Fpt University

TRƯỜNG ĐẠI HỌC FPT

Computer Sale

Software Design Specification

–Summer 2024 –

Record of changes

Date	A* M, D	In charge	Change Description
11 thg 5, 2024	M	Tô Việt Hoàng	Code Packages
11 thg 5, 2024	A	Vũ Đặng Quang Vinh	Database Schema_Database Schema
12 thg 5, 2024	M	Vũ Đặng Quang Vinh	Database Schema
12 thg 5, 2024	A	Vũ Đặng Quang Vinh	Table Description
22 thg 7, 2024	A	Trần Khánh Linh	Code Designs
26 thg 7, 2024	M	Trần Khánh Linh	OverViews,Code Designs,
27 Jul 2024	M	Vũ Đặng Quang Vinh	
27 Jul 2024	M	Tô Việt Hoàng	

*A - Added
M - Modified
D - Deleted

Table of Contents

Document convention

Basic Content Font:

- **Font:** Calibri
- **Size:** 12pt

Heading Font

- **Font:** Roboto
- **Sizes:**
    - Heading 1 (H1): 30pt
    - Heading 2 (H2): 24pt
    - Heading 3 (H3): 20pt
    - Heading 4 (H4): 18pt
    - Heading 5 (H5): 16pt
    - Heading 6 (H6): 14pt
- Align
    - Heading 1 (H1): 0
    - Heading 2 (H2): 0
    - Heading 3 (H3): 0
    - Heading 4 (H4): 1
    - Heading 5 (H5): 1.5

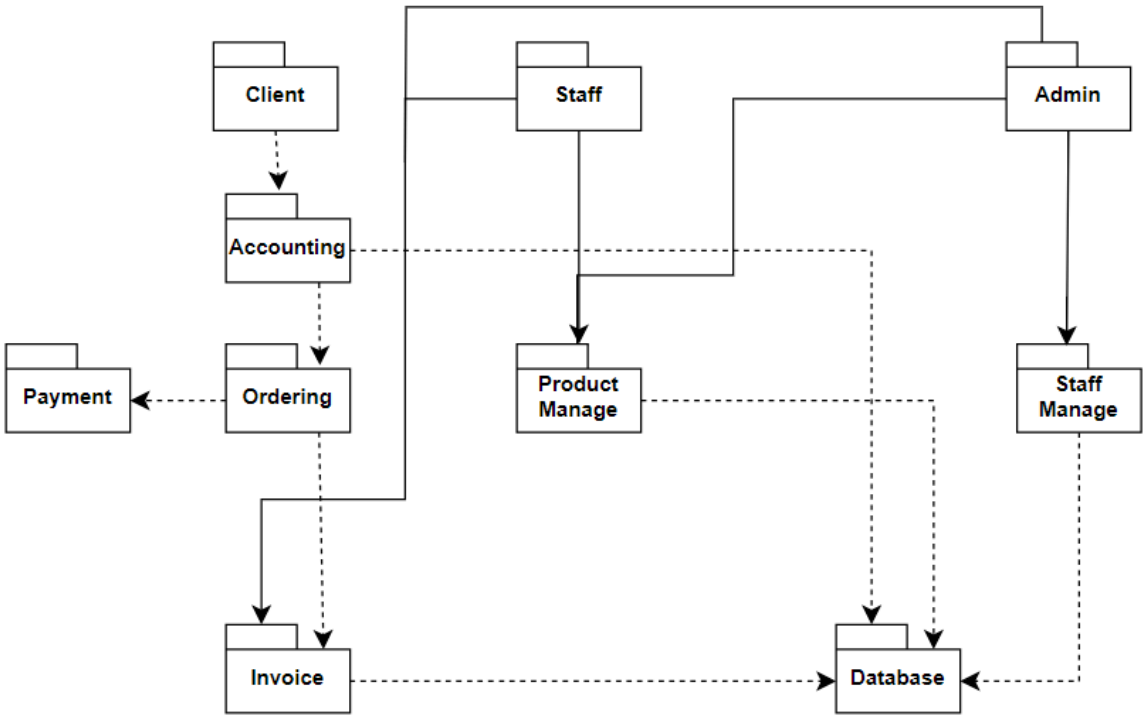
Heading 6 (H6):

I. Overview

1. Code Packages

1.1: Diagram

UML

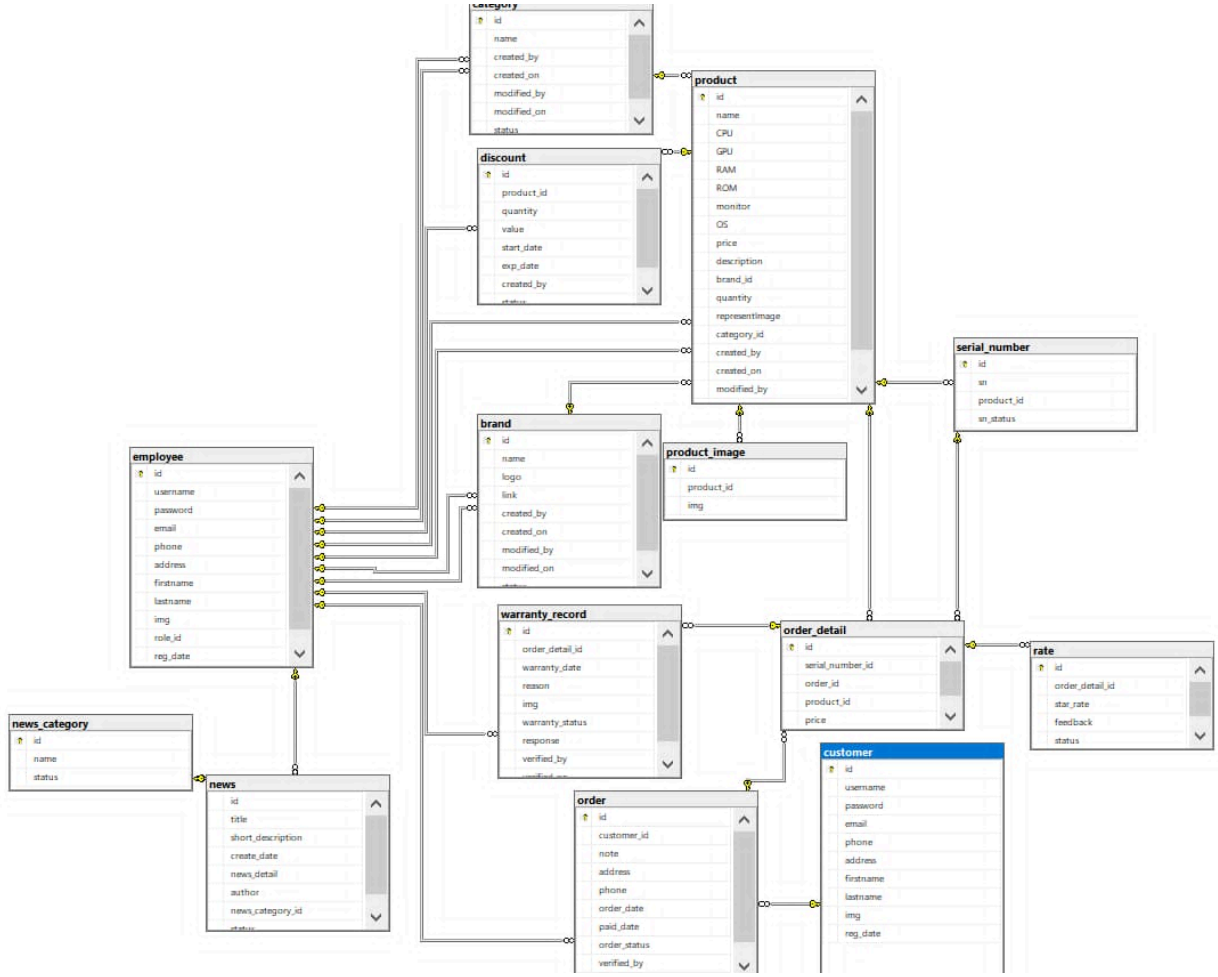


1.2 Package descriptions

No	Package	Description
1.	Admin	The highest role of manage system
2.	Staff	The Employees who manage the detail of products
3.	Client	The customer, buyer, viewer
4.	Accounting	About login or register account Login
5.	Ordering	A product or a list of products which are ordered
6.	Payment	The way to pay for invoice
7.	Invoice	Including information of products are bought and the detail of warranty
8.	Product Manage	Can CRUD the product
9.	Staff Manage	Can CRUD the staff
10.	Database	Saving system data

2. Database Design

2.1. Database Schema



2.2. Table Description

No	Table	Description		
1.	customer	id username password email phone address firstname lastname img reg_date	int identity(1,1) varchar(50) varchar(50) varchar(100) varchar(20) nvarchar(100) nvarchar(50) nvarchar(50) nvarchar(100) date	primary key
2.	employee	id username password email phone address firstname lastname img role_id reg_date status	int identity(1,1) varchar(50) varchar(50) varchar(100) varchar(20) nvarchar(100) nvarchar(50) nvarchar(50) nvarchar(100) int date tinyint	primary key
3.	category	id name created_by created_on modified_by modified_on status	int identity(1,1) nvarchar(255) int date int date tinyint	primary key reference key reference key
4.	brand	id name logo link created_by created_on modified_by modified_on status	int identity(1,1) nvarchar(255) varchar(100) varchar(100) int date int date int	primary key reference key reference key
5.	product	id name CPU GPU	int identity(1,1) nvarchar(255) varchar(100) varchar(100)	primary key

		RAM ROM monitor OS price [description] brand_id quantity img category_id created_by created_on modified_by modified_on status	varchar(100) varchar(100) varchar(100) varchar(100) int ntext int int varchar(100) int int date int date int	 reference key reference key reference key reference key
6.	product_image	id product_id img	int int varchar(100)	primary key reference key
7.	discount	id product_id quantity value exp_date created_by created_on status	int identity(1,1) int int tinyint date int date int	primary key reference key reference key
8.	serial_number	id sn product_id sn_status	int identity(1,1) varchar(100) int tinyint	primary key reference key
9.	[order]	id customer_id note [address] phone order_date paid_date order_status verified_by verified_on	int identity(1,1) int nvarchar(255) nvarchar(255) varchar(20) date date tinyint int date	primary key reference key reference key
10.	order_detail	id serial_number_id order_id product_id price	int identity(1,1) int int int int	primary key reference key reference key reference key

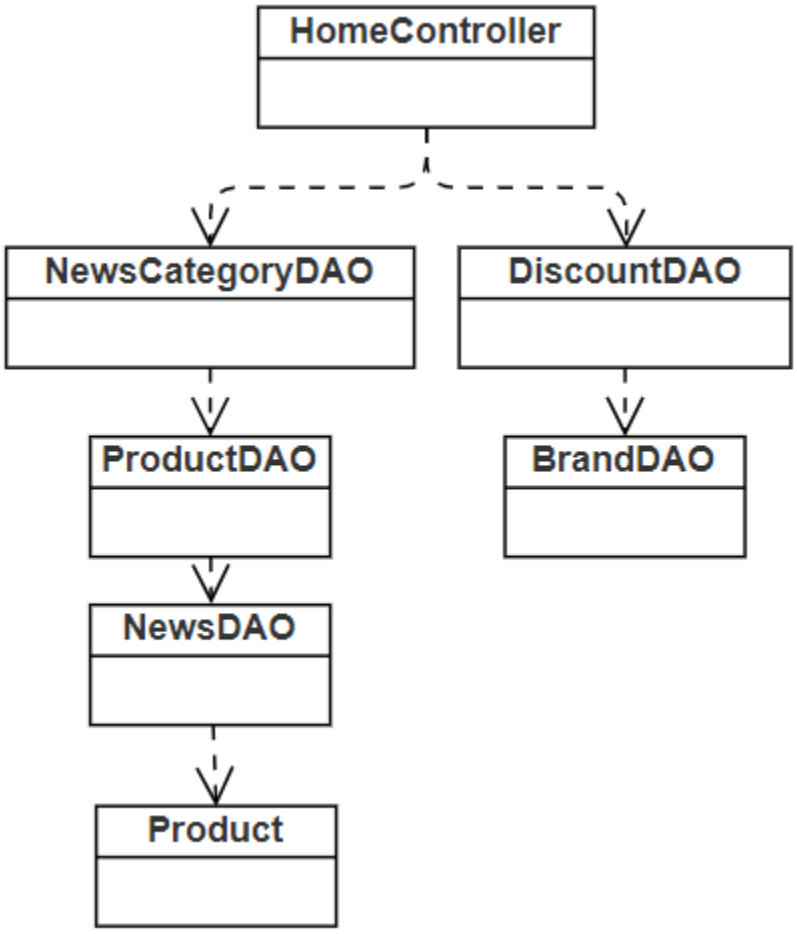
11.	rate	id order_detail_id star_rate feedback status	int identity(1,1) int tinyint nvarchar(255) int	primary key reference key
12.	warranty_record	id order_detail_id warranty_date reason img warranty_status response verified_by verified_on	int identity(1,1) int date nvarchar(255) varchar(255) tinyint nvarchar(255) int date	primary key reference key reference key
13.	news_category	id [name] status	int identity(1,1) varchar(100) int	primary key
14.	news	id title short_description create_date news_detail author news_category_id status	int identity(1,1) nvarchar(100) nvarchar(255) date text int int int	 reference key reference key

II. Code Designs

1. Guest

1.1 Home

a. Class Diagram



b. Class Specifications

AddNews

No	Method	Description
	doGet(HttpServletRequest request, HttpServletResponse response)	Handles HTTP GET requests; fetches news categories and existing news for editing, then forwards to the `Home.jsp` view.

	doPost(HttpServletRequest request, HttpServletResponse response)	Handles HTTP POST requests; processes form submissions for adding, updating, or deleting news based on the provided action.

CategoryDAO

No	Method	Description
1	getAll()	Retrieves all Category categories.

DiscountDAO

No	Method	Description
1	getDiscountProductID()	Retrieves a map of discount IDs and their corresponding values for active discounts.

BrandDAO

No	Method	Description
1	getAll()	Retrieves all Brand categories.

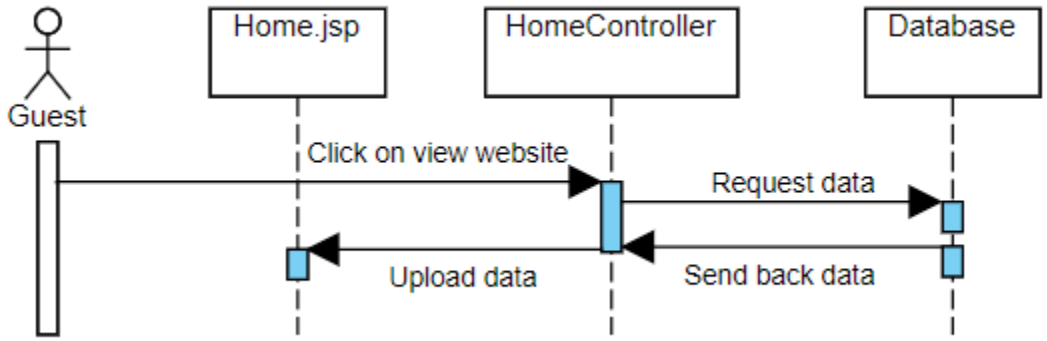
ProductDAO

No	Method	Description
1	getHomeProduct()	Retrieves a list of products that have the top 8 highest discount values and are currently active.

NewsDAO

No	Method	Description
1	getHomeNews()	Retrieves a list of the top 3 news articles in a specific category (category ID 1), ordered by their creation date.

c. Sequence Diagram(s)

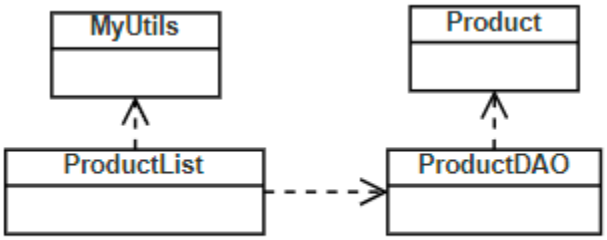


d. Database Queries

- getAll()
  - select * from category
- getDiscountProductID()
  - SELECT * FROM discount where (start_date < ? or start_date = ?) and exp_date > ?
- getAll()
  - select * from brand
- getHomeProduct()
  - SELECT * FROM product \n"
  - + "WHERE id IN (\n"
  - + "SELECT TOP 8 product_id FROM discount\n"
  - + "WHERE (start_date < ? OR start_date= ?) AND exp_date > ?\n"
  - + "GROUP BY product_id,value,start_date\n"
  - + "ORDER BY [value])
- getHomeNews()
  - SELECT top 3 * FROM news where news_category_id =1 order by create_date

1.2 Product List

a. Class Diagram



b. Class Specifications

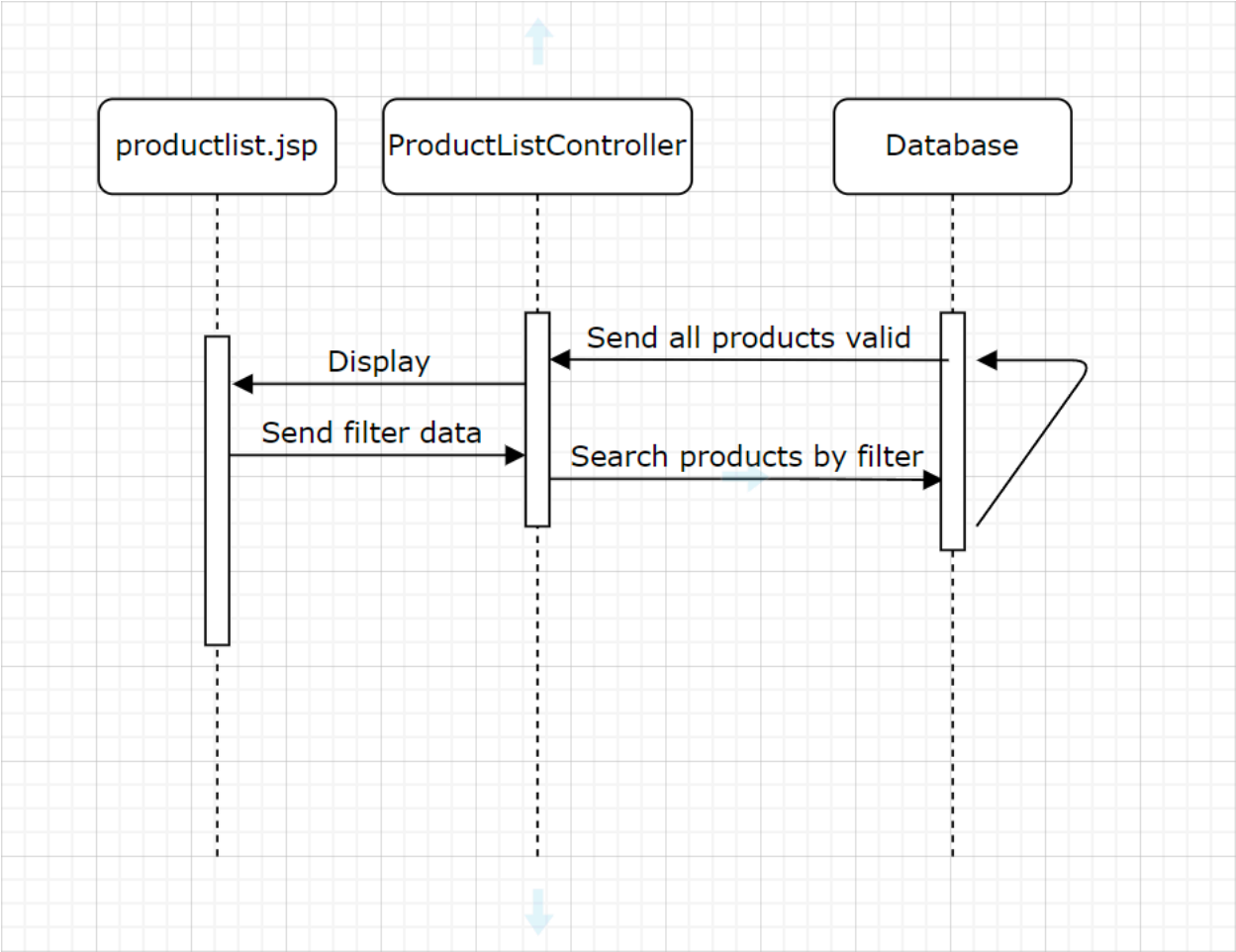
ProductDAO

No	Method	Description
1	getAvailable(String extend)	Retrieves a list of available products with an optional filter condition.
2	getAllByFilter(String filter)	Retrieves a list of available products filtered by the given condition.

