

**FASHION SHOP ONLINE**

**Software Design Document**

– Hanoi, May 2021 –

**Table of Contents**

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

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

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

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

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

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

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

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

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

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

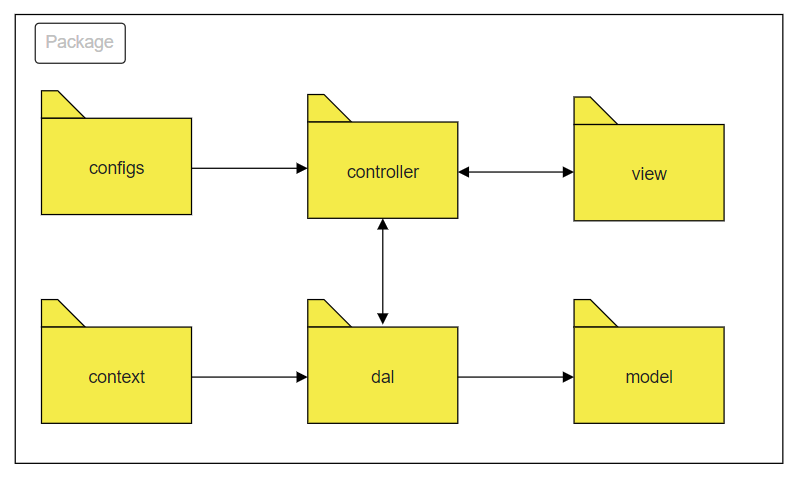
[III. Database Tables 5](#_heading=h.17dp8vu)

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

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

# I. Overview

## 1. Code Packages/Namespaces



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

| **No** | **Package** | **Description** |
| --- | --- | --- |
| *01* | *controller* | *Acts as a go-between for the service layer and the Front-end, passing on different input requests*  *<Class naming convention>* |
| *02* | *dal* | *This component is used to get data directly FROM the database*  *<Class naming convention>* |
| *03* | *configs* | *This component helps define configuration*  *<Class naming convention>* |
| *04* | *context* | *This component helps connect database*  *<Class naming convention>* |
| *05* | *model* | *This component includes interfaces used to receive data retrieved FROM the database and classes used to receive data*  *<Class naming convention>* |
| *05* | *view* | *This component display front-end*  *<Class naming convention>* |

## 2. Coding Conventions

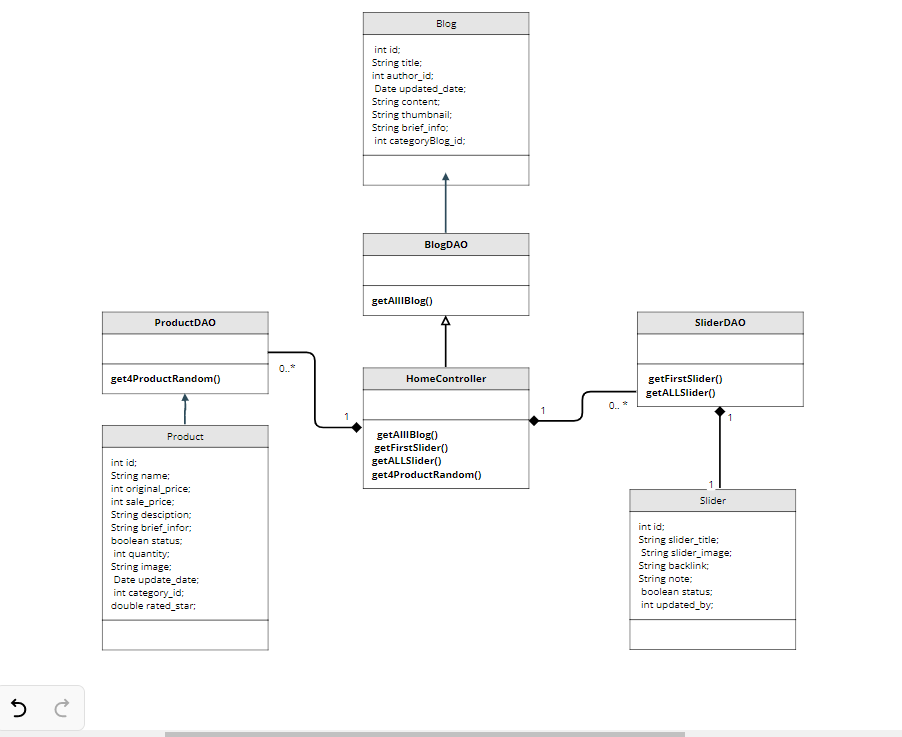
# II. Code Designs

## 1. Public/Home Page

*[Provide the detailed design for the function <Feature/Function Name1>. It include Class Diagram, Class Specifications, and Sequence Diagram(s)]*

