

**BOOK LIBRARY**

**Software Design Specification**

– Hanoi, January 30th 2023 –

**Record of changeS**

| **Date** | **A\* M, D** | **In charge** | **Change Description** |
| --- | --- | --- | --- |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

\*A - Added M - Modified D - Deleted

**Table of Contents**

[I. Overview 4](#_heading=h.gjdgxs)

[1. Code Packages 4](#_heading=h.30j0zll)

[2. Database Design 4](#_heading=h.1fob9te)

[a. Database Schema 4](#_heading=h.3znysh7)

[b. Table Description 4](#_heading=h.2et92p0)

[II. Code Designs 5](#_heading=h.tyjcwt)

[1. <Feature/Function Name1> 5](#_heading=h.3dy6vkm)

[a. Class Diagram 5](#_heading=h.1t3h5sf)

[b. Class Specifications 5](#_heading=h.4d34og8)

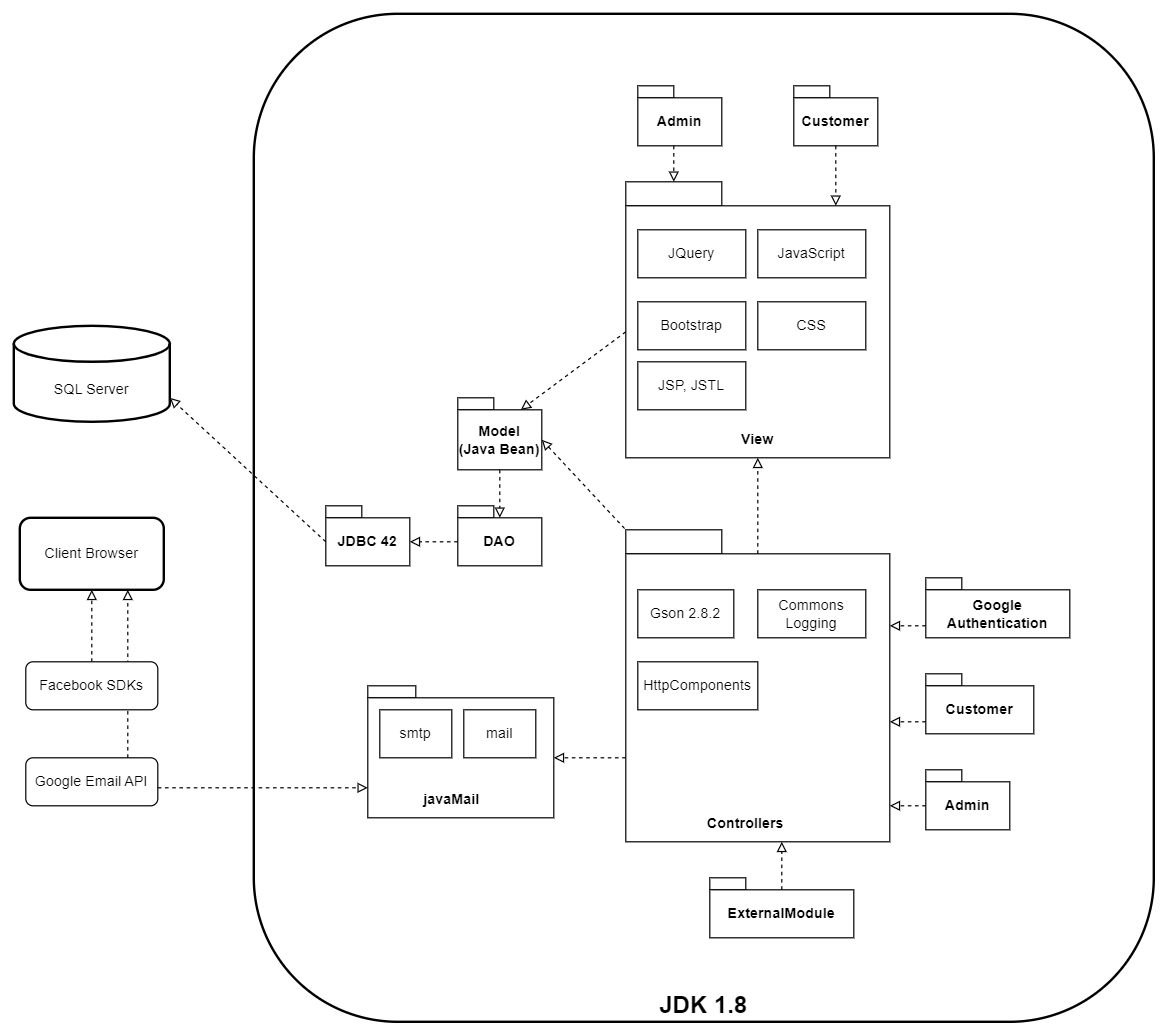
[c. Sequence Diagram(s) 5](#_heading=h.2s8eyo1)

