Misr University for Science & Technology

College: Information Technology

Course Name: CS 201

Project Title:

Gym Management System

Made by:

Mostafa Tarek 94071

Ahmed Abdelhamed 94120

Mohamed Yehia 94104

Mohamed Ezz Elragel 94303

Abo Elhamad Sharkawy 86517

Abstract: -

The main objective of the project is to develop a software that facilitates the data storage, data maintenance. To store the record of the customers, the staff that has the privileges to access, modify and delete any record and finally the service, gym provides to its customers.

Table of contents: -

1.Introduction: -

Our Gym Management Software is a gym membership management system. You can keep records on your members, their memberships, and have quick and easy communication between you and your members.

Gym Management also includes a booking system, point of sale, banking, accounting, concessions and has a range of reports that help in the management of your club. Our gym Management Software is a complete gym and recreation facility system program which looks after all your members, memberships, and activities. It is designed for gyms, recreation centers, and health clubs.

Our gym management Software provides lots of functions such data entry of customer, keeping records of all the things about customers fees, plan, and physical fitness which help to provide good quality of services to customer from Gym managers.

In this proposed system also provide the total information about machinery and data of coaches is also stored in it. Services provided by the gym are also handled by this system. This system structure is become very simple to understand because of Data Flow Diagram provided by us.

Our Goal To develop easy-to-use software which handles the customer-staff relationship in an effective manner. To develop a user-friendly system that requires minimal user training.

2.Analysis and Specification: -

Abstract Class Person: -

```
package gym;

public abstract class Person implements GymInterface {
    protected String ID;
    protected String firstName;
    protected String lastName;
    protected String birthDate;
    protected String age;
    protected String medicalCase;
    protected double money;
    protected String phoneNumber;
    protected boolean attendace;
    //this is constructor.
    public Person (String id , String fName , String lName , String BD , String Age ,
    this.ID=id;
    this.firstName=fName;
    this.lastName=lName;
    this.birthDate=BD;
    this.age=Age;
    this.medicalCase=mCase;
    this.money=Money;
    this.phoneNumber=PNumber;
    this.attendace=attend;
}

    public String getID() {
        return ID;
    }

    public void setID(String ID) {
        this.ID = ID;
    }

    public String getFirstName() {
        return firstName;
    }

    public void setFirstName(String firstName) {
        this.firstName = firstName;
    }

    public String getLastName() {
        return lastName;
    }

    public void setLastName(String lastName) {
        this.lastName = lastName;
    }

    public String getBirthDate() {
        return birthDate;
    }

    public void setBirthDate(String birthDate) {
        this.birthDate = birthDate;
    }

    public String getAge() {
        return age;
    }

    public void setAge(String age) {
        this.age = age;
    }

    public String getMedicalCase() {
        return medicalCase;
    }

    public void setMedicalCase(String medicalCase) {
        this.medicalCase = medicalCase;
    }

}
```

- The Person class is the super class for many classes, and it implements GymInterface.
- The fields in the class are protected for security, but encapsulation was used to access them.
- The constructor was created to initialize all the fields.

GymInterface Interface: -

```
package gym;

public interface GymInterface {
        public abstract double moneypay();
}
```

- The interface contains all the abstract methods to be overridden in its implementations.

Members Class: -

```
package gym;

public class  Members extends Person {

    public Members(String id, String fName, String lName, String BD, String Age, String mCase, double Money, String PNumber, boolean attend)
        super(id, fName, lName, BD, Age, mCase, Money, PNumber, attend);
    }
    public double getMoney() {
        return money;
    }

    public void setMoney(double money) {
        this.money = money;
    }

    public String getPhoneNumber() {
        return phoneNumber;
    }

    public void setPhoneNumber(String phoneNumber) {
        this.phoneNumber = phoneNumber;
    }

    public boolean isAttendace() {
        return attendace;
    }

    public void setAttendace(boolean attendace) {
        this.attendace = attendace;
    }

    @Override
        public double moneypay(){
            return 0;
        }
}
```

- This class extends the Person class to achieve reusability.
- The class is used to create Members objects and store them in the database.
- The constructor was created to call the superclass constructor.

