

**MOTORCYCLE ACCESSORY**

**PROJECT**

**Software Design Specification**

**Nguyễn Thái Thịnh**

**Lead Software Engineer**

– SPRING 2023 –

Record of changeS

|  |  |  |  |
| --- | --- | --- | --- |
| Date | A\* M, D | In charge | Change Description |
| 1/7/2023 | A | Nguyễn Thái Thịnh, Cát Bá Lân, Nguyễn Xuân Thắng, Nguyễn Quang Minh,  Trần Sỹ Cao Thiên | Iitialize key entries for SDS |
| 6/7/2023 | A | Nguyễn Thái Thịnh, Cát Bá Lân, Nguyễn Xuân Thắng, Nguyễn Quang Minh,  Trần Sỹ Cao Thiên | Basic input of fields for class diagram, class specification |
| 9/7/2023 | M | Nguyễn Thái Thịnh, Cát Bá Lân, Nguyễn Xuân Thắng, Nguyễn Quang Minh,  Trần Sỹ Cao Thiên | Edit and update more for individual entries |
| 11/7/2023 | M | Nguyễn Thái Thịnh, Cát Bá Lân, Nguyễn Xuân Thắng, Nguyễn Quang Minh,  Trần Sỹ Cao Thiên | Correct basic errors and complete the document |

\*A - Added M - Modified D - Deleted

**Table of Contents**

[I. Overview 4](#_Toc140098109)

[1. Code Packages 4](#_Toc140098110)

[2. Database Design 4](#_Toc140098111)

[a. Database Schema 5](#_Toc140098112)

[b. Table Description 6](#_Toc140098113)

[II. Code Designs 11](#_Toc140098114)

[1. Customer 11](#_Toc140098115)

[a. Class Diagram 11](#_Toc140098116)

[b. Class Specifications 12](#_Toc140098117)

[c. Sequence Diagram(s) 16](#_Toc140098118)

[2. Public Order 25](#_Toc140098119)

[a. . Class Diagram 25](#_Toc140098120)

[b.Class Specifictions 26](#_Toc140098121)

[c. Sequence Diagram(s) 28](#_Toc140098122)

[3. Public Product 34](#_Toc140098123)

[a. Class Diagram 34](#_Toc140098124)

[b. Class Specifications 35](#_Toc140098125)

[4. Public News 44](#_Toc140098126)

[a. Class Diagram 44](#_Toc140098127)

[b. Class Specifications 45](#_Toc140098128)

[c. Sequence Diagram(s) 47](#_Toc140098129)

[5. Coupon management 56](#_Toc140098130)

[a. Class diagram 56](#_Toc140098131)

[b. Class Specifications 57](#_Toc140098132)

[c. Sequence Diagram(s) 59](#_Toc140098133)

[6. Admin 63](#_Toc140098134)

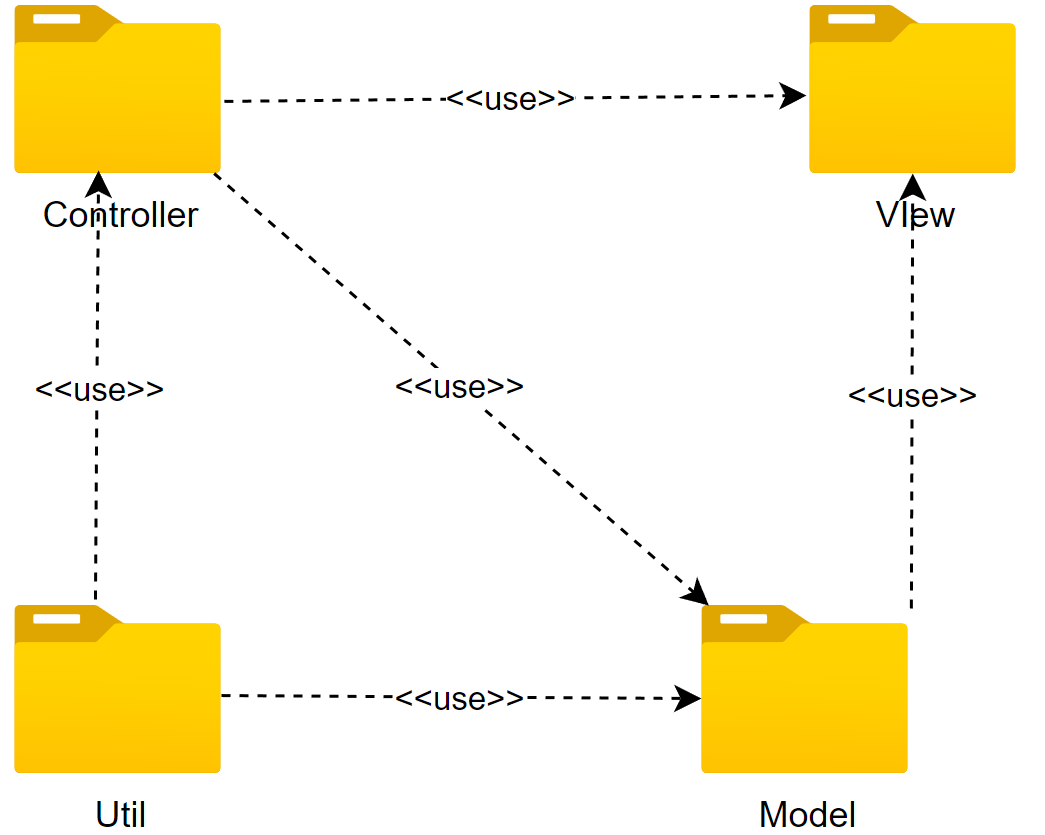
[a. Class Diagram 63](#_Toc140098135)

[b.Class Specifictions 64](#_Toc140098136)

[c. Sequence Diagram(s) 68](#_Toc140098137)

# I. Overview

## 1. Code Packages

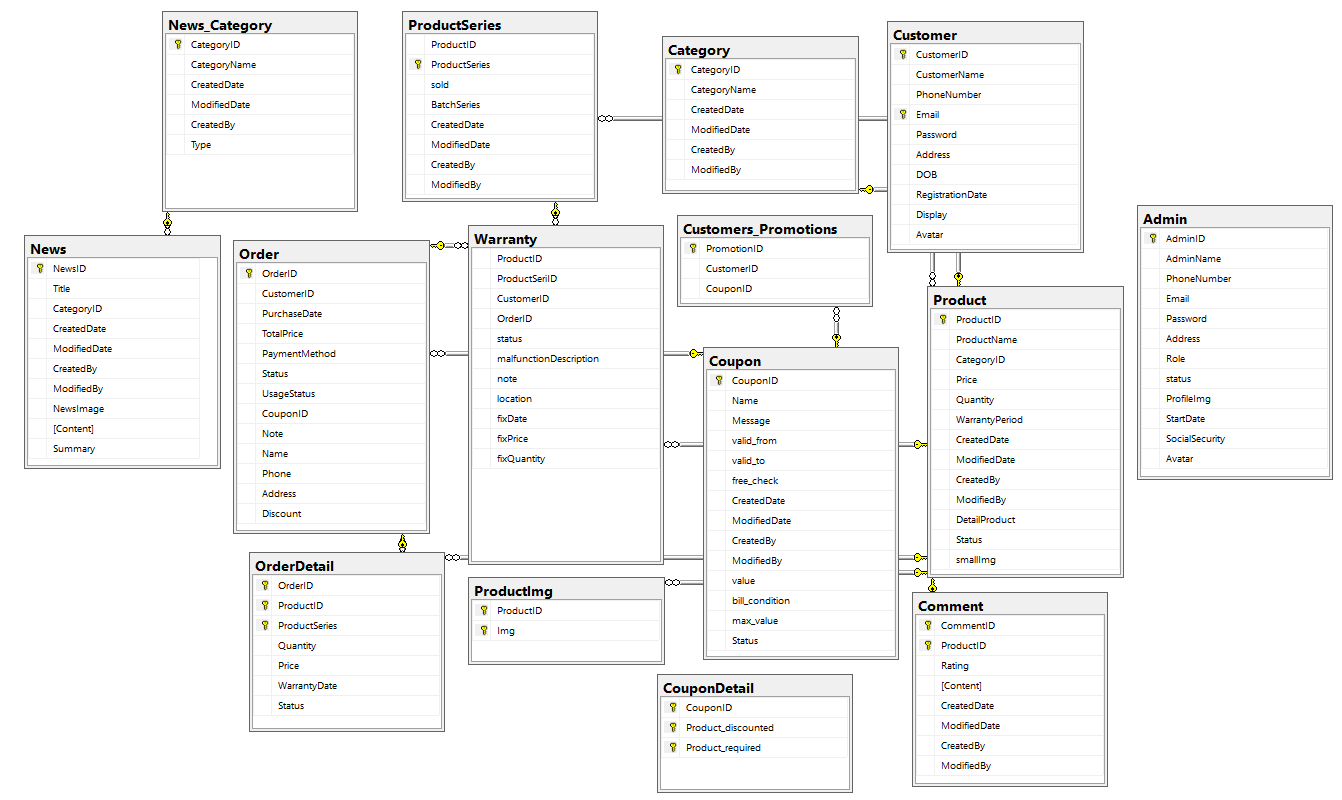


***Package descriptions***

|  |  |  |
| --- | --- | --- |
| **No** | **Package** | **Description** |
| 1 | Model | Model package helps connect to database to get, process and save data. Also, this package allow to access enities to process function of application. |
| 2 | Controller | Controller package contains APIs, servlet files to process data and show in the website. This package use for Admin site. |
| 3 | VIew | View package contains inter |
| 4 | Util | Contains common functionality and support tools used throughout the application. Utilities can include utility functions, helper classes, string handler functions, etc. |

## 2. Database Design

### a. Database Schema



### 

### b. Table Description

Category:

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **No** | **Field name** | **Type** | **Size** | **Unique** | **Not null** | **PK/ FK** | **Notes** |
| 1 | CategoryID | bigint |  | x |  | x | PK |
| 2 | CategoryName | nvarchar | 100 |  | x |  |  |
| 3 | CreatedDate | datetime | 100 |  | x |  |  |
| 4 | ModifiedDate | datetime |  |  |  |  |  |
| 5 | CreatedBy | bigint |  |  |  |  |  |
| 6 | ModifiedBy | bigint |  |  |  |  |  |

Customer:

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **No** | **Field name** | **Type** | **Size** | **Unique** | **Not null** | **PK/ FK** | **Notes** |
| 1 | CustomerID | bigint |  | x |  | x | PK |
| 2 | CustomerName | nvarchar | 50 |  | X |  |  |
| 3 | PhoneNumber | varchar | 10 |  |  |  |  |
| 4 | Email | nvarchar | 50 | x | X |  | FK |
| 5 | Password | nvarchar | 255 |  | X |  |  |
| 6 | Address | nvarchar | 255 |  |  |  |  |
| 7 | DOB | date | 50 |  |  |  |  |
| 8 | RegistrationDate | datetime |  |  | X |  |  |
| 9 | Display | bit |  |  | x |  |  |

Customer promotion:

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **No** | **Field name** | **Type** | **Size** | **Unique** | **Not null** | **PK/ FK** | **Notes** |
| 1 | PromotionID | bigint |  | x |  | x | PK |
| 2 | CustomerID | bigint |  |  |  |  |  |
| 3 | CouponID | bigint |  |  |  |  |  |

Admin:

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **No** | **Field name** | **Type** | **Size** | **Unique** | **Not null** | **PK/ FK** | **Notes** |
| 1 | AdminId | int |  | x |  | x | PK |
| 2 | AdminName | nvarchar | 50 |  | x |  |  |
| 3 | PhoneNumber | nvarchar | 10 |  | x |  |  |
| 5 | Email | nvarchar | 50 |  | x |  |  |
| 6 | Password | nvarchar | 255 |  | x |  |  |
| 7 | Address | nvarchar | 255 |  |  |  |  |
| 8 | Role | Int |  |  | x |  |  |
| 9 | Status | int |  |  | x |  |  |

News:

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **No** | **Field name** | **Type** | **Size** | **Unique** | **Not null** | **PK/ FK** | **Notes** |
| 1 | NewsID | bigint |  | x |  | x | PK |
| 2 | Title | nvarchar | 255 |  |  |  |  |
| 3 | CategoryID | datetime |  |  | x | x | FK |
| 4 | CreatedDate | datetime |  |  |  |  |  |
| 5 | ModifiedDate | datetime |  |  |  |  |  |
| 6 | CreatedBy | bigint |  |  |  |  |  |
| 7 | ModifiedBy | bigint |  |  |  |  |  |
| 8 | NewsImage | nvarchar | MAX |  |  |  |  |
| 9 | Content | nvarchar | MAX |  |  |  |  |

News\_Category:

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **No** | **Field name** | **Type** | **Size** | **Unique** | **Not null** | **PK/ FK** | **Notes** |
| 1 | CategoryID | bigint |  | x |  | x | PK |
| 2 | CategoryName | nvarchar | 100 |  | x |  |  |
| 3 | CreatedDate | datetime |  |  |  |  |  |
| 4 | ModifiedDate | datetime |  |  |  |  |  |
| 5 | CreatedBy | bigint |  |  |  |  |  |
| 6 | ModifiedBy | bigint |  |  |  |  |  |

Order:

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **No** | **Field name** | **Type** | **Size** | **Unique** | **Not null** | **PK/ FK** | **Notes** |
| 1 | OrderID | bigint |  | x |  | x | PK |
| 2 | CustomerID | bigint |  |  | x | x | FK |
| 3 | PurchaseDate | datetime |  |  | X |  |  |
| 4 | TotalPrice | float |  |  | X |  |  |
| 5 | PaymentMethod | bit |  |  | X |  |  |
| 6 | Status | int |  |  | X |  |  |
| 7 | UsageStatus | bit |  |  | X |  |  |
| 8 | CouponID | bigint |  |  |  |  |  |
| 9 | Note | nvarchar | Max |  |  |  |  |
| 10 | Name | nvarchar | Max |  |  |  |  |
| 11 | Phone | nvarchar | Max |  |  |  |  |
| 12 | Address | nvarchar | Max |  |  |  |  |
| 13 | Discount | int |  |  |  |  |  |

OrderDetail:

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **No** | **Field name** | **Type** | **Size** | **Unique** | **Not null** | **PK/ FK** | **Notes** |
| 1 | OrderID | int |  | x |  | x | PK |
| 2 | ProductID | int |  |  |  | x | FK |
| 3 | Quantity | int |  |  |  |  |  |
| 4 | Price | int |  |  |  |  |  |
| 5 | WarrantyDate | float |  |  |  |  |  |
| 6 | Status | int |  |  | x |  |  |

