Group Report

# COMP 1640

**Enterprise Web Software Development**

**Supervisor: Ho Nguyen Phu Bao**

Team Member:

1. Bui An Quoc Quan
2. Nguyen Tan Dat
3. Huynh Anh Tuan (Leader)
4. Dinh Quang Duy
5. Dinh Tri Thach
6. Nguyen Tran Dang Quang

Table of Contents

1. [Project Introduction 4](#_bookmark0)
2. [Database 4](#_bookmark1)
3. [Functionality 5](#_bookmark2)
   1. [Product Backlogs 5](#_bookmark3)
      * [Admin](#_bookmark4) [5](#_bookmark4)
      * [Student:](#_bookmark5) [5](#_bookmark5)
      * [Marketing Coordinator](#_bookmark6) [5](#_bookmark6)
      * [Marketing Manager](#_bookmark7) [5](#_bookmark7)
      * [Guest:](#_bookmark8) [5](#_bookmark8)
   2. [Use Case Diagram: 6](#_bookmark9)
      * [Admin](#_bookmark10) [6](#_bookmark10)
      * [Student:](#_bookmark11) [6](#_bookmark11)
      * [Marketing Coordinator](#_bookmark12) [7](#_bookmark12)
      * [Marketing Manager](#_bookmark13) [7](#_bookmark13)
      * [Guest:](#_bookmark14) [7](#_bookmark14)
      * [Full roles](#_bookmark15) [8](#_bookmark15)
   3. [Use Case Specification 9](#_bookmark16)

[Use Case1: Login 9](#_bookmark17)

[Use Case 2: Create account 10](#_bookmark18)

[Use Case 3: Edit Account 11](#_bookmark19)

[Use Case 4: Delete account 12](#_bookmark20)

[Use Case 5: Create topic 13](#_bookmark21)

[Use Case 6: Edit student information 14](#_bookmark22)

[Use Case 7: Submit contributions 15](#_bookmark23)

[Use Case 8: Edit and Resubmit Contributions 16](#_bookmark24)

[Use Case 9: MC Comment (Marketing coordinator) 17](#_bookmark25)

[Use Case 10: Statistics for the number of contributions 18](#_bookmark26)

[Use Case 11: Read the contributions (Guest) 19](#_bookmark27)

[Use Case 12: Download the contributions (Marketing Manager) 20](#_bookmark28)

[Use Case 13: Read the contributions (MM) 21](#_bookmark29)

[Use Case 14: Guest 22](#_bookmark30)

* 1. [Activity Diagram: 23](#_bookmark31)
* [Student submit the contribution](#_bookmark32) [23](#_bookmark32)
* [Marketing Coordinator commented on the student's contribution](#_bookmark33) [24](#_bookmark33)
* [Admin create topic for the student to submit their contribution](#_bookmark34) [25](#_bookmark34)
* [Marketing Manager watch all the contribution and download them:](#_bookmark35) [26](#_bookmark35)
* [Login](#_bookmark36) [26](#_bookmark36)
  1. [Sequence Diagram: 27](#_bookmark37)
* [Student submit the contribution](#_bookmark38) [27](#_bookmark38)
* [Marketing Coordinator commented on the student’s contribution](#_bookmark39) [27](#_bookmark39)
* [Admin create topic for the student to submit their contribution](#_bookmark40) [28](#_bookmark40)
* [Marketing Manager watch all the contribution and download them:](#_bookmark41) [28](#_bookmark41)
* [Login](#_bookmark42) [28](#_bookmark42)

1. [Design 29](#_bookmark43)

* [Login](#_bookmark44) [29](#_bookmark44)
* [Student Dashboard](#_bookmark45) [29](#_bookmark45)
* [Student Information](#_bookmark46) [30](#_bookmark46)
* [Student Submit:](#_bookmark47) [30](#_bookmark47)
* [Admin Dashboard](#_bookmark48) [31](#_bookmark48)
* [Admin Create Topic](#_bookmark49) [31](#_bookmark49)
* [Marketing Coordinator Dashboard](#_bookmark50) [31](#_bookmark50)
* [Marketing Coordinator Comment:](#_bookmark51) [32](#_bookmark51)
* [Marketing Coordinator Count Articles](#_bookmark52) [32](#_bookmark52)
* [Marketing Manager Dashboard](#_bookmark53) [33](#_bookmark53)
* [Guest Dashboard](#_bookmark54) [33](#_bookmark54)

1. [Testing 34](#_bookmark55)

* [Test Plan](#_bookmark56) [34](#_bookmark56)
* [Test Log](#_bookmark57) [35](#_bookmark57)

1. [Agile Method 39](#_bookmark58)

* [Burndown Charts](#_bookmark59) [40](#_bookmark59)

1. [Product: 41](#_bookmark60)
2. [Links: 47](#_bookmark61)

# Project Introduction:

In today's life, English has become an indispensable part of our surroundings. Recognize this and want to improve students' English writing skills in school, and look for contributions to contribute to university magazines. Now we decided to build a web system to collect these contributed contributions. This system will provide a place for a Student to submit contributions, the Marketing Coordinator will manage the quality of the posts, the Marketing Director will manage the download of quality contributions and remove them from the system, The administrator will be the one who selects the topic for student to write the contributions, and a guest account will only be able to view the selected contributions.

# Database:

Our whole team discussed together and was decided that the system would consist of three main boards: **User, Group, Articles.** We’ll talk about the relationship between these tables. The first one is the relationship between **User** and **Group,** one user can join multiple and one group can have a lot of user so we’ve many-to-many relationship, it’ll create a new table with both primary key from **User** and **Group.** The next one is **Group** and **Articles,** one group can have a lot of articles and one articles can only have in one group so we’ve one-to-many relationship. And the last one is **User** and **Articles,** one user can have a lot of articles and one articles can only from one user so we’ve one-to- many relationship.

# Functionality:

# Product Backlogs:

We have 5 main types of accounts in this system: Admin, Student, Marketing Coordinator, Marketing Manager, Guest.

# Admin:

* + - * Create accounts for Student, Marketing Coordinator, Marketing Manager and Guest
      * Create Topic for Student to write the contributions
      * Maintains system data

# Student:

* + - * Students can submit one or more contributions in Word format
      * High-quality pictures can be uploaded.
      * Must agree to the terms and conditions before submitting.
      * Will be corrected when it is not overdue.
      * Will be not able to add new contributions and edit contributions when they are overdue.

# Marketing Coordinator:

* + - * Will be notified by Email when the contribution is sent to the system.
      * Must be made to comment within 14 days.
      * Ability to interact with students to edit and select contributions to publish.
      * Only interact with faculty students managed by themselves
      * Statistically check the number of contributions contributed by students.

# Marketing Manager:

* + - * The selected article can be viewed but not edited
      * The contributions can be downloaded after submitting them in ZIP format to transfer out of the system.

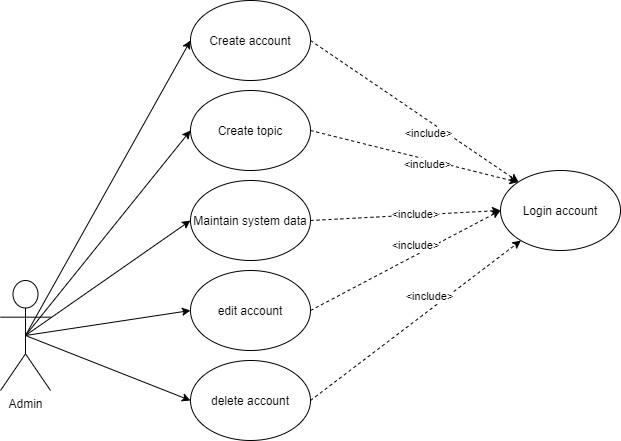
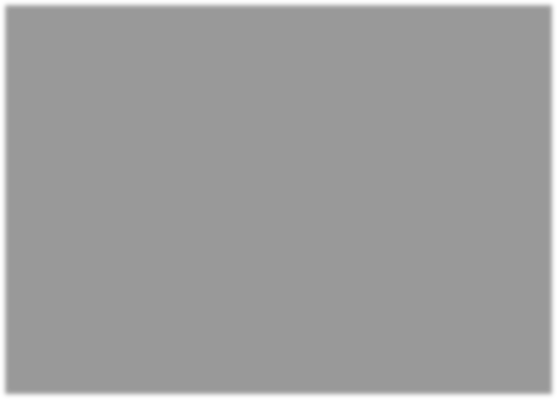
# Guest:

* + - * View selected contributions.

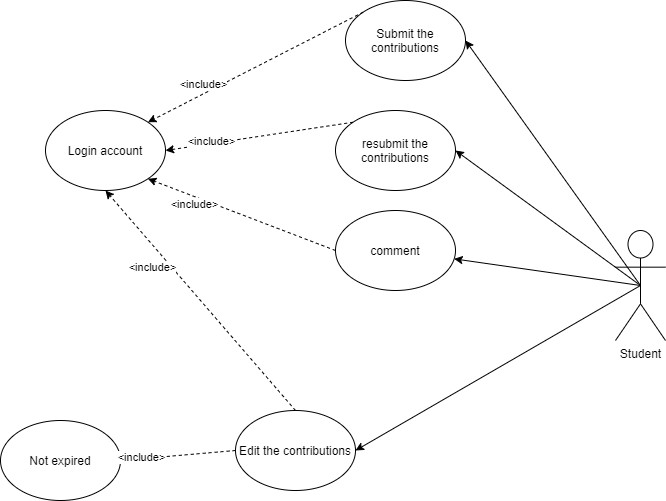
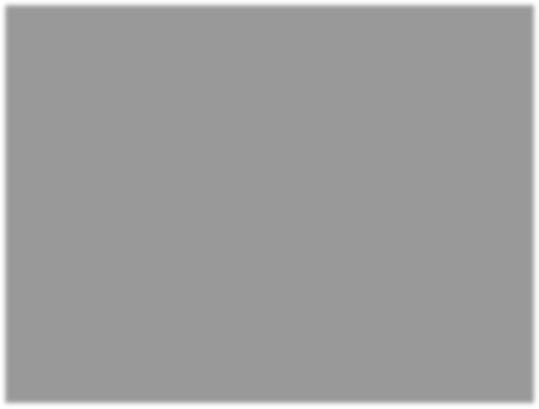
In addition, the common point of the above accounts is to log in to a registered Gmail account in order to use this system. Not only that, this system must be suitable for different interfaces such as laptops, phones, ...

# Use Case Diagram:

# Admin:

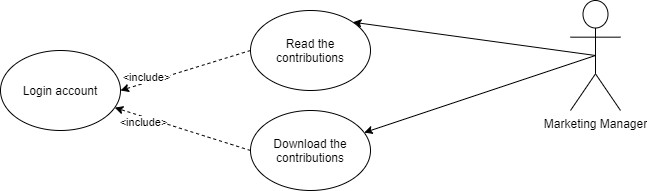
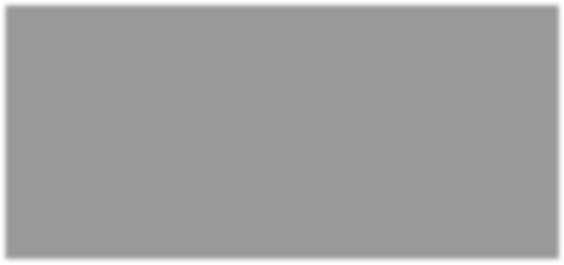


# Student:

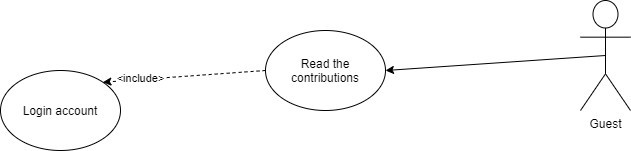
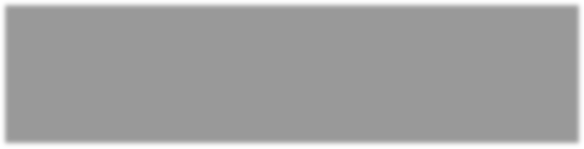


