|  |  |
| --- | --- |
|  | **MINISTRY OF EDUCATION AND TRAINING** |

|  |
| --- |
| **FPT UNIVERSITY** |
| Capstone Project Document |
| Gunpla World Application |
|  |
| |  |  | | --- | --- | | **Team 15** | | | **Group Members** | Hoàng Ngô Minh Tùng – Role 1 – SE62231  Nguyễn Bảo Khánh – Role 2 – SE61978  Lê Nguyễn Ngọc Sang – Role 3 – SE61647 | | **Supervisor** | Phạm Công Thành | | **Ext Supervisor** |  | | **Capstone Project code** |  | |
|  |
|  |

– TP.HCM, 09/2018 –

**C. Report No.3 Software Requirement Specification**

**1. User Requirement Specification**

Our system is designed to act as a platform of information, hosting events and trading about Gunpla World. Users can use search system with nearby location through website and mobile. We also provide many advanced features to heighten user experience and make it more effective for administrator to manage the entire system. This software consists of three main user roles which are Member, Buyer/Seller and Admin with Admin are our employees.

**1.1. Member Requirement**

Member is a person who doesn’t have access to the system. Member can get gunpla’s information, publish articles, register for attending events, proposal to open events, contests, order the exchanges and evaluate. Below are functions of member:

* Register
* Login
* Search for models, get models’s detail
* Search for articles, get articles’s information
* Manipulate articles, articles
* Search for events, get events’s detail
* Search for exchange, get exchange’s detail
* Register event with payment
* Proposal to open events, contests
* Manipulate event proposals
* Order the exchanges
* Evaluate with rating and feedback

**1.2. Buyer/Seller Requirement**

Buyer/Seller is a role that is upgrade from Member, has all functions of Member and have more features about exchange, trading. Below are functions of this role which member doesn’t have:

* Post for exchange, for purchase or for sale
* Manipulate the exchange
* Confirm about the transaction

**1.3. Administrator Requirement**

Administrator is a role as an employee who is responsible for managing the entire system such as manipulating account, manipulating models, articles, events, approving requests or manipulating the crawling, includes crawl models and crawl information articles. Administrator can do the following functions:

* Manipulate account
* Manipulate crawling, includes: crawl models and information articles
* Manipulate gunpla model information system
* Manipulate articles, articles
* Approve requests to publish articles, proposal to open events, contests and to post for exchange
* Manipulate events, contests
* Manipulate buyer/seller’s exchange

**1.4. Authenticated User Requirement**

Authenticated user is a person who already logined into system. Authenticated user can do the following functions:

* Logout
* Edit profile

**1.5. Scheduler Requirement**

Scheduler is a kind of system role that run functions in the backend. Scheduler can do the following functions:

* Send notifications

**2. System Requirement Specification**

**2.1. External Interface Requirement**

**2.1.1. User Interface**

The user interface uses English for both web application and mobile application.

**2.1.2. Hardware Interface**

* **Server:**
  + Ram: 4GB
  + Computer Processor: Intel® Core™ i5 (4 CPUs) @ 2.5GHz
  + Operating system: Windows 10
  + Internet connection: Cable, Wi-Fi (20 Mbps)
* **Mobile smartphone:**
  + Ram: 2GB
  + Processor: CPU Quad core 2.0 GHz
  + Operating system: Android 6 Marshmallow

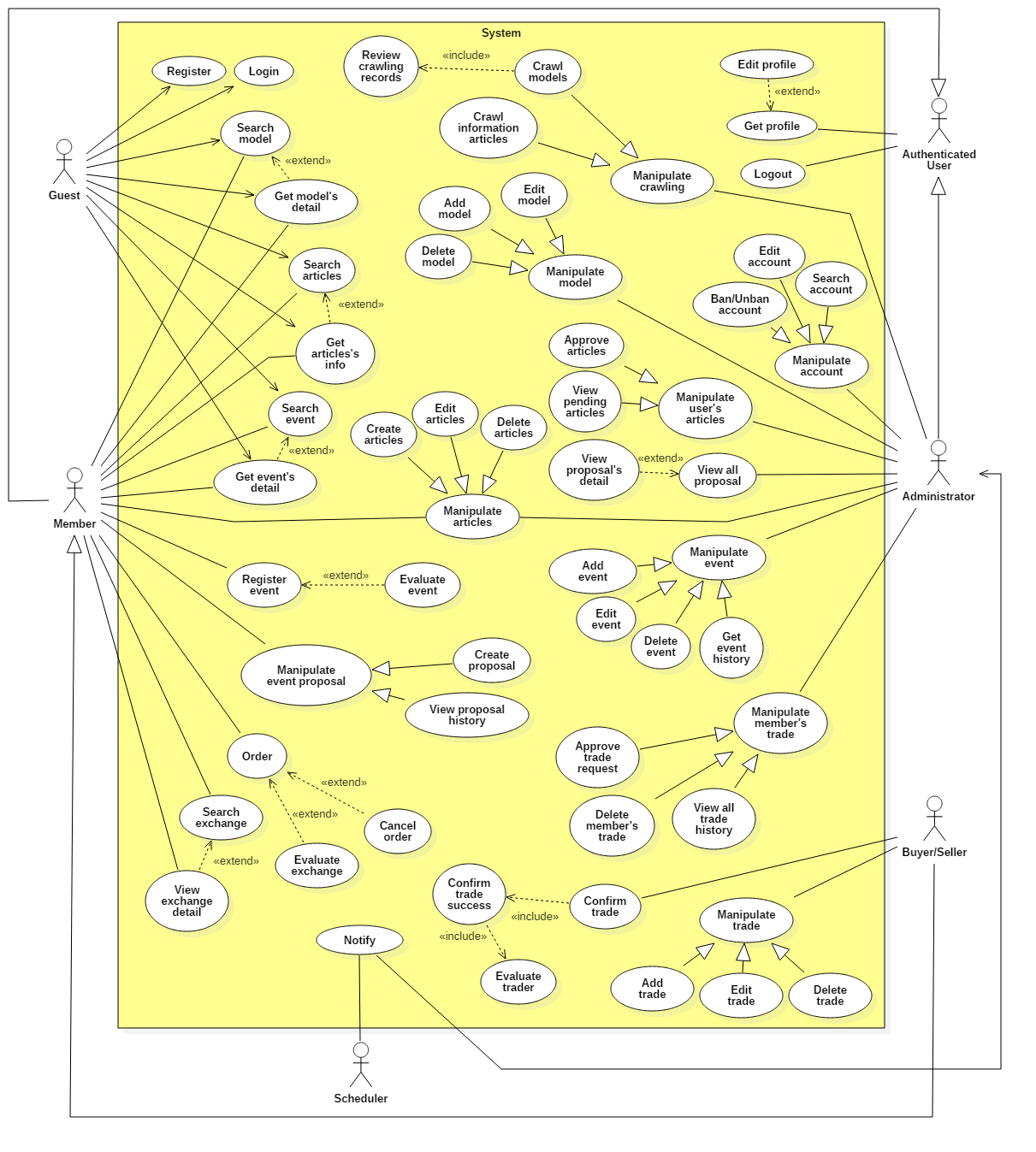
**2.1.3. Software Interface**

* Operating system: Window 10, Android 6 Marshmallow
* Environment: Java EE 8, Android SDK APIs level 15 or above
* IDE: IntelliJ IDEA 2018.2.3, Netbeans 8.0.2, Android Studio 3.1.3
* DBMS: MySQL 5.7.20, MySQL Workbench 6.3.9 CE
* Source control: SourceTree 2.6.10
* Web browser: Chrome 62 or above

**2.1.4. Communication Protocol**

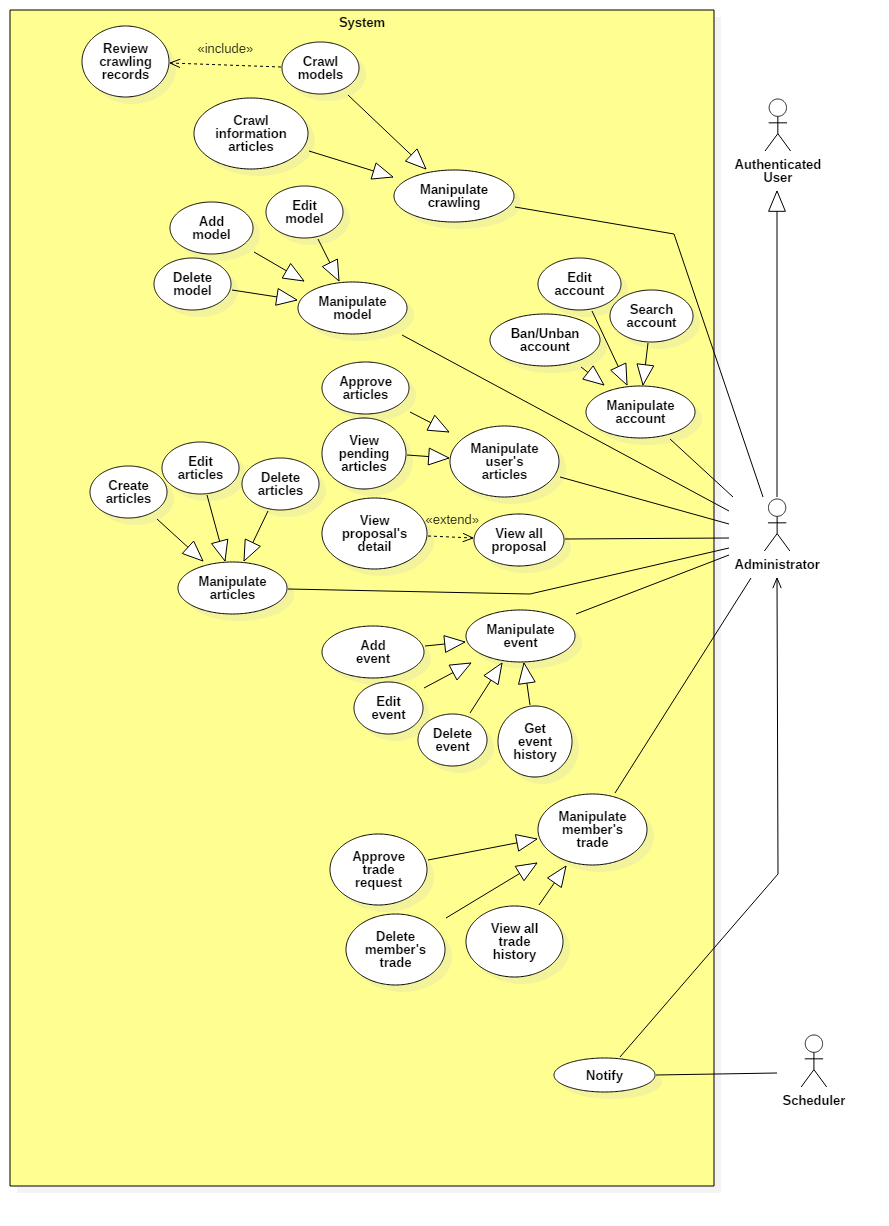
* Use HTTP protocol 1.1 for communication between:
  + Web application and web server
  + Mobile application and web server

**2.2. System Overview Usecase**

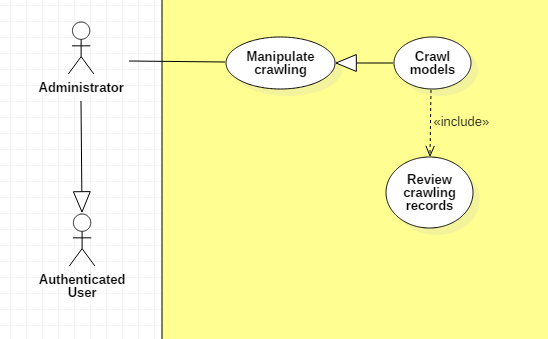


**2.3. List of Usecase**

**2.3.1. < Administrator > Overview Usecase**

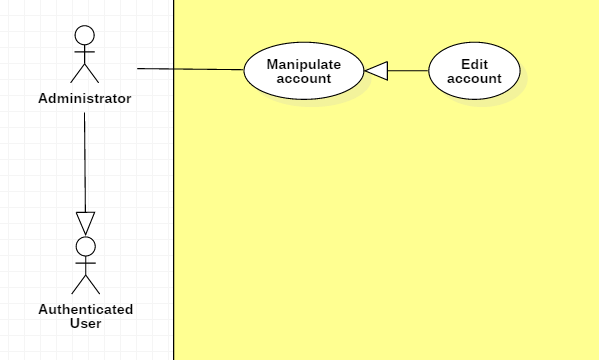


**2.3.1.1. < Administrator > Crawl models Usecase**



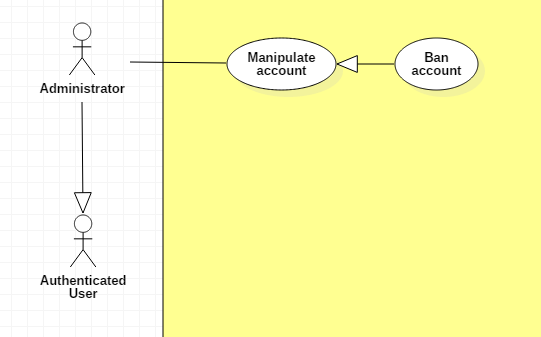
|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE – UC \_A.01** | | | |
| **Usecase No.** | A.01 | **Usecase Version** | 1.0 |
| **Usecase Name** | Crawl models | | |
| **Author** | TungHNM | | |
| **Date** | 28/09/2018 | **Priority** | Normal |
| **Actor:**   * Administrator.   **Summary:**   * This use case allows administrator to crawl models from high domain: https://www.1999.co.jp/eng/gundam.   **Goal:**   * Information about models will be crawled and saved to database after administrator review.   **Triggers:**   * Administrator sends crawl models command to the system.   **Preconditions:**   * User must login into system with role Administrator. * Crawl page is displayed.   **Post Conditions:**   * **Success**: Models’s crawling records will displayed on page for administrator to review, after that save the new records to database. * **Fail**: Show error message.   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | **Step** | **Actor Action** | **System Response** | | 1 | Administrator clicks on “Crawl Models” button | A list of models’s crawling records is shown with following labels and fields:   * Code: text, required, max length: 50 * Name: text, required, max length: 250 * Product Series: text, required, max length: 250 * Series Title: text, required, max length: 250 * Manufacturer: text, max length: 100 * Price: text, max length: 20 * Released Date: text, max length: 20 * Description: text, max length: 350 | | 2 | Administrator reviews all records.  [Alternative 1, 2] |  | | 3 | Administrator clicks on “SAVE” button | Validate data.  [Exception 1]  Match new records with existed data in database.  Save new models to database.  System shows message gunpla models have been saved successfully.  [Alternative 3] |   **Alternative Scenario:**  **Alternative Scenario 1: Delete a record**   |  |  |  | | --- | --- | --- | | **Step** | **Actor Action** | **System Response** | | 1 | Administrator delete a record | Delete the selected temporary record in database. |   **Alternative Scenario 2: Click on a record**   |  |  |  | | --- | --- | --- | | **Step** | **Actor Action** | **System Response** | | 1 | Administrator clicks on a record | System displays the selected record’s detail page. |   **Alternative Scenario 3: No new records**   |  |  |  | | --- | --- | --- | | **Step** | **Actor Action** | **System Response** | | 1 | Administrator clicks on “SAVE” button | If there is no new records after matching with existed data in database, system shows message there is no new records. |   **Exceptions:**   |  |  |  | | --- | --- | --- | | **No** | **Cause** | **System Response** | | 1 | Administrator does not fill out all the required fields | System shows error message to ask administrator to enter missing required fields. | | 2 | Length of field’s value is out of range | System shows error message that notify administrator which field’s value is out of range. |   **Relationships:** N/A  **Business** **Rules**:   * In case of success scenarios, new models’s crawling data would be added to database. * All crawling records are saved in a temporary table in database before administrator review. * If administrator edit records, all required fields must be filled out. * Matching the recent crawling records with existed data in database, if new save to the main Model table. | | | |

**2.3.1.2. < Administrator > Edit Account Usecase**



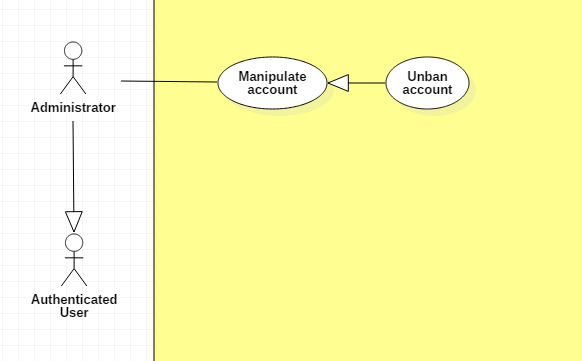
|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE – UC \_A.02** | | | |
| **Usecase No.** | A.02 | **Usecase Version** | 1.0 |
| **Usecase Name** | Edit account | | |
| **Author** | TungHNM | | |
| **Date** | 29/09/2018 | **Priority** | Normal |
| **Actor:**   * Administrator.   **Summary:**   * This use case allows administrator to update Members’s information.   **Goal:**   * Information will be customized to be more reasonable.   **Triggers:**   * Administrator sends edit account command to the system.   **Preconditions:**   * User must login into system with role Administrator. * The profile page of selected account is displayed.   **Post Conditions:**   * **Success**: Member’s information is updated. * **Fail**: Show error message.   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | **Step** | **Actor Action** | **System Response** | | 1 | Administrator clicks on “Edit profile” button | Edit Account page is shown with following labels and fields:   * Email: text, required, max length: 50, RegEx: /^([a-z0-9\_\.-]+)@([\da-z\.-]+)\.([a-z\.]{2,6})$/ * First Name: text, max length: 50 * Middle Name: text, max length: 50 * Last Name: text, max length: 50 * Tel: text, length: 10-11 * Address: text, max length: 250 * Birthday: DateTimePicker   System automatically fills out the form with information of the selected user’s profile in database. | | 2 | Administrator fills out the form |  | | 3 | Administrator clicks on “SAVE” button | Validate data.  [Exception 1, 2, 3, 4, 5]  Update account information to database  System displays Profile page with updated information. |   **Alternative Scenario:** N/A  **Exceptions:**   |  |  |  | | --- | --- | --- | | **No** | **Cause** | **System Response** | | 1 | Email does not match the required format: /^([a-z0-9\_\.-]+)@([\da-z\.-]+)\.([a-z\.]{2,6})$/ | System shows error message: Invalid email ! | | 2 | Phone number is not string of number | System shows error message: Invalid phone number ! | | 3 | Email is duplicate in database | System shows error message: The email has been existed ! | | 4 | Missing of required fields | System shows error message that notify administrator to input the required fields | | 5 | Length of field’s value is out of range | System shows error message that notify administrator which field’s value is out of range. |   **Relationships:** N/A  **Business** **Rules**:   * Account’s information will be automatically filled out when Edit Account page is loaded or whenever an exception occurs. * In case of success scenarios, account’s information would be updated to the database and displays the profile page with updated information. * Tel must be string of numbers. * Email must not be duplicate in database. * An email address must be validated by this regular expression:   /^([a-z0-9\_\.-]+)@([\da-z\.-]+)\.([a-z\.]{2,6})$/ | | | |

