

**BLAZO**

**Software Design Specification**

**Version: 1.2**

Prepare by:

Le Thai An (HE181137)

Tran Minh Tien(HE182319)

**– Hanoi, July 2024 –**

**Record of changeS**

| **Date** | **A\* M, D** | **In charge** | **Change Description** |
| --- | --- | --- | --- |
| 19/05 | A | TienTM,  AnLT | Add overview and code design |
| 12/07 | M | TienTM,  AnLT | Edit code design for some feature |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

\*A - Added M - Modified D - Deleted

**Table of Contents**

[**I. Overview** 4](#_gjdgxs)

[1. Code Packages](#_30j0zll) 4

[2. Database Design](#_1fob9te) 5

[a. Database Schema](#_3znysh7) 5

[b. Table Description](#_2et92p0) 6

[**II. Code Designs**](#_tyjcwt) 10

[1. <Visit Store/ Add product to cart>](#_udbs2czglga) 10

[2. <Checkout/Manage Order>](#_kheqb8ma4sg3) 12

[3. <Add product/ Delete product/ Edit product>](#_n1orb1bgaz0) 14

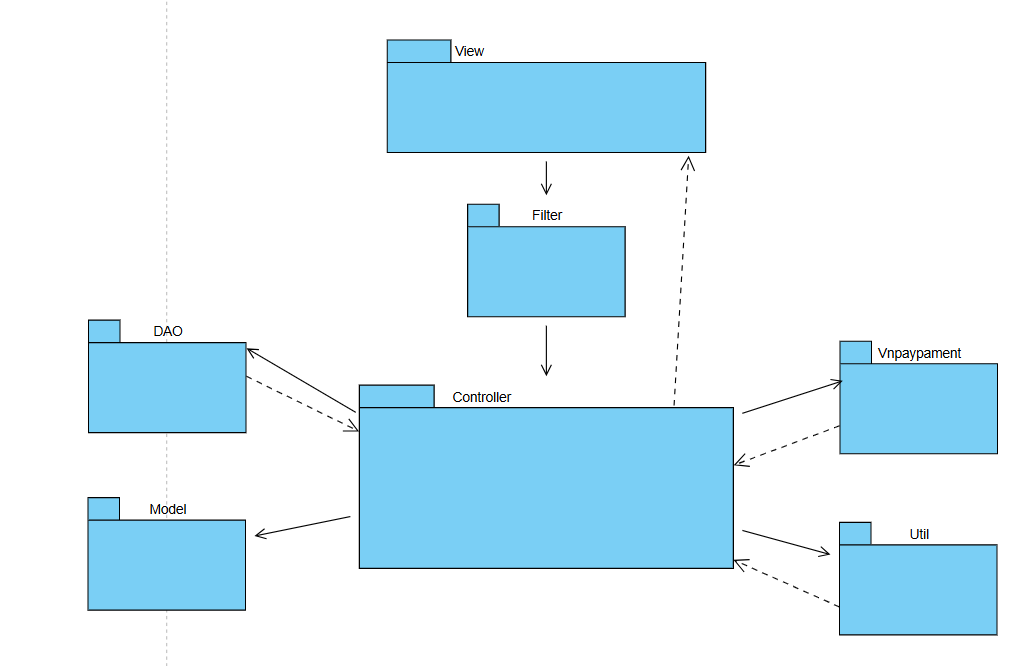
[4. <Authentication/Login/Forgot password/Log out>](#_89pdtat6zhio) 16

