

**FASHION SHOP ONLINE**

**Software Design Document**

– Hanoi, May 2021 –

**Table of Contents**

[**I. Overview**](#_heading=h.gjdgxs) **9**

[1. Code Packages/Namespaces](#_heading=h.30j0zll) 9

[2. Coding Conventions](#_heading=h.1fob9te) 10

[**II. Code Designs**](#_heading=h.3znysh7) **10**

[1. Public/Home Page](#_heading=h.2et92p0) 10

[a. Class Diagram](#_heading=h.tyjcwt) 10

[b. Class Specifications](#_heading=h.3dy6vkm) 11

[XYZ Class](#_heading=h.p90tm3z0w87w) 11

[c. Sequence Diagram(s)](#_heading=h.1t3h5sf) 11

[d. Database queries](#_heading=h.4d34og8) 11

[2. Public/Blogs List](#_heading=h.8d7epojxy9ga) 12

[a. Class Diagram](#_heading=h.omt4owacggr4) 12

[b. Class Specifications](#_heading=h.gygscju7xd8t) 12

[XYZ Class](#_heading=h.z9dusl5m8puj) 12

[c. Sequence Diagram(s)](#_heading=h.54cu8mem5j0t) 13

[d. Database queries](#_heading=h.y803i1r2rta0) 13

[3. Public/Blog Details](#_heading=h.ig47nmbi41wc) 14

[a. Class Diagram](#_heading=h.eod3gdhn6x23) 14

[b. Class Specifications](#_heading=h.iehk7k1x1so9) 14

[XYZ Class](#_heading=h.k7xbmijp2r2m) 14

[c. Sequence Diagram(s)](#_heading=h.eambmtnwvpq) 15

[d. Database queries](#_heading=h.1b4qmsby52sz) 15

[4. Public/Products List](#_heading=h.766zkvsn4liv) 16

[a. Class Diagram](#_heading=h.a02i5gaut3z2) 16

[b. Class Specifications](#_heading=h.yfz5ryzd15eg) 16

[c. Sequence Diagram(s)](#_heading=h.9kh0jbvh1vck) 17

[d. Database queries](#_heading=h.xkj7mfvbatym) 17

[5. Public/Product Details](#_heading=h.pls6cdhi9kt0) 18

[a. Class Diagram](#_heading=h.re0c1to7efhz) 18

[b. Class Specifications](#_heading=h.qi755iruo20r) 18

[XYZ Class](#_heading=h.8ma53fr1sw5y) 18

[c. Sequence Diagram(s)](#_heading=h.a0j3jqfi5err) 18

[d. Database queries](#_heading=h.xah7m2e0c9vq) 18

[6. Public/Cart Details](#_heading=h.lmp0k967ukgg) 18

[a. Class Diagram](#_heading=h.7tg9zhdctqgl) 18

[b. Class Specifications](#_heading=h.pc0pnw4mikdy) 19

[XYZ Class](#_heading=h.86qldaey7pdf) 19

[c. Sequence Diagram(s)](#_heading=h.cwgpamyf81vx) 19

[d. Database queries](#_heading=h.cq5o8th1wgij) 19

[7. Public/Cart Contact](#_heading=h.mapgg35kg2oi) 20

[a. Class Diagram](#_heading=h.bzyh24t99s5o) 20

[b. Class Specifications](#_heading=h.su13d5ae3ign) 20

[XYZ Class](#_heading=h.y4304ytp4ikp) 20

[c. Sequence Diagram(s)](#_heading=h.whc13qsxy0h) 21

[d. Database queries](#_heading=h.f2tei1dq1ygf) 21

[8. Public/Cart Completion](#_heading=h.m1rkpecb9kj9) 22

[a. Class Diagram](#_heading=h.nknm254ayqv3) 22

[b. Class Specifications](#_heading=h.tvd84j4fhax) 22

[XYZ Class](#_heading=h.thv475sunc7e) 22

[c. Sequence Diagram(s)](#_heading=h.tf1m194zjru1) 23

[d. Database queries](#_heading=h.x8txdn50pc) 23

[9. Public/Feedback](#_heading=h.1zvuv1nz5oc) 23

[a. Class Diagram](#_heading=h.jyb49az003j9) 24

[b. Class Specifications](#_heading=h.vuk238rm4k04) 24

[XYZ Class](#_heading=h.l4joaga854ss) 24

[c. Sequence Diagram(s)](#_heading=h.z8v60dgplff9) 25

[d. Database queries](#_heading=h.kdim1i9sbf3q) 25

[11. Common/Login](#_heading=h.lf7t7r5lfgzy) 26

[a. Class Diagram](#_heading=h.70phfl7osakb) 26

[b. Class Specifications](#_heading=h.c1fxhun0brma) 26

[c. Sequence Diagram(s)](#_heading=h.bi7mowoq2w43) 27

[d. Database queries](#_heading=h.ryzlm0wpxhhi) 27

[6. Common/Register](#_heading=h.q857my8y8qzf) 27

[a. Class Diagram](#_heading=h.lnc128wg4ebs) 28

[b. Class Specifications](#_heading=h.jzymlgy1zos1) 28

[c. Sequence Diagram(s)](#_heading=h.2qlf9371k415) 28

[d. Database queries](#_heading=h.x1d0i61yrer3) 29

[12. Common/Reset Password](#_heading=h.2s8eyo1) 29

[a. Class Diagram](#_heading=h.j4bzwh6s7rie) 29

[c. Sequence Diagram(s)](#_heading=h.w8pkbxcehddp) 30

[d. Database queries](#_heading=h.9ofbimtzb3ud) 30

[13. Common/Change Password](#_heading=h.74sp5wi727po) 31

[a. Class Diagram](#_heading=h.smlif03wjxgr) 31

[b. Class Specifications](#_heading=h.3ziabw3lg31f) 31

[c. Sequence Diagram(s)](#_heading=h.mlpfvphanpa5) 32

[d. Database queries](#_heading=h.nz63qosropyd) 32

[14. Common/User Profile](#_heading=h.rn2u37hula5m) 32

[a. Class Diagram](#_heading=h.72f1isxd5voz) 33

[b. Class Specifications](#_heading=h.agg2h4wfj6ar) 33

[c. Sequence Diagram(s)](#_heading=h.necaobbpm0h0) 33

[d. Database queries](#_heading=h.aygvxv50x4gf) 34

[15. Common/User Authorization](#_heading=h.hkd5yj2850mr) 35

[a. Class Diagram](#_heading=h.hup624p1yuk6) 35

[b. Class Specifications](#_heading=h.pujr7woef4qb) 35

[c. Sequence Diagram(s)](#_heading=h.9y7xtinnxks1) 36

[d. Database queries](#_heading=h.odevy8ip9vv5) 36

[16. Customer/My Orders](#_heading=h.opovx5fsbz3j) 36

[a. Class Diagram](#_heading=h.5ppwbs1yoe4c) 37

[b. Class Specifications](#_heading=h.tyg4qa7lurb9) 37

[XYZ Class](#_heading=h.qr6jvnxoxxri) 37

[c. Sequence Diagram(s)](#_heading=h.9zeu7rrfd55m) 37

[d. Database queries](#_heading=h.hyoow113m2y1) 38

[17. Customer/Order Information](#_heading=h.dnruueubawiw) 39

[a. Class Diagram](#_heading=h.y01lxrqe55ky) 39

[b. Class Specifications](#_heading=h.498tp35rhnyy) 39

[XYZ Class](#_heading=h.yzb7wpsho0ln) 39

[c. Sequence Diagram(s)](#_heading=h.lwetguu4gy3e) 39

[d. Database queries](#_heading=h.far7pel3usg) 40

[18. Marketing/MKT Dashboard](#_heading=h.bs38qeam81m) 41

[a. Class Diagram](#_heading=h.jltfkos2h72e) 41

[b. Class Specifications](#_heading=h.bo2u4wfe9gx) 41

[XYZ Class](#_heading=h.cxonfzlh5zem) 41

[c. Sequence Diagram(s)](#_heading=h.sfgh023qwgm4) 41

[d. Database queries](#_heading=h.v4ogmmv839zf) 42

[19. Marketing/Posts List](#_heading=h.4yoegnval7x0) 42

[a. Class Diagram](#_heading=h.fs8lo1arx8wc) 43

[b. Class Specifications](#_heading=h.dw936i56mxzx) 43

[XYZ Class](#_heading=h.ssdv9ove0rha) 43

[c. Sequence Diagram(s)](#_heading=h.yuvg358h6col) 43

[d. Database queries](#_heading=h.jmb1vtjbmuga) 44

[20. Marketing/Post Details](#_heading=h.ja1vgu73j75i) 45

[a. Class Diagram](#_heading=h.dvdwpkyrgwoe) 45

[b. Class Specifications](#_heading=h.161aj7y0vr9f) 45

[XYZ Class](#_heading=h.ms4y8r3vwqkp) 45

[c. Sequence Diagram(s)](#_heading=h.lq5tt2s91ocv) 46

[d. Database queries](#_heading=h.salcpqtdj29) 46

[21. Marketing/Sliders List](#_heading=h.y8ftwher3mon) 47

[a. Class Diagram](#_heading=h.r9xqvulk1u5o) 47

[b. Class Specifications](#_heading=h.on0bfwq51a33) 47

[XYZ Class](#_heading=h.koordmsfg664) 47

[c. Sequence Diagram(s)](#_heading=h.4a0u2t63eqlc) 47

[d. Database queries](#_heading=h.kq1wzgneaha) 48

[22. Marketing/Slider Details](#_heading=h.wbkb6e7dnuhi) 49

[a. Class Diagram](#_heading=h.iu6us8sp7vf3) 49

[b. Class Specifications](#_heading=h.j9vfltfeftzs) 49

[XYZ Class](#_heading=h.ixvbe7ur88eh) 49

[c. Sequence Diagram(s)](#_heading=h.4bb2a020i3y7) 49

[d. Database queries](#_heading=h.p0tmzrmrh1p) 50

[23. Marketing/Products List](#_heading=h.ekh717a18wwc) 50

[a. Class Diagram](#_heading=h.2qtbg0a5swjf) 50

[b. Class Specifications](#_heading=h.trgbymfk2ati) 50

[c. Sequence Diagram(s)](#_heading=h.o42xkavfugxk) 51

[d. Database queries](#_heading=h.bprf2xaerwb8) 51

[24. Marketing/Product Details](#_heading=h.naukl6j79m02) 52

[a. Class Diagram](#_heading=h.oyx5zwb4vjql) 52

[b. Class Specifications](#_heading=h.k21boxna2cgz) 52

[XYZ Class](#_heading=h.rrehh9f11nlz) 52

[c. Sequence Diagram(s)](#_heading=h.1mie99rvop63) 53

[d. Database queries](#_heading=h.cmdpj7be09ox) 53

[25. Marketing/Customers List](#_heading=h.oll9vt44kudn) 55

[a. Class Diagram](#_heading=h.wgkg93aoi8hh) 55

[b. Class Specifications](#_heading=h.d60ohlal6g7n) 55

[c. Sequence Diagram(s)](#_heading=h.84akea6db8zl) 55

[d. Database queries](#_heading=h.gaoodcdpa707) 56

[26. Marketing/Customer Details](#_heading=h.70trtzvzjw95) 57

[a. Class Diagram](#_heading=h.fczrncww9522) 57

[b. Class Specifications](#_heading=h.3y0tr1oz6l16) 57

[c. Sequence Diagram(s)](#_heading=h.vcoyncpt8ig2) 57

[d. Database queries](#_heading=h.fwarl72see03) 58

[27. Marketing/Feedbacks List](#_heading=h.7d5mj5xoas17) 59

[a. Class Diagram](#_heading=h.8cxh4pdtiwb9) 59

[b. Class Specifications](#_heading=h.du6pfil0wmcv) 59

[XYZ Class](#_heading=h.u0rn723348gx) 59

[c. Sequence Diagram(s)](#_heading=h.bwjbmhapp41i) 59

[d. Database queries](#_heading=h.90t07vtlf23l) 60

[28. Marketing/Feedback Details](#_heading=h.qlzsixk425r5) 60

[a. Class Diagram](#_heading=h.k5oed8fwgtuw) 60

[b. Class Specifications](#_heading=h.9byf1e3u5svl) 60

[XYZ Class](#_heading=h.cmvxttlwebnq) 60

[c. Sequence Diagram(s)](#_heading=h.rqlts3vsy7z8) 60

[d. Database queries](#_heading=h.6ftczynfrad4) 61

[29. Sale/Sale Dashboard](#_heading=h.onvajjnxit9u) 61

[a. Class Diagram](#_heading=h.37kuyj24vkxj) 62

[b. Class Specifications](#_heading=h.l10rkmlz43fm) 62

[XYZ Class](#_heading=h.csai4sr3u9np) 62

[c. Sequence Diagram(s)](#_heading=h.e6znc0e7bnhj) 63

[d. Database queries](#_heading=h.82p6oz311hkd) 63

[30. Sale/Orders List](#_heading=h.zazuq3g216v9) 64

[a. Class Diagram](#_heading=h.yqbo8l55h6hf) 65

[b. Class Specifications](#_heading=h.b20zo2q7lp49) 65

[XYZ Class](#_heading=h.pp885kpujt34) 65

[c. Sequence Diagram(s)](#_heading=h.823akbu8r9wm) 65

[d. Database queries](#_heading=h.13ap5yhh6598) 66

[31. Sale/Order Details](#_heading=h.68j710o4ki7g) 66

[a. Class Diagram](#_heading=h.h4clbsh9whh6) 67

[b. Class Specifications](#_heading=h.ezp36xiin879) 67

[XYZ Class](#_heading=h.8uh7f8u7v3cj) 67

[c. Sequence Diagram(s)](#_heading=h.ogf6foo1nbis) 67

[d. Database queries](#_heading=h.hhkorlbod0ev) 68

[32. Admin/Admin Dashboard](#_heading=h.ea47ljqc02hm) 69

[a. Class Diagram](#_heading=h.5qr9zewjdjxk) 69

[b. Class Specifications](#_heading=h.xkd9nvlcbfrc) 69

[XYZ Class](#_heading=h.on6mkg6t6g43) 69

[c. Sequence Diagram(s)](#_heading=h.kzn3f24021f4) 69

[d. Database queries](#_heading=h.86snmxl2hebe) 70

[33. Admin/Users List](#_heading=h.jpwprfnfhptk) 70

[a. Class Diagram](#_heading=h.4vfvgnegijkt) 71

[b. Class Specifications](#_heading=h.xxgjp9n4rtqc) 71

[XYZ Class](#_heading=h.pfaew7rdzbmy) 71

[c. Sequence Diagram(s)](#_heading=h.79dqo43ivdsm) 71

[d. Database queries](#_heading=h.pgbg60g3jbfo) 72

[34. Admin/User Details](#_heading=h.qe23r38zbyhl) 72

[a. Class Diagram](#_heading=h.jms229k6rb0v) 73

[b. Class Specifications](#_heading=h.8icw83ajwrw3) 73

[XYZ Class](#_heading=h.8o2rt94y6qv6) 73

[c. Sequence Diagram(s)](#_heading=h.i8v998efxtln) 73

[d. Database queries](#_heading=h.xwydr1c5lg60) 73

[35. Admin/Settings List](#_heading=h.ygnpcur9a9i) 73

[a. Class Diagram](#_heading=h.93n5ne368ywa) 73

[b. Class Specifications](#_heading=h.2ry1q9qhlse8) 74

[c. Sequence Diagram(s)](#_heading=h.quq30k1ddmck) 74

[d. Database queries](#_heading=h.pfpgn1aoei4u) 75

[36. Admin/Setting Details](#_heading=h.uut8oz8hujvh) 75

[a. Class Diagram](#_heading=h.3xkanxc2kg7w) 75

[b. Class Specifications](#_heading=h.q7vr3p8n2z0f) 75

[XYZ Class](#_heading=h.v24k0vc25knx) 75

[c. Sequence Diagram(s)](#_heading=h.vug7hnox85s) 75

[d. Database queries](#_heading=h.9v89uxra7fpf) 76

[37. Public/Blog List](#_heading=h.tzyhveb5gurs) 76

[a. Class Diagram](#_heading=h.essiqxrzd32o) 77

[b. Class Specifications](#_heading=h.rn4okvu43tv8) 77

[XYZ Class](#_heading=h.skt714z8inka) 77

[c. Sequence Diagram(s)](#_heading=h.12ljjws97qxe) 77

[d. Database queries](#_heading=h.iv4ldsip1kh5) 78

[38. Public/ Blog Details](#_heading=h.d8lbqz75in15) 78

[a. Class Diagram](#_heading=h.tmbe7mdxqcdd) 79

[b. Class Specifications](#_heading=h.c573e5ip7ora) 79

[XYZ Class](#_heading=h.l8e4lvhb5t9k) 79

[c. Sequence Diagram(s)](#_heading=h.kvmytqsy70sy) 80

[d. Database queries](#_heading=h.jr2ynz8wmzxt) 80

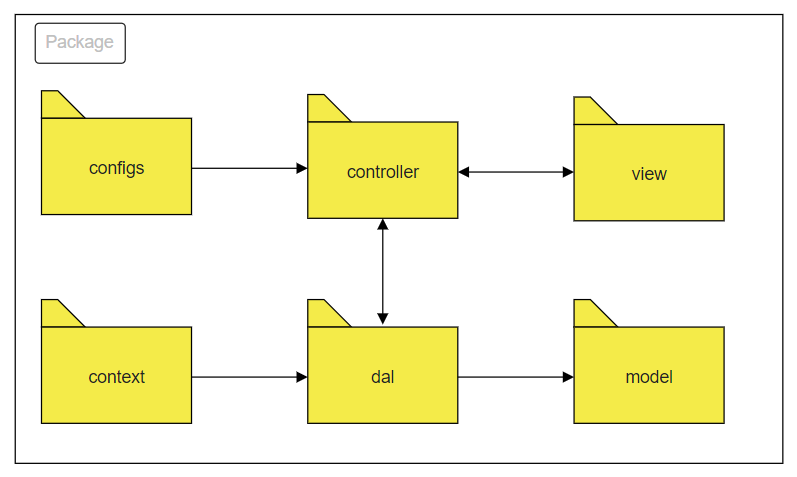
[**III. Database Design**](#_heading=h.17dp8vu) **80**

[1. Database Schema](#_heading=h.3r9tkbxldg4) 81

[2. Table Description](#_heading=h.coef5r48r46k) 81

# I. Overview

## 1. Code Packages/Namespaces



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

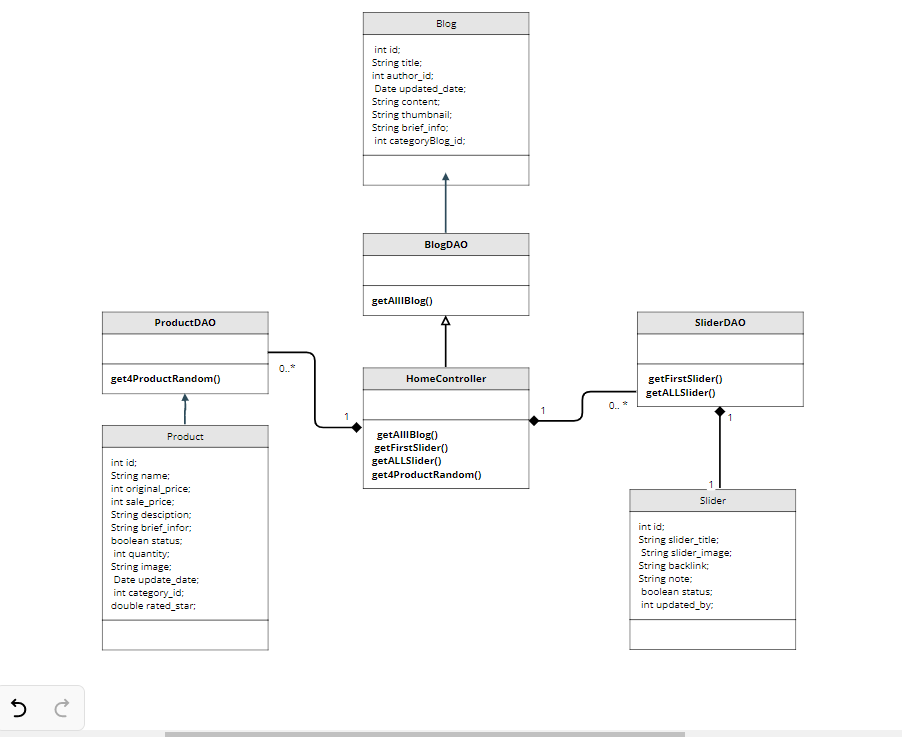
| **No** | **Package** | **Description** |
| --- | --- | --- |
| *01* | *controller* | *Acts as a go-between for the service layer and the Front-end, passing on different input requests*  *<Class naming convention>* |
| *02* | *dal* | *This component is used to get data directly FROM the database*  *<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

### a. Class Diagram

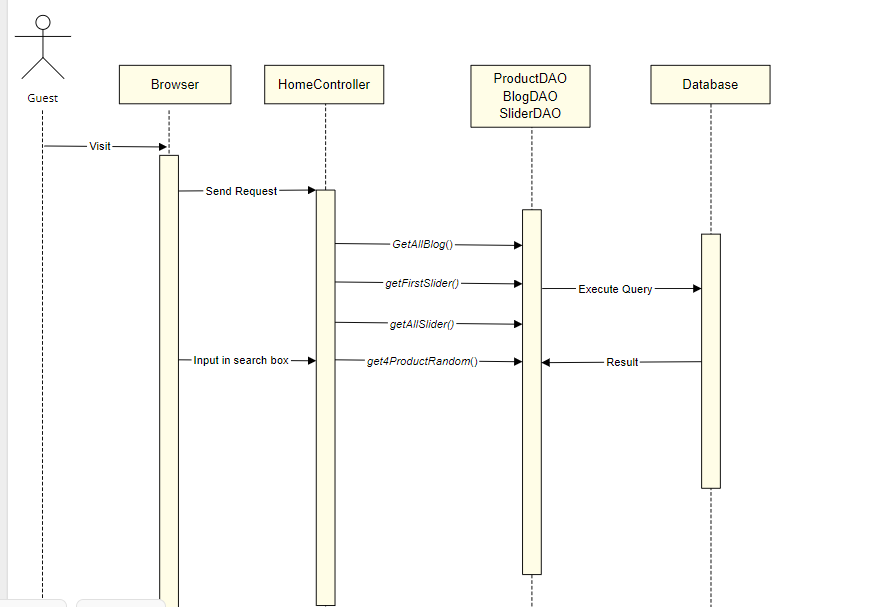


### b. Class Specifications

#### XYZ Class

| **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)



### d. Database queries

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

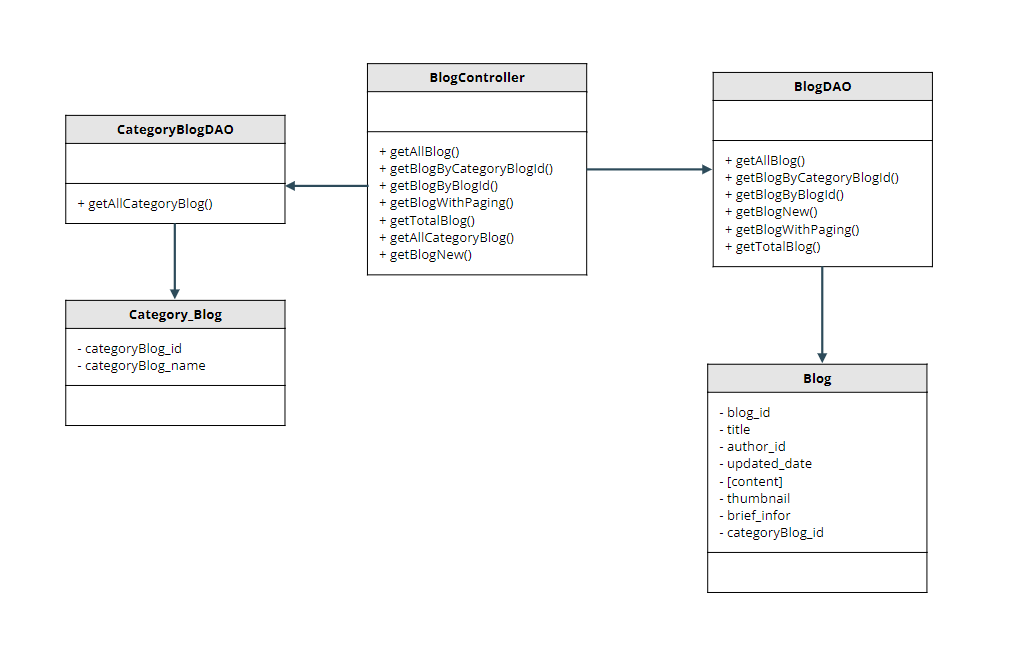
SELECT \* FROM [dbo].[Slider]

Select \* from Blog

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

## 2. Public/Blogs List

### a. Class Diagram

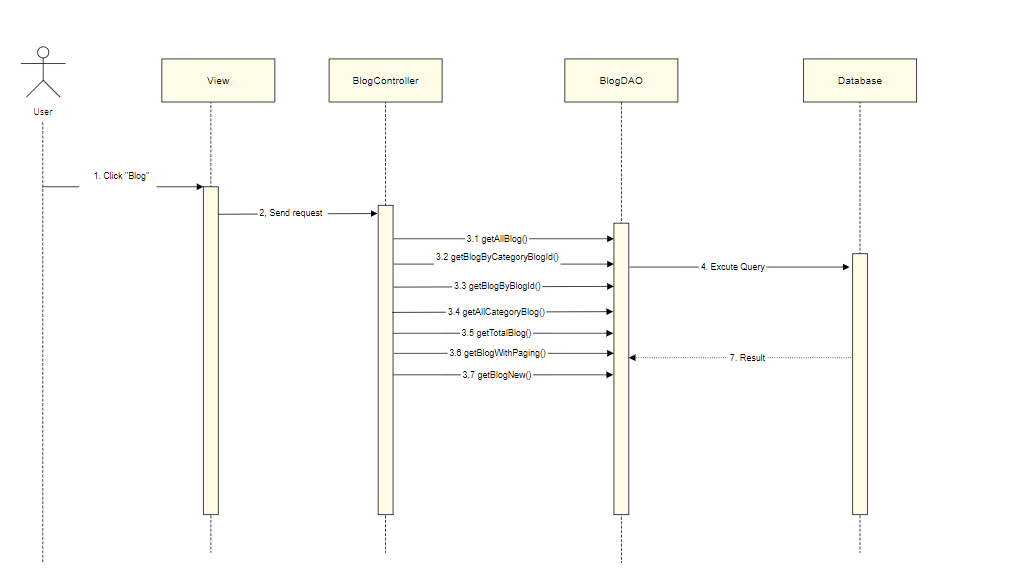


### b. Class Specifications

#### XYZ Class

| **No** | **Method** | **Description** |
| --- | --- | --- |
| *01* | *getAllBlog* | *get all blog* |
| *02* | *getBlogByCategoryBlogId* | *get blog by category id* |
| *03* | *getBlogByBlogId* | *get blog by blog id* |
| *04* | *getBlogWithPaging* | *git blog with paging* |
| *05* | *getTotalBlog* | *get total blog* |
| *06* | *getAllCategoryBlog* | *get all category blog* |
| *07* | *getBlogNew* | *get blog new* |

### c. Sequence Diagram(s)



### d. Database queries

Select top 2 \* from Blog order by newid()

Select \* from Blog where categoryBlog\_id = ?

