***TABLE OF CONTENT***

|  |  |  |
| --- | --- | --- |
| ***Serial No*** | ***CONTENT*** | ***Page No*** |
| ***Section1*** | ***Username and Password for all Accounts*** | ***2*** |
| ***Section2*** | ***UML Class Diagram*** | ***3*** |
| ***Section3*** | ***Features Implemented in System*** | ***4*** |
| ***Section4*** | ***Explanation of Main/Fragments Section of code*** | ***5-6*** |
| ***Section5*** | ***Screenshots of System*** | ***7-13*** |
| ***Section6*** | ***Evidence of Testing*** | ***14*** |
| 1. ***Black Box Testing*** | ***14*** |
| 1. ***White Box Testing*** | ***15*** |
| 1. ***List of Bugs and Weakness*** | ***16*** |
| ***Section7*** | ***References*** | ***16*** |

***SECTION1:***

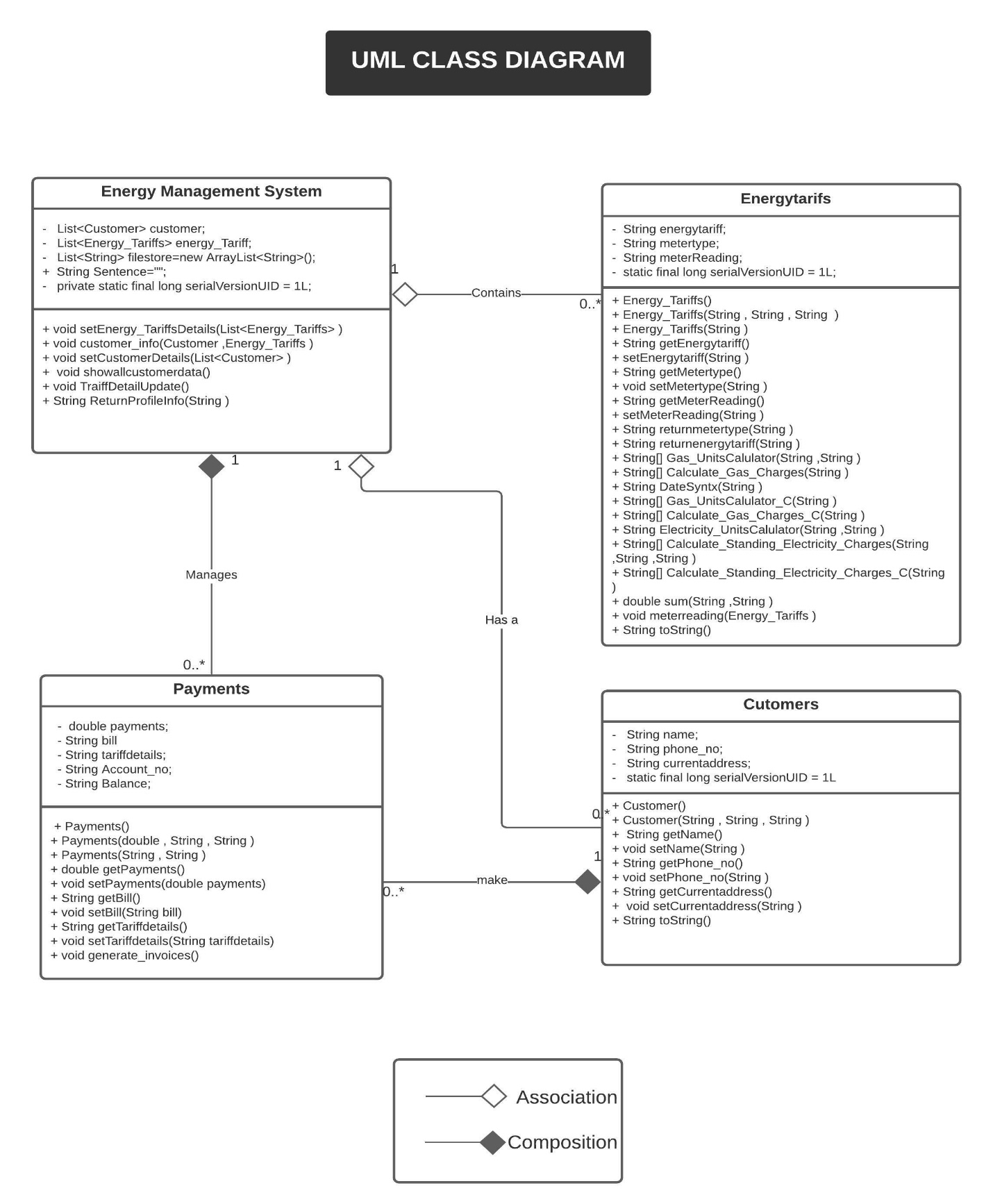
* ***Username and password for all relevant accounts (if implemented)***

Password implemented in the System is only for ***login*** of the system.

|  |  |
| --- | --- |
| **Username** | **Admin** |
| admin | admin |

***SECTION2:***

* ***UML Class Diagrams showing relationships between the main classes in the model***

****

***SECTION3:***

* ***A list of all the features implemented in a tabular format***

|  |  |  |
| --- | --- | --- |
| ***Feature*** | ***Implemented*** | ***Any comments*** |
| ***Dashboard*** | *Fully implemented* | *Error validation done* |
| ***Add new Customer*** | *Fully implemented* | *Error validation done* |
| ***Search for a customer by account name*** | *Fully implemented* | *Error validation done* |
| ***Show all customers registered on the system*** | *Fully implemented* | *Error validation done* |
| ***Energy Usage Chart*** | *Fully implemented* | *Error validation done* |
| ***Energy Bill*** | *Fully implemented* | *Error validation done* |
| ***Tariff Info*** | *Fully implemented* | *Error validation done* |
| ***Login to system*** | *Fully implemented* | *Error validation done* |
| ***Log out from the system*** | *Fully implemented* | *Error validation done* |
| ***Permanent storage of data and CRUD*** | *Fully implemented* | *Error validation done* |
| ***Input/update tariff information*** | *Fully implemented* | *Error validation done* |
| ***Creating pdf file*** | *Fully implemented* | *Error validation done* |

***SECTION4:***

* ***Explanation of the main sections/fragments of the code. Provide information that would be useful for another developer (not an end-user!) who may want to extend/maintain your system. You may want to refer to the class diagrams to explain the code.***
* **The main section/Fragment of code is our GUI main controller.**

This class is useful for another developer who wants to extend this system in the future.

Basically, it handles the GUI features, by using different classes.

