

**SOCICAL COMMERCIAL**

**Software Design Document**

– Hanoi, May 2021 –

**Table of Contents**

[I. Overview 3](#_heading=h.gjdgxs)

[1. Code Packages 3](#_heading=h.30j0zll)

[2. Database Schema 3](#_heading=h.1fob9te)

[II. Code Designs 4](#_heading=h.3znysh7)

[1. <Feature/Function Name1> 4](#_heading=h.2et92p0)

[a. Class Diagram 4](#_heading=h.tyjcwt)

[b. Class Specifications 4](#_heading=h.3dy6vkm)

[c. Sequence Diagram(s) 4](#_heading=h.1t3h5sf)

[d. Database queries 5](#_heading=h.4d34og8)

[2. <Feature/Function Name2> 5](#_heading=h.2s8eyo1)

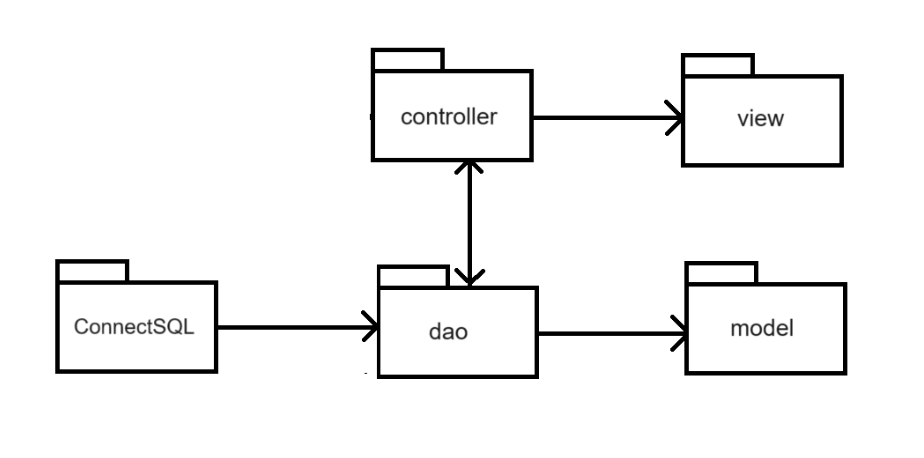
[III. Database Tables 5](#_heading=h.17dp8vu)

[1. <Table name 1> 5](#_heading=h.3rdcrjn)

[2. <Table name 2…> 5](#_heading=h.26in1rg)