Product:

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **No** | **Field name** | **Type** | **Size** | **Unique** | **Not null** | **PK/ FK** | **Notes** |
| 1 | ProductID | bigint |  | x |  | x | PK |
| 2 | ProductName | nvarchar | 50 |  | x |  |  |
| 3 | CategoryID | bigint |  |  | X | x | FK |
| 4 | Price | float |  |  | X |  |  |
| 5 | Quantity | nvarchar |  |  | X |  |  |
| 6 | WarrantyPeriod | float |  |  | X |  |  |
| 7 | CreatedDate | datetime |  |  |  |  |  |
| 8 | ModifiedDate | datetime |  |  |  |  |  |
| 9 | CreatedBy | bigint |  |  |  |  |  |
| 10 | ModifiedBy | bigint |  |  |  |  |  |
| 11 | DetailProduct | nvarchar | Max |  |  |  |  |
| 12 | Status | int |  |  |  |  |  |
| 13 | smallImg | nvarchar | Max |  |  |  |  |

ProductImg:

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **No** | **Field name** | **Type** | **Size** | **Unique** | **Not null** | **PK/ FK** | **Notes** |
| 1 | ProductID | bigint |  | x |  | x | PK |
| 2 | Img | nvarchar | 200 |  | X |  |  |

Warranty:

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **No** | **Field name** | **Type** | **Size** | **Unique** | **Not null** | **PK/ FK** | **Notes** |
| 1 | ProductID | bigint |  | x |  | x | PK |
| 2 | CustomerID | bigint |  |  | X |  |  |
| 3 | WarrantyDate | datetime |  |  |  |  |  |
| 4 | Note | nvarchar | MAX |  |  |  |  |

Comment:

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **No** | **Field name** | **Type** | **Size** | **Unique** | **Not null** | **PK/ FK** | **Notes** |
| 1 | CommentID | bigint |  | x |  | x | PK |
| 2 | ProductID | bigint |  |  | x |  |  |
| 3 | Rating | int |  |  | x |  |  |
| 5 | Content | nvarchar | MAX |  | x |  |  |
| 6 | CreatedDate | datetime |  |  |  |  |  |
| 7 | ModifiedDate | datetime |  |  |  |  |  |
| 8 | CreatedBy | bigint |  |  |  |  |  |
| 9 | ModifiedBy | bigint |  |  |  |  |  |

Coupon:

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **No** | **Field name** | **Type** | **Size** | **Unique** | **Not null** | **PK/ FK** | **Notes** |
| 1 | CouponID | bigint |  | x |  | x | PK |
| 2 | Name | nvarchar | 255 |  | x |  |  |
| 3 | Message | text |  |  | x |  |  |
| 5 | valid\_from | datetime |  |  | x |  |  |
| 6 | valid\_to | datetime |  |  | x |  |  |
| 7 | free\_check | bit |  |  | X |  |  |
| 8 | CreatedDate | datetime |  |  |  |  |  |
| 9 | ModifiedDate | datetime |  |  |  |  |  |
| 10 | CreatedBy | bigint |  |  |  |  |  |
| 11 | ModifiedBy | bigint |  |  |  |  |  |
| 12 | value | nvarchar | 20 |  | x |  |  |
| 13 | bill\_condition | bigint |  |  |  |  |  |
| 14 | max\_value | bigint |  |  |  |  |  |
| 15 | Status | int |  |  |  |  |  |

Coupon detail:

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **No** | **Field name** | **Type** | **Size** | **Unique** | **Not null** | **PK/ FK** | **Notes** |
| 1 | CouponID | bigint |  | x |  | x | PK |
| 2 | Product\_discounted | bigint |  |  | X |  |  |
| 3 | Product\_required | bigint |  |  | x |  |  |

# II. Code Designs

## 1. Customer

### a. Class Diagram



### b. Class Specifications

#### CRUD\_Customer

|  |  |  |
| --- | --- | --- |
| **No** | **Method** | **Description** |
| 01 | Protected void doPost(HttpServletRequest req, HttpServletResponse resp) throws ServletException, IOException | Handle request update Customer information then respond a list Customer |
| 02 | Protected void doGet(HttpServletRequest req, HttpServletResponse resp) throws ServletException, IOException | Handle request get Customer detail or delete that Customer. If get Customer detail, it will send request to Detail site, else return new list Customer |

#### CustomerController

|  |  |  |
| --- | --- | --- |
| **No** | **Method** | **Description** |
| 01 | Protected void doPost(HttpServletRequest req, HttpServletResponse resp) throws ServletException, IOException | Handle request return List Customer order by condition |
| 02 | Protected void doGet(HttpServletRequest req, HttpServletResponse resp) throws ServletException, IOException | Handle request get Customer List |

#### LoginController

|  |  |  |
| --- | --- | --- |
| **No** | **Method** | **Description** |
| 01 | Protected void doPost(HttpServletRequest req, HttpServletResponse resp) throws ServletException, IOException | Handle request Log in then get data of that Customer |
| 02 | Protected void doGet(HttpServletRequest req, HttpServletResponse resp) throws ServletException, IOException | Handle request log out |
| 03 | public String getEncryptedPassword(String input) throws NoSuchAlgorithmException | Handle request encryte Password |
| 04 | public boolean verifyPassword(String input, String verify) throws NoSuchAlgorithmException | Handle request verify Password |

#### OrderController

|  |  |  |
| --- | --- | --- |
| **No** | **Method** | **Description** |
| 01 | Protected void doPost(HttpServletRequest req, HttpServletResponse resp) throws ServletException, IOException | Handle empty request |
| 02 | Protected void doGet(HttpServletRequest req, HttpServletResponse resp) throws ServletException, IOException | Handle request get Customer Order List and Customer information |

#### RegisterController

|  |  |  |
| --- | --- | --- |
| **No** | **Method** | **Description** |
| 01 | Protected void doPost(HttpServletRequest req, HttpServletResponse resp) throws ServletException, IOException | Handle request regist a new Customer |
| 02 | Protected void doGet(HttpServletRequest req, HttpServletResponse resp) throws ServletException, IOException | Handle empty request |
| 03 | public String getEncryptedPassword(String input) throws NoSuchAlgorithmException | Handle request encryte Password |
| 04 | public boolean verifyPassword(String input, String verify) throws NoSuchAlgorithmException | Handle request verify Password |

#### CustomerSearch

|  |  |  |
| --- | --- | --- |
| **No** | **Method** | **Description** |
| 01 | Protected void processRequest (HttpServletRequest req, HttpServletResponse resp) throws ServletException, IOException | Handle request print out html |

#### CustomerController (Client)

|  |  |  |
| --- | --- | --- |
| **No** | **Method** | **Description** |
| 01 | Protected void doPost(HttpServletRequest req, HttpServletResponse resp) throws ServletException, IOException | Handle request update Customer information |
| 02 | Protected void doGet(HttpServletRequest req, HttpServletResponse resp) throws ServletException, IOException | Handle request get Customer Detail |

#### LoginCheck

|  |  |  |
| --- | --- | --- |
| **No** | **Method** | **Description** |
| 01 | Protected void doPost(HttpServletRequest req, HttpServletResponse resp) throws ServletException, IOException | Handle empty request |
| 02 | Protected void doGet(HttpServletRequest req, HttpServletResponse resp) throws ServletException, IOException | Handle request add Customer Account login into cookie (Remember me) |

#### LoginGG

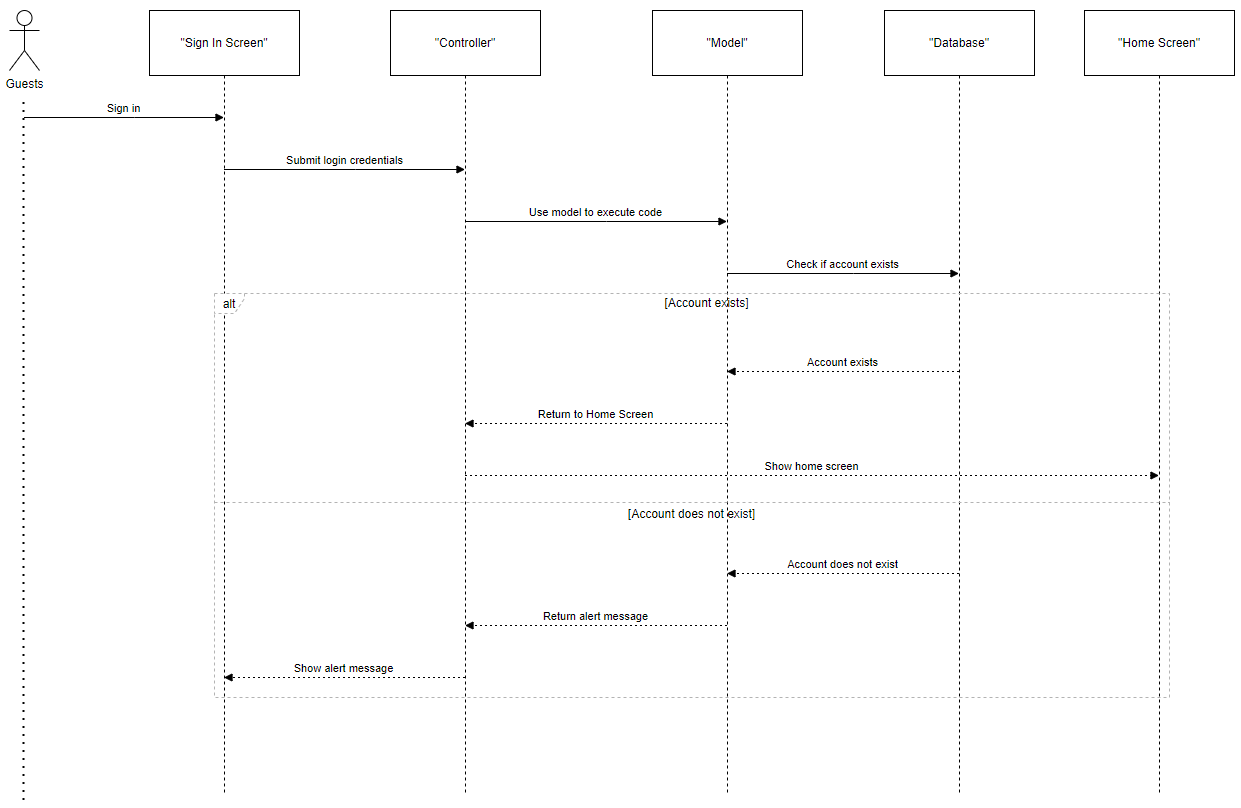
|  |  |  |
| --- | --- | --- |
| **No** | **Method** | **Description** |
| 01 | Protected void doPost(HttpServletRequest req, HttpServletResponse resp) throws ServletException, IOException | Handle empty request |
| 02 | Protected void doGet(HttpServletRequest req, HttpServletResponse resp) throws ServletException, IOException | Handle request login and regist with google account |
| 03 | public static String getToken(String code) throws ClientProtocolException, IOException | Handle request get token given by Google |
| 04 | public String getEncryptedPassword(String input) throws NoSuchAlgorithmException | Handle request encryte Password |
| 05 | public boolean verifyPassword(String input, String verify) throws NoSuchAlgorithmException | Handle request verify Password |
| 06 | public static UserGoogleDto getUserInfo(final String accessToken) throws ClientProtocolException, IOException | Handle request get Information of account given by Google |

#### Customer

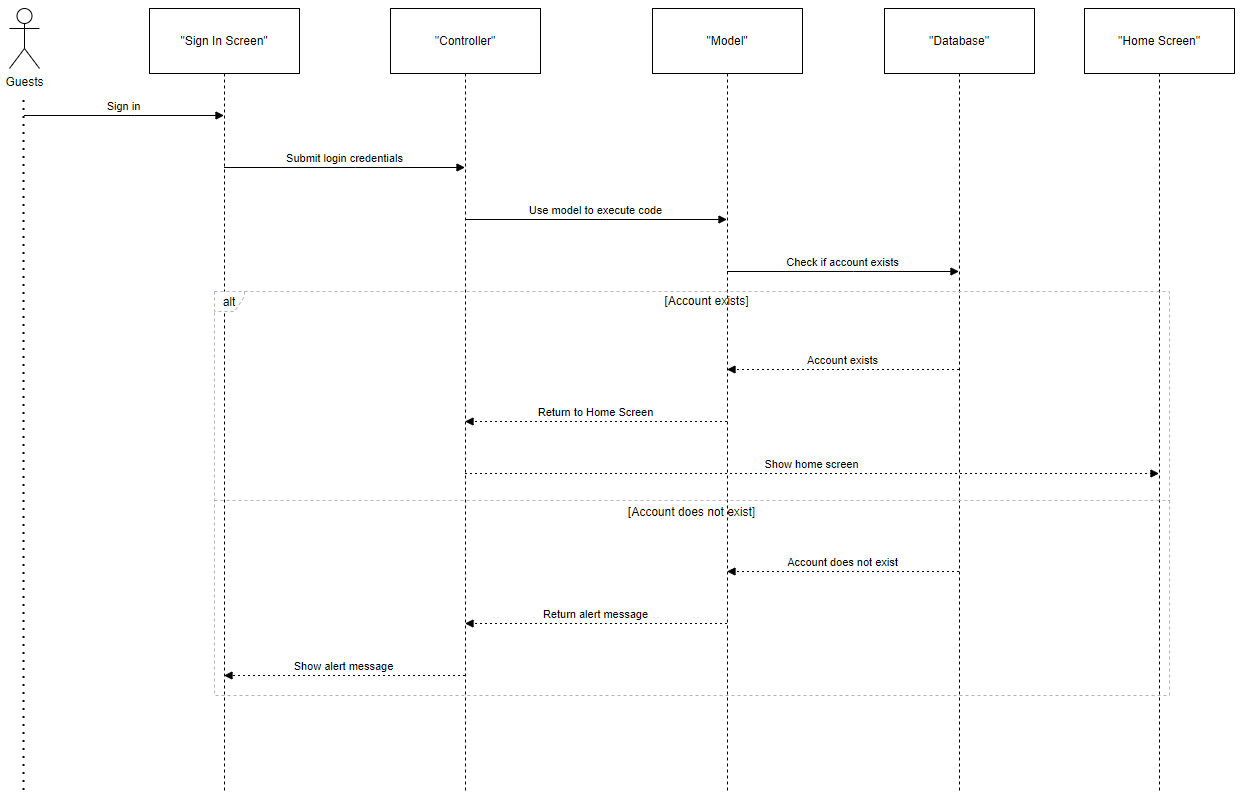
|  |  |  |
| --- | --- | --- |
| **No** | **Method** | **Description** |
| 01 | public ArrayList<Customer> getListCustomer() | Handle request get Customer List |
| 02 | public ArrayList<Customer> getListCustomerbyName(String name) | Handle request get Customer List by Customer Name |
| 03 | public boolean checkCustomerExist() | Handle request check if Customer already exist |
| 04 | public void register(String name, String email, String password, String dob, String phone, String address, String registrationDate) | Handle request regist a new Customer |
| 05 | public ArrayList<Customer> getListOrderNewCustomer() | Handle request return a Customer list sort by Registration date |
| 06 | public void deleteThisCustomerId(String ID) | Handle request delete that Customer by Customer ID |
| 07 | public Customer getCustomerByID(String ID) | Handle request return Customer Detail by CustomerID |
| 08 | public boolean update(String id, String name, String phone, String email, String adress, String dob, String redate) | Handle request update Customer Detail by CustomerID |
| 09 | public boolean checkCustomer() | Handle request check Customer Account and Password match |
| 10 | public String getEncryptedPassword(String input) throws NoSuchAlgorithmException | Handle request encryte Password |
| 11 | public boolean verifyPassword(String input, String verify) throws NoSuchAlgorithmException | Handle request verify Password |
| 12 | public Customer getCustomer() | Handle request get Customer by Email |
| 13 | public boolean checkCustomerExist(String email) | Handle request check if Customer Email exist |
| 14 | public void register(String name, String email, String encryptedPassword, String format) | Handle request regist a new Customer by Google Account |
| 15 | public List<String> getAllAddress(int cid) | Handle request get all adress from every customers |
| 16 | public boolean update(String id, String name, String phone, String email, String address, String dob, String redate, String avatar) | Handle request update Customer Detail by CustomerID |
| 17 | public void register(String name, String email, String encryptedPassword, String format, String picture) | Handle request regist a new Customer by Google Account with new avatar |
| 18 | public List searchCustomer(int choice, String content) | Handle request get Customer list by condition |

