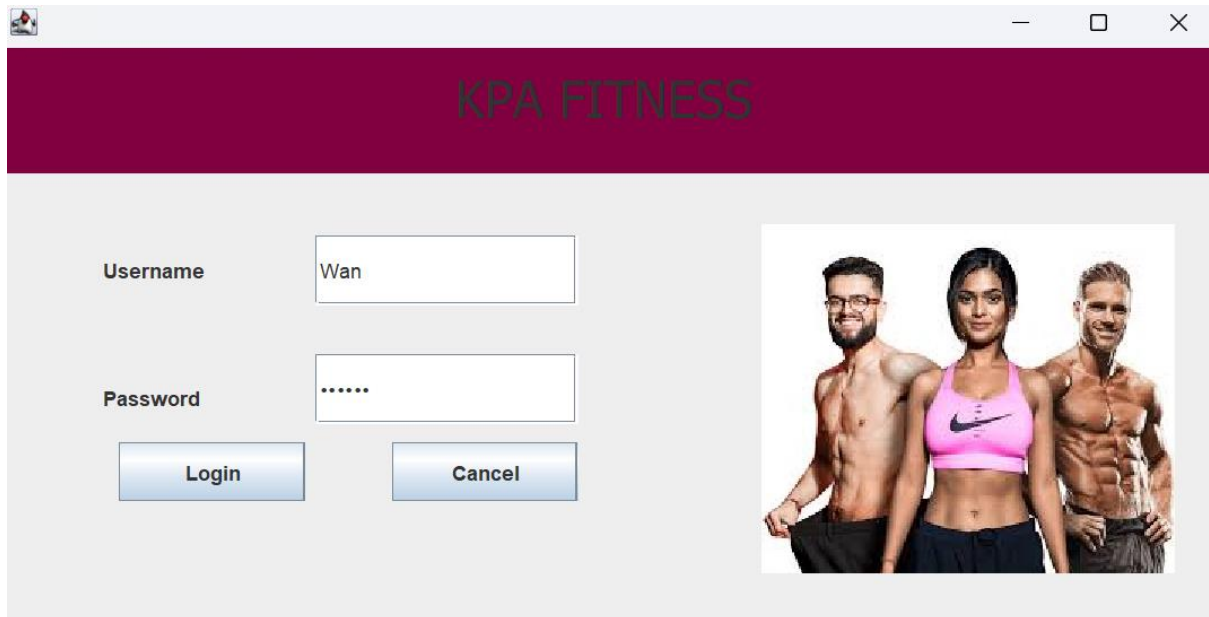


Homework javafx

1. Login Interface



```
JFrame frame;  
private JTextField textField_1;  
private JPasswordField passwordField;  
  
/**  
 * Launch the application.  
 */  
public static void main(String[] args) {  
    EventQueue.invokeLater(new Runnable() {  
        public void run() {  
            try {  
                Kpa window = new Kpa();  
                window.frame.setVisible(true);  
            } catch (Exception e) {  
                e.printStackTrace();  
            }  
        }  
    });  
}  
  
/**  
 * Create the application.  
 */  
public Kpa() {  
    initialize();  
}  
  
/**  
 * Initialize the contents of the frame.  
 */  
private void initialize() {  
    frame = new JFrame();
```

```

frame.getContentPane().setEnabled(true);
frame.setBounds(100, 100, 735, 380);
frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
frame.getContentPane().setLayout(null);

textField_1 = new JTextField();
textField_1.setColumns(10);
textField_1.setBounds(188, 111, 155, 41);
frame.getContentPane().add(textField_1);

passwordField = new JPasswordField();
passwordField.setBounds(188, 181, 155, 41);
frame.getContentPane().add(passwordField);

JPanel panel = new JPanel();
panel.setBackground(new Color(128, 0, 64));
panel.setBounds(0, 0, 721, 74);
frame.getContentPane().add(panel);
panel.setLayout(null);

JLabel lblNewLabel_1 = new JLabel("KPA FITNESS");
lblNewLabel_1.setBounds(270, 11, 209, 37);
lblNewLabel_1.setFont(new Font("Tahoma", Font.PLAIN, 30));
lblNewLabel_1.setBackground(new Color(255, 255, 255));
panel.add(lblNewLabel_1);

JButton btnNewButton = new JButton("Cancel");
btnNewButton.setBounds(233, 233, 110, 35);
frame.getContentPane().add(btnNewButton);

JButton btnLogin = new JButton("Login");
btnLogin.setBounds(72, 233, 110, 35);
frame.getContentPane().add(btnLogin);

// Add ActionListener to the "Login" button
btnLogin.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        frame.dispose(); // Close the current frame
        Coach coach = new Coach(); // Create an instance of Coach class
        coach.setVisible(true); // Set the Coach frame visible
    }
});

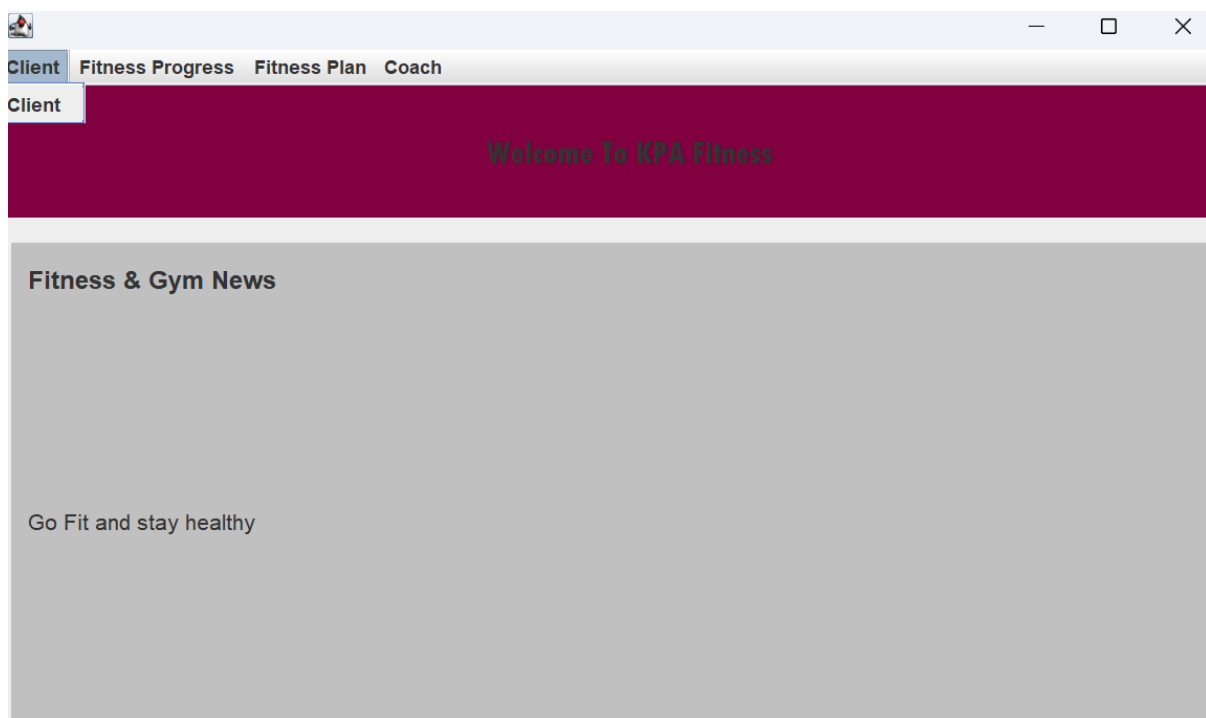
JLabel lblUsername = new JLabel("Username");
lblUsername.setBounds(62, 119, 87, 24);
frame.getContentPane().add(lblUsername);

JLabel lblPassword = new JLabel("Password");
lblPassword.setBounds(62, 194, 87, 24);
frame.getContentPane().add(lblPassword);

JLabel lblimg = new JLabel(new
ImageIcon(this.getClass().getResource("/img.jpg")));
lblimg.setBounds(435, 83, 276, 249);
frame.getContentPane().add(lblimg);
}
}

