

**<<BOOKS SHOPPING ONLINE>>**

**Software Design Specification**

**Group 3**

– Hanoi, July 2023 –

**Record of changes**

| **Date** | **A\* M, D** | **In charge** | **Change Description** |
| --- | --- | --- | --- |
| 06/12/2023 | A | create new SDS | add the basic form for project |
| 06/15/2023 | M | add [Code Packages](https://docs.google.com/document/d/1apZfo7-TvS777-WGCfM2OK7HQjI4EUsRSnFeZtsKyng/edit#heading=h.30j0zll) | add code packages, description |
| 06/16/2023 | M | add Database Design | add datatable schema, diagram relationship and table Schema |
| 06/17/2023 | M | add Home Page | modify Class Diagram, Class Specification, Sequence Diagram, Data Queries |
| 06/19/2023 | M | add Products List | modify Class Diagram, Class Specification, Sequence Diagram, Data Queries |
| 06/20/2023 | M | add [Reset Password](https://docs.google.com/document/d/1apZfo7-TvS777-WGCfM2OK7HQjI4EUsRSnFeZtsKyng/edit#heading=) | modify Class Diagram, Class Specification, Sequence Diagram, Data Queries |
| 06/20/2023 | M | add [Change Password](https://docs.google.com/document/d/1apZfo7-TvS777-WGCfM2OK7HQjI4EUsRSnFeZtsKyng/edit#heading=) | modify Class Diagram, Class Specification, Sequence Diagram, Data Queries |
| 06/21/2023 | M | add [Login](https://docs.google.com/document/d/1apZfo7-TvS777-WGCfM2OK7HQjI4EUsRSnFeZtsKyng/edit#heading=) | modify Class Diagram, Class Specification, Sequence Diagram, Data Queries |
| 06/23/2023 | M | add [Register](https://docs.google.com/document/d/1apZfo7-TvS777-WGCfM2OK7HQjI4EUsRSnFeZtsKyng/edit#heading=) | modify Class Diagram, Class Specification, Sequence Diagram, Data Queries |
| 06/23/2023 | M | add [User ProfileProfile](https://docs.google.com/document/d/1apZfo7-TvS777-WGCfM2OK7HQjI4EUsRSnFeZtsKyng/edit#heading=) | modify Class Diagram, Class Specification, Sequence Diagram, Data Queries |
| 06/24/2023 | M | add [PRODUCT DETAIL](https://docs.google.com/document/d/1apZfo7-TvS777-WGCfM2OK7HQjI4EUsRSnFeZtsKyng/edit#heading=) | modify Class Diagram, Class Specification, Sequence Diagram, Data Queries |
| 06/25/2023 | M | add [Feedback](#_2mqo7hof28ap) | modify Class Diagram, Class Specification, Sequence Diagram, Data Queries |
| 06/27/2023 | M | add [Blogs List](https://docs.google.com/document/d/1apZfo7-TvS777-WGCfM2OK7HQjI4EUsRSnFeZtsKyng/edit#heading=) | modify Class Diagram, Class Specification, Sequence Diagram, Data Queries |
| 06/27/2023 | M | add [Produsts List](https://docs.google.com/document/d/1apZfo7-TvS777-WGCfM2OK7HQjI4EUsRSnFeZtsKyng/edit#heading=h.kvl17jm5pfe6) | modify Class Diagram, Class Specification, Sequence Diagram, Data Queries |
| 06/29/2023 | M | add [Sliders List](https://docs.google.com/document/d/1apZfo7-TvS777-WGCfM2OK7HQjI4EUsRSnFeZtsKyng/edit#heading=h.xmd24puqpob) | modify Class Diagram, Class Specification, Sequence Diagram, Data Queries |
| 06/30/2023 | M | add [CartDetail](https://docs.google.com/document/d/1apZfo7-TvS777-WGCfM2OK7HQjI4EUsRSnFeZtsKyng/edit#heading=h.5tge5a1e6frm) | modify Class Diagram, Class Specification, Sequence Diagram, Data Queries |
| 07/01/2023 | M | add [Cart Contact](https://docs.google.com/document/d/1apZfo7-TvS777-WGCfM2OK7HQjI4EUsRSnFeZtsKyng/edit#heading=h.y9dvp8oi2o91) | modify Class Diagram, Class Specification, Sequence Diagram, Data Queries |
| 07/02/2023 | M | add [My orde](https://docs.google.com/document/d/1apZfo7-TvS777-WGCfM2OK7HQjI4EUsRSnFeZtsKyng/edit#heading=h.creltwlv7gkl)r | modify Class Diagram, Class Specification, Sequence Diagram, Data Queries |
| 07/02/2023 | M | add [Order Information](https://docs.google.com/document/d/1apZfo7-TvS777-WGCfM2OK7HQjI4EUsRSnFeZtsKyng/edit#heading=h.oeup4e6rvt7) | modify Class Diagram, Class Specification, Sequence Diagram, Data Queries |
| 07/04/2023 | M | add [BlogDetail](https://docs.google.com/document/d/1apZfo7-TvS777-WGCfM2OK7HQjI4EUsRSnFeZtsKyng/edit#heading=h.cboua77o8dtm) | modify Class Diagram, Class Specification, Sequence Diagram, Data Queries |
| 07/05/2023 | M | add [MKT Product Detail](#_2fqeaesdjyyb) | modify Class Diagram, Class Specification, Sequence Diagram, Data Queries |
| 07/07/2023 | M | add [MKT Slider Detail](https://docs.google.com/document/d/1apZfo7-TvS777-WGCfM2OK7HQjI4EUsRSnFeZtsKyng/edit#heading=h.26vtias1z0ll) | modify Class Diagram, Class Specification, Sequence Diagram, Data Queries |
| 07/07/2023 | M | add [Posts List](https://docs.google.com/document/d/1apZfo7-TvS777-WGCfM2OK7HQjI4EUsRSnFeZtsKyng/edit#heading=h.2e2c2xoj0pjc) | modify Class Diagram, Class Specification, Sequence Diagram, Data Queries |
| 07/08/2023 | M | add [Posts Detail](https://docs.google.com/document/d/1apZfo7-TvS777-WGCfM2OK7HQjI4EUsRSnFeZtsKyng/edit#heading=h.26ibmty71jrm) | modify Class Diagram, Class Specification, Sequence Diagram, Data Queries |
| 07/09/2023 | M | add [Sale Dashboard](https://docs.google.com/document/d/1apZfo7-TvS777-WGCfM2OK7HQjI4EUsRSnFeZtsKyng/edit#heading=h.9g24akjt8kph) | modify Class Diagram, Class Specification, Sequence Diagram, Data Queries |
| 07/09/2023 | M | add [Orders Details](https://docs.google.com/document/d/1apZfo7-TvS777-WGCfM2OK7HQjI4EUsRSnFeZtsKyng/edit#heading=h.ijwbwr39reyu) | modify Class Diagram, Class Specification, Sequence Diagram, Data Queries |
| 07/09/2023 | M | add [Order List](#_850ip899vnu5) | modify Class Diagram, Class Specification, Sequence Diagram, Data Queries |
| 07/10/2023 | M | add [User List](https://docs.google.com/document/d/1apZfo7-TvS777-WGCfM2OK7HQjI4EUsRSnFeZtsKyng/edit#heading=h.jgsr0b3x9u61) | modify Class Diagram, Class Specification, Sequence Diagram, Data Queries |
| 07/11/2023 | M | add [User Details](https://docs.google.com/document/d/1apZfo7-TvS777-WGCfM2OK7HQjI4EUsRSnFeZtsKyng/edit#heading=h.92ingsz1do2i) | modify Class Diagram, Class Specification, Sequence Diagram, Data Queries |
| 07/11/2023 | M | add [Setting List](https://docs.google.com/document/d/1apZfo7-TvS777-WGCfM2OK7HQjI4EUsRSnFeZtsKyng/edit#heading=h.2kwn2340zws1) | modify Class Diagram, Class Specification, Sequence Diagram, Data Queries |
| 07/11/2023 | M | add [Setting Details](https://docs.google.com/document/d/1apZfo7-TvS777-WGCfM2OK7HQjI4EUsRSnFeZtsKyng/edit#heading=h.4u2doxak5ao4) | modify Class Diagram, Class Specification, Sequence Diagram, Data Queries |
| 07/12/2023 | M | add [Admin DashBoard](https://docs.google.com/document/d/1apZfo7-TvS777-WGCfM2OK7HQjI4EUsRSnFeZtsKyng/edit#heading=h.7neoyat5i0e2) | modify Class Diagram, Class Specification, Sequence Diagram, Data Queries |
| 07/12/2023 | M | add [Feedback List](https://docs.google.com/document/d/1apZfo7-TvS777-WGCfM2OK7HQjI4EUsRSnFeZtsKyng/edit#heading=h.ezi8ekf59371) | modify Class Diagram, Class Specification, Sequence Diagram, Data Queries |
| 07/13/2023 | M | add [Feedback Detail](https://docs.google.com/document/d/1apZfo7-TvS777-WGCfM2OK7HQjI4EUsRSnFeZtsKyng/edit#heading=h.52q9elpdgk5a) | modify Class Diagram, Class Specification, Sequence Diagram, Data Queries |
| 07/13/2023 | M | add [Customer List](https://docs.google.com/document/d/1apZfo7-TvS777-WGCfM2OK7HQjI4EUsRSnFeZtsKyng/edit#heading=h.recmyfn1dbkm) | modify Class Diagram, Class Specification, Sequence Diagram, Data Queries |
| 07/13/2023 | M | add [Customer Detail](https://docs.google.com/document/d/1apZfo7-TvS777-WGCfM2OK7HQjI4EUsRSnFeZtsKyng/edit#heading=h.2ntoxwlhjgsl) | modify Class Diagram, Class Specification, Sequence Diagram, Data Queries |
| 07/14/2023 | M | add [Authorization](https://docs.google.com/document/d/1apZfo7-TvS777-WGCfM2OK7HQjI4EUsRSnFeZtsKyng/edit#heading=h.6daj4has1m2h) | modify Class Diagram, Class Specification, Sequence Diagram, Data Queries |
| 07/14/2023 | M | add [CartCompletion](#_9jgegaz7rshp) | modify Class Diagram, Class Specification, Sequence Diagram, Data Queries |

\*A - Added M - Modified D - Deleted

**Table of Contents**

[**I. Overview 9**](#_gjdgxs)

[1. Code Packages 9](#_30j0zll)

[2. Database Design 10](#_1fob9te)

[a. Database Schema 10](#_3znysh7)

[b. Table Description 11](#_2et92p0)

[**II. Code Designs 15**](#_tyjcwt)

[1. Public/Home Page 15](#_3dy6vkm)

[a. Class Diagram 15](#_1t3h5sf)

[b. Class Specifications 16](#_4d34og8)

[c. Sequence Diagram(s) 17](#_2s8eyo1)

[d. Database Queries 17](#_17dp8vu)

[2. Public/Products List 18](#_3rdcrjn)

[a. Class Diagram 18](#_ozrihvd5h79q)

[b. Class Specifications 18](#)

[c. Sequence Diagram(s) 19](#)

[d. Database queries 19](#)

[3. <Common/Reset Password> 20](#)

[a. Class Diagram 20](#)

[b. Class Specifications 20](#_dpr4h0rjyfnj)

[c. Sequence Diagram(s) 21](#)

[d. Database queries 21](#)

[4. <Common/Change Password> 22](#)

[a. Class Diagram 22](#_26in1rg)

[b. Class Specifications 22](#)

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

[d. Database queries 23](#)

[5. <Common/Login> 24](#)

[a. Class Diagram 24](#_lnxbz9)

[b. Class Specifications 24](#)

[c. Sequence Diagram(s) 25](#_1ksv4uv)

[d. Database queries 25](#)

[6. <Common/Register> 26](#)

[a. Class Diagram 26](#_44sinio)

[b. Class Specifications 26](#)

[c. Sequence Diagram(s) 27](#_2jxsxqh)

[d. Database queries 27](#)

[7. <Common/User ProfileProfile> 28](#)

[a. Class Diagram 28](#_z337ya)

[b. Class Specifications 28](#)

[c. Sequence Diagram(s) 29](#_3j2qqm3)

[d. Database queries 29](#)

[8. <Public/PRODUCT DETAIL> 30](#)

[a. Class Diagram 30](#)

[b. Class Specifications 30](#)

[c. Sequence Diagram(s) 31](#)

[d. Database queries 31](#)

[9. <Common/Feedback > 32](#_2mqo7hof28ap)

[a. Class Diagram 32](#)

[b. Class Specifications 32](#)

[c. Sequence Diagram(s) 33](#)

[d. Database queries 33](#)

[10. Public/Blogs List 34](#)

[a. Class Diagram 34](#_6z8viafsa4nn)

[b. Class Specifications 34](#)

[c. Sequence Diagram(s) 35](#)

[d. Database queries 35](#)

[11.Marketing/Produsts List 36](#_kvl17jm5pfe6)

[a. Class Diagram 36](#_tsgjslk7xrsd)

[b. Class Specifications 36](#_d6p9ecijk872)

[c. Sequence Diagram(s) 37](#_kz4us0h13wp0)

[d. Database queries 37](#_jc5tnn2tyjoh)

[12.Marketing/Sliders List 38](#_xmd24puqpob)

[a. Class Diagram 38](#_mb2v2ga32pdx)

[b. Class Specifications 38](#_7bmox6fmssr1)

[c. Sequence Diagram(s) 38](#_fjilelacyn8n)

[d. Database queries 38](#_h8n4l7g3auqx)

[13.Public/CartDetail 39](#_5tge5a1e6frm)

[a. Class Diagram 39](#_qq5r9bvift3i)

[b. Class Specifications 39](#_p4zv2bu9nytp)

[c. Sequence Diagram(s) 40](#_b3q6pcfmq11t)

[d. Database queries 40](#_jy2ulp1ogkzs)

[14.Public/Cart Contact 41](#_y9dvp8oi2o91)

[a. Class Diagram 41](#_5d5ua3xy9s9)

[b. Class Specifications 41](#_475ymuym2elb)

[c. Sequence Diagram(s) 42](#_d7w4dp7q3ned)

[d. Database queries 42](#_x8boya61trvx)

[15.My order 42](#_creltwlv7gkl)

[a. Class Diagram 42](#_y4xsbppzp1et)

[b. Class Specifications 42](#_i3895mgjfkf)

[c. Sequence Diagram(s) 43](#_2rt8ufmg38cm)

[d. Database queries 43](#_6eb761txcc19)

[16.Order Information 44](#_oeup4e6rvt7)

[a. Class Diagram 44](#_vodvyqp7jmwz)

[b. Class Specifications 44](#_ze837e7hth1m)

[c. Sequence Diagram(s) 45](#_90d15pdgrlr7)

[d. Database queries 45](#_bjxyvrpnhoy)

[17.Public/BlogDetail 47](#_cboua77o8dtm)

[a. Class Diagram 47](#_i46ho4d9aye5)

[b. Class Specifications 47](#_mjmgupgqpw1e)

[c. Sequence Diagram(s) 48](#_874wt1xaq4g2)

[d. Database queries 48](#_50ezuofeksju)

[18.Marketing/MKT Dashboard 48](#_wg24cqklccaf)

[a. Class Diagram 48](#_ex24zodk2cfh)

[b. Class Specifications 49](#_fpup2dg6edjf)

[c. Sequence Diagram(s) 49](#_w7scidmdv6e3)

[d. Database queries 49](#_n4jmk5druvcu)

[19.Marketing/MKT Product Detail 50](#_2fqeaesdjyyb)

[a. Class Diagram 50](#_udmd4zgjz9fz)

[b. Class Specifications 50](#_l9ip3zhqj60v)

[c. Sequence Diagram(s) 51](#_vhycjpfwb8vq)

[d. Database queries 51](#_x6rj5ktaq47)

[20.Marketing/MKT Slider Detail 52](#_26vtias1z0ll)

[a. Class Diagram 52](#_8wb14rupksaz)

[b. Class Specifications 52](#_djfci9uuuz51)

[c. Sequence Diagram(s) 53](#_g4x2dvvh3rfm)

[d. Database queries 53](#_228cs83ti9zm)

[21.Marketing/Posts List 54](#_2e2c2xoj0pjc)

[a. Class Diagram 54](#_mffogh6odjnj)

[b. Class Specifications 54](#_ubo1gsgi1nxg)

[c. Sequence Diagram(s) 55](#_hdc9ads9wpe5)

[d. Database queries 55](#_wu2uoq33c7mh)

[22.Marketing/Posts Detail 56](#_26ibmty71jrm)

[a. Class Diagram 56](#_w14it1b36ups)

[b. Class Specifications 57](#_14bkbl33sg7i)

[c. Sequence Diagram(s) 58](#_4sw4scpy6l00)

[d. Database queries 58](#_atv2dzxidox0)

[23.Sale/Sale Dashboard 59](#_9g24akjt8kph)

[a. Class Diagram 59](#_mknfaoqnev6)

[b. Class Specifications 59](#_laek3i69rzpy)

[c. Sequence Diagram(s) 60](#_jrweuyu06afw)

[d. Database queries 60](#_ksmhfnihzxtb)

[24.Sale/Orders Details 61](#_ijwbwr39reyu)

[a. Class Diagram 61](#_mrn8avx9vek9)

[b. Class Specifications 62](#_r1id8prhwu0r)

[c. Sequence Diagram(s) 63](#_9p4ny9ssdpi4)

[d. Database queries 63](#_m3xoglfh02en)

[25.Sale/Order List 64](#_850ip899vnu5)

[a. Class Diagram 64](#_3h83ofxey85p)

[b. Class Specifications 64](#_5qc24grz0mab)

[c. Sequence Diagram(s) 65](#_r03foin8946n)

[d. Database queries 65](#_dyv7r526b8cv)

[26.Admin/User List 66](#_jgsr0b3x9u61)

[a. Class Diagram 66](#_sly4wuecutk0)

[b. Class Specifications 66](#_choiybwi2ede)

[c. Sequence Diagram(s) 67](#_iwpdmgwrh5i3)

[d. Database queries 67](#_nkpr41avr78)

[27.Admin/User Details 68](#_92ingsz1do2i)

[a. Class Diagram 68](#_5pt0gkodonh0)

[b. Class Specifications 68](#_8qzt897h68ln)

[c. Sequence Diagram(s) 69](#_o2p9k7dx7uqa)

[d. Database queries 69](#_z0tyydm7r5p2)

[28.Admin/Setting List 70](#_2kwn2340zws1)

[a. Class Diagram 70](#_szigq8y5xvxo)

[b. Class Specifications 70](#_mdzipw1b6rx8)

[c. Sequence Diagram(s) 70](#_9y4zqkx4m3vm)

[d. Database queries 71](#_v4yafetax7bz)

[29.Admin/Setting Details 72](#_4u2doxak5ao4)

[a. Class Diagram 72](#_szv634qgfgnz)

[b. Class Specifications 72](#_wu7nggzg0w1f)

[c. Sequence Diagram(s) 72](#_nuw69nqtfk0b)

[d. Database queries 72](#_8gwz10pcshjo)

[30.Admin/Admin DashBoard 73](#_7neoyat5i0e2)

[a. Class Diagram 73](#_ldp2cpqelwhf)

[b. Class Specifications 73](#_it0y3rnj39mi)

[c. Sequence Diagram(s) 74](#_gdrnfgbh0j8n)

[d. Database queries 74](#_8neqse28hs46)

[31.Marketing/Feedback List 75](#_ezi8ekf59371)

[a. Class Diagram 75](#_deyeh96agdi7)

[b. Class Specifications 75](#_s7yh17qukrec)

[c. Sequence Diagram(s) 76](#_wnsszkt99862)

[d. Database queries 76](#_hxz2ni4cpc3h)

[32.Marketing/Feedback Detail 77](#_52q9elpdgk5a)

[a. Class Diagram 77](#_h1b7lswysidq)

[b. Class Specifications 77](#_sk67n3rs0yjw)

[c. Sequence Diagram(s) 77](#_19q1m3kg8h3r)

[d. Database queries 77](#_cyv5t5faworv)

[33.Marketing/Customer List 78](#_recmyfn1dbkm)

[a. Class Diagram 78](#_ihwdf33fiwd7)

[b. Class Specifications 79](#_yza94f6efiky)

[c. Sequence Diagram(s) 79](#_y0rus7a6tg3m)

[d. Database queries 79](#_7je98r57uxyt)

[34.Marketing/Customer Detail 81](#_2ntoxwlhjgsl)

[a. Class Diagram 81](#_omlmnsxd1yq9)

[b. Class Specifications 81](#_v4rczw4j4j5f)

[c. Sequence Diagram(s) 82](#_q3p9htyuhvhg)

[d. Database queries 82](#_cwvi2fgm9s0i)

[35. <Common/Authorization > 83](#_6daj4has1m2h)

[a. Class Diagram 84](#_6k2ih6a0e4ck)

[b. Class Specifications 84](#_y9jf0ye1ove3)

[c. Sequence Diagram(s) 85](#_1gboaj8lusiq)

[d. Database queries 86](#_rdxl0yhhars)

[36.Public/CartCompletion 86](#_9jgegaz7rshp)

[a. Class Diagram 86](#_lvbtuw4g1sj4)

[b. Class Specifications 88](#_y3wbf2rurvg7)

[c. Sequence Diagram(s) 89](#_dg7zoe7tlffw)

[d. Database queries 90](#_64nybbcc1w7w)

# I. Overview

## 1. Code Packages

*[Provide the package diagram for each sub-system. The content of this section including the overall package diagram, the explanation, package and class naming conventions in each package. Please see the sample & description table format below (****please note: package names don’t follow Java package naming convention yet****)]*

