SOEN 343 Project (Hexagon - Sprint 1)

Guillaume Lachapelle
ID: 40203325 Role: Team leader / Developer

Ann-Marie Czuboka
ID: 40209452 Role: Developer

Isabelle Czuboka
ID: 40209525 Role: Developer

Nicholas Piperni
ID: 40207764 Role: Developer

Karina Sanchez-Duran
ID: 40189860 Role: Developer

Oliver Vilney
ID: 40214948 Role: Developer

I- Project Definition

I.1. Objectives:

The rise of e-commerce business is an ongoing phenomenon in today's economy. During the pandemic, retail e-commerce sales rose from 3.9 to 6.2 percent of total retail sales (Zanzana, S., & Martin, J., 2023). Consequently, more online shopping correlates directly with more delivery. Delivery is an important part of any online business as it impacts the customers outlook on said business. As such, the need for better delivery systems is required more and more. The objectives of our delivery application are to have an easy-to use interface, provide services fitted to your delivery needs and provide excellent timely feedback throughout any given service.

I.2. Defined method of approach:

The way this project will be carried out is through many team meetings. We will have 2-3 meetings a week, as a team of 6 people, to assess how the software will be built. Guillaume Lachapelle will be our team leader and will assume the responsibility of delivering the project in increments. All members of the team will be responsible for programming the software needed. Moreover, this project will consist of several diagrams like domain model, context diagram, use cases, etc. Therefore, during the weekly meetings, we will all participate in the creation of these diagrams and tables, since all software will be based fundamentally on these drawings. It will be important to go back to these drawings throughout the software development to assure the final product remains as intended. In many cases, we will also do pair-programming, as in we will work on bigger tasks in teams. In doing so, we will alleviate the burden of a big task on a single person as well as spreading insight on the functionality of the software to more members of the team. This will avoid having only one person know how to solve a problem in a particular area. In all, the deliverables will consist of these drawings as well as implementation of code.

I.3. Project scope:

The project scope consists of a myriad of functionalities, in which many interlace. The project will be a website that will deliver many features. The website must allow someone to login as a particular (individual account) or as a corporate user (company account). It will be capable of inputting a delivery order with the following specifications: items, source and destination, volume and weight of items. In response to this input, the product should be able to make a unique quotation (price). Moreover, an email service should be available for confirmation of order, possible status updates, payment confirmation, etc. The emails should be capable of including images for a clearer message. Customers must be able to pay via credit. The website should also allow a review section for each completed delivery. This will allow our product to evolve more towards customer needs. In addition, when a delivery order is active, there must be a tracking system in place, so that a user can see the progress of their orders. The website must support staff. More specifically, there must be accounts for drivers: the workers who will be actually performing the service. They will have access to orders they are in charge of. In addition, they must be able to

change the status of their deliveries (received, en route, delivered, …). The product must have admin capabilities. Therefore, an admin must be able to modify, create and delete users like staff and customers or delivery orders.

II- Problem Definition
II.1 What is the problem?

Whilst delivery is the last step in any online purchase, it leaves a big mark on the customer's satisfaction. Delivery becomes an important step in keeping loyal customers and attracting new business. However, it can be expensive for companies to deliver their products on their own. It also involves added logistics and complications. This can create dissatisfaction for both the customers as well as the companies or particulars wanting to deliver their items. Therefore, the problem is how companies keep up with fast and efficient delivery times. This problem grows if the company isn't big enough or settled enough to perform large amounts of deliveries.

II.2 How the problem emerged?

The problem comes mostly from the concept of last-mile delivery. The last part of delivery, referred to as the last-mile, is the last step for companies to take their products from their warehouse to the customers' location. This is usually the most costly and inefficient part of the delivery, as it involves many obstacles (Dolan, 2023). There is also another problem, which is that companies cannot always keep up with the demand, especially during some periods like the holiday season (Sayar, 2022). During festive seasons or high traffic times, there are many more parcels to deliver at once which increases the Last-mile problem, since there are more packages to drop-off in one location. Companies also have labor shortages or have to cut some employees. This shortage in employees makes it harder for companies to deliver packages rapidly and efficiently.

II.3 What is your solution?

To solve the issue of labor shortages, companies can use our services to take over (subcontract) their necessary deliveries. On top of regular delivery services, we will provide a service to deliver orders to the final destinations. To expand on that, in many cases, large companies like Amazon will have large amounts of delivery orders that have to be shipped to Canada from international waters. It is the local delivery that becomes time consuming. Therefore, our company will help the process by delivering the parcels locally for the "Last-mile" once those orders have landed.


II.4 What are the advantages of your solution in comparison with the existing solutions?

One of the biggest advantages of our solution in comparison with the existing solutions is that we will provide delivery for both companies and particulars, as well as adding an option for companies to have recurrent deliveries. For example, a company will be able to deliver items every week to one of their retail locations by simply selecting "recurring delivery" during the checkout process. This will make it easier for companies to get their products delivered, while saving time, money, and resources on the last-mile period of delivery. Since our service is also offered to individuals, we also provide a wider range of service than existing delivery services offer.