### c. Sequence Diagram(s)

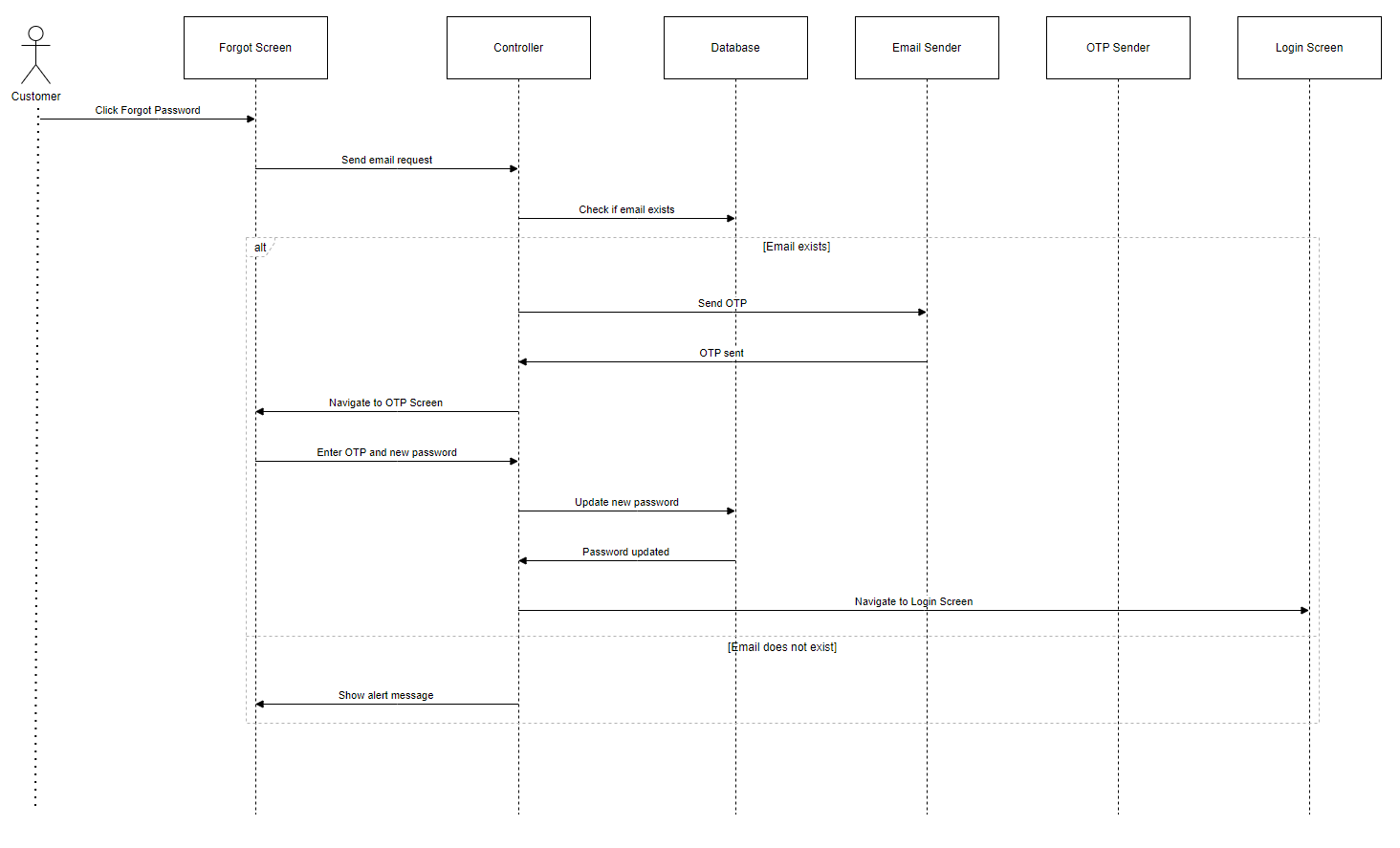
#### Login



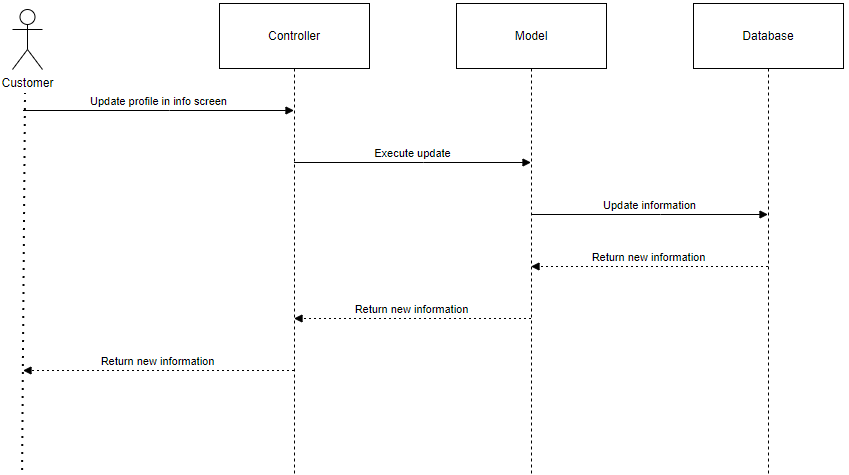
#### Signup



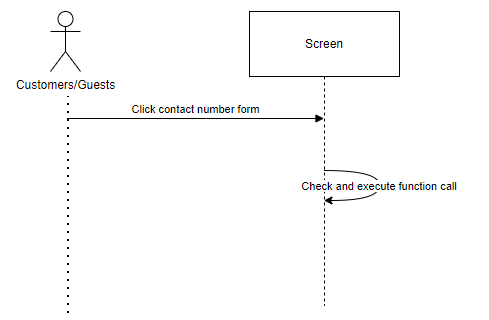
#### Forgot Password



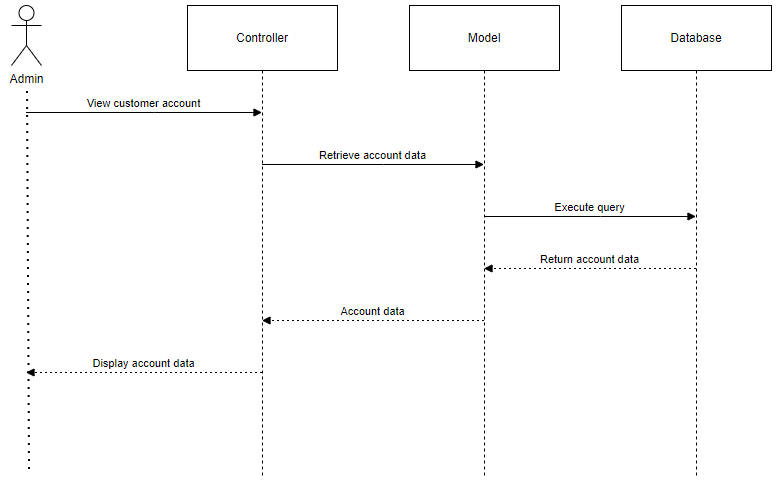
#### Update Profile



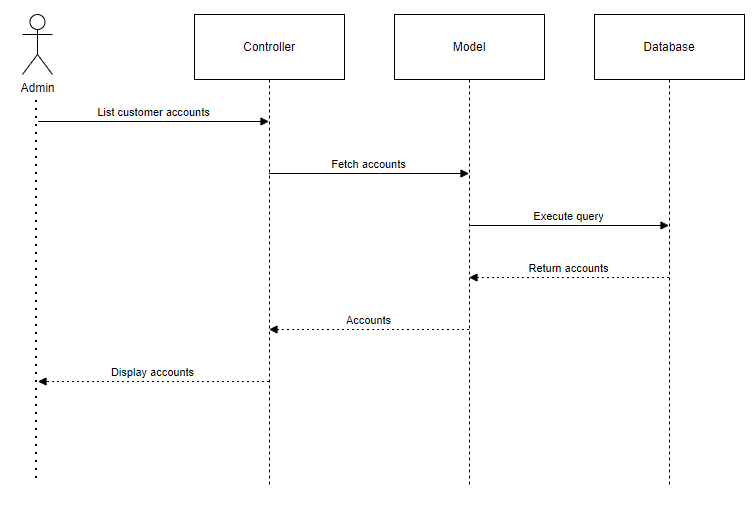
#### Click to call



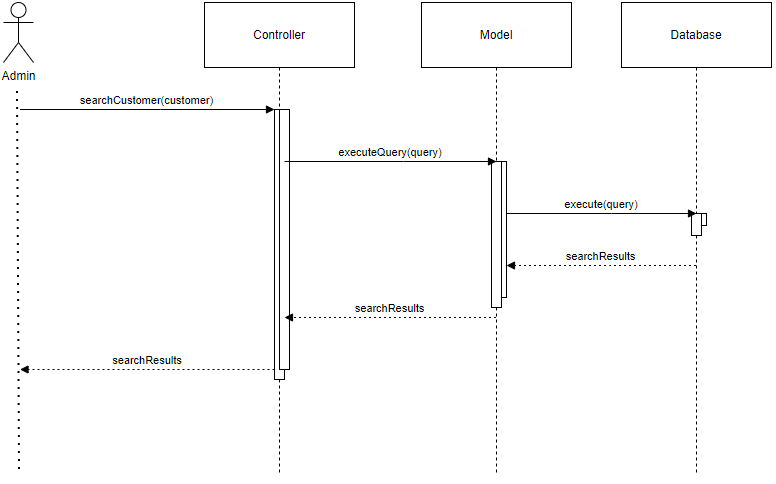
#### View customer account



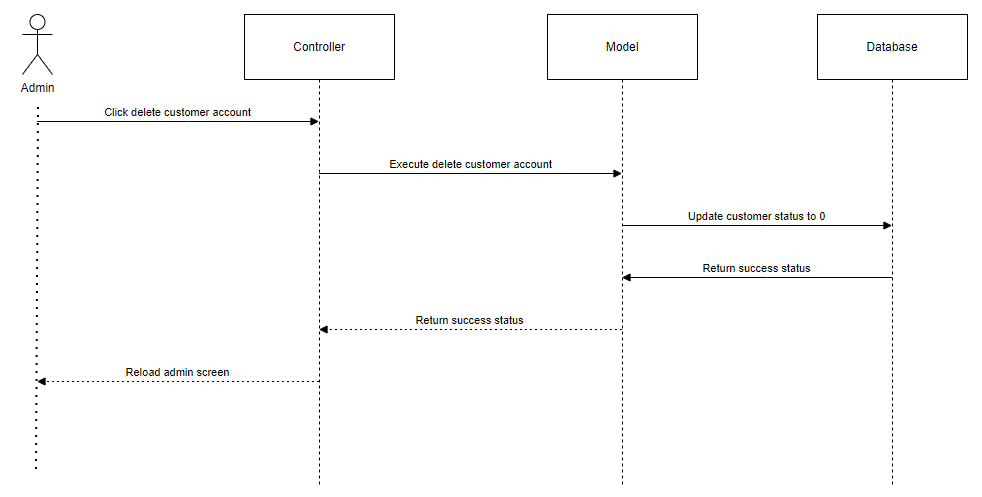
#### View all customer



#### Search customer



#### Delete customer



## 2. Public Order

### a. . Class Diagram



### b.Class Specifictions

#### Sale controller

|  |  |  |
| --- | --- | --- |
| **No** | **Method** | **Description** |
| 01 | protected void doPost(HttpServletRequest req, HttpServletResponse resp) throws ServletException, IOException | N/A |
| 02 | protected void doGet(HttpServletRequest req, HttpServletResponse resp) throws ServletException, IOException | Handle paging and search for the sale related requests |

