

**STOPIC**

-CanTho, July 2022-

Table of Contents

[**I. Introduction** 4](#_Toc109387326)

[1. Group members: 4](#_Toc109387327)

[2. Purpose: 4](#_Toc109387328)

[3. Document conventions: 4](#_Toc109387329)

[4. Intended audience and reading suggestions: 4](#_Toc109387330)

[5. Project scope: 4](#_Toc109387331)

[6. References: 4](#_Toc109387332)

[**II. Overall Description** 5](#_Toc109387333)

[1. Product perspective: 5](#_Toc109387334)

[2. Product functions: 5](#_Toc109387335)

[3. Product features: 6](#_Toc109387339)

[4. Use case diagram: 7](#_Toc109387340)

[5. Use case specification: 8](#_Toc109387341)

[6. Sequence diagram: 8](#_Toc109387342)

[7. System features: 25](#_Toc109387362)

[**III. System features** 26](#_Toc109387363)

[1. User Interfaces: 26](#_Toc109387364)

[2. Functional requirements: 27](#_Toc109387365)

[3. Test cases: 28](#_Toc109387366)

[**IV. Non-Function Features** 31](#_Toc109387371)

[1. Performance requirements: 31](#_Toc109387372)

[2. Safety requirements: 31](#_Toc109387373)

[3. Security requirements: 31](#_Toc109387374)

[4. Software quality attributes: 31](#_Toc109387375)

# Introduction

## Group members:

|  |  |
| --- | --- |
| Huynh Quang Tien – Leader | CE150130 |
| Le Xuan Tu | CE150344 |
| Cao Hoang Anh Uy | CE150661 |
| Nguyen Minh Khoi | CE150103 |
| Nguyen Dang Khoa | CE140165 |

## Purpose:

Our application is basically a large group of topics in which registered users can talk about almost anything you can imagine, from news, to pop culture, to technology, to comics, to film, to literature, to the weirdest things in the world.

## Document conventions:

This document uses the following conventions:

|  |  |
| --- | --- |
| DB | Database |
| UC | Use-case |
| UI | User Interface |
| CRUD | Create - Read - Update - Delete |

## Intended audience and reading suggestions:

This project is a type of chat forum like Reddit and Quora, so there is almost no limit on the audience. Anyone can register for an account and participate in interesting topics available on the system.

## Project scope:

The purpose of the project is to create a playground, where everyone can talk and discuss freely about the topic they like. The topic of discussion is almost unlimited in terms of content, so it can range from topics to talk about exchanging experiences to other things like space, politics or exploring the ocean floor. The system is based on a relational database with its comment management, account management and more other functions. Most importantly, we hope to provide a comfortable user experience down the road.

## References:

# Overall Description

## Product perspective:

* + 1. Create an environment for communication between users.
    2. Where users can rate a certain topic through voting.
    3. Guests can also see existing topics on the system without having to log in.
    4. Discussion topics are almost limitless.
    5. Create a topic about a hot news, event or maybe something.

## Product functions:

### Public functions:

1. Register Account
2. Get Comments
3. Get Comment by ID
4. Get Comment by Category ID
5. Get Comment by Parent ID
6. Get Vote Count
7. Get Categories
8. Get Category by ID

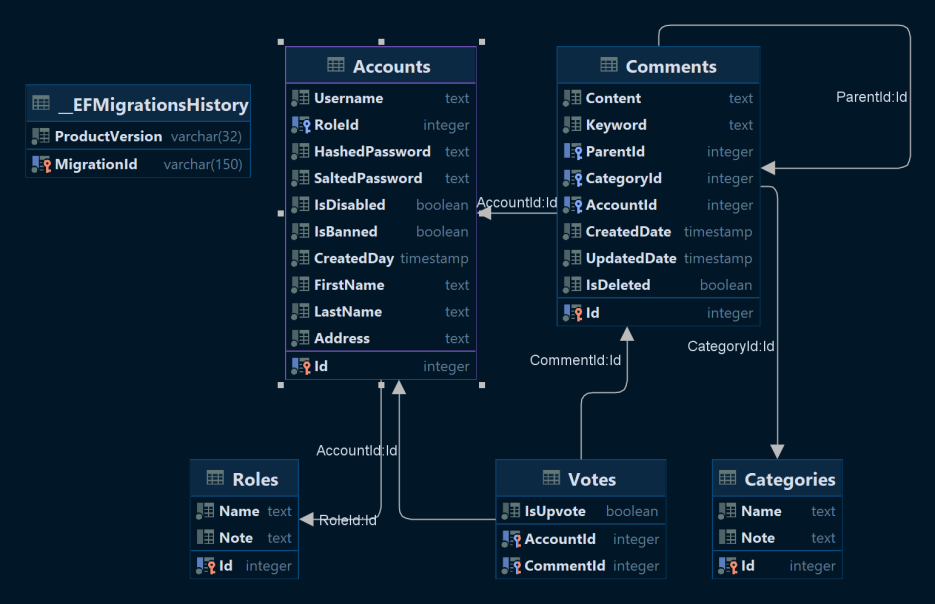
### User functions:

1. Login
2. Register Account
3. Get Comments
4. Get Comment by ID
5. Get Comment by Category ID
6. Get Comment by Parent ID
7. Get Vote Count
8. Get Categories
9. Get Category by ID
10. Handle Vote
11. Update Profile
12. Add Comment
13. Delete Comment
14. Update Comment
15. Get Accounts
16. Get Account by ID

