

-----BILL.JAVA-----

```
package week;

import java.sql.Timestamp;
import java.util.Date;

public class Bill {

    private int billid;
    private int consumerid;
    private Date billDate;
    private String billdescription;
    private double billAmount;

    Bill(){} // Default Constructor

    // Parameterized constructor

    Bill(int billid,int consumerid,Date billDate,String
billdescription, double billAmount){
        this.billid=billid;
        this.consumerid=consumerid;
        this.billDate=billDate;
        this.billdescription=billdescription;
        this.billAmount=billAmount;
    }

    //Getters and Setters

    public int getBillid() {
        return billid;
    }
    public void setBillid(int billid) {
        this.billid = billid;
    }
    public int getConsumerid() {
        return consumerid;
    }
    public void setConsumerid(int consumerid) {
        this.consumerid = consumerid;
    }
    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 billamount2) {
        this.billAmount = billamount2;
    }

    // to string

    @Override
    public String toString() {
        return "Bill [billid=%s\t consumerID=%s\t Date=%s\t\nDescription=%s\t Amount=%s]";
    }

}

```

```

-----BILL OPERATION-----

```

```

package week;

```

```

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

```

```

public interface BillOperation {

```

```

        int saveBillRecord(int consumerid,Date billdate,String
billdescription,double billamount);

        int editBillRecord(int bill, int consumerId, Date billdate,
String billDescription, double billamount);

        int removeBillRecord(int bil);

        List<Bill>getAllBillRecord();

        Bill getBillRecordById(int bil);

}

```

```

-----BILLOPERATIONIMPL-----
-----

```

```

package week;

import java.security.Timestamp;
import java.text.ParseException;
import java.util.Date;
import java.util.List;

public class BillOperationImpl implements BillOperation {
    Bill[]bill=new Bill [100];
    static int index;

    @Override
    public int saveBillRecord(int consumerid,Date
billdate,String billdescription,double billamount) {

```

```

        for(int i=0;i<index;i++) {

            bill[i].setConsumerid(consumerid);
            bill[i].setBillDate((java.sql.Timestamp)
billdate);

            bill[i].setBilldescription(billdescription);
            bill[i].setBillAmount(billamount);
            break;

        }

```

```

        return 0;
    }

```

```

@Override
    public int editBillRecord(int bil,int consumerId,Date
billdate,String billDescription,double billamount) {
        // TODO Auto-generated method stub
        for(int i=0;i<index;i++) {
            if(bill[i].getBillid()==bil) {
                bill[i].setConsumerid(consumerId);
                bill[i].setBillDate((java.sql.Timestamp)
billdate);

                bill[i].setBilldescription(billDescription);
                bill[i].setBillAmount(billamount);

                break;
            }
        }
        return 0;
    }

```

```

@Override
    public int removeBillRecord(int bil) {
        // TODO Auto-generated method stub
        for (int i=0;i<index;i++) {
            if(bill[i].getBillid()==bil) {
                bill[i].setConsumerid(-1);
                bill[i].setBillDate(null);
                bill[i].setBilldescription(null);
                bill[i].setBillAmount(-1);
                // bill[i].setBill(-1);

            }

```

```

        else
            System.out.println("Employee id not found");
    }
    return 0;
}

@Override
public List<Bill> getAllBillRecord() {
    // TODO Auto-generated method stub
    return null;
}

@Override
public Bill getBillRecordById(int bil) {
    // TODO Auto-generated method stub
    for (int i=0;i<index;i++) {
        if(bill[i].getBillid()==bil) {

            System.out.println(bill[i]);
        }
        else
            System.out.println("Employee id not
found");
    }
    return null;
}
}

```

 BillOpeartionMain-----

```

package week;

import java.sql.Connection;

```

```

import java.sql.DriverManager;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.sql.Statement;
import java.sql.Timestamp;
import java.text.ParseException;
import java.text.SimpleDateFormat;
import java.time.LocalDate;
import java.util.Date;
import java.util.Scanner;

public class BillOperationMain {

    public static void main(String[] args) throws
ParseException, SQLException {
        // TODO Auto-generated method stub

        BillOperationImpl bill=new BillOperationImpl();

        Scanner sc=new Scanner(System.in);
        int bil;
        int consumerId;
        Timestamp billdate;
        String billDescription;
        double billamount;
        do {
            try {
                Connection con = null;
                try {
                    con =
DBConnection.getConnection();
                } catch (Exception e) {
                    e.printStackTrace();
                }

                //Write and Execute query
                Statement st=con.createStatement();

                int ch;
                ResultSet a;
                System.out.println("\t1.save bill \t\t2.edit
bill\n\t3.remove bill\t\t4. gel all bil\n\t5 search\n");
                System.out.println("enter your choice : ");
                ch=sc.nextInt();

```

```

        switch(ch) {
case 1: //-----add-----

        System.out.println("Enter billid number :
");
        bil=sc.nextInt();

        System.out.println("Enter consumer id :
");
        consumerld=sc.nextInt();

        java.sql.Date sqldate;
        java.util.Date utildate;
        SimpleDateFormat sdf=new
SimpleDateFormat("dd/MM/yyyy");

        System.out.println("Enter date");
        String strdate= new
Scanner(System.in).nextLine();
        utildate=sdf.parse(strdate);
        sqldate=new java.sql.Date(utildate.getTime());

        System.out.println("Enter bill Description
: ");

        sc.nextLine();
        billDescription=sc.nextLine();

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

        String sql="insert into bill
values("+bil+", "+consumerld+", '"+sqldate+"', '"+billDescription+"
', "+billamount+"");

        bill.saveBillRecord( consumerld, sqldate,
billDescription, billamount);
        ch=st.executeUpdate(sql);
        System.out.println("\n * __Saved__ * \n");
        break;

case 2: //-----edit-----