Select \* from Blog where blog\_id = ?

select top 1 \* from Blog order by updated\_date desc

Select count(\*) from Blog "

+ "where categoryBlog\_id " + categoryId + " and status = 1 and title like N'%" + searchKey + "%'\n

select \* from Blog\n"

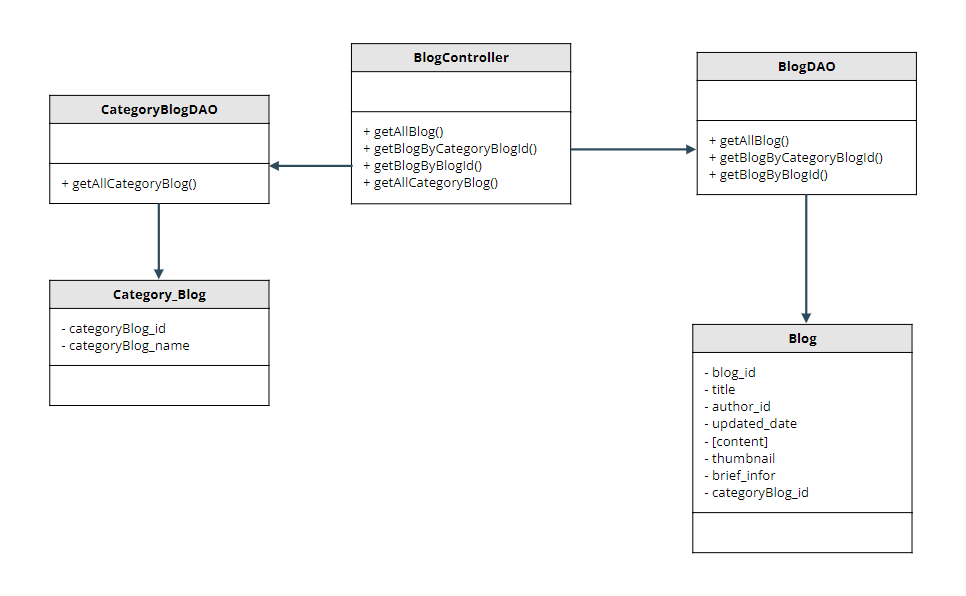
+ "where categoryBlog\_id " + categoryId + " and status = 1 and title like N'%" + searchKey + "%'\n"

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

Select \* from Category\_Blog and status = 1

## 3. Public/Blog Details

### a. Class Diagram



### b. Class Specifications

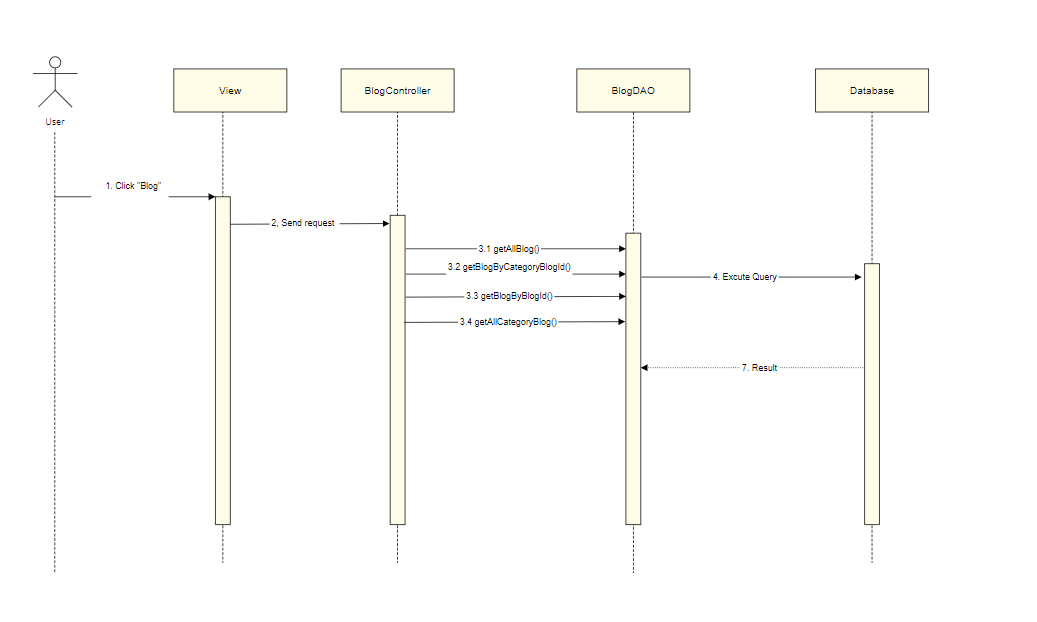
#### XYZ Class

#### 

| **No** | **Method** | **Description** |
| --- | --- | --- |
| *01* | *getAllBlog* | *get all blog* |
| *02* | *getBlogByCategoryBlogId* | *get blog by category id* |
| *03* | *getBlogByBlogId* | *get blog by blog id* |
| *04* | *getAllCategoryBlog* | *get all category blog* |

### 

### c. Sequence Diagram(s)



### d. Database queries

Select top 2 \* from Blog order by newid()

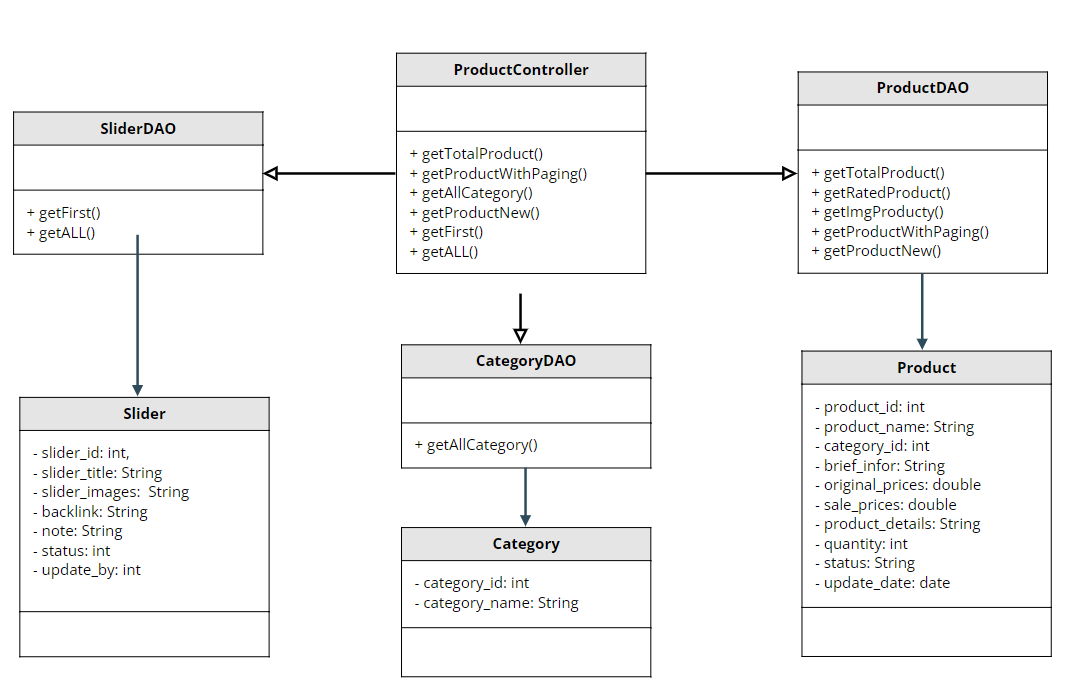
Select \* from Blog where categoryBlog\_id = ?

Select \* from Blog where blog\_id = ?

Select \* from Category\_Blog and status = 1

## 4. Public/Products List

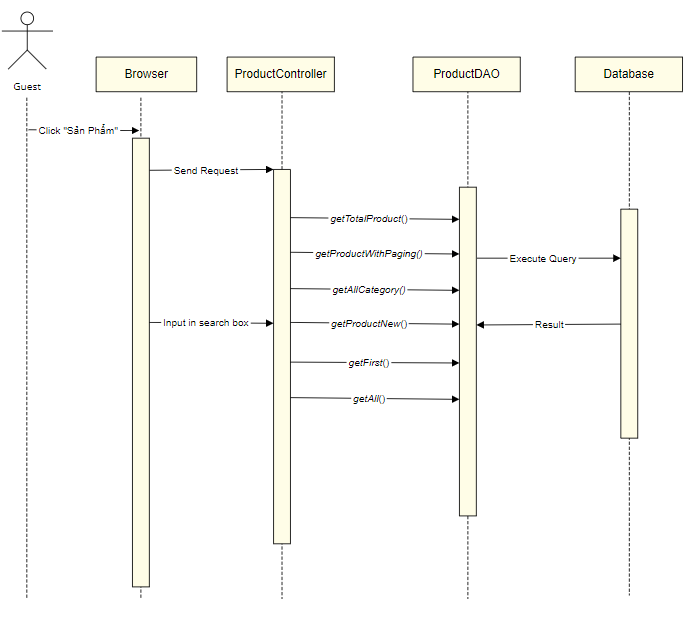
### a. Class Diagram



### 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]

## 5. Public/Product Details

### a. Class Diagram

### b. Class Specifications

#### XYZ Class

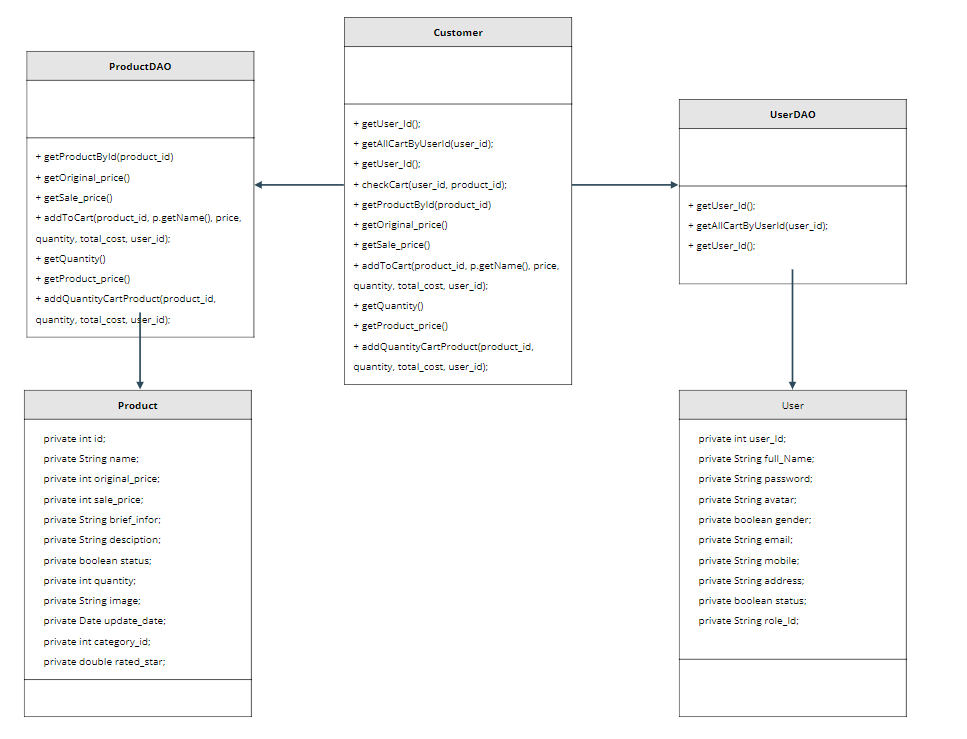
| **No** | **Method** | **Description** |
| --- | --- | --- |
| *01* |  |  |
| *02* |  |  |
| *03* |  |  |
| *04* |  |  |

### c. Sequence Diagram(s)

### d. Database queries

## 6. Public/Cart Details

### a. Class Diagram

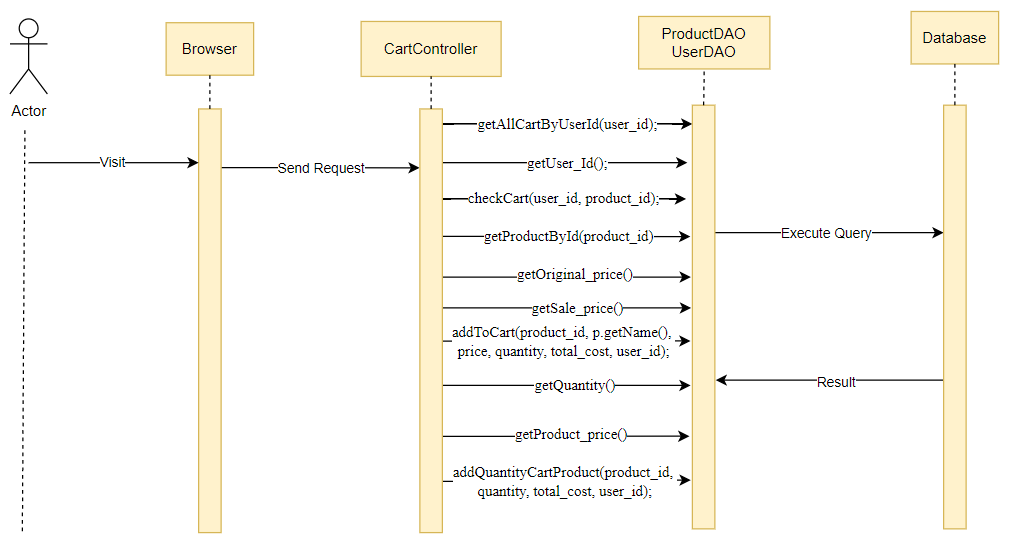


### b. Class Specifications

#### XYZ Class

| **No** | **Method** | **Description** |
| --- | --- | --- |
| *01* | *getAllCartByUserId(user\_id);* | *get all product by userID in Database* |
| *02* | *getUser\_Id();* | *get userID in User Database* |
| *03* | *checkCart(user\_id, product\_id);* | *Check product in cart by user\_id in user and product\_Id in product* |
| *04* | *getProductById(product\_id)* | *get Product by Id\_producrt in the product modal* |
| *05* | *getOriginal\_price()* | *get original\_price in product in database* |
| *06* | *getSale\_price()* | *get sale\_price in product in Database* |
| *07* | *addToCart(product\_id, p.getName(), price, quantity, total\_cost, user\_id);* | *save new product in database cart* |
| *08* | *getQuantity()* | *get number at table product in database* |
| *09* | *getProduct\_price()* | *Last value at table product in database* |
| *10* | *addQuantityCartProduct(product\_id, quantity, total\_cost, user\_id);* | *add new number at table product to database* |

### c. Sequence Diagram(s)



### d. Database queries

UPDATE [dbo].[Cart]\n"

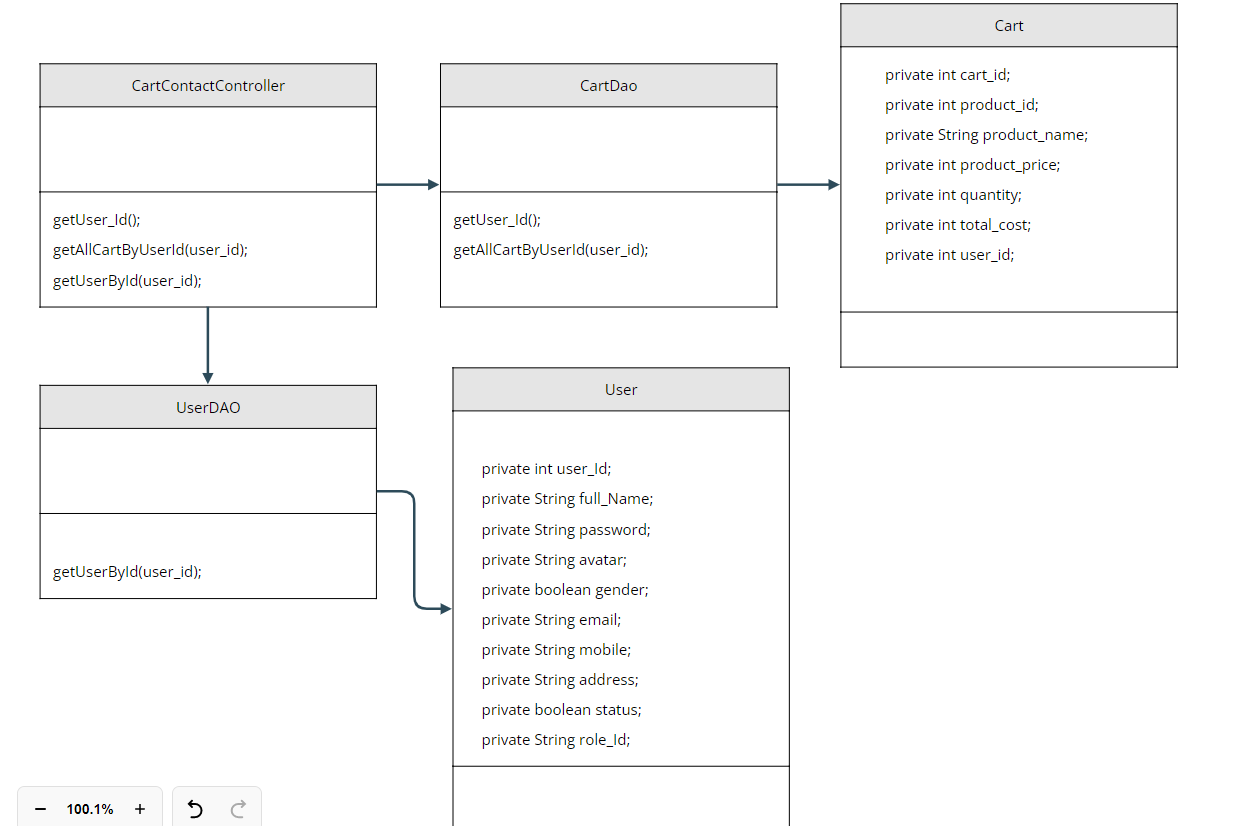
+ " SET [quantity] = ? \n"

+ " ,[total\_cost] = ? \n"

+ " WHERE [product\_id] = ? and [user\_id] = ?"

## 7. Public/Cart Contact

### a. Class Diagram

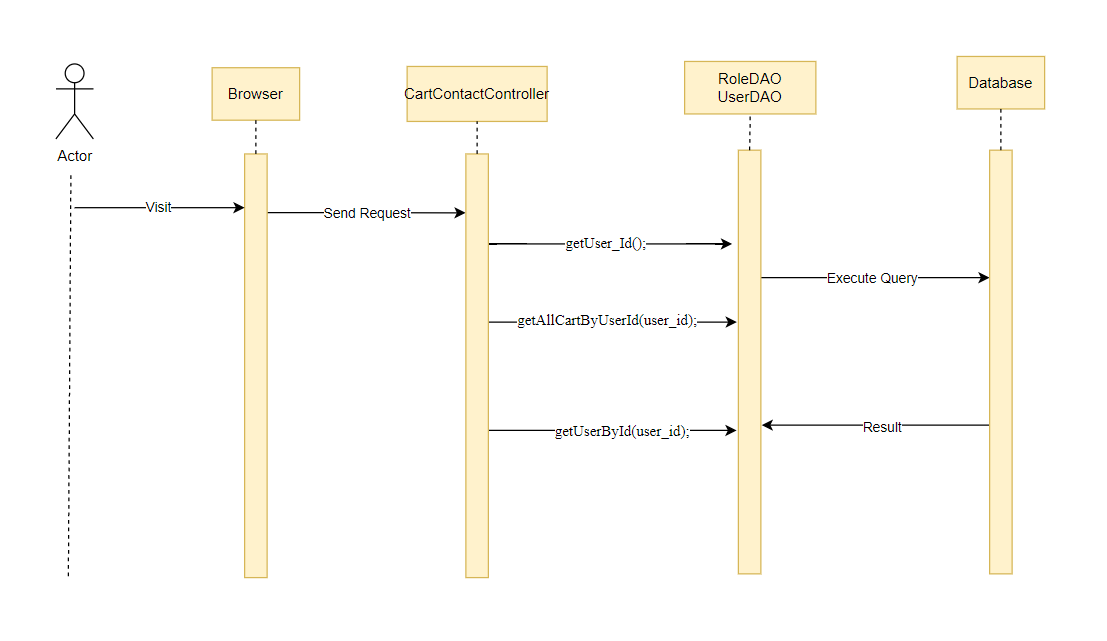


### b. Class Specifications

#### XYZ Class

| **No** | **Method** | **Description** |
| --- | --- | --- |
| *01* | *getUser\_Id();* | *get User by id in table cart database* |
| *02* | *getAllCartByUserId(user\_id);* | *get all carrt by User in table cart database* |
| *03* | *getUserById(user\_id);* | *get User by usser in table database* |

### c. Sequence Diagram(s)



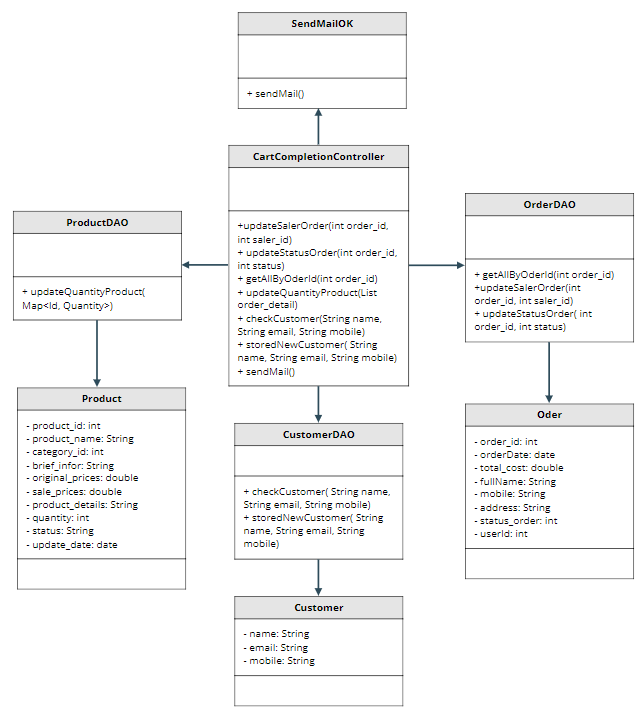
### d. Database queries

select \* from Cart where [user\_id] = ?

select \* from [User] where userId = ?

