



# **Requirement & Design Specification**

**Global Assess Management System (GAMS)**

## **Food Review Website**

**Group 3\_SE1825-NET**

– Hanoi, June 2024 –

**Version: 1.0**

– Hanoi, August 2022 –

## Record of Changes

| Version | Date | A*<br>M, D | In charge          | Change Description  |
|---------|------|------------|--------------------|---|
|         | 12/6 | A          | Phong, Quan        | Add user requirements (actors, use cases)                                     |
|         | 14/6 | M          | Phong , Quan       | Modify user requirements (actors)   |
|         | 15/6 | M          | Phong ,<br>HoangNM | Modify user requirements (use cases)  |
|         | 17/6 | A          | Hoang              | Modify user requirement (descriptions)  |
|         | 17/6 | A          | Phong              | Add Database (diagram , table description)                                    |
|         | 20/6 | M          | TungPT             | Requirement Specification (Feature : Login , Sign Up , Email Authenticate)    |
|         | 20/6 | M          | HoangTV            | Requirement Specification (Feature :User Profile , Edit User Profile)         |
|         | 20/6 | A          | Phong, Tung        | Requirement Specification (Feature :Create Post , Edit User Profile)          |
|         | 20/6 | A          | Phong              | Requirement Specification(Feature: Comment)                                   |
|         | 30/6 | A          | Tung               | Requirement Specification(Feature: like)                                      |
|         | 3/7  | A          | Tung               | Requirement Specification ( Feature : Follower-Following)                     |
|         | 3/7  | A          | Quan               | Add screenflow and screen descriptions  |
|         | 7/7  | A          | HoangNM            | Requirement Specification(Feature: Post report and admin report post control) |
|         | 18/7 | A          | Tung               | Requirement Specification ( Feature : save post)                              |
|         | 18/7 | M          | Phong              | Update UI for interacting functions in post detail                            |
|         | 20/7 | A          | HoangNM            | Requirement Specification ( Feature : User report)                            |
|         | 21/7 | A          | Phong, Tung        | Detailed Designs  |
|         | 22/7 | M          | Phong              | Complete screenflow   |
|         | 23/7 | M          | HoangNM            | Update Detailed Designs   |
|         | 24/7 | M          | HoangNM            | Update Business Rules   |

\*A - Added M - Modified D - Deleted

## Contents

|  |    |
|--|----|
| Record of Changes.....                 | 2  |
| I. Overview.....                       | 6  |
| 1. User Requirements.....              | 6  |
| 1.1 Actors.....                        | 6  |
| 1.2 Use Cases.....                     | 6  |
| 2. System Functionalities.....         | 8  |
| 2.1 Screens Flow.....                  | 8  |
| 2.2 Screen Authorization.....          | 8  |
| 2.3 Non-UI Functions.....              | 9  |
| 3. High Level Design.....              | 9  |
| 3.1 Database Design.....               | 9  |
| 3.1.2 Table Descriptions .....         | 10 |
| 3.2 Code Packages.....                 | 12 |
| 3.2.1 Package Diagram.....             | 12 |
| 3.2.2 Package Descriptions .....       | 12 |
| 4. Business Rules .....                | 13 |
| II. Functional Requirements.....       | 14 |
| 1. Authentication.....                 | 15 |
| 1.1 Login.....                         | 15 |
| 1.2 Sign Up.....                       | 18 |
| 1.3 Forgot Password .....              | 21 |
| 1.4 Remember account.....              | 23 |
| 1.5 Change Password .....              | 25 |
| 2. Patron FeaturesCommon Features..... | 27 |
| 2.1 View Explore.....                  | 27 |
| 2.2 View Feed .....                    | 29 |
| 2.3 View Posts.....                    | 31 |
| 2.4 View Restaurant.....               | 33 |
| 2.5 View Profile.....                  | 34 |
| 3. Patron Features .....               | 36 |
| 3.1 Create Posts .....                 | 36 |
| 3.2 Update Profile .....               | 39 |
| 3.3 Update Avatar.....                 | 40 |
| 3.4 Make Comment.....                  | 42 |

|   |    |
|---|----|
| 3.5 Report Post .....                     | 43 |
| 3.6 Edit Post.....                        | 45 |
| 3.7 Like Post .....                       | 46 |
| 3.8 Save Post.....                        | 48 |
| 3.9 Follow User.....                      | 48 |
| 3.10 Add Restaurant.....                  | 50 |
| 3.11 Search Restaurant .....              | 51 |
| 3.12 Search Post.....                     | 53 |
| 3.13 Edit Restaurant.....                 | 54 |
| 3.14 Report Restaurant .....              | 55 |
| 3.15 View Save Post.....                  | 57 |
| 3.16 Like Comment.....                    | 58 |
| 4. Admin Features .....                   | 60 |
| 4.1 Dashboard.....                        | 60 |
| 4.2 Manager User.....                     | 61 |
| 4.3 Manager Restaurant.....               | 63 |
| 4.4 Process Post Request.....             | 65 |
| 4.5 Processes Restaurant Request.....     | 66 |
| 4.6 Process Request Edit Post.....        | 68 |
| 4.7 Process Request Edit Restaurant ..... | 70 |
| 4.8 Process Post Report.....              | 71 |
| 4.9 Processes Restaurant Report.....      | 73 |
| III. Detailed Designs.....                | 76 |
| 1.Authentication.....                     | 76 |
| 1.1 Login .....                           | 76 |
| 1.2 Register.....                         | 77 |
| 1.3 Forgot Password.....                  | 78 |
| 1.4 Change password.....                  | 79 |
| 2.Common Page.....                        | 80 |
| 2.1 Feed Page .....                       | 80 |
| 2.2 Explore Page.....                     | 81 |
| 2.3 Restaurant List Page .....            | 82 |
| 2.4 Post Page.....                        | 83 |
| 2.5 Restaurant Page .....                 | 84 |
| 3.Function .....                          | 85 |
| 3.1 Update Profile Information.....       | 85 |
| 3.2 Update Avatar .....                   | 86 |
| 3.3 Follow/ UnFollow.....                 | 87 |

|   |     |
|---|-----|
| 3.4 Add Restaurant.....                   | 88  |
| 3.5 Edit Restaurant .....                 | 89  |
| 3.6 Delete Restaurant .....               | 90  |
| 3.7 Report Restaurant .....               | 90  |
| 3.8 Create Post .....                     | 92  |
| 3.9 Edit Post .....                       | 94  |
| 3.10 Delete Post .....                    | 95  |
| 3.11 Report Post .....                    | 96  |
| 3.12 Like/ UnLike Post .....              | 97  |
| 3.13 Save/UnSave Post .....               | 98  |
| 3.14 Comment .....                        | 99  |
| 3.15 Like/ UnLike Comment .....           | 100 |
| 3.16 Delete Comment .....                 | 100 |
| 4.Admin Function.....                     | 101 |
| 4.1 Dashboard Page .....                  | 101 |
| 4.2 Manager User.....                     | 102 |
| 4.3 Manager Restaurant .....              | 103 |
| 4.4 Process Post Request.....             | 104 |
| 4.5 Process Restaurant Request.....       | 105 |
| 4.6 Process Request Edit Post .....       | 106 |
| 4.7 Process Edit Restaurant Request ..... | 107 |
| 4.8 Process Post Report.....              | 108 |
| 4.9 Process Restaurant Report .....       | 109 |

# I. Overview

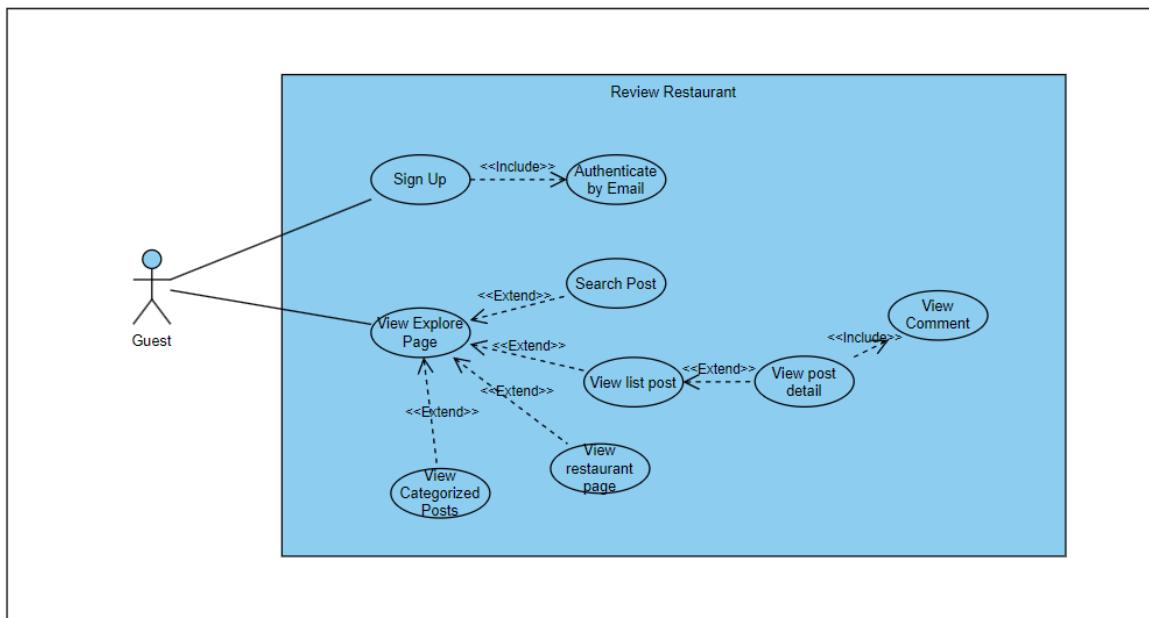
## 1. User Requirements

### 1.1 Actors

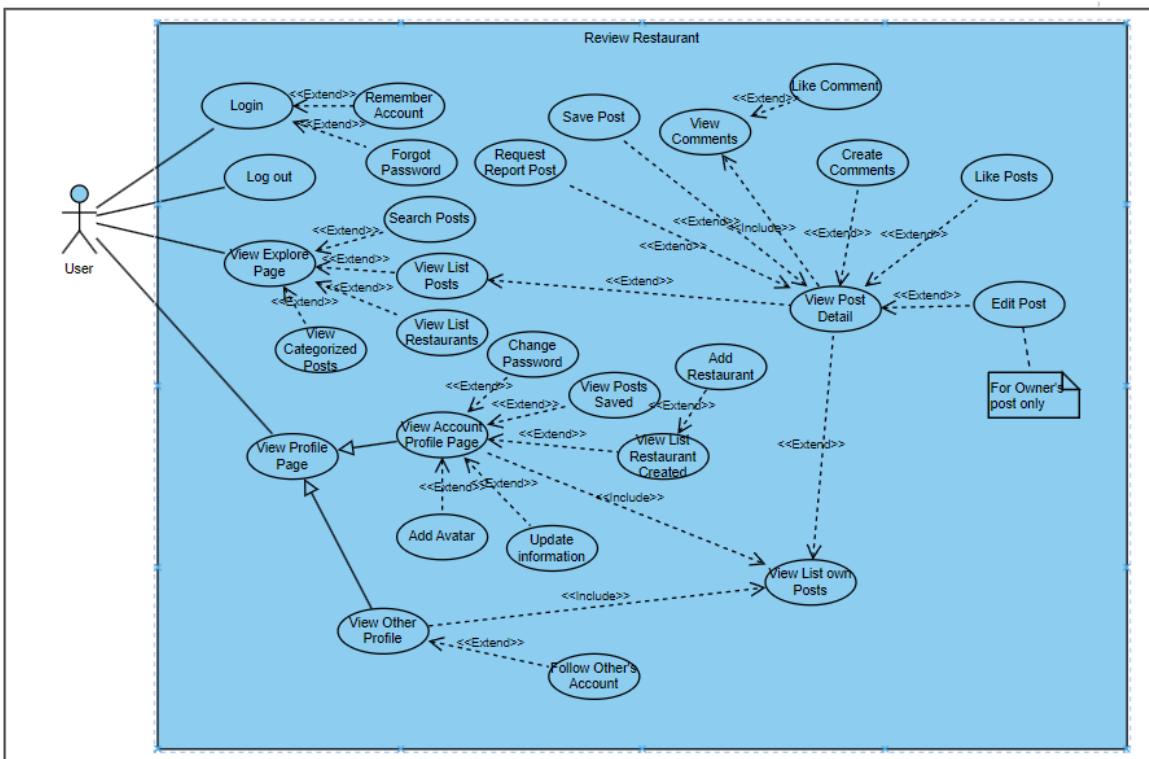
| # | Actor         | Description  |
|---|---------------|--|
| 1 | Administrator | The admin is the person with the highest management authority, general management of the entire system, and has full responsibility for personnel, techniques and some member support issues                                 |
| 2 | User          | Users within the restaurant review system, underscoring their proactive engagement in both contributing and accessing information, interacting with the platform, and shaping its capabilities through reviews and feedback. |
| 3 | Guest         | Guests can enter the system and view basic information such as posts or the user's personal page. However, they cannot interact with posts or follow users   |

### 1.2 Use Cases

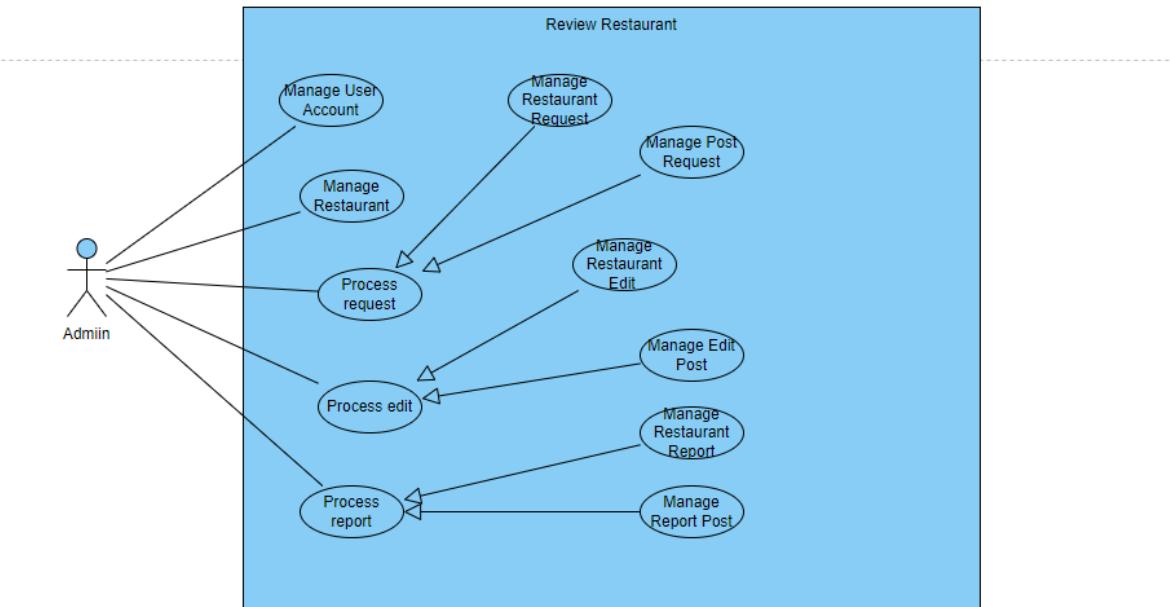
#### 1.2.1 UCs for Guest



### 1.2.2 UCs for User

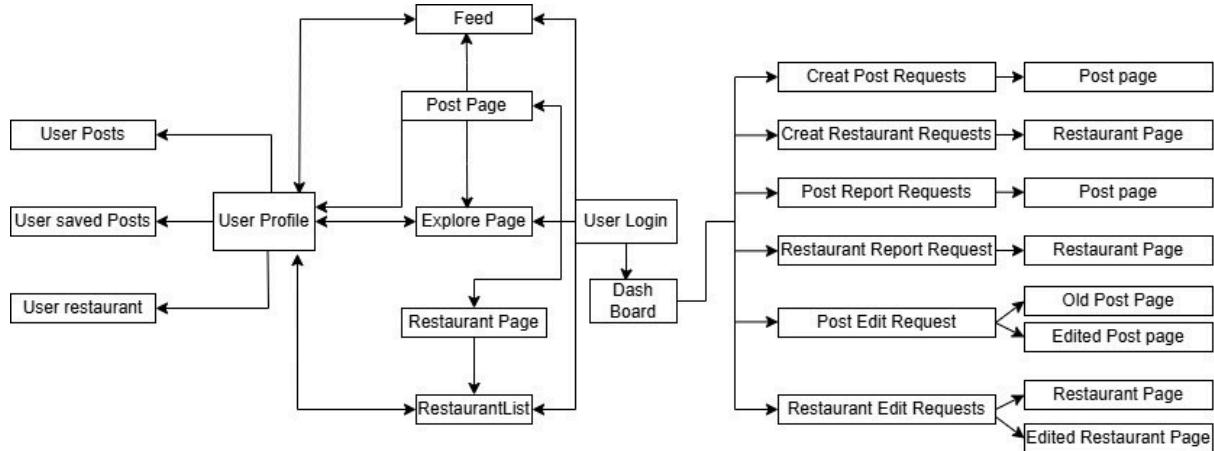


### 1.2.3 UCs for Admin



## 2. System Functionalities

### 2.1 Screens Flow



### 2.2 Screen Authorization

| Screen                            | User | Admin | Guest |
|-----------------------------------|------|-------|-------|
| <b>Login</b>                      | X    | X     | X     |
| Register                          | X    |       | X     |
| Forgot Password                   | X    | X     |       |
| Login                             | X    | X     |       |
| <b>Feed</b>                       | X    |       |       |
| Create Post                       | X    |       |       |
| <b>Explore</b>                    | X    | X     | X     |
| Create Post                       | X    |       |       |
| <b>Restaurants List</b>           | X    | X     | X     |
| <b>User Profile</b>               | X    | X     | X     |
| Update Information                | X    |       |       |
| Change Password                   | X    |       |       |
| Add New Restaurant                | X    |       |       |
| Follow User                       | X    | X     |       |
| <b>Post Page</b>                  | X    | X     | X     |
| Comment                           | X    | X     |       |
| Edit Post                         | X    |       |       |
| Delete Post                       | X    |       |       |
| <b>Restaurant Page</b>            | X    | X     | X     |
| Create Post                       | X    |       |       |
| Edit Restaurant                   | X    |       |       |
| Delete Restaurant                 | X    |       |       |
| <b>Dashboard</b>                  |      | X     |       |
| <b>Create Post Requests</b>       |      | X     |       |
| Accept                            |      | X     |       |
| Deny                              |      | X     |       |
| View Post                         |      | X     |       |
| <b>Create Restaurant Requests</b> |      | X     |       |
| Accept                            |      | X     |       |

|                                   |  |   |  |
|-----------------------------------|--|---|--|
| Deny                              |  | X |  |
| View Restaurant                   |  | X |  |
| <b>Post Report Requests</b>       |  | X |  |
| Accept                            |  | X |  |
| Deny                              |  | X |  |
| View Post                         |  | X |  |
| <b>Restaurant Report Requests</b> |  | X |  |
| Accept                            |  | X |  |
| Deny                              |  | X |  |
| View Restaurant                   |  | X |  |
| <b>Post Edit Requests</b>         |  | X |  |
| Accept                            |  | X |  |
| Deny                              |  | X |  |
| View Old Post                     |  | X |  |
| View Edited Post                  |  | X |  |
| <b>Restaurant Edit Requests</b>   |  | X |  |
| Accept                            |  | X |  |
| Deny                              |  | X |  |
| View Old Restaurant               |  | X |  |
| View Edited Restaurant            |  | X |  |

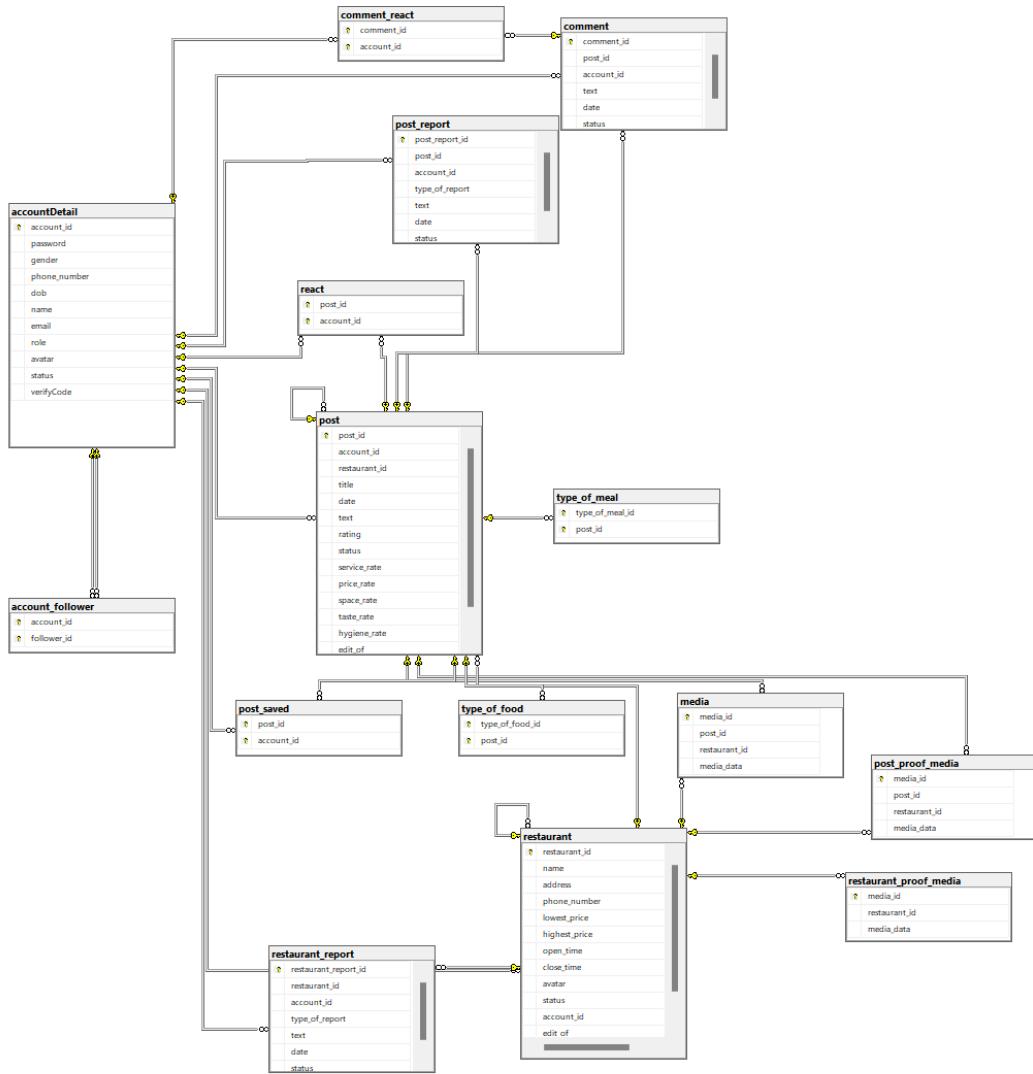
### 2.3 Non-UI Functions

| # | Feature                 | System Function                                | Description   |
|---|-------------------------|--|---|
| 1 | Register By Email       | Register By Email                              | Use Email To Register   |
| 2 | Password Encryption     | Password Encryption                            | Encrypt Password And Save To Database   |
| 3 | Save Images To DataBase | Add images when Create Post, Update Avatar,... | Use @MultipartConfig in Servlet to get images, then save to database by Binary Char data type |
| 4 | Google Map              | Insert Google Map to web pages                 | Use Google Map API  |
| 5 | Vietnam Address         | Insert selects of Vietnam addresses            | Use VN public APIs  |

## 3. High Level Design

### 3.1 Database Design

### 3.1.1 Database Schema



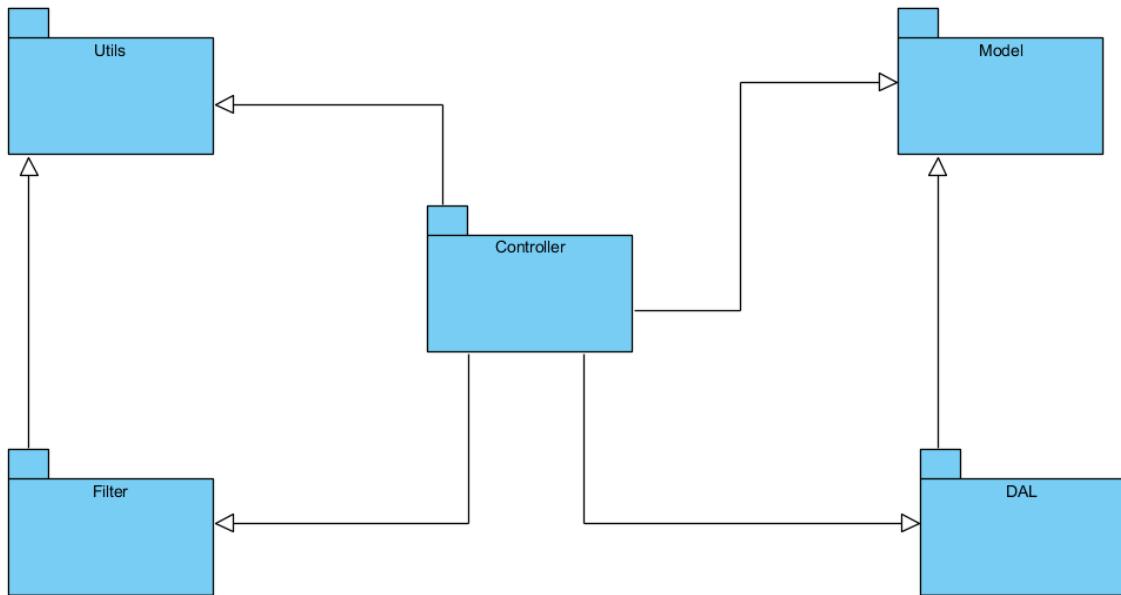
### 3.1.2 Table Descriptions

| No | Table                  | Description  |
|----|------------------------|--|
| 01 | accountDetail          | <ul style="list-style-type: none"> <li>- Primary keys: account_id</li> <li>- Foreign keys: &lt;&lt;list of foreign key fields&gt;&gt;</li> <li>- Unique keys:&lt;&lt;list of unique field or unique field set&gt;&gt;</li> </ul>   |
| 02 | account_follower       | <ul style="list-style-type: none"> <li>- Primary keys: (account_id, follower_id)</li> <li>- Foreign keys: <ul style="list-style-type: none"> <li>(account_id) REFERENCES accountDetail(account_id),</li> <li>(follower_id) REFERENCES accountDetail(account_id)</li> </ul> </li> </ul> |
| 03 | restaurant             | <ul style="list-style-type: none"> <li>- Primary keys: restaurant_id</li> <li>- Foreign keys: <ul style="list-style-type: none"> <li>(edit_of) REFERENCES restaurant(restaurant_id),</li> <li>(account_id) REFERENCES accountDetail(account_id)</li> </ul> </li> </ul>                 |
| 04 | restaurant_proof_media | <ul style="list-style-type: none"> <li>- Primary keys: media_id</li> <li>- Foreign keys: <ul style="list-style-type: none"> <li>(restaurant_id) REFERENCES restaurant(restaurant_id)</li> </ul> </li> </ul>  |
| 05 | post                   | <ul style="list-style-type: none"> <li>- Primary keys: post_id</li> <li>- Foreign keys:</li> </ul>   |

|    |                   |   |
|----|-------------------|---|
|    |                   | (restaurant_id) REFERENCES restaurant(restaurant_id),<br>(account_id) REFERENCES accountDetail(account_id)  |
| 06 | comment           | - Primary keys: comment_id<br>- Foreign keys:<br>(post_id) REFERENCES post(post_id),<br>account_id REFERENCES accountDetail(account_id)                               |
| 07 | comment_react     | - Primary keys: (comment_id , account_id)<br>- Foreign keys:<br>(comment_id) REFERENCES comment(comment_id),<br>(account_id) REFERENCES accountDetail(account_id)     |
| 08 | post_report       | - Primary keys: (comment_id , account_id)<br>- Foreign keys:<br>(post_id) REFERENCES post(post_id),<br>(account_id) REFERENCES accountDetail(account_id)              |
| 09 | restaurant_report | - Primary keys: restaurant_report_id<br>- Foreign keys:<br>(restaurant_id) REFERENCES restaurant(restaurant_id),<br>(account_id) REFERENCES accountDetail(account_id) |
| 10 | post_proof_media  | - Primary keys: media_id<br>- Foreign keys:<br>(post_id) REFERENCES post(post_id),<br>(restaurant_id) REFERENCES restaurant(restaurant_id)                            |
| 11 | media_id          | - Primary keys: media_id<br>- Foreignkeys:<br>(post_id) REFERENCES post(post_id)  |
| 12 | type_of_meal      | - Primary keys: (type_of_meal_id, post_id)<br>- Foreign keys:<br>(post_id) REFERENCES post(post_id)   |
| 13 | type_of_food      | - Primary keys: (type_of_food_id, post_id)<br>- Foreign keys:<br>(post_id) REFERENCES post(post_id)   |
| 14 | react             | - Primary keys: (post_id , account_id)<br>- Foreign keys:<br>(post_id) REFERENCES post(post_id),<br>account_id REFERENCES accountDetail(account_id)                   |
| 15 | post_saved        | - Primary keys: (post_id , account_id)<br>- Foreign keys:<br>(post_id) REFERENCES post(post_id),<br>account_id REFERENCES accountDetail(account_id)                   |

## 3.2 Code Packages

### 3.2.1 Package Diagram



### 3.2.2 Package Descriptions

| No | Package    | Description   |
|----|------------|---|
| 01 | Controller | This package contains servlets and controllers that handle user requests and determine the responses.   |
| 02 | DAL        | This package includes classes that interact with the database. These classes contain methods for CRUD (Create, Read, Update, Delete) operations.  |
| 03 | Model      | This package contains the data models or entities that represent the application's core data structures. These classes are typically mapped to database tables.   |
| 04 | Filter     | This package includes classes that act as filters to process requests and responses before they reach the controller or after they leave the controller. Filters are used for tasks: decentralization, logging. |
| 05 | Utils      | This package contains utility classes and methods that provide common functionalities used throughout the application. These are verifying email, encoding password, generating random string, validating.      |

## 4. Business Rules

| ID   | Business Rule      | Business Rule Description  |
|------|--------------------|--|
| BR-1 | Password Encoding  | User's password must be encoded with SHA-1 hashing   |
| BR-2 | Invalid Logging In | User can't be authenticated to login the system if below cases:<br>User logging-in details are incorrect |

|       |  |  |
|-------|--|--|
|       |  | User account has not been verified   |
| BR-3  | Remember Me  | Option for users to stay logged in for a longer period on trusted devices.   |
| BR-4  | Password Recovery  | Enable users to reset their passwords through a secure password recovery process with email verification.  |
| BR-5  | OTP Verification   | User must verify their email address with OTP  |
| BR-6  | Data Privacy and Security                                | The system must comply with relevant data privacy regulations and protect user data from unauthorized access or misuse   |
| BR-7  | Email must exist   | User's email must exist  |
| BR-8  | Invalid Current Password, New Password, Confirm Password | User's current password aren't string(white space or blank)  |
| BR-9  | Encoding password  | User's new password must encoding with SHA-1 algorithm   |
| BR-10 | Fact   | Search keywords cannot be empty or contain only spaces.  |
| BR-11 | Computations   | The system must automatically update posts after the user posts and the admin approves the post  |
| BR-12 | Accurate Display of Information                          | Contents(posts name , posts description, posts images ) in posts detail page must be accurate to the posts that user clicked on at the homepage                |
| BR-13 |  | If the article was to be deleted , it must not be shown to other users under no circumstances  |
| BR-14 | User Authentication Requirement                          | Users must be authenticated (logged in) to interact with the website   |
| BR-15 | Reporting and Admin                                      | Users have the ability to report inappropriate or offensive content on other users' profiles, and administrators have the authority to take appropriate action |
| BR-16 | No empty title   | The article title cannot be empty and must have at least one character.  |
| BR-17 | Redirect to Profile after successfully created           | After creating a post, the system must redirect the user to the list of posts screen in user profile or the details screen of the newly created post.          |
| BR-18 | Save Confirmation  | Changes made to the profile must be saved successfully for the update to be considered complete.   |
| BR-19 | Upload proof picture                                     | User must enter proof picture in order to the admin can review the content   |