The main GUI Features that are implemented by this class are

1. **Log in to system functionality.**

**public** **void** GoToDashboard(ActionEvent event) {

}

1. **Add customer** **functionality.**

**public** **void** AddNewCustomer(ActionEvent event) {

}

1. **Customer Profile** **functionality.**

**public** **void** SearchCustomer(ActionEvent event) {

}

1. **Show all Customer functionality.**

**public** **void** ShowAllCustomersInfo(ActionEvent event) {

}

1. **Energy Usage** **functionality.**

**public** **void** DrawChartOfEnergyUsage(ActionEvent event) {

}

1. **Gas Bill** **functionality.**

**public** **void** CalculateGasBillEType(ActionEvent event) {

}

1. **Electricity Bill** **functionality.**

**public** **void** CalculateElectricityBillButtons(ActionEvent event){

}

1. **Payment of Customers** **functionality.**

**public** **void** MakingInvioce(ActionEvent event) {

}

1. **Tariff Info** **functionality.**

**public** **void** SetTariffLabels(ActionEvent event) {

}

1. **Update Tariff Info** **functionality.**

**void** DashboardTabButtons(ActionEvent event) {

**else** **if** (event.getSource() == UpdateTariffInfo) {

AnchorUpdateTraiffInfo.toFront();

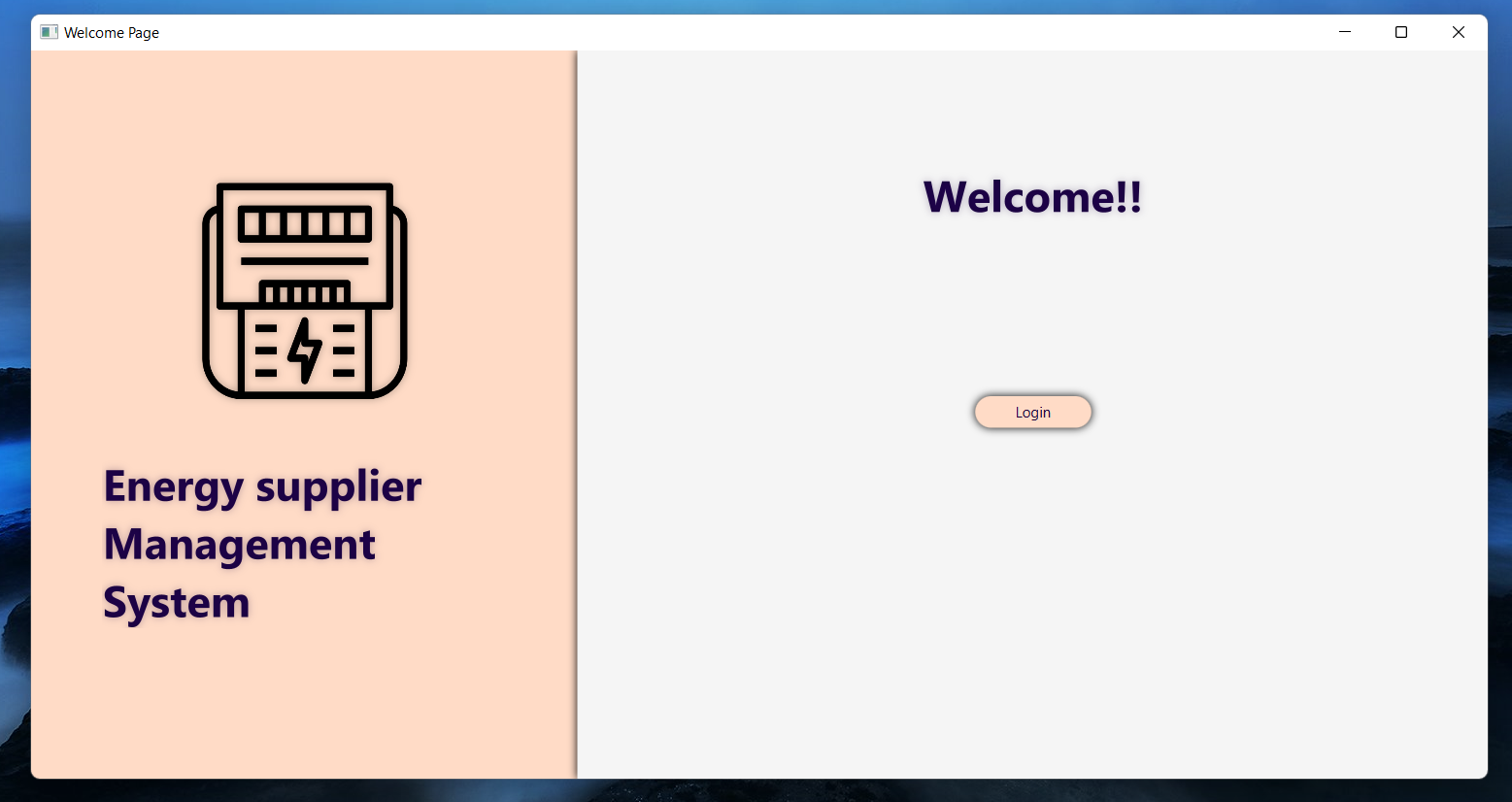
System.***out***.println("Anchor Update Traiff Info");

