

**TV ONLINE SHOP PROJECT**

**Software Design Specification**

**Group 1**

– Hanoi, August 2022 –

Record of changeS

|  |  |  |  |
| --- | --- | --- | --- |
| Date | A\* M, D | In charge | Change Description |
| 03/01 | M |  | Choose Topic Online TV Shop. |
| 09/01 | M |  | Create plan for single iteration and milestone, goal for single iteration. |
| 12/01 | M |  | Create Database |
| 18/01 | M |  | Fix Database |
| 03/02 | M |  | Fix Database |
| 03/02 | M |  | Create SRS Documentary |
| 07/02 | M |  | Find and create template |
| 08/02 | M |  | Learning Using git to push code |
| 10/02 | M |  | Code Login and logout feature |
| 13/02 | M |  | Code CRUD of user |
| 13/02 | M |  | Code UserProfile |
| 13/02 | M |  | Add document to SRS documentary |
| 17/02 | A |  | Code Home Page |
| 21/02 | A |  | Create SDS documentary |

\*A - Added M - Modified D - Deleted

**Table of Contents**

[I. Overview 6](#_Toc22619)

[1. Code Packages 6](#_Toc29384)

[2. Database Design 6](#_Toc2323)

[a. Database Schema 6](#_Toc26606)

[b. Table Description 7](#_Toc11997)

[II. Code Designs 8](#_Toc5500)

[1. Login 8](#_Toc15749)

[a. Class Diagram 8](#_Toc18215)

[d. Database queries 9](#_Toc9407)

[2. Sign up 9](#_Toc20107)

[a. Class Diagram 9](#_Toc29411)

[b. Class Specifications 9](#_Toc32339)

[Class 1 : Register control 9](#_Toc3783)

[c. Database queries 10](#_Toc12433)

[3. Home 10](#_Toc20344)

[a. Class Diagram 10](#_Toc11491)

[b. Class Specifications 11](#_Toc20845)

[c. Database queries 11](#_Toc11225)

[4. User Profile 11](#_Toc2344)

[a. Class Diagram 11](#_Toc7536)

[b. Class Specifications 11](#_Toc17963)

[5. Change Password 12](#_Toc7350)

[a. Class Diagram 12](#_Toc25235)

[b. Class Specifications 12](#_Toc16751)

[c. Sequence Diagram(s) 13](#_Toc11438)

[d. Database queries 13](#_Toc18410)

[6. Change Profile 13](#_Toc16893)

[a. Class Diagram 13](#_Toc2148)

[b. Class Specifications 13](#_Toc3967)

[c. Sequence Diagram(s) 14](#_Toc17072)

[d. Database queries 14](#_Toc2371)

[7. Forgot password 15](#_Toc10242)

[a. Class Diagram 15](#_Toc31631)

[b. Class Specifications 15](#_Toc7168)

[c. Sequence Diagram(s) 15](#_Toc10772)

[d. Database queries 16](#_Toc12179)

[8. Edit account 16](#_Toc22035)

[a. Class Diagram 16](#_Toc11022)

[b. Class Specifications 16](#_Toc19037)

[c. Sequence Diagram(s) 16](#_Toc27290)

[d. Database queries 16](#_Toc17553)

[9. Add staff account 17](#_Toc10004)

[a. Class Diagram 17](#_Toc3333)

[b. Class Specifications 17](#_Toc22199)

[c. Sequence Diagram(s) 18](#_Toc23355)

[d. Database queries 18](#_Toc27729)

[10. Delete staff account 19](#_Toc21211)

[a. Class Diagram 19](#_Toc12492)

[b. Class Specifications 19](#_Toc1405)

[c. Sequence Diagram(s) 20](#_Toc4689)

[d. Database queries 20](#_Toc1997)

[11. Export order to excel 20](#_Toc23902)

[a. Class Diagram 20](#_Toc23762)

[b. Class Specifications 20](#_Toc14192)

[c. Sequence Diagram(s) 21](#_Toc20570)

[12. Product manage 21](#_Toc12350)

[a. Class Diagram 22](#_Toc25051)

[b. Class Specifications 22](#_Toc9626)

[c. Database queries 22](#_Toc11082)

[13. Add product 23](#_Toc22376)

[a.Class Diagram 23](#_Toc19378)

[e. Class Specifications 23](#_Toc13396)

[b. Sequence Diagram(s) 24](#_Toc23452)

[d. Database queries 24](#_Toc27947)

[14. Edit product 25](#_Toc24956)

[a. Class Diagram 25](#_Toc3072)

[d. Database queries 26](#_Toc11968)

[15. Search product 27](#_Toc30090)

[a. Class Diagram 27](#_Toc4669)

[b. Class Specifications 27](#_Toc11868)

[c. Sequence Diagram(s) 28](#_Toc29446)

[d. Database queries 28](#_Toc19171)

[16. Manage Order 28](#_Toc16113)

[d. Database queries 29](#_Toc26701)

[17. Update order 30](#_Toc12887)

[a. Class Diagram 30](#_Toc15822)

[b. Class Specifications 30](#_Toc27352)

[c. Sequence Diagram(s) 31](#_Toc4315)

[d. Database queries 31](#_Toc1315)

[18. Search order 31](#_Toc27720)

[a. Class Diagram 31](#_Toc17937)

[b. Class Specifications 32](#_Toc14716)

[c. Sequence Diagram(s) 32](#_Toc4506)

[d. Database queries 32](#_Toc5137)

[19. Add to cart 33](#_Toc18026)

[a. Class Diagram 33](#_Toc13205)

[b. Class Specifications 33](#_Toc2847)

[c. Sequence Diagram(s) 33](#_Toc25682)

[d. Database queries 34](#_Toc17015)

[20. Purchase 34](#_Toc3902)

[a. Class Diagram 34](#_Toc17913)

[b. Class Specifications 34](#_Toc4739)

[c. Sequence Diagram(s) 35](#_Toc17121)

[d. Database queries 35](#_Toc30534)

[21. Tracking order 35](#_Toc13132)

[a. Class Diagram 35](#_Toc13964)

[b. Class Specifications 36](#_Toc32257)

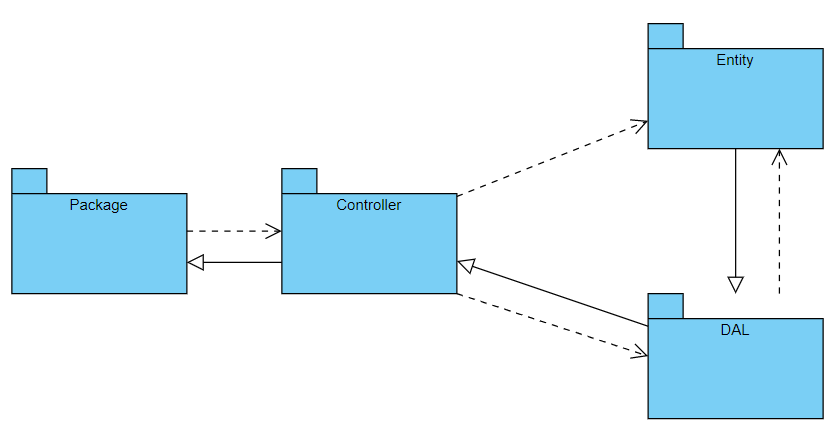
[c. Sequence Diagram(s) 36](#_Toc3702)

[c. Database queries 36](#_Toc22856)

# 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 and description table format below – following Java project naming convention]



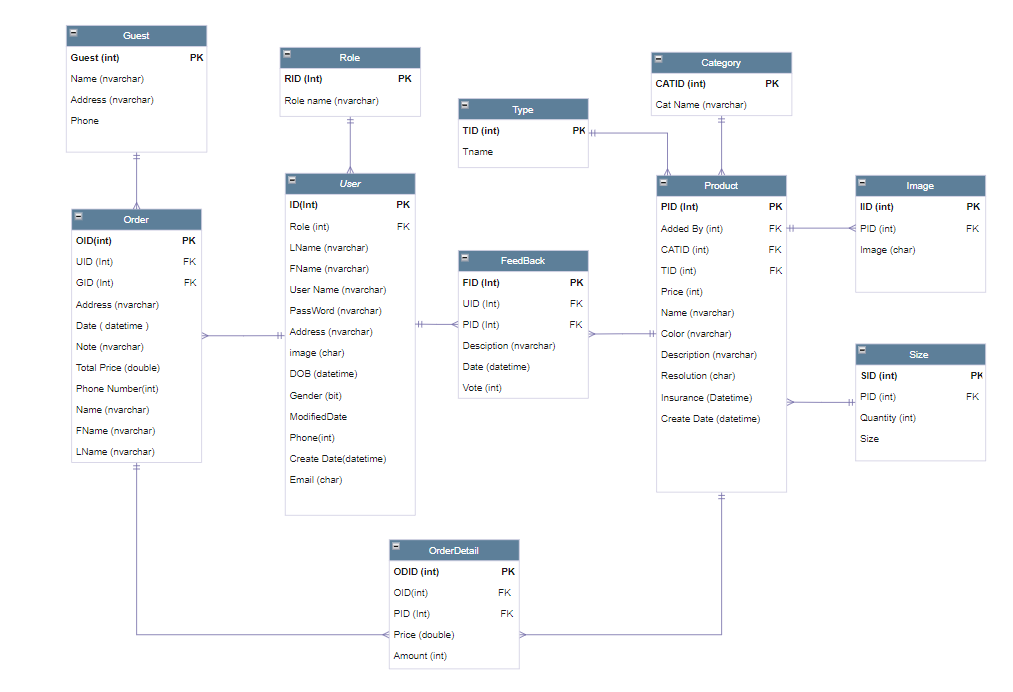
**Package descriptions & package class naming conventions**

|  |  |  |
| --- | --- | --- |
| **No** | **Package** | **Description** |
| 01 | DAL |  |
| 02 | connectDB | This package is to connect with database. |
| 03 | Controller | This package is to manage servlets. Get data, process data and control the pages. |
| 04 | Entity | This package have entities for tables in database and have properties the same with tables in database. |

