Relational Databases with MySQL Week 6 Coding Assignment

Points possible: 70

Category	Criteria	% of Grade
Functionality	Does the code work?	25
Organization	Is the code clean and organized? Proper use of white space, syntax, and consistency are utilized. Names and comments are concise and clear.	25
Creativity	Student solved the problems presented in the assignment using creativity and out of the box thinking.	25
Completeness	All requirements of the assignment are complete.	25

Instructions: In Eclipse, or an IDE of your choice, write the code that accomplishes the objectives listed below. Ensure that the code compiles and runs as directed. Take screenshots of the code and of the running program (make sure to get screenshots of all required functionality) and paste them in this document where instructed below. Create a new repository on GitHub for this week's assignments and push this document, with your Java project code, to the repository. Add the URL for this week's repository to this document where instructed and submit this document to your instructor when complete.

Coding Steps:

This week you will be working together as a **team** to create a full CRUD application.

Your console CRUD application will need to use a database to store all the application data.

As a team, decide what you want your project to do. Get instructor approval early in the week before beginning development.

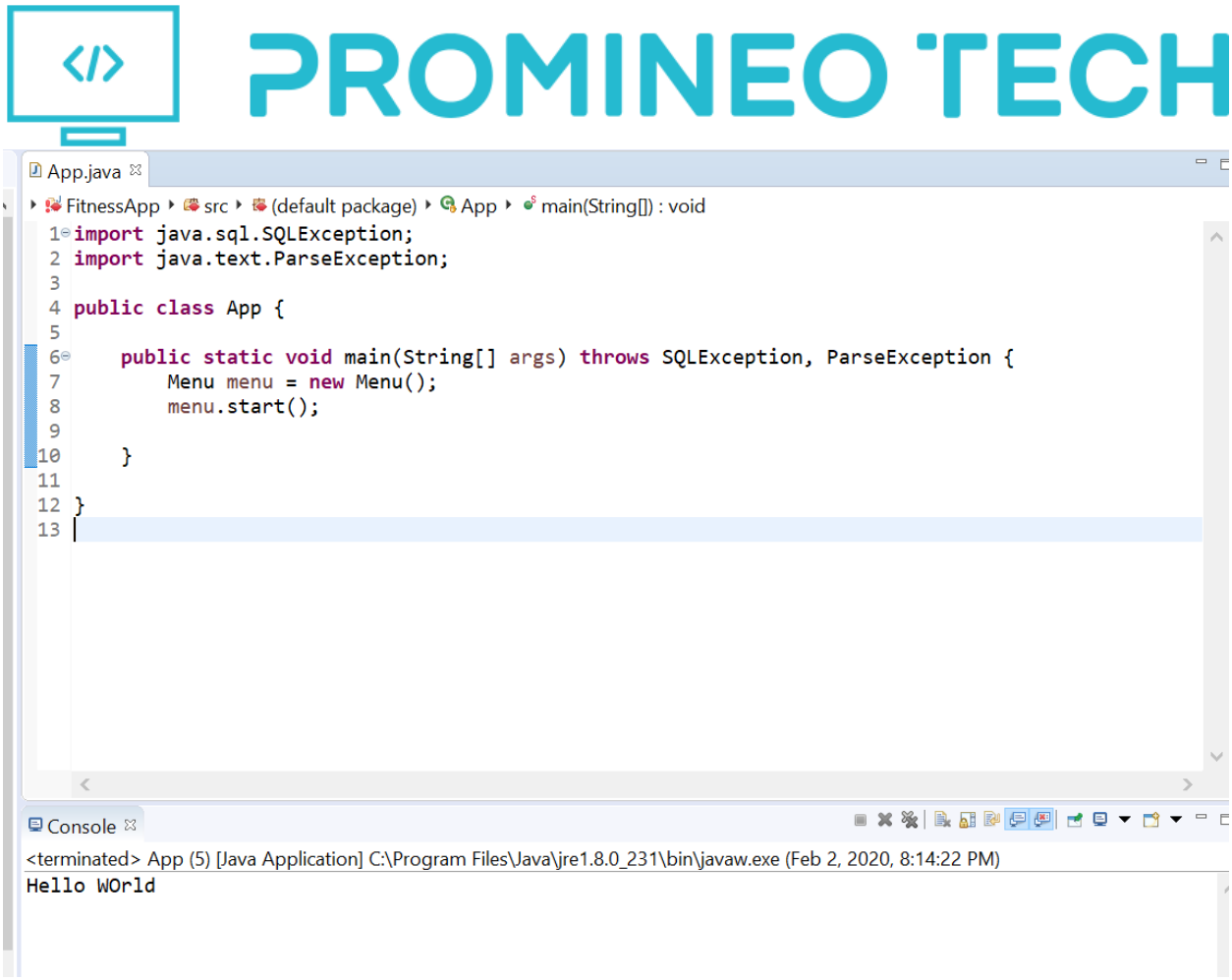
You need to have at least 3 entities.