}

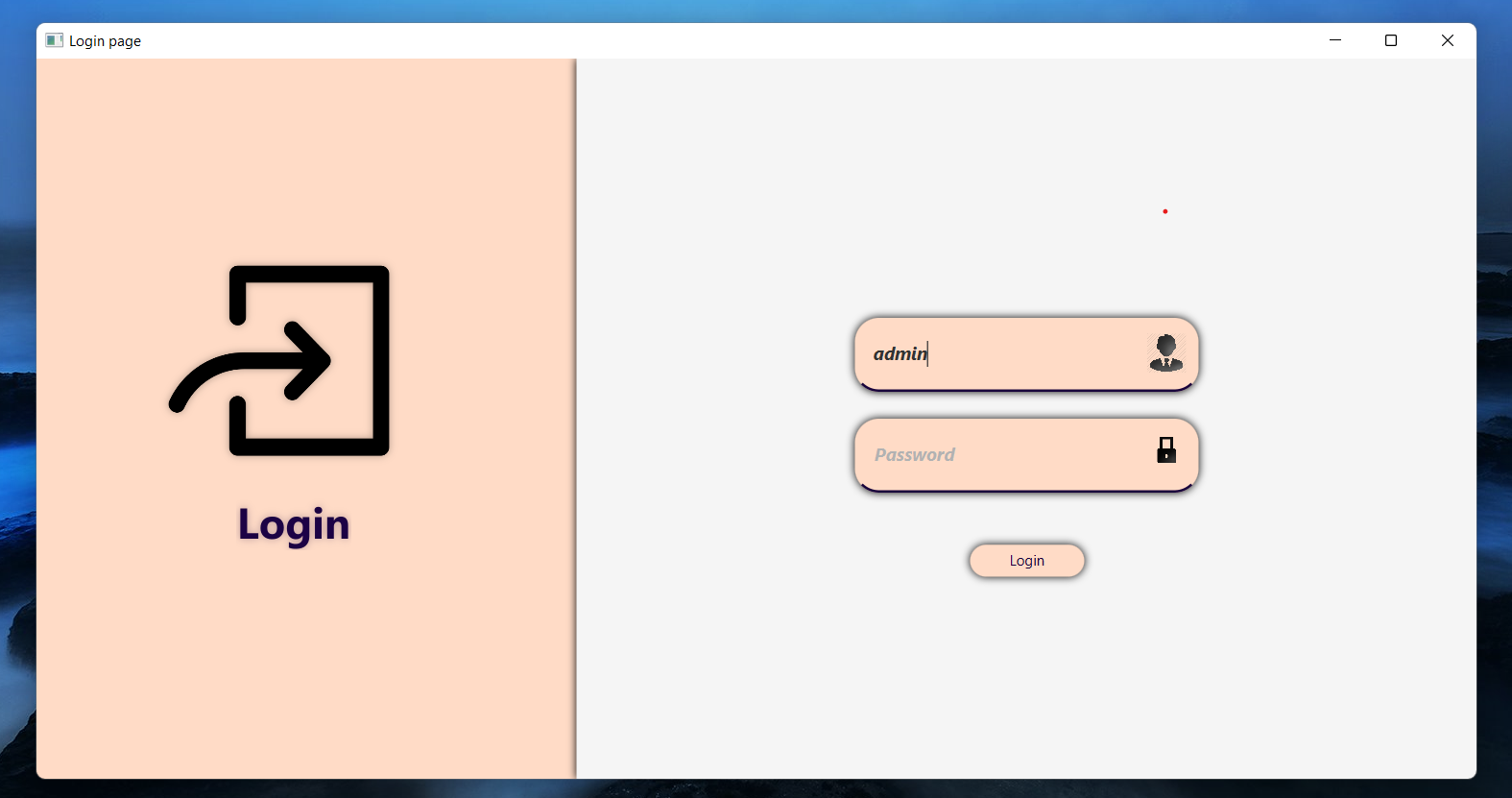
}

***SECTION5:***

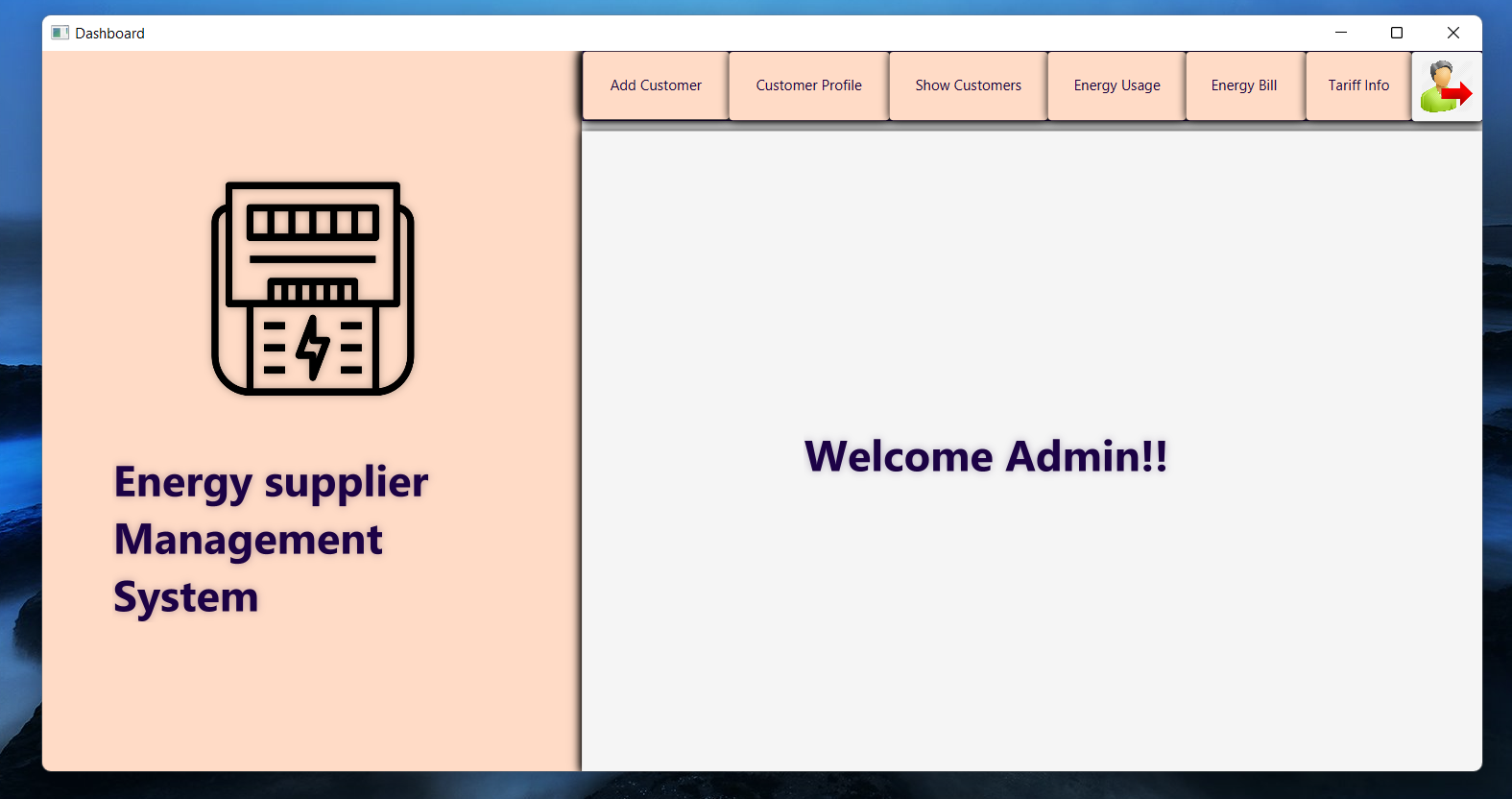
* ***Screenshots of the system showing all key features***
* ***Welcome Page:***



