

**CLOTHING SHOP PROJECT REPORT**

**Software Design Document**

– Hanoi, Summer 2022 –

**Table of Contents**

[I. Overview 6](#_Toc20740)

[1. Code Packages/Namespaces 6](#_Toc5140)

[2. Coding Conventions 7](#_Toc27083)

[II. Code Designs 7](#_Toc1187)

[1. <Feature/Function Name1> 8](#_Toc13845)

[2. Login 8](#_Toc23040)

[a. Class Diagram 8](#_Toc19176)

[b. Class Specifications 8](#_Toc4584)

[c. Sequence Diagram(s) 9](#_Toc8368)

[d. Database queries 9](#_Toc30251)

[4. Product detail 10](#_Toc10417)

[a. Class Diagram 10](#_Toc3821)

[b. Class Specifications 10](#_Toc24256)

[c. Sequence Diagram(s) 10](#_Toc13287)

[d. Database queries 11](#_Toc7346)

[5. Register 11](#_Toc13454)

[a. Class Diagram 11](#_Toc4193)

[b. Class Specifications 12](#_Toc286)

[c. Sequence Diagram(s) 12](#_Toc5879)

[d. Database queries 13](#_Toc25465)

[6. Cart Detail 13](#_Toc20796)

[a. Class Diagram 13](#_Toc9135)

[b. Class Specifications 13](#_Toc22037)

[c. Sequence Diagram(s) 14](#_Toc2769)

[d. Database queries 14](#_Toc27970)

[7. Blog List 14](#_Toc22194)

[a. Class Diagram 14](#_Toc28686)

[b. Class Specifications 14](#_Toc19236)

[c. Sequence Diagram(s) 15](#_Toc3024)

[d. Database queries 16](#_Toc15794)

[8. Blog Detail 16](#_Toc20304)

[a. Class Diagram 16](#_Toc23514)

[b. Class Specifications 16](#_Toc3200)

[c. Sequence Diagram(s) 17](#_Toc26736)

[d. Database queries 17](#_Toc30952)

[9. Logout 17](#_Toc20697)

[a. Class Diagram 17](#_Toc2668)

[b.Class Specifications 18](#_Toc16687)

[c. Sequence Diagram(s) 18](#_Toc15197)

[d. Database queries 19](#_Toc22173)

[10. Home Page 19](#_Toc30705)

[a. Class Diagram 19](#_Toc18793)

[b. Class Specifications 19](#_Toc18580)

[c. Sequence Diagram(s) 20](#_Toc29835)

[11. Change amount 21](#_Toc10163)

[a. Class Diagram 21](#_Toc3778)

[b. Class Specifications 21](#_Toc9969)

[c. Sequence Diagram(s) 22](#_Toc18818)

[12. Delete cart 22](#_Toc26337)

[a. Class Diagram 22](#_Toc27706)

[b. Class Specifications 22](#_Toc15807)

[c. Sequence Diagram(s) 23](#_Toc30388)

[13. View List User 24](#_Toc3259)

[a. Class Diagram 24](#_Toc27558)

[b. Class Specifications 24](#_Toc210)

[c. Sequence Diagram(s) 25](#_Toc11444)

[d. Database queries 26](#_Toc9041)

[14. Edit Information User 26](#_Toc23393)

[a. Class Diagram 26](#_Toc32618)

[b. Class Specifications 26](#_Toc3791)

[c. Sequence Diagram(s) 27](#_Toc3469)

[d. Database queries 28](#_Toc27711)

[15.Set Role User 28](#_Toc23131)

[a. Class Diagram 28](#_Toc24736)

[b. Class Specifications 28](#_Toc20083)

[c. Sequence Diagram(s) 29](#_Toc27586)

[d. Database queries 29](#_Toc25963)

[16. Block User 30](#_Toc12767)

[a. Class Diagram 30](#_Toc21928)

[b. Class Specifications 30](#_Toc28305)

[c. Sequence Diagram(s) 30](#_Toc30716)

[d. Database queries 31](#_Toc31866)

[17. Reset Password 31](#_Toc30001)

[a. Class Diagram 31](#_Toc32549)

[b. Class Specifications 31](#_Toc12523)

[c. Sequence Diagram(s) 32](#_Toc26855)

[d. Database queries 33](#_Toc25726)

[18. Change Password 33](#_Toc9414)

[a. Class Diagram 33](#_Toc12727)

[b. Class Specifications 33](#_Toc11448)

[c. Sequence Diagram(s) 34](#_Toc29863)

[d. Database queries 35](#_Toc22198)

[19. User profile 35](#_Toc5350)

[a. Class Diagram 35](#_Toc14192)

[b. Class Specifications 35](#_Toc23029)

[c. Sequence Diagram(s) 36](#_Toc12204)

[d. Database queries 37](#_Toc9124)

[20. Order History 37](#_Toc8597)

[a. Class Diagram 37](#_Toc11940)

[b. Class Specifications 37](#_Toc30083)

