**FoodForFun**

**Development’s Guide**

**Faculty: Pham Cong Thanh**

**Batch No: 2017-1609**

**Group No: SE1168**

**Student:**

|  |  |  |
| --- | --- | --- |
| **No** | **Full Name** | **Roll No** |
| **1** | **Hoang  Yen Minh** | **SE62327** |
| 2 | Nguyen Hoang Minh | SE62286 |
| 3 | Dao Nhat Minh | SE62295 |

**Table of contents**

Table of contents............................................................................................................................. 2

**Review 1**............................................................................................................ 3

1.      Introduction............................................................................................................................... 4

2.      Existing Scenario........................................................................................................................ 4

3.      Customer Requirement Specification....................................................................................... 4

4.      Functional Requirement Specification...................................................................................... 5

5.      System Requirements................................................................................................................ 9

6.      Development Software.............................................................................................................. 9

7.      Technology................................................................................................................................ 9

**Review 2**.......................................................................................................... 10

8.      Use Case and Sequence Diagram............................................................................................ 11

9.      Entity Relationship Diagram (ERD).......................................................................................... 73

**Review 3**.......................................................................................................... 78

10.  Database Relationship Diagram.............................................................................................. 79

11.  Table Definition....................................................................................................................... 80

12.  Other Constraint...................................................................................................................... 86

13.  Site map................................................................................................................................... 88

14.  GUI Design............................................................................................................................... 90

15.  Report.................................................................................................................................... 166

**Task sheet**...................................................................................................... 170

Task sheet.................................................................................................................................... 171

**Monitoring Report**......................................................................................... 174

Unit Test...................................................................................................................................... 175

**REVIEW 1**

**1.** **Introduction**

**FoodForFun** is a company that specialized in professional PR for restaurants, food stores about their food. They started off in HCM city by doing lot of surveys to get the information about the local markets, local restaurants. They took their food information, demand of users. Now, they create a website to post reviews about what they had surveyed for the diners in the city and a place for every restaurants or food stores can post advertisement about their stores (view, food, address, price).

**2.** **Existing Scenario**

Nowadays, there are lots of restaurants and food stores have their own websites to introduce about their places and food. So, they decide to create a website to summarize all of them into only one website. Besides that, the website can be a place to advertise for any restaurant or small food store, which doesn’t have enough budgets to make their own website. It makes user feel more convenient and comfortable to find a suitable place for their meals.

**3.** **Customer Requirement Specification**

A site needs to be designed where user will register either as a member, editor and moderator.

If registered as an editor then we can accept the posts of a moderator. Post, edit, delete the comments, posts.

If registered as a moderator, we can post the advertisement for restaurants, food stores.

If registered as a member, user can read the post or find places for their meals. They also can comment to review about the restaurants or save any of their favorite places.

A user can also log-in to the system as a guest. User only can read the post or find place for their meals. They don’t have permissions to comment or save any of their favorite restaurant.

**4. Functional Requirement Specification**

**4.1. Guest can register as Moderator or Member**

**4.1.1.**        **Description**

The system must provide a page to allow guest create their accounts with the following information:

·         User Name

·         Password

·         Name

·         Birthday

·         Sex

·         Address

·         Phone

·         Email

·         Select to be Moderator or Member (an account cannot be Moderator and Member)

**4.1.2.**        **Functional Requirements**

REQ-01.User must provide information to create an account.

. Username

. Password

. Name

. Birthday

. Sex

. Address

. Phone

. Email

REQ-02.      User must select Moderator or Member.

REQ-03.      Username are unique.

**4.2. Guest can login as Moderator, Editor or Member**

**4.2.1.**        **Description**

The system must provide a page to allow guest login as Editor, Moderator or Member.

. Username

. Password

**4.2.2.**        **Functional Requirements**

REQ-01.Customers must provide Username and Password to login into the website.

REQ-02.System must distinguish Customer type to display pages which customers can view.

REQ-03.Customers can only access to pages they have permission.

**4.3. Enquiry from Member, Moderator**

**4.3.1.**        **Description**

The system must provide a page to allow guests send enquiries.

. Username

. Title

. Content

. Date of feedback

**4.3.2.**        **Functional Requirements**

REQ-01.The enquiries from guests must be sent to feedback page for admin.

**4.4. Management of Moderator**

**4.4.1.**        **Description**

System must provide for Moderator a management page their profile.

. Name

. Birthday

. Sex

. Email

. Address

. Phone

**4.4.2.**        **Functional Requirements**

REQ-01.        Moderator can modify their profile such as:

. Name

. Birthday

. Sex

. Email

. Address

. Phone

**4.5. Management of Member**

**4.5.1.**        **Description**

System must provide for Membera management page.

. Email

. Address

. Phone

**4.5.2.**        **Functional Requirements**

REQ-01.        Member can modify their profile.

. Email

. Address

. Phone

**4.6. Member Comments**

**4.6.1.**        **Description**

System must provide for Member a text area to write a comment and view all of other member’s comments.

. Content

. Time of post

. Username

**4.6.2.**        **Functional Requirements**

REQ-01.        Member can comment on the post such as:

. Content

. Time of post

. Username

**4.7. Management Post Of Moderator**

**4.7.1.**        **Description**

System must provide for Moderator a page to write and edit the post:

. Image

. Title

. Content

. Date of post

. Restaurant’s address

. Author

. Category

. Menu

**4.7.2.**        **Functional Requirements**

REQ-01.        Moderator can post the article.

REQ-02.        Moderator can edit the article.

**4.8. Editor Inspect The Post**

**4.8.1.**        **Description**

System must provide for Editor a page to inspect the posts.

. Title

. Content

. Image

. Date of post

. Restaurant’s address

. Author

. Type of restaurant

. Menu

**4.8.2.**        **Functional Requirements**

REQ-01.        Editor can inspect the post.

REQ-02.        Editor can disable the post.

**4.9. Supporting directories**

**4.9.1.**        **Description**

System must provide list of directories support for Customers. The Editor can manage this list.

. Place

. Area

. Type of food

**4.9.2.**        **Functional Requirements**

REQ-01.        Customers can view list of directories, Directory detail.

. Place

. Area

. Type of food

REQ-02.        Editor can manage directories, such as add new, modify, delete.

**4.10.**        **GUI Requirements**

**4.10.1.**  **Description**

System should provide to Customers an attractive home page.

**4.10.2.**  **Functional Requirements**

REQ-01.        An attractive home page.

**4.11.**        **Editor can view information of Member or Moderator**

**4.11.1.**  **Description**

System should provide for Editor a page which allows they view list of Members or Moderators, view Member’s information or Moderator’s information.

. Username

. Name

. Birthday

. Sex

. Address

. Phone

. Email

**4.11.2. Functional Requirements**

REQ-01.        Editor can view Member’s information.

. Username

. Name

. Birthday

. Sex

. Address

. Phone

. Email

REQ-02.        Editor can view list of Members

**4.12.**        **Editor can block of Member**

**4.12.1.**  **Description**

System should provide a function for Editor to block Member.

. Member’s username

**4.12.2. Functional Requirements**

REQ-01. Editor can block Member

**4.13.**        **Ranking Posts base on like of Member**

**4.13.1.**  **Description**

System should provide a function for Member Like a post and rank top 10 posts.

**4.13.2 Functional Requirements**

REQ-01. Member can like a post.

REQ-02. System rank a post bases on number of like.

**4.14.**        **Member, Moderator and Editor can change password**

**4.13.1.**  **Description**

System should provide a function for Member and Moderator change their password.

**4.13.2 Functional Requirements**

REQ-01. Member can change their password.

REQ-02. Moderator can change their password.

REQ-03. Editor can change their password.

**4.15.**        **Editor can add new Editor**

**4.13.1.**  **Description**

System should provide a function Editor Add new Editor.

**4.13.2 Functional Requirements**

SEQ-01. Editor can add new Editor.

**5.    System Requirements**

**5.1. Hard Ware**

**5.1.1.**        **Web Server**

|  |  |
| --- | --- |
| Processor | Pentium IV or higher. |
| Memory | 512MB RAM or greater. |
| Modem/ADSL | Connecting to internet 24/24 |

**5.1.2.**        **Client**

|  |  |
| --- | --- |
| Processor | Pentium III or higher. |
| Memory | 64MB RAM or greater. |
| Monitor | Super VGA (800x600) or higher resolution with 256 colors. |
| Pointing Device | Microsoft Mouse, Microsoft IntelliMouse®, or compatible pointing device. |
| Modem/ADSL | Connecting to internet |

**5.2. Software**

**5.2.1.**        **Web Server**

|  |  |
| --- | --- |
| Operation System | Windows 7, 8, 10. |
| Browser | Google Chrome.. |
| Database | Microsoft SQL Server 2014. |
| Software | Java JDK 1.8. |
|  | Apache Tomcat 5.5 |

**5.2.2.**        **Client**

|  |  |
| --- | --- |
| Operation System | Windows 7 or later. |
| Browser | Google Chrome. |

# 6. Development Software

* Microsoft Windows 7,8,10
* Netbean 8.2
* Java JDK 1.8
* Apache Tomcat 5.5
* Microsoft SQL Server 2014 SP2
* Google Chrome

# 7. Technology

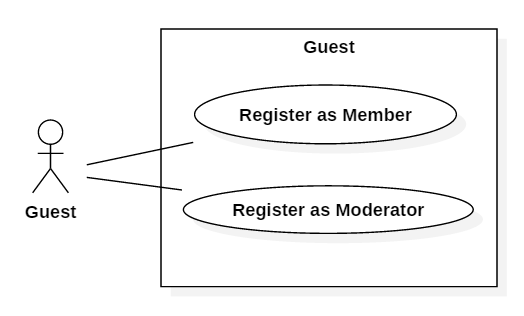
* J2EE
* Java Web Service
* JSP, Servlet
* HTML, CSS, JS
* Github

|  |  |  |
| --- | --- | --- |
|  | **Prepared by (Student)** | **Approved by (Faculty)** |
| **Signature** | Team leader  **Hoang Yen Minh** | **Pham Cong Thanh** |
| **Date** |  |  |

**REVIEW 2**

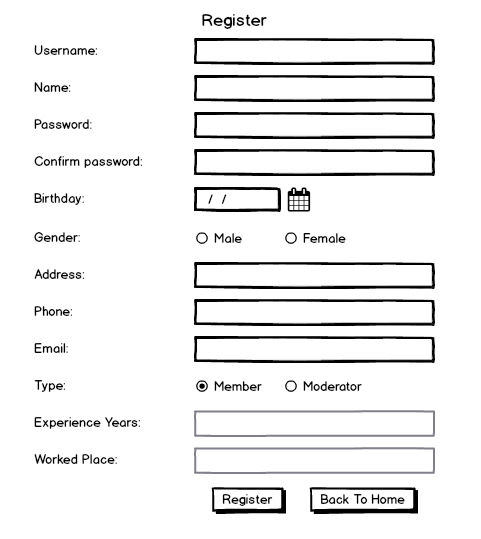
**8. Use Case and Sequence Diagram**

**8.1. Guest**



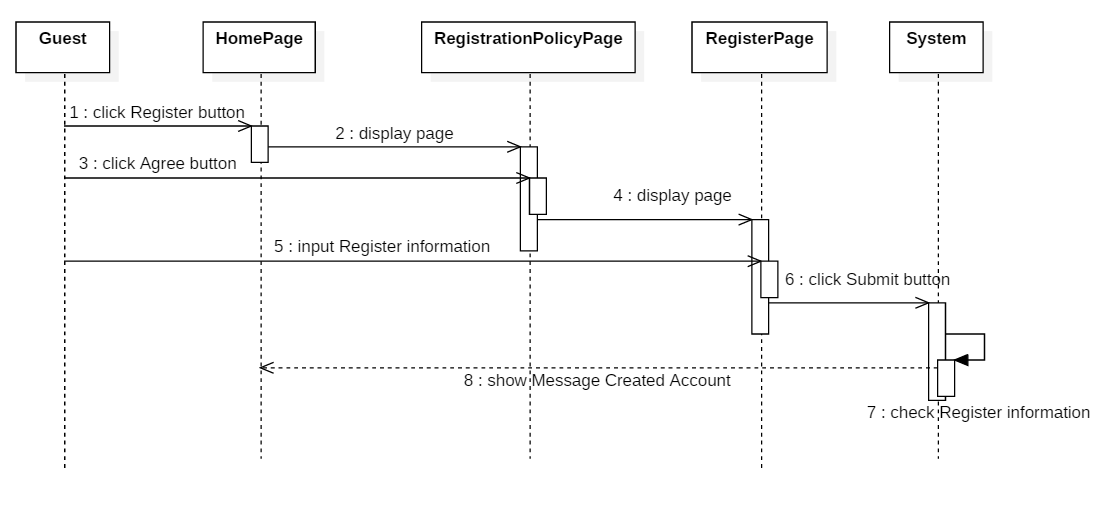
**8.1.1. Register as Member**

|  |  |  |
| --- | --- | --- |
| **Author** | Hoang Yen Minh | |
| **Use Case Name** | Register as Member | |
| **Actors** | Guest | |
| **Description** | Guest wants to register with the website to become a member. | |
| **Requirements** | Guest provides Guest’s information | |
| **Pre-conditions** |  | |
| **Post-conditions** | Success: Guest creates a Member account in this website  Fail: Refill information | |
| **Basic flow** | Actor Actions:  1. Actor clicks ‘Register’ link in Home page  3. Actor clicks on to Agree button.  5. Actor inputs register’s information, select ‘Member’ radio button and click ‘Register’ button. [Alternative 1] | System Responses:  2. System redirects to Registration Policy page.  4. System redirects to Register page and displays Registration form with the following control :  - ‘UserName’ text field  - ‘Password’ text field  - ‘Confirm Password’ text field  - ‘Name’ text field  - ‘Birthday’ datebox  - ‘Gender’ checkbox group  - ‘Address’ text field  - ‘Phone’ text field  - ‘Email’ text field  - ‘Member’ radio button  - ‘Moderator’ radio button  - ‘Register’ button  - ‘Back to Home’ button  6. System checks the information  [Exception 1]  7. System inserts the account into database, show message: ‘Your account has been created!’ and then redirects to Login page. |
| **Alternative flow** | Actor Actions:  [Alternative 1]  1. Actor click ‘Back to Home’ button. | System Responses:  2. System returns to Home page. |
| **Exception** | Actor Actions:  [Exception 1]  1. Actor lefts ‘UserName’ text field blank.  2. UserName has already exists.  3. Actor lefts ‘Password’ text field blank.  4. Actor lefts ‘Confirm Password’ text field blank.  5. Password and confirm password don’t match.  6. Actor lefts ‘Name’ text field blank.  7. Actor lefts ‘Birthday’ datebox blank.  8. Actor lefts ‘Address’ text field blank.  9. Actor lefts ‘Phone’ text field blank.  10. Input phone not in numeric or phone length not in range 8 – 11 character.  11. Actor lefts ‘Email’ text field blank or input an invalid email address. | System Responses:  System displays message to Actor :  - ‘UserName cannot be blank’.  - ‘This Username has already exist!’.  - ‘Password cannot be blank’’.  - ‘Confirm Password can’t be blank’.  - ‘Confirm password not match password.’  - ‘Name cannot be blank’.  - ‘Birthday is invalid’.  - ‘Address cannot be blank’.  - ‘Phone is invalid’.  - ‘The phone number should be numeric only and must be between 8 and 11 digits’.  - ‘Email is invalid’. |

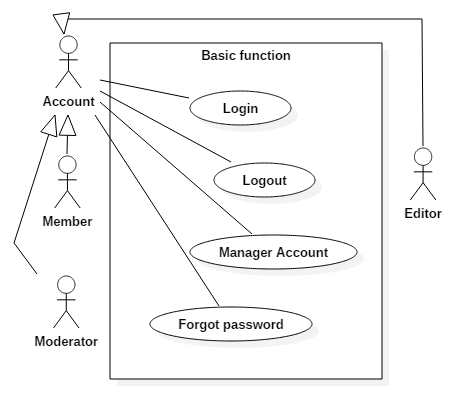


**8.1.2. Register as Moderator**

|  |  |  |
| --- | --- | --- |
| **Author** | Hoang Yen Minh | |
| **Use Case Name** | Register as Moderator | |
| **Actors** | Guest | |
| **Description** | Guest wants to register with the website to become a Moderator. | |
| **Requirements** | Guest provides Guest’s information | |
| **Pre-conditions** |  | |
| **Post-conditions** | Success: Guest creates a Moderator account in this website  Fail: Refill information | |
| **Basic flow** | Actor Actions:  1. Actor clicks ‘Register’ link in Home page  3. Actor clicks on to Agree button.  5. Actor inputs register’s information, select ‘Moderator’ radio button.  7. Actor click ‘Register’ button [Alternative 1] | System Responses:  2. System redirects to Registration Policy page.  4. System redirects to Register page and displays Registration form with the following control :  - ‘UserName’ text field  - ‘Password’ text field  - ‘Confirm Password’ text field  - ‘Name’ text field  - ‘Birthday’ datebox  - ‘Gender’ checkbox group  - ‘Address’ text field  - ‘Phone’ text field  - ‘Email’ text field  - ‘Member’ radio button  - ‘Moderator’ radio button  - ‘Register’ button  - ‘Back to Home’ button  6. System Display  - ‘Experience Years‘ text field  - ‘Worked Place’ text field  8. System checks the information  [Exception 1]  7. System inserts the account into database, show message: ‘Your account has been created!’ and then redirects to Login page. |
| **Alternative flow** | Actor Actions:  [Alternative 1]  1. Actor click ‘Back to Home’ button. | System Responses:  2. System returns to Home page. |
| **Exception** | Actor Actions:  [Exception 1]  1. Actor lefts ‘UserName’ text field blank.  2. UserName has already exists.  3. Actor lefts ‘Password’ text field blank.  4. Actor lefts ‘Confirm Password’ text field blank.  5. Password and confirm password don’t match.  6. Actor lefts ‘Name’ text field blank.  7. Actor lefts ‘Birthday’ datebox blank.  8. Actor lefts ‘Address’ text field blank.  9. Actor lefts ‘Phone’ text field blank.  10. Input phone not in numeric or phone length not in range 8 – 11 character.  11. Actor lefts ‘Email’ text field blank or input an invalid email address.  12. Actor lefts ‘Experience Years‘ text field blank  13. Input ‘Experience Years‘ not in numeric length not bigger than age | System Responses:  System displays message to Actor :  - ‘UserName cannot be blank’.  - ‘This Username has already exist!’.  - ‘Password cannot be blank’’.  - ‘Confirm Password can’t be blank’.  - ‘Confirm password not match password.’  - ‘Name cannot be blank’.  - ‘Age must over 16’.  - ‘Address cannot be blank’.  - ‘Phone is invalid’.  - ‘The phone number should be numeric only and must be between 8 and 11 digits’.  - ‘Email is invalid’.  - “Experience Years cannot be blank”  - “Experience Years is invaild” |

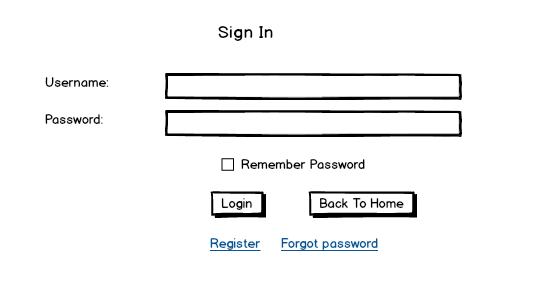


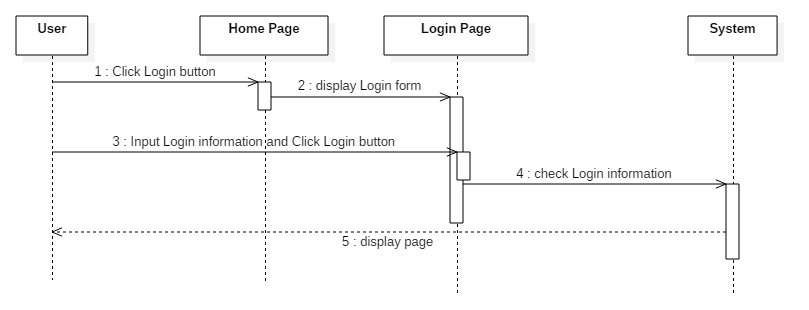
**8.2. Member, Moderator, Editor**

****

**8.2.1. Login**

|  |  |  |
| --- | --- | --- |
| **Author** | Nguyen Hoang Minh | |
| **Use Case Name** | Login | |
| **Actors** | Member or Moderator or Editor | |
| **Description** | Member or Moderator or Editor wants to login to website as a Member or Moderator or Editor. | |
| **Requirements** | Actor provides username and password | |
| **Pre-conditions** | User registered. | |
| **Post-conditions** | Success: User logged in to website.  Fail: Refill information. | |
| **Basic flow** | Actor Actions:  1. Actor go to Home page of website and click ‘Login’ button in submenu.  [Alternative 1] | System Responses:  2. System redirects to Home page and displays Login form with the following control:  - ‘UserName’ text field  - ‘Password’ text field  - ‘Login’ button  - ‘Back to Home’ button  - ‘Register’ hyperlink  - ‘Forget Password’ hyperlink  - ‘Remember Account’ checkbox |
| **Alternative flow** | Actor Actions:  [Alternative 1]  1. Actor inputs UserName and Password, then click the ‘Login’ button.  [Alternative 2]  1.Actor click ‘Back to Home’ button.  [Alternative 3]  1. Actor click ‘Register’ hyperlink.  [Alternative 4]  1. Actor click ‘Forget Password’ hyperlink.  [Alternative 5]  1. Actor inputs UserName and Password, click ‘Remember password’ checkbox, then click the ‘Login’ button. | System Responses:  2. System checks the information  [Exception 1]  3. System redirects to Actor’s Home page.  2. System return to Home page.  2. System return to Register page form.  2. System return to ‘Forget Password’ form.  2. System remember password into cookie.  2. System checks the information.  [Exception 1]  3. System redirects to Actor’s Home page. |
| **Exception** | Actor Actions:  [Exception 1]  1. Actor input invalid username and password. | System Responses:  System show a message box: “Invalid Username and password”. |



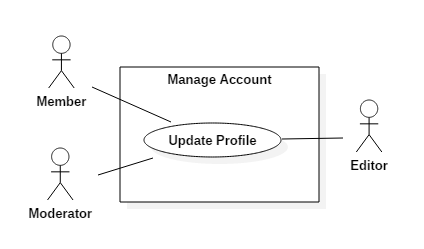


**8.2.2. Logout**

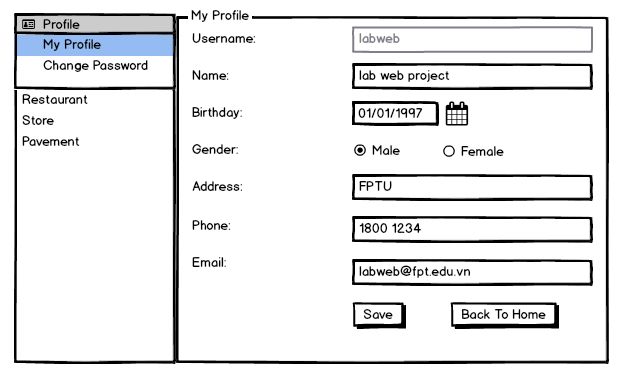
|  |  |  |
| --- | --- | --- |
| **Author** | Nguyen Hoang Minh | |
| **Use Case Name** | Logout | |
| **Actors** | Member or Moderator or Editor | |
| **Description** | Member or Moderator or Editor wants to log out of the System. | |
| **Requirements** |  | |
| **Pre-conditions** | Actor has logged in. | |
| **Post-conditions** | Success: Actor Logged out. | |
| **Basic flow** | Actor Actions:  1. Actor clicks ‘Log out’ button on the Actor’s Home menu.  3. Actor clicks ‘Yes’ button.[Alternative 1] | System Responses:  2. System redirect to display logout confirm form with message “Are you sure?”  4. System remove session, and then redirect to Home page. |
| **Alternative flow** | [Alternative 1]   1. Actor clicks ‘No’ button. | 2. System turn back to actor Home page. |
| **Exception** |  |  |