A picture containing diagram, plan, rectangle, text

Description automatically generated

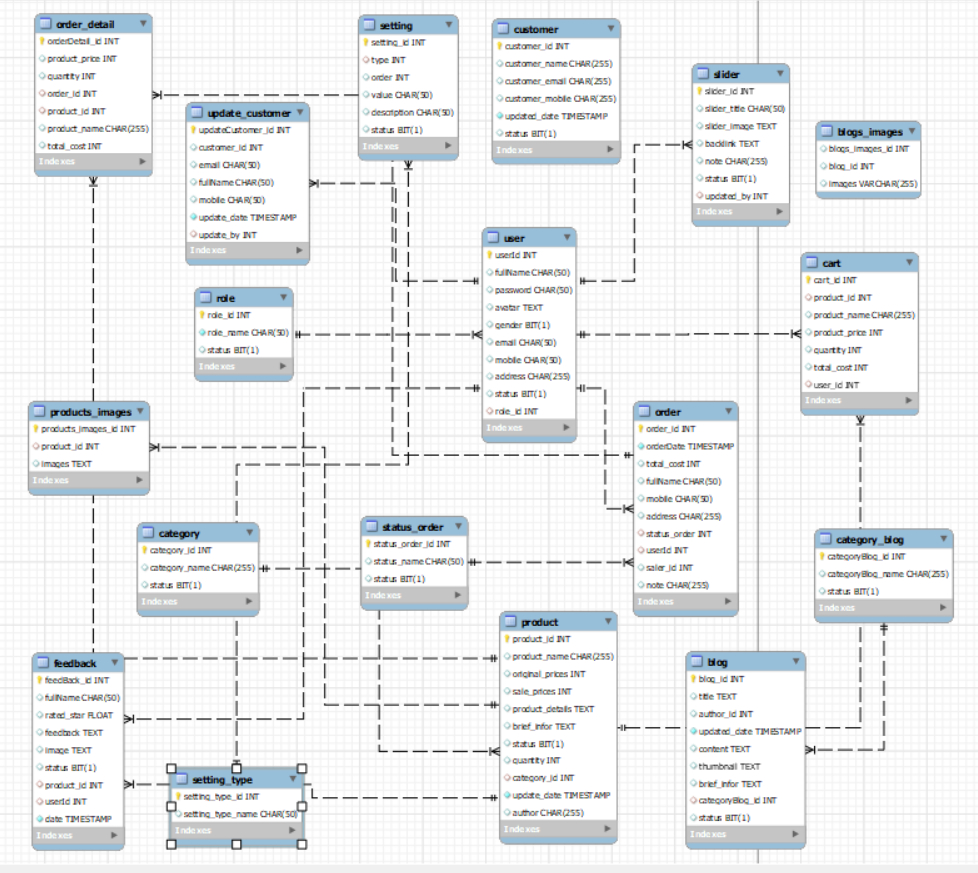
***Package descriptions & package class naming conventions***

| **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 databases.* |
| *05* | *model* | *This component includes interfaces used to receive data retrieved FROM the database and classes used to receive data.* |
| *05* | *view* | *This component displays front-end.* |

## 2. Database Design

### a. Database Schema

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



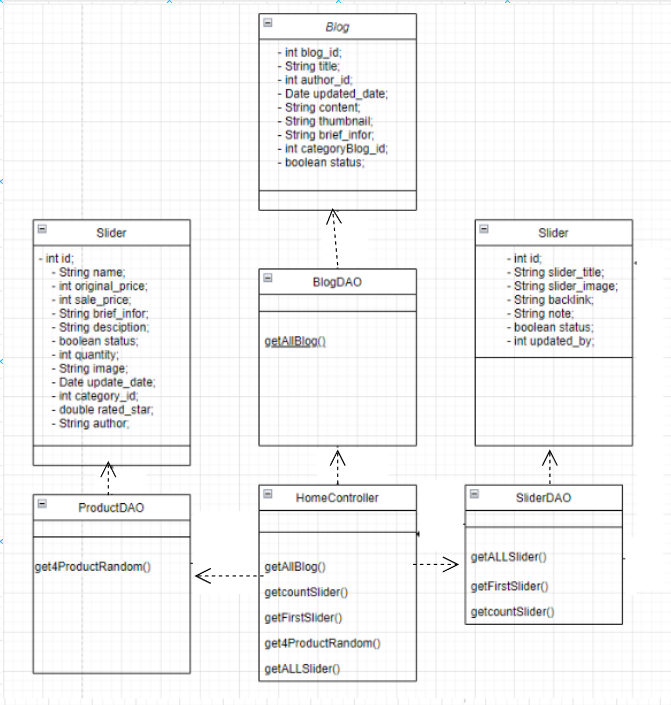
### b. Table Description

| **#** | **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)  **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)  Author 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)  **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** |

# II. Code Designs

## 1. Public/Home Page

### a. Class Diagram



### b. Class Specifications

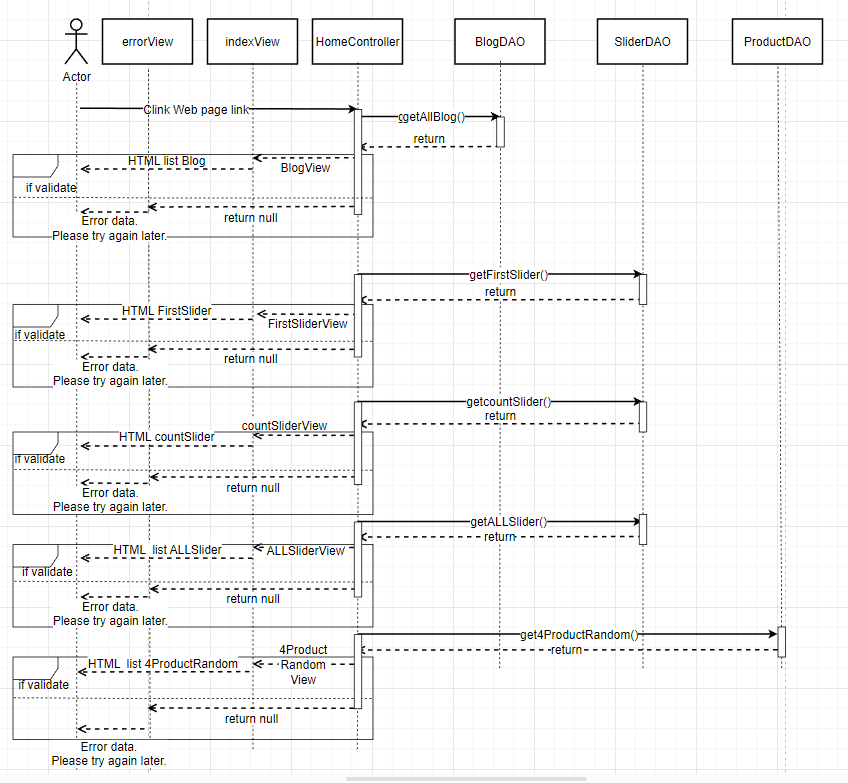
*Provide the description for each class and the methods in each class, following the table format as below]*

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

| **No** | **Method** | **Description** |
| --- | --- | --- |
| *01* | *getAllIBlog()* | *get 2 Blog random from Blog* |
| *02* | *getFirstSlider()* | *get all property in slider first* |
| *03* | *getALLSlider()* | *get all property in slider* |
| *04* | *get4ProductRandom ()* | *get 4 product random from products* |
| *05* | *getcountSlider()* | *count All Slide* |

### c. Sequence Diagram(s)

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



### d. Database Queries

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

SELECT count (\*) FROM books\_shop\_online.slider where status = 1

SELECT \* FROM books\_shop\_online.slider WHERE status = 1 LIMIT 1;

SELECT \* FROM Slider where status = 1

SELECT \* FROM Product ORDER BY RAND () LIMIT 4;

SELECT \* FROM books\_shop\_online.blog ORDER BY RAND() LIMIT 2

## 2. Public/Products List

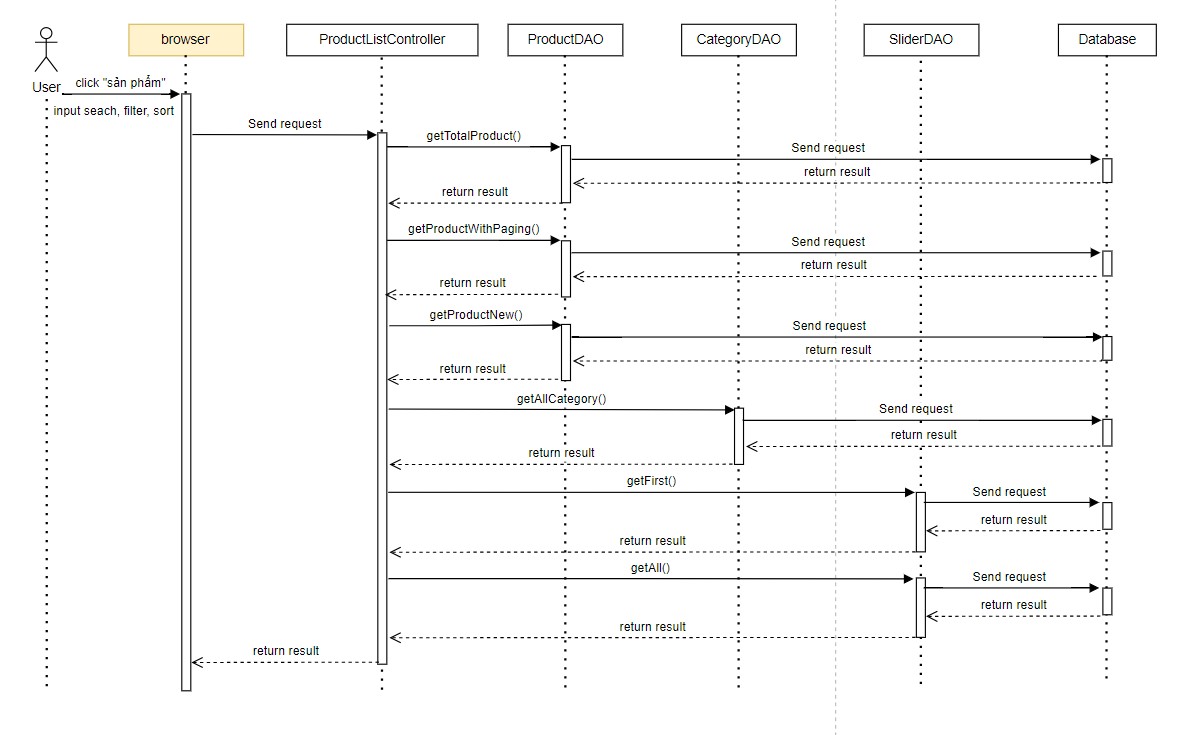
### a. Class Diagram

# 

### b. Class Specifications

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

### c. Sequence Diagram(s)



### d. Database queries

* SELECT COUNT(product\_id) FROM Product\n"

+ "WHERE category\_id " + categoryId + " AND status " + status + " AND product\_name LIKE '%" + searchKey + "%';

* select \* from product\n"

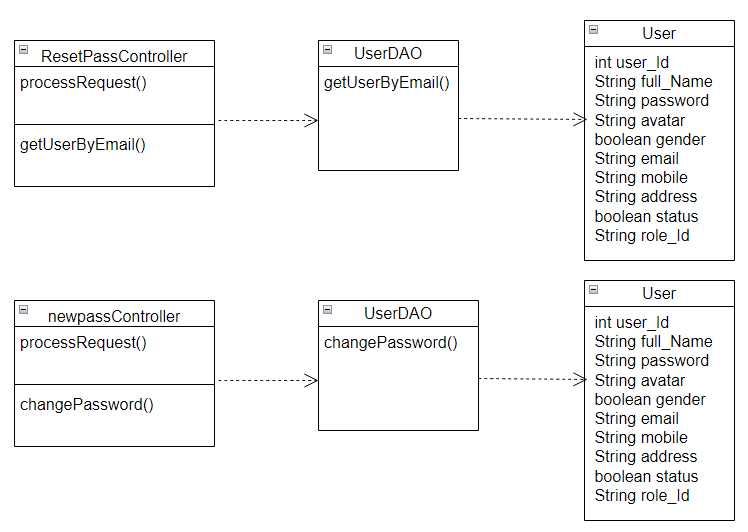
+ "where category\_id " + categoryId + " and status " + status + " and product\_name like N'%" + searchKey + "%'\n"

+ " order by " + value + " " + type + " LIMIT ?, ?;

* Select \* from Category where status = 1
* SELECT \* FROM Product ORDER BY update\_date DESC LIMIT 1;
* SELECT \* FROM Slider

## 3. <Common/Reset Password>

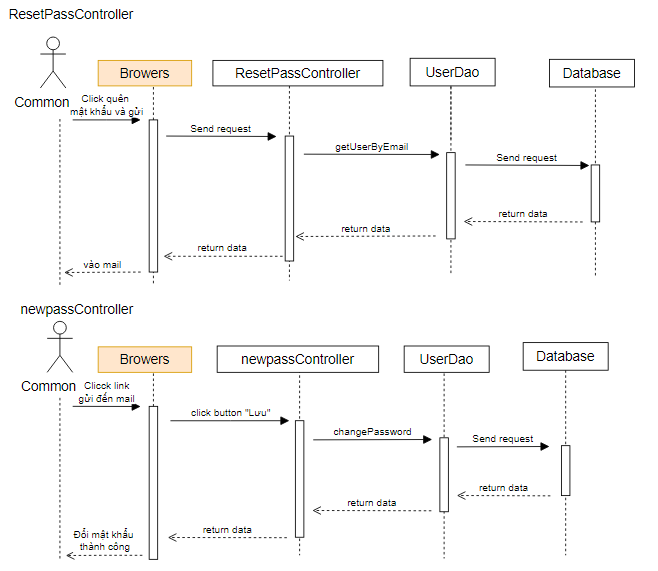
### a. Class Diagram



### b. Class Specifications

| **No** | **Method** | **Description** |
| --- | --- | --- |
| *01* | *getUserByEmail()* | *get User By Email* |
| *02* | *changePassword()* | *change Password* |

### c. Sequence Diagram(s)

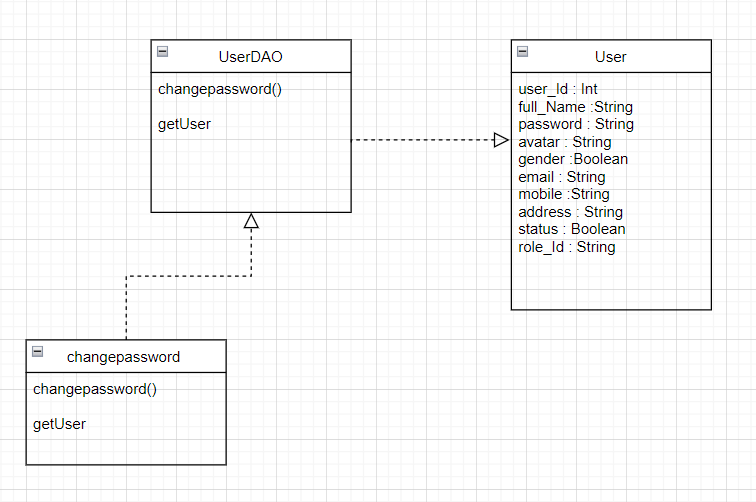


### d. Database queries

* "select \* from user where email = ?"
* "UPDATE `User` SET `password` = ? WHERE `userId` = ?"

## 4. <Common/Change Password>

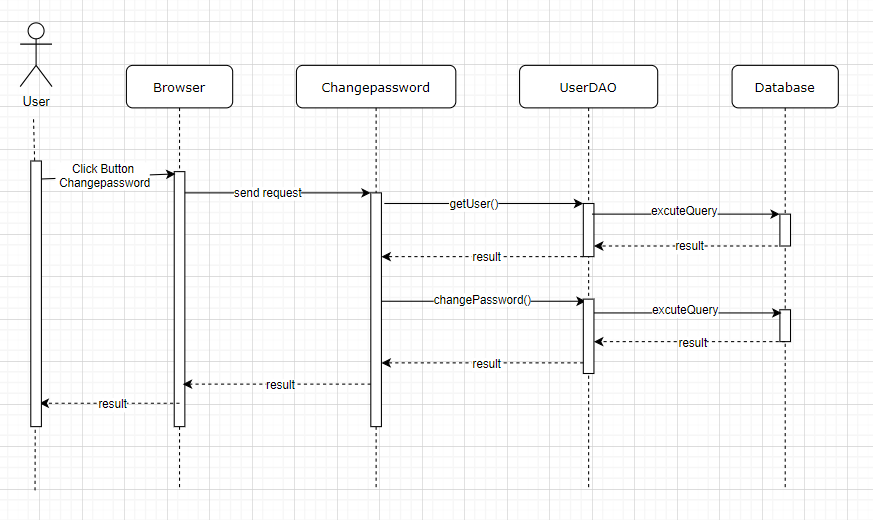
### a. Class Diagram



### b. Class Specifications

| **No** | **Method** | **Description** |
| --- | --- | --- |
| *01* | *getUser()* | *Get the user's old password* |
| *02* | *changePassword()* | *Change the new password to the database* |

### c. Sequence Diagram(s)



### d. Database queries

* SELECT \* FROM `User` WHERE userId = ?
* UPDATE `User`\n"

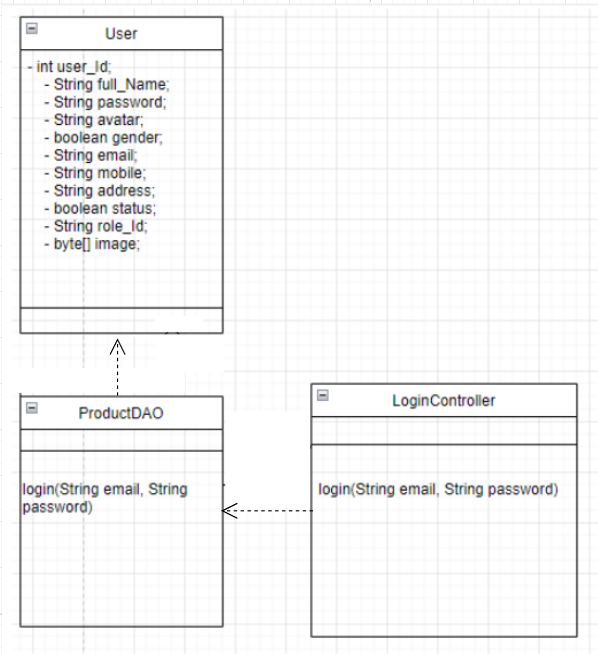
+ " SET \n"

+ " `password` = ?\n"

+ " WHERE `userId` = ?

## 5. <Common/Login>

### a. Class Diagram



### b. Class Specifications

#### 

| **No** | **Method** | **Description** |
| --- | --- | --- |
| *01* | *Login(String email, String password)***l** | *Using to login into the system* |

### c. Sequence Diagram(s)

### 

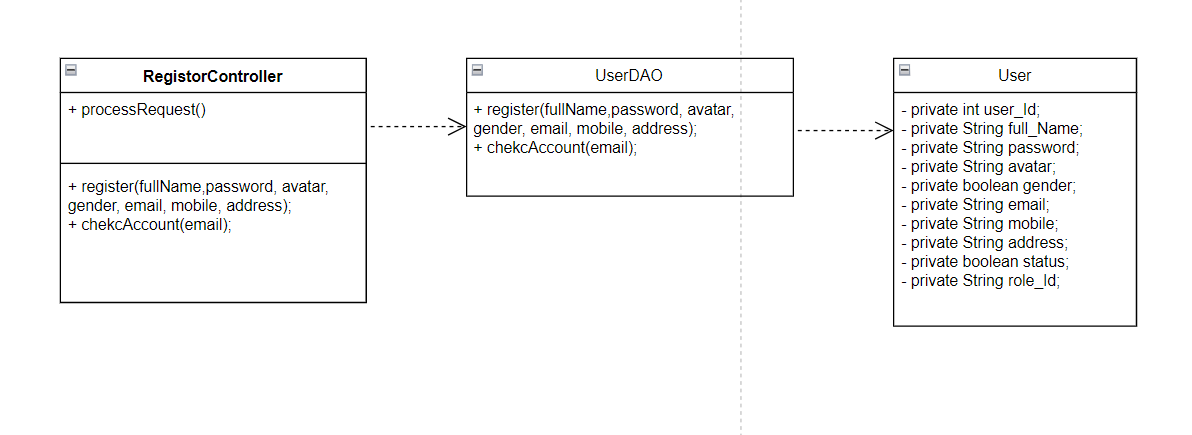
### d. Database queries

select \* from user\n"

+ "where email = ? and password = ?

