

**Software Design Specification**

**Project name: BoonBo Shop**

**Author: Group 5**

– Hanoi, March 2023 –

# **Record of changes**

| **Date** | **A, M, D** | **In charge** | **Description** |
| --- | --- | --- | --- |
| 08/02 | A | Mai Phuong Linh | Add package diagram |
| 09/02 | A | Can Duy Khanh | Add database schema |
| 09/2 | A | Bui Quang Truong | Code design |
| 09/2 | A | Nguyen Tuan Linh | Code design |
| 09/2 | A | Mai Ngoc Giang | Code design |
| 12/02 | A | Nguyen Tuan Linh | Code design |
| 12/02 | A | Mai Phuong Linh | Code design |
| 14/02 | A | Can Duy Khanh | Code design |
| 16/02 | A | Mai Ngoc Giang | Code design |
| 19/02 | A | Mai Phuong Linh | Package description, code design |
| 26/02 | M | Can Duy Khanh | Code design |
| 28/02 | M | Bui Quang Truong | Code design |
| 03/03 | A | Nguyen Tuan Linh | Code design |
| 03/03 | A | Mai Phuong Linh | Code design |
| 08/03 | A | Can Duy Khanh | Code design |
| 10/03 | M | Mai Ngoc Giang | Code design |
| 10/03 | A | Mai Ngoc Giang | Code design |
| 13/03 | M | Nguyen Tuan Linh | Code design |
| 13/03 | A | Nguyen Tuan Linh | Code design |
| 16/03 | M | Mai Phuong Linh | Code design |
| 16/03 | A | Mai Phuong Linh | Code design |
| 18/03 | A | All | Code design |

\*A - Added M - Modified D - Deleted

https://drive.google.com/file/d/1QbCyYP93emWjfTfAVLbDhxN1eKG4w4V5/view?fbclid=IwAR3oxsSu1nJ3mhbmZPlbv7\_v8e1qFeiGU8KjkWxUvM98Za6AHhvcRg0H1KE

# **Table of Contents**

1. **Overview**………………………………………………….

1. Code Package

2. Database Design

a. Database Schema

b. Table Description

1. **Code Designs**.

1. Register

2. View profile

3. Update profile

4. Change password

5. Forgot password

6. Shopping cart and order

7. Feedback

8. Search product

8.1 Search product by name

8.2 Search product by price

8.3 Search product by category

8.4 Search product by sort

9. Manage order

9.1 View own order

9.2 View own order detail

10. Address

10.1 View address

10.2 Add address

10.3 Edit address

10.4 Delete address

11. Home

12. Manage product

12.1 Add product

12.2 Delete product

12.3 Edit product

13. Manage size

13.1 Add size

13.2 Delete size

13.3 Edit size

14. Manage gallery

14.1 Add gallery

14.2 Delete gallery

14.3 Edit gallery

15 Manage discount

15.1 Add discount

15.2 Delete discount

15.3 Edit discount

16. Manage category

16.1 Add category

16.2 Delete category

16.3 Edit category

17. Product detail

18. Relative product

19. Statistic order, product, category, revenue

20. List order

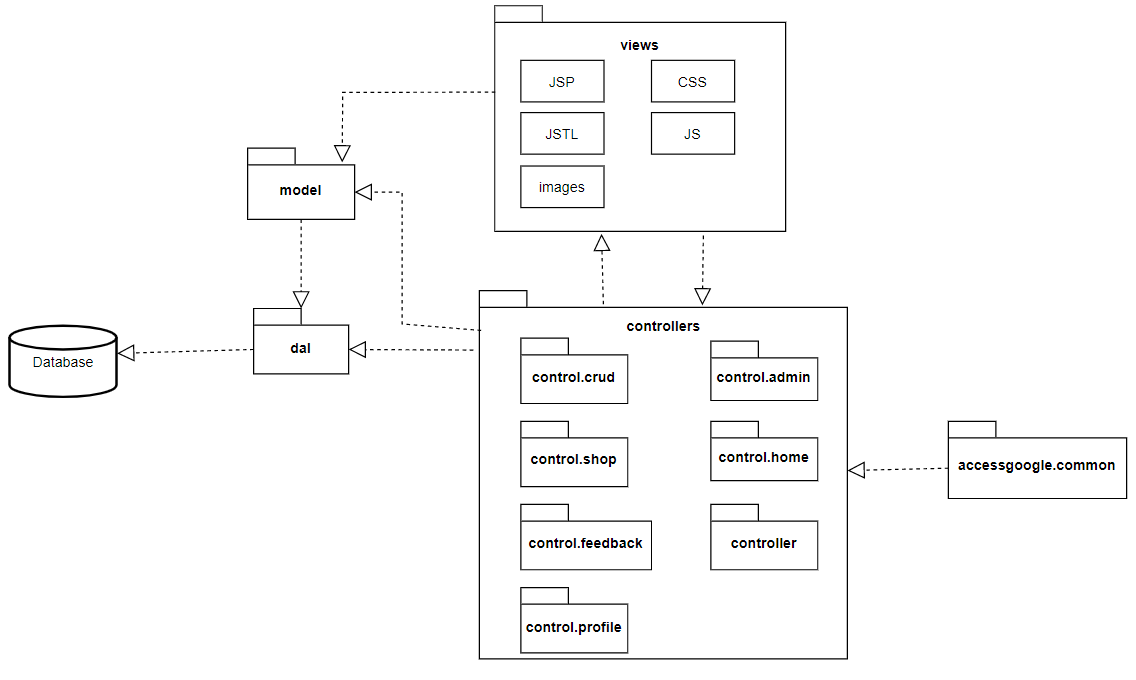
20.1 View order detail and change order’s status

21. List account, view profile and change status

21.1 Add new account and default address

# **I. Overview**

## **1. Code package**



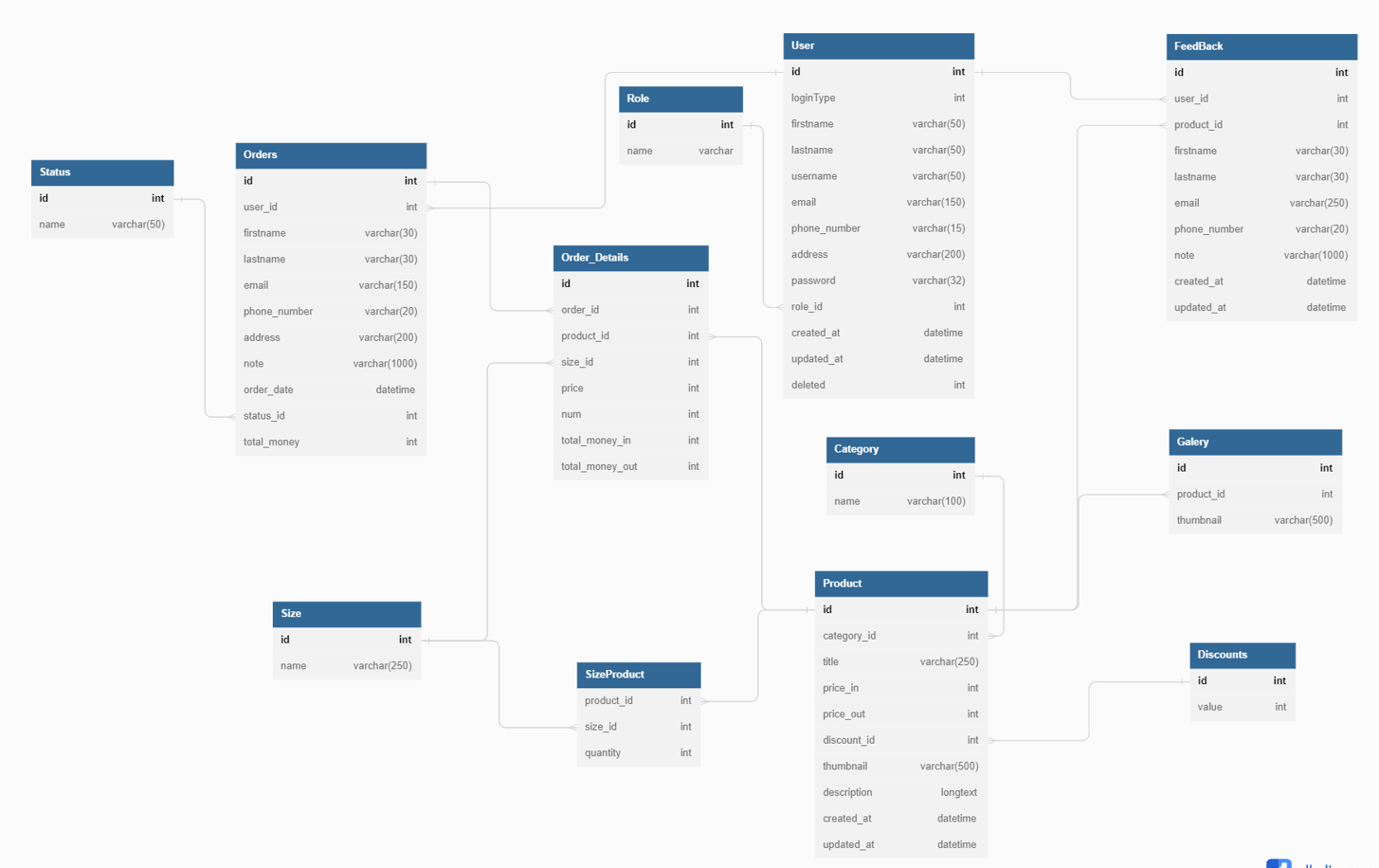
<https://app.diagrams.net/#G1QbCyYP93emWjfTfAVLbDhxN1eKG4w4V5>

**Package descriptions**

| **No** | **Package** | **Description** |
| --- | --- | --- |
| 01 | controllers | Package contains all the business classes that process the logic tasks. |
| 02 | control.crud | Package contains servlets for CRUD entity Product, Size, Gender, and other entities related to Product. |
| 03 | control.shop | Package contains servlet for implementing functions search, sort Product. |
| 04 | control.feedback | Package contains servlets for implementing functions that give feedback after order. |
| 05 | control.profile | Package contains servlets for function manage own profile |
| 06 | control.admin | Package contains servlets for dashboard |
| 07 | control.home | Package contains servlets for implementing the home page |
| 08 | control.controller | Package contains servlets for implementing common features such as login, register, and so on. |
| 09 | accessgoogle.common | Package support to function login by google |
| 10 | views | Package contains all the assets, resource to display information on the client browser. |
| 10 | dal | Package contains all the object classes that access directly to Database  The classes contained in this all have the ending suffix [DBContext] |

## **2. Database design**

**a. Database schema**

****

<https://dbdiagram.io/d/63bd22896afaa541e5d1739c>

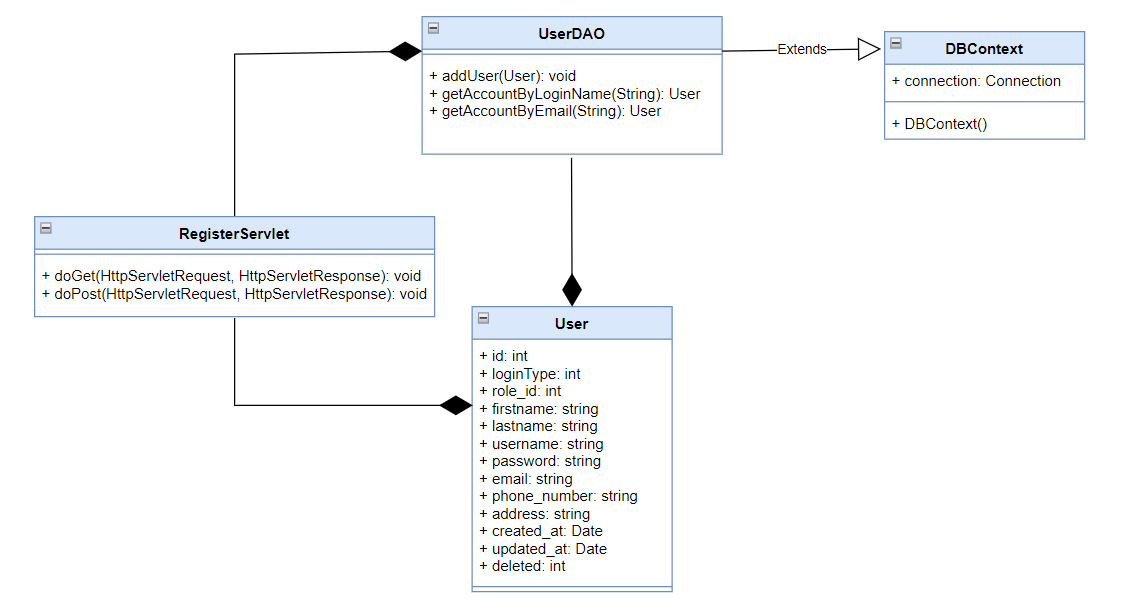
**b. Table description**

| **No** | **Table** | **Description** |
| --- | --- | --- |
| 1 | Users | Information of users |
| 2 | Roles | Role of users |
| 3 | Categories | Category of products |
| 4 | Products | Products of shop |
| 5 | Genders | Characteristics to classify products |
| 6 | Geleries | Images of products |
| 7 | Sizes | Size of products |
| 8 | SizeProduct | Classify products following sizes |
| 9 | Discounts | Discount offer for orders |
| 10 | Orders | Order placed by the customer |
| 11 | Order\_Details | Detail about order |
| 12 | Status | Status of order |
| 13 | Feedbacks | Feedbacks of customers after buy product |

# **II.** **Code Designs**

**1. Register**

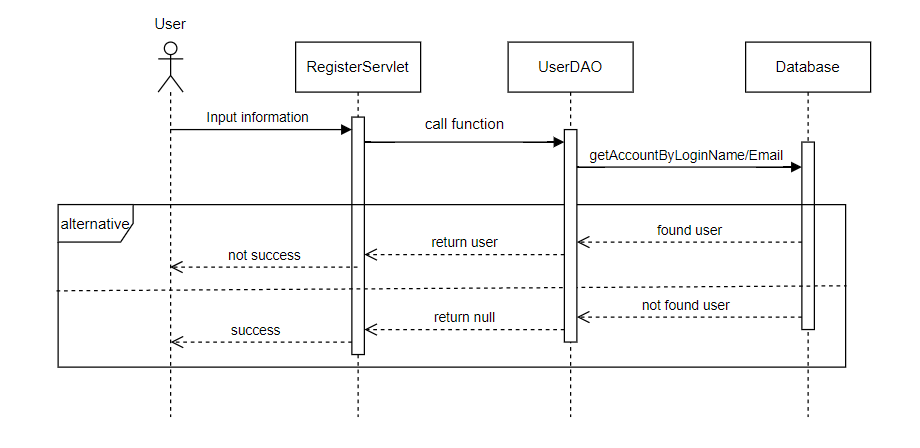
**a. Class Diagram**

****

**b. Class specifications**

| **Class** | **Method** | **Description** |
| --- | --- | --- |
| UserDAO | addUser(User): void | Add new user in the table Users in the database. |
| getAccountByLoginName(String): User | Find the user who has the same username with the given string. If it exists, return User, else, return null. |
| getAccountByEmail(String): User | Find the user who has the same email with the given string. If it exists, return User, else, return null. |
| DBContext | DBContext() | Connect to the database |
| RegisterServlet | doGet() | Send request to JSP |
| doPost() | Get parameters from JSP and do addUser() method |

**c. Sequence Diagram**

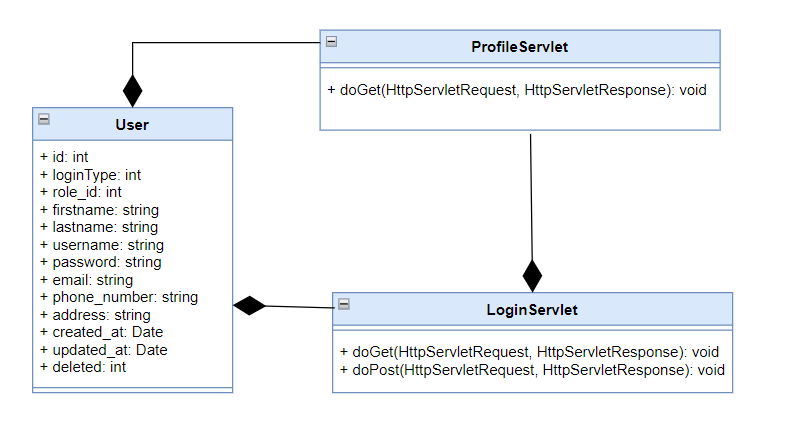
****

**d. Database Queries**

| **Class** | **Method** | **SQL** |
| --- | --- | --- |
| UserDAO | addUser(User): void | INSERT INTO [dbo].[Users]  ………………  VALUES (1, 1, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?)"; |
| getAccountByLoginName(String): User | Select \* from Users where username = ? |
| getAccountByEmail(String): User | Select \* from Users where email = ? |

**2. View profile**

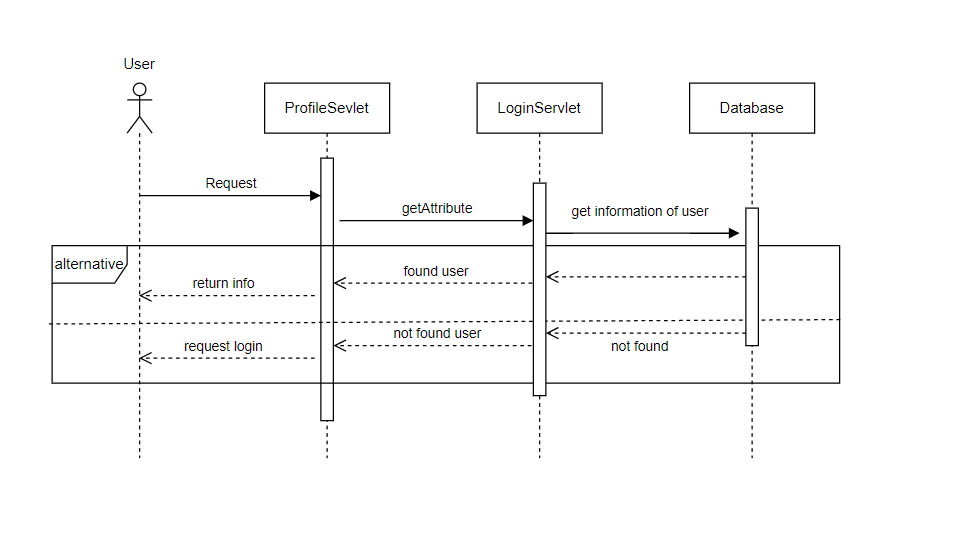
**a. Class Diagram**

****

**b. Class specifications**

| **Class** | **Method** | **Description** |
| --- | --- | --- |
| LoginLoginServlet | doGet() | Send request to JSP |
| doPost() | setAttribute |
| ProfileServlet | doGet() | Send request and attribute about user to JSP |

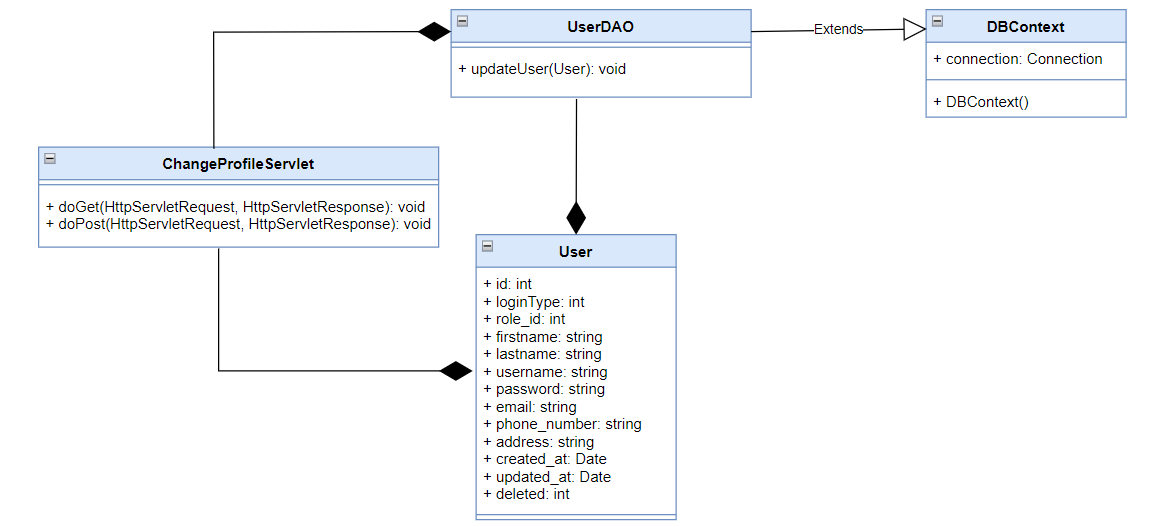
**c. Sequence Diagram**

****

**d. Database Queries**

**3. Update profile**

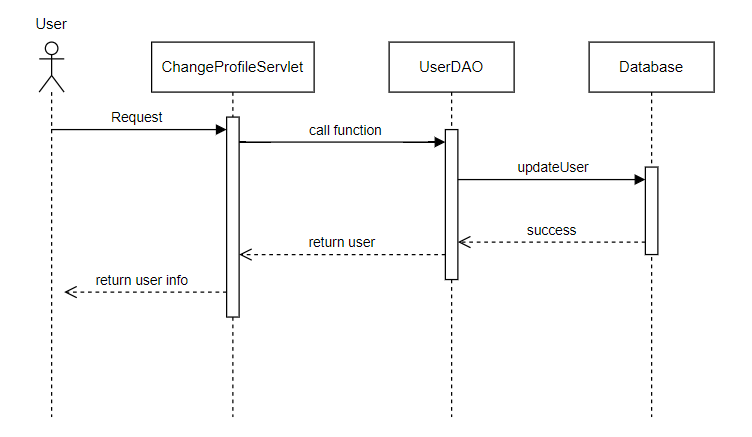
**a. Class Diagram**

****

**b. Class specifications**

| **Class** | **Method** | **Description** |
| --- | --- | --- |
| UserDAO | updateUser(User): void | Update information of user in the table Users in the database. |
| DBContext | DBContext() | Connect to the database |
| ChangeProfileServlet | doGet() | Send request to JSP |
| doPost() | Get parameters from JSP and do updateUser() method |

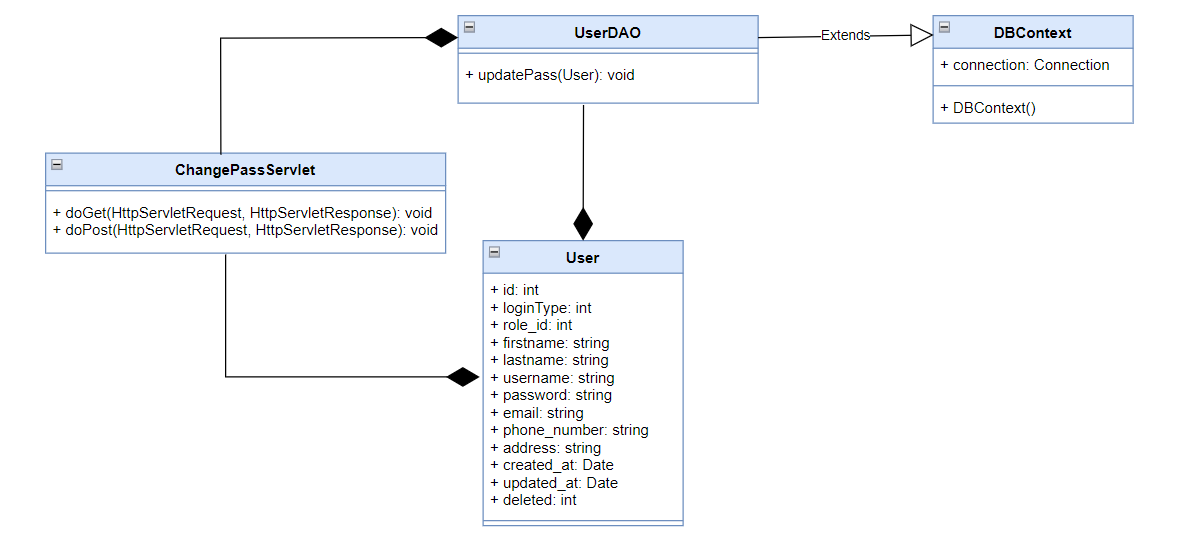
**c. Sequence Diagram**

**d. Database Queries**

| **Class** | **Method** | **SQL** |
| --- | --- | --- |
| UserDAO | updateUser(User): void | UPDATE [dbo].[Users]  SET [firstname] = ?  ,[lastname] = ?  ,[phone\_number] = ? ,[address] = ?  WHERE username = ? |

