



BOOK LIBRARY

Software Design Specification

– Hanoi, March 20th 2023 –

Record of changes

Date	A* M, D	In charge	Change Description
26/2/2023	A		Code Packages
26/2/2023	A		1.1 View Profile
26/2/2023	A		1.2 Update Profile
27/2/2023	A		1.3 View MyOrders
27/2/2023	A		1.4. Change Order Status
1/3/2023	A		1.5. OrderAction
1/3/2023	A		1.6. AddToCart
1/3/2023	A		2.1 Dashboard
1/3/2023	A		2.2. Add New Product
1/3/2023	A		2.3. Edit Product
1/3/2023	A		2.4. Delete Product
3/3/2023	A		3.1. View Order
3/3/2023	A		3.2. Change order status
3/3/2023	A		3.3. Filter Order By Date
3/3/2023	A		3.4. View Employee
3/3/2023	A		3.5. Create Employees
4/3/2023	A		4.1 Homepage

*A - Added M - Modified D - Deleted

Table of contents

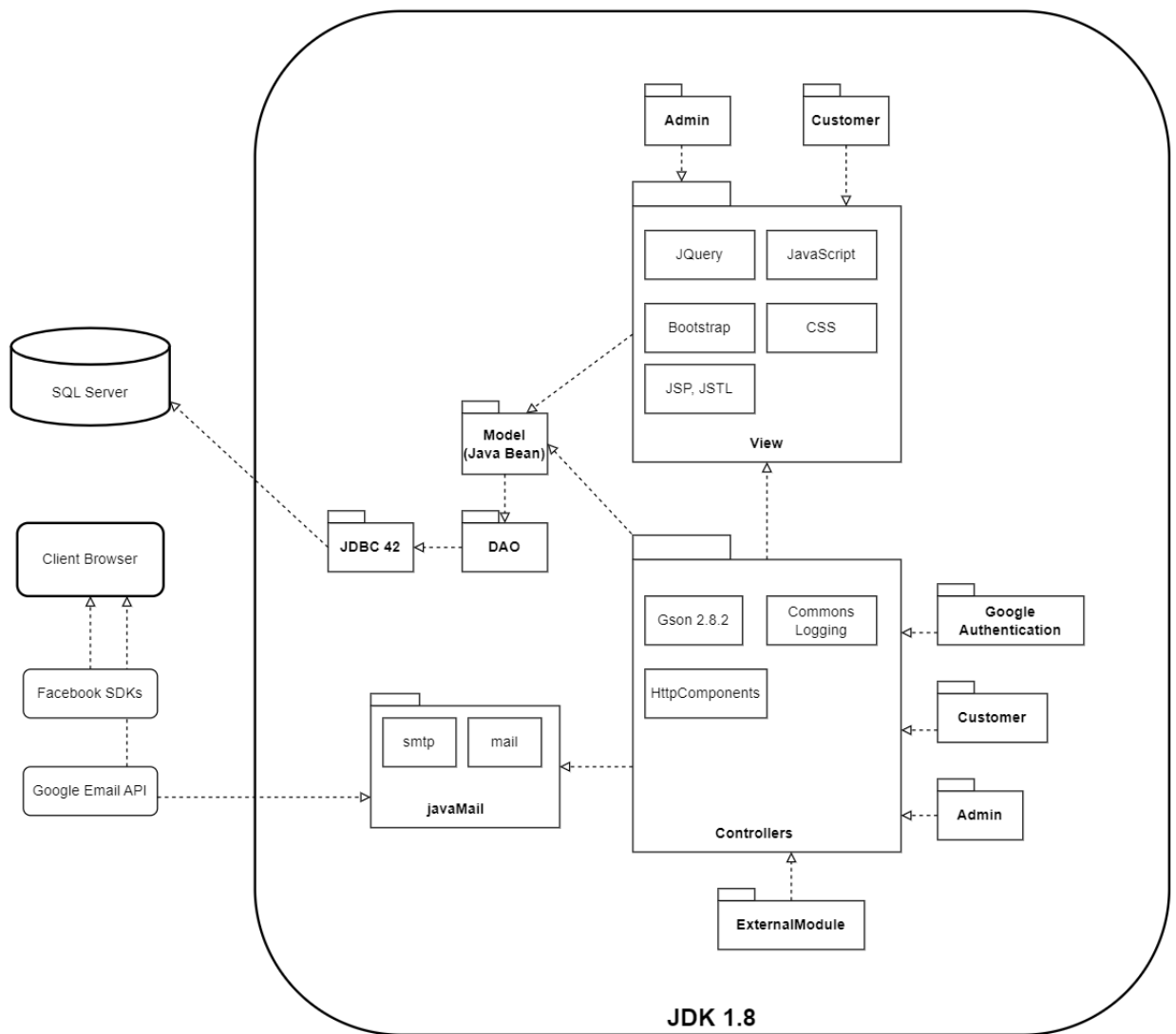
Record of changes	2
I. Overview	6
1. Code Packages	6
Package descriptions	6
2. Database Design	8
a. Database Schema	8
b. Table Description	8
II. Code Designs	9
1. Customer/Guest feature	9
1.1 View Profile	9
a. Class Diagram	9
b. Class Specifications	10
c. Sequence Diagram(s)	11
1.2 Update Profile	12
a. Class Diagram	12
b. Class Specifications	12
c. Sequence Diagram(s)	14
1.3 View MyOrders	15
a. Class Diagram	15
b. Class Specifications	15
c. Sequence Diagram(s)	16
1.4. Change Order Status	17
a. Class Diagram	17
b. Class Specifications	17
c. Sequence Diagram(s)	18
1.5. OrderAction	19
a. Class Diagram	19
b. Class Specifications	19
c. Sequence Diagram(s)	22
d. Database Queries	22
1.6. AddToCart	24
a. Class Diagram	24
b. Class Specifications	24
c. Sequence Diagram(s)	26
1.7. Change cart	26
a. Class Diagram	27
b. Class Specifications	28

c. Sequence Diagram(s)	28
1.8. Checkout	29
a. Class Diagram	29
b. Class Specifications	30
c. Sequence Diagram(s)	30
1.9. Review	31
a. Class Diagram	31
b. Class Specifications	32
c. Sequence Diagram(s)	32
1.10 Product Catalog	33
a. Class Diagram	33
b. Class Specifications	33
c. Sequence Diagram(s)	33
2. Admin features	34
2.1 Dashboard	34
a. Class Diagram	34
b. Class Specifications	34
c. Sequence Diagram(s)	36
2.2. Add New Product	37
a. Class Diagram	37
b. Class Specifications	37
c. Sequence Diagram(s)	39
2.3. Edit Product	40
a. Class Diagram	40
b. Class Specifications	41
c. Sequence Diagram(s)	42
2.4. Delete product	43
a. Class Diagram	43
b. Class Specifications	44
c. Sequence Diagram(s)	45
3. Employee feature	46
3.1: View Order	46
a. Class Diagram	46
b. Class Specifications	46
c. Sequence Diagram(s)	47
d. Database Queries	48
3.2. Change order status	48
a. Class diagram	48

b. Class Specifications	49
c. Sequence Diagram(s)	50
d. Database Queries	50
3.3. Filter Order By Date	51
a. Class diagram	51
b. Class Specifications	51
c. Sequence Diagram(s)	52
3.4. View Employee	53
a. class diagram	53
b. Class Specifications	53
c. Sequence Diagram(s)	54
3.5. Create Employees	55
a. class diagram	55
b. Class Specifications	55
c. Sequence Diagram(s)	56
4. Common features	57
4.1 Homepage	57
a. Class Diagram	57
b. Class Specifications	57
c. Sequence Diagram(s)	59
d. Database Queries	59
4.2 Login	60
a. Class Diagram	60
b. Class Specifications	60
c. Sequence Diagram(s)	60
d. Database Queries	60
4.3 Logout	61
a. Class Diagram	61
b. Class Specifications	61
c. Sequence Diagram(s)	61
d. Database Queries	61
4.4 Forgot Password	61
a. Class Diagram	61
b. Class Specifications	61
c. Sequence Diagram(s)	61
d. Database Queries	61

I. Overview

1. Code Packages

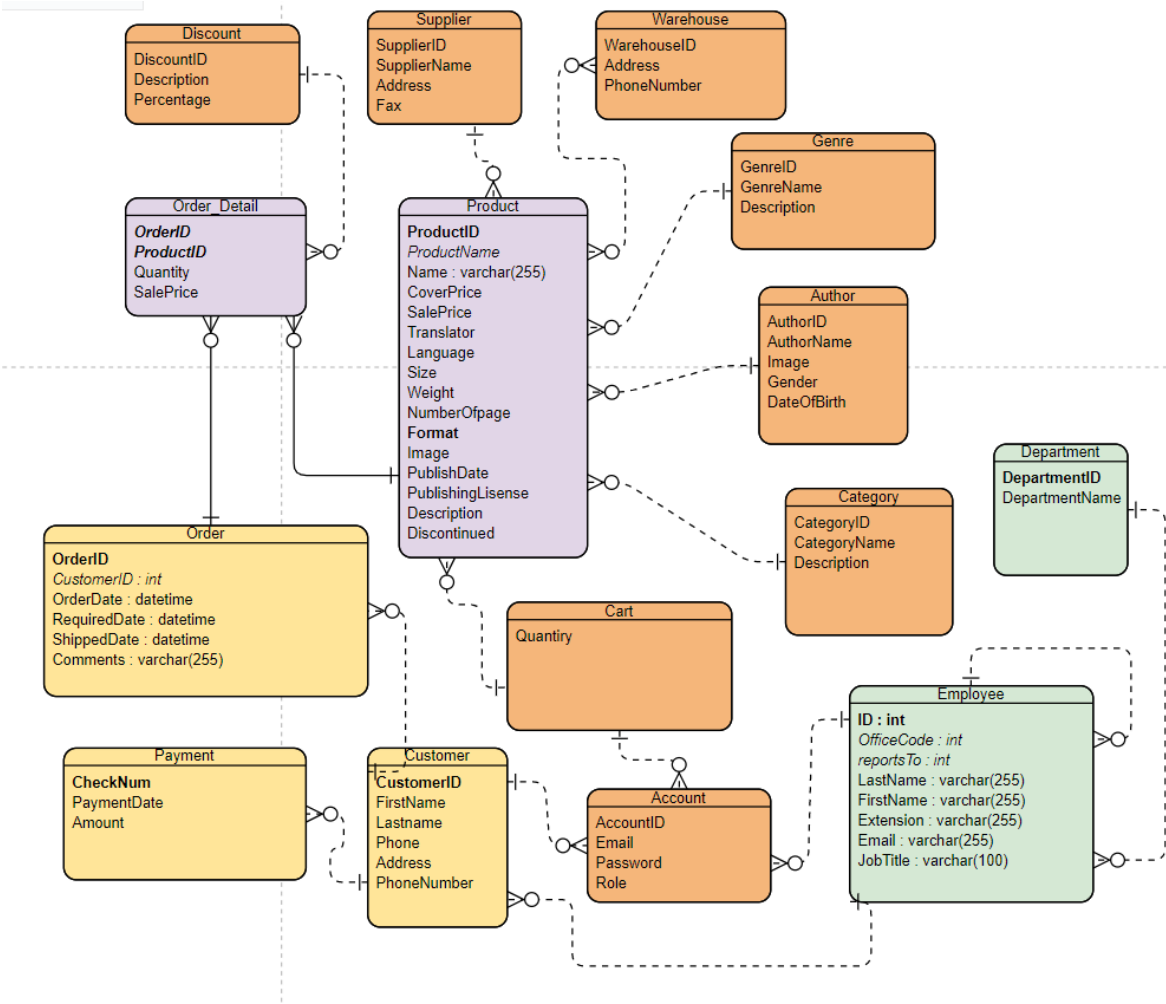


Package descriptions

No	Package	Description
01	model	Package contains all the object classes represent the entities and relationships in Database The classes all have names corresponding to the entity in the Database
02	dao	Package contains all the object classes that access directly to Database The classes contained in this all have the ending suffix [DBContext]
03	controller	Package contains all the business classes that process the logic tasks
04	controller.customer	Package contains all the business classes that process the logic tasks for customer role only
05	controller.admin	Package contains all the business classes that process the logic tasks for admin role only
06	view	Package contains all the assets, resource to display information on client browser
07	view.admin	Package contains resource of admin role to display information on client browser
08	view.customer	Package contains resource of customer role to display information on client browser
09	controller.GoogleAuth entication	Handle google login when user selects google account login feature
10	controller.ExternalModule	Contains other processing modules The classes included in this must contain the ending suffix: [module]

2. Database Design

a. Database Schema



b. Table Description

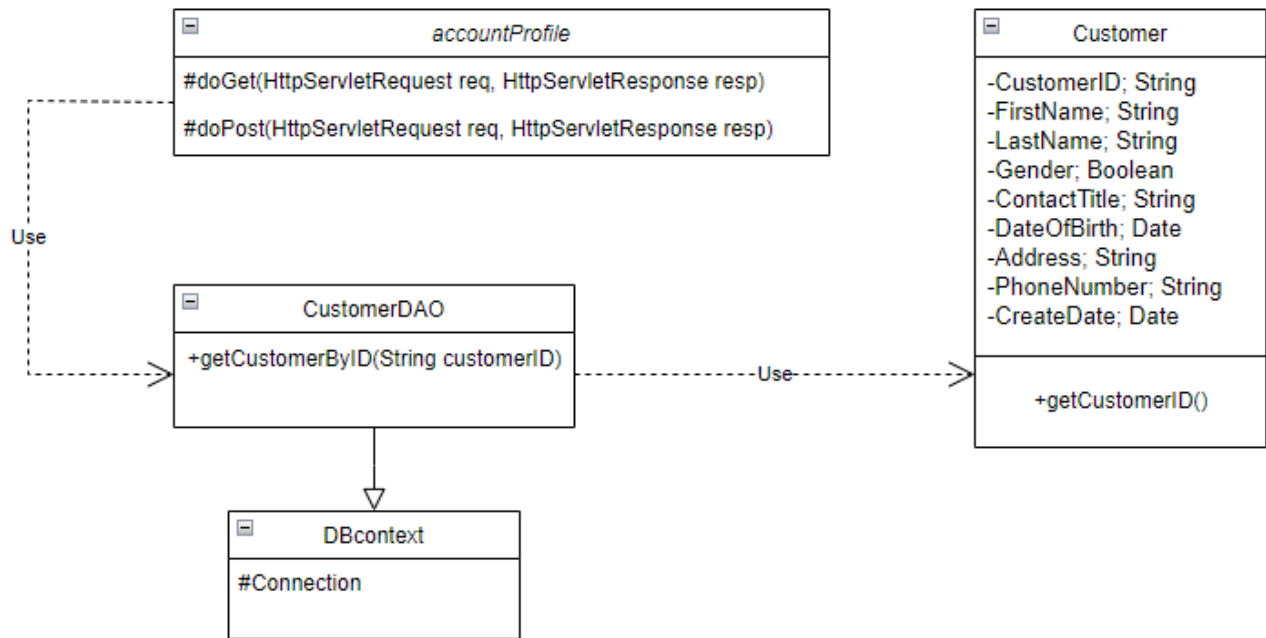
No	Table	Description
1	Product	Main object is book
2	Customer	Consumer, who wants to buy book
3	Employee	Manager, saler performs business operation
4	Account	The registered user of the system
5	Order	Record information about an order
6	Order Detail	Record more detail in an order
7	Category	Classify book by categories
8	Genre	Classify book by Genres
9	Warehouse	Repository to store book, perform send books to customer
10	Supplier	Publisher will print book and supply them into the market
11	Discount	Store information about voucher
12	Cart	Store book which customer want to buy
13	Payment	payment method of customer
14	Department	Classify employee into department, admin/saler...etc

II. Code Designs

1. Customer/Guest feature

1.1 View Profile

a. Class Diagram



b. Class Specifications

accountProfile Class

No	Method	Description
01	doGet	Called by the server (via the service method) to allow a servlet to handle a GET request.

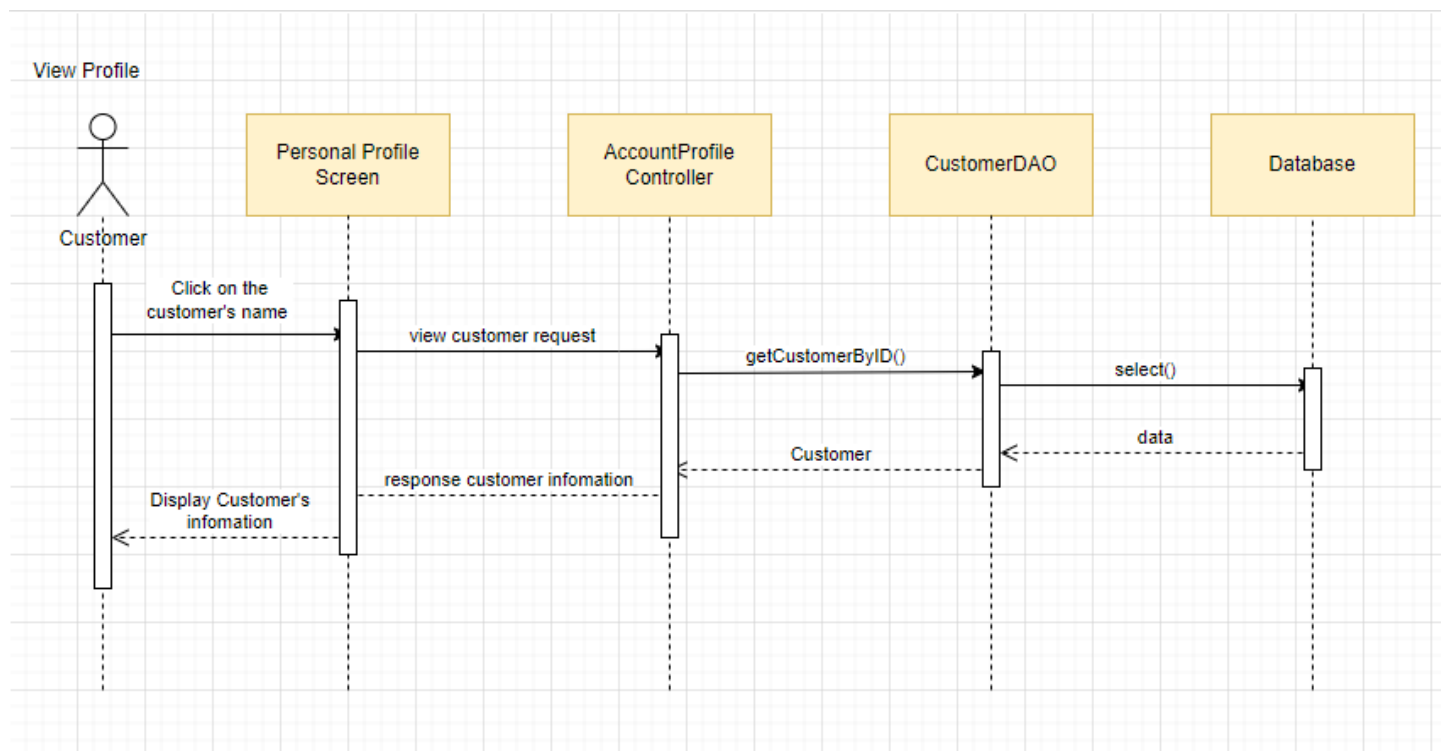
CustomerDAO Class

No	Method	Description
01	+getCustomerByID(a.getCustomerID())	Get customer information by customer's ID - Input: Customer ID - Output: Customer information

Customer Class

No	Method	Description
01	+getCustomerID()	return customer ID of an instance derive from customer class - Input: None - Output: Customer ID

c. Sequence Diagram(s)

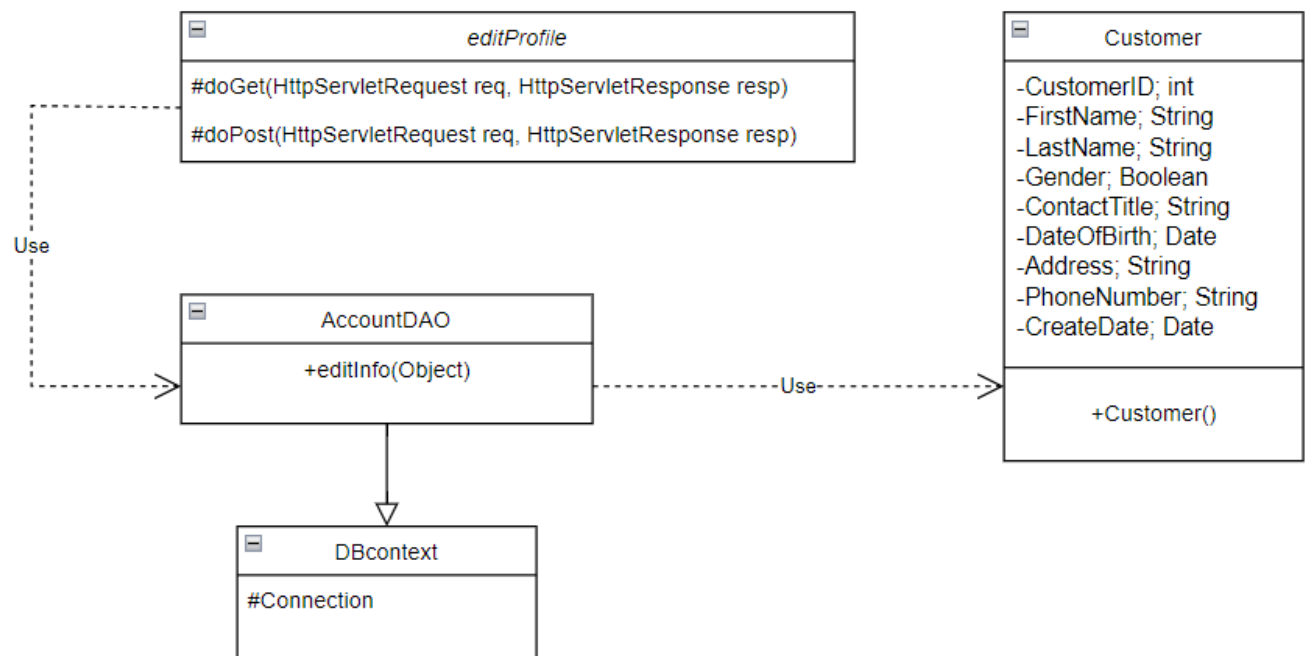


d. Database Queries

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

1.2 Update Profile

a. Class Diagram



b. Class Specifications

EditProfile Class

No	Method	Description
01	doGet	Called by the server (via the service method) to allow a servlet to handle a GET request.
02	doPost	Called by the server (via the service method) to allow a servlet to handle a POST request.