```

2. Main interface



```
package kpa_fitness;
import java.awt.EventQueue;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;

import javax.swing.JFrame;
import javax.swing.JMenu;
import javax.swing.JMenuBar;
import javax.swing.JMenuItem;
import javax.swing.JPanel;
import java.awt.Color;
import javax.swing.JLabel;
import javax.swing.SwingConstants;
import java.awt.Font;

public class Main {

    private JFrame frame;

    /**
     * Launch the application.
     */
    public static void Main(String[] args) {
        EventQueue.invokeLater(new Runnable() {
            public void run() {
                try {
                    Main window = new Main();
                    window.frame.setVisible(true);
                } catch (Exception e) {
                    e.printStackTrace();
                }
            }
        });
    }
}
```

```

    });
}

/**
 * Create the application.
 */
public Coach() {
    initialize();
}

/**
 * Initialize the contents of the frame.
 */
private void initialize() {
    frame = new JFrame();
    frame.setBounds(100, 100, 793, 468);
    frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);

    // Create the menu bar
    JMenuBar menuBar = new JMenuBar();
    frame.setJMenuBar(menuBar);

    // Create each menu item
    JMenu menuClient = new JMenu("Client");
    JMenuItem menuItemClient = new JMenuItem("Client");
    menuItemClient.addActionListener(new ActionListener() {
        public void actionPerformed(ActionEvent e) {
            // Add your action when Client is clicked
        }
    });
    menuClient.add(menuItemClient);
    menuBar.add(menuClient);

    JMenu menuFitnessProgress = new JMenu("Fitness Progress");
    JMenuItem menuItemFitnessProgress = new JMenuItem("Fitness
Progress");
    menuItemFitnessProgress.addActionListener(new ActionListener() {
        public void actionPerformed(ActionEvent e) {
            // Add your action when Fitness Progress is clicked
        }
    });
    menuFitnessProgress.add(menuItemFitnessProgress);
    menuBar.add(menuFitnessProgress);

    JMenu menuFitnessPlan = new JMenu("Fitness Plan");
    JMenuItem menuItemFitnessPlan = new JMenuItem("Fitness Plan");
    menuItemFitnessPlan.addActionListener(new ActionListener() {
        public void actionPerformed(ActionEvent e) {
            // Add your action when Fitness Plan is clicked
        }
    });
    menuFitnessPlan.add(menuItemFitnessPlan);
    menuBar.add(menuFitnessPlan);

    JMenu menuCoach = new JMenu("Coach");
    JMenuItem menuItemCoach = new JMenuItem("Coach");
    menuItemCoach.addActionListener(new ActionListener() {
        public void actionPerformed(ActionEvent e) {
            // Add your action when Coach is clicked