[d. Database queries 6](#_heading=h.17dp8vu)

[2. <Feature/Function Name2> 6](#_heading=h.3rdcrjn)

# I. Overview

## 1. Code Packages

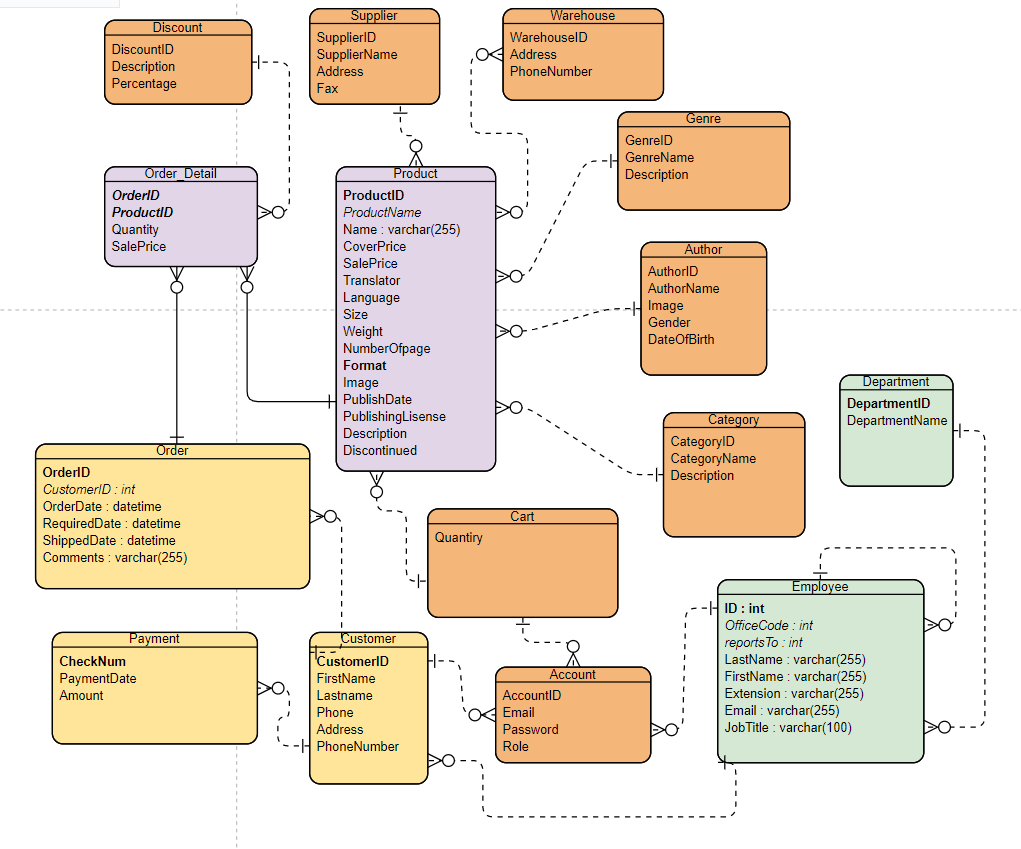


***Package descriptions***

| **No** | **Package** | **Description** |
| --- | --- | --- |
| 01 | model | Package contains all the object classes represent the entities and relationships in Database  The classes all have names corresponding to the entity in the Database |
| 02 | dao | Package contains all the object classes that access directly to Database  The classes contained in this all have the ending suffix [DBContext] |
| 03 | controller | Package contains all the business classes that process the logic tasks |
| 04 | controller.customer | Package contains all the business classes that process the logic tasks for customer role only |
| 05 | controller.admin | Package contains all the business classes that process the logic tasks for admin role only |
| 06 | view | Package contains all the assets, resource to display information on client browser |
| 07 | view.admin | Package contains resource of admin role to display information on client browser |
| 08 | view.customer | Package contains resource of customer role to display information on client browser |
| 09 | controller.GoogleAuthentication | Handle google login when user selects google account login feature |
| 10 | controller.ExternalModule | Contains other processing modules  The classes included in this must contain the ending suffix: [module] |