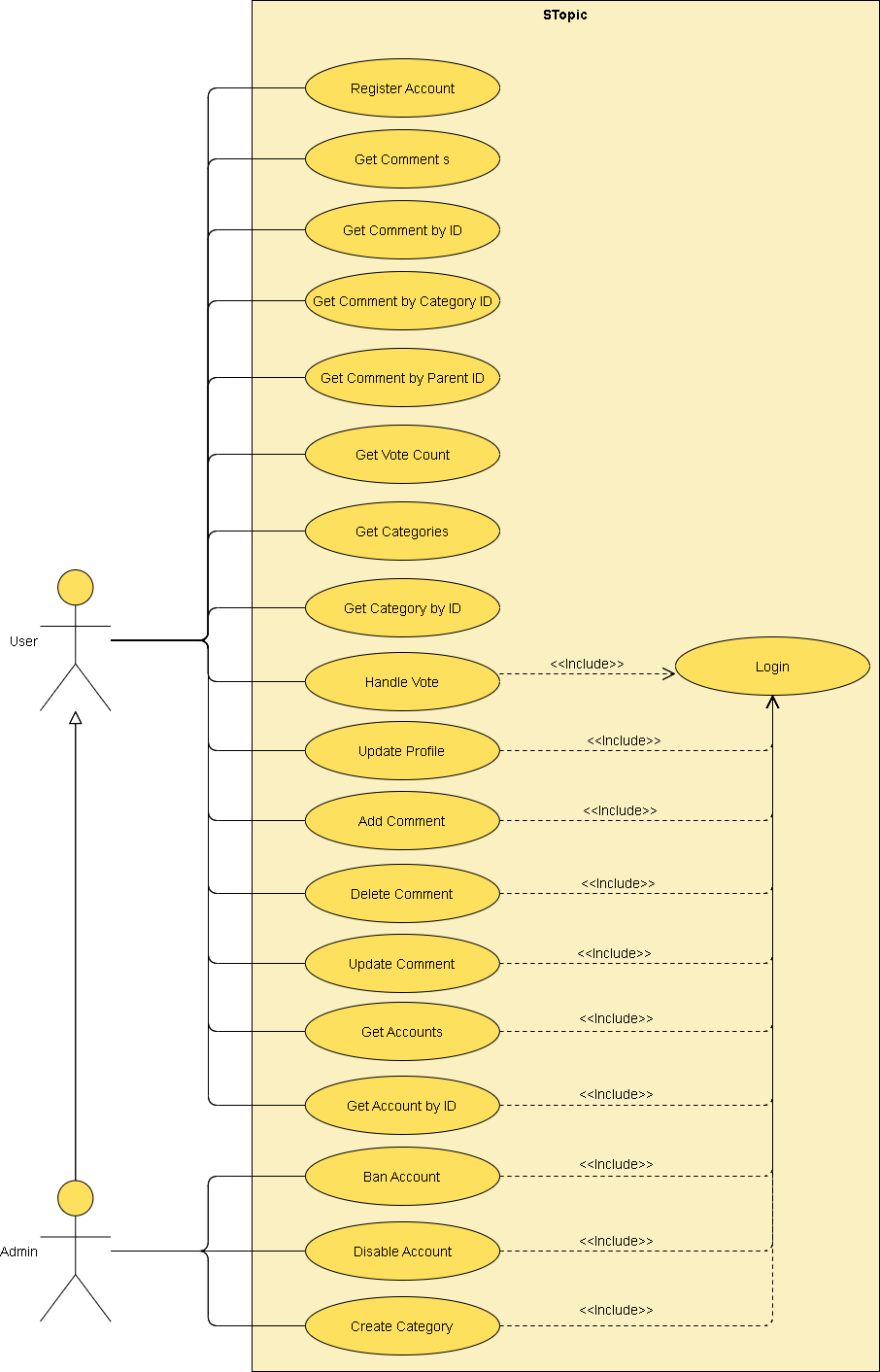
### Admin functions

1. Login
2. Register Account
3. Get Comments
4. Get Comment by ID
5. Get Comment by Category ID
6. Get Comment by Parent ID
7. Get Vote Count
8. Get Categories
9. Get Category by ID
10. Handle Vote
11. Update Profile
12. Add Comment
13. Delete Comment
14. Update Comment
15. Get Accounts
16. Get Account by ID
17. Ban Account
18. Disable Account
19. Create Category

## Product features:



## Use case diagram:



## Use case specification:

|  |  |  |  |
| --- | --- | --- | --- |
| ID and Name: | **UC-1 Login** | | |
| Created By: | Nguyen Dang Khoa | Created Date: | 6/7/2022 |
| Primary Actor: | Users, Admin | Secondary Actors: |  |
| Description: | A method allows users to login into the system and perform some functions . | | |
| Trigger: | A method indicates that user wants to login into the system and already have an account. | | |
| Preconditions: | PRE-1. User accessed the login page.  PRE-2. User already has an account.  PRE-3. The user's device has network connection. | | |
| Postconditions: | POST-1. User login successfully.  POST-2. The system notify to the User. | | |
| Normal Flow: | 1. The user accesses the login page. 2. The user input their account information. 3. The system validates the credentials successfully and allows the user to access the application. 4. The system notify to the User. | | |
| Alternative Flows: | None | | |
| Exceptions: | 3a. The system validates the login info failed and displays a message.  3a1. The user chooses to cancel the login.  *(Use Case stop)*  3b. The user chooses Forget password.  *Use Case continue to Use Case UC-3*  3c. The user chooses Register  *Use Case continue to Use Case UC-2* | | |
| Priority: | High | | |
| Business Rules: | BR-1, BR-4, BR-8, BR-9, BR-10 | | |
| Non-function: | 1. User password must be hashed by HMACSHA1. | | |

|  |  |  |  |
| --- | --- | --- | --- |
| ID and Name: | **UC-2 Register Account** | | |
| Created By: | Nguyen Dang Khoa | Created Date: | 6/7/2022 |
| Primary Actor: | Public, Admin, User | Secondary Actors: |  |
| Description: | A method allows users to register an account. | | |
| Trigger: | A method indicates that he/she wants to register an account. | | |
| Preconditions: | PRE-1. User accessed the register page.  PRE-2. The user's device has network connection. | | |
| Postconditions: | POST-1. User registered successfully.  POST-2. System will check the input account infomation.  POST-3. The system notify to the User. | | |
| Normal Flow: | 1. The user accesses the register page. 2. The user input their account information. 3. The system validates the information successfully. 4. The system notify to the User. | | |
| Alternative Flows: | None | | |
| Exceptions: | 3a. The system validates the register info failed and displays a message.  3a1. The user chooses to cancel the register.  *(Use Case stop)*  3b. The user chooses Login.  *Use Case continue to Use Case UC-1* | | |
| Priority: | High | | |
| Business Rules: | BR-2, BR-3, BR-4, BR-8, BR-9, BR-10, BR-14 | | |
| Non-function: | 1. User password must be hashed by HMACSHA1. | | |

|  |  |  |  |
| --- | --- | --- | --- |
| ID and Name: | **UC-3 Update Profile** | | |
| Created By: | Nguyen Dang Khoa | Created Date: | 6/7/2022 |
| Primary Actor: | Admin, User | Secondary Actors: |  |
| Description: | A method allows users to update an account’s information. | | |
| Trigger: | A method indicates that he/she wants to update an account’s information. | | |
| Preconditions: | PRE-1. User logged in.  PRE-2. User accessed UpdateProfile page.  PRE-3. The user's device has network connection. | | |
| Postconditions: | POST-1. System will check the input account infomation.  POST-2. User updated successfully.  POST-3. The system notify to the User. | | |
| Normal Flow: | 1. The user accesses the UpdateProfile page. 2. The user input their account information. 3. The system validates the information successfully. 4. The system notify to the User. | | |
| Alternative Flows: | None | | |
| Exceptions: | 3a. The system validates the Update failed and displays a message.  3a1. The user chooses to cancel the update.  *(Use Case stop)* | | |
| Priority: | High | | |
| Business Rules: | BR-4, BR-5, BR-8, BR-9, BR-10, BR-11, BR-12, BR-13, BR-14 | | |
| Non-function: |  | | |

|  |  |  |  |
| --- | --- | --- | --- |
| ID and Name: | **UC-4 Disable Account** | | |
| Created By: | Nguyen Dang Khoa | Created Date: | 6/7/2022 |
| Primary Actor: | Admin | Secondary Actors: |  |
| Description: | A method allows admins to disable an account. | | |
| Trigger: | A method indicates that he/she wants to disable an account. | | |
| Preconditions: | PRE-1. User logged in.  PRE-2. User accessed Account page.  PRE-3. The user's device has network connection. | | |
| Postconditions: | POST-1. System will check the account infomation.  POST-2. Account status updated successfully.  POST-3. The system notify to the Admin. | | |
| Normal Flow: | 1. The admin accesses the Account page. 2. The admin click on the Disable button belong to an account. | | |
| Alternative Flows: | None | | |
| Exceptions: | 2a. The system validates the Disable function failed and displays a message. | | |
| Priority: | High | | |
| Business Rules: | BR-2, BR-5, BR-8, BR-9, BR-10, BR-12, BR-14, BR-15 | | |
| Non-function: |  | | |

|  |  |  |  |
| --- | --- | --- | --- |
| ID and Name: | **UC-5 Ban Account** | | |
| Created By: | Nguyen Dang Khoa | Created Date: | 6/7/2022 |
| Primary Actor: | Admin | Secondary Actors: |  |
| Description: | A method allows admins to ban an account. | | |
| Trigger: | A method indicates that he/she wants to ban an account. | | |
| Preconditions: | PRE-1. User logged in.  PRE-2. User accessed Account page.  PRE-3. The user's device has network connection. | | |
| Postconditions: | POST-1. System will check the account infomation.  POST-2. Account status updated successfully.  POST-3. The system notify to the Admin. | | |
| Normal Flow: | 1. The admin accesses the Account page. 2. The admin click on the Ban button belong to an account. | | |
| Alternative Flows: | None | | |
| Exceptions: | 2a. The system validates the Ban function failed and displays a message. | | |
| Priority: | Medium | | |
| Business Rules: | BR-2, BR-5, BR-7, BR-8, BR-9, BR-10, BR-12, BR-14 | | |
| Non-function: |  | | |

|  |  |  |  |
| --- | --- | --- | --- |
| ID and Name: | **UC-6 Get Account by ID** | | |
| Created By: | Nguyen Dang Khoa | Created Date: | 6/7/2022 |
| Primary Actor: | Admin, User | Secondary Actors: |  |
| Description: | A method allows the Admin to view a single account by ID. | | |
| Trigger: | A method indicates that he/she wants to view an account based on its ID. | | |
| Preconditions: | PRE-1. Admin logged in.  PRE-2. Admin accessed the Account page.  PRE-3. The user's device has network connection. | | |
| Postconditions: | POST-1. Application load the wanted Acocunt. | | |
| Normal Flow: | 1. The admin accesses the account list page. 2. Client requests the account from the server. 3. Server return the wanted account to client. | | |
| Alternative Flows: | None | | |
| Exceptions: | 2a. Request Timeout\_the server didn't receive a complete request message within the server's allowed timeout period. | | |
| Priority: | Medium | | |
| Business Rules: | BR-4, BR-5, BR-8, BR-9, BR-10, BR-14 | | |
| Non-function: |  | | |