## 8. Public/Cart Completion

### a. Class Diagram

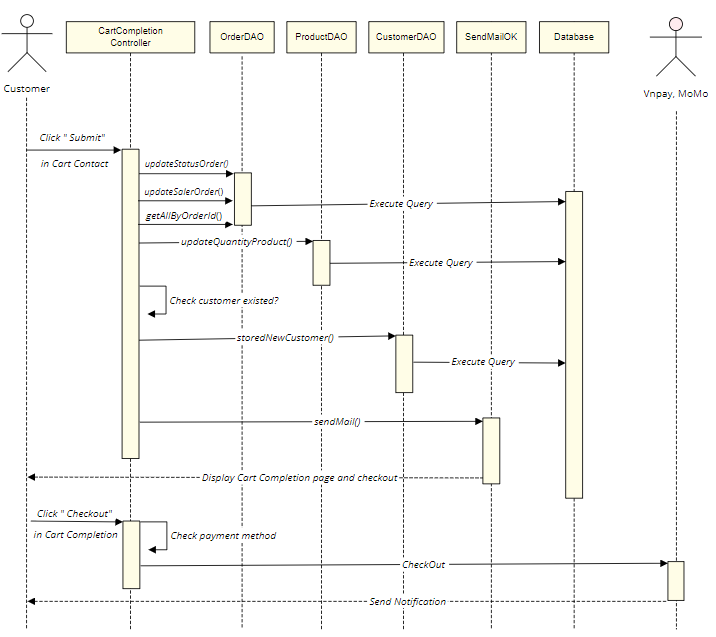


### b. Class Specifications

#### XYZ Class

| **No** | **Method** | **Description** |
| --- | --- | --- |
| *01* | *updateSalerOrder()* | *Update saler random for order selected* |
| *02* | *updateStatusOrder()* | *Update status for order selected* |
| *03* | *getAllByOrderId()* | *Get all information of specific order* |
| *05* | *updateQuantityProduct()* | *Update quantity for each product in order* |
| *06* | *checkCustomer()* | *Check customer existed* |
| *07* | *storedNewCustomer()* | *Stored new customer in the database* |
| *08* | *sendMail()* | *Send mail to customer* |

### c. Sequence Diagram(s)

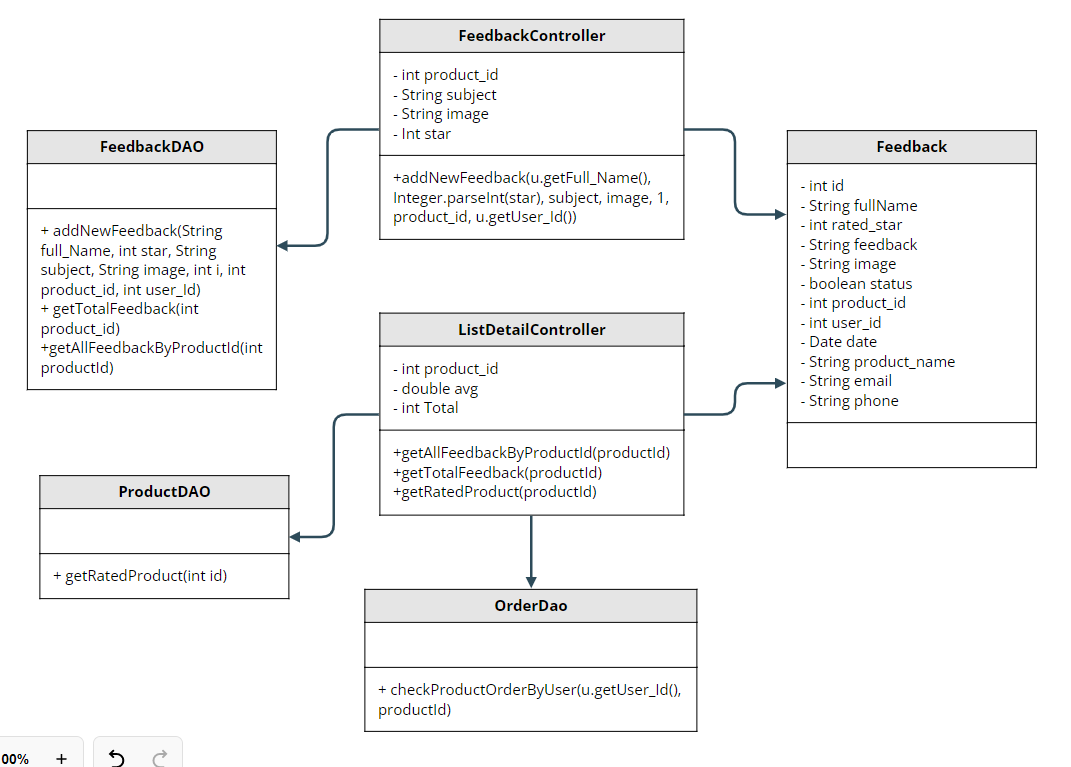


### d. Database queries

* UPDATE [dbo].[Order] SET [saler\_id] = ? WHERE order\_id = ?
* UPDATE [dbo].[Order] SET [status\_order] = ? WHERE order\_id = ?
* SELECT \* From Order\_Detail where order\_id = ?
* UPDATE [dbo].[Product] SET [quantity] = (quantity - ? ) WHERE product\_id = ?
* SELECT [customer\_id], [customer\_name], [customer\_email], [customer\_mobile] FROM [dbo].[Customer] WHERE customer\_name = ? and customer\_email = ? and customer\_mobile = ?
* INSERT INTO [dbo].[Customer] ([customer\_name], [customer\_email], [customer\_mobile]) VALUES (?,?,?)

## 9. Public/Feedback

### a. Class Diagram

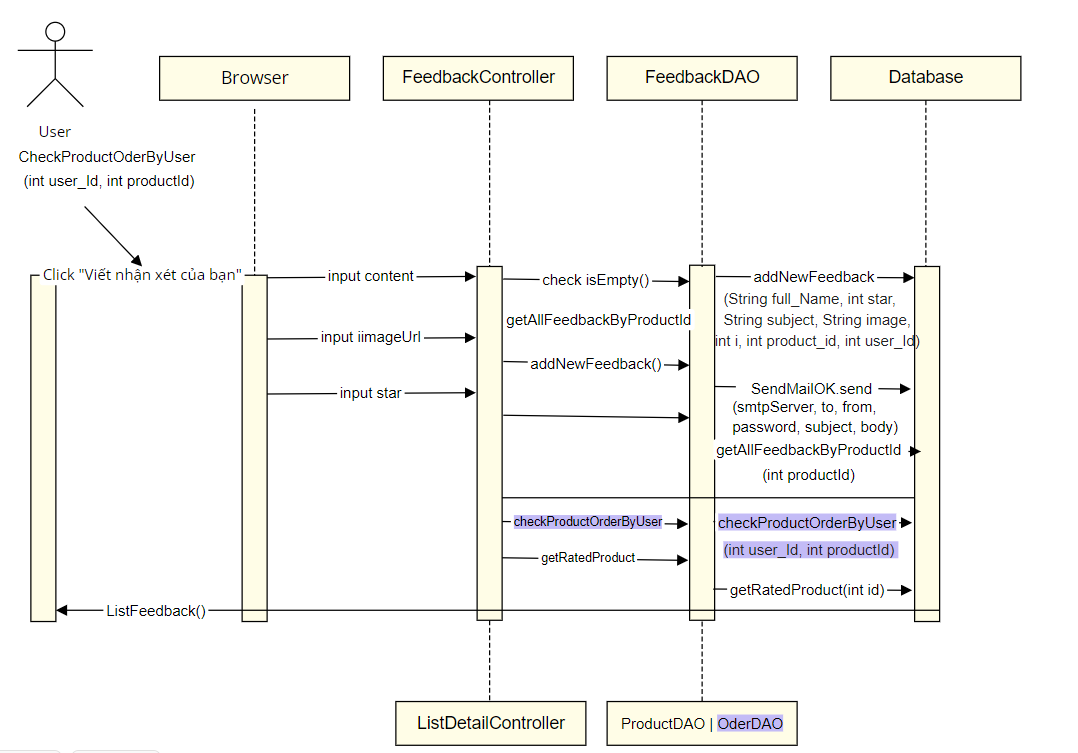


### b. Class Specifications

#### XYZ Class

| **No** | **Method** | **Description** |
| --- | --- | --- |
| *01* | *addNewFeedback(String full\_Name, int star, String subject, String image, int i, int product\_id, int user\_Id)* | *Add new feedback by userid and productid* |
| *02* | *getTotalFeedback(int product\_id)* | *Total feedback for show number of all feedback fora product* |
| *03* | *getAllFeedbackByProductId(int productId)* | *Get all feedback for show all feedback for a product* |
| *04* | *getRatedProduct(int id)* | *Avg rated star for a product* |
| *05* | *checkProductOrderByUser(u.getUser\_Id(), productId)* | *Check user who purchased the product* |

### c. Sequence Diagram(s)



### d. Database queries

-INSERT INTO [dbo].[Feedback]\n"

+ " ([fullName]\n"

+ " ,[rated\_star]\n"

+ " ,[feedback]\n"

+ " ,[image]\n"

+ " ,[status]\n"

+ " ,[product\_id]\n"

+ " ,[userId])\n"

+ " VALUES\n"

+ " (?,?,?,?,?,?,?)

-select COUNT(feedBack\_id) from Feedback where product\_id = ?

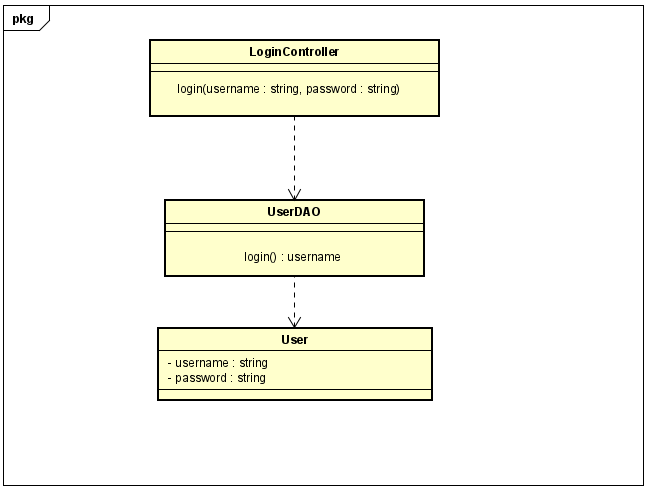
-checkProductOrderByUser(int user\_Id, int productId)

-getAllFeedbackByProductId(int productId)

-select AVG(rated\_star) from Feedback where product\_id = ?

## 11. Common/Login

### a. Class Diagram

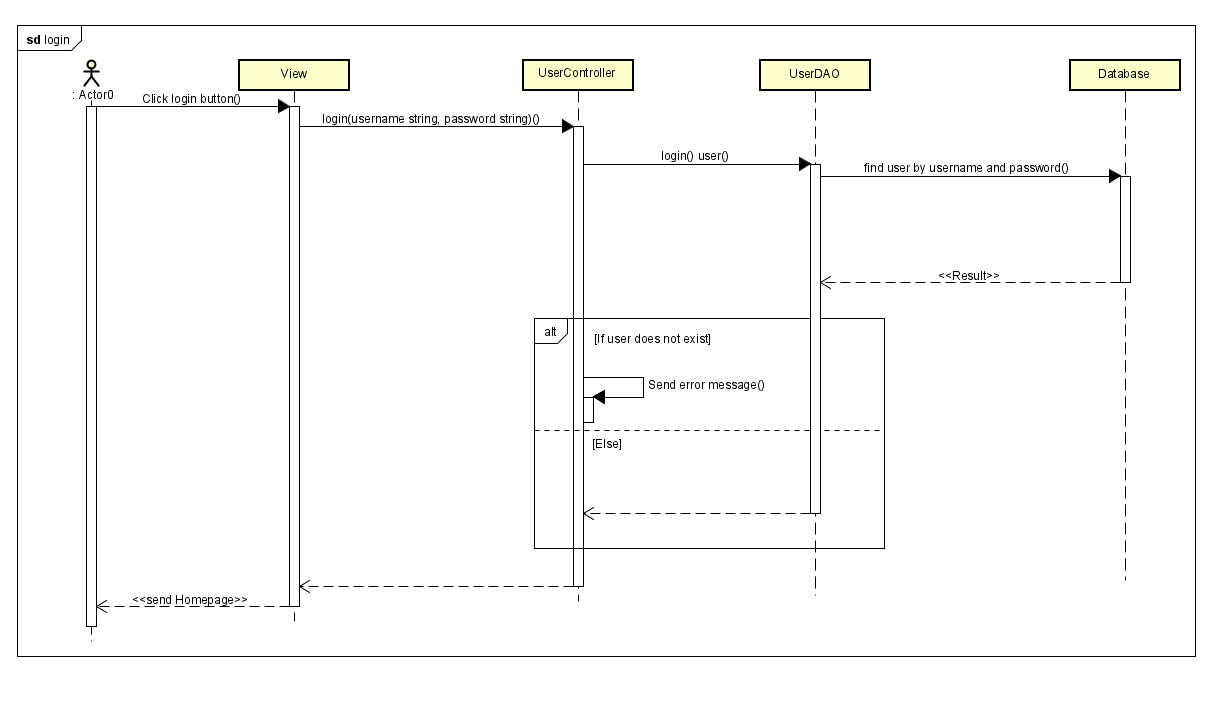


### b. Class Specifications

#### 

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

### c. Sequence Diagram(s)

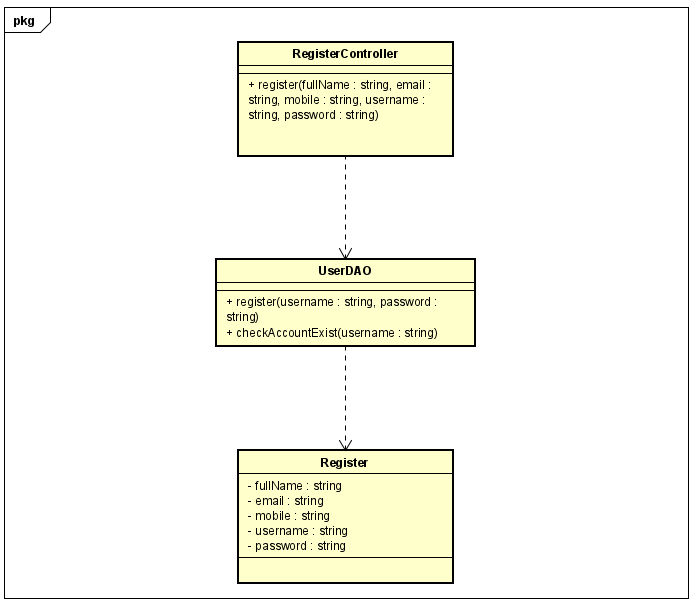


### d. Database queries

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

## 6. Common/Register

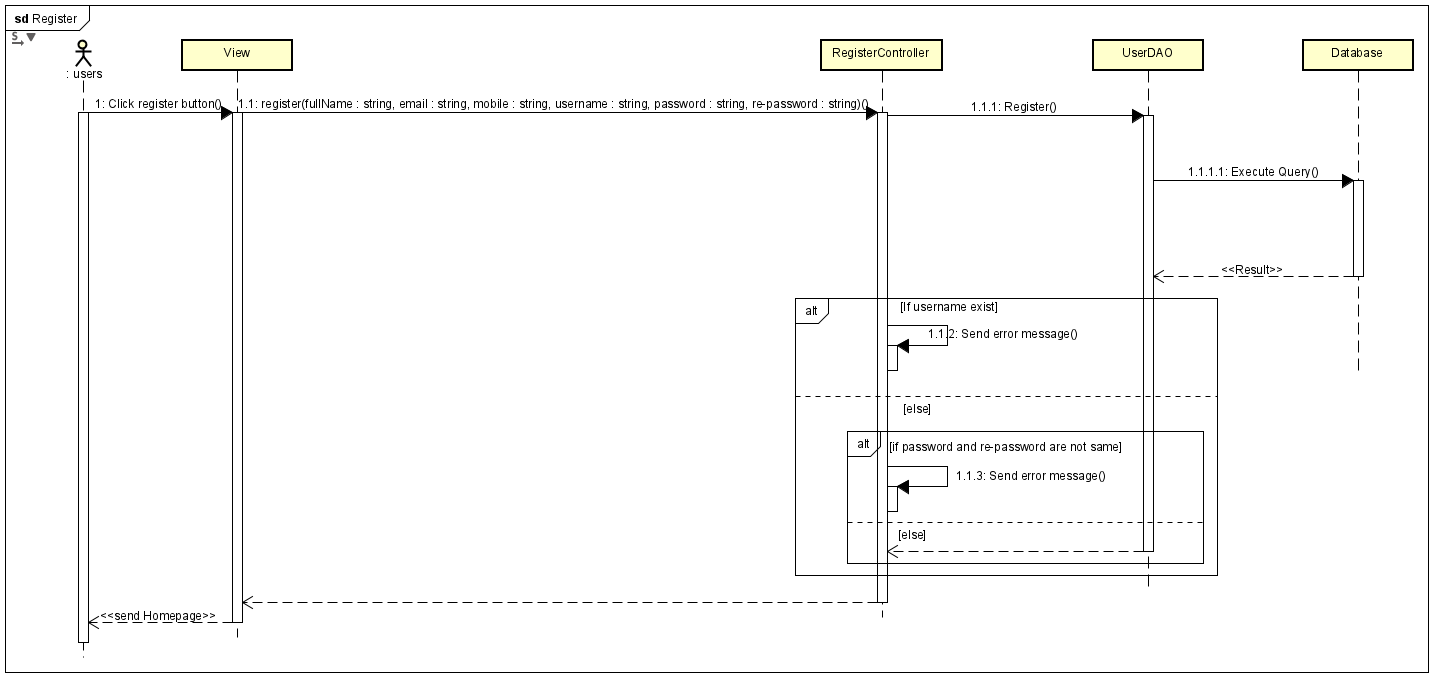
### a. Class Diagram



### b. Class Specifications

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

### c. Sequence Diagram(s)

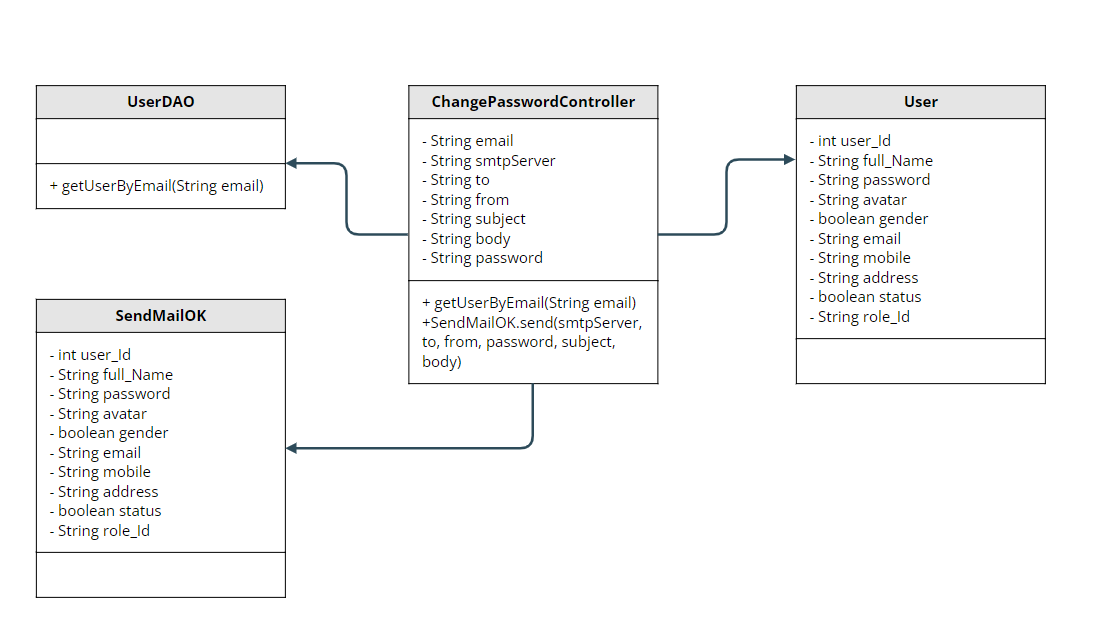


### d. Database queries

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

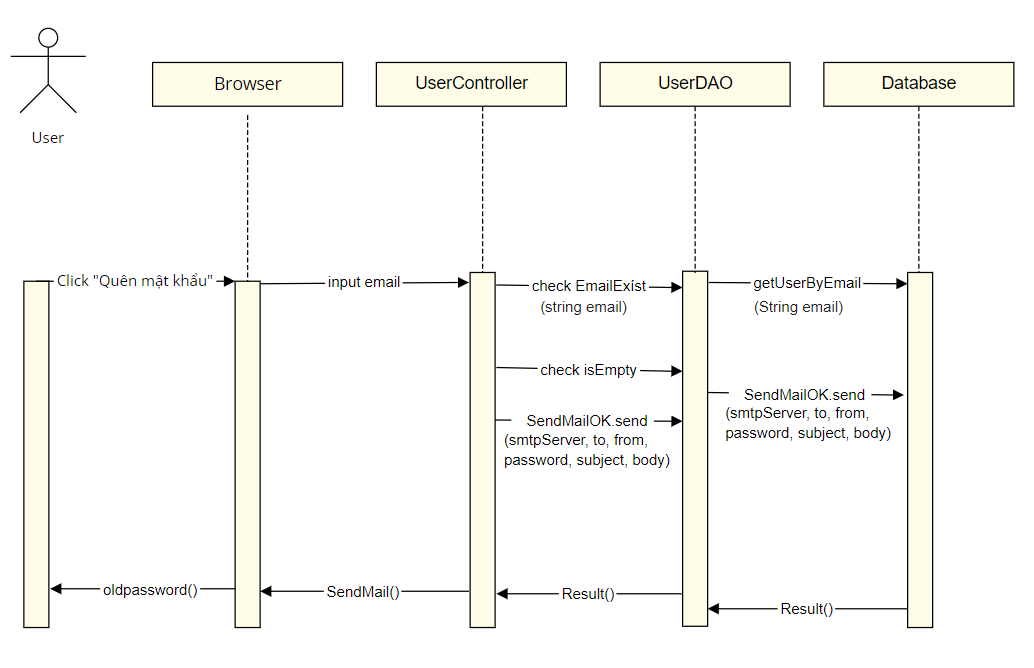
## 12. Common/Reset Password

### a. Class Diagram

**b. Class Specifications**

| **No** | **Method** | **Description** |
| --- | --- | --- |
| *01* | *getUserByEmail(String email)* | *Check email existence* |
| *02* | *SendMailOK.send(smtpServer, to, from, password, subject, body)* | *Send a link that leads the user to the reset password page* |

### c. Sequence Diagram(s)

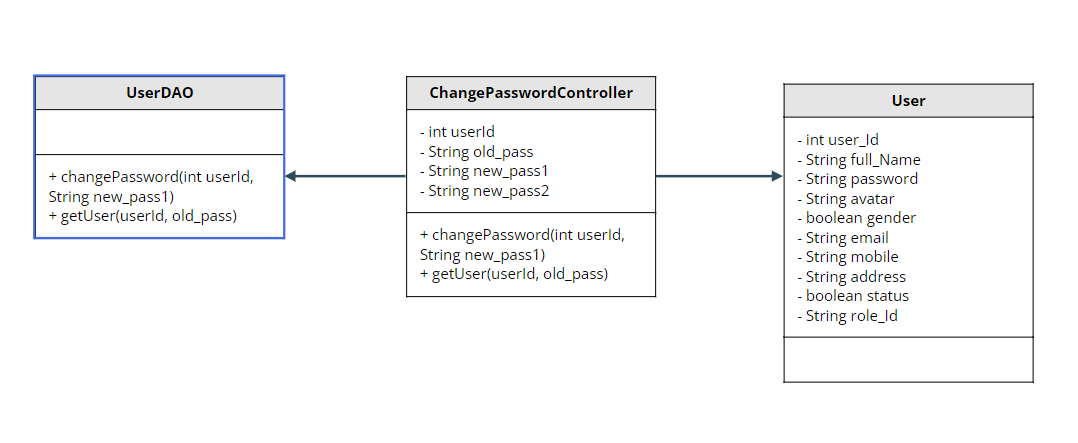


### d. Database queries

* select \* from [User] where email = ?