## 2. Database Design

### a. Database Schema



### b. Table Description

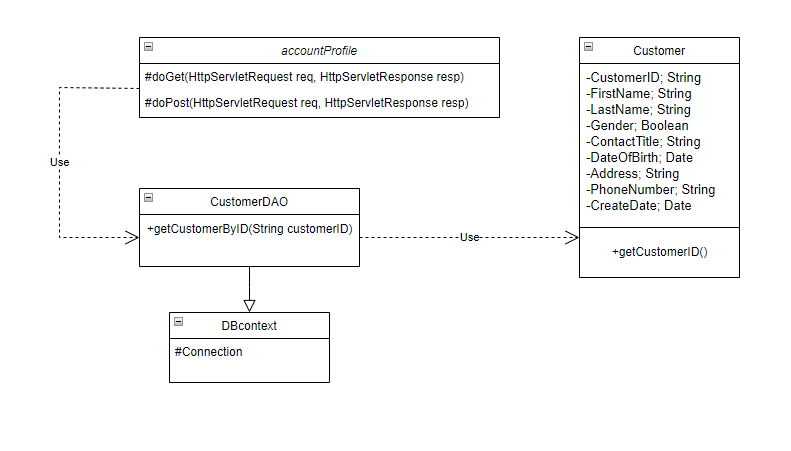
| **No** | **Table** | **Description** |
| --- | --- | --- |
| 1 | Product | Main object is book |
| 2 | Customer | Consumer, who wants to buy book |
| 3 | Employee | Manager, saler performs business operation |
| 4 | Account | The registered user of the system |
| 5 | Order | Record information about an order |
| 6 | Order Detail | Record more detail in an order |
| 7 | Category | Classify book by categories |
| 8 | Genre | Classify book by Genres |
| 9 | Warehouse | Repository to store book, perform send book to customer |
| 10 | Supplier | Publisher will print book and supply them into the market |
| 11 | Discount | Store information about voucher |
| 12 | Cart | Store book which customer want to buy |
| 13 | Payment | payment method of customer |
| 14 | Department | Classify employee into department, admin/saler…etc |

# II. Code Designs

## 1. Customer feature

### 1.1 View Profile

### a. Class Diagram



### b. Class Specifications

#### accountProfile Class

| No | Method | Description |
| --- | --- | --- |
| 01 | doGet | Called by the server (via the service method) to allow a servlet to handle a GET request. |

#### CustomerDAO Class

| No | Method | Description |
| --- | --- | --- |
| 01 | +getCustomerByID(a.getCustomerID()) | Get customer information by customer’s ID- Input: Customer ID- Output: Customer information |

### 

#### Customer Class

| No | Method | Description |
| --- | --- | --- |
| 01 | +getCustomerID() | return customer ID of an instance derive from customer class- Input: None- Output: Customer ID |

### 

### c. Sequence Diagram(s)

### 

### 

### d. Database Queries

### [Provide the detailed SQL (select, insert, update...) which are used in implementing the function/screen]

### 

## *1.2 Update Profile*

### a. Class Diagram

### 

### b. Class Specifications

#### EditProfile Class

| No | Method | Description |
| --- | --- | --- |
| 01 | doGet | Called by the server (via the service method) to allow a servlet to handle a GET request. |
| 02 | doPost | Called by the server (via the service method) to allow a servlet to handle a POST request. |

#### 

#### AccountDAO Class

| No | Method | Description |
| --- | --- | --- |
| 01 | +editInfo(customer) | Update customer information by new customer class- Input: customer- Output: new customer information |

#### 

#### Customer Class

| No | Method | Description |
| --- | --- | --- |
| 01 | Customer(int CustomerID, String FirstName, String LastName, boolean Gender, String ContactTitle, Date DateOfBirth, String Address, String PhoneNumber, Date CreateDate) | return customer object of an instance derive from customer class- Input: None- Output: customers object |

### 

#### 

### c. Sequence Diagram(s)

### 

### d. Database Queries

### [Provide the detailed SQL (select, insert, update...) which are used in implementing the function/screen]

### 

## *1.3 View MyOrders*

### a. Class Diagram

### 

### b. Class Specifications

#### ViewProfile Class

| No | Method | Description |
| --- | --- | --- |
| 01 | doGet | Called by the server (via the service method) to allow a servlet to handle a GET request. |

#### 

#### OrderDAO Class

| No | Method | Description |
| --- | --- | --- |
| 01 | +getAllOrdersByCusID(getCustomerID())+getDetailOfOrderByCusID(getCustomerID()) | Update customer information by new customer class- Input: CustomerID- Output: Customer’s order |

#### 

#### Customer Class

| No | Method | Description |
| --- | --- | --- |
| 01 | getCustomerID() | return customer ID of an instance derive from customer class- Input: None- Output: Customer ID |