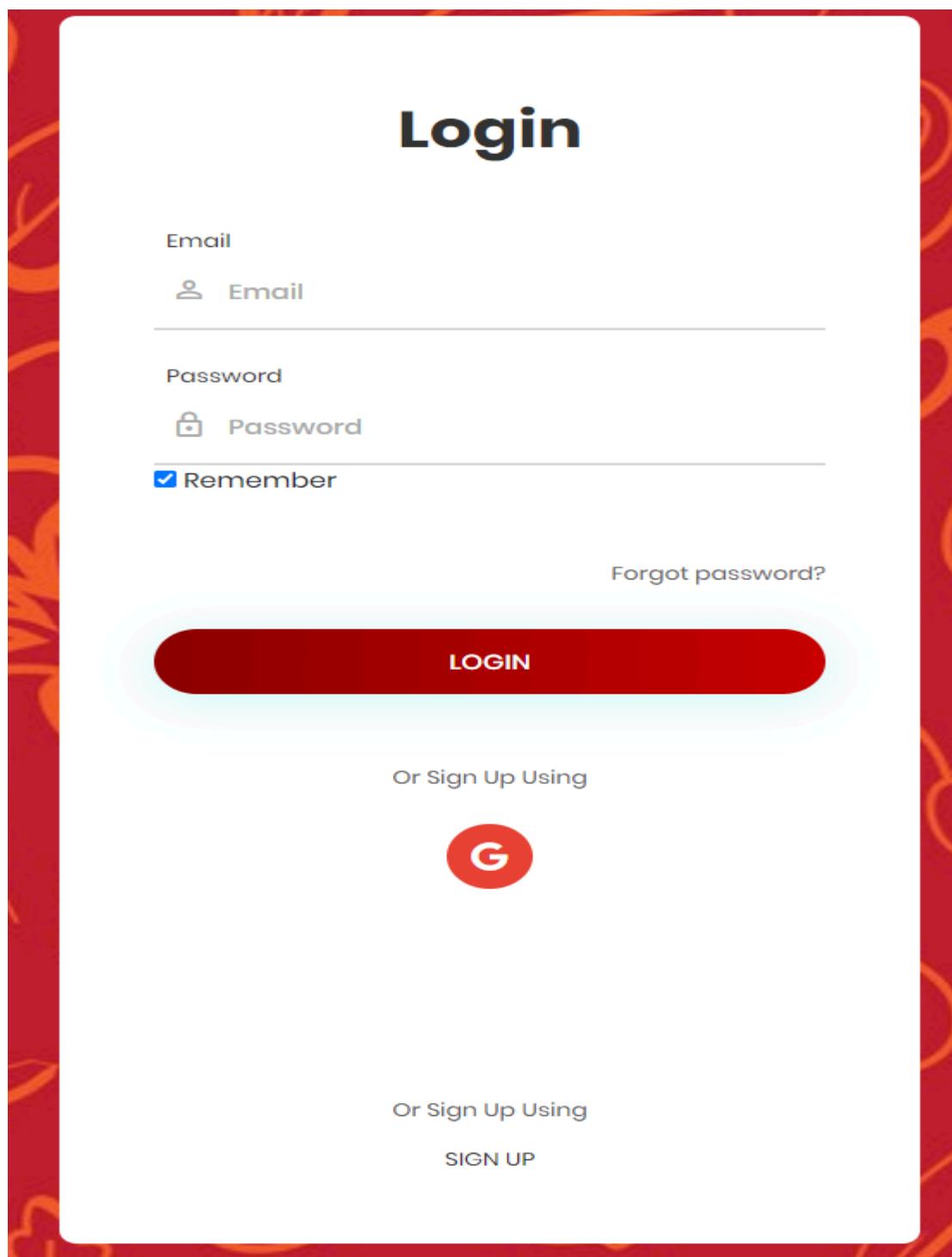
|       |   |   |
|-------|---|---|
| BR-20 | Respectful Communication                  | Users are expected to engage in respectful and constructive communication when commenting on posts, avoiding offensive language or behavior.          |
| BR-21 | Redirect to Profile after successful edit | After editing a post, the system must redirect the user to the post list screen in the user's profile or the detail screen of the newly created post. |
| BR-22 | Admin Authentication Requirement          | Admins must be authenticated (logged in) to access post review functionality.   |
| BR-23 | Admin Access Levels                       | Different access levels may exist for Admins, allowing varying degrees of control over post review and publication.                                   |
| BR-24 | Admin Page Access Confirmation            | After accessing the Admin page, Admins must confirm their identity if prompted by the system for security reasons.                                    |
| BR-25 | Post Review Confirmation                  | After reviewing a post, Admin must confirm their decision (approve or reject) for the action to be considered complete.                               |
| BR-26 | Confirm Restaurant Requests               | After reviewing a post, the Admin must confirm his decision (approve or deny) for the action to be considered complete.                               |
| BR-27 | Data Accuracy Confirmation                | The system must ensure the data displayed is accurate and up-to-date.   |
| BR-28 | Ban Request Review Confirmation           | After reviewing a ban request, Admin must confirm their decision (approve or deny) for the action to be considered complete.                          |
| BR-29 | Account Owner Authentication              | Only the owner can change them self profile, post, restaurant   |
| BR-30 | Account Owner Accessibility               | Only the owner can view their saved posts   |

## II. Functional Requirements

## 1. Authentication

### 1.1 Login

#### a. UI Design



*b. Functionalities*

|                    |   |                   |           |
|--------------------|---|-------------------|-----------|
| UC ID and Name:    | <b>LG-1_Login</b>   |                   |           |
| Created By:        | Trương Việt Hoàng   | Date Created:     | 20/5/2024 |
| Primary Actor:     | All   | Secondary Actors: | None      |
| Trigger:           | User clicks Login button from the page header   |                   |           |
| Description:       | Users are authenticated by verifying their credentials and granting access to the application or system.  |                   |           |
| Preconditions:     | User account has been registered and verified.  |                   |           |
| Postconditions:    | User logs in the system successfully<br><br>The system redirect to the Explore  |                   |           |
| Normal Flow:       | <p><b>1.0 Login</b></p> <ol style="list-style-type: none"> <li>1. User accesses the User Login screen.</li> <li>2. User types in the login details.</li> <li>3. User clicks the Login button.</li> <li>4. System validates the login details.</li> <li>5. System allows users to access.</li> <li>6. System accesses the Explore.</li> </ol>  |                   |           |
| Alternative Flows: | N/A   |                   |           |
| Exceptions:        | <p><b>1.0.1 The account doesn't exist</b></p> <ol style="list-style-type: none"> <li>1. The Error Message is shown to the user.</li> <li>2. Still retain the entered information.</li> <li>3. Users can start typing again.</li> </ol> <p><b>1.0.2 Username or password is not correct</b></p> <ol style="list-style-type: none"> <li>1. The Error Message is shown to the user.</li> <li>2. Still retain the entered information.</li> <li>3. Users can start typing again.</li> </ol> <p><b>1.0.3 The account is not verified</b></p> <ol style="list-style-type: none"> <li>1. User is redirected to the OTP confirmation page.</li> <li>2. User types the OTP code sent via email.</li> </ol> |                   |           |

|                    |   |
|--------------------|---|
|                    | <p>3. If the code is correct, redirect the user to the login page.</p> <p>4. Else show the error message and make the user type again.</p> <p><b>1.0.4 The account is banned</b></p> <p>1. User received an error message.</p> <p>2. Users must login with another account.</p> |
| Priority:          | High  |
| Frequency of Use:  | Any one who wants to comment, create and every modder, admin must login to use their corresponding functions.   |
| Business Rules:    | BR-1, BR-2, BR-3, BR-4  |
| Other Information: |   |
| Assumptions:       |   |

### c. Business Rules

Provide the business rules those are applied only to the use case

| ID   | Business Rule      | Business Rule Description  |
|------|--------------------|--|
| BR-1 | Password Encoding  | User's password must be encoded with SHA-1 hashing   |
| BR-2 | Invalid Logging In | User can't be authenticated to login the system if below cases:<br><br>User logging-in details are incorrect<br><br>User account has not been verified |
| BR-3 | Remember Me        | Option for users to stay logged in for a longer period on trusted devices.   |
| BR-4 | Password Recovery  | Enable users to reset their passwords through a secure password recovery process with email verification.  |

## 1.2 Sign Up

### a. UI Design

# Register

Nickname

 Nickname

---

Email

 Email

---

Password

 Password

---

Phone number

 Phone number

---

Gender

Male  Female

---

Date of birth

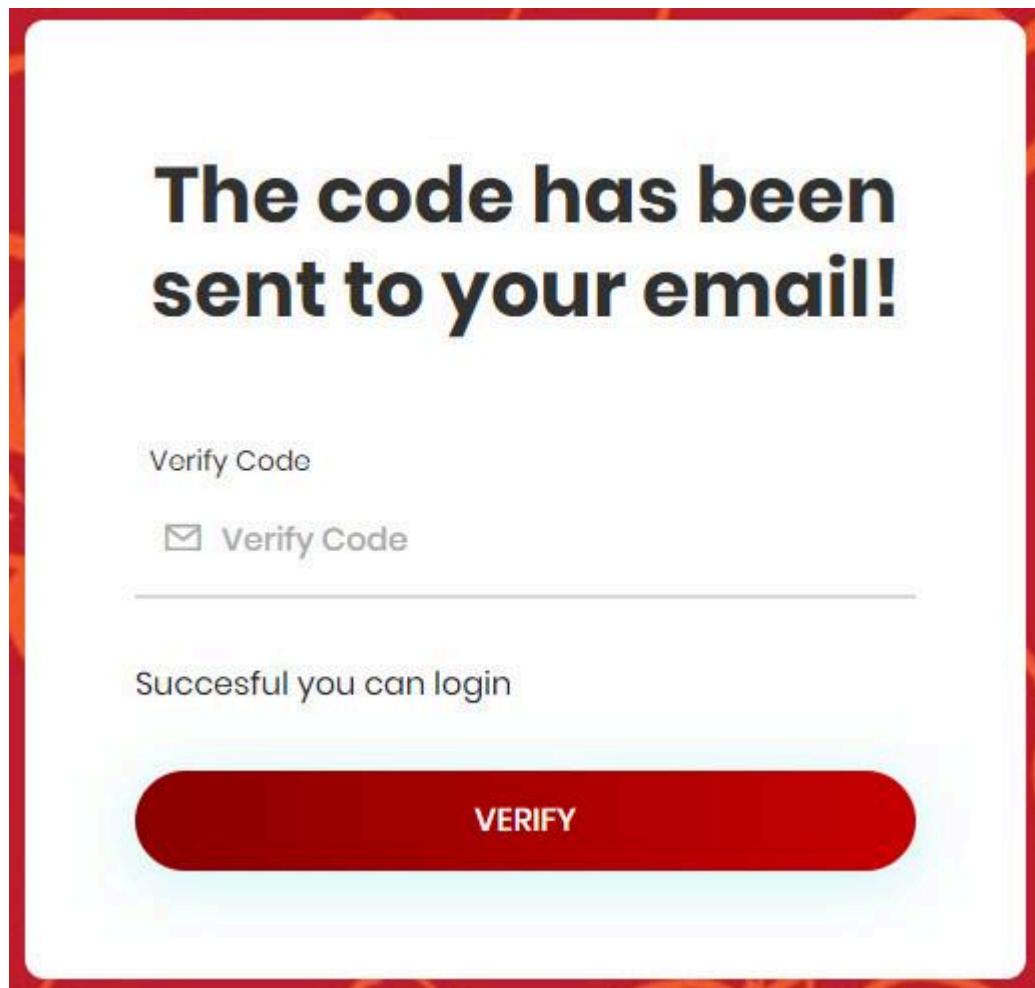
21/06/2024 

---

**REGISTER**

Or Sign Up Using





*b. Functionalities*

|                 |  |                   |           |
|-----------------|--|-------------------|-----------|
| UC ID and Name: | LG-2_Sign Up   |                   |           |
| Created By:     | Phạm Thanh Tùng  | Date Created:     | 20/5/2024 |
| Primary Actor:  | User   | Secondary Actors: | None      |
| Trigger:        | User clicks on Register button in Login pop-up screen  |                   |           |
| Description:    | User use the register function to sign up an account then they can have their own account and use other functions such as: view user profile, comment, create blog, .... |                   |           |
| Preconditions:  | Email  |                   |           |
| Postconditions: | System sends a OTP code to guest's email for verifying account   |                   |           |
| Normal Flow     | <b>1. Register</b><br>1. User accesses the User Login screen.  |                   |           |

|                    |   |
|--------------------|---|
|                    | <p>2. The user clicks the Register button to go to the Register page.</p> <p>3. User enters NickName, email, password, Phone number and select gender, Date of birth.</p> <p>4. User clicks the Sign up button .</p> <p>5. Users type the OTP code sent via email.</p> <p>6. System accesses the Login screen.</p>  |
| Alternative Flows: | <p><b>1.1 The account is registered but not verified</b></p> <p>1. User is redirected to the OTP confirmation page.</p> <p>2. User types the OTP code sent via email.</p> <p>3. If the code is correct, redirect the user to the login page.</p> <p>4. Else show the error message and make the user type again.</p>  |
| Exceptions:        | <p><b>1.1 Any field left blank</b></p> <p>1. User receives an error message.</p> <p><b>2. User gets redirected to registration page</b></p> <p><b>1.2 Password does not meet the requirements</b></p> <p>1. User receives an error message.</p> <p><b>2. User gets redirected to registration page</b></p> <p><b>1.3 Email already exists</b></p> <p>1. User receives an error message.</p> <p><b>2. User gets redirected to registration page</b></p> <p><b>1.4 Username already exists</b></p> <p>1. User receives an error message.</p> <p><b>2. User gets redirected to registration page</b></p> <p><b>1.6 Invalid email</b></p> <p>1. User receives an error message.</p> <p><b>2. User gets redirected to registration page</b></p> <p><b>1.7 Incorrect account</b></p> <p>1. User must enter correct gmail and password</p> |
| Priority:          | Must Have   |
| Frequency of Use:  | Guests that want to use the available function for user's only  |
| Business Rules:    | BR-1, BR-2, BR-5, BR-6  |
| Other Information: | N/A   |

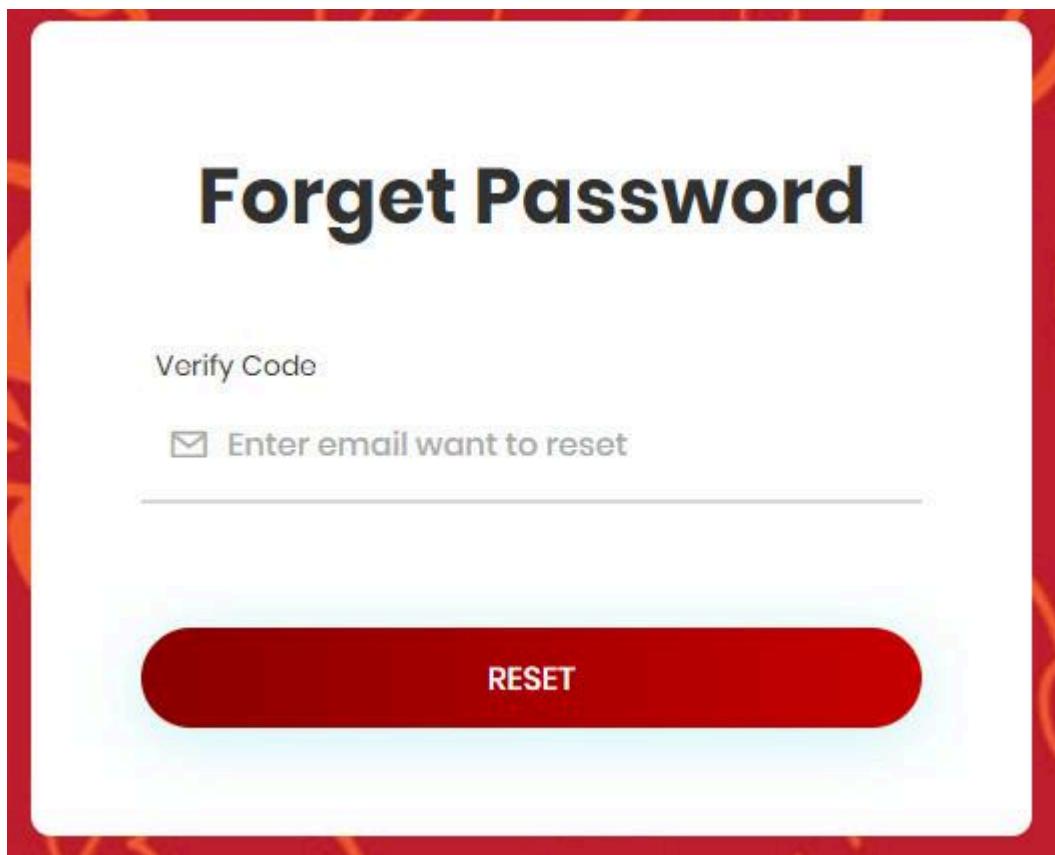
|              |     |
|--------------|-----|
| Assumptions: | N/A |
|--------------|-----|

*c. Business Rules*

| ID   | Business Rule             | Business Rule Description  |
|------|---------------------------|--|
| BR-1 | Password Encoding         | User's password must be encoded with SHA-1 hashing   |
| BR-2 | Invalid Logging In        | User can't register in below cases<br><br>User email are in wrong format<br><br>User email existed in system database  |
| BR-5 | OTP Verification          | User must verify their email address with OTP  |
| BR-6 | Data Privacy and Security | The system must comply with relevant data privacy regulations and protect user data from unauthorized access or misuse |

**1.3 Forgot Password**

*a. UI Design*



*b. Functionalities*

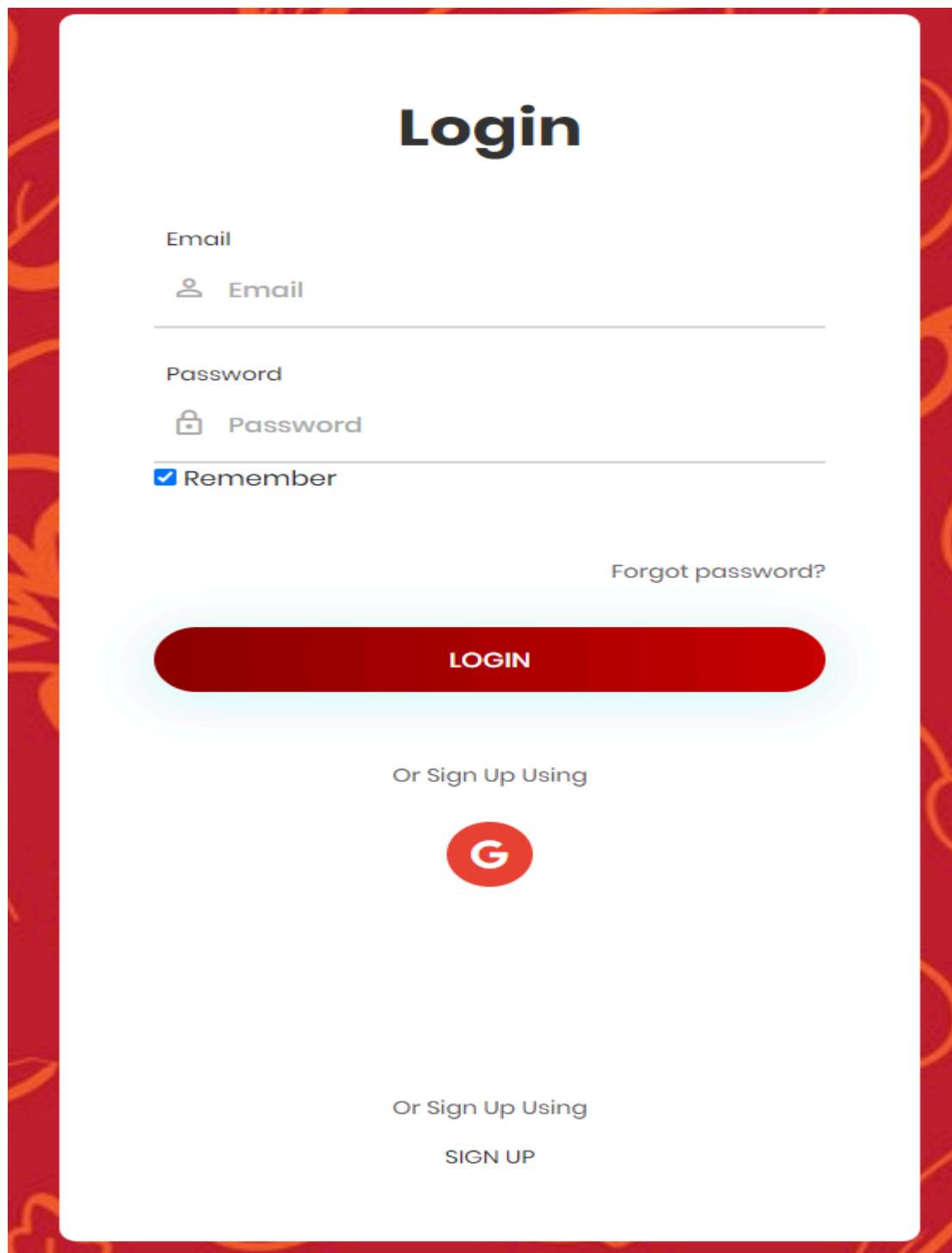
|                    |   |                   |           |
|--------------------|---|-------------------|-----------|
| UC ID and Name:    | LG-03 Forgot Password   |                   |           |
| Created By:        | Phạm Thanh Tùng   | Date Created:     | 20/5/2024 |
| Primary Actor:     | All user  | Secondary Actors: | None      |
| Trigger:           | Registered user clicks on "Forgot Password" link on the Sign in page  |                   |           |
| Description:       | The "Forgot Password" feature allows users who have forgotten their password to regain access to their accounts securely. When triggered, users are prompted to enter their email address associated with their account. An email containing an OTP code to reset their password is then sent to the provided email address.            |                   |           |
| Preconditions:     | User account has been registered and verified.  |                   |           |
| Postconditions:    | User successfully resets their password and gains access to their account.  |                   |           |
| Normal Flow:       | 1. User clicks on the "Forgot Password" link on the login screen.<br>2. User is prompted to enter their email address or username associated with their account.<br>3. User submits the form.<br>4. The system verifies the provided information and sends an email with new password<br>5. User can now log in using the new password. |                   |           |
| Alternative Flows: | N/A   |                   |           |
| Exceptions:        | Incorrect Email or Username: If the provided email or username does not match any account in the system, an error message is displayed prompting the user to try again.   |                   |           |
| Priority:          | High  |                   |           |
| Frequency of Use:  | Every user may use this feature.  |                   |           |
| Business Rules:    | BR-5, BR-7  |                   |           |
| Other Information: | N/A   |                   |           |
| Assumptions:       | N/A   |                   |           |

*c. Business Rules*

| ID   | Business Rule    | Business Rule Description         |
|------|------------------|-----------------------------------|
| BR-5 | OTP Verification | OTP must send to the User's email |
| BR-7 | Email must exist | User's email must exist           |

## 1.4 Remember account

### a. UI Design



### b. Functionalities

|                    |   |                   |           |
|--------------------|---|-------------------|-----------|
| UC ID and Name:    | LG-04 Remember account  |                   |           |
| Created By:        | Trương Việt Hoàng   | Date Created:     | 20/5/2024 |
| Primary Actor:     | All user  | Secondary Actors: | None      |
| Trigger:           | Registered user click the checkbox "Remember" on the Sign in page   |                   |           |
| Description:       | The "Remember" feature allows users to be automatically logged in the next time they come to the website.   |                   |           |
| Preconditions:     | User account has been registered and verified.  |                   |           |
| Postconditions:    | The next time they come to the website, they are automatically logged in.   |                   |           |
| Normal Flow:       | <p><b>1.0 Check the checkbox</b></p> <ol style="list-style-type: none"> <li>1. User clicks and puts a check on the check box on the login screen.</li> <li>2. User logs in like usual.</li> </ol> |                   |           |
| Alternative Flows: | <p><b>1.1 Uncheck the checkbox</b></p> <ol style="list-style-type: none"> <li>1. User clicks and unchecks the checkbox on the login screen.</li> <li>2. User logs in like usual.</li> </ol>       |                   |           |
| Exceptions:        | <p><b>1.0.E1 User did not log in successfully</b></p> <ol style="list-style-type: none"> <li>1. Users will need to re-checks or re-unchecks the box until they successfully logged in.</li> </ol> |                   |           |
| Priority:          | High  |                   |           |
| Frequency of Use:  | Almost every user may use this feature.   |                   |           |
| Business Rules:    | N/A   |                   |           |
| Other Information: | N/A   |                   |           |
| Assumptions:       | N/A   |                   |           |

1.5 Change Password:

a. UI Design

# Change Password

Current Password



Current Password

---

New Password



New Password

---

Confirm Password



Confirm Password

---

CHANGE

*b. Functionalities*

|                    |  |                   |           |
|--------------------|--|-------------------|-----------|
| UC ID and Name:    | LG-06 Change Password  |                   |           |
| Created By:        | TungPT   | Date Created:     | 20/5/2024 |
| Primary Actor:     | User   | Secondary Actors: | None      |
| Trigger:           | The user submits the change password form..  |                   |           |
| Description:       | As a user, I want to change my account's password.   |                   |           |
| Preconditions:     | PRE-1: The user must be logged in to their account.<br>PRE-2: The user is on the change password page  |                   |           |
| Postconditions:    | POST-1: The user's password in the database has been updated to the new password..<br>POST-2: The user receives a notification of the successful password change (e.g., a success message on the UI).  |                   |           |
| Normal Flow        | 1.The user click button Change Password in Profile page<br>2.User input information in field<br>3.Server check old password and new password.<br>4.Directly back login page.   |                   |           |
| Alternative Flows: | None   |                   |           |
| Exceptions:        | <b>2. The user enters an incorrect current password.</b><br><b>3a. The new password does not meet the required complexity criteria (e.g., minimum length, special characters, etc.).</b><br><b>3b. The new password and verify new password not match.</b> |                   |           |
| Priority:          | Medium   |                   |           |
| Frequency of Use:  | About average 100 access times per day   |                   |           |
| Business Rules:    | BR-1,BR-8  |                   |           |
| Other Information: | None   |                   |           |
| Assumptions:       | The user email is existed and have in the profile  |                   |           |

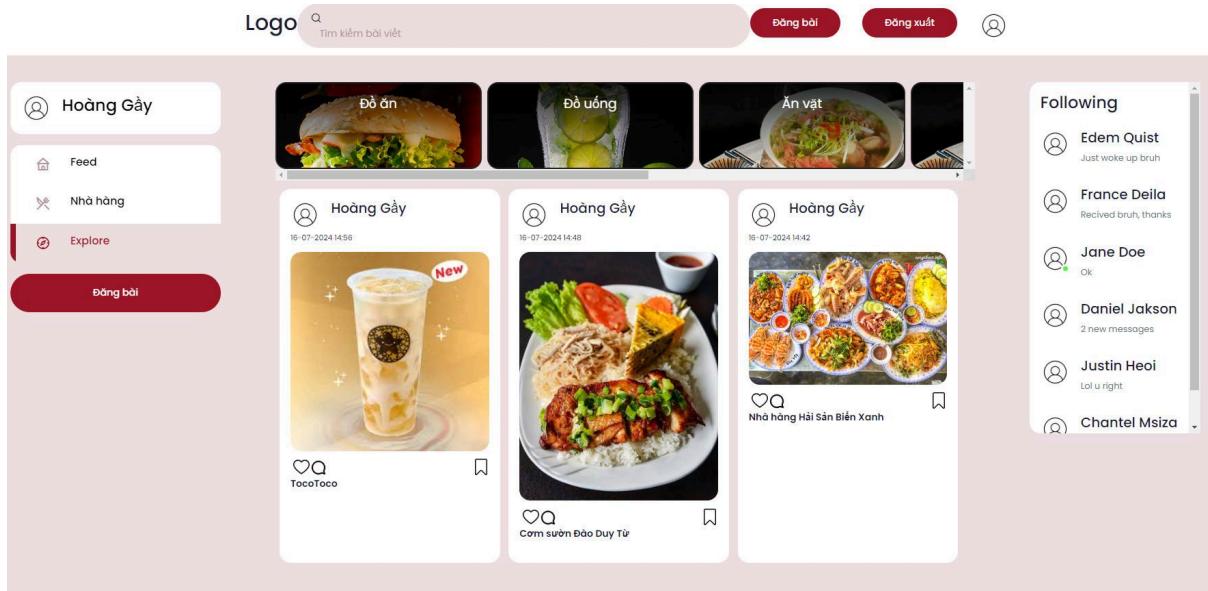
*c. Business Rules*

| ID   | Business Rule  | Business Rule Description                                   |
|------|--|---|
| BR-1 | Password Encoding  | User's password must be encoded with SHA-1 hashing          |
| BR-8 | Invalid Current Password, New Password, Confirm Password | User's current password aren't string(white space or blank) |

## 2. Patron Features

### 2.1 View Explore

#### a. UI Design



#### b. Functional Description

|                    |  |                |           |
|--------------------|--|----------------|-----------|
| ID and Name:       | UC-04 View Explore   |                |           |
| Created By:        |  | Date Created:  | 18/5/2024 |
| Primary Actor:     | All user   | Primary Actor: | All user  |
| Description:       | Everyone can access the Explore after successfully logging in or directly navigating to the Explore to explore available features. |                |           |
| Trigger:           | User successfully logs in, or<br>User directly accesses the Explore (by clicking the homepage link or entering the URL)            |                |           |
| Preconditions:     | User account has been created & authorized<br><br>User has successfully logged in, or the user has directly accessed the homepage  |                |           |
| Postconditions:    | + User views the Explore successfully  |                |           |
| Normal Flow:       | 1 User gain access to the website<br>2 User is redirected to the Explore.<br>3 The system display info : Posts                     |                |           |
| Alternative Flows: |  |                |           |
| Exceptions:        | The Error Message screen is shown to the user.   |                |           |
| Priority:          | High   |                |           |

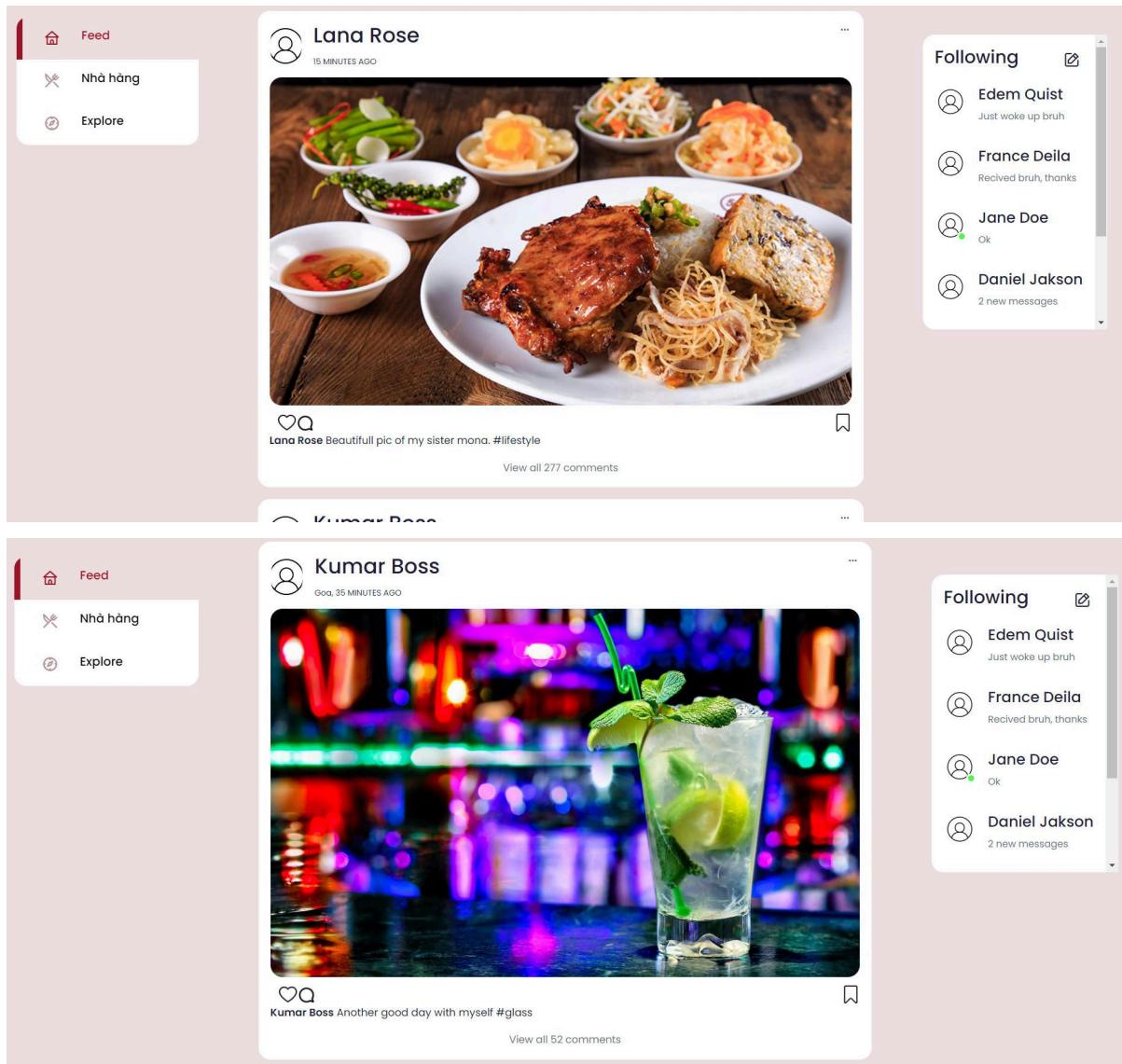
|                    |             |
|--------------------|-------------|
| Frequency of Use:  | Always      |
| Business Rules:    | BR-10,BR-11 |
| Other Information: |             |
| Assumptions:       |             |

*c. Business Rules*

| ID        | Business Rule | Business Rule Description   |
|-----------|---------------|---|
| BR-1<br>0 | Fact          | Search keywords cannot be empty or contain only spaces.   |
| BR-1<br>1 | Computations  | The system must automatically update posts after the user posts and the admin approves the post |

## 2.2 View Feed

### a. UI Design



### b. Functional Description

|                |  |                |           |
|----------------|--|----------------|-----------|
| ID and Name:   | UC-04 View Feed  |                |           |
| Created By:    |  | Date Created:  | 18/5/2024 |
| Primary Actor: | All user   | Primary Actor: | All user  |
| Description:   | Everyone can access the Feed after successfully logging in or directly navigating to the Feed to explore available features. |                |           |
| Trigger:       | User successfully logs in<br><br>User directly accesses the Feed (by clicking the homepage link or entering the URL)         |                |           |

|                    |  |
|--------------------|--|
| Preconditions:     | User account has been created & authorized<br>User has successfully logged in, or the user has directly accessed the Feed                        |
| Postconditions:    | + User views the Feed successfully   |
| Normal Flow:       | 1 User has access to the website<br>2 User is redirected to Explore.<br>3 User is redirected to the Feed .<br>4 Information display system: Post |
| Alternative Flows: |  |
| Exceptions:        | The Error Message screen is shown to the user.   |
| Priority:          | High   |
| Frequency of Use:  | Always   |
| Business Rules:    | BR-10,BR-11  |
| Other Information: |  |
| Assumptions:       |  |

*c. Business Rules*

| ID        | Business Rule | Business Rule Description   |
|-----------|---------------|---|
| BR-1<br>0 | Fact          | Search keywords cannot be empty or contain only spaces.   |
| BR-1<br>1 | Computations  | The system must automatically update posts after the user posts and the admin approves the post |

## 2.3 View Posts

### a. UI Design

**Hoàng Gầy**

16-07-2024 14:42

Phục vụ:

Giá cả:

Không gian:

Hương vị:

Về sinh:

Phục vụ:

Giá cả:

Không gian:

Hương vị:

Về sinh:

3.6/5 điểm

**Nhà hàng Hải Sản Biển Xanh**

Nhà hàng Hải Sản Biển Xanh nằm trên con phố nhộn nhịp của thành phố, nổi tiếng với thực đơn phong phú và tươi ngon. Với thiết kế hiện đại và không gian rộng rãi, nhà hàng tạo cảm giác thoải mái và sang trọng cho khách hàng. Ngay từ khi bước vào, tôi đã bị ấn tượng bởi không gian bài trí tinh tế và thoáng đãng. Nhà hàng sử dụng tông màu xanh biển nhẹ nhàng, kết hợp với nội thất gỗ tạo nên một không gian gần gũi mà không kém phần sang trọng. Các bàn ăn được bố trí hợp lý, đảm bảo sự riêng tư cho từng nhóm khách. Đặc biệt, nhà hàng có một khu vực ngoài trời với tầm nhìn ra biển, rất lý tưởng cho những buổi tối lãng mạn.

