

**online shop**

**Software Design Document**

– Hanoi, May 2021–

**Table of Contents**

[I. Overview 3](#_Toc69802609)

[1. Code Packages 3](#_Toc69802610)

[2. Database Schema 4](#_Toc69802611)

[II. Code Designs 4](#_Toc69802612)

[1. <Feature/Function Name1> 4](#_Toc69802613)

[a. Class Diagram 4](#_Toc69802614)

[b. Class Specifications 4](#_Toc69802615)

[c. Sequence Diagram(s) 4](#_Toc69802616)

[d. Database queries 5](#_Toc69802617)

[2. <Feature/Function Name2> 5](#_Toc69802618)

[III. Database Tables 6](#_Toc69802619)

[1. Product 6](#_Toc69802620)

[2. User 6](#_Toc69802621)

[3. Category 7](#_Toc69802620)

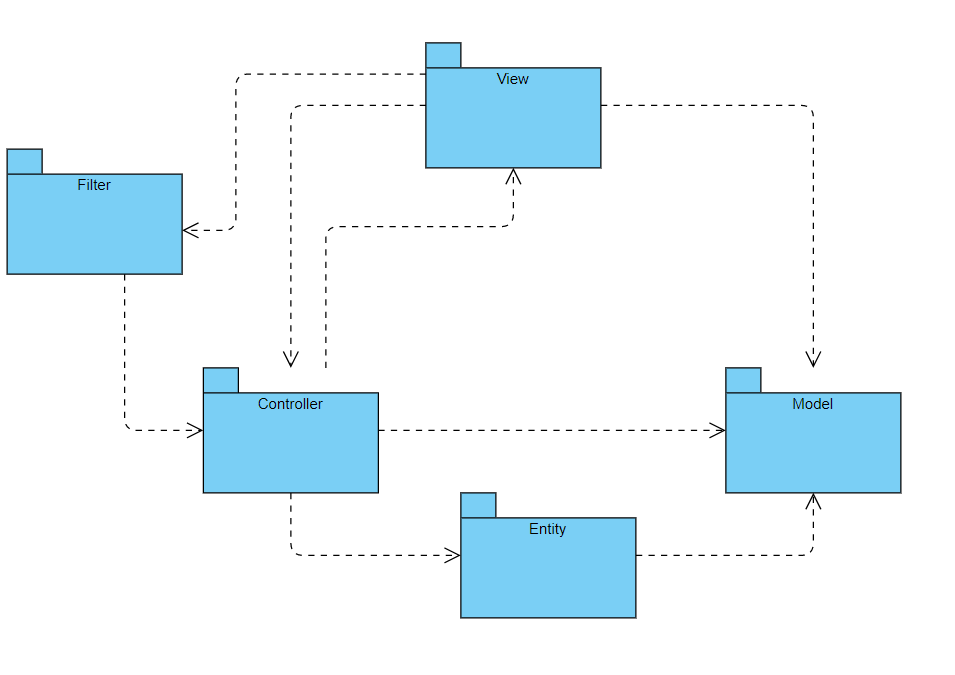
[4. Information 7](#_Toc69802621)

[5. Cart 7](#_Toc69802620)

[6. Ship 7](#_Toc69802621)