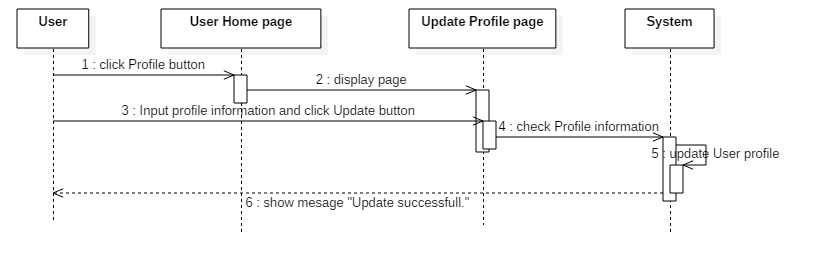


**8.2.3. Manage Account**

****

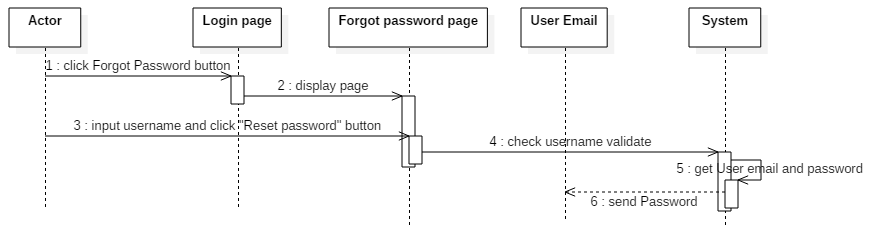
|  |  |  |
| --- | --- | --- |
| **Author** | Nguyen Hoang Minh | |
| **Use Case Name** | Update Profile | |
| **Actors** | Member or Moderator or Editor | |
| **Description** | Member or Moderator or Editor wants to Update his/her profile’s information. | |
| **Requirements** | Actor provides profile information | |
| **Pre-conditions** | Actor has logged in. | |
| **Post-conditions** | Success: Profile’s information of this user has been updated.  Fail: Refill information | |
| **Basic flow** | Actor Actions:  1. Actor clicks ‘My Profile’ button in Actor’s Home page menu.  3. Actor updates information and click ‘Save’ button. [Alternative 1] | System Responses:  2. System redirects to Update Profile page and displays Update Profile form with the following control :  - ‘UserName’ text field (read only)  - ‘Name’ text field  - ‘Birthday’ text field  - ‘Gender’ checkbox group  - ‘Address’ text field  - ‘Phone’ text field  - ‘Email’ text field  - ‘Save’ button  - ‘Back to Home’ button  4. System checks the information  [Exception 1]  5. System update the account into database and then redirects to Actor Home page. |
| **Alternative flow** | Actor Actions:  [Alternative 1]  1. Actor click ‘Back to Home’ button. | System Responses:  2. System returns to Home page. |
| **Exception** | Actor Actions:  [Exception 1]  1. Actor lefts ‘Name’ text field blank.  2. Actor lefts ‘Birthday’ datebox blank.  3. Actor lefts ‘Address’ text field blank.  4. Actor lefts ‘Phone’ text field blank.  5. Input phone not in numeric or phone length not in range 10 – 11 character.  6. Actor lefts ‘Email’ text field blank or input an invalid email address. | System Responses:  System displays message to Actor :  - ‘Name cannot be blank’.  - ‘Birthday is invalid’.  - ‘Address cannot be blank’.  - ‘Phone is invalid’.  - ‘The phone number should be numeric only and must be between 10 and 11 digits’.  - ‘Email is invalid’. |



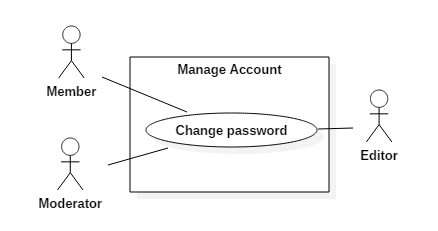


**8.2.4. Forgot password**

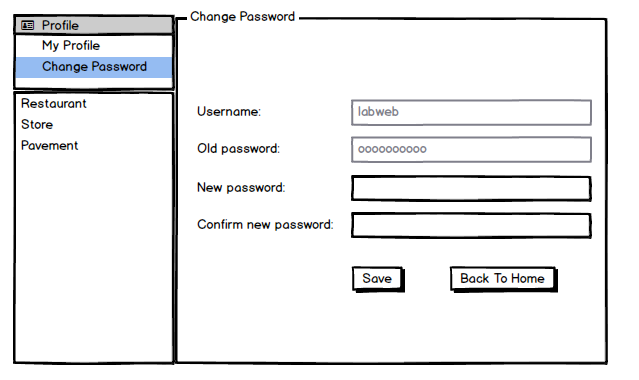
|  |  |  |
| --- | --- | --- |
| **Author** | Dao Nhat Minh | |
| **Use Case Name** | Forgot password | |
| **Actors** | Member or Moderator or Editor | |
| **Description** | Member or Moderator or Editor wants to get back his/her password when forgotten. | |
| **Requirements** | Actor provides username | |
| **Pre-conditions** |  | |
| **Post-conditions** | Success: Password will be send to email of this account.  Fail: Refill username. | |
| **Basic flow** | Actor Actions:  1. Actor clicks ‘Forgot password’ button in ‘Login’ Page  3. Actor insert username and click ‘Reset password’ button. | System Responses:  2. System redirects to Forgot password form and displays the form with the following control :  - ‘UserName’ text field  - ‘Reset password’ button  4. System checks the email of this user base on username that actor input and send password to that email address which user register in database. [Exception 1] |
| **Alternative flow** |  |  |
| **Exception** | Actor Actions:  1. Actor left ‘Username’ text field blank  2. Actor input username haven’t registered yet. | System Responses:  1. ‘Username can’t be blank’  2. ‘Username haven’t registered yet.’ |

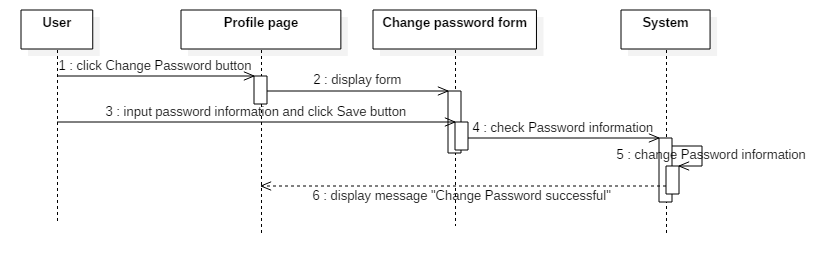


**8.3. Change password**

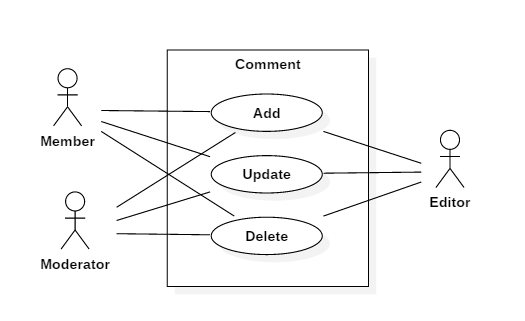
****

|  |  |  |
| --- | --- | --- |
| **Author** | Nguyen Hoang Minh | |
| **Use Case Name** | Change Password | |
| **Actors** | Member or Moderator or Editor | |
| **Description** | Member or Moderator or Editor wants to change his/her password. | |
| **Requirements** | Actor provides password | |
| **Pre-conditions** | Actor has logged in.  Actor has in Profile page. | |
| **Post-conditions** | Success: Password of this user has been changed.  Fail: Refill password, new password and confirm password. | |
| **Basic flow** | Actor Actions:  1. Actor clicks ‘Change password’ button in Profile menu.  3. Actor change password and click ‘Save’ button. [Alternative 1] | System Responses:  2. System redirects to Change password form and displays the form with the following control :  - ‘UserName’ text field (read only)  - ‘Old Password’ password  - ‘New Password’ password  - ‘New Password Confirm’ password  - ‘Save’ button  - ‘Back to Home’ button  4. System checks the password.  [Exception 1]  5. System update password into database and then redirects to Profile page. |
| **Alternative flow** | Actor Actions:  [Alternative 1]  1. Actor click ‘Back to Home’ button. | System Responses:  2. System returns to Home page. |
| **Exception** | Actor Actions:  [Exception 1]  1. Actor lefts ‘New Password’ text field blank.  2. Actor lefts ‘New Confirm Password’ text field blank.  3. New Password and New Password Confirm don’t match. | System Responses:  System displays message to Actor :  - ‘Password cannot be blank’’.  - ‘Confirm Password can’t be blank’.  - ‘Confirm password not match Password.’ |



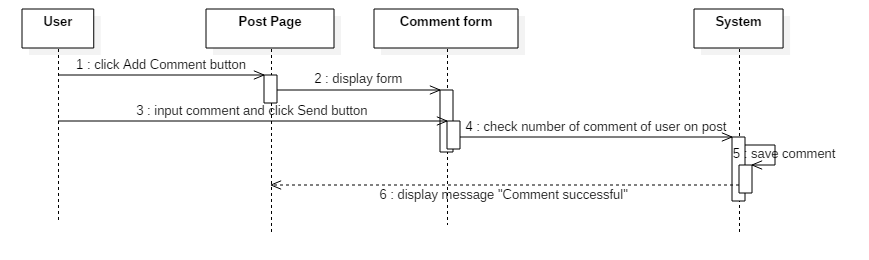


**8.4. Add, Update, Delete comment**

****

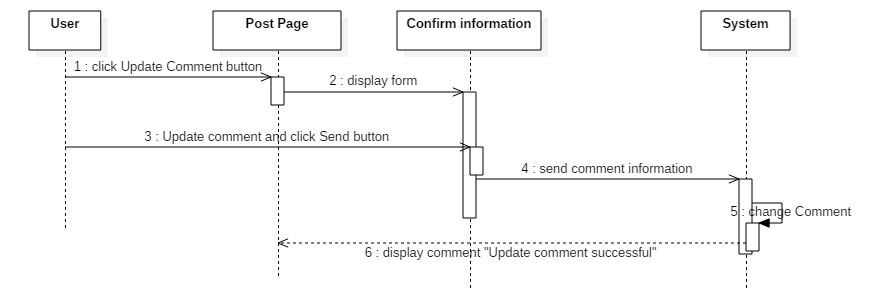
**8.4.1. Add Comment**

|  |  |  |
| --- | --- | --- |
| **Author** | Nguyen Hoang Minh | |
| **Use Case Name** | Add Comment | |
| **Actors** | Member or Moderator or Editor | |
| **Description** | Member or Moderator or Editor wants to Add a comment on the Post. | |
| **Requirements** | Must have the Post, user must in the post to comment. | |
| **Pre-conditions** | Actor has logged in.  Actor must in the post. | |
| **Post-conditions** | Success: Comment has been posted. | |
| **Basic flow** | Actor Actions:  1. Actor click ‘Add Comment’ button to comment on the post.  3. Actor comment on the text area and click ‘Send’ button.[Alternative 1] | System Responses:  2. System allow to comment on the post in the form with the following control:  -’ Comment’ text area.  -‘Send’ button.  - ‘Cancel’ button.  4. System check the number of comment on the post.  [Exception 1]  5. System save comment, username, time of comment take on the system into database and then redirect to the current Post. |
| **Alternative flow** | Actor Actions:  [Alternative 1]  1. Actor click ‘Cancel’ button. | System Responses:  2. System returns to the current Post. |
| **Exception** | Actor Actions:   1. Actor left ‘Comment’ text area blank. | System Responses:   1. “Comment can’t be blank” |



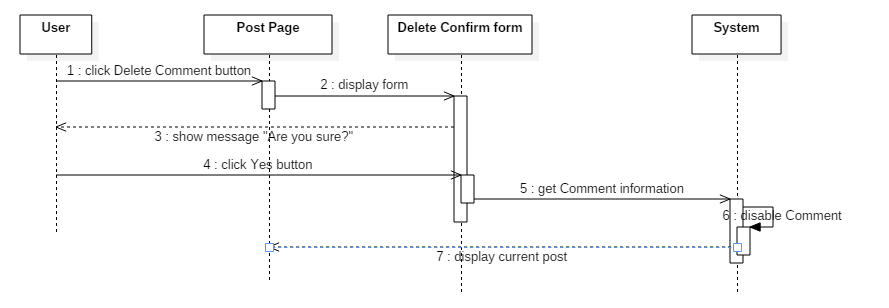
**8.4.2. Update Comment**

|  |  |  |
| --- | --- | --- |
| **Author** | Nguyen Hoang Minh | |
| **Use Case Name** | Update Comment | |
| **Actors** | Member or Moderator or Editor | |
| **Description** | Member or Moderator or Editor wants to Update a comment on the Post. | |
| **Requirements** | Must have the Post to comment.  Must have the comment on the post. | |
| **Pre-conditions** | Actor has logged in.  The Post must publish.  Actor must have a comment on the post. | |
| **Post-conditions** | Success: Comment has been updated. | |
| **Basic flow** | Actor Actions:  1. Actor click ‘Update Comment’ button on the comment.  3. Actor update comment on the text area and click ‘Send’ button.[Alternative 1] | System Responses:  2. System allow to update the comment with the following control:  -’ Comment’ text area.  -‘Send’ button.  - ‘Cancel’ button.  4. System save update comment, username, time of update take on the system into database and then redirect to the current Post. |
| **Alternative flow** | Actor Actions:  [Alternative 1]  1. Actor click ‘Cancel’ button. | System Responses:  2. System returns to the current Post. |
| **Exception** | Actor Actions:   1. Actor left ‘Comment’ text area blank | System Responses:   1. “Comment can’t be blank” |

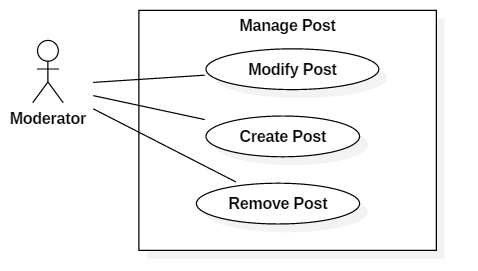
****

**8.4.3. Delete Comment**

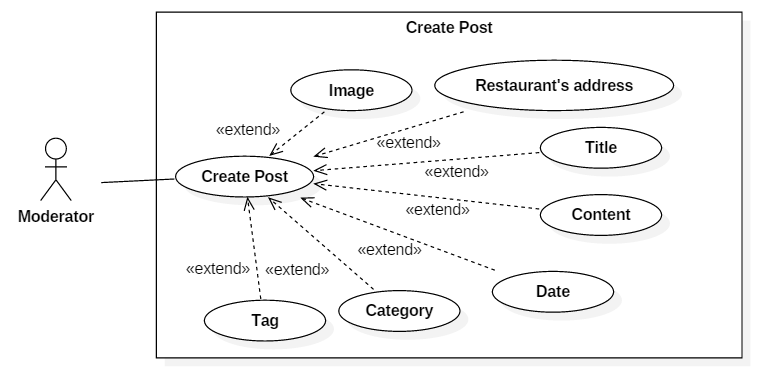
|  |  |  |
| --- | --- | --- |
| **Author** | Nguyen Hoang Minh | |
| **Use Case Name** | Delete Comment | |
| **Actors** | Member or Moderator or Editor | |
| **Description** | Member or Moderator or Editor wants to Update a comment on the Post. | |
| **Requirements** | Must have the Post to comment.  Must have the comment on the post. | |
| **Pre-conditions** | Actor has logged in.  Actor must in the post.  Actor must have a comment on the post. | |
| **Post-conditions** | Success: Comment has been Delete. | |
| **Basic flow** | Actor Actions:  1. Actor click on the ‘Delete’ button on the comment.  3. Actor click ‘Yes’ button. [Alternative 1] | System Responses:  2. System redirects the Delete confirm form.  4. System delete the comment on the database. Then return to the current Post. |
| **Alternative flow** | Actor Actions:  [Alternative 1]  1. Actor click ‘No’ button. | System Responses:  2. System returns to the current Post. |
| **Exception** |  |  |



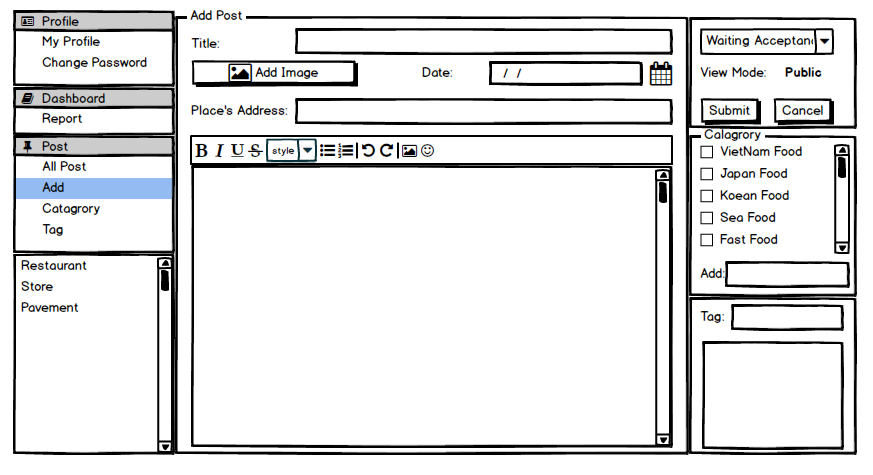
**8.5. Moderator**

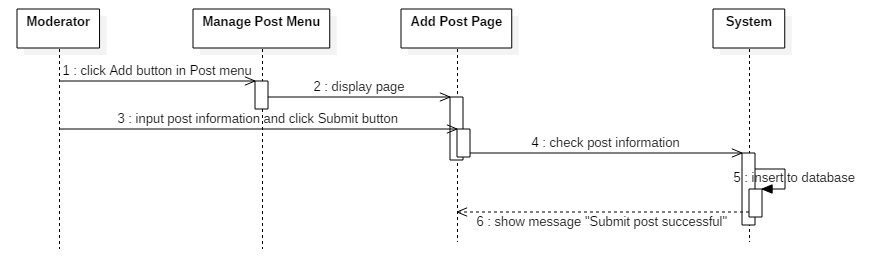


**8.5.1. Create Post**



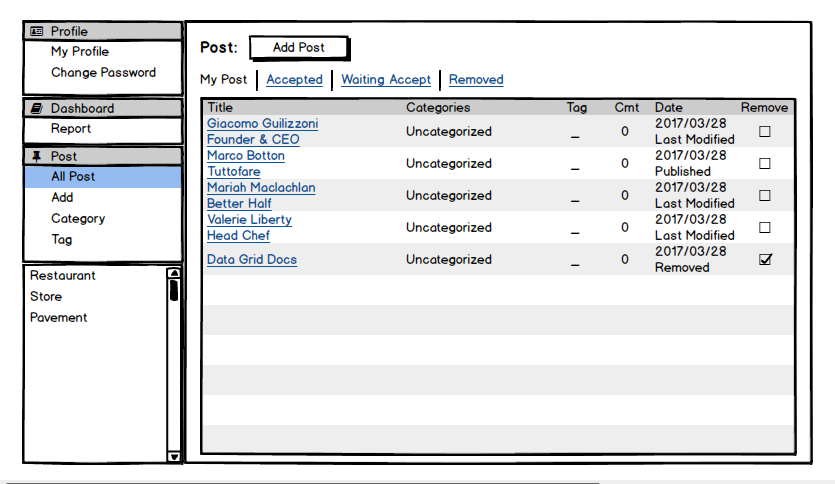
|  |  |  |
| --- | --- | --- |
| **Author** | Hoang Yen Minh | |
| **Use Case Name** | Create Post | |
| **Actors** | Moderator | |
| **Description** | Moderator want to submit a post to Editor and waiting for accepting | |
| **Requirements** | Actor provides Post information and submit Post. | |
| **Pre-conditions** | Actor has logged in as moderator | |
| **Post-conditions** | Success: Moderator’s posts must be accepted by Editor  Fail: Refill information or Editor did not accept | |
| **Basic flow** | Actor Actions:  1. Actor clicks Add button in Post on Moderator menu.  3. Actor input information then click ‘Submit’ button [Alternative 1] | System Response:  2. System displays Add Post page. System displays form for Actor to input information with the following control:  - Button ‘Browse’ to upload image  - ‘Title’ text field  - ‘Content’ text area  - ‘Place’s address’ text field  - ‘Phone’ text field  - ‘Name’ text field  - ‘Date’ date  - ‘Category’ checkbox group  - ‘Tag’ hyperlink  - ‘Tool Box’ text area  - ‘Submit’ button  - ‘Cancel’ button  4. System check information [Exception 1]  5. System inserts the account into database and waits for Editor to accept and back to Manage Post page |
| **Alternative flow** | [Alternative 2]  1. Actor click ‘Back to Home’ button. | 2. System returns Manage Post page. |
| **Exceptions** | Actor Actions:  [Exception 1]  - Actor left ‘Title’ field blank  - Actor left ‘Content’ field blank  - Actor left ‘Place’s address’ field blank  - Actor left ‘Name’ field blank  - Actor left ‘Category’ in checkboxgroup not choose  -Actor clicks ‘Upload’ button but upload process fail because some unknown problem. | System Response: System display message to Actor:  - ‘The title can not be blank’  - ‘The content can not be blank’  - ‘The content can not be blank’  - ‘Place’s address can not blank’  - ‘Name can not blank’  - ‘Category must be choose’  - ‘Uploading image has failed, Please try again’ |

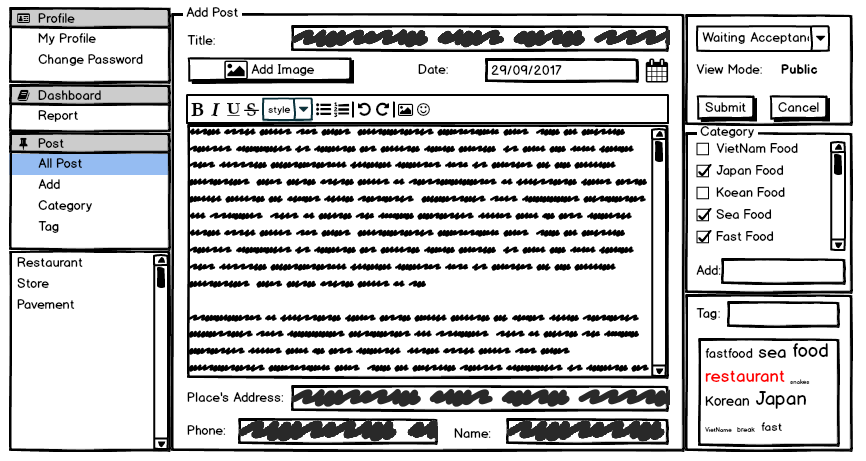
****



**8.5.2. Modify Post**

|  |  |  |
| --- | --- | --- |
| **Author** | Hoang Yen Minh | |
| **Use Case Name** | Modify Post | |
| **Actors** | Moderator | |
| **Description** | Moderator wants to Update his/her post’s information. | |
| **Requirements** | Actor provides post information | |
| **Pre-conditions** | Actor has logged in as Moderator.  Modify post must be created by this moderator | |
| **Post-conditions** | Success: Profile’s information of this user has been updated.  Fail: Refill information | |
| **Basic flow** | Actor Actions:  1. Actor clicks ‘All Post’ sub menu in Actor’s Home page menu.  3. Actor clicks ‘Title’ hyperlink to view detail                      5. Actor modifies information and click ‘Submit’ button. [Alternative 1] | System Responses:  2. System redirects to Manage Post page and displays all post wrote by this Moderator.  4. System redirects to View Detail Page and loaded an information of the post:  - Button ‘Browse’ to upload image  - ‘Title’ text field  - ‘Content’ text area  - ‘Place’s address’ text field  - ‘Name’ text field  - ‘Phone’ text field  - ‘Date’ datebox  - ‘Category’ checkboxgroup  - ‘Tag’ hyperlink  - ‘ToolBox’ text area  - ‘Submit’ button  - ‘Cancel’ button  6. System checks the information  [Exception 1]  7. System update the post into database and then redirects to Actor Home page. |
| **Alternative flow** | Actor Actions:  [Alternative 1]  1. Actor click ‘Back to Home’ button. | System Responses:    2. System returns to Home page. |
| **Exception** | Actor Actions:  [Exception 1]  1. Actor left ‘Title’ field blank  2. Actor left ‘Content’ field blank  3. Actor left ‘Place’s address’ field blank  4. Actor left ‘Category’ in checkboxgroup not choose  5. Actor clicks ‘Upload’ button but upload process fail because some unknown problem. | System Response: System display message to Actor:  - ‘The title can not be blank’  - ‘The content can not be blank’  - ‘The content can not be blank’  - ‘Place’s address can not blank’  - ‘Category must be choose’  - ‘Uploading image has failed, Please try again’ |

