Electricity Billing System

Lecturer

ANAND SIR

RAJATHI MA'AM

The team

ARUN KUMAR SAH

CH.EN.U4CSE21266

ADARSH KARTHIKEYAN

CH.EN.U4CSE21201

ANIRUDDHA MUKHERJEE

CH.EN.U4CSE21233

BINOD MANDAL

CH.U4CSE21208

Abstract

The purpose of electricity billing system is to automate the already existing manual system by the help of computerized equipments and and full-fledged computer software, fulfilling their requirement so that their valuable data can be stored for a longer period with easy accessing and manipulation of the same

Abstract

Electricity Billing System, as described above can lead to error free, secure, reliable and fast management system. It can assist the user to concentrate on their other activities rather to concentrate on record keeping. Thus, it will help the organization in better utilization of resources. the organization can maintain computerized records without redundant entries. thus, one need not be distracted by the information that is not relevant, while being able to reach the information.

Electric Billing System

Manager

Customer

Billing and Payment

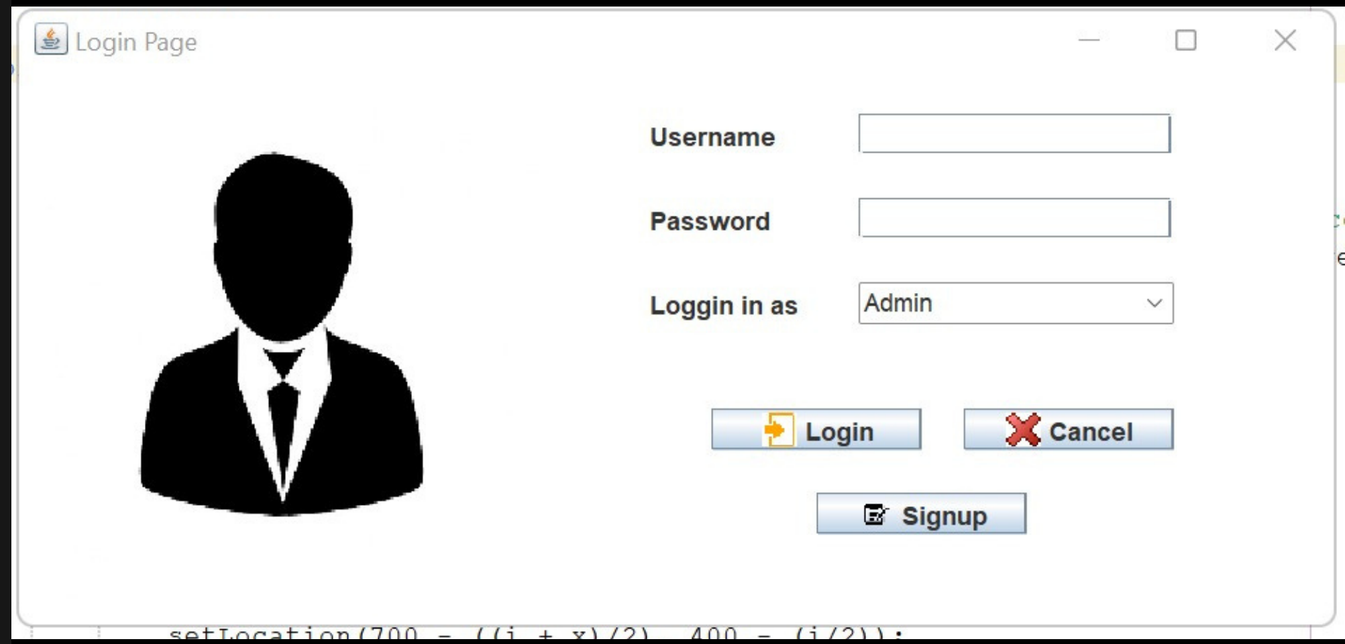
Extra Utilities

Information Management

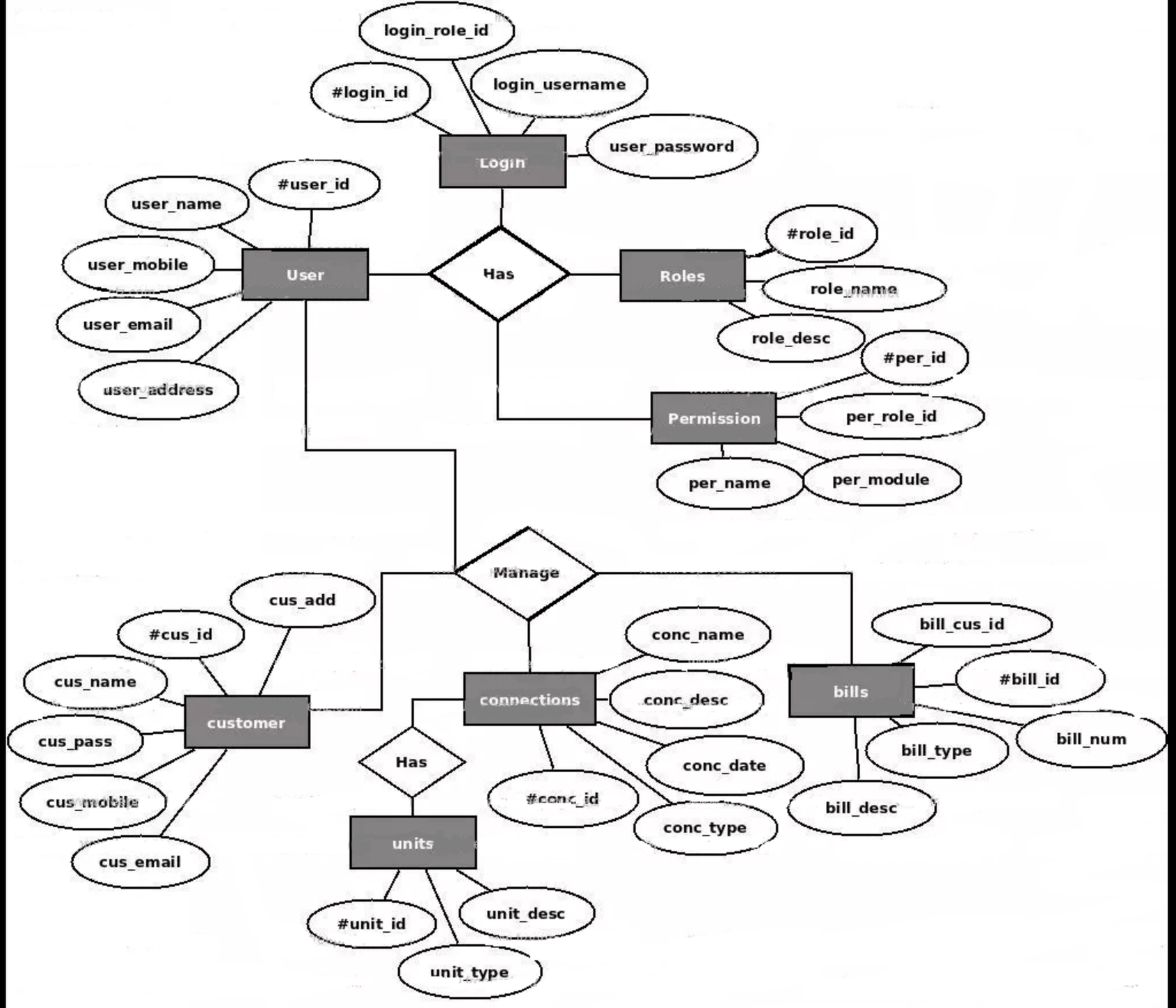
SQL Server Connect

Objective

To Build a GUI connected to SQL that can store, update, delete data as well as manage access to the entire database

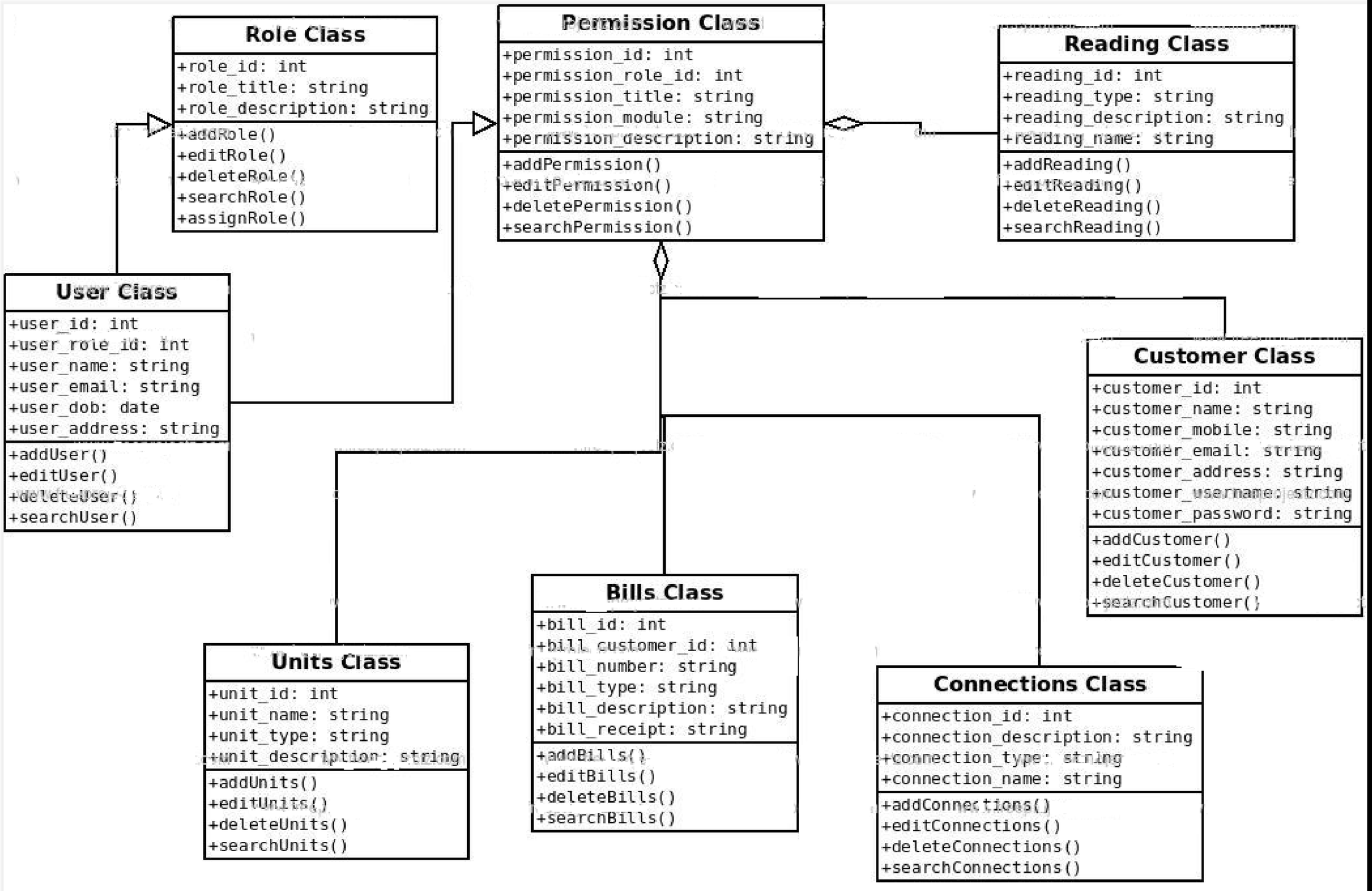


ER Diagram


login_role_id
#login_id
login_username
user_password
Login
#user_id
user_name
user_mobile
user_email
user_address
User
Has
Roles
#role_id
role_name
role_desc
#per_id
per_role_id
per_name
per_module
Permission
Manage
cus_add
#cus_id
cus_name
cus_pass
cus_mobile
cus_email
customer
connections
conc_name
conc_desc
conc_date
conc_type
#conc_id
Has
units
#unit_id
unit_desc
unit_type
bills
bill_cus_id
#bill_id
bill_num
bill_type
bill_desc