## 6. <Common/Register>

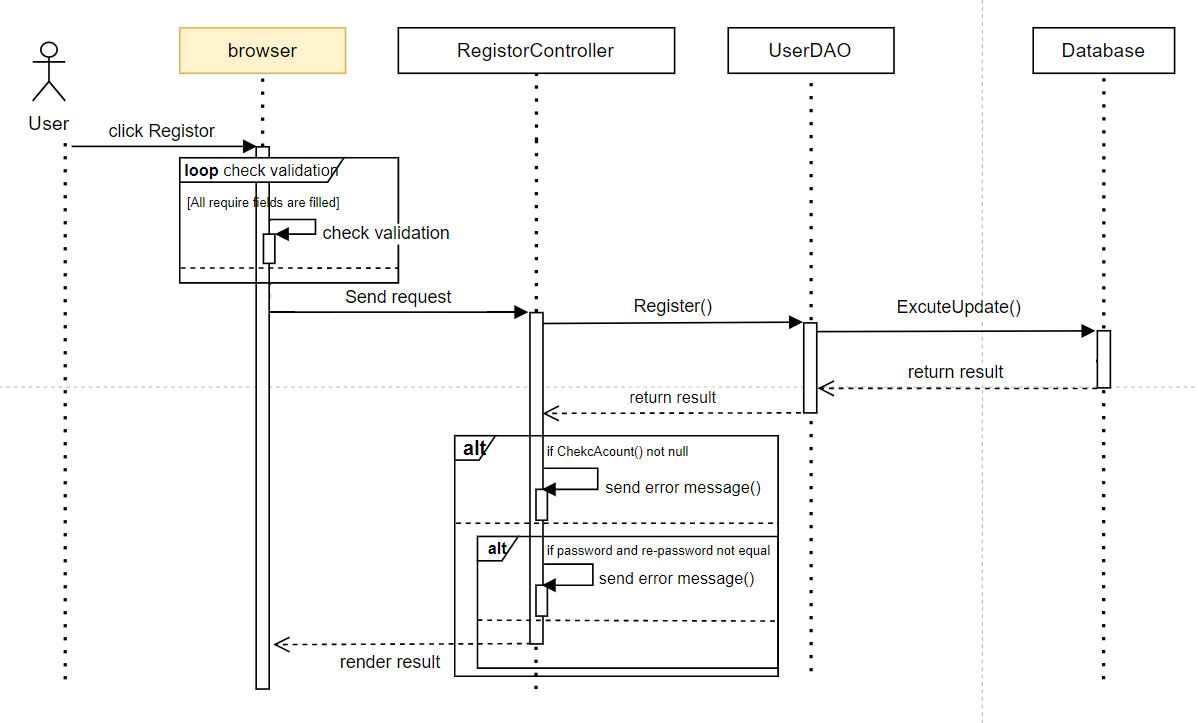
### a. Class Diagram



### b. Class Specifications

| **No** | **Method** | **Description** |
| --- | --- | --- |
| *01* | *Register()* | *use to create new user* |
| *02* | *chekcAccount()* | *use to check email exist* |

### c. Sequence Diagram(s)



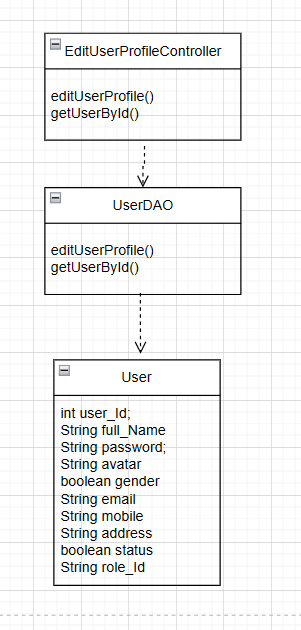
### d. Database queries

* select \* from user where email = ?
* "INSERT INTO user (`fullName`, `password`,`gender`, `email`, `mobile`, `address`, `status`, `role\_id`) VALUES \n"

+ "(?,?,b?,?,?,?,0,1)"

## 7. <Common/User ProfileProfile>

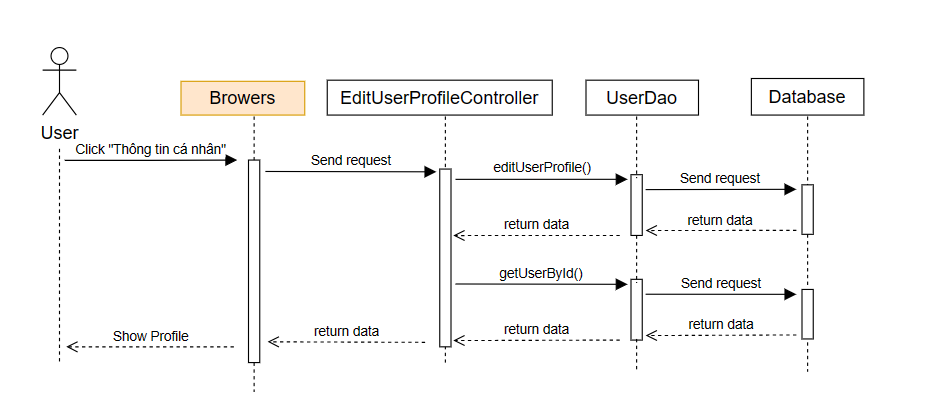
### a. Class Diagram



### b. Class Specifications

| **No** | **Method** | **Description** |
| --- | --- | --- |
| *01* | *getUserById()* | *get User By Id* |
| *02* | *editUserProfile()* | *use to update user’s profile* |

### c. Sequence Diagram(s)



### d. Database queries

update books\_shop\_online.User\n"

+ "set fullName = ?,\n"

+ "avatar = ?,\n"

+ "gender = ?,\n"

+ "mobile = ?,\n"

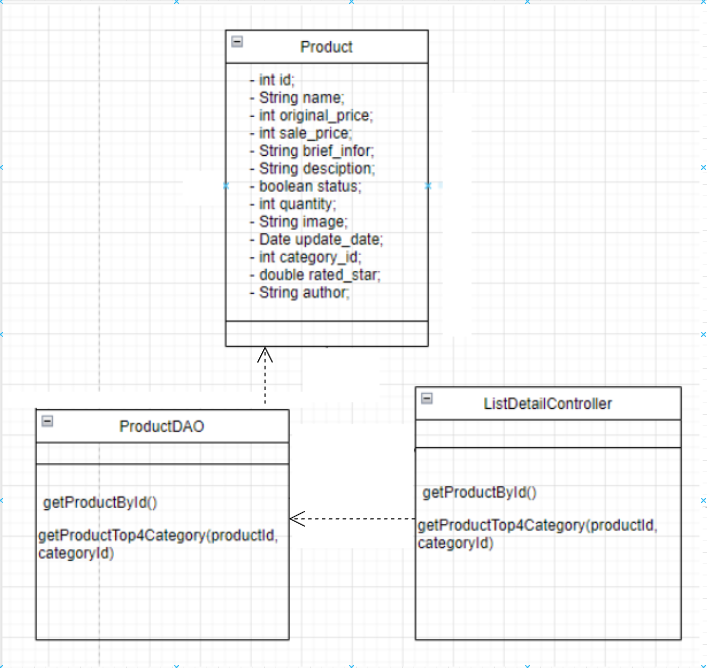
+ "address = ?\n"

+ "where userId = ?

select \* from books\_shop\_online.user where userId = ?

## 8. <Public/PRODUCT DETAIL>

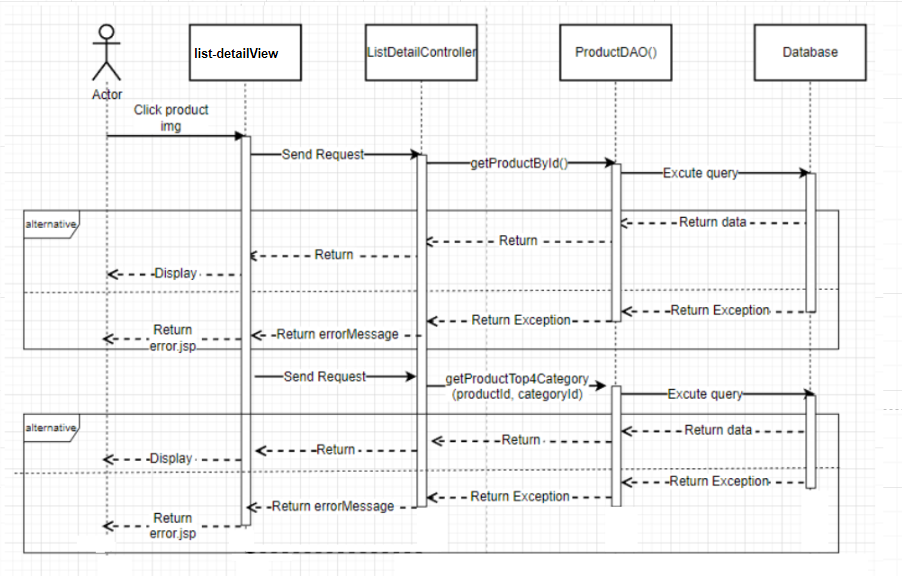
### a. Class Diagram



### b. Class Specifications

| **No** | **Method** | **Description** |
| --- | --- | --- |
| *01* | *getProductById()* | *Use to retrieve products by ID* |
| *02* | *getProductTop4Category(productId, categoryId)* | *get top 4 product have Category relate to* |

### c. Sequence Diagram(s)



### d. Database queries

* select \* from Product where product\_id = ?;
* SELECT \* FROM Product WHERE category\_id = ? AND product\_id != ? ORDER BY RAND() LIMIT 4

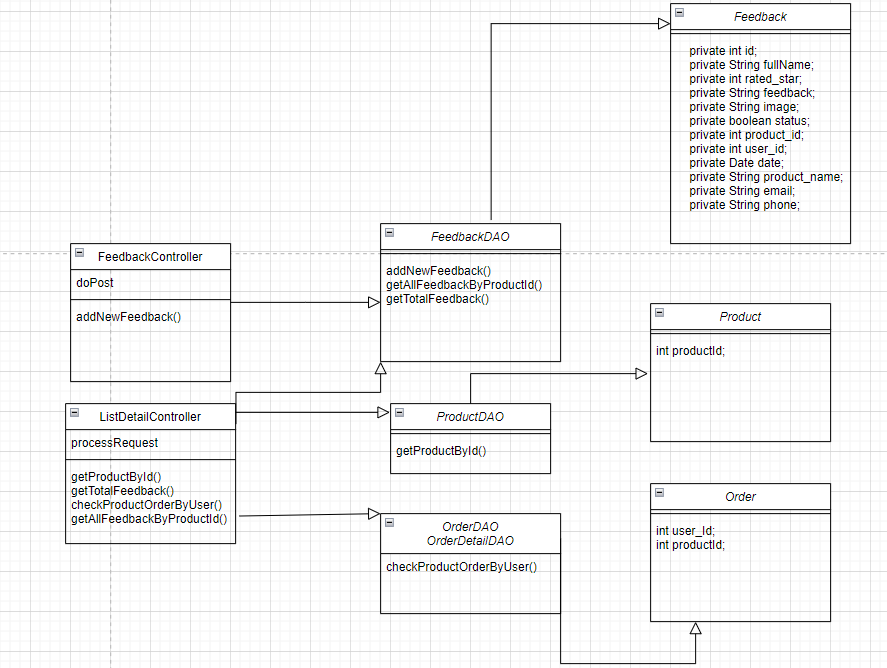
## 

## 

## 

## 9. <Common/Feedback >

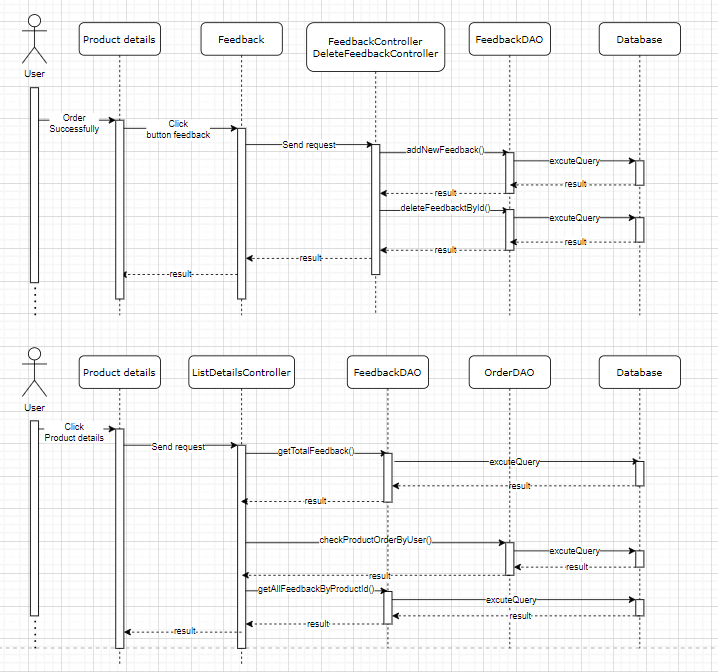
### a. Class Diagram



### b. Class Specifications

| **No** | **Method** | **Description** |
| --- | --- | --- |
| *01* | *addNewFeedback()* | *add new user feedback* |
| *02* | *getProductById()* | *get product id* |
| *03* | *getTotalFeedback()* | *get total user feedback* |
| *04* | *checkProductOrderByUser()* | *Check if the user has ordered the product or not* |
| *05* | *getAllFeedbackByProductId()* | *print all product responses to the screen* |

### c. Sequence Diagram(s)

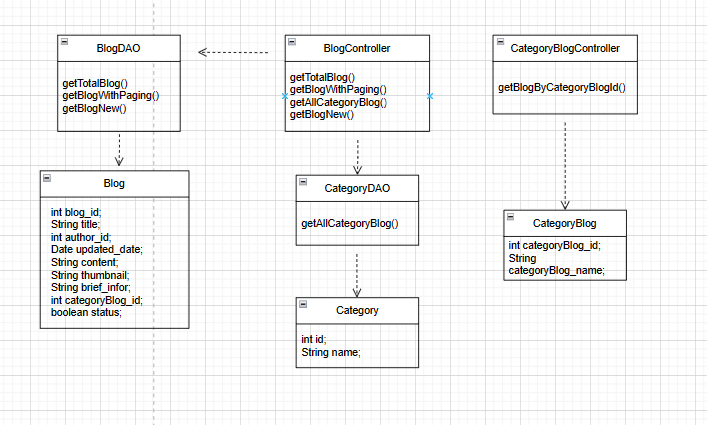


### d. Database queries

- INSERT INTO `books\_shop\_online`.`feedback` ( `fullName`, `rated\_star`, `feedback`, `image`, `status`, `product\_id`, `userId`) VALUES (?, ?, ?, ?, ?, ?, ?)"

## 10. Public/Blogs List

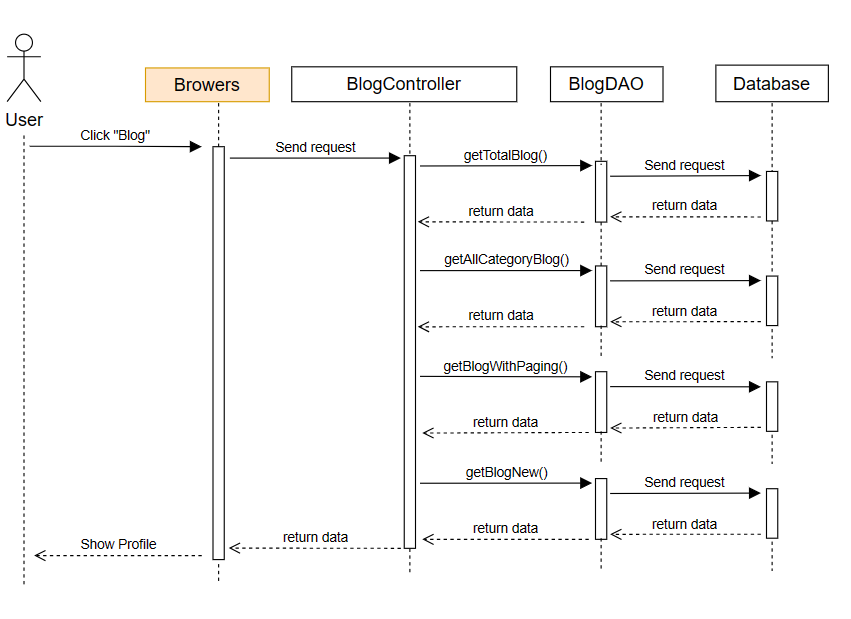
### a. Class Diagram



### b. Class Specifications

| **No** | **Method** | **Description** |
| --- | --- | --- |
| *01* | *getTotalBlog()* | *Get the amount of all Blogs* |
| *02* | *getBlogWithPaging()* | *Get all Blog pagination in database* |
| *03* | *getAllCategoryBlog()* | *Get all category in database* |
| *04* | *getBlogNew()* | *Get a blog latest post* |

### c. Sequence Diagram(s)



### d. Database queries

-SELECT COUNT(\*)\n"

+ "FROM books\_shop\_online.blog\n"

+ "WHERE categoryBlog\_id = "+ categoryId+"\n"

+ " AND status = 1\n"

+ " AND title LIKE CONCAT('%', '"+searchKey+"', '%');

* select \* from books\_shop\_online.blog \n"

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

+ " order by " + value + " " + type + " LIMIT ?, ?;

* Select \* from books\_shop\_online.category\_blog where status = 1
* SELECT \*\n"
* + "FROM books\_shop\_online.blog\n"
* + "WHERE status = 1\n"
* + "ORDER BY updated\_date DESC\n"
* + "LIMIT 1;

## 11.Marketing/Produsts List

### a. Class Diagram

## 

## 

### b. Class Specifications

| **No** | **Method** | **Description** |
| --- | --- | --- |
| *01* | *getTotalPublishedProduct()* | *Get total of all pushlish Product* |
| *02* | *getTotalProduct()* | *Get total of all Product* |
| *03* | *getTotalProduct(searchKey, categoryId, status)* | *Get all product by searching, filter* |
| *04* | *getProductWithPaging()* | *Get Product with page* |
| *05* | *getAllCategory()* | *Get All category for filter product* |

### c. Sequence Diagram(s)



### d. Database queries

* select COUNT(product\_id) from Product;
* select COUNT(product\_id) from Product where status = 1;
* SELECT COUNT(product\_id) FROM Product\n"

+ "WHERE category\_id " + categoryId + " AND status " + status + " AND product\_name LIKE '%" + searchKey + "%';

* select \* from product\n"

+ "JOIN products\_images ON product.`product\_id` = products\_images.`product\_id`"

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

+ " order by " + value + " " + type + " LIMIT ?, ?;

* Select \* from books\_shop\_online.category where status = 1

## 12.Marketing/Sliders List

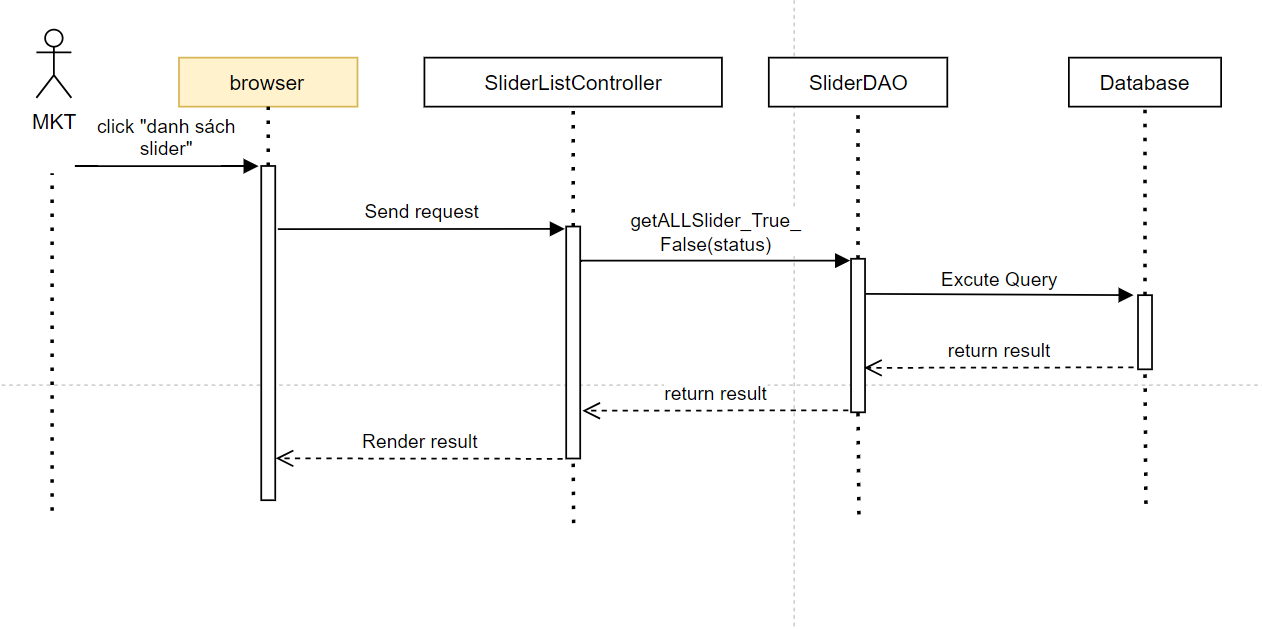
### a. Class Diagram

### 

### b. Class Specifications

| **No** | **Method** | **Description** |
| --- | --- | --- |
| *01* | *getALLSlider\_True\_False(status)* | *Get all Slider by status (default null) in database* |

### c. Sequence Diagram(s)

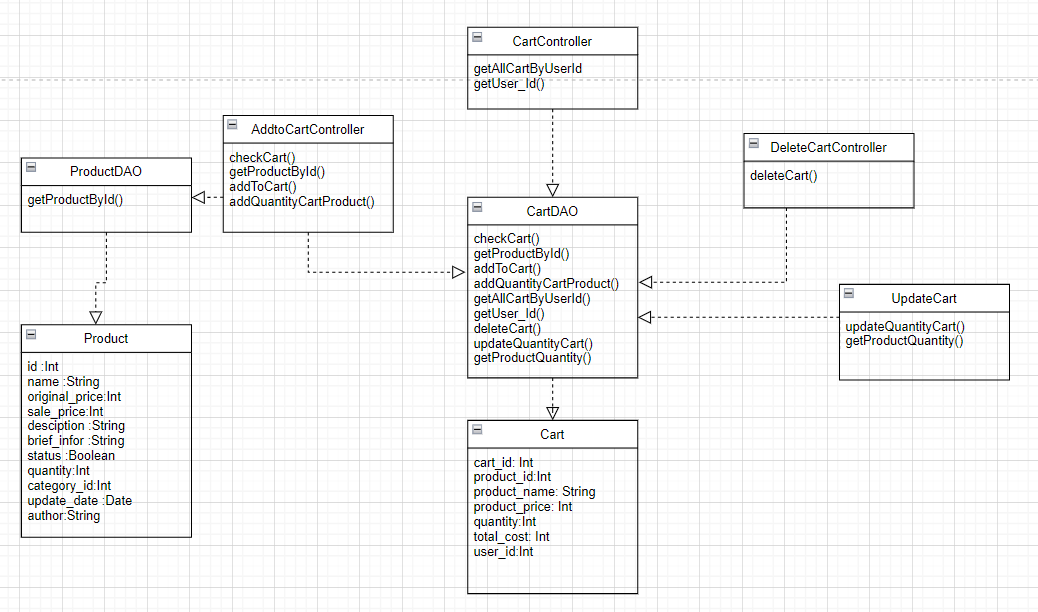


### d. Database queries

* "SELECT \* FROM Slider where status " + status;

## 13.Public/CartDetail

## *a. Class Diagram*



### b. Class Specifications

| **No** | **Method** | **Description** |
| --- | --- | --- |
| *01* | *checkCart()* | *check user and product exist* |
| *02* | *getProductById()* | *get product by id* |
| *03* | *addToCart()* | *Add product to cart* |
| *04* | *addQuantityCartProduct()* | *calculate total quantity and price when adding to cart* |
| *05* | *getAllCartByUserId()* | *get the products added to the cart by checking the user id* |
| *06* | *deleteCart()* | *remove products from cart* |
| *07* | *updateQuantityCart()* | *correct the number of products in the cart* |

### c. Sequence Diagram(s)

### 

### d. Database queries

* SELECT \* FROM Cart WHERE user\_id = ?
* SELECT \* FROM Cart WHERE user\_id = ? AND product\_id = ?
* "INSERT INTO Cart (product\_id, product\_name, product\_price, quantity, total\_cost, user\_id)"

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

* UPDATE Cart"

+ " SET quantity = ?, total\_cost = ?"