Thực đơn của Biển Xanh rất đa dạng, từ các món ăn truyền thống Việt Nam cho đến các món hải sản hiện đại. Tôi đã chọn thử một vài món tiêu biểu như tôm hùm nướng phô mai, cua rang me, và hàu nướng mỡ hành. Tôm hùm được nướng vừa tới, thịt tôm ngọt và dai, kết hợp với lớp phô mai béo ngậy, tạo nên một hương vị tuyệt vời. Cua được chế biến kỹ lưỡng, sot me chua ngọt thấm đều vào từng thớ thịt cua, khiến món ăn trở nên hấp dẫn hơn bao giờ hết. Hầu tươi sống, được nướng cùng mỡ hành và đậu phộng rang, mang đến một trải nghiệm âm thực đậm đà và đầy sức sống.

Nhân viên tại Biển Xanh rất nhiệt tình và chuyên nghiệp. Từ khi bước vào cửa, tôi đã được chào đón nồng nhiệt và được hướng dẫn đến bàn một cách nhanh chóng. Đội ngũ phục vụ luôn sẵn sàng hỗ trợ và đáp ứng mọi yêu cầu của khách hàng. Đặc biệt, tôi rất ấn tượng với sự am hiểu về món ăn và khả năng tư vấn của nhân viên.

**Hoàng Gầy**

Địa chỉ: Số 2 thị trấn Nga Sơn, huyện Nga Sơn, tỉnh Thanh Hóa  
Số điện thoại: 0979763115

**Hoàng Trương**

Địa chỉ: Số 2 thị trấn Nga Sơn, huyện Nga Sơn, tỉnh Thanh Hóa  
Số điện thoại: 0979763115

### b. Functional Description

| ID and Name:   | UC-05 View post content   |                |           |
|----------------|---|----------------|-----------|
| Created By:    |   | Date Created:  | 20/5/2024 |
| Primary Actor: | All user  | Primary Actor: | All user  |
| Description:   | Show post details (including title, author, updated date, category, and post details) |                |           |

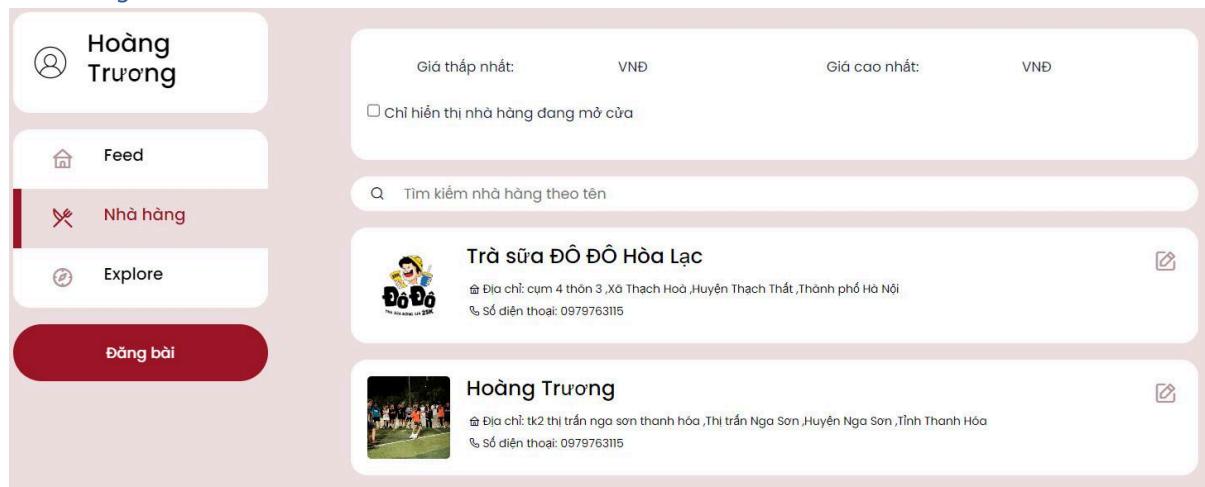
|                    |  |
|--------------------|--|
| Trigger:           | When a user clicks on “Posts” on the navigation bar.   |
| Preconditions:     | Display the Posts list screen to the user.   |
| Postconditions:    | Display the Posts detail screen to the user.   |
| Normal Flow:       | <p>1 Users access the original link of the website.</p> <p>2 Users log in with the role</p> <p>3 The system displays the home page after login.</p> <p>4 The system displays the Posts list screen.</p> <p>5 Users click on a Posts to see the details of that blog.</p> <p>6 The system redirect User to the page with all the information of that Posts loaded</p> |
| Alternative Flows: |  |
| Exceptions:        | The system redirect to error page if user try to access to deleted blog  |
| Priority:          | High   |
| Frequency of Use:  |  |
| Business Rules:    | BR-12,BR-13  |
| Other Information: |  |
| Assumptions:       |  |

### c. Business Rules

| ID    | Business Rule                   | Business Rule Description   |
|-------|---------------------------------|---|
| BR-12 | Accurate Display of Information | Contents(posts name , posts description, posts images ) in posts detail page must be accurate to the posts that user clicked on at the homepage |
| BR-13 |                                 | If the article was to be deleted , it must not be shown to other users under no circumstances   |

## 2.4 View Restaurant

### a. UI Design



### b. Functional Description

|                    |  |                |           |
|--------------------|--|----------------|-----------|
| ID and Name:       | UC-04 View Restaurant  |                |           |
| Created By:        |  | Date Created:  | 18/5/2024 |
| Primary Actor:     | All user   | Primary Actor: | All user  |
| Description:       | Everyone can access the Restaurant after successfully logging in to explore the available features.                                    |                |           |
| Trigger:           | User logged in successfully<br>Users access the Restaurant directly  |                |           |
| Preconditions:     | User account has been created & authorized<br>User has successfully logged in, or the user has directly accessed the Feed              |                |           |
| Postconditions:    | + The user viewed the Restaurant successfully  |                |           |
| Normal Flow:       | 1 User has access to the website<br>2 User logs in.<br>3 User is redirected to Restaurant.<br>4 Information display system: Restaurant |                |           |
| Alternative Flows: |  |                |           |
| Exceptions:        | The Error Message screen is shown to the user.   |                |           |
| Priority:          | High   |                |           |
| Frequency of Use:  | Always   |                |           |
| Business Rules:    | BR-7,BR-8  |                |           |

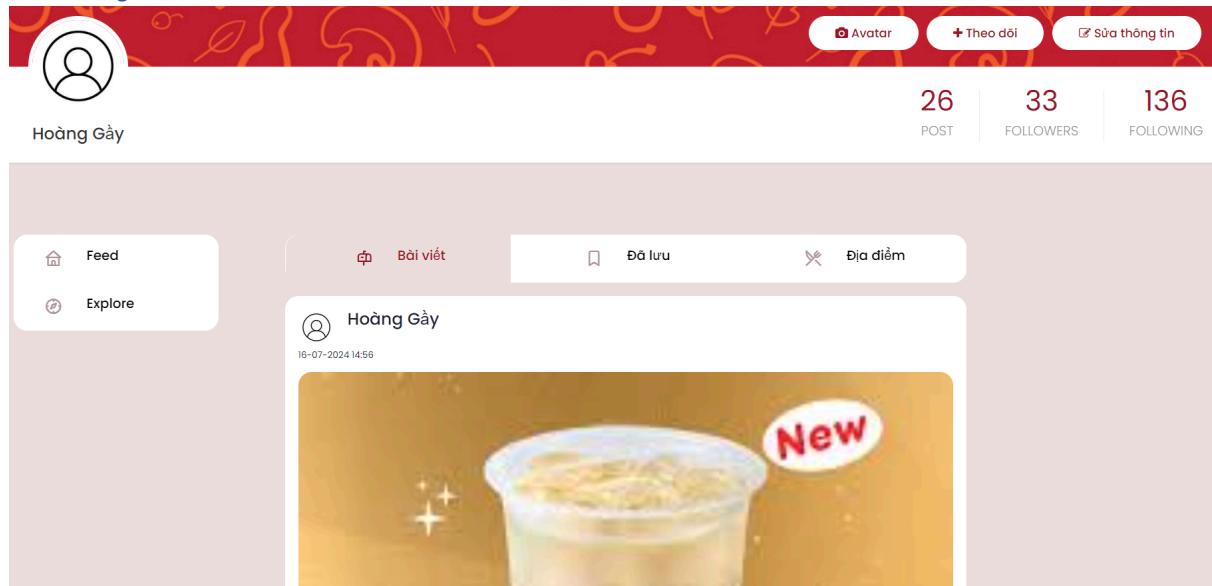
|                    |  |
|--------------------|--|
| Other Information: |  |
| Assumptions:       |  |

c. Business Rules

| ID | Business Rule | Business Rule Description |
|----|---------------|---------------------------|
|    |               |                           |
|    |               |                           |

## 2.5 View Profile

a. UI Design



b. Functionalities

|                 |  |                   |           |
|-----------------|--|-------------------|-----------|
| UC ID and Name: | View profile   |                   |           |
| Created By:     | Trương Việt Hoàng  | Date Created:     | 18/5/2024 |
| Primary Actor:  | All user   | Secondary Actors: | None      |
| Trigger:        | Register user clicks on User-profile pop-up screen                                       |                   |           |
| Description:    | Users can edit their personal information on their profile and view other users profiles |                   |           |
| Preconditions:  | User account has been created & authorized   |                   |           |
| Postconditions: | Their personal information and activities are displayed correctly and completely.        |                   |           |

|                    |   |
|--------------------|---|
| Normal Flow:       | <p>User Profile</p> <ol style="list-style-type: none"> <li>1. User accesses the User Login screen</li> <li>2. User clicks the Login button</li> <li>3. User accesses into User Profile screen</li> <li>4. Users view and edit personal information</li> </ol> |
| Alternative Flows: |   |
| Exceptions:        | <ol style="list-style-type: none"> <li>1. Username or password is incorrect</li> <li>2. Account is locked or disabled</li> </ol>  |
| Priority:          | High  |
| Frequency of Use:  | BR-15   |
| Business Rules:    |   |
| Other Information: |   |
| Assumptions:       |   |

*c. Business Rules*

| ID        | Business Rule       | Business Rule Description  |
|-----------|---------------------|--|
| BR-1<br>5 | Reporting and Admin | Users have the ability to report inappropriate or offensive content on other users' profiles, and administrators have the authority to take appropriate action |

### 3. Patron Features

#### 3.1 Create Post

##### a. UI Design

Tìm kiếm nhà hàng:

Nhập tên nhà hàng

Thêm bằng chứng đã đến quán (Bill có địa chỉ, ảnh chụp mặt tiền của quán)

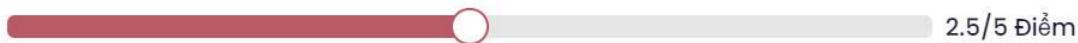
Chọn loại bữa ăn:

- Ăn sáng
- Ăn trưa
- Ăn tối
- Ăn vặt

Chọn loại thức ăn:

- Đồ ăn
- Đồ uống

Phục vụ:



Giá cả:



Không gian:



Hương vị:



Vệ sinh:



Điểm tổng quát:



Tiêu đề

Phục vụ:


2.5/5 Điểm

Giá cả:


2.5/5 Điểm

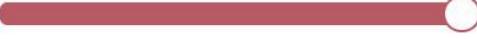
Không gian:


2.5/5 Điểm

Hương vị:


2.5/5 Điểm

Vệ sinh:


2.5/5 Điểm

Điểm tổng quát:



Tiêu đề

Nội dung

 Thêm ảnh/video

**Close** **Save changes**

#### b. Functional Description

|                 |  |                |         |
|-----------------|--|----------------|---------|
| ID and Name:    | <b>Create Post</b>   |                |         |
| Created By:     |  | Created On :   | /5/2024 |
| Primary Actor:  | Registered User  | Primary Actor: |         |
| Description:    | Allows users to create and publish a new post  |                |         |
| Trigger:        | When the user clicks on "Đăng Bài" button  |                |         |
| Preconditions:  | Users must create a restaurant before posting new articles   |                |         |
| Postconditions: | New posts are created by the restaurant and visible to all users   |                |         |
| Normal Flow:    | <ol style="list-style-type: none"> <li>1. Users access the original link of the website.</li> <li>2. Users log in with their credentials.</li> <li>3. The system displays the home page after login.</li> <li>4. Then the user goes to the location in the user profile to create a new restaurant</li> <li>5. Then click post, the system displays the post creation form.</li> </ol> |                |         |

|                    |  |
|--------------------|--|
|                    | <p>6. Users fill in the required details including title, category, and post details.</p> <p>7. Users click on the "Save Changes" button to create the post.</p> <p>8. The system saves the post and waits for admin approval.</p> |
| Alternative Flows: | <p>1. User press Explore while the creation process is happening</p> <p>2. The system redirects the user back to the home page or the previous page all the previous information user given in the process is removed</p>          |
| Exceptions:        | <p><b>Any fields are in the wrong format or missing:</b></p> <p>The system prompts the user to correct the fields and re-enter the information.</p>  |
| Priority:          | High   |
| Frequency of Use:  | Depends on user activity and content creation frequency  |
| Business Rules:    | BR-16, BR-17,  |
| Other Information: |  |
| Assumptions:       |  |

### c. Business Rules

| ID    | Business Rule                                  | Business Rule Description   |
|-------|--|---|
| BR-16 | No empty title                                 | The article title cannot be empty and must have at least one character.   |
| BR-17 | Redirect to Profile after successfully created | After creating a post, the system must redirect the user to the list of posts screen in user profile or the details screen of the newly created post. |
| BR-19 | Upload proof picture                           | User must enter proof picture in order to the admin can review the content  |

### 3.2 Update Profile

#### a. UI Design

Chỉnh sửa thông tin ×

Email

Tên

Số điện thoại

Giới tính  Nam  Nữ

Ngày sinh dd/mm/yyyy

Save

#### b. Functionalities

|                 |   |                   |          |
|-----------------|---|-------------------|----------|
| UC ID and Name: | Update Profile  |                   |          |
| Created By:     | Trương Việt Hoàng   | Date Created:     | 1/6/2024 |
| Primary Actor:  | Registered Users  | Secondary Actors: | None     |
| Trigger:        | Register user clicks on User-profile pop-up screen                |                   |          |
| Description:    | Users can view and edit their personal information.               |                   |          |
| Preconditions:  | User account has been created and authorized.                     |                   |          |
| Postconditions: | User's personal information are updated correctly and completely. |                   |          |

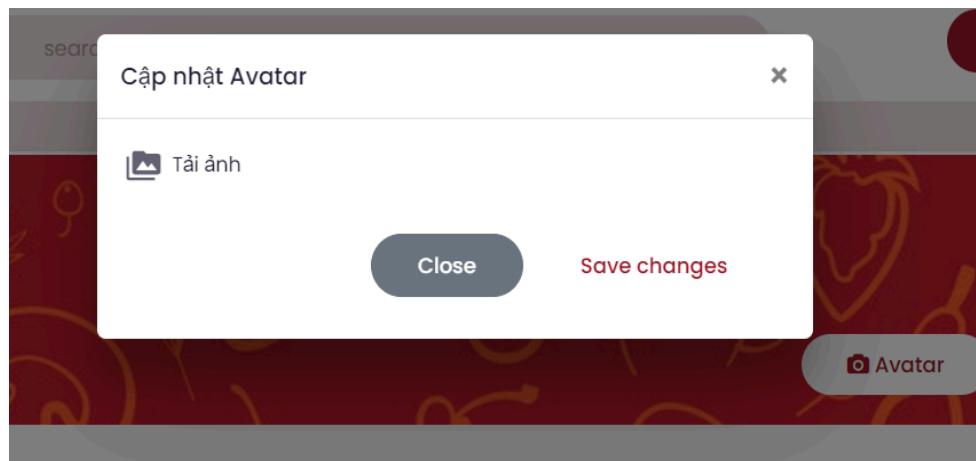
|                    |   |
|--------------------|---|
| Normal Flow:       | User accesses the User Profile edit screen<br>User navigates to the desired section for editing personal information<br>User makes desired changes to personal information<br>User saves the changes              |
| Alternative Flows: |   |
| Exceptions:        | Email or password is incorrect<br>System displays an error message and prompts the user to enter correct credentials<br>System displays a notification that the account is locked or disabled, preventing editing |
| Priority:          | High  |
| Frequency of Use:  | when users want to update their information   |
| Business Rules:    | BR-29, BR-18  |
| Other Information: |   |
| Assumptions:       |   |

*c. Business Rules*

| ID    | Business Rule                | Business Rule Description  |
|-------|------------------------------|--|
| BR-29 | Account Owner Authentication | Only the owner can change them self profile, post, restaurant                                    |
| BR-18 | Save Confirmation            | Changes made to the profile must be saved successfully for the update to be considered complete. |

### 3.3 Update Avatar

a. UI Design



#### b. Functionalities

|                    |   |                   |          |
|--------------------|---|-------------------|----------|
| UC ID and Name:    | Update Avatar   |                   |          |
| Created By:        | Trương Việt Hoàng   | Date Created:     | 1/6/2024 |
| Primary Actor:     | Registered Users  | Secondary Actors: | None     |
| Trigger:           | Registered users click on the update avatar pop-up screen   |                   |          |
| Description:       | Users can edit their personal avatar.   |                   |          |
| Preconditions:     | User account has been created and authorized.   |                   |          |
| Postconditions:    | The user's avatar was changed successfully  |                   |          |
| Normal Flow:       | User accesses the User Profile edit screen<br>User navigate to the avatar editing section to edit<br>User saves the changes |                   |          |
| Alternative Flows: |   |                   |          |
| Exceptions:        |   |                   |          |
| Priority:          | High  |                   |          |
| Frequency of Use:  | when users want to update avatar their information  |                   |          |
| Business Rules:    | BR-29   |                   |          |
| Other Information: |   |                   |          |
| Assumptions:       |   |                   |          |

#### c. Business Rules

|       |                              |   |
|-------|------------------------------|---|
| BR-29 | Account Owner Authentication | Only the owner can change them self profile, post, restaurant |
|-------|------------------------------|---|

### 3.4 Make Comment

#### a. UI Design



#### b. Functionalities

|                    |   |                   |           |
|--------------------|---|-------------------|-----------|
| UC ID and Name:    | Make Comment  |                   |           |
| Created By:        |   | Date Created:     | 25/5/2024 |
| Primary Actor:     | Registered User   | Secondary Actors: | None      |
| Trigger:           | User wants to comment on a post   |                   |           |
| Description:       | Users register to comment on posts. Users can provide their feedback, opinions, and likes related to the post content. Comments enhance engagement and interaction within the community.  |                   |           |
| Preconditions:     | User is logged in and has access to the post with commenting enabled  |                   |           |
| Postconditions:    | User's comment is successfully submitted and displayed on the post  |                   |           |
| Normal Flow:       | 1. User navigates to a post where commenting is enabled.<br>2. User reads through the post content.<br>3. User scrolls down to the comment section.<br>4. User enters their comment in the provided text box.<br>6. User clicks the "Gửi" to post the comment.<br>7. The system processes the comment and adds it to the comment section below the post.<br>8. User's comment is now visible to other users viewing the post. |                   |           |
| Alternative Flows: |   |                   |           |

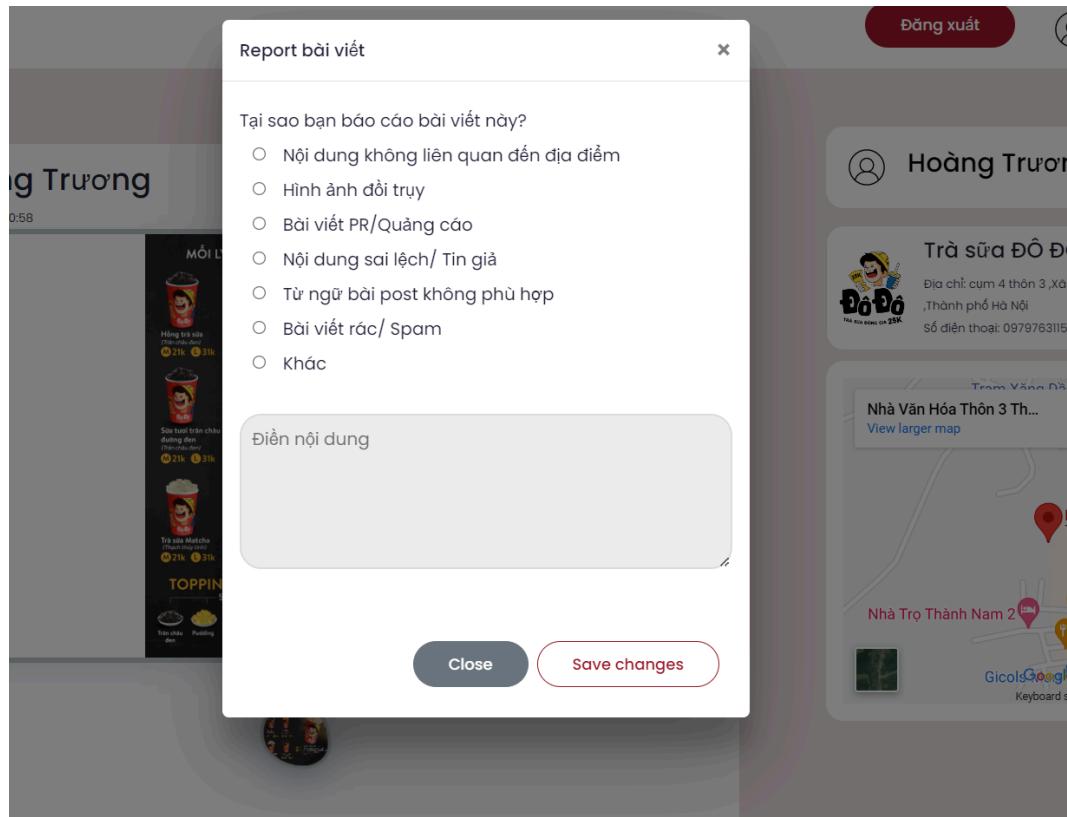
|                    |   |
|--------------------|---|
| Exceptions:        | User encounters technical issues while submitting the comment: The system displays an error message and does not allow current action to be executed. |
| Priority:          | High  |
| Frequency of Use:  |   |
| Business Rules:    | BR-18 ,BR-20  |
| Other Information: |   |
| Assumptions:       |   |

### c. Business Rules

| ID    | Business Rule                   | Business Rule Description  |
|-------|---------------------------------|--|
| BR-14 | User Authentication Requirement | Users must be authenticated (logged in) to interact with the website   |
| BR-20 | Respectful Communication        | Users are expected to engage in respectful and constructive communication when commenting on posts, avoiding offensive language or behavior. |

## 3.5 Report Post

### a. UI Design



*b. Functionalities*

|                    |   |                   |           |
|--------------------|---|-------------------|-----------|
| UC ID and Name:    | Report Post   |                   |           |
| Created By:        |   | Date Created:     | 30/5/2024 |
| Primary Actor:     | Registered User   | Secondary Actors: | None      |
| Trigger:           | User wants to report a post   |                   |           |
| Description:       | Users choose a reason to report the post or just give their own reason. This helps to eliminate inappropriate posts , creating a more friendly website  |                   |           |
| Preconditions:     | User is logged in and has access to the post with commenting and replying enabled   |                   |           |
| Postconditions:    | User's report is successfully submitted and displayed in the post report screen of Admin  |                   |           |
| Normal Flow:       | 1.User read the post and see that something is going wrong with the content<br>2.User clicks the report button<br>3.User choose report reason<br>4.User submit the report and there will be a notification saying that the submit was successful      |                   |           |
| Alternative Flows: | 1.User read the post and see that something is going wrong with the content<br>2.User clicks the report button<br>3.User write down another reason<br>4.User submit the report and there will be a notification saying that the submit was successful |                   |           |
| Exceptions:        | User encounters technical issues while submitting the report: The system displays an error message and does not allow current action to be executed.  |                   |           |
| Priority:          | High  |                   |           |
| Frequency of Use:  |   |                   |           |
| Business Rules:    | BR-14   |                   |           |
| Other Information: |   |                   |           |
| Assumptions:       |   |                   |           |

*c. Business Rules*

Provide the business rules those are applied only to the use case

| ID    | Business Rule                   | Business Rule Description                                   |
|-------|---------------------------------|---|
| BR-14 | User Authentication Requirement | Users must be authenticated (logged in) to report to posts. |

### 3.6 Edit Post

#### a. UI Design

The screenshot displays the mobile application's user interface for editing a post. At the top, there is a header bar with the text "Logo" and "Hoàng Gà". Below this, the main content area shows a detailed review for "Hoàng Gà". The review includes a profile picture of the reviewer, the date "16-07-2024 14:56", and the review title "Hoàng Gà". The review content is as follows:

- Thêm bảng chứng đã đến quán (Bill có địa chỉ, ảnh chụp mặt tiền của quán)**
- Chọn loại bữa ăn:**
  - Ăn sáng
  - Ăn trưa
  - Ăn tối
  - Ăn vặt
- Chọn loại thức ăn:**
  - Đồ ăn
  - Đồ uống
- Phục vụ:** 3.9/5 Điểm
- Giá cả:** 3.4/5 Điểm
- Không gian:** 3.6/5 Điểm
- Hương vị:** 4.1/5 Điểm
- Vệ sinh:** 3.9/5 Điểm
- Điểm tổng quát:** 3.8 ★

Below the review content, there is a section for adding images or videos, which currently displays two images of a drink. At the bottom right of the screen, there are "Close" and "Save changes" buttons.

#### b. Functionalities

|                 |           |               |           |
|-----------------|-----------|---------------|-----------|
| UC ID and Name: | Edit Post |               |           |
| Created By:     |           | Date Created: | 30/5/2024 |

|                    |   |                   |      |
|--------------------|---|-------------------|------|
| Primary Actor:     | Registered User   | Secondary Actors: | None |
| Trigger:           | Restaurant wants to edit a post   |                   |      |
| Description:       | Users choose to edit the post to create a more complete post.   |                   |      |
| Preconditions:     | The user is logged in and has access to the post for editing  |                   |      |
| Postconditions:    | The user's edit is successfully corrected and displayed on the screen.  |                   |      |
| Normal Flow:       | 1.The restaurant selects the posted article<br>2.Restaurant click on the three dots to edit<br>3.The restaurant edits the post<br>4. After editing, click "Save Changes" to save the Post |                   |      |
| Alternative Flows: |   |                   |      |
| Exceptions:        | The restaurant encountered technical problems while editing: The system displayed an error message and the editing could not be performed.  |                   |      |
| Priority:          | High  |                   |      |
| Frequency of Use:  |   |                   |      |
| Business Rules:    | BR-16,BR-21   |                   |      |
| Other Information: |   |                   |      |
| Assumptions:       |   |                   |      |

### c. Business Rules

Provide the business rules those are applied only to the use case

| ID    | Business Rule                             | Business Rule Description   |
|-------|---|---|
| BR-16 | No empty title                            | The article title cannot be empty and must have at least one character.   |
| BR-21 | Redirect to Profile after successful edit | After editing a post, the system must redirect the user to the post list screen in the user's profile or the detail screen of the newly created post. |

## 3.7 Like Post

### a. UI Design

|                 |           |
|-----------------|-----------|
| UC ID and Name: | Like Post |
|-----------------|-----------|

|                    |   |                   |           |
|--------------------|---|-------------------|-----------|
| Created By:        | Pham Thanh Tung   | Date Created:     | 30/5/2024 |
| Primary Actor:     | User  | Secondary Actors: | None      |
| Trigger:           | User views a post and everyone can like it.   |                   |           |
| Description:       | As a user, I want to be able to like a post so that I can show my appreciation or agreement with the content.   |                   |           |
| Preconditions:     | PRE-1: The user must be logged in to their account.<br>PRE-2: The post must exist and be visible on the profilePost page.   |                   |           |
| Postconditions:    | POST-1: The post's like count is incremented.<br>POST-2: The system stores the like action in the database.<br>POST-3: The icon like must change to show that user liked.   |                   |           |
| Normal Flow        | 1.The user clicks the "Like" button on a specific post.<br>2. The UI immediately changes the "Like" button to an active state (e.g., changing its color or icon to indicate the post has been liked).<br>3.The client-side JavaScript sends an AJAX request to the server to record the like show all like and add in database<br>4. If the user clicks the "Like" button on a specific post again.<br>5.The UI immediately changes the "Like" button to an active state (e.g., changing its color or icon to indicate the post has been unlike).<br>6.The client-side JavaScript sends an AJAX request to the server to record the unlike show all like in post and remove in database |                   |           |
| Alternative Flows: | None  |                   |           |
| Exceptions:        |   |                   |           |
| Priority:          | Medium  |                   |           |
| Frequency of Use:  | About average 100 access times per day  |                   |           |
| Business Rules:    | BR-14   |                   |           |
| Other Information: | None  |                   |           |
| Assumptions:       | The user email is existed and have in the profile   |                   |           |

*c. Business Rules*

|       |                              |   |
|-------|------------------------------|---|
| BR-29 | Account Owner Authentication | Only the owner can change them self profile, post, restaurant |
|-------|------------------------------|---|

**3.8 Save Post**

*a. UI Design*

*b. Functionalities*

|                    |  |                   |           |
|--------------------|--|-------------------|-----------|
| UC ID and Name:    | <b>Save Post</b>   |                   |           |
| Created By:        | Pham Thanh Tung  | Date Created:     | 30/5/2024 |
| Primary Actor:     | User   | Secondary Actors: | None      |
| Trigger:           | User views a post and everyone can save it to see it again.  |                   |           |
| Description:       | As a user, I want to be able to save a post so that I can show my appreciation or agreement with the content.  |                   |           |
| Preconditions:     | PRE-1: The user must be logged in to their account.<br>PRE-2: The post must exist and be visible on the profilePost page.  |                   |           |
| Postconditions:    | POST-1: The icon save must change to show that user saved.<br>POST-2: The system stores the save action in the database.   |                   |           |
| Normal Flow        | 1.The user clicks the "Save" button on a specific post.<br>2.The UI immediately changes the "Save" button to an active state (e.g., changing its color or icon to indicate the post has been saved).<br>3.The client-side JavaScript sends an AJAX request to the server to record the save<br>4. If the user clicks the "Save" button on a specific post again.<br>5.The UI immediately changes the "Save" button to an active state (e.g., changing its color or icon to indicate the post has been unsaved).<br>3.The client-side JavaScript sends an AJAX request to the server to record the unsave |                   |           |
| Alternative Flows: | None   |                   |           |
| Exceptions:        |  |                   |           |
| Priority:          | Medium   |                   |           |
| Frequency of Use:  | About average 100 access times per day   |                   |           |
| Business Rules:    | BR-14  |                   |           |
| Other Information: | None   |                   |           |
| Assumptions:       | The user email is existed and have in the profile  |                   |           |

### c. Business Rules

|       |                                 |  |
|-------|---------------------------------|--|
| BR-14 | User Authentication Requirement | Users must be authenticated (logged in) to interact with the website |
|-------|---------------------------------|--|

### 3.9 Follow User

#### a. UI Design



*b. Functionalities*

|                    |  |                   |           |
|--------------------|--|-------------------|-----------|
| UC ID and Name:    | Follow User  |                   |           |
| Created By:        | PhongTT  | Date Created:     | 30/5/2024 |
| Primary Actor:     | User   | Secondary Actors: | None      |
| Trigger:           | User views another user's profile and decides to follow them.  |                   |           |
| Description:       | As a user, I want to be able to follow another user so that I can receive updates about their posts and activities.  |                   |           |
| Preconditions:     | PRE-1: The user must be logged in to their account.<br>PRE-2: The target user (the one being followed) must exist  |                   |           |
| Postconditions:    | POST-1: The target user's follower count is incremented.<br>POST-2: The system stores the follow action in the database.<br>POST-3: The target user's following count is incremented.  |                   |           |
| Normal Flow        | 1.The user click button follow in Profile page<br>2.The UI immediately changes the "Follow" button to an active state (e.g., changing its label to "Following"). Increment following below follow button.<br>3.The client-side JavaScript sends an AJAX request to the server to record the follow action<br>4.If the user click button follow in Profile page again<br>5.The UI immediately changes the "Follow" button to an active state (e.g., changing its label to "Follow"). Decrement following below follow button.<br>6.The client-side JavaScript sends an AJAX request to the server to record the follow action |                   |           |
| Alternative Flows: | None   |                   |           |
| Exceptions:        |  |                   |           |
| Priority:          | Medium   |                   |           |
| Frequency of Use:  | About average 100 access times per day   |                   |           |
| Business Rules:    | BR-14  |                   |           |

|                    |   |
|--------------------|---|
| Other Information: | None  |
| Assumptions:       | The user email is existed and have in the profile |

### c. Business Rules

| ID    | Business Rule              | Business Rule Description  |
|-------|----------------------------|--|
| BR-14 | Authentication Requirement | Users must be authenticated (logged in) to view their own profile or profiles of other users |

## 3.10 Add Restaurant

### a. UI Design

Thêm địa điểm

Tên địa điểm: \_\_\_\_\_

Địa chỉ: \_\_\_\_\_

Tỉnh Thành  
Quận Huyện  
Phường Xã  
[Số nhà, tên đường]



Thêm bằng chứng đã đến quán (Bill có địa chỉ ảnh chụp mặt tiền của quán)