# I. Overview

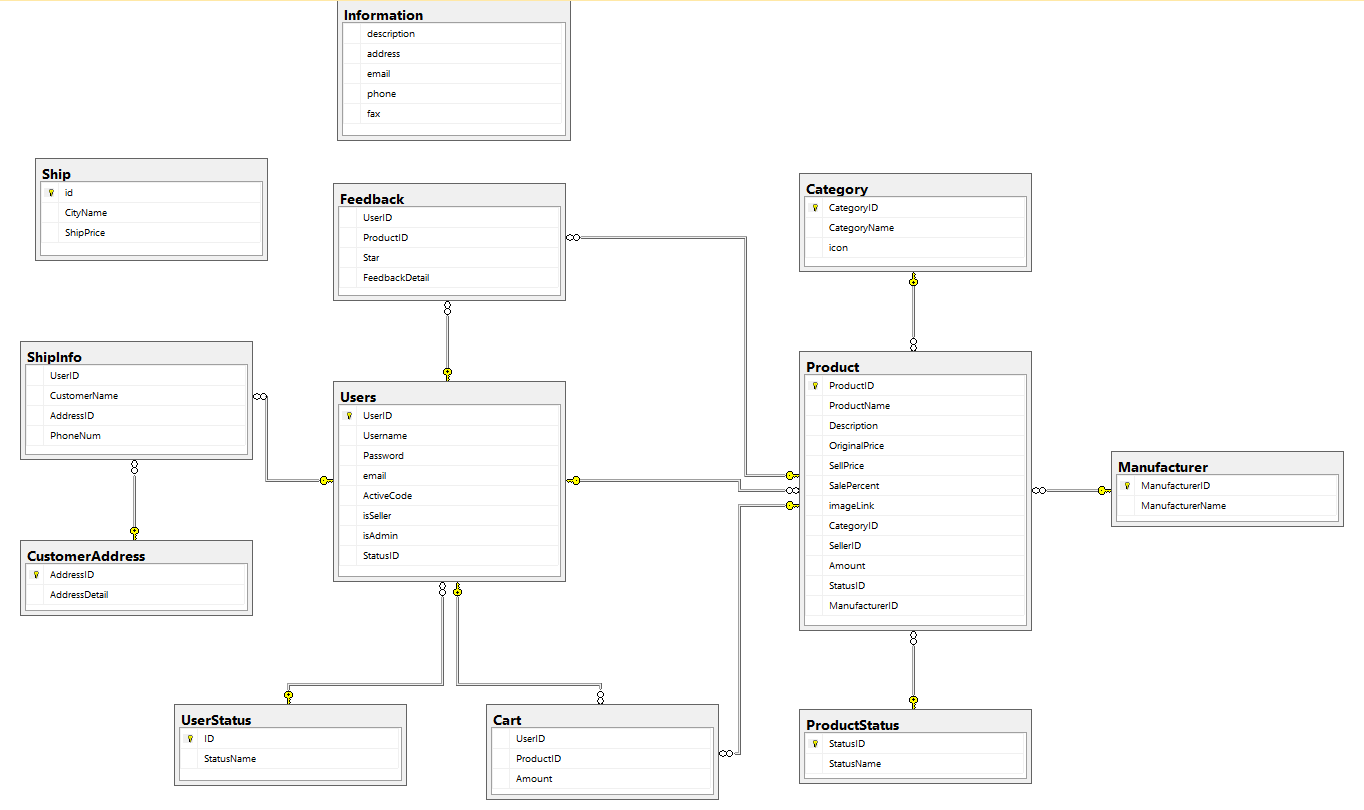
## 1. Code Packages/Namespaces



***Package descriptions & package class naming conventions***

|  |  |  |
| --- | --- | --- |
| **No** | **Package** | **Description** |
| *01* | *View (Web Pages)* | *Provide interaction with user, GUI*  *Java Class Naming Convention* |
| *02* | *Controller* | *Control the data flow between View and Model (DAO), Control the logic of the system*  *Java Class Naming Convention* |
| *03* | *DAO (Model)* | *Provide the connection to the Database, Extract data for the system* |
| *04* | *Filter* | *Control the redirection between Pages in View and from View to Controller*  *Java Class Naming Convention* |
| *05* | *Entity* | *Define the logic of data extracted from Database*  *Java Class Naming Convention* |