## 13. Common/Change Password

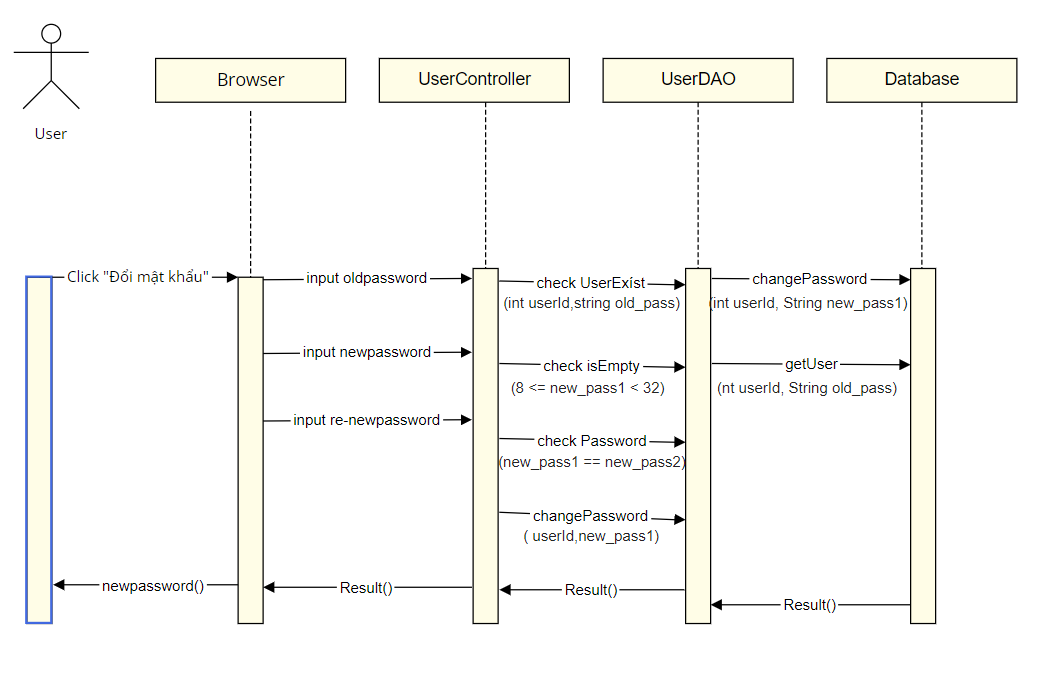
### a. Class Diagram



### b. Class Specifications

| **No** | **Method** | **Description** |
| --- | --- | --- |
| *01* | *changePassword(int userId, String new\_pass1)* | *Change Password* |
| *02* | *getUser(int userId, String old\_pass)* | *Check logged in* |

### c. Sequence Diagram(s)



### d. Database queries

* select \* from [User] where userId = ? and password = ?
* UPDATE [dbo].[User]\n"

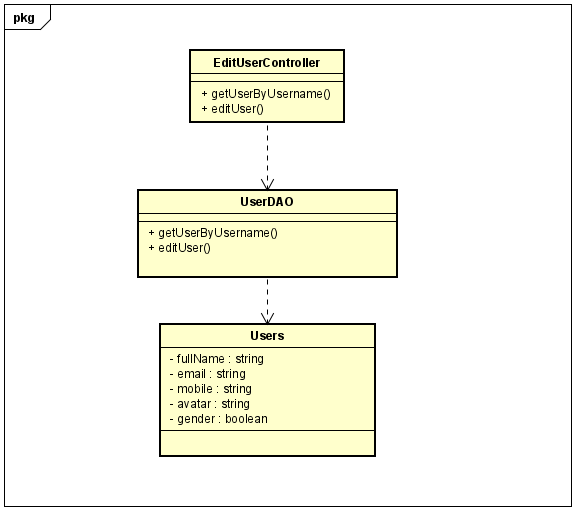
+ " SET \n"

+ " [password] = ?\n"

+ " WHERE userId = ?\n

## 14. Common/User Profile

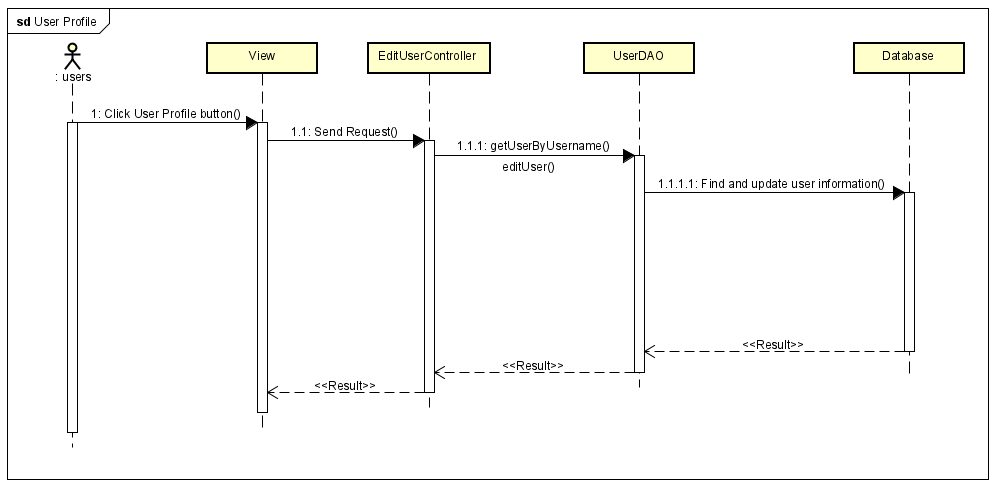
### a. Class Diagram



### 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**

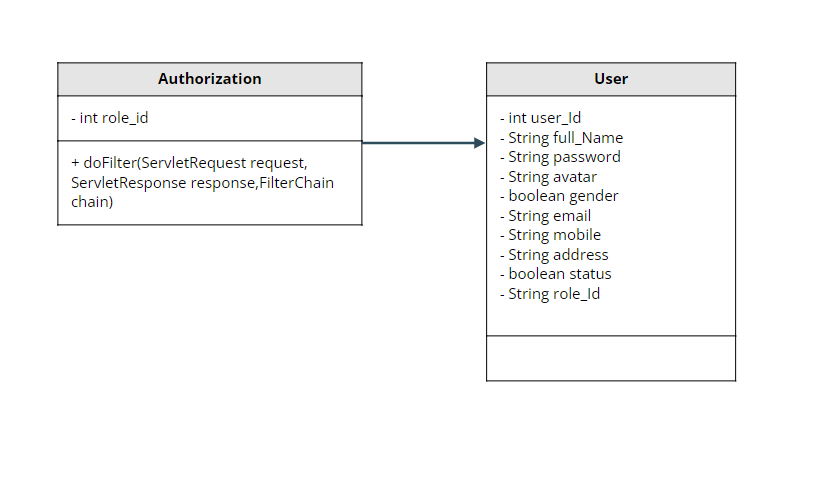
* **UPDATE users**

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

**WHERE username = ?**

## 15. Common/User Authorization

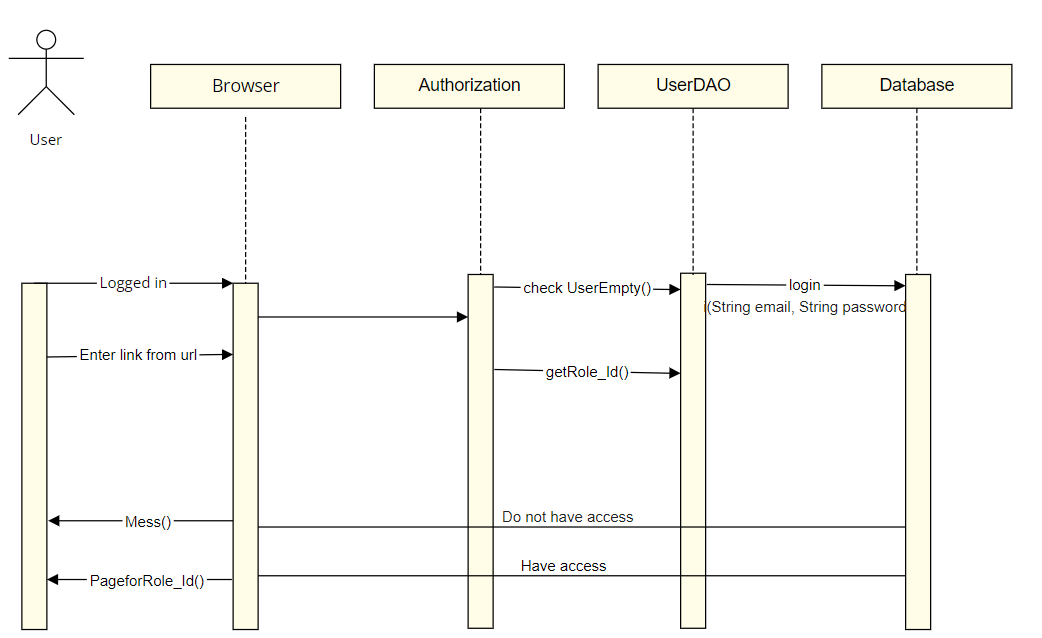
### a. Class Diagram



### b. Class Specifications

| **No** | **Method** | **Description** |
| --- | --- | --- |
| *01* | *getRole\_Id()* | *Get Role\_Id of User* |

### c. Sequence Diagram(s)



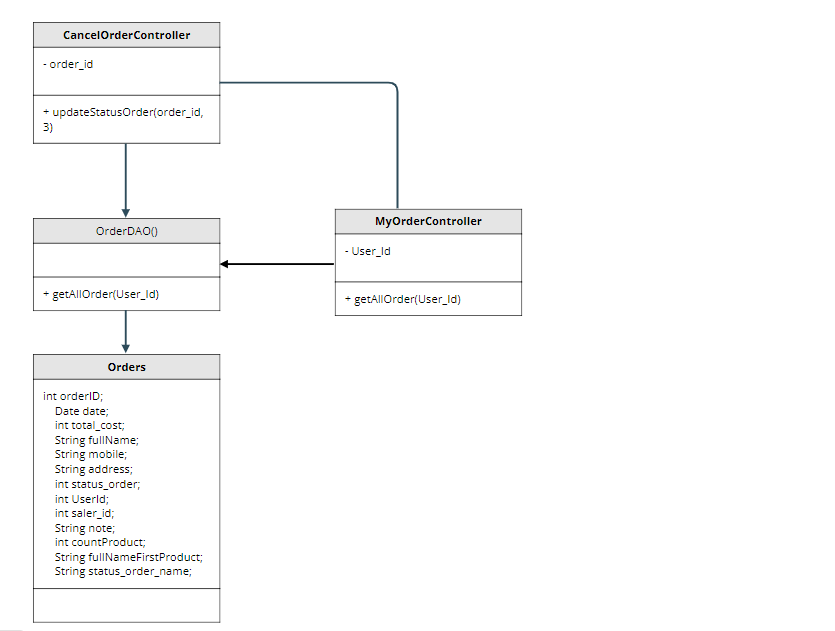
### d. Database queries

-select \* from dbo.[User]\n"

+ "where email = ? and [password] = ?

## 16. Customer/My Orders

### a. Class Diagram

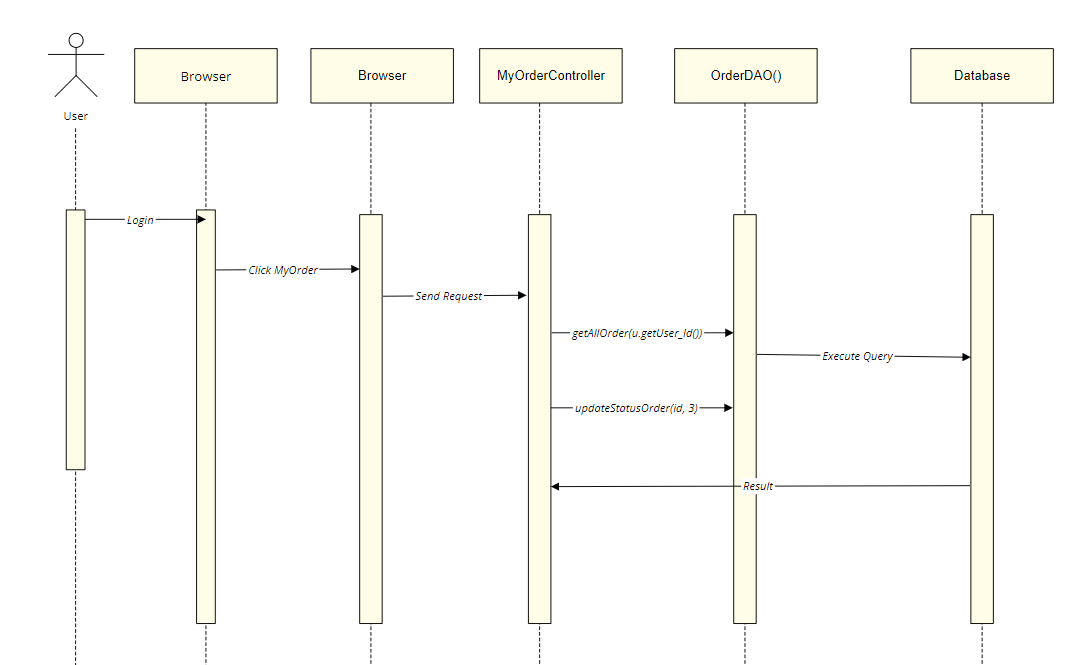


### b. Class Specifications

#### XYZ Class

| **No** | **Method** | **Description** |
| --- | --- | --- |
| *01* | *getAllOrder(User\_Id)* | *get customer orders based on id* |
| *02* | *updateStatusOrder(order\_id, 3)* | *Update order status to cancel* |

### c. Sequence Diagram(s)



### d. Database queries

UPDATE [dbo].[Order]\n"

+ " SET [status\_order] = ?\n"

+ " WHERE order\_id = ?

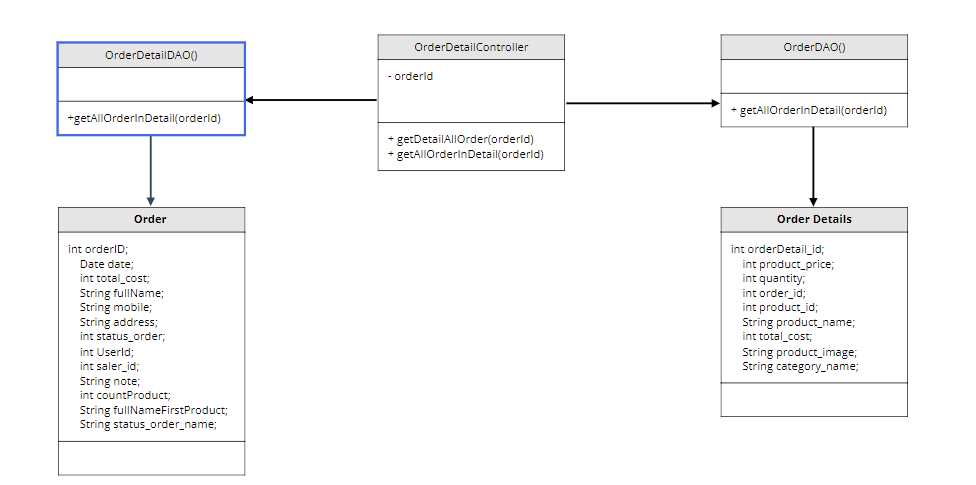
select \* from [Order] join Status\_Order\n"

+ "on [Order].status\_order= Status\_Order.status\_order\_id\n"

+ "where userId = ?

## 17. Customer/Order Information

### a. Class Diagram

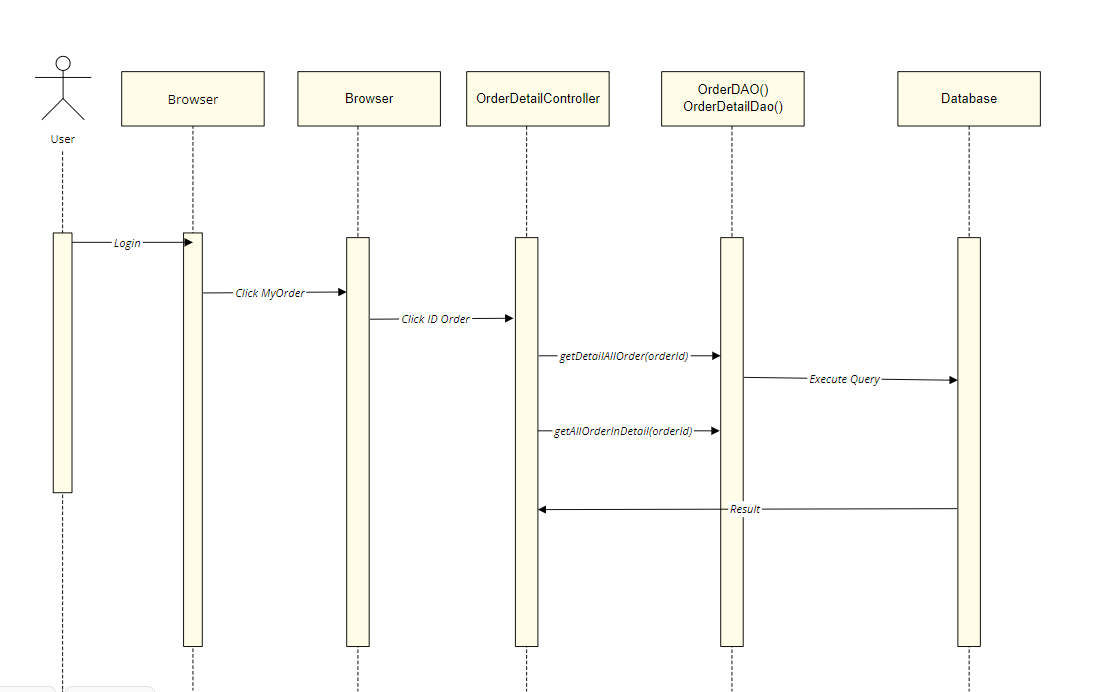


### b. Class Specifications

#### XYZ Class

| **No** | **Method** | **Description** |
| --- | --- | --- |
| *01* | *getDetailAllOrder(orderId)* | *Get information about purchased products* |
| *02* | *getAllOrderInDetail(orderId)* | *get customer information of the order* |

### c. Sequence Diagram(s)



### d. Database queries

select \* from [Order] join Status\_Order\n"

+ " on [Order].status\_order= Status\_Order.status\_order\_id\n"

+ " where order\_id = ?

select \* from Order\_Detail join Product \n"

+ " on Order\_Detail.product\_id = Product.product\_id\n"

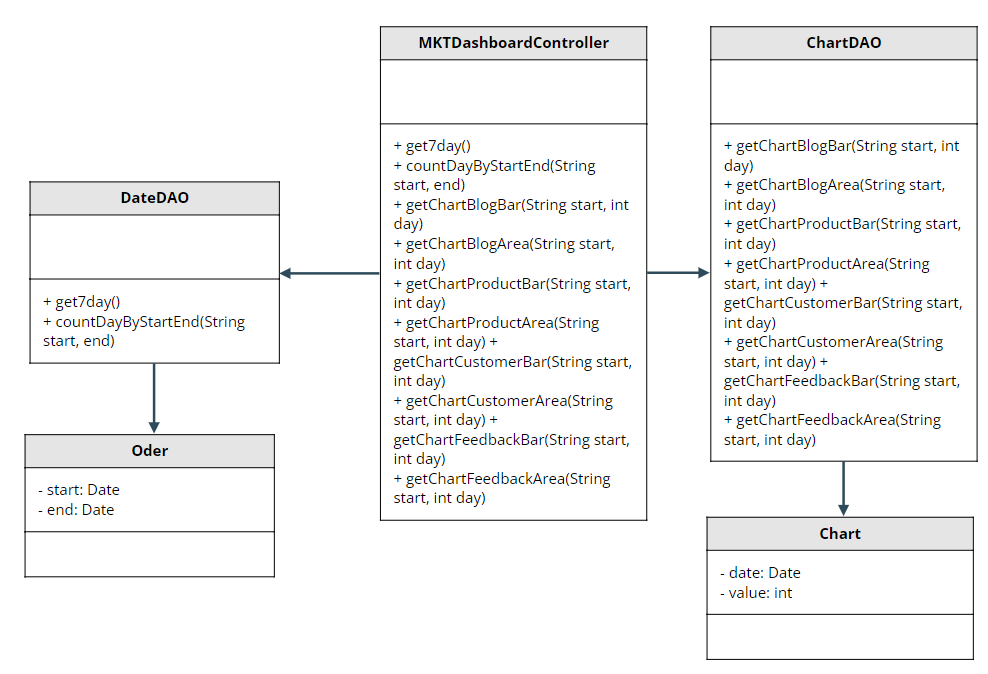
+ " join Products\_Images on Order\_Detail.product\_id = Products\_Images.product\_id\n"

+ " join Category on Product.category\_id = Category.category\_id\n"

+ " where order\_id = ?

## 18. Marketing/MKT Dashboard

### a. Class Diagram

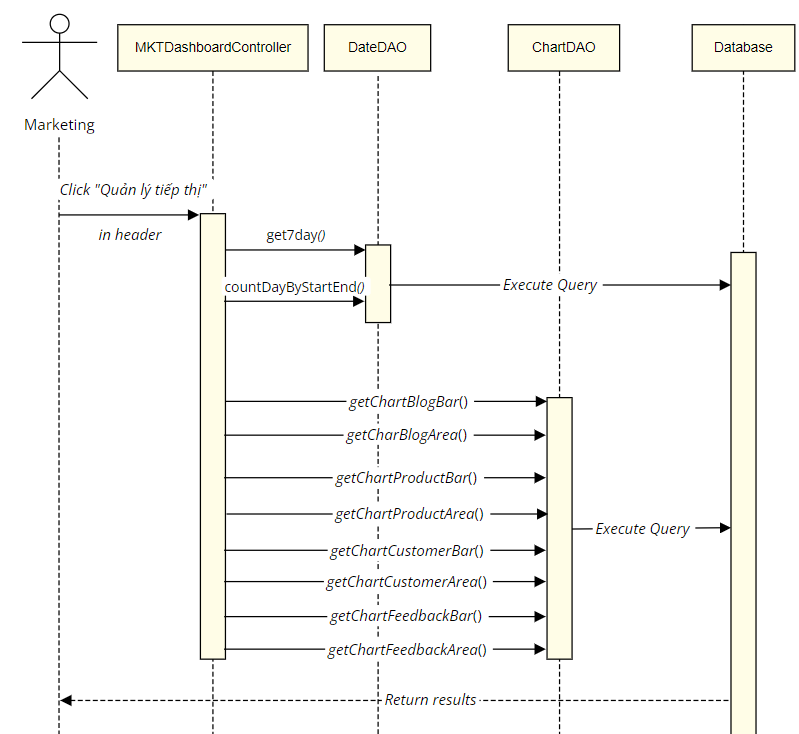


### b. Class Specifications

#### XYZ Class

| **No** | **Method** | **Description** |
| --- | --- | --- |
| *01* | *get7day()* | *Get last seven days* |
| *02* | *countDayByStartEnd()* | *Count the number of days difference between start and end* |
| *03* | *getChartBlogBar()* | *Get statistics by bar chart from start date to end date of post* |
| *04* | *getChartBlogArea()* | *Get statistics by area chart from start date to end date of post* |
| *05* | *getChartProductBar()* | *Get statistics by bar chart from start date to end date of product* |
| *06* | *getChartProductArea()* | *Get statistics by area chart from start date to end date of product* |
| *07* | *getChartCustomerBar()* | *Get statistics by bar chart from start date to end date of customer* |
| *08* | *getChartCustomerArea()* | *Get statistics by area chart from start date to end date of customer* |
| *09* | *getChartFeedbackBar()* | *Get statistics by bar chart from start date to end date of feedback* |
| *10* | *getChartFeedbackArea()* | *Get statistics by area chart from start date to end date of feedback* |

### c. Sequence Diagram(s)

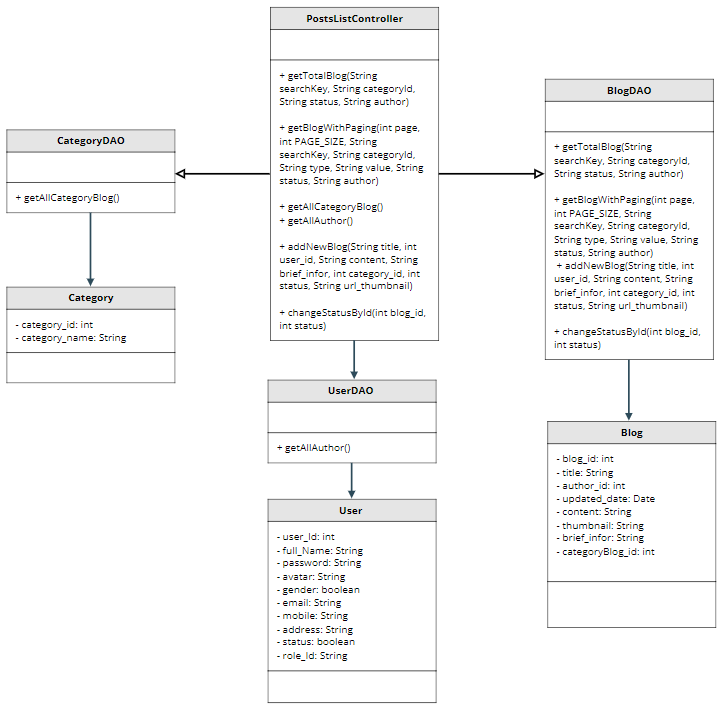
****

### d. Database queries

* select GETDATE(), GETDATE()-6
* SELECT DATEDIFF(day, ?, ?)
* select DATEADD(DAY, ?, ?)
* select count(\*) from Blog where updated\_date = DATEADD(DAY, ?, ?) and status = 1
* select count(\*) from Blog where updated\_date <= DATEADD(DAY, ?, ?) and status = 1
* select count(\*) from Product where update\_date = DATEADD(DAY, ?, ?) and status = 1
* select count(\*) from Product where update\_date <= DATEADD(DAY, ?, ?) and status = 1
* select count(\*) from Customer where updated\_date = DATEADD(DAY, ?, ?) and status = 1
* select count(\*) from Customer where updated\_date <= DATEADD(DAY, ?, ?) and status = 1
* select count(\*) from Feedback where date = DATEADD(DAY, ?, ?) and status = 1
* select count(\*) from Feedback where date <= DATEADD(DAY, ?, ?) and status = 1