**2.3.1.3. < Administrator > Ban Account Usecase**



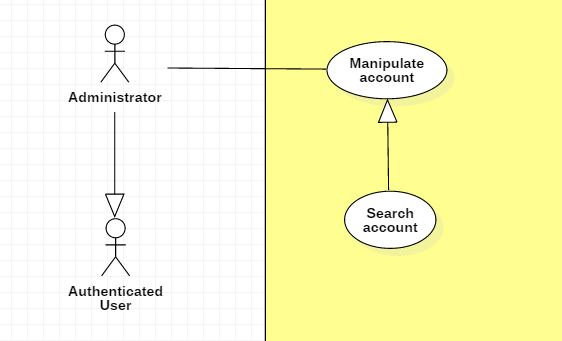
|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE – UC \_A.03** | | | |
| **Usecase No.** | A.03 | **Usecase Version** | 1.0 |
| **Usecase Name** | Ban account | | |
| **Author** | TungHNM | | |
| **Date** | 29/09/2018 | **Priority** | Normal |
| **Actor:**   * Administrator.   **Summary:**   * This use case allows administrator to ban a member’s account.   **Goal:**   * Disable members who post with invalid sensitive contents and avoid fraud from exchange.   **Triggers:**   * Administrator sends ban account command to the system.   **Preconditions:**   * User must login into system with role Administrator. * The profile page of selected account is displayed. * The account’s status is active.   **Post Conditions:**   * **Success**: Member’s account is banned and can’t login the system. * **Fail**: Show error message.   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | **Step** | **Actor Action** | **System Response** | | 1 | Administrator clicks on “BAN ACCOUNT” button | A dialog is displayed for administrator to confirm with message: “Are you sure?” and two buttons: “Yes” and “No” | | 2 | Administrator clicks on “Yes” button  [Alternative 1] | The status of selected account is updated to “Banned” in database  The profile page is loaded with updated status. |   **Alternative Scenario:**  **Alternative Scenario 1: Click “No” button**   |  |  |  | | --- | --- | --- | | **Step** | **Actor Action** | **System Response** | | 1 | Administrator clicks on “No” button | The dialog is disappeared and system views the profile page. |   **Exceptions:** N/A  **Relationships:** N/A  **Business** **Rules**:   * The account’s status must be active before admin bans. | | | |

**2.3.1.4. < Administrator > Unban Account Usecase**



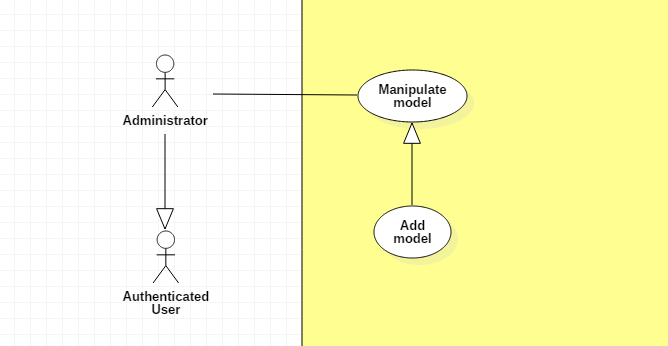
|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE – UC \_A.04** | | | |
| **Usecase No.** | A.04 | **Usecase Version** | 1.0 |
| **Usecase Name** | Unban account | | |
| **Author** | TungHNM | | |
| **Date** | 29/09/2018 | **Priority** | Normal |
| **Actor:**   * Administrator.   **Summary:**   * This use case allows administrator to unban a member’s account.   **Goal:**   * Allow a member to come back to system.   **Triggers:**   * Administrator sends unban account command to the system.   **Preconditions:**   * User must login into system with role Administrator. * The profile page of selected account is displayed. * The account’s status is banned.   **Post Conditions:**   * **Success**: Member’s account is unbanned and can login the system. * **Fail**: Show error message.   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | **Step** | **Actor Action** | **System Response** | | 1 | Administrator clicks on “UNBAN ACCOUNT” button | A dialog is displayed for administrator to confirm with message: “Are you sure?” and two buttons: “Yes” and “No” | | 2 | Administrator clicks on “Yes” button  [Alternative 1] | The status of selected account is updated to “Active” in database  The profile page is loaded with updated status. |   **Alternative Scenario:**  **Alternative Scenario 1: Click on “No” button**   |  |  |  | | --- | --- | --- | | **Step** | **Actor Action** | **System Response** | | 1 | Administrator clicks on “No” button | The dialog is disappeared and system views the profile page. |   **Exceptions:** N/A  **Relationships:** N/A  **Business** **Rules**:   * The account’s status must be banned. | | | |

**2.3.1.5. < Administrator > Search account Usecase**

****

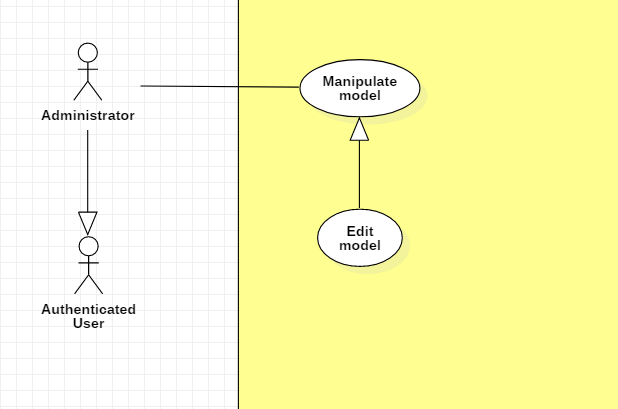
|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE – UC \_A.05** | | | |
| **Usecase No.** | A.05 | **Usecase Version** | 1.0 |
| **Usecase Name** | Search account | | |
| **Author** | TungHNM | | |
| **Date** | 30/09/2018 | **Priority** | Normal |
| **Actor:**   * Administrator.   **Summary:**   * This use case allows administrator to search accounts by username.   **Goal:**   * Make it more convenient for administrator to seek for accounts.   **Triggers:**   * Administrator sends search account command to the system.   **Preconditions:**   * User must login into system with role Administrator. * The search account page is displayed.   **Post Conditions:**   * **Success**: A list of matching records are shown. * **Fail**: Show error message.   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | **Step** | **Actor Action** | **System Response** | | 1 | Administrator clicks on search navbar | The text input is displayed with blue border and placeholder: Please input keyword | | 2 | Administrator fills out the input text |  | | 3 | Administrator clicks on “SEARCH” button or press Enter  [Alternative 2] | A list of matching accounts by username will be displayed on the view  [Alternative 1] |   **Alternative Scenario:**  **Alternative Scenario 1: No matching records**   |  |  |  | | --- | --- | --- | | **Step** | **Actor Action** | **System Response** | | 1 | Administrator clicks on “SEARCH” button or press Enter | There is no matching records and system displays message No matching records ! |   **Alternative Scenario 2: List all accounts in database**   |  |  |  | | --- | --- | --- | | **Step** | **Actor Action** | **System Response** | | 1 | Administrator clicks on “SEARCH” button or press Enter with empty input string | A list of all accounts in database is shown on the view with pagination. |   **Exceptions:** N/A  **Relationships:** N/A  **Business** **Rules**:   * A list of all accounts is displayed when administrator inputs search field with empty value * System queries accounts by username which is matched with the input string | | | |

**2.3.1.6. < Administrator > Add new model Usecase**

****

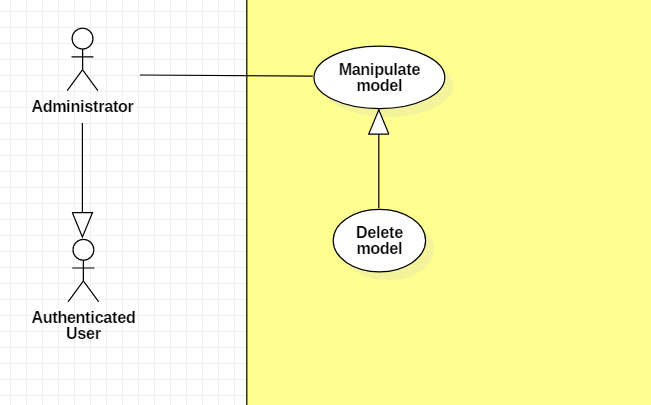
|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE – UC \_A.06** | | | |
| **Usecase No.** | A.06 | **Usecase Version** | 1.0 |
| **Usecase Name** | Add new model | | |
| **Author** | TungHNM | | |
| **Date** | 30/09/2018 | **Priority** | Normal |
| **Actor:**   * Administrator.   **Summary:**   * This use case allows administrator to create a new model.   **Goal:**   * To add new model to system.   **Triggers:**   * Administrator sends add new model command to the system.   **Preconditions:**   * User must login into system with role Administrator. * The create new model page is displayed.   **Post Conditions:**   * **Success**: A model with information is created and saved to database successfully. * **Fail**: Show error message.   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | **Step** | **Actor Action** | **System Response** | | 1 | Administrator goes to create new model view | An add new model form is shown with following labels and fields:   * Code: text, required with regEx: /^[A-z]+$/ * Name: text, required, max length: 50 * Product Series: dropdown list * Series Title: dropdown list * Manufacturer: dropdown list * Price: free text input * Released Date: DateTimePicker * Description: free text input, max length: 350 * Import image: input type file * Image type: dropdown list * “+”: button | | 2 | Administrator fills out the form  [Alternative 1, 2, 3] |  | | 3 | Administrator clicks on “CREATE” button | Validate data  [Exception 1, 2, 3, 5]  New model is created and saved to database.  System displays message Model {code} is saved successfully! |   **Alternative Scenario:**  **Alternative Scenario 1: Click on “Import image” button**   |  |  |  | | --- | --- | --- | | **Step** | **Actor Action** | **System Response** | | 1 | Administrator clicks on “Import image” button | System shows dialog for administrator to choose local image | | 2 | Administrator chooses an image | Validate file type  [Exception 4]  System shows the selected image’s name on the view |   **Alternative Scenario 2: Click on “+” button**   |  |  |  | | --- | --- | --- | | **Step** | **Actor Action** | **System Response** | | 1 | Administrator clicks on “+” button | System displays 1 new row with the following lables and fields in a horizontal line:   * Import image: input type file * Image type: dropdown list * “+”: button * “-”: button |   **Alternative Scenario 3: Click on “-” button**   |  |  |  | | --- | --- | --- | | **Step** | **Actor Action** | **System Response** | | 1 | Administrator clicks on “-” button | System removes the selected rows, includes:   * Import image: input type file * Image type: dropdown list * “+”: button |   **Exceptions:**   |  |  |  | | --- | --- | --- | | **No** | **Cause** | **System Response** | | 1 | Administrator does not input all the required fields | System shows error message to ask administrator to fill all the required fields | | 2 | Administrator inputs the invalid code. | System shows error message: Invalid format of code ! | | 3 | The model’s code input from administrator is duplicate | System shows error message: This code is already existed ! | | 4 | File which is imported by administrator is not type of image | System shows error message: Invalid file, please select file image only ! | | 5 | Length of field’s value is out of range | System shows error message that notify administrator which field’s value is out of range. |   **Relationships:** N/A  **Business** **Rules**:   * Model’s code must be characters. * Each of local image which is imported by administrator must go with an image type: dropdown list and a button “+” to add multi images. * “-” button will appear when there is at least an local image that is imported by administrator. * System checks the duplicate model’s code in database. * File which is imported by administrator must be type of images: \*.jpeg, \*.jpg, \*.png, \*.gif. | | | |

**2.3.1.7. < Administrator > Edit model Usecase**

****

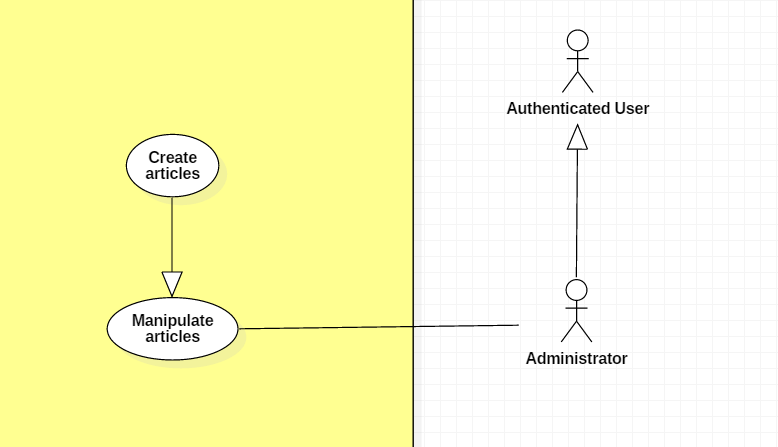
|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE – UC \_A.07** | | | |
| **Usecase No.** | A.07 | **Usecase Version** | 1.0 |
| **Usecase Name** | Edit a model | | |
| **Author** | TungHNM | | |
| **Date** | 30/09/2018 | **Priority** | Normal |
| **Actor:**   * Administrator.   **Summary:**   * This use case allows administrator to edit a gunpla model’s information.   **Goal:**   * Information about gunpla models will be customized to be more reasonable.   **Triggers:**   * Administrator sends edit model command to the system.   **Preconditions:**   * User must login into system with role Administrator. * The detail page of the model which will be edited is displayed.   **Post Conditions:**   * **Success**: Information of the selected gunpla model is updated and saved to database successfully. * **Fail**: Show error message.   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | **Step** | **Actor Action** | **System Response** | | 1 | Administrator clicks on “Edit” button on the detail page | System displays Edit Model page and a form with following labels and fields:   * Code: text, required with regEx: /^[A-z]+$/ * Name: text, required, max length: 50 * Product Series: dropdown list, * Series Title: dropdown list * Manufacturer: dropdown list * Price: free text input * Released Date: DateTimePicker * Description: free text input, max length: 350 * Import image: input type file * Image type: dropdown list * “+”: button   System automatically loads the selected gunpla model’s information to the form. | | 2 | Administrator fills out the form  [Alternative 1, 2, 3] |  | | 3 | Administrator clicks on “SAVE” button | Validate data  [Exception 1, 2, 3, 5]  Model’s information is updated and saved to database.  System displays message: Model {code} is updated successfully! |   **Alternative Scenario:**  **Alternative Scenario 1: Click on “Import image” button**   |  |  |  | | --- | --- | --- | | **Step** | **Actor Action** | **System Response** | | 1 | Administrator clicks on “Import image” button | System shows dialog for administrator to choose local image | | 2 | Administrator chooses an image | Validate file type  [Exception 4]  System shows the selected image’s name on the view |   **Alternative Scenario 2: Click on “+” button**   |  |  |  | | --- | --- | --- | | **Step** | **Actor Action** | **System Response** | | 1 | Administrator clicks on “+” button | System displays 1 new row with the following lables and fields in a horizontal line:   * Import image: input type file * Image type: dropdown list * “+”: button * “-”: button |   **Alternative Scenario 3: Click on “-” button**   |  |  |  | | --- | --- | --- | | **Step** | **Actor Action** | **System Response** | | 1 | Administrator clicks on “-” button | System removes the selected rows, includes:   * Import image: input type file * Image type: dropdown list * “+”: button * “-”: button |   **Exceptions:**   |  |  |  | | --- | --- | --- | | **No** | **Cause** | **System Response** | | 1 | Administrator does not input all the required fields | System shows error message to ask administrator to fill all the required fields | | 2 | Administrator inputs the invalid code. | System shows error message: Invalid format of code ! | | 3 | The model’s code input from administrator is duplicate | System shows error message: This code is already existed ! | | 4 | File which is imported by administrator is not type of image | System shows error message: Invalid file, please select file image only ! | | 5 | Length of field’s value is out of range | System shows error message that notify administrator which field’s value is out of range |   **Relationships:** N/A  **Business** **Rules**:   * Model’s code must be characters. * Each of local image which is imported by administrator must go with an image type: dropdown list and a button “+” to add multi images. * “-” button will appear when there is at least an local image that is imported by administrator. * System checks the duplicate model’s code in database. * File which is imported by administrator must be type of images: \*.jpeg, \*.jpg, \*.png, \*.gif. | | | |

**2.3.1.8. < Administrator > Delete model Usecase**

****

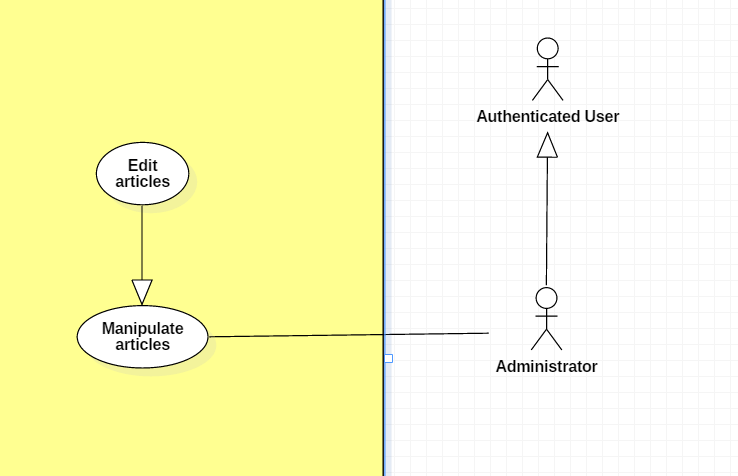
|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE – UC \_A.08** | | | |
| **Usecase No.** | A.08 | **Usecase Version** | 1.0 |
| **Usecase Name** | Delete a model | | |
| **Author** | TungHNM | | |
| **Date** | 30/09/2018 | **Priority** | Normal |
| **Actor:**   * Administrator.   **Summary:**   * This use case allows administrator to delete a gunpla model.   **Goal:**   * Models which are no longer available will be removed from system and will not be displayed on the user view.   **Triggers:**   * Administrator sends delete model command to the system.   **Preconditions:**   * User must login into system with role Administrator. * The detail page of the model which will be deleted is displayed.   **Post Conditions:**   * **Success**: Model and its information are deleted from the system. * **Fail**: Show error message.   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | **Step** | **Actor Action** | **System Response** | | 1 | Administrator clicks on “Delete” button on the detail page | System displays a dialog for administrator to confirm, the dialog is displayed with message: Are you sure ?, with 2 buttons:   * Yes * No | | 2 | Administrator clicks on “Yes” button  [Alternative 1] | Model information is deleted from database.  System displays message: Model {code} is removed from system successfully ! |   **Alternative Scenario:**  **Alternative Scenario 1: Clicks on “No” button**   |  |  |  | | --- | --- | --- | | **Step** | **Actor Action** | **System Response** | | 1 | Administrator clicks on “No” button | The dialog disappears and system views detail page of the selected gunpla model, automatically fills out all fields. |   **Exceptions:** N/A  **Relationships:** N/A  **Business** **Rules**:N/A | | | |

**2.3.1.9. < Administrator > Create articles Usecase**



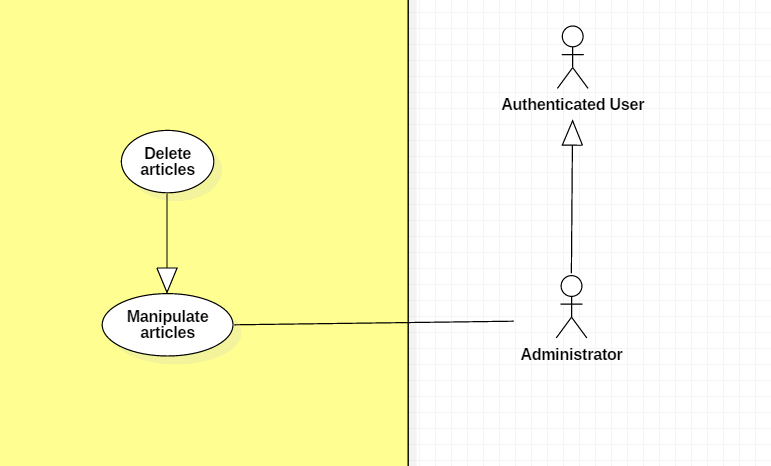
|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE – UC \_A.09** | | | |
| **Usecase No.** | A.09 | **Usecase Version** | 1.0 |
| **Usecase Name** | Create articles | | |
| **Author** | KhanhNB | | |
| **Date** | 25/09/2018 | **Priority** | Normal |
| **Actor:**   * Administrator   **Summary:**   * This use case allows user to create a new article.   **Goal:**   * A new article added to the database system.   **Triggers:**   * Admin write an articles and submit it.   **Preconditions:**   * User must login into system with role Administrator.   **Post Conditions:**   * **Success**: New article is added to the database system. * **Fail**: Show error message.   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | **Step** | **Actor Action** | **System Response** | | 1 | Admin go to “Create a new article” page | “Create new article” view is shown with following labels and fields:   * Input field for article’s Title * A textarea using CKEditor library * Submit button | | 2 | Admin fills out the form. |  | | 3 | Admin submit article. | Validate data.  Add new article the database.  System shows message article has been created successfully with a link to the article. |   **Alternative Scenario:**  N/A  **Exceptions:**   |  |  |  | | --- | --- | --- | | **No** | **Cause** | **System Response** | | 1 | Admin submit article with empty content | System shows error message to ask the admin to enters missing required fields. | | 2 | Admin upload file too large. | System shows error message to ask the admin to upload correct required file size. |   **Relationships:** N/A  **Business** **Rules**:   * In case of success scenarios, a new article would be posted to the database. * Article id is automatically initialized by the system. File upload must be file type of images: \*.jpeg, \*.jpg, \*.png, \*.gif. | | | |

