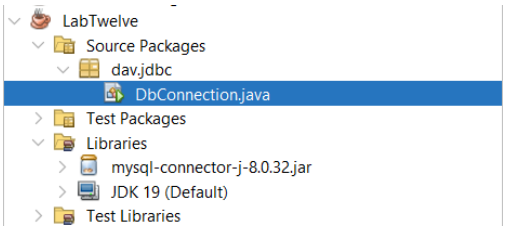
Appendix:




```java
package dav.jdbc;
import java.sql.*;
public class DbConnection {
    public static final String DB_NAME = "lab_twelve";
    public static final String DB_HOST = "localhost";
    public static final String DB_USER = "root";
    public static final String DB_PASSWORD = "";
    public static final int PORT = 3306;
    public static final String URL = "jdbc:mysql://" + DB_HOST + ":" + PORT + "/" + DB_NAME;
    public static Connection con;
    public static Statement stmt;

    public static void doConnection() {
        try {
            Class.forName(className:"com.mysql.cj.jdbc.Driver");
            con = DriverManager.getConnection(url: URL, user:DB_USER, password: DB_PASSWORD);
            if (con != null) {
                System.out.println(x: "Connection Success");
            } else {
                System.out.println(x: "Connection failed");
            }
        } catch (SQLException | ClassNotFoundException e) {
            System.out.println("Error: " + e.getMessage());
        }
    }
    public static void insertData(String first_name, String last_name, String username, String password) {
        try {
            String sql = "INSERT INTO lab_12_user(first_name,last_name,username,password) VALUES ('"+ first_name +"','"
                + last_name +"','"+ username + "','" + password + "')";
                stmt = con.createStatement();
                int status  = stmt.executeUpdate(string:sql);
                if(status  == 1){
                    System.out.println(x: "data inserted");
                }else{
                    System.out.println(x: "Data not inserted");
                }
        } catch (SQLException e) {
            System.out.println("Error: " + e.getMessage());
        }
    }
public static void updateData(String first_name, String last_name, String username, String password, int id) {
    try {
        String sql = "UPDATE lab_12_user SET first_name='"+first_name+"', last_name='"+last_name+"',"
                + " username='"+username+"', password='"+password+"' WHERE id='"+id+"'";
        stmt = con.createStatement();
        int status  = stmt.executeUpdate(string:sql);
        if(status  == 1){
            System.out.println(x: "data inserted");
        }else{
            System.out.println(x: "Data not inserted");
        }
    } catch (SQLException e) {
        System.out.println("Error: " + e.getMessage());
    }
}

public static void getUserList(){
    try{
        String sql="SELECT * FROM lab_12_user";
        stmt=con.createStatement();
        ResultSet rs = stmt.executeQuery(string:sql);
        while(rs.next()){
            System.out.println("First Name: "+rs.getString(string:"first_name"));
            System.out.println("Last Name: "+rs.getString(string:"last_name"));
            System.out.println("Username: "+rs.getString(string:"username"));
            System.out.println("Password: "+rs.getString(string:"password"));
        }
    }catch(SQLException e){
        System.out.println("Error:"+e.getMessage());
    }
}
```

```
72      public static void main(String[] args) {
73          doConnection();
74          insertData(first_name: "md",last_name:"khan",username: "noor",password: "noor123");
75          updateData(first_name: "Noorullah",last_name:"Khan",username: "meow",password: "meow123",id: 1);
76          getUserList();
77      }
78  }
```

Output:

doConnection():

```
Output - LabTwelve (run)
run:
Connection Success
BUILD SUCCESSFUL (total time: 0 seconds)
```

insertData():

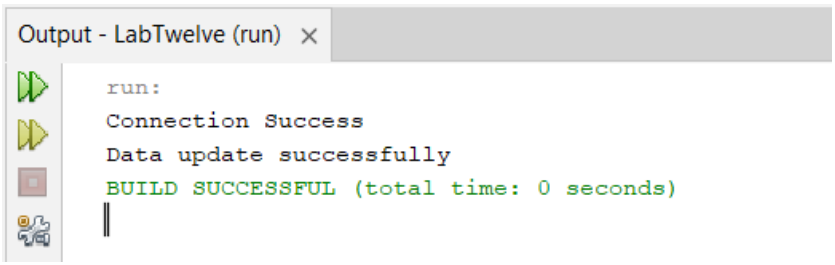
```
    public static void main(String[] args) {
        doConnection();
        insertData(first_name: "md",last_name:"khan",username: "noor",password: "noor123");
//          updateData("Noorullah","Khan","meow","meow123",1);
//          getUserList();
    }
}
```

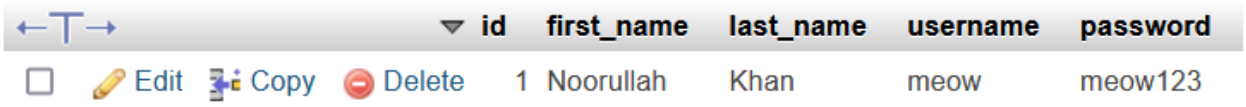
```
Output - LabTwelve (run)
run:
Connection Success
data inserted
BUILD SUCCESSFUL (total time: 0 seconds)
```

	id	first_name	last_name	username	password
Edit Copy Delete	1	md	khan	noor	noor123

updateData():