[c. Sequence Diagram(s) 38](#_Toc3758)

[d. Database queries 39](#_Toc4614)

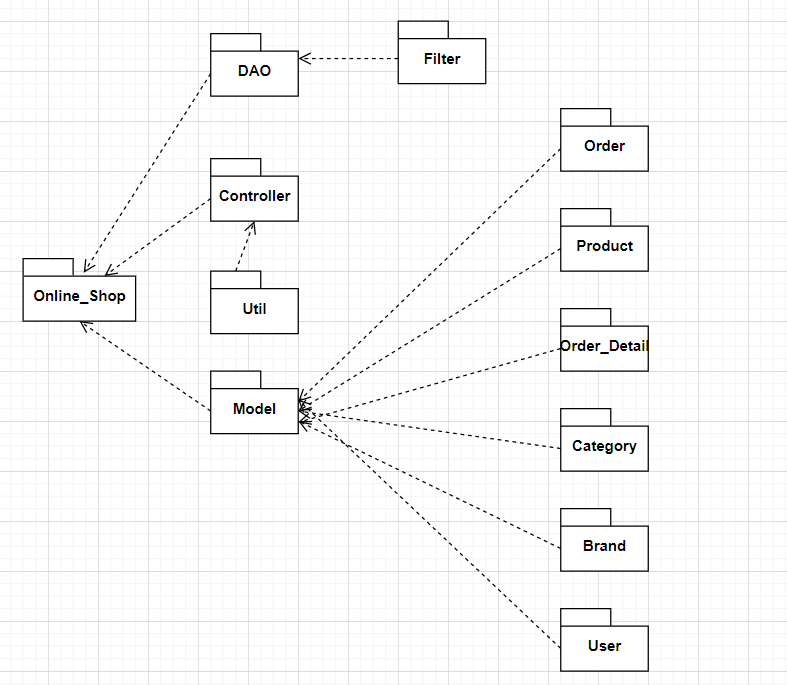
[III. Database Design 40](#_Toc4839)

[1. Database Schema 40](#_Toc18740)

[2. Table Description 40](#_Toc11886)

# 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 |

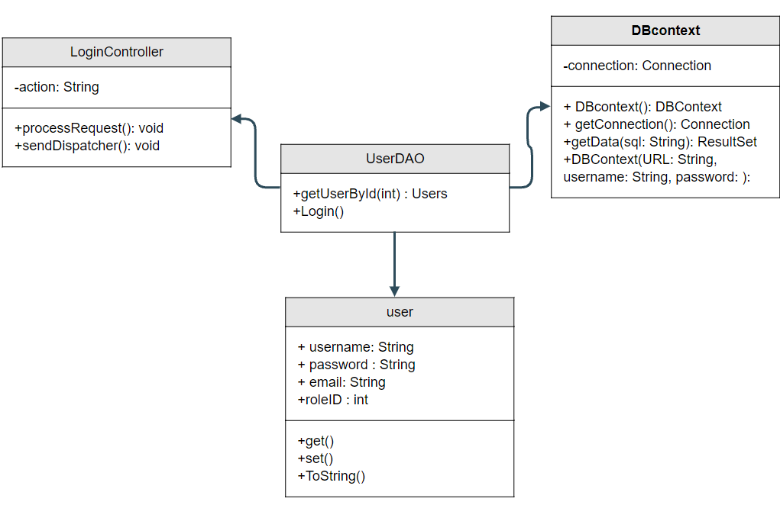
## 2. Coding Conventions

# II. Code Designs

## 1. <Feature/Function Name1>

## 2. Login

### Class Diagram



### Class Specifications

UserController

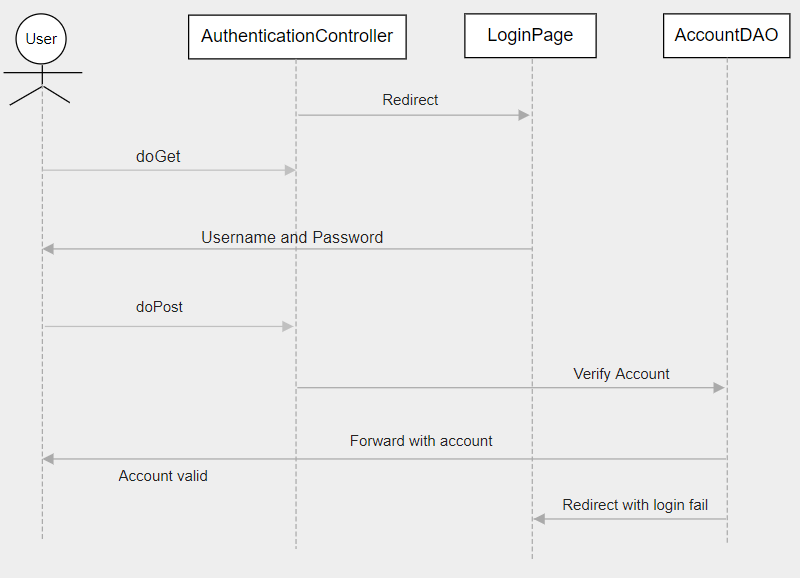
|  |  |  |
| --- | --- | --- |
| No | Method | Description |
| 01 | processRequest() | get all new information from user Form and update the database |

