

**<<F-Gear>>**

**Software Design Document**

– Ho Chi Minh, September 2022 –

**Table of Contents**

[I. Overview 3](#_Toc69802609)

[1. Code Packages 3](#_Toc69802610)

[2. Database Schema 3](#_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 5](#_Toc69802619)

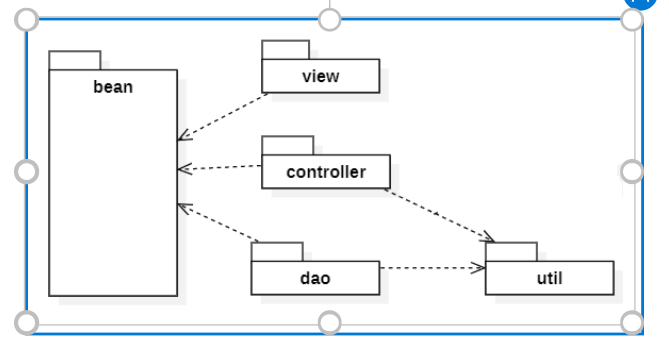
[1. <Table name 1> 5](#_Toc69802620)

[2. <Table name 2…> 5](#_Toc69802621)

# I. Overview

## 1. Code Packages/Namespaces

*[Provide the package diagram for each sub-system. The content of this section including the overall package diagram, the explanation, package and class naming conventions in each package. Please see the sample and description table format below – following Java project naming convention]*