## 2. Database Schema



***Table descriptions & package class naming conventions are as below***

|  |  |  |
| --- | --- | --- |
| **No** | **Table** | **Description** |
| *01* | *Product* | *Store the information of all products*  *- Primary keys: ProductID*  *- Foreign keys: CategoryID; SellerID* |
| *02* | *Users* | *Store the information of all users (including customers, sellers and admin)*  *- Primary keys: UserID* |
| *03* | *Cart* | *Store information of the Cart (a list of product) of each Customer*  *- Foreign keys: UserID, ProductID* |
| *04* | *Category* | *Store the information of Categories of products*  *- Primary keys: CategoryID* |
| *05* | *Ship* | *Store the price of ship for each City*  *- Primary keys: CityName* |
| *06* | *Information* | *Store some information of the shop* |

# II. Code Designs

## 1. Home Page

### a. Class Diagram

### b. Class Specifications

#### HomePageControl

|  |  |  |
| --- | --- | --- |
| **No** | **Method** | **Description** |
| *01* | *processRequest* | *Get the data from the database and dump it on the homepage to display the store's user interface* |

#### ProductDAO,InfoDAO,CategoryDAO,UserDAO

***Class Methods***

|  |  |  |
| --- | --- | --- |
| **No** | **Method** | **Description** |
| *01* | *getAllCategory()* | *Create function to get All Category* |
| *02* | *getHotProduct()* | *Get First Product* |
| *03* | *getFavoriteProduc()* | *Get Last Product* |
| *04* | *getInfor()* | *Get Information* |
| *05* | *countProductByCategory(CID)* | *If the number of Product isn't divided by 3 -> Need 1 more Page/If the number of Product isn't divided by 3 -> Need 1 more Page* |
| *06* | *pagingByCategory()* | *List of Product to Display after Paging by Category ID* |

### c. Sequence Diagram(s)

### d. Database queries

## 2. Product List

#### Products Class

|  |  |  |
| --- | --- | --- |
| **No** | **Method** | **Description** |
| *01* | *getID* | *get ID Products* |
| *02* | *setID* | *set ID Products* |
| *03* | *getName()* | *get Name Products* |
| *04* | *setName()* | *set Name Products* |
| *05* | *getPrice()* | *get Price Products* |
| *06* | *setPrice()* | *set Price Products* |
| *07* | g*etImageLink()* | *get ImageLink Products* |
| *08* | *setImageLink* | *set ImageLink Products* |
| *09* | *getDescription()* | *get Description Products* |
| *10* | *setDescription()* | *set Description Products* |
| *11* | *ToString()* | *Output information* |
| *12* | *getPriceWithDot()* | *Function to write the Product's Price in proper form* |

## 3. <Product Detail >

#### DetailControl

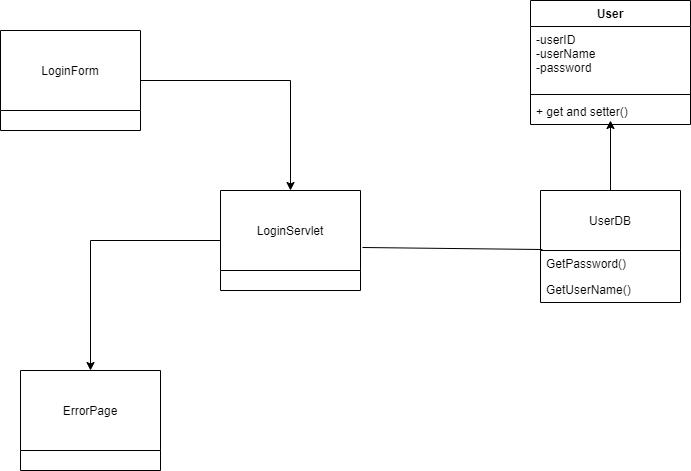
|  |  |  |
| --- | --- | --- |
| **No** | **Method** | **Description** |
| *01* | *processRequest()* | *Get data from homepag when the user returns and then transfer it to ProductDAO to get the product's data>* |

ProductDAO()

|  |  |  |
| --- | --- | --- |
| **No** | **Method** | **Description** |
| *01* | *getProductByID(id)* | *Get Product for Detail* |