## 2. Database Design

### a. Database Schema

[Provide the tables relationship like example below – following SQL server database naming convention]



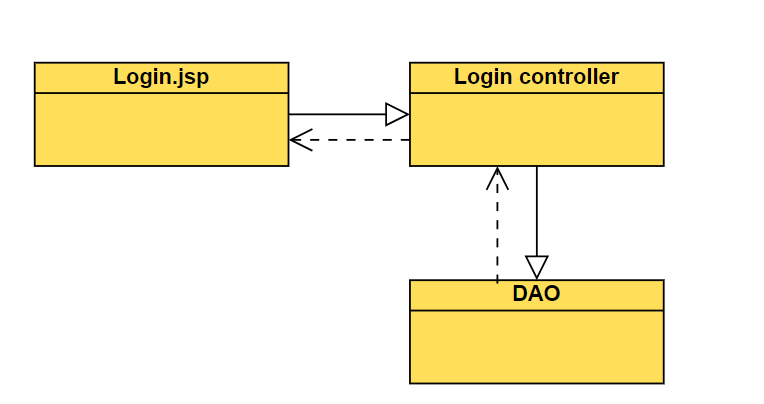
### b. Table Description

|  |  |  |
| --- | --- | --- |
| **No** | **Table** | **Description** |
| 01 | Guest | Show properties of guest in database to upload data from SQL |
| 02 | Role | Show properties of role in database to upload data from SQL |
| 03 | Order | Show properties of order in database to upload data from SQL |
| 04 | User | Show properties of user in database to upload data from SQL |
| 05 | OrderDetail | Show properties of order detail in database to upload data from SQL |
| 06 | FeedBack | Show properties of feed back in database to upload data from SQL |
| 07 | Type | Show properties of type in database to upload data from SQL |
| 08 | Category | Show properties of category in database to upload data from SQL |
| 09 | Product | Show properties of product in database to upload data from SQL |
| 10 | Image | Show properties of image in database to upload data from SQL |
| 11 | Size | Show properties of size in database to upload data from SQL |

# II. Code Designs

## 1. Login

### a. Class Diagram



**b. Class Specifications**

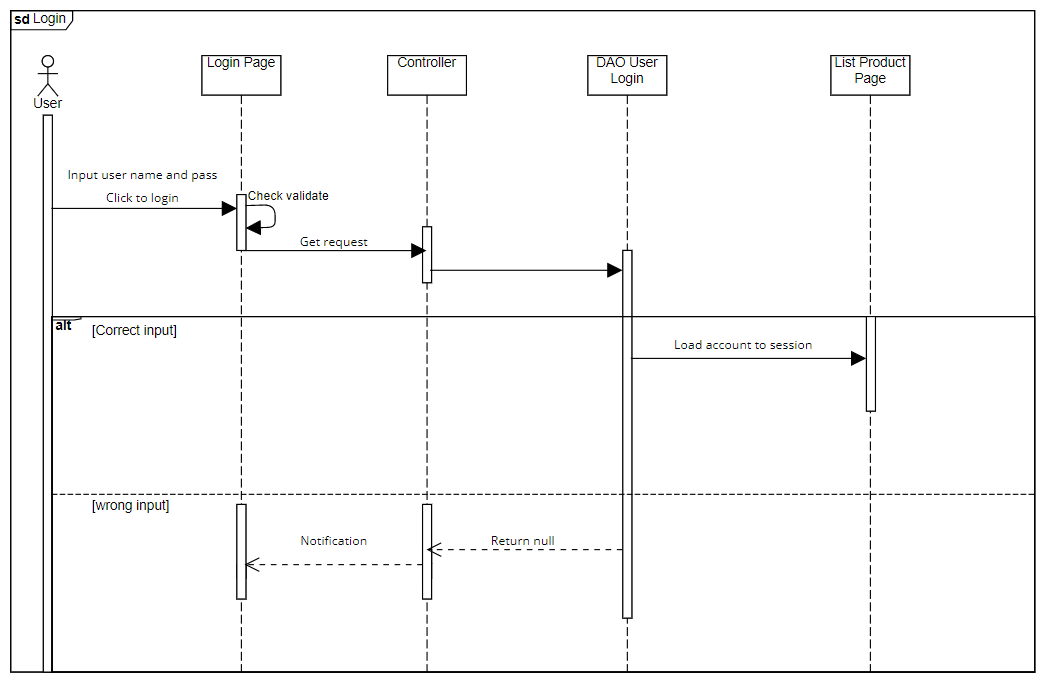
#### Class 1 : DAO

|  |  |  |
| --- | --- | --- |
| **No** | **Method** | **Description** |
| 01 | Login | This method will take user name + password information of user inputted to return account. |

#### Class 2 : LoginControl

|  |  |  |
| --- | --- | --- |
| **No** | **Method** | **Description** |
| 01 | Login | Login this method will use to call dao login method. |

**c. Sequence Diagram(s)**



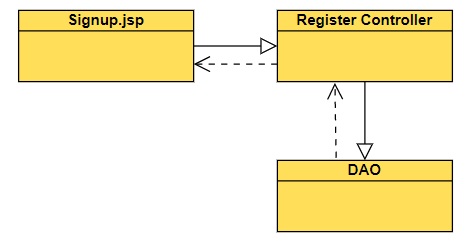
### d. Database queries

## **"Select \* from [User] where [username] =?\n"**

## **+ " and [password] =?"**

## 2. Sign up

### a. Class Diagram



### b. Class Specifications

### Class 1 : Register control

|  |  |  |
| --- | --- | --- |
| **No** | **Method** | **Description** |
| 01 | existedUser | This method will check if username existed or not , if existed return sign up.jsp. |
| 02 | register | This method take all information of user sign up inputted to database. |

#### Class 2 : DAO

|  |  |  |
| --- | --- | --- |
| **No** | **Method** | **Description** |
| 01 | register | This method will take all user information inputted to database. |

### c. Database queries

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

**+ " ([Role]\n"**

**+ " ,[FName]\n"**

**+ " ,[UserName]\n"**

**+ " ,[PassWord]\n"**

**+ " ,[Phone]\n"**

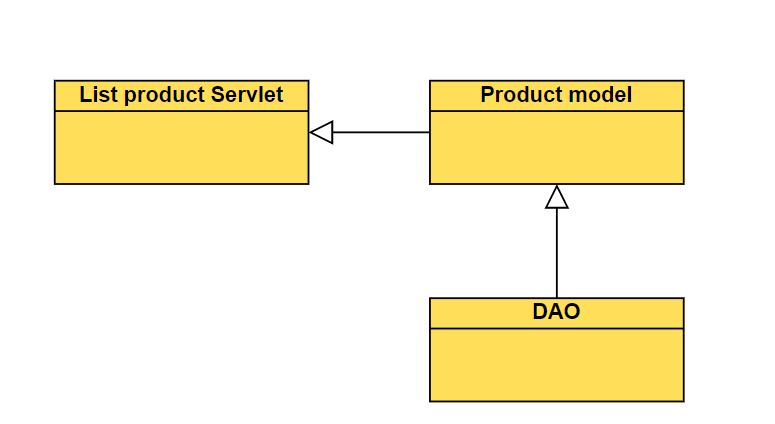
**+ " ,[CreatedDate]\n"**

**+ " ,[ModifiedDate])\n"**

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

## 3. Home

### a. Class Diagram



### b. Class Specifications

#### Class 1 : DAO

|  |  |  |
| --- | --- | --- |
| **No** | **Method** | **Description** |
| 01 | GetAlltype | This method take type by product id in database. |
| 02 | Get all image by id | This method take all image by product id in database. |
| 03 | Getcategory by id | This method take category by product id in database. |

### c. Database queries

**"select \* from Type" get all type**

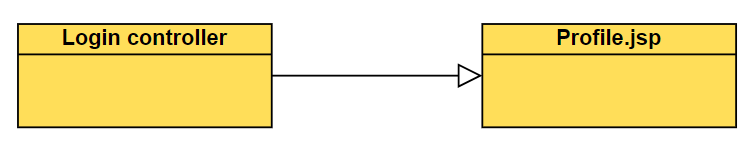
**"select \* from Type where TID = ?" get type by id**

