

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	

A. Report No.1: Introduction

1. Project Information

- Project name: **Gunpla World Application**
- Project code:
- Product Type: **Web app, Mobile app**
- Start Date: 10-Sep/2018
- End Date: 18-Dec/2018

2. Introduction

In this document, we introduce a hangout place for Gunpla builders community and fans. Gundam models, as well as the hobby of assembling and painting them, is known in Japan as **Gunpla**, a portmanteau of "*Gundam plastic model*".

The number of people who take up Gunpla as a hobby is increasing rapidly day by day, as well as the hobby of assembling and painting plastic models. Based on our researches and analysis, we realized that it is inconvenient for users to find information about Gunpla or to search for already-built Gunpla model or a specific part from it, which have been built or painted by other Gunpla builders.

We build a system to make it easier and more convenient for people who want to purchase Gunpla from other hobbyist or to find useful information about Gunpla. We have searching mechanism to search for users who need to exchange their products within the required or nearest range, so that buyers can seek for sellers easier. We also create a playground for Gunpla builders to share their achievements as well as experience, provide information of the Gunpla world and especially attend events, contests, or even propose them. We believe this system will attract lots of Gunpla builders and bring them closer together.

This document contains our perspective in the system, requirement analysis, component design and detailed core workflow. Our hope is that the system will bring the Gunpla community and people looking into the hobby information and new experiences .

3. Current Situation

People who have just taken up the hobby will have to search for information from many resources to resolve their questions about what Gunpla is, how to assemble or to paint them. They may also want to look for a particular Gunpla model's information, or guides and tutorial. Many Gunpla builders also look for forums to share their experience and events or contests to express their talents.

Gunpla owners may also want to exchange their models or parts with other people. There are few e-markets for exchanging Gunpla, the most popular one being Facebook groups.

4. Problem Definition

Inconvenience in searching exchangeable Gunpla: when people wants to purchase certain Gunpla parts or a finished Gunpla model assembled by other builders, they will usually post their intentions on certain Facebook groups. The sellers will comment on that post and offer their price. The same situation will happen when people want to sell their products, with the buyer commenting on the sellers' post. However, in the case of when there are a large number of posts, without searching mechanism using filters it maybe difficult for the buyers to search for what they want, this also could lead to spamming. This could leads to a lot of problems like misinformation or scam due to the way Facebook groups are moderated.

5. Proposed Solution

We proposed a solution is to build a system named "Gunpla World Application" to resolve the current situation and also continue to develop the application, making it more convenient, improving user experience and satisfying the customers.

GWA includes a web application and a mobile application with following functions:

5.1 Feature functions

Web/Mobile Application:

- Search for exchangeable Gunpla models or parts filtered by name and location suggestion
- Proposals to open events/contests

- Create events/contests
- Search for events, contests with location suggestion
- Crawl data from Gundam corporations' websites

5.2 Advantages and disadvantages

Advantages:

- It's now easier and more convenient for people who need to exchange Gunpla through the exchange forum when there are filters and location suggestions.
- Gunpla fans can propose to open events, contests by their creative opinions.
- Provide integrated Gunpla information: each Gundam model may have guides and detailed information. This is convenient for newbies who have just taken up the hobby.

Disadvantages:

- Don't have anti-spam mechanism
- Top rating may not be accurate

6. Functional Requirements

a. Gunpla model:

User component:

- Search for gunpla model's information

Staff component:

- Crawl data
- Add/modify/delete gunpla model's information

b. Information article (tutorial, sharing experience):

User component:

- Search for information article
- Add/modify/delete the user's own information article

Staff component:

- Approve/modify/delete member's information article
- Add/modify/delete staff's own information article

c. Event:

User component:

- Search events/contests using filters and location suggestion
- Proposals to open events/contests, prizes
- Register events/contests
- Evaluate by star ratings & feedback for events/contests
- View own proposals history

Staff component:

- Approve proposals to open events/contests, prizes
- Manage event calendar, location
- Add/modify/delete events/contests and rewards
- View all members's proposals history

d. Exchange:

User component:

- Search exchangeable gunpla by filters and location suggestion
- Evaluate by star ratings & feedback for the exchange
- Add/modify/delete post for exchanging gunpla
- Report the exchange post
- View own exchange history

Staff component:

- Approve/modify/delete seller's post for exchanging gunpla
- View member's reports on exchange post
- View all members's exchange history

e. Report:

- List proposals sort by time
- List exchange sort by time and good rating/feedback
- List events/contests with winners sort by time and good rating/feedback

7. Role and Responsibility

No	Full Name	Role	Position	Contact
1	Phạm Công Thành	Project Manager	Supervisor	thanhpc3@fe.edu.vn
2	Hoàng Ngô Minh Tùng	Developer	Leader	tunghnmse62231@fpt.edu.vn
3	Nguyễn Bảo Khánh	Developer	Member	khanhnbse61978@fpt.edu.vn
4	Lê Nguyễn Ngọc Sang	Developer	Member	sanglnnse61647@fpt.edu.vn

B. Report No.2 Software Project Management Plan

1. Problem Definition

1.1. Name of this Capstone Project

- Official name: Gunpla World Application
- Vietnamese name: Thế giới Gunpla
- Abbreviation: GWA

1.2. Problem Abstract

Currently in Vietnam, most people use Facebook as their primary form of communication with one another in the Gunpla community. While there are many Facebook groups that can act as a platform for trading, these “groups” are mostly moderated by volunteers. This can lead to many potential problems such as moderation quality, they might not have the skills or the tools to deal with complicated situations such as spamming or when there is a scammer.

We provide a software to act as a platform of information, hosting events and trading. We have many advanced features to heighten user experience, such as the accurate user location system we use in our trading platform. Our administrator site will also have many important feature to manage the entire system effectively.

1.3. Project Overview

1.3.1. Current Situation

Below are the problems encountered in this project:

- **Security:** While spamming can be minimized with the use of filters and admin system, an attack can create many accounts to attack the sites with spam.

- **Legal issues:** There maybe legal issues regarding the information we use on our website.
- **Absence of team members:** team members can get sick or unexpected problems.

1.3.2. The Proposed System

To help the administrator get hold of the latest and most accurate information and news, we get data from many popular and trusted sites and show them to the admin. This data includes individual Gunpla info and articles written from other websites. The admin can then use that information to help them write their articles and manage the current data.

Our website will have disclaimers about the validity of our articles and their sources to avoid any legal issues.

We also build a high available web server to maintain the main system to work 24/7 to make sure that if mobile applications need access to the information there will be always available.

We assign responsibility in vertical to make sure if any member in this problem cannot continue to work in our team there will be the least harmful to the project processes.

Our system includes three main subsystems: an administrator website to manage the system for the admins, the main web application that will be the final product, and a mobile application.

1.3.2.1 Website

The website is the main platform of our system, it is used by the admin staffs and the users (customers). The website provides the following features:

- For users (customers):
 - ❖ Users can view any Gunpla's info.

- ❖ Users can read any articles posted.
- ❖ Users can view the trade market.
- ❖ Users can register a new account.
- ❖ User with an account can write new articles and post them, the article will need admin approval to be posted.
- ❖ User with an account can post trade requests to buy or to sell on the market, the request will need admin approval to be posted.
- ❖ User with an account can write an event proposal.
- ❖ User can report an article or a trade request to the admin
- For staffs:
 - ❖ Staffs can write new Gunpla info and post them
 - ❖ Staffs can post new Gunpla info using the crawled data
 - ❖ Staffs can write new articles and post them.
 - ❖ Staffs can post new articles using the crawled data.
 - ❖ Staffs can edit or delete any Gunpla info or article
 - ❖ Staffs can approve requests to post articles or trade request from users.
 - ❖ Staffs can edit or delete any article or trade request.
 - ❖ Staffs can check event proposal sent by users
 - ❖ Staffs can post info for an event.
 - ❖ Staffs can see reports from users.
 - ❖ Staffs can ban users.

Besides above, the website system can crawl data from other websites to get Gunpla info and articles. The system can also get the current user location to get an accurate address.

1.3.3. Boundaries of the System

The system is designed to serve:

- Any Gunpla enthusiast with experience or just a newbie.
- Event hosting companies.
- Traders or any person who is looking for trading anything Gunpla related.

The system is designed with the following main functions:

- Compile Gunpla related news.

- Create a community to share knowledge, experience, tutorial or just to show off to one another.
- Let people propose events with their own ideas and let the host company review those ideas to host events.
- Create an environment to trade any Gunpla related goods.
- The system can search and suggest

The system is designed to run on these platforms:

- Devices with internet connection and internet browsers supporting HTML5, Javascript
- Devices running Android Ice Cream Sandwich or newer.

1.3.4. Future Plans

Most event organizing model and trade markets for Gunpla players are primarily on Facebook, which has limited functionalities and not tailored for Gunpla players' use. Our application was primarily built for Gunpla players and professionally tailored for their use.

- Gunpla players won't have to go to different pages and groups to find information or to trade. With Gunpla World Application, everything is concentrated in one place. The user only needs to bookmark the website address or download the mobile application to access all the information and news they need on Gunpla, and all the functions they need to trade Gunpla with other players. Everything is combined on one application.
- Professional aspects:
 - + Event host won't have to worry about false information that may happen from spam or other kind of attacks. The system will have a fee system to defend against attacks, and it can provide important information like the number of attendees for an event so the company can manage that event effectively.
 - + A registered user can propose an event from their idea and the host company can use that information to judge what is the best way to host an event.
 - + A post about selling or trading have a locating function that accurately locates the user, the sellers can also enter the address by themselves if they choose to.
 - + If you're looking to buy something, the system can show the nearest sellers that are selling what you're looking for based on your location.

1.3.5. Development Environment

1.3.5.1. Hardware requirements

Windows	Minimum requirements	Recommended
Internet connection	Cable, Wi-Fi (8 Mbps)	Cable, Wi-Fi (20 Mbps)
Operating system	Window 7	Window 10
Computer Processor	Intel® Core™ i3 @ 2.0Ghz	Intel® Core™ i5 (4 CPUs) @ 2.5GHz
Computer Memory	4GB RAM	4GB RAM or more

Mobile	Minimum requirements	Recommended
Operating System	Android 4.0.3 Ice Cream Sandwich	Android 6 Marshmallow or above
Processor	CPU dual core 1.4 GHz	CPU Quad core 2.0 GHz
Memory	1GB RAM	2GB RAM or above

1.3.5.2. Software requirements

Software	Name and version	Description
Operating System	Window 10 1803	Operating system and platform for development
Environment	Java EE 8, Android SDK APIs level 15 or above	Specification for developing web and android application
Modeling Tool	StarUML 2.8.1	Used to implement website and web service
IDE	IntelliJ IDEA 2018.2.3, Netbeans 8.0.2, Android Studio 3.1.3	Programming tools
DBMS	MySQL 5.7.20, MySQL Workbench 6.3.9 CE	Used to create & manage the database for system
Source control	SourceTree 2.6.10	Used for source control

Web browser	Chrome 62 or above	Testing browser
-------------	--------------------	-----------------

2. Project Organization

2.1. Software Process Model

This project is developed under waterfall model. It's the most appropriate process model that ensure the success and quality of project organization. We use this process model because of the following reasons:

- The project requirements are very well documented, clear and fixed without ambiguous.
- The project technology is static and well understood by all team members, the risk of changing algorithm is low.
- The whole development process needs to be departmentalized and controlled, divided into separate phases sequentially. Each phase must be completed and has specific deliverables for the next phase to begin. This is easy to arrange and manage task due to the rigidity of the model and clearly defined stages.

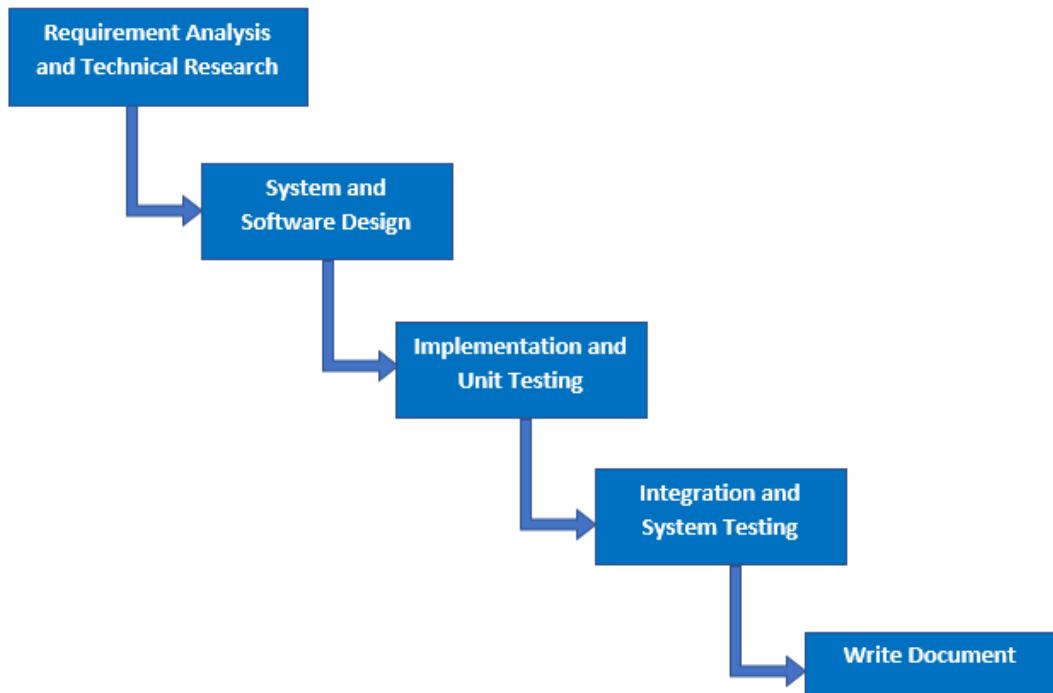


Figure 1: Software Process Model

2.2. Roles and responsibilities

No	Full name	Role in Group	Responsibilities
1	Phạm Công Thành	Project Manager	<ul style="list-style-type: none"> • Specify user requirement • Control the development process • Give out technique and business analysis support
2	Hoàng Ngô Minh Tùng	Team Leader, QA, DEV, Tester	<ul style="list-style-type: none"> • Manage the process, quality, risk • Design database • Clarify requirements • Prepare documents, reports • Arrange meeting • Design project architect • Coding • Testing
3	Nguyễn Bảo Khánh	Team Member, BA, DEV, Tester	<ul style="list-style-type: none"> • GUI design • Design database • Clarify requirements, analyze business • Review documents, reports • Design project architect • Coding • Testing
4	Lê Nguyễn Ngọc Sang	Team Member, QC, DEV, Tester	<ul style="list-style-type: none"> • Create test plan • GUI design • Design database • Clarify requirements, analyze business • Design project architect • Coding • Testing

2.3. Tools and Techniques

Tool	Name and version
IDE	<ul style="list-style-type: none"> - IntelliJ IDEA 2018.2.3 - Netbeans 8.0.2

	- Android Studio 3.1.3
Source control	SourceTree 2.6.10
Database Manager	MySQL Workbench 6.3.9 CE

Technique	Name and version
Front end	<ul style="list-style-type: none"> • HTML v5.0 • JQuery v3.2.1 • Bootstrap v3.3.7 • CSS • Javascript • Ajax • ReactJS
Back end	<ul style="list-style-type: none"> • Java EE 8 • Spring Boot v2.0.5 RELEASE • Hibernate v5.0.12 • Log4J v1.2.17 • JUnit v3.8.1
Google Map API libraries	google maps
Web server	Apache Tomcat v8.5.15
Database system	MySQL 5.7.20

3. Project Management Plan

3.1. Software development life cycle

Phase	Description	Deliverables	Resource needed	Dependencies and Constraints	Risks
Requirement analysis and technical research	- Identify and clarify requirements for system	<ul style="list-style-type: none"> - Software Requirement Specification - Basic 	4 weeks	N/A	- Lack of member share of understand

	<ul style="list-style-type: none"> - Research for google maps api and crawling technique (StAX) 	<p>knowledge Google Maps API</p> <ul style="list-style-type: none"> - Understand about StAX technique - Introduction document - Project Management Plan - Prototypes 			<ul style="list-style-type: none"> - SRS does not match user requirement - Misunderstanding of learning
System and software design	<ul style="list-style-type: none"> - Architect design for the system - Detail design using top-down break down 	<ul style="list-style-type: none"> - Software Design Description Document - Base code structure - Technology notes 	2 weeks	Depend on “Requirement Analysis and Technical Research”	<ul style="list-style-type: none"> - Lack of experience - Diagrams don’t match SRS - Not fulfil requirement
Implementation and unit testing	<ul style="list-style-type: none"> - Implement the system - Crawling data - Building mobile application - Prepare test plan and test cases - Unit testing of 	<ul style="list-style-type: none"> - System Implementation and Test Document - Programs (web and mobile app) - Unit test cases and results 	4 weeks	Depend on “System and Software Design”	<ul style="list-style-type: none"> - Delay in implementation - Inaccurate programming

	the code				
Integration and system testing	<ul style="list-style-type: none"> - Perform functional and non functional testing - Make sure the system meets the requirements 	<ul style="list-style-type: none"> - Test report - Defect report 	1 week	<p>Depend on “Implementation and Unit Testing”</p>	<ul style="list-style-type: none"> - Missing of functions that need to be tested
Write document	<ul style="list-style-type: none"> - Write “Software User’s Manual” - Review all documents - Write paper 	<ul style="list-style-type: none"> - Software User’s Manual - A paper 	3 weeks	<p>Depend on “System Design” and “Implementation”</p>	<ul style="list-style-type: none"> - Don’t have enough time to complete paper.

Table <number>: Software development life cycle

3.2. Phase Detail

3.2.1. Requirement Analysis and Technical Research

Task	Description	Author
Identify and clarify functions	<p>Identify and clarify user requirements</p> <p>Divide all functions into groups by concept</p>	TungHNM, KhanhNB, SangLNN
Research on Google Maps and Location	<p>Research on google maps api</p> <p>Research on HTML5 Location, Android Location</p> <p>Research on ReactJS, StAX crawling technique</p>	TungHNM, KhanhNB, SangLNN
Design application	Design GUI for web and mobile application base on main functions	TungHNM, KhanhNB, SangLNN

	Design Use Case Diagram	
Write document	Write Introduction document Write Project Management Plan Write Software Requirement Specification	TungHNM, KhanhNB, SangLNN

3.2.2. System and Software Design

Task	Description	Author
Design ERD	Design Entity Relationship Diagram for database	TungHNM, KhanhNB, SangLNN
Design architect	Design architect and main functions of application Design Class Diagram, Interaction Diagram, Component Diagram	TungHNM, KhanhNB, SangLNN
Write document	Write Software Design Description	TungHNM, KhanhNB, SangLNN

3.2.3. Implementation and Unit Testing

Task	Description	Author
Implement Web Service	Implement main functions Implement REST api	TungHNM, KhanhNB, SangLNN
Implement GUI	Implement Web and Mobile UI	TungHNM, KhanhNB, SangLNN
Implement Algorithms	Implement location suggestion	TungHNM, KhanhNB, SangLNN

	Implement pagination Implement data crawling	
Unit testing	Prepare test cases for all functions Test each functions and provide check list of defects Fix bugs base on the check list	TungHNM, KhanhNB, SangLNN
Write document	Write System Implementation and Test Document	TungHNM, KhanhNB, SangLNN

3.2.4. Integration and System Testing

Task	Description	Author
Prepare test plan	Prepare overview plan for testing	TungHNM, KhanhNB, SangLNN
Prepare test case	Prepare test cases for the whole system base on user requirement	TungHNM, KhanhNB, SangLNN
Integration and system test	Perform all test cases	TungHNM, KhanhNB, SangLNN

3.2.5. Write document

Task	Description	Author
Write document	Write Software User's Manual Document	TungHNM, KhanhNB, SangLNN
Review all documents	Review 6 reports	TungHNM, KhanhNB, SangLNN

Write paper	<p>Write abstract and introduction</p> <p>Write about Pagination and Location Suggestion algorithm</p> <p>Write comparison</p> <p>Write conclusion</p>	TungHNM, KhanhNB, SangLNN
--------------------	--	------------------------------

3.3. All Meeting Minutes

All meeting minutes could be found at [here](#).

4. Coding Convention

We use Java to develop the application. Here is our specified coding convention:

- Packaging Convention:
 - Divide application into layers: presentation layer, service layer, business layer, data access layer,...
- Code block Convention:
 - No line break before the opening brace.
 - Line break after the opening brace.
 - Line break before the closing brace.
 - Line break after the closing brace, only if that brace terminates a statement or terminates the body of a method, constructor, or named class.
- Naming Convention:
 - Package names are all lowercase, with consecutive words simply concatenated together (no underscores).
 - Class names are written in UpperCamelCase.
 - Constant names use CONSTANT_CASE: all uppercase letters, with words separated by underscores.
 - Use camel-case style for variables and functions name.
 - Names should reflect a business meaning or purpose.
- Commenting Convention:

- Place the comment on a separate line, not at the end of a line of code.
 - Begin comment text with an uppercase letter.
- Log data Convention:
 - Log fatal errors, exception to file.
 - Log URL on console before sending request.
- Documentation Convention:
 - Add java docs at the top of the class describing the purpose of the class and the responsibility of it.
 - Add a description at the top of each method describing the business flow of the method, the input/output and the success/failure scenarios it handles.
 - Add business and meaningful comments for complex algorithms or blocks written by each member.

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 logged into system. Authenticated user can do the following functions:

- Logout
- Edit profile

1.5. Scheduler Requirement

Scheduler is a kind of system role that runs 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

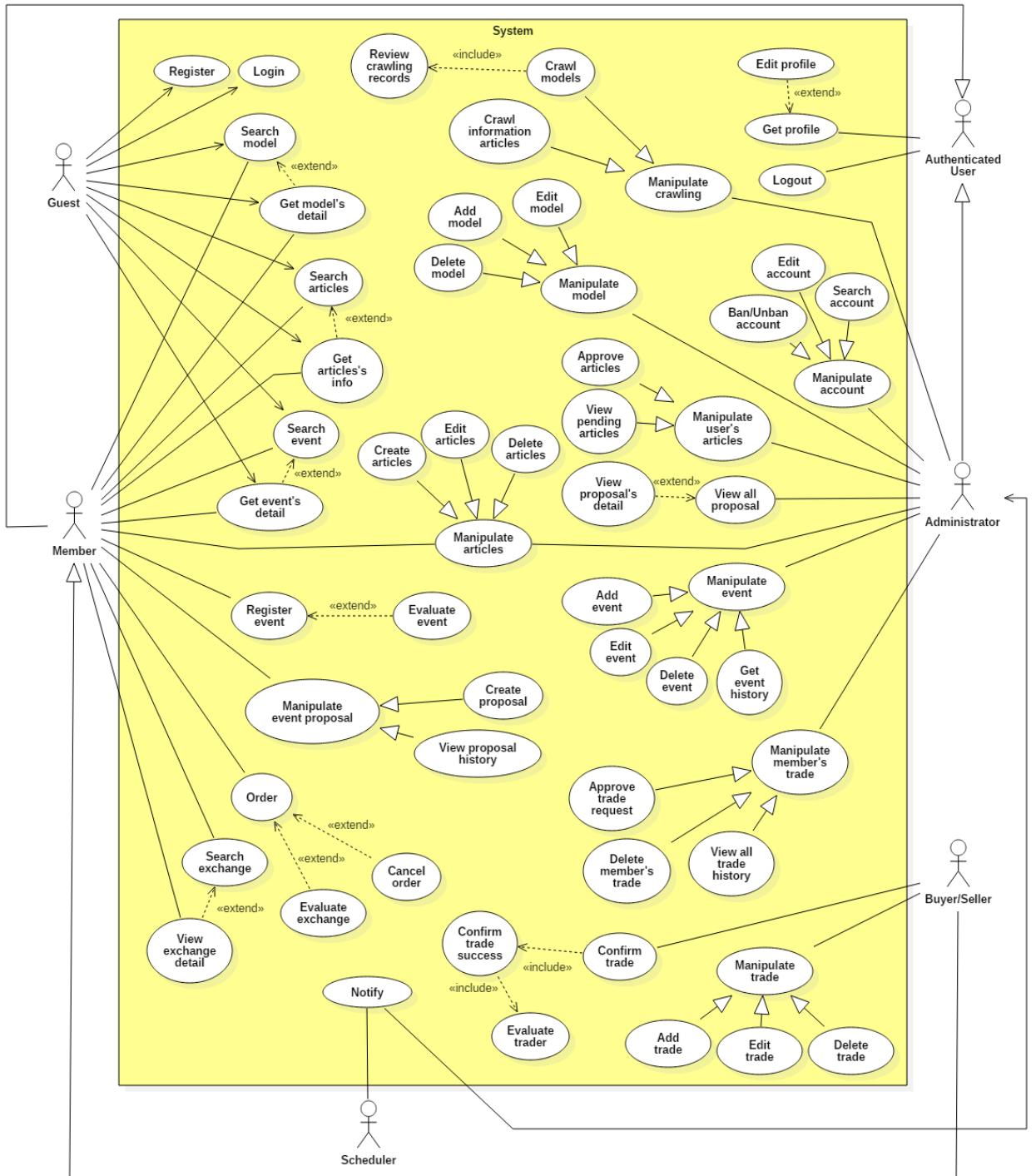
- Operating system: Window 10, Android 6 Marshmallow
- Environment: Java EE 8, Android SDK APIs level 15 or above
- IDE: IntelliJ IDEA 2018.2.3, Netbeans 8.0.2, Android Studio 3.1.3

- 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

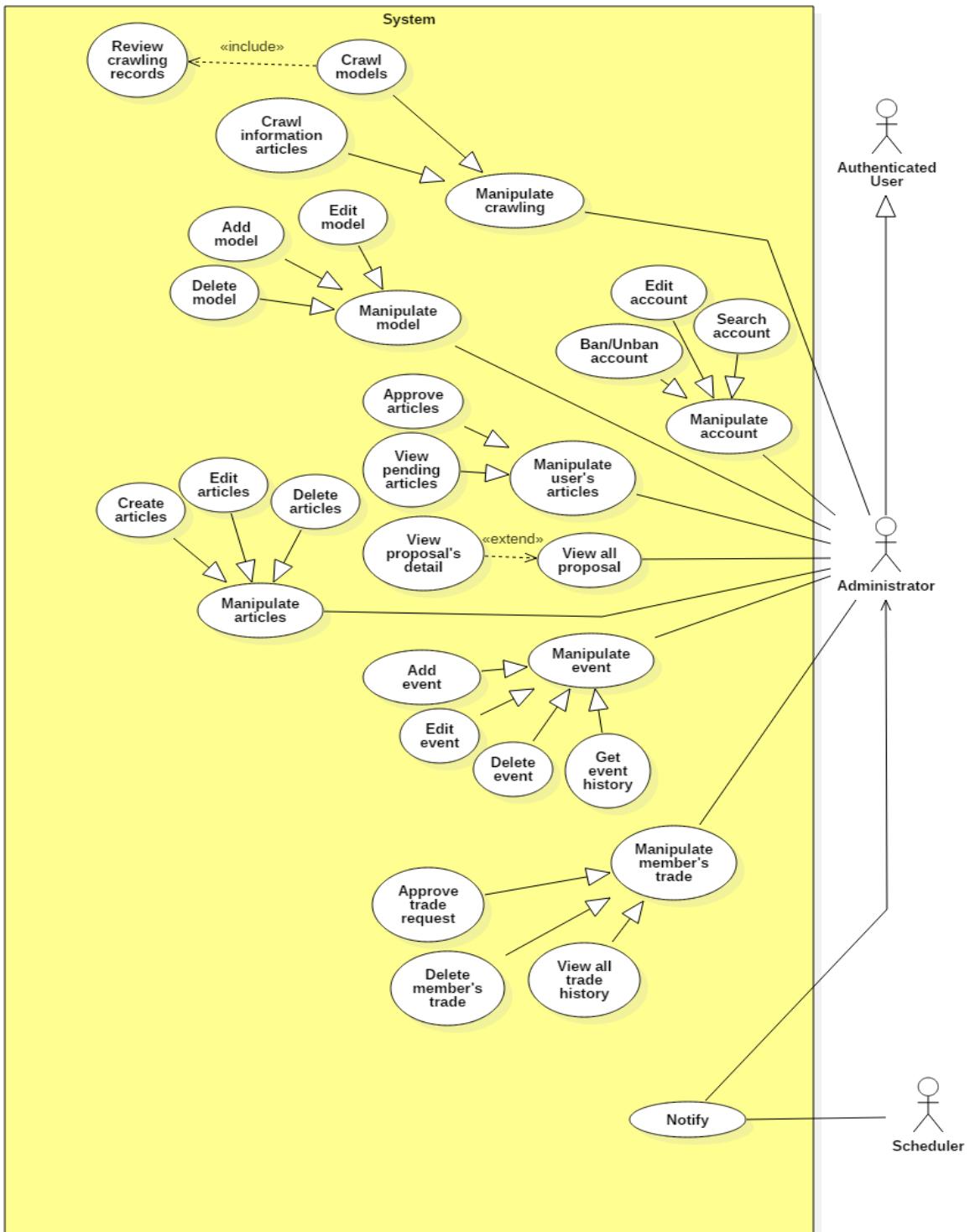
- Use HTTP protocol 1.1 for communication between:
 - Web application and web server
 - Mobile application and web server

2.2. System Overview Usecase

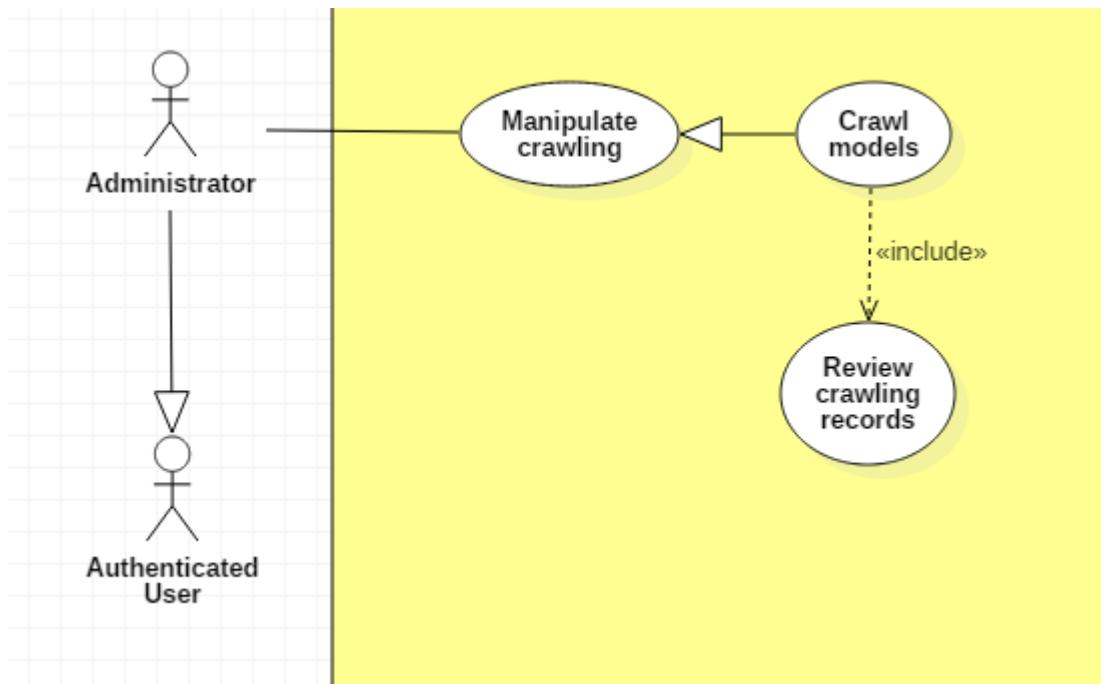


2.3. List of Usecase

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	<p>A list of models's crawling records is shown with following labels and fields:</p> <ul style="list-style-type: none">• 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	<p>Validate data.</p> <p>[Exception 1]</p> <p>Match new records with existed data in database.</p>

		<p>Save new models to database.</p> <p>System shows message gunpla models have been saved successfully.</p> <p>[Alternative 3]</p>
--	--	--

Alternative Scenario:

Alternative Scenario 1: Delete a record

Step	Actor Action	System Response
1	Administrator delete a record	<p>Delete the selected temporary record in database.</p>

Alternative Scenario 2: Click on a record

Step	Actor Action	System Response
1	Administrator clicks on a record	<p>System displays the selected record's detail page.</p>

Alternative Scenario 3: No new records

Step	Actor Action	System Response
1	Administrator clicks on “SAVE” button	<p>If there is no new records after matching with existed data in database, system shows</p>

		message there is no new records.
--	--	----------------------------------

Exceptions:

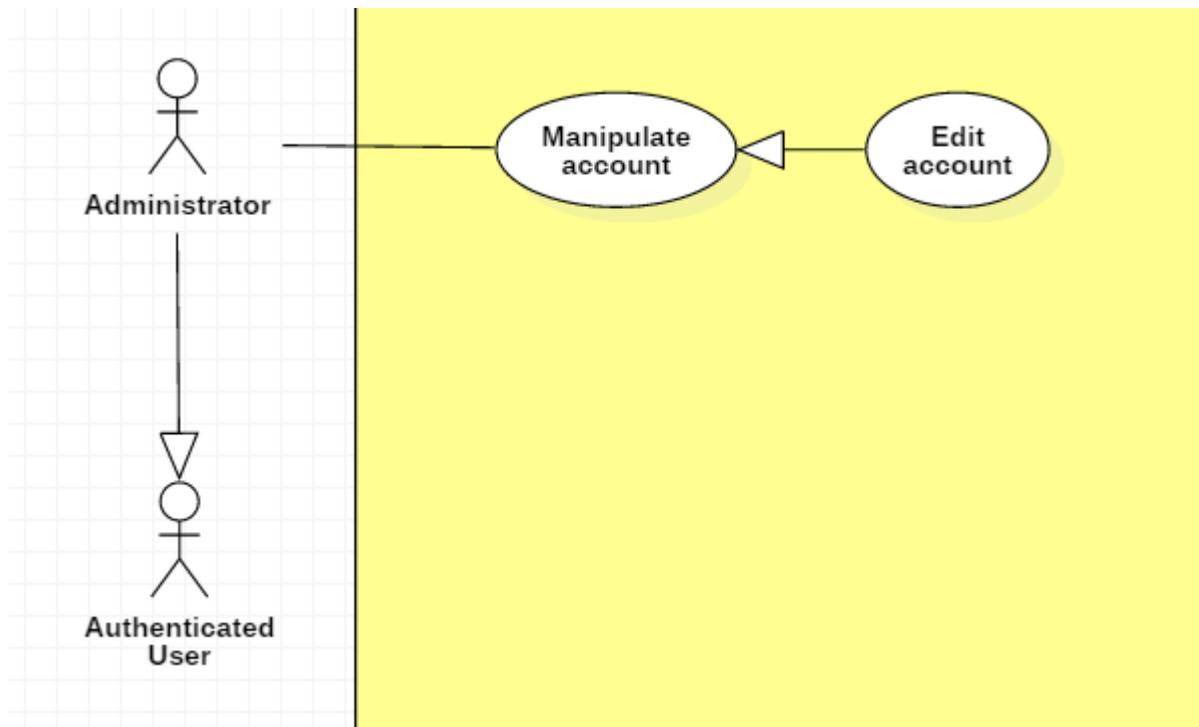
No	Cause	System Response
1	Administrator does not fill out all the required fields	System shows error message to ask administrator to enter missing required fields.
2	Length of field's value is out of range	System shows error message that notify administrator which field's value is out of range.

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	<p>Edit Account page is shown with following labels and fields:</p> <ul style="list-style-type: none"> • Email: text, required, max length: 50, RegEx: /^[a-z0-9_\.-]+@[\\da-zA-Z\.\-]+\.[a-zA-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 <p>System automatically fills out the form with information of the selected user's profile in database.</p>
2	Administrator fills out the form	
3	Administrator clicks on “SAVE” button	<p>Validate data.</p> <p>[Exception 1, 2, 3, 4, 5]</p> <p>Update account information to database</p> <p>System displays Profile page with updated</p>

		information.
--	--	--------------

Alternative Scenario: N/A

Exceptions:

No	Cause	System Response
1	Email does not match the required format: <code>/^(a-z0-9_\.[-]+)@([\da-z\.-]+)\.([a-zA-Z\.]{2,6})\$/</code>	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

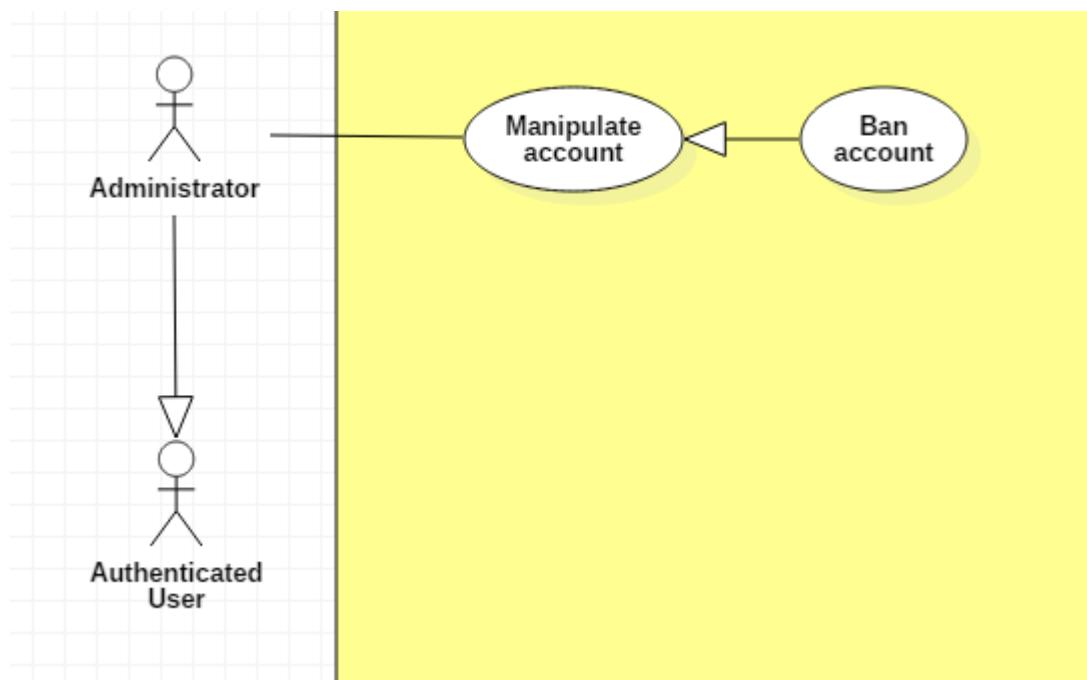
Business Rules:

- 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-zA-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.

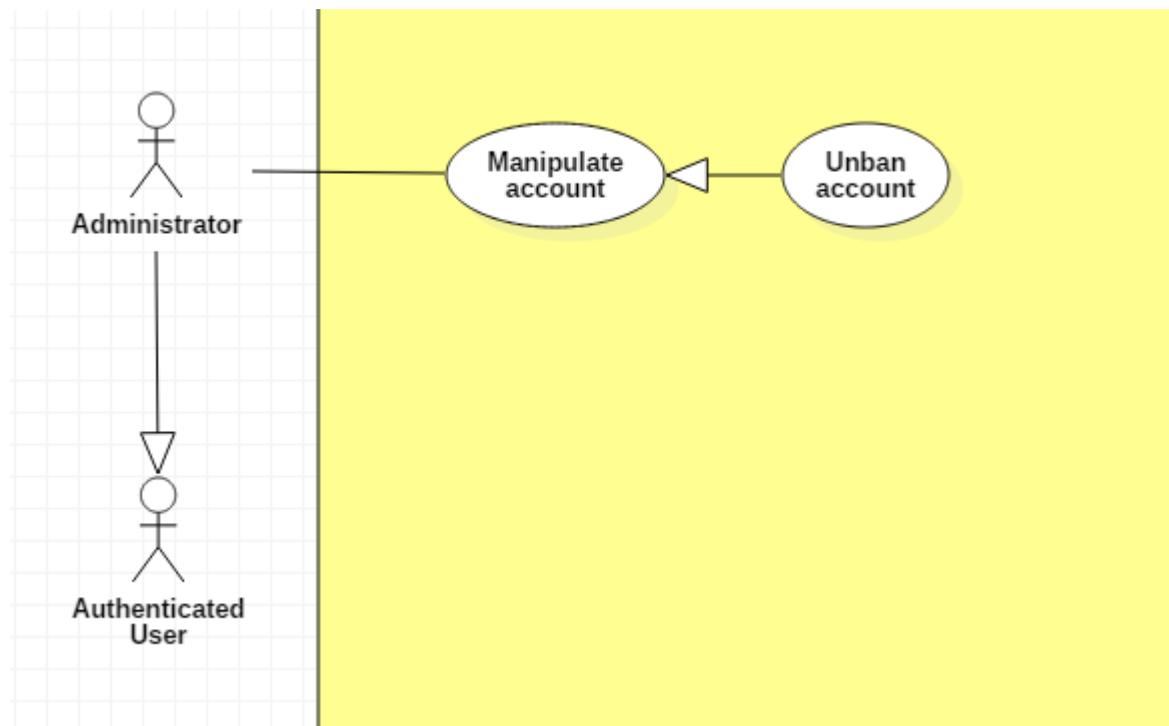
Exceptions: N/A

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
<p>Actor:</p> <ul style="list-style-type: none"> - Administrator. <p>Summary:</p> <ul style="list-style-type: none"> - This use case allows administrator to unban a member's account. <p>Goal:</p> <ul style="list-style-type: none"> - Allow a member to come back to system. <p>Triggers:</p> <ul style="list-style-type: none"> - Administrator sends unban account command to the system. <p>Preconditions:</p> <ul style="list-style-type: none"> - User must login into system with role Administrator. - The profile page of selected account is displayed. - The account's status is banned. <p>Post Conditions:</p> <ul style="list-style-type: none"> - Success: Member's account is unbanned and can login the system. - Fail: Show error message. <p>Main Success Scenario:</p>			
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.

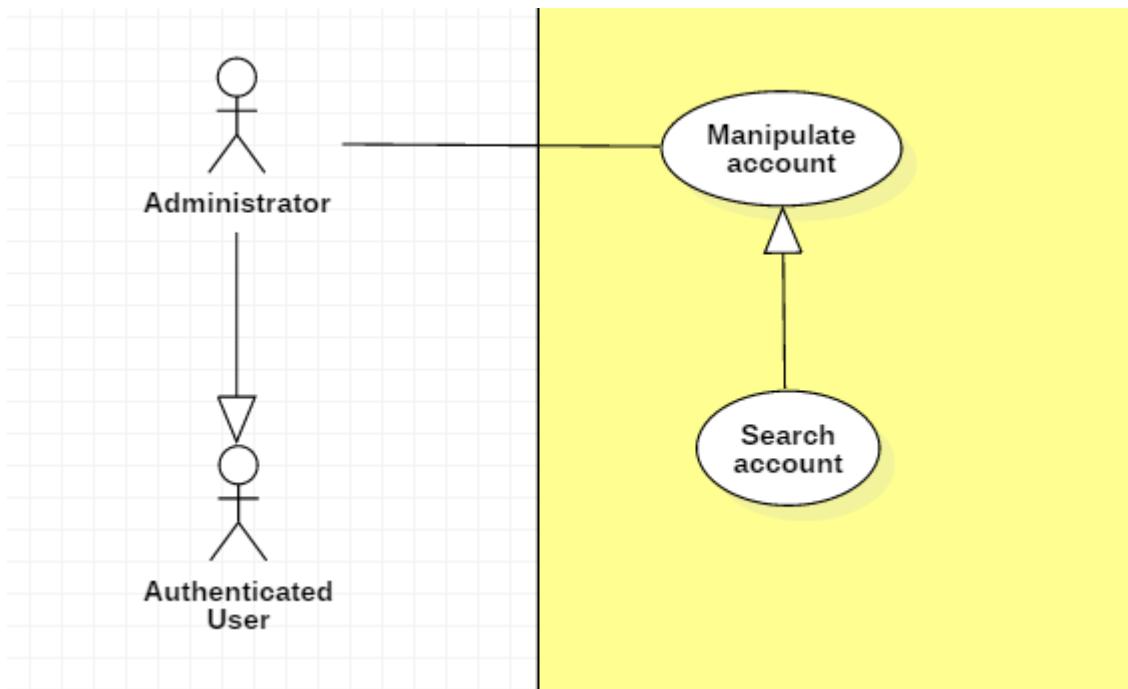
Exceptions: N/A

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.