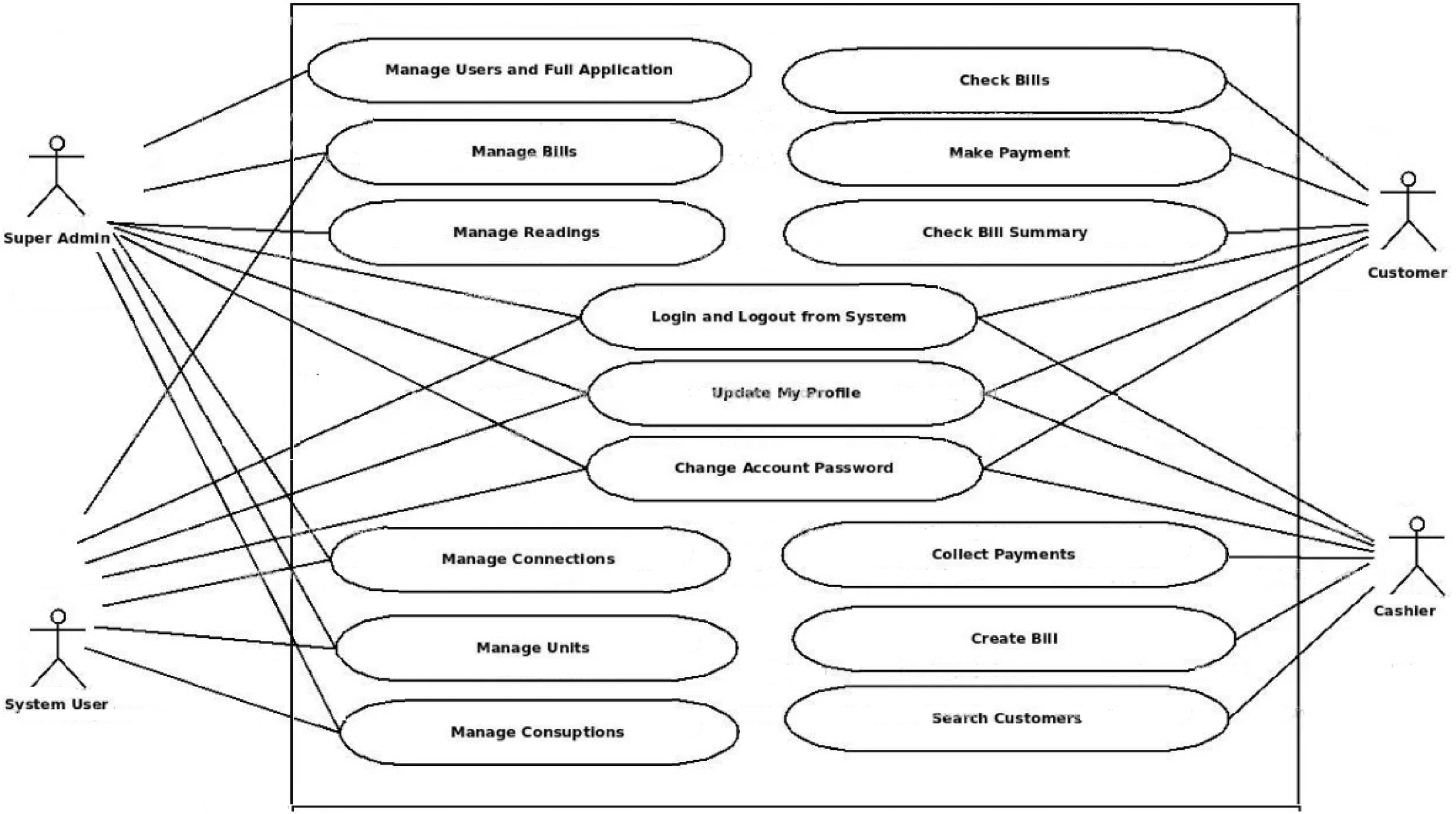
UML Diagrams

Class Diagram

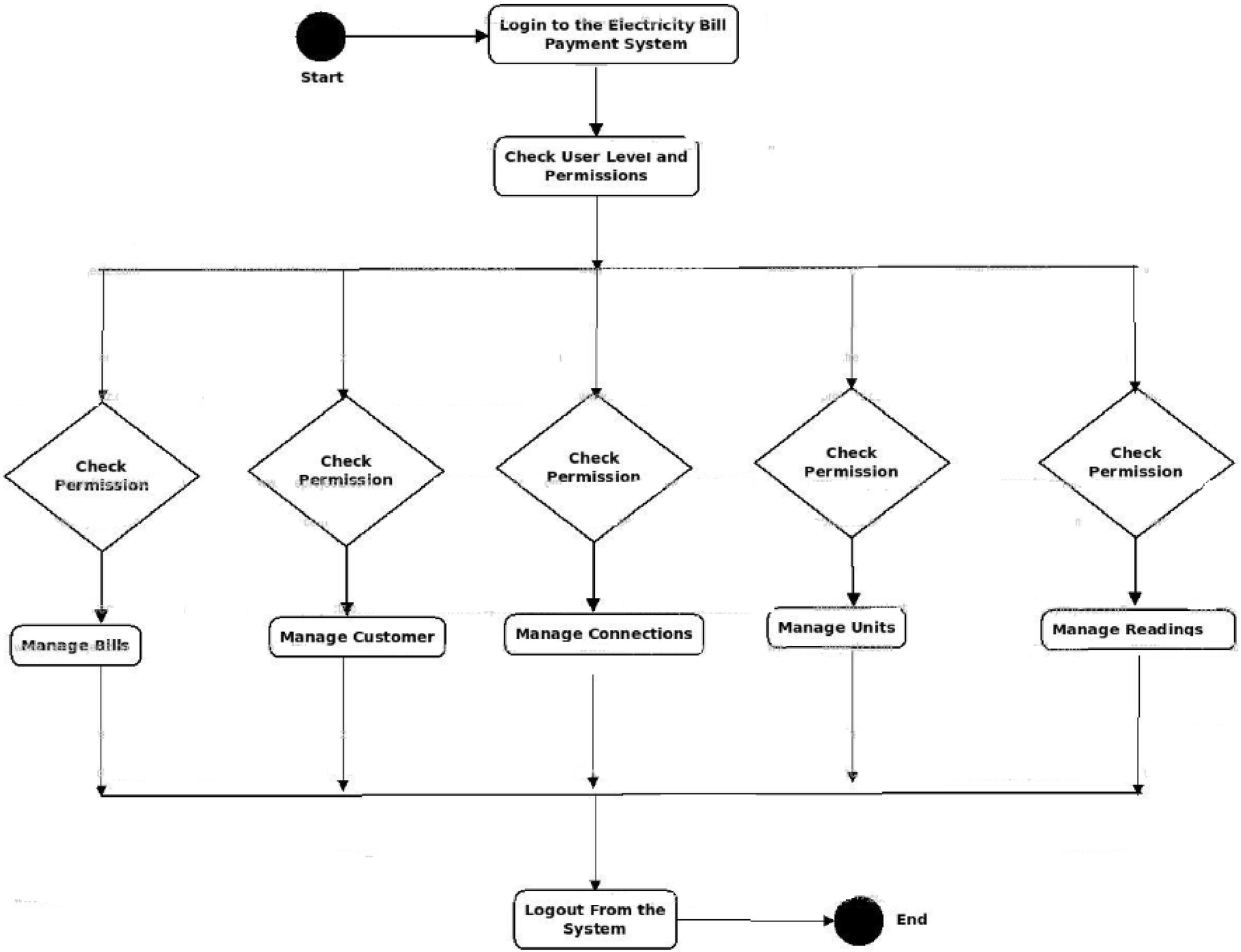

Role Class
+role_id: int
+role_title: string
+role_description: string
+addRole()
+editRole()
+deleteRole()
+searchRole()
+assignRole()
Permission Class
+permission_id: int
+permission_role_id: int
+permission_title: string
+permission_module: string
+permission_description: string
+addPermission()
+editPermission()
+deletePermission()
+searchPermission()
Reading Class
+reading_id: int
+reading_type: string
+reading_description: string
+reading_name: string
+addReading()
+editReading()
+deleteReading()
+searchReading()
User Class
+user_id: int
+user_role_id: int
+user_name: string
+user_email: string
+user_dob: date
+user_address: string
+addUser()
+editUser()
+deleteUser()
+searchUser()
Customer Class
+customer_id: int
+customer_name: string
+customer_mobile: string
+customer_email: string
+customer_address: string
+customer_username: string
+customer_password: string
+addCustomer()
+editCustomer()
+deleteCustomer()
+searchCustomer()
Units Class
+unit_id: int
+unit_name: string
+unit_type: string
+unit_description: string
+addUnits()
+editUnits()
+deleteUnits()
+searchUnits()
Bills Class
+bill_id: int
+bill_customer_id: int
+bill_number: string
+bill_type: string
+bill_description: string
+bill_receipt: string
+addBills()
+editBills()
+deleteBills()
+searchBills()
Connections Class
+connection_id: int
+connection_description: string
+connection_type: string
+connection_name: string
+addConnections()
+editConnections()
+deleteConnections()
+searchConnections()

Use Case Diagram

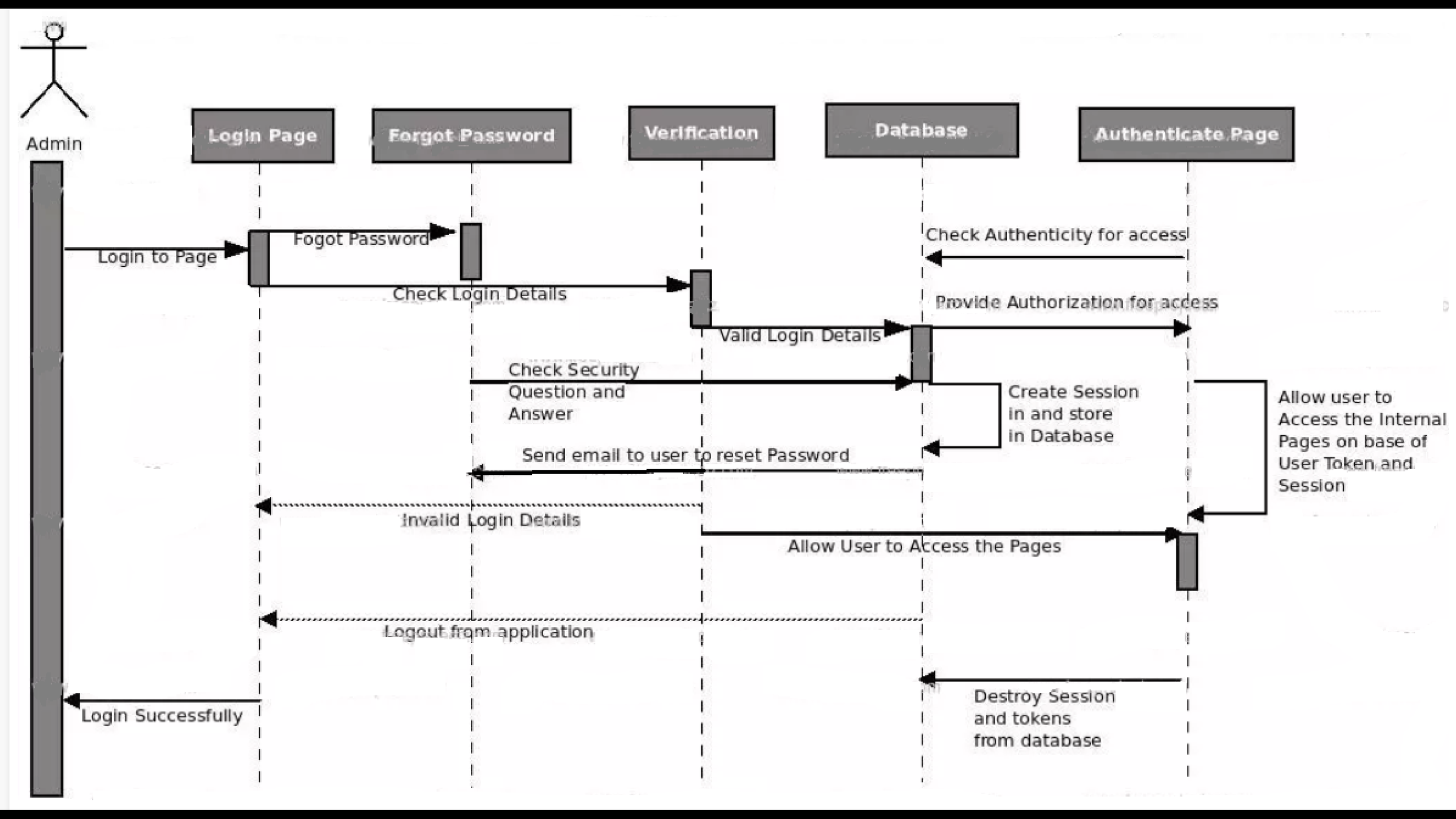
Use Case Diagram of Electricity Bill Payment System :



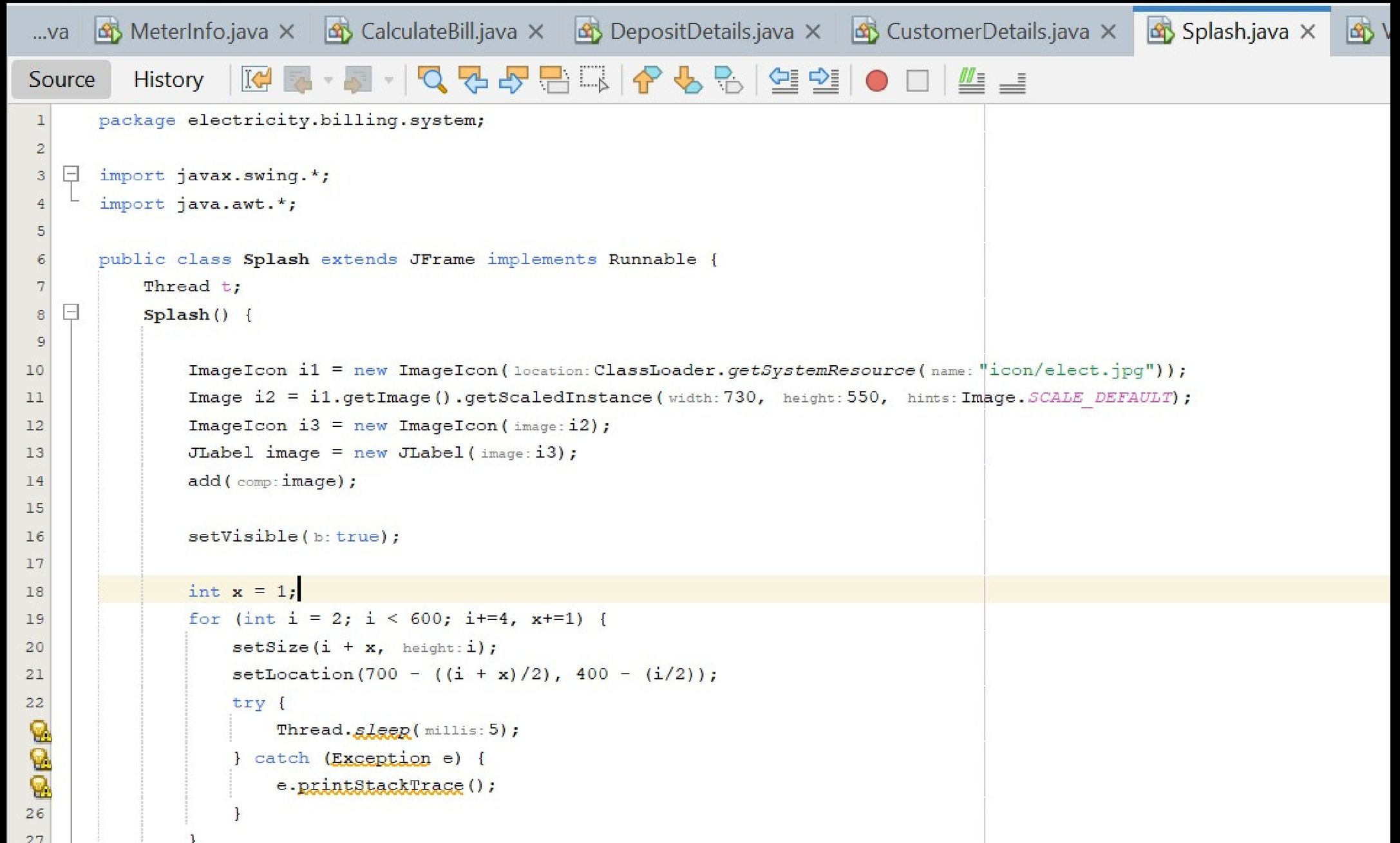
Activity Diagram


Start
Login to the Electricity Bill Payment System
Check User Level and Permissions
Check Permission
Check Permission
Check Permission
Check Permission
Check Permission
Manage Bills
Manage Customer
Manage Connections
Manage Units
Manage Readings
Logout From the System
End