# Marketing Coordinator:

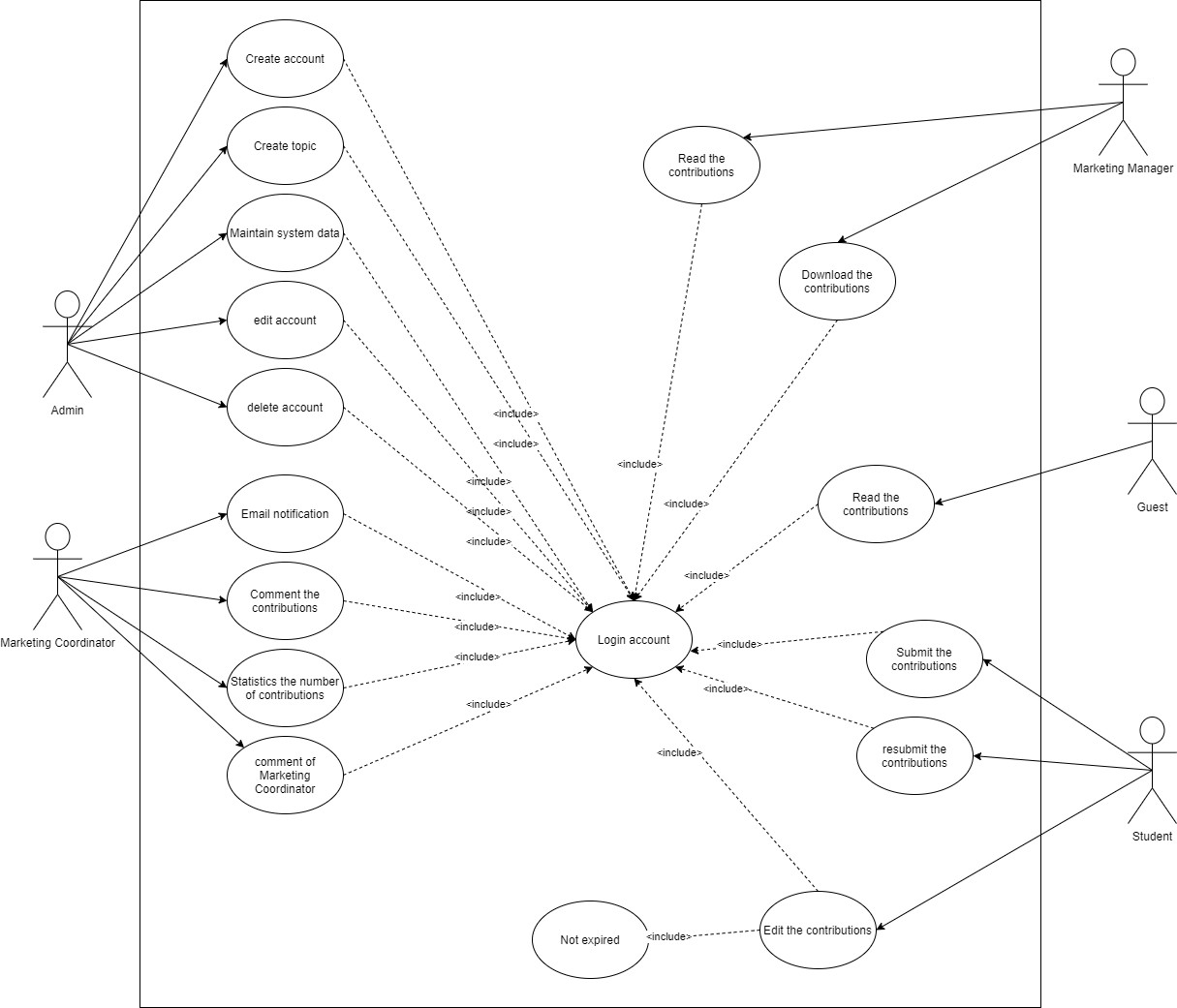
# Marketing Manager:



# Guest:



# Full roles:



* 1. **Use Case Specification:** **Use Case1: Login**

|  |  |  |  |
| --- | --- | --- | --- |
| **Use case:** Login | | **ID:** UC01 | **Priority:** High |
| **Description:** This use case describe how user can login to this system | | | |
| **Trigger:** User want to login to this system | | | |
| **Actor**: all | **Type:** Internal | | |
| **Pre-conditions**   * User account has been created * User account has been authorized * The user's device was connected to the internet when logging in | | | |
| **Post-conditions**   * The user has successfully logged in to the application * System displays the interface according to the authority | | | |
| **Basic flow**   1. User accesses to the website of this system 2. User must be logged in with an account created by admin or login by Admin account 3. Users input account and password 4. The system validates login information successfully and allows users to access the system 5. System displays the interface according to the authority | | | |
| **Alternative flow**   1. User accesses to the website of the system 2. User chooses login by Gmail 3. The system switches to the Google login screen 4. User inputs his/her Google account 5. From this step, the flow is the same the basic flow | | | |
| **Exception flow**   1. The system validates login information unsuccessfully and doesn’t allow users to access the system 2. User stop login request | | | |
| **Includes:** None | | | |
| **Business rules:** Allow users login incorrectly maximum 3 times | | | |

# Use Case 2: Create account

|  |  |  |  |
| --- | --- | --- | --- |
| **Use case:** create Admin account | | **ID:** UC02 | **Priority:** High |
| **Description:** This use case describe how user can create new account | | | |
| **Trigger:** User want to create account. | | | |
| **Actor**: Admin | **Type:** Internal | | |
| **Pre-conditions**   * User had to login to the system with admin role already * This new account has to not be in the database of the system * User has to fill enough information that is required in creation form | | | |
| **Post-conditions**   * The system displays on the screen: the account has been created successfully * The system stores this account's information in the database | | | |
| **Basic flow**   1. Administrator access to this system. 2. Administrator clicks on the "create account" tab. 3. Administrator fills in all the required information in the blanks of the creation form and chooses the role for each account 4. The system verifies that the account information matches any existing account in the database and inserts it into the database. 5. The system notifies the administrator that this account has been created | | | |
| **Alternative flow: None** | | | |
| **Exception flow:** This account is already available in the database | | | |
| **Includes:** Login | | | |
| **Business rules:** Require user fill all the blanks of add forms | | | |

# Use Case 3: Edit Account

|  |  |  |  |
| --- | --- | --- | --- |
| **Use case:** Edit account | | **ID:** UC03 | **Priority:** High |
| **Description:** This use case describe how Admin edit account | | | |
| **Trigger:** Administrator need to edit an account | | | |
| **Actor**: Admin | **Type:** Internal | | |
| **Pre-conditions**   * User had to login into the system with admin role already * Accounts that need to be edited must be in the system database | | | |
| **Post-conditions**   * System notifies to user that the account’s information was edited successfully * System change new information for that account in database | | | |
| **Basic flow**   * User accesses to this system as Administrators * admin navigate to the account you need to edit, click on (...) on the "Other" tab then select "Edit account" * system will display the information of this account in the form * admin has the authorities to change "account name", "Email", "Role" * System edit information of this account in database * System notifies to Admin about that account was edited successfully and display new information in list | | | |
| **Alternative flow: None** | | | |
| **Exception flow:** Admin cancel edit account | | | |
| **Includes:** Login | | | |
| **Business rules:**   * Require user fill all the blanks of add forms * The new data has to have the same data type of the old data | | | |

# Use Case 4: Delete account

|  |  |  |  |
| --- | --- | --- | --- |
| **Use case:** Delete account | | **ID:** UC04 | **Priority:** High |
| **Description:** This use case describe how Admin delete account | | | |
| **Trigger:** Administrator want to delete an account | | | |
| **Actor**: Admin | **Type:** Internal | | |
| **Pre-conditions**   * User had to login into the system with admin role already * Accounts that need to be deleted must be in the system database | | | |
| **Post-conditions**   * The system notifies the user that the account has been successfully deleted * The system has deleted the account from the database | | | |
| **Basic flow**   * User accesses to this system as Administrators * admin navigate to the account you need to delete, click (...) on the "Other" tab then select “Delete account" * system will display "are you sure to delete this account" * admin choose "YES" to delete that account * System delete account from database * System notifies to Admin about that account was deleted successfully and does not display account in list | | | |
| **Alternative flow: None** | | | |
| **Exception flow:** Admin click “NO” to cancel delete account | | | |
| **Includes:** Login | | | |
| **Business rules:** User has to confirm twice to delete the account | | | |

# Use Case 5: Create topic

|  |  |  |  |
| --- | --- | --- | --- |
| **Use case:** Create topic | | **ID:** UC05 | **Priority:** High |
| **Description:** This use case describe how Admin create new topic | | | |
| **Trigger:** Administrator want to create a topic | | | |
| **Actor**: Admin | **Type:** Internal | | |
| **Pre-conditions**   * User had to login into the system with admin role already * Topic name had not been stored in database of the system before * Admin must have enough information for the topic to create | | | |
| **Post-conditions**   * System displays on the screen: Topic has been created successfully * System stores topic in database * System display topic information in list | | | |
| **Basic flow**   * User accesses to this system as Administrators * admin search and click on “Create topic” button * system will display "Create topic" form * admin fill enough required information into the blanks of creation form and choose schedule * Click “create” to finish * System will add new topic into database * System display new database in list | | | |
| **Alternative flow: None** | | | |
| **Exception flow:** Admin cancel create topic | | | |
| **Includes:** Login | | | |
| **Business rules:** Require user fill all the blanks of add form | | | |

# Use Case 6: Edit student information

|  |  |  |  |
| --- | --- | --- | --- |
| **Use case:** Edit student account | | **ID:** UC06 | **Priority:** High |
| **Description:** This use case describe how student edit their account | | | |
| **Trigger:** Student need to change their information | | | |
| **Actor**: Student | **Type:** Internal | | |
| **Pre-conditions**   * Students already have an account in the system * User had to login into the system with student role already | | | |
| **Post-conditions**   * System notifies to student that their accounts was edited successfully * System change new information for their account in database | | | |
| **Basic flow**   * User accesses to this system as student role * Student find and click to "Edit" button * System will display new form to edit, student can edit their "Email" and "Password" * Student fill new information to blank space * System will edit information of their account in database and display new information in list | | | |
| **Alternative flow: None** | | | |
| **Exception flow:**   * Student cancel edit account * There is no change of information between old and new accounts | | | |
| **Includes:** Login | | | |
| **Business rules:**   * Require user fill all the blanks of add forms * The new data has to have the same data type of the old data | | | |

# Use Case 7: Submit contributions

|  |  |  |  |
| --- | --- | --- | --- |
| **Use case:** Submit contributions | | **ID:** UC07 | **Priority:** High |
| **Description:** This use case describe how student submit their contributions | | | |
| **Trigger:** Student need to submit contributions | | | |
| **Actor**: Student | **Type:** Internal | | |
| **Pre-conditions**   * Students already have contributions to submit * User had to login into the system with student role already | | | |
| **Post-conditions**   * System notifies to student that their contributions was submitted successfully * System will save their contributions in database * Display contributions in list | | | |
| **Basic flow**   * User accesses to this system as student role * Click “click to upload” to upload student article in Submit Contributions form * Click “Submit” if student upload successful * System will display “rule” form, student agree all rule to submit contributions * System will save that information in database and display in “submit contributions” page | | | |
| **Alternative flow: None** | | | |
| **Exception flow:**   * Empty Contributions * Student cancel upload Contributions * Contribution’s file are too large | | | |
| **Includes:** Edit and Resubmit Contributions | | | |
| **Business rules:**   * Require user fill all of submit forms * The data entered must be the same as the specified data type | | | |

# Use Case 8: Edit and Resubmit Contributions

|  |  |  |  |
| --- | --- | --- | --- |
| **Use case:** Edit & Resubmit contributions | | **ID:** UC08 | **Priority:** High |
| **Description:** This use case describe how student edit & resubmit their contributions | | | |
| **Trigger:** Student need to change their contributions | | | |
| **Actor**: Student | **Type:** Internal | | |
| **Pre-conditions**   * User had to login into the system with student role already * Students already have contributions on system | | | |
| **Post-conditions**   * System notifies to student that their contributions was changed successfully * System will save their new contributions in database * Display new contributions in list | | | |
| **Basic flow**   * User accesses to this system as student role * Find and delete file that student submitted before * Upload or drop new Contribution’s file * Click “Submit” if student upload successful * System will display “rule” form, student agree all rule to submit contributions * System will save new contributions in database and display in “submit contributions” page | | | |
| **Alternative flow: None** | | | |
| **Exception flow:**   * Empty Contributions * Student cancel upload Contributions * Contribution file are too large * There is no changes are made between new and old files | | | |
| **Includes:** none | | | |
| **Business rules:** The data entered must be the same as the specified data type | | | |

# Use Case 9: MC Comment (Marketing coordinator)

|  |  |  |  |
| --- | --- | --- | --- |
| **Use case:** Comment | | **ID:** UC09 | **Priority:** High |
| **Description:** This use case describe how MC comment student contributions | | | |
| **Trigger:** Marketing manager want to comment student contributions | | | |
| **Actor**: MC | **Type:** Internal | | |
| **Pre-conditions**   * User had to login into the system with MC role already * Students already submit contributions on system | | | |
| **Post-conditions**   * System notifies to student that their contributions already have comment * System notifies to MM that their comment was successful | | | |
| **Basic flow**   * User accesses to this system as MC role * Find and click to “comment” button in the contributions that need to comment * System will display “Comment” form * Submit an MC's comment in the blank space and click “Create” * System will save comments and send notifications to students | | | |
| **Alternative flow: None** | | | |
| **Exception flow: None** | | | |
| **Includes:** Login | | | |
| **Business rules:** Require user fill all of comment forms | | | |