[5. <Register>](#_yzirbrltgqqu) 19

[6. <Web Blog>](#_dslr81abf65d) 21

# **I. Overview**

## **1. Code Packages**

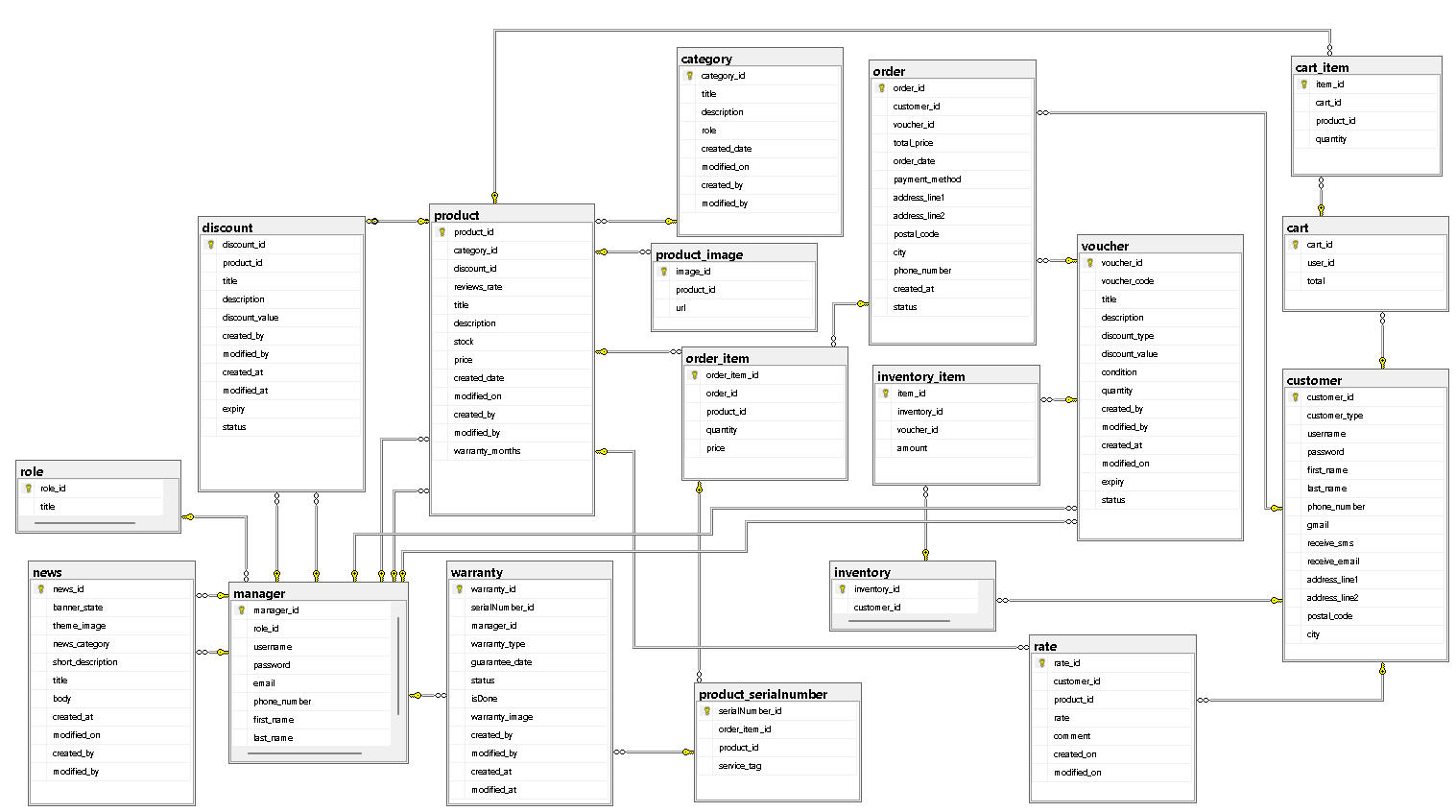


***Package descriptions***

| **No** | **Package** | **Description** |
| --- | --- | --- |
| 1 | Util | The "Util" package contains utility classes and helper methods used to perform common tasks and facilitate reusability within the application. Classes in this package typically include functions such as string manipulation, date formatting, encoding and decoding, and other utility functions. |
| 2 | DAO | The "DAO" package contains classes used to access and interact with the database. Classes in this package typically include methods to perform operations such as adding, updating, deleting, and querying data from the database. The "dal" package often serves as an intermediary layer between the application and the database, helping to separate application logic from data access logic. |
| 3 | Model | The "Model" package contains classes representing data entities within the application. Classes in this package are designed to represent objects in the application but do not contain any business logic. Instead, they simply model the data and are often used to transfer data between different layers and components of the application. |
| 4 | Vnpaypayment | The "Vnpaypaymentl" package contains vnpay payment api |
| 5 | Controller | The "Controller" package contains classes and method to control, process data and connect each other package |
| 6 | Filter | The "Filter" package contains classes to verifies the identity of users and determines whether they have permission to access certain resources or perform specific actions within the application |
| 7 | View | The "Model" package contains jsp file to display user interface |