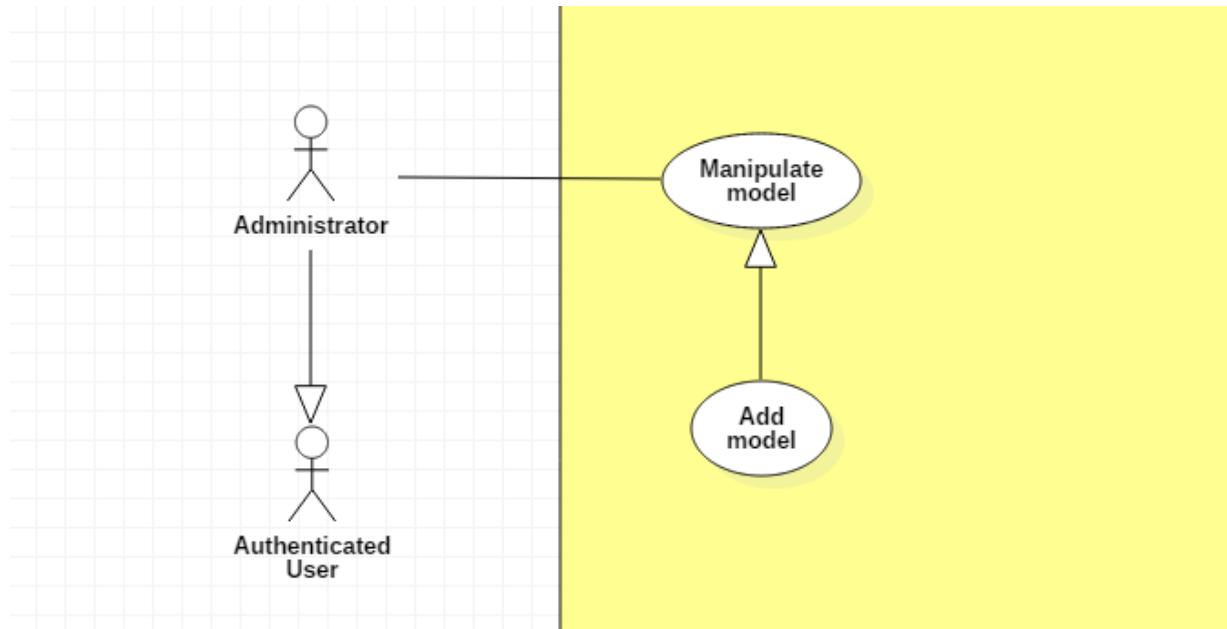
Exceptions: N/A

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	<p>An add new model form is shown with following labels and fields:</p> <ul style="list-style-type: none"> • Code: text, required with regEx: /^[A-z]+\$/ • Name: text, required, max length: 50 • Product Series: dropdown list • Series Title: dropdown list

		<ul style="list-style-type: none"> • 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	<p>Validate data</p> <p>[Exception 1, 2, 3, 5]</p> <p>New model is created and saved to database.</p> <p>System displays message Model {code} is saved successfully!</p>

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	<p>System displays 1 new row with the following labels and fields in a horizontal line:</p> <ul style="list-style-type: none"> • 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	<p>System removes the selected rows, includes:</p> <ul style="list-style-type: none"> • Import image: input type file • Image type: dropdown list • “+”: button

Exceptions:

No	Cause	System Response
----	-------	-----------------

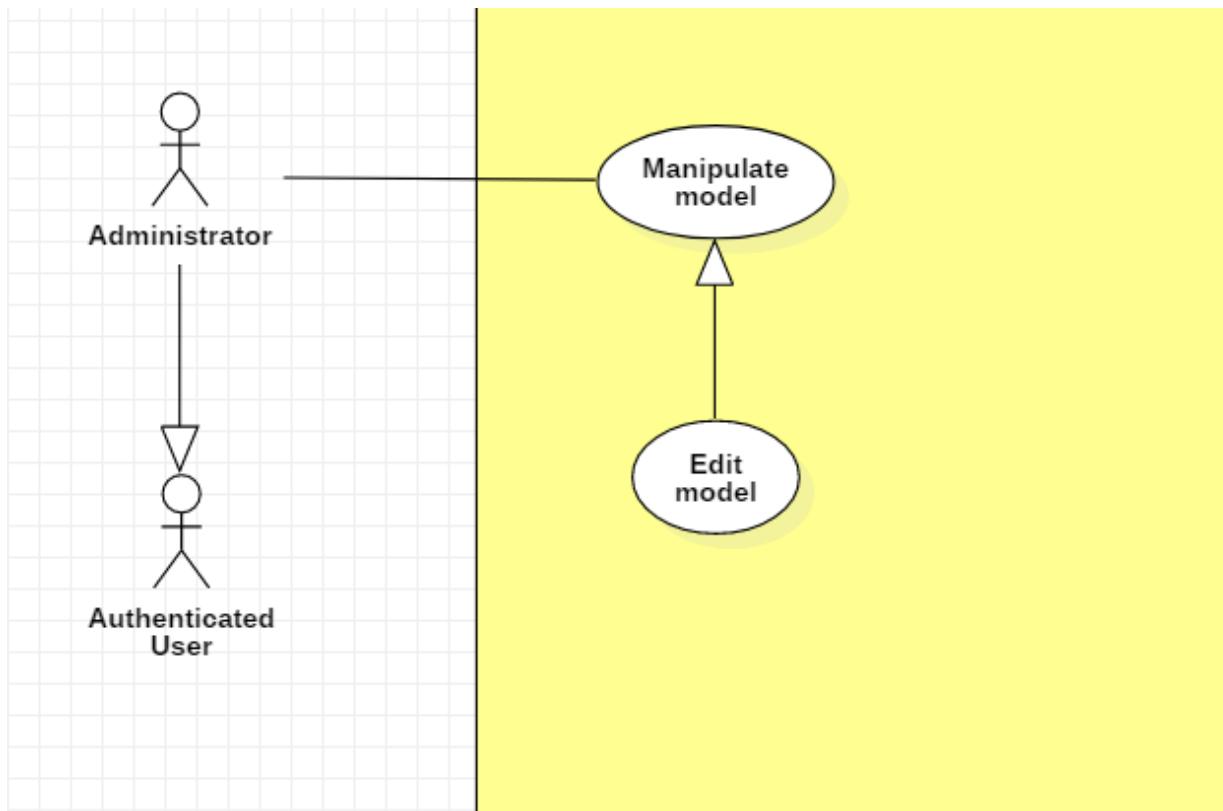
1	Administrator does not input all the required fields	System shows error message to ask administrator to fill all the required fields
2	Administrator inputs the invalid code.	System shows error message: Invalid format of code !
3	The model's code input from administrator is duplicate	System shows error 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	<p>System displays Edit Model page and a form with following labels and fields:</p> <ul style="list-style-type: none">• 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

		<ul style="list-style-type: none"> “+”: button <p>System automatically loads the selected gunpla model’s information to the form.</p>
2	Administrator fills out the form [Alternative 1, 2, 3]	
3	Administrator clicks on “SAVE” button	<p>Validate data</p> <p>[Exception 1, 2, 3, 5]</p> <p>Model’s information is updated and saved to database.</p> <p>System displays message: Model {code} is updated successfully!</p>

Alternative Scenario:

Alternative Scenario 1: Click on “Import image” button

Step	Actor Action	System Response
1	Administrator clicks on “Import image” button	<p>System shows dialog for administrator to choose local image</p>
2	Administrator chooses an image	<p>Validate file type</p> <p>[Exception 4]</p>

		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	<p>System displays 1 new row with the following lables and fields in a horizontal line:</p> <ul style="list-style-type: none"> ● 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	<p>System removes the selected rows, includes:</p> <ul style="list-style-type: none"> ● 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	

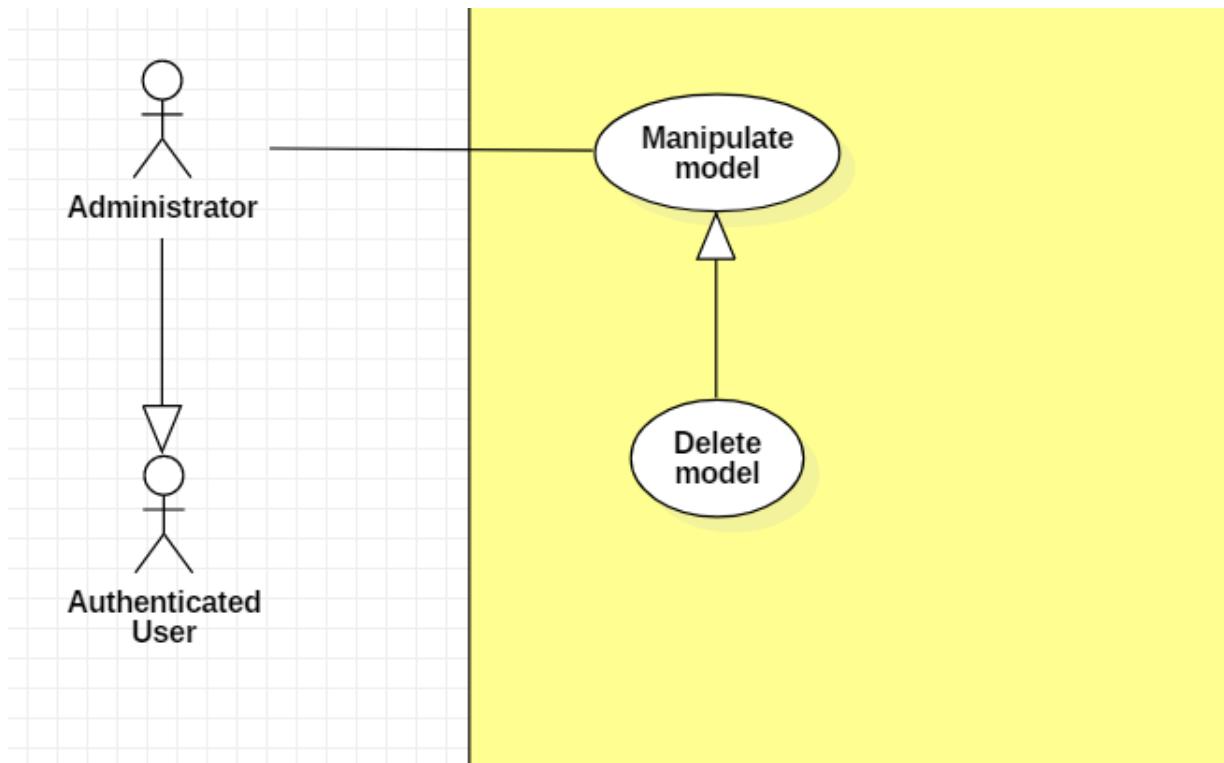
		System shows error message to ask administrator to fill all the required fields
2	Administrator inputs the invalid code.	System shows error message: Invalid format of code !
3	The model's code input from administrator is duplicate	System shows error message: This code is already existed !
4	File which 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	<p>System displays a dialog for administrator to confirm, the dialog is displayed with message: Are you sure ?, with 2 buttons:</p> <ul style="list-style-type: none">• Yes• No
2	Administrator clicks on “Yes” button [Alternative 1]	<p>Model information is deleted from database.</p> <p>System displays message: Model {code} is removed from system successfully !</p>

Alternative Scenario:**Alternative Scenario 1: Clicks on “No” button**

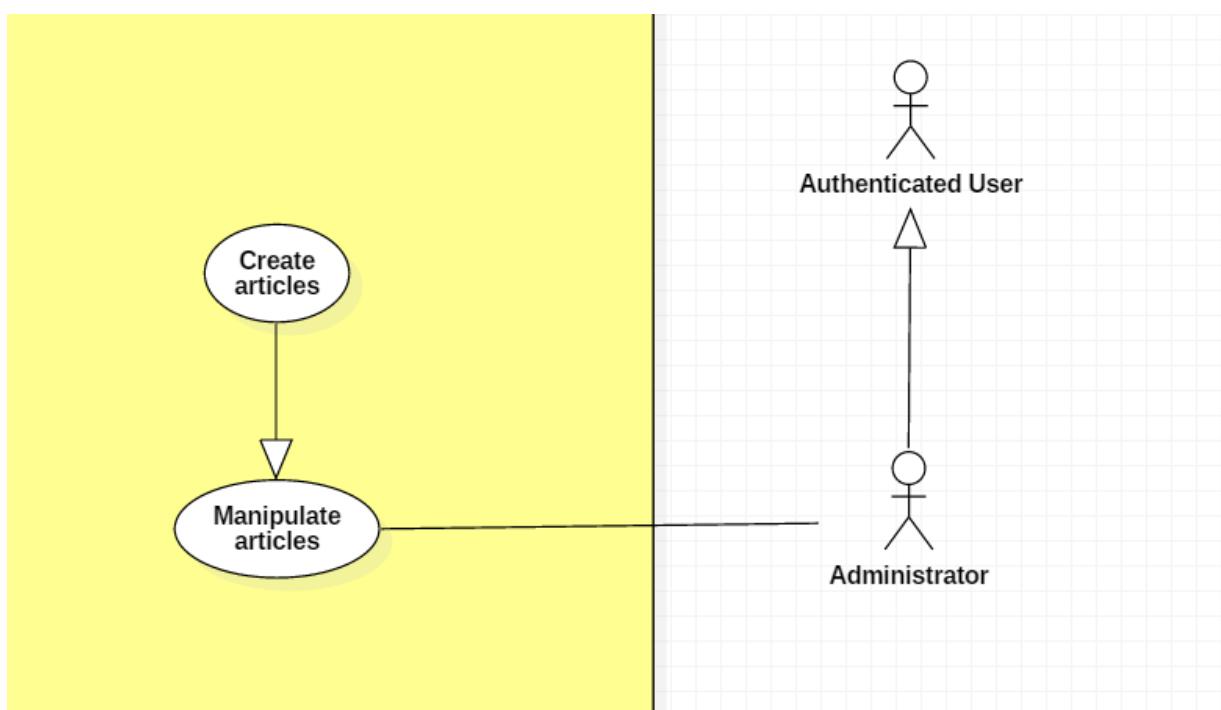
Step	Actor Action	System Response
1	Administrator clicks on “No” button	<p>The dialog disappears and system views detail page of the selected gunpla model, automatically fills out all fields.</p>

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: <ul style="list-style-type: none"> • 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.

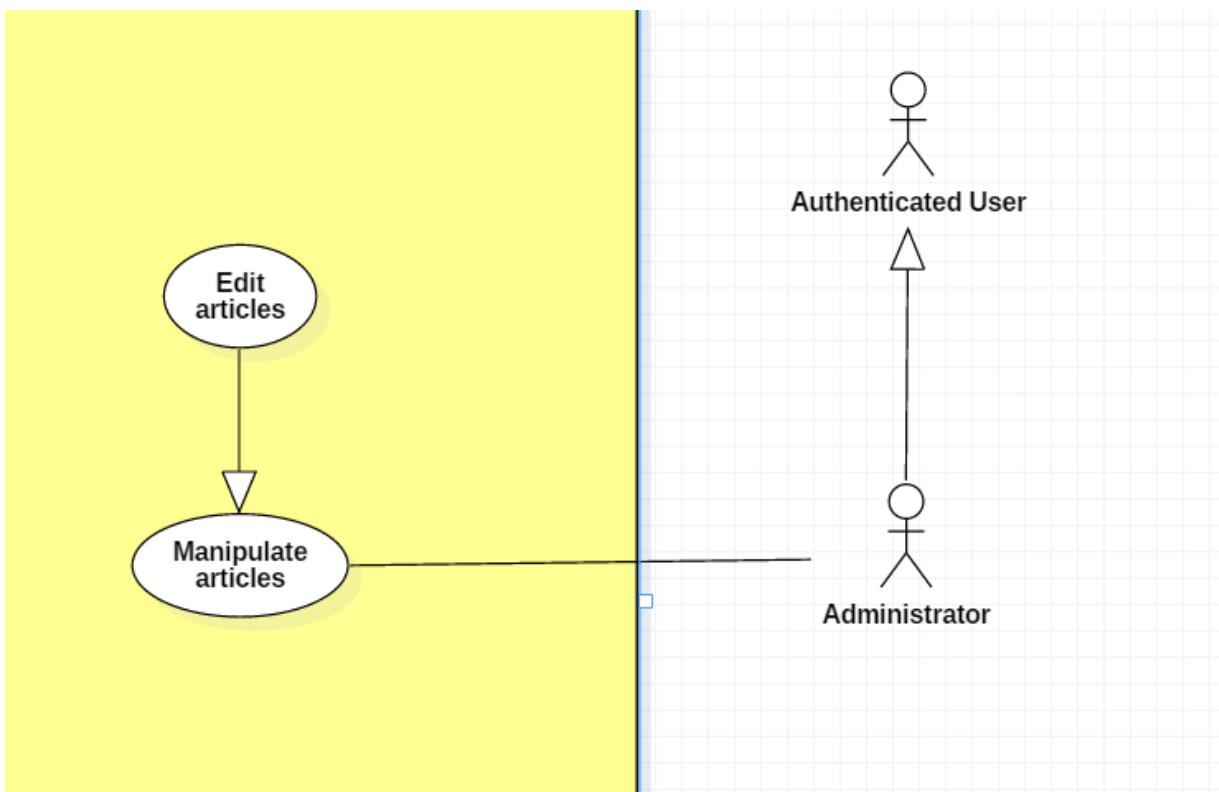
2	Admin upload file too large.	System shows error message to ask the admin to upload correct required file size.
---	------------------------------	---

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: <ul style="list-style-type: none">● 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:			

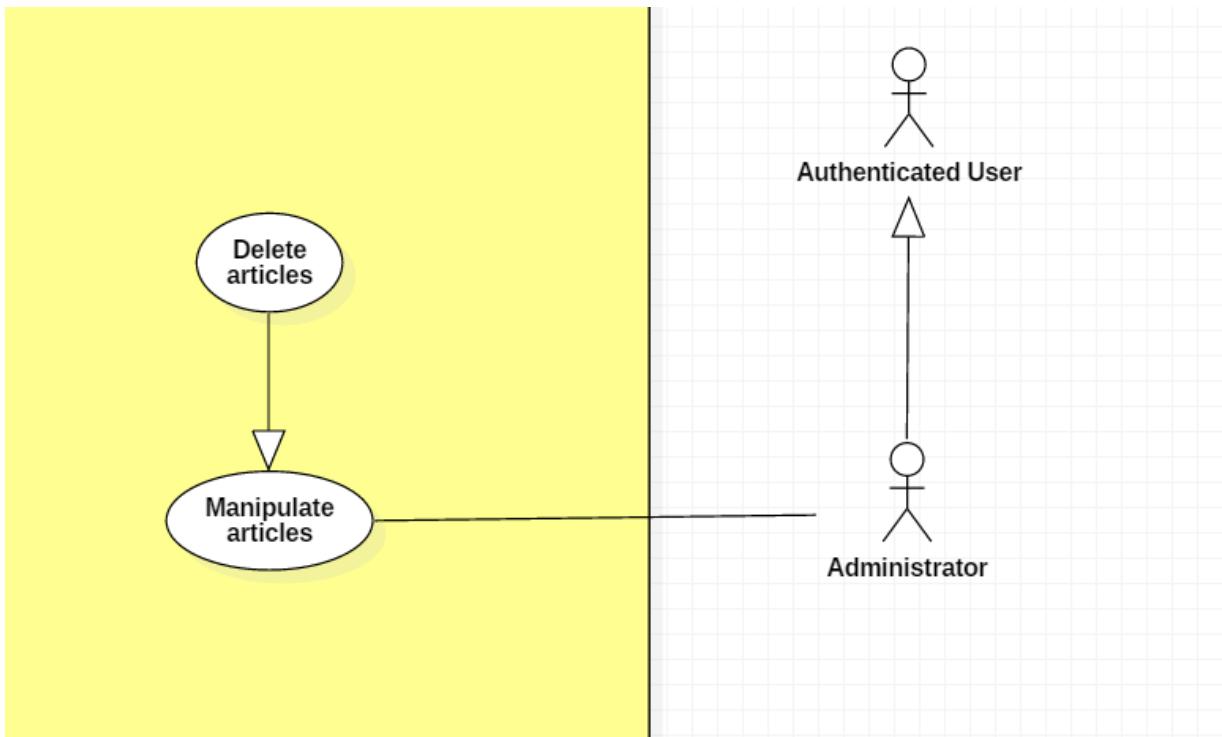
No	Cause	System Response
1	Admin submit article with empty content	System shows error message to ask the admin to enter missing required fields.
2	Admin upload file too large.	System shows error message to ask the admin to upload correct required file size.

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”	<p>System displays a dialog for administrator to confirm, the dialog is displayed with message: Are you sure ?, with 2 buttons:</p> <ul style="list-style-type: none"> ● Yes ● No
2	Admin selects “Yes”	<p>System deletes the article from database.</p> <p>Message saying “This article was deleted successfully”</p>

Alternative Scenario:

Step	Actor Action	System Response
2	Admin selects “No”	Dialog disappears, admin can view the article’s detail again.

Exceptions:

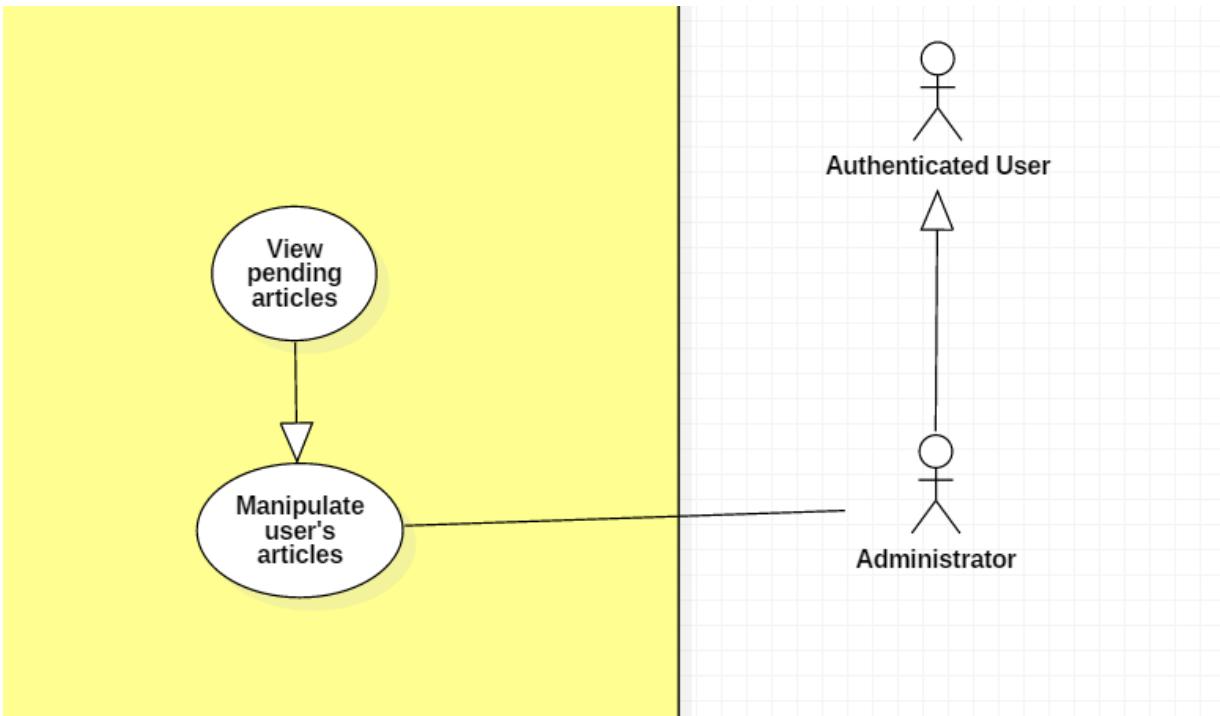
No	Cause	System Response
1	Article was already deleted	System shows error message “This article has been already deleted”

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.

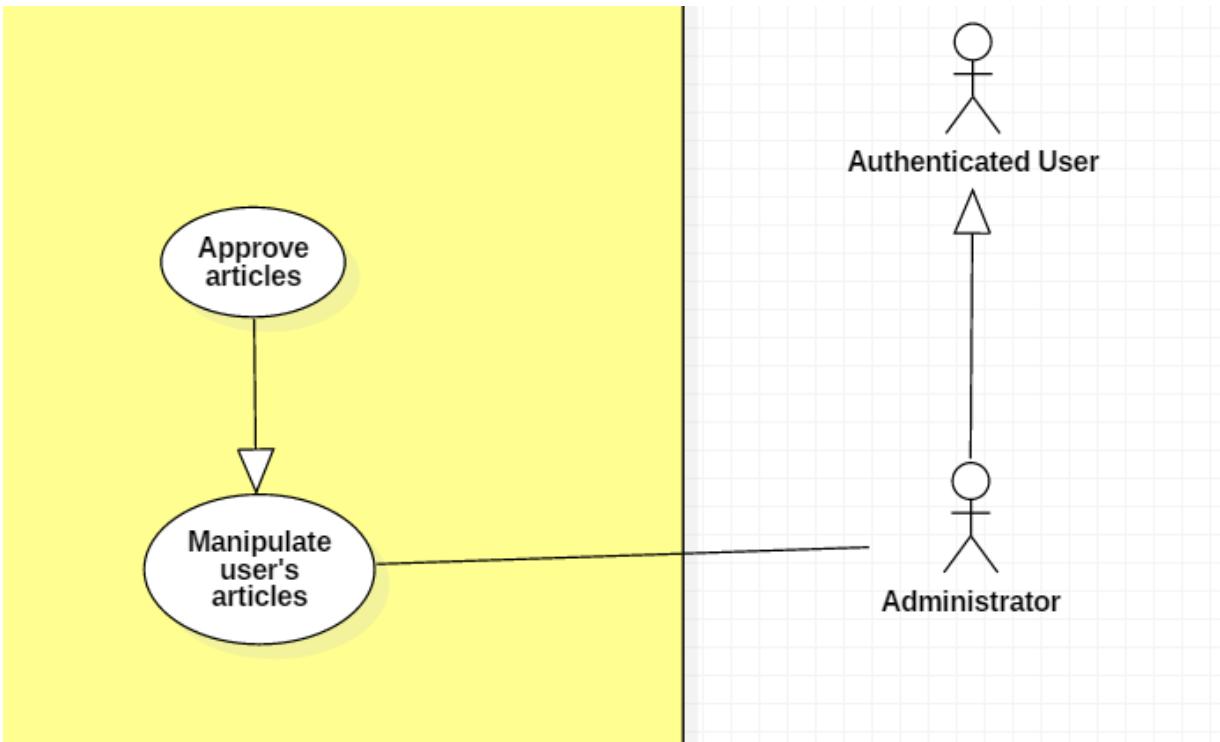
Main Success Scenario:

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 its 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:	<ul style="list-style-type: none"> - Administrator 		
Summary:	<ul style="list-style-type: none"> - This use case allows admin to approve a pending article 		
Goal:	<ul style="list-style-type: none"> - The article is posted on the website 		
Triggers:	<ul style="list-style-type: none"> - 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.

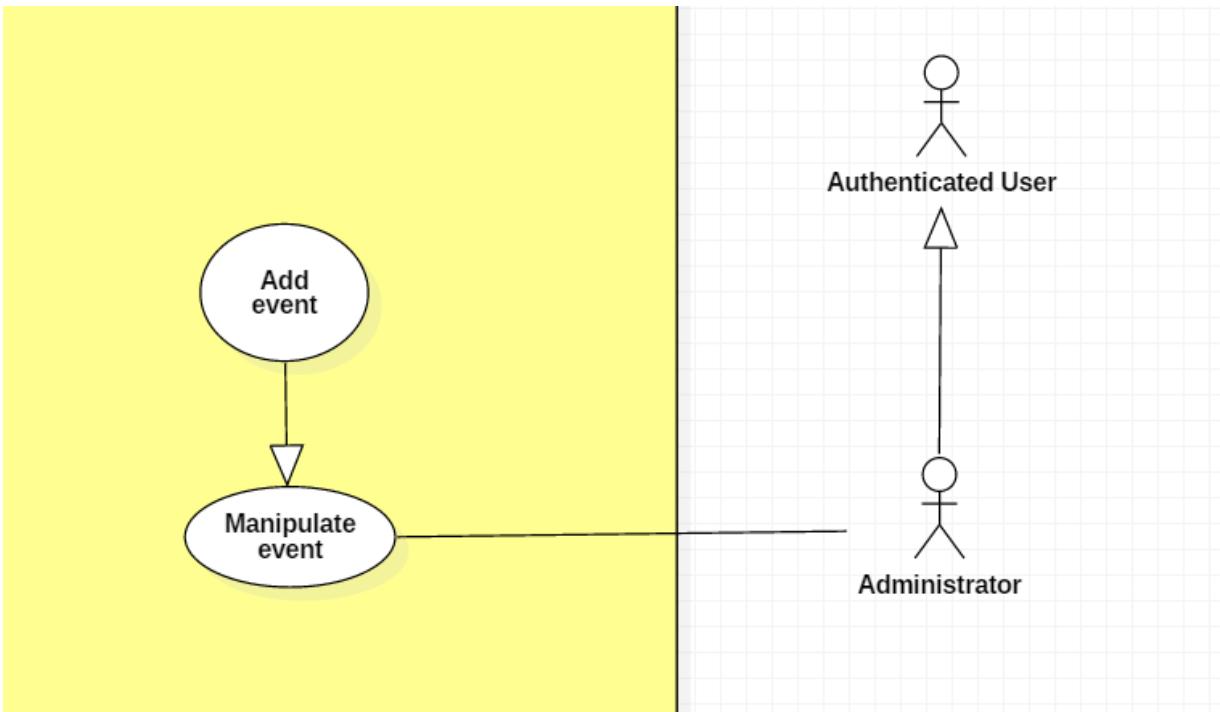
Main Success Scenario:

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	<p>“Create new event” view is shown with following labels and fields:</p> <ul style="list-style-type: none"> • 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:

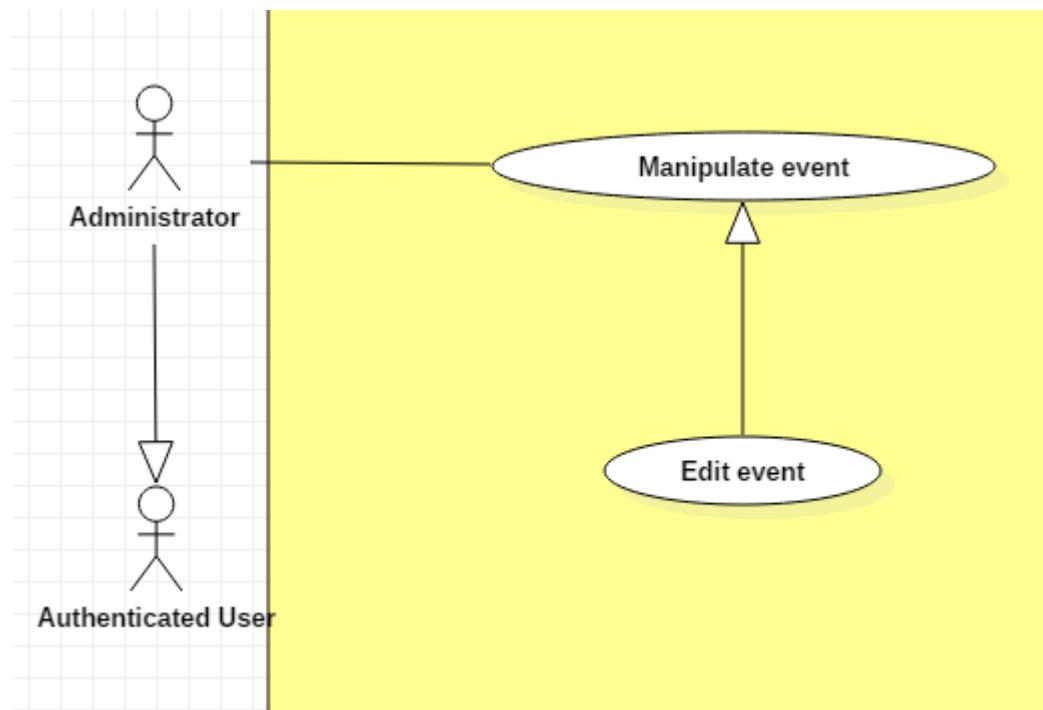
No	Cause	System Response

1	Admin submit with empty content	System shows error message to ask the admin to enter 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: N/A

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	<p>“Edit event” view is shown with following labels and fields with the old data:</p> <ul style="list-style-type: none"> • 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:

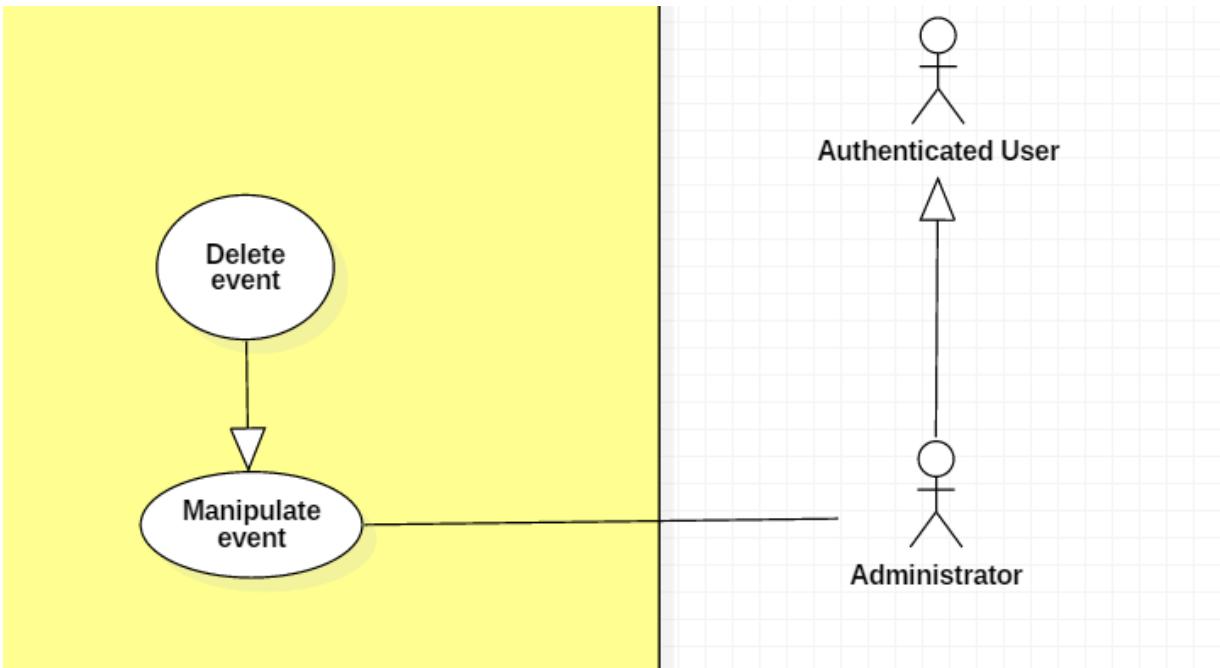
No	Cause	System Response
1	Admin submit with empty content	System shows error message to ask the admin to enter missing required fields.
2	Admin upload file too large.	System shows error message to ask the admin to upload correct required file size.

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: <ul style="list-style-type: none"> • 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.

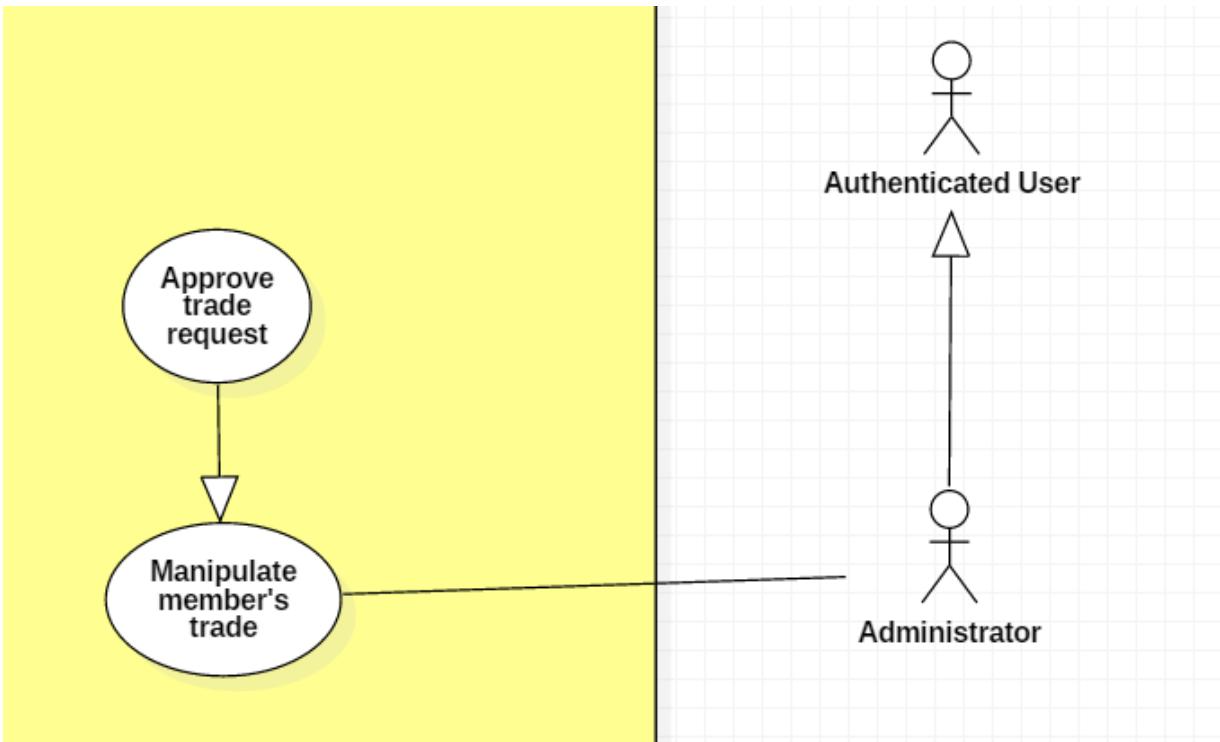
Exceptions:

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:

		<ul style="list-style-type: none"> • 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.	<p>Validate form.</p> <p>[Exception 1]</p> <p>System show message the trade post has been rejected and sends reason message to member who posted this trade.</p>

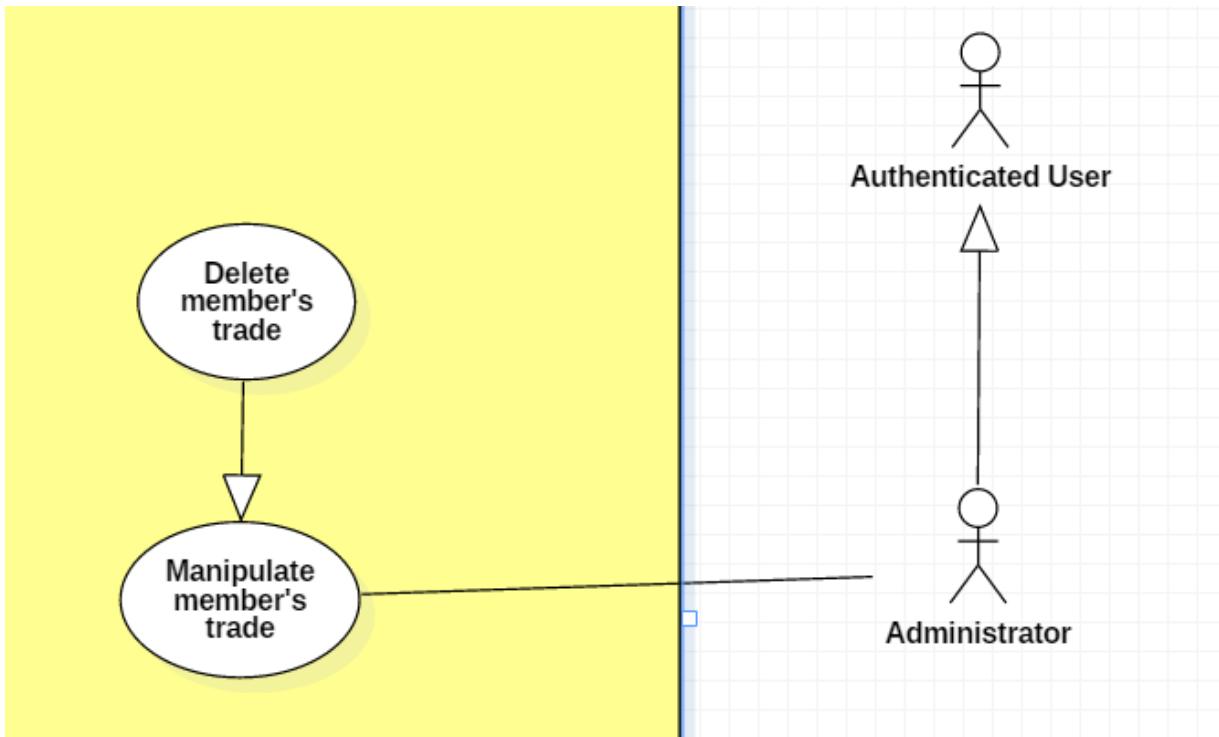
Exceptions:

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: <ul style="list-style-type: none"> - Administrators. 			
Summary: <ul style="list-style-type: none"> - This use case allows administrator to delete a trade post. 			
Goal: <ul style="list-style-type: none"> - 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.	<p>System create view show a delete form with following labels and fields:</p> <ul style="list-style-type: none">• 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.	<p>Validate form.</p> <p>[Exception 1]</p> <p>System show message deleted successfully.</p>

Alternative Scenario: N/A

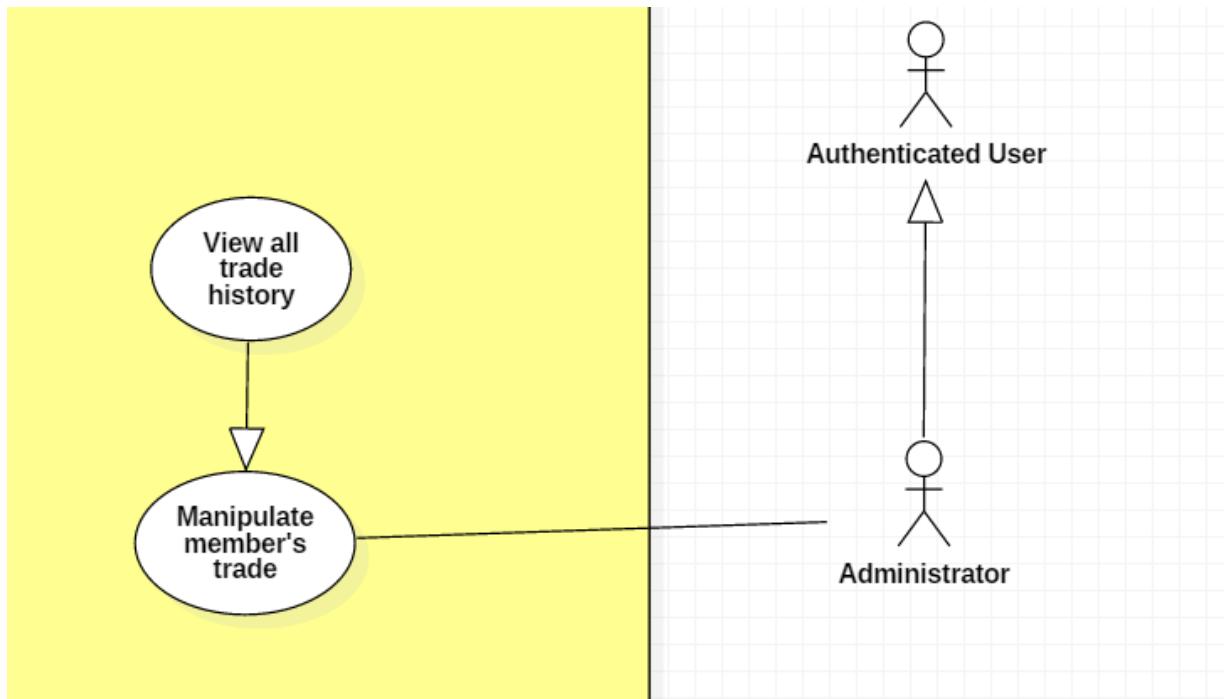
Exceptions:

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
-------------	------	-----------------	-----

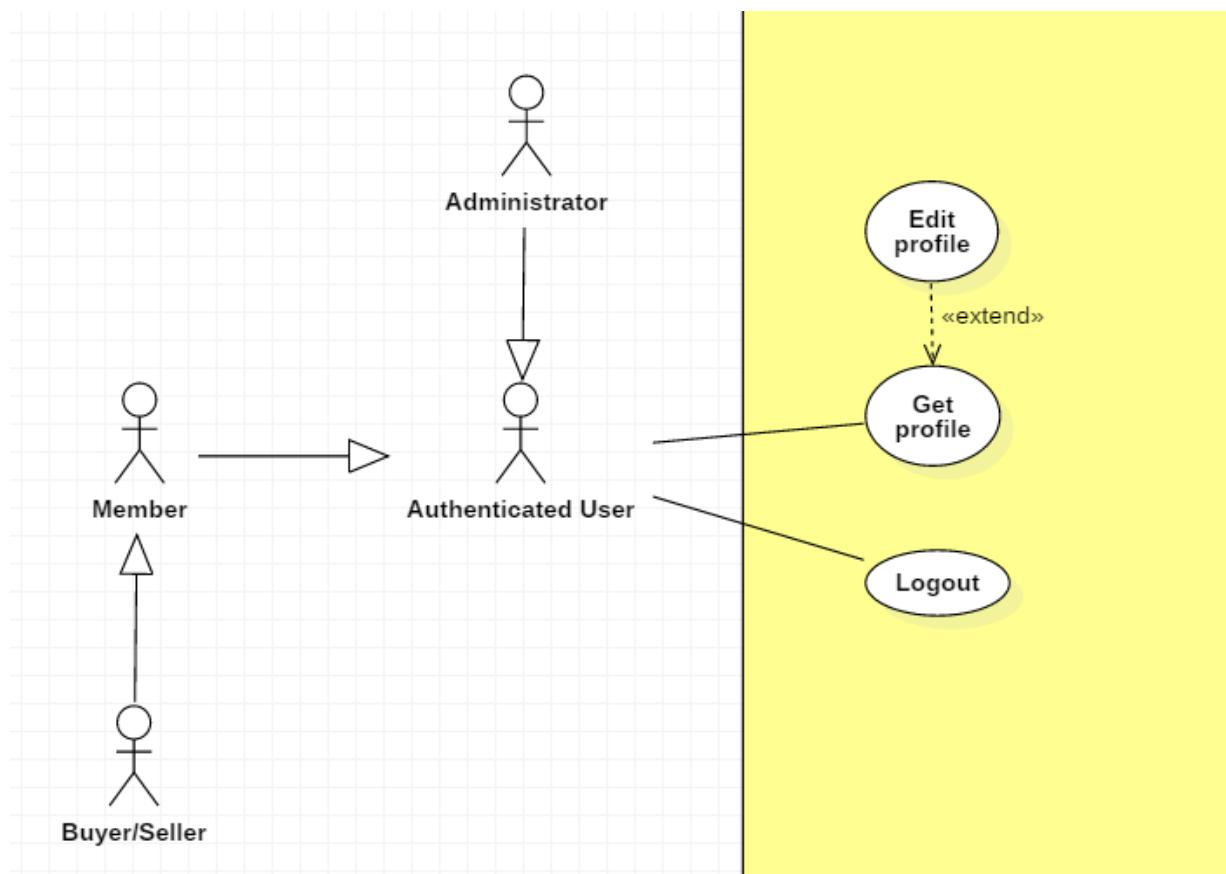
Usecase Name	View trade history		
Author	SangLNN		
Date	26/09/2018	Priority	Normal
Actor:			
<ul style="list-style-type: none"> - Administrator. 			
Summary:			
<ul style="list-style-type: none"> - This use case allows administrator view trade history. 			
Goal:			
<ul style="list-style-type: none"> - Administrator will see all order history. 			
Triggers:			
<ul style="list-style-type: none"> - Administrator sends view trade history command to the system. 			
Preconditions:			
<ul style="list-style-type: none"> - User must login into system with role Administrator. 			
Post Conditions:			
<ul style="list-style-type: none"> - 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

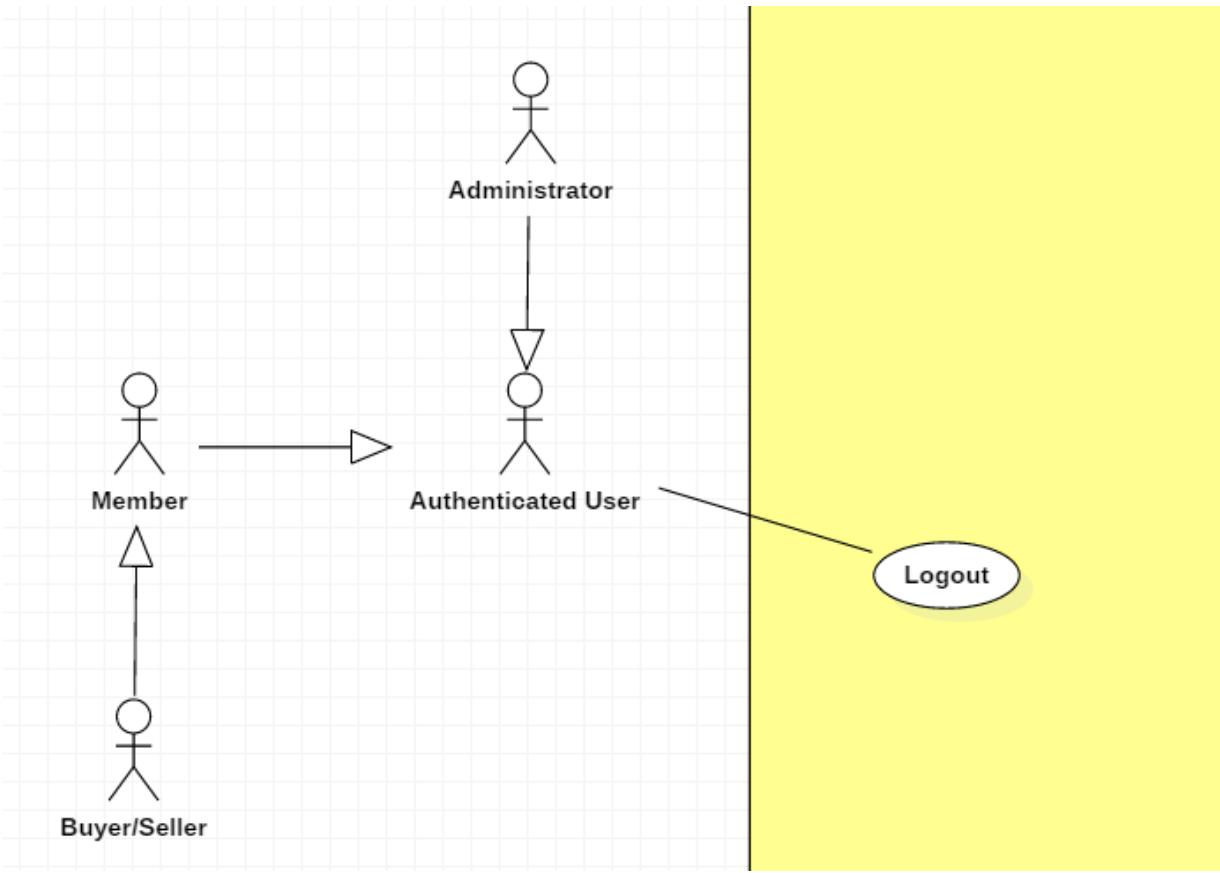
Relationships: N/A

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.

