x



**FASHION SHOP ONLINE**

**Software Design Document**

**Table of Contents**

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

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

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

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

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

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

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

[XYZ Class](#_heading=h.1t3h5sf) [11](#_heading=h.1t3h5sf)

[c. Sequence Diagram(s)](#_heading=h.4d34og8) [11](#_heading=h.4d34og8)

[d. Database queries](#_heading=h.2s8eyo1) [11](#_heading=h.2s8eyo1)

[2. Public/Blogs List](#_heading=h.17dp8vu) [12](#_heading=h.17dp8vu)

[a. Class Diagram](#_heading=h.3rdcrjn) [12](#_heading=h.3rdcrjn)

[b. Class Specifications](#_heading=h.26in1rg) [12](#_heading=h.26in1rg)

[XYZ Class](#_heading=h.lnxbz9) [12](#_heading=h.lnxbz9)

[c. Sequence Diagram(s)](#_heading=h.35nkun2) [13](#_heading=h.35nkun2)

[d. Database queries](#_heading=h.1ksv4uv) [13](#_heading=h.1ksv4uv)

[3. Public/Blog Details](#_heading=h.44sinio) [14](#_heading=h.44sinio)

[a. Class Diagram](#_heading=h.2jxsxqh) [14](#_heading=h.2jxsxqh)

[b. Class Specifications](#_heading=h.z337ya) [14](#_heading=h.z337ya)

[XYZ Class](#_heading=h.3j2qqm3) [14](#_heading=h.3j2qqm3)

[c. Sequence Diagram(s)](#_heading=h.2xcytpi) [15](#_heading=h.2xcytpi)

[d. Database queries](#_heading=h.1ci93xb) [15](#_heading=h.1ci93xb)

[4. Public/Products List](#_heading=h.3whwml4) [16](#_heading=h.3whwml4)

[a. Class Diagram](#_heading=h.2bn6wsx) [16](#_heading=h.2bn6wsx)

[b. Class Specifications](#_heading=h.qsh70q) [16](#_heading=h.qsh70q)

[c. Sequence Diagram(s)](#_heading=h.3as4poj) [17](#_heading=h.3as4poj)

[d. Database queries](#_heading=h.1pxezwc) [17](#_heading=h.1pxezwc)

[5. Public/Product Details](#_heading=h.49x2ik5) [18](#_heading=h.49x2ik5)

[a. Class Diagram](#_heading=h.2p2csry) [18](#_heading=h.2p2csry)

[b. Class Specifications](#_heading=h.147n2zr) [18](#_heading=h.147n2zr)

[XYZ Class](#_heading=h.3o7alnk) [18](#_heading=h.3o7alnk)

[c. Sequence Diagram(s)](#_heading=h.23ckvvd) [18](#_heading=h.23ckvvd)

[d. Database queries](#_heading=h.ihv636) [18](#_heading=h.ihv636)

[6. Public/Cart Details](#_heading=h.32hioqz) [18](#_heading=h.32hioqz)

[a. Class Diagram](#_heading=h.1hmsyys) [18](#_heading=h.1hmsyys)

[b. Class Specifications](#_heading=h.41mghml) [19](#_heading=h.41mghml)

[XYZ Class](#_heading=h.2grqrue) [19](#_heading=h.2grqrue)

[c. Sequence Diagram(s)](#_heading=h.vx1227) [19](#_heading=h.vx1227)

[d. Database queries](#_heading=h.3fwokq0) [19](#_heading=h.3fwokq0)

[7. Public/Cart Contact](#_heading=h.1v1yuxt) [20](#_heading=h.1v1yuxt)

[a. Class Diagram](#_heading=h.4f1mdlm) [20](#_heading=h.4f1mdlm)

[b. Class Specifications](#_heading=h.2u6wntf) [20](#_heading=h.2u6wntf)

[XYZ Class](#_heading=h.19c6y18) [20](#_heading=h.19c6y18)

[c. Sequence Diagram(s)](#_heading=h.3tbugp1) [21](#_heading=h.3tbugp1)

[d. Database queries](#_heading=h.28h4qwu) [21](#_heading=h.28h4qwu)

[8. Public/Cart Completion](#_heading=h.nmf14n) [22](#_heading=h.nmf14n)

[a. Class Diagram](#_heading=h.37m2jsg) [22](#_heading=h.37m2jsg)

[b. Class Specifications](#_heading=h.1mrcu09) [22](#_heading=h.1mrcu09)

[XYZ Class](#_heading=h.46r0co2) [22](#_heading=h.46r0co2)

[c. Sequence Diagram(s)](#_heading=h.2lwamvv) [23](#_heading=h.2lwamvv)

[d. Database queries](#_heading=h.111kx3o) [23](#_heading=h.111kx3o)

[9. Public/Feedback](#_heading=h.3l18frh) [23](#_heading=h.3l18frh)

[a. Class Diagram](#_heading=h.1au1eum) [24](#_heading=h.1au1eum)

[b. Class Specifications](#_heading=h.3utoxif) [24](#_heading=h.3utoxif)

[XYZ Class](#_heading=h.29yz7q8) [24](#_heading=h.29yz7q8)

[c. Sequence Diagram(s)](#_heading=h.p49hy1) [25](#_heading=h.p49hy1)

[d. Database queries](#_heading=h.393x0lu) [25](#_heading=h.393x0lu)

[11. Common/Login](#_heading=h.206ipza) [26](#_heading=h.206ipza)

[a. Class Diagram](#_heading=h.4k668n3) [26](#_heading=h.4k668n3)

[b. Class Specifications](#_heading=h.2zbgiuw) [26](#_heading=h.2zbgiuw)

[c. Sequence Diagram(s)](#_heading=h.3ygebqi) [27](#_heading=h.3ygebqi)

[d. Database queries](#_heading=h.2dlolyb) [27](#_heading=h.2dlolyb)

[6. Common/Register](#_heading=h.sqyw64) [27](#_heading=h.sqyw64)

[a. Class Diagram](#_heading=h.3cqmetx) [28](#_heading=h.3cqmetx)

[b. Class Specifications](#_heading=h.1rvwp1q) [28](#_heading=h.1rvwp1q)

[c. Sequence Diagram(s)](#_heading=h.4bvk7pj) [28](#_heading=h.4bvk7pj)

[d. Database queries](#_heading=h.2r0uhxc) [29](#_heading=h.2r0uhxc)

[12. Common/Reset Password](#_heading=h.1664s55) [29](#_heading=h.1664s55)

[a. Class Diagram](#_heading=h.3q5sasy) [29](#_heading=h.3q5sasy)

[c. Sequence Diagram(s)](#_heading=h.25b2l0r) [30](#_heading=h.25b2l0r)

[d. Database queries](#_heading=h.kgcv8k) [30](#_heading=h.kgcv8k)

[13. Common/Change Password](#_heading=h.34g0dwd) [31](#_heading=h.34g0dwd)

[a. Class Diagram](#_heading=h.1jlao46) [31](#_heading=h.1jlao46)

[b. Class Specifications](#_heading=h.43ky6rz) [31](#_heading=h.43ky6rz)

[c. Sequence Diagram(s)](#_heading=h.2iq8gzs) [32](#_heading=h.2iq8gzs)

[d. Database queries](#_heading=h.xvir7l) [32](#_heading=h.xvir7l)

[14. Common/User Profile](#_heading=h.3hv69ve) [32](#_heading=h.3hv69ve)

[a. Class Diagram](#_heading=h.1x0gk37) [33](#_heading=h.1x0gk37)

[b. Class Specifications](#_heading=h.4h042r0) [33](#_heading=h.4h042r0)

[c. Sequence Diagram(s)](#_heading=h.2w5ecyt) [33](#_heading=h.2w5ecyt)

[d. Database queries](#_heading=h.1baon6m) [34](#_heading=h.1baon6m)

[15. Common/User Authorization](#_heading=h.3vac5uf) [35](#_heading=h.3vac5uf)

[a. Class Diagram](#_heading=h.2afmg28) [35](#_heading=h.2afmg28)

[b. Class Specifications](#_heading=h.pkwqa1) [35](#_heading=h.pkwqa1)

[c. Sequence Diagram(s)](#_heading=h.39kk8xu) [36](#_heading=h.39kk8xu)

[d. Database queries](#_heading=h.1opuj5n) [36](#_heading=h.1opuj5n)

[16. Customer/My Orders](#_heading=h.48pi1tg) [36](#_heading=h.48pi1tg)

[a. Class Diagram](#_heading=h.2nusc19) [37](#_heading=h.2nusc19)

[b. Class Specifications](#_heading=h.1302m92) [37](#_heading=h.1302m92)

[XYZ Class](#_heading=h.3mzq4wv) [37](#_heading=h.3mzq4wv)

[c. Sequence Diagram(s)](#_heading=h.2250f4o) [37](#_heading=h.2250f4o)

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

[17. Customer/Order Information](#_heading=h.319y80a) [39](#_heading=h.319y80a)

[a. Class Diagram](#_heading=h.1gf8i83) [39](#_heading=h.1gf8i83)

[b. Class Specifications](#_heading=h.40ew0vw) [39](#_heading=h.40ew0vw)

[XYZ Class](#_heading=h.2fk6b3p) [39](#_heading=h.2fk6b3p)

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

[d. Database queries](#_heading=h.3ep43zb) [40](#_heading=h.3ep43zb)

[18.Staff/MKT Dashboard](#_heading=h.1tuee74) [41](#_heading=h.1tuee74)

[a. Class Diagram](#_heading=h.4du1wux) [41](#_heading=h.4du1wux)

[b. Class Specifications](#_heading=h.2szc72q) [41](#_heading=h.2szc72q)

[XYZ Class](#_heading=h.184mhaj) [41](#_heading=h.184mhaj)

[c. Sequence Diagram(s)](#_heading=h.3s49zyc) [41](#_heading=h.3s49zyc)

[d. Database queries](#_heading=h.279ka65) [42](#_heading=h.279ka65)

[19.Staff/Posts List](#_heading=h.meukdy) [42](#_heading=h.meukdy)

[a. Class Diagram](#_heading=h.36ei31r) [43](#_heading=h.36ei31r)

[b. Class Specifications](#_heading=h.1ljsd9k) [43](#_heading=h.1ljsd9k)

[XYZ Class](#_heading=h.45jfvxd) [43](#_heading=h.45jfvxd)

[c. Sequence Diagram(s)](#_heading=h.2koq656) [43](#_heading=h.2koq656)

[d. Database queries](#_heading=h.zu0gcz) [44](#_heading=h.zu0gcz)

[20. Staff/Post Details](#_heading=h.3jtnz0s) [45](#_heading=h.3jtnz0s)

[a. Class Diagram](#_heading=h.1yyy98l) [45](#_heading=h.1yyy98l)

[b. Class Specifications](#_heading=h.4iylrwe) [45](#_heading=h.4iylrwe)

[XYZ Class](#_heading=h.2y3w247) [45](#_heading=h.2y3w247)

[c. Sequence Diagram(s)](#_heading=h.1d96cc0) [46](#_heading=h.1d96cc0)

[d. Database queries](#_heading=h.3x8tuzt) [46](#_heading=h.3x8tuzt)

[21. Staff/Sliders List](#_heading=h.2ce457m) [47](#_heading=h.2ce457m)

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

[b. Class Specifications](#_heading=h.3bj1y38) [47](#_heading=h.3bj1y38)

[XYZ Class](#_heading=h.1qoc8b1) [47](#_heading=h.1qoc8b1)

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

[d. Database queries](#_heading=h.2pta16n) [48](#_heading=h.2pta16n)

[22. Staff/Slider Details](#_heading=h.14ykbeg) [49](#_heading=h.14ykbeg)

[a. Class Diagram](#_heading=h.3oy7u29) [49](#_heading=h.3oy7u29)

[b. Class Specifications](#_heading=h.243i4a2) [49](#_heading=h.243i4a2)

[XYZ Class](#_heading=h.j8sehv) [49](#_heading=h.j8sehv)

[c. Sequence Diagram(s)](#_heading=h.338fx5o) [49](#_heading=h.338fx5o)

[d. Database queries](#_heading=h.1idq7dh) [50](#_heading=h.1idq7dh)

[23. Staff/Products List](#_heading=h.42ddq1a) [50](#_heading=h.42ddq1a)

[a. Class Diagram](#_heading=h.2hio093) [50](#_heading=h.2hio093)

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

[c. Sequence Diagram(s)](#_heading=h.1vsw3ci) [51](#_heading=h.1vsw3ci)

[d. Database queries](#_heading=h.4fsjm0b) [51](#_heading=h.4fsjm0b)

[24. Staff/Product Details](#_heading=h.2uxtw84) [52](#_heading=h.2uxtw84)

[a. Class Diagram](#_heading=h.1a346fx) [52](#_heading=h.1a346fx)

[b. Class Specifications](#_heading=h.3u2rp3q) [52](#_heading=h.3u2rp3q)

[XYZ Class](#_heading=h.2981zbj) [52](#_heading=h.2981zbj)

[c. Sequence Diagram(s)](#_heading=h.odc9jc) [53](#_heading=h.odc9jc)

[d. Database queries](#_heading=h.38czs75) [53](#_heading=h.38czs75)

[25. Staff/Customers List](#_heading=h.1nia2ey) [55](#_heading=h.1nia2ey)

[a. Class Diagram](#_heading=h.47hxl2r) [55](#_heading=h.47hxl2r)

[b. Class Specifications](#_heading=h.2mn7vak) [55](#_heading=h.2mn7vak)

[c. Sequence Diagram(s)](#_heading=h.11si5id) [55](#_heading=h.11si5id)

[d. Database queries](#_heading=h.3ls5o66) [56](#_heading=h.3ls5o66)

[26.Staff/Customer Details](#_heading=h.20xfydz) [57](#_heading=h.20xfydz)

[a. Class Diagram](#_heading=h.4kx3h1s) [57](#_heading=h.4kx3h1s)

[b. Class Specifications](#_heading=h.302dr9l) [57](#_heading=h.302dr9l)

[c. Sequence Diagram(s)](#_heading=h.1f7o1he) [57](#_heading=h.1f7o1he)

[d. Database queries](#_heading=h.3z7bk57) [58](#_heading=h.3z7bk57)

[27. Staff/Feedbacks List](#_heading=h.2eclud0) [59](#_heading=h.2eclud0)

[a. Class Diagram](#_heading=h.thw4kt) [59](#_heading=h.thw4kt)

[b. Class Specifications](#_heading=h.3dhjn8m) [59](#_heading=h.3dhjn8m)

[XYZ Class](#_heading=h.1smtxgf) [59](#_heading=h.1smtxgf)

[c. Sequence Diagram(s)](#_heading=h.4cmhg48) [59](#_heading=h.4cmhg48)

[d. Database queries](#_heading=h.2rrrqc1) [60](#_heading=h.2rrrqc1)

[28.Staff/Feedback Details](#_heading=h.16x20ju) [60](#_heading=h.16x20ju)

[a. Class Diagram](#_heading=h.3qwpj7n) [60](#_heading=h.3qwpj7n)

[b. Class Specifications](#_heading=h.261ztfg) [60](#_heading=h.261ztfg)

[XYZ Class](#_heading=h.l7a3n9) [60](#_heading=h.l7a3n9)

[c. Sequence Diagram(s)](#_heading=h.356xmb2) [60](#_heading=h.356xmb2)

[d. Database queries](#_heading=h.1kc7wiv) [61](#_heading=h.1kc7wiv)

[29. Staff/Sale Dashboard](#_heading=h.44bvf6o) [61](#_heading=h.44bvf6o)

[a. Class Diagram](#_heading=h.2jh5peh) [62](#_heading=h.2jh5peh)

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

[XYZ Class](#_heading=h.3im3ia3) [62](#_heading=h.3im3ia3)

[c. Sequence Diagram(s)](#_heading=h.1xrdshw) [63](#_heading=h.1xrdshw)

[d. Database queries](#_heading=h.4hr1b5p) [63](#_heading=h.4hr1b5p)

[30. Staff/Orders List](#_heading=h.2wwbldi) [64](#_heading=h.2wwbldi)

[a. Class Diagram](#_heading=h.1c1lvlb) [65](#_heading=h.1c1lvlb)

[b. Class Specifications](#_heading=h.3w19e94) [65](#_heading=h.3w19e94)

[XYZ Class](#_heading=h.2b6jogx) [65](#_heading=h.2b6jogx)

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

[d. Database queries](#_heading=h.3abhhcj) [66](#_heading=h.3abhhcj)

[31. Staff/Order Details](#_heading=h.1pgrrkc) [66](#_heading=h.1pgrrkc)

[a. Class Diagram](#_heading=h.49gfa85) [67](#_heading=h.49gfa85)

[b. Class Specifications](#_heading=h.2olpkfy) [67](#_heading=h.2olpkfy)

[XYZ Class](#_heading=h.13qzunr) [67](#_heading=h.13qzunr)

[c. Sequence Diagram(s)](#_heading=h.3nqndbk) [67](#_heading=h.3nqndbk)

[d. Database queries](#_heading=h.22vxnjd) [68](#_heading=h.22vxnjd)

[32. Admin/Admin Dashboard](#_heading=h.i17xr6) [69](#_heading=h.i17xr6)

[a. Class Diagram](#_heading=h.320vgez) [69](#_heading=h.320vgez)

[b. Class Specifications](#_heading=h.1h65qms) [69](#_heading=h.1h65qms)

[XYZ Class](#_heading=h.415t9al) [69](#_heading=h.415t9al)

[c. Sequence Diagram(s)](#_heading=h.2gb3jie) [69](#_heading=h.2gb3jie)

[d. Database queries](#_heading=h.vgdtq7) [70](#_heading=h.vgdtq7)

[33. Admin/Users List](#_heading=h.3fg1ce0) [70](#_heading=h.3fg1ce0)

[a. Class Diagram](#_heading=h.1ulbmlt) [71](#_heading=h.1ulbmlt)

[b. Class Specifications](#_heading=h.4ekz59m) [71](#_heading=h.4ekz59m)

[XYZ Class](#_heading=h.2tq9fhf) [71](#_heading=h.2tq9fhf)

[c. Sequence Diagram(s)](#_heading=h.18vjpp8) [71](#_heading=h.18vjpp8)

[d. Database queries](#_heading=h.3sv78d1) [72](#_heading=h.3sv78d1)

[34. Admin/User Details](#_heading=h.280hiku) [72](#_heading=h.280hiku)

[a. Class Diagram](#_heading=h.n5rssn) [73](#_heading=h.n5rssn)

[b. Class Specifications](#_heading=h.375fbgg) [73](#_heading=h.375fbgg)

[XYZ Class](#_heading=h.1maplo9) [73](#_heading=h.1maplo9)

[c. Sequence Diagram(s)](#_heading=h.46ad4c2) [73](#_heading=h.46ad4c2)

[d. Database queries](#_heading=h.2lfnejv) [73](#_heading=h.2lfnejv)

[35. Admin/Settings List](#_heading=h.10kxoro) [73](#_heading=h.10kxoro)

[a. Class Diagram](#_heading=h.3kkl7fh) [73](#_heading=h.3kkl7fh)

[b. Class Specifications](#_heading=h.1zpvhna) [74](#_heading=h.1zpvhna)

[c. Sequence Diagram(s)](#_heading=h.4jpj0b3) [74](#_heading=h.4jpj0b3)

[d. Database queries](#_heading=h.2yutaiw) [75](#_heading=h.2yutaiw)

[36. Admin/Setting Details](#_heading=h.1e03kqp) [75](#_heading=h.1e03kqp)

[a. Class Diagram](#_heading=h.3xzr3ei) [75](#_heading=h.3xzr3ei)

[b. Class Specifications](#_heading=h.2d51dmb) [75](#_heading=h.2d51dmb)

[XYZ Class](#_heading=h.sabnu4) [75](#_heading=h.sabnu4)

[c. Sequence Diagram(s)](#_heading=h.3c9z6hx) [75](#_heading=h.3c9z6hx)

[d. Database queries](#_heading=h.1rf9gpq) [76](#_heading=h.1rf9gpq)

[37. Public/Blog List](#_heading=h.4bewzdj) [76](#_heading=h.4bewzdj)

