

**CLOTHING SHOP PROJECT REPORT**

**Software Design Document**

– Hanoi, Summer 2022 –

**Table of Contents**

[I. Overview 7](#_Toc28206)

[1. Code Packages/Namespaces 7](#_Toc30026)

[2. Coding Conventions 8](#_Toc26962)

[II. Code Designs 8](#_Toc13523)

[1. Home Page 8](#_Toc20597)

[2. Login 9](#_Toc25675)

[a. Class Diagram 9](#_Toc4358)

[b. Class Specifications 9](#_Toc15571)

[c. Sequence Diagram(s) 10](#_Toc9056)

[d. Database queries 10](#_Toc29201)

[4. Product detail 11](#_Toc32353)

[a. Class Diagram 11](#_Toc12788)

[b. Class Specifications 12](#_Toc2215)

[c. Sequence Diagram(s) 13](#_Toc32639)

[d. Database queries 13](#_Toc31887)

[5. Register 14](#_Toc10561)

[a. Class Diagram 14](#_Toc7545)

[b. Class Specifications 14](#_Toc24598)

[c. Sequence Diagram(s) 15](#_Toc20444)

[d. Database queries 16](#_Toc22888)

[6. Add to cart 17](#_Toc29473)

[a. Class Diagram 17](#_Toc12)

[b. Class Specifications 17](#_Toc4981)

[c. Sequence Diagram(s) 18](#_Toc4459)

[d. Database queries 18](#_Toc31460)

[7. Blog List 19](#_Toc12446)

[a. Class Diagram 19](#_Toc9331)

[b. Class Specifications 19](#_Toc1945)

[c. Sequence Diagram(s) 20](#_Toc26920)

[d. Database queries 20](#_Toc3751)

[8. Blog Detail 21](#_Toc15893)

[a. Class Diagram 21](#_Toc5309)

[b. Class Specifications 21](#_Toc21468)

[c. Sequence Diagram(s) 22](#_Toc14192)

[d. Database queries 22](#_Toc11085)

[9. Logout 23](#_Toc12181)

[a. Class Diagram 23](#_Toc4115)

[b.Class Specifications 23](#_Toc16330)

[c. Sequence Diagram(s) 23](#_Toc12671)

[d. Database queries 24](#_Toc15493)

[10. Add Product 25](#_Toc4850)

[a. Class Diagram 25](#_Toc8538)

[b. Class Specifications 25](#_Toc6277)

[c. Sequence Diagram(s) 26](#_Toc1421)

[11. Change amount 27](#_Toc22989)

[a. Class Diagram 27](#_Toc4661)

[b. Class Specifications 27](#_Toc29686)

[c. Sequence Diagram(s) 27](#_Toc28281)

[12. Delete cart 28](#_Toc7565)

[a. Class Diagram 28](#_Toc31405)

[b. Class Specifications 28](#_Toc4520)

[c. Sequence Diagram(s) 29](#_Toc14679)

[d. Database queries 30](#_Toc21845)

[13. View List User 30](#_Toc32106)

[a. Class Diagram 30](#_Toc12659)

[b. Class Specifications 30](#_Toc28462)

[c. Sequence Diagram(s) 31](#_Toc19557)

[d. Database queries 31](#_Toc17852)

[14. Edit Information User 32](#_Toc27192)

[a. Class Diagram 32](#_Toc16621)

[b. Class Specifications 32](#_Toc20286)

[c. Sequence Diagram(s) 32](#_Toc4143)

[d. Database queries 33](#_Toc24610)

[15.Set Role User 33](#_Toc29858)

[a. Class Diagram 33](#_Toc13318)

[b. Class Specifications 33](#_Toc8544)

[c. Sequence Diagram(s) 33](#_Toc12332)

[d. Database queries 34](#_Toc13036)

[16. Block User 35](#_Toc16943)

[a. Class Diagram 35](#_Toc7952)

[b. Class Specifications 35](#_Toc13540)

[c. Sequence Diagram(s) 36](#_Toc18611)

[d. Database queries 36](#_Toc15700)

[17. Reset Password 37](#_Toc32631)

[a. Class Diagram 37](#_Toc2158)

[b. Class Specifications 37](#_Toc12566)

[c. Sequence Diagram(s) 38](#_Toc5987)

[d. Database queries 39](#_Toc730)

[18. Change Password 39](#_Toc26571)

[a. Class Diagram 39](#_Toc31653)

[b. Class Specifications 39](#_Toc3585)

[c. Sequence Diagram(s) 40](#_Toc27488)

[d. Database queries 41](#_Toc23388)

[19. User profile 41](#_Toc25297)

[a. Class Diagram 41](#_Toc14122)

[b. Class Specifications 41](#_Toc26402)

[c. Sequence Diagram(s) 42](#_Toc18593)

[d. Database queries 43](#_Toc9888)

[20. Order History 43](#_Toc16664)

[a. Class Diagram 43](#_Toc24705)

[b. Class Specifications 43](#_Toc20807)

[c. Sequence Diagram(s) 44](#_Toc32678)

[d. Database queries 45](#_Toc26983)

[21. Delete Product 46](#_Toc10428)

[a. Class Diagram 46](#_Toc9058)

[b. Class Specifications 46](#_Toc15920)

[c. Sequence Diagram(s) 47](#_Toc28534)

[d. Database queries 47](#_Toc12482)

[22. Edit Product 48](#_Toc7927)

[a. Class Diagram 48](#_Toc27074)

[b. Class Specifications 48](#_Toc1552)

[c. Sequence Diagram(s) 49](#_Toc17476)

[d. Database queries 49](#_Toc29885)

[23. Checkout 49](#_Toc29892)

[a. Class Diagram 49](#_Toc2124)

[b. Class Specifications 49](#_Toc31503)

[c. Sequence Diagram(s) 50](#_Toc8740)

[d. Database queries 50](#_Toc22648)

[24.Manager Product 51](#_Toc30129)

[a. Class diagram 51](#_Toc21657)

[b. Class specifications 51](#_Toc28311)

[c. Sequence Diagram(s) 52](#_Toc11567)

[d. Database queries 52](#_Toc31134)

[25.HomePage 52](#_Toc12761)

[a. Class Diagram 52](#_Toc19966)

[b. Class Specifications 52](#_Toc4797)

[c. Sequence Diagram 53](#_Toc489)

[d. Database queries 54](#_Toc8540)

[26. List Customer 54](#_Toc19592)

[a. Class Diagram 54](#_Toc32042)

[b. Class Specification 55](#_Toc10439)

[c. Sequence Diagram(s) 55](#_Toc12694)

[d. Database queries 55](#_Toc26535)

[26. Product list 55](#_Toc14762)

[a.Class diagram 55](#_Toc30188)

[b.Class specifications 55](#_Toc15734)

[c. Sequence Diagram(s) 56](#_Toc30812)

[d. Database queries 56](#_Toc16335)

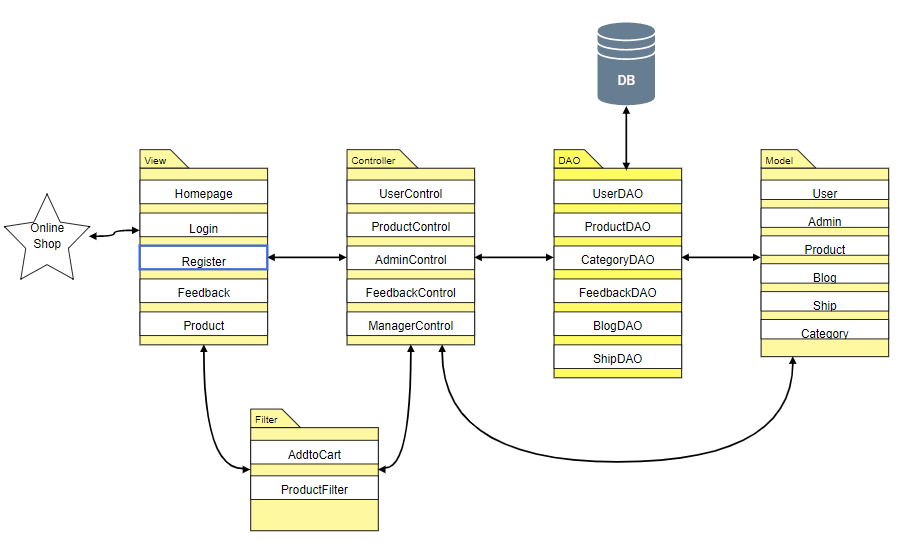
[III. Database Design 57](#_Toc15091)

[1. Database Schema 57](#_Toc26429)

[2. Table Description 58](#_Toc4788)

# I. Overview

## 1. Code Packages/Namespaces



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

|  |  |  |
| --- | --- | --- |
| **No** | **Package** | **Description** |
| *01* | *Controllers* | This package contains all controllers for this MVC2 project and authentication or dispatcher filters  Servlet Naming convention: Starting with a verb.  Ex: LoginController, SignUpController,... |
| *02* | *Views* | This package contains all views (HTML,CSS,JS,JSP file) for this project  HTML/CSS/JS/JSP file naming convention: first letter is a lowercase letter, using camelCase naming |
| *03* | *Model* | This package provide the connection to the Database, Extract data for the system  Naming convention: first letter is a lowercase letter, the name is using camelCase naming  This package .java files which allow developers to skip writing DAOs for their persistence objects when they usingJDBC. |
| *04* | *Utils* | This package contains DBcontext.java file which is a bridge connection between database and programming and contains sendEmail.java allows users to send email  Naming convention: first letter is a lowercase letter, the name is using camelCase naming |
| *05* | *Filter* | This package handles the Authentication that helps avoid duplicate in controller  Naming convention: first letter is a lowercase letter, the name is using camelCase naming |

## 2. Coding Conventions

# II. Code Designs

## 1. Feature Product

### a. Class Diagram



### b. Class Specifications

|  |  |  |
| --- | --- | --- |
| **No** | **Method** | **Description** |
| *01* | *getAllProduct()* | *Query will get all product from database* |

### c. Sequence Diagram(s)

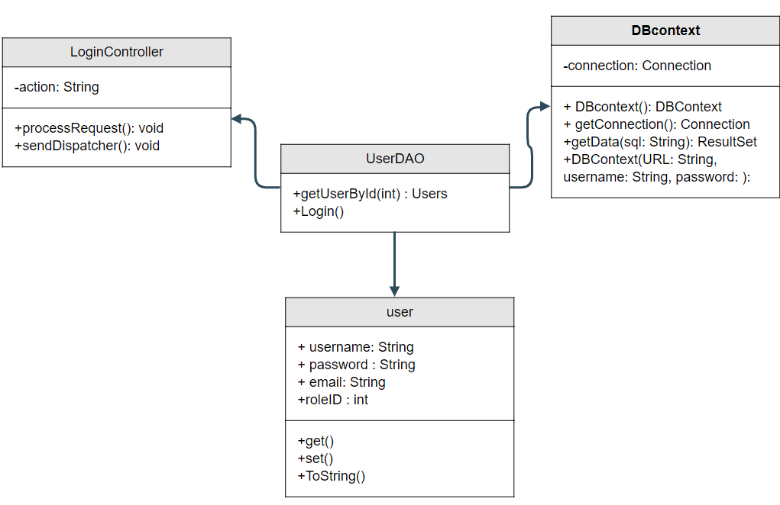
**

### d. Database queries

select p.ProductID , ProductName , Description , OriginalPrice , SellPrice , SalePercent ,SubCategoryID , SellerID , Amount , StatusID , StatusID ,BrandID , height , width ,weight , s.ProductImgURL from Product p join ProductImg s on p.ProductID = s.ProductID

## 2. Login

### Class Diagram



### Class Specifications

UserController

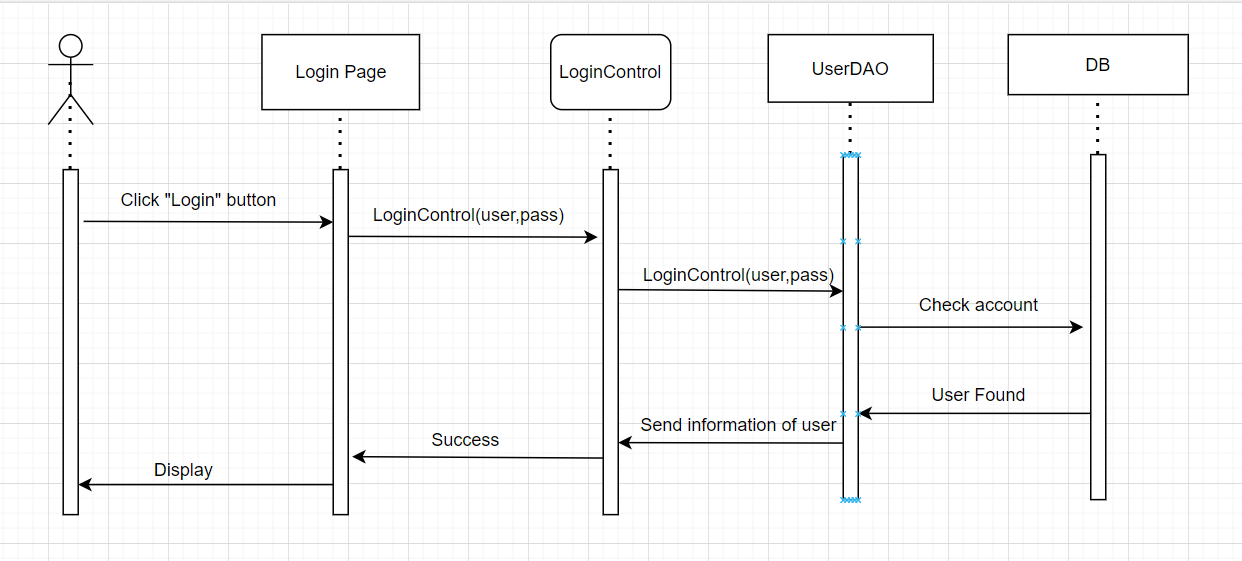
|  |  |  |
| --- | --- | --- |
| No | Method | Description |
| 01 | processRequest() | get all new information from user Form and update the database |