**2.3.1.10. < Administrator > Edit article Usecase**

****

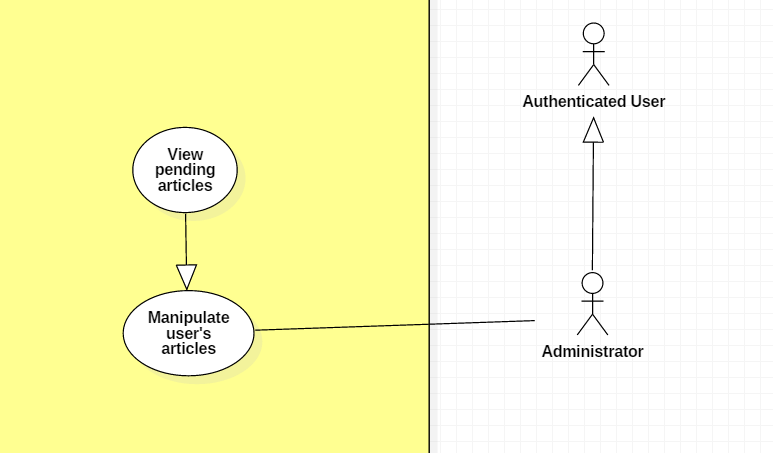
|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE – UC \_A.10** | | | |
| **Usecase No.** | A.10 | **Usecase Version** | 1.0 |
| **Usecase Name** | Edit articles | | |
| **Author** | KhanhNB | | |
| **Date** | 25/09/2018 | **Priority** | Normal |
| **Actor:**   * Administrator   **Summary:**   * This use case allows admin to edit an articles.   **Goal:**   * The articles is updated in database   **Triggers:**   * Admin edit an article and submit it.   **Preconditions:**   * User must login into system with role Administrator.   **Post Conditions:**   * **Success**: New article is updated in the database system. * **Fail**: Show error message.   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | **Step** | **Actor Action** | **System Response** | | 1 | Admin go to “Edit article” page | “Post new article” view is shown with following labels and fields:   * Input field for article’s Title * A textarea using CKEditor library with the existing data * Submit button | | 2 | Admin fills out the form. |  | | 3 | Admin submit article. | Validate data.  System update the article in the database |   **Alternative Scenario:**  N/A  **Exceptions:**   |  |  |  | | --- | --- | --- | | **No** | **Cause** | **System Response** | | 1 | Admin submit article with empty content | System shows error message to ask the admin to enters missing required fields. | | 2 | Admin upload file too large. | System shows error message to ask the admin to upload correct required file size. |   **Relationships:** N/A  **Business** **Rules**:   * In case of success scenarios, the article would be updated in the database. * Article id is automatically initialized by the system. File upload must be file type of images: \*.jpeg, \*.jpg, \*.png, \*.gif. | | | |

**2.3.1.11 < Administrator > Delete article Usecase**

****

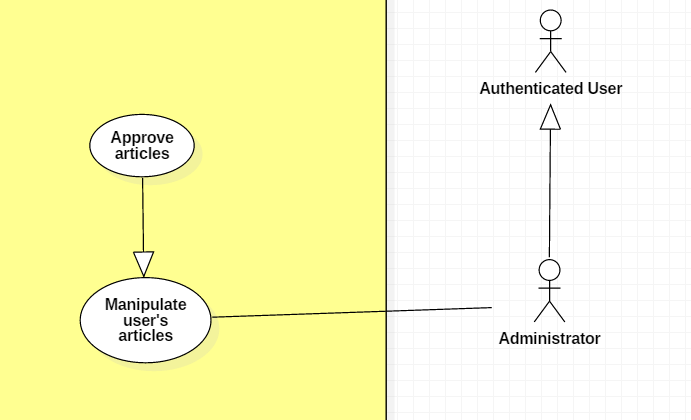
|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE – UC \_A.11** | | | |
| **Usecase No.** | A.11 | **Usecase Version** | 1.0 |
| **Usecase Name** | Delete article | | |
| **Author** | KhanhNB | | |
| **Date** | 25/09/2018 | **Priority** | Normal |
| **Actor:**   * Administrator   **Summary:**   * This use case allows admin to delete an article.   **Goal:**   * The article is deleted from the database.   **Triggers:**   * Admin choose delete article   **Preconditions:**   * User must login into system with role Administrator. * The admin is viewing an article’s detail page.   **Post Conditions:**   * **Success**: The article is deleted from the database. * **Fail**: Show error message.   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | **Step** | **Actor Action** | **System Response** | | 1 | Admin select “Delete Article” | System displays a dialog for administrator to confirm, the dialog is displayed with message: Are you sure ?, with 2 buttons:   * Yes * No | | 2 | Admin selects “Yes” | System deletes the article from database.  Message saying “This article was deleted successfully” |   **Alternative Scenario:**   |  |  |  | | --- | --- | --- | | **Step** | **Actor Action** | **System Response** | | 2 | Admin selects “No” | Dialog disappears, admin can view the article’s detail again. |   **Exceptions:**   |  |  |  | | --- | --- | --- | | **No** | **Cause** | **System Response** | | 1 | Article was already deleted | System shows error message “This article has been already deleted” |   **Relationships:** N/A  **Business** **Rules**: | | | |

**2.3.1.12 < Administrator > View pending articles Usecase**

****

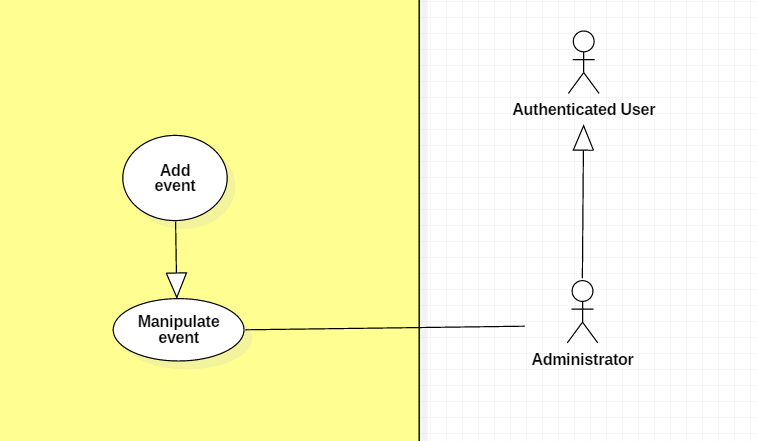
|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE – UC \_A.12** | | | |
| **Usecase No.** | A.12 | **Usecase Version** | 1.0 |
| **Usecase Name** | View pending articles | | |
| **Author** | KhanhNB | | |
| **Date** | 25/09/2018 | **Priority** | Normal |
| **Actor:**   * Administrator   **Summary:**   * This use case allows admin to view a list of pending articles   **Goal:**   * Admin can see a list of pending articles that were submitted by users   **Triggers:**   * Admin go to “Pending article” view   **Preconditions:**   * User must login into system with role Administrator.   **Post Conditions:**   * **Success**: Admin can see a list of pending articles * **Fail**: Show error message.   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | **Step** | **Actor Action** | **System Response** | | 1 | Admin select “Pending article” | System redirect the admin to the “Pending article” view  A list of articles is shown |   **Alternative Scenario:** N/A  **Exceptions:** N/A  **Relationships:** N/A  **Business** **Rules**:   * These articles are shown to the admin only * The admin can select each article to see it’s detail | | | |

**2.3.1.12 < Administrator > Approve article Usecase**

****

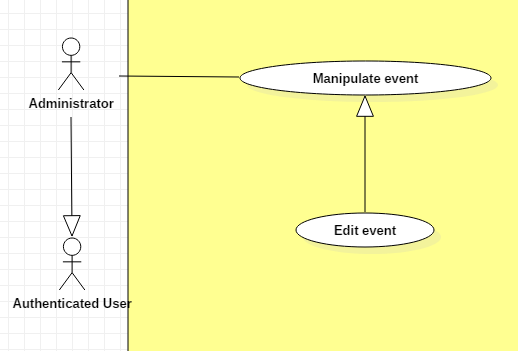
|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE – UC \_A.12** | | | |
| **Usecase No.** | A.12 | **Usecase Version** | 1.0 |
| **Usecase Name** | Approve articles | | |
| **Author** | KhanhNB | | |
| **Date** | 25/09/2018 | **Priority** | Normal |
| **Actor:**   * Administrator   **Summary:**   * This use case allows admin to approve a pending article   **Goal:**   * The article is posted on the website   **Triggers:**   * Admin is viewing a pending article posted by the user * Admin choose approve article   **Preconditions:**   * User must login into system with role Administrator.   **Post Conditions:**   * **Success**: The accepted article changed its status to ACCEPTED and is posted on the website * **Fail**: Show error message.   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | **Step** | **Actor Action** | **System Response** | | 1 | Admin select “Approve article” | System changed the status of the article to APPROVED, article is now visible on website  System redirects to the “Pending articles” view |   **Alternative Scenario:**   |  |  |  | | --- | --- | --- | | **Step** | **Actor Action** | **System Response** | | 1 | Admin select “Decline article” | System changed the status of the article to DECLINED, article is not visible on website  System redirects to the “Pending articles” view |   **Exceptions:** N/A  **Relationships:** N/A  **Business** **Rules**: N/A | | | |

**2.3.1.13 < Administrator > Create new event Usecase**

****

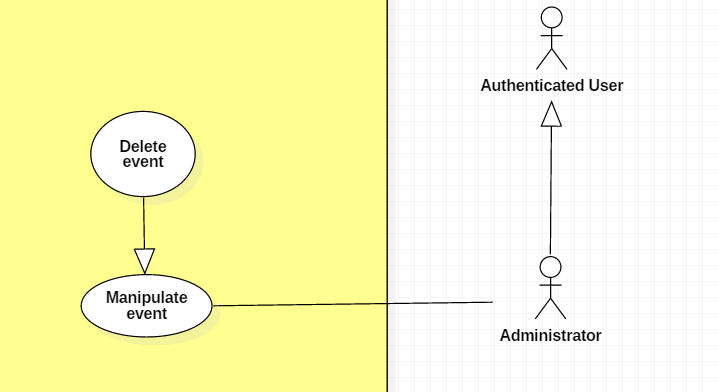
|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE – UC \_A.13** | | | |
| **Usecase No.** | A.13 | **Usecase Version** | 1.0 |
| **Usecase Name** | Create new event | | |
| **Author** | KhanhNB | | |
| **Date** | 25/09/2018 | **Priority** | Normal |
| **Actor:**   * Administrator   **Summary:**   * This use case allows admin to post a new event   **Goal:**   * The event is posted on the website   **Triggers:**   * Admin fill in the event’s info and submit   **Preconditions:**   * User must login into system with role Administrator.   **Post Conditions:**   * **Success**: The event is added to the database * **Fail**: Show error message.   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | **Step** | **Actor Action** | **System Response** | | 1 | Admin go to “Create new event” page | “Create new event” view is shown with following labels and fields:   * Input field for the event’s Title * A textarea using CKEditor library * A checkbox with label saying “Check if you want this event to require a credit card” * Input field for the event’s number of tickets * Input field for the event’s start date * Input field for the event’s end date * Input field for the event’s ticket price * Submit button | | 2 | Admin fill out form |  | | 3 | Admin submit | System add the event to the database, the event is now posted on the website |   **Alternative Scenario:**  N/A  **Exceptions:**   |  |  |  | | --- | --- | --- | | **No** | **Cause** | **System Response** | | 1 | Admin submit with empty content | System shows error message to ask the admin to enters missing required fields. | | 2 | Admin upload file too large. | System shows error message to ask the admin to upload correct required file size. |   **Relationships:** N/A  **Business** **Rules**: | | | |

**2.3.1.14. < Administrator > Edit event Usecase**

****

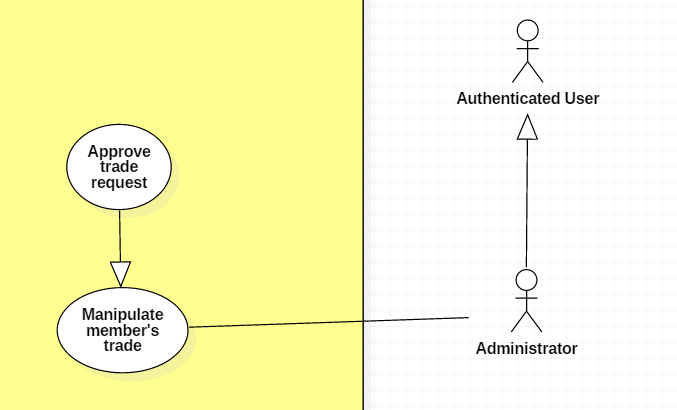
|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE – UC \_A.14** | | | |
| **Usecase No.** | A.14 | **Usecase Version** | 1.0 |
| **Usecase Name** | Update event | | |
| **Author** | KhanhNB | | |
| **Date** | 25/09/2018 | **Priority** | Normal |
| **Actor:**   * Administrator   **Summary:**   * This use case allows admin to update an event’s info.   **Goal:**   * The event is updated in database   **Triggers:**   * Admin edit an event and submit it.   **Preconditions:**   * User must login into system with role Administrator.   **Post Conditions:**   * **Success**: The event is updated in the database system. * **Fail**: Show error message.   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | **Step** | **Actor Action** | **System Response** | | 1 | Admin go to “Edit event” page | “Edit event” view is shown with following labels and fields with the old data:   * Input field for the event’s Title * A textarea using CKEditor library * A checkbox with label saying “Check if you want this event to require a credit card” * Input field for the event’s number of tickets * Input field for the event’s start date * Input field for the event’s end date * Input field for the event’s ticket price * Save button * Delete event button | | 2 | Admin fills out the form. |  | | 3 | Admin submit event. | Validate data.  System update the article in the database |   **Alternative Scenario:**  N/A  **Exceptions:**   |  |  |  | | --- | --- | --- | | **No** | **Cause** | **System Response** | | 1 | Admin submit with empty content | System shows error message to ask the admin to enters missing required fields. | | 2 | Admin upload file too large. | System shows error message to ask the admin to upload correct required file size. |   **Relationships:** N/A  **Business** **Rules**:   * In case of success scenarios, the event would be updated in the database. * Article id is automatically initialized by the system. File upload must be file type of images: \*.jpeg, \*.jpg, \*.png, \*.gif. | | | |

**2.3.1.15 < Administrator > Delete event Usecase**

****

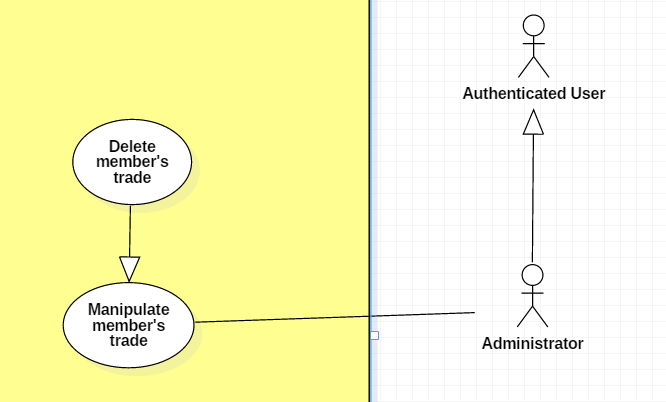
|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE – UC \_A.15** | | | |
| **Usecase No.** | A.15 | **Usecase Version** | 1.0 |
| **Usecase Name** | Delete event | | |
| **Author** | KhanhNB | | |
| **Date** | 25/09/2018 | **Priority** | Normal |
| **Actor:**   * Administrator   **Summary:**   * This use case allows admin to delete an existing event.   **Goal:**   * The event is deleted from the database.   **Triggers:**   * Admin choose delete event   **Preconditions:**   * User must login into system with role Administrator. * Admin is viewing the Edit Event View of an article.   **Post Conditions:**   * **Success**: The event is deleted from the database. * **Fail**: Show error message.   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | **Step** | **Actor Action** | **System Response** | | 1 | Admin clicks “Delete event” | System displays a dialog for administrator to confirm, the dialog is displayed with message: Are you sure ?, with 2 buttons:   * Yes * No | | 2 | Admin clicks “Yes” button  [Alternative 1] | Event is deleted from database  System displays message “Event deleted successfully” |   **Alternative Scenario:**  Alternative Scenario 1: Admin clicks on “No” button   |  |  |  | | --- | --- | --- | | **Step** | **Actor Action** | **System Response** | | 1 | Admin clicks “Yes” button  [Alternative 1] | Dialog disappears and the admin can see the Edit Event View again. |   **Exceptions:**   |  |  |  | | --- | --- | --- | | **No** | **Cause** | **System Response** | | 1 | Article was already deleted | System shows error message. |   **Relationships:** N/A  **Business** **Rules**: N/A | | | |

**2.3.1.16. < Administrator > Approve trade request Usecase**



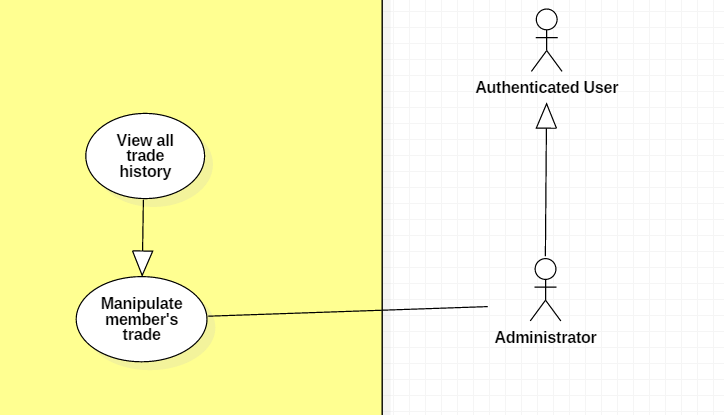
|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE – UC \_A.16** | | | |
| **Usecase No.** | A.16 | **Usecase Version** | 1.0 |
| **Usecase Name** | Approve trade post request | | |
| **Author** | SangLNN | | |
| **Date** | 26/09/2018 | **Priority** | Normal |
| **Actor:**   * Administrators.   **Summary:**   * This use case allows administrator to approve a trade post request.   **Goal:**   * After administrator approved, the trade post will listing in trade market.   **Triggers:**   * Administrator sends approve trade command with trade id to the system.   **Preconditions:**   * User must login into system with role Administrator.   **Post Conditions:**   * **Success**: System show message approved successfully and the trade post will listing in trade market. * **Fail**: Show error message.   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | **Step** | **Actor Action** | **System Response** | | 1 | Administrator click “approve” button on request trade post list to sends approve trade post command.  [Alternative 1] | System show a modal to review trade post with message “Are you sure to cancel order?” and two option Y/N. | | 2 | Administrator confirm by click “Yes”  Administrator cancel by click “No”. | System show message the trade has approved.  Nothing has change. |   **Alternative Scenario:**  Alternative Scenario 1   |  |  |  | | --- | --- | --- | | **Step** | **Actor Action** | **System Response** | | 1 | Administrator click on “Reject” button on request trade post list. | System create view show a reject form with following labels and fields:   * Reason : text input, required, length 7 – 250. * Submit: button. | | 2 | Administrator fills out the form. |  | | 3 | Administrator click “submit” button on reject form to send reject trade post command. | Validate form.  [Exception 1]  System show message the trade post has been rejected and sends reason message to member who posted this trade. |   **Exceptions:**   |  |  |  | | --- | --- | --- | | **No** | **Cause** | **System Response** | | 1 | Administrator sends submit reject trade post command without fills out the form. | System shows error message to ask administrator enters missing required fields. |   **Relationships:** N/A  **Business** **Rules**: N/A | | | |