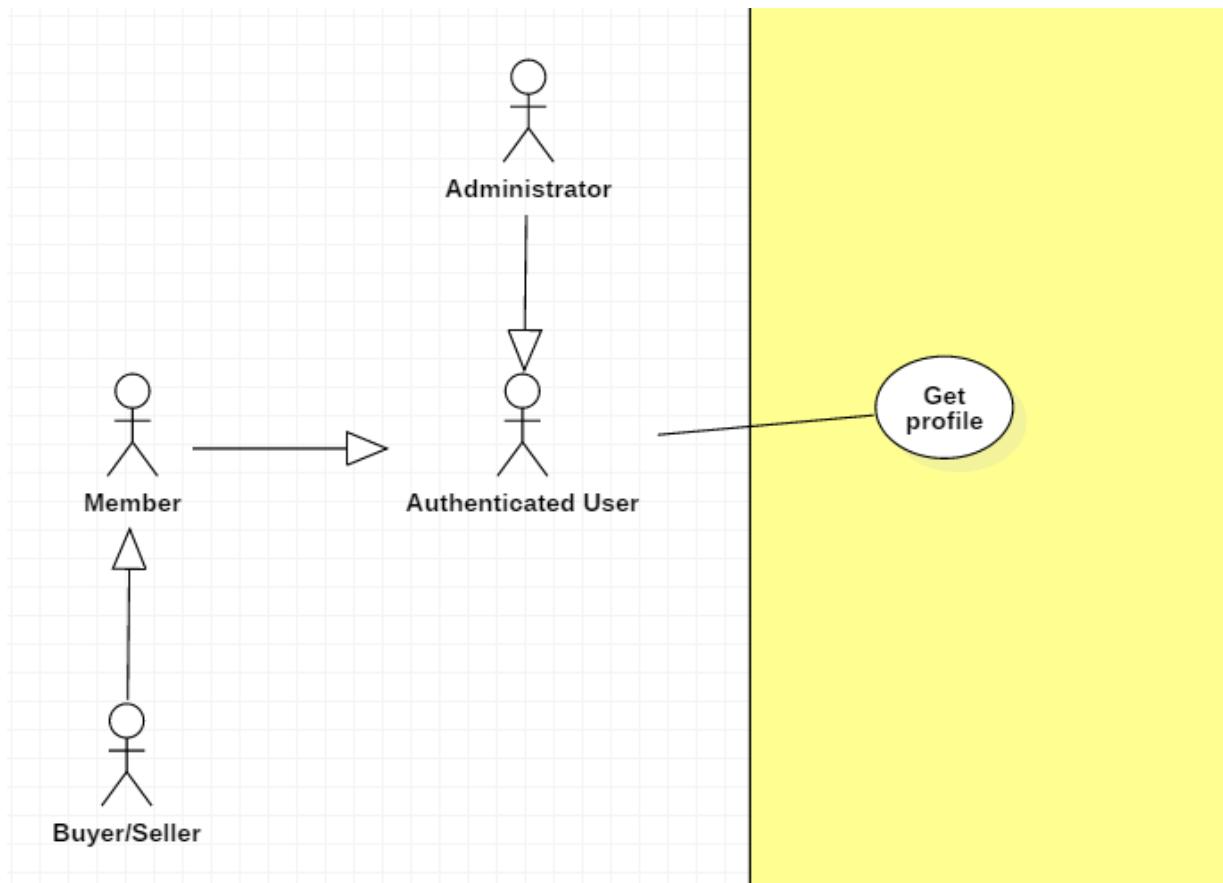
Main Success Scenario:

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.

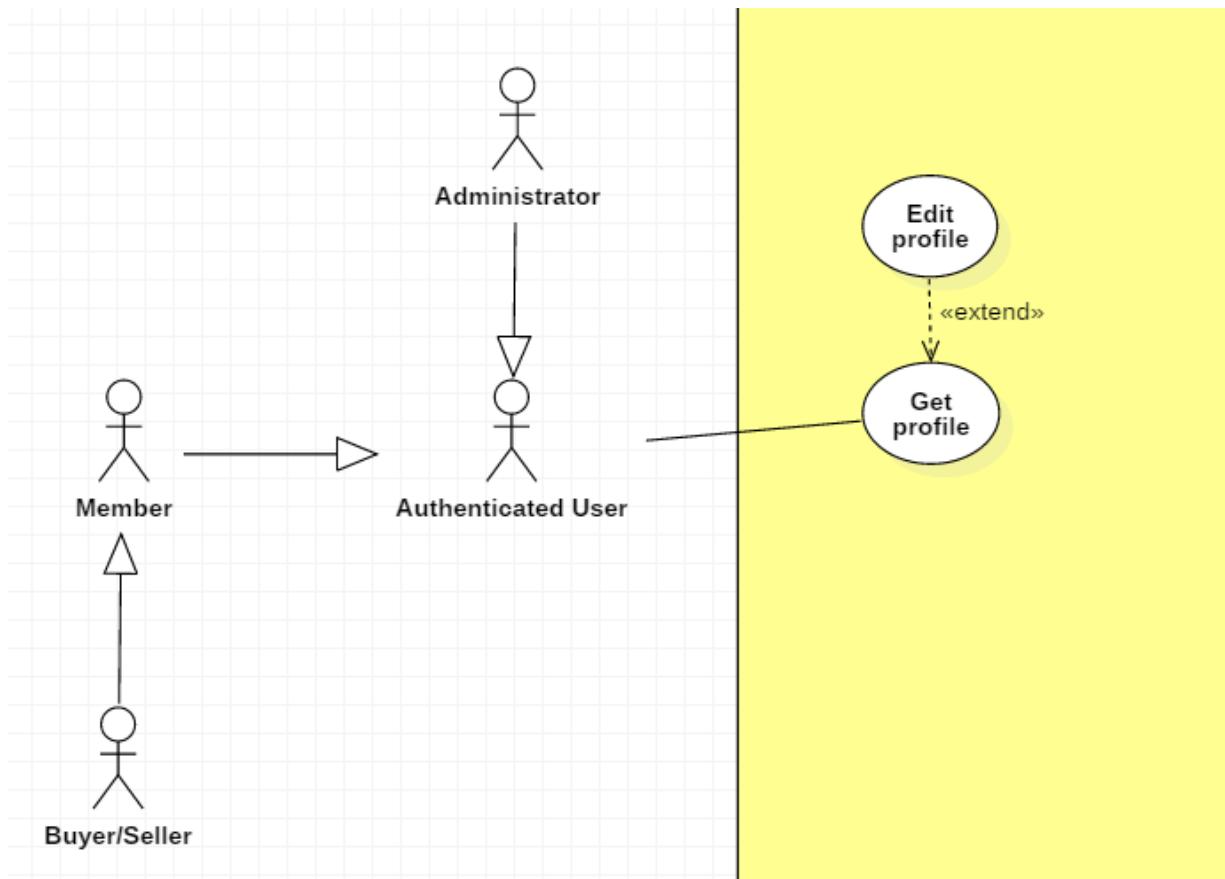
Post Conditions:

- **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	<p>System displays the selected user's profile page with the following labels and text:</p> <ul style="list-style-type: none">• 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:			
<ul style="list-style-type: none"> - Authenticated User. 			
Summary:			
<ul style="list-style-type: none"> - This use case allows authenticated user to update his/her profile information. 			
Goal:			
<ul style="list-style-type: none"> - 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	<p>Edit Profile page is shown with following labels and fields:</p> <ul style="list-style-type: none">• Email: text, required, max length: 50, RegEx: /^[a-z0-9_.-]+@[da-z.-]+\.[a-zA-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
2	User fills out the form	System automatically fills out the form with information of the selected user's profile in database .

3	User clicks on “SAVE” button	<p>Validate data.</p> <p>[Exception 1, 2, 3, 4, 5]</p> <p>Update account information to database</p> <p>System displays Profile page with updated information.</p>
---	------------------------------	--

Alternative Scenario: N/A

Exceptions:

No	Cause	System Response
1	Email does not match the required format: /^[a-z0-9_\.]-]+@[\\da-zA-Z\.-]+\.(a-zAZ\.)\{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

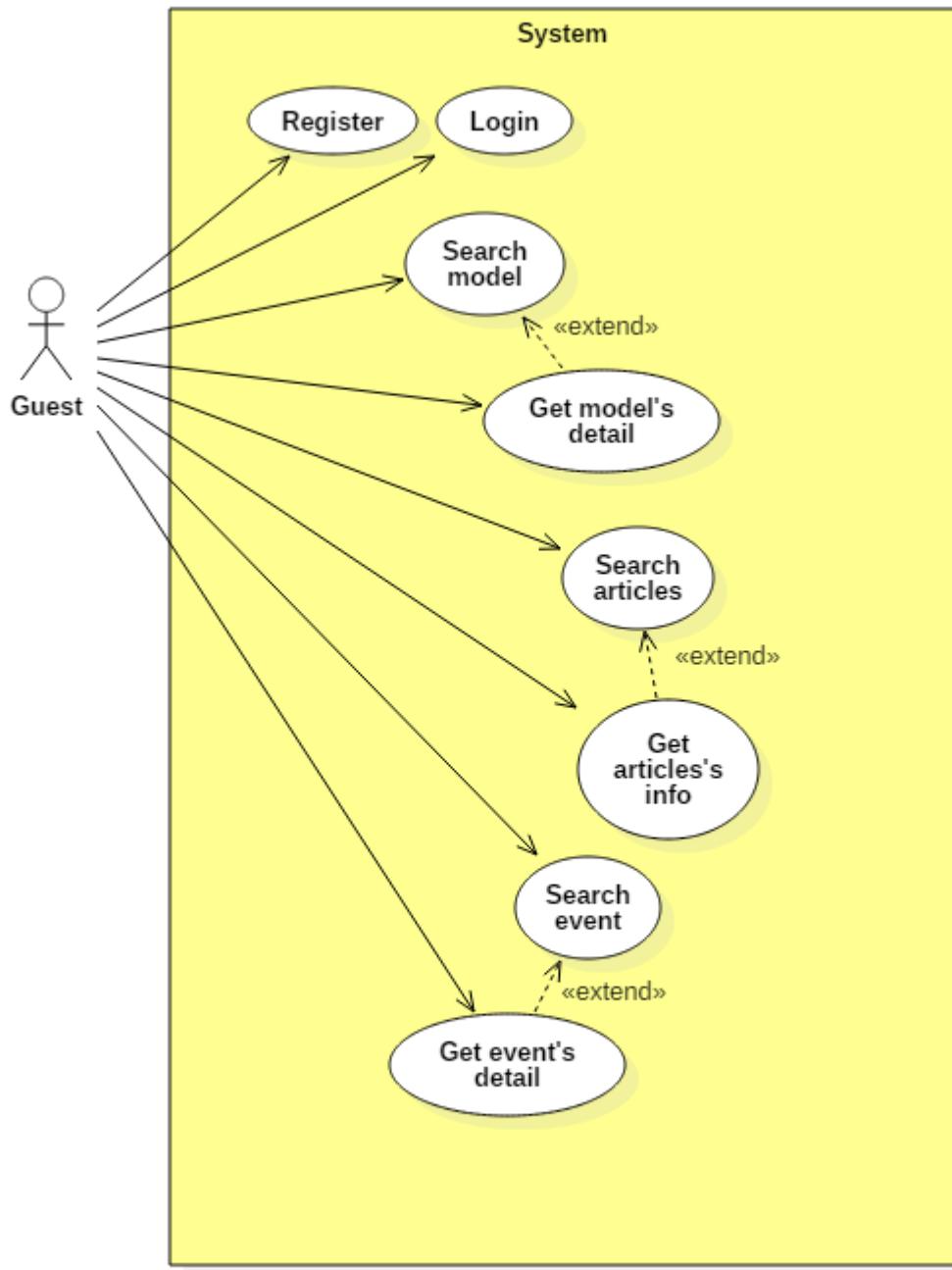
5	Length of field's value is out of range	System shows message that notify user which field's value is out of range
---	---	---

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	<p>Register page is shown with following labels and fields:</p> <ul style="list-style-type: none"> • Username: text, required, max length: 50 • Email: text, required, max length: 50, RegEx: /^[a-z0-9_.-]+@[\\da-z\\.-]+\\.(\\da\\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	<p>Validate data. [Exception 1, 2, 3, 4, 5]</p> <p>New account is added to system.</p> <p>System displays message: Register successfully !</p>

Alternative Scenario: N/A

Exceptions:

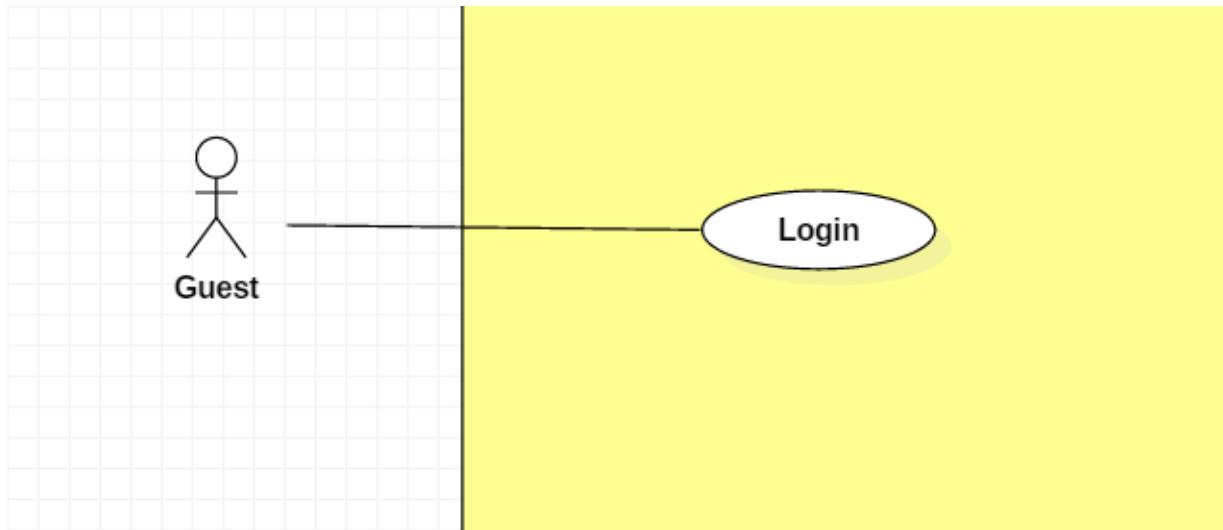
No	Cause	System Response
1	Entered email does not match the required format: /^[a-z0-9_\.]-+)([\da-z\.-]+)\.([a-zA-Z\.-]{2,6})\$/	System shows error message: Invalid email !
2	Entered email is duplicate in database	System shows error message: The email has been existed !
3	Missing of required fields	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:
 $/^([a-z0-9_\.]-+)([\da-z\.-]+)\.([a-zA-Z\.-]{2,6})$/$
- 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 logsins into system successfully.
- **Fail:** Show error message.

Main Success Scenario:

Step	Actor Action	System Response
1	Guest clicks on “Login” button	<p>Login page is shown with following labels and fields:</p> <ul style="list-style-type: none"> • Username: text, required, max length: 50 • Password: text, required, max length: 50
2	Guest fills out the form	
3	Guest clicks on “Login” button	<p>Validate data.</p> <p>[Exception 1, 2, 3, 4, 5, 6]</p> <p>New account is added to system.</p> <p>System displays message: Register successfully !</p>

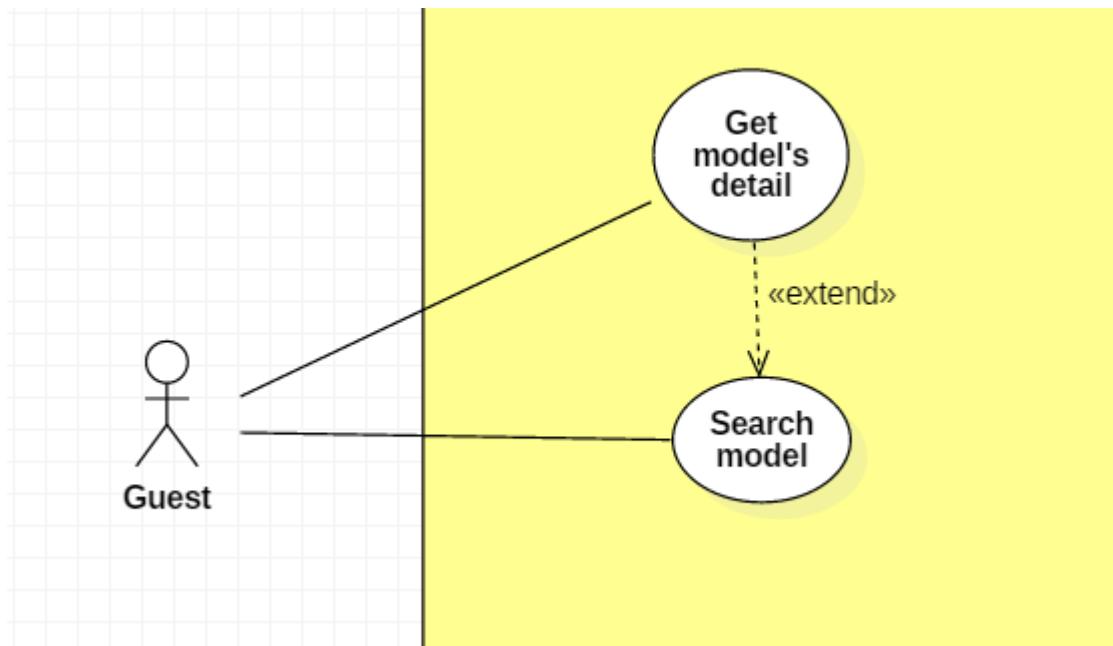
Alternative Scenario: N/A

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:		
<ul style="list-style-type: none"> - After login to system, guest will be redirected to specific page based on their role on the system: <ul style="list-style-type: none"> • 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	<p>System displays search model view with the following elements:</p> <ul style="list-style-type: none"> ● Search navbar: free text input ● SEARCH BY: dropdown list, includes values: <ul style="list-style-type: none"> ○ 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.
--	--	----------------

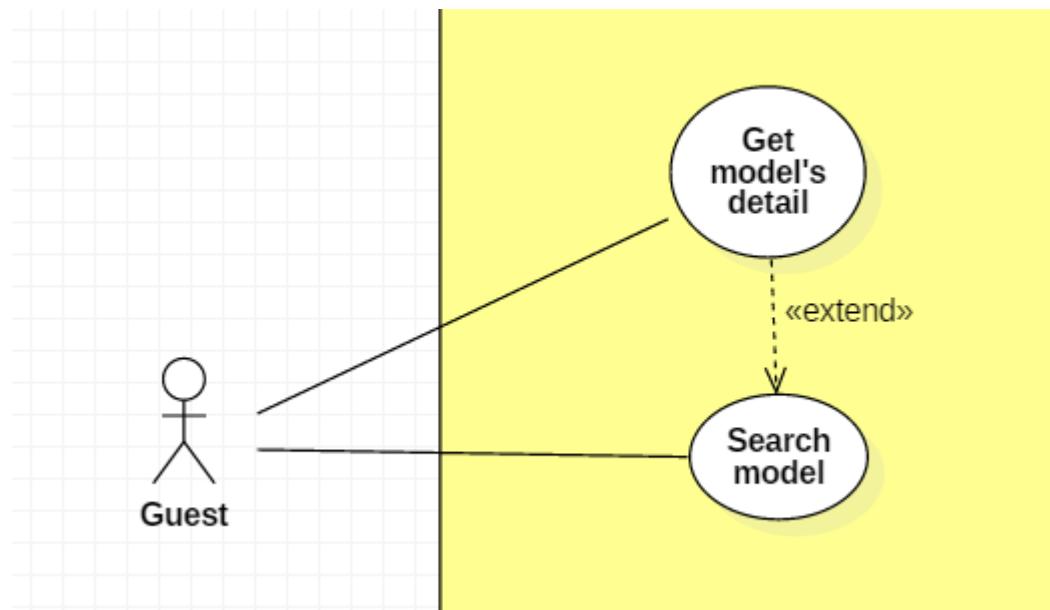
Exceptions: N/A

Relationships: N/A

Business Rules:

- A dropdown list with all product series values will be displayed when guest clicks on option “Product Series” in SEARCH BY dropdown list
- A dropdown list with all series title values will be 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	<p>System displays the selected model's detail page with the following labels and text:</p> <ul style="list-style-type: none"> ● Big image: img ● Manufacturer: text ● Scale: text ● Series: text ● Released Date: text ● Price: text ● Description: text

		<p>System displays a slideshow of images divided into following categories:</p> <ul style="list-style-type: none">● Package● Item picture● Other picture● Contents● About item● Color● Assembly guide
--	--	---

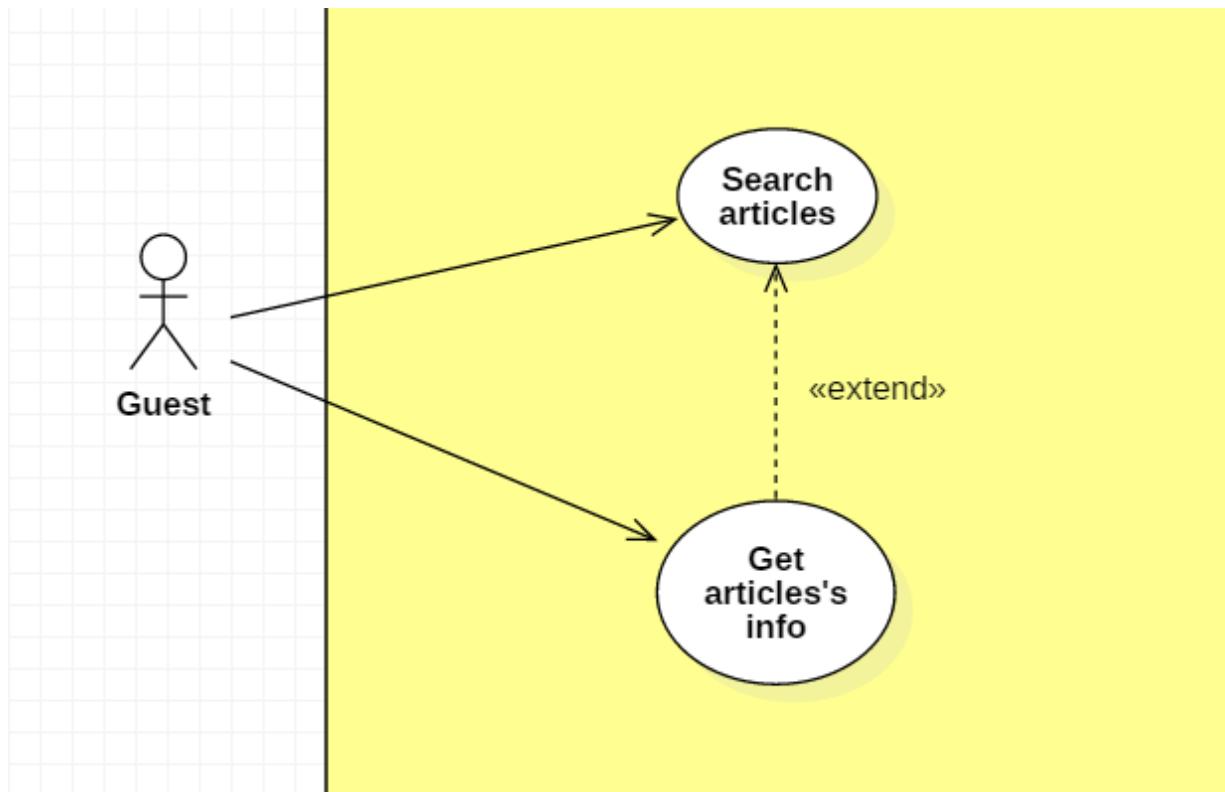
Alternative Scenario: N/A

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	Khanhn		
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	<p>System displays search news view with the following elements:</p> <ul style="list-style-type: none">• 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

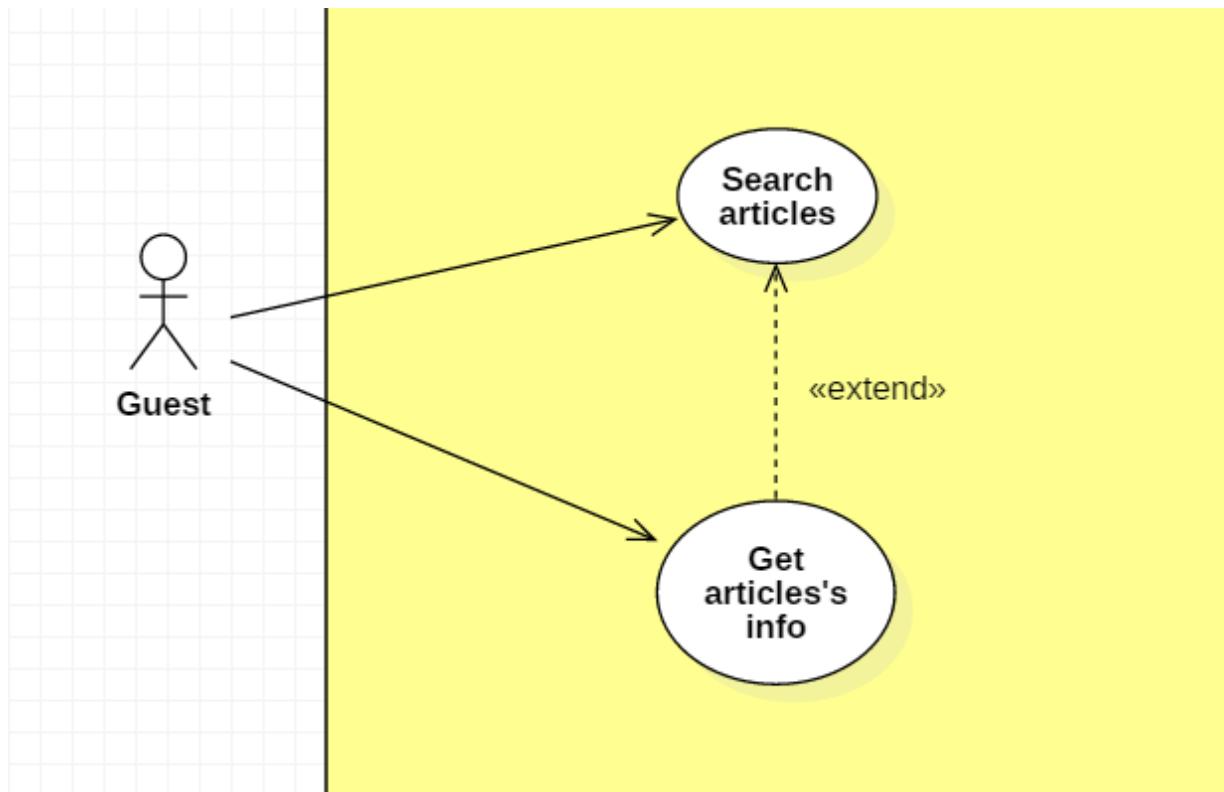
Exceptions: N/A

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 article's info		
Author	Khanhn		
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: <ul style="list-style-type: none">• The title of the article: text• The article's content

Alternative Scenario:

N/A

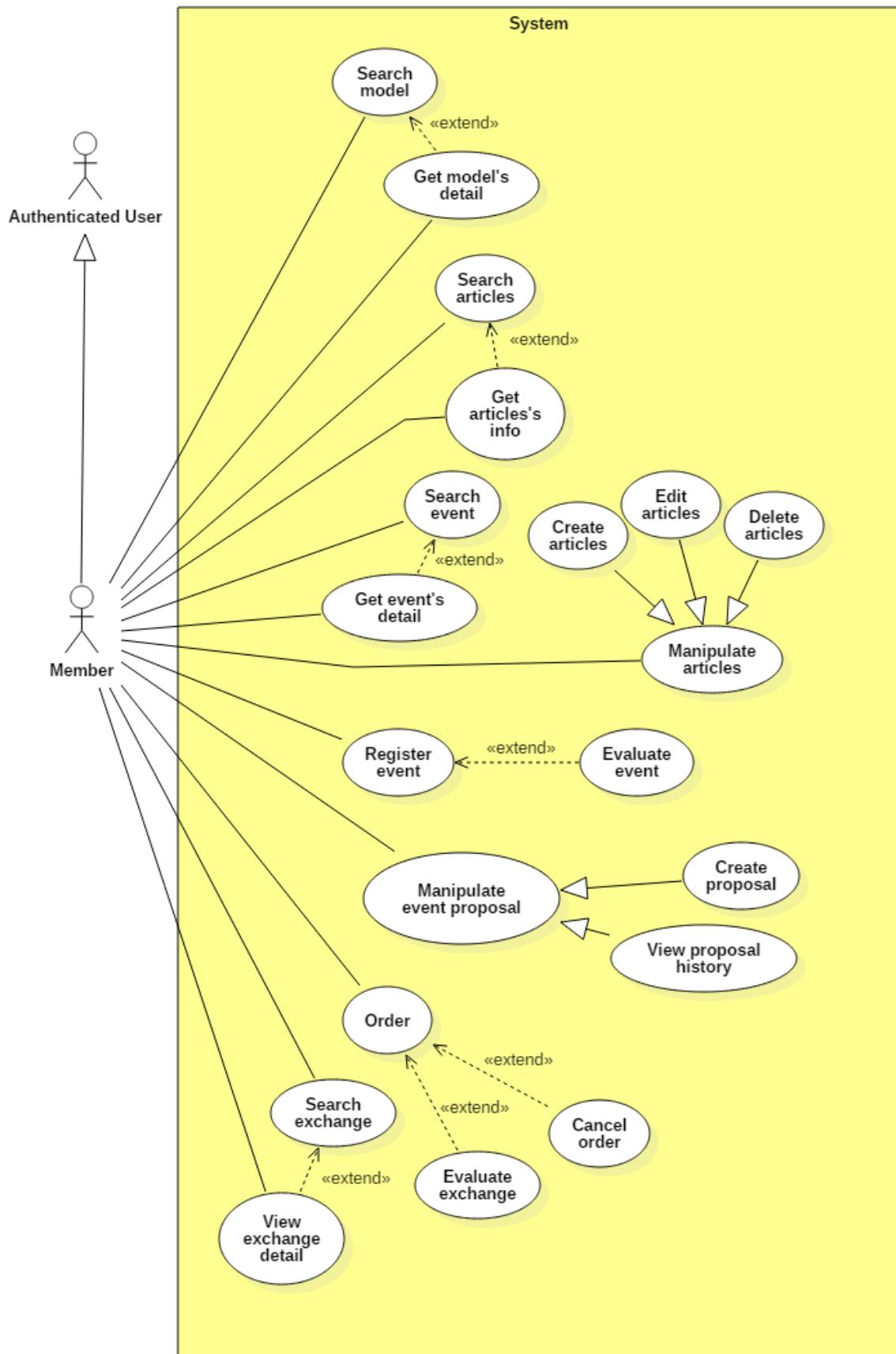
Exceptions: N/A

Relationships: N/A

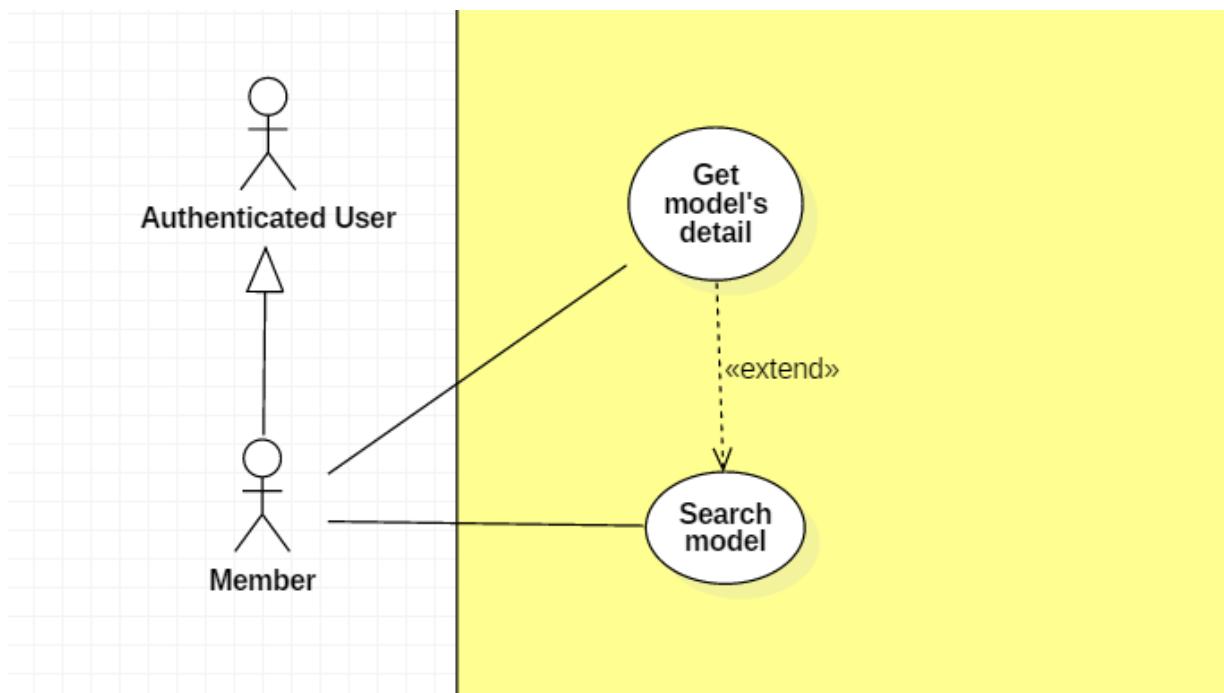
Business Rules:

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	<p>System displays search model view with the following elements:</p> <ul style="list-style-type: none">• Search navbar: free text input• SEARCH BY: dropdown list, includes values:<ul style="list-style-type: none">○ 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.

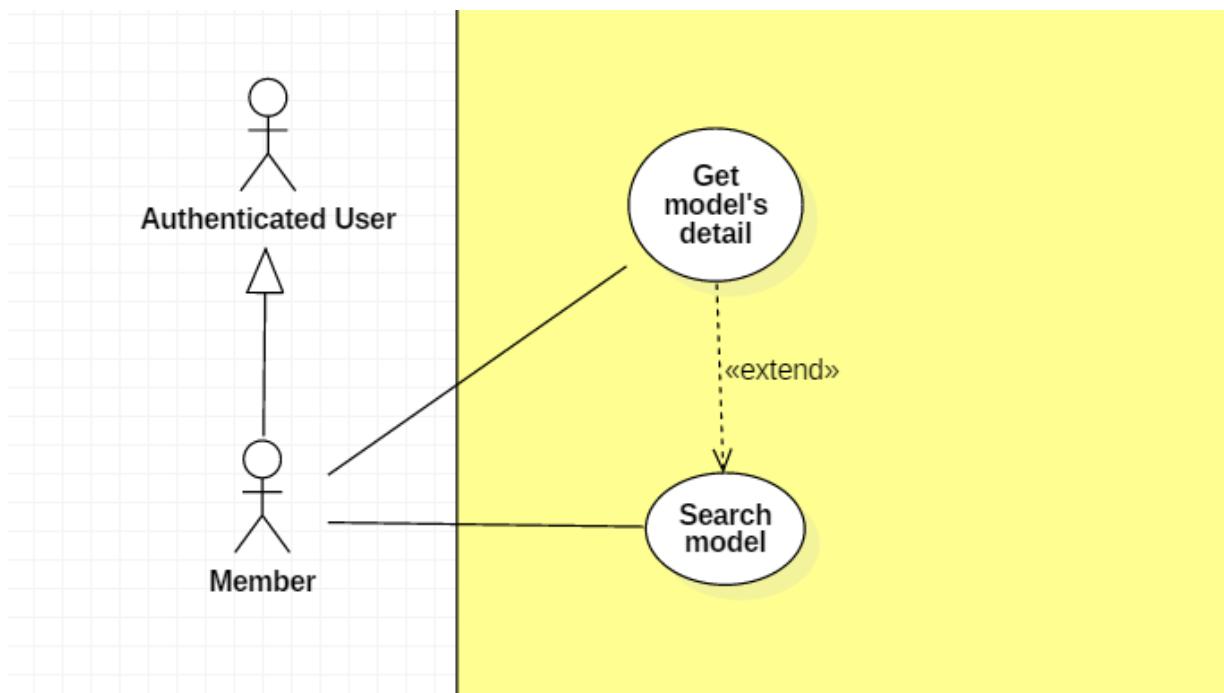
Exceptions: N/A

Relationships: N/A

Business Rules:

- A dropdown list with all product series values will be displayed when guest clicks on option “Product Series” in SEARCH BY dropdown list
- A dropdown list with all series title values will be 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:

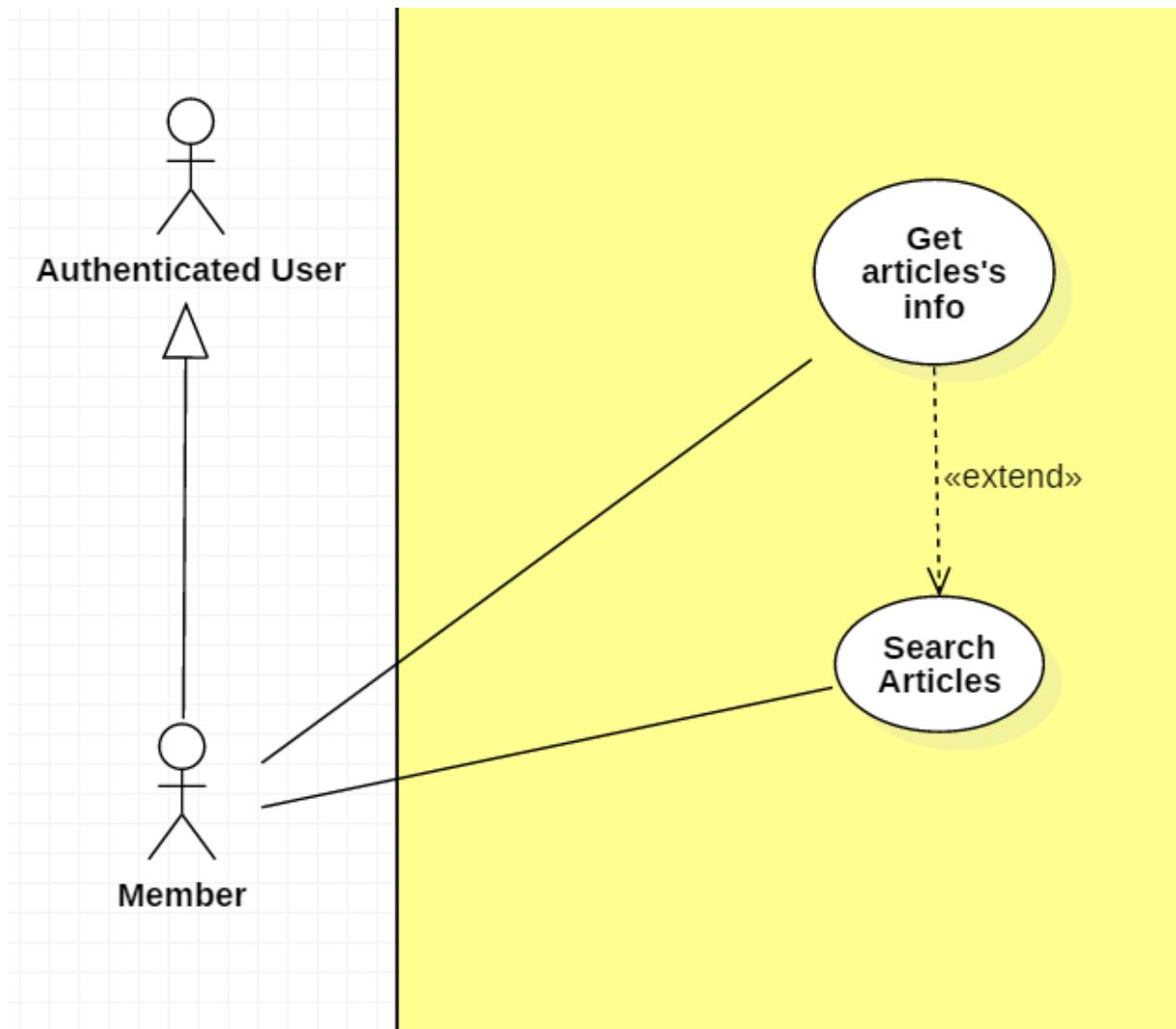
- **Success:** Model's detail page is shown with information.
- **Fail:** Show error message.

Main Success Scenario:

Step	Actor Action	System Response
1	Member clicks on a model and goes to model detail view	<p>System displays the selected model's detail page with the following labels and text:</p> <ul style="list-style-type: none"> • Big image: img • Manufacturer: text • Scale: text • Series: text • Released Date: text • Price: text • Description: text <p>System displays a slideshow of images divided into following categories:</p> <ul style="list-style-type: none"> • Package

		<ul style="list-style-type: none">● 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	Khanhn		
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	<p>System displays search news view with the following elements:</p> <ul style="list-style-type: none">• 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:

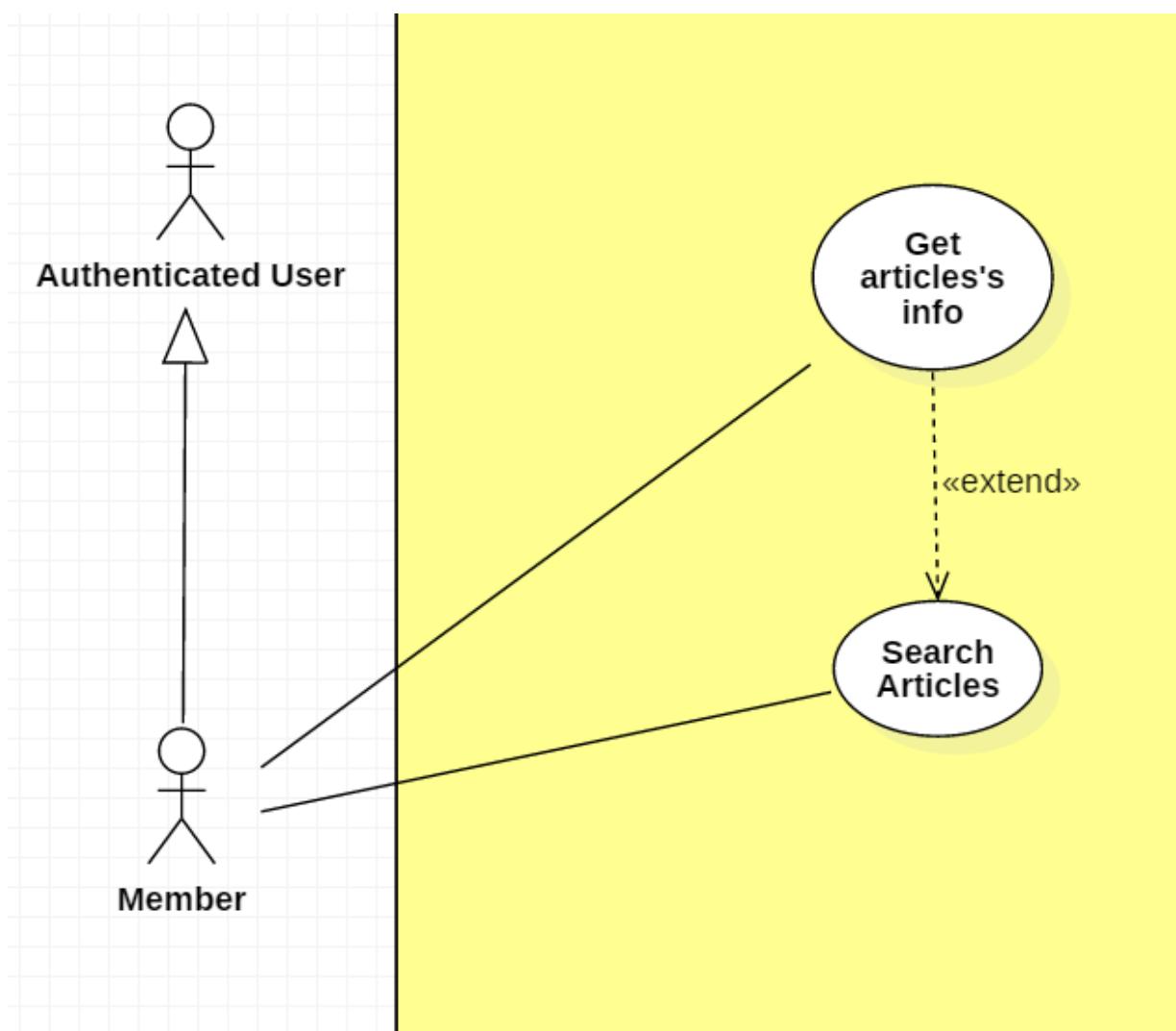
N/A

Exceptions: N/A

Relationships: N/A

Business Rules:

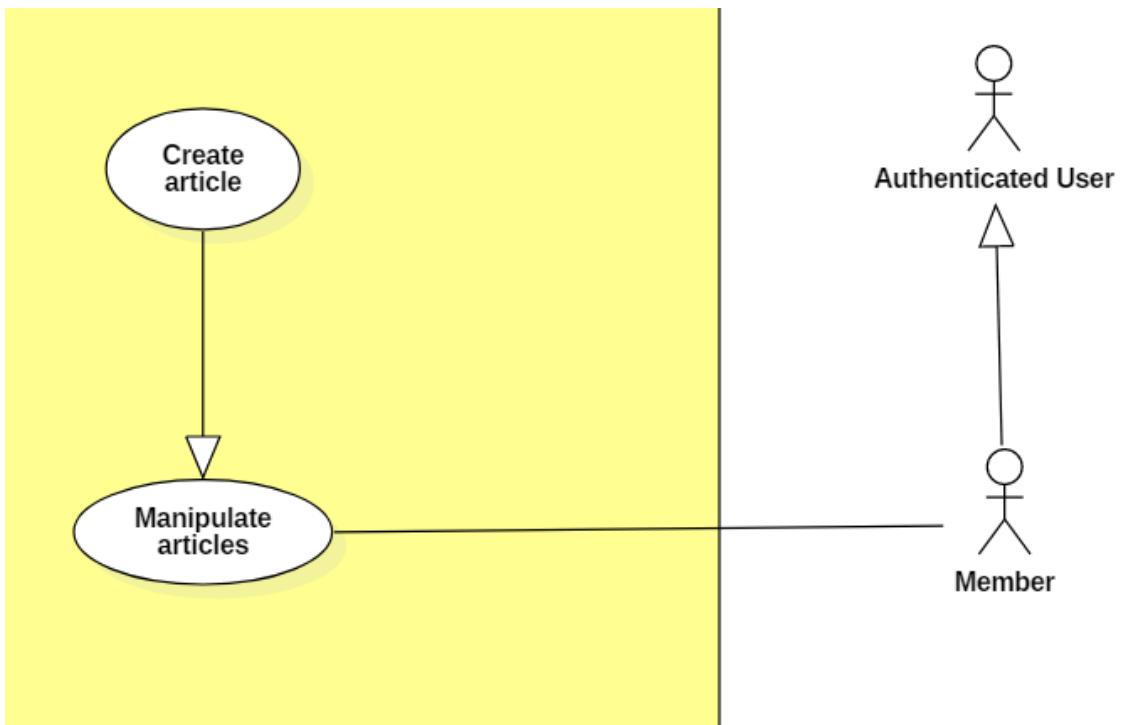
- System search article by title matching with the keyword input from guest
- A list of records will go with pagination and sorting filters

2.3.4.4. < Member > Get articles's info Usecase**USE CASE – UC _UC_M.04**

Usecase No.	M.04	Usecase Version	1.0						
Usecase Name	Get articles's info								
Author	Khanhn								
Date	03/10/2018	Priority	Normal						
Actor: - Member .									
Summary: <ul style="list-style-type: none"> - This use case allows Member to get an article's info. 									
Goal: <ul style="list-style-type: none"> - Display an article's content 									
Triggers: <ul style="list-style-type: none"> - Member sends get news's detail command to the system. 									
Preconditions: <ul style="list-style-type: none"> - Member select a article from the list of articles. 									
Post Conditions: <ul style="list-style-type: none"> - Success: The matching record is shown. - Fail: Show error message. 									
Main Success Scenario: <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="text-align: center; padding: 5px;">Step</th> <th style="text-align: center; padding: 5px;">Actor Action</th> <th style="text-align: center; padding: 5px;">System Response</th> </tr> </thead> <tbody> <tr> <td style="text-align: center; padding: 5px;">1</td> <td style="padding: 5px;">Member click on a article and goes to article detail view</td> <td style="padding: 5px;"> System displays news's detail view with the following elements: <ul style="list-style-type: none"> • The title of the article: text • The article's content </td> </tr> </tbody> </table>				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: <ul style="list-style-type: none"> • The title of the article: text • The article's content
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: <ul style="list-style-type: none"> • 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: <ul style="list-style-type: none"> • 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.