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=?"**

## 4. Product detail

### Class Diagram

### Class Specifications

UserController

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

UserDAO

[Provide the detailed description for the class methods]

|  |  |  |
| --- | --- | --- |
| No | Method | Description |
| 01 |  |  |
| 02 |  |  |

### 

### c. Sequence Diagram(s)

### 

### d. Database queries

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

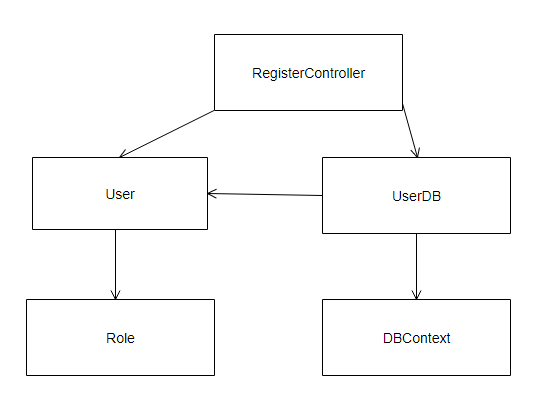
+ "on p.SubCategoryID = s.SubCategoryID\n"

+ "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 | Create User | Insert new user to database |
| 02 | Validate user | Check user if exists |

### 

### c. Sequence Diagram(s)

### 

### d. Database queries

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

([Username

,[Password]

,[email]

,[RoleID]

,[isBlock])