# Use Case 10: Statistics for the number of contributions

|  |  |  |  |
| --- | --- | --- | --- |
| **Use case:** statistics the number of contributions | | **ID:** UC10 | **Priority:** High |
| **Description:** This use case describe how MC statistics the number of contributions | | | |
| **Trigger:** MC want to statistics the number of contributions | | | |
| **Actor**: MC | **Type:** Internal | | |
| **Pre-conditions**   * User had to login into the system with MC role already * Students already submit contributions on system * There are more than 2 contributions on the database | | | |
| **Post-conditions**   * System notifies to MC that the data has been statistically successful | | | |
| **Basic flow**   * User accesses to this system as Marketing Manager role * Find and click to “Statistics” button in the contributions that need to statistics * Click “Yes” in new form to finish statistics * System automatically statistics and send data to MC | | | |
| **Alternative flow: None** | | | |
| **Exception flow:**   * No contributions on database * MM click “No” to cancel action | | | |
| **Includes:** Login | | | |
| **Business rules:** None | | | |

# Use Case 11: Read the contributions (Guest)

|  |  |  |  |
| --- | --- | --- | --- |
| **Use case:** Read contributions | | **ID:** UC11 | **Priority:** High |
| **Description:** This use case describes how Guest read student contributions | | | |
| **Trigger:** Guest want to read student contributions | | | |
| **Actor**: Guest | **Type:** Internal | | |
| **Pre-conditions**   * User had to login into the system with Guest role already | | | |
| **Post-conditions**   * Notice of successful login * Display contributions form | | | |
| **Basic flow**   * User accesses to this system as Guest role * Find contributions in list an click to the contributions that user want to read * System will display article | | | |
| **Alternative flow: None** | | | |
| **Exception flow:**   * Contributions has no marks * Guest cannot login to the system | | | |
| **Includes:** Login | | | |
| **Business rules:** None | | | |

# Use Case 12: Download the contributions (Marketing Manager)

|  |  |  |  |
| --- | --- | --- | --- |
| **Use case:** Download contributions | | **ID:** UC12 | **Priority:** High |
| **Description:** This use case describes how Marketing download student contributions | | | |
| **Trigger:** MM want to read student contributions | | | |
| **Actor**: MM | **Type:** Internal | | |
| **Pre-conditions**   * User had to login into the system with MM role already * There are contributions available on the database | | | |
| **Post-conditions**   * Display notification that the contributions have been download successfully * Display contributions page | | | |
| **Basic flow**   * User accesses to this system as MM role * Find contributions in list and click “download” the contributions that user want to download * System will download ZIP file * System will automatically delete contributions from the list * Back to “contributions” page | | | |
| **Alternative flow: None** | | | |
| **Exception flow:**   * No contributions on database * MM click “X” to cancel download | | | |
| **Includes:** Login | | | |
| **Business rules:** None | | | |

# Use Case 13: Read the contributions (MM)

|  |  |  |  |
| --- | --- | --- | --- |
| **Use case:** Read contributions | | **ID:** UC13 | **Priority:** High |
| **Description:** This use case describes how MM read student contributions | | | |
| **Trigger:** MM want to read student contributions | | | |
| **Actor**: MM | **Type:** Internal | | |
| **Pre-conditions**   * User had to login into the system with MM role already | | | |
| **Post-conditions**   * Notice of successful login * Display contributions page | | | |
| **Basic flow**   * User accesses to this system as MM role * Find contributions in list and click to the contributions that user want to read * System will display article | | | |
| **Alternative flow: None** | | | |
| **Exception flow:**   * MM cannot login to the system | | | |
| **Includes:** Login | | | |
| **Business rules:** None | | | |

Use Case 14: Guest

|  |  |  |  |
| --- | --- | --- | --- |
| **Use case:** Read contributions | | **ID:** UC14 | **Priority:** low |
| **Description:** This use case describes how Guest read student contributions | | | |
| **Trigger:** Guest want to read student contributions | | | |
| **Actor**: Guest | **Type:** Internal | | |
| **Pre-conditions**   * User had to login into the system with Guest role already | | | |
| **Post-conditions**   * Notice of successful login * Display articles form | | | |
| **Basic flow**   * User accesses to this system as Guest role * Find articles in list and click to the contributions that user want to read * System will display article | | | |
| **Alternative flow: None** | | | |
| **Exception flow:**   * Articles has no marks * Guest cannot login to the system | | | |
| **Includes:** Login | | | |
| **Business rules:** None | | | |

# Activity Diagram:

We have 4 main flows for our system: + Student submit the contribution

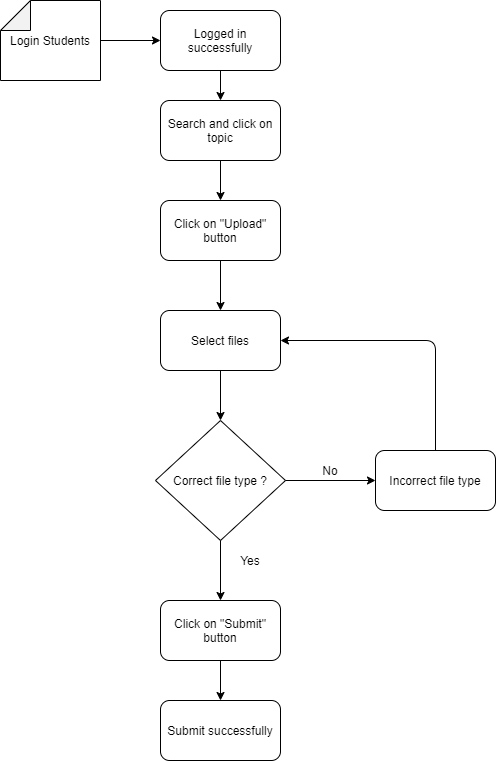
+ Marketing Coordinator commented on the student's contribution

+ Admin create topic for the student to submit their contribution

+ Marketing Manager watch all the contribution and download them

Because there are only 4 flows, we will also have 4 main diagrams:

# Student submit the contribution:



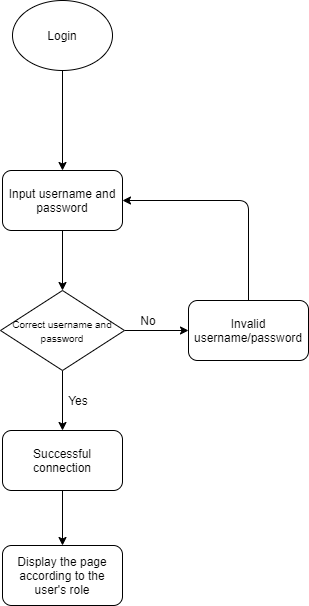
# Marketing Coordinator commented on the student's contribution:

# Admin create topic for the student to submit their contribution:

# Marketing Manager watch all the contribution and download them:

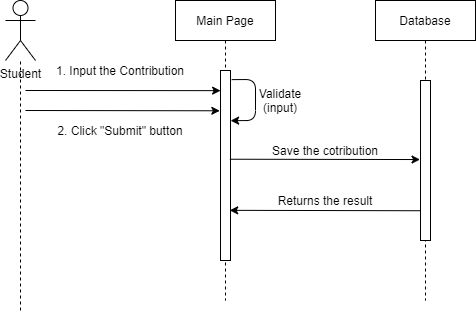
We also have a diagram of the login function of the system:

# Login:

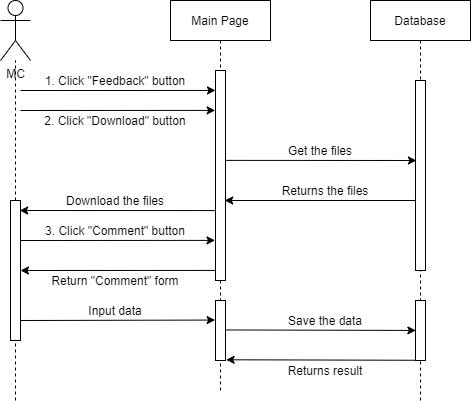


* 1. **Sequence Diagram:**

# Student submit the contribution:

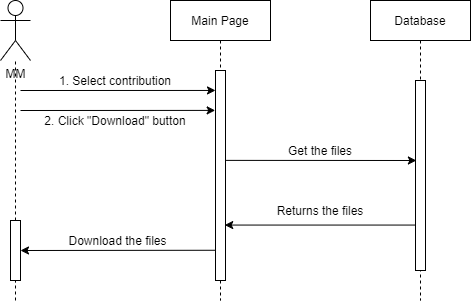


# Marketing Coordinator commented on the student’s contribution:

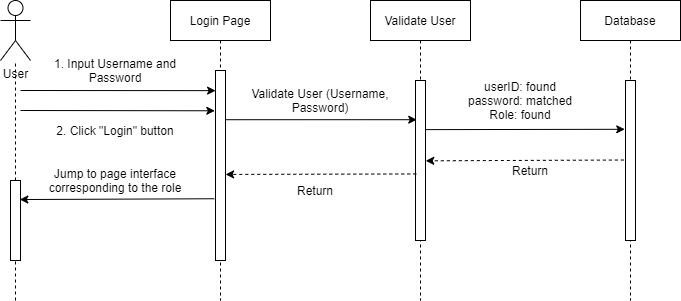


# Admin create topic for the student to submit their contribution:

# Marketing Manager watch all the contribution and download them:

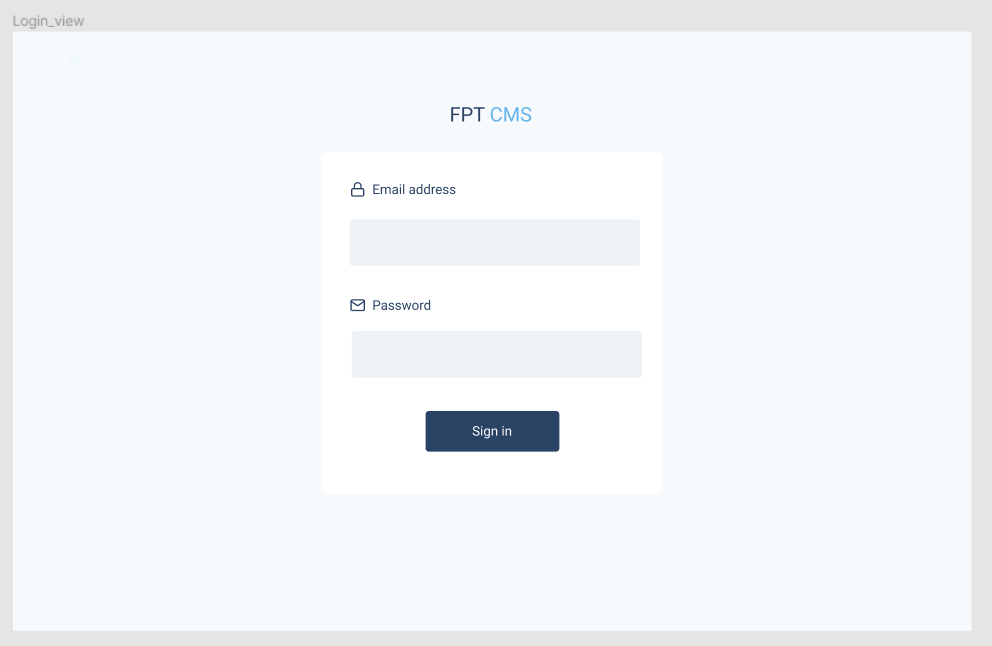


# Login:

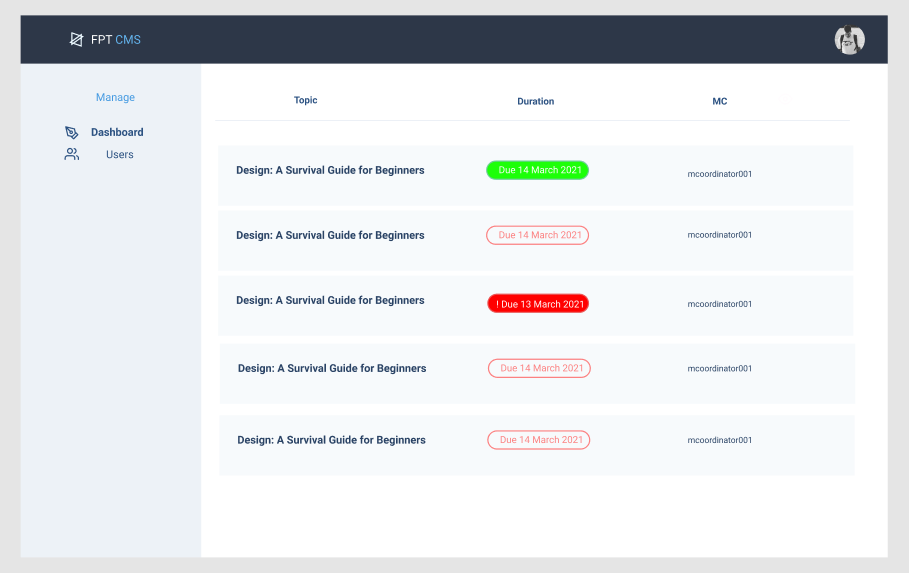


1. **Design:**

# Login:

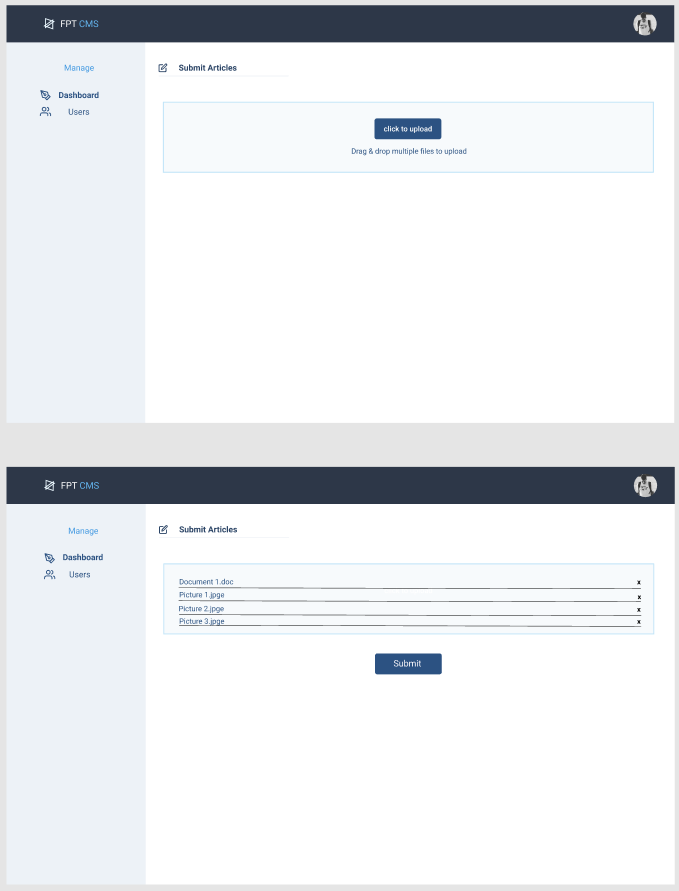


# Student Dashboard:



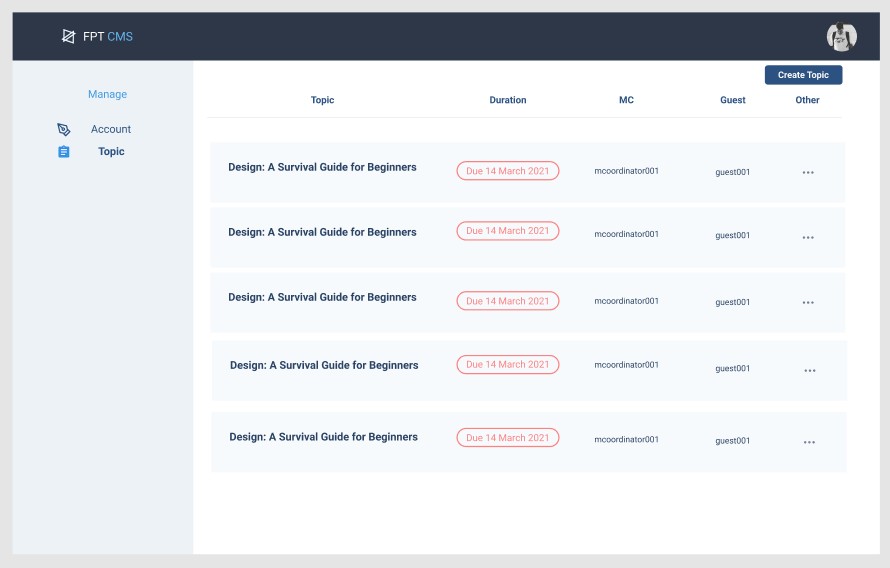
# Student Information:

# Student Submit:

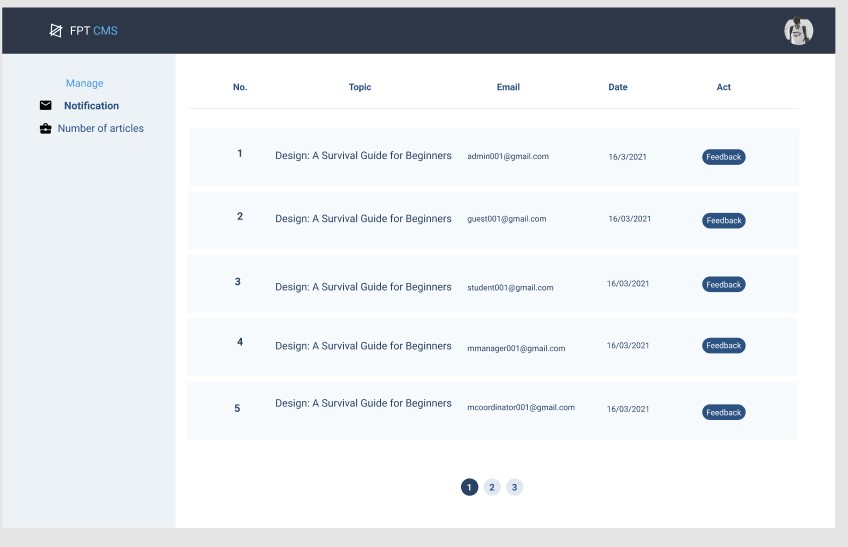


# Admin Dashboard:

# Admin Create Topic:

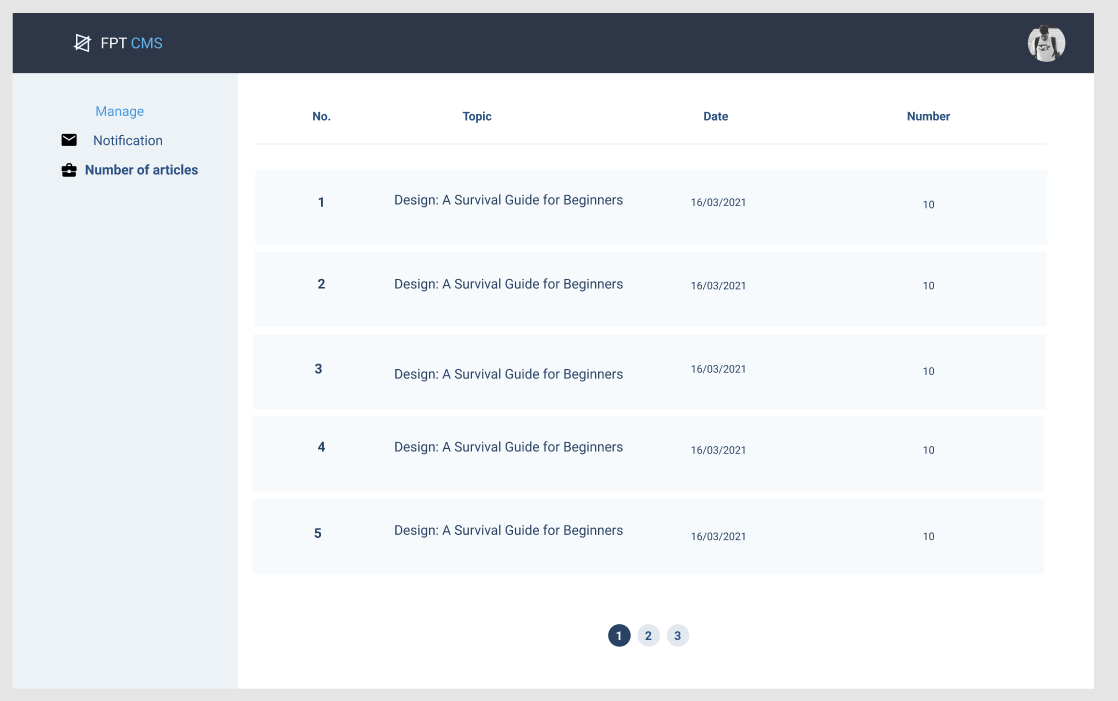


# Marketing Coordinator Dashboard:



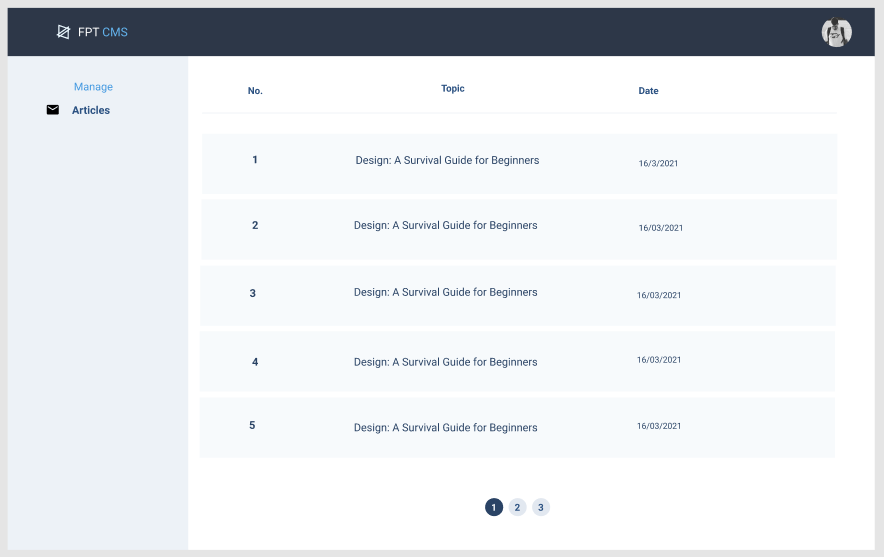
# Marketing Coordinator Comment:

# Marketing Coordinator Count Articles:



# Marketing Manager Dashboard:

# Guest Dashboard:



1. **Testing:**

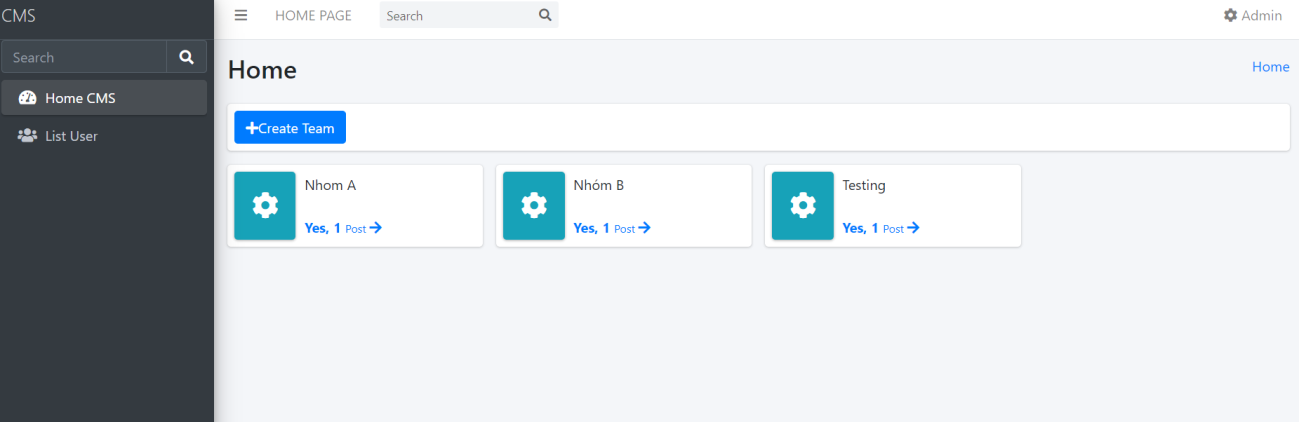
# Test Plan:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Test | What is being test ? | How | Test Data Used | Expected Results |
| 1 | Login into Admin Page | Enter data into the form on the Login Page | Username: [admin@gmail.com](mailto:admin@gmail.com) Password: 123456 | Go to the Admin Page |
| 2 | Login into Student Page | Enter data into the form on the Login Page | Username:  [user2@gmail.com](mailto:user2@gmail.com) Password: 123456 | Go to the Student Page |
| 3 | Login into Marketing Coordinator | Enter data into the form on the Login Page | Username: [user1@gmail.com](mailto:user1@gmail.com) Password: 123456 | Go to the Marketing Coordinator page |
| 4 | Login into Marketing Manager | Enter data into the form on the Login Page | Username: [testuser@gmail.com](mailto:testuser@gmail.com)  Password: 123456 | Go to the Marketing Manager page |
| 5 | Login into Guest | Enter data into the form on the Login Page | Username: [guest@gmail.com](mailto:guest@gmail.com)  Password: 123456 | Go to the Guest page |
| 6 | Add new topic | Enter data into the form on the data entry screen | Topic name: Testing | The new topic was stored into the database |
| 7 | Submit the contribution | Enter data into the form on the data entry screen | Contribution’s name: test | The contribution was stored into the database |
| 8 | Download the contribution | Click on the “Download” button on the screen in the detail page of the  contribution | Contribution’s name: test | Successfully downloaded the contribution |
| 9 | See all the contributions | All contributions in the system display on the screen |  | All contributions in  the system display on the screen |