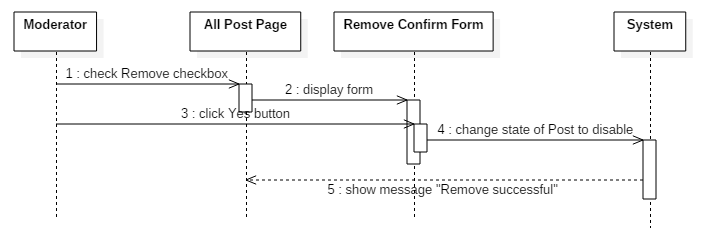
****

****

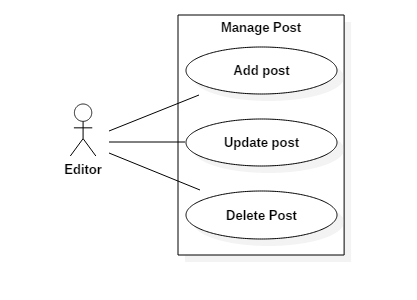


**8.5.3. Remove Post**

|  |  |  |
| --- | --- | --- |
| **Author** | Hoang Yen Minh | |
| **Use Case Name** | Remove Post | |
| **Actors** | Moderator | |
| **Description** | Moderator wants to Remove a post. | |
| **Requirements** | Must have the published Post. | |
| **Pre-conditions** | Actor has logged in as Moderator.  Moderator who wrote that post.  Post was published. | |
| **Post-conditions** | Success: Post has been disable | |
| **Basic flow** | Actor Actions:  1. Actor checks on the ‘Remove’ checkbox on the All Post Page.  3. Actor click ‘Yes’ button. [Alternative 1] | System Responses:  2. System redirects the Remove confirm form.  4. System change state of post to wait for Editor accept. Then return to the Manage Post page. |
| **Alternative flow** | Actor Actions:  [Alternative 1]  1. Actor click ‘No’ button. | System Responses:  2. System returns to the current Post. |
| **Exception** |  |  |

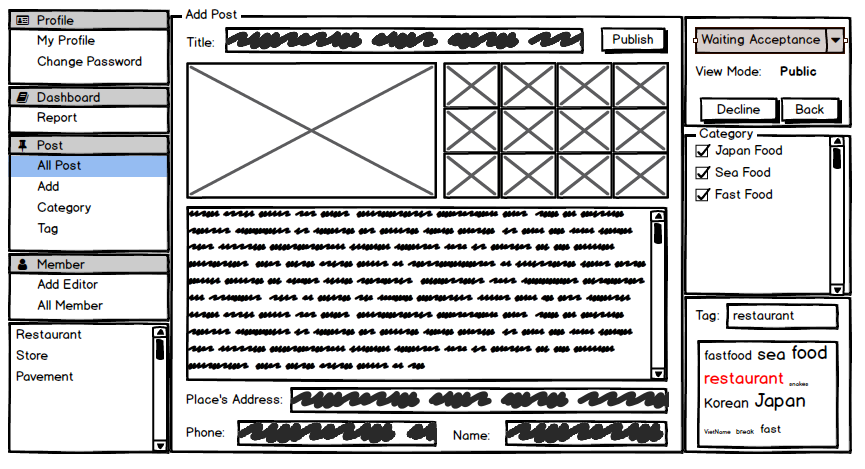


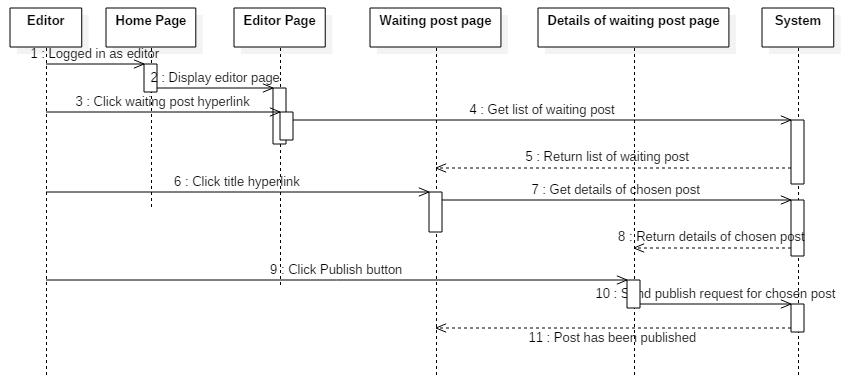
**8.6. Editor**



**8.6.1. Accept Post**

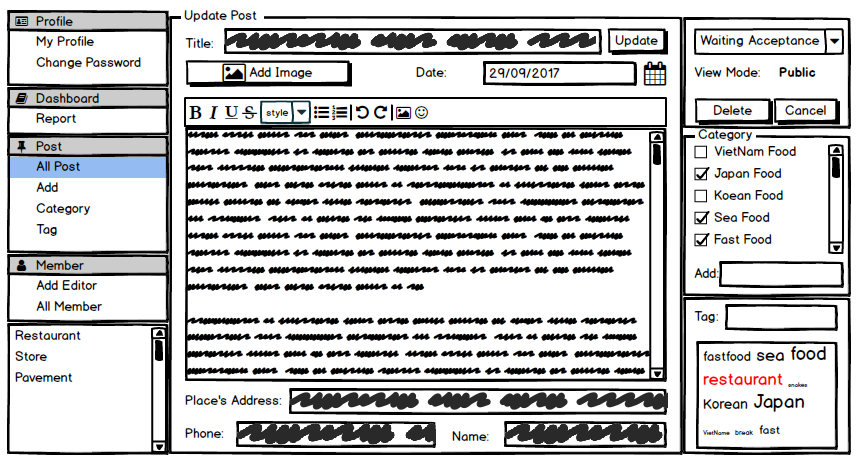
|  |  |  |
| --- | --- | --- |
| **Author** | Dao Nhat Minh | |
| **Use case name** | Accept post | |
| **Actors** | Editor | |
| **Description** | When moderator send a post to web site, that post will wait until editor accept to publish it. | |
| **Requirements** | Accept post were sent from moderator | |
| **Pre-conditions** | Must logged in as editor and the post is already exist | |
| **Post-conditions** | Success: accepted post will be show on website.  Fail: Post will be disabled from waiting post page. | |
| **Basic flow** | Actor actions:  1.From Home page, actor clicks login as Editor.  3.Actor click ‘All Post’ button in the Home Page menu.  5. Actor click ‘Waiting Accept’ hyperlink in the Post page menu.  7.Actor click ‘Title’ hyperlink to view detail  9.Actor click ‘Publish’ button | System response:  2. System check role and turn back to Home Page.  4.System redirect to All Post page.  6.System redirects to Waiting Post page and display list of all post in table form:   * ‘No.’ * ‘Author.’ * ‘Date ‘ * ‘Categories ‘ * ‘Tag’ * ‘Title’   8.System redirects to Detail Of Waiting Post Page and display:   * ‘Title’ * ‘Place’s Address’ * ‘Phone’ * ‘Name’ * ‘Content’ * ‘Image’ * ‘Tags’ * ‘Category’ * ‘Publish’ button * ‘Decline’ button * ‘Back’ button   10.System change state of post from ‘waiting’ to ‘accepted’ in database, publish on website and show message ‘Post published successfully’ and return to Waiting Post Page. |
| **Alternative flow** | Actor Actions:  [Alternative 1]   1. Actor click ‘Decline’ button.   [Alternative 2]  1. Actor click ‘Back’ button. | System Responses:   1. System returns to Waiting Post Page and that post will be disabled and show message ‘Post declined successfully’.   System Responses:  2. System returns to Waiting Post page. |
| **Exception** | 1. Publish but not show on web 2. Decline but not disable. | 1. Unknown error, please try again. 2. Unknown error, please try again. |

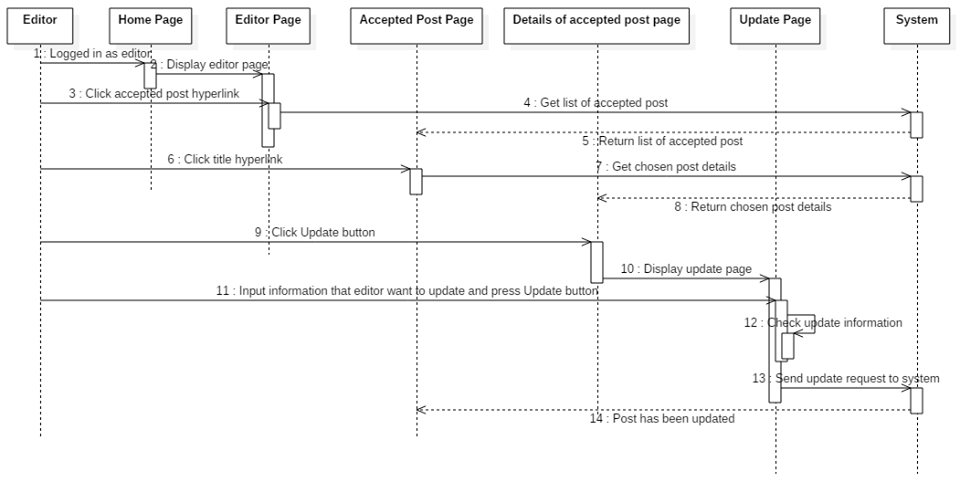
****

****

**8.6.2. Update Post**

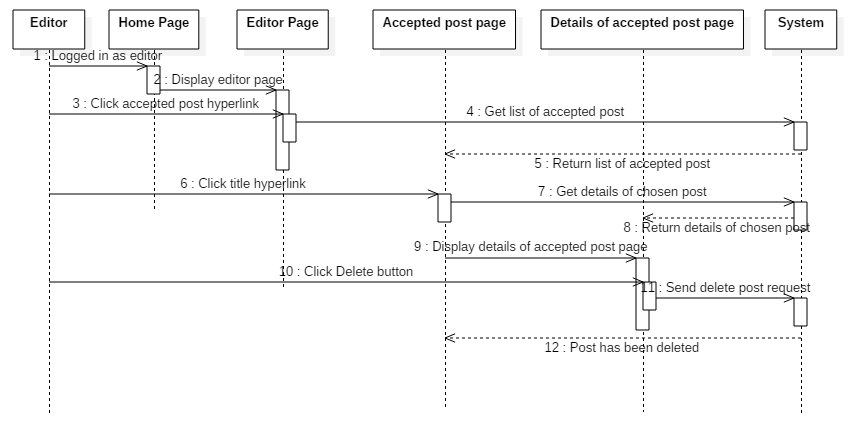
|  |  |  |
| --- | --- | --- |
| **Author** | Dao Nhat Minh | |
| **Use case name** | Update post | |
| **Actors** | Editor | |
| **Description** | When moderator send a modify post, editor can view, update and post it to website | |
| **Requirements** | Update a post by editor | |
| **Pre-conditions** | Must logged in as editor and the post is already exist | |
| **Post-conditions** | Success: post has been updated.  Fail: post can’t update. | |
| **Basic flow** | Actor actions:  1.From Home Page, actor clicks login as Editor.  3.Actor click to go Editor Page. In Editor Page, actor click ‘View Accepted Post’ hyperlink  5.Actor click ‘Title’ hyperlink to view detail  7.Actor click ‘Update’ button  9.Actor click ‘Update’ button | System response:  2.System check role and turn back to Home Page.  4.System redirects to Accepted Post List Page and display list of all post in table form:   * ‘No.’ * ‘Author.’ * ‘Date’ * ‘Categories’ * ‘Tag ‘ * ‘Title’   6.System redirects to View Detail Of Accepted Post Page and display list of all post in following control:   * Title * Address * Content * Image * Category * Tag * Title * ‘Update’ button * ‘Delete’ button * ‘Back’ button   8.System redirects to Update page and display list of all post in following control   * ‘Title’ text field * ‘Place’s Address’ text field * ‘Content’ text area * ‘Image’ button * ‘Category’ checkbox group * ‘Tag’ hyperlink * ‘Update’ button * ‘Back’ button   10.System will save new information of the post, show message ‘Update successfully’ and return to Post List Page. |
| **Alternative flow** | Actor Actions:  [Alternative 1]  1. Actor click ‘Back’ button. | System Responses:    2. System returns to View Detail Page. |
| **Exception** | Exception:  1. Actor lefts ‘Title’ text field blank.  2. Actor lefts ‘Content’ text area blank.  3. Actor lefts ‘Address’ text field blank.  4. Actor don’t upload any image  5. Actor lefts ‘Category’ in checkbox group not choose  6. Actor click ‘Upload’ but upload image process fail because some unknown problem. | System response:   1. ’Title can’t be blank’. 2. ’Content can’t be blank’. 3. ’Please choose restaurant’s location’. 4. ‘Please add some image’. 5. ’Please choose restaurant’s tag’. 6. ‘Upload image has failed, Please try again’ |

****

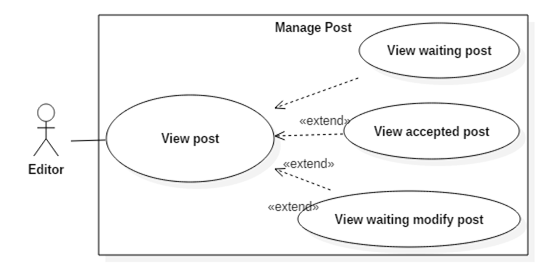
****

**8.6.3. Delete Post**

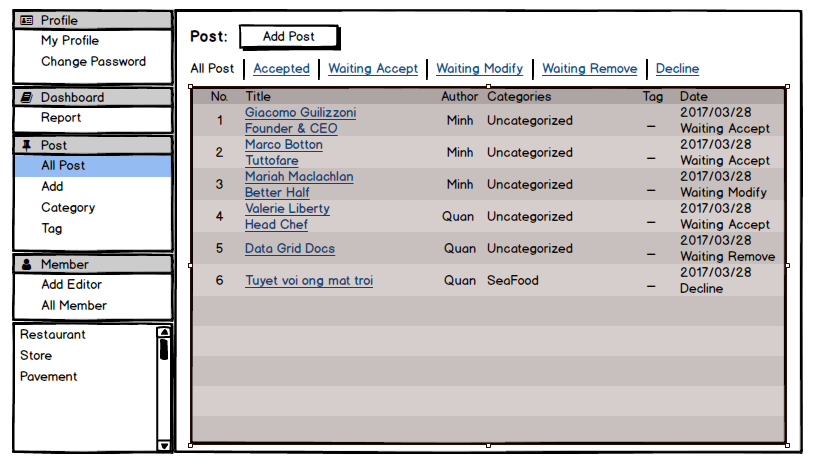
|  |  |  |
| --- | --- | --- |
| **Author** | Dao Nhat Minh | |
| **Use case name** | Delete post | |
| **Actors** | Editor | |
| **Description** | When editor want to delete a post. | |
| **Requirements** | Disable a post | |
| **Pre-conditions** | Must logged in as editor and the post is already exist | |
| **Post-conditions** | Success: post has been disabled.  Fail: post hasn’t been disabled. | |
| **Basic flow** | Actor actions:  1.From Home Page, actor clicks login as Editor.  3.Actor click to go Editor Page. In Editor Page, actor click ‘View Accepted Post’ hyperlink  5.Actor click ‘Title’ hyperlink to view detail  7.Actor click ‘Delete’ button [Alternative 1] | System response:  2. System check role and turn back to Home Page.  4.System redirects to Accepted Post List Page and display list of all post in table form:   * No. * Author. * Date create * Categories * Tag * Title   6.System redirects to View Detail of Accepted Page and display list of details in table form:   * Title * Address * Content * Image * Category * Tag * Title * ‘Update’ button * ‘Delete’ button * ‘Back’ button   8.System will disable the post, show message ‘Delete successfully’ and return to Post List Page. |
| **Alternative flow** | Actor Actions:  [Alternative 1]  1. Actor click ‘Back’ button. | System Responses:    2. System returns to Post List Page. |
| **Exception** |  |  |

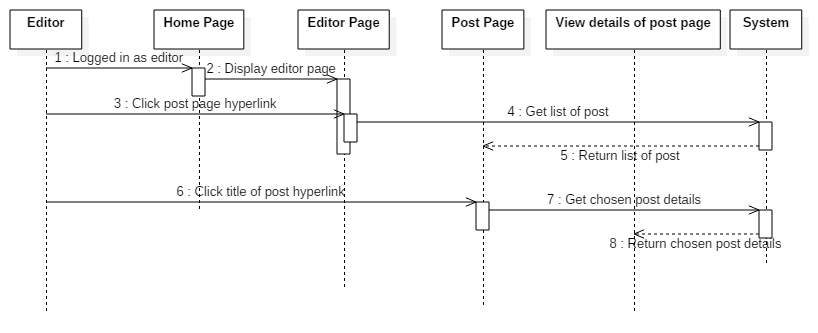
****

**8.7. View post**

****

|  |  |  |
| --- | --- | --- |
| **Author** | Dao Nhat Minh | |
| **Use case name** | View post | |
| **Actors** | Editor | |
| **Description** | Editor can view every post in the website | |
| **Requirements** | View post were sent from moderator | |
| **Pre-conditions** | Must logged in as editor and the post is already exist | |
| **Post-conditions** | Success: view list of post and details  Fail: Can’t view post | |
| **Basic flow** | Actor actions:  1.From Home page, actor clicks login as Editor.  3.Actor click to go Editor Page. In Editor Page, actor click ‘Post’ hyperlink  5.Actor click ‘Title’ hyperlink to view details of that post | System response:  2. System check role and turn back to Home Page.  4.System redirects to Post page and display list of all post in table form:   * No. * Author. * Date create * Categories * Tag * Title   6.System redirects to View Detail Of Post Page and display list of details in table form:   * Title * Address * Content * Image * Tags * Category * Comment * ‘Back’ button |
| **Alternative flow** |  |  |
| **Exception** | 1.Can’t view | 1.’Unknown error, please try again.’ |





**8.7.1. View waiting Post**

|  |  |  |
| --- | --- | --- |
| **Author** | Dao Nhat Minh | |
| **Use case name** | View waiting post | |
| **Actors** | Editor | |
| **Description** | When moderator send a post, editor can view it | |
| **Requirements** | View post were sent from moderator | |
| **Pre-conditions** | Must logged in as editor and the post is already exist | |
| **Post-conditions** | Success: view list of post and details  Fail: Can’t view post | |
| **Basic flow** | Actor actions:  1.From Home page, actor clicks login as Editor.  3.Actor click to go Editor Page. In Editor Page, actor click ‘Waiting Post’ hyperlink  5.Actor click ‘Title’ hyperlink to view details of that waiting post | System response:  2. System check role and turn back to Home Page.  4.System redirects to Post page and display list of all waiting post in table form:   * No. * Author. * Date create * Categories * Tag * Title   6.System redirects to View Detail Of Post Page and display list of details in table form:   * Title * Address * Content * Image * Tags * Category * Comment * ‘Back’ button |
| **Alternative flow** |  |  |
| **Exception** | 1.Can’t view | 1.’Unknown error, please try again.’ |

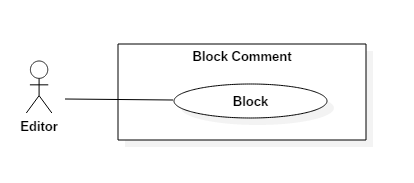
**8.7.2. View accepted Post**

|  |  |  |
| --- | --- | --- |
| **Author** | Dao Nhat Minh | |
| **Use case name** | View accepted post | |
| **Actors** | Editor | |
| **Description** | Editor can view every post which is accepted and published on the website | |
| **Requirements** | View accepted post that published on the website | |
| **Pre-conditions** | Must logged in as editor and the post is already exist | |
| **Post-conditions** | Success: view list of accepted post and details  Fail: Can’t view post | |
| **Basic flow** | Actor actions:  1.From Home page, actor clicks login as Editor.  3.Actor click to go Editor Page. In Editor Page, actor click ‘Accepted Post’ hyperlink  5.Actor click ‘Title’ hyperlink to view details of that accepted post | System response:  2. System check role and turn back to Home Page.  4.System redirects to Post page and display list of all accepted post in table form:   * No. * Author. * Date create * Categories * Tag * Title   6.System redirects to View Detail Of Post Page and display list of details in table form:   * Title * Address * Content * Image * Tags * Category * Comment * ‘Back’ button |
| **Alternative flow** |  |  |
| **Exception** | 1.Can not view | 1.’Unknown error, please try again.’ |

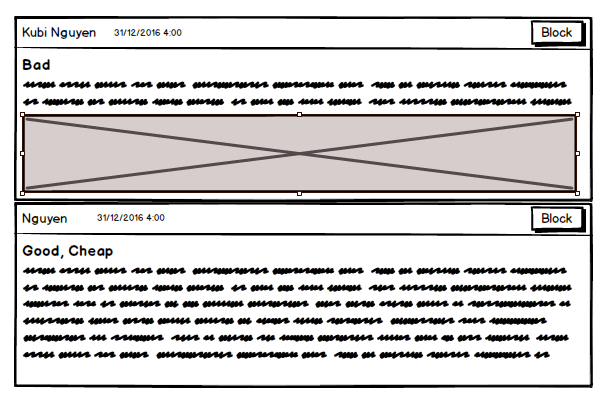
**8.7.3. View waiting modify Post**

|  |  |  |
| --- | --- | --- |
| **Author** | Dao Nhat Minh | |
| **Use case name** | View waiting modify post | |
| **Actors** | Editor | |
| **Description** | When moderator send a modify post, editor can view it | |
| **Requirements** | View modify post were sent from moderator | |
| **Pre-conditions** | Must logged in as editor and the post is already exist | |
| **Post-conditions** | Success: view list of modify post and details  Fail: Can’t view post | |
| **Basic flow** | Actor actions:  1.From Home page, actor clicks login as Editor.  3.Actor click to go Editor Page. In Editor Page, actor click ‘Waiting modify Post’ hyperlink  5.Actor click ‘Title’ hyperlink to view details of that waiting post | System response:  2. System check role and turn back to Home Page.  4.System redirects to Post page and display list of all waiting modify post in table form:   * No. * Author. * Date create * Categories * Tag * Title   6.System redirects to View Detail Of Post Page and display list of details in table form:   * Title * Address * Content * Image * Tags * Category * Comment * ‘Back’ button |
| **Alternative flow** |  |  |
| **Exception** | 1.Can not view | 1.’Unknown error, please try again.’ |

**8.8. Block comment**

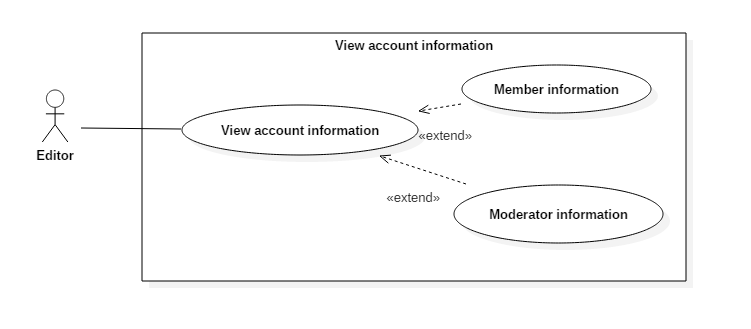
****

|  |  |  |
| --- | --- | --- |
| **Author** | Dao Nhat Minh | |
| **Use case name** | Block comment | |
| **Actors** | Editor | |
| **Description** | When a comment is not conformed to the post, editor can block it. | |
| **Requirements** | A comment not comfortable to block. | |
| **Pre-conditions** | Must logged in as editor and the post, not conformed comment is existed. | |
| **Post-conditions** | Success: Comment has been disabled. | |
| **Basic flow** | Actor actions:  1.From Home page, actor clicks login as Editor.  3.At Home page, actor clicks to any ‘Title’ hyperlink  5.Acotr click on any comment that not conformed and choose ‘Block’ button | System response:  2. System check role and turn back to Home Page.  4.System redirects to View Detail Of Post Page and display list of details and list of Comment:   * ‘Username’ * ‘Date Comment’ * ‘Title’ * ‘Content’ * ‘Image’ * ‘Block’ button   6.Systems will disable the comment, show message ‘Comment block successfully’. |
| **Alternative flow** | Actor Actions: | System Responses: |
| **Exception** | 1. Blocked but that comment hasn’t been disabled | 1. ‘Unknown error, please try again.’ |

