

# PET SHOP

# Software Design Specification

– Hanoi, March 2023 –

**Record of changeS**

| **Date** | **A\* M, D** | **In charge** | **Change Description** |
| --- | --- | --- | --- |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

\*A - Added M - Modified D - Deleted

**Table of Contents**

[**PET SHOP 1**](#_heading=h.hzh6ub2y50r2)

[**Software Design Specification 1**](#_heading=h.byy173to2y2f)

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

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

[2. Database Design 10](#_heading=h.1fob9te)

[a. Database Schema 10](#_heading=h.3znysh7)

[b. Table Description 11](#_heading=h.2et92p0)

[II. Code Designs 11](#_heading=h.tyjcwt)

[1.View homepage 11](#_heading=h.18vwvbjaf8ul)

[a. Class Diagram 11](#_heading=h.fjmda6x8zv57)

[b. Class Specifications 11](#_heading=h.24jbodg0oxri)

[ProductDAO 12](#_heading=h.os482moo806c)

[Category 12](#_heading=h.xvk85icu8u3)

[Product 12](#_heading=h.s2gow4uwjysw)

[CategoryServices 12](#_heading=h.7a8h9lcxua4r)

[ProductServices 12](#_heading=h.we2l70lhysev)

[c. Sequence Diagram(s) 12](#_heading=h.fddt7ku85rp5)

[d. Database Queries 13](#_heading=h.2bbeiw87yc70)

[2.Login feature 13](#_heading=h.laynx0ry1crx)

[a. Class Diagram 13](#_heading=h.5wrphutb3u80)

[b. Class Specifications 13](#_heading=h.zi5nla7p5fqj)

[Login 13](#_heading=h.o0odb950r7v9)

[AccountDAO 13](#_heading=h.ye4zb7n29hxc)

[Account 13](#_heading=h.4mi0ri8hlryq)

[c. Sequence Diagram(s) 14](#_heading=h.3ljvtrbybnrx)

[d. Database Queries 14](#_heading=h.jozyp6wzh2vb)

[3.Logout 14](#_heading=h.qhl2nx6dafhs)

[a. Class Diagram 14](#_heading=h.b7zma6w47il5)

[b. Class Specifications 14](#_heading=h.g2ekjttftblv)

[Logout 14](#_heading=h.c1slvw537sg2)

[c. Sequence Diagram(s) 14](#_heading=h.54yqe0xmmt2y)

[4. Register feature 15](#_heading=h.m8kg6q9bpd7b)

[a. Class Diagram 15](#_heading=h.mnprskasoov7)

[b. Class Specifications 15](#_heading=h.dkgaijel1nuz)

[RegisterController 15](#_heading=h.yr4f2gxu9lx6)

[AccountService 16](#_heading=h.oa1w1wizjvot)

[CustomerService 16](#_heading=h.sbc4rk4l9906)

[CustomerDao 16](#_heading=h.qkfk9735v0p8)

[AccountDao 16](#_heading=h.og7osgn9bwe3)

[AccountMapping 16](#_heading=h.eq4ki05buand)

[CustomerMapping 16](#_heading=h.8l4ygvjnq3ej)

[Account 16](#_heading=h.8br8dglrw59f)

[Customer 17](#_heading=h.vhlo5rwmwb1v)

[c. Sequence Diagram(s) 17](#_heading=h.8ad0h4hqytuc)

[d. Database Queries 17](#_heading=h.eokf11d2iddj)

[5. Forgot password 17](#_heading=h.as7w7u3r5usl)

[a. Class Diagram 17](#_heading=h.b162xg83atfh)

[b. Class Specifications 17](#_heading=h.3fuxb44wtqfo)

[XYZ Class 17](#_heading=h.m4qmhi6dzob0)

[ABC Class 17](#_heading=h.nblb98xnfjxt)

[c. Sequence Diagram(s) 18](#_heading=h.wv1czfeijlnp)

[d. Database Queries 18](#_heading=h.rwxp98ur1lv)

[6. Product detail 18](#_heading=h.4wll0oa9i7sk)

[a. Class Diagram 18](#_heading=h.ad51m5z1146u)

[b. Class Specifications 18](#_heading=h.nv4ytztuxb7y)

[XYZ Class 18](#_heading=h.hjzgxagh8bux)

[ABC Class 18](#_heading=h.r6gjau7vuvvt)

[c. Sequence Diagram(s) 18](#_heading=h.pnkl5abq0tb3)

[d. Database Queries 19](#_heading=h.oa7x5nza54h0)

[7. Search 19](#_heading=h.axtxorv868xk)

[a. Class Diagram 19](#_heading=h.u8v2go33pee2)

[b. Class Specifications 19](#_heading=h.41kvbw7o89hl)

[XYZ Class 19](#_heading=h.snrnxx3crr9)

[ABC Class 19](#_heading=h.x7a2opnvdjqw)

[c. Sequence Diagram(s) 19](#_heading=h.249f5tgse0a7)

[d. Database Queries 19](#_heading=h.z6oaum8nhi6s)

[8 .View Profile feature 20](#_heading=h.2adu4dw801mw)

[a. Class Diagram 20](#_heading=h.r6bszzmpzx7y)

[b. Class Specifications 20](#_heading=h.704xd2sv3orq)

[ProfileController 20](#_heading=h.skguf2vcerqp)

[AccountService 20](#_heading=h.xalxtjtjhd6i)

[CustomerService 21](#_heading=h.njicdimg6vi8)

[CustomerDao 21](#_heading=h.4oa0f11343ol)

[AccountDao 21](#_heading=h.gc6n6vwkhvwc)

[AccountMapping 21](#_heading=h.zbz4k5lrl3n0)

[CustomerMapping 21](#_heading=h.bdag07m6whgy)

[Account 21](#_heading=h.2337g57nnzwl)

[Customer 22](#_heading=h.43yu8hnm957o)

[c. Sequence Diagram(s) 22](#_heading=h.i6c5u13c7bfc)

[d. Database Queries 22](#_heading=h.ffbio0m34rkd)

[9. Update Profile 22](#_heading=h.wa8s2h2r1zd1)

[a. Class Diagram 22](#_heading=h.pxrk6tlkfqbq)

[b. Class Specifications 22](#_heading=h.medlpw145reu)

[ChangeProfileController 22](#_heading=h.wgkzw4l1oeq6)

[AccountService 23](#_heading=h.hcc793npqb3j)

[CustomerService 23](#_heading=h.f4knim1mk9wj)

[CustomerDao 23](#_heading=h.iyb2hmy27tsp)

[AccountDao 23](#_heading=h.mi2anfkb3j8s)

[AccountMapping 23](#_heading=h.cofppbqnrjl4)

[CustomerMapping 23](#_heading=h.sj8zj6f0bngw)

[Account 23](#_heading=h.4v9auuwi4jt8)

[Customer 24](#_heading=h.bmi9m6qti6r)

[c. Sequence Diagram(s) 24](#_heading=h.w6iqhv2qdi4)

[d. Database Queries 24](#_heading=h.d1fea6ajtr1j)

[10. Change Password 24](#_heading=h.m4f2slrwln9o)

[a. Class Diagram 24](#_heading=h.u2l16idyp7rd)

[b. Class Specifications 24](#_heading=h.nd0tk02f5ezl)

[ChangePasswordController 24](#_heading=h.34m0oaq3xps8)

[AccountService 25](#_heading=h.1fj02ch7fat1)

[CustomerDao 25](#_heading=h.n8vat25j9feh)

[AccountDao 25](#_heading=h.18td8c6uupi9)

[AccountMapping 25](#_heading=h.edcjxmawulrb)

[Account 25](#_heading=h.tezi6wahcgn9)

[c. Sequence Diagram(s) 25](#_heading=h.74l00iavjzxf)

[d. Database Queries 25](#_heading=h.ml663v28xngi)

[11. Managed Order 26](#_heading=h.qjcbnfmgmm5z)

[a. Class Diagram 26](#_heading=h.6h3jcdzalc02)

[b. Class Specifications 26](#_heading=h.7bk5mhc6q8z2)

[Order Class 26](#_heading=)

[ABC Class 26](#_heading=h.yx8yrqvzsozr)

[c. Sequence Diagram(s) 27](#_heading=h.gxe4vld3g75v)

[d. Database Queries 27](#_heading=h.jr5kc7x4tcui)

[12. Create Order 27](#_heading=h.ljlfubvtz1pg)

[a. Class Diagram 27](#_heading=h.j28fcq5h380s)

[b. Class Specifications 28](#_heading=h.wnm22w9k7qft)

[Order Class 28](#_heading=h.fxjqt7q43un4)

[ABC Class 28](#_heading=h.nuwozs2mwzcf)

[c. Sequence Diagram(s) 28](#_heading=h.b92ug4vbbrp)

[d. Database Queries 28](#_heading=h.xyfppqxd6j9v)

[13. View Order 29](#_heading=h.i3vb2w6kcn80)

[a. Class Diagram 29](#_heading=h.bqhh9woc3clm)

[b. Class Specifications 29](#_heading=h.40riqy3m7185)

[Order Class 29](#_heading=h.fgwhmwxv1frm)

[ABC Class 29](#_heading=h.kl26a9u3focq)

[c. Sequence Diagram(s) 30](#_heading=h.lvmzrw6a1mwm)

[d. Database Queries 30](#_heading=h.2y7eugh79h5r)

[14. Update Order 30](#_heading=h.5326hjstqdb)

[a. Class Diagram 30](#_heading=h.tsu80klm9wbg)

[b. Class Specifications 30](#_heading=h.i0vczlc40nyu)

[Order Class 30](#_heading=h.emo1c417ofga)

[ABC Class 31](#_heading=h.tkrv618wl859)

[c. Sequence Diagram(s) 31](#_heading=h.5fnmrjnvj178)

[d. Database Queries 31](#_heading=h.z9p7i0df24vf)

[15. Cancel Order 32](#_heading=h.9h8thbirugp5)

[a. Class Diagram 32](#_heading=h.yeet89x0uof5)

[b. Class Specifications 32](#_heading=h.peikmbfopc47)

[Order Class 32](#_heading=h.csc92lw390qi)

[ABC Class 32](#_heading=h.ubpspt28s352)

[c. Sequence Diagram(s) 33](#_heading=h.pxljy5abqp27)

[d. Database Queries 33](#_heading=h.opfwunc30w0i)

[16. Add to Cart 33](#_heading=h.fzd7sc5ilcre)

[a. Class Diagram 33](#_heading=h.hty5qxg7zdbh)

[b. Class Specifications 33](#_heading=h.c0ay27kfajin)

[Order Class 33](#_heading=h.ycuzcmugi6mo)

[ABC Class 34](#_heading=h.fx1hldrdruf3)

[c. Sequence Diagram(s) 34](#_heading=h.qlhhs5zg5bi3)

[d. Database Queries 34](#_heading=h.wq4yjq6mhbyd)

[17. View Cart 35](#_heading=h.73mcytx9i7sz)

[a. Class Diagram 35](#_heading=h.2mnjp14nmw7g)

[b. Class Specifications 35](#_heading=h.mybx9p61w1kv)

[AddCartController 35](#_heading=h.q534vkg17bkj)

[Cart Service 35](#_heading=h.terp2n3a0tx2)

[Product Service 36](#_heading=h.cchdulpol0qi)

[CartDAO 36](#_heading=h.qyv70pi73ewg)

[CartItemDAO 36](#_heading=h.jnczjrrv7f6b)

[ProductDAO 36](#_heading=h.i9kyfssf9102)

[CartMapping 36](#_heading=h.i7l3ikr0t1fw)

[CartItemMapping 36](#_heading=h.dopep4sukhsy)

[ProductMapping 36](#_heading=h.22ikdplkbcmp)

[Cart 37](#_heading=h.flbt5llxwa87)

[c. Sequence Diagram(s) 37](#_heading=h.yzek8xlm0mq)

[d. Database Queries 37](#_heading=h.q068z9se7b2t)

[18. Update cart 38](#_heading=h.3dy6vkm)

[a. Class Diagram 38](#_heading=h.1t3h5sf)

[b. Class Specifications 38](#_heading=h.4d34og8)

[XYZ Class 38](#_heading=)

[ABC Class 38](#_heading=h.r176lu6yosw1)

[c. Sequence Diagram(s) 38](#_heading=h.2s8eyo1)

[d. Database Queries 38](#_heading=h.zh2lwmrwfpf1)

[19. Clear cart 39](#_heading=h.q9lvz4l8u164)

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

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

[XYZ Class 39](#_heading=h.e1bnmyykr27q)

[ABC Class 39](#_heading=h.w8zhnnrrtqqz)

[c. Sequence Diagram(s) 39](#_heading=h.weesg4nqmszt)

[d. Database Queries 39](#_heading=h.a0e9n3ifbjuh)

[20. Managed account 40](#_heading=h.xsi6k97gdt5s)