UserDAO

[Provide the detailed description for the class methods]

|  |  |  |
| --- | --- | --- |
| No | Method | Description |
| 01 | getUserList() | get the list of user |
| 02 | getUserById(int uId) | get the user where the user id is |

### Sequence Diagram(s)

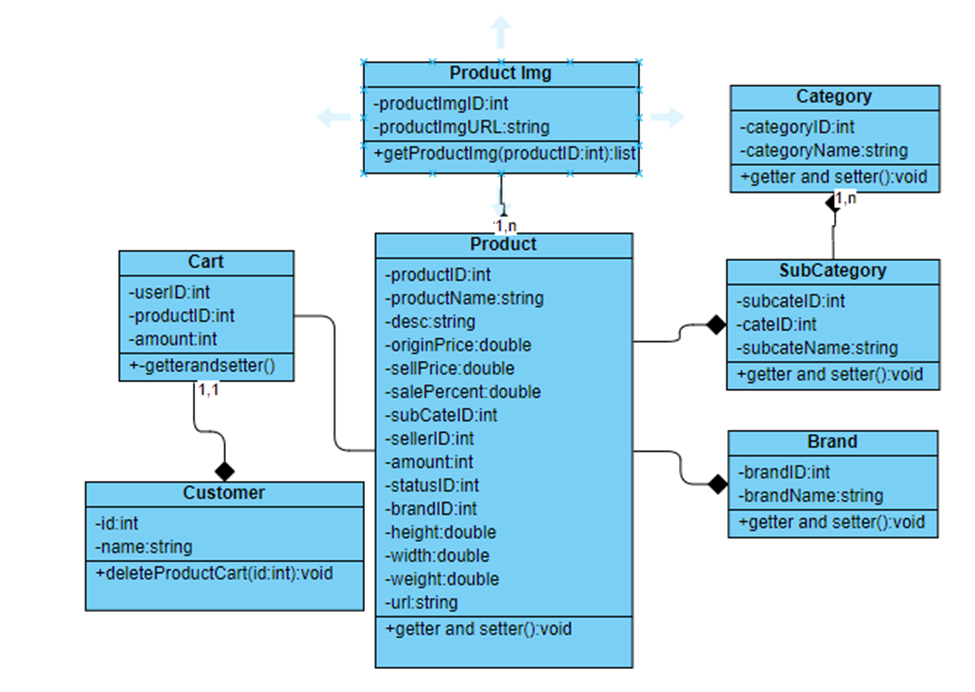


### Database queries

**"select \* from [User] where username=? and password=?"**

**3.Cart Detail**

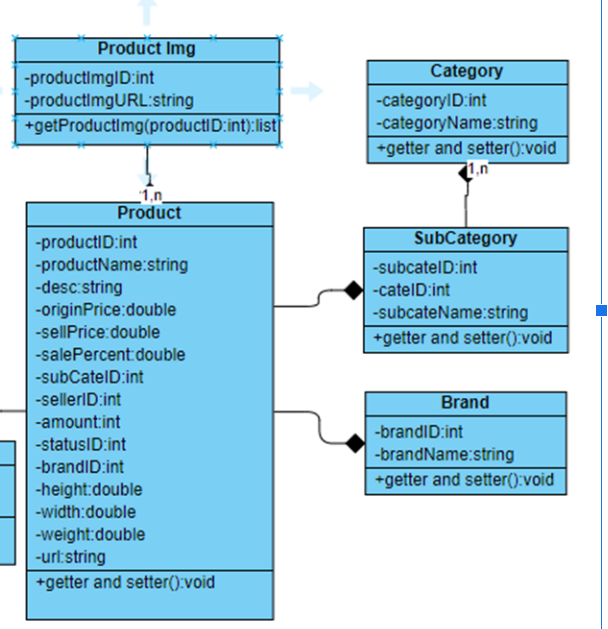
1. **Class Diagram**



### 

## 4. Product detail

### Class Diagram



### Class Specifications

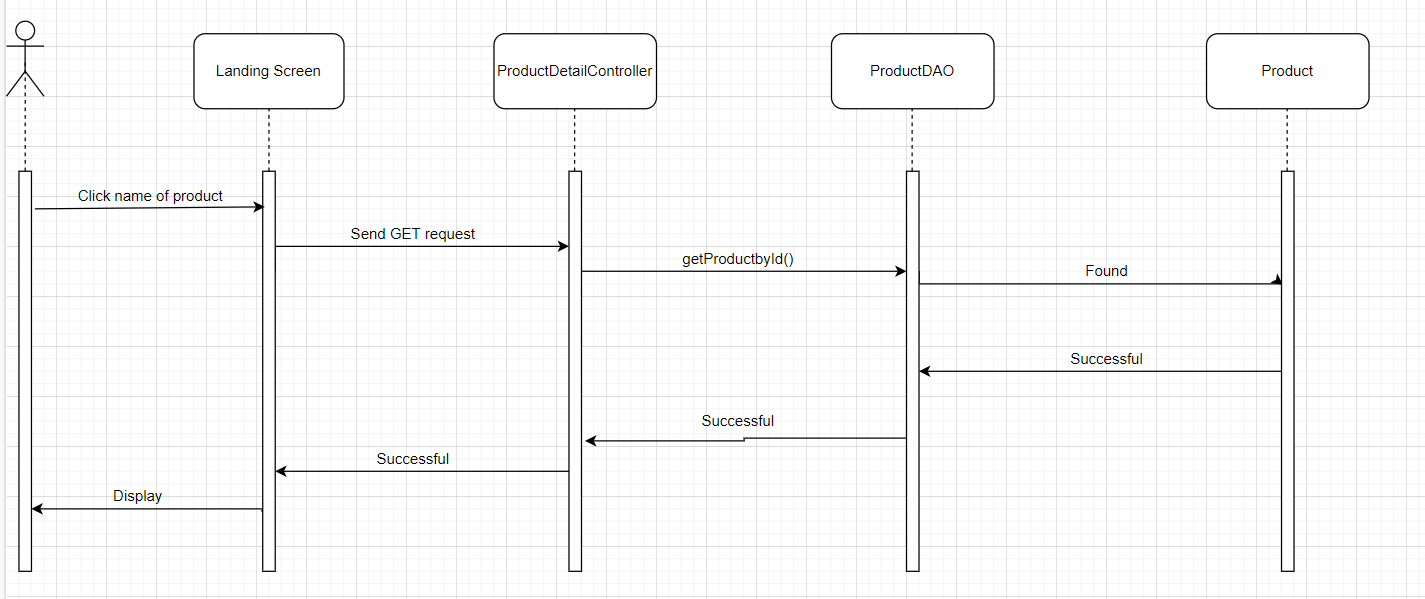
|  |  |  |
| --- | --- | --- |
| No | Method | Description |
| 01 | processRequest() |  |

ProductDAO

|  |  |  |
| --- | --- | --- |
| No | Method | Description |
| 01 | getProductByID(int id) | get Product by ID |

### 

### c. Sequence Diagram(s)



### d. Database queries

"select p.\*,s.SubCategoryName,c.\*,b.BrandName from Product p join SubCategory s \n"

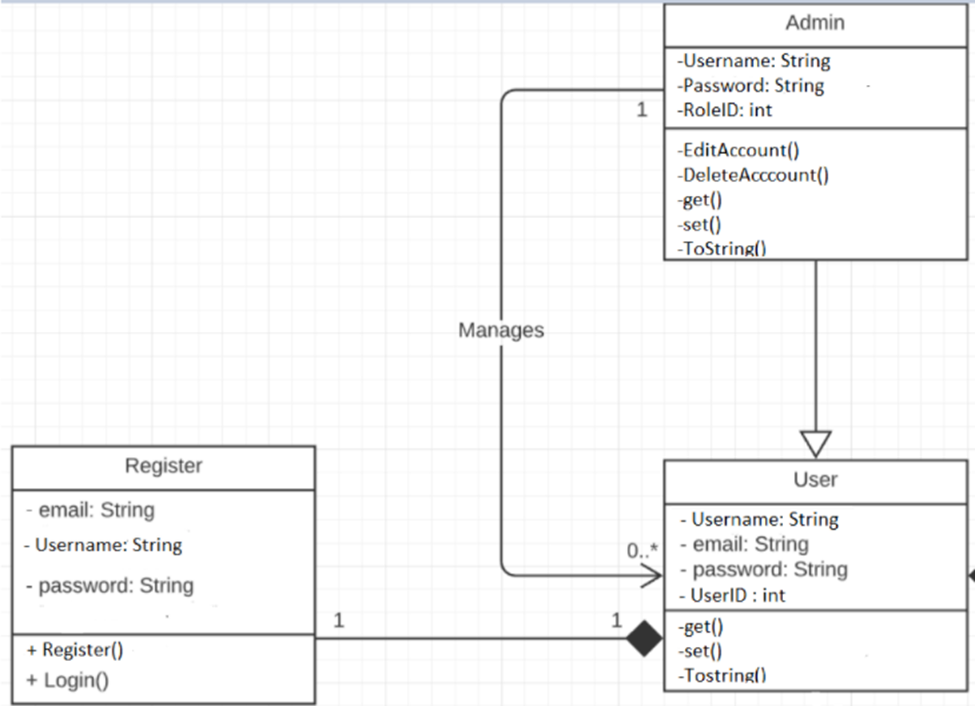
+ "on p.SubCategoryID = s.SubCategoryID\n"

+ "join Brand b on b.BrandID = p.BrandID\n"

+ "join Category c on c.CategoryID = s.CategoryID"

## 5. Register

### **a. Class Diagram**



### b. Class Specifications

UserController

|  |  |  |
| --- | --- | --- |
| No | Method | Description |
| 01 | processRequest() |  |

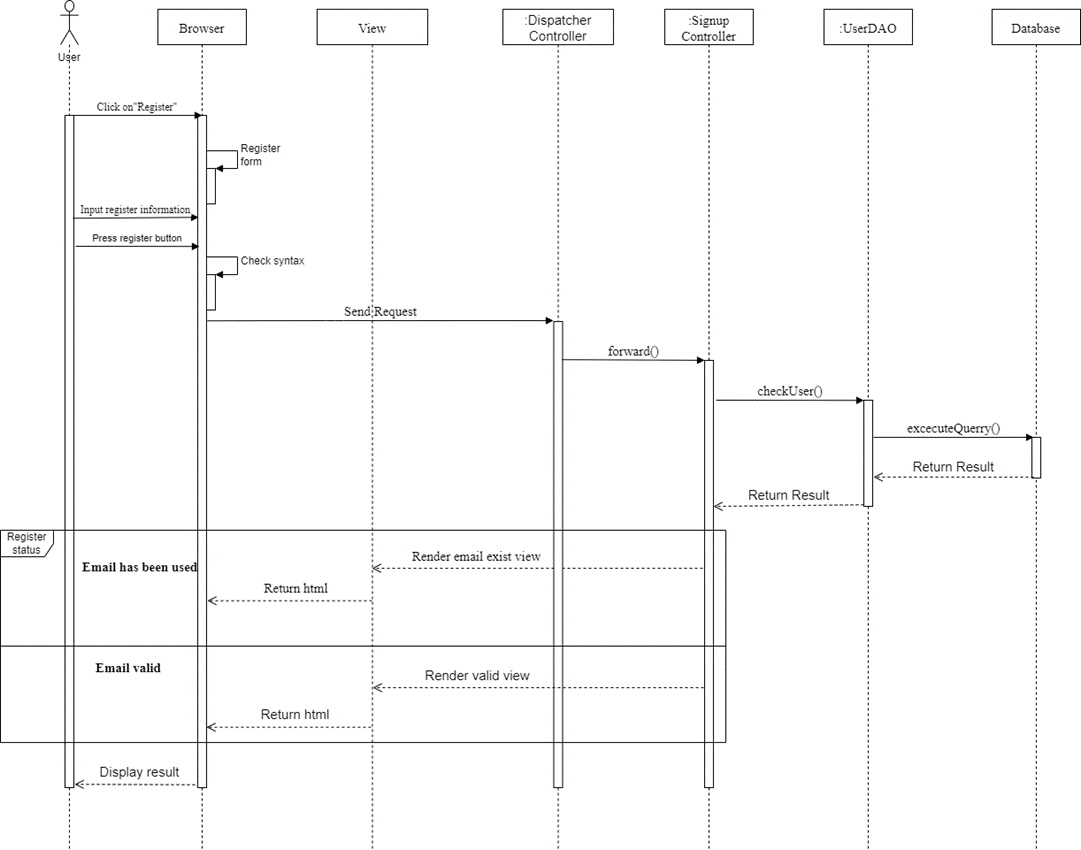
UserDAO

[Provide the detailed description for the class methods]

|  |  |  |
| --- | --- | --- |
| No | Method | Description |
| 01 | Register( String username,String password, String mai) | insert a new user with Name , Pass, Mail) |
| 02 | checkExistAccount(String xName) | - check if the username is already exist  - return null |

