

**<<Online Shop>>**

**Software Design Specification**

– Hanoi, August 2022 –

Record of changeS

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

\*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)

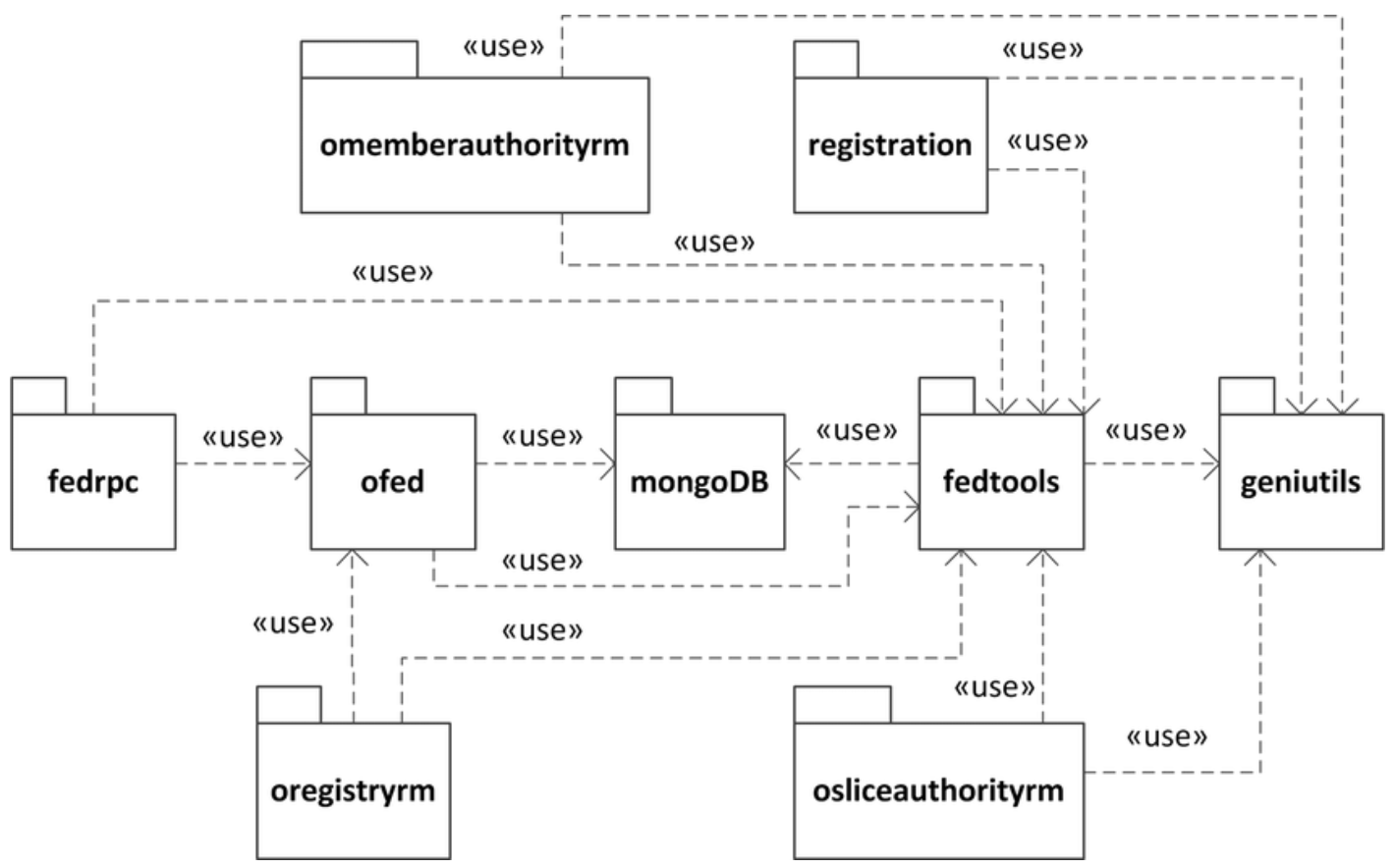
[d. Database queries 6](#_Toc96516296)

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

# I. Overview

## 1. Code Packages

*[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 & description table format below (****please note: package names don’t follow Java package naming convention yet****)]*