MyUtils

No	Method	Description
1	getArrayListByPaging(ArrayList<T> list, int pageNumber, int itemsPerPage)	calculates the start and end indices for the desired page and returns the corresponding sublist.

c. Sequence Diagram(s)

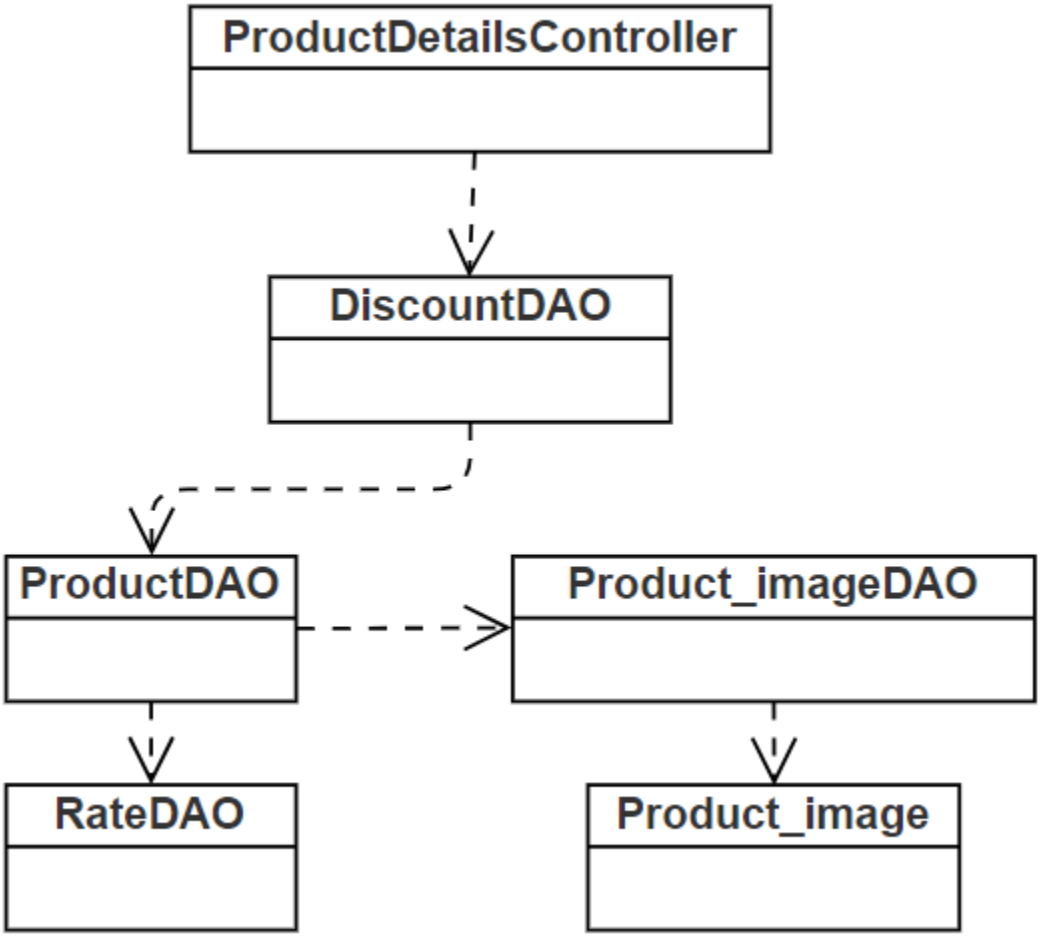


d. Database Queries

- getAvailable(String extend)
    - select * from product where status=1
-

1.3 Product Details

a. Class Diagram



b. Class Specifications

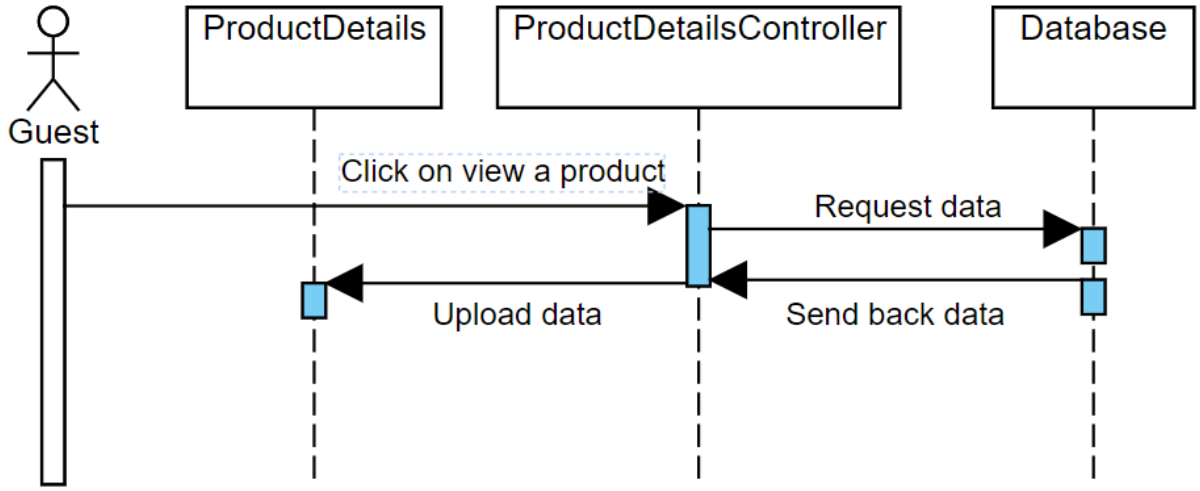
ProductDAO

No	Method	Description
1	getById(int id)	Retrieves a product by its ID.
2	getRelatedProducts(int id)	Retrieves a list of up to 8 products related to the product with the specified ID, based on the same brand.

DiscountDAO

No	Method	Description
1	getArrayListByPaging(ArrayList<T> list, int pageNumber, int itemsPerPage)	calculates the start and end indices for the desired page and returns the corresponding sublist.

c. Sequence Diagram(s)

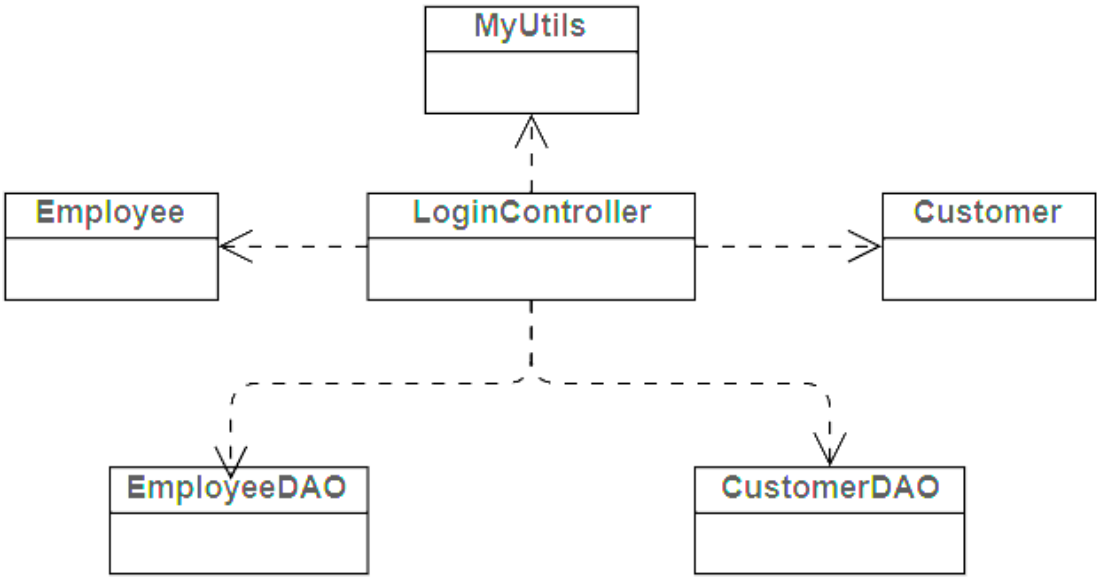


d. Database Queries

- getById(int id)
  - select * from product where id =
- getRelatedProducts(int id)
  -
  - String sql = "SELECT top 8 p.* \n"
  - + "FROM product p\n"
  - + "JOIN product p1 ON p.brand_id = p1.brand_id\n"
  - + "WHERE p1.id = ?;

1.5 Login

a. Class Diagram



b. Class Specifications

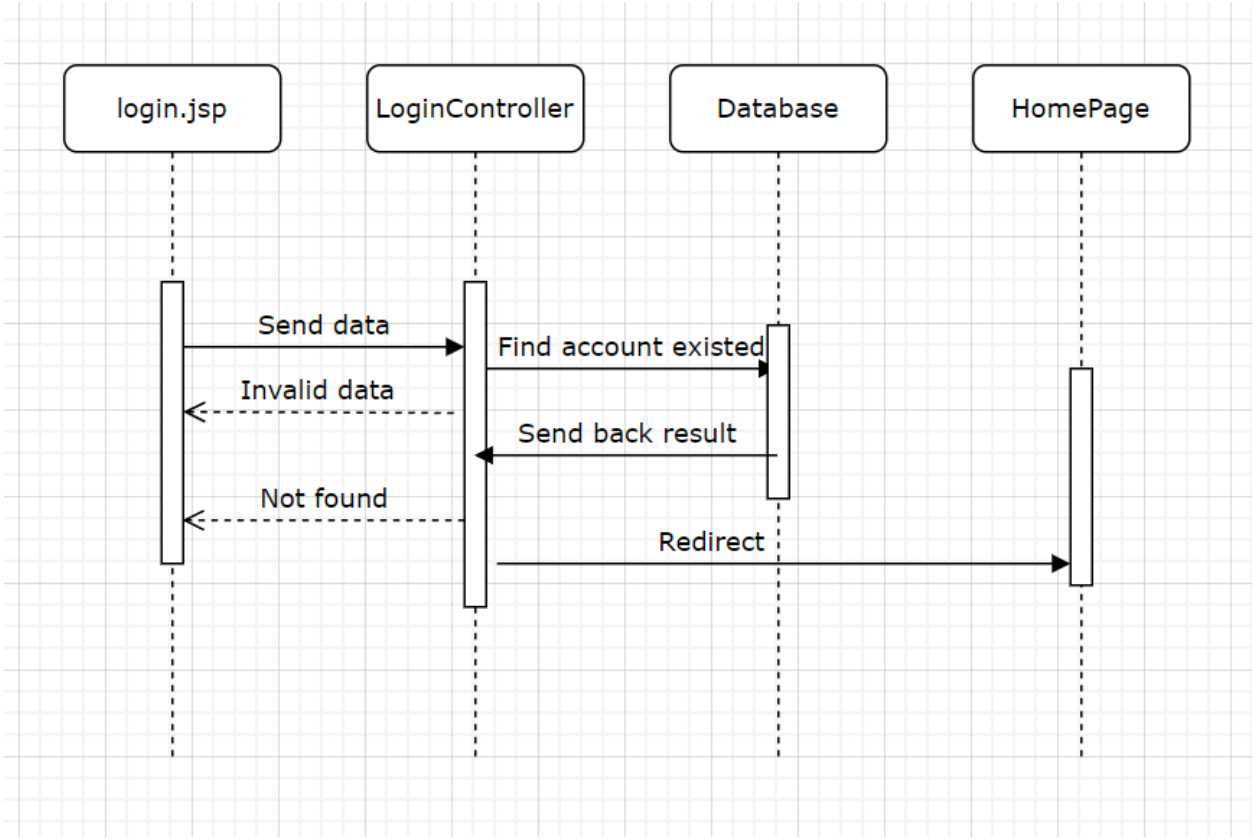
CustomerDAO

No	Method	Description
1	checkLogin(String username, String password)	Checks if a customer with the provided username and password exists in the database and returns the corresponding `Customer` object if found.

EmployeeDAO

No	Method	Description
1	checkLogin(String username, String password)	Checks if a customer with the provided username and password exists in the database and returns the corresponding `Employee` object if found.

c. Sequence Diagram(s)

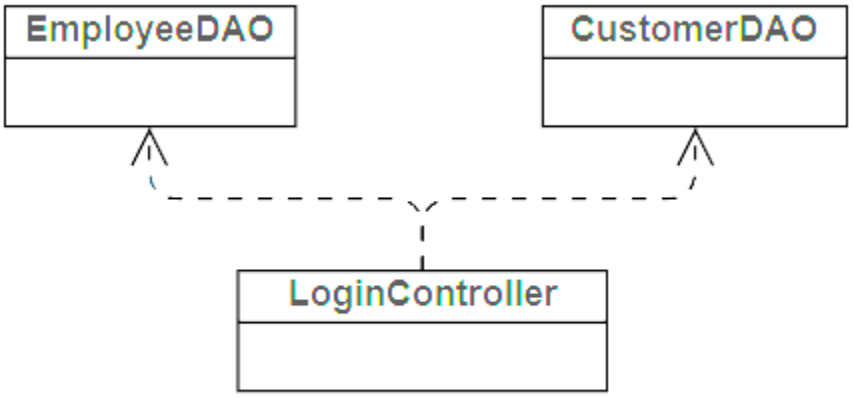


d. Database Queries

- checkLogin(String username, String password)
    - select * from customer where username like '" + username + "' and password like '" + MyUtils.getMd5(password) + "'
- checkLogin(String username, String password)
    - select * from employee where username like '" + username + "' and password like '" + MyUtils.getMd5(password) + "'

1.6 Signup

a. Class Diagram



b. Class Specifications

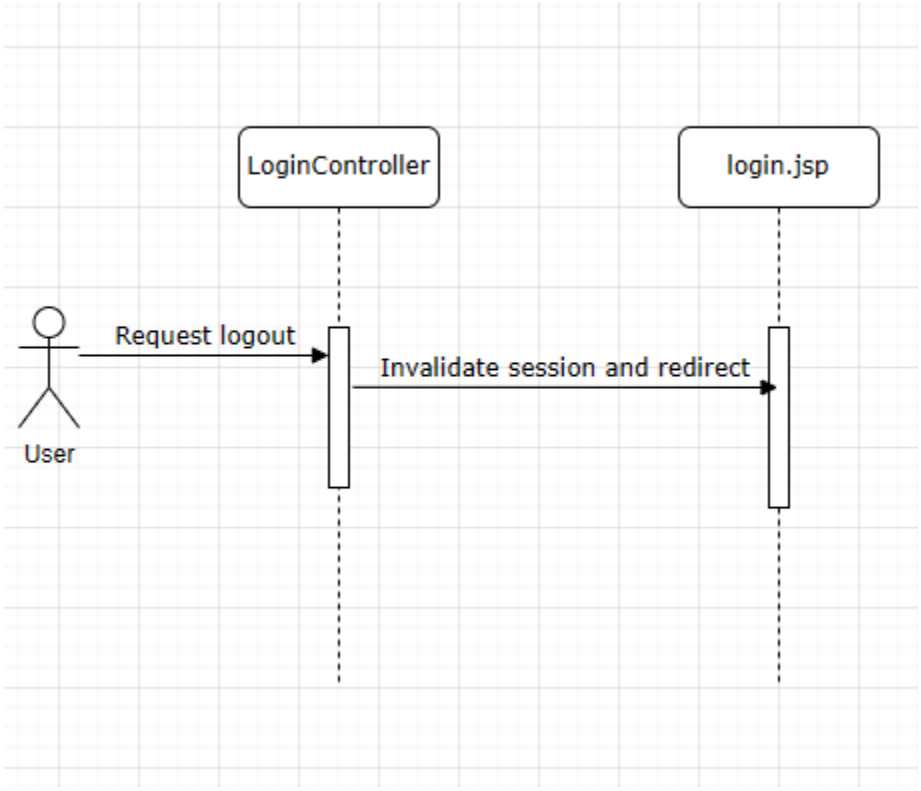
CustomerDAO

No	Method	Description
1	checkSignUp(String username, String email)	Checks if a username or email is already in use by another customer