Số điện thoại: \_\_\_\_\_

Giá thấp nhất(VND): \_\_\_\_\_

Giá cao nhất(VND): \_\_\_\_\_

Khung giờ: \_\_\_\_\_ - \_\_\_\_\_

Thêm ảnh đại diện

### b. Functionalities

|                 |                       |                   |           |
|-----------------|-----------------------|-------------------|-----------|
| UC ID and Name: | <b>Add restaurant</b> |                   |           |
| Created By:     | PhongTT               | Date Created:     | 30/5/2024 |
| Primary Actor:  | User                  | Secondary Actors: | None      |

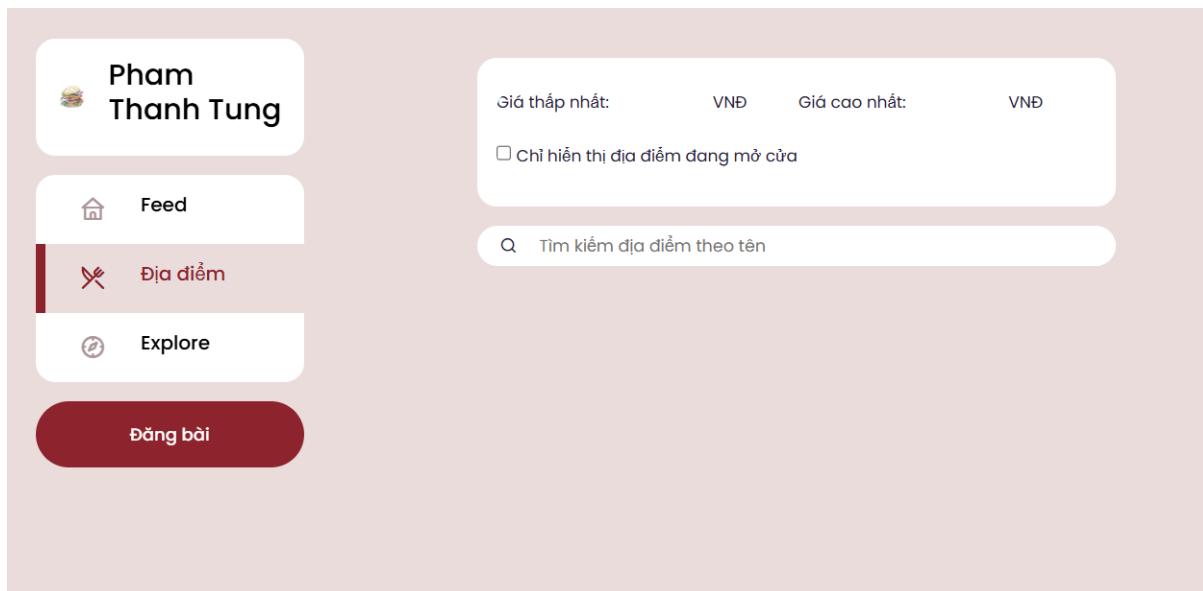
|                    |  |
|--------------------|--|
| Trigger:           | User wants to add a new restaurant to the system.  |
| Description:       | As a user, I want to add new restaurant to review about this restaurant.   |
| Preconditions:     | PRE-1: The user must be logged in to their account.  |
| Postconditions:    | POST-1: A new restaurant is added to the system and is available for review.<br>POST-2: A new restaurant is approved by admin.   |
| Normal Flow        | 1.In Profile of user, click add restaurant function<br>2.User input information in field<br>3.User submits the form to add the restaurant.<br>4.Return to profile - location page. |
| Alternative Flows: | None   |
| Exceptions:        | <b>1.Invalid information. System display a message prompts user try again</b>  |
| Priority:          | Medium   |
| Frequency of Use:  | About average 100 access times per day   |
| Business Rules:    | BR-14  |
| Other Information: | None   |
| Assumptions:       |  |

### c. Business Rules

|       |                                 |  |
|-------|---------------------------------|--|
| BR-14 | User Authentication Requirement | Users must be authenticated (logged in) to interact with the website |
|-------|---------------------------------|--|

## 3.11 Search Restaurant

### a. UI Design



*b. Functionalities*

|                    |   |  |                   |           |
|--------------------|---|--|-------------------|-----------|
| UC ID and Name:    | <b>Search restaurant</b>  |  |                   |           |
| Created By:        | PhongTT   |  | Date Created:     | 30/5/2024 |
| Primary Actor:     | User  |  | Secondary Actors: | Guest     |
| Trigger:           | User wants to search restaurant.  |  |                   |           |
| Description:       | As a user, I want to search restaurant to view information about this restaurant.   |  |                   |           |
| Precondition S:    | PRE-1: The restaurant must be created.  |  |                   |           |
| Postconditions:    | POST-1: The user successfully views the search results for the restaurant.  |  |                   |           |
| Normal Flow        | 1. Navigate to restaurant page<br>2. Input name restaurant want to find.<br>3. Use filter to find restaurant<br>4. Display list restaurant. |  |                   |           |
| Alternative Flows: | None  |  |                   |           |
| Exceptions:        |   |  |                   |           |
| Priority:          | Medium  |  |                   |           |
| Frequency of Use:  | About average 100 access times per day  |  |                   |           |
| Business Rules:    | BR-10   |  |                   |           |
| Other Information: | None  |  |                   |           |

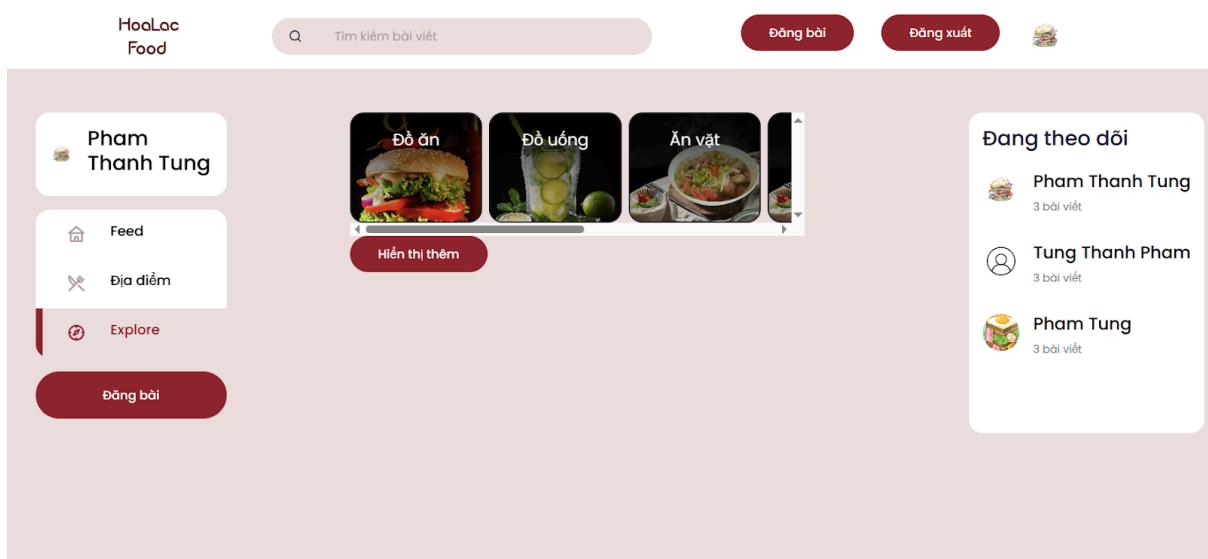
|               |  |
|---------------|--|
| Assumptions : |  |
|---------------|--|

### c. Business Rules

|       |      |   |
|-------|------|---|
| BR-10 | Fact | Search keywords cannot be empty or contain only spaces. |
|-------|------|---|

## 3.12 Search Post

### a. UI Design



### b. Functionalities

| UC ID and Name: | Search Post  |                   |           |
|-----------------|--|-------------------|-----------|
| Created By:     | HoangTV  | Date Created:     | 30/5/2024 |
| Primary Actor:  | User   | Secondary Actors: | Guest     |
| Trigger:        | User wants to search for a post about a restaurant   |                   |           |
| Description:    | As a user, I want to search for posts about specific restaurants to read reviews and gather information. |                   |           |
| Precondition S: | PRE-1: The post must be created.   |                   |           |
| Postconditions: | POST-1: The user successfully views the search results for the post.                                     |                   |           |

|                    |   |
|--------------------|---|
| Normal Flow        | 1. Navigate in home page<br>2. Input keyword about post<br>3. Use filters to find posts following categories such as: food, drink,...<br>4. Display list post . |
| Alternative Flows: | None  |
| Exceptions:        | <b>System error while processing the search. System displays an error message and prompts the user to try again later.</b>                                      |
| Priority:          | Medium  |
| Frequency of Use:  | About average 100 access times per day  |
| Business Rules:    | BR-10   |
| Other Information: | None  |
| Assumptions :      |   |

*c. Business Rules*

| ID    | Business Rule | Business Rule Description                               |
|-------|---------------|---|
| BR-10 | Fact          | Search keywords cannot be empty or contain only spaces. |

**3.13 Edit Restaurant**

*a. UI Design*

*b. Functionalities*

|                 |  |  |                   |           |
|-----------------|--|--|-------------------|-----------|
| UC ID and Name: | <b>Edit Restaurant</b>   |  |                   |           |
| Created By:     | PhongTT  |  | Date Created:     | 30/5/2024 |
| Primary Actor:  | User   |  | Secondary Actors: | None      |
| Trigger:        | User wants to edit restaurant information  |  |                   |           |
| Description:    | As a user, I want to edit restaurant information after seeing it wrong or if the restaurant had changed any information. |  |                   |           |
| Precondition s: | PRE-1: The restaurant must be created.<br>PRE-2: The User must logged in successful                                      |  |                   |           |
| Postconditions: | POST-1: The restaurant information is changed and display.<br>POST-2: The data of restaurant update in database          |  |                   |           |
| Normal Flow     | 1. Navigate in restaurant page   |  |                   |           |

|                    |  |
|--------------------|--|
|                    | 2.Click restaurant detail want to edit<br>3. Change information<br>4.Click button change .   |
| Alternative Flows: | None   |
| Exceptions:        | <b>1.User inputs invalid information. System displays a validation error and prompts the user to correct the input.</b><br><b>2.System error while saving the changes. System displays an error message and prompts the user to try again later.</b> |
| Priority:          | Medium   |
| Frequency of Use:  | About average 100 access times per day   |
| Business Rules:    | BR-29  |
| Other Information: | None   |
| Assumptions :      |  |

*c. Business Rules*

|       |                              |   |
|-------|------------------------------|---|
| BR-29 | Account Owner Authentication | Only the owner can change them self profile, post, restaurant |
|-------|------------------------------|---|

**3.14 Report Restaurant**

*a. UI Design*

*b. Functionalities*

| UC ID and Name: | Report Restaurant  |                   |           |
|-----------------|--|-------------------|-----------|
| Created By:     | HoangNM  | Date Created:     | 30/5/2024 |
| Primary Actor:  | User   | Secondary Actors: | None      |
| Trigger:        | User wants to report restaurant had the information incorrect                        |                   |           |
| Description:    | As a user, I want to report if a restaurant had wrong information or does not exist. |                   |           |

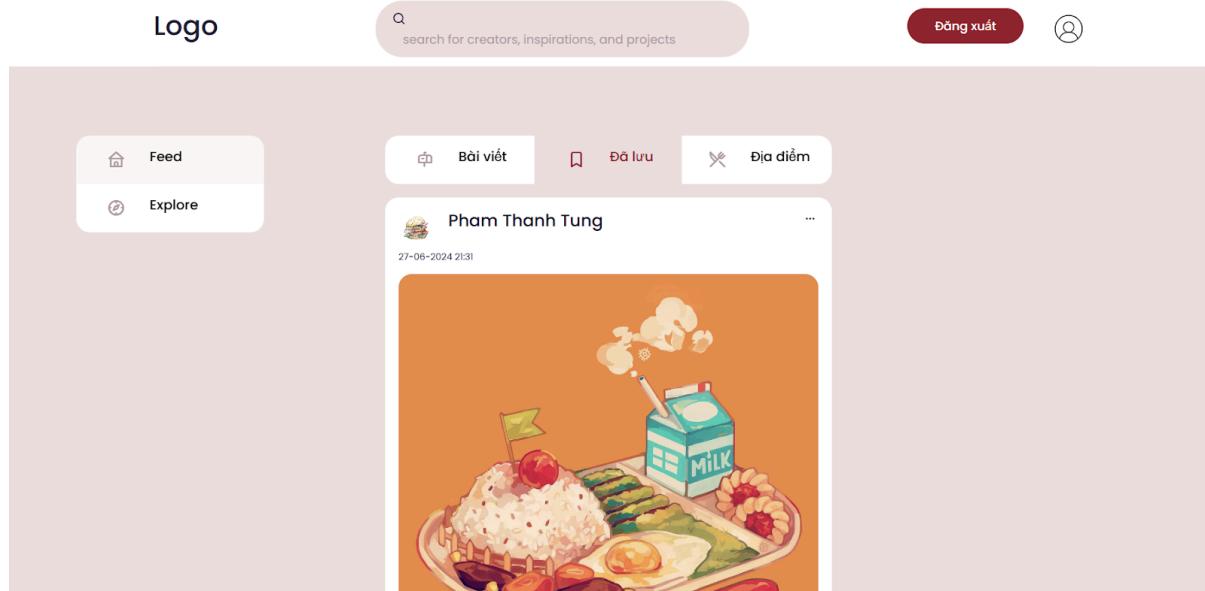
|                    |  |
|--------------------|--|
| Preconditions:     | PRE-1: The restaurant must be created.<br>PRE-2: The User must be logged in successfully   |
| Postconditions:    | POST-1: The reported restaurant is flagged for review..<br>POST-2: Admin reviews the report and takes necessary action (corrects the information or dismisses the report).   |
| Normal Flow        | <ol style="list-style-type: none"> <li>1. Navigate to restaurant page</li> <li>2. Click restaurant detail want to report</li> <li>3. Click icon report</li> <li>4. Add information to verify information want to report.</li> <li>5. Report sent to admin</li> <li>6. Admin checks information and approves or rejects.</li> </ol> |
| Alternative Flows: | None   |
| Exceptions:        |  |
| Priority:          | Medium   |
| Frequency of Use:  | About average 100 access times per day   |
| Business Rules:    | BR-14  |
| Other Information: | None   |
| Assumptions :      |  |

#### *c. Business Rules*

|       |                                 |  |
|-------|---------------------------------|--|
| BR-14 | User Authentication Requirement | Users must be authenticated (logged in) to interact with the website |
|-------|---------------------------------|--|

### 3.15 View Saved Post

#### a. UI Design



#### b. Functionalities

| UC ID and Name:    | View Saved Post   |                   |           |
|--------------------|---|-------------------|-----------|
| Created By:        | TungPT  | Date Created:     | 30/5/2024 |
| Primary Actor:     | User  | Secondary Actors: | None      |
| Trigger:           | User wants to view the post saved in the past   |                   |           |
| Description:       | As a user, I want to view some posts they saved in the past.  |                   |           |
| Precondition s:    | PRE-1: User must save at least one post.<br>PRE-2: The User must logged in successful   |                   |           |
| Postconditions:    | POST-1: The user successfully views their saved posts.  |                   |           |
| Normal Flow        | 1. Navigate in profile user page<br>2.Click post saved page<br>3.System displays a list of posts that the user has saved.<br>4.User clicks on a specific post to view its details.<br>5.System displays the details of the selected post. |                   |           |
| Alternative Flows: | None  |                   |           |
| Exceptions:        |   |                   |           |
| Priority:          | Medium  |                   |           |

|                    |  |
|--------------------|--|
| Frequency of Use:  | About average 100 access times per day |
| Business Rules:    | BR-14                                  |
| Other Information: | None                                   |
| Assumptions :      |  |

c. Business Rules

|       |                             |  |
|-------|-----------------------------|--|
| BR-30 | Account Owner Accessibility | Only the owner can change can view their saved posts |
|-------|-----------------------------|--|

3.16 Like comment

a. UI Design

test quan com

quan com tam

Cơm Thố Bách Khoa

Địa chỉ: 2G49+7C2, Thạch Hoà, Thạch Thất, Hà Nội, Vietnam, 0,0,0  
Số điện thoại: 0987123454

Viết bình luận...

Gửi

Pham Thanh Tung 08-07-2024 11:58  
abc

55

b. Functionalities

|                    |   |                   |           |
|--------------------|---|-------------------|-----------|
| UC ID and Name:    | <b>View Saved Post</b>  |                   |           |
| Created By:        | TungPT  | Date Created:     | 30/5/2024 |
| Primary Actor:     | User  | Secondary Actors: | None      |
| Trigger:           | User wants to like a comment  |                   |           |
| Description:       | As a user, I want to like comments on posts to show my appreciation or agreement with the comment.            |                   |           |
| Preconditions:     | PRE-1: The comment must exist on a post.<br>PRE-2: The User must logged in successful                         |                   |           |
| Postconditions:    | POST-1: The comment is liked by the user.<br>POST-2: The like count on the comment is incremented by one.     |                   |           |
| Normal Flow        | 1. In list post, click on a post detail<br>2.Move to comment function<br>3.Click icon heart to like a comment |                   |           |
| Alternative Flows: | None  |                   |           |
| Exceptions:        | <b>User is not logged in and tries to like a comment. System prompts the user to log in.</b>                  |                   |           |
| Priority:          | Medium  |                   |           |
| Frequency of Use:  | About average 100 access times per day  |                   |           |
| Business Rules:    | BR-14   |                   |           |
| Other Information: | None  |                   |           |
| Assumptions :      |   |                   |           |

### c. Business Rules

|       |                                 |  |
|-------|---------------------------------|--|
| BR-14 | User Authentication Requirement | Users must be authenticated (logged in) to interact with the website |
|-------|---------------------------------|--|

## 4. Admin Features

### 4.1 Dashboard

#### a. UI Design

The screenshot shows the Admin Dashboard interface. On the left, there is a sidebar with the following navigation links:

- Trang quản lý
- Quản lý người dùng
- Quản lý nhà hàng
- Yêu cầu tạo bài viết
- Yêu cầu tạo nhà hàng
- Yêu cầu chỉnh sửa bài viết
- Yêu cầu chỉnh sửa nhà hàng
- Yêu cầu báo cáo bài viết
- Yêu cầu báo cáo nhà hàng

The main dashboard area displays the following metrics in cards:

| TỔNG SỐ TÀI KHOẢN NGƯỜI DÙNG | TỔNG SỐ NHÀ HÀNG | YÊU CẦU ĐĂNG BÀI VIẾT | YÊU CẦU TẠO NHÀ HÀNG |
|------------------------------|------------------|-----------------------|----------------------|
| 1                            | 7                | 3                     | 1                    |

| YÊU CẦU CHỈNH SỬA BÀI VIẾT | YÊU CẦU CHỈNH SỬA NHÀ HÀNG | YÊU CẦU BÁO CÁO BÀI VIẾT | YÊU CẦU BÁO CÁO NHÀ HÀNG |
|----------------------------|----------------------------|--------------------------|--------------------------|
| 0                          | 0                          | 2                        | 0                        |

#### b. Functionalities

|                    |   |                   |            |
|--------------------|---|-------------------|------------|
| UC ID and Name:    | Admin Dashboard   |                   |            |
| Created By:        | HoangNM   | Date Created:     | 02/07/2024 |
| Primary Actor:     | Admin   | Secondary Actors: | None       |
| Trigger:           | Admin wants to view the static of the website   |                   |            |
| Description:       | Admin dashboard serves the purpose of displaying all the static of the website including the total users, restaurants, post requests, restaurant requests, edit post and restaurant requests, post and restaurant report. |                   |            |
| Preconditions:     | Admin is logged in to the system  |                   |            |
| Postconditions:    | The total number of each category is displayed to the Admin   |                   |            |
| Normal Flow:       | 1. Log in to the system<br>2. Admin access the DashBoard Page   |                   |            |
| Alternative Flows: |   |                   |            |
| Exceptions:        | Admin encounters technical issues while accessing the interface<br>System displays an error message and prompts the admin to try again later  |                   |            |
| Priority:          | High  |                   |            |
| Frequency of Use:  |   |                   |            |

|                    |                   |
|--------------------|-------------------|
| Business Rules:    | BR-22,BR-23,BR-24 |
| Other Information: |                   |
| Assumptions:       |                   |

*c. Business Rules*

|       |                                  |   |
|-------|----------------------------------|---|
| BR-22 | Admin Authentication Requirement | Admins must be authenticated (logged in) to access post review functionality.                                       |
| BR-23 | Admin Access Levels              | Different access levels may exist for Admins, allowing varying degrees of control over post review and publication. |
| BR-24 | Admin Page Access Confirmation   | After accessing the Admin page, Admins must confirm their identity if prompted by the system for security reasons.  |

## 4.2 Manage User

*a. UI Design*

| Users                       |      |        |                |                       |   |  |  Roles |
|-----------------------------|------|--------|----------------|-----------------------|---|--|---|
| Email                       | Role | Status | Total of posts | Total of hidden posts |   |  |   |
| hoangtvhe163201@fpt.edu.vn  | 1    | Active | 4              | 4                     |  |  |   |
| truongviethoang64@gmail.com | 1    | Active | 1              | 1                     |  |  |   |
|                             |      |        | Previous       | 1                     | Next  |  |   |

\*View user information

**ID tài khoản** 2

**Tên tài khoản** Hoang Nguyen

**Email** dixec38797@qodiq.com

**Số điện thoại** 0795122467

**Giới tính**  Nam  Nữ

**Ngày sinh** 07/31/2024

**Trạng thái**  Active  Disable

**Save**

#### b. Functionalities

|                 |  |                   |            |
|-----------------|--|-------------------|------------|
| UC ID and Name: | Manage User  |                   |            |
| Created By:     | HoangNM  | Date Created:     | 03/07/2024 |
| Primary Actor:  | Admin  | Secondary Actors: | None       |
| Trigger:        | Admin wants to manage all the registered users on the platform   |                   |            |
| Description:    | Admin accesses the user statistics interface to view the list of registered users and admin can view user information and change their status.   |                   |            |
| Preconditions:  | Admin is logged in to the system<br>Admin has the necessary permissions to view user statistics  |                   |            |
| Postconditions: | The total number of registered users is displayed to the Admin   |                   |            |
| Normal Flow:    | 1. Admin log in to the system<br>2.The system verifies the Admin's authentication and permissions<br>3. Admin access the DashBoard Page<br>4.Admin accesses the "User" interface button<br>5.The system show list of registered users and some basic information |                   |            |

|                    |   |
|--------------------|---|
|                    | 6.Admin can view each user information and change status of the account   |
| Alternative Flows: |   |
| Exceptions:        | <p>Admin encounters technical issues while accessing the interface</p> <p>System displays an error message and prompts the admin to try again later</p> |
| Priority:          | High  |
| Frequency of Use:  |   |
| Business Rules:    | BR-22,BR-23   |
| Other Information: |   |
| Assumptions:       |   |

### c. Business Rules

| ID    | Business Rule                    | Business Rule Description  |
|-------|----------------------------------|--|
| BR-22 | Admin Authentication Requirement | Admin must be authenticated (logged in) to access post review functionality.                         |
| BR-23 | Admin Access Levels              | There can be different access levels for Admins, allowing for varying degrees of control over users. |

## 4.3 Manage Restaurant

### a. UI Design

| Danh sách nhà hàng |               |              |                |              |  |
|--------------------|---------------|--------------|----------------|--------------|--|
| ID nhà hàng        | ID người đăng | Tên nhà hàng | Trạng thái     | Xem nhà hàng |  |
| 1                  | 1             | z            | Dang chờ duyệt | Xem          |  |
| 2                  | 1             | q            | Báo cáo        | Xem          |  |
| 3                  | 1             | q            | Báo cáo        | Xem          |  |
| 4                  | 1             | asd          | Báo cáo        | Xem          |  |
| 5                  | 1             | asd          | Báo cáo        | Xem          |  |
| 6                  | 1             | zxc          | Hoạt động      | Xem          | <button>Đổi trạng thái nhà hàng</button> |
| 7                  | 1             | zxc          | Dang chờ duyệt | Xem          |  |
| 8                  | 2             | zxc          | Hoạt động      | Xem          | <button>Đổi trạng thái nhà hàng</button> |
| 9                  | 2             | zxc          | Dang chờ duyệt | Xem          |  |

### b. Functionalities

|                 |                   |
|-----------------|-------------------|
| UC ID and Name: | Manage Restaurant |
|-----------------|-------------------|

|                    |   |                   |            |
|--------------------|---|-------------------|------------|
| Created By:        | HoangNM   | Date Created:     | 03/07/2024 |
| Primary Actor:     | Admin   | Secondary Actors: | None       |
| Trigger:           | Admin wants to manage restaurants in the system.  |                   |            |
| Description:       | Admin can view list of and status of each restaurants in the system and have ability to change restaurant's status  |                   |            |
| Preconditions:     | Admin is logged in<br>Admin has the necessary permissions to view restaurant statistics   |                   |            |
| Postconditions:    | The list of restaurant is displayed to the Admin  |                   |            |
| Normal Flow:       | 1. Admin log in to the system<br>2. The system verifies the Admin's authentication and permissions<br>3. Admin access the DashBoard Page<br>4. Admin accesses the "Restaurant List" interface button<br>5. The system show list of restaurant in the system and some basic information<br>6. Admin can view each restaurant information and change status of the restaurant |                   |            |
| Alternative Flows: |   |                   |            |
| Exceptions:        | Admin encounters technical issues while accessing the interface<br>System displays an error message and prompts the admin to try again later  |                   |            |
| Priority:          | High  |                   |            |
| Frequency of Use:  |   |                   |            |
| Business Rules:    | BR-22, BR-23  |                   |            |
| Other Information: |   |                   |            |
| Assumptions:       |   |                   |            |

### c. Business Rules

| ID    | Business Rule                    | Business Rule Description  |
|-------|----------------------------------|--|
| BR-22 | Admin Authentication Requirement | Admin must be authenticated (logged in) to access post review functionality.                         |
| BR-23 | Admin Access Levels              | There can be different access levels for Admins, allowing for varying degrees of control over users. |

## 4.4 Process Post Request

### a. UI Design

| Yêu cầu tạo bài viết |              |             |                  |              |           |         |
|----------------------|--------------|-------------|------------------|--------------|-----------|---------|
| ID bài viết          | ID tài khoản | ID nhà hàng | Ngày đăng        | Xem bài viết | ...       |         |
| 8                    | 1            | 6           | 26-07-2024 02:05 | Xem          | Chấp nhận | Từ chối |
| 9                    | 1            | 1           | 29-07-2024 22:57 | Xem          | Chấp nhận | Từ chối |
| 12                   | 1            | 1           | 30-07-2024 02:24 | Xem          | Chấp nhận | Từ chối |

\*When click “Từ chối”

| Yêu cầu tạo bài viết |              |             |                  |              |           |         |
|----------------------|--------------|-------------|------------------|--------------|-----------|---------|
| ID bài viết          | ID tài khoản | ID nhà hàng | Ngày đăng        | Xem bài viết | ...       |         |
| 8                    | 1            | 6           | 26-07-2024 02:05 | Xem          | Chấp nhận | Từ chối |
| 9                    | 1            | 1           |                  |              | Chấp nhận | Từ chối |
| 12                   | 1            | 1           |                  |              | Chấp nhận | Từ chối |

### b. Functionalities

|                 |   |                   |           |
|-----------------|---|-------------------|-----------|
| UC ID and Name: | Process Post Request  |                   |           |
| Created By:     | HoangNM   | Date Created:     | 28/6/2024 |
| Primary Actor:  | Admin   | Secondary Actors: | None      |
| Trigger:        | Admin wants to review and process a submitted post            |                   |           |
| Description:    | Admin reviews and approves or denies posts submitted by users |                   |           |

|                    |   |
|--------------------|---|
| Preconditions:     | Admin is logged in and has access to the post review function   |
| Postconditions:    | Post is either approved and published or rejected and remains unpublished   |
| Normal Flow:       | <ol style="list-style-type: none"> <li>1. Admin accesses the post request to view the requested article</li> <li>2. Admin selects a post for review</li> <li>3. Admin reads through the post content and evaluates its quality</li> <li>4. Admin decides to either approve or reject the post</li> <li>5. If approved, the post is published and made visible to users. If rejected, the post remains unpublished and the admin may provide feedback to the user and only the user can see why the post got denied</li> </ol> |
| Alternative Flows: |   |
| Exceptions:        | <p>Admin encounters technical issues while reviewing the post<br/>System displays an error message and prompts the admin to try again later</p>   |
| Priority:          | High  |
| Frequency of Use:  |   |
| Business Rules:    | BR-22, BR-23, BR-25   |
| Other Information: |   |
| Assumptions:       |   |

### c. Business Rules

| ID    | Business Rule                    | Business Rule Description   |
|-------|----------------------------------|---|
| BR-22 | Admin Authentication Requirement | Admin must be authenticated (logged in) to access post review functionality.  |
| BR-23 | Admin Access Levels              | Different access levels may exist for Admin , allowing varying degrees of control over post review and publication.     |
| BR-25 | Post Review Confirmation         | After reviewing a post, Admin must confirm their decision (approve or reject) for the action to be considered complete. |

## 4.5 Process Restaurant Request

### a. UI Design

Yêu cầu tạo nhà hàng

| ID | Tên | ID người đăng | Xem | Chấp nhận                  | Từ chối                  |
|----|-----|---------------|-----|----------------------------|--------------------------|
| 7  | zxc | 1             | Xem | <button>Chấp nhận</button> | <button>Từ chối</button> |

Previous 1 Next

\*When click “Từ chối”

Yêu cầu tạo nhà hàng

| ID | Tên | ID người đăng | Xem | Chấp nhận                  | Từ chối                  |
|----|-----|---------------|-----|----------------------------|--------------------------|
| 7  | zxc | 1             | Xem | <button>Chấp nhận</button> | <button>Từ chối</button> |

Previous 1 Next

Lý do từ chối

Xác nhận rằng bạn từ chối nhà hàng với lý do sau

Hủy Gửi

### b. Functionalities

|                   |   |
|-------------------|---|
| UC ID and Name:   | Process Restaurant Request  |
| Created By:       | HoangNM   |
| Date Created:     | 28/6/2024   |
| Primary Actor:    | Admin   |
| Secondary Actors: | None  |
| Trigger:          | Admin wants to review and process a submitted restaurant  |
| Description:      | Admin reviews and approves or denies restaurants submitted by users   |
| Preconditions:    | Admin is logged in and has access to the restaurant review function   |
| Postconditions:   | Restaurant is either approved and published or rejected and remains unpublished   |
| Normal Flow:      | <ol style="list-style-type: none"> <li>1. Admin accesses the restaurant request to view the requested restaurants</li> <li>2. Admin selects a restaurant for review</li> <li>3. Admin reads through the restaurant content and evaluates its quality</li> </ol> |

|                    |   |
|--------------------|---|
|                    | <p>4. Admin decides to either approve or reject the restaurant</p> <p>5. If approved, the restaurant is published and made visible to users. If rejected, the restaurant remains unpublished and the admin may provide feedback to the user and only the user can see why the restaurant got denied</p> |
| Alternative Flows: |   |
| Exceptions:        | Admin encounters technical issues while reviewing the post<br>System displays an error message and prompts the admin to try again later   |
| Priority:          | High  |
| Frequency of Use:  |   |
| Business Rules:    | BR-22,BR-23,BR-26   |
| Other Information: |   |
| Assumptions:       |   |

### c. Business Rules

| ID    | Business Rule                    | Business Rule Description   |
|-------|----------------------------------|---|
| BR-22 | Admin Authentication Requirement | Admin must be authenticated (logged in) to access post review functionality.  |
| BR-23 | Admin Access Levels              | There can be different access levels for Admins, allowing for different levels of control over restaurant requests.     |
| BR-26 | Confirm Restaurant Requests      | After reviewing a post, the Admin must confirm his decision (approve or deny) for the action to be considered complete. |

## 4.6 Process Request Edit Post

### a. UI Design

Yêu cầu chỉnh sửa bài viết

| ID bài viết | ID tài khoản | ID nhà hàng | Date             | Xem bài viết chính sửa | Xem bài viết gốc | ...              |
|-------------|--------------|-------------|------------------|------------------------|------------------|------------------|
| 14          | 2            | 7           | 31-07-2024 03:05 | Xem                    | Xem              | Duyệt   Tùy chọn |

Search:

Previous | Next

### b. Functionalities

|                    |  |                   |           |
|--------------------|--|-------------------|-----------|
| UC ID and Name:    | View User  |                   |           |
| Created By:        | HoangNM  | Date Created:     | 28/6/2024 |
| Primary Actor:     | Admin  | Secondary Actors: | None      |
| Trigger:           | Admin wants to review and process the edit request on user post  |                   |           |
| Description:       | Admin reviews and approves or denies post edit submitted by users  |                   |           |
| Preconditions:     | Admin is logged in and has access to the post edit review function   |                   |           |
| Postconditions:    | Post is either approved and published or rejected and remains unpublished  |                   |           |
| Normal Flow:       | 1. Admin accesses the post edit request to view the users requests<br>2. Admin selects a edit of a post for review<br>3. Admin reads through the post content and evaluates its quality<br>4. Admin decides to either approve or reject the post<br>5. If approved, the edit of the post is published, visible and the old post is hidden to users. If rejected, the post remain unchanged |                   |           |
| Alternative Flows: |  |                   |           |
| Exceptions:        | Admin encounters technical issues while reviewing the post<br>System displays an error message and prompts the admin to try again later  |                   |           |
| Priority:          | High   |                   |           |
| Frequency of Use:  |  |                   |           |
| Business Rules:    | BR-22, BR-23, BR-25, BR27  |                   |           |
| Other Information: |  |                   |           |
| Assumptions:       |  |                   |           |

### c. Business Rules

| ID    | Business Rule                    | Business Rule Description   |
|-------|----------------------------------|---|
| BR-22 | Admin Authentication Requirement | Admins must be authenticated (logged in) to process user ban requests.  |
| BR-23 | Admin Access Levels              | Different access levels may exist for Admins, allowing varying degrees of control over banning users..                  |
| BR-25 | Post Review Confirmation         | After reviewing a post, Admin must confirm their decision (approve or reject) for the action to be considered complete. |
| BR-27 | Data Accuracy Confirmation       | The system must ensure the data displayed is accurate and up-to-date.   |