|  |  |  |  |
| --- | --- | --- | --- |
| ID and Name: | **UC-7 Get Accounts** | | |
| Created By: | Nguyen Dang Khoa | Created Date: | 6/7/2022 |
| Primary Actor: | Admin, User | Secondary Actors: |  |
| Description: | A method allows Admin to view account lists. | | |
| Trigger: | A method indicates that he/she wants to view a account list. | | |
| Preconditions: | PRE-1. Admin logged in.  PRE-2. User accessed the Account page.  PRE-3. The user's device has network connection. | | |
| Postconditions: | POST-1. Application load the Acocunt list. | | |
| Normal Flow: | 1. The Admin accesses the account list page. 2. Client requests the account list from the server. 3. Server return the account list to client. | | |
| Alternative Flows: | None | | |
| Exceptions: | 2a. Request Timeout\_the server didn't receive a complete request message within the server's allowed timeout period. | | |
| Priority: | High | | |
| Business Rules: | BR-4, BR-5, BR-8, BR-9, BR-10, BR-14 | | |
| Non-function: | 1. Only display accounts that are unban and not disabled. | | |

|  |  |  |  |
| --- | --- | --- | --- |
| ID and Name: | **UC-8 Get comments** | | |
| Created By: | Cao Hoang Anh Uy | Created Date: | 6/7/2022 |
| Primary Actor: | Public, Admin, User | Secondary Actors: |  |
| Description: | A method allows Users to view comment lists. | | |
| Trigger: | A method indicates that he/she wants to view a comment list. | | |
| Preconditions: | PRE-1. Users accessed the Comment page.  PRE-2. The user's device has network connection. | | |
| Postconditions: | POST-1. Application load comment list. | | |
| Normal Flow: | 1. The User accesses the comment list page. 2. Client requests the comment list from the server. 3. Server return the comment list to client. | | |
| Alternative Flows: | None | | |
| Exceptions: | 2a. Request Timeout\_the server didn't receive a complete request message within the server's allowed timeout period. | | |
| Priority: | High | | |
| Business Rules: | BR-5, BR-6, BR-8, BR-9, BR-10 | | |
| Non-function: |  | | |

|  |  |  |  |
| --- | --- | --- | --- |
| ID and Name: | **UC-9 Get Comment by ID** | | |
| Created By: | Cao Hoang Anh Uy | Created Date: | 6/7/2022 |
| Primary Actor: | Public, Admin, User | Secondary Actors: |  |
| Description: | A method allows the Admin to view a single comment by ID. | | |
| Trigger: | A method indicates that he/she wants to view an comment based on its ID. | | |
| Preconditions: | PRE-1. Admin logged in.  PRE-2. Admin accessed the Comment Management page.  PRE-3. The user's device has network connection. | | |
| Postconditions: | POST-1. Application load the wanted Comment. | | |
| Normal Flow: | 1. The admin accesses the comment list page. 2. Client requests the account from the server. 3. Server return the wanted comment to client. | | |
| Alternative Flows: | None | | |
| Exceptions: | 2a. Request Timeout\_the server didn't receive a complete request message within the server's allowed timeout period. | | |
| Priority: | Medium | | |
| Business Rules: | BR-4, BR-5, BR-6, BR-8, BR-9, BR-10, BR-14 | | |
| Non-function: |  | | |

|  |  |  |  |
| --- | --- | --- | --- |
| ID and Name: | **UC-10 Get comment by Category id** | | |
| Created By: | Cao Hoang Anh Uy | Created Date: | 6/7/2022 |
| Primary Actor: | Public, Admin, User | Secondary Actors: |  |
| Description: | A method allows users to view comments by its category. | | |
| Trigger: | A method indicates that he/she wants to view a comment form a comment list by specific category. | | |
| Preconditions: | PRE-1. User accessed the comment list page.  PRE-1. User selects comment category from comment list  PRE-2. The user's device has network connection. | | |
| Postconditions: | POST-1. Application load comment by its category. | | |
| Normal Flow: | 1. The user accesses the comment list page. 2. User selects the comment category from the comment list. 3. Client requests the comment by its category id from the server. 4. Server returns the comment to client. | | |
| Alternative Flows: | None | | |
| Exceptions: | 2a. Request Timeout\_the server didn't receive a complete request message within the server's allotted timeout period. | | |
| Priority: | Medium | | |
| Business Rules: | BR-4, BR-5, BR-6, BR-8, BR-9, BR-10, BR-14 | | |
| Non-function: |  | | |

|  |  |  |  |
| --- | --- | --- | --- |
| ID and Name: | **UC-11 Get comment by Parent id** | | |
| Created By: | Cao Hoang Anh Uy | Created Date: | 6/7/2022 |
| Primary Actor: | Public, Admin, User | Secondary Actors: |  |
| Description: | A method allows users to view comments by its category. | | |
| Trigger: | A method indicates that he/she wants to view a child comment form a parent comment. | | |
| Preconditions: | PRE-1. User accessed the comment page.  PRE-2. The user's device has network connection. | | |
| Postconditions: | POST-1. Application load comment by its parent. | | |
| Normal Flow: | 1. The user accesses the specific comment page. 2. Client requests the comment by its parent id from the server. 3. Server returns the comment to client. | | |
| Alternative Flows: | None | | |
| Exceptions: | 2a. Request Timeout\_the server didn't receive a complete request message within the server's allotted timeout period. | | |
| Priority: | Medium | | |
| Business Rules: | BR-4, BR-5, BR-6, BR-8, BR-9, BR-10, BR-14 | | |
| Non-function: |  | | |

|  |  |  |  |
| --- | --- | --- | --- |
| ID and Name: | **UC-12 Add comment** | | |
| Created By: | Cao Hoang Anh Uy | Created Date: | 6/7/2022 |
| Primary Actor: | Admin, User | Secondary Actors: |  |
| Description: | A method allows users to add new comments to the comment list or child comment list. | | |
| Trigger: | A method indicates that he/she wants to add a comment on the comment list or child comment list. | | |
| Preconditions: | PRE-1. User accessed the comment list page or comment page.  PRE-2. Users must be logged in.  PRE-3. Users fill correct comment content.  PRE-4. The user's device has network connection. | | |
| Postconditions: | POST-1. Application add comment into comment list or child comment list. | | |
| Normal Flow: | 1. The user accesses the comment list page or comment page. 2. The user fill in comment content and hit submit button 3. Client requests to add the comment to the server. 4. The server returned the successfully added comment | | |
| Alternative Flows: | None | | |
| Exceptions: | 2a. Request Timeout\_the server didn't receive a complete request message within the server's allotted timeout period.  2b. The user is not logged in.  3a. The comment information entered is not valid. | | |
| Priority: | High | | |
| Business Rules: | BR-4, BR-5, BR-8, BR-9, BR-10, BR-12, BR-13, BR-14 | | |
| Non-function: |  | | |

|  |  |  |  |
| --- | --- | --- | --- |
| ID and Name: | **UC-13 Update comment** | | |
| Created By: | Cao Hoang Anh Uy | Created Date: | 6/7/2022 |
| Primary Actor: | Admin, User | Secondary Actors: |  |
| Description: | A method allows users to update existing comments from the comment list or child comment list. | | |
| Trigger: | A method indicates that he/she wants to update a comment from the comment list or child comment list. | | |
| Preconditions: | PRE-1. User accessed the comment list page or comment page.  PRE-2. Users must be logged in.  PRE-3. Updated comments must be existed.  PRE-4. Updated comments must belong to the user.  PRE-5. Users fill correct comment content.  PRE-6. The user's device has network connection. | | |
| Postconditions: | POST-1. Application update comment into comment list or child comment list. | | |
| Normal Flow: | 1. The user accesses the comment list page or comment page. 2. The user selects a comment to update and hits the update button. 3. The user fill in comment content and hit submit button 4. Client requests to update the comment to the server. 5. The server returned the successfully updated comment | | |
| Alternative Flows: | None | | |
| Exceptions: | 2a. Request Timeout\_the server didn't receive a complete request message within the server's allotted timeout period.  4a. The comment information entered is not valid.  4b. The comment did not exist.  4c. The comment did not belong to the logged user. | | |
| Priority: | Medium | | |
| Business Rules: | BR-4, BR-5, BR-8, BR-9, BR-10, BR-11, BR-12, BR-13, BR-14 | | |
| Non-function: |  | | |