## **2. Database Design**

### ***a. Database Schema***



### ***b. Table Description***

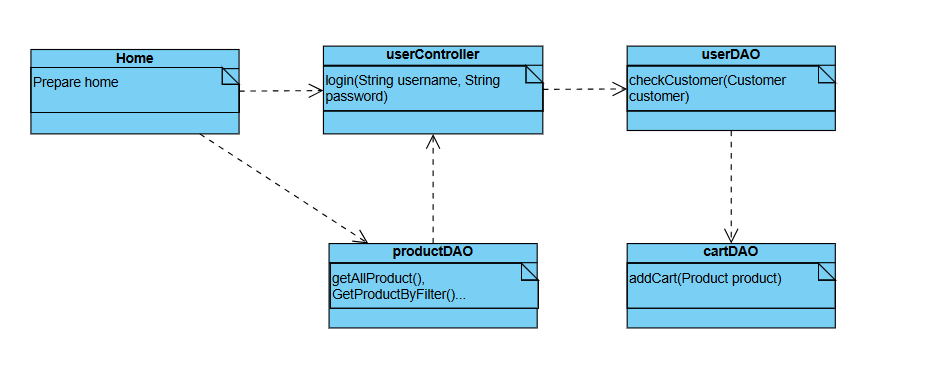
#### 

| **No** | **Table** | **Description** |
| --- | --- | --- |
| **01** | customer | customer\_id INT PRIMARY KEY IDENTITY,  customer\_type INT,  customer\_img NVARCHAR(max),  username NVARCHAR(255),  [password] NVARCHAR(255),  first\_name NVARCHAR(255),  last\_name NVARCHAR(255),  phone\_number NVARCHAR(255),  gmail NVARCHAR(255),  receive\_sms BIT,  receive\_email BIT,  address\_line1 NVARCHAR(255),  address\_line2 NVARCHAR(255),  postal\_code NVARCHAR(255),  city NVARCHAR(255) |
| **02** | inventory | inventory\_id INT PRIMARY KEY IDENTITY,  customer\_id INT |
| **03** | inventory\_item | item\_id INT PRIMARY KEY IDENTITY,  inventory\_id INT,  voucher\_id INT,  amount INT |
| **04** | manager | manager\_id INT PRIMARY KEY IDENTITY,  role\_id INT,  username NVARCHAR(255),  password NVARCHAR(255),  email NVARCHAR(255),  phone\_number NVARCHAR(255),  first\_name NVARCHAR(255),  last\_name NVARCHAR(255) |
| **05** | role | role\_id INT PRIMARY KEY IDENTITY,  title NVARCHAR(255) |
| **06** | product | product\_id INT PRIMARY KEY IDENTITY,  category\_id INT,  discount\_id INT,  reviews\_rate INT,  title NVARCHAR(255),  description NVARCHAR(MAX),  stock INT,  price FLOAT,  created\_date DATE,  modified\_on DATE,  created\_by INT,  modified\_by INT,  warranty\_months INT |
| **07** | rate | rate\_id INT PRIMARY KEY IDENTITY,  customer\_id INT,  product\_id INT,  rate INT,  comment NVARCHAR(MAX),  created\_on DATETIME,  modified\_on DATETIME |
| **08** | category | category\_id INT PRIMARY KEY IDENTITY,  title NVARCHAR(255),  description NVARCHAR(max),  role INT,  created\_date DATE,  modified\_on DATE,  created\_by INT,  modified\_by INT |
| **09** | order | order\_id INT PRIMARY KEY IDENTITY,  customer\_id INT,  voucher\_id INT,  total\_price FLOAT,  order\_date DATE,  payment\_method INT,  address\_line1 NVARCHAR(255),  address\_line2 NVARCHAR(255),  postal\_code NVARCHAR(255),  city NVARCHAR(255),  phone\_number NVARCHAR(255),  created\_at DATE,  status tinyint |
| **10** | order\_item | order\_item\_id INT PRIMARY KEY IDENTITY,  order\_id INT,  product\_id INT,  quantity INT,  price FLOAT |
| **11** | discount | discount\_id INT PRIMARY KEY IDENTITY,  product\_id INT,  title NVARCHAR(255),  description NVARCHAR(MAX),  discount\_value FLOAT,  created\_by INT,  modified\_by INT,  created\_at DATE,  modified\_at DATE,  expiry datetime,  status bit |
| **12** | voucher | voucher\_id INT PRIMARY KEY IDENTITY,  voucher\_code NVARCHAR(MAX),  title NVARCHAR(MAX),  description NVARCHAR(MAX),  discount\_type INT,  discount\_value FLOAT,  condition float,  quantity INT,  created\_by INT,  modified\_by INT,  created\_at DATE,  modified\_on DATE,  expiry DATE,  statusBIT |
| **13** | product\_image | image\_id INT PRIMARY KEY IDENTITY,  product\_id INT,  url NVARCHAR(255) |
| **14** | news | news\_id INT PRIMARY KEY IDENTITY,  banner\_state INT,  theme\_image NVARCHAR(MAX),  news\_category NVARCHAR(255),  short\_description NVARCHAR(MAX),  title NVARCHAR(255),  body NVARCHAR(MAX),  created\_at DATE,  modified\_on DATE,  created\_by INT,  modified\_by INT |
| **15** | cart | cart\_id INT PRIMARY KEY IDENTITY,  user\_id INT,  total FLOAT |
| **16** | cart\_item | item\_id INT PRIMARY KEY IDENTITY,  cart\_id INT,  product\_id INT,  quantity INT |
| **17** | product\_serialnumber | serialNumber\_id INT PRIMARY KEY IDENTITY,  order\_item\_id INT,  product\_id int,  service\_tag NVARCHAR(MAX), |
| **18** | warranty | warranty\_id INT PRIMARY KEY IDENTITY,  serialNumber\_id int,  manager\_id int,  warranty\_type int,  guarantee\_date DATE,  status NVARCHAR(255),  isDone BIT,  warranty\_image NVARCHAR(MAX),  created\_by INT,  modified\_by INT,  created\_at DATE,  modified\_at DATE, |

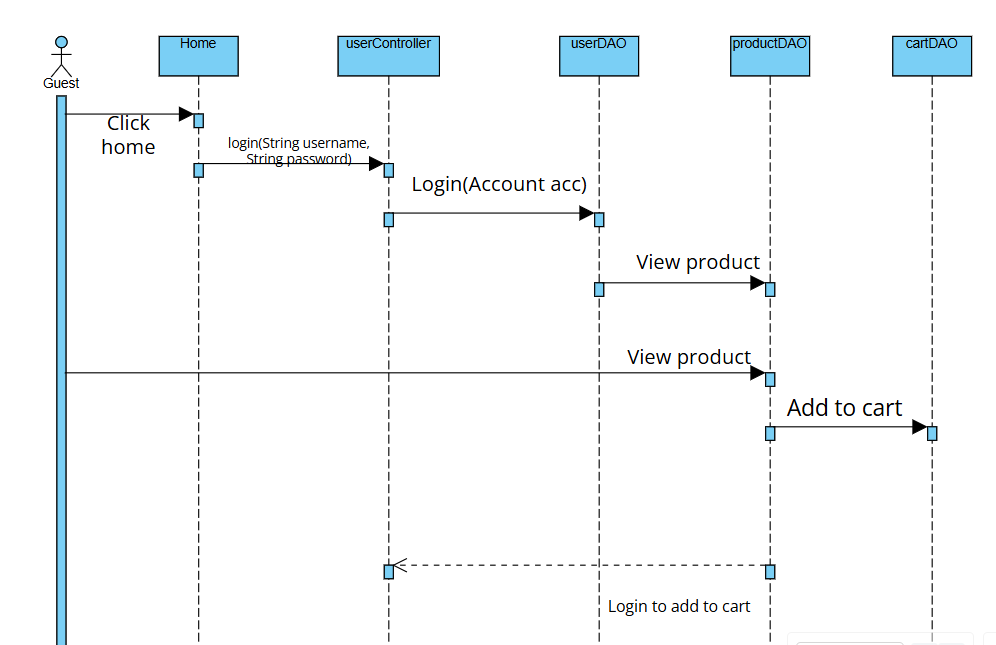
# **II. Code Designs**

## **1. <Visit Store/ Add product to cart>**

### **a. Class Diagram**



### **b. Sequence Diagram**



### **c. Class Specifications**

***Guest***

| **No** | **Method** | **Description** |
| --- | --- | --- |
| *01* | *get()* | *Returns the current value of the property* |
| *02* | *set()* | Set new value of the property |

