Brief:

Design, implement and test a management system for an energy supplier using object-oriented principles in Java FX. Design should include class diagrams. Testing should include both white box (JUnit tests) and black box (test logs). The system should enable the supplier to record customer accounts, energy tariffs, generate customer invoices and so on. Here are the specific requirements:

Basic System Requirements:

The system must allow the supplier to:

1. Record details of a new customer (name, phone number, current address, energy tariff, meter type, etc.)

2. Display energy usage dashboard of a particular customer (e.g., meter readings, account summary (payments, bills, tariff details and so on)

3. Create a monthly invoice/bill for each customer. The bill should also show detailed calculations of energy charges (i.e., conversion of meter readings to charges)

4. Record a payment for each customer (i.e., paid/unpaid)

Enhancements (in order of importance – high to low):

Additional features that you may include are as follows:

1. Permanent storage of data and CRUD [Create, Read, Update and Delete] (object serialisation)

2. Input/Read energy meter readings from a file (object serialisation or text/csv files).

3. Search for a customer by account number or name

4. Input/update tariff information (i.e., add/modify new/existing tariff offered)

5. Display annual energy usage chart and predicted monthly usage

6. Send an email containing the invoice as an attachment (pdf file) to a customer

7. Login feature for the system

These additional features are only suggestions and are not exhaustive. You may include any other useful features relevant to this application. Use an agile methodology for software development.

**Deliverables:**

**All requirements (A, B and C below)**

A) **Technical Reprt**

The report should consist of the following sections (in the same order):

1. Username and password for all relevant accounts (if implemented)

2. UML Class Diagrams showing relationships between the main classes in the model

3. A list of all the features implemented in a tabular format. For example:

Feature Implemented (Partial/Full) Any comments

Record New Customer details Full No error validation.

Display Energy Usage Dashboard Full

4. 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 code.

5. Screenshots of the system showing all key features

6. Evidence of Testing:

a. Blackbox Testing: Test logs providing information of all the tests carried out (including any failed tests for functionality not implemented)

b. Whitebox Testing: Code Listing of the JUnit test case for at least two methods.

c. 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.

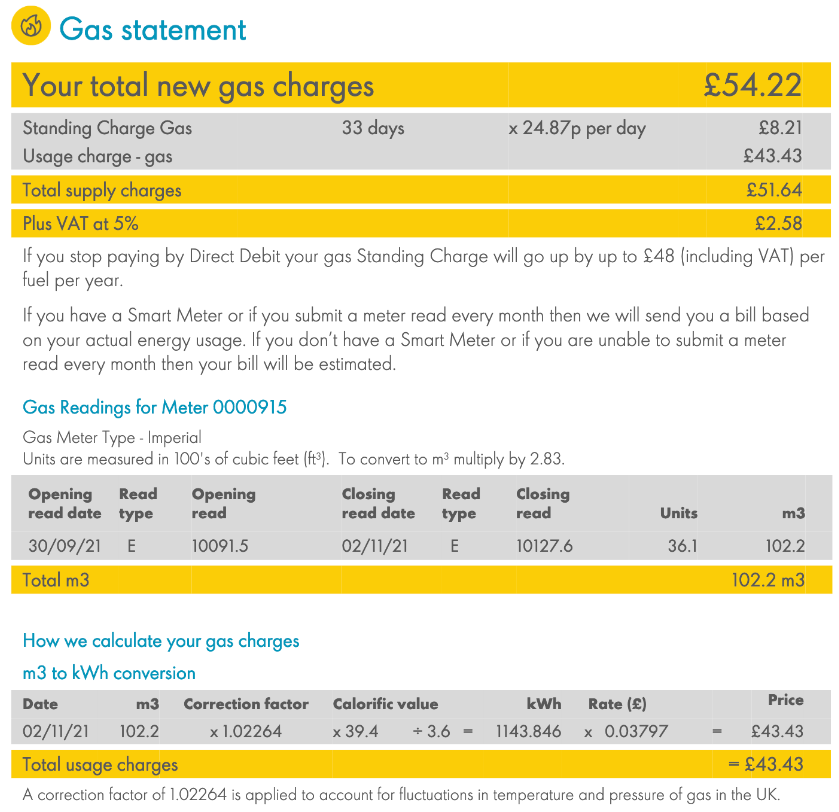
7. References (use Harvard referencing):