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

## 6. Cart Detail

### Class Diagram

### Class Specifications

UserController

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

UserDAO

[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

## Blog List

### Class Diagram

### Class Specifications

UserController

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

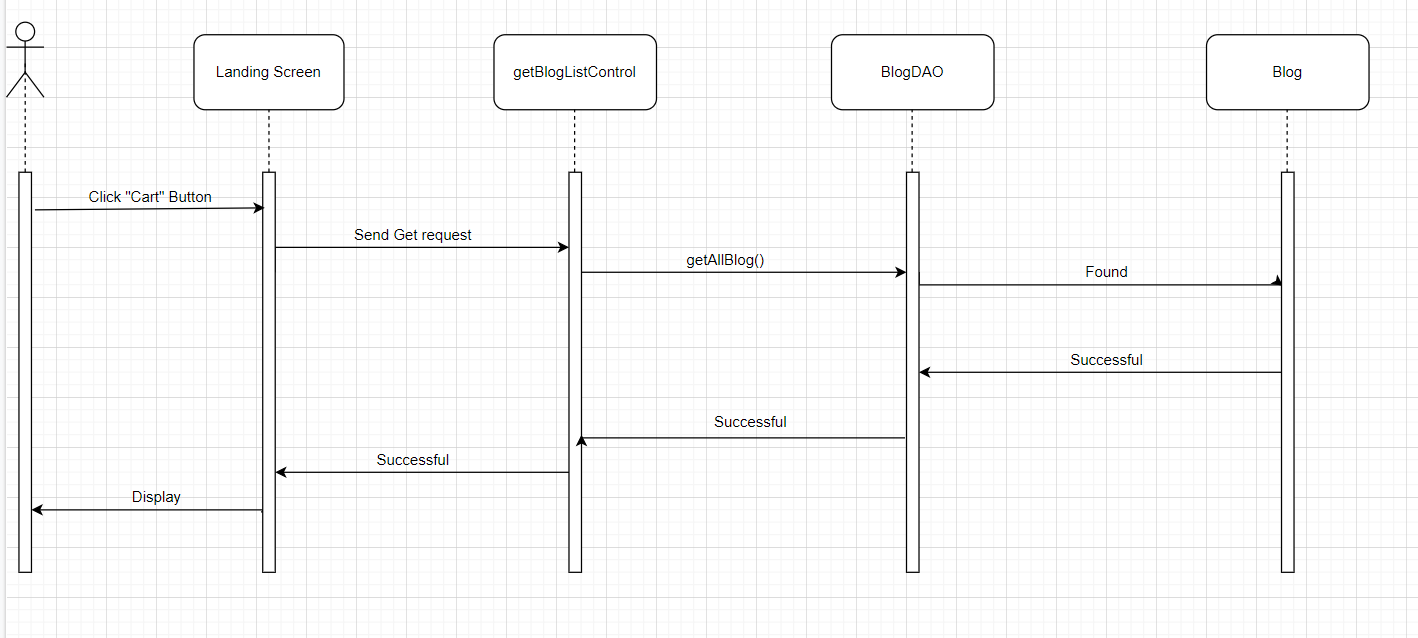
UserDAO

[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

## 8. Blog Detail

### Class Diagram

### Class Specifications

UserController

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

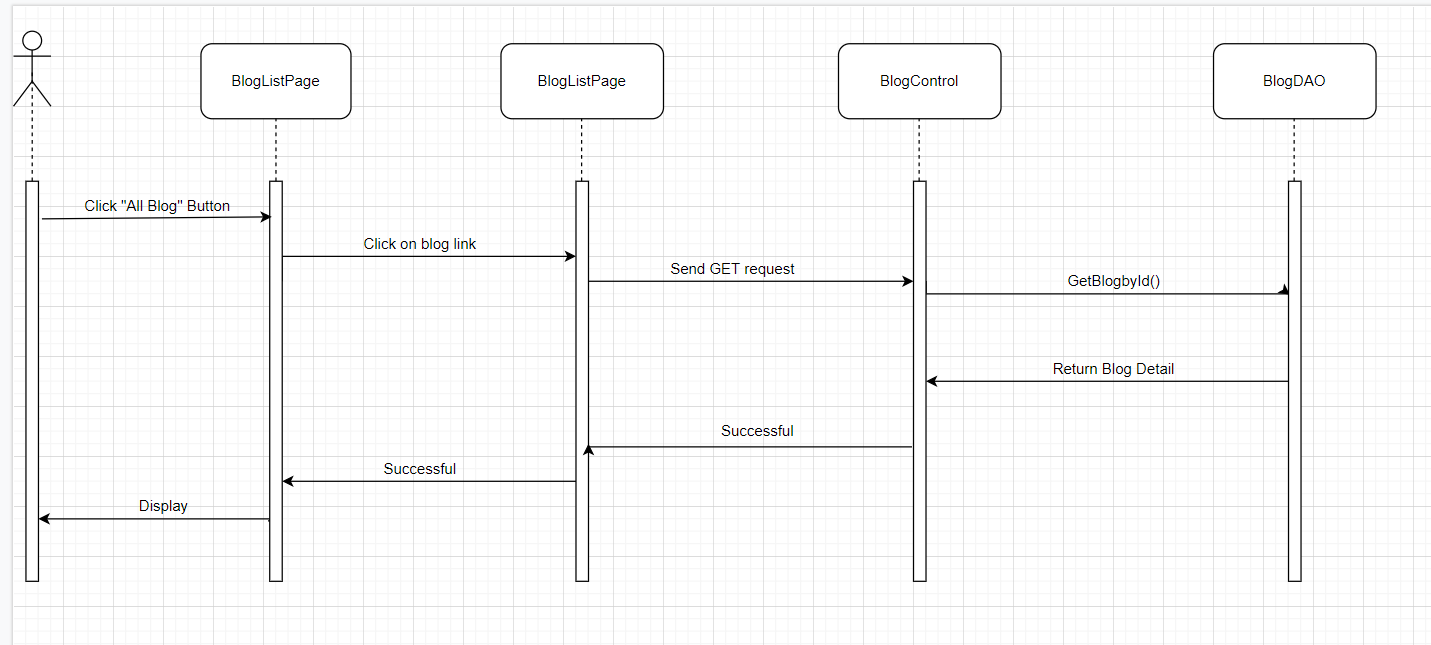
BlogDAO

[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 Blog WHERE ID = ?

