A picture containing text

Description automatically generatedA picture containing graphical user interface

Description automatically generated

**GAMES SHOP**

**Software Design Specification**

– Hanoi, April 2022 –

Record of changeS

|  |  |  |  |
| --- | --- | --- | --- |
| Date | A\* M, D | In charge | Change Description |
| 14/4/2022 | A | Trần Sơn | Database ERD diagram |
| 18/4/2022 | M | Trần Sơn | Update ERD diagram |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

\*A - Added M - Modified D - Deleted

Table of Contents

[I. Overview 4](file:///C:\Users\trson\Downloads\SWP391-G3-AppDesign.docx#_Toc83365441)

[1. Code Packages 4](file:///C:\Users\trson\Downloads\SWP391-G3-AppDesign.docx#_Toc83365442)

[2. Database Design 4](file:///C:\Users\trson\Downloads\SWP391-G3-AppDesign.docx#_Toc83365443)

[a. Database Schema 4](file:///C:\Users\trson\Downloads\SWP391-G3-AppDesign.docx#_Toc83365444)

[b. Table Description 4](file:///C:\Users\trson\Downloads\SWP391-G3-AppDesign.docx#_Toc83365445)

[3. Coding Conventions 5](file:///C:\Users\trson\Downloads\SWP391-G3-AppDesign.docx#_Toc83365446)

[II. Code Designs 5](file:///C:\Users\trson\Downloads\SWP391-G3-AppDesign.docx#_Toc83365447)

[1. <Feature/Function Name1> 5](file:///C:\Users\trson\Downloads\SWP391-G3-AppDesign.docx#_Toc83365448)

[a. Class Diagram 5](file:///C:\Users\trson\Downloads\SWP391-G3-AppDesign.docx#_Toc83365449)

[b. Class Specifications 5](file:///C:\Users\trson\Downloads\SWP391-G3-AppDesign.docx#_Toc83365450)

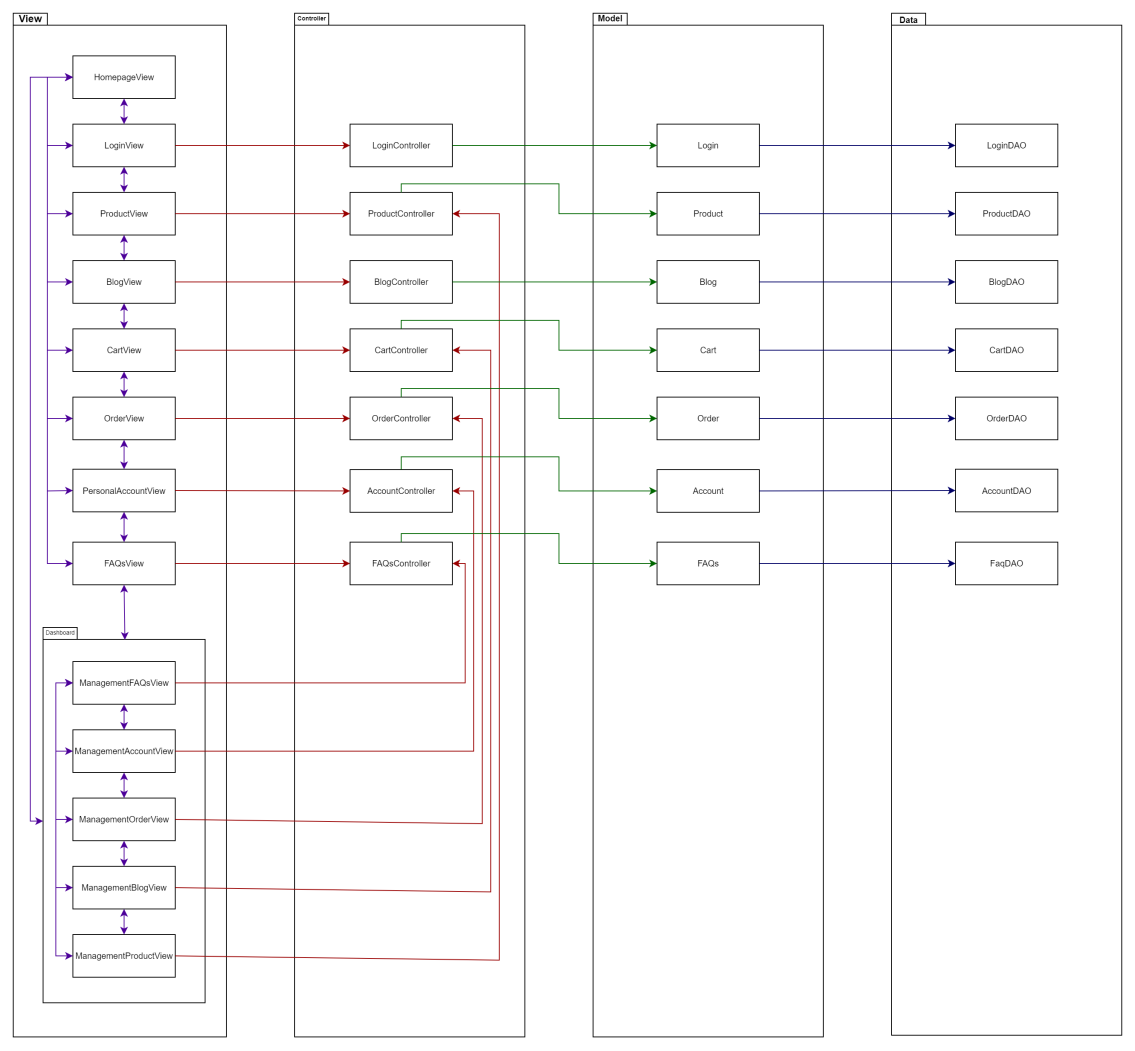
[c. Sequence Diagram(s) 6](file:///C:\Users\trson\Downloads\SWP391-G3-AppDesign.docx#_Toc83365451)