***Product***

| **No** | **Method** | **Description** |
| --- | --- | --- |
| *01* | *get()* | *Returns the current value of the property* |
| *02* | *set()* | Set new value of the property |

***Cart***

| **No** | **Method** | **Description** |
| --- | --- | --- |
| *01* | *get()* | *Returns the current value of the property* |
| *02* | *set()* | Set new value of the property |

***UserDAO***

| **No** | **Method** | **Description** |
| --- | --- | --- |
| *01* | *checkSignIn*() | Parameter: User object. This function check username and password is correct or not. |
| *02* | *checkUsernameEmailIsDuplicated()* | Parameter: User object. This function check if username and email is correct to reset password. |

***UserControll***

| **No** | **Method** | **Description** |
| --- | --- | --- |
| *01* | signIn() | *Parameter: HttpServletRequest, HttpServletResponse. This funtion help user sign in to the system.* |
| *02* | logOut() | *Parameter: HttpServletRequest, HttpServletResponse. This function in order to log out the system.* |

### **d. Database queries**

Select \* from product

Select \* from cart where user\_id=?

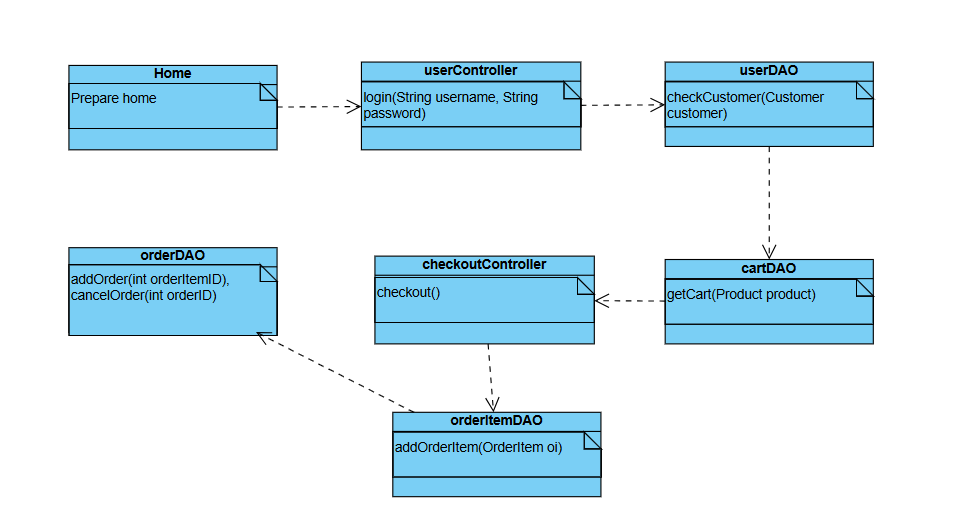
Insert into cart(productID,quantity) values (?,?) where user\_id=?

"SELECT \* FROM users WHERE username = ? AND email = ?" **resetPassword**

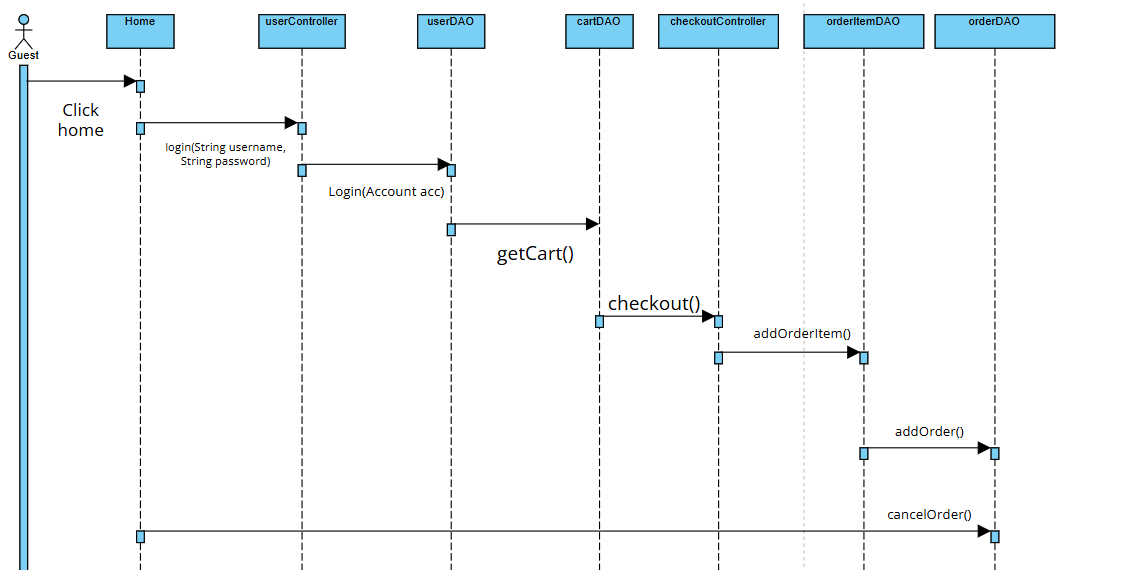
"UPDATE users SET users.password = ? WHERE username = ?"