[a. Class Diagram](#_heading=h.2qk79lc) [77](#_heading=h.2qk79lc)

[b. Class Specifications](#_heading=h.15phjt5) [77](#_heading=h.15phjt5)

[XYZ Class](#_heading=h.3pp52gy) [77](#_heading=h.3pp52gy)

[c. Sequence Diagram(s)](#_heading=h.24ufcor) [77](#_heading=h.24ufcor)

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

[38. Public/ Blog Details](#_heading=h.33zd5kd) [78](#_heading=h.33zd5kd)

[a. Class Diagram](#_heading=h.1j4nfs6) [79](#_heading=h.1j4nfs6)

[b. Class Specifications](#_heading=h.434ayfz) [79](#_heading=h.434ayfz)

[XYZ Class](#_heading=h.2i9l8ns) [79](#_heading=h.2i9l8ns)

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

[d. Database queries](#_heading=h.3hej1je) [80](#_heading=h.3hej1je)

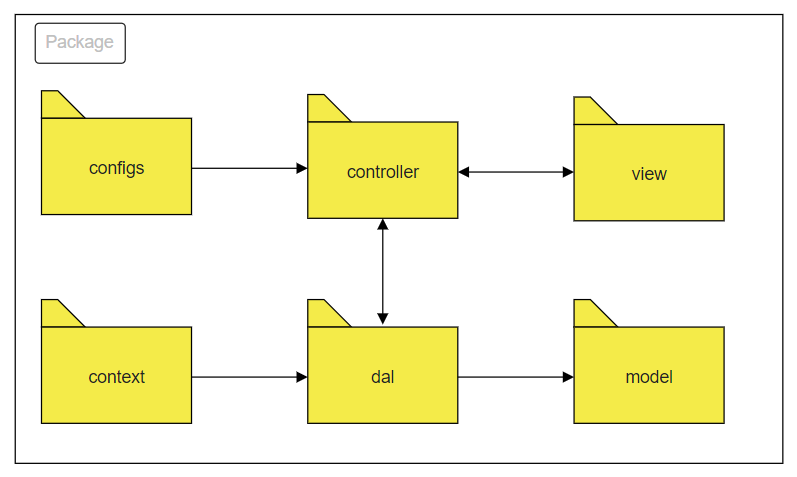
[**III. Database Design**](#_heading=h.1wjtbr7) **80**

[1. Database Schema](#_heading=h.4gjguf0) [81](#_heading=h.4gjguf0)

[2. Table Description](#_heading=h.2vor4mt) 81

# **I. Overview**

## **1. Code Packages/Namespaces**



**Package descriptions & package class naming conventions**

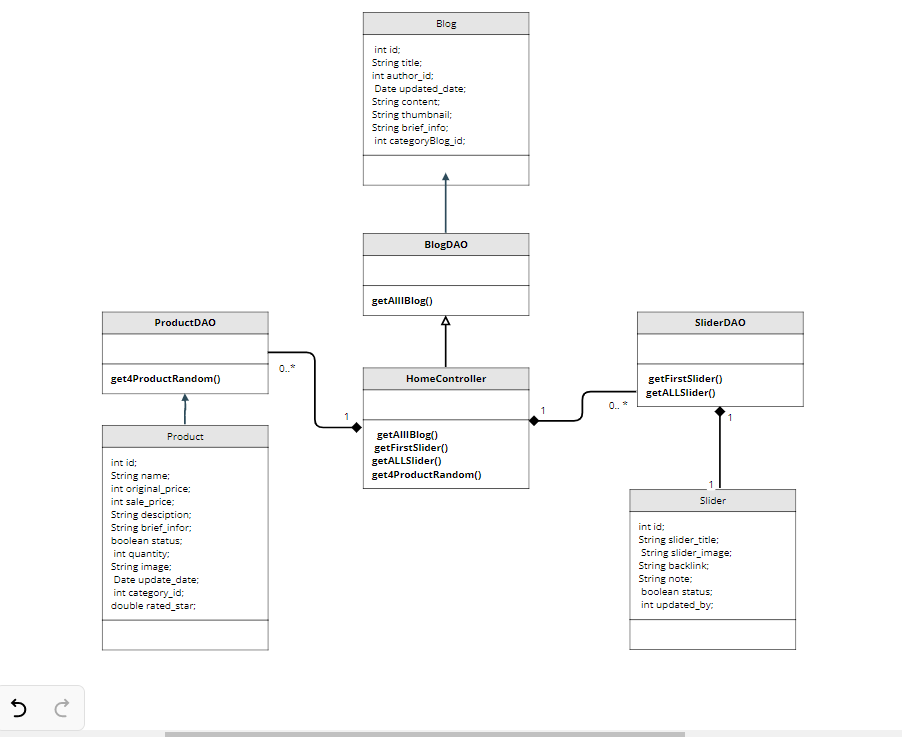
|  |  |  |
| --- | --- | --- |
| **No** | **Package** | **Description** |
| 01 | controller | Acts as a go-between for the service layer and the Front-end, passing on different input requests |
| 02 | dal | This component is used to get data directly FROM the database |
| 03 | configs | This component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

## **2. Coding Conventions**

# **II. Code Designs**

## **1. Public/Home Page**

### **a. Class Diagram**

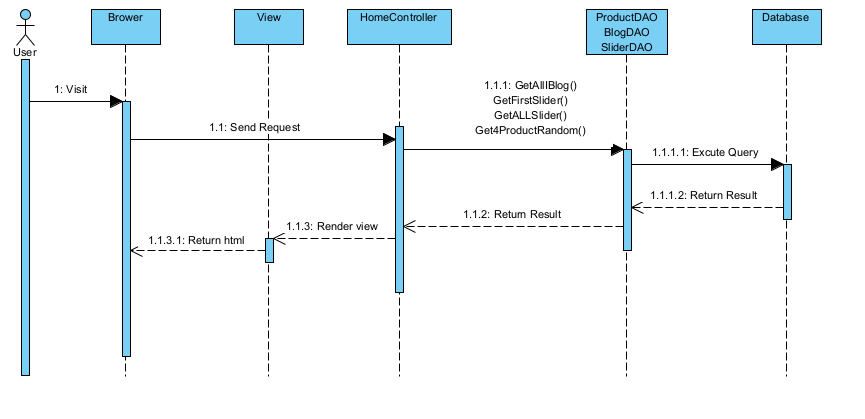


### **b. Class Specifications**

#### **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]

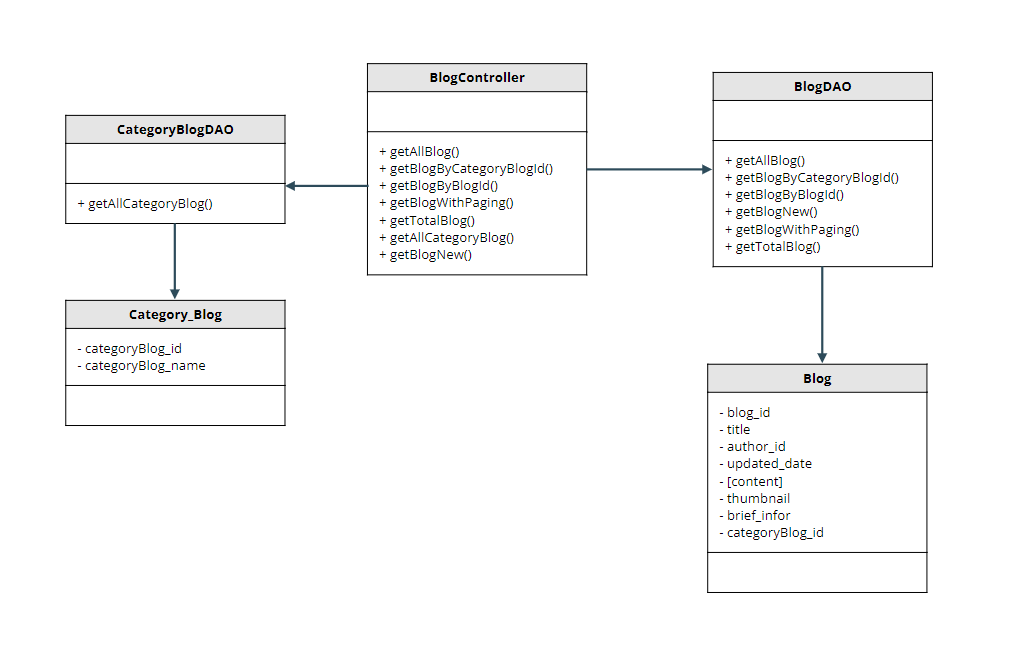
SELECT  \*  FROM [dbo].[Slider]

Select \* from Blog

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

## **2. Public/Blogs List**

### **a. Class Diagram**

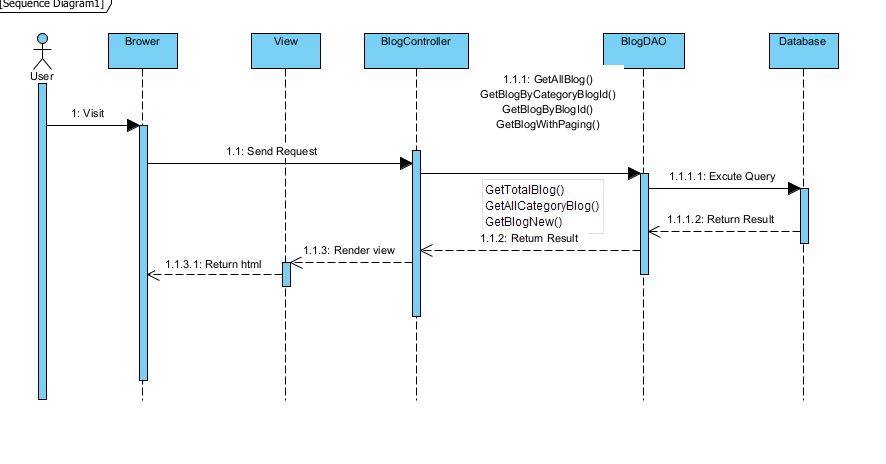


### **b. Class Specifications**

#### **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"

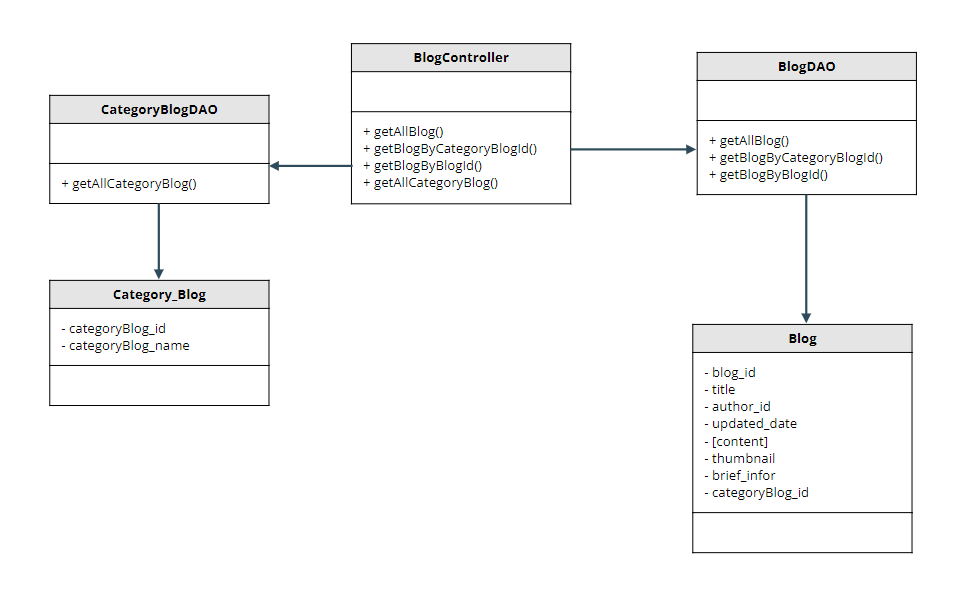
+ "where categoryBlog\_id " + categoryId + " and status = 1 and title like N'%" + searchKey + "%'\n"

+ " 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**

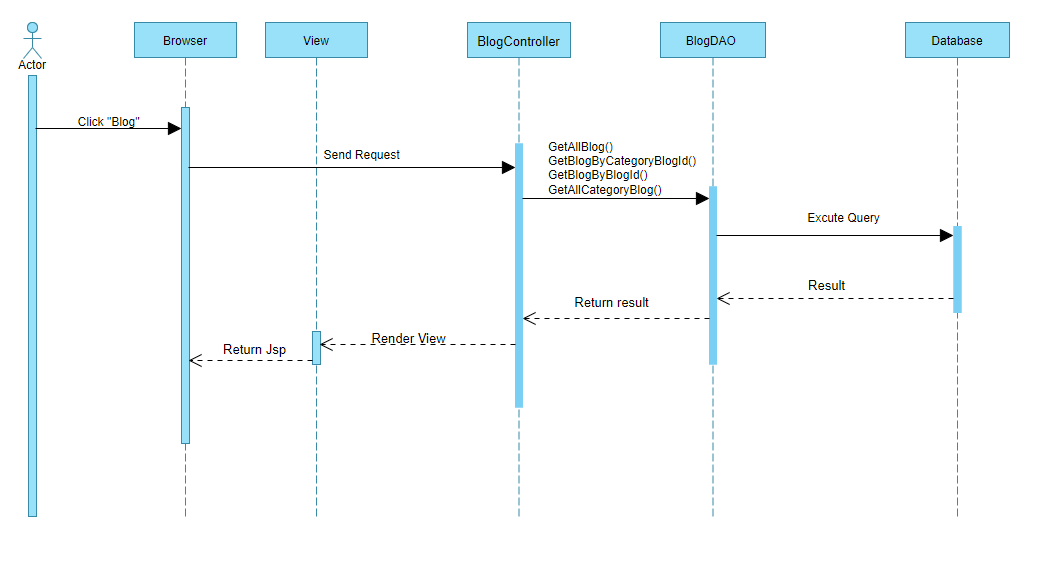
#### **XYZ Class**

#### 

|  |  |  |
| --- | --- | --- |
| **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()

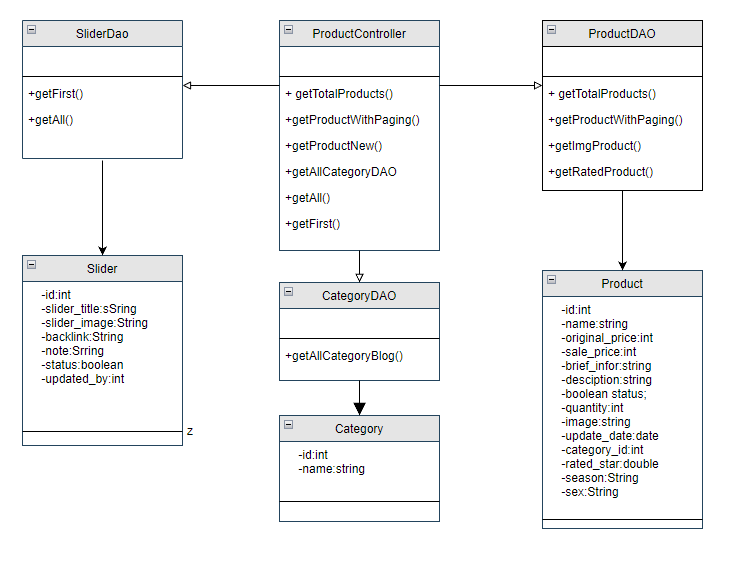
Select \* from Blog where categoryBlog\_id = ?

Select \* from Blog where blog\_id = ?

Select \* from Category\_Blog and status = 1

## **4. Public/Products List**

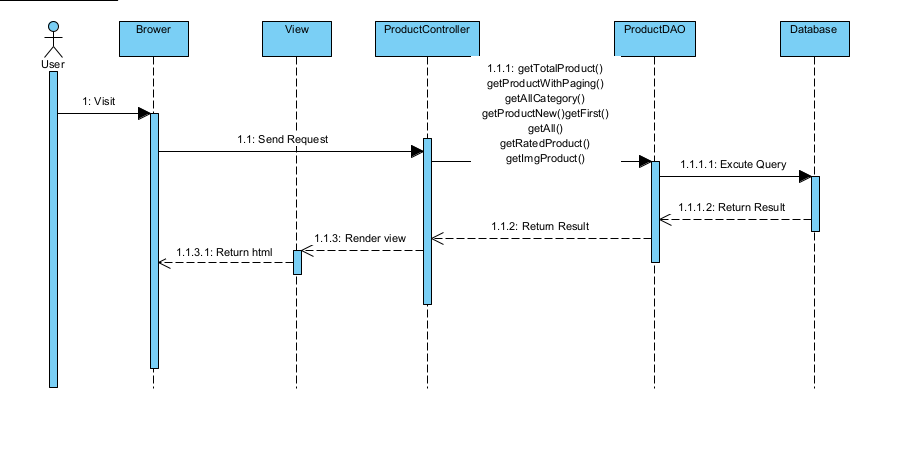
### **a. Class Diagram**



### **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**

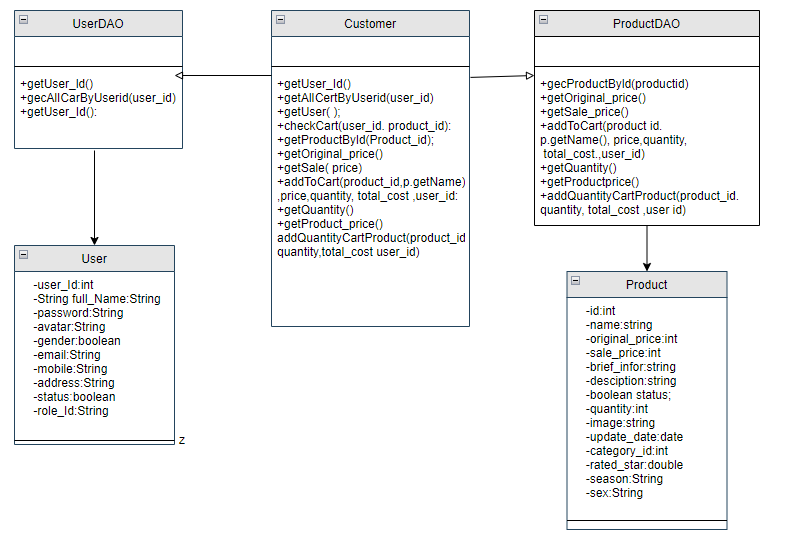
|  |  |  |
| --- | --- | --- |
| **No** | **Method** | **Description** |
| 01 |  |  |
| 02 |  |  |
| 03 |  |  |
| 04 |  |  |

### **c. Sequence Diagram(s)**

### **d. Database queries**

## **6. Public/Cart Details**

### **a. Class Diagram**

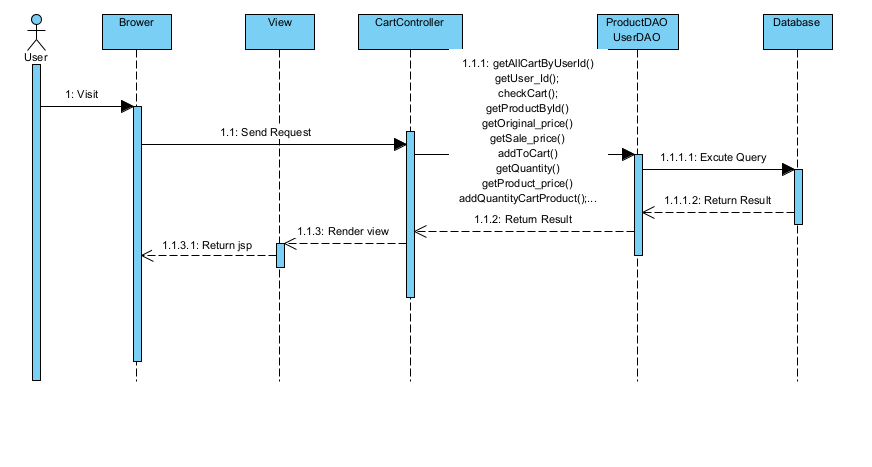
****

### **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**

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

+ " ([product\_id]\n"

+ " ,[product\_name]\n"

+ " ,[product\_price]\n"

+ " ,[quantity]\n"