***Package descriptions***

|  |  |  |
| --- | --- | --- |
| **No** | **Package** | **Description** |
| *01* | *omemberauthorityrm* | *<Description of the package>* |
| *02* | *registration* | *<Description of the package>* |
| *03* | *fedrpc* | *<Description of the package>* |
| *04* | *ofed* | *<Description of the package>* |
| *05* | *mongoDB* | *<Description of the package>* |
| *06* | *fedtools* | *<Description of the package>* |
| *07* | *…* |  |

## 2. Database Design

### a. Database Schema

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



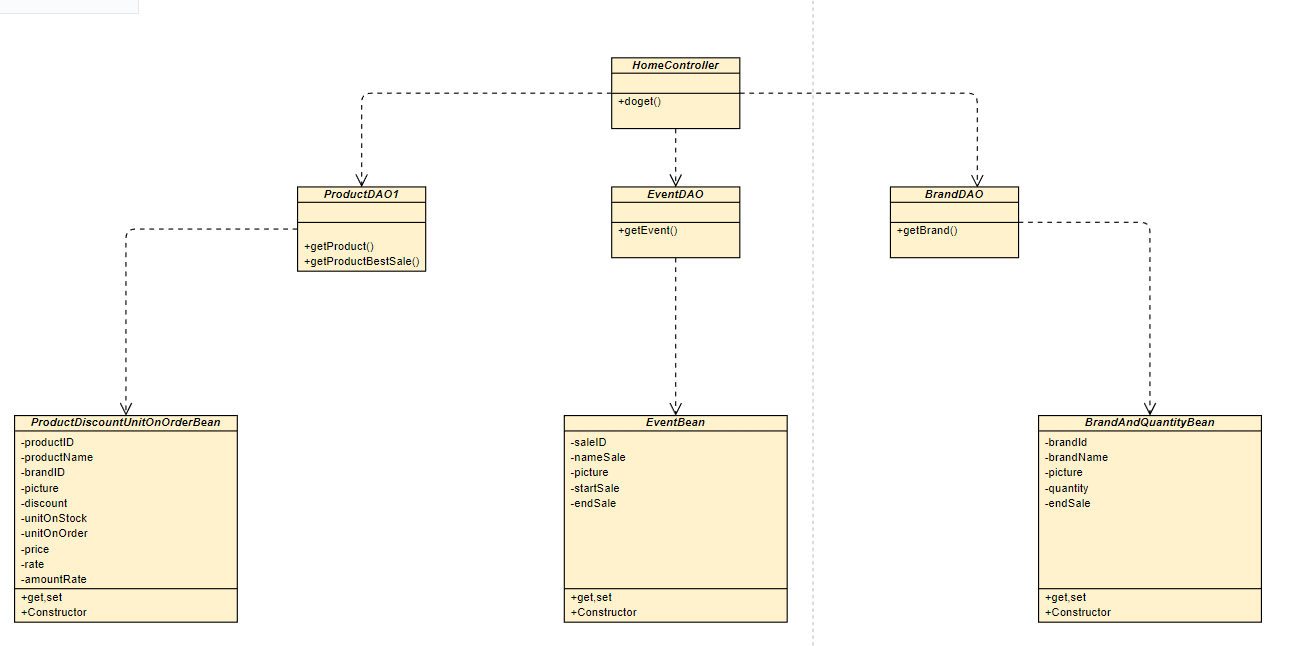
### b. Table Description

|  |  |  |
| --- | --- | --- |
| **No** | **Table** | **Description** |
| *01* | *<Table name>* | *<Description of the table>* |

# II. Code Designs

## 1. <Home>

### a. Class Diagram



### b. Class Specifications

*ProductDAO1*

|  |  |  |
| --- | --- | --- |
| **No** | **Method** | **Description** |
| *01* | *getProduct()* | *Get list of products sorted by input status.*  *Input int 1: Sorted by Discount.*  *Input int 2: Sorted by UnitOnOrder.*  *Input int 3: Sorted by UnitOnStock* |
|  |  |  |