#### SearchSaleController

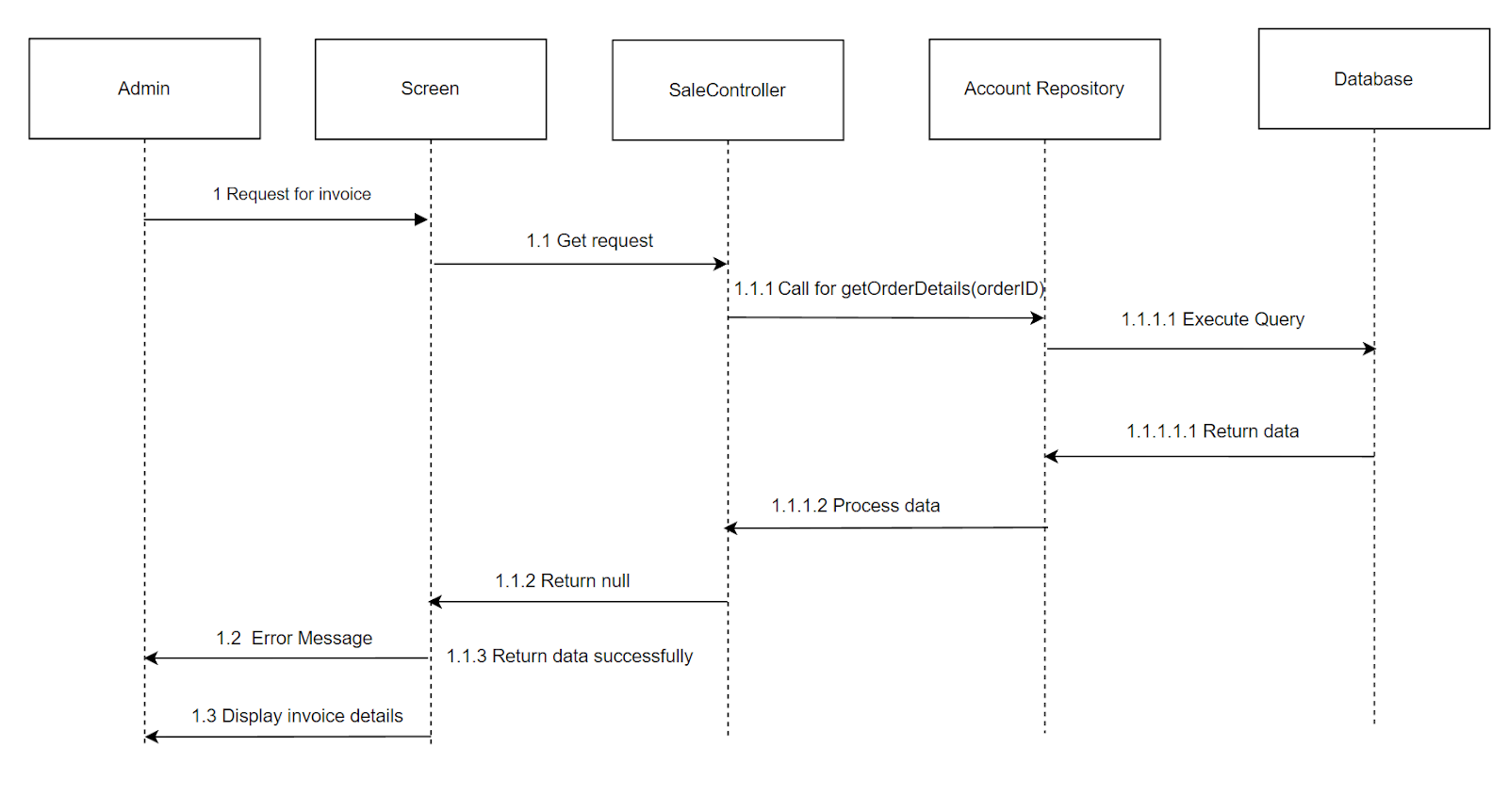
|  |  |  |
| --- | --- | --- |
| **No** | **Method** | **Description** |
| 01 | protected void doPost(HttpServletRequest req, HttpServletResponse resp) throws ServletException, IOException | N/A |
| 02 | protected void doGet(HttpServletRequest req, HttpServletResponse resp) throws ServletException, IOException | Handle paging and search for the sale related requests |

#### Order

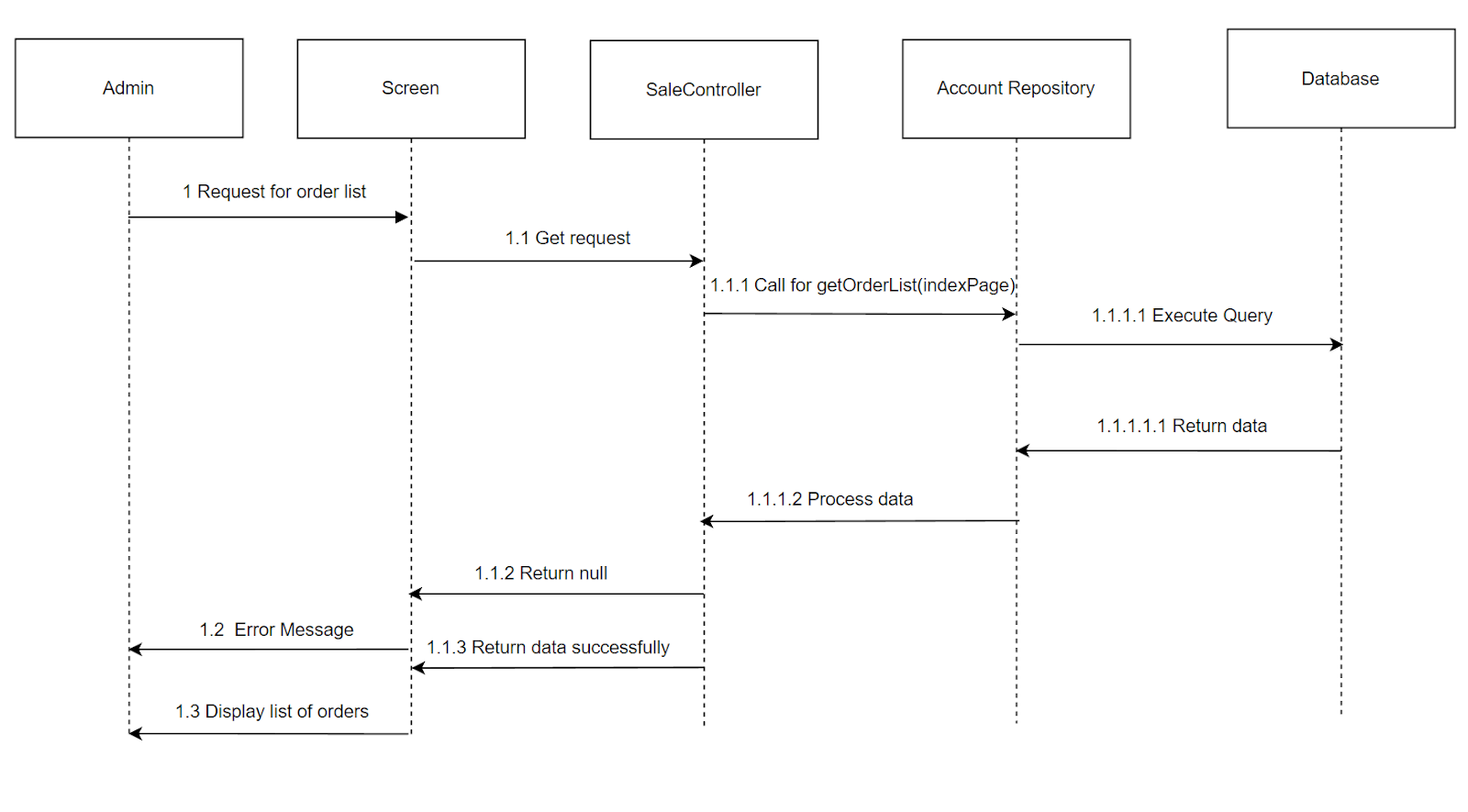
|  |  |  |
| --- | --- | --- |
| **No** | **Method** | **Description** |
| 01 | public List<Orders> getOrderList(int index) | Handle all request for list of orders in history incorporating with paging |
| 02 | public List<Orders> getOrderList() | Handle all request for list of orders in history |
| 03 | public List<Orders> getOrderListLimitted() | Handle request for list of limited number of orders |
| 04 | public List<Orders> getOrderDetails(int id) | Handle request for information of invoice |
| 05 | public int todaySale() | Handle request for the total income of the day |
| 06 | public int monthlySale() | Handle request for the total income of the month |
| 07 | public int lastYearSale() | Handle request for the total income of last year |
| 08 | public int yearSale() | Handle request for the total income of the year |
| 09 | public List<Orders> getTodayOrderList() | Handle request for the list of orders of the day |
| 10 | public List<Orders> getMonthOrderList() | Handle request for the list of orders of the month |
| 11 | public List<Orders> getLastYearOrderList() | Handle request for the list of orders of last year |
| 12 | public List<Orders> getYearOrderList() | Handle request for the list of orders of the year |
| 13 | public List<Orders> getOrderManagement(int index) | Handle request for the list of orders status incorporating with paging |
| 14 | public List<Orders> getOrderManagement() | Handle request for the list of orders |
| 15 | public String moneyconfig(int today) | Handle request for configure the currency to the correct form |
| 16 | public void updateStatus() | Handle request for updating the status of the orders |
| 17 | public List<Orders> getOrderManagement2() | Handle request for the list of orders |
| 18 | public List<Orders> getOrderByCustomerName(String search) | Handle request for the list of orders by customer name |
| 19 | public List<Orders> getOrderByOrderId(String search) | Handle request for the list of orders by order ID |
| 20 | public List<Orders> getOrderByPhone(String search) | Handle request for the list of orders by phone number |
| 21 | public List<Orders> getOrderManagementByCustomerName(String search) | Handle request for the list of orders and status by customer name |
| 22 | public List<Orders> getOrderManagementByOrderId(String search) | Handle request for the list of orders and status by order ID |
| 23 | public List<Orders> getOrderManagementByPhone(String search) | Handle request for the list of orders and status by phone |
| 24 | public int getNumberPage() | Handle request for getting the page number |
| 25 | public void createOrder() | Handle request for creating an order |

### c. Sequence Diagram(s)

#### View invoice



#### View order status management list



#### View warranty

#### Create product maintenance record

#### 

#### Manage orders

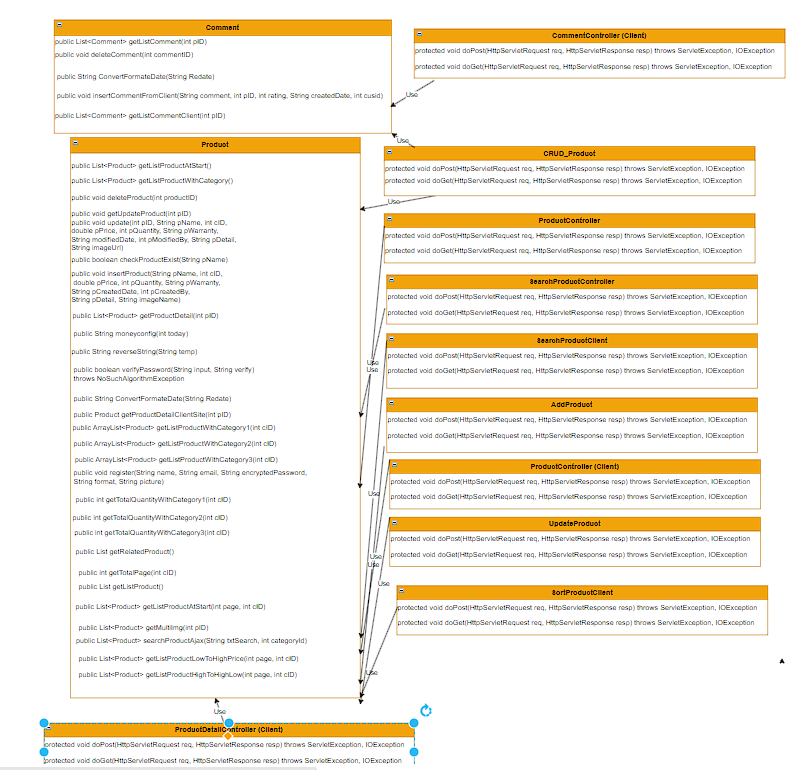
#### 

#### Search Orders/Product Maintenance History/Order Status Management

#### 

## 3. Public Product

### a. Class Diagram



### b. Class Specifications

#### CRUD\_Product

|  |  |  |
| --- | --- | --- |
| **No** | **Method** | **Description** |
| 01 | Protected void doPost(HttpServletRequest req, HttpServletResponse resp) throws ServletException, IOException | Handle request from Main to CRUD |
| 02 | Protected void doGet(HttpServletRequest req, HttpServletResponse resp) throws ServletException, IOException | Handle request get Product detail, update, delete or Show comment of that Product. If get Product detail, it will send request to Detail site, if get Product Update or Show Comment it will send request to approriate site, if delete Product, it will delete the product |

#### ProductController

|  |  |  |
| --- | --- | --- |
| **No** | **Method** | **Description** |
| 01 | Protected void doPost(HttpServletRequest req, HttpServletResponse resp) throws ServletException, IOException | Handle request return Category List to filter by Category in Product List Site |
| 02 | Protected void doGet(HttpServletRequest req, HttpServletResponse resp) throws ServletException, IOException | Handle request get Product List |

#### SearchProductController

|  |  |  |
| --- | --- | --- |
| **No** | **Method** | **Description** |
| 01 | Protected void doPost(HttpServletRequest req, HttpServletResponse resp) throws ServletException, IOException | Handle empty request |
| 02 | Protected void doGet(HttpServletRequest req, HttpServletResponse resp) throws ServletException, IOException | Handle request Search Product in Product List Site |

#### SearchProductClient

|  |  |  |
| --- | --- | --- |
| **No** | **Method** | **Description** |
| 01 | Protected void doPost(HttpServletRequest req, HttpServletResponse resp) throws ServletException, IOException | Handle request Search Product in Product List Site in Client Site |

#### ProductController (Client)

|  |  |  |
| --- | --- | --- |
| **No** | **Method** | **Description** |
| 01 | Protected void doPost(HttpServletRequest req, HttpServletResponse resp) throws ServletException, IOException | Handle request To load out Category to filter by Category in Product List Site in Client Site and also the pagination |
| 02 | Protected void doGet(HttpServletRequest req, HttpServletResponse resp) throws ServletException, IOException | Handle request Show Product List, Category, most of Information in Product List Site also the pagination in Client Site |

#### SortProductClient

|  |  |  |
| --- | --- | --- |
| **No** | **Method** | **Description** |
| 01 | Protected void doPost(HttpServletRequest req, HttpServletResponse resp) throws ServletException, IOException | Handle empty request |
| 02 | Protected void doGet(HttpServletRequest req, HttpServletResponse resp) throws ServletException, IOException | Handle request Sort the Product in ascending order or descending order |

#### ProductDetailController

|  |  |  |
| --- | --- | --- |
| **No** | **Method** | **Description** |
| 01 | Protected void doPost(HttpServletRequest req, HttpServletResponse resp) throws ServletException, IOException | Handle empty request |
| 02 | Protected void doGet(HttpServletRequest req, HttpServletResponse resp) throws ServletException, IOException | Handle request redirect to Detail Site of that selected Product and Show the Comment and Related Product in Client Site |