# I. Overview

## 1. Code Packages/Namespaces

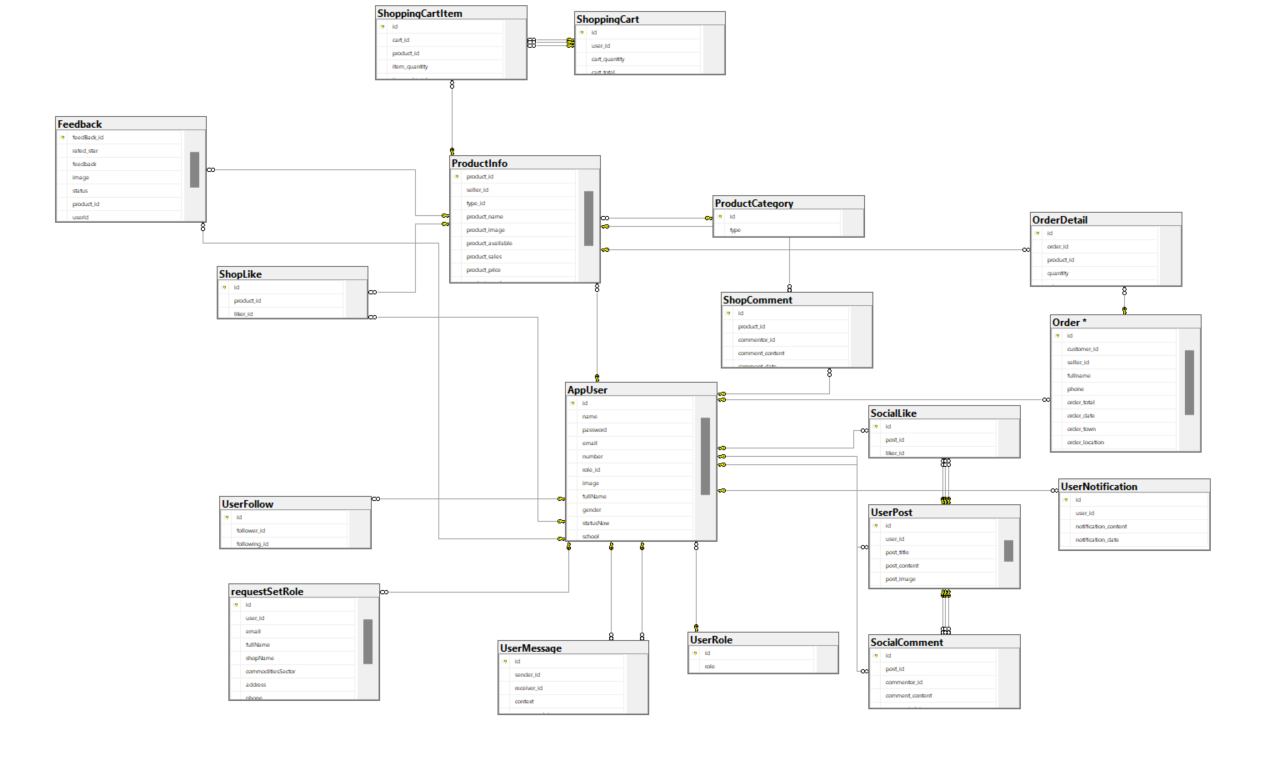


***Package descriptions & package class naming conventions***

|  |  |  |
| --- | --- | --- |
| **No** | **Package** | **Description** |
| *01* | *dao* | *This component is used to get data directly FROM the database* |
| *02* | *connectSQL* | *Contains code related to database connectivity and management.* |
| *03* | *model* | *This component includes interfaces used to receive data retrieved FROM the database and classes used to receive data* |
| *04* | *controller* | *Acts as a go-between for the service layer and the Front-end, passing on different input requests* |
| *05* | *view* | *This component display front-end* |

## 2. Database Design

### a. Database Schema

**

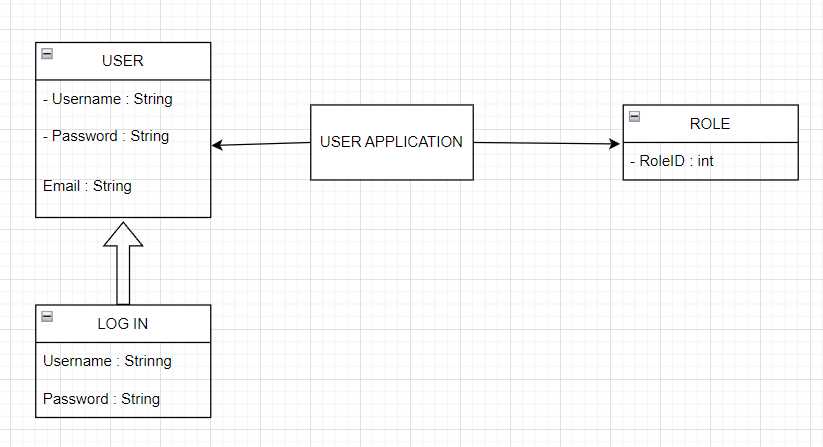
### b. Table Description

|  |  |  |
| --- | --- | --- |
| **No** | **Table** | **Description** |
| 1 | AppUser | Entity representing the user of the application or system. |
| 2 | UserNotification | Entity representing a notification sent to or received from a user. |
| 3 | SocialComment | Entity representing comments in a social networking environment or similar application. |
| 4 | UserPost | Entity representing posts or content posted by users. |
| 5 | SocialLike | Entity representing likes or favorites on social media posts or content. |
| 6 | UserFollow | Entity representing following or following between users. |
| 7 | UserRole | Entity representing the user's role in the system or application. |
| 8 | ShopLike | Entity representing a like or favorite on a store or product. |
| 9 | UserMessage | Entity representing messages between users. |
| 10 | CustomerOrder | Entity representing an order from a customer. |
| 11 | OrderStatus | Entity representing the status of an order, such as processing, shipped, completed, etc. |
| 12 | OrderItemInfo | Entity representing detailed information about an item in an order, such as product, quantity, price, etc. |
| 13 | ProductCategory | Entity representing product category, to classify products into different groups. |
| 14 | ProductInfo | Entity representing detailed information about a product, such as name, description, price, etc. |
| 15 | ShopComment | Entity representing comments about a store or product. |
| 16 | ShoppingCart | Entity representing a shopping cart, where users can add products for later purchase. |
| 17 | ShoppingCartItem | Entity representing an item in a shopping cart, typically related to product and quantity. |

# II. Code Designs

## 1. Login

### a. Class Diagram



### b. Class Specifications

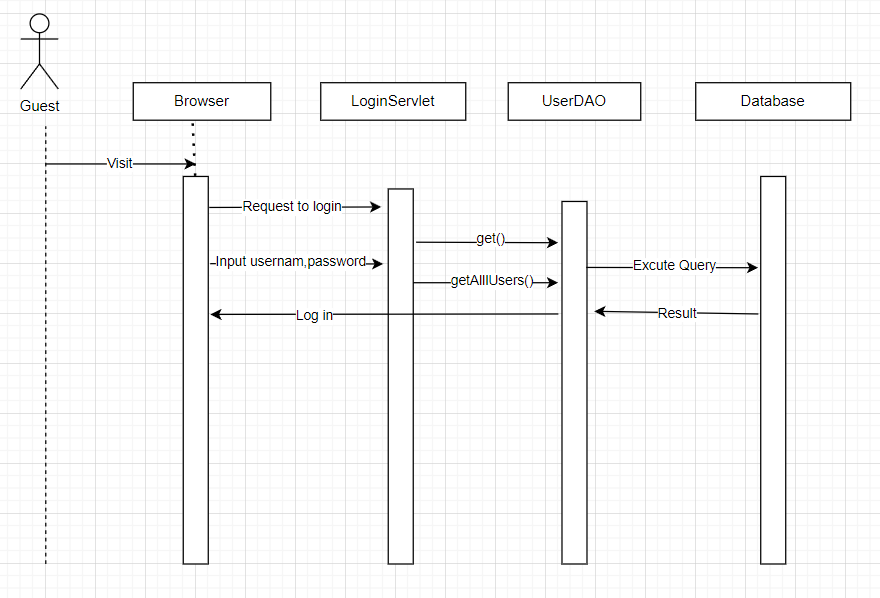
* LoginServlet Class

|  |  |  |
| --- | --- | --- |
| **No** | **Method** | **Description** |
| *01* | *doGet()* | *Used to handle HTTP GET requests, typically for retrieving data from the server.* |
| *02* | *doPost()* | *Used to handle HTTP POST requests, usually for sending data to the server.* |