HourlyStaff Class: -

```
package gym;


public class HourlyStaff extends Person  {
    protected double hours;
    protected double wage;
    //////////////
    public HourlyStaff (String id , String fName , String lName , String BD , String
    super(id,fName,lName,BD,Age,mCase,Money,PNumber,attend);
    setHours(hoursWorked);
    setWage(hourWage);
}

    public double getHours() {
        return hours;
    }

    public void setHours(double hoursWorked) {
        hours=((hoursWorked>=0.0)&&(hoursWorked<=168.0))?hoursWorked:0.0;
    }

    public double getWage() {
        return wage;
    }

    public void setWage(double hourWage) {
        wage=(hourWage<0.0)?0.0:hourWage;
    }

    public double getMoney() {
        return money;
    }

    public void setMoney(double money) {
        this.money = money;
    }

    public String getPhoneNumber() {
        return phoneNumber;
    }

    public void setPhoneNumber(String phoneNumber) {
        this.phoneNumber = phoneNumber;
    }

    public boolean isAttendace() {
        return attendace;
    }

    public void setAttendace(boolean attendace) {
        this.attendace = attendace;
    }
    @Override
        public double moneypay(){
            if(getHours()<=40)
                return getWage()*getHours();
            else
                return 40*getWage()*(getHours()-40)*getWage()*1.5;
        }
```

- This class extends the Person class to achieve reusability.
- The class is used to create Hourly Staff objects and store them in the database.
- The constructor was created to call the superclass constructor and initialize all the fields.
- The fields in the class are protected for security, but encapsulation was used to validate them and access them.
- The class overrides the “moneypay” method to calculate the salary of hourly staff.

CommissionStaff Class: -

```
package gym;


public class CommisionStaff extends Person {
    protected double commissionrate;
    protected double grosssales;
    public CommisionStaff (String id , String fName , String lName , String BD , Strin
        super(id,fName,lName,BD,Age,mCase,Money,PNumber,attend);
        setCommissionrate(rate);
        setGrosssales(sales);
}

    public double getCommissionrate() {
        return commissionrate;
    }

    public void setCommissionrate(double rate) {
        commissionrate =(rate>0.0&&rate<1.0)?rate:0.0;
    }

    public double getGrosssales() {
        return grosssales;
    }

    public void setGrosssales(double sales) {
        grosssales = (sales<0.0)?0.0:sales;
    }

    public double getMoney() {
        return money;
    }

    public void setMoney(double money) {
        this.money = money;
    }

    public String getPhoneNumber() {
        return phoneNumber;
    }

    public void setPhoneNumber(String phoneNumber) {
        this.phoneNumber = phoneNumber;
    }

    public boolean isAttendace() {
        return attendace;
    }

    public void setAttendace(boolean attendace) {
        this.attendace = attendace;
    }
    @Override
        public double moneypay(){
                return getCommissionrate()*getGrosssales();
        }
}
```

- This class extends the Person class to achieve reusability.
- The class is used to create commission Staff objects and store them in the database.
- The constructor was created to call the superclass constructor and initialize all the fields.
- The fields in the class are protected for security, but encapsulation was used to validate them and access them.
- The class overrides the “moneypay” method to calculate the salary of commission staff.