```

```

    }
});
menuCoach.add(menuItemCoach);
menuBar.add(menuCoach);
frame.getContentPane().setLayout(null);

JPanel panel = new JPanel();
panel.setBackground(new Color(128, 0, 64));
panel.setBounds(0, 0, 779, 84);
frame.getContentPane().add(panel);
panel.setLayout(null);

JLabel lblNewLabel = new JLabel("Welcome To KPA Fitness");
lblNewLabel.setFont(new Font("Tw Cen MT Condensed Extra Bold",
Font.PLAIN, 20));
lblNewLabel.setBounds(312, 11, 255, 62);
panel.add(lblNewLabel);

// News Section
JPanel newsPanel = new JPanel();
newsPanel.setBackground(Color.LIGHT_GRAY);
newsPanel.setBounds(10, 100, 763, 319);
frame.getContentPane().add(newsPanel);
newsPanel.setLayout(null);

JLabel lblNewLabel_1 = new JLabel("Fitness & Gym News");
lblNewLabel_1.setFont(new Font("Arial", Font.BOLD, 16));
lblNewLabel_1.setBounds(10, 11, 200, 25);
newsPanel.add(lblNewLabel_1);

JLabel newsLabel = new JLabel("Go Fit and stay healthy");
newsLabel.setFont(new Font("Arial", Font.PLAIN, 14));
newsLabel.setBounds(10, 47, 743, 261);
newsPanel.add(newsLabel);
}

public void setVisible(boolean b) {
    // TODO Auto-generated method stub
}
}

```

3. Client Interface

The screenshot shows a Java Swing window titled "Fitness App". The window contains a vertical stack of text input fields for "Client Name", "Client Phone", "Fitness Plan", "Fitness Progress", "Coach Name", and "Coach Phone". A "Submit" button is located below the "Coach Phone" field. The window has standard OS window controls (minimize, maximize, close) in the top right corner.

```
import javax.swing.*;
import java.awt.*;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;

public class Main {

    public static void main(String[] args) {

        SwingUtilities.invokeLater(() -> {

            JFrame frame = new JFrame("Fitness App");

            frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);

            frame.setSize(400, 400);

            JPanel panel = new JPanel();

            panel.setLayout(new BoxLayout(panel, BoxLayout.Y_AXIS));

            JTextField clientNameField = new JTextField(15);

            JTextField clientPhoneField = new JTextField(15);
```

```

    JTextField fitnessPlanField = new JTextField(15);
    JTextField fitnessProgressField = new JTextField(15);
    JTextField coachNameField = new JTextField(15);
    JTextField coachPhoneField = new JTextField(15);

    JButton submitButton = new JButton("Submit");
    JTextArea outputArea = new JTextArea(10, 30);
    outputArea.setEditable(false);

    submitButton.addActionListener(new ActionListener() {
        @Override
        public void actionPerformed(ActionEvent e) {
            String clientName = clientNameField.getText();
            String clientPhone = clientPhoneField.getText();
            String fitnessPlan = fitnessPlanField.getText();
            String fitnessProgress = fitnessProgressField.getText();
            String coachName = coachNameField.getText();
            String coachPhone = coachPhoneField.getText();

            Coach coach = new Coach(coachName, coachPhone);
            Client client = new Client(clientName, clientPhone, fitnessPlan, fitnessProgress,
coach);

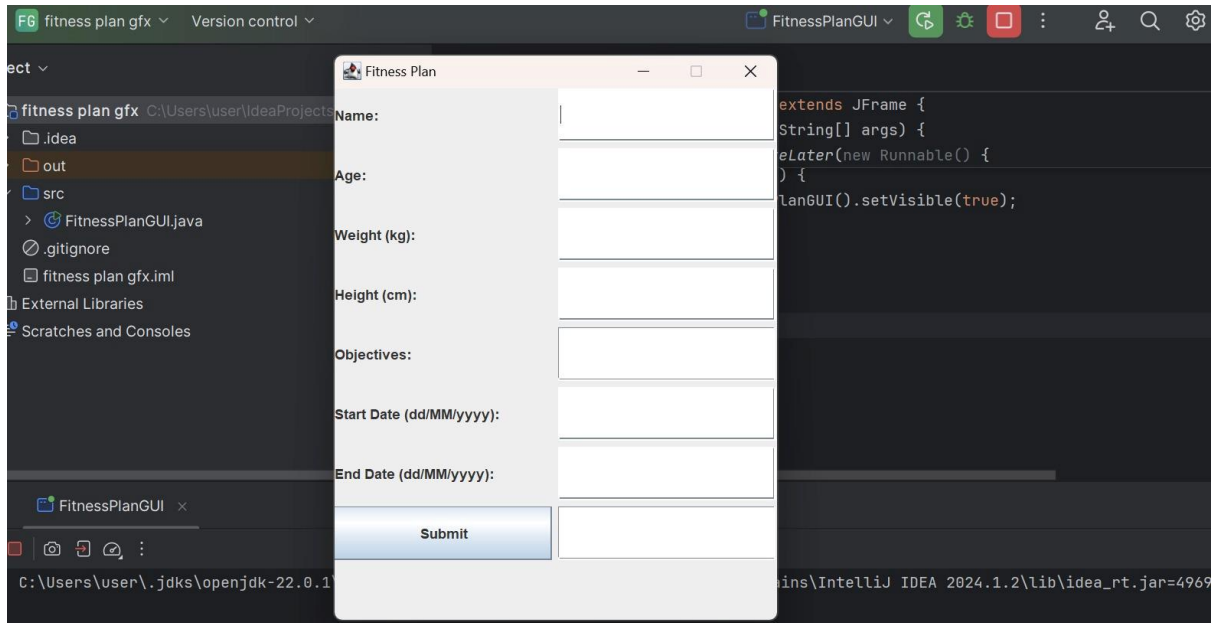
            outputArea.setText("Client Name: " + client.getName() + "\n"
                + "Client Phone: " + client.getPhoneNumber() + "\n"
                + "Fitness Plan: " + client.getFitnessPlan() + "\n"
                + "Fitness Progress: " + client.getFitnessProgress() + "\n"
                + "Coach Name: " + client.getCoach().getName() + "\n"
                + "Coach Phone: " + client.getCoach().getPhoneNumber());
        }
    });