[d. Database queries 6](file:///C:\Users\trson\Downloads\SWP391-G3-AppDesign.docx#_Toc83365452)

[2. <Feature/Function Name2> 6](file:///C:\Users\trson\Downloads\SWP391-G3-AppDesign.docx#_Toc83365453)

# I. Overview

## 1. Code Packages

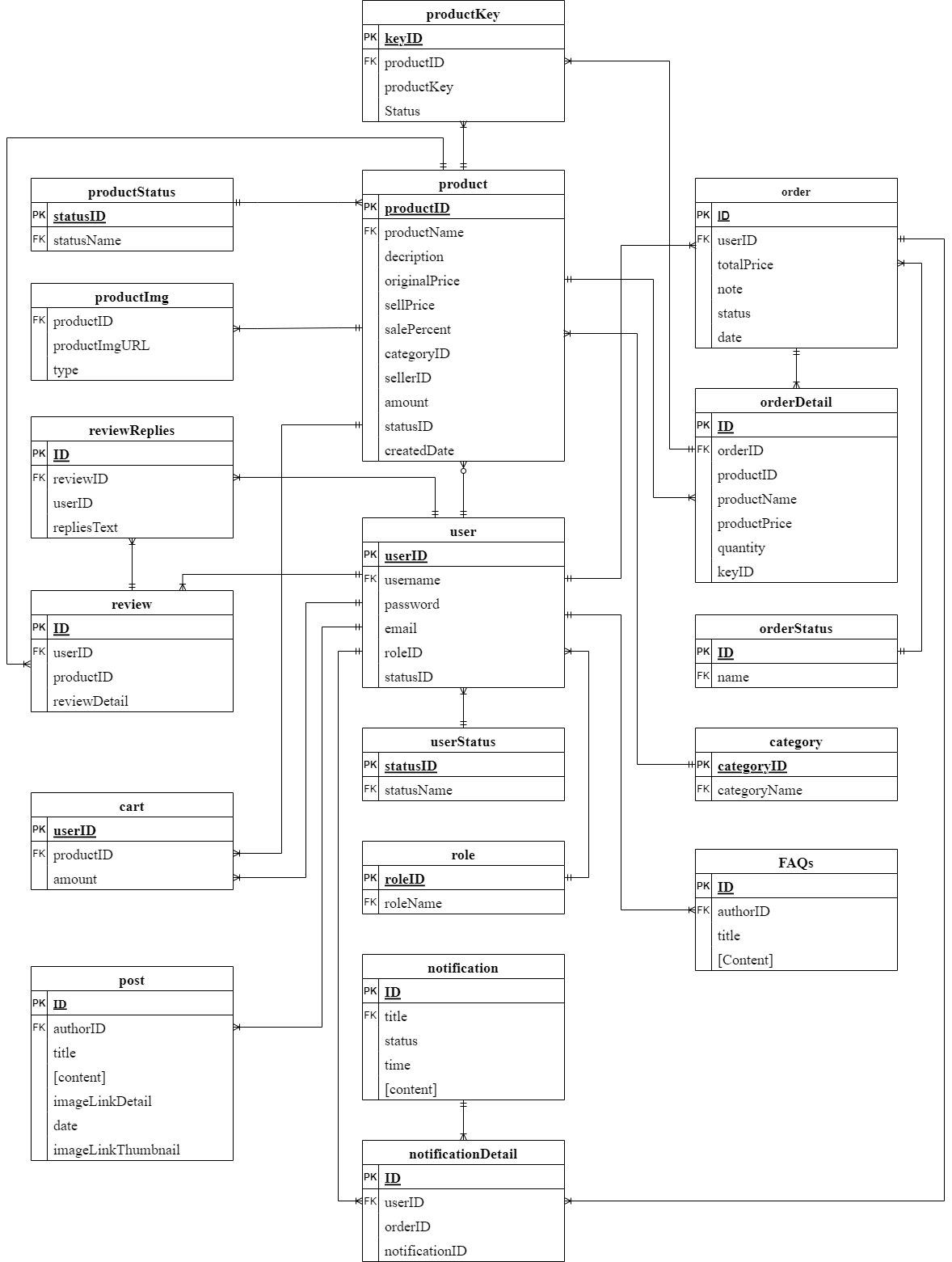


***Package descriptions & package class naming conventions***

|  |  |  |
| --- | --- | --- |
| **No** | **Package** | **Description** |
| *01* | *view* | *This package used to contain html/css/js file, besides is assets with image, logo,…*  *\*.jsp, \*.css, \*.js, …* |
| *02* | *controller* | *This package used to contain controller file.*  *<Object>Controller.java* |
| *03* | *model* | *This package used to contain model file.*  *<Object>.java* |
| *04* | *data* | *This package used to contain DAO file, connect between database with model.*  *<Object>DAO.java* |

## 2. Database Design

### a. Database Schema



### b. Table Description

|  |  |  |
| --- | --- | --- |
| **No** | **Table** | **Description** |
| 01 | FQAs | Store all common question and answer |
| 02 | products | Store product information |
| 03 | productStatus | Store product’s status information |
| 04 | productImg | Store product image link |
| 05 | reviewReplies | Store staff reply to customer |
| 06 | review | Store user review and information |
| 07 | orders | Store orders information |
| 08 | orderDetail | Store order detailed information |
| 09 | orderStatus | Store order current status |
| 10 | category | Store category information |
| 11 | users | Store user informatioms |
| 12 | userStatus | Store user account status |
| 13 | role | Storea roles in system |
| 14 | notifications | Store user notifications |
| 15 | Cart | Store user cart |