#### AddProduct

|  |  |  |
| --- | --- | --- |
| **No** | **Method** | **Description** |
| 01 | Protected void doPost(HttpServletRequest req, HttpServletResponse resp) throws ServletException, IOException | Handle request Create new Product in Admin Site |
| 02 | Protected void doGet(HttpServletRequest req, HttpServletResponse resp) throws ServletException, IOException | Handle request redirect to Create new Product Site in Admin Site |

#### UpdateProduct

|  |  |  |
| --- | --- | --- |
| **No** | **Method** | **Description** |
| 01 | Protected void doPost(HttpServletRequest req, HttpServletResponse resp) throws ServletException, IOException | Handle request update Product in Admin Site |
| 02 | Protected void doGet(HttpServletRequest req, HttpServletResponse resp) throws ServletException, IOException | Handle empty request |

#### Product

|  |  |  |
| --- | --- | --- |
| **No** | **Method** | **Description** |
| 01 | public List<Product> getListProductAtStart() | Handle request get Product List |
| 02 | public List<Product> getListProductWithCategory() | Handle request get Category in Filter dropdown |
| 03 | public void deleteProduct(int productID) | Handle request delete Product |
| 04 | public void getUpdateProduct(int pID) | Handle request get all information of seleted Product to Update |
| 05 | public void update(int pID, String pName, int cID, double pPrice, int pQuantity, String pWarranty, String modifiedDate, int pModifiedBy, String pDetail, String imageUrl) | Handle request update Product |
| 06 | public boolean checkProductExist(String pName) | Handle request check Product exist in database |
| 07 | public void insertProduct(String pName, int cID, double pPrice, int pQuantity, String pWarranty, String pCreatedDate, int pCreatedBy, String pDetail, String imageName) | Handle request insert new Product to database |
| 08 | public List<Product> getProductDetail(int pID) | Handle request get all information of selected Product |
| 09 | public String moneyconfig(int today) | Config, format price |
| 10 | public String reverseString(String temp) | Config date |
| 11 | public String ConvertFormateDate(String Redate) | Format date |
| 12 | public List searchProduct(String categories, String content) | Handle request search Product |
| 13 | public Product getProductByID(int pID) | Handle request get Product by ID |
| 14 | public Product getProductDetailClientSite(int pID) | Handle request get Product Detail in Client Site |
| 15 | public ArrayList<Product> getListProductWithCategory1(int cID) | Handle request getCategory1 in Client Site |
| 16 | public ArrayList<Product> getListProductWithCategory2(int cID) | Handle request getCategory2 in Client Site |
| 17 | public ArrayList<Product> getListProductWithCategory3(int cID) | Handle request getCategory3 in Client Site |
| 18 | public int getTotalQuantityWithCategory1(int cID) | Handle request get total Quantity of Category1 in client site |
| 19 | public int getTotalQuantityWithCategory2(int cID) | Handle request get total Quantity of Category2 in client site |
| 20 | public int getTotalQuantityWithCategory3(int cID) | Handle request get total Quantity of Category3 in client site |
| 21 | public List<Product> getRelatedProduct() | Handle request get all Related Product of selected Product in Client Site |
| 22 | public int getTotalPage(int cID) | Handle request get total page in pagination |
| 23 | public List<Product> getListProduct() | Handle request get Product List |
| 24 | public List<Product> getListProductAtStart(int page, int cID) | Handle request get Product List in pagination |
| 25 | public List<Product> getMultiImg(int pID) | Handle request get multi image |
| 26 | public List<Product> searchProductAjax(String txtSearch, int categoryId) | Handle request search Product with Ajax |
| 27 | public List<Product> getListProductLowToHighPrice(int page, int cID) | Handle request sort Product in ascending order in Client site |
| 28 | public List<Product> getListProductHighToHighLow(int page, int cID) | Handle request sort Product in descending order in Client Site |

#### CommentController (Client)

|  |  |  |
| --- | --- | --- |
| **No** | **Method** | **Description** |
| 01 | Protected void doPost(HttpServletRequest req, HttpServletResponse resp) throws ServletException, IOException | Handle request insert Comment in Client site to database |
| 02 | Protected void doGet(HttpServletRequest req, HttpServletResponse resp) throws ServletException, IOException | Handle empty request |

#### Comment

|  |  |  |
| --- | --- | --- |
| **No** | **Method** | **Description** |
| 01 | public List<Comment> getListComment(int pID) | Handle request get Comment List in Admin site |
| 02 | public void deleteComment(int commentID) | Handle request delete Comment from Admin site |
| 03 | public String ConvertFormateDate(String Redate) | Format Date |
| 04 | public void insertCommentFromClient(String comment, int pID, int rating, String createdDate, int cusid) | Handle request insert Comment from Client site to database |
| 05 | public List<Comment> getListCommentClient(int pID) | Handle request display all the Comment in database to Client site |

#### c. Sequence Diagram(s) Display Product List in Admin

#### 

#### Update product

#### 

#### Add new product

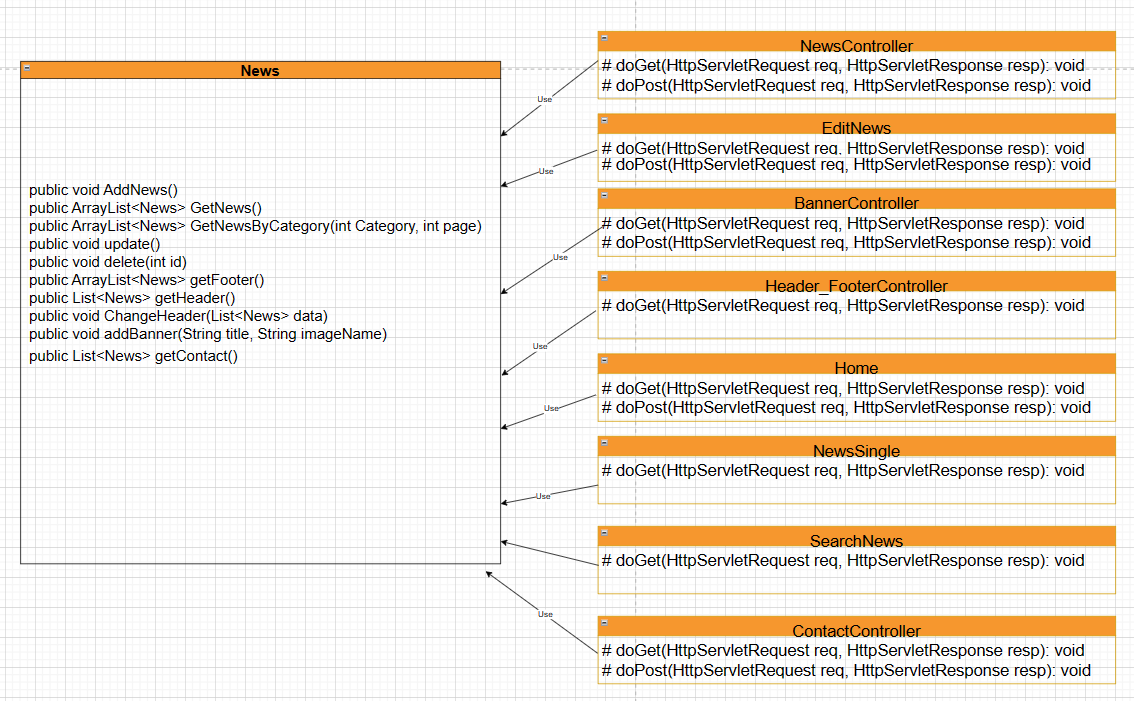
#### 

#### Display Product Detail in Admin Site

#### 

## 4. Public News

### a. Class Diagram



### b. Class Specifications

#### NewsController

|  |  |  |
| --- | --- | --- |
| **No** | **Method** | **Description** |
| 01 | Protected void doPost(HttpServletRequest req, HttpServletResponse resp) throws ServletException, IOException | Handle request add News then respond a list News |
| 02 | Protected void doGet(HttpServletRequest req, HttpServletResponse resp) throws ServletException, IOException | Handle request delete News or respond a list News |

#### EditNews

|  |  |  |
| --- | --- | --- |
| **No** | **Method** | **Description** |
| 01 | Protected void doPost(HttpServletRequest req, HttpServletResponse resp) throws ServletException, IOException | Handle request update News |
| 02 | Protected void doGet(HttpServletRequest req, HttpServletResponse resp) throws ServletException, IOException | Handle request get News detail by id. Send data to edit site |

#### Banner Controller

|  |  |  |
| --- | --- | --- |
| **No** | **Method** | **Description** |
| 01 | Protected void doPost(HttpServletRequest req, HttpServletResponse resp) throws ServletException, IOException | Handle request add Banner then respond a list Banner |
| 02 | Protected void doGet(HttpServletRequest req, HttpServletResponse resp) throws ServletException, IOException | Handle request get to addbanner site or banner management site |

#### Header FooterController

|  |  |  |
| --- | --- | --- |
| **No** | **Method** | **Description** |
| 01 | Protected void doPost(HttpServletRequest req, HttpServletResponse resp) throws ServletException, IOException | Handle request update Header/ Footer or add Header/Footer then respond a list Header and Footer |
| 02 | Protected void doGet(HttpServletRequest req, HttpServletResponse resp) throws ServletException, IOException | Handle request redirect user to edit site or management site or delete a Header/Footer |

#### Home

|  |  |  |
| --- | --- | --- |
| **No** | **Method** | **Description** |
| 01 | Protected void doGet(HttpServletRequest req, HttpServletResponse resp) throws ServletException, IOException | Handle request get Header,Footer,Banner,Contact |

#### NewsSingle

|  |  |  |
| --- | --- | --- |
| **No** | **Method** | **Description** |
| 01 | Protected void doGet(HttpServletRequest req, HttpServletResponse resp) throws ServletException, IOException | Handle request get a News by Id and display it |

#### SearchNews

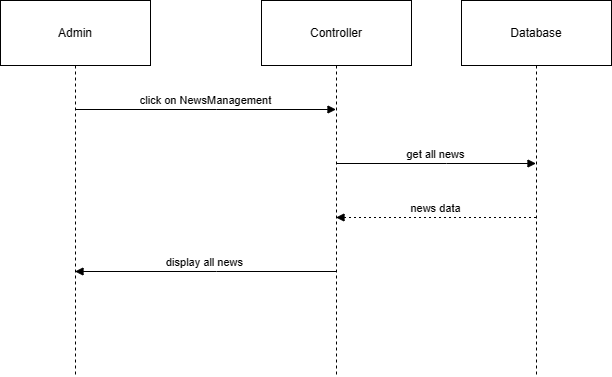
|  |  |  |
| --- | --- | --- |
| **No** | **Method** | **Description** |
| 01 | Protected void doGet(HttpServletRequest req, HttpServletResponse resp) throws ServletException, IOException | Handle request get list News by search text, categoryid |

#### ContactController

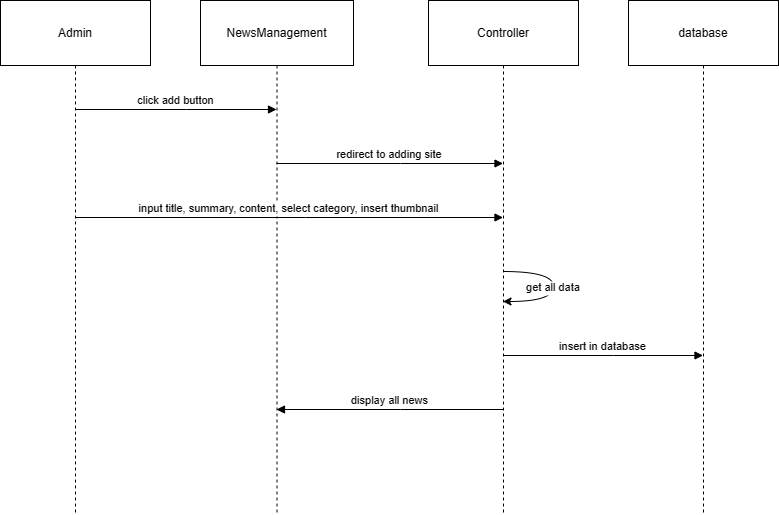
|  |  |  |
| --- | --- | --- |
| **No** | **Method** | **Description** |
| 01 | Protected void doPost(HttpServletRequest req, HttpServletResponse resp) throws ServletException, IOException | Handle request update contact information then respond a list contact |
| 02 | Protected void doGet(HttpServletRequest req, HttpServletResponse resp) throws ServletException, IOException | Handle request get list of contact |