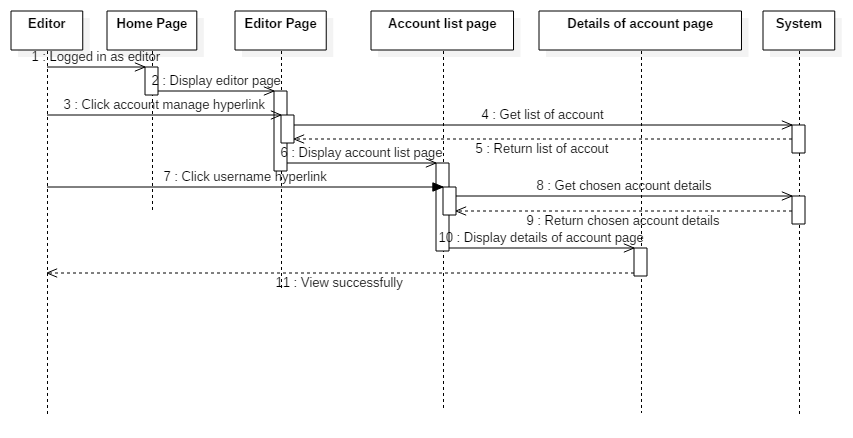
****

****

**8.9. View account information**

****

|  |  |  |
| --- | --- | --- |
| **Author** | Dao Nhat Minh | |
| **Use case name** | View account information | |
| **Actors** | Editor | |
| **Description** | Editor can view all information of member or moderator | |
| **Requirements** | View information of member and moderator | |
| **Pre-conditions** | Must logged in as editor and the account editor want to view must exist | |
| **Post-conditions** | Success: View success.  Fail: Can’t view. | |
| **Basic flow** | Actor actions:  1.From Home page, actor clicks login as Editor.  3.At Home page, actor clicks to ‘Account manage’ hyperlink  5.At Account List page, actor clicks to ‘View’ button | System response:  2. System check role and turn back to Home Page.  4.System redirects to Account List Page and display list of details in table form:   * No * Username * Sex * ‘View’ button * ‘Back’ button   4.System redirects to Details Of Account Page and display list of details in table form:   * Username * Name * Birthday * Sex * Address * Phone * Email * ‘Back’ button |
| **Alternative flow** |  |  |
| **Exception** |  |  |



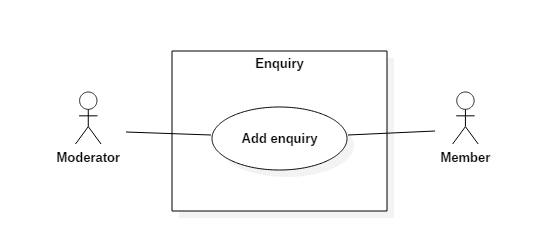
**8.9.1. View member account information**

|  |  |  |
| --- | --- | --- |
| **Author** | Dao Nhat Minh | |
| **Use case name** | View account information | |
| **Actors** | Editor | |
| **Description** | Editor can view all information of member | |
| **Requirements** | View information of member | |
| **Pre-conditions** | Must logged in as editor and the account editor want to view must exist | |
| **Post-conditions** | Success: View success.  Fail: Can’t view. | |
| **Basic flow** | Actor actions:  1.From Home page, actor clicks login as Editor.  3.At Home page, actor clicks to ‘Member account manage’ hyperlink  5.At Member account List page, actor clicks to ‘View’ button | System response:  2. System check role and turn back to Home Page.  4.System redirects to Member Account List Page and display list of details in table form:   * No * Username * Sex * ‘View’ button * ‘Back’ button   6.System redirects to Details Of Member Account Page and display list of details in table form:   * Username * Name * Birthday * Sex * Address * Phone * Email * ‘Back’ button |
| **Alternative flow** |  |  |
| **Exception** |  |  |

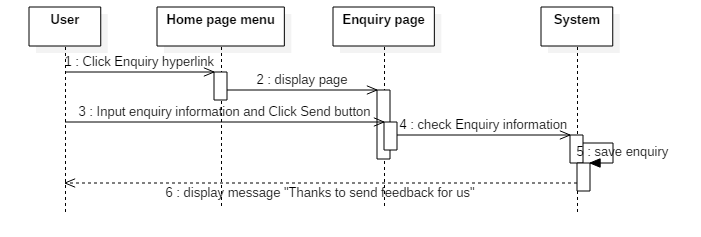
**8.9.2. View moderator account information**

|  |  |  |
| --- | --- | --- |
| **Author** | Dao Nhat Minh | |
| **Use case name** | View account information | |
| **Actors** | Editor | |
| **Description** | Editor can view all information of moderator | |
| **Requirements** | View information of moderator | |
| **Pre-conditions** | Must logged in as editor and the account editor want to view must exist | |
| **Post-conditions** | Success: View success.  Fail: Can’t view. | |
| **Basic flow** | Actor actions:  1.From Home page, actor clicks login as Editor.  3.At Home page, actor clicks to ‘Moderator Account manage’ hyperlink  5.At Moderator Account List page, actor clicks to ‘View’ button | System response:  2. System check role and turn back to Home Page.  4.System redirects to Moderator Account List Page and display list of details in table form:   * No * Username * Sex * ‘View’ button * ‘Back’ button   6.System redirects to Details Of Moderator Account Page and display list of details in table form:   * Username * Name * Birthday * Sex * Address * Phone * Email * ‘Back’ button |
| **Alternative flow** |  |  |
| **Exception** |  |  |

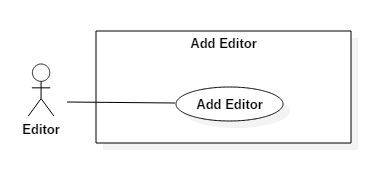
**8.10. Enquiry**

****

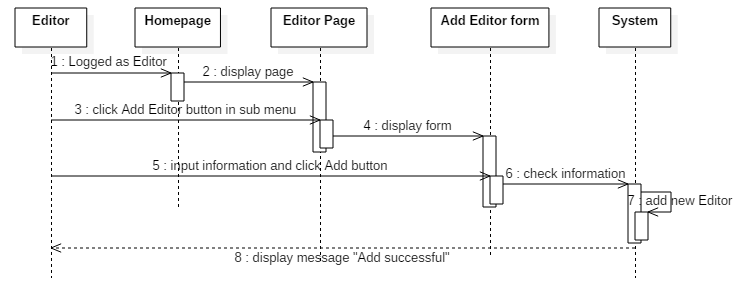
|  |  |  |
| --- | --- | --- |
| **Author** | Nguyen Hoang Minh | |
| **Use Case Name** | Enquiry | |
| **Actors** | Member or Moderator | |
| **Description** | Member or Moderator wants to send Enquiry for Editor. | |
| **Requirements** | Actor must login in system as Member and Moderator. | |
| **Pre-conditions** | User Login in system. | |
| **Post-conditions** | Success: User send Enquiry to website. | |
| **Basic flow** | Actor Actions:  1. Actor go to Home page of website and click ‘Enquiry’ button in submenu.  3. Actor input title and content and click ‘Send’ button.  [Alternative 1] | System Responses:  2. System redirects to Home page and displays Enquiry form with the following control:  - ‘Username’ text field (readonly)  - ‘Title’ text field  - ‘Content’ text arae  - ‘Date of feedback’ datetime  - ‘Send‘ button  - ‘Back’ button  4. System check the information and ‘Date of feedback’ get current day.  [Exception 1]  5. System send feedback to Editor feedback Page and redirect to Actor’s Home page and display message “Send Enquiry successful”. |
| **Alternative flow** | Actor Actions:  [Alternative 1]  1. Actor input title and content and click ‘Back’ button. | System Responses:  2. System return to Home page. |
| **Exception** | Actor Actions:  [Exception 1]   1. Actor left ‘Title’ field blank. 2. Actor left ‘Content’ field blank | System Responses:   1. Title can’t be blank. 2. Content can’t be blank. |



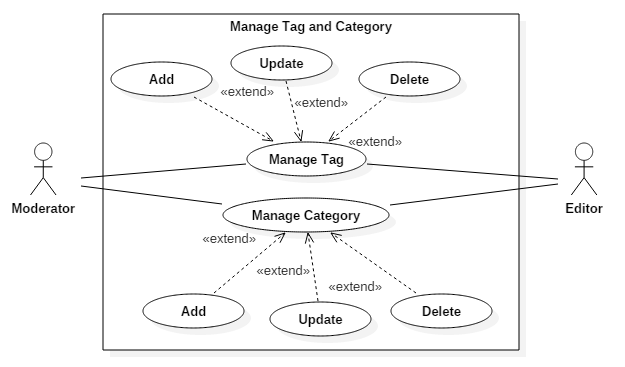
**8.11. Add Editor**

****

|  |  |  |
| --- | --- | --- |
| **Author** | Nguyen Hoang Minh | |
| **Use case name** | Add new Editor | |
| **Actors** | Editor | |
| **Description** | Editor can Add new editor | |
| **Requirements** | Logged in as Editor. | |
| **Pre-conditions** | Must login as editor. | |
| **Post-conditions** | Success: New Editor has been Add to system.  Fail: Can’t Add. | |
| **Basic flow** | Actor actions:  1.From Home page, actor clicks login as Editor.  3.At Home page, actor clicks to ‘Add Editor’ hyperlink  5.Actor input information and click ‘Add’ button.[Alternative 1] | System response:  2. System check role and turn back to Home Page.  4.System redirects to Add Editor form and display list of details in table form:  - ‘UserName’ text field  - ‘Password’ text field  - ‘Confirm Password’ text field  - ‘Name’ text field  - ‘Birthday’ datebox  - ‘Gender’ checkbox group  - ‘Address’ text field  - ‘Phone’ text field  - ‘Email’ text field  - ‘Experience Years‘ text field  - ‘Worked Place’ text field  - ‘Add’ button  - ‘Back to Home’ button  6. System checks the information  [Exception 1]  7. System inserts the account into database, show message: ‘Add new Editor successful’ and then Redirect to Editor Page. |
| **Alternative flow** | Actor actions:  [Alternative 1]  1. Actor click ‘Back to Home’ button | System response:  2. System redirect to Editor Home page. |
| **Exception** | Actor Actions:  [Exception 1]  1. Actor lefts ‘UserName’ text field blank.  2. UserName has already exists.  3. Actor lefts ‘Password’ text field blank.  4. Actor lefts ‘Confirm Password’ text field blank.  5. Password and confirm password don’t match.  6. Actor lefts ‘Name’ text field blank.  7. Actor lefts ‘Birthday’ datebox blank.  8. Actor lefts ‘Address’ text field blank.  9. Actor lefts ‘Phone’ text field blank.  10. Input phone not in numeric or phone length not in range 8 – 11 character.  11. Actor lefts ‘Email’ text field blank or input an invalid email address.  12. Actor lefts ‘Experience Years‘ text field blank  13. Input ‘Experience Years‘ not in numeric length not bigger than age | System Responses:  System displays message to Actor :  - ‘UserName cannot be blank’.  - ‘This Username has already exist!’.  - ‘Password cannot be blank’’.  - ‘Confirm Password can’t be blank’.  - ‘Confirm password not match password.’  - ‘Name cannot be blank’.  - ‘Age must over 16’.  - ‘Address cannot be blank’.  - ‘Phone is invalid’.  - ‘The phone number should be numeric only and must be between 8 and 11 digits’.  - ‘Email is invalid’.  - “Experience Years cannot be blank”  - “Experience Years is invaild” |

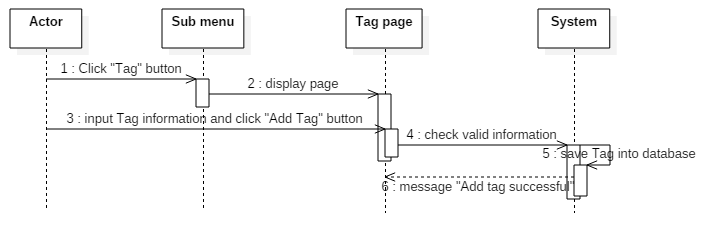


**8.12. Manage Tag, Category**

****

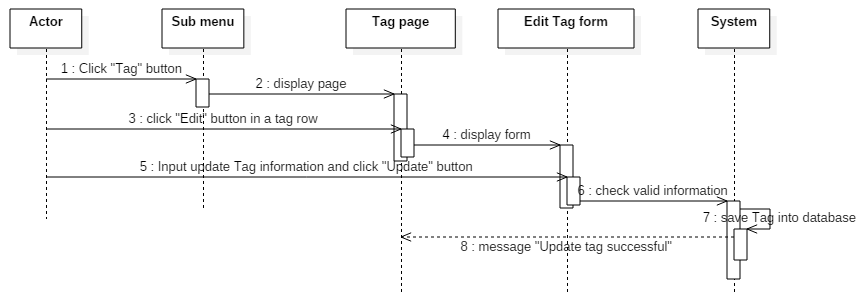
**8.12.1. Add Tag**

|  |  |  |
| --- | --- | --- |
| **Author** | Nguyen Hoang Minh | |
| **Use case name** | Add Tag | |
| **Actors** | Editor, Moderator | |
| **Description** | Editor, Moderator want to create a new tag to use. | |
| **Requirements** |  | |
| **Pre-conditions** | Must logged in as editor or moderator | |
| **Post-conditions** | Success: Tag has been create.. | |
| **Basic flow** | Actor actions:  1.From Home Page, actor click ‘Tag’ in submenu  3.Actor input tag information and click ‘Add Tag’ button | System response:  2. System redirect to tag page, return all tag and Add Tag group with the following control:  - ‘Name’ text field  - ‘Description’ text area  - ‘Add Tag’ button  4.System check add tag information, save into database and return message: “Add tag successful!” in Tag Page |
| **Alternative flow** | Actor Actions: | System Responses: |
| **Exception** | Actor Actions:  1. Actor left ‘Name’ text field blank  2. Actor left ‘Description’ text area blank | System Responses:  1. ‘Name can’t be blank.’  2. ‘Description can’t be blank.’ |

****

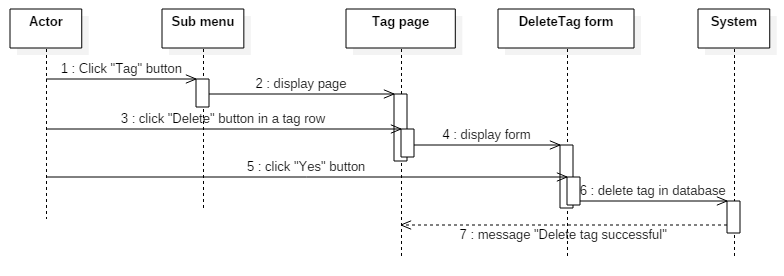
**8.12.2. Update Tag**

|  |  |  |
| --- | --- | --- |
| **Author** | Nguyen Hoang Minh | |
| **Use case name** | Update Tag | |
| **Actors** | Editor, Moderator | |
| **Description** | Editor, Moderator want to update a tag. | |
| **Requirements** | Tag have been create. | |
| **Pre-conditions** | Must logged in as editor or moderator | |
| **Post-conditions** | Success: Tag has been update. | |
| **Basic flow** | Actor actions:  1.From Home Page, actor click ‘Tag’ in submenu  3. Actor click ‘Edit’ button on a tag row which want to update.  5. Actor input update tag information and click ‘Update’ button. | System response:  2. System redirect to tag page, return all tag in a table with the following control:  - ‘Name’  - ‘Author’  - ‘Description’  - ‘Count’  - ‘Edit’ button  - ‘Delete’ button  4. System redirect to an ‘Edit Tag’ form with the following control:  - ‘Name’ text field  - ‘Description’ text area  - ‘Update button  6. System check update tag information, save into database and return message: “Update tag successful!” in Tag Page |
| **Alternative flow** | Actor Actions: | System Responses: |
| **Exception** | Actor Actions:  1. Actor left ‘Name’ text field blank  2. Actor left ‘Description’ text area blank | System Responses:  1. ‘Name can’t be blank.’  2. ‘Description can’t be blank.’ |

****

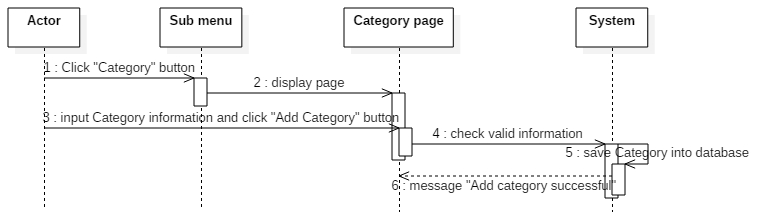
**8.12.23. Delete Tag**

|  |  |  |
| --- | --- | --- |
| **Author** | Nguyen Hoang Minh | |
| **Use case name** | Delete Tag | |
| **Actors** | Editor, Moderator | |
| **Description** | Editor, Moderator want to delete a tag. | |
| **Requirements** | Tag have been create. | |
| **Pre-conditions** | Must logged in as editor or moderator | |
| **Post-conditions** | Success: Tag has been delete. | |
| **Basic flow** | Actor actions:  1.From Home Page, actor click ‘Tag’ in submenu  3. Actor click ‘Delete’ button on a tag row which want to delete.  5. Actor click ‘Yes’ button | System response:  2. System redirect to tag page, return all tag in a table with the following control:  - ‘Name’  - ‘Author’  - ‘Description’  - ‘Count’  - ‘Edit’ button  - ‘Delete’ button  4. System redirect to an ‘Delete Tag’ message: “Are you sure you want to delete this tag?” and control:  - ‘Yes’ button  - ‘No’ button  6. System delete tag in database and return to Tag Page with message “Delete tag successful.” |
| **Alternative flow** | Actor Actions:  Actor click ‘No’ button | System Responses:  System return to Tag Page. |
| **Exception** | Actor Actions: | System Responses |

****

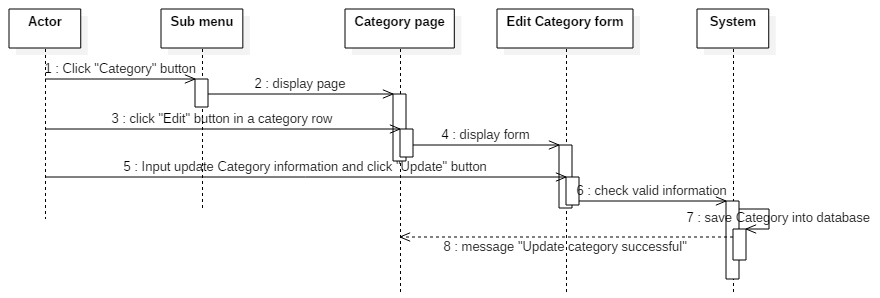
**8.12.4. Add Category**

|  |  |  |
| --- | --- | --- |
| **Author** | Nguyen Hoang Minh | |
| **Use case name** | Add Category | |
| **Actors** | Editor, Moderator | |
| **Description** | Editor, Moderator want to create a new category to use. | |
| **Requirements** |  | |
| **Pre-conditions** | Must logged in as editor or moderator | |
| **Post-conditions** | Success: Tag has been create.. | |
| **Basic flow** | Actor actions:  1.From Home Page, actor click ‘Category’ in submenu  3.Actor input tag information and click ‘Add Category ‘button | System response:  2. System redirect to tag page, return all tag and Add Category group with the following control:  - ‘Name’ text field  - ‘Description’ text area  - ‘Add Category’ button  4.System check add tag information, save into database and return message: “Add category successful!” in Category Page |
| **Alternative flow** | Actor Actions: | System Responses: |
| **Exception** | Actor Actions:  1. Actor left ‘Name’ text field blank  2. Actor left ‘Description’ text area blank | System Responses:  1. ‘Name can’t be blank.’  2. ‘Description can’t be blank.’ |

****

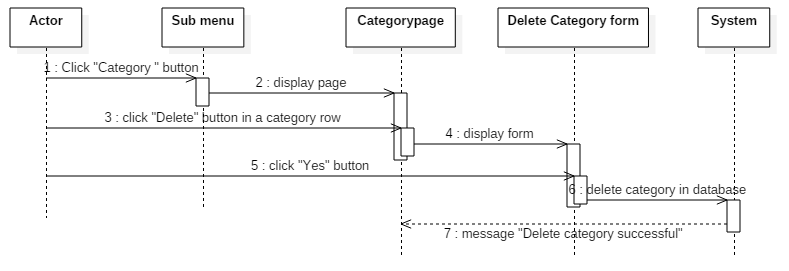
**8.12.5. Update Category**

|  |  |  |
| --- | --- | --- |
| **Author** | Nguyen Hoang Minh | |
| **Use case name** | Update Category | |
| **Actors** | Editor, Moderator | |
| **Description** | Editor, Moderator want to update a category. | |
| **Requirements** | Category have been create. | |
| **Pre-conditions** | Must logged in as editor or moderator | |
| **Post-conditions** | Success: Category has been update. | |
| **Basic flow** | Actor actions:  1.From Home Page, actor click ‘Category’ in submenu  3. Actor click ‘Edit’ button on a category row which want to update.  5. Actor input update category information and click ‘Update’ button. | System response:  2. System redirect to category page, return all tag in a table with the following control:  - ‘Name’  - ‘Author’  - ‘Description’  - ‘Count’  - ‘Edit’ button  - ‘Delete’ button  4. System redirect to an ‘Edit Category’ form with the following control:  - ‘Name’ text field  - ‘Description’ text area  - ‘Update button  6. System check update category information, save into database and return message: “Update category successful!” in Category Page |
| **Alternative flow** | Actor Actions: | System Responses: |
| **Exception** | Actor Actions:  1. Actor left ‘Name’ text field blank  2. Actor left ‘Description’ text area blank | System Responses:  1. ‘Name can’t be blank.’  2. ‘Description can’t be blank.’ |

****

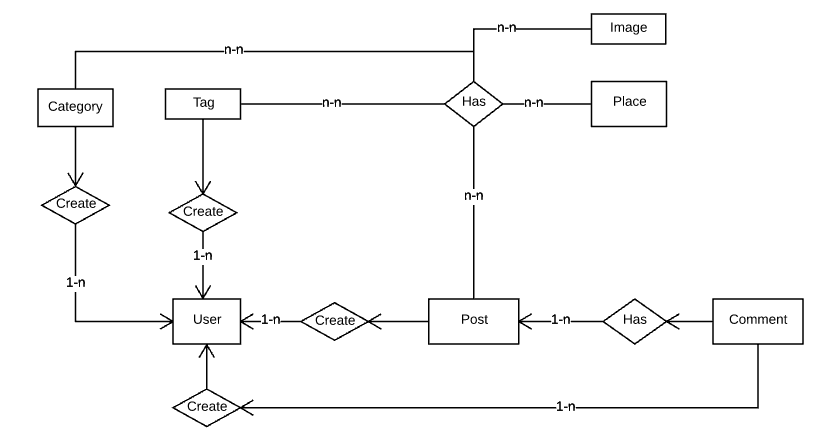
**8.12.23. Delete Category**

|  |  |  |
| --- | --- | --- |
| **Author** | Nguyen Hoang Minh | |
| **Use case name** | Delete category | |
| **Actors** | Editor, Moderator | |
| **Description** | Editor, Moderator want to delete a category. | |
| **Requirements** | Category have been create. | |
| **Pre-conditions** | Must logged in as editor or moderator | |
| **Post-conditions** | Success: Category has been delete. | |
| **Basic flow** | Actor actions:  1.From Home Page, actor click ‘Category’ in submenu  3. Actor click ‘Delete’ button on a category row which want to delete.  5. Actor click ‘Yes’ button | System response:  2. System redirect to category page, return all tag in a table with the following control:  - ‘Name’  - ‘Author’  - ‘Description’  - ‘Count’  - ‘Edit’ button  - ‘Delete’ button  4. System redirect to an ‘Delete Category’ message: “Are you sure you want to delete this category?” and control:  - ‘Yes’ button  - ‘No’ button  6. System delete category in database and return to Category Page with message “Delete category successful.” |
| **Alternative flow** | Actor Actions:  Actor click ‘No’ button | System Responses:  System return to Category Page. |
| **Exception** | Actor Actions: | System Responses |

