**SCC204 Coursework**

**Model Solutions**

**Question 1 (Class Diagram).**

* A model solution for the **PowerManagement** **System** class diagram is shown below. We recognise that there are viable variants to this solution. However, all potential solutions should be premised on:

(1) Good design principles

(2) Correct use the UML notation. For example, use of class rather than architectural or use case notation. Interfaces don’t have a state, therefore no attributes. Interfaces don’t implement other interfaces, and the “*implements*” relationship notation is a broken line. The direction of the “*extends*” and “*implements*” notation is important. There are two sides to an interface “implements” and “uses”

Diagram

Description automatically generated

**[1 mark** for each correct class diagram and **0.5 marks** for each correct relationship (**max 25 marks**)].

**Question2 (SOLID Principles)**

1. The **Single Responsibility Principle (SRP)** states that every class, module or function should have one responsibility or purpose in a program. The objective is to promote cohesion and minimise situations where a change in a class ripples through the rest of the system. Our class diagram promotes SRP in every class, for example the *ReportGenerator* class is solely responsible for report generation. The *BidirectionalPowerInverter* is only responsible for controlling power between various power consumers and sources.
2. The **Open/Closed Principle (OCP)** states that objects should be open for extension but closed for modification. This means that a class should be extendable without modifying the class itself. A great way to achieve this is to use interfaces instead of abstract classes to allow different implementations which can be easily substituted without changing the code that uses them. The interfaces are closed for modifications, and you can provide new implementations to extend the functionality of your software. The class diagram uses interfaces in several “volatile” areas of the system to allow for future evolution, for example power sources, consumers, renewables.
3. The **Liskov Substitution Principle (LSP)** states that the objects of a superclass should be replaceable with objects of its subclasses without breaking the application. This can be achieved by: (i) ensuring that overridden methods of a subclass accept the same input parameter values as the method of their superclass; (ii) ensuring that return value of a method of the subclass complies with the same rules as the return value of the method of the superclass. This can be seen in the structure of the *Appliance* and *Heating* classes, where objects the subclasses are can replace the objects of their superclass
4. The **Interface Segregation Principle (ISP)** states that no code should be forced to depend on methods it does not use. In the class diagram, this can be seen in separation of *PowerSource* and *PowerConsumer* interfaces, so that power sources and consumers don’t implement operations that they don’t use. See also the separation between *PowerSource* and *Renewables* interfaces.
5. The **Dependency Inversion Principle (DIP)** states that high level modules should not depend on low level modules; both should depend on abstractions. The Dependency Inversion Principle has of two parts:
   * High-level modules should not depend on low-level modules. Both should depend on abstractions.
   * Abstractions should not depend on details. Details should depend on abstractions.

The best abstraction to use for the DIP are interfaces. A good example of this is in the interaction between the *BidireactionalInverter* and, power sources and consumers. The *PowerConsumer* and *PowerSource* interfaces hide underlying details and allow for loosely coupling between the classes.

[**1 mark** for each principle used and **1 mark** for an example of where it has been applied]

**Question 3 (Assumptions)**

Sensible design assumptions are needed based on the particular design presented. It is important that these are not statements that are obvious from the problem specification. Below are some examples, but a variety of assumptions are possible.

1. System can have more than 1 battery, water cylinder, etc.
2. Only renewable energy can charge batteries.
3. Energy use is saved in the cloud.

[**1 mark** for each sensible assumption suggested]

**Question 4 (Reports)**

Four reports that cover these aspects

1. Renewable power sources consumption/generation report (weekly, monthly, etc)
2. Non-renewable power sources consumption report (daily, weekly, monthly, etc.)
3. Batteries report – total power generated versus total power consumed in week, month, etc. Trends in charge retention over weeks, months, etc.
4. Weekly/monthly heating power consumption

[**1 mark** for each sensible report suggested]