+ " WHERE product\_id = ? AND user\_id = ?

* DELETE FROM Cart"

+ " WHERE product\_id = ? AND user\_id = ?

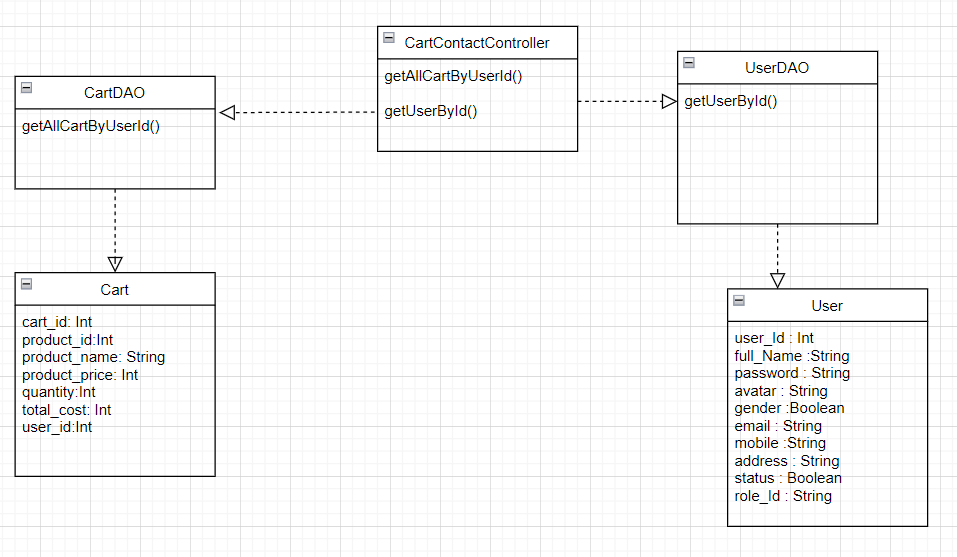
* UPDATE Cart"

+ " SET quantity = ? "

+ " WHERE cart\_id = ? AND product\_id = ?

## 14.Public/Cart Contact

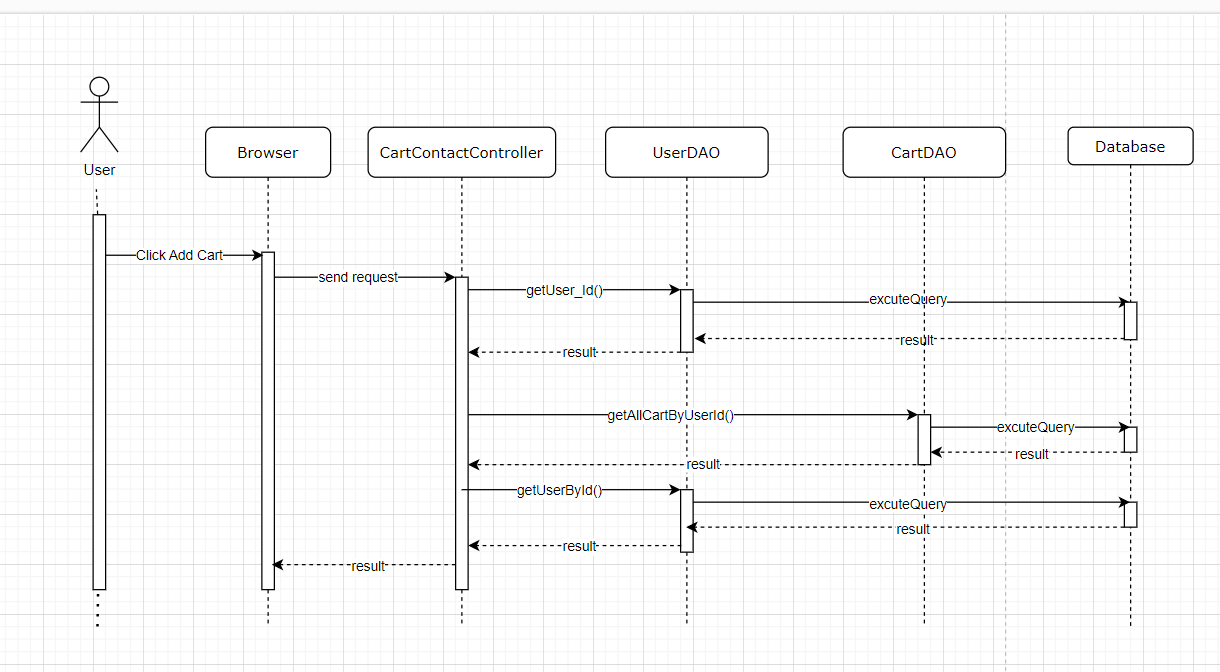
### a. Class Diagram



### b. Class Specifications

| **No** | **Method** | **Description** |
| --- | --- | --- |
| *01* | *getAllCartByUserId()* | *get the products added to the cart by checking the user id* |
| *02* | *getUserById()* | *get user info by checking user id* |

### c. Sequence Diagram(s)

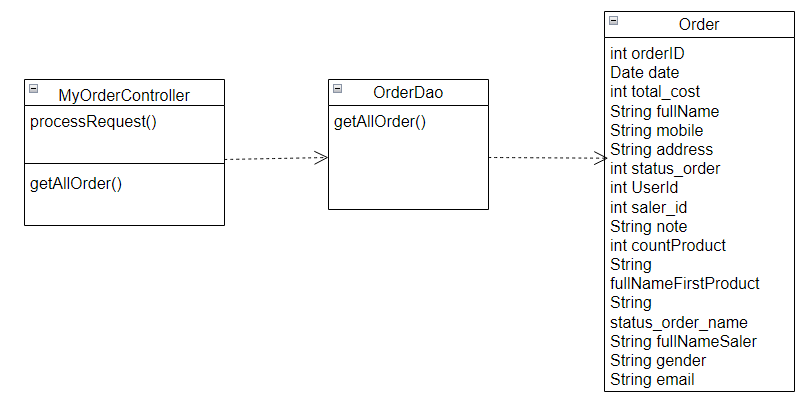


### d. Database queries

* SELECT \* FROM Cart WHERE user\_id = ?
* SELECT \* FROM User WHERE userId = ?

## 15.My order

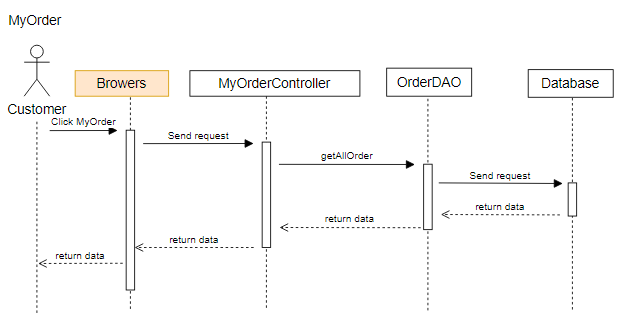
### a. Class Diagram



### b. Class Specifications

| **No** | **Method** | **Description** |
| --- | --- | --- |
| *01* | *getAllOrder()* | *Get all Order and status\_order in database* |

### c. Sequence Diagram(s)



### d. Database queries

* "SELECT \*\n"

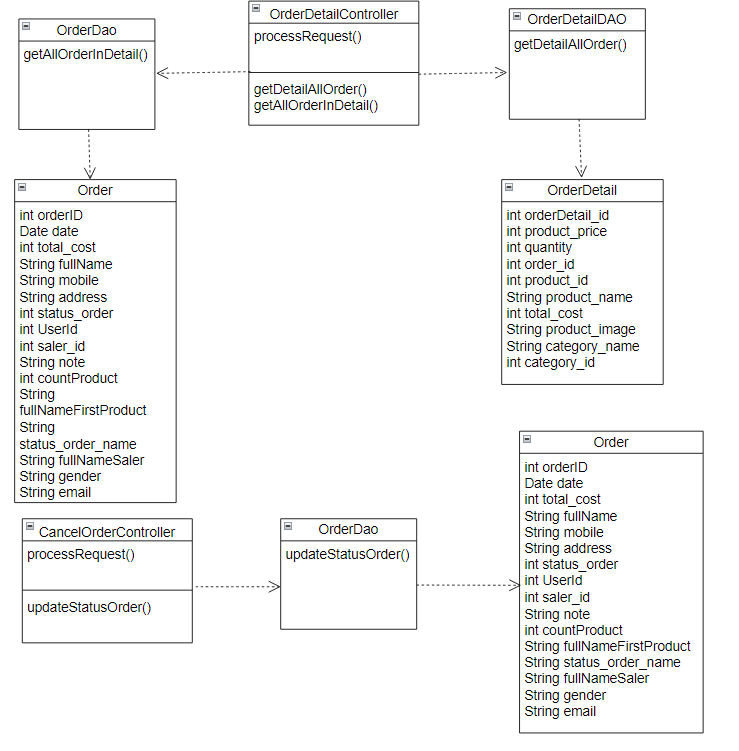
+ "FROM `Order`\n"

+ "JOIN Status\_Order ON `Order`.status\_order = Status\_Order.status\_order\_id\n"

+ "WHERE userId = ?";

## 16.Order Information

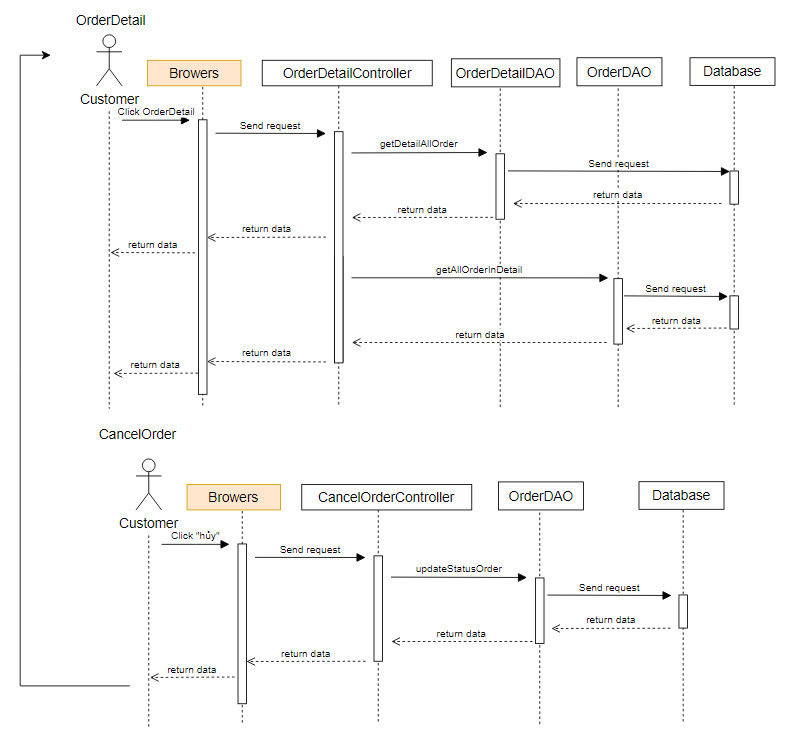
### a. Class Diagram



### b. Class Specifications

| **No** | **Method** | **Description** |
| --- | --- | --- |
| *01* | *getDetailAllOrder()* | *Get all Order\_Detail ,Product ,Category,Products\_Images in database* |
| *02* | *getAllOrderInDetail()* | *Get all Orders and status\_order and part user in database* |
| *03* | *updateStatusOrder()* | *update Status Order* |

### c. Sequence Diagram(s)



### d. Database queries

* String sql = "SELECT COUNT(\*)\n"

+ "FROM Order\_Detail\n"

+ "JOIN Product ON Order\_Detail.product\_id = Product.product\_id\n"

+ "WHERE order\_id = ?";

* String sql = "SELECT Product.product\_name\n"

+ "FROM Order\_Detail\n"

+ "JOIN Product ON Order\_Detail.product\_id = Product.product\_id\n"

+ "WHERE order\_id = ?\n"

+ "LIMIT 1";

* SELECT \* FROM books\_shop\_online.Order\_Detail \n"

+ "JOIN books\_shop\_online.Product ON Order\_Detail.product\_id = Product.product\_id\n"

+ "JOIN books\_shop\_online.Products\_Images ON Order\_Detail.product\_id = Products\_Images.product\_id\n"

+ "JOIN books\_shop\_online.Category ON Product.category\_id = Category.category\_id\n"

+ "WHERE order\_id = ?;

* String sql = "SELECT \*FROM books\_shop\_online.Order\n"

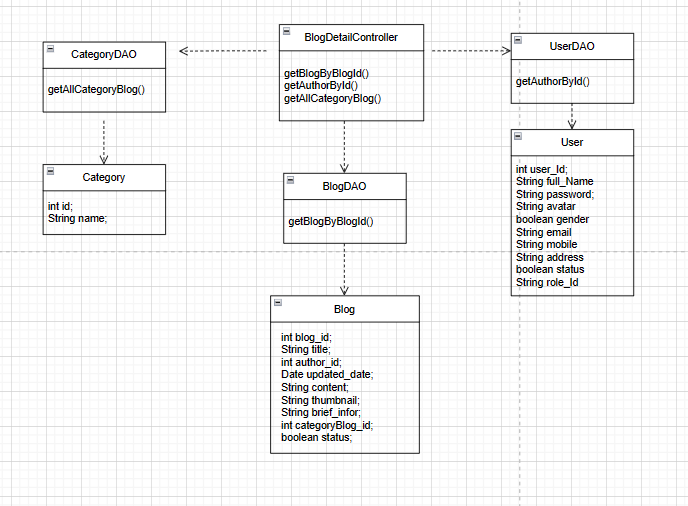
+ " JOIN books\_shop\_online.Status\_Order ON books\_shop\_online.Order.status\_order = books\_shop\_online.Status\_Order.status\_order\_id\n"

+ " join books\_shop\_online.user on books\_shop\_online.Order.userId=books\_shop\_online.user.userId\n"

+ " WHERE order\_id = ?";

## 17.Public/BlogDetail

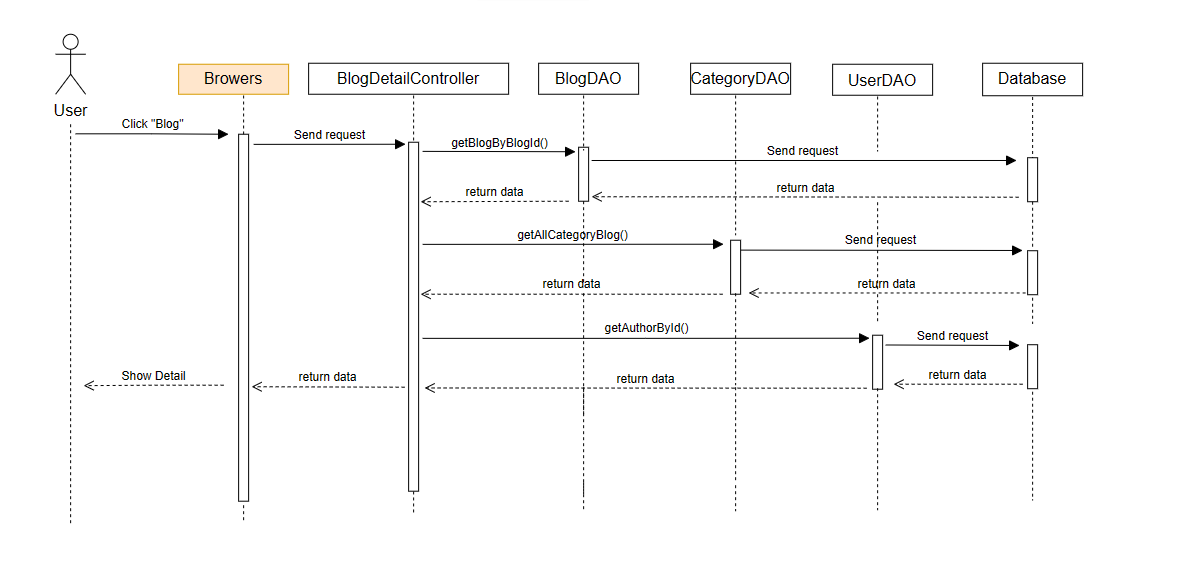
### a. Class Diagram



### b. Class Specifications

| **No** | **Method** | **Description** |
| --- | --- | --- |
| *01* | *getBlogByBlogId()* | *Get the Blog of by IIdd* |
| *02* | *getAuthorById()* | *Get all Author in database* |
| *03* | *getAllCategoryBlog()* | *Get all category in database* |

### c. Sequence Diagram(s)

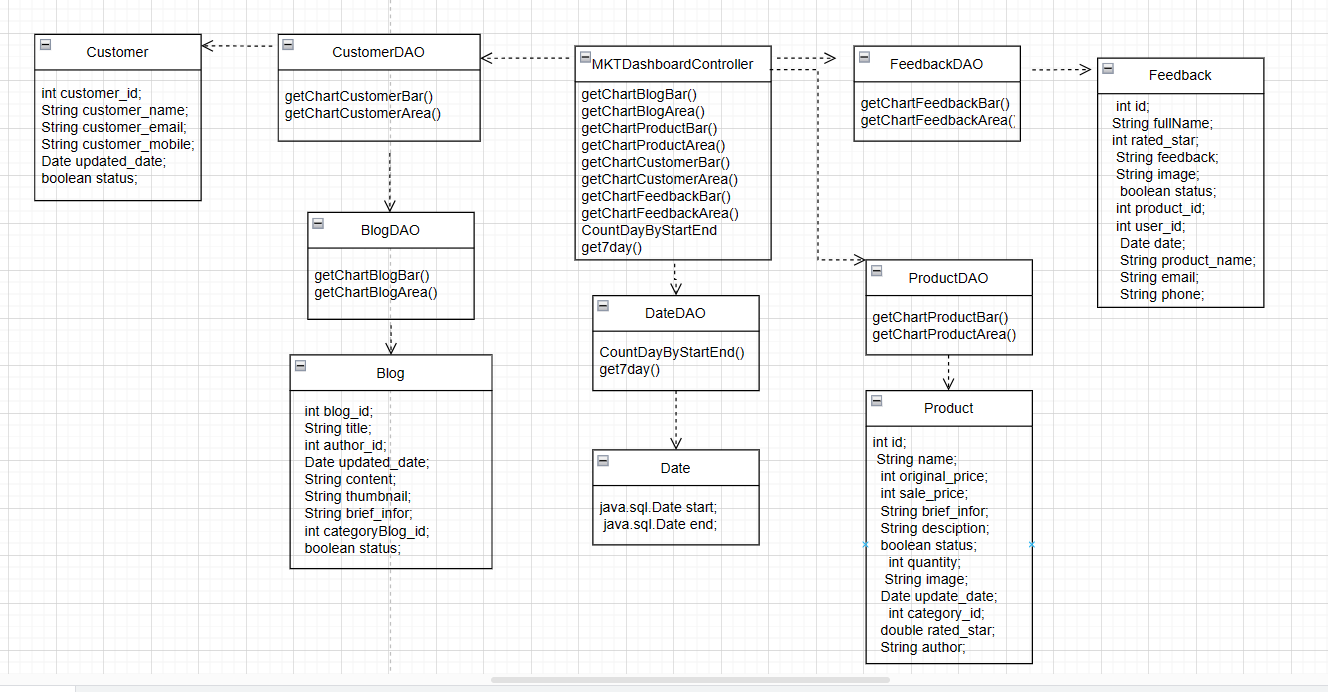


### d. Database queries

* Select \* from books\_shop\_online.blog \n"
* + "where blog\_id = ?
* Select \* from books\_shop\_online.category\_blog where status = 1
* select \* from books\_shop\_online.User where userId = ?

