

Bill.java

```
package Assign;

import java.util.Date;

public class Bill {

    private int billId;
    private int customerId;
    private Date billDate;
    private String billDescription;
    private double billAmount;

    //Default constructor
    public Bill() {}

    //Parameterized constructor
    public Bill(int billId, int customerId, Date billDate, String billDescription,
double billAmount) {
        super();
        this.billId = billId;
        this.customerId = customerId;
        this.billDate = billDate;
        this.billDescription = billDescription;
        this.billAmount = billAmount;
    }

    //Creating getter-setters
    public int getBillId() {
        return billId;
    }

    public void setBillId(int billId) {
        this.billId = billId;
    }

    public int getCustomerId() {
        return customerId;
    }

    public void setCustomerId(int customerId) {
        this.customerId = customerId;
    }

    public Date getBillDate() {
        return billDate;
    }

    public void setBillDate(Date billDate) {
        this.billDate = billDate;
    }

    public String getBillDescription() {
        return billDescription;
    }

    public void setBillDescription(String billDescription) {
        this.billDescription = billDescription;
    }

    public double getBillAmount() {
        return billAmount;
    }

    public void setBillAmount(double billAmount) {
        this.billAmount = billAmount;
    }
}
```

```

    }

    //Creating toString()
    @Override
    public String toString() {
        return "Bill [billId=" + billId + ", customerId=" + customerId + ", billDate=" +
billDate + ", billDescription="
        + billDescription + ", billAmount=" + billAmount + "];"
    }
}

```

IBillOperation.java

```

package Assign;

import java.sql.SQLException;
import java.text.ParseException;
import java.util.Date;
import java.util.List;

public interface IBillOperation {

    //Method 1
    public int saveBillRecord(int customer_id,
        Date bill_date,
        String
        bill_description,
        double bill_amount) throws SQLException, ParseException,
ClassNotFoundException;

    //Method 2
    public int editBillRecord(int billid, int customer_id,String bill_description,
        double bill_amount) throws ClassNotFoundException, SQLException;

    //Method 3
    public int removeBillRecord(int bill_id) throws ClassNotFoundException,
SQLException;

    //Method 4
    public List<Bill> getAllBillRecord() throws ClassNotFoundException, SQLException;
}

```

BillOperationImpl.java

```
package Assign;

import java.sql.SQLException;
import java.sql.Statement;
import java.sql.Timestamp;
import java.text.ParseException;
import java.text.SimpleDateFormat;
import java.util.ArrayList;
import java.util.Date;
import java.sql.Connection;
import java.sql.ResultSet;
import java.util.List;
import java.util.Scanner;

public class BillOperationImpl implements IBillOperation{

    DBConnection dbc =new DBConnection();

    Scanner sc= new Scanner(System.in);

    List<Bill>billList=new ArrayList<>();
    int i;

    //Method 1 implementation
    @Override
    public int saveBillRecord(int customer_id, Date bill_date, String bill_description,
        double bill_amount) throws SQLException, ParseException,
        ClassNotFoundException{

        Connection con=dbc.getConnection();
        Statement st=con.createStatement();
        String sql="insert into bill(customerid,billdate,billdescription,billamount) "
            +
            "values (" +customer_id+", '"+bill_date+"', '"+bill_description+"', "+bill_amount+"));

        //DML(Data Manipulation Language)
        int n=st.executeUpdate(sql);
        if(n>=0)
            System.out.println(n+" record(s) affected");

        return n;
    }

    //Method 2 implementation
    @Override
    public int editBillRecord(int billid, int customer_id, String bill_description,
        double bill_amount) throws ClassNotFoundException, SQLException {
        Connection con=dbc.getConnection();
        Statement st=con.createStatement();

        String sql = "update bill set customerId=(" + customer_id + "),"
            + "billDescription = ('" + bill_description + "'),"
            + "billAmount = (" + bill_amount
            + ") where billId=(" + billid + ")";

        //DML(Data Manipulation Language)
        int n=st.executeUpdate(sql);
        if(n>=0)
            System.out.println(n+" record(s) edited");

        return n;
    }
}
```

```

    }

    //Method 3 implementation
    @Override
    public int removeBillRecord(int billid) throws ClassNotFoundException, SQLException
    {
        Connection con=dbc.getConnection();
        Statement st=con.createStatement();

        String sql = "delete from bill where billId="+billid+"";

        //DML(Data Manipulation Language)
        int n=st.executeUpdate(sql);
        if(n>=0)
            System.out.println(n+" record(s) deleted");

        return n;
    }

    //Method 4 implementation
    @Override
    public List<Bill> getAllBillRecord() throws ClassNotFoundException, SQLException {
        Connection con=dbc.getConnection();
        Statement st=con.createStatement();

        String sql="select * from bill";
        ResultSet billSet=st.executeQuery(sql);
        while(billSet.next()) {
            System.out.println(billSet.getString(1)+" "+billSet.getString(2)+"
"+billSet.getString(3)+" "+billSet.getString(4)+" "+billSet.getString(5));
        }

        return (List<Bill>) billSet;
    }
}