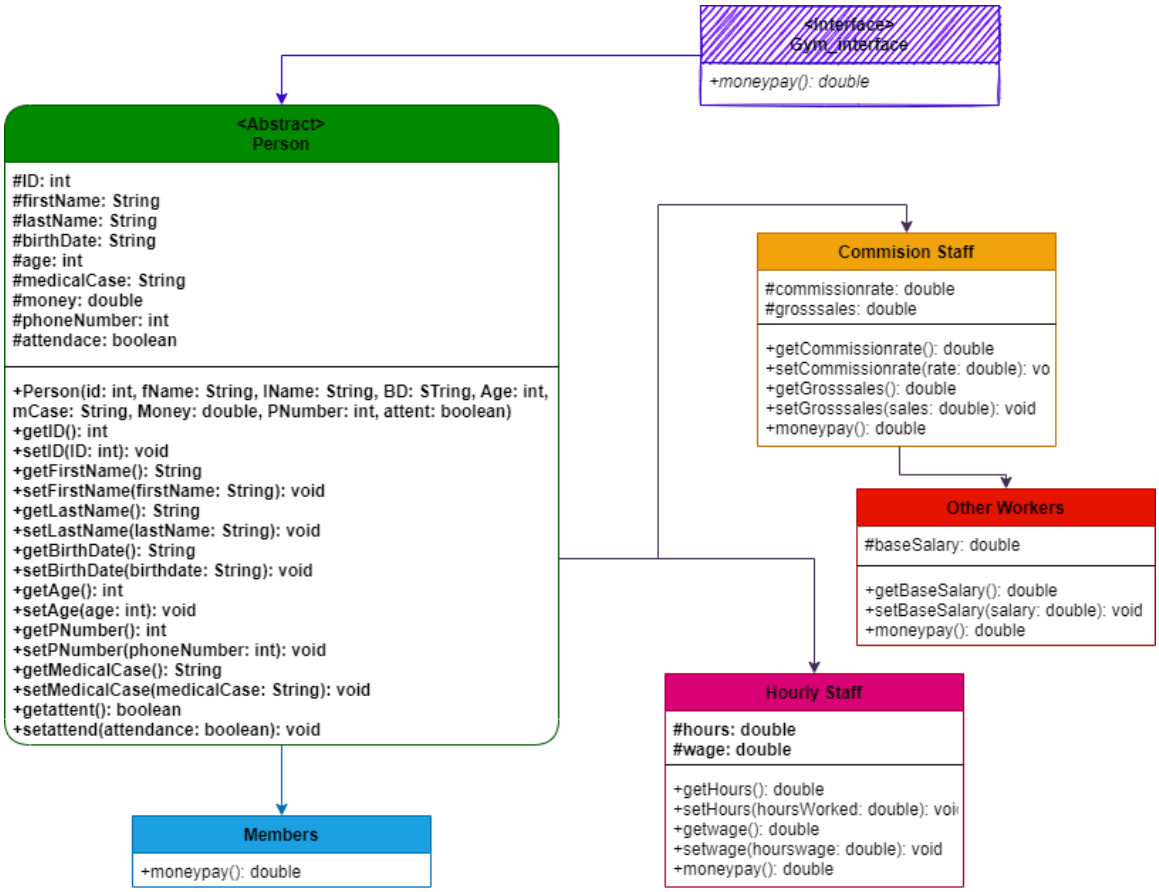
OtherWorkers Class: -

```
package gym;

public class OtherWorkers extends CommisionStaff {
    protected double baseSalray;
    public OtherWorkers(String id , String fName , String lName , String BI
        super(id,fName,lName,BD,Age,mCase,Money,PNumber,attend,rate,sales);
        setBaseSalray(salary);
    }

    public double getBaseSalray() {
        return baseSalray;
    }

    public void setBaseSalray(double salary) {
        baseSalray = (salary <0.0)?0.0:salary;
    }
    @Override
        public double moneypay(){
                return getBaseSalray()*super.moneypay();
        }
}
```

- This class extends the CommissionStaff class to achieve reusability.
- The class is used to create commission Staff objects and store them in the database.
- The constructor was created to call the superclass constructor and initialize all the fields.
- The fields in the class are protected for security, but encapsulation was used to validate them and access them.
- The class overrides the “moneypay” method to calculate the salary of other workers.