## 18.Marketing/MKT Dashboard

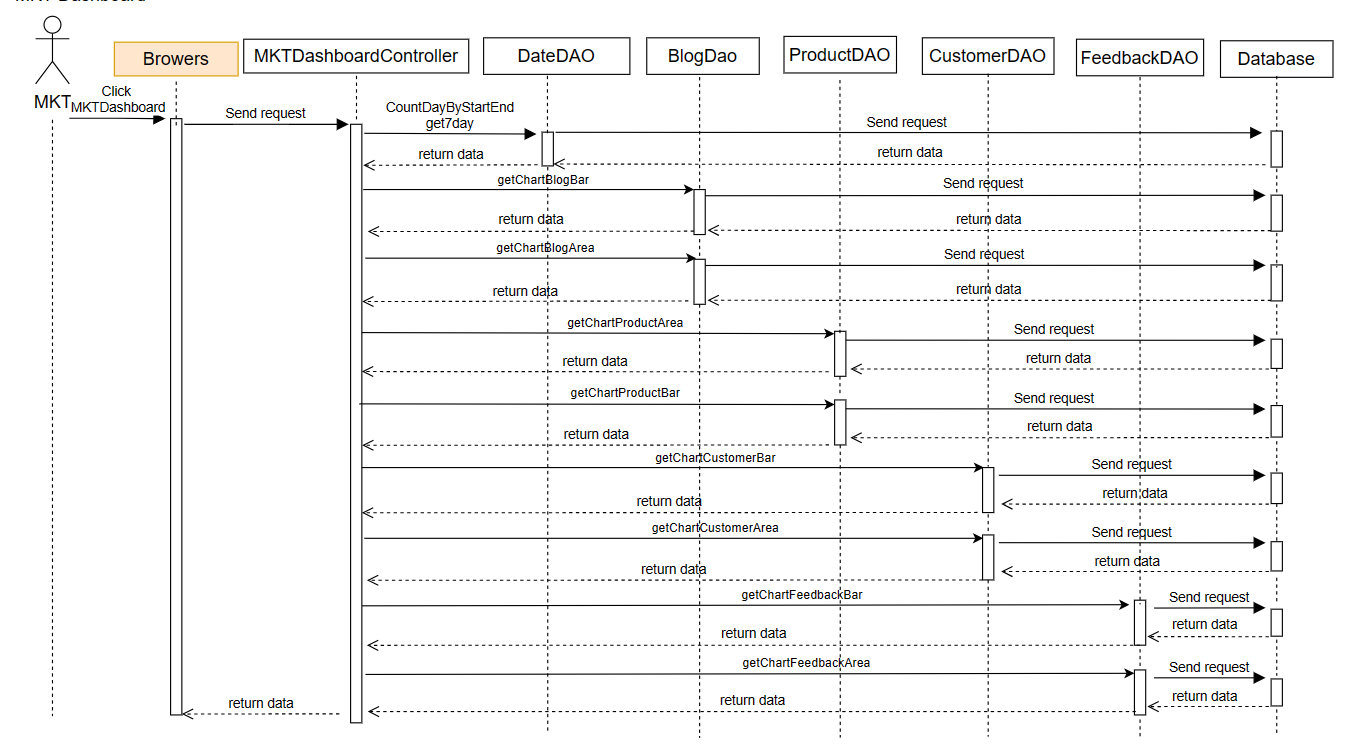
### a. Class Diagram



### b. Class Specifications

| **No** | **Method** | **Description** |
| --- | --- | --- |
| *01* | *getChartBlogBar()* | *get chart blog bar* |
| *02* | *getChartBlogArea()* | *get chart blog area* |
| *03* | *getChartProductBar()* | *get Chart Product Bar* |
| *04* | *getChartProductArea()* | *get Chart Product Area* |
| *05* | *get7day()* | *get 7 day* |
| *06* | *CountDayByStartEnd()* | *Count Day By Start End* |
| *07* | *getChartCustomerBar()* | *get Chart Customer Bar* |
| *08* | *getChartCustomerArea()* | *get Chart Customer Area* |
| *09* | *getChartFeedbackBar()* | *get Chart Feedback Bar* |
| *10* | *getChartFeedbackArea()* | *get Chart Feedback Area* |

### c. Sequence Diagram(s)



### d. Database queries

SELECT COUNT(\*) FROM books\_shop\_online.Blog WHERE updated\_date = DATE\_ADD(?, INTERVAL ? DAY)AND status = 1;

SELECT COUNT(\*) FROM Blog WHERE updated\_date <= DATE\_ADD(?, INTERVAL ? DAY) AND status = 1;

SELECT COUNT(\*) FROM books\_shop\_online.product WHERE update\_date = DATE\_ADD(?, INTERVAL ? DAY ) AND status = 1;

SELECT COUNT(\*) FROM product WHERE update\_date <= DATE\_ADD(?, INTERVAL ? DAY) AND status = 1;

SELECT COUNT(\*) FROM books\_shop\_online.customer WHERE updated\_date = DATE\_ADD(?, INTERVAL ? DAY)

SELECT COUNT(\*) FROM books\_shop\_online.customer WHERE updated\_date <= DATE\_ADD(?, INTERVAL ? DAY)

SELECT COUNT(\*) FROM Feedback WHERE `date` = DATE\_ADD(?, INTERVAL ? DAY) AND `status` = 1;

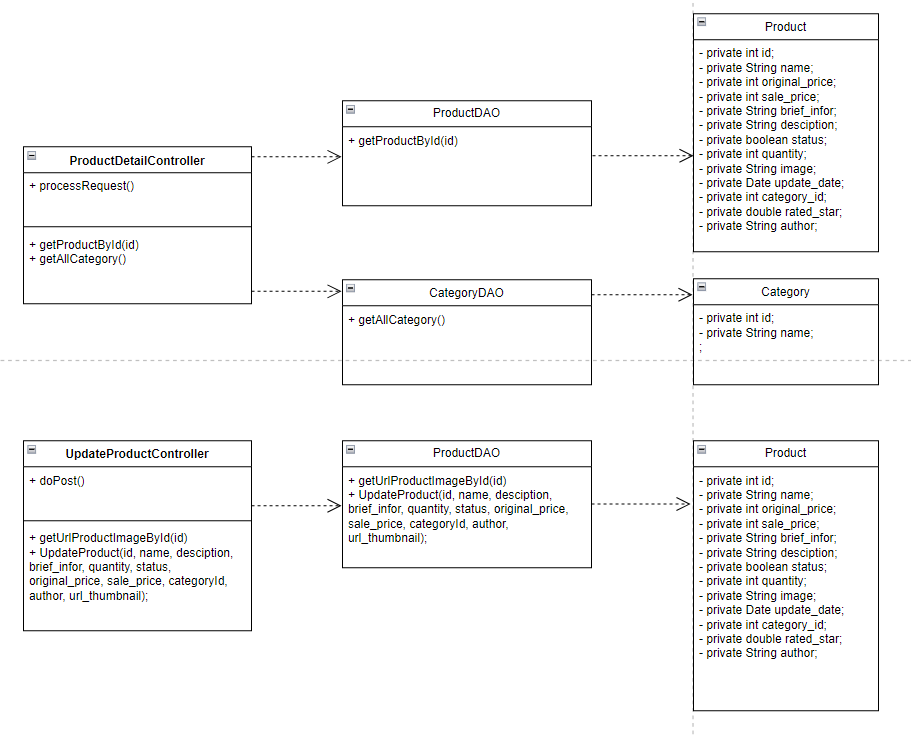
SELECT COUNT(\*) FROM Feedback WHERE `date` <= DATE\_ADD(?, INTERVAL ? DAY) AND `status` = 1;

SELECT CURRENT\_TIMESTAMP, DATE\_SUB(CURRENT\_TIMESTAMP, INTERVAL 6 DAY);

SELECT DATEDIFF(DATE(?), DATE(?));

## 19.Marketing/MKT Product Detail

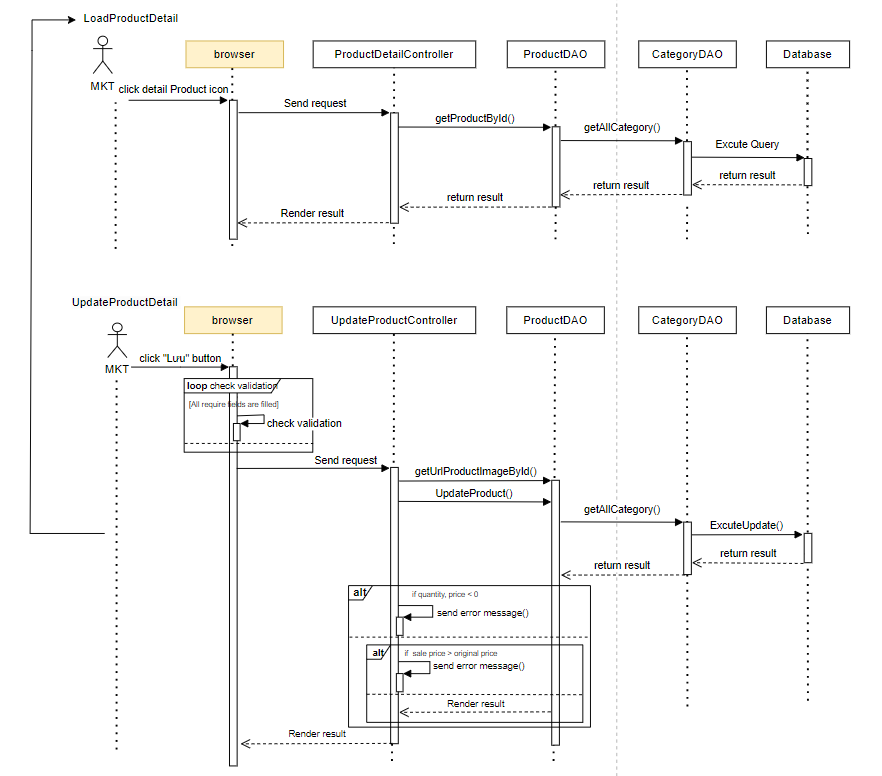
### a. Class Diagram



### b. Class Specifications

| **No** | **Method** | **Description** |
| --- | --- | --- |
| *01* | *getProductById()* | *get detail Product to the screen* |
| *02* | *getAllCategory()* | *get all category show to the screen* |
| *03* | *getUrlProductImageById()* | *get url of product image show to the screen* |
| *04* | *UpdateProduct()* | *update the detail of product show to the screen* |

### c. Sequence Diagram(s)



### d. Database queries

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

- Select \* from books\_shop\_online.category where status = 1

- SELECT images FROM Products\_Images where product\_id = ?

- UPDATE 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"

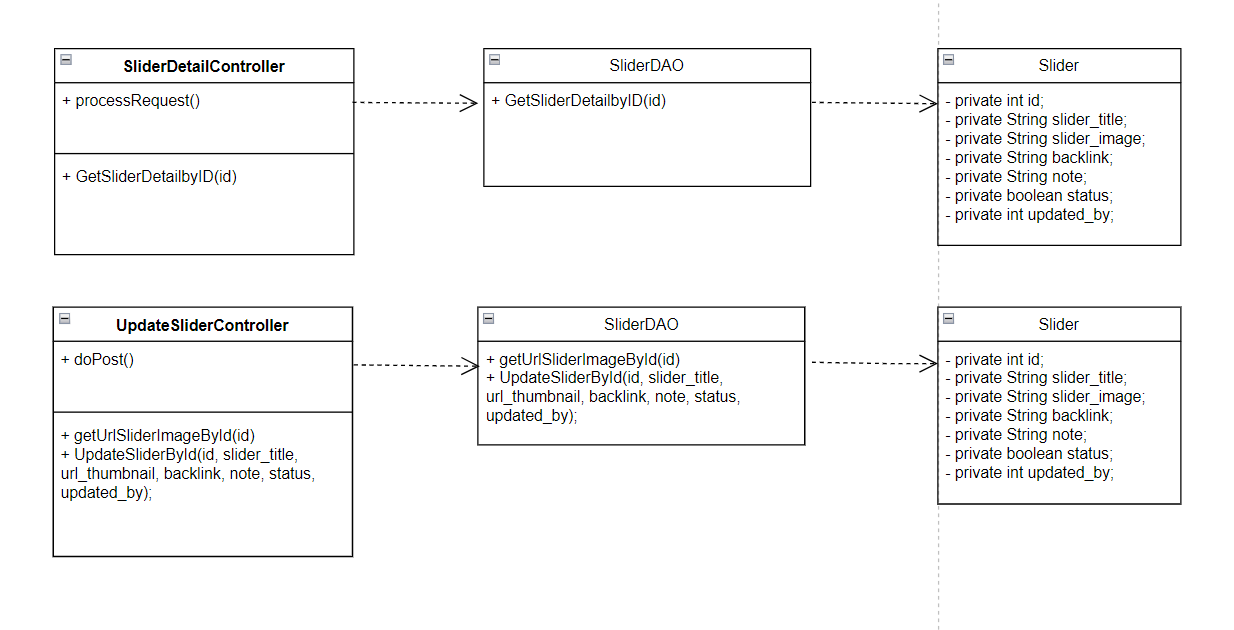
+ " ,`author` = ?\n"

+ " ,`update\_date` = NOW()\n"

+ " WHERE `product\_id` = ?

## 20.Marketing/MKT Slider Detail

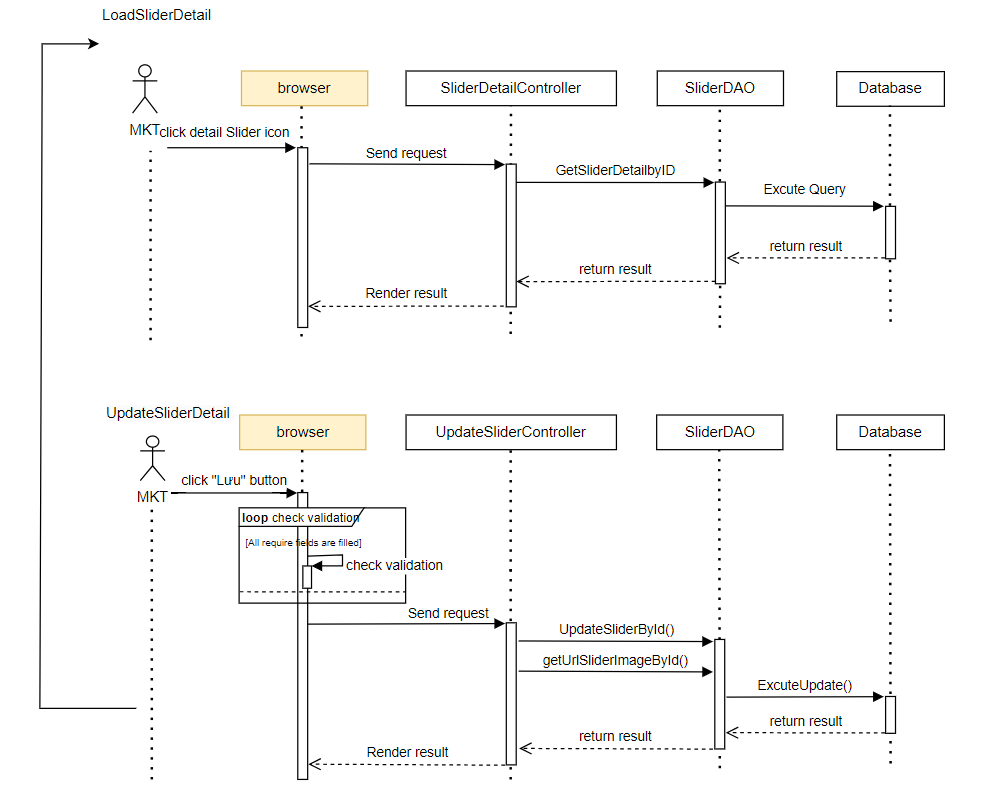
### a. Class Diagram



### b. Class Specifications

| **No** | **Method** | **Description** |
| --- | --- | --- |
| *01* | *GetSliderDetailbyID()* | *get detail Slider to the screen* |
| *02* | *getUrlSliderImageById()* | *get url of slider image show to the screen* |
| *03* | *UpdateSliderById()* | *update the detail of slider show to the screen* |

### c. Sequence Diagram(s)



### d. Database queries

- SELECT \* FROM Slider where slider\_id = ?

- select slider\_image from Slider where slider\_id = ?

- UPDATE Slider\n"

+ " SET `slider\_title` = ?\n"

+ " ,`slider\_image` = ?\n"

+ " ,`backlink` = ?\n"

+ " ,`note` = ?\n"

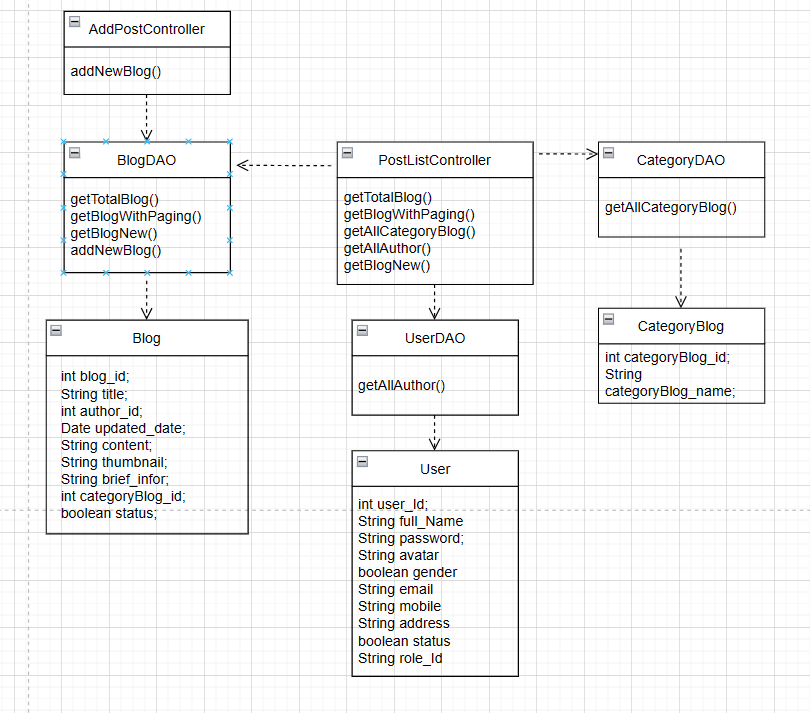
+ " ,`status` = ?\n"

+ " ,`updated\_by` = ? \n"

+ " WHERE `slider\_id` = ?

## 21.Marketing/Posts List

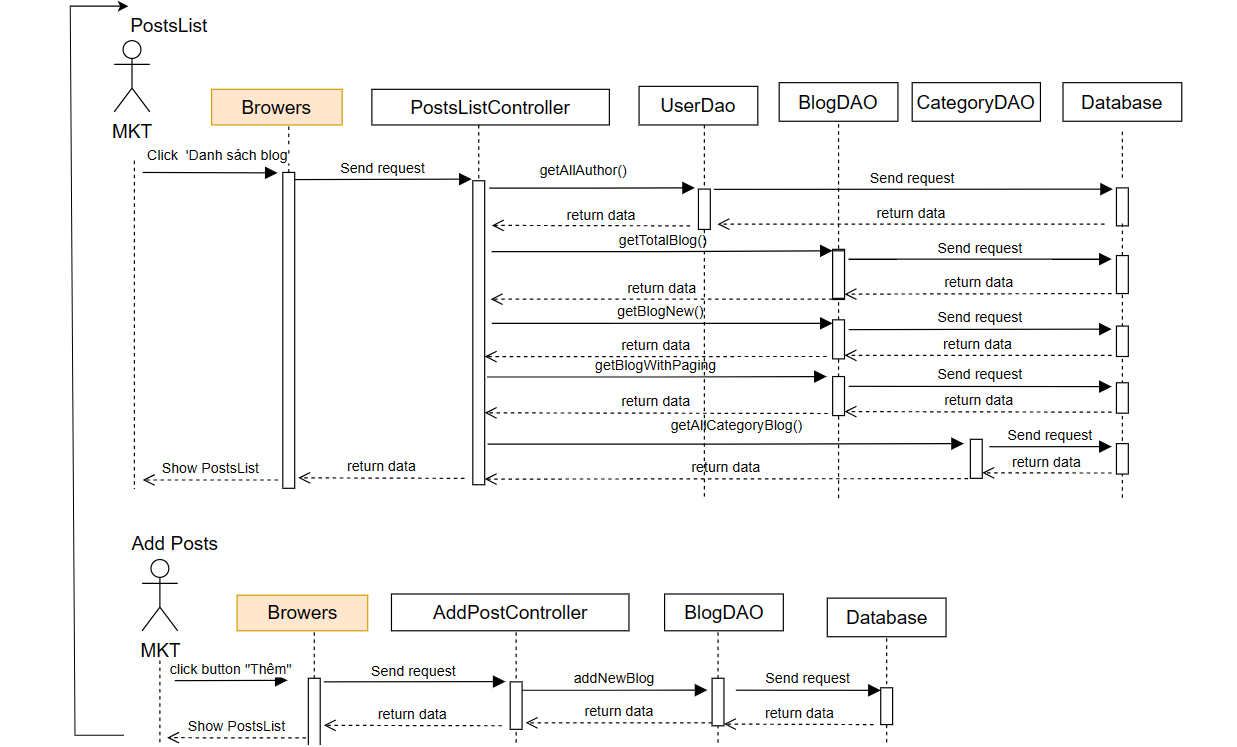
### a. Class Diagram



### b. Class Specifications

| **No** | **Method** | **Description** |
| --- | --- | --- |
| *01* | *getAllCategoryBlog()* | *get All Category Blog* |
| *02* | *getBlogWithPaging()* | *get Blog With Paging* |
| *03* | *getBlogNew()* | *Get blog new* |
| *04* | *getTotalBlog* | *get Total Blog* |
| *05* | *addNewBlog()* | *add New Blog* |
| *06* | *getAllAuthor()* | *get All Author* |