### c. Sequence Diagram(s)

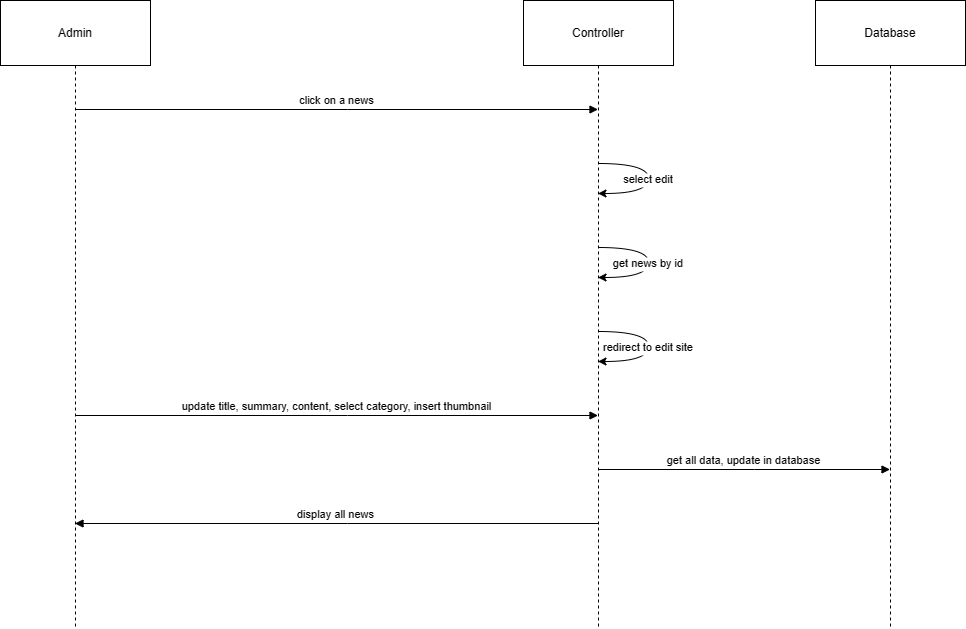
#### View list news



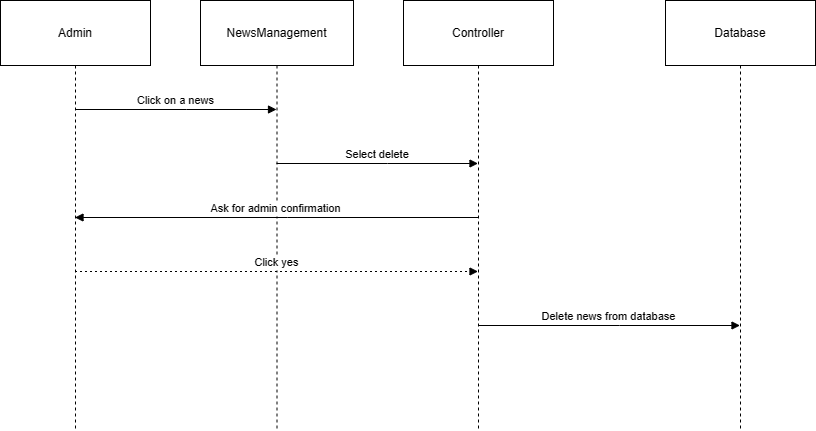
#### Add news



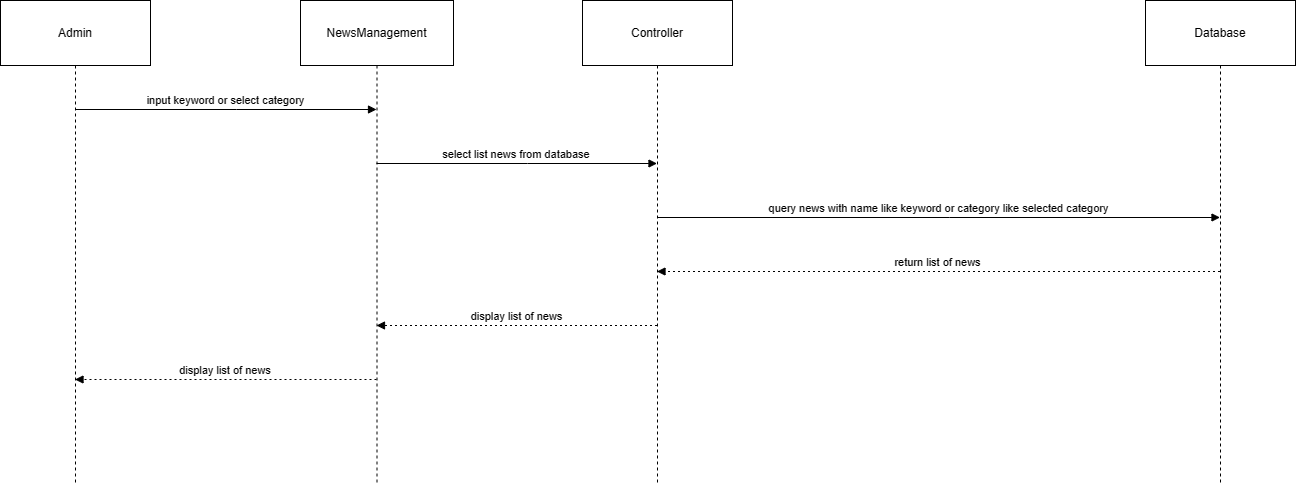
#### Edit news



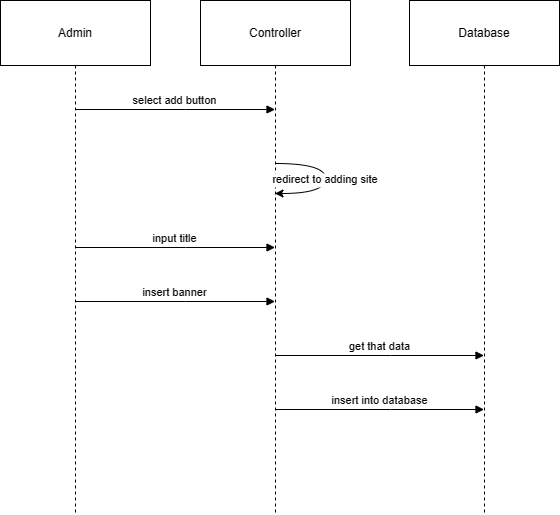
#### Delete News



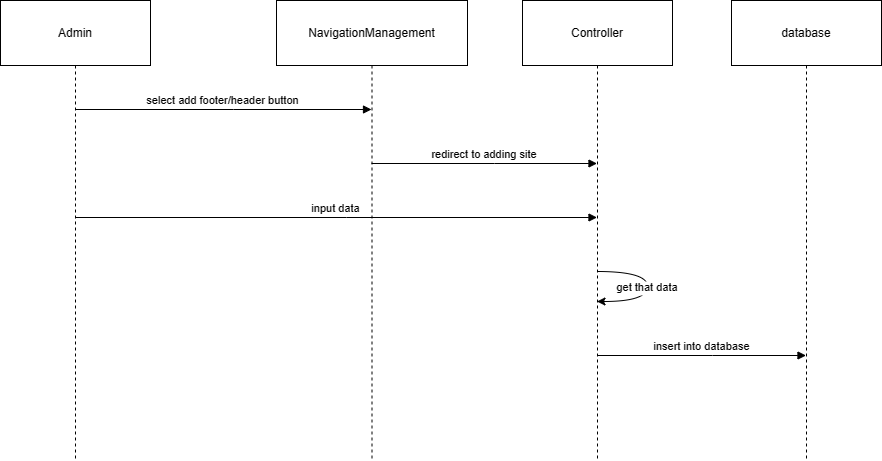
#### Search News



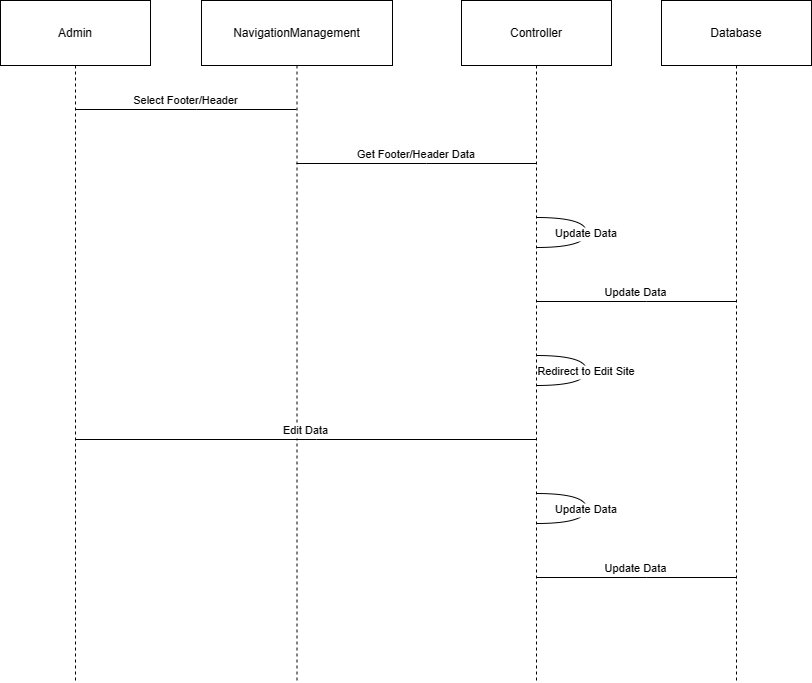
#### Add Banner



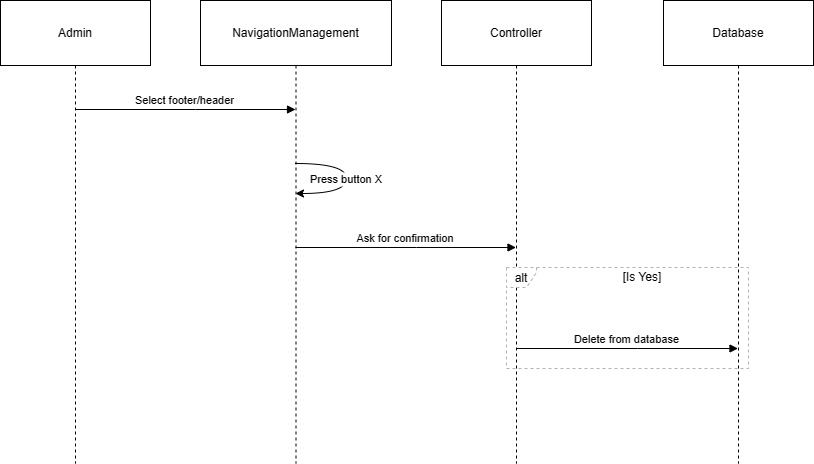
#### Add footer/header



#### Edit header or footer

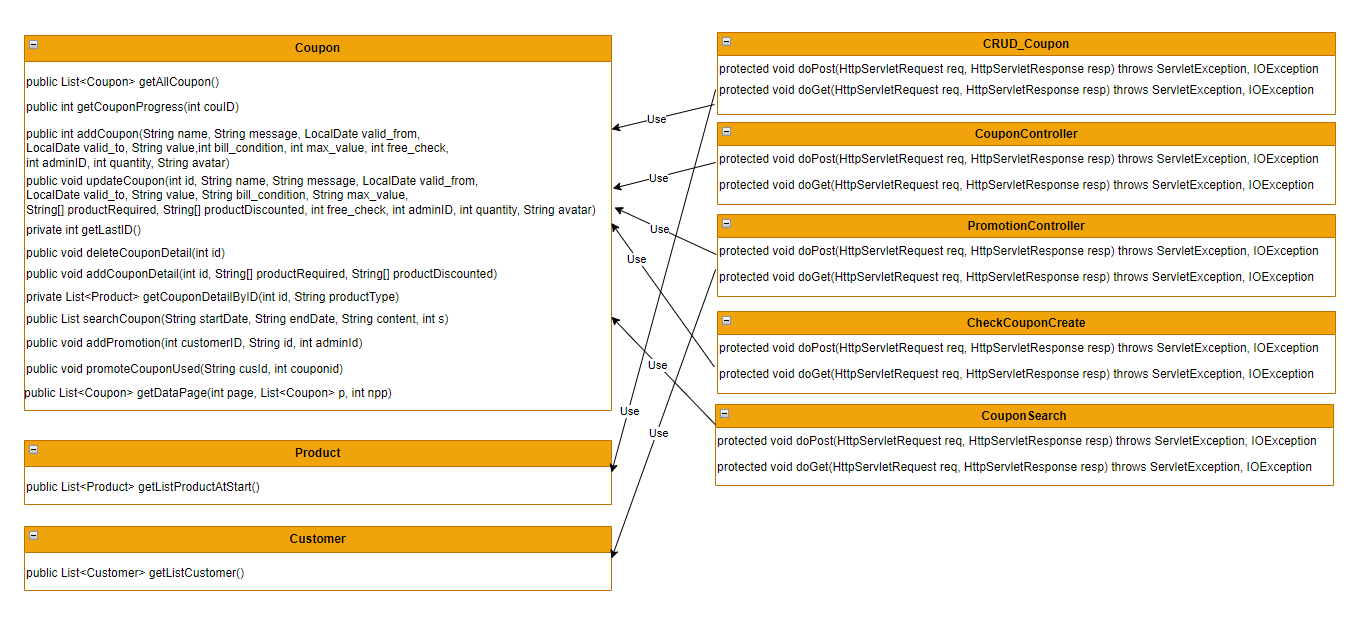


#### Delete header or footer navigation



## 5. Coupon management

### a. Class diagram



### b. Class Specifications

#### Coupon.java

|  |  |  |
| --- | --- | --- |
| **No** | **Method** | **Description** |
| 01 | public List<Coupon> getAllCoupon() | Handle request get a List of coupon |
| 02 | public int getCouponProgress(int couID) | Handle request calculate percents used of a coupon |
| 03 | public int addCoupon(String name, String message, LocalDate valid\_from, LocalDate valid\_to, String value,int bill\_condition, int max\_value, int free\_check, int adminID, int quantity, String avatar) | Handle request create new coupon |
| 04 | public void updateCoupon(int id, String name, String message, LocalDate valid\_from, LocalDate valid\_to, String value, String bill\_condition, String max\_value, String[] productRequired, String[] productDiscounted, int free\_check, int adminID, int quantity, String avatar) | Handle request update a given coupon |
| 05 | private int getLastID() |  |
| 06 | public void addCouponDetail(int id, String[] productRequired, String[] productDiscounted) | Handle request add discounted product id or required id of a given coupon |
| 07 | private List<Product> getCouponDetailByID(int id, String productType) | Handle request get products related to a coupon |
| 08 | public List searchCoupon(String startDate, String endDate, String content, int s) |  |
| 09 | public void addPromotion(int customerID, String id, int adminId) | Handle request add promotion detail |
| 10 | public void promoteCouponUsed(String cusId, int couponid) | Change usage state of a promotion coupon for a customer |
| 11 | public List<Coupon> getDataPage(int page, List<Coupon> p, int npp) | For pagination of display list of coupon |