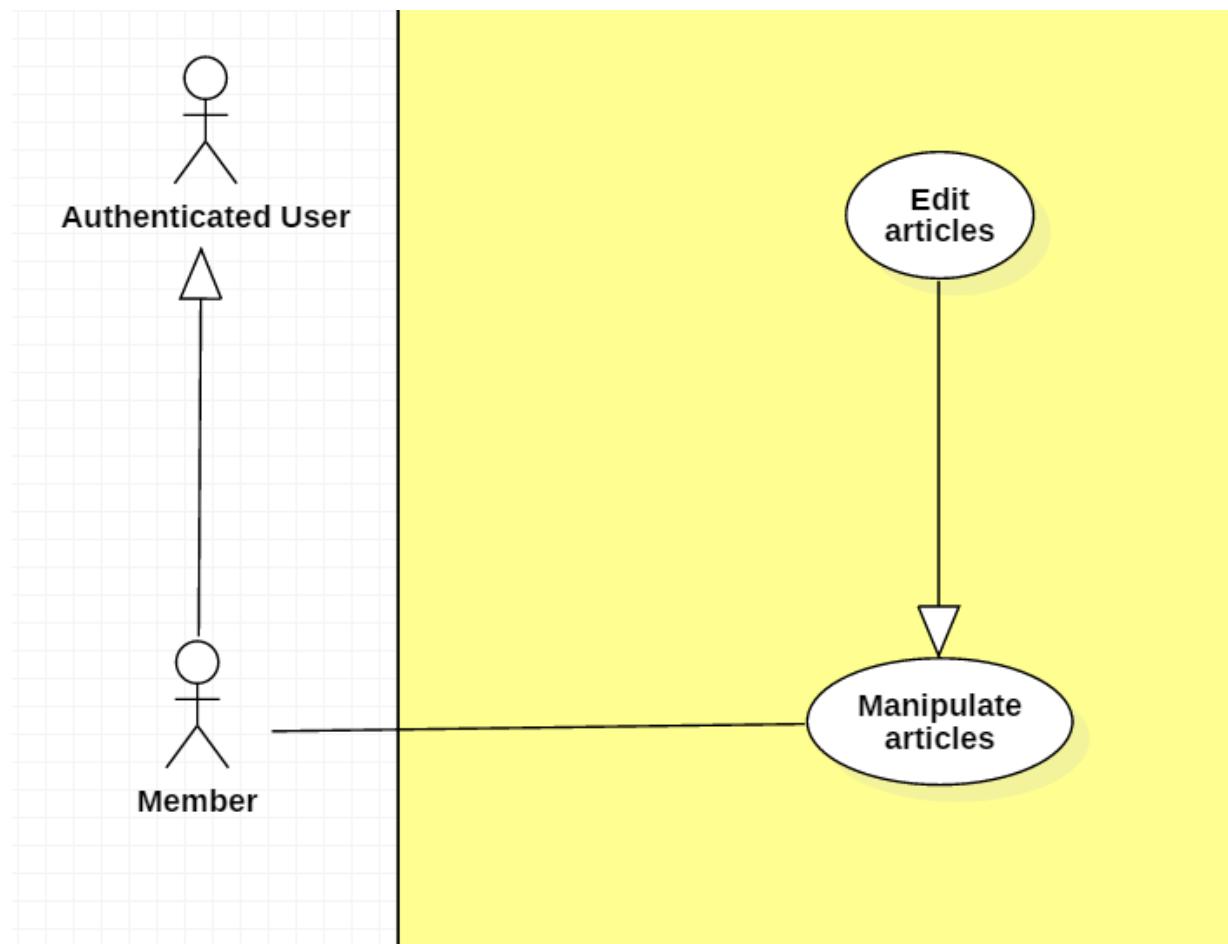
2	Member upload file too large.	System shows error message to ask the admin to upload correct required file size.
---	-------------------------------	---

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:			
<ul style="list-style-type: none"> - Member 			
Summary:			
<ul style="list-style-type: none"> - This use case allows member to edit an articles. 			
Goal:			
<ul style="list-style-type: none"> - The articles is updated in database 			
Triggers:			
<ul style="list-style-type: none"> - Member edit an article and submit it. 			
Preconditions:			
<ul style="list-style-type: none"> - User must login into system with role Member. - Member viewing an article that they posted - Member selects “Edit article” 			
Post Conditions:			
<ul style="list-style-type: none"> - 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	<p>“Edit” view is shown with following labels and fields:</p> <ul style="list-style-type: none"> • 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:

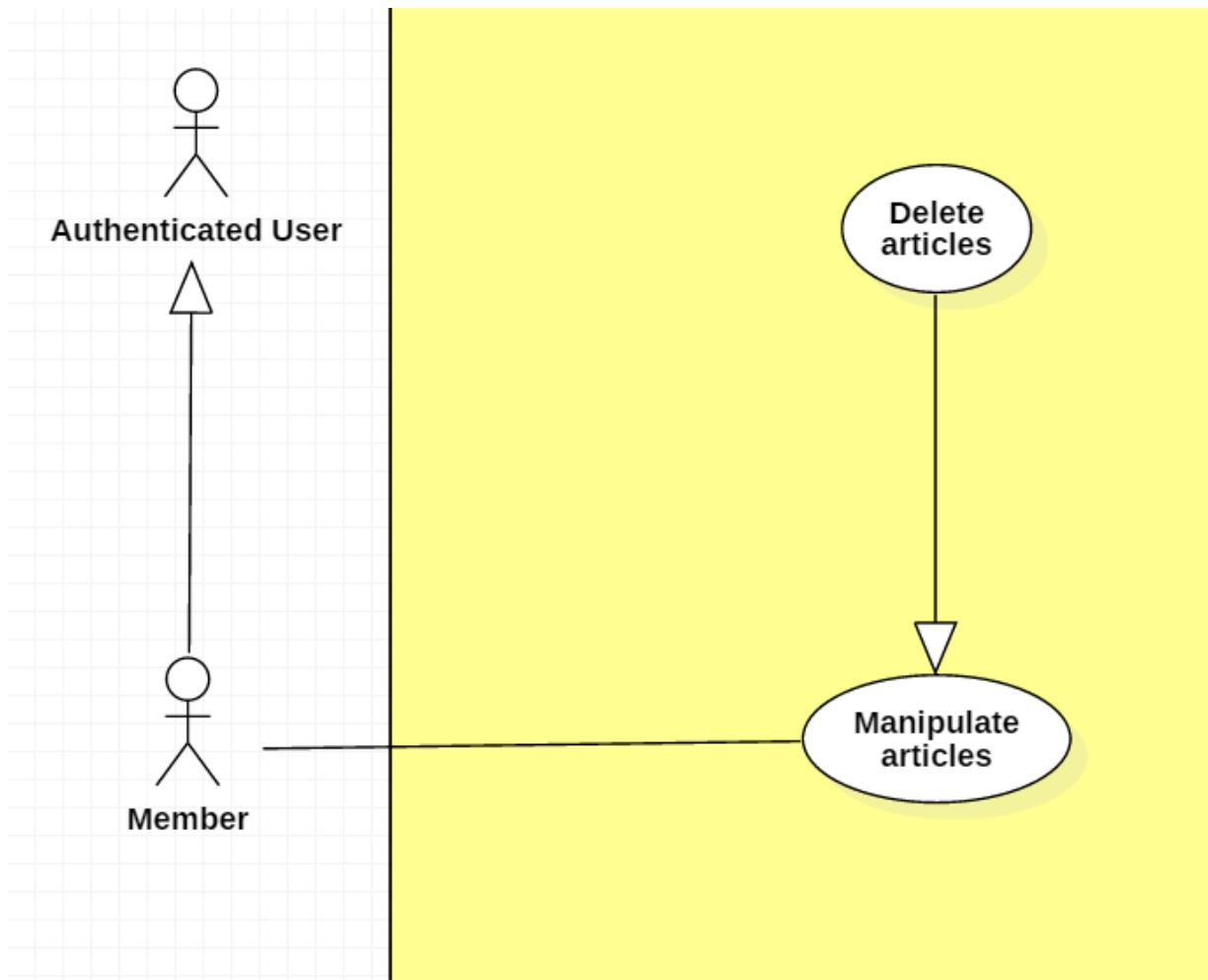
No	Cause	System Response
1	Admin submit article with empty content	System shows error message to ask the admin to enter missing required fields.
2	Admin upload file too large.	System shows error message to ask the admin to upload correct required file size.

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: <ul style="list-style-type: none"> • 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
-----------	--------------	------------------------

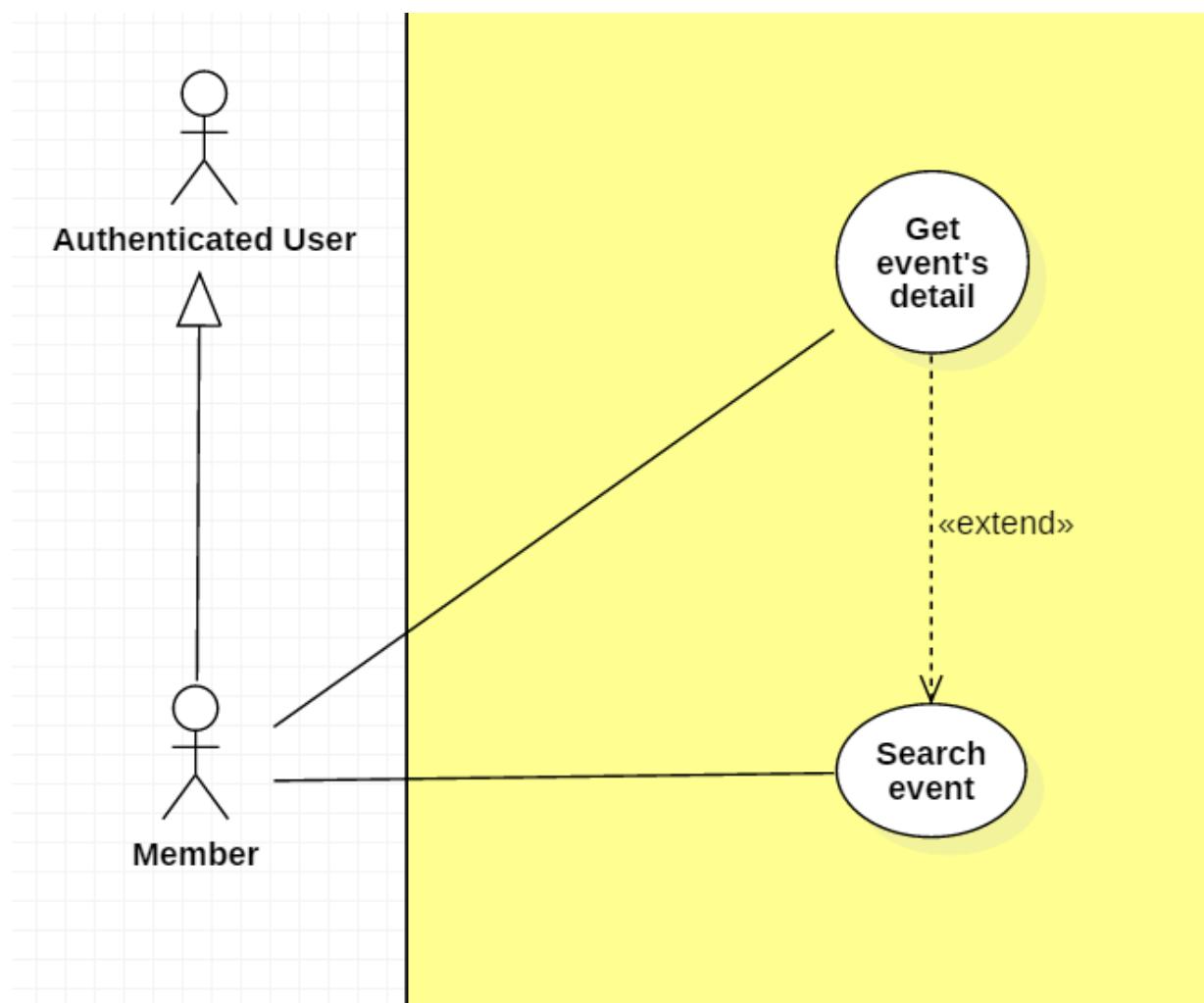
1	Article was already deleted	System shows error message “This article has been already deleted”
---	-----------------------------	--

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 _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	<p>“Edi article” view is shown with following labels and fields:</p> <ul style="list-style-type: none"> • 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.	<p>Validate data.</p> <p>System update the article in the database</p>

Alternative Scenario:

N/A

Exceptions:

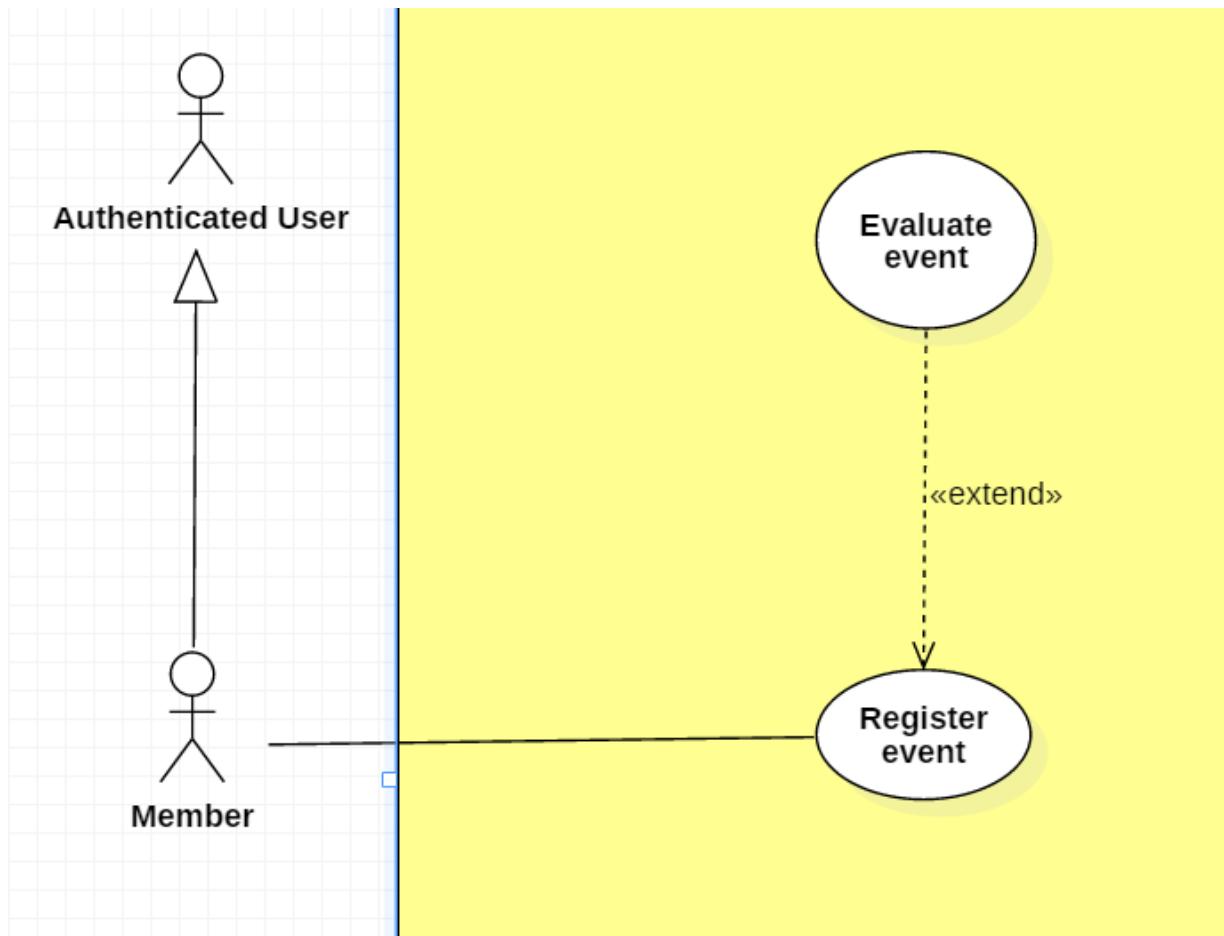
No	Cause	System Response
1	Member submit article with empty content	System shows error message to ask the admin to enter missing required fields.
2	Member upload file too large.	System shows error message to ask the admin to upload correct required file size.

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	Khanhb		
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: <ul style="list-style-type: none"> • 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	“Event detail” view is shown with following

	requires payment	labels and fields: <ul style="list-style-type: none">• 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:

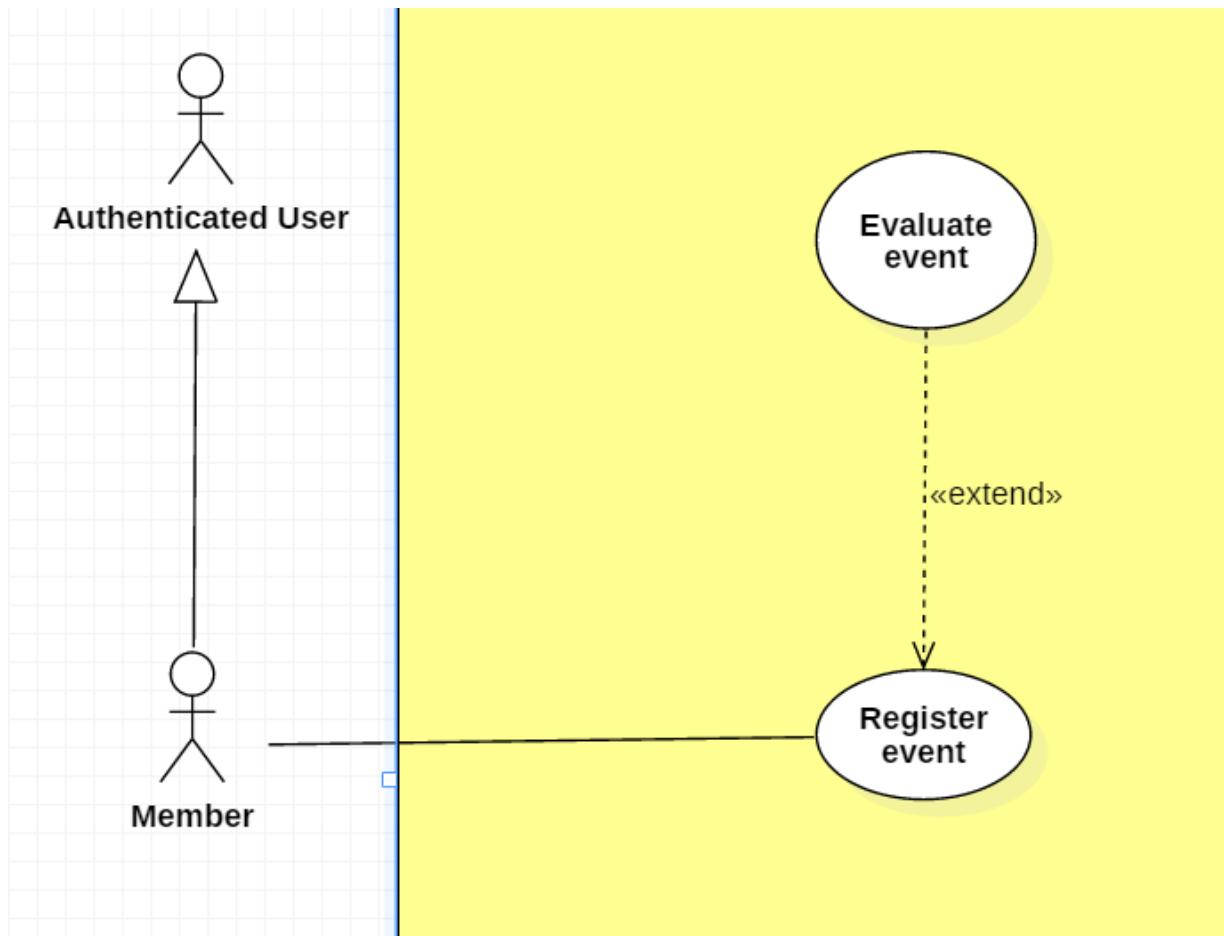
No	Cause	System Response
1	Card number or card owner's name is invalid	System shows Popup saying "Card is invalid!"
2	Number of tickets the user entered exceeds the amount of remaining tickets for that event	A message saying "The number of tickets you've entered exceeds the remaining tickets amount!" 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	Khanhb		
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.

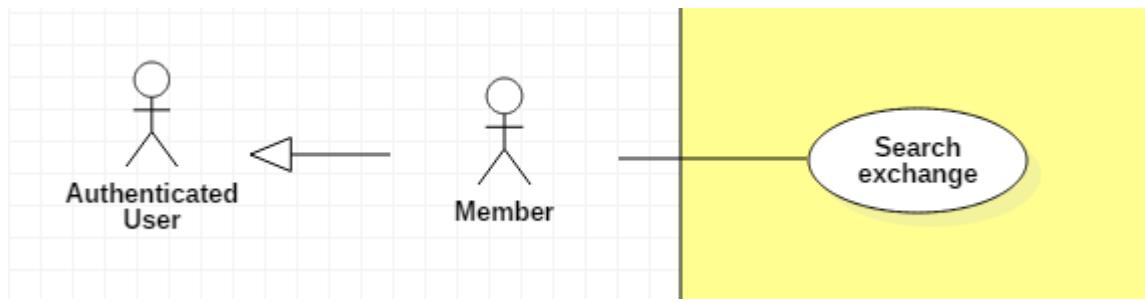
Main Success Scenario:

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:	<ul style="list-style-type: none"> - Member. 		
Summary:	<ul style="list-style-type: none"> - This use case allows member or guest search a trade by keyword and distance. 		
Goal:	<ul style="list-style-type: none"> - System create view to show search result. 		
Triggers:	<ul style="list-style-type: none"> - User sends search a trade command with keyword and distance to the system. 		
Preconditions:	<ul style="list-style-type: none"> - User must on website page. 		
Post Conditions:	<ul style="list-style-type: none"> - 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	<p>User click on “Trade market” link from header navbar or “Trade listing” on trade market page to send list all trade command.</p>	<p>System create a view show search form with following fields:</p> <ul style="list-style-type: none"> • Search input: text input , required. • “More” button <p>[Alternative Scenario 1]</p>
2	<p>User type keyword and press enter to send search command.</p>	<p>[Exception 1]</p> <ul style="list-style-type: none"> - System create a view show all result. - Filter <p>[Alternative Scenario 2]</p> <ul style="list-style-type: none"> - Group by <p>[Alternative Scenario 3]</p>

Alternative Scenario:

Alternative Scenario 1: Search with location and range

Step	Actor Action	System Response
1	<p>User click on “more” button.</p>	<p>System will show additional search form with following labels and fields:</p> <ul style="list-style-type: none"> • 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.	<p>[Exception 1]</p> <ul style="list-style-type: none"> - System create a view show all result. - Filter <p>[Alternative Scenario 2]</p> <ul style="list-style-type: none"> - Group by <p>[Alternative Scenario 3]</p>

Alternative Scenario 2 : Filter Scenario.

Step	Actor Action	System Response
1	User select on filter option send filter 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:

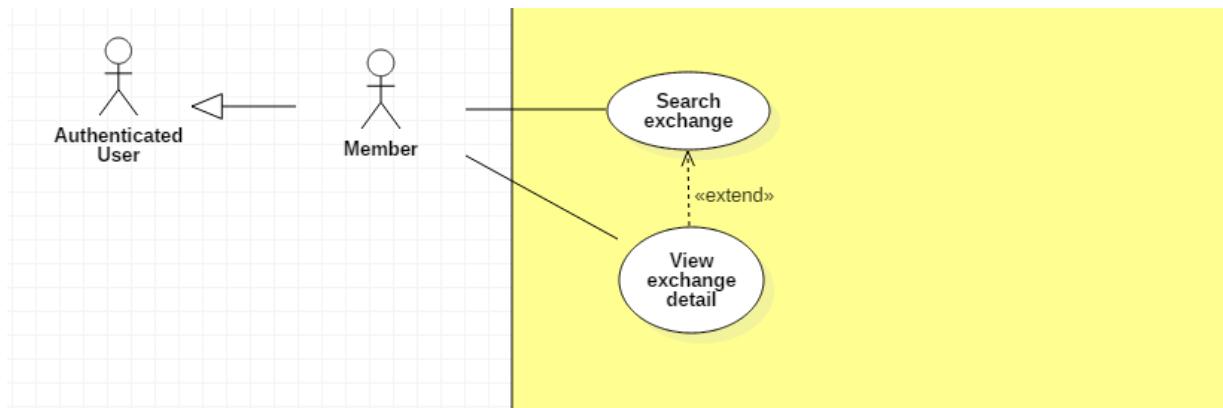
No	Cause	System Response
1	No result found.	System shows message no result found.

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 filter: 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.	<p>System get all information from database and create a view show information in 3 section:</p> <p>Product information section:</p> <ul style="list-style-type: none"> • Photo slide show • Price • Title • Trade id • Date posted • Location • Type of trade • Brand • Model

		<ul style="list-style-type: none"> • Quantity • Rating • “Send order” button. • Share this post. <p>Trader information section:</p> <ul style="list-style-type: none"> • Delivery • More trade by owner • Print this trade • Send to friend • Save as Favorite • Report this trade <p>Description section:</p> <ul style="list-style-type: none"> • Description
--	--	--

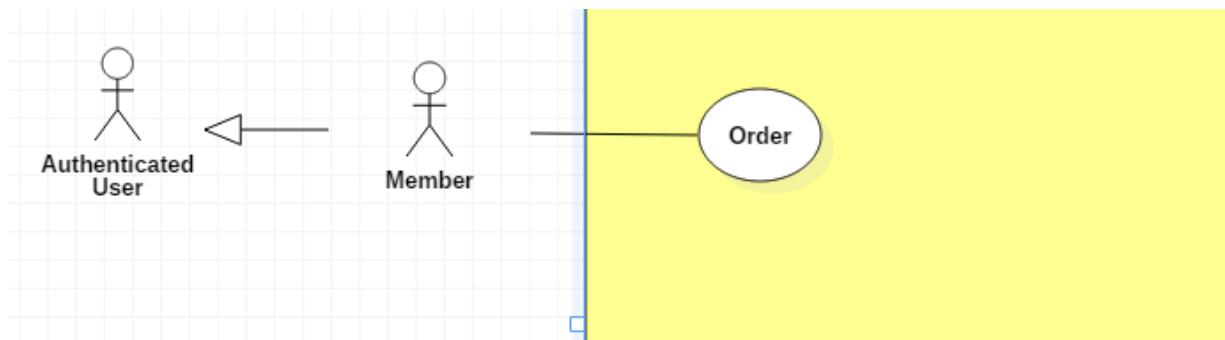
Alternative Scenario: N/A

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 waiting 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.	<p>System create view show a order form with following labels and fields:</p> <ul style="list-style-type: none"> • Quantity: Number, required, positive integer and below than 1000. • Billing information: button • Cancel order: button <p>[Alternative scenario 1]</p>
2	User fills out the form and click “Billing information”	

	button	<p>Validate quantity data.</p> <p>[Exception 1]</p> <p>System create view show billing information form with following labels and fields:</p> <ul style="list-style-type: none"> • 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 <p>[Alternative Scenario 2]</p> <ul style="list-style-type: none"> • Change your location: button <p>[Alternative Scenario 3]</p> <ul style="list-style-type: none"> • Confirm order: button • Cancel order: button <p>[Alternative scenario 1]</p>
3	User click “Confirm order” button.	<p>System show a confirm modal with message “Are you sure to order?” and two option Y/N.</p>
4	User confirm order by click “Yes” to send order command.	<p>The order details will save to database and system will notify to owner of this trade post this request order.</p> <p>System close modal confirm and show message your order has been sent.</p>

	User cancel confirm order by click “No” .	System close modal back to billing information view.
--	---	--

Alternative Scenario:

Alternative Scenario 1: Cancel order

Step	Actor Action	System Response
1	User click “Cancel 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.	<p>System create and show modal with following labels and fields:</p> <ul style="list-style-type: none"> • 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. <p>And “submit” modal button.</p>
2	User fills out the form.	
3	User click “submit” button.	<p>System close modal and get text data from “Your full address” field and fill to “Billing address” field on trader information form.</p>

Exceptions:

No	Cause	System Response
1	User input quantity is not valid.	

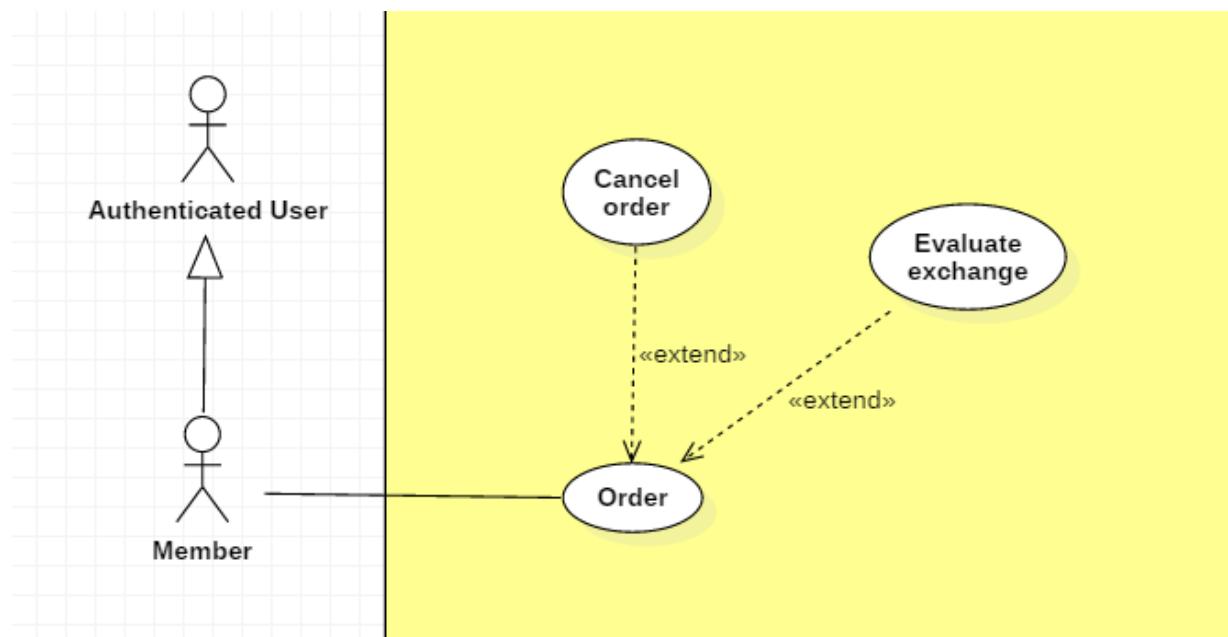
		System shows error message to ask member enters valid quantity.
2	User do not allows location permission.	System shows error message to ask member provide required permission.

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 _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 cancel order by click “Yes” to send cancel order command.	System remove this order request from database.
	User cancel confirm cancel order by click “No” .	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.	<p>System create and show modal with following labels and fields:</p> <ul style="list-style-type: none"> • Reason: text input, length 10 - 250. <p>And “submit” modal button.</p>
2	User fills out the form.	
3	User click “submit” button.	<p>System close modal and set order state to cancelled then notify to trade post owner the reason.</p>

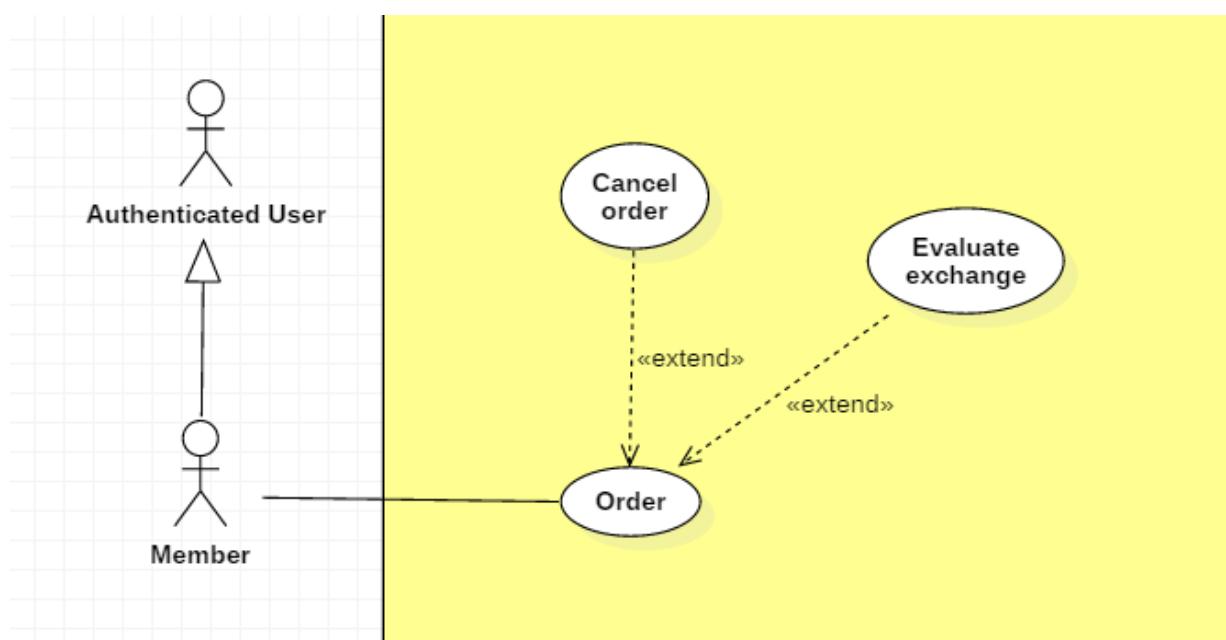
Exceptions: N/A

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.	<p>System create view show a feedback form with following labels and fields:</p> <ul style="list-style-type: none">• 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 feebback data to database system and show message feedback successfully.
--	-------------------	---

Alternative Scenario: N/A

Exceptions:

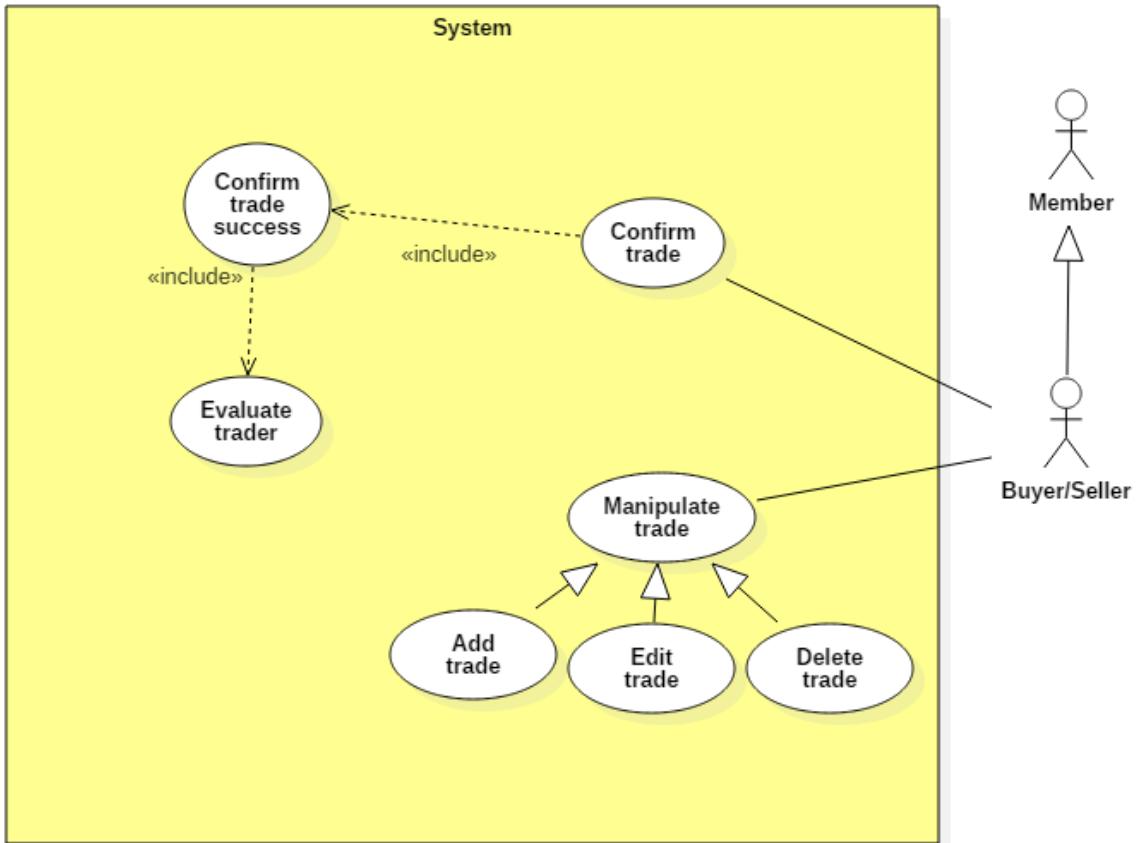
No	Cause	System Response
1	Member send submit feedback command without fills out the form.	System shows error message to ask member enters missing required fields.

Relationships: N/A

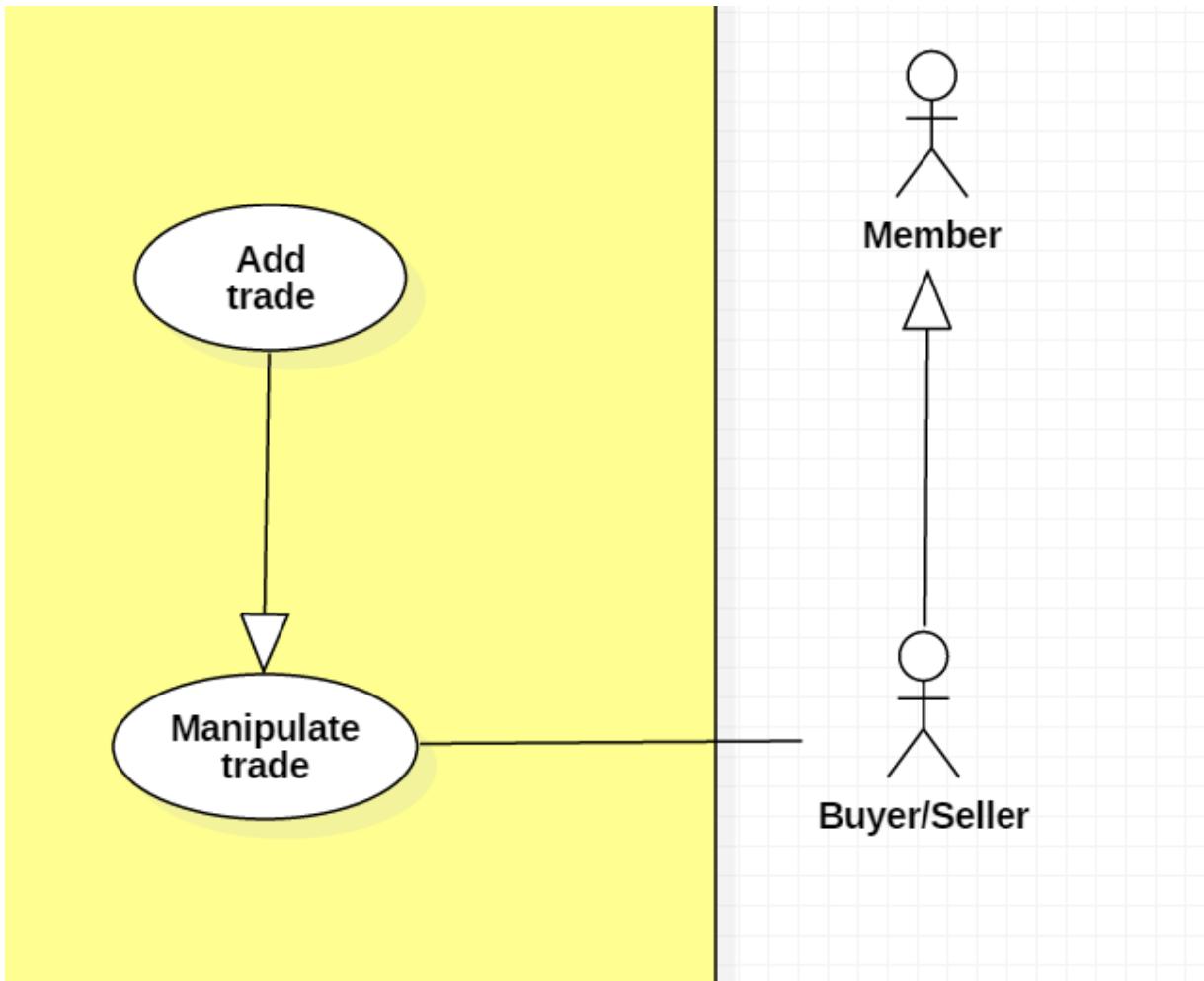
Business Rules:

- 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.	<p>Create post new trade view is shown with following labels and fields:</p> <ul style="list-style-type: none">• 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.

		<ul style="list-style-type: none"> • Model: text input, required, length 3 – 50. • Description : text input, required, length 20 – 5000. <p>Create trade post owner's information view is shown with following labels and fields:</p> <ul style="list-style-type: none"> • 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 <p>[Alternative Scenario 1]</p> <ul style="list-style-type: none"> • Change your location: Button <p>[Alternative Scenario 2]</p> <p>And “Post your trade” submit form button.</p>
2	User fills out the form.	
3	User sends post new trade command by click “Post your trade” button.	<p>Validate data.</p> <p>[Exception 1]</p> <p>[Exception 2]</p> <p>Add new trade post in the database.</p> <p>Update trader information (optional).</p> <p>System shows message trade post has been created successfully and waiting to administrators approve.</p>
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.	<p>[Exception 3]</p> <p>System auto gets your location and generates to text address then autoforms to address field.</p>

Alternative Scenario 2: Change your location Scenario.