### 

### c. Sequence Diagram(s)



### d. Database queries

**"**INSERT INTO [dbo].[Users]

([Username

,[Password]

,[email]

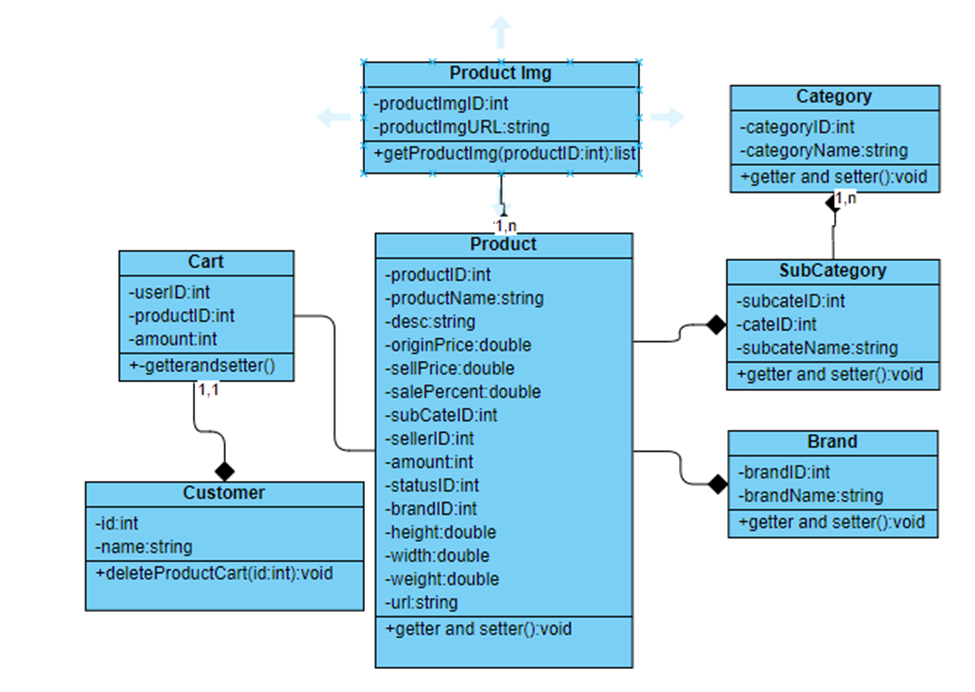
,[RoleID]

,[isBlock])

VALUES (?,?,?,?,?)"**"**

## 6. Add to cart

### Class Diagram



### Class Specifications

CartDetailController

|  |  |  |
| --- | --- | --- |
| No | Method | Description |
| 01 | processRequest() |  |

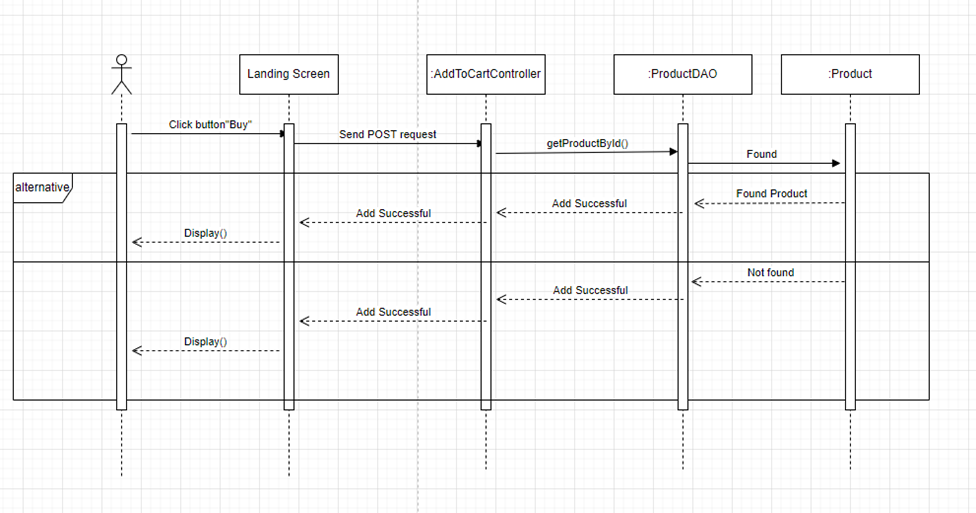
CartDAO

[Provide the detailed description for the class methods]

|  |  |  |
| --- | --- | --- |
| No | Method | Description |
| *01* | *deleteProductCart()* | *Remove a product() from cart* |

### 

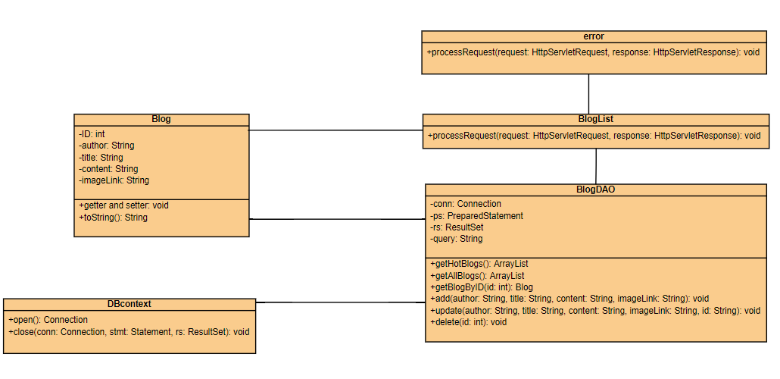
### c. Sequence Diagram(s)



### d. Database queries

## Blog List

### Class Diagram



### Class Specifications

- BlogList class

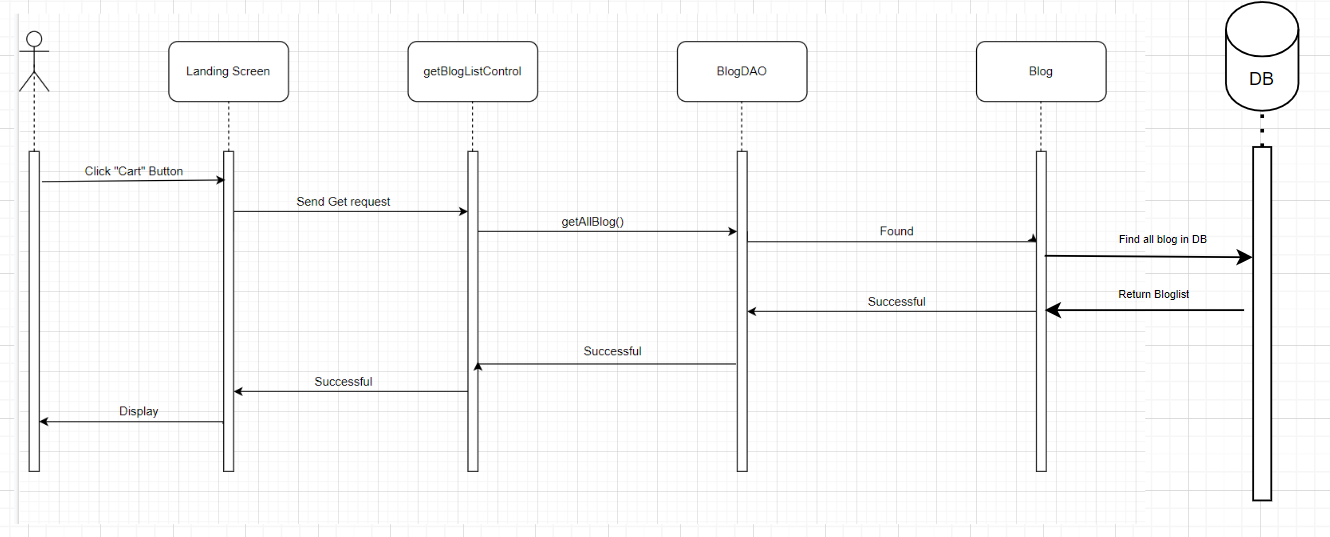
|  |  |  |
| --- | --- | --- |
| **No** | **Method** | **Description** |
| 01 | processRequest() | get blog list from the database to display on BlogList.jsp |

- BlogDAO class

|  |  |  |
| --- | --- | --- |
| **No** | **Method** | **Description** |
| 01 | getBlogByID(int id) | Return: Blog Detail where blog id contain parameter  Input: ID of blog |
| 02 | getHotBlogs() | Return: top 3 newest blog from database |
| 03 | getAllBlog() | Return: All blog in database |
| 04 | add(String author, String title, String content, String imageLink) | Add blog with detail entered in addBlog.jsp page  Return: null  Input: author, title, content, imageLink |
| 05 | update(String author, String title, String content, String imageLink, String id) | Update blog with detail entered in updateBlog.jsp page  Return: null  Input: author, title, content, imageLink, id |
| 06 | delete(int id) | Update blog with id entered in blogManage.jsp  Return: null  Input: id |

### 

### c. Sequence Diagram(s)



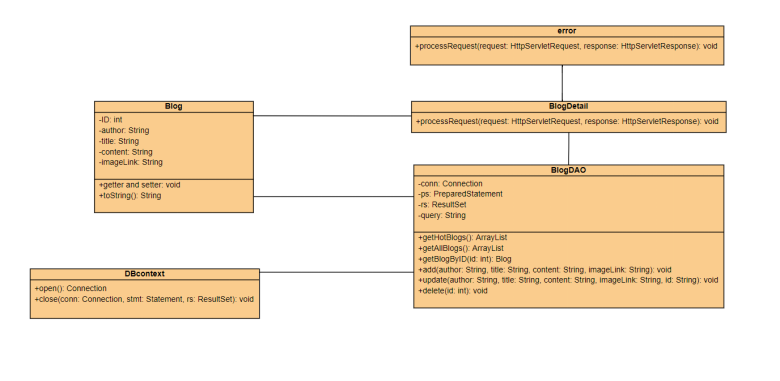
### 

### d. Database queries

**Select \* from Blog desc**

## 8. Blog Detail

### Class Diagram



### Class Specifications

- BlogDetail class

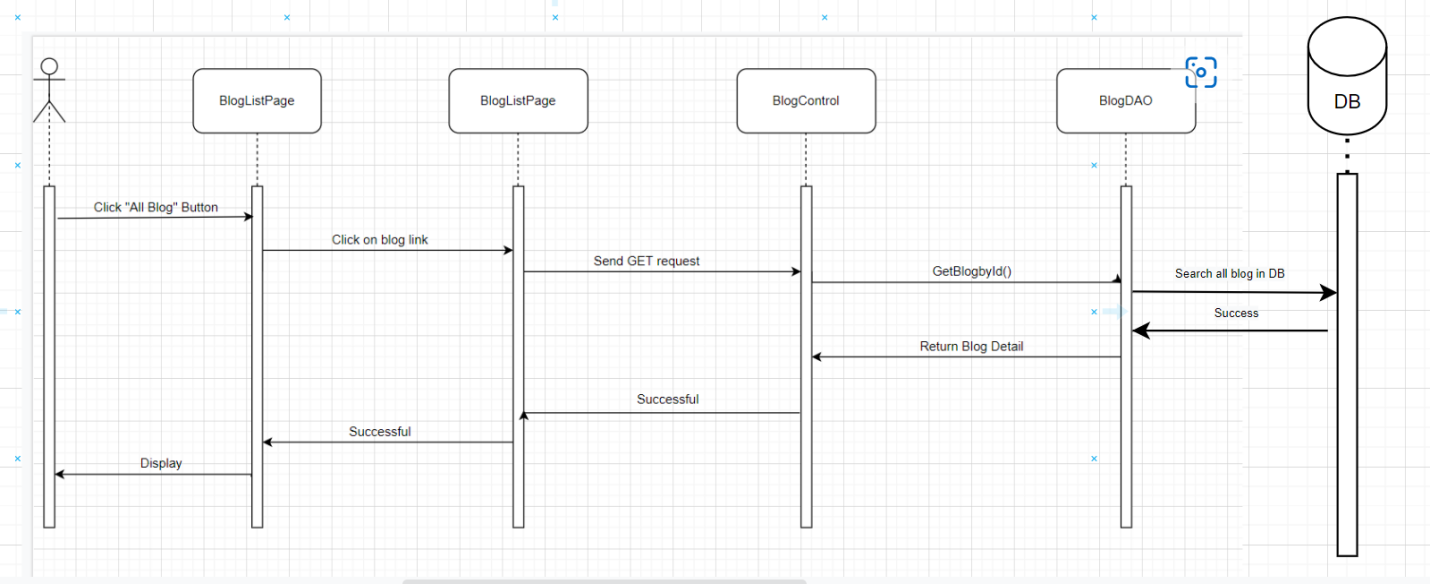
|  |  |  |
| --- | --- | --- |
| **No** | **Method** | **Description** |
| 01 | processRequest() | get blog detail from the database to display on BlogDetail.jsp |

- BlogDAO class

|  |  |  |
| --- | --- | --- |
| **No** | **Method** | **Description** |
| 01 | getBlogByID(int id) | Return: Blog Detail where blog id contain parameter  Input: ID of blog |
| 02 | getHotBlogs() | Return: top 3 newest blog from database |
| 03 | getAllBlog() | Return: All blog in database |
| 04 | add(String author, String title, String content, String imageLink) | Add blog with detail entered in addBlog.jsp page  Return: null  Input: author, title, content, imageLink |
| 05 | update(String author, String title, String content, String imageLink, String id) | Update blog with detail entered in updateBlog.jsp page  Return: null  Input: author, title, content, imageLink, id |
| 06 | delete(int id) | Update blog with id entered in blogManage.jsp  Return: null  Input: id |

### 

### c. Sequence Diagram(s)



### 

### d. Database queries

SELECT \* FROM Blog WHERE ID = ?