* UserDAO Class

|  |  |  |
| --- | --- | --- |
| **No** | **Method** | **Description** |
| *01* | *get()* | *get username and password* |
| *02* | *getAllUsers()* | *get user information include id, name, roleID, email, image* |

### c. Sequence Diagram(s)



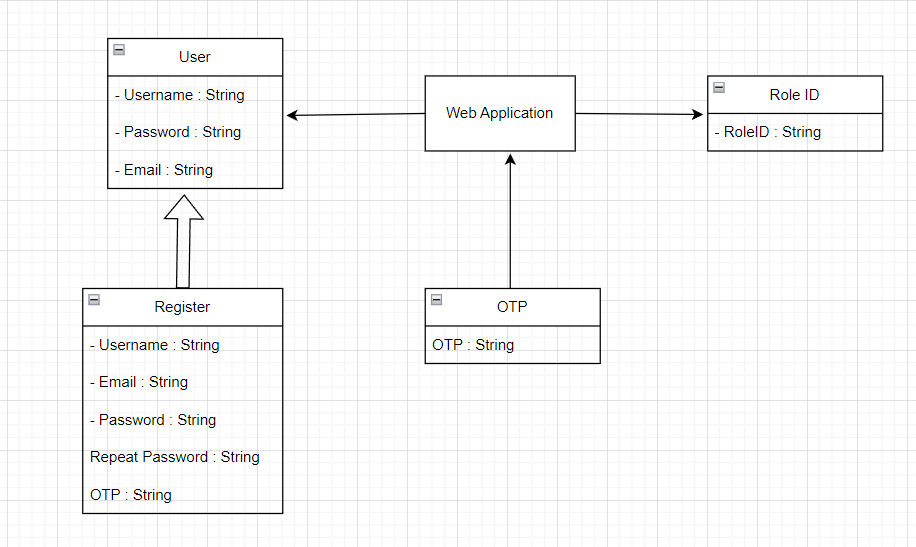
### d. Database queries

SELECT \* FROM [SWP391].[dbo].[AppUser]

SELECT \* FROM [SWP391].[dbo].[AppUser] WHERE name=? AND password=?

## 2. Register

### a. Class Diagram



### b. Class Specifications

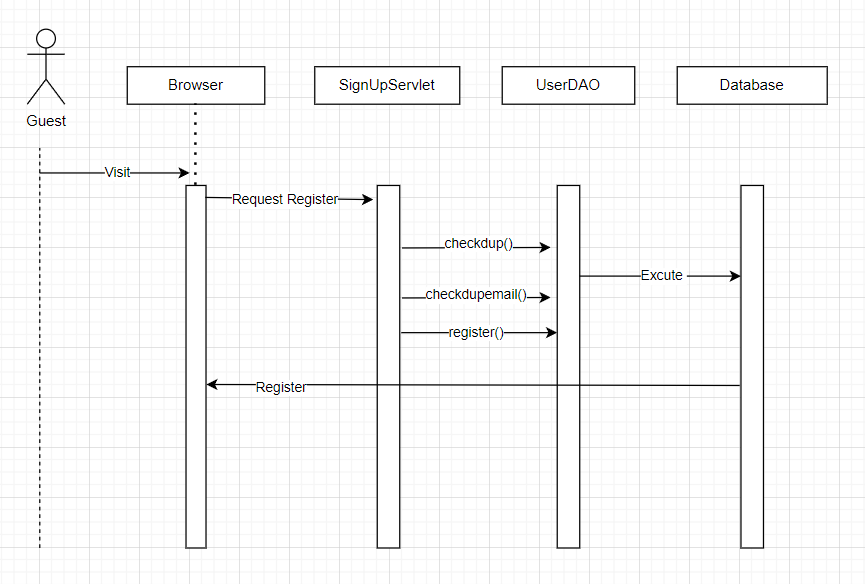
* SignUpServlet Class

|  |  |  |
| --- | --- | --- |
| **No** | **Method** | **Description** |
| *01* | *doGet()* | *Used to handle HTTP GET requests, typically for retrieving data from the server.* |
| *02* | *doPost()* | *Used to handle HTTP POST requests, usually for sending data to the server.* |

* UserDAO Class

|  |  |  |
| --- | --- | --- |
| **No** | **Method** | **Description** |
| *01* | *checkdup()* | *Check username was use or not* |
| *02* | *checkdupemail()* | *Check email was use or not* |
| *03* | *register()* | *Add new user to database* |

### c. Sequence Diagram(s)



### d. Database queries

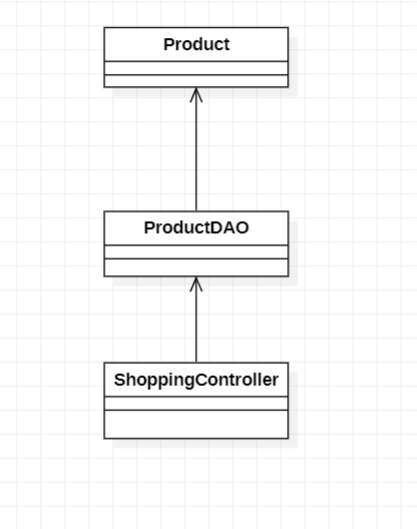
SELECT \* FROM [SWP391].[dbo].[AppUser] WHERE name=?