+ " ,[total\_cost]\n"

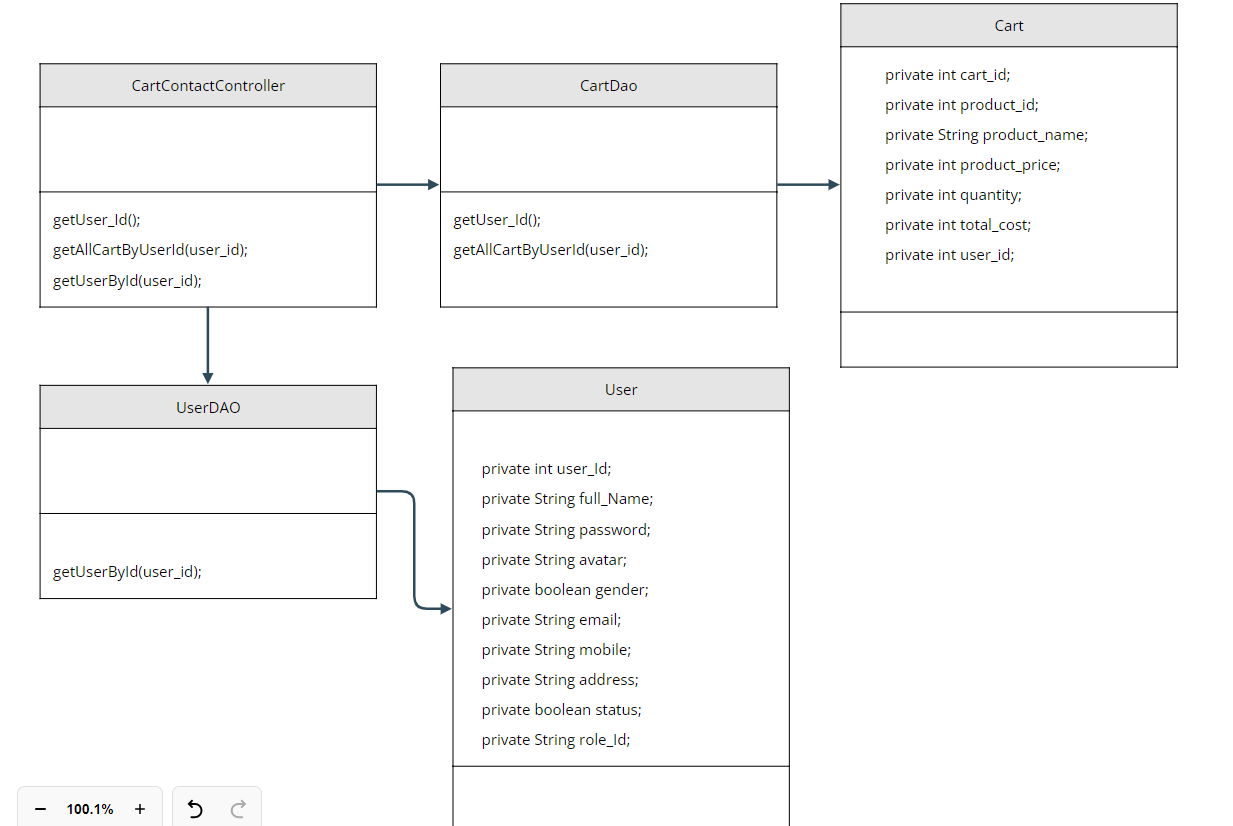
+ " ,[user\_id])\n"

+ " VALUES\n"

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

## **7. Public/Cart Contact**

### **a. Class Diagram**

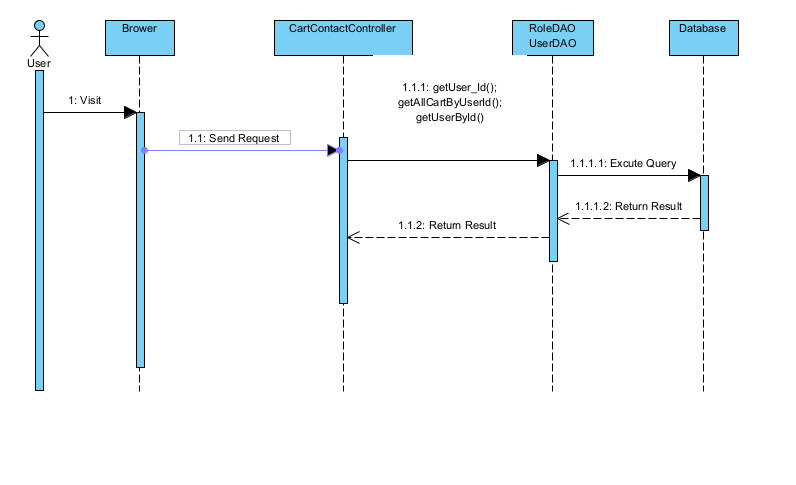


### **b. Class Specifications**

#### **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)**



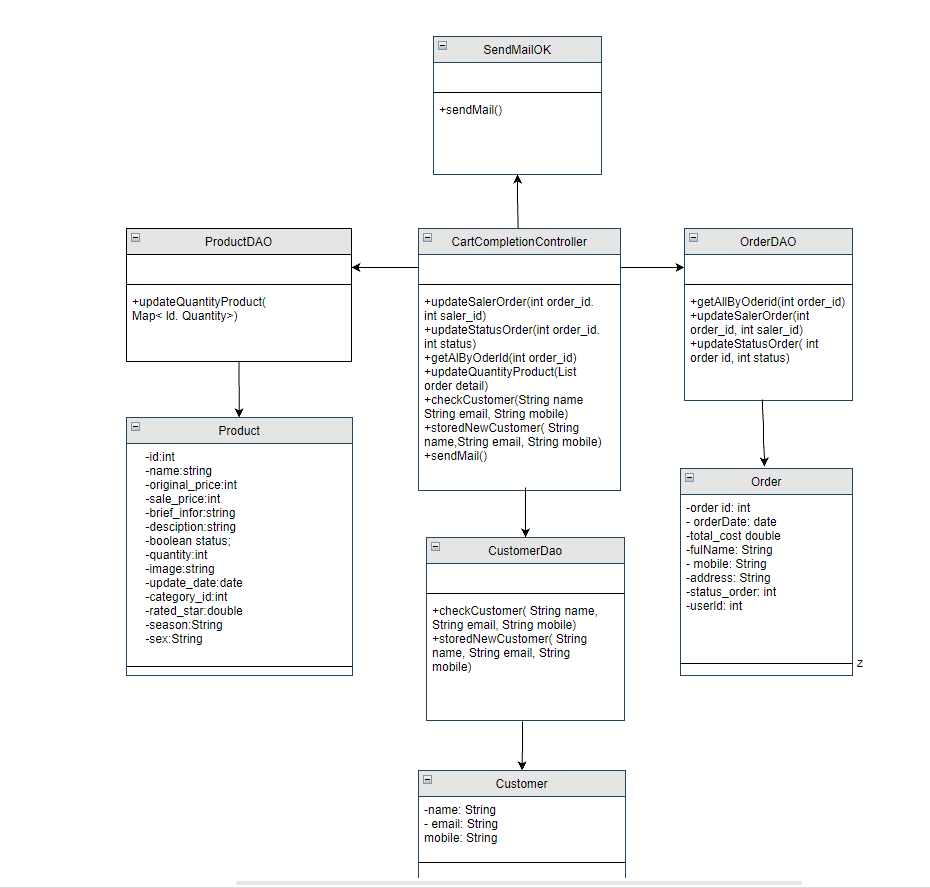
### **d. Database queries**

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

select \* from [User] where userId = ?

## **8. Public/Cart Completion**

### **a. Class Diagram**

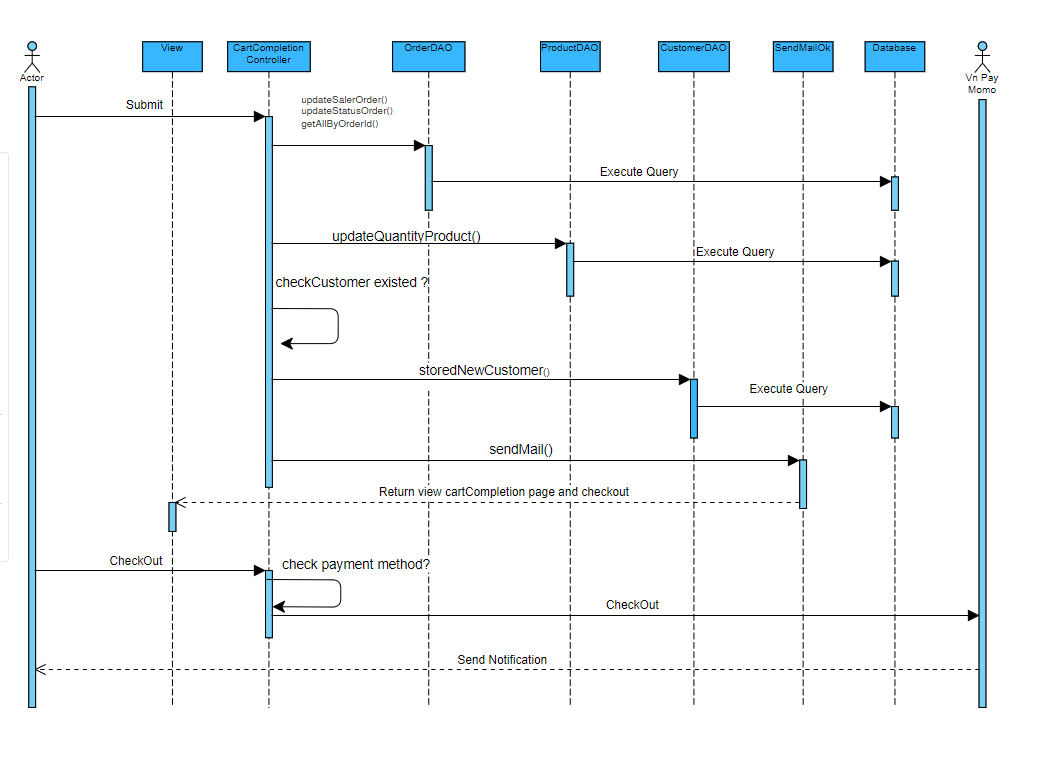


### **b. Class Specifications**

#### **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 = ?

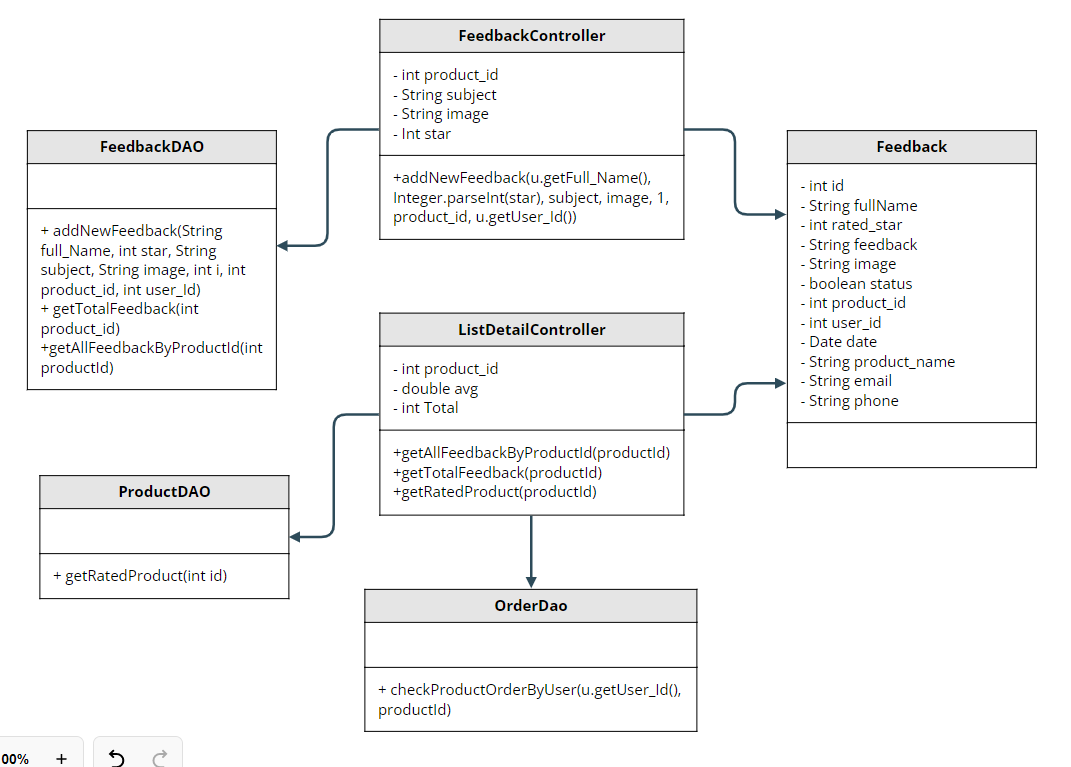
UPDATE [dbo].[Product]  SET [quantity] = (quantity - ? )  WHERE product\_id = ?

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**

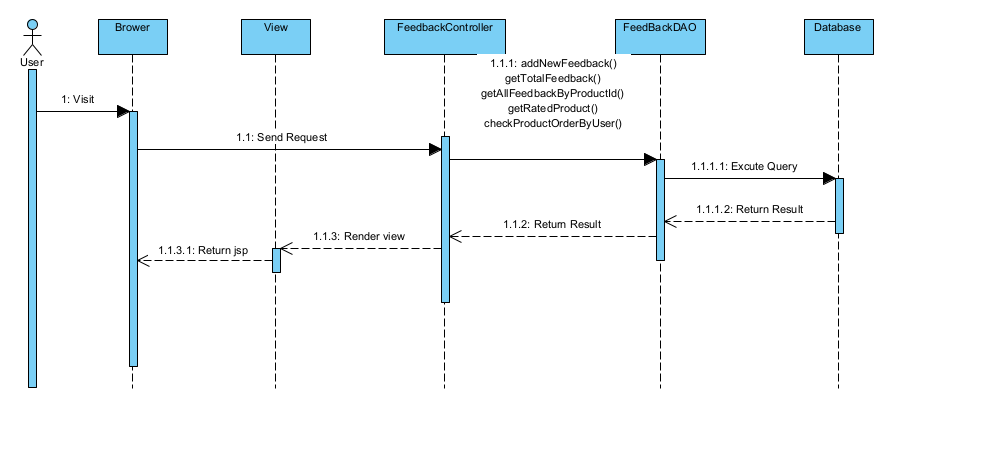


### **b. Class Specifications**

#### **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 = ?

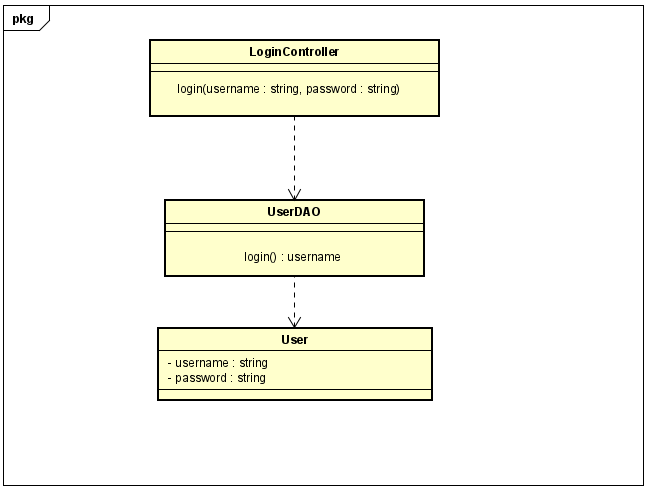
-checkProductOrderByUser(int user\_Id, int productId)

-getAllFeedbackByProductId(int productId)