EmployeeDAO

No	Method	Description
1	checkSignUp(String username, String email)	Checks if a username or email is already in use by another customer

c. Sequence Diagram(s)

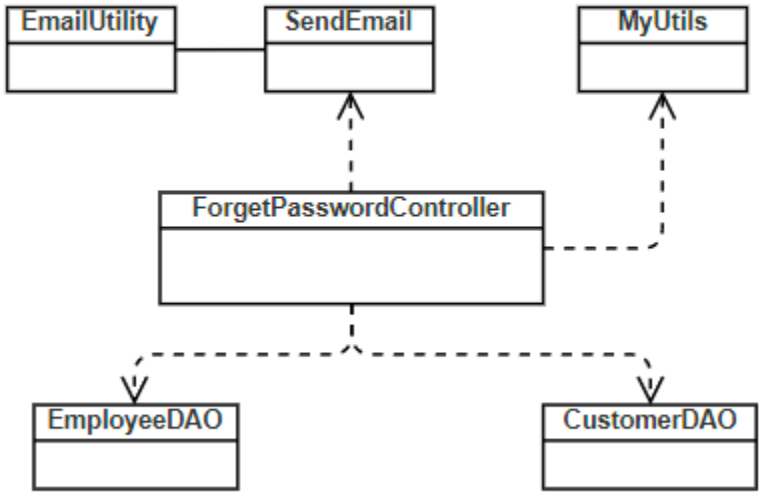


d. Database Queries

- checkSignUp(String username, String email)
    - select * from customer where username like '" + username + "' or email like '" + email + "'
- checkSignUp(String username, String email)
    - select * from employee where username like '" + username + "' or email like '" + email + "'

1.7 Forgot password

a. Class Diagram



b. Class Specifications

CustomerDAO

No	Method	Description
1	getByEmail(email)	Checks if a username or email is already in use by another customer
2	changePassword(int id, String newPassword)	Updates the password for a customer with the specified ID

EmployeeDAO

No	Method	Description
1	getByEmail(email)	Checks if a username or email is already in use by another customer
2	changePassword(int id, String newPassword)	Updates the password for a employee with the specified ID

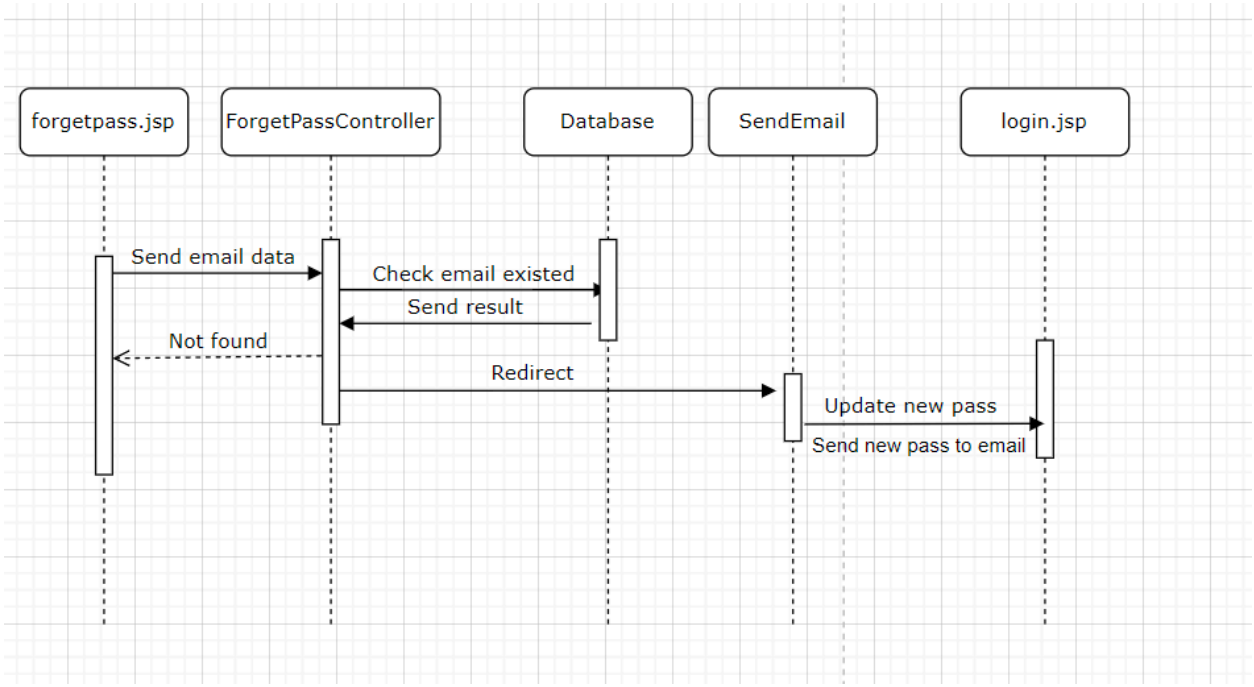
MyUtils

No	Method	Description
1	genCode(length)	Gen a random string which have the length given

EmailUtility

No	Method	Description
1	sendMail(to,subject,body,content)	Using library to send an email

c. Sequence Diagram(s)



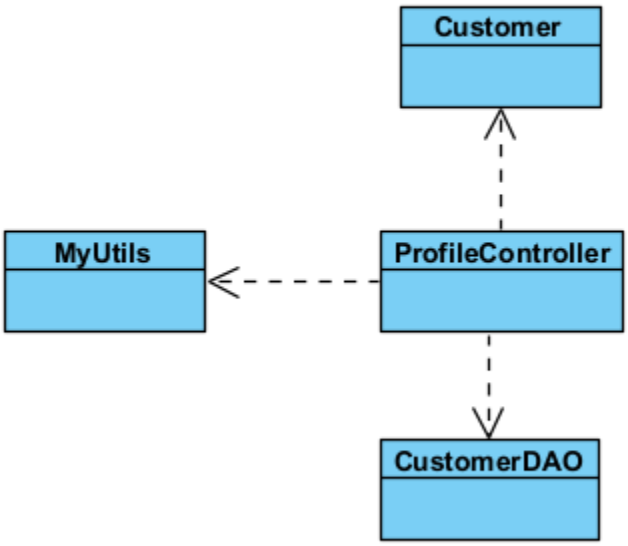
d. Database Queries

- getByEmail(email)
    - elect * from customer where email like '" + email + "'
- changePassword(int id, String newPassword)
    - UPDATE [dbo].[customer] SET [password] = ? WHERE [id] = ?

2. Customer

2.1 Profile

a. Class Diagram



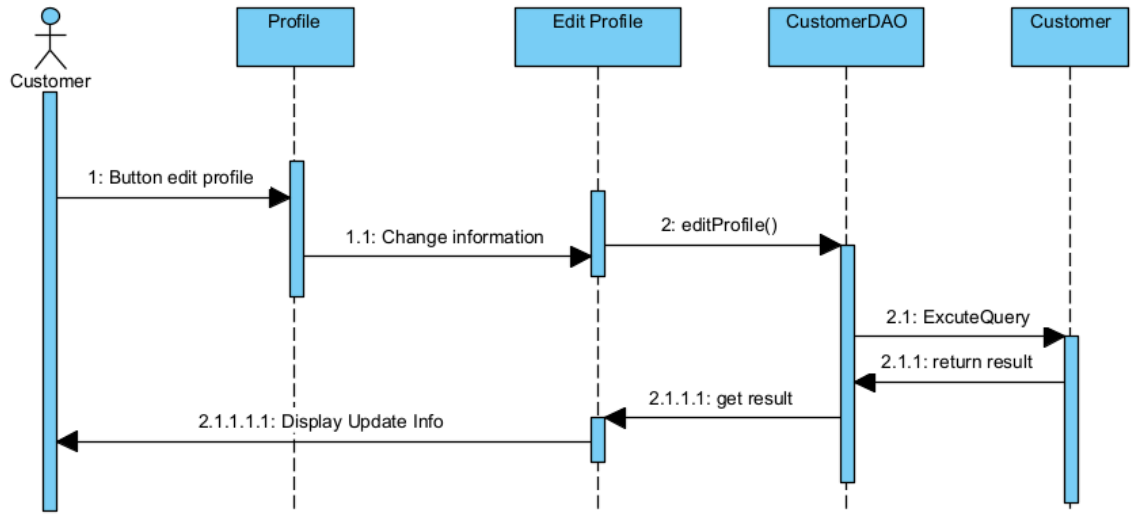
b. Class Specifications

CustomerDAO

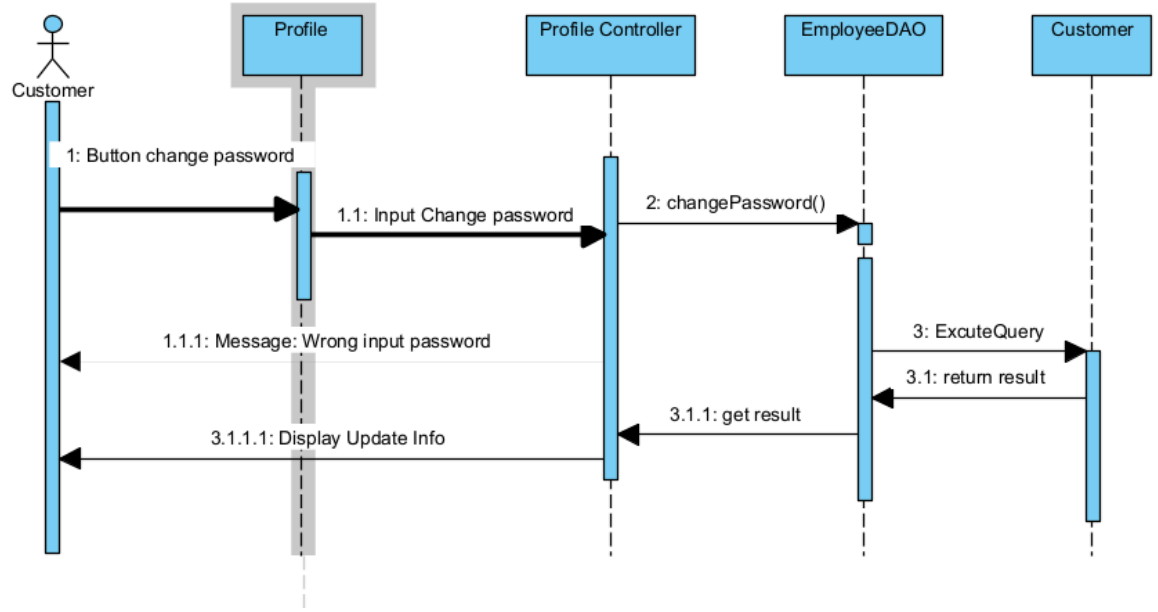
No	Method	Description
1	editProfile()	Checks if a username or email is already in use by another customer
2	checkPassWord(String oldPassword, int id)	Check the password of the selected id

c. Sequence Diagram(s)

Edit profile



Change password



d. Database Queries

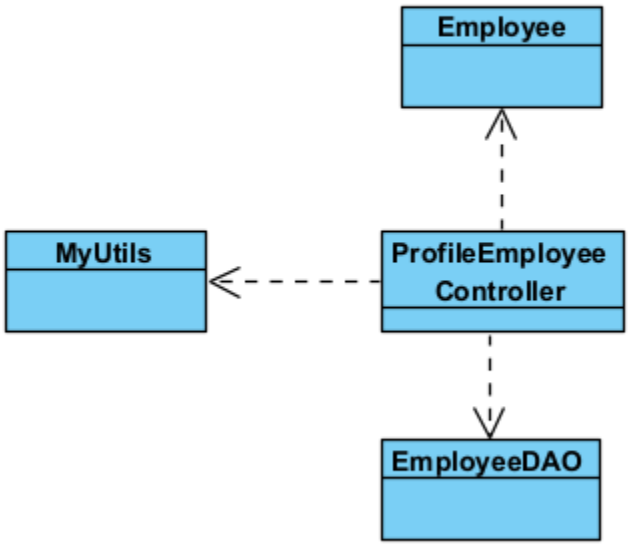
- editProfile()
  - UPDATE [dbo].[customer]\n"
  - + " SET [email] = ?\n"
  - + " ,[phone] = ?\n"
  - + " ,[address] = ?\n"
  - + " ,[firstname] = ?\n"
  - + " ,[lastname] = ?\n"
  - + " ,[img] = ?\n"
  - + " WHERE [id] = ?
- checkPassWord(String oldPassword, int id)

- SELECT password FROM [dbo].[customer] WHERE id = ?

3. Employee

3.0 Employee Profile

a. Class Diagram



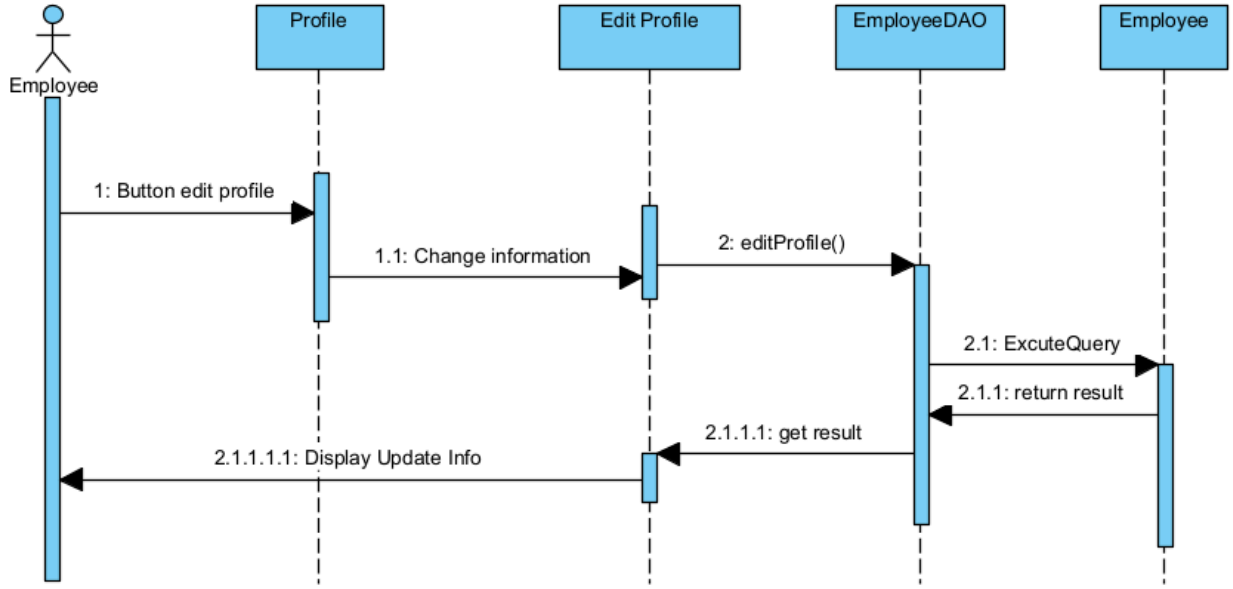
b. Class Specifications

EmployeeDAO

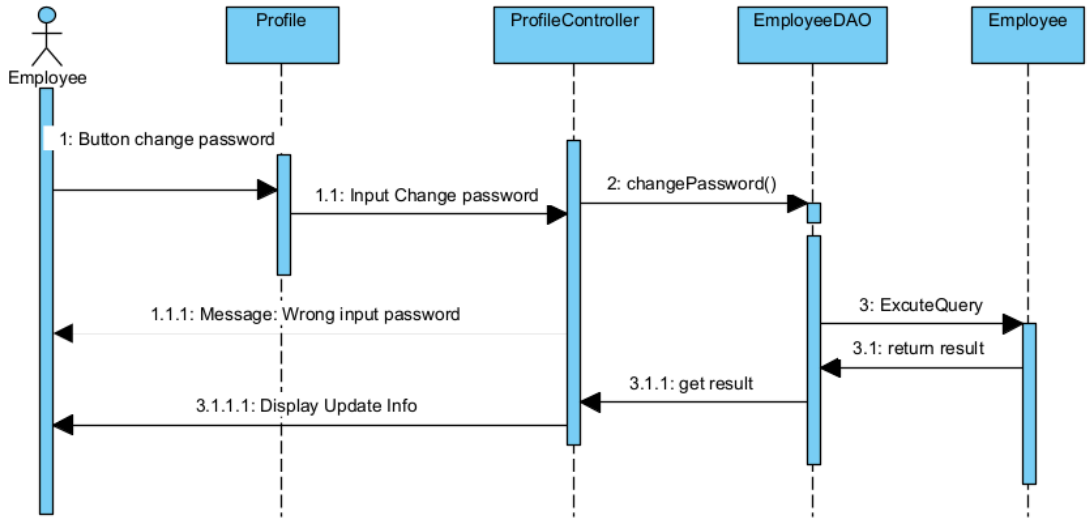
No	Method	Description
1	editProfile()	Checks if a username or email is already in use by another customer
2	checkPassWord(String oldPassword, int id)	Check the password of the selected id

c. Sequence Diagram(s)

Edit Profile



Change Password



d. Database Queries

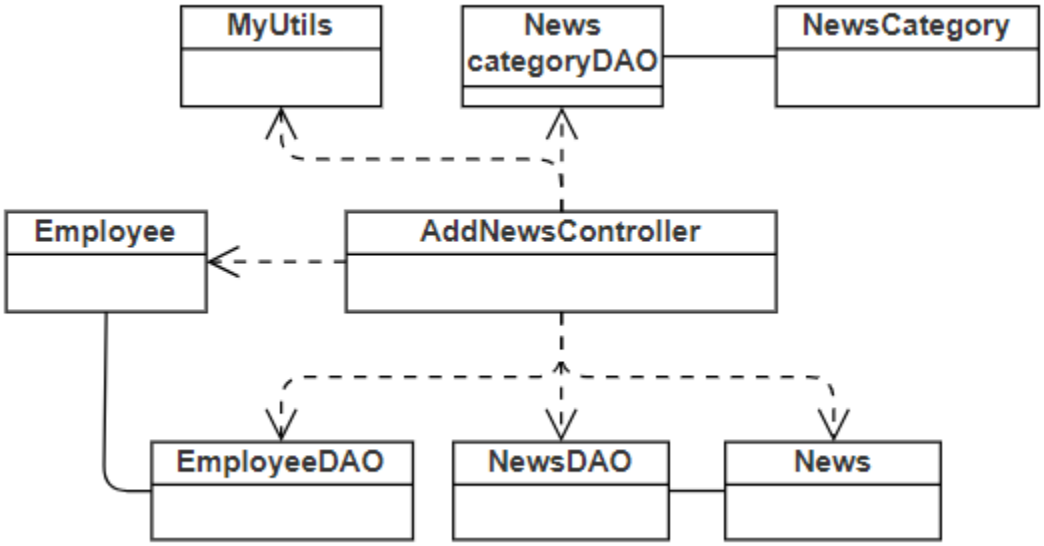
- editProfile()
  - UPDATE [dbo].[employee]\n"
  - + " SET [email] = ?\n"
  - + " ,[phone] = ?\n"
  - + " ,[address] = ?\n"
  - + " ,[firstname] = ?\n"
  - + " ,[lastname] = ?\n"
  - + " ,[img] = ?\n"

    - + " WHERE [id] = ?
- checkPassWord(String oldPassword, int id)
    - SELECT password FROM [dbo].[employee] WHERE id = ?