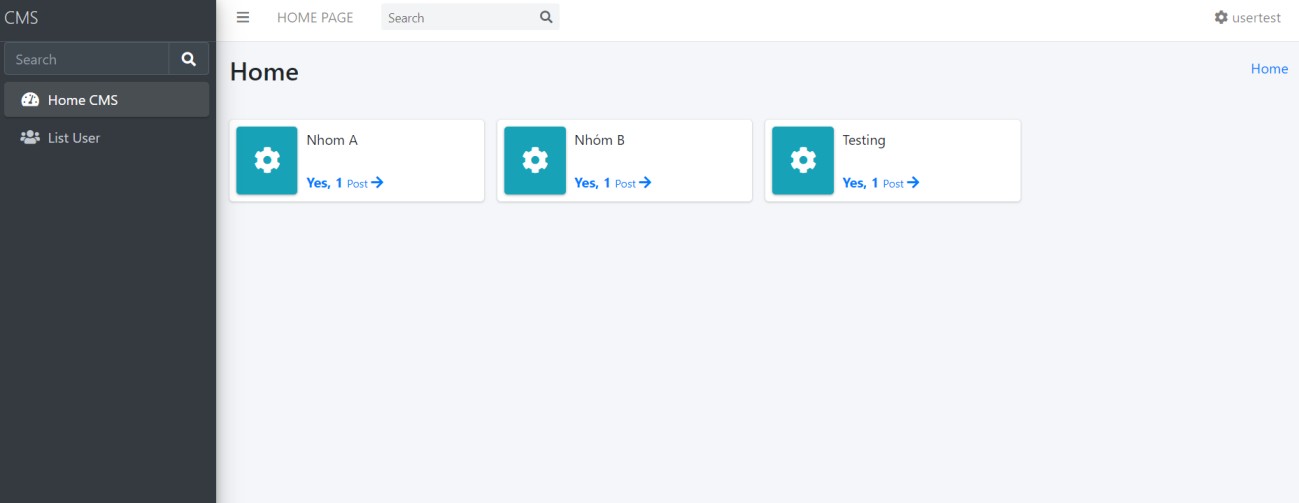
# Test Log:

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Test | What is being test ? | How | Test Data Used | Expected Results | Date | Actual results | Action taken |
| 1 | Login into Admin Page | Enter data into the form on the Login Page | Username: [admin@gmail.com](mailto:admin@gmail.com) Password: 123456 | Go to the Admin Page | April 27,  2021 | OK | None |
| 2 | Login into Student Page | Enter data into the form on the Login Page | Username: [user2@gmail.com](mailto:user2@gmail.com) Password: 123456 | Go to the Student Page | April 27,  2021 | OK | None |
| 3 | Login into Marketing Coordinator | Enter data into the form on the Login Page | Username: [user1@gmail.com](mailto:user1@gmail.com) Password: 123456 | Go to the Marketing Coordinator  page | April 27,  2021 | OK | None |
| 4 | Login into Marketing  Manager | Enter data into the form on the  Login Page | Username: [testuser@gmail.com](mailto:testuser@gmail.com)  Password: 123456 | Go to the Marketing  Manager page | April 27,  2021 | OK | None |
| 5 | Login into Guest | Enter data into  the form on the Login Page | Username:  [guest@gmail.com](mailto:guest@gmail.com) Password: 123456 | Go to the Guest page | April 27,  2021 | OK | None |
| 6 | Add new topic | Enter data into the form on the data entry screen | Topic name: Testing | The new topic was stored into the database | April 27,  2021 | OK | None |
| 7 | Submit the contribution | Enter data into the form on the  data entry screen | Contribution’s name: test | The contribution  was stored into the database | April 27,  2021 | OK | None |
| 8 | Download the contribution | Click on the “Download” button on the screen in the detail page of  the contribution | Contribution’s name: test | Successfully downloaded the contribution | April 27,  2021 | OK | None |
| 9 | See all the contributions | All contributions in the system display on the screen |  | All contributions in the system display on the screen | April 27,  2021 | OK | None |

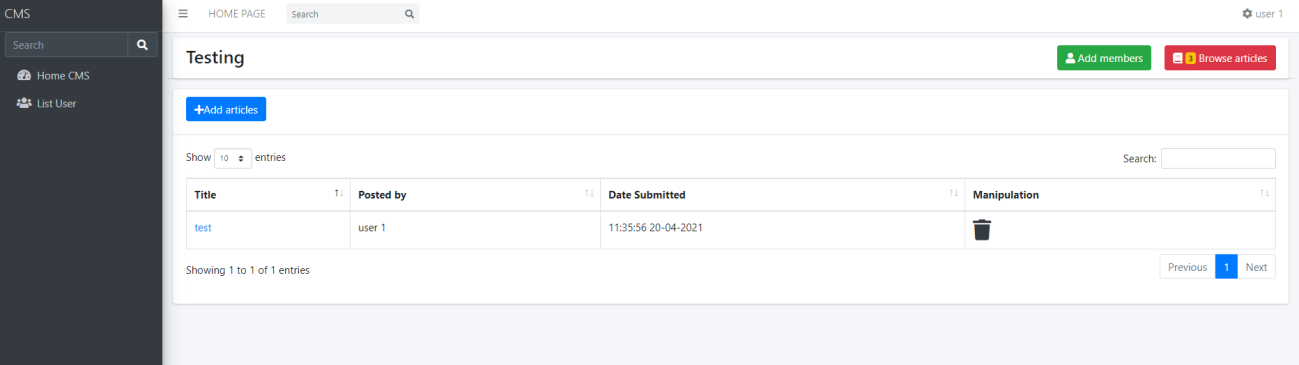
# Test case 1:



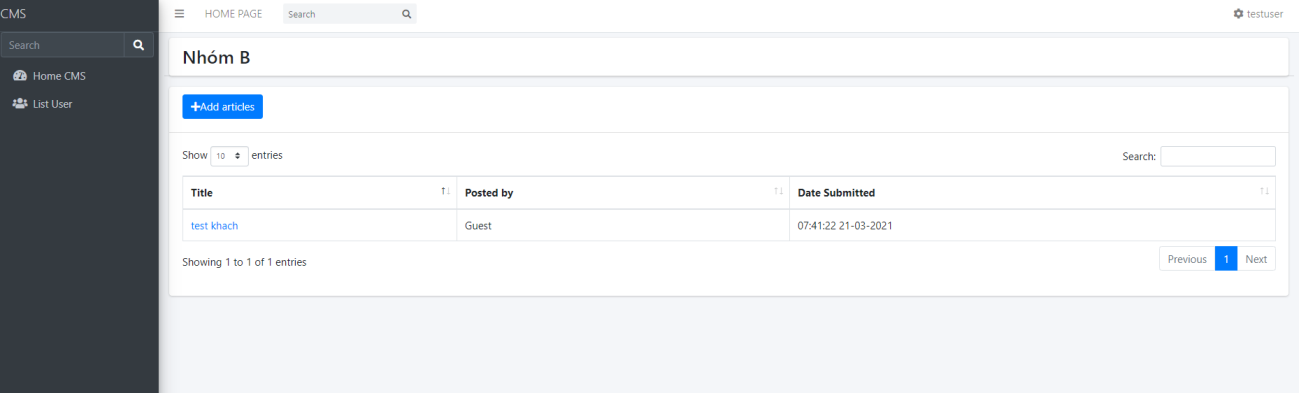
**Test case 2:**



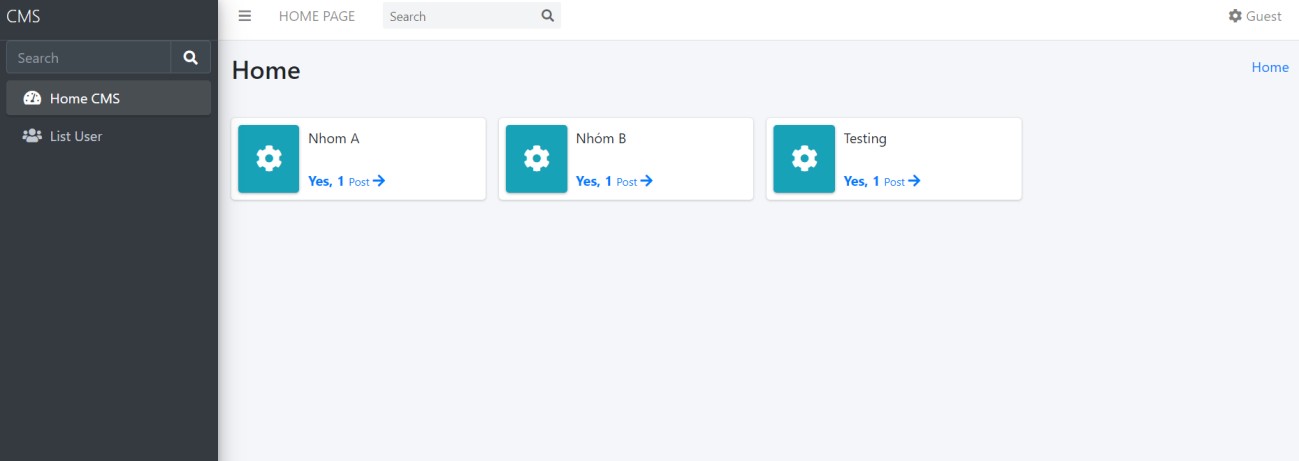
# Test case 3:



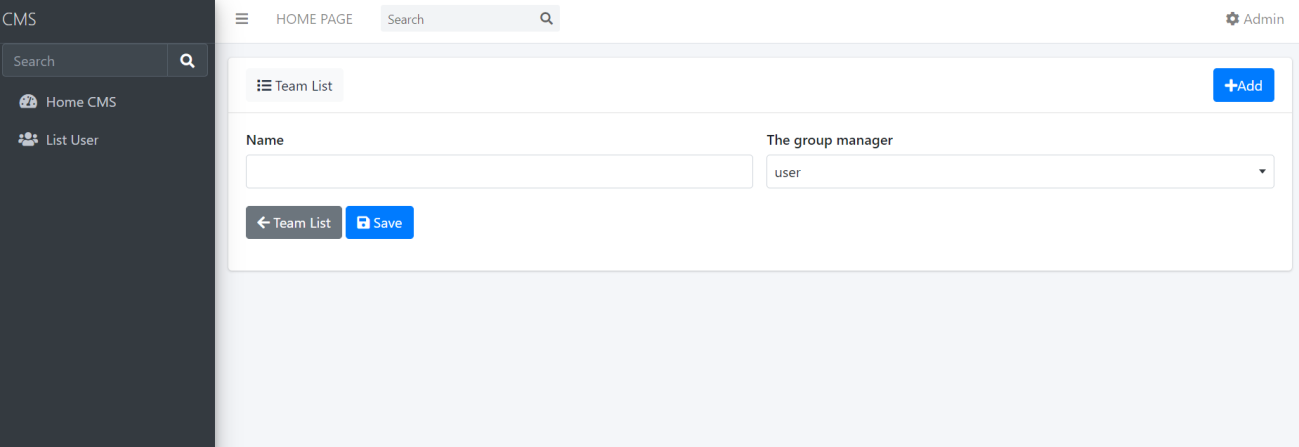
**Test case 4:**



# Test case 5:

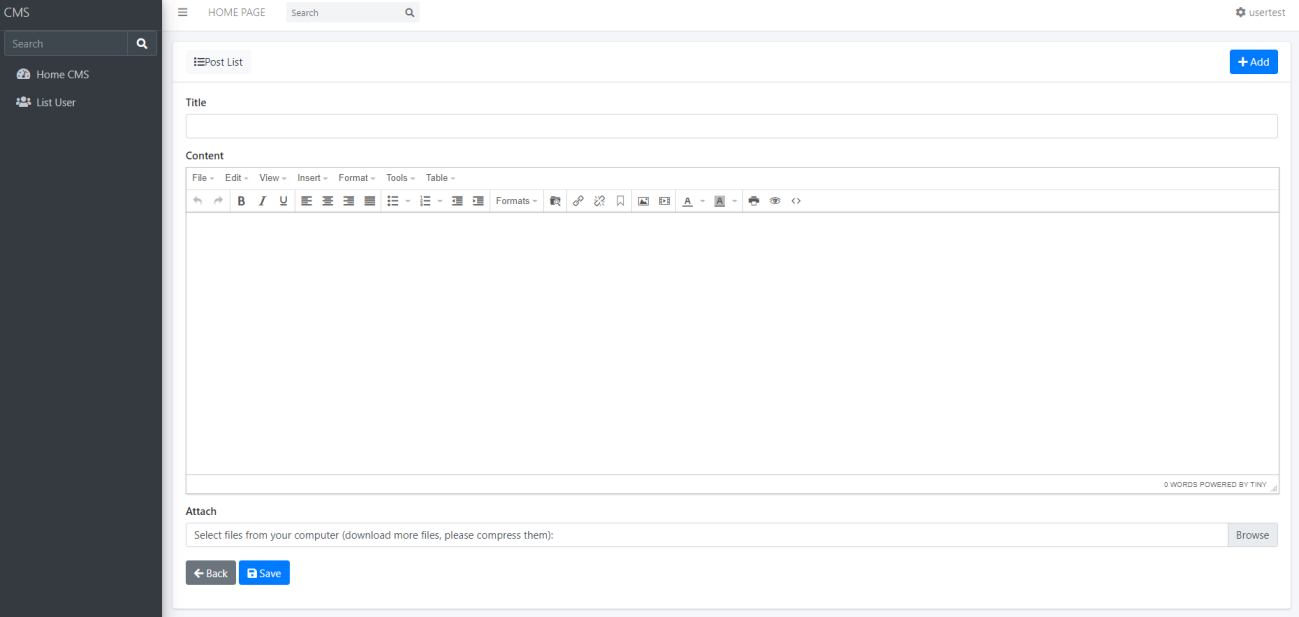


**Test case 6:**

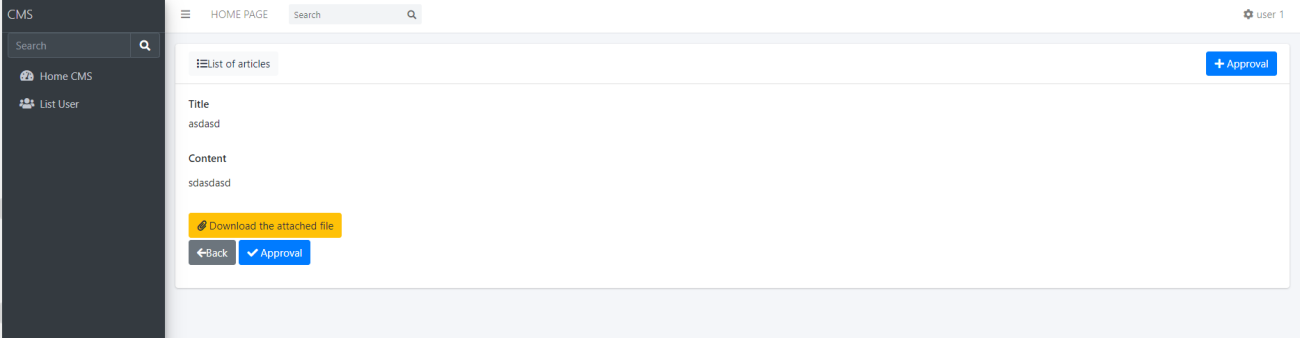




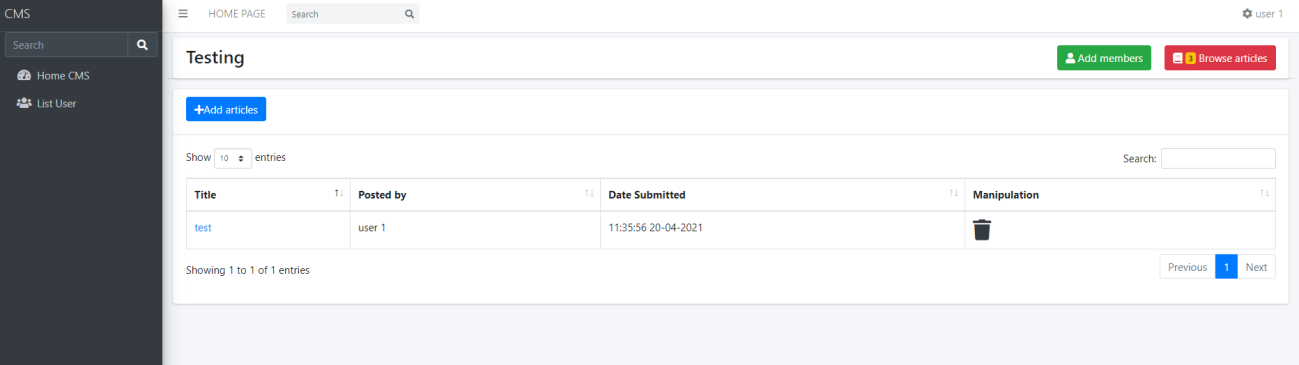
# Test case 7:



**Test case 8:**



# Test case 9:



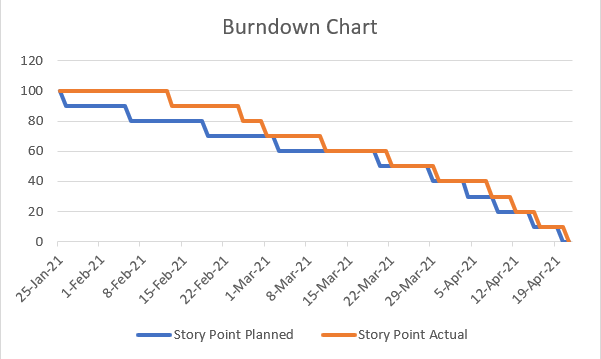
1. **Agile Method:**

* In the Website project that my team is working on, we have applied the Agile method to the project and made a plan to monitor the project implementation. Product Backlogs in this section we set long-term goals for future development of the features within the proposal and Sprint Backlogs is the part where we will set problems or goals that need to be resolved quickly in the short time.
* We use Trello to break down suggestions and by goals we set deadlines to complete. Product Backlogs focuses on the key skills that come out of suggestions in order to stay responsive. out put out. During Sprint 1 from January 25 to February 21, we built the layout to let people imagine how the website is suitable for the user or not. In Sprint 2 (from February 23 to March 08), this phase we’re designing functions for each relevant department such as admin, marketing manager. In Sprint 3 (from March 03 to March 23), this phase we continue to design functionality for roles like admin and marketing manager. In addition, we also design more functions for the next two roles: student and manager coordinator. And in the last Sprint 4 (from March 25 to April 17) we design function for the last roles in this system: Guest.
  + **For administrators**, it will be responsible for maintaining the system, creating accounts, adding, modifying, deleting, updating accounts and inventorying reports, setting a deadline for students to report to their completion. term and can download any student report if they request a review.
  + **Marketing Director:** View all contributions but cannot edit, download all files in ZIP format to transfer out of the system and can unzip the student report.
  + **Marketing Coordinator:** Each Coordinator will manage the process for the Faculty for which they are responsible. When students submit a marketing report, they will receive a notification from the system, then communicate with students through comments based on the report and can suggest ideas to each other and exchange time. within 14 days, only interact with the department for which the coordinator is responsible.
  + **Students** will be created an account when they become a student of the system, based on the data provided by the student, the administrator will create an account in the system and agree to the conditions of the school / system. When participating in certain subjects, students will make reports and submit reports to the system from which the system will be assigned to mark students. Students can submit one

or more papers in Word format, download high-quality images, edited ahead of time, after the deadline any reports or edits will not be accepted in any form.

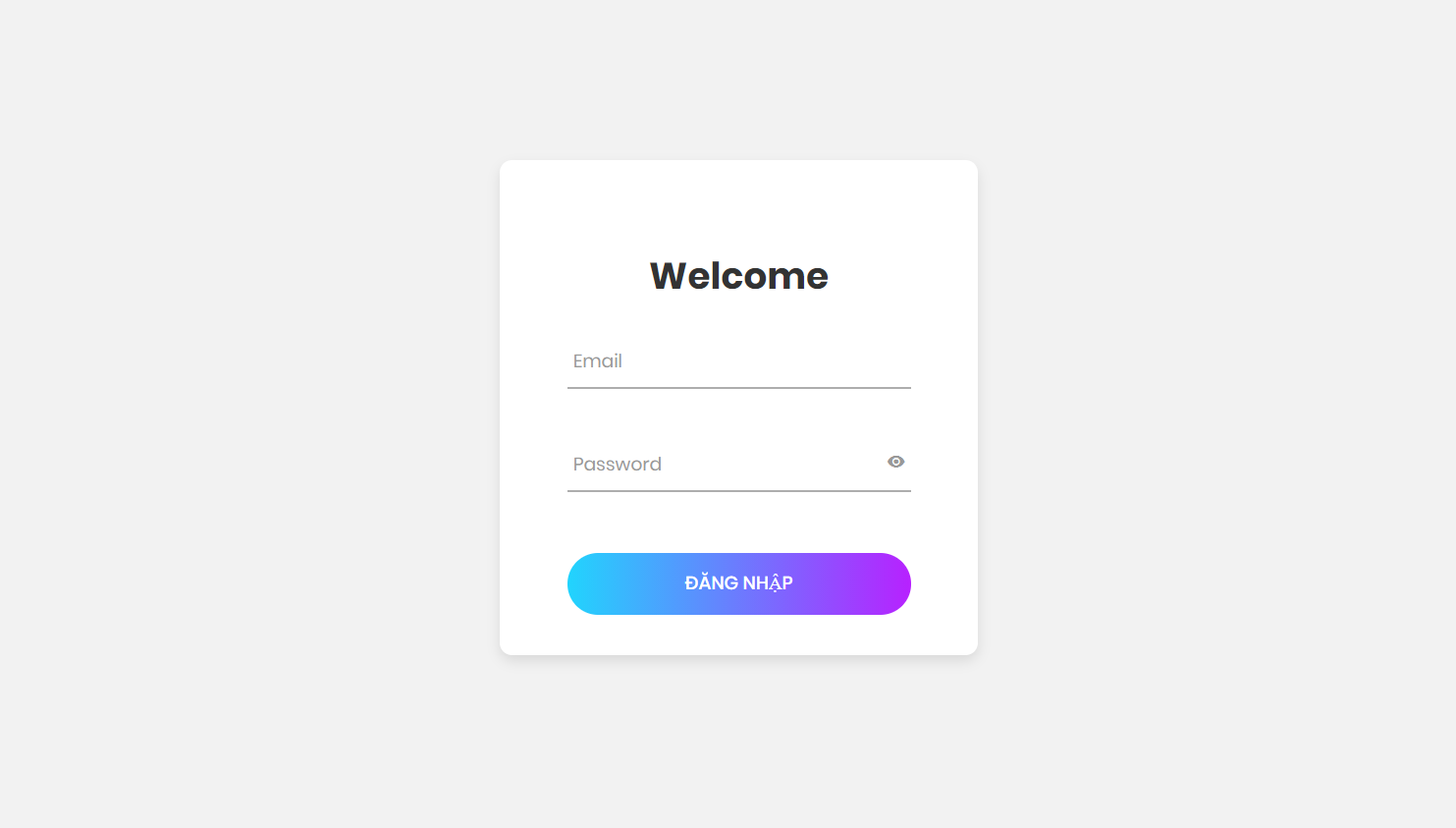
* + Finally, **Guest** will be able to see the articles posted by the system and the events of the system.
* But it was not all according to the plan the team had planning. We have difficulty implementing code that matches the functions of the system. Also, a team member had a relative in the family dies, so we were delayed for the most 2 days.
* About Daily Meeting: Because this is our first time working on a project with the Agile methodology, the whole team often forgot to have daily meetings with each other every day to inform project progress. We have agreed that the daily meeting will be at 5 pm every day. But some team members forgot about this and contributed to the delay in the project.