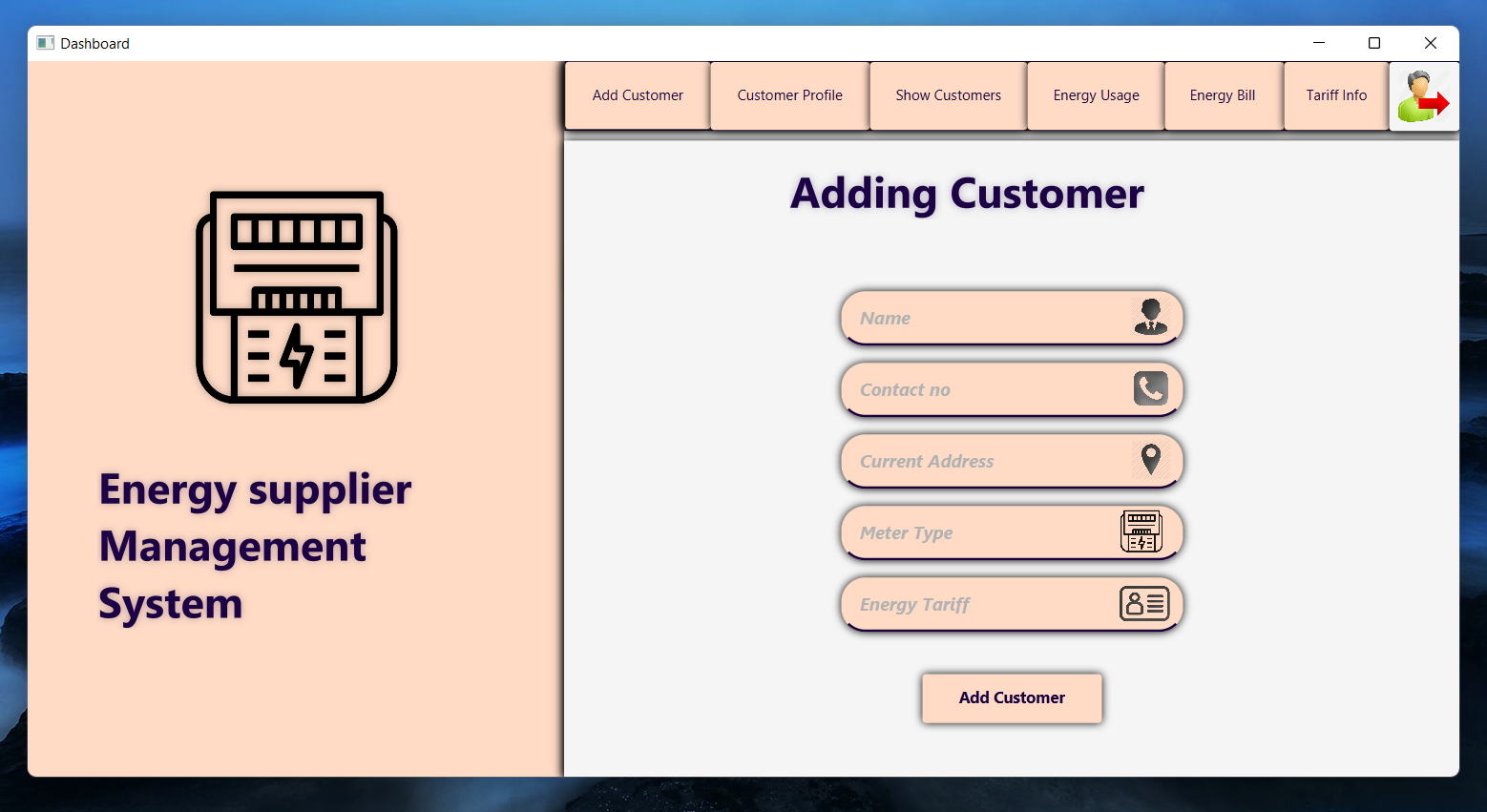
* ***Login Page:***



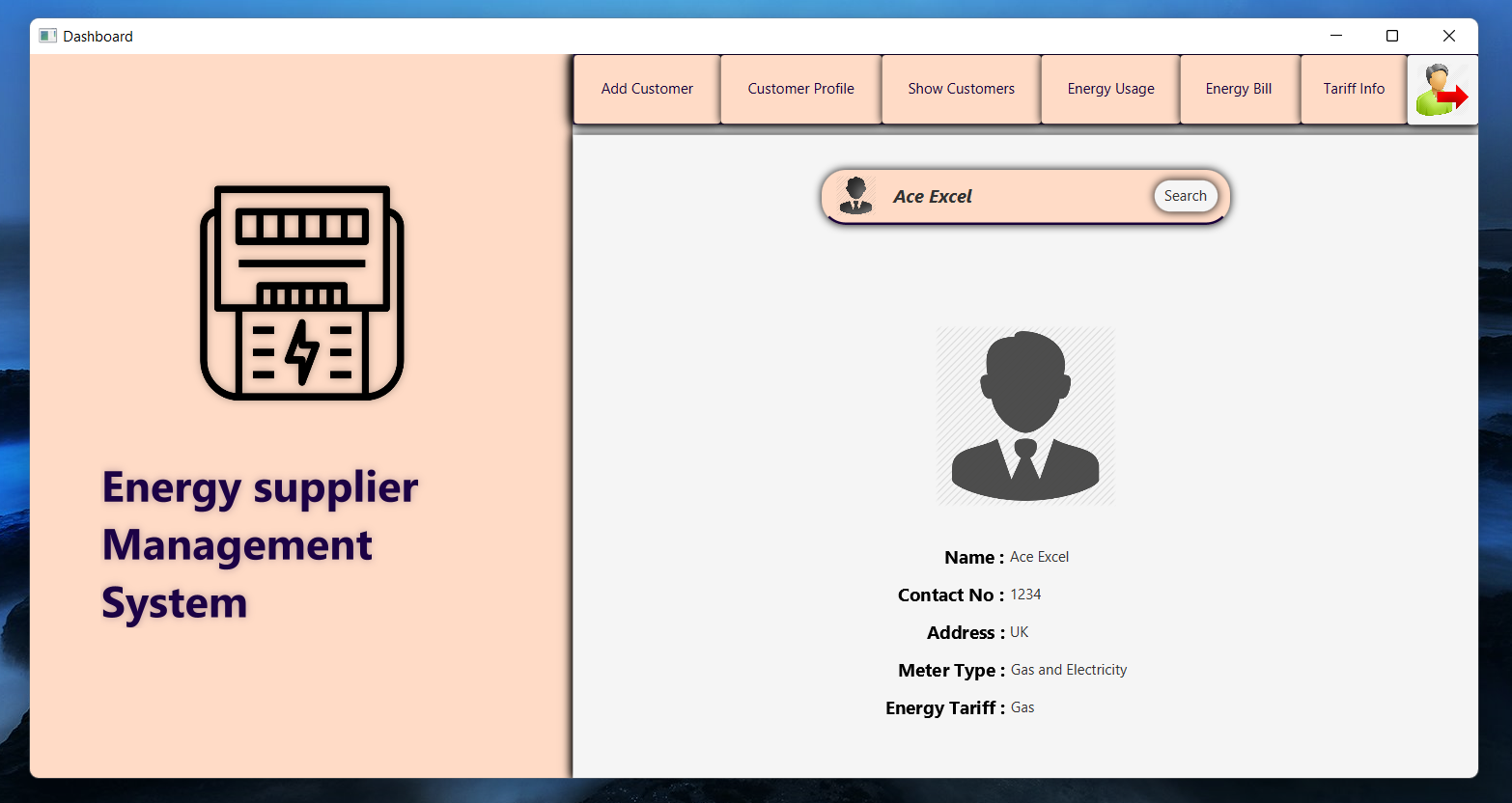
* ***Dashboard Page:***