3.1 Staff

3.1.1 Add News

a. Class Diagram



b. Class Specifications

controller.AddNews

No	Method	Description
	doGet(HttpServletRequest request, HttpServletResponse response)	Handles HTTP GET requests; fetches news categories and existing news for editing, then forwards to the `AddNews.jsp` view.
	doPost(HttpServletRequest request, HttpServletResponse response)	Handles HTTP POST requests; processes form submissions for adding, updating, or deleting news based on the provided action.

News

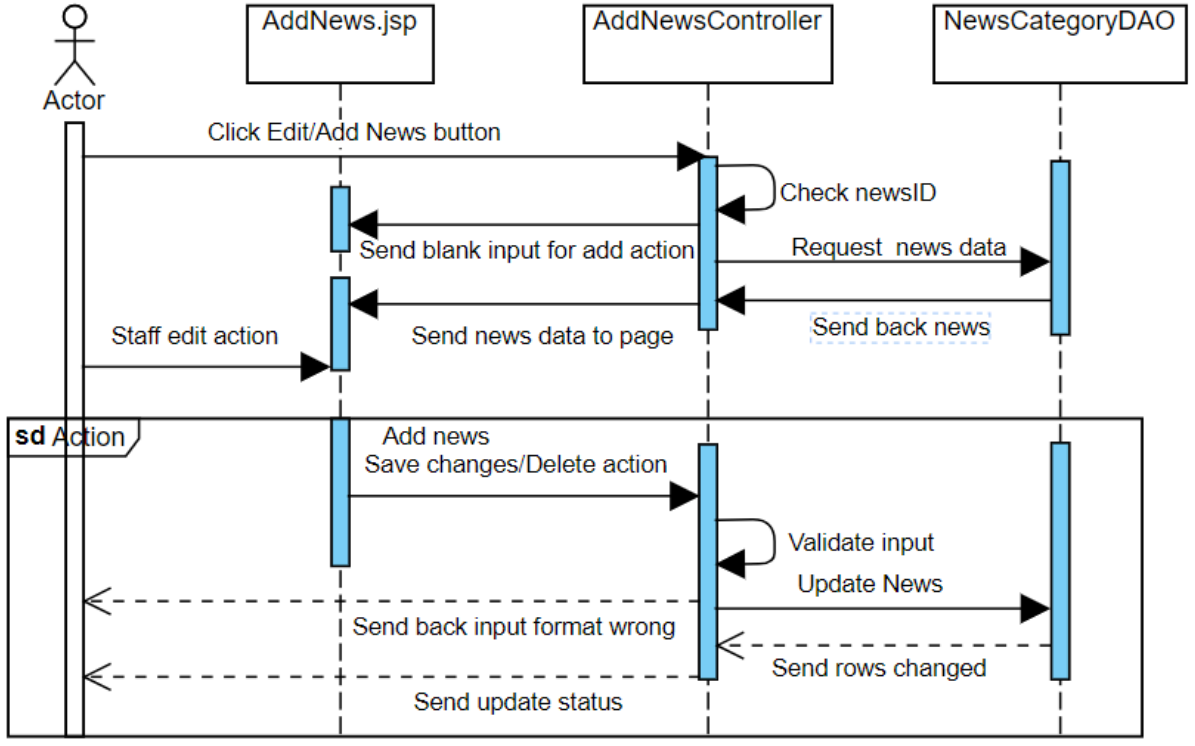
No	Method	Description
1	getManageNews()	Retrieves all news entries from the database, ordered by ID and creation date in descending order.
2	searchByNewString(String search)	Searches for news entries where the title or author's name matches the provided search string.

3	searchByNewsCategory(int search)	Searches for news entries that belong to a specific news category ID.

MyUtils

No	Method	Description
1	validateParameters(String... params)	throws an `IllegalArgumentException` if any of the parameters are null, ensuring that all required parameters are provided.
2	setAlertAttributes(HttpServletRequest request, boolean status, String action)	sets an alert message and type based on the success or failure of an action, which can be used for displaying user feedback in a web application.

c. Sequence Diagram(s)



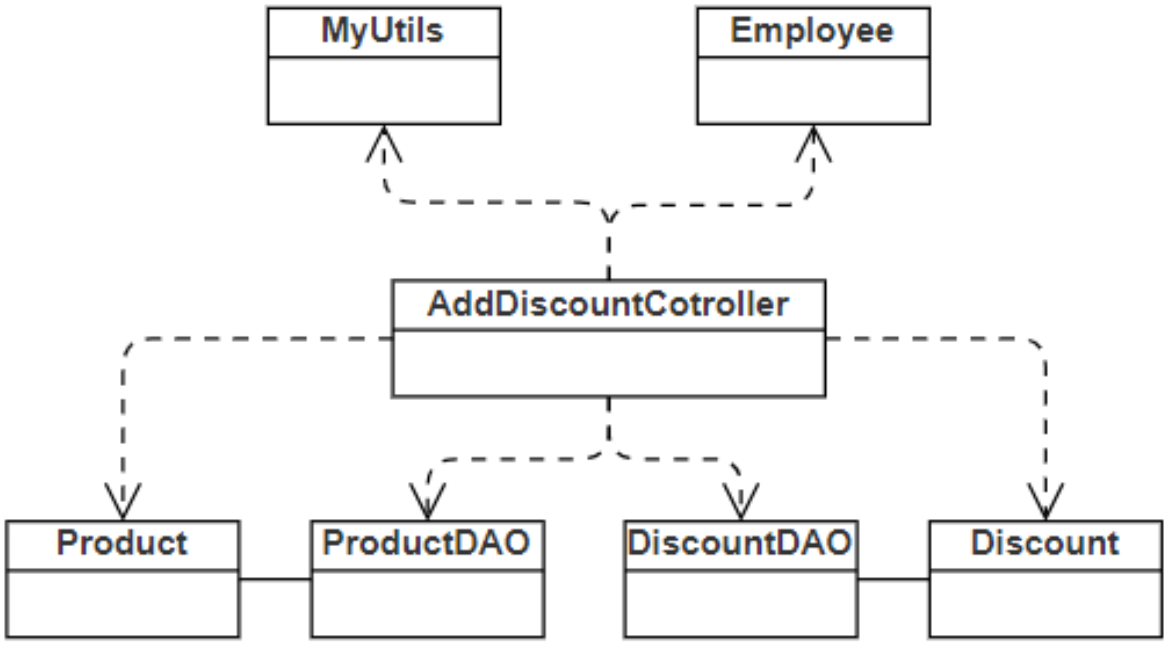
d. Database Queries

- 1, getManageNews()
  - SELECT * FROM news ORDER BY id desc, create_date DESC
- 2,searchByNewString(String search)
  - SELECT * FROM news where title LIKE '%" + search + "%' or author = (select id from employee where username like '%" + search+ "%' or firstname like '%" + search + "%' or lastname like '%" + search + "%'

- searchByNewsCategory(int search)
    - SELECT * FROM news where news_category_id = '' + search

3.1.2 Add New Discount

a. Class Diagram



b. Class Specifications

controller.AddDiscount.java

No	Method	Description
	doGet(HttpServletRequest request, HttpServletResponse response)	Handles HTTP GET requests; fetches news categories and existing news for editing, then forwards to the `AddDiscount.jsp` view.
	doPost(HttpServletRequest request, HttpServletResponse response)	Handles HTTP POST requests; processes form submissions for adding, updating, or deleting news based on the provided action.

Discount

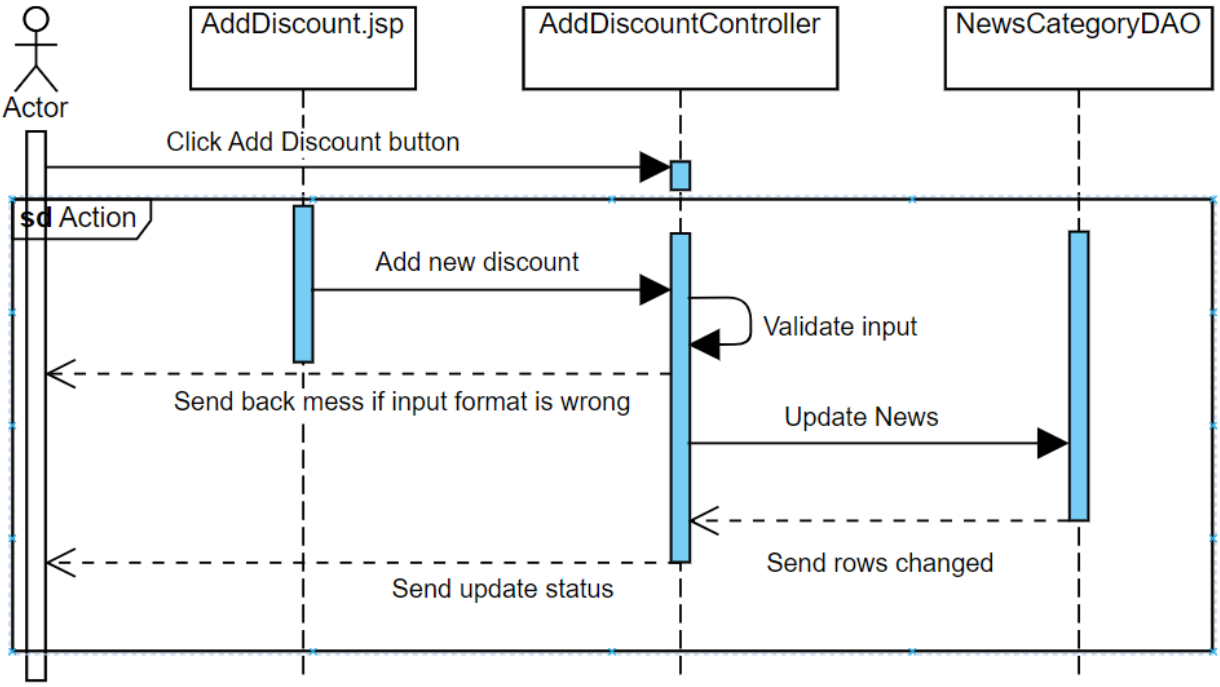
No	Method	Description
1	addDiscount(Discount discount)	Inserts a new discount record into the database

MyUtils

No	Method	Description

1	validateParameters(String... params)	throws an `IllegalArgumentException` if any of the parameters are null, ensuring that all required parameters are provided.
2	setAlertAttributes(HttpServlet Request request, boolean status, String action)	sets an alert message and type based on the success or failure of an action, which can be used for displaying user feedback in a web application.

c. Sequence Diagram(s)

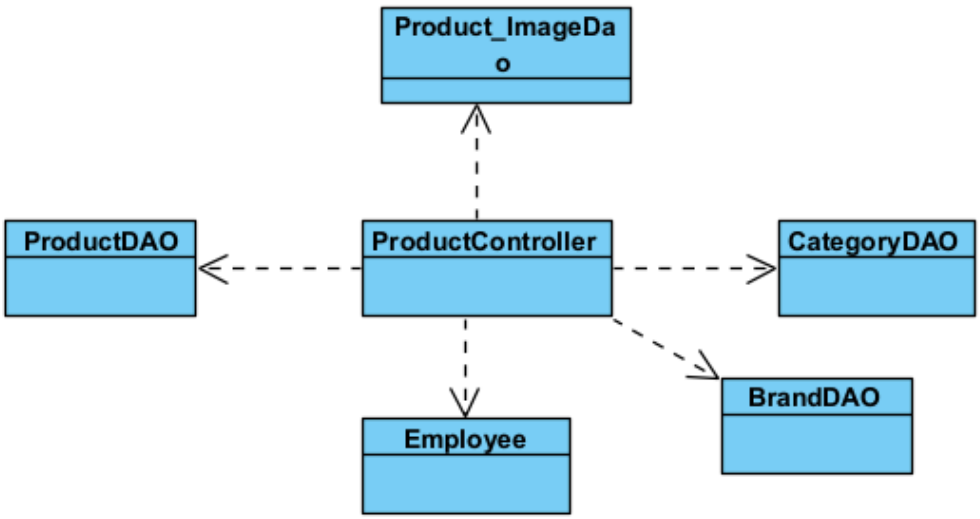


d. Database Queries

- addDiscount(Discount discount)
    - INSERT INTO discount (product_id, quantity, value, start_date, exp_date, created_by) VALUES (?, ?, ?, ?, ?, ?)

3.1.3 Add New Products

a. Class Diagram



b. Class Specifications

controller.ManageProducts.java

No	Method	Description
	doGet(HttpServletRequest request, HttpServletResponse response)	Handles HTTP GET requests; fetches news product then forwards to the `AddProduct.jsp` view.
	doPost(HttpServletRequest request, HttpServletResponse response)	Handles HTTP POST requests; processes form submissions for adding, updating, or hidden/display products based on the provided action.

ProductDAO

No	Method	Description
1	addProduct()	Inserts a new Product record into the database
2	getLastInsertedProductId()	Retrieves the ID of the most recently inserted product from the `product` table.

Product_ImageDAO

No	Method	Description
1	addProductImage()	Inserts a image name with customer id into database
2	getAll()	Get all image from product_image table

CategoryDAO

No	Method	Description
1	getAll()	Get all category from category table

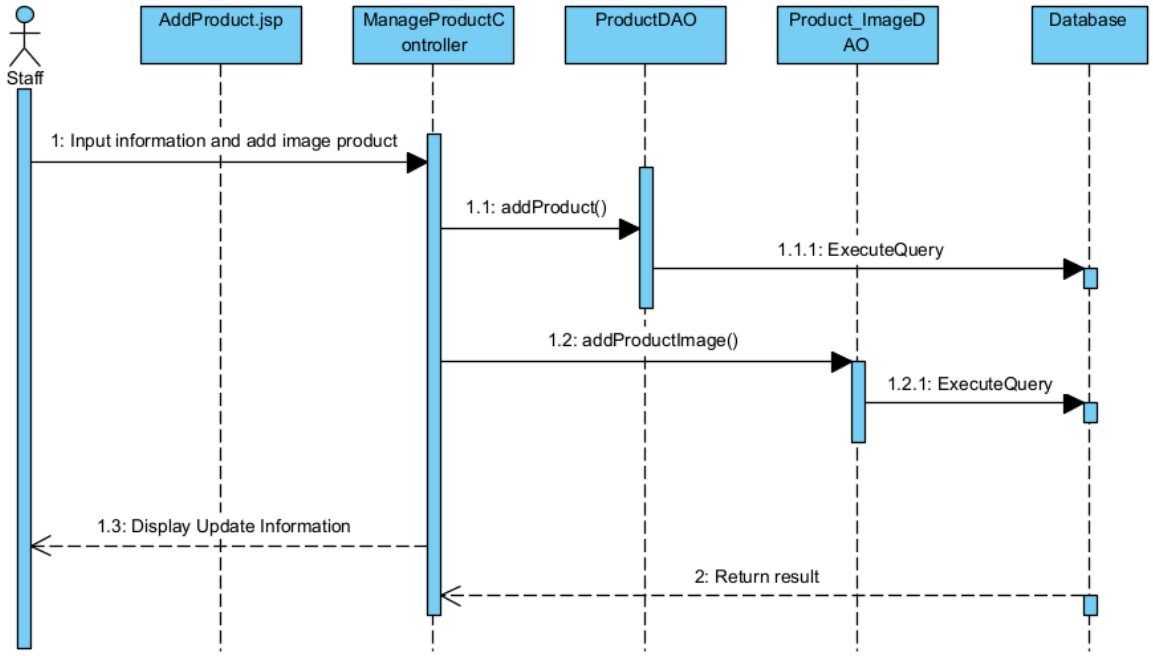
BrandDAO

No	Method	Description
1	getAll()	Get all category from brand table

Employee

No	Method	Description
1	(Employee) session.getAttribute("currentEmployee")	Set the current id of Employee as soon as the user login as Employee roles

c. Sequence Diagram(s)