## 9. Logout

### a. Class Diagram

### b.Class Specifications

UserController

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

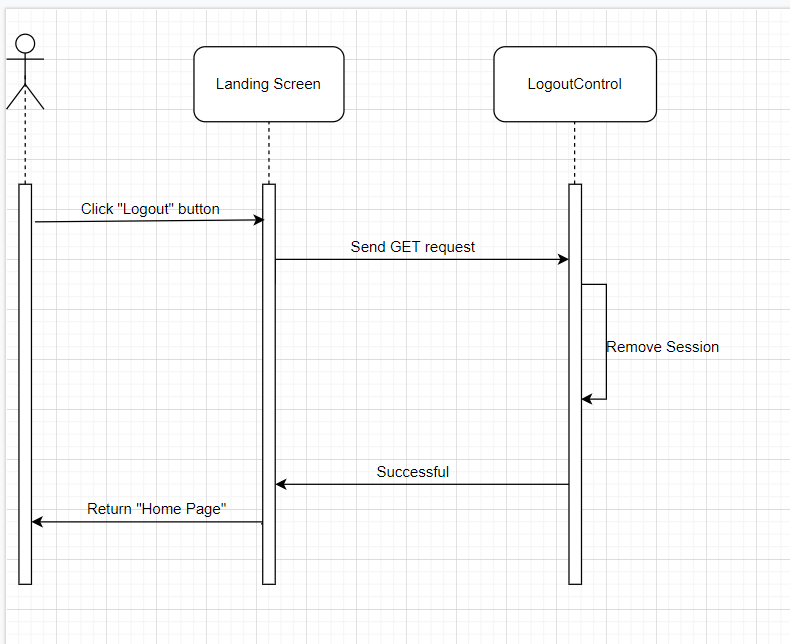
BlogDAO

[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

## 10. Home Page

### a. Class Diagram

### b. Class Specifications

UserController

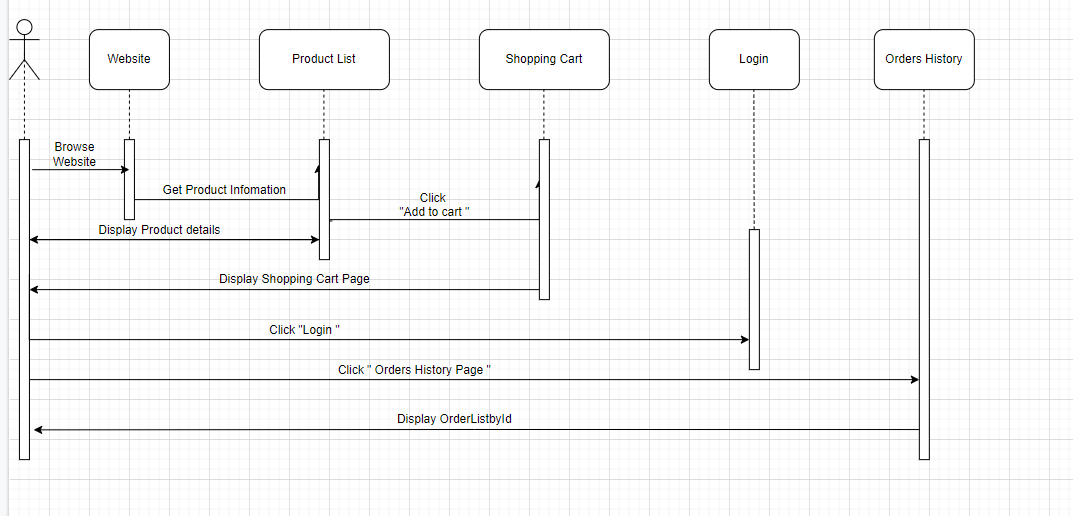
|  |  |  |
| --- | --- | --- |
| No | Method | Description |
| 01 | processRequest() |  |

[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)



## 

## 11. Change amount

### Class Diagram

### Class Specifications

### 

UserController

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

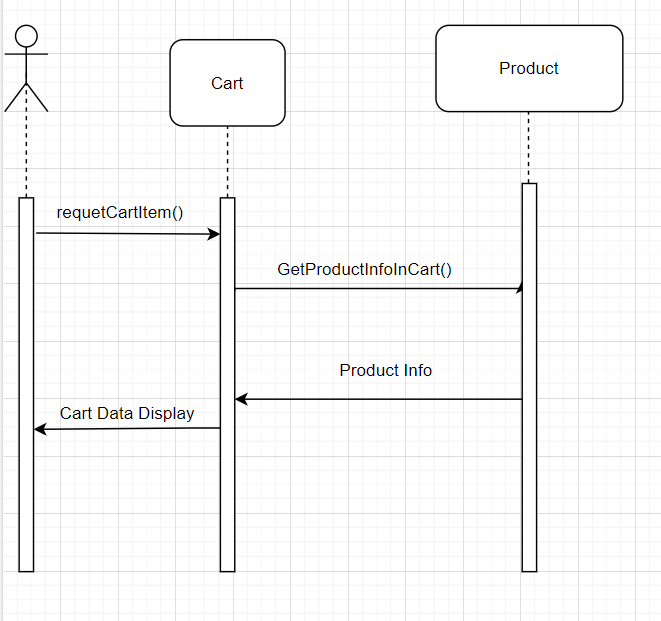
BlogDAO

[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)



### 

## 12. Delete cart

### a. Class Diagram

### b. Class Specifications

UserController

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

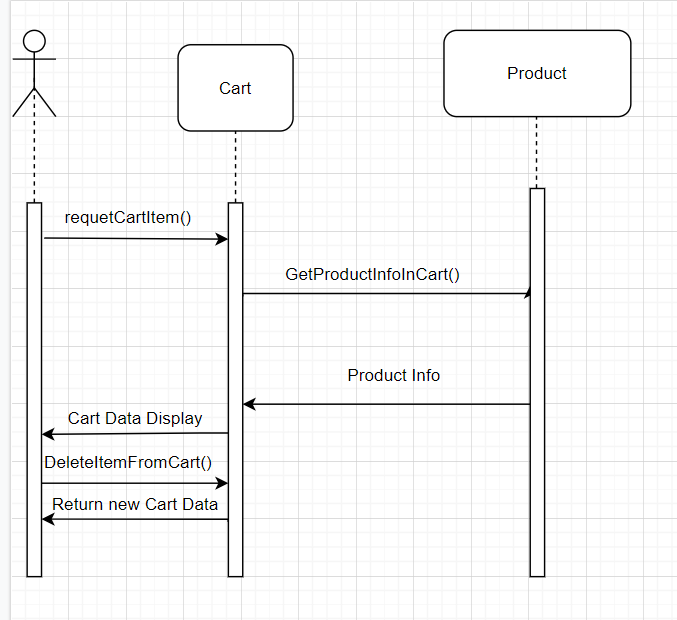
BlogDAO

[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)



