");
    message = new JLabel ("WELCOME TO THE ARCHERY CLUB");

    //adjust size and set layout
    setPreferredSize (new Dimension (416, 242));
    setLayout (null);

    //add components
    add (label_uname);
    add (label_password);
    add (username);
    add (password);
    add (btnLogin);
    add (btnRegister);
    add (message);

    //set component bounds (only needed by Absolute
    Positioning)
    label_uname.setBounds (5, 65, 100, 25);
    label_password.setBounds (5, 110, 100, 25);
    username.setBounds (125, 65, 210, 30);
    password.setBounds (125, 110, 210, 30);
    btnLogin.setBounds (50, 165, 135, 35);
    btnRegister.setBounds (230, 165, 135, 35);
    message.setBounds (110, 10, 230, 40);

    //action listener for login button
    btnLogin.addActionListener(
        new ActionListener(){
            public void actionPerformed(ActionEvent e) {
```

```

        //go to admin page
        if(username.getText().equals("Admin") &&
password.getText().equals("Admin")){
            frame.getContentPane().removeAll();
            frame.getContentPane().add (new
AdminPanel());
            frame.pack();
            frame.setVisible (true);

        }
        else{
            if(!username.getText().equals(""))
&& !password.getText ().equals ("")){
                boolean status=false;
                status =
log.doLogin(username.getText(), password.getText());
                if(status){
                    //call method
                    frame.getContentPane().removeAll();
                    frame.getContentPane().add (new
MemberPanel());
                    frame.pack();
                    frame.setVisible (true);
                }
                if(!status){
                    dialog = new JFrame();
                    JOptionPane.showMessageDialog(dialog,"Password or Username
is incorrect!","Alert",JOptionPane.WARNING_MESSAGE);
                }
            }
        }
    });

//action listener for register button
btnRegister.addActionListener(
    new ActionListener(){
        public void actionPerformed(ActionEvent e) {
            frame.getContentPane().removeAll();
            frame.getContentPane().add (new

```

```
RegisterPanel());
        frame.pack();
        frame.setVisible (true); } });
    }
}

//Register Form panel to register new user
class RegisterPanel extends JPanel implements
ActionListener{
    private JFrame message;
    private JMenuBar mbRegister;
    private JLabel lbl_registration;
    private JLabel lbl_uname;
    private JLabel lbl_hp;
    private JLabel lbl_dob;
    private JLabel lbl_gender;
    private JLabel memberType;
    private JButton btnRegister;
    private JButton btnReset;
    private JLabel lbl_email;
    private JComboBox cb_day;
    private JComboBox cb_month;
    private JRadioButton jrb_male;
    private JRadioButton jrb_female;
    private JComboBox cb_year;
    private JTextField hp_no;
    private JTextField email;
    private JTextField name;
    private JRadioButton jrb_cmon;
    private JRadioButton jrb_gold;
    private JRadioButton jrb_platinum;
    private JLabel lbl_passwor;
    private JLabel lbl_passconfirm;
    private JPasswordField password;
    private JPasswordField confirm_pass;

    private String gender_selection="";
    private String dob_selection="";
    private String type_selection="";
    private String fee="";
    private static int userID = 1200;
```

```

public RegisterPanel() {
    //construct preComponents
    JMenu menuMenu = new JMenu ("Menu");
    JMenuItem backItem = new JMenuItem ("Back");
    menuMenu.add (backItem);
    String[] cb_dayItems = {"1", "2", "3", "4", "5", "6",
    "7", "8", "9", "10", "11", "12", "13", "14", "15", "16",
    "17", "18", "19", "20", "21", "22", "23", "24", "25", "26",
    "27", "28", "29", "30", "31"};
    String[] cb_monthItems = {"January", "February",
    "March", "April", "May", "June", "July", "August",
    "September", "October", "November", "December"};
    String[] cb_yearItems = {"1990", "1991", "1992",
    "1993", "1994", "1995", "1996", "1997", "1998", "1999",
    "2000", "2001", "2002", "2003", "2004", "2005", "2006",
    "2007", "2008", "2009", "2010", "2011", "2012", "2013",
    "2014", "2015", "2016", "2017", "2018", "2019", "2020"};