d. Database Queries

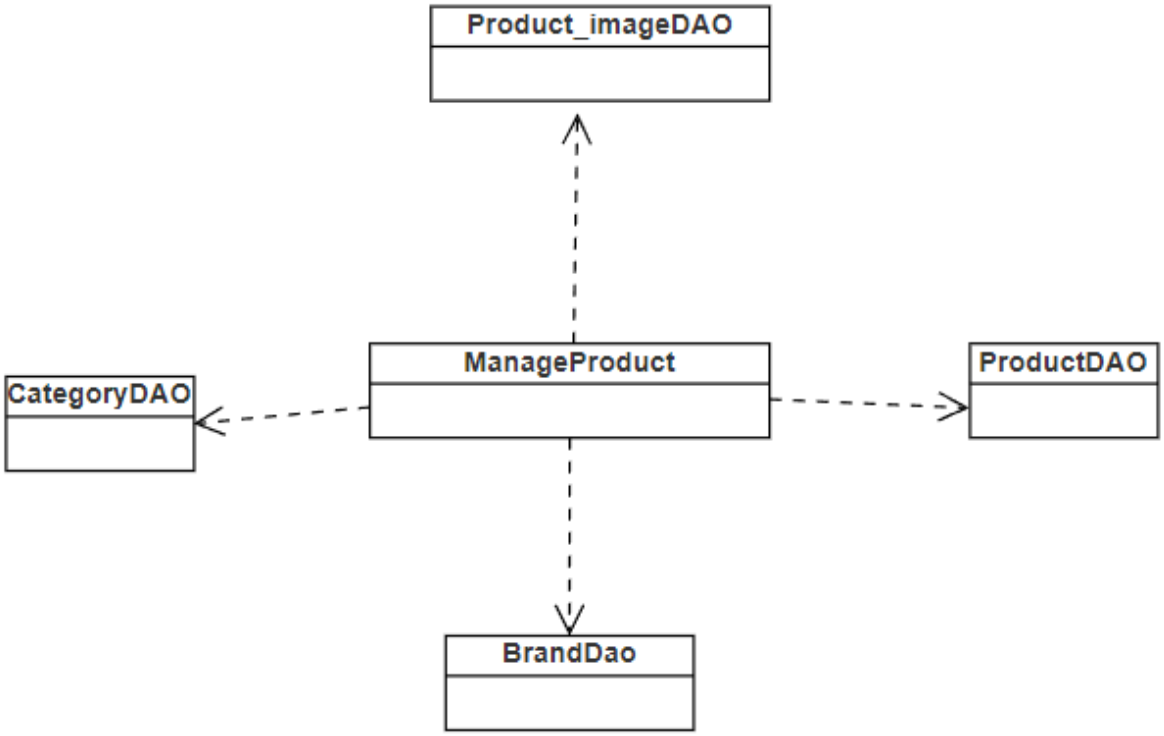
- addProduct()
  - INSERT INTO [dbo].[product]\n"
  - + " ([name]\n"
  - + " ,[CPU]\n"
  - + " ,[GPU]\n"
  - + " ,[RAM]\n"
  - + " ,[ROM]\n"
  - + " ,[monitor]\n"
  - + " ,[OS]\n"
  - + " ,[price]\n"
  - + " ,[quantity]\n"
  - + " ,[representImage]\n"
  - + " ,[description]\n"
  - + " ,[brand_id]\n"
  - + " ,[category_id]\n"
  - + " ,[created_by]\n"
  - + " ,[created_on])\n"
  - + " VALUES (?,?,?,?,?,?,?,?,?,?,?,?,?,?,?)
- getLastInsertedProductId()
  “SELECT MAX(id) AS id FROM product”

- getAllByProductID()
  "select * from product_image where product id = ?"

- addProductImage(int productId, String imageName)
  “INSERT INTO [dbo].[product_image] (product_id, img) VALUES (?, ?)”

3.1.4 Manage Product

a. Class Diagram



b. Class Specifications

controller.ManageProducts.java

No	Method	Description
1	doGet(HttpServletRequest request, HttpServletResponse response)	Handles HTTP GET requests; fetches news product then forwards to the `AddProduct.jsp` view.
2	doPost(HttpServletRequest request, HttpServletResponse response)	Handles HTTP POST requests; processes form submissions for adding, updating, or hidden/display products based on the provided action.

ProductDAO

No	Method	Description
1	editProductById()	Change product properties into data base
2	getLastInsertedProductId()	Retrieves the ID of the most recently inserted product from the `product` table.
3	hiddenProduct()	Change product status into hidden
4	displayProduct	Display product which is hidden

Product_ImageDAO

No	Method	Description
1	addProductImage()	Inserts a image name with customer id into database
2	getAll()	Get all image from product_image table

CategoryDAO

No	Method	Description
1	getAll()	Get all category from category table

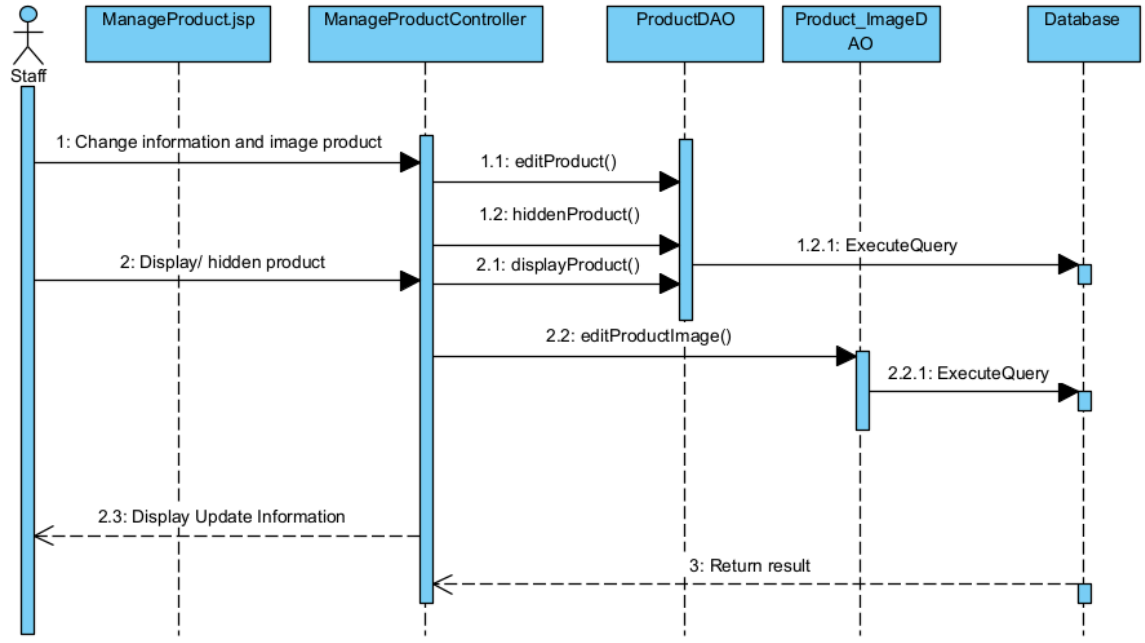
BrandDAO

No	Method	Description
1	getAll()	Get all category from brand table

Employee

No	Method	Description
1	(Employee) session.getAttribute("currentEmployee")	Set the current id of Employee as soon as the user login as Employee roles

c. Sequence Diagram(s)



d. Database Queries

- editProductById()

```
"UPDATE [dbo].[product]\n"
        + "   SET [name] = ?\n"
        + "      ,[CPU] = ?\n"
        + "      ,[GPU] = ?\n"
        + "      ,[RAM] = ?\n"
        + "      ,[ROM] = ?\n"
        + "      ,[monitor] = ?\n"
        + "      ,[OS] = ?\n"
        + "      ,[price] = ?\n"
        + "      ,[description] = ?\n"
        + "      ,[brand_id] = ?\n"
        + "      ,[category_id] = ?\n"
        + "      ,[quantity]= ?\n"
        + "      ,[representImage]= ?\n"
        + "      ,[modified_by] = ?\n"
        + "      ,[modified_on] = ?\n"
        + "  WHERE [id] = ?"
```

- hiddenProduct()

```
 "UPDATE [dbo].[product]\n"
        + "   SET [status] = 0\n"
        + "      WHERE [id] = ?"
```

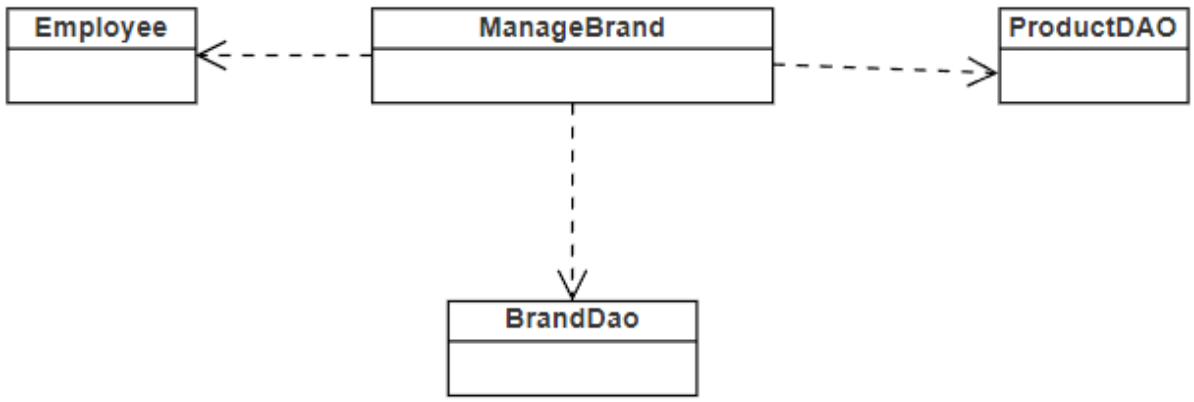
displayProduct()

"UPDATE [dbo].[product] SET [status] = 1 WHERE [id] = (SELECT p.id \n" +
"   FROM product p \n" +
"   JOIN brand b ON b.id=p.brand_id\n" +
"   Join category c on c.id=p.category_id\n" +
"   WHERE b.status = 1 and c.status=1 and p.id=?)"

- getAllProductByCategory()
  "select * from product where category = ?"

3.1.5. Manage Brand

a. Class Diagram



b. Class Specifications

controller.ManageBrand.java

No	Method	Description
	doGet(HttpServletRequest request, HttpServletResponse response)	Handles HTTP GET requests; fetches news categories and existing news for editing, then forwards to the `ManageCategory.jsp` view.
	doPost(HttpServletRequest request, HttpServletResponse response)	Handles HTTP POST requests; processes form submissions for adding, updating, or deleting news based on the provided action.

CategoryDAO

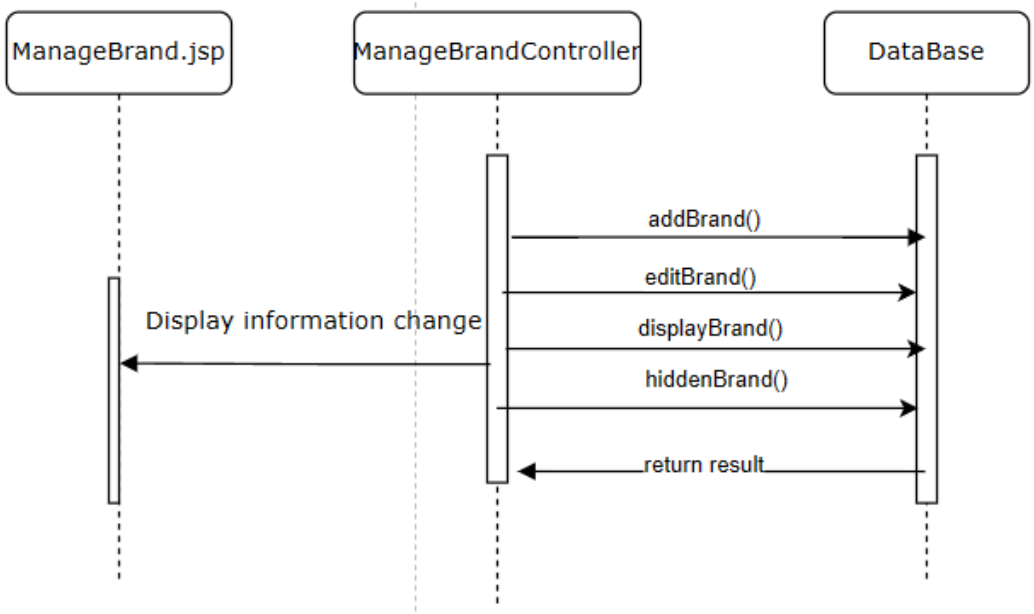
No	Method	Description
1	getAll()	This method retrieves all brands from the database by

		calling the `get` method with `null` as the parameter
2	addBrand(String name, String logo, String link, int created_by)	This method adds a new brand to the database. It constructs and executes an SQL `INSERT` statement with the given brand name, logo, link, and creator ID.
3	editBrandById(String name, String logo, String link, int modified_by, int id)	This method updates an existing brand in the database. It constructs and executes an SQL `UPDATE` statement with the new brand name, logo, link, modifier ID, and brand ID.
4	hiddenBrand(int id)	This method hides a brand and all associated products in the database by setting their status to `0`. It uses a transaction to ensure both updates are performed atomically.
5	displayBrand(int id)	This method makes a brand and all associated products with active categories visible in the database by setting their status to `1`. It uses a transaction to ensure both updates are performed atomically.

MyUtils

No	Method	Description
1	setAlertAttributes(HttpServletRequest request, boolean status, String action)	sets an alert message and type based on the success or failure of an action, which can be used for displaying user feedback in a web application.
2	getArrayListByPaging(ArrayList<T> list, int pageNumber, int ItemsOfPage)	calculates the start and end indices for the desired page and returns the corresponding sublist.

c. Sequence Diagram(s)



d. Database Queries

- addBrand()

```
"INSERT INTO [dbo].[brand] "
        + "([name], [logo], [link], [created_by], [created_on]) "
        + "VALUES (?, ?, ?, ?, ?)"
```

- editBrandById()

```
"UPDATE [dbo].[brand]\n"
        + "   SET [name] = ?\n"
        + "      ,[logo] = ?\n"
        + "      ,[link] = ?\n"
        + "      ,[modified_by] = ?\n"
        + "      ,[modified_on] = ?\n"
        + " WHERE [id] = ?"
```

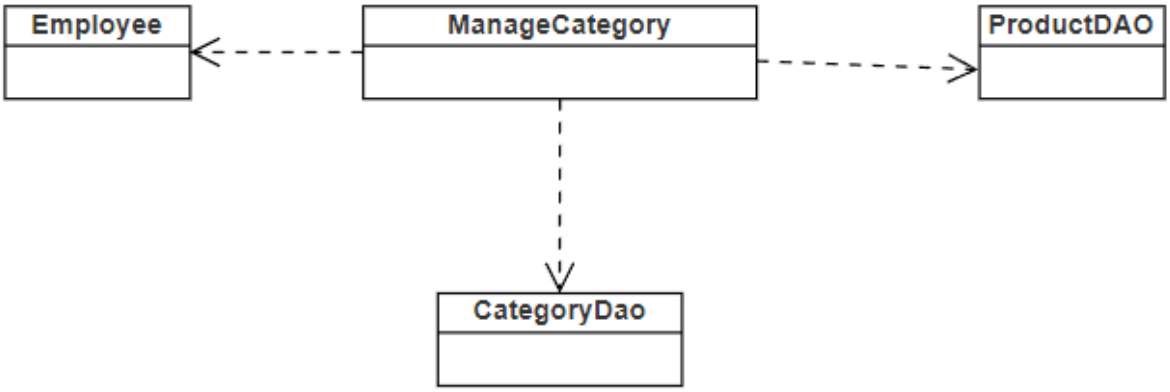
- hiddenBrand()

```
 "UPDATE [dbo].[brand]\n"
        + "   SET [status] = 0\n"
        + "      WHERE [id] = ?"
"UPDATE [dbo].[product]\n"
        + "   SET [status] = 0\n"
        + " WHERE [brand_id] = ?"
```

- displayProduct()
  "UPDATE [dbo].[brand]\n"
  + " SET [status] = 1\n"
  + " WHERE [id] = ?"
  """ UPDATE product
  SET status = 1
  WHERE id IN (
  SELECT p.id
  FROM product p
  JOIN category c ON c.id = p.category_id
  WHERE c.status = 1
) and [brand_id] = ?"""

3.1.6. Manage Product Category

a. Class Diagram



b. Class Specifications

controller.ManageCategory.java

No	Method	Description
	doGet(HttpServletRequest request, HttpServletResponse response)	Handles HTTP GET requests; fetches news categories and existing news for editing, then forwards to the `ManageCategory.jsp` view.
	doPost(HttpServletRequest request, HttpServletResponse response)	Handles HTTP POST requests; processes form submissions for adding, updating, or deleting news based on the provided action.

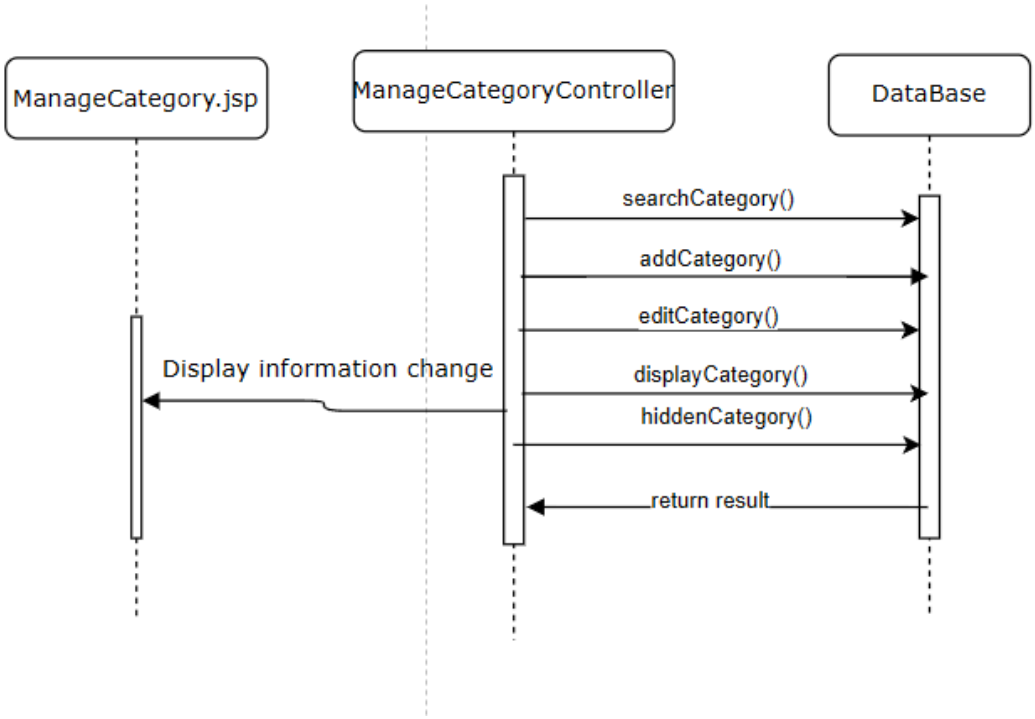
CategoryDAO

No	Method	Description
1	getAll()	This method retrieves all categories from the database by calling the `get` method with `null` as the parameter.
2	addCategory(String name, int created_by)	This method adds a new category to the database. It constructs and executes an SQL `INSERT` statement with the given category name and creator ID.
3	editCategory(String name, int modified_by, int id)	This method updates an existing category in the database. It constructs and executes an SQL `UPDATE` statement with the new category name, modifier ID, and category ID.
4	hiddenCategory(int id)	This method hides a category and all associated products in the database by setting their status to `0`. It uses a transaction to ensure both updates are performed atomically.
5	displayCategory(int id)	This method makes a category and all associated products with active brands visible in the database by setting their status to `1`. It uses a transaction to ensure both updates are performed atomically.