```

```

        System.out.println("Enter bill id number :
");
        bil=sc.nextInt();

        System.out.println("Enter consumer id :
");
        consumerId=sc.nextInt();

        java.sql.Date sqldate1;
        java.util.Date utildate1;
        SimpleDateFormat sdf1=new
SimpleDateFormat("dd/MM/yyyy");

        System.out.println("Enter date");
        String strdate1= new
Scanner(System.in).nextLine();

        utildate1=sdf1.parse(strdate1);
        sqldate1=new
java.sql.Date(utildate1.getTime());

        sc.nextLine();
        System.out.println("Enter bill Description : ");
        billDescription=sc.nextLine();

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

        String sql2="update bill
consumerid=("+consumerId+"),billdate = ('"+sqldate1+"'),
billdescription = ('"+billDescription+"'),billamount =
("+billamount+") where billid=("+bil+")";
        ch=st.executeUpdate(sql2);
        System.out.println("\n * __Edited__ * \n");
        break;

    case 3:// -----remove-----
        System.out.println("Enter bill id number :
");

        bil=sc.nextInt();
        bill.removeBillRecord(bil);
        String sql1="delete from bill where
billid=("+bil+")";

        ch=st.executeUpdate(sql1);
        break;

```



```

---
        case 4: //-----show all-----

            bill.getAllBillRecord();
            String sqlq="select * from bill";
            ResultSet rs=st.executeQuery(sqlq);
            while(rs.next())
            {
                System.out.println(rs.getInt(1)+"
"+rs.getInt(2)+" "+rs.getDate(3)+" "+rs.getString(4)+"
"+rs.getDouble(5));
            }
            break;

        case 5: //-----search-----
            System.out.println("Enter id number : ");
            bil=sc.nextInt();
            bill.getBillRecordById(bil);
            String sql3="Select * from bill where
billid=("+bil+")";

            a = st.executeQuery(sql3);
            while(a.next()) {
                System.out.println(a.getInt(1)+"
"+a.getInt(2)+" "+a.getDate(3)+" "+a.getString(4)+"
"+a.getDouble(5));
            }
            break;

        }

    }

    catch(SQLException e) {
        System.out.println(e.getMessage());
    }
    while(true);

}
}

```

-----DBConnection-----

```
package week;

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

public class DBConnection {

    public static Connection getConnection() throws
    ClassNotFoundException, SQLException
    {

        String driver="com.mysql.cj.jdbc.Driver";
        String
dburl="jdbc:mysql://localhost:3306/billpayment";
        String user="root";
        String password="root";

        Class.forName(driver);

        //create the connection

        Connection con=
DriverManager.getConnection(dburl,user,password);

        return con;
    }
}
```

-----BillTest-----

```
package week;
```

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

```
import org.junit.jupiter.api.Test;
```

```
class BillTest {
```

```
    Bill a=new Bill();
```

```
    @Test
```

```
    void testGetBillid() {
```

```
        //fail("Not yet implemented");
```

```
        assertEquals(0,a.getBillid());
```

```
    }
```

```
    @Test
```

```
    void testGetConsumerid() {
```

```
        //fail("Not yet implemented");
```

```
        assertEquals(0,a.getConsumerid());
```

```
    }
```

```
    @Test
```

```
    void testGetBillDate() {
```

```
        //fail("Not yet implemented");
```

```

        assertEquals(null,a.getBillDate());
    }

    @Test
    void testGetBilldescription() {
        //fail("Not yet implemented");
        assertEquals(null,a.getBilldescription());
    }

    @Test
    void testGetBillAmount() {
        //fail("Not yet implemented");
        assertEquals(0,a.getBillAmount());
    }
}

```

-----BillImplTest-----

```

package week;

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

import org.junit.jupiter.api.Test;

```

```
class BillOperationImplTest {

    BillOperationImpl billImpl=new BillOperationImpl();

    @Test
    void testSaveBillRecord() {
        //fail("Not yet implemented");
        assertEquals(0,billImpl.saveBillRecord(0, null, null,
0));
    }

    @Test
    void testEditBillRecord() {
        //fail("Not yet implemented");
        assertEquals(0,billImpl.editBillRecord(0, 0, null,
null, 0));
    }

    @Test
    void testRemoveBillRecord() {
        //fail("Not yet implemented");
        assertEquals(0,billImpl.removeBillRecord(0));
    }

    @Test
    void testGetAllBillRecord() {
        //fail("Not yet implemented");
        assertEquals(null,billImpl.getAllBillRecord());
    }
}
```

```

@Test
void testGetBillRecordById() {
    //fail("Not yet implemented");
    assertEquals(null,billImpl.getBillRecordById(0));
}

}

```

-----Sql Query-----

```

create database billpayment;
use billpayment;
create table customer (
    consumerid int NOT NULL auto_increment,
    consumername varchar(100),
    contactno varchar(100),
    PRIMARY KEY (consumerid)
);

Insert into
customer(consumerid,consumername,contactno) values (1,'vimal',2345
60),(2,'amit',997147),(3,'pankaj',22334455);

```

```

create table bill(
    billid int NOT NULL auto_increment,
    consumerid int,
    PRIMARY KEY (billid),

```

```
FOREIGN KEY (consumerid )  
REFERENCES customer(consumerid),  
billdate datetime,  
billdescription varchar(100),  
billamount double  
);  
select*from customer;  
select*from bill;
```