| 16 | Blog | Store written blog |
| 17 | notificationsDetail | Detail of notification |
| 18 | productKey | Store key of product |

## 1. Login

### a. Class Diagram

### b. Class Specifications

#### LoginController Class

|  |  |  |
| --- | --- | --- |
| **No** | **Method** | **Description** |
| *01* | *CheckInput* | *Check user input error* |

#### LoginDAO Class

***Class Methods***

|  |  |  |
| --- | --- | --- |
| **No** | **Method** | **Description** |
| *01* | *processRequest* | *Get username and password than pass to LoginDAO than query in database to get User Entity to save entity to web* |

### c. Sequence Diagram(s)

### d. Database queries

## 2. User Management

### a. Class Diagram

### b. Class Specifications

#### AccountDeleteController

***Class Methods***

|  |  |  |
| --- | --- | --- |
| **No** | **Method** | **Description** |
| *01* | *doGet* | *Get user information from web interface than get user ID. After that, the ID will be passed to the AccountDAO and will be deleted in the Database* |

#### AccountEditController

***Class Methods***

|  |  |  |
| --- | --- | --- |
| **No** | **Method** | **Description** |
| *01* | *doGet* | *After an account is selected from Account list, the account information is passed to the editAccount.jsp and be displayed.* |
| *02* | *doPost* | *After account’s information is changed in Web interface. Create an user entiy than pass to AccountDAO to Update in Database.* |

#### AccountListController

***Class Methods***

|  |  |  |
| --- | --- | --- |
| **No** | **Method** | **Description** |
| *01* | *processRequest* | *Get user list from database and pass to adminAccountManagemen and be displayed on web browser.* |