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

## **10. Common/Login**

### **a. Class Diagram**

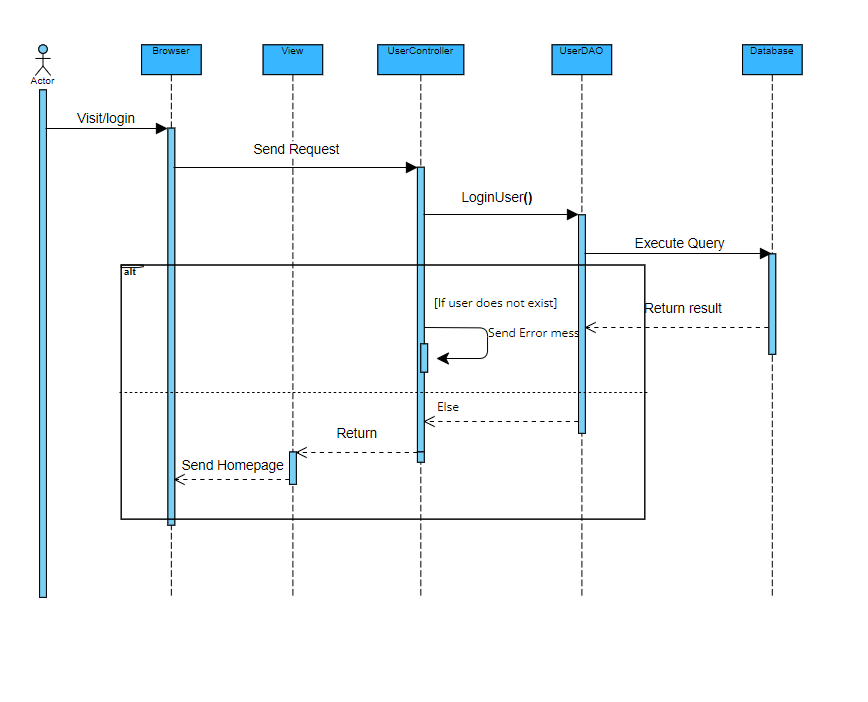


### **b. Class Specifications**

#### 

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

### **c. Sequence Diagram(s)**

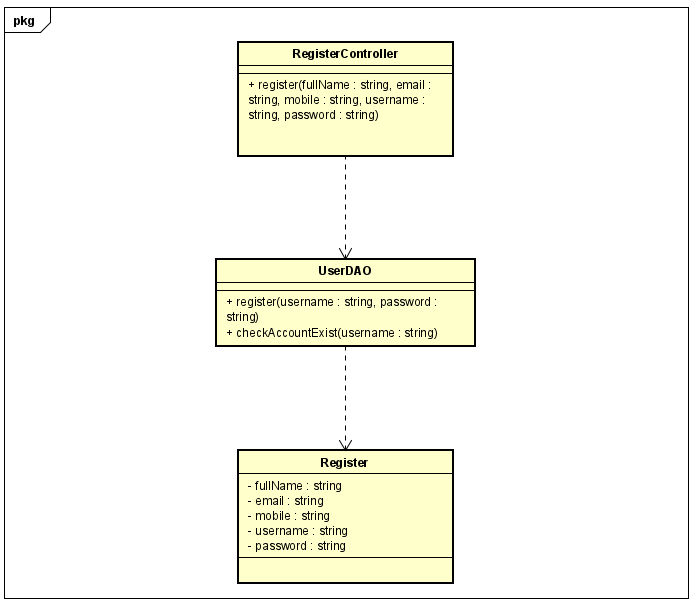


### **d. Database queries**

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

## **11 Common/Register**

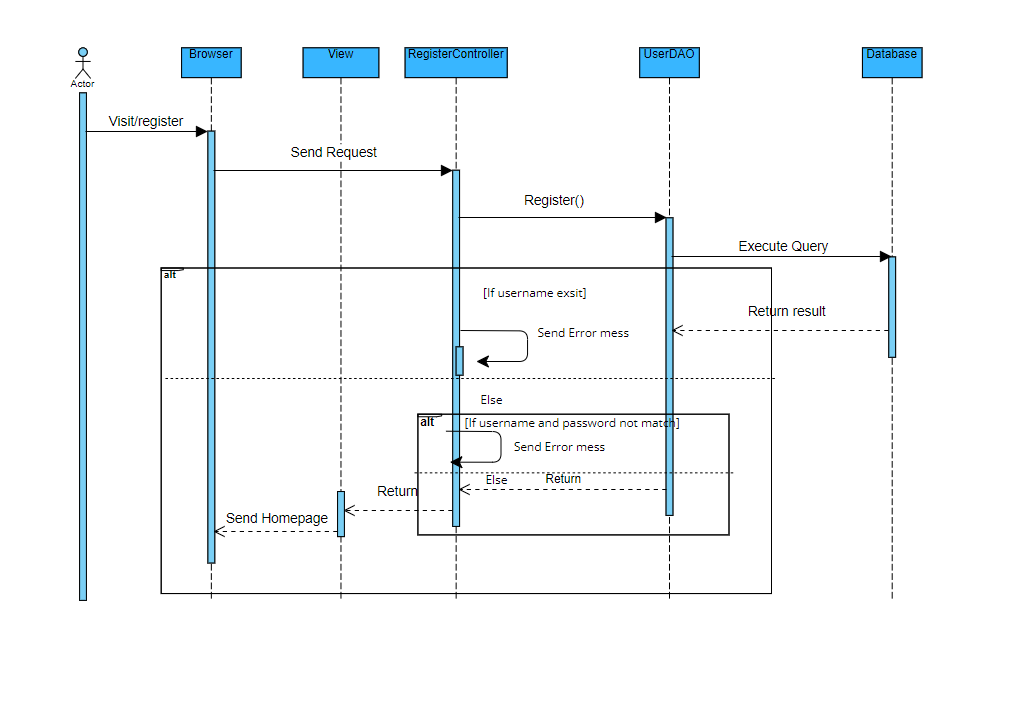
### **a. Class Diagram**



### **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 (?, ?, ?, ?, ?)

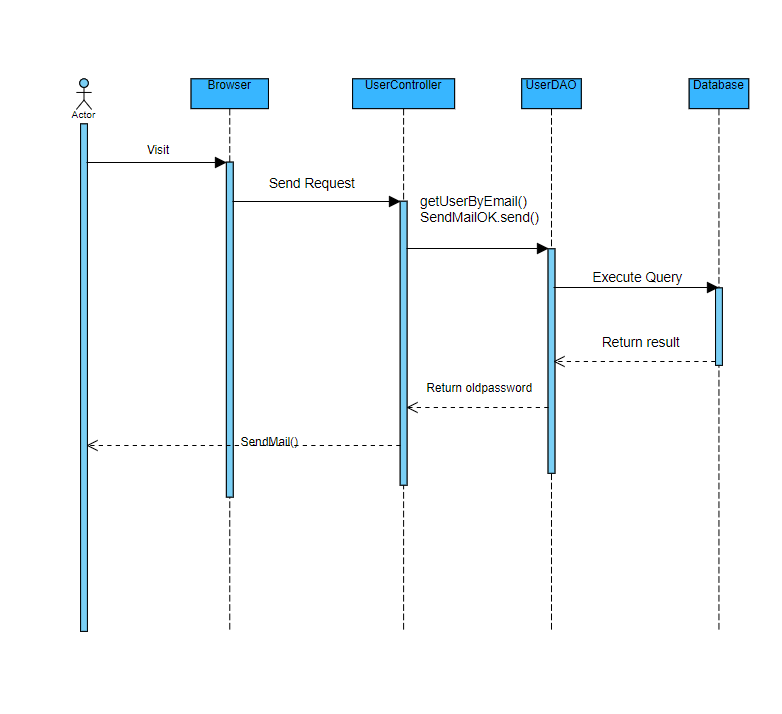
## **12. Common/Reset Password**

### **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)**

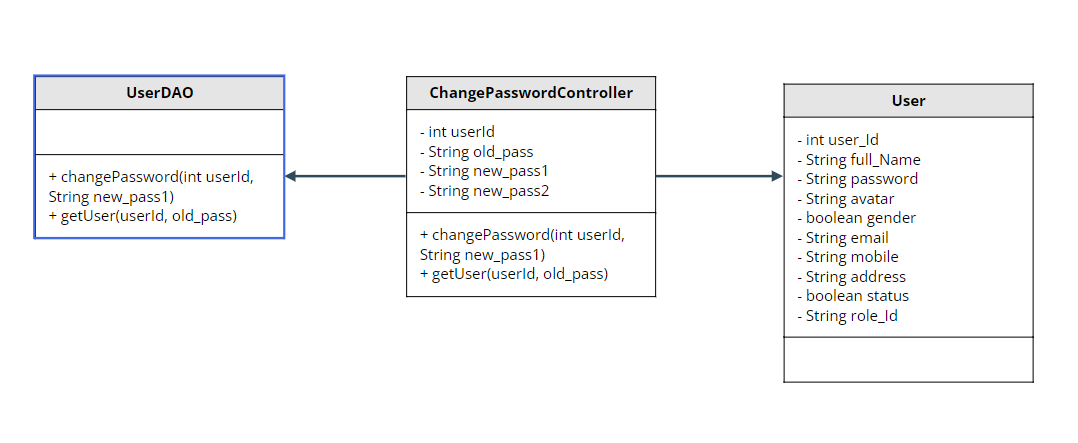


### **d. Database queries**

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

## **13. Common/Change Password**

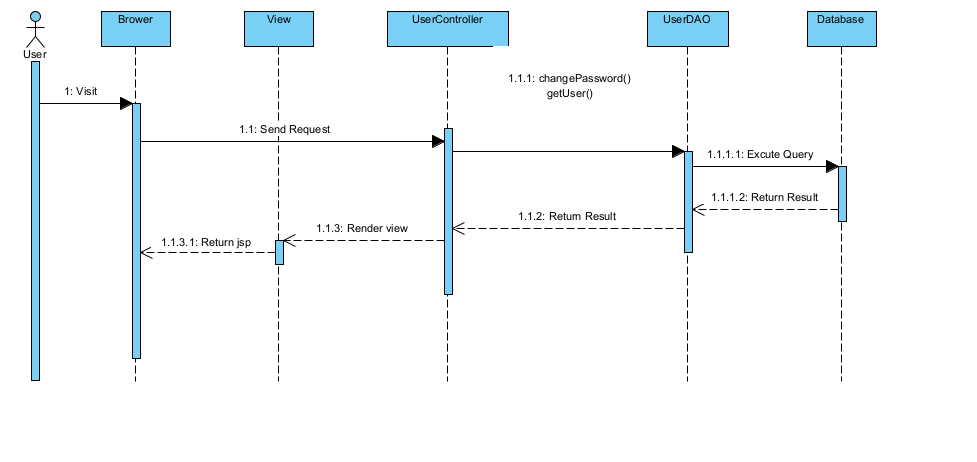
### **a. Class Diagram**



### **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"

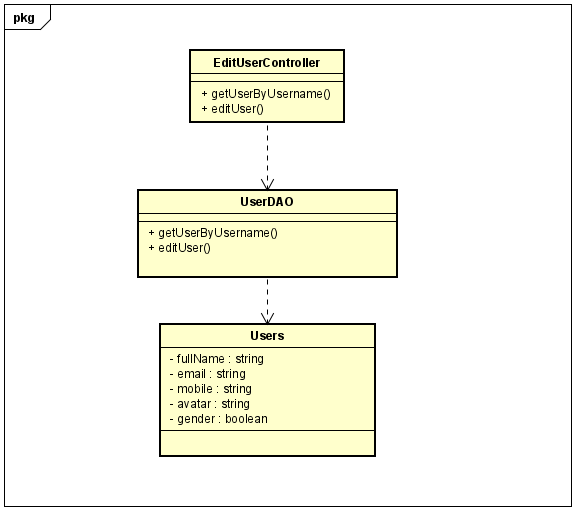
+ " SET \n"

+ " [password] = ?\n"

+ " WHERE userId = ?\n

## **14. Common/User Profile**

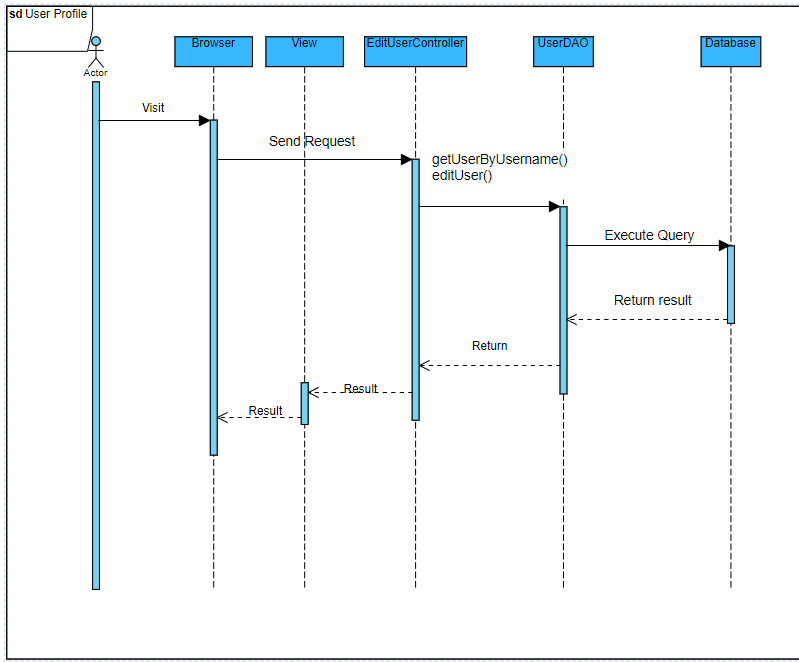
### **a. Class Diagram**



### **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

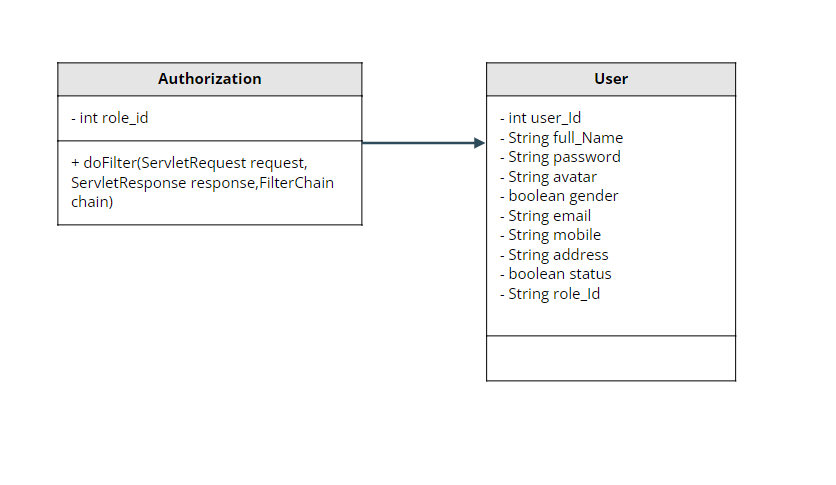
* UPDATE users

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

WHERE username = ?

## **15. Common/User Authorization**

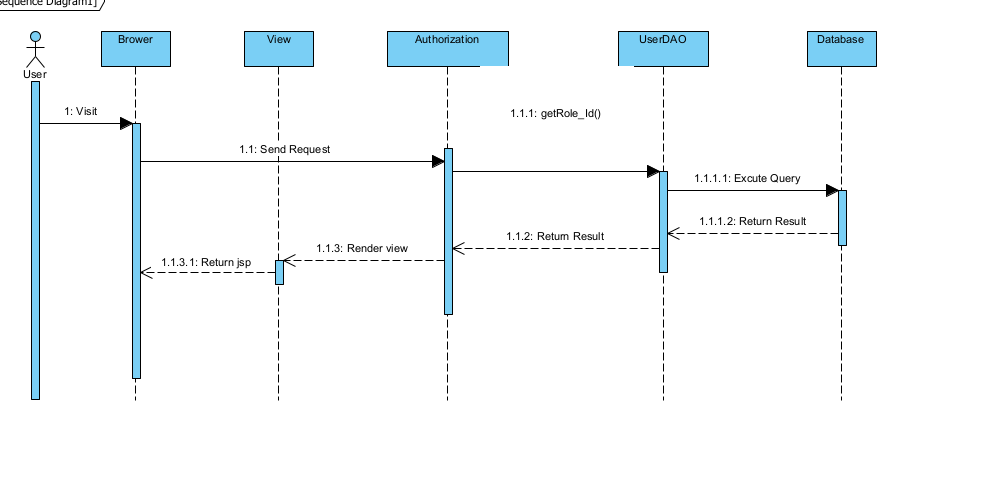
### **a. Class Diagram**



### **b. Class Specifications**

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

### **c. Sequence Diagram(s)**



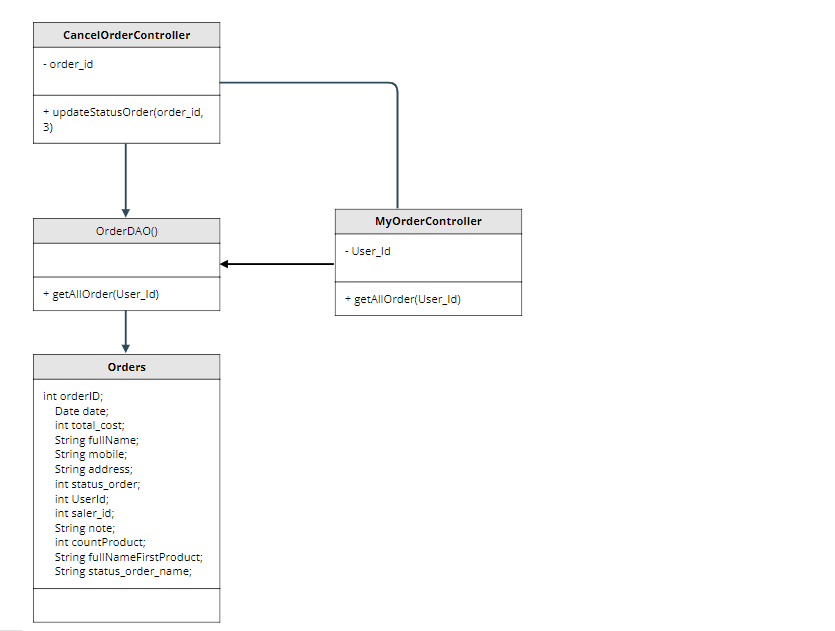
### **d. Database queries**

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

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

## **16. Customer/My Orders**

### **a. Class Diagram**

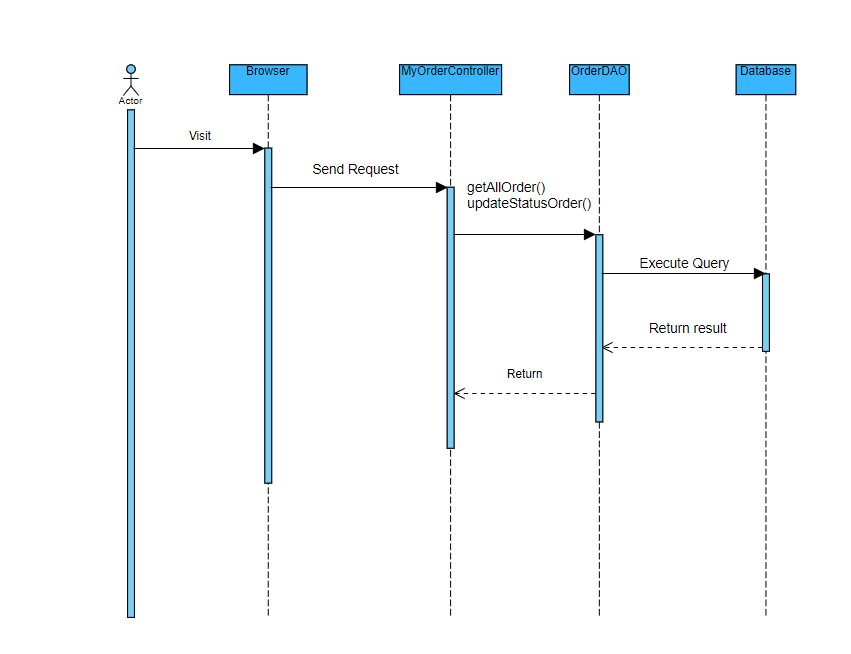


### **b. Class Specifications**

#### **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 = ?

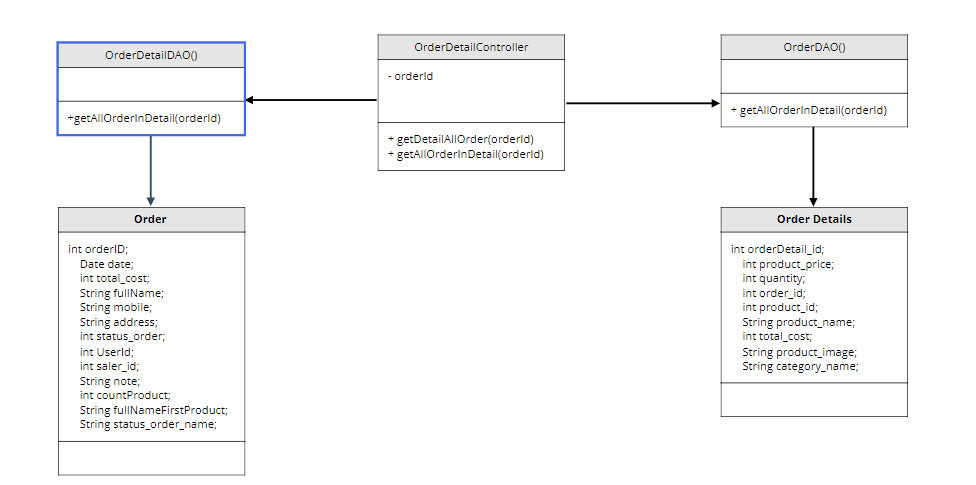
select \* from [Order] join Status\_Order\n"

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

+ "where userId = ?

## **17. Customer/Order Information**

### **a. Class Diagram**

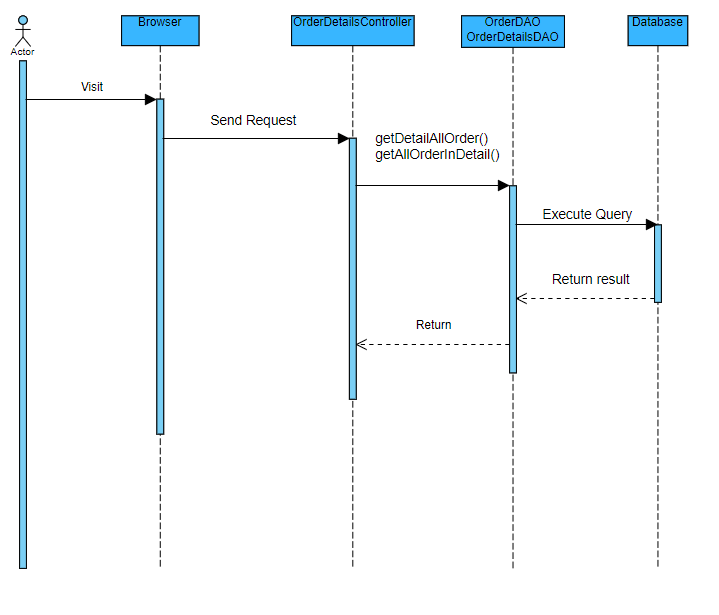


### **b. Class Specifications**

#### **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"

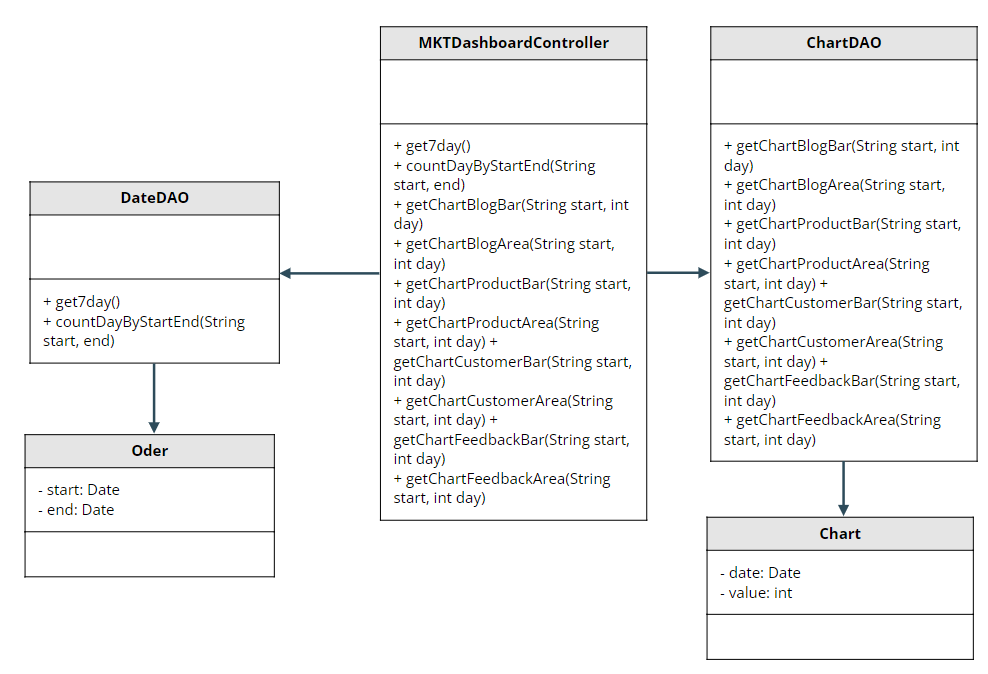
+ " join Products\_Images on Order\_Detail.product\_id = Products\_Images.product\_id\n"

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

+ " where order\_id = ?