## 9. Logout

### a. Class Diagram

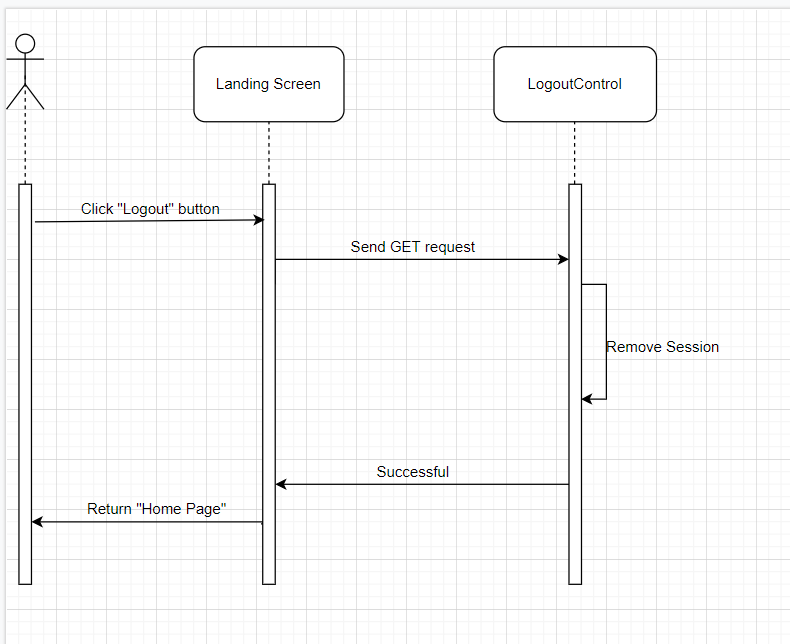
### b.Class Specifications

LogoutController

|  |  |  |
| --- | --- | --- |
| No | Method | Description |
| 01 | processRequest() | Remove attribute from session |

### 

### c. Sequence Diagram(s)

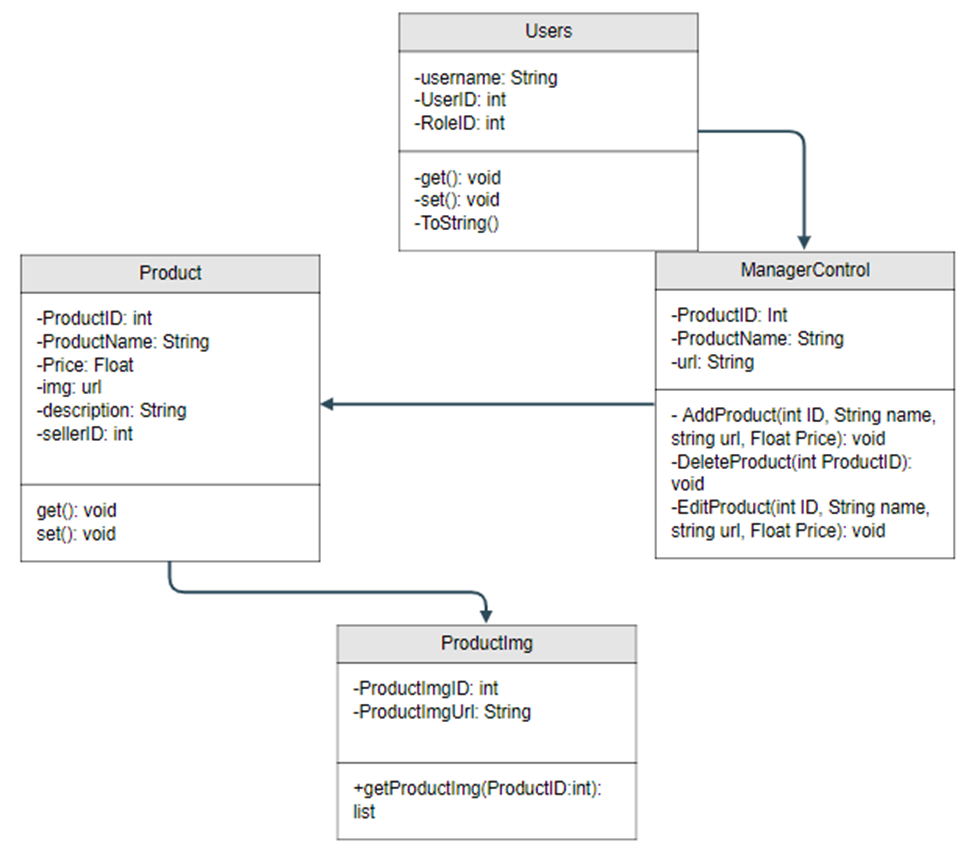


### 

### d. Database queries

## 10. Add Product

### a. Class Diagram

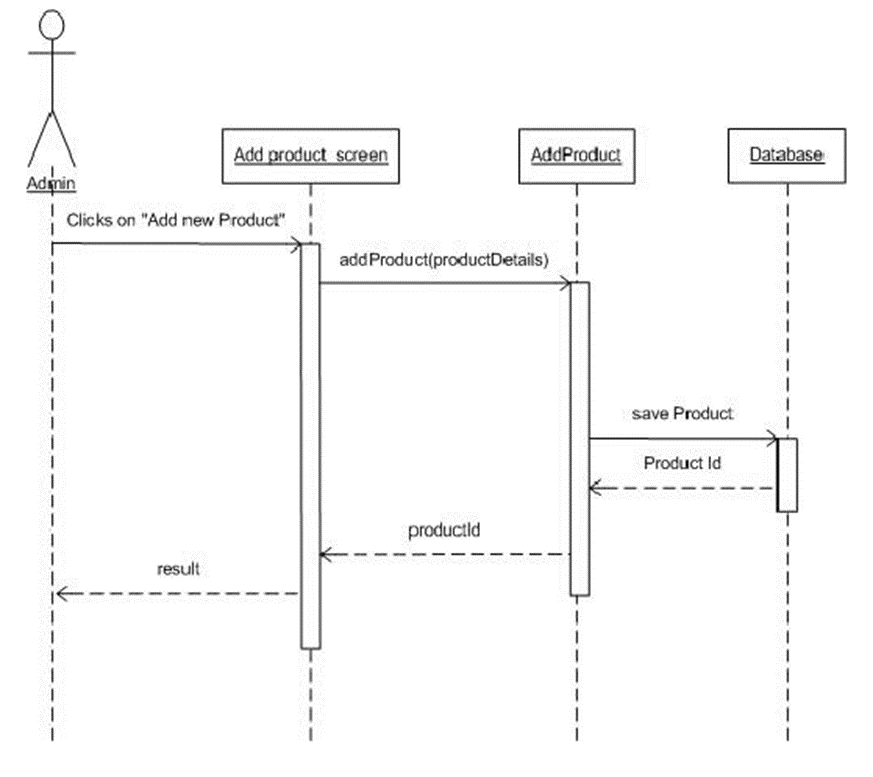


### b. Class Specifications

|  |  |  |
| --- | --- | --- |
| No | Method | Description |
| 01 | addProduct() | AddProduct from the database to display on ManageProduct.jsp |

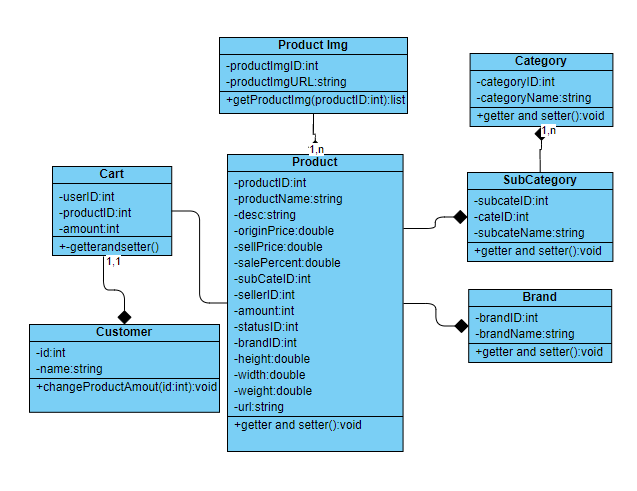
### 

### c. Sequence Diagram(s)

 d. Database queries

## 11. Change amount

### a. Class Diagram

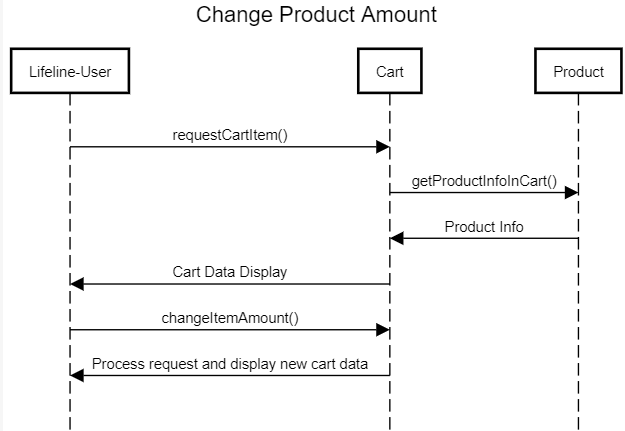


### b. Class Specifications

Customer

|  |  |  |
| --- | --- | --- |
| **No** | **Method** | **Description** |
| *01* | *changeProductAmount()* | *Add or reduce amount of a product from cart* |

### c. Sequence Diagram(s)



### d. Database queries

UPDATE dbo.cart

SET amount = ?

WHERE userID = ? and productID = ?

## 12. Delete cart

### a. Class Diagram

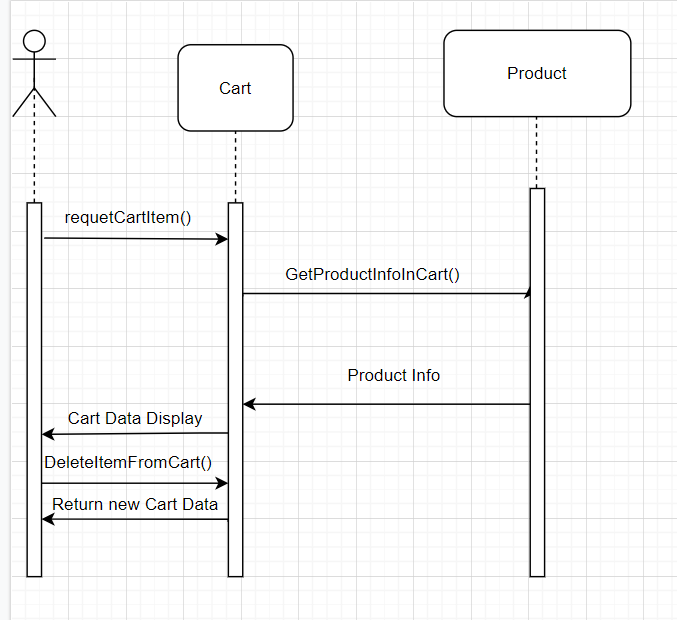
### b. Class Specifications

UserController

|  |  |  |
| --- | --- | --- |
| No | Method | Description |
| 01 | processRequest() | Delete Cart by ProductID through HashMap |

### 

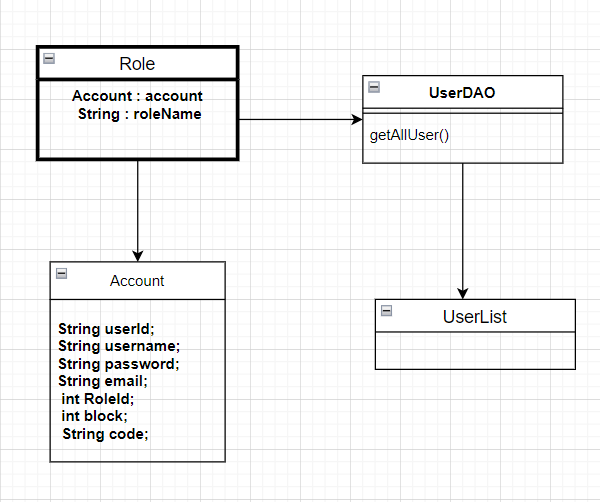
### c. Sequence Diagram(s)