Users should be able to interact via the console (i.e. Scanner(System.in)))

Use git to collaborate.

Everyone will be graded on their individual contributions.

Screenshots of Code:


PROMINEO TECH
App.java
FitnessApp ▸ src ▸ (default package) ▸ App ▸ main(String[]) : void
1 import java.sql.SQLException;
2 import java.text.ParseException;
3
4 public class App {
5
6 public static void main(String[] args) throws SQLException, ParseException {
7 Menu menu = new Menu();
8 menu.start();
9
10 }
11
12 }
13
Console
<terminated> App (5) [Java Application] C:\Program Files\Java\jre1.8.0_231\bin\javaw.exe (Feb 2, 2020, 8:14:22 PM)
Hello WOrld

PROMINEO TECH

*DBConnector.java

FitnessApp ▸ src ▸ (default package) ▸ DBConnector ▸ PASSWORD : String

```java


import java.sql.Connection;

public class DBConnector {

    private static final String URL = "jdbc:mysql://localhost:3306/FitnessDB";
    private static final String USERNAME = "root";
    private static final String PASSWORD = "";

    private static Connection instance;

    public static Connection getConnection() {
        if (instance == null) {
            try {
                instance = DriverManager.getConnection(URL, USERNAME, PASSWORD);

            } catch (SQLException e) {
                System.out.println("Connection Failed!!");
                e.printStackTrace();
            }

        }
        return instance;

    }

}

```

PROMINEO TECH

Menu.java

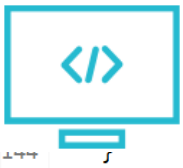
FitnessApp ▸ src ▸ (default package) ▸ Menu ▸

```java
import java.sql.Date;

public class Menu {

    private ClientsDao clientsDao = new ClientsDao();
    private ClassesDao classesDao = new ClassesDao();
    private InstructorsDao instructorDao = new InstructorsDao();

    private Scanner scanner = new Scanner(System.in);
    private List<String> options = Arrays.asList(
            "View List of All Classes",
            "View List of All Instructors",
            "View List of All Clients",
            "View Clients Within a Specific Class",
            "Update Class Day/Time",
            "Delete Class",
            "Add New Class",
            "Update Client's Class",
            "Delete Client",
            "Add New Client",
            "Update Instructor Pay Rate",
            "Delete Instructor",
            "Add New Instrcutor");

    public void start() throws SQLException, ParseException {
        String selection = "";

        do {
            printMenu();
            selection = scanner.nextLine();

            try {
                if(selection.equals("1")) {
                    displayClasses();
                } else if (selection.equals("2")) {
```

```
50                  viewAllInstructors();
51              } else if (selection.equals("3")) {
52                  viewAllClients();
53              } else if (selection.equals("4")) {
54                  viewClientsInSpecificClass();
55              } else if (selection.equals("5")) {
56                  updateClassDateAndTime();
57              } else if (selection.equals("6")) {
58                  deleteClass();
59              } else if (selection.equals("7")) {
60                  createClass();
61              } else if (selection.equals("8")) {
62                  updateClientClass();
63              } else if (selection.equals("9")) {
64                  deleteClient();
65              } else if (selection.equals("10")) {
66                  addNewClient();
67              } else if (selection.equals("11")) {
68                  updatePayRate();
69              } else if (selection.equals("12")) {
70                  deleteInstructor();
71              } else if (selection.equals("13")) {
72                  addNewInstructor();
73              }
74          } catch (SQLException e) {
75              e.printStackTrace();
76          }
77
78          System.out.println("Press enter to continue...");
79          scanner.nextLine();
80      } while (!selection.equals("-1"));
81  }
```