### 

## 13. View List User

### a. Class Diagram

### b. Class Specifications

UserController

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

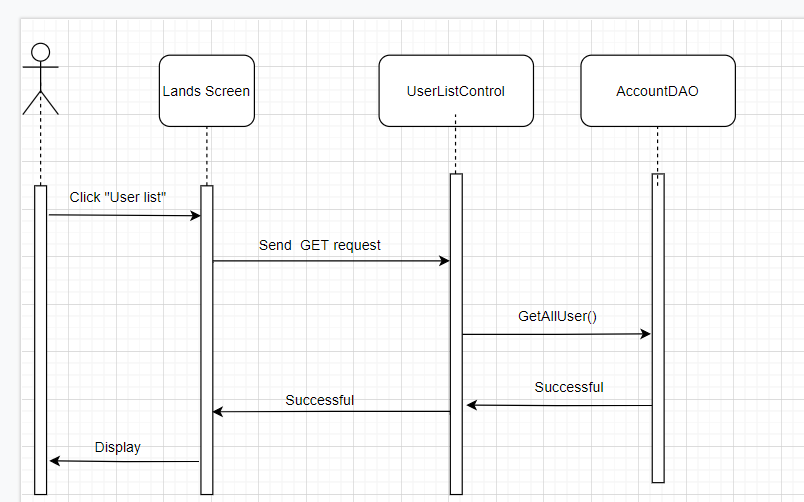
BlogDAO

[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

## 14. Edit Information User

### a. Class Diagram

### b. Class Specifications

UserController

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

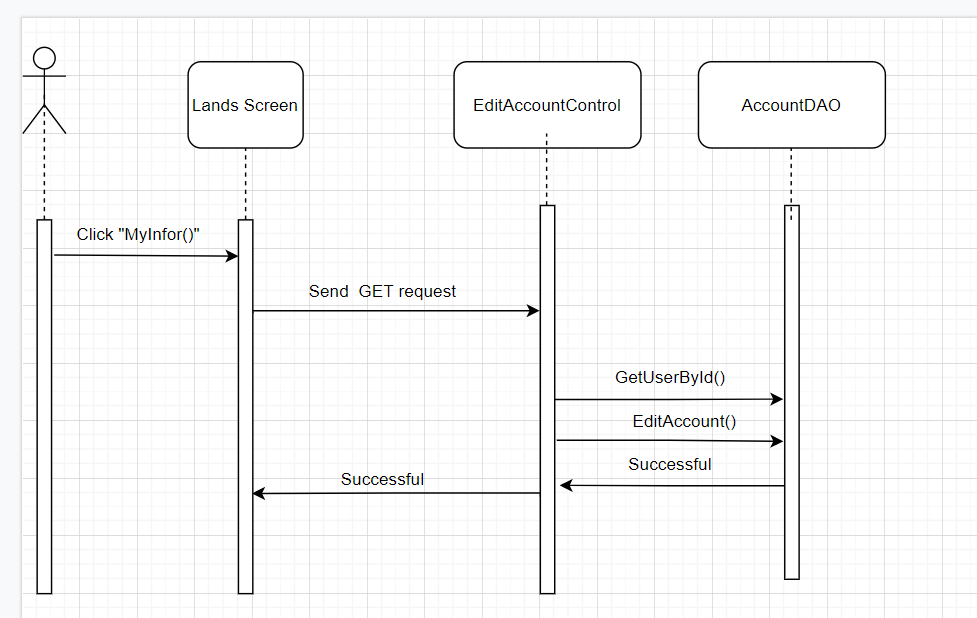
BlogDAO

[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

## 15.Set Role User

### a. Class Diagram

### b. Class Specifications

UserController

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