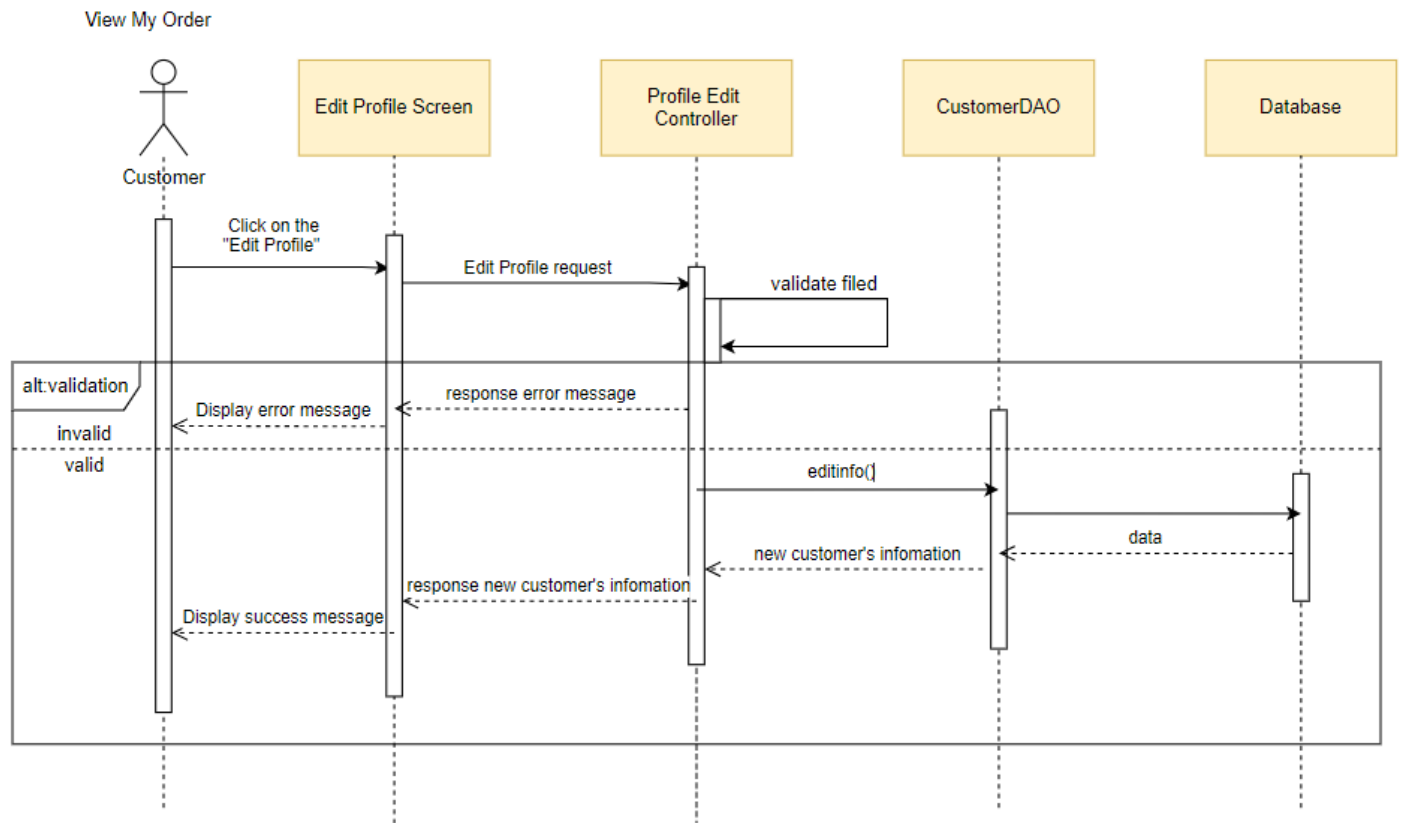
AccountDAO Class

No	Method	Description
01	+editInfo(customer)	Update customer information by new customer class - Input: customer - Output: new customer information

Customer Class

No	Method	Description
01	Customer(int CustomerID, String FirstName, String LastName, boolean Gender, String ContactTitle, Date DateOfBirth, String Address, String PhoneNumber, Date CreateDate)	return customer object of an instance derive from customer class - Input: None - Output: customers object

c. Sequence Diagram(s)



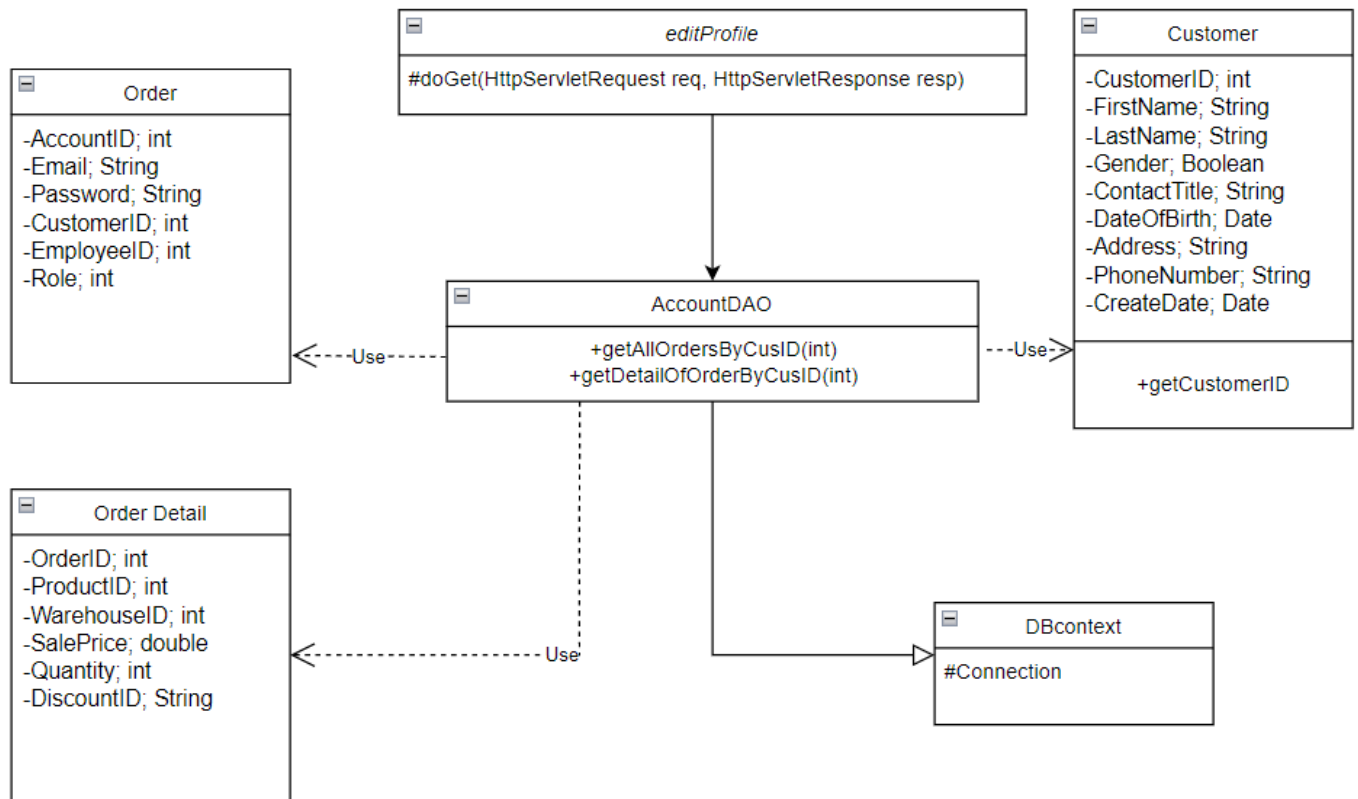
d. Database Queries

AccountDAO Class

No	Method	Query
01	editInfo()	update Customers set FirstName = ? , LastName = ? , Gender = ? , ContactTitle = ? , DateOfBirth = ? , Address = ? , PhoneNumber = ? , CreateDate = ? where CustomerID = ?

1.3 View MyOrders

a. Class Diagram



b. Class Specifications

ViewProfile Class

No	Method	Description
01	doGet	Called by the server (via the service method) to allow a servlet to handle a GET request.

OrderDAO Class

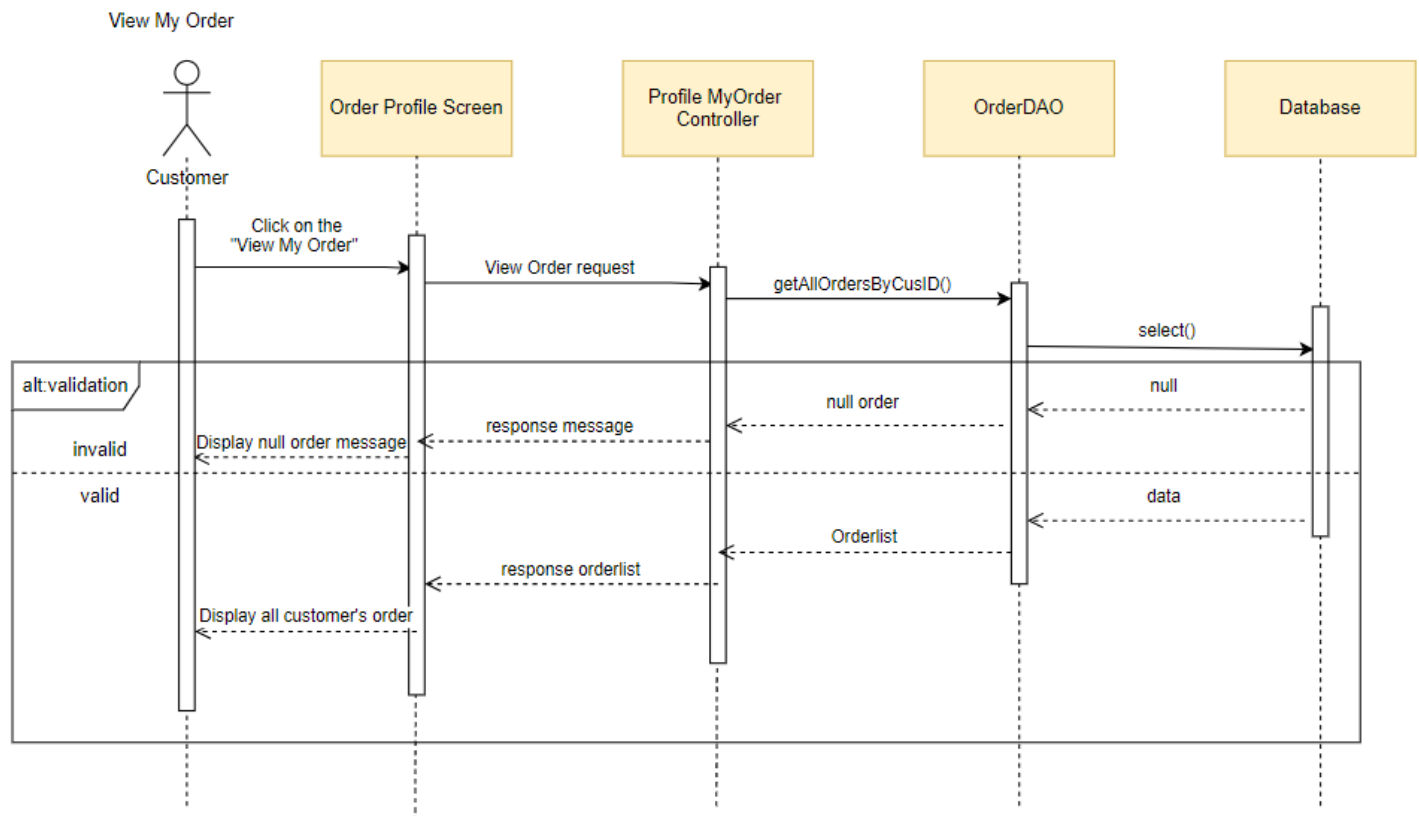
No	Method	Description
01	+getAllOrdersByCusID(getCustomerID())	Update customer information by new customer class

	+getDetailOfOrderByCusID(getCustomerID())	- Input: CustomerID - Output: Customer's order
--	---	---

Customer Class

No	Method	Description
01	getCustomerID()	return customer ID of an instance derive from customer class - Input: None - Output: Customer ID

c. Sequence Diagram(s)

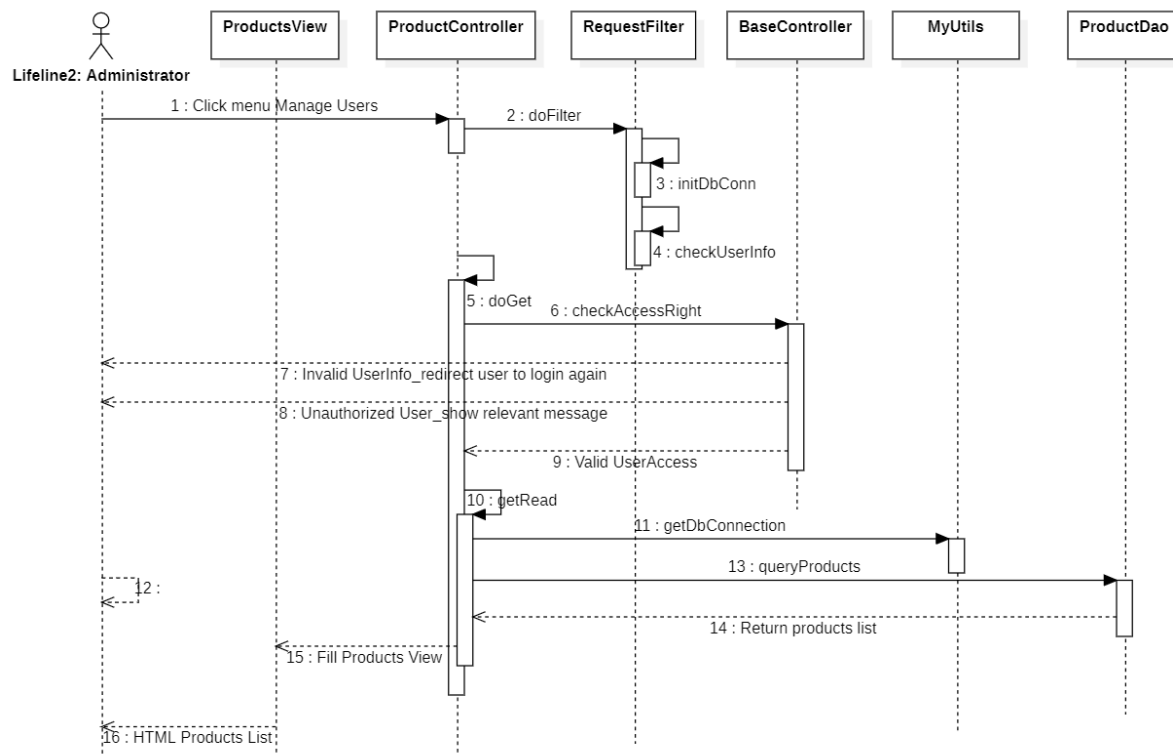


d. Database Queries

OrderDAO Class

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

c. Sequence Diagram(s)



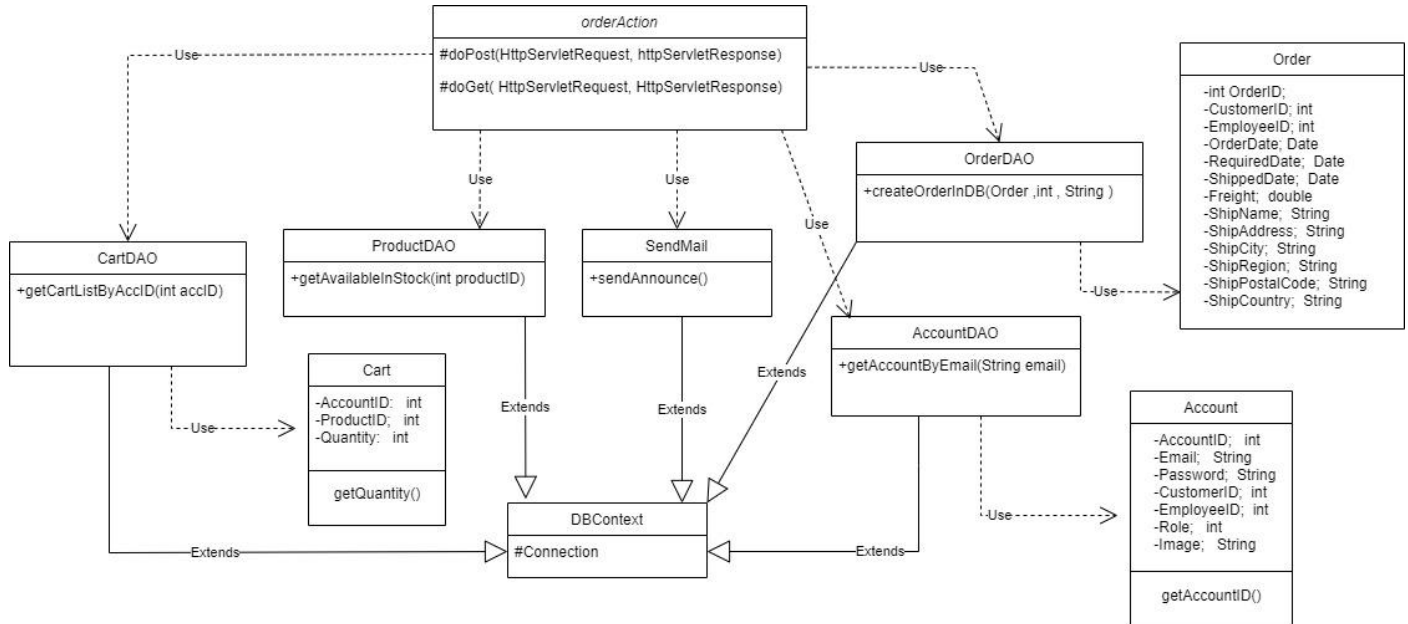
d. Database Queries

OrderDAO Class

No	Method	Query
01	updateOrderStatus(String orderId, String status)	UPDATE Orders SET status = ? WHERE OrderID = ?

1.5. Checkout

a. Class Diagram



b. Class Specifications

OrderAction Class

No	Method	Description
01	doGet	Called by the server (via the service method) to allow a servlet to handle a GET request.
02	doPost	is Called by the server (via the service method) to allow a servlet to handle a POST request.

CartDAO Class

No	Method	Description
----	--------	-------------

01	+getCartListByAccountID(int accountID)	Get cart list belonging to one person with accountID - Input: accountID - Output: list of products in cart
----	--	--

Cart Class

No	Method	Description
01	getQuantity()	Get number of the same item in cart - Input: None - Output: number of the same item

ProductDAO Class

No	Method	Description
01	+getAvailableInStock(int productID)	Get the list of available inventories of a product - Input: productID - Output: number of available product

SendMail Class

No	Method	Description
01	+sendAnnounce()	send an email to customer when they buy book successfully - Input: Order information - Output: None

AccountDAO Class

No	Method	Description
01	+getAccountByEmail(String email)	Get account information by email address

		- Input: email address - Output: account information
--	--	---

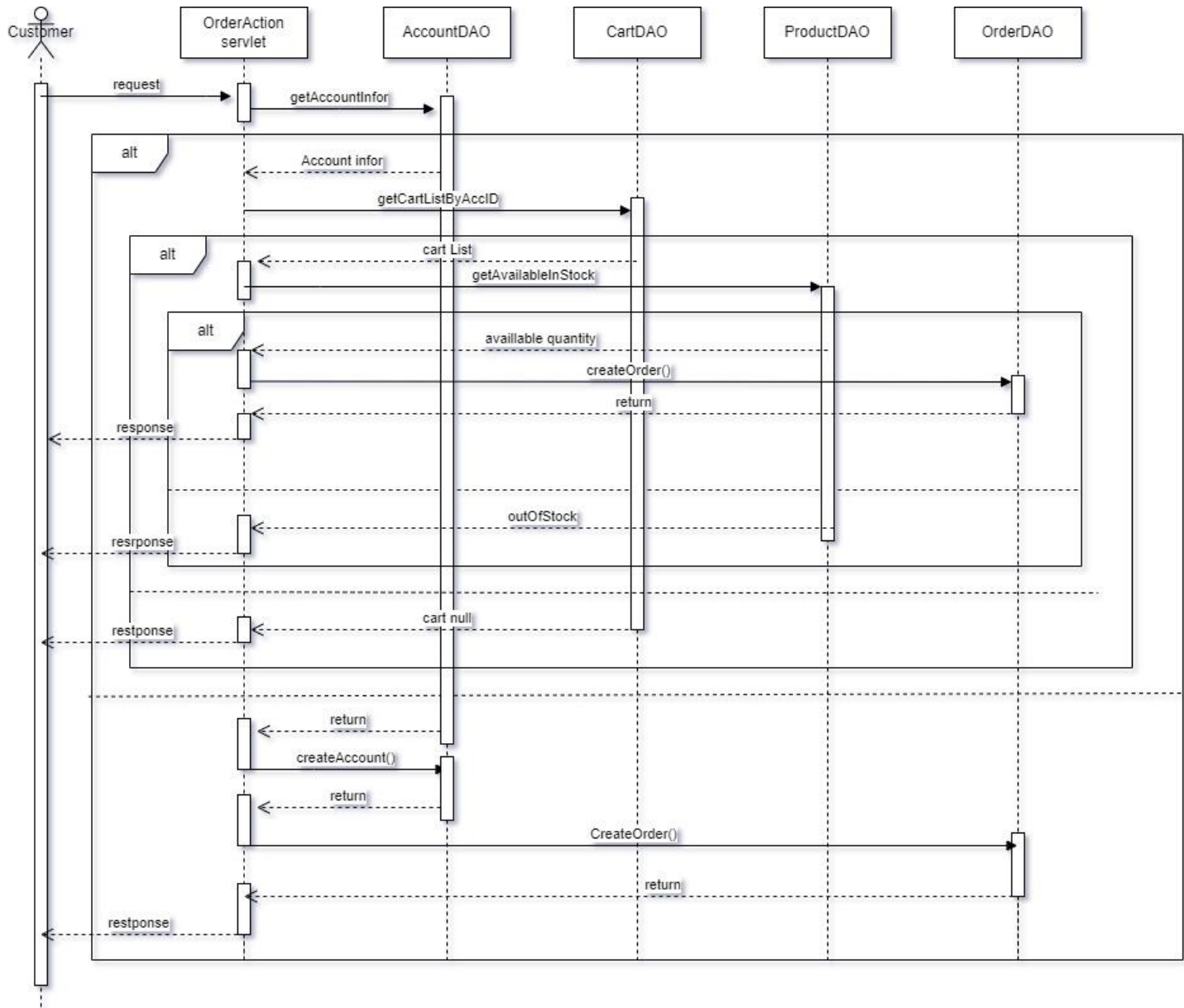
Account Class

No	Method	Description
01	getAccountID()	return account ID of an instance derive from Account class - Input: None - Output: account ID

OrderDAO Class

No	Method	Description
01	+createOrderInDB(Order ,int , String)	create Order field and Order Details into Database - Input: Order information, accountID, voucher - Output: None

c. Sequence Diagram(s)



d. Database Queries

AccountDAO Class

No	Method	Query
01	createAccount()	exec createAccount ?,?,?,?,?,?,?,?,?,?;
02	getAccountByEmail()	select * from Accounts where Email=?

CartDAO Class

No	Method	Query
01	getCartListByAccID() ()	select * from Cart where AccountID=?