**4. Change password**

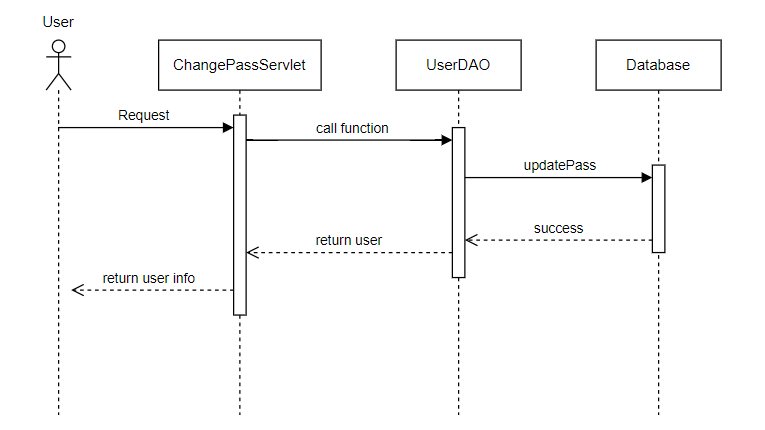
**a. Class Diagram**

****

**b. Class specifications**

| **Class** | **Method** | **Description** |
| --- | --- | --- |
| UserDAO | updatePass(User): void | Update pass of user in the table Users in the database. |
| DBContext | DBContext() | Connect to the database |
| ChangePassServlet | doGet() | Send request to JSP |
| doPost() | Get parameters from JSP and do updateUser() method |

**c. Sequence Diagram**

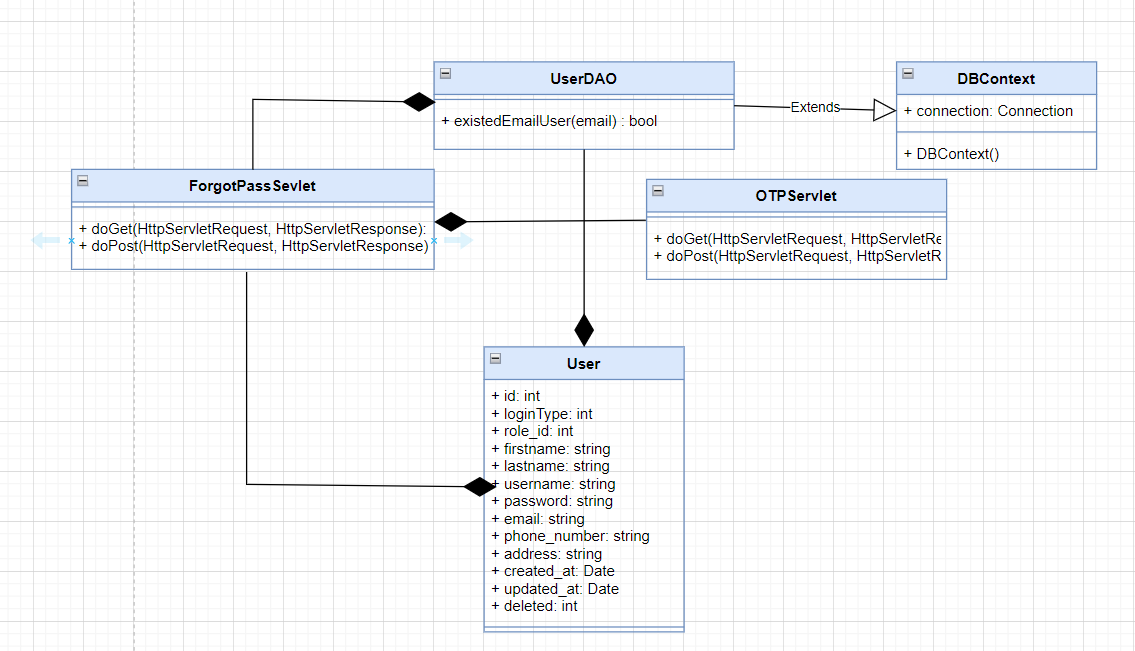
****

**d. Database Queries**

| **Class** | **Method** | **SQL** |
| --- | --- | --- |
| UserDAO | updatePass(User): void | UPDATE [dbo].[Users] SET [password] = ?  WHERE username = ? |

**5. Forgot Password**

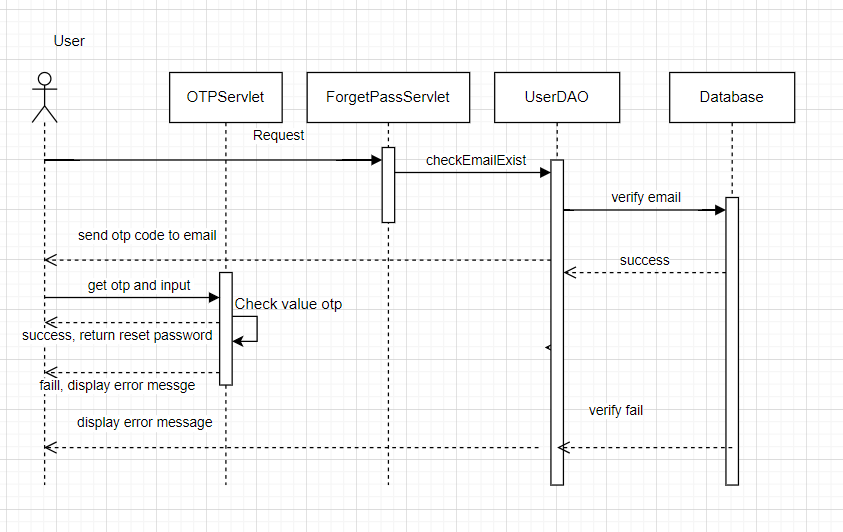
**a. Class Diagram**

****

**b. Class specifications**

| **Class** | **Method** | **Description** |
| --- | --- | --- |
| UserDAO | existedEmailUser( string email) boolean | Check if the account with this email already exists |
| DBContext | DBContext() | Connect to the database |
| ForgotPassServlet | doGet() | Send request to forgot.jsp |
|
| doPost() | Get parameters email from jsp,check and send otp to this email |
| OTPServlet |
| doGet() | Send request to otp.jsp |
|
|  | doPost() | Get parameters otp from jsp, check and send to reset.jsp |

**c. Sequence Diagram**

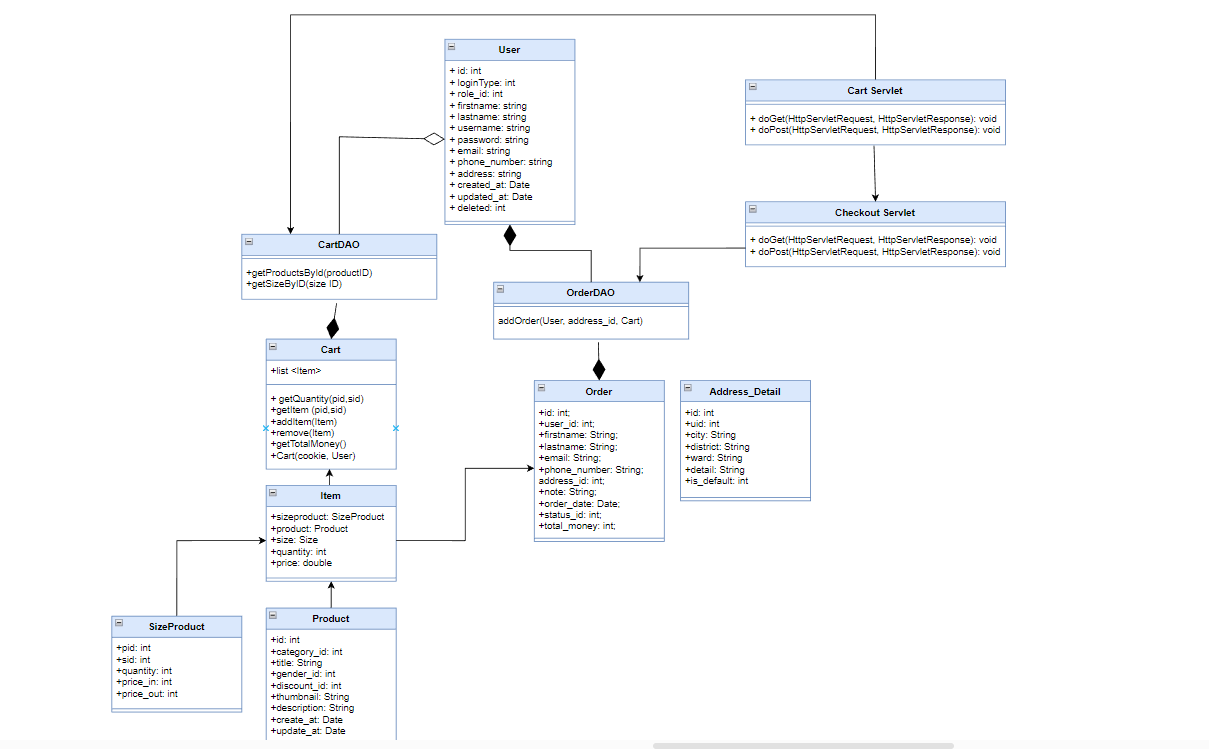
****

**d. Database Queries**

| **Class** | **Method** | **SQL** |
| --- | --- | --- |
| UserDAO | +existedEmailUser(email) : bool | SELECT [Username]\n"  + " ,[Password]\n"  + " ,[email]\n"  + " FROM [dbo].[Users]\n"  + " where email=? |

**6. Shopping Cart and Order**

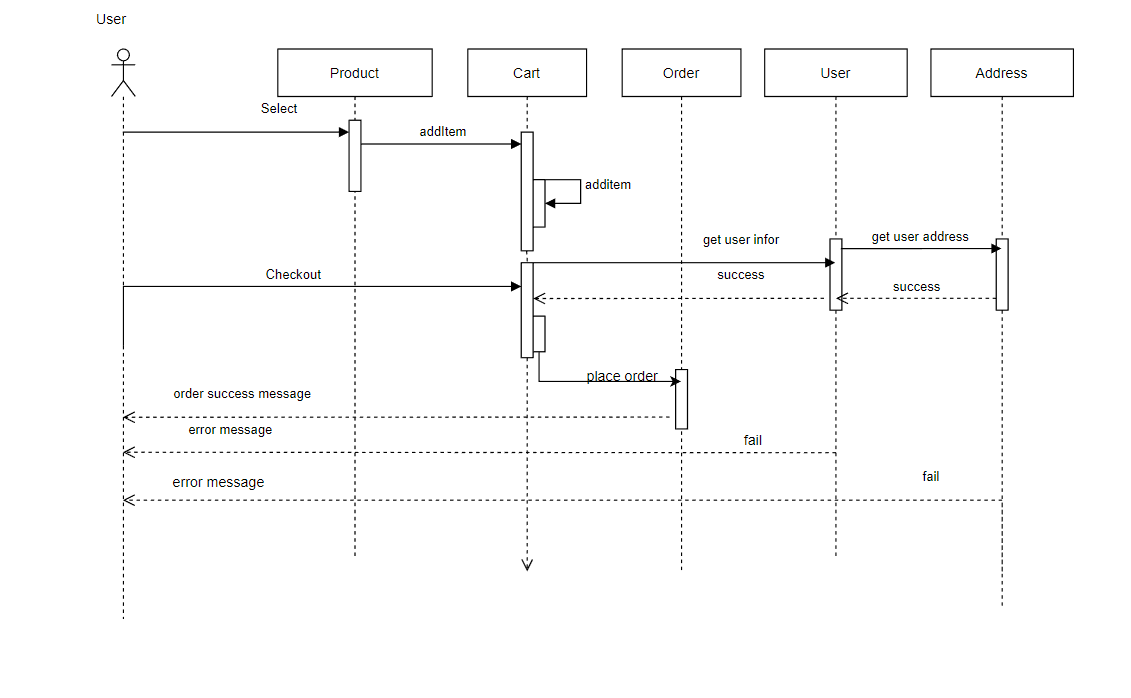
**a.Class Diagram**

****

**b. Class specifications**

| **Class** | **Method** | **Description** |
| --- | --- | --- |
| CartDAO | getProductsById(int id)  getSizeByID(int id) | Get Product by Product ID  Get Size of product by Size ID |
| OrderDAO | -getOrderDetailByOrderId(int): -List<OrderDetail>  -addOrder | Get list of detail of an order  Get list of order detail  -Add order to db |
| DBContext | DBContext() | Connect to the database |
| Cart | -List<Item> getItems()  -getQuantity(int pid, int sid)  -getItem(int pid, int sid)  -addItem(Item t)  -removeItem(int pid, int sid)  -getTotalMoney()  -Cart(String txt, User user) | -Get list Item of the cart  -Get total quantity of the cart  -get Item of the cart via product id and size id  -Add item to the cart  -Remove an item from the cart  -get total money  -split cookies via user |
| Item | Get, set |  |
|  |  |  |

**c. Sequence Diagram**

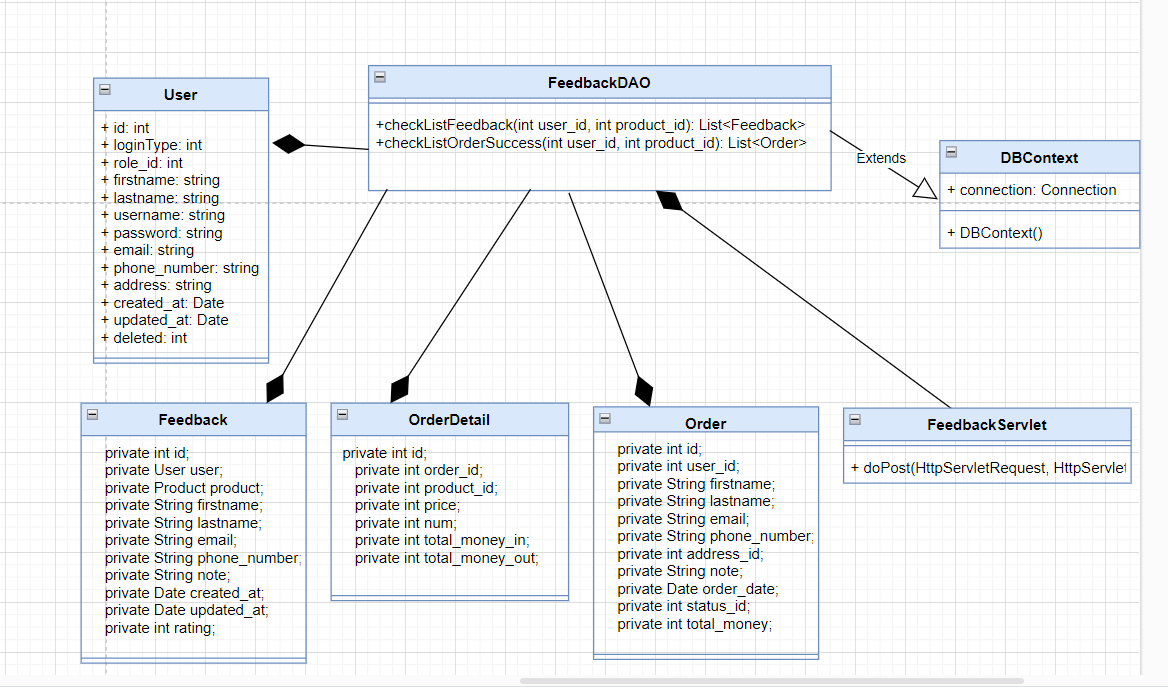
****

**d. Database Queries**

| **Class** | **Method** | **SQL** |
| --- | --- | --- |
| OrderDAO | addOrder | -insert into [orders] values(?,?,?,?,?,?,null,?,1,?)  -select top 1 id from orders order by id desc  -insert into Order\_Details values(?,?,?,?,?,?,?)  -update SizeProduct set quantity=quantity-? where pid=? AND sid=? |
| CartDAO | -getProductById(id)  -getSizeByID(size id) | -select \* from Products where id=?  -select \* from Sizes where id=? |

**7.Feedback**

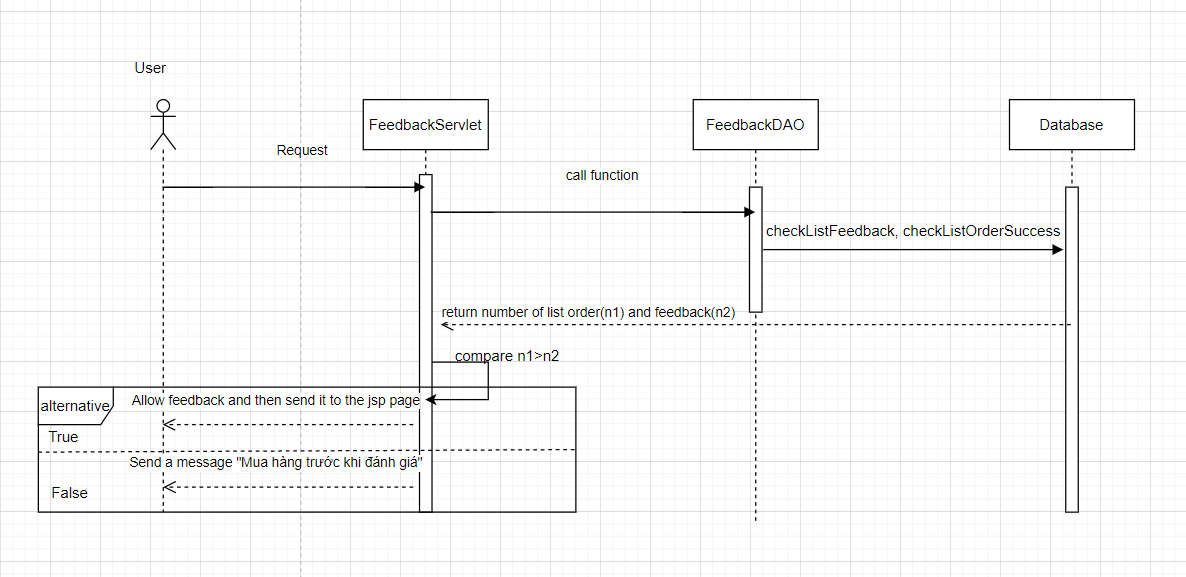
**a. Class Diagram**

****

**b. Class specifications**

| **Class** | **Method** | **Description** |
| --- | --- | --- |
| FeedbackDAO | +checkListFeedback(int user\_id, int product\_id) : List<Feedback>  +checkListOrderSucess(int user\_id, int product\_id) : List<Order> | +Returns a list of each user's successful orders for a specific product  +Returns a list of comments per user for a specific product |
| DBContext | DBContext() | Connect to the database |
| FeedbackServlet | doPost() | Get parameter by feedback form in product details servlet  Find and compare the number of orders and the amount of feedback for that product and then decide to give feedback or not |
|
|
|

**c. Sequence Diagram**

****

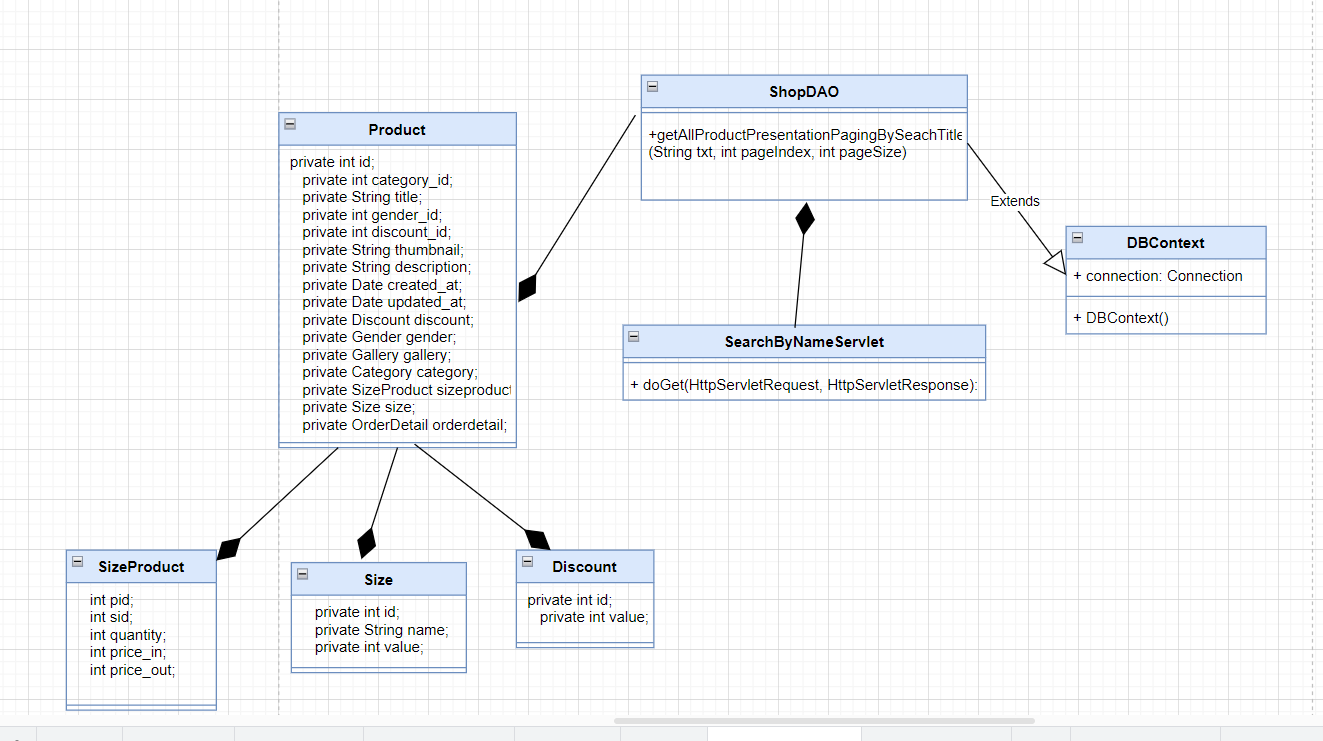
**d. Database Queries**

| **Class** | **Method** | **SQL** |
| --- | --- | --- |
| FeedbackDAO | +checkListFeedback(int user\_id, int product\_id) : List<Feedback>  +checkListOrderSucess(int user\_id, int product\_id) : List<Order> | select id from FeedBacks where user\_id = " + u + " and product\_id = " + i  select o1.status\_id from Orders o1 \n"  + "join Order\_Details o2 on o1.id = o2.order\_id\n"  + "where o1.user\_id = " + u + " and o2.product\_id = " + i + " and o1.status\_id = 3" |

**8.Search product**

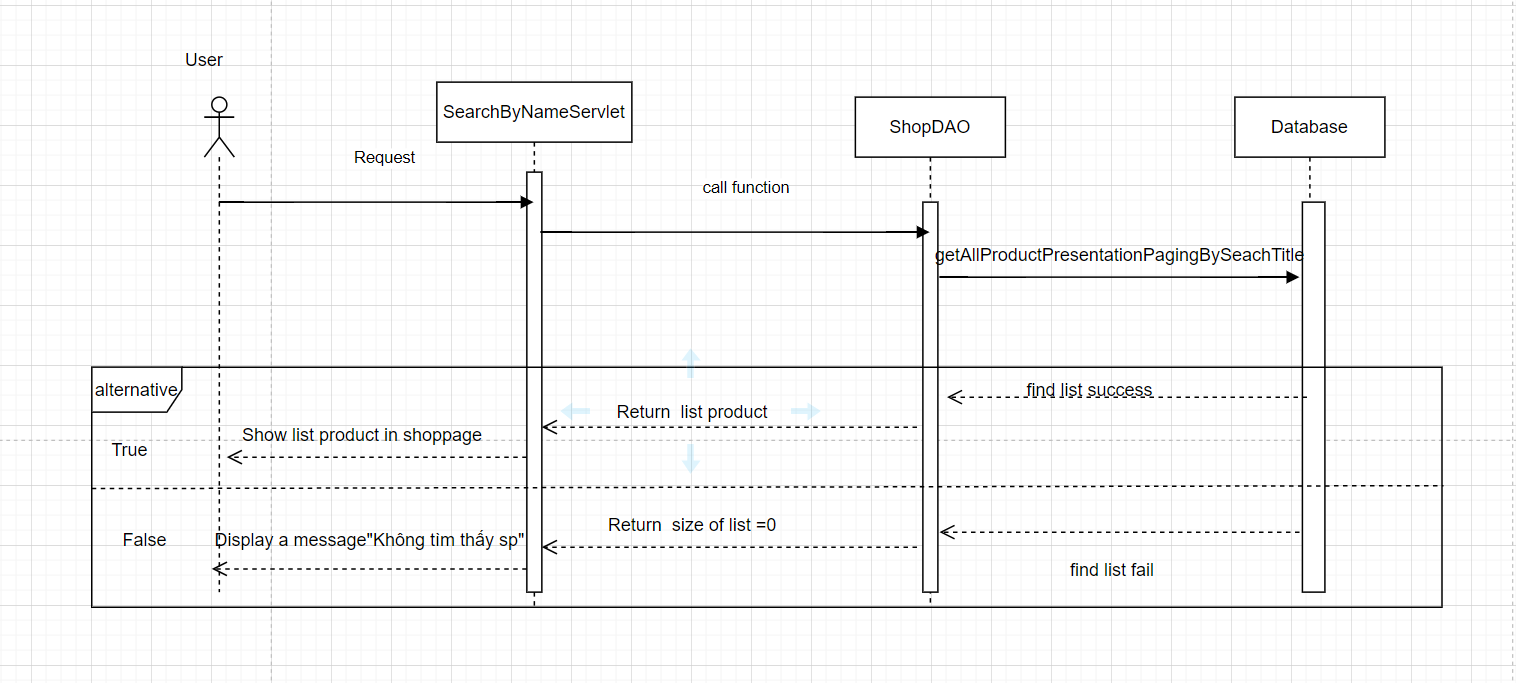
**8.1 Search product by name**

**a. Class Diagram**

** b. Class specifications**

| **Class** | **Method** | **Description** |
| --- | --- | --- |
| ShopDAO | +countAllProductBySearchTitle(String txt):int  +getAllProductPresentationPagingBySeachTitle(String txt, int pageIndex, int pageSize) : List<Product> | +Count total number of products found by name for pagination  +Returns a list of products searched by name |
| DBContext | DBContext() | Connect to the database |
| SearchByNameServlet | doGet() | Get parameters from the search form on shop.jsp  Search and return product list for each page |
|
|
|