    //construct components
    mbRegister = new JMenuBar();
    mbRegister.add (menuMenu);
    lbl_registration = new JLabel ("ARCHERY CLUB MEMBER
REGISTRATION FORM");
    lbl_uname = new JLabel ("Name:");
    lbl_hp = new JLabel ("HP No:");
    lbl_dob = new JLabel ("Date Of Birth");
    lbl_gender = new JLabel ("Gender");
    memberType = new JLabel ("Member Type:");
    btnRegister = new JButton ("Register");
    btnReset = new JButton ("Reset");
    lbl_email = new JLabel ("Email:");

    cb_day = new JComboBox (cb_dayItems);
    cb_day.addActionListener(
        new ActionListener(){
            public void actionPerformed(ActionEvent ae) {
                //get selected item
                dob_selection = (String)
cb_day.getSelectedItem() + "/";
            }
        });
}

```

```

cb_month = new JComboBox (cb_monthItems);
cb_month.addActionListener(
    new ActionListener() {
        public void actionPerformed(ActionEvent ae) {
            //get selected item
            dob_selection += (String)
cb_month.getSelectedItem() + "/";
        }
    });
}

cb_year = new JComboBox (cb_yearItems);
cb_year.addActionListener(
    new ActionListener() {
        public void actionPerformed(ActionEvent ae) {
            //get selected item
            dob_selection += (String)
cb_year.getSelectedItem();
        }
    });
}

jrb_male = new JRadioButton ("Male");
jrb_male.addActionListener(
    new ActionListener() {
        public void actionPerformed(ActionEvent ae) {
            gender_selection = (String)
jrb_male.getText();
        }
    });
jrb_female = new JRadioButton ("Female");
jrb_female.addActionListener(
    new ActionListener() {
        public void actionPerformed(ActionEvent ae) {
            gender_selection = (String)
jrb_female.getText();
        }
    });
}

hp_no = new JTextField (5);
email = new JTextField (5);
name = new JTextField (5);

jrb_cmon = new JRadioButton ("Common");
jrb_cmon.addActionListener(this);
jrb_gold = new JRadioButton ("Gold");

```

```
jrb_gold.addActionListener(this);
jrb_platinum = new JRadioButton ("Platinum");
jrb_platinum.addActionListener(this);

lbl_passwor = new JLabel ("Password:");
lbl_passconfirm = new JLabel ("Confirm Password:");
password = new JPasswordField (5);
confirm_pass = new JPasswordField (5);

//adjust size and set layout
setPreferredSize (new Dimension (436, 468));
setLayout (null);

//add components
add (mbRegister);
add (lbl_registration);
add (lbl_uname);
add (lbl_hp);
add (lbl_dob);
add (lbl_gender);
add (memberType);
add (btnRegister);
add (btnReset);
add (lbl_email);
add (cb_day);
add (cb_month);
add (jrb_male);
add (jrb_female);
add (cb_year);
add (hp_no);
add (email);
add (name);
add (jrb_cmmmon);
add (jrb_gold);
add (jrb_platinum);
add (lbl_passwor);
add (lbl_passconfirm);
add (password);
add (confirm_pass);

//define button group
```

```

        ButtonGroup bg_gender = new ButtonGroup();
        bg_gender.add(jrb_male);
        bg_gender.add(jrb_female);

        ButtonGroup bg_membertype = new ButtonGroup();
        bg_membertype.add(jrb_cmon);
        bg_membertype.add(jrb_gold);
        bg_membertype.add(jrb_platinum);

        //handle button register action listener
        //and write to file
        btnRegister.addActionListener(
            new ActionListener(){
                public void actionPerformed(ActionEvent e) {
                    //input validation to be registered
                    message = new JFrame();

                    if(name.getText().equals("")){JOptionPane.showMessageDialog(message,"Name field cannot be
empty","Alert",JOptionPane.WARNING_MESSAGE); }

                    else
                    if(email.getText().equals("")){JOptionPane.showMessageDialog(message,"Email field cannot be
empty","Alert",JOptionPane.WARNING_MESSAGE); }

                    else
                    if(hp_no.getText().equals("")){JOptionPane.showMessageDialog(message,"HP No field cannot be
empty","Alert",JOptionPane.WARNING_MESSAGE); }

                    else
                    if(isNullOrEmpty(dob_selection)) {JOptionPane.showMessageDialog(message,"Date of birth is not
selected","Alert",JOptionPane.WARNING_MESSAGE); }

                    else
                    if(isNullOrEmpty(gender_selection)) {JOptionPane.showMessageDialog(message,"Gender is not
selected","Alert",JOptionPane.WARNING_MESSAGE); }

                    else

```

```

if(isNullOrEmpty(type_selection)) { JOptionPane.showMessageDialog(message,"Member type is not selected","Alert",JOptionPane.WARNING_MESSAGE); }

        else if(password.getText().equals("") || confirm_pass.getText().equals(""))