## **18.Staff/MKT Dashboard**

### **a. Class Diagram**

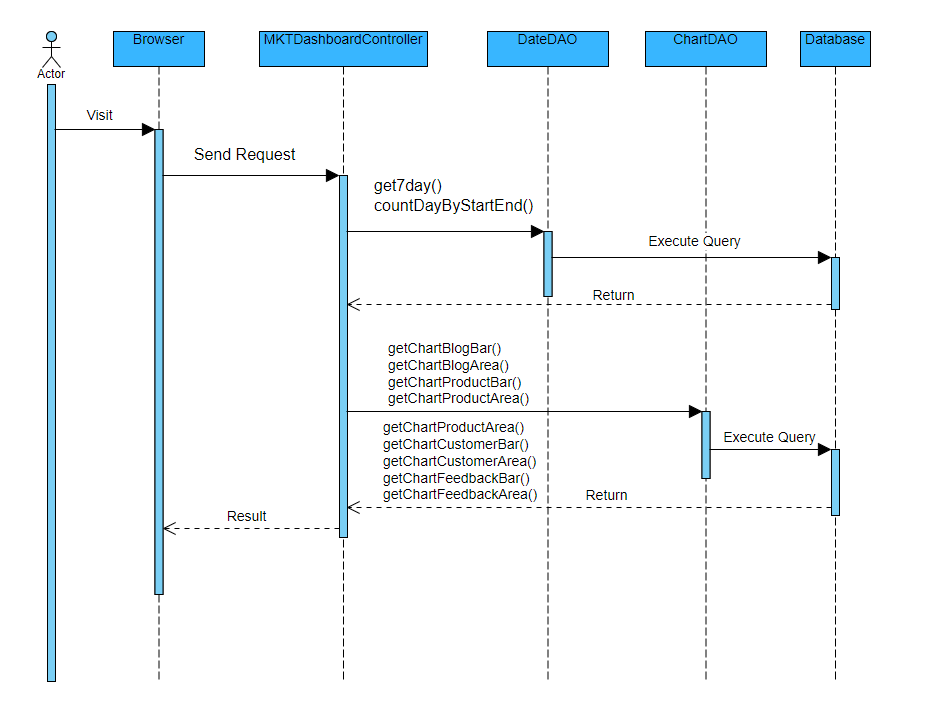


### **b. Class Specifications**

#### **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)**

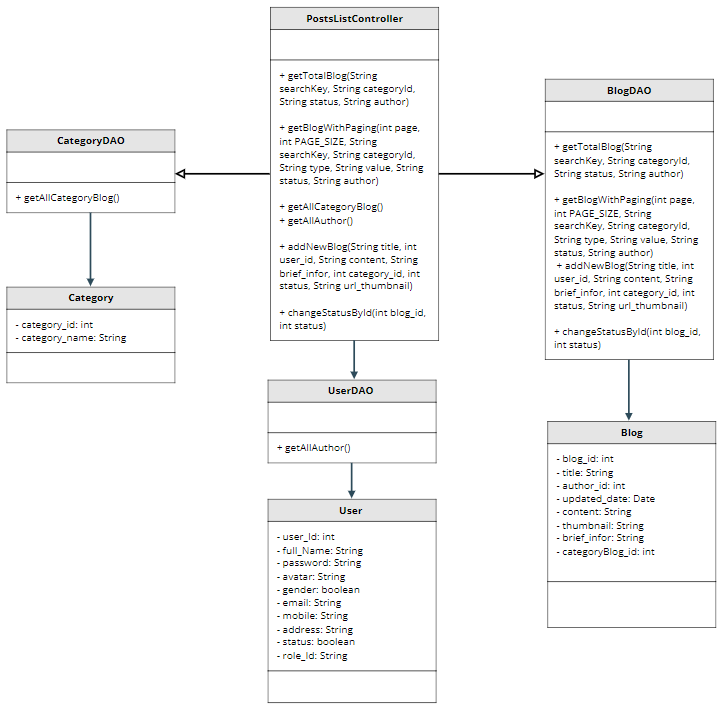
****

### **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.Staff/Posts List**

### **a. Class Diagram**

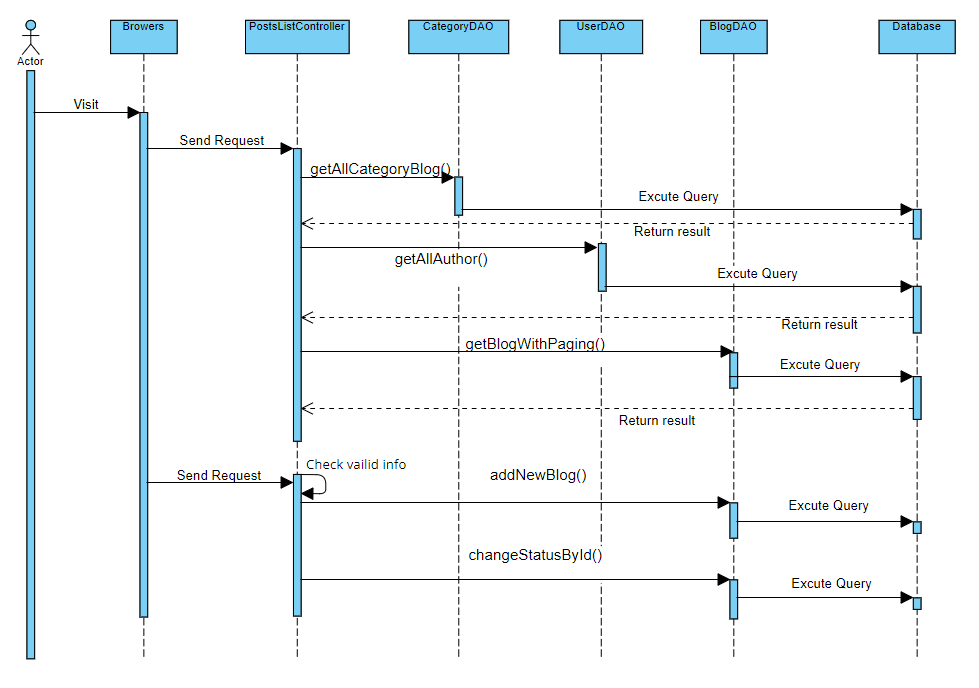


### **b. Class Specifications**

#### **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)**

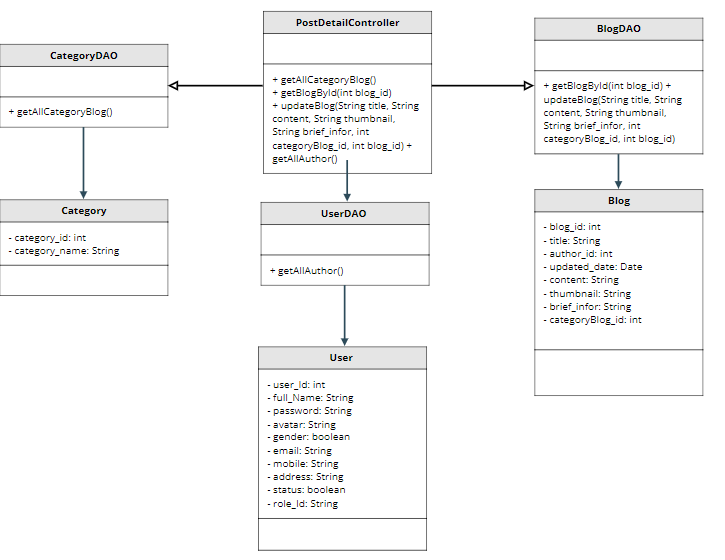
****

### **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.Staff/Post Details**

### **a. Class Diagram**

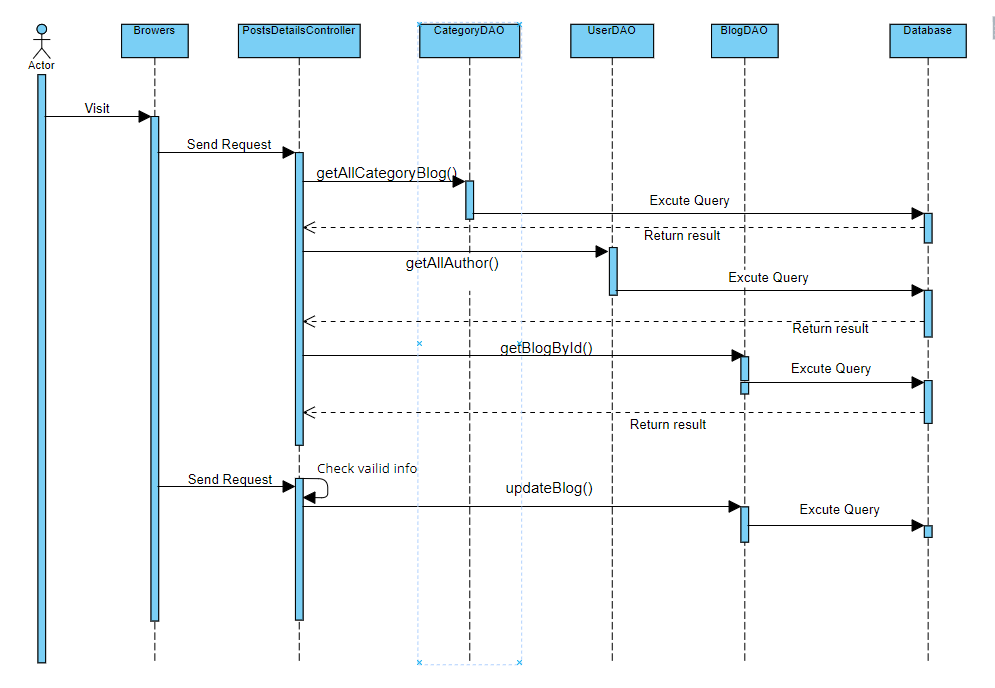


### **b. Class Specifications**

#### **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)**

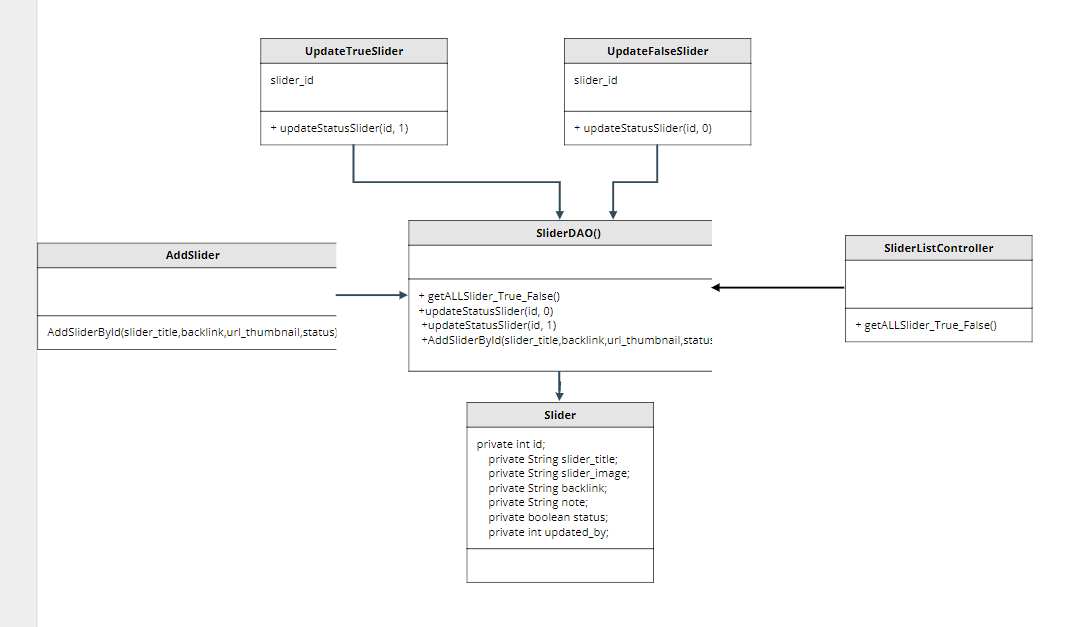


### **d. Database queries**

* 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.Staff/Sliders List**

### **a. Class Diagram**

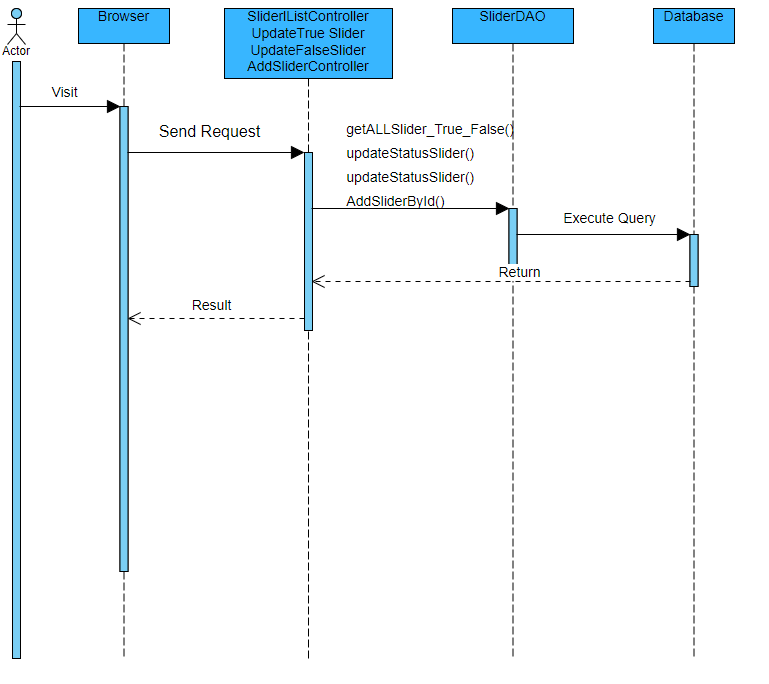


### **b. Class Specifications**

#### **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"

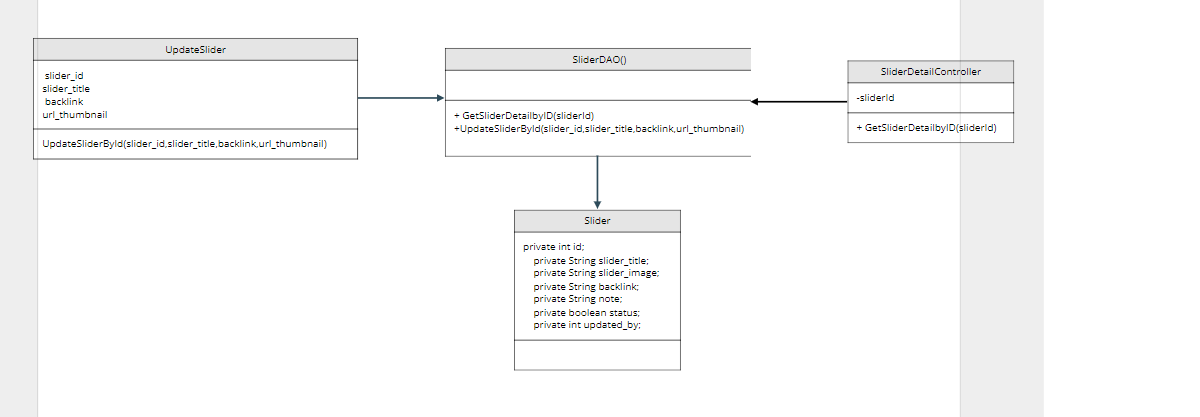
+ " SET status = ?\n"

+ " WHERE slider\_id = ?

SELECT \* FROM [dbo].[Slider]

## **22.Staff/Slider Details**

### **a. Class Diagram**

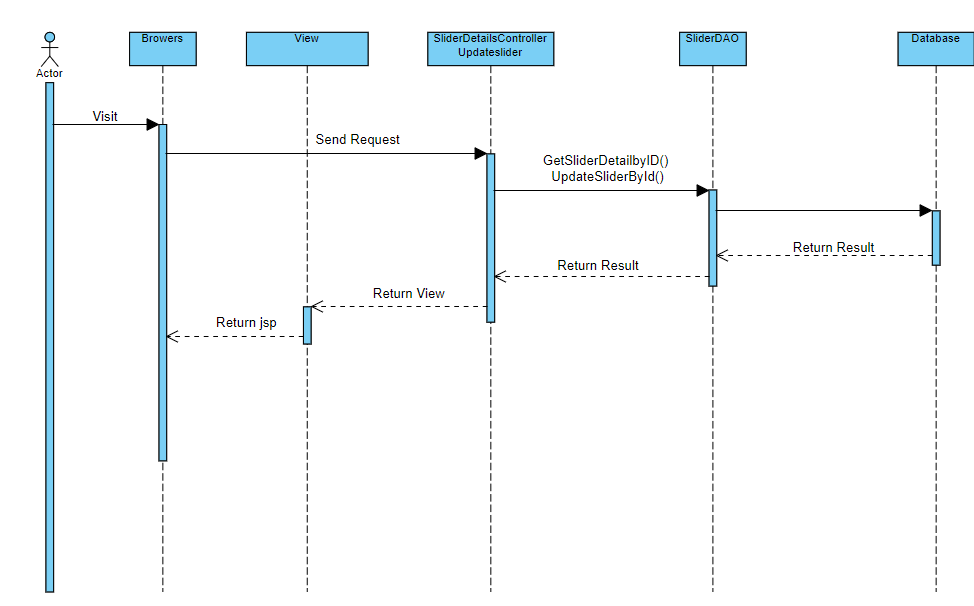


### **b. Class Specifications**

#### **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"

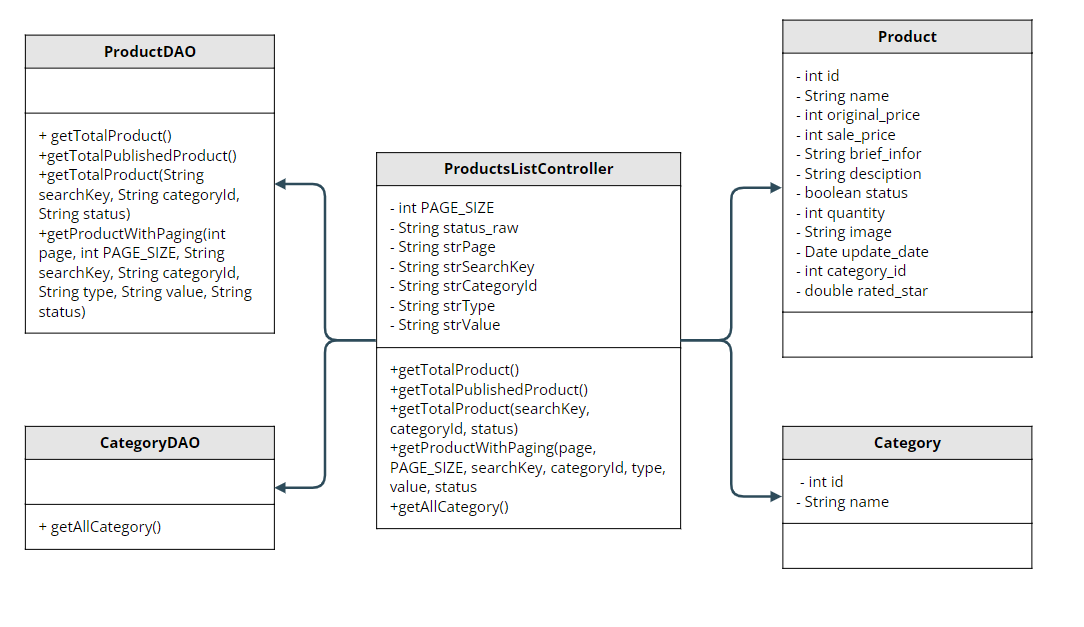
+ " WHERE slider\_id = ?