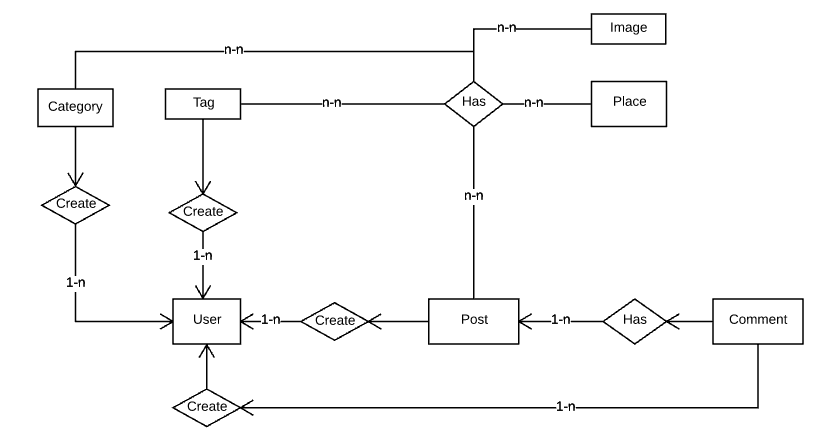
****

**9. Entity Relationship Diagram (ERD)**

**9.1. Conceptual**

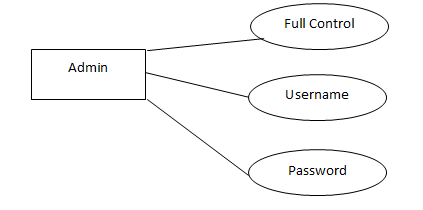
****

**9.2. Logical**

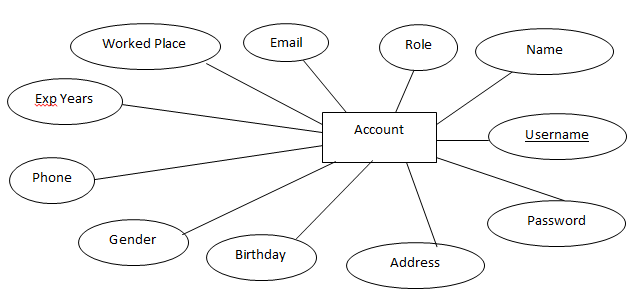
****

**9.3. Table**

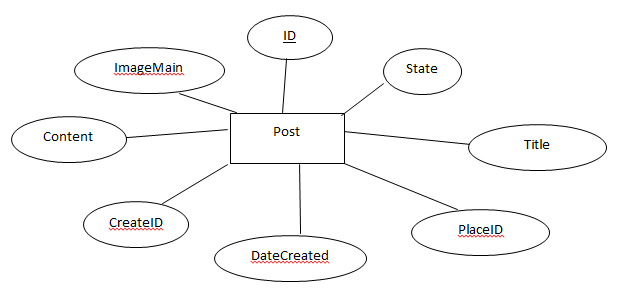
**9.3.1. Admin**

****

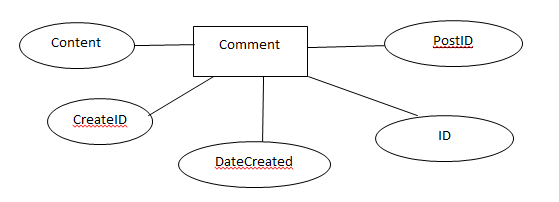
**9.3.2. Account**

****

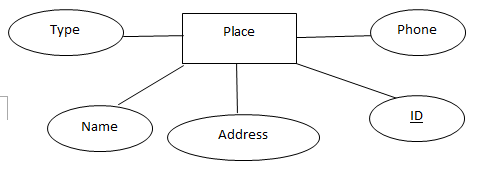
**9.3.3. Post**

****

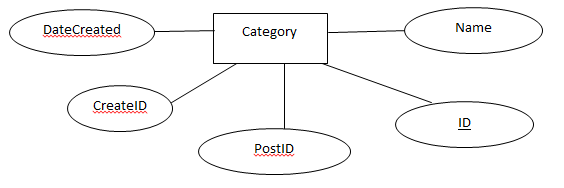
**9.3.4. Comment**

****

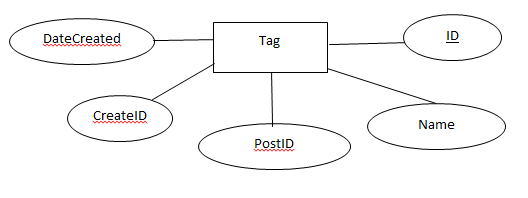
**9.3.5. Place**

****

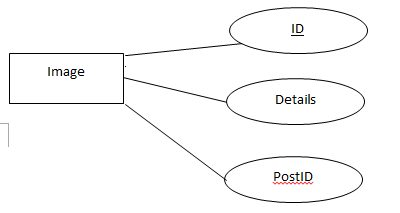
**9.3.6. Category**



**9.3.7. Tag**

****

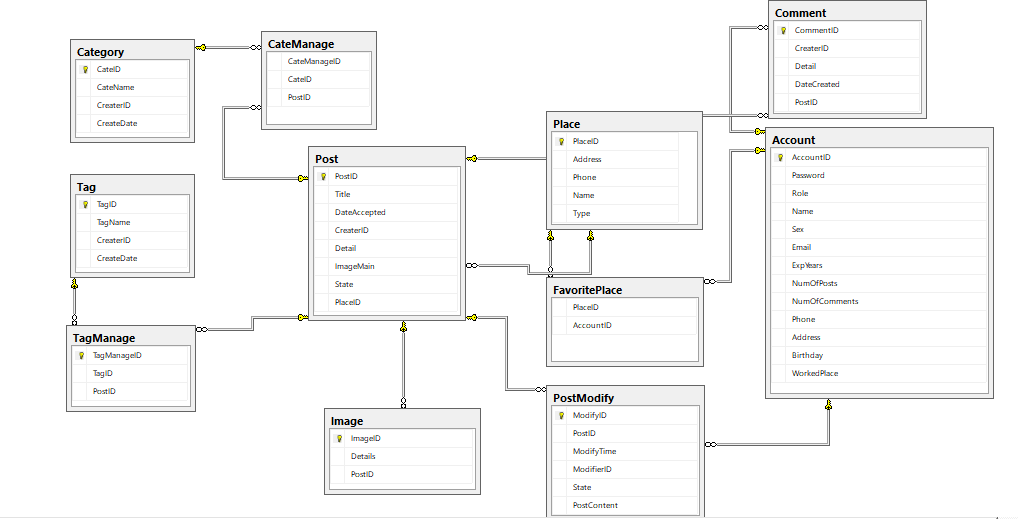
**9.3.8. Image**

****

|  |  |  |
| --- | --- | --- |
|  | **Prepared by (Student)** | **Approved by (Faculty)** |
| **Signature** | Team leader  **Hoang Yen Minh** | **Pham Cong Thanh** |
| **Date** |  |  |

**REVIEW 3**

**10. Database Relationship Diagram**

****

# Table Definition

# Category

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Key | Field Name | Datatype | Leghth | Allow null | Default value | Table-Field-Ref | Description |
| PK | CateID | Varchar | 20 | No |  |  | Category ID |
|  | CateName | Varchar | 50 | No |  |  | Category Name |
|  | CreaterID | Varchar | 20 | No |  |  | ID of who create this category |
|  | CreateDate | Datetime |  |  |  |  | Date create this category |

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Add** | **Delete** | **Update** |
| Category | + | + | - |

# Tag

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Key | Field Name | Datatype | Leghth | Allow null | Default value | Table-Field-Ref | Description |
| PK | TagID | Varchar | 20 | No |  |  | Tag ID |
|  | TagName | Varchar | 50 | No |  |  | Tag Name |
|  | CreaterID | Varchar | 20 | No |  |  | ID of who create this category |
|  | CreateDate | Datetime |  |  |  |  | Date create this tag |

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Add** | **Delete** | **Update** |
| Tag | + | + | - |

# TagManage

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Key | Field Name | Datatype | Leghth | Allow null | Default value | Table-Field-Ref | Description |
| PK | TagManageID | int |  | No |  |  | Tag Manage ID |
|  | TagID | Varchar | 20 | No |  |  | Tag ID |
|  | PostID | Varchar | 20 | No |  |  | Post ID |

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Add** | **Delete** | **Update** |
| TagManage | - | - | - |

# CateManage

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Key | Field Name | Datatype | Leghth | Allow null | Default value | Table-Field-Ref | Description |
| PK | CateManageID | int |  | No |  |  | Tag Manage ID |
|  | CateID | Varchar | 20 | No |  |  | Tag ID |
|  | PostID | Varchar | 20 | No |  |  | Post ID |

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Add** | **Delete** | **Update** |
| CateManage | - | - | - |

# Post

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Key | Field Name | Datatype | Leghth | Allow null | Default value | Table-Field-Ref | Description |
| PK | PostID | Varchar | 20 | No |  |  | Post ID |
|  | Title | Varchar | 100 | No |  |  | Title of post |
|  | DateAccepted | Datetime |  | No |  |  | Date editor accept this post |
|  | CreaterID | Varchar | 20 | No |  |  | ID of post’s creater |
|  | Detail | Varchar | MAX | No |  |  | Detail of post |
|  | ImageMain | Varchar | 100 | No |  |  | Main image of post |
|  | State | Varchar | 20 | No |  |  | Status of post |
|  | PlaceID | Varchar | 20 | No |  |  | ID of place |

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Add** | **Delete** | **Update** |
| Post | + | + | + |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Key | Field Name | Datatype | Leghth | Allow null | Default value | Table-Field-Ref | Description |
| PK | ImageID | Varchar | 20 | No |  |  | ID of image |
|  | Details | Varchar | 100 | No |  |  | Image file location |
|  | PostID | Varchar | 20 | No |  |  | ID of post have this image |

# Image

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Add** | **Delete** | **Update** |
| Image | + | + | + |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Key | Field Name | Datatype | Leghth | Allow null | Default value | Table-Field-Ref | Description |
|  | PlaceID | Varchar | 20 | No |  |  | ID of place |
|  | AccountID | Varchar | 20 | No |  |  | Account ID |

# FavoritePlace

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Add** | **Delete** | **Update** |
| FavoritePlace | + | + | - |

* 1. **Place**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Key | Field Name | Datatype | Leghth | Allow null | Default value | Table-Field-Ref | Description |
| PK | PlaceID | Varchar | 20 | No |  |  | ID of place |
|  | Address | Varchar | 100 | No |  |  | Address of place |
|  | Phone | Varchar | 11 |  |  |  | Phone number |
|  | Name | Varchar | 50 | No |  |  | Name of place |
|  | Type | Varchar | 20 | No |  |  | Type of place(restaurant, small food store, …) |

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Add** | **Delete** | **Update** |
| Place | + | + | + |

# Post Modify

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Key | Field Name | Datatype | Leghth | Allow null | Default value | Table-Field-Ref | Description |
| PK | ModifyID | Varchar | 20 | No |  |  | ModifyID |
|  | PostID | Varchar | 20 | No |  |  | ID of post modify |
|  | ModifyTime | Datetime |  |  |  |  | Modify time |
|  | ModifierID | Varchar | 20 | No |  |  | ID of modifier |
|  | State | Varchar | 20 | No |  |  | State of post |
|  | PostContent | Varchar | Max | No |  |  | Post content |

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Add** | **Delete** | **Update** |
| Post modify | + | + | + |

# Comment

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Key | Field Name | Datatype | Leghth | Allow null | Default value | Table-Field-Ref | Description |
| PK | CommentID | Varchar | 20 | No |  |  | ID of comment |
|  | CreaterID | Varchar | 20 | No |  |  | ID of who comment |
|  | Detail | Varchar | 300 |  |  |  | Detail of comment |
|  | DateCreated | datetime |  |  |  |  | Comment date |
|  | PostID | Varchar | 20 | No |  |  | ID of post has this comment |

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Add** | **Delete** | **Update** |
| Comment | + | + | - |

# Account

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Key | Field Name | Datatype | Leghth | Allow null | Default value | Table-Field-Ref | Description |
| PK | AccountID | Varchar | 20 | No |  |  | ID of account |
|  | Password | Varchar | 50 | No |  |  | Password of account |
|  | Role | Varchar | 20 | No |  |  | Role of account |
|  | Name | Varchar | 100 | No |  |  | Name of user |
|  | Sex | Bit |  |  |  |  | Sex |
|  | Email | Varchar | 100 |  |  |  | Email |
|  | ExpYears | Int |  |  |  |  | Number of experience year |
|  | NumOfPosts | Int |  |  |  |  | Number of post |
|  | NumOfComments | Int |  |  |  |  | Number of comments |
|  | Phone | Varchar | 11 |  |  |  | Phone number of user |
|  | Address | Varchar | 100 |  |  |  | Address of user |
|  | Birthday | datetime |  |  |  |  | Birthday of user |
|  | WorkedPlace | Varchar | 100 |  |  |  | Worked place of user |

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Add** | **Delete** | **Update** |
| Account | + | + | + |

# Other constrains:

## **Category and CateManage**

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Add** | **Delete** | **Update** |
| Category | + | - | - |
| CateManage | +  (Category.CateID must exist) | - | - |

## **Tag and Tagmanage**

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Add** | **Delete** | **Update** |
| Tag | + | - | - |
| TagManage | +  (Tag.TagID must exist) | - | - |

## **Post and CateManage**

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Add** | **Delete** | **Update** |
| Post | + | +  (Disable) | + |
| CateManage | +  (Post.PostID must exist) | - | - |

## **Post and TagManage**

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Add** | **Delete** | **Update** |
| Post | + | +  (Disable) | + |
| TagManage | +  (Post.PostID must exist) | - | - |

## 

## **12.5 Post and Image**

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Add** | **Delete** | **Update** |
| Post | + | +  (Disable) | + |
| TagManage | +  (Post.PostID must exist) | - | - |

## 

## **12.6 Post and Place**

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Add** | **Delete** | **Update** |
| Place | + | - | - |
| Post | +  (Place.PlaceID must exist) | +  (Disable) | + |

## 

## **12.7 Post and Comment**

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Add** | **Delete** | **Update** |
| Comment | + | + | - |
| Post | +  (Comment.CommentID must exist) | +  (Disable) | + |

## **12.8 Post and PostModify**

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Add** | **Delete** | **Update** |
| Post | + | +  (Disable) | + |
| PostModify | +  (Post.PostID must exist) | - | - |

## **12.9 Place and FavoritePlace**

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Add** | **Delete** | **Update** |
| Place | + | - | - |
| FavoritePlace | +  (Place.PlaceID must exist) | +  (Disable) | + |

## **12.10 Account and FavoritePlace**

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Add** | **Delete** | **Update** |
| Account | + | - | +  (Can’t update AccountID) |
| FavoritePlace | +  (Account.AccountID must exist) | +  (Disable) | + |

## **12.11 Account and PostModify**

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Add** | **Delete** | **Update** |
| Account | + | - | +  (Can’t update AccountID) |
| PostModify | +  (Account.AccountID must exist) | - | - |

## **12.12 Account and Comment**

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Add** | **Delete** | **Update** |
| Account | + | - | +  (Can’t update AccountID) |
| Comment | +  (Account.AccountID must exist) | + | - |

1. **Side map**

**13.1 Home side**

home.jsp

Login.jsp

registerPolicy.jsp

Register.jsp

My Account

aboutUs.jsp

sendEnquiry.jsp



commentOnPost.jsp

postDetail.jsp



Post.jsp

changePassword.jsp

updateProfile.jsp



**13.2 Moderator side**

home.jsp

Login.jsp

registerPolicy.jsp

Register.jsp

My Account

aboutUs.jsp

sendEnquiry.jsp



commentOnPost.jsp

postDetail.jsp



Post.jsp

changePassword.jsp

updateProfile.jsp



updatePost.jsp



removePost.jsp



viewAllPost.jsp



addPost.jsp



Post

**13.3 Editor side**

home.jsp

Login.jsp

registerPolicy.jsp

Register.jsp

My Account

changePassword.jsp

updateProfile.jsp



Post.jsp

postDetail.jsp



commentOnPost.jsp

sendEnquiry.jsp



aboutUs.jsp

Post

addPost.jsp



viewAllPost.jsp



updatePost.jsp



removePost.jsp



acceptedPost.jsp



declinePost.jsp



viewWaitingPost.jsp



blockMember.jsp



viewMember.jsp



Member

addEditor.jsp



# GUI Design



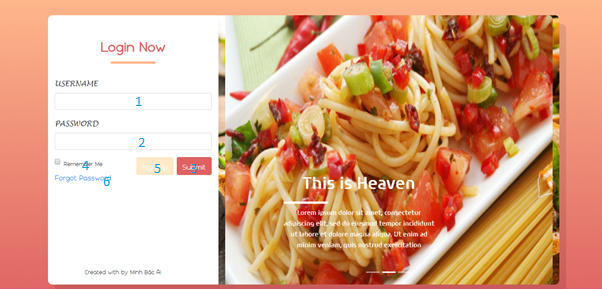
**14.1.1 Home page**

This is Home page of the website, the customers will go to this page first.



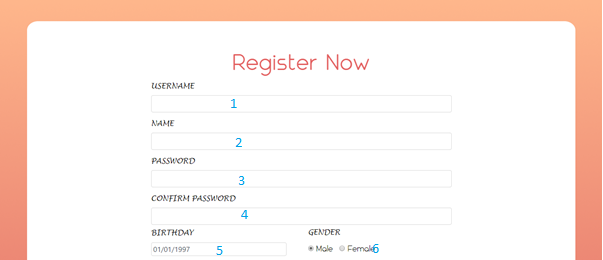
|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **No.** | **Name** | **Type** | **Validation** | **Event** | **Description** | **Status** |
| 1 | hplHome | Hyperlink |  | click | Redirect to Home page |  |
| 2 | hplAbout | Hyperlink |  | click | Move to about part of Home page |  |
| 3 | hplArticle | Hyperlink |  | click | Move to article part of Home page |  |
| 4 | hplContact | Hyperlink |  | click | Move to contact part of Home page |  |
| 5 | hplLogin | Hyperlink |  | click | Redirect to Login page |  |
| 6 | hplSignUp | Hyperlink |  | click | Redirect to Sign up page |  |

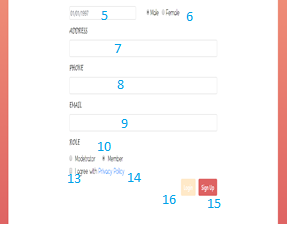
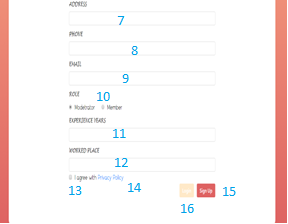
**14.1.2 Login page**

****

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **No.** | **Name** | **Type** | **Validation** | **Event** | **Description** | **Status** |
| 1 | txtUsername | textbox |  | click | Input username |  |
| 2 | txtPass | textbox |  | click | Input password |  |
| 3 | btnSubmit | button |  | click | Check login information and redirect to Home page |  |
| 4 | cbRemember | checkbox |  | click | Remember user’s username and password |  |
| 5 | btnSignUp | button |  | Click | Redirect to Sign up page |  |
| 6 | hplForgotPass | Hyperlink |  | Click | Redirect to forgot pass page |  |

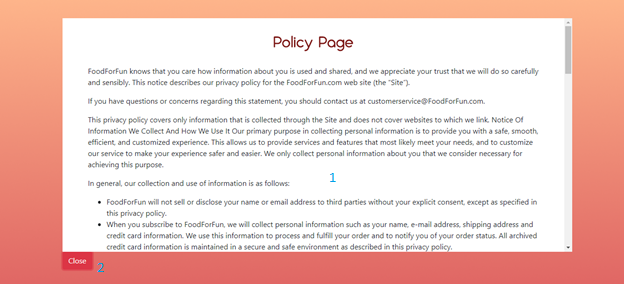
**14.1.3 Register**

****

****

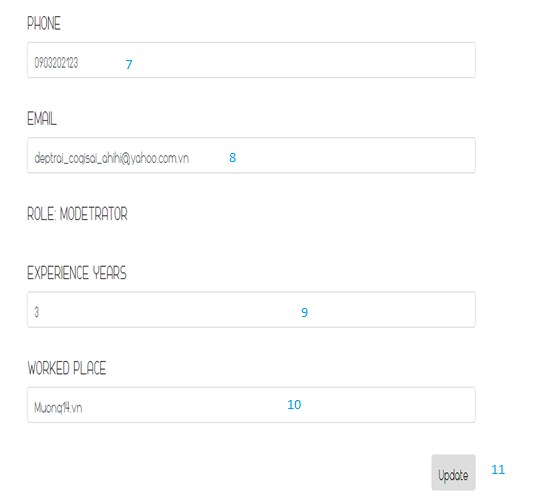
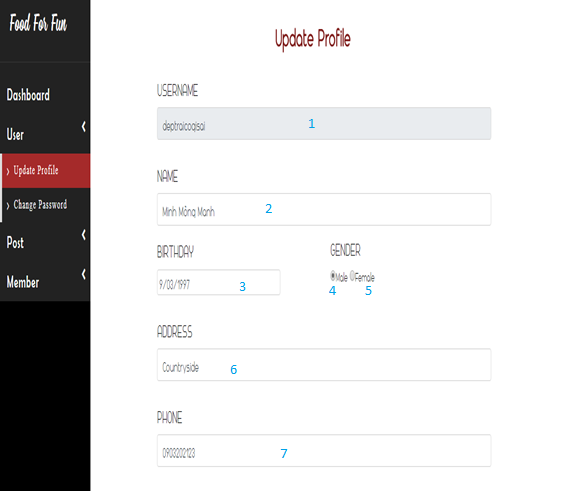
|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **No.** | **Name** | **Type** | **Validation** | **Event** | **Description** | **Status** |
| 1 | txtUsername | textbox |  | click | Input username |  |
| 2 | txtName | textbox |  | click | Input name |  |
| 3 | txtPass | textbox |  | click | Input password |  |
| 4 | txtConfirmPass | textbox |  | click | Confirm password |  |
| 5 | txtDOB | Datetimepicker |  | click | User birthday |  |
| 6 | rbGender | radiobutton |  | check | User gender |  |
| 7 | txtAddress | textbox |  | click | User address |  |
| 8 | txtPhone | textbox |  | click | User phone  number |  |
| 9 | txtEmail | textbox |  | click | User email |  |
| 10 | rbRole | radiobutton |  | check | User role |  |
| 11 | txtExpYear | textbox |  | click | Num ber of moderator experience year |  |
| 12 | txtWorkedPlace | textbox |  | click | Moderator worked place |  |
| 13 | cbAgree | Checkbox |  | Click | Agree with policy |  |
| 14 | hplPolicy | Hyperlink |  | Click | Redirect to policy page |  |
| 15 | btnSignUp | button |  | Click | Redirect to Sign up page |  |
| 16 | btnLogin | button |  | Click | Redirect to login page |  |

**14.1.4 Register Policy Page**

****

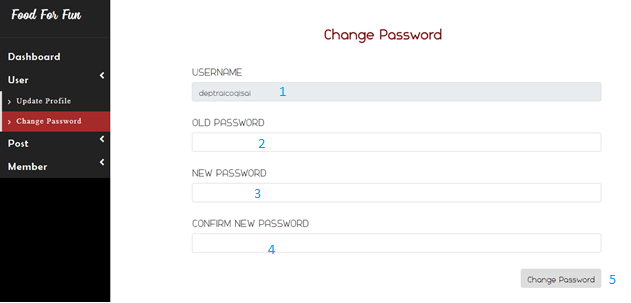
|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **No.** | **Name** | **Type** | **Validation** | **Event** | **Description** | **Status** |
| 1 | lblRegistrationPolicy | Lable |  |  | Registration Policy |  |
| 2 | btnClose | Button |  | Click | Back to previous Page |  |