PROMINEO TECH

```
82
83      private void printMenu() {
84          System.out.println("Select an Option:\n--------------------------------");
85          for (int i = 0; i < options.size(); i++) {
86              System.out.println(i + 1 + ") " + options.get(i));
87          }
88      }
89
90
91      private void displayClasses() throws SQLException {
92          List<Classes> classes = classesDao.getClasses();
93          for (Classes classe : classes) {
94              System.out.println(classe.getClassId() + ": " + classe.getClassType() + "  Date and
95          }
96      }
97
98
99      private void deleteInstructor() throws SQLException {
100         System.out.println("Enter nstructor ID you would like to remove: ");
101         int instructor_ID = Integer.parseInt(scanner.nextLine());
102
103         instructorDao.removeInstructor(instructor_ID);
104         System.out.println("Instructor Removed");
105     }
106
107
108     private void updatePayRate() throws SQLException {
109         System.out.println("Enter Instructor ID: ");
110         int instructor_ID = Integer.parseInt(scanner.nextLine());
111         System.out.println("Enter Pay Rate (xxxx.xx): ");
112         double pay_rate = Double.parseDouble(scanner.nextLine());
113         instructorDao.updatePay(instructor_ID, pay_rate);
114
114
115         System.out.println("Pay Rate Updated");
116     }
117
118
119     private void addNewInstructor() throws SQLException {
120
121
122         System.out.println("Enter First Name: ");
123         String instructor_FN = scanner.nextLine();
124         System.out.println("Enter Last Name: ");
125         String instructor_LN = scanner.nextLine();
126         System.out.println("Enter Classes (Zumba, Yoga, ...)");
127         String classes_taught = scanner.nextLine();
128         System.out.println("Enter Pay Rate (xxxx.xx): ");
129         double pay_rate = Double.parseDouble(scanner.nextLine());
130
131         instructorDao.newInstructor(instructor_FN, instructor_LN, classes_taught, pay_rate);
132
133         System.out.println("New Instructor Added");
134     }
135
136
137     private void viewAllInstructors() throws SQLException {
138         List<Instructors> instructors = instructorDao.getInstructors();
139         for (Instructors instructor : instructors) {
140             System.out.println("Instructor ID: " + instructor.getInstructor_ID() + ", Instructo
141         }
142
143
144     }
```



PROMINEO TECH

```
145
146    private void viewAllClients() throws SQLException {
147        List<Clients> clients = clientsDao.getClients();
148        for (Clients client : clients) {
149            System.out.println("Client ID:" + client.getId() + ", Client Name:" + client.getfNa
150            + ", DOB:" + client.getBirthdate() + ", Class ID:" + client.getClassId());
151        }
152    }
153
154    private void viewClientsInSpecificClass() throws SQLException {
155        System.out.println("Enter class id: ");
156        int classId = Integer.parseInt(scanner.nextLine());
157        Clients clients = clientsDao.getClientsByClassID(classId);
158        System.out.println(clients.getfName() + " " + clients.getlName());
159    }
160
161    private void updateClassDateAndTime() throws SQLException {
162        System.out.println("Enter Class ID you want to change the date and time for: ");
163        int classId = Integer.parseInt(scanner.nextLine());
164        System.out.println("Enter new date and time: ");
165        String dateAndTime = scanner.nextLine();
166        classesDao.updateClassDateAndTimeById(classId, dateAndTime);
167    }
168
169    private void deleteClass() throws SQLException {
170        System.out.println("Enter the class ID you would like to delete: ");
171        int classId = Integer.parseInt(scanner.nextLine());
172        classesDao.deleteClassById(classId);
173    }
174
175    private void createClass() throws SQLException {
176        System.out.println("Enter class type: ");
177        String classType = scanner.nextLine();
178        System.out.println("Enter the date and time of the class: ");
```

