Assignment 1: System Design

Using the scenario provided, you should design and develop:

- A class diagram.
- An activity diagram for the process of a customer completing an order.
- A state diagram highlighting the states of an order (and their transitions) for your proposed system.
- A sequence diagram.

Submission Date: 27 September 2021

Introduction

Modelling is an important aspect in software engineering as it gives an idea of the functioning of a system before it is developed, coded or implemented. It gives a certain level of precision and detail (Gomaa, 2011: 3), as different models are designed and analysed, thus helping to see the system "from different perspectives". This is "also referred to as multiple views" (Gomaa, 2011: 3). In fact, modelling is used in most phases of a software life cycle (Seidl et al, 2012).

Unified Modeling Language (UML), as a consolidation of the best unified modelling practices that have been established over the years (Seidl et al, 2012) and a "graphical modeling language", is therefore used "in developing, understanding, and communicating the different views" (Gomaa, 2011: 3). UML allows the software engineer to illustrate the diverse aspects of a software system in "a single framework using object-oriented concepts" (Seidl, 2012: 1). As a standardized modelling language, it enables software developers to work together and thus "promote communication and productivity”, by using “an integrated set of diagrams” (Chonoles & Schardt, 2003: 2).

For this assignment, four UML diagrams (class, activity, sequence and data diagrams) will be designed and developed to visualise the different aspects (classes, activities, relationships...) of an online shopping centre.

1. Class Diagram

Class diagrams, a main part of Structure Diagrams, “originate from conceptual data modeling and object-oriented software development” and “are used to specify the data structures and object structures of a system” (Seidl, 2012: 17). The class diagram is based primarily on the concepts of class, generalisation, and association.

Classes are like blueprints for creating an object. For instance, each customer is a distinct object, but all customers have the attributes and behaviours associated with one class, the general class of Customer. A class can define specific sets of characteristics that are shared by all objects from that class.

In our online shopping system, we have thought of having the following classes, as shown in the simplified diagram:

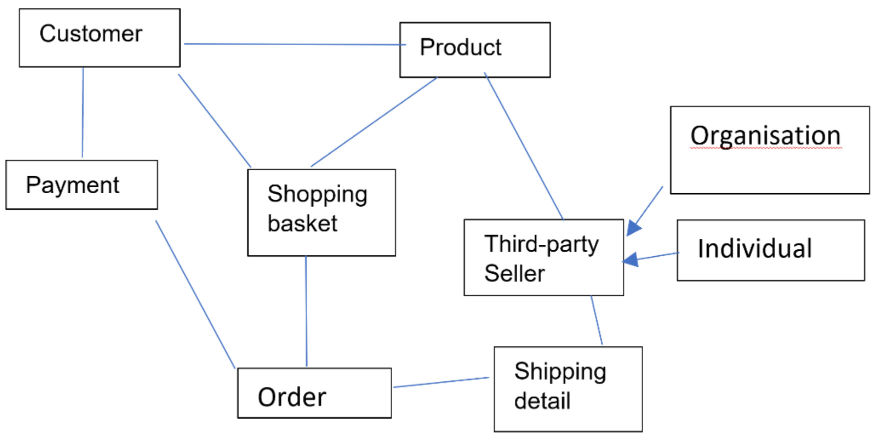

Figure 1: Classes

Classes were identified by looking at the keywords or nouns given in the scenario and by analysing the relationship between them. For an online shopping application, the important classes are: **customers**, who will search and select **products** (which can be sold by different **sellers**) and put them in a **shopping basket**. The **order** will then be awaiting **shipping** and **payment** details before being delivered. From there, we have created a more detailed class diagram on the online application, Creately, with the class name, attributes, functions and the relationships.