**2.3.1.17. < Administrator > Delete member’s exchange Usecase**

****

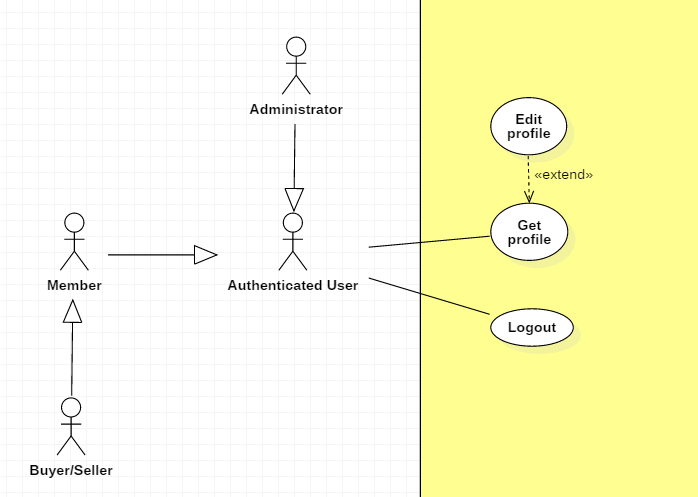
|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE – UC \_A.17** | | | |
| **Usecase No.** | A.17 | **Usecase Version** | 1.0 |
| **Usecase Name** | Delete member trade post | | |
| **Author** | SangLNN | | |
| **Date** | 26/09/2018 | **Priority** | Normal |
| **Actor:**   * Administrators.   **Summary:**   * This use case allows administrator to delete a trade post.   **Goal:**   * When a trade post received to much report from users, administrator can review and delete this post.   **Triggers:**   * Administrator sends delete trade command with trade id to the system.   **Preconditions:**   * User must login into system with role Administrator.   **Post Conditions:**   * **Success**: System show message deleted successfully and show the reason to member who posted this post. * **Fail**: Show error message.   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | **Step** | **Actor Action** | **System Response** | | 1 | Administrator click “delete” button on reported trade post list. | System create view show a delete form with following labels and fields:   * Reason : text input, required, length 7 – 250. * Submit: button. | | 2 | Administrator fills out the form. |  | | 3 | Administrator click “Submit” on delete form to send delete trade post command. | Validate form.  [Exception 1]  System show message deleted successfully. |   **Alternative Scenario:** N/A  **Exceptions:**   |  |  |  | | --- | --- | --- | | **No** | **Cause** | **System Response** | | 1 | Administrator sends submit delete trade post command without fills out the form. | System shows error message to ask administrator enters missing required fields. |   **Relationships:** N/A  **Business** **Rules**: | | | |

**2.3.1.18. < Administrator > View all exchange history Usecase**

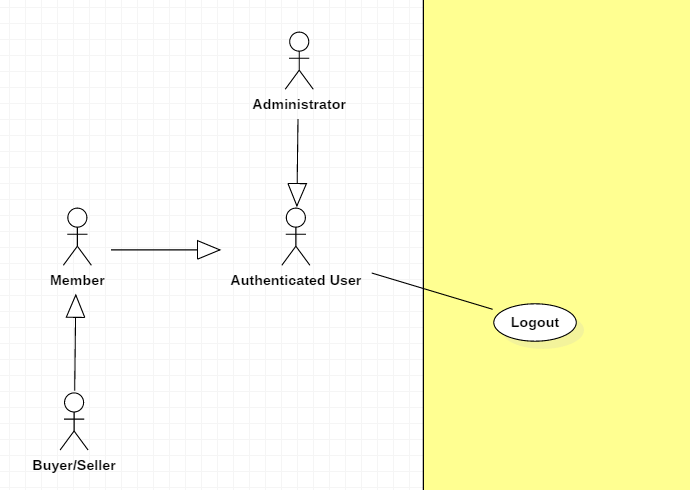
****

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE – UC \_A.18** | | | |
| **Usecase No.** | A.18 | **Usecase Version** | 1.0 |
| **Usecase Name** | View trade history | | |
| **Author** | SangLNN | | |
| **Date** | 26/09/2018 | **Priority** | Normal |
| **Actor:**   * Administrator.   **Summary:**   * This use case allows administrator view trade history.   **Goal:**   * Administrator will see all order history.   **Triggers:**   * Administrator sends view trade history command to the system.   **Preconditions:**   * User must login into system with role Administrator.   **Post Conditions:**   * **Success**: System show list of all order history. * **Fail**: Show error message.   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | **Step** | **Actor Action** | **System Response** | | 1 | Administrator click “view trade history” button on administrator panel to sends view trade history command. | System create view show a datatable contains all information about trade history. |   **Alternative Scenario:** N/A  **Exceptions:** N/A  **Relationships:** N/A  **Business** **Rules**:N/A | | | |

**2.3.2. < Authenticated User > Overview Usecase**

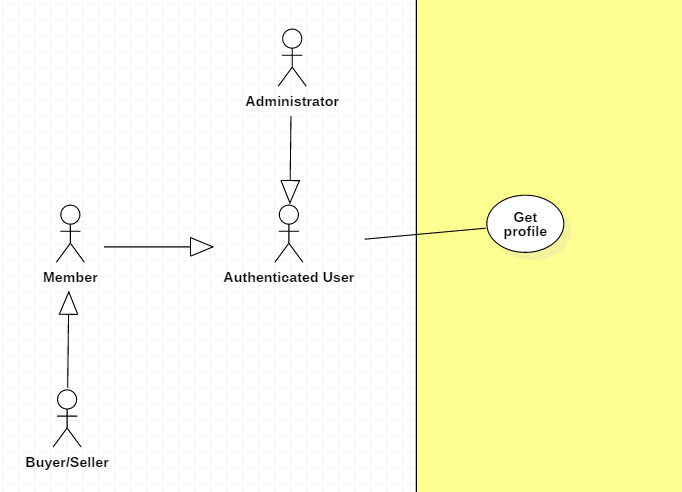


**2.3.2.1. < Authenticated User > Log out Usecase**

****

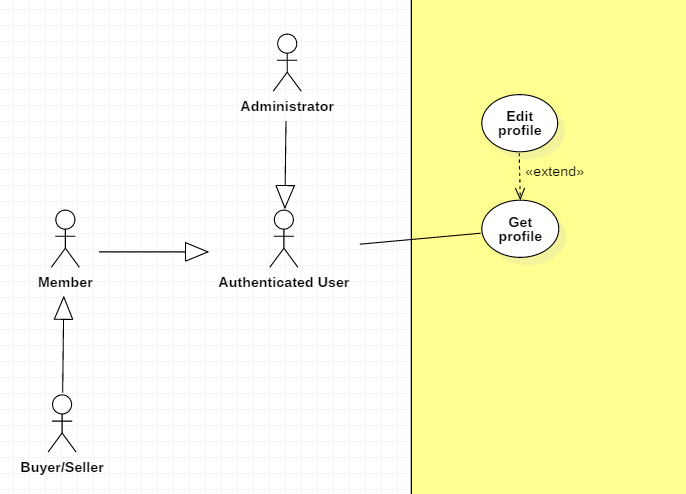
|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE – UC \_AU.01** | | | |
| **Usecase No.** | AU.01 | **Usecase Version** | 1.0 |
| **Usecase Name** | Log out | | |
| **Author** | TungHNM | | |
| **Date** | 03/10/2018 | **Priority** | Normal |
| **Actor:**   * Authenticated User.   **Summary:**   * This use case allows authenticated user to logout the system.   **Goal:**   * User’s session is removed from the system.   **Triggers:**   * Authenticated user sends logout command to the system.   **Preconditions:**   * User logged in the system.   **Post Conditions:**   * **Success**: Authenticated user logs out with session is removed from system. * **Fail**: Show error message.   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | **Step** | **Actor Action** | **System Response** | | 1 | Authenticated user clicks on “Logout” button | Authenticated user exits the system |   **Alternative Scenario:** N/A  **Exceptions:** N/A  **Relationships:** N/A  **Business** **Rules**:   * If authenticated user is inactive in 30 minutes, system will automatically log authenticated user out. | | | |

**2.3.2.2. < Authenticated User > Get profile Usecase**

****

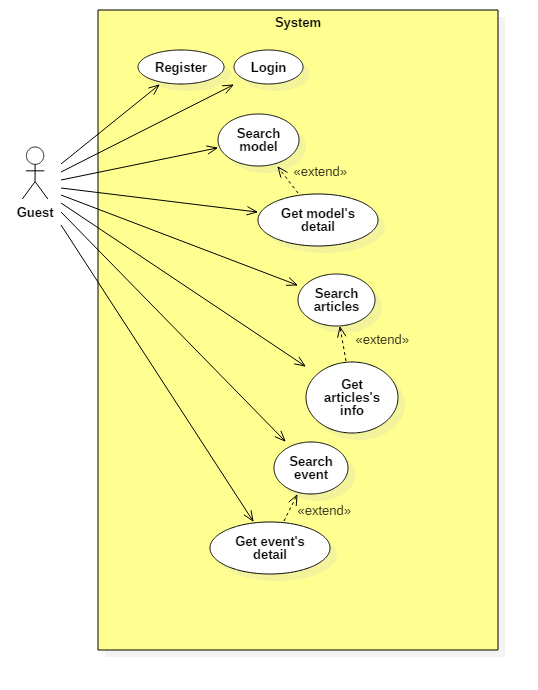
|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE – UC \_AU.02** | | | |
| **Usecase No.** | AU.02 | **Usecase Version** | 1.0 |
| **Usecase Name** | Get profile | | |
| **Author** | TungHNM | | |
| **Date** | 03/10/2018 | **Priority** | Normal |
| **Actor:**   * Authenticated User.   **Summary:**   * This use case allows authenticated user to get his/her or other’s profile’s detail.   **Goal:**   * Authenticated user can get information from his/her or other’s profile’s detail.   **Triggers:**   * Authenticated user sends get profile command to the system.   **Preconditions:**   * User must login into the system.   **Post Conditions:**   * **Success**: User’s profile page is shown with information. * **Fail**: Show error message.   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | **Step** | **Actor Action** | **System Response** | | 1 | Authenticated user clicks on a user’s name | System displays the selected user’s profile page with the following labels and text:   * Avatar: img * Email: text * First Name: text * Middle Name: text * Last Name: text * Tel: text * Address: text * Birthday: text * Status: text |   **Alternative Scenario:** N/A  **Exceptions:** N/A  **Relationships:** N/A  **Business** **Rules**:N/A | | | |

**2.3.2.3. < Authenticated User > Edit profile Usecase**

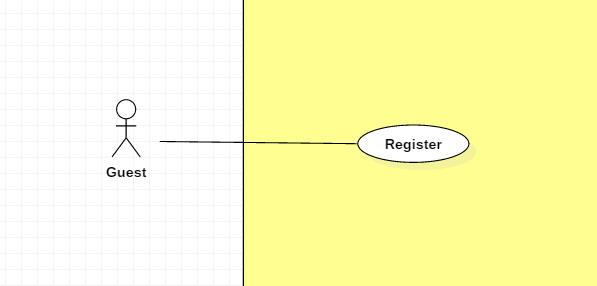
****

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE – UC \_AU.03** | | | |
| **Usecase No.** | AU.03 | **Usecase Version** | 1.0 |
| **Usecase Name** | Edit profile | | |
| **Author** | TungHNM | | |
| **Date** | 03/10/2018 | **Priority** | Normal |
| **Actor:**   * Authenticated User.   **Summary:**   * This use case allows authenticated user to update his/her profile information.   **Goal:**   * Authenticated User can edit his/her profile.   **Triggers:**   * Authenticated User sends edit profile command to the system.   **Preconditions:**   * User must login into system. * User’s profile page is displayed.   **Post Conditions:**   * **Success**: User’s information is updated. * **Fail**: Show error message.   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | **Step** | **Actor Action** | **System Response** | | 1 | Authenticated User clicks on “Edit profile” button | Edit Profile page is shown with following labels and fields:   * Email: text, required, max length: 50, RegEx: /^([a-z0-9\_\.-]+)@([\da-z\.-]+)\.([a-z\.]{2,6})$/ * First Name: text, max length: 50 * Middle Name: text, max length: 50 * Last Name: text, max length: 50 * Tel: text, length: 10-11 * Address: text, max length: 250 * Birthday: DateTimePicker * Password: text, disabled * Status: text, disabled   System automatically fills out the form with information of the selected user’s profile in database . | | 2 | User fills out the form |  | | 3 | User clicks on “SAVE” button | Validate data.  [Exception 1, 2, 3, 4, 5]  Update account information to database  System displays Profile page with updated information. |   **Alternative Scenario:** N/A  **Exceptions:**   |  |  |  | | --- | --- | --- | | **No** | **Cause** | **System Response** | | 1 | Email does not match the required format: /^([a-z0-9\_\.-]+)@([\da-z\.-]+)\.([a-z\.]{2,6})$/ | System shows error message: Invalid email ! | | 2 | Phone number is not string of number | System shows error message: Invalid phone number ! | | 3 | Email is duplicate in database | System shows error message: The email has been existed ! | | 4 | Missing of required fields | System shows message that notify user to input the missed fields | | 5 | Length of field’s value is out of range | System shows message that notify user which field’s value is out of range |   **Relationships:** N/A  **Business** **Rules**:   * User’s profile will be automatically filled out when Edit Account page is loaded or whenever an exception occurs. * In case of success scenarios, account’s information would be updated to the database and displays the profile page with updated information. * Tel must be string of numbers. * Email must not be duplicate in database. * An email address must be validated by this regular expression:   /^([a-z0-9\_\.-]+)@([\da-z\.-]+)\.([a-z\.]{2,6})$/ | | | |

**2.3.3. < Guest > Overview Usecase**

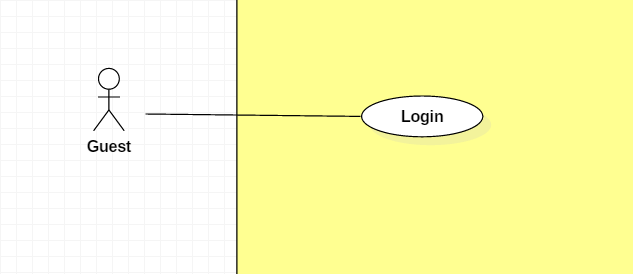


**2.3.3.1. < Guest > Register Usecase**

****

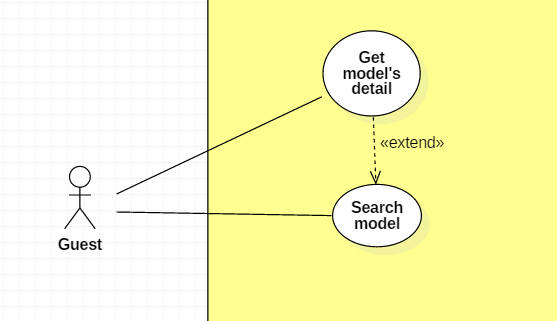
|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE – UC \_G.01** | | | |
| **Usecase No.** | G.01 | **Usecase Version** | 1.0 |
| **Usecase Name** | Register | | |
| **Author** | TungHNM | | |
| **Date** | 03/10/2018 | **Priority** | Normal |
| **Actor:**   * Guest.   **Summary:**   * This use case allows guest to register to system.   **Goal:**   * System can recognize the new account. * User can login the system with role “Member”   **Triggers:**   * Guest sends register command to the system.   **Preconditions:** N/A  **Post Conditions:**   * **Success**: New account is added to the system with role “Member”. * **Fail**: Show error message.   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | **Step** | **Actor Action** | **System Response** | | 1 | Guest clicks on “Register” button | Register page is shown with following labels and fields:   * Username: tex, required, max length: 50 * Email: text, required, max length: 50, RegEx: /^([a-z0-9\_\.-]+)@([\da-z\.-]+)\.([a-z\.]{2,6})$/ * Password: free text input, required, max length: 50 * Confirm password: free text input, required * First Name: text, max length: 50 * Last Name: text, max length: 50 | | 2 | Guest fills out the form |  | | 3 | Guest clicks on “SAVE” button | Validate data.  [Exception 1, 2, 3, 4, 5]  New account is added to system.  System displays message: Register successfully ! |   **Alternative Scenario:** N/A  **Exceptions:**   |  |  |  | | --- | --- | --- | | **No** | **Cause** | **System Response** | | 1 | Entered email does not match the required format: /^([a-z0-9\_\.-]+)@([\da-z\.-]+)\.([a-z\.]{2,6})$/ | System shows error message: Invalid email ! | | 2 | Entered email is duplicate in database | System shows error message: The email has been existed ! | | 3 | Missing of required fields | System shows error message to notify guest to input missing fields | | 4 | Password and Confirm password don’t match | System shows error message: The password and confirm password don’t match ! | | 5 | Length of field’s value is out of range | System shows message that notify guest which field’s value is out of range | | 6 | Entered username is duplicate in database | System shows error message: Username has been existed ! |   **Relationships:** N/A  **Business** **Rules**:   * The entered password and confirm password must match to each other. * Email and username must not be duplicate in database. * An email address must be validated by this regular expression:   /^([a-z0-9\_\.-]+)@([\da-z\.-]+)\.([a-z\.]{2,6})$/   * Password is encrypted before saving to database. * In case of success scenario, the account is created with default status: Member | | | |

**2.3.3.2. < Guest > Login Usecase**

****

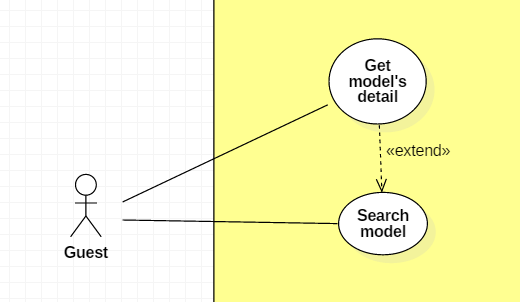
|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE – UC \_G.02** | | | |
| **Usecase No.** | G.02 | **Usecase Version** | 1.0 |
| **Usecase Name** | Login | | |
| **Author** | TungHNM | | |
| **Date** | 03/10/2018 | **Priority** | Normal |
| **Actor:**   * Guest.   **Summary:**   * This use case allows guest to login into system.   **Goal:**   * New session is created.   **Triggers:**   * Guest sends login command to the system.   **Preconditions:** N/A  **Post Conditions:**   * **Success**: Guest logins into system successfully. * **Fail**: Show error message.   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | **Step** | **Actor Action** | **System Response** | | 1 | Guest clicks on “Login” button | Login page is shown with following labels and fields:   * Username: text, required, max length: 50 * Password: text, required, max length: 50 | | 2 | Guest fills out the form |  | | 3 | Guest clicks on “Login” button | Validate data.  [Exception 1, 2, 3, 4, 5, 6]  New account is added to system.  System displays message: Register successfully ! |   **Alternative Scenario:** N/A  **Exceptions:**   |  |  |  | | --- | --- | --- | | **No** | **Cause** | **System Response** | | 1 | Missing of required fields | System shows error message to notify guest to input missing fields | | 2 | Wrong username or password | System shows error message: Wrong username or password ! |   **Relationships:** N/A  **Business** **Rules**:   * After login to system, guest will be redirected to specific page based on their role on the system:   + If role is “Admin”: the system will display Admin Dashboard view.   + If role is “Member”: the system will display Customer view with homepage | | | |

