# System design

University of Essex

Following the requirements of the online store system, I present an architecture to deliver the desired system to a success. There may be some objects that can be expanded or are not present in my design, nevertheless my main goal was to have a simple structure that can be escalated if needed for a more complex system.

For the registration/login system is to not allow guest checkout. Some online stores let users buy products without registration, nonetheless It can have some drawbacks for tracking orders (Pape, n.d.) and it’s incompatible with the requirement of saving payment details for future purchase, then the choice was to force registration before payment and after that, let the user proceed to checkout and payment.

## Classes, objects and attributes

### Customer

This class has the possibility to create orders vía the Shopping Cart, also can directly cancel the orders he issued; however, this does not erase orders from the database, they can be accessed or viewed by staff and sellers for statistical purposes.

#### CustomerID

The customerId attribute is the main attribute that relates to Shopping Cart, Order and to all the purchase history from all users, as a primary key, this ID is unique.

#### Personal details

These fields, firstName, lastName, email, shippingAddress and billingAddress are needed for security reasons, not only for shipping reasons, also it is important to have data that could be used to validate the user identity in disputes or refund cases.

#### Functions

Common functions like login and register are thought in this model, also those for cancel order, update order and to save further details to its profile. Also added the pay() function to send card details to the payment gateway. I did not include the payment gateway process as I prefer this to be implemented with some external provider, even when the Debit/Credit card option is selected.

### Cart, Order, Order Item and Product

When Customer adds products to the Shopping cart, this creates a Cart object.The Cart class exists by itself, separated from order, to follow abandoned carts relating them to users.

Most objects in the model are related to the Order class. It is created when a user adds items to the Cart, the Order Item object relates products to the order and sets the quantity of the added products. When the Customer is ready to pay, payment details are sent with the GatewayBridge class, and when payment is accepted, order is updated. Order has functions to update status, get the total amount of that order and get order details.

### Seller, Product, Catalogue

The seller class is the one that manages catalogues and products related to them, this is the base for the storefront system. Products related to these catalogues can be similar, but they have their own id, prices and also taxes. Seller contains the details of the admin of a given Store or individual seller. The main store could be a seller class as well. This class also has functions to delete, create and change Product objects, and has the possibility to update an order status. Similar class is the Staff class, however it cannot create or delete any product, only update stock and order status.

References:

Pape, Marina (n.d) Determining if Guest Checkout is Right For Your Store. Available from: <https://woocommerce.com/es-es/posts/guest-checkout-right-for-store> [Accessed 19 June 2021].