### c. Sequence Diagram(s)

### [Provide the sequence diagram(s) for the feature, see the sample below]

### 

### d. Database Queries

### [Provide the detailed SQL (select, insert, update...) which are used in implementing the function/screen]

### 

## *1.4. Change Order Status*

### a. Class Diagram

### [This part presents the class diagram for the relevant feature]

### 

### b. Class Specifications

### [Provide the description for each class and the methods in each class, following the table format as below]

#### XYZ Class

### [Provide the detailed description for the class methods]

| No | Method | Description |
| --- | --- | --- |
|  |  |  |
| 01 | <method name> | <Description of the method, including the inputs, outputs & internal method processing> |
|  |  |  |

### 

#### ABC Class

### Class Methods

### [Provide the detailed description for the class methods]

| No | Method | Description |
| --- | --- | --- |
| 01 | <method name> | <Description of the method, including the inputs, outputs & internal method processing> |
|  |  |  |

### 

### c. Sequence Diagram(s)

### [Provide the sequence diagram(s) for the feature, see the sample below]

### 

### d. Database Queries

### [Provide the detailed SQL (select, insert, update...) which are used in implementing the function/screen]

**1.5. OrderAction**

## a. Class Diagram

### 

### b. Class Specifications

#### OrderAction Class

| No | Method | Description |
| --- | --- | --- |
| 01 | doGet | Called by the server (via the service method) to allow a servlet to handle a GET request. |
| 02 | doPost | is Called by the server (via the service method) to allow a servlet to handle a POST request. |

### 

#### CartDAO Class

| No | Method | Description |
| --- | --- | --- |
| 01 | +getCartListByAccID(int accID) | Get cart list belonging to one person with accountID- Input: accountID- Output: list of products in cart |

#### Cart Class

| No | Method | Description |
| --- | --- | --- |
| 01 | getQuantity() | Get number of the same item in cart- Input: None- Output: number of the same item |

### 

#### ProductDAO Class

| No | Method | Description |
| --- | --- | --- |
| 01 | +getAvailableInStock(int productID) | Get the list of available inventories of a product- Input: productID- Output: number of available product |

#### SendMail Class

| No | Method | Description |
| --- | --- | --- |
| 01 | +sendAnnounce() | send an email to customer when they buy book successfully- Input: Order information- Output: None |

#### AccountDAO Class

| No | Method | Description |
| --- | --- | --- |
| 01 | +getAccountByEmail(String email) | Get account information by email address- Input: email address- Output: account information |

### 

#### Account Class

| No | Method | Description |
| --- | --- | --- |
| 01 | getAccountID() | return account ID of an instance derive from Account class- Input: None- Output: account ID |

#### OrderDAO Class

| No | Method | Description |
| --- | --- | --- |
| 01 | +createOrderInDB(Order ,int , String ) | create Order field and Order Details into Database- Input: Order information, accountID, voucher- Output: None |

### 

### c. Sequence Diagram(s)

### 

### 

### d. Database Queries

### 

**1.6. AddToCart**

## a. Class Diagram

### 

### b. Class Specifications

#### OrderAction Class

| No | Method | Description |
| --- | --- | --- |
| 01 | doGet | Called by the server (via the service method) to allow a servlet to handle a GET request. |
| 02 | doPost | Called by the server (via the service method) to allow a servlet to handle a POST request. |

### 

#### CartDAO Class

| No | Method | Description |
| --- | --- | --- |
| 01 | +getCartListByAccID(int accID) | Get cart list which belongs to one person with accountID- Input: accountID- Output: list of product in cart |
| 02 | +addToCart(Cart cart) | adding the product into the cart of the customer- Input: cart information- Output: True if add successful, False if fail |

#### Cart Class

| No | Method | Description |
| --- | --- | --- |
| 01 | getQuantity() | Get number of the same item in cart- Input: None- Output: number of the same item |

### 

#### ProductDAO Class

| No | Method | Description |
| --- | --- | --- |
| 01 | +getAvailableInStock(int productID) | Get the list of available inventories of a product- Input: productID- Output: number of available product |
| 02 | +getProducts(boolean isAdmin) | Get products list from Database- Input: isAdmin = true: get all product in Database isAdmin = false: get product list is in business currently- Output: list of Product |

#### AccountDAO Class

| No | Method | Description |
| --- | --- | --- |
| 01 | +getAccountByEmail(String email) | Get account information by email address- Input: email address- Output: account information |

### 

#### Account Class

| No | Method | Description |
| --- | --- | --- |
| 01 | getAccountID() | return account ID of an instance derive from Account class- Input: None- Output: account ID |