|  |  |  |  |
| --- | --- | --- | --- |
| ID and Name: | **UC-14 Delete comment** | | |
| Created By: | Cao Hoang Anh Uy | Created Date: | 6/7/2022 |
| Primary Actor: | Admin, User | Secondary Actors: |  |
| Description: | A method allows users to delete existing comments away from the comment list or child comment list. | | |
| Trigger: | A method indicates that he/she wants to delete a comment away from the comment list or child comment list. | | |
| Preconditions: | PRE-1. User accessed the comment list page or comment page.  PRE-2. Users must be logged in.  PRE-3. Deleted comments must exist.  PRE-4. Deleted comments must belong to the user.  PRE-5. The user's device has network connection. | | |
| Postconditions: | POST-1. Application delete comment from comment list or child comment list. | | |
| Normal Flow: | 1. The user accesses the comment list page or comment page. 2. the user selects a comment to delete and hits the delete button. 3. Client requests to delete the comment to the server. 4. The server returned the successfully deleted comment | | |
| Alternative Flows: | None | | |
| Exceptions: | 2a. Request Timeout\_the server didn't receive a complete request message within the server's allotted timeout period.  3a. The comment information entered is not valid.  3b. The comment did not exist.  3c. The comment did not belong to the logged user. | | |
| Priority: | Medium | | |
| Business Rules: | BR-5, BR-8, BR-9, BR-10, BR-11, BR-12, BR-13, BR-14 | | |
| Non-function: |  | | |

|  |  |  |  |
| --- | --- | --- | --- |
| ID and Name: | **UC-15 Handle Vote** | | |
| Created By: | Nguyen Minh Khoi | Created Date: | 6/7/2022 |
| Primary Actor: | Admin, User | Secondary Actors: |  |
| Description: | A method allows users to up-vote or downvote a comment. | | |
| Trigger: | A method indicates that he/she wants to up-vote or down-vote a comment. | | |
| Preconditions: | PRE-1. User accessed the comment list page or comment page.  PRE-2. Users must be logged in.  PRE-3. Voted comment must be existed.  PRE-4. The user's device has network connection. | | |
| Postconditions: | POST-1. Application update vote of a comment. | | |
| Normal Flow: | 1. The user accesses the comment list page or comment page. 2. The user selects up-vote or down-vote button beside a comment. 3. Client requests to update the comment to the server. 4. The server returned the successfully updated comment | | |
| Alternative Flows: | None | | |
| Exceptions: | 2a. Request Timeout\_the server didn't receive a complete request message within the server's allotted timeout period.  3a. The comment did not exist. | | |
| Priority: | Medium | | |
| Business Rules: | BR-5, BR-8, BR-9, BR-10, BR-14 | | |
| Non-function: |  | | |

|  |  |  |  |
| --- | --- | --- | --- |
| ID and Name: | **UC-16 Has Voted** | | |
| Created By: | Nguyen Minh Khoi | Created Date: | 6/7/2022 |
| Primary Actor: | Admin, User | Secondary Actors: |  |
| Description: | A method allows users to see if a comment is voted. | | |
| Trigger: | A method indicates that he/she wants to see a comment is voted or not. | | |
| Preconditions: | PRE-1. User accessed the comment list page or comment page.  PRE-2. Users must be logged in.  PRE-3. Voted comment must be existed.  PRE-4. The user's device has network connection. | | |
| Postconditions: | POST-1. Application update vote of a comment. | | |
| Normal Flow: | 1. The user accesses the comment list page or comment page. 2. The server returned the successfully list of comments. | | |
| Alternative Flows: | None | | |
| Exceptions: | 1a. Request Timeout\_the server didn't receive a complete request message within the server's allotted timeout period.  2a. The comment did not exist. | | |
| Priority: | Medium | | |
| Business Rules: | BR-5, BR-8, BR-9, BR-10, BR-14 | | |
| Non-function: |  | | |

|  |  |  |  |
| --- | --- | --- | --- |
| ID and Name: | **UC-17 Get Vote Count** | | |
| Created By: | Nguyen Minh Khoi | Created Date: | 6/7/2022 |
| Primary Actor: | Public, Admin, User | Secondary Actors: |  |
| Description: | A method allows Admin to see a total of votes for a comment. | | |
| Trigger: | A method indicates that he/she wants to see how many votes for a comment. | | |
| Preconditions: | PRE-1. User accessed the comment list page or comment page.  PRE-2. Users must be logged in.  PRE-3. Voted comment must be existed.  PRE-4. The user's device has network connection. | | |
| Postconditions: | None | | |
| Normal Flow: | 1. The user accesses the comment list page. 2. The server returned the successfully. | | |
| Alternative Flows: | None | | |
| Exceptions: | 1a. Request Timeout\_the server didn't receive a complete request message within the server's allotted timeout period.  2a. The comment did not exist. | | |
| Priority: | Medium | | |
| Business Rules: | BR-5, BR-8, BR-9, BR-10, BR-12, BR-13, BR-14 | | |
| Non-function: |  | | |

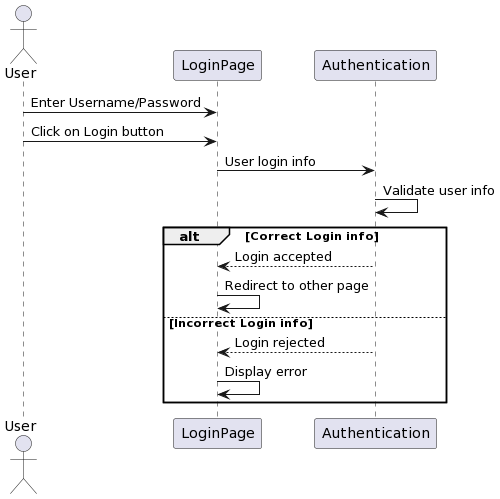
|  |  |  |  |
| --- | --- | --- | --- |
| ID and Name: | **UC-18 Get Categories** | | |
| Created By: | Nguyen Minh Khoi | Created Date: | 6/7/2022 |
| Primary Actor: | Public, Admin, User | Secondary Actors: |  |
| Description: | A method allows Users to view category list. | | |
| Trigger: | A method indicates that he/she wants to view a category list. | | |
| Preconditions: | PRE-1. Users accessed the Topic page.  PRE-2. The user's device has network connection. | | |
| Postconditions: | POST-1. Application load category list. | | |
| Normal Flow: | 1. The Users accesses the topic list page. 2. Client requests the category list from the server. 3. Server return the category list to client. | | |
| Alternative Flows: | None | | |
| Exceptions: | 2a. Request Timeout\_the server didn't receive a complete request message within the server's allowed timeout period. | | |
| Priority: | High | | |
| Business Rules: | BR-5, BR-6, BR-8, BR-9, BR-10, BR-12 | | |
| Non-function: |  | | |

|  |  |  |  |
| --- | --- | --- | --- |
| ID and Name: | **UC-19 Get Category by ID** | | |
| Created By: | Nguyen Minh Khoi | Created Date: | 6/7/2022 |
| Primary Actor: | Public, Admin, User | Secondary Actors: |  |
| Description: | A method allows Users to view a category by ID. | | |
| Trigger: | A method indicates that he/she wants to view a category. | | |
| Preconditions: | PRE-1. Users accessed the Topic page.  PRE-2. The user's device has network connection. | | |
| Postconditions: | POST-1. Application load category list. | | |
| Normal Flow: | 1. The Users accesses the topic list page. 2. Client requests the category by its ID from the server. 3. Server return the category to client. | | |
| Alternative Flows: | None | | |
| Exceptions: | 2a. Request Timeout\_the server didn't receive a complete request message within the server's allowed timeout period. | | |
| Priority: | High | | |
| Business Rules: | BR-5, BR-6, BR-8, BR-9, BR-10, BR-12 | | |
| Non-function: |  | | |

