# User story

|  |  |
| --- | --- |
| **Persona** | **User Story** |
| Shop Owner | As an a ShopOwner I would like to be able to manage my items so that I can accommodate changes to my physical inventory. |
| Customer | As a Customer I would like a shopping cart so that I can checkout multiple items at once.  As a Customer, I would like to be able the shopping cart to “remember” my choices so that I can resume my shopping experience even after closing the browser.  As a Customer I would like to have an order history so that I can check what I have ordered. |

# User requirements

1. Shop owner:
   1. CRUD with regards to Items
2. Customer:
   1. Add Items to Shopping Cart
   2. Create Order
   3. View Order History

Future requirements

1. Authentication and authorization
2. Shop owner can set quantity available
3. Shop owner can view pending orders

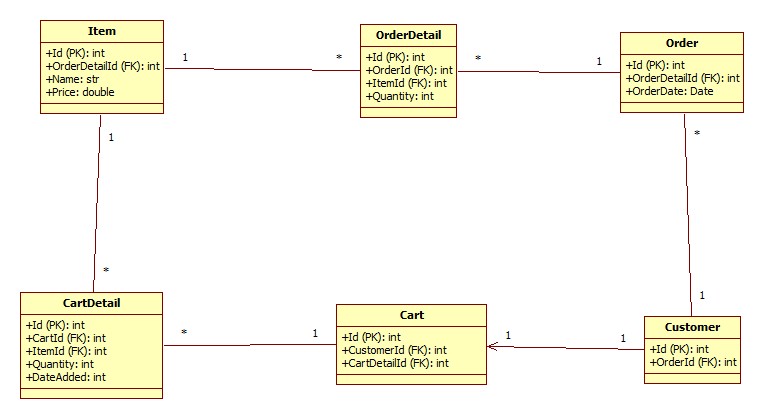
Assumptions

1. There is only 1 Shop owner, i.e. this platform is only for the Shop Owner in question

# Project resources

1. Backend: Java Spring REST
2. Frontend: Angular
3. DataBase: MySQL

# ERD diagram



# API Documentation

The root URL is “<http://localhost:8080/api>”

So, just append the following API URL paths to the root URL, e.g. <http://localhost:8080/api/items>

“items” For Shop Owner to create, read, update, and delete Items

|  |  |  |  |
| --- | --- | --- | --- |
| **url path** | **Action** | **Description** | **Returns** |
| “/items” | GET | Get all items | Array of Items with 200 Ok |
| “/items/{itemId}” | GET | Get item, with item Id specified in the path variable | Single Item with 200 Ok |
| “/items” | POST | Create item, from item in Http Request Body | Created Item with 201 Created |
| “/items” | PUT | Update item, from item in Http Request Body | Updated Item with 200 Ok |
| “/items/{itemId}” | DELETE | Delete item, with item Id specified in the path variable | 200 Ok |

“carts” For Customer to add Items to Cart

|  |  |  |  |
| --- | --- | --- | --- |
| **url path** | **Action** | **Description** | **Returns** |
| “/carts/{customerId}” | GET | Get Customer’s Cart, with Customer Id specified in the path variable | Single Cart with 200 Ok |
| “/carts” | POST | Add Items to the Cart, with item Id, quantity, and customer Id in Http Request Body | Single Cart with 200 Ok |
| “/carts” | PUT | Update the items in the Cart | 204 No Content |
| “/carts/{cartDetailId}” | DELETE | Delete an item from the Cart, with the CartDetail Id specified in the path variable | 200 Ok |

“orders” For Customers to create and read orders

|  |  |  |  |
| --- | --- | --- | --- |
| **url path** | **Action** | **Description** | **Returns** |
| “/orders/{customerId}” | GET | Get all Orders of Customer, with the Customer’s Id specified in the path variable | Array of Orders with 200 Ok |
| “/orders” | POST | Create Order for Customer, with the Customer’s Id specified in the Http Request Body | Created Order with 200 Ok |
| “/orders/latestOrder/{customerId}” | GET | Get the latest order of Customer, with Customer’s Id specified in the path variable | Singe Order with 200 Ok |

## Errors

|  |  |  |
| --- | --- | --- |
| **Error Name** | **Status Code** | **Description** |
| CartDetailNotFoundException | 404 | Cannot find the specified CartDetail with the provided CartDetail Id. |
| CustomerNotFoundException | 404 | Cannot find the specified Customer with the provided Customer Id |
| EmptyCartException | 400 | Occurs when Customer tries to create an Order with no Items in Cart. |
| ItemCascadeDeleteError | 500 | Shop Owner tries to delete an Item that is related to a Cart or an Order. Cascade delete is disabled. |
| ItemNotFoundException | 404 | Cannot find the specified Item with the provided Item Id |
| QuantityLessThanOneException | 400 | Customer tries to update Item quantity in Cart by providing a quantity less than 1 |
| PaymentFailureException | 401 | Customer submitted payment credentials that are unauthorized by the payment service provider |