```

BillOperationMain.java

```

package Assign;

import java.sql.SQLException;
import java.text.ParseException;
import java.text.SimpleDateFormat;
import java.util.Date;
import java.util.Scanner;

public class BillOperationMain {

    public static void main(String[] args) throws ClassNotFoundException,
    SQLException, ParseException {
        // TODO Auto-generated method stub
        //taking data from user
        Scanner sc = new Scanner(System.in);

        java.util.Date sqlDate;
        java.util.Date utilDate=null;

        BillOperationImpl impl = new BillOperationImpl();
    }
}

```

```

DBConnection dbc= new DBConnection();
dbc.getConnection();
int choice=0;

System.out.println("Enter you choice:\n1-Add\n2-Edit\n3-Delete\n4-Show");
choice = sc.nextInt();

switch(choice) {
case 1:

    System.out.println("Enter customer id: ");
    int customer_id= sc.nextInt();

    System.out.println("Enter date: ");
    SimpleDateFormat sdf = new SimpleDateFormat("dd-MM-yyyy");
    String bill_date=sc.next();
    utilDate =sdf.parse(bill_date);
    sqlDate= new java.sql.Date(utilDate.getTime());

    System.out.println("Enter bill description: ");
    String bill_description= sc.next();

    System.out.println("Enter bill amount: ");
    Double bill_amount= sc.nextDouble();

    impl.saveBillRecord(customer_id, sqlDate, bill_description,
    bill_amount);
    break;

case 2:
    System.out.println("Enter bill id: ");
    int billid= sc.nextInt();

    System.out.println("Enter customer id: ");
    int customerid= sc.nextInt();

    System.out.println("Enter bill description: ");
    String billdescription= sc.next();

    System.out.println("Enter bill amount: ");
    Double billamount= sc.nextDouble();

    impl.editBillRecord(billid, customerid, billdescription,billamount);
    break;

case 3:
    System.out.println("Enter bill_id: ");
    int b_id= sc.nextInt();
    impl.removeBillRecord(b_id);
    break;

case 4:

    impl.getAllBillRecord();
    break;

default:
    System.out.println("Invalid choice!");
}
}
}

```

DBConnection.java

```
package Assign;

import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.SQLException;
import java.sql.Statement;

public class DBConnection {

    public static java.sql.Connection getConnection() throws ClassNotFoundException,
SQLException{

        String driver="com.mysql.cj.jdbc.Driver";
        String dburl="jdbc:mysql://localhost:3306/billpayment";
        String user="root";
        String password="123456789";
        Connection con=null;
        Statement st= null;

        //load the driver
        Class.forName(driver);

        //create the connection
        con=DriverManager.getConnection(dburl,user,password);

        if(con!=null)
            System.out.println("Connection successful");
        else
            System.out.println("Connection failed");

        //write and execute query
        st= con.createStatement();

        return con;
    }
}
```

BillTest.java

```
package Assign;

import static org.junit.jupiter.api.Assertions.*;
import org.junit.jupiter.api.Test;

class BillTest {

    Bill b= new Bill();

    void testGetBillId() {

        assertEquals(0,b.getBillId());

    }

    @Test
```

```

        void testGetCustomerId() {
            assertEquals(0,b.getCustomerId());
        }

        @Test
        void testGetBillDate() {
            assertEquals(null,b.getBillDate());
        }

        @Test
        void testGetBillDescription() {
            assertEquals(null,b.getBillDescription());
        }

        @Test
        void testGetBillAmount() {
            assertEquals(0, b.getBillAmount());
        }
    }
}

```

BillOperationImplTest.java

```

package Assign;

import static org.junit.jupiter.api.Assertions.*;
import java.sql.SQLException;
import java.text.ParseException;
import org.junit.jupiter.api.Test;

class BillOperationImplTest {

    BillOperationImpl bl=new BillOperationImpl();

    @Test
    void testSaveBillRecord() throws ClassNotFoundException, SQLException,
    ParseException {
        // fail("Not yet implemented");
        assertEquals(0, bl.saveBillRecord(0, null, null, 0));
    }

    @Test
    void testEditBillRecord() throws ClassNotFoundException, SQLException {
        assertEquals(0, bl.editBillRecord(0, 0, null, 0));
    }

    @Test
    void testGetRemoveBillRecord() throws ClassNotFoundException, SQLException {
        assertEquals(0, bl.removeBillRecord(0));
    }
}

```

```

    }

    @Test
    void testGetAllBillRecord() throws ClassNotFoundException, SQLException {
        assertEquals(null, bl.getAllBillRecord());
    }
}

```

SQL Query

```

create database billpayment;

use billpayment;

create table customer(
customerid int primary key not null auto_increment,
customername varchar(100),
contactno varchar(100)
);

insert into customer (customerid,customername,contactno) values
(101,'Tom',768945378), (102,'John',7689453879),
(103,'Michael',898945378), (404,'Mick',698945378);

desc customer;

select*from customer;


create table bill(
billid int primary key not null auto_increment,
customerid int,
billdate date,
billdescription varchar(100),
billamount double,
FOREIGN KEY(customerid) References customer(customerid)
);

desc bill;

select*from bill;

```