**2.3.3.3. < Guest > Search model Usecase**

****

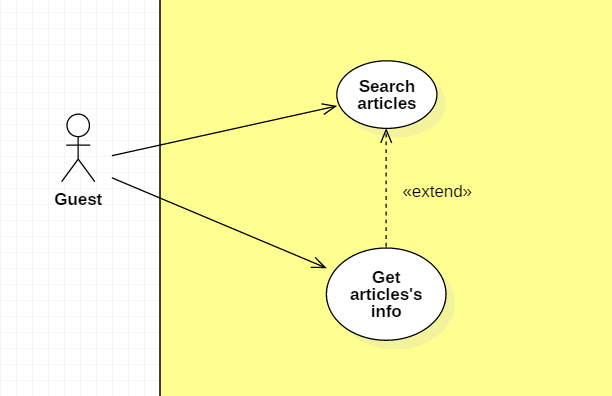
|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE – UC \_G.03** | | | |
| **Usecase No.** | G.03 | **Usecase Version** | 1.0 |
| **Usecase Name** | Search model | | |
| **Author** | TungHNM | | |
| **Date** | 03/10/2018 | **Priority** | Normal |
| **Actor:**   * Guest.   **Summary:**   * This use case allows guest to search model.   **Goal:**   * Make it more convenient for guest to seek for gunpla information.   **Triggers:**   * Guest sends search model command to the system.   **Preconditions:**   * The search model page is displayed.   **Post Conditions:**   * **Success**: A list of matching records are shown. * **Fail**: Show error message.   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | **Step** | **Actor Action** | **System Response** | | 1 | Guest goes to search model view | System displays search model view with the following elements:   * Search navbar: free text input * SEARCH BY: dropdown list, includes values:   + All: default   + Product Series   + Series Title | | 1 | Guest clicks on search navbar | An text input is displayed with border blue and placeholder: Please input keyword | | 2 | Guest fills out the input text  [Alternative 1]  [Alternative 2] |  | | 3 | Guest clicks on “SEARCH” button or press Enter  [Alternative 4]  [Alternative 5] | A list of matching models by name and Product Series or Series Title based on SEARCH BY filters will be displayed on the view with pagination and sorting filters.  [Alternative 3] |   **Alternative Scenario:**  **Alternative Scenario 1: Choose option “Product Series” in SEARCH BY dropdown list**   |  |  |  | | --- | --- | --- | | **Step** | **Actor Action** | **System Response** | | 1 | Guest clicks on option “Product Series” in SEARCH BY dropdown list | System displays a dropdown list next by with values as all ProductSeries in database. |   **Alternative Scenario 2: Choose option “GUNDAM SERIES” in SEARCH BY dropdown list**   |  |  |  | | --- | --- | --- | | **Step** | **Actor Action** | **System Response** | | 1 | Guest clicks on option “Series Title” in SEARCH BY dropdown list | System displays a dropdown list next by with values as all SeriesTitle in database. |   **Alternative Scenario 3: No matching records**   |  |  |  | | --- | --- | --- | | **Step** | **Actor Action** | **System Response** | | 1 | Guest clicks on “SEARCH” button or press Enter | There is no matching records and system displays message No matching records ! |   **Alternative Scenario 4: Click on a record**   |  |  |  | | --- | --- | --- | | **Step** | **Actor Action** | **System Response** | | 1 | Guest clicks on a record in the list | System view model detail page of the selected model. |   **Alternative Scenario 5: Choose filter “Order by” options**   |  |  |  | | --- | --- | --- | | **Step** | **Actor Action** | **System Response** | | 1 | Guest chooses an option in filter “Order by” dropdown list and option “Ascending” next by | System sorts the list records by ascending order follow with the option in “Order by” dropdown list. | | 2 | Guest chooses option “Descending” | System sorts the list records by descending order follow with the option in “Order by” dropdown list. |   **Exceptions:** N/A  **Relationships:** N/A  **Business** **Rules**:   * A dropdown list with all product series values will displayed when guest clicks on option “Product Series” in SEARCH BY dropdown list * A dropdown list with all series title values will displayed when guest clicks on option “Series Title” in SEARCH BY dropdown list * Category dropdown list or gundam series dropdown list will disappear when guest clicks on option “All” in SEARCH BY dropdown list * System search models by name matching with the keyword input from guest * A list of records will go with pagination and sorting filters | | | |

**2.3.3.4. < Guest > Get model’s detail Usecase**

****

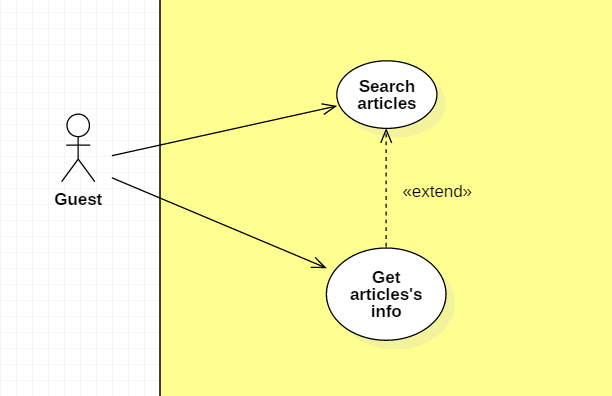
|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE – UC \_G.04** | | | |
| **Usecase No.** | G.04 | **Usecase Version** | 1.0 |
| **Usecase Name** | Get model’s detail | | |
| **Author** | TungHNM | | |
| **Date** | 03/10/2018 | **Priority** | Normal |
| **Actor:**   * Guest.   **Summary:**   * This use case allows guest user to get model’s information detail.   **Goal:**   * Guest can get information from models.   **Triggers:**   * Guest sends get model’s detail command to the system.   **Preconditions:** N/A  **Post Conditions:**   * **Success**: Model’s detail page is shown with information. * **Fail**: Show error message.   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | **Step** | **Actor Action** | **System Response** | | 1 | Guest clicks on a model and goes to model detail view | System displays the selected model’s detail page with the following labels and text:   * Big image: img * Manufacturer: text * Scale: text * Series: text * Released Date: text * Price: text * Description: text   System displays a slideshow of images divided into following categories:   * Package * Item picture * Other picture * Contents * About item * Color * Assembly guide |   **Alternative Scenario:** N/A  **Exceptions:** N/A  **Relationships:** N/A  **Business** **Rules**:N/A | | | |

**2.3.3.5. < Guest > Search articles Usecase**

****

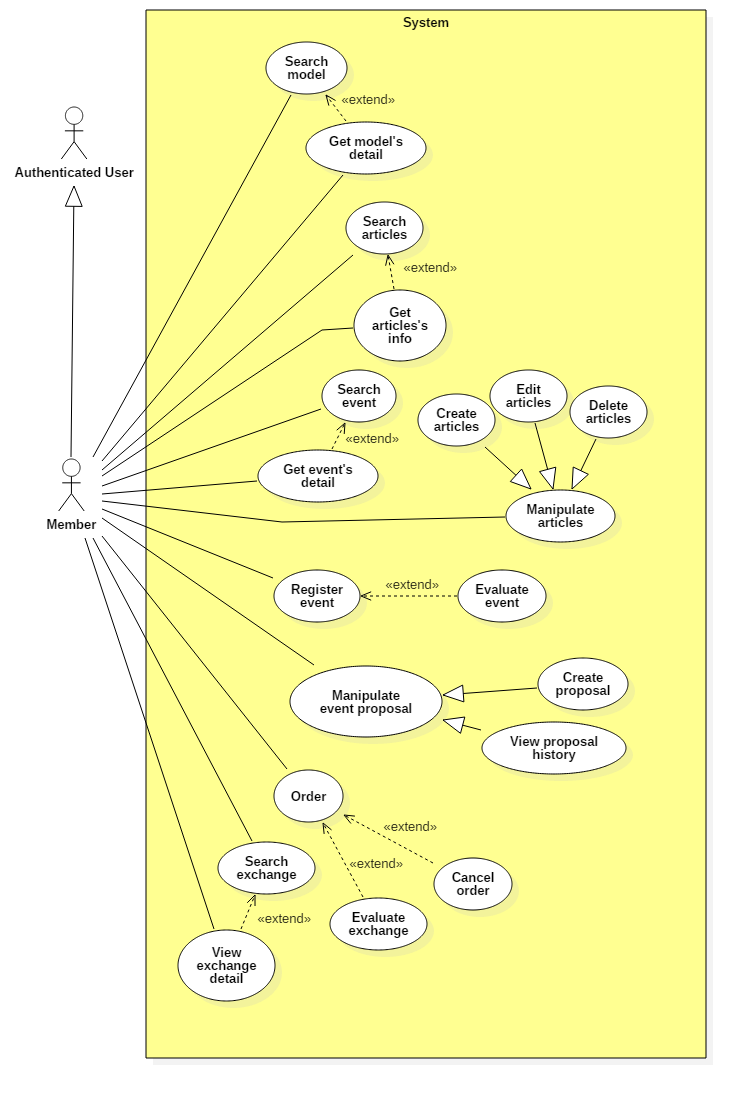
|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE – UC \_G.05** | | | |
| **Usecase No.** | G.05 | **Usecase Version** | 1.0 |
| **Usecase Name** | Search articles | | |
| **Author** | Khanhnb | | |
| **Date** | 03/10/2018 | **Priority** | Normal |
| **Actor:**   * Guest.   **Summary:**   * This use case allows guest to search articles.   **Goal:**   * Make it more convenient for guest to seek for article.   **Triggers:**   * Guest sends search news command to the system.   **Preconditions:**   * The search model news page is displayed.   **Post Conditions:**   * **Success**: A list of matching records are shown. * **Fail**: Show error message.   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | **Step** | **Actor Action** | **System Response** | | 1 | Guest goes to search news view | System displays search news view with the following elements:   * Search bar: free text input * SEARCH BY: article’s title | | 2 | Guest fills out the input text |  | | 3 | Guest clicks on “SEARCH” button or press Enter | A list of articles with matching titles is shown |   **Alternative Scenario:**  N/A  **Exceptions:** N/A  **Relationships:** N/A  **Business** **Rules**:   * System search article by title matching with the keyword input from guest * A list of records will go with pagination and sorting filters | | | |

**2.3.3.6. < Guest > Get articles’s info Usecase**

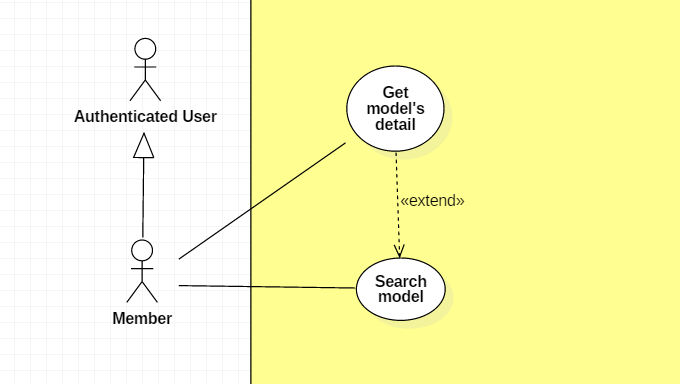
****

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE – UC \_G.06** | | | |
| **Usecase No.** | G.06 | **Usecase Version** | 1.0 |
| **Usecase Name** | Get articles’s info | | |
| **Author** | Khanhnb | | |
| **Date** | 03/10/2018 | **Priority** | Normal |
| **Actor:**   * Guest.   **Summary:**   * This use case allows guest to get an article’s info.   **Goal:**   * Display an article’s content   **Triggers:**   * Guest sends get news’s detail command to the system.   **Preconditions:**   * Guest select a article from the list of articles.   **Post Conditions:**   * **Success**: The matching record is shown. * **Fail**: Show error message.   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | **Step** | **Actor Action** | **System Response** | | 1 | Guest clicks on a model and goes to model detail view | System displays news’s detail view with the following elements:   * The title of the article: text * The article’s content |   **Alternative Scenario:**  N/A  **Exceptions:** N/A  **Relationships:** N/A  **Business** **Rules**:  N/A | | | |

**2.3.4. < Member > Overview Usecase**

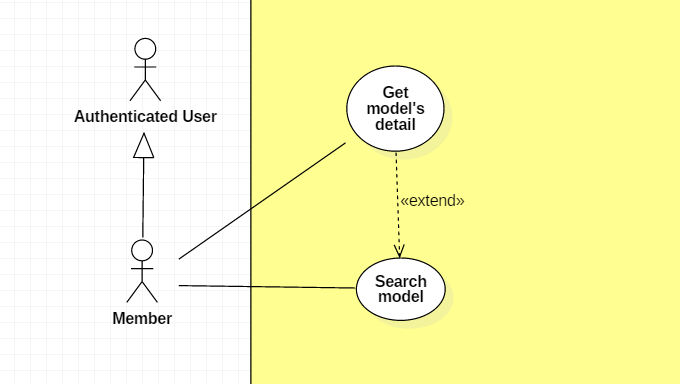


**2.3.4.1. < Member > Search model Usecase**

****

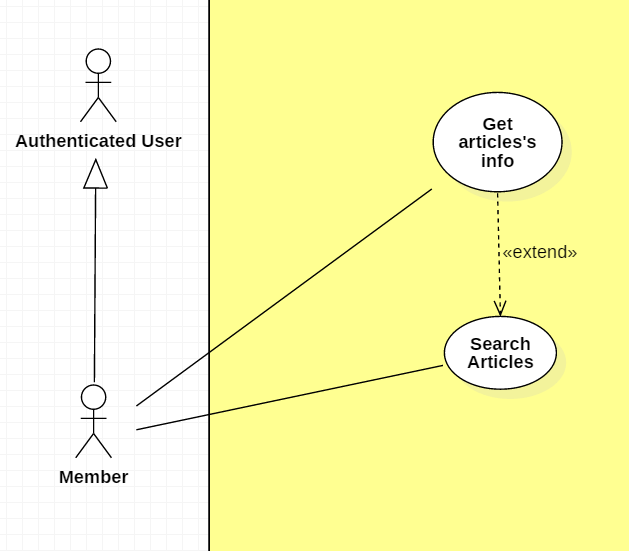
|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE – UC \_M.01** | | | |
| **Usecase No.** | M.01 | **Usecase Version** | 1.0 |
| **Usecase Name** | Search model | | |
| **Author** | TungHNM | | |
| **Date** | 03/10/2018 | **Priority** | Normal |
| **Actor:**   * Member.   **Summary:**   * This use case allows member to search model.   **Goal:**   * Make it more convenient for member to seek for gunpla information.   **Triggers:**   * Member sends search model command to the system.   **Preconditions:**   * The search model page is displayed.   **Post Conditions:**   * **Success**: A list of matching records are shown. * **Fail**: Show error message.   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | **Step** | **Actor Action** | **System Response** | | 1 | Member goes to search model view | System displays search model view with the following elements:   * Search navbar: free text input * SEARCH BY: dropdown list, includes values:   + All: default   + Product Series   + Series Title | | 1 | Member clicks on search navbar | An text input is displayed with border blue and placeholder: Enter a search term | | 2 | Member fills out the input text  [Alternative 1]  [Alternative 2] |  | | 3 | Member clicks on “SEARCH” button or press Enter  [Alternative 4]  [Alternative 5] | A list of matching models by name and Product Series or Series Title based on SEARCH BY filters will be displayed on the view with pagination and sorting filters.  [Alternative 3] |   **Alternative Scenario:**  **Alternative Scenario 1: Choose option “Product Series” in SEARCH BY dropdown list**   |  |  |  | | --- | --- | --- | | **Step** | **Actor Action** | **System Response** | | 1 | Member clicks on option “Product Series” in SEARCH BY dropdown list | System displays a dropdown list next by with values as all ProductSeries in database. |   **Alternative Scenario 2: Choose option “GUNDAM SERIES” in SEARCH BY dropdown list**   |  |  |  | | --- | --- | --- | | **Step** | **Actor Action** | **System Response** | | 1 | Member clicks on option “Series Title” in SEARCH BY dropdown list | System displays a dropdown list next by with values as all SeriesTitle in database. |   **Alternative Scenario 3: No matching records**   |  |  |  | | --- | --- | --- | | **Step** | **Actor Action** | **System Response** | | 1 | Member clicks on “SEARCH” button or press Enter | There is no matching records and system displays message No matching records ! |   **Alternative Scenario 4: Click on a record**   |  |  |  | | --- | --- | --- | | **Step** | **Actor Action** | **System Response** | | 1 | Member clicks on a record in the list | System view model detail page of the selected model. |   **Alternative Scenario 5: Choose filter “Order by” options**   |  |  |  | | --- | --- | --- | | **Step** | **Actor Action** | **System Response** | | 1 | Guest chooses an option in filter “Order by” dropdown list and option “Ascending” next by | System sorts the list records by ascending order follow with the option in “Order by” dropdown list. | | 2 | Guest chooses option “Descending” | System sorts the list records by descending order follow with the option in “Order by” dropdown list. |   **Exceptions:** N/A  **Relationships:** N/A  **Business** **Rules**:   * A dropdown list with all product series values will displayed when guest clicks on option “Product Series” in SEARCH BY dropdown list * A dropdown list with all series title values will displayed when guest clicks on option “Series Title” in SEARCH BY dropdown list * Category dropdown list or gundam series dropdown list will disappear when guest clicks on option “All” in SEARCH BY dropdown list * System search models by name matching with the keyword input from guest * A list of records will go with pagination and sorting filters | | | |

**2.3.4.2. < Member > Get model’s detail Usecase**

****

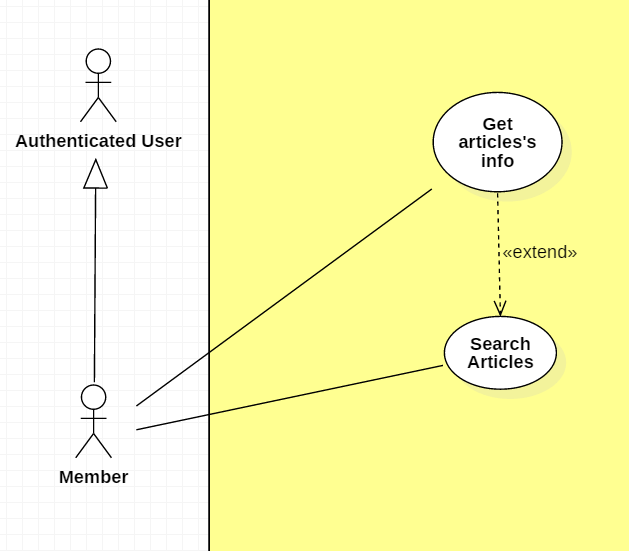
|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE – UC \_M.02** | | | |
| **Usecase No.** | M.02 | **Usecase Version** | 1.0 |
| **Usecase Name** | Get model’s detail | | |
| **Author** | TungHNM | | |
| **Date** | 03/10/2018 | **Priority** | Normal |
| **Actor:**   * Member.   **Summary:**   * This use case allows member to get model’s information detail.   **Goal:**   * Member can get information from models.   **Triggers:**   * Member sends get model’s detail command to the system.   **Preconditions:** N/A  **Post Conditions:**   * **Success**: Model’s detail page is shown with information. * **Fail**: Show error message.   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | **Step** | **Actor Action** | **System Response** | | 1 | Member clicks on a model and goes to model detail view | System displays the selected model’s detail page with the following labels and text:   * Big image: img * Manufacturer: text * Scale: text * Series: text * Released Date: text * Price: text * Description: text   System displays a slideshow of images divided into following categories:   * Package * Item picture * Other picture * Contents * About item * Color * Assembly guide |   **Alternative Scenario:** N/A  **Exceptions:** N/A  **Relationships:** N/A  **Business** **Rules**:N/A | | | |