## 19. Marketing/Posts List

### a. Class Diagram

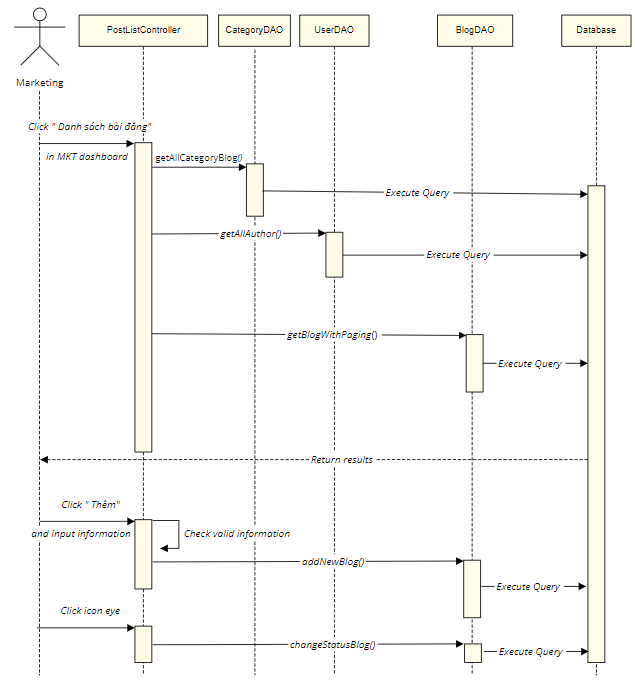


### b. Class Specifications

#### XYZ Class

| **No** | **Method** | **Description** |
| --- | --- | --- |
| *01* | *getTotalBlog()* | *Get count total blog* |
| *02* | *getBlogWithPaging()* | *Get all post list is paginated* |
| *03* | *getAllCategoryBlog()* | *Get all category of blog* |
| *04* | *getAllAuthor()* | *Get all author of blog* |
| *05* | *addNewBlog()* | *Add new blog* |
| *06* | *changeStatusById()* | *Change status of specific blog* |

### c. Sequence Diagram(s)

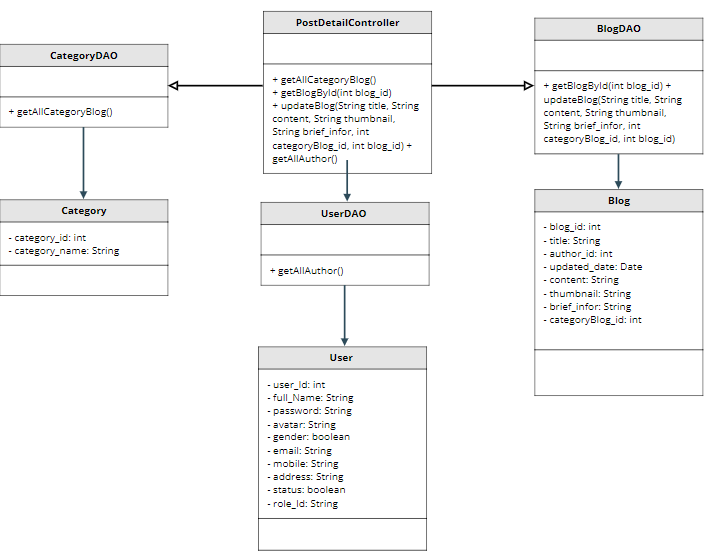
****

### d. Database queries

* Select count(\*) from Blog where categoryBlog\_id " + categoryId + " and [status] " + status + " and author\_id " + author + " and title like N'%" + searchKey + "%'
* select \* from Blog where categoryBlog\_id " + categoryId + " and [status] " + status + " and author\_id " + authorId + " and [title] like N'%" + searchKey + "%' order by " + value + " " + type + " offset (?-1)\*? row fetch next ? row only
* Select \* from Category\_Blog
* select \* from [User] where role\_id = 2
* INSERT INTO [dbo].[Blog] ( [title], [author\_id], [content], [thumbnail], [brief\_infor], [categoryBlog\_id], [status]) VALUES (?,?,?,?,?,?,?)
* UPDATE [dbo].[Blog] SET [status] = ? WHERE blog\_id = ?

## 20. Marketing/Post Details

### a. Class Diagram

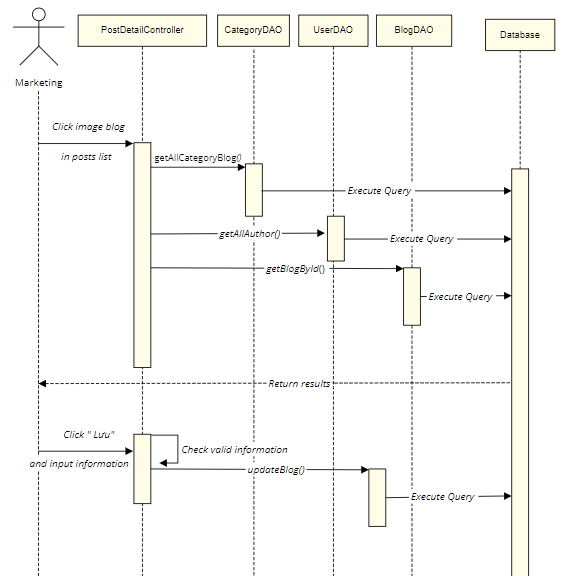


### b. Class Specifications

#### XYZ Class

| **No** | **Method** | **Description** |
| --- | --- | --- |
| *01* | *getAllCategoryBlog()* | *Get all category of blog* |
| *02* | *getBlogById()* | *Get all information of specific blog* |
| *03* | *updateBlog()* | *Update information of specific blog* |
| *04* | *getAllAuthor* | *Get all author of blog* |

### c. Sequence Diagram(s)

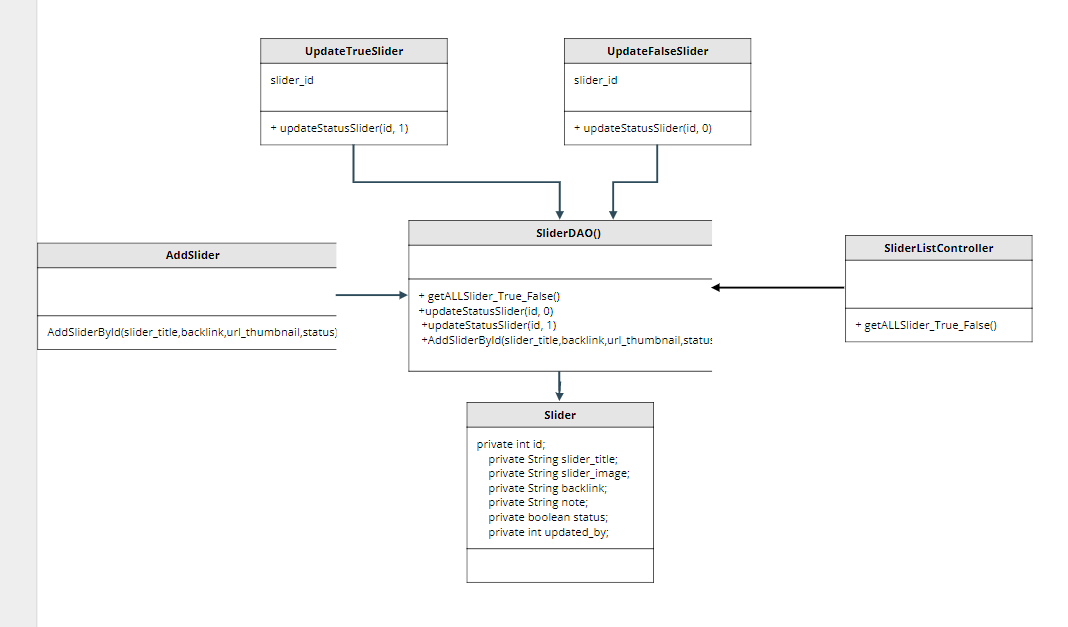


### d. Database queries

* Select \* from Blog where blog\_id = ?
* Select \* from Category\_Blog
* select \* from [User] where role\_id = 2
* UPDATE [dbo].[Blog] SET [title] = ? ,[author\_id] = ? ,[updated\_date] = getdate() ,[content] = ? ,[thumbnail] = ? ,[brief\_infor] = ? ,[categoryBlog\_id] = ? ,[status] = ? WHERE blog\_id = ?

## 21. Marketing/Sliders List

### a. Class Diagram

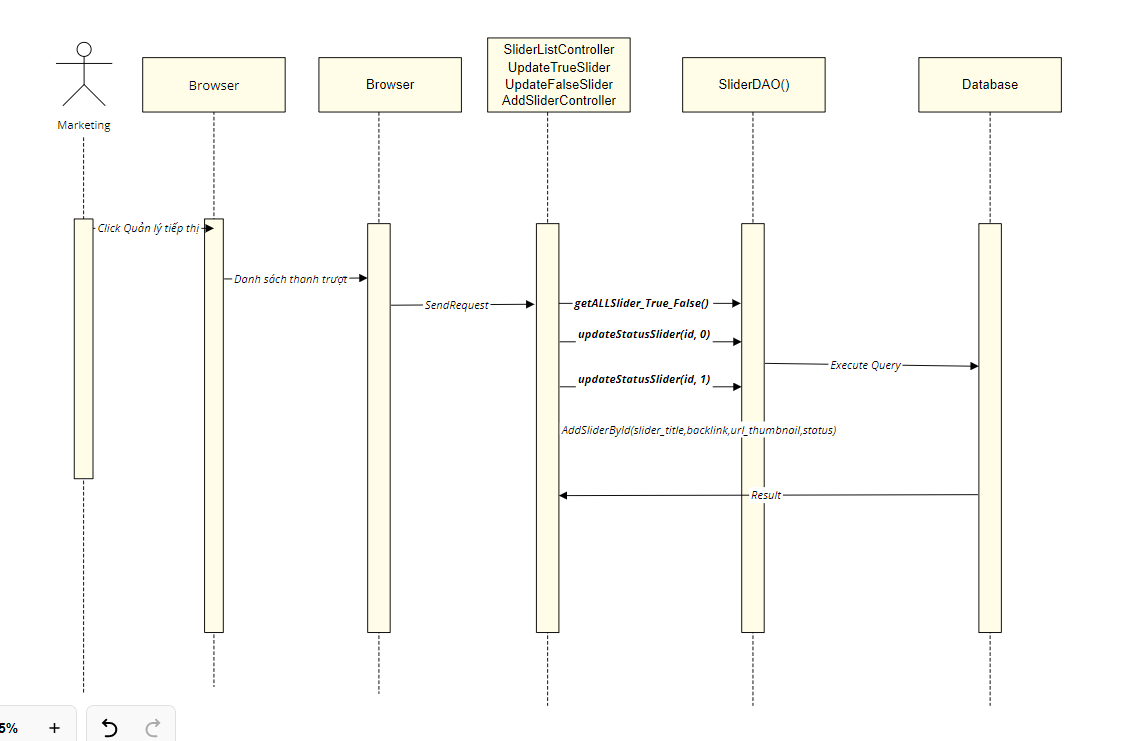


### b. Class Specifications

#### XYZ Class

| **No** | **Method** | **Description** |
| --- | --- | --- |
| *01* | *getALLSlider\_True\_False()* | *get all sliders (including status true and false)* |
| *02* | *updateStatusSlider(id, 0)* | *update slider status from true to false* |
| *03* | *updateStatusSlider(id, 1)* | *update slider status from false to true* |
| *04* | *AddSliderById(slider\_title,backlink,url\_thumbnail,status)* | *Add 1 new slider (include title, backlink, thumbnail and status)* |

### c. Sequence Diagram(s)



### d. Database queries

INSERT INTO [dbo].[Slider]\n"

+ " (slider\_title\n"

+ " ,slider\_image\n"

+ " ,backlink\n"

+ " ,[status])\n"

+ " VALUES\n"

+ " (?,?,?,?)

UPDATE [dbo].[Slider]\n"

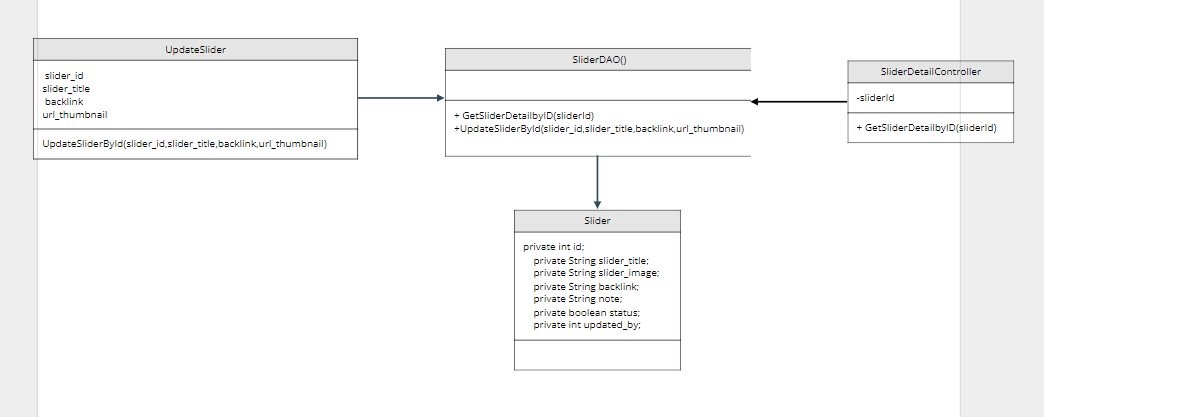
+ " SET status = ?\n"

+ " WHERE slider\_id = ?

SELECT \* FROM [dbo].[Slider]

## 22. Marketing/Slider Details

### a. Class Diagram

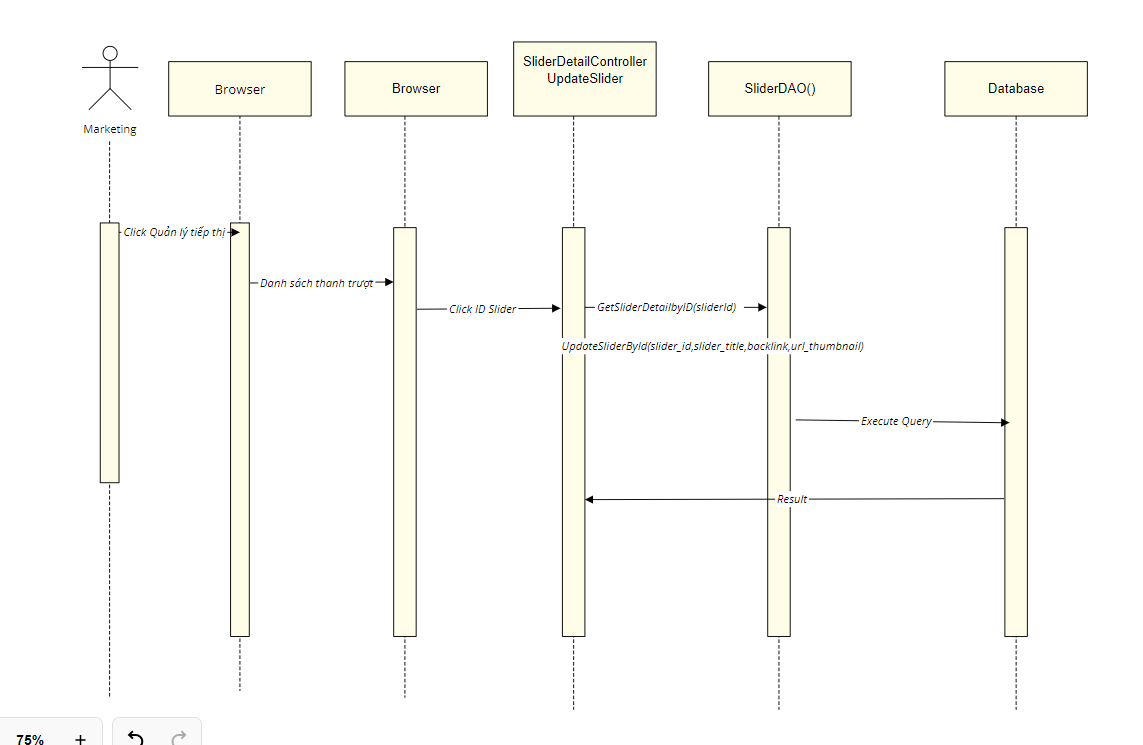


### b. Class Specifications

#### XYZ Class

| **No** | **Method** | **Description** |
| --- | --- | --- |
| *01* | *GetSliderDetailbyID(sliderId)* | *Get slider's information by sliderId* |
| *02* | *UpdateSliderById(slider\_id,slider\_title,backlink,url\_thumbnail)* | *update information of slider* |

### c. Sequence Diagram(s)



### d. Database queries

UPDATE [dbo].[Slider]\n"

+ " SET [slider\_title] = ?\n"

+ " ,[slider\_image] = ?\n"

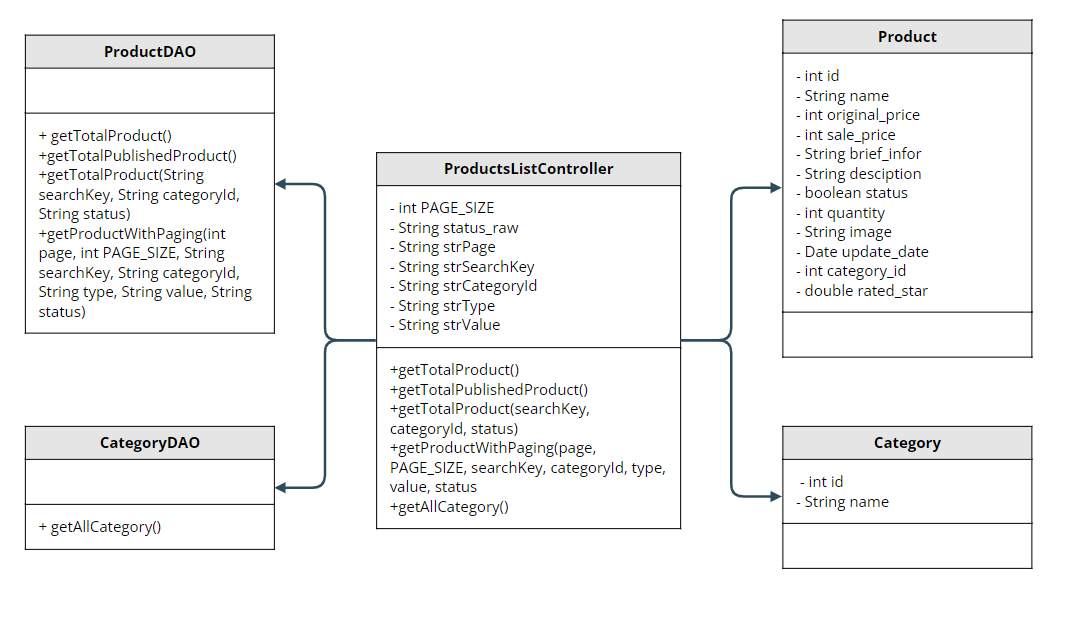
+ " ,[backlink] = ?\n"

+ " WHERE slider\_id = ?

SELECT \* FROM [dbo].[Slider] where slider\_id = ?

## 23. Marketing/Products List

### a. Class Diagram

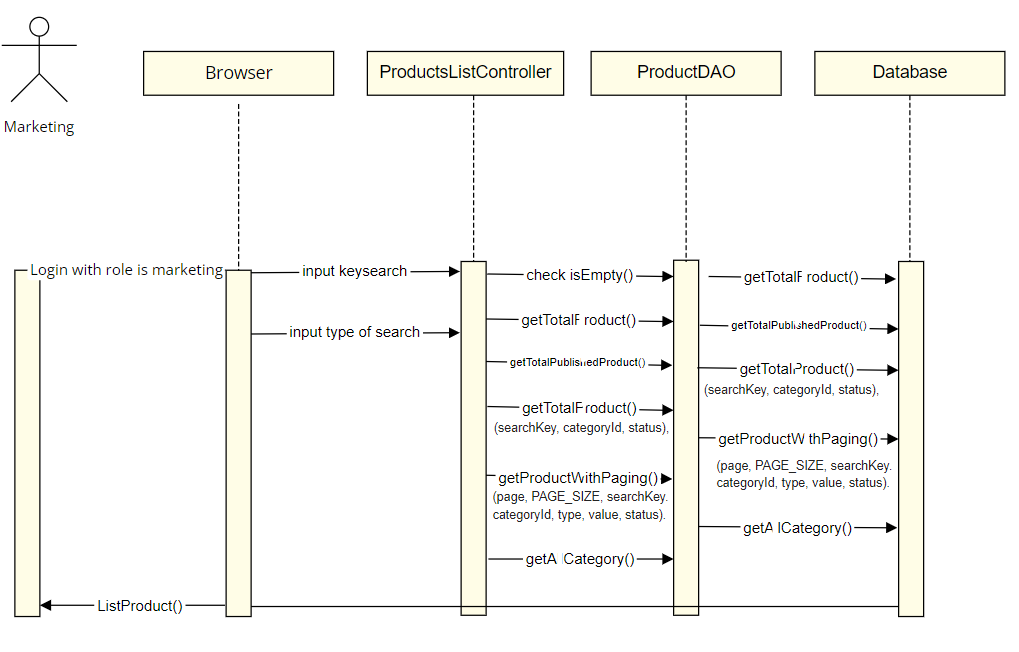


### b. Class Specifications

#### 

| **No** | **Method** | **Description** |
| --- | --- | --- |
| *01* | *getTotalProduct()* | *Get number of all product* |
| *02* | *getTotalPublishedProduct()* | *Get number of all publushedProduct* |
| *03* | *getTotalProduct(String searchKey, String categoryId, String status)* | *Get number of all product by searchKey,categoryId,status* |
| *04* | *getProductWithPaging(int page, int PAGE\_SIZE, String searchKey, String categoryId, String type, String value, String status)* | *Get list product by type of search and paging* |
| *05* | *getAllCategory()* | *Get all category for search by category* |

### c. Sequence Diagram(s)



### d. Database queries

-Select \* from Category

-select COUNT(product\_id) from Product

-select COUNT(product\_id) from Product where status = 1

-Select count(product\_id) from Product "

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

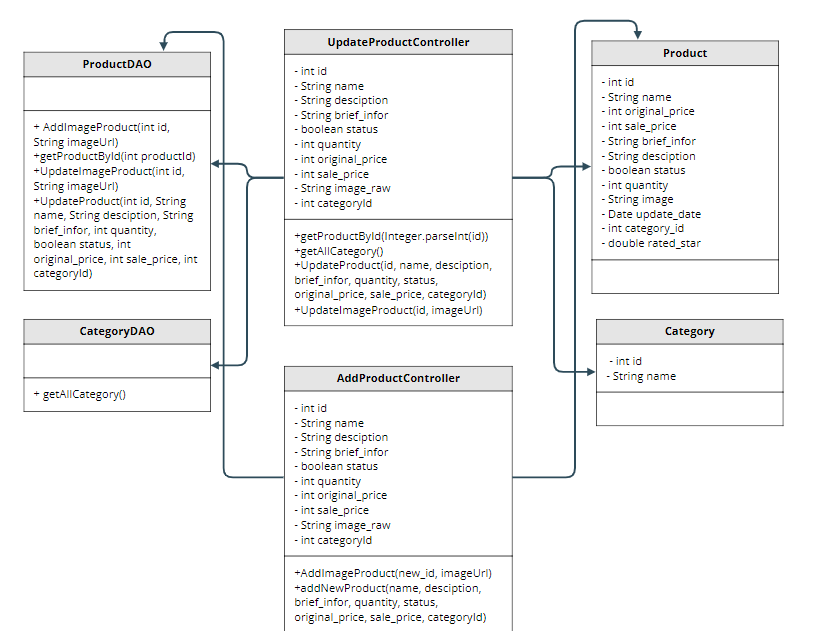
-select \* from Product\n"

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

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

## 24. Marketing/Product Details

### a. Class Diagram

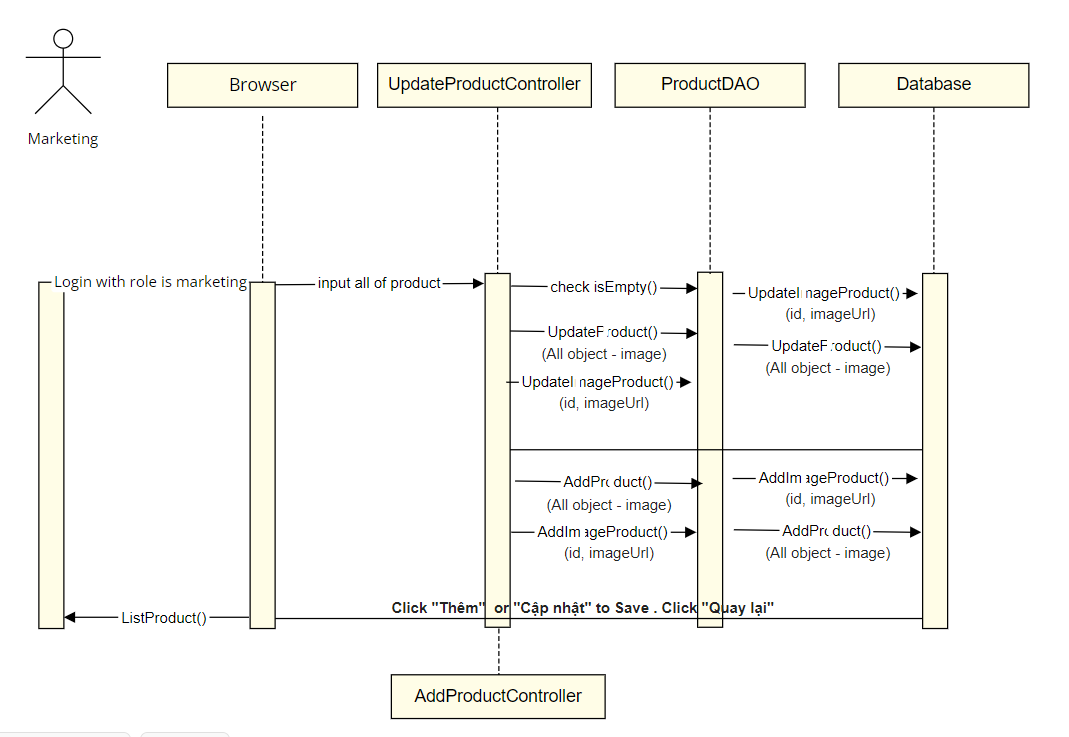


### b. Class Specifications

#### XYZ Class

| **No** | **Method** | **Description** |
| --- | --- | --- |
| *01* | *getProductById(int productId)* | *Get product by id* |
| *02* | *UpdateProduct(int id, String name, String desciption, String brief\_infor, int quantity, boolean status, int original\_price, int sale\_price, int categoryId)* | *Update product with all (- image) by id* |
| *03* | *UpdateImageProduct(int id, String imageUrl)* | *Update image product by id* |
| *04* | *addNewProduct(String name, String desciption, String brief\_infor, int quantity, boolean status, int original\_price, int sale\_price, int categoryId)* | *Add new product will all (- image)* |
| *05* | *AddImageProduct(int id, String imageUrl)* | *Add image product by id* |
| *06* | *getAllCategory()* | *Get all category* |

### c. Sequence Diagram(s)



### d. Database queries

-INSERT INTO [dbo].[Product]\n"

+ " ([product\_name]\n"

+ " ,[original\_prices]\n"

+ " ,[sale\_prices]\n"

+ " ,[product\_details]\n"

+ " ,[brief\_infor]\n"

+ " ,[status]\n"

+ " ,[quantity]\n"

+ " ,[category\_id])\n"

+ " VALUES\n"

+ " (?,?,?,?,?,?,?,?)