{ JOptionPane.showMessageDialog(message,"Password field cannot be empty","Alert",JOptionPane.WARNING_MESSAGE); }

        else
if(!password.getText().equals(confirm_pass.getText()))

{ JOptionPane.showMessageDialog(message,"Password did not matched","Alert",JOptionPane.WARNING_MESSAGE); }

        else{
            if(type_selection.equals("Common")){
                fee = "RM 20";
            } else if(type_selection.equals("Gold")){
                fee = "RM 40";
            } else
if(type_selection.equals("Platinum")){
                fee = "RM 60";
            }
//call method
writeInput();
writeloginData();
message = new JFrame();

JOptionPane.showMessageDialog(message,"Registered successfully.");//Show message if data is saved to text
userID++;
}

}

}

);

//handle button clear action listener
btnReset.addActionListener(
    new ActionListener(){
        public void actionPerformed(ActionEvent e) {
            name.setText("");
            hp_no.setText("");
        }
    }
);

```

```

        email.setText("");
        cb_day.setSelectedIndex(0);
        cb_month.setSelectedIndex(0);
        cb_year.setSelectedIndex(0);
        bg_gender.clearSelection();
        bg_membertype.clearSelection();
        password.setText("");
        confirm_pass.setText("");
    }
}

//handle back menu bar action listener
backItem.addActionListener(
    new ActionListener(){
        public void actionPerformed(ActionEvent e) {
            message = new JFrame();
            int
a= JOptionPane.showConfirmDialog(message, "Go to login?");
            if(a==JOptionPane.YES_OPTION) {

ArcheryClubRegistrationApp.frame.getContentPane().removeAll
();
ArcheryClubRegistrationApp.frame.getContentPane().add (new
LoginPanel(ArcheryClubRegistrationApp.frame,
ArcheryClubRegistrationApp.log));
            ArcheryClubRegistrationApp.frame.pack();

ArcheryClubRegistrationApp.frame.setVisible (true);}});

//set component bounds (only needed by Absolute
Positioning)
mbRegister.setBounds (0, 0, 625, 25);
lbl_registration.setBounds (115, 40, 250, 30);
lbl_uname.setBounds (10, 95, 100, 25);
lbl_hp.setBounds (10, 175, 100, 25);
lbl_dob.setBounds (10, 210, 100, 25);
lbl_gender.setBounds (10, 250, 100, 25);
memberType.setBounds (10, 290, 100, 25);
btnRegister.setBounds (90, 415, 120, 35);
btnReset.setBounds (255, 415, 115, 35);
lbl_email.setBounds (10, 135, 100, 25);

```

```

        cb_day.setBounds (115, 210, 80, 25);
        cb_month.setBounds (195, 210, 100, 25);
        jrb_male.setBounds (115, 250, 100, 25);
        jrb_female.setBounds (230, 250, 100, 25);
        cb_year.setBounds (295, 210, 100, 25);
        hp_no.setBounds (115, 170, 280, 25);
        email.setBounds (115, 135, 280, 25);
        name.setBounds (115, 100, 280, 25);
        jrb_cmon.setBounds (115, 290, 100, 25);
        jrb_gold.setBounds (230, 290, 100, 25);
        jrb_platinum.setBounds (335, 290, 100, 25);
        lbl_passwor.setBounds (10, 325, 100, 25);
        lbl_passconfirm.setBounds (10, 365, 110, 25);
        password.setBounds (125, 325, 270, 25);
        confirm_pass.setBounds (125, 365, 270, 25);
    }

    //write to file
    public void writeInput(){
        File file = new File(Global.filePath);
        FileWriter fr = null;
        BufferedWriter br = null;
        PrintWriter pr = null;