### c. Sequence Diagram(s)

### 

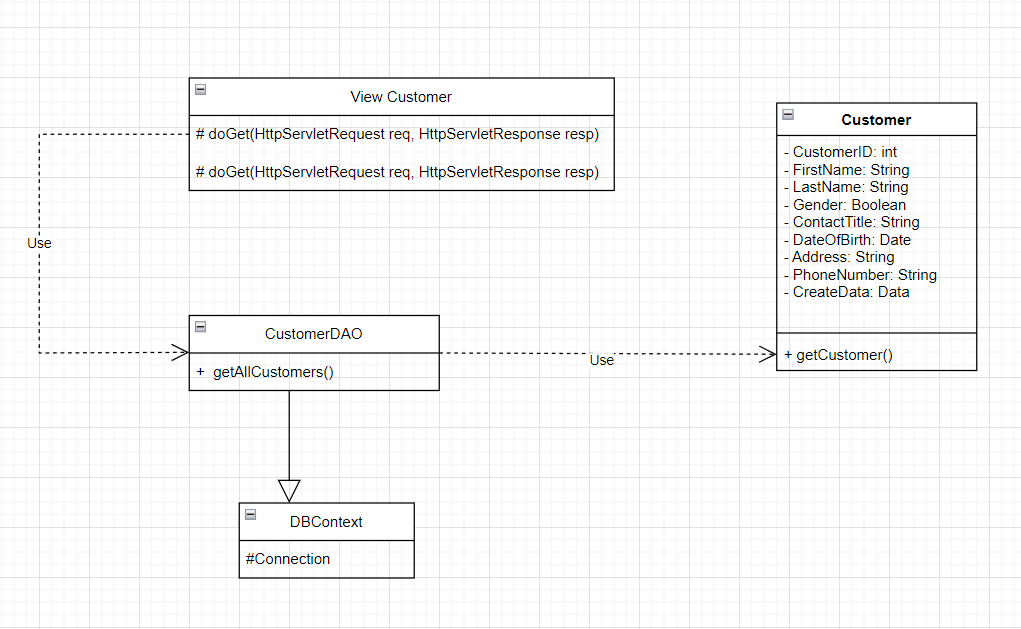
### 

### d. Database Queries

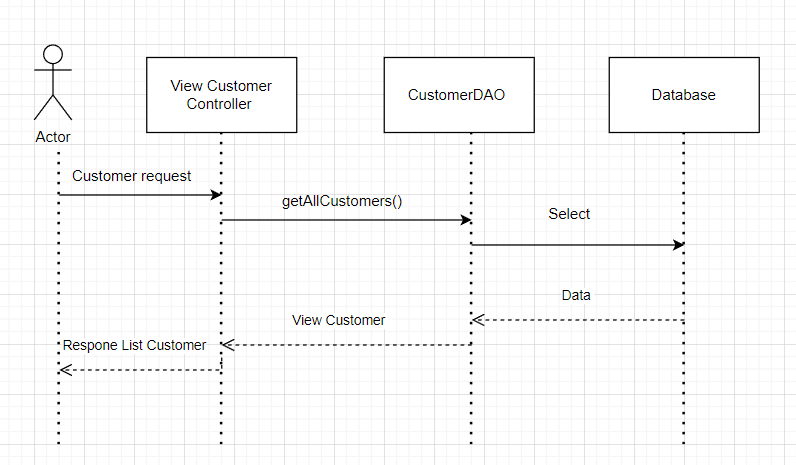
### 

### 1.7.

## a. Class Diagram



### b. Class Specifications



#### View Customer Controller Class

| No | Method | Description |
| --- | --- | --- |
| 01 | doGet | Called by the server (via the service method) to allow a servlet to handle a GET request. |

#### ProductDAO Class

| No | Method | Description |
| --- | --- | --- |
| 01 | getAllCustomers() | - Input: ID- Output: view List |

**2. Admin features**

***2.1 Dashboard***

## a. Class Diagram

### 

### b. Class Specifications

#### DashboardController Class

| No | Method | Description |
| --- | --- | --- |
| 01 | doGet | Called by the server (via the service method) to allow a servlet to handle a GET request. |
| 02 | doPost | Called by the server (via the service method) to allow a servlet to handle a POST request. |

### 

#### OrderDAO Class

| No | Method | Description |
| --- | --- | --- |
| 01 | getObjectList | Get the object list by ResultSet pass from getAuthorList()- Input: list ResultSet- Output: list of Author |
| 02 | getAuthorList | Get the Author list- Input: None- Output: list of Author |

#### Class

| No | Method | Description |
| --- | --- | --- |
| 01 | getCategory | Get the Category list- Input: None- Output: list of Category |

### 

#### CustomerDAOClass

| No | Method | Description |
| --- | --- | --- |
| 01 | getObjectList | Get the object list by ResultSet pass from getGenreList()- Input: list ResultSet- Output: list of Genre |
| 02 | getGenreList | Get the Genre list- Input: None- Output: list of Genre |