```

```
panel.add(new JLabel("Client Name:"));
panel.add(clientNameField);
panel.add(new JLabel("Client Phone:"));
panel.add(clientPhoneField);
panel.add(new JLabel("Fitness Plan:"));
panel.add(fitnessPlanField);
panel.add(new JLabel("Fitness Progress:"));
panel.add(fitnessProgressField);
panel.add(new JLabel("Coach Name:"));
panel.add(coachNameField);
panel.add(new JLabel("Coach Phone:"));
panel.add(coachPhoneField);
panel.add(submitButton);
panel.add(new JScrollPane(outputArea));

frame.getContentPane().add(panel);
frame.setVisible(true);
});
}
}
```


3. FitnessPlan Interface



```
import javax.swing.*;
import java.awt.*;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.text.ParseException;
import java.text.SimpleDateFormat;
import java.util.Date;
```

```
// Define the FitnessPlan class
```

```
class FitnessPlan {
    private String name;
    private int age;
    private double weight;
    private double height;
    private double bmi;
    private String[] objectives;
```

```
private Date startDate;
```

```
private Date endDate;
```

```
public FitnessPlan(String name, int age, double weight, double height, String[] objectives,  
Date startDate, Date endDate) {
```

```
    this.name = name;
```

```
    this.age = age;
```

```
    this.weight = weight;
```

```
    this.height = height;
```

```
    this.objectives = objectives;
```

```
    this.startDate = startDate;
```

```
    this.endDate = endDate;
```

```
    this.bmi = calculateBMI(weight, height);
```

```
}
```

```
public String getName() {
```

```
    return name;
```

```
}
```

```
public int getAge() {
```

```
    return age;
```

```
}
```

```
public double getWeight() {
```

```
    return weight;
```

```
}
```

```
public double getHeight() {
```

```
    return height;
```

```
}
```

```

public double getBMI() {
    return bmi;
}

public String[] getObjectives() {
    return objectives;
}

public String getStartDate() {
    return formatDate(startDate);
}

public String getEndDate() {
    return formatDate(endDate);
}

private double calculateBMI(double weight, double height) {
    return weight / Math.pow(height / 100, 2);
}

private String formatDate(Date date) {
    SimpleDateFormat formatter = new SimpleDateFormat("dd/MM/yyyy");
    return formatter.format(date);
}
}

// Define the FitnessPlanGUI class
public class FitnessPlanGUI extends JFrame {
    private JTextField nameField;
    private JTextField ageField;
    private JTextField weightField;

```

```
private JTextField heightField;

private JTextArea objectivesArea;

private JTextField startDateField;

private JTextField endDateField;

private JButton submitButton;

private JTextArea outputArea;


public FitnessPlanGUI() {

    createView();


    setTitle("Fitness Plan");

    setDefaultCloseOperation(EXIT_ON_CLOSE);

    setSize(400, 500);

    setLocationRelativeTo(null);

    setResizable(false);
}


private void createView() {

    JPanel panel = new JPanel();

    getContentPane().add(panel);


    panel.setLayout(new GridLayout(9, 2, 5, 5));


    JLabel nameLabel = new JLabel("Name:");

    panel.add(nameLabel);

    nameField = new JTextField();

    panel.add(nameField);


    JLabel ageLabel = new JLabel("Age:");

    panel.add(ageLabel);

    ageField = new JTextField();
```

```
panel.add(ageField);
```

```
JLabel weightLabel = new JLabel("Weight (kg):");
```

```
panel.add(weightLabel);
```

```
weightField = new JTextField();
```

```
panel.add(weightField);
```

```
JLabel heightLabel = new JLabel("Height (cm):");
```

```
panel.add(heightLabel);
```

```
heightField = new JTextField();
```

```
panel.add(heightField);
```

```
JLabel objectivesLabel = new JLabel("Objectives:");
```

```
panel.add(objectivesLabel);
```

```
objectivesArea = new JTextArea();
```

```
panel.add(new JScrollPane(objectivesArea));
```

```
JLabel startDateLabel = new JLabel("Start Date (dd/MM/yyyy):");
```

```
panel.add(startDateLabel);
```

```
startDateField = new JTextField();
```

```
panel.add(startDateField);
```

```
JLabel endDateLabel = new JLabel("End Date (dd/MM/yyyy):");
```

```
panel.add(endDateLabel);
```

```
endDateField = new JTextField();
```

```
panel.add(endDateField);
```

```
submitButton = new JButton("Submit");
```

```
panel.add(submitButton);
```

```
outputArea = new JTextArea();
```

```

outputArea.setEditable(false);

panel.add(new JScrollPane(outputArea));

submitButton.addActionListener(new ActionListener() {
    @Override
    public void actionPerformed(ActionEvent e) {
        submitFitnessPlan();
    }
});
}

private void submitFitnessPlan() {
    try {
        String name = nameField.getText().trim();
        int age = Integer.parseInt(ageField.getText().trim());
        double weight = Double.parseDouble(weightField.getText().trim());
        double height = Double.parseDouble(heightField.getText().trim());
        String[] objectives = objectivesArea.getText().split("\\n");
        String startDate = startDateField.getText().trim();
        String endDate = endDateField.getText().trim();

        if (name.isEmpty() || startDate.isEmpty() || endDate.isEmpty()) {
            showError("Name, start date, and end date are required.");
            return;
        }

        Date start = parseDate(startDate);
        Date end = parseDate(endDate);

        if (start.after(end)) {
            showError("Start date must be before end date.");
        }
    }
}