        String input = "ID: " +String.valueOf(userID)+", Name: "+name.getText()+", Email: "+email.getText()+" , HP No: "+hp_no.getText()+", Date Of Birth: "+ dob_selection + ", Gender: "+gender_selection+", Member Type: "+type_selection+", Fee: "+fee;

        //exception implementation
        try {
            // to append to file, you need to initialize
FileWriter using below constructor
            fr = new FileWriter(file, true);
            br = new BufferedWriter(fr);
            pr = new PrintWriter(br);
            pr.println(input);
        } catch (IOException e) {
            System.out.println(e.toString());
        } finally {
            try {

```

```

        pr.close();
        br.close();
        fr.close();
    } catch (IOException e) {
        JOptionPane.showMessageDialog(null,"IOException
file is missing", "Error", JOptionPane.ERROR_MESSAGE);
    }
}

//write to file for login data
public void writeloginData(){
    File file = new File(Global.loginFilePath);
    FileWriter fr = null;
    BufferedWriter br = null;
    PrintWriter pr = null;

    String input = "ID: "
+String.valueOf(userID)+", "+name.getText()+", "+password.get
Text();

    //exception implementation
    try {
        // to append to file, you need to initialize
FileWriter using below constructor
        fr = new FileWriter(file, true);
        br = new BufferedWriter(fr);
        pr = new PrintWriter(br);
        pr.println(input);
    } catch (IOException e) {
        System.out.println(e.toString());
    } finally {
        try {
            pr.close();
            br.close();
            fr.close();
        } catch (IOException e) {
            JOptionPane.showMessageDialog(null,"IOException
file is missing", "Error", JOptionPane.ERROR_MESSAGE);
        }
    }
}

```

```

//get member type selection
public void actionPerformed(ActionEvent ae) {
    type_selection = ae.getActionCommand();
}

//check if String is Empty
public boolean isNullOrEmpty(String str) {
    if(str != null && !str.isEmpty())
        return false;
    return true;
}

}

//Main Panel show after login to view member
class MemberPanel extends JPanel {
    private JLabel lbl_output;
    private JMenuBar golf_mb;
    private JLabel lbl_golf;
    private JButton btnView;
    private JFrame message;
    private JScrollPane sp;
    private Border border =
BorderFactory.createLineBorder(Color.BLACK, 1);

    public MemberPanel() {
        //construct preComponents
        JMenu menu = new JMenu ("Menu");
        JMenuItem logoutItem = new JMenuItem ("Logout");
        menu.add (logoutItem);

        //construct components
        lbl_output = new JLabel ("Output");
        lbl_output.setBorder(border);
        lbl_output.setVerticalAlignment(JLabel.TOP);
        golf_mb = new JMenuBar();
        golf_mb.add (menu);
        lbl_golf = new JLabel ("          ARCHERY CLUB");
        lbl_golf.setFont(new Font("Arial", Font.PLAIN, 30));
        btnView = new JButton ("View Member");
        sp = new JScrollPane(lbl_output);
    }
}

```

```

//buttonView action listener to view registered
member
btnView.addActionListener(
    new ActionListener(){
        public void actionPerformed(ActionEvent e) {
            File file = new File(Global.filePath);
            try{
                BufferedReader reader = new
BufferedReader(new FileReader(file));
                String line = reader.readLine();
                String output=<html>;
                while (line != null) {
                    output += line + "<br>";
                    // read next line
                    line = reader.readLine();
                }
                output += "<br>";
                lbl_output.setText(output);
                reader.close();
            }
            catch (IOException ae) {
                JOptionPane.showMessageDialog(null,"User
Database Not Found", "Error",JOptionPane.ERROR_MESSAGE);
            }});
}

//Logout action listener to logout from mainpanel to
login panel
logoutItem.addActionListener(
    new ActionListener(){
        public void actionPerformed(ActionEvent e) {
            message = new JFrame();
            int
a=JOptionPane.showConfirmDialog(message,"Are you sure you
want to logout?");
            if(a==JOptionPane.YES_OPTION){

ArcheryClubRegistrationApp.frame.getContentPane().removeAll
();

ArcheryClubRegistrationApp.frame.getContentPane().add (new
LoginPanel(ArcheryClubRegistrationApp.frame,
ArcheryClubRegistrationApp.log));

```

```
        ArcheryClubRegistrationApp.frame.pack();

ArcheryClubRegistrationApp.frame.setVisible (true);}});