**14.1.5 Update profile**

****

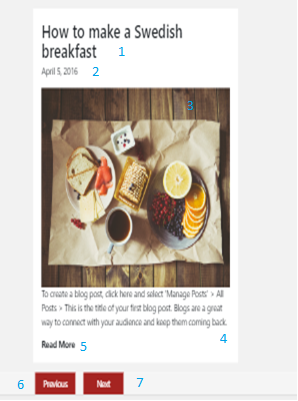
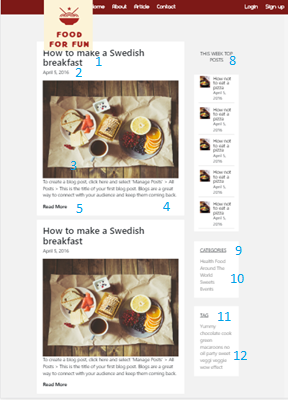
|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **No.** | **Name** | **Type** | **Validation** | **Event** | **Description** | **Status** |
| 1 | txtUsername | Textbox | 6<=length<=30  readonly |  | User’s Signing Name |  |
| 2 | txtName | Textbox | 6<=length<=100 |  | User’s Name |  |
| 3 | txtBirthday | Textbox |  |  | User’s Birthday |  |
| 4 | rdMale | Radio button |  | Click | Gender selection |  |
| 5 | rdFemale | Radio button |  | click | Gender selection |  |
| 6 | txtAddress | Textbox | 6<=length<=100 |  | User’s Address |  |
| 7 | txtPhone | Textbox | 8<=length<=11  [0-9], allow null |  | User ‘s Phone |  |
| 8 | txtEmail | Textbox | Email Type |  | User’s Email |  |
| 9 | txtExpYear | Textbox | int |  | Year of experience |  |
| 10 | txtWorkedPlace | Textbox |  |  | Worked place |  |
| 11 | btnUpdate | Button |  | Click | Save user’s profile, redirect to user’s home page. |  |

**14.1.6 Change Password**

****

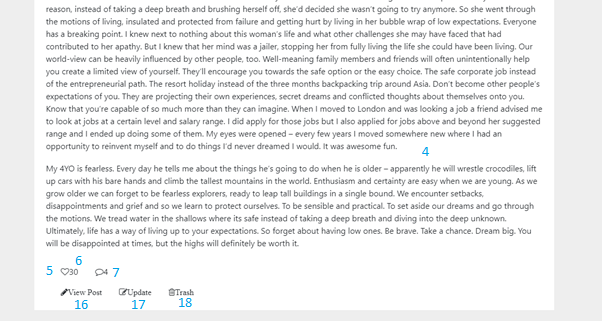
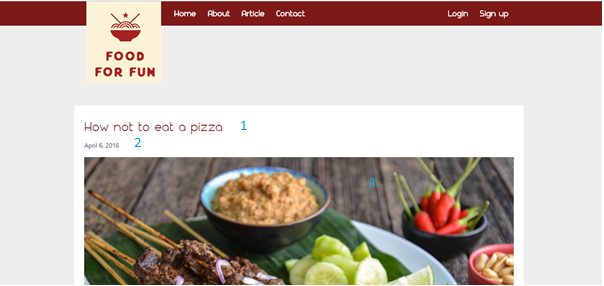
|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **No.** | **Name** | **Type** | **Validation** | **Event** | **Description** | **Status** |
| 1 | txtUsername | Textbox | 6<=length<=30  readonly |  | User’s Signing Name |  |
| 2 | txtOldPassword | Textbox | 6<=length<=100,  Readonly |  | User’s Old Password |  |
| 3 | txtNewPassword | Textbox | 6<=length<=30 |  | User’s New Password |  |
| 4 | txtConfirmPassword | Textbox | 6<=length<=30 |  | Confirm Password |  |
| 5 | btnChangePassword | Button |  | Click | Save user’s new password, redirect to user’s home page. |  |

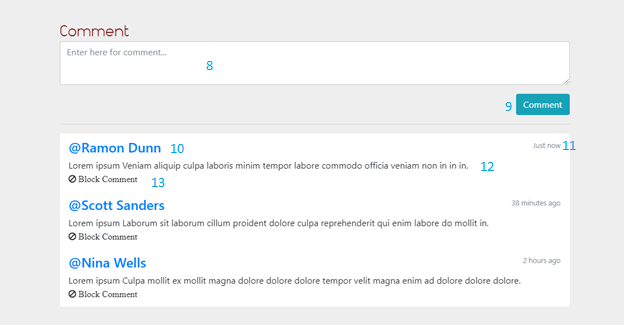
**14.1.7 Post**

****

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **No.** | **Name** | **Type** | **Validation** | **Event** | **Description** | **Status** |
| 1 | lblTitle | Lable |  |  | Title of a Post |  |
| 2 | lbDateOfPost | Lable |  |  | Datetime of post when allow public |  |
| 3 | imgMailPost | Image |  |  | Image of a Post |  |
| 4 | lblContent | Lable |  |  | Content of a Post |  |
| 5 | btnReadMore | Button |  | Click | Redirect to Post’s detail |  |
| 6 | btnSaPrevious | Button |  | Click | Redirect article previous page with older post |  |
| 7 | btnNext | Button |  | Click | Redirect to article new page with newer post |  |
| 8 | gridWeekPost | Grid |  |  | Grid contains post of weeks |  |
| 9 | gridCategories | Grid |  |  | Grid contains controls: 10 |  |
| 10 | hyperlinkCategories | Hyperlink |  | click | Hyperlink categories |  |
| 11 | gridTags | Grid |  |  | Grid contains controls: 12 |  |
| 12 | hyperlinkTags | Hyperlink |  | Click | Hyperlink tags |  |

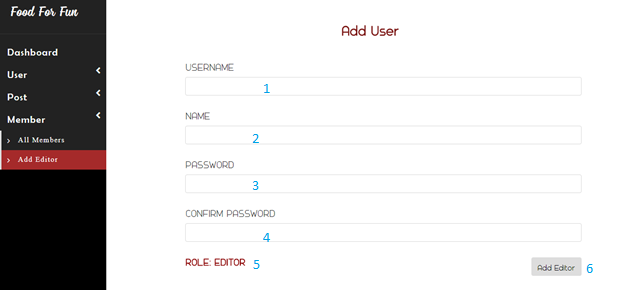
**14.1.8 Post Detail**





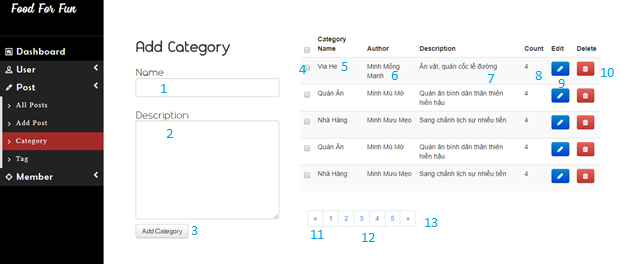
|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **No.** | **Name** | **Type** | **Validation** | **Event** | **Description** | **Status** |
| 1 | lbTitle | label |  |  | Title of post |  |
| 2 | lbDateCreate | label |  |  | Date create this post |  |
| 3 | imgMain | image |  |  | Main image of post |  |
| 4 | txtareaContent | text area |  |  | Content of post |  |
| 5 | btnLike | button |  | click | Like this post |  |
| 6 | lbLike | label |  |  | Liked number of this post |  |
| 7 | lbCmtIcon | label |  |  | Number of this post’s comment |  |
| 8 | txtArea | textarea |  | click | Input comment |  |
| 9 | btnComment | button |  | click | Send comment to system |  |
| 10 | lbCmtUser | label |  |  | Username of this comment creator |  |
| 11 | lbCmtTime | label |  |  | How much time does this comment exist |  |
| 12 | lbCmtContent | label |  |  | Comment’s content |  |
| 13 | btnBlockCmt | button |  | click | Block this comment |  |
| 14 | hplViewPost | Hyperlink |  | click | View this post |  |
| 15 | hplUpdate | Hyperlink |  | Click | Update this post |  |
| 16 | hplTrash | Hyperlink |  | Click | Delete this post |  |

**14.1.9 Add user**

****

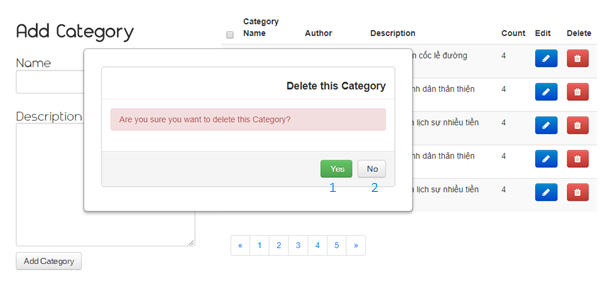
|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **No.** | **Name** | **Type** | **Validation** | **Event** | **Description** | **Status** |
| 1 | txtUsername | Textbox | 6<=length<=30  readonly |  | User’s Signing Name |  |
| 2 | txtName | Textbox |  |  | User’s Name |  |
| 3 | txtPassword | Textbox | 6<=length<=30 |  | User’s Password |  |
| 4 | txtConfirmPassword | Textbox | 6<=length<=30  Value==txtPassword |  | Confirm Password |  |
| 5 | lblRoleEditor | Label |  |  | Role of adding user |  |
| 6 | btnAddEditor | Button |  | Click | Save new editor, redirect to home page. |  |

**14.1.10 Add category**

****

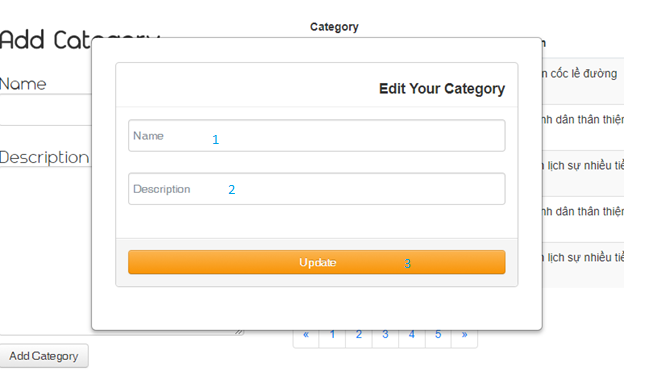
|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **No.** | **Name** | **Type** | **Validation** | **Event** | **Description** | **Status** |
| 1 | txtCateName | Textbox |  |  | Category name |  |
| 2 | txtCateDescription | Textbox |  |  | Category description |  |
| 3 | btnAddCate | Button |  | Click | Add new category to category list |  |
| 4 | cbChoose | Combobox |  |  | Selected category |  |
| 5 | lblCateName | Label |  |  | Category name |  |
| 6 | lblAuthor | Label |  |  | Who create this category |  |
| 7 | lblDescription | Label |  |  | Category description |  |
| 8 | lblCount | Label |  |  | Number of post use this category |  |
| 9 | btnEdit | Button |  | Click | Open edit window to edit category |  |
| 10 | btnDelete | Button |  | Click | Open delete category window |  |
| 11 | btnPrevious | Button |  | Click | Show previous category list |  |
| 12 | btnPageNum | Button |  | Click | Show page number of category list and move to that page number when clicked |  |
| 13 | btnNext | Button |  | Click | Show next category list |  |

**14.1.11 Delete category**

****

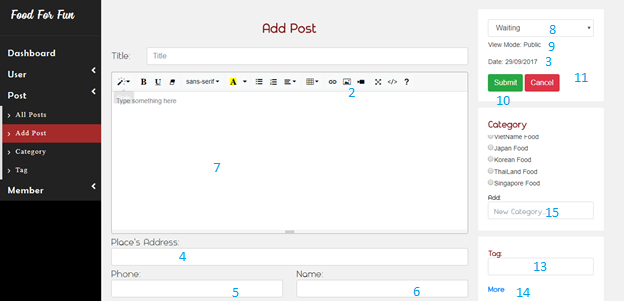
|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **No.** | **Name** | **Type** | **Validation** | **Event** | **Description** | **Status** |
| 1 | btnYes | Button |  | Click | Close this window and delete this category |  |
| 2 | btnNo | Button |  | Click | Close this window without delete this category |  |

**14.1.12 Edit category**

****

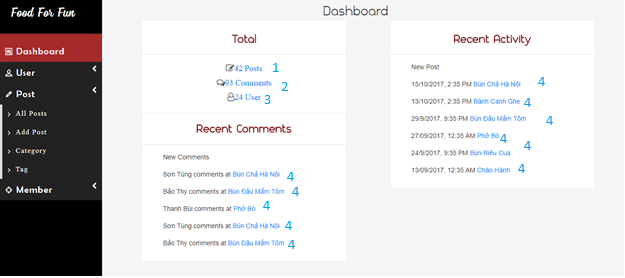
|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **No.** | **Name** | **Type** | **Validation** | **Event** | **Description** | **Status** |
| 1 | txtCateName | Textbox |  |  | Category name |  |
| 2 | txtCateDescription | Textbox |  |  | Category description |  |
| 3 | btnUpdate | Button |  | Click | Update the chosen category and close update window |  |

**14.1.13 Add post**

****

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **No.** | **Name** | **Type** | **Validation** | **Event** | **Description** | **Status** |
| 1 | txtTitle | textbox |  |  | Post title |  |
| 2 | btnAddImage | button |  | click | Add image |  |
| 3 | date | label |  |  | Local date |  |
| 4 | txtAddress | textbox |  |  | Store address |  |
| 5 | txtPhone | textbox |  |  | Phone number of restaurant manager |  |
| 6 | txtName | textbox |  |  | Name of restaurant manager |  |
| 7 | txtareaContent | textarea |  |  | Post content |  |
| 8 | cbTypeOfPost | combobox |  | click | Post type |  |
| 9 | lbViewMode | label |  |  | View mode |  |
| 10 | btnSubmit | button |  | click | Submit post |  |
| 11 | btnCancel | button |  | click | Cancel post |  |
| 12 | rbCategory | radiobutton |  | check | Choose category |  |
| 13 | txtTag | textbox |  | click | Add tag |  |
| 14 | hplMore | Hyperlink |  | click | More tag |  |
| 15 | txtAddNewCate | Textbox |  |  | New Cate |  |

**14.1.14 Dashboard**

****

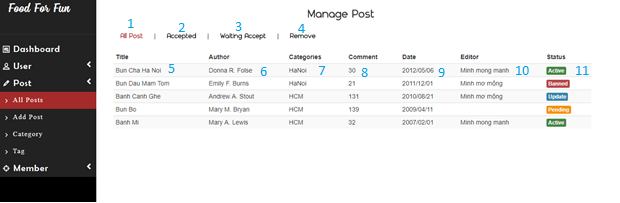
|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **No.** | **Name** | **Type** | **Validation** | **Event** | **Description** | **Status** |
| 1 | btnPostNum | button |  | Click | View all post |  |
| 2 | btnCommentNum | Button |  | Click | View all comment |  |
| 3 | btnUserNum | Button |  | Click | View all user |  |
| 4 | hplPostName | Hyperlink |  | click | View details of this post |  |

**14.1.15 All user**

****

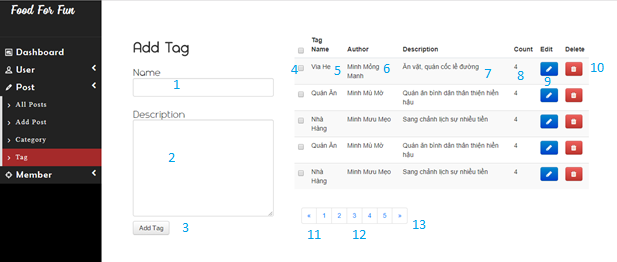
|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **No.** | **Name** | **Type** | **Validation** | **Event** | **Description** | **Status** |
| 1 | lblName | label |  |  | Name of user |  |
| 2 | lblUsername | label |  |  | Username |  |
| 3 | btnRole | button |  |  | Role of user |  |
| 4 | btnDelete | button |  | click | Delete this user |  |

**14.1.16 Manage post**



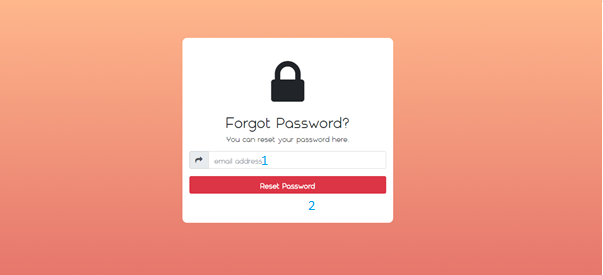
|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **No.** | **Name** | **Type** | **Validation** | **Event** | **Description** | **Status** |
| 1 | txtAllPost | button |  | click | Show all of post |  |
| 2 | txtAccepted | button |  | click | Show accepted post |  |
| 3 | txtWaitingAccept | button |  | click | Show waiting post |  |
| 4 | txtRemove | button |  | click | Show remove post window |  |
| 5 | lbTitle | label |  |  | Post title |  |
| 6 | lbAuthor | label |  |  | Post author |  |
| 7 | lbCate | label |  |  | Post category |  |
| 8 | lbNumOfCmt | label |  |  | Number of comment on this post |  |
| 9 | lbDate | label |  |  | Date create |  |
| 10 | lbEditor | label |  |  | Editor working on this post |  |
| 11 | lbStatus | label |  |  | Status of post |  |

**14.1.17 Add tag**

****

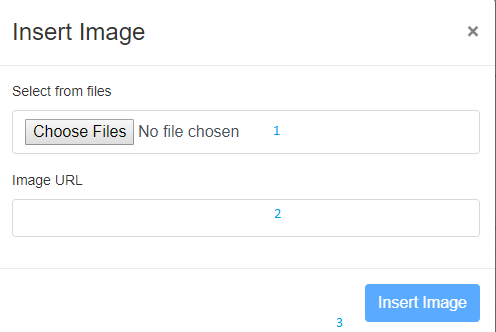
|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **No.** | **Name** | **Type** | **Validation** | **Event** | **Description** | **Status** |
| 1 | txtTagName | Textbox |  |  | Tag name |  |
| 2 | txtTagDescription | Textbox |  |  | Tagdescription |  |
| 3 | btnAddTag | Button |  | Click | Add new tag to tag list |  |
| 4 | cbChoose | Combobox |  |  | Selected Tag |  |
| 5 | lblTagName | Label |  |  | Tag name |  |
| 6 | lblAuthor | Label |  |  | Who create this Tag |  |
| 7 | lblDescription | Label |  |  | Tag description |  |
| 8 | lblCount | Label |  |  | Number of post use this tag |  |
| 9 | btnEdit | Button |  | Click | Open edit window to edit tag |  |
| 10 | btnDelete | Button |  | Click | Open delete tag window |  |
| 11 | btnPrevious | Button |  | Click | Show previous tag list |  |
| 12 | btnPageNum | Button |  | Click | Show page number of tag list and move to that page number when clicked |  |
| 13 | btnNext | Button |  | Click | Show next tag list |  |

**14.1.18 Forgot password**



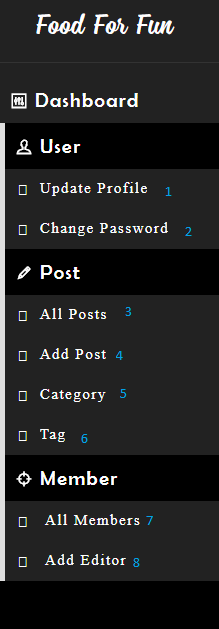
|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **No.** | **Name** | **Type** | **Validation** | **Event** | **Description** | **Status** |
| 1 | txtEmailAddress | Textbox | Email formatting |  | Enter email address to get new password |  |
| 2 | btnResetPassword | Button |  |  | Send new password to email |  |

**14.1.19 Insert Image**



|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **No.** | **Name** | **Type** | **Validation** | **Event** | **Description** | **Status** |
| 1 | btnChooseFile | Button |  | Click | Choose image |  |
| 2 | txtImageURL | Textfield |  |  | URL of chosen image |  |
| 3 | btnInsertImage | Button |  | Click | Insert chosen image to post |  |

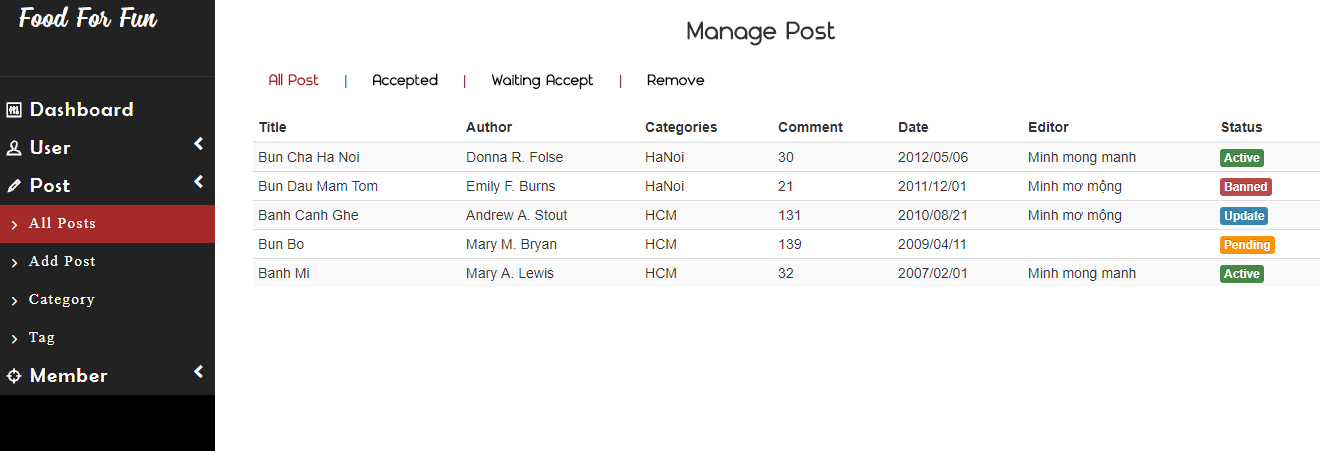
**14.1.20 Sub menu**



|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **No.** | **Name** | **Type** | **Validation** | **Event** | **Description** | **Status** |
| 1 | hplUpdateProfile | hyperlink |  | Click | Redirect to Update Profile Page |  |
| 2 | hplChangePassword | hyperlink |  | Click | Redirect to Change password Page |  |
| 3 | hplAllPost | hyperlink |  | Click | Redirect to All Post Page |  |
| 4 | hplAddPost | hyperlink |  | Click | Redirect to Add post Page |  |
| 5 | hplCategory | hyperlink |  | Click | Redirect to Category Manage Page |  |
| 6 | hplTag | hyperlink |  | Click | Redirect to Tag Manage Page |  |
| 7 | hplAllMember | hyperlink |  | Click | Redirect to All Member Page |  |
| 8 | hplAddEditor | hyperlink |  | Click | Redirect to Add Editor Page |  |

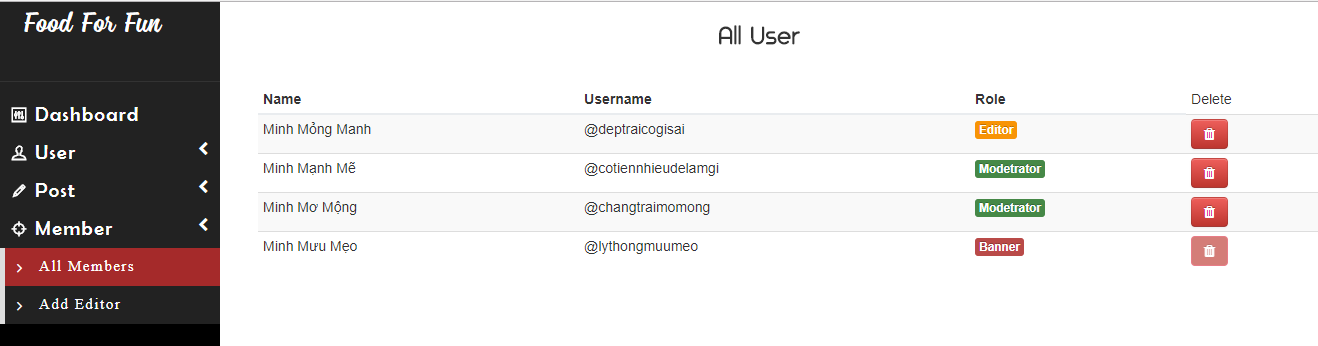
1. **Report**

**15.1 Post**



|  |  |
| --- | --- |
| **Report Resources** | |
| **Table name** | **Field name** |
| Manage Post | Title, Author, Categories, Comment, Date, Editor, Status |