### a. Class Diagram

*[This part presents the class diagram for the relevant feature]*



### b. Class Specifications

*[Provide the description for each class and the methods in each class, following the table format as below]*

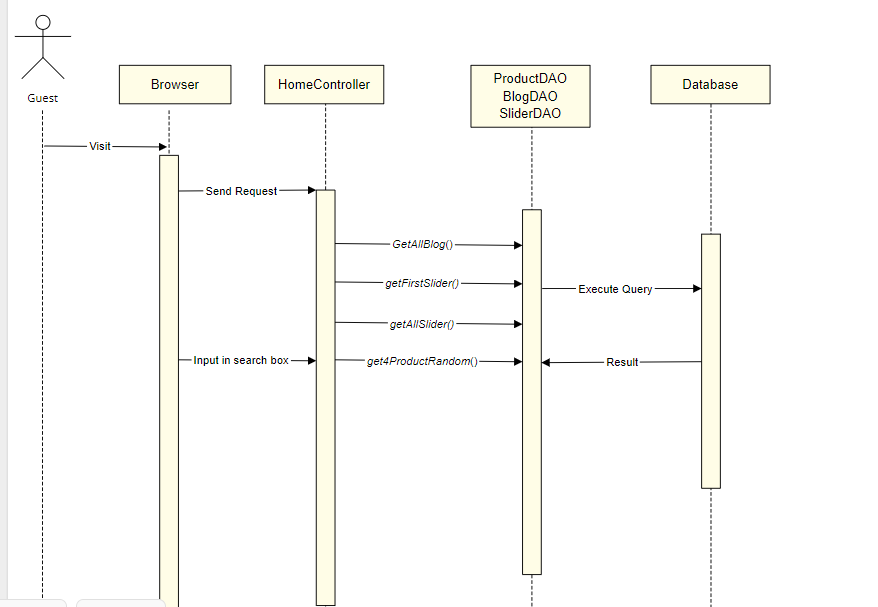
#### XYZ Class

*[Provide the detailed description for the class methods]*

| **No** | **Method** | **Description** |
| --- | --- | --- |
| *01* | *getAllIBlog()* | *get all property in Blog* |
| *02* | *getFirstSlider()* | *get all property in slider first* |
| *03* | *getALLSlider()* | *get all property in slider* |
| *04* | *get4ProductRandom()* | *get 4 product random from products* |
|  |  |  |

### c. Sequence Diagram(s)

*[Provide the sequence diagram(s) for the feature, see the sample below]*



### d. Database queries

*[Provide the detailed SQL (select, insert, update...) which are used in implementing the function/screen]*

SELECT top 1 \* FROM [dbo].[Slider]

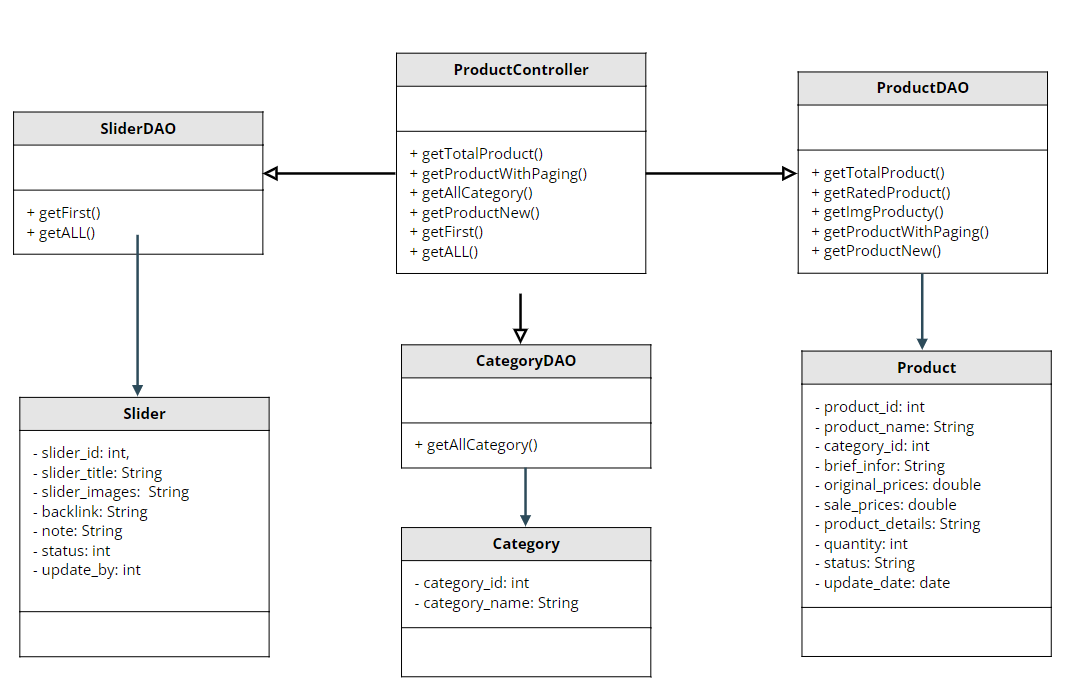
SELECT \* FROM [dbo].[Slider]