//adjust size and set layout
setPreferredSize (new Dimension (765, 454));
setLayout (null);

//add components
add (golf_mb);
add (lbl_golf);
add (btnView);
add (sp);

//set component bounds (only needed by Absolute
Positioning)
sp.setBounds (10, 220, 740, 220);
golf_mb.setBounds (0, 0, 770, 30);
lbl_golf.setBounds (195, 40, 420, 55);
btnView.setBounds (325, 140, 125, 40);
}

//Admin page to delete data
class AdminPanel extends JPanel {
private JMenuBar golf_mb;
private JLabel lbl_admin;
private JButton btn_viewData;
private JLabel lbl_output;
private JButton btn_showPass;
private JTextField id;
private JButton btn_del;
private JLabel lbl_id;
private JScrollPane jsp_admin;
private Border border =
BorderFactory.createLineBorder(Color.RED, 1);
private Frame message;

public AdminPanel() {
    //construct preComponents
    JMenu menuMenu = new JMenu ("Menu");
    JMenuItem logoutItem = new JMenuItem ("Logout");
    menuMenu.add (logoutItem);
```

```

//construct components
golf_mb = new JMenuBar();
golf_mb.add (menuMenu);
lbl_admin = new JLabel ("           ADMIN ARCHERY
CLUB");
lbl_admin.setFont(new Font("Arial", Font.PLAIN, 30));
btn_viewData = new JButton ("View Data");
lbl_output = new JLabel ("Output");
lbl_output.setBorder(border);
jsp_admin = new JScrollPane(lbl_output);
lbl_output.setVerticalAlignment(JLabel.TOP);
btn_showPass = new JButton ("Show Password");
id = new JTextField (5);
btn_del = new JButton ("Delete");
lbl_id = new JLabel ("      ID");
lbl_id.setFont(new Font("Arial", Font.PLAIN, 20));

//adjust size and set layout
setPreferredSize (new Dimension (765, 454));
setLayout (null);

//logout for admin panel
logoutItem.addActionListener(
    new ActionListener(){
        public void actionPerformed(ActionEvent e) {
            message = new JFrame();
            int
a=JOptionPane.showConfirmDialog(message,"Are you sure you
want to logout?");
            if(a== JOptionPane.YES_OPTION) {

ArcheryClubRegistrationApp.frame.getContentPane().removeAll
();

ArcheryClubRegistrationApp.frame.getContentPane().add (new
LoginPanel(ArcheryClubRegistrationApp.frame,
ArcheryClubRegistrationApp.log));
                ArcheryClubRegistrationApp.frame.pack();

ArcheryClubRegistrationApp.frame.setVisible (true);}});

```

```

        //button view data action listener to view
registered member
    btn_viewData.addActionListener(
        new ActionListener(){
            public void actionPerformed(ActionEvent e) {
                File file = new File(Global.filePath);
                try{
                    BufferedReader reader = new
BufferedReader(new FileReader(file));
                    String line = reader.readLine();
                    String output=<html>;
                    while (line != null) {
                        output += line + "<br>";
                    // read next line
                        line = reader.readLine();
                    }
                    output += "<br>";
                    lbl_output.setText(output);
                    reader.close();
                }
                catch (IOException ae) {
                    JOptionPane.showMessageDialog(null,"User
Database Not Found", "Error",JOptionPane.ERROR_MESSAGE);
                } });
            }

        //button viewPassword action listener to view
username & password of registered member
    btn_showPass.addActionListener(
        new ActionListener(){
            public void actionPerformed(ActionEvent e) {
                File file = new File(Global.loginFilePath);
                try{
                    BufferedReader reader = new
BufferedReader(new FileReader(file));
                    String line = reader.readLine();
                    String output=<html>;
                    while (line != null) {
                        output += line + "<br>";
                    // read next line
                        line = reader.readLine();
                    }
                    output += "<br>";

```

```

        lbl_output.setText(output);
        reader.close();
    }
    catch (IOException ae) {
        JOptionPane.showMessageDialog(null,"User
Database Not Found", "Error",JOptionPane.ERROR_MESSAGE);
    } });