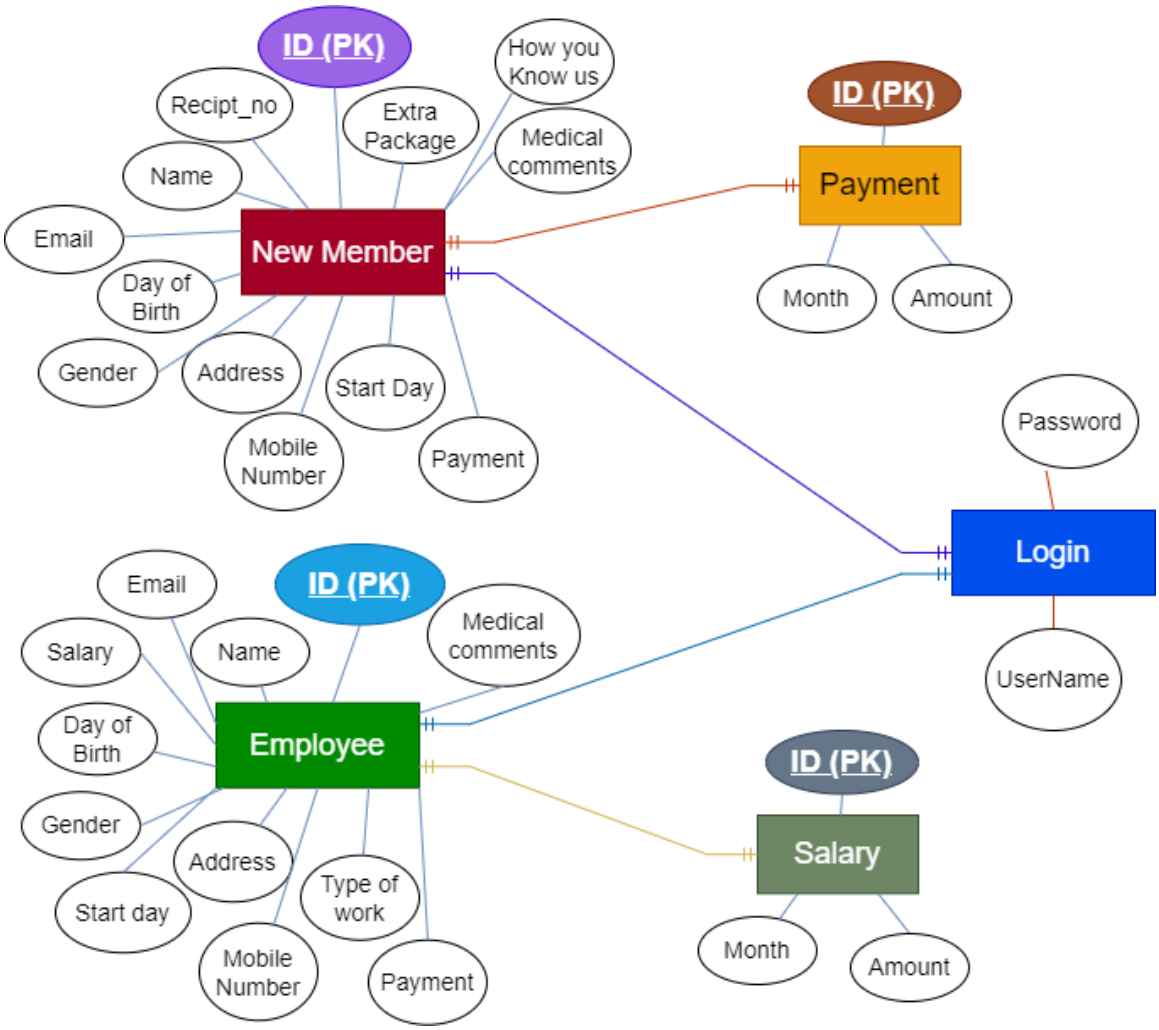
3.Design: -

UML Diagram: -



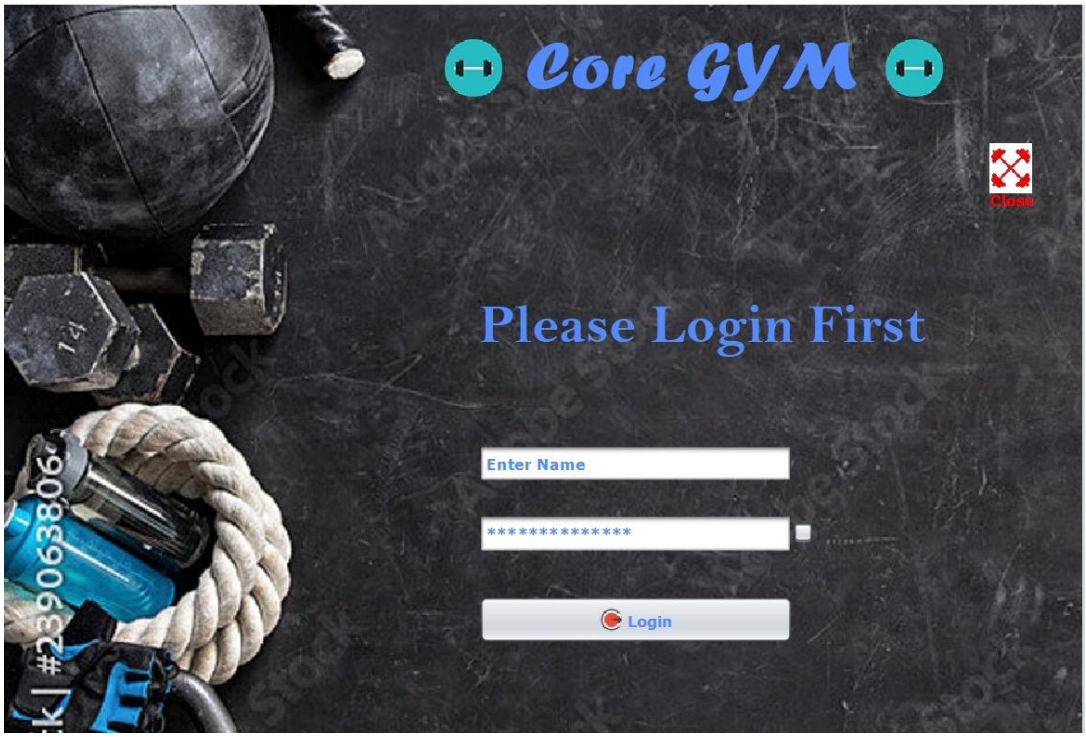
1. The GymInterface includes the “moneypay” abstract method.
2. Person abstract class implements the GymInterface. The class uses accessors and mutators to access the protected fields.
3. Members class inherits the Person abstract class and overrides its abstract method.
4. HourlyStaff class inherits the Person abstract class and overrides its abstract method.
5. CommissionStaff class inherits the Person abstract class and overrides its abstract method.
6. OtherWorkers inherits the CommssionStaff class.

