

**Online shopping system**

**Software Design Specification**

**Team 1**

– Hanoi, Feb 2023 –

Record of changeS

|  |  |  |  |
| --- | --- | --- | --- |
| Date | A\* M, D | In charge | Change Description |
| 17/2 | A | GiangNXT | Add DataBase Design (I.2) |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

\*A - Added M - Modified D - Deleted

**Table of Contents**

[I. Overview 4](#_Toc96516286)

[1. Code Packages 4](#_Toc96516287)

[2. Database Design 4](#_Toc96516288)

[a. Database Schema 4](#_Toc96516289)

[b. Table Description 4](#_Toc96516290)

[II. Code Designs 5](#_Toc96516291)

[1. <Feature/Function Name1> 5](#_Toc96516292)

[a. Class Diagram 5](#_Toc96516293)

[b. Class Specifications 5](#_Toc96516294)

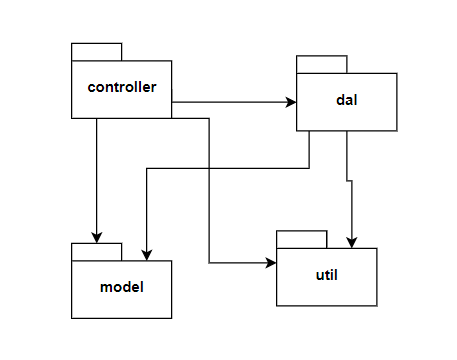
[c. Sequence Diagram(s) 5](#_Toc96516295)

[d. Database queries 6](#_Toc96516296)

[2. <Feature/Function Name2> 6](#_Toc96516297)