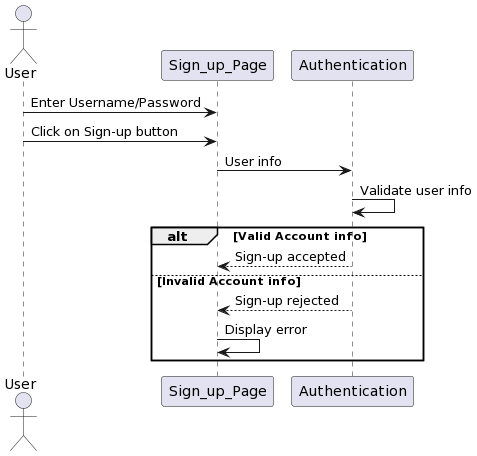
|  |  |  |  |
| --- | --- | --- | --- |
| ID and Name: | **UC-20 Create Category** | | |
| Created By: | Nguyen Minh Khoi | Created Date: | 6/7/2022 |
| Primary Actor: | Admin | Secondary Actors: |  |
| Description: | A method allows Admin to add new category. | | |
| Trigger: | A method indicates that he/she wants to add a category into the category list. | | |
| Preconditions: | PRE-1. Admin accessed the category list page.  PRE-2. Users must be logged in.  PRE-3. The user's device has network connection. | | |
| Postconditions: | POST-1. Application add comment into comment list or child comment list. | | |
| Normal Flow: | 1. The Admin accesses the Category list page. 2. The Admin fill in Category information and hit submit button. 3. Client requests to add the category to the server. 4. The server returned the successfully added category. | | |
| Alternative Flows: | None | | |
| Exceptions: | 2a. Request Timeout\_the server didn't receive a complete request message within the server's allotted timeout period.  2b. The user is not logged in.  3a. The category information entered is not valid. | | |
| Priority: | High | | |
| Business Rules: | BR-4, BR-5, BR-8, BR-9, BR-10, BR-12, BR-13, BR-14 | | |
| Non-function: |  | | |