**c. Sequence Diagram**

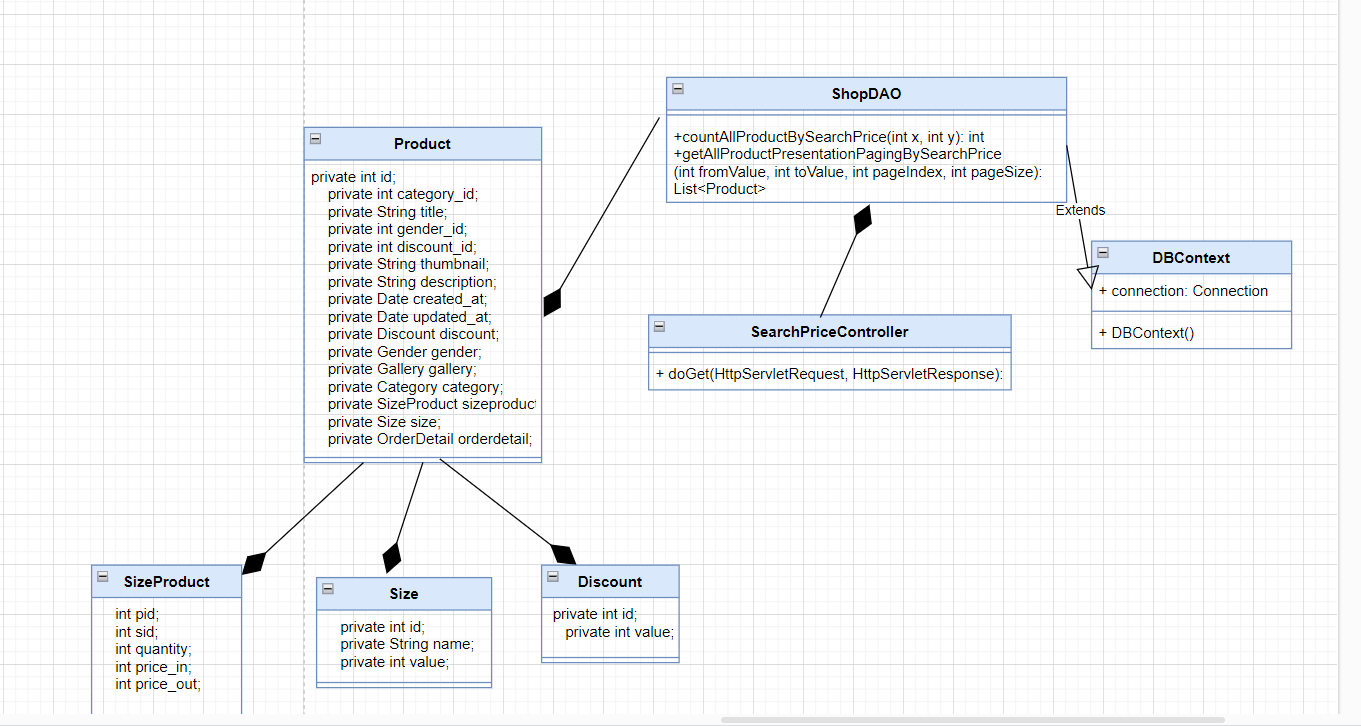
****

**d. Database Queries**

| **Class** | **Method** | **SQL** |
| --- | --- | --- |
| ShopDAO | +countAllProductBySearchTitle(String txt):int  +getAllProductPresentationPagingBySeachTitle(String txt, int pageIndex, int pageSize) : List<Product> | select count(\*) from  (select p.id,p.title, p.thumbnail,p.discount\_id,p.gender\_id,p.category\_id, d.value ,min(sp.price\_out-sp.price\_out\*d.value/100) as price, min(sp.sid) as sid  from Products p  join SizeProduct sp on p.id = sp.pid  join Sizes s on s.id = sp.sid  join Discounts d on d.id = p.discount\_id  group by p.id,p.title, p.thumbnail , p.discount\_id, d.value,p.gender\_id,p.category\_id  having p.title like '%morra%'  ) as p    select p.id,p.title, p.thumbnail, p.discount\_id,p.gender\_id,p.category\_id, d.value ,min(sp.price\_out-sp.price\_out\*d.value/100) as price, min(sp.sid) as sid  from Products p  join SizeProduct sp on p.id = sp.pid  join Sizes s on s.id = sp.sid  join Discounts d on d.id = p.discount\_id  group by p.id,p.title, p.thumbnail , p.discount\_id, d.value,p.gender\_id,p.category\_id  having p.title like ?  order by p.id  OFFSET (?-1)\*? rows FETCH NEXT ? rows only |

**8.2 Search product by price**

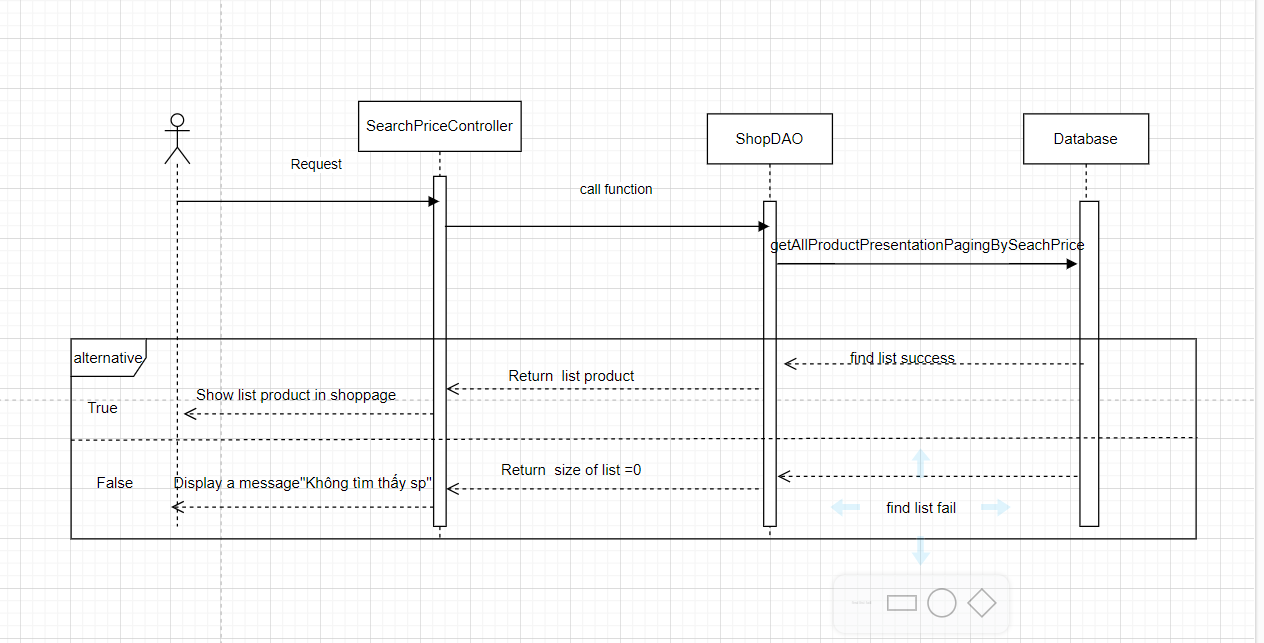
**a. Class Diagram**

****

**b. Class specifications**

| **Class** | **Method** | **Description** |
| --- | --- | --- |
| ShopDAO | +countAllProductBySearchPrice(int x,int y):int  +getAllProductPresentationPagingBySeachPrice(int x,int y, int pageIndex, int pageSize) : List<Product> | +Count total number of products found by price for pagination  +Returns a list of products searched by price |
| DBContext | DBContext() | Connect to the database |
| SearchPriceController | doGet() | Get parameters from the search form on shop.jsp  Search and return product list for each page  If don't find list return a message “Không tìm thấy sản phẩm phù hợp” |
|
|
|

**c. Sequence Diagram**

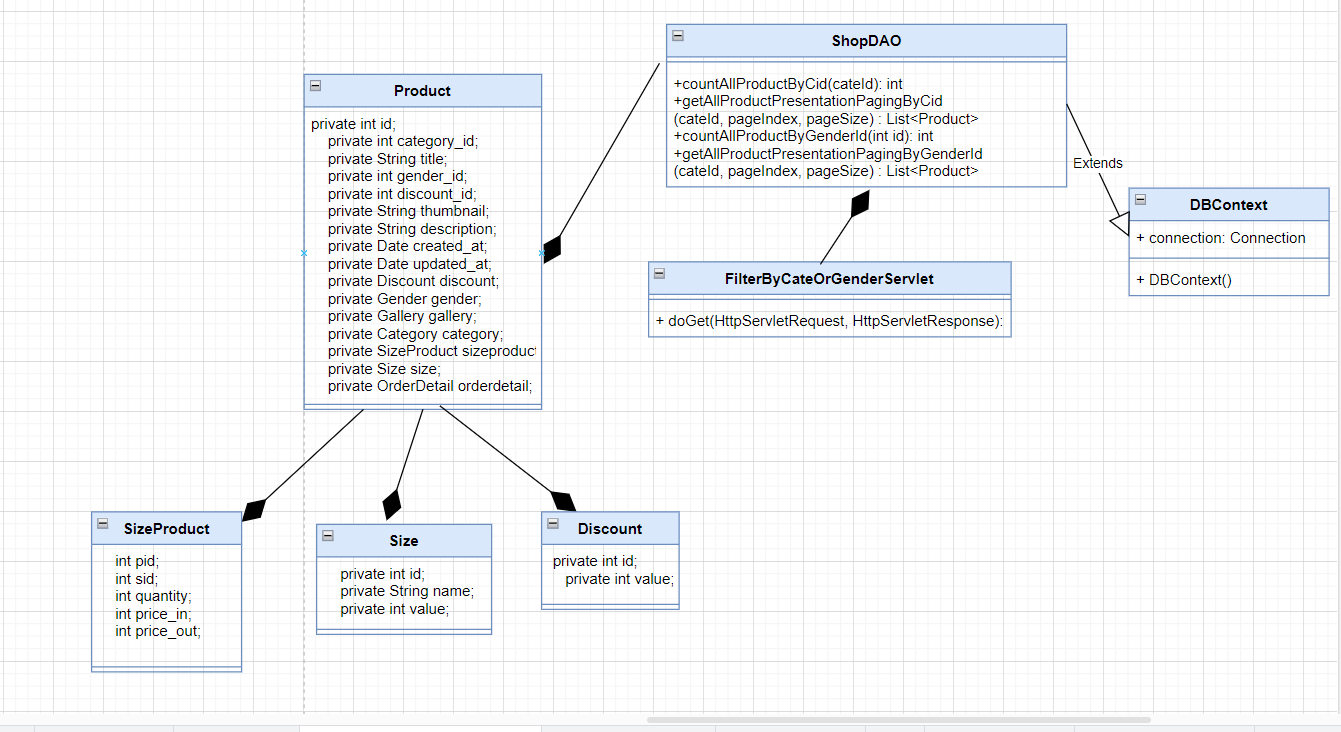
****

**d. Database Queries**

| **Class** | **Method** | **SQL** |
| --- | --- | --- |
| ShopDAO | +countAllProductBySearchPrice(int x,int y):int  +getAllProductPresentationPagingBySeachPrice(int x,int y, int pageIndex, int pageSize) : List<Product> | select count(\*) from (select p.id,p.title, p.thumbnail, p.discount\_id,p.gender\_id,p.category\_id, d.value ,min(sp.price\_out-sp.price\_out\*d.value/100) as price, min(sp.sid) as sid  from Products p join SizeProduct sp on p.id = sp.pid  join Sizes s on s.id = sp.sid  join Discounts d on d.id = p.discount\_id  group by p.id,p.title, p.thumbnail , p.discount\_id, d.value,p.gender\_id,p.category\_id) as t  where t.price between ? and ?    select \* from (select p.id,p.title, p.thumbnail, p.discount\_id,p.gender\_id,p.category\_id, d.value ,min(sp.price\_out-sp.price\_out\*d.value/100) as price, min(sp.sid) as sid  from Products p join SizeProduct sp on p.id = sp.pid  join Sizes s on s.id = sp.sid  join Discounts d on d.id = p.discount\_id  group by p.id,p.title, p.thumbnail , p.discount\_id, d.value,p.gender\_id,p.category\_id) as t  where t.price between ? and ?  order by t.price  OFFSET (?-1)\*? rows  FETCH NEXT ? rows only |

**8.3 Search product by category,gender**

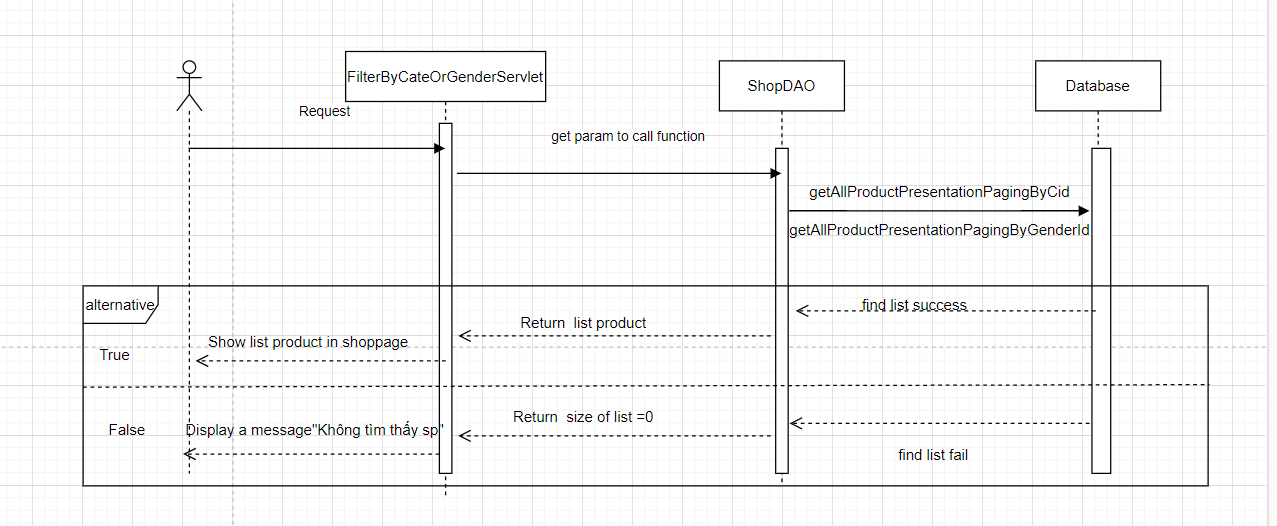
**a. Class Diagram**

****

**b. Class specifications**

| **Class** | **Method** | **Description** |
| --- | --- | --- |
| ShopDAO | +countAllProductByCid(cateId): int  +getAllProductPresentationPagingByCid(cateId, pageIndex, pageSize) : List<Product>  +countAllProductByGenderId(int id): int  +getAllProductPresentationPagingByGenderId(cateId, pageIndex, pageSize) : List<Product> | +Count total number of products found by category id for pagination  +Returns a list of products searched by category\_id  +Count total number of products found by category id for pagination  +Returns a list of products searched by category\_id |
| DBContext | DBContext() | Connect to the database |
| FilterByCateOrGenderServlet | doGet() | Get parameters from the category menu or gender menu on shop.jsp  Search and return product list for each page  If don't find list return a message “Không tìm thấy sản phẩm phù hợp” |
|
|
|

**c. Sequence Diagram**

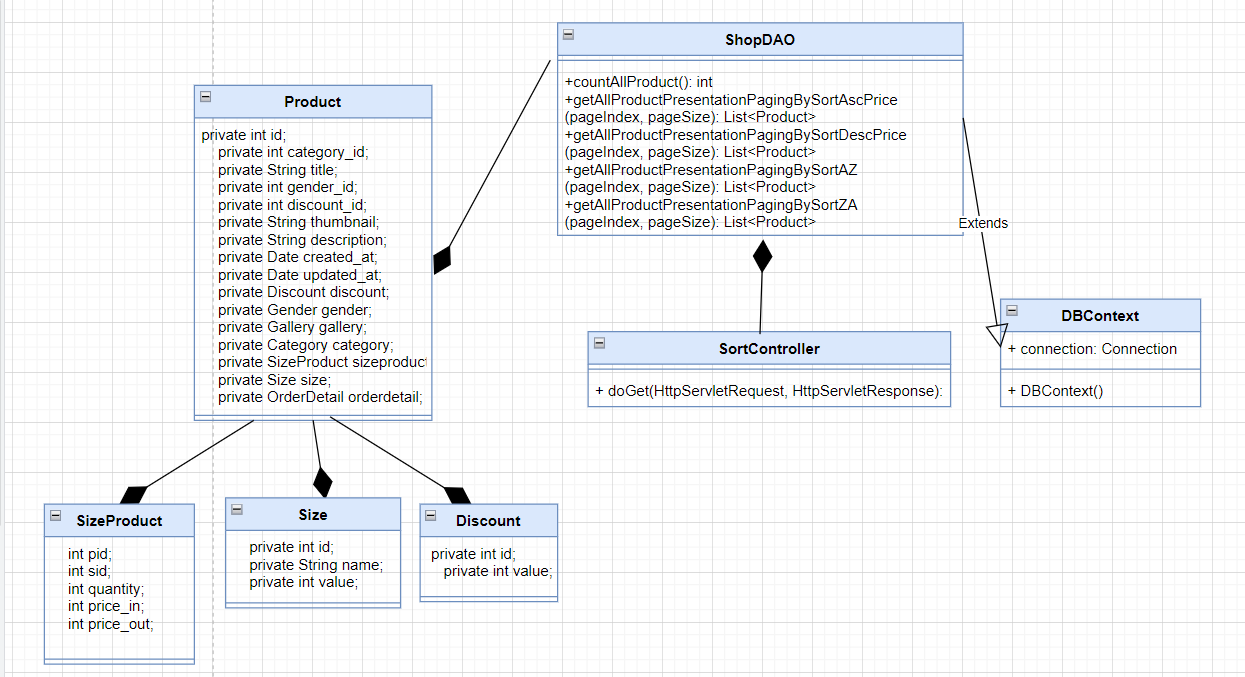
****

**d. Database Queries**

| **Class** | **Method** | **SQL** |
| --- | --- | --- |
| ShopDAO | +countAllProductByCid(cateId): int  +getAllProductPresentationPagingByCid(cateId, pageIndex, pageSize) : List<Product>  +countAllProductByGenderId(int id): int  +getAllProductPresentationPagingByGenderId(id, pageIndex, pageSize) : List<Product> | +select count(\*) from (select p.id,p.title, p.thumbnail, p.discount\_id,p.gender\_id,p.category\_id, d.value ,min(sp.price\_out-sp.price\_out\*d.value/100) as price, min(sp.sid) as sid  from Products p  join SizeProduct sp on p.id = sp.pid  join Sizes s on s.id = sp.sid  join Discounts d on d.id = p.discount\_id  group by p.id,p.title, p.thumbnail , p.discount\_id, d.value,p.gender\_id,p.category\_id  having p.category\_id= ?  ) as p  +select p.id,p.title, p.thumbnail, p.discount\_id,p.gender\_id,p.category\_id, d.value ,min(sp.price\_out-sp.price\_out\*d.value/100) as price, min(sp.sid) as sid  from Products p  join SizeProduct sp on p.id = sp.pid  join Sizes s on s.id = sp.sid  join Discounts d on d.id = p.discount\_id  group by p.id,p.title, p.thumbnail , p.discount\_id, d.value,p.gender\_id,p.category\_id  having p.category\_id= ?  order by p.id  OFFSET (?-1)\*? rows  FETCH NEXT ? rows only  +select count(\*) from (select p.id,p.title, p.thumbnail, p.discount\_id,p.gender\_id,p.category\_id, d.value ,min(sp.price\_out-sp.price\_out\*d.value/100) as price, min(sp.sid) as sid  from Products p  join SizeProduct sp on p.id = sp.pid  join Sizes s on s.id = sp.sid  join Discounts d on d.id = p.discount\_id  group by p.id,p.title, p.thumbnail , p.discount\_id, d.value,p.gender\_id,p.category\_id  having p.gender\_id= ?  ) as p  +select p.id,p.title, p.thumbnail, p.discount\_id,p.gender\_id,p.category\_id, d.value ,min(sp.price\_out-sp.price\_out\*d.value/100) as price, min(sp.sid) as sid  from Products p  join SizeProduct sp on p.id = sp.pid  join Sizes s on s.id = sp.sid  join Discounts d on d.id = p.discount\_id  group by p.id,p.title, p.thumbnail , p.discount\_id, d.value,p.gender\_id,p.category\_id  having p.gender\_id= ?  order by p.id  OFFSET (?-1)\*? rows  FETCH NEXT ? rows only |

**8.4 Search product by sort**

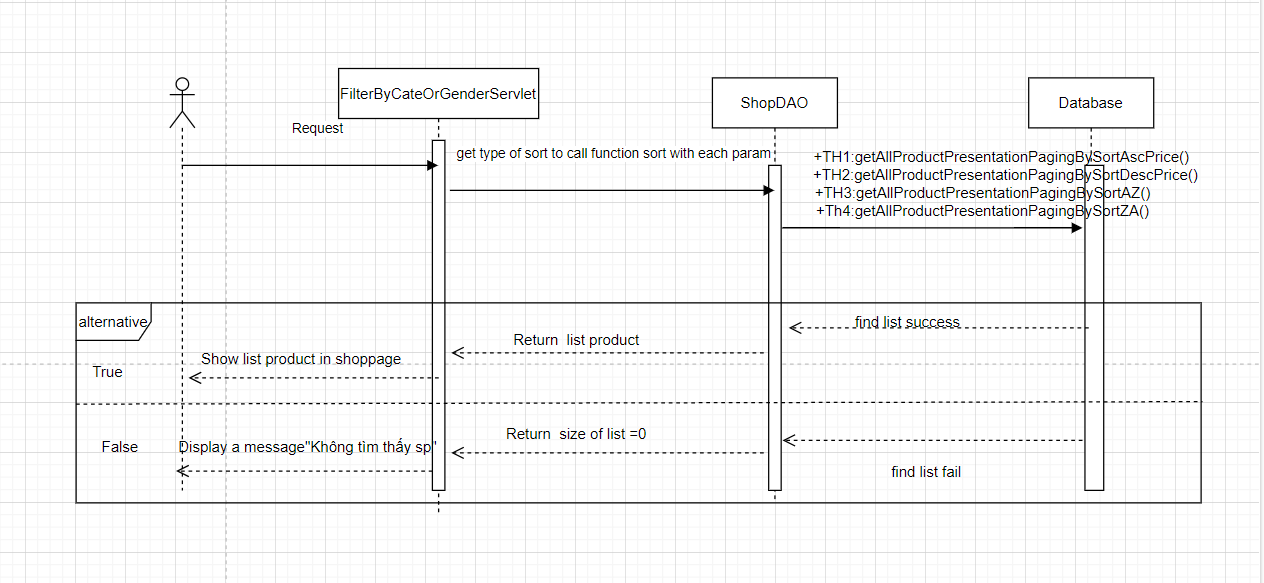
**a. Class Diagram**

****

**b. Class specifications**

| **Class** | **Method** | **Description** |
| --- | --- | --- |
| ShopDAO | +countAllProduct(): int  +getAllProductPresentationPagingBySortAscPrice() : List<Product>  +getAllProductPresentationPagingBySortDescPrice() : List<Product>: int  +getAllProductPresentationPagingBySortAZ(pageIndex, pageSize) : List<Product>  +getAllProductPresentationPagingBySortZA(pageIndex, pageSize) : List<Product> | +Count total number of products found for pagination  +Returns a list of products searched by sorting ascending by price  +Returns a list of products searched by sorting descending by price  +Returns a list of products searched by sorting ascending by title  +Returns a list of products searched by sorting descending by title |
| DBContext | DBContext() | Connect to the database |
| FilterByCateOrGenderServlet | doGet() | Get type of sort from the select-option form on shop.jsp  Search and return product list for each page  If don't find list return a message “Không tìm thấy sản phẩm phù hợp” |
|
|
|