ProductDAO Class

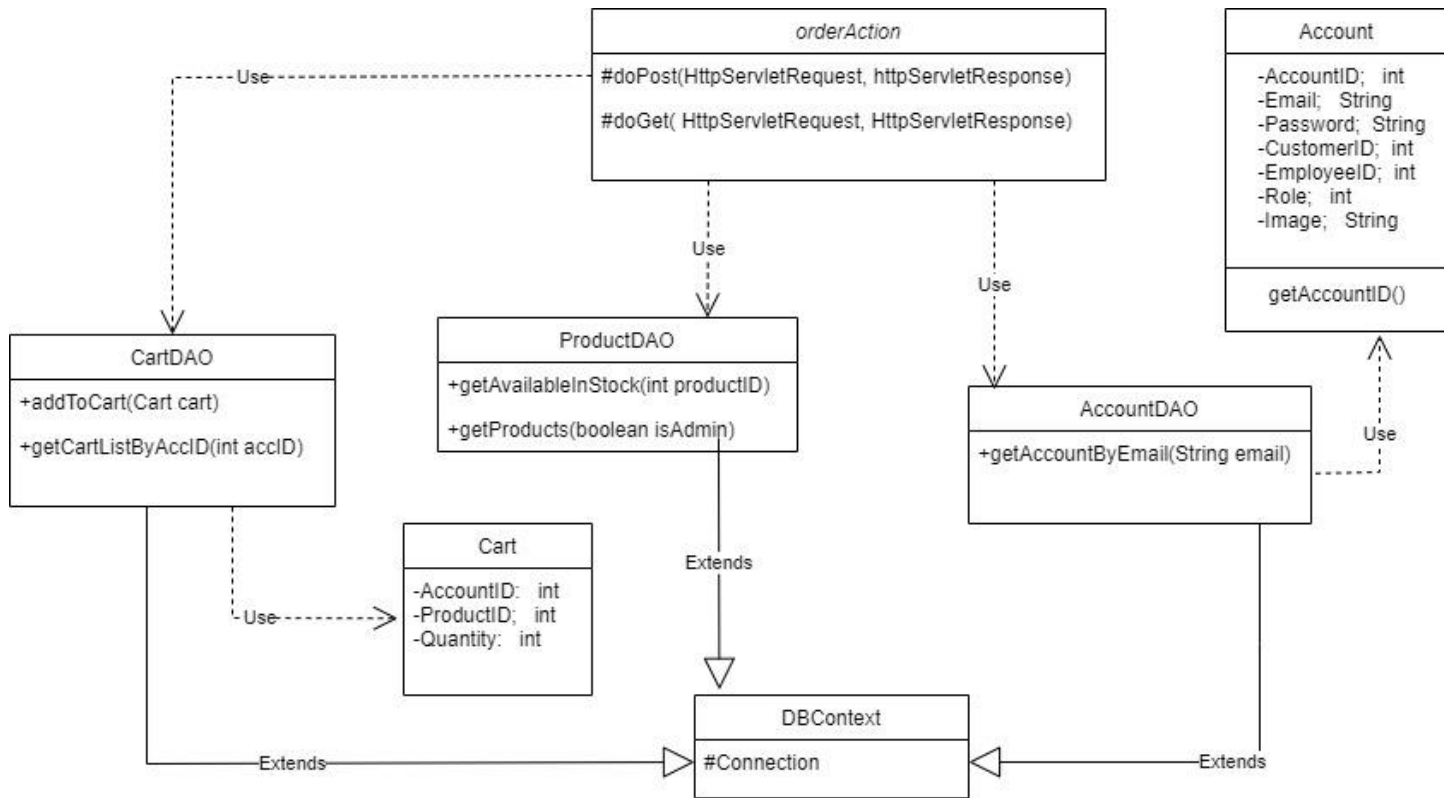
No	Method	Query
01	getAvailableInStock()	<p>"select SUM(UnitsInStock) as [AvailableInStock] from Inventories where ProductID=?"</p> <p>"select SUM(Quantity) as [Pending] from Orders od, [Order Details] odd where od.OrderID=odd.OrderID AND ShippedDate IS NULL AND odd.ProductID = ?"</p>

OrderDAO Class

No	Method	Query
01	getNewOrderID()	select Max(OrderID) as Maximum from Orders
02	createOrderInDB()	exec OrderAction ?,?,?,?,?,,?

1.6. AddToCart

a. Class Diagram



b. Class Specifications

OrderAction Class

No	Method	Description
01	doGet	Called by the server (via the service method) to allow a servlet to handle a GET request.
02	doPost	Called by the server (via the service method) to allow a servlet to handle a POST request.

CartDAO Class

No	Method	Description
01	+getCartListByAccID(int accID)	Get cart list which belongs to one person with accountID - Input: accountID - Output: list of product in cart

02	+addToCart(Cart cart)	adding the product into the cart of the customer - Input: cart information - Output: True if add successful, False if fail
----	-----------------------	--

Cart Class

No	Method	Description
01	getQuantity()	Get number of the same item in cart - Input: None - Output: number of the same item

ProductDAO Class

No	Method	Description
01	+getAvailableInStock(int productID)	Get the list of available inventories of a product - Input: productID - Output: number of available product
02	+getProducts(boolean isAdmin)	Get products list from Database - Input: isAdmin = true: get all product in Database isAdmin = false: get product list is in business currently - Output: list of Product

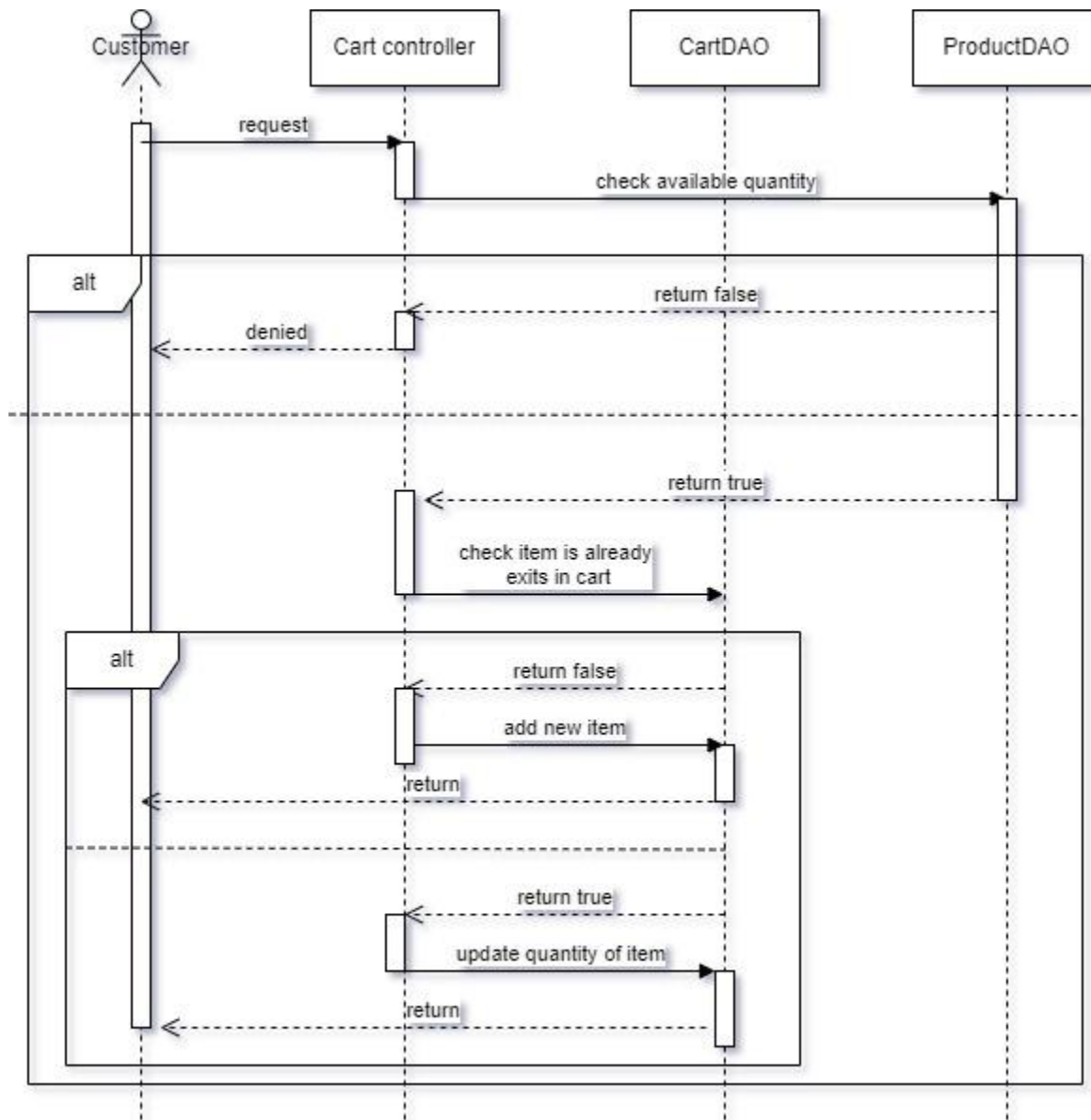
AccountDAO Class

No	Method	Description
01	+getAccountByEmail(String email)	Get account information by email address - Input: email address - Output: account information

Account Class

No	Method	Description
01	getAccountID()	return account ID of an instance derive from Account class - Input: None - Output: account ID

c. Sequence Diagram(s)



d. Database Queries

CartDAO Class

No	Method	Query
01	addToCart()	UPDATE Cart SET Quantity = ? WHERE AccountID = ? AND ProductID = ?

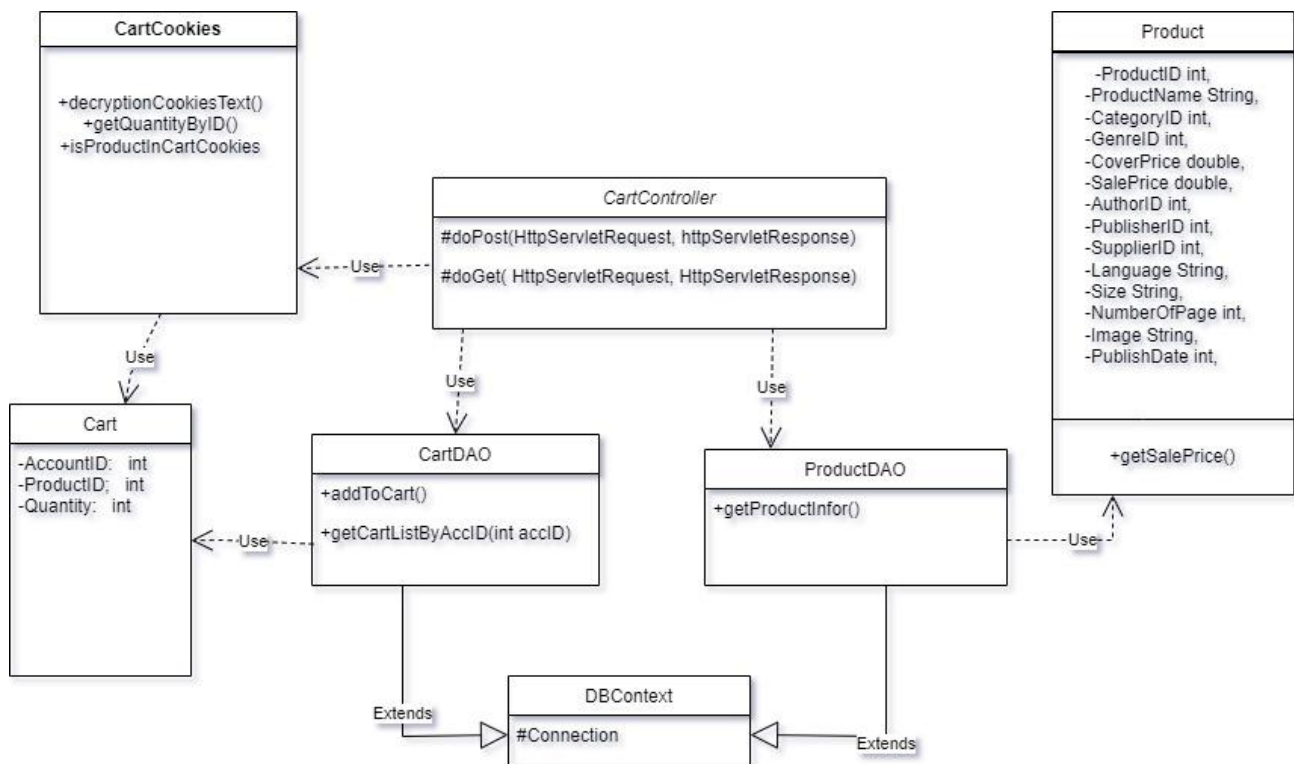
02	updateCart()	UPDATE Cart SET Quantity = ? WHERE AccountID = ? AND ProductID = ?
03	getCartListByAccID()	select * from Cart where AccountID=?

ProductDAO Class

No	Method	Query
01	getProductInfor	select * from Products where ProductID = ?

1.7. Change cart

a. Class Diagram



b. Class Specifications

OrderAction Class

No	Method	Description
01	doGet	Called by the server (via the service method) to allow a servlet to handle a GET request.
02	doPost	Called by the server (via the service method) to allow a servlet to handle a POST request.

CartDAO Class

No	Method	Description
01	+getCartListByAccID(int accID)	Get cart list which belongs to one person with accountID - Input: accountID - Output: list of product in cart
02	+addToCart(Cart cart)	adding the product into the cart of the customer - Input: cart information - Output: True if add successful, False if fail

Cart Class

No	Method	Description
01	getQuantity()	Get number of the same item in cart - Input: None - Output: number of the same item

ProductDAO Class

No	Method	Description
01	+getAvailableInStock(int productID)	Get the list of available inventories of a product - Input: productID - Output: number of available product
02	+getProducts(boolean isAdmin)	Get products list from Database - Input: isAdmin = true: get all product in Database isAdmin = false: get product list is in business currently - Output: list of Product

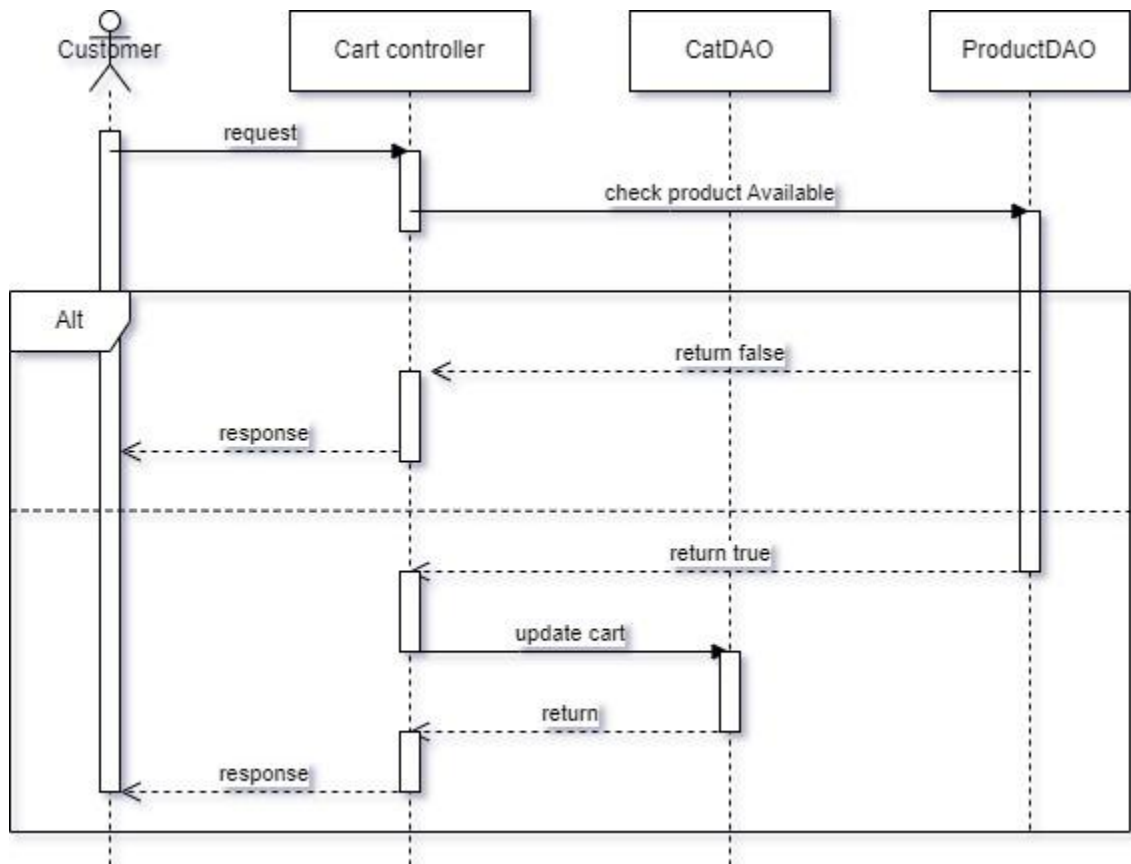
AccountDAO Class

No	Method	Description
01	+getAccountByEmail(String email)	Get account information by email address - Input: email address - Output: account information

Account Class

No	Method	Description
01	getAccountID()	return account ID of an instance derive from Account class - Input: None - Output: account ID

c. Sequence Diagram(s)



d. Database Queries

CartDAO Class

No	Method	Query
----	--------	-------

01	addToCart()	UPDATE Cart SET Quantity = ? WHERE AccountID = ? AND ProductID = ?
02	updateCart()	UPDATE Cart SET Quantity = ? WHERE AccountID = ? AND ProductID = ?
03	getCartListByAccID()	select * from Cart where AccountID=?

ProductDAO Class

No	Method	Query
01	getProductInfor	select * from Products where ProductID = ?

CartDAO Class

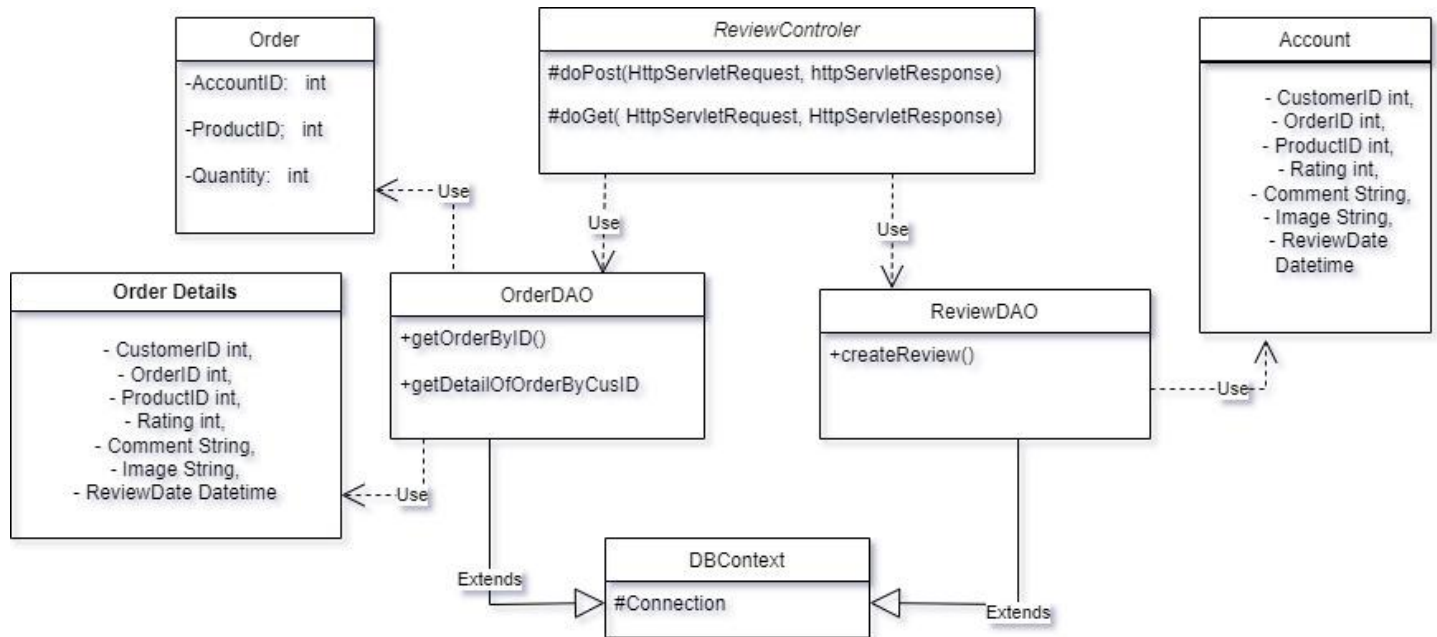
No	Method	Query
01	addToCart()	UPDATE Cart SET Quantity = ? WHERE AccountID = ? AND ProductID = ?
02	updateCart()	UPDATE Cart SET Quantity = ? WHERE AccountID = ? AND ProductID = ?
03	getCartListByAccID()	select * from Cart where AccountID=?

ProductDAO Class

No	Method	Query
01	getProductInfor	select * from Products where ProductID = ?

1.8. Review

a. Class Diagram



b. Class Specifications

Account_profile2_Review Class

No	Method	Description
01	doGet	Called by the server (via the service method) to allow a servlet to handle a GET request.
02	doPost	Called by the server (via the service method) to allow a servlet to handle a POST request.