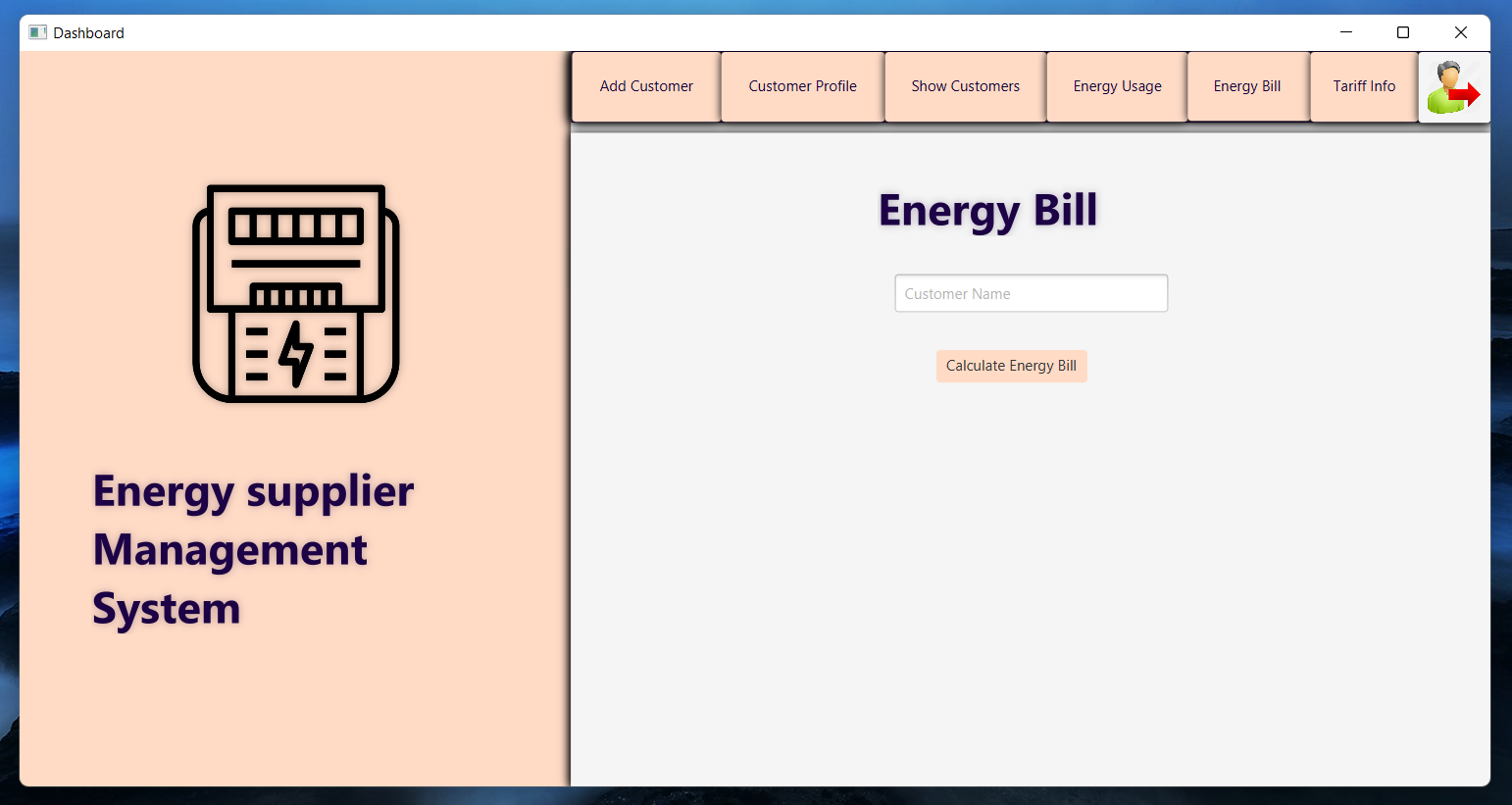
* ***Adding new Customer Page:***



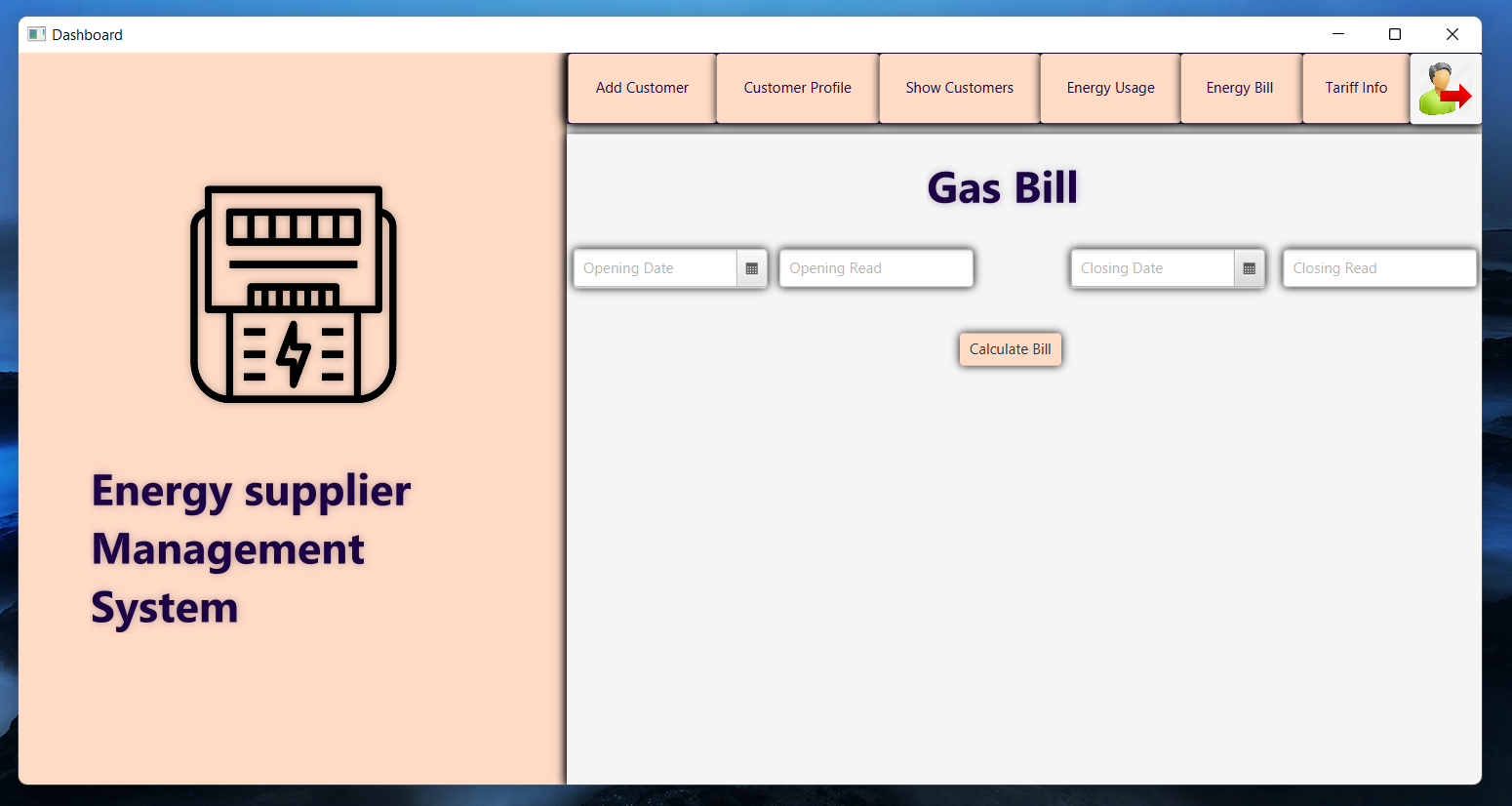
* ***Profile Page:***