```

```

        return;
    }

    FitnessPlan plan = new FitnessPlan(name, age, weight, height, objectives, start, end);
    displayFitnessPlan(plan);
} catch (NumberFormatException ex) {
    showError("Age, weight, and height must be valid numbers.");
} catch (ParseException ex) {
    showError("Dates must be in the format dd/MM/yyyy.");
}
}

private void showError(String message) {
    JOptionPane.showMessageDialog(this, message, "Error", JOptionPane.ERROR_MESSAGE);
}

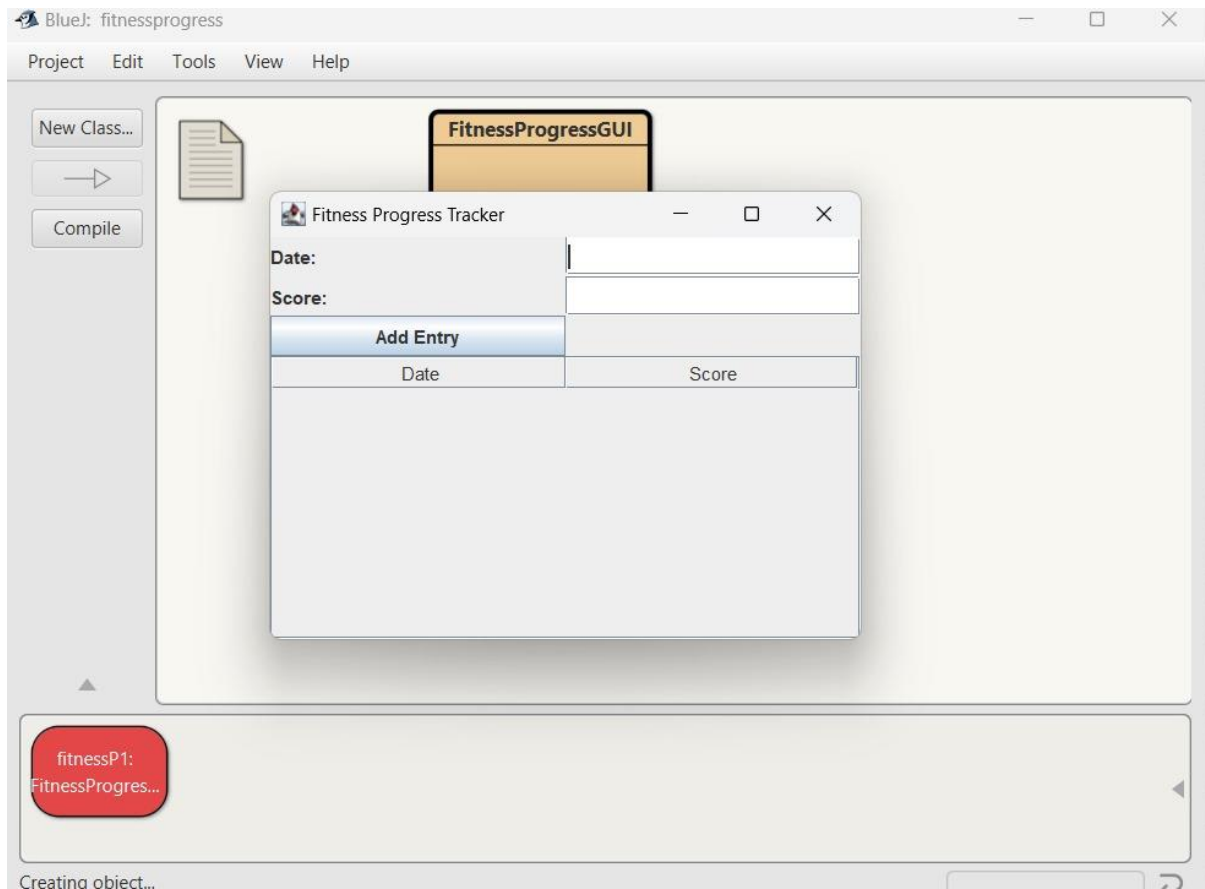
private void displayFitnessPlan(FitnessPlan plan) {
    outputArea.setText("Name: " + plan.getName() + "\n"
        + "Age: " + plan.getAge() + "\n"
        + "Weight: " + plan.getWeight() + " kg\n"
        + "Height: " + plan.getHeight() + " cm\n"
        + "BMI: " + String.format("%.2f", plan.getBMI()) + "\n"
        + "Start Date: " + plan.getStartDate() + "\n"
        + "End Date: " + plan.getEndDate() + "\n"
        + "Objectives:\n");

    for (String objective : plan.getObjectives()) {
        outputArea.append("- " + objective + "\n");
    }
}
}