-INSERT INTO [dbo].[Products\_Images]\n"

+ " ([product\_id]\n"

+ " ,[images])\n"

+ " VALUES\n"

+ " (?,?)

-select \* from Product where product\_id = ?

-Select \* from Category

-UPDATE [dbo].[Product]\n"

+ " SET [product\_name] = ?\n"

+ " ,[original\_prices] = ?\n"

+ " ,[sale\_prices] = ?\n"

+ " ,[product\_details] = ?\n"

+ " ,[brief\_infor] = ?\n"

+ " ,[status] = ?\n"

+ " ,[quantity] = ?\n"

+ " ,[category\_id] = ?\n"

+ "\n"

+ " WHERE product\_id = ?

-UPDATE [dbo].[Products\_Images]\n"

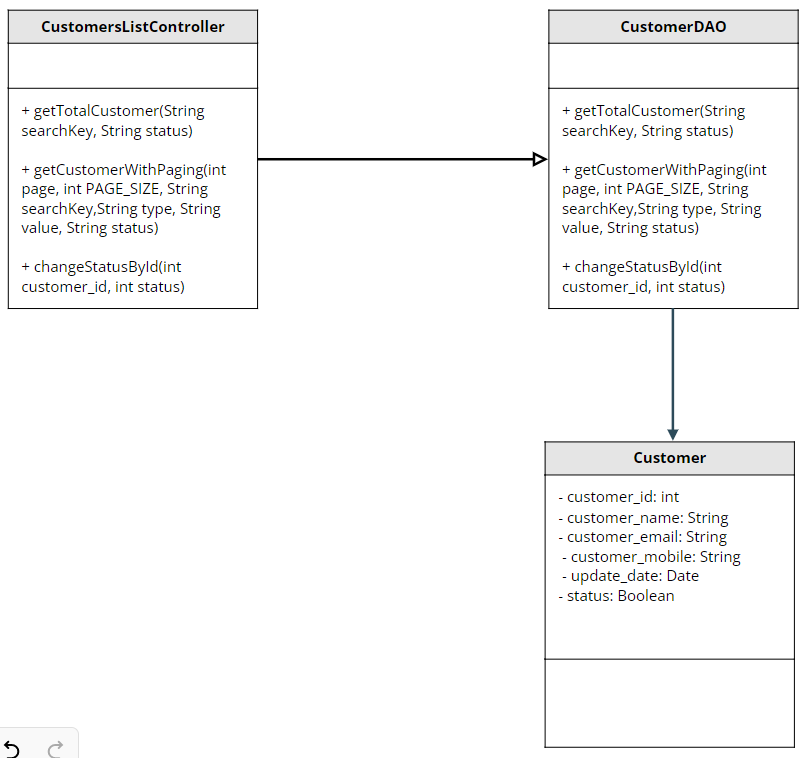
+ " SET\n"

+ " [images] = ?\n"

+ " WHERE product\_id = ?

## 25. Marketing/Customers List

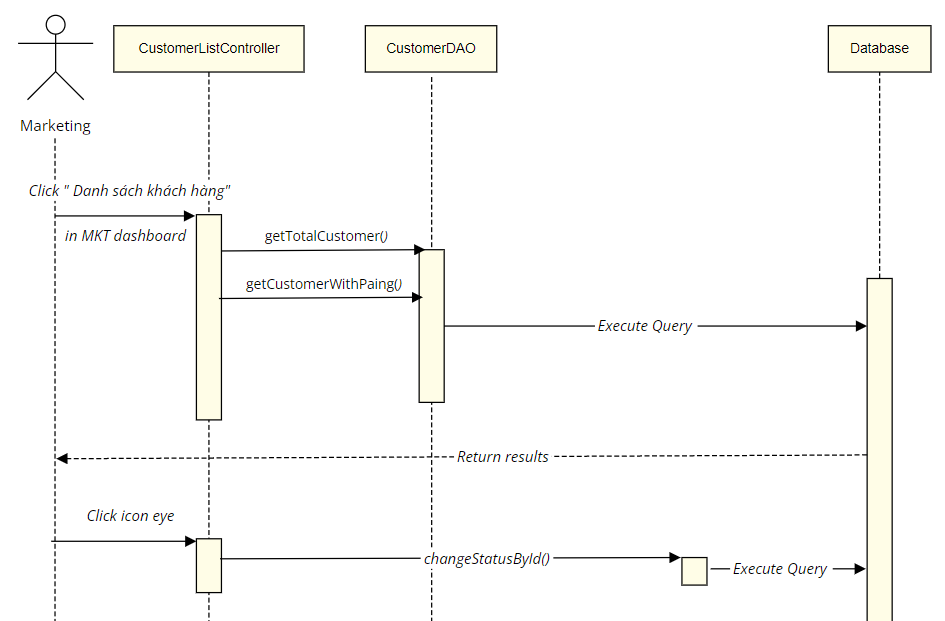
### a. Class Diagram

****

### b. Class Specifications

| **No** | **Method** | **Description** |
| --- | --- | --- |
| *01* | *getTotalCustomer()* | *Get total customer in the database* |
| *02* | *getCustomerWithPaing()* | *Get list all customer in the database pagined* |
| *03* | *changeStatusById()* | *Change status of specific customer* |

### c. Sequence Diagram(s)

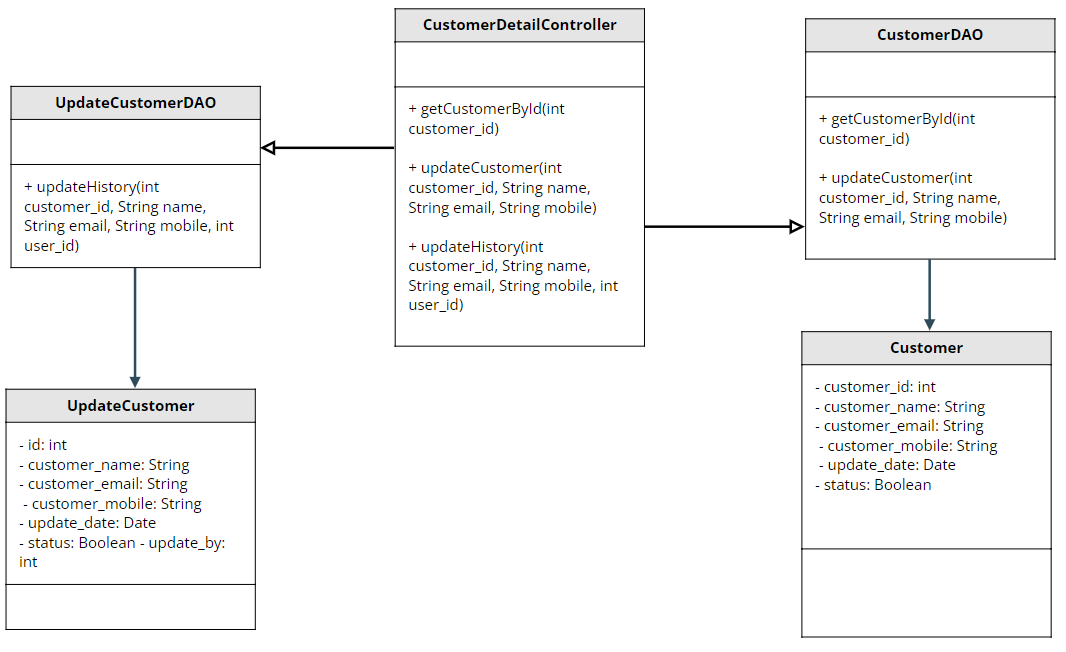


### d. Database queries

* Select count(\*) from Customer where [status] " + status + " and customer\_name like N'%" + searchKey + "%'
* select \* from Customer where [status] " + status + " and [customer\_name] like N'%" + searchKey + "%' order by " + value + " " + type + " offset (?-1)\*? row fetch next ? row only
* UPDATE [dbo].[Customer] SET [status] = ? WHERE customer\_id = ?

## 26. Marketing/Customer Details

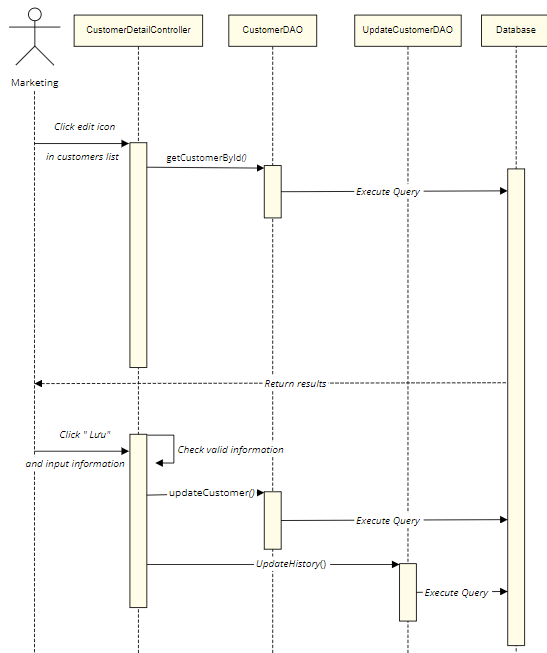
### a. Class Diagram



### b. Class Specifications

| **No** | **Method** | **Description** |
| --- | --- | --- |
| *01* | *getCustomerById()* | *Get customer by id* |
| *02* | *updateCustomer()* | *Update customer in the database* |
| *03* | *UpdateHistory()* | *Update history for each time update customer* |

### c. Sequence Diagram(s)

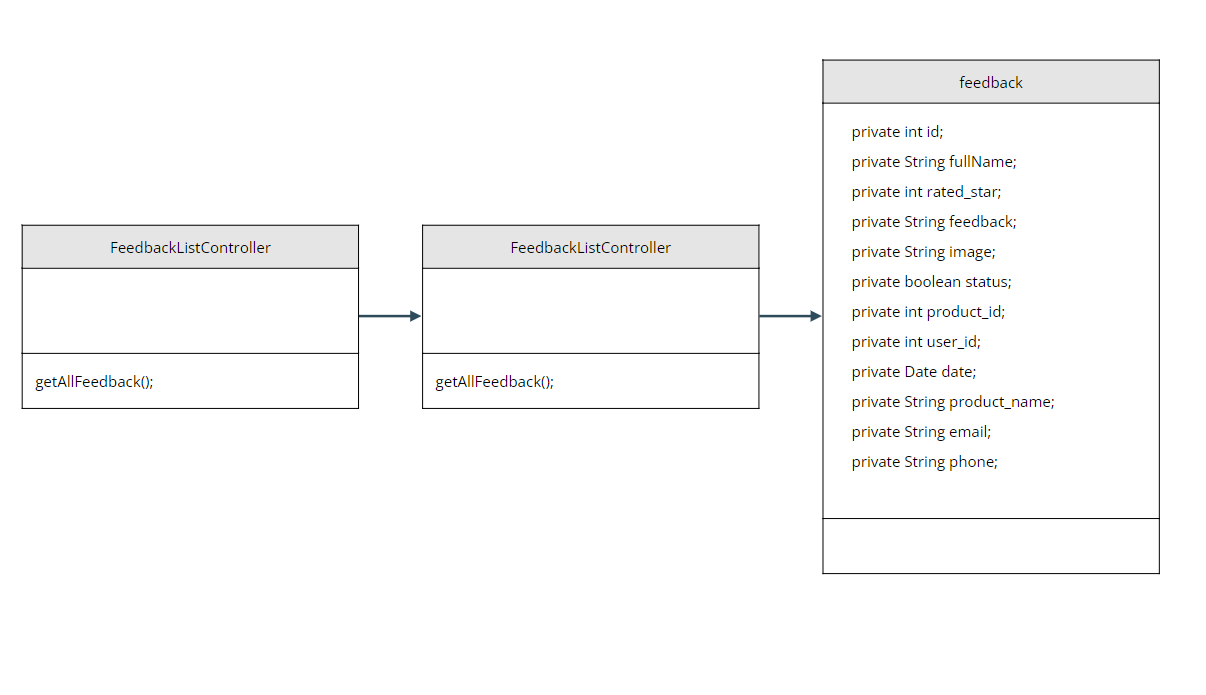


### d. Database queries

* select \* from [Customer] where customer\_id = ?
* update dbo.[Customer] set [customer\_name] = ?, customer\_email = ?, customer\_mobile = ?, updated\_date = getdate() where customer\_id = ?
* INSERT INTO [dbo].[Update\_Customer] ([customer\_id] ,[email],[fullName] ,[mobile] ,[update\_by]) VALUES (?,?,?,?,?)

## 27. Marketing/Feedbacks List

### a. Class Diagram

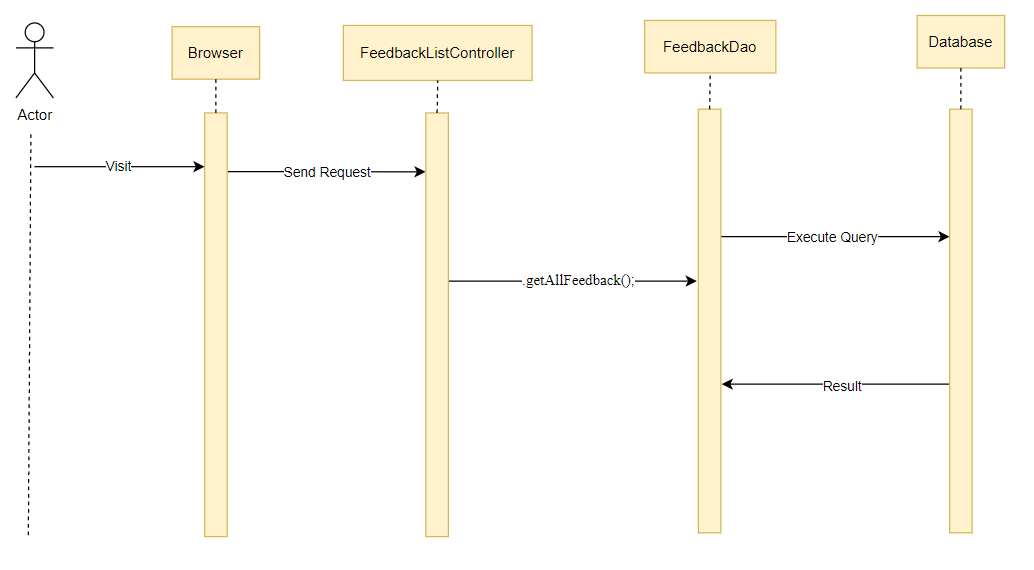


### b. Class Specifications

#### XYZ Class

| **No** | **Method** | **Description** |
| --- | --- | --- |
| *01* | *getAllFeedback();* | *get all feedback in table feedback database* |

### c. Sequence Diagram(s)



### d. Database queries

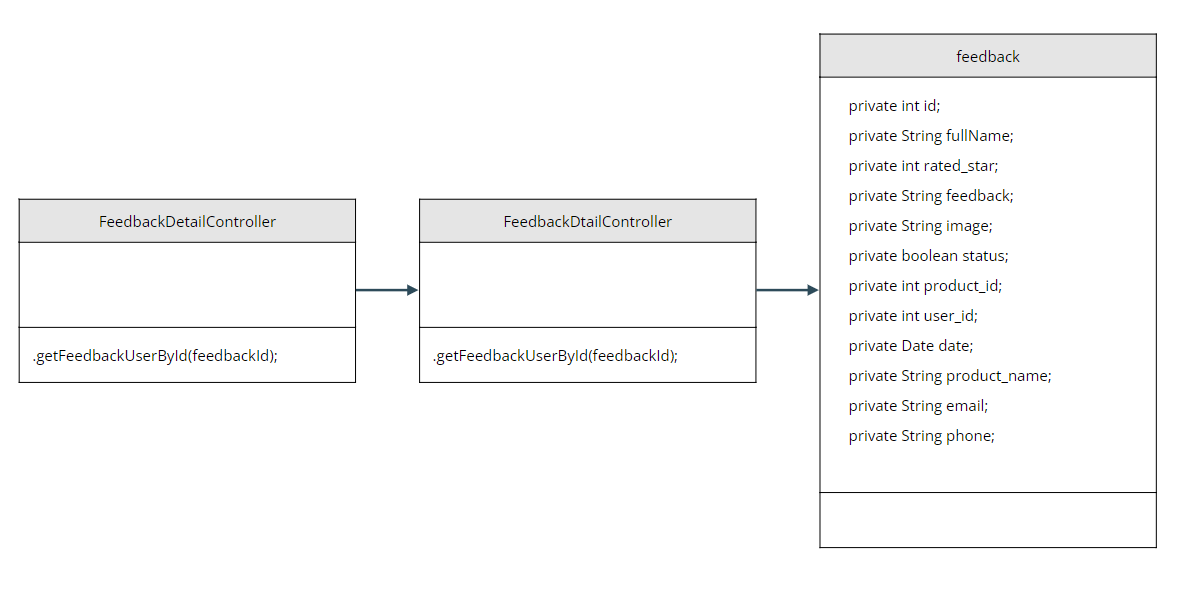
select f.\*, p.product\_name \n"

+ "from Feedback f join Product p \n"

+ "on f.product\_id = p.product\_id";

## 28. Marketing/Feedback Details

### a. Class Diagram

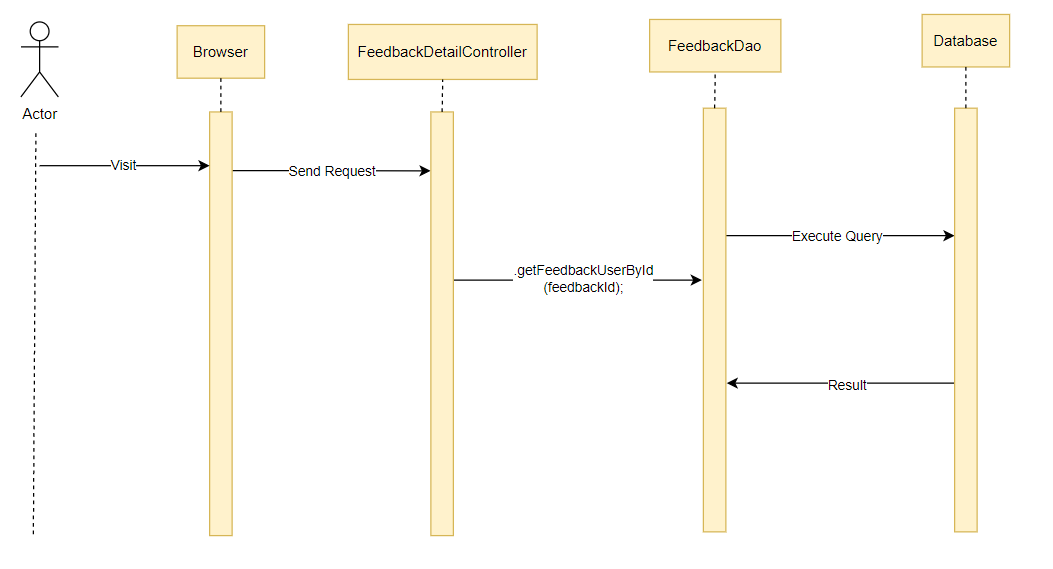


### b. Class Specifications

#### XYZ Class

| **No** | **Method** | **Description** |
| --- | --- | --- |
| *01* | *.getFeedbackUserById(feedbackId);* | *get feedback user by id\_feedback in table Feedback* |

### c. Sequence Diagram(s)



### d. Database queries

select f.\*, p.product\_name, u.email , u.mobile\n"

+ "from Feedback f \n"

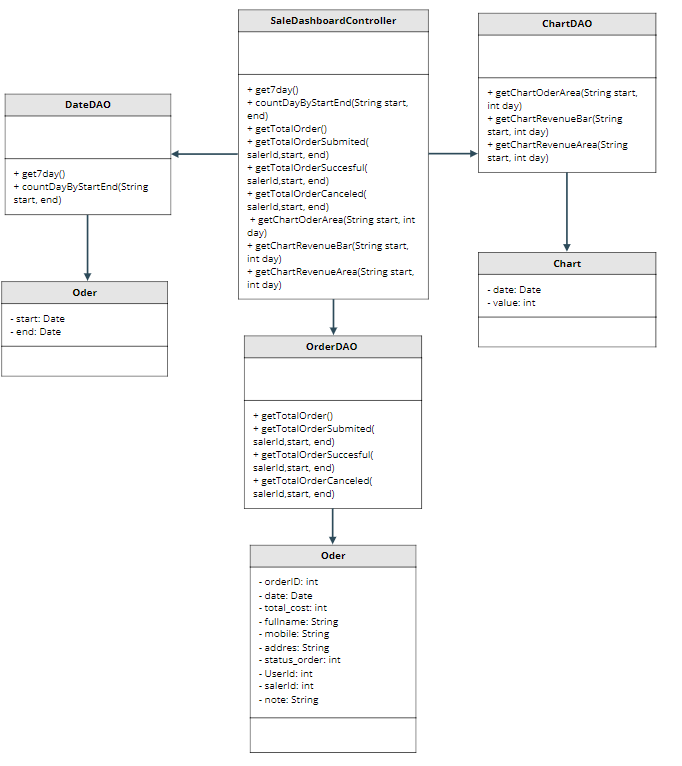
+ "join Product p on f.product\_id = p.product\_id\n"

+ "join [User] u on f.userId = u.userId\n"

+ "where f.feedBack\_id = ?"