# I. Overview

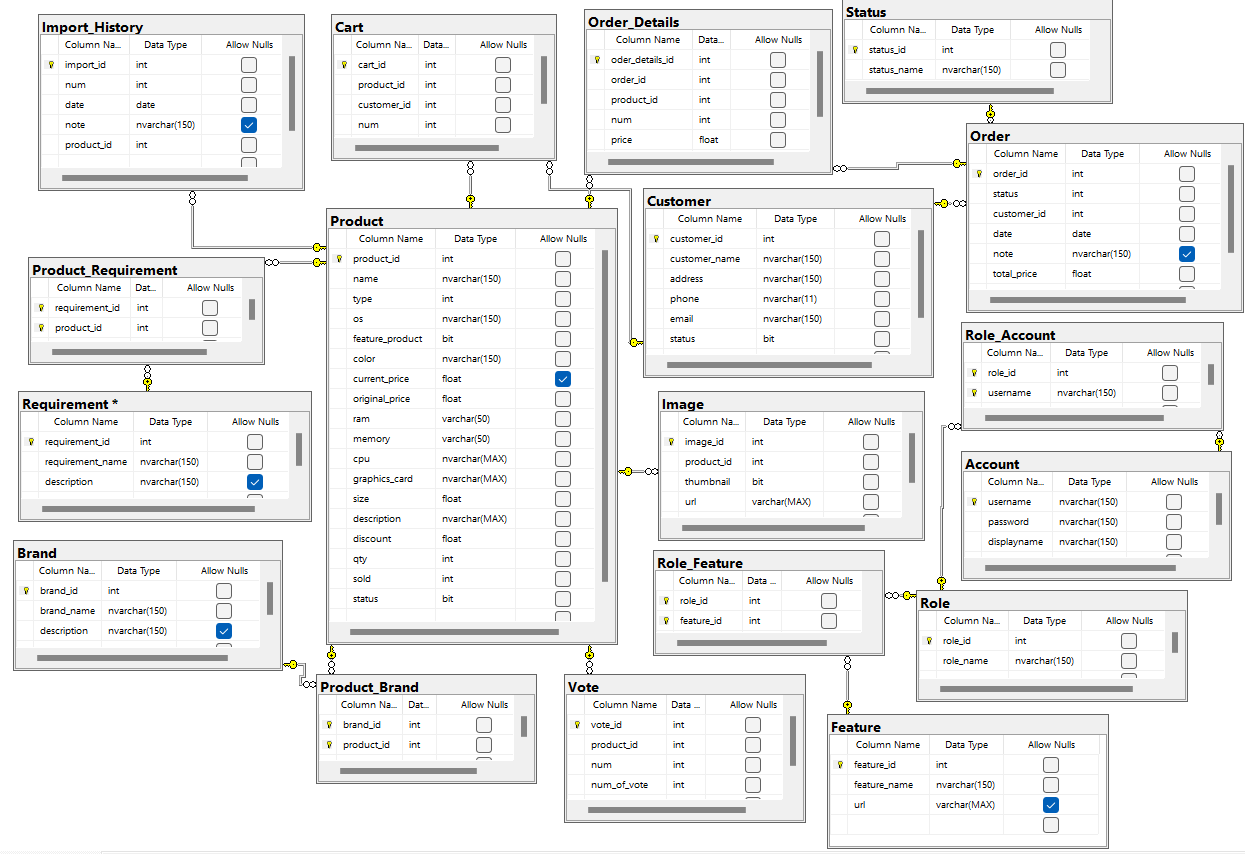
## 1. Code Packages



***Package descriptions***

|  |  |  |
| --- | --- | --- |
| **No** | **Package** | **Description** |
| 1 | Controller | Data processing and navigate to jsp |
| 2 | Dal | Interact with data base |
| 3 | Model | Include objectives |
| 4 | Util | Function to support data processing |

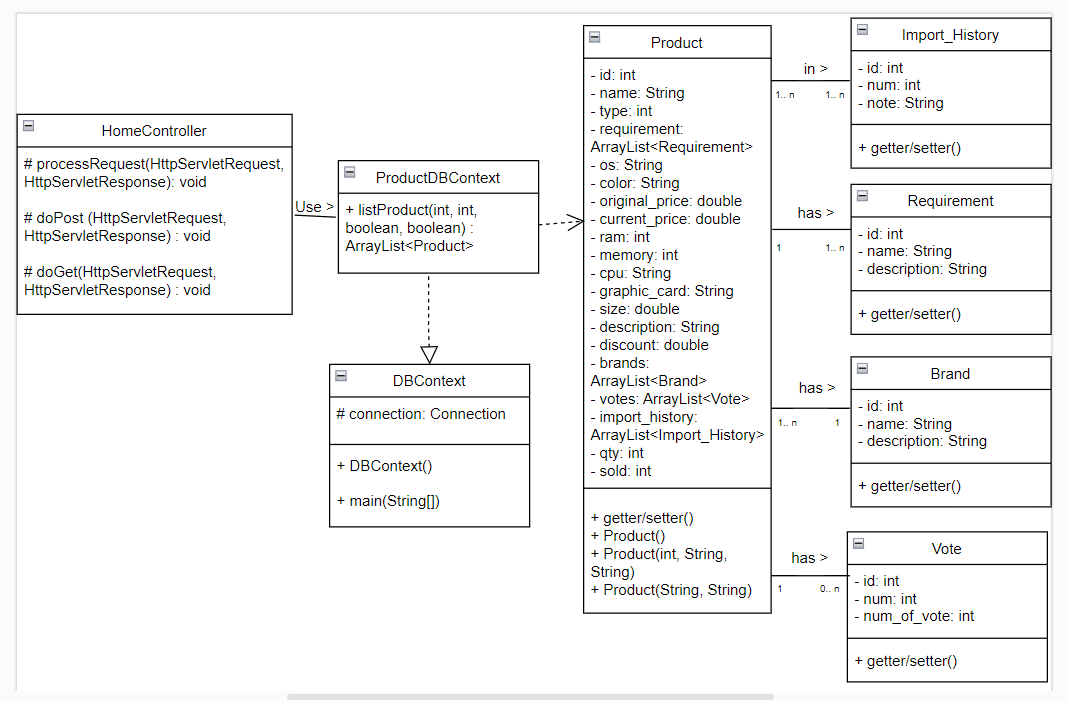
## 2. Database Design



# II. Code Designs

## 1. Home Page – Display Product Lists

### a. Class Diagram



### b. Class Specifications

#### HomeController

***Class Methods***

|  |  |  |
| --- | --- | --- |
| **No** | **Method** | **Description** |
| 01 | processRequest(HttpServletRequest, HttpServletResponse): void | To handle request, get data from database and display to jsp |
| 02 | doPost(HttpServletRequest, HttpServletResponse): void | Call processRequest(HttpServletRequest, HttpServletResponse) |
| 03 | doGet(HttpServletRequest, HttpServletResponse): void | Call processRequest(HttpServletRequest, HttpServletResponse) |