#### AccountAddController

***Class Methods***

|  |  |  |
| --- | --- | --- |
| **No** | **Method** | **Description** |
| *01* | *doPost* | *Get user information from web browers, create an object of user and insert into database* |

### c. Sequence Diagram(s)

### d. Database queries

## 4. Blog Management

### a. Class Diagram

### b. Class Specifications

#### BlogDeleteController

***Class Methods***

|  |  |  |
| --- | --- | --- |
| **No** | **Method** | **Description** |
| *01* | *doGet* | *Get blog information from web interface than get blog ID. After that, the ID will be passed to the BlogDAO and will be deleted in the Database* |

#### BlogEditController

***Class Methods***

|  |  |  |
| --- | --- | --- |
| **No** | **Method** | **Description** |
| *01* | *doGet* | *After an Blog is selected from Blog list, the account information is passed to the editBlog.jsp and be displayed.* |
| *02* | *doPost* | *After blog’s information is changed in Web interface. Create an blog entiy than pass to BlogDAO to Update in Database.* |

#### BlogListController

***Class Methods***

|  |  |  |
| --- | --- | --- |
| **No** | **Method** | **Description** |
| *01* | *processRequest* | *Get blog list from database and pass to blogManagement and be displayed on web browser.* |

#### BlogAddController

***Class Methods***

|  |  |  |
| --- | --- | --- |
| **No** | **Method** | **Description** |
| *01* | *doPost* | *Get blog information from web browers, create an object of blog and insert into database* |

### c. Sequence Diagram(s)

### d. Database queries

## 5. Logout

### a. Class Diagram

### b. Class Specifications

#### LogoutController Class

|  |  |  |
| --- | --- | --- |
| **No** | **Method** | **Description** |
| *01* | *Logout* | *Delete user from browser than redirect to homepage* |

### c. Sequence Diagram(s)

### d. Database queries

## 6. Product Management

### a. Class Diagram

### b. Class Specifications

#### ProductDeleteController

***Class Methods***

|  |  |  |
| --- | --- | --- |
| **No** | **Method** | **Description** |
| *01* | *doGet* | *Get product information from web interface than get productID. After that, the ID will be passed to the ProductDAO and will be deleted in the Database* |

#### ProductEditController

***Class Methods***

|  |  |  |
| --- | --- | --- |
| **No** | **Method** | **Description** |
| *01* | *doGet* | *After an Product is selected from Product list, the account information is passed to the ProductEdit.jsp and be displayed.* |
| *02* | *doPost* | *After Product’s information is changed in Web interface. Create an Product entiy than pass to ProductDAO to Update in Database.* |

#### ProductListController

***Class Methods***

|  |  |  |
| --- | --- | --- |
| **No** | **Method** | **Description** |
| *01* | *processRequest* | *Get product list from database and pass to productManagement and be displayed on web browser.* |

#### ProductAddController

***Class Methods***

|  |  |  |
| --- | --- | --- |
| **No** | **Method** | **Description** |
| *01* | *doPost* | *Get product information from web browers, create an object of Product and insert into database* |

### c. Sequence Diagram(s)

### d. Database queries

## 7. Order Management

### a. Class Diagram

### b. Class Specifications

#### OrderDeleteController

***Class Methods***

|  |  |  |
| --- | --- | --- |
| **No** | **Method** | **Description** |
| *01* | *doGet* | *Get order information from web interface than get orderID.. After that, the ID will be passed to the OrderDAO and will be deleted in the Database* |

#### OrderEditController

***Class Methods***

|  |  |  |
| --- | --- | --- |
| **No** | **Method** | **Description** |
| *01* | *doGet* | *After an Order is selected from Order list, the account information is passed to the OrderEdit.jsp and be displayed.* |
| *02* | *doPost* | *After Order’s information is changed in Web interface. Create an Order entiy than pass to ProductDAO to Update in Database.* |

#### OrdertListController

***Class Methods***

|  |  |  |
| --- | --- | --- |
| **No** | **Method** | **Description** |
| *01* | *processRequest* | *Get order list from database and pass to orderManagement and be displayed on web browser.* |

#### OrderAddController

***Class Methods***

|  |  |  |
| --- | --- | --- |
| **No** | **Method** | **Description** |
| *01* | *doPost* | *Get order information from web browers, create an object of Order and insert into database* |

### c. Sequence Diagram(s)

### d. Database queries