## **2. <Checkout/Manage Order>**

### **a. Class Diagram**



### **b. Sequence Diagram**



### **c. Class Specifications**

***Guest***

| **No** | **Method** | **Description** |
| --- | --- | --- |
| *01* | *get()* | *Returns the current value of the property* |
| *02* | *set()* | Set new value of the property |

***Cart***

| **No** | **Method** | **Description** |
| --- | --- | --- |
| *01* | *get()* | *Returns the current value of the property* |
| *02* | *set()* | Set new value of the property |

***OrderItem***

| **No** | **Method** | **Description** |
| --- | --- | --- |
| *01* | *get()* | *Returns the current value of the property* |
| *02* | *set()* | Set new value of the property |

***Order***

| **No** | **Method** | **Description** |
| --- | --- | --- |
| *01* | *get()* | *Returns the current value of the property* |
| *02* | *set()* | Set new value of the property |

***UserDAO***

| **No** | **Method** | **Description** |
| --- | --- | --- |
| *01* | *checkSignIn*() | Parameter: User object. This function check username and password is correct or not. |
| *02* | *checkUsernameEmailIsDuplicated()* | Parameter: User object. This function check if username and email is correct to reset password. |

***UserControll***

| **No** | **Method** | **Description** |
| --- | --- | --- |
| *01* | signIn() | *Parameter: HttpServletRequest, HttpServletResponse. This funtion help user sign in to the system.* |
| *02* | logOut() | *Parameter: HttpServletRequest, HttpServletResponse. This function in order to log out the system.* |

### 

### **d. Database queries**

Select \* from orderItem where order\_id=?

Select \* from order where id=?

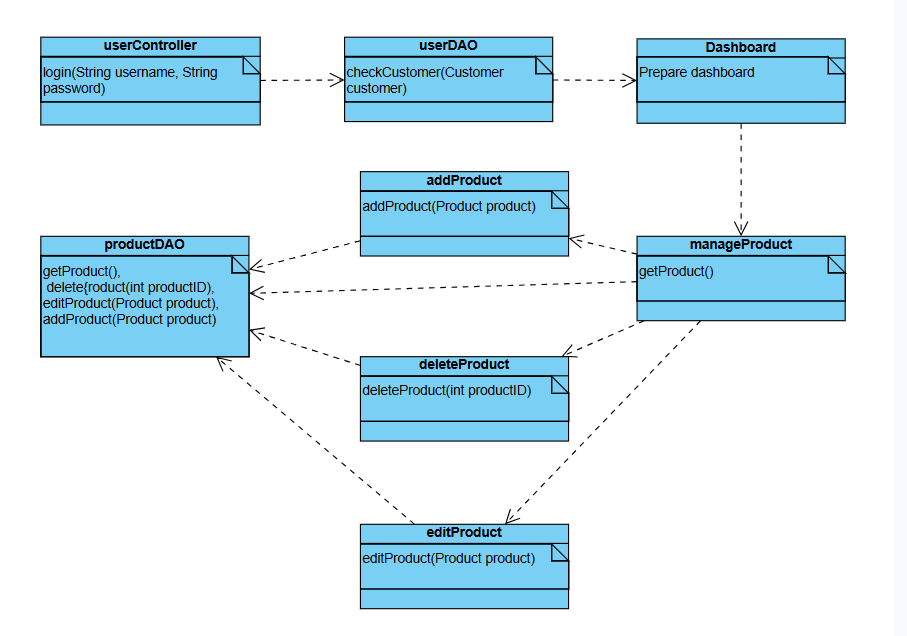
Delete from order where id=?

"SELECT \* FROM users WHERE username = ? AND email = ?" **resetPassword**

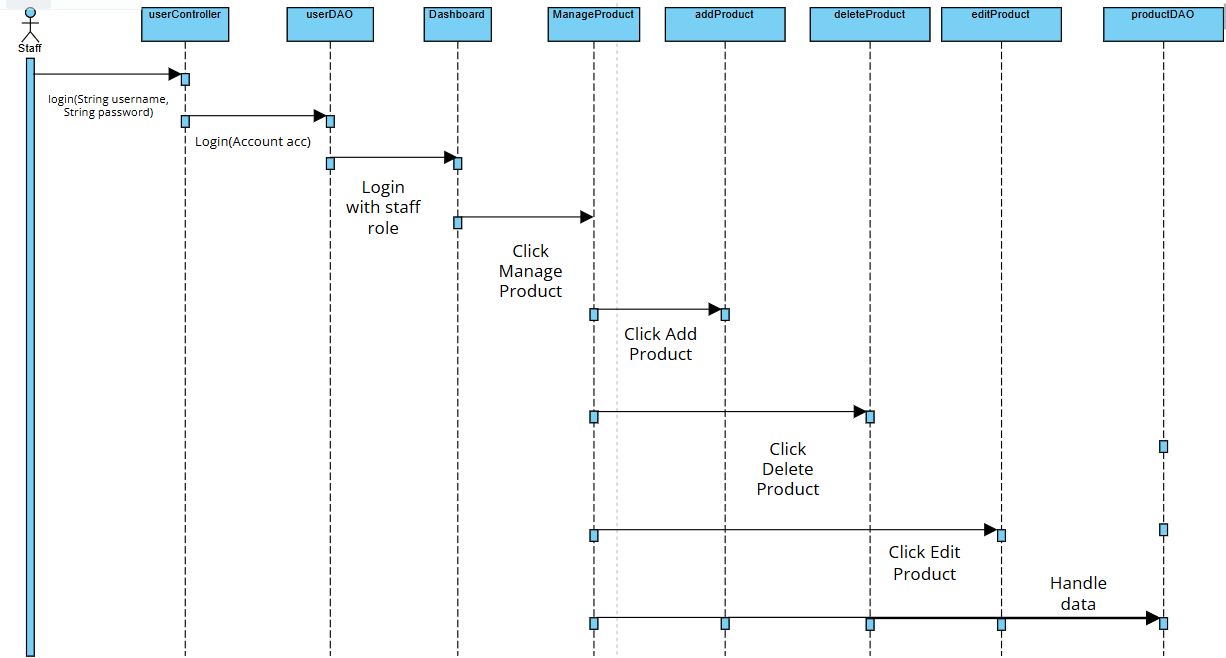
"UPDATE users SET users.password = ? WHERE username = ?"