**"select \* from category where CATID = ?" get category by id**

**"select \* from Image where PID = ? "get all image by id**

## 4. User Profile

### a. Class Diagram

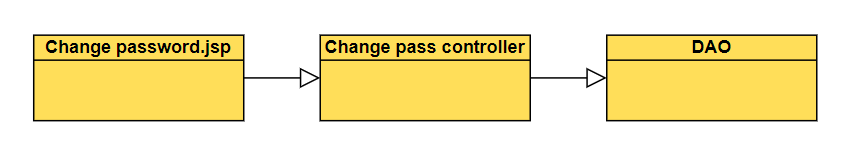


### b. Class Specifications

|  |  |  |
| --- | --- | --- |
| **No** | **Method** | **Description** |
| 01 | sessionscope.acc.lname | this medthod will take last name from session of user |
| 02 | sessionscope.acc.fname | This method will take first name form session of user. |
| 03 | sessionscope.acc.dob | This method will take dob form session of user. |
| 04 | sessionscope.acc.gender | This method will take gender form session of user. |
| 05 | sessionscope.acc.email | This method will take email form session of user. |
| 06 | sessionscope.acc.phone | This method will take phone form session of user. |
| 07 | sessionscope.acc.address | This method will take address form session of user. |
| 08 | sessionscope.acc.image | This method will take image form session of user. |

## **5. Change Password**

### a. Class Diagram

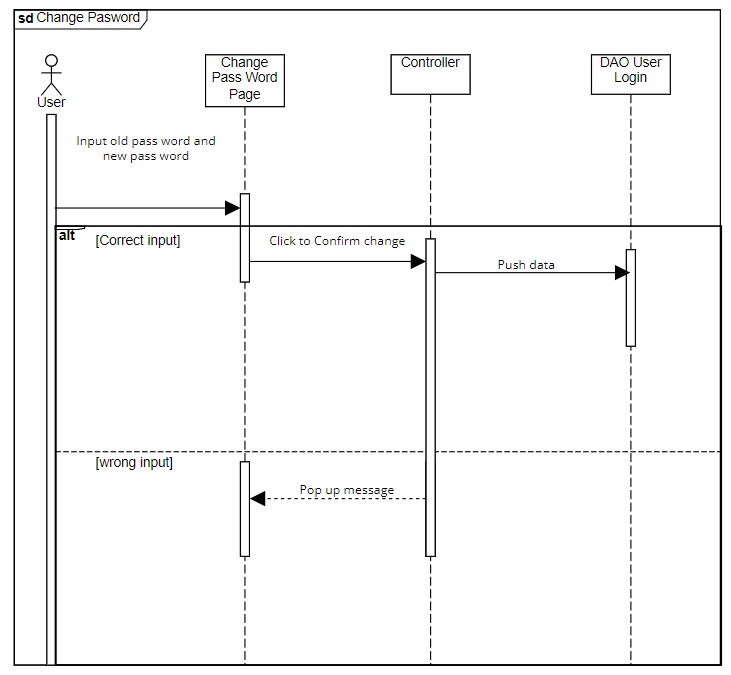


### b. Class Specifications

|  |  |  |
| --- | --- | --- |
| No | Method | Description |
| 1 | (User)session.getAttribute("acc"); | This method will take session of user when login |
| 2 | changePass | This method will change password of user in database |

|  |  |  |
| --- | --- | --- |
| **No** | **Method** | **Description** |
| 01 | (User)session.getAttribute("acc"); | This method will take session of user when login |
| 02 | changePass | This method will change password of user in database |

### c. Sequence Diagram(s)

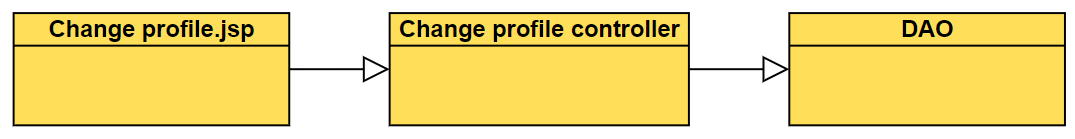


### d. Database queries

update [User] set PassWord = ? where [UserName] = ?

## 6. Change Profile

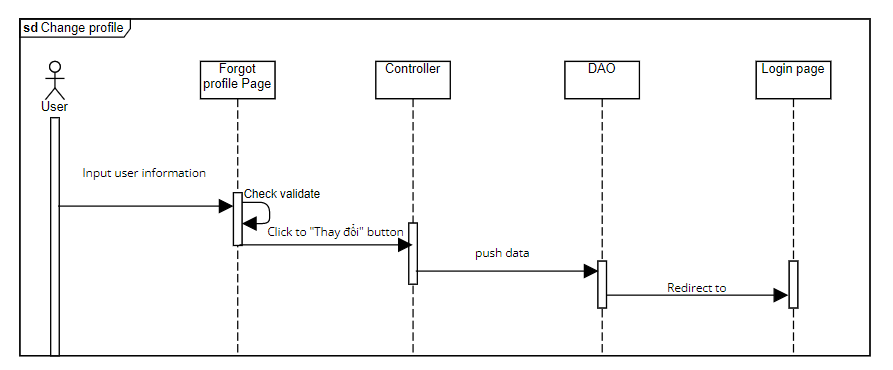
### a. Class Diagram



### b. Class Specifications

|  |  |  |
| --- | --- | --- |
| **No** | **Method** | **Description** |
| 01 | checkUsUid(uid) | This method will check user exits or not |
| 02 | changeprofile(lname, fname, address, image, dob, gen, phone, email, format, uid) | This method will change information user in database |

### c. Sequence Diagram(s)



### d. Database queries

Select \* from [User] where [ID] =?

UPDATE [dbo].[User]\n"

+ " SET [LName] = ?"

+ " ,[FName] = ?"

+ " ,[Address] = ?"

+ " ,[image] = ? "

+ " ,[DOB] = ? "

+ " ,[Gender] = ? "

+ " ,[Phone] = ? "

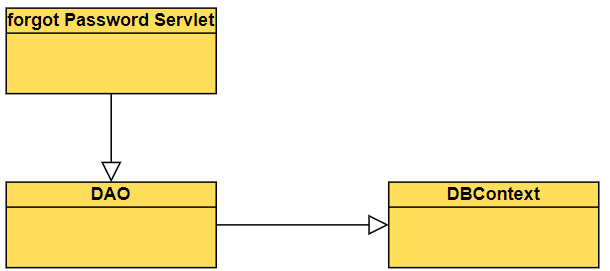
+ " ,[Email] = ? "

+ " ,[ModifiedDate] = ?"

+ " WHERE [ID] = ? "

## Forgot password

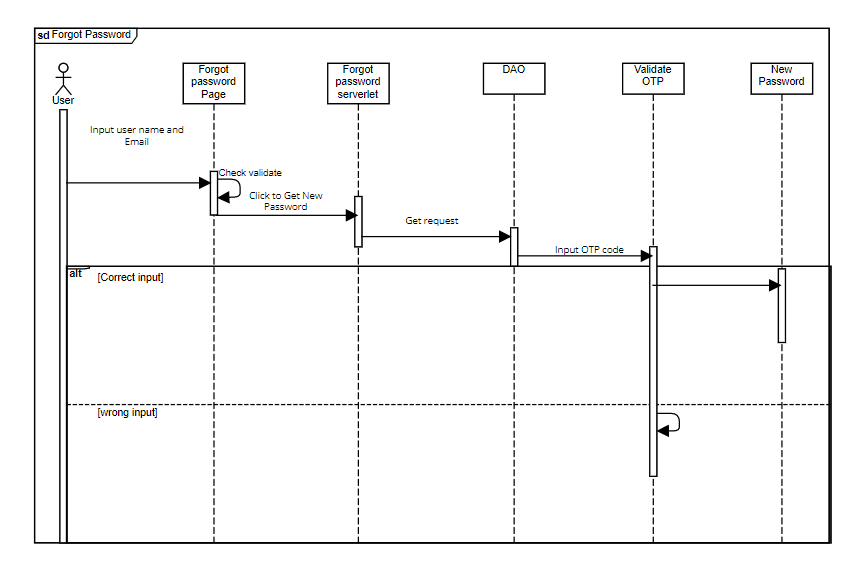
### Class Diagram



### Class Specifications

|  |  |  |
| --- | --- | --- |
| **No** | **Method** | **Description** |
| 01 | checkUsername() | This method will check user name. |

### Sequence Diagram(s)

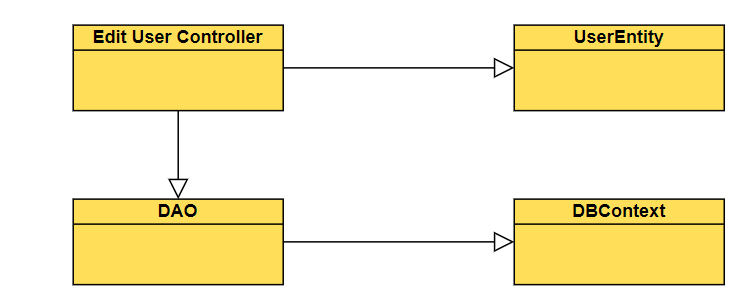


### Database queries

"Select \* from [User] where [username] =?\n"