PROMINEO TECH

```
74
75    private void createClass() throws SQLException {
76        System.out.println("Enter class type: ");
77        String classType = scanner.nextLine();
78        System.out.println("Enter the date and time of the class: ");
79        String dateAndTime = scanner.nextLine();
80        classesDao.createNewClassById(classType, dateAndTime);
81    }
82
83
84    private void updateClientClass() throws SQLException {
85        System.out.println("Enter client's new class id:");
86        int classId = Integer.parseInt(scanner.nextLine());
87        System.out.println("Enter client id:");
88        int clientId = Integer.parseInt(scanner.nextLine());
89        clientsDao.updateClientClassById(classId, clientId);
90    }
91
92    private void deleteClient() throws SQLException {
93        System.out.println("Enter client id to delete:");
94        int id = Integer.parseInt(scanner.nextLine());
95        clientsDao.deleteClientById(id);
96    }
97
98    private void addNewClient() throws SQLException, ParseException {
99        System.out.println("Enter first name:");
100       String fName = scanner.nextLine();
101       System.out.println("Enter last name");
102       String lName = scanner.nextLine();
103       System.out.println("Enter birthdate (YYYY-MM-DD");
104       SimpleDateFormat simpleDateFormat = new SimpleDateFormat("YYYY-MM-DD");
105       Date birthdate = (Date) simpleDateFormat.parse(scanner.nextLine());
106       System.out.println("Enter 4-digit class ID:");
107       int classId = Integer.parseInt(scanner.nextLine());
108       clientsDao.createNewClient(fName, lName, birthdate, classId);
```

PROMINEO TECH

FitnessApp ▸ src ▸ dao ▸ ClassesDao ▸

```java
package dao;

import java.sql.Connection;


public class ClassesDao {

    private Connection connection;
    private ClassesDao classesDao;
    private final String GET_CLASSES_QUERY = "SELECT * FROM CLASSES";
    private final String UPDATE_CLASS_DATE_AND_TIME_QUERY = "UPDATE classes SET date_and_time =
    private final String DELETE_CLASS_QUERY = "DELETE FROM classes WHERE class_id = ?";
    private final String CREATE_NEW_CLASS_QUERY = "INSERT INTO classes(type, date_and_time) VALU

    public ClassesDao() {
        connection = DBConnector.getConnection();
        classesDao = new ClassesDao();

    }

    public List<Classes> getClasses() throws SQLException {
        ResultSet rs = connection.prepareStatement(GET_CLASSES_QUERY).executeQuery();
        List<Classes> classes = new ArrayList<Classes>();

        while (rs.next()) {
            classes.add(populateClasses(rs.getInt(1), rs.getString(2), rs.getString(3)));
        }
        return classes;
    }

    public void updateClassDateAndTimeById(int classId, String dateAndTime) throws SQLException
        PreparedStatement ps = connection.prepareStatement(UPDATE_CLASS_DATE_AND_TIME_QUERY);
        ps.setInt(1, classId);
        ps.setString(2, dateAndTime);
        ps.executeUpdate();
    }

    public void deleteClassById(int classId) throws SQLException {
        PreparedStatement ps = connection.prepareStatement(DELETE_CLASS_QUERY);
        ps.setInt(1, classId);
        ps.executeUpdate();
    }

    public void createNewClassById(String classType, String dateAndTime) throws SQLException {
        PreparedStatement ps = connection.prepareStatement(CREATE_NEW_CLASS_QUERY);
        ps.setString(1, classType);
        ps.setString(2, dateAndTime);
        ps.executeUpdate();
    }



    private Classes populateClasses(int classId, String classType, String dateAndTime) throws SQ
        return new Classes(classId, classType, dateAndTime, classesDao.getClientsByClassID(class

    }

public class ClassesDao {


}