Select \* from Blog

Select top 4 \* from Product ORDER BY NEWID()

## 2. Public/Products List

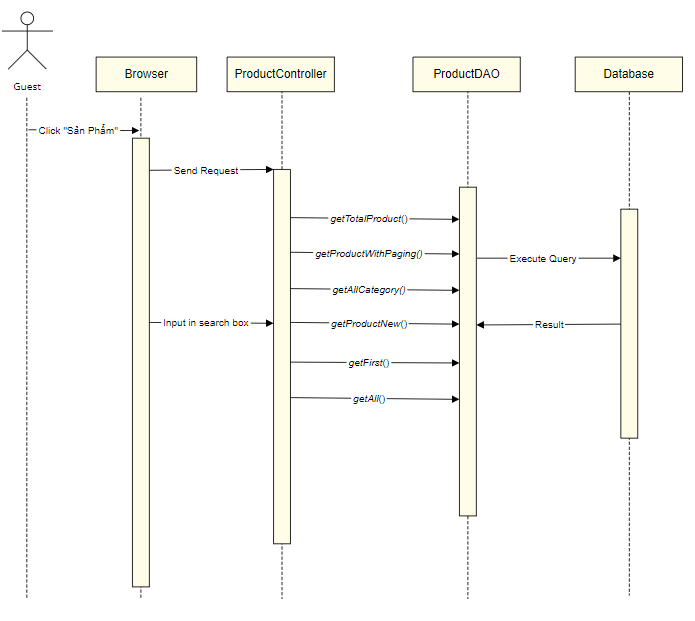
### a. Class Diagram



### b. Class Specifications

| **No** | **Method** | **Description** |
| --- | --- | --- |
| *01* | *getTotalProduct()* | *Get the amount of all products* |
| *02* | *getProductWithPaging()* | *Get all product pagination in database* |
| *03* | *getAllCategory()* | *Get all category in database* |
| *04* | *getProductNew()* | *Get a product latest post* |
| *05* | *getFirst()* | *Get first slider in database* |
| *06* | *getAll()* | *Get all slider in database* |
| *07* | *getRatedProduct()* | *Get rated vote for the product* |
| *08* | *getImgProduct()* | *Get image for the product* |

### c. Sequence Diagram(s)



### d. Database queries

* Select count(product\_id) from Product where category\_id "+categoryId+" and product\_name like N'%"+searchKey+"%'
* Select \* from Product\n"

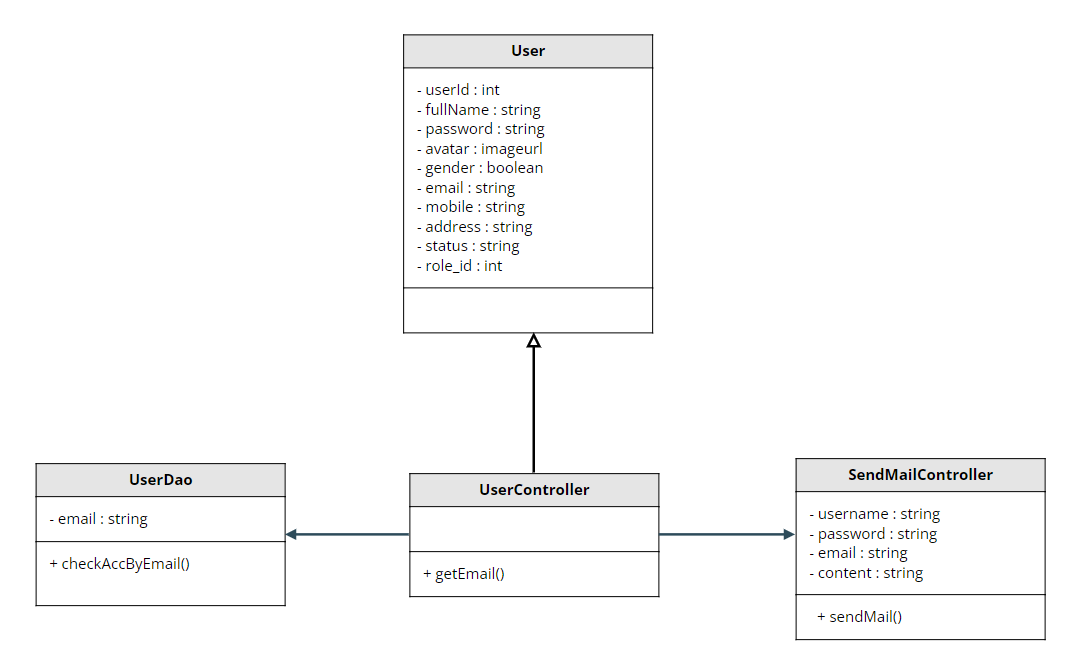
+ "where category\_id "+categoryId+" and product\_name like N'%"+searchKey+"%'\n"