### c. Sequence Diagram(s)



### d. Database queries

SELECT COUNT(blog\_id) FROM blog\n"

+ "WHERE categoryBlog\_id " + categoryId + " AND status " + status + " AND title LIKE '%" + searchKey + "%';

select \* from books\_shop\_online.blog\n"

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

+ " order by " + value + " " + type + " LIMIT ?, ?;

Select \* from books\_shop\_online.category\_blog where status = 1

select \* from User where role\_id = 2  
SELECT \*\n"

+ "FROM books\_shop\_online.blog\n"

+ "WHERE status = 1\n"

+ "ORDER BY updated\_date DESC\n"

+ "LIMIT 1;

INSERT INTO Blog\n"

+ " (title\n"

+ " ,author\_id\n"

+ " ,content\n"

+ " ,thumbnail\n"

+ " ,brief\_infor\n"

+ " ,categoryBlog\_id\n"

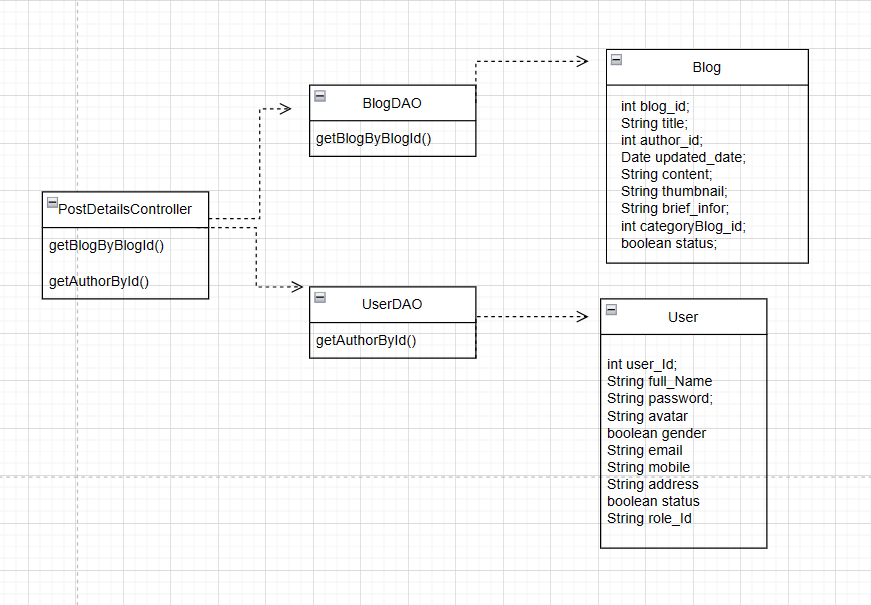
+ " ,status)\n"

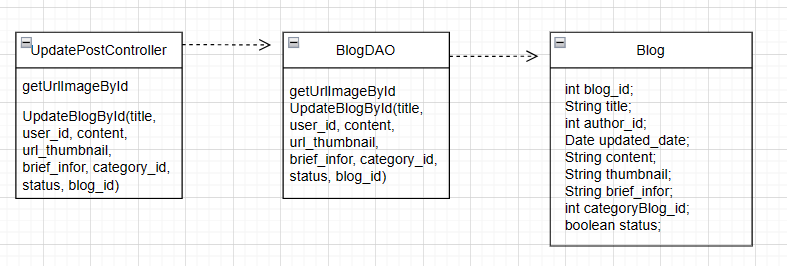
+ " VALUES\n"

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

## 22.Marketing/Posts Detail

### a. Class Diagram

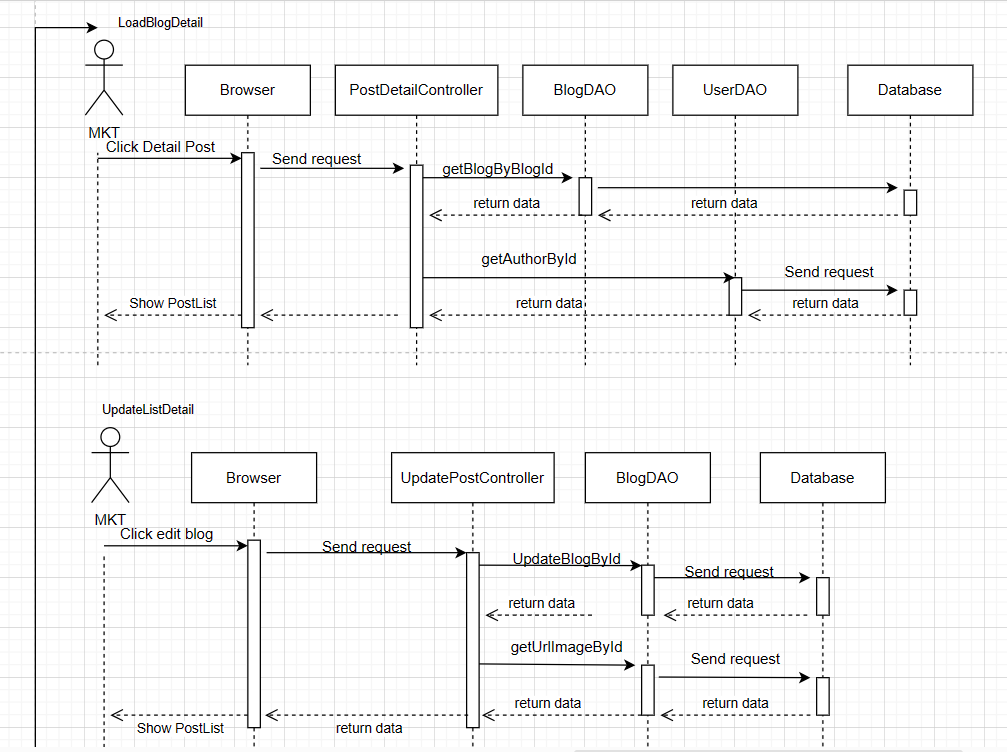




### b. Class Specifications

| **No** | **Method** | **Description** |
| --- | --- | --- |
| *01* | *getBlogByBlogId()* | *get Blog By Blog Id* |
| *02* | *getAuthorById()* | *get Author By Id* |
| *03* | *getUrlImageById()* | *get url of blog image show to the screen* |
| *04* | *UpdateBlogById()* | *update the detail of blog show to the screen* |

### c. Sequence Diagram(s)



### d. Database queries

Select \* from books\_shop\_online.blog \n"

+ "where blog\_id = ?

select \* from books\_shop\_online.User where userId = ?

UPDATE Blog\n"

+ "SET title = ?,\n"

+ " author\_id = ?,\n"

+ " updated\_date = CURRENT\_TIMESTAMP,\n"

+ " content = ?,\n"

+ " thumbnail = ?,\n"

+ " brief\_infor = ?,\n"

+ " categoryBlog\_id = ?,\n"

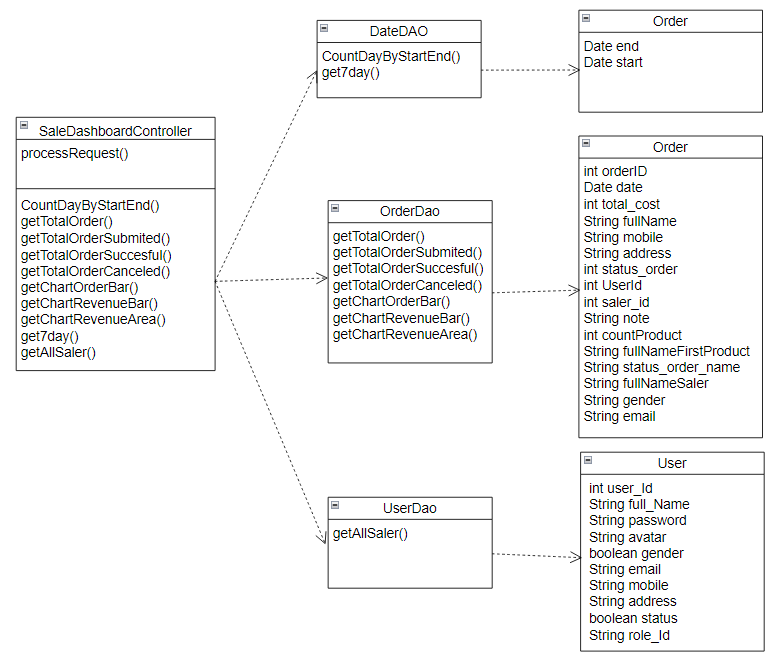
+ " status = ?\n"

+ "WHERE blog\_id = ?

select thumbnail from books\_shop\_online.blog where blog\_id = ?

## 23.Sale/Sale Dashboard

### a. Class Diagram



### b. Class Specifications

| **No** | **Method** | **Description** |
| --- | --- | --- |
| *01* | *CountDayByStartEnd()* | *Count Day By Start End* |
| *02* | *getTotalOrder()* | *get Total Order* |
| *03* | *getTotalOrderSubmited()* | *get Total Order Submitted* |
| *04* | *getTotalOrderSuccesful()* | *get Total Order Successful* |
| *05* | *getTotalOrderCanceled()* | *get Total Order Canceled* |
| *06* | *getChartOrderBar()* | *get Chart Order Bar* |
| *07* | *getChartRevenueBar()* | *get Chart Revenue Bar* |
| *08* | *getChartRevenueArea()* | *get Chart Revenue Area* |
| *09* | *get7day()* | *get 7 day* |
| *10* | *getAllSaler()* | *get All Saler* |

### c. Sequence Diagram(s)

### d. Database queries

- SELECT DATEDIFF(DATE(?), DATE(?));

- SELECT CURRENT\_TIMESTAMP, DATE\_SUB(CURRENT\_TIMESTAMP, INTERVAL 6 DAY);

- SELECT COUNT(\*) FROM `Order` \n"

+ "WHERE saler\_id " + salerId + " AND orderDate <= ? AND orderDate >= ?;

- SELECT COUNT(\*) FROM `Order`\n"

+ "WHERE saler\_id " + salerId + " AND orderDate <= ? AND orderDate >= ? AND status\_order =1;

- SELECT COUNT(\*) FROM `Order`\n"

+ "WHERE saler\_id " + salerId + " AND orderDate <= ? AND orderDate >= ? AND status\_order =2;

- SELECT COUNT(\*) FROM `Order`\n"

+ "WHERE saler\_id " + salerId + " AND orderDate <= ? AND orderDate >= ? AND status\_order =3;

- SELECT COUNT(\*) FROM `Order`\n"

+ "WHERE saler\_id " + salerId + " AND orderDate = DATE\_ADD(?, INTERVAL ? DAY);

- SELECT SUM(total\_cost) FROM `Order`\n"

+ "WHERE saler\_id " + salerId + " AND orderDate = DATE\_ADD(?, INTERVAL ? DAY);

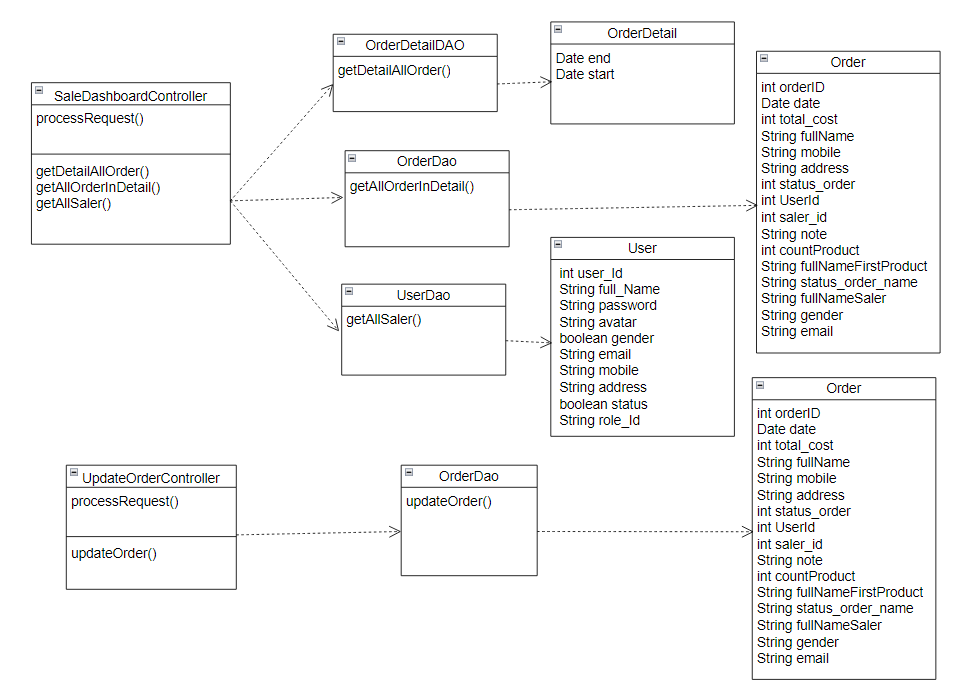
- SELECT SUM(total\_cost) FROM `Order`\n"

+ "WHERE saler\_id " + salerId + " AND orderDate <= DATE\_ADD(?, INTERVAL ? DAY) AND orderDate >= ?;

- SELECT \* FROM `User` WHERE role\_id = 3;

## 24.Sale/Orders Details

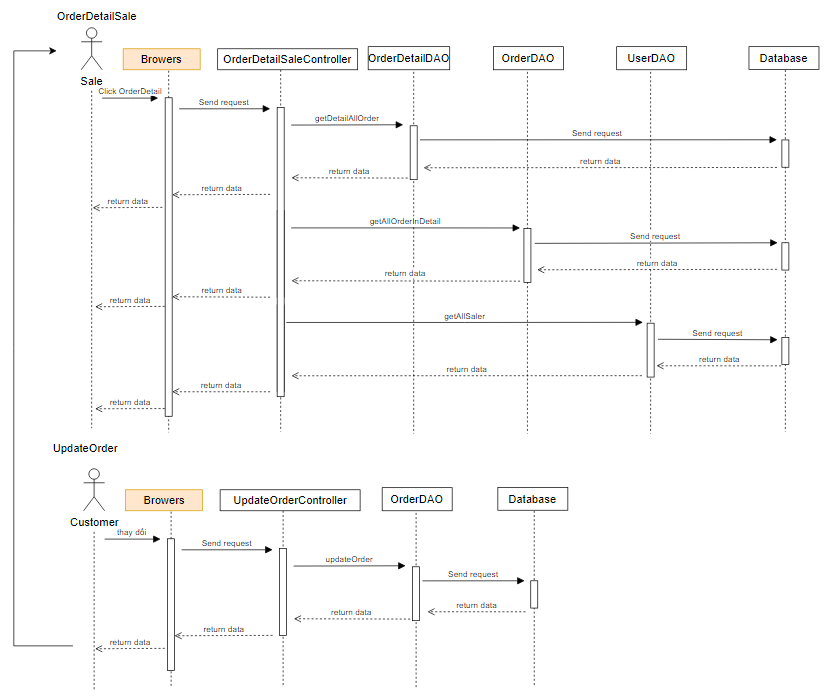
### a. Class Diagram



### b. Class Specifications

| **No** | **Method** | **Description** |
| --- | --- | --- |
| *01* | *getDetailAllOrder()* | *get Detail All Order* |
| *02* | *getAllOrderInDetail()* | *get All Order In Detail* |
| *03* | *getAllSaler()* | *get All Saler* |
| *04* | *updateOrder()* | *update Order* |

### c. Sequence Diagram(s)



### d. Database queries

- SELECT \*\n"

+ "FROM books\_shop\_online.Order\_Detail\n"

+ "JOIN books\_shop\_online.Product ON books\_shop\_online.Order\_Detail.product\_id = books\_shop\_online.Product.product\_id\n"

+ "JOIN books\_shop\_online.Products\_Images ON books\_shop\_online.Order\_Detail.product\_id = books\_shop\_online.Products\_Images.product\_id\n"

+ "JOIN books\_shop\_online.Category ON books\_shop\_online.Product.category\_id = books\_shop\_online.Category.category\_id\n"

+ "WHERE books\_shop\_online.Order\_Detail.order\_id = ?

- SELECT \*FROM books\_shop\_online.Order\n"

+ " JOIN books\_shop\_online.Status\_Order ON books\_shop\_online.Order.status\_order = books\_shop\_online.Status\_Order.status\_order\_id\n"

+ " join books\_shop\_online.user on books\_shop\_online.Order.userId=books\_shop\_online.user.userId\n"

+ " WHERE order\_id = ?

- SELECT COUNT(\*)\n"

+ "FROM Order\_Detail\n"

+ "JOIN Product ON Order\_Detail.product\_id = Product.product\_id\n"

+ "WHERE order\_id = ?

- SELECT Product.product\_name\n"

+ "FROM Order\_Detail\n"

+ "JOIN Product ON Order\_Detail.product\_id = Product.product\_id\n"

+ "WHERE order\_id = ?\n"

+ "LIMIT 1

- SELECT \* FROM `User` WHERE role\_id = 3;

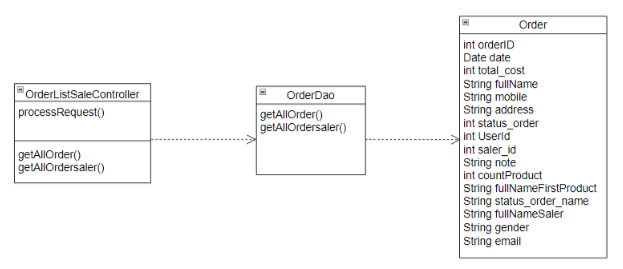
- "UPDATE `Order`\n"

+ "SET `status\_order` = ?, `saler\_id` = ?,`note` = ?\n"

+ "WHERE `order\_id` = ?; "

## 25.Sale/Order List

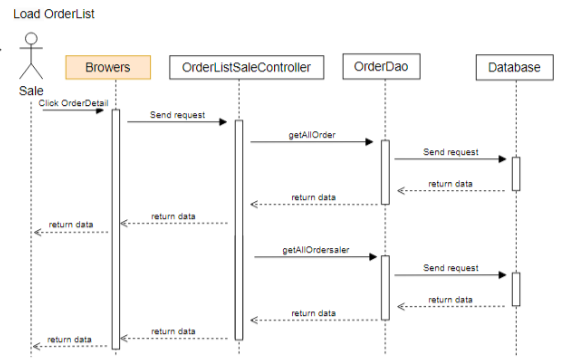
### a. Class Diagram



### b. Class Specifications

| **No** | **Method** | **Description** |
| --- | --- | --- |
| *01* | *getAllOrder()* | *get All Order* |
| *02* | *getAllOrdersaler()* | *get All Order according to seller* |

### c. Sequence Diagram(s)



### d. Database queries

- SELECT \* FROM `Order`\n"

+ "JOIN Status\_Order ON `Order`.status\_order = Status\_Order.status\_order\_id\n"

+ "JOIN `User` ON `User`.userId = `Order`.saler\_id;

- SELECT \* FROM `Order`\n"

+ "JOIN Status\_Order ON `Order`.status\_order = Status\_Order.status\_order\_id\n"

+ "JOIN `User` ON `User`.userId = `Order`.saler\_id where saler\_id=?;

## 26.Admin/User List

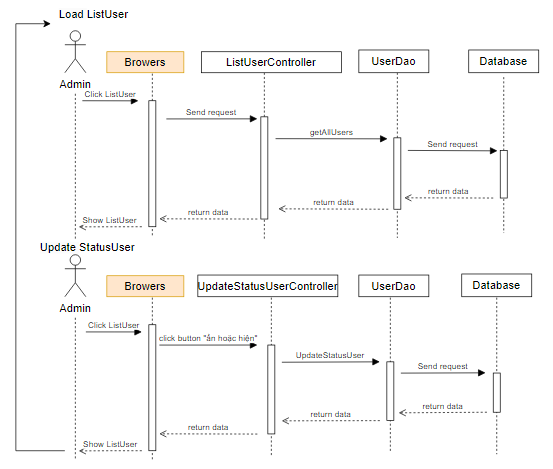
### a. Class Diagram



### b. Class Specifications

| **No** | **Method** | **Description** |
| --- | --- | --- |
| *01* | *getAllUsers()* | *get All Users* |
| *02* | *UpdateStatusUser()* | *Update Status User* |

### c. Sequence Diagram(s)



### d. Database queries

- "SELECT \*\n"

+ "FROM `User`;"

- "UPDATE `User`\n"

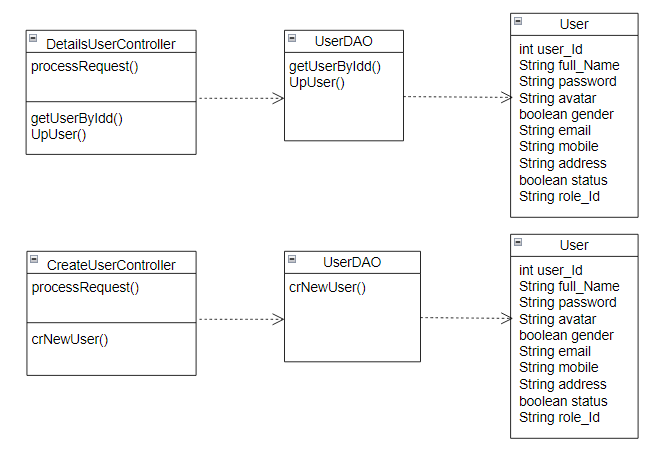
+ "SET `status` = ?\n"

+ "WHERE `userId` = ?;"

## 

## 27.Admin/User Details

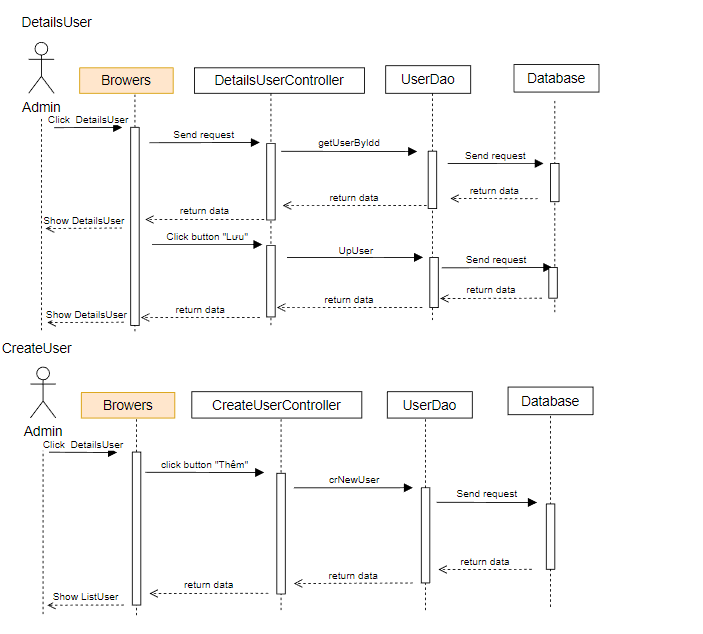
### a. Class Diagram



### b. Class Specifications

| **No** | **Method** | **Description** |
| --- | --- | --- |
| *01* | *getUserByIdd()* | *get User By Id* |
| *02* | *UpUser()* | *Update user* |
| *03* | *crNewUser()* | *Create New User* |

### c. Sequence Diagram(s)



### d. Database queries

- "INSERT INTO `books\_shop\_online`.`user` (`fullName`, `password`, `gender`, `email`, `mobile`, `address`, `status`, `role\_id`) VALUES (?,?,?,?,?,?,0,?);"

- "select \* from books\_shop\_online.user where userId = ?"