OrderDAO Class

No	Method	Description
01	getOrderByID()	Get order information - Input: Order ID - Output: Order information
02	getDetailOfOrderByCusID()	get all detail of a order - Input: Customer ID - Output: Detail list of all order of a customer

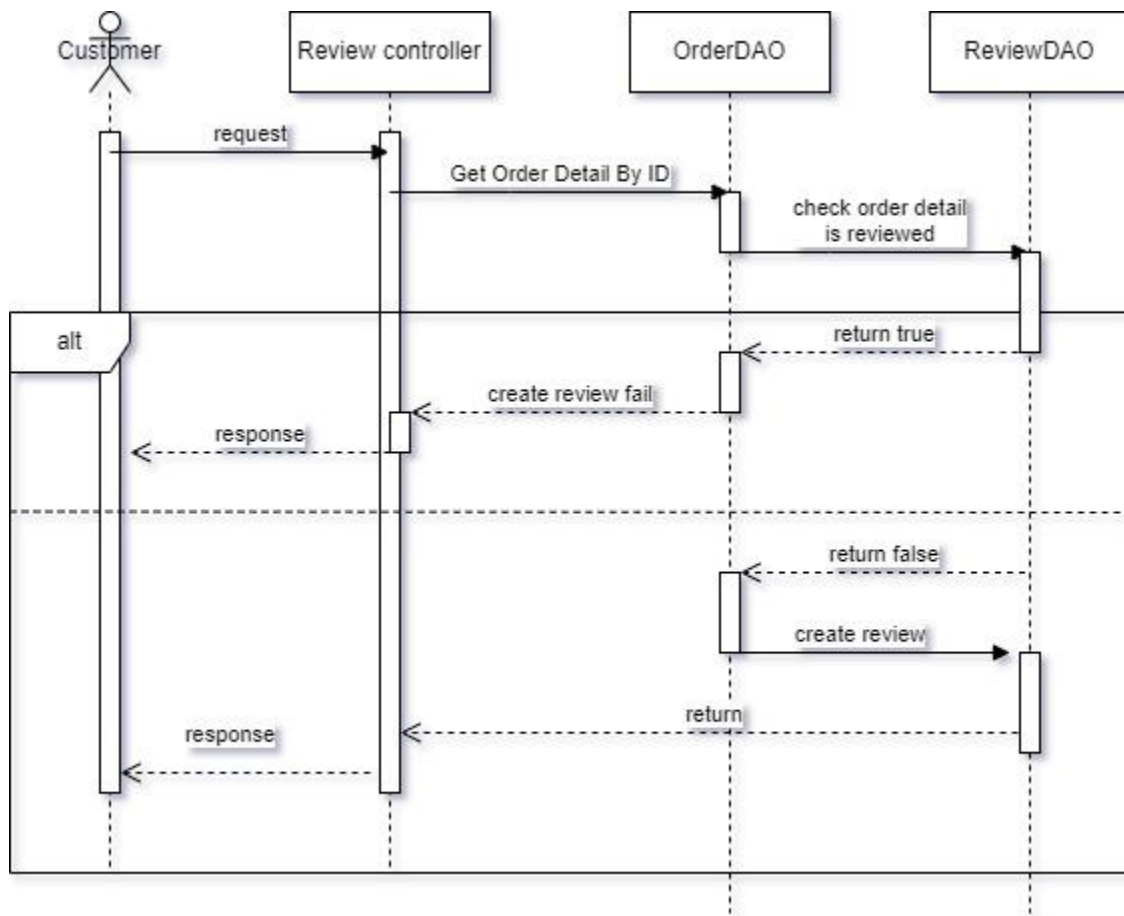
ReviewDAO Class

No	Method	Description
01	createReview()	Create a new review - Input: review information -output: true if review created successfully

ProductDAO Class

No	Method	Description
01	getProductInfor()	Get product information of product by product ID - Input: productID - Output: product's information

c. Sequence Diagram(s)



d. Database Queries

ReviewDAO Class

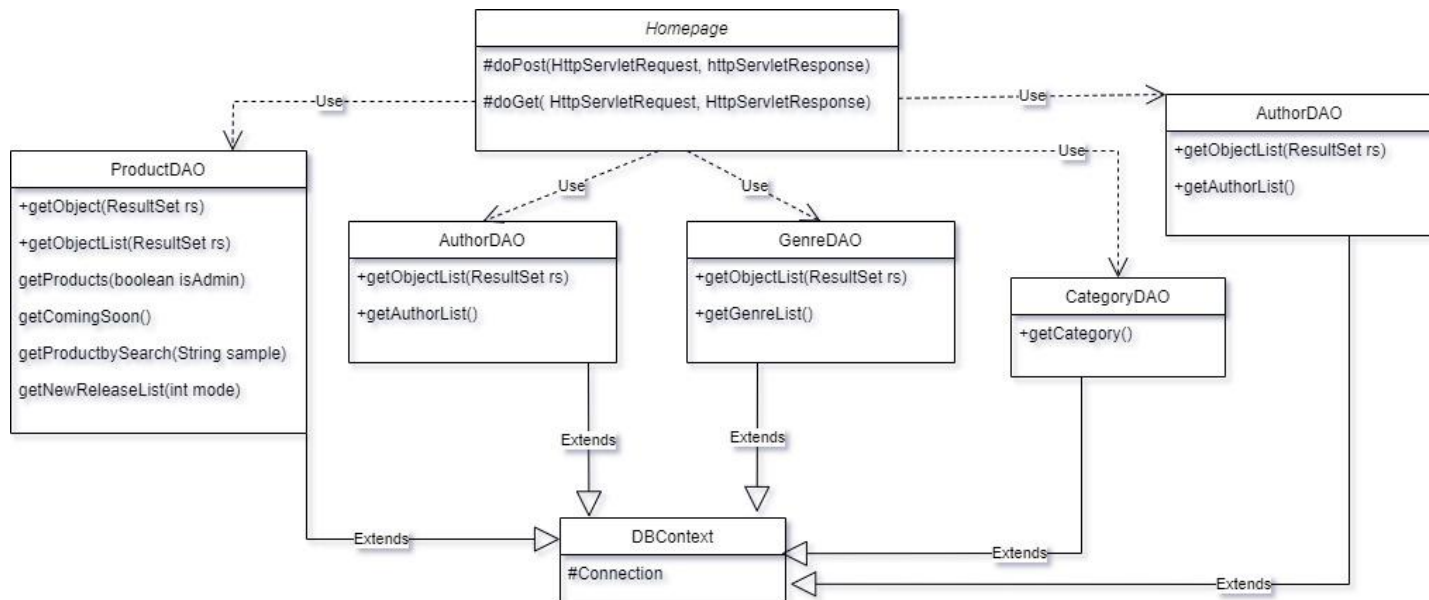
No	Method	Query
01	createReview()	insert into Reviews values(?,?,?,?,?, ?,GETDATE())

OrderDAO Class

No	Method	Query
01	getOrderByID()	select * from Orders WHERE OrderID=?
02	getDetailOfOrder ByCusID()	select * from [Order Details] od , Orders o where od.OrderID=o.OrderID AND o.CustomerID=?

1.9 Product Catalog

a. Class Diagram



b. Class Specifications

Homepage Class

No	Method	Description
01	doGet	Called by the server (via the service method) to allow a servlet to handle a GET request.
02	doPost	Called by the server (via the service method) to allow a servlet to handle a POST request.

AuthorDAO Class

No	Method	Description
01	getObjectList	Get the object list by ResultSet pass from getAuthorList() - Input: list ResultSet - Output: list of Author
02	getAuthorList	Get the Author list - Input: None - Output: list of Author

CategoryDAO Class

No	Method	Description
01	getCategory	Get the Category list - Input: None - Output: list of Category

GenreDAO Class

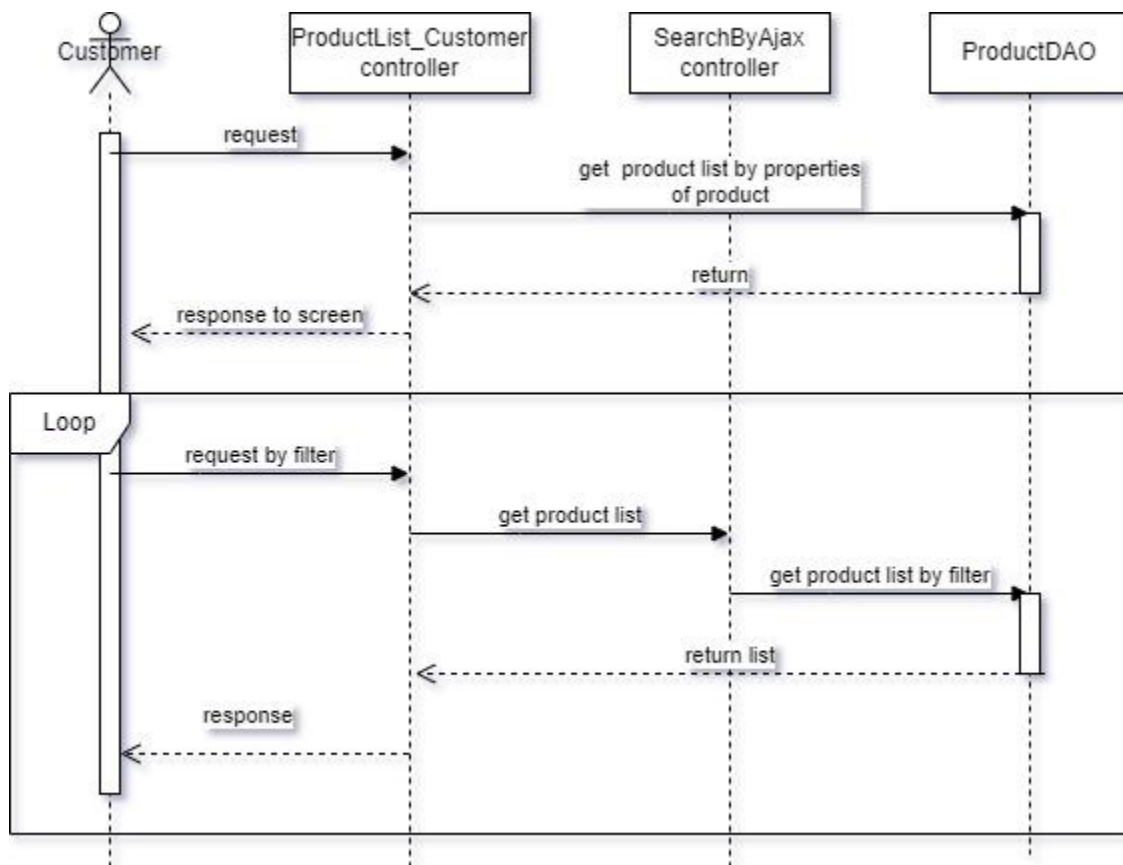
No	Method	Description
01	getObjectList	Get the object list by ResultSet pass from getGenreList() - Input: list ResultSet - Output: list of Genre
02	getGenreList	Get the Genre list - Input: None - Output: list of Genre

ProductDAO Class

No	Method	Description
01	getObjectList	Get the object list by ResultSet - Input: list ResultSet - Output: list of Product
02	getObject	Get the object

		- Input: None - Output: an instance of Product
03	getProducts(boolean isAdmin)	Get products - Input: isAdmin = true: get all product in Database isAdmin = false: get product list is in business currently - Output: list of Product
04	getComingSoon()	Get products which do not release - Input: None - Output: a list of Product which does not release
05	getProductbySearch(String sample)	Get products which is similar with a pattern - Input: None - Output: a list of Product
06	getNewReleaseList()	Get products which do not release yet - Input: sample (pattern) which user want to find out - Output: a list of Product which does not release

c. Sequence Diagram(s)



d. Database Queries

ProductDAO Class

No	Method	Query
01	getProducts()	"select * from Products" "select * from Products where Discontinued=?"
02	getNewReleaseList()	"select TOP 3 * from Products WHERE PublishDate IS NOT NULL AND PublishDate < getdate() AND Discontinued = 0 Order by PublishDate DESC" "select * from Products WHERE PublishDate IS NOT NULL AND Discontinued = 0 Order by PublishDate DESC"
03	getComingSoon()	"select TOP 1 * from Products where PublishDate > getdate() order by PublishDate ASC"

CategoryDAO Class

No	Method	Query
01	getCategory()	SELECT * FROM Categories

GenreDAO Class

No	Method	Query
01	getGenreList()	select * from Genre

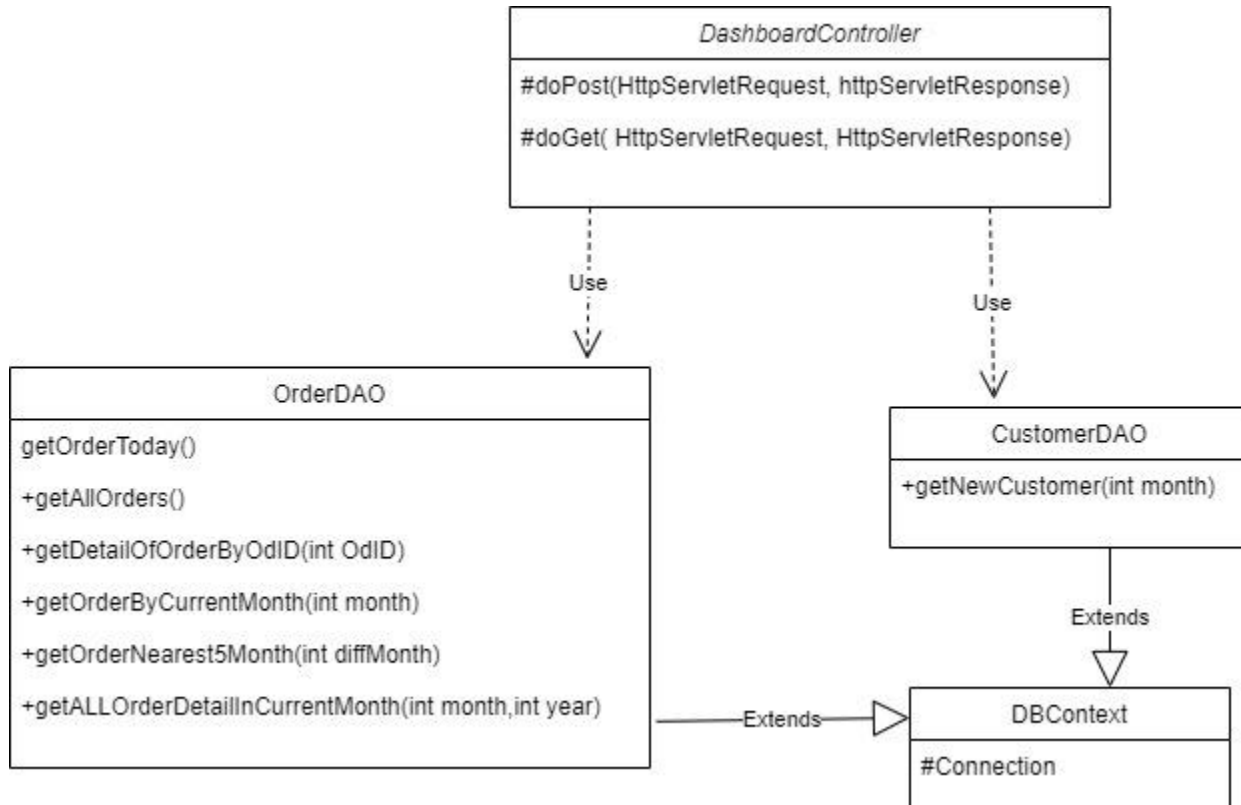
AuthorDAO Class

No	Method	Query
01	getAuthorList()	select * from Authors

2. Admin features

2.1 Dashboard

a. Class Diagram



b. Class Specifications

DashboardController Class

No	Method	Description
01	doGet	Called by the server (via the service method) to allow a servlet to handle a GET request.
02	doPost	Called by the server (via the service method) to allow a servlet to handle a POST request.

OrderDAO Class

No	Method	Description
----	--------	-------------

01	getObjectList	Get the object list by ResultSet pass from getAuthorList() - Input: list ResultSet - Output: list of Author
02	getAuthorList	Get the Author list - Input: None - Output: list of Author

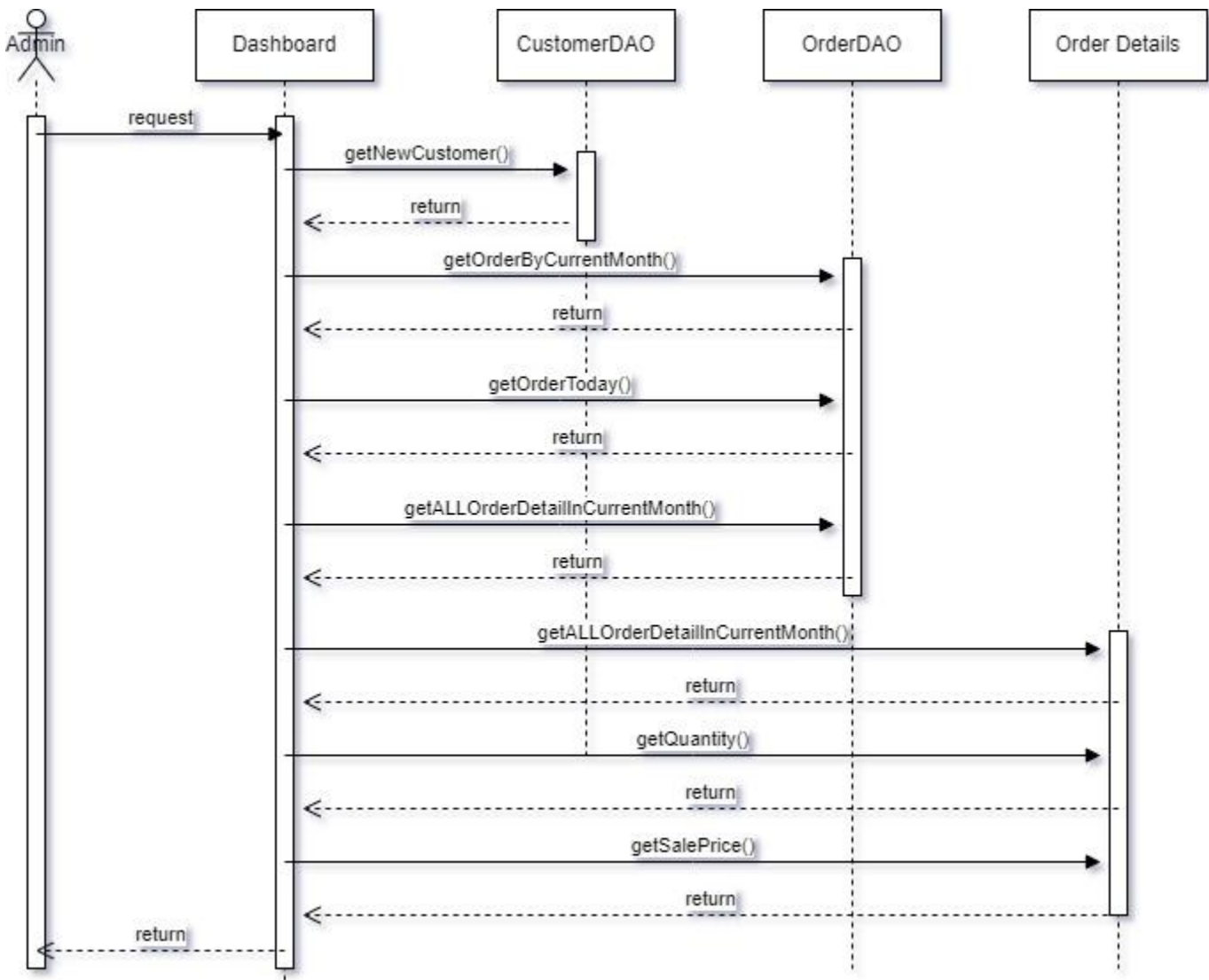
Class

No	Method	Description
01	getCategory	Get the Category list - Input: None - Output: list of Category

CustomerDAOClass

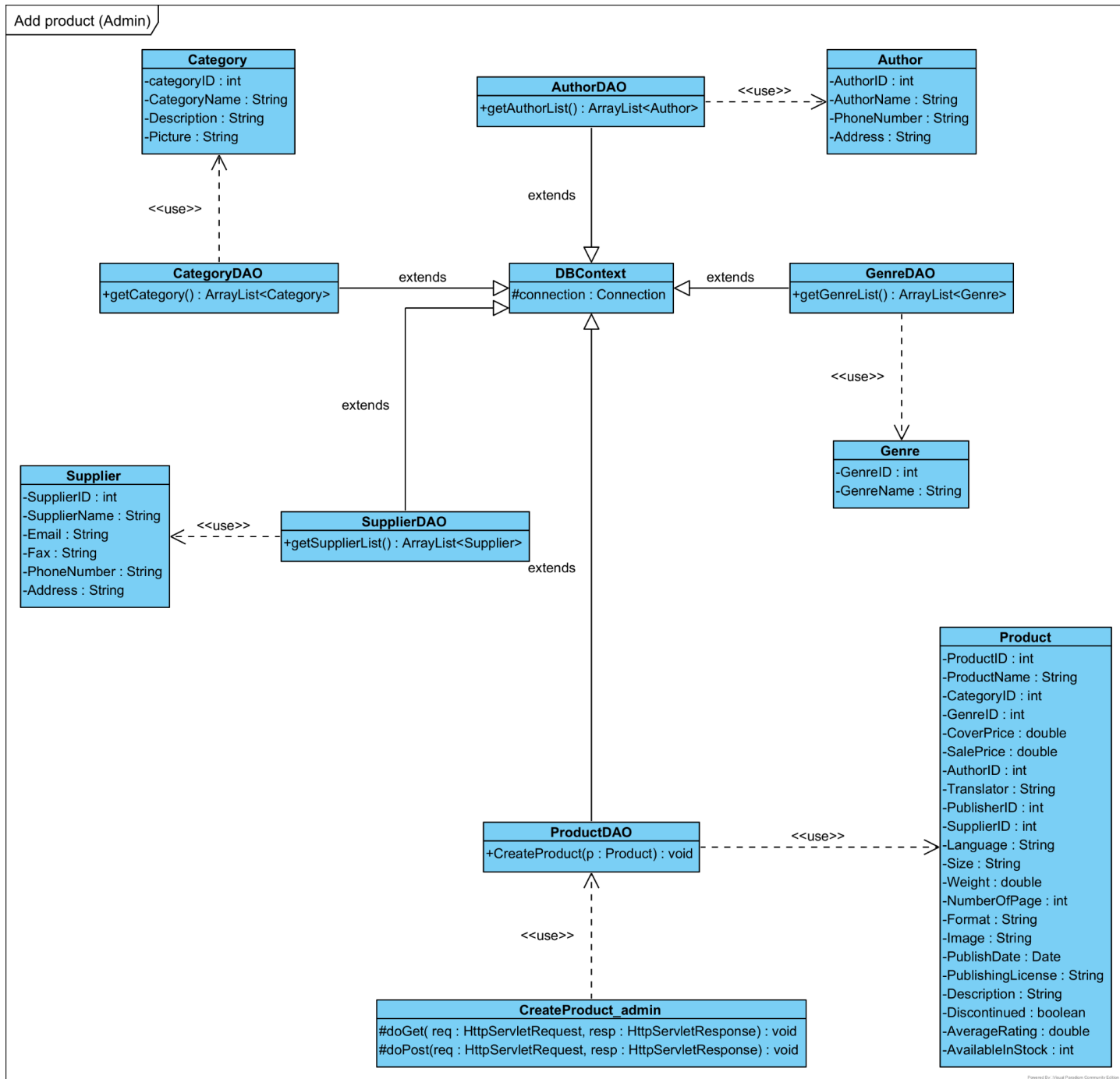
No	Method	Description
01	getObjectList	Get the object list by ResultSet pass from getGenreList() - Input: list ResultSet - Output: list of Genre
02	getGenreList	Get the Genre list - Input: None - Output: list of Genre

c. Sequence Diagram(s)



2.2. Add New Product

a. Class Diagram



b. Class Specifications

CreateProduct_admin Class

No	Method	Description
----	--------	-------------

01	doGet	Called by the server (via the service method) to allow a servlet to handle a GET request.
02	doPost	Called by the server (via the service method) to allow a servlet to handle a POST request.

ProductDAO Class

No	Method	Description
01	CreateProduct(Product p)	Catch request from client to create a new product. - Input: The information that is needed to create a new product. - Output: none

CategoryDAO Class

No	Method	Description
01	getCategory()	Catch request from client to get category list. - Input: none. - Output: list of categories.

SupplierDAO Class

No	Method	Description
01	getSupplierList()	Catch request from client to get supplier list. - Input: none. - Output: list of suppliers.

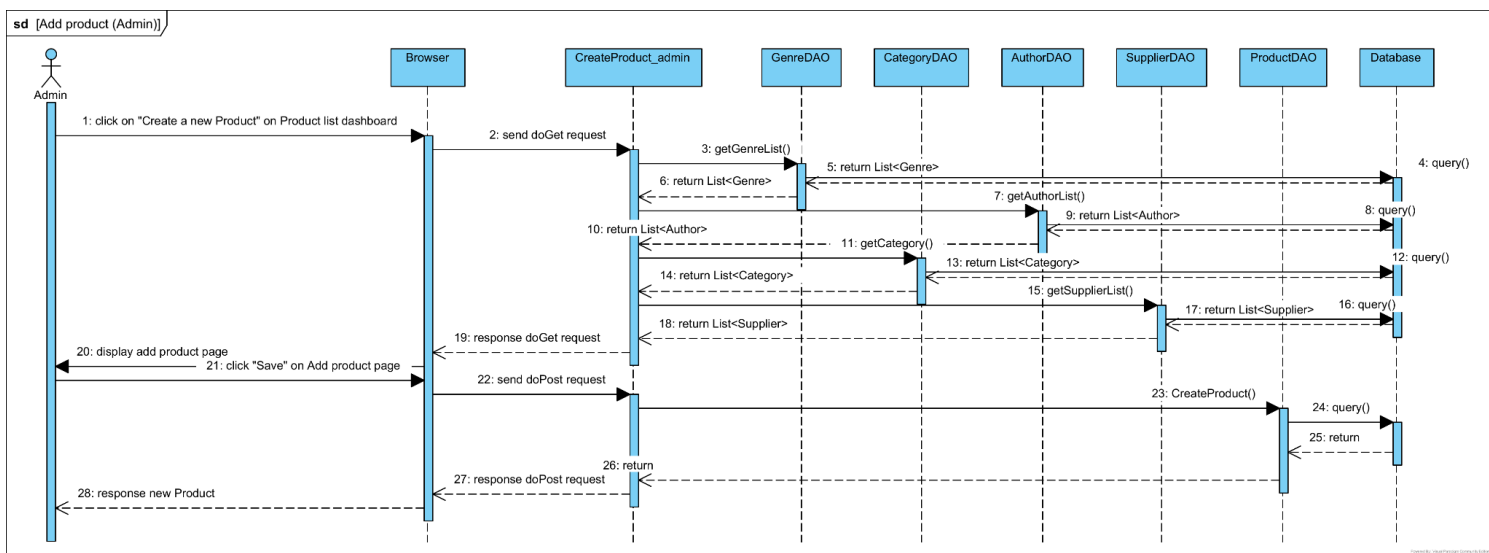
AuthorDAO Class

No	Method	Description
01	getAuthorList()	Catch request from client to get author list. - Input: none. - Output: list of authors.