+ " order by "+value+" "+type+" offset (?-1)\*? row fetch next ? row only

* Select \* from Category
* Select top 1 \* from Product order by update\_date desc
* SELECT top 1 \* FROM [dbo].[Slider]
* SELECT \* FROM [dbo].[Slider]

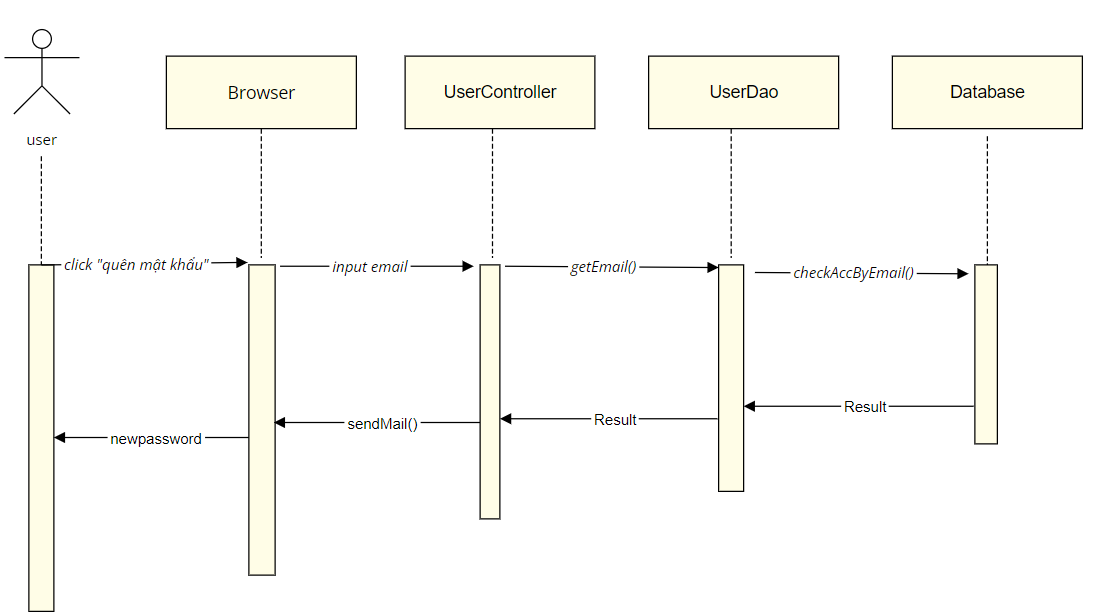
## 3. <Common/Reset Password>

### a. Class Diagram

**b. Class Specifications**

| **No** | **Method** | **Description** |
| --- | --- | --- |
| *01* | *getEmail()* | *Retrieve the User's Email entered* |
| *02* | *checkAccByEmail()* | *Check email existence* |
| *03* | *sendMail()* | *Send a link that leads the user to the reset password page* |