**15.2 User**



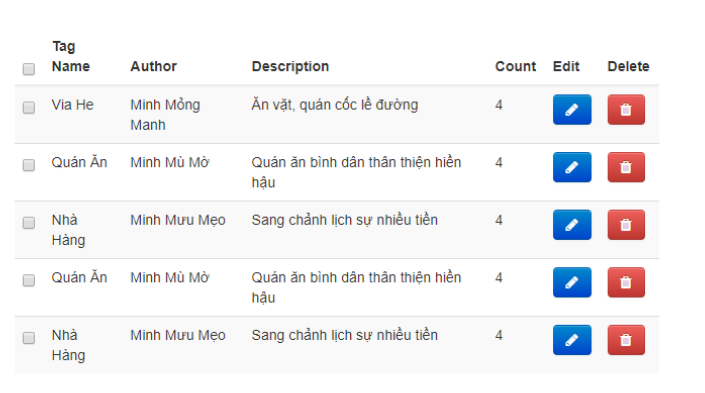
|  |  |
| --- | --- |
| **Report Resources** | |
| **Table name** | **Field name** |
| All user | Name, Username, Role, Delete |

**15.3 Category**



|  |  |
| --- | --- |
| **Report Resources** | |
| **Table name** | **Field name** |
| Category | Category Name, Author, Description, Count, Edit, Delete |

**15.4 Tag**



|  |  |
| --- | --- |
| **Report Resources** | |
| **Table name** | **Field name** |
| Tag | Tag Name, Author, Description, Count, Edit, Delete |

|  |  |  |
| --- | --- | --- |
|  | **Prepared by (Student)** | **Approved by (Faculty)** |
| **Signature** | Team leader  **Hoang Yen Minh** | **Pham Cong Thanh** |
| **Date** |  |  |

TASK SHEET

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Project Ref. No 1** | | **Project Title** | **Activity Plan Prepared By** | **Date of Preparation of Activity Plan** | | | | |
| **Sr. No** | **Task** | **Start Date** | **End Date** | **Actual Days** | **Member’s name** | **Status** |
| **CRS document** | | | | | | | | |
| 1 | Introduction | FoodForFun | Hoang Yen Minh | 16/09/2017 | 23/09/2017 | 3 | All of team | FINISH |
| 2 | Existing Scenario | FoodForFun | Hoang Yen Minh | 16/09/2017 | 23/09/2017 | 3 | All of team | FINISH |
| 3 | Customer Requirement Specification | FoodForFun | Hoang Yen Minh | 16/09/2017 | 23/09/2017 | 3 | All of team | FINISH |
| 4 | Functional Requirement Specification | FoodForFun | Hoang Yen Minh | 16/09/2017 | 23/09/2017 | 3 | All of team | FINISH |
| 5 | System Requirements | FoodForFun | Hoang Yen Minh | 16/09/2017 | 23/09/2017 | 3 | All of team | FINISH |
| 6 | Development Software | FoodForFun | Hoang Yen Minh | 16/09/2017 | 23/09/2017 | 3 | All of team | FINISH |
| **Use Case and Sequence Diagrams** | | | | | | | | |
| 7 | Register as Member | FoodForFun | Hoang Yen Minh | 16/09/2017 | 23/09/2017 | 4 | Hoang Yen Minh | FINISH |
| 8 | Register as Moderator | FoodForFun | Hoang Yen Minh | 16/09/2017 | 23/09/2017 | 4 | Hoang Yen Minh | FINISH |
| 9 | Login | FoodForFun | Hoang Yen Minh | 23/09/2017 | 30/09/2017 | 4 | Nguyen Hoang Minh | FINISH |
| 10 | Logout | FoodForFun | Hoang Yen Minh | 23/09/2017 | 30/09/2017 | 4 | Nguyen Hoang Minh | FINISH |
| 11 | Update Profile | FoodForFun | Hoang Yen Minh | 23/09/2017 | 30/09/2017 | 4 | Nguyen Hoang Minh | FINISH |
| 12 | Forgot Password | FoodForFun | Hoang Yen Minh | 21/10/2017 | 28/10/2017 | 1 | Dao Nhat Minh | FINISH |
| 13 | Change Password | FoodForFun | Hoang Yen Minh | 23/09/2017 | 30/09/2017 | 4 | Nguyen Hoang Minh | FINISH |
| 14 | Add Comment | FoodForFun | Hoang Yen Minh | 23/09/2017 | 30/09/2017 | 4 | Nguyen Hoang Minh | FINISH |
| 15 | Update Comment | FoodForFun | Hoang Yen Minh | 23/09/2017 | 30/09/2017 | 4 | Nguyen Hoang Minh | FINISH |
| 16 | Delete Comment | FoodForFun | Hoang Yen Minh | 23/09/2017 | 30/09/2017 | 4 | Nguyen Hoang Minh | FINISH |
| 17 | Modify Post | FoodForFun | Hoang Yen Minh | 23/09/2017 | 30/09/2017 | 4 | Hoang Yen Minh | FINISH |
| 18 | Create Post | FoodForFun | Hoang Yen Minh | 23/09/2017 | 30/09/2017 | 4 | Hoang Yen Minh | FINISH |
| 19 | Remove Post | FoodForFun | Hoang Yen Minh | 23/09/2017 | 30/09/2017 | 4 | Hoang Yen Minh | FINISH |
| 20 | Accept Post | FoodForFun | Hoang Yen Minh | 23/09/2017 | 30/09/2017 | 4 | Dao Nhat Minh | FINISH |
| 21 | Update Post | FoodForFun | Hoang Yen Minh | 23/09/2017 | 30/09/2017 | 4 | Dao Nhat Minh | FINISH |
| 22 | Delete Post | FoodForFun | Hoang Yen Minh | 23/09/2017 | 30/09/2017 | 4 | Dao Nhat Minh | FINISH |
| 23 | View Post | FoodForFun | Hoang Yen Minh | 23/09/2017 | 30/09/2017 | 4 | Dao Nhat Minh | FINISH |
| 24 | View Waiting Post | FoodForFun | Hoang Yen Minh | 23/09/2017 | 30/09/2017 | 4 | Dao Nhat Minh | FINISH |
| 25 | View Accepted Post | FoodForFun | Hoang Yen Minh | 23/09/2017 | 30/09/2017 | 4 | Dao Nhat Minh | FINISH |
| 26 | View Waiting Modify Post | FoodForFun | Hoang Yen Minh | 23/09/2017 | 30/09/2017 | 4 | Dao Nhat Minh | FINISH |
| 27 | Block Comment | FoodForFun | Hoang Yen Minh | 23/09/2017 | 30/09/2017 | 4 | Dao Nhat Minh | FINISH |
| 28 | View Account Information | FoodForFun | Hoang Yen Minh | 23/09/2017 | 30/09/2017 | 4 | Dao Nhat Minh | FINISH |
| 29 | View Member Account Information | FoodForFun | Hoang Yen Minh | 23/09/2017 | 30/09/2017 | 4 | Dao Nhat Minh | FINISH |
| 30 | View Moderator Account Information | FoodForFun | Hoang Yen Minh | 23/09/2017 | 30/09/2017 | 4 | Dao Nhat Minh | FINISH |
| 31 | Enquiry | FoodForFun | Hoang Yen Minh | 23/09/2017 | 30/09/2017 | 4 | Nguyen Hoang Minh | FINISH |
| 32 | Add Editor | FoodForFun | Hoang Yen Minh | 23/09/2017 | 30/09/2017 | 4 | Nguyen Hoang Minh | FINISH |
| 33 | Add Tag | FoodForFun | Hoang Yen Minh | 21/10/2017 | 28/10/2017 | 3 | Nguyen Hoang Minh | FINISH |
| 34 | Update Tag | FoodForFun | Hoang Yen Minh | 21/10/2017 | 28/10/2017 | 3 | Nguyen Hoang Minh | FINISH |
| 35 | Delete Tag | FoodForFun | Hoang Yen Minh | 21/10/2017 | 28/10/2017 | 3 | Nguyen Hoang Minh | FINISH |
| 36 | Add Category | FoodForFun | Hoang Yen Minh | 21/10/2017 | 28/10/2017 | 3 | Nguyen Hoang Minh | FINISH |
| 37 | Update Category | FoodForFun | Hoang Yen Minh | 21/10/2017 | 28/10/2017 | 3 | Nguyen Hoang Minh | FINISH |
| 38 | Delete Category | FoodForFun | Hoang Yen Minh | 21/10/2017 | 28/10/2017 | 3 | Nguyen Hoang Minh | FINISH |
| **ERD** | | | | | | | | |
| 39 | Conceptual | FoodForFun | Hoang Yen Minh | 23/09/2017 | 30/09/2017 | 4 | Hoang Yen Minh | FINISH |
| 40 | Logical | FoodForFun | Hoang Yen Minh | 23/09/2017 | 30/09/2017 | 4 | Hoang Yen Minh | FINISH |
| 41 | Table | FoodForFun | Hoang Yen Minh | 23/09/2017 | 30/09/2017 | 4 | Hoang Yen Minh | FINISH |
| 42 | Database Relationship | FoodForFun | Hoang Yen Minh | 23/09/2017 | 30/09/2017 | 4 | Dao Nhat Minh | FINISH |
| 43 | Table Definition | FoodForFun | Hoang Yen Minh | 23/09/2017 | 30/09/2017 | 4 | Dao Nhat Minh | FINISH |
| 44 | Side Map | FoodForFun | Hoang Yen Minh | 23/09/2017 | 30/09/2017 | 4 | Nguyen Hoang Minh | FINISH |
| **GUI** | | | | | | | | |
| 45 | Homepage | FoodForFun | Hoang Yen Minh | 30/09/2017 | 07/10/2017 | 5 | All of team | FINISH |
| 46 | login | FoodForFun | Hoang Yen Minh | 30/09/2017 | 07/10/2017 | 5 | All of team | FINISH |
| 47 | register | FoodForFun | Hoang Yen Minh | 30/09/2017 | 07/10/2017 | 5 | All of team | FINISH |
| 48 | Register policy | FoodForFun | Hoang Yen Minh | 30/09/2017 | 07/10/2017 | 5 | All of team | FINISH |
| 49 | updateProfile | FoodForFun | Hoang Yen Minh | 30/09/2017 | 07/10/2017 | 5 | All of team | FINISH |
| 50 | Change password | FoodForFun | Hoang Yen Minh | 30/09/2017 | 07/10/2017 | 5 | All of team | FINISH |
| 51 | Post | FoodForFun | Hoang Yen Minh | 30/09/2017 | 07/10/2017 | 5 | All of team | FINISH |
| 52 | Post Detail | FoodForFun | Hoang Yen Minh | 30/09/2017 | 07/10/2017 | 5 | All of team | FINISH |
| 53 | Add user | FoodForFun | Hoang Yen Minh | 30/09/2017 | 07/10/2017 | 5 | All of team | FINISH |
| 54 | Add category | FoodForFun | Hoang Yen Minh | 30/09/2017 | 07/10/2017 | 5 | All of team | FINISH |
| 55 | Delete Category | FoodForFun | Hoang Yen Minh | 30/09/2017 | 07/10/2017 | 5 | All of team | FINISH |
| 56 | Edit Category | FoodForFun | Hoang Yen Minh | 30/09/2017 | 07/10/2017 | 5 | All of team | FINISH |
| 57 | Add post | FoodForFun | Hoang Yen Minh | 30/09/2017 | 07/10/2017 | 5 | All of team | FINISH |
| 58 | Dashboard | FoodForFun | Hoang Yen Minh | 30/09/2017 | 07/10/2017 | 5 | All of team | FINISH |
| 59 | All user | FoodForFun | Hoang Yen Minh | 30/09/2017 | 07/10/2017 | 5 | All of team | FINISH |
| 60 | Manage post | FoodForFun | Hoang Yen Minh | 30/09/2017 | 07/10/2017 | 5 | All of team | FINISH |
| 61 | Add tag | FoodForFun | Hoang Yen Minh | 30/09/2017 | 07/10/2017 | 5 | All of team | FINISH |
| 62 | Forgot password | FoodForFun | Hoang Yen Minh | 30/09/2017 | 07/10/2017 | 5 | All of team | FINISH |
| 63 | Insert Image | FoodForFun | Hoang Yen Minh | 30/09/2017 | 07/10/2017 | 5 | All of team | FINISH |
| 64 | Sub menu | FoodForFun | Hoang Yen Minh | 30/09/2017 | 07/10/2017 | 5 | All of team | FINISH |
| **Report** | | | | | | | | |
| 65 | reportPosts | FoodForFun | Hoang Yen Minh | 07/10/2017 | 14/10/2017 | 5 | Dao Nhat Minh | FINISH |
| 66 | reportUsers | FoodForFun | Hoang Yen Minh | 07/10/2017 | 14/10/2017 | 5 | Dao Nhat Minh | FINISH |
| 67 | reportCategories | FoodForFun | Hoang Yen Minh | 07/10/2017 | 14/10/2017 | 5 | Dao Nhat Minh | FINISH |
| 68 | reportTags | FoodForFun | Hoang Yen Minh | 07/10/2017 | 14/10/2017 | 5 | Dao Nhat Minh | FINISH |
| **Task sheet** | | | | | | | | |
| 69 | Task sheet | FoodForFun | Hoang Yen Minh | 07/10/2017 | 14/10/2017 | 5 | Nguyen Hoang Minh | FINISH |

**MONITORING REPORT**

**Unit Test**

# Login

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Author** | Dao Nhat Minh | | | | | |
| **Test case name** | Login | | | | | |
| **Actor** | Customer | | | | | |
| **Pre-requisite** |  | | | | | |
| **Process** | **Step No.** | **Action/Description** | **Input Data** | **Expected Result** | **Actual Result** | **Pass/Fail** |
| Login | 1 | From ‘Login’ page, left all textfield blank and then click on to the ‘Login’ button. |  | A message box appears with content: “Invalid username or password, try again!” |  |  |
| 2 | Enter UserName in ‘UserName’ textfield and then click ‘Login’ button. | UserName: abcde  Password: | A message box appears with content: “Please input username!” |  |  |
| 3 | Enter Password in ‘Password’ textfield and then click ‘Login’ button. | UserName:  Password: 123456 | A message box appears with content: “Please input password!” |  |  |
| 4 | Enter UserName in ‘UserName’ textfield, enter Password in ‘Password’ textfield (with UserName and Password of an Editor account) and then click ‘Login’ button. | UserName: editor01  Password: 123456 | Redirect to ‘Home’ page of Editor. |  |  |
| 5 | Enter UserName in ‘UserName’ textfield, enter Password in ‘Password’ textfield (with UserName and Password of an User account) and then click ‘Login’ button. | UserName: user01  Password: 123456 | Redirect to ‘List of Properties’ page. |  |  |
| 6 | Enter UserName in ‘UserName’ textfield, enter Password in ‘Password’ textfield (with UserName and Password of an Moderator account) and then click ‘Login’ button. | UserName: moderator01  Password: 123456 | Redirect to ‘List of Properties’ page. |  |  |

# Register

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Author** | Dao Nhat Minh | | | | | |
| **Test case name** | Register | | | | | |
| **Actor** | Customer | | | | | |
| **Pre-requisite** |  | | | | | |
| **Process** | **Step No.** | **Action/Description** | **Input Data** | **Expected Result** | **Actual Result** | **Pass/Fail** |
| Register | 1 | From ‘Home’ page, click on to the ‘Register’ button. |  | Redirect to ‘Register’ page. |  |  |
| 2 | From ‘Register’ page, left all textfield blank and then click on to the ‘Register’ button. |  | A validation message point to ‘UserName’ text field with content: “UserName cannot be blank” |  |  |
| 3 | Enter UserName in ‘UserName’ textfield and then click ‘Register’ button. | g0qc1201 | A validation message point to ‘Password’ text field with content: “Password cannot be blank” |  |  |
| 4 | Enter Name in ‘Name’ textfield and then click ‘Register’ button. | Ai Khoa | A validation message point to ‘Birthday’ datebox with content: “Birthday cannot be blank” |  |  |
| 5 | Enter Password in ‘Password’ textfield and then click ‘Register’ button. | 123456a | A validation message point to ‘ConFirm Password’ text field with content: “Password Confirm cannot be blank” |  |  |
| 6 | Enter Confirm Password in ‘ConFirm Password’ textfield and then click ‘Register’ button. | 123456a | A validation message point to ‘Birthday’ text field with content: “Birthday cannot be blank” |  |  |
| 7 | Enter Birthday in ‘Birthday’ textfield and then click ‘Register’ button. | 12/01/2001 | A validation message point to ‘Address’ text field with content: “Address cannot be blank” |  |  |
| 8 | Enter Address in ‘Address’ textfield and then click ‘Register’ button. | 555 CMT8 | A validation message point to ‘Phone’ text field with content: “Phone cannot be blank” |  |  |
| 9 | Enter Phone in ‘Phone’ textfield and then click ‘Register’ button. | 0123456789 | A validation message point to ‘Email’ text field with content: “Email cannot be blank” |  |  |
| 10 | Enter Email in ‘Email’ textfield and then click ‘Register’ button. | fff@gmail.com | A validation message point to ‘Experience’ label with content: “Experience cannot be blank” |  |  |
| 11 | Enter Experience in ‘Experience’ textfield and then click ‘Register’ button. | 5 | A validation message point to ‘Worked place’ label with content: “Worked place cannot be blank” |  |  |
| 12 | Enter Worked place in ‘Worked place’ textfield and then click ‘Register’ button. | Somewhere | The new User account, as a Moderator (default Role when not select Role in Role Radio Group), and Gender is Male (default Gender when not select Gender in Gender Radio Group) is added into the database, and a message box appears with content: “Your account has been created successfully!”. Then redirect to ‘Login’ page. |  |  |

# Accept post

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Author** | Dao Nhat Minh | | | | | |
| **Test case name** | Accept post | | | | | |
| **Actor** | Editor | | | | | |
| **Pre-requisite** |  | | | | | |
| **Process** | **Step No.** | **Action/Description** | **Input Data** | **Expected Result** | **Actual Result** | **Pass/Fail** |
| Accept post | 1 | From ‘Home’ page, click on to the ‘Login’ button. |  | Redirect to ‘Login’ page. |  |  |
| 2 | From ‘Login’ page, input username and pass word for editor and lick ‘Login’ button. | Username: editor01  Pass: 123456a | Redirect to ‘Editor’ page |  |  |
| 3 | Click ‘All post’ button  in ‘Post’ div at the middle right of website |  | List of all post will be shown at ‘list of all post’ page. |  |  |
| 4 | Choose ‘Waiting acceptance’ in the combobox at the top right corner of the website |  | List of all waiting post will be shown at ‘list of all waiting post’ page. |  |  |
| 5 | Click one post hyperlink |  | Details of that post will be shown. |  |  |
| 6 | Click ‘Publish’ button |  | Message “Published successfully!” will be shown and return to ‘list of all waiting post’ page. |  |  |
| 7 | Click ‘Decline’ button |  | Message “Declined successfully!” will be shown and return to ‘list of all waiting post’ page. |  |  |
| 8 | Click ‘Cancel’ button |  | Return to ‘list of all waiting post’ page. |  |  |

# Delete post

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Author** | Dao Nhat Minh | | | | | |
| **Test case name** | Delete post | | | | | |
| **Actor** | Editor | | | | | |
| **Pre-requisite** |  | | | | | |
| **Process** | **Step No.** | **Action/Description** | **Input Data** | **Expected Result** | **Actual Result** | **Pass/Fail** |
| Delete post | 1 | From ‘Home’ page, click on to the ‘Login’ button. |  | Redirect to ‘Login’ page. |  |  |
| 2 | From ‘Login’ page, input username and pass word for editor and lick ‘Login’ button. | Username: editor01  Pass: 123456a | Redirect to ‘Editor’ page |  |  |
| 3 | Click ‘All post’ button  in ‘Post’ div at the middle right of website |  | List of all post will be shown at ‘list of all post’ page. |  |  |
| 4 | Choose ‘Waiting acceptance’ in the combobox at the top right corner of the website |  | List of all waiting post will be shown at ‘list of all waiting post’ page. |  |  |
| 5 | Click one post hyperlink |  | Details of that post will be shown. |  |  |
| 6 | Click ‘Delete’ button |  | Message “Deleted successfully!” will be shown and return to ‘list of all waiting post’ page. |  |  |
| 7 | Click ‘Cancel’ button |  | Return to ‘list of all waiting post’ page. |  |  |

# Change password

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Author** | Dao Nhat Minh | | | | | |
| **Test case name** | Change password | | | | | |
| **Actor** | Editor, Moderator and member | | | | | |
| **Pre-requisite** |  | | | | | |
| **Process** | **Step No.** | **Action/Description** | **Input Data** | **Expected Result** | **Actual Result** | **Pass/Fail** |
| Delete post | 1 | From ‘Home’ page, click on to the ‘Login’ button. |  | Redirect to ‘Login’ page. |  |  |
| 2 | From ‘Login’ page, input username and pass word for editor and lick ‘Login’ button. | Username:  Editor01  Pass: 123456a  (example editor) | Redirect to ‘Editor’ page |  |  |
| 3 | From ‘Editor’ page click to ‘Change Password’ in Profile div at top left cornor |  | Change password form will be shown |  |  |
| 4 | Left ‘New password’ textfield and ‘Confirm new password’ textfield empty then chick ‘Save’ button |  | A validation message point to ‘New password’ text field with content: “New password can’t be blank!” |  |  |
| 5 | Input ‘New password’ textfield and left ‘Confirm new password’ textfield empty then chick ‘Save’ button | New password: 123456a | A validation message point to ‘Confirm new password’ text field with content: “Confirm new password can’t be blank!” |  |  |
| 6 | Input ‘New password’ textfield and ‘Confirm new password’ textfield then chick ‘Save’ button (New password != Confirm new password) | New password: 123456a  Confirm new password: 123456b | A validation message point to ‘Confirm new password’ text field with content: “Confirm new password must match with password!” |  |  |
| 7 | Input ‘New password’ textfield and ‘Confirm new password’ textfield then chick ‘Save’ button (New password == Confirm new password) | New password: 123456a  Confirm new password: 123456a | Message “Password is changed successfully!” was shown and return to ‘Editor’ page. |  |  |
| 8 | Click ‘Back to home’ button |  | Return to ‘Editor’ page |  |  |