Step	Actor Action	System Response
1	User click “Change your location” button.	<p>System create and show modal with following labels and fields:</p> <ul style="list-style-type: none"> • 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. <p>And “submit” modal button.</p>
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:

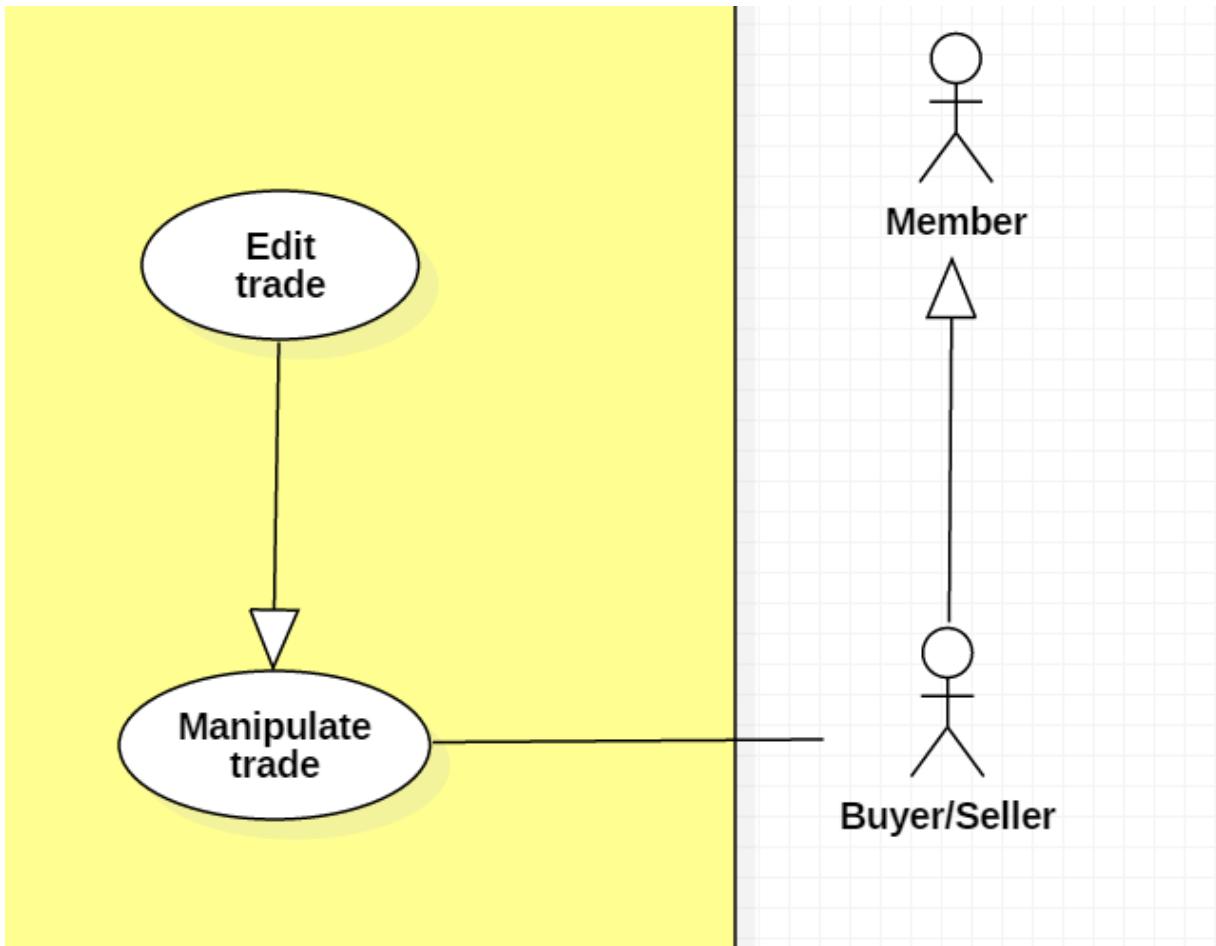
No	Cause	System Response
1	User sends post new trade command by click “Post your trade” without filling out full form.	System shows error message to ask member enters missing required fields.
2	User upload file too large.	System shows error message to ask member upload correct requied file size.
3	User 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 permissstion 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.	<p>Create edit trade post view is show with following labels and fields:</p> <ul style="list-style-type: none">• 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.

		<ul style="list-style-type: none"> • Brand: text input, required, length 3 – 50. • Model: text input, required, length 3 – 50. • Description : text input, required, length 20 – 5000. <p>Create trader information view is show with following labels and fields:</p> <ul style="list-style-type: none"> • 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 <p>[Alternative Scenario 1]</p> <ul style="list-style-type: none"> • Change your location: Button <p>[Alternative Scenario 2]</p> <p>And “Save your change” submit form button.</p>
2	User fills out the form.	
3	User sends edit trade post command by click “Save your change” button.	<p>Validate data.</p> <p>[Exception 1]</p> <p>[Exception 2]</p> <p>New trade post information is updated in the database.</p> <p>System shows message trade post has been updated successfully and waiting to</p>

		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: <ul style="list-style-type: none"> • 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.

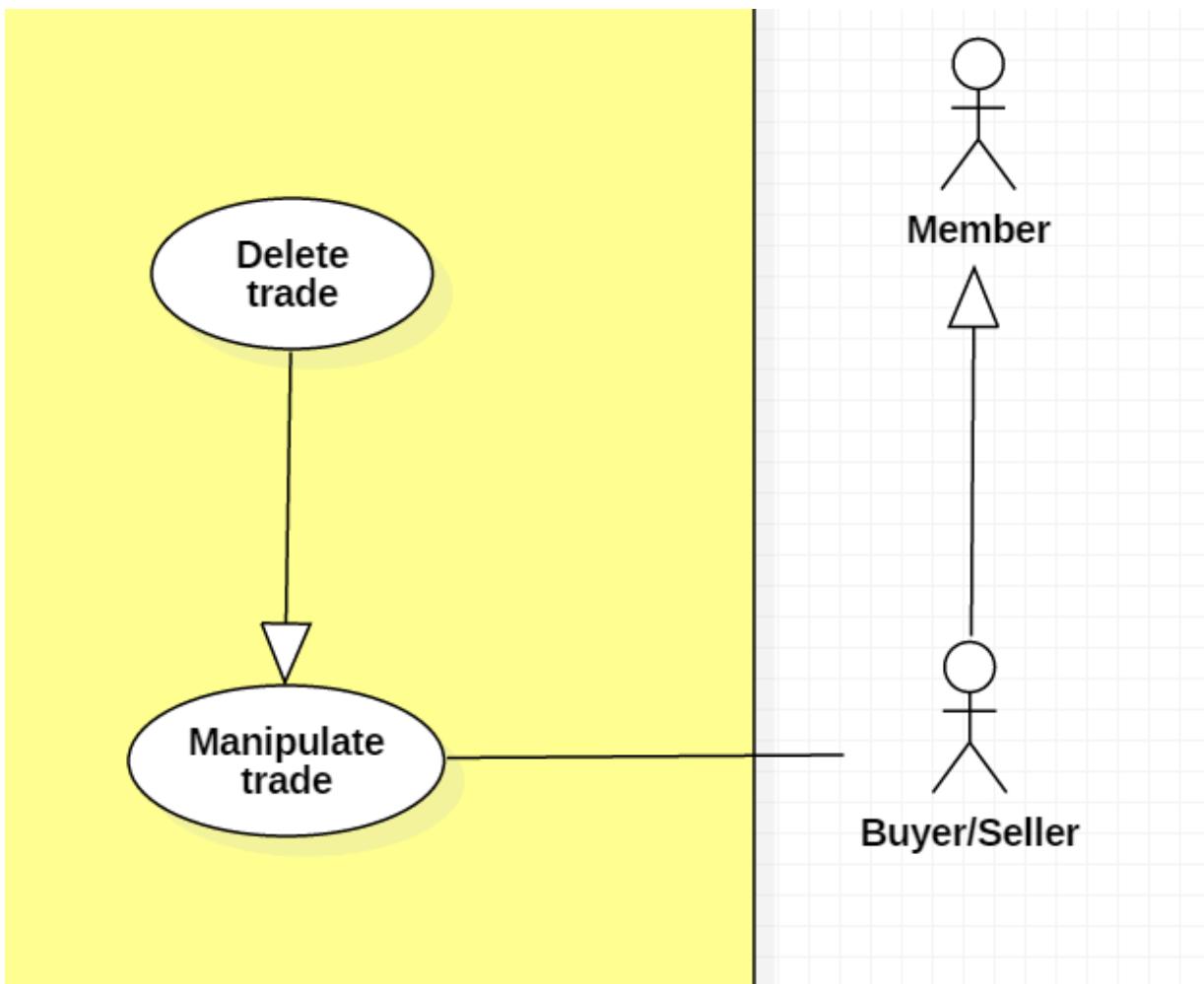
Relationships: N/A

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 permission 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:

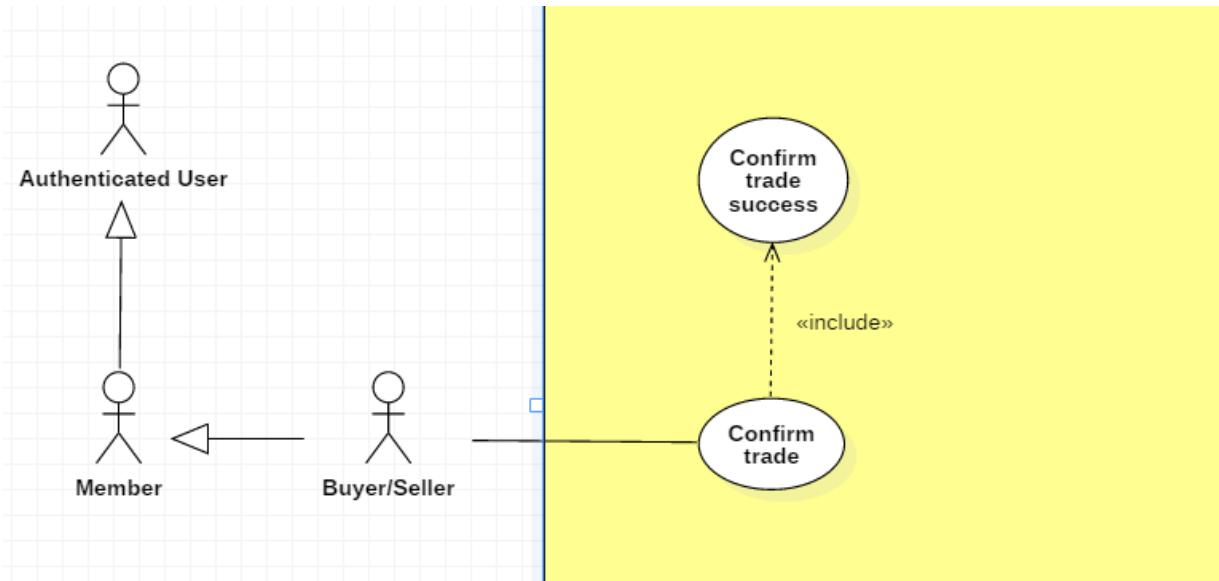
No	Cause	System Response
1	The trade post already deleted before.	System shows error message the trade post is no longer existed.

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:

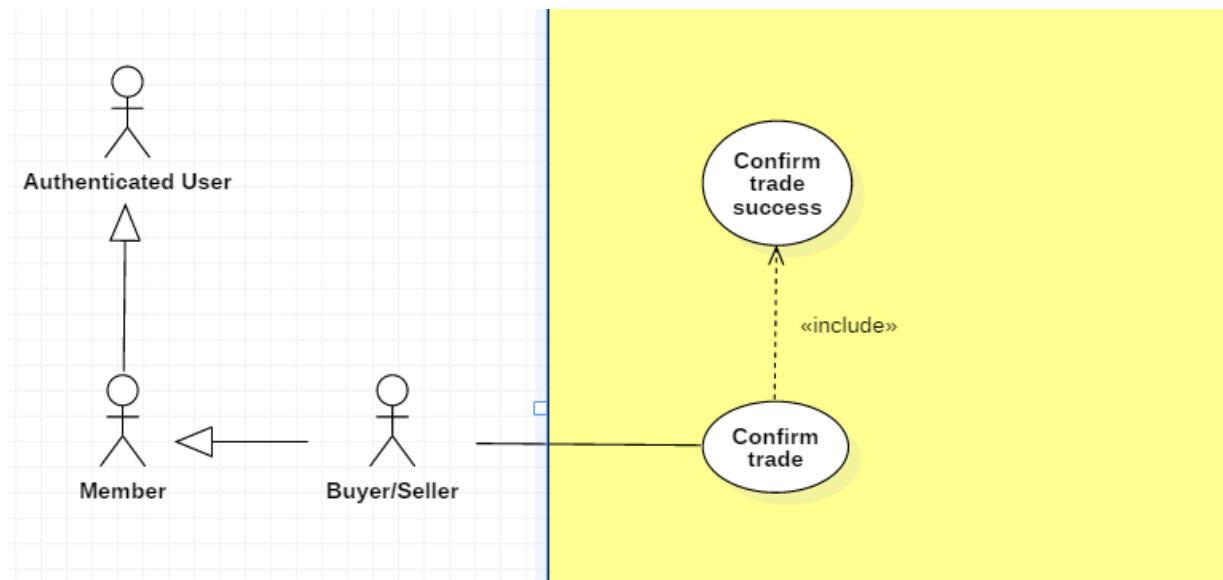
Step	Actor Action	System Response
1	<p>User click “Accept” button on request order list to send accept request command.</p> <p>[Alternative 1]</p>	<p>System notify to who ordered the request has been accepted and set order state to accepted. The trade post owner information will send to trader.</p>

Alternative Scenario:**Alternative Scenario 1: Decline request order.**

Step	Actor Action	System Response
1	If buyer/seller decline request by click “reject” button.	<p>System create and show modal with following labels and fields:</p> <ul style="list-style-type: none"> • Reason: text input, length 10 - 250. <p>And “submit” modal button.</p>
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.	<p>System create and show modal with following labels and fields:</p> <ul style="list-style-type: none"> • Reason: text input, length 10 - 250. <p>And “submit” modal button..</p>
2	User fills out the form.	
3	User click “submit” button to send cancel payment command.	<p>System close dialog and set order state to cancelled then notify to trader traded has been cancelled with reason.</p>

Alternative Scenario 2: Confirm by click “no”

Step	Actor Action	System Response
1	User confirm by click “No”	<p>System close dialog and nothing has change.</p>

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
<p>Actor:</p> <ul style="list-style-type: none"> - Buyer/Seller. <p>Summary:</p> <ul style="list-style-type: none"> - This use case allows buyer/seller to feedback a trade. <p>Goal:</p> <ul style="list-style-type: none"> - After a trade has been confirm success, system will send a feedback form to owner and request them to feedback the trade. <p>Triggers:</p> <ul style="list-style-type: none"> - User sends feedback trader command with userid to the system. <p>Preconditions:</p> <ul style="list-style-type: none"> - User must login into system with role Member. - User have a trade has been confirm trade success. <p>Post Conditions:</p> <ul style="list-style-type: none"> - Success: System show message feedback successfully and store feedback data to database system. - Fail: Show error message. <p>Main Success Scenario:</p>			
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:	

		<ul style="list-style-type: none"> • 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.	<p>Validate form data.</p> <p>[Exception 1]</p> <p>System store feedback data to database system and show message feedback successfully.</p>

Alternative Scenario: N/A

Exceptions:

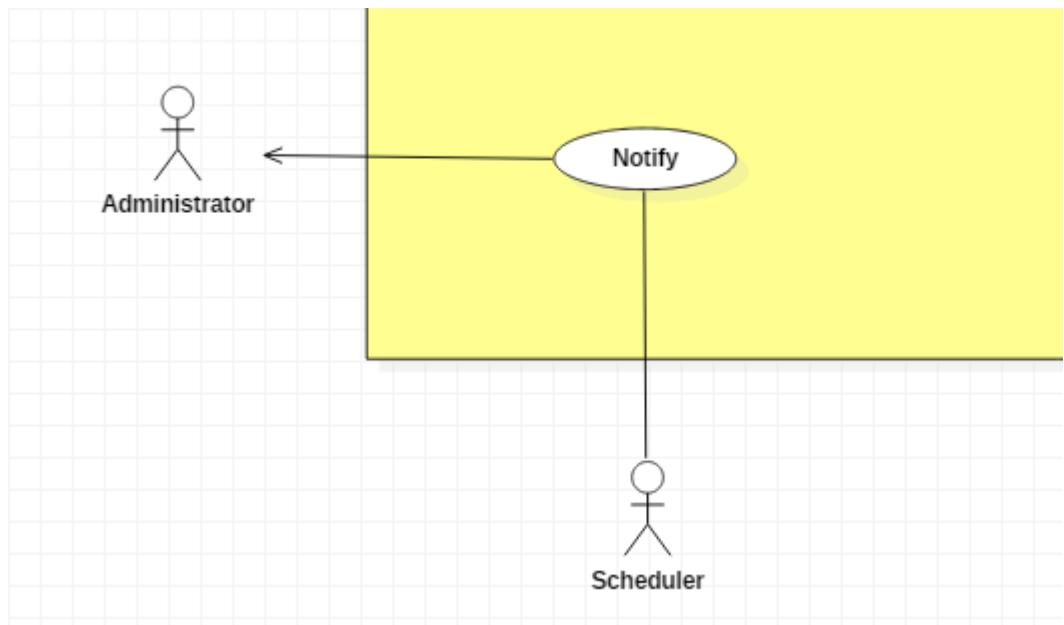
No	Cause	System Response
1	User send submit feedback command without fills out the form.	System shows error message to ask member enters missing required fields.

Relationships: N/A

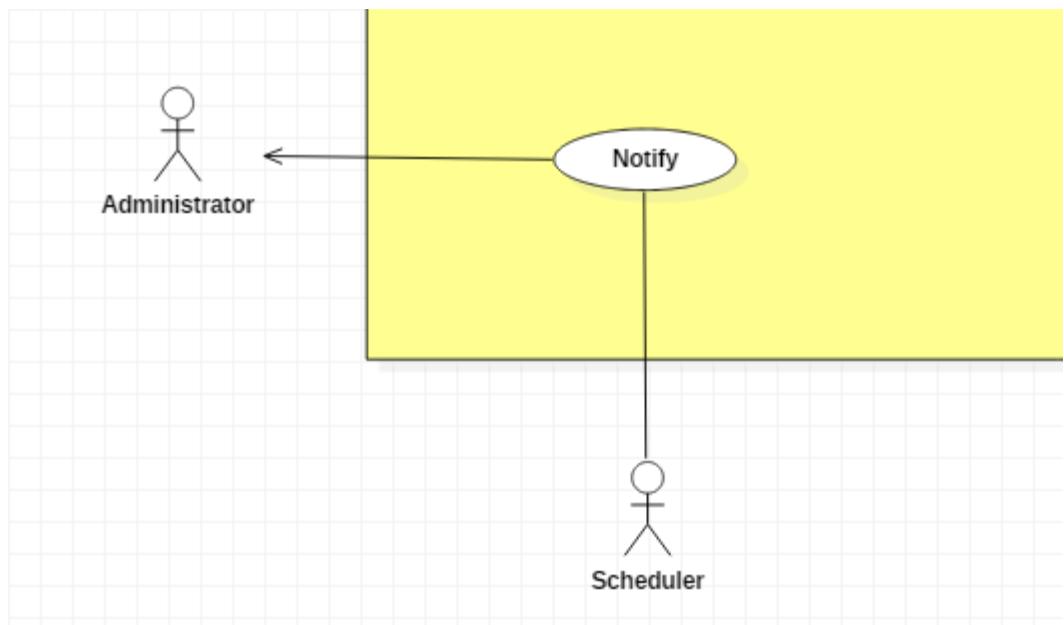
Business Rules:

- Buyer/Sell must be do the feedback to continuous using system.

2.3.6. < Scheduler > Overview Usecase



2.3.6.1. < Scheduler > Notify Usecase



USE CASE – UC _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:									
<ul style="list-style-type: none"> - 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:									
<ul style="list-style-type: none"> - Success: Notification is sent to user. - Fail: Log error to console and file.. 									
Main Success Scenario:									
<table border="1"> <thead> <tr> <th>Step</th><th>Actor Action</th><th>System Response</th></tr> </thead> <tbody> <tr> <td>1</td><td>Scheduler sends send notification command</td><td>System sends notification to user</td></tr> </tbody> </table>				Step	Actor Action	System Response	1	Scheduler sends send notification command	System sends notification to user
Step	Actor Action	System Response							
1	Scheduler sends send notification command	System sends notification to user							
Alternative Scenario: N/A									
Exceptions: N/A									
Relationships:									
<ul style="list-style-type: none"> - < 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
 - Font style: Helvetica, Arial, Helvetica Neue, Roboto, Arial, Droid Sans, sans-serif
 - Font size: 12px -26px
 - Color: green, black, red, white, blue, Light Slate Grey, Fuego, Honeysuckle...
 - 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

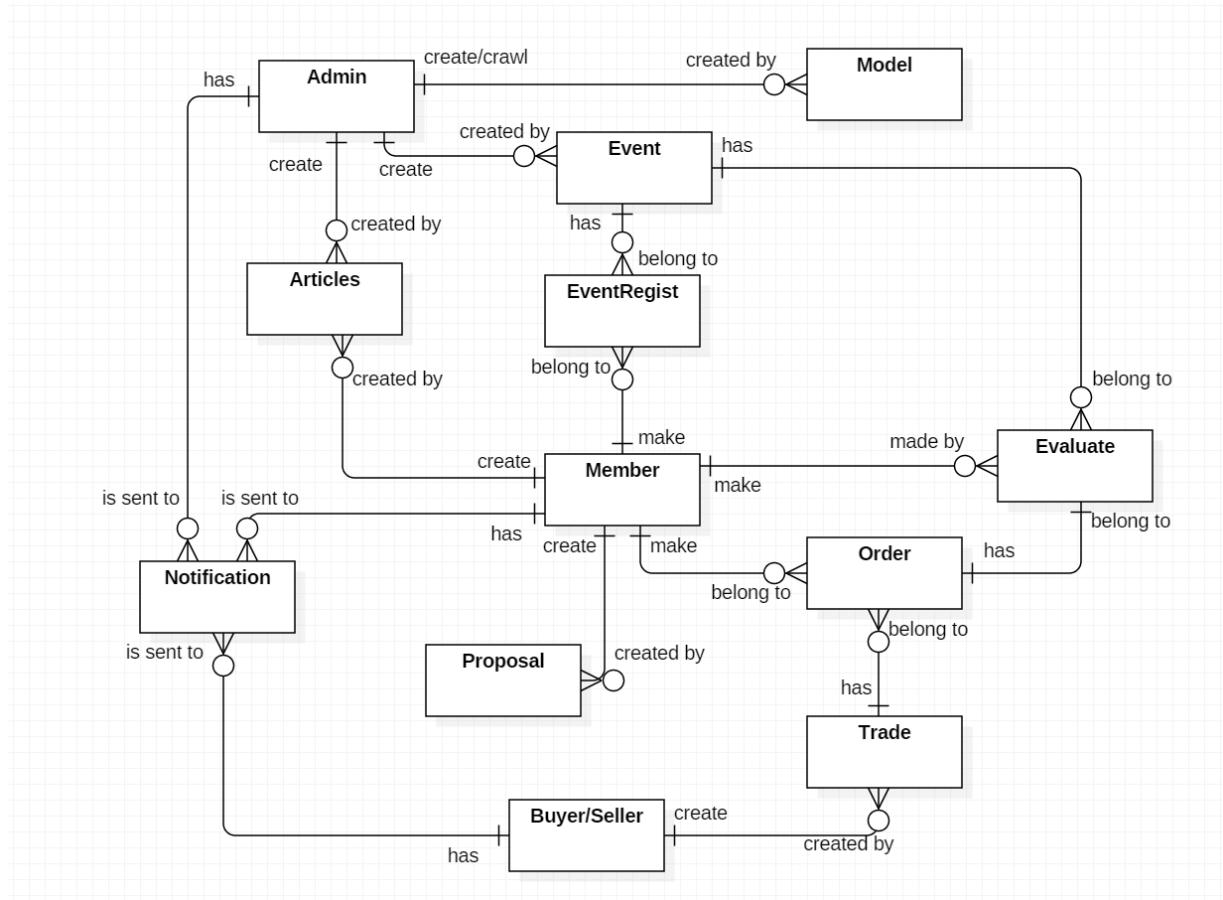
3.6. Portability

- 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

D. Report No. 4 Software Design Description

1. Design Overview

This document describes the technical and user interface design of **GWA System**. It includes the architectural design, the detailed design of common functions and business functions and the design of database model.

The architectural design describes the overall architecture of the system and the architecture of each main component and subsystem.

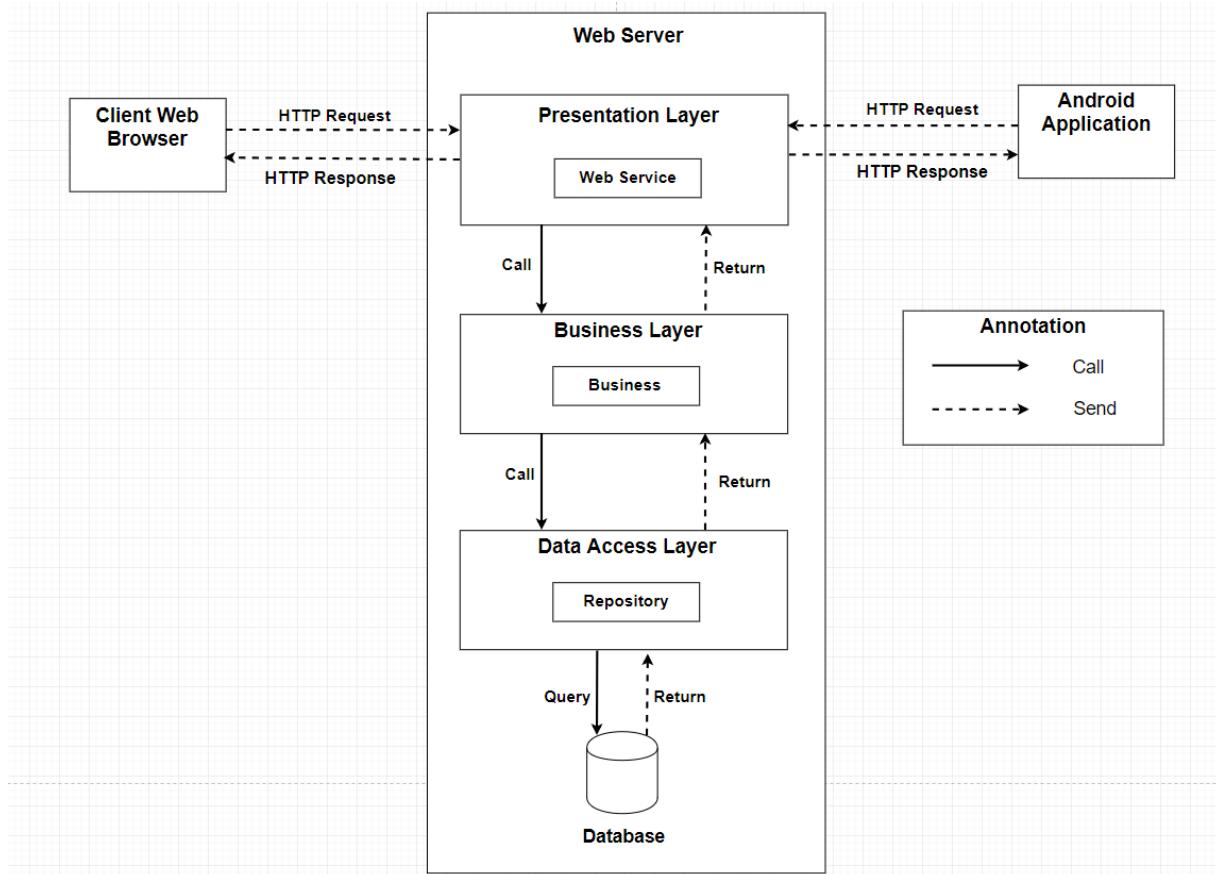
The detailed design describes static and dynamic structure for each components and functions. It includes class diagram, class explanations and sequence diagrams for each use cases.

The database design describes the relationship between entities and detail of each entity.

Document overview:

- Section 2: gives an overall description of the system architecture design.
- Section 3: gives component diagrams that describe the connection and integration of the system.
- Section 4: gives the detail design description which includes class diagram, class explanation, and sequence diagram to details the application functions.
- Section 5: describe screens design.
- Section 6: describe a fully attributed ERD.
- Section 7: describe algorithms.

2. System Architectural Design



2.1. Web application architecture description

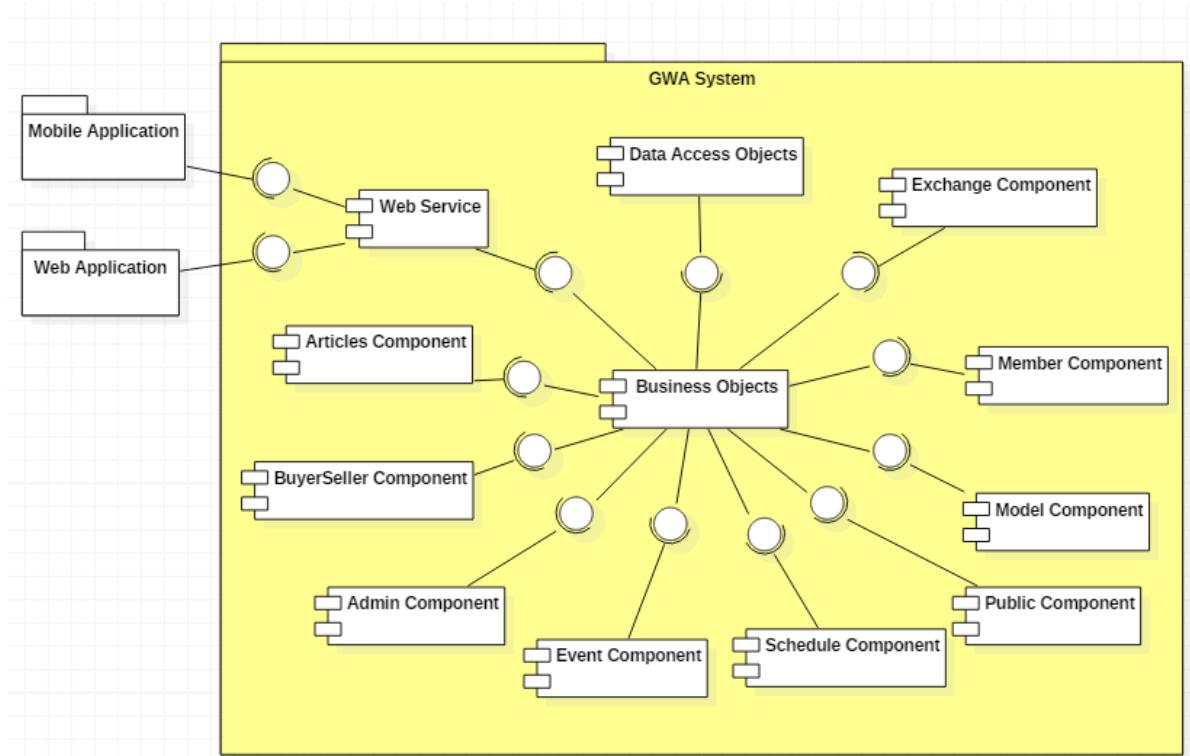
In Web Application, the system is developed under 3-layers architecture style, with Spring Boot technology. We choose this architecture for Web application because of the following advantages:

- We can separate our business code into logical layers, each layer performing a specific role within the application. Presentation layer is responsible for handling all user interface and browser communication logic, whereas business layer is responsible for executing business logics and data access layer is responsible for executing anythings that relate to database.
- The presentation layer provides Rest API for web and mobile app to communicate by **json** data through network environment, so we can use the business code in web service without repeating the code.
- It is easy to maintain, to apply object oriented concept and to update data provider queries.

This project follows 3-layers architecture with the following components:

- **WebController:** is the part of presentation layer that acts like event handler to handle user interaction. It receives request from user by request mapping and select the appropriate view to return to user.
- **RestController:** is the part of presentation layer that provides api for client to communicate and reads data from response, call the business layer to execute business logic, then return **json** data to client. Web and Mobile app will render that **json** data to UI.
- **Service:** is the part of business layer that is responsible for executing complex business logics. This component will get user's data from controller, call repository and perform business operations.
- **Model:** is the part of data access layer that contains the mapping of actual tables in database to classes.
- **Repository:** is the part of data access layer that uses models to execute CRUD database operations. This component only execute operations that are related to and is meant for directly accessing the database.

3. Component Diagram

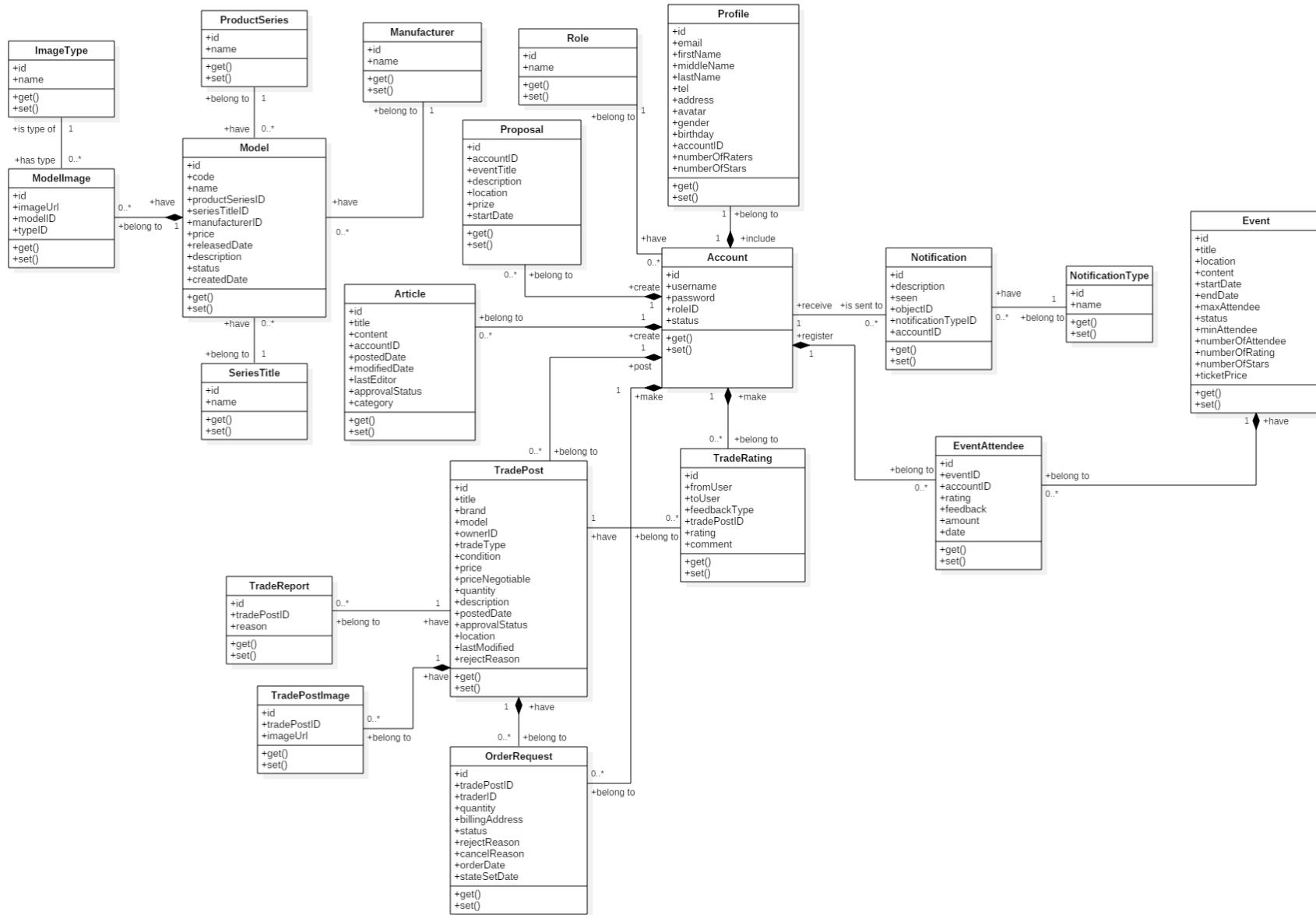


Component Dictionary: Describes components	
Web Application	Web application package
Mobile Application	Mobile application package
Web Service	Provide API for client to interact with the system
Admin Component	Component to handle admin activities in the system
BuyerSeller Component	Component to handle buy/seller activities in the system
Member Component	Component to handle member activities in the system
Public Component	Component to handle guest activities in the system
Schedule Component	Component to handle scheduler in the system
Event Component	Component about manipulating events
Articles Component	Component about manipulating articles
Exchange Component	Component about manipulating exchange
Model Component	Component about manipulating models
Business Objects	Common objects to handle domain business operations for

	each components
Data Access Objects	Component to handle interaction between the system and database

4. Detailed Description

4.1. Class Diagram



Class Name	Mapping column with Conceptual diagram	Description
Model	Model	Contain model's information
ModelImage	N/A	Not exist in conceptual diagram. It's used to contain models's images
ImageType	N/A	Not exist in conceptual diagram. It's used to contain types of images
ProductSeries	N/A	Not exist in conceptual diagram. It's used to contain product series information
SeriesTitle	N/A	Not exist in conceptual diagram. It's used to contain series title information
Manufacturer	N/A	Not exist in conceptual diagram. It's used to contain manufacturers information
Article	Articles	Contain article's information
Proposal	Proposal	Contain member's proposal information
Account	User	Contain user's information for authentication and authorization
Profile	N/A	Not exist in conceptual diagram. It's used to contain user's profile information
Role	N/A	Not exist in conceptual diagram. It's used to contain user's role information
TradePost	Trade	Contain buyer/seller's trade post information
TradePostImage	N/A	Not exist in conceptual diagram. It's

		used to contain trade post's images
OrderRequest	Order	Contain information of user's order request to trade
TradeRating	Evaluate	Contain trade rating information
TradeReport	N/A	Not exist in conceptual diagram. It's used to contain user's report on trade post
Event	Event	Contain event information
EventAttendee	EventRegist	Contain user's registration on events
Notification	Notification	Contain notification information
NotificationType	N/A	Not exist in conceptual diagram. It's used to contain notification type information

4.2. Class Diagram Explanation

4.2.1. Model

Attribute

Attribute	Type	Visibility	Description
id	int	Private	Unique identifier of model
name	String	Private	Model name
productSeries	ProductSeries	Private	Model's product series
seriesTitle	SeriesTitle	Private	Model's series title
manufacturer	Manufacturer	Private	Model's manufacturer
price	float	Private	Model's price
realeasedDate	String	Private	Model's released date
description	String	Private	Model's description
status	String	Private	Model's status
createdDate	String	Private	Model's created date

Method

Method	Return Type	Visibility	Description
getter	Attribute type	Public	Get attribute value
setter	Void	Public	Set value of attribute

4.2.2. SeriesTitle

Attribute

Attribute	Type	Visibility	Description
id	int	Private	Unique identifier of series title
name	String	Private	Series title's name

Method

Method	Return Type	Visibility	Description
getter	Attribute type	Public	Get attribute value
setter	Void	Public	Set value of attribute

4.2.3. ProductSeries

Attribute

Attribute	Type	Visibility	Description
id	int	Private	Unique identifier of product series
name	String	Private	Product series's name

Method

Method	Return Type	Visibility	Description
getter	Attribute type	Public	Get attribute value
setter	Void	Public	Set value of attribute

4.2.4. Manufacturer

Attribute

Attribute	Type	Visibility	Description
id	int	Private	Unique identifier of manufacturer

name	String	Private	Manufacturer's name
------	--------	---------	---------------------

Method

Method	Return Type	Visibility	Description
getter	Attribute type	Public	Get attribute value
setter	Void	Public	Set value of attribute

4.2.5. ModelImage

Attribute

Attribute	Type	Visibility	Description
id	int	Private	Unique identifier of model image
imageUrl	String	Private	Image's url
model	Model	Private	The model which image belongs to
type	ImageType	Private	Image's type

Method

Method	Return Type	Visibility	Description
getter	Attribute type	Public	Get attribute value
setter	Void	Public	Set value of attribute

4.2.6. ImageType

Attribute

Attribute	Type	Visibility	Description
id	int	Private	Unique identifier of image type
name	String	Private	Image type's name

Method

Method	Return Type	Visibility	Description
getter	Attribute type	Public	Get attribute value
setter	Void	Public	Set value of attribute

4.2.7. Article

Attribute

Attribute	Type	Visibility	Description
id	int	Private	Unique identifier of article
title	String	Private	Article title
content	String	Private	Article content detail information
account	Account	Private	User who posts article
postedDate	String	Private	Article posted date
modifiedDate	String	Private	Article last modified date
lastEditor	String	Private	Username of the last editor
approvalStatus	String	Private	Article's approval status
category	String	Private	Article's category

Method

Method	Return Type	Visibility	Description
getter	Attribute type	Public	Get attribute value
setter	Void	Public	Set value of attribute

4.2.8. Proposal

Attribute

Attribute	Type	Visibility	Description
id	int	Private	Unique identifier of proposal
account	Account	Private	User who posts proposal
eventTitle	String	Private	Proposal event title
description	String	Private	Proposal description
location	String	Private	Proposal event location
prize	String	Private	Proposal event winng rewards
startDate	String	Private	Proposal event start date

Method

Method	Return Type	Visibility	Description
getter	Attribute type	Public	Get attribute value

setter	Void	Public	Set value of attribute
--------	------	--------	------------------------

4.2.9. Account

Attribute

Attribute	Type	Visibility	Description
id	int	Private	Unique identifier of account
username	String	Private	Account username
password	String	Private	Account password
role	Role	Private	Account role
status	String	Private	Account status

Method

Method	Return Type	Visibility	Description
getter	Attribute type	Public	Get attribute value
setter	Void	Public	Set value of attribute

4.2.10. Role

Attribute

Attribute	Type	Visibility	Description
id	int	Private	Unique identifier of role
name	String	Private	Role name

Method

Method	Return Type	Visibility	Description
getter	Attribute type	Public	Get attribute value
setter	Void	Public	Set value of attribute

4.2.11. Profile

Attribute

Attribute	Type	Visibility	Description
-----------	------	------------	-------------

id	int	Private	Unique identifier of user profile
email	String	Private	User's email
firstName	String	Private	User's first name
middleName	String	Private	User's middle name
lastName	String	Private	User's last name
tel	String	Private	User's mobile number
address	String	Private	User's address
avatar	String	Private	User's avatar as image
gender	String	Private	User's gender
birthday	String	Private	User's birthday
account	Account	Private	Profile owner

Method

Method	Return Type	Visibility	Description
getter	Attribute type	Public	Get attribute value
setter	Void	Public	Set value of attribute

4.2.12. Event

Attribute

Attribute	Type	Visibility	Description
id	int	Private	Unique identifier of event
title	String	Private	Event title
location	String	Private	Event location occur
content	String	Private	Event content
startDate	String	Private	Event start date
endDate	String	Private	Event end date
maxAttendee	int	Private	Event's maximum attendee
status	String	Private	Event status
minAttendee	int	Private	Event's minimum attendee
numberOfAttendee	int	Private	Event's current number of attendee, this attribute is for improving performance
numberOfRating	int	Private	Number of users who evaluate the events, this attribute is for improving performance in calculating event rating
numberOfStars	int	Private	Number of star rating, this attribute is for

			improving performance in calculating event rating
ticketPrice	float	Private	Price of event's ticket

Method

Method	Return Type	Visibility	Description
getter	Attribute type	Public	Get attribute value
setter	Void	Public	Set value of attribute

4.2.13. EventAttendee

Attribute

Attribute	Type	Visibility	Description
id	int	Private	Unique identifier of eventattendee
event	Event	Private	Object event
account	Account	Private	Object account which represents the event attendee
rating	int	Private	User's rating on events
feedback	String	Private	User's feedback on events
amount	int	Private	Number of tickets which a user regist for
date	String	Private	Event registration date

Method

Method	Return Type	Visibility	Description
getter	Attribute type	Public	Get attribute value
setter	Void	Public	Set value of attribute

4.2.14. Notification

Attribute

Attribute	Type	Visibility	Description
id	int	Private	Unique identifier of notification
description	String	Private	Notification description
seen	int	Private	Notification seen status

object	object	Private	Notification object message
notificationType	NotificationType	Private	Notification type object
account	Account	Private	User whom notification is sent to

Method

Method	Return Type	Visibility	Description
getter	Attribute type	Public	Get attribute value
setter	Void	Public	Set value of attribute

4.2.15. NotificationType

Attribute

Attribute	Type	Visibility	Description
id	int	Private	Unique identifier of notificationtype
name	String	Private	NotificationType name

Method

Method	Return Type	Visibility	Description
getter	Attribute type	Public	Get attribute value
setter	Void	Public	Set value of attribute

4.2.16. TradePost

Attribute

Attribute	Type	Visibility	Description
id	int	Private	Unique identifier of trade post
title	String	Private	Trade post title
brand	String	Private	Product's brand
model	String	Private	Product name
owner	Account	Private	Trade post owner
tradeType	String	Private	Trade type
condition	String	Private	Trade condition
price	float	Private	Products's price
priceNegotiable	float	Private	Price may be negotiable when exchange

quantity	int	Private	Product quantity
description	String	Private	Trade post description
postedDate	String	Private	Trade posted date
approvalStatus	String	Private	Trade's approval status
location	String	Private	Location for exchange
lastModified	String	Private	Trade post's last modified
rejectReason	String	Private	Reject reason from administrator

Method

Method	Return Type	Visibility	Description
getter	Attribute type	Public	Get attribute value
setter	Void	Public	Set value of attribute

4.2.17. OrderRequest

Attribute

Attribute	Type	Visibility	Description
id	int	Private	Unique identifier of order request
tradePost	TradePost	Private	Trade post that user request to order
trader	Account	Private	Member who sends order request to buyer/seller for exchange
quantity	int	Private	Number of order products
billingAddress	String	Private	Negotiable exchange location
statusID	int	Private	Order request status
rejectReason	String	Private	Reason that buyer/seller reject the exchange after accepted order request
cancelReason	String	Private	Reason that buyer/seller cancel order request
orderDate	String	Private	Order request date
stateSetDate	String	Private	The datetime which the order request is reject or canceled

Method

Method	Return Type	Visibility	Description
getter	Attribute type	Public	Get attribute value

setter	Void	Public	Set value of attribute
--------	------	--------	------------------------

4.2.18. TradePostImage

Attribute

Attribute	Type	Visibility	Description
id	int	Private	Unique identifier of trade post image
tradePost	TradePost	Private	The trade post which image belongs to
imageUrl	String	Private	Image url

Method

Method	Return Type	Visibility	Description
getter	Attribute type	Public	Get attribute value
setter	Void	Public	Set value of attribute

4.2.19. TradeReport

Attribute

Attribute	Type	Visibility	Description
id	int	Private	Unique identifier of trade report
tradePost	TradePost	Private	The trade post which trade report belongs to
reason	String	Private	Reason of report

Method

Method	Return Type	Visibility	Description
getter	Attribute type	Public	Get attribute value
setter	Void	Public	Set value of attribute

4.2.20. TradeRating

Attribute

Attribute	Type	Visibility	Description
-----------	------	------------	-------------

id	int	Private	Unique identifier of trade rating
fromUser	Account	Private	User who does the evaluation
toUser	Account	Private	User who receives the rating
feedbackType	String	Private	Represents for type of rating, a member evaluate the exchange for trade post owner or trade post owner evaluates the traders, for calculating the reputation
tradePost	TradePost	Private	Trade post that is evaluated
rating	int	Private	Number of stars rating
comment	String	Private	Feedback about exchange