GenreDAO Class

No	Method	Description
01	getGenreList()	Catch request from client to get genre list. - Input: none. - Output: list of genres.

c. Sequence Diagram(s)



d. Database Queries

GenreDAO

No	Method	Query
01	getGenreList()	select * from Genre

AuthorDAO

No	Method	Query
01	getAuthorList()	select * from Authors

CategoryDAO

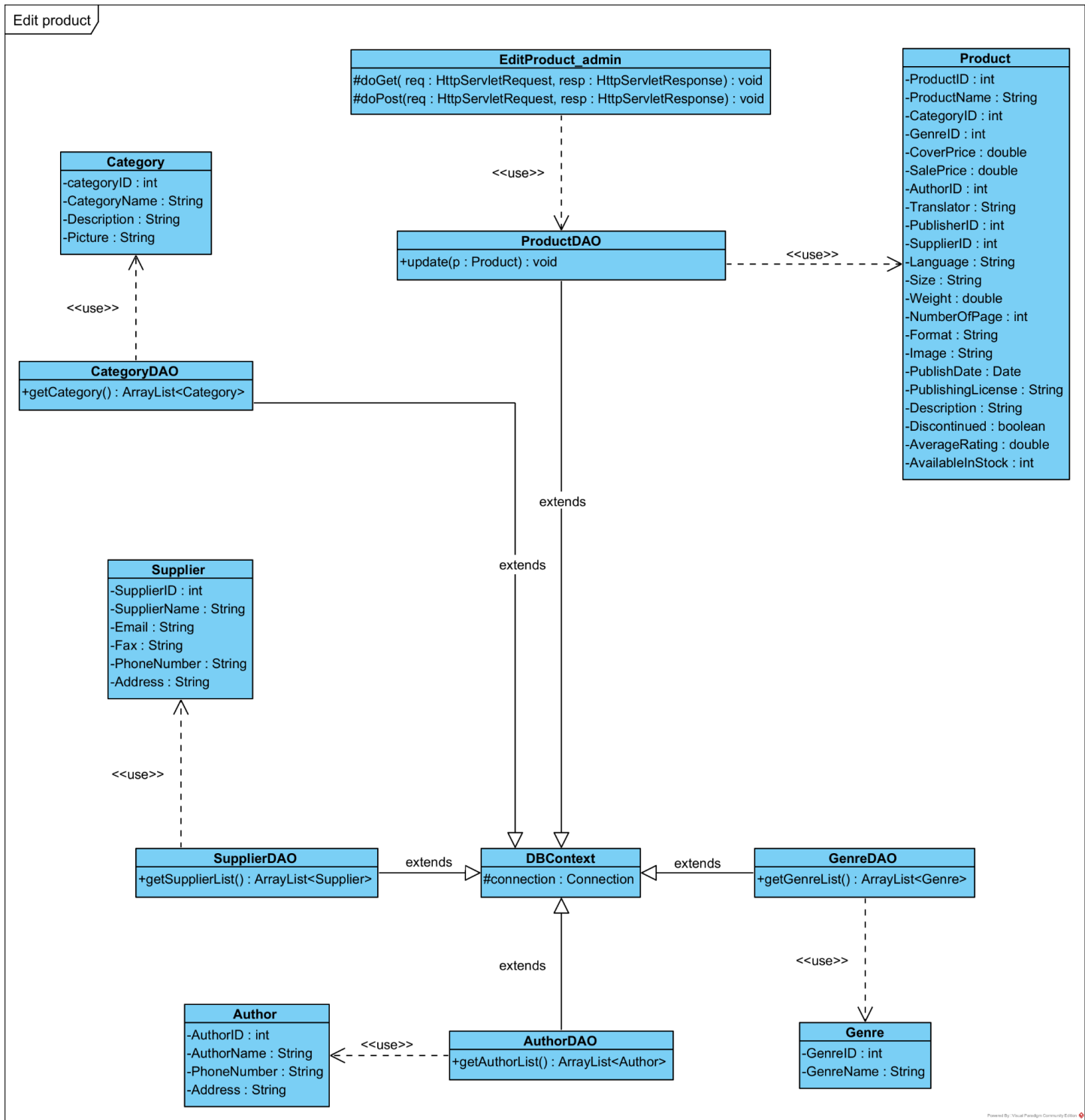
No	Method	Query
01	getCategory()	SELECT * FROM Categories

SupplierDAO

No	Method	Query
01	getSupplierList()	select * from Supplier

2.3. Edit Product

a. Class Diagram



b. Class Specifications

EditProduct_admin Class

No	Method	Description
01	doGet	Called by the server (via the service method) to allow a servlet to handle a GET request.
02	doPost	Called by the server (via the service method) to allow a servlet to handle a POST request.

ProductDAO Class

No	Method	Description
01	update(Product p)	Catch request from client to update a product. - Input: The information that is needed to update a product. - Output: none

CategoryDAO Class

No	Method	Description
01	getCategory()	Catch request from client to get category list. - Input: none. - Output: list of categories.

SupplierDAO Class

No	Method	Description
01	getSupplierList()	Catch request from client to get supplier list. - Input: none. - Output: list of suppliers.

AuthorDAO Class

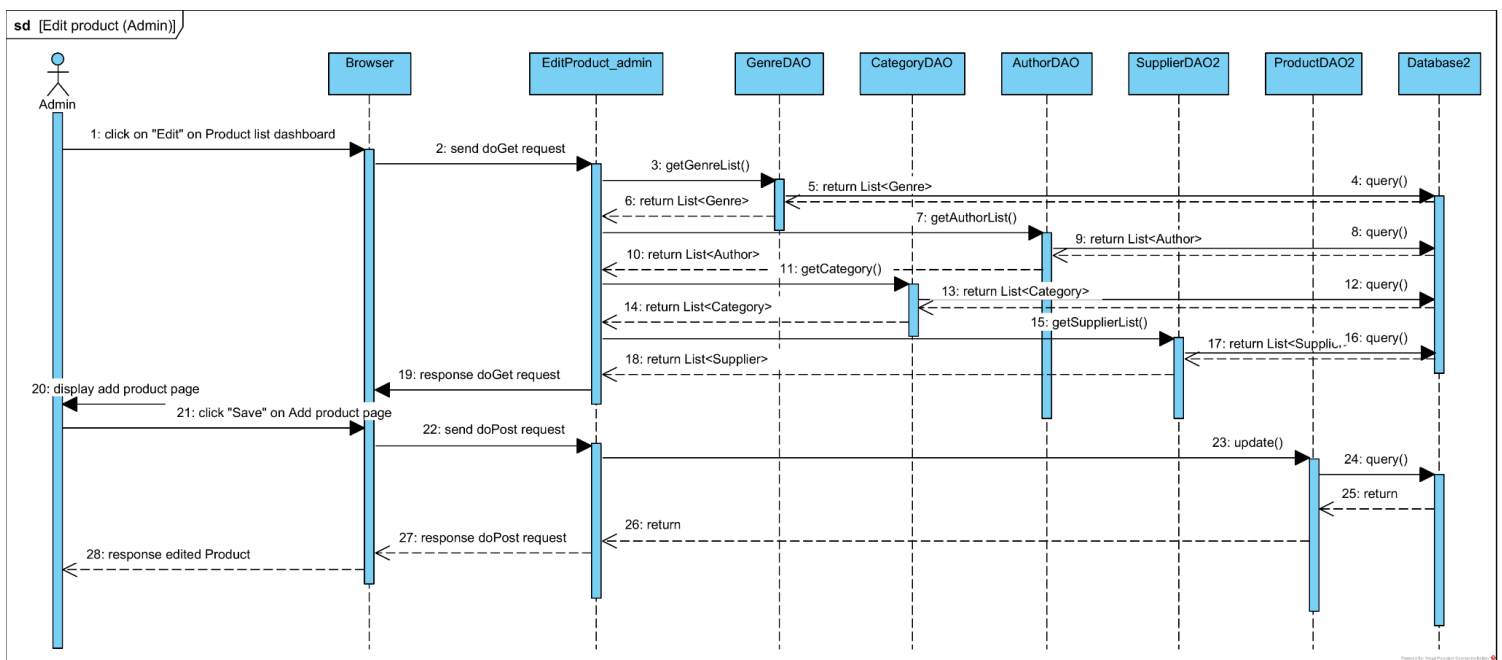
No	Method	Description
----	--------	-------------

01	getAuthorList()	Catch request from client to get author list. - Input: none. - Output: list of authors.
----	-----------------	---

GenreDAO Class

No	Method	Description
01	getGenreList()	Catch request from client to get genre list. - Input: none. - Output: list of genres.

c. Sequence Diagram(s)



d. Database Queries

GenreDAO

No	Method	Query
01	getGenreList()	select * from Genre

--	--	--

AuthorDAO

No	Method	Query
01	getAuthorList()	select * from Authors

CategoryDAO

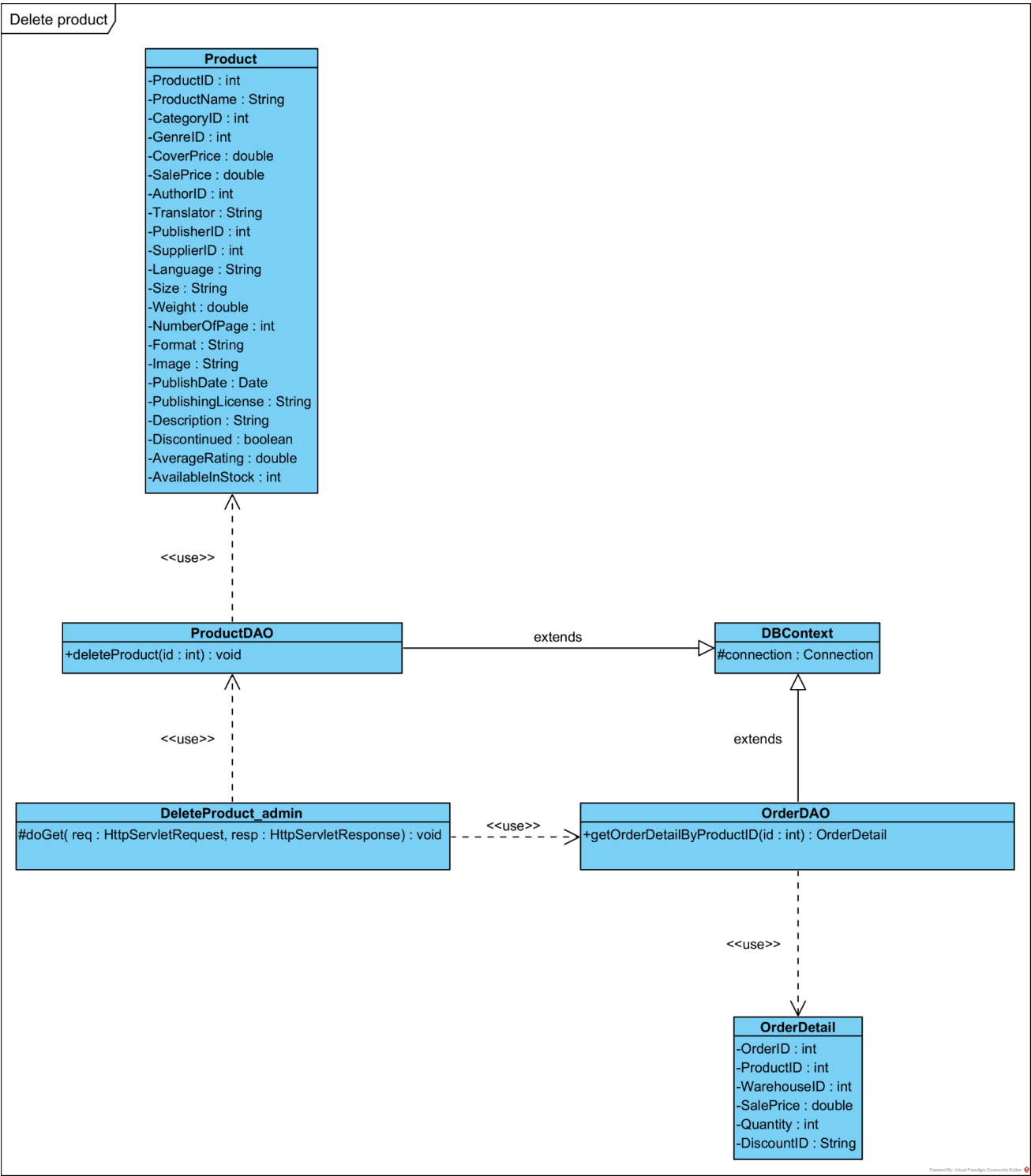
No	Method	Query
01	getCategory()	SELECT * FROM Categories

SupplierDAO

No	Method	Query
01	getSupplierList()	select * from Supplier

2.4. Delete product

a. Class Diagram



b. Class Specifications

DeleteProduct_admin Class

No	Method	Description
01	doGet	Called by the server (via the service method) to allow a servlet to handle a GET request.

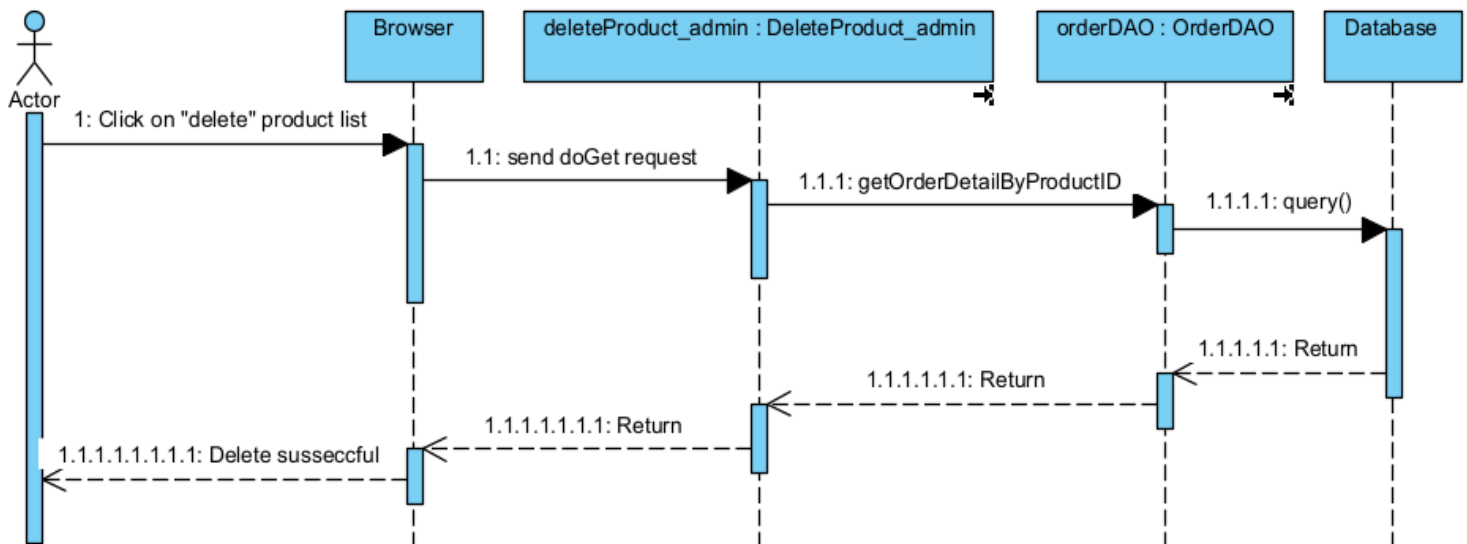
ProductDAO Class

No	Method	Description
01	deleteProduct(int id)	Catch request from client to delete a product. - Input: The id of the product want to delete. - Output: none

OrderDAO Class

No	Method	Description
01	getOrderDetailByProductID(int id)	Catch request from client to get category list. - Input: The id of the product want to find. - Output: order detail of the product with id.

c. Sequence Diagram(s)



d. Database Queries

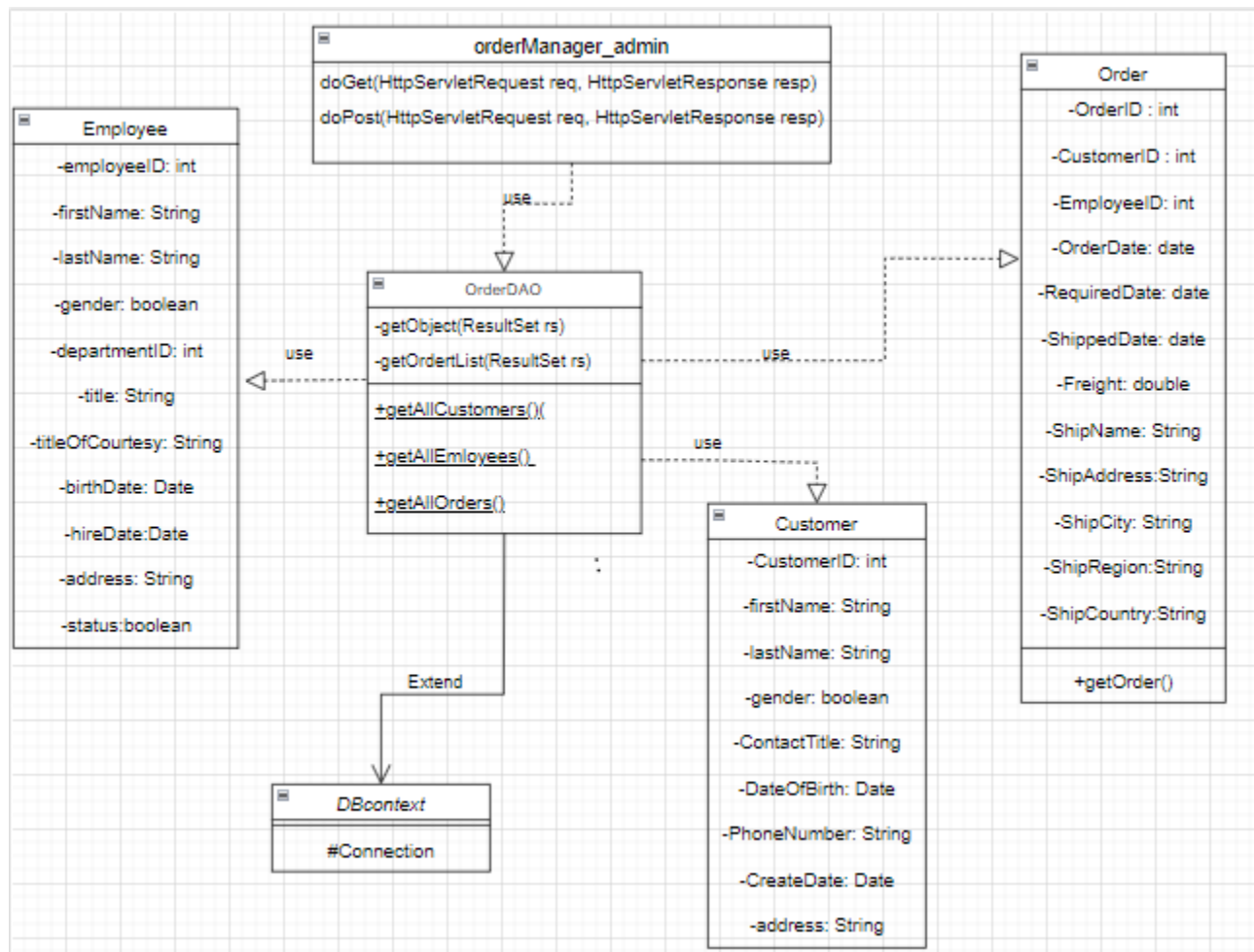
OrderDetailDAO

No	Method	Query
01	getOrderDetailByProductID()	select * from [Order Details] where ProductID=?

3. Employee feature

3.1: View Order

a. Class Diagram



b. Class Specifications

orderManager_admin Class

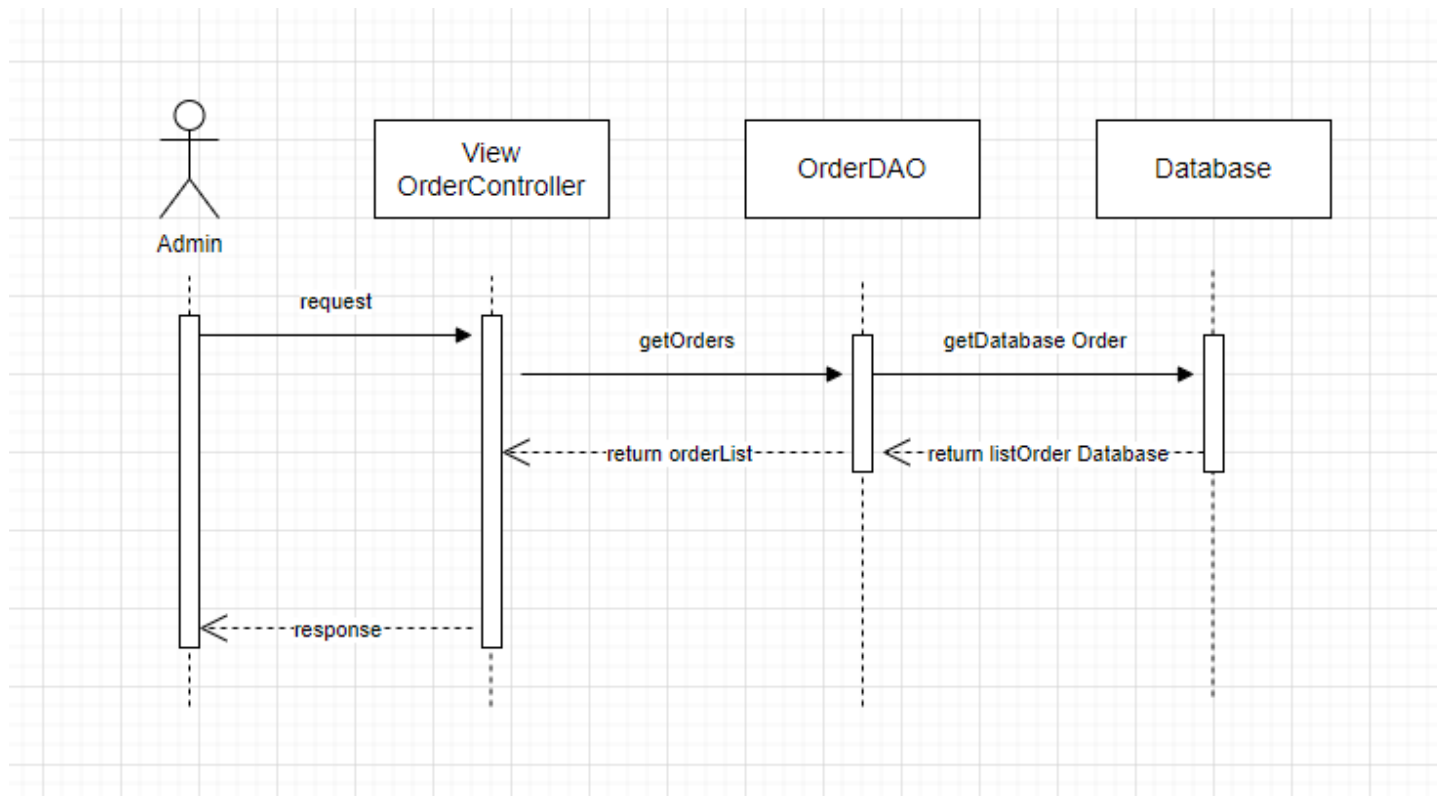
No	Method	Description
01	doGet	Called by the server (via the service method) to allow a servlet to handle a GET request.
02	doPost	is Called by the server (via the service method) to allow a servlet to handle a POST request.