## **3. <Add product/ Delete product/ Edit product>**

### **a. Class Diagram**



### **b. Sequence Diagram**



### **c. Class Specifications**

***Guest***

| **No** | **Method** | **Description** |
| --- | --- | --- |
| *01* | *get()* | *Returns the current value of the property* |
| *02* | *set()* | Set new value of the property |

***Product***

| **No** | **Method** | **Description** |
| --- | --- | --- |
| *01* | *get()* | *Returns the current value of the property* |
| *02* | *set()* | Set new value of the property |
| *03* | *addProduct(Product p)* | Add new Product |
| *04* | *deleteProduct(int id)* | Delete Product |
| *05* | *editProduct(Product p)* | Update Product |

***UserDAO***

| **No** | **Method** | **Description** |
| --- | --- | --- |
| *01* | *checkSignIn*() | Parameter: User object. This function check username and password is correct or not. |
| *02* | *checkUsernameEmailIsDuplicated()* | Parameter: User object. This function check if username and email is correct to reset password. |

***UserControll***

| **No** | **Method** | **Description** |
| --- | --- | --- |
| *01* | signIn() | *Parameter: HttpServletRequest, HttpServletResponse. This funtion help user sign in to the system.* |
| *02* | logOut() | *Parameter: HttpServletRequest, HttpServletResponse. This function in order to log out the system.* |

### 

### **d. Database queries**

Select \* from product where user\_id=?

Select \* from product where id=?

Delete from product where id=?

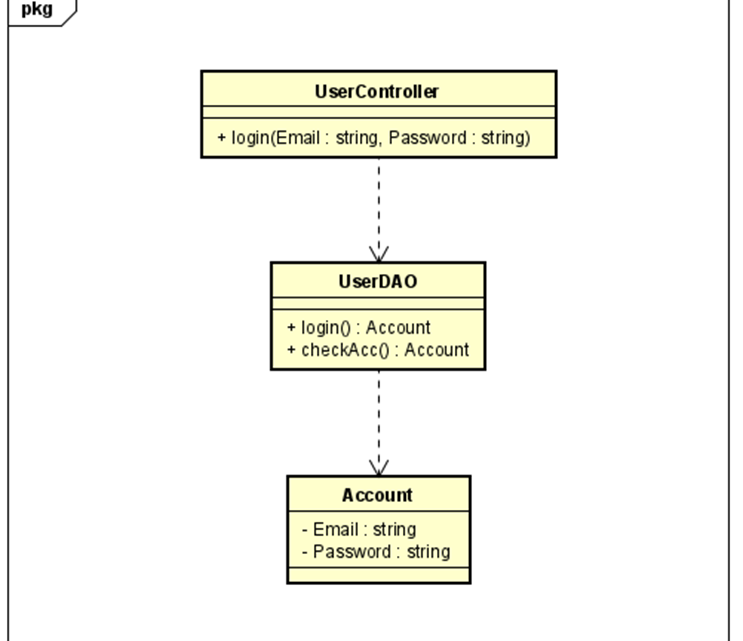
"SELECT \* FROM users WHERE username = ? AND email = ?" **resetPassword**

"UPDATE users SET users.password = ? WHERE username = ?"