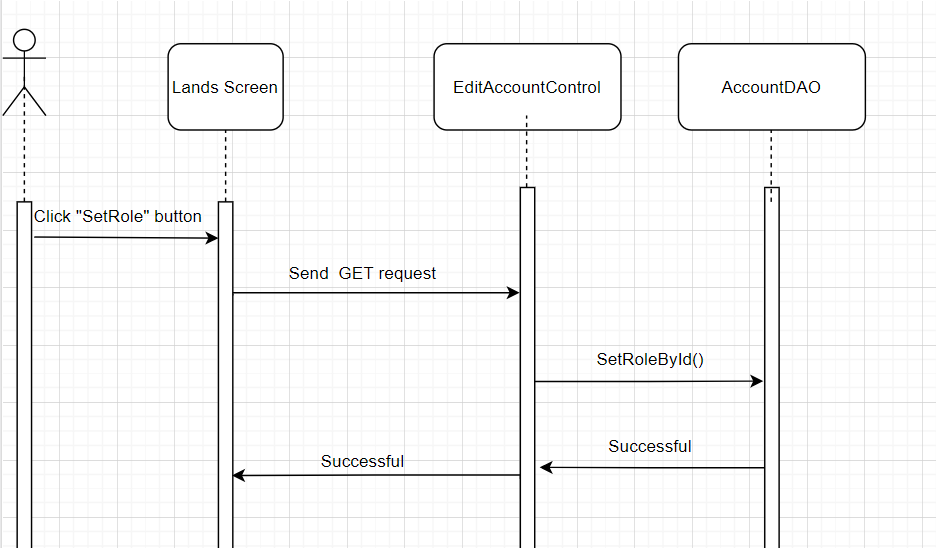
BlogDAO

[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

UserController

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

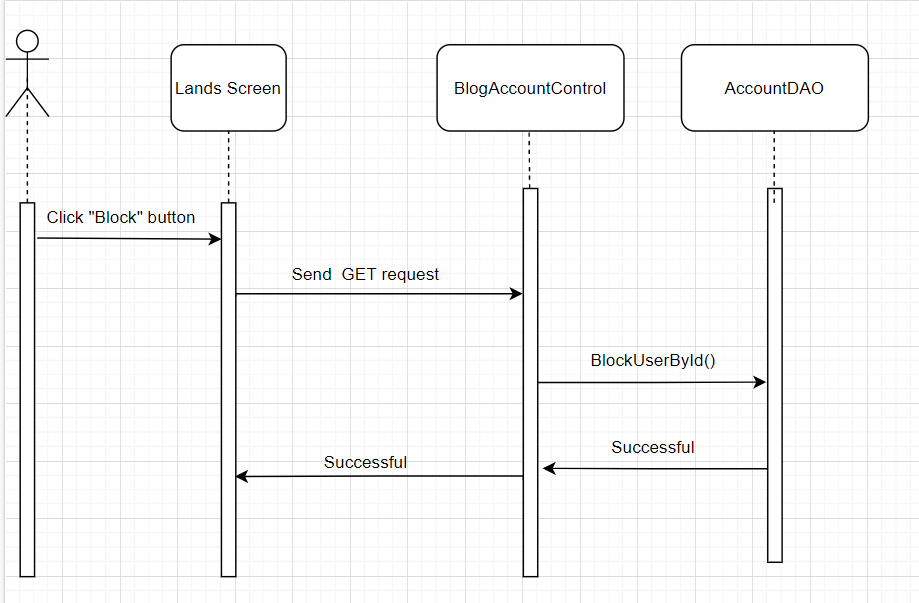
BlogDAO

[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

## 17. Reset Password

### Class Diagram

### Class Specifications

### 

UserController

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

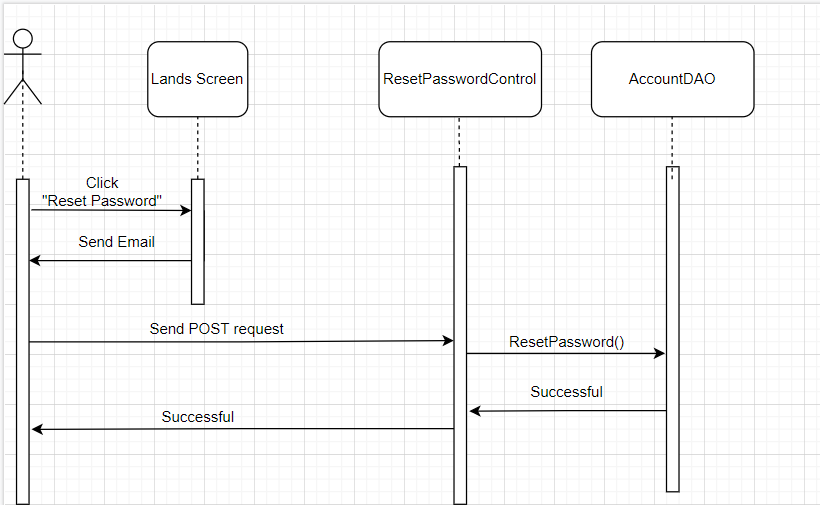
BlogDAO

[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

UserController

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

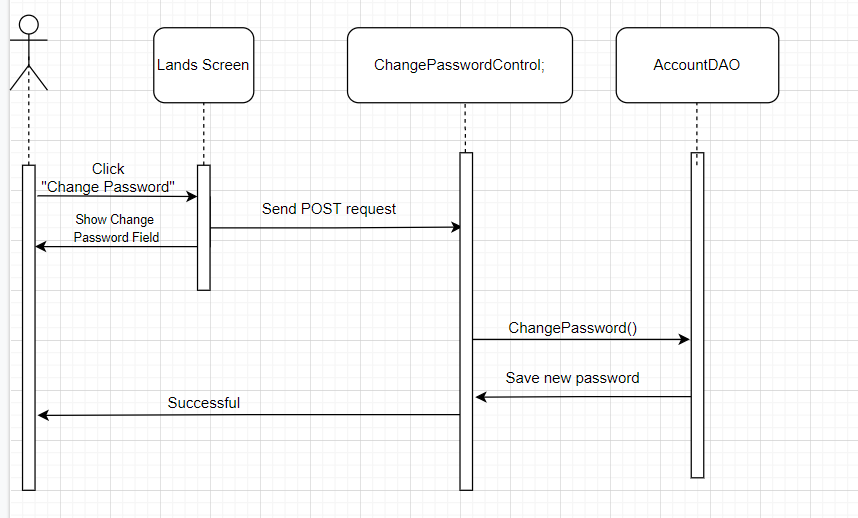
BlogDAO

[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