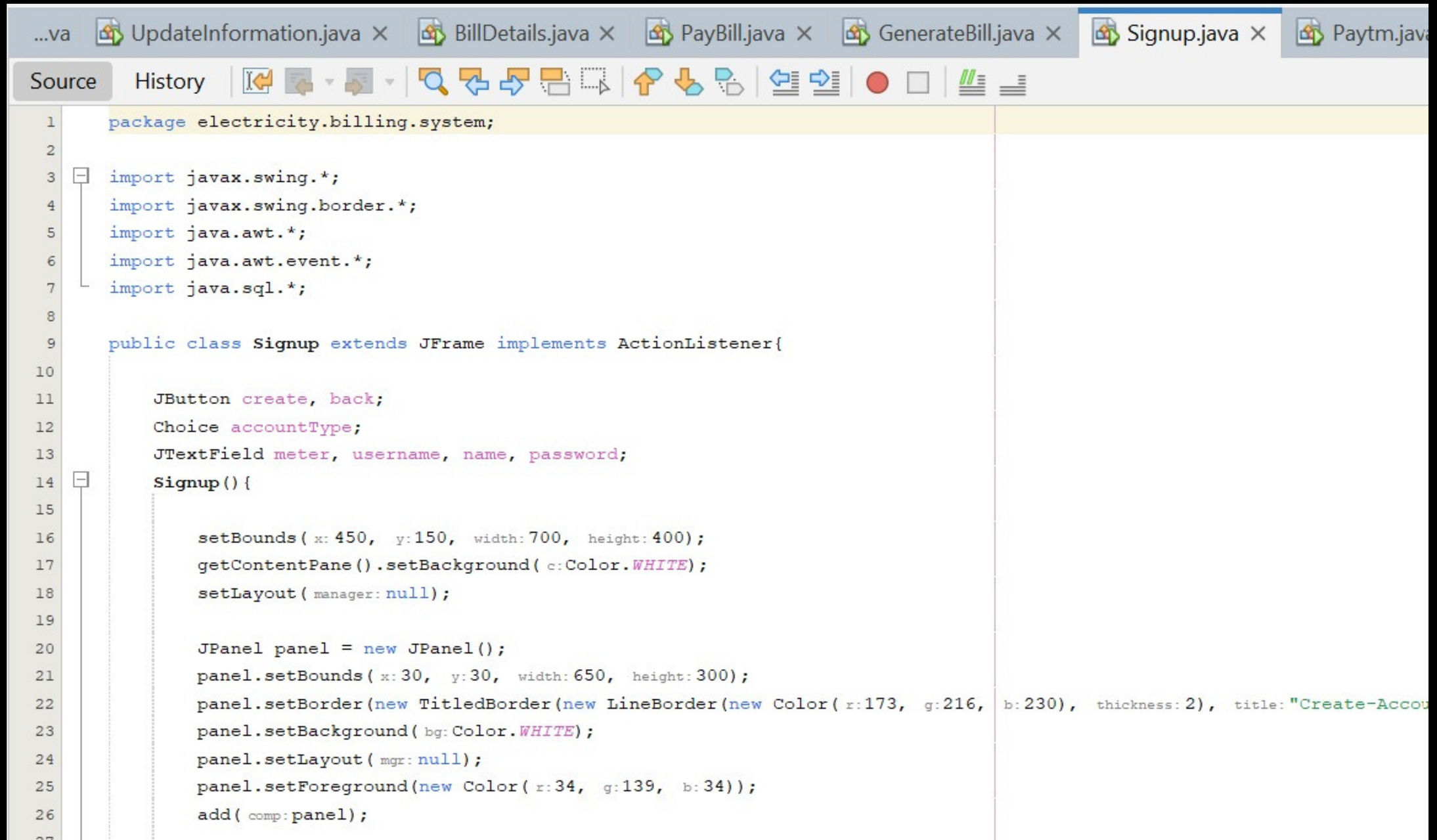
Sequence Diagram


Admin
Login Page
Forgot Password
Verification
Database
Authenticate Page
Login to Page
Fogot Password
Check Login Details
Valid Login Details
Check Authenticity for access
Provide Authorization for access
Check Security Question and Answer
Create Session in and store in Database
Allow user to Access the Internal Pages on base of User Token and Session
Send email to user to reset Password
Invalid Login Details
Allow User to Access the Pages
Logout from application
Destroy Session and tokens from database
Login Successfully

The screenshot of the codes. Splash Java

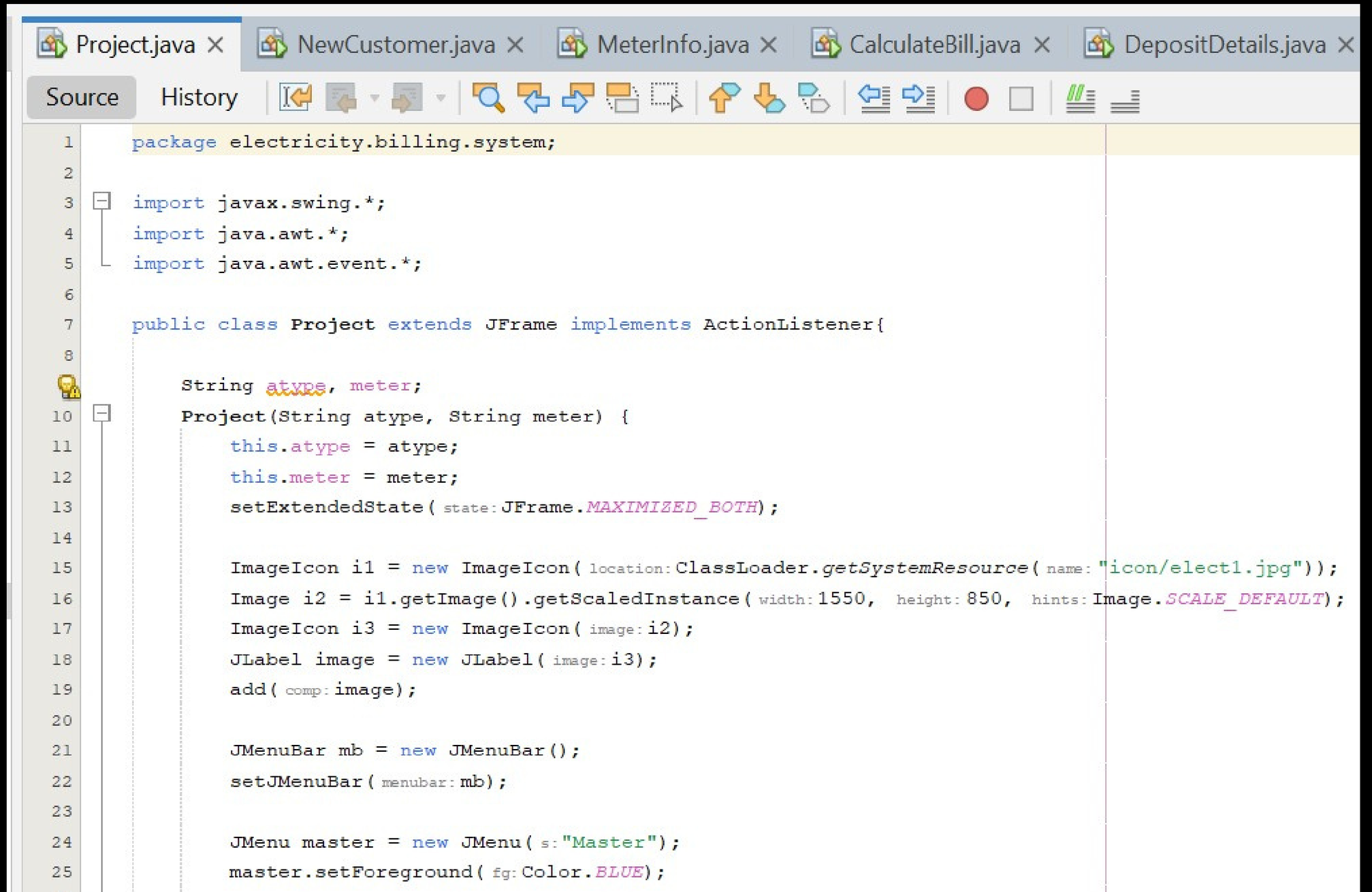


The screenshot of the codes. Sign Up Java

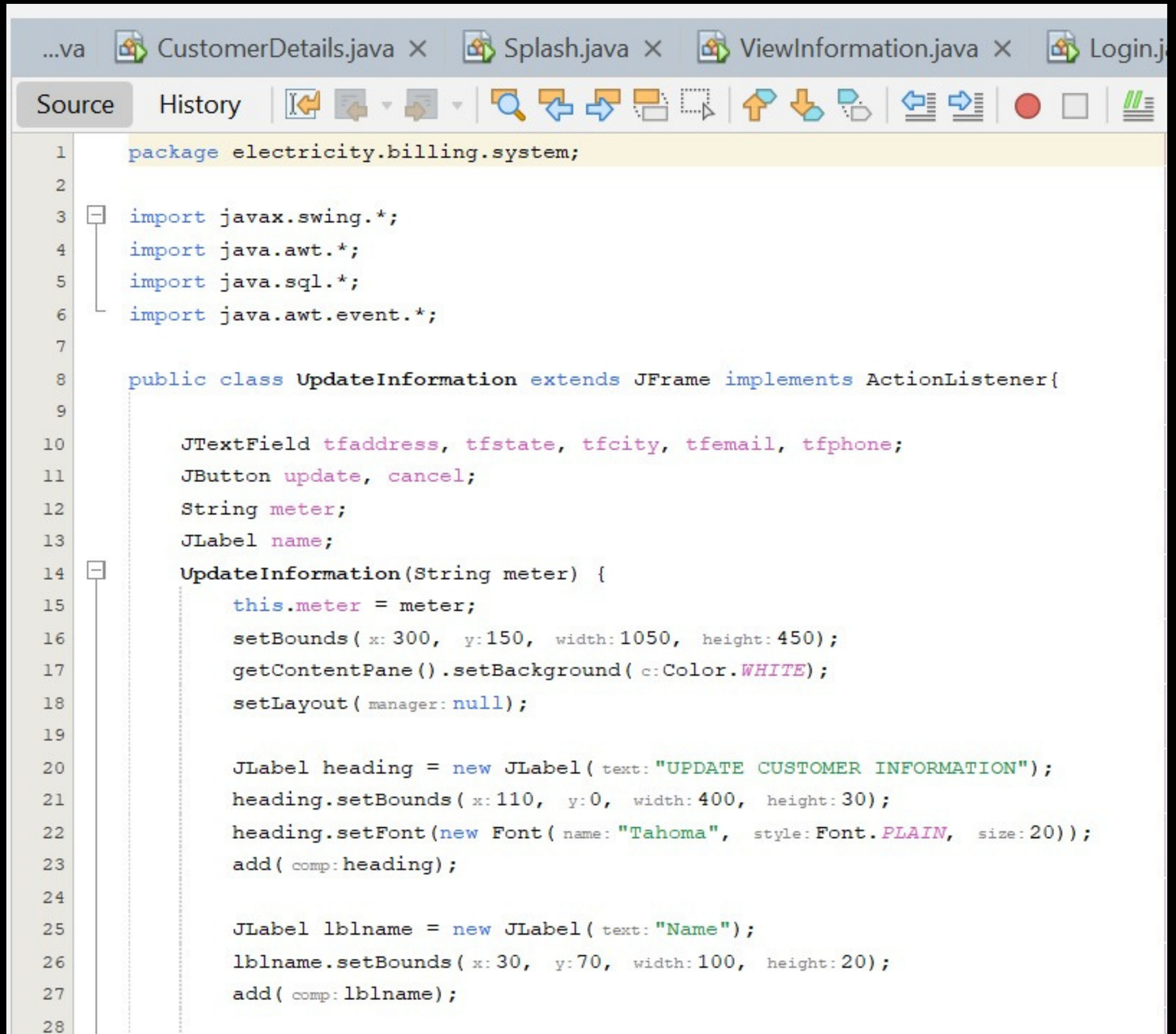



```java
package electricity.billing.system;

import javax.swing.*;
import javax.swing.border.*;
import java.awt.*;
import java.awt.event.*;
import java.sql.*;

public class Signup extends JFrame implements ActionListener{

    JButton create, back;
    Choice accountType;
    JTextField meter, username, name, password;
    Signup(){

        setBounds( x: 450,  y: 150,  width: 700,  height: 400);
        getContentPane().setBackground( c: Color.WHITE);
        setLayout( manager: null);

        JPanel panel = new JPanel();
        panel.setBounds( x: 30,  y: 30,  width: 650,  height: 300);
        panel.setBorder(new TitledBorder(new LineBorder(new Color( r: 173,  g: 216,  b: 230),  thickness: 2),  title: "Create-Accou
        panel.setBackground( bg: Color.WHITE);
        panel.setLayout( mgr: null);
        panel.setForeground(new Color( r: 34,  g: 139,  b: 34));
        add( comp: panel);
```