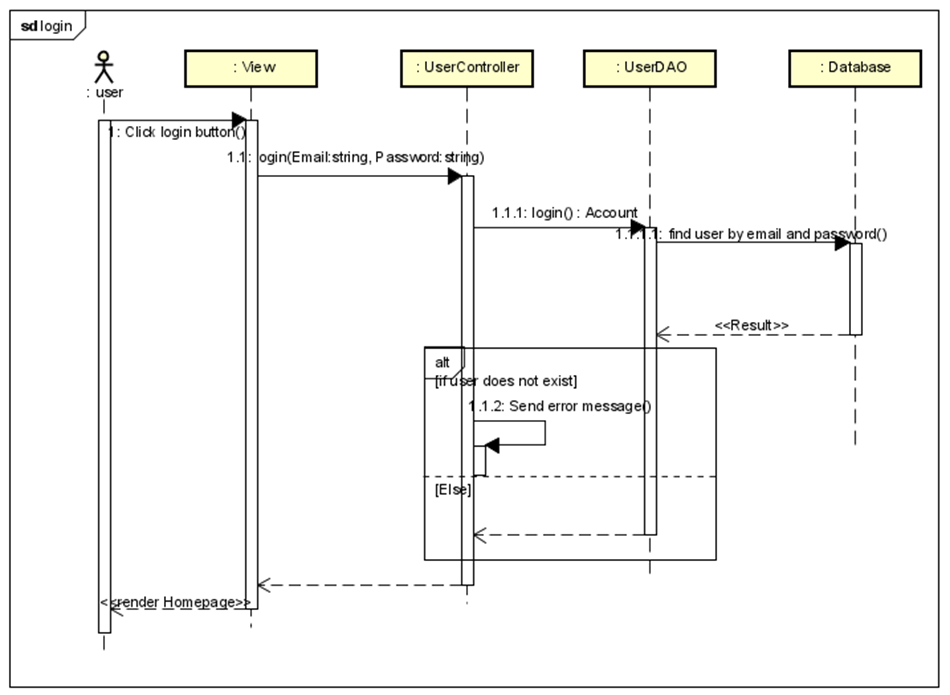
## 

## **4. <Authentication/Login/Forgot password/Log out>**

### **a. Class Diagram**



### **b. Sequence Diagram**



### **c. Class Specifications**

***User***

| **No** | **Method** | **Description** |
| --- | --- | --- |
| *01* | *get()* | *Returns the current value of the property* |
| *02* | *set()* | Set new value of the property |

***Employee***

| **No** | **Method** | **Description** |
| --- | --- | --- |
| *01* | *get()* | *Returns the current value of the property* |
| *02* | *set()* | Set new value of the property |

***UserDAO***

| **No** | **Method** | **Description** |
| --- | --- | --- |
| *01* | *checkSignIn*() | Parameter: User object. This function check username and password is correct or not. |
| *02* | *resetPassword()* | Parameter: User object. This function will reset new password |
| *03* | *checkUsernameEmailIsDuplicated()* | Parameter: User object. This function check if username and email is correct to reset password. |

***UserControll***

| **No** | **Method** | **Description** |
| --- | --- | --- |
| *01* | signIn() | *Parameter: HttpServletRequest, HttpServletResponse. This funtion help user sign in to the system.* |
| *02* | logOut() | *Parameter: HttpServletRequest, HttpServletResponse. This function in order to log out the system.* |
| *03* | resetPassword(  ) | *Parameter: HttpServletRequest, HttpServletResponse. This function to reset password if user forgot password.* |
| *04* | validateEmail() | *Parameter: HttpServletRequest, HttpServletResponse. This function check check if email is correct or not* |

### **d. Database queries**

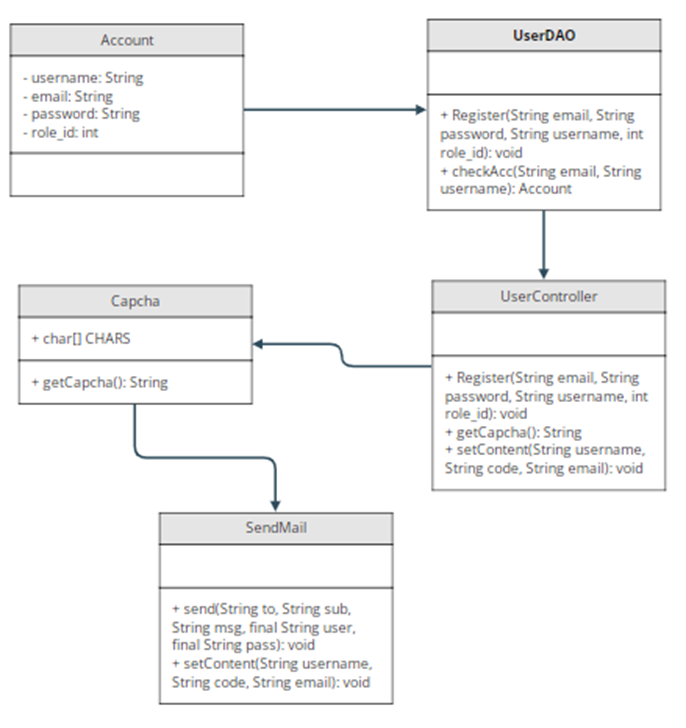
checkUsernameEmailIsDuplicated

"SELECT \* FROM users WHERE username = ? AND email = ?" **resetPassword**

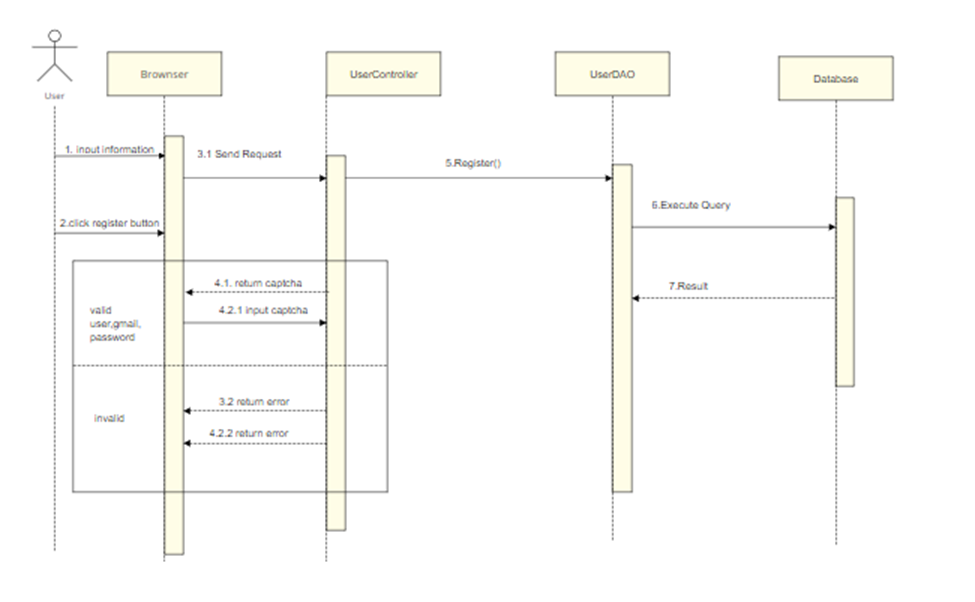
"UPDATE users SET users.password = ? WHERE username = ?"