## 29. Sale/Sale Dashboard

### a. Class Diagram

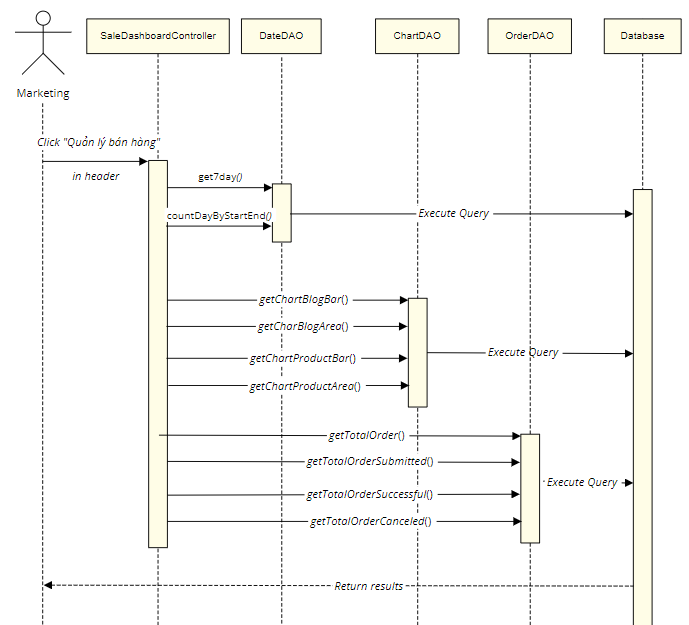


### b. Class Specifications

#### XYZ Class

| **No** | **Method** | **Description** |
| --- | --- | --- |
| *01* | *get7day()* | *Get last seven days* |
| *02* | *countDayByStartEnd()* | *Count the number of days difference between start and end* |
| *03* | *getTotalOrder()* | *Get total order* |
| *04* | *getTotalOrderSubmitted()* | *Get total order have status is submitted* |
| *05* | *getTotalOrderSuccessful()* | *Get total order have status is successful* |
| *06* | *getTotalOrderCanceled()* | *Get total order have status is canceled* |
| *07* | *getChartOrderBar()* | *Get statistics by bar chart from start date to end date of order* |
| *08* | *getChartOrderrArea()* | *Get statistics by area chart from start date to end date of order* |
| *09* | *getChartRevenueBar()* | *Get statistics by bar chart from start date to end date of revenue* |
| *10* | *getChartRevenuekArea()* | *Get statistics by area chart from start date to end date of revenue* |

### c. Sequence Diagram(s)

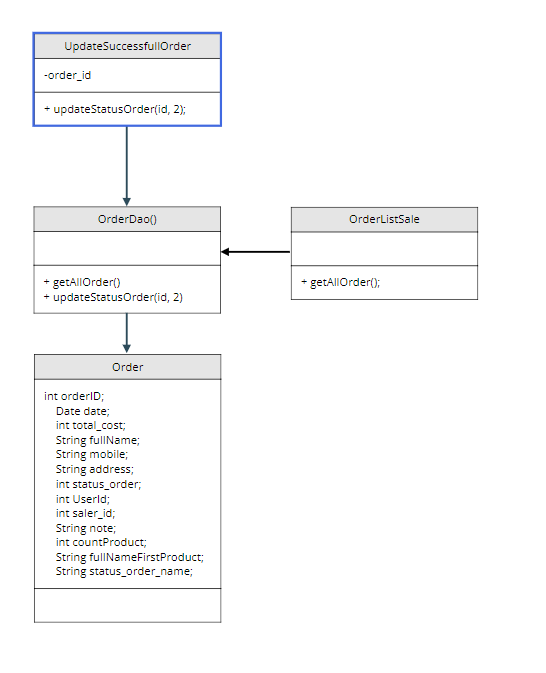
****

### d. Database queries

* select GETDATE(), GETDATE()-6
* SELECT DATEDIFF(day, ?, ?)
* select DATEADD(DAY, ?, ?)
* select count(\*) from [Order] where saler\_id " + salerId + " and orderDate <= ? and orderDate >= ?
* select count(\*) from [Order] where saler\_id " + salerId + " and orderDate <= ? and orderDate >= ? and status\_order = 1
* select count(\*) from [Order] where saler\_id " + salerId + " and orderDate <= ? and orderDate >= ? and status\_order = 2
* select count(\*) from [Order] where saler\_id " + salerId + " and orderDate <= ? and orderDate >= ? and status\_order = 3
* select count(\*) from [Order] where saler\_id " + salerId + " and orderDate = DATEADD(DAY, ?, ?)
* select sum(total\_cost) from [Order] where saler\_id " + salerId + " and orderDate = DATEADD(DAY, ?, ?)
* select sum(total\_cost) from [Order] where saler\_id " + salerId + " and orderDate <= DATEADD(DAY, ?, ?) and orderDate >= ?

## 30. Sale/Orders List

### a. Class Diagram

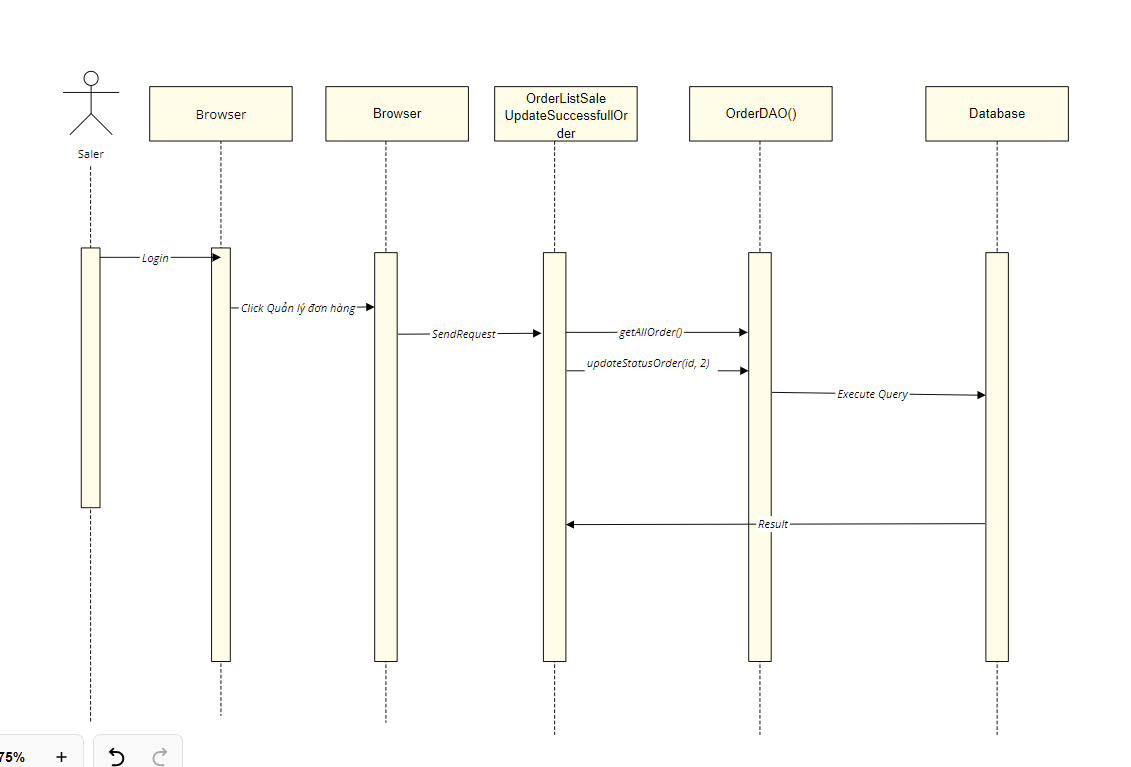


### b. Class Specifications

#### XYZ Class

| **No** | **Method** | **Description** |
| --- | --- | --- |
| *01* | *getAllOrder()* | *Display information of orders (including product, total cost, sale\_id, user\_id)* |
| *02* | *updateStatusOrder(id, 2)* | *Update the successful status of the order* |

### c. Sequence Diagram(s)



### d. Database queries

UPDATE [dbo].[Order]\n"

+ " SET [status\_order] = ?\n"

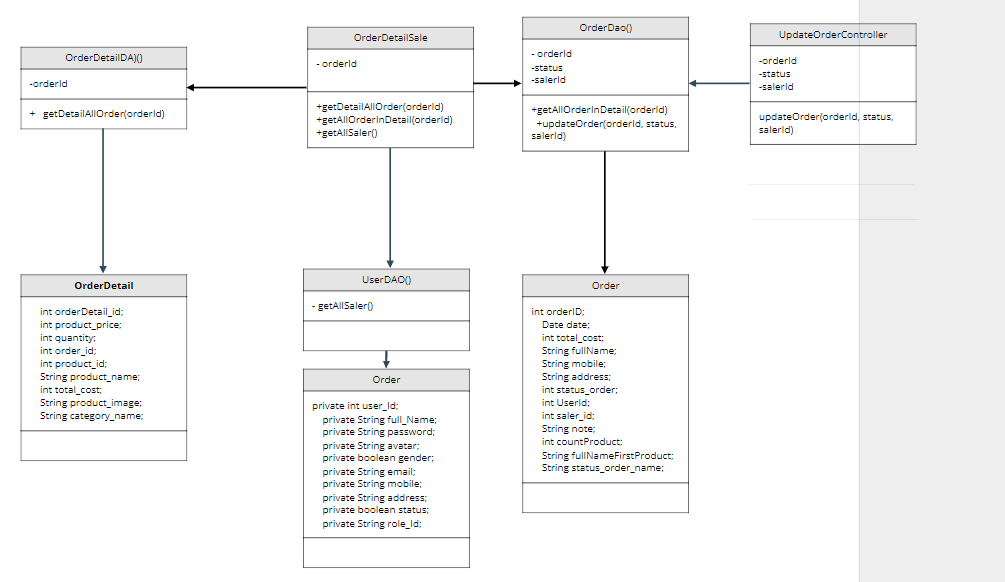
+ " WHERE order\_id = ?

select \* from [Order] join Status\_Order\n"

+ " on [Order].status\_order= Status\_Order.status\_order\_id

## 31. Sale/Order Details

### a. Class Diagram

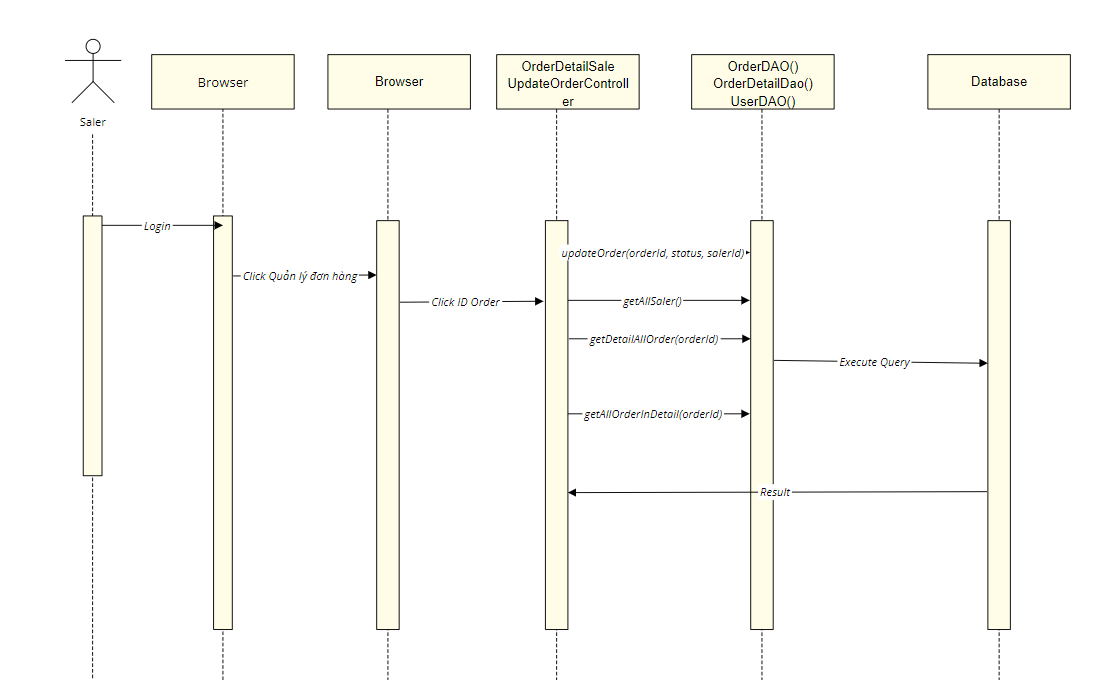


### b. Class Specifications

#### XYZ Class

| **No** | **Method** | **Description** |
| --- | --- | --- |
| *01* | *getDetailAllOrder(orderId)* | *Get information about purchased products* |
| *02* | *getAllOrderInDetail(orderId)* | *get customer information of the order* |
| *03* | *getAllSaler()* | *Get the IDs of all the salers* |
| *04* | *updateOrder(orderId, status, salerId)* | *Update order again (based on orderId, status, salerId)* |

### c. Sequence Diagram(s)



### d. Database queries

UPDATE [dbo].[Order]\n"

+ " SET [status\_order] = ? \n"

+ " ,[saler\_id] = ? \n"

+ " WHERE order\_id = ?

select \* from Order\_Detail join Product \n"

+ " on Order\_Detail.product\_id = Product.product\_id\n"

+ " join Products\_Images on Order\_Detail.product\_id = Products\_Images.product\_id\n"

+ " join Category on Product.category\_id = Category.category\_id\n"

+ " where order\_id = ?

select \* from [Order] join Status\_Order\n"

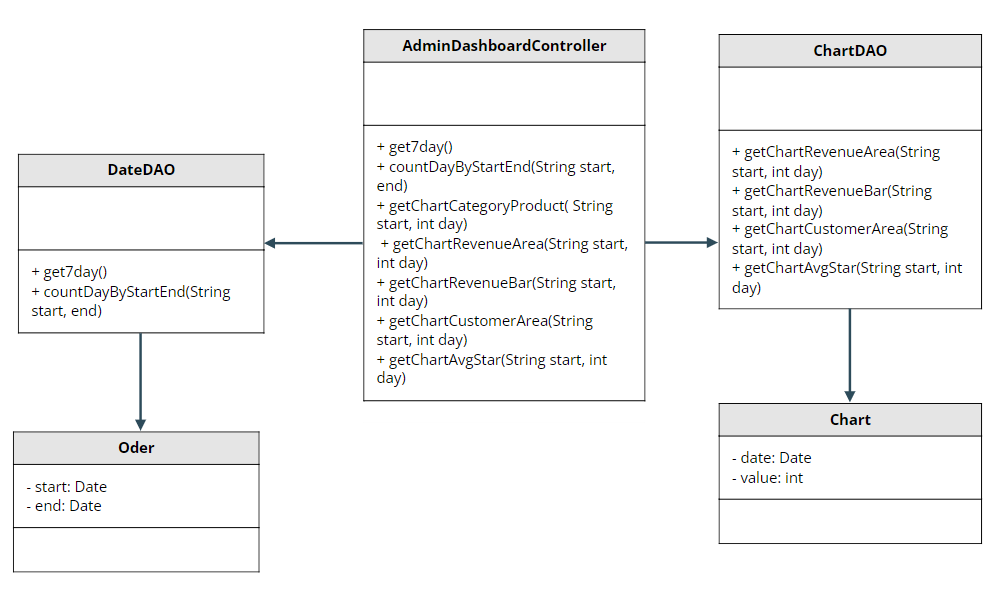
+ " on [Order].status\_order= Status\_Order.status\_order\_id\n"

+ " where order\_id = ?

select \* from [User] where role\_id = 3

## 32. Admin/Admin Dashboard

### a. Class Diagram

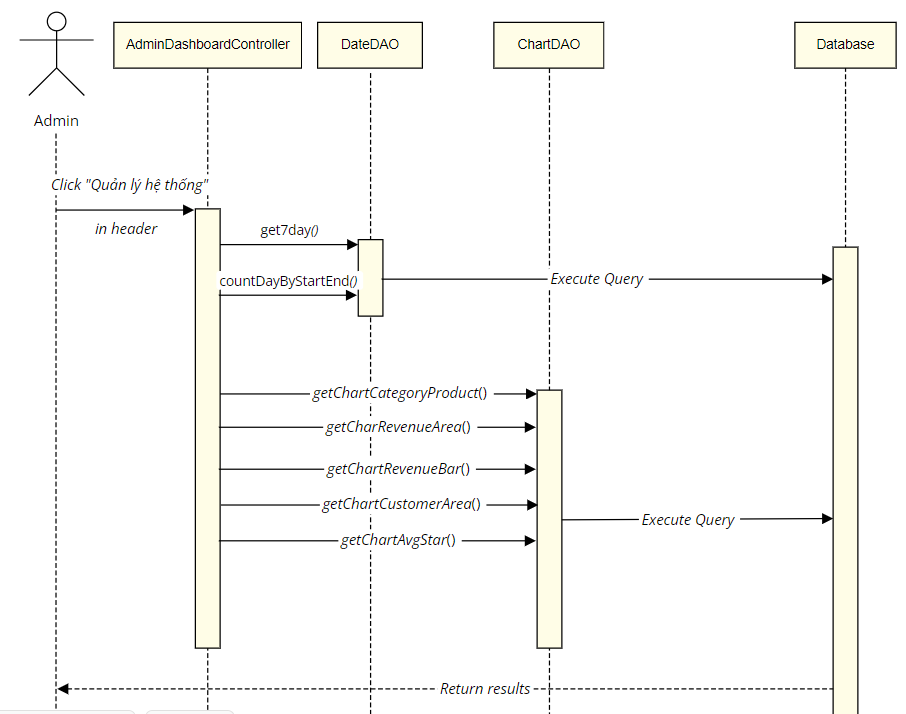


### b. Class Specifications

#### XYZ Class

| **No** | **Method** | **Description** |
| --- | --- | --- |
| *01* | *get7day()* | *Get last seven days* |
| *02* | *countDayByStartEnd()* | *Count the number of days difference between start and end* |
| *03* | *getChartCategoryProduct()* | *Get chart for each category of product* |
| *04* | *getChartRevenueBar()* | *Get statistics by bar chart from start date to end date of revenue* |
| *05* | *getChartRevenueArea()* | *Get statistics by area chart from start date to end date of revenue* |
| *06* | *getChartCustomerArea()* | *Get statistics by area chart from start date to end date of customer* |
| *07* | *getChartAvgStar()* | *Get statistics chart for average rating stars* |

### c. Sequence Diagram(s)

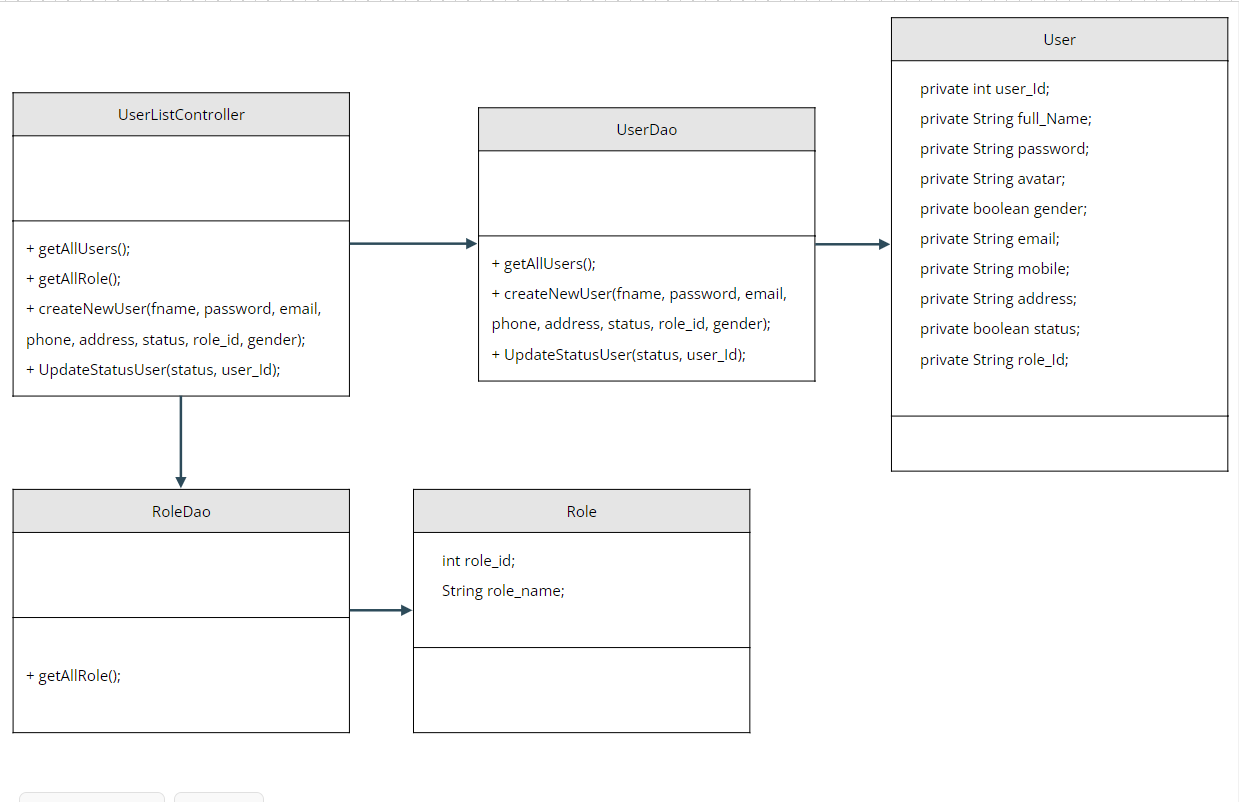
****

### d. Database queries

* select GETDATE(), GETDATE()-6
* SELECT DATEDIFF(day, ?, ?)
* select DATEADD(DAY, ?, ?)
* select COUNT(\*) from Product where update\_date < ?
* select COUNT(\*) from Product where category\_id = ? and update\_date < ?
* select sum(total\_cost) from [Order] where saler\_id " + salerId + " and orderDate <= DATEADD(DAY, ?, ?) and orderDate >= ?
* select count(\*) from Customer where updated\_date <= DATEADD(DAY, ?, ?)
* select CAST(AVG(rated\_star) AS DECIMAL(10,1)) from Feedback where date < DATEADD(DAY, ?, ?)

## 33. Admin/Users List

### a. Class Diagram

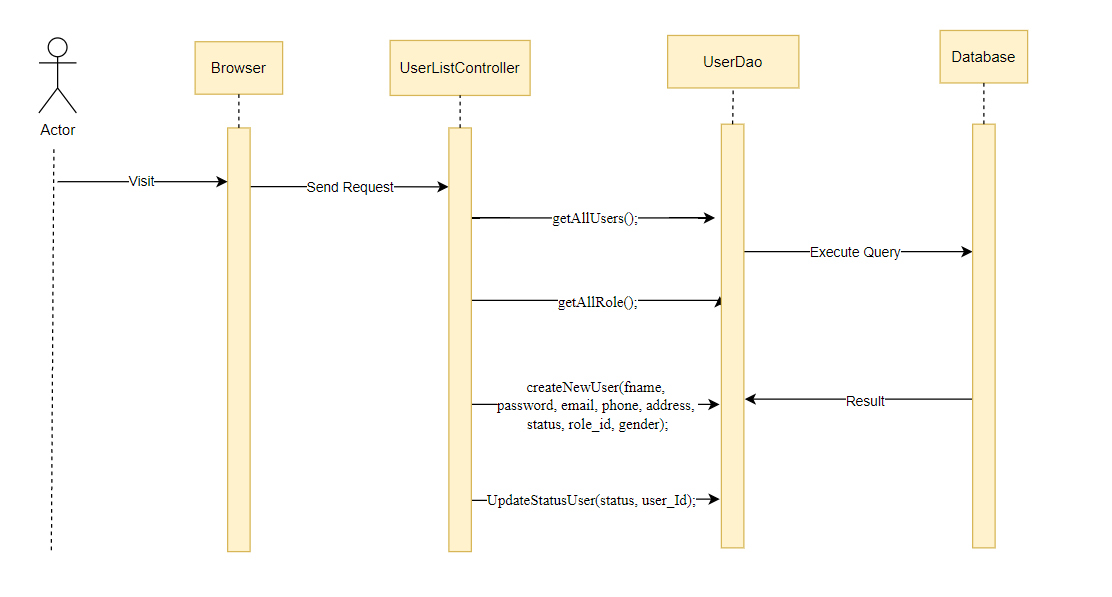


### b. Class Specifications

#### XYZ Class

| **No** | **Method** | **Description** |
| --- | --- | --- |
| *01* | *getAllUsers();* | *get all user in database* |
| *02* | *getAllRole();* | *get all role in database* |
| *03* | *createNewUser(fname, password, email, phone, address, status, role\_id, gender);* | *create new User to table User in database* |
| *04* | *UpdateStatusUser(status, user\_Id);* | *update status to table User in database* |

### c. Sequence Diagram(s)



### d. Database queries

select \* from [User]

select \* from Role

UPDATE [dbo].[User]\n"

+ " SET [status] = ?\n"

+ " WHERE userId = ?"

INSERT INTO [dbo].[User]\n"

+ " ([fullName]\n"

+ " ,[password]\n"

+ " ,[email]\n"

+ " ,[mobile]\n"

+ " ,[address]\n"

+ " ,[status]\n"

+ " ,[role\_id]\n"

+ " ,[gender])\n"

+ " VALUES\n"

+ " (?,?,?,?,?,?,?,?)"