1. **Update Profile**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Author** | Nguyen Hoang Minh | | | | | |
| **Test case name** | Update Profile | | | | | |
| **Actor** | Editor, Moderator and member | | | | | |
| **Pre-requisite** | Actor has already logged into the System as a Customer | | | | | |
| **Process** | **Step No.** | **Action/Description** | **Input Data** | **Expected Result** | **Actual Result** | **Pass/Fail** |
| Update profile | 1 | From ‘Home’ page, click on to the ‘Update Profile’ button. |  | Redirect to ‘Update Profile’ page, load all User profile information, and fill onto the form. |  |  |
| 1 | Clear the ‘Name’ text field, and then click ‘Update’ button. |  | A validation message point to ‘Name’ text field with content: “Name cannot be blank” |  |  |
| 1 | Select another birthday in the ‘Birthday’ datebox, and then click ‘Update’ button. | 01-01-2000 | A validation message point to ‘Birthday’ datebox with content: “Age must be over 18” |  |  |
| 1 | Clear the ‘Address’ text field, and then click ‘Update’ button. |  | A validation message point to ‘Address’ text field with content: “Address cannot be blank” |  |  |
| 1 | Clear the ‘Phone’ text field, and then click ‘Update’ button. |  | A validation message point to ‘Phone’ text field with content: “Phone cannot be blank” |  |  |
| 1 | Enter another Phone in ‘Phone’ textbox, and then click ‘Update’ button. | 123456 | A validation message point to ‘Phone’ text box with content: “Invalid phone number! Must be 8-11 digits.” |  |  |
| 1 | Clear the ‘Email’ text field, and then click ‘Update’ button. |  | A validation message point to ‘Email’ text field with content: “Email cannot be blank” |  |  |
| 1 | Enter another Email in ‘Email’ text field and then click ‘Save’ button. | minh@ | A validation message point to ‘Email’ text field with content: “Email is invalid” |  |  |
| 1 | Input valid all information and then click “Update” button. | Name: minh  Birthday: 01-01-1999  Gender: Male  Address: 1234 Le Van Si  Phone: 2534345434  Email: [labjavaweb@gmail.com](mailto:labjavaweb@gmail.com) | The User account is updated into the database, with the new fields:  - Name: Minh  - Birthday: 01-01-1999  - Gender is Male.  - Address is “1234 Le Van Si”.  - Phone is “2534345434”.  - Email is “[labjavaweb@gmail.com](mailto:labjavaweb@gmail.com)”.  And then a message box appears with content: “Your account has been updated!”. Then redirect to ‘Home’ page. |  |  |
| 1 | Click ‘Back to home’ button |  | Return to ‘Home’ page |  |  |

1. **Add comment**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Author** | Nguyen Hoang Minh | | | | | |
| **Test case name** | Add comment | | | | | |
| **Actor** | Editor, Moderator and member | | | | | |
| **Pre-requisite** | Actor has already logged into the System as a Customer, and a post is published. | | | | | |
| **Process** | **Step No.** | **Action/Description** | **Input Data** | **Expected Result** | **Actual Result** | **Pass/Fail** |
| Add comment | 1 | From ‘Post Details’ page, click on to the ‘Comment’ text area and then click ‘Comment’ button. |  | A validation message show on the page: “Comment cannot be blank” |  |  |
| 1 | Enter a comment details in ‘Comment’ text area and click ‘Comment’ button. | aaa | System save Comment into database then A message show on the page: “Comment successful” and reload ‘Post detail’ page. |  |  |
|  | 1 | Enter a comment details in ‘Comment’ text area and then click another button accept ‘Comment’ button | aaa | A confirm message show on the page: “Do you want to comment?” |  |  |
|  | 2  2 | User click ‘Yes’ button on the confirm page.  User click ‘No’ button on the confirm page. |  | System save Comment into database then A message show on the page: “Comment successful” and reload ‘Post detail’ page.  System clears the ‘Comment’ text area and then redirect to a page user clicked. |  |  |

1. **Update comment**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Author** | Nguyen Hoang Minh | | | | | |
| **Test case name** | Update comment | | | | | |
| **Actor** | Editor, Moderator and member | | | | | |
| **Pre-requisite** | Actor has already logged into the System as a Customer, a post is published and have commented already in the post. | | | | | |
| **Process** | **Step No.** | **Action/Description** | **Input Data** | **Expected Result** | **Actual Result** | **Pass/Fail** |
| Update comment | 1 | Click on to the ‘Update’ button in the user’s comment. |  | Redirect the comment in the ‘Comment’ text area. |  |  |
| 2 | Clear the comment details and click ‘Comment’ button |  | A validation message show on the page: “Comment cannot be blank” |  |  |
| 2 | Update a comment details in ‘Comment’ text area and click ‘Comment’ button. | aaabbbb | A message show on the page: “Update comment successful” and reload ‘Post detail’ page. |  |  |
| 2 | Update a comment details in ‘Comment’ text area and then click another button accept ‘Comment’ button | aaabbbb | A confirm message show on the page: “Do you want to comment?” |  |  |
| 3  3 | User click ‘Yes’ button on the confirm page.  User click ‘No’ button on the confirm page. |  | System save Comment into database then A message show on the page: “Comment successful” and reload ‘Post detail’ page.  System clears the ‘Comment’ text area and then redirect to a page user clicked. |  |  |

1. **Delete Comment**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Author** | Nguyen Hoang Minh | | | | | |
| **Test case name** | Delete comment | | | | | |
| **Actor** | Editor, Moderator and member | | | | | |
| **Pre-requisite** | Actor has already logged into the System as a Customer, a post is published and have commented already in the post. | | | | | |
| **Process** | **Step No.** | **Action/Description** | **Input Data** | **Expected Result** | **Actual Result** | **Pass/Fail** |
| Delete comment | 1 | Click on to the ‘Delete’ button in the user’s comment. |  | A confirm message show on the page: “Are you sure to detele the comment?” |  |  |
| 2 | Click ‘Yes’ button. |  | The system delete comment in database and, message show on the page: “Delete comment successful” and reload ‘Post detail’ page. |  |  |
| 2 | Click ‘No’ button |  | Redirect to: ‘Post detail’ page. |  |  |

1. **View post**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Author** | Nguyen Hoang Minh | | | | | |
| **Test case name** | View post | | | | | |
| **Actor** | Customer | | | | | |
| **Pre-requisite** |  | | | | | |
| **Process** | **Step No.** | **Action/Description** | **Input Data** | **Expected Result** | **Actual Result** | **Pass/Fail** |
| View post | 1 | From ‘Home’ page, user click ‘Article’ button in the main menu. |  | Redirect to ‘Article’ page with the following content in 1 box:  -‘Title’ lable  -‘Date’ lable  -‘Image’ lable  -‘Content’ lable  -‘See more’ button |  |  |
| 1 | From ‘Home’ page, user click ‘Pavement’ button in the ‘Home’ page. |  | Redirect to ‘Article’ page of Pavement post with the following content in 1 box:  -‘Title’ lable  -‘Date’ lable  -‘Image’ lable  -‘Content’ lable  -‘See more’ button |  |  |
| 1 | From ‘Home’ page, user click ‘Store’ button in the ‘Home’ page. |  | Redirect to ‘Article’ page of Store post with the following content in 1 box:  -‘Title’ lable  -‘Date’ lable  -‘Image’ lable  -‘Content’ lable  -‘See more’ button |  |  |
|  | 1 | From ‘Home’ page, user click ‘Restaurant’ button in the ‘Home’ page. |  | Redirect to ‘Article’ page of Restaurant post with the following content in 1 box:  -‘Title’ lable  -‘Date’ lable  -‘Image’ lable  -‘Content’ lable  -‘See more’ button |  |  |
|  | 1 | From ‘Home’ page, user click ‘See more’ button in the ‘Home’ page. |  | Redirect to ‘Article’ page with the following content in 1 box:  -‘Title’ lable  -‘Date’ lable  -‘Image’ lable  -‘Content’ lable  -‘See more’ button |  |  |

1. **View Post details**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Author** | Nguyen Hoang Minh | | | | | |
| **Test case name** | View post | | | | | |
| **Actor** | Customer | | | | | |
| **Pre-requisite** |  | | | | | |
| **Process** | **Step No.** | **Action/Description** | **Input Data** | **Expected Result** | **Actual Result** | **Pass/Fail** |
| Guest and Member View post | 1 | From ‘Article’ page, user click ‘Read more’ button at the under a post. |  | Redirect to ‘Article Details’ page with the following content in 1 box:  -‘Title’ lable  -‘Date’ lable  -‘Image’ lable  -‘Content’ lable  -‘Number of Like’ button  -‘Number of Comment’ button  -‘Comment’ text area  -‘Comment’ button |  |  |
| Moderator and Editor View post | 1 | From ‘Article’ page, user click ‘Read more’ button at the under a post. |  | Redirect to ‘Article Details’ page with the following content in 1 box:  -‘Title’ lable  -‘Date’ lable  -‘Image’ lable  -‘Content’ lable  -‘Number of Like’ button  -‘Number of Comment’ button  -‘View Post’, ‘Update’, ‘Trash’ button  -‘Comment’ text area  -‘Comment’ button |  |  |

1. **Add post**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Author** | Nguyen Hoang Minh | | | | | |
| **Test case name** | Add post | | | | | |
| **Actor** | Moderator and Editor | | | | | |
| **Pre-requisite** |  | | | | | |
| **Process** | **Step No.** | **Action/Description** | **Input Data** | **Expected Result** | **Actual Result** | **Pass/Fail** |
| Add post | 1 | From sub menu, user click ‘Add Post’ button |  | Redirect to ‘Add post’ page with the following content:  -‘Title’ text field  -‘Image’ button  -‘Content’ text area  -‘Place’s Address’ text field  -‘Phone’ text field  -‘Name’ text field  -‘Submit’ button  -‘Cancel’ button  -‘Category’ combo radio box and text field  -‘Tag’ text field |  |  |
|  | 2 | Left all field blank and click ‘Submit’ button. |  | A validation message point to ‘Title’ text field with content: “Title cannot be blank”  A validation message point to ‘Content’ text area with content: “Content cannot be blank”  A validation message point to ‘Place’s address’ text field with content: “Place’s address cannot be blank”  A validation message point to ‘Phone’ text field with content: “Phone cannot be blank”  A validation message point to ‘Name’ text field with content: “Name cannot be blank” |  |  |
|  | 2 | Click ‘Cancel’ button |  | System redirect to ‘Home’ page. |  |  |
|  | 3 | Input information in all text field and click ‘Image’ button | Title: Lab Web  Content: a  Place’s Address: a  Phone: 09010101010  Name: a | System redirect a form to insert a image. |  |  |
|  | 4 | Choose an image in ‘Choose Files’ button and click ‘Insert Image’ |  | System saves a link of file image in a form. |  |  |
|  | 5 | Click ‘Submit’ button |  | System save information into database and redirect a message “Add Post successful”. |  |  |
|  | 3 | Input information in all text field and click ‘Submit’ button | Title: Lab Web  Content: a  Place’s Address: a  Phone: 09010101010  Name: a | System save information into database and redirect a message “Add Post successful”. |  |  |

1. **Update post**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Author** | Nguyen Hoang Minh | | | | | |
| **Test case name** | Update post | | | | | |
| **Actor** | Moderator and Editor | | | | | |
| **Pre-requisite** |  | | | | | |
| **Process** | **Step No.** | **Action/Description** | **Input Data** | **Expected Result** | **Actual Result** | **Pass/Fail** |
| Update post | 1 | From ‘All Posts’ page, actor click on ‘Post Title’ hyperlink. |  | Redirect to ‘Update post’ page with the following content and filled:  -‘Title’ text field  -‘Image’ button  -‘Content’ text area  -‘Place’s Address’ text field  -‘Phone’ text field  -‘Name’ text field  -‘Submit’ button  -‘Cancel’ button  -‘Category’ combo radio box and text field  -‘Tag’ text field |  |  |
|  | 2 | Delete ‘Title’ text field blank and Click ‘Submit’ button |  | A validation message point to ‘Title’ text field with content: “Title cannot be blank” |  |  |
|  | 3 | Input in ‘Title’ text field and left ‘content’ text area blank. | Title: Lab Web | A validation message point to ‘Content’ text area with content: “Content cannot be blank” |  |  |
|  | 4 | Input in ‘Content’ text area and left ‘Place’s Address’ text field blank. | Content: a | A validation message point to ‘Place’s address’ text field with content: “Place’s address cannot be blank” |  |  |
|  | 5 | Input in ‘Place’s Address’ text field and left ‘Phone’ text field blank. | Place’s address: a | A validation message point to ‘Phone’ text field with content: “Phone cannot be blank” |  |  |
|  | 6 | Input in ‘Phone’ text field and left ‘Name’ text field blank. | Phone: 09010101010 | A validation message point to ‘Name’ text field with content: “Name cannot be blank” |  |  |
|  | 7 | Input ‘Name’ text field and click ‘Submit’ button | Name: a | System save information into database and redirect a message “Update Post successful”. |  |  |
|  | 8 | Left all field blank and click ‘Submit’ button. |  | A validation message point to ‘Title’ text field with content: “Title cannot be blank”  A validation message point to ‘Content’ text area with content: “Content cannot be blank”  A validation message point to ‘Place’s address’ text field with content: “Place’s address cannot be blank”  A validation message point to ‘Phone’ text field with content: “Phone cannot be blank”  A validation message point to ‘Name’ text field with content: “Name cannot be blank” |  |  |
|  | 8 | Click ‘Cancel’ button |  | System redirect to ‘Home’ page. |  |  |

1. **Delete post**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Author** | Nguyen Hoang Minh | | | | | |
| **Test case name** | Delete post | | | | | |
| **Actor** | Editor, Moderator | | | | | |
| **Pre-requisite** | Actor has already logged into the System as Editor or Moderator, a post is published. | | | | | |
| **Process** | **Step No.** | **Action/Description** | **Input Data** | **Expected Result** | **Actual Result** | **Pass/Fail** |
| Delete post | 1 | On the ‘Post Detail’ button, actor click ‘Delete’ button to delete a post. |  | A confirm message show on the page: “Are you sure to detele the post?” |  |  |
| 2 | Click ‘Yes’ button. |  | The system visible a post in database and, message show on the page: “Delete post successful” and reload ‘All Post’ page. |  |  |
| 2 | Click ‘No’ button |  | Redirect to: ‘All post’ page. |  |  |

1. **Add user**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Author** | Nguyen Hoang Minh | | | | | |
| **Test case name** | Add user | | | | | |
| **Actor** | Editor | | | | | |
| **Pre-requisite** | Actor has already logged into the System as Editor. | | | | | |
| **Process** | **Step No.** | **Action/Description** | **Input Data** | **Expected Result** | **Actual Result** | **Pass/Fail** |
| Add user | 1 | On the sub menu in Editor page, actor click ‘Add Editor’ button |  | Redirect to a ‘Add User’ page with the following:  -‘Username’ text field  -‘Name’ text field  -‘Password’ text field  -‘Confirm password’ text field  -‘Add Editor’ button |  |  |
| 2 | Actor click ‘Add Editor’ button |  | System redirect message point to ‘Username’ text field with message: “Username can’t be blank!”  Point to “Name” text field with message: “Name can’t be blank!”  Point to “Password” text field with message: “Password can’t be blank”  Point to “Confirm password” text field with message: “Confirm password can’t be blank!” |  |  |
| 2 | Actor input information and click ‘Add Editor’ button | Username: labweb  Name: minh  Password: aaa  Confirm password: aab | Redirect a message point to ‘Confirm’ text field with message: “Confirm password must be same with password” |  |  |
|  | 2 | Actor input information and click ‘Add Editor’ button | Username: labweb  Name: minh  Password: aaa  Confirm password: aaa | System save information into database and redirect a notify: “Add editor successful” |  |  |

1. **Add category**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Author** | Nguyen Hoang Minh | | | | | |
| **Test case name** | Add category | | | | | |
| **Actor** | Editor, Moderator | | | | | |
| **Pre-requisite** | Actor has already logged into the System as Editor or Moderator. | | | | | |
| **Process** | **Step No.** | **Action/Description** | **Input Data** | **Expected Result** | **Actual Result** | **Pass/Fail** |
| Add Category | 1 | On the sub menu, actor click ‘Category’ button to add category |  | A ‘Category’ page redirect to actor with the list of category at the right screen and the add catalog at the left with the following:  -‘Name’ text field  -‘Description’ text area  ‘Add Category’ button |  |  |
| 2 | Click ‘Add category’ button |  | The system show message and point to ‘Name’ text field with message: “Name can’t be blank”  The system show message and point to ‘Description’ text area with message: “Description can’t be blank” |  |  |
| 2 | Input information at ‘Name’ text field and left ‘Description’ blank and click ‘Add category’ button | Name: bun bo | The system show message and point to ‘Description’ text area with message: “Description can’t be blank” |  |  |
| 2 | Input information at ‘Description’ text area and left ‘Name’ blank and click ‘Add category’ button |  | The system show message and point to ‘Name’ text field with message: “Name can’t be blank” |  |  |
| 2 | Input information and click ‘Add category’ button | Name: bun bo  Description: an bun bo | System save into database and return a message: “Add category successful” |  |  |

1. **Update Category**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Author** | Nguyen Hoang Minh | | | | | |
| **Test case name** | Update Category | | | | | |
| **Actor** | Editor, Moderator | | | | | |
| **Pre-requisite** | Actor has already logged into the System as Editor or Moderator, a category already exit. | | | | | |
| **Process** | **Step No.** | **Action/Description** | **Input Data** | **Expected Result** | **Actual Result** | **Pass/Fail** |
| Update category | 1 | On the sub menu, actor click ‘Category’ button. |  | A ‘Category’ page redirect to actor with the list of category at the right screen. |  |  |
| 2 | Click ‘Edit’ button on the Category |  | The system redirect a edit category form with the following content:  -‘Name’ text field  -‘Description’ text area  -‘Update’ button |  |  |
| 3 | Click ‘Update’ button |  | The system show message and point to ‘Name’ text field with message: “Name can’t be blank”  The system show message and point to ‘Description’ text area with message: “Description can’t be blank” |  |  |
| 3 | Input information at ‘Name’ text field and left ‘Description’ blank and click ‘Update’ button | Name: bun bo | The system show message and point to ‘Description’ text area with message: “Description can’t be blank” |  |  |
| 3 | Input information at ‘Description’ text area and left ‘Name’ blank and click ‘Update’ button | Description: an bun bo | The system show message and point to ‘Name’ text field with message: “Name can’t be blank” |  |  |
| 3 | Input information and click ‘Update’ button | Name: bun bo  Description: an bun bo | System save into database and return a message: “Update successful” and redirect to ‘Category’ page |  |  |

1. **Delete category**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Author** | Nguyen Hoang Minh | | | | | |
| **Test case name** | Delete Category | | | | | |
| **Actor** | Editor, Moderator | | | | | |
| **Pre-requisite** | Actor has already logged into the System as Editor or Moderator, a category already exit. | | | | | |
| **Process** | **Step No.** | **Action/Description** | **Input Data** | **Expected Result** | **Actual Result** | **Pass/Fail** |
| Delete Catefory | 1 | On the sub menu, actor click ‘Category’ button. |  | A ‘Category’ page redirect to actor with the list of category at the right screen. |  |  |
| 2 | Click ‘Delete’ button on one Category |  | The system redirect a confirm form with message: “Are you sure you want to delete this Category” |  |  |
| 3 | Click ‘Yes’ button |  | System redirect to ‘Category’ page with a message: “Delete successful” |  |  |
|  | 3 | Click ‘No’ button |  | System redirect to ‘Category’ page |  |  |

1. **Add tag**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Author** | Nguyen Hoang Minh | | | | | |
| **Test case name** | Add tag | | | | | |
| **Actor** | Editor, Moderator | | | | | |
| **Pre-requisite** | Actor has already logged into the System as Editor or Moderator. | | | | | |
| **Process** | **Step No.** | **Action/Description** | **Input Data** | **Expected Result** | **Actual Result** | **Pass/Fail** |
|  | 1 | On the sub menu, actor click ‘Tag’ button to add category |  | A ‘Tag’ page redirect to actor with the list of tag at the right screen and the add tag at the left with the following:  -‘Name’ text field  -‘Description’ text area  -‘Add Tag’ button |  |  |
|  | 2 | Click ‘Add Tag’ button |  | The system show message and point to ‘Name’ text field with message: “Name can’t be blank”  The system show message and point to ‘Description’ text area with message: “Description can’t be blank” |  |  |
|  | 2 | Input information at ‘Name’ text field and left ‘Description’ blank and click ‘Add Tag’ button | Name: via he | The system show message and point to ‘Description’ text area with message: “Description can’t be blank” |  |  |
| Add tag | 2 | Input information at ‘Description’ text area and left ‘Name’ blank and click ‘Add Tag’ button | Description: an vat hia he | The system show message and point to ‘Name’ text field with message: “Name can’t be blank” |  |  |
| 2 | Input information and click ‘Add Tag’ button | Name: via he  Description: an vat via he | System save into database and return a message: “Add Tag successful” |  |  |

1. **Update tag**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Author** | Nguyen Hoang Minh | | | | | |
| **Test case name** | Update tag | | | | | |
| **Actor** | Editor, Moderator | | | | | |
| **Pre-requisite** | Actor has already logged into the System as Editor or Moderator, a tag already exit. | | | | | |
| **Process** | **Step No.** | **Action/Description** | **Input Data** | **Expected Result** | **Actual Result** | **Pass/Fail** |
|  | 1 | On the sub menu, actor click ‘Tag’ button. |  | A ‘Tag’ page redirect to actor with the list of tag at the right screen. |  |  |
|  | 2 | Click ‘Edit’ button on the Tag |  | The system redirect a edit tag form with the following content:  -‘Name’ text field  -‘Description’ text area  -‘Update’ button |  |  |
|  | 3 | Click ‘Update’ button |  | The system show message and point to ‘Name’ text field with message: “Name can’t be blank”  The system show message and point to ‘Description’ text area with message: “Description can’t be blank” |  |  |
| Update Tag | 3 | Input information at ‘Name’ text field and left ‘Description’ blank and click ‘Update’ button | Name: via he | The system show message and point to ‘Description’ text area with message: “Description can’t be blank” |  |  |
| 3 | Input information at ‘Description’ text area and left ‘Name’ blank and click ‘Update’ button | Description: an vat via he | The system show message and point to ‘Name’ text field with message: “Name can’t be blank” |  |  |
| 3 | Input information and click ‘Update’ button | Name: via he  Description: an vat via he | System save into database and return a message: “Update successful” and redirect to ‘Tag’ page |  |  |

1. **Delete tag**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Author** | Nguyen Hoang Minh | | | | | |
| **Test case name** | Delete post | | | | | |
| **Actor** | Editor, Moderator | | | | | |
| **Pre-requisite** | Actor has already logged into the System as Editor or Moderator, a tag already exit. | | | | | |
| **Process** | **Step No.** | **Action/Description** | **Input Data** | **Expected Result** | **Actual Result** | **Pass/Fail** |
| Delete tag | 1 | On the sub menu, actor click ‘Tag’ button. |  | A ‘Tag’ page redirect to actor with the list of tag at the right screen. |  |  |
| 2 | Click ‘Delete’ button on one Tag |  | The system redirect a confirm form with message: “Are you sure you want to delete this Tag” |  |  |
| 3 | Click ‘Yes’ button |  | System redirect to ‘Tag’ page with a message: “Delete successful” |  |  |
| 3 | Click ‘No’ button |  | System redirect to ‘Tag’ page |  |  |

1. **Forgot password**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Author** | Nguyen Hoang Minh | | | | | |
| **Test case name** | Forgot password | | | | | |
| **Actor** | Editor, Moderator, Member | | | | | |
| **Pre-requisite** | Actor has account in the database. | | | | | |
| **Process** | **Step No.** | **Action/Description** | **Input Data** | **Expected Result** | **Actual Result** | **Pass/Fail** |
| Forgot password | 1 | On the login page, actor click on the ‘Forgot password’ hyperlink |  | System redirect to a ‘Forgot password’ page with following content:  -‘Email’ text field  -‘Reset Password’ button |  |  |
| 2 | Click ‘Reset password’ button |  | System redirect a error message point to ‘Email’ text field with message: “Email can’t be blank” |  |  |
| 2 | Input your email and click ‘Reset password’ button | Email: labweb@gmail.com | System send actor password to the email that user input and return a ‘Login’ page. |  |  |

1. **Ban user**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Author** | Nguyen Hoang Minh | | | | | |
| **Test case name** | Ban user | | | | | |
| **Actor** | Editor | | | | | |
| **Pre-requisite** | Actor has already logged into the System as Editor. | | | | | |
| **Process** | **Step No.** | **Action/Description** | **Input Data** | **Expected Result** | **Actual Result** | **Pass/Fail** |
| Ban user | 1 | On the sub menu, actor click on the ‘All user’ button. |  | System redirect to ‘All user’ page with a user list and ‘Delete’ button in each user. |  |  |
| 2 | Click ‘Delete’ button. |  | System redirect a confirm page with message “Are you sure to delete user?” |  |  |
| 3 | Click ‘Yes’ button |  | System visible user into database and redirect to ‘All user’ page with message: “Delete user successful”. |  |  |
| 3 | Click ‘No’ button |  | Redirect to: ‘All user’ page. |  |  |