[a. Class Diagram 40](#_heading=h.xal84ktp6aoe)

[b. Class Specifications 40](#_heading=h.vdezrqhty69i)

[XYZ Class 40](#_heading=h.3ldhzy35dafh)

[ABC Class 40](#_heading=h.s3sc67alezf4)

[c. Sequence Diagram(s) 40](#_heading=h.izsrpfi1tbd4)

[d. Database Queries 40](#_heading=h.39sestabi84h)

[21. Add account 41](#_heading=h.wx5afra2jzg0)

[a. Class Diagram 41](#_heading=h.8t50wn66o44o)

[b. Class Specifications 41](#_heading=h.6iceb4mmerku)

[XYZ Class 41](#_heading=h.npe09oh1f6xp)

[ABC Class 41](#_heading=h.elfsquskhyw4)

[c. Sequence Diagram(s) 41](#_heading=h.jdgo6aoj8g97)

[d. Database Queries 41](#_heading=h.bd3yv5g4x1xc)

[22. View account 42](#_heading=h.t1oostmjvc3s)

[a. Class Diagram 42](#_heading=h.k4hwvj4xolbr)

[b. Class Specifications 42](#_heading=h.66y383in7jhs)

[XYZ Class 42](#_heading=h.lszfdm6tcet5)

[ABC Class 42](#_heading=h.di6pfm9crm0x)

[c. Sequence Diagram(s) 42](#_heading=h.yixlzrotavow)

[d. Database Queries 42](#_heading=h.9lqmswwnjh6q)

[23. Delete account 43](#_heading=h.ptvlsidya5up)

[a. Class Diagram 43](#_heading=h.strfx86j4e9w)

[b. Class Specifications 43](#_heading=h.itvgoab3yqtc)

[XYZ Class 43](#_heading=h.76t61z8h42gh)

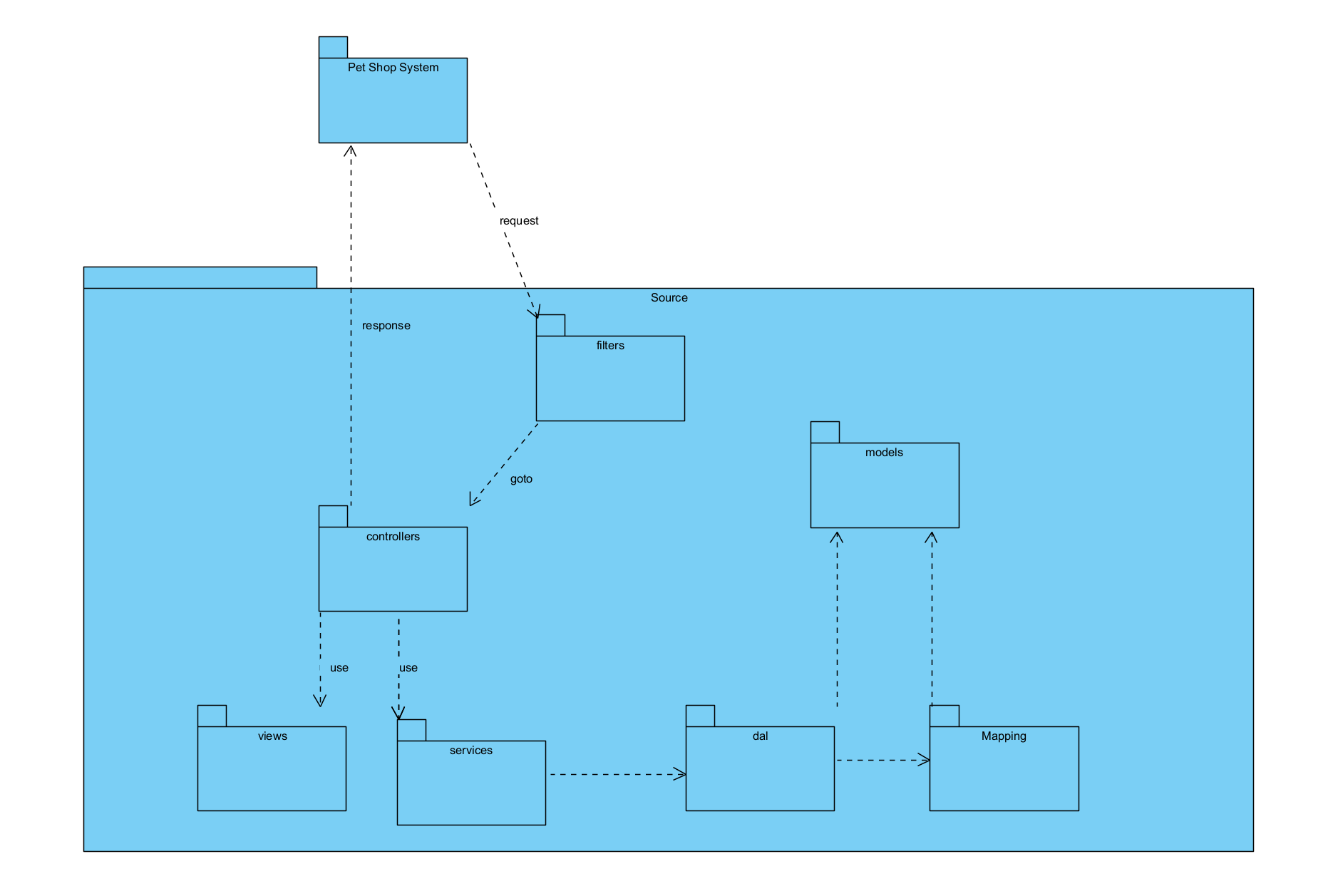
[ABC Class 43](#_heading=h.wllu0m1l6gse)

[c. Sequence Diagram(s) 43](#_heading=h.4gvafgpq3m8m)

[d. Database Queries 43](#_heading=h.l8jglfko823l)

# I. Overview

## 1. Code Packages

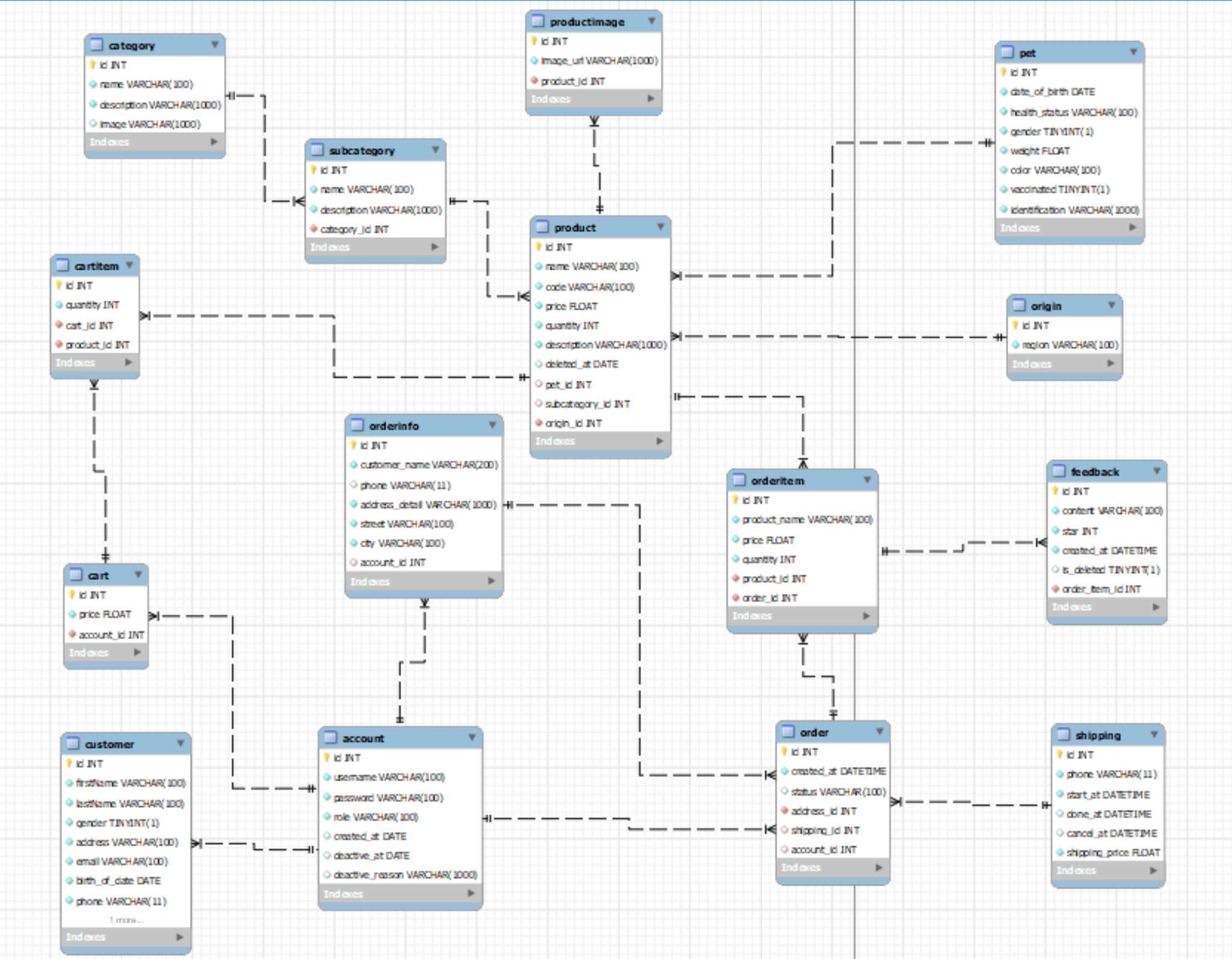


***Package descriptions***

| **No** | **Package** | **Description** |
| --- | --- | --- |
| 01 | AppConfig | This component is used to provides a convenient way to define and access session and cookie names in a web application |
| 02 | controllers | This component contains login, logout,register and display all front-end common features . |
| 03 | controllers.account | This component is used to change password of users |
| 04 | controllers.cart | This component provide functionality for managing a shopping cart |
| 05 | controllers.category | This component is designed to contain list category and subcategory |
| 06 | controllers.customer | This component is designed to provide functionality for managing self-user information |
| 07 | controllers.feedback | This component is designed to provide functionality for managing customer feedback |
| 08 | controllers.order | This component is designed to provide functionality for managing order |
| 09 | controllers.orderInfo | This component is designed to provide functionality for managing order of customer |
| 10 | controllers.product | This component is designed to display product detail |
| 11 | controllers.search | This component is designed to provide functionality for searching |
| 12 | controllers.store | This component is designed to provide functionality for display product |
| 13 | dal | This component is used to get data directly FROM the database |
| 14 | mapping | This component provides a way to map models to database records and vice versa. |
| 15 | models | This component includes interfaces used to receive data retrieved FROM the database and classes used to receive data |
| 16 | services | This component is used to process data after it is retrieved from the database. (dal) |