### **d. Database queries**

## 13. View List User

### a. Class Diagram



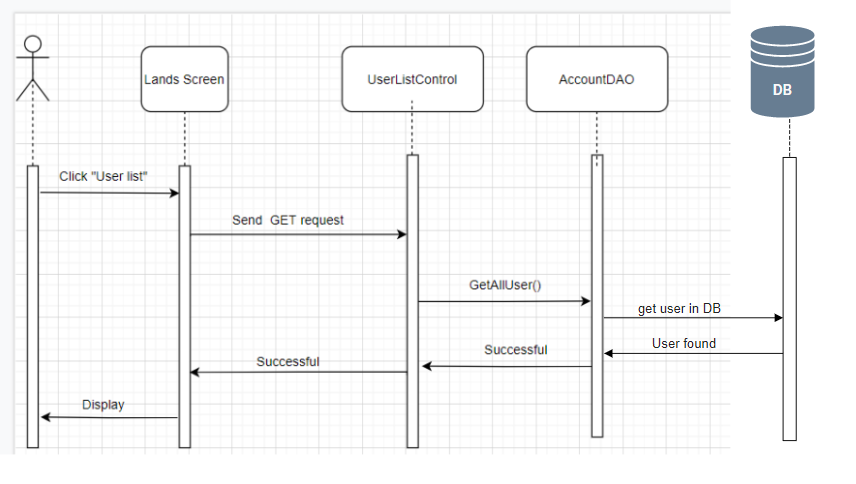
### b. Class Specifications

UserDAO

|  |  |  |
| --- | --- | --- |
| No | Method | Description |
| 01 | GetAllUser() | Get All User from Database |

### 

### c. Sequence Diagram(s)



### 

### d. Database queries

select u.\*,r.RoleName from Users u inner join role r on u.RoleID = r.RoleID

## 14. Edit Information User

### a. Class Diagram

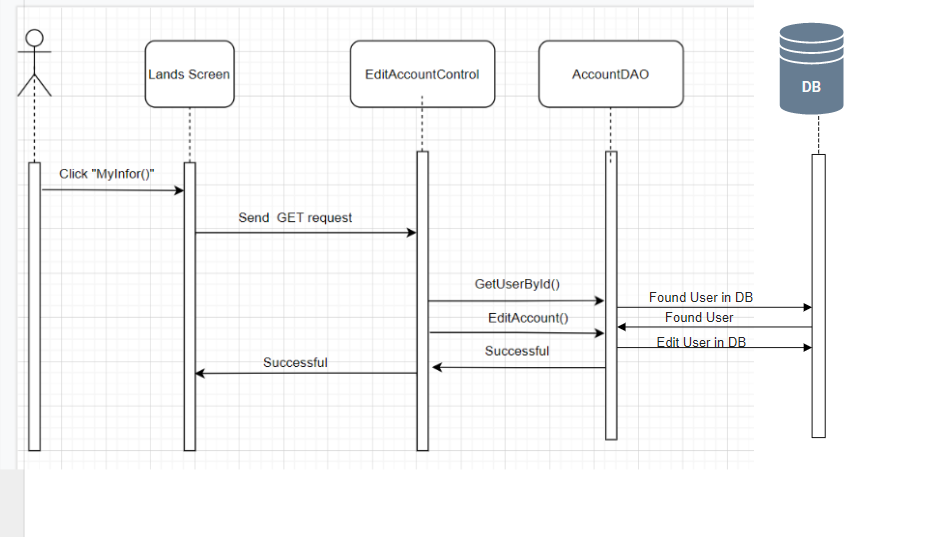
### b. Class Specifications

UserDAO

|  |  |  |
| --- | --- | --- |
| No | Method | Description |
| 01 | Create User | Insert new user to database |
| 02 | Validate user | Check user if exists |

### 

### c. Sequence Diagram(s)



### d. Database queries

Select \* from Users

## 15.Set Role User

### a. Class Diagram

### b. Class Specifications

UserController

|  |  |  |
| --- | --- | --- |
| No | Method | Description |
| 01 | processRequest() |  |

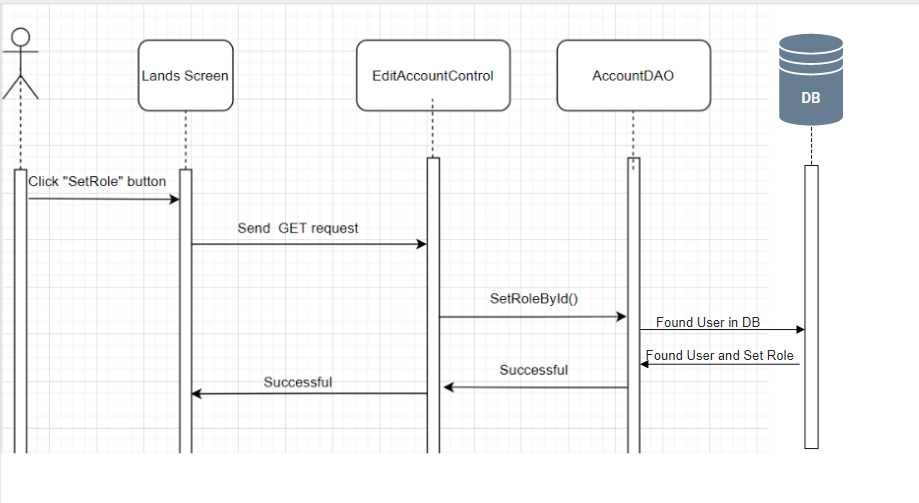
BlogDAO

[Provide the detailed description for the class methods]

|  |  |  |
| --- | --- | --- |
| No | Method | Description |
| 01 | Create User | Insert new user to database |
| 02 | Validate user | Check user if exists |

### 

### c. Sequence Diagram(s)



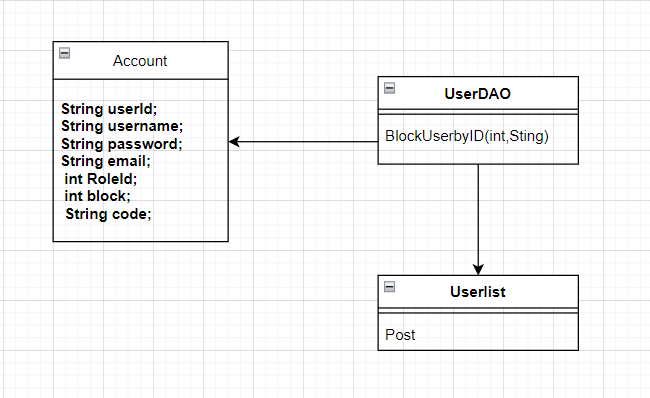
### 

### d. Database queries

Select \* from Users

## 16. Block User

### a. Class Diagram



### b. Class Specifications

Userlist

|  |  |  |
| --- | --- | --- |
| No | Method | Description |
| 01 | Post() | get UserID and change isBlock on Users |

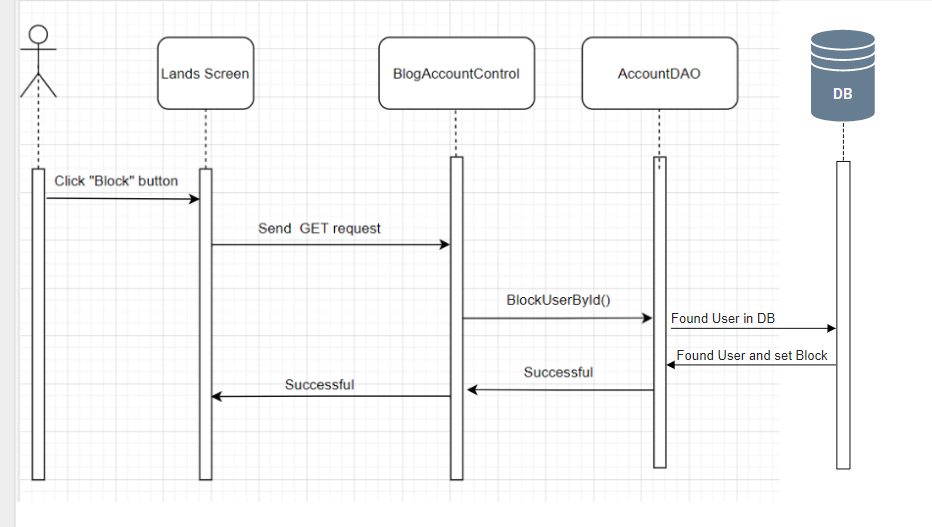
UserDAO

[Provide the detailed description for the class methods]

|  |  |  |
| --- | --- | --- |
| No | Method | Description |
| 01 | BlockUserbyID | Change isBlock user on database |

### 

### c. Sequence Diagram(s)



### 

### d. Database queries

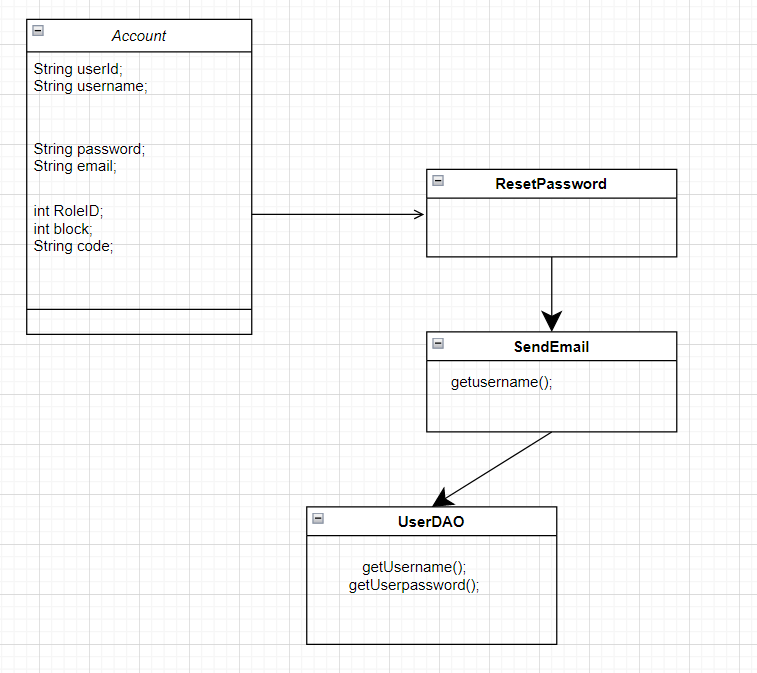
Update users

set isBlock = ?

where UserID = ?

## 17. Reset Password

### Class Diagram



### Class Specifications

### 

UserController

|  |  |  |
| --- | --- | --- |
| No | Method | Description |
| 01 | processRequest() |  |

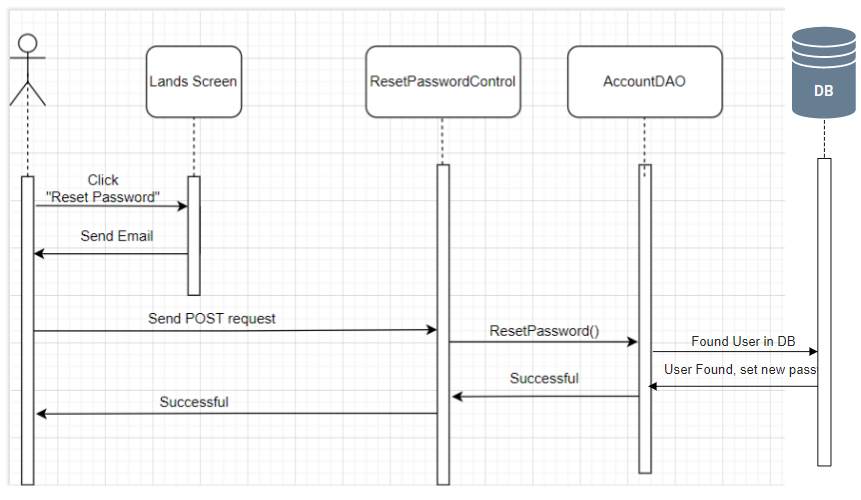
BlogDAO

[Provide the detailed description for the class methods]

|  |  |  |
| --- | --- | --- |
| No | Method | Description |
| 01 | Create User | Insert new user to database |
| 02 | Validate user | Check user if exists |