## 4.7 Process Request Edit Restaurant

### a. UI Design

Yêu cầu chỉnh sửa nhà hàng

| ID nhà hàng | ID của người đăng | Tên | Địa chỉ      | Xem địa điểm đã chỉnh sửa | Xem địa điểm ban đầu | ...   |
|-------------|-------------------|-----|--------------|---------------------------|----------------------|---|
| 9           | 2                 | zxc | qw,0,0,0,0,0 | Xem                       | Xem                  | <button>Duyệt</button> <button>Tù chối</button> |

Search:

Previous **1** Next

### b. Functionalities

|                    |   |                   |           |
|--------------------|---|-------------------|-----------|
| UC ID and Name:    | View User   |                   |           |
| Created By:        | HoangNM   | Date Created:     | 28/6/2024 |
| Primary Actor:     | Admin   | Secondary Actors: | None      |
| Trigger:           | Admin wants to review and process the edit request on user restaurant   |                   |           |
| Description:       | Admin reviews and approves or denies restaurant edit submitted by users   |                   |           |
| Preconditions:     | Admin is logged in and has access to the restaurant edit review function  |                   |           |
| Postconditions:    | Restaurant is either approved and published or rejected and remains unpublished   |                   |           |
| Normal Flow:       | 1. Admin accesses the restaurant edit request to view the users requests<br>2. Admin selects a edit of a restaurant for review<br>3. Admin reads through the restaurant content and evaluates its quality<br>4. Admin decides to either approve or reject the restaurant<br>5. If approved, the edit of the restaurant is published, visible and the old restaurant information is hidden to users. If rejected, the restaurant remains unchanged |                   |           |
| Alternative Flows: |   |                   |           |
| Exceptions:        | Admin encounters technical issues while reviewing the post<br>System displays an error message and prompts the admin to try again later   |                   |           |

|                    |                            |
|--------------------|----------------------------|
| Priority:          | High                       |
| Frequency of Use:  |                            |
| Business Rules:    | BR-22, BR-23, BR-26, BR-27 |
| Other Information: |                            |
| Assumptions:       |                            |

### c. Business Rules

| ID    | Business Rule                    | Business Rule Description   |
|-------|----------------------------------|---|
| BR-22 | Admin Authentication Requirement | Admins must be authenticated (logged in) to process user ban requests.  |
| BR-23 | Admin Access Levels              | Different access levels may exist for Admins, allowing varying degrees of control over banning users..                  |
| BR-26 | Confirm Restaurant Requests      | After reviewing a post, the Admin must confirm his decision (approve or deny) for the action to be considered complete. |
| BR-27 | Data Accuracy Confirmation       | The system must ensure the data displayed is accurate and up-to-date.   |

## 4.8 Process Post Report

### a. UI Design

Bài viết bị báo cáo

| ID của báo cáo | ID bài viết | ID tài khoản | Loại báo cáo               | Lý do | Ngày             | Xem bài viết | Search:   |
|----------------|-------------|--------------|----------------------------|-------|------------------|--------------|---|
| 6              | 10          | 1            | Nội dung sai lệch/ Tin giả | asd   | 29-07-2024 23:07 | Xem          | <button>Chấp nhận</button> <button>Từ chối</button> |
| 8              | 11          | 1            | Nội dung sai lệch/ Tin giả | asd   | 29-07-2024 23:11 | Xem          | <button>Chấp nhận</button> <button>Từ chối</button> |

Previous **1** Next

\*When click “Chấp nhận”

| ID của báo cáo | ID bài viết | ID tài khoản | Loại báo cáo               | Lý do                      | Ngày             | Xem bài viết |   |
|----------------|-------------|--------------|----------------------------|----------------------------|------------------|--------------|---|
| 6              | 10          | 1            | Nội dung sai lệch/ Tin giả | asd                        | 29-07-2024 23:07 | Xem          | <button>Chấp nhận</button> <button>Từ chối</button> |
| 8              | 11          | 1            | Nội dung sai lệch/ Tin giả | Nội dung sai lệch/ Tin giả | 29-07-2024 23:07 | Xem          | <button>Chấp nhận</button> <button>Từ chối</button> |

**Duyệt báo cáo**

Nhập lý do án bài viết

Hủy
Gửi

#### b. Functionalities

|                    |  |
|--------------------|--|
| UC ID and Name:    | View User  |
| Created By:        | HoangNM  |
| Date Created:      | 28/6/2024  |
| Primary Actor:     | Admin  |
| Secondary Actors:  | None   |
| Trigger:           | Admin wants to review and process the report on user post  |
| Description:       | Admin reviews and approves or denies post report sent by users   |
| Preconditions:     | Admin is logged in and has access to the post report review function   |
| Postconditions:    | Report is either approved and the post that got reported will be unpublish and only the user who upload that post can see why it got reported or rejected and the post remains published   |
| Normal Flow:       | <ol style="list-style-type: none"> <li>1. Admin accesses the post report to view the post that got reported</li> <li>2. Admin selects a reported post for review</li> <li>3. Admin reads through the post content and evaluates its quality</li> <li>4. Admin decides to either approve or reject the report request</li> <li>5. If approved , the post will be unpublished and the admin can send a message to the user and only the user can see why the post got rejected. If rejected, the post will remain unchanged</li> </ol> |
| Alternative Flows: |  |
| Exceptions:        | Admin encounters technical issues while reviewing the post<br>System displays an error message and prompts the admin to try again later  |
| Priority:          | High   |
| Frequency of Use:  |  |

|                    |                            |
|--------------------|----------------------------|
| Business Rules:    | BR-22, BR-23, BR-26, BR-28 |
| Other Information: |                            |
| Assumptions:       |                            |

### c. Business Rules

| ID    | Business Rule                      | Business Rule Description   |
|-------|------------------------------------|---|
| BR-22 | Admin Authentication Requirement   | Admins must be authenticated (logged in) to process user ban requests.  |
| BR-23 | Admin Access Levels                | Different access levels may exist for Admins, allowing varying degrees of control over banning users..                          |
| BR-27 | Data Accuracy Confirmation         | The system must ensure the data displayed is accurate and up-to-date.   |
| BR-28 | Report Request Review Confirmation | After reviewing a report request, Admin must confirm their decision (approve or deny) for the action to be considered complete. |

## 4.9 Process Restaurant Report

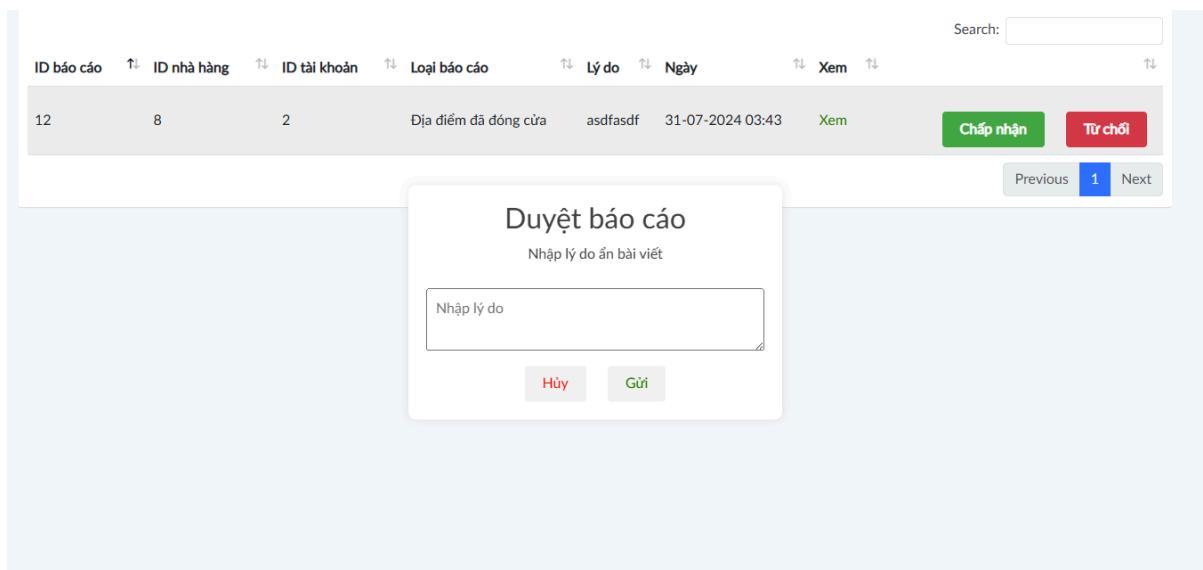
### a. UI Design

Nhà hàng bị báo cáo

| ID báo cáo | ID nhà hàng | ID tài khoản | Loại báo cáo         | Lý do    | Ngày             | Xem | Chấp nhận   | Từ chối   |
|------------|-------------|--------------|----------------------|----------|------------------|-----|---|---|
| 12         | 8           | 2            | Địa điểm đã đóng cửa | asdfasdf | 31-07-2024 03:43 | Xem | <span style="background-color: green; color: white; padding: 2px;">Chấp nhận</span> | <span style="background-color: red; color: white; padding: 2px;">Từ chối</span> |

Previous 1 Next

\*When click “Chấp nhận”



### b. Functionalities

|                    |  |                   |           |
|--------------------|--|-------------------|-----------|
| UC ID and Name:    | View User  |                   |           |
| Created By:        | HoangNM  | Date Created:     | 28/6/2024 |
| Primary Actor:     | Admin  | Secondary Actors: | None      |
| Trigger:           | Admin wants to review and process the report on user restaurant  |                   |           |
| Description:       | Admin reviews and approves or denies restaurant report sent by users   |                   |           |
| Preconditions:     | Admin is logged in and has access to the post report review function   |                   |           |
| Postconditions:    | Report is either approved and the restaurant that got reported will be unpublish and only the user who upload that post can see why it got reported or rejected and the post remains unchanged   |                   |           |
| Normal Flow:       | <ol style="list-style-type: none"> <li>1. Admin accesses the restaurant report to view the restaurant that got reported</li> <li>2. Admin selects a reported restaurant for review</li> <li>3. Admin reads through the restaurant content and evaluates its quality</li> <li>4. Admin decides to either approve or reject the report request</li> <li>5. If approved , the restaurant will be unpublished and the admin can send a message to the user and only the user can see why the post got rejected. If rejected, the restaurant will remain unchanged</li> </ol> |                   |           |
| Alternative Flows: |  |                   |           |
| Exceptions:        | Admin encounters technical issues while reviewing the post   |                   |           |

|                    |   |
|--------------------|---|
|                    | System displays an error message and prompts the admin to try again later |
| Priority:          | High  |
| Frequency of Use:  |   |
| Business Rules:    | BR-22,BR-23,BR-27,BR-28   |
| Other Information: |   |
| Assumptions:       |   |

*c. Business Rules*

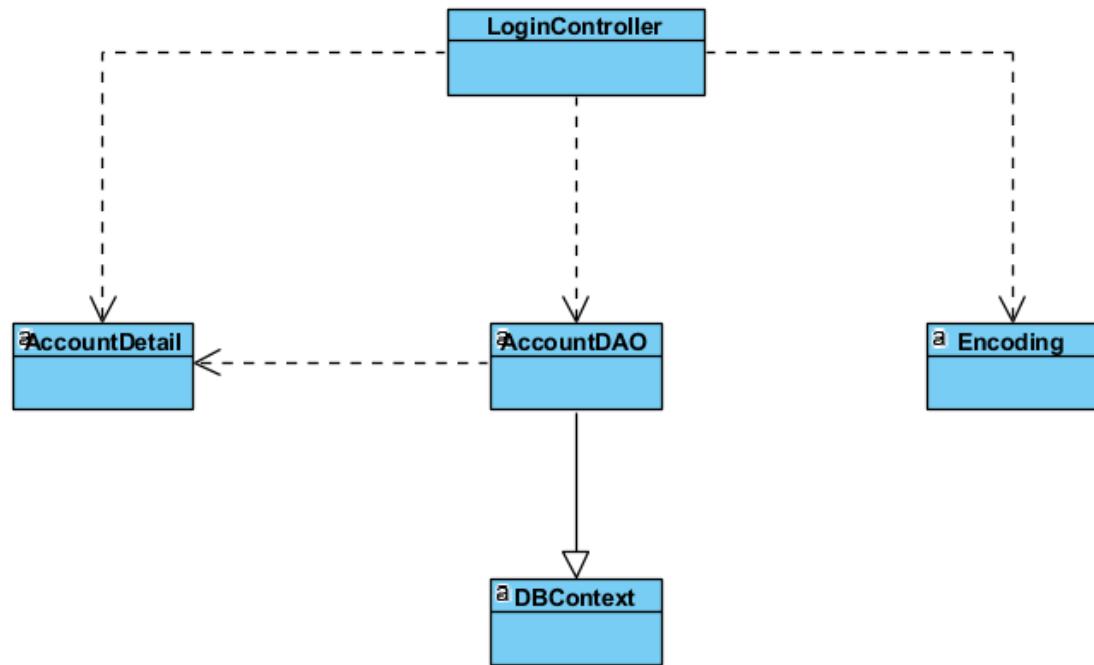
| ID    | Business Rule                      | Business Rule Description   |
|-------|------------------------------------|---|
| BR-22 | Admin Authentication Requirement   | Admins must be authenticated (logged in) to process user ban requests.  |
| BR-23 | Admin Access Levels                | Different access levels may exist for Admins, allowing varying degrees of control over banning users..                          |
| BR-27 | Data Accuracy Confirmation         | The system must ensure the data displayed is accurate and up-to-date.   |
| BR-28 | Report Request Review Confirmation | After reviewing a report request, Admin must confirm their decision (approve or deny) for the action to be considered complete. |

### III. Detailed Designs

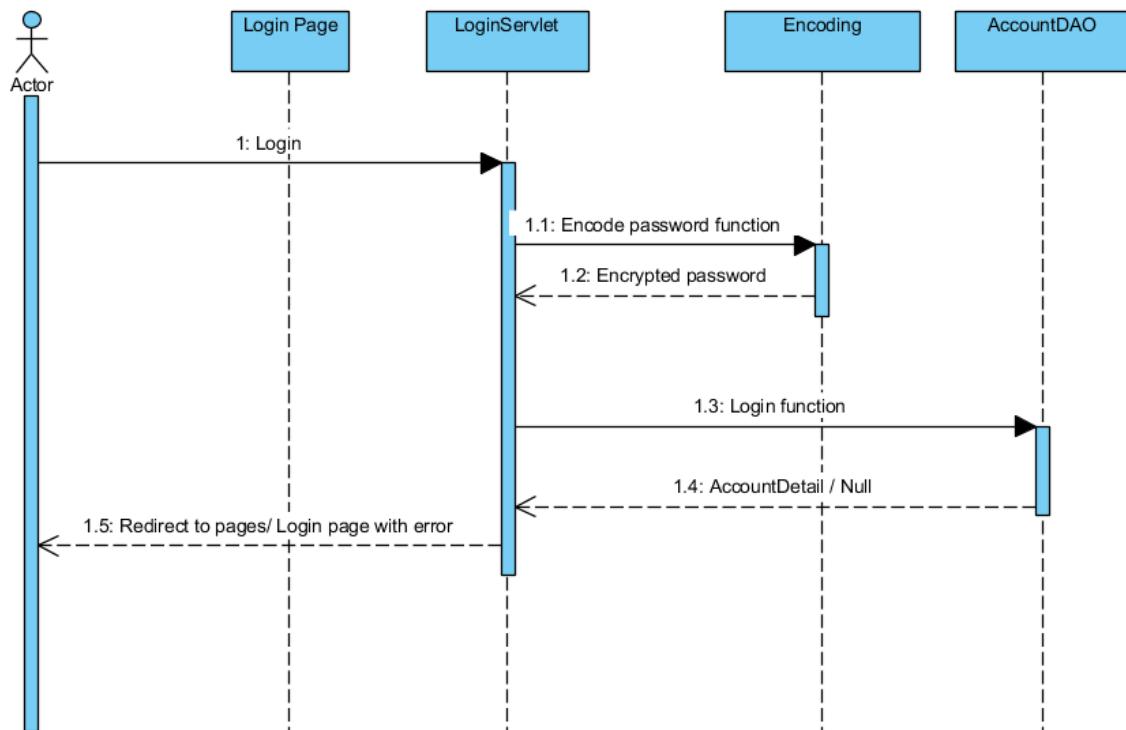
#### 1 Authentications

##### 1.1. Login

###### 1.1.1 Class Diagram



###### 1.1.2 Sequence Diagram(s)

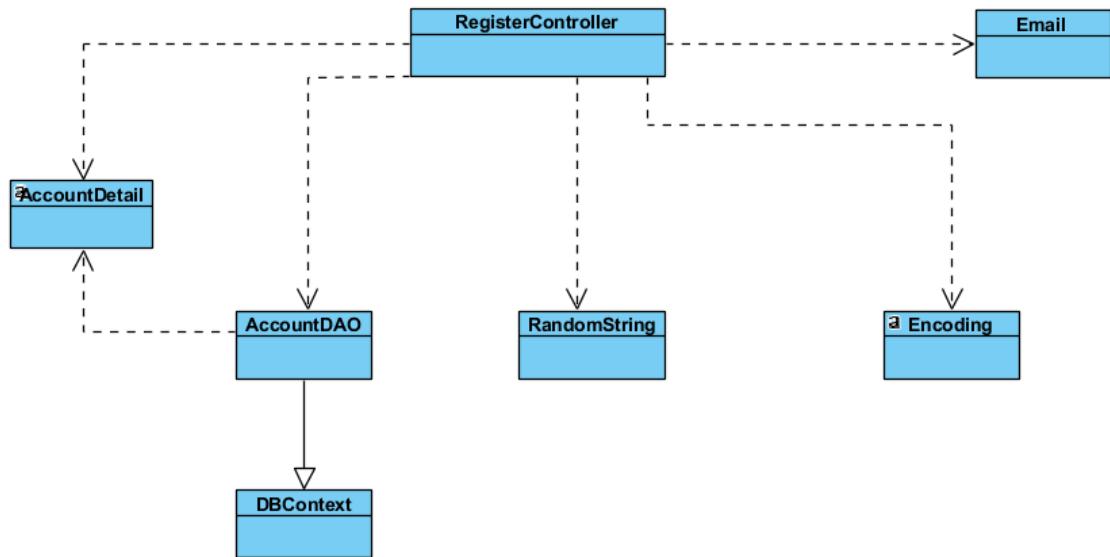


### 1.1.3 Database Queries

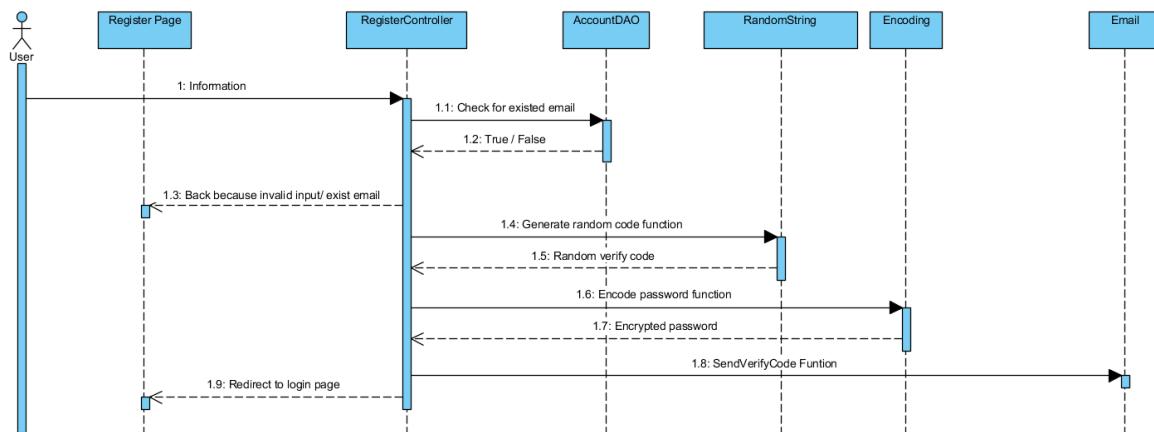
- AccountDetail login(String email, String pass, ServletContext context): "select \* from accountDetail where [email] = ? and [password] = ?"

## 1.2. Register

### 1.2.1 Class Diagram



### 1.2.2 Sequence Diagram

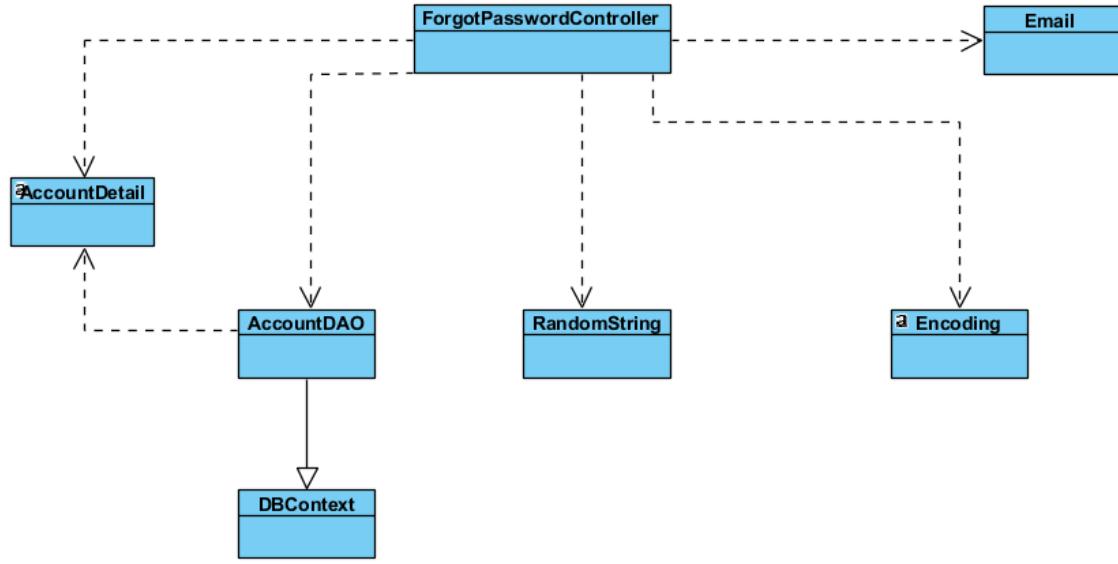


### 1.2.3 Database Queries

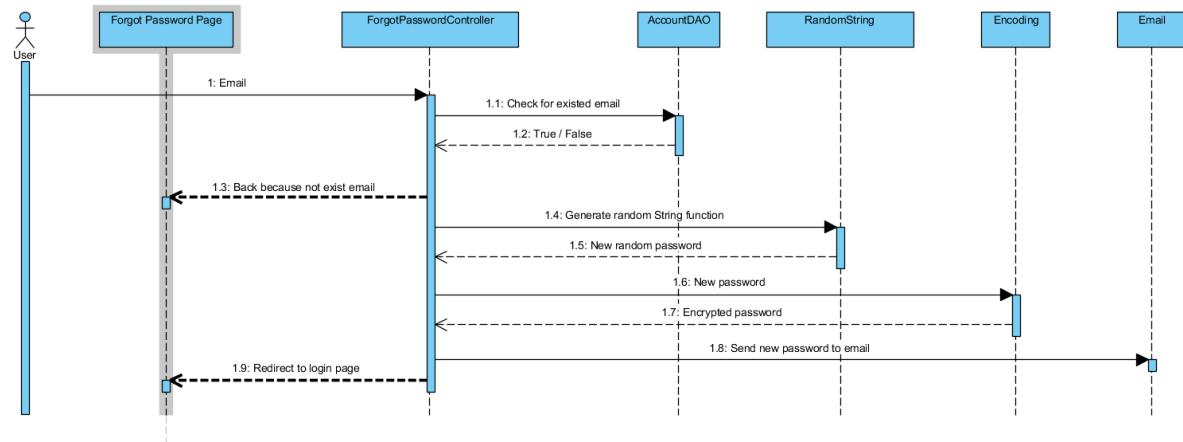
- boolean checkEmailExists(String email): "SELECT 1 FROM accountDetail WHERE email = ?"

### 1.3. Forgot password

#### 1.3.1 Class Diagram



#### 1.3.2 Sequence Diagram

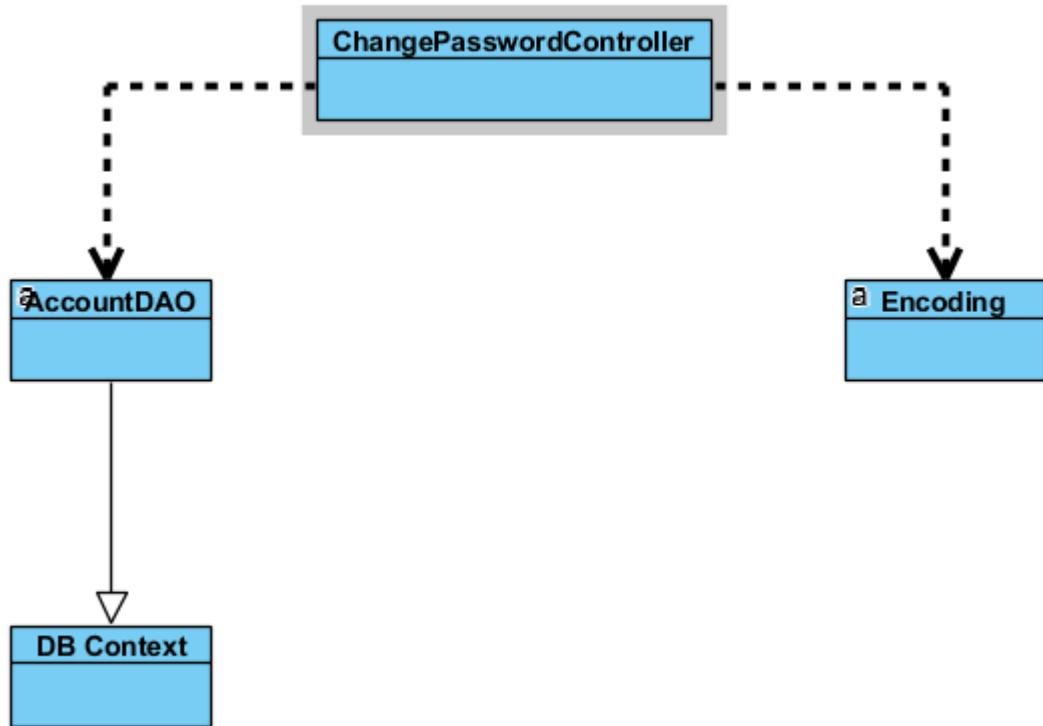


#### 1.3.3 Database Queries

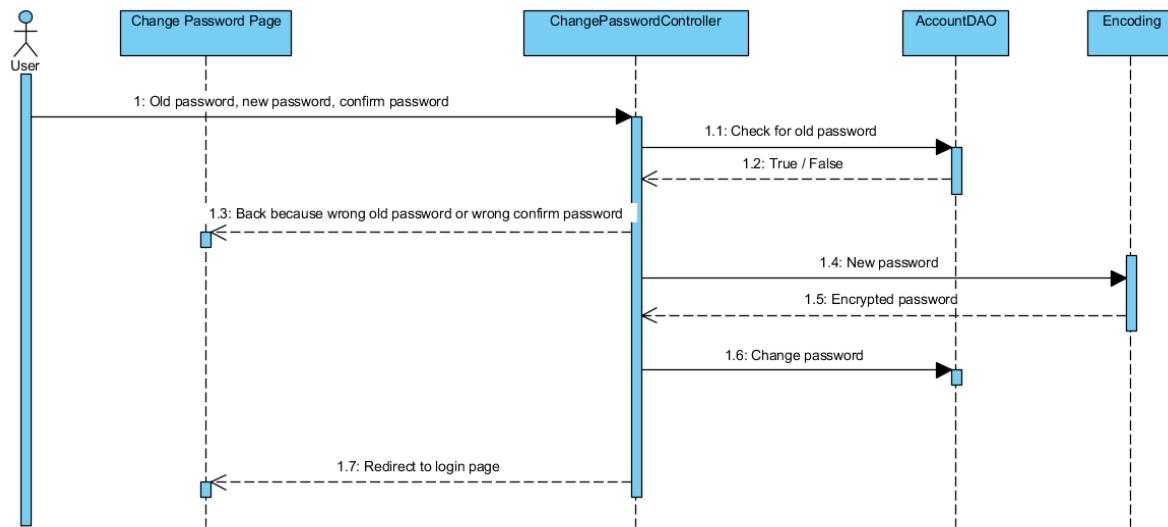
- boolean checkEmailExists(String email): "SELECT 1 FROM accountDetail WHERE email = ?"

## 1.4. Change password

### 1.4.1 Class Diagram



### 1.4.2 Sequence Diagram



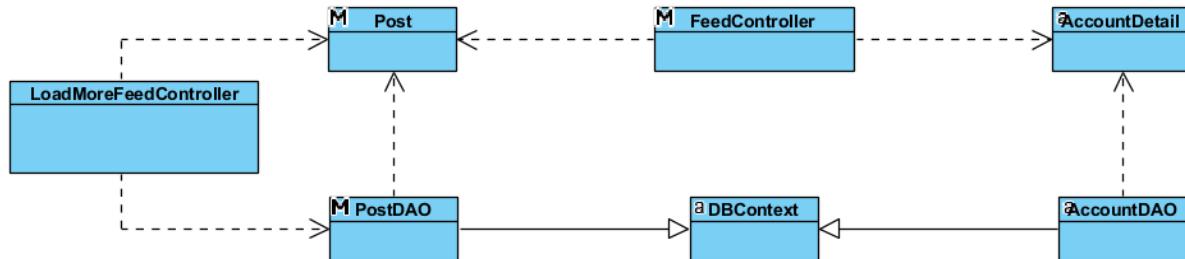
### 1.4.3 Database Queries

- boolean checkCurrentPassword(int account\_id, String encryptedCurrentPassword): "SELECT \* FROM [dbo].[accountDetail] WHERE account\_id = ? AND password = ?"
- void changePassword(String encryptedNewPassword, int account\_id): "UPDATE [dbo].[accountDetail] SET [password] = ? WHERE [account\_id] = ?"