## 2. Database Design

### a. Database Schema



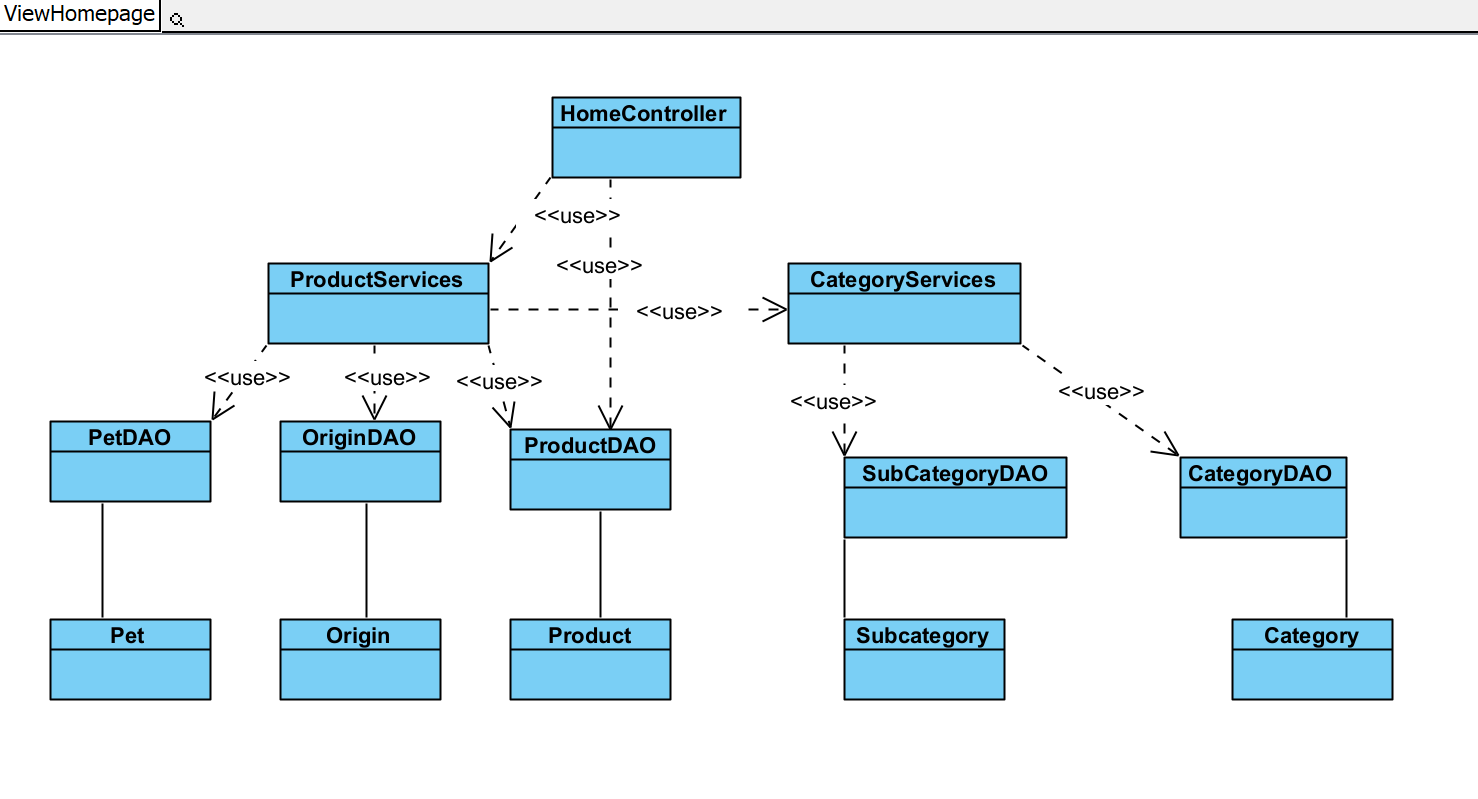
### b. Table Description

| **No** | **Table** | **Description** |
| --- | --- | --- |
| 01 | account | The registered Users, Employee, Admin of the system |
| 02 | order | Contains information about order |
| 03 | orderinfo | Contains information about order and customer |
| 04 | orderitem | Contains information about product customer buy |
| 05 | shipping | Contains information about shipping status |
| 06 | feedback | Contains information about feedback of customer after bought |
| 07 | customer | Contains information about Users that have an account |
| 08 | product | Contains information about product |
| 09 | origin | Contains information about region of that product |
| 10 | pet | Contains information about pet |
| 11 | productimage | Contains images of each product |
| 12 | cart | Contains information about cart |
| 13 | cart item | Contains item in cart |
| 14 | subcategory | Contains information about subcategory |
| 15 | category | Contains information on subcategory and category |

# II. Code Designs

## 1.View homepage

### a. Class Diagram



### b. Class Specifications

**HomeController**

| **No** | **Method** | **Description** |
| --- | --- | --- |
| 01 | doGet() | initializes some services and a data access object, receives and parses a parameter from the request, retrieves a list of products based on this parameter, retrieves a list of categories, sets some attributes to be used in the JSP page, and forwards the request and response objects to the home.jsp view. |

#### ProductDAO

| **No** | **Method** | **Description** |
| --- | --- | --- |
| 01 | List<Product> getAllProduct() | Gets all products from a database table called "Product" using JDBC, maps the results to Product objects, and returns them as a List. Any errors are logged, and there's a commented out section to close the database connection after use. |
| 02 | pagingProductsByRandom(String code) | Takes in a String parameter code and returns a list of Product objects. The method connects to a database, executes an SQL query that selects products based on a code pattern, orders the results in a random manner and limits the number of results to 5 |

#### Category

| **No** | **Method** | **Description** |
| --- | --- | --- |
| 01 | Category() | Represents a category object. |
| 02 | toString() | Generating a string representation of the object. |

#### Product

| **No** | **Method** | **Description** |
| --- | --- | --- |
| 01 | Product | Represents a product object. |

#### CategoryServices

| **No** | **Method** | **Description** |
| --- | --- | --- |
| 01 | getSubAndCategory() | uses two DAO objects to retrieve data from a database. The constructor initializes the two DAO objects. |

#### ProductServices

| **No** | **Method** | **Description** |
| --- | --- | --- |
| *01* | *<method name>* | *<Description of the method, including the inputs, outputs & internal method processing>* |
|  |  |  |

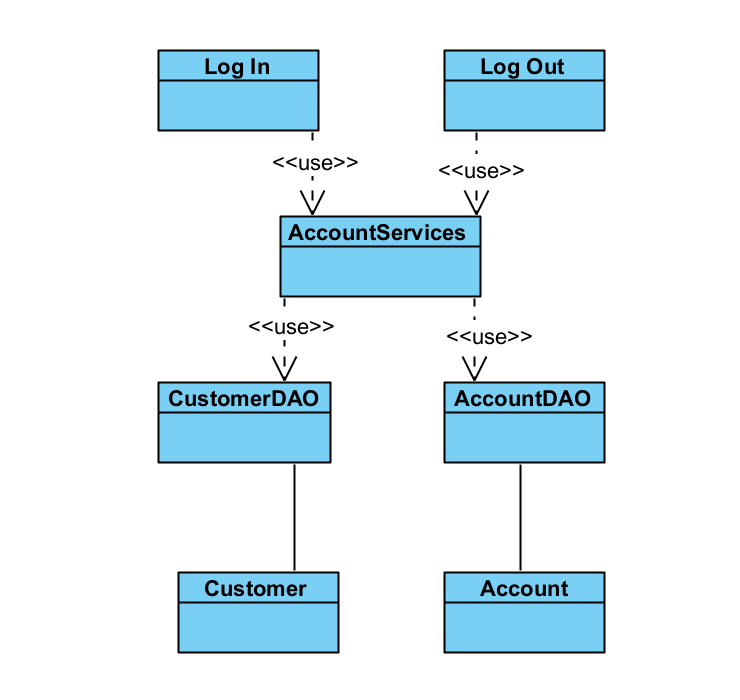
### c. Sequence Diagram(s)

### d. Database Queries

* select \* from shop\_pet.product spp inner join shop\_pet.pet spp2 on spp.pet\_id=spp2.id where spp.code like ? ORDER BY RAND() LIMIT 5
* select\*from shop\_pet.product spp inner join shop\_pet.subcategory sps on spp.subcategory\_id = sps.id inner join shop\_pet.pet spp2 on spp.pet\_id = spp2.id where spp.subcategory\_id = ? and spp.id != ?
* select\*from shop\_pet.product spp inner join shop\_pet.subcategory sps \n"
* + "on spp.subcategory\_id = sps.id where spp.subcategory\_id = ? and spp.id !=
* SELECT \*\n"
* + " FROM shop\_pet.product spp2 inner join shop\_pet.Pet spp on spp.id=spp2.pet\_id\n"
* + " inner join shop\_pet.origin spo on spp2.origin\_id=spo.id \n"
* + " inner join shop\_pet.subcategory sps on sps.id=spp2.subcategory\_id\n"
* + " inner join shop\_pet.category spc on sps.category\_id=spc.id \n"
* + " where spc.id = ?\n"
* + " limit ? offset ?

## 2.Login feature

### a. Class Diagram



### b. Class Specifications

#### Login

| **No** | **Method** | **Description** |
| --- | --- | --- |
| 01 | doGet() | Checks if a session attribute is present, and if it is, redirects the user to the home page. If the attribute is not present, the user is forwarded to the login page |
| 02 | doPost() | Handles a user login request, It retrieves the username and password from the request parameters, creates cookies with them. If the login is successful, it sets the user object in the session attribute and redirects the user to the home page. If there is an exception during the login process, it sets an error message and forwards the user to the login page. |

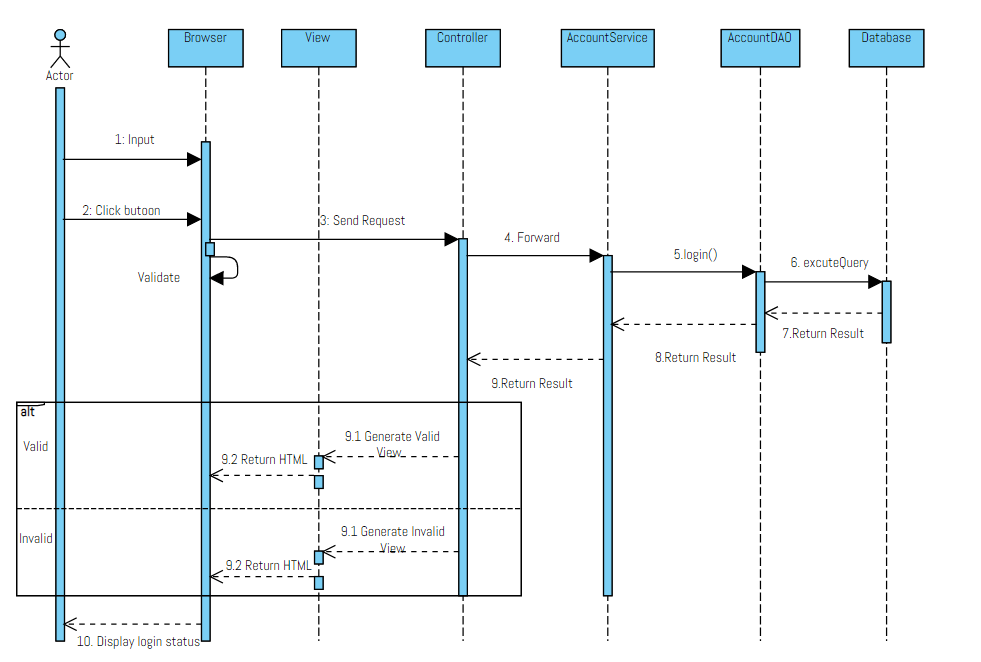
#### AccountDAO

| **No** | **Method** | **Description** |
| --- | --- | --- |
| 01 | login(usename: String) | Attempts to log in an account by querying the database for a row where the username and password match the given parameters. If a matching row is found, it converts it into an Account object and returns it. If there is an exception during the process, it returns null. |

#### Account

| **No** | **Method** | **Description** |
| --- | --- | --- |
| 01 | Account() | The Account class is a simple Java bean that represents an account in the system. It has private instance variables for the ID, username, password, role, created date, deactive date and reason, as well as a reference to its associated Customer object. The class provides getters and setters for all of its instance variables, allowing them to be accessed and modified from outside the class. |
| 02 | toString() | Generating a string representation of the object. |