OrderDAO Class

No	Method	Description
01	getObjectList	Get the object list by ResultSet pass from getOrderList() - Input: list ResultSet - Output: list of Order
02	getOrderList	Get the Orderlist

		- Input: None - Output: list of Order
03	getAllCustomers()	Get all Customer -input : Non -output: List order by CustomerID
04	getAllEmployees()	Get all Employees -input: Non -output: List order by EmployeeID
05	getAllOrders()	Get the Orderlist - Input: None - Output: list all of Order

c. Sequence Diagram(s)



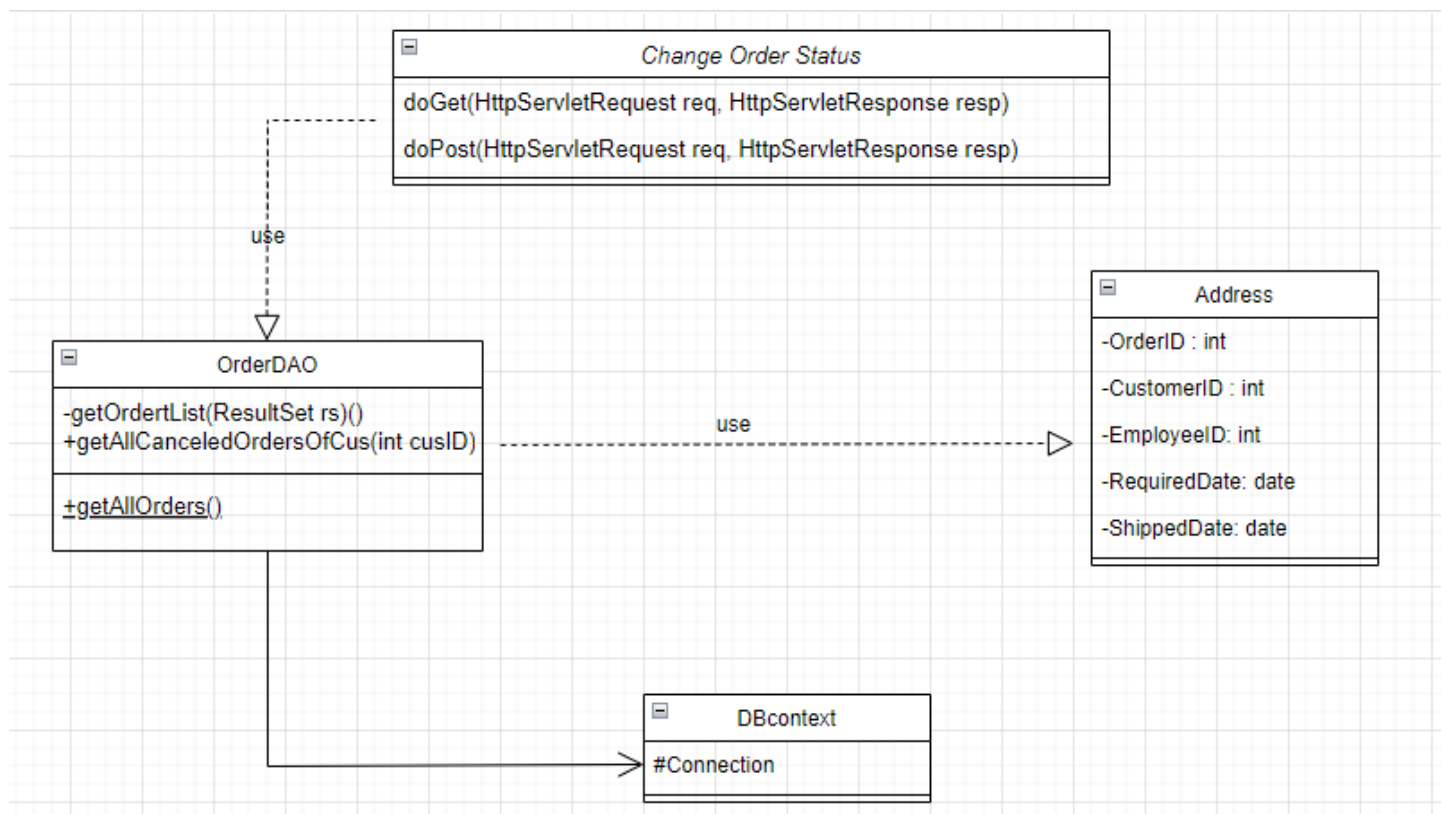
d. Database Queries

OrderDAO Class

No	Method	Query
01	getAllOrders()	select * from Orders
02	getAllCustomer()	select * from Customer
03	getAllEmployee()	select * from Employee

3.2. Change order status

a. Class diagram



b. Class Specifications

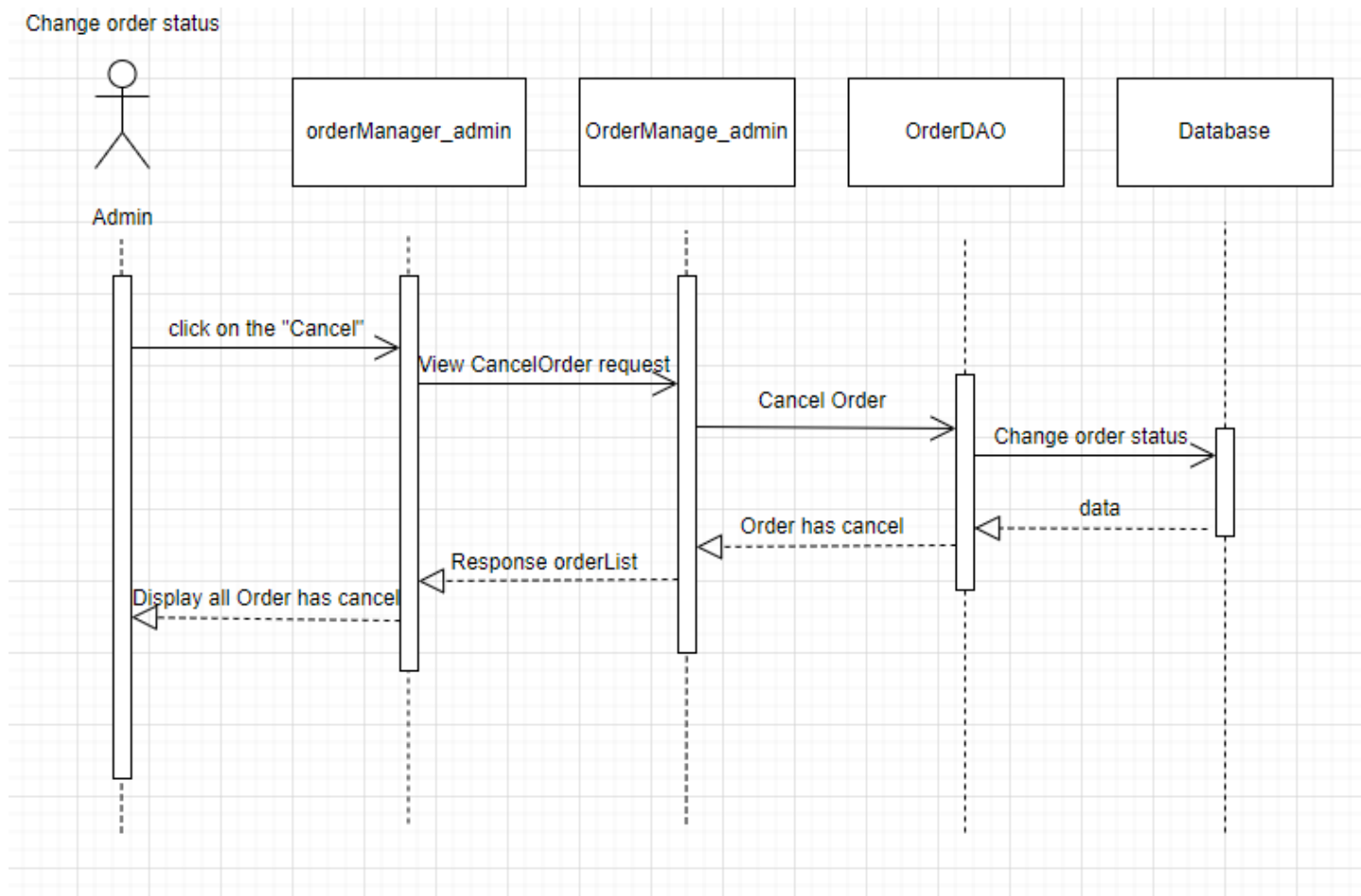
Change Order Status Class

	Method	Description
	doGet	Called by the server (via the service method) to allow a servlet to handle a GET request.
	doPost	is Called by the server (via the service method) to allow a servlet to handle a POST request.

OrderDAO Class

	Method	Description
	getOrderList	Get the Orderlist - Input: None - Output: list of Order
	getAllOrders()	Get all Order by cusID -input : Non -output: List order
	getAllCanceledOrdersOfCus(int cusID)	Get all order has cancel by customerID -input: Non -output: List order has Cancel in list order by customerID

c. Sequence Diagram(s)



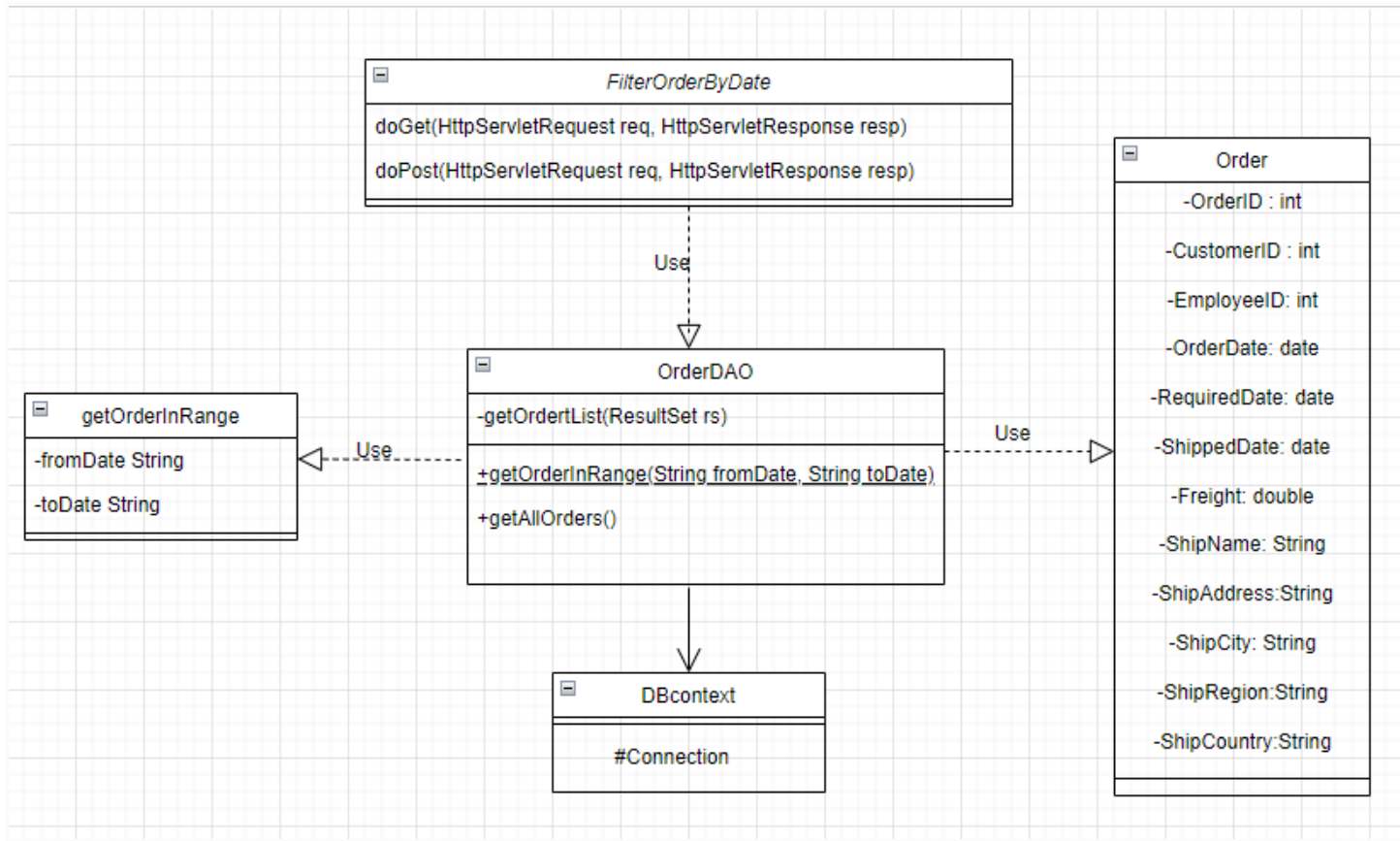
d. Database Queries

OrderDAO Class

No	Method	Query
01	getAllOrders()	select * from Orders if RequiredDate != null && ShippedDate == null =>Cancel

3.3. Filter Order By Date

a. Class diagram



b. Class Specifications

FilterOrderByDate Class

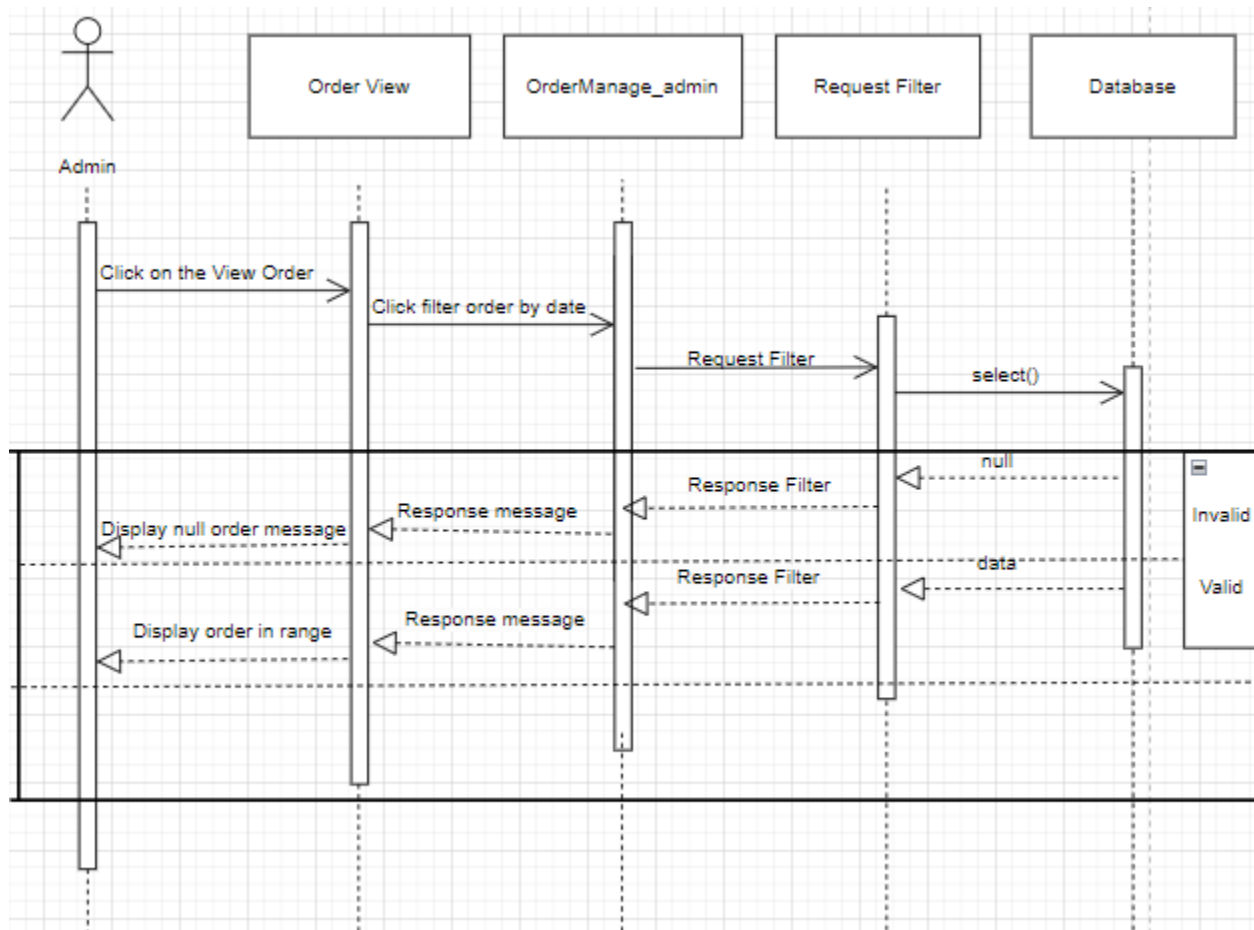
No	Method	Description
01	doGet	Called by the server (via the service method) to allow a servlet to handle a GET request.
02	doPost	is Called by the server (via the service method) to allow a servlet to handle a POST request.

OrderDAO Class

No	Method	Description
02	getOrderList	Get the Orderlist - Input: None - Output: list of Order
03	getAllOrders()	Get all Order by cusID -input : Non

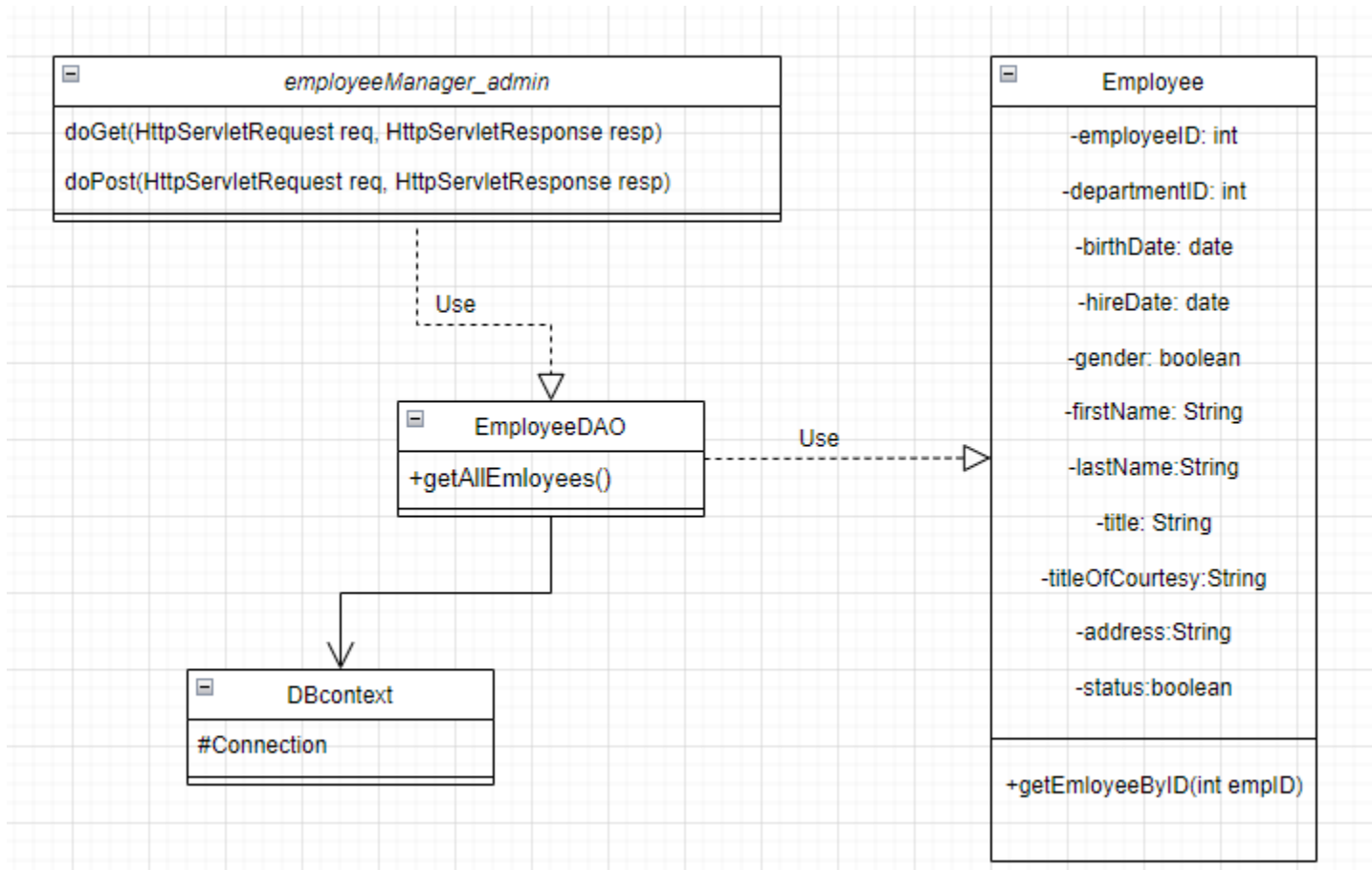
		-output: List order
04	+getOrderInRange(String fromDate, String toDate)	Get all order has in range -input: Date (from-to) -output: List order in range

c. Sequence Diagram(s)



3.4. View Employee

a.class diagram



b. Class Specifications

employeeManager_admin class

No	Method	Description
01	doGet	Called by the server (via the service method) to allow a servlet to handle a GET request.
02	doPost	is Called by the server (via the service method) to allow a servlet to handle a POST request.