ERD Diagram


ID (PK)
How you Know us
Recipt_no
Extra Package
Medical comments
Name
Email
New Member
Day of Birth
Gender
Address
Start Day
Mobile Number
Payment
ID (PK)
Payment
Month
Amount
Password
Login
UserName
Email
ID (PK)
Medical comments
Salary
Name
Day of Birth
Employee
Gender
Address
Type of work
Start day
Mobile Number
Payment
ID (PK)
Salary
Month
Amount

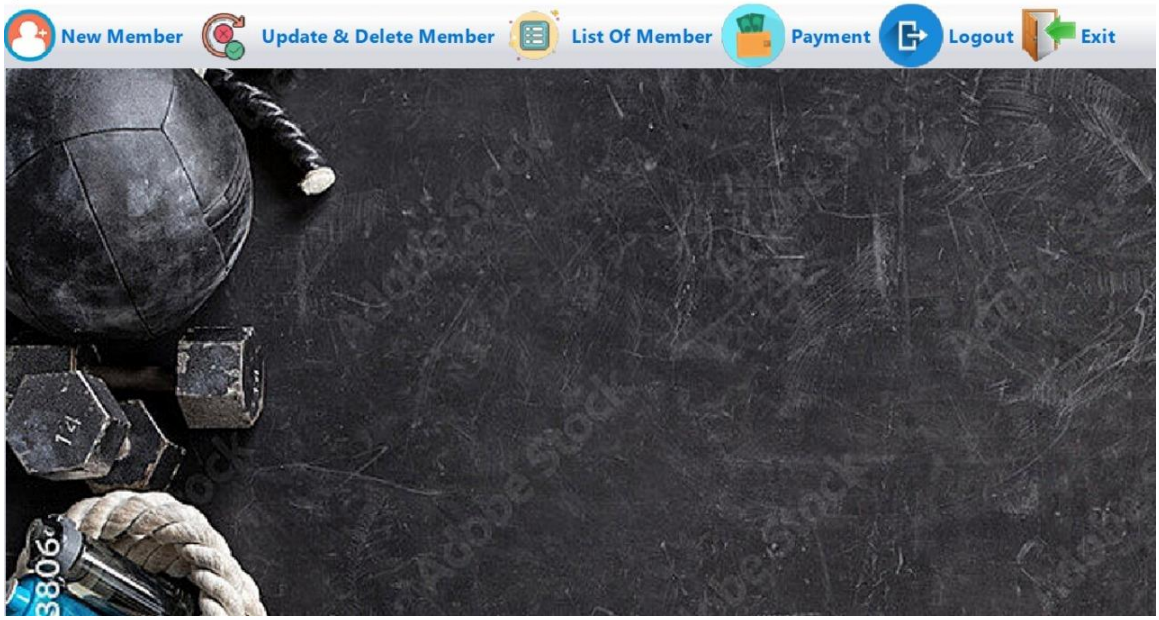
4.Implementaion: -

-Login screen: -

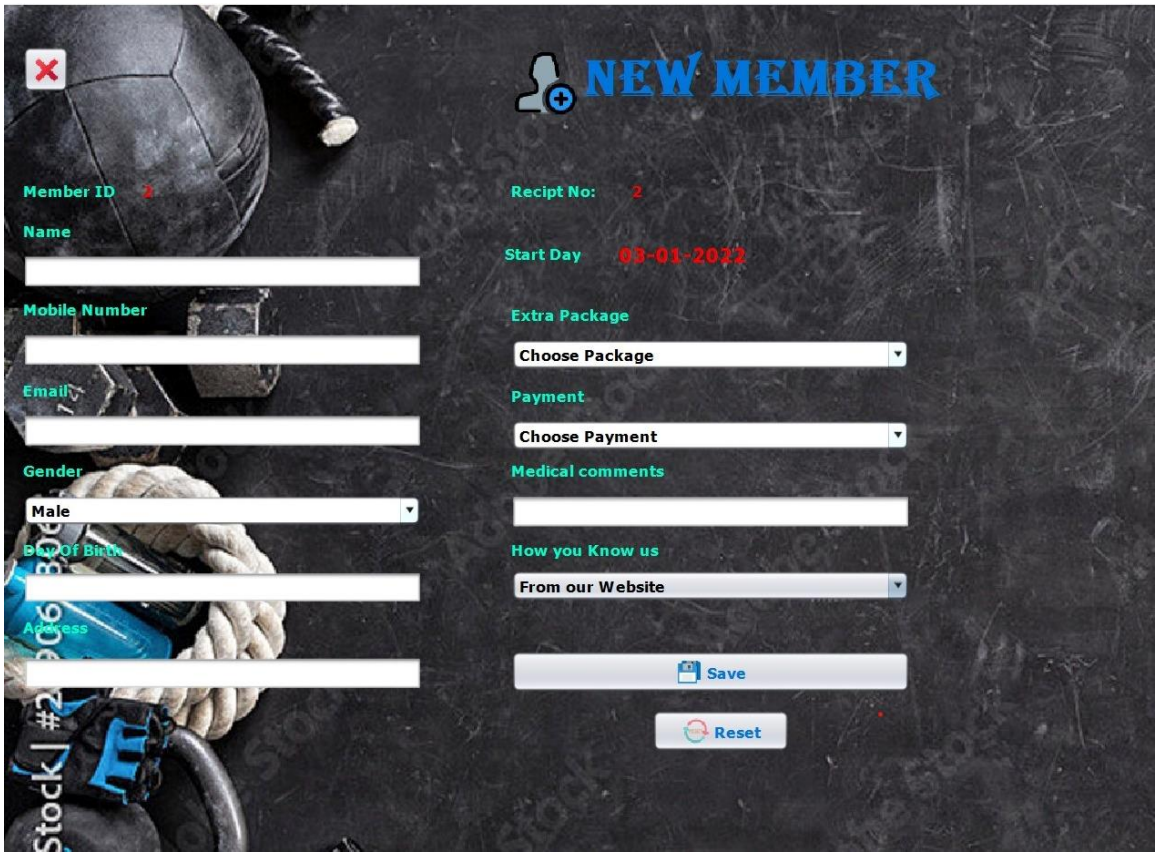