- "UPDATE `books\_shop\_online`.`user`\n"

+ "SET `fullName` = ?,\n"

+ " `password` = ?,\n"

+ " `gender` = ?,\n"

+ " `email` = ?,\n"

+ " `mobile` = ?,\n"

+ " `address` = ?,\n"

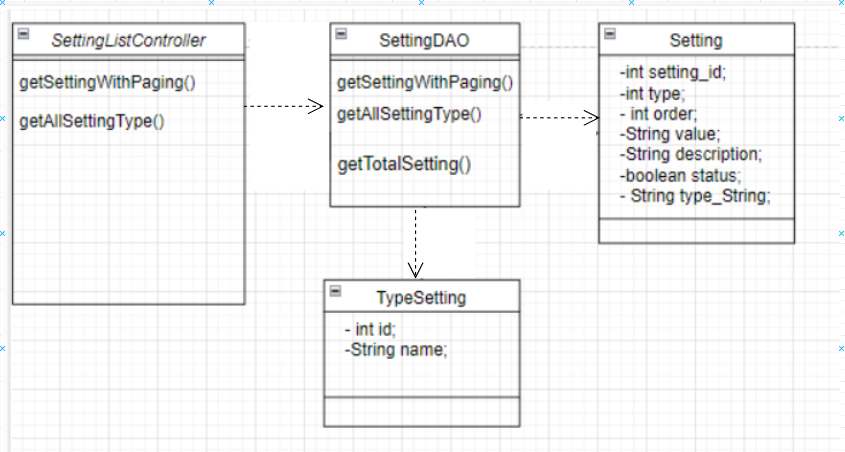
+ " `status` = ?,\n"

+ " `role\_id` = ?\n"

+ "WHERE `userId` = ?;"

## 28.Admin/Setting List

### a. Class Diagram



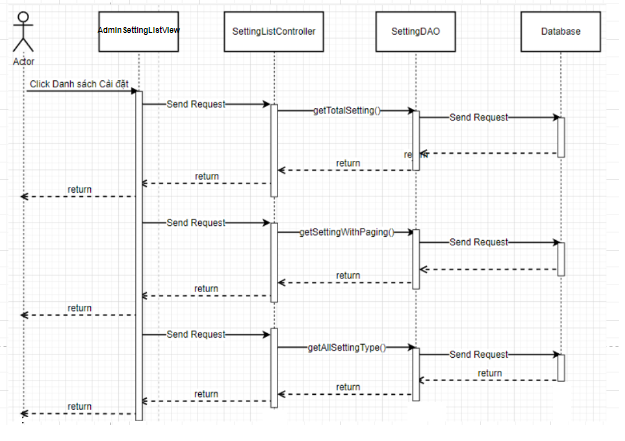
### 

### 

### b. Class Specifications

| **No** | **Method** | **Description** |
| --- | --- | --- |
| *01* | *getSettingWithPaging()* | *setting paging* |
| *02* | *getAllSettingType()* | *get all setting type* |
| *03* | *getTotalSetting()* | *count all setting follow keysearch input* |

### c. Sequence Diagram(s)



### d. Database queries

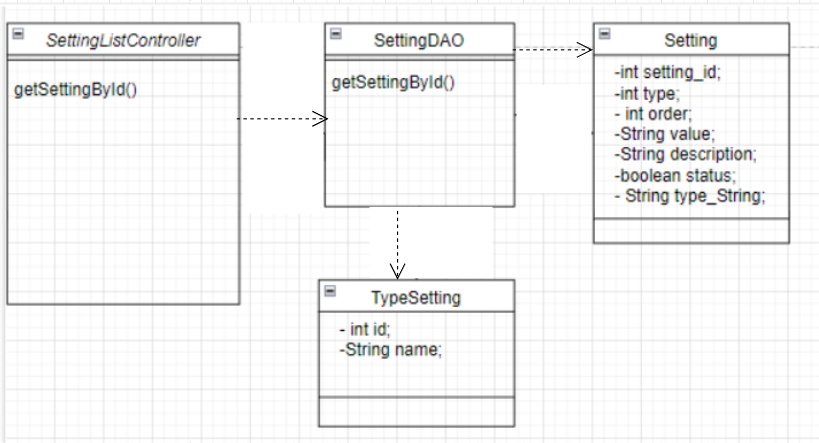
-SELECT \* FROM `setting` where `type` " + typeId + " and `status` " + status + " and `value` like N'%" + searchKey + "%' order by " + value + " " + type + " LIMIT ?, ?;

-select \* from setting\_type

-SELECT COUNT(\*) FROM `setting` where `type` " + typeId + " and `status` " + status + " and `value` like N'%" + searchKey + "%';

## 29.Admin/Setting Details

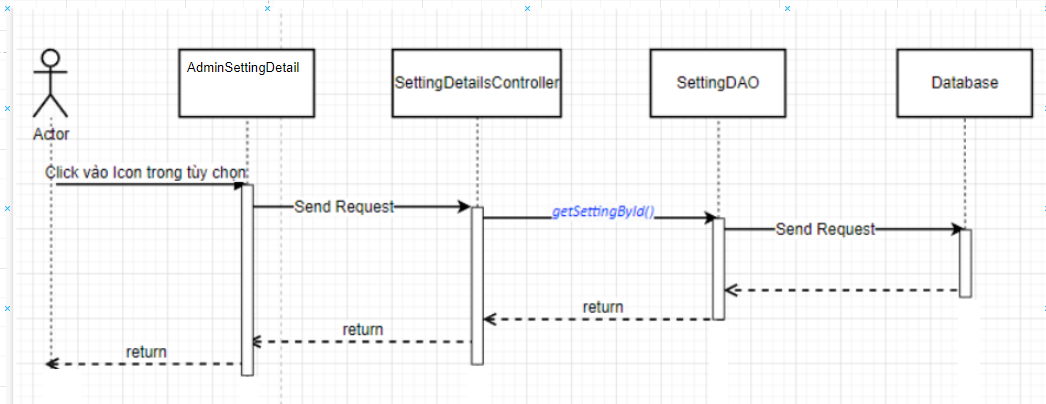
### a. Class Diagram



### b. Class Specifications

| **No** | **Method** | **Description** |
| --- | --- | --- |
| *01* | *getSettingById()* | *get setting by ID* |

### c. Sequence Diagram(s)

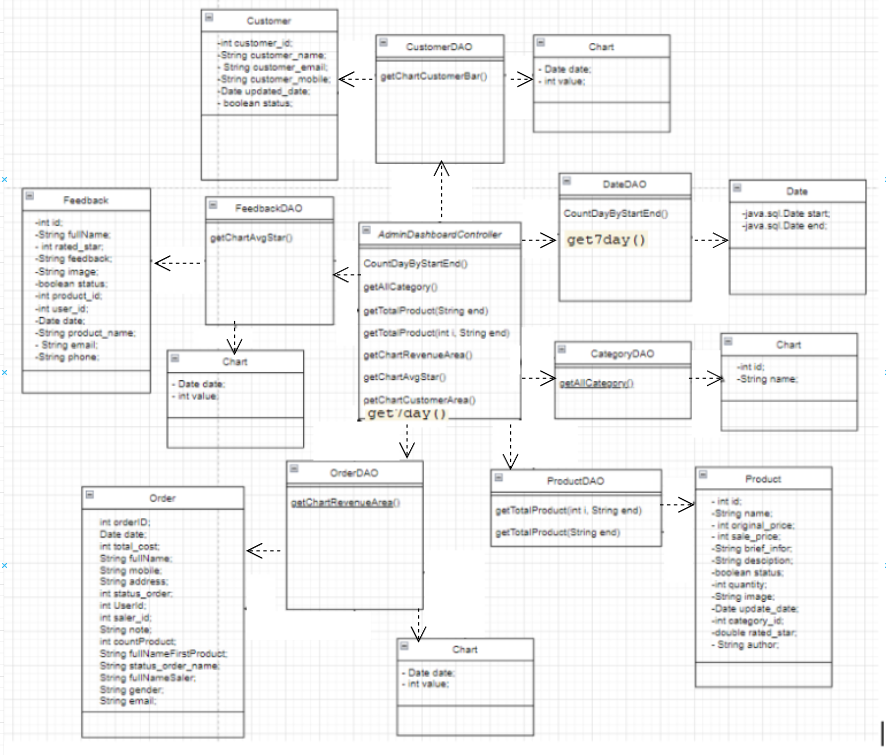


### d. Database queries

-select \* from setting s join setting\_type st on s.type = st.setting\_type\_id where setting\_id = ?

## 30.Admin/Admin DashBoard

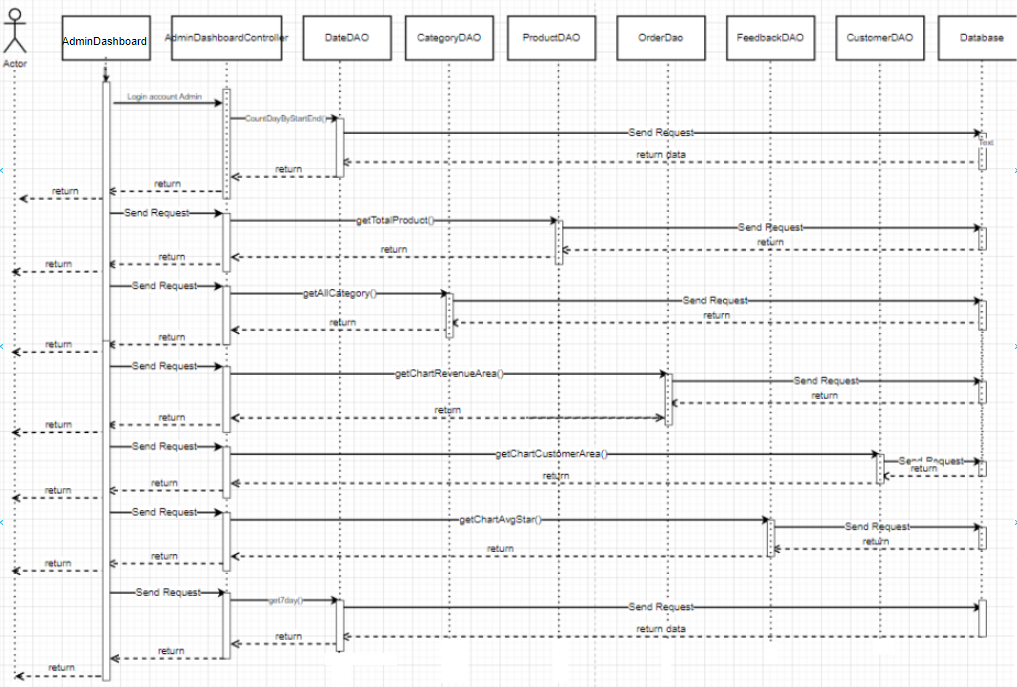
### a. Class Diagram



### b. Class Specifications

| **No** | **Method** | **Description** |
| --- | --- | --- |
| *01* | *CountDayByStartEnd()* | *calculate the number of days difference between two date values.* |
| *02* | *getAllCategory()* | *get all Category* |
| *03* | *getTotalProduct(String end)* | *count all product have update\_date < ended* |
| *04* | *getTotalProduct(int i, String end)* | *count all product have update\_date < ended with category\_id* |
| *05* | *getChartRevenueArea()* | *calculate sum (total\_cost)* |
| *06* | *getChartAvgStar()* | *average of values in rated\_star column* |
| *07* | *getChartCustomerArea()* | *COUNT all Customer in condition date* |
| *08* | *get7day()* | *subtract 6 days from current time value* |

### c. Sequence Diagram(s)



### d. Database queries

-SELECT DATEDIFF(DATE(?), DATE(?));

-select COUNT(\*) from Product where update\_date < ?;

-select COUNT(\*) from Product where category\_id = ? and update\_date < ?;

-Select \* from books\_shop\_online.category where status = 1

-select sum(total\_cost) from `Order` where saler\_id " + salerId + " and orderDate <= DATE\_ADD(?, INTERVAL ? DAY) and orderDate >= ?;

-SELECT DATE\_ADD(?, INTERVAL ? DAY);

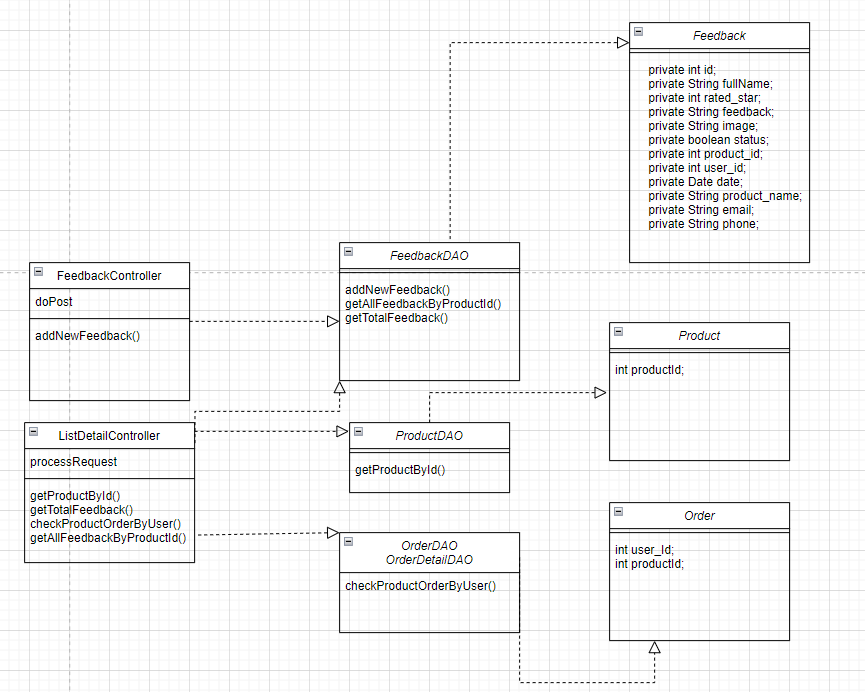
-SELECT COUNT(\*) FROM books\_shop\_online.customer WHERE updated\_date <= DATE\_ADD(?, INTERVAL ? DAY)

-SELECT CAST(AVG(rated\_star) AS DECIMAL(10,1)) FROM Feedback WHERE date < DATE\_ADD(?, INTERVAL ? DAY);

-SELECT CURRENT\_TIMESTAMP, DATE\_SUB(CURRENT\_TIMESTAMP, INTERVAL 6 DAY);

## 31.Marketing/Feedback List

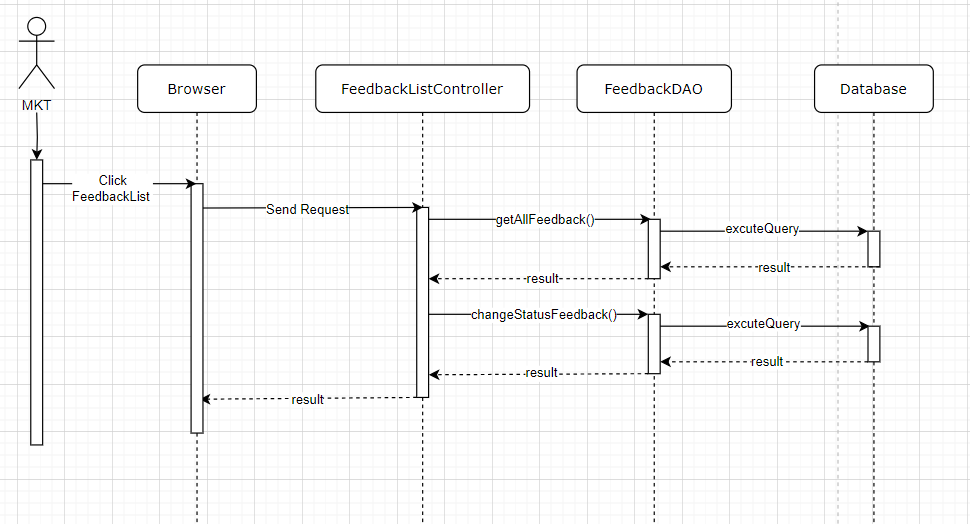
### a. Class Diagram



### b. Class Specifications

| **No** | **Method** | **Description** |
| --- | --- | --- |
| *01* | *getAllFeedback()* | *get list of user feedback* |
| *02* | *changeStatusFeedback()* | *change the status of user feedback* |

### c. Sequence Diagram(s)



### d. Database queries

- SELECT f.\*, p.product\_name \n" +

"FROM Feedback f \n" +

"JOIN Product p ON f.product\_id = p.product\_id;

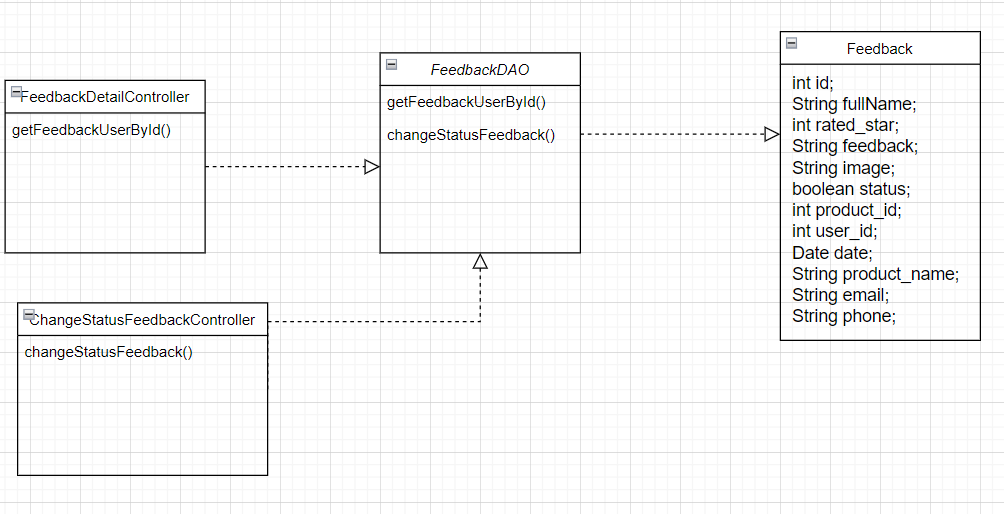
- UPDATE `Feedback` "

+ "SET `status` = ? "

+ "WHERE `feedBack\_id` = ?