**c. Sequence Diagram**

****

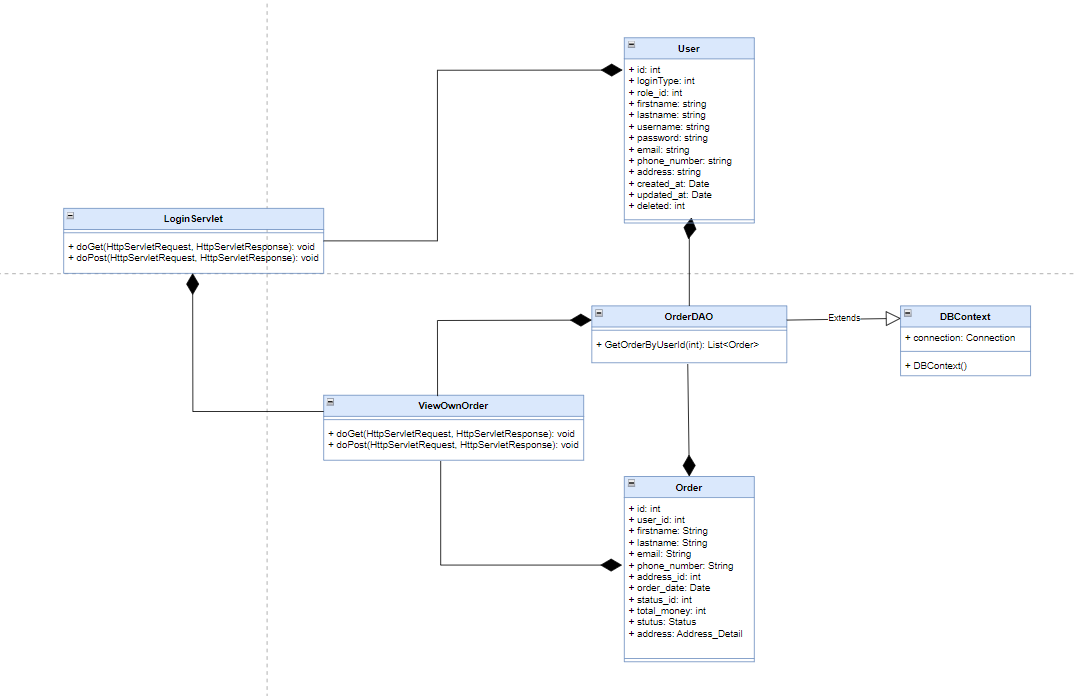
**d. Database Queries**

| **Class** | **Method** | **SQL** |
| --- | --- | --- |
| ShopDAO | +countAllProduct(): int  +getAllProductPresentationPagingBySortAscPrice() : List<Product>  +getAllProductPresentationPagingBySortDescPrice(): List<Product>  +getAllProductPresentationPagingBySortAZ() : List<Product>  +getAllProductPresentationPagingBySortZA() : List<Product> | +select count(\*)from (select p.id,p.title, p.thumbnail, p.discount\_id,p.gender\_id,p.category\_id, d.value ,min(sp.price\_out-sp.price\_out\*d.value/100) as price, min(sp.sid) as sid  from Products p  join SizeProduct sp on p.id = sp.pid  join Sizes s on s.id = sp.sid  join Discounts d on d.id = p.discount\_id  group by p.id,p.title, p.thumbnail , p.discount\_id, d.value,p.gender\_id,p.category\_id) as P  + select p.id,p.title, p.thumbnail, p.discount\_id,p.gender\_id,p.category\_id, d.value ,min(sp.price\_out-sp.price\_out\*d.value/100) as price, min(sp.sid) as sid  from Products p  join SizeProduct sp on p.id = sp.pid  join Sizes s on s.id = sp.sid  join Discounts d on d.id = p.discount\_id  group by p.id,p.title, p.thumbnail , p.discount\_id, d.value,p.gender\_id,p.category\_id  order by price asc  OFFSET (?-1)\*? rows FETCH NEXT ? rows only  + select p.id,p.title, p.thumbnail, p.discount\_id,p.gender\_id,p.category\_id, d.value ,min(sp.price\_out-sp.price\_out\*d.value/100) as price, min(sp.sid) as sid  from Products p  join SizeProduct sp on p.id = sp.pid  join Sizes s on s.id = sp.sid  join Discounts d on d.id = p.discount\_id  group by p.id,p.title, p.thumbnail , p.discount\_id, d.value,p.gender\_id,p.category\_id  order by price desc  OFFSET (?-1)\*? rows FETCH NEXT ? rows only  + select p.id,p.title, p.thumbnail, p.discount\_id,p.gender\_id,p.category\_id, d.value ,min(sp.price\_out-sp.price\_out\*d.value/100) as price, min(sp.sid) as sid  from Products p  join SizeProduct sp on p.id = sp.pid  join Sizes s on s.id = sp.sid  join Discounts d on d.id = p.discount\_id  group by p.id,p.title, p.thumbnail , p.discount\_id, d.value,p.gender\_id,p.category\_id  order by by p.title asc  OFFSET (?-1)\*? rows FETCH NEXT ? rows only  + select p.id,p.title, p.thumbnail, p.discount\_id,p.gender\_id,p.category\_id, d.value ,min(sp.price\_out-sp.price\_out\*d.value/100) as price, min(sp.sid) as sid  from Products p  join SizeProduct sp on p.id = sp.pid  join Sizes s on s.id = sp.sid  join Discounts d on d.id = p.discount\_id  group by p.id,p.title, p.thumbnail , p.discount\_id, d.value,p.gender\_id,p.category\_id  order by by p.title desc  OFFSET (?-1)\*? rows FETCH NEXT ? rows only |

**9. Manage own orders**

**9.1 View own orders**

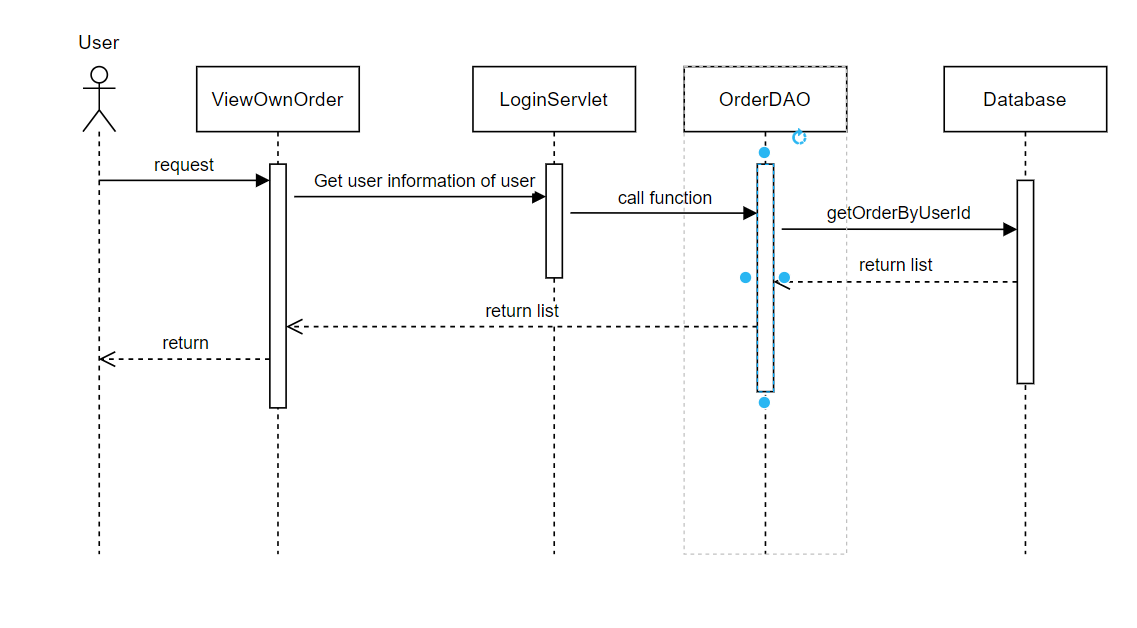
**a. Class Diagram**

****

**b. Class specifications**

| **Class** | **Method** | **Description** |
| --- | --- | --- |
| Order | Get, set | Get, set attribute of class |
| User | Get, set | Get, set attribute of class |
| OrderDAO | getOrderByUserId(int): List<Order> | Get list of orders of an user |
| DBContext | DBContext() | Connect to the database |
| ViewOwnOrder | doGet() | Get parameter from JSP and call function from OrderDAO, set attribute and send to JSP |

**c. Sequence Diagram**

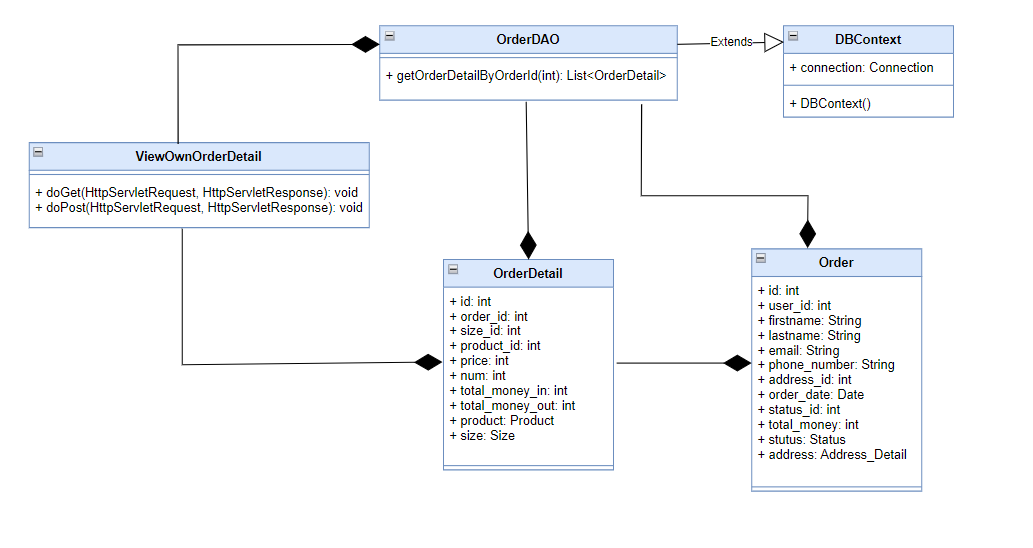
****

**d. Database Queries**

| **Class** | **Method** | **SQL** |
| --- | --- | --- |
| OrderDAO | +getOrderByUserId(int): List<Order> | select o.id as oid, user\_id, firstname, lastname, email, phone\_number, address\_id, note,  order\_date, status\_id, total\_money, s.name as sname  from Orders o join Status s on o.status\_id = s.id where user\_id = ? |

**9.2. View own order detail**

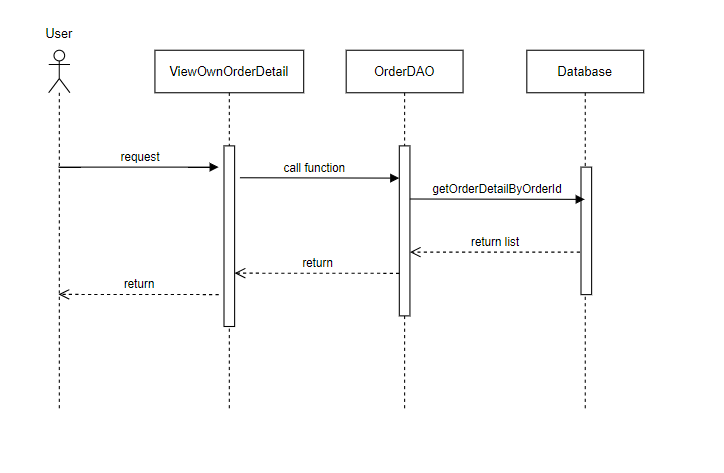
**a. Class Diagram**

****

**b. Class specifications**

| **Class** | **Method** | **Description** |
| --- | --- | --- |
| Order | Get, set | Get, set attribute of class |
| OrderDetail | Get, set | Get, set attribute of class |
|  |  |  |
|  |  |  |
| DBContext | DBContext() | Connect to the database |
| RegisterServlet | doGet() | Get parameter from JSP and call function from OrderDAO set attribute and send to JSP |

**c. Sequence Diagram**

****

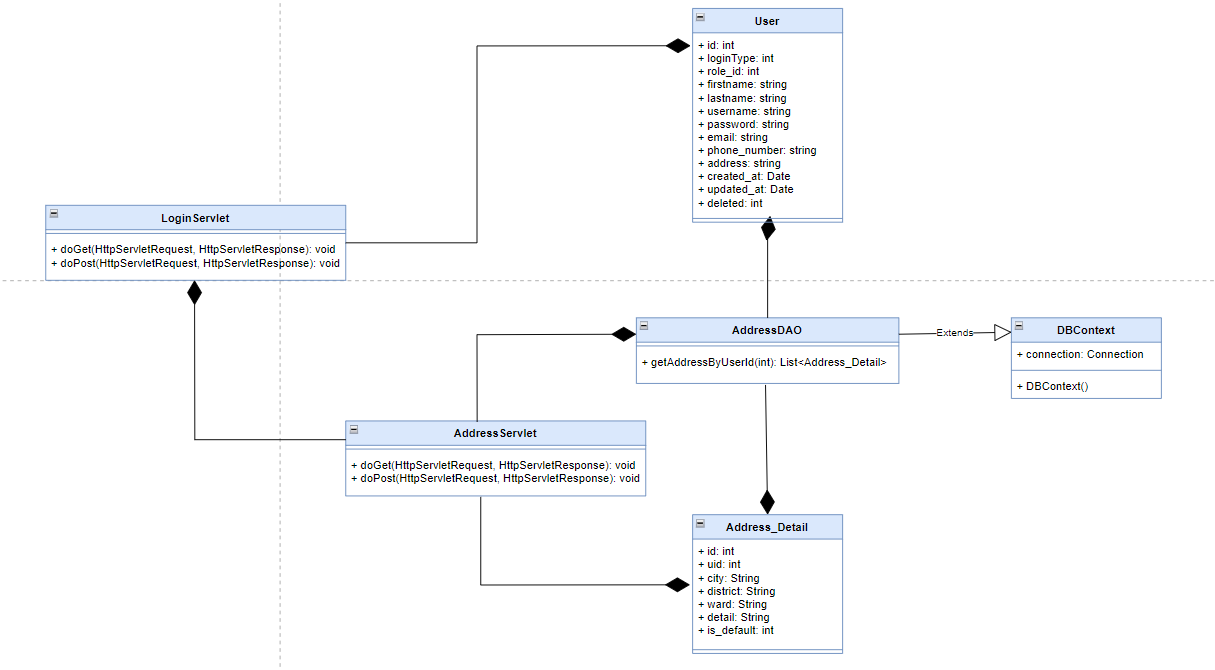
**d. Database Queries**

| **Class** | **Method** | **SQL** |
| --- | --- | --- |
| OrderDAO | +getOrderByUserId(int): List<Order> | select \* from Order\_Details where order\_id = ? |

**10. Address**

**10.1 View address**

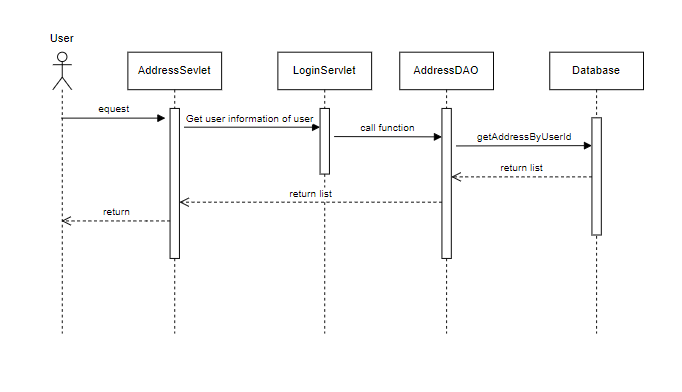
**a. Class Diagram**

****

**b. Class specifications**

| **Class** | **Method** | **Description** |
| --- | --- | --- |
| Address\_Detail | Get, set | Get, set attribute of class |
| User | Get, set | Get, set attribute of class |
| AddressDAO | getAddressByUserId(int): List<Address> | Get list of addresses of an user |
| DBContext | DBContext() | Connect to the database |
| ViewOwnOrder | doGet() | Get parameter from JSP, call function from AddressDAO, set attribute and send to JSP |

**c. Sequence Diagram**

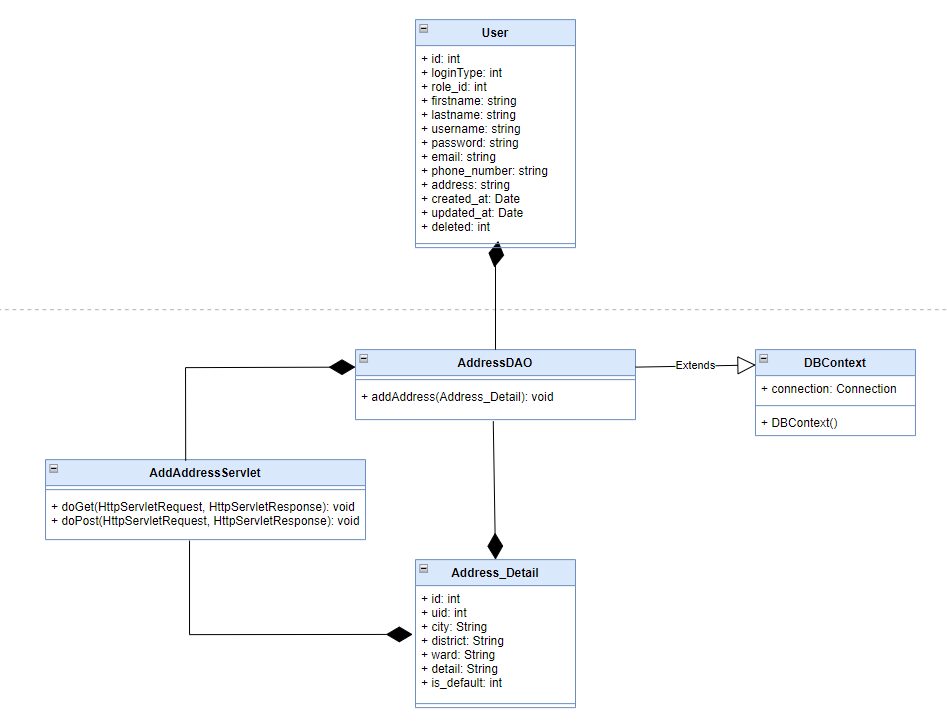
****

**d. Database Queries**

| **Class** | **Method** | **SQL** |
| --- | --- | --- |
| AddressDAO | +getAddressByUserId(int): List<Address> | select \* from Address\_Detail where uid=? |

**10.2 Add address**

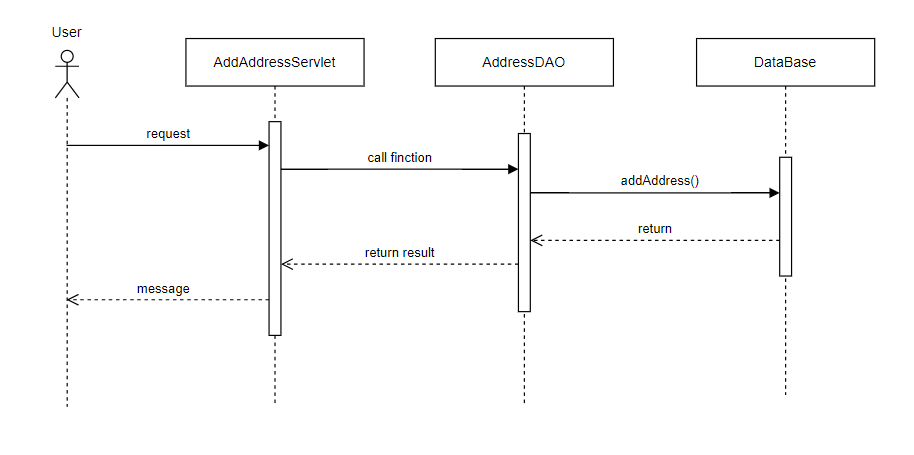
**a. Class Diagram**

****

**b. Class specifications**

| **Class** | **Method** | **Description** |
| --- | --- | --- |
| Address\_Detail | Get, set | Get, set attribute of class |
| User | Get, set | Get, set attribute of class |
| AddressDAO | addAddress(Address\_Detail): void | Get list of addresses of an user |
| DBContext | DBContext() | Connect to the database |
| AddAddressServlet | doGet() | Request to JSP to user input information |
| doPost() | Call function from AddressDAO and send message to JSP |

**c. Sequence Diagram**

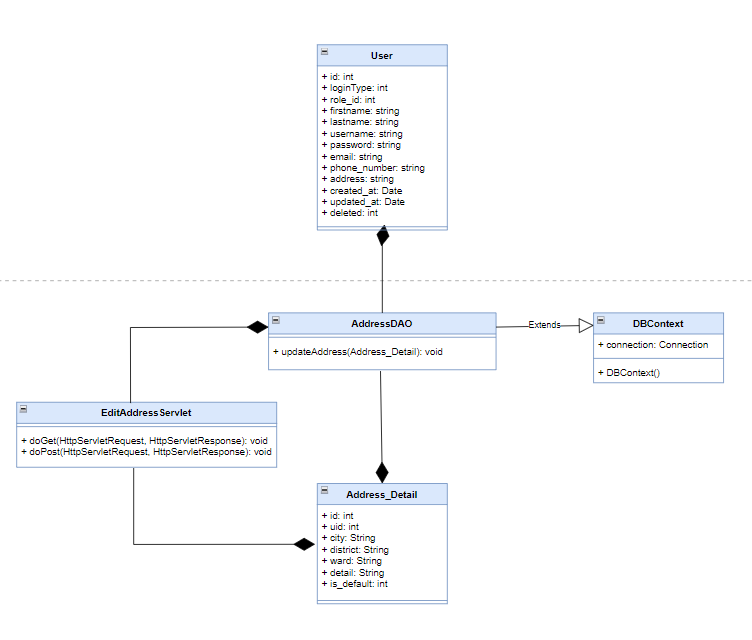
****

**d. Database Queries**

| **Class** | **Method** | **SQL** |
| --- | --- | --- |
| AddressDAO | +addAddress(Address\_Detail): void | INSERT INTO [dbo].[Address\_Detail]\n"  + " ([uid]\n"  + " ,[city]\n"  + " ,[district]\n"  + " ,[ward]\n"  + " ,[detail]\n"  + " ,[is\_default])\n"  + " VALUES\n"  + " (?, ?, ?, ?, ?, ?) |

**10.3 Edit address**

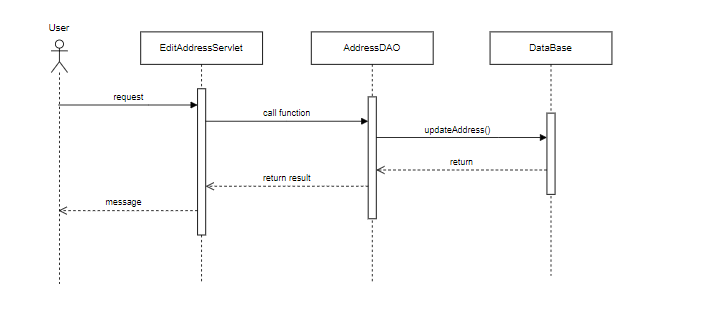
**a. Class Diagram**

****

**b. Class specifications**

| **Class** | **Method** | **Description** |
| --- | --- | --- |
| Address\_Detail | Get, set | Get, set attribute of class |
| User | Get, set | Get, set attribute of class |
| AddressDAO | updateAddress(Address\_Detail): void | Update an address |
| DBContext | DBContext() | Connect to the database |
| EditAddressServlet | doGet() | Request to JSP to user input information |
| doPost() | Call function from AddressDAO and send message to JSP |

**c. Sequence Diagram**

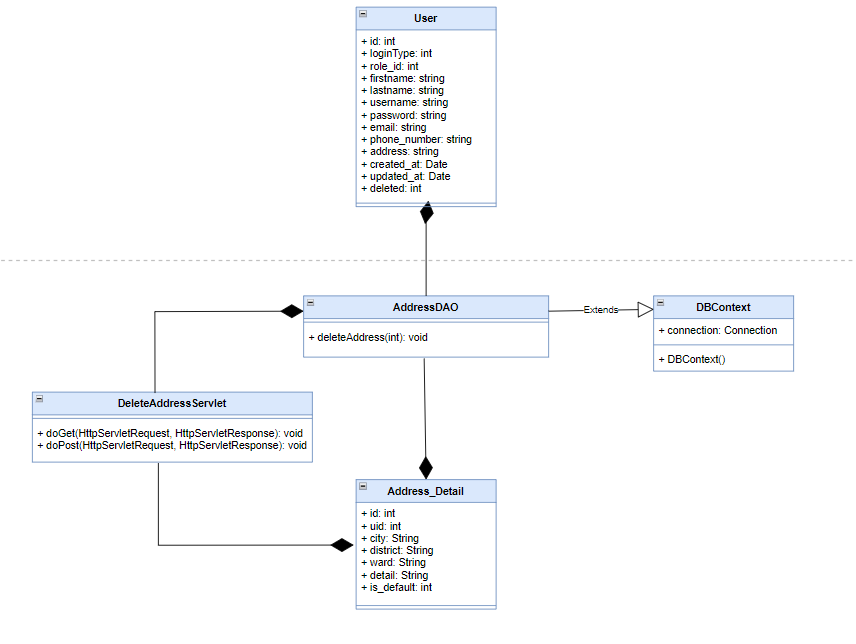
****

**d. Database Queries**

| **Class** | **Method** | **SQL** |
| --- | --- | --- |
| AddressDAO | +updateAddress(Address\_Detail): void | UPDATE [dbo].[Address\_Detail]\n"  + " SET [city] = ?\n"  + " ,[district] = ?\n"  + " ,[ward] = ?\n"  + " ,[detail] = ?\n"  + " ,[is\_default] = ?\n"  + " WHERE id = ? |