SELECT \* FROM [SWP391].[dbo].[AppUser] WHERE email=?

INSERT INTO AppUser (name, password, email, role\_id) VALUES (?,?,?,2);

## 3. View Product

### a. Class Diagram



### b. Class Specifications

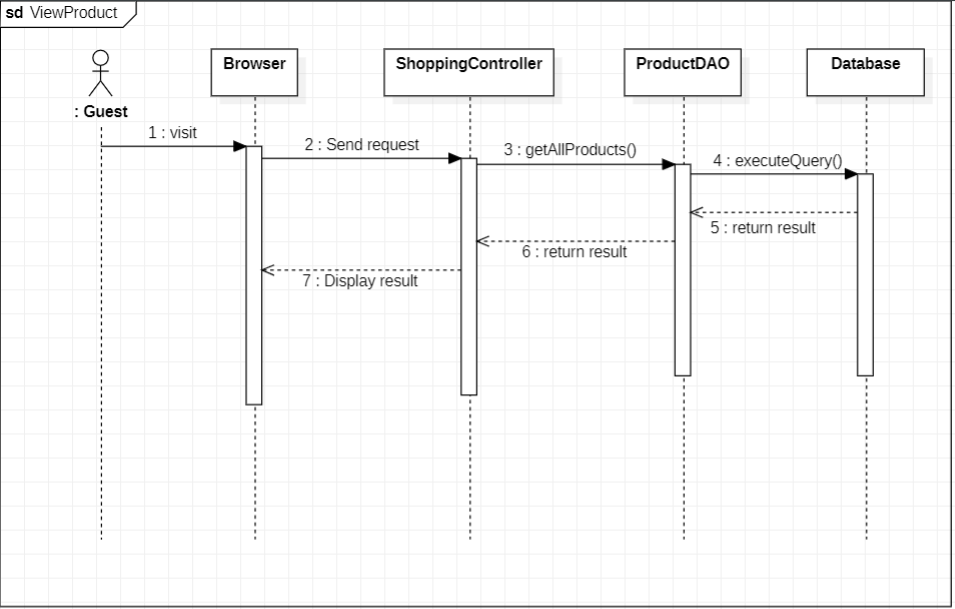
* ShoppingController Class

|  |  |  |
| --- | --- | --- |
| **No** | **Method** | **Description** |
| *01* | *doGet()* | *Used to handle HTTP GET requests, typically for retrieving data from the server.* |
| *02* | *doPost()* | *Used to handle HTTP POST requests, usually for sending data to the server.* |

* ProductDAO Class

|  |  |  |
| --- | --- | --- |
| **No** | **Method** | **Description** |
| *01* | *getAllProducts()* | *get all property in Product* |

### c. Sequence Diagram(s)

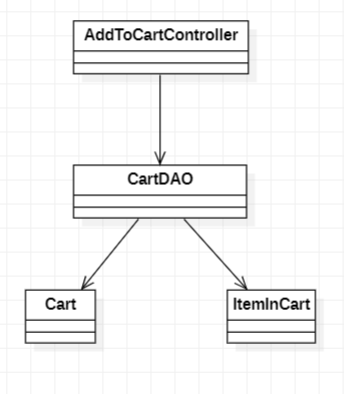
****

### d. Database queries

“SELECT \* FROM ProductInfo”