### c. Sequence Diagram(s)



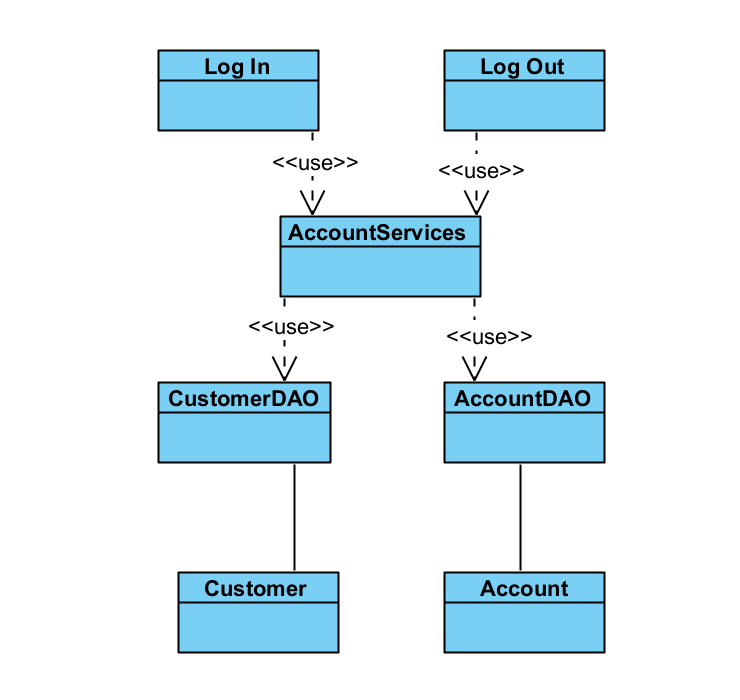
### d. Database Queries

*SELECT \* FROM Account WHERE username = ? AND password = ?*

## 

## 3.Logout

### a. Class Diagram

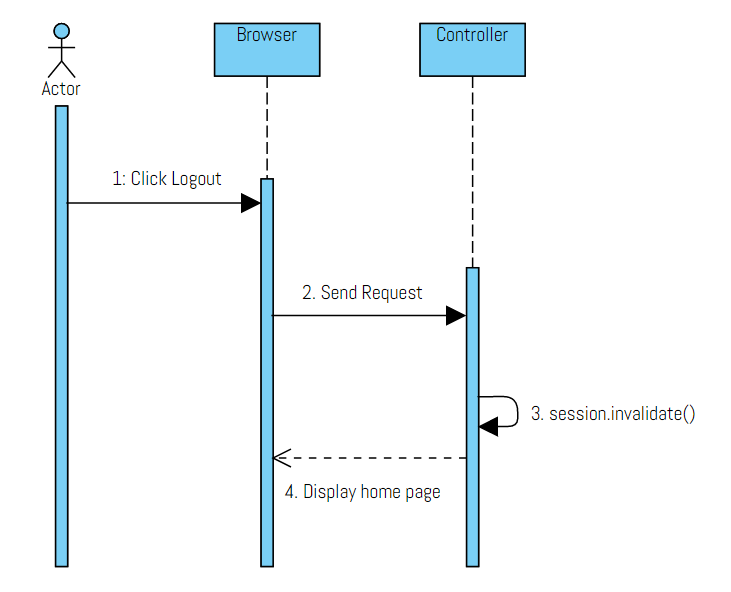


### b. Class Specifications

#### Logout

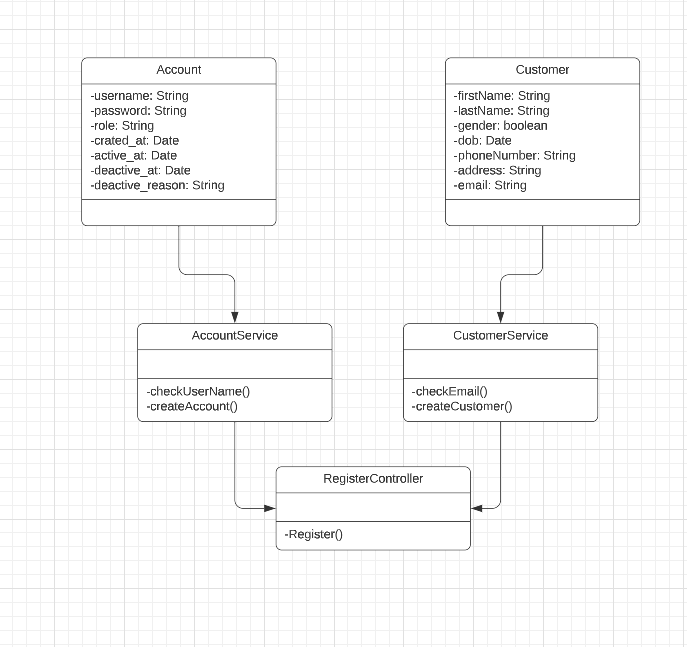
| **No** | **Method** | **Description** |
| --- | --- | --- |
| 01 | session.invalidate() | Destroys the current session and frees up any resources associated with it. |

### c. Sequence Diagram(s)



## 4. Register feature

### a. Class Diagram



### b. Class Specifications

#### RegisterController

| **No** | **Method** | **Description** |
| --- | --- | --- |
| 01 | doPost | handles registration form submissions. It checks for validation errors, creates new Account and Customer objects, and saves them to the database using AccountService and CustomerService. If there are errors, it displays error messages and prompts users to fix their inputs. Otherwise, it redirects users to the login page. Any errors encountered along the way are caught and printed to the response writer. |
| 02 | doGet | forwards the current request and response objects to the registration page named "register.jsp". The forward method is used to transfer control from one servlet to another during the execution of a request. |

#### AccountService

| **No** | **Method** | **Description** |
| --- | --- | --- |
| *01* | *<method name>* | *<Description of the method, including the inputs, outputs & internal method processing>* |
|  |  |  |

#### 

#### CustomerService

| **No** | **Method** | **Description** |
| --- | --- | --- |
| *01* | *<method name>* | *<Description of the method, including the inputs, outputs & internal method processing>* |
|  |  |  |

#### 

#### CustomerDao

| **No** | **Method** | **Description** |
| --- | --- | --- |
| *01* | *<method name>* | *<Description of the method, including the inputs, outputs & internal method processing>* |
|  |  |  |

#### AccountDao

| **No** | **Method** | **Description** |
| --- | --- | --- |
| *01* | *<method name>* | *<Description of the method, including the inputs, outputs & internal method processing>* |
|  |  |  |

#### AccountMapping

| **No** | **Method** | **Description** |
| --- | --- | --- |
| *01* | *<method name>* | *<Description of the method, including the inputs, outputs & internal method processing>* |
|  |  |  |

#### CustomerMapping

| **No** | **Method** | **Description** |
| --- | --- | --- |
| *01* | *<method name>* | *<Description of the method, including the inputs, outputs & internal method processing>* |
|  |  |  |

#### Account

| **No** | **Method** | **Description** |
| --- | --- | --- |
| *01* | *<method name>* | *<Description of the method, including the inputs, outputs & internal method processing>* |
|  |  |  |

#### Customer

| **No** | **Method** | **Description** |
| --- | --- | --- |
| *01* | *<method name>* | *<Description of the method, including the inputs, outputs & internal method processing>* |
|  |  |  |

### c. Sequence Diagram(s)

### d. Database Queries

## 5. Forgot password

### a. Class Diagram

### b. Class Specifications

#### XYZ Class

| **No** | **Method** | **Description** |
| --- | --- | --- |
| *01* | *<method name>* | *<Description of the method, including the inputs, outputs & internal method processing>* |
|  |  |  |

#### ABC Class

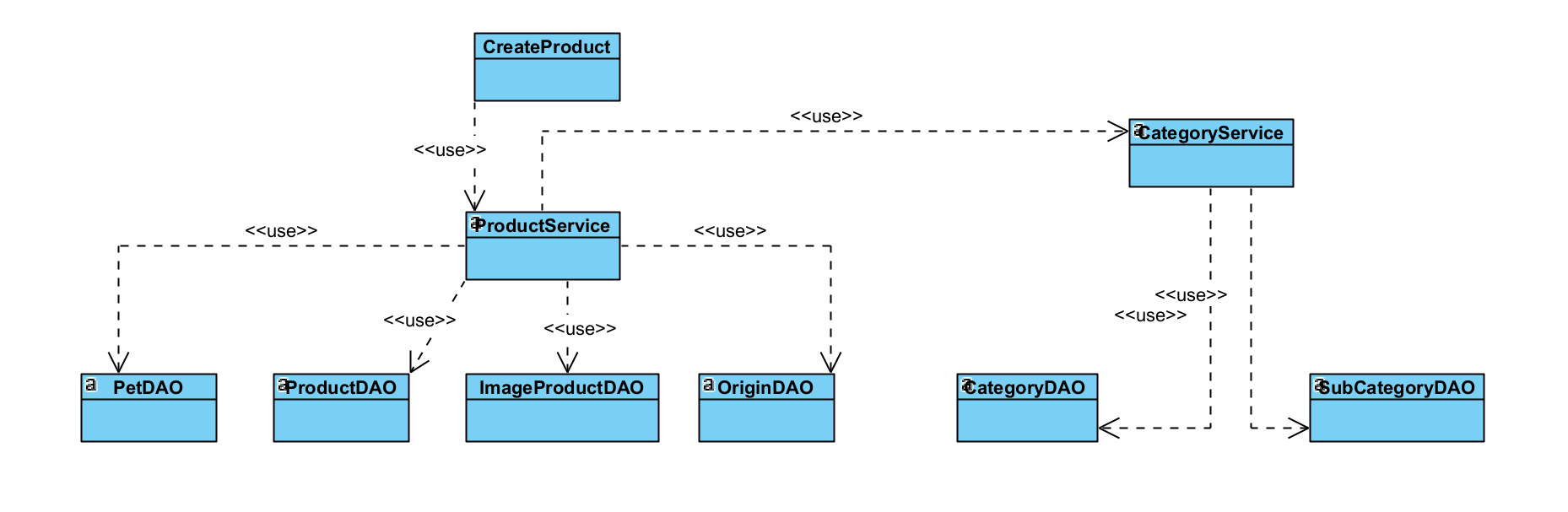
| **No** | **Method** | **Description** |
| --- | --- | --- |
| *01* | *<method name>* | *<Description of the method, including the inputs, outputs & internal method processing>* |
|  |  |  |

### c. Sequence Diagram(s)

### d. Database Queries

## 6. Create Product

### a. Class Diagram



### b. Class Specifications

#### ProductDAO

| **No** | **Method** | **Description** |
| --- | --- | --- |
| *01* | *<method name>* | *<Description of the method, including the inputs, outputs & internal method processing>* |
|  |  |  |

#### PetDAO

| **No** | **Method** | **Description** |
| --- | --- | --- |
| *01* | *<method name>* | *<Description of the method, including the inputs, outputs & internal method processing>* |
|  |  |  |

#### OriginDAO

| **No** | **Method** | **Description** |
| --- | --- | --- |
| *01* | *<method name>* | *<Description of the method, including the inputs, outputs & internal method processing>* |
|  |  |  |

#### ProductService

| **No** | **Method** | **Description** |
| --- | --- | --- |
| *01* | *<method name>* | *<Description of the method, including the inputs, outputs & internal method processing>* |
|  |  |  |

### c. Sequence Diagram(s)

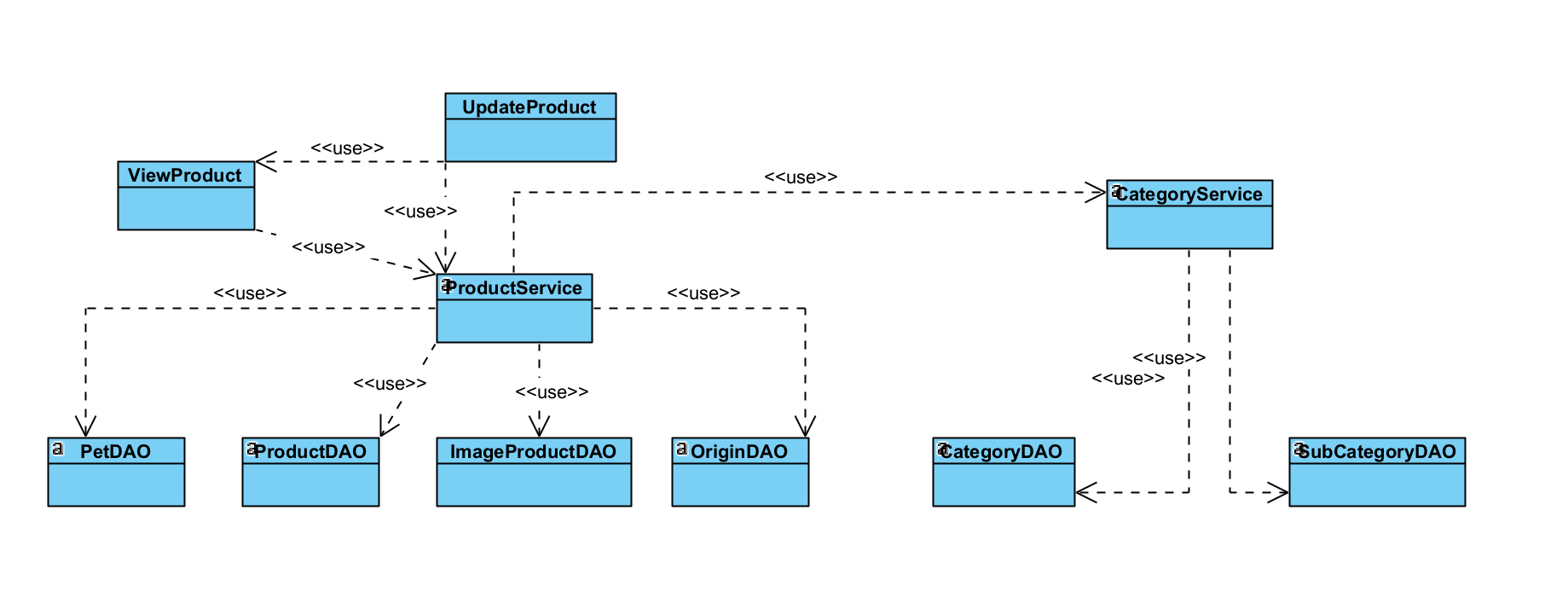
### d. Database Queries

## 

## 

## 7. Update Product

### a. Class Diagram



### b. Class Specifications

#### ProductDAO

| **No** | **Method** | **Description** |
| --- | --- | --- |
| *01* | *<method name>* | *<Description of the method, including the inputs, outputs & internal method processing>* |
|  |  |  |

#### PetDAO

| **No** | **Method** | **Description** |
| --- | --- | --- |
| *01* | *<method name>* | *<Description of the method, including the inputs, outputs & internal method processing>* |
|  |  |  |

#### OriginDAO

| **No** | **Method** | **Description** |
| --- | --- | --- |
| *01* | *<method name>* | *<Description of the method, including the inputs, outputs & internal method processing>* |
|  |  |  |

