N10644024

Ng Zhang Lin (Simon)

IFN557 Assignment 1

Part A

# Description

Page Type : E-commerce

The website ultimately should be a functional ecommerce website where users can search for products/services available in the store, add the product/services into a “basket”, and ultimately complete the ordering process by providing payment details, shipping method, and shipping address.

On the main page, there should be a “category” system where items are grouped into various “categories” to allow the user to find the appropriate product/services in a timely manner. Furthermore, there should also be a “list” of recommended products/services based on the popularity (in a real world there should be existing data for this selection but for this assignment, I would list some random products) so that users can select them without needing to search through the catalogue and add it into their “basket”. Finally, the user should also be able to view their “basket” that shows all the added products alongside details such as the “quantity” and “price” of each individual item and a “total price” of every item inside the basket.

On the product details page, an example product/service’s details (name, image, description, price, and any item specific information) is listed down for the user so that they can understand the product fully before deciding on adding it to their “basket”.

In terms of the design of the page, it should be intuitively designed in a way that any new users would be able to utilize the page to search for any items available in the store without too many external help.

# User Stories

**Number 1**

Tittle : Ordering an item

Story :

As a <customer> I want to <add item to cart and go through the payment process> because <I wish to purchase it and have it delivered to my address>

Acceptance Criteria :

1. A function for adding items into a “cart”
2. A function to “checkout” with cart with multiple items
3. A function to select shipping methods and their price for the specific item
4. A function to process order that prompts users to enter payment details and shipping details

**Number 2**

Tittle : Searching for an item by categories

Story :

As a <customer> I want to <view all items related to stationery> because <I need to order new office supplies>

Acceptance Criteria :

1. Items sorted according to categories
2. Function to browse items based on categories using the navigation bar
3. Function to add items to “cart” on the main page without needing to enter the specific item page

**Number 3**

Tittle : View Item details

Story :

As a <customer> I want to <view more details about an item> to <ensure that the item details are suitable for my needs>

Acceptance Criteria :

1. Each Item having its own unique “item page” where details such as name, image, description, price, and any item specific information is listed down)
   1. For this assignment only 1 item page is necessary\*
2. Function to filter items based on categories

**Number 4**

Tittle : View Basket

Story :

As a <customer> I want to <view my basket> to <ensure that all the items I have added is within the basket>

Acceptance Criteria :

1. Function to store and list out items the user had “added” to the basket throughout the entire time they were on the page
   1. Best if there is a storage function that stores the list of items so that even if the user closes the page, the user can continue using the same basket later just by visiting the website
2. Function to list out the “quantity” and “price” of each individual item inside the basket
3. Function to calculate and show the “total price” of all the items within the list

**Number 5**

Tittle : Searching item

Story :

As a <customer> I want to <search for a specific item by its name> to <purchase it >

Acceptance Criteria :

1. Functional search bar that accepts string input
2. Search bar system that can compare the input with its existing database to output multiple results that contains the input