## 4. Add product to cart

### a. Class Diagram



### b. Class Specifications

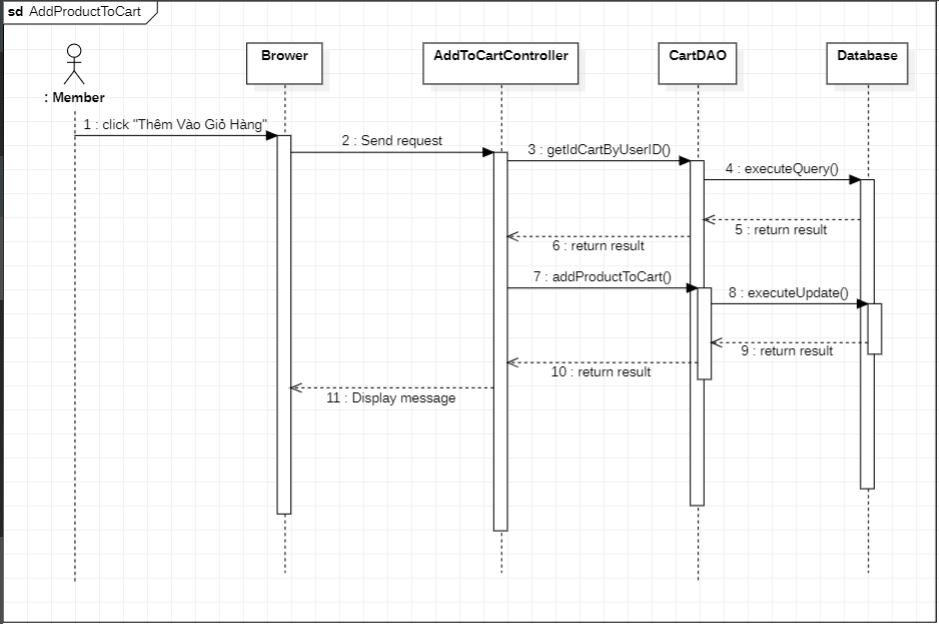
* AddToCartController Class

|  |  |  |
| --- | --- | --- |
| **No** | **Method** | **Description** |
| *01* | *doGet()* | *Used to handle HTTP GET requests, typically for retrieving data from the server.* |
| *02* | *doPost()* | *Used to handle HTTP POST requests, usually for sending data to the server.* |

* CartDAO Class

|  |  |  |
| --- | --- | --- |
| **No** | **Method** | **Description** |
| *01* | *getIdCartByUserID()* | *Create a shopping cart* |
| *02* | *addProductToCart()* | *Add products to the cart* |

### c. Sequence Diagram(s)

****

### d. Database queries

* "SELECT \*\n" +

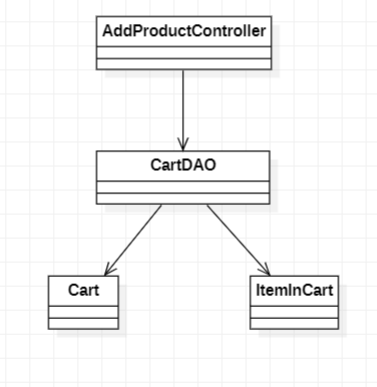
" FROM [SWP391].[dbo].[ShoppingCart]\n" +

" WHERE user\_id=?";

* “INSERT INTO ShoppingCart(user\_id) Values(?)”;
* "INSERT INTO ShoppingCartItem(product\_id,item\_quantity,cart\_id) Values(?,?,?)";

## 5. Add product

### a. Class Diagram



### b. Class Specifications

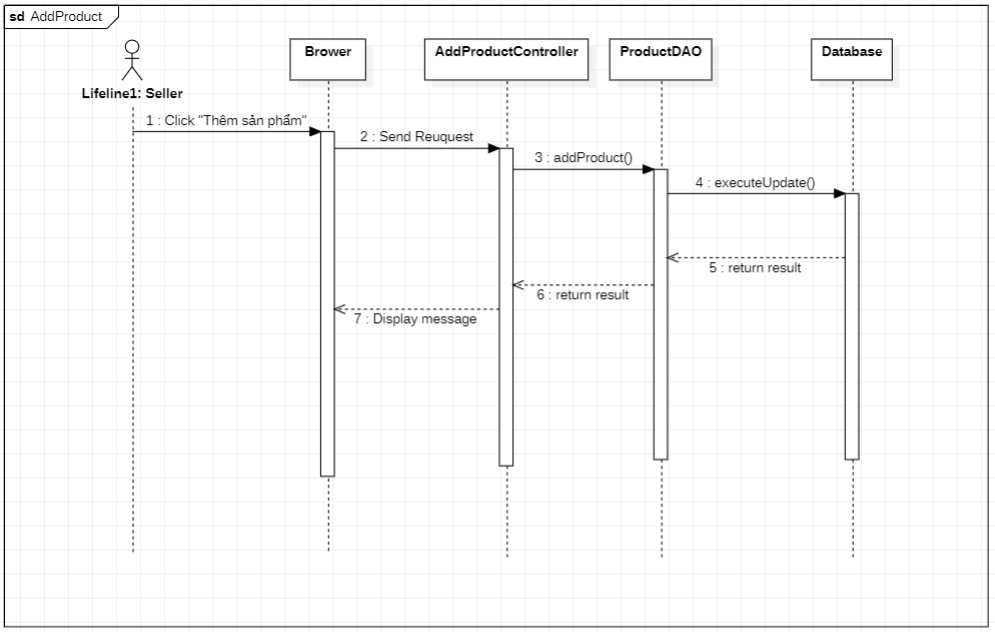
* AddProductController Class

|  |  |  |
| --- | --- | --- |
| **No** | **Method** | **Description** |
| *01* | *doGet()* | *Used to handle HTTP GET requests, typically for retrieving data from the server.* |
| *02* | *doPost()* | *Used to handle HTTP POST requests, usually for sending data to the server.* |

* ProductDAO Class

|  |  |  |
| --- | --- | --- |
| **No** | **Method** | **Description** |
| *01* | *addProduct()* | *Add information about new products to the database.* |

### c. Sequence Diagram(s)

****

### d. Database queries

* "INSERT INTO ProductInfo\n"

+ " (seller\_id\n"

+ " ,type\_id\n"

+ " ,product\_name\n"

+ " ,product\_image\n"

+ " ,product\_available\n"