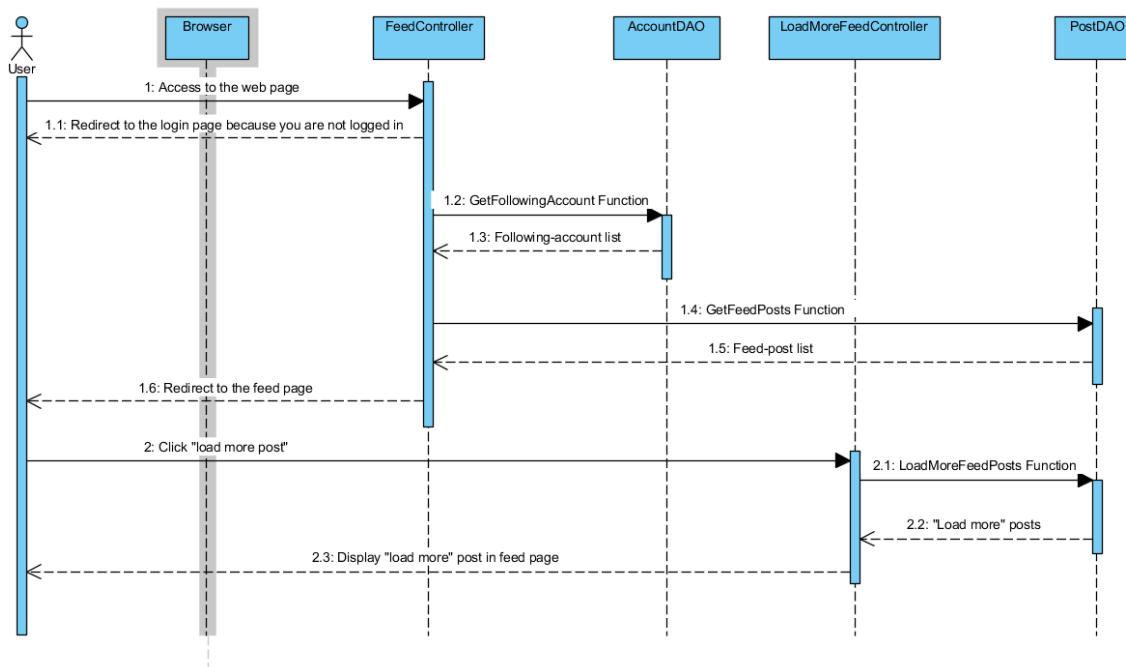
## 2 Common page

### 2.1. Feed page

#### 2.1.1 Class Diagram



#### 2.1.2 Sequence Diagram

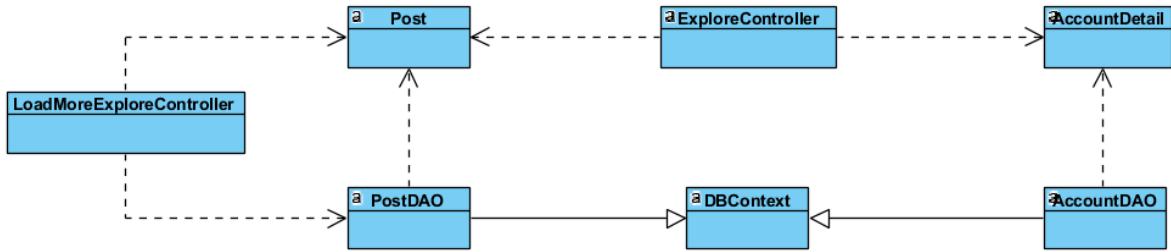


#### 2.1.3 Database Queries

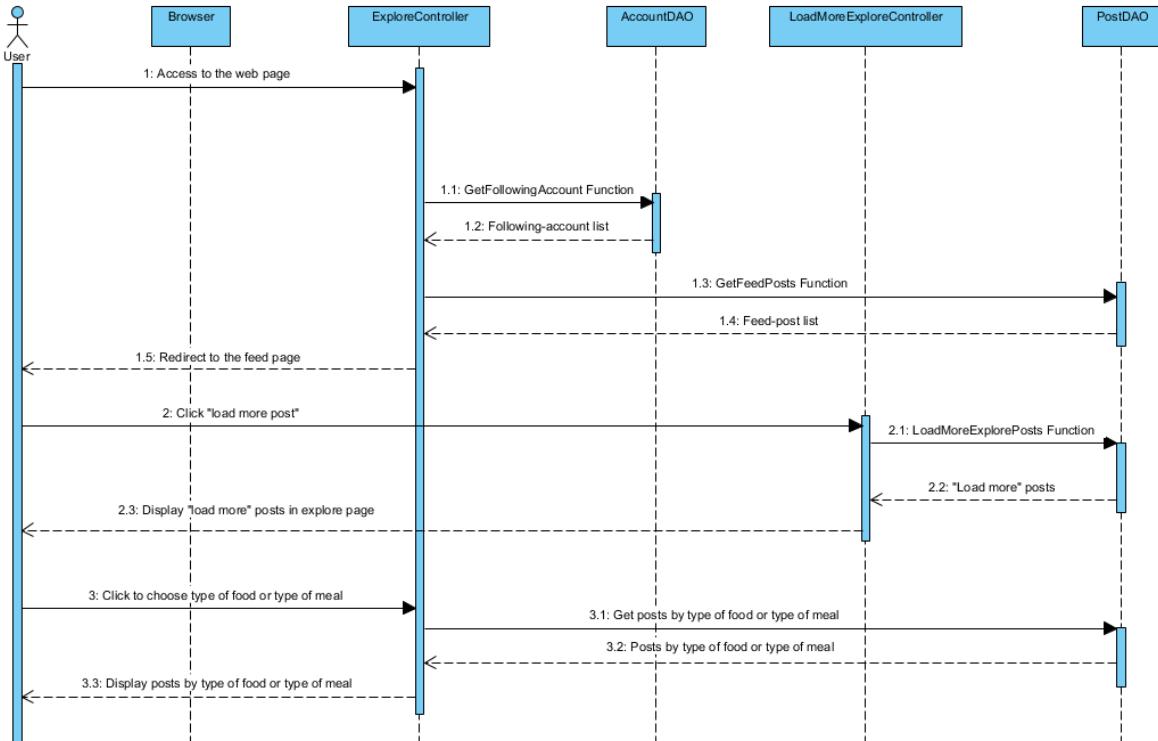
- `List<AccountDetail> getFollowingAccount(int accountId, ServletContext context): "SELECT ad.* FROM accountDetail ad JOIN account_follower af ON ad.account_id = af.account_id WHERE af.follower_id = " + accountId"`
- `List<Post> getFeedPosts(int accountId, ServletContext context): "SELECT top 10 p.* FROM post p WHERE (p.account_id = ? OR p.account_id IN ( SELECT af.account_id FROM account_follower af WHERE af.follower_id = ?)) and p.status = 1 "ORDER BY p.date desc"`
- `List<Post> getNextFeedPosts(int accountId, int amount, ServletContext context): "SELECT p.* FROM post p WHERE (p.account_id = ? OR p.account_id IN ( SELECT af.account_id FROM account_follower af WHERE af.follower_id = ?)) AND p.status = 1 ORDER BY p.date DESC OFFSET ? ROWS FETCH NEXT 10 ROWS ONLY;"`

## 2.2. Explore page

### 2.2.1 Class Diagram



### 2.2.2 Sequence Diagram



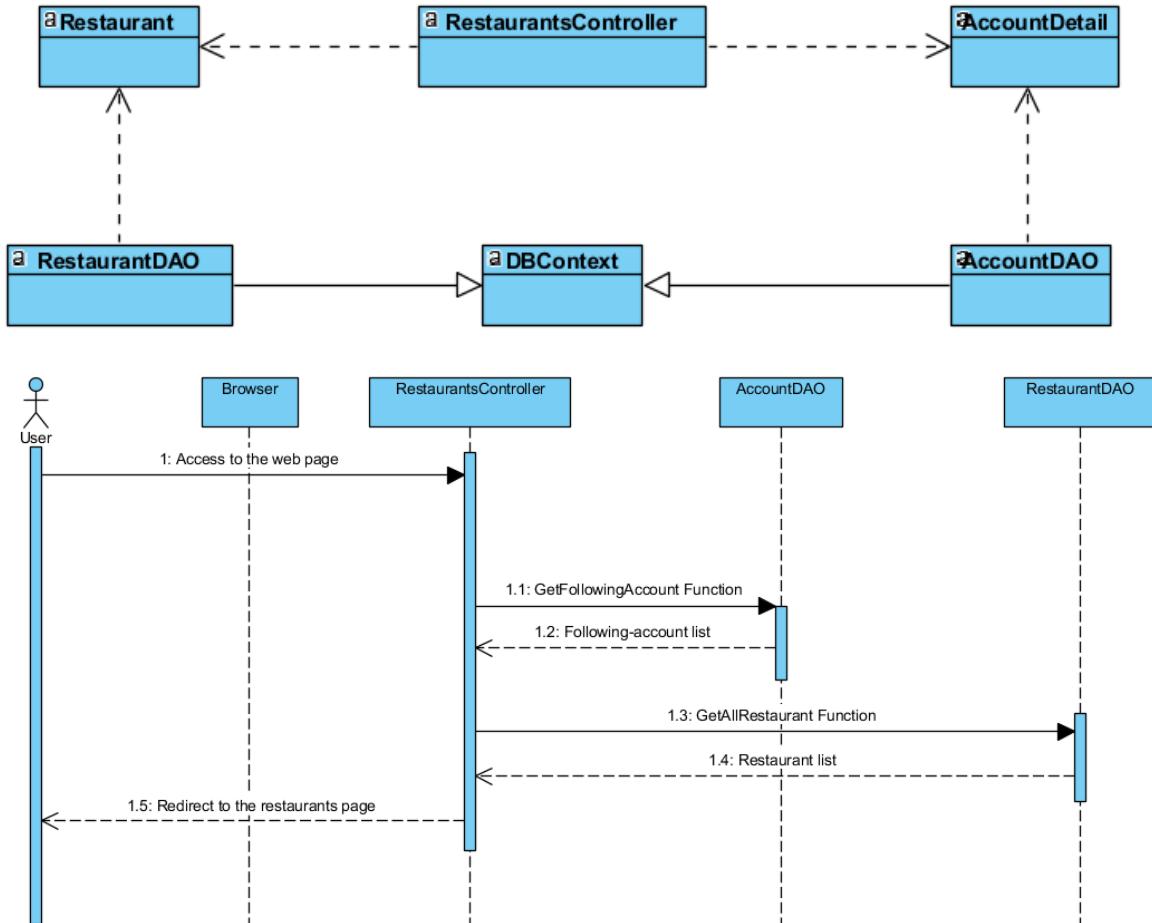
### 2.2.3 Database Queries

- List<AccountDetail> getFollowingAccount(int accountId, ServletContext context): "SELECT ad.\* FROM accountDetail ad JOIN account\_follower af ON ad.account\_id = af.account\_id WHERE af.follower\_id = " + accountId"
- List<Post> getTop9Posts(ServletContext context): "SELECT TOP 9 \* FROM post where status=1 order by date desc"
- List<Post> getNext9Posts(int amount, ServletContext context): "SELECT \* FROM POST WHERE status=1 order by date desc OFFSET " + amount + " ROWS FETCH NEXT 9 ROWS ONLY"
- List<Post> getPostsByTypeOfMeal(ServletContext context, String typeOfMeal): "SELECT p.\* FROM post p JOIN type\_of\_meal tm ON p.post\_id = tm.post\_id WHERE tm.type\_of\_meal\_id = ?"
- List<Post> getPostsByTypeOfFood(ServletContext context, String typeOfFood): "SELECT p.\* FROM post p JOIN type\_of\_food tm ON p.post\_id = tm.post\_id WHERE tm.type\_of\_food\_id = ?"

## 2.3. Restaurant list page

### 2.3.1 Class Diagram

### 2.3.2 Sequence Diagram

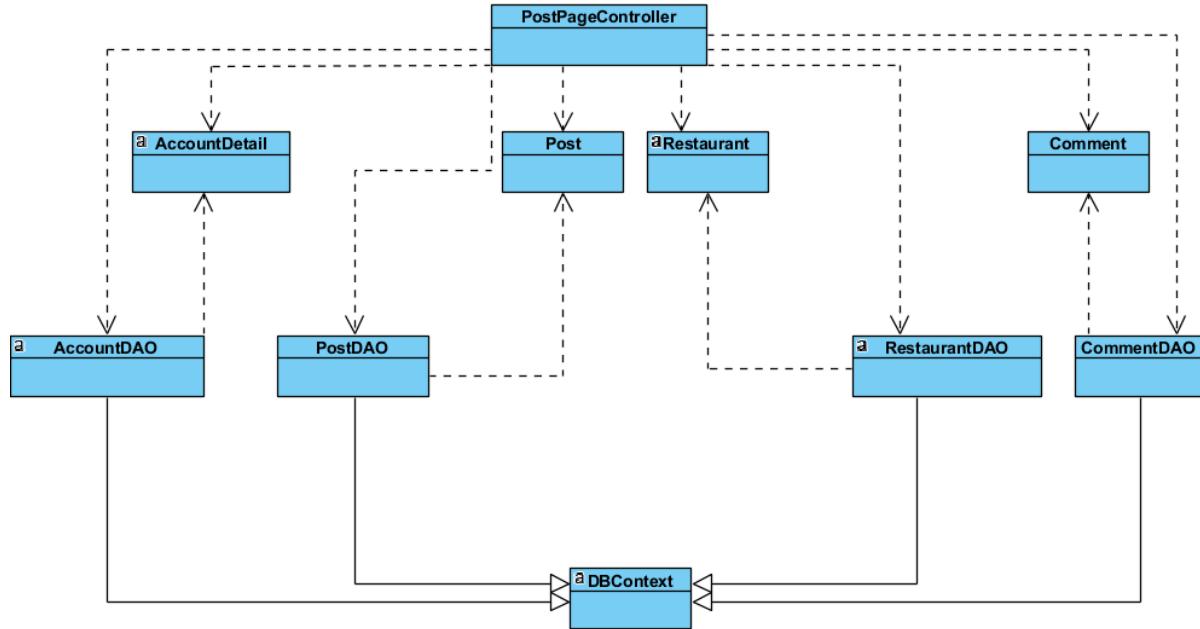


### 2.3.3 Database Queries

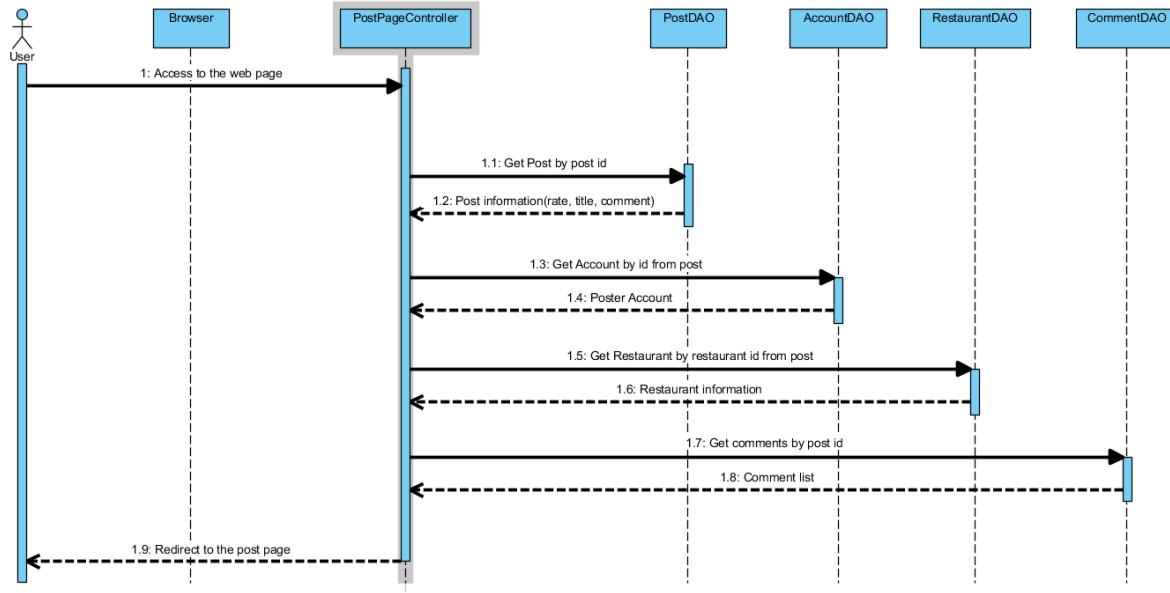
- `List<Restaurant> getAllRestaurant(ServletContext context): "SELECT * FROM restaurant where status=1"`

## 2.4. Post page

### 2.4.1 Class Diagram



### 2.4.2 Sequence Diagram

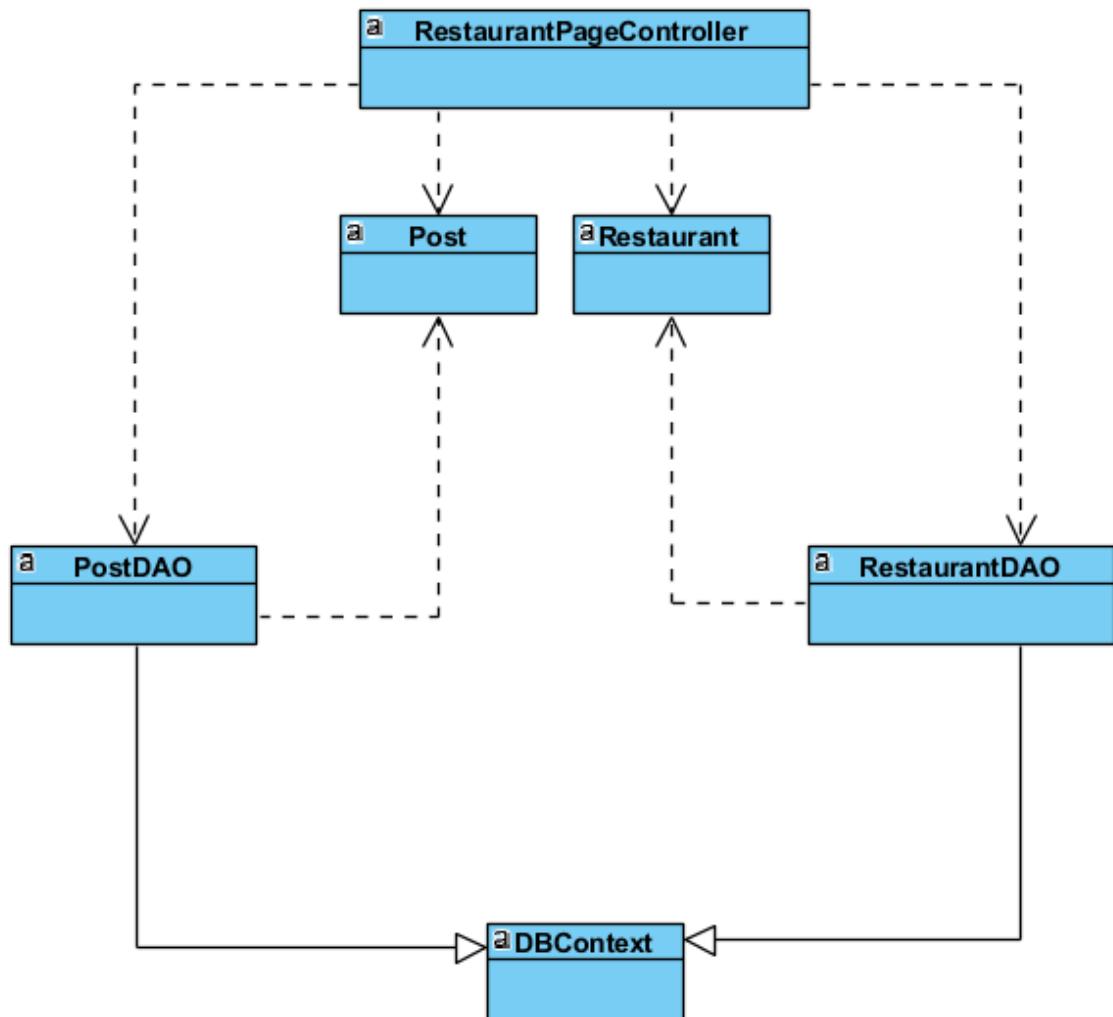


### 2.4.3 Database Queries

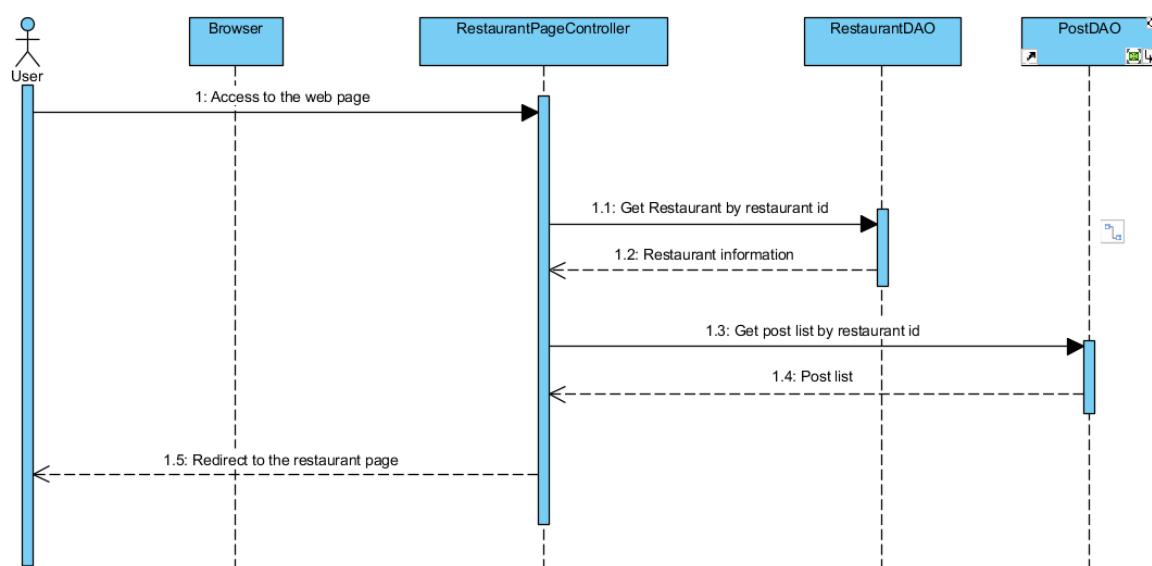
- Post getPostById(String postId, ServletContext context): "SELECT \* FROM post where post\_id = ?"
- AccountDetail getAccountById(int accountId, ServletContext context): "select \* from [dbo].[accountDetail] where [account\_id]=?"
- Restaurant getRestaurantById(int RestaurantId, ServletContext context): "SELECT \* FROM restaurant where restaurant\_id = ?"
- List<Comment> getCommentsByPostID(String postId): "SELECT comment.\* FROM comment JOIN post ON comment.post\_id = post.post\_id WHERE post.post\_id = ?"

## 2.5. Restaurant page

### 2.5.1 Class Diagram



### 2.5.2 Sequence Diagram



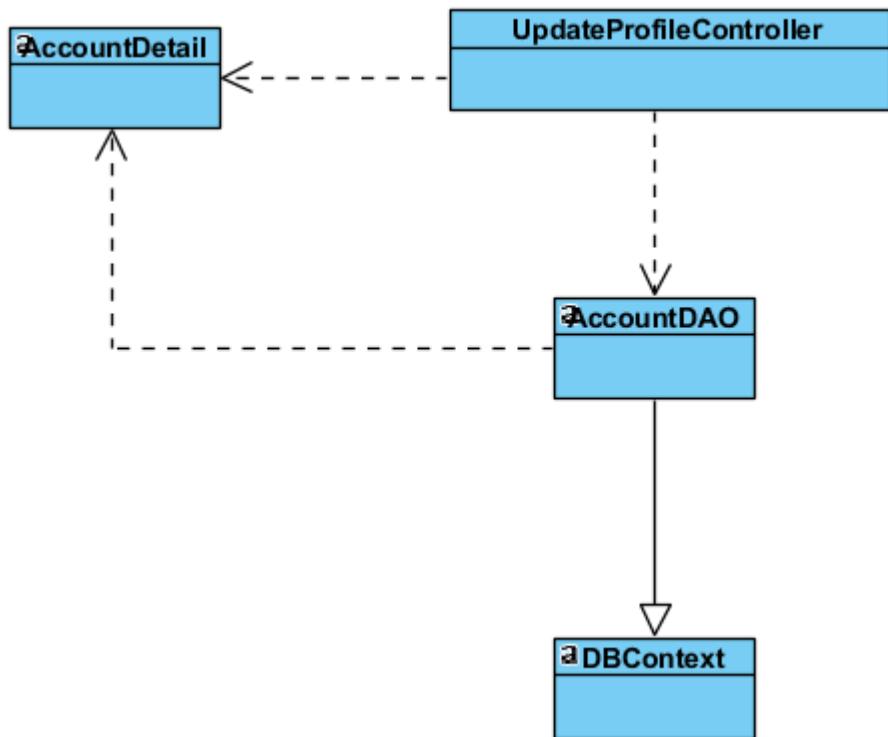
### 2.5.3 Database Queries

- Restaurant getRestaurantById(int RestaurantId, ServletContext context): "SELECT \* FROM restaurant where restaurant\_id = ?"
- List<Post> getPostsByRestaurantId(String restaurantId, ServletContext context): "SELECT \* FROM post where restaurant\_id = ? and post.status=1"

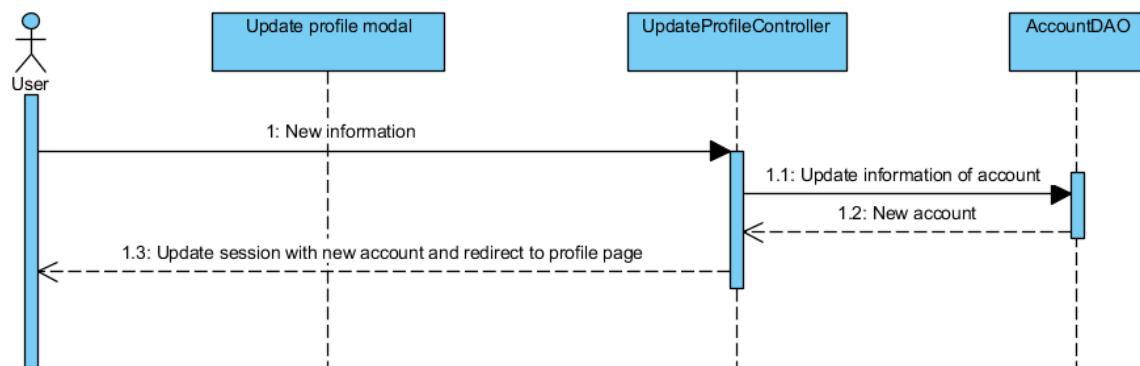
## 3 Function

### 3.1. Update profile information

#### 3.1.1 Class Diagram



#### 3.1.2 Sequence Diagram

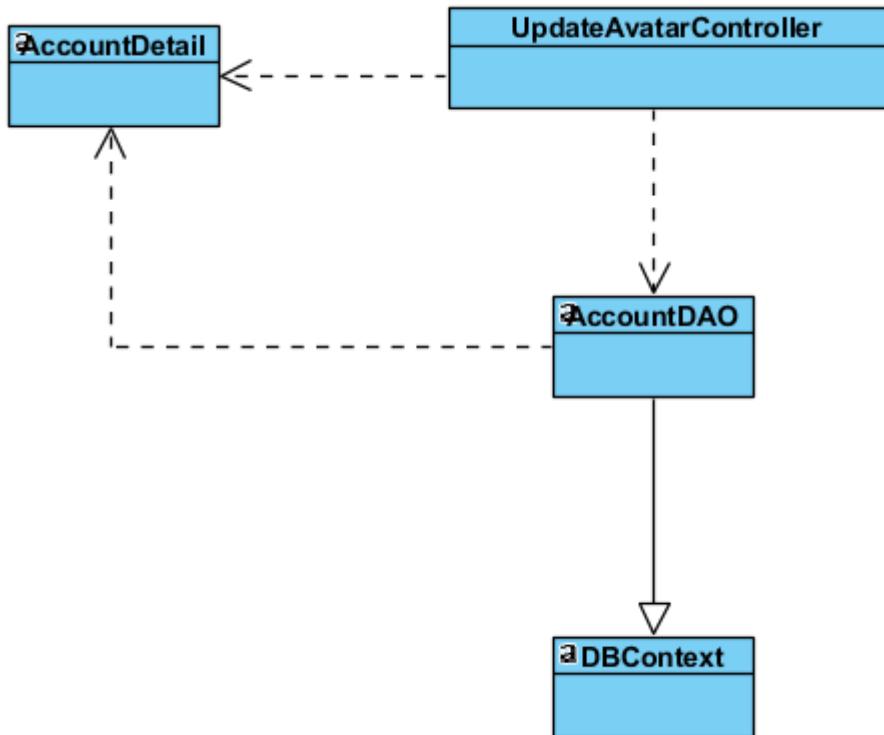


### 3.1.3 Database Queries

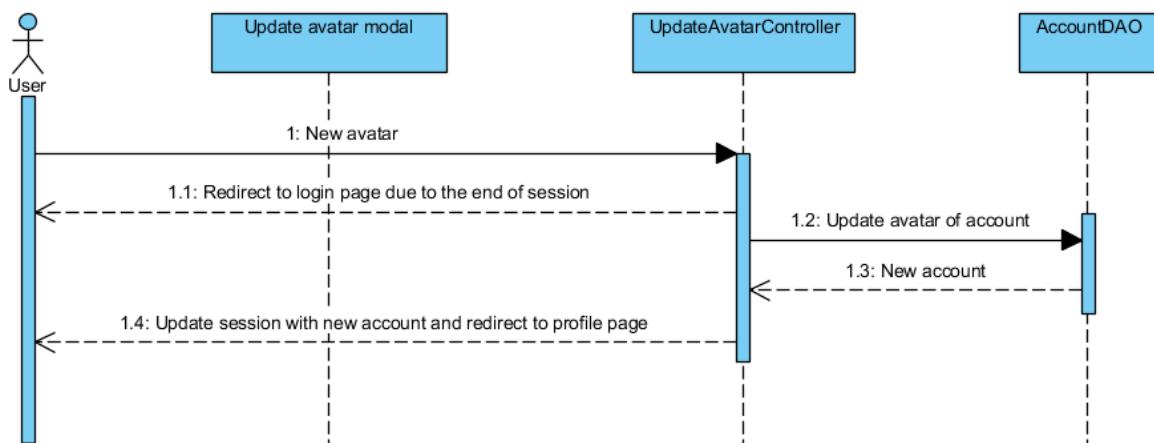
- boolean editProfile(int account\_id, boolean gender, String phone\_number, Date dob, String name, String email): "UPDATE accountDetail SET gender = ?, phone\_number = ?, dob = ?, name = ?, email = ? WHERE account\_id = ?"

## 3.2. Update Avatar

### 3.2.1 Class Diagram



### 3.2.2 Sequence Diagram

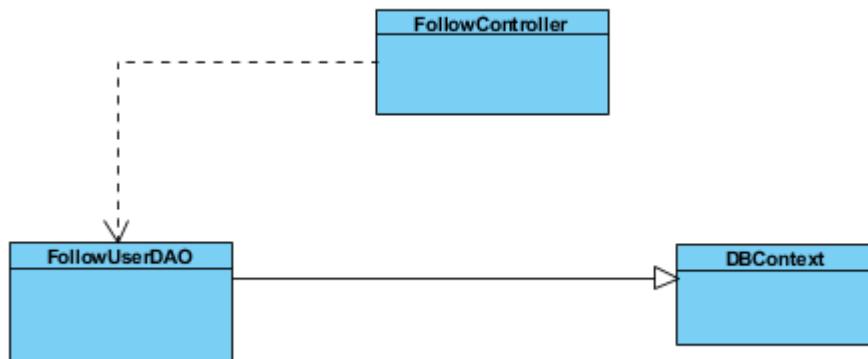


### 3.2.3 Database Queries

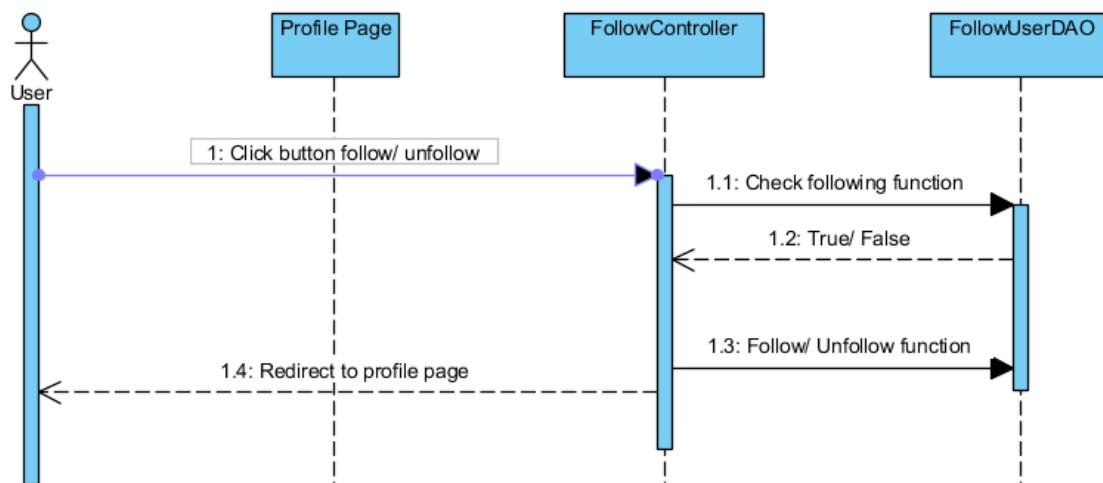
- void updateAvatar(String accountId, String filePath): "UPDATE accountDetail SET avatar = ? WHERE account\_id = ?"

### 3.3. Follow/ Unfollow

#### 3.3.1 Class Diagram



#### 3.3.2 Sequence Diagram

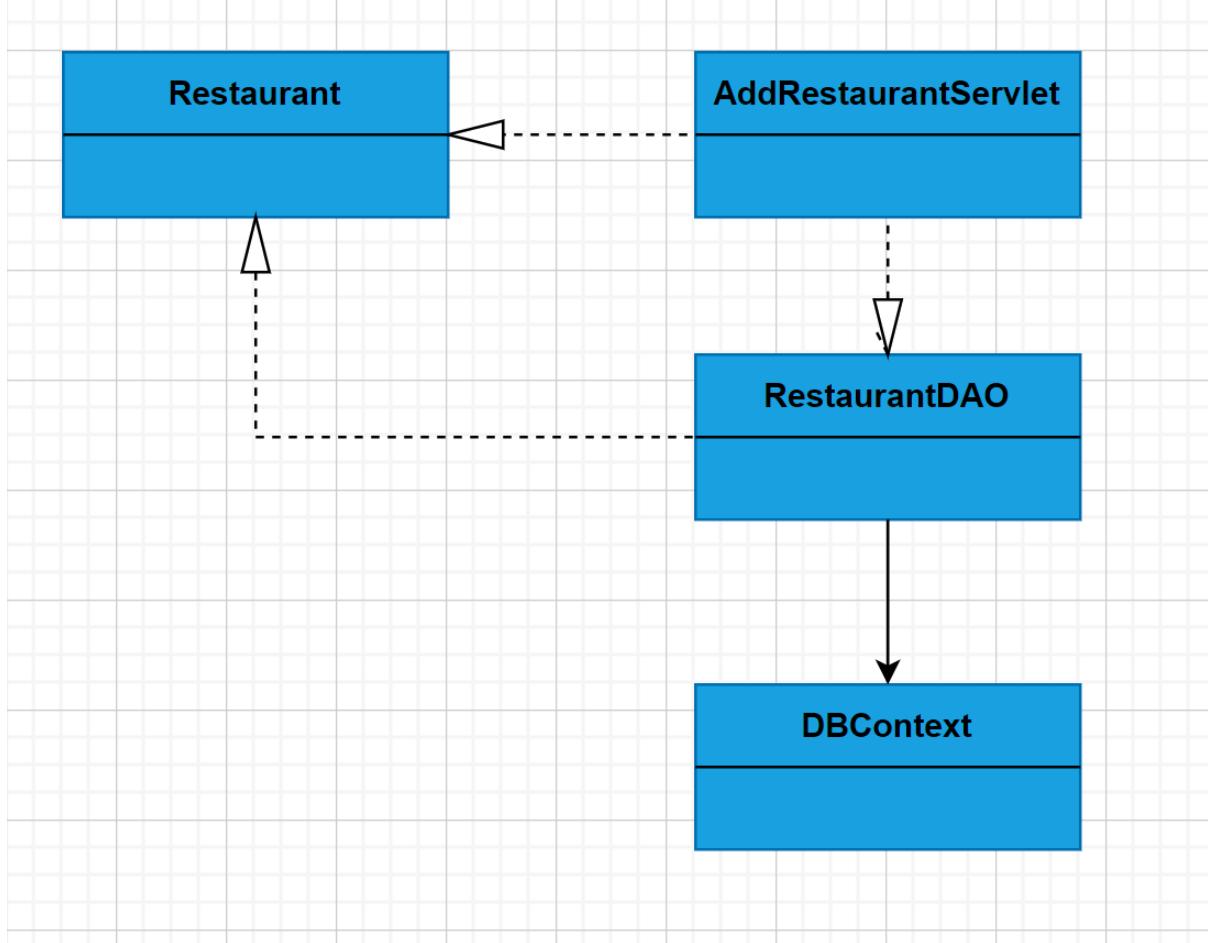


#### 3.3.3 Database Queries

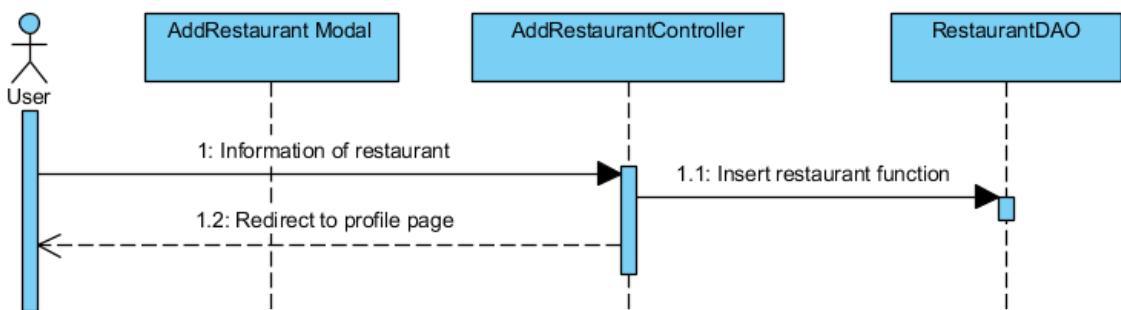
- boolean isFollow(int account\_id, int follower\_id): "SELECT \* FROM [dbo].[account\_follower] WHERE account\_id = ? AND follower\_id = ?"
- boolean insertFollow(int account\_id, int follower\_id): "INSERT INTO [dbo].[account\_follower] VALUES (?,?)"
- boolean deleteFollow(int account\_id, int follower\_id): "DELETE FROM [dbo].[account\_follower] WHERE account\_id = ? AND follower\_id = ?"