## Edit account

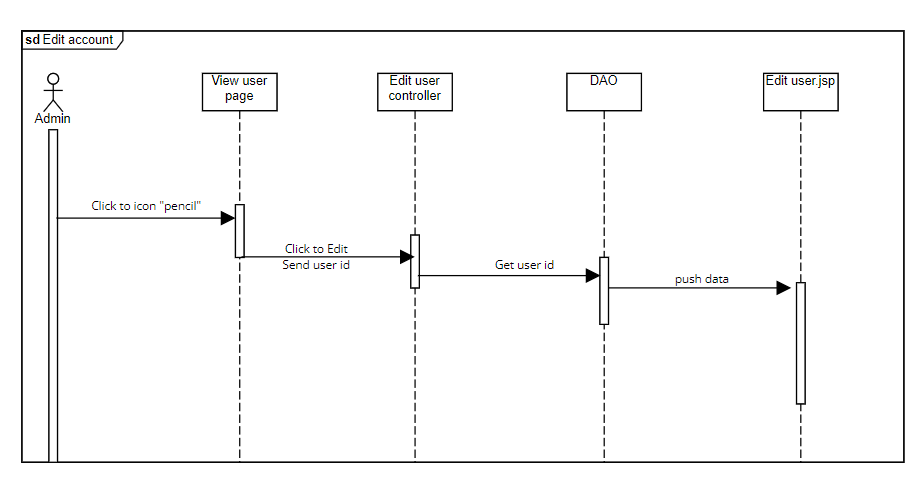
### Class Diagram



### Class Specifications

|  |  |  |
| --- | --- | --- |
| **No** | **Method** | **Description** |
| 01 | checkUsUid() | This method will check user id name |
| 02 | updateRole() | This method will check user role |

### Sequence Diagram(s)



### d. Database queries

"Select \* from [User] where [ID] =?\n"

"UPDATE [dbo].[User]\n"

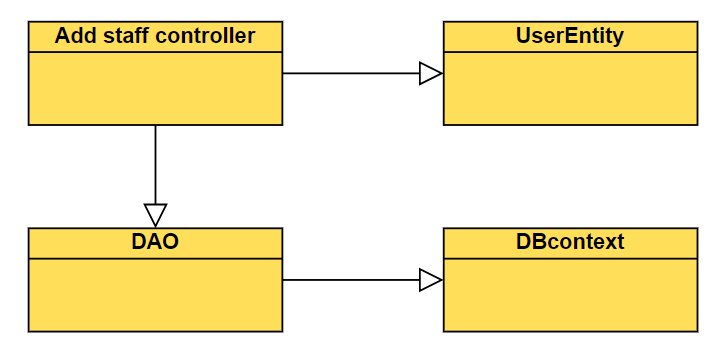
+ " SET [Role] = ?\n"

+ " \n"

+ " WHERE ID = ?"

## Add staff account

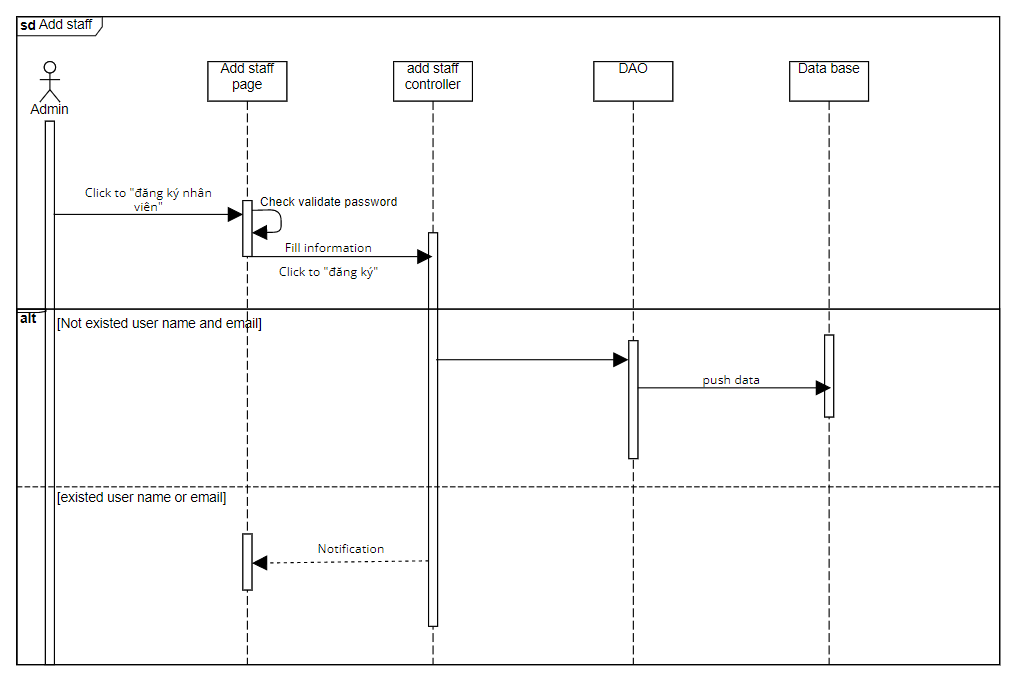
### Class Diagram



### Class Specifications

|  |  |  |
| --- | --- | --- |
| **No** | **Method** | **Description** |
| 01 | existedUser() | This method will check if user existed or not |
| 02 | getUserbyEmail() | This method will get user by email |
| 03 | checkUsername() | This method will check iff user name existed or not |
| 04 | setRoleId() | This method will set role id for staff |
| 05 | setfName() | This method will set first name for staff |
| 06 | setUsername() | This method will set user name for staff |
| 07 | setEmail() | This method will set email for staff |
| 08 | setPass() | This method will set password for staff |
| 09 | setCreateDate() | This method will set create date for staff |
| 10 | setModifyDate() | This method will set modify date for staff |
| 11 | register() | This method will set input information of staff in to database |

### Sequence Diagram(s)



### Database queries

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

+ " ([Role]\n"

+ " ,[FName]\n"

+ " ,[UserName]\n"

+ " ,[PassWord]\n"

+ " ,[Phone]\n"

+ " ,[Email]\n"

+ " ,[CreatedDate]\n"

+ " ,[ModifiedDate])\n"

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

"Select \* from [User] where [username] =?\n"

"SELECT \* "

+ " FROM [dbo].[User]"

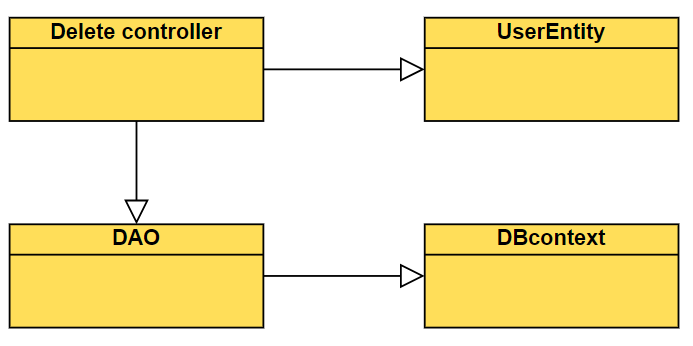
+ "Where [UserName] = ? "

"select \* from [User] \n"

+ "where Email = ?"

## Delete staff account

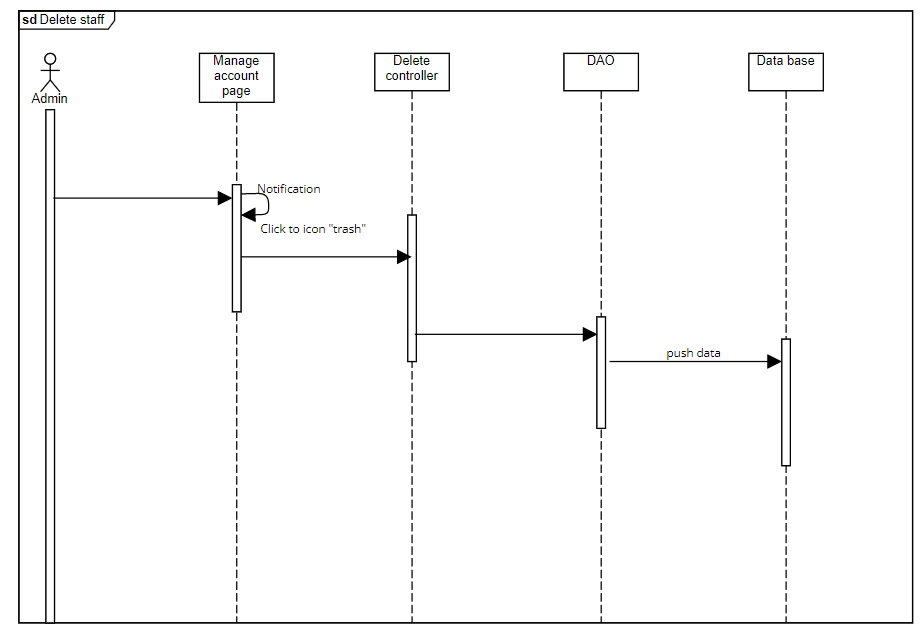
### Class Diagram



### Class Specifications

|  |  |  |
| --- | --- | --- |
| **No** | **Method** | **Description** |
| 01 | delete() | This method will delete staff by uid |