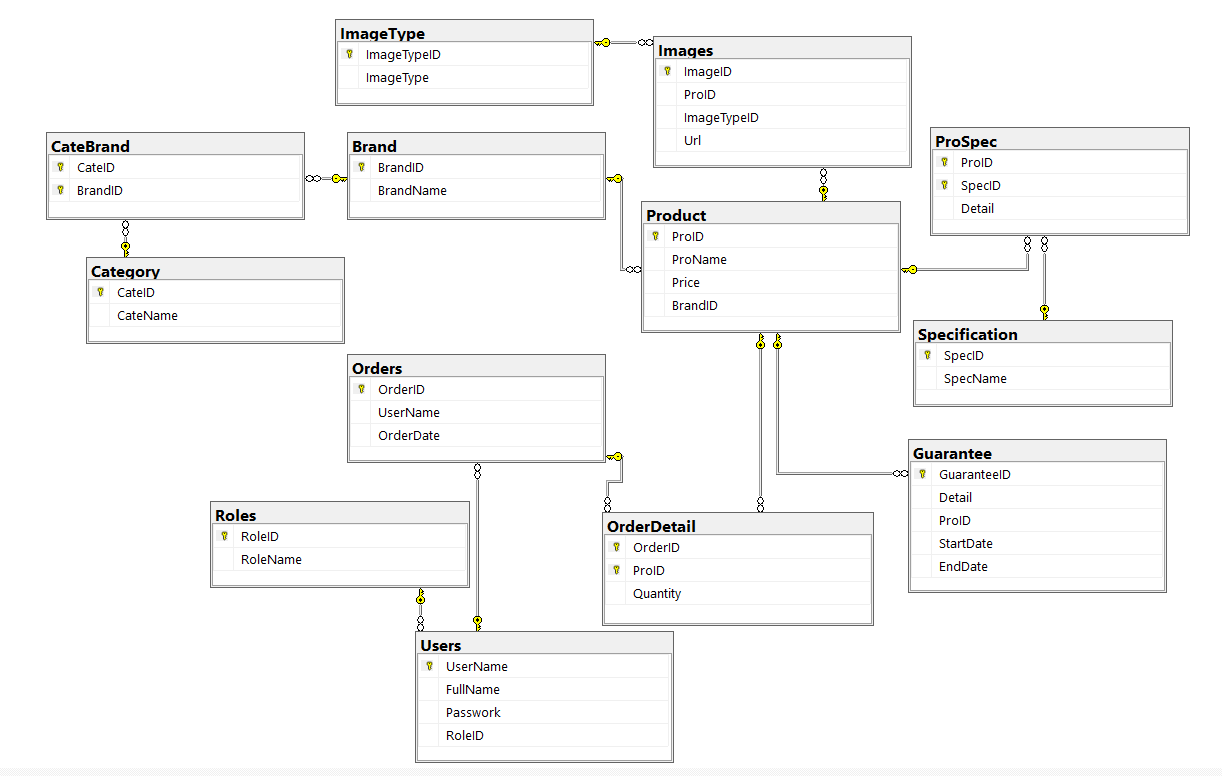


***Package descriptions & package class naming conventions***

|  |  |  |
| --- | --- | --- |
| **No** | **Package** | **Description** |
| *01* | *Bean* | *JavaBeans are classes that encapsulate many objects into a single object (the bean)* |
| *02* | *View* | *web app user interface* |
| *03* | *Controller* | *where to navigate requests, responses* |
| *04* | *Dao* | *Contains methods for handling the object's data* |
| *05* | *Util* | *Where to connect to the database* |

## 2. Database Schema

*[Provide the tables relationship like example below – following MySQL database naming convention]*



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

|  |  |  |
| --- | --- | --- |
| **No** | **Table** | **Description** |
| *01* | *Order Details* | Table *Order* Details contains a list of items that the user has bought. With : - Primary keys: OrderID, ProID  - Forgin key: OrderID, ProID |
| *02* | *Brand* | Table Brand contains the names of the brands of the products.  With : - Primary keys: BrandID |
| *03* | *Product* | Table Product contains product description information  With : - Primary keys: ProID  - Forgin key: BrandID |
| *04* | *Images* | *Table Images contain pictures of products.*  With : - Primary keys: ImageID  - Forgin key: ProID, ImageTypeID |
| *05* | *Category* | Table Category contains the main items the store sells  With : - Primary keys: CateID |
| *06* | *Prospec* | Table Prospec contains configuration details of products  With : - Primary keys: ProID, SpecID  - Forgin key: ProID, SpecID |
| *07* | *ImageType* | Table is used to specify which images are main and secondary images. With : -Primary key: ImageTypeID |
| *08* | *CateBrand* | Table CateBrand is used to show brand correspond to product  With : - Primary keys: CateID, BrandID  - Forgin key: CateID, BrandID |
| *09* | *Order* | Table Bill used to contain User purchase information  With : - Primary keys: OrderID  - Forgin key: UserName |
| *10* | *Specification* | Table Specification contains configuration type names of products  With : - Primary keys: SpecID |
| *11* | *Guarantee* | Table Guarantee contains warranty information of products  With : - Primary keys: GuaranteeID  - Forgin key: ProID |
| *12* | *User* | Table User contains user information and account  With : - Primary keys: UserName  - Forgin key: RoleID |
| *13* | *Role* | Table Role used to authorize web app users  With : - Primary keys: RoleID |

# II. Code Designs

## 1. < Authentication >

*[Provide the detailed design for the function <Feature/Function Name1>. It includes Class Diagram, Class Specifications, and Sequence Diagram(s)]*

### a. Class Diagram

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



### b. Class Specifications

#### Access Controller

*In this class, controller will receive requests from Access Forms, then send back the response to views*

#### Access Management

|  |  |  |
| --- | --- | --- |
| **No** | **Method** | **Description** |
| *01* | *check* | *Used to authenticate user login into our system by comparison username + password parameters with username + password inside database.* |
|  |  |  |

### c. Sequence Diagram(s)

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



### d. Database queries

*1. select UserName, Password from Users where UserName=? and password=? ;*

## 2. <Load product to views>

### b. Class Specifications

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

#### Product Controller

*In this class, controller will receive requests from Product Page views, then send back the response to views.*

#### Detail Controller

*In this class , controlle will receive the parameter proID sent from Product page, then it put this param in the method which has a mission that finds the product has corresponding .*

#### Product Management

*This class contains all of methods which relates to process data of product.*

|  |  |  |
| --- | --- | --- |
| **No** | **Method** | **Description** |
| *01* | *GetAllOfProduct* | *This method is used to retrieve datas which involves to product from database. After processing data, it just got, method will return an array List of Product objects.* |
| *02* | *FindKeyExistInArray* | *Used to check the image of product existed or not by passing an id parameter. If inside the ArrayList of images already contains an image with id passed -> method return true, and vice versa ….* |
| *03* | *selectTop8InHomepage* | *Used to choose 8 products from database, then put it in featured product at homepage* |
| *04* | *getProductById* | *Used to find the the product has corresponding id with id parameter, then return that product.* |