#### Product.java

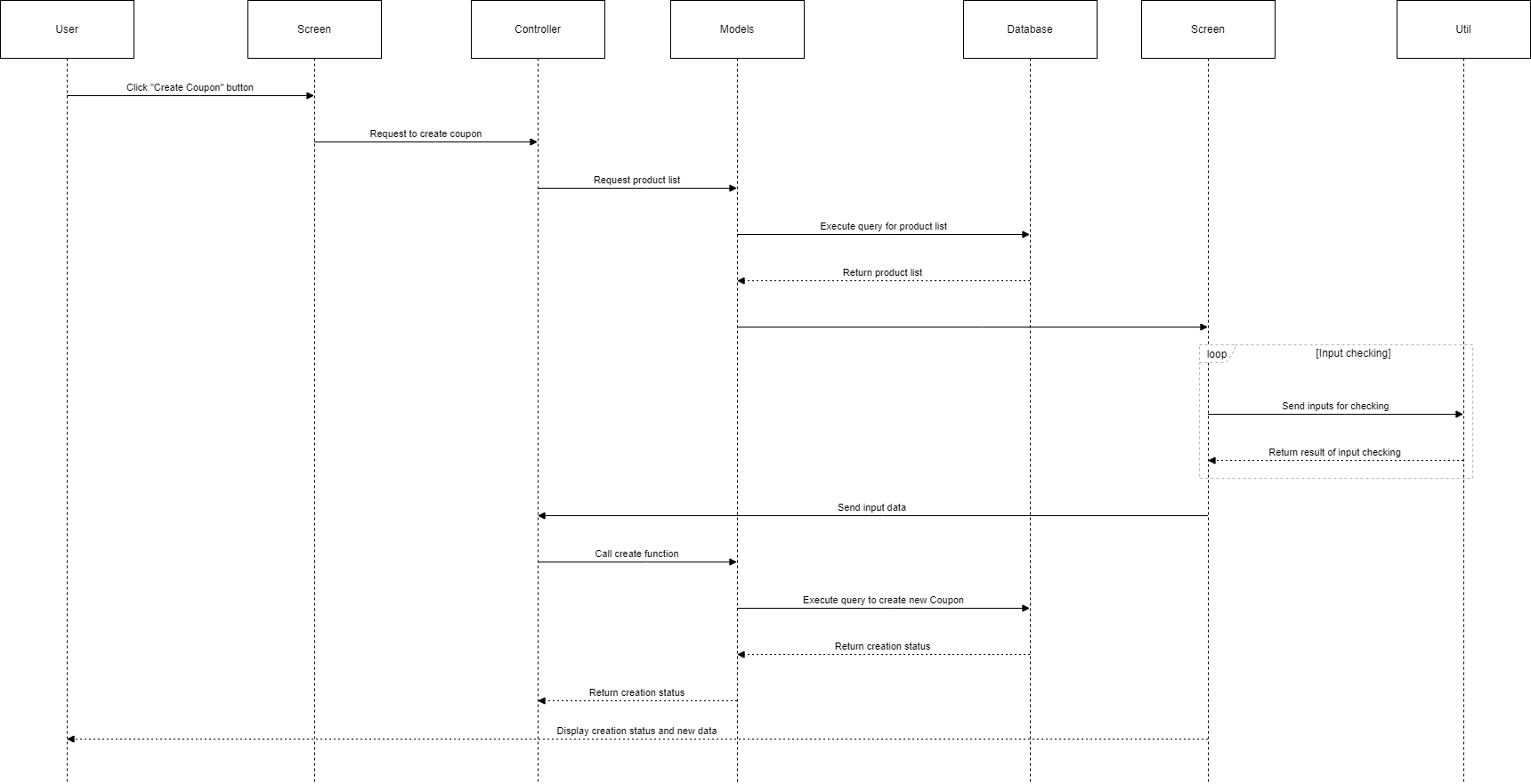
|  |  |  |
| --- | --- | --- |
| **No** | **Method** | **Description** |
| 01 | public List<Product> getListProductAtStart() | Handle request get list of all product |

#### Customer.java

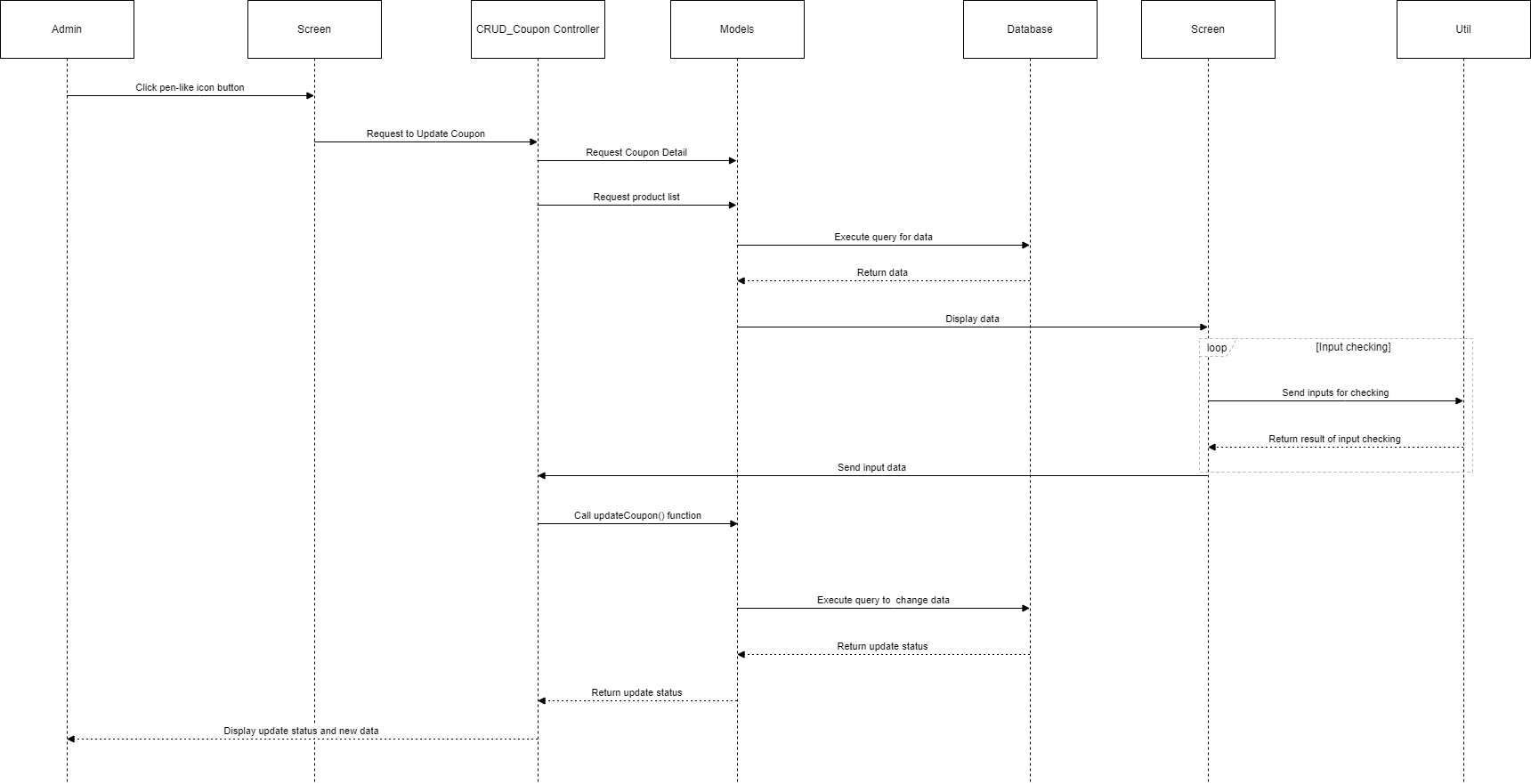
|  |  |  |
| --- | --- | --- |
| **No** | **Method** | **Description** |
| 01 | public List<Customer> getListCustomer() | Handle request get list of all customer |

### c. Sequence Diagram(s)

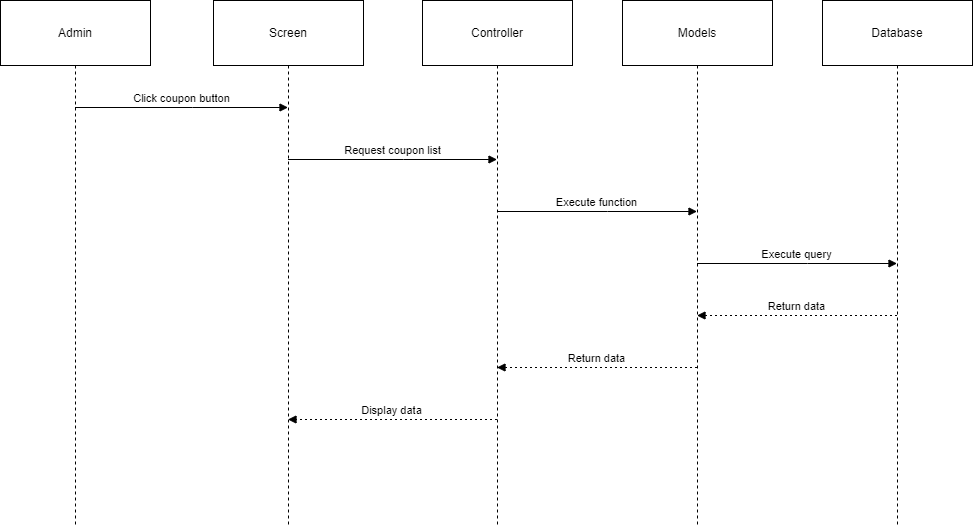
#### Create Coupon



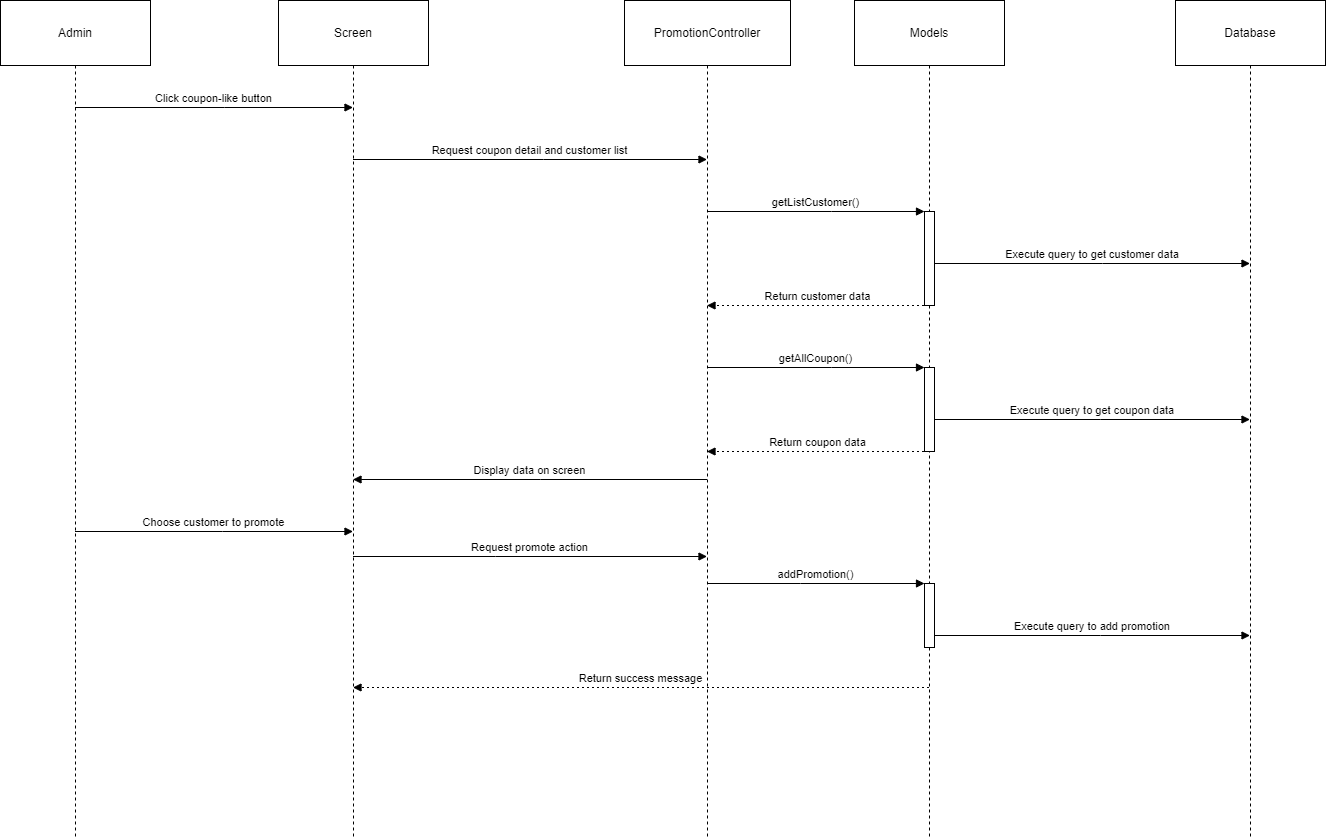
#### Update coupon



#### View Coupon

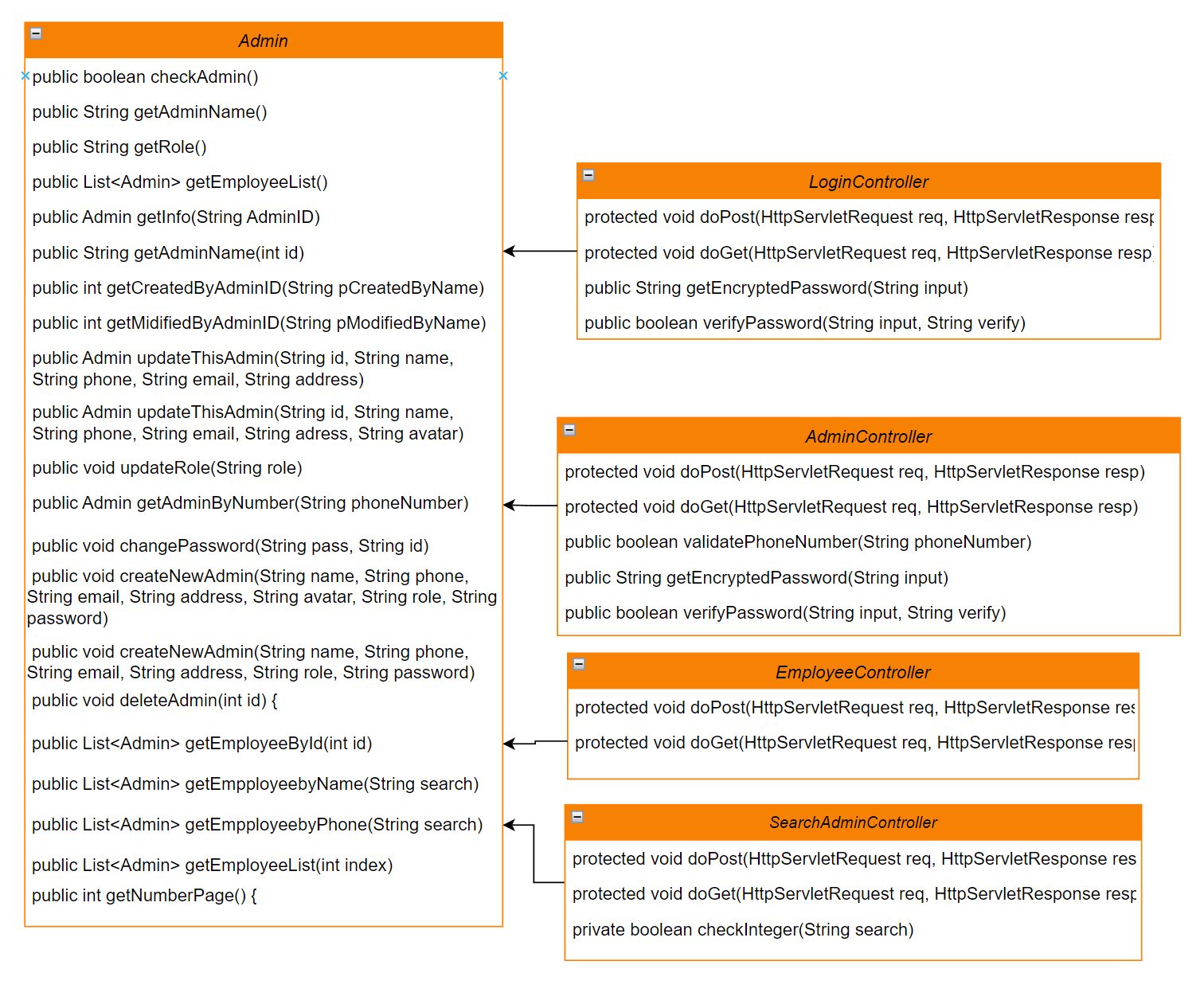


#### Promote Coupon for Customer



## 6. Admin

### a. Class Diagram



### b.Class Specifictions

#### LoginController

|  |  |  |
| --- | --- | --- |
| No | Method | Description |
| 01 | protected void doPost(HttpServletRequest req, HttpServletResponse resp) throws ServletException, IOException | Handle login request |
| 02 | protected void doGet(HttpServletRequest req, HttpServletResponse resp) throws ServletException, IOException | Handle logout request |
| 03 | public String getEncryptedPassword(String input) | Handle the encryption of password |
| 04 | public boolean verifyPassword(String input, String verify | Handle request to verify password |