#### ProductService

| **No** | **Method** | **Description** |
| --- | --- | --- |
| *01* | *<method name>* | *<Description of the method, including the inputs, outputs & internal method processing>* |
|  |  |  |

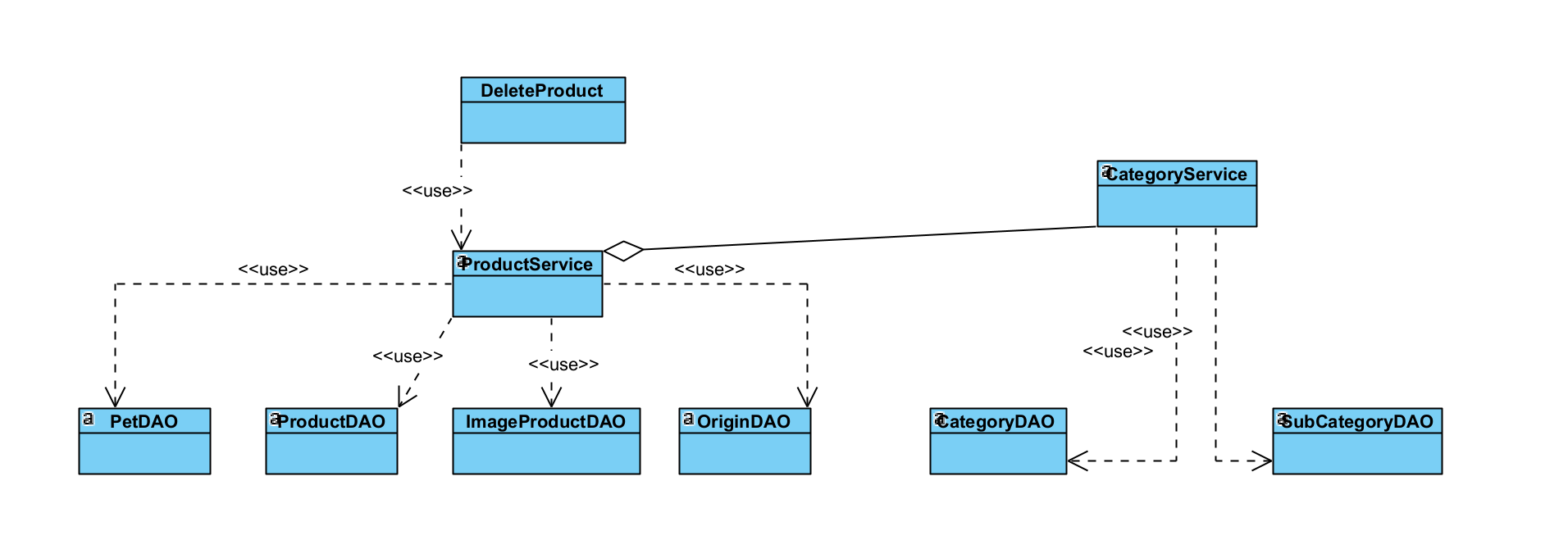
### c. Sequence Diagram(s)

### d. Database Queries

### 

## 8. Delete Product

### a. Class Diagram



### b. Class Specifications

#### ProductDAO

| **No** | **Method** | **Description** |
| --- | --- | --- |
| *01* | *<method name>* | *<Description of the method, including the inputs, outputs & internal method processing>* |
|  |  |  |

#### PetDAO

| **No** | **Method** | **Description** |
| --- | --- | --- |
| *01* | *<method name>* | *<Description of the method, including the inputs, outputs & internal method processing>* |
|  |  |  |

#### OriginDAO

| **No** | **Method** | **Description** |
| --- | --- | --- |
| *01* | *<method name>* | *<Description of the method, including the inputs, outputs & internal method processing>* |
|  |  |  |

#### ProductService

| **No** | **Method** | **Description** |
| --- | --- | --- |
| *01* | *<method name>* | *<Description of the method, including the inputs, outputs & internal method processing>* |
|  |  |  |

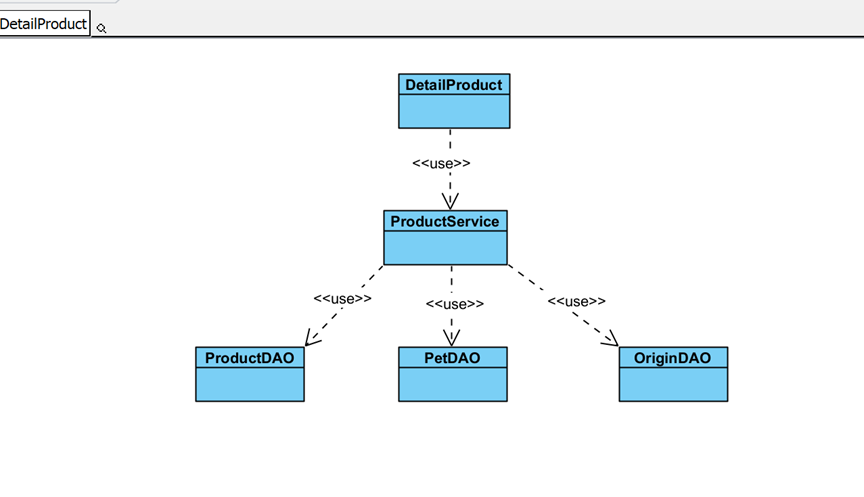
### c. Sequence Diagram(s)

### d. Database Queries

## 

## 9. Product detail

### a. Class Diagram



### b. Class Specifications

#### ProductDAO

| **No** | **Method** | **Description** |
| --- | --- | --- |
| *01* | *<method name>* | *<Description of the method, including the inputs, outputs & internal method processing>* |
|  |  |  |

#### PetDAO

| **No** | **Method** | **Description** |
| --- | --- | --- |
| *01* | *<method name>* | *<Description of the method, including the inputs, outputs & internal method processing>* |
|  |  |  |

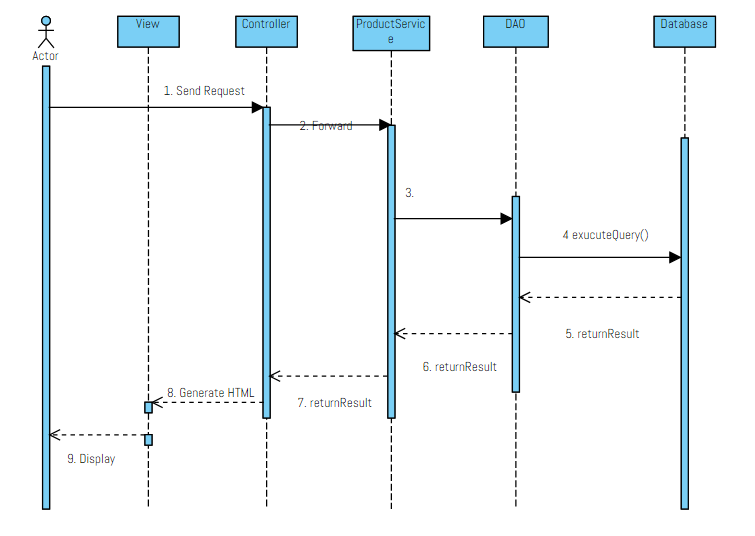
#### OriginDAO

| **No** | **Method** | **Description** |
| --- | --- | --- |
| *01* | *<method name>* | *<Description of the method, including the inputs, outputs & internal method processing>* |
|  |  |  |

#### ProductService

| **No** | **Method** | **Description** |
| --- | --- | --- |
| *01* | *<method name>* | *<Description of the method, including the inputs, outputs & internal method processing>* |
|  |  |  |

### c. Sequence Diagram(s)

**

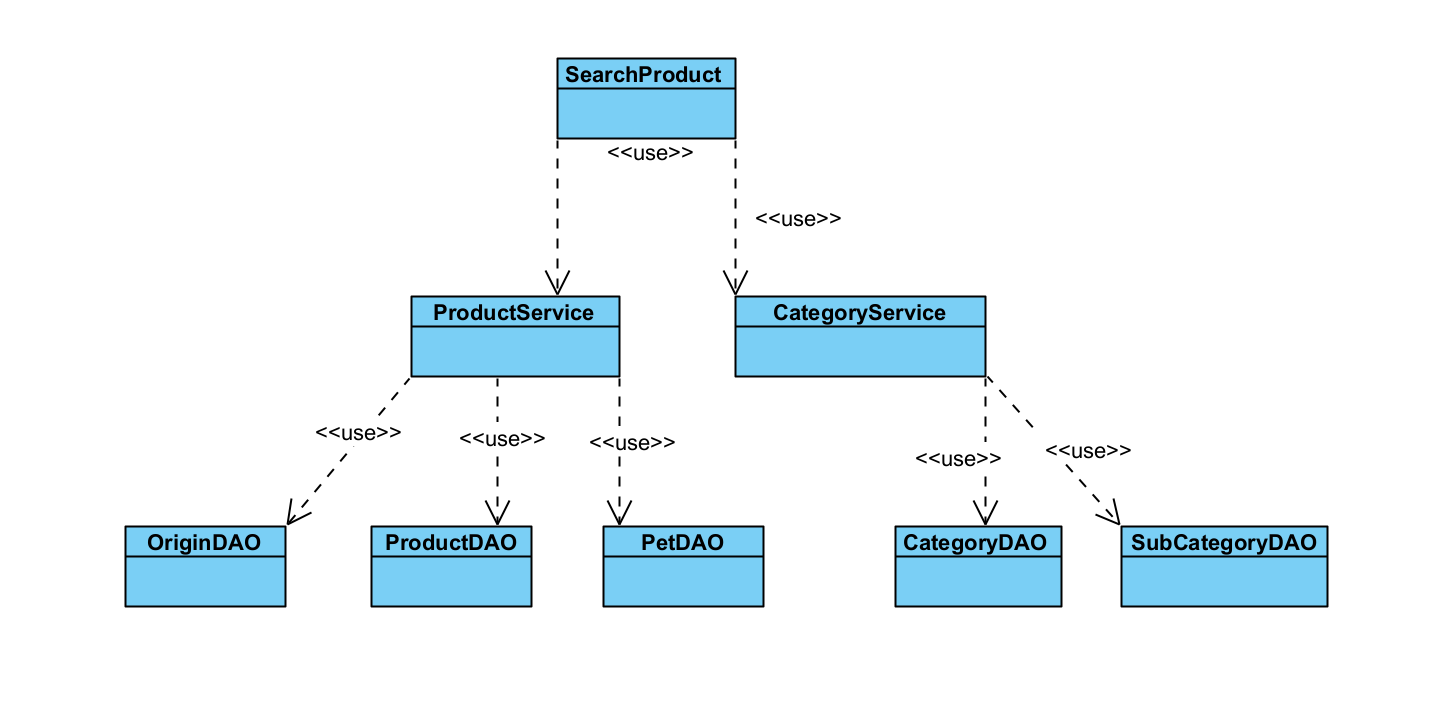
### d. Database Queries

* *SELECT \* FROM product WHERE id = ?*
* *SELECT \* FROM `shop\_pet`.`ProductImage` WHERE product\_id= ?*
* *SELECT \* FROM shop\_pet.Pet WHERE id = ?*
* *SELECT \* FROM shop\_pet.Origin WHERE id = ?*
* *select \* from subcategory where id = ?*
* *select \* from category where id = ?*

## 

## 10. Search

### a. Class Diagram



### b. Class Specifications

#### XYZ Class

| **No** | **Method** | **Description** |
| --- | --- | --- |
| *01* | *<method name>* | *<Description of the method, including the inputs, outputs & internal method processing>* |
| 02 | List<Product> searchProductAccessory(int categoryid, String search) | The search query is built dynamically based on the passed parameters. |

#### ABC Class

| **No** | **Method** | **Description** |
| --- | --- | --- |
| *01* | *<method name>* | *<Description of the method, including the inputs, outputs & internal method processing>* |
|  |  |  |