**10.4 Delete Address**

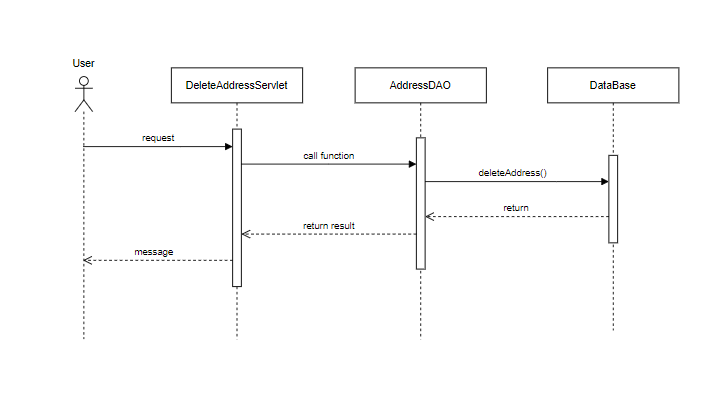
**a. Class Diagram**

****

**b. Class specifications**

| **Class** | **Method** | **Description** |
| --- | --- | --- |
| Address\_Detail | Get, set | Get, set attribute of class |
| User | Get, set | Get, set attribute of class |
| AddressDAO | deleteAddress(int): void | Delete address by id |
| DBContext | DBContext() | Connect to the database |
| DeleteAddressServlet | doGet() | Get information from JSP and call function from AddressDAO, then send redirect JSP with message |

**c. Sequence Diagram**

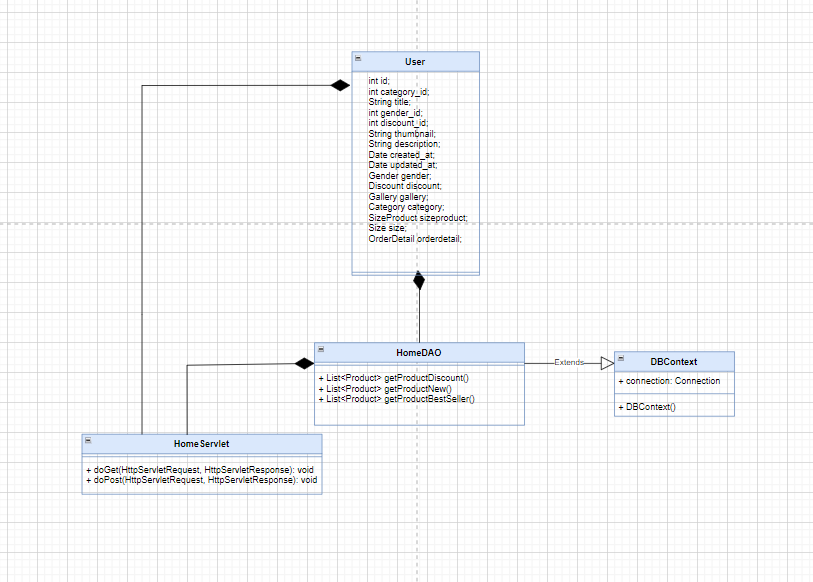
****

**d. Database Queries**

| **Class** | **Method** | **SQL** |
| --- | --- | --- |
| AddressDAO | +deleteAddress(int): void | delete from address\_detail where id=? |

**11. Home**

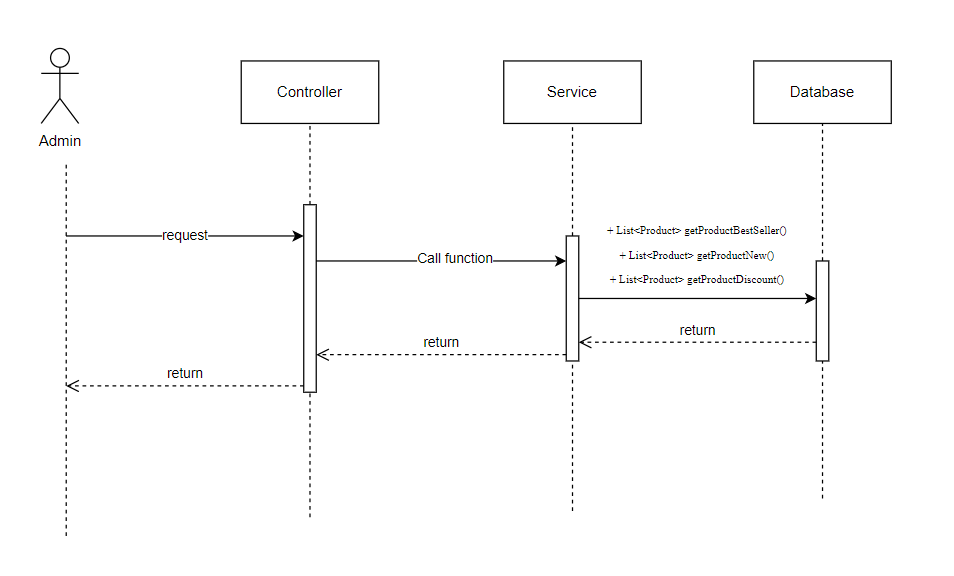
**a. Class Diagram**

****

**b. Class specifications**

| **Class** | **Method** | **Description** |
| --- | --- | --- |
| User | Get, set | Get, set attribute of class |
| HomeDAO | + List<Product> getProductDiscount()  + List<Product> getProductNew()  + List<Product> getProductBestSeller() | Get product discount  Get Product New  Get Product Best Seller |
| DBContext | DBContext() | Connect to the database |
| HomeServlet | doGet() | Request to JSP to user input information |
| doPost() | Call function from HomeDAO and send message to JSP |

**c. Sequence Diagram**

****

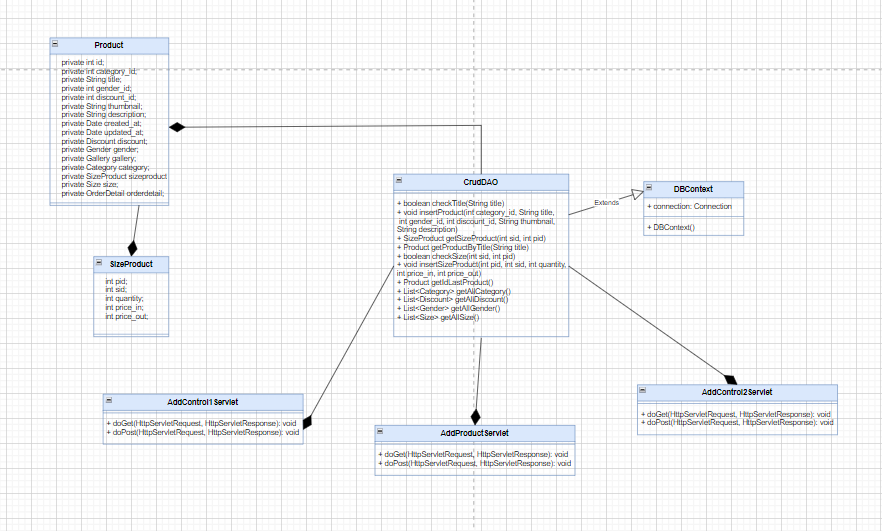
**d. Database Queries**

| **Class** | **Method** | **SQL** |
| --- | --- | --- |
| HomeDAO | + List<Product> getProductDiscount()  + List<Product> getProductNew()  + List<Product> getProductBestSeller() | select top 12 p.gender\_id, p.title, p.id, p.thumbnail, p.updated\_at, p.discount\_id, d.value, min(sp.price\_out) as price, min(sp.sid) as sid from Products p\n"  + "join SizeProduct sp on p.id = sp.pid\n"  + "join Sizes s on s.id = sp.sid\n"  + "join Discounts d on d.id = p.discount\_id\n"  + "group by p.title, p.id, p.thumbnail, p.updated\_at, p.gender\_id, p.discount\_id, d.value\n"  + "order by p.updated\_at desc  select top 4 p.gender\_id, p.title, p.id, p.thumbnail, min(sp.price\_out) as price, p.discount\_id, d.value , min(sp.sid) as sid from Products p\n"  + "join SizeProduct sp on p.id = sp.pid\n"  + "join Sizes s on s.id = sp.sid\n"  + "join Discounts d on p.discount\_id = d.id\n"  + "where p.discount\_id != 0\n"  + "group by p.title, p.id, p.thumbnail, d.value, p.gender\_id, p.discount\_id\n"  + "order by p.discount\_id desc  select top 6 p.gender\_id, p.title, p.id, p.thumbnail, od.num, p.discount\_id, d.value, min(sp.price\_out) as price, min(sp.sid) as sid from Products p\n"  + "join SizeProduct sp on p.id = sp.pid\n"  + "join Sizes s on s.id = sp.sid\n"  + "join Order\_Details od on od.product\_id = p.id\n"  + "join Discounts d on d.id = p.discount\_id\n"  + "group by p.title, p.id, p.thumbnail, od.num , p.discount\_id, p.gender\_id, d.value\n"  + "order by num desc |

**12. Manage Product**

**12.1 Add Product**

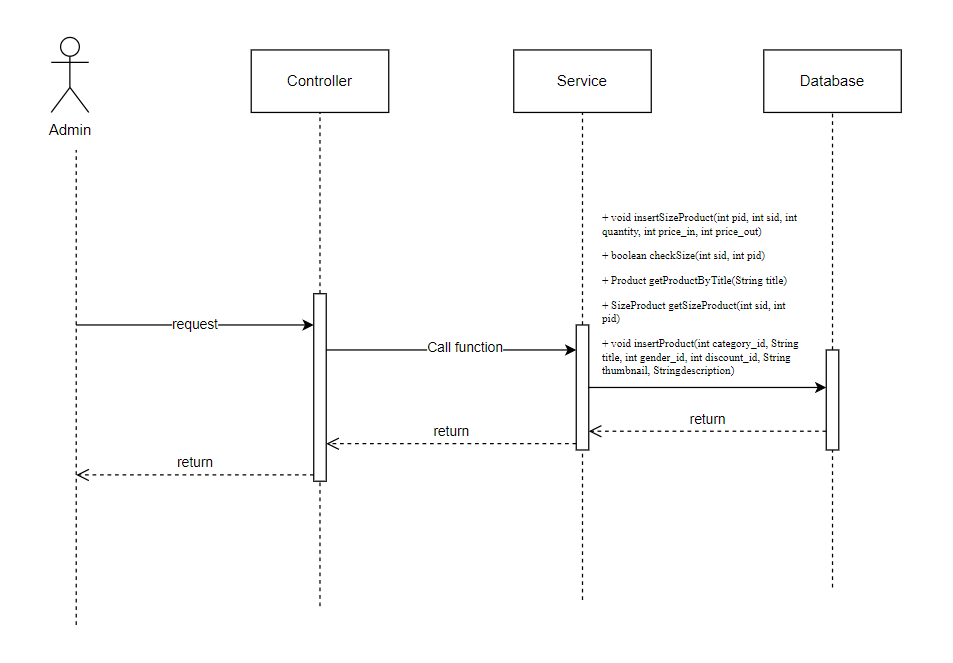
**a. Class Diagram**

****

**b. Class specifications**

| **Class** | **Method** | **Description** |
| --- | --- | --- |
| User | Get, set | Get, set attribute of class |
| SizeProduct | Get, set | Get, set attribute of class |
| CrudDAO | + boolean checkTitle(String title)  + void insertProduct(int category\_id, String title,  int gender\_id, int discount\_id, String thumbnail,  String description)  + SizeProduct getSizeProduct(int sid, int pid)  + Product getProductByTitle(String title)  + boolean checkSize(int sid, int pid)  + void insertSizeProduct(int pid, int sid, int quantity,  int price\_in, int price\_out)  + Product getIdLastProduct()  + List<Category> getAllCategory()  + List<Discount> getAllDiscount()  + List<Gender> getAllGender()  + List<Size> getAllSize() | Check title exist  Insert Product  Get size product by size id and product id  Get product by title  Check size exist  Insert size product  Get product id last  Get all category  Get all discount  Get all gender  Get all size |
| DBContext | DBContext() | Connect to the database |
| AddProductServlet | doGet() | Request to JSP to user input information |
| doPost() | Call function from CrudDAO and send message to JSP |
| AddControl1Servlet | doGet() | Request to JSP to user input information |
| doPost() | Call function from CrudDAO and send message to JSP |
| AddControl2Servlet | doGet() | Request to JSP to user input information |
| doPost() | Call function from CrudDAO and send message to JSP |

**c. Sequence Diagram**

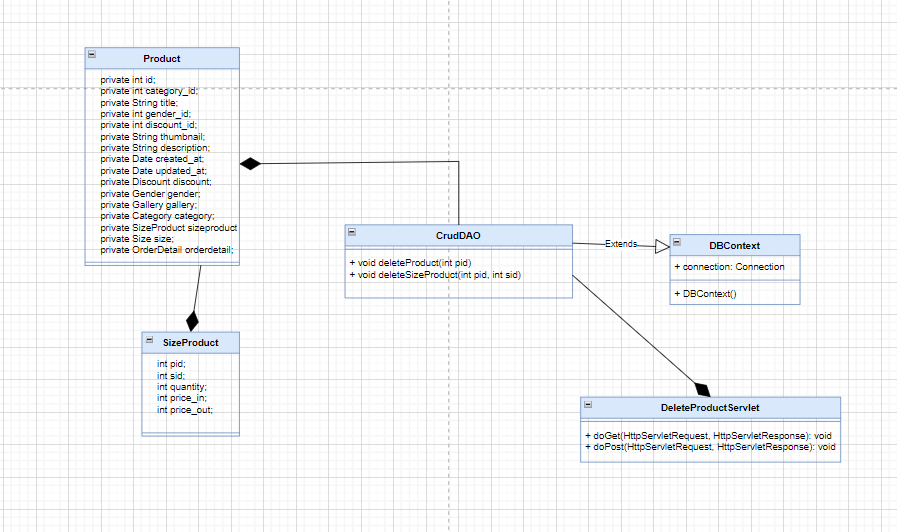
****

**d. Database Queries**

| **Class** | **Method** | **SQL** |
| --- | --- | --- |
| CrudDAO | + boolean checkTitle(String title)  + void insertProduct(int category\_id, String title,  int gender\_id, int discount\_id, String thumbnail,  String description)  + SizeProduct getSizeProduct(int sid, int pid)  + Product getProductByTitle(String title)  + boolean checkSize(int sid, int pid)  + void addQuantity(int sid, int pid, int quantity)  + void insertSizeProduct(int pid, int sid, int quantity,  int price\_in, int price\_out)  + Product getIdLastProduct()  + List<Category> getAllCategory()  + List<Discount> getAllDiscount()  + List<Gender> getAllGender()  + List<Size> getAllSize() | select title from products  INSERT INTO [dbo].[Products]\n"  + " ([category\_id]\n"  + " ,[title]\n"  + " ,[gender\_id]\n"  + " ,[discount\_id]\n"  + " ,[thumbnail]\n"  + " ,[description]\n"  + " ,[created\_at]\n"  + " ,[updated\_at])\n"  + " VALUES\n"  + " (?\n"  + " ,?\n"  + " ,?\n"  + " ,?\n"  + " ,?\n"  + " ,?\n"  + " ,getdate()\n"  + " ,getdate())  SELECT [pid]\n"  + " ,[sid]\n"  + " ,[quantity]\n"  + " ,[price\_in]\n"  + " ,[price\_out]\n"  + " FROM [dbo].[SizeProduct]\n"  + " where sid = ? and pid = ?  select \* from products where title = ?  select sid from SizeProduct where pid = ?  UPDATE [dbo].[SizeProduct]\n"  + " SET \n"  + " [quantity] = ?\n"  + " \n"  + " WHERE sid = ? and pid = ?  INSERT INTO [dbo].[SizeProduct]\n"  + " ([pid]\n"  + " ,[sid]\n"  + " ,[quantity]\n"  + " ,[price\_in]\n"  + " ,[price\_out])\n"  + " VALUES\n"  + " (?\n"  + " ,?\n"  + " ,?\n"  + " ,?\n"  + " ,?)  SELECT top 1 id\n"  + "\n"  + " FROM [dbo].[Products]\n"  + "\n"  + " order by id desc  select \* from Categories  select \* from Discounts  select \* from Genders  select \* from Sizes |

**12.2 Delete Product**

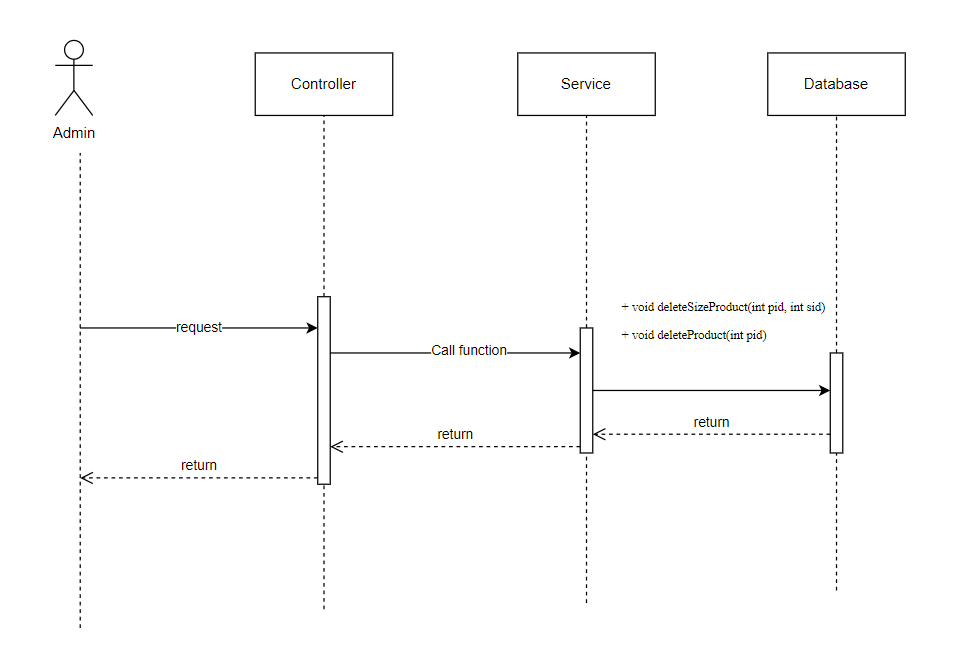
**a. Class Diagram**

****

**b. Class specifications**

| **Class** | **Method** | **Description** |
| --- | --- | --- |
| User | Get, set | Get, set attribute of class |
| SizeProduct | Get, set | Get, set attribute of class |
| CrudDAO | + void deleteProduct(int pid)  + void deleteSizeProduct(int pid, int sid) | Delete Product |
| DBContext | DBContext() | Connect to the database |
| DeleteProductServlet | doGet() | Request to JSP to user input information |
| doPost() | Call function from CrudDAO and send message to JSP |

**c. Sequence Diagram**

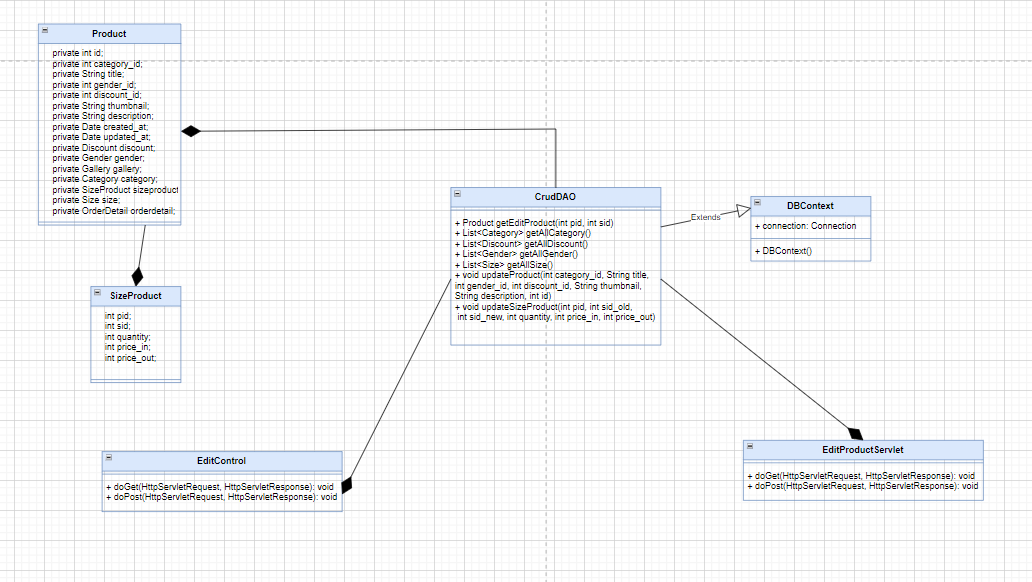
****

**d. Database Queries**

| **Class** | **Method** | **SQL** |
| --- | --- | --- |
| CrudDAO | + void deleteProduct(int pid)  + void deleteSizeProduct(int pid, int sid) | DELETE FROM [dbo].[Products]\n"  + " WHERE id = ?  DELETE FROM [dbo].[SizeProduct]\n"  + " WHERE pid = ? and sid = ? |

**12.3 Edit Product**

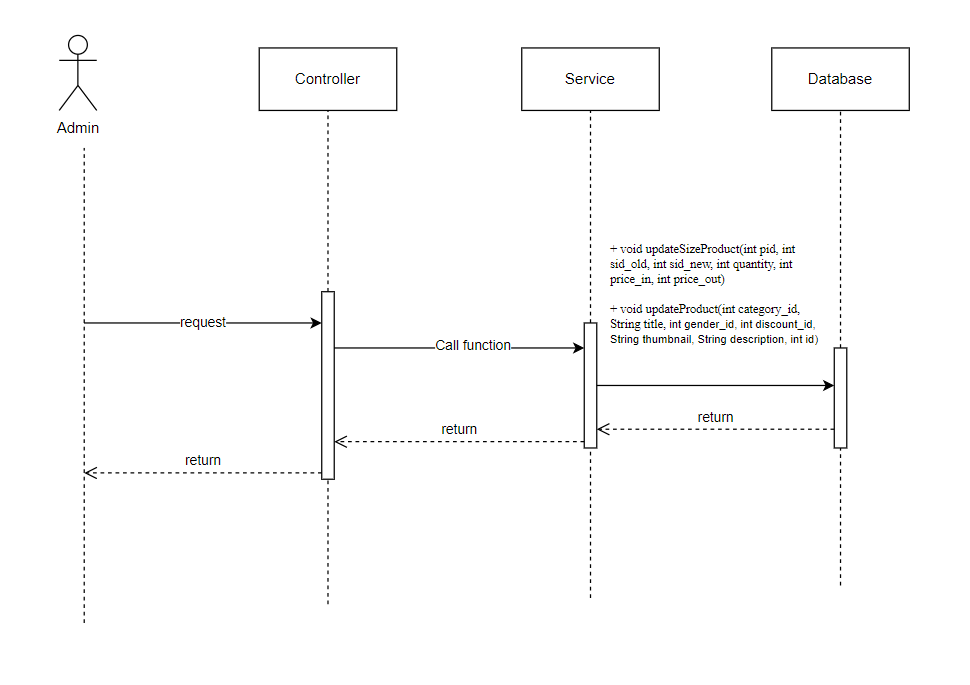
**a. Class Diagram**

****

**b. Class specifications**

| **Class** | **Method** | **Description** |
| --- | --- | --- |
| User | Get, set | Get, set attribute of class |
| SizeProduct | Get, set | Get, set attribute of class |
| CrudDAO | + Product getEditProduct(int pid, int sid)  + List<Category> getAllCategory()  + List<Discount> getAllDiscount()  + List<Gender> getAllGender()  + List<Size> getAllSize()  + void updateProduct(int category\_id, String title,  int gender\_id, int discount\_id, String thumbnail,  String description, int id)  + void updateSizeProduct(int pid, int sid\_old,  int sid\_new, int quantity, int price\_in, int price\_out) | Get product to edit  Get all category  Get all discount  Get all gender  Get all size  Edit Product |
| DBContext | DBContext() | Connect to the database |
| EditProductServlet | doGet() | Request to JSP to user input information |
| doPost() | Call function from CrudDAO and send message to JSP |
| EditControl | doGet() | Request to JSP to user input information |
| doPost() | Call function from CrudDAO and send message to JSP |

