Name=Prem Khatri

Week 4 Assignment

```
//-----------------------------------------------DataBase Part-----------------------------------------
//------------------Creating Database
create database billpayment;
use billpayment;
//------------------creating Customer Table
create table customer (
consumerid int  NOT NULL  auto_increment,
consumername varchar(100),
contactno varchar(100),
PRIMARY KEY (consumerid)
);

//-------------Creating Bill Table
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
);

//-----------------Displaying Table bill and customer
select * from bill;
select * from customer;
```

```
//--------------------Inserting Data into customer Table
insert into customer(consumerId,consumerName,contactNo) values (1,"Prem","8839584937");
insert into customer(consumerId,consumerName,contactNo) values (2,"Martin","9245262920");
insert into customer(consumerId,consumerName,contactNo) values (3,"Robert ","924764540");

//--------------------------_________ DataBase Part End_________________----------------------

//-------------------------------------------------Bill Class---------------------------------------
package week4Assignment;
import java.sql.Timestamp;
import java.util.Date;

public class Bill {
//-----------------------------------------------------Private Member of Bill class-------------------------
	private int billId;
	private int consumerId;
	private Date billDate;
	private String billDescription;
	private double billAmount;

	//---------------------------------------------Constructor-----------------------------------------
	public Bill() {}
	public Bill(int billId, int consumerId, Date billDate, String billDescription, double billAmount) {
		super();
		this.billId = billId;
		this.consumerId = consumerId;
		this.billDate = billDate;
		this.billDescription = billDescription;
		this.billAmount = billAmount;
	}
```

```
//-----------------------------------------Getters And setter-----------------------------------------
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 billAmount) {
```

```
		this.billAmount = billAmount;
	}

//-------------------------------------------To String-----------------------------------------------------------
	@Override
	public String toString() {
		return "Bill [billId=" + billId + ", consumerId=" + consumerId + ", billDate=" + billDate
		+ ", billDescription="+ billDescription + ", billAmount=" + billAmount + "]";
	}	}
//-------------------------_____________________Bill Class End_____________________________----------------------



//---------------------------------- _____________IBillOperation_______________-------------------------------

package week4Assignment;
import java.sql.Timestamp;
import java.util.Date;
import java.util.List;

//Creating Interface and implement it into IBillOperationImpl
public interface IBillOperation {
	int saveBillRecord(Bill b);
	int editBillRecord(int billId,int consumerId,Date billDate,String billDescription,double
	billAmount);
	int removeBillRecord(int bill);
	List<Bill> getAllBillRecord();
	Bill getBillRecordById(int bill);
}

//-------------------------------------------End IBillOperation-----------------------------------------------------
```

//----------IBillOperationImpl----------

```
package week4Assignment;

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

//Implementing IBillOperation

public class BillOperationImpl implements IBillOperation{

	Bill[]bill=new Bill[100];
	static int index;

	@Override
	public int saveBillRecord(Bill b) {
		bill[index]=b;
		index++;
		System.out.println("Employee has been Added:");
		return 0;
	}

	@Override
		public int editBillRecord(int billId,int consumerId,Date
	billDate,String billDescription,double billAmount) {
		boolean edited=false;
	  for(int i=0;i<index;i++) {
		if(bill[i].getBillId()==billId) {
			bill[i].setConsumerId(consumerId);
			bill[i].setBillDate(billDate);
			bill[i].setBillDescription(billDescription);
			bill[i].setBillAmount(billAmount);
			edited=true;
			break;
		}
	  }

		return 0;
	}

	@Override
	public int removeBillRecord(int billl) {
		for(int i=0;i<index;i++) {
		if(bill[i].getBillId()==billl) {
			  bill[i].setConsumerId(-1);
			bill[i].setBillDate(null);
			bill[i].setBillDescription(null);
			bill[i].setBillAmount(-1);
		}
	  else {
		System.out.println("Bill id not found");
	  }
	}
		return billl;
	}

	@Override
	public List<Bill> getAllBillRecord() {
		for(int i=0;i<index;i++)
```

```
            {
                  System.out.println(bill[i]);
            }
            return null;
      }

      @Override
      public Bill getBillRecordById(int billl) {
            for (int i=0;i<index;i++) {
            if(bill[i].getBillId()==billl) {

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

}

//---------------------------End IBillOperationImpl-----------------------


//----------------------------BillOperationMain----------------------------
```

```
package week4Assignment;


import java.sql.Connection;

import java.sql.DriverManager;

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.util.Date;

import java.util.Scanner;


public class BillOperationMain {

        public static void main(String[] args) throws ParseException {

        BillOperationImpl billImpl=new BillOperationImpl();

        Scanner sc=new Scanner(System.in);
```

```
		int billId;
		int consumerId;
		Date billDate;
		String billDescription;
		double billAmount;
		ResultSet a;

		 SimpleDateFormat sdf=new SimpleDateFormat("yyyy/MM/dd");
		do {
			try {
				Connection con = null;
				try {
					con = DBConnection.getConnection();
				} catch (Exception e) {
					e.printStackTrace();
				}

//Write and Execute query
	Statement st=con.createStatement();
	int ch;
	System.out.println("Select the operation to perform:");
	System.out.println("1.Save \n2.Edit\n3.Delete\n4.Fetch All\n5.Fetch");
	System.out.println("enter your choice:");
	ch=sc.nextInt();
	Date utildate;
	java.sql.Date sqlDate;
	String s;
	switch(ch) {
	case 1:
		System.out.println("Enter BillId id: ");
		billId=sc.nextInt();
```

```
                System.out.println("Enter Consumer id : ");
                consumerId=sc.nextInt();
                System.out.println("Enter bill Date : ");
                s=new Scanner(System.in).nextLine();
                utildate=sdf.parse(s);
                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  sql2="insert into bill values
        ("+billId+","+consumerId+",'"+sqlDate+"','"+billDescription+"',"+billAmount+")";
Bill b1=new Bill(billId,consumerId,sqlDate,billDescription,billAmount);
billImpl.saveBillRecord(b1);
ch=st.executeUpdate(sql2);
System.out.println("\n *__Inserted__* \n");
break;

case 2:// Edit
        System.out.println("Enter the Employee id which u want to edit:");
        System.out.println("Enter Bill id: ");
        billId=sc.nextInt();

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