# Burndown Charts:



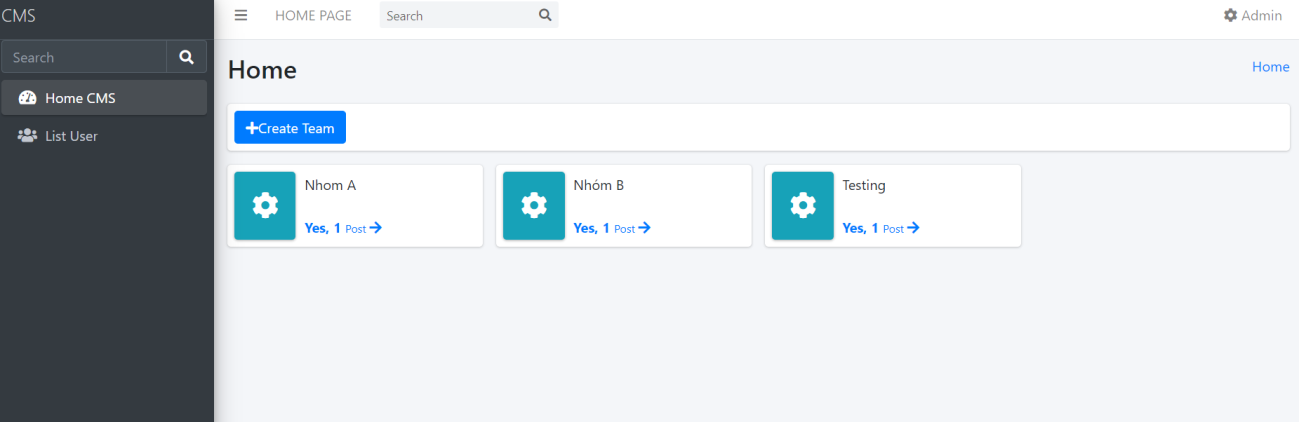
1. **Product:**

Login

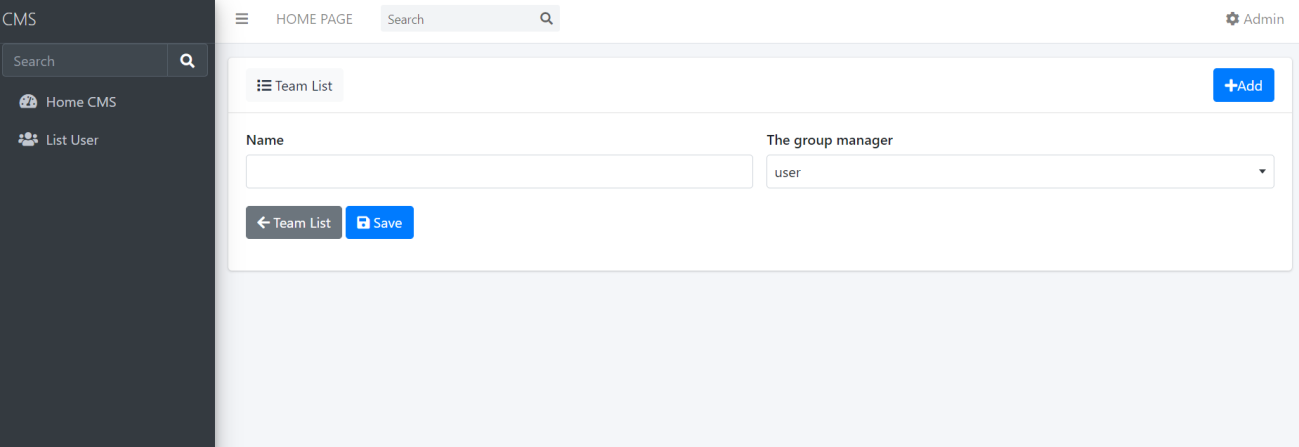


# Admin:

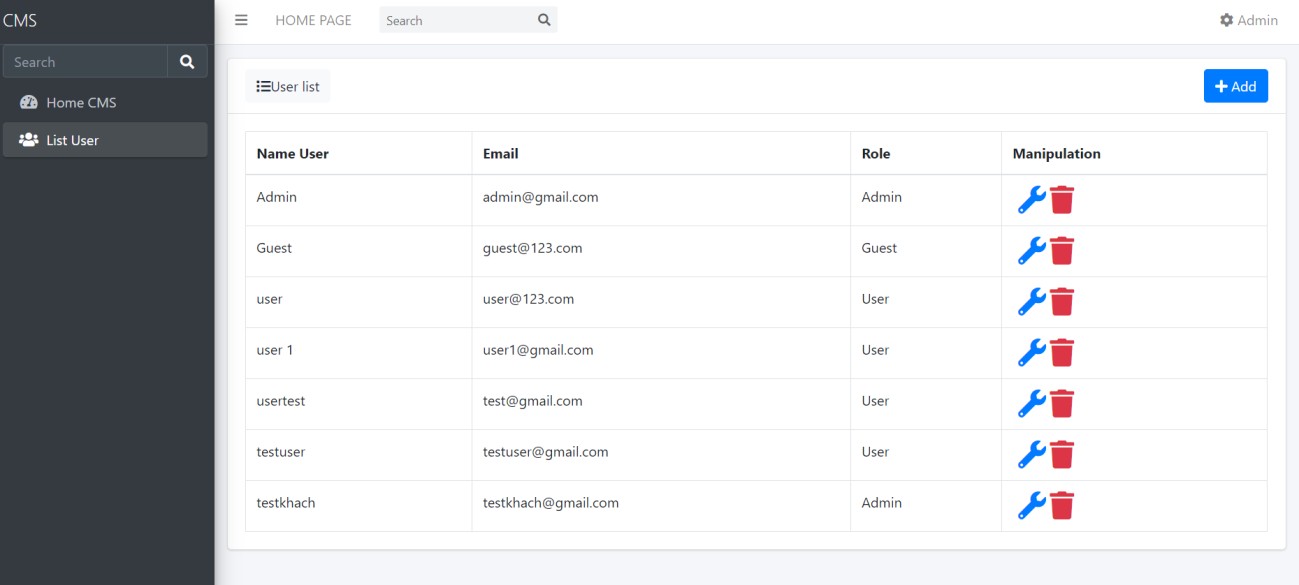
This is Dashboard of Admin



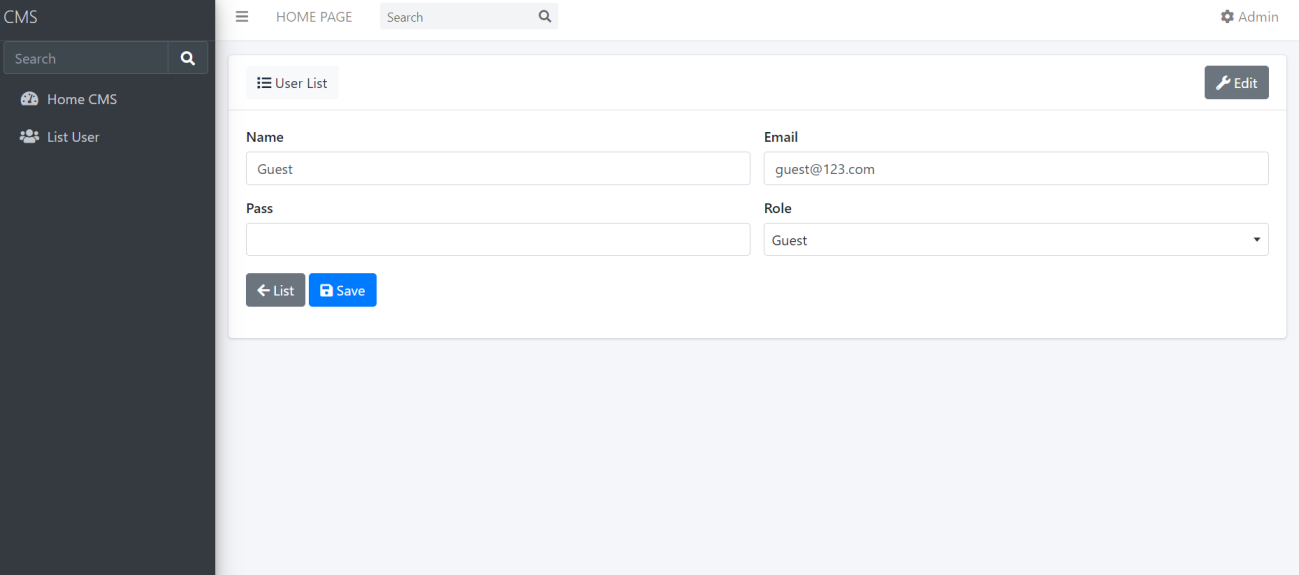
Creat Team by Admin:



User Of List:

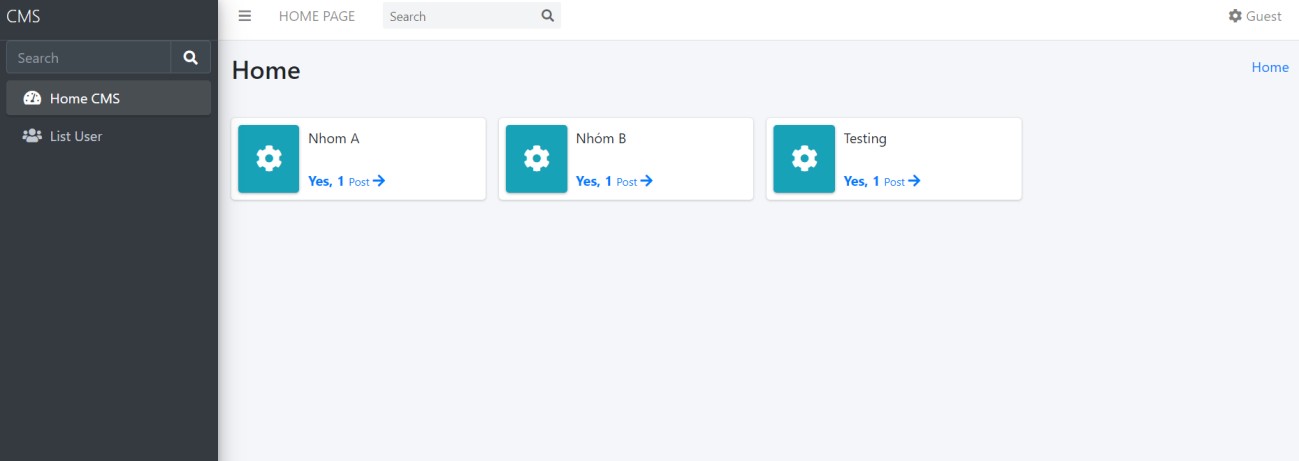


Edit user account by admin:

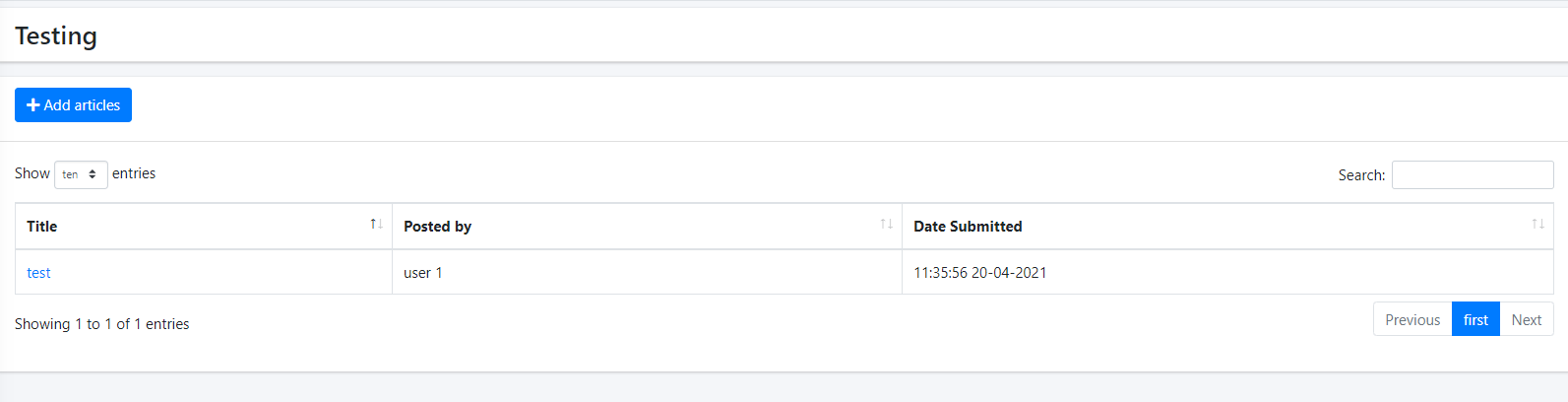


# GUEST :

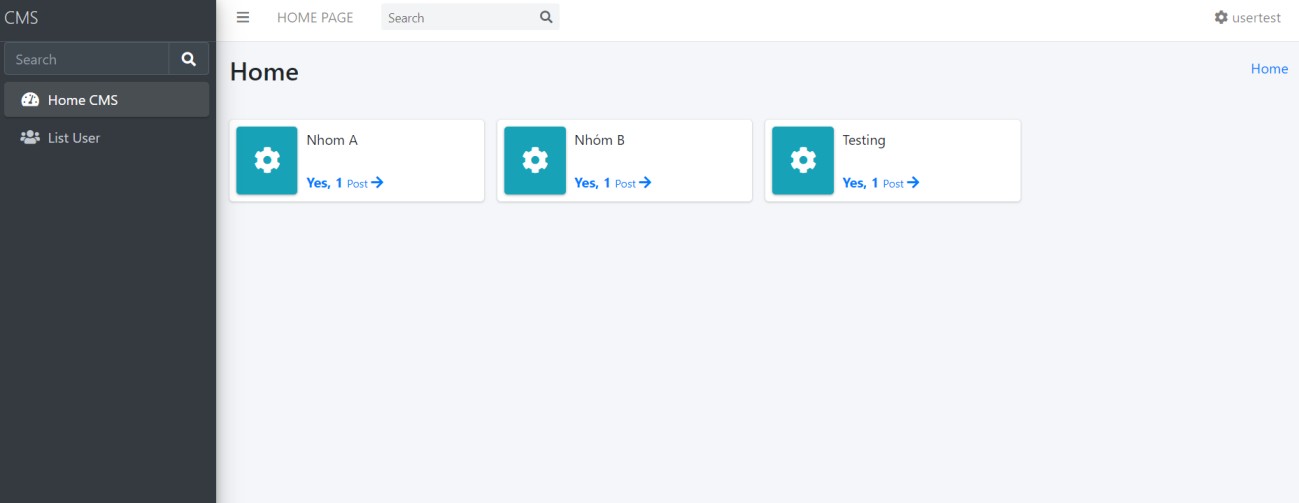
This is Dashboard of Guest



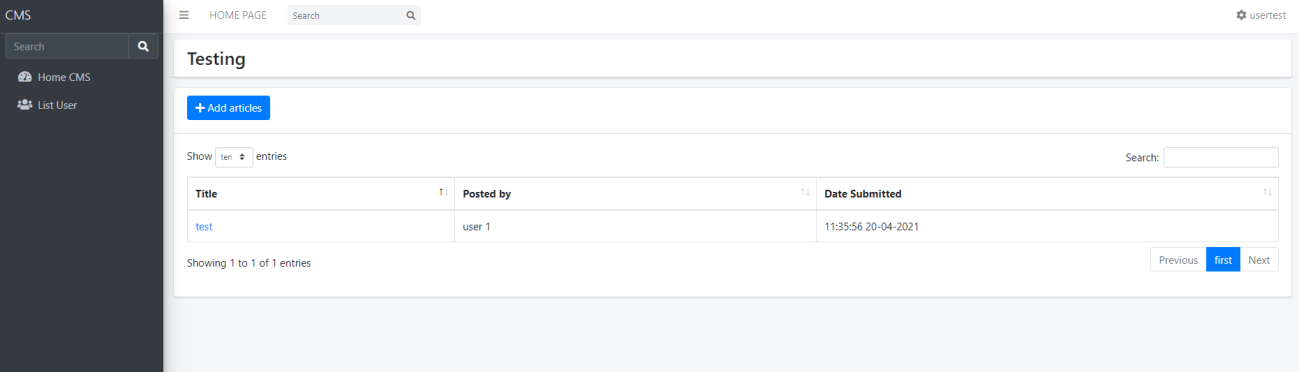
View topic :