```

```
private Date parseDate(String date) throws ParseException {  
    SimpleDateFormat formatter = new SimpleDateFormat("dd/MM/yyyy");  
    return formatter.parse(date);  
}  
  
public static void main(String[] args) {  
    SwingUtilities.invokeLater(new Runnable() {  
        @Override  
        public void run() {  
            new FitnessPlanGUI().setVisible(true);  
        }  
    });  
}  
}
```


4. FitnessProgress interface



```
import javax.swing.*;
import javax.swing.table.DefaultTableModel;
import java.awt.*;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
```

```
public class FitnessProgressGUI {
    private JFrame frame;
    private JTextField dateField;
    private JTextField scoreField;
    private DefaultTableModel tableModel;

    public FitnessProgressGUI() {
        // Create the frame
```

```
frame = new JFrame("Fitness Progress Tracker");
frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
frame.setSize(400, 300);
frame.setLayout(new BorderLayout());

// Create the input panel
JPanel inputPanel = new JPanel();
inputPanel.setLayout(new GridLayout(3, 2));

// Add date label and text field
inputPanel.add(new JLabel("Date:"));
dateField = new JTextField();
inputPanel.add(dateField);

// Add score label and text field
inputPanel.add(new JLabel("Score:"));
scoreField = new JTextField();
inputPanel.add(scoreField);

// Add submit button
JButton submitButton = new JButton("Add Entry");
inputPanel.add(submitButton);

// Add a placeholder to keep grid layout consistent
inputPanel.add(new JLabel(""));

frame.add(inputPanel, BorderLayout.NORTH);

// Create the table to display data
String[] columnNames = {"Date", "Score"};
tableModel = new DefaultTableModel(columnNames, 0);
```

```

JTable table = new JTable(tableModel);

// Add the table to a scroll pane
JScrollPane scrollPane = new JScrollPane(table);
frame.add(scrollPane, BorderLayout.CENTER);

// Add action listener to the submit button
submitButton.addActionListener(new ActionListener() {

    @Override

    public void actionPerformed(ActionEvent e) {

        String date = dateField.getText();

        String score = scoreField.getText();

        if (!date.isEmpty() && !score.isEmpty()) {

            tableModel.addRow(new Object[]{date, score});

            dateField.setText("");

            scoreField.setText("");

        } else {

            JOptionPane.showMessageDialog(frame, "Please enter both date and score.", "Input
Error", JOptionPane.ERROR_MESSAGE);

        }

    }

});

// Make the frame visible

frame.setVisible(true);

}

public static void main(String[] args) {

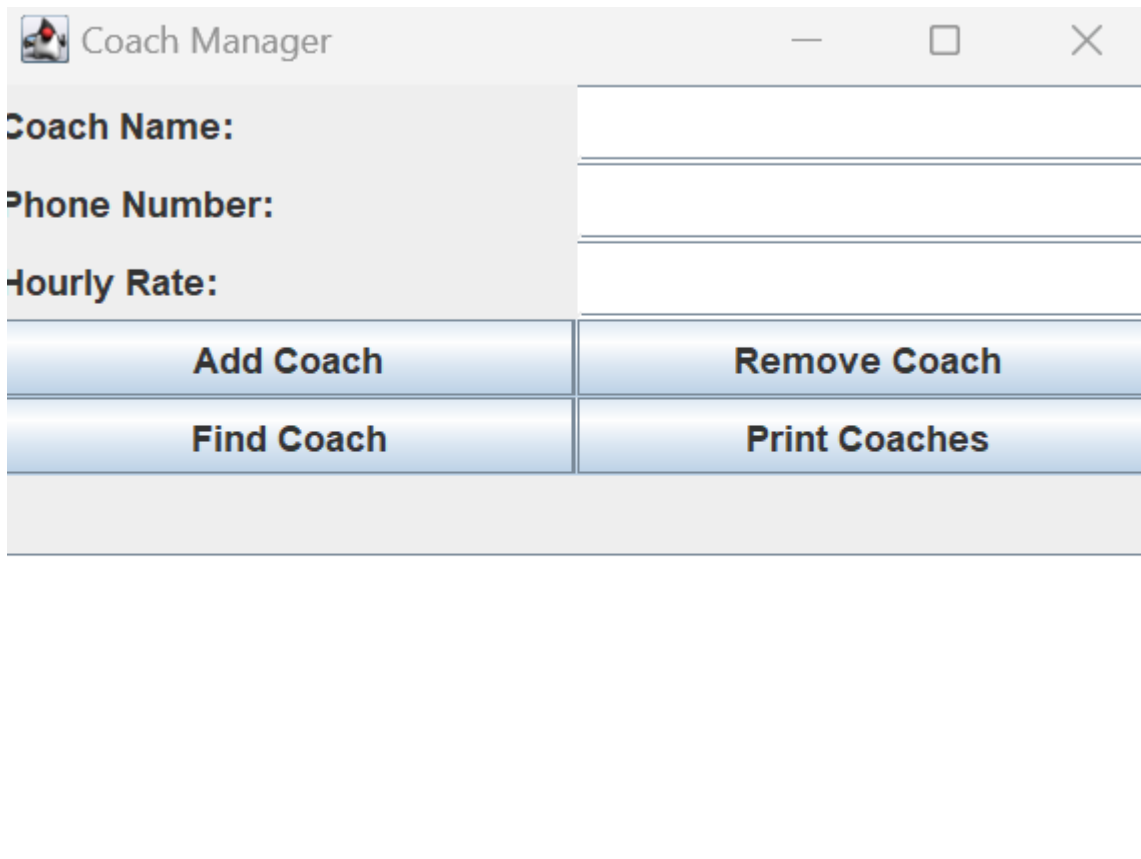
    new FitnessProgressGUI();

}