-Type of account to be created or updated: -

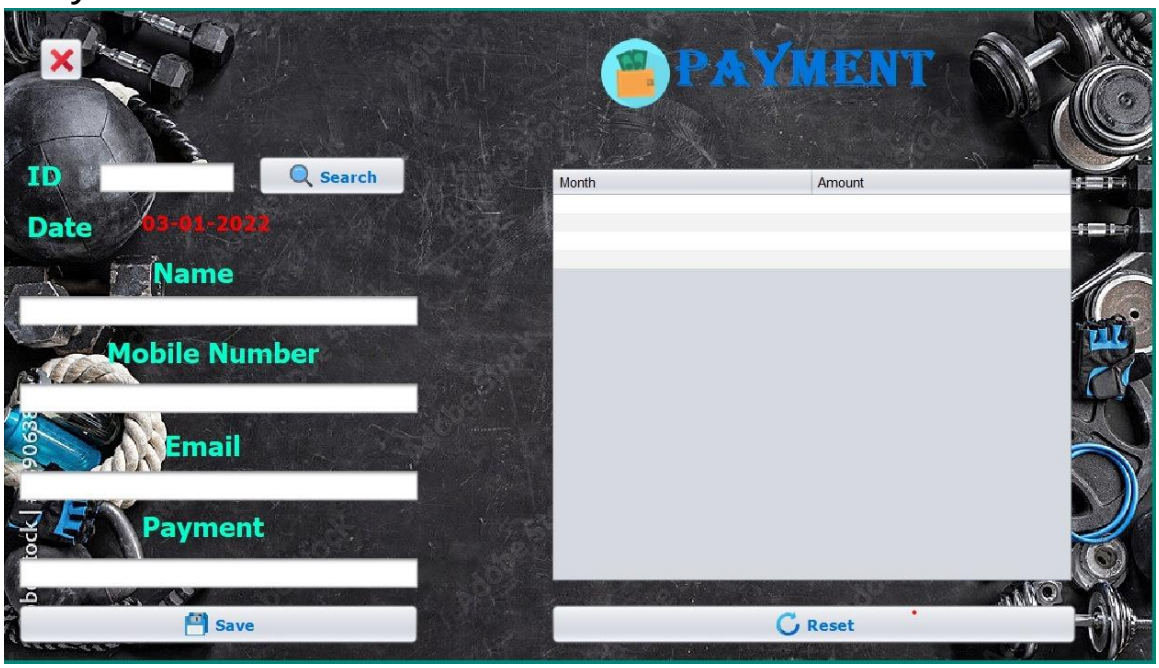
-List of the program functions: -



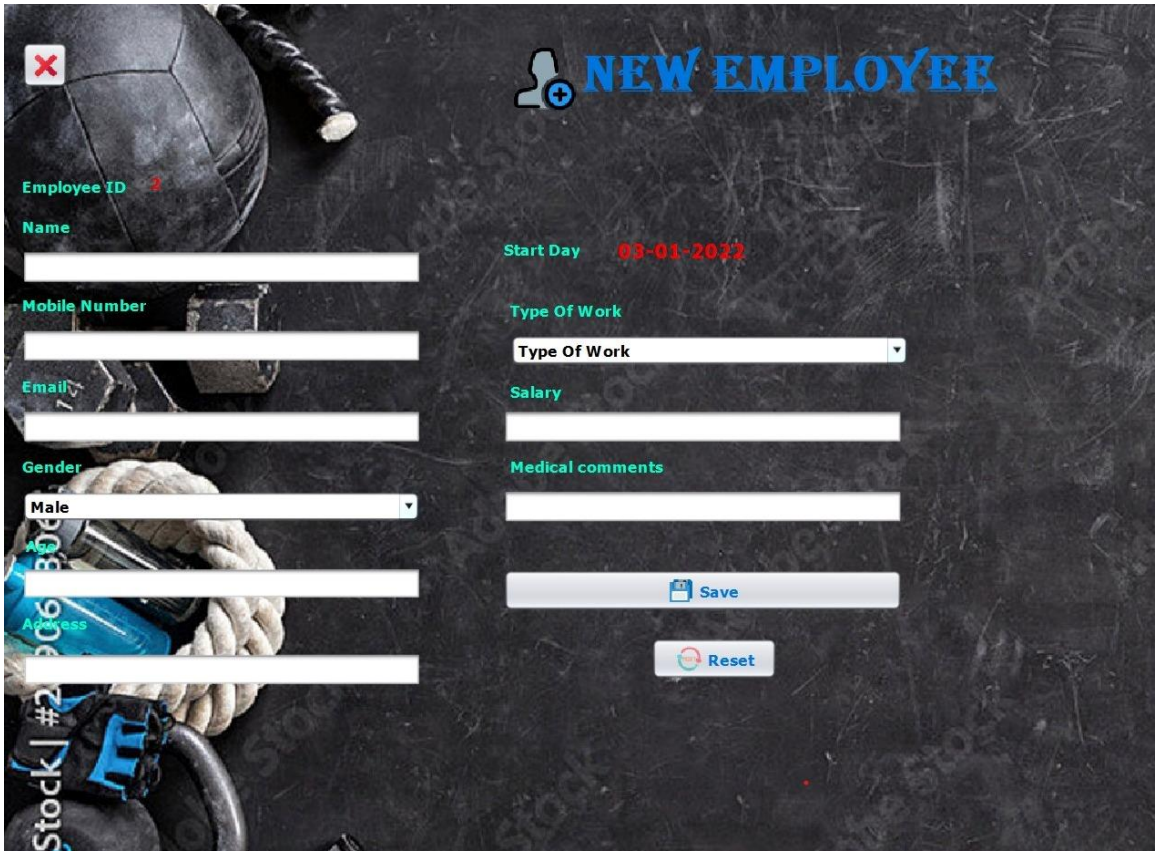
-New member creation screen: -



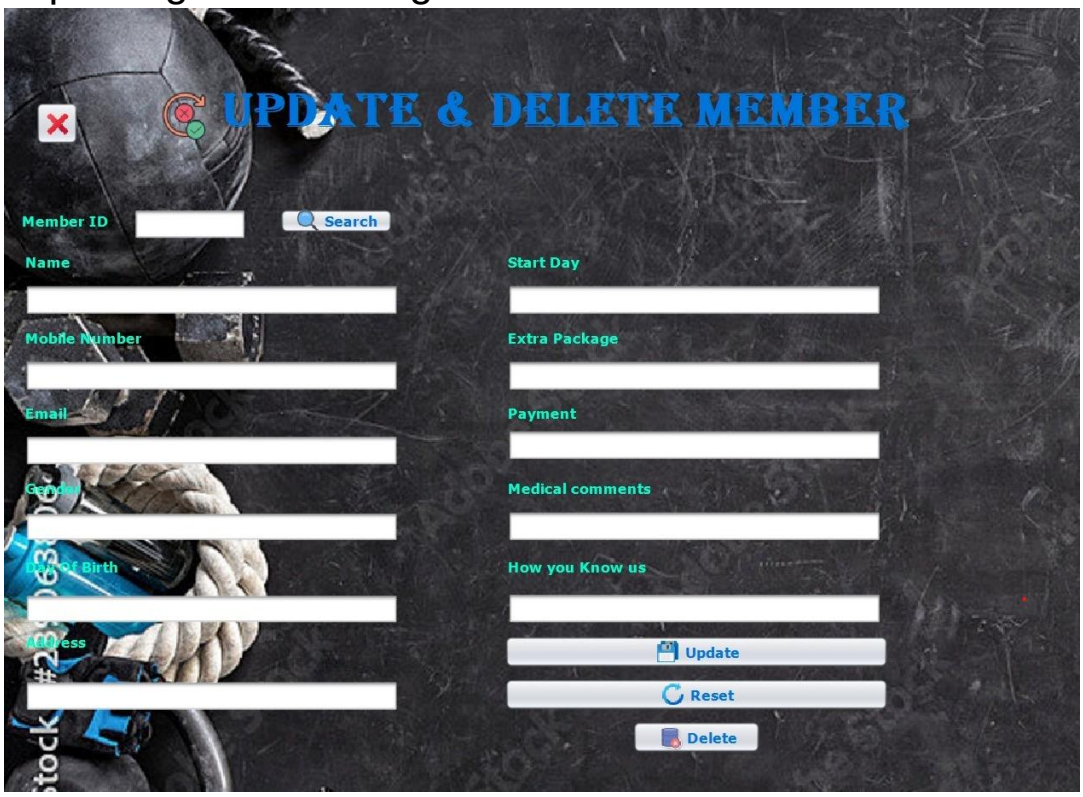
