**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:  **Nawa Silumelume Mubukwanu** | |
| --- | --- |
| CU Student ID Number:  **14008784** | |
| 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.    Signature(s): | |

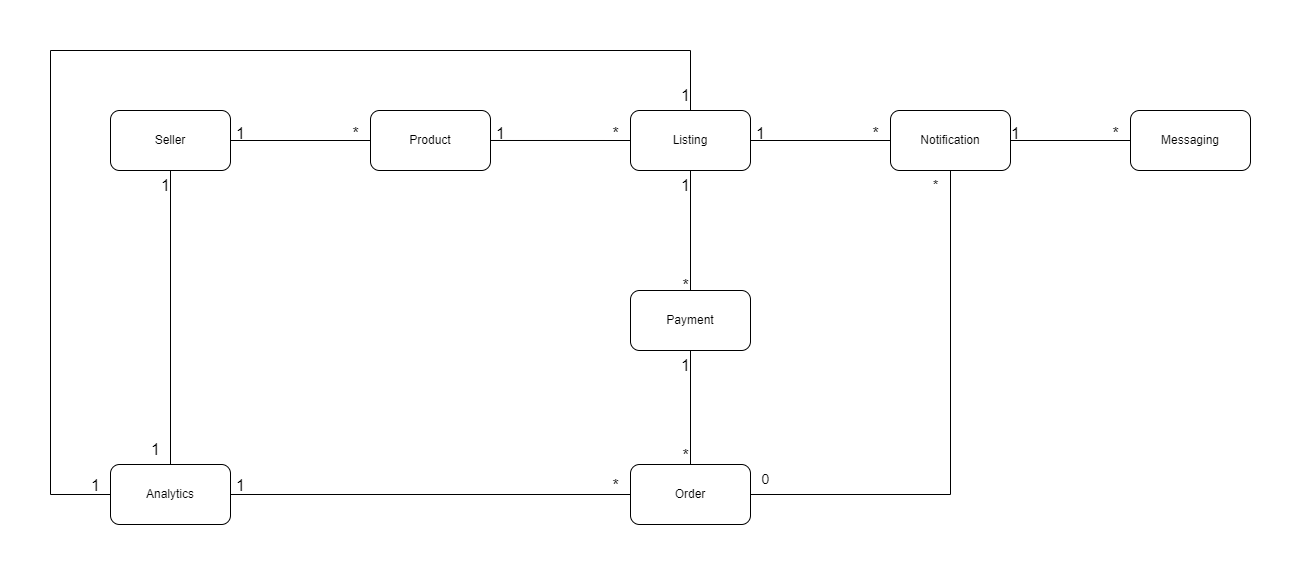
# Section B - To be completed by the module leader

| Intended learning outcomes assessed by this work:   1. Understand and apply appropriate concepts, tools, and techniques to each stage of the software development. 2. Understand and apply design patterns to software components in developing new software. 3. 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 |  |

Task 3: Class Diagram and Design Pattern

# 3.1 Class Diagram

A class diagram is a representation of the relationships and dependencies between classes in the Unified Modeling Language (UML). It illustrates the structure of classes in an object-oriented program and how they interact with each other. Class diagrams are commonly used in object-oriented programming to visualize the relationships between classes. In a class diagram, classes are grouped based on their shared characteristics. Each class is depicted as a box with three sections: the top section contains the class name, the middle section lists the attributes, and the bottom section includes the methods or operations of the class. The connections between the boxes, represented by lines with optional arrows, depict the relationships or associations between the classes. Over the years, class diagrams have evolved alongside object-oriented modeling paradigms, becoming a valuable tool in software development(TechTarget, 2019).



### Figure 3.1 Shows the class diagram for the student business application

1. Seller: Manages profiles, products, listings, pricing, negotiations, notifications, analytics, orders, and communication with buyers.

2. Product: Stores information about products, including descriptions, images, and details.

3. Listing: Connects sellers, products, and buyers, managing availability, prices, and status of individual listings.

4. Notification: Sends alerts to sellers regarding buyer interest and offers.

5. Messaging: Facilitates communication between sellers and buyers, allowing questions, negotiations, and information exchange.

6. Payment: Handles secure and reliable financial transactions between buyers and sellers.

7. Order: Manages order details, status, payment coordination, and fulfillment.

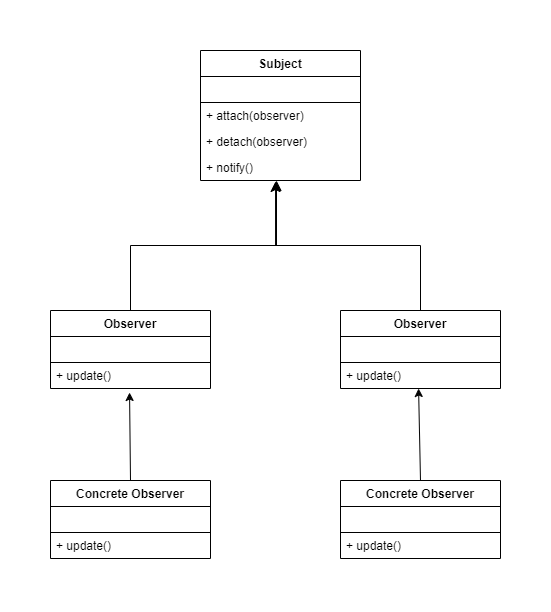
8. Analytics: Gathers and analyzes data to provide insights on listing performance, helping sellers improve their strategies.

# 3.2 Design Pattern

Gulati (2022) defines a design pattern as a documented solution that has been extensively tested and can be applied to address common and recurring challenges in object-oriented software design. It serves as a reliable guide for solving frequently encountered problems in software development.

## 3.2.1 Observer Pattern

The Observer pattern allows for a loosely coupled relationship between objects, where one object (the subject) maintains a list of dependents (observers) and notifies them of any state changes. This pattern is suitable for implementing the notification system, where sellers receive notifications about buyer interest and offers.



### Figure 3.2.1 shows the UML diagram representing the design pattern

Justification for selecting the Observer design pattern:

The Observer pattern offers flexibility, scalability, and loose coupling for the notification system. It allows dynamic addition/removal of observers, promotes separation of concerns, and aligns with an event-driven model, meeting the requirements effectively.

# 3.4 Conclusion

This task covered the use of class diagrams to represent class relationships and dependencies. It introduced the Observer design pattern, justifying its suitability for the notification system. The chapter emphasized the benefits of the Observer pattern, including flexibility, scalability, and loose coupling in software development.

# REFERENCES

*What is a class diagram? definition from whatis.com.* (2019) *TechTarget*. Available at: https://www.techtarget.com/searchapparchitecture/definition/class-diagram (Accessed: 07 July 2023).

Gulati, V. (2022) *What are design patterns?*, *Scaler Topics*. Available at: https://www.scaler.com/topics/design-patterns/what-are-design-patterns/ (Accessed: 07 July 2023).