### c. Sequence Diagram(s)

### d. Database Queries

* *WITH x AS (SELECT @row := @row + 1 AS r, h.\* \n"*

*+ "FROM shop\_pet.product h, (SELECT @row := 0) AS init \n"*

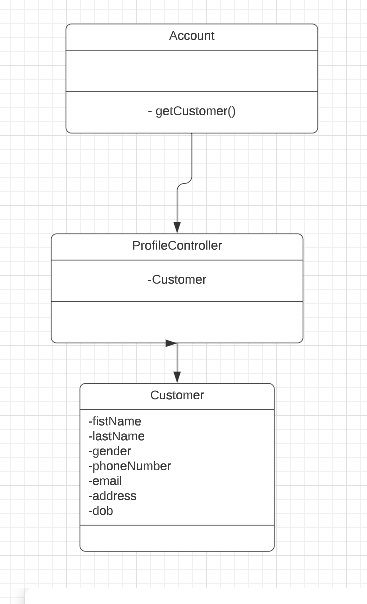
*+ " WHERE name like ?) \n"*

*+ " SELECT \* FROM x WHERE r BETWEEN (? - 1) \* 6 + 1 AND ? \* 6*

* *SELECT \* FROM `shop\_pet`.`ProductImage` WHERE product\_id= ?*
* *SELECT \* FROM shop\_pet.Pet WHERE id = ?*
* *SELECT \* FROM shop\_pet.Origin WHERE id = ?*
* *SELECT \* FROM shop\_pet.product spp2 inner join shop\_pet.Pet spp on spp.id=spp2.pet\_id\n"*
* *+ " inner join shop\_pet.origin spo on spp2.origin\_id=spo.id\n"*
* *+ " inner join shop\_pet.subcategory sps on sps.id=spp2.subcategory\_id\n"*
* *+ " inner join shop\_pet.category spc on sps.category\_id=spc.id \n"*
* *+ " where 1 = 1*
* *SELECT \* FROM shop\_pet.product spp2 inner join shop\_pet.Pet spp on spp.id=spp2.pet\_id\n"*
* *+ " inner join shop\_pet.origin spo on spp2.origin\_id=spo.id\n"*
* *+ " inner join shop\_pet.subcategory sps on sps.id=spp2.subcategory\_id\n"*
* *+ " inner join shop\_pet.category spc on sps.category\_id=spc.id \n"*
* *+ " where 1 = 1*
* *SELECT \* FROM shop\_pet.product spp2 \n"*
* *+ " inner join shop\_pet.origin spo on spp2.origin\_id=spo.id \n"*
* *+ " inner join shop\_pet.subcategory sps on sps.id=spp2.subcategory\_id\n"*
* *+ " inner join shop\_pet.category spc on sps.category\_id=spc.id \n"*
* *+ " where spc.id = ?\n"*
* *+ " limit ? offset*
* *SELECT \* FROM shop\_pet.product spp2 \n"*
* *+ " inner join shop\_pet.origin spo on spp2.origin\_id=spo.id \n"*
* *+ " inner join shop\_pet.subcategory sps on sps.id=spp2.subcategory\_id\n"*
* *+ " inner join shop\_pet.category spc on sps.category\_id=spc.id \n"*
* *+ " where 1=1*
* *SELECT \* FROM shop\_pet.product spp2 \n"*
* *+ " inner join shop\_pet.origin spo on spp2.origin\_id=spo.id \n"*
* *+ " inner join shop\_pet.subcategory sps on sps.id=spp2.subcategory\_id\n"*
* *+ " inner join shop\_pet.category spc on sps.category\_id=spc.id \n"*
* *+ " where 1=1"*
* *SELECT \* FROM shop\_pet.product spp2 inner join shop\_pet.Pet spp on spp.id=spp2.pet\_id\n"*
* *+ " inner join shop\_pet.origin spo on spp2.origin\_id=spo.id\n"*
* *+ " inner join shop\_pet.subcategory sps on sps.id=spp2.subcategory\_id\n"*
* *+ " inner join shop\_pet.category spc on sps.category\_id=spc.id \n"*
* *+ " where 1 = 1*
* *SELECT \* FROM shop\_pet.product spp2 inner join shop\_pet.Pet spp on spp.id=spp2.pet\_id\n"*
* *+ " inner join shop\_pet.origin spo on spp2.origin\_id=spo.id\n"*
* *+ " inner join shop\_pet.subcategory sps on sps.id=spp2.subcategory\_id \n"*
* *+ " inner join shop\_pet.category spc on sps.category\_id=spc.id \n"*
* *+ " where sps.id=*
* *SELECT \* FROM shop\_pet.product spp2\n"*
* *+ " inner join shop\_pet.origin spo on spp2.origin\_id=spo.id\n"*
* *+ " inner join shop\_pet.subcategory sps on sps.id=spp2.subcategory\_id \n"*
* *+ " inner join shop\_pet.category spc on sps.category\_id=spc.id \n"*
* *+ " where sps.id= ?*
* *SELECT \* FROM shop\_pet.product spp2 inner join shop\_pet.Pet spp on spp.id=spp2.pet\_id\n"*
* *+ " inner join shop\_pet.origin spo on spp2.origin\_id=spo.id\n"*
* *+ " inner join shop\_pet.subcategory sps on sps.id=spp2.subcategory\_id\n"*
* *+ " inner join shop\_pet.category spc on sps.category\_id=spc.id \n"*
* *+ " where 1 = 1*
* *SELECT \* FROM shop\_pet.product spp2 \n"*
* *+ " inner join shop\_pet.origin spo on spp2.origin\_id=spo.id\n"*
* *+ " inner join shop\_pet.subcategory sps on sps.id=spp2.subcategory\_id\n"*
* *+ " inner join shop\_pet.category spc on sps.category\_id=spc.id \n"*
* *+ " where 1=1*
* *SELECT \* FROM shop\_pet.product spp2 \n"*
* *+ " inner join shop\_pet.origin spo on spp2.origin\_id=spo.id \n"*
* *+ " inner join shop\_pet.subcategory sps on sps.id=spp2.subcategory\_id\n"*
* *+ " inner join shop\_pet.category spc on sps.category\_id=spc.id \n"*
* *+ " where 1=1*

## 11 . View Profile feature

### a. Class Diagram



### b. Class Specifications

#### ProfileController

| **No** | **Method** | **Description** |
| --- | --- | --- |
| *01* | *<method name>* | *<Description of the method, including the inputs, outputs & internal method processing>* |
|  |  |  |

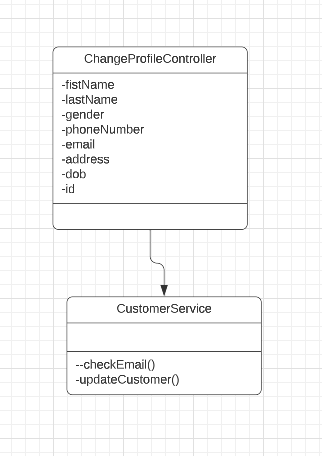
#### 

### c. Sequence Diagram(s)

### d. Database Queries

## 12. Update Profile

### a. Class Diagram



### b. Class Specifications

#### ChangeProfileController

| **No** | **Method** | **Description** |
| --- | --- | --- |
| *01* | *<method name>* | *<Description of the method, including the inputs, outputs & internal method processing>* |
|  |  |  |

#### 

#### AccountDao

| **No** | **Method** | **Description** |
| --- | --- | --- |
| *01* | *<method name>* | *<Description of the method, including the inputs, outputs & internal method processing>* |
|  |  |  |

#### 

#### Customer

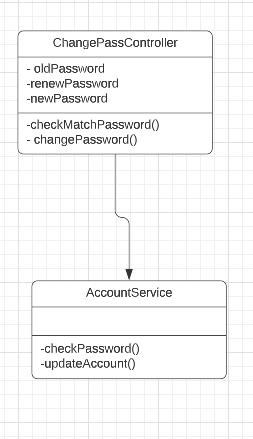
| **No** | **Method** | **Description** |
| --- | --- | --- |
| *01* | *<method name>* | *<Description of the method, including the inputs, outputs & internal method processing>* |
|  |  |  |

### c. Sequence Diagram(s)

### d. Database Queries

## 13. Change Password

### a. Class Diagram



### b. Class Specifications

#### ChangePasswordController

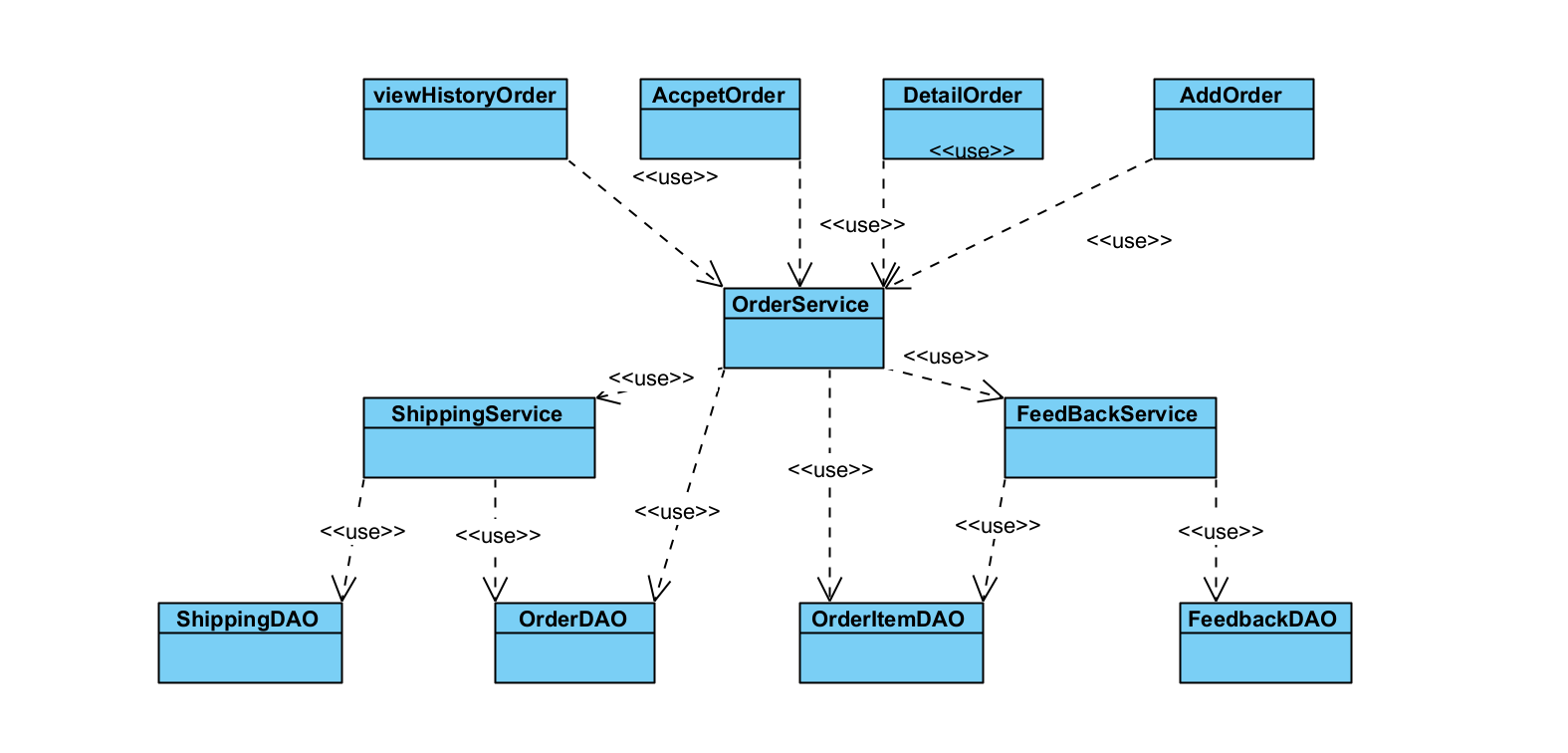
| **No** | **Method** | **Description** |
| --- | --- | --- |
| *01* | *<method name>* | *<Description of the method, including the inputs, outputs & internal method processing>* |
|  |  |  |

### c. Sequence Diagram(s)

### d. Database Queries

## 14. Managed Order

### a. Class Diagram



### b. Class Specifications

#### Order Class

| **No** | **Method** | **Description** |
| --- | --- | --- |
| *01* | *<method name>* | *<Description of the method, including the inputs, outputs & internal method processing>* |
|  |  |  |

#### ABC Class

| **No** | **Method** | **Description** |
| --- | --- | --- |
| *01* | *<method name>* | *<Description of the method, including the inputs, outputs & internal method processing>* |
|  |  |  |