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

## **23.Staff/Products List**

### **a. Class Diagram**

//\*

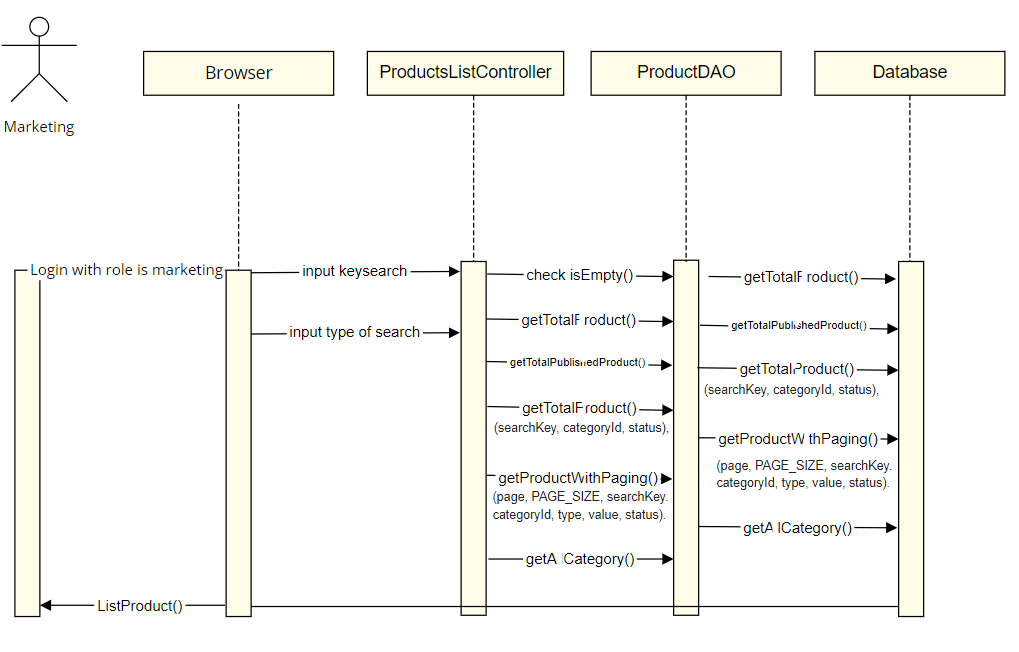


### **b. Class Specifications**

#### 

|  |  |  |
| --- | --- | --- |
| **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.Staff/Product Details**

### **a. Class Diagram**

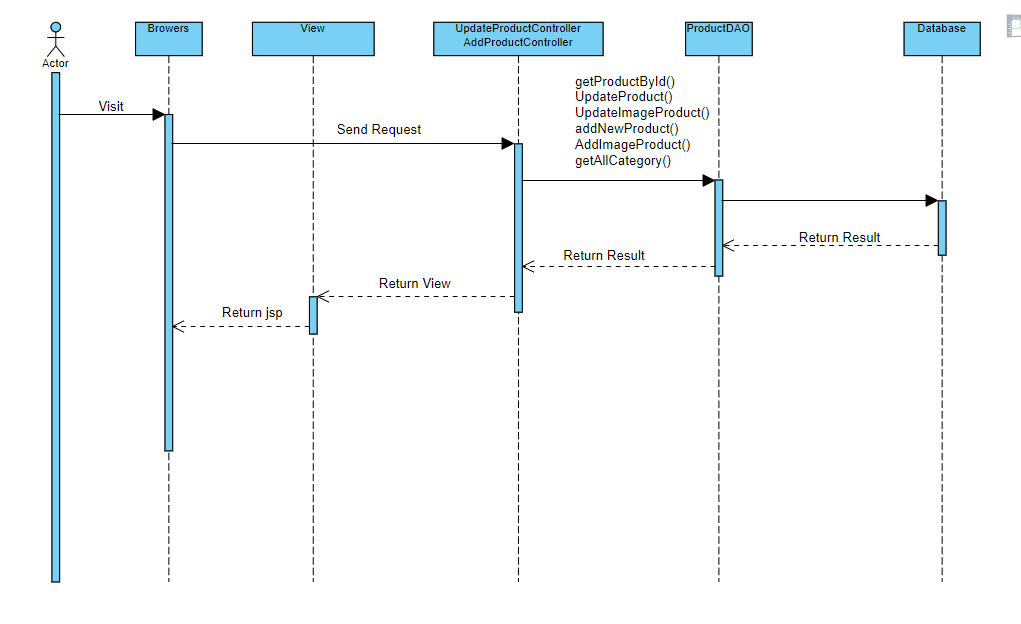
//\*

### **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 Product (product\_name, original\_prices, sale\_prices, product\_details, brief\_infor, status, quantity, category\_id, update\_date, season, sex) values(?,?,?,?,?,?,?,?,?,?,?);

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

+ " ([product\_id]\n"

+ " ,[images])\n"

+ " VALUES\n"

+ " (?,?)

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

-Select \* from Category

-update Product set product\_name = ?, original\_prices = ?, sale\_prices = ?, product\_details = ?, brief\_infor = ?, status = ?, "

+ "quantity = ?, category\_id = ?, update\_date = ?, season = ?, sex = ? where product\_id = ?

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

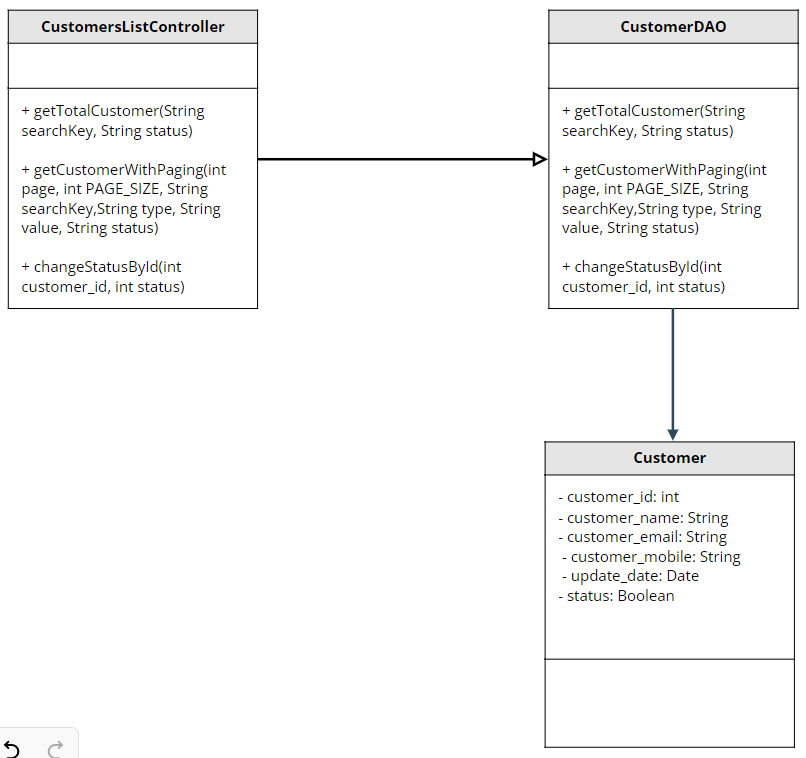
+ " SET\n"

+ " [images] = ?\n"

+ " WHERE product\_id = ?

## **25.Staff/Customers List**

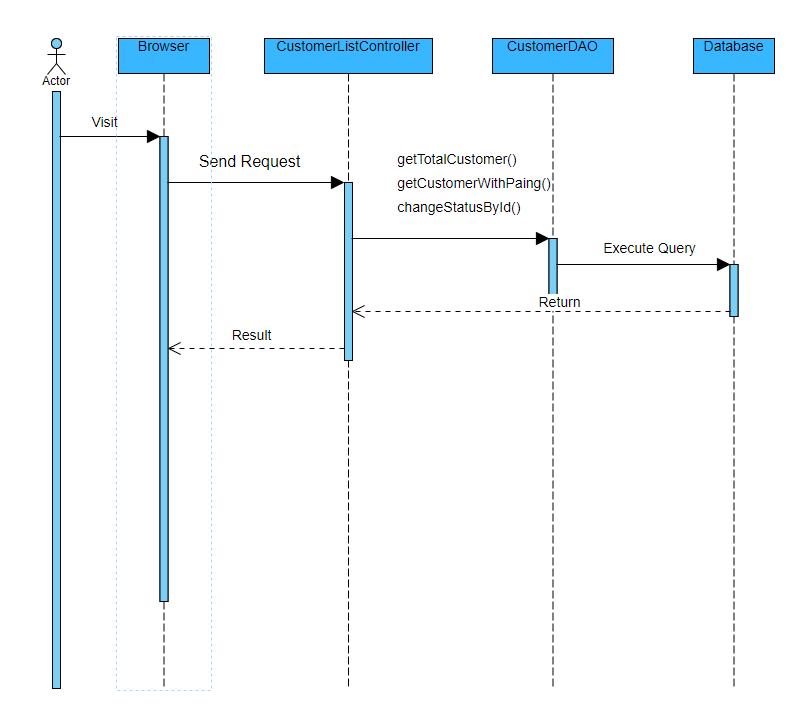
### **a. Class Diagram**

****

### **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 |

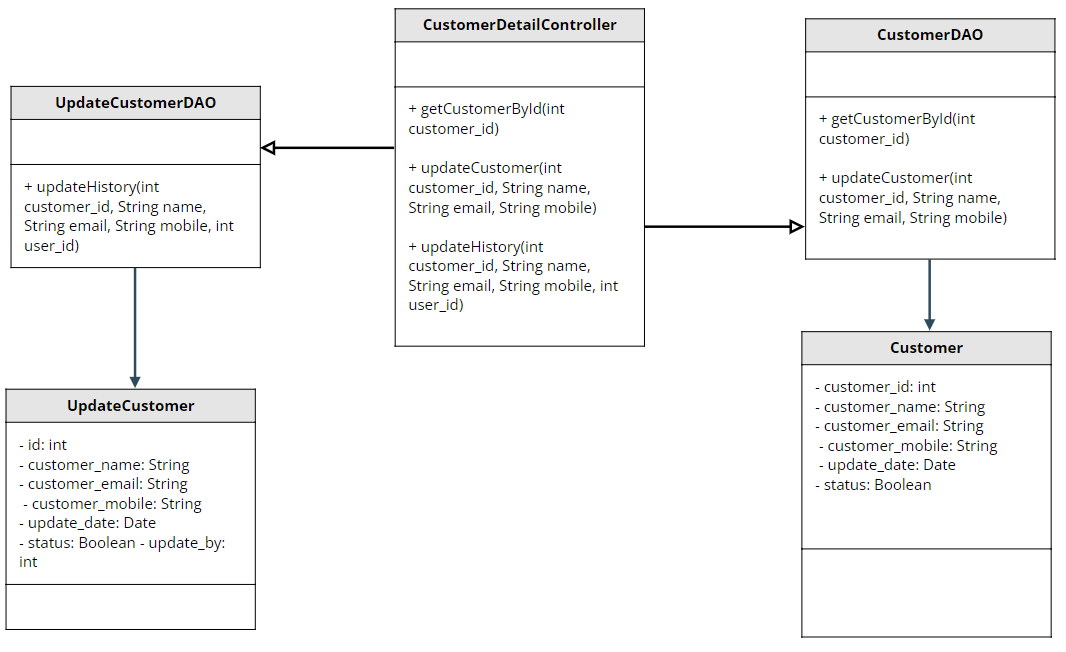
### **c. Sequence Diagram(s)**



### **d. Database queries**

## **26.Staff/Customer Details**

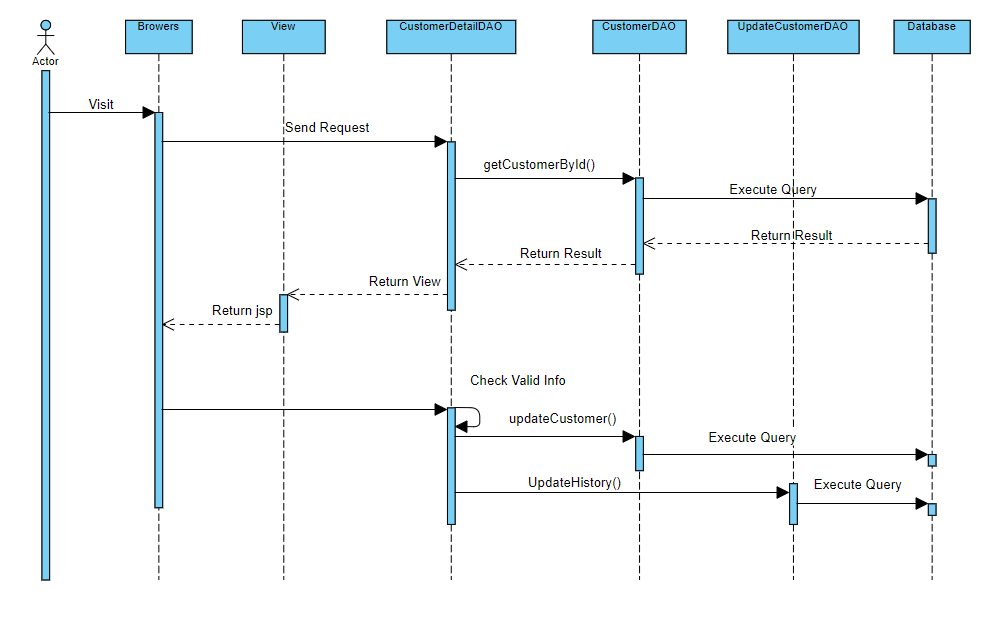
### **a. Class Diagram**



### **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)**

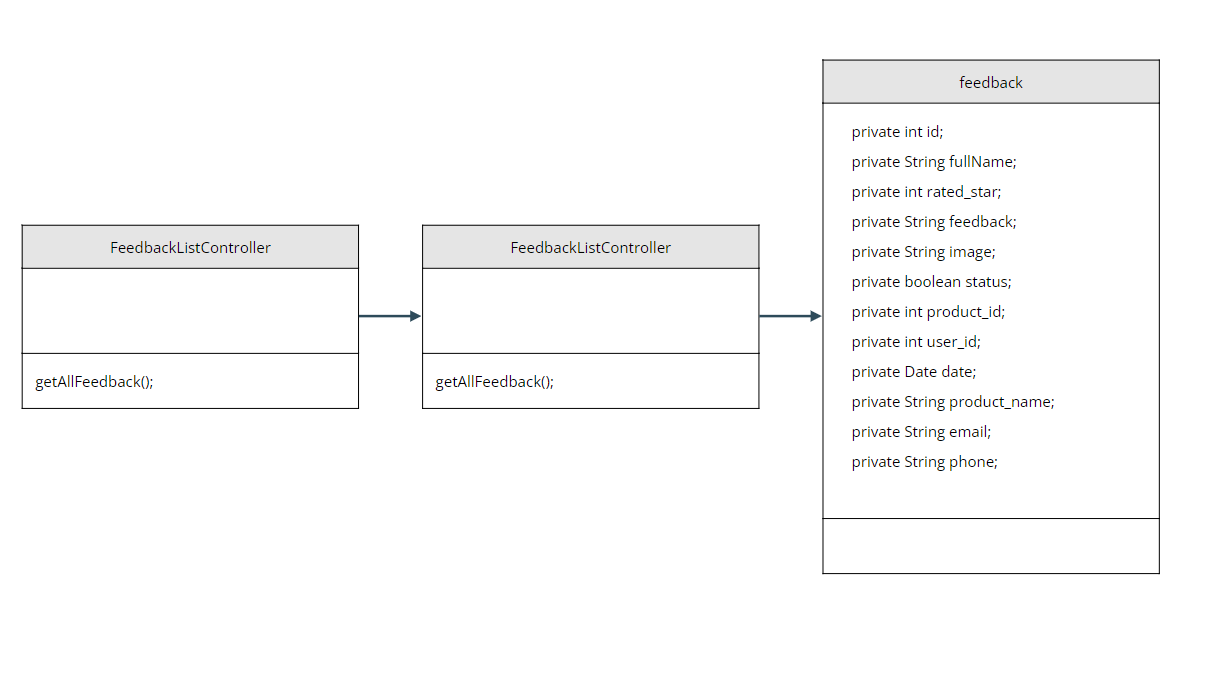


### **d. Database queries**

* 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.Staff/Feedbacks List**

### **a. Class Diagram**

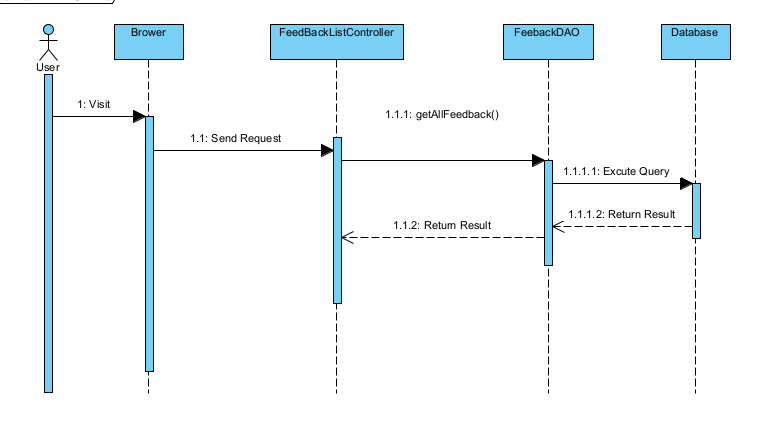


### **b. Class Specifications**

#### **XYZ Class**

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

### **c. Sequence Diagram(s)**



### **d. Database queries**

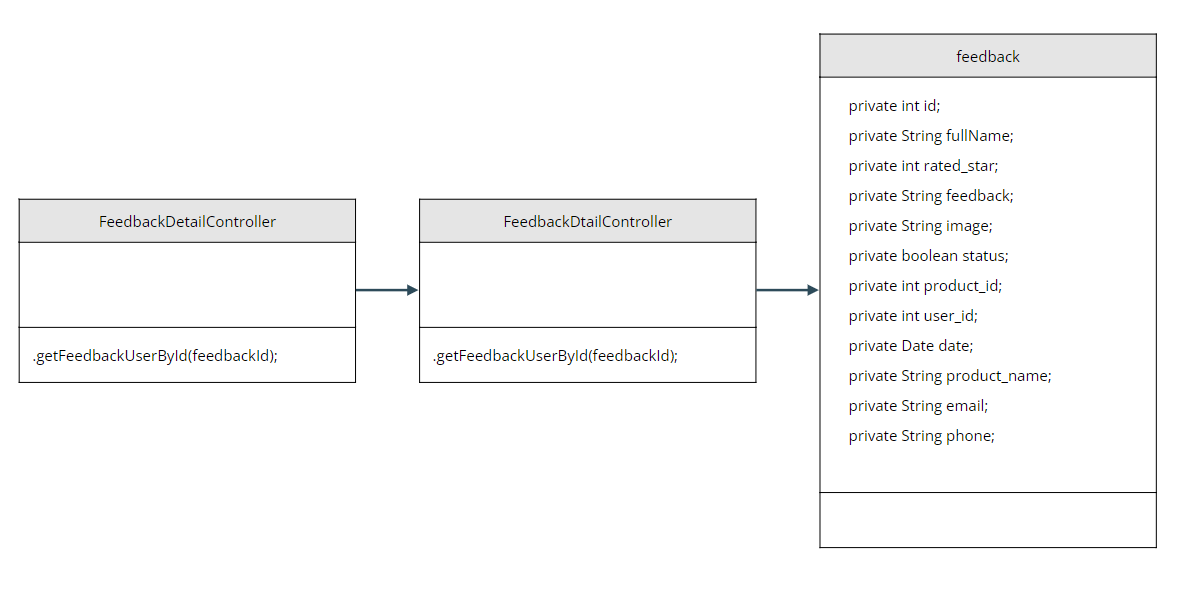
select f.\*, p.product\_name \n"