-Payment screen: -



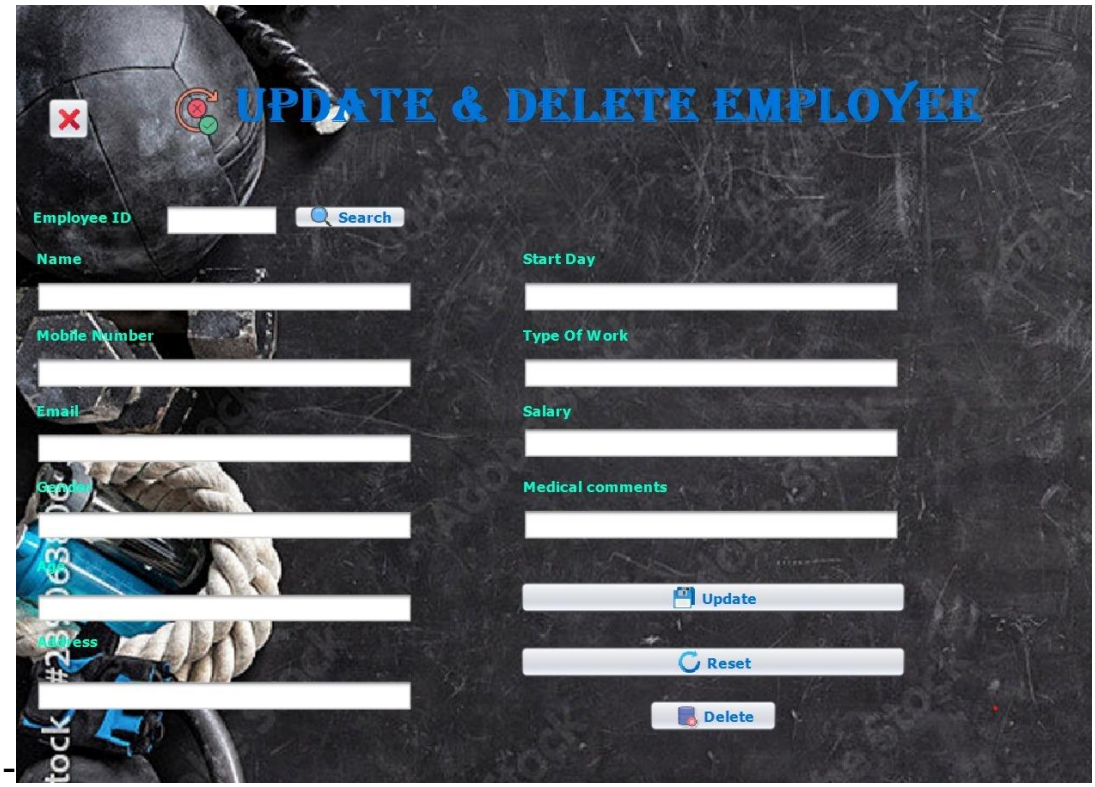
-New employee screen: -



-Updating and deleting members screen: -



- Updating and deleting employees screen: -



-

-List of members: -

-List of employees: -

-Salary screen: -