### 3.4. Add Restaurant

#### 3.4.1 Class Diagram



#### 3.4.2 Sequence Diagram



#### 3.4.3 Database Queries

```
insertRestaurant(String name, String address, String phoneNumber, String lowestPrice, String highestPrice, Time openTime, Time closeTime, String accountId, String avatarPath):
```

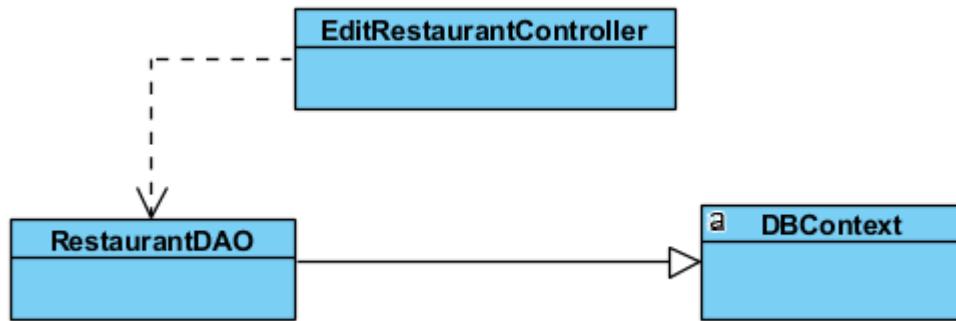
```

“INSERT INTO restaurant (restaurant_id, name, address, phone_number, lowest_price, highest_price,
open_time, close_time, avatar, status, account_id)
VALUES ((SELECT ISNULL(MAX(restaurant_id) + 1, 1) FROM restaurant), ?, ?, ?, ?, ?, ?, ?, ?, 2, ?)”

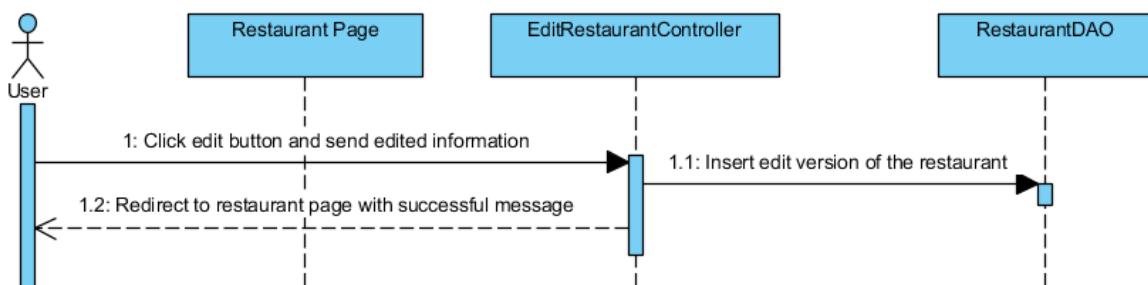
```

### 3.5.Edit Restaurant

#### 3.5.1 Class Diagram



#### 3.5.2 Sequence Diagram



#### 3.5.3 Database Queries

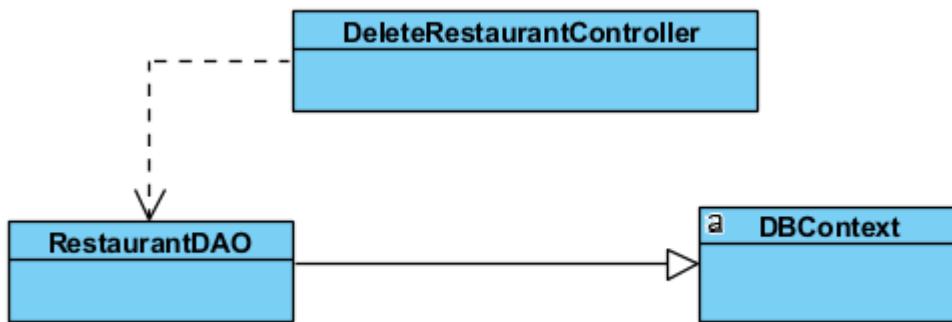
```

insertEditRestaurant(String name, String address, String phoneNumber, String lowestPrice, String
highestPrice, Time openTime, Time closeTime, String accountId, String avatarPath, String edit_of):
“INSERT INTO restaurant (restaurant_id, name, address, phone_number, lowest_price, highest_price,
open_time, close_time, avatar, status, account_id, edit_of)      VALUES ((SELECT
ISNULL(MAX(restaurant_id) + 1, 1) FROM restaurant), ?, ?, ?, ?, ?, ?, ?, ?, 4, ?, ?)”

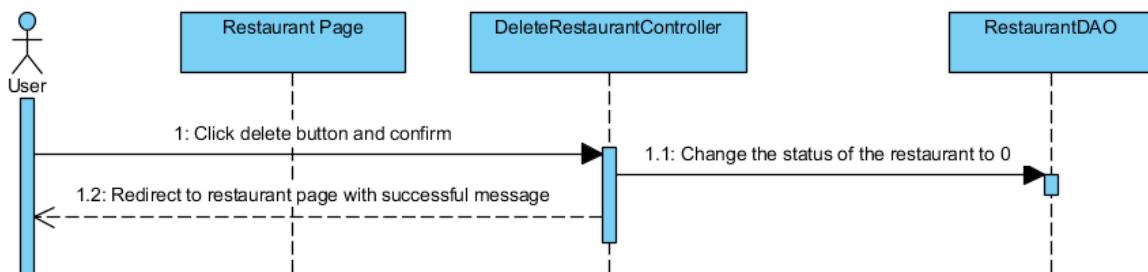
```

### 3.6.Delete Restaurant

#### 3.6.1 Class Diagram



#### 3.6.2 Sequence Diagram

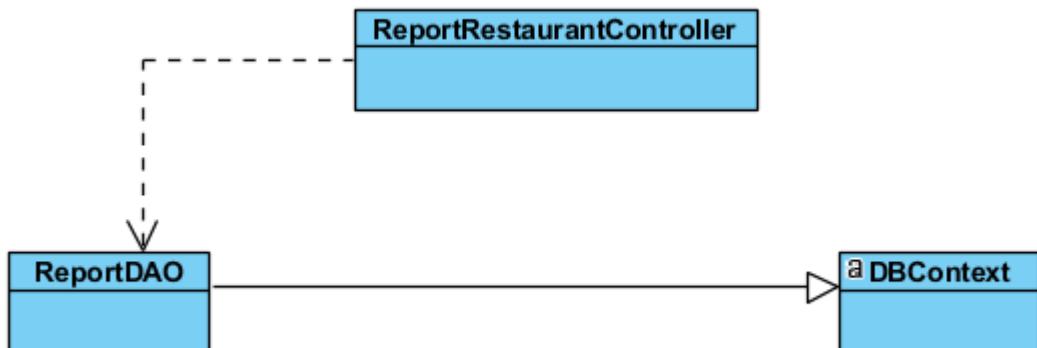


#### 3.6.3 Database Queries

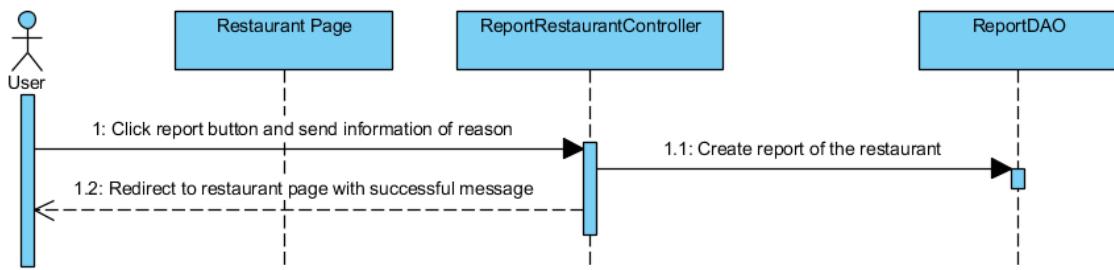
- `void DeleteRestaurant(String restaurantId): "UPDATE restaurant SET status = 0 WHERE restaurant_id = ?;"`

### 3.7.Report Restaurant

#### 3.7.1 Class Diagram



### 3.7.2 Sequence Diagram

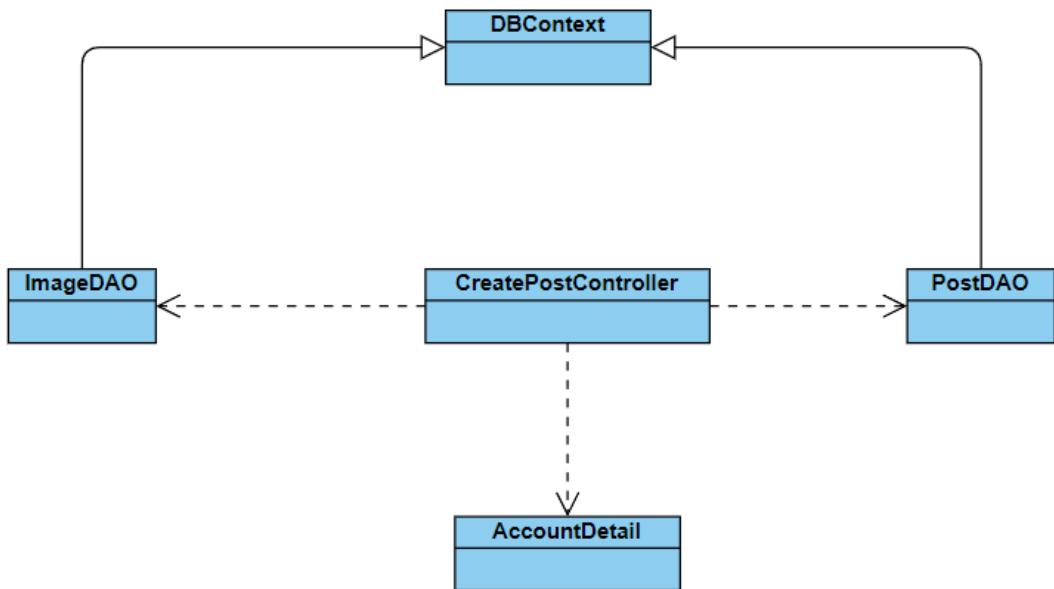


### 3.7.3 Database Queries

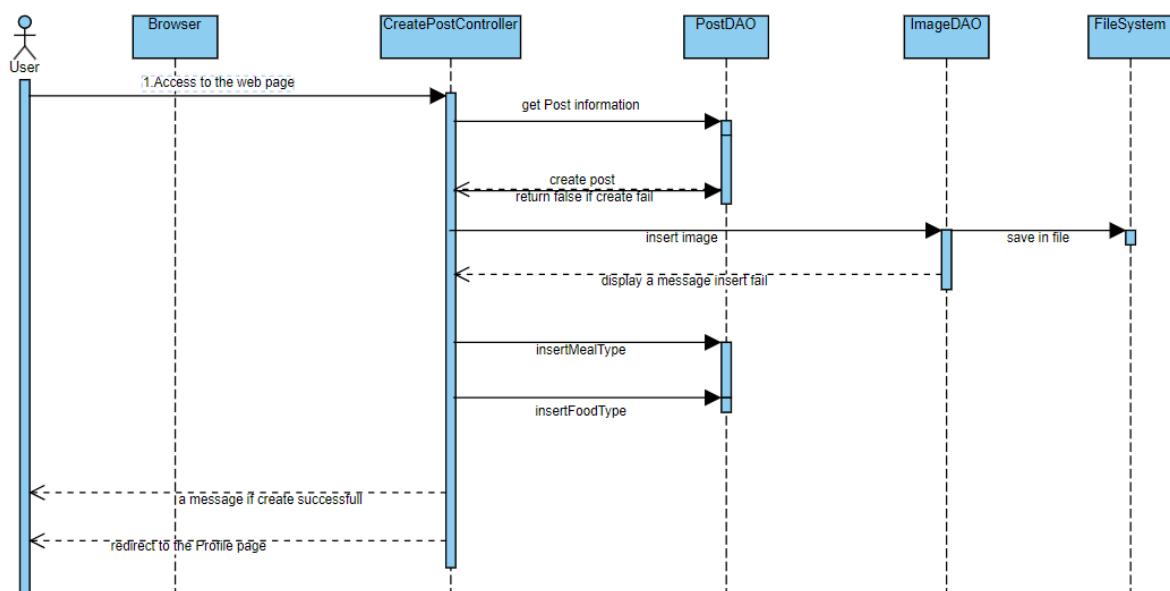
- void insertRestaurantReport(String restaurantId, String accountId, String reportType, String text):  
"INSERT INTO restaurant\_report (restaurant\_report\_id, restaurant\_id, account\_id, type\_of\_report, text, date, status)\n" + "VALUES (" + "(SELECT ISNULL(MAX(restaurant\_report\_id) + 1, 1) FROM restaurant\_report WHERE restaurant\_report\_id > 0), " + "?, " + "?, " + "?, " + "?, " + "CURRENT\_TIMESTAMP, " + "1);"

### 3.8.Create Post

#### 3.8.1 Class Diagram



#### 3.8.2 Sequence Diagram



#### 3.8.3 Database Queries

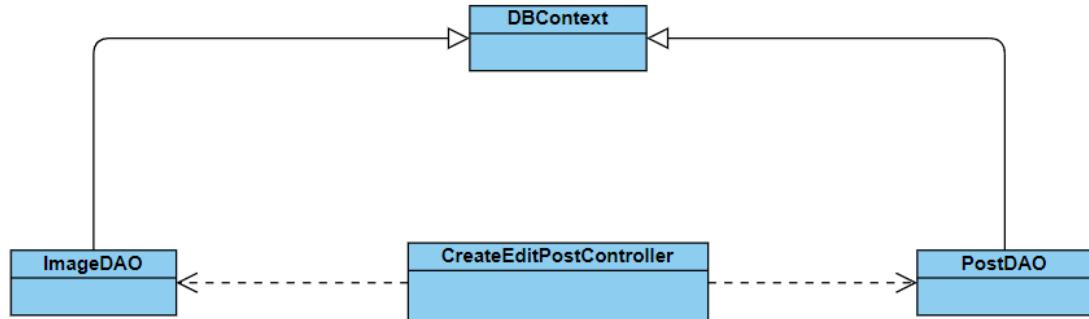
```

-insertPost(...): "INSERT INTO POST (post_id, account_id, restaurant_id, title, date, text, rating, status, service_rate, price_rate, space_rate, taste_rate, hygiene_rate)\n" +
  "VALUES (\n" +
  "  (SELECT ISNULL(MAX(post_id) + 1, 1) FROM post WHERE post_id > 0),\n" +
  "  ?," //1-accountId
  
```

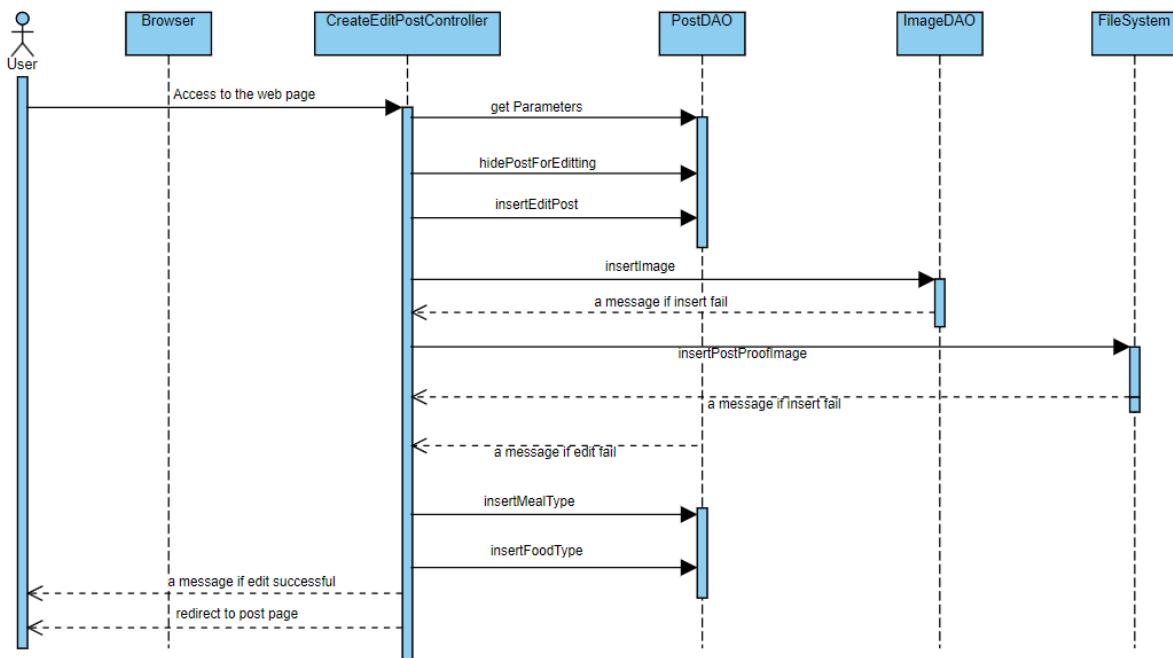


### 3.9.Edit Post

#### 3.9.1 Class Diagram



#### 3.9.2 Sequence Diagram



#### 3.9.3 Database Queries

```

-hidePostForEditting(String postId): "UPDATE post\n"
    + "SET status = 5\n"
    + "WHERE post_id = ?;"

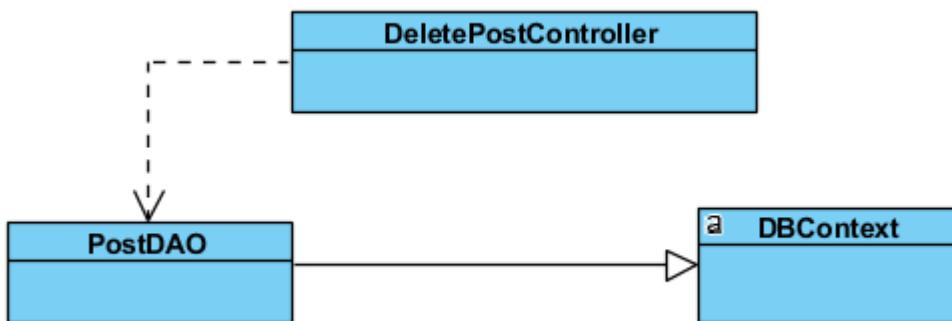
-insertEditPort(...): "INSERT INTO POST (post_id, account_id, restaurant_id, title, date, text, rating,
status, service_rate, price_rate, space_rate, taste_rate, hygiene_rate, edit_of)\n"
    + "VALUES (\n"
    + "    (SELECT ISNULL(MAX(post_id) + 1, 1) FROM post WHERE post_id > 0),\n"

```

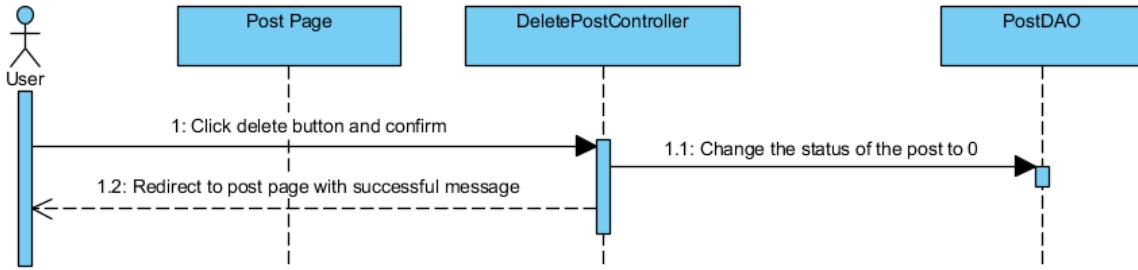
```
+ " ?," //1-accountId
+ " ?," 
+ " ?," 
+ " CURRENT_TIMESTAMP,"
+ " ?," 
+ " ?," 
+ " 4," 
+ " ?," 
+ " ?," 
+ " ?," 
+ " ?," 
+ " ?," 
+ " ?," 
+ " ?," 
+ " ?," 
+ " ?," 
+ " ?," 
+ " ?," 
+ " ?," 
+ " ?);"
-insertImage(...): "INSERT INTO media (media_id, post_id, restaurant_id, media_data) VALUES (
+ "(SELECT ISNULL(MAX(media_id) + 1, 1) FROM media WHERE media_id > 0)," 
+ " ?, ?, ?)"
-insertPostProofImage(...): "INSERT INTO post_proof_media (media_id, post_id, restaurant_id,
media_data) VALUES (
+ "(SELECT ISNULL(MAX(media_id) + 1, 1) FROM post_proof_media WHERE media_id > 0)," 
+ " ?, ?, ?)"
-insertMealType(String mealType, String postId): "INSERT INTO type_of_meal (type_of_meal_id,
post_id) VALUES(
+ "?," 
+ "?)"
-insertFoodType(String foodType, String postId): "INSERT INTO type_of_food (type_of_food_id,
post_id) VALUES(
+ "?," 
+ "?)"
```

### 3.10.Delete Post

### *3.10.1 Class Diagram*



### 3.10.2 Sequence Diagram



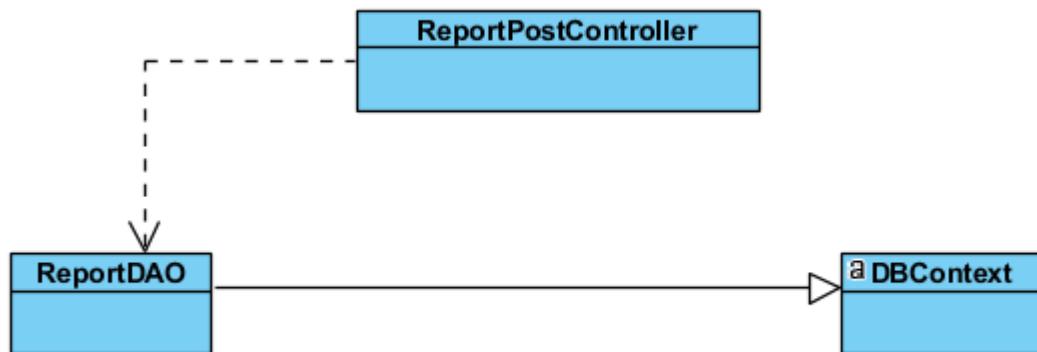
### 3.10.3 Database Queries

```

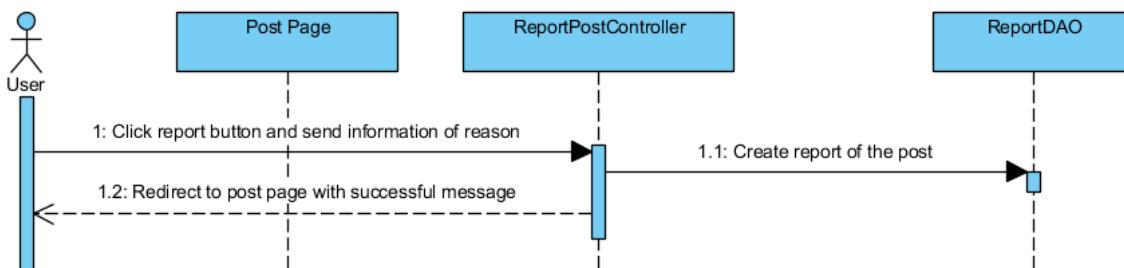
-void DeletePost(String postId): "UPDATE post\n"
+ "SET status = 0\n"
+ "WHERE post_id = ?;"
```

## 3.11. Report Post

### 3.11.1 Class Diagram



### 3.11.2 Sequence Diagram



### 3.11.3 Database Queries

```

-void insertPostReport(String postId, String accountId, String reportType, String text): "INSERT INTO
post_report (post_report_id, post_id, account_id, type_of_report, text, date, status)\n"
+ "VALUES ("
```

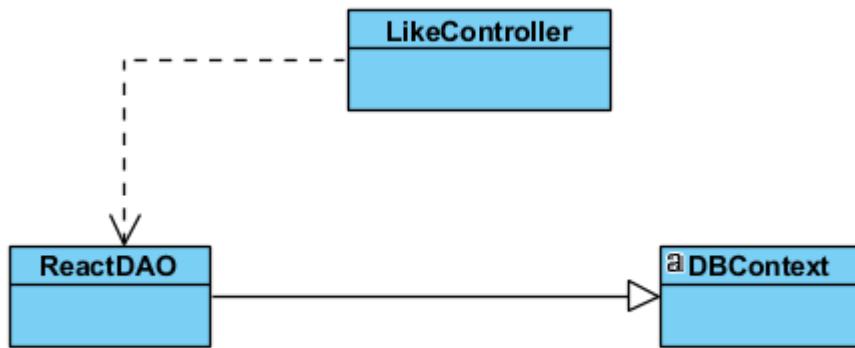
```

+ "(SELECT ISNULL(MAX(post_report_id) + 1, 1) FROM post_report WHERE post_report_id >
0,"
+ "?,"
+ "?,"
+ "?,"
+ "?,"
+ "CURRENT_TIMESTAMP,"
+ "1);"

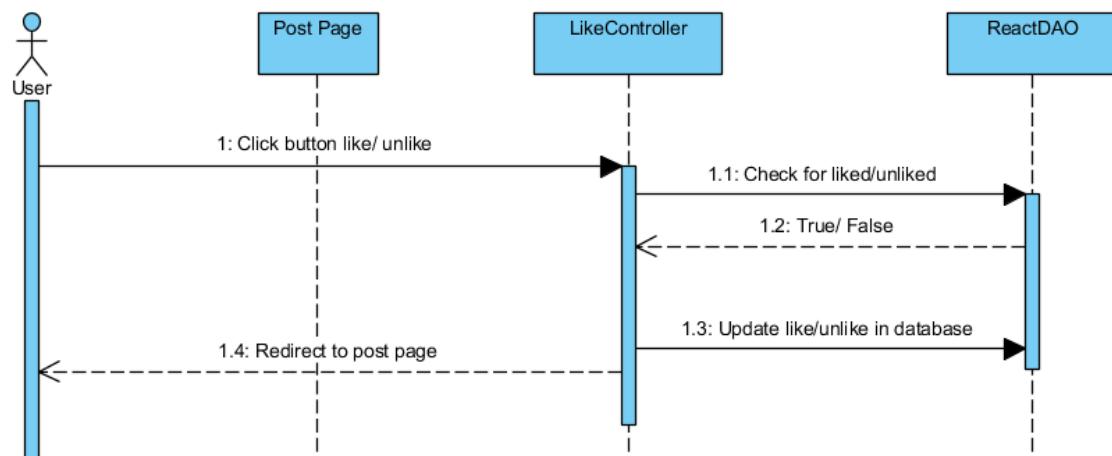
```

### 3.12. Like/Unlike Post

#### 3.12.1 Class Diagram



#### 3.12.2 Sequence Diagram

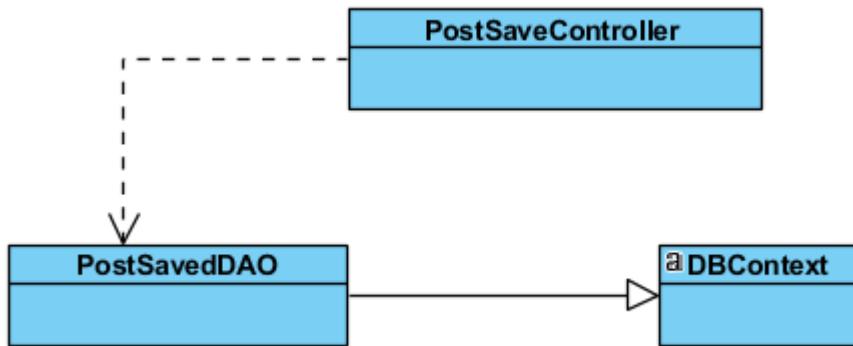


#### 3.12.3 Database Queries

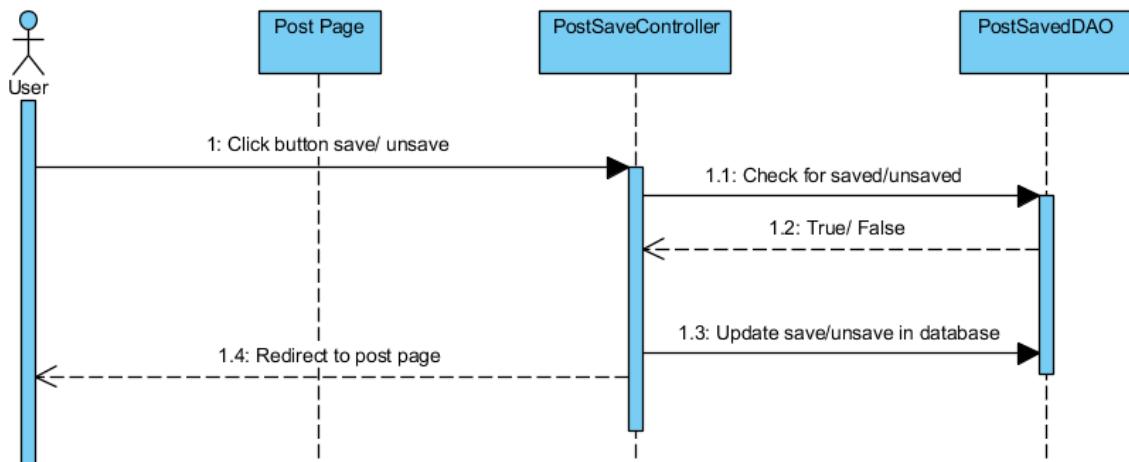
- boolean isLikeByUser(int post\_id, int account\_id): "SELECT \* FROM react WHERE post\_id = ? AND account\_id = ?"
- void insertLike(int post\_id, int account\_id): "INSERT INTO [dbo].[react]\n + \" VALUES (?,?)"
- void deleteLike(int post\_id, int account\_id): "DELETE FROM react WHERE post\_id = ? AND account\_id = ?"

### 3.13. Save/Unsave Post

#### 3.13.1 Class Diagram



#### 3.13.2 Sequence Diagram

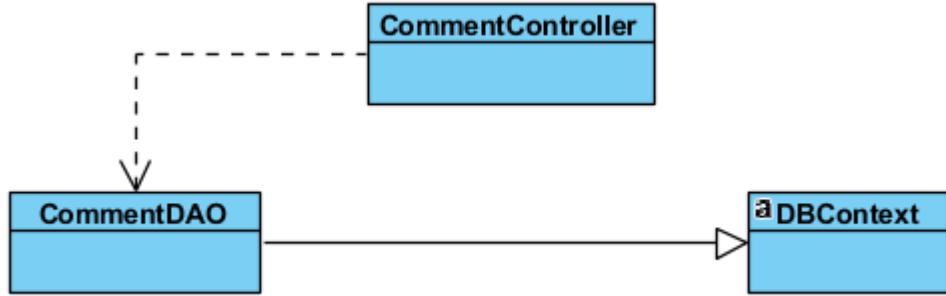


#### 3.13.3 Database Queries

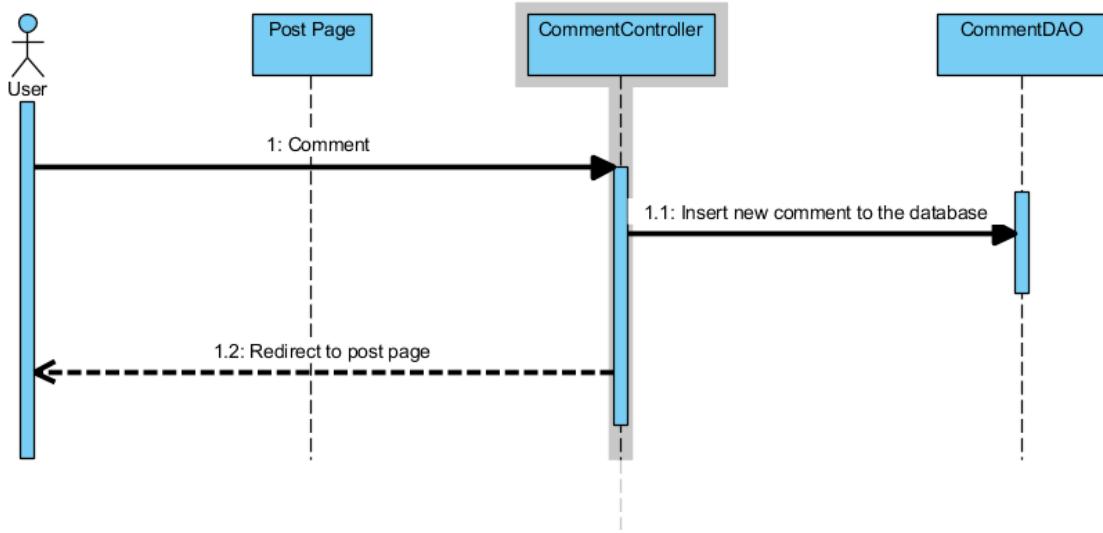
- boolean isSavedByUser(int post\_id, int account\_id): "SELECT \* FROM post\_saved WHERE post\_id = ? AND account\_id = ?"
- void insertSave(int postId, int accountId): "INSERT INTO [dbo].[post\_saved]\n + \" VALUES (?,?)"
- void deleteSaved(int post\_id, int account\_id): "DELETE FROM post\_saved WHERE post\_id = ? AND account\_id = ?"