### c. Sequence Diagram(s)

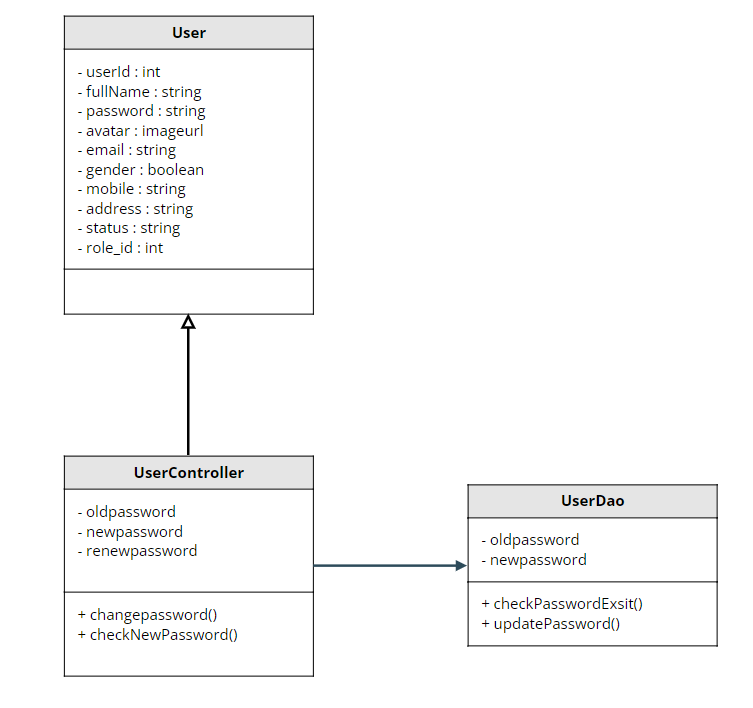


### d. Database queries

* select \* from user where email = ?
* update user set password = ? where userId = ?

## 4. <Common/Change Password>

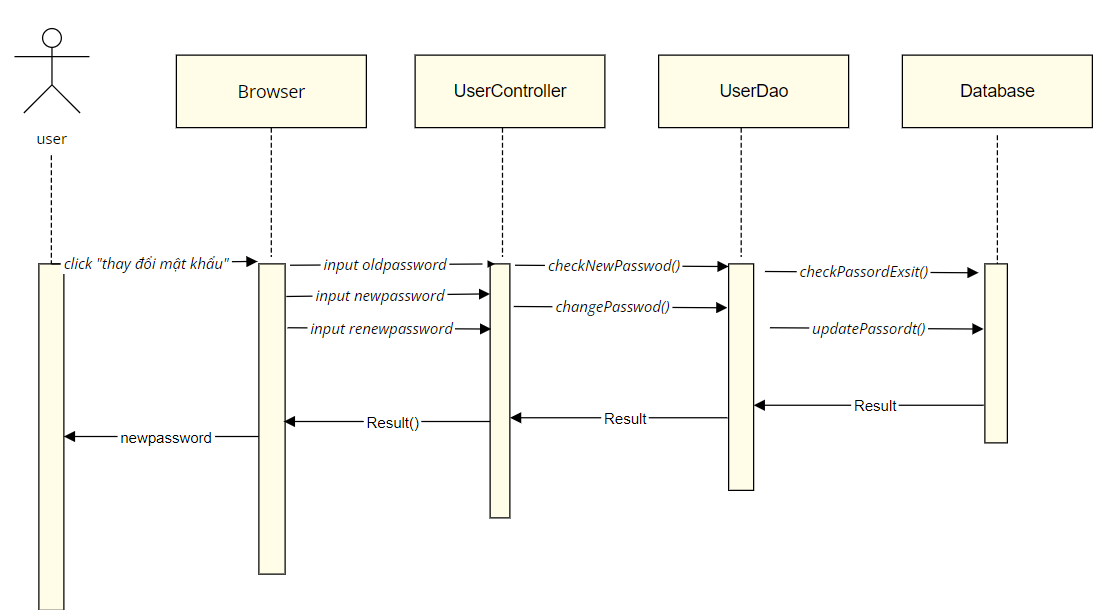
### a. Class Diagram



### b. Class Specifications

| **No** | **Method** | **Description** |
| --- | --- | --- |
| *01* | *changepassworld()* | *Changepassword* |
| *02* | *checkNewPassword()* | *Check the new password is the same(pass)* |
| *03* | *checkPasswordExsit()* | *Check the old password entered in the database* |
| *04* | *updatePassword()* | *Change the new password to the database* |

### c. Sequence Diagram(s)

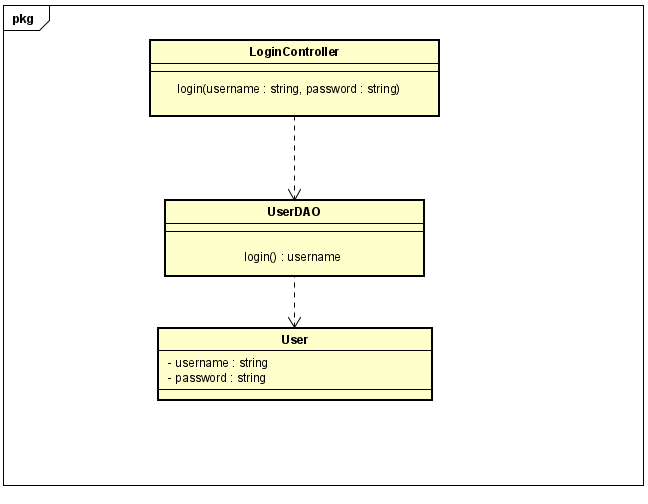


### d. Database queries

* select password = ? from user where useId = ?
* update user set password = ? where userId = ?