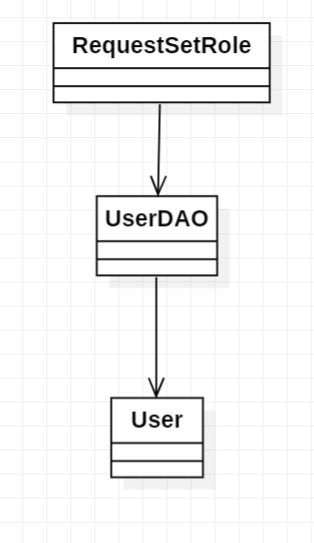
+ " ,product\_price\n"

+ " ,product\_description)\n"

+ "Values(?,?,?,?,?,?,?)";

## 6. Sales registration

### a. Class Diagram



### b. Class Specifications

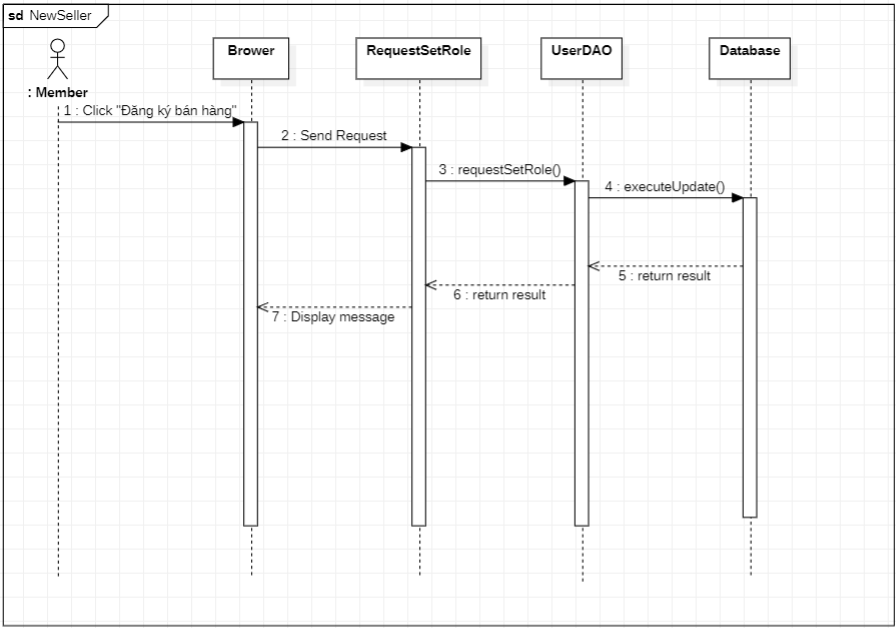
* RequestSetRole Class

|  |  |  |
| --- | --- | --- |
| **No** | **Method** | **Description** |
| *01* | *doGet()* | *Used to handle HTTP GET requests, typically for retrieving data from the server.* |
| *02* | *doPost()* | *Used to handle HTTP POST requests, usually for sending data to the server.* |

* UserDAO Class

|  |  |  |
| --- | --- | --- |
| **No** | **Method** | **Description** |
| *01* | *requestSetRole()* | *Add information about new seller to the database.* |

### c. Sequence Diagram(s)

****

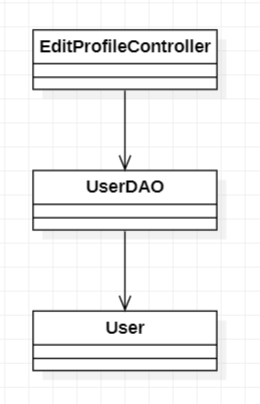
### d. Database queries

* INSERT INTO requestSetRole(user\_id,email,fullName,shopName, commoditiesSector,address,phone)

Values(?,?,?,?,?,?,?)

## 7. Edit Profile

### a. Class Diagram



### b. Class Specifications

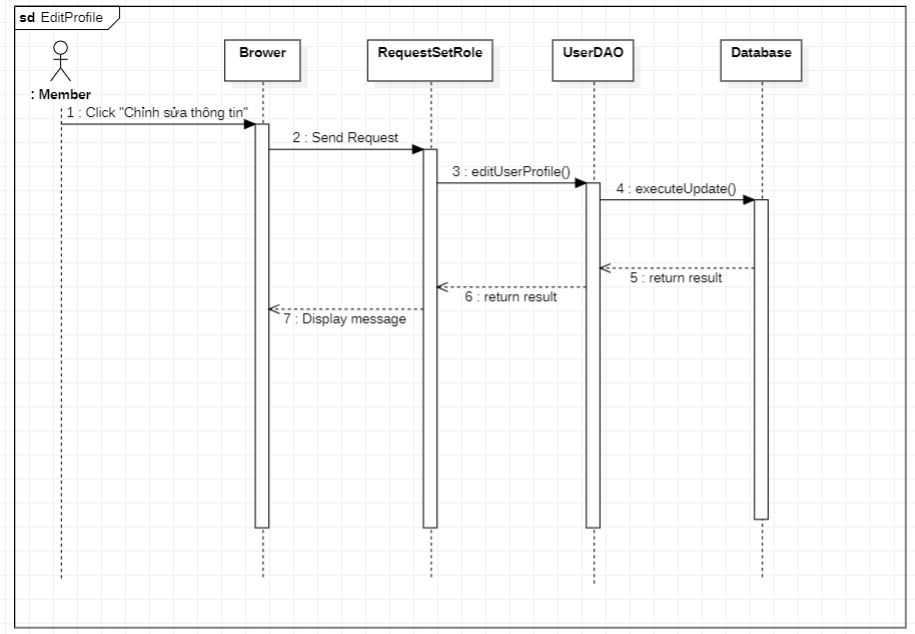
* EditProfileController Class

|  |  |  |
| --- | --- | --- |
| **No** | **Method** | **Description** |
| *01* | *doGet()* | *Used to handle HTTP GET requests, typically for retrieving data from the server.* |
| *02* | *doPost()* | *Used to handle HTTP POST requests, usually for sending data to the server.* |