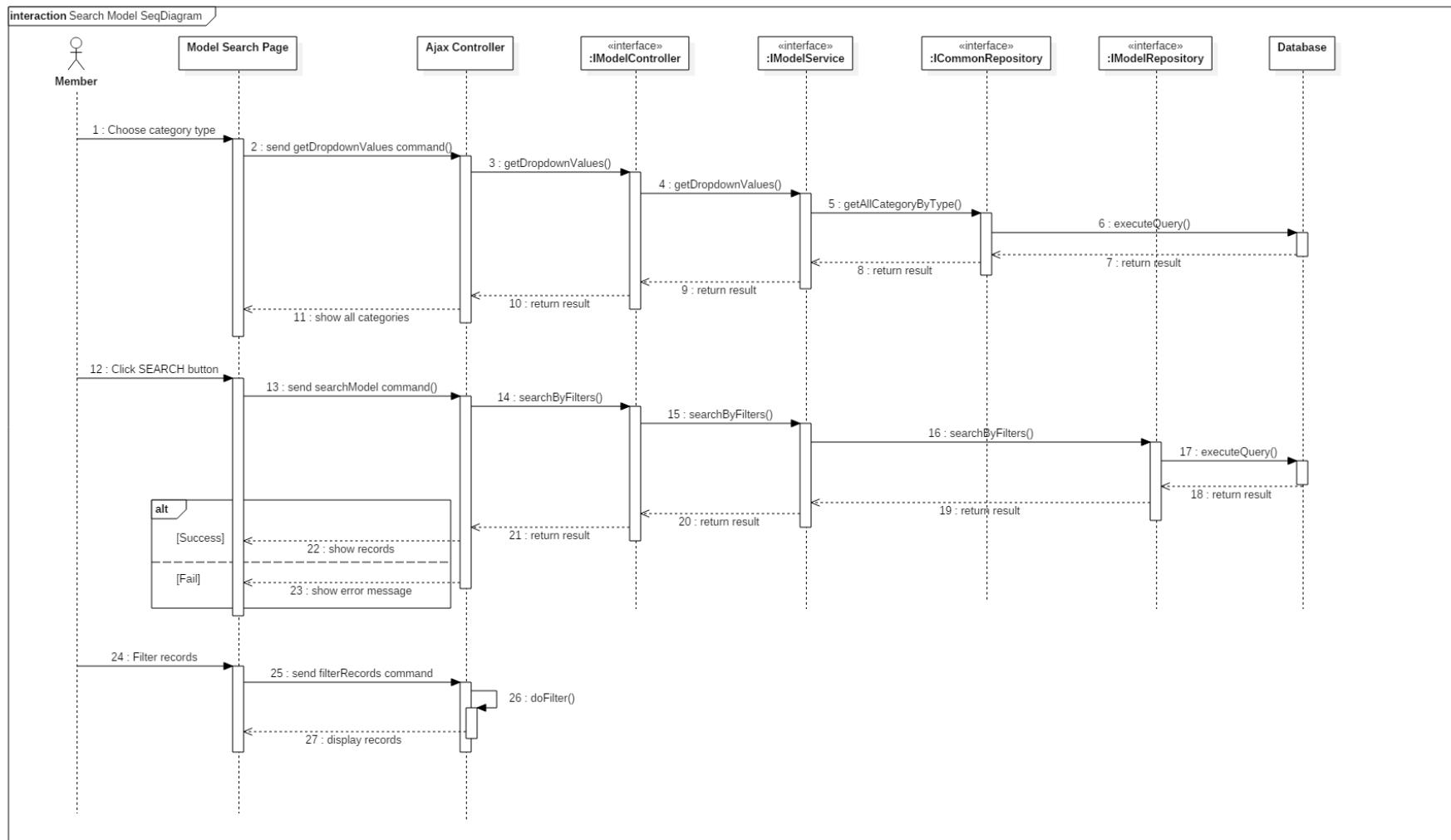
Method

Method	Return Type	Visibility	Description
getter	Attribute type	Public	Get attribute value
setter	Void	Public	Set value of attribute

4.3. Interaction Diagram

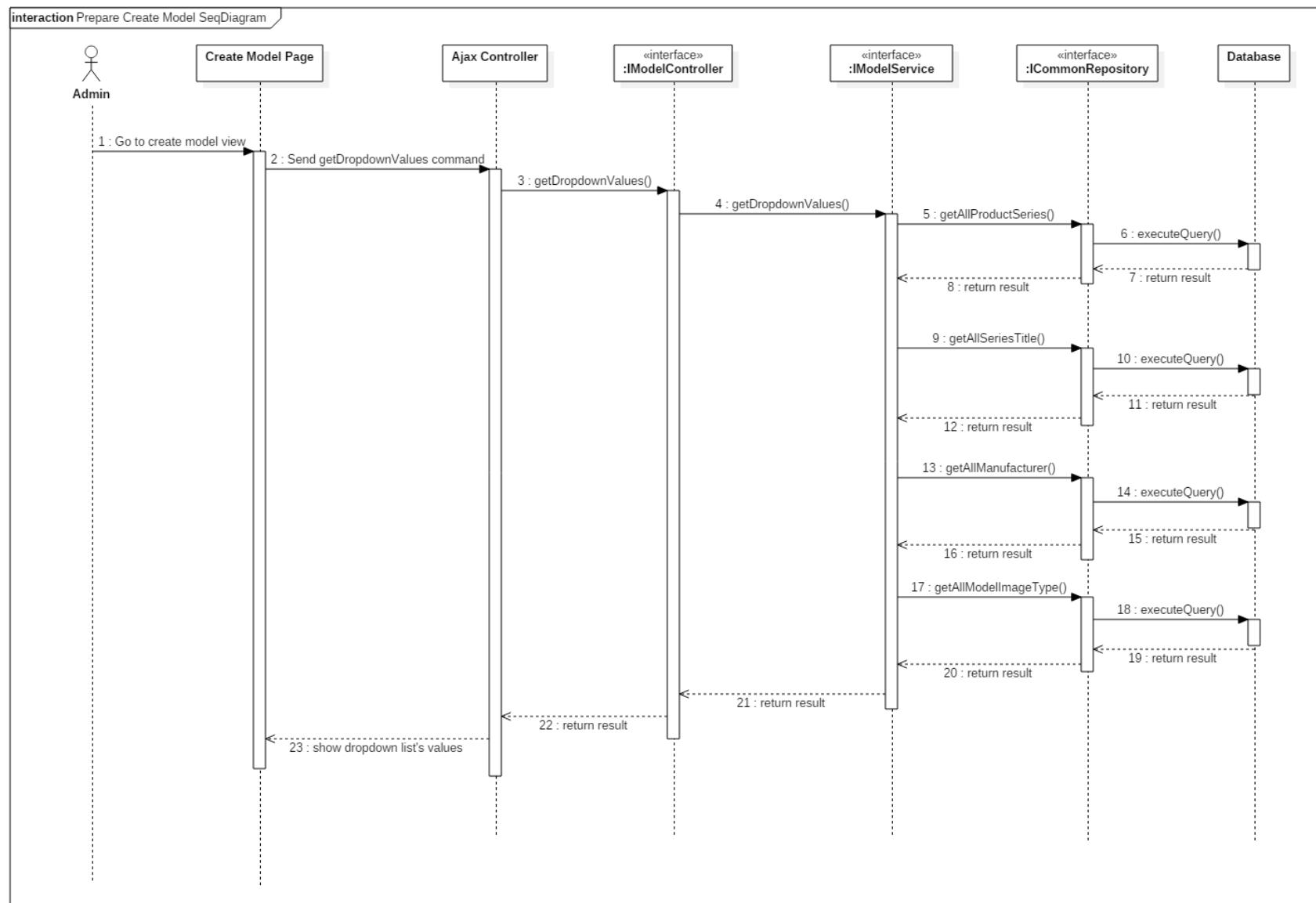
4.3.1. Sequence Diagram

4.3.1.1. Search model

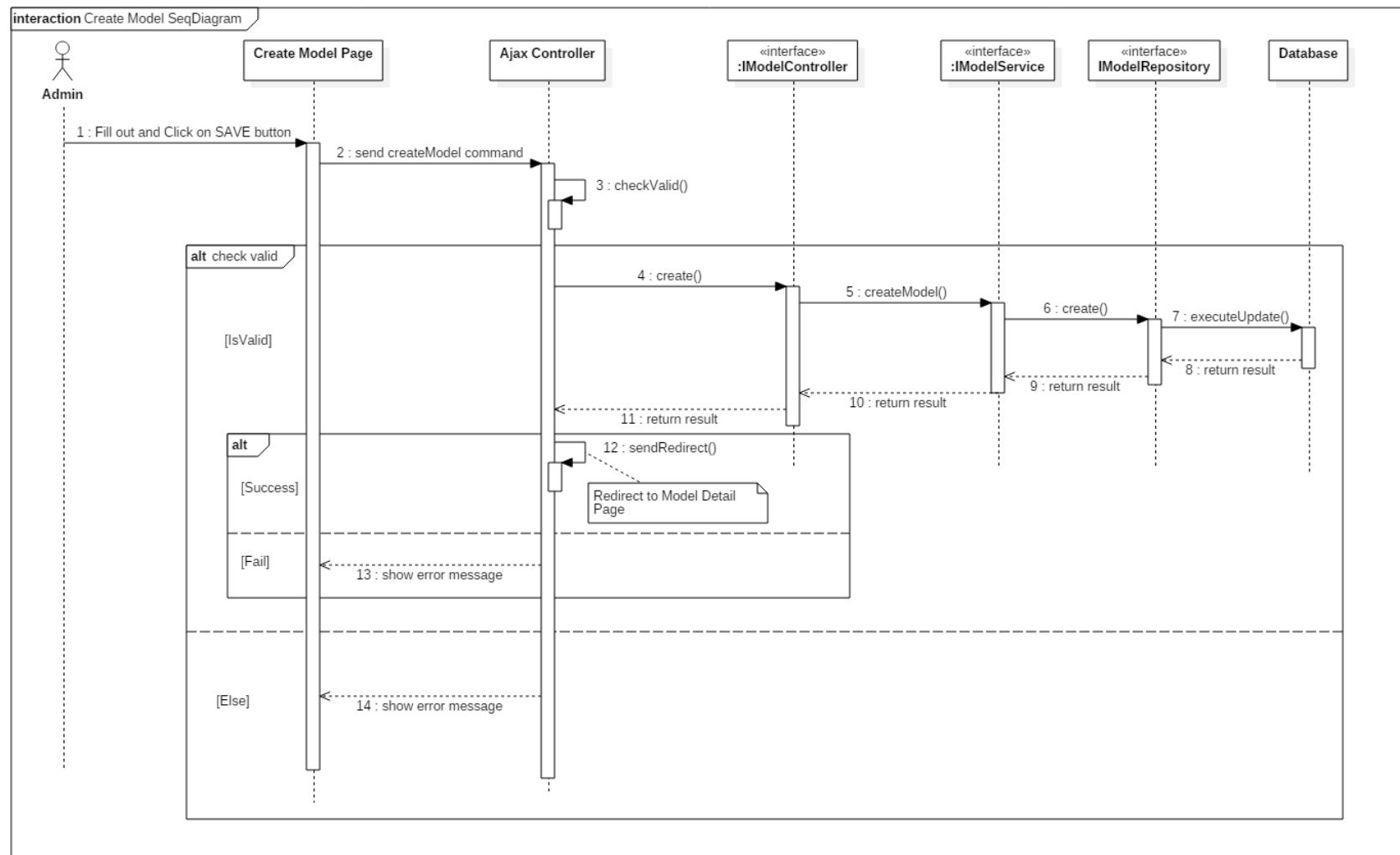


4.3.1.2. Create model

4.3.1.2.1. Prepare Create Model

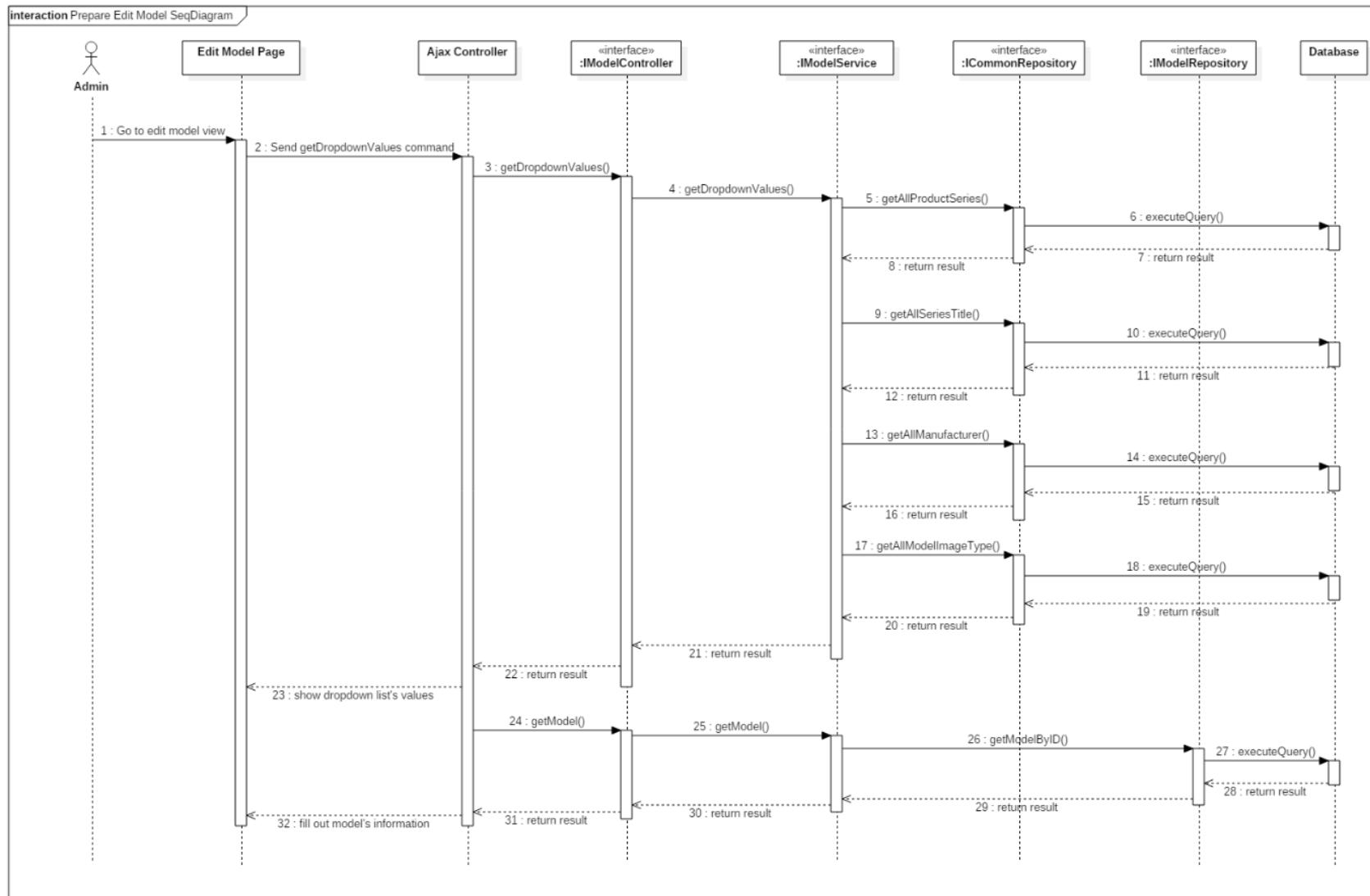


4.3.1.2.2. Create Model

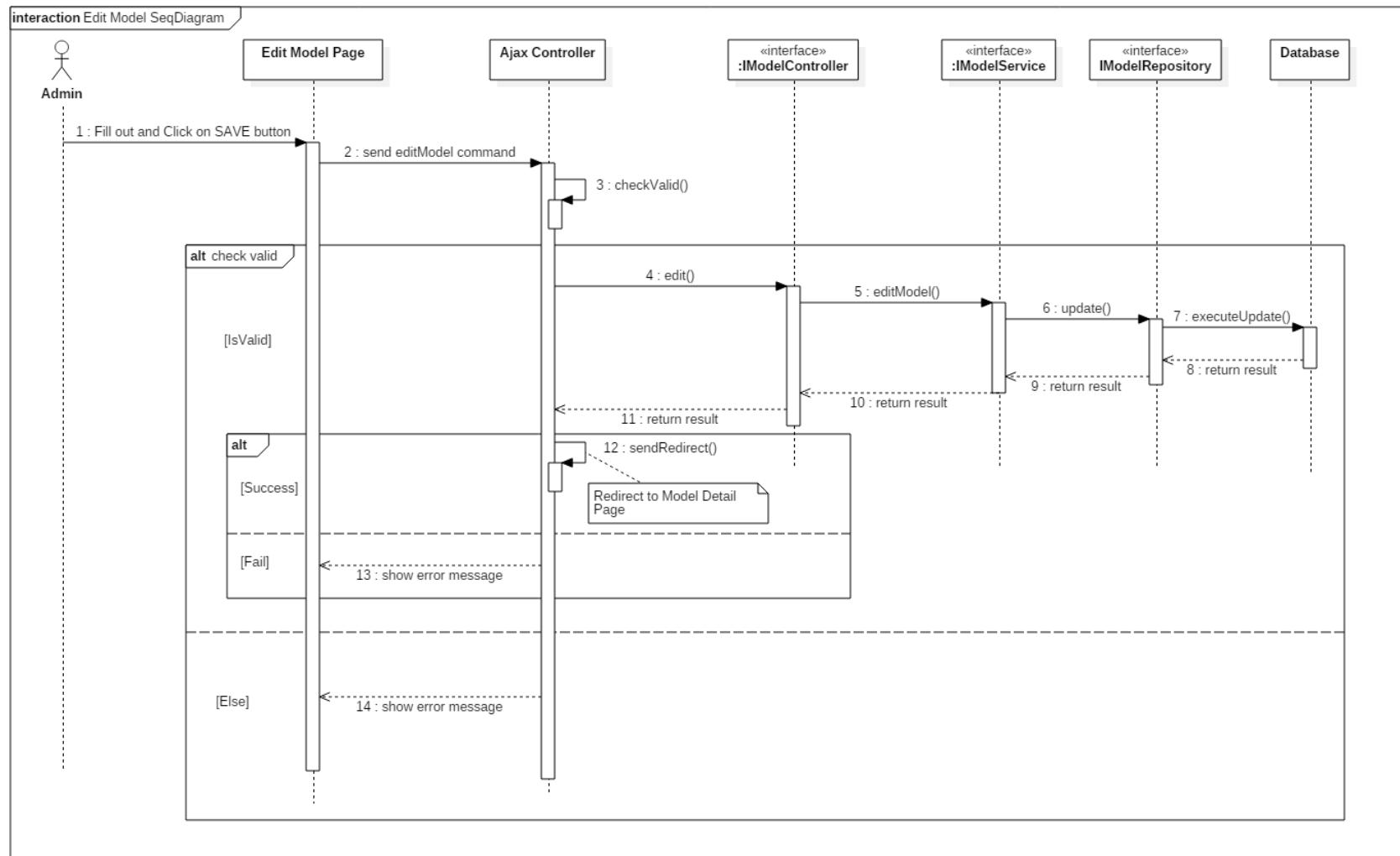


4.3.1.3. Edit model

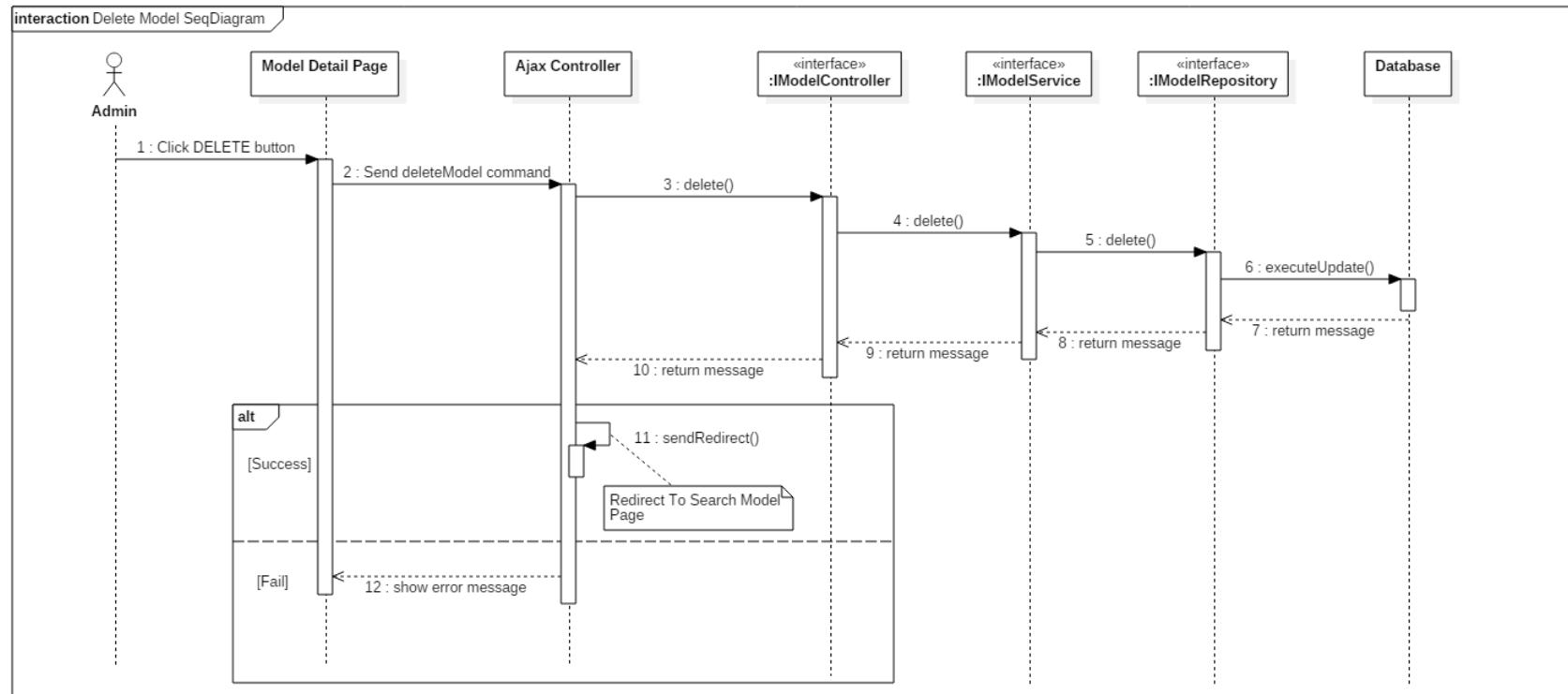
4.3.1.3.1. Prepare Edit Model



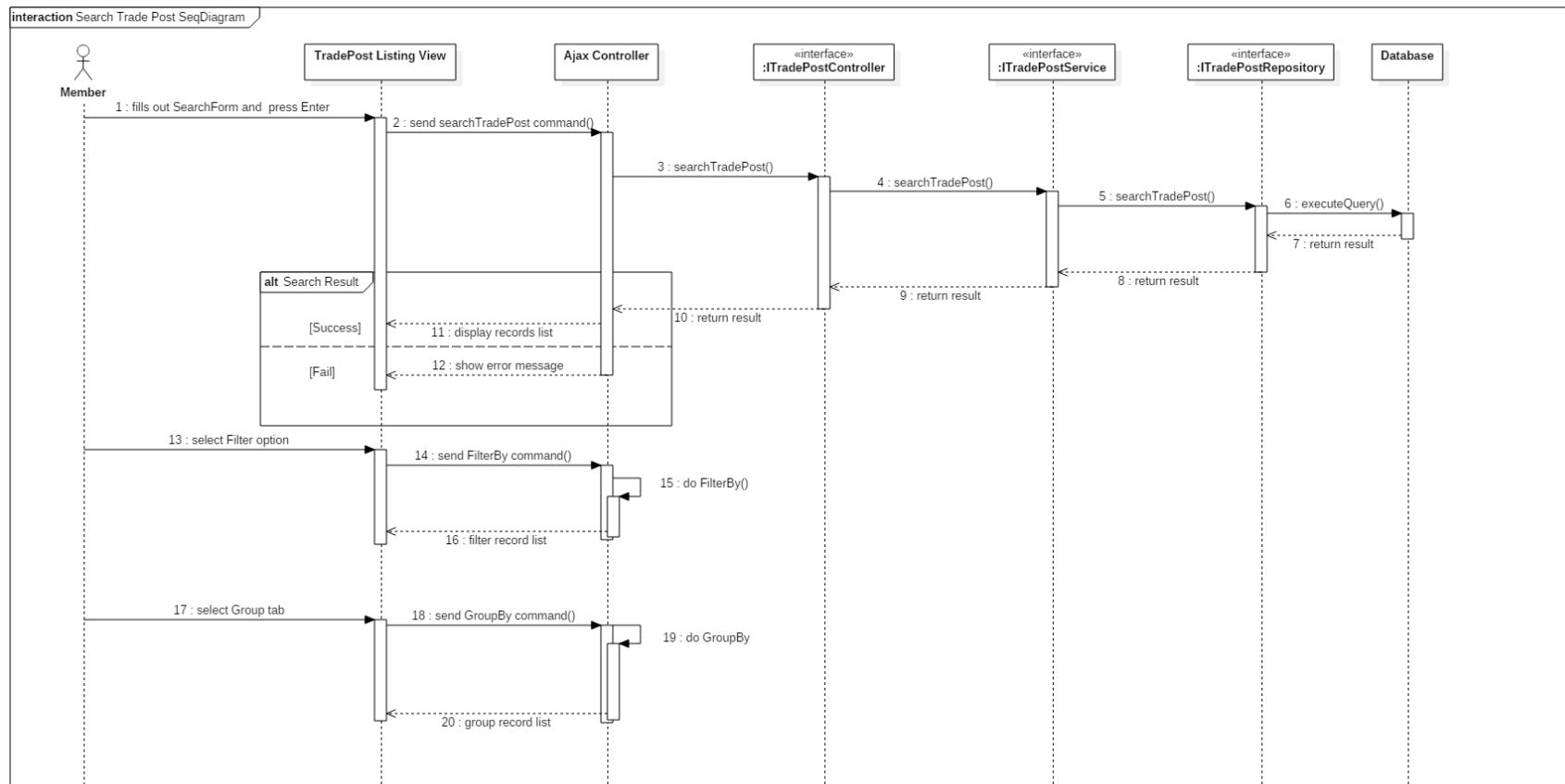
4.3.1.3.2. Edit Model



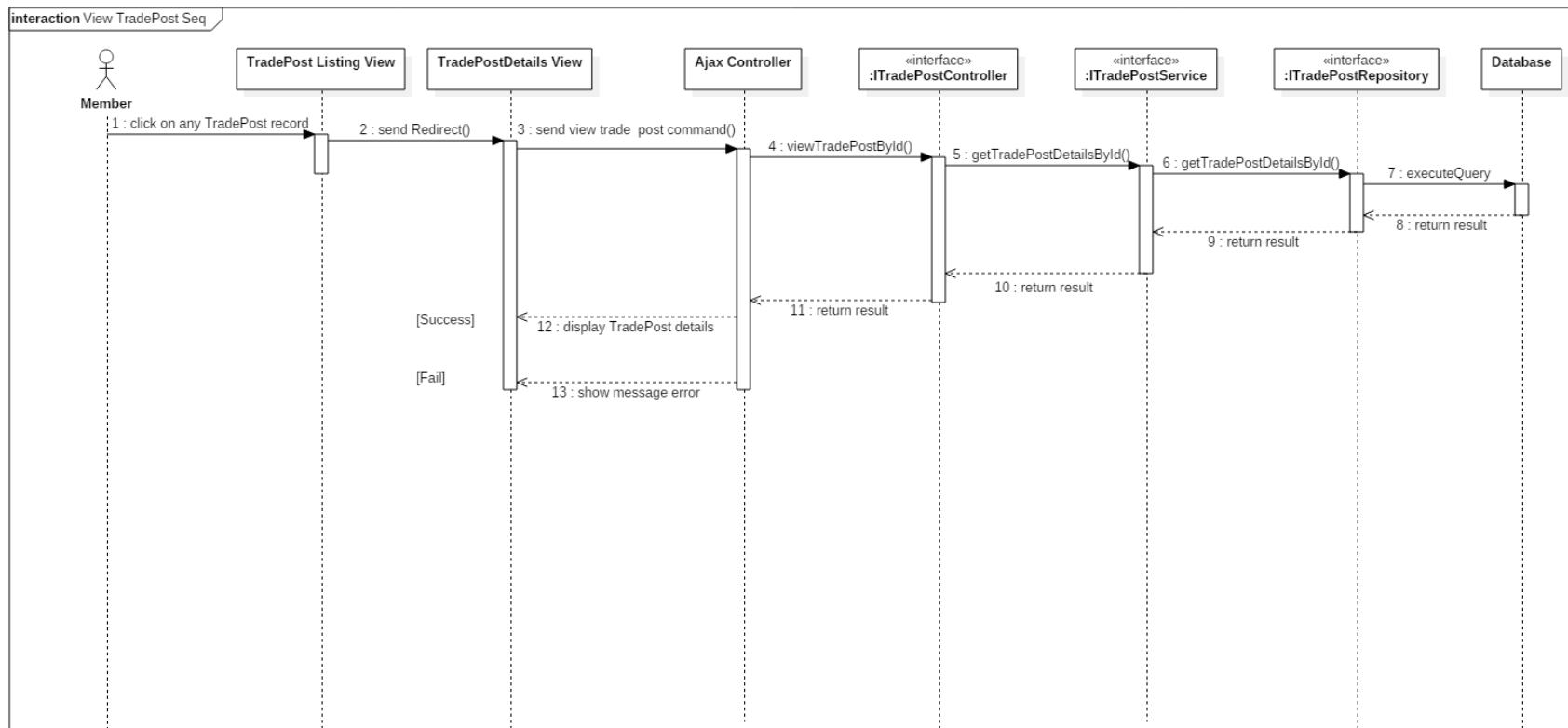
4.3.1.4. Delete model



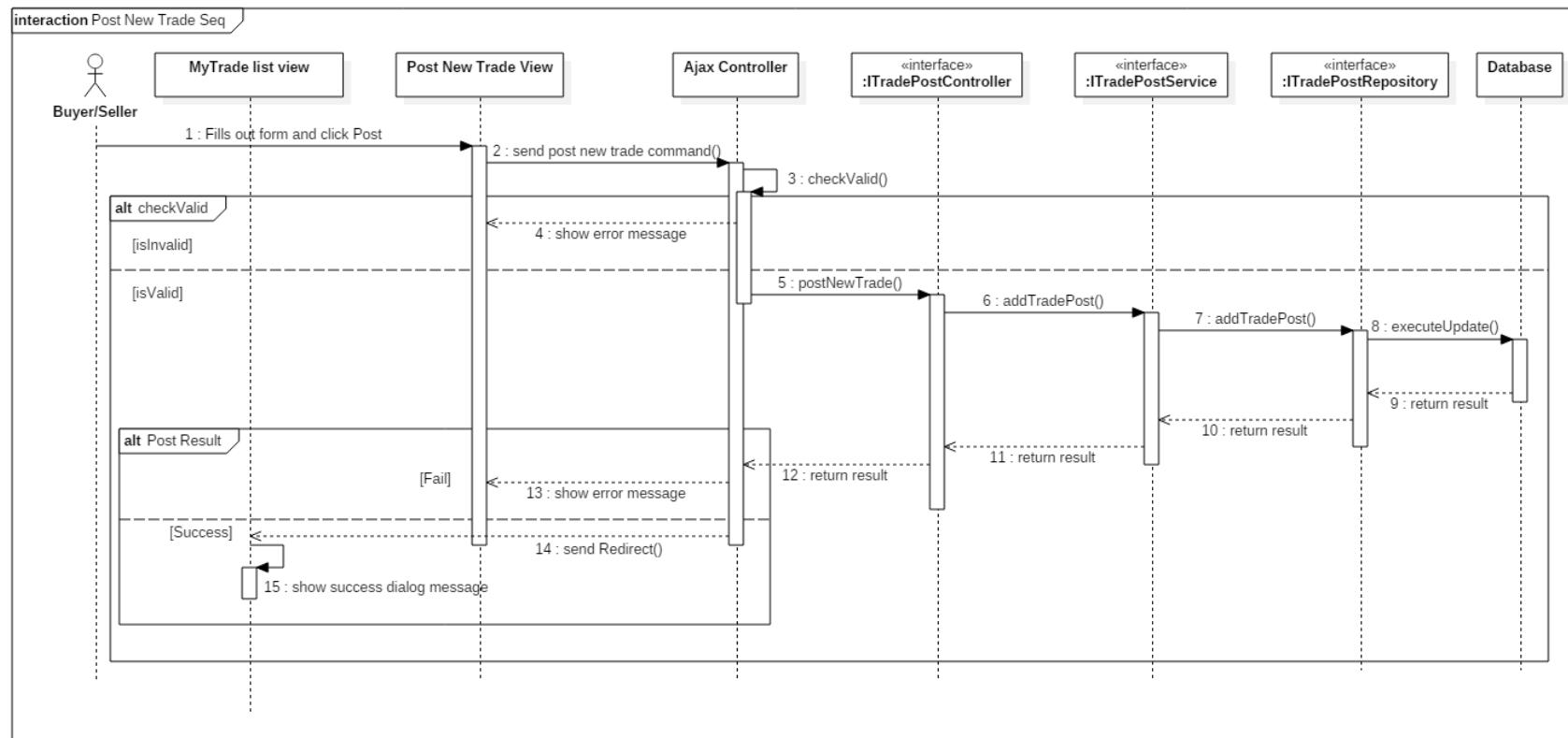
4.3.1.5. Search Trade Post



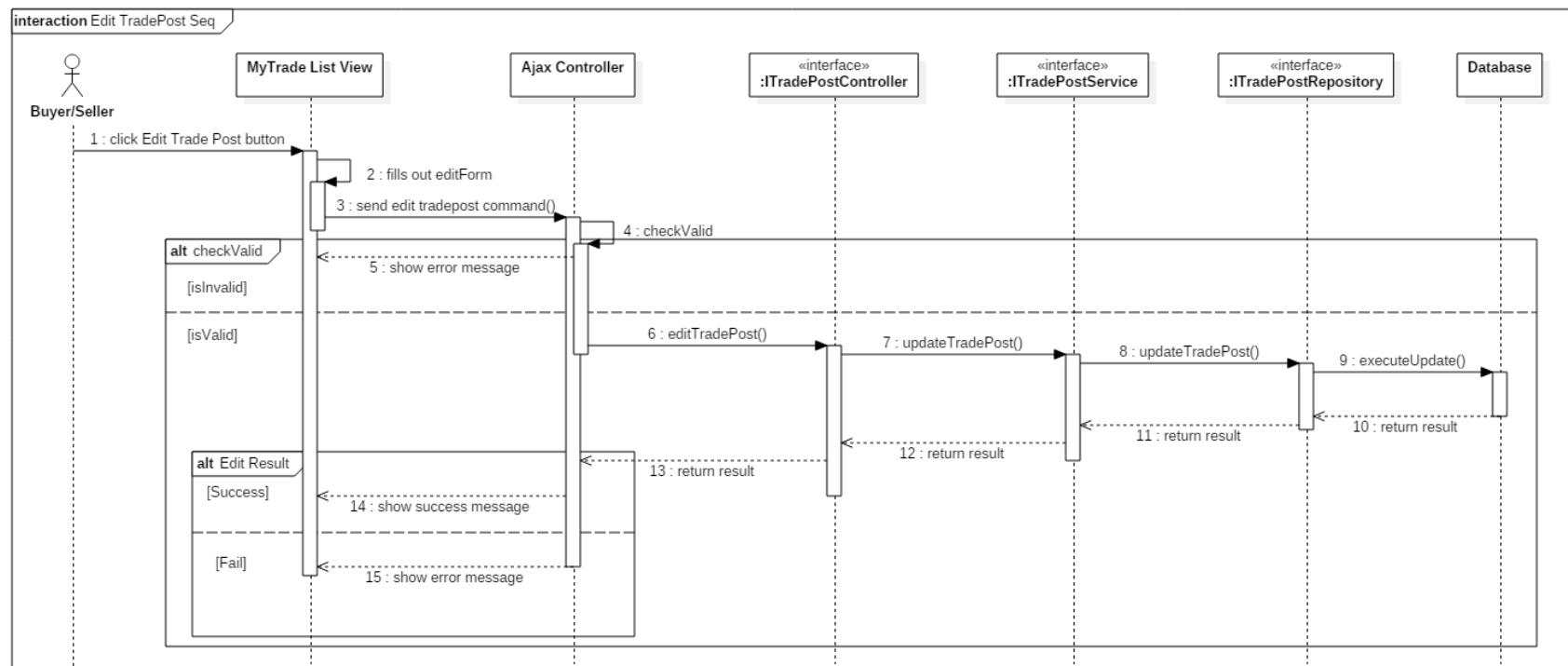
4.3.1.6. Get Trade Post Detail Information