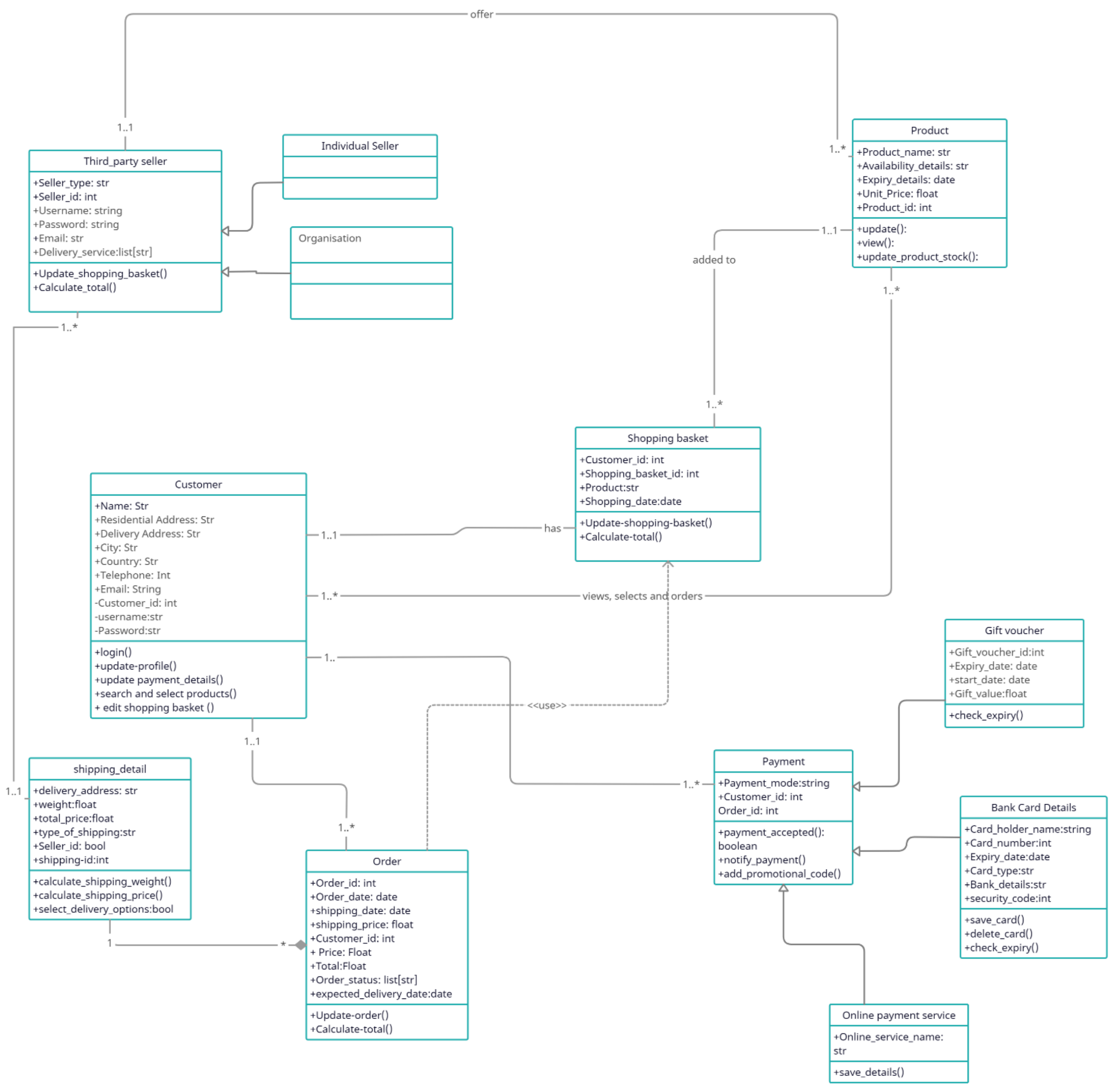

Figure 2: Class diagram

Most design patterns rely on two basic object-oriented principles known as composition and inheritance.

- ◈ Composition is the act of collecting several objects together to create a new one. Composition is usually a good choice when one object is part of another object (Seidl et al, 2012). In the above diagram, a composition line is drawn from the class "shipping details" to the class "order". If the "order" class is cancelled, then, the shipping details class is automatically cancelled.
- ◈ Inheritance is the principle that allows classes to derive from other classes. In the above example, sellers are the superclass and website sellers and third-party sellers are the subclasses. Inheritance is also applied to the Payment class, which is the superclass here. "Gift voucher", "Bank card details" and "Online Payment service" are the subclasses.
- ◈ A dependency line (dotted line) is drawn from Order class to shopping basket, as the order depends heavily on the shopping basket to exist.
- ◈ All the other relationships are association. This enables communication between classes. And multiplicity (1..* etc) indicates the number of instances of the element.

The remaining diagrams which will be developed are mainly Behaviour Diagrams.

2. An activity diagram for the process of a customer completing an order

Activities or processes can be modelled by using an activity diagram (Seidl et al., 2012). The figure below illustrates the actions which are necessary for the customer to order a product and for the product to be shipped.