### Sequence Diagram(s)



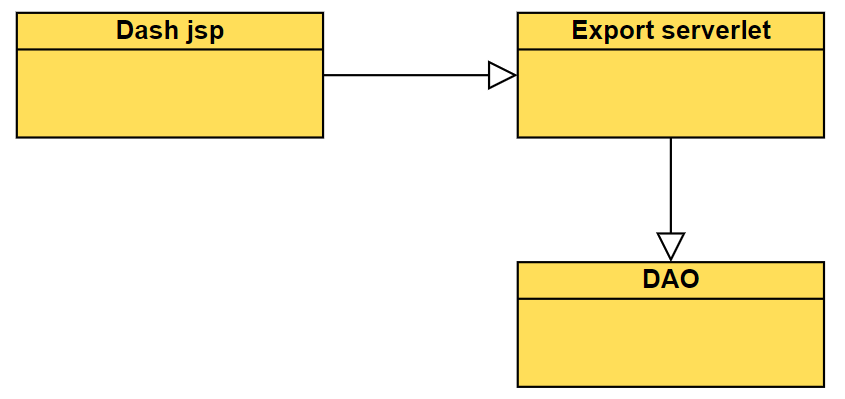
### d. Database queries

"DELETE FROM [dbo].[User]\n"

+ " WHERE ID=?"

## Export order to excel

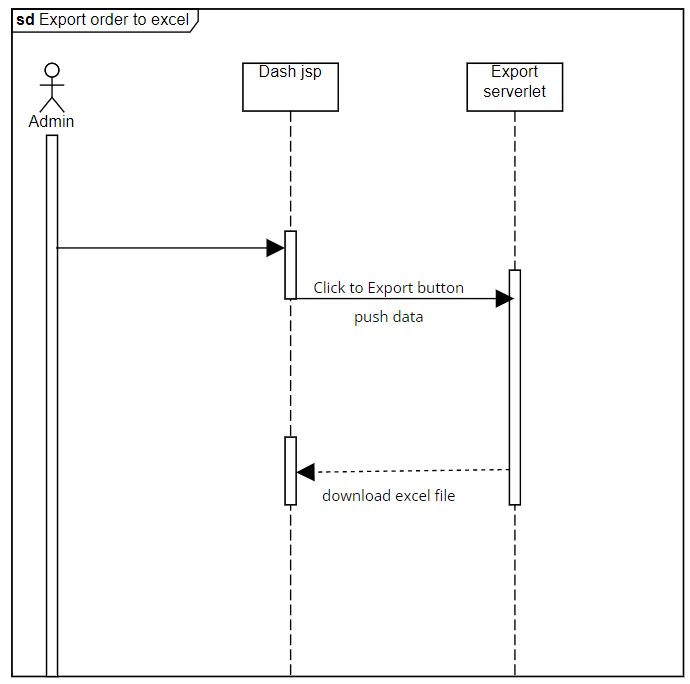
### Class Diagram



### Class Specifications

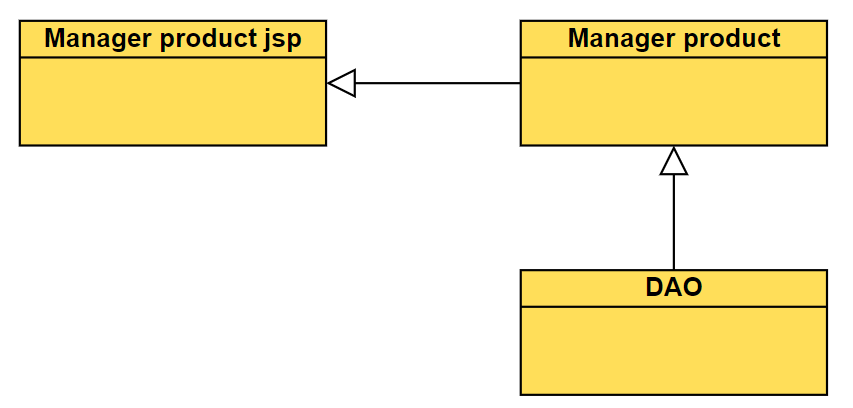
|  |  |  |
| --- | --- | --- |
| **No** | **Method** | **Description** |
| 01 | getAllOrder() | This method will return all list of order |

### Sequence Diagram(s)



## Product manage

### Class Diagram



### b. Class Specifications

#### Class 1 : DAO

|  |  |  |
| --- | --- | --- |
| **No** | **Method** | **Description** |
| 01 | getAllProd() | This method will take all Product information |
| 02 | getProductCount() | This method will calculate the total number of products |
| 03 | getProductCountInurance() | This method will count product insurance more than 12 month |
| 04 | getProductCountQuantity() | This method will count quantity of all Product |

#### Class 2 : ManagerProduct

|  |  |  |
| --- | --- | --- |
| **No** | **Method** | **Description** |
| 01 | List<Product> | . This function will list all products from the database |
| 02 | searchByName | . This function will Search products by name |

### c. Database queries

select \* from Product

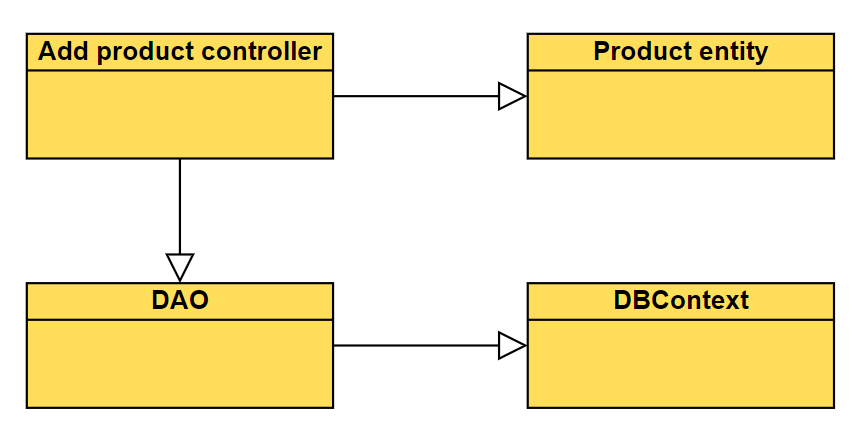
SELECT COUNT(\*) FROM Product

SELECT COUNT(\*) FROM Product Where Insurance >= 12

SELECT Sum(Quantity) FROM Product

## Add product

### a.Class Diagram



### Class Specifications

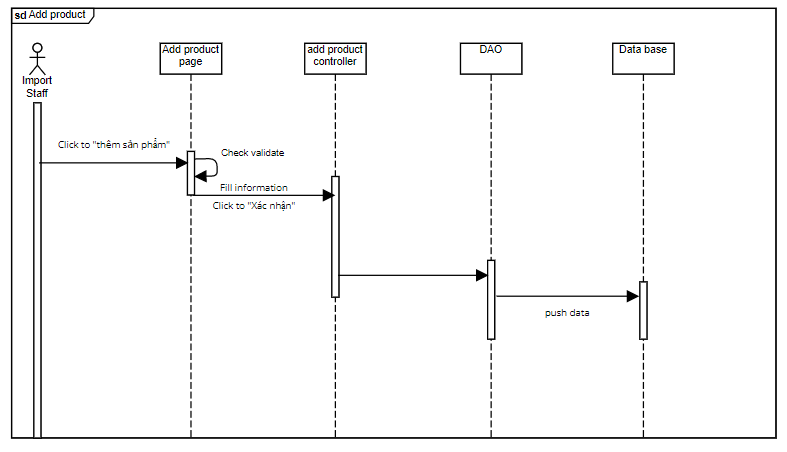
#### Class 1 : DAO

|  |  |  |
| --- | --- | --- |
| **No** | **Method** | **Description** |
| 01 | insertProduct | This method will take insert all Product information |

#### Class 2 : AddProduct

|  |  |  |
| --- | --- | --- |
| **No** | **Method** | **Description** |
| 01 | dao.insertProduct(a.getuId(), pcatid, pprice, pname, pcolor, pdescription, presolution, pinsurance, format, ptid, pimage, psize, pquantity, pdiscount, ppriceout); | . This function will add id of the person who added the product, category, price in, color, name of product, description, resolution, insurance, Date modified, type, image, size, quantity, discount, price out |

### Sequence Diagram(s)



### d. Database queries

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

+ " ([AddedBy]\n"

+ " ,[CATID]\n"

+ " ,[PriceIn]\n"

+ " ,[Name]\n"

+ " ,[Color]\n"