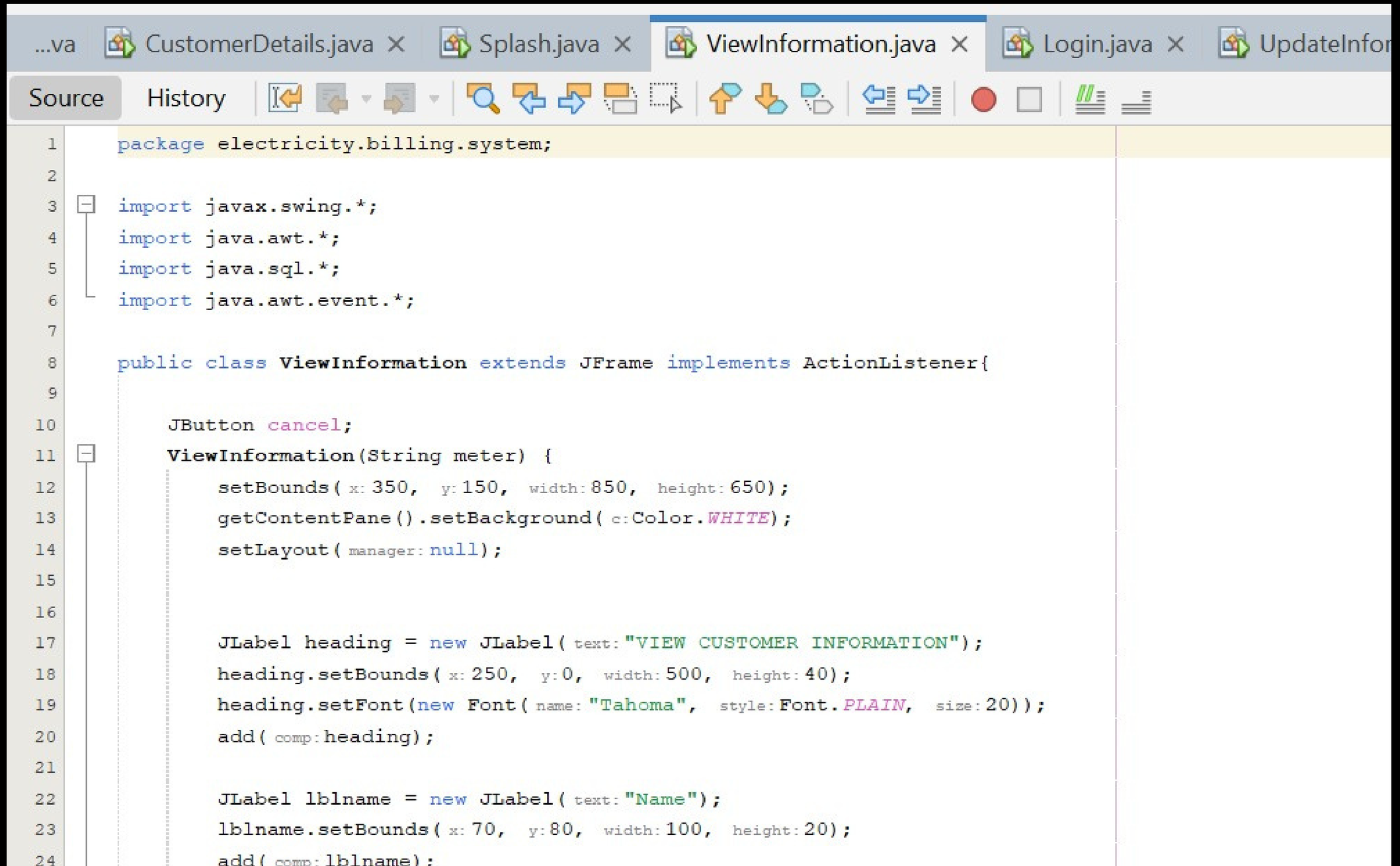
The screenshot of the codes. Project Java



The screenshot of the codes. Update Java

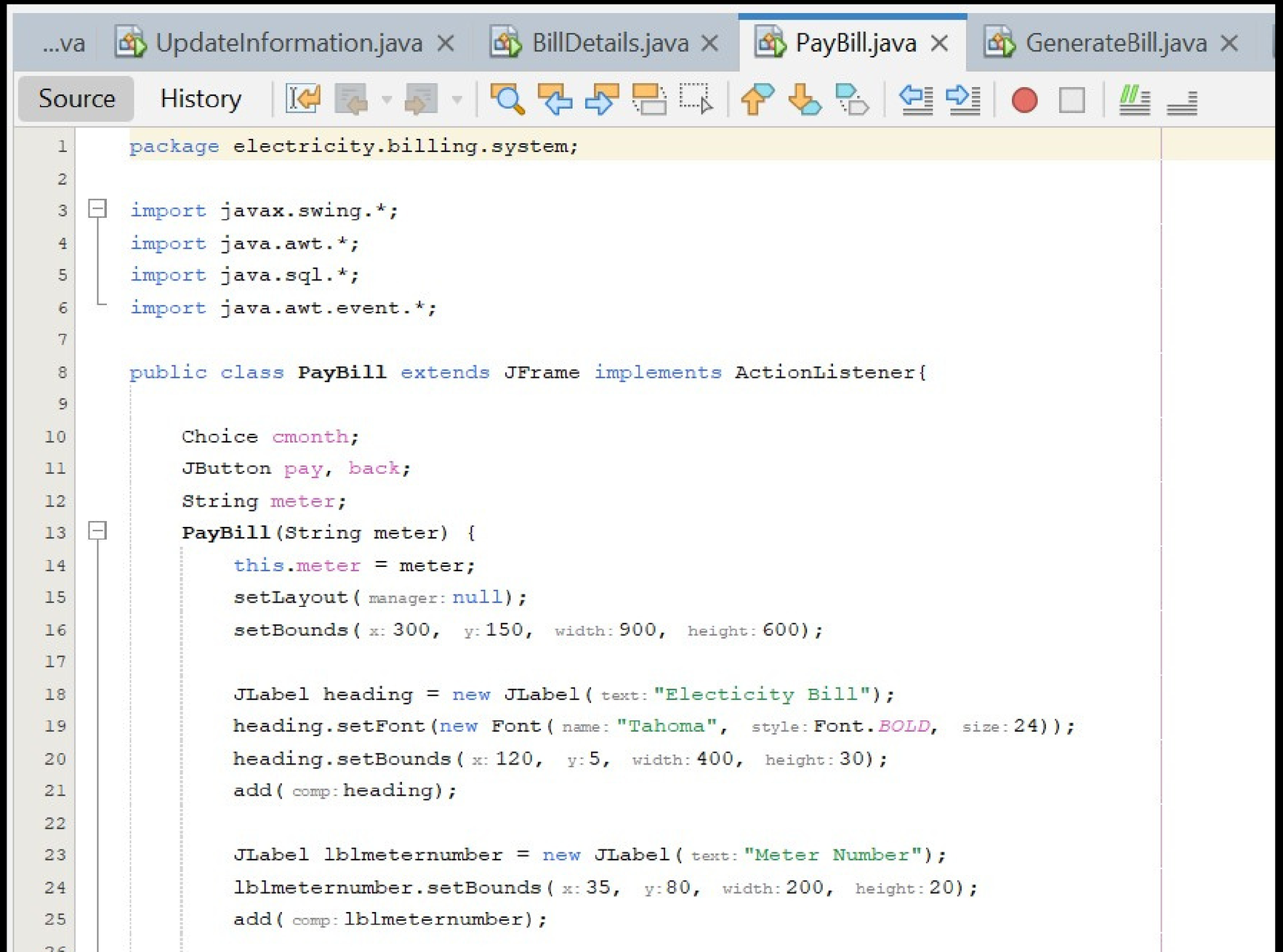


The screenshot of the codes. View Java

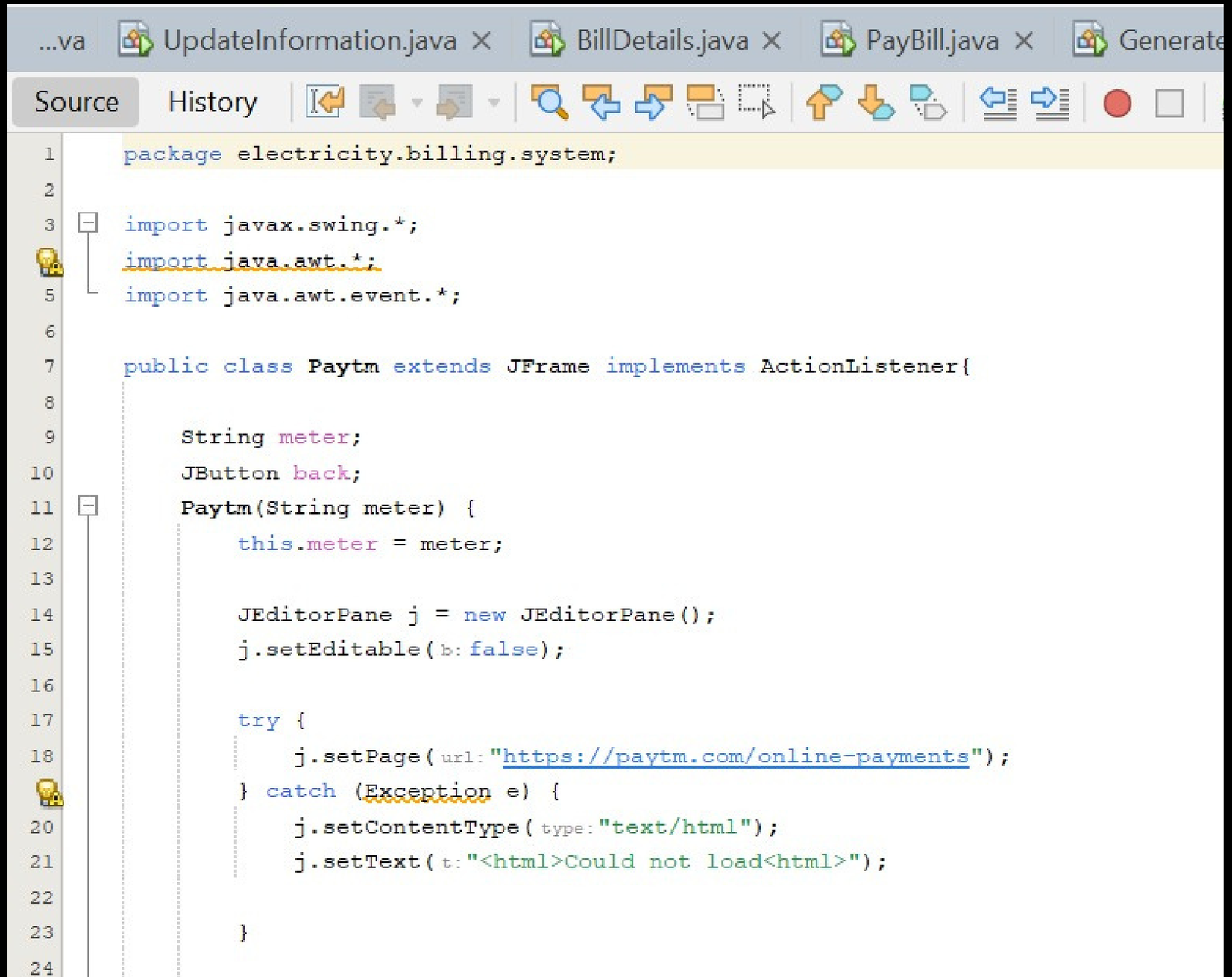



```java
package electricity.billing.system;

import javax.swing.*;
import java.awt.*;
import java.sql.*;
import java.awt.event.*;

public class ViewInformation extends JFrame implements ActionListener{

    JButton cancel;
    ViewInformation(String meter) {
        setBounds( x: 350,  y: 150,  width: 850,  height: 650);
        getContentPane().setBackground( c: Color.WHITE);
        setLayout( manager: null);


        JLabel heading = new JLabel( text: "VIEW CUSTOMER INFORMATION");
        heading.setBounds( x: 250,  y: 0,  width: 500,  height: 40);
        heading.setFont(new Font( name: "Tahoma",  style: Font.PLAIN,  size: 20));
        add( comp: heading);

        JLabel lblname = new JLabel( text: "Name");
        lblname.setBounds( x: 70,  y: 80,  width: 100,  height: 20);
        add( comp: lblname);
```