#### ProductDBContext

***Class Methods***

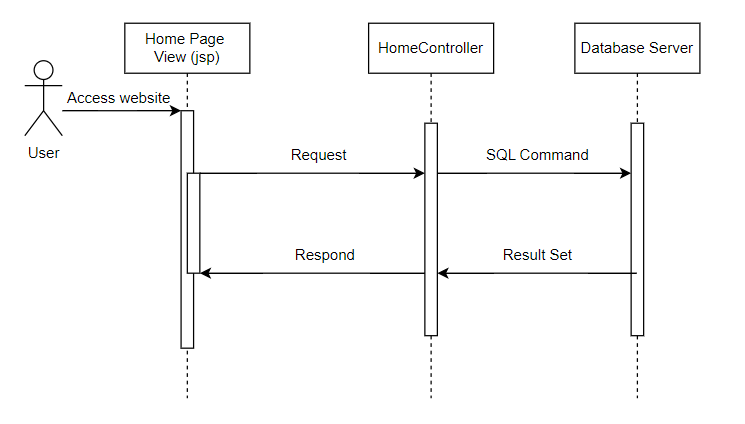
|  |  |  |
| --- | --- | --- |
| **No** | **Method** | **Description** |
| 01 | listProduct(int, int, boolean, boolean): ArrayList<Product> | Get list of product base on: what type? (phone or laptop), how much? (number of product), Is top of sale? Is top of sold? |

#### DBContext

***Class Methods***

|  |  |  |
| --- | --- | --- |
| **No** | **Method** | **Description** |
| 01 | DBContext() | To connect with database |
| 02 | main(String[]) | Call DBContext() |

### c. Sequence Diagram(s)



### d. Database Queries

//Select top 8 of best-seller

SELECT TOP 8 [product\_id], [name], [type], [os], [color], [current\_price], [original\_price], [ram]

, [memory], [cpu] , [graphics\_card], [size], [description] , [discount] , [qty] , [sold]

, [status]

FROM [Product] where [status] = 1

ORDER BY [sold] DESC

//Select top 8 of promotional product

SELECT TOP 8 [product\_id], [name], [type], [os], [color], [current\_price], [original\_price], [ram]

, [memory], [cpu] , [graphics\_card], [size], [description] , [discount] , [qty] , [sold]

, [status]

FROM [Product] where [status] = 1

ORDER BY [discount] DESC

// Select first of 8 phones

SELECT TOP 8 [product\_id], [name], [type], [os], [color], [current\_price], [original\_price], [ram]

, [memory], [cpu] , [graphics\_card], [size], [description] , [discount] , [qty] , [sold]

, [status]

FROM [Product] where [status] = 1 AND [type] = 0

//Select first of 8 laptops

SELECT TOP 8 [product\_id], [name], [type], [os], [color], [current\_price], [original\_price], [ram]