III- Technology used: technology used for the following activities:

During the creation of this product, the team will communicate via a Discord group. This will help us brainstorm, plan meetings and share insight or concerns regarding anything on the project. In order to monitor and verify our work, we will use GitHub. Here, we will most likely post our team meeting minutes and document basic progress. As well, GitHub comes equipped with different functionalities that will verify source code like the use of pipelines. The design of the project is done using draw.io. This tool was used to create the context diagram, as well as the domain model, and will be used for any other future diagrams to be built. The main interface will be built as a web application using the Angular Framework, which uses TypeScript, HTML, and SCSS. The database will be created using Firebase, as well as the authentication and storage of other contents such as files and images.

IV- Context diagram
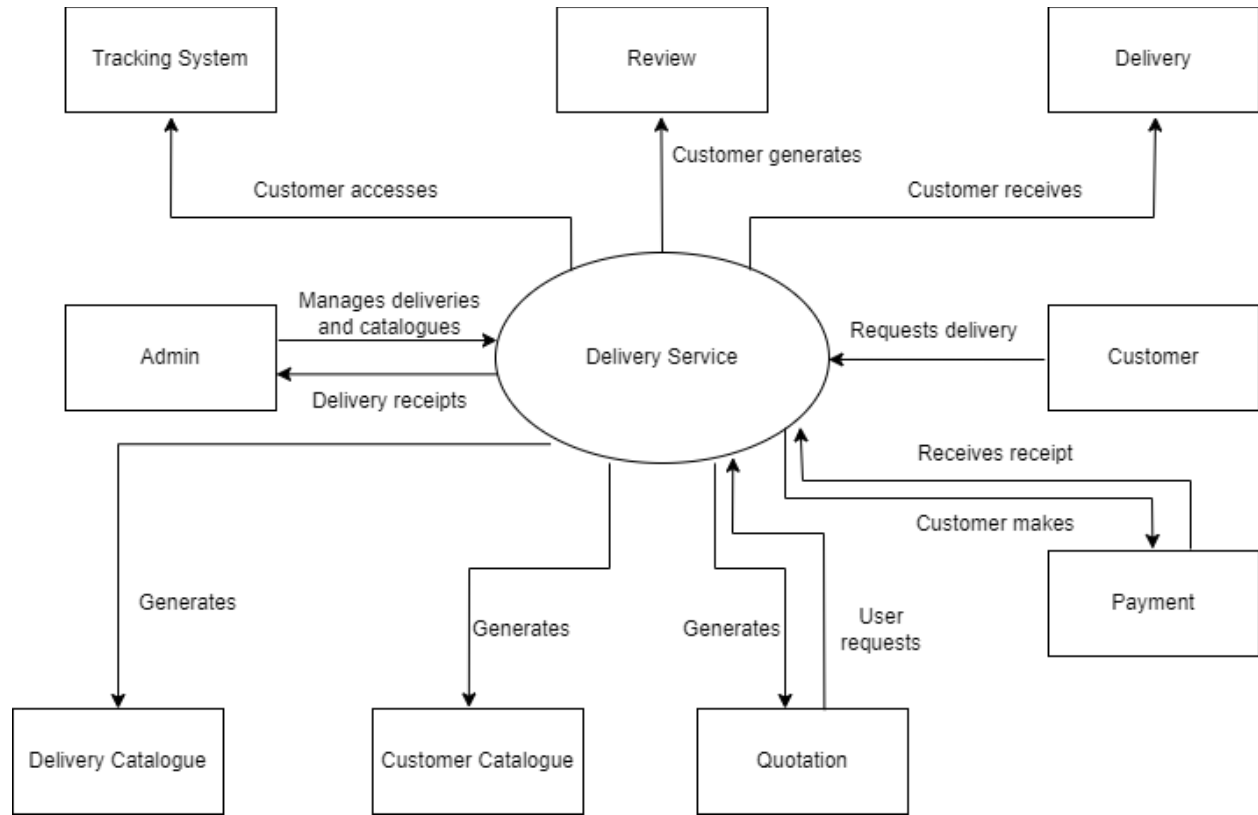
The context diagram shows the interactions between the delivery system and other actors which will be interacting with the service. It shows every aspect of the service, and every functionality which will be implemented, from the interactions of the customers with the service to the different options the service provides.