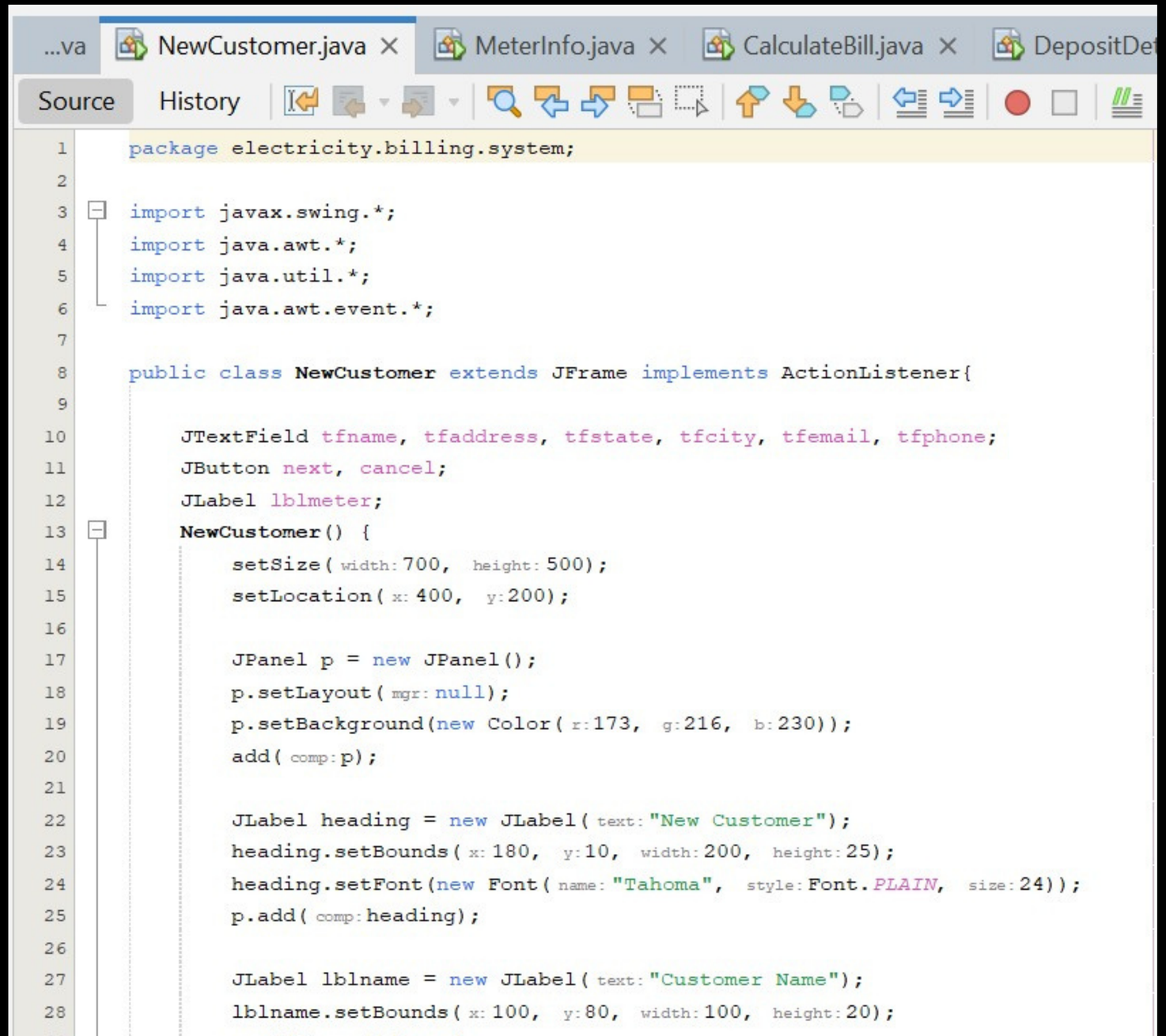
The screenshot of the codes. Pay Bill Java



The screenshot of the codes. Paytm Java

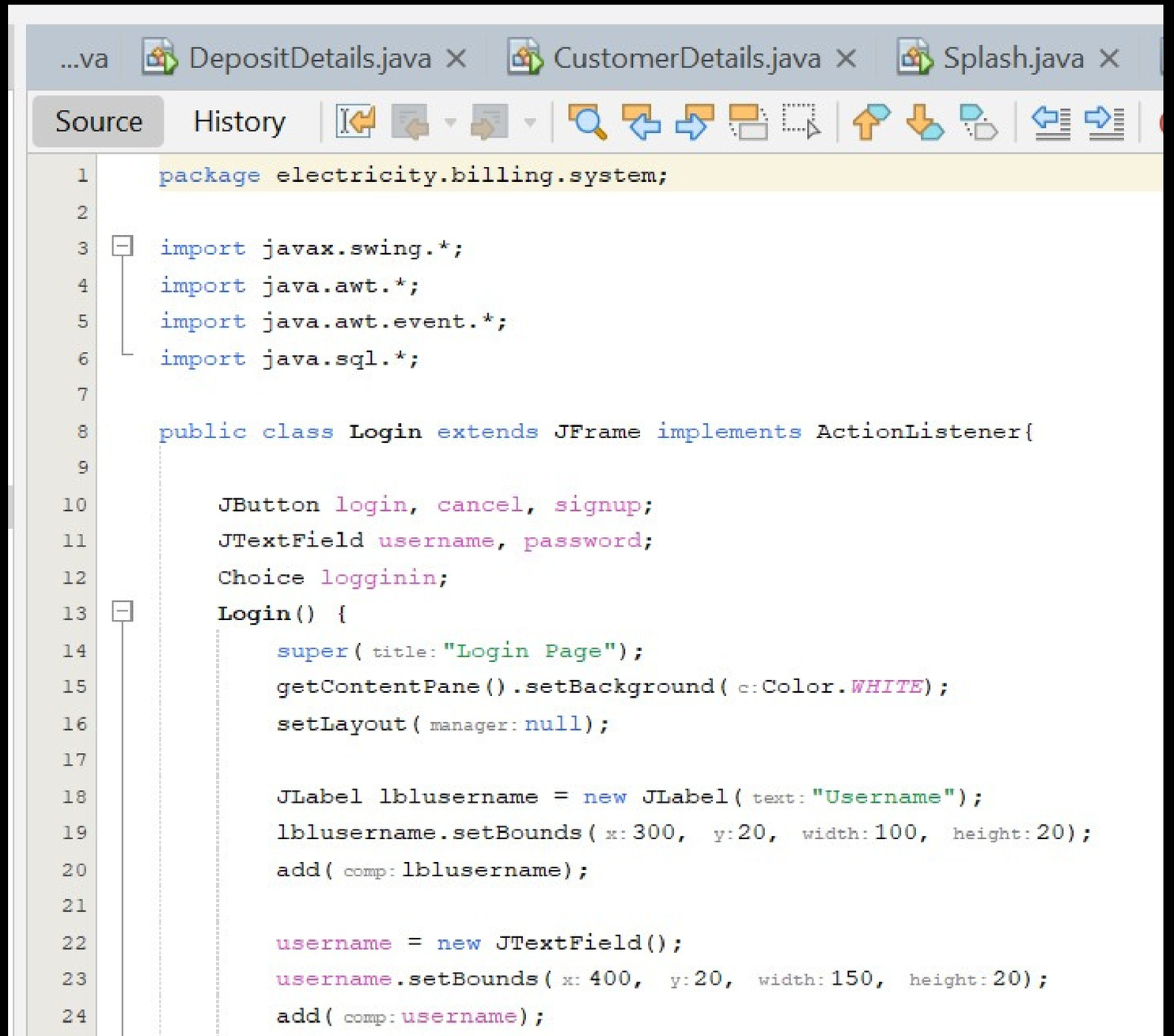


The screenshot of the codes. Add Customer Java

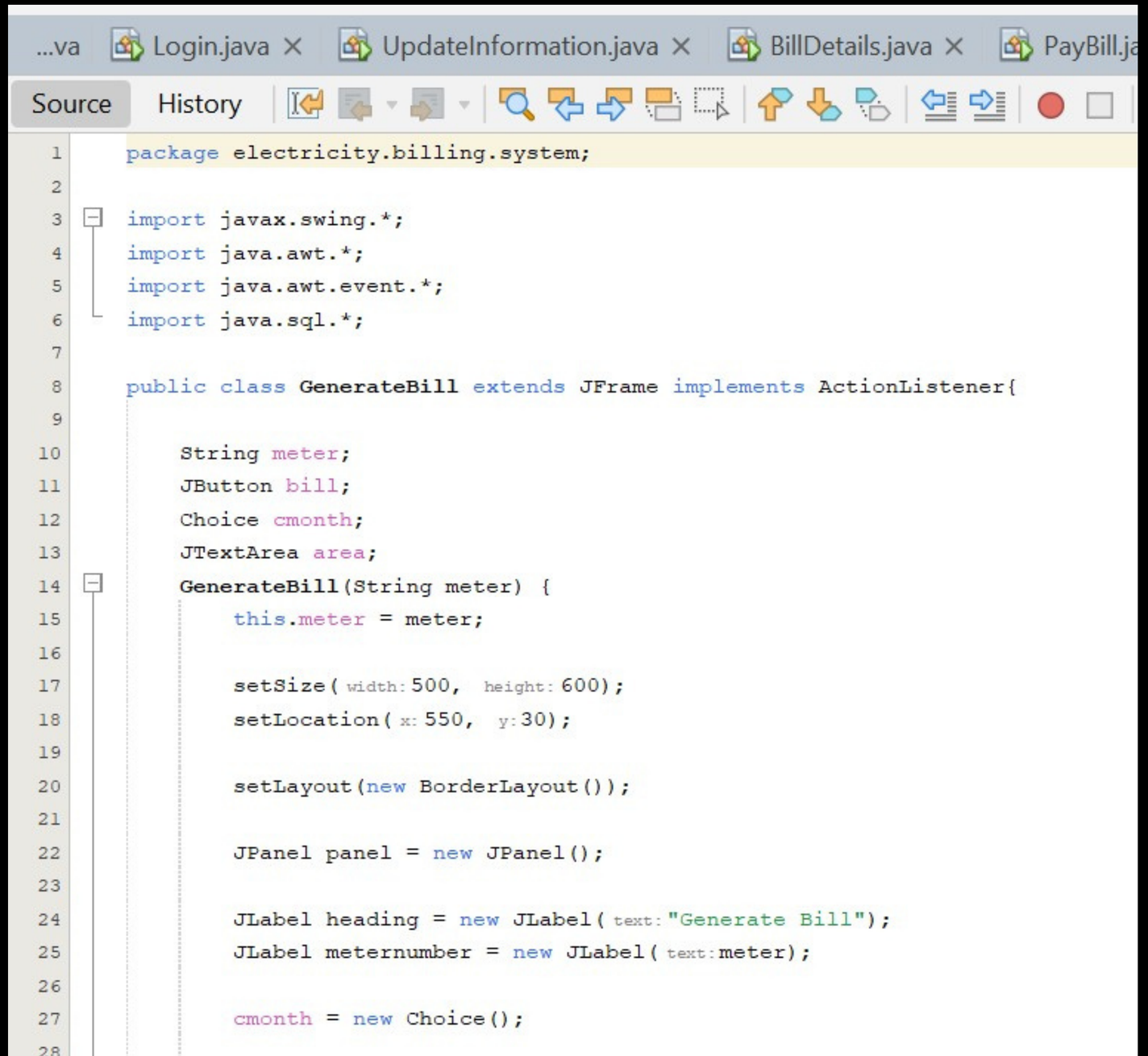



```java
package electricity.billing.system;

import javax.swing.*;
import java.awt.*;
import java.util.*;
import java.awt.event.*;

public class NewCustomer extends JFrame implements ActionListener{

    JTextField tfname, tfaddress, tfstate, tfcity, tfemail, tfphone;
    JButton next, cancel;
    JLabel lblmeter;
    NewCustomer() {
        setSize( width: 700,  height: 500);
        setLocation( x: 400,  y: 200);

        JPanel p = new JPanel();
        p.setLayout( mgr: null);
        p.setBackground(new Color( r: 173,  g: 216,  b: 230));
        add( comp: p);

        JLabel heading = new JLabel( text: "New Customer");
        heading.setBounds( x: 180,  y: 10,  width: 200,  height: 25);
        heading.setFont(new Font( name: "Tahoma",  style: Font.PLAIN,  size: 24));
        p.add( comp: heading);

        JLabel lblname = new JLabel( text: "Customer Name");
        lblname.setBounds( x: 100,  y: 80,  width: 100,  height: 20);
```