```
System.out.println("Enter Bill Date : ");
s=new Scanner(System.in).nextLine();
utildate=sdf.parse(s);
 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 sql3="update bill set billId=("+billId+"),billDate=('"+sqlDate+"')
                billDescription=('"+billDescription+"'),billAmount=("+billAmount+")
                 where billId=("+billId+")";
billImpl.editBillRecord(billId, consumerId, sqlDate, billDescription, billAmount);
 ch= st.executeUpdate(sql3);
 System.out.println("...Edited...");
break;

case 3:                 //delete
    System.out.println("Enter id number : ");
    billId=sc.nextInt();
    billImpl.removeBillRecord(billId);
    String sql1="delete from bill where billId=("+billId+")";
    ch=st.executeUpdate(sql1);
    System.out.println("\n *__Delete succesfull__* \n");
    break;

case 4:             //FetchAll
                 billImpl.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:   //Fetch
                    System.out.println("Enter BillId number : ");
                    billId=sc.nextInt();
                    billImpl.getBillRecordById(billId);
                    String sql4="Select * from bill where billId=("+billId+")";
                    a = st.executeQuery(sql4);
                    while(a.next()) {
                   System.out.println(a.getInt(1)+"  "+a.getInt(2)+"  "+a.getDate(3)+"  "+a.getString(4)+"
                    "+a.getDouble(5));              }
                    break;

                  }

               }
               catch(SQLException e1)
               {
                  System.out.println(e1.getMessage());
               }

                   }while(true);

}
}
//-------------------------------------------------End BillOperationMain-----------------------------------------------
```

//----------Class DBConnection----------

```java
package week4Assignment;

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="Bootcamp@48";

		Class.forName(driver);

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

}
```

//----------End DBConnection----------

//----------BillTest----------

```java
package week4Assignment;

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

import org.junit.jupiter.api.Test;

//Testing the Bill class
class BillTest {
 Bill b=new Bill();
	@Test
	void testGetBillId() {
		assertEquals(0,b.getBillId());
	}

	@Test
	void testGetConsumerId() {
		assertEquals(0,b.getConsumerId());
	}

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

	@Test
	void testGetBillDescription() {
```

```
			assertEquals(null,b.getBillDescription());
		}

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

}

//--------------------------End Bill Test----------------------------------


//--------------------------BillOperationMainTest-------------------------

package week4Assignment;

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

import org.junit.jupiter.api.Test;

//Testing the BillOperationImpl class
class BillOperationMainTest {
		BillOperationImpl billImpl=new BillOperationImpl();
		@Test
		void testSaveBillRecord() {
			assertEquals(0,billImpl.saveBillRecord(null));
		}

		@Test
		void testEditBillRecord() {
			assertEquals(0,billImpl.editBillRecord(0, 0, null, null, 0));
		}

		@Test
		void testRemoveBillRecord() {
		assertEquals(0,billImpl.removeBillRecord(0));
		}

		@Test
		void testGetAllBillRecord() {
			assertEquals(null,billImpl.getAllBillRecord());
		}

		@Test
		void testGetBillRecordById() {
			assertEquals(null,billImpl.getBillRecordById(0));
		}

}


//----------------------End BillOperationMainTest-----------------------
```