V- Domain model
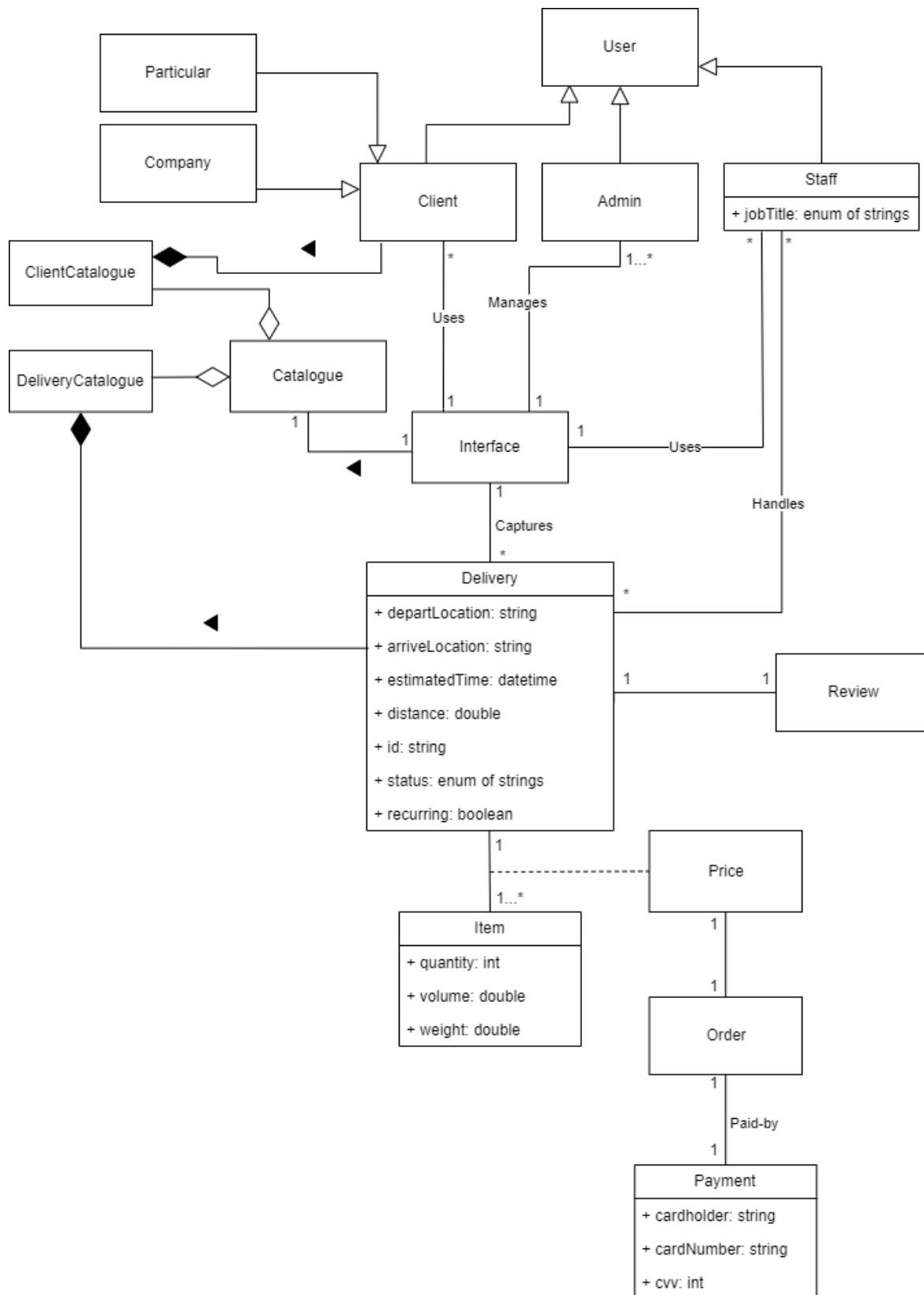
All the types of user classes such as *Client*, *Admin* and *Staff* are associated with the *Interface* class. The *Client* and *Staff* physically use the interface to either request or update a given delivery while the *Admin* manages the interface through the firebase account. Members of *Staff* are in charge of handling deliveries.

The *Delivery* class has a status attribute to show the tracking status.

Our domain model contains two client classes, *Particular* client and *Company* client. The reason there are two types of clients is because the *Company* client has access to recurring deliveries to meet their business demands and needs.

There are two catalogue classes, *ClientCatalogue* and *DeliveryCatalogue*. Both catalogues essentially represent the client table and delivery table in our firebase database which are both populated from the interactions and events done in the interface such as creating a client account and creating a delivery. These catologues are represented as compositions, since if the client catalogue is deleted, then the clients would subsequently delete. Same goes for the delivery catalogue.

The *Price* class is an association class between *Delivery* and *Item* and is responsible for calculating the price of a specific delivery. The *Price* class can be used for generating a quotation for the clients, as well as calculating the total price during checkout. Upon request of the client, one then proceeds to *Order* and, subsequently, the *Payment* of the delivery. Every delivery is associated with one review.

References

Dolan, S. (2023, January 10). *Last Mile Delivery Logistics explained: Problems & Solutions*. Insider Intelligence. https://www.insiderintelligence.com/insights/last-mile-delivery-shipping-explained/

Sayar, J. (2022, November 29). *How to avoid ecommerce shipping delays during holiday season?*. Shopney. https://shopney.co/blog/how-to-avoid-e-commerce-shipping-delays-during-holidays/

Zanzana, S., & Martin, J. (2023, February 21). Retail e-commerce and COVID-19: How online sales evolved as in-person shopping resumed. https://www150.statcan.gc.ca/n1/pub/11-621-m/11-621-m2023002-eng.htm