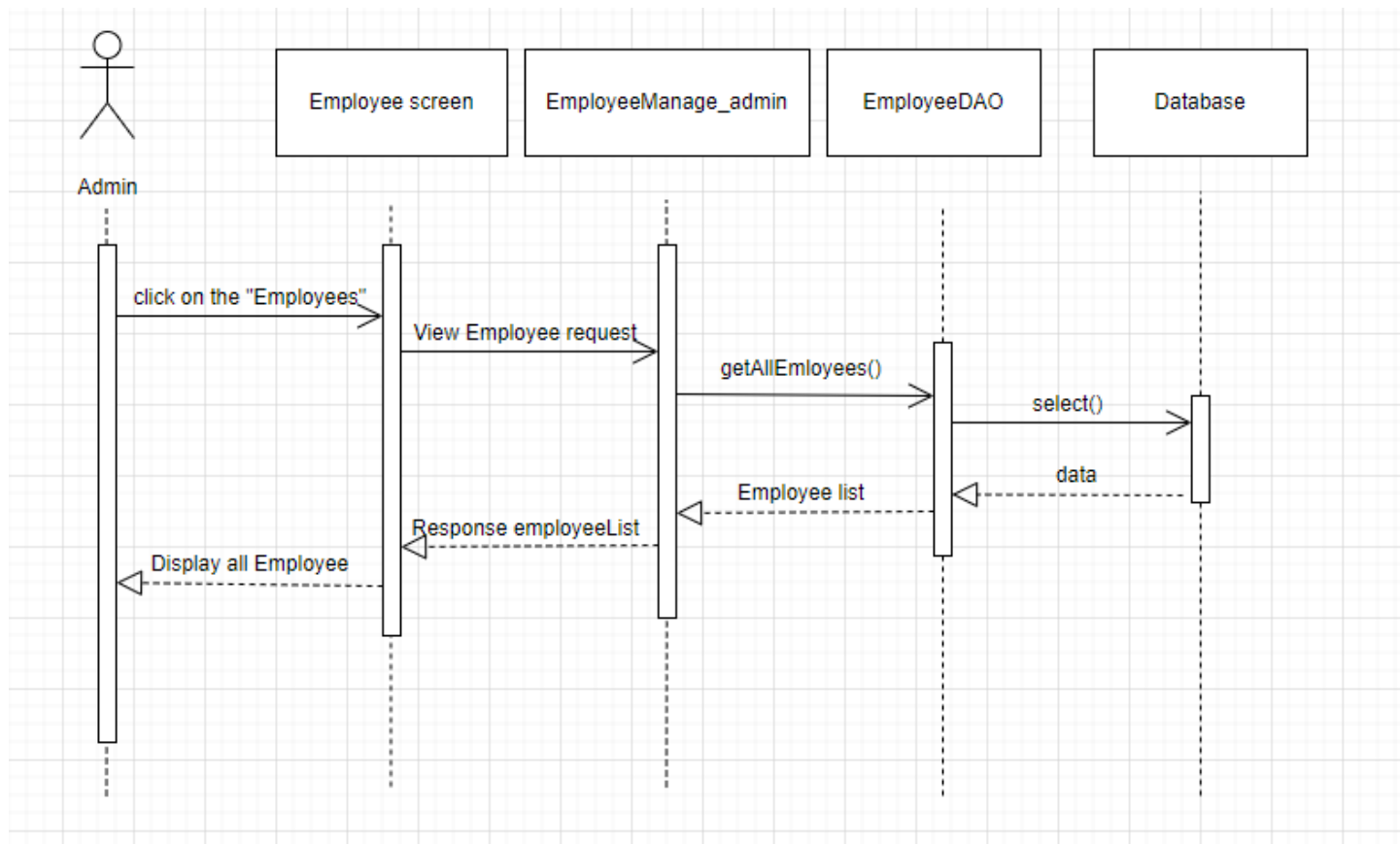
EmployeeDAO Class

No	Method	Description
01	getAllEmployees() ()	Get all Employees have status = 1 - Input: None - Output: list of employees

EmployeeByID Class

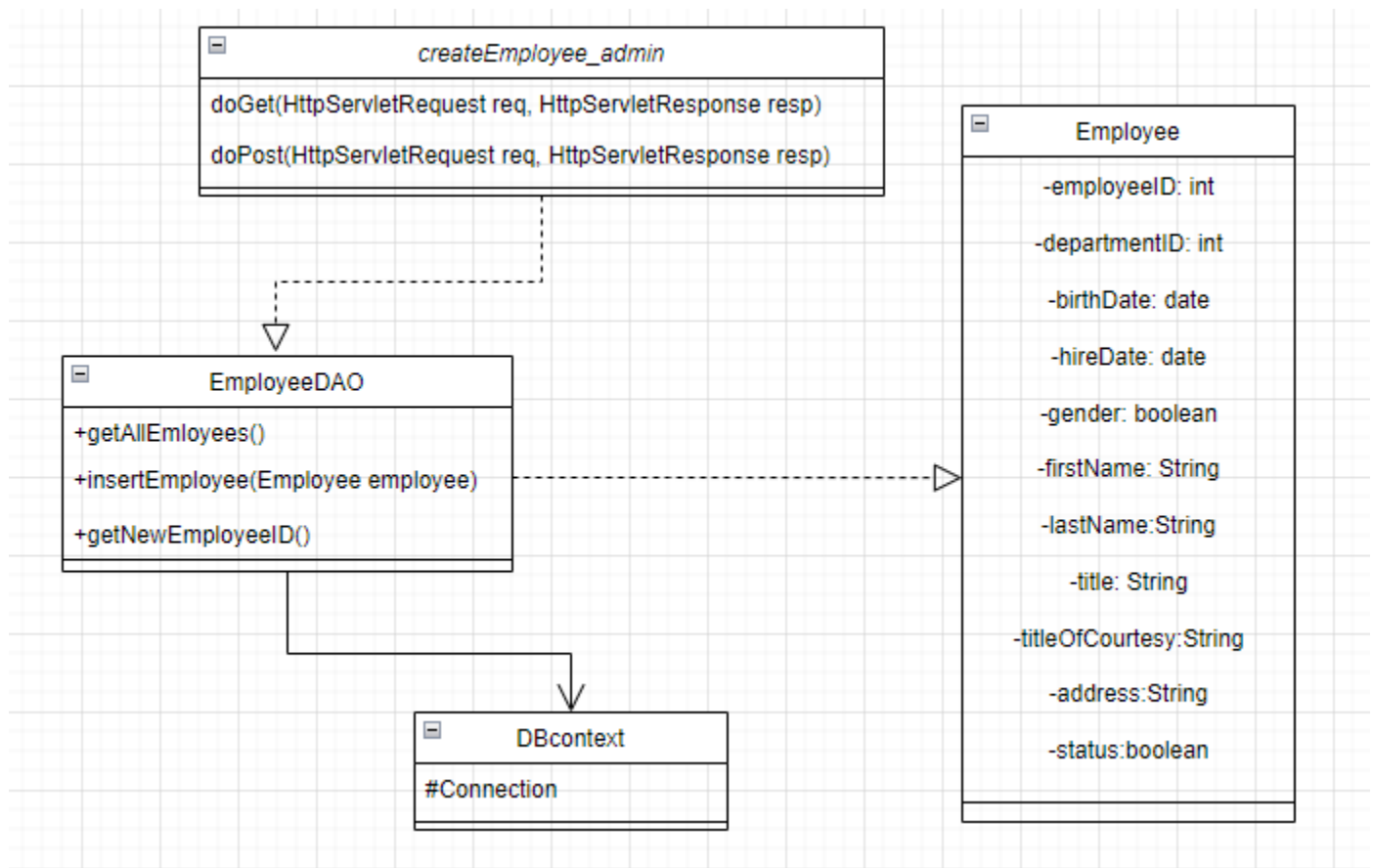
No	Method	Description
01	+getEmployeeByID (int empID)	Get all Employees by employeeID - Input: None - Output: list of employees

c. Sequence Diagram(s)



3.5. Create Employees

a.class diagram



b. Class Specifications

createEmployee_admin class

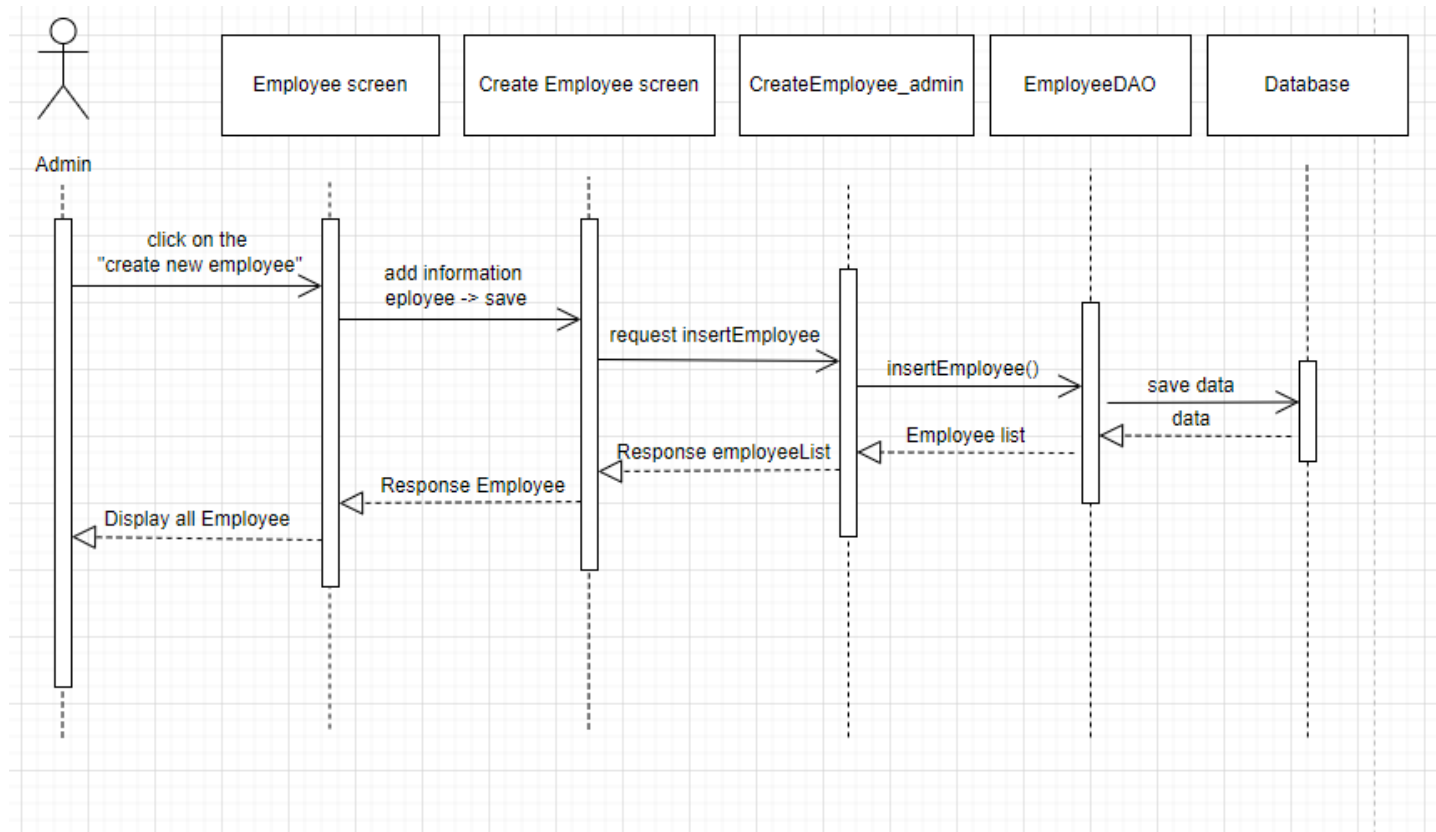
No	Method	Description
01	doGet	Called by the server (via the service method) to allow a servlet to handle a GET request.
02	doPost	is Called by the server (via the service method) to allow a servlet to handle a POST request.

EmployeeDAO

No	Method	Description
01	getAllEmployees()	Get all Employees have status = 1 - Input: None - Output: list of employees

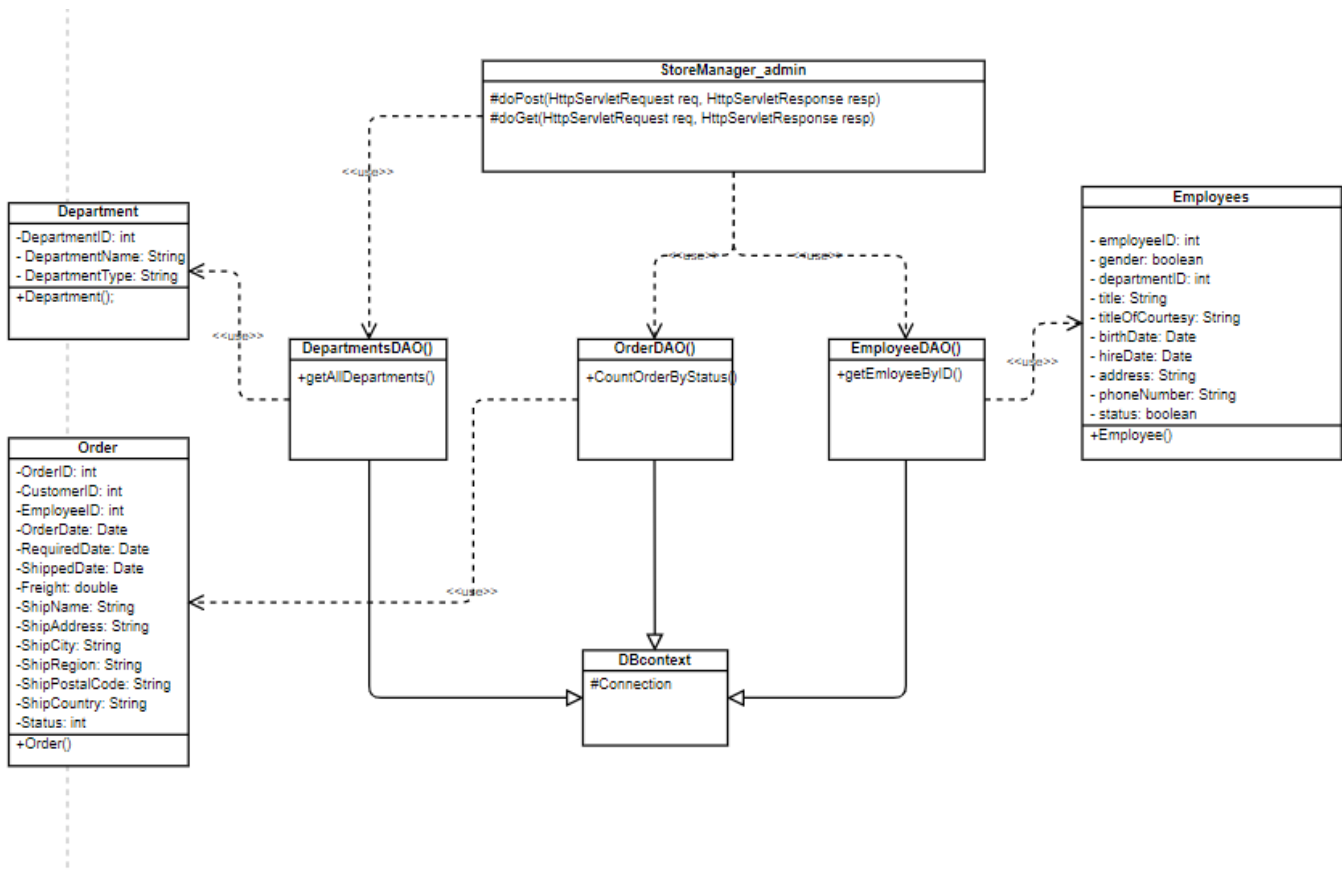
02	insertEmployee(Employee employee)	Add a new Employee to the data - Input: information Employee - Output: add employee into list employee
03	getNewEmployeeID()	Get maximum EmployeeID + 1 - Input: None - Output: maximum employeeID

c. Sequence Diagram(s)



3.6. View Store management

a.class diagram



b. Class Specifications

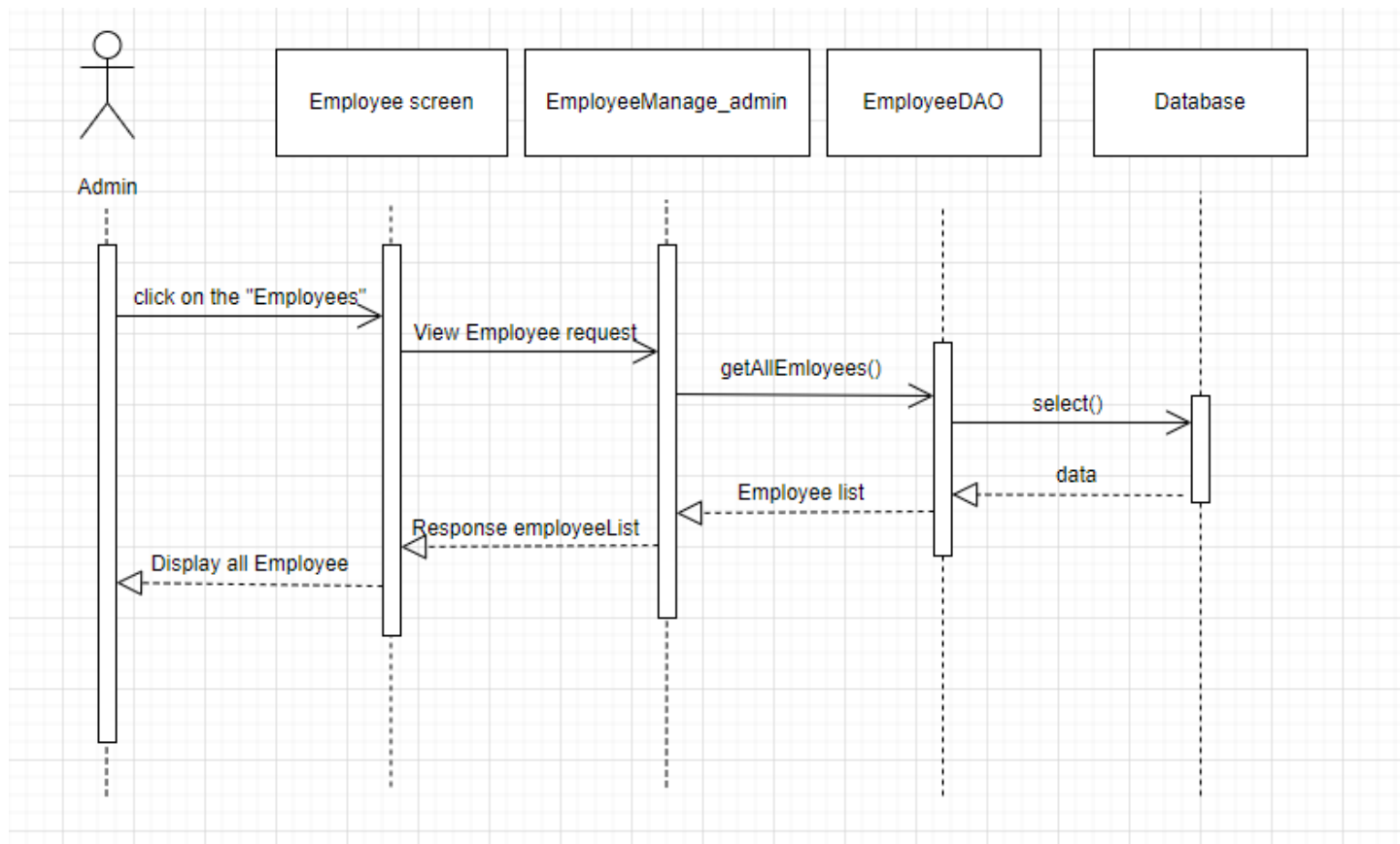
StoreManager_admin class

No	Method	Description
01	doGet	Called by the server (via the service method) to allow a servlet to handle a GET request.
02	doPost	is Called by the server (via the service method) to allow a servlet to handle a POST request.

EmployeeDAO

No	Method	Description
01	getEmployeeByID()	Get employee have ID input - Input:ID - Output:Employee

c.Sequence Diagram(s)



d.Database Queries

OrderDAO

No	Method	Description
01	CountOrderByStatus()	Get all order by order status - Input:status - Output:list order

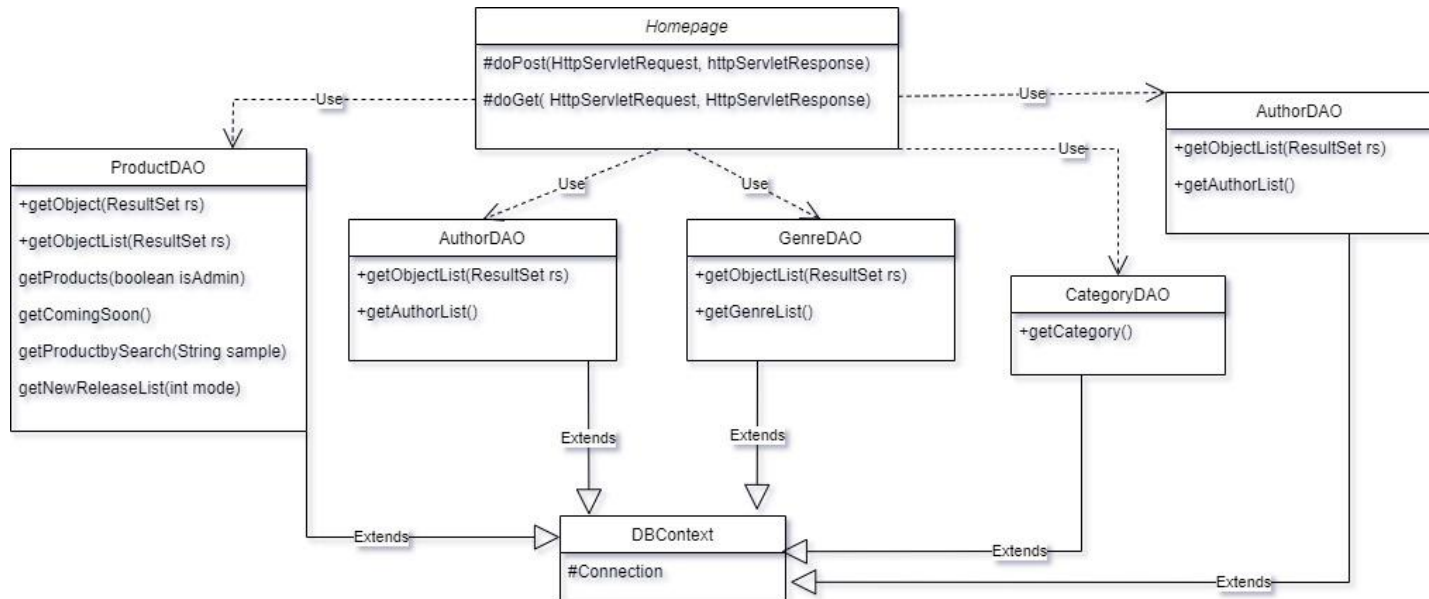
DepartmentDAO

No	Method	Description
01	DepartmentsDAO()	Get all department in department - Input:none - Output:list department

4. Common features

4.1 Homepage

a. Class Diagram



b. Class Specifications

Homepage Class

No	Method	Description
01	doGet	Called by the server (via the service method) to allow a servlet to handle a GET request.
02	doPost	Called by the server (via the service method) to allow a servlet to handle a POST request.