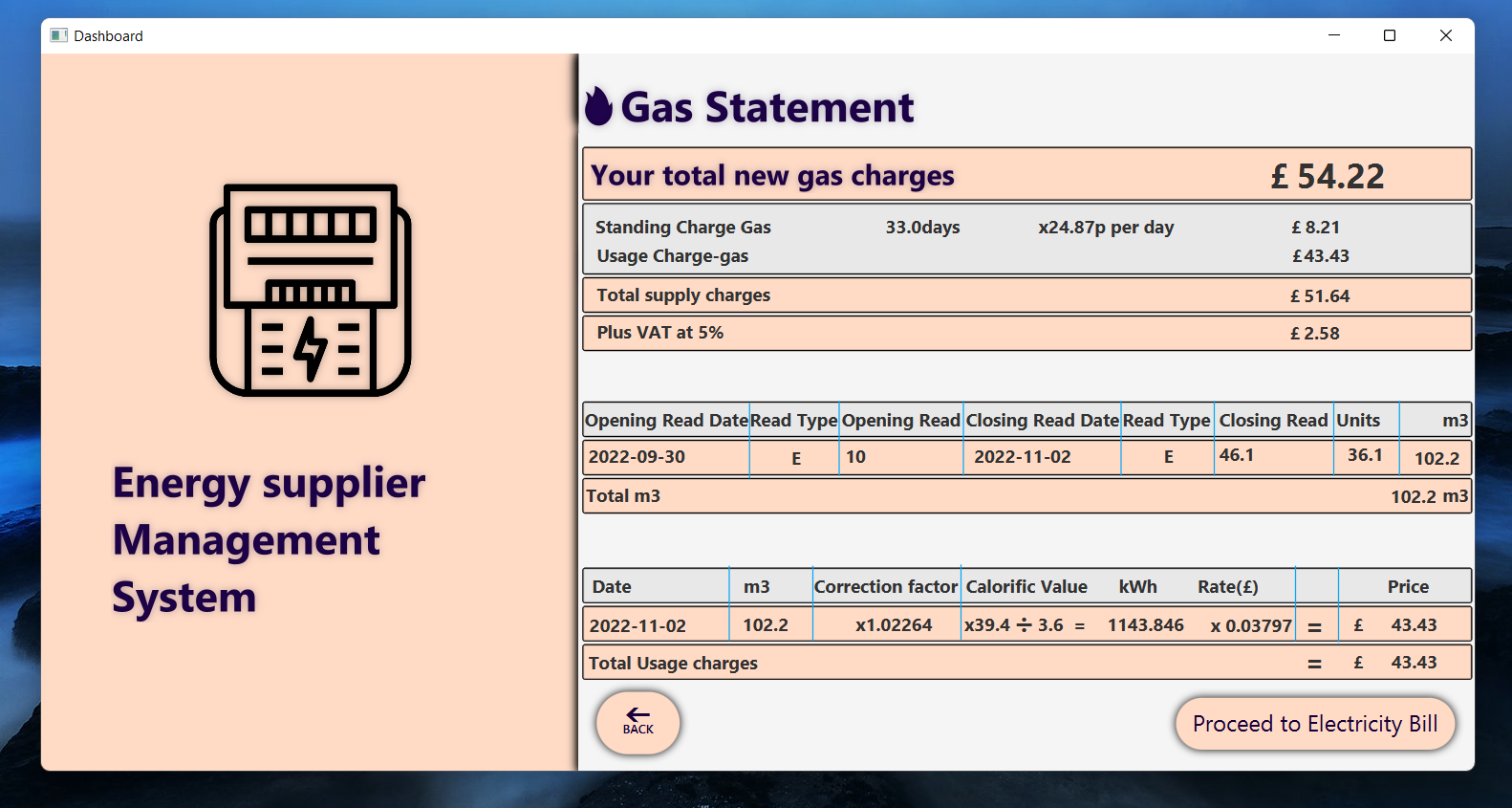
* ***Energy Bill Page:***



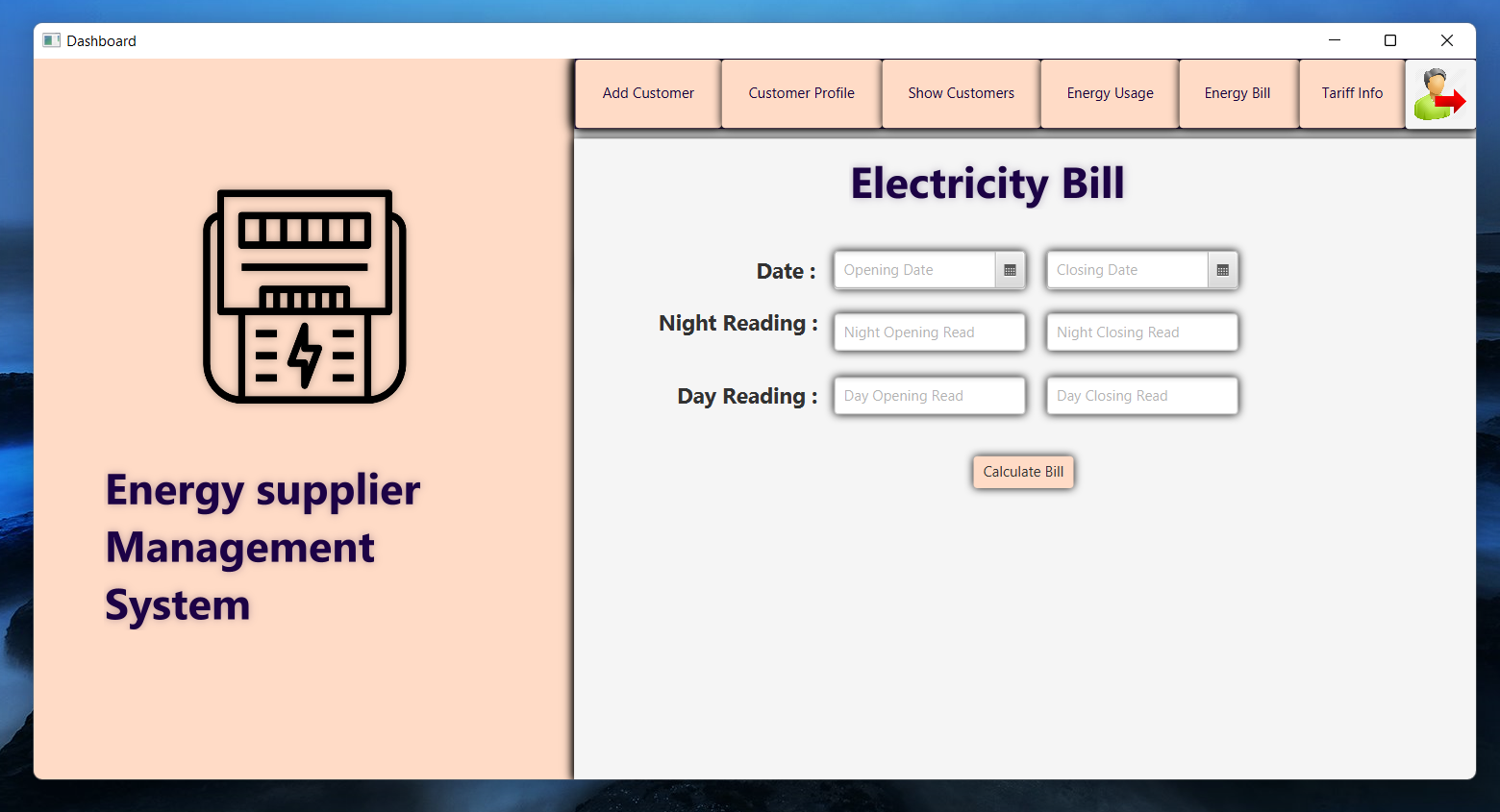
* ***Gas Bill Calculation Page:***