### 

### c. Sequence Diagram(s)



### 

### d. Database queries

Select \* from Users

## 18. Change Password

### a. Class Diagram

### b. Class Specifications

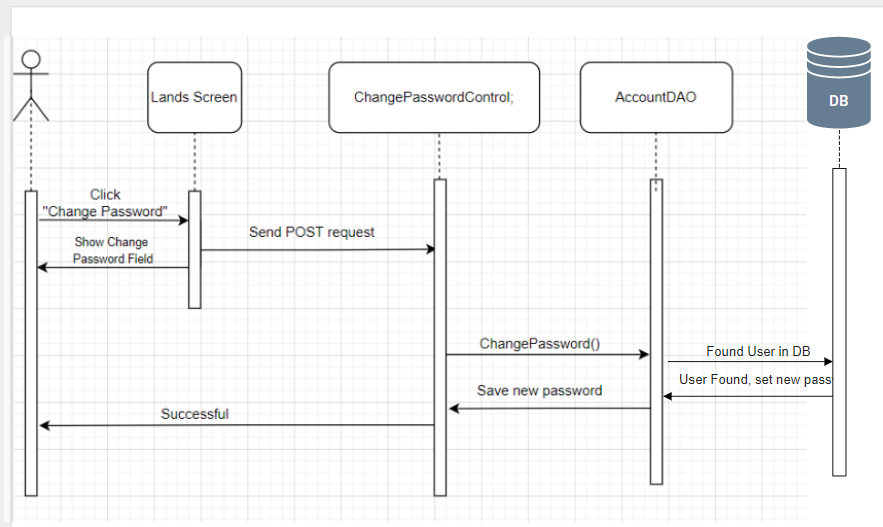
UserDAO

|  |  |  |
| --- | --- | --- |
| No | Method | Description |
| 01 | processRequest() |  |

|  |  |  |
| --- | --- | --- |
| No | Method | Description |
| 01 | Create User | Insert new user to database |
| 02 | Validate user | Check user if exists |

### 

### c. Sequence Diagram(s)



### 

### d. Database queries

## 19. User profile

### a. Class Diagram

### b. Class Specifications

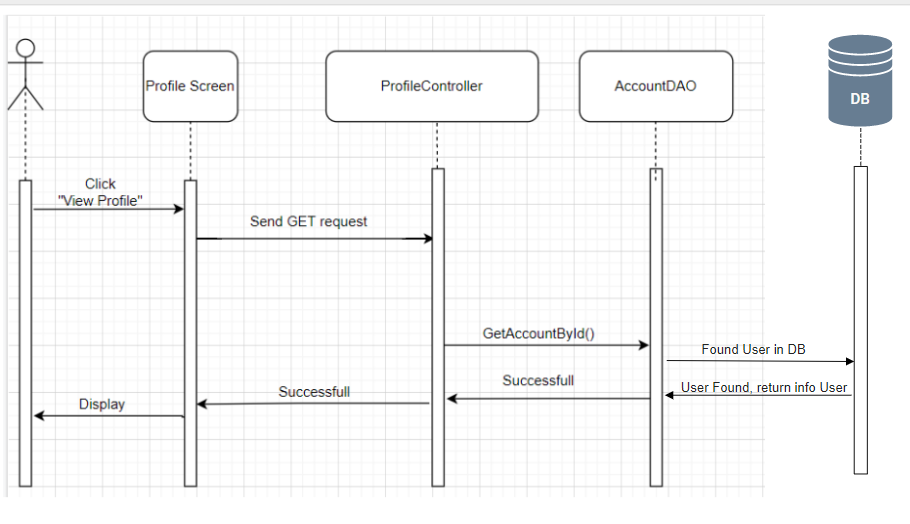
UserController

|  |  |  |
| --- | --- | --- |
| No | Method | Description |
| 01 | processRequest() |  |

|  |  |  |
| --- | --- | --- |
| No | Method | Description |
| 01 | Create User | Insert new user to database |
| 02 | Validate user | Check user if exists |

### 

### c. Sequence Diagram(s)



### 

### d. Database queries

Select \* from Users

## 20. Order History

### a. Class Diagram

### b. Class Specifications

UserController

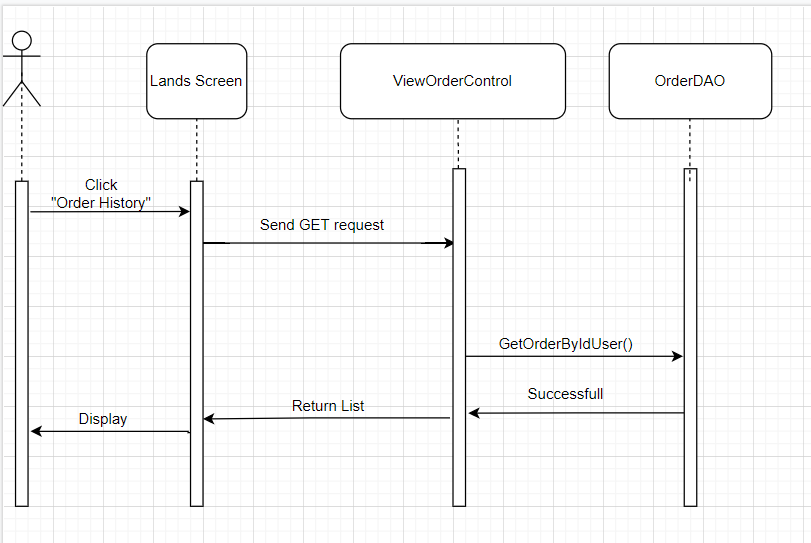
|  |  |  |
| --- | --- | --- |
| No | Method | Description |
| 01 | processRequest() |  |

OrderDAO

|  |  |  |
| --- | --- | --- |
| No | Method | Description |
| 01 | viewOrderWait(String userId) | See the product that userId ordered is in the waiting state |
| 02 |  | Check user if exists |
|  |  |  |

### 

### c. Sequence Diagram(s)



### 

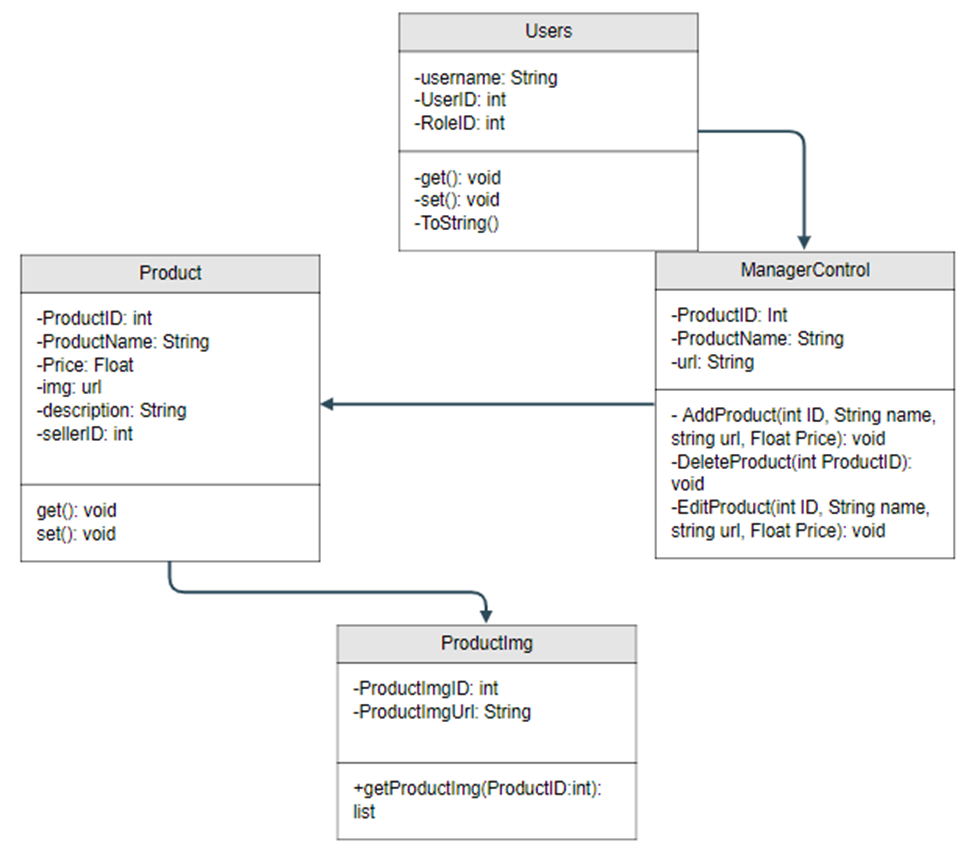
### d. Database queries

Select \* from Users

,

## 21. Delete Product

### a. Class Diagram



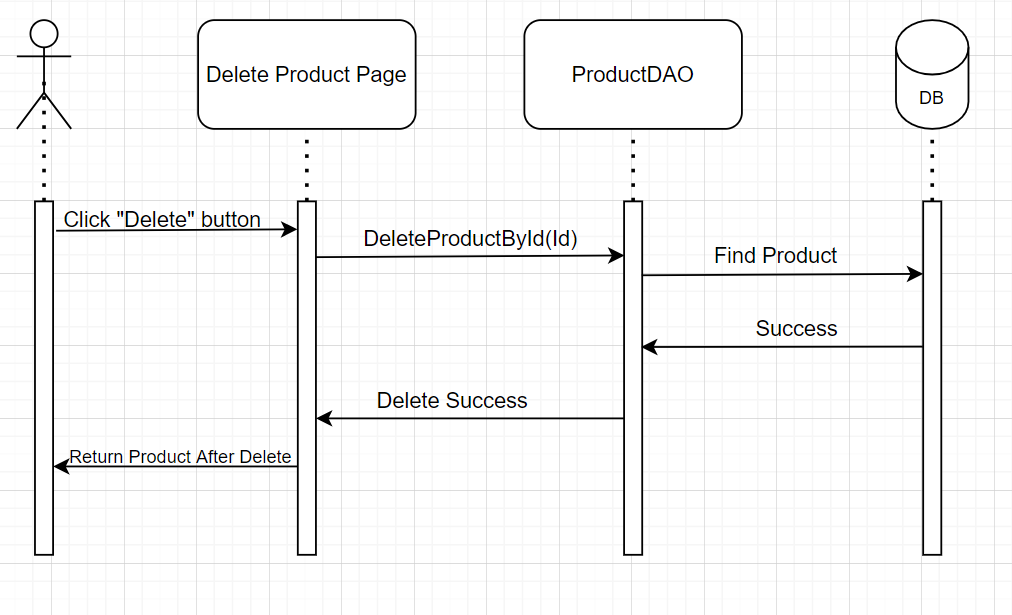
### b. Class Specifications

ProductDAO

|  |  |  |
| --- | --- | --- |
| No | Method | Description |
| 01 | deleteProduct() | *DeleteProduct from the database to display on ManageProduct.jsp* |

### 

### c. Sequence Diagram(s)

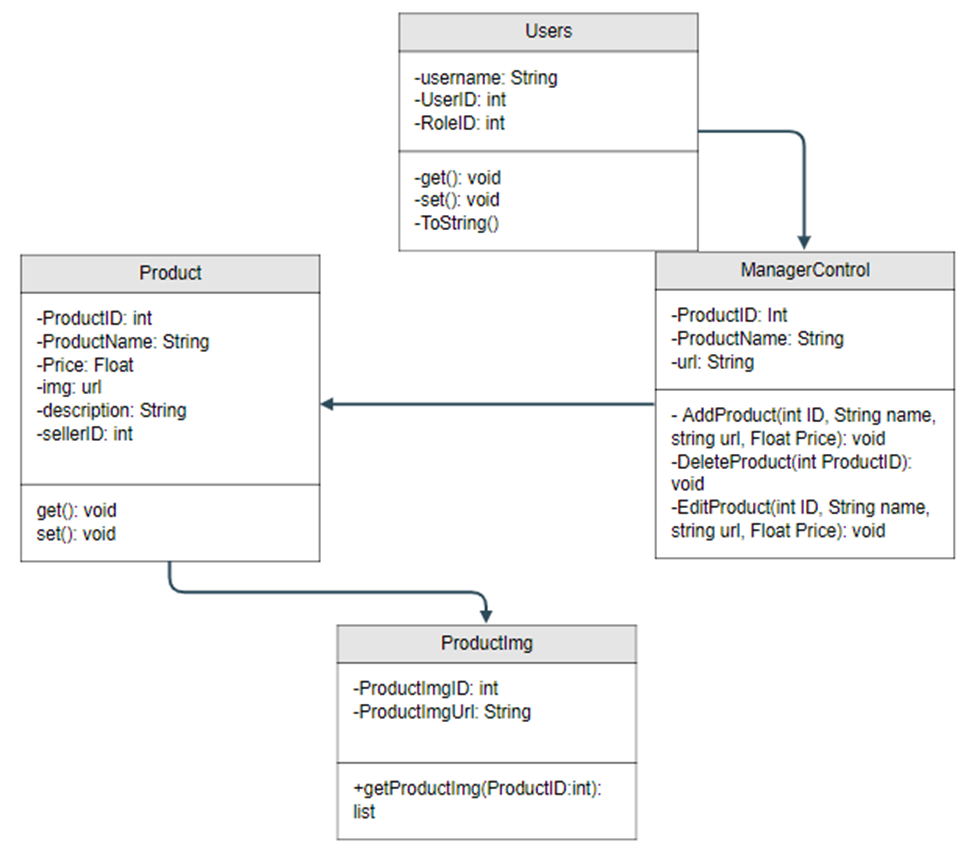


### d. Database queries

delete from Product where ProductID = ?