AuthorDAO Class

No	Method	Description
01	getObjectList	Get the object list by ResultSet pass from getAuthorList() - Input: list ResultSet - Output: list of Author
02	getAuthorList	Get the Author list - Input: None - Output: list of Author

CategoryDAO Class

No	Method	Description
01	getCategory	Get the Category list - Input: None - Output: list of Category

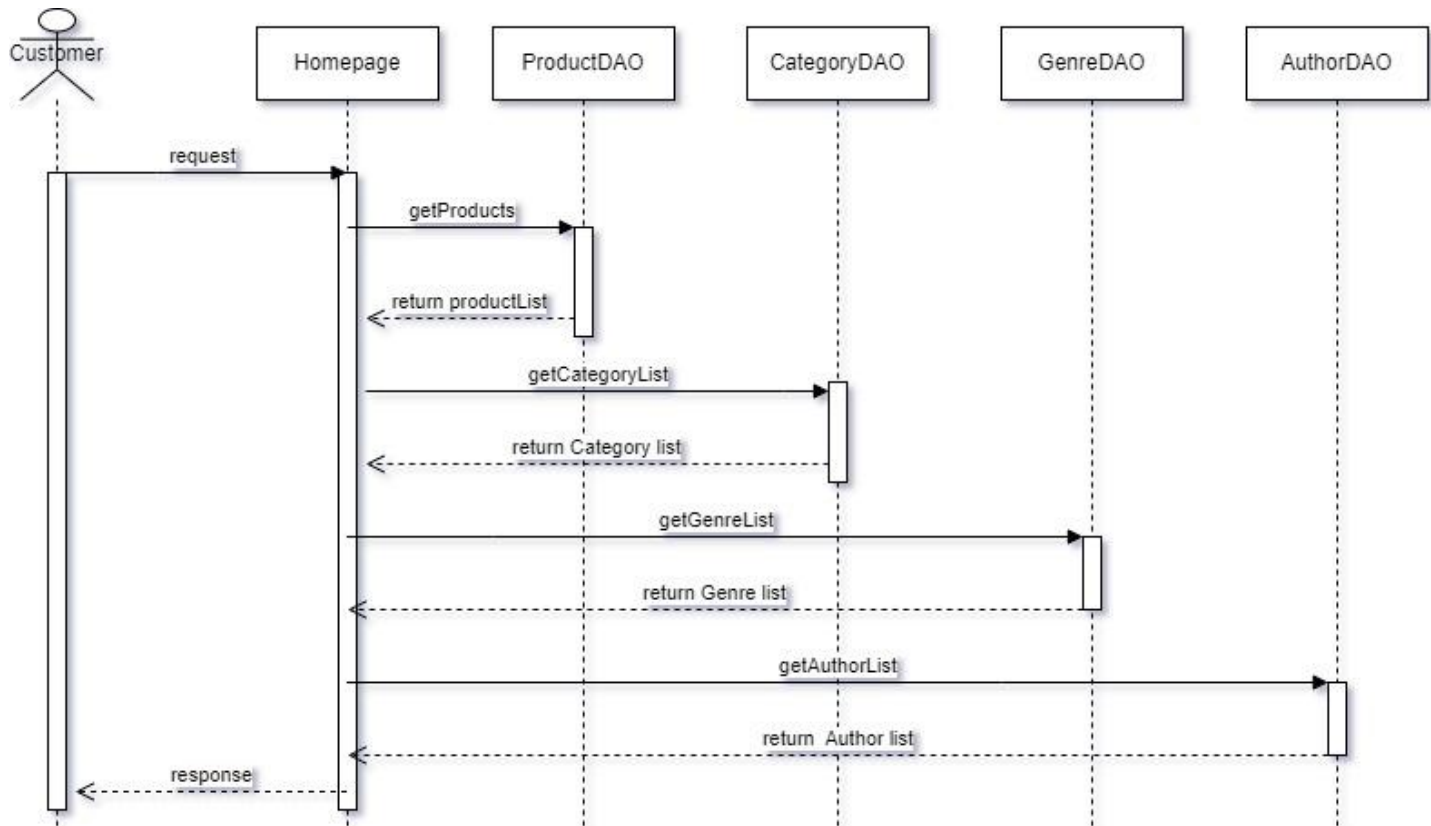
GenreDAO Class

No	Method	Description
01	getObjectList	Get the object list by ResultSet pass from getGenreList() - Input: list ResultSet - Output: list of Genre
02	getGenreList	Get the Genre list - Input: None - Output: list of Genre

ProductDAO Class

No	Method	Description
01	getObjectList	Get the object list by ResultSet - Input: list ResultSet - Output: list of Product
02	getObject	Get the object - Input: None - Output: an instance of Product
03	getProducts(boolean isAdmin)	Get products - Input: isAdmin = true: get all product in Database isAdmin = false: get product list is in business currently - Output: list of Product
04	getComingSoon()	Get products which do not release - Input: None - Output: a list of Product which does not release
05	getProductbySearch(String sample)	Get products which is similar with a pattern - Input: None - Output: a list of Product
06	getNewReleaseList()	Get products which do not release yet - Input: sample (pattern) which user want to find out - Output: a list of Product which does not release

c. Sequence Diagram(s)



d. Database Queries

ProductDAO Class

No	Method	Query
01	getProducts()	"select * from Products" "select * from Products where Discontinued=?"
02	getNewReleaseList()	"select TOP 3 * from Products WHERE PublishDate IS NOT NULL AND PublishDate < getdate() AND Discontinued = 0 Order by PublishDate DESC" "select * from Products WHERE PublishDate IS NOT NULL AND Discontinued = 0 Order by PublishDate DESC"
03	getComingSoon()	"select TOP 1 * from Products where PublishDate > getdate() order by PublishDate ASC"

CategoryDAO Class

No	Method	Query
01	getCategory()	SELECT * FROM Categories

GenreDAO Class

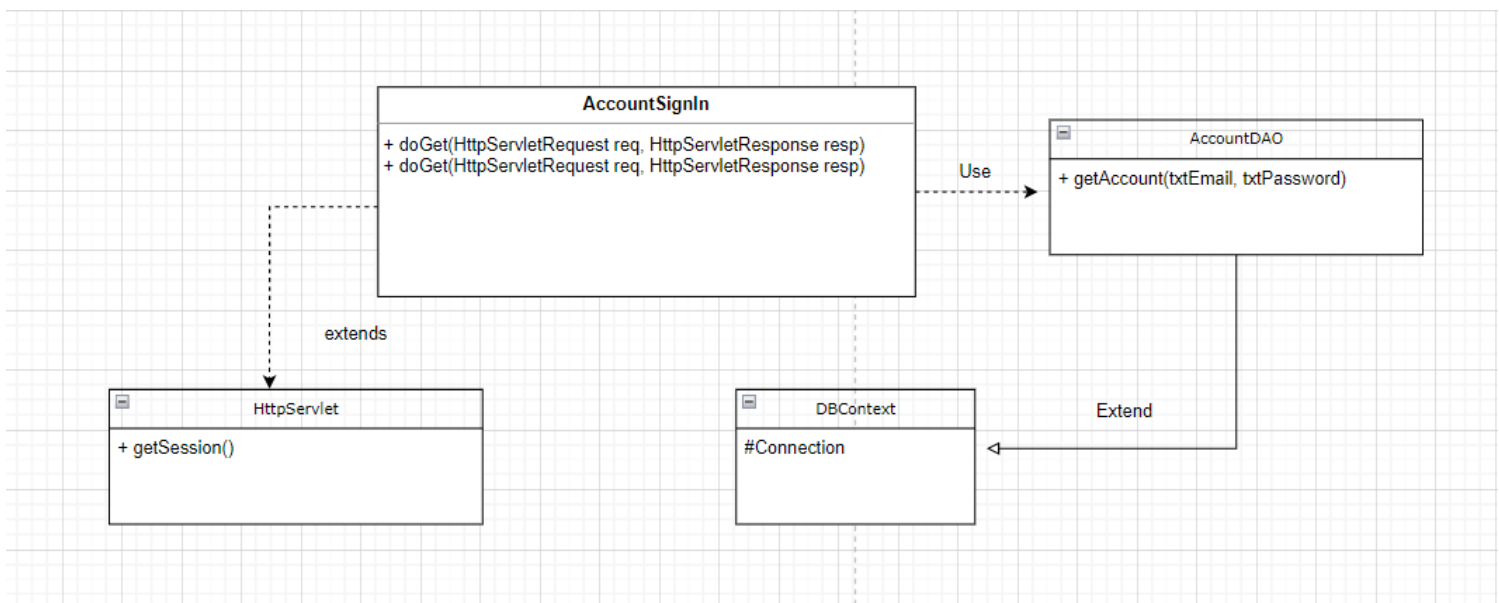
No	Method	Query
01	getGenreList()	select * from Genre

AuthorDAO Class

No	Method	Query
01	getAuthorList()	select * from Authors

4.2 Sign in

a. Class Diagram



b. Class Specifications

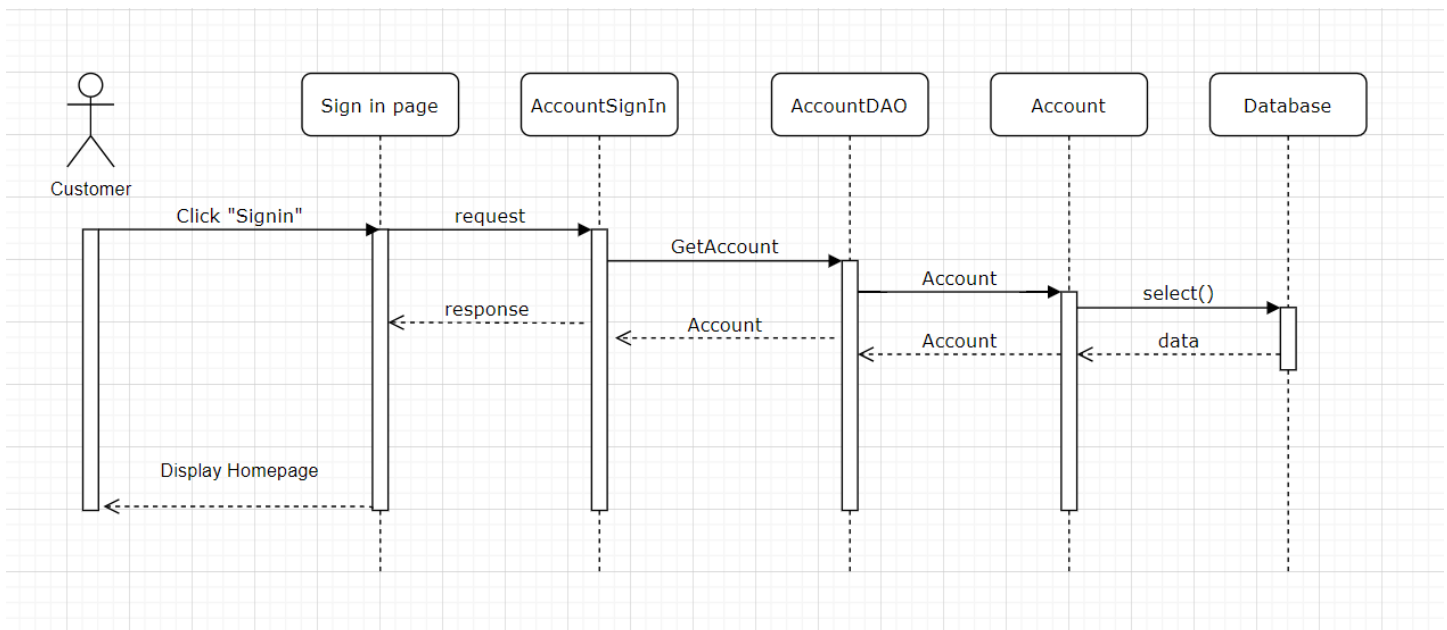
AccountSignIn Class

No	Method	Description
01	doGet	Called by the server (via the service method) to allow a servlet to handle a GET request.
02	doPost	is Called by the server (via the service method) to allow a servlet to handle a POST request.

AccountDAO Class

No	Method	Description
01	getAccount(String email, String pass)	Get the account object by ResultSet - Input: account object ResultSet - Output: Account object

c. Sequence Diagram(s)



d. Database Queries

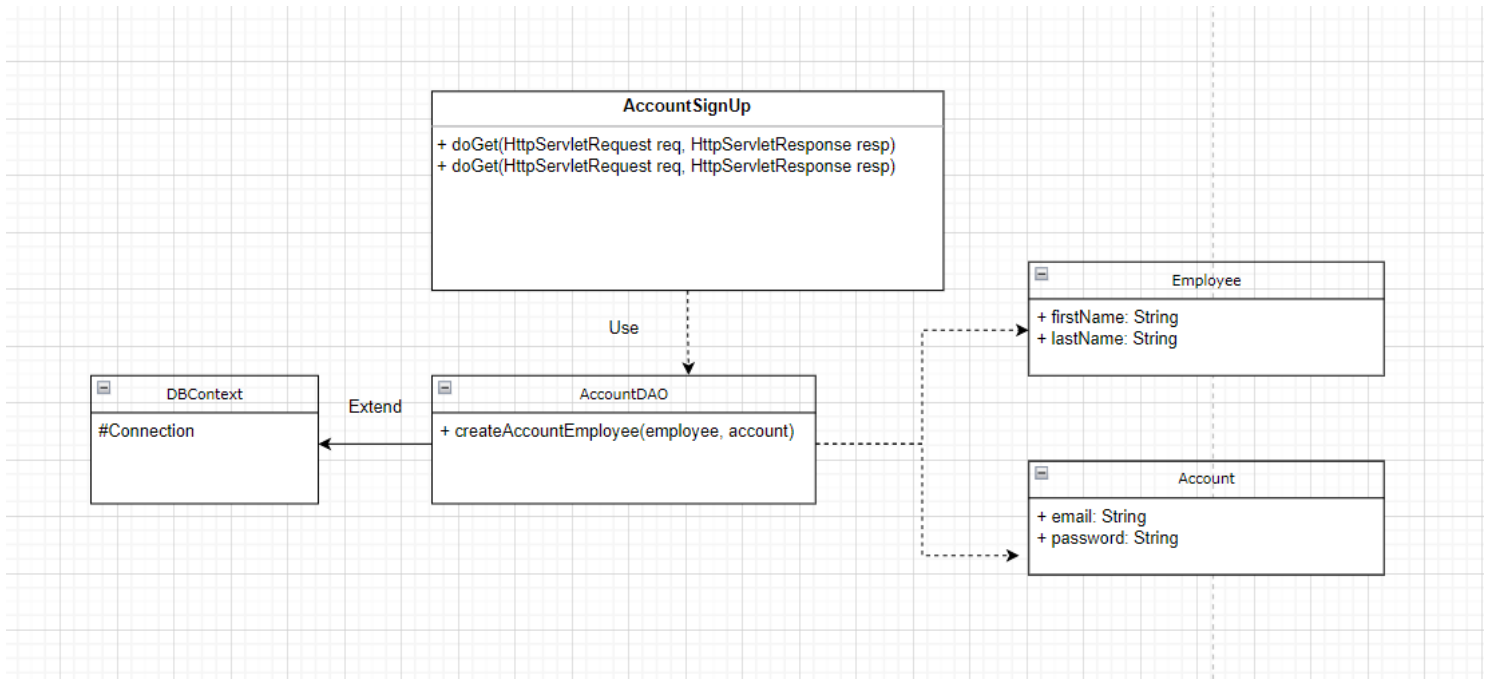
AccountDAO Class

No	Method	Query
----	--------	-------

01	getAccount (String email, String pass)	select * from Accounts where Email=? and Password=?
----	---	---

4.3 Sign up

a. Class Diagram



b. Class Specifications

AccountSignIn Class

No	Method	Description
01	doGet	Called by the server (via the service method) to allow a servlet to handle a GET request.
02	doPost	Called by the server (via the service method) to allow a servlet to handle a POST request.

AccountDAO Class

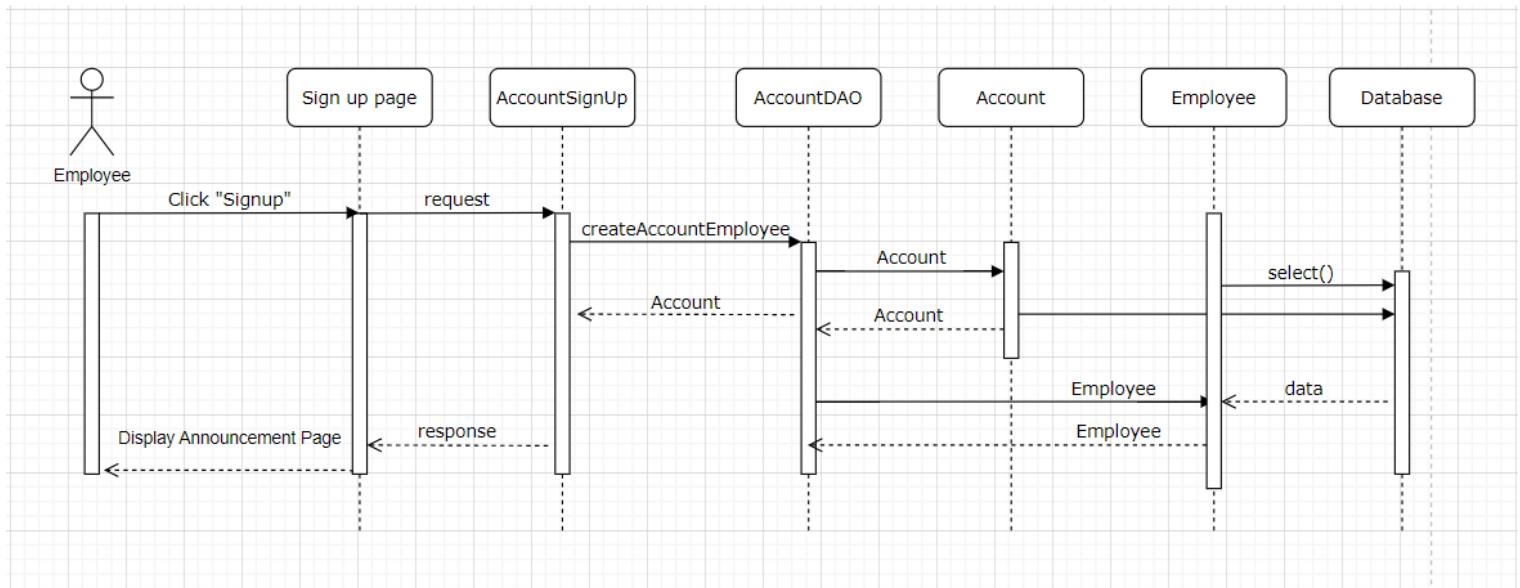
No	Method	Description
01	createAccountEmployee(employee, account)	Catch requests from users to create a new account. - Input: The information that is needed to create a new account. - Output: Announcement for users
02	createNewEmpID()	Get the last EmployeeID and create a newest

03

createNewAccID()

Get the last AccountID and create a newest

c. Sequence Diagram(s)



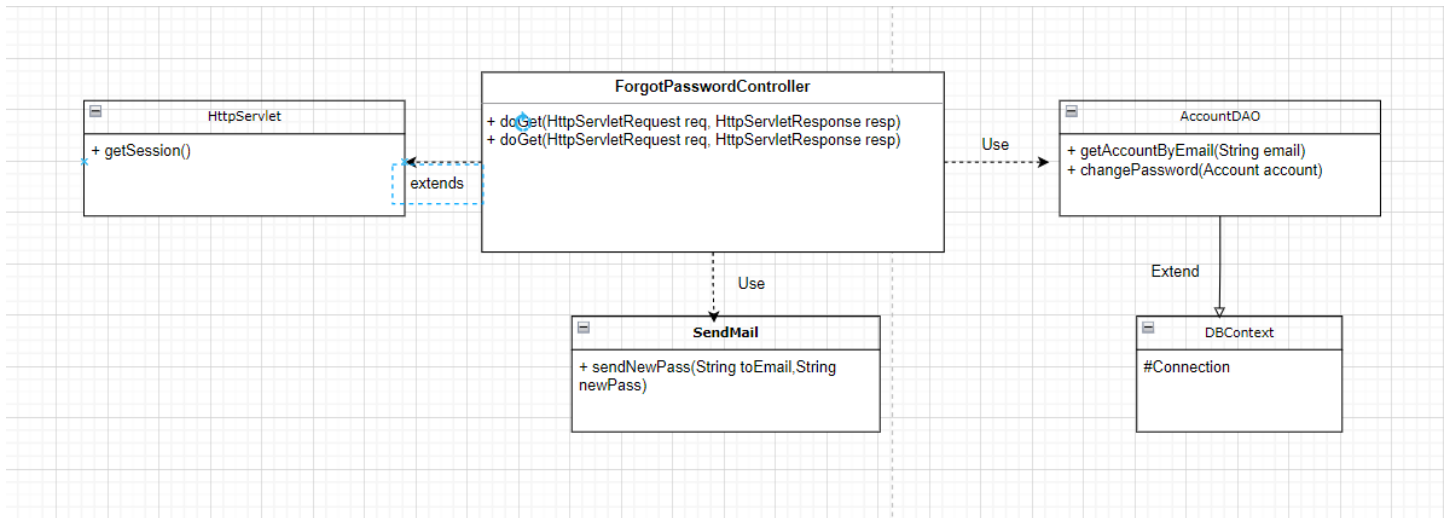
d. Database Queries

AccountDAO Class

No	Method	Query
01	createAccountEmployee(Employee emp, Account acc)	exec createAccountEmp ?, ?, ?, ?, ?, ?;
02	createNewEmpID()	select Max(Employees.EmployeeID) as 'MaxEmpID' from Employees
03	createNewAccID()	select Max(Accounts.AccountID) as 'MaxAccID' from Accounts

4.4 Forgot Password

a. Class Diagram



b. Class Specifications

ForgotPasswordController Class

No	Method	Description
01	doGet	Called by the server (via the service method) to allow a servlet to handle a GET request.
02	doPost	Called by the server (via the service method) to allow a servlet to handle a POST request.

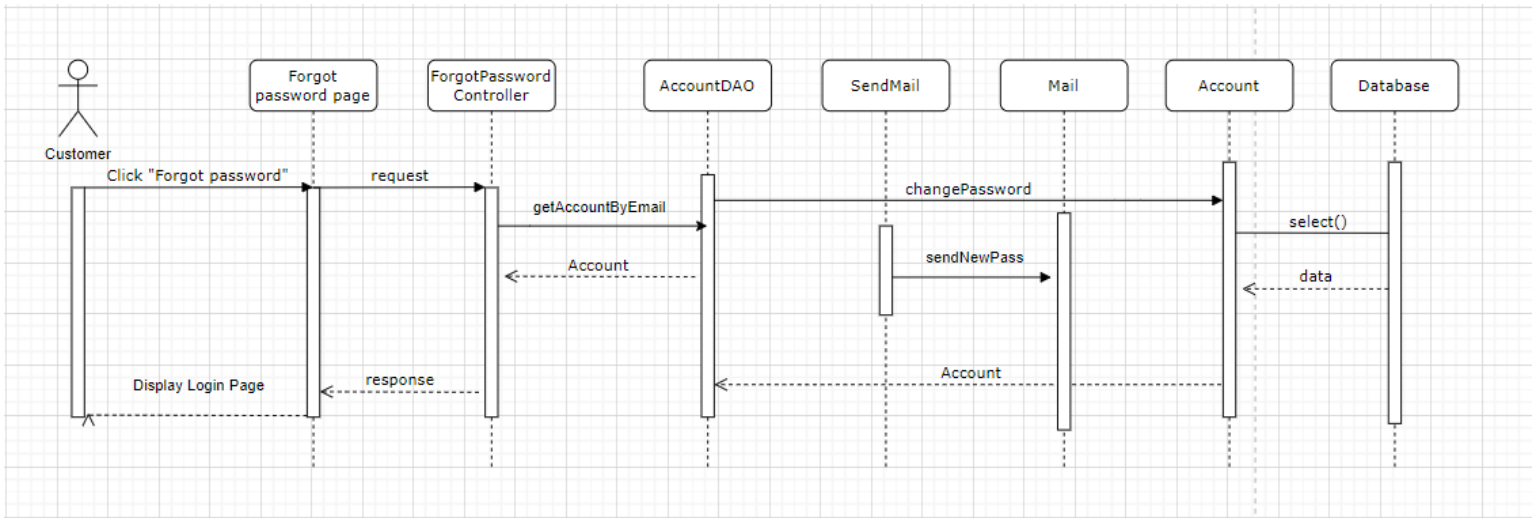
AccountDAO Class

No	Method	Description
01	getAccountByEmail(String email)	Get the account object by ResultSet - Input: email of account object ResultSet - Output: Account object
02	changePassword(Account account)	Change password of account - Input: account object ResultSet - Output: none

SendMail Class

No	Method	Description
01	sendNewPass(String toEmail, String newPass)	Send password change notification via email - Input: email & new random password - Output: none

c. Sequence Diagram(s)



d. Database Queries

AccountDAO Class

No	Method	Query
01	getAccountByEmail(String email)	select * from Accounts where Email=?
02	changePassword(Account account)	update Accounts set Password =? where Email=?