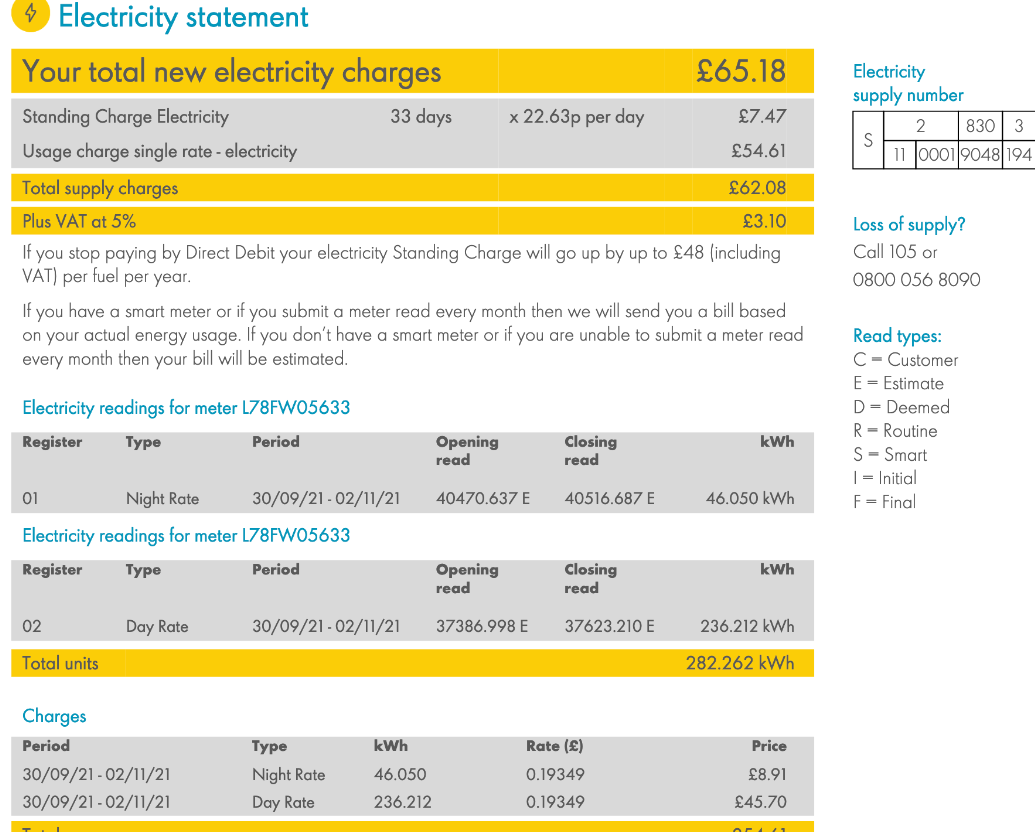
B) **Source Code**

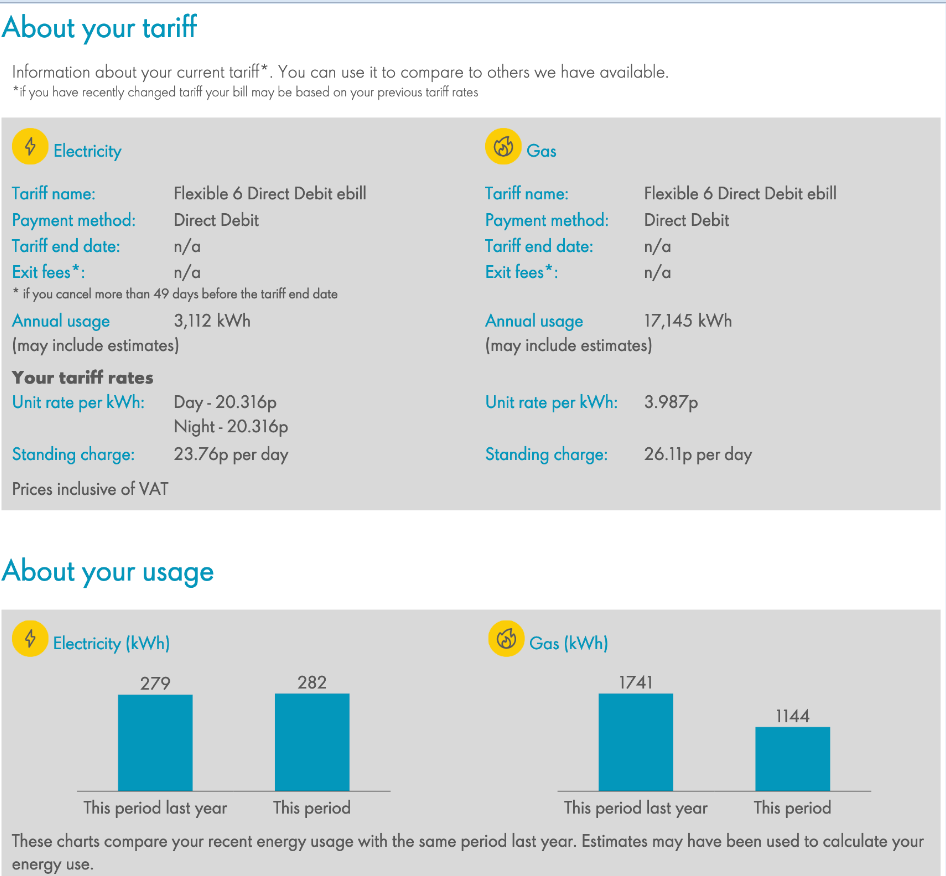
A commented full listing in a separate Word document named “Full Source Code Listing”.