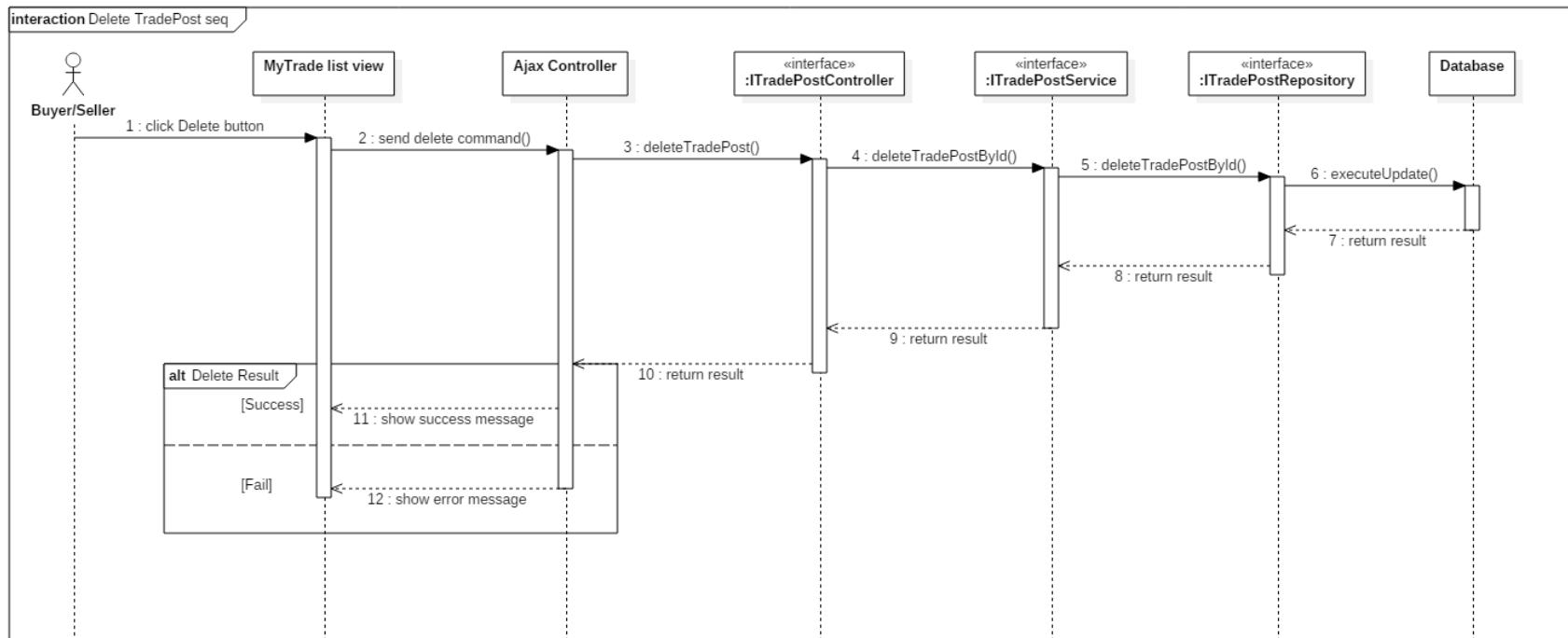
4.3.1.7. Post new trade



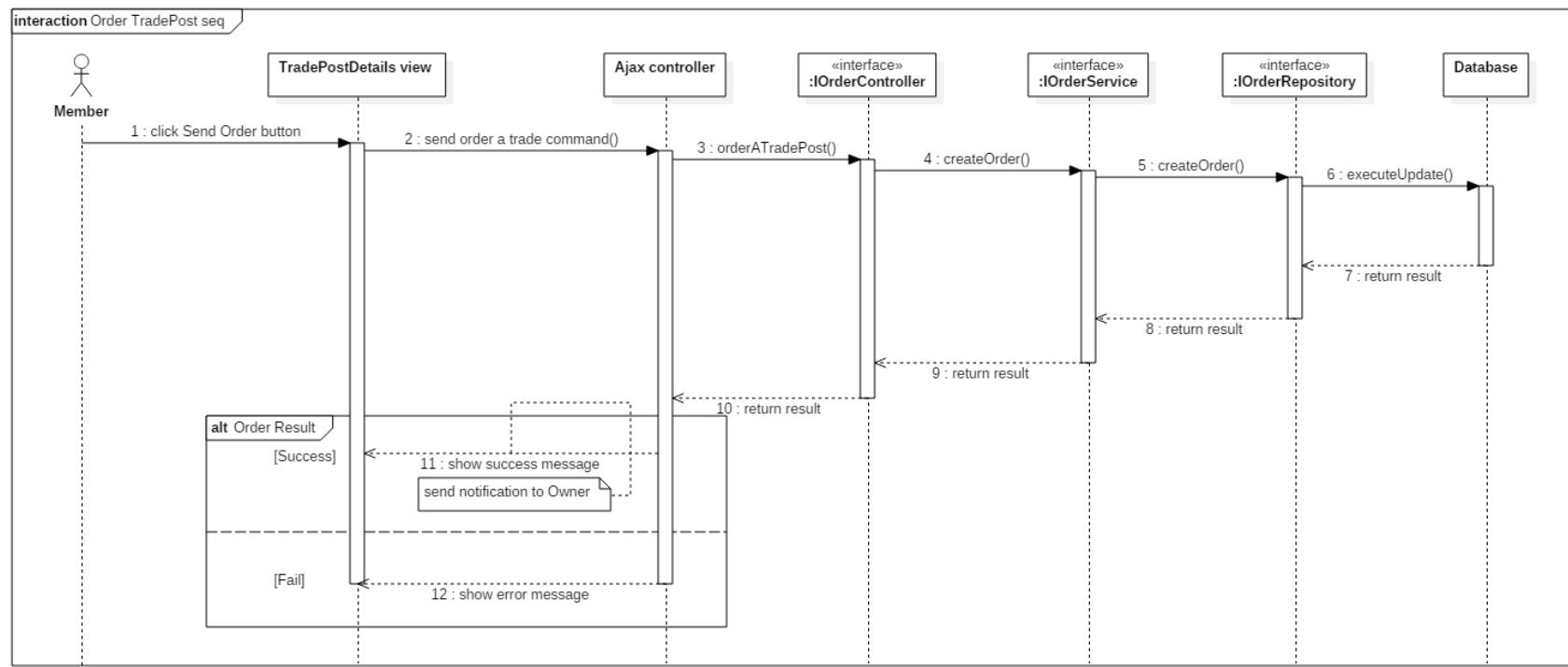
4.3.1.8. Edit trade post



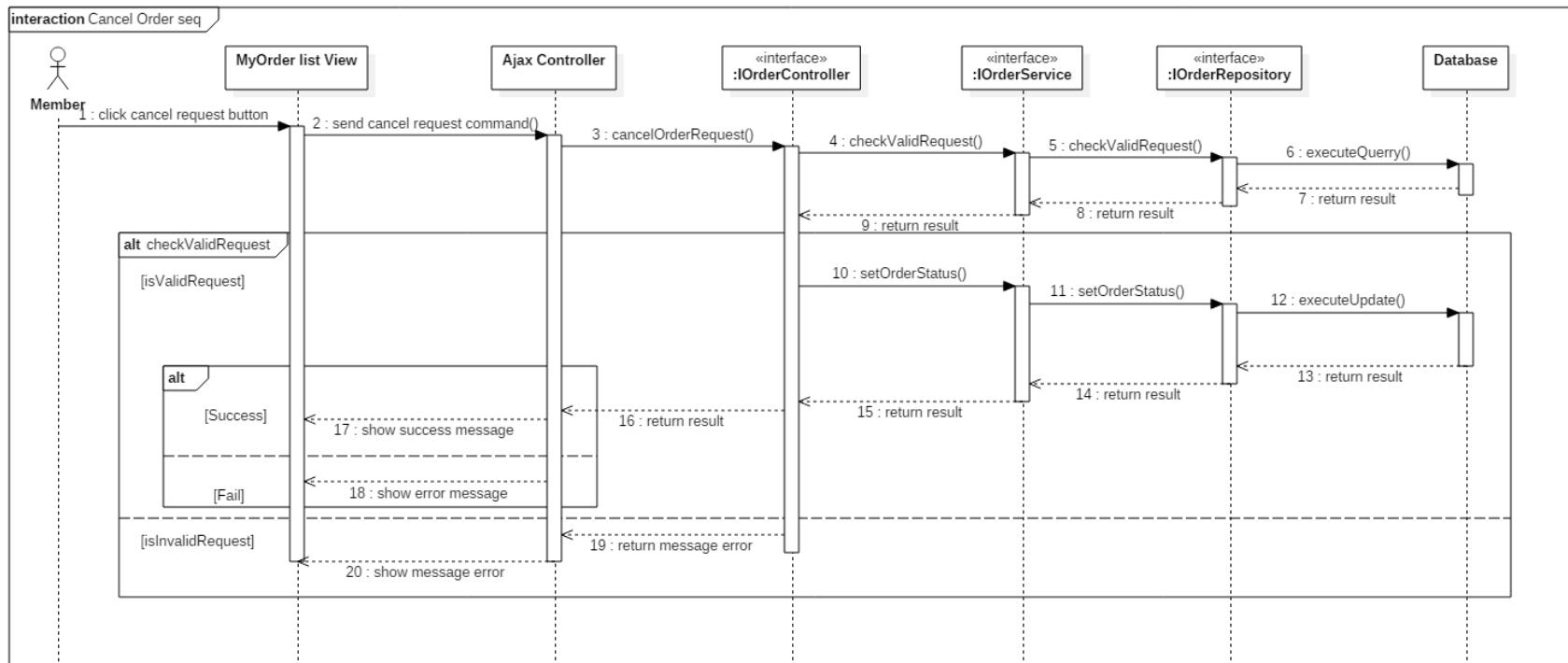
4.3.1.9. Delete trade post



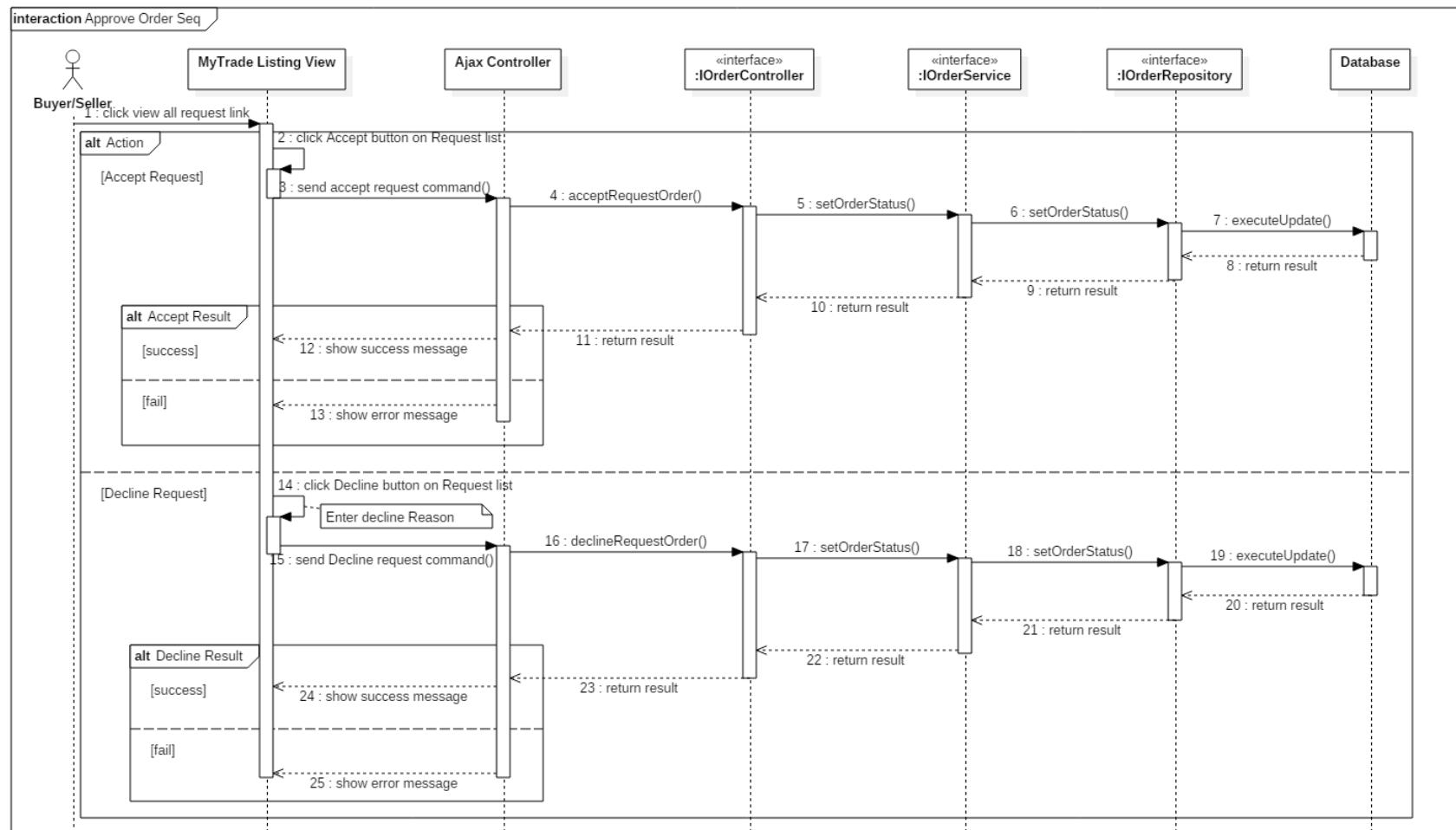
4.3.1.10. Request order to a trade post



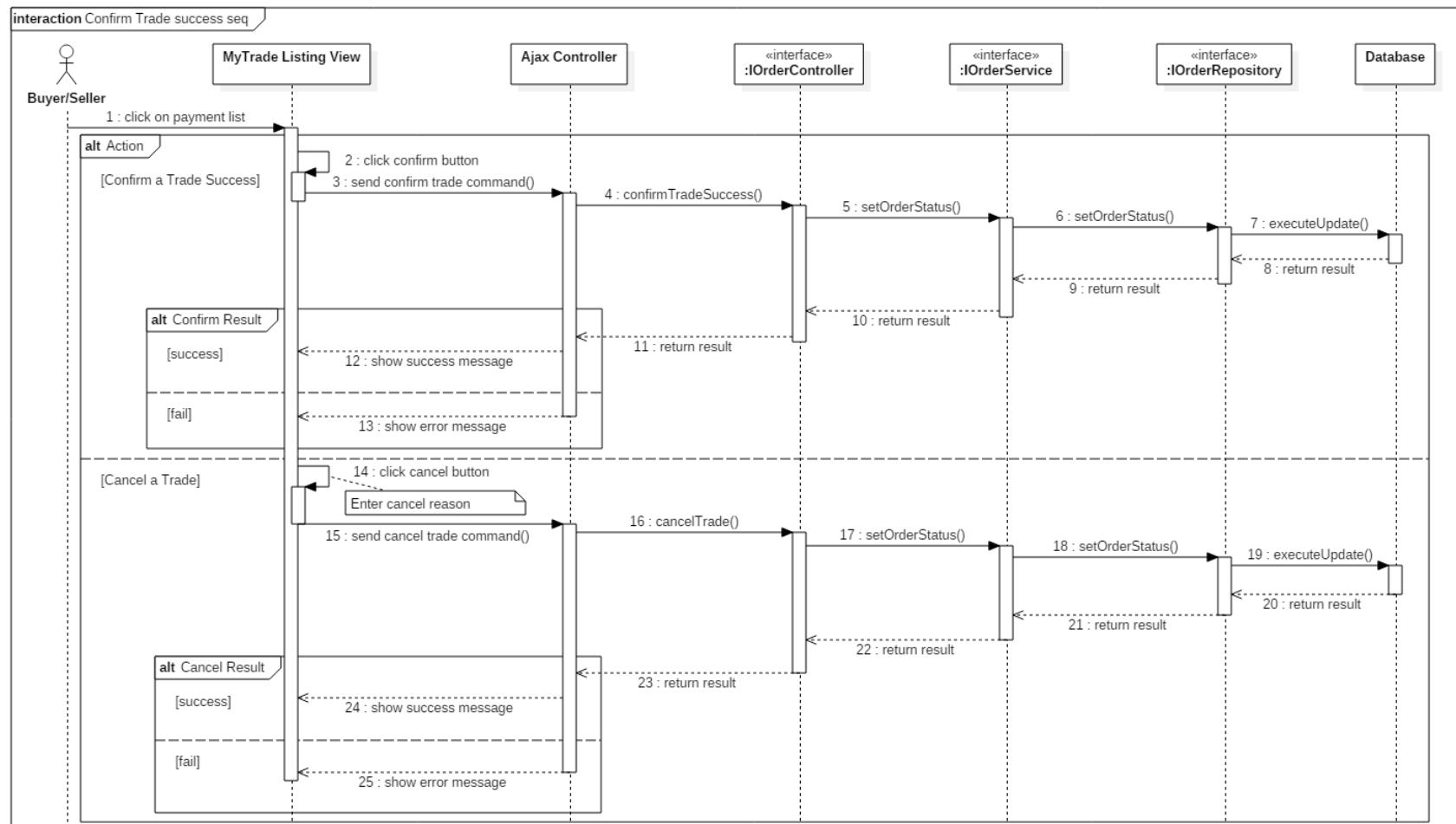
4.3.11. Cancel an order



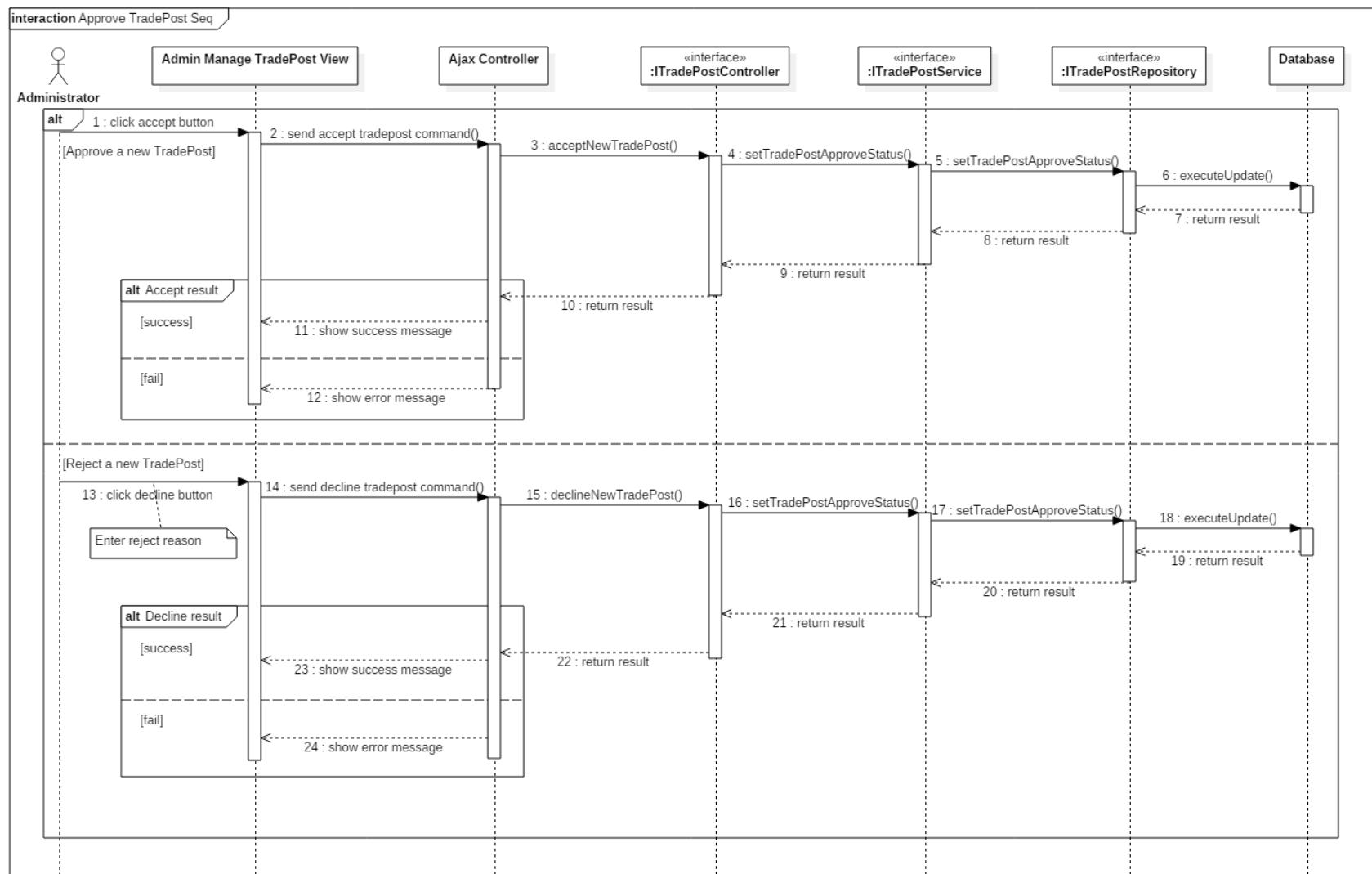
4.3.1.12. Approve order request



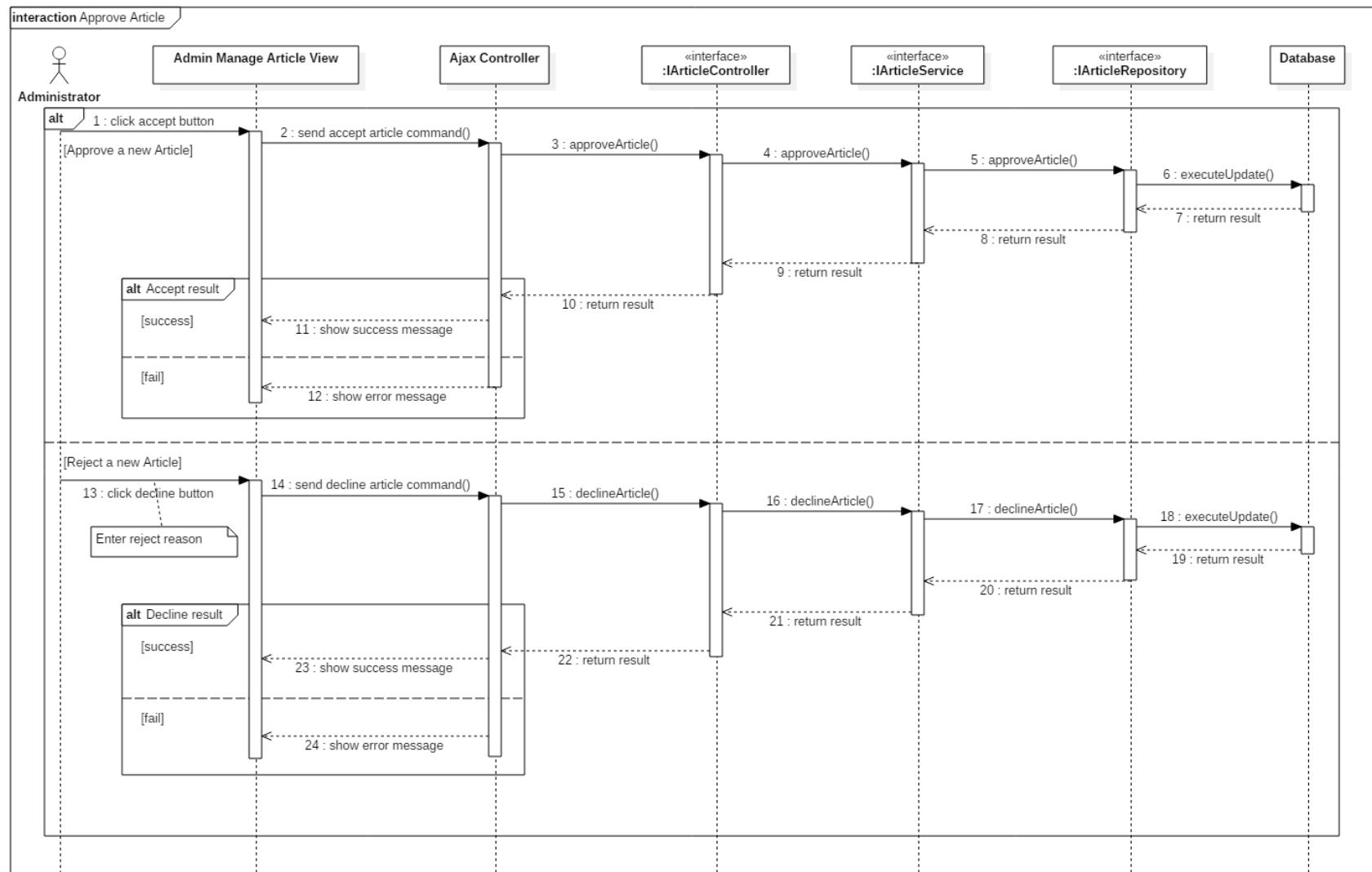
4.3.1.13. Confirm trade success



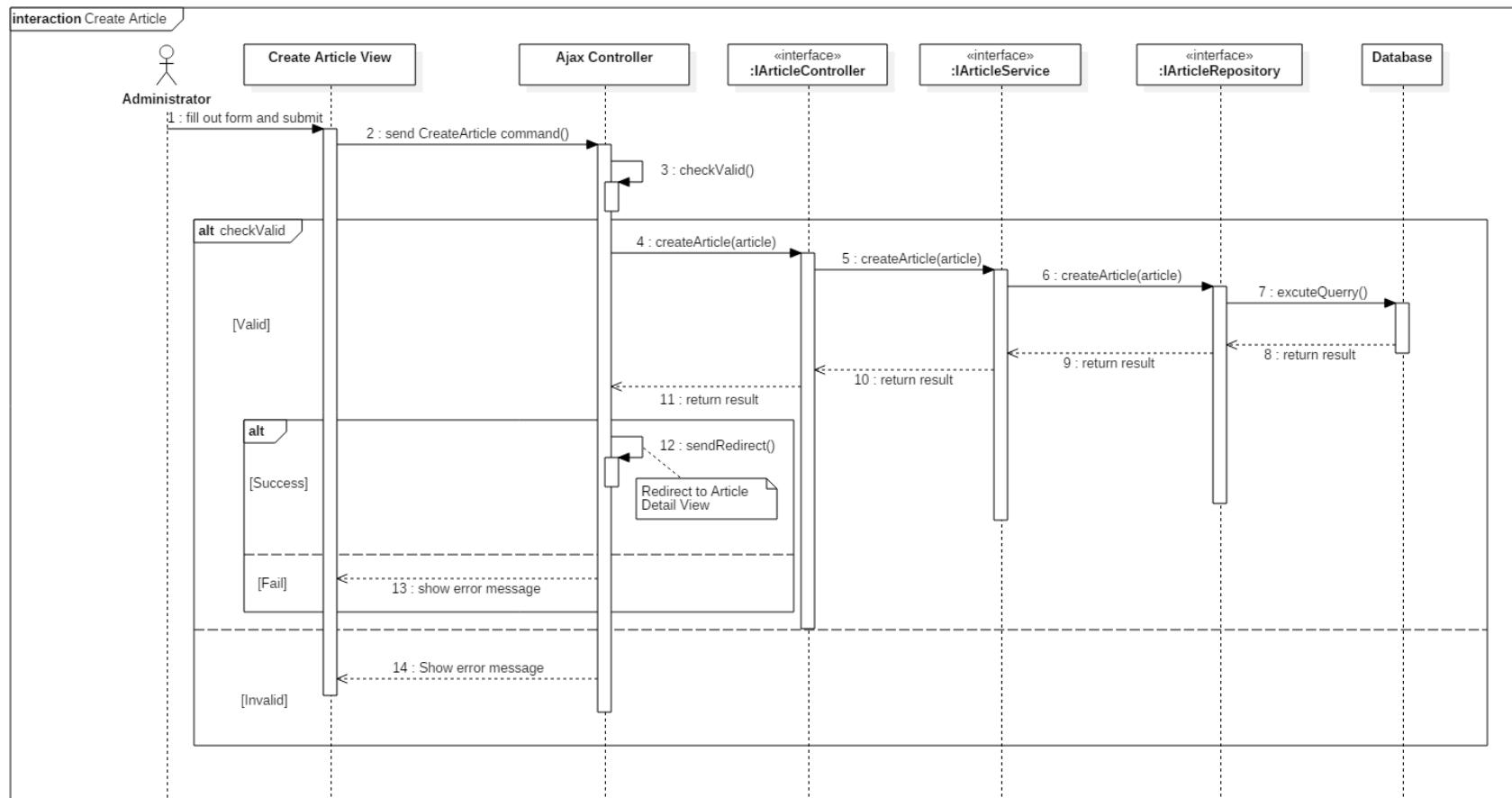
4.3.1.14. Approve a trade post from buyer/seller



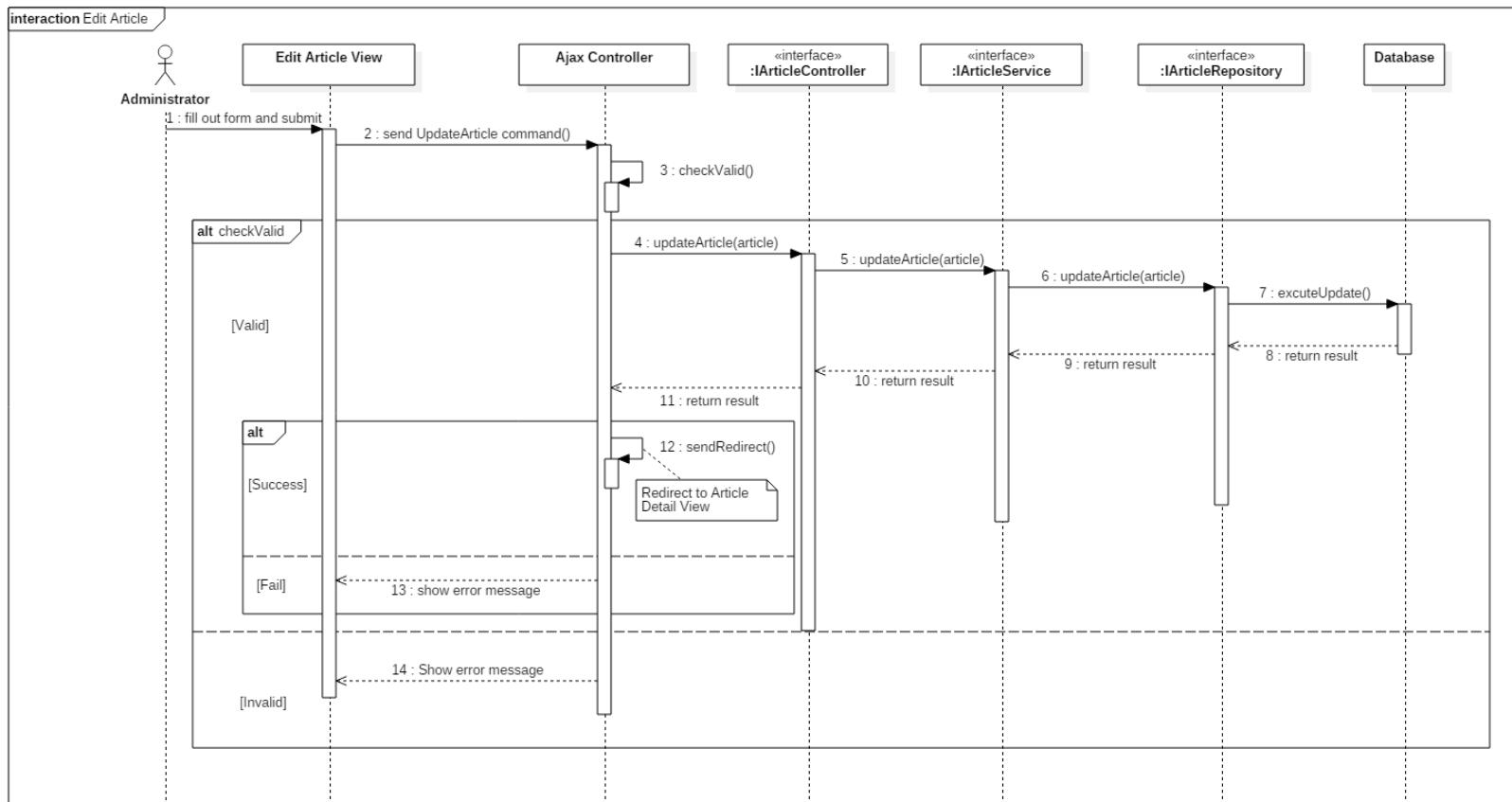
4.3.1.15. Administrator approve member's articles



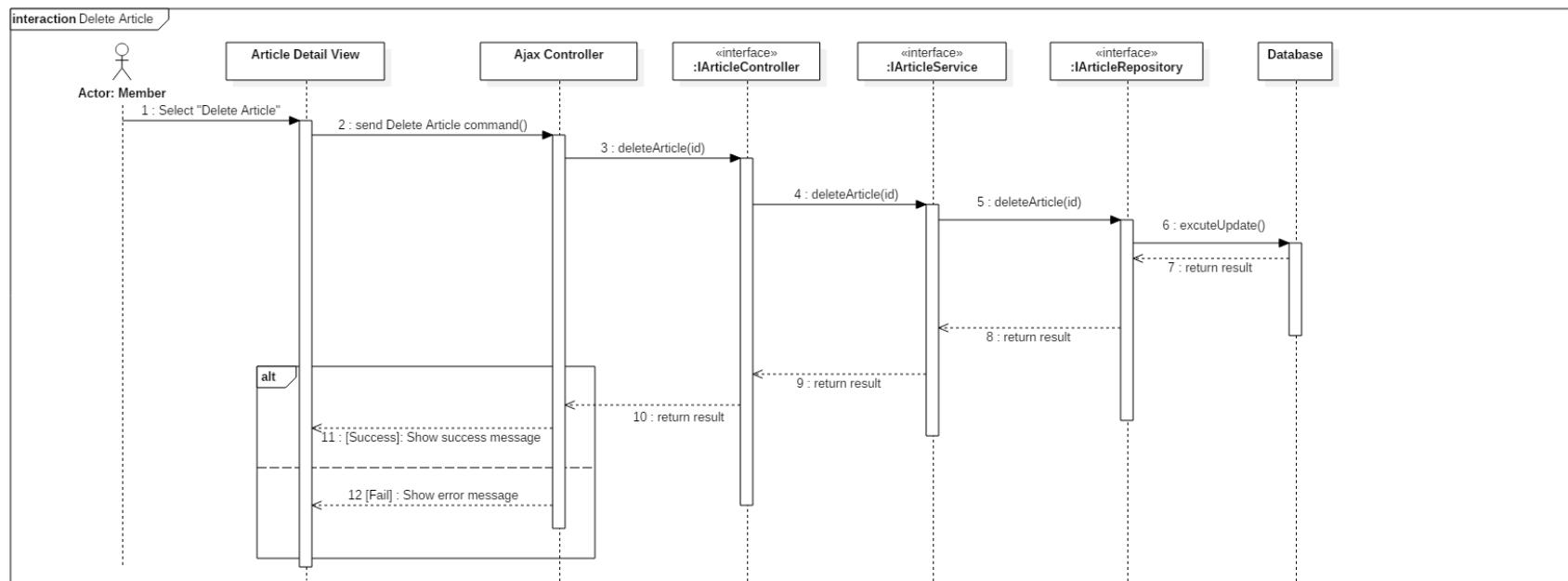
4.3.1.16. Create new article



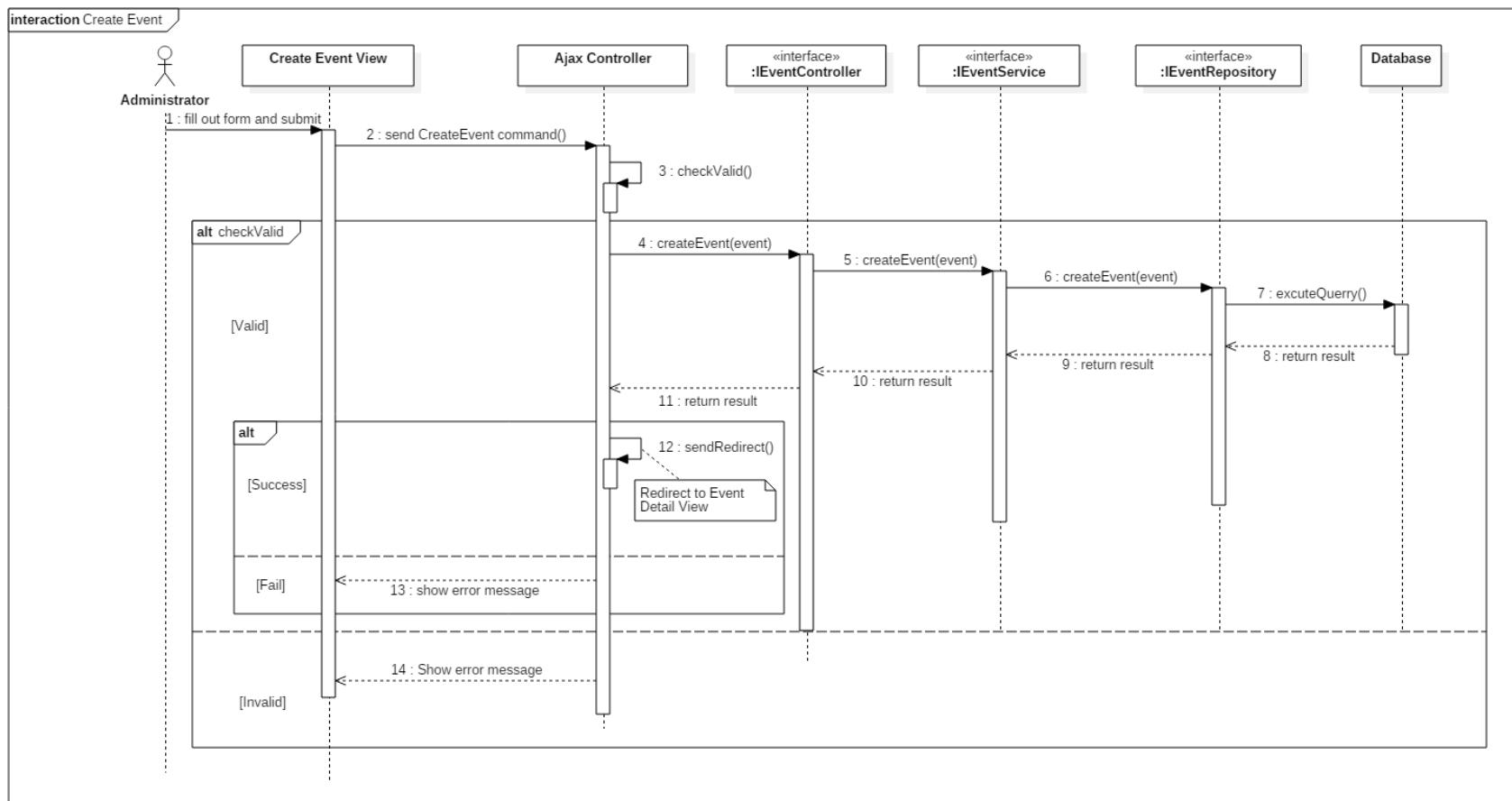
4.3.1.17. Edit article



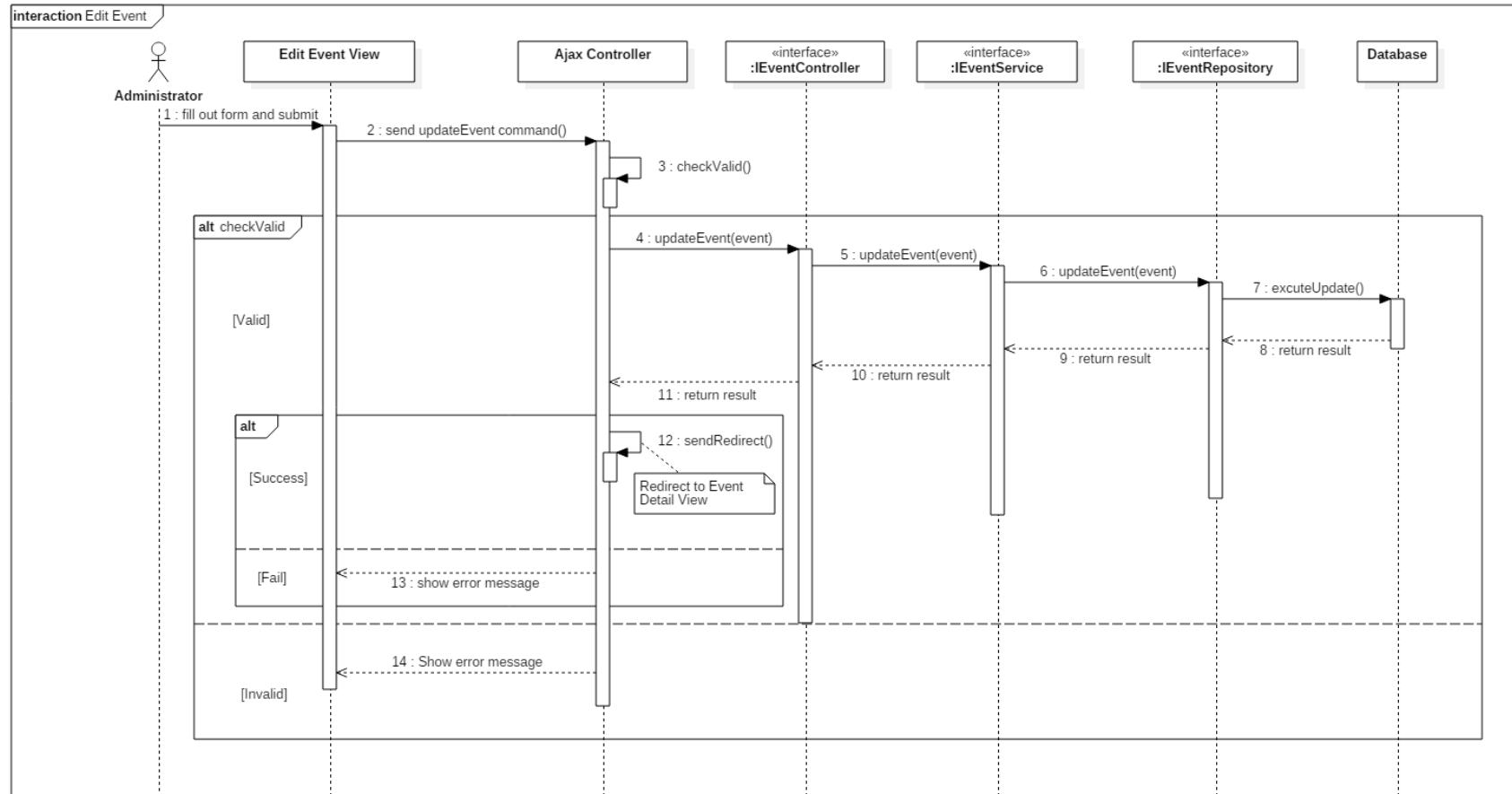
4.3.1.18. Delete article



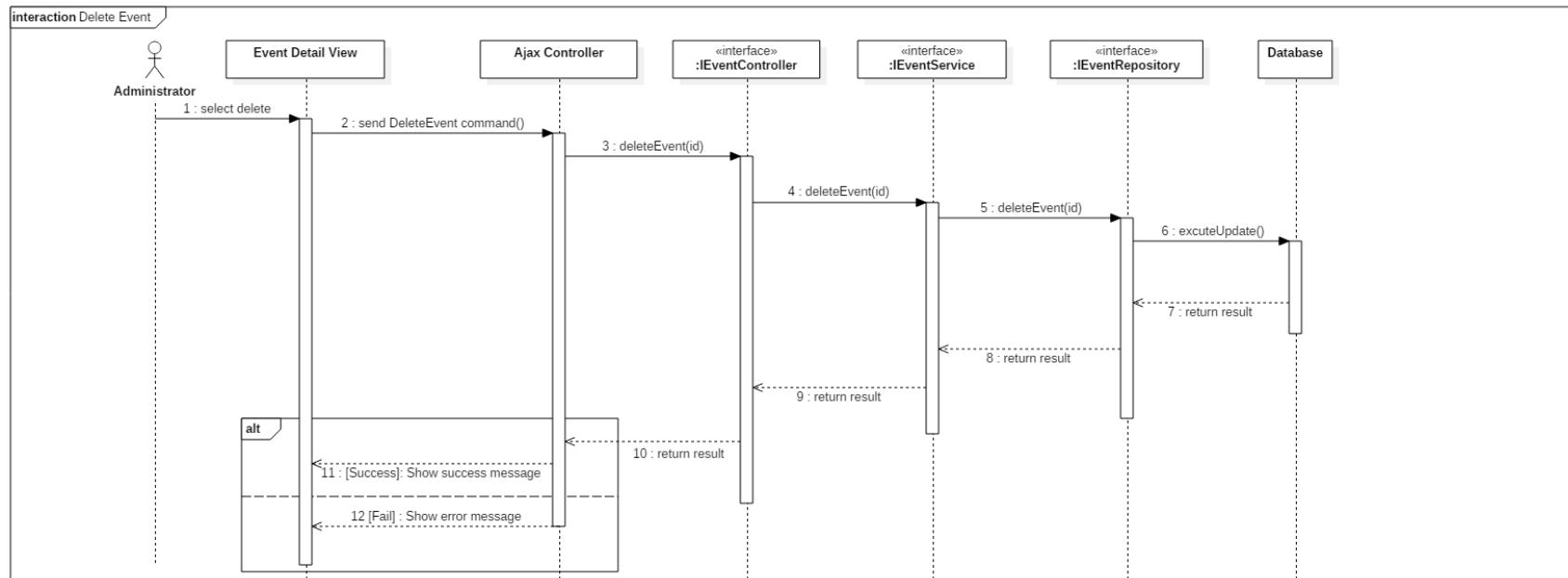
4.3.1.19. Create event



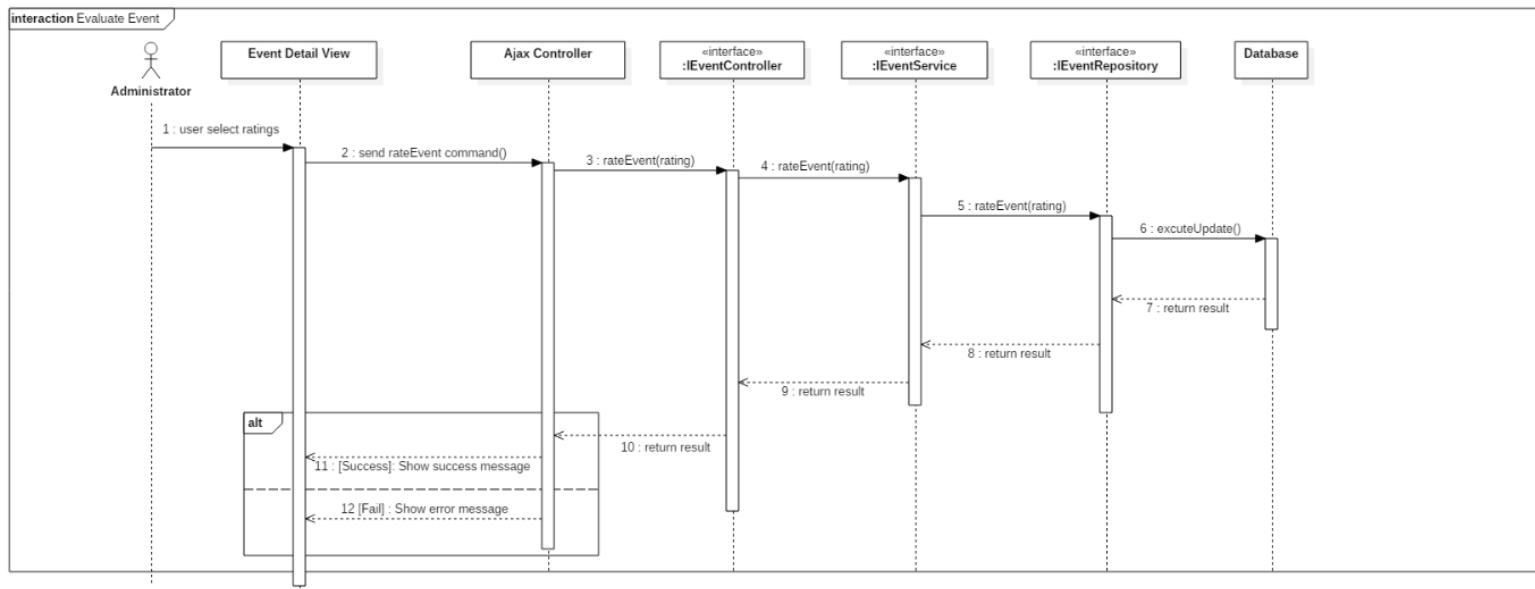
4.3.1.20. Edit event



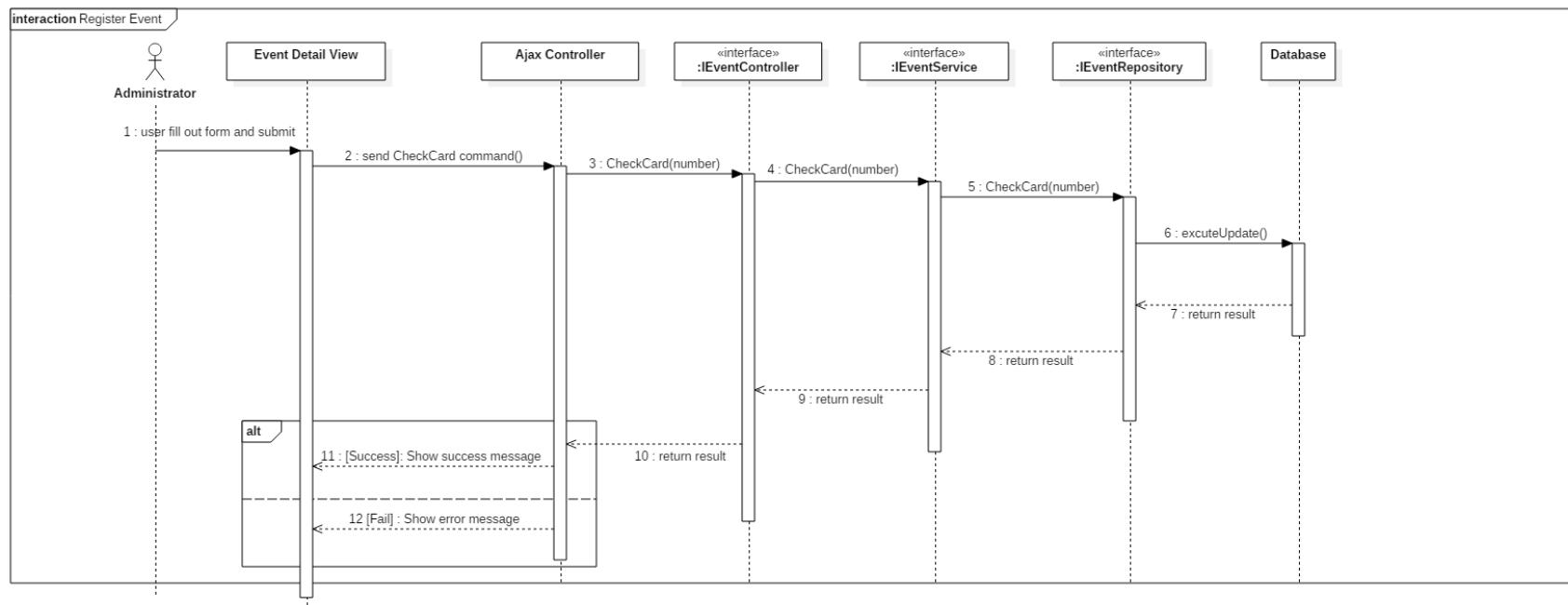
4.3.1.21. Delete event



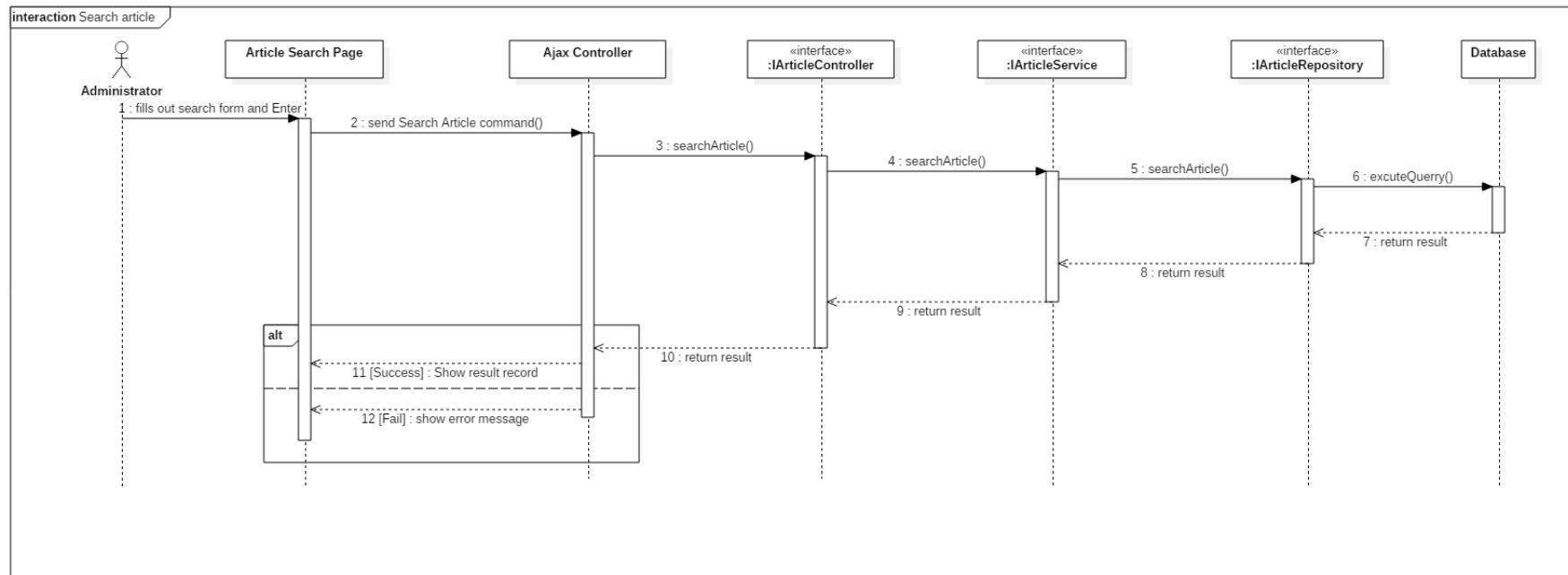
4.3.1.22. Evaluate Event



4.3.1.23. Register event



4.3.1.24. Search article



5. Interface

5.1. Member

5.1.1. Web application

5.1.1.1. Display search model view

The screenshot shows a web-based search interface for a product catalog. On the left, there's a sidebar with two sections: 'CATEGORY OPTIONS' and 'Series Title'. The 'CATEGORY OPTIONS' section contains a 'Product Series' dropdown with items like Master Grade (25), Perfect Grade (20), Real Grade (89), High Grade (50), Other Grade (200), SD Gundam (30), and Premium Web Exclusives. The 'Series Title' section lists items such as First Gundam (12), Mobile Suit Variation MSV (20), 0080 War in the Pocket (4), and 0083 Stardust Memory (20). The main content area has a search bar labeled 'Enter a search term' with a magnifying glass icon. Below the search bar is a 'Search Model Information' section with several filter fields: 'Category Type' (dropdown, value 3), 'Status' (dropdown, value 4), 'Price' (dropdown, value 5), 'Category' (dropdown, value 6), 'Manufacturer' (dropdown, value 7), 'Updated Date' (dropdown, value 8), and a 'SEARCH' button (value 9) next to a 'CLEAR' button (value 10). To the right of the search form is a scroll-up arrow (value 11). Below the search form is an 'Order by' section with 'Price' (dropdown, value 12) and 'Ascending' (dropdown, value 13), followed by a pagination bar with buttons for 'First', 'Prev', '1', '2', '3', '4', '5', '...', 'Next', and 'Last' (value 14). At the bottom, there are three cards for 'Gundam Mouts': one card on the left with details (Price: 900\$, Released Date: 2/12/1996, Manufacturer: Bandai, Product Series: Perfect Grade) and two cards on the right showing images of the models (value 15, 16, 17).

Fields

No	Field Name	Description	Read only	Mandatory	Control Type	Data Type	Length
1	Search input text	Input search keyword, search by model name	No	No	Text input	String	N/A
3	CategoryType	Choose category types filter	No	No	Dropdown list	String	N/A
4	Status	Choose status filter	No	No	Dropdown list	String	N/A
5	Price	Choose price filter	No	No	Dropdown list	Float	N/A

6	Category	This field is loaded follow with user's option in category type dropdown list, to choose the specified category	No	No	Dropdown list	String	N/A
7	Manufacturer	Choose manufacturer filter	No	No	Dropdown list	String	N/A
8	Updated Date	Choose updated date time filter	No	No	Dropdown list	String	N/A
12	Order By options	Choose order by options	No	No	Dropdown list	String	N/A
13	Ascending or Descending options	This dropdown list follows with order by dropdown list	No	No	Dropdwon list	String	N/A
15	Model name	Display model name	Yes	Yes	Label	String	N/A
16	Search record	Search record	Yes	Yes	N/A	N/A	N/A
17	Model detail info	Display model detail info when hover on a record	Yes	Yes	N/A	String	N/A

Buttons/Hyperlinks

No	Function	Description	Validation	Outcome
2	Search	Send search command to server with value in search input text	N/A	List records match with the input keyword
9	Search with filters in search model information form	Send search command to server with values in search input text and all filters's values in search model	N/A	List records match with the input keyword and the filters

		information form		
10	Reset all field in search model information form	All dropdown list values will set to option “All”	N/A	All dropdown list values are set to option “All”
11	Open/close search information form	Open or close the search filters form	N/A	The search filters form will be pulled up/pushed down when click
14	Pagination	For paging with first, next, page numbers, next and last	N/A	The list records is updated match with page numbers
18	Redirect to the selected model's detail page	Redirect to the selected model's detail page	N/A	The selected model's detail page is shown

5.1.1.2. Display model detail

The screenshot shows a detailed product page for a mobile suit model. At the top, there is a navigation bar with links to Home, Pages, Mega Menu, GUNDAM, Articles, Events, Trade Market, and Contact. Below the navigation is a sidebar with categories like Product Series, Category Options, and Series Title.