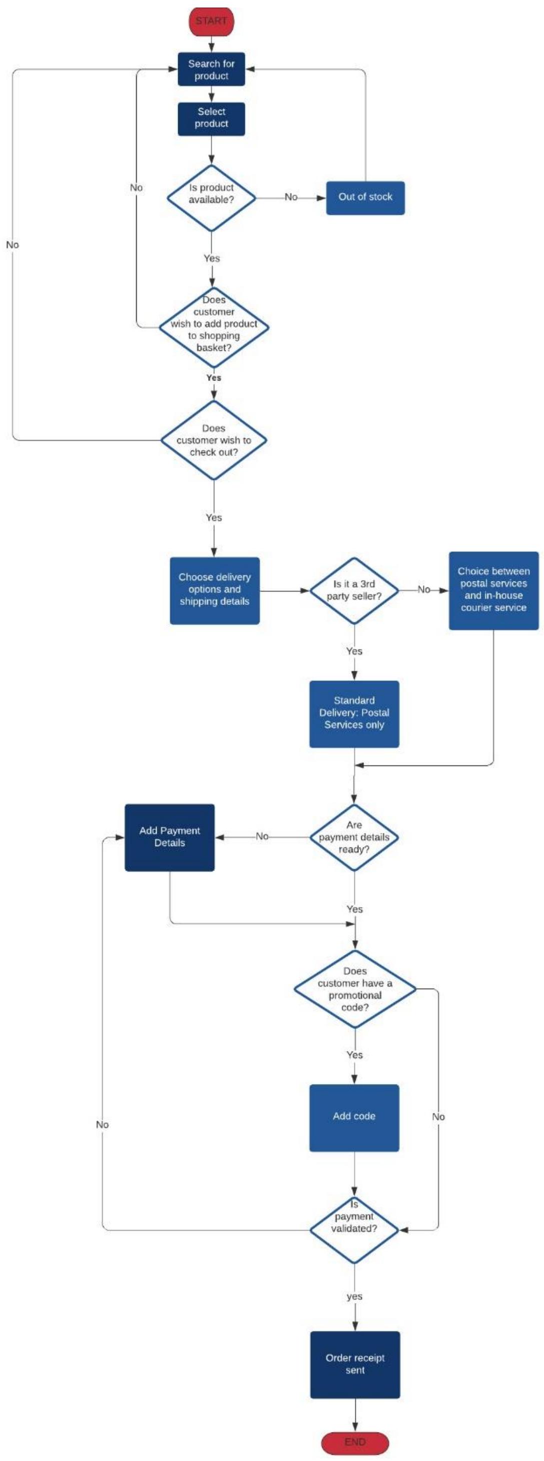
START
Search for product
Select product
Is product available?
No
Out of stock
Yes
Does customer wish to add product to shopping basket?
No
Yes
Does customer wish to check out?
No
Yes
Choose delivery options and shipping details
Is it a 3rd party seller?
No
Choice between postal services and in-house courier service
Yes
Standard Delivery: Postal Services only
Are payment details ready?
No
Add Payment Details
Yes
Does customer have a promotional code?
Yes
Add code
No
Is payment validated?
No
yes
Order receipt sent
END

Figure 3: Activity Diagram

3. A state diagram highlighting the states of an order (and their transitions) for your proposed system

Another behaviour diagram is the State Machine Diagram. As stated by Seidl et al (2012: 19), “within their life cycle, objects go through different states”. In the figure below, when a customer initiates an order, the order is said to be in progress. Once the shopping is completed, the state changes to Processing Order. And the order then moves to another state, “Order pending”, when the check-out process (such as the payment details and delivery details are given) is initiated. Once the payment is validated, the warehouse is informed of the order and the order is in the ‘picking’ state. The order is collected and packaged and joins another state – that of being ready for delivery. Finally, the order is picked up by the delivery service and the order ends at the shipped/delivered state.

This behaviour can be represented in UML using the state machine diagram. The diagram below describes the different possible states and state transitions triggered by various events (Seidl et al 2012), during the online shopping.

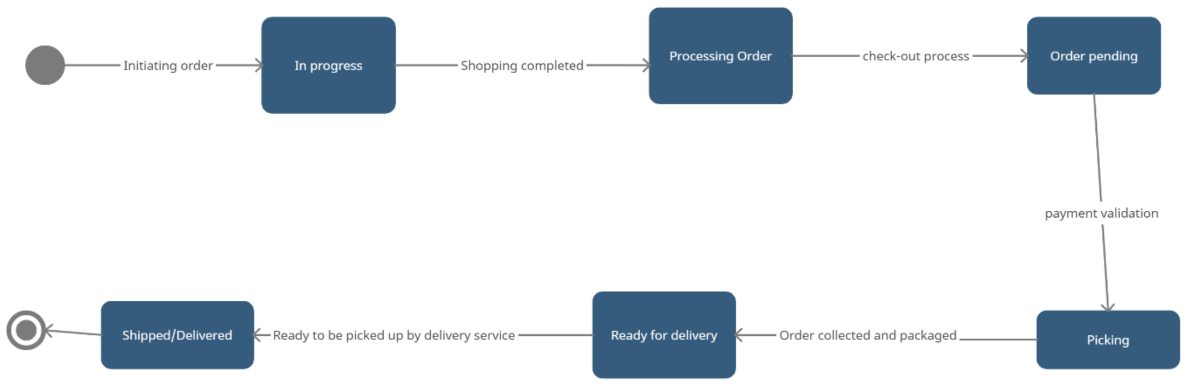

Figure 4: State Diagram

4. Sequence diagram

The sequence diagram, a two-dimensional diagram, models objects and describes the interactions in detail, especially how elements in the system or infrastructure communicate and work together. The objects participating in the interaction are depicted horizontally and the vertical dimension represents time (Gomaa, 2011; Seidl et al, 2012). The double solid line shows when the object is executing. The focus is on the chronological order of the messages (sent, received and self-messages) exchanged between the interaction partners.