## 19. User profile

### a. Class Diagram

### b. Class Specifications

UserController

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

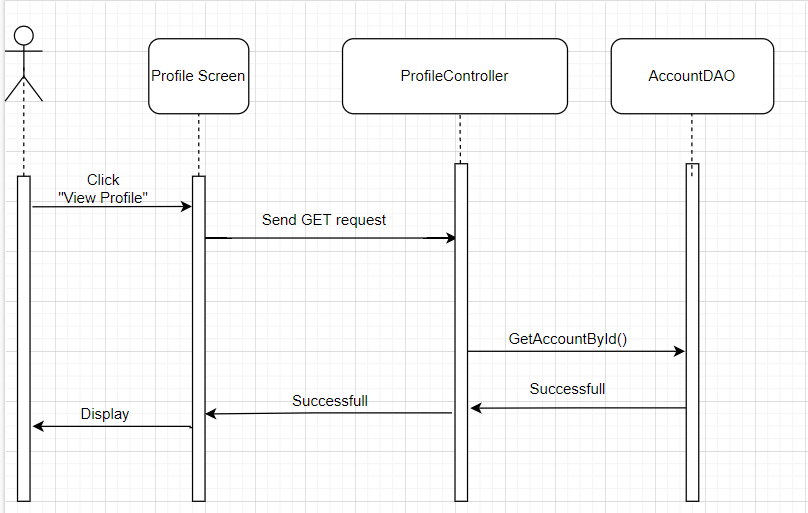
BlogDAO

[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

## 20. Order History

### a. Class Diagram

### b. Class Specifications

UserController

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

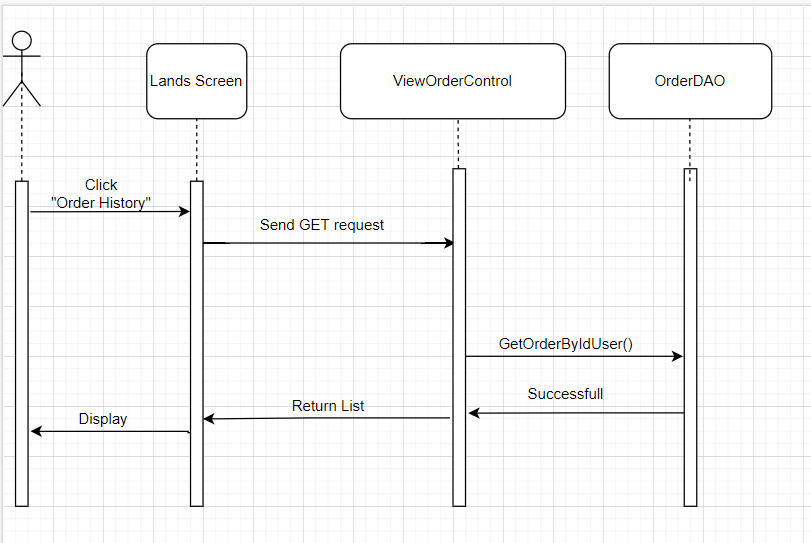
BlogDAO

[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)



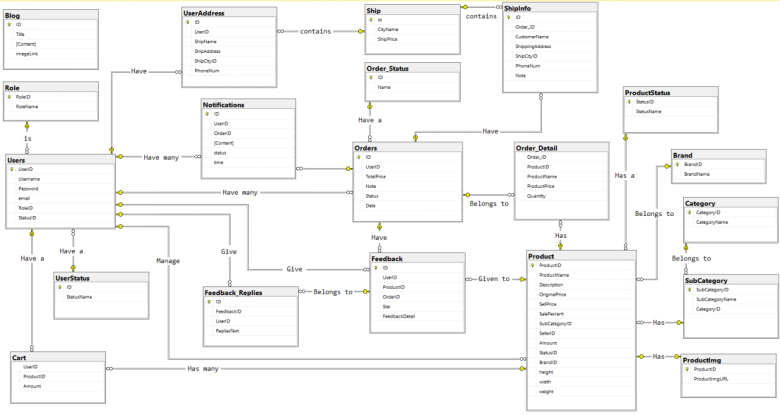
### 

### d. Database queries

Select \* from Users

# 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* |