### 

### c. Sequence Diagram(s)

***2.2. View Products***

## a. Class Diagram

### 

### b. Class Specifications

#### Product Controller Class

| No | Method | Description |
| --- | --- | --- |
| 01 | doGet | Called by the server (via the service method) to allow a servlet to handle a GET request. |

### 

#### ProductDAO Class

| No | Method | Description |
| --- | --- | --- |
| 01 | getProductInfor(int proID) | Get product list by productID- Input: productID- Output: List product |

#### Class

| No | Method | Description |
| --- | --- | --- |
| 01 | getProductID | Get the ProductList- Input: None- Output: list of Products |

### 

### c. Sequence Diagram(s)

### 

### 

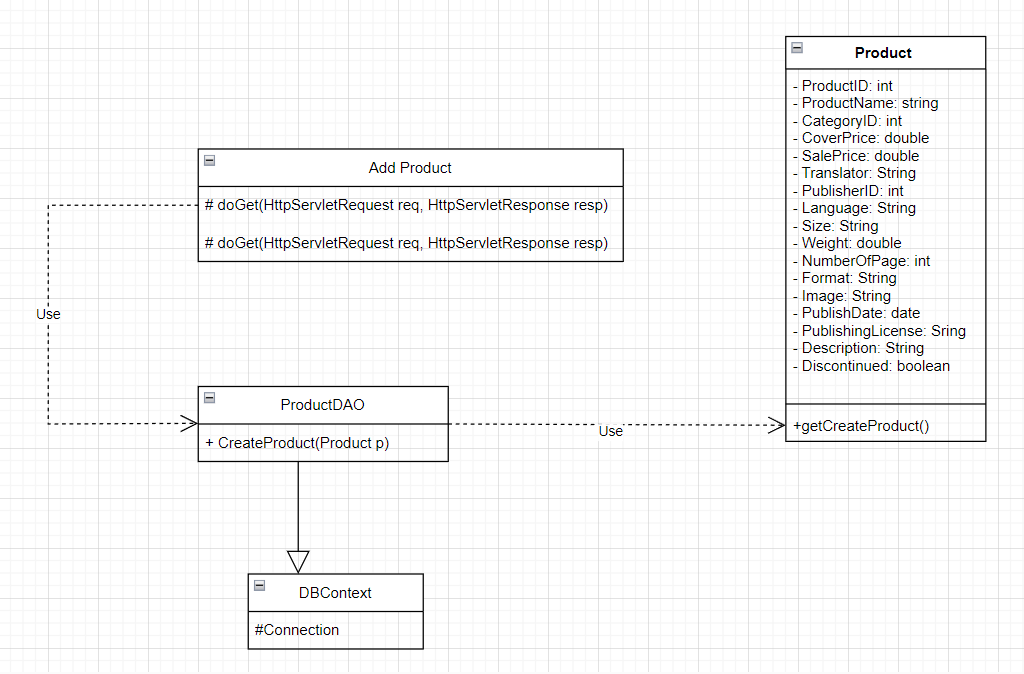
### d. Database Queries

### [Provide the detailed SQL (select, insert, update...) which are used in implementing the function/screen]

### 

***2.33. Add New Product***

**a. Class Diagram**



### b. Class Specifications

#### Product Controller Class

| No | Method | Description |
| --- | --- | --- |
| 01 | doGet | Called by the server (via the service method) to allow a servlet to handle a GET request. |

### 

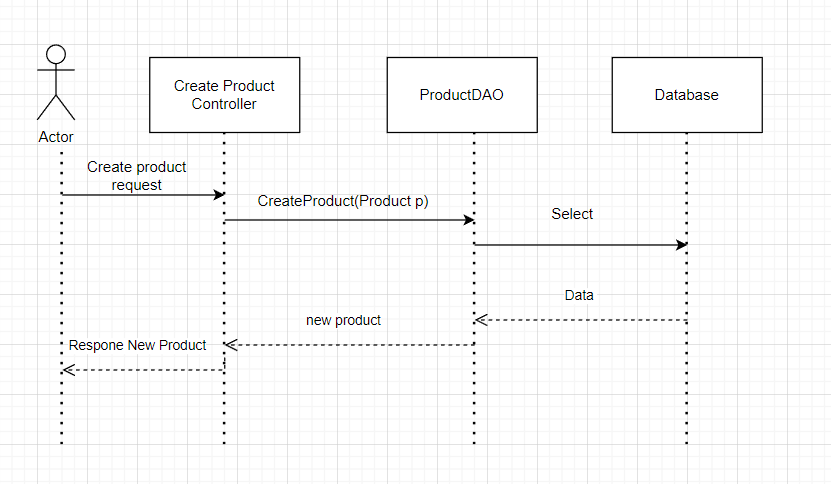
#### ProductDAO Class

| No | Method | Description |
| --- | --- | --- |
| 01 | update(Product p) | - Input: data- Output: product update |