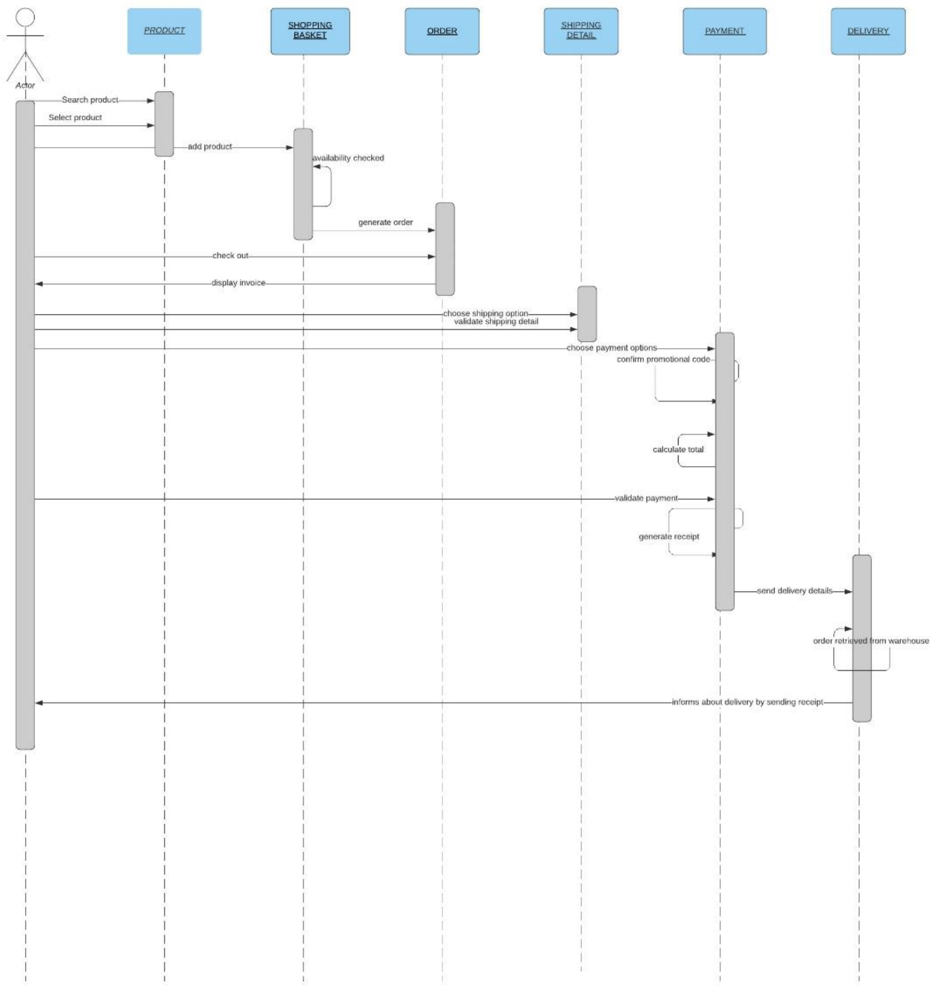

Figure 5: Sequence Diagram

No of words: 947 (without question and references)

References:

Bourgeois, D. (2014) *Information Systems for Business and Beyond*. Saylor Academy.

Buckley, O. (2021) Units 1-6, Lecture Notes, Object-oriented Information Systems OOIS_PCOM7E, University of Essex Online, delivered August and September 2021.

Chonoles, M.J. & Schardt, J.A. (2003) *UML 2 for Dummies*. New York: Wiley Publishing.

Giden, T. (November 15, 2015) Introduction to UML. Available from: https://www.youtube.com/watch?v=vgYKW9O6fFE [Accessed 15 September 2021]

Gomaa, H. (2011) *Software Modeling and Design Uml, Use Cases, Patterns, and Software Architectures*. Cambridge University Press.

Lucidchart (July 21, 2017) UML Class Diagram Tutorial. Available from: https://www.youtube.com/watch?v=UI6lqHOVHic [Accessed 15 September 2021]

Seidl M. et al (2012) *UML @ Classroom – An Introduction to Object-Oriented Modeling*. UK: Springer.