+ " ,[Description]\n"

+ " ,[Resolution]\n"

+ " ,[Insurance]\n"

+ " ,[CreateDate]\n"

+ " ,[TID]\n"

+ " ,[ImageDefault]\n"

+ " ,[Size]\n"

+ " ,[Quantity]\n"

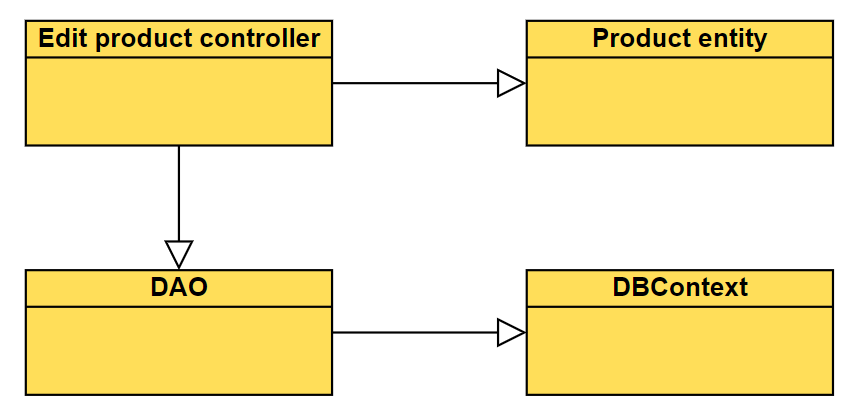
+ " ,[Discount]\n"

+ " ,[PriceOut])\n"

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

## Edit product

## Class Diagram



**b. Class Specifications**

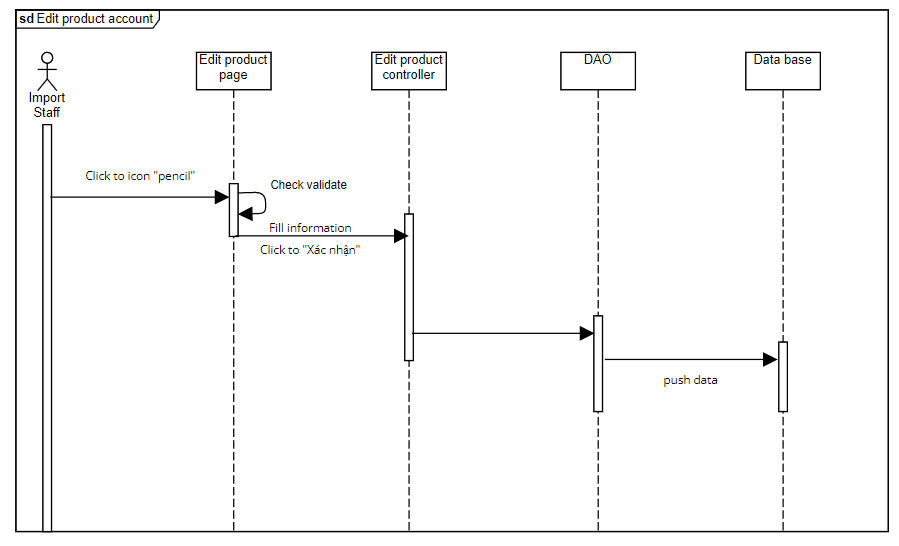
#### Class 1 : DAO

|  |  |  |
| --- | --- | --- |
| **No** | **Method** | **Description** |
| 01 | editProduct | This method will edit all Product information |

#### Class 2 : EditProduct

|  |  |  |
| --- | --- | --- |
| **No** | **Method** | **Description** |
| 01 | editProduct(pcatid, pprice, pname, pcolor, pdescription, presolution, pinsurance, format, ptid, pimage, psize, pquantity, pdiscount, ppriceout, Integer.parseInt(pid)); | This function will edit category, price in , name of product, color, description, resolution, insurance, Date modified, type, image, size, quantity, discount, price out based on the ProductID |

**c. Sequence**



### d. Database queries

UPDATE [dbo].[Product]\n"

+ " SET "

+ " [CATID] = ?\n"

+ " ,[PriceIn] = ?\n"

+ " ,[Name] = ?\n"

+ " ,[Color] = ?\n"

+ " ,[Description] = ?\n"

+ " ,[Resolution] = ?\n"

+ " ,[Insurance] = ?\n"

+ " ,[CreateDate] = ?\n"

+ " ,[TID] = ? \n"

+ " ,[ImageDefault] = ?\n"

+ " ,[Size] = ?\n"

+ " ,[Quantity] = ?\n"

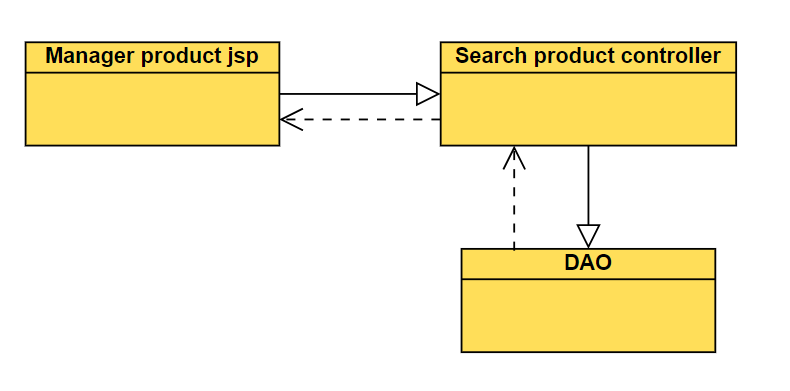
+ " ,[Discount] = ?\n"

+ " ,[PriceOut] = ?\n"

+ " Where [PID] = ?

## Search product

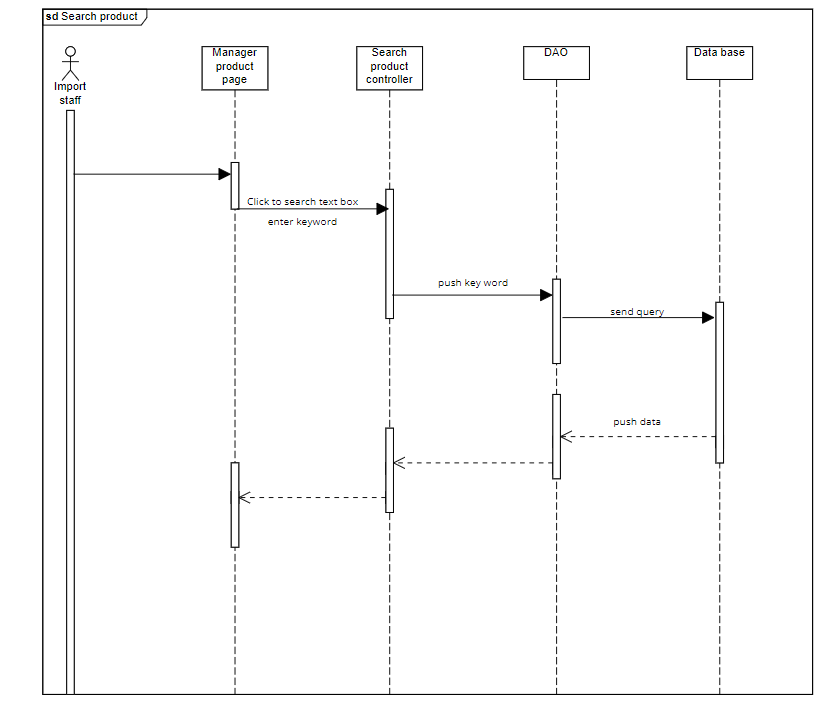
### Class Diagram



### Class Specifications

|  |  |  |
| --- | --- | --- |
| **No** | **Method** | **Description** |
| 01 | searchByName() | This method will return list of product by key word of name |

### Sequence Diagram(s)

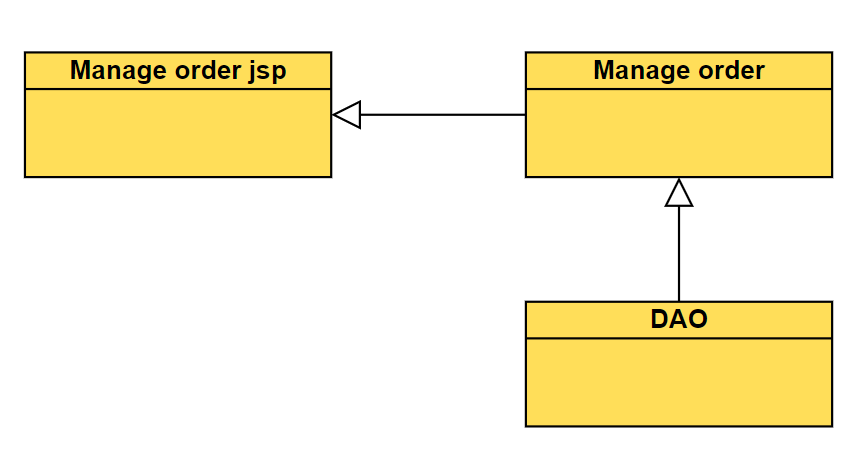


### Database queries

"select \* from Product\n"

+ " where [Name] like ?"