//button delete data action listener to delete
registered member
btn_del.addActionListener(
    new ActionListener(){
        public void actionPerformed(ActionEvent e) {
            if(id.getText().equals(""))
                JOptionPane.showMessageDialog(null,"Please enter user ID!",
                "Warning",JOptionPane.ERROR_MESSAGE);

            else{
                String removeTerm = "ID: " + id.getText();
                removeUserRecord(Global.filePath,
removeTerm);
                removeUserLogin(Global.loginFilePath,
removeTerm);
            }
        } });

//add components
add (golf_mb);
add (lbl_admin);
add (btn_viewData);
add (btn_showPass);
add (id);
add (btn_del);
add (lbl_id);
add (jsp_admin);

//set component bounds (only needed by Absolute
Positioning)
golf_mb.setBounds (0, 0, 770, 30);
lbl_admin.setBounds (195, 40, 420, 55);
btn_viewData.setBounds (550, 150, 130, 40);

```

```

        btn_showPass.setBounds (550, 200, 130, 40);
        id.setBounds (115, 200, 130, 40);
        btn_del.setBounds (255, 200, 100, 40);
        lbl_id.setBounds (0, 200, 100, 40);
        jsp_admin.setBounds (10, 250, 740, 200);
    }

    //method to remove user data
    public void removeUserRecord(String filepath, String
removeTerm){
    String tempFile = "temp.txt";
    File oldFile = new File(filepath);
    File newFile = new File(tempFile);
    String id = ""; String name = ""; String email = "";
String hp = "";
    String dob = ""; String gender = ""; String membertype
= ""; String fee = "";

    try{
        FileWriter fw = new FileWriter(tempFile,true);
        BufferedWriter bw = new BufferedWriter(fw);
        PrintWriter pw = new PrintWriter(bw);
        Scanner reader = new Scanner(new File(filepath));
        reader.useDelimiter("[,\n]");

        while(reader.hasNext())
        {
            id = reader.next();
            name = reader.next();
            email = reader.next();
            hp = reader.next();
            dob = reader.next();
            gender = reader.next();
            membertype = reader.next();
            fee = reader.next();
            if(!id.trim().equals(removeTerm.trim()))
            {
                pw.println(id + ", " + name +", " +email+",
"+hp+", "+dob+", "+gender+", "+membertype+", "+fee);
            }
        }
        reader.close();
    }
}

```

```

        pw.flush();
        pw.close();
        oldFile.delete();
        File dump = new File(filepath);
        newFile.renameTo(dump);
    }catch (Exception e)
    {JOptionPane.showMessageDialog(null,"User Database Not
Found", "Error", JOptionPane.ERROR_MESSAGE);}
}

//method to remove user login data
public void removeUserLogin(String filepath, String
removeTerm){
    String tempFile = "temp2.txt";
    File oldFile = new File(filepath);
    File newFile = new File(tempFile);
    String id =""; String name =""; String password ="";

    try{
        FileWriter fw = new FileWriter(tempFile,true);
        BufferedWriter bw = new BufferedWriter(fw);
        PrintWriter pw = new PrintWriter(bw);
        Scanner reader = new Scanner(new File(filepath));
        reader.useDelimiter("[,\n]");

        while(reader.hasNext())
        {
            id = reader.next();
            name = reader.next();
            password = reader.next();
            if(!id.trim().equals(removeTerm.trim()))
            {
                pw.println(id + "," + name + "," + password);
            }
        }
        reader.close();
        pw.flush();
        pw.close();
        oldFile.delete();
        File dump = new File(filepath);
        newFile.renameTo(dump);
    }catch (Exception e)
}

```

```
{JOptionPane.showMessageDialog(null,"User Database Not
Found", "Error", JOptionPane.ERROR_MESSAGE); }
}

//main program
//Program run here
public class ArcheryClubRegistrationApp{
    static Login log;
    static JFrame frame;
    static JFrame frame2;
    public static void main(String args[]){
        log = new Login();

        frame = new JFrame ("Archery Club");
        frame.setDefaultCloseOperation (JFrame.EXIT_ON_CLOSE);

        //CourtApp app = new CourtApp();
        boolean logged = log.checkLogin();

        if(!logged){
            frame.getContentPane().add (new LoginPanel(frame,
log));
            frame.pack();
            frame.setVisible (true);
        }
        else{
            frame2.getContentPane().add (new MemberPanel());
            frame2.getContentPane().repaint();
            frame2.getContentPane().revalidate();
            frame2.pack();
            frame2.setVisible (true);
        }
    }
}
```