## Sequence diagram:

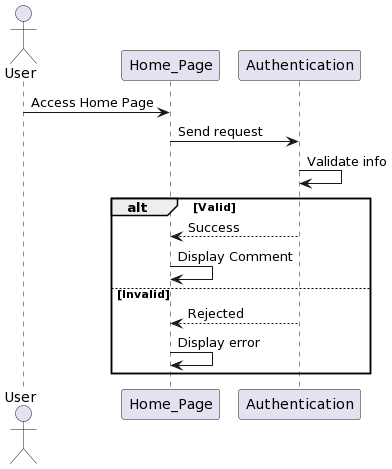
### Login



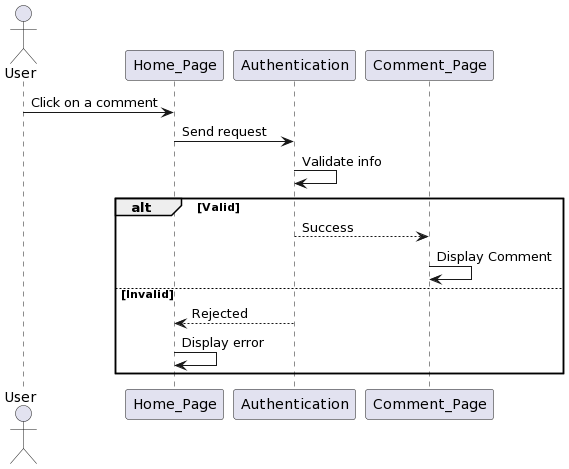
### Register



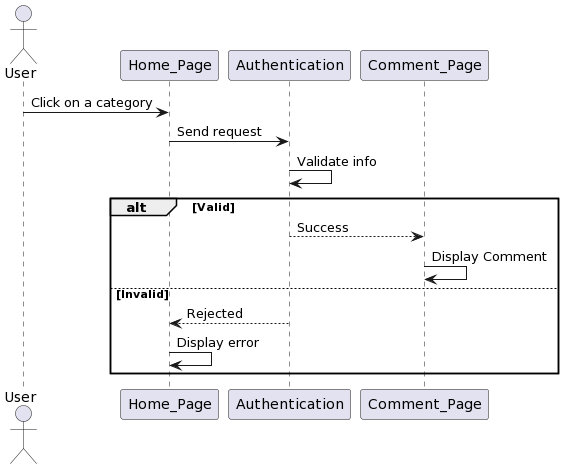
### Get Comments



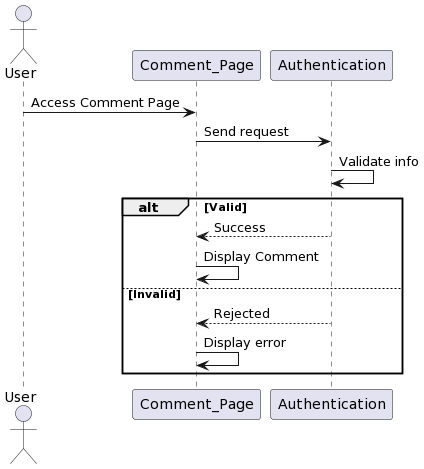
### Get Comment by ID



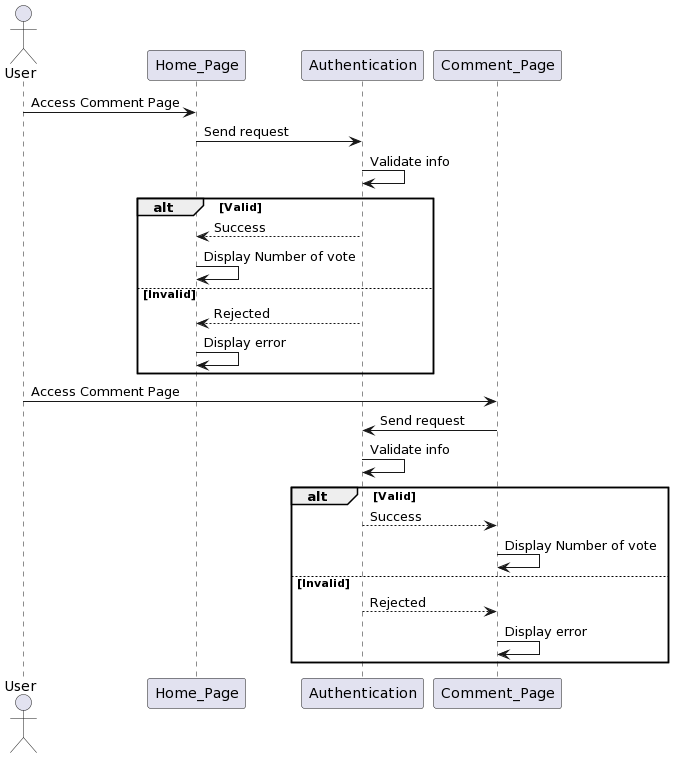
### Get Comment by Category ID



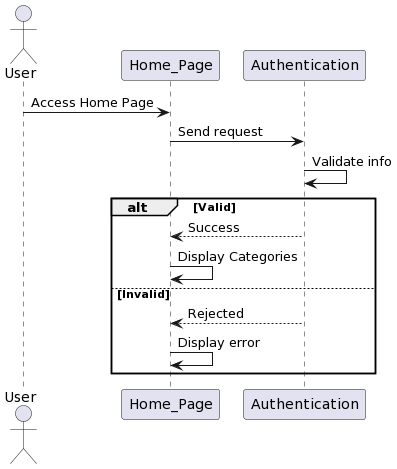
### Get Comment by Parent ID



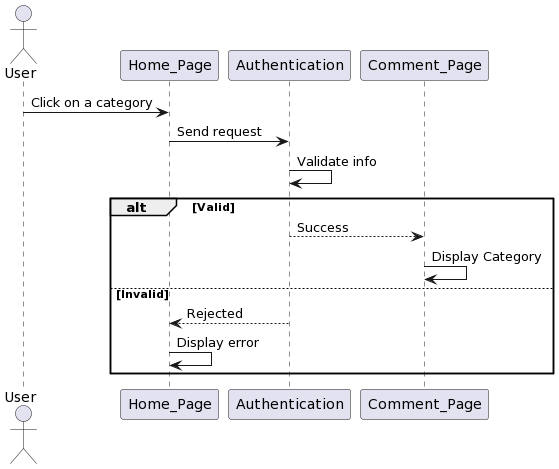
### Get Vote Count



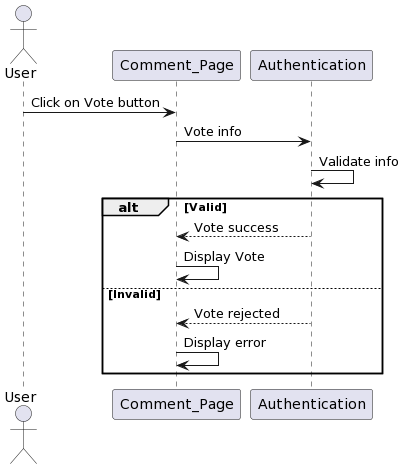
### Get Categories



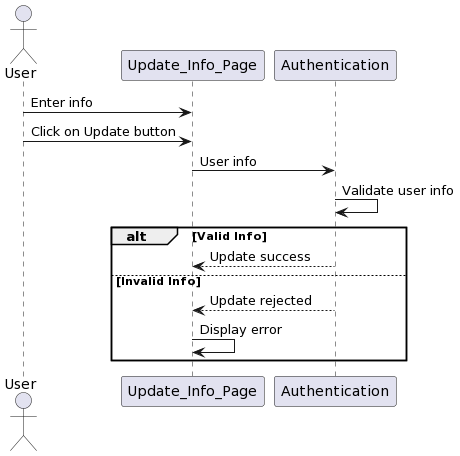
### Get Category by ID



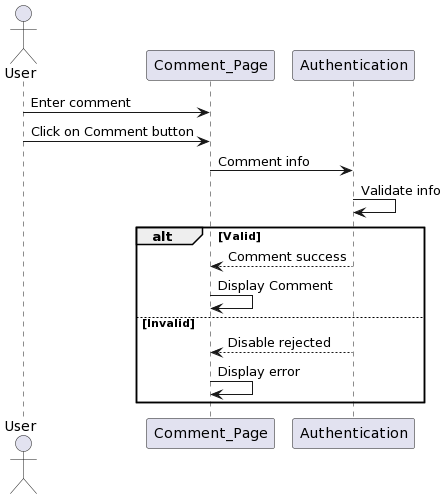
### Handle Vote



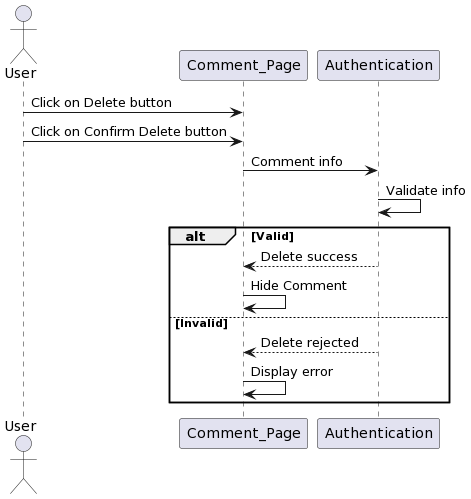