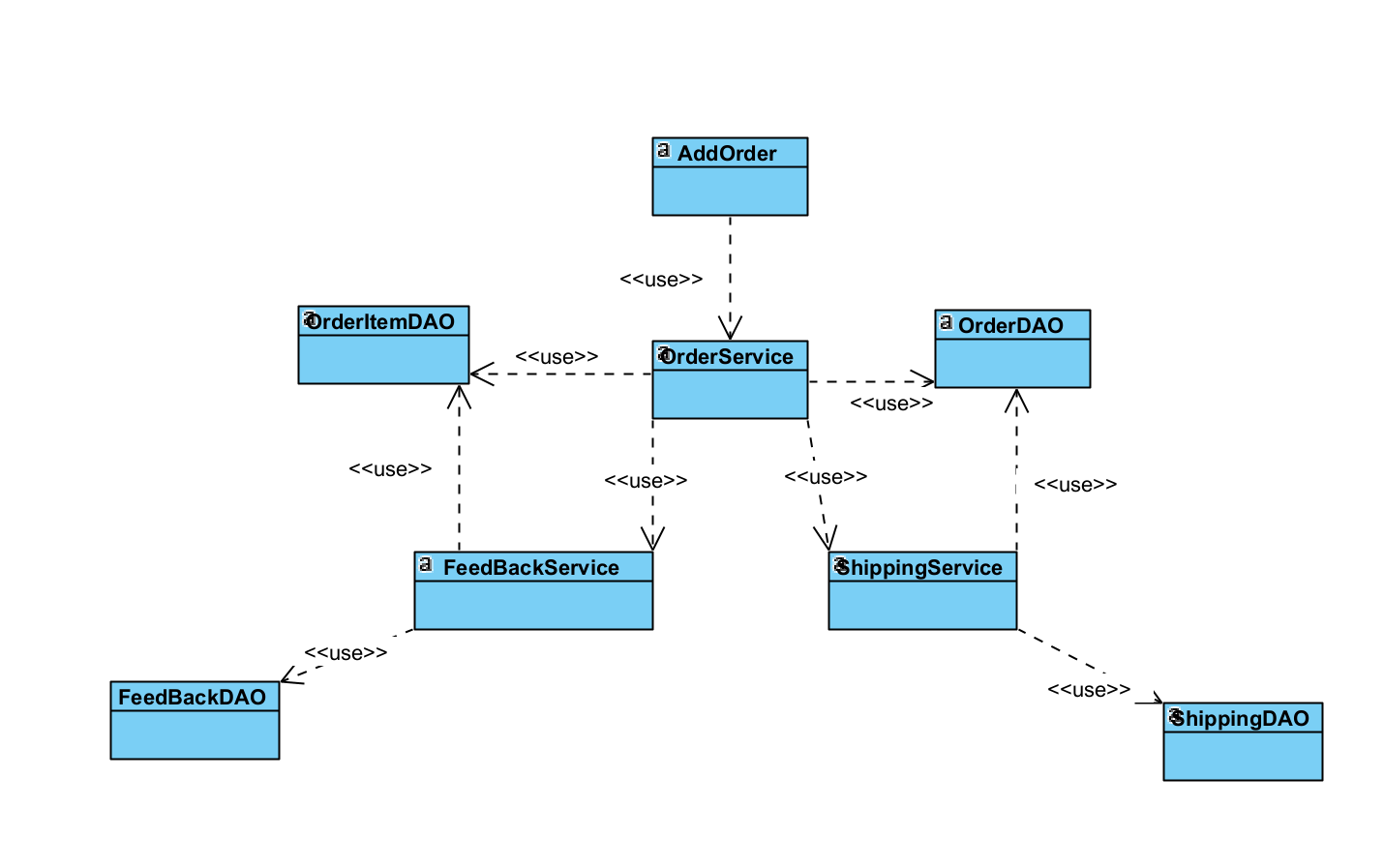
### c. Sequence Diagram(s)



### d. Database Queries

## 15. Create Order

### a. Class Diagram



### b. Class Specifications

#### Order Class

| **No** | **Method** | **Description** |
| --- | --- | --- |
| *01* | *<method name>* | *<Description of the method, including the inputs, outputs & internal method processing>* |
|  |  |  |

#### ABC Class

| **No** | **Method** | **Description** |
| --- | --- | --- |
| *01* | *<method name>* | *<Description of the method, including the inputs, outputs & internal method processing>* |
|  |  |  |

### c. Sequence Diagram(s)



### d. Database Queries

## 16. View Order

### a. Class Diagram

### b. Class Specifications

#### Order Class

| **No** | **Method** | **Description** |
| --- | --- | --- |
| *01* | *<method name>* | *<Description of the method, including the inputs, outputs & internal method processing>* |
|  |  |  |

#### ABC Class

| **No** | **Method** | **Description** |
| --- | --- | --- |
| *01* | *<method name>* | *<Description of the method, including the inputs, outputs & internal method processing>* |
|  |  |  |

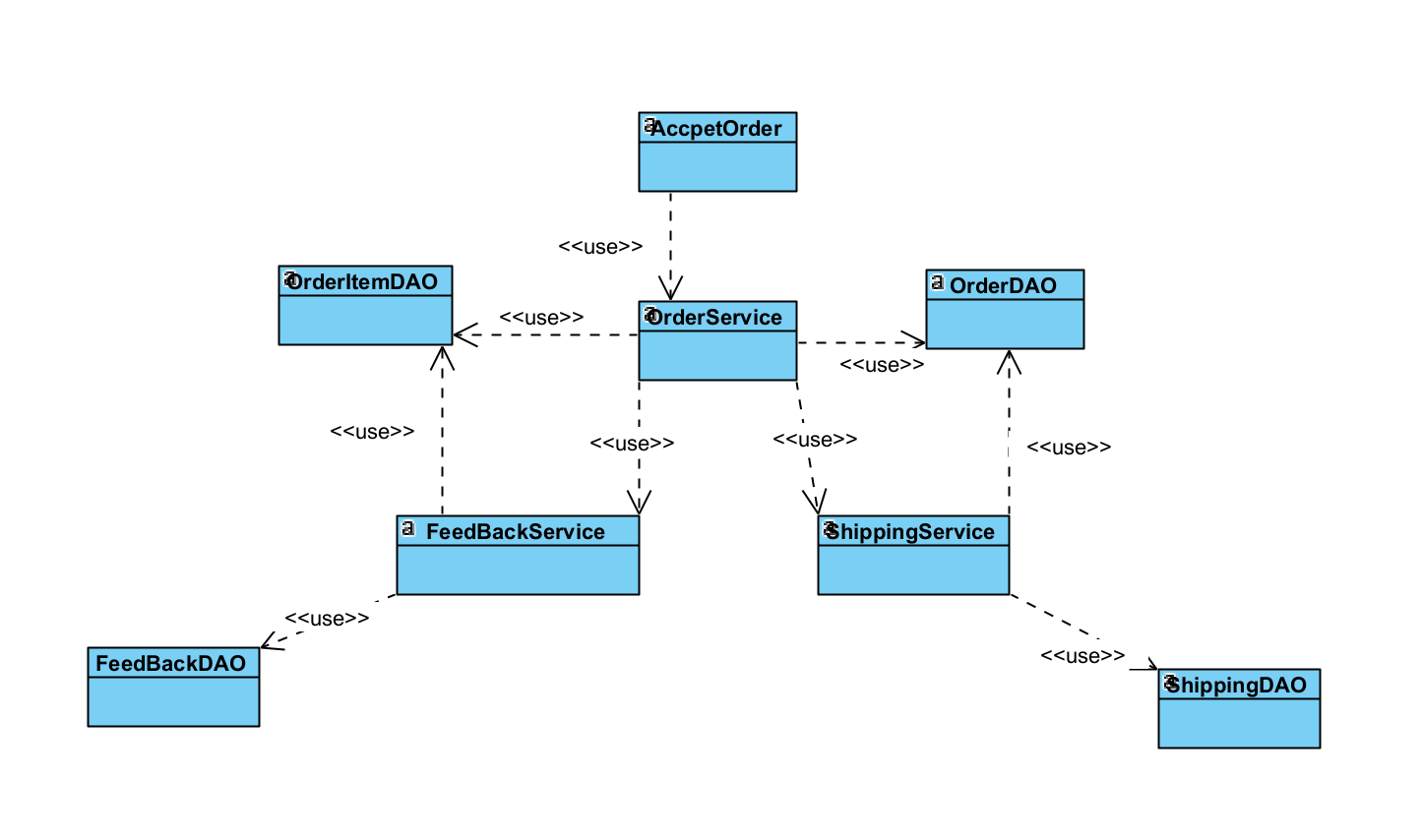
### c. Sequence Diagram(s)

### d. Database Queries

### 

## 17. Accpet\_Deny Order

### a. Class Diagram



### b. Class Specifications

#### Order Class

| **No** | **Method** | **Description** |
| --- | --- | --- |
| *01* | *<method name>* | *<Description of the method, including the inputs, outputs & internal method processing>* |
|  |  |  |

#### ABC Class

| **No** | **Method** | **Description** |
| --- | --- | --- |
| *01* | *<method name>* | *<Description of the method, including the inputs, outputs & internal method processing>* |
|  |  |  |

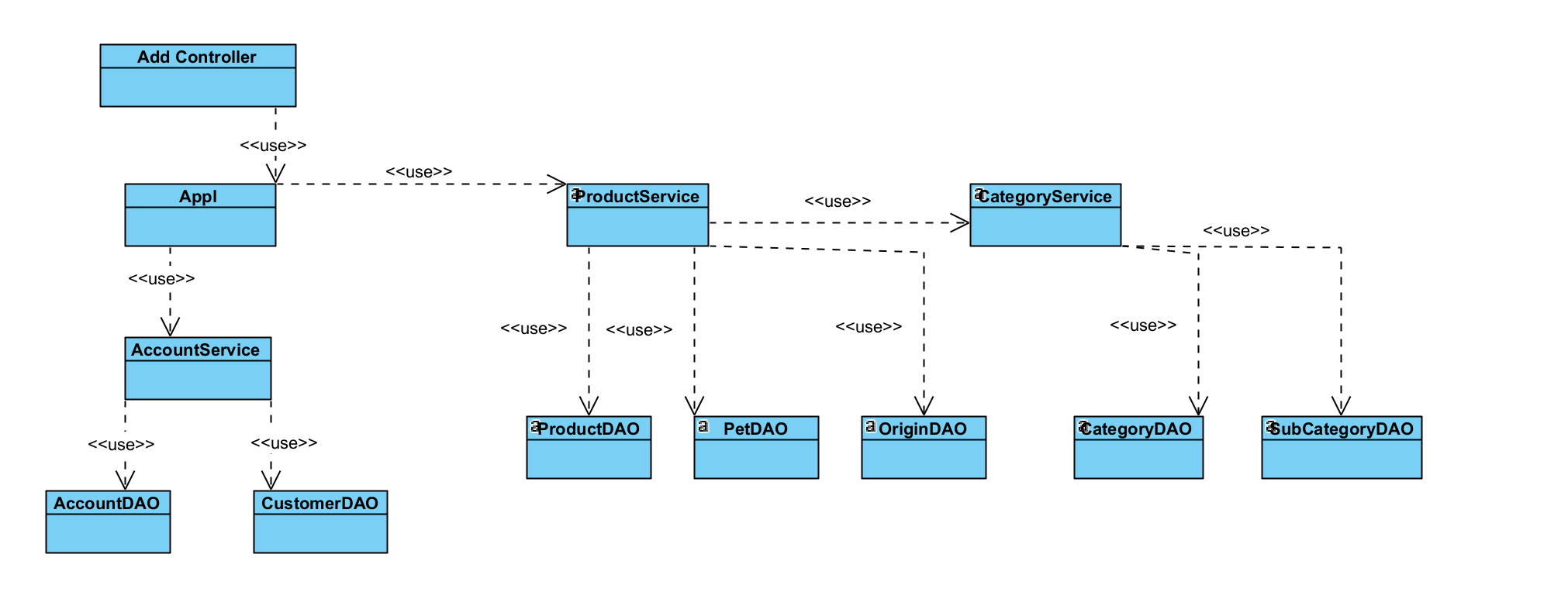
### c. Sequence Diagram(s)



### d. Database Queries

## 18. Add to Cart

### a. Class Diagram



### b. Class Specifications

#### Order Class

| **No** | **Method** | **Description** |
| --- | --- | --- |
| *01* | *<method name>* | *<Description of the method, including the inputs, outputs & internal method processing>* |
|  |  |  |

#### ABC Class

| **No** | **Method** | **Description** |
| --- | --- | --- |
| *01* | *<method name>* | *<Description of the method, including the inputs, outputs & internal method processing>* |
|  |  |  |

### c. Sequence Diagram(s)



### d. Database Queries

## 19. View Cart

### a. Class Diagram

### 

### b. Class Specifications

#### AddCartController

| **No** | **Method** | **Description** |
| --- | --- | --- |
| 01 | doGet(HttpServletRequest request, HttpServletResponse response) : void | Connect to web |

#### Cart Service

| **No** | **Method** | **Description** |
| --- | --- | --- |
| 01 | getCart(accountId : int) : Cart | get Cart with CartItem by Account Id |

#### 

#### Product Service

***Class Methods***

| **No** | **Method** | **Description** |
| --- | --- | --- |
| 01 | getProductsById(productsId : int[]) : Product [] | get Product to map with CartItem |

#### CartDAO

| **No** | **Method** | **Description** |
| --- | --- | --- |
| 01 | getProductsById(productsId : int[]) : Product [] | get Product to map with CartItem |

#### CartItemDAO

| **No** | **Method** | **Description** |
| --- | --- | --- |
| 01 | getProductsById(productsId : int[]) : Product [] | get Product to map with CartItem |