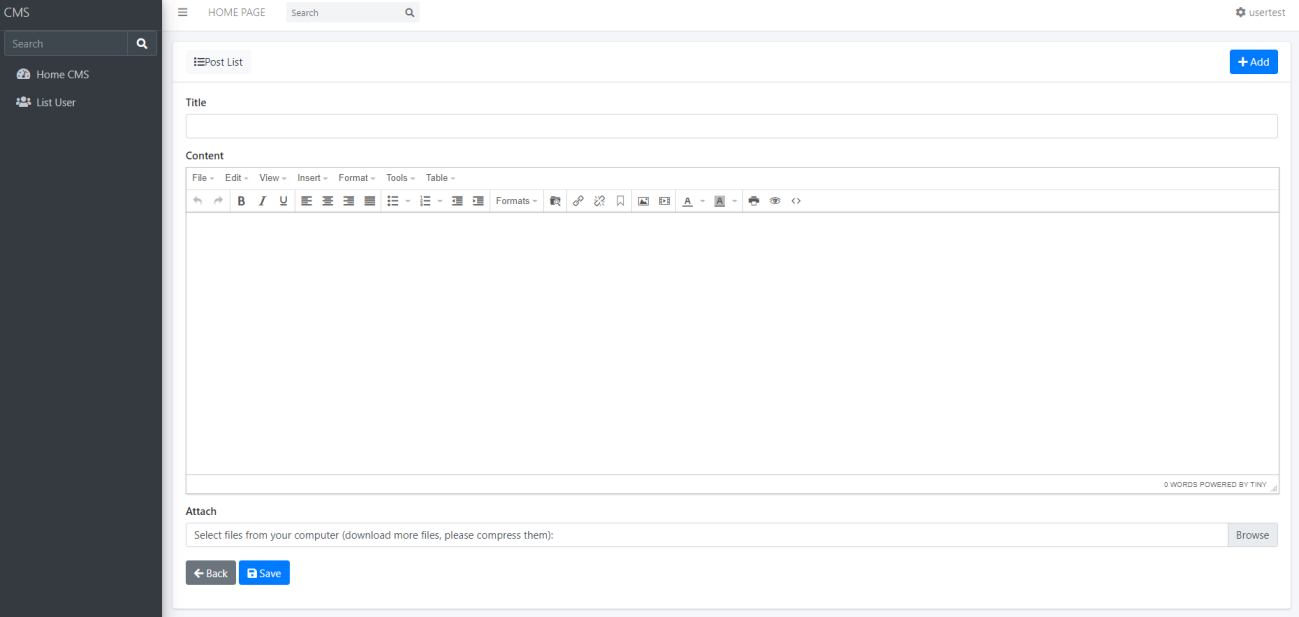
User(Students):



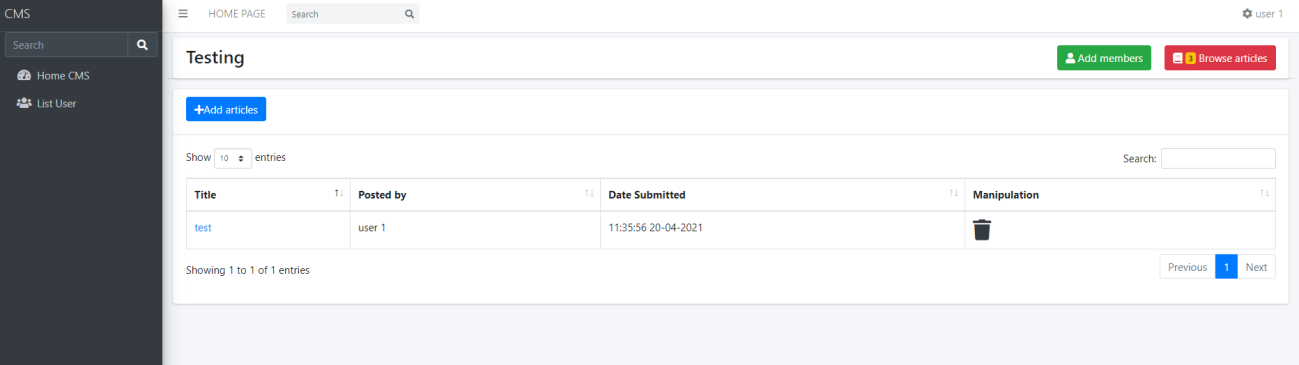
View Report list

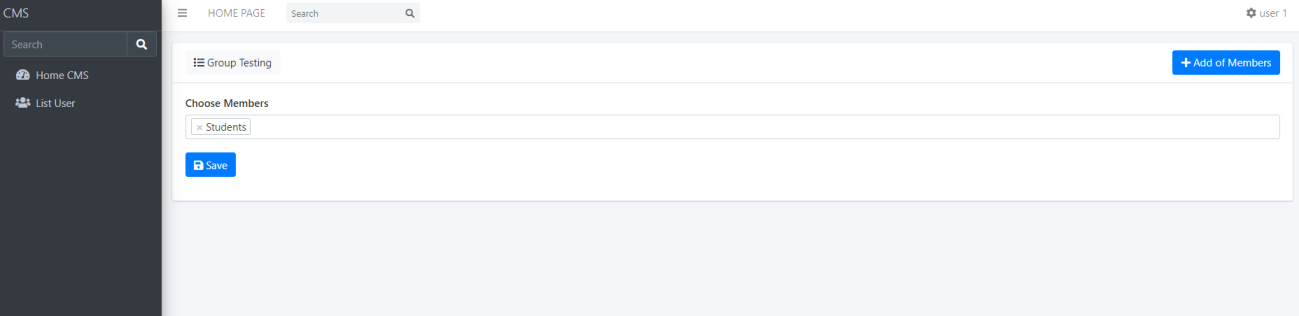


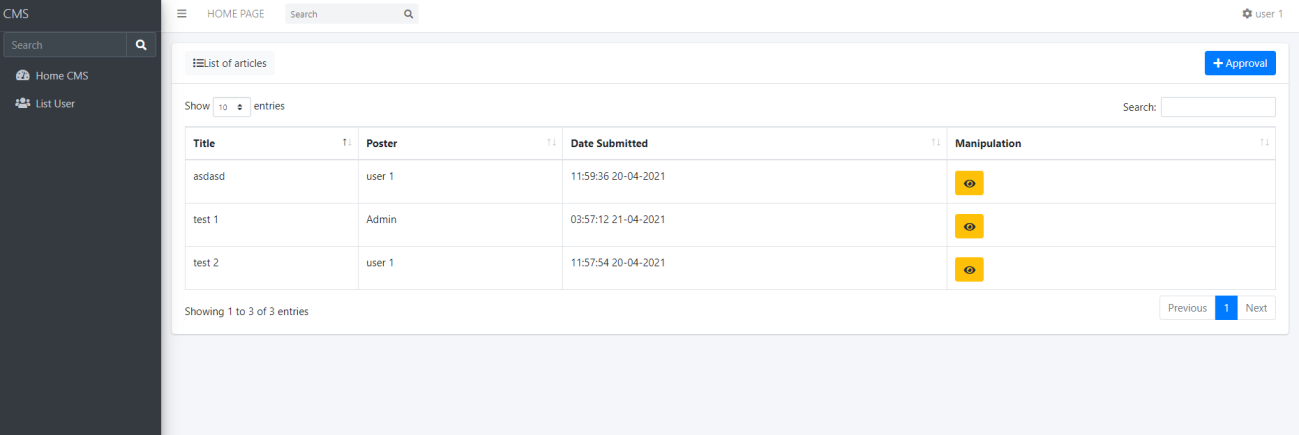
Upload File

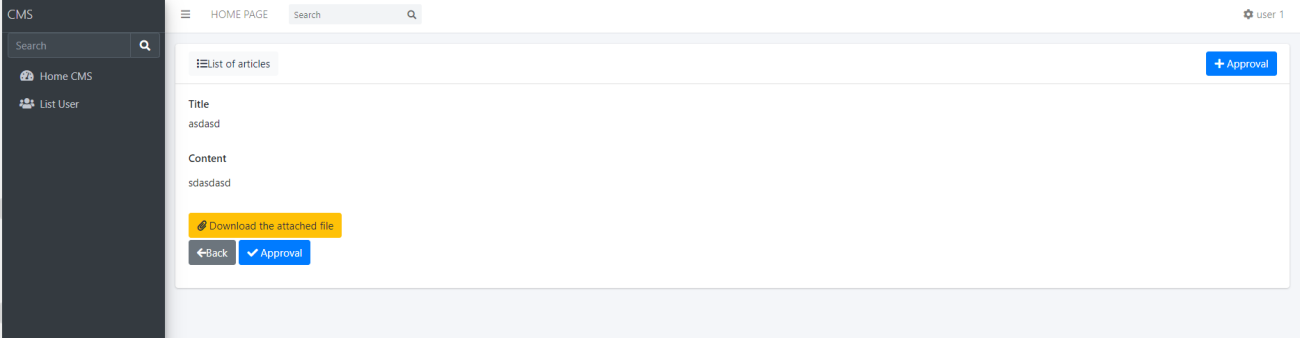


Marketing Coordinator

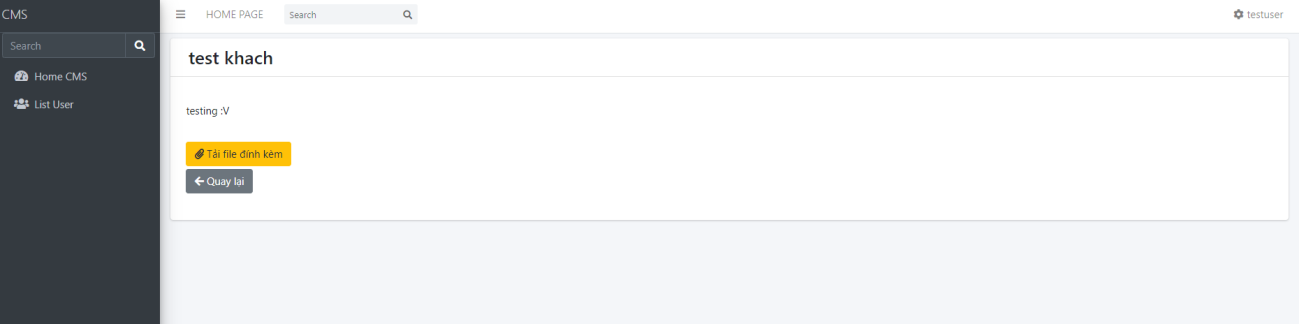
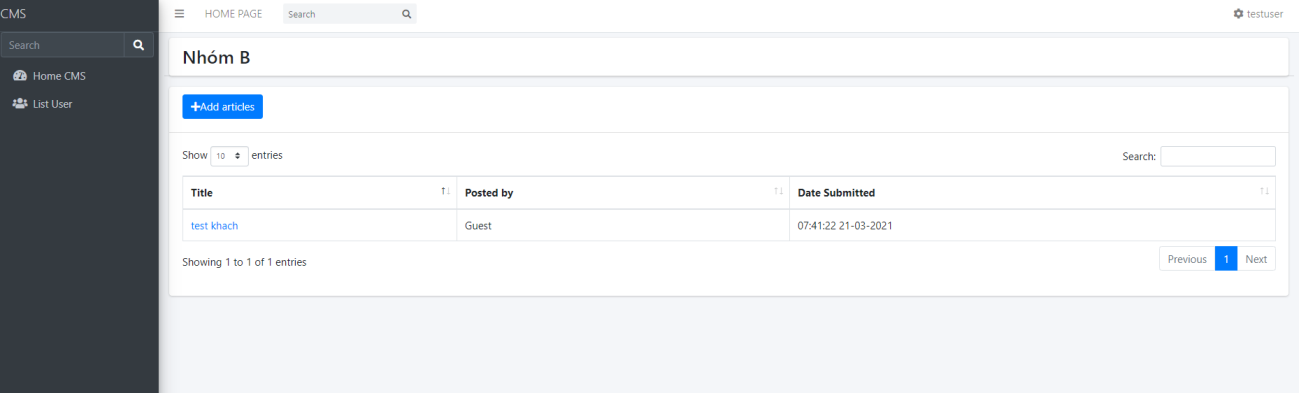








Marketing Manager :



# Links:

* **Daily Meeting:**

<https://drive.google.com/drive/u/1/folders/1o5xHTW7hNtcsM_t10BLFya2lvABMKQsz>

# Trello:

<https://trello.com/b/6XkAKik0/enterprise-web>

# Github:

<https://github.com/quanlam123m/enterprise-website>