MyUtils

No	Method	Description
1	setAlertAttributes(HttpServletRequest request, boolean status, String action)	sets an alert message and type based on the success or failure of an action, which can be used for displaying user feedback in a web application.
2	getArrayListByPaging(ArrayList<T> list, int pageNumber, int ItemsOfPage)	calculates the start and end indices for the desired page and returns the corresponding sublist.

c. Sequence Diagram(s)



d. Database Queries

- addCategory()

```
"INSERT INTO [dbo].[category] "
        + "([name], [created_by], [created_on]) "
        + "VALUES (?, ?, ?)"
```

- editCategory()

```
 "UPDATE [dbo].[category]\n"
        + "   SET [name] = ?\n"
        + "      ,[modified_by] = ?\n"
        + "      ,[modified_on] = ?\n"
        + " WHERE [id] = ?"
```

- displayCategory()

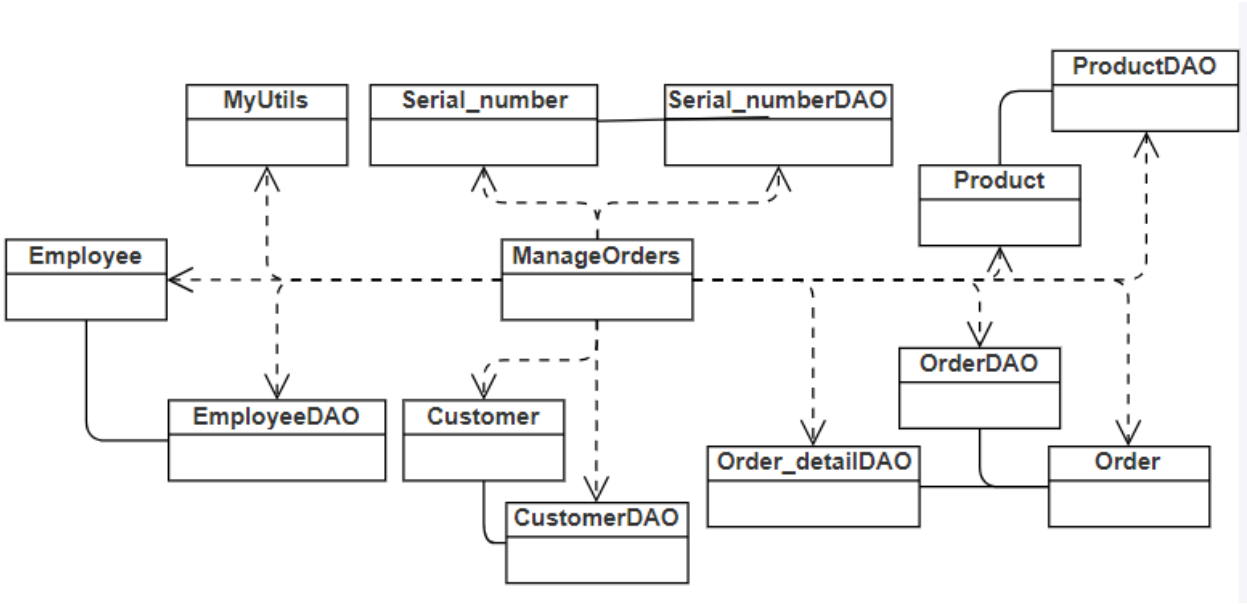
```
"UPDATE [dbo].[category]\n"
        + "   SET [status] = 1\n"
        + " WHERE [id] = ?"
 """UPDATE product
    SET status = 1
    WHERE id IN (
    SELECT p.id
    FROM product p
    JOIN brand b ON b.id=p.brand_id
    WHERE b.status = 1
    ) and [category_id] = ?"""
```

- hiddenCategory()

```
"UPDATE [dbo].[category]\n"
        + "   SET [status] = 0\n"
        + " WHERE [id] = ?"
"UPDATE [dbo].[product]\n"
        + "   SET [status] = 0\n"
        + " WHERE [category_id] = ?"
```

3.1.7 Manage Orders

a. Class Diagram



b. Class Specifications

controller.ManageOrders.java

No	Method	Description
	doGet(HttpServletRequest request, HttpServletResponse response)	Handles HTTP GET requests; fetches news categories and existing news for editing, then forwards to the `AddDiscount.jsp` view.
	doPost(HttpServletRequest request, HttpServletResponse response)	Handles HTTP POST requests; processes form submissions for adding, updating, or deleting news based on the provided action.

OrderDAO

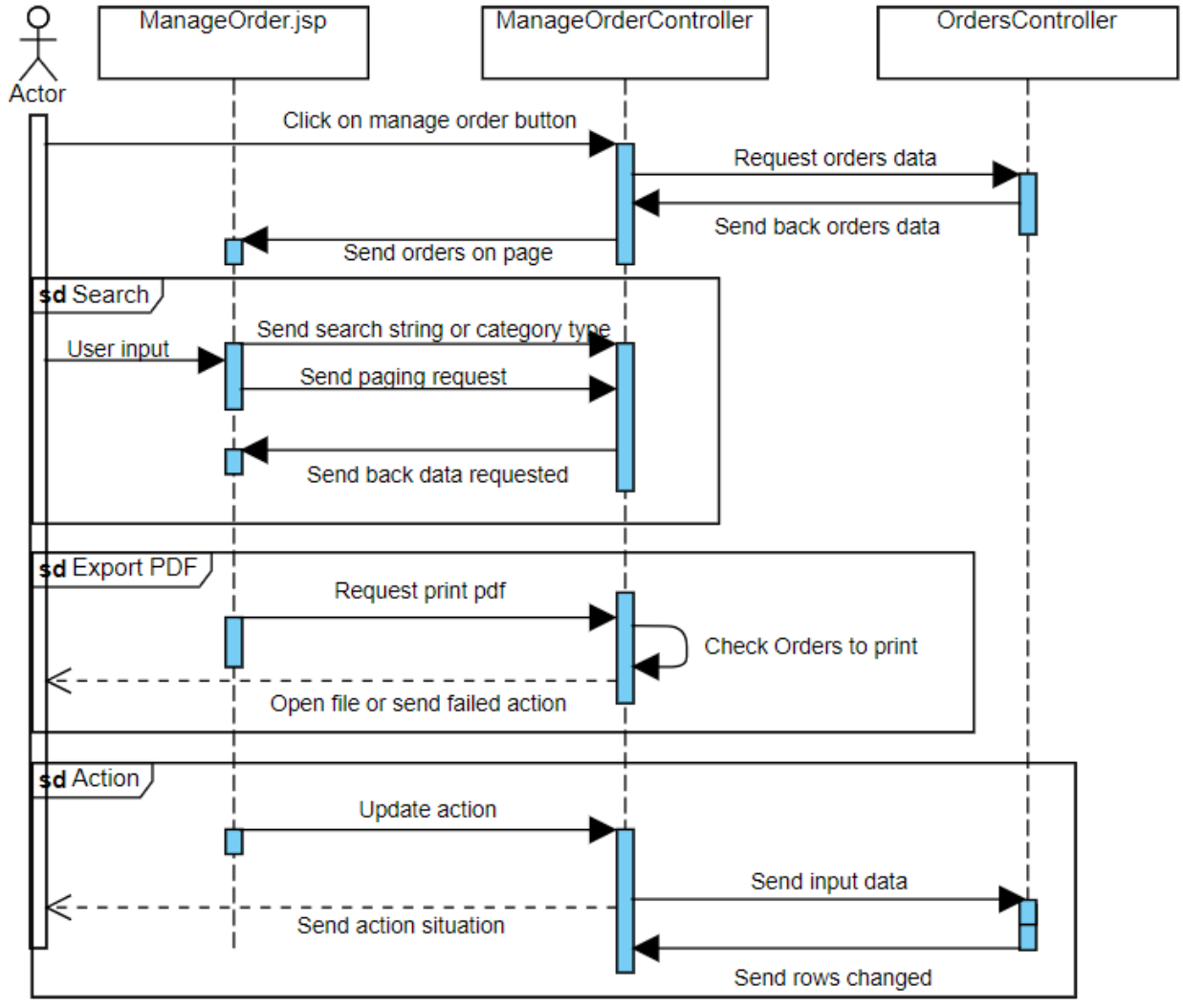
No	Method	Description
1	getOrders()	Retrieves orders that have been paid and have order details.
2	searchByOrderStatus(int order_status)	Retrieves orders with a specific status.

MyUtils

No	Method	Description

1	removeDiacritics(String input)	replaces characters with diacritical marks with their non-diacritical equivalents based on a predefined map
2	setAlertAttributes(HttpServletRequest request, boolean status, String action)	sets an alert message and type based on the success or failure of an action, which can be used for displaying user feedback in a web application.
3	getArrayListByPaging(ArrayList<T> list, int pageNumber, int ItemsOfPage)	calculates the start and end indices for the desired page and returns the corresponding sublist.

c. Sequence Diagram(s)



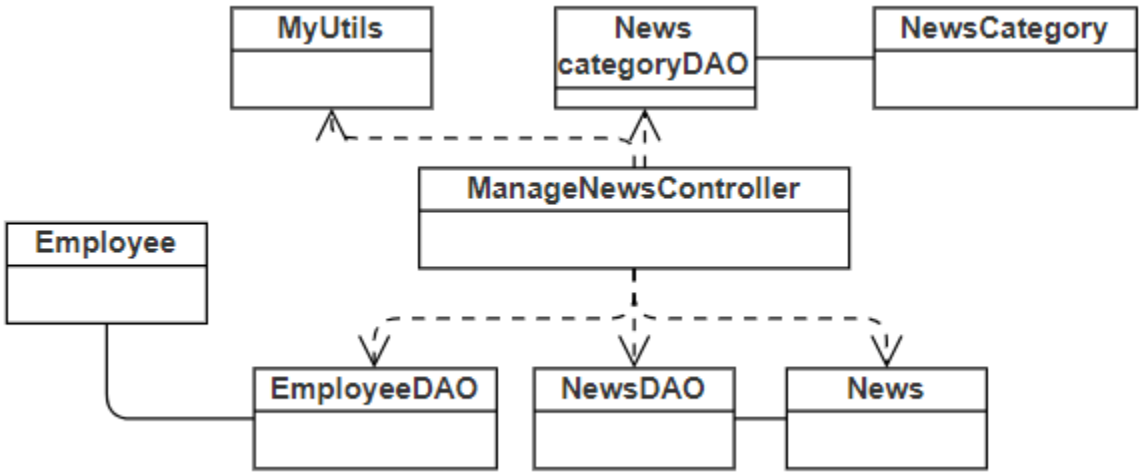
d. Database Queries

- getOrders()
  - SELECT * FROM [order] where paid_date is not null and id in (select od.order_id from order_detail od) order by id desc

- searchByOrderStatus(int order_status)
    - SELECT * FROM [order] where order_status = " + order_status

3.1.8 Manage News

a. Class Diagram



b. Class Specifications

controller.ManageNews.java

No	Method	Description
	doGet(HttpServletRequest request, HttpServletResponse response)	Handles HTTP GET requests; fetches news categories and existing news for editing, then forwards to the `AddDiscount.jsp` view.
	doPost(HttpServletRequest request, HttpServletResponse response)	Handles HTTP POST requests; processes form submissions for adding, updating, or deleting news based on the provided action.

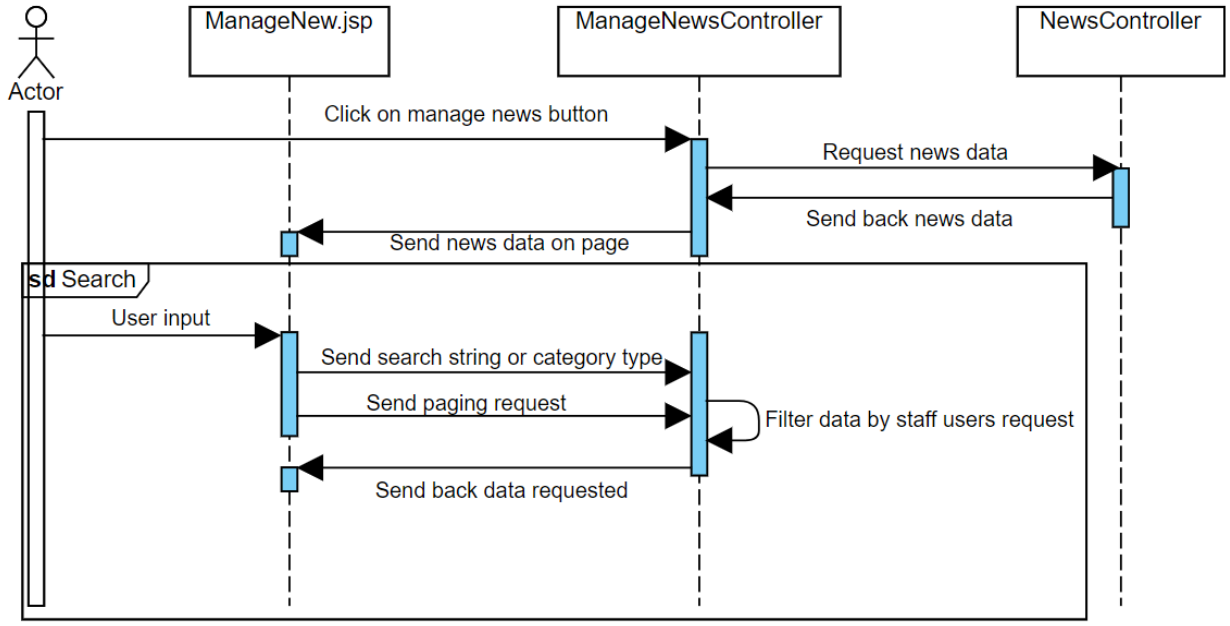
NewsDAO

No	Method	Description
1	getManageNews()	Retrieves all news sorted by ID and creation date in descending order.
2	searchByNewString(String search)	Searches for news items by a search string in the title or author's name.
3	searchByNewsCategory(int search)	Searches for news items by a specific news category.

MyUtils

No	Method	Description
1	setAlertAttributes(HttpServletRequest request, boolean status, String action)	sets an alert message and type based on the success or failure of an action, which can be used for displaying user feedback in a web application.
2	getArrayListByPaging(ArrayList<T> list, int pageNumber, int ItemsOfPage)	calculates the start and end indices for the desired page and returns the corresponding sublist.

c. Sequence Diagram(s)

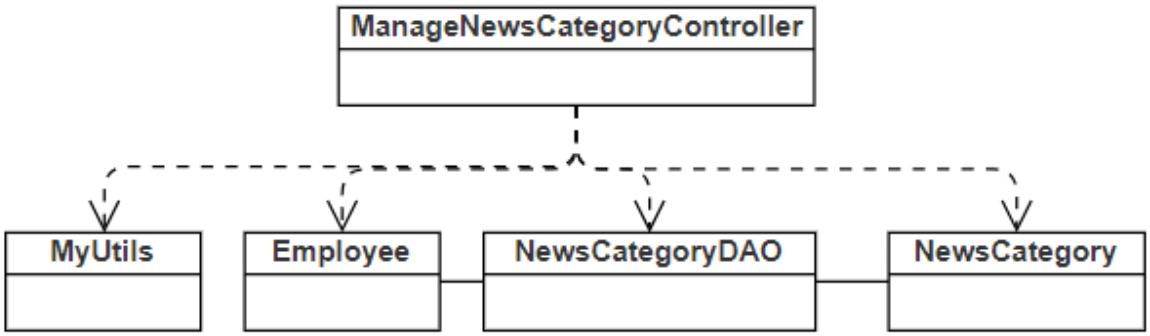


d. Database Queries

- getManageNews()
  - SELECT * FROM news ORDER BY id desc, create_date DESC
- searchByNewString(String search)
  - SELECT * FROM news where title LIKE '%" + search + "%' or author = (select id from employee where username like '%" + search + "%' or firstname like '%" + search + "%' or lastname like '%" + search + "%')
- searchByNewsCategory(int search)
  - SELECT * FROM news where news_category_id = " + search

3.1.9 Manage News Category

a. Class Diagram



b. Class Specifications

controller.ManageNews.java

No	Method	Description
	doGet(HttpServletRequest request, HttpServletResponse response)	Handles HTTP GET requests; fetches news categories and existing news for editing, then forwards to the `AddDiscount.jsp` view.
	doPost(HttpServletRequest request, HttpServletResponse response)	Handles HTTP POST requests; processes form submissions for adding, updating, or deleting news based on the provided action.