## Manage Order



**b. Class Specifications**

#### Class 1 : DAO

|  |  |  |
| --- | --- | --- |
| **No** | **Method** | **Description** |
| 01 | dao.getAllOrderLog() | This method will take all Order information |
| 02 | getTotalOrder() | This method will Calculate all total order |
| 03 | getOrder1 | This method will Calculate all order have statusId = 2 |
| 04 | getOrder2 | This method will Calculate all order have statusId = 3 |
| 05 | getOrder3 | This method will Calculate all order have statusId = 4 |
| 06 | getStatusOrder(statusid) | This method will Find all order have statusId = (1,2,3,4) |
| 07 | editOrder | This method will edit Order |

#### Class 2 : ManagerProduct

|  |  |  |
| --- | --- | --- |
| **No** | **Method** | **Description** |
| 01 | List<OrderLog> list = dao.getAllOrderLog(); | This function will list all Order from the database |
| 02 | FindStatus | This function will Find Order by statusId |
| 03 | editOrder(oid,status, format); | This method will edit statusId of Order from the database |

### d. Database queries

SELECT ol.OID, MAX(ol.StatusID) as MaxStatusID, o.UID, o.Address, o.[Date], o.Note, o.TotalPrice, o.GID, o.Phone FROM [SWP].[dbo].[OrderLog] ol

INNER JOIN [Order] o ON o.OID = ol.OID

WHERE ol.StatusID = (SELECT MAX(StatusID) FROM [SWP].[dbo].[OrderLog] WHERE OID = ol.OID)

GROUP BY ol.OID, o.UID, o.Address, o.[Date], o.Note, o.TotalPrice, o.GID, o.Phone

SELECT COUNT(\*) FROM [OrderLog] Where [StatusID] =1

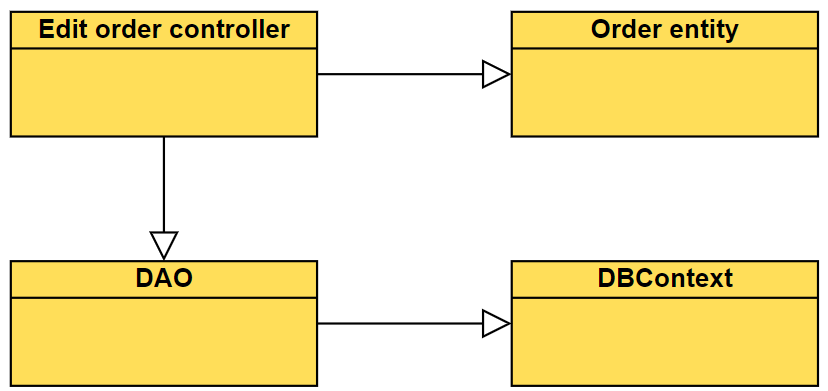
SELECT COUNT(DISTINCT OID) FROM [OrderLog] WHERE [StatusID] = 2

SELECT COUNT(DISTINCT OID) FROM [OrderLog] WHERE [StatusID] = 3

SELECT COUNT(DISTINCT OID) FROM [OrderLog] WHERE [StatusID] = 4

## Update order

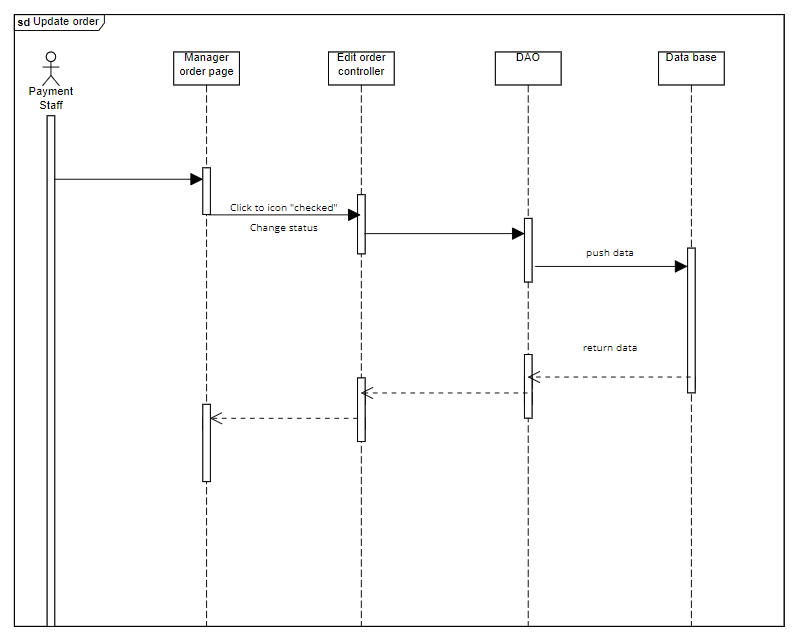
### Class Diagram



### Class Specifications

|  |  |  |
| --- | --- | --- |
| **No** | **Method** | **Description** |
| 01 | editOrder() | This function will update status of order. |

### Sequence Diagram(s)



### d. Database queries

"UPDATE [dbo].[OrderLog]\n"

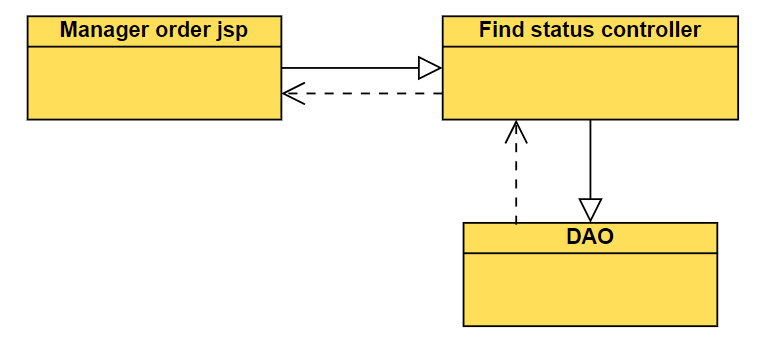
+ " SET\n"

+ " [StatusID] = ?\n"

+ " WHERE [OrderLogId] = ?"

## Search order

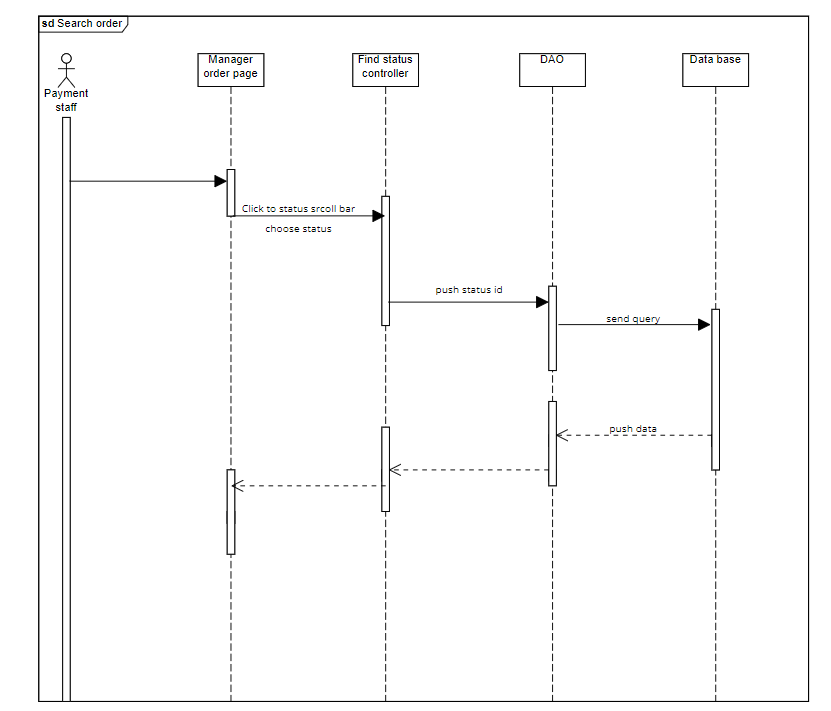
### Class Diagram



### Class Specifications

|  |  |  |
| --- | --- | --- |
| **No** | **Method** | **Description** |
| 01 | getStatusOrder() | This function will return list of order by order status id |

### Sequence Diagram(s)



### d. Database queries

"SELECT [OrderLogId]\n"

+ " ,[OID]\n"

+ " ,[StatusID]\n"

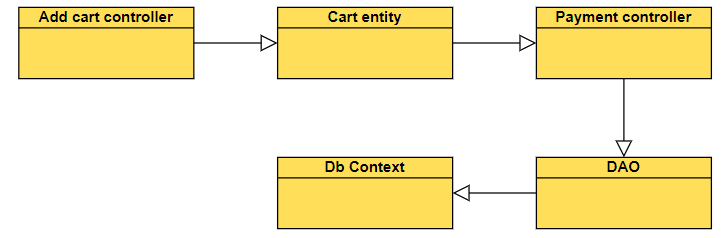
+ " ,[Date]\n"

+ " ,[Confirm]\n"

+ " FROM [SWP].[dbo].[OrderLog] Where [StatusID] = ?"