### d. Database queries

*1. SELECT P.ProID, P.ProName,P.OriginalPrice, P.Discount, P.CurrentPrice, I.ImageID, I.Url, B.BrandName,C.CateName, S.SpecName, PS.Detail"*

*FROM dbo.Product P , dbo.Images I , dbo.Brand B , dbo.Category C ,dbo.Specification S,dbo.ProSpec PS*

*WHERE P.BrandID = B.BrandID AND P.CateID = C.CateID AND I.ProID = P.ProID AND S.SpecID = PS.SpecID AND PS.ProID = P.ProID (used in getAllProduct() method)*

*2. SELECT top 8 b.BrandName, p.ProID, p.ProName, p.OriginalPrice, i.ImageID, i.Url \n"*

*+ "FROM Category c, Brand b, Product p, Images i \n"*

*+ " WHERE c.CateID = p.CateID AND b.BrandID = p.BrandID AND i.ProID = p.ProID AND i.ImageTypeID = 1 AND c.CateName LIKE '%Laptop%'*

# III. Database Tables

## 1. <Category>

*This table is used to categorize products*

*[Table fields, in the form of table format as below]*

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **#** | **Field name** | **Type** | **Size** | **Unique** | **Not Null** | **PK/FK** | **Notes** |
| 1 | CateID | Int(identity(1,1)) | 4 bytes |  | yes | PK | CateID describes the code of product type |
| 2 | CateName | nvarchar | Length: 50 |  | Yes | none | CateID describes the name of product type |

## 2. <CateBrand>

*This table contains the details between Category table and Brand table*

*[Table fields, in the form of table format as below]*

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **#** | **Field name** | **Type** | **Size** | **Unique** | **Not Null** | **PK/FK** | **Notes** |
| 1 | CateID | Int | 4 bytes |  | yes | PK, FK | Identity for category of product |
| 2 | BrandID | Int | 4 bytes |  | yes | PK, FK | Identity for brand of product |

## 3. <Brand>

*This table contains the brand of product*

*[Table fields, in the form of table format as below]*

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **#** | **Field name** | **Type** | **Size** | **Unique** | **Not Null** | **PK/FK** | **Notes** |
| 1 | BrandID | Int(identity(1,1)) | 4 bytes |  | yes | PK | BrandID describes the code of Brand |
| 2 | BrandName | nvarchar | Length: 50 |  | yes | none | BrandName describes the name of Brand |

## 4. <Guarantee>

*This table shows the warranty policies for customers when buying products*

*[Table fields, in the form of table format as below]*

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **#** | **Field name** | **Type** | **Size** | **Unique** | **Not Null** | **PK/FK** | **Notes** |
| 1 | GuaranteeID | Int(identity(1,1)) | 4 bytes |  | yes | PK | CateID describes the product code |
| 2 | Detail | nvarchar | Length: 100 |  | yes | none | Describes information of guarantee |
| 3 | ProID | int | 4 bytes |  | yes | FK | Describes the the code product has guarantee |
| 4 | StartDate | date | 3 bytes |  | yes | none | Describes the guarantee start date |
| 5 | EndDate | date | 3 bytes |  | yes | none | Describes the guarantee end date |

## 5. <ProSpec>

*This table contains the details between Product table and Specification table*

*[Table fields, in the form of table format as below]*

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **#** | **Field name** | **Type** | **Size** | **Unique** | **Not Null** | **PK/FK** | **Notes** |
| 1 | ProID | Int | 4 bytes |  | yes | PK, FK | Identity of product |
| 2 | SpecID | Int | 4 bytes |  | yes | PK, FK | Identity of specification of product |
| 3 | Detail | Nvarchar | 100 |  | no | none | Describe features of product |