**2.3.4.3. < Member > Search articles Usecase**

****

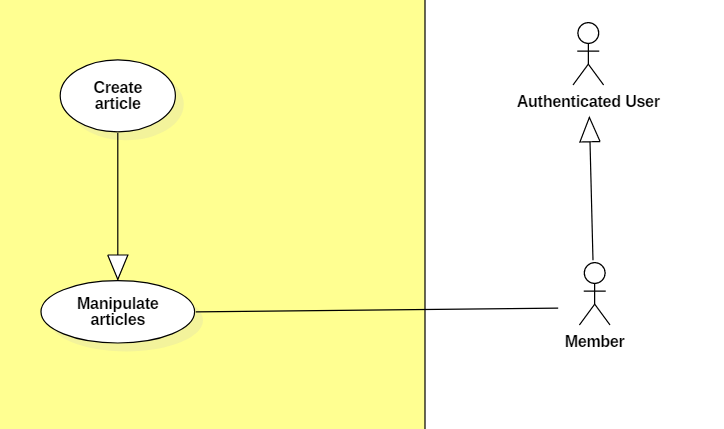
|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE – UC \_M.03** | | | |
| **Usecase No.** | M.03 | **Usecase Version** | 1.0 |
| **Usecase Name** | Search articles | | |
| **Author** | Khanhnb | | |
| **Date** | 03/10/2018 | **Priority** | Normal |
| **Actor:**   * Guest.   **Summary:**   * This use case allows member to search articles.   **Goal:**   * Make it more convenient for member to seek for article.   **Triggers:**   * Member sends search news command to the system.   **Preconditions:**   * The search news page is displayed.   **Post Conditions:**   * **Success**: A list of matching records are shown. * **Fail**: Show error message.   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | **Step** | **Actor Action** | **System Response** | | 1 | Member goes to search news view | System displays search news view with the following elements:   * Search bar: free text input * SEARCH BY: article’s title | | 2 | Member fills out the input text |  | | 3 | Member clicks on “SEARCH” button or press Enter | A list of articles with matching titles is shown |   **Alternative Scenario:**  N/A  **Exceptions:** N/A  **Relationships:** N/A  **Business** **Rules**:   * System search article by title matching with the keyword input from guest * A list of records will go with pagination and sorting filters | | | |

**2.3.4.4. < Member > Get articles’s info Usecase**

****

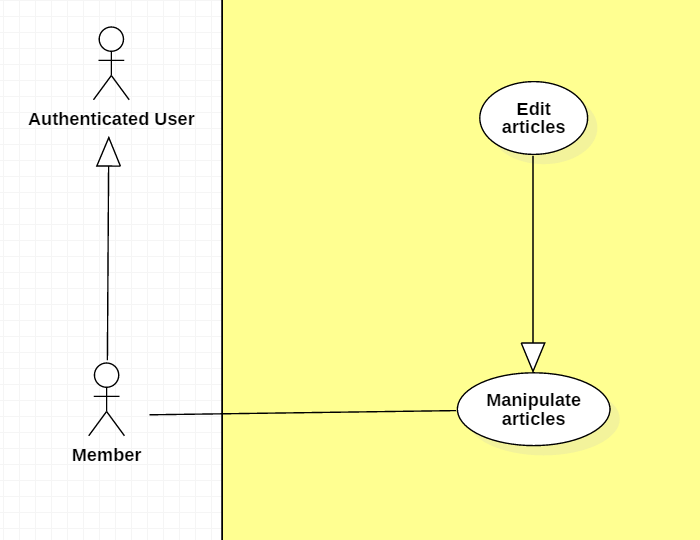
|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE – UC \_M.04** | | | |
| **Usecase No.** | M.04 | **Usecase Version** | 1.0 |
| **Usecase Name** | Get articles’s info | | |
| **Author** | Khanhnb | | |
| **Date** | 03/10/2018 | **Priority** | Normal |
| **Actor:**   * Member .   **Summary:**   * This use case allows Member to get an article’s info.   **Goal:**   * Display an article’s content   **Triggers:**   * Member sends get news’s detail command to the system.   **Preconditions:**   * Member select a article from the list of articles.   **Post Conditions:**   * **Success**: The matching record is shown. * **Fail**: Show error message.   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | **Step** | **Actor Action** | **System Response** | | 1 | Member click on a article and goes to article detail view | System displays news’s detail view with the following elements:   * The title of the article: text * The article’s content |   **Alternative Scenario:**  N/A  **Exceptions:** N/A  **Relationships:** N/A  **Business** **Rules**:  N/A | | | |

**2.3.4.5. < Administrator > Create article Usecase**



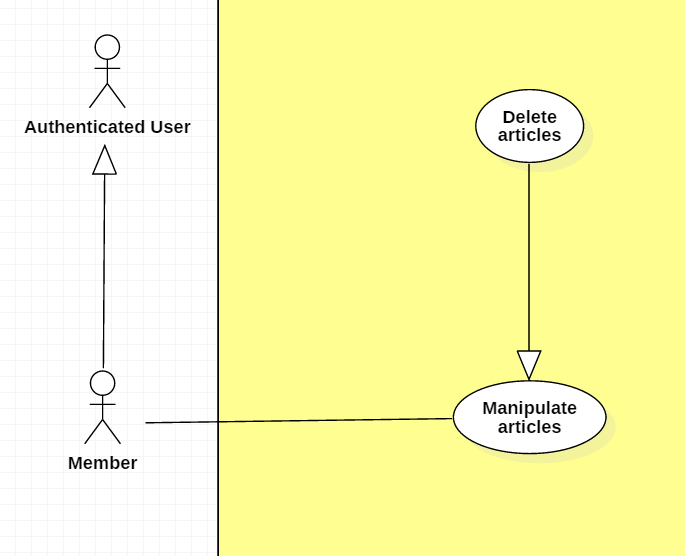
|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE – UC \_A.09** | | | |
| **Usecase No.** | A.09 | **Usecase Version** | 1.0 |
| **Usecase Name** | Create articles | | |
| **Author** | KhanhNB | | |
| **Date** | 25/09/2018 | **Priority** | Normal |
| **Actor:**   * Administrator   **Summary:**   * This use case allows user to create a new article.   **Goal:**   * A new article added to the database system.   **Triggers:**   * Admin write an articles and submit it.   **Preconditions:**   * User must login into system with role Member.   **Post Conditions:**   * **Success**: New article is added to the database system. * **Fail**: Show error message.   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | **Step** | **Actor Action** | **System Response** | | 1 | Member go to “Create a new article” page | “Create new article” view is shown with following labels and fields:   * Input field for article’s Title * A textarea using CKEditor library * Submit button | | 2 | Member fills out the form. |  | | 3 | Member submit article. | Add new article the database.  System shows message saying “Article submitted! Please wait for admin’s approval!” with a link to the article. |   **Alternative Scenario:**  N/A  **Exceptions:**   |  |  |  | | --- | --- | --- | | **No** | **Cause** | **System Response** | | 1 | Member submit article with empty content | System shows error message to ask the admin to enters missing required fields. | | 2 | Member upload file too large. | System shows error message to ask the admin to upload correct required file size. |   **Relationships:** N/A  **Business** **Rules**:   * In case of success scenarios, a new article would be posted to the database. * Article id is automatically initialized by the system. File upload must be file type of images: \*.jpeg, \*.jpg, \*.png, \*.gif. | | | |

**2.3.4.5. < Member > Edit articles Usecase**

****

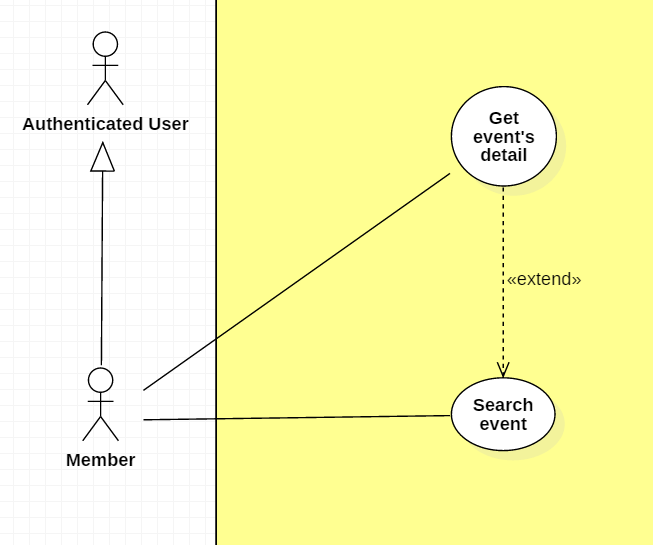
|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE – UC \_M.05** | | | |
| **Usecase No.** | M.05 | **Usecase Version** | 1.0 |
| **Usecase Name** | Edit articles | | |
| **Author** | KhanhNB | | |
| **Date** | 25/09/2018 | **Priority** | Normal |
| **Actor:**   * Member   **Summary:**   * This use case allows member to edit an articles.   **Goal:**   * The articles is updated in database   **Triggers:**   * Member edit an article and submit it.   **Preconditions:**   * User must login into system with role Member. * Member viewing an article that they posted * Member selects “Edit article”   **Post Conditions:**   * **Success**: New article is updated in the database system. * **Fail**: Show error message.   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | **Step** | **Actor Action** | **System Response** | | 1 | Member go to “Edit article” page | “Edit” view is shown with following labels and fields:   * Input field for article’s Title * A textarea using CKEditor library with the existing data * Submit button * Delete article button | | 2 | Member fills out the form. |  | | 3 | Member submit article. | System update the article in the database |   **Alternative Scenario:**  N/A  **Exceptions:**   |  |  |  | | --- | --- | --- | | **No** | **Cause** | **System Response** | | 1 | Admin submit article with empty content | System shows error message to ask the admin to enters missing required fields. | | 2 | Admin upload file too large. | System shows error message to ask the admin to upload correct required file size. |   **Relationships:** N/A  **Business** **Rules**:   * In case of success scenarios, the article would be updated in the database. * Article id is automatically initialized by the system. File upload must be file type of images: \*.jpeg, \*.jpg, \*.png, \*.gif. * Member can only edit articles they posted | | | |

**2.3.4.6. < Member > Delete articles Usecase**

****

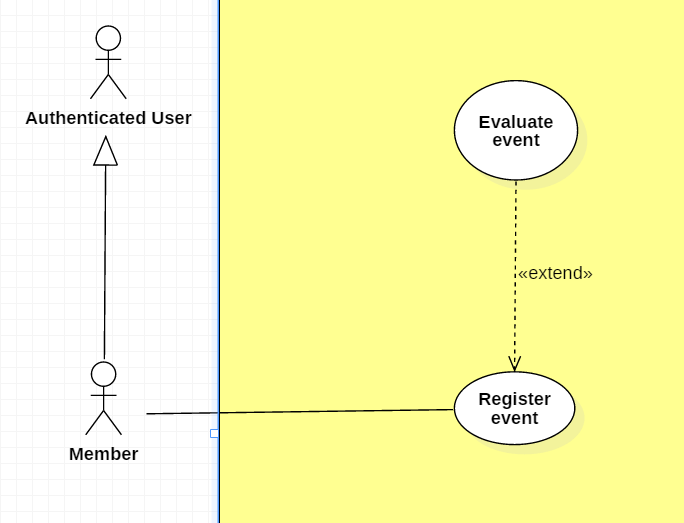
|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE – UC \_M.06** | | | |
| **Usecase No.** | M.06 | **Usecase Version** | 1.0 |
| **Usecase Name** | Delete articles | | |
| **Author** | KhanhNB | | |
| **Date** | 25/09/2018 | **Priority** | Normal |
| **Actor:**   * Member   **Summary:**   * This use case allows member to delete an articles.   **Goal:**   * The article is deleted from the database   **Triggers:**   * Member select “Delete article”   **Preconditions:**   * User must login into system with role Member. * Member viewing an article that they posted * Member selects “Delete article”   **Post Conditions:**   * **Success**: New article is updated in the database system. * **Fail**: Show error message.   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | **Step** | **Actor Action** | **System Response** | | 1 | Member select “Delete Article” | System displays a dialog for member to confirm, the dialog is displayed with message: Are you sure ?, with 2 buttons:   * Yes * No | | 2 | Member selects “Yes” | System deletes the article from database.  Message saying “This article was deleted successfully” |   **Alternative Scenario:**   |  |  |  | | --- | --- | --- | | 1 | Member selects “No” | The dialog disappears, member sees the detail page again. |   **Exceptions:**   |  |  |  | | --- | --- | --- | | **No** | **Cause** | **System Response** | | 1 | Article was already deleted | System shows error message “This article has been already deleted” |   **Relationships:** N/A  **Business** **Rules**:   * In case of success scenarios, the article would be deleted from the database. * Member can only delete articles they posted | | | |

**2.3.4.7. < Member > Search event Usecase**

****

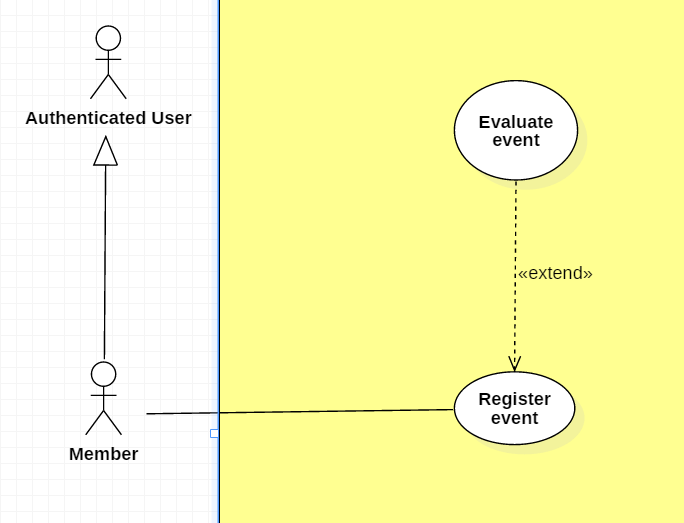
|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE – UC \_M.07** | | | |
| **Usecase No.** | M.07 | **Usecase Version** | 1.0 |
| **Usecase Name** | Search event | | |
| **Author** | KhanhNB | | |
| **Date** | 25/09/2018 | **Priority** | Normal |
| **Actor:**   * Member   **Summary:**   * This use case allows member to search for events.   **Goal:**   * Allow member to see a list of events posted on the website   **Triggers:**   * Member search for even   **Preconditions:**   * User must login into system with role Member. * Member selects “Edit article” * Member must be the author of this article   **Post Conditions:**   * **Success**: New article is updated in the database system. * **Fail**: Show error message.   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | **Step** | **Actor Action** | **System Response** | | 1 | Member go to “Edit article” page | “Edi article” view is shown with following labels and fields:   * Input field for article’s Title * A textarea using CKEditor library with the existing data * Submit button | | 2 | Member fills out the form. |  | | 3 | Member submit article. | Validate data.  System update the article in the database |   **Alternative Scenario:**  N/A  **Exceptions:**   |  |  |  | | --- | --- | --- | | **No** | **Cause** | **System Response** | | 1 | Member submit article with empty content | System shows error message to ask the admin to enters missing required fields. | | 2 | Member upload file too large. | System shows error message to ask the admin to upload correct required file size. |   **Relationships:** N/A  **Business** **Rules**:   * In case of success scenarios, the article would be updated in the database. * Article id is automatically initialized by the system. File upload must be file type of images: \*.jpeg, \*.jpg, \*.png, \*.gif. | | | |

**2.3.4.8. < Member > Register event Usecase**

****

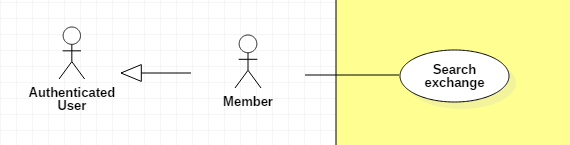
|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE – UC \_M.08** | | | |
| **Usecase No.** | M.08 | **Usecase Version** | 1.0 |
| **Usecase Name** | Register event | | |
| **Author** | Khanhnb | | |
| **Date** | 03/10/2018 | **Priority** | Normal |
| **Actor:**   * Member .   **Summary:**   * This use case allows Member to register to an event   **Goal:**   * The member is added to the event’s attendee list   **Triggers:**   * Member select “Register” while viewing an event’s detail   **Preconditions:**   * Member is viewing an event’s detail   **Post Conditions:**   * **Success**: Member successfully register to an event, member’s id is added to the list of attendees * **Fail**: Show error message.   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | **Step** | **Actor Action** | **System Response** | | 1 | Member is in Event detail View of an event that requires payment  [Alternative 1] | “Event detail” view is shown with following labels and fields:   * The event’s info is displayed * Card number text input * Number of tickets text input * Register button | | 2 | Member fill all required form |  | | 3 | Member click “Register” | System add member to the list of attendees  A message saying “You’ve been added to the list of attendees” shows. |   **Alternative Scenario:**  Alternative Scenario 1:   |  |  |  | | --- | --- | --- | | **Step** | **Actor Action** | **System Response** | | 1 | Member is in Event detail View of an event that doesn’t requires payment | “Event detail” view is shown with following labels and fields:   * The event’s info is displayed * Number of tickets text input * Register button | | 2 | Member fill all required form |  | | 3 | Member click “Register” | System add member to the list of attendees  A message saying “You’ve been added to the list of attendees” shows. |   **Exceptions:**   |  |  |  | | --- | --- | --- | | **No** | **Cause** | **System Response** | | 1 | Card number or card owner’s name is invalid | System shows Popup saying “Card is invalid! | | 2 | Number of tickets the user entered exceeds the amount of remaining tickets for that event | A message saying “The number of tickets you’ve entered exceeds the remaining tickets amount!” shows. |   **Relationships:** N/A  **Business** **Rules**:   * The system will check for the card’s validity to see if it’s real | | | |

**2.3.4.9. < Member > Evaluate event Usecase**

****

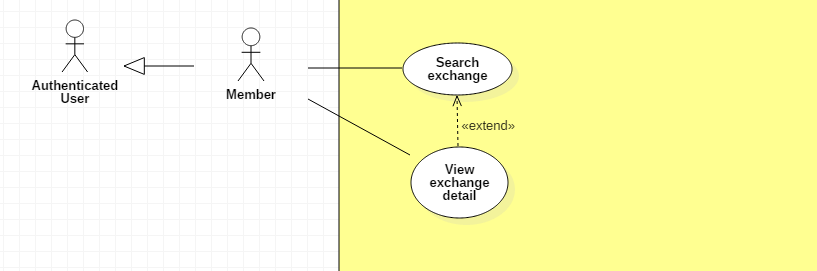
|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE – UC \_M.09** | | | |
| **Usecase No.** | M.09 | **Usecase Version** | 1.0 |
| **Usecase Name** | Evaluate event | | |
| **Author** | Khanhnb | | |
| **Date** | 03/10/2018 | **Priority** | Normal |
| **Actor:**   * Member .   **Summary:**   * This use case allows Member to evaluate to an event   **Goal:**   * The member can evaluate to show their satisfactory   **Triggers:**   * Member evaluate the event   **Preconditions:**   * User is logged in with the role “Member” * User is in the event’s list of attendees   **Post Conditions:**   * **Success**: Member successfully register to an event, member’s id is added to the list of attendees * **Fail**: Show error message.   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | **Step** | **Actor Action** | **System Response** | | 1 | Member select the star ratings | System save the member’s rating and use it to calculate the average rating for that event. |   **Alternative Scenario:** N/A  **Exceptions:** N/A  **Relationships:** N/A  **Business** **Rules**:   * The ratings range from 1 to 5 stars * The member’s rating is saved and the system will add that to the total number of rating that event has and calculate the average accordingly. | | | |