#### ProductDAO

| **No** | **Method** | **Description** |
| --- | --- | --- |
| 01 | getProductsById(productsId : int[]) : Product [] | get Product to map with CartItem |

#### CartMapping

| **No** | **Method** | **Description** |
| --- | --- | --- |
| *01* | getProductsById(productsId : int[]) : Product [] | get Product to map with CartItem |

#### CartItemMapping

| **No** | **Method** | **Description** |
| --- | --- | --- |
| 01 | getProductsById(productsId : int[]) : Product [] | get Product to map with CartItem |

#### ProductMapping

| **No** | **Method** | **Description** |
| --- | --- | --- |
| 01 | getProductsById(productsId : int[]) : Product [] | get Product to map with CartItem |

#### Cart

| **No** | **Method** | **Description** |
| --- | --- | --- |
| 01 | getProductsById(productsId : int[]) : Product [] | get Product to map with CartItem |

### c. Sequence Diagram(s)

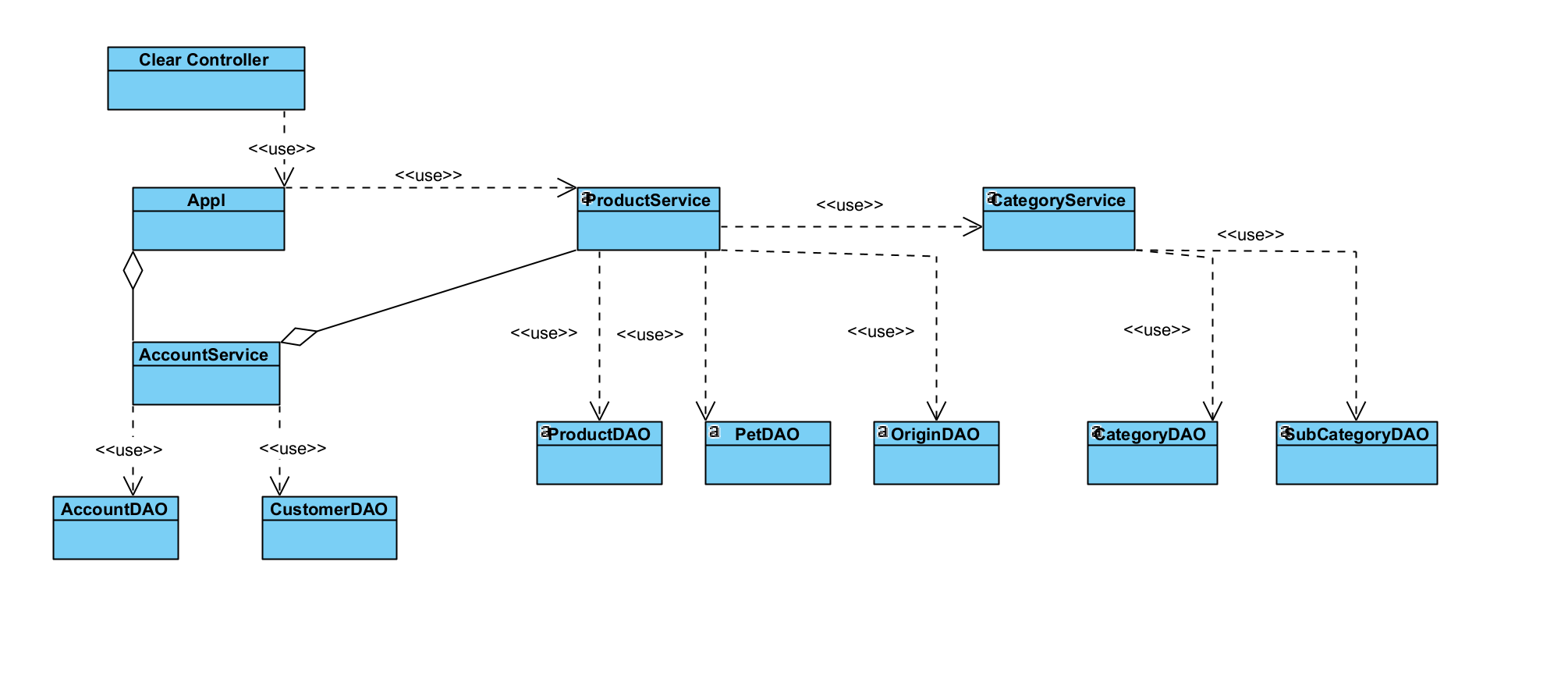


### d. Database Queries

## 

## 20. Clear cart

### a. Class Diagram



### b. Class Specifications

#### XYZ Class

| **No** | **Method** | **Description** |
| --- | --- | --- |
| *01* | *<method name>* | *<Description of the method, including the inputs, outputs & internal method processing>* |
|  |  |  |

#### ABC Class

| **No** | **Method** | **Description** |
| --- | --- | --- |
| *01* | *<method name>* | *<Description of the method, including the inputs, outputs & internal method processing>* |
|  |  |  |

### c. Sequence Diagram(s)

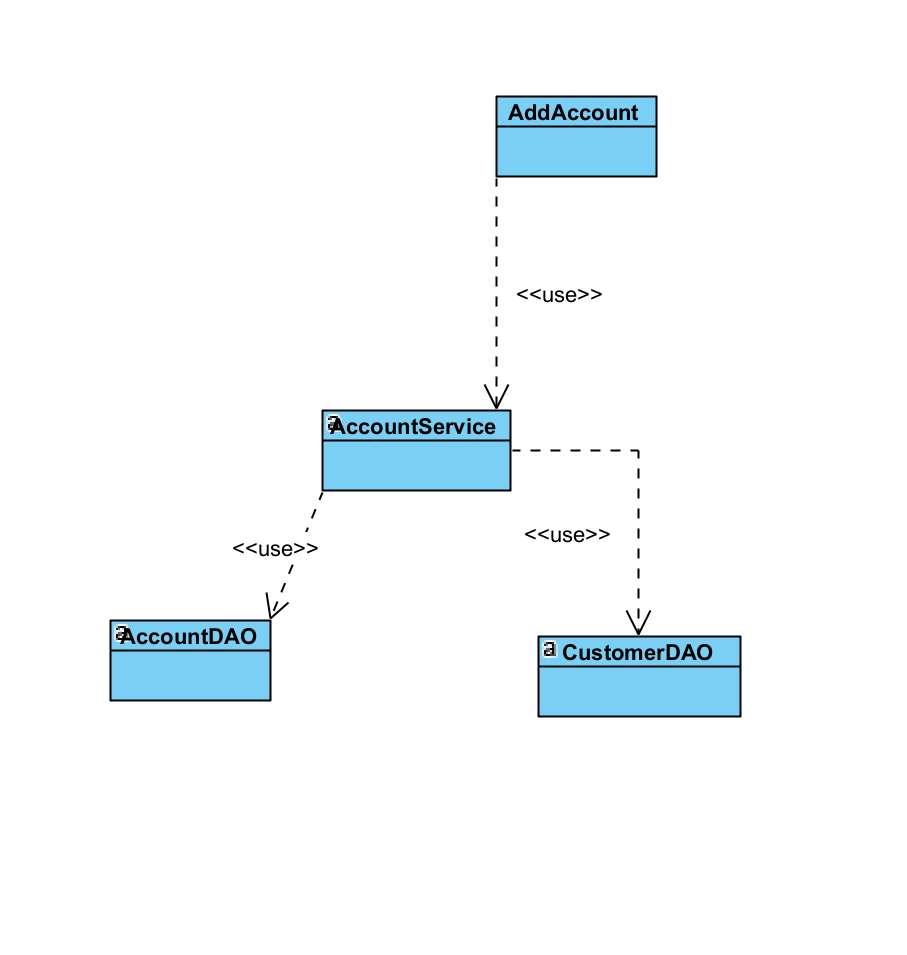
### d. Database Queries

## 

### 

## 22. Add account

### a. Class Diagram



### b. Class Specifications

#### XYZ Class

| **No** | **Method** | **Description** |
| --- | --- | --- |
| *01* | *<method name>* | *<Description of the method, including the inputs, outputs & internal method processing>* |
|  |  |  |

#### ABC Class

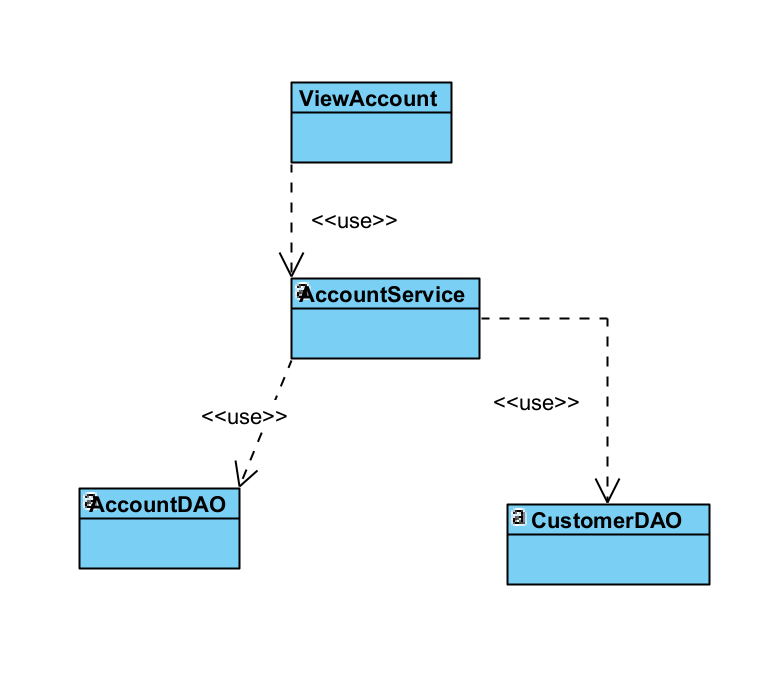
| **No** | **Method** | **Description** |
| --- | --- | --- |
| *01* | *<method name>* | *<Description of the method, including the inputs, outputs & internal method processing>* |
|  |  |  |

### c. Sequence Diagram(s)

### d. Database Queries

## 23. View account

### a. Class Diagram



### b. Class Specifications

#### XYZ Class

| **No** | **Method** | **Description** |
| --- | --- | --- |
| *01* | *<method name>* | *<Description of the method, including the inputs, outputs & internal method processing>* |
|  |  |  |

#### ABC Class

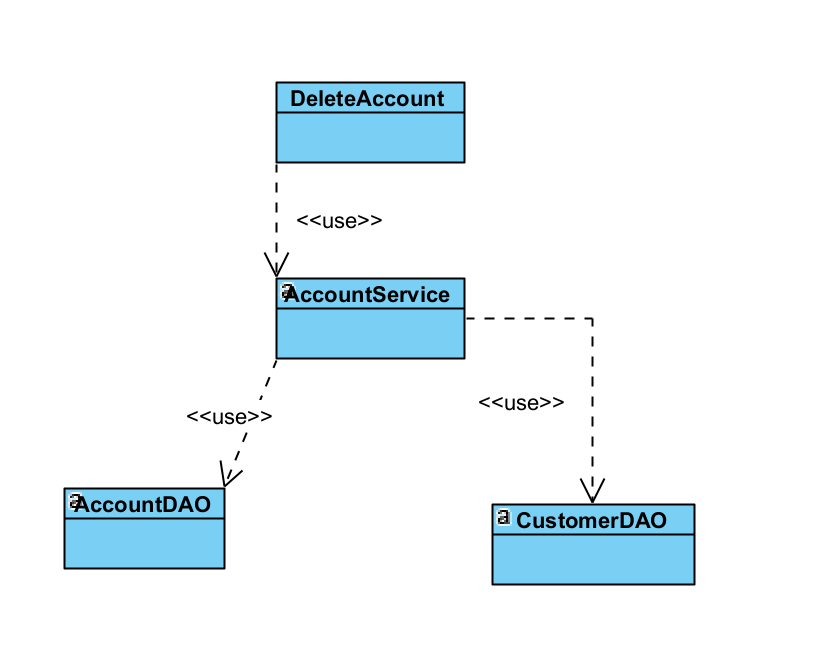
| **No** | **Method** | **Description** |
| --- | --- | --- |
| *01* | *<method name>* | *<Description of the method, including the inputs, outputs & internal method processing>* |
|  |  |  |

### c. Sequence Diagram(s)

### d. Database Queries

## 24. Delete account

### a. Class Diagram



### b. Class Specifications

#### XYZ Class

| **No** | **Method** | **Description** |
| --- | --- | --- |
| *01* | *<method name>* | *<Description of the method, including the inputs, outputs & internal method processing>* |
|  |  |  |

#### ABC Class

| **No** | **Method** | **Description** |
| --- | --- | --- |
| *01* | *<method name>* | *<Description of the method, including the inputs, outputs & internal method processing>* |
|  |  |  |

### c. Sequence Diagram(s)

### d. Database Queries

### 