## 4. User Login

### a. Class Diagram



### b. Class Specifications

|  |  |  |
| --- | --- | --- |
| **No** | **Method** | **Description** |
| *01* | *doGet()* | *get User and Pass information from Cookie and push to Login.jsp page to display* |
| *02* | *doPost()* | *Log in with the user and password available in the database* |
|  |  |  |

Account class

|  |  |  |
| --- | --- | --- |
| **No** | **Method** | **Description** |
| *01* | *getID()* | *<* *Get ID Account>* |
| *02* | *setID()* | *<Get ID Account>* |
| *03* | *getName()* | *<Get Name Accont>* |
| *04* | *setName()* | *<Set Name Account>* |

### c. Sequence Diagram(s)

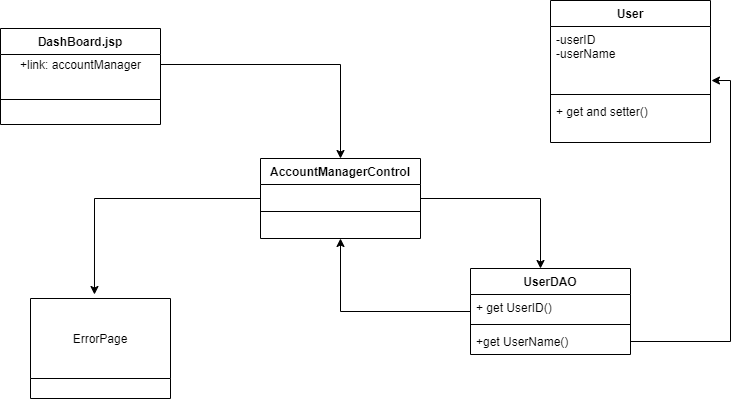
## 

### d. Database queries

-SELECT \* FROM Users WHERE Username = ? AND Password = ?

## 5. User List

### a. Class Diagram



### b. Class Specifications

|  |  |  |
| --- | --- | --- |
| **No** | **Method** | **Description** |
| *01* | *processRequest()* | *<get all accounts from the database to display on AccountManager.jsp>* |

Account class

|  |  |  |
| --- | --- | --- |
| **No** | **Method** | **Description** |
| *01* | *getID()* | *Get ID Account* |
| *02* | *setID* | *Get ID Account* |
| *03* | *getName()* | *Get Name Accont* |
| *04* | *setName()* | *Set Name Account* |

# III. Database Tables

## 1. User Status

*Store the status of an Account*

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **#** | **Field name** | **Type** | **Size** | **Unique** | **Not Null** | **PK/FK** | **Notes** |
| 1 | ID | int |  |  | Yes | PK |  |
| 2 | StatusName | nvarchar | 1000 |  |  |  |  |

## 2. Users

*Store the information of all users*

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **#** | **Field name** | **Type** | **Size** | **Unique** | **Not Null** | **PK/FK** | **Notes** |
| 1 | UserID | int |  |  |  | PK |  |
| 2 | Username | nvarchar | 500 |  |  |  |  |
| 3 | Password | nvarchar | 1000 |  |  |  |  |
| 4 | email | nvarchar | 1000 |  |  |  |  |
| 5 | ActiveCode | nvarchar | 11 |  |  |  |  |
| 4 | isSeller | int |  |  |  |  |  |
| 5 | isAdmin | int |  |  |  |  |  |
| 6 | StatusID | int |  |  |  | FK |  |

## 3. Category

*Store the information of all categories of products*

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **#** | **Field name** | **Type** | **Size** | **Unique** | **Not Null** | **PK/FK** | **Notes** |
| 1 | CategoryID | int |  |  |  | PK |  |
| 2 | CategoryName | nvarchar | 1000 |  |  |  |  |
| 3 | icon | nvarchar | 1000 |  |  |  |  |

## 4. Product Status

*Store the status of a Product (out of stock or not)*

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **#** | **Field name** | **Type** | **Size** | **Unique** | **Not Null** | **PK/FK** | **Notes** |
| 1 | StatusID | int |  |  | Yes | PK |  |
| 2 | StatusName | nvarchar | 1000 |  |  |  |  |