* UserDAO Class

|  |  |  |
| --- | --- | --- |
| **No** | **Method** | **Description** |
| *01* | *editUserProfile()* | *Update user information in the database.* |

### c. Sequence Diagram(s)

****

### d. Database queries

* "UPDATE AppUser SET number = ?,"

+ " image = ?, fullName = ?,"

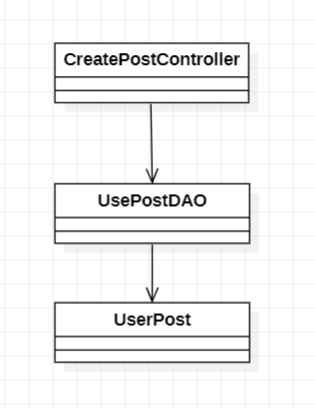
+ "gender = ?, statusNow = ?,"

+ "school = ?, favour = ?,"

+ "bio = ? WHERE id = ?";

## 8. Create Post

### a. Class Diagram



### b. Class Specifications

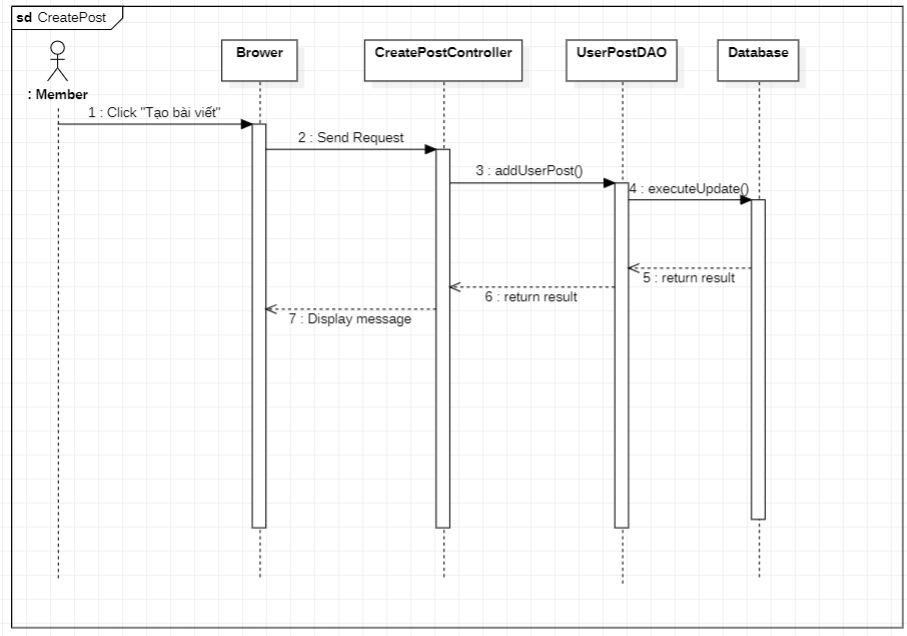
* CreatePostController Class

|  |  |  |
| --- | --- | --- |
| **No** | **Method** | **Description** |
| *01* | *doGet()* | *Used to handle HTTP GET requests, typically for retrieving data from the server.* |
| *02* | *doPost()* | *Used to handle HTTP POST requests, usually for sending data to the server.* |

* UserPostDAO Class

|  |  |  |
| --- | --- | --- |
| **No** | **Method** | **Description** |
| *01* | *addUserPost()* | *Add information of new post to the database.* |

### c. Sequence Diagram(s)

****

### d. Database queries

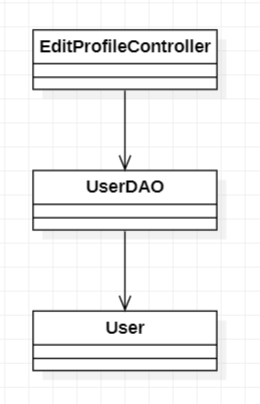
* UPDATE UserPost SET post\_title = ?,

post\_content = , post\_image = ?,

post\_image = ?, WHERE id = ?;