## 32.Marketing/Feedback Detail

### a. Class Diagram

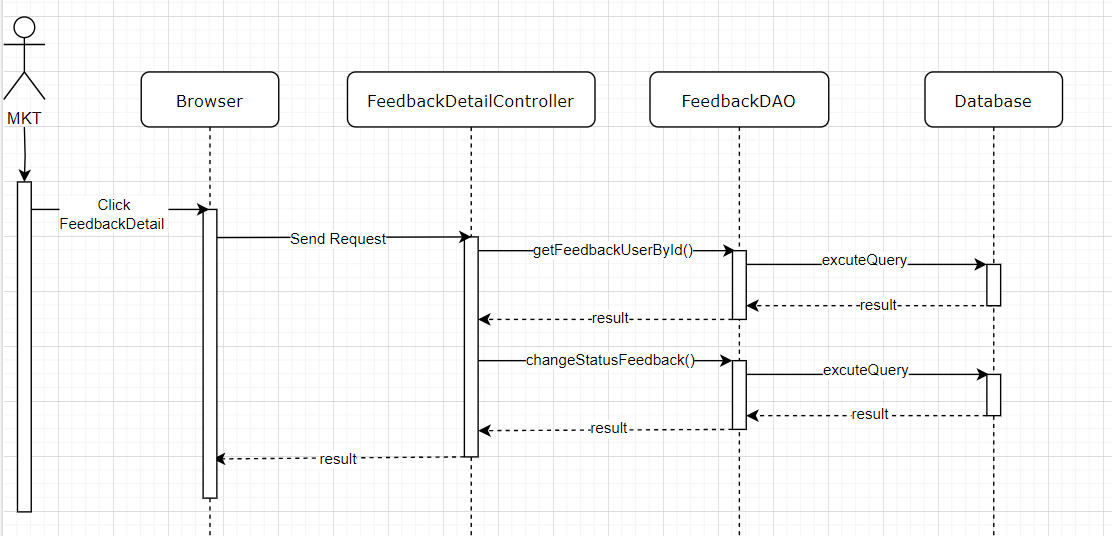


### b. Class Specifications

| **No** | **Method** | **Description** |
| --- | --- | --- |
| *01* | *getFeedbackUserById()* | *get user feedback by id* |
| *02* | *changeStatusFeedback()* | *change the status of user 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 = ?;

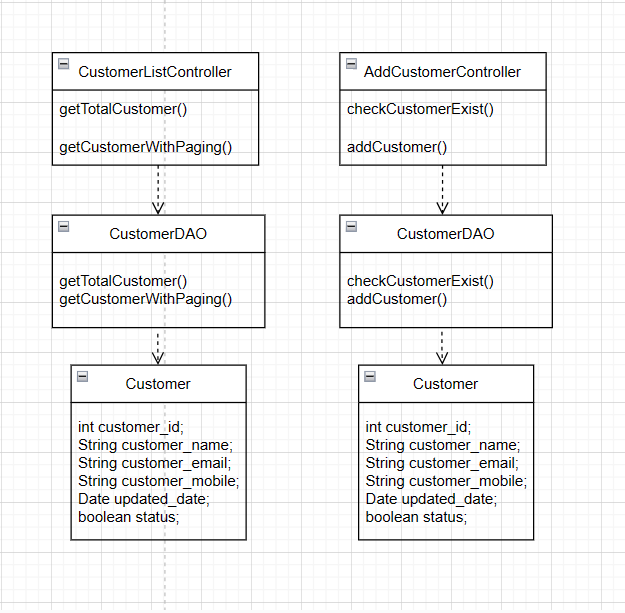
- UPDATE `Feedback` "

+ "SET `status` = ? "

+ "WHERE `feedBack\_id` = ?

## 33.Marketing/Customer List

### a. Class Diagram



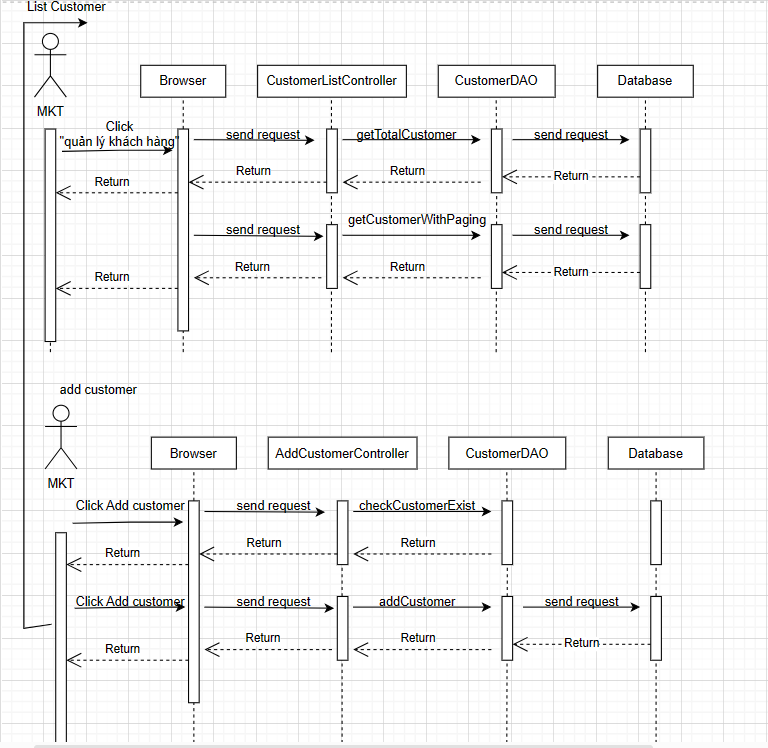
### b. Class Specifications

### 

| **No** | **Method** | **Description** |
| --- | --- | --- |
| *01* | *getTotalCustomer()* | *get Total Customer* |
| *02* | *getCustomerWithPaging()* | *get Customer With Paging* |
| *03* | *checkCustomerExist()* | *check Customer duplicate* |
| *04* | *addCustomer()* | *add new Customer* |

### 

### c. Sequence Diagram(s)



### d. Database queries

Select count(\*) from books\_shop\_online.customer\n"

+ "where status " + status + " and customer\_name like N'%" + searchKey + "%'

select \* from Customer\n"

+ "where status " + status + " and customer\_name like N'%" + searchKey + "%'\n"

+ " order by " + value + " " + type + " LIMIT ?, ?;

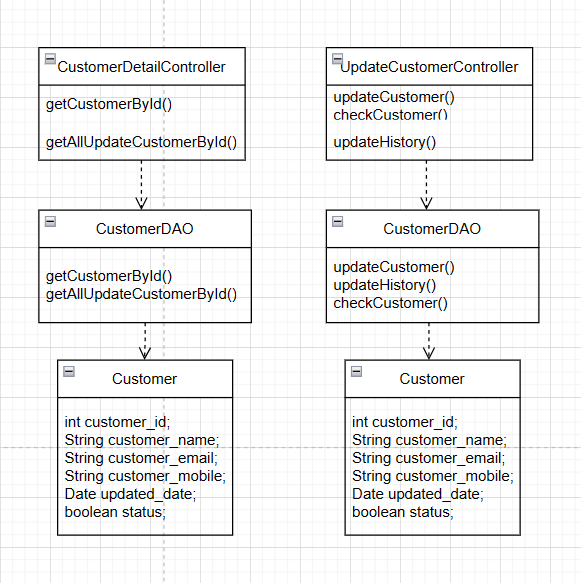
select \* from books\_shop\_online.customer\n"

+ "where customer\_email = ?

INSERT INTO customer (customer\_name, customer\_email, customer\_mobile, updated\_date, status) VALUES (?, ?, ?, CURRENT\_TIMESTAMP, ?)

## 34.Marketing/Customer Detail

### a. Class Diagram



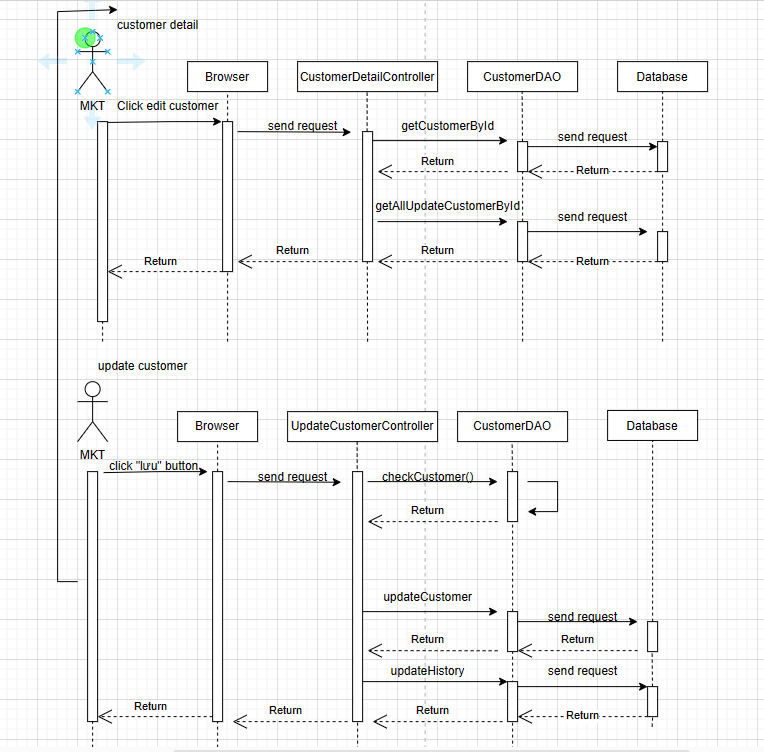
### b. Class Specifications

### 

| **No** | **Method** | **Description** |
| --- | --- | --- |
| *01* | *getCustomerById* | *getCustomerById* |
| *02* | *getAllUpdateCustomerById()* | *getAllUpdateCustomerById* |
| *03* | *checkCustomer()* | *check duplicate Customer* |
| *04* | *updateCustomer()* | *update Customer* |
| *05* | *updateHistory()* | *update History* |

### 

### c. Sequence Diagram(s)



### d. Database queries

select \* from customer\n"

+ "where customer\_id = ?

SELECT uc.\*, u.fullName\n"

+ "FROM Update\_Customer uc\n"

+ "JOIN User u ON uc.update\_by = u.userId\n"

+ "WHERE uc.customer\_id = ?;

SELECT customer\_id, customer\_name, customer\_email, customer\_mobile\n"

+ "FROM Customer\n"

+ "WHERE (customer\_email = ? AND customer\_id != ?) OR (customer\_mobile = ? AND customer\_id != ?)

UPDATE customer\n"

+ "SET customer\_name = ?,\n"

+ " customer\_email = ?,\n"

+ " customer\_mobile = ?,\n"

+ " updated\_date = CURRENT\_TIMESTAMP\n"

+ "WHERE customer\_id = ?

INSERT INTO Update\_Customer\n"

+ " (customer\_id\n"

+ " ,email\n"

+ " ,fullName\n"

+ " ,mobile\n"

+ " ,update\_by)\n"

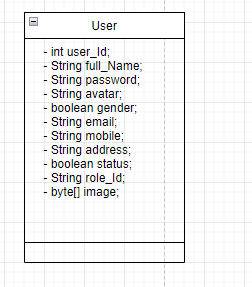
+ " VALUES\n"

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

## 

## 35. <Common/Authorization >

### a. Class Diagram



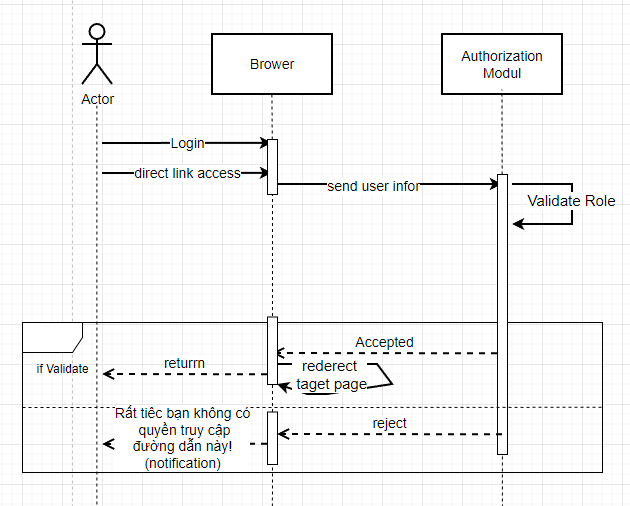
### b. Class Specifications

### 

| **No** | **Method** | **Description** |
| --- | --- | --- |

### 

### c. Sequence Diagram(s)

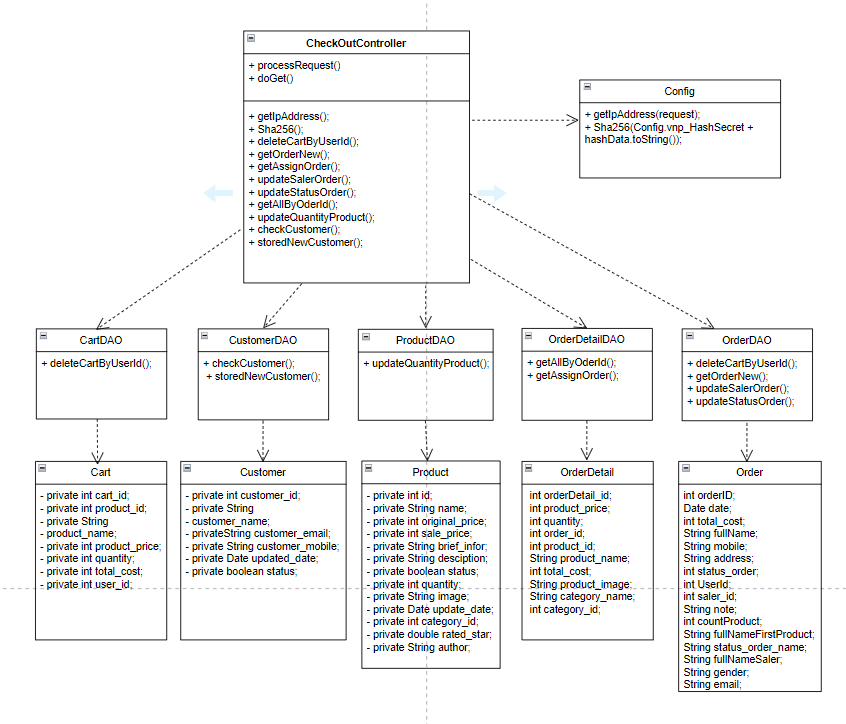


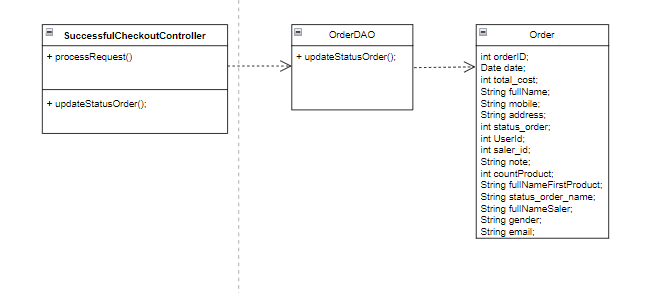
### d. Database queries

## 36.Public/CartCompletion

### a. Class Diagram

### 



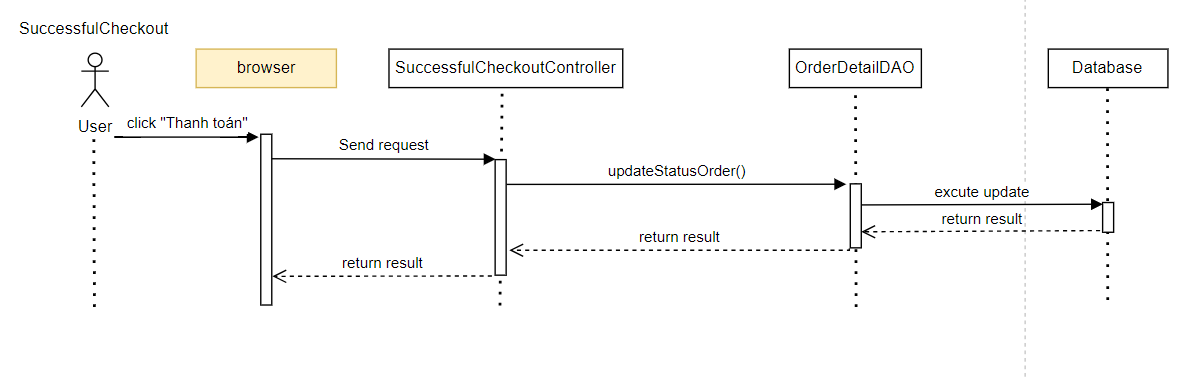
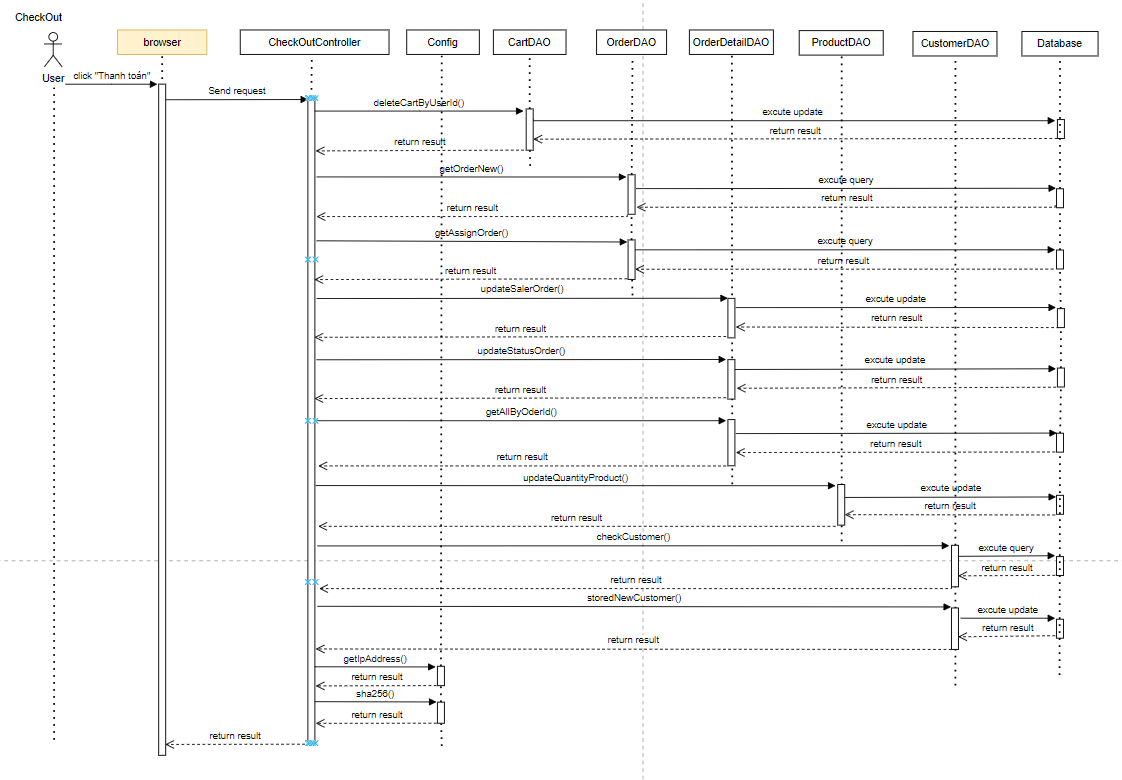
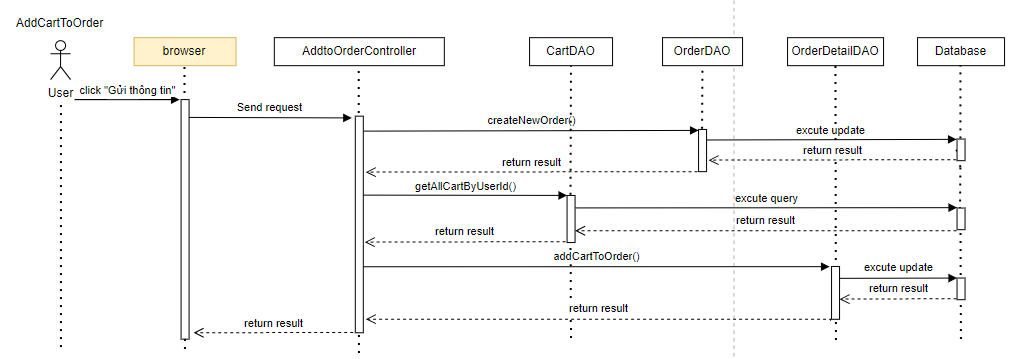


### b. Class Specifications

### 

| **No** | **Method** | **Description** |
| --- | --- | --- |
| *01* | *createNewOrder();* | *create New Order* |
| *02* | *getAllCartByUserId();* | *get All of Carts By User* |
| *03* | *addCartToOrder();* | *add Cart To User Order* |
| *04* | *getIpAddress();* | *get the Ip Address of user* |
| *05* | *Sha256();* | *compute the SHA-256 hash value of a given message.* |
| *06* | *deleteCartByUserId();* | *delete Cart By User* |
| *07* | *getOrderNew();* | *get the new Order* |
| *08* | *getAssignOrder();* | *get random saler for order* |
| *09* | *updateSalerOrder();* | *update Saler for Order* |
| *10* | *updateStatusOrder();* | *update Status for Order* |
| *11* | *getAllByOderId();* | *get All OderDetail by OrderId* |
| *12* | *updateQuantityProduct();* | *update Quantity of Product* |
| *13* | *checkCustomer();* | *check Customer exist* |
| *14* | *storedNewCustomer();* | *stored a New Customer* |

### c. Sequence Diagram(s)



### d. Database queries

* "INSERT INTO `Order` " +

"(`total\_cost`, `fullName`, `mobile`, `address`, `userId`, `note`) " +

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

* "SELECT \* FROM Cart WHERE user\_id = ?"
* "INSERT INTO `Order\_Detail`\n"

+ " (`product\_price`, `quantity`, `order\_id`, `product\_id`, `product\_name`, `total\_cost`)\n"

+ "VALUES\n"

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

* "DELETE FROM `Cart` "

+ "WHERE `user\_id` = ?"

* SELECT \* FROM `Order` WHERE `userId` = ? AND `status\_order` IS NULL
* "SELECT \* FROM `User` WHERE role\_id = 3\n"

+ "ORDER BY RAND()\n"

+ "LIMIT 1;"

* "UPDATE `Order`\n"

+ "SET `saler\_id` = ?\n"

+ "WHERE `order\_id` = ?;"

* "UPDATE `Order` " +

"SET `status\_order` = ? " +

"WHERE `order\_id` = ?"

* SELECT \*\n"

+ "FROM `Order\_Detail`\n"

+ "WHERE order\_id = ?"

* "UPDATE `Product` " +

"SET `quantity` = (quantity - ?) " +

"WHERE `product\_id` = ?";

* "SELECT customer\_id, customer\_name, customer\_email, customer\_mobile "

+ "FROM books\_shop\_online.customer "

+ "WHERE customer\_name = ? AND customer\_email = ? OR customer\_mobile = ?"

* "INSERT INTO books\_shop\_online.customer "

+ "(customer\_name, customer\_email, customer\_mobile) "

+ "VALUES (?, ?, ?)"