## 5. <Common/Login>

### a. Class Diagram

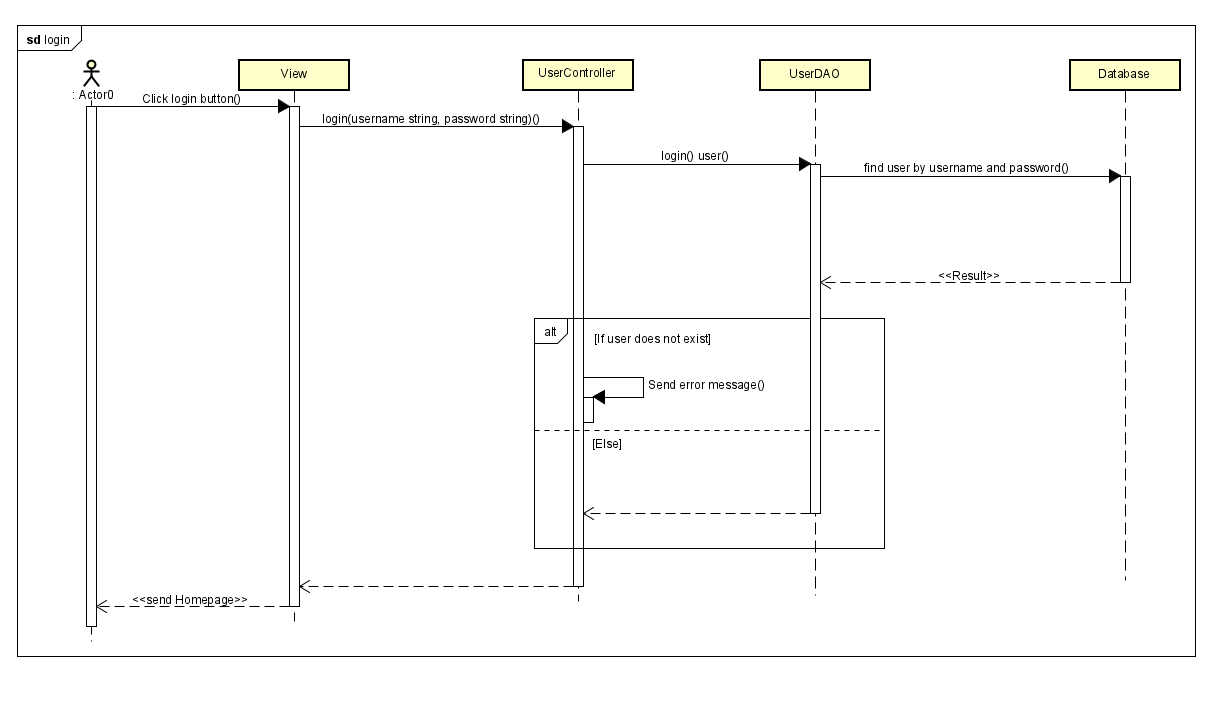


### b. Class Specifications

#### 

| **No** | **Method** | **Description** |
| --- | --- | --- |
| **01** | **login** | **Using to login into the system** |