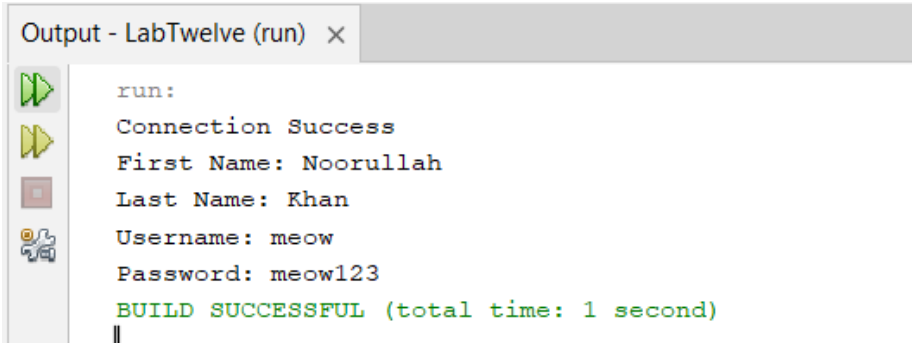
```
    public static void main(String[] args) {
        doConnection();
//          insertData("md","khan","noor","noor123");
        updateData(first_name: "Noorullah",last_name:"Khan",username: "meow",password: "meow123",id: 1);
//          getUserList();
    }
}
```



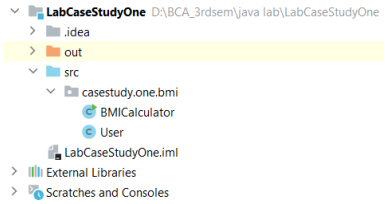


getUserList():

```
    public static void main(String[] args) {
        doConnection();
//          insertData("md","khan","noor","noor123");
//          updateData("Noorullah","Khan","meow","meow123",1);
        getUserList();
    }
}
```



Appendix:




```java
package casestudy.one.bmi;
import java.util.Scanner;
2 usages
public class BMICalculator {
    9 usages
    User user;
    1 usage
    public BMICalculator() { this.user= new User(); }
    1 usage
    public double calculateBMI(){
        Scanner sc = new Scanner(System.in);
        System.out.println("Enter your Weight in KG: ");
        try{
            this.user.setWeight(sc.nextDouble());
        }catch (Exception e){
            System.out.println("Please enter valid input");
        }
        System.out.println("Enter your height in Meter: ");
        try{
            this.user.setHeight(sc.nextDouble());
        }
        catch (Exception e){
            System.out.println("Please enter valid input");
        }
        double bmi= this.user.getWeight()/Math.pow(this.user.getHeight(),2);
        return bmi;
    }
    2 usages
    public  String showIndex(double bmi){
        String index=null;
        if(bmi<0&&bmi>100){
            index="please insert proper data: Height in meter and weight in kg";
        }else if(bmi>0 && bmi<=16){
            index = "You're severly underweight";
        } else if (bmi>16 && bmi <=18.4){
            index = "You are underweight";
        } else if (bmi>18.4 && bmi<=24.9) {
            index = "You are Normal";
        }else if (bmi>24.9 && bmi<= 29.9){
            index ="You are overweight";
        }else if (bmi>29.9 && bmi<= 34.9){
            index ="You are moderately obese";
        }else if (bmi>34.9 && bmi<= 39.9){
            index ="You are severely obese";
        }else if (bmi>39.9){
            index ="You are Morbidly obese";
        }
        return index;
    }
    1 usage
    public void calculate(String flag){
        switch (flag.toLowerCase()){
            case "us":
                System.out.println("Your index: " + this.showIndex(this.calculateUsBmi()));
                break;
            case "metric":
                System.out.println("Your index: " + this.showIndex(this.calculateBMI()));
                break;
```

```
            default:
                System.out.println("Enter either 'us' or 'metric'");}}
    1 usage
    public double calculateUsBmi(){
        Scanner sc = new Scanner(System.in);
        System.out.println("Enter your Weight in LBS: ");
        try{
            this.user.setWeight(sc.nextDouble());
        }catch (Exception e){
            System.out.println("Please enter valid input");}
        System.out.println("Enter your height in Inches: ");
        try{
            this.user.setHeight(sc.nextDouble());}
        catch (Exception e){
            System.out.println("Please enter valid input");}
        double bmi= this.user.getWeight()/(Math.pow(this.user.getHeight(),2)*703);
        return bmi;
    }
    no usages
    public static void main(String[] args) {
        BMICalculator s = new BMICalculator();
        Scanner sc = new Scanner(System.in);
        System.out.println("Enter the unit 'us' or 'metric' ");
        s.calculate(sc.next());
    }
}
```

User.java:

```
package casestudy.one.bmi;
import java.util.ArrayList;
2 usages
public class User{
    2 usages
    private String name;
    2 usages
    private String dob;
    2 usages
    private String blood_group;
    2 usages
    private String email;
    2 usages
    private int age;
    2 usages
    private double weight;
    2 usages
    private double height;
    no usages
    public String getName(){return this.name;}
    no usages
    public void setName(String name){this.name=name;}
    no usages
    public String getDob(){return this.dob;}
    no usages
    public  void setDob(String dob){this.dob=dob;}
    no usages
    public String getBlood_group() {return this.blood_group;}
    no usages
    public void setBlood_group(String blood_group){this.blood_group=blood_group;}
    no usages
    public String getEmail() {return this.email;}
    no usages
    public void setEmail(String email){this.email=email;}
    no usages
    public int getAge(){return this.age;}
    no usages
    public void setAge(int age) {this.age = age;}
    2 usages
    public double getWeight(){return this.weight;}
```

```
    2 usages
22      public void setWeight(double weight) {this.weight = weight;}
    2 usages
23      public double getHeight() {return height;}
    2 usages
24      public void setHeight(double height) {this.height = height;}
25  }
```

Output: