PSNA College of Engineering and Technology

Computer science and engineering

ELECTRICITY BILLING SYSTEM

Mini project report

Submitted by :

- *J.MOHAMMED ASHIK*
- *S.MOHANA*
- *S.MOHANALAKSHMI*
- *K.MONIKA*

PROBLEM STATEMENT:

Java project idea: Design a electricity billing system using javaFX.

INTRODUCTION:

This system is named as the electricity bill management system.This system is a made to keep the records about the bills of the customers.The admin can manage all the accounts and the registered user like employees and the customers can only manage their own accounts.

This system help maintaing the bills and the payments.A different module is there for employees to check the customer's details if their job requires it. Admin,employees, and customers all have a different interfaces and different privileges according to their needs.

Like a customer can only manage his account and cannot see any details of other

customers,employees can see the details of all the customer's accounts and the admin can admin can manage all the accounts including the customers and

employees accounts.This system also has the option for customers to pay their electricity bills online mode.

Either through internet banking or by debit card. This system also has the feature to add and delete customer and employee's accounts in case a customer wants to cut the connection or an employee wants to leave the job.All the employees are divided into different departments according to their job profile and the customers are divided according to their wards.

ABSTRACT:

Java project idea: Electricity billing system is a web based project in java that

provides online platform for user to pay electricity bills .The system automates billing system and calculate the amount of money to be paid according to the units consumed in specific duration of time.

The customer will just feed the meter's readings into the system and software will generate the bill.This system will be helpful to both consumers and companies

OBJECTIVES:

The main objective of the electricity billing system is to manage the details of electricity bill store,record,customer. It manages all the information about the electricity board, customers,electricity.The purpose of the project is to build an applications program to reduces manual work for managing the electricity,bill,electricity board,connections. It tracks all the details about the store, record,customers.

JAVAFX:

JavaFX is a java library used to develp desktop application as well as Rich Internet Application(RIA).The application built in javaFX,can run on multiple paltforms including web,Mobile and Desktops.

JavaFX is intended to replace swing in java application as a GUI framework. However,it provides functionalities than swing. Like swing,javaFX also also provides its own components and doesn't depends upon the operating system .It is lightweight and hardware accelerated. It supports various operating sysytems including windows,Linux and Mac OS.

SCENE:

The javaFX Scene class is the container for all content in a scene graph.The

background of the scene is filled as specified by the fill property.The applications must specify the root Node for the scene graph by settings the root property.

PRIMARY STAGE:

The JavaFX stage class is the top level JavaFX container.The primary stage is constructed by the platform. Additional Stage objects may be construted by the constructed and modified on the JavaFX application thread.

Layouts used :

1.Grid pane

Explanation :

1. If we use this layout in our application,all nodes are added to it & arranged in the form of grid of rows and columns.

2. The class named Gridpane of the package **javafx.scene.layout** represent the gridpane.

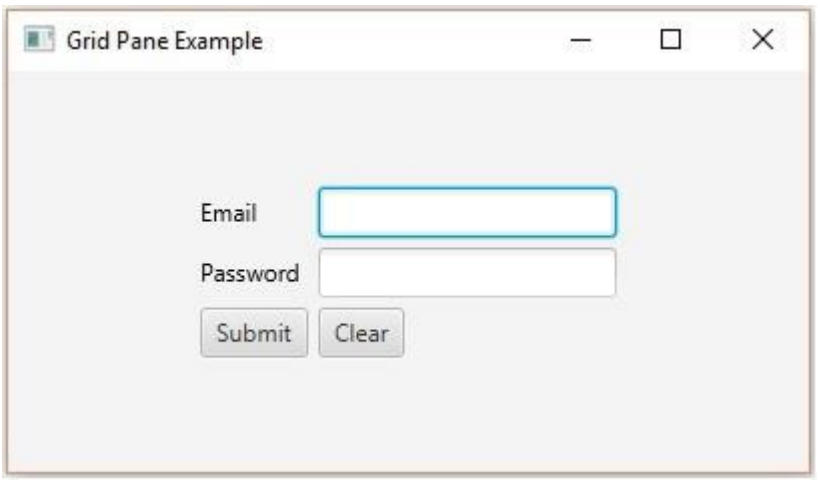
3.Properties of this class are,alignment, hgap,vgap,gridLinesVisible.

COMMAND:

Javac GridPaneExample.java

Java GridPaneExample

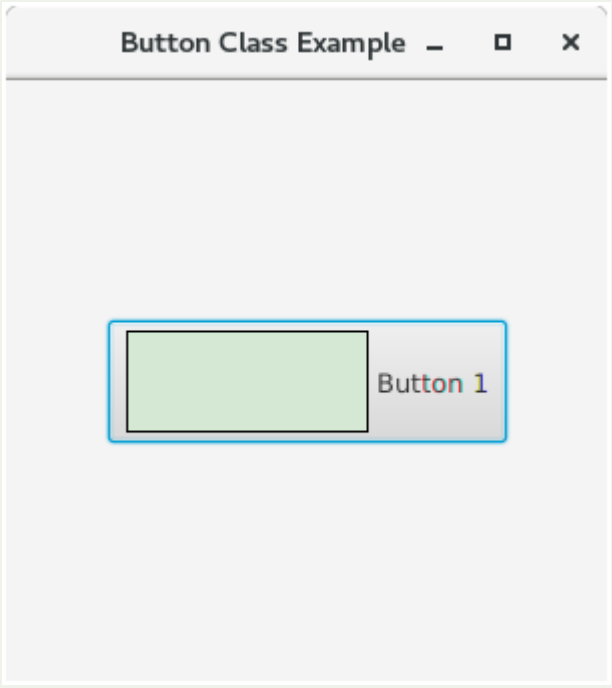
Example:

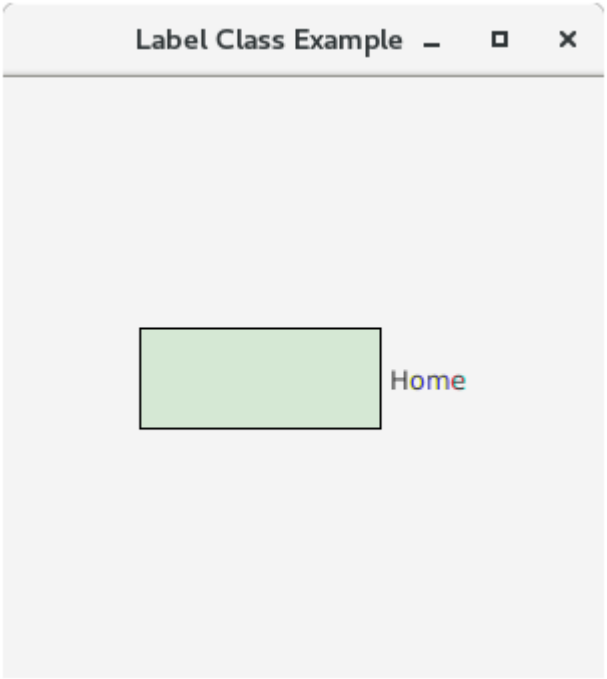


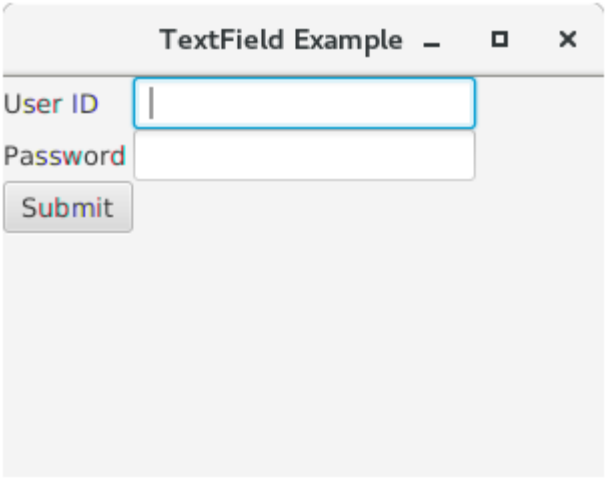
JAVAFX UI CONTROLS:

The package **javafx.scene.control** provides all the necessary classes for the UI components like Button, Label, etc.

JAVAFX UI CONTROLS USED:

BUTTON	Button is a component that controls the function of the application. Button class is used to create a labelled button. Button btn = new Button("My Button"); 

LABEL	Label is a component that is used to define a simple text on the screen. Typically, a label is placed with the node, it describes. **javafx.scene.control.Label** 
TEXTFIELD	Text Field is basically used to get the input from the user in the form of text. javafx.scene.control.TextField represents TextField **javafx.scene.control.TextField**

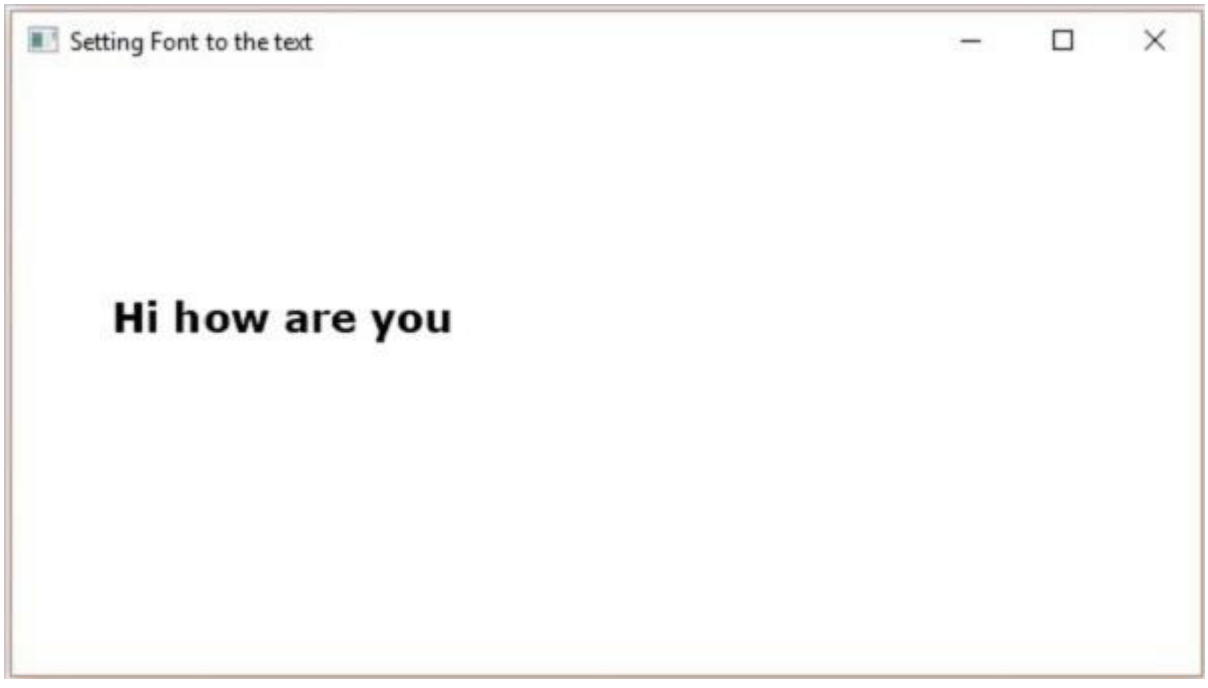


TEXT :

COMMAND:

javac TextExample.java

java TextExample



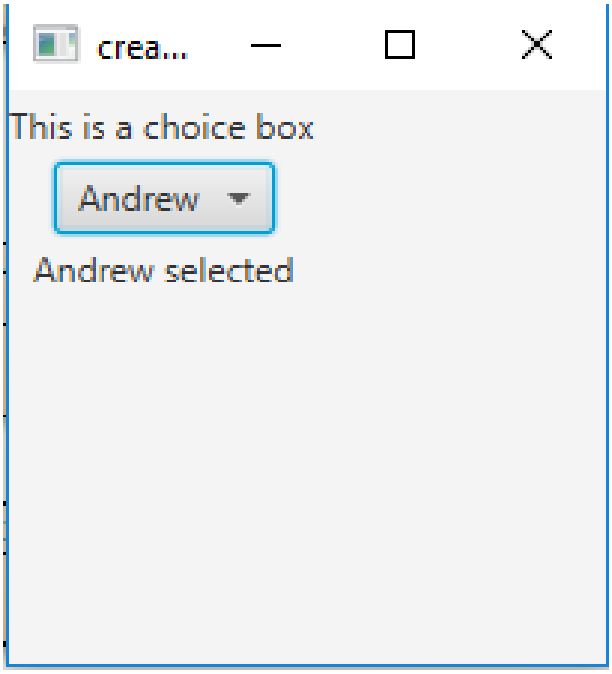
CHOICE BOX:

ChoiceBox is a part of the JavaFX package. ChoiceBox shows a set of items and allows the user to select a single choice and it will show the currently selected item on the top.

Commonly used methods:

method	**explanation**
getItems()	Gets the value of the property items.
getValue()	Gets the value of the property value.

method	explanation
hide()	Closes the list of choices.
setItems(ObservableList value)	Sets the value of the property items.
setValue(T value)	Sets the value of the property value.
show()	Opens the list of choices.



CODE :

```
package App;

// Importing all needed imports
import java.time.LocalDateTime;
import
java.time.format.DateTimeFormatter;
import java.util.Random;
import App.Compute;
import javafx.application.Application;
```

```
import javafx.event.ActionEvent;
import javafx.event.EventHandler;
import javafx.geometry.Pos;
import javafx.scene.Scene;
import javafx.scene.control.Button;
import
javafx.scene.control.ChoiceBox;
import javafx.scene.control.Label;
import javafx.scene.control.TextField;
import javafx.stage.Stage;
import javafx.scene.layout.GridPane;
import javafx.scene.paint.Color;
import javafx.scene.text.Font;
import javafx.scene.text.FontWeight;
import javafx.scene.text.Text;
```

```
public class ElectricityBill extends
Application {
    int units, amt;
    String cu, names, bc, time;

    @Override
    public void start(Stage
primaryStage) throws Exception {

        // Creating labels

        Label cid = new
Label("Customer id");
        Label name = new
Label("Name");
```

```
        Label unit = new Label("Units
Consumed");
        Label cycle = new Label("Billing
cycle");
        Label amount = new
Label("Amount");


        // Creating Textfields
        TextField tf1 = new TextField();
        tf1.setPromptText("enter
customer id");
        TextField tf2 = new TextField();
        tf2.setPromptText("enter your
name");
        TextField tf3 = new TextField();
```

```
        tf3.setPromptText("enter units consumed");
        TextField tf4 = new TextField();

        // Creating ChoiceBox
        ChoiceBox choiceBox = new ChoiceBox();

    choiceBox.getItems().add("Monthly");

    choiceBox.getItems().add("Bimonthly");

        // Creating Submit and Paynow Button
```

```
        Button btn1 = new
Button("Submit");
        Button button = new
Button("Paynow");

        // Creating Text
        Text txt = new Text();

    txt.setFont(Font.font("Verdana",
25));
        txt.setFill(Color.BLACK);
        Text txt1 = new Text();

    txt1.setFont(Font.font("Verdana",
25));
        txt1.setFill(Color.BLACK);
```

```
        Text txt2 = new Text();

    txt2.setFont(Font.font("Verdana",
25));
        txt2.setFill(Color.BLACK);
        Text txt3 = new Text();

    txt3.setFont(Font.font("Verdana",
25));
        txt3.setFill(Color.BLACK);
        Text txt4 = new Text();

    txt4.setFont(Font.font("Verdana",
25));
        txt4.setFill(Color.BLACK);
```

```
        Text txt5 = new Text();


    txt5.setFont(Font.font("Verdana",
25));
        txt5.setFill(Color.BLACK);


        // Getting the SystemDate and
Time
        DateTimeFormatter dtf =
DateTimeFormatter.ofPattern("dd/M
M/yyyy HH:mm:ss");
        LocalDateTime now =
LocalDateTime.now();
        time = dtf.format(now);
        btn1.setOnAction(new
EventHandler<ActionEvent>() {
```

```
            // Action on pressing the
Submit button
            @Override
            public void
handle(ActionEvent arg0) {

    System.out.println("Payment
successfull");

                // Getting units consumed
from Textfield
                units =
Integer.parseInt(tf3.getText());

                // Calculating Amount to
be paid
```

```
                Compute c = new
Compute();
                c.calculate(units);

    System.out.println("Amount " +
c.billpay);

                // Setting Amount

    tf4.setText(String.valueOf(c.billpay)
);

                // Getting Customer Id ,
Name and choice
                cu = tf1.getText();
                names = tf2.getText();
```

```
                bc = (String)
choiceBox.getValue();


                // Setting Text to
Textfields
                txt.setText("Customer id :
" + cu);
                txt1.setText("Name : " +
names);
                txt2.setText("Units
consumed : " + units);
                txt3.setText("Cycle : " +
bc);
                txt4.setText("Amount : "
+ c.billpay);
                txt5.setText("Date of
Payment : " + time);
```

```
            }
        });

        // Creating gridPane for first
scene
        GridPane gp = new GridPane();
        Scene scene = new Scene(gp,
800, 400);
        gp.setAlignment(Pos.CENTER);

        // Adding TextField and buttons
to GridPane
        gp.addRow(0, cid, tf1);
        gp.addRow(1, name, tf2);
        gp.addRow(2, unit, tf3);
        gp.addRow(4, btn1);
```

```
        gp.addRow(3, cycle, choiceBox);
        gp.addRow(5, amount, tf4);
        gp.add(button, 1, 6);

        // Creating gridPane for second
scene
        GridPane gp1 = new GridPane();
        Scene scene1 = new Scene(gp1,
1000, 600);

        // Generating Random number
for Bill number
        Random random = new
Random();
        Text txt6 = new Text("\n
Electricity Bill \n ");
```

```
        Text txt8 = new Text("\nBill no :
" + random.nextInt(1000));

        // Setting Font

    txt8.setFont(Font.font("Verdana",
25));

    txt6.setFont(Font.font("Verdana",
FontWeight.BOLD, 50));
        txt6.setFill(Color.BLUE);
        txt8.setFill(Color.BLACK);

        // Creating Text
        Text txt7 = new Text("\n
Payment Successfull! \n Thankyou
Visit Again!!!\n");
```

```
    txt7.setFont(Font.font("Verdana", 25));
        txt7.setFill(Color.BLUE);

        // Creating Go Back
        Button button2 = new Button("Go Back");
        gp1.add(button2, 0, 0);

        // Adding all Texts to gridPane in second scene
        gp1.setAlignment(Pos.CENTER);
        gp1.addRow(1, txt6);
        gp1.addRow(2, txt8);
        gp1.addRow(3, txt);
```

```
        gp1.addRow(4, txt1);
        gp1.addRow(5, txt2);
        gp1.addRow(6, txt3);
        gp1.addRow(8, txt4);
        gp1.addRow(7, txt5);
        gp1.addRow(9, txt7);

        // Setting action on Paynow
Button
        button.setOnAction(e ->
primaryStage.setScene(scene1));

        // Setting Action on Go Back
Button
        button2.setOnAction(e ->
primaryStage.setScene(scene));
```

```
        // Setting scene on Stage
        primaryStage.setScene(scene);

    primaryStage.setTitle("Electricity
Bill");
        primaryStage.show();


    }


    public static void main(String[]
args) {
        launch(args);
    }
}
```

CALCULATING AMOUNT CODE :

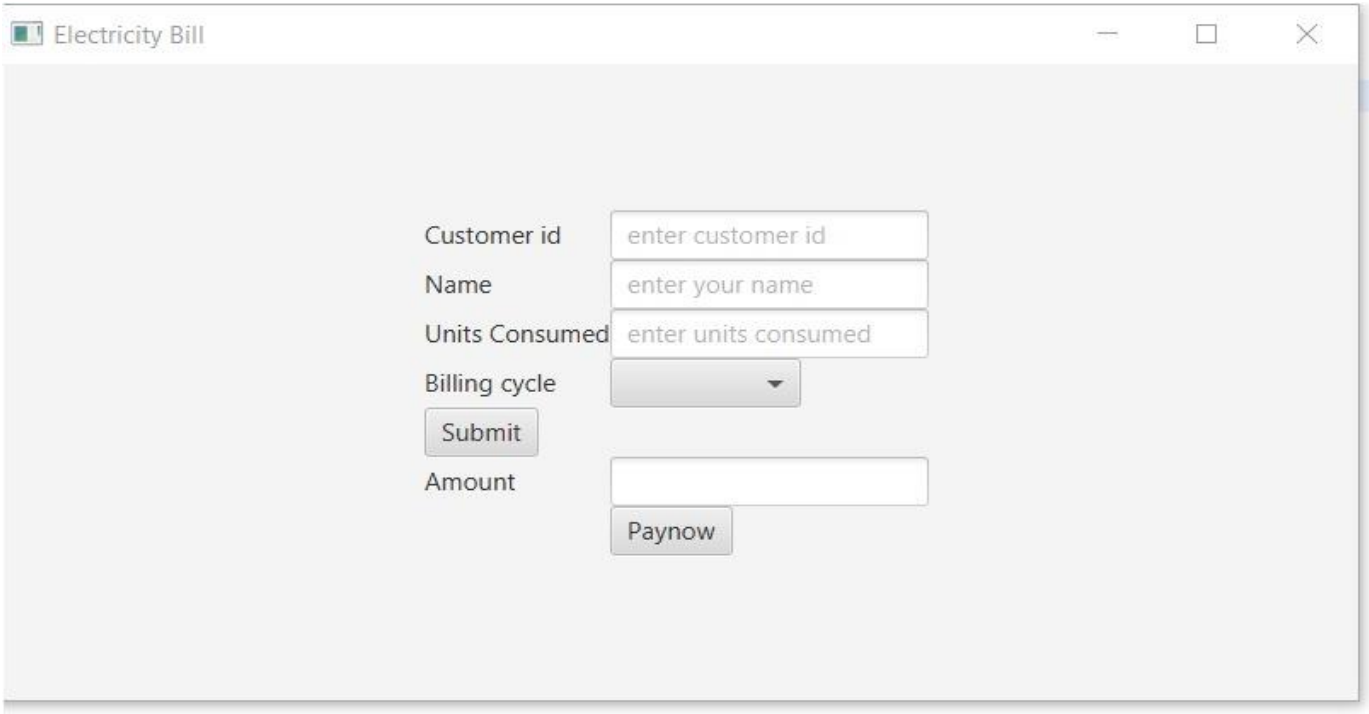
```
package App;
class Compute extends ElectricityBill
 {
    double billpay = 0;
    public void calculate(int units)
    {
        // calculating amount to be paid according to units consumed
        if (units <= 100)
         {
            billpay = units * 0;
          }
        else if (units <= 200)
         {
```

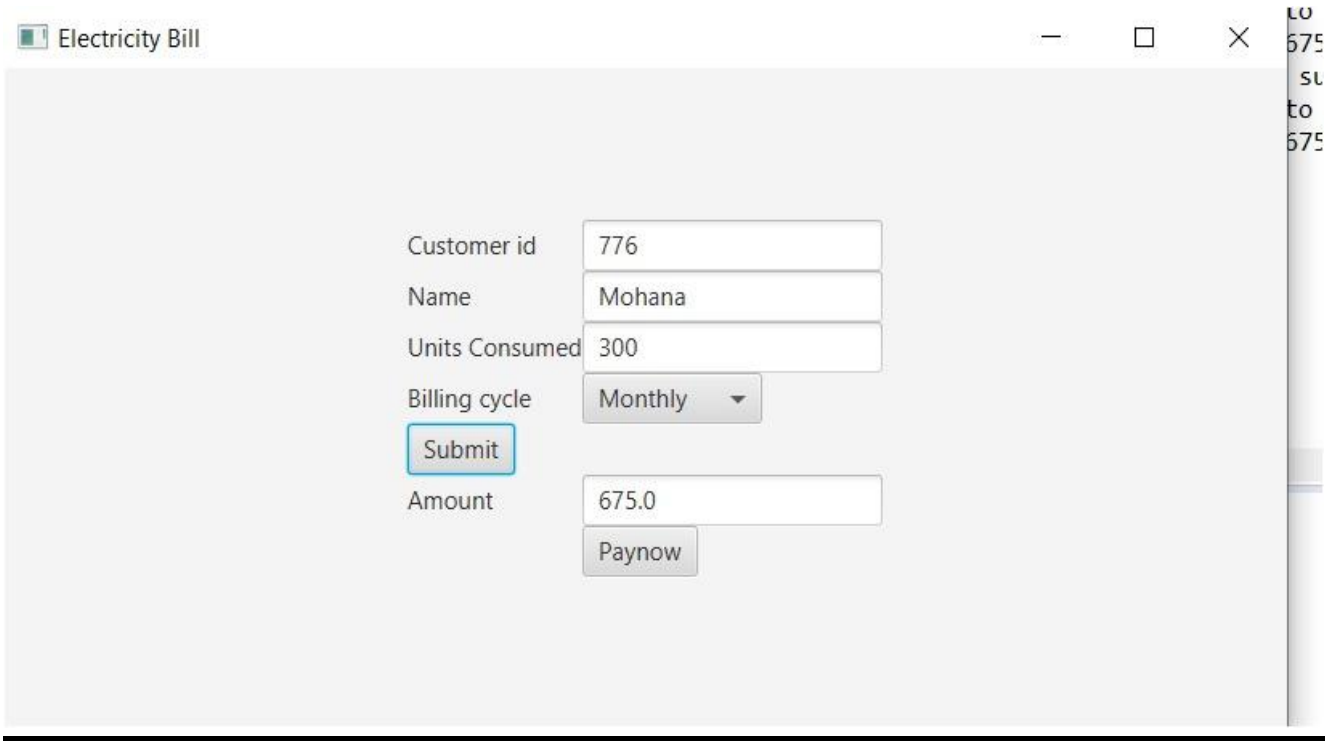
```
            billpay = (units - 100) * 2.25  ;
          }
          else if (units <= 400)
          {
            billpay =(units - 200) * 4.5 +
100 * 2.25;
          }
           else if (units <= 500)
          {
            billpay =(units - 400) * 6 +
100 * 2.25+ 200 * 4.5;
           }
           else if ( units <= 600)
           {
             billpay =(units - 500) * 8 +
300 * 4.5 + 100 *6;
```

```
        }
        else if( units <= 800) {
            billpay =(units - 600) * 9 +
300 * 4.5 + 100 *6 + 100 * 8;
        }
        else if (units<= 1000)
        {
            billpay =(units - 800) * 10 +
300 * 4.5 + 100 *6 + 100 * 8 + 200 * 9;
        }
        else if (units > 1000)
        {
            billpay = (units - 1000) * 10 +
300 * 4.5 + 100 *6 + 100 * 8 + 200 * 9+
200 * 10;
        }
```

```
        System.out.println("Amount to
pay : " + billpay);
    }
}
```

OUTPUT :




Electricity Bill
Customer id
776
Name
Mohana
Units Consumed
300
Billing cycle
Monthly
Submit
Amount
675.0
Paynow


Electricity Bill
Go Back
Electricity Bill
Bill no : 165
Customer id : 776
Name : Mohana
Units consumed : 300
Cycle : Monthly
Date of Payment : 20/11/2022 14:12:38
Amount : 675.0
Payment Successfull!
Thankyou Visit Again!!!

THANK YOU!!!