**c. Sequence Diagram**

****

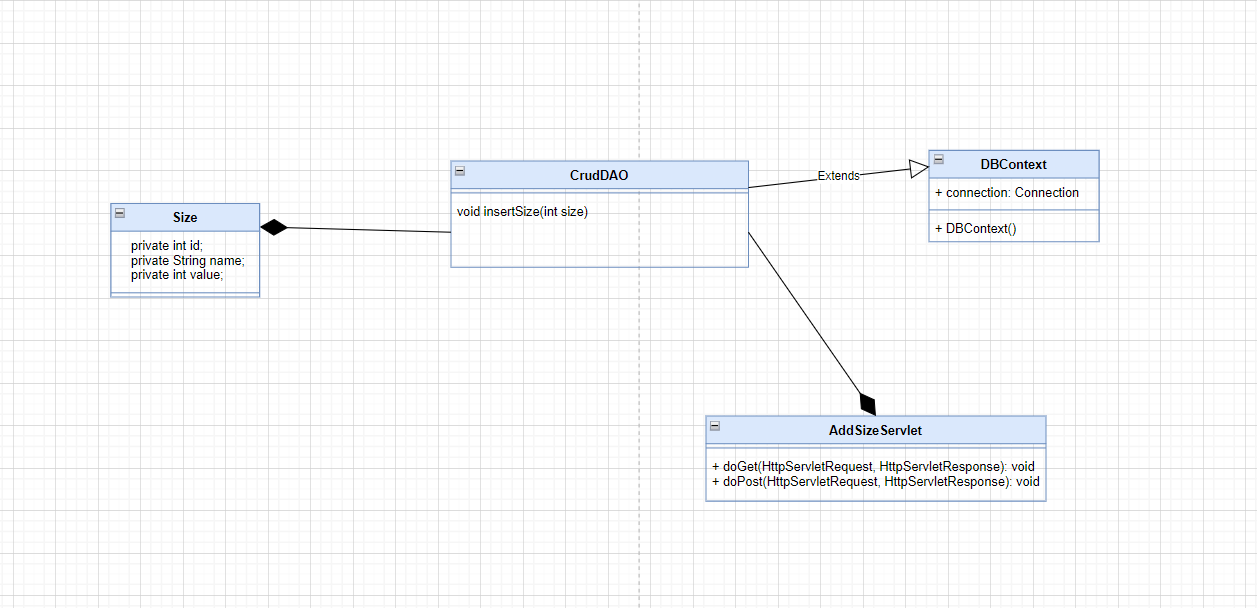
**d. Database Queries**

| **Class** | **Method** | **SQL** |
| --- | --- | --- |
| CrudDAO | + Product getEditProduct(int pid, int sid)  + List<Category> getAllCategory()  + List<Discount> getAllDiscount()  + List<Gender> getAllGender()  + List<Size> getAllSize()  + void updateProduct(int category\_id, String title,  int gender\_id, int discount\_id, String thumbnail,  String description, int id)  + void updateSizeProduct(int pid, int sid\_old,  int sid\_new, int quantity, int price\_in, int price\_out) | select \* from products p join SizeProduct sp \n"  + " on p.id = sp.pid \n"  + " where sid = ? and pid = ?  select \* from Categories  select \* from Discounts  select \* from Genders  select \* from Sizes  UPDATE [dbo].[Products]\n"  + " SET [category\_id] = ?\n"  + " ,[title] = ?\n"  + " ,[gender\_id] = ?\n"  + " ,[discount\_id] = ?\n"  + " ,[thumbnail] = ?\n"  + " ,[updated\_at] = getdate()"  + " ,[description] = ?\n"  + " WHERE id = ?  UPDATE [dbo].[SizeProduct]\n"  + " SET [quantity] = ?\n"  + " ,[sid] = ?\n"  + " ,[price\_in] = ?\n"  + " ,[price\_out] = ?\n"  + " WHERE pid = ? and sid = ? |

**13. Manage Size**

**13.1 Add Size**

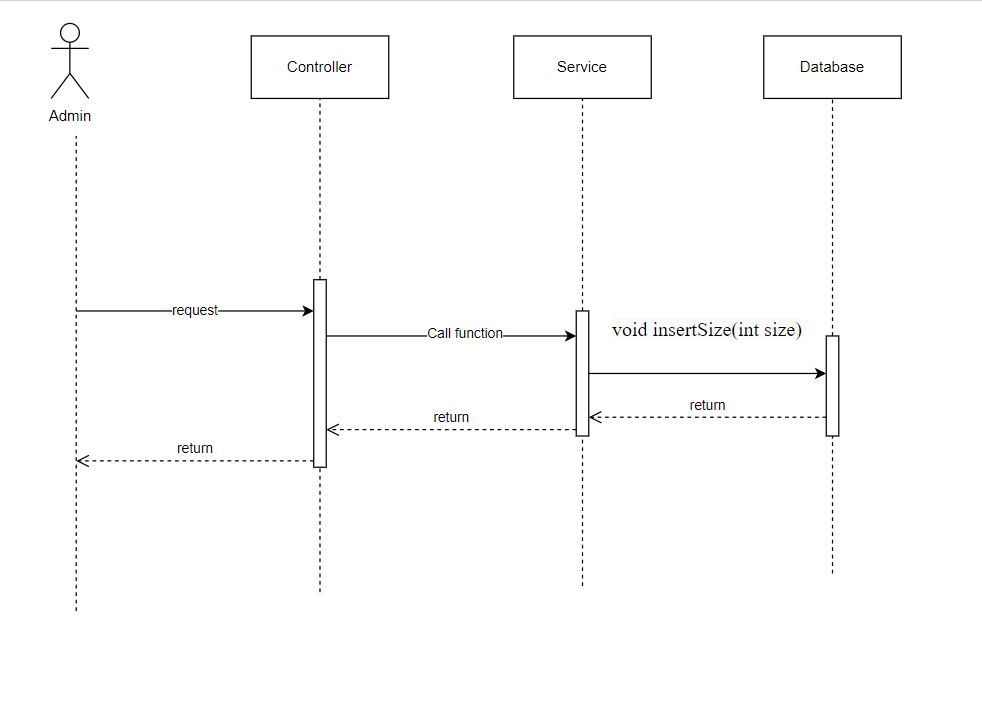
**a. Class Diagram**

****

**b. Class specifications**

| **Class** | **Method** | **Description** |
| --- | --- | --- |
| Size | Get, set | Get, set attribute of class |
| CrudDAO | void insertSize(int size) | Insert size |
| DBContext | DBContext() | Connect to the database |
| AddSizeServlet | doGet() | Request to JSP to user input information |
| doPost() | Call function from CrudDAO and send message to JSP |

**c. Sequence Diagram**

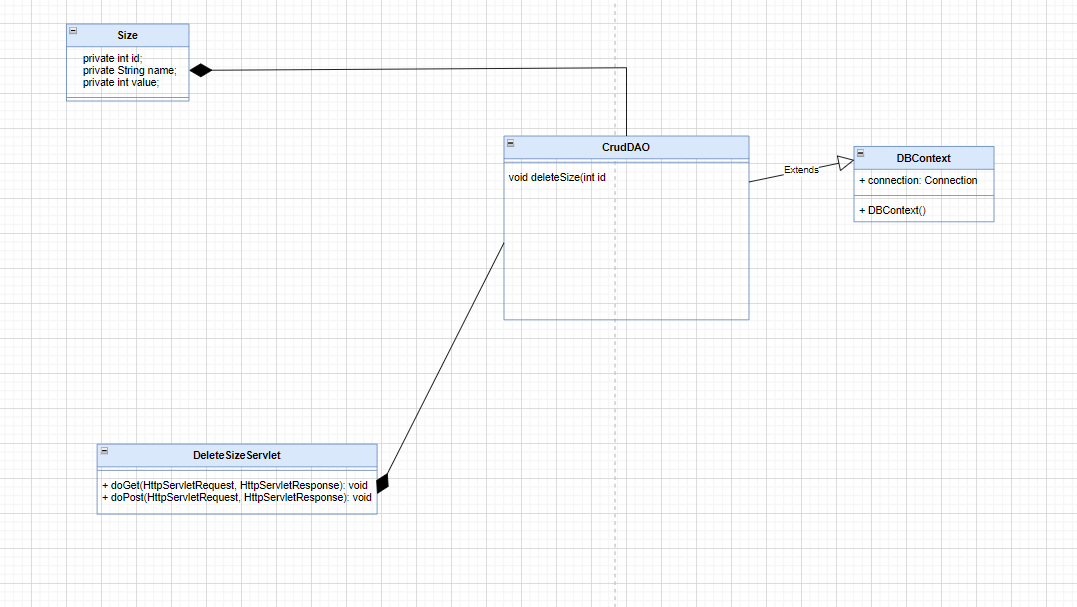
****

**d. Database Queries**

| **Class** | **Method** | **SQL** |
| --- | --- | --- |
| CrudDAO | void insertSize(int size) | INSERT INTO [dbo].[Sizes]\n"  + " ([name]\n"  + " ,[value])\n"  + " VALUES\n"  + " (?\n"  + " ,?) |

**13.2 Delete Size**

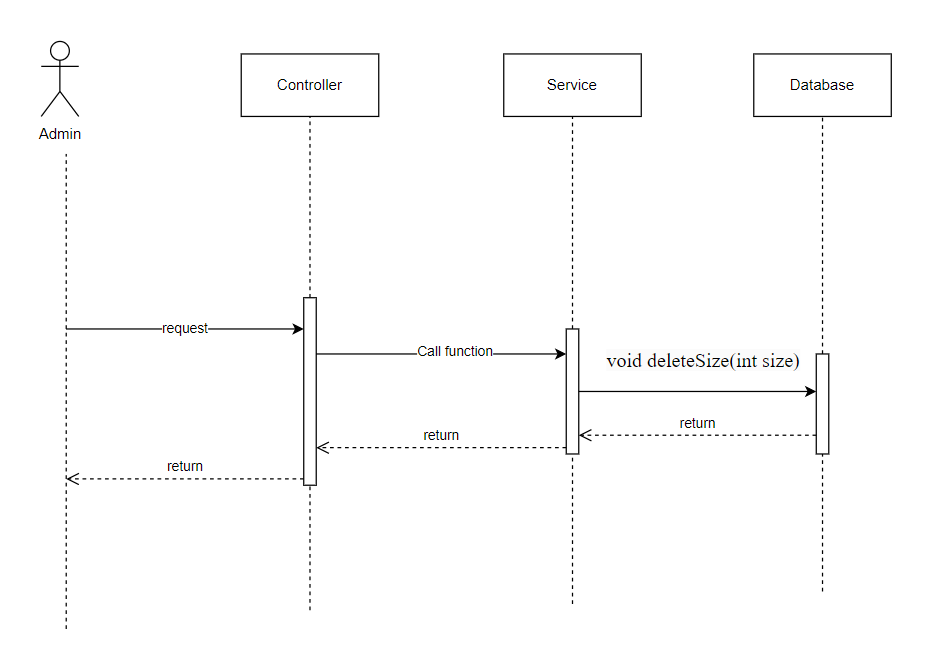
**a. Class Diagram**

****

**b. Class specifications**

| **Class** | **Method** | **Description** |
| --- | --- | --- |
| Size | Get, set | Get, set attribute of class |
| CrudDAO | void deleteSize(int size) | Delete size |
| DBContext | DBContext() | Connect to the database |
| AddSizeServlet | doGet() | Request to JSP to user input information |
| doPost() | Call function from CrudDAO and send message to JSP |

**c. Sequence Diagram**

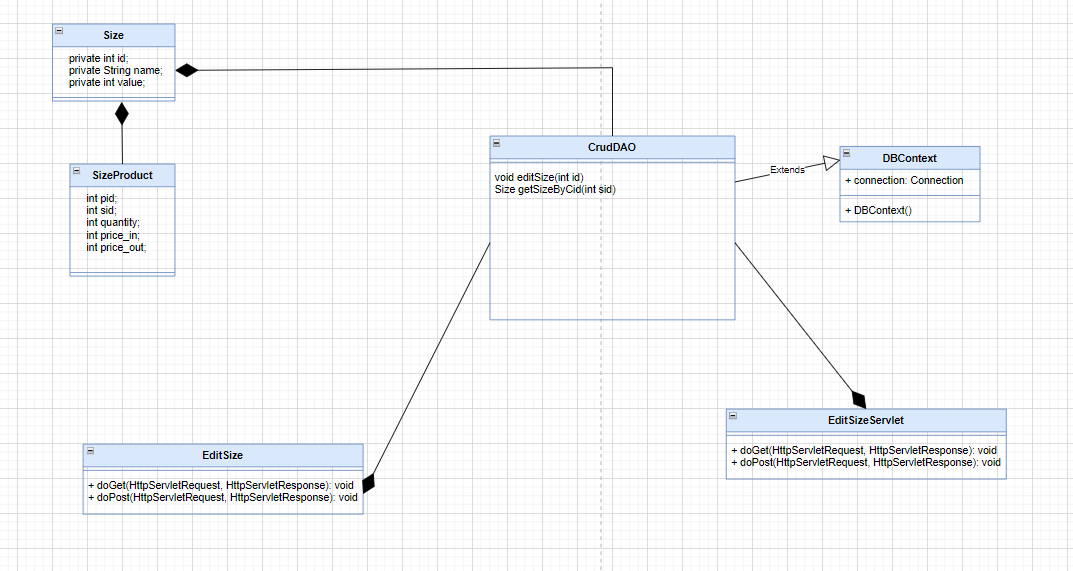
****

**d. Database Queries**

| **Class** | **Method** | **SQL** |
| --- | --- | --- |
| CrudDAO | void deleteSize(int size) | DELETE FROM [dbo].[Sizes]\n"  + " WHERE id = ? |

**13.3 Edit Size**

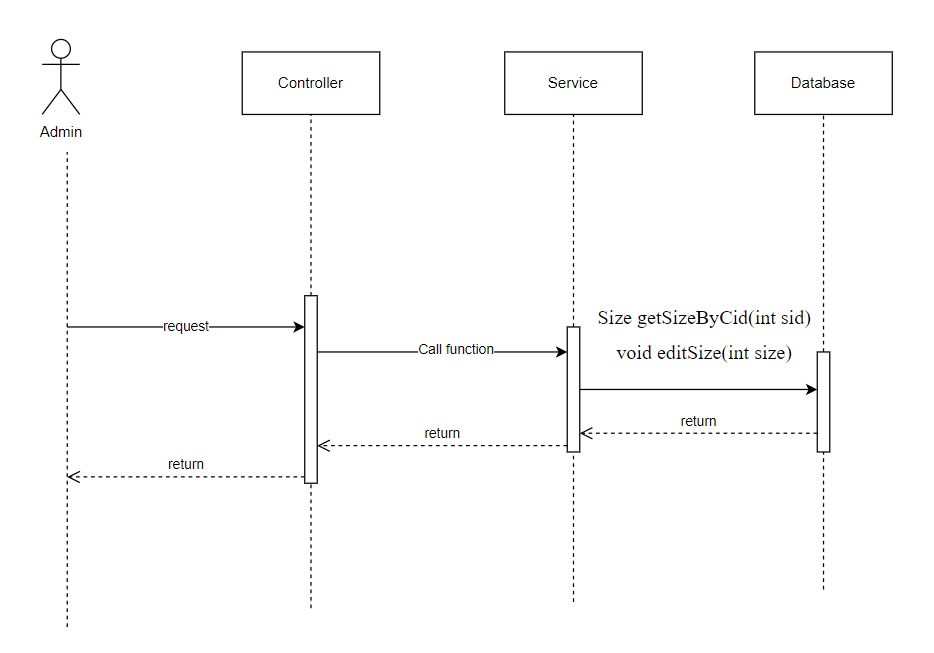
**a. Class Diagram**

****

**b. Class specifications**

| **Class** | **Method** | **Description** |
| --- | --- | --- |
| Size | Get, set | Get, set attribute of class |
| CrudDAO | void editSize(int size)  Size getSizeByCid(int sid) | Delete size |
| DBContext | DBContext() | Connect to the database |
| AddSizeServlet | doGet() | Request to JSP to user input information |
| doPost() | Call function from CrudDAO and send message to JSP |

**c. Sequence Diagram**

****

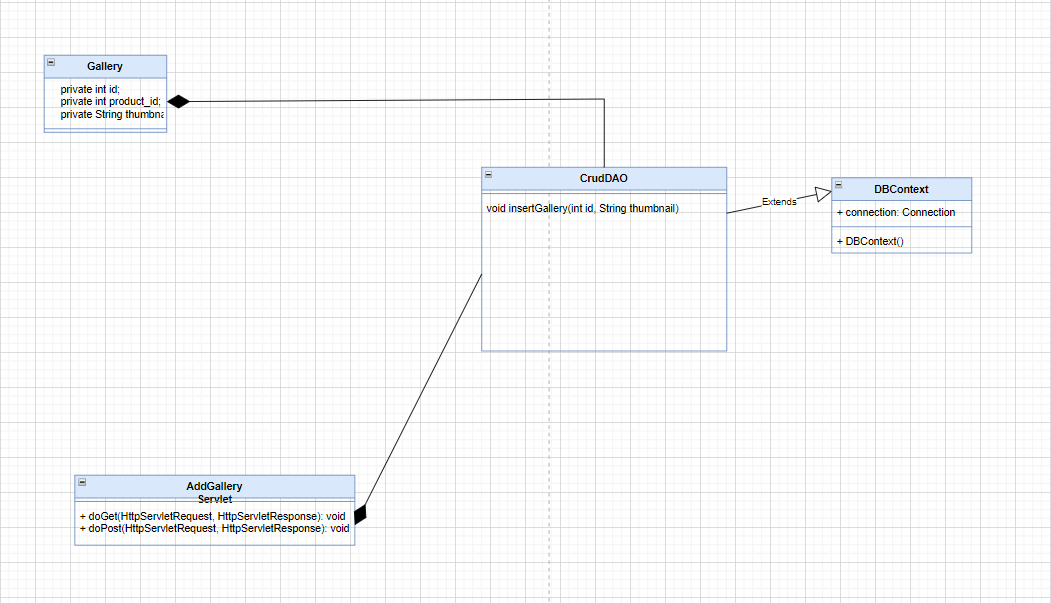
**d. Database Queries**

| **Class** | **Method** | **SQL** |
| --- | --- | --- |
| CrudDAO | Size getSizeByCid(int sid)  void editSize(int id, int size) | SELECT [id]\n"  + " ,[name]\n"  + " ,[value]\n"  + " FROM [dbo].[Sizes]"  + " where id = ?  UPDATE [dbo].[Sizes]\n"  + " SET [name] = ? \n"  + " ,[value] = ? \n"  + " WHERE id = ? |

**14. Manage Gallery**

**14.1 Add Gallery**

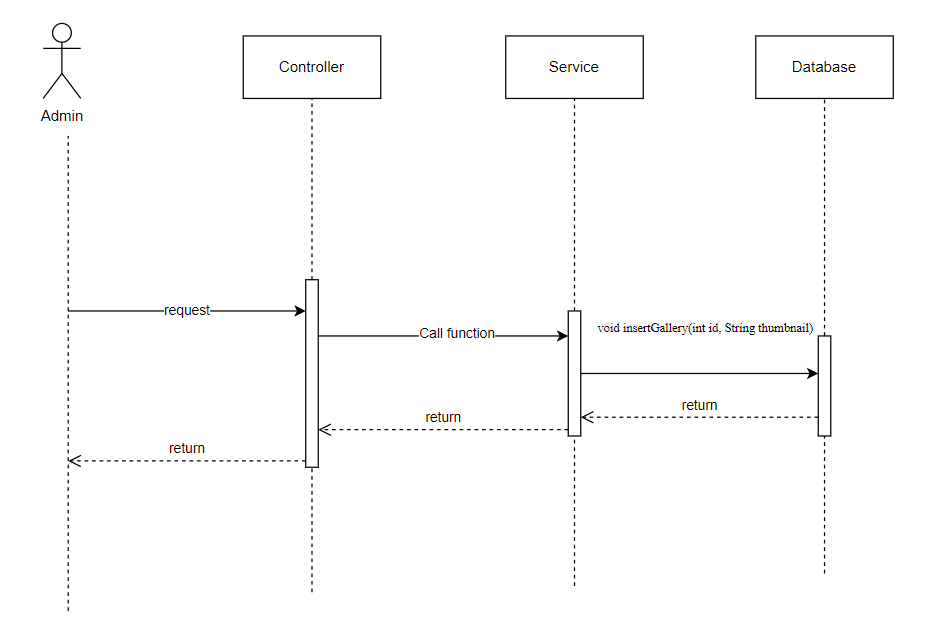
**a. Class Diagram**

****

**b. Class specifications**

| **Class** | **Method** | **Description** |
| --- | --- | --- |
| Gallery | Get, set | Get, set attribute of class |
| CrudDAO | void insertGallery(int id, String thumbnail) | Insert gallery |
| DBContext | DBContext() | Connect to the database |
| AddGalleryServlet | doGet() | Request to JSP to user input information |
| doPost() | Call function from CrudDAO and send message to JSP |

**c. Sequence Diagram**

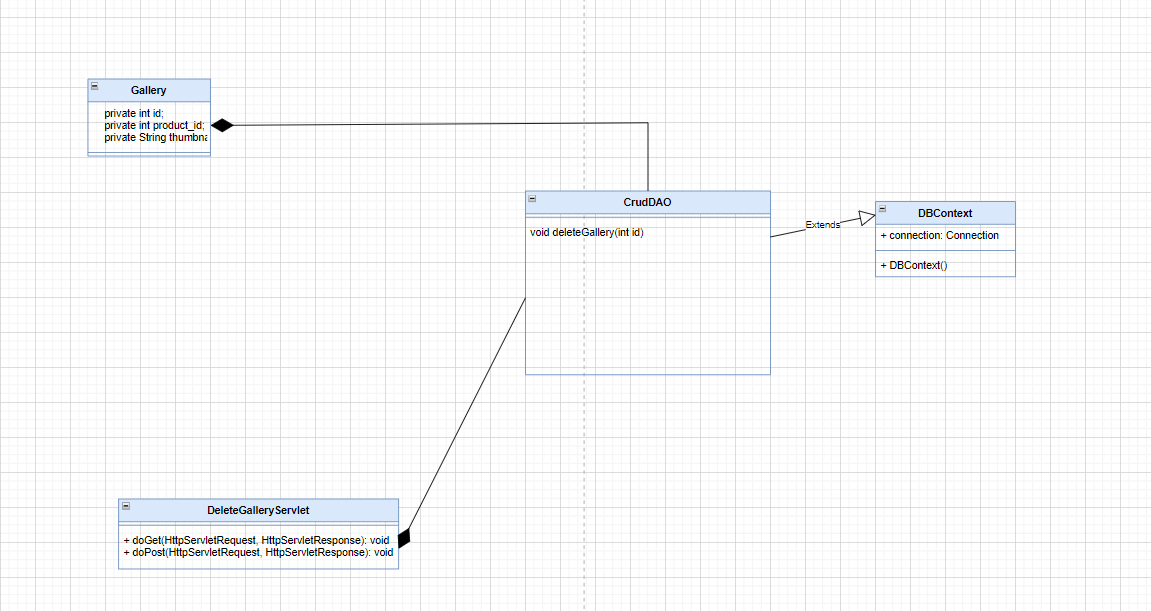
****

**d. Database Queries**

| **Class** | **Method** | **SQL** |
| --- | --- | --- |
| CrudDAO | void insertGallery(int id, String thumbnail) | INSERT INTO [dbo].[Galeries]\n"  + " ([product\_id]\n"  + " ,[thumbnail])\n"  + " VALUES\n"  + " (?\n"  + " ,?) |

**14.2 Delete Gallery**

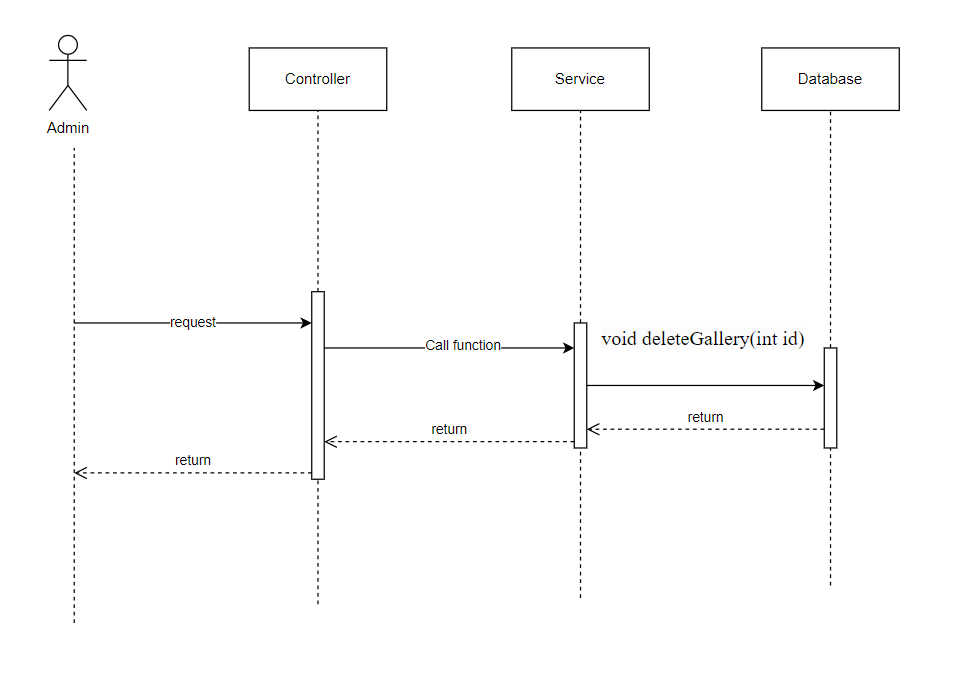
**a. Class Diagram**

****

**b. Class specifications**

| **Class** | **Method** | **Description** |
| --- | --- | --- |
| Gallery | Get, set | Get, set attribute of class |
| CrudDAO | void deleteGallery(int id) | Delete image |
| DBContext | DBContext() | Connect to the database |
| DeleteGalleryServlet | doGet() | Request to JSP to user input information |
| doPost() | Call function from CrudDAO and send message to JSP |