#### AdminController

|  |  |  |
| --- | --- | --- |
| No | Method | Description |
| 01 | protected void doPost(HttpServletRequest req, HttpServletResponse resp) throws ServletException, IOException | Handle request for updating employees info |
| 02 | protected void doGet(HttpServletRequest req, HttpServletResponse resp) throws ServletException, IOException | Handle request to redirect to AdminInfo.jsp |
| 03 | public boolean validatePhoneNumber(String phoneNumber) | Handle request to validate the input phone number |
| 04 | public String getEncryptedPassword(String input) | Handle the encryption of password |
| 05 | public boolean verifyPassword(String input, String verify | Handle request to verify password |

#### EmployeeController

|  |  |  |
| --- | --- | --- |
| No | Method | Description |
| 01 | protected void doPost(HttpServletRequest req, HttpServletResponse resp) throws ServletException, IOException | Handle request to manage employee permission |
| 02 | protected void doGet(HttpServletRequest req, HttpServletResponse resp) throws ServletException, IOException | Handle request for the list of employees |

#### SearchAdminController

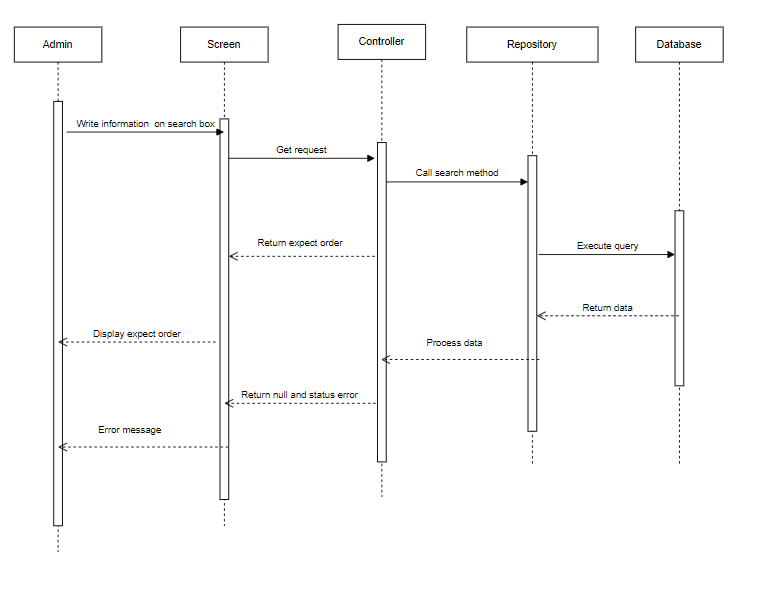
|  |  |  |
| --- | --- | --- |
| No | Method | Description |
| 01 | protected void doPost(HttpServletRequest req, HttpServletResponse resp) throws ServletException, IOException | Handle search function for employees |
| 02 | protected void doGet(HttpServletRequest req, HttpServletResponse resp) throws ServletException, IOException | Handle paging for the employee page |
| 03 | private boolean checkInteger(String search) | Validate the input string on whether it’s integer |

#### Admin

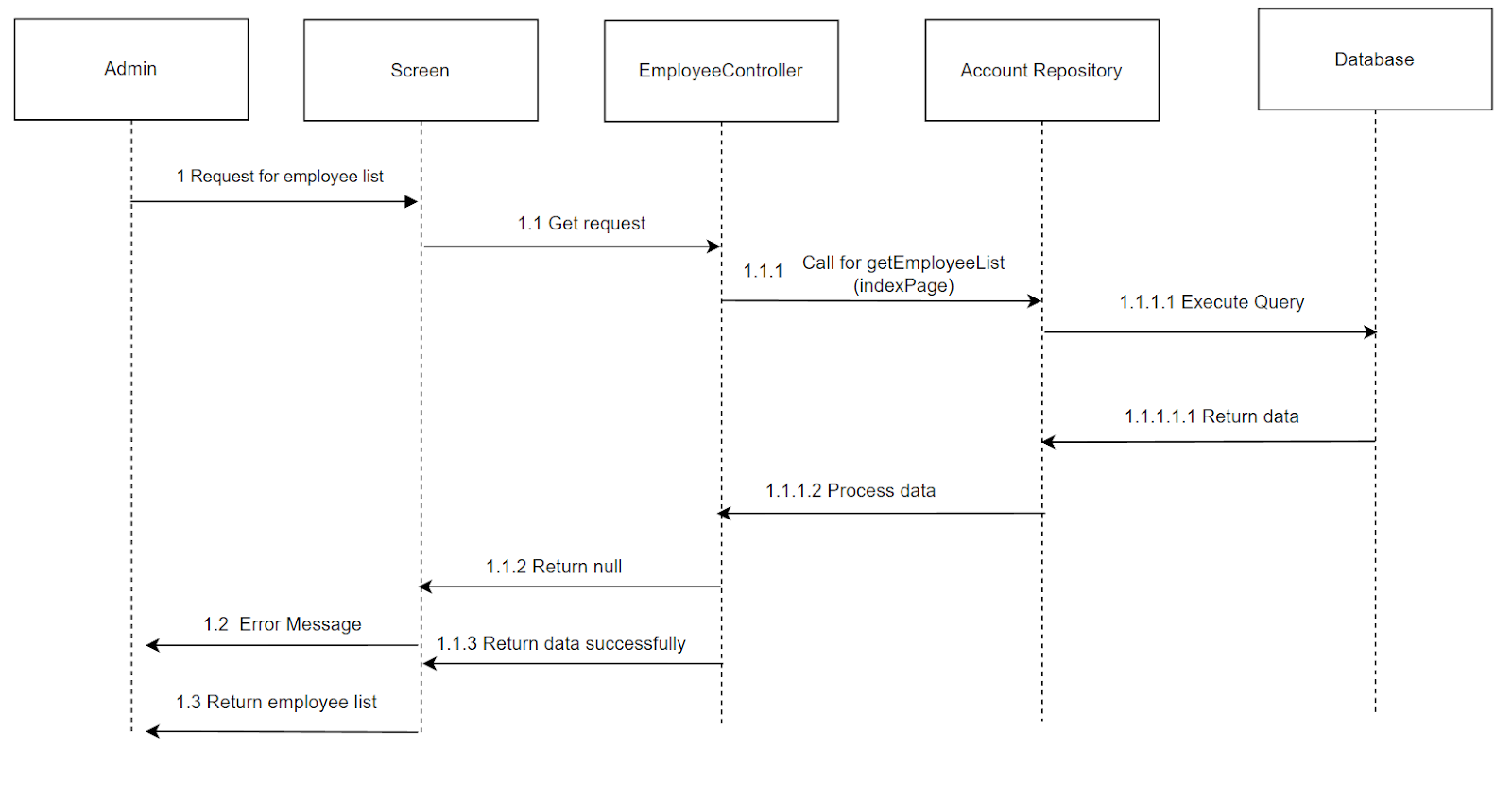
|  |  |  |
| --- | --- | --- |
| No | Method | Description |
| 01 | public boolean checkAdmin() | Handle request to validate login |
| 02 | public String getAdminName() | Handle request for employee name by their ID |
| 03 | public String getRole() | Handle request for employee’s role by their ID |
| 04 | public List<Admin> getEmployeeList() | Handle request for list of employees |
| 05 | public Admin getInfo(String AdminID) | Handle request for the employee’s info by their ID |
| 06 | public String getAdminName(int id) | Handle request for employee name by their ID |
| 07 | public int getCreatedByAdminID(String pCreatedByName) | Handle request for the employee ID from their name |
| 08 | public int getMidifiedByAdminID(String pModifiedByName) | Handle request for the employee ID from their name |
| 09 | public Admin updateThisAdmin(String id, String name, String phone, String email, String adress) | Handle request for updating the employee’s info |
| 10 | public Admin updateThisAdmin(String id, String name, String phone, String email, String adress, String avatar) | Handle request for updating the employee’s info |
| 11 | public void updateRole(String role) | Handle request for updating employee’s role |
| 12 | public Admin getAdminByNumber(String phoneNumber) | Handle request for employee’s info by their phone number |
| 13 | public void changePassword(String pass, String id) | Handle request for changing password |
| 14 | public void createNewAdmin(String name, String phone, String email, String address, String avatar, String role, String password) { | Handle request for adding new employee to the database |
| 15 | public void createNewAdmin(String name, String phone, String email, String address, String role, String password) | Handle request for adding new employee to the database |
| 16 | public void deleteAdmin(int id) | Handle request for removing an employee |
| 17 | public List<Admin> getEmployeeById(int id) | Handle request for list of employee by their ID |
| 18 | public List<Admin> getEmpployeebyName(String search) | Handle request for list of employee by their name |
| 19 | public List<Admin> getEmpployeebyPhone(String search) | Handle request for list of employee by their phone number |
| 20 | public List<Admin> getEmployeeList(int index) | Handle request for list of employee incorporating with paging |
| 21 | public int getNumberPage() | Handle request for the current page number |
| 22 | public List<Admin> getEmpployeebyEmail(String search) | Handle request for list of employee by their email |

### c. Sequence Diagram(s)

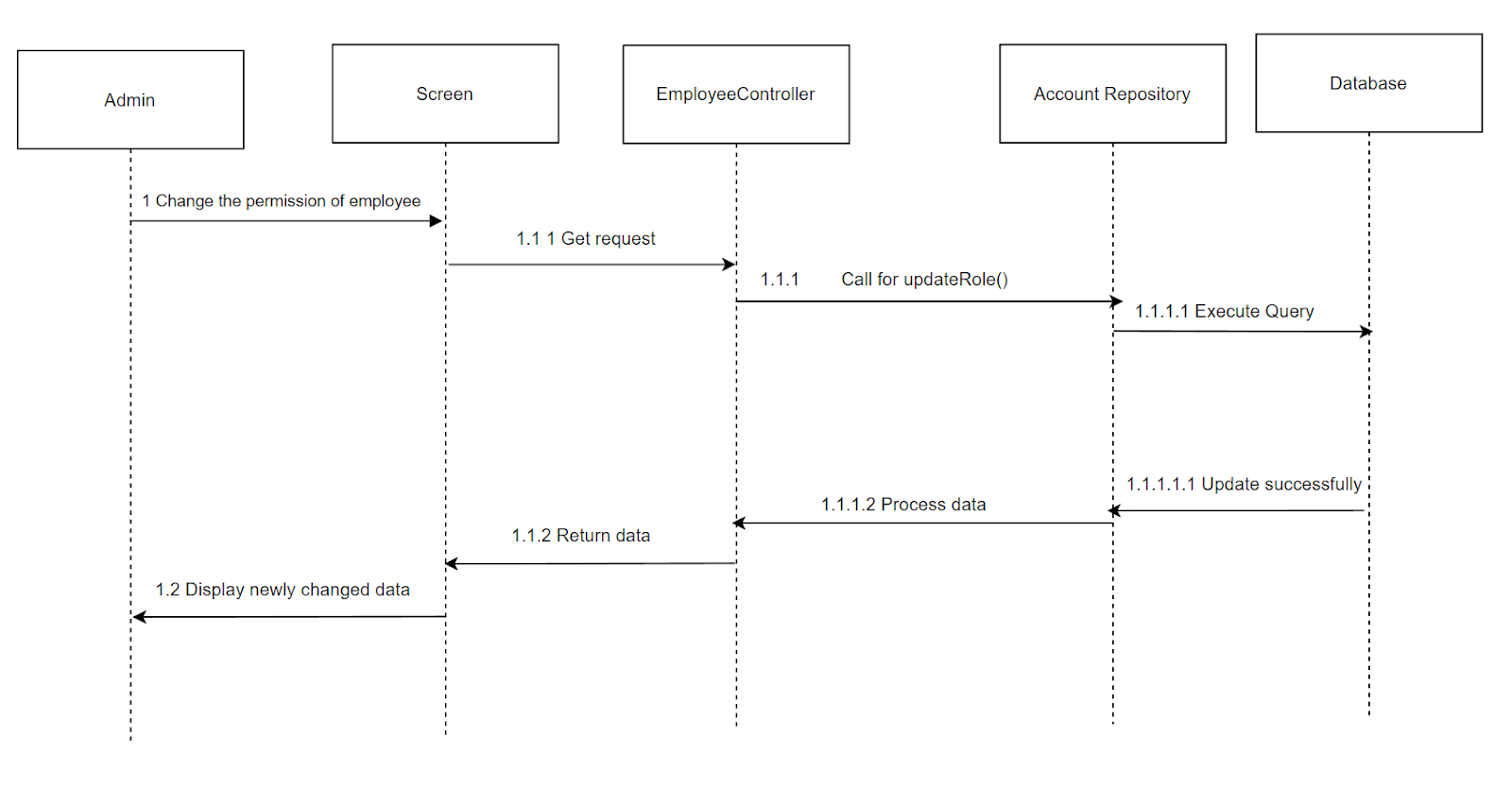
#### Search Employees



#### View employee list



#### Update Admin role



#### Create Employee