## **5. <Register>**

### **a. Class Diagram**



### **b. Sequence Diagram**



### **c. Class Specifications**

| No | Method | Description |
| --- | --- | --- |
| 1 | Register | use to create new user |
| 2 | checkAcc | use to check username or email exist |
| 3 | getCapcha | get a random string |
| 4 | send | send mail to user |
| 5 | setContent | set email and username that user use to register to send |

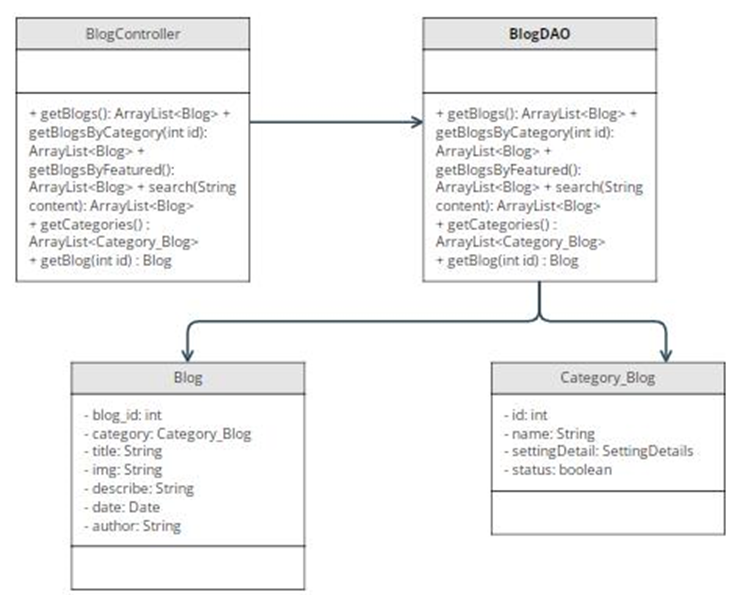
### **d. Database queries**

SELECT \* FROM users WHERE email=? OR username=?

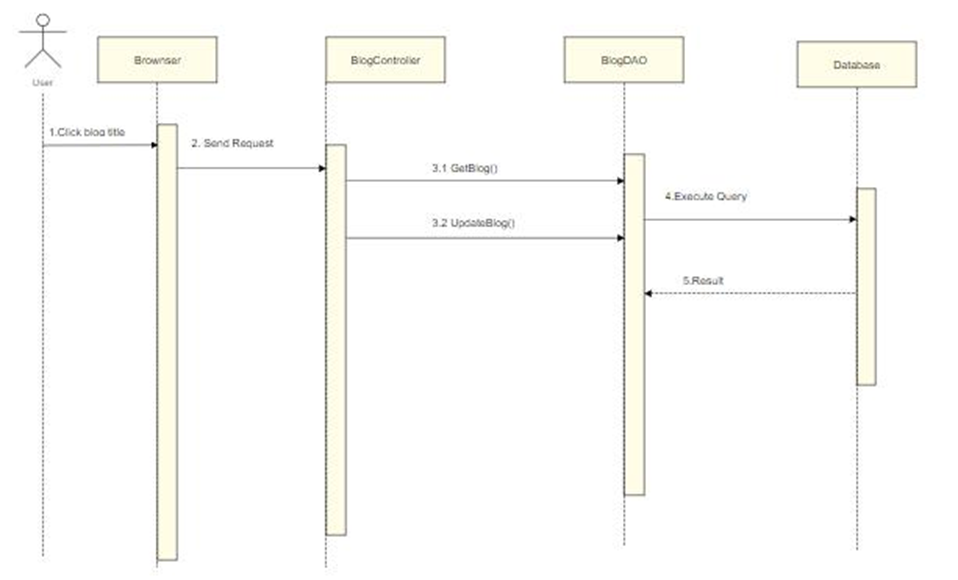
INSERT INTO `doctris\_system`.`users` (`username`, `password`, `email`, `role\_id`) VALUES (?, ?, ?, ?)

## **6. <Web Blog>**

### **a. Class Diagram**



### **b. Sequence Diagram**



### **c. Class Specifications**

| No | Method | Description |
| --- | --- | --- |
| 1 | getBlogs | get all blog in database order by date |
| 2 | getCategories | get all categories in database |
| 3 | getBlog | get blog by blog id |
| 4 | getBlogBy Featured | get blog which featured |
| 5 | search | get blog that having or contain keyword in its title |
| 6 | getBlogByCategory | get blog in the same category |

### **d. Database queries**

- SELECT news.\*, category\_news.name FROM blog

inner join category\_news

on blog.category\_id = category\_news.id

order by date desc

* SELECT \* FROM category\_news;
* SELECT \* FROM news WHERE title LIKE ? ORDER BY date desc