***EventDAO***

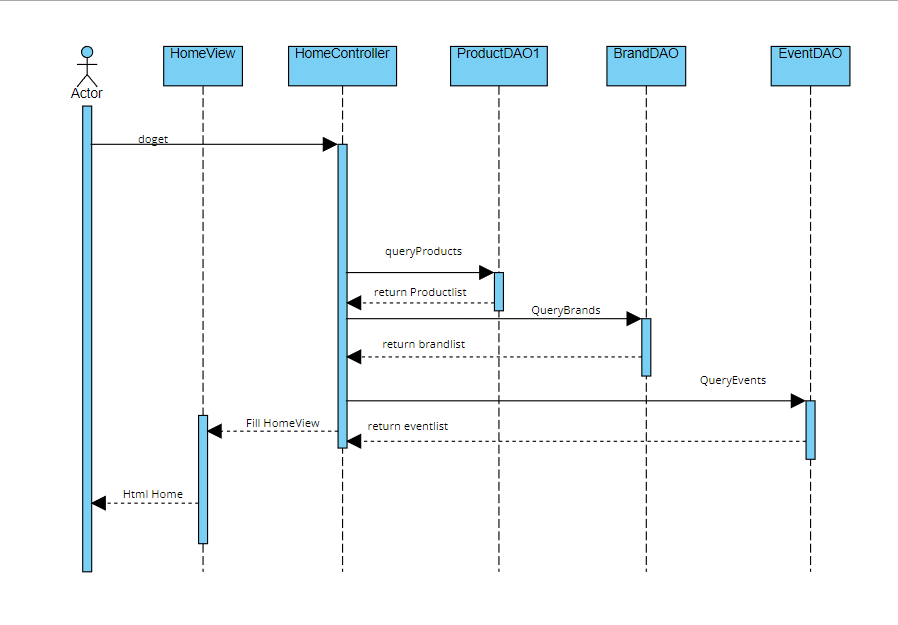
|  |  |  |
| --- | --- | --- |
| **No** | **Method** | **Description** |
| *01* | *getEvent();* | *Get list of event have exist this time.* |
|  |  |  |

***BrandDAO***

|  |  |  |
| --- | --- | --- |
| **No** | **Method** | **Description** |
| *01* | *getBrand();* | *Get list of brand and quantity product of this brand.* |
|  |  |  |

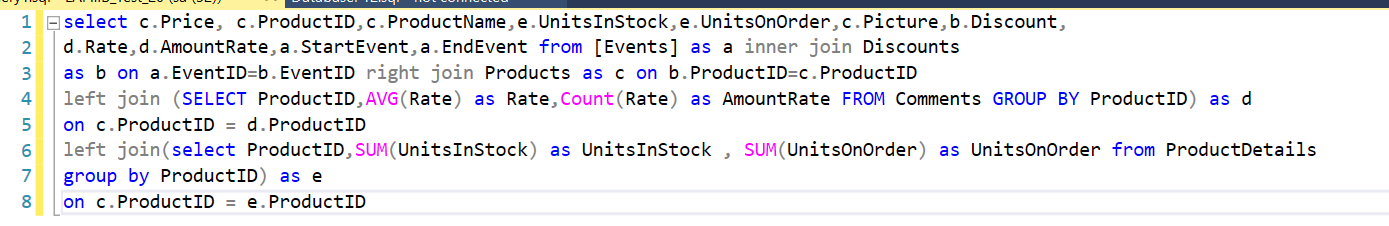
### c. Sequence Diagram(s)

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



### d. Database Queries

**SQL method getProduct():**



inner join with two table Event and Discount on ProductID

right join with Product to take all Product.

Left join with query(Comment table Group by ProductID to take AVG(Rate), Count(Rate))

Left join with query(ProductDetailts Group by ProductID to take Sum(UnitInStock) and Sum(UnitsOnOrder))

**SQL method getEvent():**

select \* from [Events]:

Take all event form table event.

**SQL method getBrand():**

Select \* from Brands inner join (select BrandID,Count(ProductID) as Quantity from Products group by BrandID)as b ON Brands.BrandID=b.BrandID

Brands inner join with query( Products table group by BrandID to take Quantity of Product) on BrandID

## 2. <Feature/Function Name2>

…