```

}

5. Coach Interface



The screenshot shows a Java Swing window titled "Coach Manager". The window has a light gray title bar with standard minimize, maximize, and close buttons. The main content area is divided into two columns. The left column contains three labels: "Coach Name:", "Phone Number:", and "Hourly Rate:", each followed by a text input field. The right column contains four buttons: "Add Coach", "Remove Coach", "Find Coach", and "Print Coaches", arranged vertically. The buttons have a blue gradient and a 3D effect. Below the buttons is a large, empty rectangular area, likely for displaying a list of coaches.

```
package kpa_fitness;
import javax.swing.*.*;
import java.awt.*.*;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.util.ArrayList;
import java.util.List;

public class Coach {
    private List<Coach> coaches;
    private JFrame frame;
    private JTextField coachNameField;
    private JTextField coachPhoneNumberField;
    private JTextField hourlyRateField;
    private JTextArea displayArea;

    public Coach() {
        coaches = new ArrayList<>();
        initializeGUI();
    }

    private void initializeGUI() {
        frame = new JFrame("Coach Manager");
```

```

frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
frame.setSize(400, 300);

JPanel panel = new JPanel();
panel.setLayout(new GridLayout(6, 2));

panel.add(new JLabel("Coach Name:"));
coachNameField = new JTextField();
panel.add(coachNameField);

panel.add(new JLabel("Phone Number:"));
coachPhoneNumberField = new JTextField();
panel.add(coachPhoneNumberField);

panel.add(new JLabel("Hourly Rate:"));
hourlyRateField = new JTextField();
panel.add(hourlyRateField);

JButton addButton = new JButton("Add Coach");
addButton.addActionListener(new ActionListener() {
    @Override
    public void actionPerformed(ActionEvent e) {
        addCoach(coachNameField.getText(),
coachPhoneNumberField.getText(), Double.parseDouble(hourlyRateField.getText()));
    }
});
panel.add(addButton);

JButton removeButton = new JButton("Remove Coach");
removeButton.addActionListener(new ActionListener() {
    @Override
    public void actionPerformed(ActionEvent e) {
        removeCoach(coachNameField.getText());
    }
});
panel.add(removeButton);

JButton findButton = new JButton("Find Coach");
findButton.addActionListener(new ActionListener() {
    @Override
    public void actionPerformed(ActionEvent e) {
        findCoach(coachNameField.getText());
    }
});
panel.add(findButton);

JButton printButton = new JButton("Print Coaches");
printButton.addActionListener(new ActionListener() {
    @Override
    public void actionPerformed(ActionEvent e) {
        printCoaches();
    }
});
panel.add(printButton);

displayArea = new JTextArea();
displayArea.setEditable(false);
JScrollPane scrollPane = new JScrollPane(displayArea);

```

```

        frame.getContentPane().add(panel, BorderLayout.NORTH);
        frame.getContentPane().add(scrollPane, BorderLayout.CENTER);

        frame.setVisible(true);
    }

    public void addCoach(String name, String phoneNumber, double hourlyRate) {
        coaches.add(new Coach());
        displayArea.append("Added coach: " + name + "\n");
    }

    public void removeCoach(String name) {
        boolean removed = coaches.removeIf(coach -> coach.getName().equals(name));
        if (removed) {
            displayArea.append("Removed coach: " + name + "\n");
        } else {
            displayArea.append("Coach not found: " + name + "\n");
        }
    }

    private Object getName() {
        // TODO Auto-generated method stub
        return null;
    }

    public void printCoaches() {
        displayArea.append("Current Coaches:\n");
        for (Coach coach : coaches) {
            displayArea.append("Name: " + coach.getName() + ", Phone Number: " +
coach.getPhoneNumber() + ", Hourly Rate: " + coach.getHourlyRate() + "\n");
        }
    }

    private String getHourlyRate() {
        // TODO Auto-generated method stub
        return null;
    }

    private String getPhoneNumber() {
        // TODO Auto-generated method stub
        return null;
    }

    public void findCoach(String name) {
        for (Coach coach : coaches) {
            if (coach.getName().equals(name)) {
                displayArea.append("Found coach: Name: " + coach.getName() + ",
Phone Number: " + coach.getPhoneNumber() + ", Hourly Rate: " +
coach.getHourlyRate() + "\n");
                return;
            }
        }
        displayArea.append("Coach not found: " + name + "\n");
    }

    public static void main(String[] args) {
        SwingUtilities.invokeLater(new Runnable() {
            @Override
            public void run() {