**2.3.4.10. < Member > Search exchange Usecase**



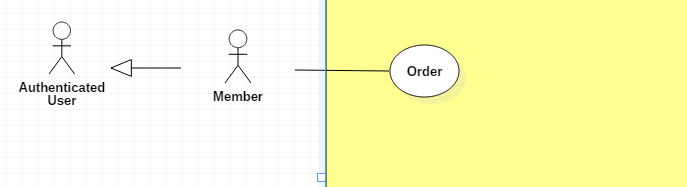
|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE – UC \_M.10** | | | |
| **Usecase No.** | M.10 | **Usecase Version** | 1.0 |
| **Usecase Name** | Search a Trade | | |
| **Author** | SangLNN | | |
| **Date** | 26/09/2018 | **Priority** | Normal |
| **Actor:**   * Member.   **Summary:**   * This use case allows member or guest search a trade by keyword and distance.   **Goal:**   * System create view to show search result.   **Triggers:**   * User sends search a trade command with keyword and distance to the system.   **Preconditions:**   * User must on website page.   **Post Conditions:**   * **Success**: System create view to show all result matching keyword and distance. * **Fail**: Show error message.   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | **Step** | **Actor Action** | **System Response** | | 1 | User click on “Trade market” link from header navbar or “Trade listing” on trade market page to send list all trade command. | System create a view show search form with following fields:   * Search input: text input , required. * “More” button   [Alternative Scenario 1] | | 2 | User type keyword and press enter to send search command. | [Exception 1]  - System create a view show all result.  - Fillter  [Alternative Scenario 2]  - Group by  [Alternative Scenario 3] |   **Alternative Scenario:**  Alternative Scenario 1: Search with location and range   |  |  |  | | --- | --- | --- | | **Step** | **Actor Action** | **System Response** | | 1 | User click on “more” button. | System will show additional search form with following labels and fields:   * Search with location and form: radio button. * Location: text input, readonly. * Change your location: button. * Range: range input, required, min value: 500, max value: 20000. | | 2 | User fills out the form. |  | | 3 | User type keyword and press enter to send search command. | [Exception 1]  - System create a view show all result.  - Fillter  [Alternative Scenario 2]  - Group by  [Alternative Scenario 3] |   Alternative Scenario 2 : Fillter Scenario.   |  |  |  | | --- | --- | --- | | **Step** | **Actor Action** | **System Response** | | 1 | User select on fillter option send fillter command. | System will list trade post sorting by user selected option. |   Alternative Scenario 3 : Group by Scenario.   |  |  |  | | --- | --- | --- | | **Step** | **Actor Action** | **System Response** | | 1 | User select on tab group option send group by command. | System will list trade post group by user selected option. |   **Exceptions:**   |  |  |  | | --- | --- | --- | | **No** | **Cause** | **System Response** | | 1 | No result found. | System shows message no result found. |   **Relationships:** N/A  **Business** **Rules**:   * Short information per post contains: Cover image, title, condition, trade type, price, brand, model. * Result list will show in datatables with pagination and 5 trade post per page. * Have 3 grouping tab: All, Selling , Buying. * Have 4 fillter: Date descending, Date ascending, Price descending, price ascending. | | | |

**2.3.4.11. < Member > View exchange detail Usecase**



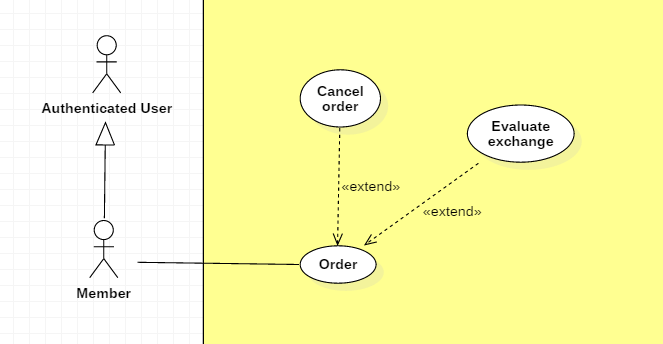
|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE – UC \_M.11** | | | |
| **Usecase No.** | M.11 | **Usecase Version** | 1.0 |
| **Usecase Name** | View trade details | | |
| **Author** | SangLNN | | |
| **Date** | 26/09/2018 | **Priority** | Normal |
| **Actor:**   * Member.   **Summary:**   * This use case allows member or guest view trade post details.   **Goal:**   * System create view to show all information for trade post.   **Triggers:**   * User sends view trade command with trade post id to the system.   **Preconditions:**   * User must on website page.   **Post Conditions:**   * **Success**: System create view to show all information for trade post. * **Fail**: Show error message.   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | **Step** | **Actor Action** | **System Response** | | 1 | User or Guest click on trade post link from trade post list to send view trade details command. | System get all information from database and create a view show informatiom in 3 section:  Product information section:   * Photo slide show * Price * Title * Trade id * Date posted * Location * Type of trade * Brand * Model * Quantity * Rating * “Send order” button. * Share this post.   Trader information section:   * Delivery * More trade by owner * Print this trade * Send to friend * Save as Favorite * Report this trade   Description section:   * Description |   **Alternative Scenario:** N/A  **Exceptions:** N/A  **Relationships:** N/A  **Business** **Rules**: N/A | | | |

**2.3.4.12. < Member > Order Usecase**



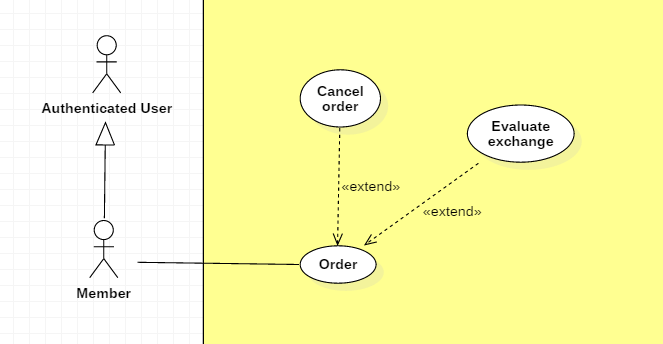
|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE – UC \_M.12** | | | |
| **Usecase No.** | M.12 | **Usecase Version** | 1.0 |
| **Usecase Name** | Order a trade | | |
| **Author** | SangLNN | | |
| **Date** | 26/09/2018 | **Priority** | Normal |
| **Actor:**   * Member.   **Summary:**   * This use case allows member to order a trade.   **Goal:**   * User can get request order one trade post to trade post owner and wating them to accept request.   **Triggers:**   * User sends order trade command with trade post id to the system.   **Preconditions:**   * User must login into system with role Member. * Member need complete all feedback request to send order.   **Post Conditions:**   * **Success**: System notify to trade post owner the request order. * **Fail**: Show error message.   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | **Step** | **Actor Action** | **System Response** | | 1 | User click “Send order” button on trade post detail view. | System create view show a order form with following labels and fields:   * Quantity: Number, required, positive integer and bellow than 1000. * Billing information: button * Cancel order: button   [Alternative scenario 1] | | 2 | User fills out the form and click “Billing information” button | Validate quantity data.  [Exception 1]  System create view show billing information form with following labels and fields:   * Full name: text, required, length 3 - 50 * Phone number: text, required, length 7 - 12 * Email: text, readonly * Billing address: text, required, length 7 - 250 * Auto get your location: button   [Alternative Scenario 2]   * Change your location: button [Alternative Scenario 3] * Confirm order: button * Cancel order: button   [Alternative scenario 1] | | 3 | User click “Confirm order” button. | System show a confirm modal with message “Are you sure to order?” and two option Y/N. | | 4 | User confirm order by click “Yes” to send order command.  User cancel confirm order by click “No” . | The order details will save to database and system will nofity to owner of this trade post this request order.  System close modal confirm and show message your order has been sent.  System close modal back to billing information view. |   **Alternative Scenario:**  Alternative Scenario 1: Cancel order   |  |  |  | | --- | --- | --- | | **Step** | **Actor Action** | **System Response** | | 1 | User click “Cancle order” on Main scenario step 1,2,3 . | System remove the form order and back to view trade post details. |   Alternative Scenario 2: Auto get your location Scenario.   |  |  |  | | --- | --- | --- | | **Step** | **Actor Action** | **System Response** | | 1 | User click “Auto get your location” button to send get your location command. | [Exception 2]  System auto get your location and generated to text address then autoform to address field. |   Alternative Scenario 3: Change your location Scenario.   |  |  |  | | --- | --- | --- | | **Step** | **Actor Action** | **System Response** | | 1 | User click “Change your location” button. | System create and show modal with following labels and fields:   * Province/City: select option, required. * District: select option, required. * Ward: select option, required. * Street name: text input, required, length 3 - 50. * Apartment number: text input, required, length 3 – 50. * Your full address: text input, readonly.   And “submit” modal button. | | 2 | User fills out the form. |  | | 3 | User click “submit” button. | System close modal and get text data from “Your full address” field and fill to “Billing address” field on trader information form. |   **Exceptions:**   |  |  |  | | --- | --- | --- | | **No** | **Cause** | **System Response** | | 1 | User input quantity is not valid. | System shows error message to ask member enters valid quantity. | | 2 | User do not allows location permission. | System shows error message to ask member provide required permission. |   **Relationships:** N/A  **Business** **Rules**:   * System auto fill billing information from profile database. * Order quantity must be lower or equal stock quantity. * Order id is automatic initialized by the system. | | | |

**2.3.4.13. < Member > Cancel order Usecase**



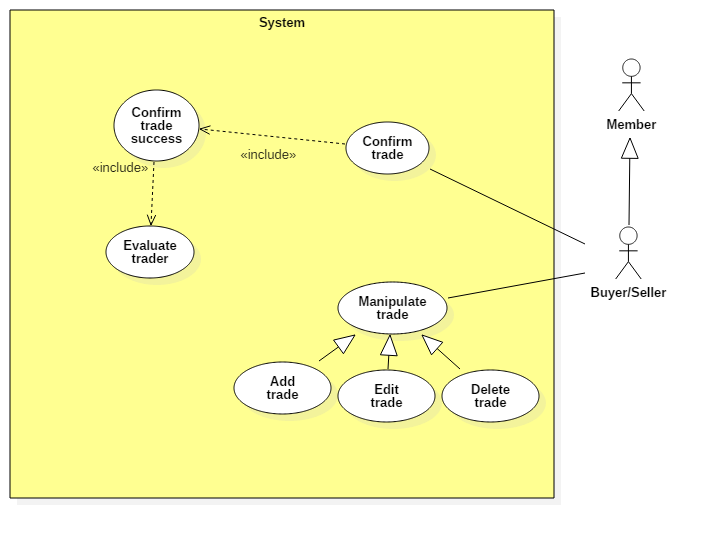
|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE – UC \_M.05** | | | |
| **Usecase No.** | M.13 | **Usecase Version** | 1.0 |
| **Usecase Name** | Cancel an order | | |
| **Author** | SangLNN | | |
| **Date** | 26/09/2018 | **Priority** | Normal |
| **Actor:**   * Member.   **Summary:**   * This use case allows member to cancel an order.   **Goal:**   * Member can cancel an order .   **Triggers:**   * Member sends cancel order command with order id to the system.   **Preconditions:**   * User must login into system with role Member. * Order on pending state. * Order has been accepted bellow 24hrs ago.   **Post Conditions:**   * **Success**: System show message to cancel order successfully and notify to buyer/seller this order has been cancelled. * **Fail**: Show error message.   **Main Success Scenario: Cancel pending order.**   |  |  |  | | --- | --- | --- | | **Step** | **Actor Action** | **System Response** | | 1 | User click “Cancel order” button on pending order list on my order page.  [Alternative scenario 1] | System show a confirm dialog with message “Are you sure to cancel order?” and two option Y/N. | | 2 | User confirm cancle order by click “Yes” to send cancel order command.  User cancel confirm cancel order by click “No” . | System remove this order request from database.  System close dialog back to pending order list view. |   **Alternative Scenario:**  Alternative Scenario 1: Cancel Accepted order Scenario.   |  |  |  | | --- | --- | --- | | **Step** | **Actor Action** | **System Response** | | 1 | User click “Cancel order” button on accepted order list on my order page. | System create and show modal with following labels and fields:   * Reason: text input, length 10 - 250.   And “submit” modal button. | | 2 | User fills out the form. |  | | 3 | User click “submit” button. | System close modal and set order state to cancelled then notify to trade post owner the reason. |   **Exceptions: N/A**  **Relationships:** N/A  **Business** **Rules**:   * User can only cancel an accepted order within 24 hours after you ordered. | | | |

**2.3.4.14. < Member > Evaluate exchange Usecase**

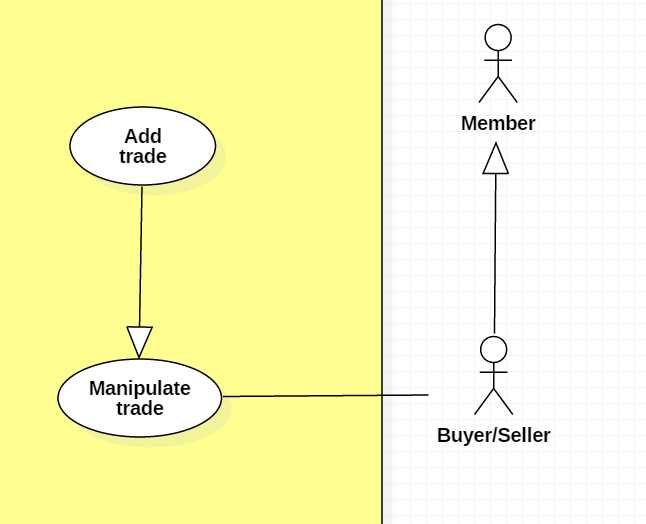


|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE – UC \_M.06** | | | |
| **Usecase No.** | M.14 | **Usecase Version** | 1.0 |
| **Usecase Name** | Evaluate a trade | | |
| **Author** | SangLNN | | |
| **Date** | 26/09/2018 | **Priority** | Normal |
| **Actor:**   * Member.   **Summary:**   * This use case allows member to feedback a trade.   **Goal:**   * After a trade has been confirm success, system will send a feedback form to member and request them to feedback the trade.   **Triggers:**   * Member sends feedback trade command with order id to the system.   **Preconditions:**   * User must login into system with role Member. * User have a trade has been confirm trade success.   **Post Conditions:**   * **Success**: System show message feedback successfully and store feedback data to database system. * **Fail**: Show error message.   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | **Step** | **Actor Action** | **System Response** | | 1 | Member click “feedback” button on histoty order list. | System create view show a feedback form with following labels and fields:   * Feedback star: radio button, required * Comment: text input, required, length 7-1000. | | 2 | Member fills out the form. |  | | 3 | Member click “submit” on feedback form to send submit feedback command. | Validate form data.  [Exception 1]  System store feeback data to database system and show message feedback successfully. |   **Alternative Scenario:** N/A  **Exceptions:**   |  |  |  | | --- | --- | --- | | **No** | **Cause** | **System Response** | | 1 | Member send submit feedback command without fills out the form. | System shows error message to ask member enters missing required fields. |   **Relationships:** N/A  **Business** **Rules**:   * Member must be do the feedback to continous send an other order. | | | |

**2.3.5. < Buyer/Seller > Overview Usecase**

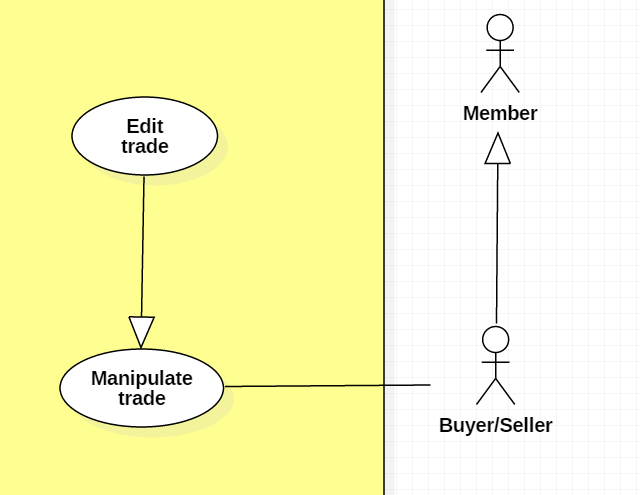


**2.3.5.1. < Buyer/Seller > Add tradeUsecase**



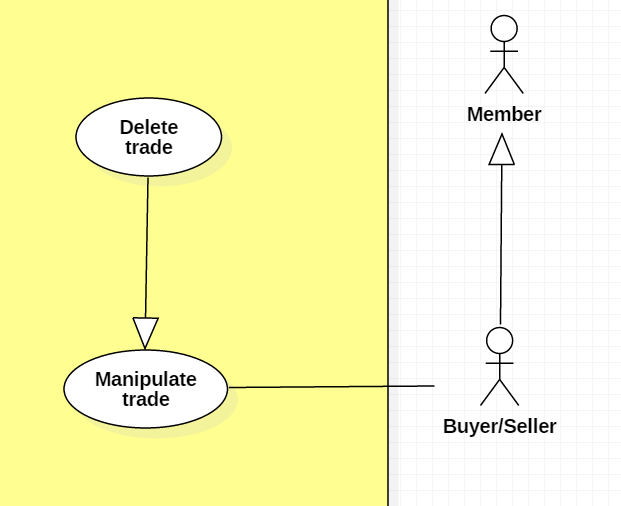
|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE – UC \_BS.01** | | | |
| **Usecase No.** | BS.01 | **Usecase Version** | 1.0 |
| **Usecase Name** | Post a new trade | | |
| **Author** | SangLNN | | |
| **Date** | 25/09/2018 | **Priority** | Normal |
| **Actor:**   * Buyer/Seller   **Summary:**   * This use case allows member to create a new trade post for trade.   **Goal:**   * A new trade post information is added to the database system.   **Triggers:**   * User sends post new trade command with post information to the system.   **Preconditions:**   * User must login into system with role Member.   **Post Conditions:**   * **Success**: New trade post is added to the database system. * **Fail**: Show error message.   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | **Step** | **Actor Action** | **System Response** | | 1 | User click “Post new trade” button on trade market page. | Create post new trade view is shown with following labels and fields:   * Type of Trade: radio button , required, 2 options: “I want to sell” and “I want to buy”. * Title: text input, required, length 7 – 250. * Photos: file input, required, number of file 1 – 5, maximum size per file is 2.5mb. * Condition : radio button , required, 2 options: “New” and “Used”. * Price: text, required, length 1 – 7. * Quantity: number, required, positive integer and bellow than 1000. * Brand: text input, required, length 3 – 50. * Model: text input, required, length 3 – 50. * Description : text input, required, length 20 – 5000.   Create trade post owner’s information view is shown with following labels and fields:   * Full name: text input, required, length 3 – 50. * Email: text input, readonly. * Phone Number: text input, required, length 7 – 12. * Address: text input, readonly. * Auto get your location: Button   [Alternative Scenario 1]   * Change your location: Button   [Alternative Scenario 2]  And “Post your trade” submit form button. | | 2 | User fills out the form. |  | | 3 | User sends post new trade command by click “Post your trade” button. | Validate data.  [Exception 1]  [Exception 2]  Add new trade post in the database.  Update trader information (optional).  System shows message trade post has been created successfully and waiting to administrators approve. |   **Alternative Scenario:**  **Alternative Scenario 1: Auto get your location Scenario.**   |  |  |  | | --- | --- | --- | | **Step** | **Actor Action** | **System Response** | | 1 | User click “Auto get your location” button to send get your location command. | [Exception 3]  System auto gets your location and generates to text address then autoforms to address field. |   **Alternative Scenario 2: Change your location Scenario.**   |  |  |  | | --- | --- | --- | | **Step** | **Actor Action** | **System Response** | | 1 | User click “Change your location” button. | System create and show modal with following labels and fields:   * Province/City: select option, required. * District: select option, required. * Ward: select option, required. * Street name: text input, required, length 3 - 50. * Apartment number: text input, required, length 3 – 50. * Your full address: text input, readonly.   And “submit” modal button. | | 2 | User fills out the form. |  | | 3 | User click “submit” button. | System close modal and get text data from “Your full address” field and fill to “Address” field on trade post owner’s information form. |   **Exceptions:**   |  |  |  | | --- | --- | --- | | **No** | **Cause** | **System Response** | | 1 | User sends post new trade command by click “Post your trade” without filling out full form. | System shows error message to ask member enters missing required fields. | | 2 | User upload file too large. | System shows error message to ask member upload correct requied file size. | | 3 | User do not allows location permission. | System shows error message to ask member provide required permission. |   **Relationships:** N/A  **Business** **Rules**:   * In case of success scenarios, a trade post would be added to the system but still waiting for administrators to approve. * System auto fills out trade post owner’s information from profile database. * Member can get their location by using auto get location service or manual input. * Phone number and price must be string of numbers. * Post trade id is automatic initialized by the system. * Member need provide a location permisstion to use auto get your location. * File upload must be file type of images: \*.jpeg, \*.jpg, \*.png, \*.gif. | | | |