+ "from Feedback f join Product p \n"

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

## **28.Staff/Feedback Details**

### **a. Class Diagram**

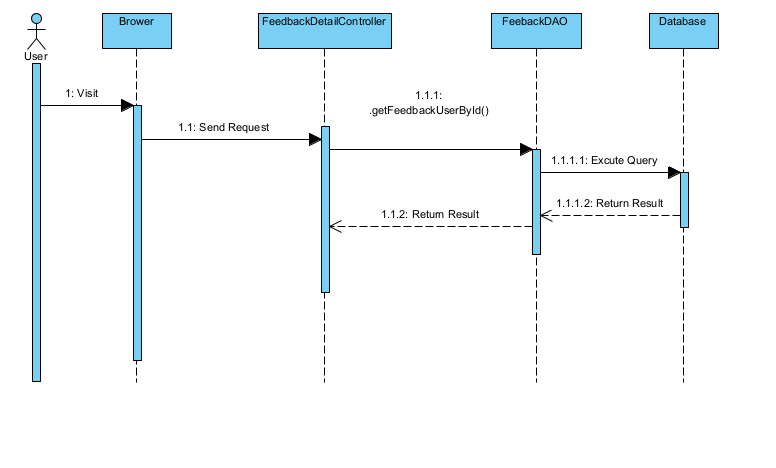


### **b. Class Specifications**

#### **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"

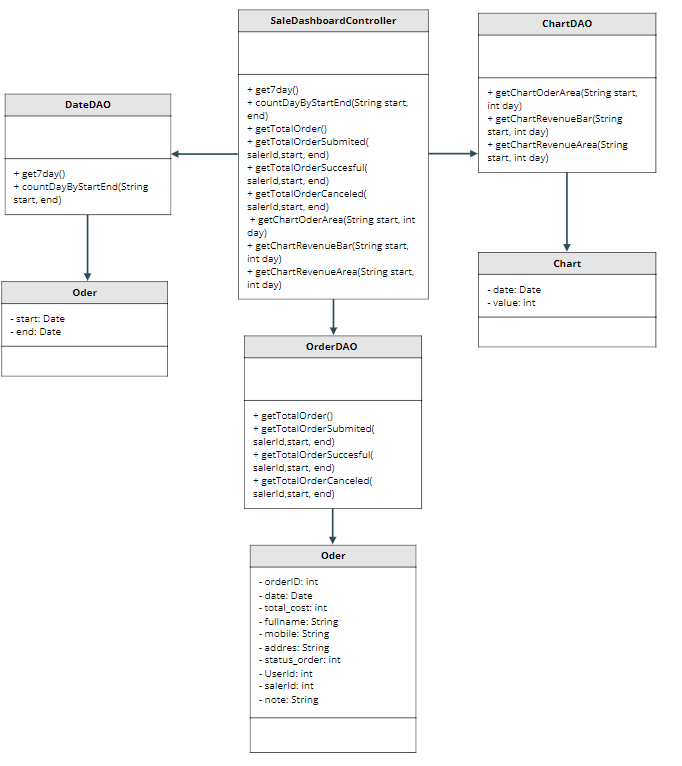
+ "join Product p on f.product\_id = p.product\_id\n"

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

+ "where f.feedBack\_id = ?"

## **29. Staff/Sale Dashboard**

### **a. Class Diagram**

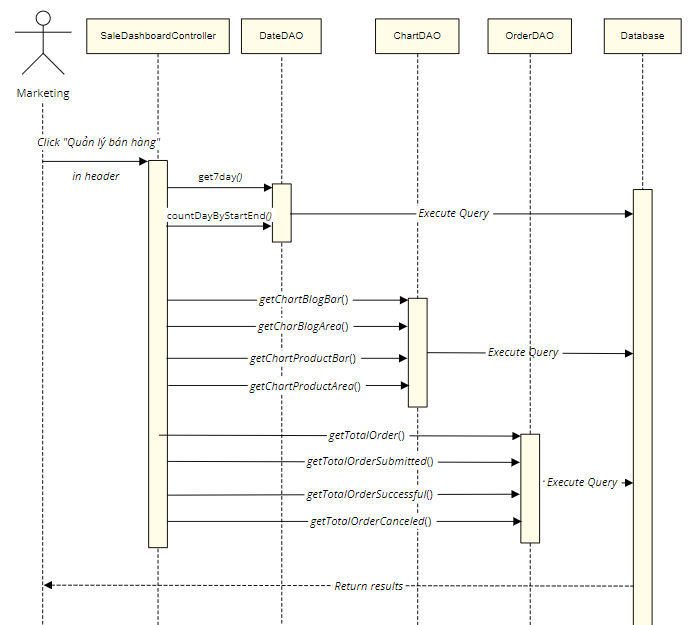


### **b. Class Specifications**

#### **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)**

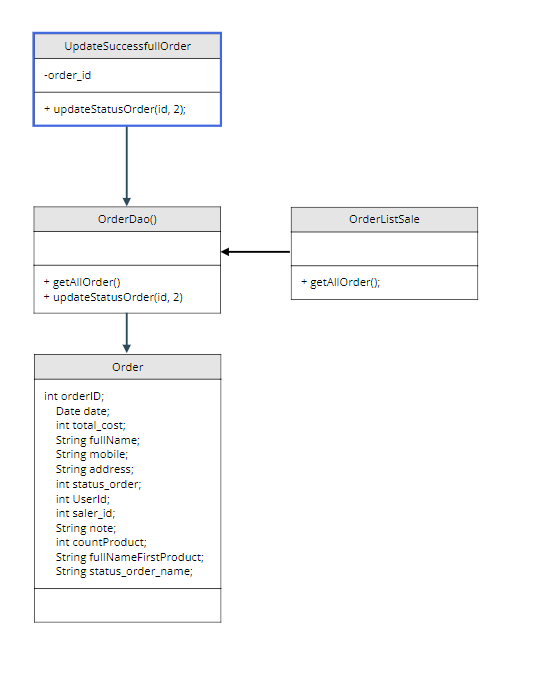
****

### **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.Staff/Orders List**

### **a. Class Diagram**

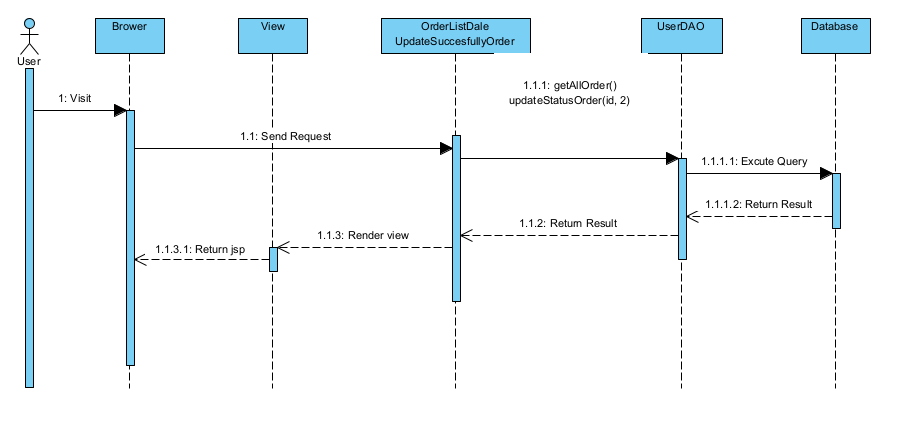


### **b. Class Specifications**

#### **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 |

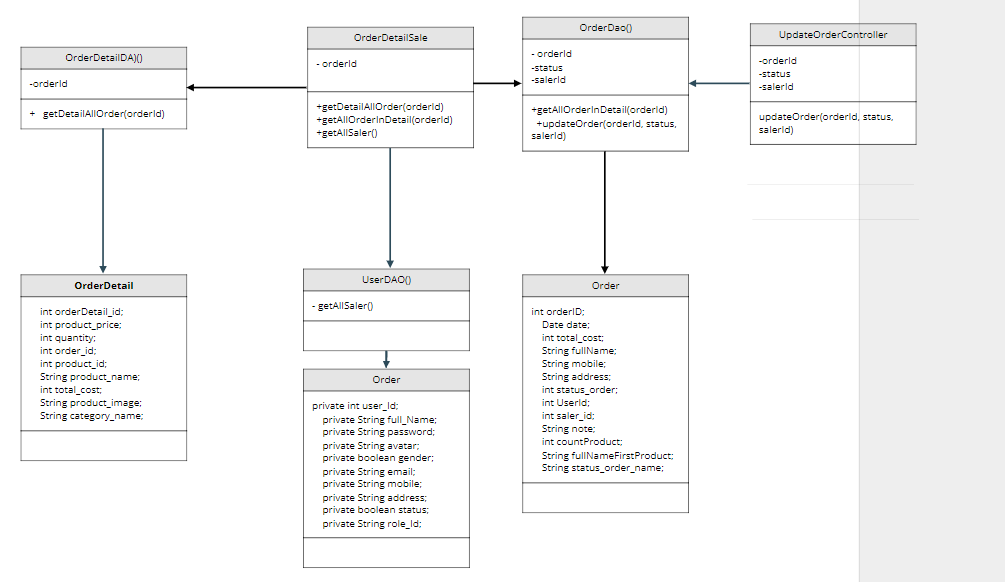
### **c. Sequence Diagram(s)**



### **d. Database queries**

## **31.Staff/Order Details**

### **a. Class Diagram**

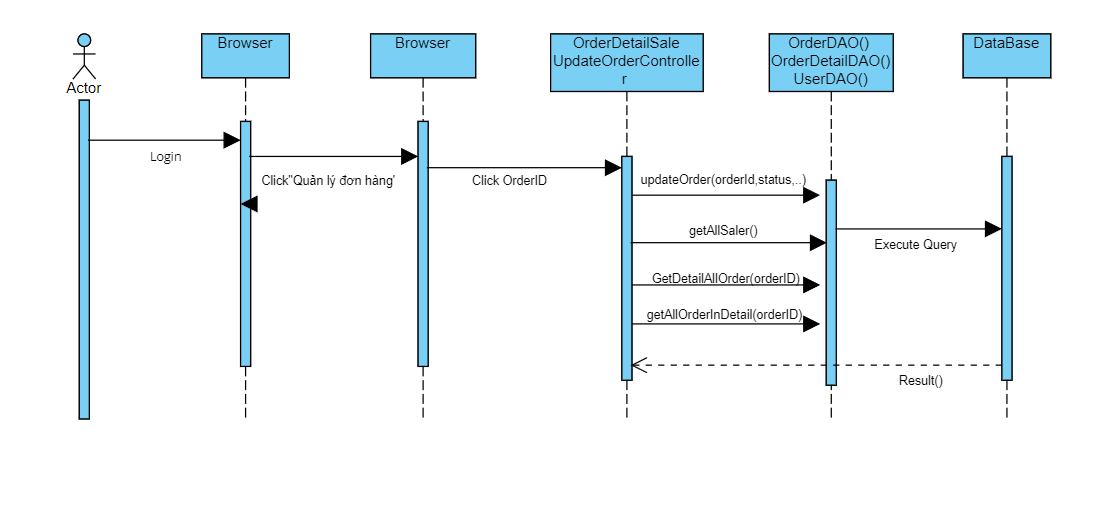


### **b. Class Specifications**

#### **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 = ?

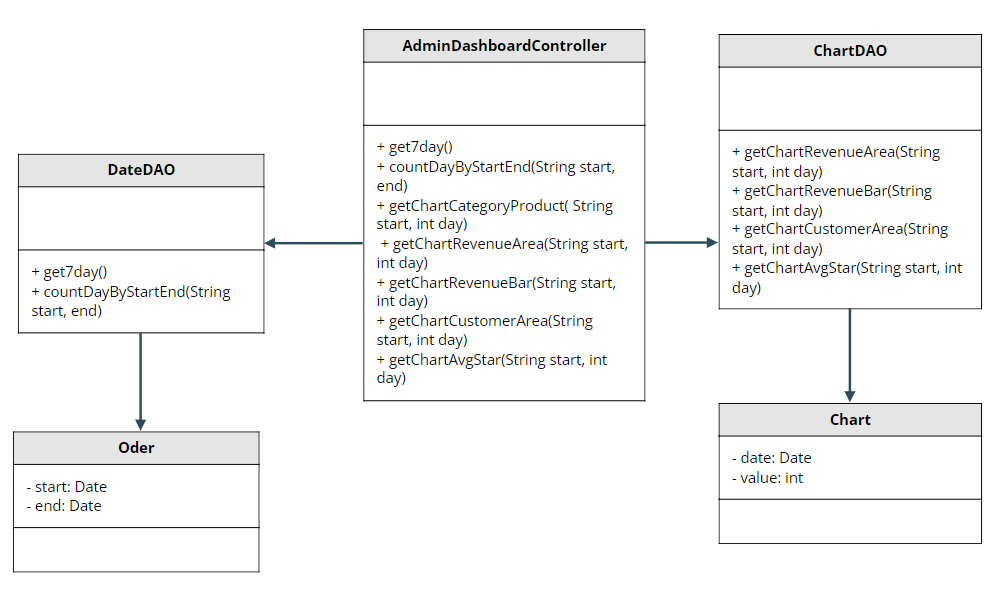
select \* from [Order] join Status\_Order\n"

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

+ " where order\_id = ?

## **32. Admin/Admin Dashboard**

### **a. Class Diagram**

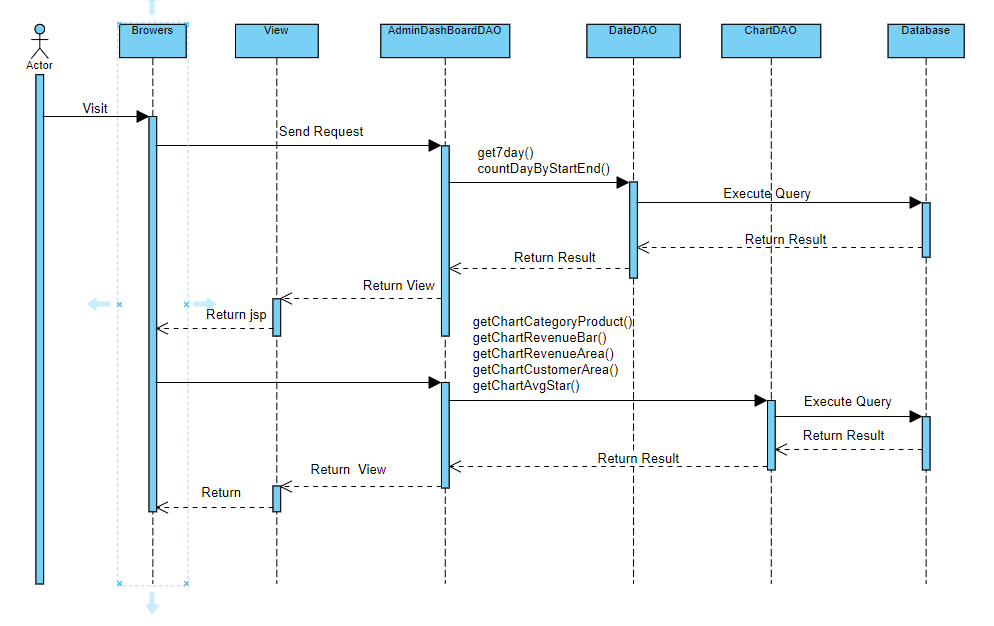


### **b. Class Specifications**

#### **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)**

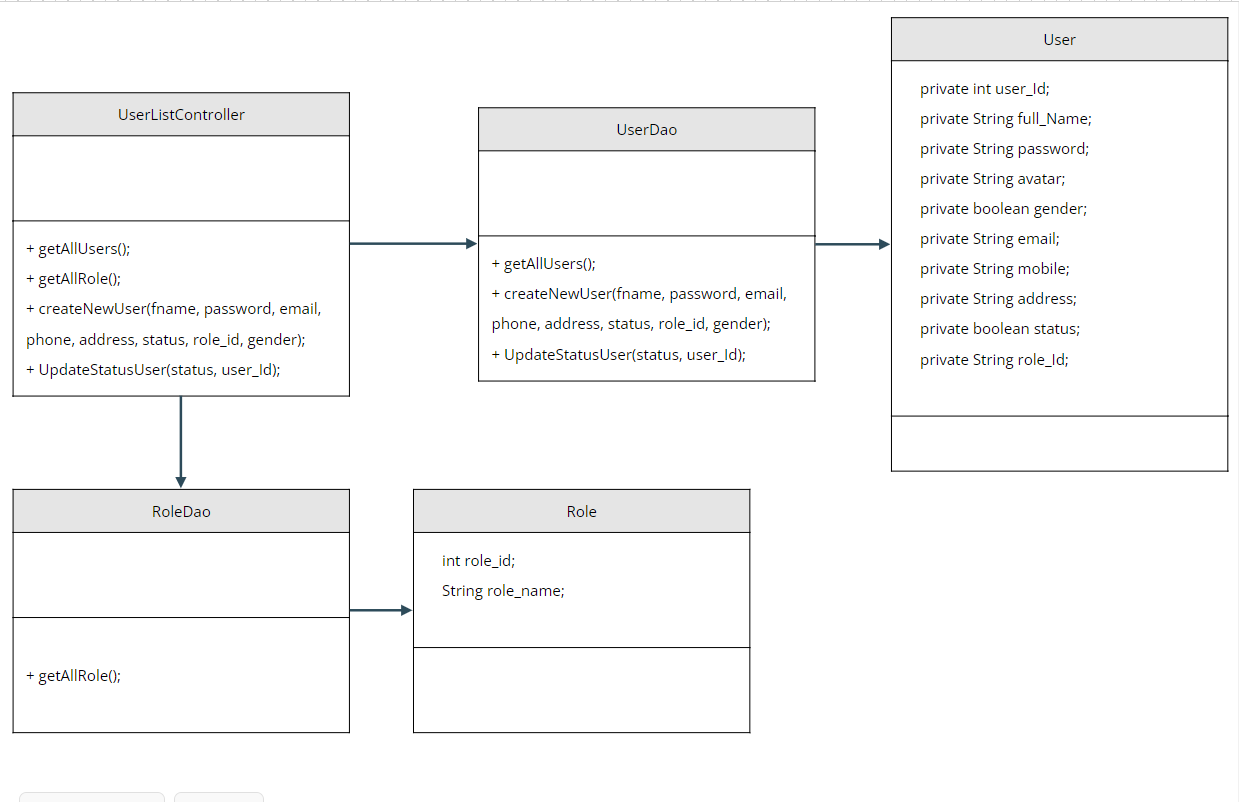
****

### **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**

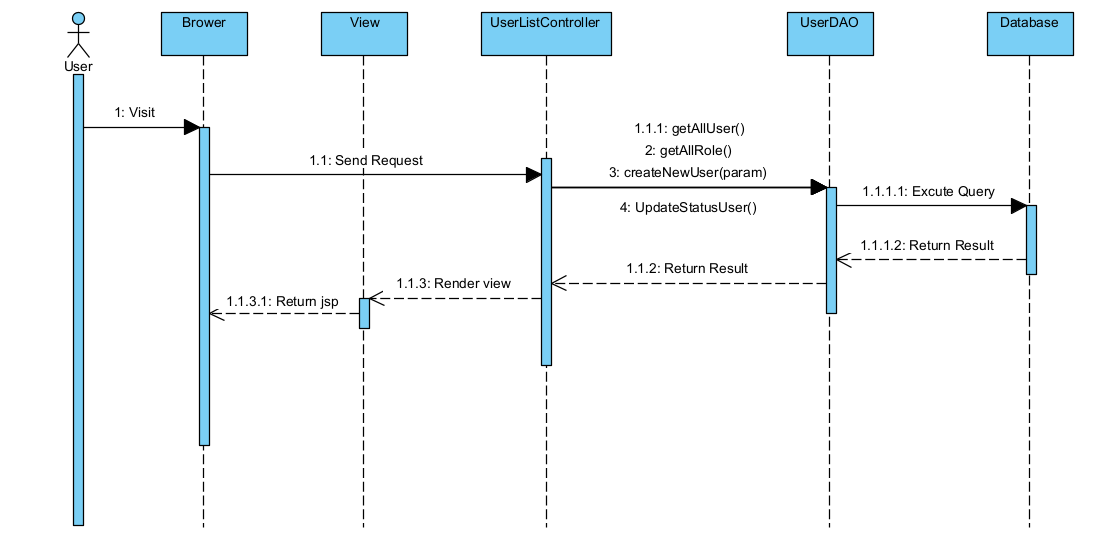


### **b. Class Specifications**

#### **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**

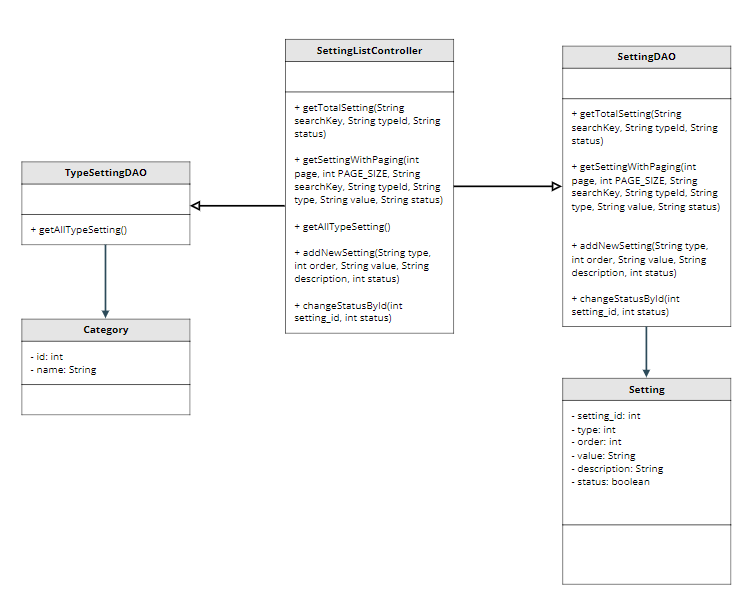
|  |  |  |
| --- | --- | --- |
| **No** | **Method** | **Description** |
| 01 |  |  |
| 02 |  |  |
| 03 |  |  |
| 04 |  |  |