## 9. Edit Profile

### a. Class Diagram



### b. Class Specifications

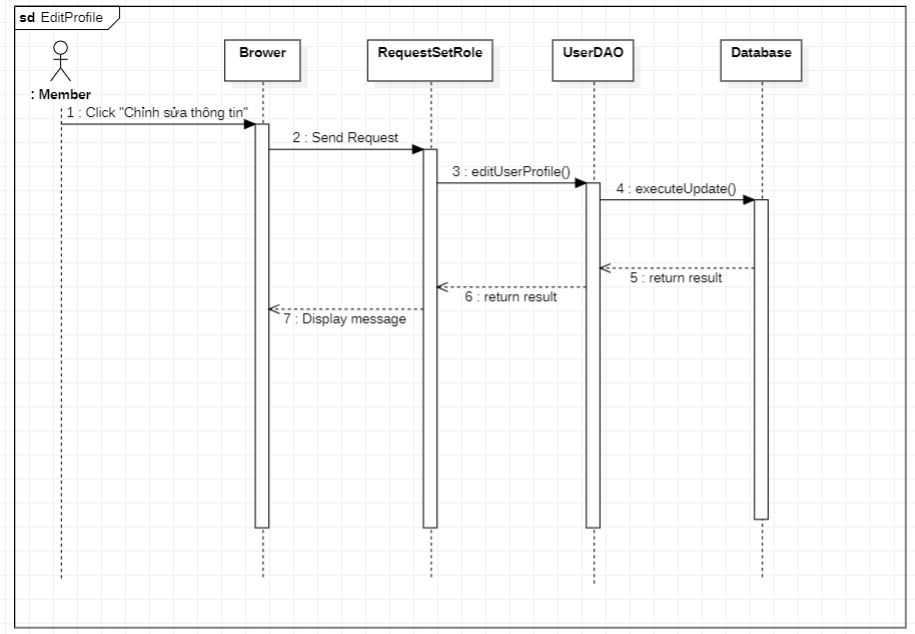
* EditProfileController Class

|  |  |  |
| --- | --- | --- |
| **No** | **Method** | **Description** |
| *01* | *doGet()* | *Used to handle HTTP GET requests, typically for retrieving data from the server.* |
| *02* | *doPost()* | *Used to handle HTTP POST requests, usually for sending data to the server.* |

* UserDAO Class

|  |  |  |
| --- | --- | --- |
| **No** | **Method** | **Description** |
| *01* | *editUserProfile()* | *Update user information in the database.* |

### c. Sequence Diagram(s)

****

### d. Database queries

* "UPDATE AppUser SET number = ?,"

+ " image = ?, fullName = ?,"

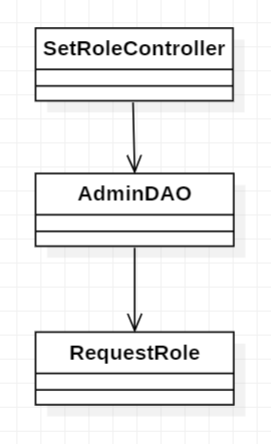
+ "gender = ?, statusNow = ?,"

+ "school = ?, favour = ?,"

+ "bio = ? WHERE id = ?";

## 10. Manage role upgrade requests

### a. Class Diagram



### b. Class Specifications

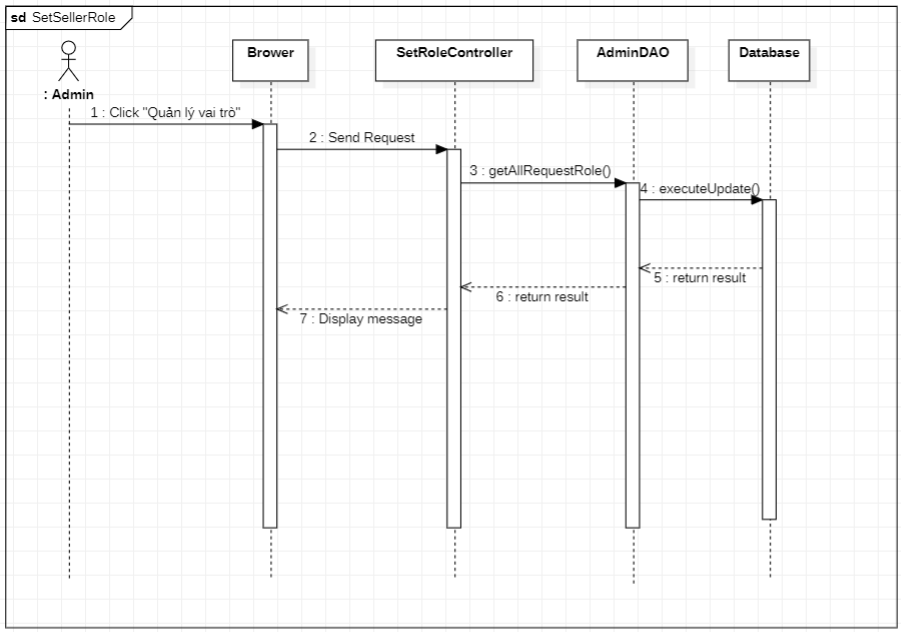
* SetRoleController Class

|  |  |  |
| --- | --- | --- |
| **No** | **Method** | **Description** |
| *01* | *doGet()* | *Used to handle HTTP GET requests, typically for retrieving data from the server.* |
| *02* | *doPost()* | *Used to handle HTTP POST requests, usually for sending data to the server.* |

* AdminDAO Class

|  |  |  |
| --- | --- | --- |
| **No** | **Method** | **Description** |
| *01* | *getAllRequestRole()* | *Get all role upgrade request list in the database.* |

### c. Sequence Diagram(s)

****

### d. Database queries

* SELECT \* FROM [SWP391].[dbo].[requestSetRole]