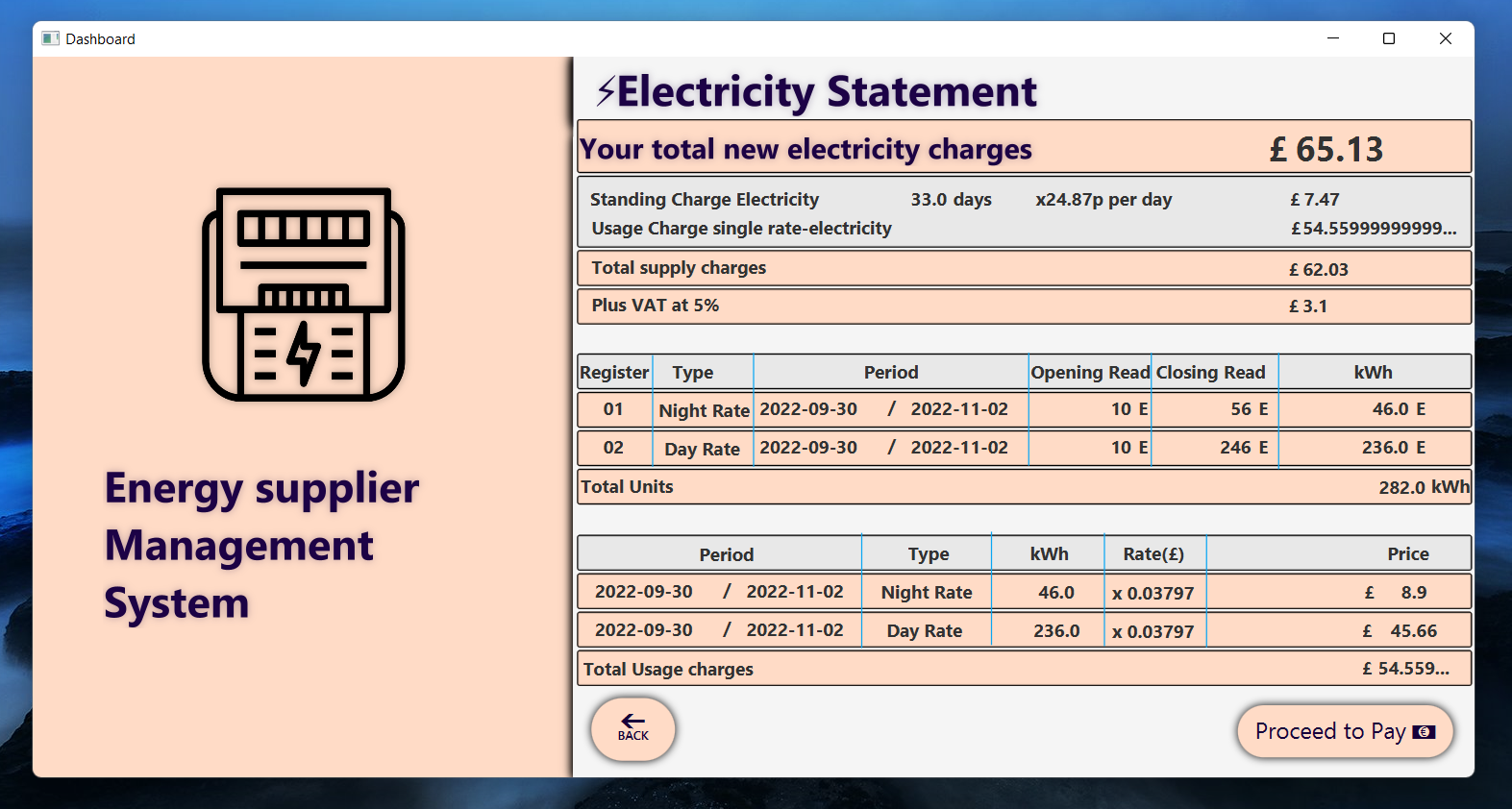
* ***Gas Bill Statement Page:***



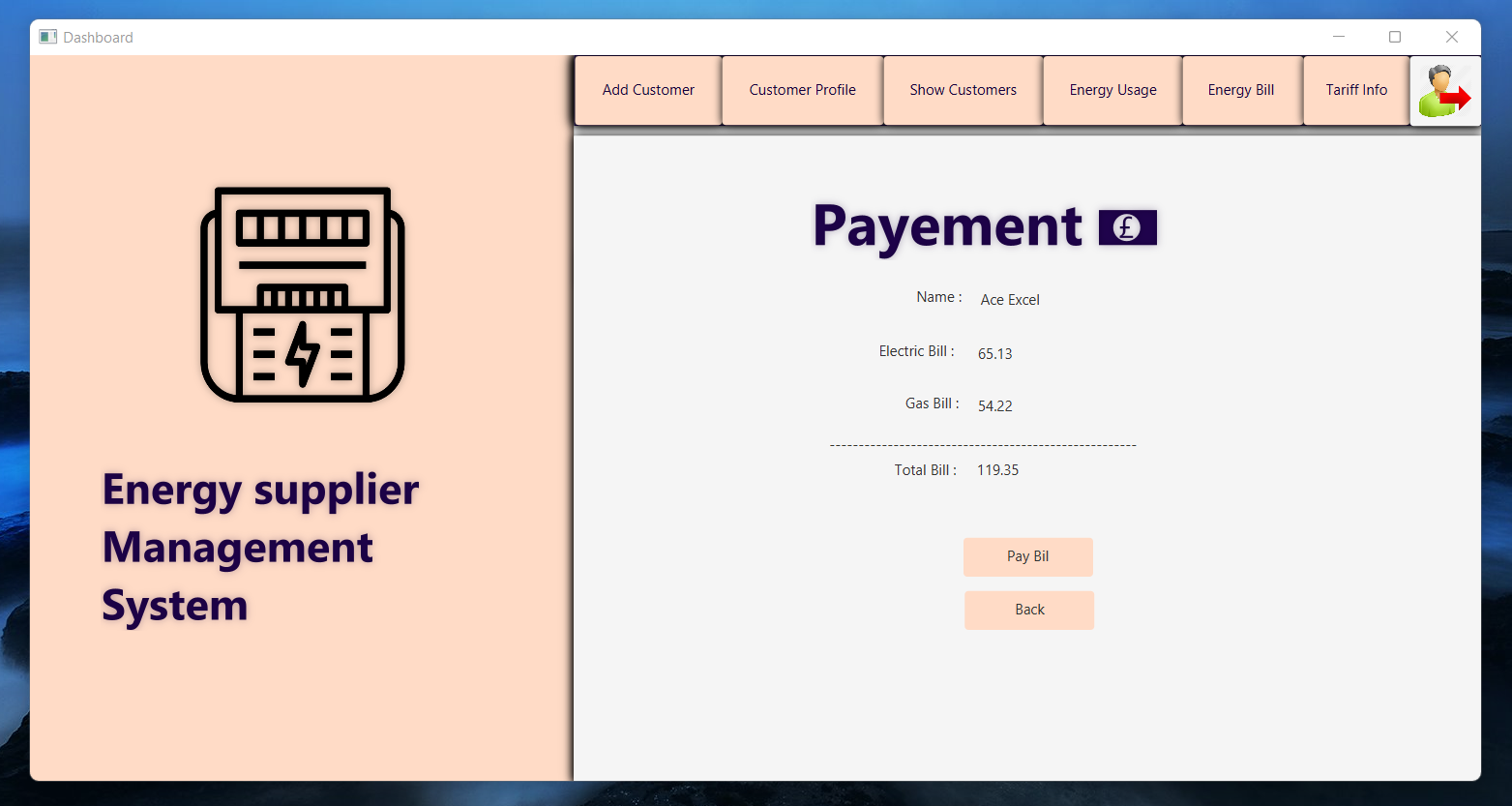
* ***Electricity Bill Calculation Page:***