## 34. Admin/User Details

### a. Class Diagram

### b. Class Specifications

#### XYZ Class

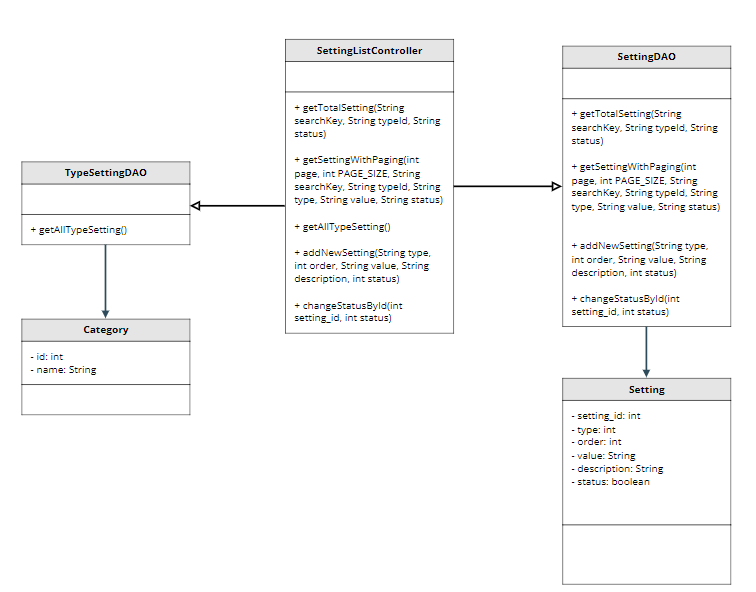
| **No** | **Method** | **Description** |
| --- | --- | --- |
| *01* |  |  |
| *02* |  |  |
| *03* |  |  |
| *04* |  |  |

### c. Sequence Diagram(s)

### d. Database queries

## 35. Admin/Settings List

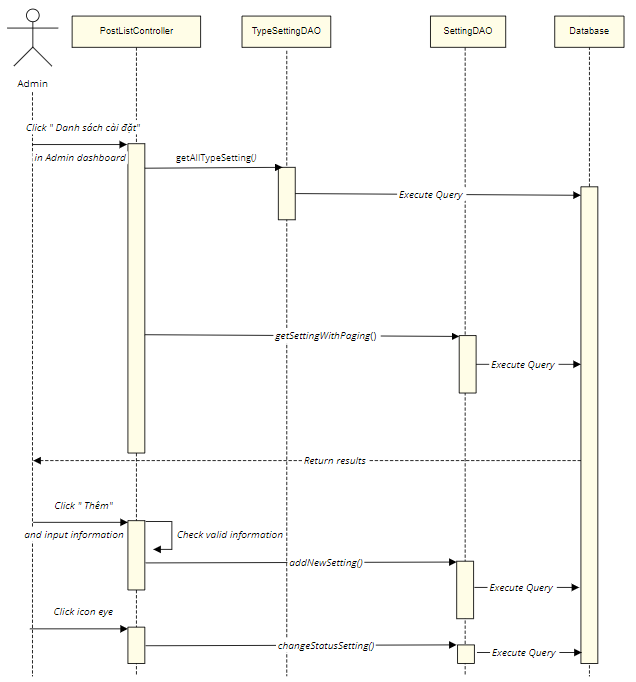
### a. Class Diagram

****

### b. Class Specifications

| **No** | **Method** | **Description** |
| --- | --- | --- |
| *01* | *getTotalSetting()* | *Get total setting in the database* |
| *02* | *getSettingWithPaging()* | *Get list setting pagination* |
| *03* | *getAllTypeSetting()* | *Get list type setting in the database* |
| *04* | *addNewSetting()* | *Add new setting* |
| *05* | *changeStatusById()* | *Change status of specific setting* |

### c. Sequence Diagram(s)

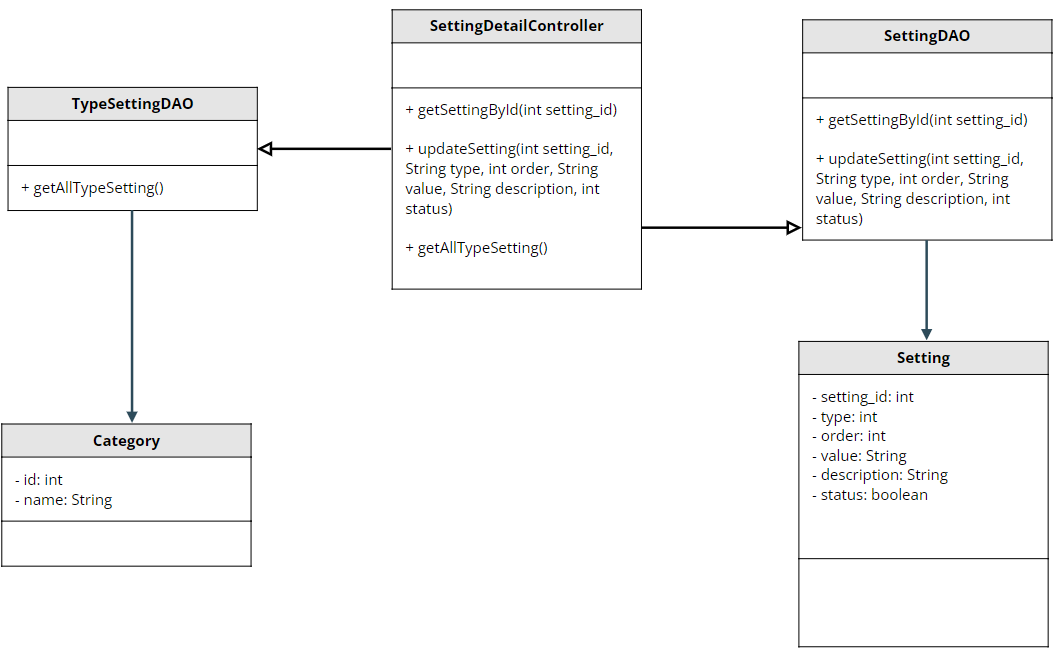


### d. Database queries

* Select count(\*) from Setting where [type] " + typeId + " and [status] " + status + " and [value] like N'%" +searchKey + "%'
* select \* from Setting where [type] " + typeId + " and [status] " + status + " and [value] like N'%" +searchKey + "%' order by " + value + " " + type + " offset (?-1)\*? row fetch next ? row only
* INSERT INTO [dbo].[Setting] ([type] ,[order],[value],[description],[status]) VALUES (?,?,?,?,?)
* select \* from Setting\_Type

## 36. Admin/Setting Details

### a. Class Diagram

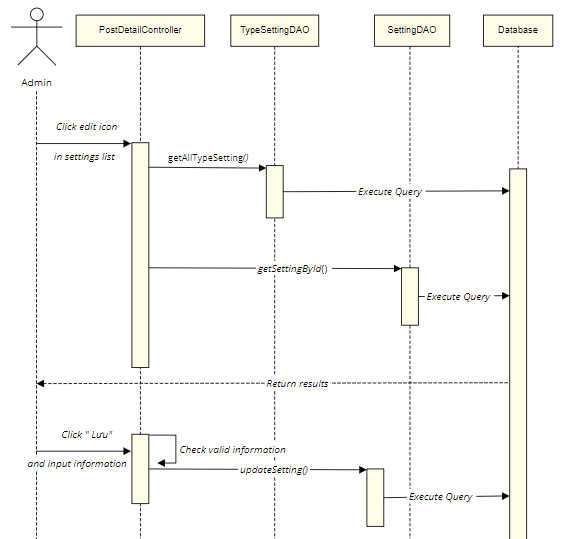


### b. Class Specifications

#### XYZ Class

| **No** | **Method** | **Description** |
| --- | --- | --- |
| *01* | *getSettingById(int setting\_id)* | *Get specific setting by id* |
| *02* | *updateSetting()* | *Update setting by id* |
| *03* | *getAllTypeSetting()* | *Get list type setting in the database* |

### c. Sequence Diagram(s)

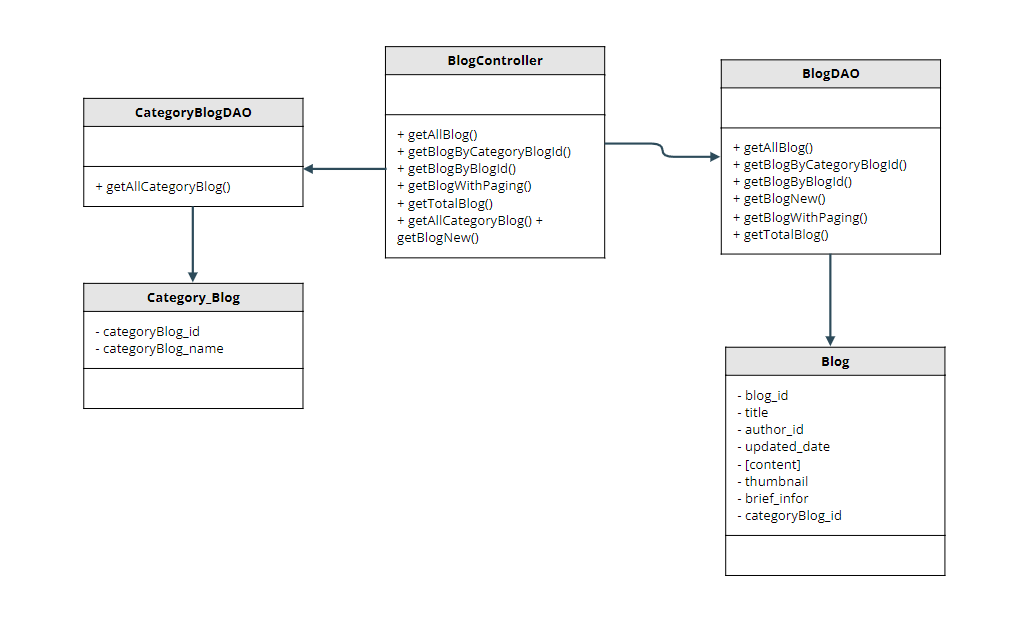


### d. Database queries

* select \* from Setting s join Setting\_Type st on s.type = st.setting\_type\_id where setting\_id = ?
* UPDATE [dbo].[Setting] SET [value] = ?, [description] = ?, [status] = ? WHERE setting\_id = ?
* UPDATE [dbo].[Category] SET [category\_name] = ?,[status] = ? WHERE category\_id = ?
* UPDATE [dbo].[Category\_Blog] SET [categoryBlog\_name] = ? ,[status] = ? WHERE categoryBlog\_id = ?
* UPDATE [dbo].[Status\_Order] SET [status\_name] = ? ,[status] = ? WHERE status\_order\_id = ?
* UPDATE [dbo].[Role] SET [role\_name] = ?,[status] = ? WHERE role\_id = ?

## 37. Public/Blog List

### a. Class Diagram

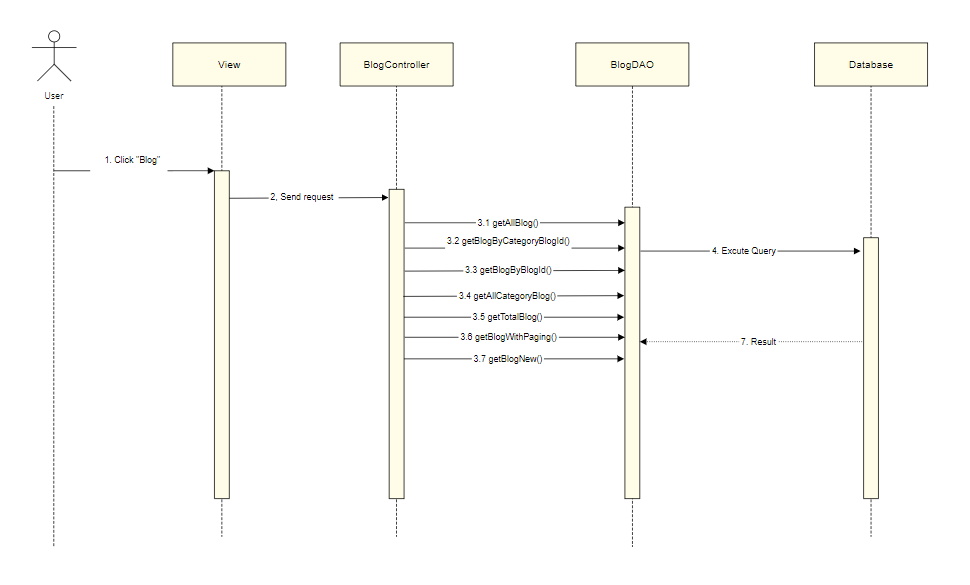


### b. Class Specifications

#### XYZ Class

| **No** | **Method** | **Description** |
| --- | --- | --- |
| *01* | *getAllBlog* | *get all blog in database* |
| *02* | *getBlogByCategoryBlogId* | *get blog by the category blog id* |
| *03* | *getBlogByBlogId* | *get blog by blog id* |
| *04* | *getBlogWithPaging* | *get blog with paging* |
| *05* | *getTotalBlog* | *get all blog by count* |
| *06* | *getAllCategoryBlog* | *get all category blog* |
| *07* | *getBlogNew* | *get new blog order by date* |

### c. Sequence Diagram(s)



### d. Database queries

Select \* from Blog

Select \* from Blog where categoryBlog\_id = ?

Select \* from Blog where blog\_id = ?

Select \* from Category\_Blog

select top 1 \* from Blog order by updated\_date desc

Select count(\*) from Blog where categoryBlog\_id + categoryId + " and title like N'% + searchKey + %'

select \* from Blog

+ "where categoryBlog\_id " + categoryId + " and title like N'%" + searchKey + "%'\n"

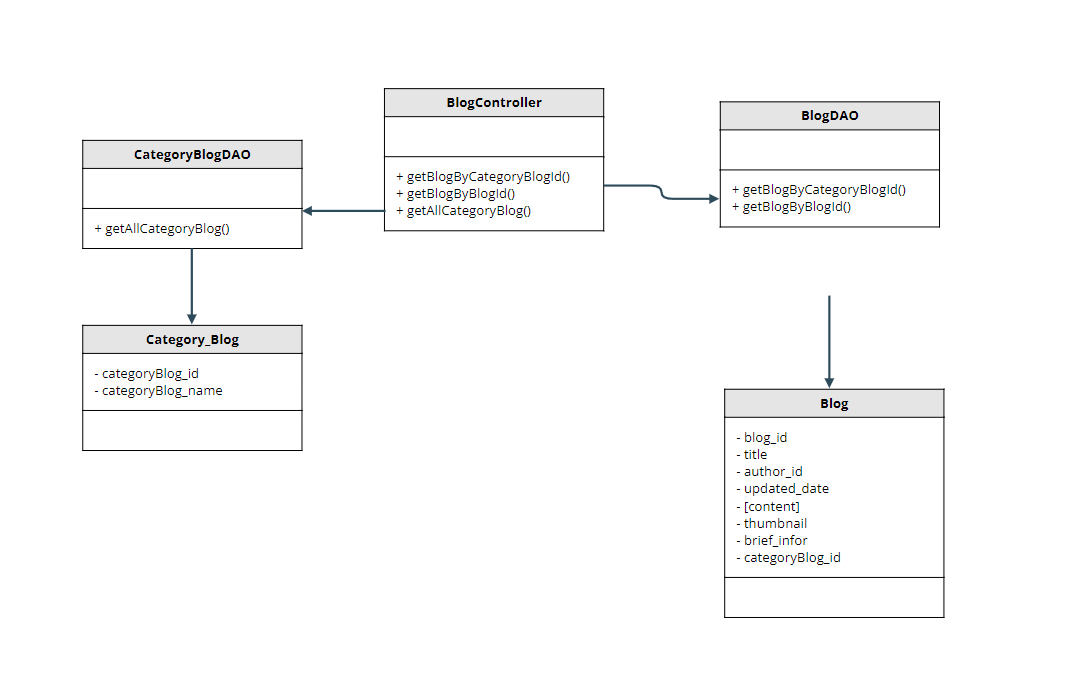
+ " order by " + value + " " + type + " offset (?-1)\*? row fetch next ? row only

Select count(\*) from Blog "

+ "where categoryBlog\_id " + categoryId + " and [status] " + status + " and author\_id " + author + " and title like N'%" + searchKey + "%'\n

## 38. Public/ Blog Details

### a. Class Diagram

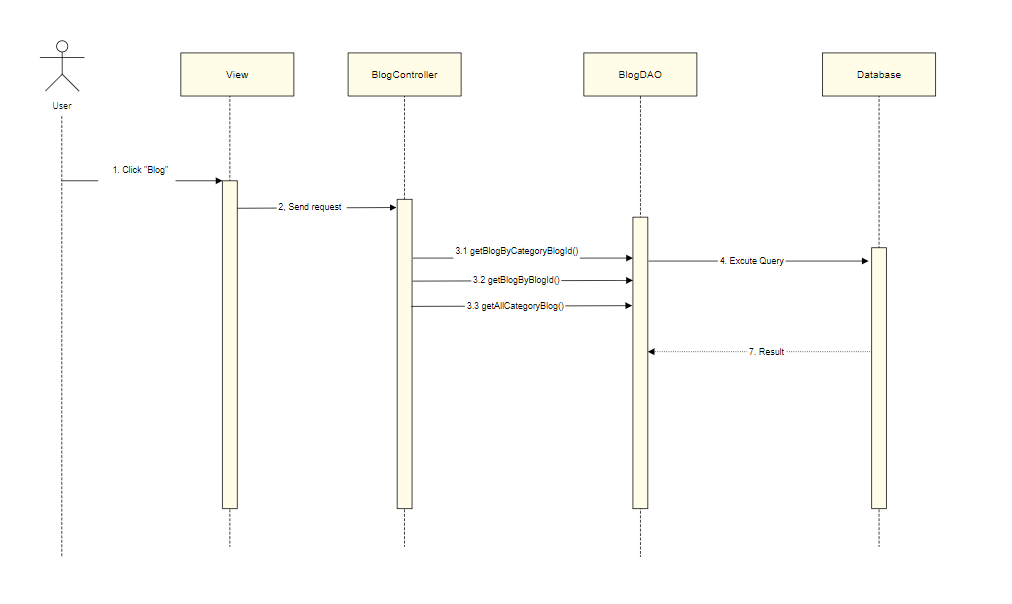


### b. Class Specifications

#### XYZ Class

| **No** | **Method** | **Description** |
| --- | --- | --- |
| *01* | *getAllBlog* | *get all blog in database* |
| *02* | *getBlogByCategoryBlogId* | *get blog by the category blog id* |
| *03* | *getBlogByBlogId* | *get blog by blog id* |

### c. Sequence Diagram(s)



### d. Database queries

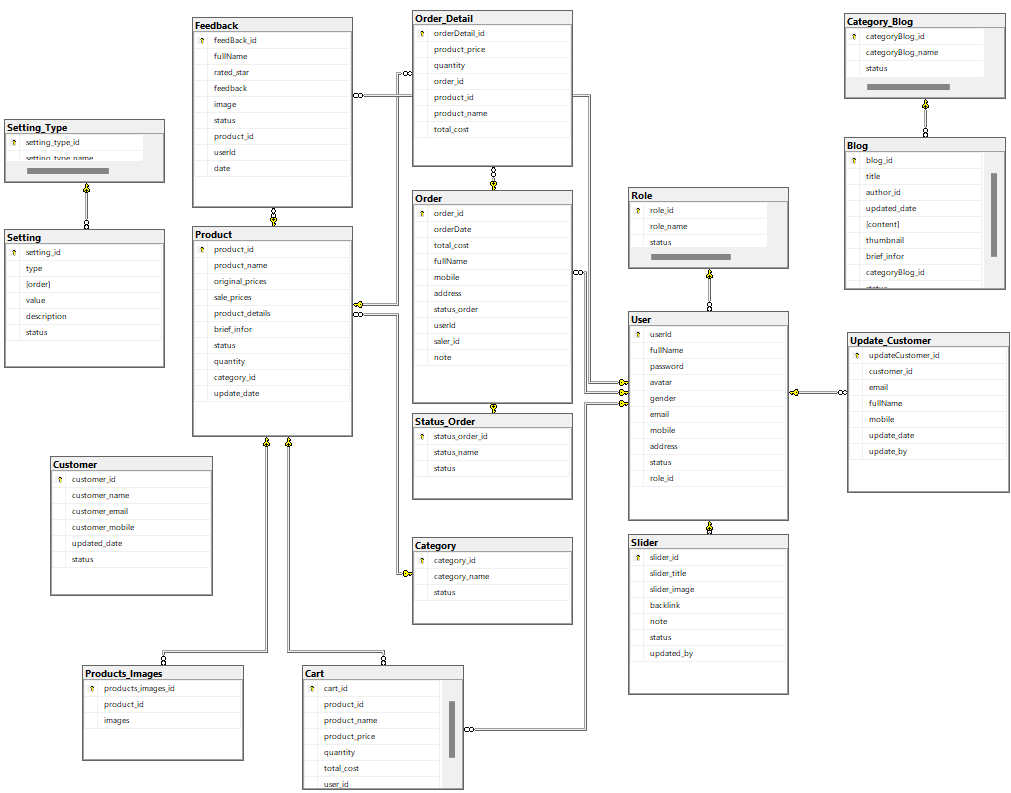
Select \* from Blog

Select \* from Blog where categoryBlog\_id = ?

Select \* from Blog where blog\_id = ?

# III. Database Design

## 1. Database Schema



## 2. Table Description

| **#** | **Table** | **Description** |
| --- | --- | --- |
| 1 | User | userId int ,  fullName nvarchar(50),  password varchar(50),  avatar varbinary(max),  gender bit,  email varchar(50),  mobile varchar(50),  role\_id int,  [address] nvarchar(255),  [status] nvarchar(50)  **Primary key: userId**  **Foreign key: role\_id** |
| 2 | Role | role\_id int,  role\_name varchar(50)  **Primary key: role\_id** |
| 3 | Category | category\_id int,  category\_name nvarchar(50)  Status bit  **Primary key: category\_id** |
| 4 | Product | product\_id,  product\_name nvarchar(255),  category\_id int,  brief\_infor  original\_prices float,  sale\_prices float,  product\_details nvarchar(max),  quantity int,  Status nvarchar(50)  **Primary key: product\_id**  **Foreign key: category\_id** |
| 5 | Product\_Images | Product\_Images\_id int,  Product\_id int,  images varchar(max)  **Primary key: statusOrder\_id**  **Foreign key: Product\_id** |
| 6 | Order | order\_id int,  fullName nvarchar(50),  email varchar(50),  mobile varchar(50),  Address nvarchar(255),  orderDate date,  total\_cost float,  statusOrder\_id int,  user\_id int  **Primary key: order\_id**  **Foreign key: user\_id** |
| 7 | orderDetail | orderDetail\_id int,  product\_id int,  prices float,  quantity int,  order\_id int  **Primary key: orderDetail\_id**  **Foreign key:product\_id int\_id** |
| 8 | Feedback | feedBack\_id int,  fullName nvarchar(50),  rated\_star float,  feedback nvarchar(max),  image\_status varbinary(max),  Status bit,  product\_id int,  userId int,  **Primary key: feedBack\_id**  **Foreign key: product\_id, userId** |
| 9 | categoryBlog | categoryBlog\_id int,  categoryBlog\_name nvarchar(50),  Status int  **Primary key: categoryBlog\_id** |
| 10 | Blog | blog\_id int,  title nvarchar(50),  author\_id int,  updated\_date date,  content text,  Thumbnail text,  Brief\_infor nvarchar(255)  categoryBlog\_id int,  **Primary key: blog\_id**  **Foreign key: author\_id, categoryBlog\_id** |
| 11 | UpdateCustomer | udateCustomer\_id int,  email varchar(50),  fullName nvarchar(50),  Gender bit,  mobile varchar(50),  Address nvarchar(255)  updated\_date date,  updated\_by int,  **Primary key: udateCustomer\_id**  **Foreign key: updated\_by** |
| 12 | Slider | slider\_id int,  slider\_title nvarchar(50),  Slider\_images text,  Backlink varchar(max),  Note nvarchar(50),  Status int,  update\_by int,  **Primary key: slider\_id**  **Foreign key: update\_by** |
| 13 | Status\_Order | status\_order\_id int,  status\_name nvarchar(50),  Status bit,  **Primary key: status\_order\_id** |
| 14 | Customer | customer\_id int,  customer\_name nvarchar(50),  customer\_email varchar(50),  customer\_mobile varchar(50),  updated\_date date,  status bit  **Primary key: customer\_id** |
| 15 | Cart | cart\_id int,  product\_id int,  product\_name nvarchar(255),  product\_price int,  quantity int,  total\_cost int,  user\_id int  **Primary key: cart\_id**  **Foreign key: product\_id, user\_id** |
| 16 | Setting | setting\_id int,  Type int,  [order] int,  value nvarchar(50),  description nvarchar(50),  Status bit  **Primary key: setting\_id**  **Foreign key: type** |
| 17 | Setting\_Type | Setting\_type\_id int,  Setting\_type\_name nvarchar(50)  **Primary key: Setting\_type\_id** |