### Update Profile



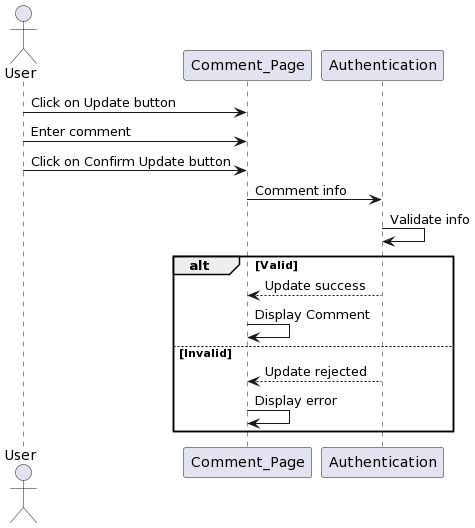
### Add Comment



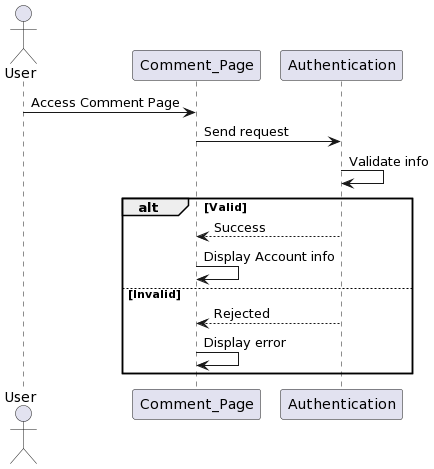
### Delete Comment



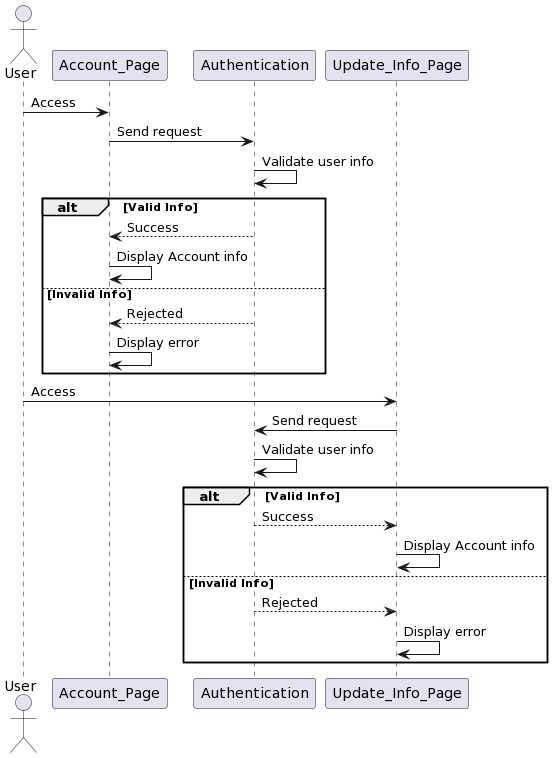
### Update Comment



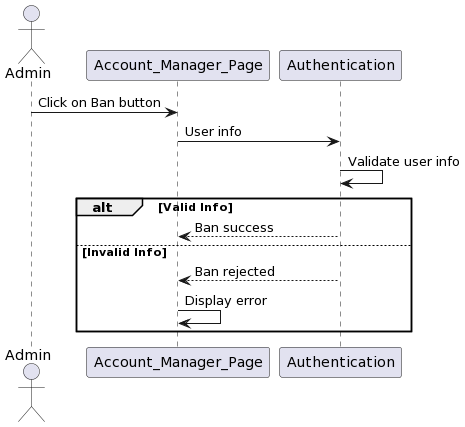
### Get Accounts



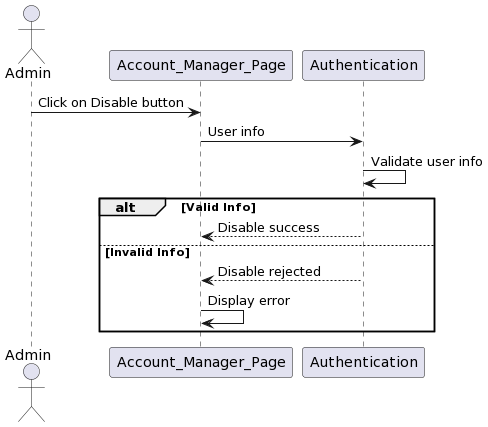
### Get Account by ID



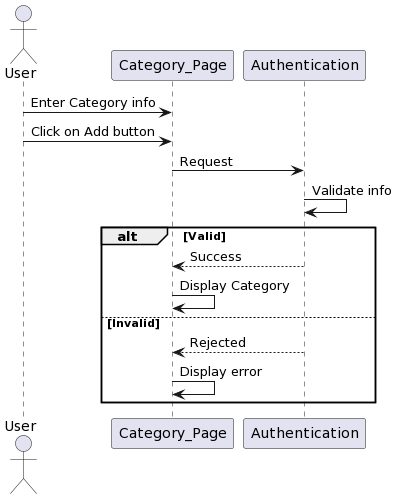
### Ban Account



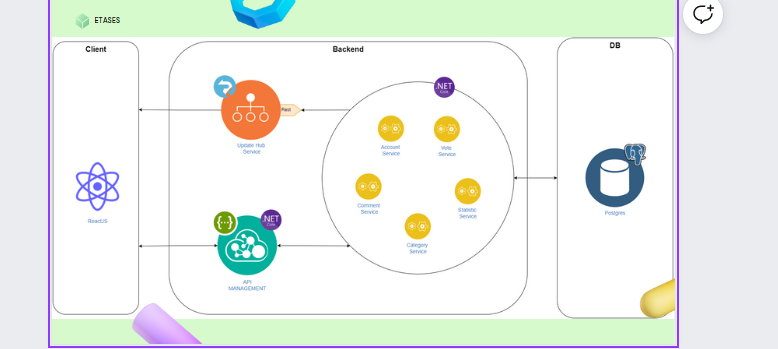
### Disable Account



### Create Category

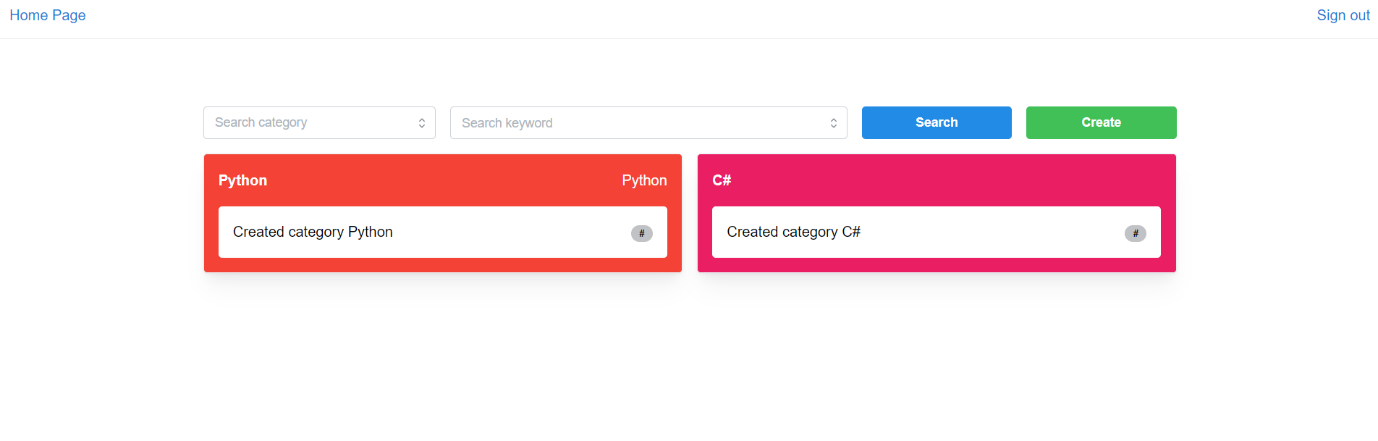
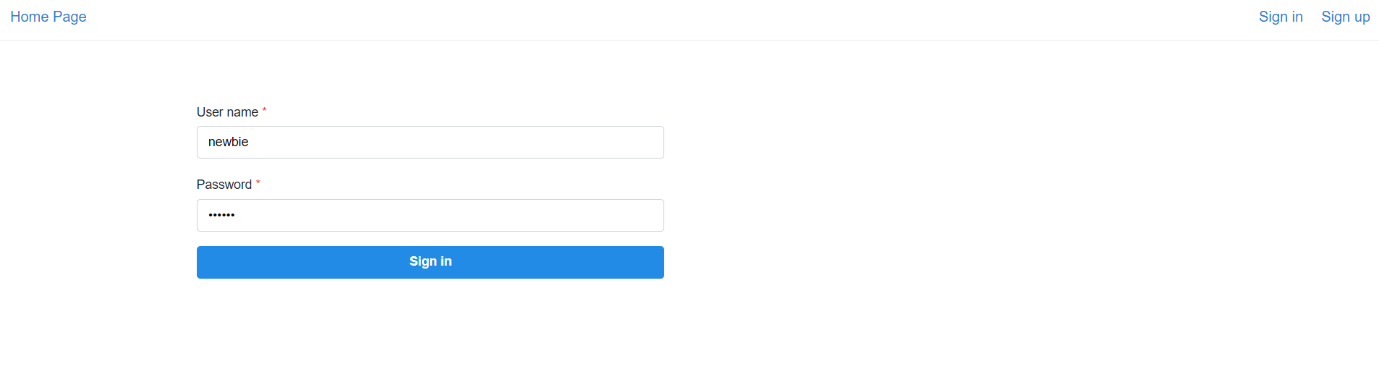
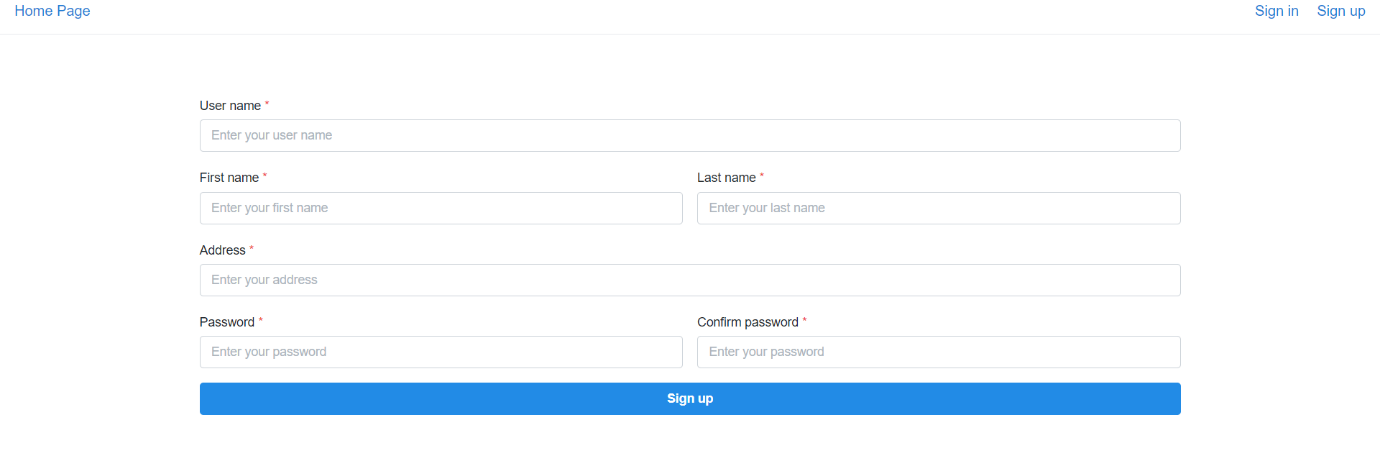
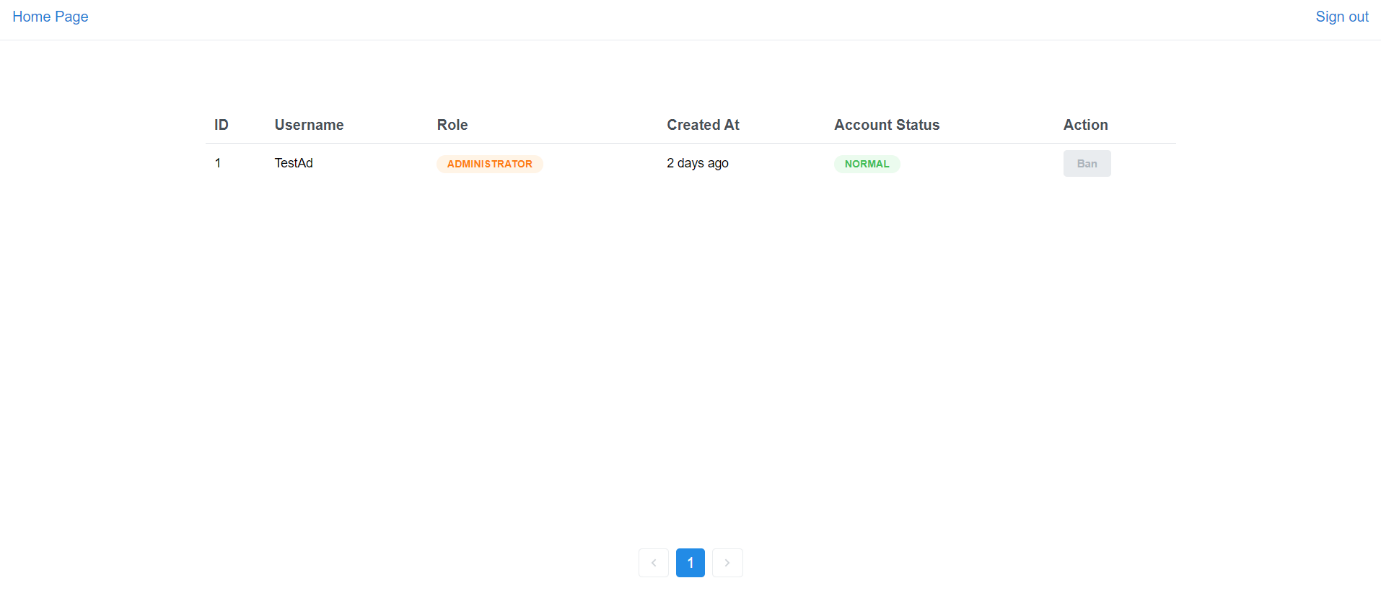
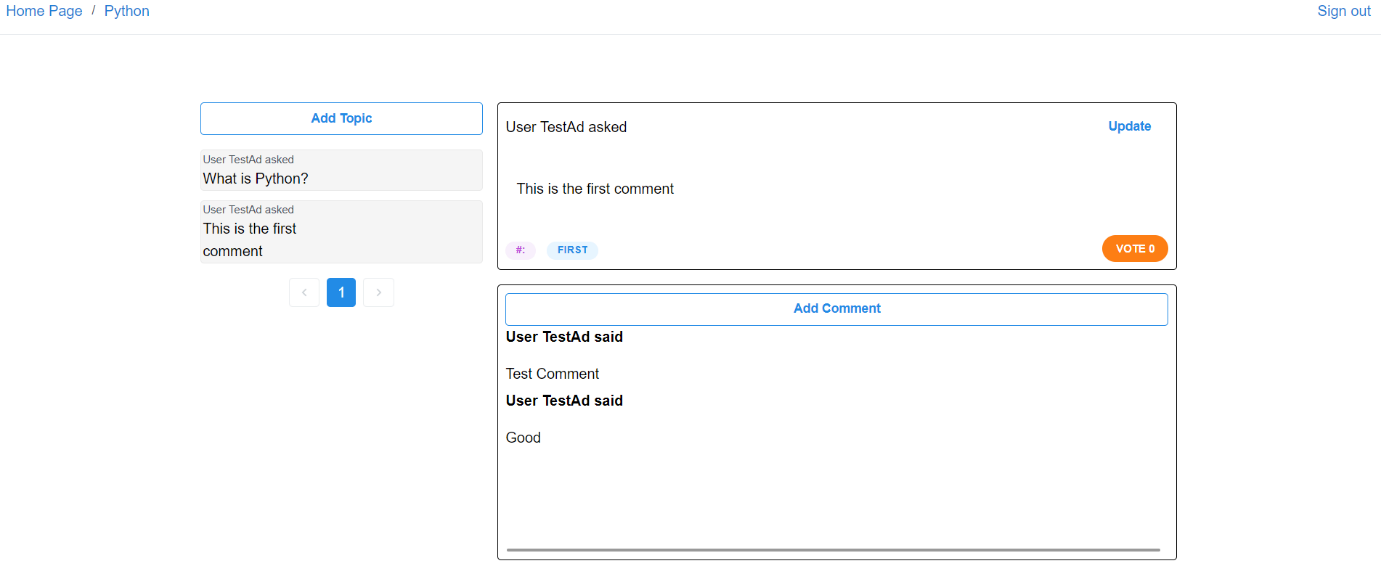