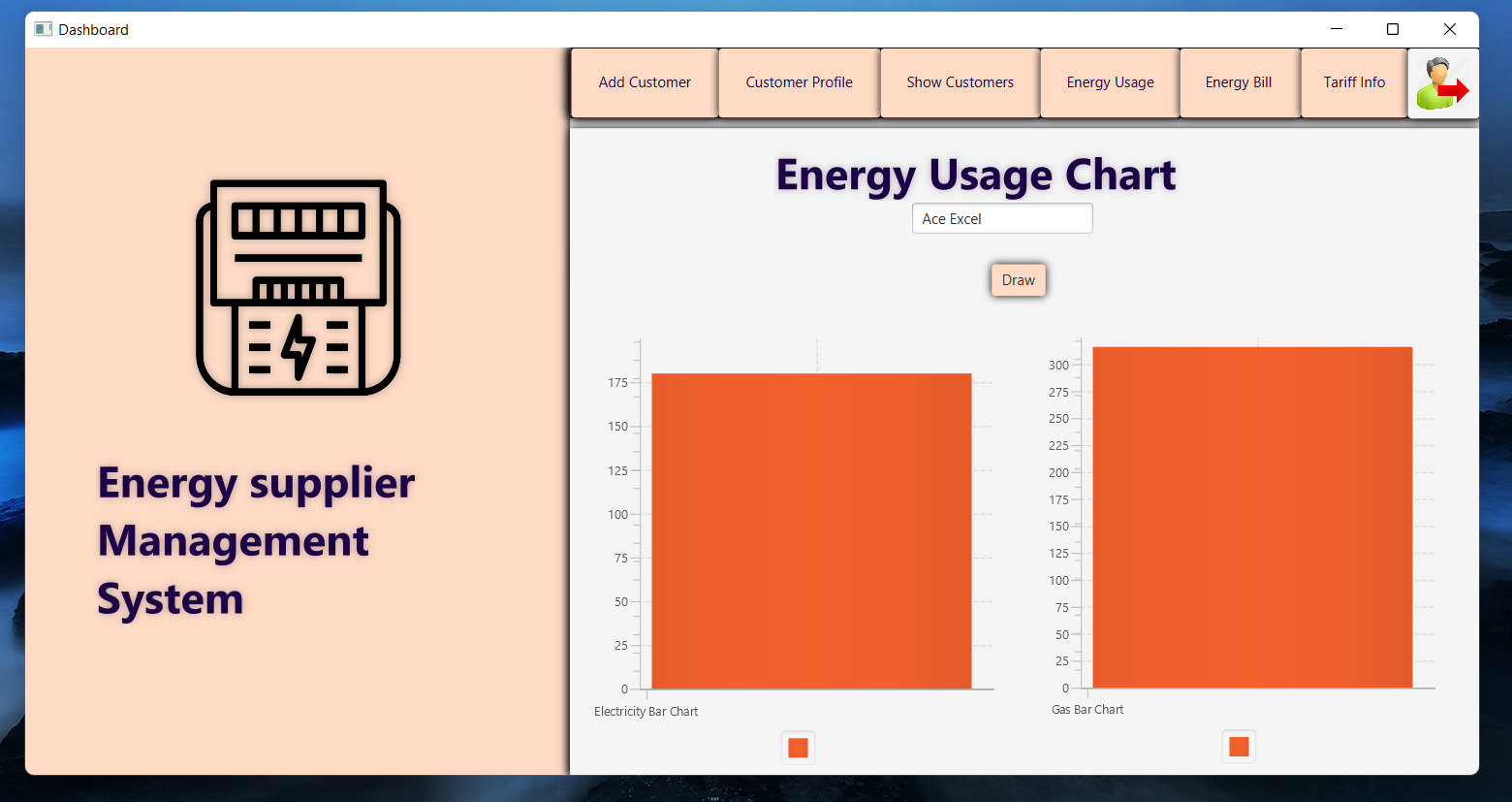
* ***Electricity Bill Statement Page:***



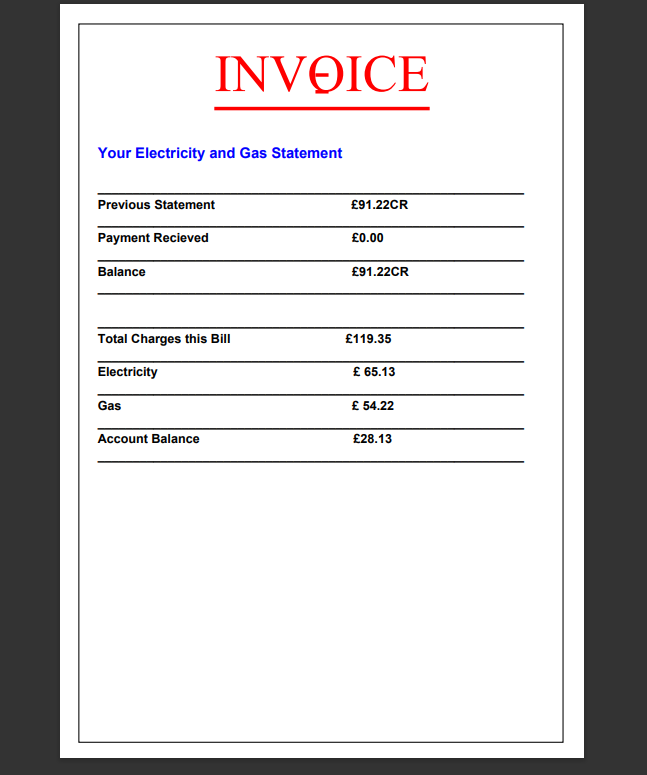
* ***Payment Page:***



* ***Energy Usage Chart Page:***



* ***Invoice of Energy Usage (Gas+Electricity) Pdf:***

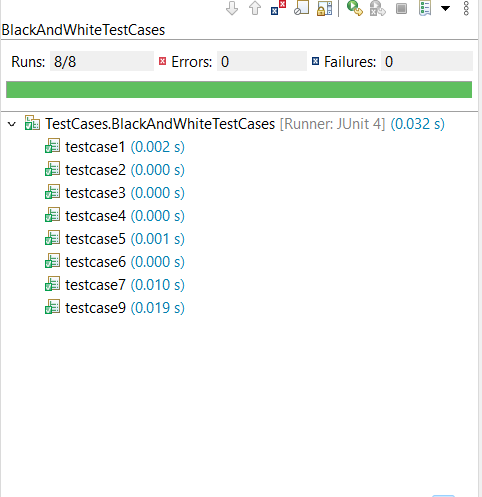


|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Test Case #** | **Test Title** | **Input Data** | **Expected Output** | **Actual Output** | **Result** |
| **Test-1** | Search for a Customer | Name | Specific customer found in system | Search Successfully | pass |
| **Test-2** | Search for a Customer | Name | Specific customer not found in system | No customer with this name exists | pass |
| **Test-3** | Search for the phone number of a customer | Name | Number of that customer found in the system | Found Successfully | pass |
| **Test-4** | Calculate Gas units by estimation | Opening read, closing read | Shows the units used | Showed successfully | pass |
| **Test-5** | Standing Gas Charges | Opening date, closing date, and days | Show BMI by calculating the result | Show the BMI of body and inform that it is normal | pass |

***SECTION6:***

* ***Evidence of Testing:***

1. ***Black box Testing*** **(*Test logs providing information of all the tests carried out (including any failed tests for functionality not implemented)***
2. ***White Box Testing (*Code Listing of the JUnit test case for at least two methods**.***)***



***Code of Two Test Cases:***

***TESTCASE1:***

@Test

**public** **void** testcase5() {// true case

**int** counter ;

File\_Handling test=**new** File\_Handling();

counter= test.Return\_Count\_MatchingLine("Customer.txt", "Khuram Ali");

String ph\_no = test.Return\_Balance("Phone\_no.txt", counter);

*assertNotEquals*("278543213",ph\_no);

}

***TESTCASE2:***

@Test

**public** **void** testcase6() {// true case

**int** counter ;

File\_Handling test=**new** File\_Handling();

counter= test.Return\_Count\_MatchingLine("Customer.txt", "Sahil

Parkash");

String ph\_no = test.Return\_Balance("Phone\_no.txt", counter);

*assertEquals*("0329846518",ph\_no);

}

1. ***List of any bugs and/or weaknesses in your system (if you do not think there are any, then say so). Bugs that are declared in this list will lose you fewer marks than ones that you do not declare.***

*As our system is perfect, so there are no bugs /or weaknesses in your system.*

***SECTION7:***

* ***References (use Harvard referencing):***

# References

*Shell Energy*. (2010). Retrieved from Shell Energy: https://help.shellenergy.co.uk/hc/en-us/articles/360001043338-Understanding-my-bill