### c. Sequence Diagram(s)

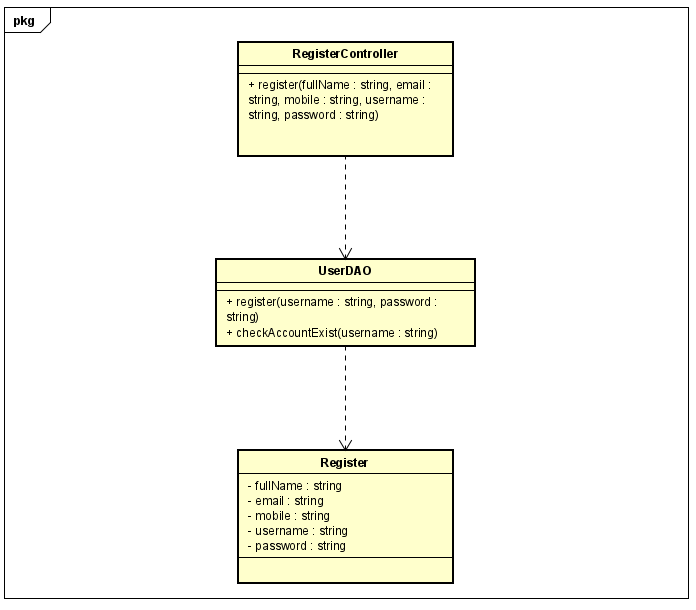


### d. Database queries

**String sql = "SELECT \* FROM user where user=? and password=?"**

## 6. <Common/Register>

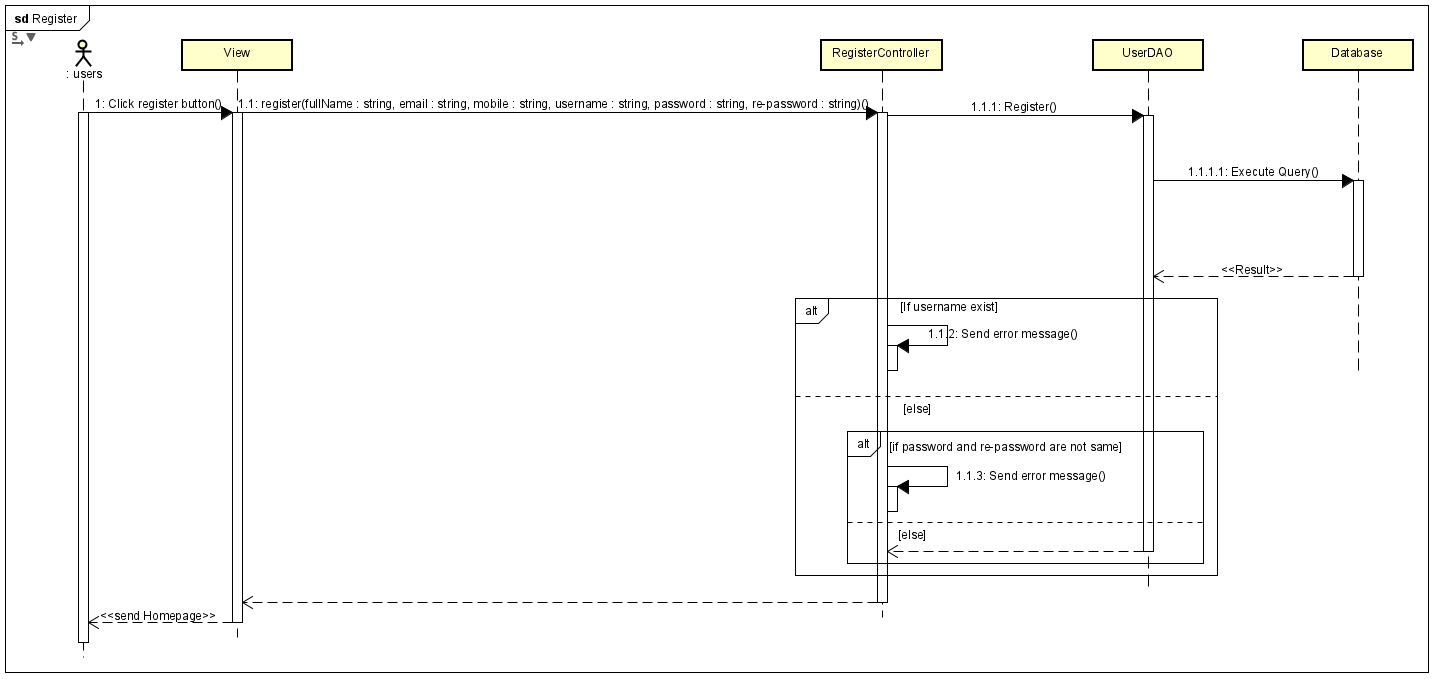
### a. Class Diagram



### b. Class Specifications

| **No** | **Method** | **Description** |
| --- | --- | --- |
| **1** | **Register** | **use to create new user** |
| **2** | **checkAccountExist** | **use to check username exist** |

### c. Sequence Diagram(s)

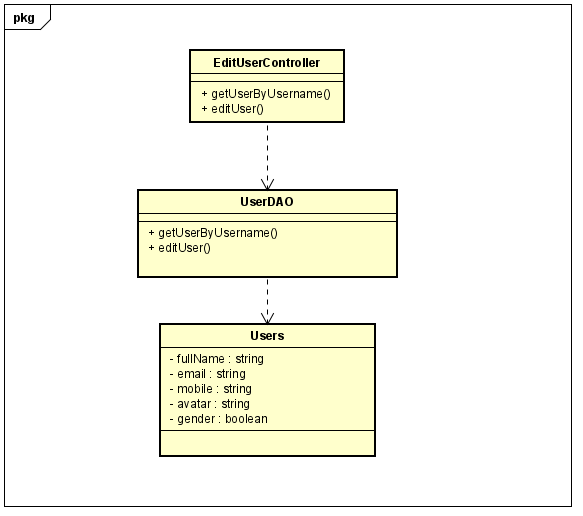


### d. Database queries

* **SELECT \* FROM users WHERE username=?**
* **INSERT INTO users (fullName, email, mobile, username, password) VALUES (?, ?, ?, ?, ?)**