```

PROMINEO TECH

ClientsDao.java

FitnessApp ▸ src ▸ dao ▸ ClientsDao ▸

```java
package dao;

import java.sql.Connection;

public class ClientsDao {

    private Connection connection;
    private final String GET_CLIENTS_QUERY = "SELECT * FROM clients";
    private final String GET_CLIENTS_BY_CLASSID_QUERY = "SELECT * FROM clients WHERE class_id = ?";
    private final String UPDATE_CLIENT_CLASS_QUERY = "UPDATE clients SET class_id = ? WHERE id = ?";
    private final String DELETE_CLIENT_BY_ID_QUERY = "DELETE FROM clients WHERE id = ?";
    private final String CREATE_NEW_CLIENT_QUERY = "INSERT INTO clients(client_fn, client_ln, birthdate, class id) VALUES(????)";


    public ClientsDao() {
        connection = DBConnector.getConnection();
    }

    public List<Clients> getClients() throws SQLException {
        ResultSet rs = connection.prepareStatement(GET_CLIENTS_QUERY).executeQuery();
        List<Clients> clients = new ArrayList<Clients>();

        while (rs.next() ) {
            clients.add(populateClients(rs.getInt(1), rs.getString(2), rs.getString(3), rs.getDate(4), rs.getInt(5)));
        }

        return clients;
    }

    public Clients getClientsByClassID(int classID) throws SQLException {
        PreparedStatement ps = connection.prepareStatement(GET_CLIENTS_BY_CLASSID_QUERY);
        ps.setInt(1, classID);
        ResultSet rs = ps.executeQuery();
        rs.next();
        return populateClients(rs.getInt(1), rs.getString(2), rs.getString(3), rs.getDate(4), rs.getInt(5));
    }

    public void updateClientClassById (int classId, int clientId) throws SQLException {
        PreparedStatement ps = connection.prepareStatement(UPDATE_CLIENT_CLASS_QUERY);
        ps.setInt(1, classId);
        ps.setInt(2, clientId);
        ps.executeUpdate();
    }

    public void deleteClientById(int id) throws SQLException {
        PreparedStatement ps = connection.prepareStatement(DELETE_CLIENT_BY_ID_QUERY);
        ps.setInt(1, id);
        ps.executeUpdate();
    }

    public void createNewClient(String fName, String lName, Date birthdate, int classId) throws SQLException {
        PreparedStatement ps = connection.prepareStatement(CREATE_NEW_CLIENT_QUERY);
        ps.setString(1, fName);
        ps.setString(2, lName);
        ps.setDate(3, birthdate);
        ps.setInt(4, classId);
        ps.executeUpdate();
    }

    private Clients populateClients(int id, String fName, String lName, Date birthdate, int classId) {
        return new Clients (id, fName, lName, birthdate, classId);
    }


}
```

PROMINEO TECH

FitnessApp ▸ src ▸ dao ▸ InstructorsDao ▸

```java
package dao;

import java.sql.Connection;


public class InstructorsDao {

    private Connection connection = DBConnector.getConnection();

    private String GET_INSTRUCTORS = "SELECT * FROM instructors";



    private final  String NEW_INSTRUCTOR = "Insert into food(instructor_FN, instructor_LN, classes_taught, pay_rate)" +
            "values(?,?,?,?)";
    private String GET_INSTRUCTORS = "SELECT * FROM instructors";
    private String UPDATE_PAY = "update instructors set pay_rate = ? where instructor_ID = ?";
    private String REMOVE_INSTRUCTOR = "delete from instructors where instructor_ID = ?";

    //NEW INSTRUCTOR
    public void newInstructor(String instructor_FN, String instructor_LN, String classes_taught, double pay_rate) throws SQLExcepti
        PreparedStatement ps = connection.prepareStatement(NEW_INSTRUCTOR);
        ps.setString(1, instructor_FN);
        ps.setString(2, instructor_LN);
        ps.setString(3, classes_taught);
        ps.setDouble(4, pay_rate);
        ps.executeUpdate();
    }

    //GET ALL INSTRUCTORS
    public List<Instructors> getInstructors() throws SQLException {
        ResultSet rs = connection.prepareStatement(GET_INSTRUCTORS).executeQuery();
        List<Instructors> instructors = new ArrayList<Instructors>();

         while (rs.next() ) {
             instructors.add(populateInstructors(rs.getInt(1), rs.getString(2), rs.getString(3), rs.getString(4), rs.getDouble(5)));
         }

         return instructors;
     }

    private  Instructors populateInstructors(int instructor_ID, String instructor_FN, String instructor_LN, String classes_taught,

         return populateInstructors(instructor_ID, instructor_FN, instructor_LN, classes_taught, pay_rate);
     }

}



    public void updatePay(int instructor_ID, double pay_rate) throws SQLException {
        PreparedStatement ps = connection.prepareStatement(UPDATE_PAY);
        ps.setDouble(1, pay_rate);
        ps.setInt(2, instructor_ID);
        ps.executeUpdate();
    }

    public  void removeInstructor(int instructor_ID) throws SQLException {
        PreparedStatement ps = connection.prepareStatement(REMOVE_INSTRUCTOR);
        ps.setInt(1, instructor_ID);
        ps.executeUpdate();
    }
}
```

```java
package entity;


import java.util.List;

public class Classes {

    private int classId;
    private String classType;

    private String dateAndTime;
    private List<Clients> clients;

    public Classes(int classId, String classType, String dateAndTime, List<Clients> clients) {
        this.setClassId(classId);
        this.setClassType(classType);
        this.setDateAndTime(dateAndTime);
        this.setClients(clients);

    }

    public int getClassId() {
        return classId;
    }

    public void setClassId(int classId) {
        this.classId = classId;
    }

    public String getClassType() {
        return classType;
    }

    public void setClassType(String classType) {
        this.classType = classType;
    }

    public String getDateAndTime() {
        return dateAndTime;
    }

    public void setDateAndTime(String dateAndTime) {
        this.dateAndTime = dateAndTime;
    }

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

    public void setClients(List<Clients> clients) {
        this.clients = clients;
    private dateAndTime;
    private List<Clients> clients;

    public Classes(int classId, String clssType, List<Clients> clients) {
        this.classId = classId;
        this.classType = classType;
        this.clients = clients;


    }


}