## 22. Edit Product

### a. Class Diagram



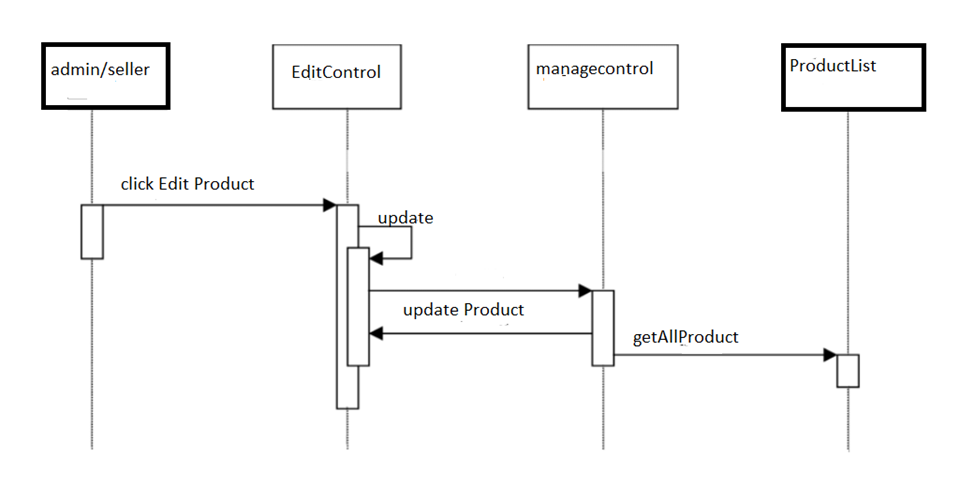
### b. Class Specifications

ProductDAO

|  |  |  |
| --- | --- | --- |
| No | Method | Description |
| 01 | editProduct() | *Update from the database to display on ManageProduct.jsp* |

### 

### c. Sequence Diagram(s)



### 

### d. Database queries

UPDATE datatable

SET ... = ?

WHERE …... = ?

## 23. Checkout

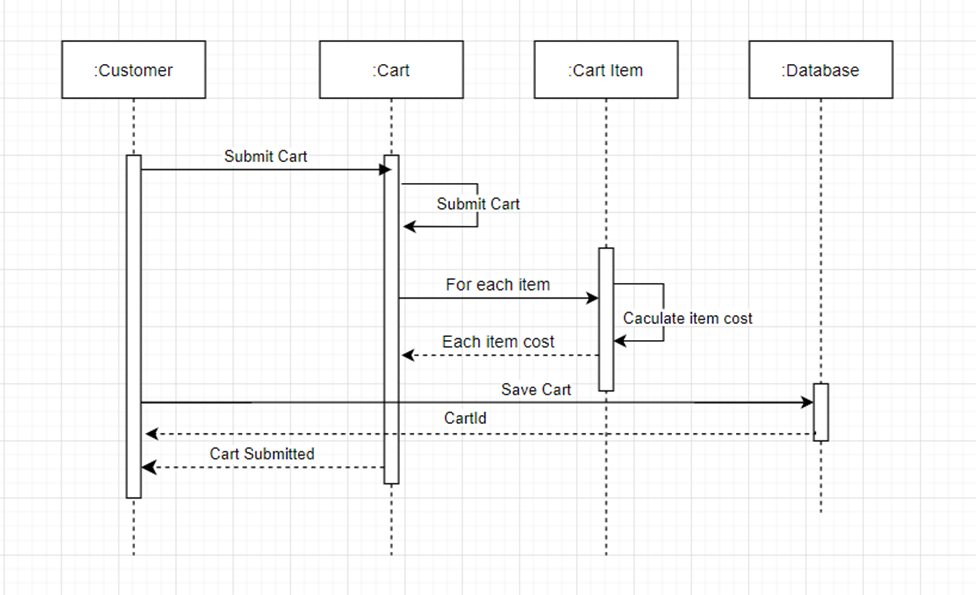
### a. Class Diagram

### b. Class Specifications

|  |  |  |
| --- | --- | --- |
| No | Method | Description |
| 01 | getProductById() | *Get Product which confirm to buy* |
| 02 | saveOrder() | *Insert information in Order table* |
| 03 | SaveOderDetail() | *Insert information in OrderDetail table* |
| 04 | SaveShipping() | *Insert information in Shipping table* |

### 

### c. Sequence Diagram(s)



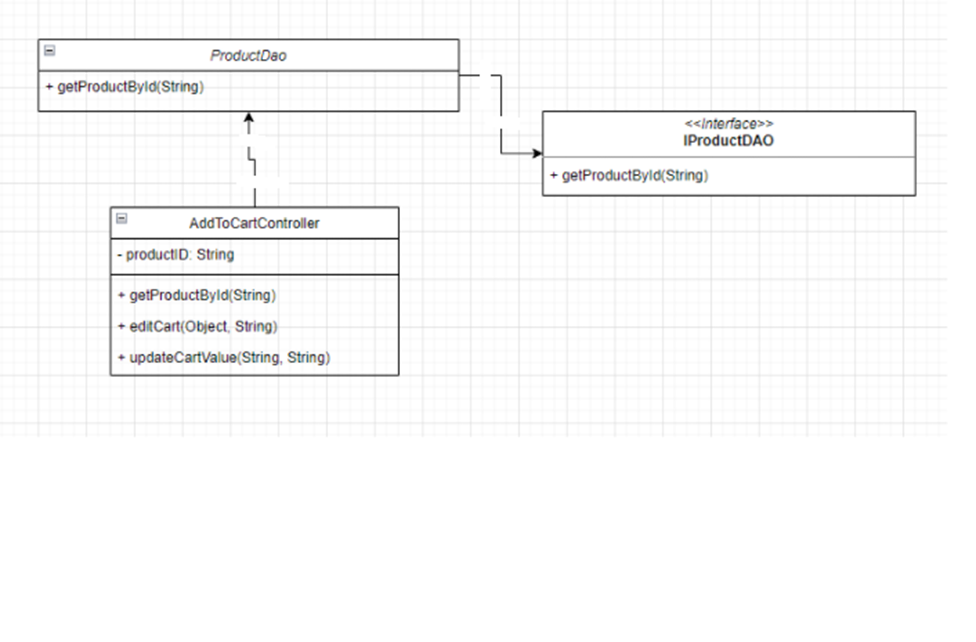
### d. Database queries

INSERT INTO [dbo].[Orders]

([UserID]

,[TotalPrice]

,[Note]

 ,[Status]

,[Date]

,[ShippingID])

VALUES

(?,?,?,?,GETDATE(),?)

## 24.Manager Product

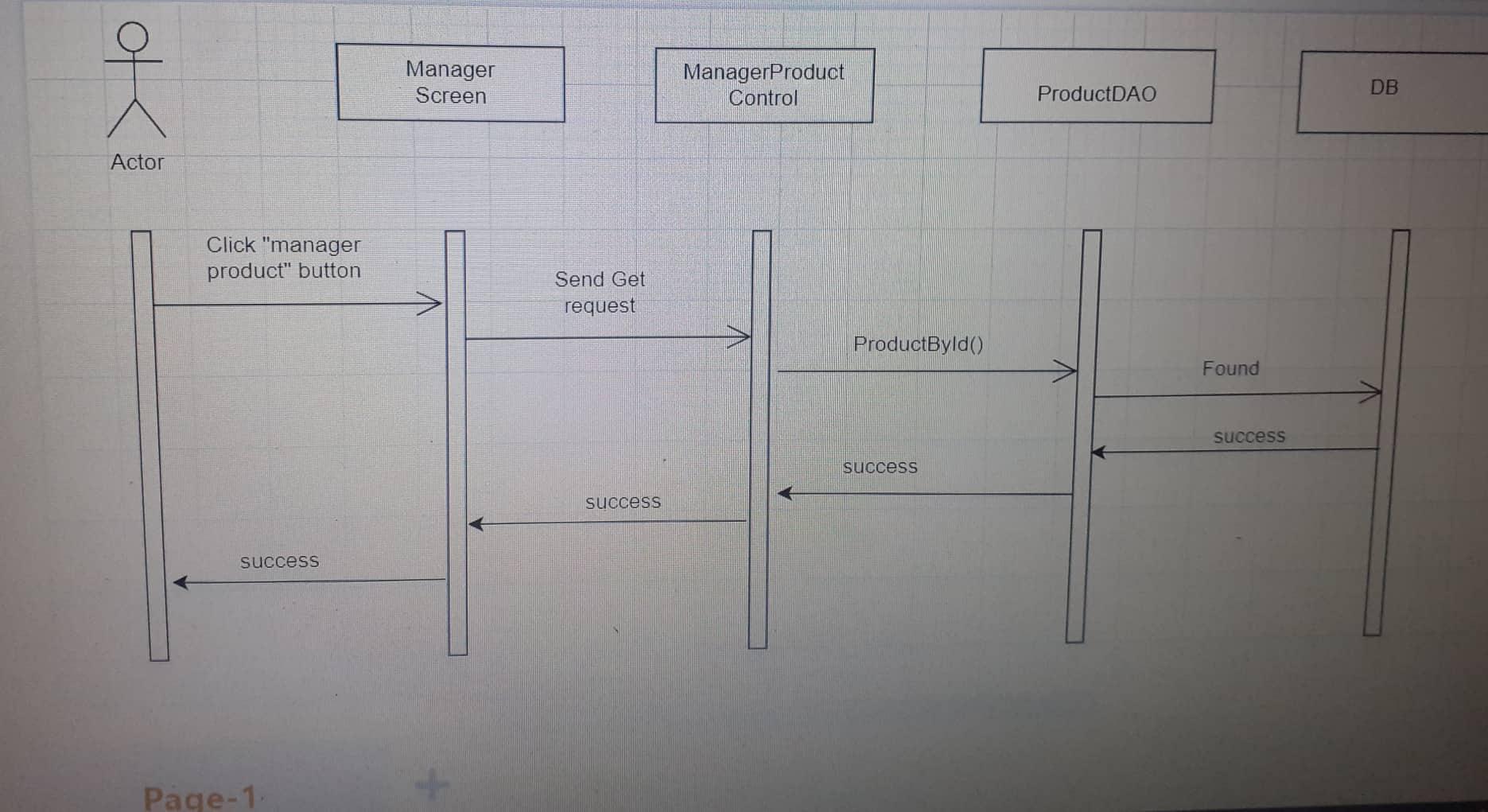
### Class diagram

### 

### Class specifications

|  |  |  |
| --- | --- | --- |
| No | Method | Description |
| 01 | processRequest() | View and manage products |

### Sequence Diagram(s)



### Database queries

select p.\*,pr.ProductImgURL from Product p join ProductImg pr

+ "on p.ProductID = pr.ProductID

## 25.HomePage

### Class Diagram

### 

### 

### Class Specifications

ProductDAO

|  |  |  |
| --- | --- | --- |
| No | Method | Description |
| 01  02 | getAll  pagingProductBy6 | *Get all product*  *product pagination 6 products 1 page* |

SubCategoryDAO

|  |  |  |
| --- | --- | --- |
| No | Method | Description |
| 01 | getAllSubCategory | *Get all SubCategory* |

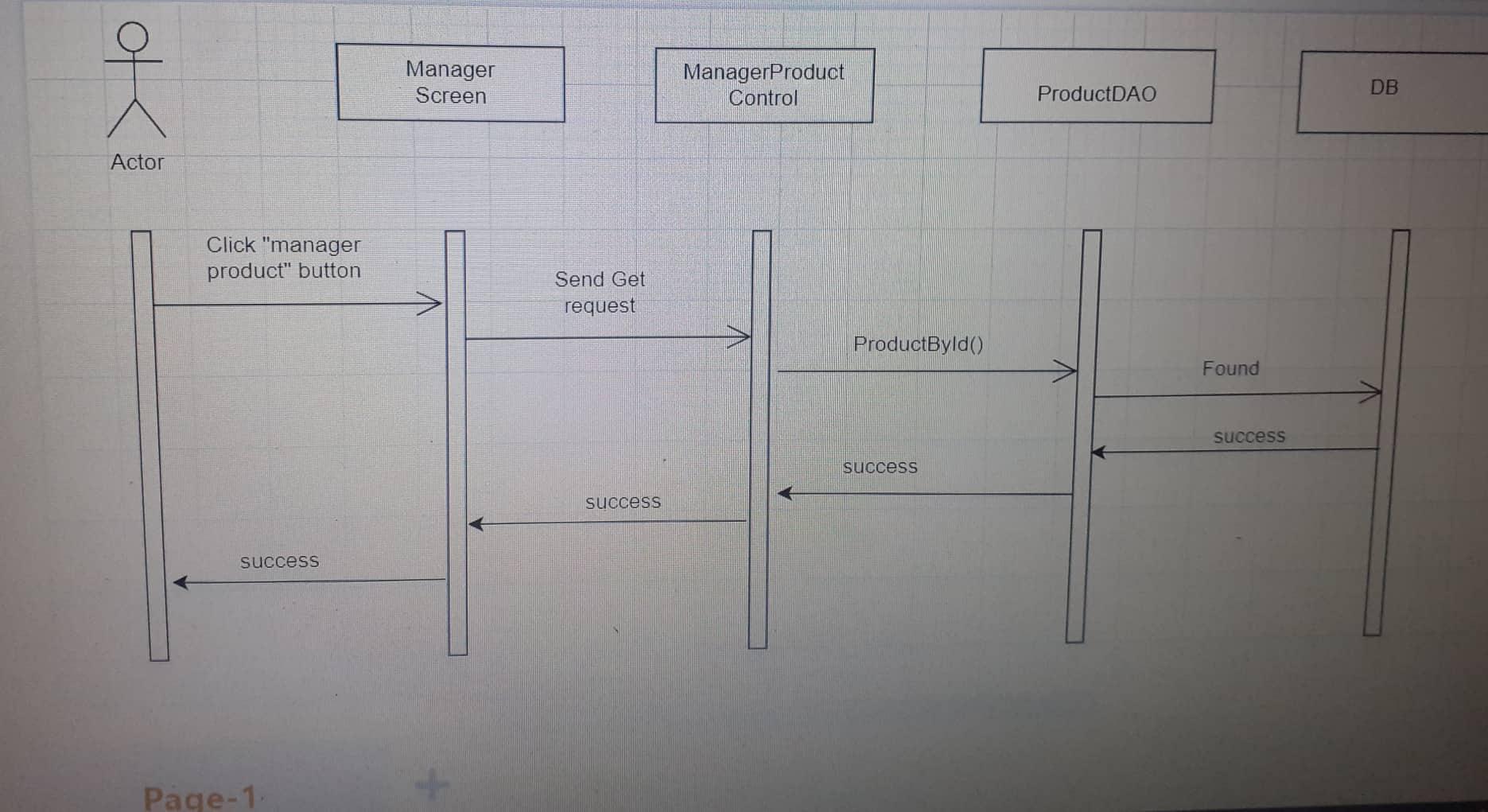
CategoryDAO

|  |  |  |
| --- | --- | --- |
| No | Method | Description |
| 01 | getAllCategory | *Get all Category* |