### **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 + "%'

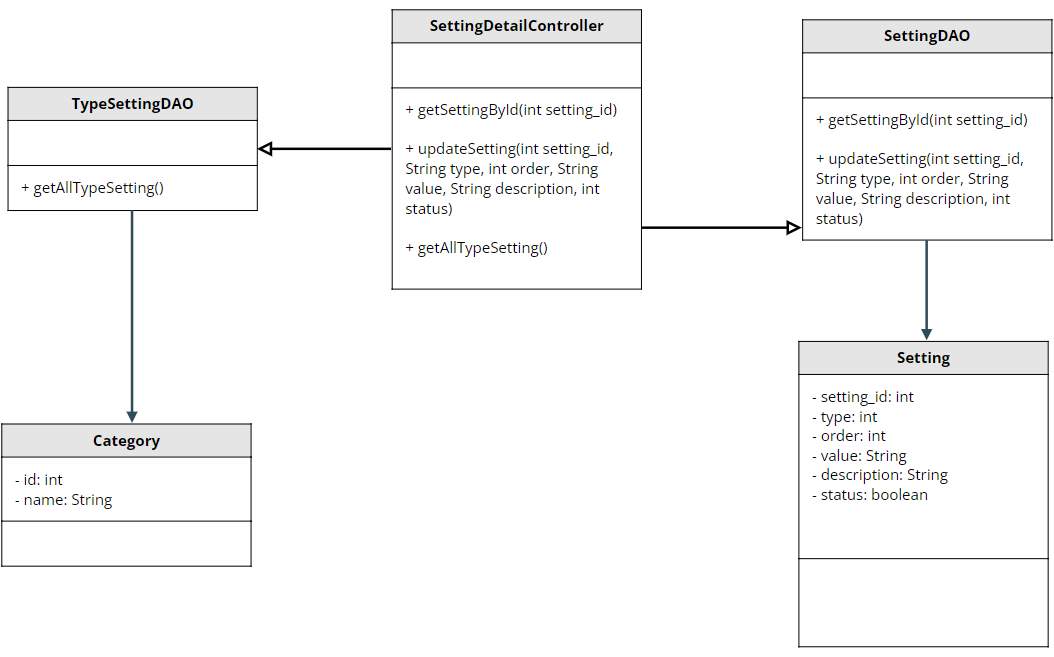
-select \* from Setting where [type] " + typeId + " and [status] " + status + " and [value] like N'%" +searchKey + "%'   order by " + value + " " + type + " offset (?-1)\*? row fetch next ? row only

-INSERT INTO [dbo].[Setting] ([type] ,[order],[value],[description],[status]) VALUES  (?,?,?,?,?)

-select \* from Setting\_Type

## **36. Admin/Setting Details**

### **a. Class Diagram**

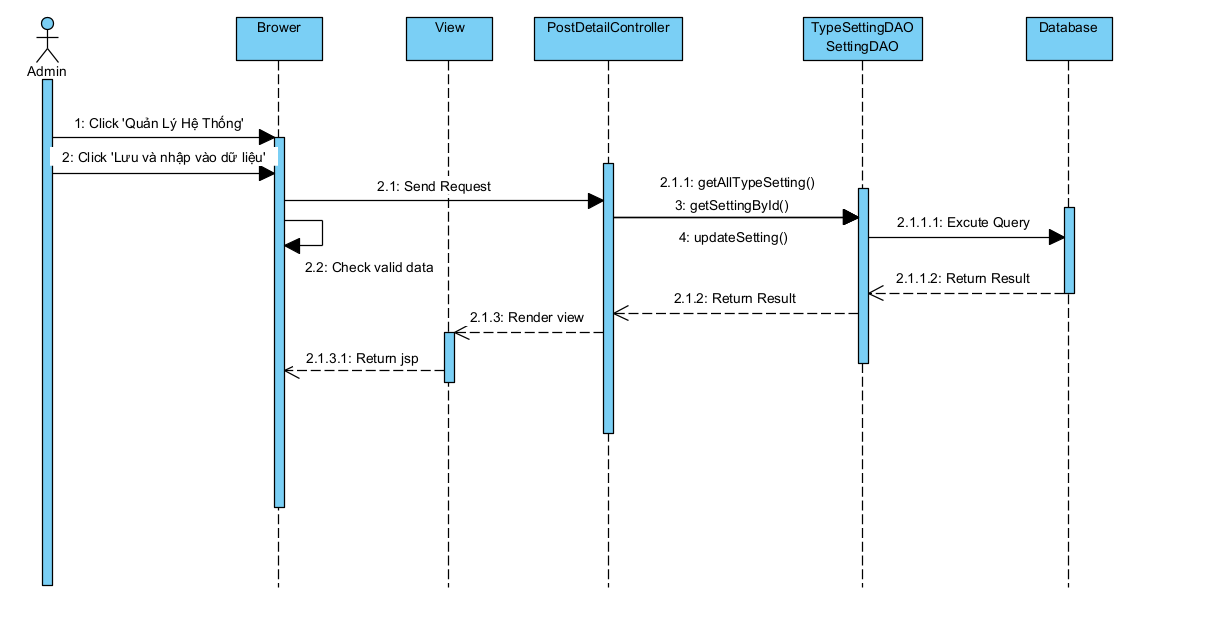


### **b. Class Specifications**

#### **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 = ?

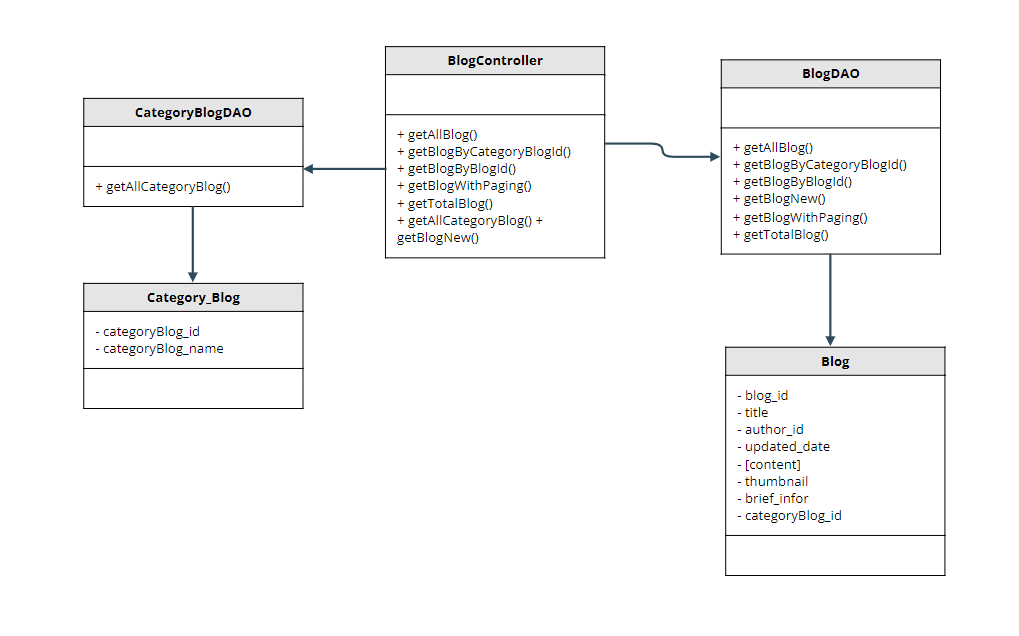
-UPDATE [dbo].[Category\_Blog] SET [categoryBlog\_name] = ? ,[status] = ? WHERE categoryBlog\_id = ?

-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**

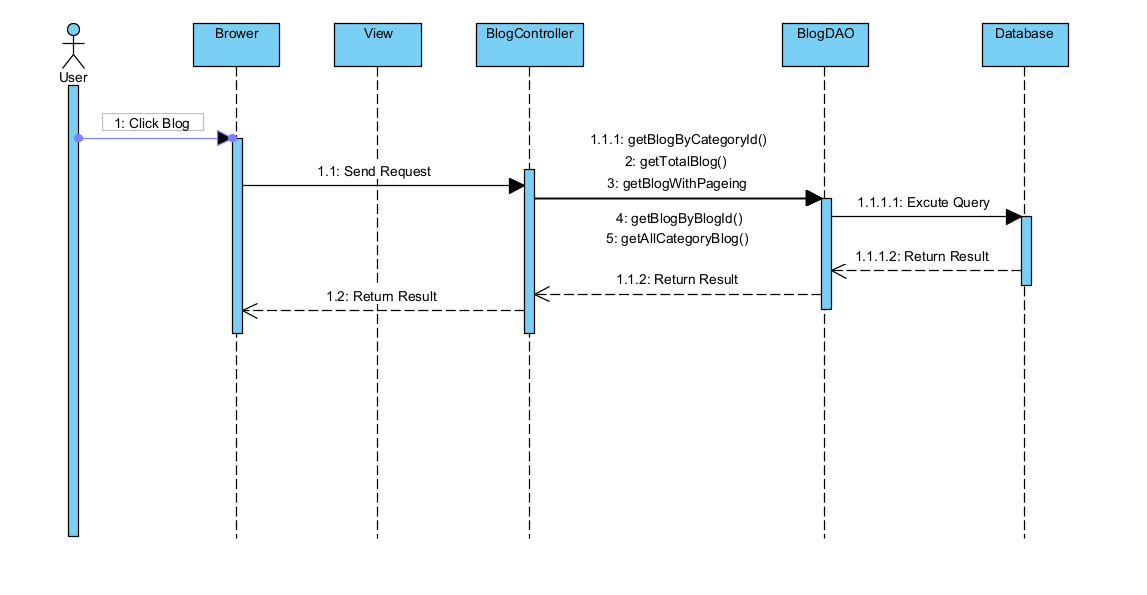


### **b. Class Specifications**

#### **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"

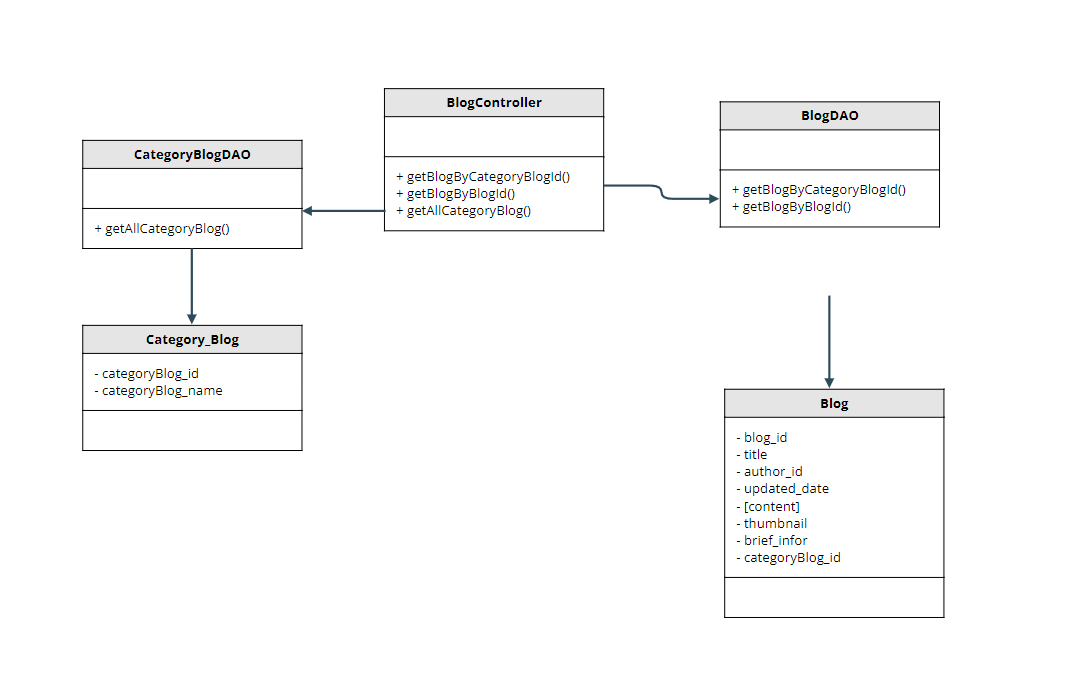
                + " order by " + value + " " + type + " offset (?-1)\*? row fetch next ? row only

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**

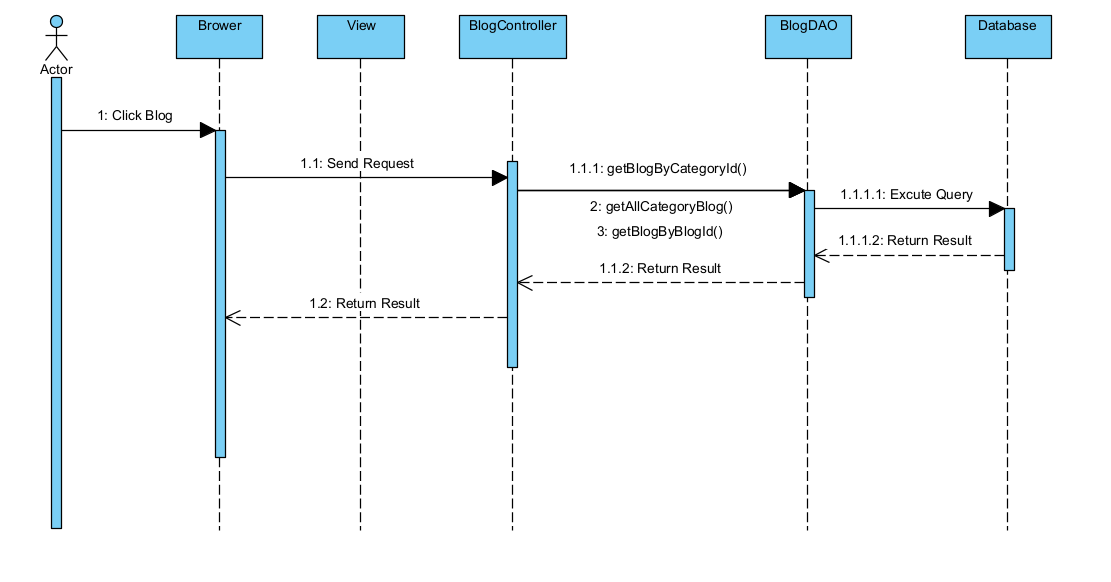


### **b. Class Specifications**

#### 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**

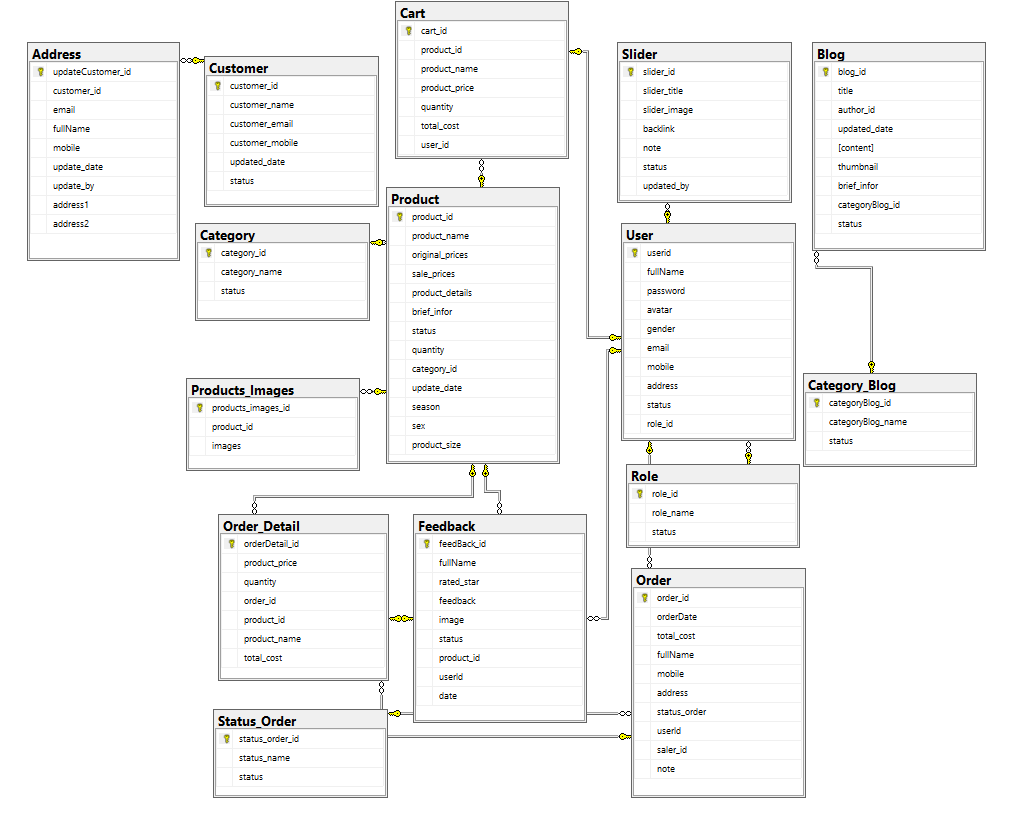
Select \* from Blog

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 int,  product\_name nvarchar(255),  category\_id int,  brief\_infor  original\_prices float,  sale\_prices float,  product\_details nvarchar(max),  quantity int,  status nvarchar,  update\_date date,  season nvarchar ,  sex bit,  product\_size varchar ,  **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 | Address | updateCustomer\_id int,  Customer\_id int,  email varchar(50),  fullName nvarchar(50),  Gender bit,  mobile varchar(50),  Address1 nvarchar(255),  Address2 nvarchar(255)  updated\_date date,  updated\_by int,  **Primary key: udateCustomer\_id**  **Foreign key: Customer\_id** |
| 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** |