**c. Sequence Diagram**

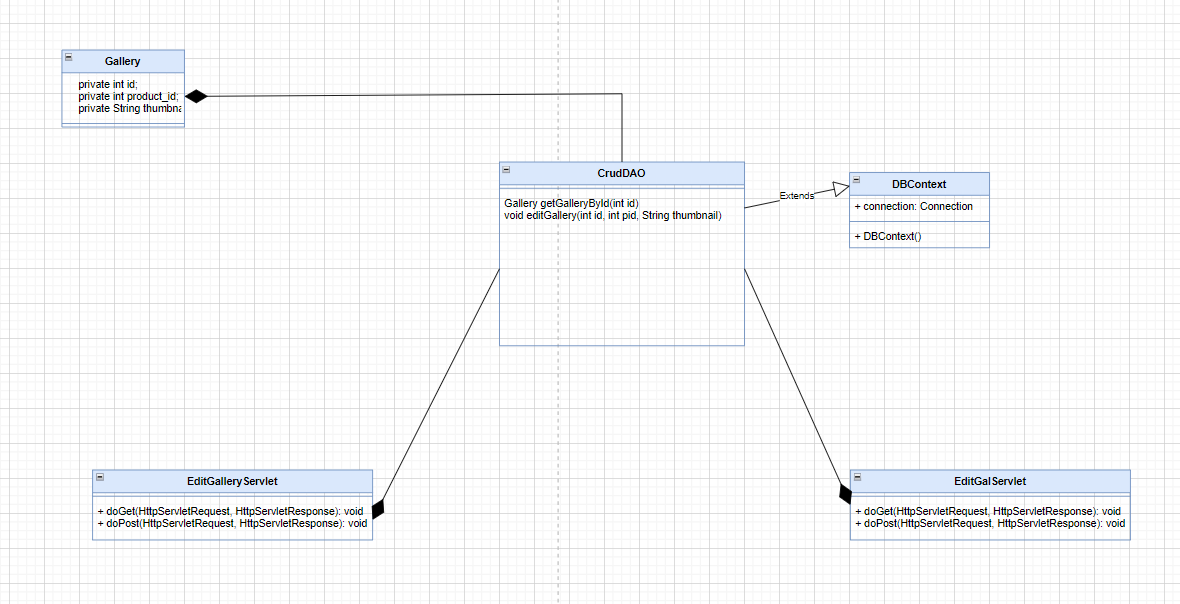
****

**d. Database Queries**

| **Class** | **Method** | **SQL** |
| --- | --- | --- |
| CrudDAO | void deleteGallery(int id) | DELETE FROM [dbo].[Galeries]\n"  + " WHERE id = ? |

**14.3 Edit Gallery**

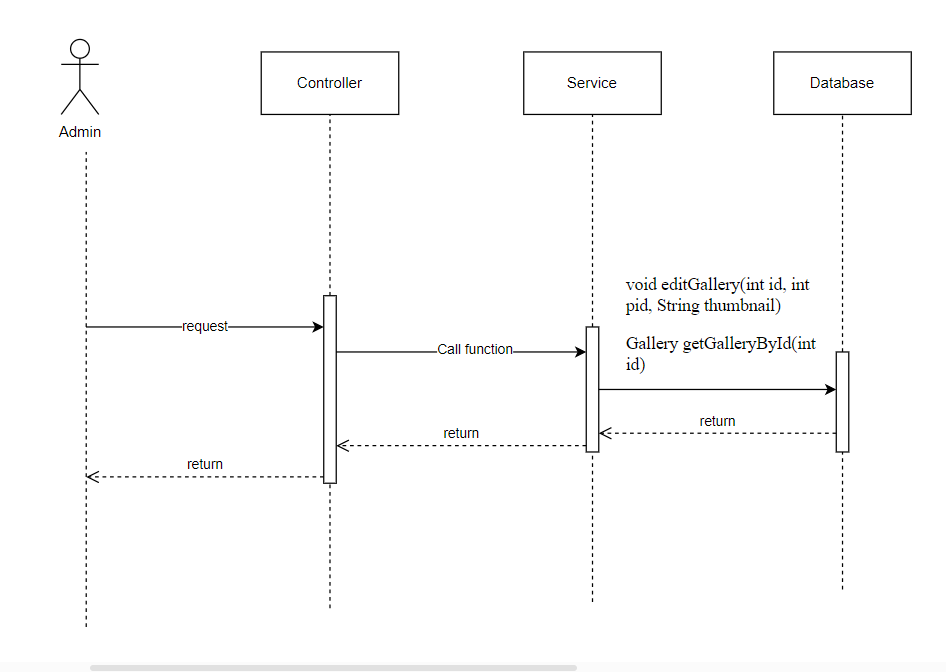
**a. Class Diagram**

****

**b. Class specifications**

| **Class** | **Method** | **Description** |
| --- | --- | --- |
| Gallery | Get, set | Get, set attribute of class |
| CrudDAO | Gallery getGalleryById(int id)  void editGallery(int id, int pid, String thumbnail) | Edit Gallery |
| DBContext | DBContext() | Connect to the database |
| EditGalleryServlet | doGet() | Request to JSP to user input information |
| doPost() | Call function from CrudDAO and send message to JSP |
| EditGalServlet | doGet() | Request to JSP to user input information |
| doPost() | Call function from CrudDAO and send message to JSP |

**c. Sequence Diagram**

****

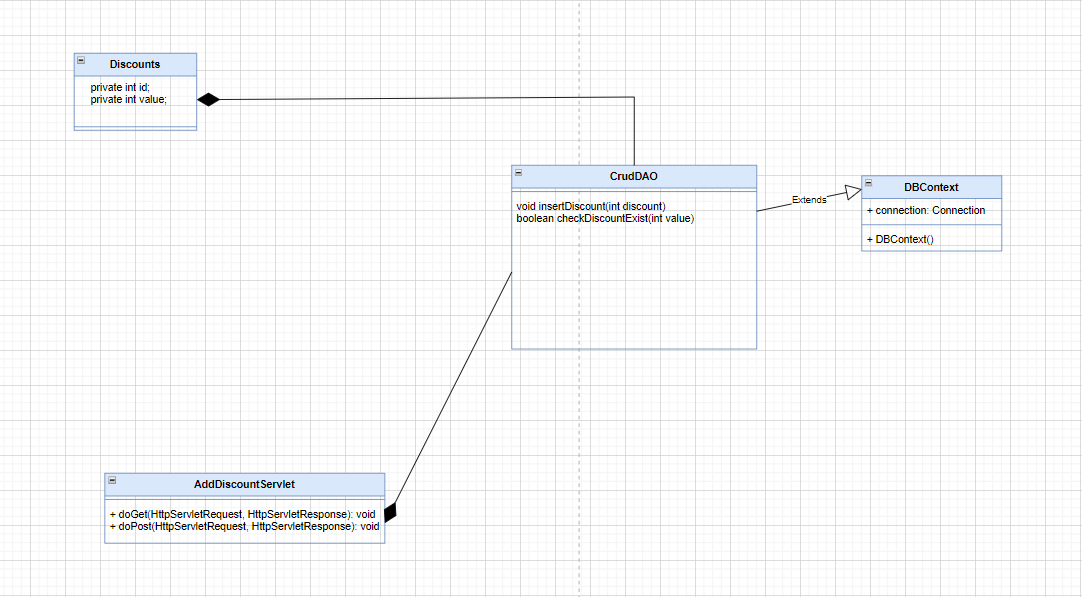
**d. Database Queries**

| **Class** | **Method** | **SQL** |
| --- | --- | --- |
| CrudDAO | Gallery getGalleryById(int id)  void editGallery(int id, int pid, String thumbnail) | SELECT [id]\n"  + " ,[product\_id]\n"  + " ,[thumbnail]\n"  + " FROM [dbo].[Galeries]\n"  + " where id = ?  UPDATE [dbo].[Galeries]\n"  + " SET [product\_id] = ?\n"  + " ,[thumbnail] = ?\n"  + " WHERE id = ? |

**15. Manage Discount**

**15.1 Add Discount**

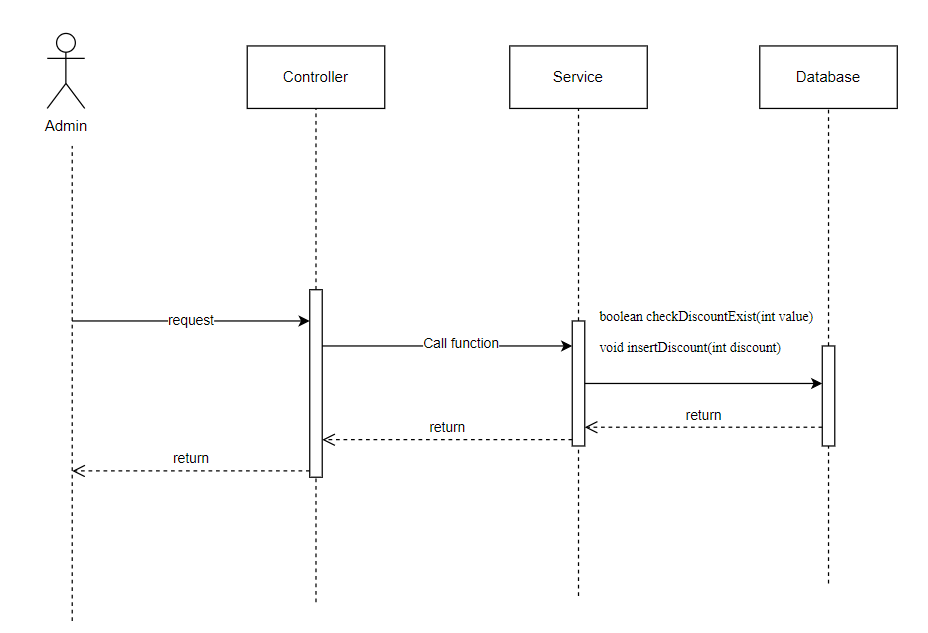
**a. Class Diagram**

****

**b. Class specifications**

| **Class** | **Method** | **Description** |
| --- | --- | --- |
| Discount | Get, set | Get, set attribute of class |
| CrudDAO | void insertDiscount(int discount)  boolean checkDiscountExist(int value) | Insert Discount |
| DBContext | DBContext() | Connect to the database |
| AddDiscountServlet | doGet() | Request to JSP to user input information |
| doPost() | Call function from CrudDAO and send message to JSP |

**c. Sequence Diagram**

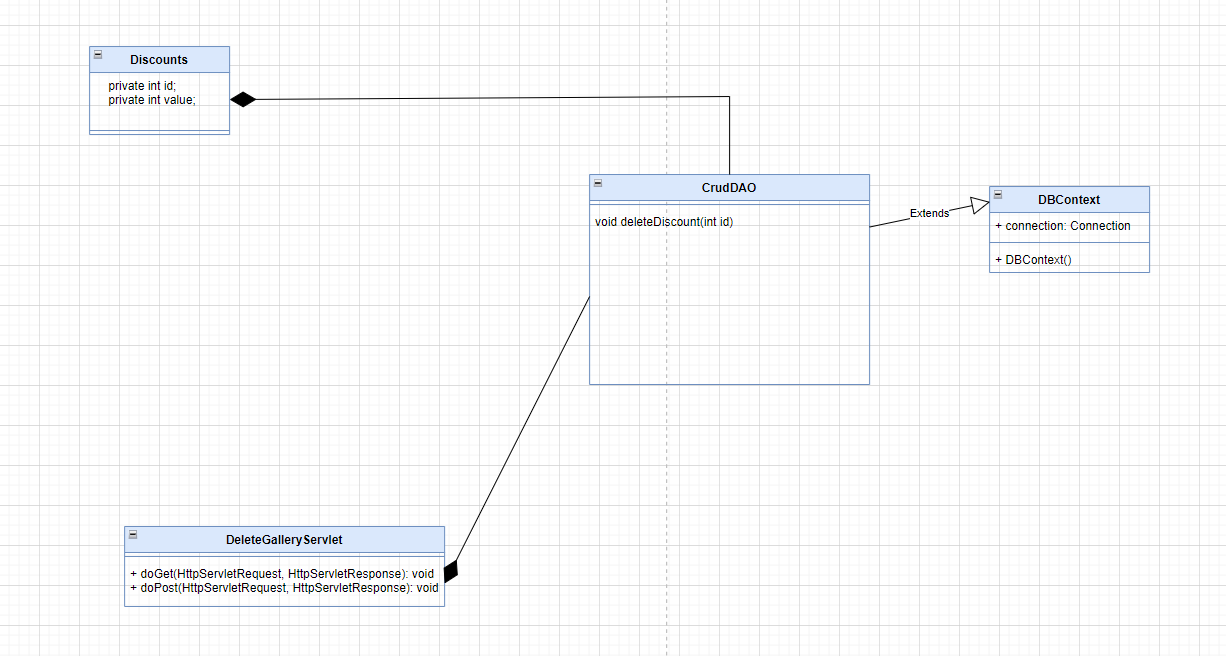
****

**d. Database Queries**

| **Class** | **Method** | **SQL** |
| --- | --- | --- |
| CrudDAO | void insertDiscount(int discount)  boolean checkDiscountExist(int value) | INSERT INTO [dbo].[Discounts]\n"  + " ([value])\n"  + " VALUES\n"  + " (?)  SELECT [id]\n"  + " ,[value]\n"  + " FROM [dbo].[Discounts]\n"  + " where value = ? |

**15.2 Delete Discount**

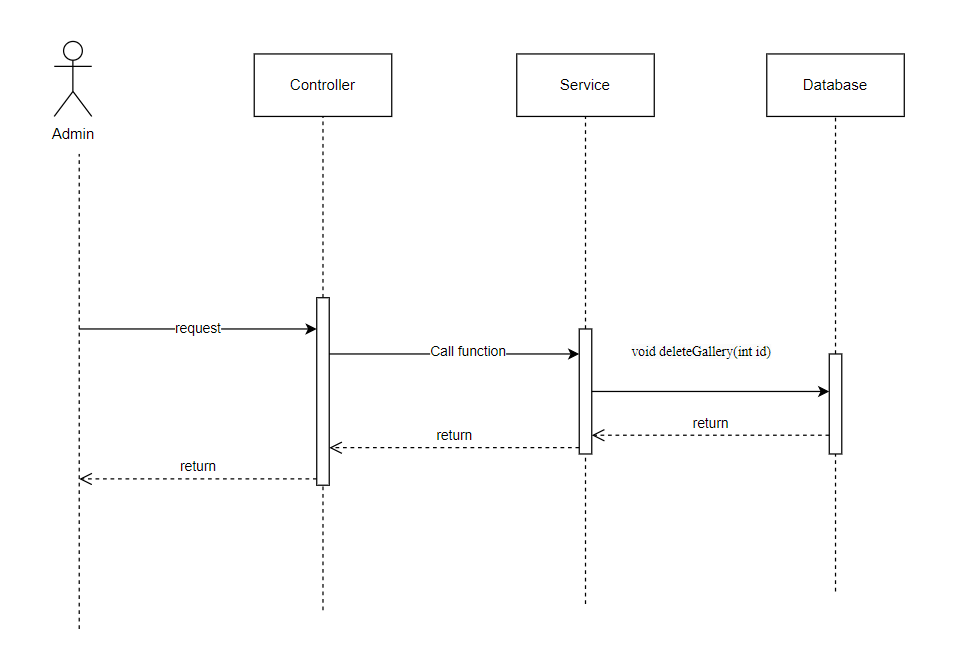
**a. Class Diagram**

****

**b. Class specifications**

| **Class** | **Method** | **Description** |
| --- | --- | --- |
| Discount | Get, set | Get, set attribute of class |
| CrudDAO | void deleteGallery(int id) | Delete Discount |
| DBContext | DBContext() | Connect to the database |
| DeleteDiscountServlet | doGet() | Request to JSP to user input information |
| doPost() | Call function from CrudDAO and send message to JSP |

**c. Sequence Diagram**

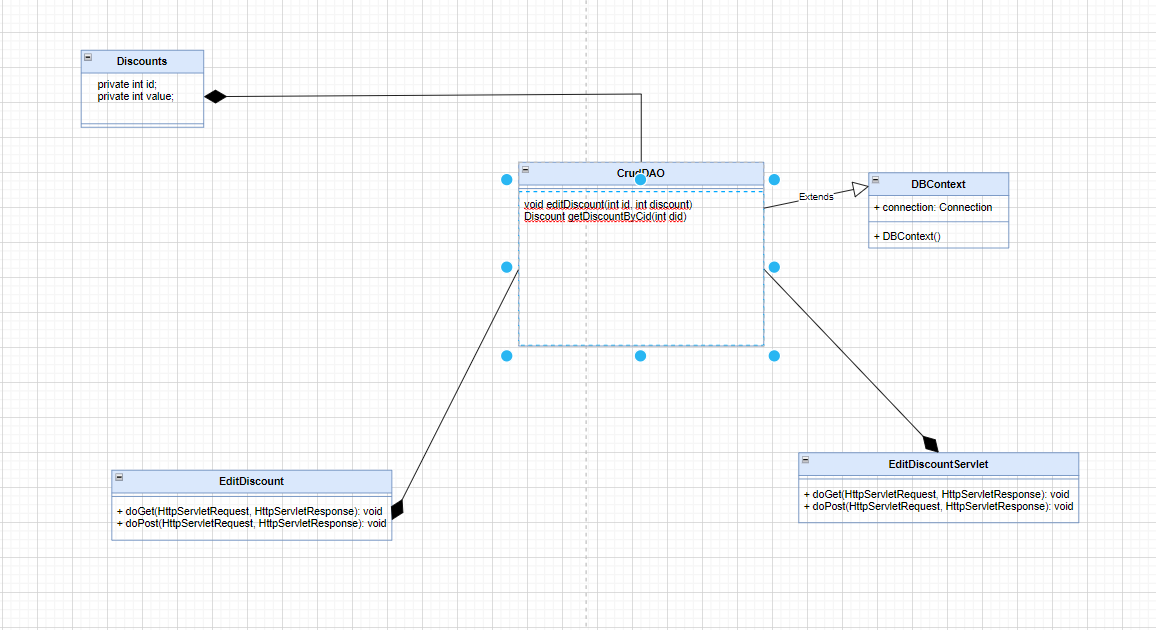
****

**d. Database Queries**

| **Class** | **Method** | **SQL** |
| --- | --- | --- |
| CrudDAO | void deleteGallery(int id) | DELETE FROM [dbo].[Galeries]\n"  + " WHERE id = ? |

**15.3 Edit Discount**

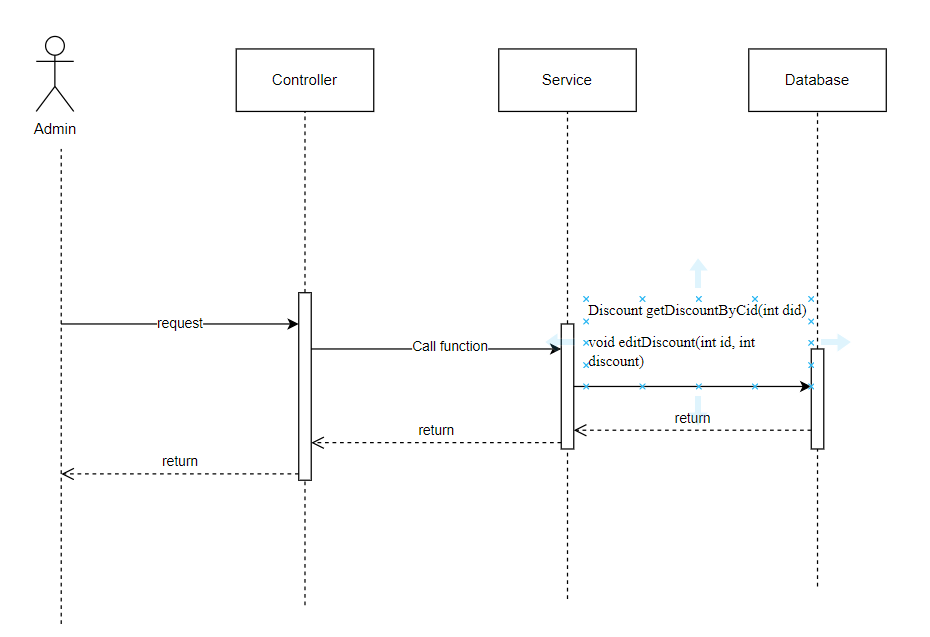
**a. Class Diagram**

****

**b. Class specifications**

| **Class** | **Method** | **Description** |
| --- | --- | --- |
| Discount | Get, set | Get, set attribute of class |
| CrudDAO | void editDiscount(int id, int discount)  Discount getDiscountByCid(int did) | Edit Discount |
| DBContext | DBContext() | Connect to the database |
| Discount getDiscountByCid(int did) | doGet() | Request to JSP to user input information |
| doPost() | Call function from CrudDAO and send message to JSP |
| void editDiscount(int id, int discount) | doGet() | Request to JSP to user input information |
| doPost() | Call function from CrudDAO and send message to JSP |

**c. Sequence Diagram**

****

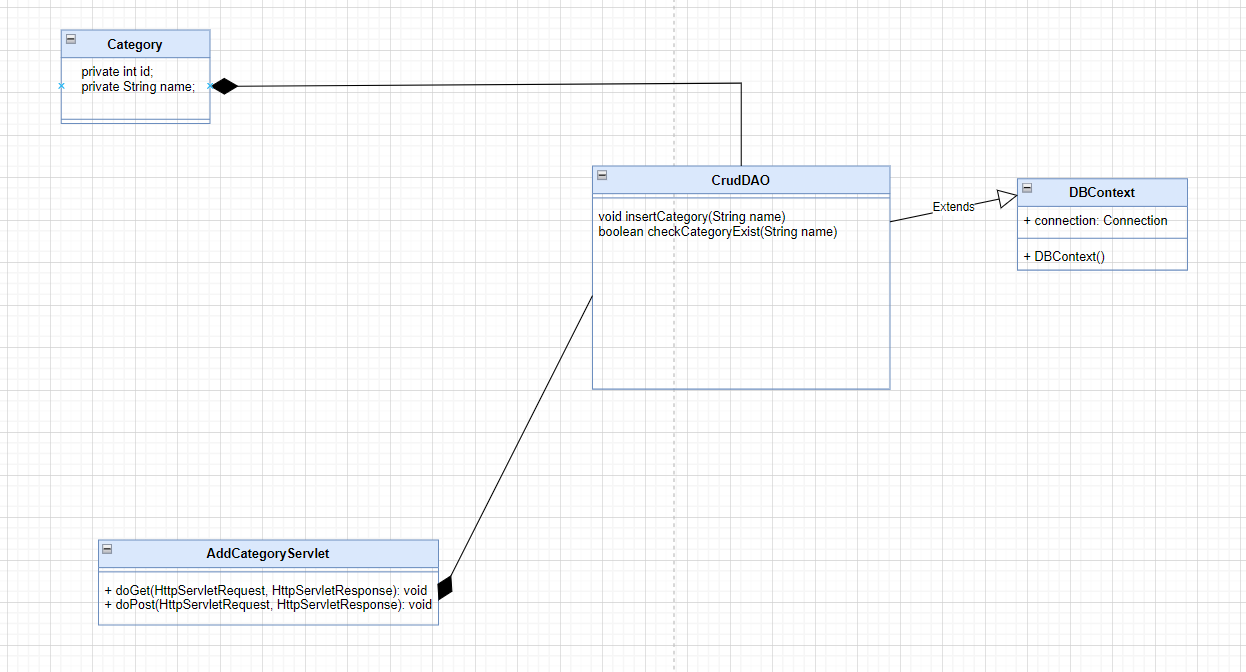
**d. Database Queries**

| **Class** | **Method** | **SQL** |
| --- | --- | --- |
| CrudDAO | void editDiscount(int id, int discount)  Discount getDiscountByCid(int did) | UPDATE [dbo].[Discounts]\n"  + " SET [value] = ?\n"  + " WHERE id = ?  SELECT [id]\n"  + " ,[value]\n"  + " FROM [dbo].[Discounts]"  + " where id = ? |

**16. Manage Category**

**16.1 Add Category**

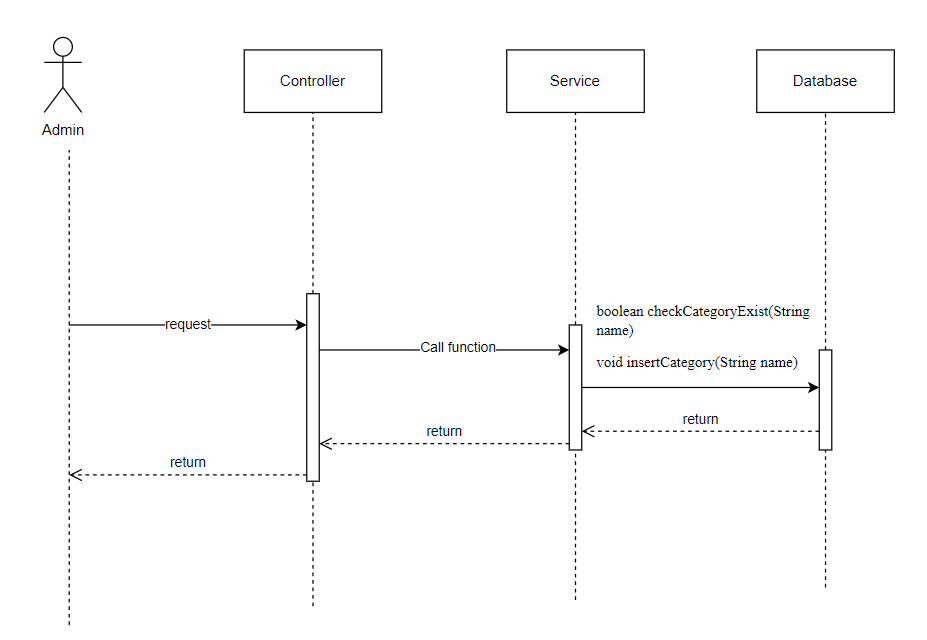
**a. Class Diagram**

****

**b. Class specifications**

| **Class** | **Method** | **Description** |
| --- | --- | --- |
| Category | Get, set | Get, set attribute of class |
| CrudDAO | void insertCategory(String name)  boolean checkCategoryExist(String name) | Insert Category |
| DBContext | DBContext() | Connect to the database |
| AddCategoryServlet | doGet() | Request to JSP to user input information |
| doPost() | Call function from CrudDAO and send message to JSP |