BrandDAO

|  |  |  |
| --- | --- | --- |
| No | Method | Description |
| 01 | getAllBrand | *Get all brand* |

### Sequence Diagram



1. Database queries

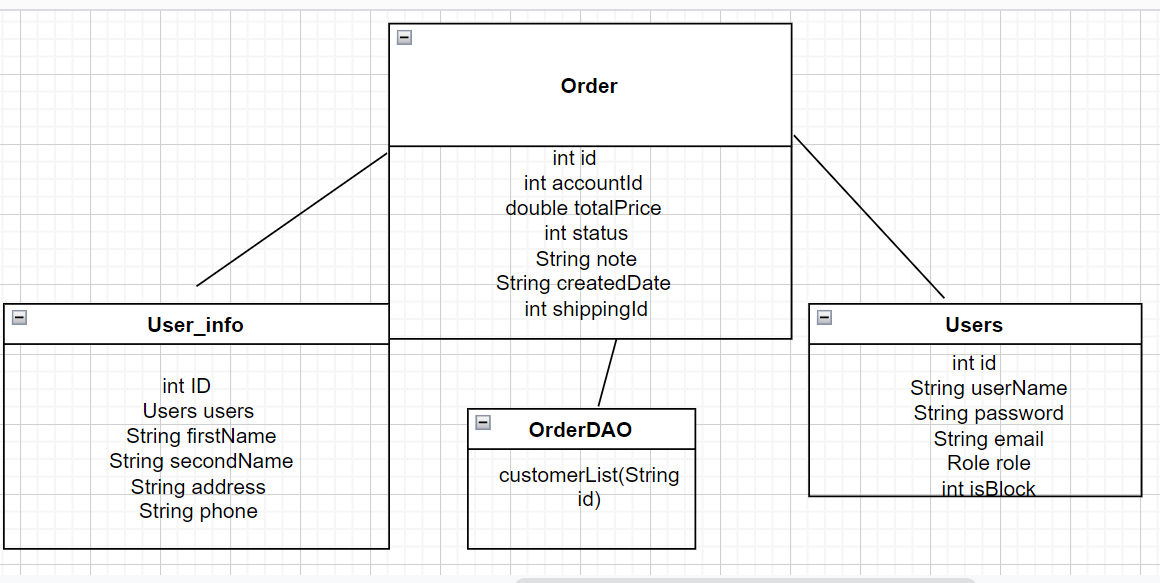
### d. Database queries

select p.\*,pr.ProductImgURL from Product p join ProductImg pr

+ "on p.ProductID = pr.ProductID

## 26. List Customer

### Class Diagram



### Class Specification

|  |  |  |
| --- | --- | --- |
| No | Method | Description |
| 01 | processRequest() | Review a list of past customers. |

### Sequence Diagram(s)

### 

### Database queries

select \* from Orders join Users

on Orders.UserID = Users.UserID

join User\_info as c

on Users.UserID = c.UserID

## 26. Product list

### a.Class diagram

### 

### 

### 

### b.Class specifications

### 

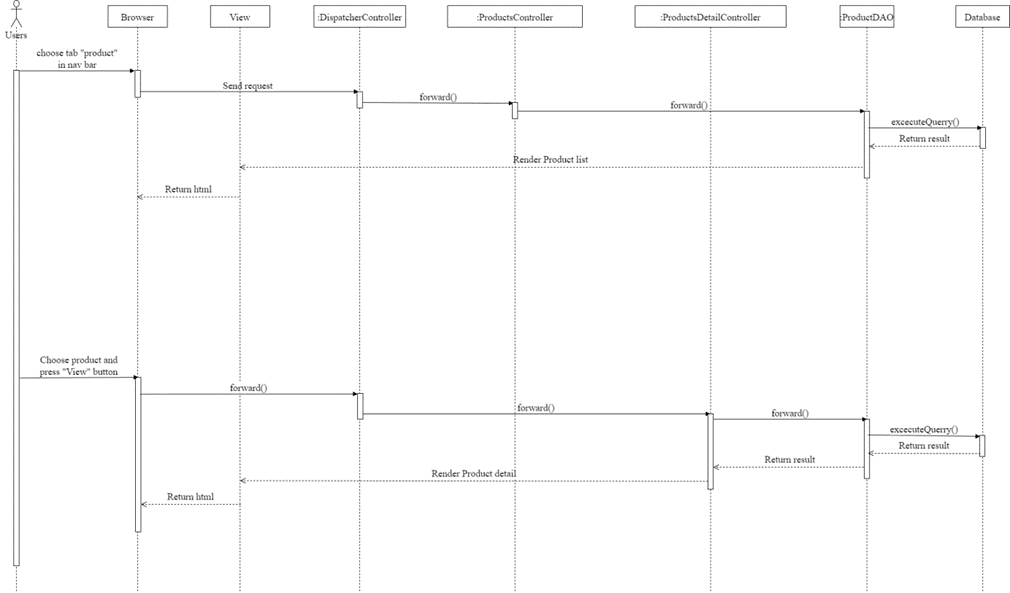
UserDAO

|  |  |  |
| --- | --- | --- |
| No | Method | Description |
| 01 | login | *Input correct username and password to login in to web* |

ProductDAO

|  |  |  |
| --- | --- | --- |
| **No** | **Method** | **Description** |
| *01* | *View Product* | *View detail of one specific post* |
| *02* | *Productlist* | *View list all post* |
| *03* | *addtoCart* | *Add product to cart* |

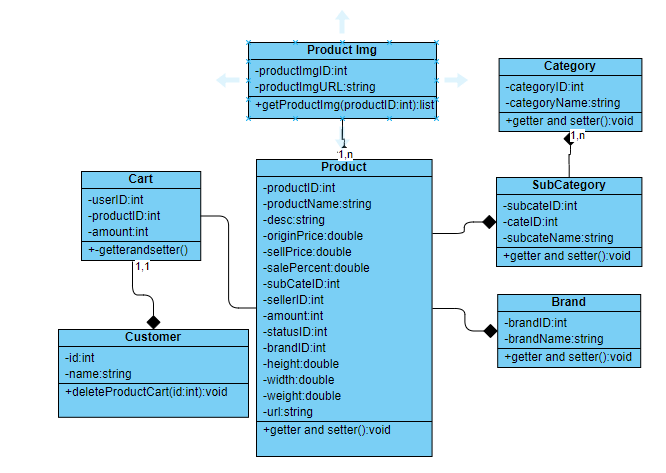
### c. Sequence Diagram(s)



### d. Database queries

## 27. Delete Product From Cart

### a. Class Diagram

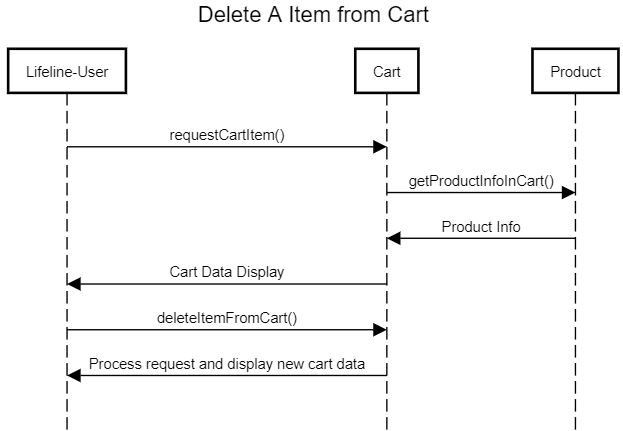


### b. Class Specifications

Customer

|  |  |  |
| --- | --- | --- |
| **No** | **Method** | **Description** |
| *01* | *deleteProductCart()* | *Remove a product() from cart* |

### c. Sequence Diagram(s)

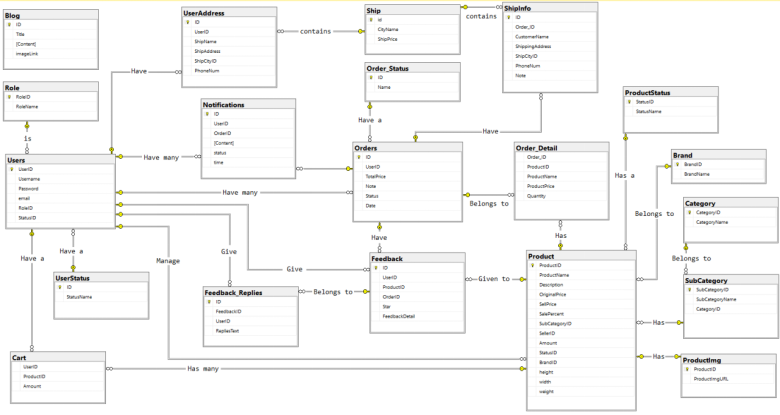


### d. Database queries

DELETE FROM *dbo.cart* WHERE *userID= ? and* productID = ?

# III. Database Design

## 1. Database Schema

**

## 2. Table Description

|  |  |  |
| --- | --- | --- |
| **No** | **Table** | **Description** |
| *01* | *Blog* | *Blog content + image*  *- Primary keys: ID*  *- Foreign keys:* |
| *02* | *Role* | *Role of a user ( Admin,Marketing,Sale,Customer )*  *- Primary keys: RoleID*  *- Foreign keys:* |
| *03* | *UserStatus* | *Account activation status*  *- Primary keys: ID*  *- Foreign keys:* |
| *04* | *Ship* | *Shipping price for each city*  *- Primary keys: id*  *- Foreign keys:* |
| *05* | *Order\_Detail* | *Detail of each product in order*  *- Primary keys:*  *- Foreign keys: OrderID, ProductID* |
| *06* | *UserAddress* | *Address for each user*  *- Primary keys:ID*  *- Foreign keys: UserID, ShipCityID* |
| *07* | *Users* | *Users*  *- Primary keys: UserID*  *- Foreign keys: RoleID,StatusID* |
| *08* | *Notification* | *Notification data*  *- Primary keys: ID*  *- Foreign keys: OrderID, UserID* |
| *09* | *ShipInfo* | *Shipping info for each order*  *- Primary keys: ID*  *- Foreign keys: Order\_ID, ShipCityID* |
| *10* | *Order* | *Order data*  *- Primary keys: ID*  *- Foreign keys: UserID,Status* |
| *11* | *Feedback* | *Feedback of a product order*  *- Primary keys: ID*  *- Foreign keys: UserID,ProductID, OrderID* |
| *12* | *Feedback\_Replies* | *Replies of feedbacks*  *- Primary keys: id*  *- Foreign keys: FeedbackID,UserID* |
| *13* | *Order\_Status* | *Status of a order*  *- Primary keys: ID*  *- Foreign keys:* |
| *14* | *Product* | *Product Data*  *- Primary keys: ProductID*  *- Foreign keys: SubCategoryID,SellerID,StatusID,BrandID* |
| *15* | *Category* | *Category Data*  *- Primary keys: CategoryID*  *- Foreign keys:* |
| *16* | *SubCategory* | *Sub Category ID*  *- Primary keys: SubCategoryID*  *- Foreign keys: CategoryID* |
| *17* | *Productimg* | *Image of products*  *- Primary keys:*  *- Foreign keys: ProductID* |
| *18* | *ProductStatus* | *Status of a product*  *- Primary keys: StatusID*  *- Foreign keys: StatusName* |
| *19* | *Brand* | *Brand data*  *- Primary keys: BrandID*  *- Foreign keys:* |
| *20* | *Cart* | *Cart info for each user*  *- Primary keys: UserID*  *- Foreign keys: ProductID* |