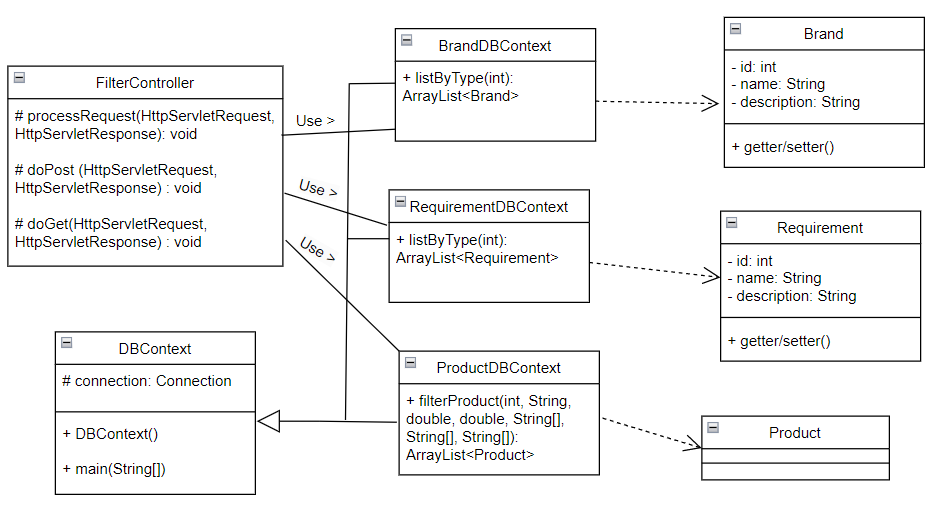
, [memory], [cpu] , [graphics\_card], [size], [description] , [discount] , [qty] , [sold]

, [status]

FROM [Product] where [status] = 1 AND [type] = 1

## 2. Product List - Filter

### a. Class Diagram



### b. Class Specifications

#### DBContext

***Class Methods***

|  |  |  |
| --- | --- | --- |
| **No** | **Method** | **Description** |
| 01 | DBContext() | To connect with database |
| 02 | main(String[]) | Call DBContext() |

#### FilterController

***Class Methods***

|  |  |  |
| --- | --- | --- |
| **No** | **Method** | **Description** |
| 01 | processRequest(HttpServletRequest, HttpServletResponse): void | To handle request, get data from database and display to jsp |
| 02 | doPost(HttpServletRequest, HttpServletResponse): void | Call processRequest(HttpServletRequest, HttpServletResponse) |
| 03 | doGet(HttpServletRequest, HttpServletResponse): void | Call processRequest(HttpServletRequest, HttpServletResponse) |

#### BrandDBContext

***Class Methods***

|  |  |  |
| --- | --- | --- |
| **No** | **Method** | **Description** |
| 01 | listByType(int): ArrayList<Brand> | Get list of brands by category product (phone or laptop) |

#### RequirementDBContext

***Class Methods***

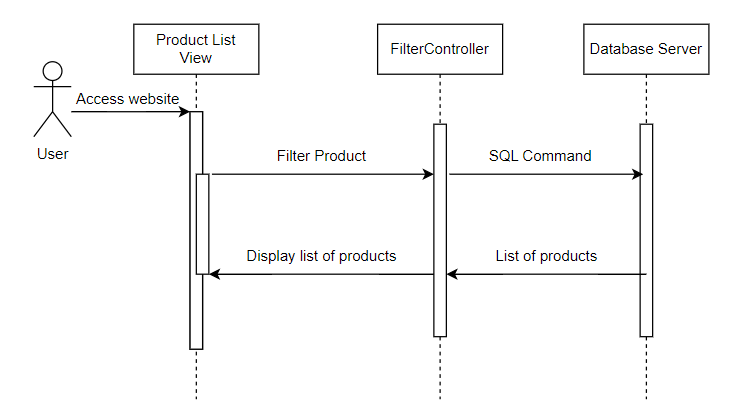
|  |  |  |
| --- | --- | --- |
| **No** | **Method** | **Description** |
| 01 | listByType(int): ArrayList<Requirement> | Get list of requirements by category product (phone or laptop) |

#### ProductDBContext

***Class Methods***

|  |  |  |
| --- | --- | --- |
| **No** | **Method** | **Description** |
| 01 | filterProduct(int, String, double, double, String[], String[], String[]): ArrayList<Product> | Get list of products by filter |

### c. Sequence Diagram(s)



### d. Database Queries

//Select all brand of laptop or iphone

SELECT DISTINCT b.[brand\_id], b.[brand\_name], b.[description]

FROM [Brand] b

INNER JOIN Product\_Brand pb ON pb.brand\_id = b.brand\_id

INNER JOIN Product pr ON pr.product\_id = pb.product\_id

WHERE pr.[type] =?