```

PROMINEO TECH

FitnessApp ▸ src ▸ entity ▸ Clients ▸

```java
package entity;

import java.sql.Date;

public class Clients {

    private int id;
    private String fName;
    private String lName;
    private Date birthdate;
    private int classId;


    public Clients (int id, String fName, String lName, Date birthdate, int classId) {
        this.setId(id);
        this.setfName(fName);
        this.setlName(lName);
        this.setBirthdate(birthdate);
        this.setClassId(classId);
    }


    public int getId() {
        return id;
    }
    public void setId(int id) {
        this.id = id;
    }
    public String getfName() {
        return fName;
    }
    public void setfName(String fName) {
        this.fName = fName;
```

PROMINEO TECH

```
33          this.fName = fName;
34      }
35      public String getlName() {
36          return lName;
37      }
38      public void setlName(String lName) {
39          this.lName = lName;
40      }
41
42
43      public Date getBirthdate() {
44          return birthdate;
45      }
46
47
48      public void setBirthdate(Date birthdate) {
49          this.birthdate = birthdate;
50      }
51
52
53      public int getClassId() {
54          return classId;
55      }
56
57
58      public void setClassId(int classId) {
59          this.classId = classId;
60      }
61
62
63  }
64
```

FitnessApp ▸ src ▸ entity ▸ Instructors ▸

```
1  package entity;
2
3  public class Instructors {
4
5      private int instructor_ID;
6      private String instructor_FN;
7      private String instructor_LN;
8      private String classes_taught;
9      private double pay_rate;
10
11
12     public Instructors(int instructor_ID, String instructor_FN, String instructor_LN, String classes_taught,
13             double pay_rate) {
14         super();
15         this.instructor_ID = instructor_ID;
16         this.instructor_FN = instructor_FN;
17         this.instructor_LN = instructor_LN;
18         this.classes_taught = classes_taught;
19         this.pay_rate = pay_rate;
20     }
21
22
23     public int getInstructor_ID() {
24         return instructor_ID;
25     }
26     public void setInstructor_ID(int instructor_ID) {
27         this.instructor_ID = instructor_ID;
28     }
29     public String getInstructor_FN() {
30         return instructor_FN;
31     }
32     public void setInstructor_FN(String instructor_FN) {
33         this.instructor_FN = instructor_FN;
34     }
```

```
30          return instructor_FN;
31      }
32      public void setInstructor_FN(String instructor_FN) {
33          this.instructor_FN = instructor_FN;
34      }
35      public String getInstructor_LN() {
36          return instructor_LN;
37      }
38      public void setInstructor_LN(String instructor_LN) {
39          this.instructor_LN = instructor_LN;
40      }
41      public String getClasses_taught() {
42          return classes_taught;
43      }
44      public void setClasses_taught(String classes_taught) {
45          this.classes_taught = classes_taught;
46      }
47      public double getPay_rate() {
48          return pay_rate;
49      }
50      public void setPay_rate(double pay_rate) {
51          this.pay_rate = pay_rate;
52      }
53
54
55  }
56
```

Screenshots of Running Application:

URL to GitHub Repository: https://github.com/Jammersg/FitnessApp