#### Class

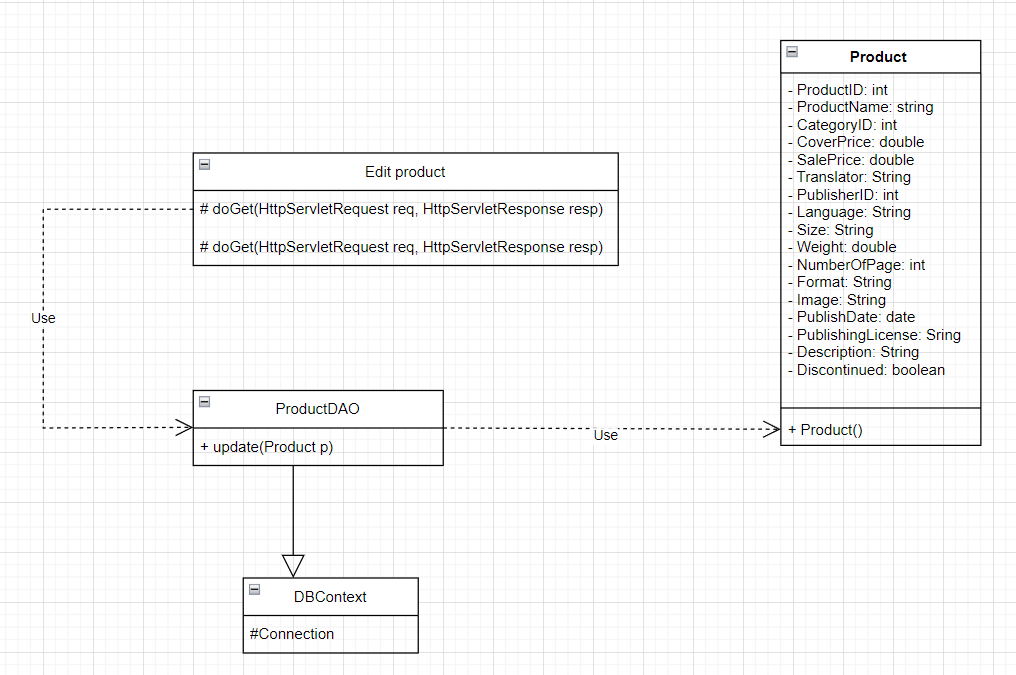
| No | Method | Description |
| --- | --- | --- |
| 01 | UpdateProduct | - Input: None- Output: list of Products |

### 

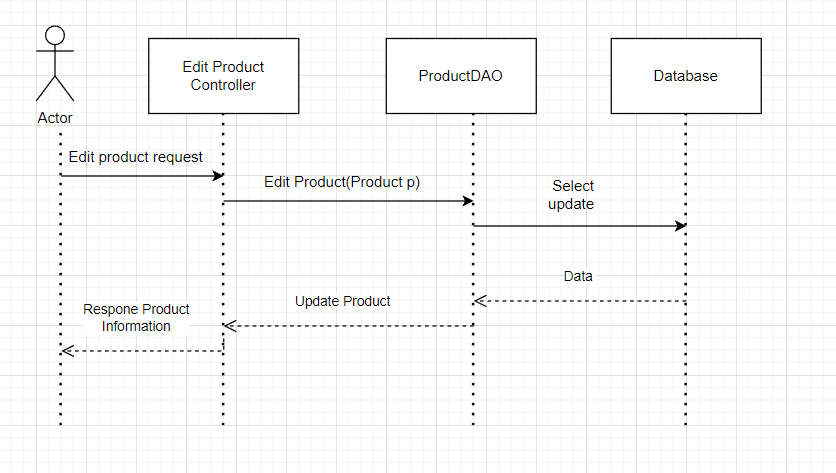


***2.4. Edit Product***

**a. Class Diagram**



### b. Class Specifications



## *4. Common features*

4.1 Homepage

## a. Class Diagram

### 

### b. Class Specifications

#### Homepage Class

| No | Method | Description |
| --- | --- | --- |
| 01 | doGet | Called by the server (via the service method) to allow a servlet to handle a GET request. |
| 02 | doPost | Called by the server (via the service method) to allow a servlet to handle a POST request. |