## Add to cart

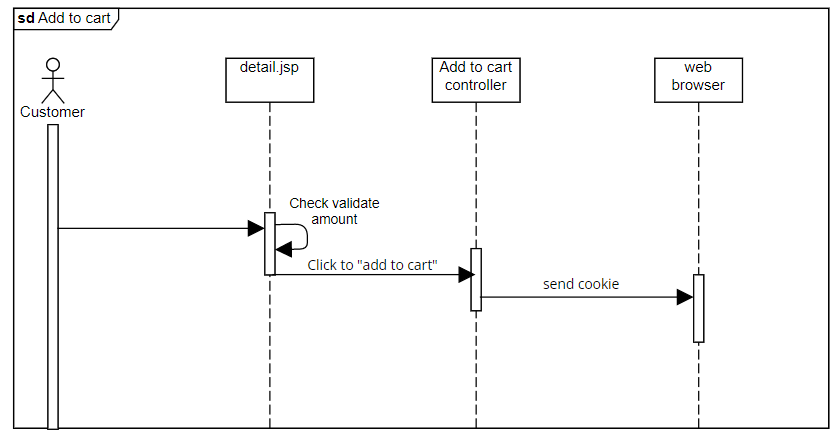
### Class Diagram

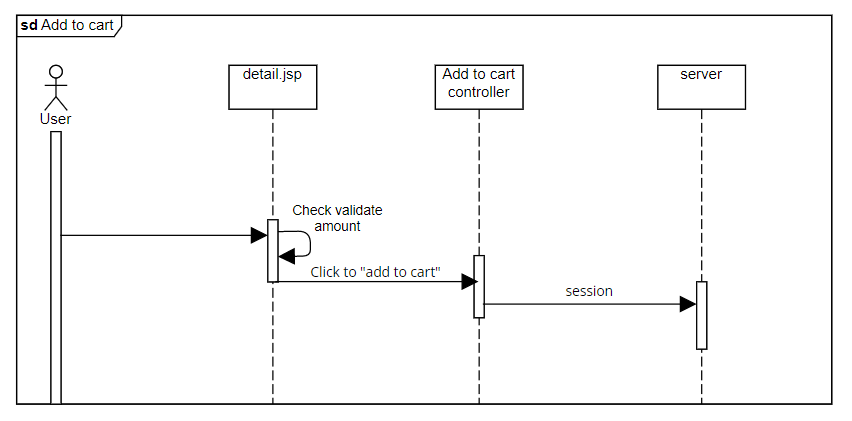


### Class Specifications

|  |  |  |
| --- | --- | --- |
| **No** | **Method** | **Description** |
| 01 | getProductByID() | This function will return list of order by order status id |
| 02 | insertOrderUser() | This function will add information of order into database |
| 03 | getuId() | This function will return user by user id |

### Sequence Diagram(s)





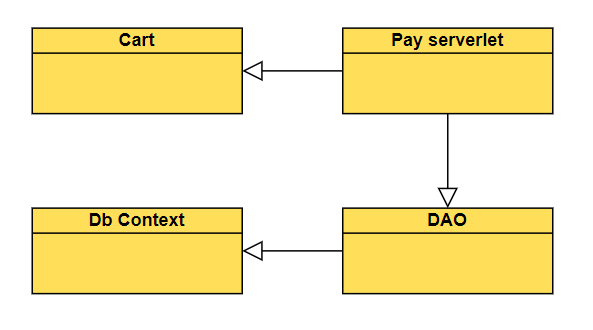
### d. Database queries

"SELECT \* \n"

+ " FROM [SWP].[dbo].[Product] Where PID = ?"

## Purchase

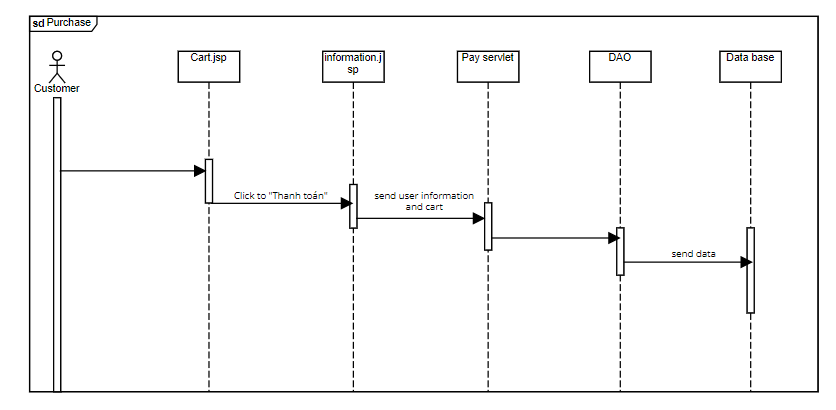
### Class Diagram



### Class Specifications

|  |  |  |
| --- | --- | --- |
| **No** | **Method** | **Description** |
| 01 | Cart() | This function will create list of product in a cart |
| 02 | insertOrderUser() | This function will add information of customer order into database |
| 03 | insertOrder() | This function will add information of user order into database |

### Sequence Diagram(s)



### Database queries

"INSERT INTO [dbo].[OrderLog]\n" +

" ([OID]\n" +

" ,[StatusID]\n" +

" ,[Date]\n" +

" ,[Confirm]) values (?,?,?,?)"

"update product set quantity=quantity-? where PID=?"

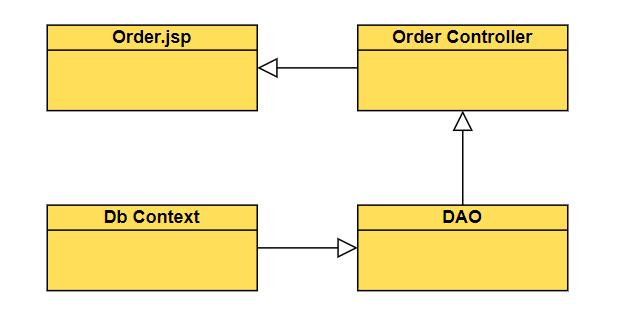
"insert into [Order Detail] (OID,PID,Price,Amount) values (?,?,?,?)"

"Select top 1 OID from [Order] order by OID desc"

"insert into [Order](UID ,Address, Date, Note, TotalPrice) values (?,?,?,?,?)"

## Tracking order

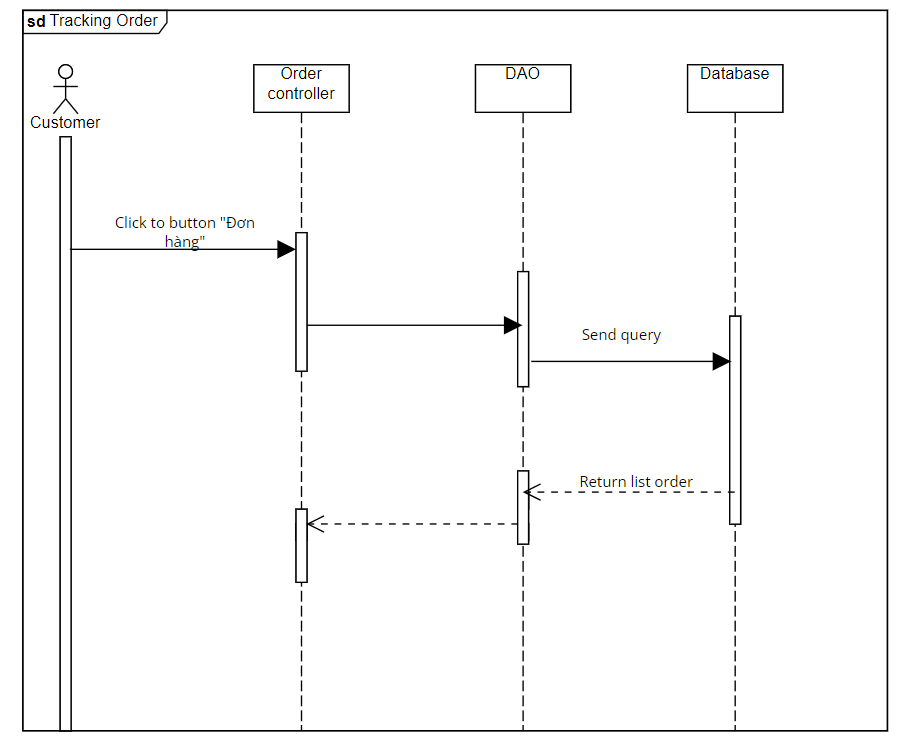
### Class Diagram



### Class Specifications

|  |  |  |
| --- | --- | --- |
| **No** | **Method** | **Description** |
| 01 | getODDTbyUID() | This function will return a list of order by user id |

### c. Sequence Diagram(s)



### Database queries

"select \* from [Order] o\n"

+ "Inner join [Order Detail] od\n"

+ "on od.OID = o.OID\n"

+ "where UID = ?\n"

+ "Order by o.Date Desc"