NewsDAO

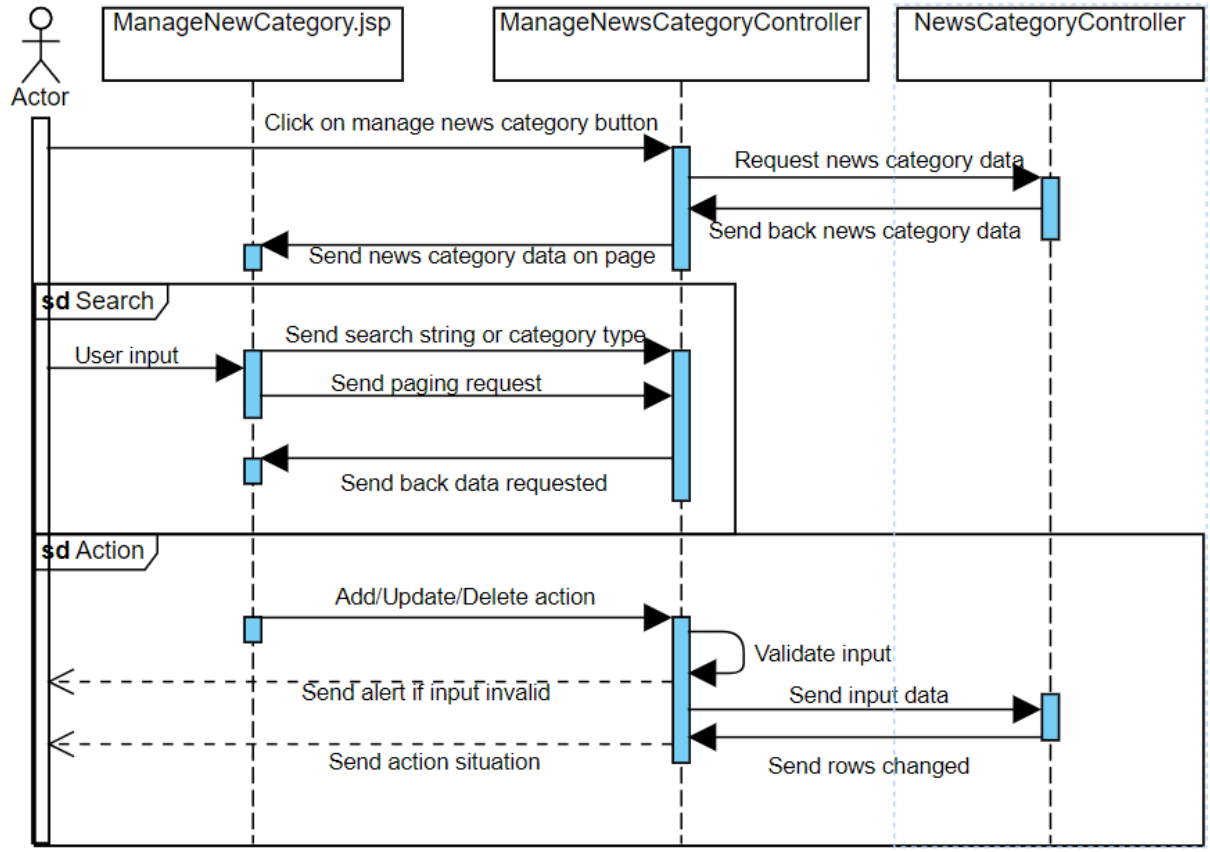
No	Method	Description
1	searchNewCategoryString(String search)	Searches for news categories by a search string in their name.
2	getAll()	Retrieves all news categories.

MyUtils

No	Method	Description
1	setAlertAttributes(HttpServletRequest request, boolean status, String action)	sets an alert message and type based on the success or failure of an action, which can be used for displaying user feedback in a web application.

2	getArrayListByPaging(ArrayList<T> list, int pageNumber, int ItemsOfPage)	calculates the start and end indices for the desired page and returns the corresponding sublist.

c. Sequence Diagram(s)

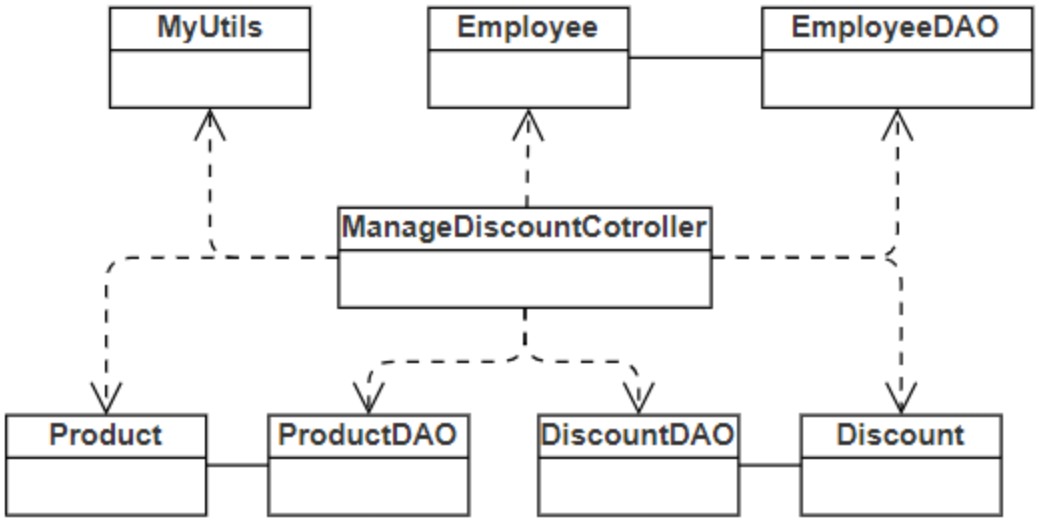


d. Database Queries

- getAll()
  - select * from news_category
- searchNewCategoryString(String search)
  - select * from news_category where name like '%" + search + "%'

3.1.10 Manage Discount

a. Class Diagram



b. Class Specifications

controller.ManageNews.java

No	Method	Description
	doGet(HttpServletRequest request, HttpServletResponse response)	Handles HTTP GET requests; fetches news categories and existing news for editing, then forwards to the `AddDiscount.jsp` view.
	doPost(HttpServletRequest request, HttpServletResponse response)	Handles HTTP POST requests; processes form submissions for adding, updating, or deleting news based on the provided action.

DiscountDAO

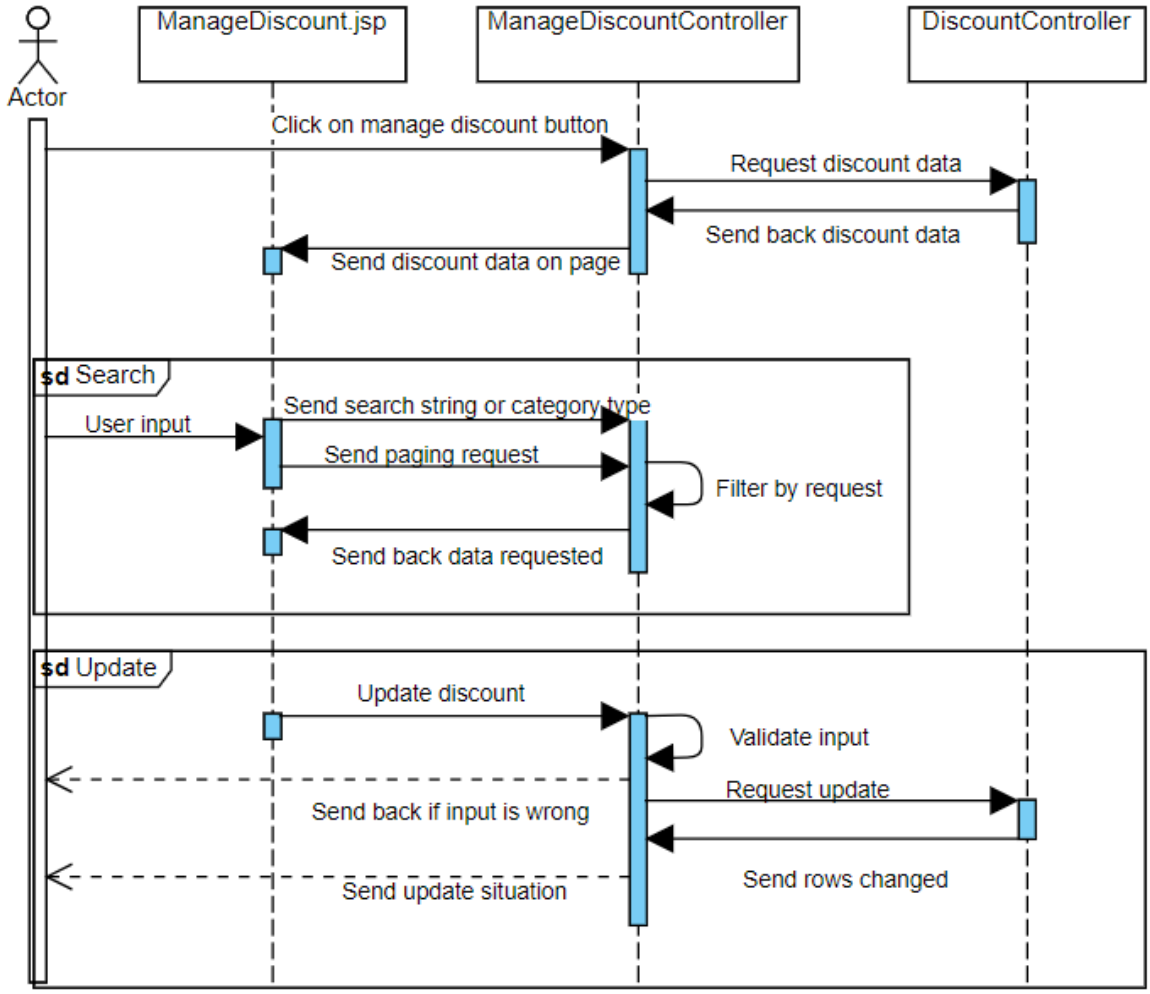
No	Method	Description
1	updateDiscount(Discount discount)	Updates an existing discount record in the database.
2	addDiscount(Discount discount)	Adds a new discount record to the database.

3	deleteDiscount(int discountID)	Deletes a discount record from the database based on its ID.

MyUtils

No	Method	Description
1	setAlertAttributes(HttpServletRequest request, boolean status, String action)	sets an alert message and type based on the success or failure of an action, which can be used for displaying user feedback in a web application.
2	validateParameters(String... params)	throws an `IllegalArgumentException` if any of the parameters are null, ensuring that all required parameters are provided.

c. Sequence Diagram(s)



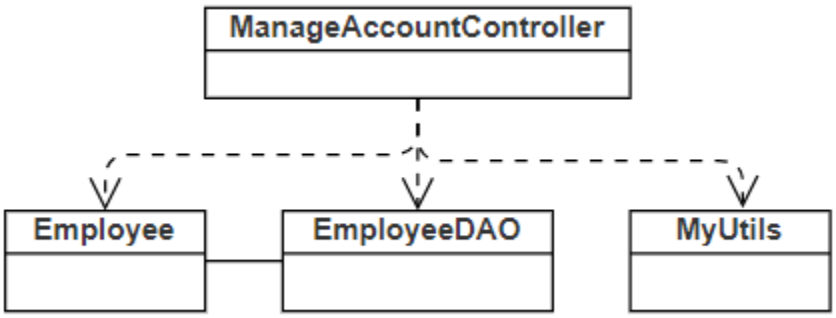
d. Database Queries

- updateDiscount(Discount discount)
    - UPDATE [dbo].[discount]\n"
    - + "   SET [product_id] = ?\n"
    - + "   ,[quantity] = ?\n"
    - + "   ,[value] = ?\n"
    - + "   ,[start_date] = ?\n"
    - + "   ,[exp_date] = ?\n"
    - + "   ,[created_by] = ?\n"
    - + " WHERE [id] = ?
- addDiscount(Discount discount)
    - INSERT INTO discount (product_id, quantity, value, start_date, exp_date, created_by) VALUES (?, ?, ?, ?, ?, ?)
- deleteDiscount(int discountID)
    - DELETE FROM [dbo].[discount] WHERE [id] = ?

3.2 Manager

3.2.1 Add new employee

a. Class Diagram



b. Class Specifications

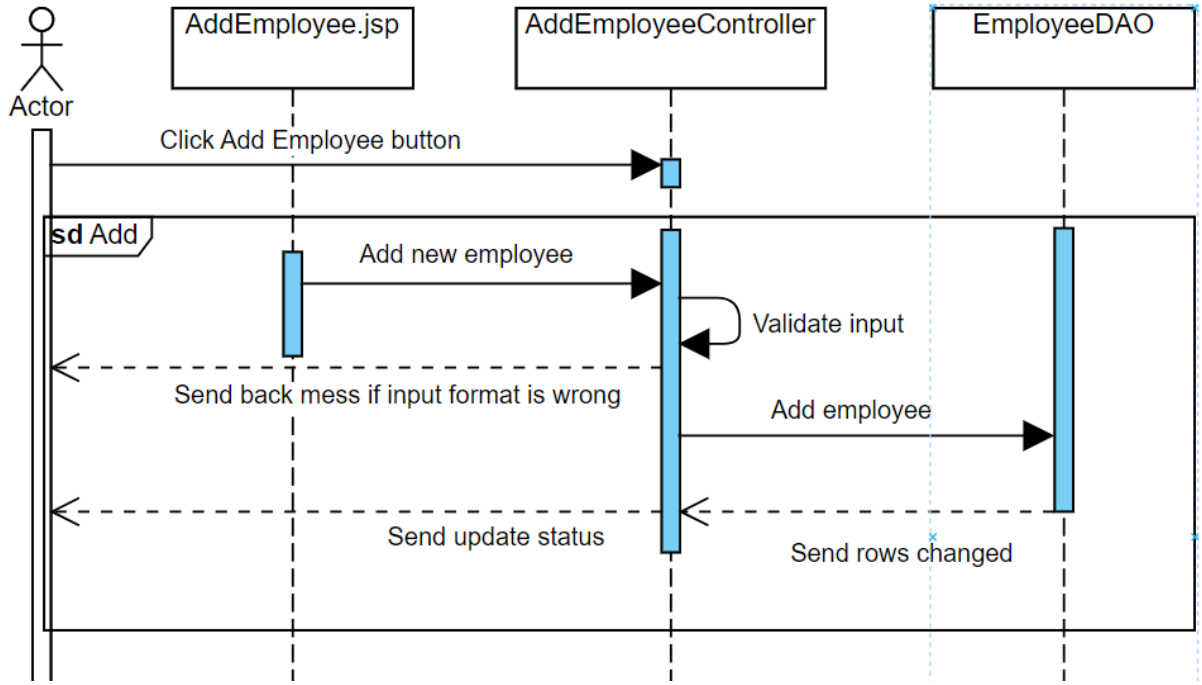
controller.ManageNews.java

No	Method	Description
	doGet(HttpServletRequest request, HttpServletResponse response)	Handles HTTP GET requests; fetches news categories and existing news for editing, then forwards to the `AddDiscount.jsp` view.
	doPost(HttpServletRequest request, HttpServletResponse response)	Handles HTTP POST requests; processes form submissions for adding, updating, or deleting news based on the provided action.

EmployeeDAO

No	Method	Description
1	checkSignUp(String username, String email)	Checks if a username or email already exists in the database
2	addEmployee(Employee e)	Adds a new employee to the database.

c. Sequence Diagram(s)

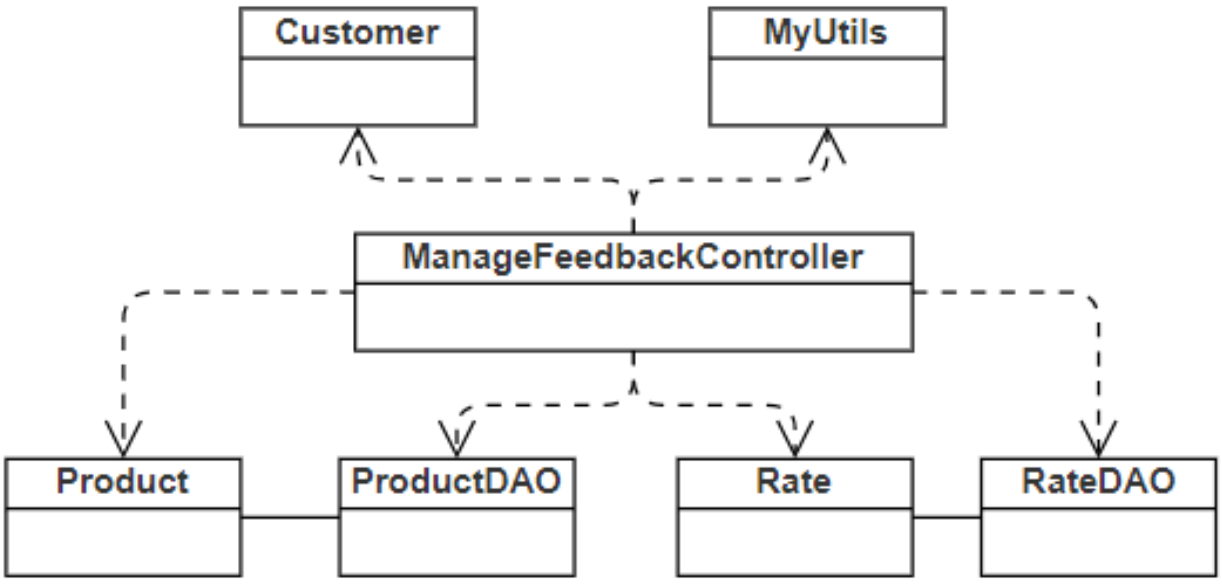


d. Database Queries

- addEmployee(Employee e)
    - INSERT INTO [dbo].[employee]\n"
    - + " ([username]\n"
    - + " ,[password]\n"
    - + " ,[email]\n"
    - + " ,[phone]\n"
    - + " ,[address]\n"
    - + " ,[firstname]\n"
    - + " ,[lastname]\n"
    - + " ,[reg_date]\n"
    - + " ,[role_id])\n"
    - + " VALUES\n"
    - + " (?,?,?,?,?,?,?,?,?)

3.2.2 Manage Feedback

a. Class Diagram



b. Class Specifications

controller.ManageNews.java

No	Method	Description
	doGet(HttpServletRequest request, HttpServletResponse response)	Handles HTTP GET requests; fetches news categories and existing news for editing, then forwards to the `AddDiscount.jsp` view.
	doPost(HttpServletRequest request, HttpServletResponse response)	Handles HTTP POST requests; processes form submissions for adding, updating, or deleting news based on the provided action.