## 6. <Product>

*This table contains products*

*[Table fields, in the form of table format as below]*

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **#** | **Field name** | **Type** | **Size** | **Unique** | **Not Null** | **PK/FK** | **Notes** |
| 1 | ProID | Int(identity(1,1)) | 4 bytes |  | yes | PK | Identity code of product |
| 2 | ProName | nvarchar | 100 |  | yes | none | Name of the product |
| 3 | OriginalPrice | int | 4 bytes |  | yes | none | The original price when haven’t applied discount yet |
| 4 | Discount | int | 4 bytes |  | no | none | The price loss that customer was get |
| 5 | CurrentPrice | int | 4 bytes |  | no | none | The current price that applied discount (if possible) |
| 6 | BrandID | int | 4 bytes |  | yes | FK | Identity for brand of product |
| 7 | CateID | int | 4 bytes |  | no | FK | Identity for category of product |

## 7 <Images>

*This table contains images that are used to when shows product*

*[Table fields, in the form of table format as below]*

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **#** | **Field name** | **Type** | **Size** | **Unique** | **Not Null** | **PK/FK** | **Notes** |
| 1 | ImageID | Int(identity(1,1)) | 4 bytes |  | yes | PK | Describes the image code |
| 2 | ProID | Int | 4 bytes |  | yes | FK | Describes the product receive images throw ProID |
|  | ImageTypeID | Int | 4 bytes |  | yes | FK | Descripe the Image type |
|  | Url | text | 16 bytes |  | yes | none | Link of image |

## 8. <Specification>

*This table contains specifications of product*

*[Table fields, in the form of table format as below]*

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **#** | **Field name** | **Type** | **Size** | **Unique** | **Not Null** | **PK/FK** | **Notes** |
| 1 | SpecID | Int(identity(1,1)) | 4 bytes |  | yes | PK | Identity for specifications of product |
| 2 | SpecName | nvarchar | 50 |  | yes | none | Name of specification want to mention |

## 9. <Roles>

*This table contains roles of users*

*[Table fields, in the form of table format as below]*

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **#** | **Field name** | **Type** | **Size** | **Unique** | **Not Null** | **PK/FK** | **Notes** |
| 1 | RoleID | Int(identity(1,1)) | 4 bytes |  | yes | PK | RoleID describes the role code of User |
| 2 | RoleName | nvarchar | Length: 50 |  | yes | none | Describes the name of role |

## 10. <Users>

*This table contains account and user's profile.*

*[Table fields, in the form of table format as below]*

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **#** | **Field name** | **Type** | **Size** | **Unique** | **Not Null** | **PK/FK** | **Notes** |
| 1 | UserName | varchar | 100 |  | yes | PK | Username of users to login Web |
| 2 | FullName | nvarchar | 50 |  | no | none | Describes the name of users |
| 3 | Password | varchar | 100 |  | yes | none | Passwork to login Web |
| 4 | PhoneNumber | char | 9 |  | no | none | Describes the phone number of users |
| 5 | RoleID |  | 4 bytes |  | no | FK | Describe the role user |

## 11. <Orders>

*This table describes customer information and product purchase date*

*[Table fields, in the form of table format as below]*

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **#** | **Field name** | **Type** | **Size** | **Unique** | **Not Null** | **PK/FK** | **Notes** |
| 1 | OrderID | Int(identity(1,1)) | 4 bytes |  | yes | PK | Indentity for order of user |
| 2 | UserName | varchar | 100 |  | no | FK | Name of user |
| 3 | OrderDate | date | 3 bytes |  | yes | none | The date when user order |

## 12. <ImageType>

*This table is used to distinguish the main image and sub image*

*[Table fields, in the form of table format as below]*

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **#** | **Field name** | **Type** | **Size** | **Unique** | **Not Null** | **PK/FK** | **Notes** |
| 1 | ImageTypeID | Int(identity(1,1)) | 4 bytes |  | yes | PK | Describes the Image type code |
| 2 | ImageType | nvarchar | 50 |  | yes | none | Describes the name of image type |

## 13. <OrderDetail>

*This table contains product code and product quantity requested by customer*

*[Table fields, in the form of table format as below]*

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **#** | **Field name** | **Type** | **Size** | **Unique** | **Not Null** | **PK/FK** | **Notes** |
| 1 | OrderID | Int | 4 bytes |  | yes | PK | Identity for order of user |
| 2 | ProID | int | 4 bytes |  | yes | FK | Identity of product |
| 3 | Quantity | int | 4 bytes |  | yes | none | Describes the quantity of product in Oder |