### 

#### AuthorDAO Class

| No | Method | Description |
| --- | --- | --- |
| 01 | getObjectList | Get the object list by ResultSet pass from getAuthorList()- Input: list ResultSet- Output: list of Author |
| 02 | getAuthorList | Get the Author list- Input: None- Output: list of Author |

#### CategoryDAO Class

| No | Method | Description |
| --- | --- | --- |
| 01 | getCategory | Get the Category list- Input: None- Output: list of Category |

### 

#### GenreDAO Class

| No | Method | Description |
| --- | --- | --- |
| 01 | getObjectList | Get the object list by ResultSet pass from getGenreList()- Input: list ResultSet- Output: list of Genre |
| 02 | getGenreList | Get the Genre list- Input: None- Output: list of Genre |

#### ProductDAO Class

| No | Method | Description |
| --- | --- | --- |
| 01 | getObjectList | Get the object list by ResultSet- Input: list ResultSet- Output: list of Product |
| 02 | getObject | Get the object- Input: None- Output: an instance of Product |
|  | getProducts(boolean isAdmin) | Get products- Input: isAdmin = true: get all product in Database isAdmin = false: get product list is in business currently- Output: list of Product |
|  | getComingSoon() | Get products which do not release- Input: None- Output: a list of Product which does not release |
|  | getProductbySearch(String sample) | Get products which is similar with a pattern- Input: None- Output: a list of Product |
|  | getNewReleaseList() | Get products which do not release yet- Input: sample (pattern) which user want to find out- Output: a list of Product which does not release |

### 

### c. Sequence Diagram(s)

### 

### 

### d. Database Queries

### [Provide the detailed SQL (select, insert, update...) which are used in implementing the function/screen]

### 

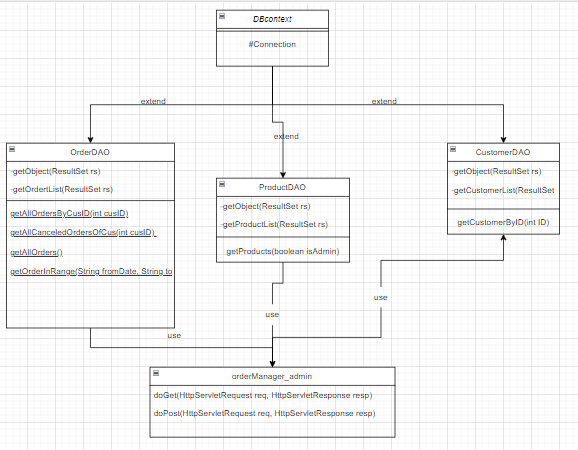
### 

### 

# 3.

**3.1: View Order**

### a. Class Diagram



### b. Class Specifications

#### orderManager\_admin Class

| No | Method | Description |
| --- | --- | --- |
| 01 | doGet | Called by the server (via the service method) to allow a servlet to handle a GET request. |
| 02 | doPost | is Called by the server (via the service method) to allow a servlet to handle a POST request. |

### 

#### OrderDAO Class

| No | Method | Description |
| --- | --- | --- |
| 01 | getObjectList | Get the object list by ResultSet pass from getOrderList()- Input: list ResultSet- Output: list of Order |
| 02 | getOrderList | Get the Orderlist- Input: None- Output: list of Order |
| 03 | getAllOrdersByCusID(int cusID) | Get all Order by cusID -input : Non  -output: List order by customer ID |
| 04 | getAllCanceledOrdersOfCus(int cusID) | Get all order has cancel by customerID -input: Non  -output: List order has Cancel in list order by customerID |
| 05 | getAllOrders() | Get the Orderlist- Input: None- Output: list all of Order |
| 06 | getOrderInRange(String fromDate, String toDate) | Get the list order in range by date -input: date(from-to)  -output: List order in range (from - to) date |

#### ProductDAO Class

| No | Method | Description |
| --- | --- | --- |
| 01 | getObjectList | Get the products list by ResultSet pass from getProductList()- Input: list ResultSet- Output: list of Product |
| 02 | getProductList | Get the Productlist- Input: None- Output: list of Products |
| 03 | getProducts(boolean isAdmin) | Get products- Input: isAdmin = true: get all product in Database isAdmin = false: get product list is in business currently- Output: list of Product |

### 

#### CustomerDAO Class

| No | Method | Description |
| --- | --- | --- |
| 01 | getObjectList | Get the Customer list by ResultSet pass from getCustomer()- Input: list ResultSet- Output: list of Customer |
| 02 | getCustomerList | Get the Customerlist- Input: None- Output: list of Customer |
| 03 | getCustomerByID(int ID) | get the CustomerID -input :non  -output: list ID of CustomerID |

### 

### c. Sequence Diagram(s)