## 5. Manufacturer

*Store the information of the Manufacturer of a Product*

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **#** | **Field name** | **Type** | **Size** | **Unique** | **Not Null** | **PK/FK** | **Notes** |
| 1 | ManufacturerID | int |  |  | Yes | PK |  |
| 2 | ManufacturerName | nvarchar | 1000 |  |  |  |  |

## 6. Product

*Store the information of all products*

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **#** | **Field name** | **Type** | **Size** | **Unique** | **Not Null** | **PK/FK** | **Notes** |
| 1 | ProductID | int |  |  | Yes | PK |  |
| 2 | ProductName | nvarchar | 1000 |  |  |  |  |
| 3 | Description | nvarchar | 2000 |  |  |  |  |
| 4 | OriginalPrice | int |  |  |  |  |  |
| 5 | SellPrice | int |  |  |  |  |  |
| 6 | SalePercent | int |  |  |  |  | SalePercent>=1 AND SalePercent<=100 |
| 5 | imageLink | nvarchar | 1000 |  |  |  |  |
| 6 | CategoryID | int |  |  |  | FK |  |
| 7 | SellerID | int |  |  |  | FK |  |
| 8 | Amount | int |  |  |  |  |  |
| 9 | StatusID | int |  |  |  | FK |  |
| 10 | ManufacturerID | int |  |  |  | FK |  |

## 7. Information

*Store some information of the shop*

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **#** | **Field name** | **Type** | **Size** | **Unique** | **Not Null** | **PK/FK** | **Notes** |
| 1 | description | nvarchar | max |  | Null |  |  |
| 2 | address | nvarchar | max |  | Null |  |  |
| 3 | email | nvarchar | max |  | Null |  |  |
| 4 | phone | nvarchar | max |  | Null |  |  |
| 5 | fax | nvarchar | max |  | Null |  |  |

## 8. Cart

*Store information of the Cart of each Customer*

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **#** | **Field name** | **Type** | **Size** | **Unique** | **Not Null** | **PK/FK** | **Notes** |
| 1 | UserID | int |  |  |  | FK |  |
| 2 | ProductID | int |  |  |  | FK |  |
| 3 | Amount | int |  |  |  |  |  |

## 9. Ship

*Store the ship Price of each City*

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **#** | **Field name** | **Type** | **Size** | **Unique** | **Not Null** | **PK/FK** | **Notes** |
| 1 | ID | int |  |  |  | PK |  |
| 2 | CityName | nvarchar | 1000 |  |  |  |  |
| 3 | ShipPrice | int |  |  |  |  |  |

## 10. Customer Address

*Store the Shipping Address of Customers*

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **#** | **Field name** | **Type** | **Size** | **Unique** | **Not Null** | **PK/FK** | **Notes** |
| 1 | ID | int |  |  |  | PK |  |
| 2 | CityName | nvarchar | 1000 |  |  |  |  |
| 3 | ShipPrice | int |  |  |  |  |  |

## 11. Ship Info

*Store the ship price of each City*

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **#** | **Field name** | **Type** | **Size** | **Unique** | **Not Null** | **PK/FK** | **Notes** |
| 1 | UserID | int |  |  |  | FK |  |
| 2 | CustomerName | nvarchar | 1000 |  |  |  |  |
| 3 | AddressID | int |  |  |  | FK |  |
| 4 | PhoneNum | Varchar | 20 |  |  |  |  |

## 12. Feedback

*Store the information of a Feedback of a Customer*

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **#** | **Field name** | **Type** | **Size** | **Unique** | **Not Null** | **PK/FK** | **Notes** |
| 1 | UserID | int |  |  |  | FK |  |
| 2 | ProductID | int | 1000 |  |  | FK |  |
| 3 | Star | int |  |  |  |  | Star < 6 AND Star > 0 |
| 4 | FeedbackDetail | nvarchar | 2000 |  |  |  |  |