```

```

        new Coach();
    }
});
}

class CoachManager {
    private String name;
    private String phoneNumber;
    private double hourlyRate;
    private List<Client> clients;

    public CoachManager(String name, String phoneNumber, double hourlyRate) {
        this.name = name;
        this.phoneNumber = phoneNumber;
        this.hourlyRate = hourlyRate;
        this.clients = new ArrayList<>();
    }

    public String getName() {
        return name;
    }

    public String getPhoneNumber() {
        return phoneNumber;
    }

    public double getHourlyRate() {
        return hourlyRate;
    }

    public List<Client> getClients() {
        return clients;
    }

    public void addClient(Client client) {
        clients.add(client);
    }

    public void removeClient(Client client) {
        clients.remove(client);
    }
}

class Client {
    private String name;
    private String phoneNumber;

    public Client(String name, String phoneNumber) {
        this.name = name;
        this.phoneNumber = phoneNumber;
    }

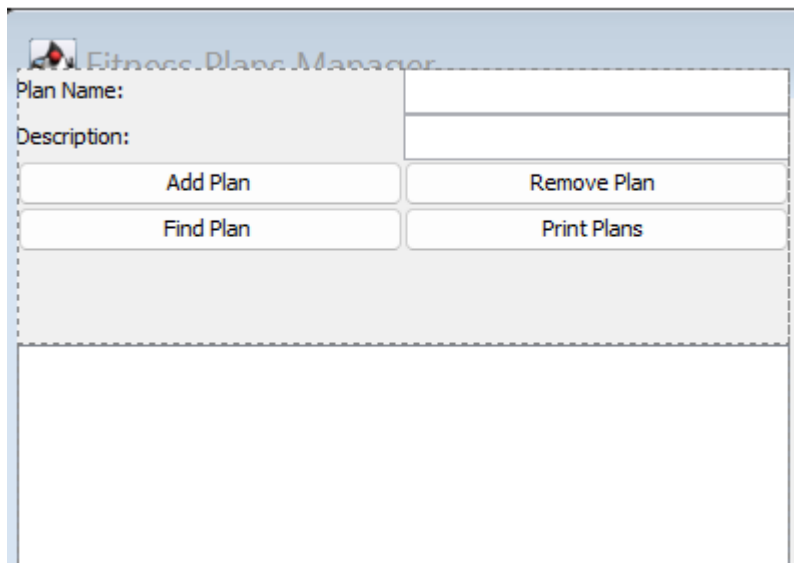
    public String getName() {
        return name;
    }

    public String getPhoneNumber() {
        return phoneNumber;
    }
}

```

```
        @Override
        public String toString() {
            return name + " (" + phoneNumber + ")";
        }
    }
}
```


6. Find FitnessPlan Interface



```
import javax.swing.*;
import java.awt.*;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.util.ArrayList;

public class Coach {
    private ArrayList<FitnessPlan> plans;
    private JFrame frame;
    private JTextField nameField;
    private JTextField descriptionField;
    private JTextArea displayArea;

    public Coach() {
        plans = new ArrayList<>();
        initializeGUI();
    }

    private void initializeGUI() {
```

```
frame = new JFrame("Fitness Plans Manager");  
frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);  
frame.setSize(400, 300);
```

```
JPanel panel = new JPanel();  
panel.setLayout(new GridLayout(6, 2));
```

```
panel.add(new JLabel("Plan Name:"));  
nameField = new JTextField();  
panel.add(nameField);
```

```
panel.add(new JLabel("Description:"));  
descriptionField = new JTextField();  
panel.add(descriptionField);
```

```
JButton addButton = new JButton("Add Plan");  
addButton.addActionListener(new ActionListener() {  
    @Override  
    public void actionPerformed(ActionEvent e) {  
        addPlan(nameField.getText(), descriptionField.getText());  
    }  
});  
panel.add(addButton);
```

```
JButton removeButton = new JButton("Remove Plan");  
removeButton.addActionListener(new ActionListener() {  
    @Override  
    public void actionPerformed(ActionEvent e) {  
        removePlan(nameField.getText());  
    }  
});
```

```
panel.add(removeButton);
```

```
JButton findButton = new JButton("Find Plan");  
findButton.addActionListener(new ActionListener() {  
    @Override  
    public void actionPerformed(ActionEvent e) {  
        findPlan(nameField.getText());  
    }  
});  
panel.add(findButton);
```

```
JButton printButton = new JButton("Print Plans");  
printButton.addActionListener(new ActionListener() {  
    @Override  
    public void actionPerformed(ActionEvent e) {  
        printPlans();  
    }  
});  
panel.add(printButton);
```

```
displayArea = new JTextArea();  
displayArea.setEditable(false);  
JScrollPane scrollPane = new JScrollPane(displayArea);
```

```
frame.getContentPane().add(panel, BorderLayout.NORTH);  
frame.getContentPane().add(scrollPane, BorderLayout.CENTER);
```

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

```
public void addPlan(String name, String description) {
```

```

        plans.add(new FitnessPlan(name, description));
        displayArea.append("Added plan: " + name + "\n");
    }

    public void removePlan(String name) {
        boolean removed = plans.removeIf(plan -> plan.getName().equals(name));
        if (removed) {
            displayArea.append("Removed plan: " + name + "\n");
        } else {
            displayArea.append("Plan not found: " + name + "\n");
        }
    }

    public void printPlans() {
        displayArea.append("Current Plans:\n");
        for (FitnessPlan plan : plans) {
            displayArea.append("Name: " + plan.getName() + ", Description: " + plan.getDescription()
+ "\n");
        }
    }

    public void findPlan(String name) {
        for (FitnessPlan plan : plans) {
            if (plan.getName().equals(name)) {
                displayArea.append("Found plan: Name: " + plan.getName() + ", Description: " +
plan.getDescription() + "\n");
                return;
            }
        }
        displayArea.append("Plan not found: " + name + "\n");
    }

```

```
public static void main(String[] args) {  
    SwingUtilities.invokeLater(new Runnable() {  
        @Override  
        public void run() {  
            new Coach();  
        }  
    });  
}  
}  
  
class FitnessPlan {  
    private String name;  
    private String description;  
  
    public FitnessPlan(String name, String description) {  
        this.name = name;  
        this.description = description;  
    }  
  
    public String getName() {  
        return name;  
    }  
  
    public String getDescription() {  
        return description;  
    }  
}
```