**INTI International College Penang School of Computing**

**3+0 Bachelor of Science (Hons) in Computer Science, in collaboration with Coventry University, UK 3+0 Bachelor of Science (Hons) in Computing, in collaboration with Coventry University, UK**

# Coursework cover sheet

**Section A - To be completed by the student.**

|  |  |
| --- | --- |
| Full Name: TAN KHOON KHYE | |
| CU Student ID Number: 14008809 | |
| Semester: 2 | |
| Session:  **April 2023** | |
| Lecturer:  **Puteri Nursyawati Azzuri (puteri.azzuri@newinti.edu.my)** | |
| Module Code and Title:  **4067CEM Software Design** | |
| Assignment No. / Title:  **Continuous Assessment** | % of Module Mark:  **50** |
| Hand out Date:  **12 May 2023** | Due Date:  **Task 1: 02 June 2023, by 11.59pm.**  **Task 2: 07 July 2023, by 11.59pm**  **Task 3: 23 June 2023, by 11.59pm.**  **Task 4: 23 June 2023, by 11.59pm.**  **Task 5: 23 June 2023, by 11.59pm.** |
| Penalties: No late work will be accepted. If you are unable to submit coursework on time due  to extenuating circumstances, you may be eligible for an extension. Please consult the lecturer. | |
| Declaration: I/we the undersigned confirm that I/we have read and agree to abide by the University regulations on plagiarism and cheating and Faculty coursework policies and procedures. I/we confirm that this piece of work is my/our own. I/we consent to the appropriate storage of our work for plagiarism checking.  A picture containing black, darkness  Description automatically generated  Signature(s): | |

# Section B - To be completed by the module leader

|  |  |  |
| --- | --- | --- |
| Intended learning outcomes assessed by this work:  Understand and apply appropriate concepts, tools, and techniques to each stage of the software development.  Understand and apply design patterns to software components in developing new software.  Demonstrate an understanding of project planning and working to agreed deadlines, along with professional, interpersonal skills and effective communication required for software production.  5. Demonstrate an awareness of, and ability to apply, social, professional, legal, and ethical standards as documented in relevant laws and professional codes of conduct such as that of  the Malaysian National Computer Confederation. | | |
| Marking scheme | Max | Mark |
| 1. User Story Mapping | 20 |  |
| 2. Setting up a GitHub |  |
| Repository | 10 |
| 3. Creating a Class diagram and |  |
| design pattern selection | 30 |
| 4. Creating a Prototype User |  |
| Interface and Usability Testing | 20 |
| 5. Discuss the ethical issue |  |
| related to the software | 20 |
| Total | 100 |  |

**Title: College Student Business System**

**Task 3: Class Diagram and Design Pattern**

**3.1 Introduction**

As per what was done in Task 1, the purpose of this task is to record the methods to create a class diagram and selecting a valid design pattern for a business system intended for college students. This means that the system allows students to be sellers and customers on the platform to sell or purchase items. This task highlights the classes and their relationships between each other in the diagram while also rationalizing the option to go with the Strategy Pattern for different mode of payment.

**3.2 Class Diagram**

A class diagram is the representation of an application from a static perspective. It is a type of Unified Model Language (UML). Despite having its own objects, a class can also be derived from other classes. Various kind of components and codes in the application can be visualized, described, and established. In a sense a class diagram can also give a brief overview of the system.

The following are components found in a typical class diagram:

1. Class: Represents an object that breaks down into three parts which are name, attributes, and methods.
2. Artefacts: Represents entities
3. Relationships: Represents the ties between classes that has five types which are inheritance, simple association, aggregation, composition and lastly dependency.

There are four classes in the following figure, each representing their own entity.

A picture containing text, diagram, line, number

Description automatically generated

Figure 3.2.1 Class Diagram of Student Business System

Due to the nature of the question, the researcher does not have to declare any attributes or operations for this task therefore only the classes and their relationships are established. The following is the explanation for the class diagram figure shown above.

Student\_Customer is a role of a student as a customer in the system. They can leave or view product reviews, choose payment method, buy items from seller, and ask sellers about items sold.

Student\_Seller is another role of a student as a seller in the system. They can sell many items on the system and receive transaction details from system.

Itemsis another class that represents the items sold on the system by seller.

Transaction is the final class that represents the payment method chosen by student customer when buying an item from a seller.

**3.3 Design Pattern**

A design pattern is a broad solution to a common issue as it is not yet a finalized to be ready shipped out product that can be written as a code. It is just a simple problem solving model that may be applied in different circumstances. There are a few design patterns to choose from like:

**3.3.1 Factory Method Pattern**

This pattern that is part of the Creational Patterns provides the flexibility to student sellers to add new product types to their store page on the system whilst allowing students to specify the type of item that they want to buy. For example, a textbook or an electronic device.

**3.3.2 Façade Pattern**

This pattern that is part of Structural Patterns limits the dependencies between the client code and the system’s subcomponents. In turn, this makes the user interface of the system simple making it easier to use for both customers and sellers alike. For example, for client code like buyItem(), that code will automatically check whether or not the product in question is still available for purchase.

**3.3.3 Strategy Pattern**

This pattern that is part of Behavioral Patterns is able to handle different payment methods provided by the system. This pattern allows for new payment strategies to be added into the system without having to know the ins and outs of said payment method. For example, debitstrategy should be able to process the logic for all debit cards payment.

Thus, the Strategy Pattern was chosen as the design pattern for this task. This is because for a business system it is crucial for users and sellers to have multiple payment methods. The Strategy Pattern does exactly that as it can handle different mode of payment without having the need to change the main logic for processing payment.

The following is the UML diagram for Strategy Pattern:

A picture containing text, diagram, line, number

Description automatically generated

The context represents the student business system itself and it encapsulates the payment function. This allows the object to interact with the selected payment method without having to understand the specifics of each strategy.

The strategy interface represents the payment\_process. It delegates the payment logic to different strategies down below via the operation processPayment().

The concrete strategies are the different payment schemes that is employed by the system. Each strategies are of their own class which are debit\_credit, google\_pay, cashondelivery and apple\_pay.

The above UML diagram was drawn via Lucidchart.

**3.4 Conclusion**

The Strategy Pattern was selected as the suitable design pattern to meet the system’s requirements. The class diagram on the other hand, offers a graphical illustration of how each classes are associated in the system. The option to go with said design pattern and class diagram allows the developers to start with a stable foundation to develop the student business system.