**c. Sequence Diagram**

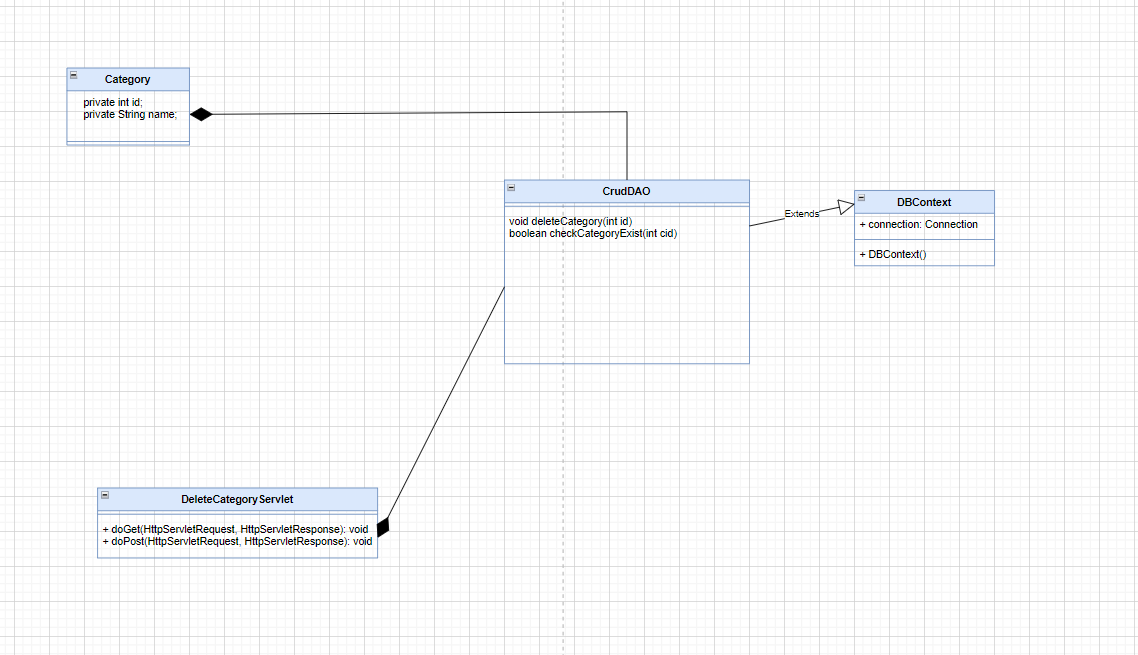
****

**d. Database Queries**

| **Class** | **Method** | **SQL** |
| --- | --- | --- |
| CrudDAO | void insertCategory(String name)  boolean checkCategoryExist(String name) | INSERT INTO [dbo].[Categories]\n"  + " ([name])\n"  + " VALUES\n"  + " (?)  SELECT [id]\n"  + " ,[name]\n"  + " FROM [dbo].[Categories]\n"  + "\n"  + " where name = ? |

**16.2 Delete Category**

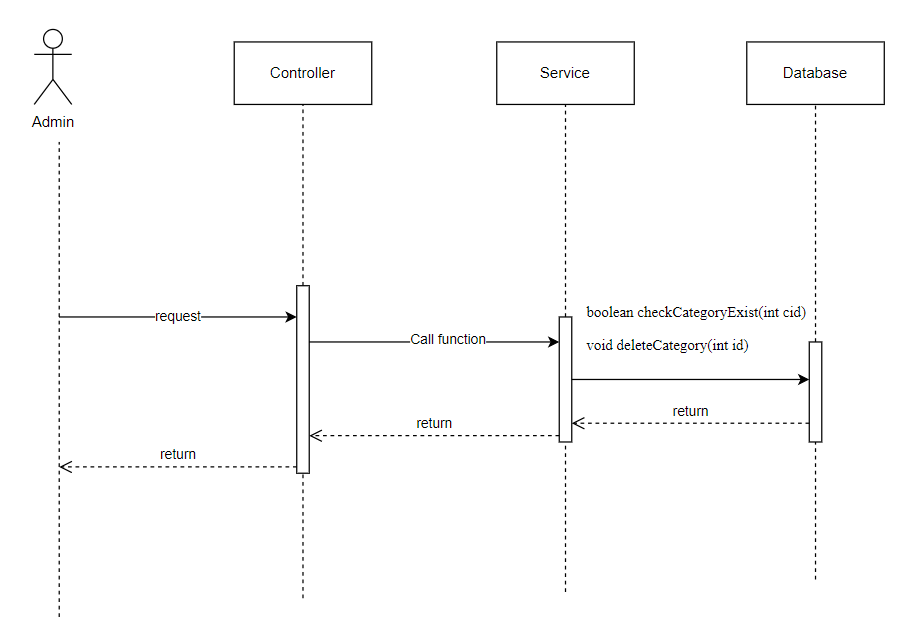
**a. Class Diagram**

****

**b. Class specifications**

| **Class** | **Method** | **Description** |
| --- | --- | --- |
| Category | Get, set | Get, set attribute of class |
| CrudDAO | void deleteCategory(int id)  boolean checkCategoryExist(int cid) | Delete Category |
| DBContext | DBContext() | Connect to the database |
| DeleteCategoryServlet | doGet() | Request to JSP to user input information |
| doPost() | Call function from CrudDAO and send message to JSP |

**c. Sequence Diagram**

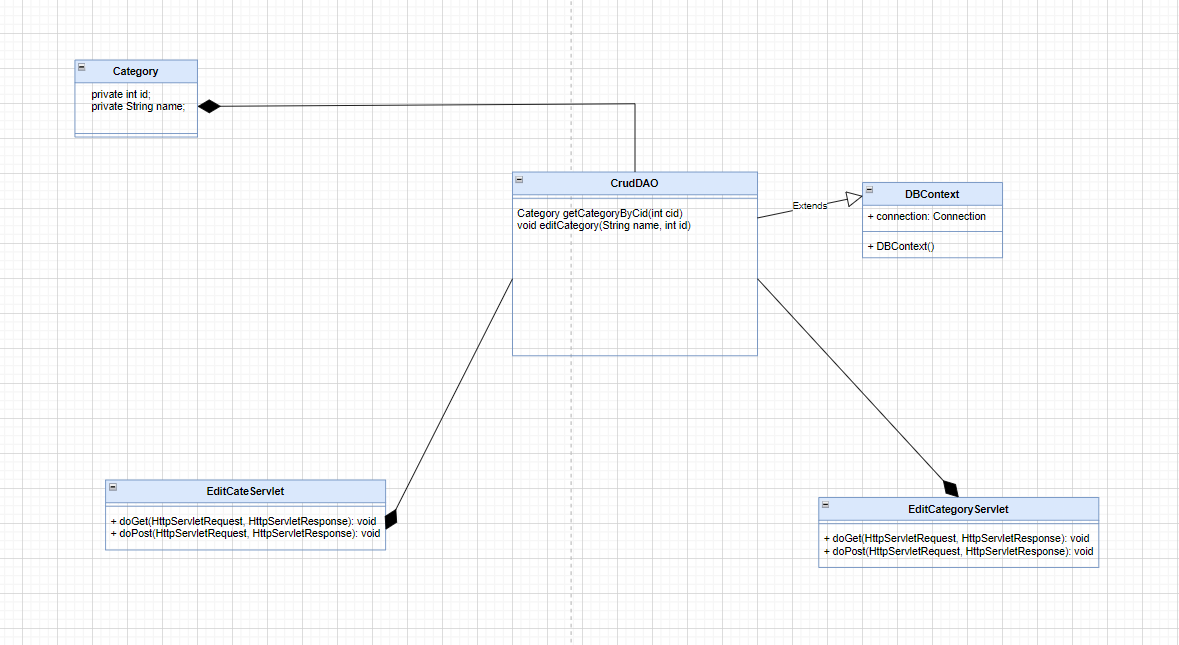
****

**d. Database Queries**

| **Class** | **Method** | **SQL** |
| --- | --- | --- |
| CrudDAO | void deleteCategory(int id)  boolean checkCategoryExist(int cid) | DELETE FROM [dbo].[Categories]\n"  + " WHERE id = ?  SELECT [category\_id]\n"  + " \n"  + " FROM [SWP391\_1].[dbo].[Products]\n"  + " group by [category\_id] |

**16.3 Edit Category**

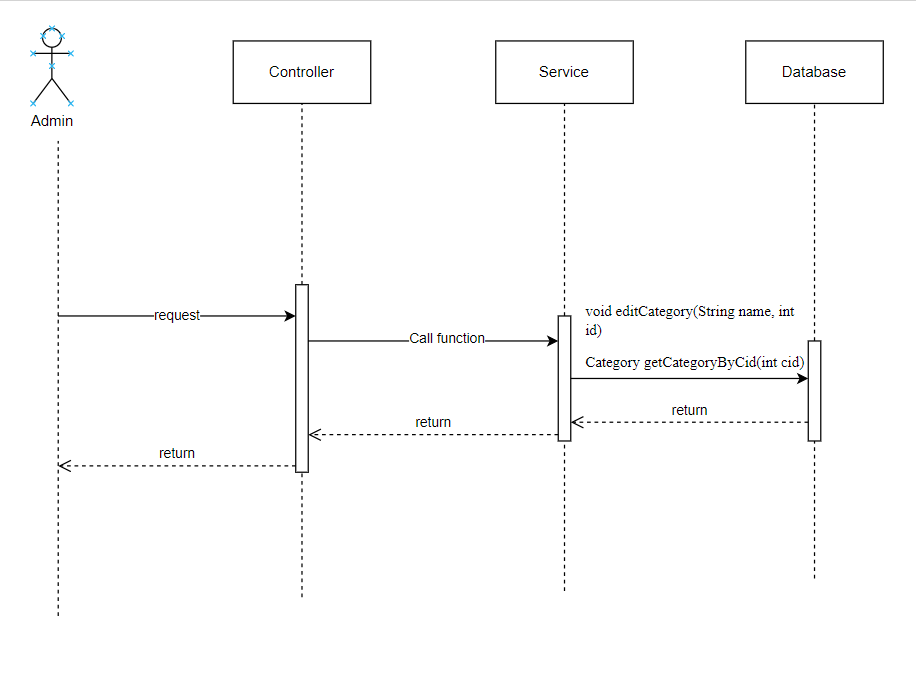
**a. Class Diagram**

****

**b. Class specifications**

| **Class** | **Method** | **Description** |
| --- | --- | --- |
| Category | Get, set | Get, set attribute of class |
| CrudDAO | Category getCategoryByCid(int cid)  void editCategory(String name, int id) | Edit Category |
| DBContext | DBContext() | Connect to the database |
| EditCategoryServlet | doGet() | Request to JSP to user input information |
| doPost() | Call function from CrudDAO and send message to JSP |
| EditCateServlet | doGet() | Request to JSP to user input information |
| doPost() | Call function from CrudDAO and send message to JSP |

**c. Sequence Diagram**

****

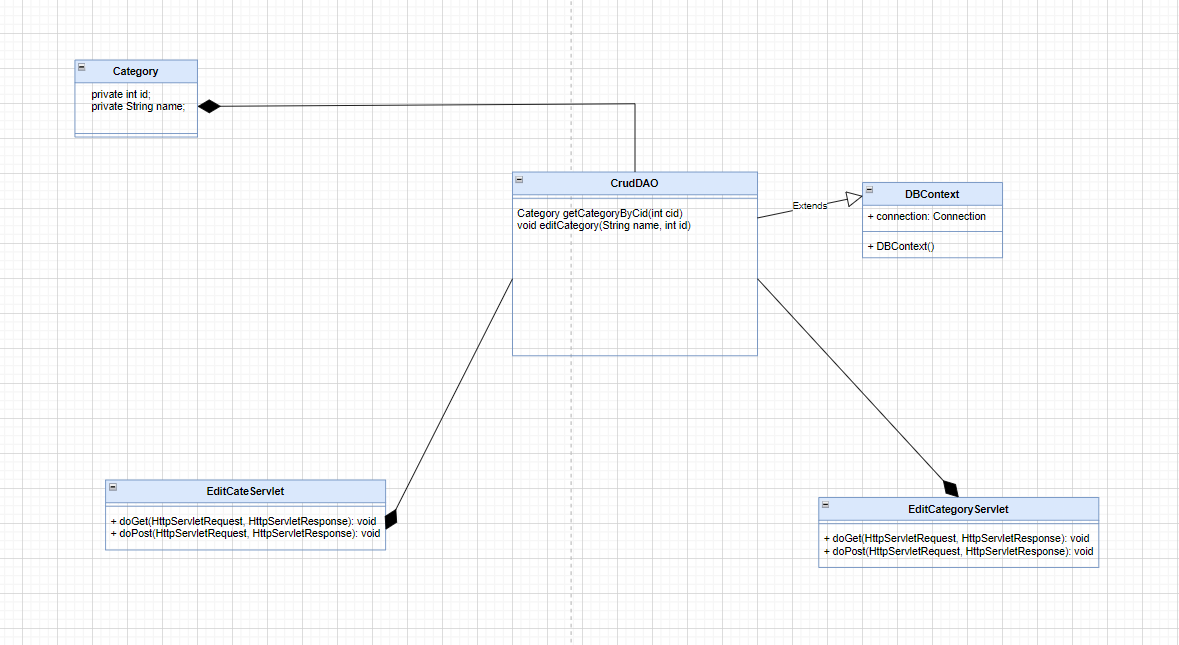
**d. Database Queries**

| **Class** | **Method** | **SQL** |
| --- | --- | --- |
| CrudDAO | Category getCategoryByCid(int cid)  void editCategory(String name, int id) | SELECT [id]\n"  + " ,[name]\n"  + " FROM [dbo].[Categories]\n"  + " where id = ?  UPDATE [dbo].[Categories]\n"  + " SET [name] = ?\n"  + " WHERE id = ? |

**=========**

**17. Product detail**

**a. Class Diagram**

****

**b. Class specifications**

| **Class** | **Method** | **Description** |
| --- | --- | --- |
| Category, Product | Get, set | Get, set attribute of class |
| ProductDAO | getProductByID(int id)  getSizeProductByPidSid(int pid, int sid)  randomRelative(int gid, int pid) | Show detail of a product by clicking on it. |
| DBContext | DBContext() | Connect to the database |
| DetailServlet | doGet() | Request to JSP to user input information |

**c. Sequence Diagram**

**d. Database Queries**

| **Class** | **Method** | **SQL** |
| --- | --- | --- |
| ProductDAO | getProductByID(int id)  getSizeProductByPid  Sid(int pid, int sid)  randomRelative(int gid, int pid) | select \* from Products p join Discounts d\n"  + "on p.discount\_id=d.id\n"  + "where p.id=?  select \* from SizeProduct where pid=? and sid=?  select top 4 p.title, p.gender\_id, p.id, p.thumbnail, p.updated\_at, p.discount\_id, d.value, min(sp.price\_out) as price, min(sp.sid) as sid from Products p\n"  + " join SizeProduct sp on p.id = sp.pid\n"  + " join Sizes s on s.id = sp.sid\n"  + " join Discounts d on d.id = p.discount\_id\n"  + " where p.gender\_id=? and p.id !=?\n"  + " group by p.title, p.id, p.thumbnail, p.updated\_at , p.discount\_id, d.value, p.gender\_id\n"  + " order by NEWID() |

**18. Relative product**

**a. Class Diagram**

**b. Class specifications**

| **Class** | **Method** | **Description** |
| --- | --- | --- |
| Product | Get, set | Get, set attribute of class |
| ProductDAO | randomRelative(int gid, int pid) | Show 4 random relative products that already on detail |
| DBContext | DBContext() | Connect to the database |
| DetailServlet | doGet() | Request to JSP to user input information |

**c. Sequence Diagram**

**d. Database Queries**

| **Class** | **Method** | **SQL** |
| --- | --- | --- |
| ProductDAO | randomRelative(int gid, int pid) | select top 4 p.title, p.gender\_id, p.id, p.thumbnail, p.updated\_at, p.discount\_id, d.value, min(sp.price\_out) as price, min(sp.sid) as sid from Products p\n"  + " join SizeProduct sp on p.id = sp.pid\n"  + " join Sizes s on s.id = sp.sid\n"  + " join Discounts d on d.id = p.discount\_id\n"  + " where p.gender\_id=? and p.id !=?\n"  + " group by p.title, p.id, p.thumbnail, p.updated\_at , p.discount\_id, d.value, p.gender\_id\n"  + " order by NEWID() |

**19. Statistic order, product, category, revenue**

**a. Class Diagram**

**b. Class specifications**

| **Class** | **Method** | **Description** |
| --- | --- | --- |
| Order, Order\_Detail, Product, Category | Get, set | Get, set attribute of class |
| DashboardDAO, ProductDAO, CategoryDAO | getAllSuccessOrder(int year)  getAllProduct()  getAllCategory()  getRevenueByEachMonth(int year)  getProfitEachMonth(int year) | Show statistic of product, category, revenue and order. |
| DBContext | DBContext() | Connect to the database |
| Dashboard | doGet() | Request to JSP to user input information |

**c. Sequence Diagram**

**d. Database Queries**

| **Class** | **Method** | **SQL** |
| --- | --- | --- |
| DashboarDAO, ProductDAO,  CategoryDAO | getAllSuccessOrder(int year)  getAllProduct()  getAllCategory()  getRevenueByEachMonth(int year)  getProfitEachMonth(int year) | select \* from Orders o join Status s \n"  + " on o.status\_id = s.id\n"  + " where o.status\_id=3 and year(order\_date) =?\n"  + " order by o.id desc  select \* from products  SELECT [id]\n"  + " ,[name]\n"  + " FROM [dbo].[Categories]  SELECT DATEFROMPARTS(?, m.month, 1) AS month\_date, \n"  + " COALESCE(SUM(o.order\_quantity), 0) AS total\_orders, \n"  + " COALESCE(SUM(o.product\_quantity), 0) AS total\_products, \n"  + " SUM(o.total\_money) AS total\_money\n"  + "FROM \n"  + " (SELECT 1 AS month UNION SELECT 2 UNION SELECT 3 UNION SELECT 4 UNION SELECT 5 UNION SELECT 6 \n"  + " UNION SELECT 7 UNION SELECT 8 UNION SELECT 9 UNION SELECT 10 UNION SELECT 11 UNION SELECT 12) AS m\n"  + "LEFT JOIN \n"  + " (SELECT MONTH(order\_date) AS month, COUNT(distinct a.id) AS order\_quantity, \n"  + " SUM(b.num) AS product\_quantity, SUM(total\_money) AS total\_money\n"  + " FROM Orders a JOIN Order\_Details b ON a.id = b.order\_id\n"  + " WHERE YEAR(order\_date) = ? and a.status\_id=3-- chỉ lấy dữ liệu cho năm 2023\n"  + " GROUP BY MONTH(order\_date)) AS o\n"  + "ON m.month = o.month\n"  + "GROUP BY DATEFROMPARTS(?, m.month, 1), m.month;  SELECT \n"  + " DATEFROMPARTS(2023, month, 1) AS month\_date, \n"  + " COALESCE(SUM( od.total\_money\_in), 0) AS tongnhap,\n"  + " COALESCE(SUM(od.total\_money\_out ), 0) AS tongra,\n"  + " COALESCE(SUM(od.total\_money\_out - od.total\_money\_in), 0) AS profit \n"  + "FROM \n"  + " (SELECT 1 AS month UNION SELECT 2 UNION SELECT 3 UNION SELECT 4 UNION SELECT 5 UNION SELECT 6 \n"  + " UNION SELECT 7 UNION SELECT 8 UNION SELECT 9 UNION SELECT 10 UNION SELECT 11 UNION SELECT 12) AS months \n"  + " LEFT JOIN orders o ON months.month = MONTH(o.order\_date) AND o.status\_id = 3 AND YEAR(o.order\_date) = ?\n"  + " LEFT JOIN order\_details od ON o.id = od.order\_id \n"  + "GROUP BY \n"  + " DATEFROMPARTS(2023, month, 1) \n"  + "ORDER BY \n"  + " DATEFROMPARTS(2023, month, 1); |

**20. List order**

**a. Class Diagram**

**b. Class specifications**

| **Class** | **Method** | **Description** |
| --- | --- | --- |
| Order, Order\_Detail, Product, User | Get, set | Get, set attribute of class |
| DashboardDAO | getOrderByID(int id)  getAllOrder() | Show list of order that have status\_id = 1, 2, 3, 4 |
| DBContext | DBContext() | Connect to the database |
| OrderList | doGet() | Request to JSP to user input information |
| doPost() | Call function from DashboardDAO and send message to JSP |

**c. Sequence Diagram**

**d. Database Queries**

| **Class** | **Method** | **SQL** |
| --- | --- | --- |
| DashboardDAO | getOrderByID(int id)  getAllOrder() | select \* from Orders o join Status s \n"  + " on o.status\_id = s.id\n"  + " \n"  + " order by o.id desc  select \* from Orders o join Status s \n"  + " on o.status\_id = s.id\n"  + " where o.status\_id = ?\n"  + " order by o.id desc |

**20.1 View order detail and change order’s status**

**a. Class Diagram**

**b. Class specifications**

| **Class** | **Method** | **Description** |
| --- | --- | --- |
| Order, Order\_Detail, Product, User | Get, set | Get, set attribute of class |
| DashboardDAO | getOrderByID(int id)  getAllOrder()  getOrderDetailById(int id) | View an order detail and change it status. |
| DBContext | DBContext() | Connect to the database |
| DetailServlet | doGet() | Request to JSP to user input information |

**c. Sequence Diagram**

**d. Database Queries**

| **Class** | **Method** | **SQL** |
| --- | --- | --- |
| DashboardDAO | getOrderByID(int id)  getAllOrder()  getOrderDetailById(int id) | select \* from Orders o join Status s \n"  + " on o.status\_id = s.id\n"  + " \n"  + " order by o.id desc  select \* from Orders o join Status s \n"  + " on o.status\_id = s.id\n"  + " where o.status\_id = ?\n"  + " order by o.id desc  select u.id as 'userid', u.firstname, u.lastname, u.email, u.phone\_number, \n"  + " o.id as'orderid', o.note, o.order\_date, o.status\_id, o.total\_money, \n"  + " od.price, od.num,\n"  + " p.id as'productid', p.title, p.gender\_id, p.discount\_id, p.thumbnail, \n"  + " s.id as 'sizeid', s.name, ad.city, ad.district, ad.ward, ad.detail,\n"  + " st.name as 'statusname', st.id as'statusid'\n"  + "from Users u join Orders o\n"  + " on u.id= o.user\_id\n"  + " join Order\_Details od\n"  + " on od.order\_id = o.id\n"  + " join Products p \n"  + " on p.id = od.product\_id\n"  + " join SizeProduct sp \n"  + " on sp.pid = p.id\n"  + " join Sizes s \n"  + " on s.id= sp.sid\n"  + " join Address\_Detail ad\n"  + " on ad.uid = u.id\n"  + " join [Status] st \n"  + " on st.id = o.status\_id\n"  + " where o.id=? and s.id=od.size\_id and ad.id= o.address\_id |

**21. List account, view profile and change status**

**a. Class Diagram**

**b. Class specifications**

| **Class** | **Method** | **Description** |
| --- | --- | --- |
| User | Get, set | Get, set attribute of class |
| DashboardDAO | getAllStaff()  getAllUser() | View list account |
| DBContext | DBContext() | Connect to the database |
| DetailServlet | doGet() | Request to JSP to user input information |

**c. Sequence Diagram**

**d. Database Queries**

| **Class** | **Method** | **SQL** |
| --- | --- | --- |
| DashboardDAO | getAllStaff()  getAllUser() | select \* from Orders o join Status s \n"  + " on o.status\_id = s.id\n"  + " \n"  + " order by o.id desc  SELECT \*\n"  + " FROM [Users]\n"  + " where role\_id=2 |

**21.1 Add new account and default address**

**a. Class Diagram**

**b. Class specifications**

| **Class** | **Method** | **Description** |
| --- | --- | --- |
| User | Get, set | Get, set attribute of class |
| DashboardDAO | addAccount(User u)  addAddress(Address\_Detail ad) | Add a new account and address |
| DBContext | DBContext() | Connect to the database |
| DetailServlet | doPost() | Request to JSP to user input information |

**c. Sequence Diagram**

**d. Database Queries**

| **Class** | **Method** | **SQL** |
| --- | --- | --- |
| DashboardDAO | addAccount(User u)  addAddress(Address\_Detail ad) | INSERT INTO [dbo].[Users]\n"  + " ([loginType]\n"  + " ,[role\_id]\n"  + " ,[firstname]\n"  + " ,[lastname]\n"  + " ,[username]\n"  + " ,[password]\n"  + " ,[email]\n"  + " ,[phone\_number]\n"  + " ,[created\_at]\n"  + " ,[updated\_at]\n"  + " ,[deleted], [avatar])\n"  + " VALUES\n"  + " (1,?,?,?,?,?,?,?,?,?,?,?)  INSERT INTO [dbo].[Address\_Detail]\n"  + " ([uid]\n"  + " ,[city]\n"  + " ,[district]\n"  + " ,[ward]\n"  + " ,[detail]\n"  + " ,[is\_default])\n"  + " VALUES\n"  + " (?,?,?,?,?,?) |