# Conceptual Model

**Concepts**

1. Customer
2. Items
3. Payment Details
4. Order

**Associations**

Customer views(Simple) Items which are added into (Composition) Order

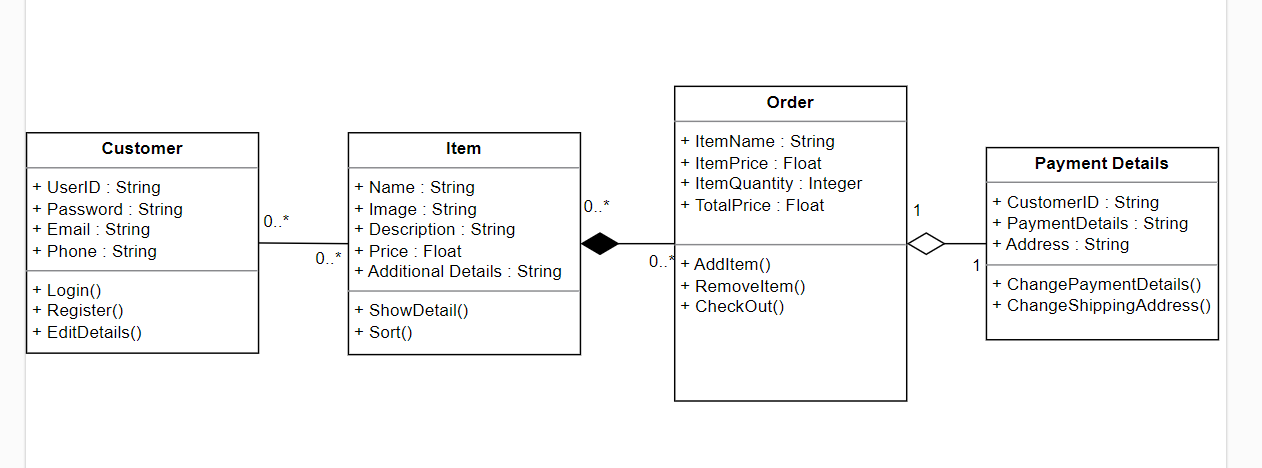
Order has(Aggregation) Payment Details

**Multiplicity**

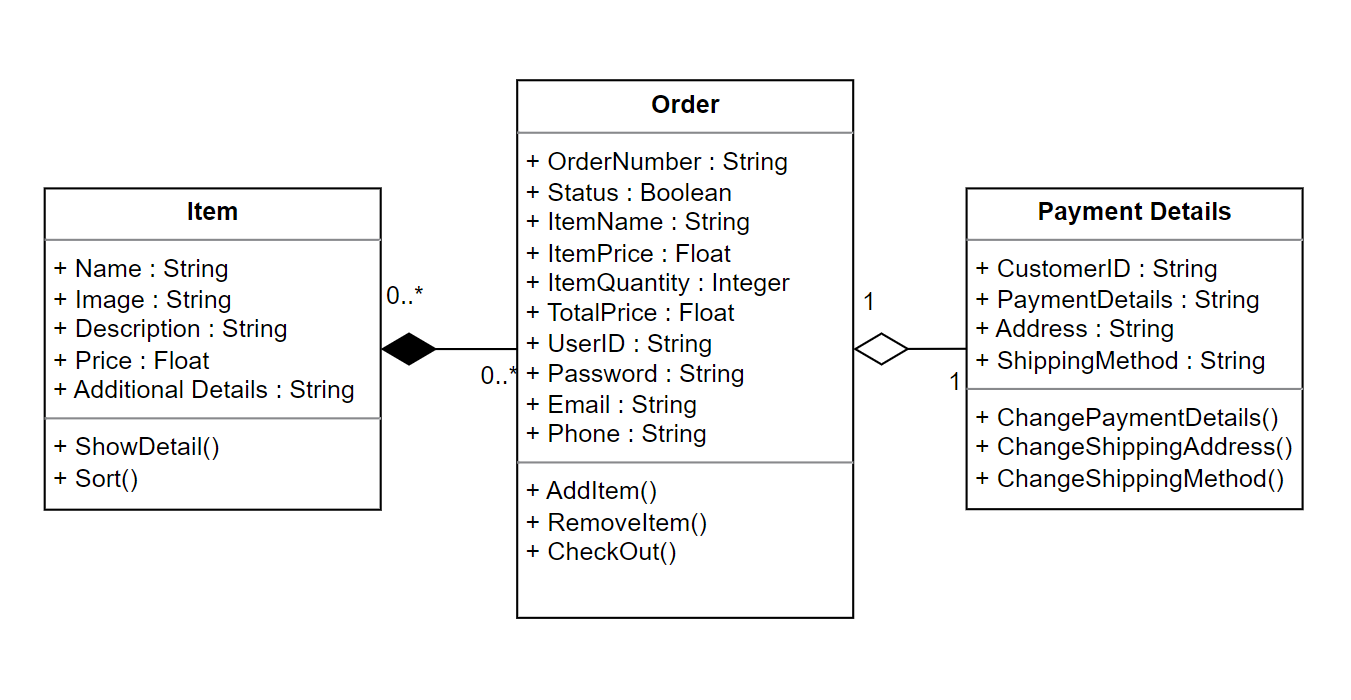
Customer – Item (0..\* to 0..\*)

Item <- Order (0..\* to 0..\*)

Order <- Payment Details (1 to 1)



**Simplified**



# Database Model