//Select all requirement of laptop or iphone

SELECT DISTINCT p.[requirement\_id], p.[requirement\_name], p.[description]

FROM [Requirement] p

INNER JOIN Product\_Requirement pr ON pr.requirement\_id = p.requirement\_id

INNER JOIN Product prd ON prd.product\_id = pr.product\_id

WHERE prd.[type] = ?

//Select all product match with filter

SELECT DISTINCT pr.[product\_id], pr.[name], pr.[type], pr.[os], pr.[feature\_product], pr.[color]

, pr.[current\_price], pr.[original\_price], pr.[ram], pr.[memory], pr.[cpu]

, pr.[graphics\_card], pr.[size], pr.[description], pr.[discount], pr.[qty]

, pr.[sold], pr.[status]

FROM [Product] pr

Inner Join [Product\_Brand] prbr On pr.product\_id = prbr.product\_id

Inner Join [Brand] br on prbr.brand\_id = br.brand\_id

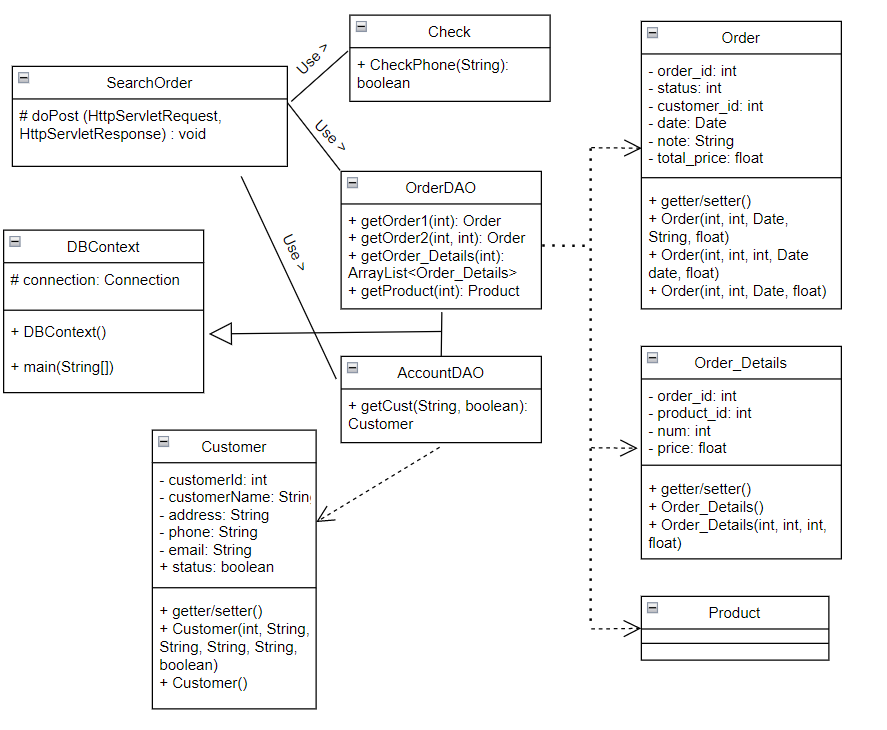
Inner Join [Product\_Requirement] prre On pr.product\_id = prre.product\_id

Inner Join [Requirement] re on prre.requirement\_id =re.requirement\_id

where pr.[status] = 1 AND pr.[current\_price] >= ? AND pr.[current\_price] <= ? AND pr.[type] = ?

## 3. Search Order

### a. Class Diagram



### b. Class Specifications

#### DBContext

***Class Methods***

|  |  |  |
| --- | --- | --- |
| **No** | **Method** | **Description** |
| 01 | DBContext() | To connect with database |
| 02 | main(String[]) | Call DBContext() |

#### Check

***Class Methods***

|  |  |  |
| --- | --- | --- |
| **No** | **Method** | **Description** |
| 01 | CheckPhone(String): boolean | Validate input is number or not |

#### OrderDAO

***Class Methods***

|  |  |  |
| --- | --- | --- |
| **No** | **Method** | **Description** |
| 01 | getOrder1(int): Order | Get order by order id |
| 02 | getOrder2(int, int): Order | Get order by order id and customer id |
| 03 | getOrder\_Details(int):  ArrayList<Order\_Details> | Get list of Order Details by order id |
| 04 | getProduct(int): Product | Get product by product id |