## System features:



# System features

## User Interfaces:



## Functional requirements:

* + 1. Business rules:

|  |  |  |
| --- | --- | --- |
| ***ID*** | ***Rule Definition*** | ***Source*** |
| BR-1 | Only users with a valid account can log in into the system | Authentication and security |
| BR-2 | Every user has their own unique account ID | Account database |
| BR-3 | Account’s password will be encrypted and can not be known by anyone but its user after completing the registration | Authentication and security |
| BR-4 | Restrictions on the format of the input information to avoid system errors | Authentication and security |
| BR-5 | User must login into system to be able to make transactions or edits | Authentication and security |
| BR-6 | Public users can view a peek of the information | Authentication and security |
| BR-7 | Admin can ban users out of the system if they have inappropriate behaviors | Account database |
| BR-8 | UI must be simple and easy to understand | UI |
| BR-9 | Error messages should be clear and easy to understand | UI |
| BR-10 | Content displayed must be honest and accurate | UI |
| BR-11 | Users have full discretion regarding their account information | Account Database |
| BR-12 | Admin can not cannot create false or fabricated information. | Database |
| BR-13 | Admin can not interfere into the log database | Security |
| BR-14 | User information will be protected and not provided or sold to outside parties. | Security |
| BR-15 | Admin can disable users for a period of time if they have inappropriate behaviors | Account database |

1. Transaction corrections, adjustments, and cancellations:  
   *These requirements examine every transaction’s entry, changing, deleting, canceling, and error checking.*
2. Authentication functions:  
   They concern the information users share with the system and their authentication level.
3. Authorization levels:  
   These functions determine various system access levels and decide who can CRUD information.
4. Searching/reporting requirements:  
   This section of requirements will tell you how users can search and retrieve data.
5. Compliance, legal, or regulatory requirements   
   These are laws, regulations from the government, and even internal policies that the organizations and their systems must follow.

## Test cases:

### **Account test**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Name** | **Test Function** | **Input** | **Expected Result** | **Actual Result** |
| Account Register | AccountRegister | input valid username, password, first name, last name, address | return status: 200 | return 200 if valid, 400 if invalid. |
| Account Login | Login | Input valid username and password | return new Bearer token | return Bearer token if login sucess, return 404 if account not exist |
| Account Update | UpdateAccount | Input valid firstname, lastName, address | return status: 200 | return 200 if success, 400 if invalid |
| Account ban | BanAccount | Input accountId and new ban status | return status: 200 | return 200 if sucess, 401 if not admin |
| Account Disable | DisableAccount | Input accountId and Visibility status | return status: 200 | return 200 if success, 401 if not admin |

### **Category test**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Name** | **Test Function** | **Input** | **Expected Result** | **Actual Result** |
| Get All Category | GetAllCategory | user send request to api’s url | return status: 200 and list of all category | return 200 if sucess, return 404 if not found. |
| Create New Category | CreateCategory | input category name | return status: 200 | return status: 200 if create success, 401 if not admin |
| Get Category By Id | GetCategoryById | input categoryId | return status: 200 and category information contain the Id | return 200 if success, 404 if not found |

### **Comment test**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Name** | **Test Function** | **Input** | **Expected Result** | **Actual Result** |
| Get topic By Id | GetTopicById | user input topicId and send request to api’s url | return status: 200 and content of topic | return 200 if sucess, return 404 if not found. |
| Update comment By Id | UpdateCommentById | input commentId, content and keyword | return status: 200 and 404 if not found | return status: 200 if create success, 404 if comment not found |
| Delete comment by Id | DeleteCommentById | input commentId and send delete request to the api’s url | return status: 200 if success, 404 if comment not found | return 200 if success, 404 if comment not found |
| Filter topic | FilteredTopic | input categoryId or keyword | return status 200 and topic data (if not provide categoryId or keyword will get all topic), return empty object if nothing match | return status 200 and topic data (if not provide categoryId or keyword will get all topic), return empty object if nothing match |
| Get Keywords List | GetKeywordsList | input keyword | return status 200 and list of keywords content the data we’ve provide, return empty empty list if nothing match | input keyword |
| Get Comment by Parent’s Id | GetCommentByParent | input parentId | return status 200 and list of comment contain parentId, return empty list if nothing found | return status 200 and list of comment contain parentId, return status empty list if nothing found |
| Create new comment | CreateComment | input content, keyword, parentId, categoryId | return status 200 if create success, 401 if user not authorize | return status 200 if create success, 401 if user not authorize |

### **Vote test**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Name** | **Test Function** | **Input** | **Expected Result** | **Actual Result** |
| Vote Action | Vote | Input commentId and vote status (if votestatus is true mean upvote in contrast mean downvote if provide the same status it’s mean unvote) | return status: 200 if success and system will update vote status, return 401 if not authorize | return status: 200 if success and system will update vote status, return 401 if not authorize |
| Check If user has voted the comment | HasVoted | provide commentId | return status: 200 and vote information, return 404 if nothing found | return status: 200 and vote information, return 404 if nothing found |
| Count Total Vote | VoteCount | Provide commentId | return status: 200 and total vote of the comment, return 404 if not found | return status: 200 and total vote of the comment, return 404 if not found |

# Non-Function Features

## Performance requirements:

* The system must respond with high speed whenever there is a request of a certain feature from the user.

## Safety requirements:

* The password of each account must be encrypted and be invulnerable to administrators.
* The password of each account must meet a certain number of characters and format to ensure safety.

## Security requirements:

* It is necessary to be logged into the system to be able to transact and interact with data and system functions.
* Functional requirements can be directly tested and observed.

## Software quality attributes:

* The application should be user-friendly.
* For development, it should be easy to add code to the existing system, should be easy to upgrade for new features and new technologies from time to time.
* The application should be correct in terms of its functionality, calculations used internally and the navigation should be correct.
* The system should be easy to test and find defects. If required, it should be easy to divide into different modules for testing.

**The End**