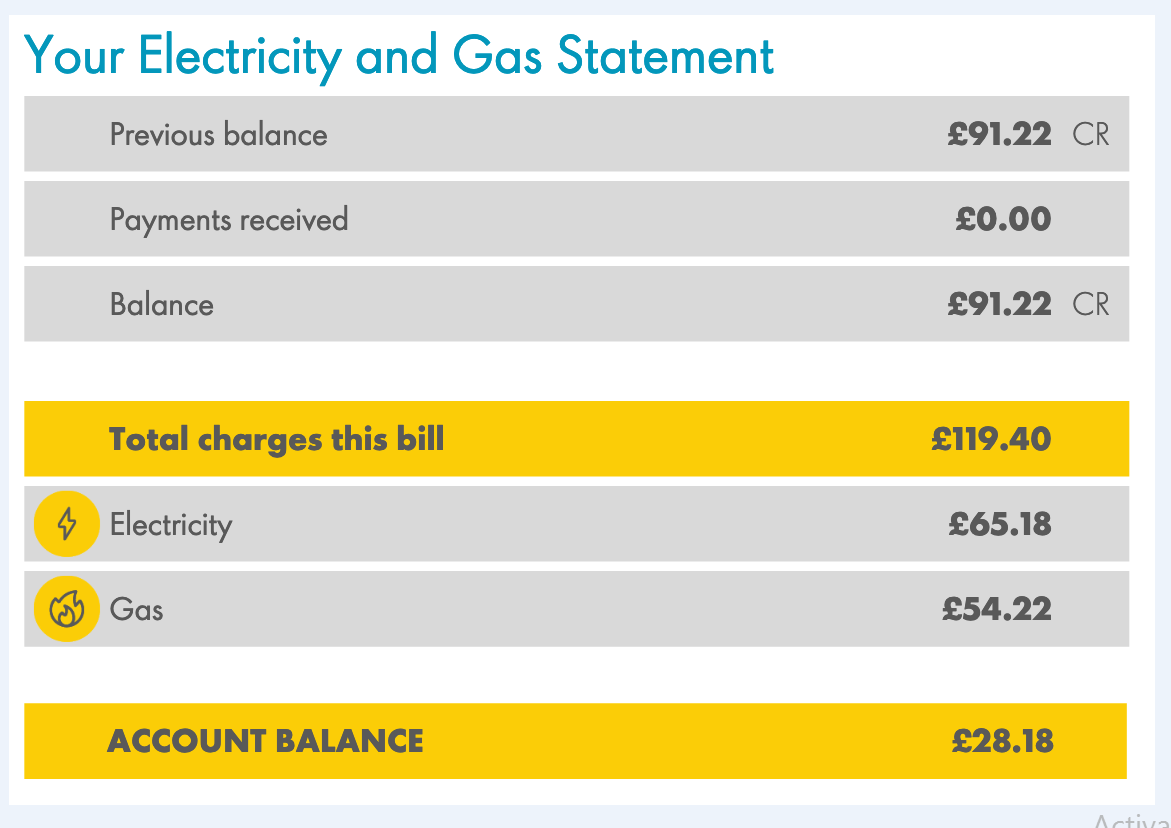
C) **Video Demonstration**

Samples









Please do in agile model

phase 1- home page, back page

phase 2 - registration, customer details etc etc