### 3.14. Comment

#### 3.14.1 Class Diagram



#### 3.14.2 Sequence Diagram

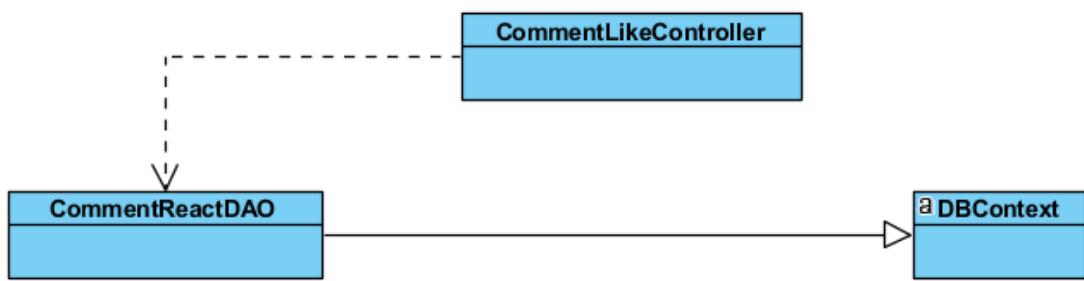


#### 3.14.3 Database Queries

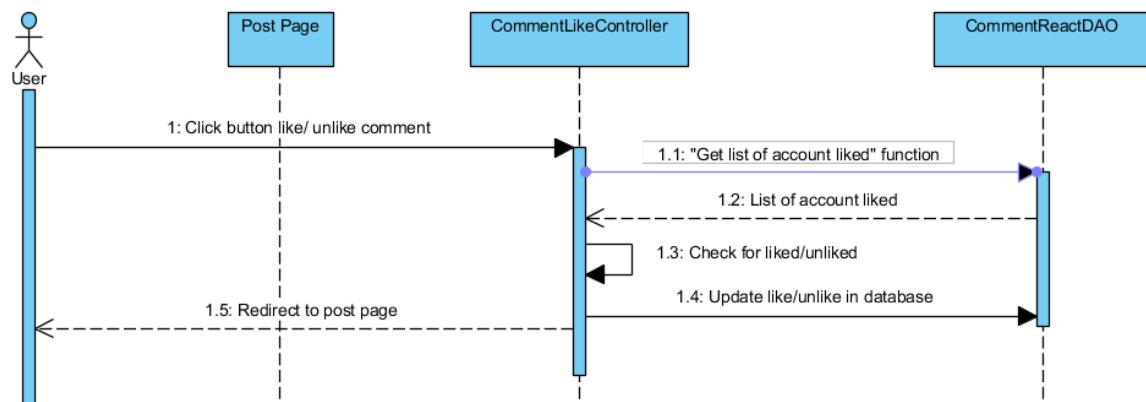
- `insertComment(String accountId, String postId, String text): "INSERT INTO comment (comment_id, post_id, account_id, text, date, status)\nVALUES (\n(SELECT ISNULL(MAX(comment_id) + 1, 1) FROM comment WHERE comment_id > 0),\n?," //post_id\n?," //account_id\n?," //text\nCURRENT_TIMESTAMP,\n1)"`

### 3.15. Like/ Unlike comment

#### 3.14.1 Class Diagram



### 3.14.2 Sequence Diagram

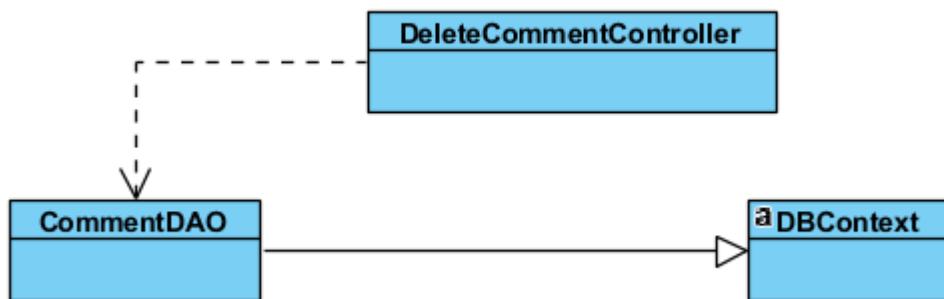


### 3.14.3 Database Queries

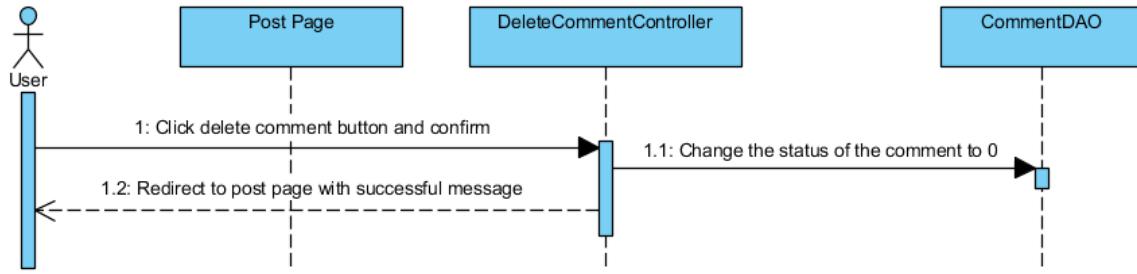
- `List<Integer> getReactAccounts(int commentId): "Select * from comment_react where comment_id = ?"`
- `void insertCommentReact(String commentId, String accountId): "INSERT INTO comment_react (comment_id, account_id) VALUES (?, ?);"`
- `void deleteCommentReact(String commentId, String accountId): "DELETE FROM comment_react WHERE comment_id = ? AND account_id = ?;"`

## 3.16. Delete comment

### 3.16.1 Class Diagram



### 3.16.2 Sequence Diagram



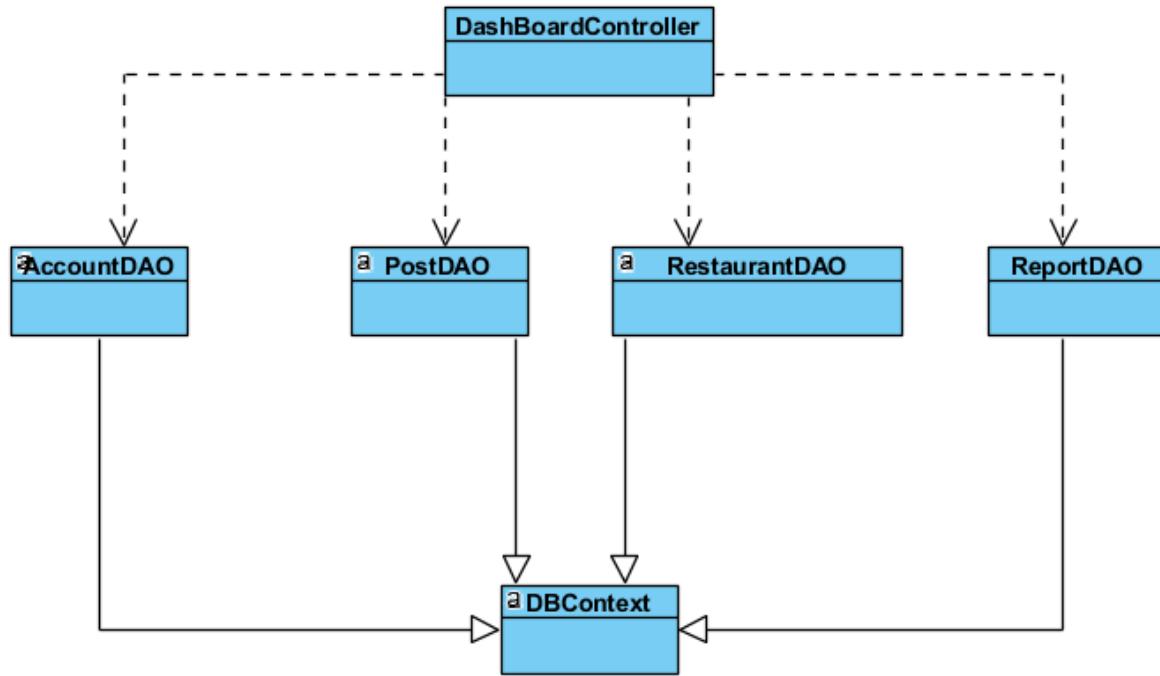
### 3.16.3 Database Queries

- void hideComment(String commentId): "UPDATE comment SET status = 0 WHERE comment\_id = ?;"

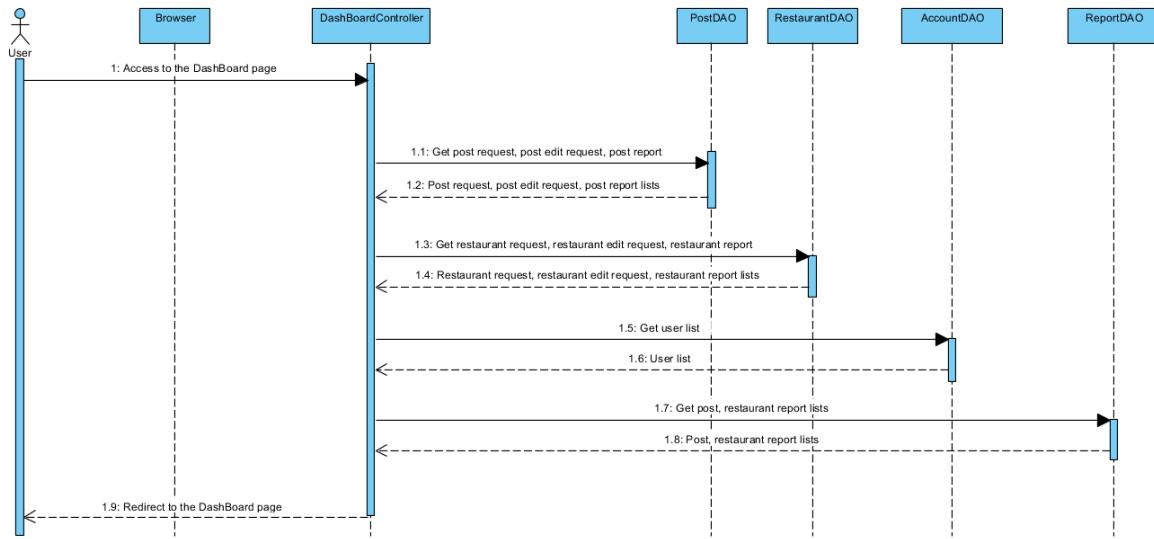
## 4. Admin Function

### 4.1. Dashboard page

#### 4.1.1 Class Diagram



#### 4.1.2 Sequence Diagram



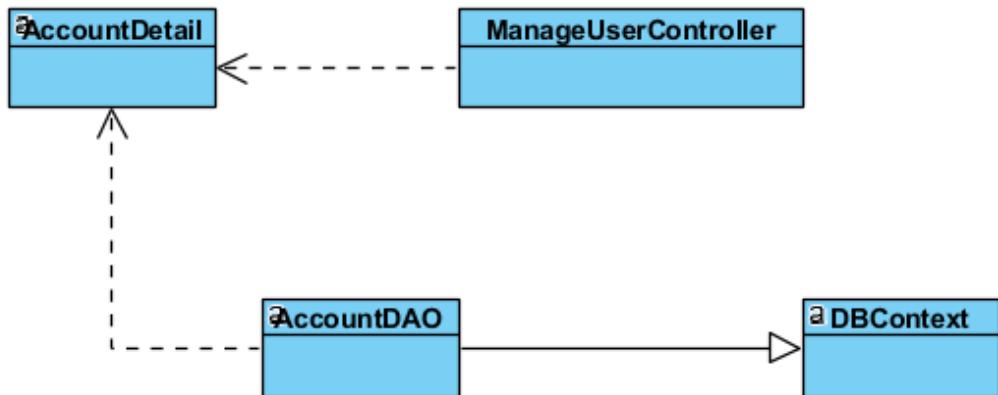
#### 4.1.3 Database Queries

```

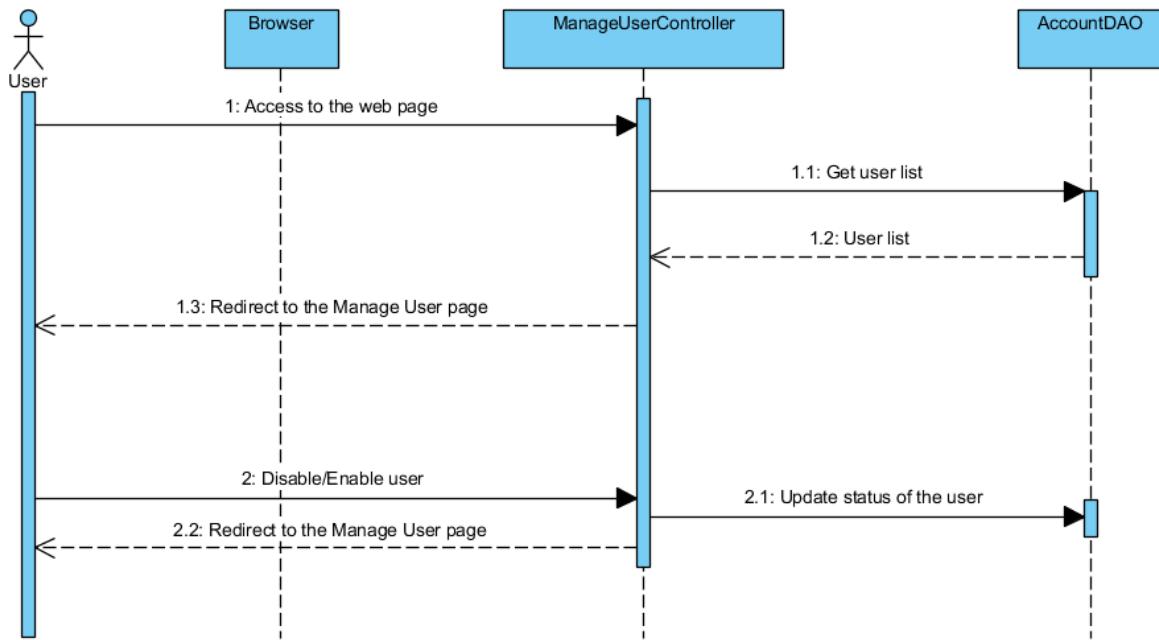
-getTotalAccount: " select count(*) from user where role_id =1 ";
-getTotalRestaurant: " select count(*) from restaurant" ;
-getTotalPostResquest : " select count(*) from post where status = 2" ;
-getTotalRestaurantResquest : " select count(*) from restaurant where status = 2" ;
-getTotalPostEditResquest : " select count(*) from post where status = 4" ;
-getTotalRestaurantEditResquest : " select count(*) from restaurant where status = 4" ;
-getTotalPostReport : " select count(*) from post_report where status = 1" ;
-getTotalRestaurantreport : " select count(*) from restaurant_report where status = 1" ;
  
```

## 4.2. Manage User

### 4.2.1 Class Diagram



#### 4.2.2 Sequence Diagram



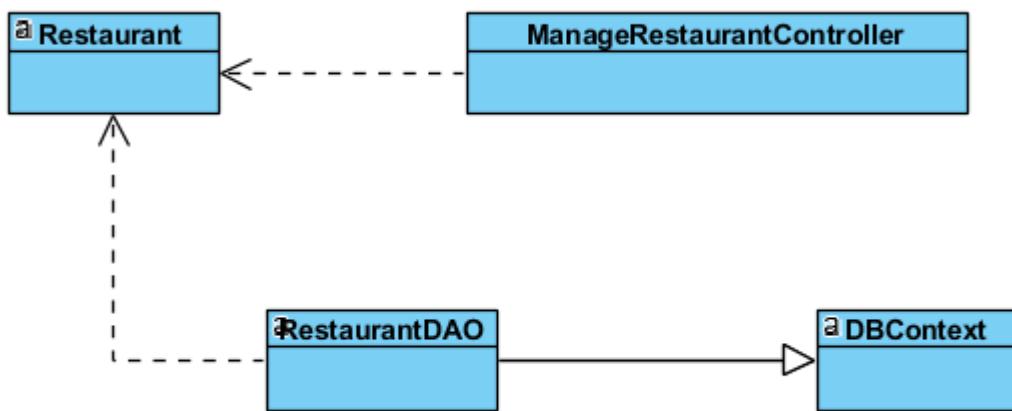
#### 4.2.3 Database Queries

```

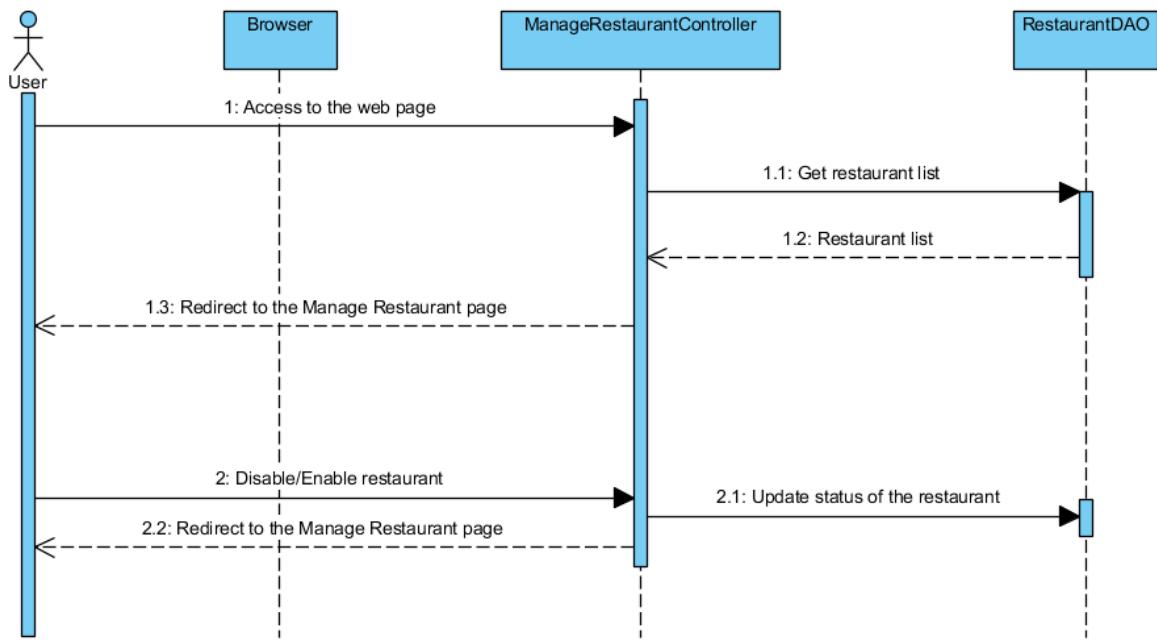
-getListAccount: " select ? from user where role_id =1 ";
-update status: " update user
    set status =?
    where id =? ";
  
```

### 4.3. Manage Restaurant

#### 4.3.1 Class Diagram



#### 4.3.2 Sequence Diagram

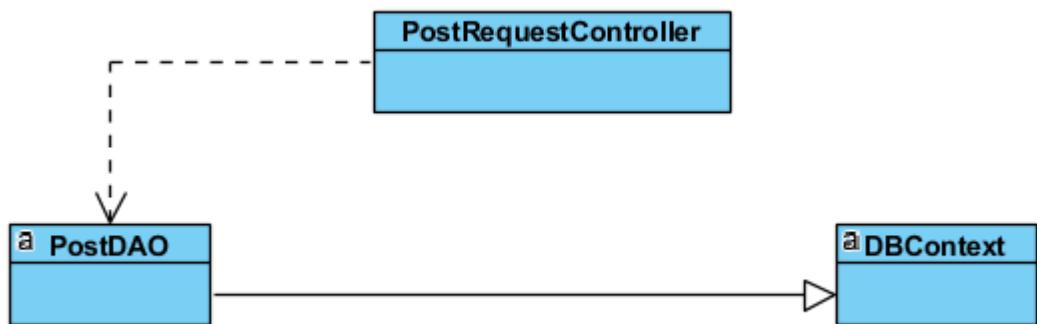


#### 4.3.3 Database Queries

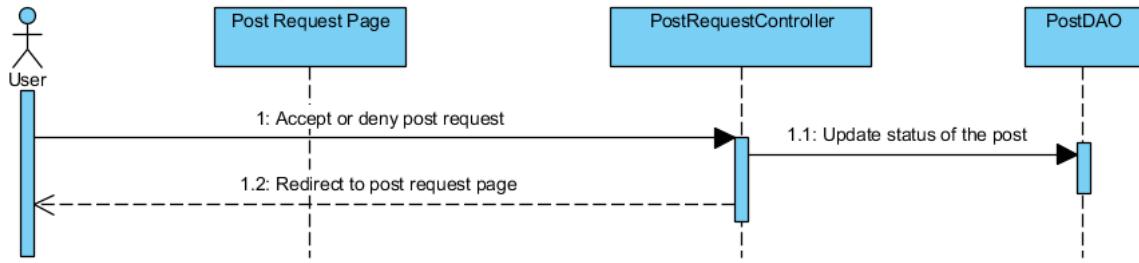
```
-getListRestaurant: " select * from restaurant ";
-update status: " update restaurant
    set status =?
    where id =? ";
```

#### 4.4. Process Post Request

##### 4.4.1 Class Diagram



#### 4.4.2 Sequence Diagram



#### 4.4.3 Database Queries

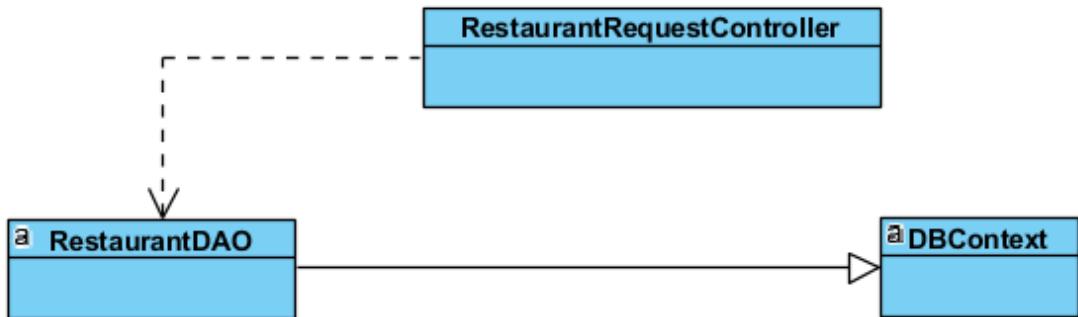
```

-acceptPostRequest: " update post
    set status =1
    where id =? ";
-denyPostRequest: " update post
    set status =3
    where id =? ";

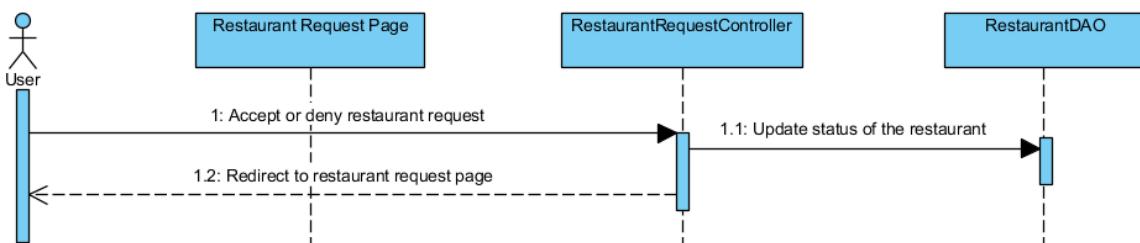
```

### 4.5. Process Restaurant Request

#### 4.5.1 Class Diagram



#### 4.5.2 Sequence Diagram



#### 4.5.3 Database Queries

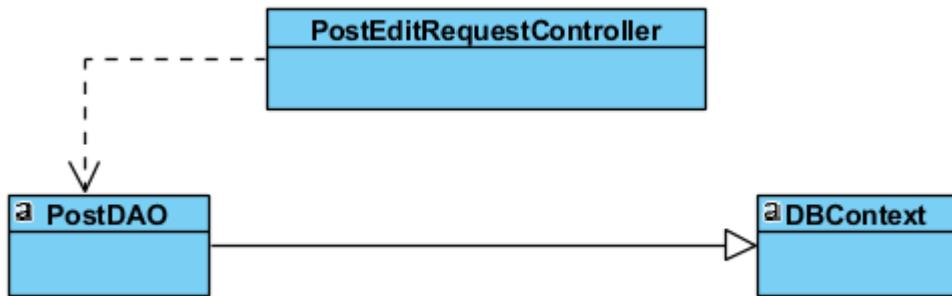
```

-acceptRestaurantRequest: " update restaurant
    set status =1
    where id =? ";
-denyRestaurantRequest: " update restaurant
    set status =3
    where id =? ";

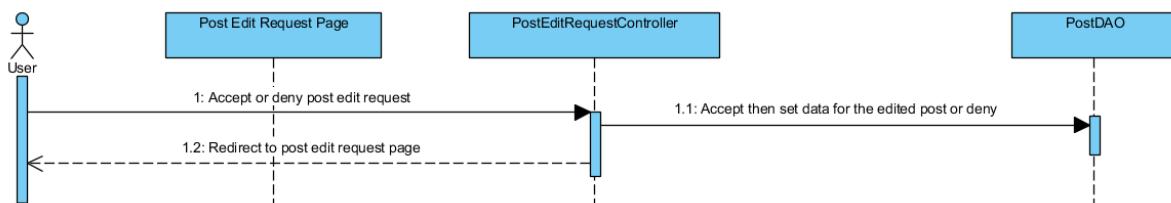
```

## 4.6. Process Request Edit Post

### *4.6.1 Class Diagram*



## 4.6.2 Sequence Diagram



### 4.6.3 Database Queries

-acceptEditPostRequest: """UPDATE g\n'

```
+ "SET \n"
+ "  g.title = e.title,\n"
+ "  g.text = e.text,\n"
+ "  g.rating = e.rating,\n"
+ "  g.status = 1,\n"
+ "  g.service_rate = e.service_rate,\n"
+ "  g.price_rate = e.price_rate,\n"
+ "  g.space_rate = e.space_rate,\n"
+ "  g.taste_rate = e.taste_rate,\n"
+ "  g.hygiene_rate = e.hygiene_rate,\n"
+ "  g.reason_deny = e.reason_deny\n"
+ "FROM post g\n"
+ "JOIN post e ON g.post_id = e.edit_of\n"
+ "WHERE e.post_id = " + edittedPostId + " AND e.edit_of = " + editOfPostId + ";\n"
+ "\n"
+ "DELETE FROM media\n"
+ "WHERE post_id = " + editOfPostId + ";\n"
+ "UPDATE media\n"
+ "SET post_id = " + editOfPostId + "\n"
+ "WHERE post_id = " + edittedPostId + ";\n"
+ "\n"
+ "DELETE FROM post_proof_media\n"
+ "WHERE post_id = " + editOfPostId + ";\n"
+ "UPDATE post_proof_media\n"
+ "SET post_id = " + editOfPostId + "\n"
+ "WHERE post_id = " + edittedPostId + ";\n"
+ "\n"
+ "\n"
```

```

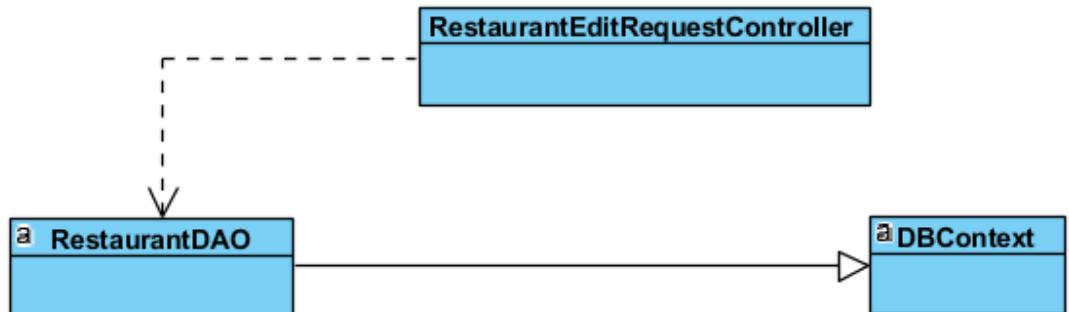
+ "DELETE FROM type_of_meal\n"
+ "WHERE post_id = " + editOfPostId + ";\n"
+ "UPDATE type_of_meal\n"
+ "SET post_id = " + editOfPostId + "\n"
+ "WHERE post_id = " + edittedPostId + ";\n"
+ "\n"
+ "DELETE FROM type_of_food\n"
+ "WHERE post_id = " + editOfPostId + ";\n"
+ "UPDATE type_of_food\n"
+ "SET post_id = " + editOfPostId + "\n"
+ "WHERE post_id = " + edittedPostId + ";\n"
+ "UPDATE post\n"
+ "SET status = 0\n"
+ "WHERE post_id = " + edittedPostId + ";\n"

-denyPostRequest: " update post
    set status =3
    where id =? ";

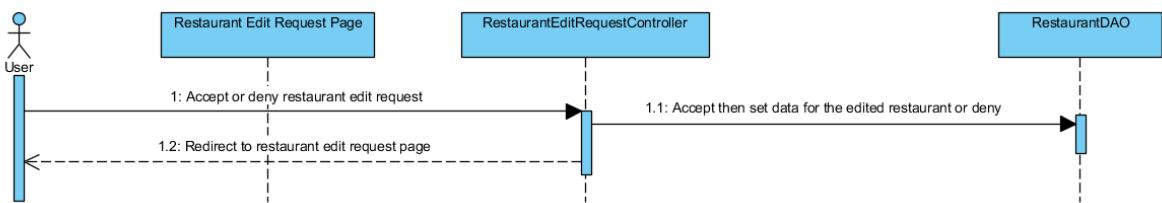
```

## 4.7. Process Edit Restaurant Request

### 4.7.1 Class Diagram



### 4.7.2 Sequence Diagram



### 4.7.3 Database Queries

```

-acceptEditRestaurant: " UPDATE g\n"
    + "SET \n"
    + "  g.name = e.name,\n"
    + "  g.address = e.address,\n"
    + "  g.phone_number = e.phone_number,\n"

```

```

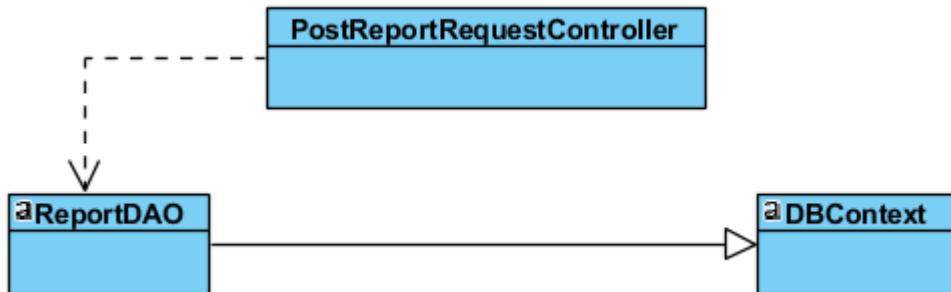
+ "    g.lowest_price = e.lowest_price,\n"
+ "    g.highest_price = e.highest_price,\n"
+ "    g.open_time = e.open_time,\n"
+ "    g.close_time = e.close_time,\n"
+ "    g.avatar = e.avatar,\n"
+ "    g.account_id = e.account_id\n"
+ "FROM restaurant g\n"
+ "INNER JOIN restaurant e ON g.restaurant_id = e.edit_of\n"
+ "WHERE e.restaurant_id = " + editRestaurantId + " AND e.edit_of = " +
editOfRestaurantId + ";\n"
+ "DELETE FROM restaurant_proof_media\n"
+ "WHERE restaurant_id = " + editOfRestaurantId + ";\n"
+ "UPDATE restaurant_proof_media\n"
+ "SET restaurant_id = " + editOfRestaurantId + "\n"
+ "WHERE restaurant_id = " + editRestaurantId + ";\n"
+ "UPDATE restaurant\n"
+ "SET status = 0\n"
+ "WHERE restaurant_id = " + editRestaurantId + ";\n"
+ "\n";

```

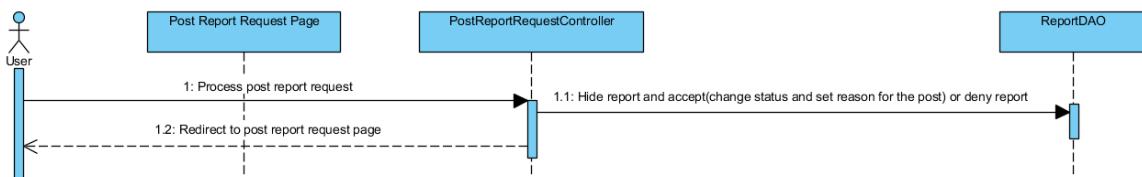
-denyRestaurantRequest: " update restaurant  
set status =3  
where id =? ";

## 4.8. Process Post Report

### 4.8.1 Class Diagram



### 4.8.2 Sequence Diagram



### 4.8.3 Database Queries

-updatePostReport"UPDATE [dbo].[post\_report]\n" + " SET [status] = 0\n"

```

+ " WHERE [post_report_id] = ?";
String sql2 = "UPDATE [dbo].[post]\n"
+ " SET [status] = ?,\n"
+ "     [reason_deny] = ?\n"
+ " WHERE [post_id] = ?";

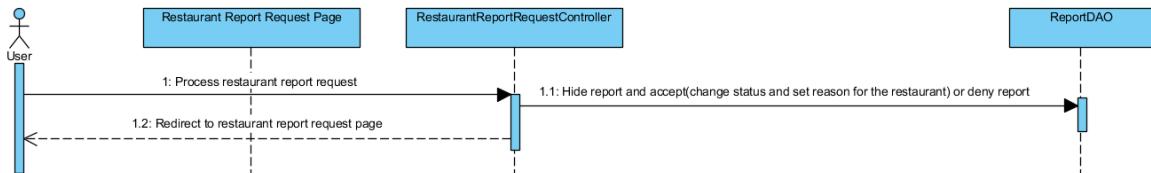
```

## 4.9. Process Restaurant Report

### 4.9.1 Class Diagram



### 4.9.2 Sequence Diagram



### 4.9.3 Database Queries

```

-updatedRestaurantReport"UPDATE [dbo].[restaurant_report]\n"
+ " SET [status] = 0\n"
+ " WHERE [restaurant_report_id]=?";

String sql2 = "UPDATE [dbo].[restaurant]\n"
+ " SET [status] = ?\n"
+ " , [reason_deny] = ?\n"
+ " WHERE [restaurant_id]=?";

```