AccountDAO

***Class Methods***

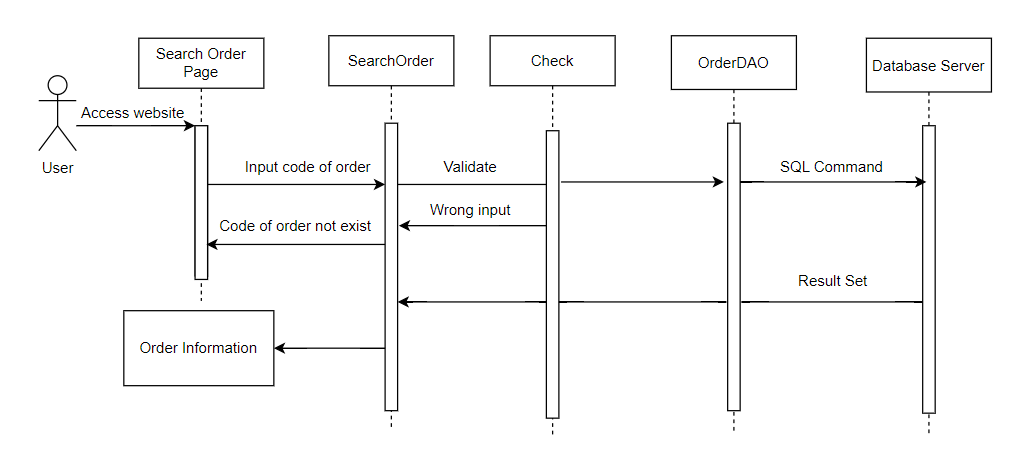
|  |  |  |
| --- | --- | --- |
| **No** | **Method** | **Description** |
| 01 | getCust(String, boolean): Customer | Get Customer by email and status |

#### SearchOrder

***Class Methods***

|  |  |  |
| --- | --- | --- |
| **No** | **Method** | **Description** |
| 01 | doPost (HttpServletRequest, HttpServletResponse): void | Handle request by doPost method |

### c. Sequence Diagram(s)



### d. Database Queries

//Select order by order id

Select \* from [Order] where order\_id = ?

//Select order by order id and customer id

Select \* from [Order] where order\_id = ? and customer\_id = ?

//Select Order Details by order id

Select \* from Order\_Details where order\_id =?

//Select product by product id

Select [name],color from Product where product\_id = ?

//Select Customer by email and status

SELECT [customer\_id], [customer\_name], [address], [phone], [email], [status]

FROM [dbo].[Customer]

WHERE [email] = ? and status =?

## 4. Add to Cart

### a. Class Diagram

### b. Class Specifications

#### DBContext

***Class Methods***

|  |  |  |
| --- | --- | --- |
| **No** | **Method** | **Description** |
| 01 | DBContext() | To connect with database |
| 02 | main(String[]) | Call DBContext() |

### c. Sequence Diagram(s)

### d. Database Queries

## 5. <Feature/Function Name2>

### a. Class Diagram

### b. Class Specifications

#### DBContext

***Class Methods***

|  |  |  |
| --- | --- | --- |
| **No** | **Method** | **Description** |
| 01 | DBContext() | To connect with database |
| 02 | main(String[]) | Call DBContext() |

### c. Sequence Diagram(s)

### d. Database Queries

## 6. <Feature/Function Name2>

### a. Class Diagram

### b. Class Specifications

#### DBContext

***Class Methods***

|  |  |  |
| --- | --- | --- |
| **No** | **Method** | **Description** |
| 01 | DBContext() | To connect with database |
| 02 | main(String[]) | Call DBContext() |

### c. Sequence Diagram(s)

### d. Database Queries

## 7. <Feature/Function Name2>

### a. Class Diagram

### b. Class Specifications

#### DBContext

***Class Methods***

|  |  |  |
| --- | --- | --- |
| **No** | **Method** | **Description** |
| 01 | DBContext() | To connect with database |
| 02 | main(String[]) | Call DBContext() |

### c. Sequence Diagram(s)

### d. Database Queries