The screenshot of the codes. Login Java

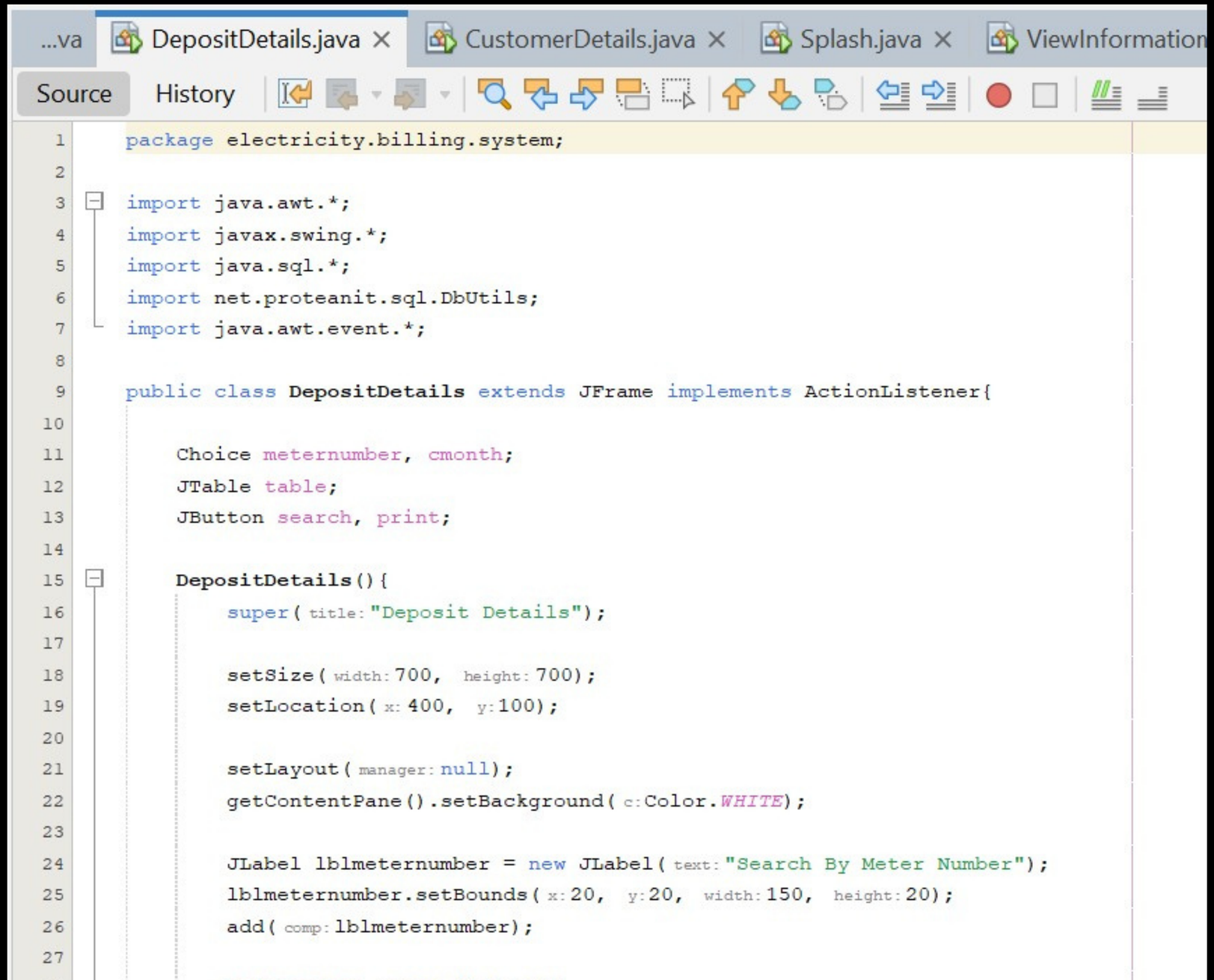


The screenshot of the codes. Generate Bill Java

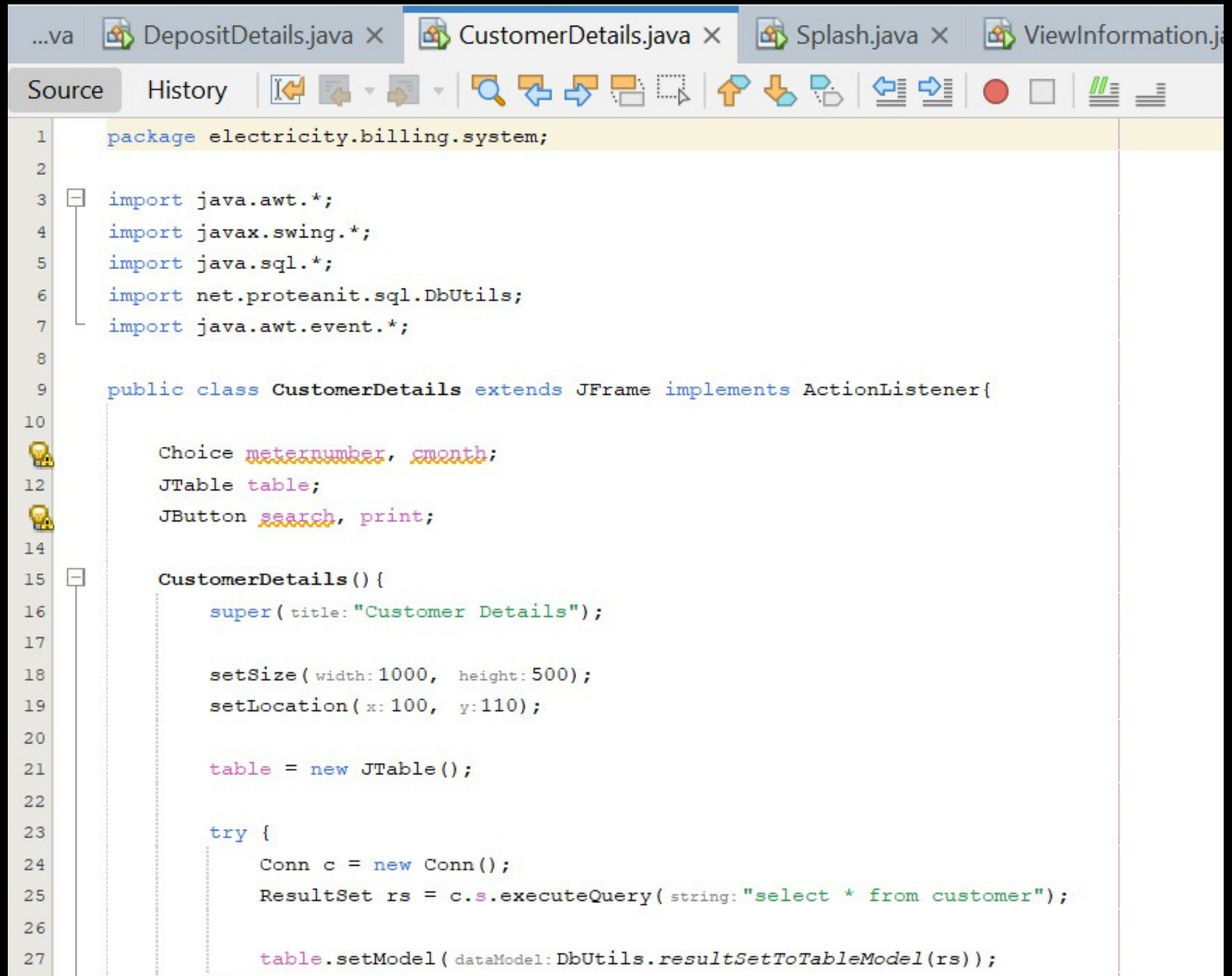



```java
package electricity.billing.system;

import javax.swing.*;
import java.awt.*;
import java.awt.event.*;
import java.sql.*;

public class GenerateBill extends JFrame implements ActionListener{

    String meter;
    JButton bill;
    Choice cmonth;
    JTextArea area;
    GenerateBill(String meter) {
        this.meter = meter;

        setSize( width: 500,  height: 600);
        setLocation( x: 550,  y: 30);

        setLayout(new BorderLayout());

        JPanel panel = new JPanel();

        JLabel heading = new JLabel( text: "Generate Bill");
        JLabel meternumber = new JLabel( text: meter);

        cmonth = new Choice();
```

The screenshot of the codes. Deposit DetailsJava

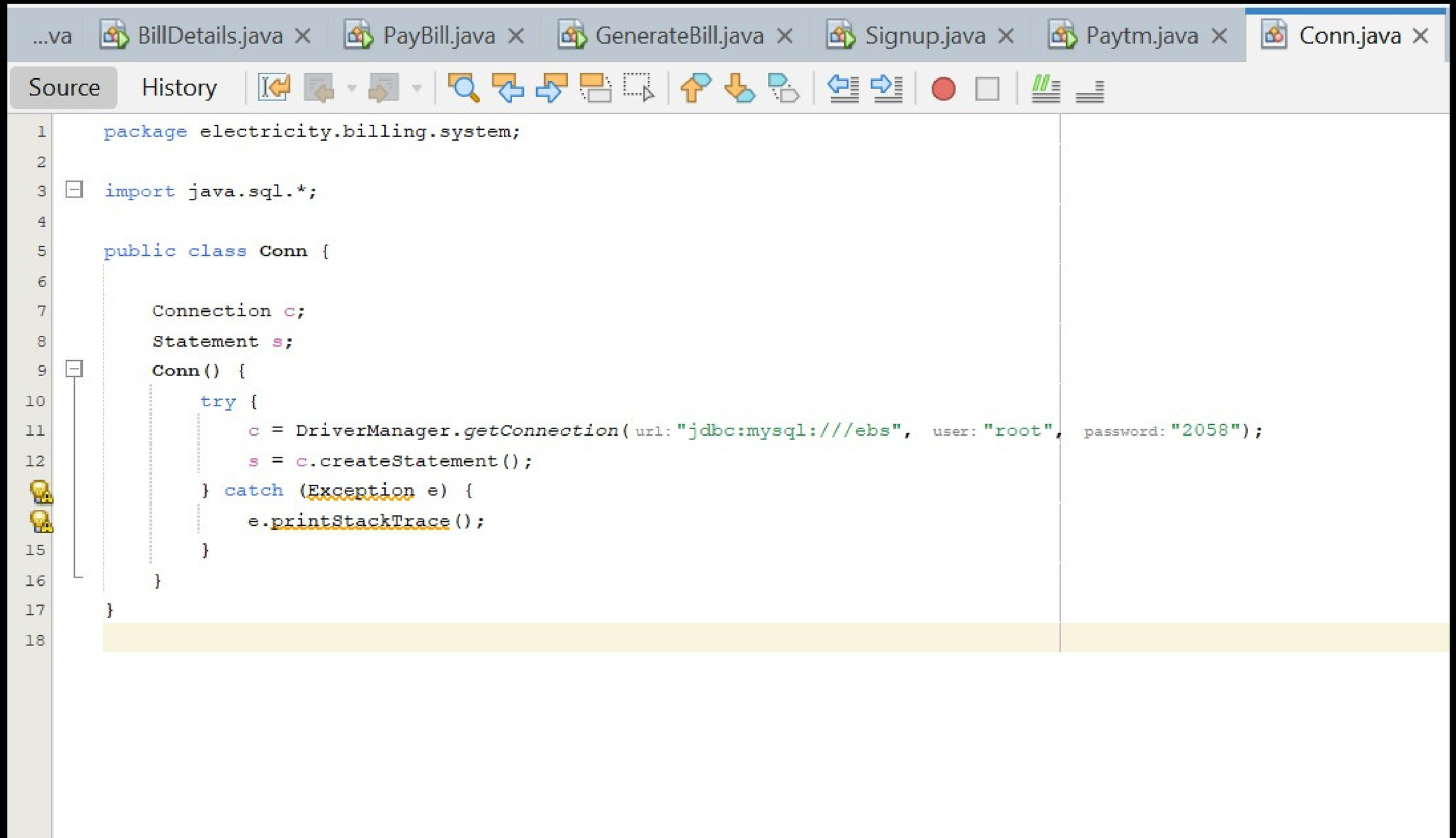


The screenshot of the codes. Customer Details Java

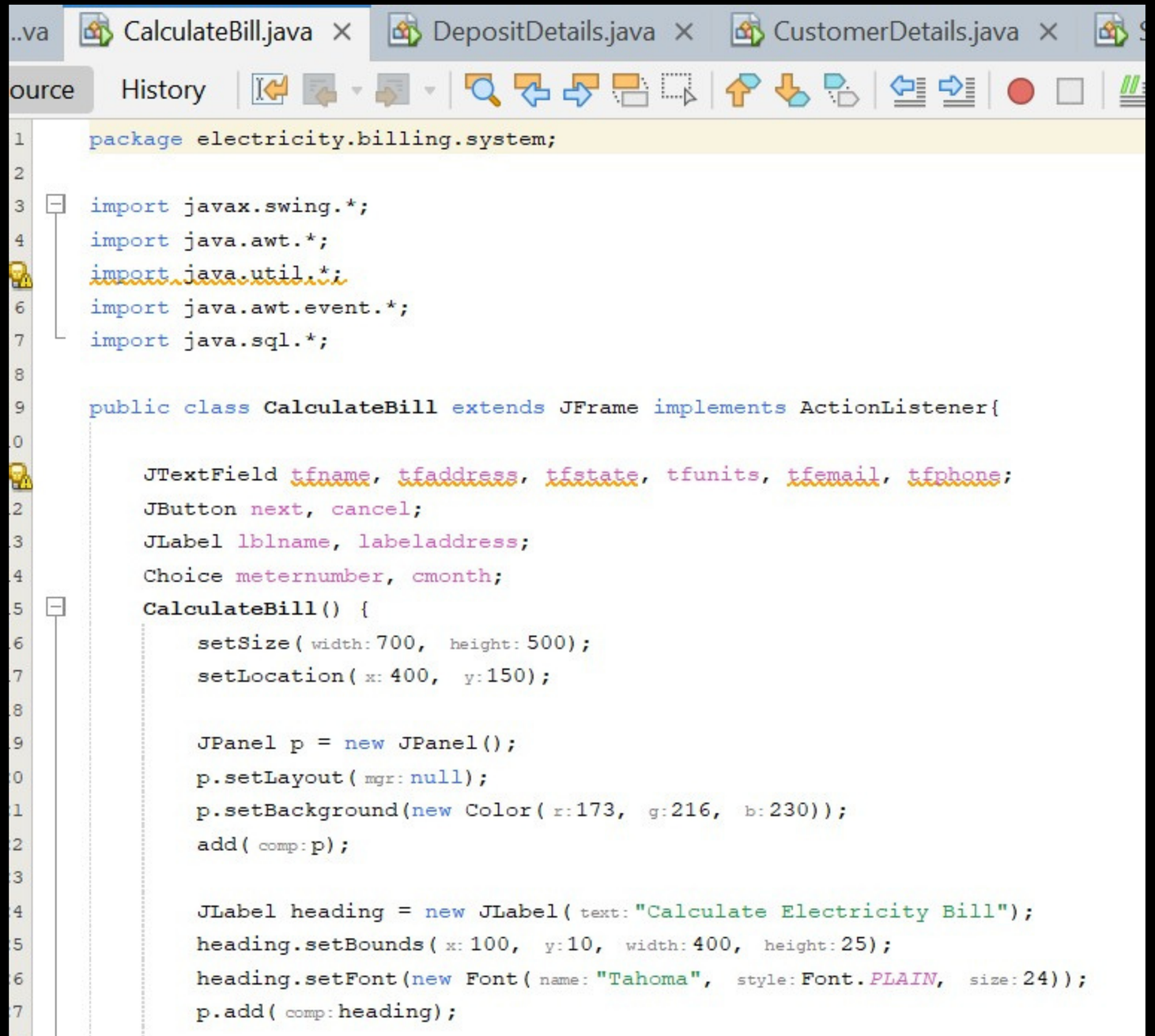



```
1  package electricity.billing.system;
2
3  import java.awt.*;
4  import javax.swing.*;
5  import java.sql.*;
6  import net.proteanit.sql.DbUtils;
7  import java.awt.event.*;
8
9  public class CustomerDetails extends JFrame implements ActionListener{
10
11     Choice meternumber, cmonth;
12     JTable table;
13     JButton search, print;
14
15     CustomerDetails(){
16         super( title: "Customer Details");
17
18         setSize( width: 1000,  height: 500);
19         setLocation( x: 100,  y: 110);
20
21         table = new JTable();
22
23         try {
24             Conn c = new Conn();
25             ResultSet rs = c.s.executeQuery( string: "select * from customer");
26
27             table.setModel( dataModel: DbUtils.resultSetToTableModel(rs));
```