NewsDAO

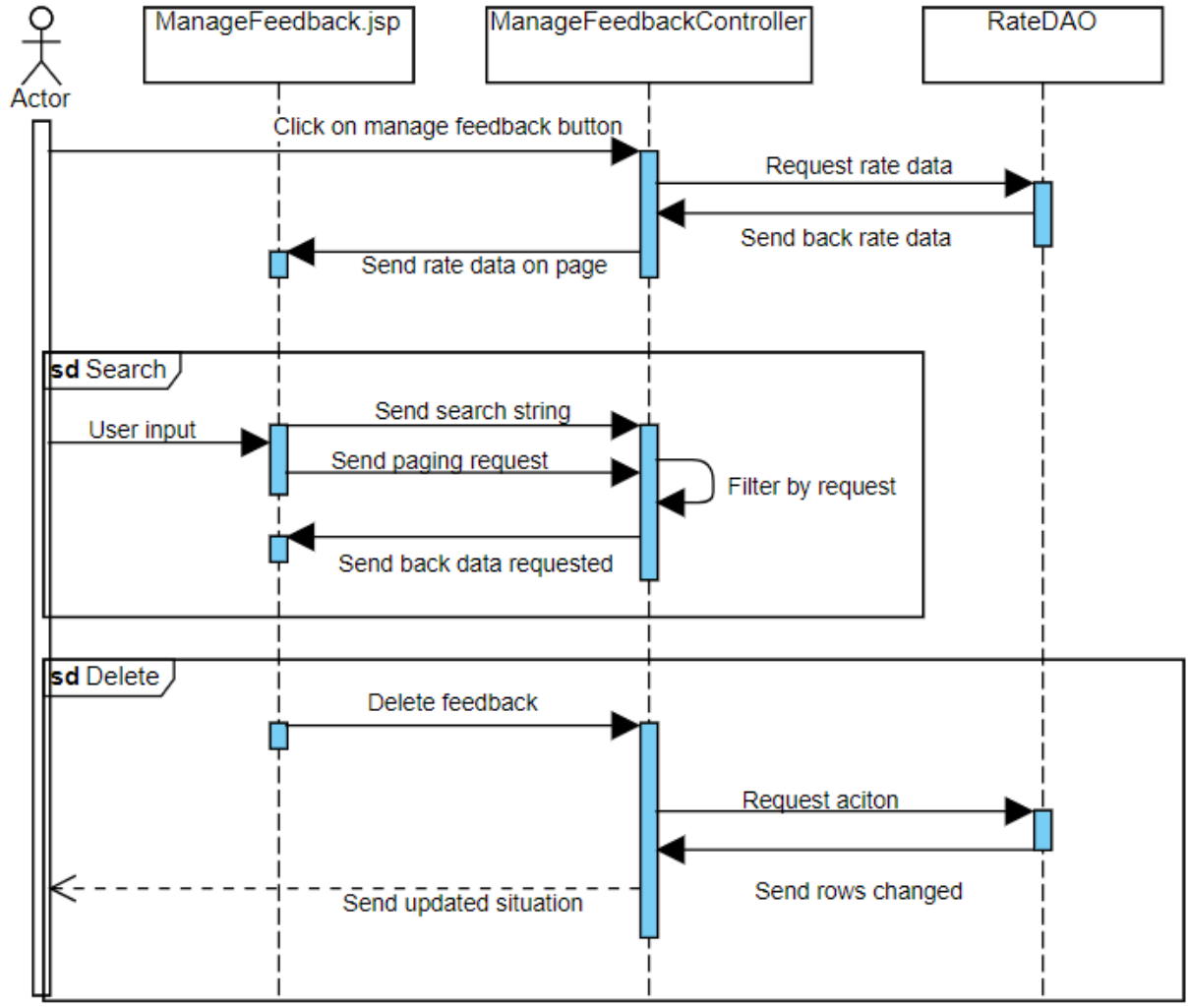
No	Method	Description
1	deleteRateById(int rateId)	Deletes a rate record from the database based on the given rate ID.

MyUtils

No	Method	Description
1	setAlertAttributes(HttpServlet	sets an alert message and type based on the success or

	Request request, boolean status, String action)	failure of an action, which can be used for displaying user feedback in a web application.

c. Sequence Diagram(s)



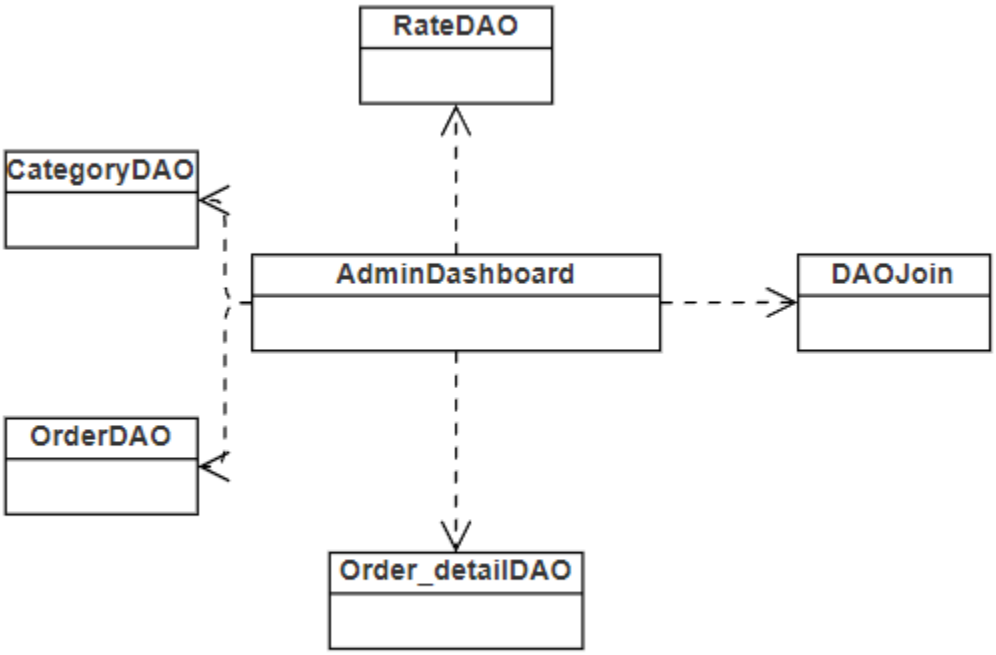
d. Database Queries

- deleteRateById(int rateId)
    - DELETE FROM rate WHERE id = ?

3.3 Admin

3.3.1 View statistic

a. Class Diagram



b. Class Specifications

controller.ManageNews.java

No	Method	Description
	doGet(HttpServletRequest request, HttpServletResponse response)	Handles HTTP GET requests; fetches news categories and existing news for editing, then forwards to the `AddDiscount.jsp` view.
	doPost(HttpServletRequest request, HttpServletResponse response)	Handles HTTP POST requests; processes form submissions for adding, updating, or deleting news based on the provided action.

OrderDAO

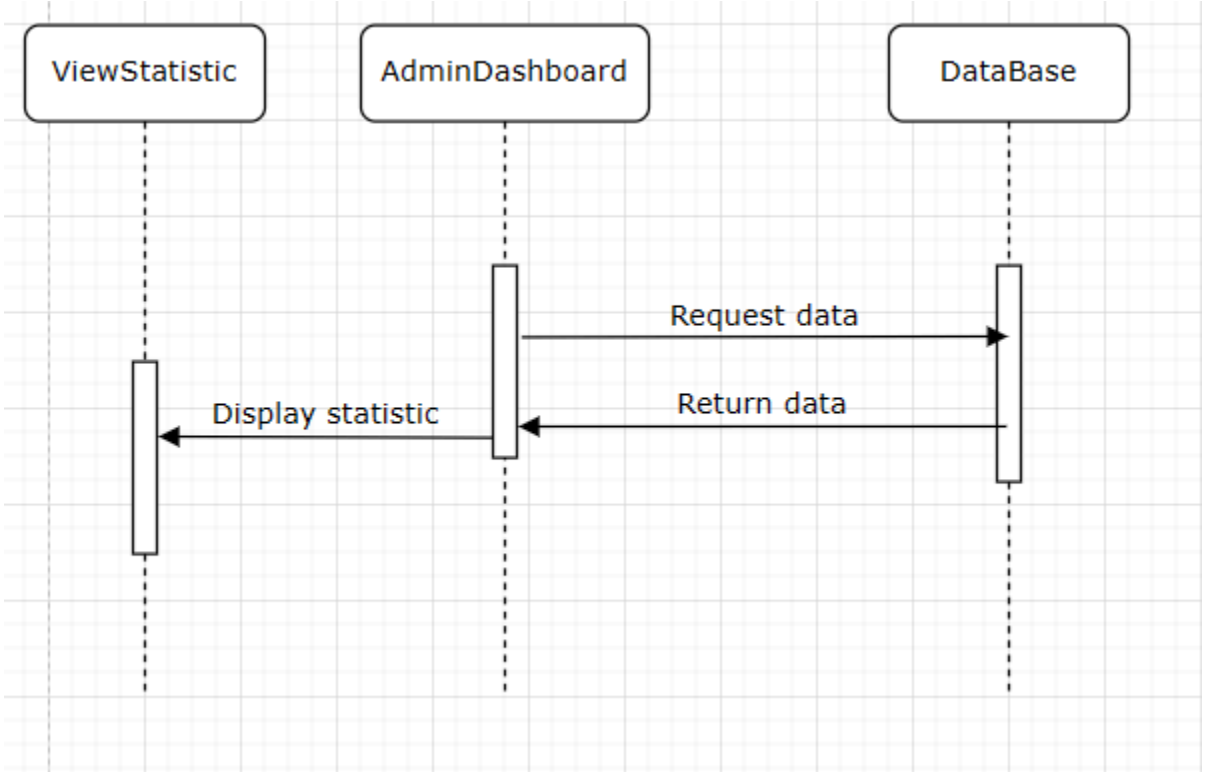
No	Method	Description

1	getTotalSales()	Retrieves the total sales amount from the orders.
2	getTotalSalesByDay(String date)	Retrieves the total sales amount for a specific day from the orders

DAOjoin

No	Method	Description
1	getNumberOfProductsSoldBy Category(int categoryId)	Retrieves the number of products sold for a specific category
2	getProductRate()	Retrieves a list of product ratings for display in the admin dashboard.
3	getOrderAdminStatistic()	Retrieves order statistics for the admin dashboard.
4	getNumberProductSoldByDay (String date)	Retrieves the number of products sold on a specific day.
5	getTotalNumberProductSold()	Retrieves the total number of products sold.
6	getProductQuality(int type)	Retrieves the product quality data based on the specified type

c. Sequence Diagram(s)

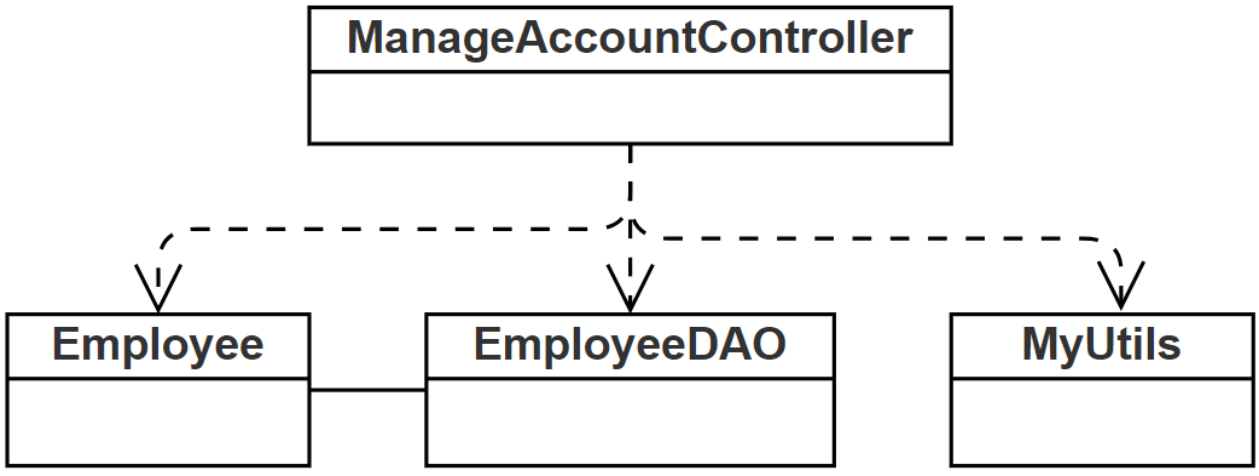


d. Database Queries

- getProductRate()
- select p.name ,ROUND(cast(sum(r.star_rate)as float)/cast(count(r.id) as float),2) as star from product p join
- order_detail od on od.product_id = p.id join
- rate r on r.order_detail_id = od.id
- group by p.id,p.name
- order by star desc
- getTotalNumberProductSold()
- select COUNT(od.id) as NumberSold from [order] o join
- order_detail od on o.id=od.order_id
- where o.order_status=2
- getOrderAdminStatistic()
- select TOP 5 o.order_date, o.id, c.firstname+' '+c.lastname as fullname, sum(od.price) as total, o.paid_date from [order] o join
- order_detail od on o.id=od.order_id join
- customer c on o.customer_id=c.id
- group by o.id, o.order_date, c.firstname, c.lastname, o.paid_date
- order by o.order_date desc
- getNumberProductSoldByDay(day)
    - select o.order_date , COUNT(od.id) as NumberSold from [order] o join
    - order_detail od on o.id=od.order_id
    - where o.order_status=2 and o.order_date=?
    - group by o.order_date

3.3.2 Manage Account

a. Class Diagram



b. Class Specifications

controller.ManageNews.java

No	Method	Description
	doGet(HttpServletRequest request, HttpServletResponse response)	Handles HTTP GET requests; fetches news categories and existing news for editing, then forwards to the `AddDiscount.jsp` view.
	doPost(HttpServletRequest request, HttpServletResponse response)	Handles HTTP POST requests; processes form submissions for adding, updating, or deleting news based on the provided action.

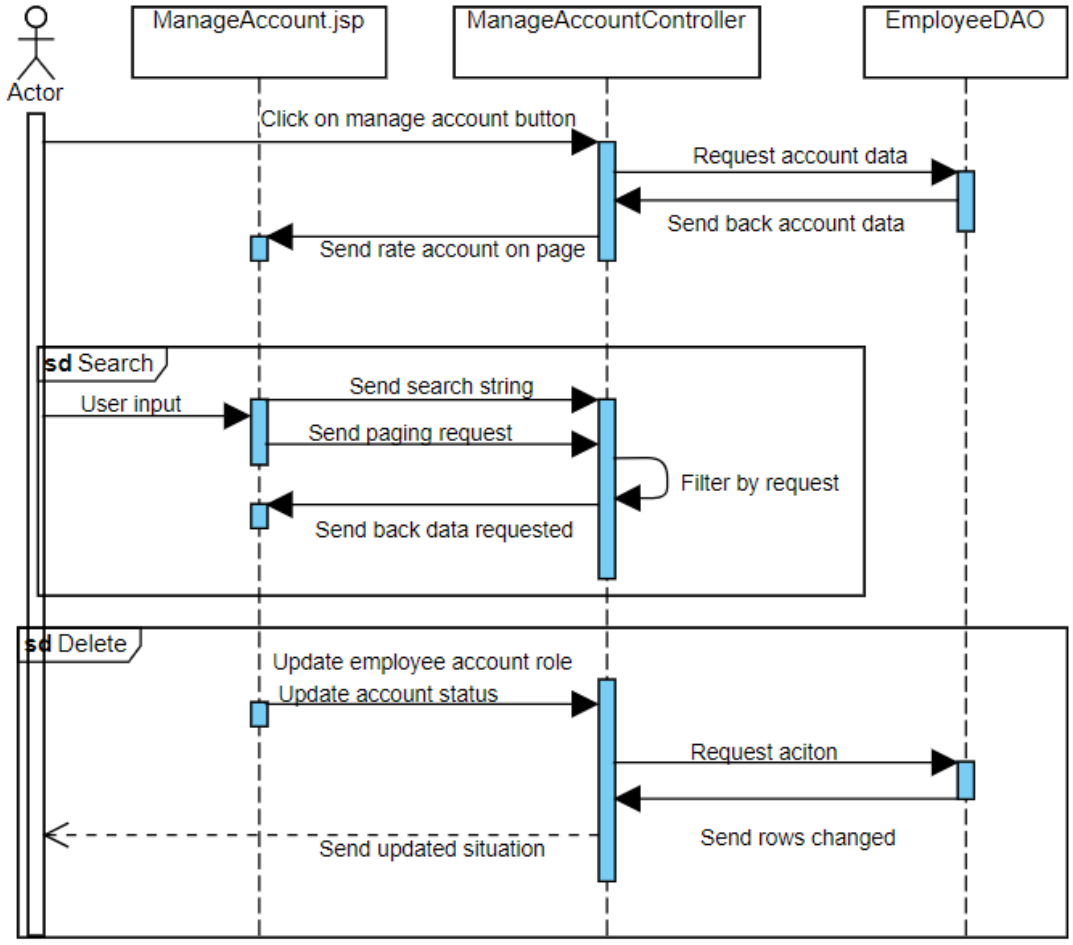
NewsDAO

No	Method	Description
1	getAll()	Retrieves all employee records from the database
2	updateEmployeeRole(int id, int roleId)	Updates the role of an employee in the database.
3	updateEmployeeStatus(int id, int statusID)	Updates the status of an employee in the database.

MyUtils

No	Method	Description
1	setAlertAttributes(HttpServletRequest request, boolean status, String action)	sets an alert message and type based on the success or failure of an action, which can be used for displaying user feedback in a web application.
2	getArrayListByPaging(ArrayList<T> list, int pageNumber, int ItemsOfPage)	calculates the start and end indices for the desired page and returns the corresponding sublist.

c. Sequence Diagram(s)



d. Database Queries

- updateEmployeeRole(int id, int roleId)
    - UPDATE employee SET role_id = ? WHERE id = ?
- updateEmployeeStatus(int id, int statusID)
    - UPDATE employee SET [status] = ? WHERE id = ?