## 7. <Common/User ProfileProfile>

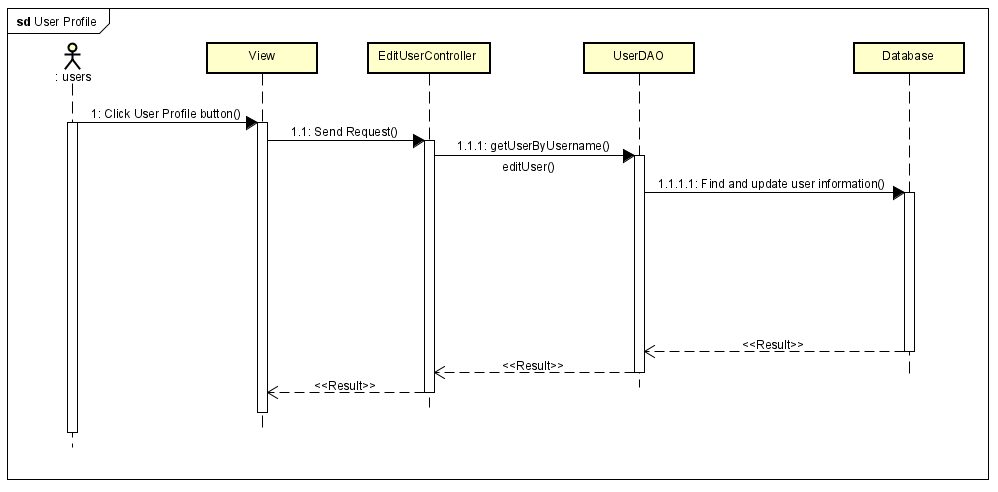
### a. Class Diagram



### b. Class Specifications

| **No** | **Method** | **Description** |
| --- | --- | --- |
| **1** | **getUserByUsername()** | **use to find account profile by it’s username** |
| **2** | **editUser()** | **use to update user’s profile** |

### c. Sequence Diagram(s)



### d. Database queries

* **SELECT fullName, gender, email, mobile, img**

**FROM users**

* **UPDATE users**

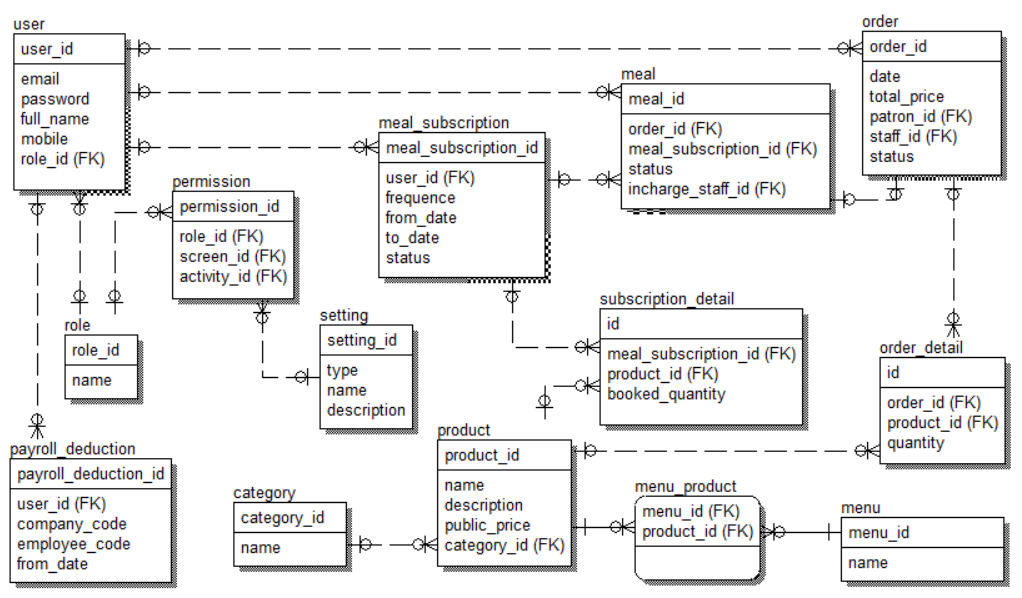
**SET fullName = ?, gender = ?, email = ?, mobile = ?, img = ?**

**WHERE username = ?**

# III. Database Design

## 1. Database Schema

*[Provide the tables relationship like example below – following MySQL database naming convention]*



## 2. Table Description

| **No** | **Table** | **Description** |
| --- | --- | --- |
| *01* | *<Table name>* | *<Description of the table>*  *- Primary keys: <<list of primary key fields>>*  *- Foreign keys: <<list of foreign key fields>>* |
| *02* |  |  |