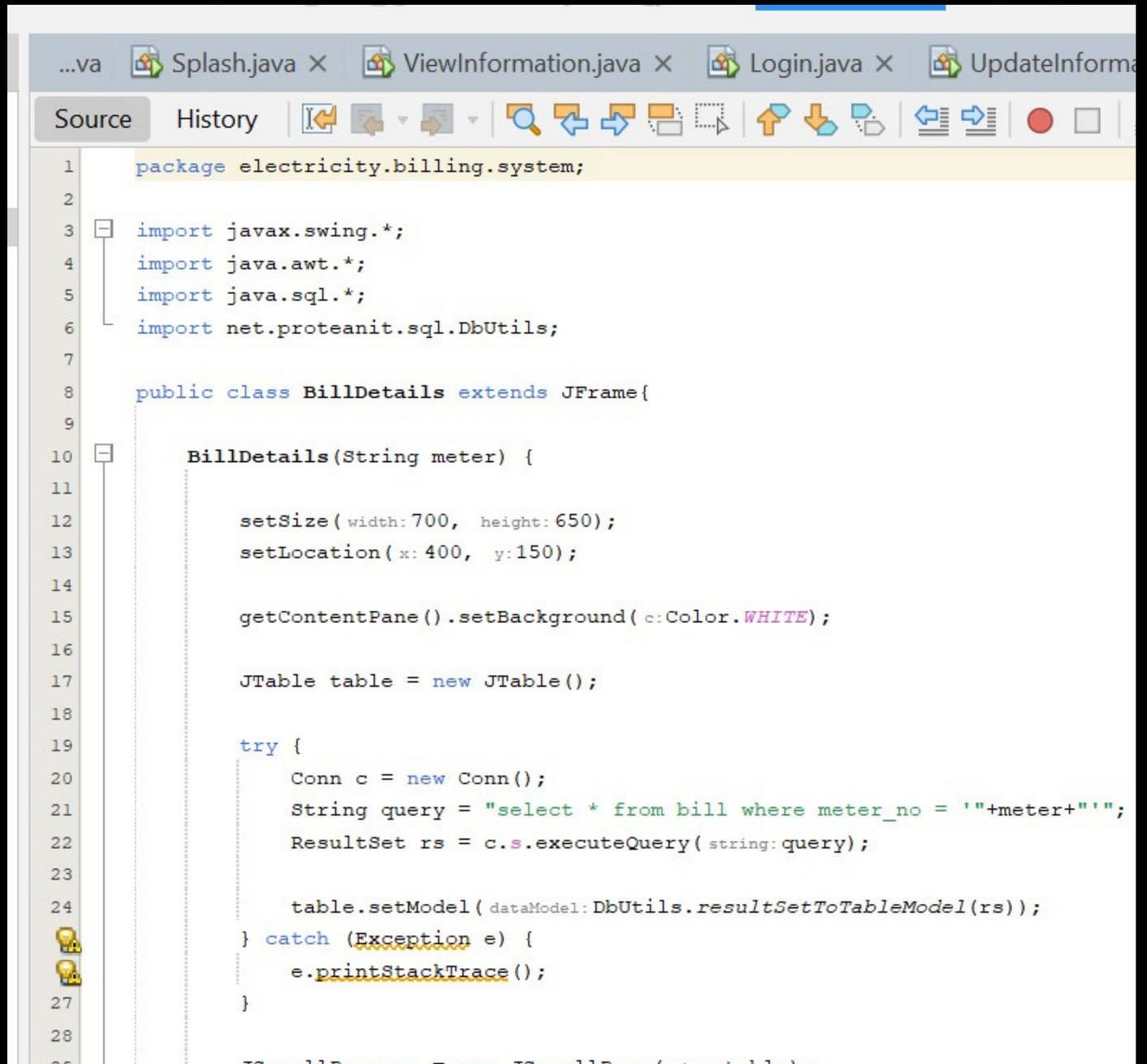
The screenshot of the codes. Connection Java



The screenshot of the codes. Calculate Bill Java

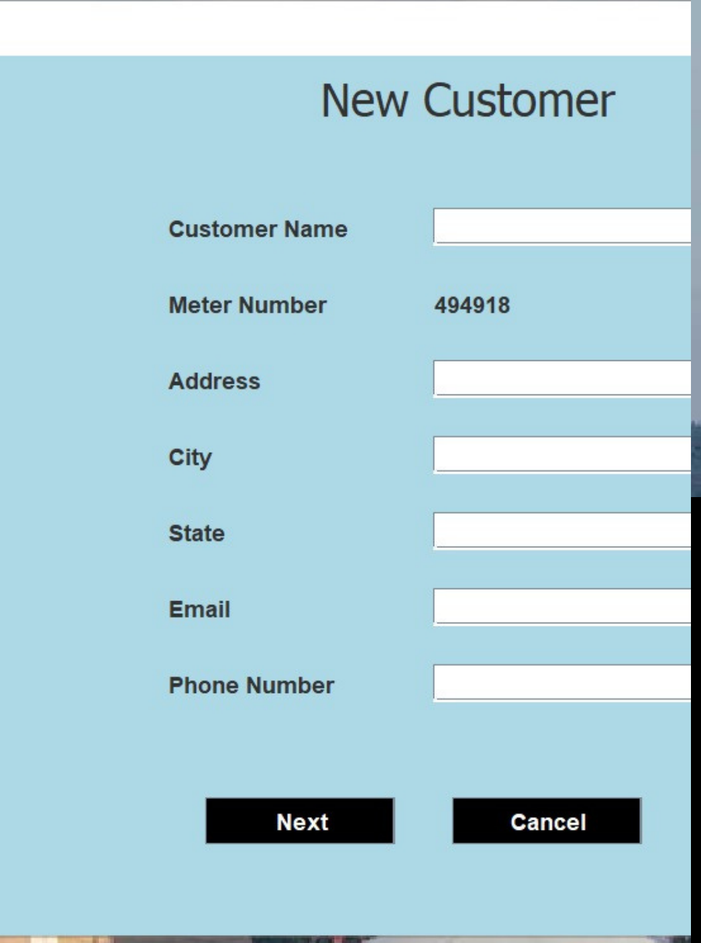


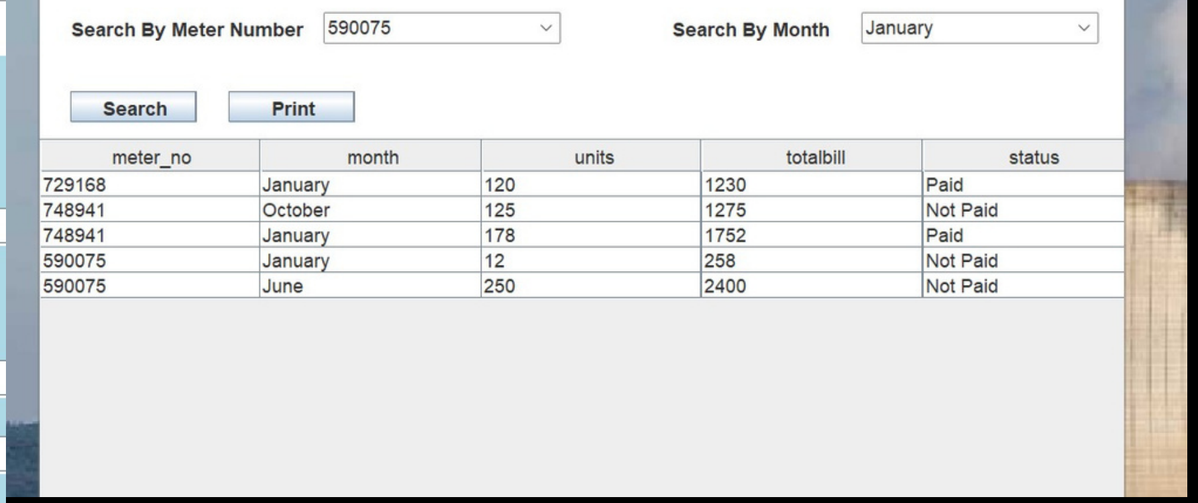
The screenshot of the codes. Bill Detalis Java

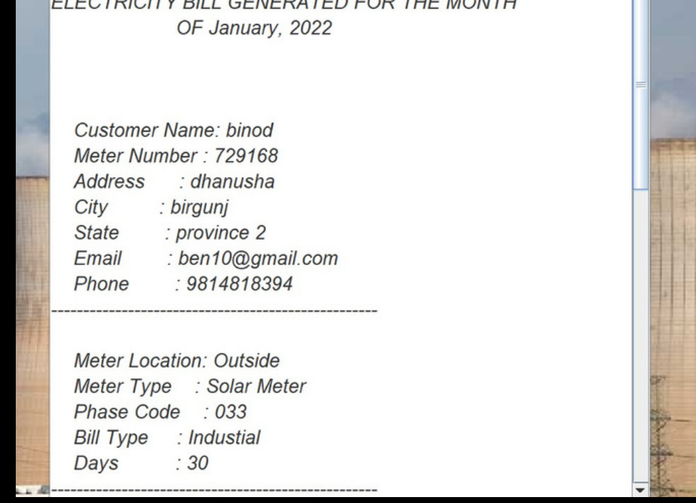


SQL Code

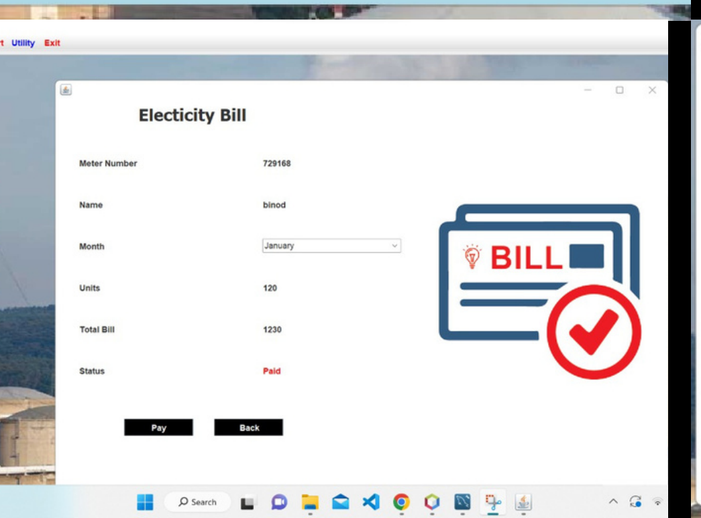
```
create database ebs;

use ebs;
create table login(meter_no varchar(20),
username varchar(30),
name varchar(30), password varchar(20), user
varchar(30));

select * from login;

create table customer(name varchar(20) ,
meter_no varchar(20),
address varchar(50), city varchar(30),state
varchar(30),email varchar(40),phone
varchar(20));

select * from login;
select * from customer;
create table meter_info(meter_no varchar(20),
meter_location
varchar(20), meter_type varchar(20), phase_code
varchar(20),
bill_type varchar(20),days varchar(20));
select * from meter_info;

create table tax(cost_per_unit varchar(20), meter_rent
varchar(20), service_charge varchar(20), service_tax
varchar(20), swacch_bharat_cess varchar(20), fixed_tax
varchar(20));
insert into tax values('9','47','22','57','6','18');
select * from tax;

create table bill(meter_no varchar(20), month varchar(30),
units varchar(20), totalbill varchar(20), status varchar(20));
select * from bill
```

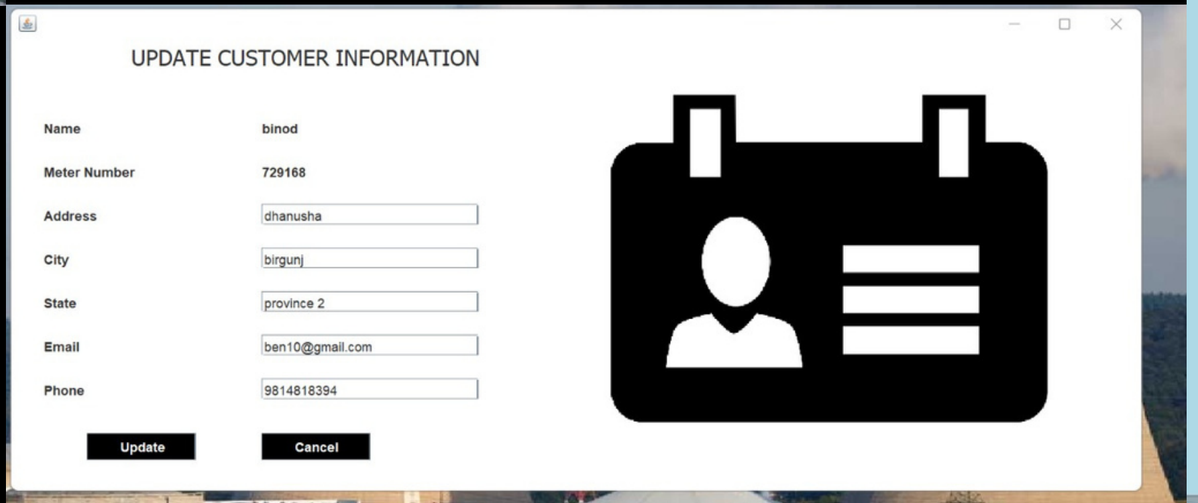

New Customer
Customer Name
Meter Number
494918
Address
City
State
Email
Phone Number
Next
Cancel

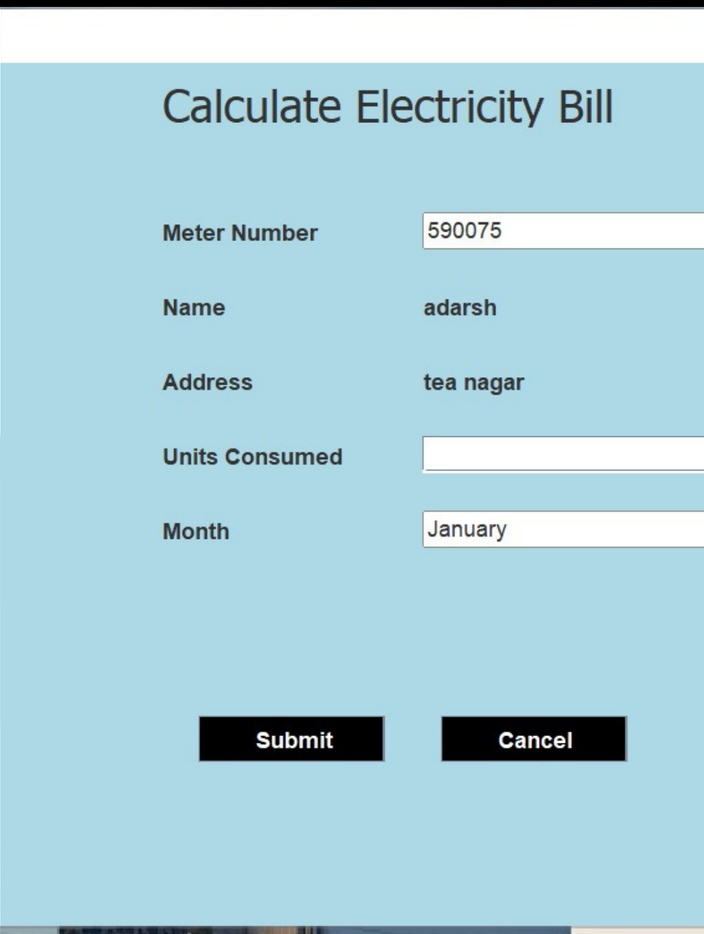

Search By Meter Number
590075
Search By Month
January
Search
Print
meter_no month units totalbill status
729168 January 120 1230 Paid
748941 October 125 1275 Not Paid
748941 January 178 1752 Paid
590075 January 12 258 Not Paid
590075 June 250 2400 Not Paid


ELECTRICITY BILL GENERATED FOR THE MONTH
OF January, 2022
Customer Name: binod
Meter Number : 729168
Address : dhanusha
City : birgunj
State : province 2
Email : ben10@gmail.com
Phone : 9814818394
Meter Location: Outside
Meter Type : Solar Meter
Phase Code : 033
Bill Type : Industrial
Days : 30

RESULTS


Utility Exit
Electicity Bill
Meter Number 729168
Name binod
Month January
Units 120
Total Bill 1230
Status Paid
Pay
Back
BILL


UPDATE CUSTOMER INFORMATION
Name binod
Meter Number 729168
Address dhanusha
City birgunj
State province 2
Email ben10@gmail.com
Phone 9814818394
Update
Cancel


Calculate Electricity Bill
Meter Number 590075
Name adarsh
Address tea nagar
Units Consumed
Month January
Submit
Cancel

Customer Details

name	meter_no	address	city	state	email	phone
adarsh	590075	tea nagar	coimbatore	tamil nadu	iloveubaby@gmai...	9003587451
binod	729168	dhanusha	birgunj	province 2	ben10@gmail.com	9814818394
anirudh	748941	anna nagar	delhi	ap	anirodh@gmail.com	9293042100
binod	828211	jeetpur	bara	3	abc@gmail.com	383435366

Print

General | Page Setup | Appearance

Print Service

Name: Microsoft Print to PDF | Properties...

Status: Accepting jobs

Type:

Info:

Print To File

Print Range

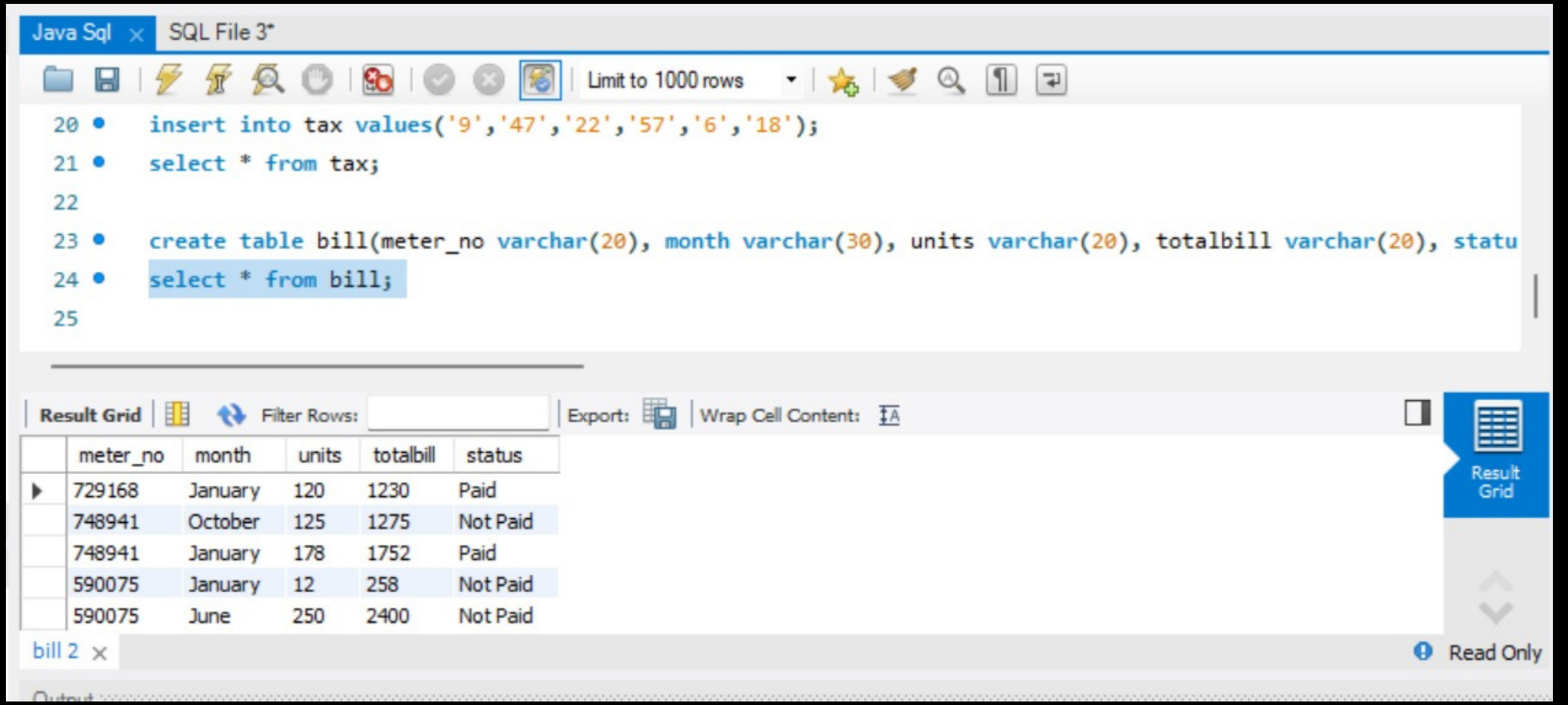
All

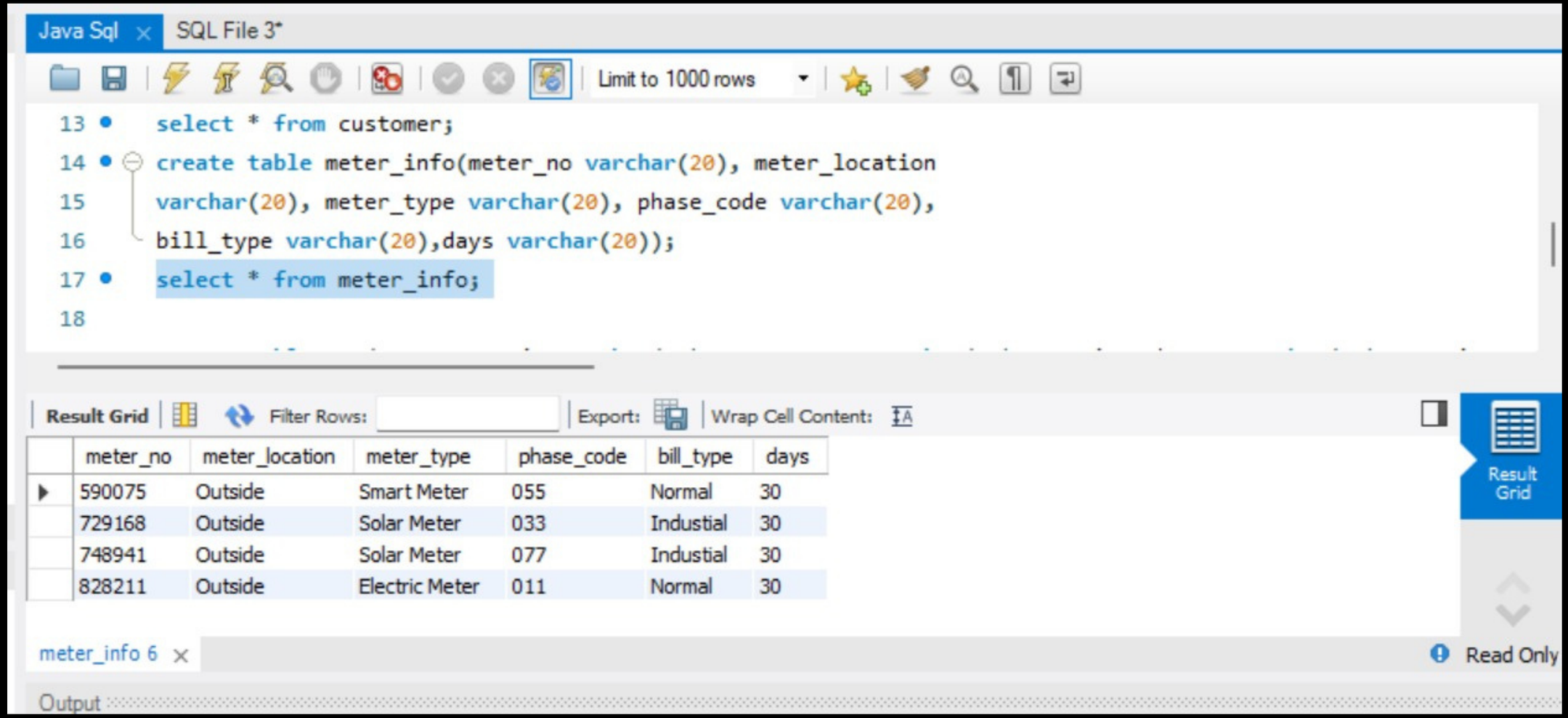
Pages 1 To 1

Copies

Number of copies: 1

Collate


Java Sql
SQL File 3*
Limit to 1000 rows
20 • insert into tax values('9','47','22','57','6','18');
21 • select * from tax;
22
23 • create table bill(meter_no varchar(20), month varchar(30), units varchar(20), totalbill varchar(20), statu
24 • select * from bill;
25
Result Grid
Filter Rows:
Export:
Wrap Cell Content:
meter_no month units totalbill status
729168 January 120 1230 Paid
748941 October 125 1275 Not Paid
748941 January 178 1752 Paid
590075 January 12 258 Not Paid
590075 June 250 2400 Not Paid
bill 2
Result Grid
Read Only


Java Sql
SQL File 3*
Limit to 1000 rows
13 select * from customer;
14 create table meter_info(meter_no varchar(20), meter_location
15 varchar(20), meter_type varchar(20), phase_code varchar(20),
16 bill_type varchar(20),days varchar(20));
17 select * from meter_info;
18
Result Grid
Filter Rows:
Export:
Wrap Cell Content:
meter_no meter_location meter_type phase_code bill_type days
590075 Outside Smart Meter 055 Normal 30
729168 Outside Solar Meter 033 Industial 30
748941 Outside Solar Meter 077 Industial 30
828211 Outside Electric Meter 011 Normal 30
Result Grid
meter_info 6
Read Only
Output

Conclusion

- We used Net Beans IDE to execute the program
- We have Used Swing Graphical User Interface Toolkit
- We imported SQL connecter package to connect SQL and JAVA
- The GUI we have designed is very user friendly and have many functionalities

Thank you!