**2.3.5.2. < Buyer/Seller > Edit trade Usecase**



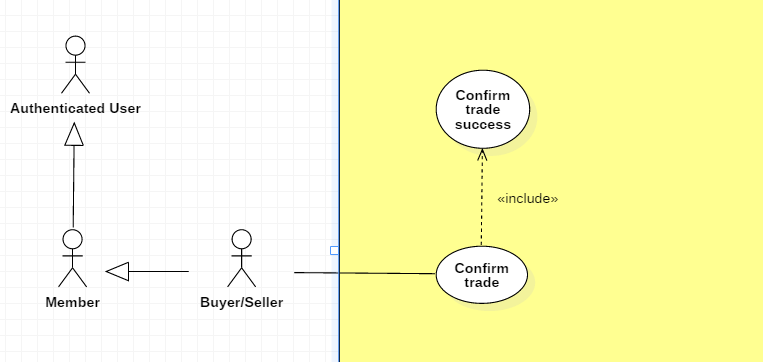
|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE – UC \_BS.02** | | | |
| **Usecase No.** | BS.02 | **Usecase Version** | 1.0 |
| **Usecase Name** | Edit a trade | | |
| **Author** | SangLNN | | |
| **Date** | 26/09/2018 | **Priority** | Normal |
| **Actor:**   * Buyer/Seller   **Summary:**   * This use case allows member to edit their trade post information.   **Goal:**   * A new trade post information is updated to the database system.   **Triggers:**   * User sends edit trade post command with updated post information to the system.   **Preconditions:**   * User must login into system with role Member. * Trade post is on pending or trade posting empty trading activity.   **Post Conditions:**   * **Success**: A new post trade information is updated to the database system. * **Fail**: Show error message.   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | **Step** | **Actor Action** | **System Response** | | 1 | User click on “Edit post” button from any trade post. | Create edit trade post view is show with following labels and fields:   * Type of Trade: radio button , required, 2 option: “I want to sell” and “I want to buy”. * Title: text input, required, length 7 – 250. * Photos: file input, required, number of file 1 – 5, maximum size per file is 2.5mb. * Condition : radio button , required, 2 option: “New” and “Used”. * Price: text, required, length 1 – 7. * Quantity: number, required, positive integer and bellow than 1000. * Brand: text input, required, length 3 – 50. * Model: text input, required, length 3 – 50. * Description : text input, required, length 20 – 5000.   Create trader information view is show with following labels and fields:   * Full name: text input, required, length 3 – 50. * Email: text input, readonly. * Phone Number: text input, required, length 7 – 12. * Address: text input, readonly. * Auto get your location: Button   [Alternative Scenario 1]   * Change your location: Button   [Alternative Scenario 2]  And “Save your change” submit form button. | | 2 | User fills out the form. |  | | 3 | User sends edit trade post command by click “Save your change” button. | Validate data.  [Exception 1]  [Exception 2]  New trade post information is updated in the database.  System shows message trade post has been updated successfully and waiting to administrators approve. |   **Alternative Scenario:** **N/A**  **Alternative Scenario:**  **Alternative Scenario 1: Auto get your location Scenario.**   |  |  |  | | --- | --- | --- | | **Step** | **Actor Action** | **System Response** | | 1 | User click “Auto get your location” button to send get your location command. | [Exception 3]  System auto get your location and generated to text address then autoform to address field. |   **Alternative Scenario 2: Change your location Scenario.**   |  |  |  | | --- | --- | --- | | **Step** | **Actor Action** | **System Response** | | 1 | User click “Change your location” button. | System create and show modal with following labels and fields:   * Province/City: select option, required. * District: select option, required. * Ward: select option, required. * Street name: text input, required, length 3 - 50. * Apartment number: text input, required, length 3 – 50. * Your full address: text input, readonly.   And “submit” modal button. | | 2 | User fills out the form. |  | | 3 | User click “submit” button. | System close modal and get text data from “Your full address” field and fill to “Address” field on trader information form. |   **Exceptions:**   |  |  |  | | --- | --- | --- | | **No** | **Cause** | **System Response** | | 1 | User sends post new trade command by click “Save your change” without fills out full form. | System shows error message to ask member enters missing required fields. | | 2 | User upload file too large. | System shows error message to ask member upload correct requied file size. | | 3 | User do not allows location permission. | System shows error message to ask member provide required permission. |   **Relationships:** N/A  **Business** **Rules**:   * “Edit trade post” button is only show on trade post is on pending or trade posting empty trading activity. * In case of success scenarios, a trade post information would be updated to the system but still waiting administrators approve. * Member can get their location by using auto get location service or manual input. * Phone number and price must be string of numbers. * Member need provide a location permisstion to use auto get your location. * File upload must be file type of images: \*.jpeg, \*.jpg, \*.png, \*.gif. | | | |

**2.3.5.3. < Buyer/Seller > Delete trade Usecase**



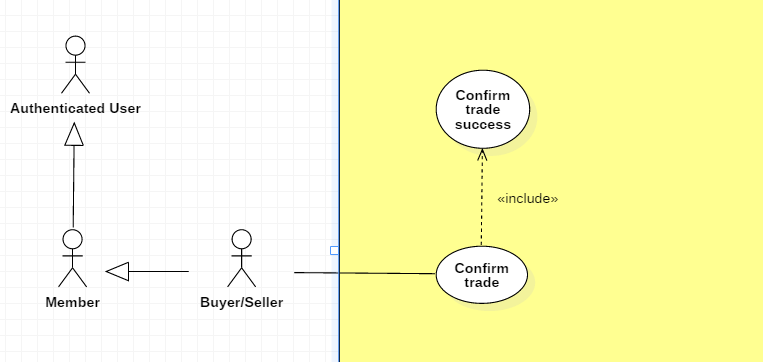
|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE – UC \_BS.03** | | | |
| **Usecase No.** | BS.03 | **Usecase Version** | 1.0 |
| **Usecase Name** | Delete a trade post | | |
| **Author** | SangLNN | | |
| **Date** | 26/09/2018 | **Priority** | Normal |
| **Actor:**   * Buyer/Seller   **Summary:**   * This use case allows member to delete a pending trade post or a trade post have no request order.   **Goal:**   * The trade post has selected is deleted from the database system.   **Triggers:**   * User sends delete trade post command with trade post id to the system.   **Preconditions:**   * User must login into system with role Member. * Trade post is on pending or trade posting empty trading activity.   **Post Conditions:**   * **Success**: The trade post has selected is deleted from the database system. * **Fail**: Show error message.   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | **Step** | **Actor Action** | **System Response** | | 1 | User click on “Delete post” button. | System show a confirm modal with message “Are you sure to delete this post?” and two option Y/N. | | 2 | User confirm delete by click “Yes” to send delete command.  User cancel delete by click “No”. | [Exception 1]  The trade post has selected is deleted from database.  System close modal and show message trade post has been deleted successfully.  System close modal and nothing has changed. |   **Alternative Scenario:** N/A  **Exceptions:**   |  |  |  | | --- | --- | --- | | **No** | **Cause** | **System Response** | | 1 | The trade post already deleted before. | System shows error message the trade post is no longer existed. |   **Relationships:** N/A  **Business** **Rules**:   * “Edit trade post” button is only show on trade post is on pending or trade posting empty trading activity. * In case of success scenarios, a trade would deleted from database system. | | | |

**2.3.5.4. < Buyer/Seller > Confirm trade Usecase**

****

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE – UC \_BS.04** | | | |
| **Usecase No.** | BS.04 | **Usecase Version** | 1.0 |
| **Usecase Name** | Confirm trade | | |
| **Author** | SangLNN | | |
| **Date** | 26/09/2018 | **Priority** | Normal |
| **Actor:**   * Buyer/Seller.   **Summary:**   * This use case allows buyer/seller to confirm a order request.   **Goal:**   * After member order a trade the request send to buyer/seller. Buyer/seller can view list of order request and choose any order to accept the order request.   **Triggers:**   * Buyer/seller sends accept request command with order id to the system.   **Preconditions:**   * User must login into system with role Member. * User have at least one request order.   **Post Conditions:**   * **Success**: System show message accepted and sends notify to who ordered. * **Fail**: Show error message.   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | **Step** | **Actor Action** | **System Response** | | 1 | User click “Accept” button on request order list to send accept request command.  [Alternative 1] | System notify to who ordered the request has been accepted and ser order state to accepted. The trade post owner information wil send to trader. |   **Alternative Scenario:**  **Alternative Scenario 1: Decline request order.**   |  |  |  | | --- | --- | --- | | **Step** | **Actor Action** | **System Response** | | 1 | If buyer/seller decline request by click “reject” button. | System create and show modal with following labels and fields:   * Reason: text input, length 10 - 250.   And “submit” modal button. | | 2 | User fills out the form. |  | | 3 | User click “submit” button. | System close modal and set order state to rejected then notify to trader the reason. |   **Exceptions: N/A**  **Relationships:** N/A  **Business** **Rules**: N/A | | | |

**2.3.5.5. < Buyer/Seller > Confirm trade success Usecase**

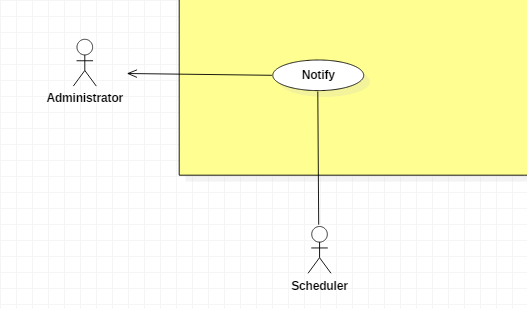
****

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE – UC \_BS.05** | | | |
| **Usecase No.** | BS.05 | **Usecase Version** | 1.0 |
| **Usecase Name** | Confirm trade success | | |
| **Author** | SangLNN | | |
| **Date** | 26/09/2018 | **Priority** | Normal |
| **Actor:**   * Buyer/Seller.   **Summary:**   * This use case allows buyer/seller to confirm a trade has been traded successfully.   **Goal:**   * After trade, buyer/seller need to confirm trade successfully.   **Triggers:**   * Buyer/seller sends confirm trade success command with order id to the system.   **Preconditions:**   * User must login into system with role Member. * User have at least one request order accepted.   **Post Conditions:**   * **Success**: System show message traded successfully and sends feedback form to who ordered. * **Fail**: Show error message.   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | **Step** | **Actor Action** | **System Response** | | 1 | User click “Confirm trade success” button on payment list to send confirm trade success command.  [Alternative Scenario 1] | System shows a confirm dialog with message “Are you sure to confirm trade success?” and two option Y/N. | | 2 | User confirm by click “Yes”  [Alternative Scenario 2] | System close dialog and set order state to success then notify to trader traded successfully. |   **Alternative Scenario:**  **Alternative Scenario 1: Reject a transaction**   |  |  |  | | --- | --- | --- | | **Step** | **Actor Action** | **System Response** | | 1 | If buyer/seller click “Cancel payment” button on payment list. | System create and show modal with following labels and fields:   * Reason: text input, length 10 - 250.   And “submit” modal button.. | | 2 | User fills out the form. |  | | 3 | User click “submit” button to send cancel payment command. | System close dialog and set order state to cancelled then notify to trader traded has been cancelled with reason. |   **Alternative Scenario 2: Confirm by click “no”**   |  |  |  | | --- | --- | --- | | **Step** | **Actor Action** | **System Response** | | 1 | User confirm by click “No” | System close dialog and nothing has change. |   **Exceptions: N/A**  **Relationships:** N/A  **Business** **Rules**: N/A | | | |

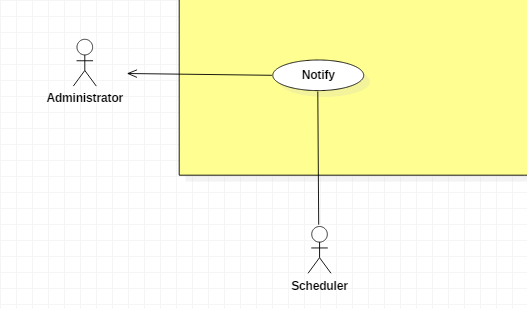
**2.3.5.5. < Buyer/Seller > Evaluate trader Usecase**

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE – UC \_BS.06** | | | |
| **Usecase No.** | BS.06 | **Usecase Version** | 1.0 |
| **Usecase Name** | Evaluate a trader | | |
| **Author** | SangLNN | | |
| **Date** | 26/09/2018 | **Priority** | Normal |
| **Actor:**   * Buyer/Seller.   **Summary:**   * This use case allows buyer/seller to feedback a trade.   **Goal:**   * After a trade has been confirm success, system will send a feedback form to owner and request them to feedback the trade.   **Triggers:**   * User sends feedback trader command with userid to the system.   **Preconditions:**   * User must login into system with role Member. * User have a trade has been confirm trade success.   **Post Conditions:**   * **Success**: System show message feedback successfully and store feedback data to database system. * **Fail**: Show error message.   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | **Step** | **Actor Action** | **System Response** | | 1 | User click “feedback” button on histoty order list. | System create view show a feedback form with following labels and fields:   * Feedback star: radio button, required * Comment: text input, required, length 7-1000. | | 2 | User fills out the form. |  | | 3 | User click “submit” on feedback form to send submit feedback command. | Validate form data.  [Exception 1]  System store feeback data to database system and show message feedback successfully. |   **Alternative Scenario:** N/A  **Exceptions:**   |  |  |  | | --- | --- | --- | | **No** | **Cause** | **System Response** | | 1 | User send submit feedback command without fills out the form. | System shows error message to ask member enters missing required fields. |   **Relationships:** N/A  **Business** **Rules**:   * Buyer/Sell must be do the feedback to continous using system. | | | |

**2.3.6. < Scheduler > Overview Usecase**



**2.3.6.1. < Scheduler > Notify Usecase**



|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE – UC \_S.01** | | | |
| **Usecase No.** | S.01 | **Usecase Version** | 1.0 |
| **Usecase Name** | Notify | | |
| **Author** | TungHNM | | |
| **Date** | 30/09/2018 | **Priority** | Normal |
| **Actor:**   * Scheduler.   **Summary:**   * Scheduler will notify notifications to users.   **Goal:**   * Make it easier for user to update information.   **Triggers:**   * Administrator approves user’s proposals. * Administrator approves user’s exchange. * Administrator update user’s profile, articles, exchange * User sends order to an exchange, notification will be sent to the exchange’s owner.   **Preconditions:** N/A  **Post Conditions:**   * **Success**: Notification is sent to user. * **Fail**: Log error to console and file..   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | **Step** | **Actor Action** | **System Response** | | 1 | Scheduler sends send notification command | System sends notification to user |   **Alternative Scenario:** N/A  **Exceptions:** N/A  **Relationships:**   * < Administrator > Approve user’s proposal * < Administrator > Approve user’s article * < Administrator > Approve user’s exchange post * < Member > orders an trade post   **Business** **Rules**:N/A | | | |

**3. Software System Attribute**

**3.1. Usability**

* UI website is fit for each browser in each device

o Font style: Helvetica, Arial, Helvetica Neue, Roboto, Arial, Droid Sans, sans-serif

o Font size: 12px -26px

o Color: green, black, red, white, blue, Light Slate Grey, Fuego, Honeysuckle…

o Background: Whitesmoke, white

**3.2. Reliability**

* Crawl data from other websites with 100% accuracy.
* Get accurate user’s location using Google Maps

**3.3. Availability**

* System replies in maximum 2 seconds

**3.4. Security**

* System always checks for authorization and authentication before doing anything.
* Input data is validated before saving to database

**3.5. Maintainability**

* The system is divided into separated modules for easy maintaining

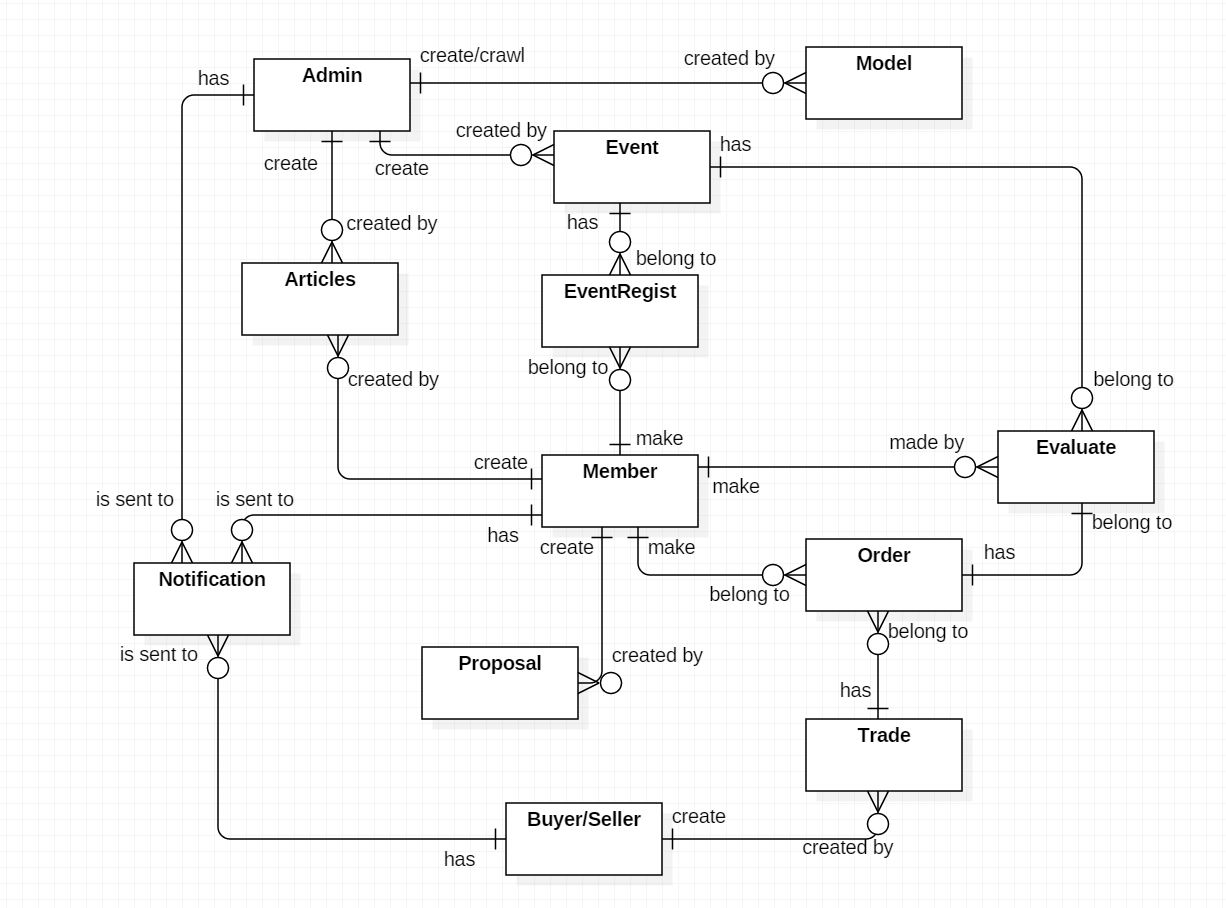
**3.6. Portability**

* Web application can run on Chrome version 42 or later, or on firefox 56 or later.

**3.7. Performance**

* Website can locate the user’s location in 2 seconds max

**4. Conceptual Diagram**

****

**Data Dictionary**

|  |  |
| --- | --- |
| **Entity Data Dictionary:** describe all contents of all entities | |
| **Entity Name** | **Description** |
| Articles | Contain information articles, sharing experience, guide which are create by administrator or members |
| Event | Contain event’s information, created by administrator |
| Model | Contain model’s information, crawled or created by administrator |
| Proposal | Contain proposal’s information that is created by member |
| EventRegist | Contain event registrations, created by member |
| Trade | Contain exchange’s information, created by Buyer/Seller |
| Order | Contain order’s information, created by member and belongs to a trade |
| Evaluate | Contain rating and feedback about events and exchange, made by member |
| Notification | Is sent to administrator when duplicate event’s calendar  Is sent to Buyer/Seller after administrator approve the exchange  Is sent to member after his/her order is approved by the trade’s owner |
| User (Administrator, Member, Buyer/Seller) | Abstract entities that describe a role in the system |