Product Series:

- Master Grade (25)
- Perfect Grade (20)
- Real Grade (89)
- High Grade (50)
- Other Grade (200)
- SD Gundam (30)
- Premium Web Exclusives

Series Title:

- First Gundam(12)
- Mobile Suit Variation MSV(20)
- 0080 War in the Pocket

MODEL INFORMATION

Model: AMX-103 Hamuna-Hamuna
Manufacturer: Bandai
Scale: Master Grade, 1/100
Series: MG Other Gundam
Release Date: Mid May, 2018
Sales Price: 3,600yen
Item Code: 257516
Rating: ★★★★☆ 2

IMAGES

Package	Item picture

Other picture Contents

A vertical sidebar on the right side of the page displays thumbnail images of other model packages, labeled "Model".

The screenshot shows a website interface for a Gundam model. On the left is a sidebar with links: G Gundam, Gundam W, Gundam X, Turn A Gundam (which is highlighted), Gundam 00, Gundam AGE, Gundam Builders, and Iron-Blooded Orphans. The main content area has a header "Assembly guide" with an image of a model. Below it is a "DESCRIPTION" section with a red border containing text about the model's features. To the right of the description are four small images of different Gundam models.

Fields

No	Field Name	Description	Read only	Mandatory	Control Type	Data Type	Length
1	Model information	Model detail info	Yes	Yes	N/A	String	N/A
3	Model images	Model images	Yes	Yes	N/A	Img	N/A
4	Model description	Model's description	Yes	Yes	N/A	String	N/A

Buttons/Hyperlinks

No	Function	Description	Validation	Outcome
2	Rating	User evaluates a model	Each user can only evaluate a model for one time	Model detail info with user's rating

5.1.1.3. Display TradePost Listing view

Trade Listing My Trade My Order

1 Post new trade

2 Search a trade...
Enter your keyword and press enter. If you need search with location and range, click on more button and fills out your information.

3 More

4 Search with location and range

5 Location: 1358/24/14 Quang Trung, Go Vap, TPHCM

6 Range: 500 meters

7

8 Sorting by: Date Descending

9 All Buying Selling

10

11 \$50.00
Gundam abc - Title here!
Brand here / Model here
7 Jan, 16 10:10 pm Used

12

13

14

15

16

17

18

19

20

21

22

23

24

25

26

27

28

29

30

31

32

33

34

35

36

37

38

39

40

41

42

43

44

45

46

47

48

49

50

51

52

53

54

55

56

57

58

59

60

61

62

63

64

65

66

67

68

69

70

71

72

73

74

75

76

77

78

79

80

81

82

83

84

85

86

87

88

89

90

91

92

93

94

95

96

97

98

99

100

101

102

103

104

105

106

107

108

109

110

111

112

113

114

115

116

117

118

119

120

121

122

123

124

125

126

127

128

129

130

131

132

133

134

135

136

137

138

139

140

141

142

143

144

145

146

147

148

149

150

151

152

153

154

155

156

157

158

159

160

161

162

163

164

165

166

167

168

169

170

171

172

173

174

175

176

177

178

179

180

181

182

183

184

185

186

187

188

189

190

191

192

193

194

195

196

197

198

199

200

201

202

203

204

205

206

207

208

209

210

211

212

213

214

215

216

217

218

219

220

221

222

223

224

225

226

227

228

229

230

231

232

233

234

235

236

237

238

239

240

241

242

243

244

245

246

247

248

249

250

251

252

253

254

255

256

257

258

259

260

261

262

263

264

265

266

267

268

269

270

271

272

273

274

275

276

277

278

279

280

281

282

283

284

285

286

287

288

289

290

291

292

293

294

295

296

297

298

299

300

301

302

303

304

305

306

307

308

309

310

311

312

313

314

315

316

317

318

319

320

321

322

323

324

325

326

327

328

329

330

331

332

333

334

335

336

337

338

339

340

341

342

343

344

345

346

347

348

349

350

351

352

353

354

355

356

357

358

359

360

361

362

363

364

365

366

367

368

369

370

371

372

373

374

375

376

377

378

379

380

381

382

383

384

385

386

387

388

389

390

391

392

393

394

395

396

397

398

399

400

401

402

403

404

405

406

407

408

409

410

411

412

413

414

415

416

417

418

419

420

421

422

423

424

425

426

427

428

429

430

431

432

433

434

435

436

437

438

439

440

441

442

443

444

445

446

447

448

449

450

451

452

453

454

455

456

457

458

459

460

461

462

463

464

465

466

467

468

469

470

471

472

473

474

475

476

477

478

479

480

481

482

483

484

485

486

487

488

489

490

491

492

493

494

495

496

497

498

499

500

501

502

503

504

505

506

507

508

509

510

511

512

513

514

515

516

517

518

519

520

521

522

523

524

525

526

527

528

529

530

531

532

533

534

535

536

537

538

539

540

541

542

543

544

545

546

547

548

549

550

551

552

553

554

555

556

557

558

559

560

561

562

563

564

565

566

567

568

569

570

571

572

573

574

575

576

577

578

579

580

581

582

583

584

585

586

587

588

589

590

591

592

593

594

595

596

597

598

599

600

601

602

603

604

605

606

607

608

609

610

611

612

613

614

615

616

617

618

619

620

621

622

623

624

625

626

627

628

629

630

631

632

633

634

635

636

637

638

639

640

641

642

643

644

645

646

647

648

649

650

651

652

653

654

655

656

657

658

659

660

661

662

663

664

665

666

667

668

669

670

671

672

673

674

675

676

677

678

679

680

681

682

683

684

685

686

687

688

689

690

691

692

693

694

695

696

697

698

699

700

701

702

703

704

705

706

707

708

709

710

711

712

713

714

715

716

717

718

719

720

721

722

723

724

725

726

727

728

729

730

731

732

733

734

735

736

737

738

739

740

741

742

743

744

745

746

747

748

749

750

751

752

753

754

755

756

757

758

759

760

761

762

763

764

765

766

767

768

769

770

771

772

773

774

775

776

777

778

779

780

781

782

783

784

785

786

787

788

789

790

791

792

793

794

795

796

797

798

799

800

801

802

803

804

805

806

807

808

809

810

811

812

813

814

815

816

817

818

819

820

821

822

823

824

825

826

827

828

829

830

831

832

833

834

835

836

837

838

839

840

841

842

843

844

845

846

847

848

849

850

851

852

853

854

855

856

857

858

859

860

861

862

863

864

865

866

867

868

869

870

871

872

873

874

875

876

877

878

879

880

881

882

883

884

885

886

887

888

889

890

891

892

893

894

895

896

897

898

899

900

901

902

903

904

905

906

907

908

909

910

911

912

913

914

915

916

917

918

919

920

921

922

923

924

925

926

927

928

929

930

931

932

933

934

935

936

937

938

939

940

941

942

943

944

945

946

947

948

949

950

951

952

953

954

955

956

957

958

959

960

961

962

963

964

965

966

967

968

969

970

971

972

973

974

975

976

977

978

979

980

981

982

983

984

985

986

987

988

989

990

991

992

993

994

995

996

997

998

999

1000

Fields

No	Field Name	Description	Read only	Mandatory	Control Type	Data Type	Length
2	Search tradepost	Enter your keyword to search your tradepost.	No	Yes	Textbox	String	N/A
4	Search with location and range	Enable/Disable search with location and range	No	No	Checkbox	Boolean	N/A
5	Location	Display your location here	Yes	Yes	Textbox	String	N/A
7	Range	Slide to choose your range you want to search	No	Yes	Range input	Integer	N/A
8	Sorting	Select your sorting type.	No	Yes	Drop list	Integer	N/A

Buttons/Hyperlinks

No	Function	Description	Validation	Outcome
1	Post New Trade	Go to Post New Trade page and display post new trade form.	N/A	Go to post new trade Page
3	More search	Show/Hide more search panel.	N/A	Show/Hide search form
6	Edit Location	Show Location editor modal.	N/A	Show edit modal

9	Change tab	Change tab view group by all/selling/buying.	N/A	Change tab
10,11	View TradePost details	Go to TradePost details view page to display all information about tradepost.	N/A	Go to post details Page
12	View owner profile	Go to tradepost owner profile.	N/A	Go to profile Page
13	Paging	Separate content to page list.	N/A	Change page content

5.1.1.4. Display TradePost details view

1 Home» Trade market» Gundam abc - Title here

2 

3 \$950.00

4 **Gundam abc - Title here**

Trade ID: 251716763

⌚ 7 Jan, 16 10:10 pm ⚖ Los Angeles, USA

 Yury Corporation need to sell

5 **Product Information**

Condition: Used

Brand: Brand here

Model: Model here

Quantity: 999

Rating: ★★★★☆

6 **SEND ORDER**

7 Share this ad



8 **Description**

9 **Trade Information**

 Delivery: Meet in person

 More trades by **Yury Corporation**

 Contact: Send order to see

 Report this trade

10 **Sed ut perspiciatis unde omnis iste natus error sit voluptatem accusantium doloremque laudantium, totam rem aperiam, eaque ipsa quae ab illo inventore veritatis et quasi architecto beatae vitae dicta sunt explicabo. Nemo enim ipsam voluptatem quia voluptas sit aspernatur aut odit aut fugit, sed quia consequuntur magni**

Fields: N/A

Buttons/Hyperlinks

No	Function	Description	Validation	Outcome
1	Page Address	Show address link to explore post.	N/A	Go to Page
2	Previous Image	Previous control for product Image slideshow.	N/A	Click for previous Image.
3	Next Image	Next control for product Image slideshow.	N/A	Click for next Image.
4	Slideshow indicator	List images in slideshow. Click to view this image.	N/A	Change Image.
5	Owner link	Click to go to owner profile page.	N/A	Go to profile Page
6	Send order button	Click to show send order modal.	N/A	Show send order modal
7	Share this trade button list	Click on any social button to share this trade.	N/A	Share link to social page.
8	More trade post from owner	Click to show modal with trade post list from this trade post owner.	N/A	Show more post modal
9	Report button	Click to show report form.	N/A	Show report modal.

5.1.1.5. Display MyOrder view

Trade Listing My Trade **My Order**

1 On Payment Pending Approval Rejected History

2 Order From: Title of Trade Post
 Phone: 0343777228
 Email: tksangit@gmail.com
 Quantity: 1 - Need to pay: 50\$
 Ordered On: 7 Jan, 16 10:10 pm Accepted On: 7 Jan, 16 10:10 pm

3 **4** **5**

6

Fields: N/A

Buttons/Hyperlinks:

No	Function	Description	Validation	Outcome
1	Change tab	Click to change tab.	N/A	Change tab group

2,3	View Post details	Click to go to trade post details page.	N/A	Go to details page.
4	Get direction	Click to get direction to trade post owner.	N/A	Show direction modal.
5	Cancel order	Click to cancel order.	N/A	Show cancel order modal.
6	Paging	Separate content to page list.	N/A	Change page content.

5.1.1.6. Display article listing view

Gunpla World

Login Register

Home Pages ▾ Mega Menu ▾ Articles Events Trade Market Contact

Featured articles

1 2

CUSTOM BUILD 3

Custom Build: MG 1/100 Wing Gundam Zero "Lucifer"
7:00 AM | April 14
MODELER: ROKU First revealed at the Gundam Build Divers GBWC GIMM BALL's World Challenge teaser. Modeler used MG 1/100 Wing Gundam...



4 **CUSTOM BUILD** 5

Custom Build: MG 1/100 RX-78-02 Gundam [THE ORIGIN] Resin Conversion
7:00 AM | April 14 6
MODELER: ROKU First revealed at the Gundam Build Divers GBWC GIMM BALL's World Challenge teaser. Modeler used MG 1/100 Wing Gundam...



7

CUSTOM BUILD
Custom Build: MG 1/100 Wing Gundam Zero "Lucifer"
7:00 AM | April 14
MODELER: ROKU First revealed at the Gundam Build Divers GBWC GIMM BALL's World Challenge teaser. Modeler used MG 1/100 Wing Gundam...



Fields

No	Field Name	Description	Read only	Mandatory	Control Type	Data Type	Length
1	Search input text	Input search keyword, search by article title	No	No	Text input	String	N/A
3	Search record	Result record from searching	Yes	No	N/A	N/A	N/A
6	Article's date	The day that the article was posted on	Yes	No	N/A	String	N/A

Buttons/Hyperlinks

No	Function	Description	Validation	Outcome
2	Search	Send search command to server with value in search input text	N/A	List records match with the input keyword
4	Article category	The category that this article belongs to	N/A	Search articles with that category
5	Article's title	Title of the article	N/A	Redirects to the article's detail view
7	Article's content	A short part of the article's description	N/A	Redirects to the article's detail view

5.1.1.7. Article detail view

HGUC 1/144 AMX-104 R-Jarja - Release Info

7:00 AM | April 14

2

[Edit this article](#)

3

HGUC 1/144 AMX-104 R-Jarja
Release Date: January 2019
Price: 2,052 Yen

4

One of the most anticipated ZZ Mobile Suit is finally getting its High Grade Universal Century release! The kit was first revealed at the Char's Counterattack World event at The Gundam Base Tokyo.



Fields

No	Field Name	Description	Read only	Mandatory	Control Type	Data Type	Length
1	Article's title	The title of the article	Yes	No	N/A	String	N/A
2	Article's date	The day that the article was posted on	Yes	No	N/A	String	N/A
4	Article's content	The content of the article	Yes	No	N/A	String	N/A

Buttons/Hyperlinks

No	Function	Description	Validation	Outcome
3	Edit this article	Let the user edit the article	N/A	Redirects the user to the edit article view

5.1.1.8. Edit Article view

Gunpla World

Login Register Search

Home Pages ▾ Mega Menu ▾ Articles Events Trade Market Contact

Edit article

Your article will be taken down and reviewed by the administrators before being posted on the website again

Article title:

Gundam #111 1

Edit your article below: 2



HGUC 1/144 AMX-104 R-Jarja
Release Date: January 2019
Price: 2,052 Yen

One of the most anticipated ZZ Mobile Suit is finally getting its High Grade Universal Century release! The kit was first revealed at the Char's Counterattack World event at The Gundam Base Tokyo.



Delete this article 3 Submit 4

Fields

No	Field Name	Description	Read only	Mandatory	Control Type	Data Type	Length
1	Article's title	The title of the article	No	Yes	N/A	String	N/A
2	Article's content	The content of the article	No	Yes	N/A	String	N/A

Buttons/Hyperlinks

No	Function	Description	Validation	Outcome
3	Delete this article	Let the user delete the article	N/A	Prompts the user for confirmation before deleting the article
4	Submit	Submit the article	All required fields must not be	Success: article's information is saved to the database and redirects

			empty	the user to that article's page.
				Failure: show error message

5.1.1.9. Event list view

The screenshot shows a web page with a red header and a white content area. At the top, there is a navigation bar with links: Home, Pages, Mega Menu, Articles, Events, Trade Market, and Contact. Below the navigation, there is a section titled "Recent events". The first event card, highlighted with a red border, is for the "Gundam Builder World Cup 2018 – GBWC 2018 Vòng Loại Quốc Gia Việt Nam". It includes a search input field (1), a search button (2), and a title (3). The second event card, also highlighted with a red border, is for the "Gunpla Day 2018 – Sự Kiện Gundam Tại Việt Nam Cực Kỳ Hấp Dẫn". It includes a timestamp (4), a search input field (5), and a description (6). At the bottom, there is a navigation bar with page numbers (1 to 10) and a "7" icon.

Fields

No	Field Name	Description	Read only	Mandatory	Control Type	Data Type	Length
1	Search input text	Input search keyword, search by article title	No	No	N/A	String	N/A
3	Search result	Event records from searching	No	No	N/A	String	N/A

Buttons/Hyperlinks

No	Function	Description	Validation	Outcome

2	Search	Search event by title	N/A	List records match with the input keyword
4	Event's title	The title of the event	N/A	Redirects the user to the event's detail view
5	Event's date	The date the event was posted on	N/A	Redirects the user to the event's detail view
6	Event's content	A short part of the event's content	N/A	Redirects the user to the event's detail view

5.1.1.10. Event detail view

The screenshot shows the Gunpla World website with a red header bar containing the logo and navigation links for Login, Register, and Search. Below the header is a navigation menu with Home, Pages, Mega Menu, Articles, Events, Trade Market, and Contact. A red box highlights the main content area.

GUNDAM BUILDER WORLD CUP 2018 – GBWC 2018 VÒNG LOẠI QUỐC GIA VIỆT NAM

7:00 AM | April 14 2 1

3

GBWC 2018 GUNPLA BUILDERS WORLD CUP 2018
CUỘC THI LẮP RÁP GUNPLA CHÍNH THỨC TRÊN TOÀN THẾ GIỚI
THẾ GIỚI MUỐN THẤY "GUNPLA" CỦA BẠN
VÒNG THI QUỐC GIA

Mùa GBWC 2018 đã trở lại sau thành công của GBWC 2017 – vòng thi quốc gia lần đầu tiên tại Việt Nam. GBWC 2017 đã chọn ra những người chiến thắng có Gunpla đẹp và sáng tạo nhất Việt Nam. Họ đã mang Gunpla của mình thi đấu tại Nhật Bản – đất nước mẹ đẻ của Gunpla thế giới. Năm 2018, một mùa GBWC mới trở lại, đại diện tại Việt Nam là công ty TNHH Giải Trí Go Go.

GBWC là giải đấu quốc tế với nội dung lắp ráp, thiết kế, sáng tạo mô hình Gunpla, được tổ chức thường niên từ năm 2011. GBWC bao gồm hai Bảng đấu: Bảng thiếu niên (dành cho đối

Chương trình GBWC 2018 chính thức diễn ra từ 18/09 – 23/09/2018 tại tầng trệt Crescent Mall
 – 101 Tôn Nhật Tiên, Phú Mỹ Hưng, phường Tân Phú, Quận 7.

Các mùa thi GBWC hấp dẫn qua từng chặng thi, từ vòng thi quốc gia ở Việt Nam đến vòng chung kết tại Nhật Bản. Đối với các thí sinh tham gia cuộc thi, đây là cơ hội để cọ xát, thử thách khả năng chơi Gunpla của mình và trên hết, thể hiện tình yêu Gunpla với thế giới.

Cột mốc thứ 2 GBWC tại Việt Nam hy vọng sẽ đón nhận nhiều bài dự thi của fan Gunpla tại Việt Nam.
 Cùng theo dõi thông tin GBWC 2018 được cập nhật thường xuyên tại Fanpage facebook.com/GundamVietnamFanpage/ và trang web chính thức của cuộc thi <http://bandai-hobby.net/GBWC/>.

3

Ratings: 4
 4
 Remaining tickets: 42 5

Registration

Enter your Credit Card number:

 6

Number of tickets (must not be higher than the amount of remaining tickets):

 7

Register 8

Fields

No	Field Name	Description	Read only	Mandatory	Control Type	Data Type	Length
1	Event's title	The event's title	Yes	Yes	N/A	String	N/A
2	Event' date	The date the event was posted on	Yes	Yes	N/A	String	N/A
3	Event's content	The content of the event	Yes	Yes	N/A	String	N/A
5	Remaining tickets	The amount of tickets left available	Yes	Yes	N/A	String	N/A
6	Credit card number input	The number of member's credit card	No	No	N/A	String	N/A
7	Number of tickets	The number of tickets the user wants to buy	No	Yes	N/A	String	N/A

Buttons/Hyperlinks

No	Function	Description	Validation	Outcome
4	Ratings	Rate the event from 1-5 stars	N/A	Save the ratings to database
8	Register	Register the user to the event	N/A	Add the user to the list of event's attendees

5.1.2. Mobile application

5.2. Buyer/Seller

5.2.1. Web application

5.2.1.1. Display Post New TradePost view

Post New Trade

Trade an item or service

* Mandatory Fields

Type of trade*	<input type="radio"/> I want to sell <input type="radio"/> want to buy	1		
Title for your trade*	ex, P-bandai xxx brandnew.			
Photos for your trade (This will be your cover photo)	Select Files to Upload / Drag and Drop Files You can add multiple images.			
Condition*	<input type="radio"/> New <input type="radio"/> Used	4		
Price*	\$USD <input type="text"/>	<input type="checkbox"/> Negotiable	5	
Quantity*	<input type="text"/>		6	
Brand Name*	ex, P-bandai		7	
Model	ex, P150		8	
Description*	Write few lines about your products			9
10				

5000 characters left

5000 characters left

Quick rules

Posting an ad on [gundamworld.com](#) is free! However, all ads must follow our rules:

Trader Information

Your Name*	Lê Nguyễn Ngọc Sang	11
Your Email ID*	tksangit@gmail.com	12
Mobile Number*	0343 777 228	13
Address*	1358/28/14 Quang Trung, Gò Vấp, TPHCM	14
	 	15

[View Details](#) [Edit](#) [Delete](#)

By clicking "Post", you agree to our [Terms of Use](#) and [Privacy Policy](#) and acknowledge that you are the rightful owner of this item and using Trade to find a genuine buyer.

18

Fields

No	Field Name	Description	Read only	Mandatory	Control Type	Data Type	Length
1	Type of trade	Select your type of trade: sell or buy	No	Yes	Radio button group	Integer	N/A
2	Title of your trade	Type your title here.	No	Yes	Textbox	String	N/A
3	Upload Image	Click to choose your image.	No	Yes	File	Image file type: png, jpeg, jpg, gif	N/A
4	Condition	Choose your product condition: Used or New	No	Yes	Radio button group	Integer	N/A
5	Price	Type your price here	No	Yes	Textbox	String	N/A
6	Price Negotiable	Set your price can negotiable: Yes or No	No	No	Radio button	Boolean	N/A
7	Quantity	Type your quantity here	No	Yes	Textbox	Integer	N/A
8	Brand name	Type your product brand here	No	No	Textbox	String	N/A

9	Model	Type your product model here	No	No	Textbox	String	N/A
10	Description	Type your product description here	No	Yes	Textbox	String	N/A
11	Your name	Input your name here	No	Yes	Textbox	String	N/A
12	Your Email	Your email will show here	Yes	Yes	Textbox	String	N/A
13	Mobile Number	Input your mobile number here	No	Yes	Textbox	String	N/A
14	Address	Your location will show here	Yes	No	Textbox	String	N/A
17	Agree term	You need agree button to post your trade	No	Yes	Radio button	Boolean	N/A

Buttons/Hyperlinks

No	Function	Description	Validation	Outcome
15	Auto get your location	Click to auto get your location and fills result to address field.	N/A	Call auto get location function
16	Edit your location	Click to show modal to select correct your location	N/A	Show edit location modal
18	Post your trade	Click to submit your trade post	N/A	Submit form.

5.2.1.2. Display My trade view

The screenshot displays the 'My Trade' section of a trading platform. At the top, there are tabs for 'Trade Listing', 'My Trade' (which is selected and highlighted in red), and 'My Order'. A green button labeled '+ Post new trade' is located in the top right corner. Below the tabs is a search bar with the placeholder 'Search in my trade...'. To the right of the search bar is a note: 'Enter your keyword and press enter.' A red number '1' points to this note.

Below the search bar is a sorting dropdown set to 'Date Descending', indicated by a red number '2'. To the right of the sorting dropdown is a red box containing three filter buttons: 'On Trading' (green), 'Pending Approval' (light blue), and 'Rejected' (light gray). A red number '3' points to the 'On Trading' button.

The main content area displays five trading items, each consisting of a thumbnail image, a price of '\$800.00', a title ('Gundam abc - Title here!'), and links to 'Brand here / Model here'. Each item also includes a 'Posted On' timestamp ('7 Jan, 16 10:10 pm') and two activity counts: 'Request Orders: 21' and 'On Payment: 12'. Red numbers from 4 to 10 are placed near these elements to identify them:

- Thumbnail: 4
- Price: 5
- Title: 5
- Brand/Model link: 5
- Posted On: 6
- Request Orders: 6
- On Payment: 7
- More options: 8
- Edit icon: 9
- Delete icon: 10

At the bottom of the list is a pagination control showing '1' (selected) and '2' through '10'.

Fields

No	Field	Description	Read	Mandatory	Control	Data	Length

	Name		only		Type	Type	
1	Search in my trade	Enter your keyword to search your tradepost.	No	Yes	Textbox	String	N/A
2	Sorting	Select your sorting type.	No	Yes	Drop list	Integer	N/A

Buttons/Hyperlinks

No	Function	Description	Validation	Outcome
3	Change tab	Change tab view group by On trading/Pending Approval/Rejected	N/A	Change page content
4,5	View TradePost details	Go to TradePost details view page to display all information about tradepost.	N/A	Go to post details Page
6	View all Request Order	View all request order for the trade post.	N/A	Show all request modal
7	View all On Payment	View all on payment for the trade post.	N/A	Show on payment modal
8	Edit this trade	Edit information of this trade.	N/A	Show edit modal
9	Delete this trade	Delete the trade	N/A	Show delete modal

10	Update Quantity	Update trade quantity	N/A	Show update quantity modal
11	Paging	Separate content to page list.	N/A	Change page content

5.2.2. Mobile Application

5.3. Guest

5.3.1. Web application

5.3.1.1. Register account

The image shows a registration form titled "Register" set against a background of a cloudy sky with a bird. The form consists of six input fields arranged in a 3x2 grid. Each field is labeled with a red number from 1 to 6. Below the grid is a link to log in using a username, and at the bottom right is a "Register" button. A note at the bottom right suggests logging in if you already have an account.

1	First Name:	2	Username:
3	Last Name:	4	Email:
5	Password:	6	Confirm Password:

Use your username to login

Register

You have an account already? Log in here

Fields

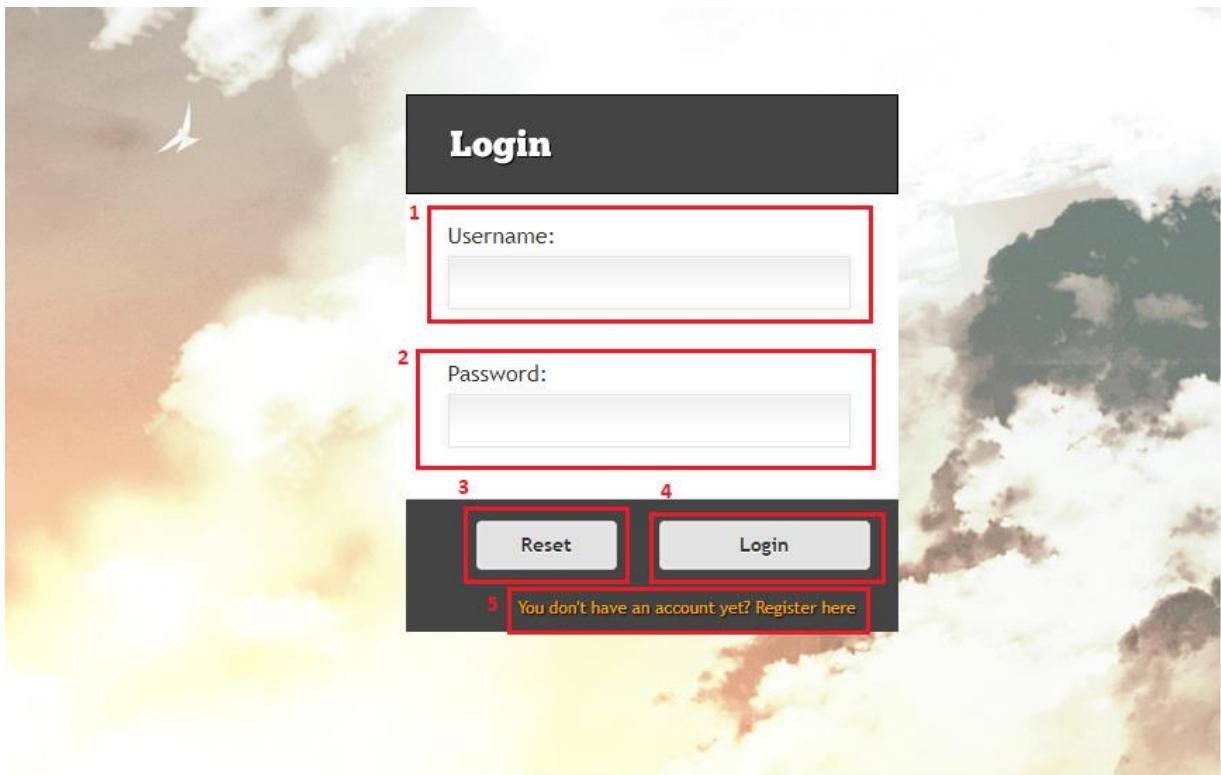
No	Field Name	Description	Read only	Mandatory	Control Type	Data Type	Length
1	Firstname	User's first name	No	Yes	Text input	String	<= 50
2	Username	User's username for login	No	Yes	Text input	String	<= 50
3	Lastname	User's lastname	No	Yes	Text input	String	<= 50
4	Email	User's email	No	Yes	Text input	String	<= 50
5	Password	User's password	No	Yes	Text input	String	<= 50
6	ConfirmPassword	To confirm password, this field's value must be the same as password	No	Yes	Text input	String	N/A

Buttons/Hyperlinks

No	Function	Description	Validation	Outcome
7	Register	Send register command with values	Entered email must match the required format Entered email and username must not be duplicate in database Length of field's value must not be out of range Password and confirm password must match to each other All required fields must be filled out	Login page with register success message

8	Redirect to login page	For whom already had account	N/A	Login page
---	------------------------	------------------------------	-----	------------

5.3.1.2. Login



Fields

No	Field Name	Description	Read only	Mandatory	Control Type	Data Type	Length
1	Username	User's username	No	Yes	Text input	String	N/A
2	Password	User's password	No	Yes	Text input	String	N/A

Buttons/Hyperlinks

No	Function	Description	Validation	Outcome
----	----------	-------------	------------	---------

3	Reset	Reset all fields	N/A	All fields are clear
4	Login	Send login command to the system with username and password	Check exist in database	Success: show homepage Fail: show error message
5	Redirect to Register page	For whom doesn't have an account	N/A	Register page is shown

5.3.2. Mobile application

5.4. Authenticated User

5.4.1. Web application

5.4.1.1. Get profile

The screenshot shows a user profile editing interface. The top navigation bar includes Home, Pages, Mega Menu, GUNDAM, Articles, Events, Trade Market, and Contact. The main content area displays a user's profile information with various input fields and a sidebar.

Numbered callouts point to specific elements:

- 1: Username input field (tunghoang122)
- 2: Edit button
- 3: First name input field (Tung)
- 4: Email input field (tunghnmse62231@fpt.edu.vn)
- 5: Middle name input field (Minh)
- 6: Address input field (You can't know)
- 7: Last name input field (Hoang Ngo)
- 8: Birthday input field (2-12-1997)
- 9: Tel input field (0981875113)
- 10: Password input field (*****)
- 11: Status dropdown (Active)
- 12: Activity sidebar showing Proposal (125), Buy (13), Sell (37), and Reputation (★★★★★)
- 13: Profile picture placeholder

Fields

No	Field Name	Description	Read	Mandatory	Control Type	Data	Length
----	------------	-------------	------	-----------	--------------	------	--------

			only			Type	
1	Username	User's username for login	Yes	Yes	Label	String	<= 50
3	Firstname	User's firstname	Yes	Yes	Text input	String	<= 50
4	Email	User's email	Yes	Yes	Text input	String	<=50
5	Middlename	User's middle name	Yes	No	Text input	String	<=50
6	Address	User's address	Yes	No	Text input	String	<= 250
7	Lastname	User's lastname	Yes	Yes	Text input	String	<= 50
8	Birthday	User's birthday	Yes	No	DateTimePicker	String	<= 20
9	Tel	User's mobile number	Yes	No	Text input	String	10-11
10	Password	User's password	Yes	Yes	Text input	String	<= 50
11	Status	User's status	Yes	Yes	Text input	String	N/A
12	Statistic	User's statistics and ratings	Yes	Yes	N/A	String	N/A
13	Avatar	User's image	Yes	Yes	Img	Img	N/A

Buttons/Hyperlinks

No	Function	Description	Validation	Outcome
2	Edit	Redirect to edit page	N/A	Edit page is shown

5.4.1.2. Edit profile

The screenshot shows a user profile editing interface. At the top, there's a navigation bar with links like Home, Pages, GUNDAM, Articles, Events, Trade Market, and Contact. Below the navigation is a header with a user icon and the text "tunghoang122". To the right of the user icon is a blue "EDIT" button. The main area contains various input fields and sections:

- Profile Picture:** A placeholder image with a "User" icon. Numbered 16.
- Username:** Input field containing "tunghoang122". Numbered 1.
- First name:** Input field containing "Tung". Numbered 3.
- Email:** Input field containing "tunghnmse62231@fpt.edu.vn". Numbered 4.
- Middle name:** Input field containing "Minh". Numbered 5.
- Address:** Input field containing "You can't know". Numbered 6.
- Last name:** Input field containing "Hoang Ngo". Numbered 7.
- Birthday:** Input field containing "2-12-1997". Numbered 8.
- Tel:** Input field containing "0981875113". Numbered 9.
- Password:** Input field containing ".....". Numbered 10.
- Activity:** A dropdown menu showing "Proposal" (125), "Buy" (13), "Sell" (37), and "Reputation" (★★★★★). Numbered 14.
- Status:** A dropdown menu showing "Active". Numbered 11.
- Buttons:** A green "SAVE" button and a grey "RESET" button at the bottom. Numbered 12 and 13 respectively.
- File Input:** A "Choose File" button with "No file chosen" displayed. Numbered 15.

Fields

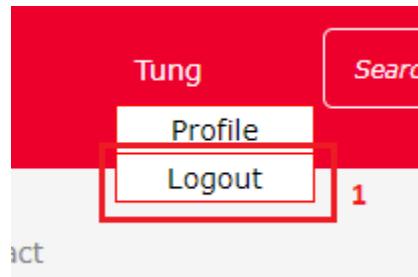
No	Field Name	Description	Read only	Mandatory	Control Type	Data Type	Length
1	Username	User's username for login	Yes	Yes	Label	String	<= 50
3	Firstname	User's firstname	No	Yes	Text input	String	<= 50
4	Email	User's email	No	Yes	Text input	String	<= 50
5	Middlename	User's middle name	No	No	Text input	String	<= 50
6	Address	User's address	No	No	Text input	String	<= 250
7	Lastname	User's lastname	No	Yes	Text input	String	<= 50
8	Birthday	User's birthday	No	No	DateTimePicker	String	<= 20
9	Tel	User's mobile number	No	No	Text input	String	10-11
10	Password	User's	Yes	Yes	Text input	String	<= 50

		password					
11	Status	User's status	Yes	Yes	Text input	String	N/A
14	Statistic	User's statistics and ratings	Yes	Yes	N/A	String	N/A
16	Avatar	User's image	No	Yes	Img	Img	N/A

Buttons/Hyperlinks

No	Function	Description	Validation	Outcome
2	Edit	Redirect to edit page	N/A	Edit page is shown
12	Save	Send edit profile with values	Email must match the required format All required fields must be filled out Phone number is string of number Email must not be duplicate in database Length of all field's values must be in range	Profile page is shown
13	Reset	All fields are reset to the beginning values	N/A	All fields are reset to the beginning values
15	Choose file	Update avatar	Check valid type of file image	Success: avatar is updated with the selected image from user Fail: show error

				message
--	--	--	--	---------



5.4.1.3. Logout

Buttons/Hyperlinks

No	Function	Description	Validation	Outcome
1	Logout	Logout from system, remove session	N/A	Homepage is shown

5.4.2. Mobile application

5.5. Administrator

5.5.1. Web application

5.5.1.1. Create model

Create Model

13 **SAVE** CANCEL 14

Model Information		Description
Code	1	Prototype MS with Psycomu appeared in RE / 100! Refreshed the details and proportions according to RE / 100 based on the original image of the ZZ fan highly anticipated hammer hammer. Flexible arm has a large number of gimmicks including soft materials and repeatable all range attack by lead wire.
Name	2	
Product Series	3	
Perfect Grade		
Series Title	4	
Reborn-One Hundred		
Manufacturer	5	
Bandai		
Price	6	
Price		
Status	7	
		Model's Images
Upload pictures		
Choose File		No file chosen
10		11
Package		12
		13

Fields

No	Field Name	Description	Read only	Mandatory	Control Type	Data Type	Length
1	Code	Model's code	No	Yes	Text input	String	<= 50
2	Name	Model's name	No	Yes	Text input	String	<= 50
3	Product Series	Model's product series	No	Yes	Dropdown list	String	N/A
4	Series Title	Model's series title	No	Yes	Dropdown list	String	N/A
5	Manufacturer	Model's manufacturer	No	Yes	Dropdown list	String	N/A
6	Price	Model's price	No	No	Text input	String	<= 20
7	Status	Model's status	No	Yes	Dropdown list	String	N/A
8	Description	Model's description	No	No	Text input	String	<= 350
10	Type of image	Type of image	No	Yes	Dropdown list	String	<= 50

Buttons/Hyperlinks

No	Function	Description	Validation	Outcome
9	Choose image	Choose model's image	Check valid type of file image	Image name is shown next by
11	Add more images	System displays 1 new row with the following lables and fields in a horizontal line: <ul style="list-style-type: none"> • Import image: input type file • Image type: dropdown list • “+”: button • “-”: button 	N/A	New row with the following lables and fields in a horizontal line: <ul style="list-style-type: none"> • Import image: input type file • Image type: dropdown list • “+”: button • “-”: button
12	Remove image	System removes the selected rows, includes: <ul style="list-style-type: none"> • Import image: input type file • Image type: dropdown list • “+”: button 	N/A	System removes the selected rows, includes: <ul style="list-style-type: none"> • Import image: input type file • Image type: dropdown list • “+”: button
13	Save	Send create model command	All required fields must be input Invalid format of code Duplicate model's code Length of fields's	Success: model information is saved to database and display model detail page Fail: show error message

			value must be in range	
14	Cancel	Cancel creating model	N/A	Search model page is displayed

5.5.1.1. Edit model

Edit Model {Code Of The Model}

Model Information

- Code (1)
- Name (2)
- Product Series (3)
- Series Title (4)
- Manufacturer (5)
- Price (6)
- Status (7)

Description

(8) Prototype MS with Psycomu appeared in RE / 100!
Refreshed the details and proportions according to RE / 100 based on the original image of the ZZ fan highly anticipated hammer hammer.
Flexible arm has a large number of gimmicks including soft materials and repeatable all range attack by lead wire.

Model's Images

Upload pictures
Choose File | No file chosen (9)
10 Package (11) (12)

13 SAVE 14 CANCEL

Fields

No	Field Name	Description	Read only	Mandatory	Control Type	Data Type	Length
1	Code	Model's code	No	Yes	Text input	String	<= 50
2	Name	Model's name	No	Yes	Text input	String	<= 50
3	Product Series	Model's product series	No	Yes	Dropdown list	String	N/A
4	Series Title	Model's series title	No	Yes	Dropdown list	String	N/A
5	Manufacturer	Model's manufacturer	No	Yes	Dropdown list	String	N/A
6	Price	Model's price	No	No	Text input	String	<= 20
7	Status	Model's status	No	Yes	Dropdown list	String	N/A

8	Description	Model's description	No	No	Text input	String	<= 350
10	Type of image	Type of image	No	Yes	Dropdown list	String	<= 50

Buttons/Hyperlinks

No	Function	Description	Validation	Outcome
9	Choose image	Choose model's image	Check valid type of file image	Image name is shown next by
11	Add more images	<p>System displays 1 new row with the following lables and fields in a horizontal line:</p> <ul style="list-style-type: none"> • Import image: input type file • Image type: dropdown list • “+”: button • “-”: button 	N/A	<p>New row with the following lables and fields in a horizontal line:</p> <ul style="list-style-type: none"> • Import image: input type file • Image type: dropdown list • “+”: button • “-”: button
12	Remove image	<p>System removes the selected rows, includes:</p> <ul style="list-style-type: none"> • Import image: input type file • Image type: dropdown list • “+”: button 	N/A	<p>System removes the selected rows, includes:</p> <ul style="list-style-type: none"> • Import image: input type file • Image type: dropdown list • “+”: button

13	Save	Send edit model command	All required fields must be input Invalid format of code Duplicate model's code Length of fields's value must be in range	Success: model information is saved to database and display model detail page Fail: show error message
14	Cancel	Cancel creating model	N/A	Search model page is displayed

5.5.1.3. Delete model

Buttons/Hyperlinks

No	Function	Description	Validation	Outcome
1	Delete model	Send delete model command	N/A	The selected model is deleted and the search model page is displayed.

				shown
--	--	--	--	-------

5.5.1.4. Search account

The screenshot shows a web application interface titled 'Account'. At the top right, there is a breadcrumb navigation: 'Home > Accounts'. Below the title, a red box highlights a section labeled '1 Latest Accounts'. This section contains a search bar with the placeholder 'Search...' and a magnifying glass icon. A second red box highlights a table below this section. The table has a header row with columns: ID, Username, Role, Email, Full Name, Address, Birthday, and Status. The body of the table contains 12 rows of account data. The last row of the table is also highlighted with a red box.

ID	Username	Role	Email	Full Name	Address	Birthday	Status
1	minhtung	buyer	Active				
2	khanhnguyen	member	Banned				
3	sanglenguyennngoc	seller	Active				
1	hieptranbuu	buyer	Active				
2	nguyenchikhac	member	Banned				
3	xuankiet	seller	Active				
1	okmenthang	buyer	Active				
2	tunghoang122	member	Banned				
3	hoangngominhthung212192	seller	Active				
ID	Username	Role	Status				

Fields

No	Field Name	Description	Read only	Mandatory	Control Type	Data Type	Length
1	Search field	Search accounts	No	No	Text input	String	N/A
2	Account information	Account's information	No	Yes	N/A	String	N/a

5.5.1.5. Ban/unban/edit account

tunghoang122

BAN EDIT



Upload a different photo...
Choose File No file chosen

Activity	125
Proposal	125

First name Tung

Middle name Minh

Last name Hoang Ngo

Tel 0981875113

Email tunghnmse62231@fpt.edu.vn

Address You can't know

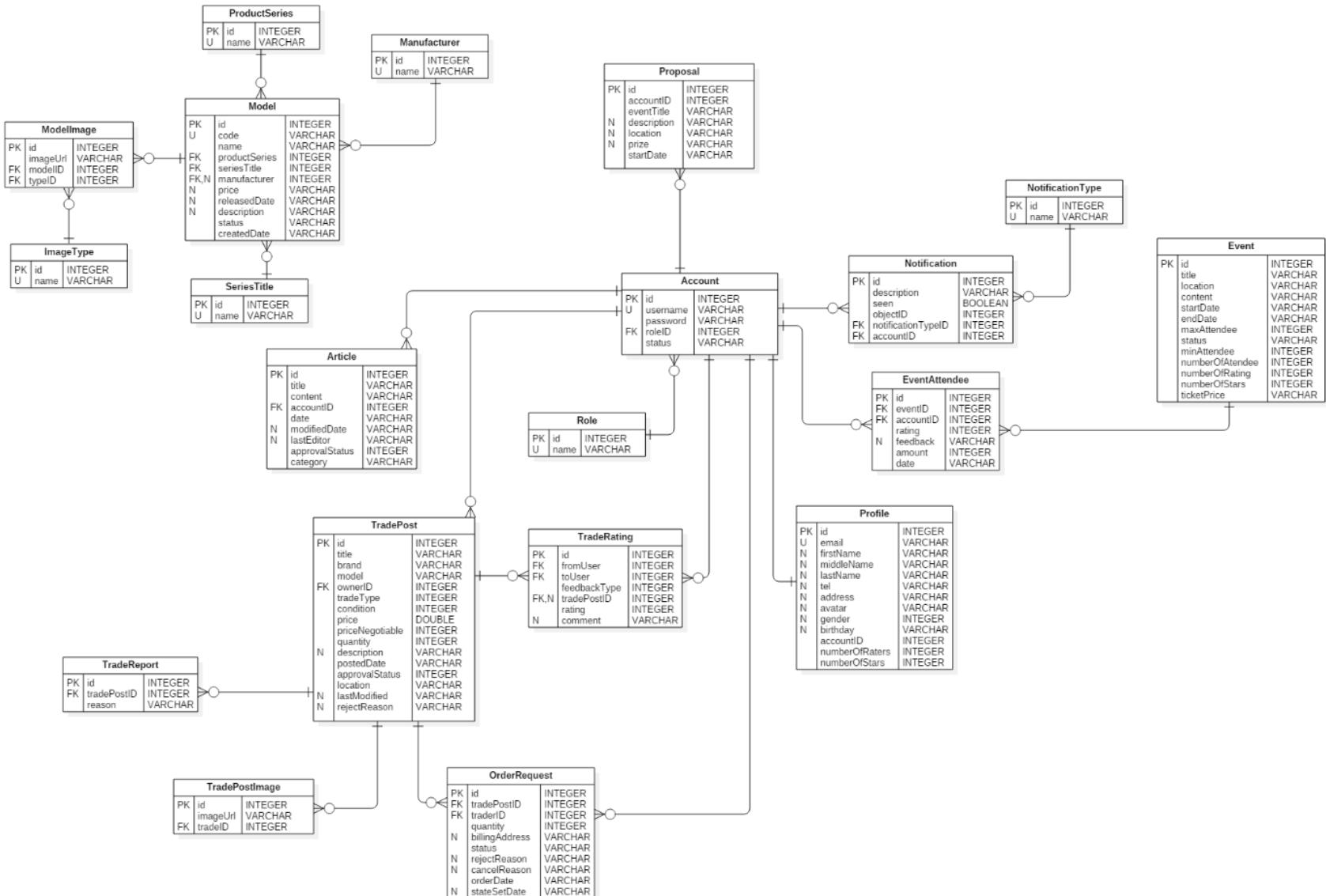
Birthday 2-12-1997

Password

5.5.2. Mobile application

6. Database Design

6.1. Entity Relationship Diagram (ERD)



6.2. Data Dictionary

Entity Data Dictionary	
Entity name	Description
Model	Contain model information
SeriesTitle	Contain series title information, this is a type of model's category
ProductSeries	Contain product series information, this is a type of model's category
Manufacturer	Contain model's manufacturer information
ModelImage	Contain model's images
ImageType	Contain model's image type
Article	Contain article information which is posted by authenticated users
Proposal	Contain proposal information which is posted by members
Account	Contain account information
Profile	Contain user's profile information
Role	Contain role information
Event	Contain event information, which is created by administrator
EventAttendee	Contain information about event's attendee, we also use this table to contain user's evaluation about the events
Notification	Contain notification information
NotificationType	Contain types of notification
TradePost	Contain trade post information, which is created by buyer/seller
TradeRating	Contain user's evaluation about the exchange
TradePostImage	Contain trade post's images
TradeReport	Contain information about user's report on exchange
OrderRequest	Contain member's order request to trade posts

7. Algorithms

E. System Implementation & Test

1. Introduction

1.1. Overview

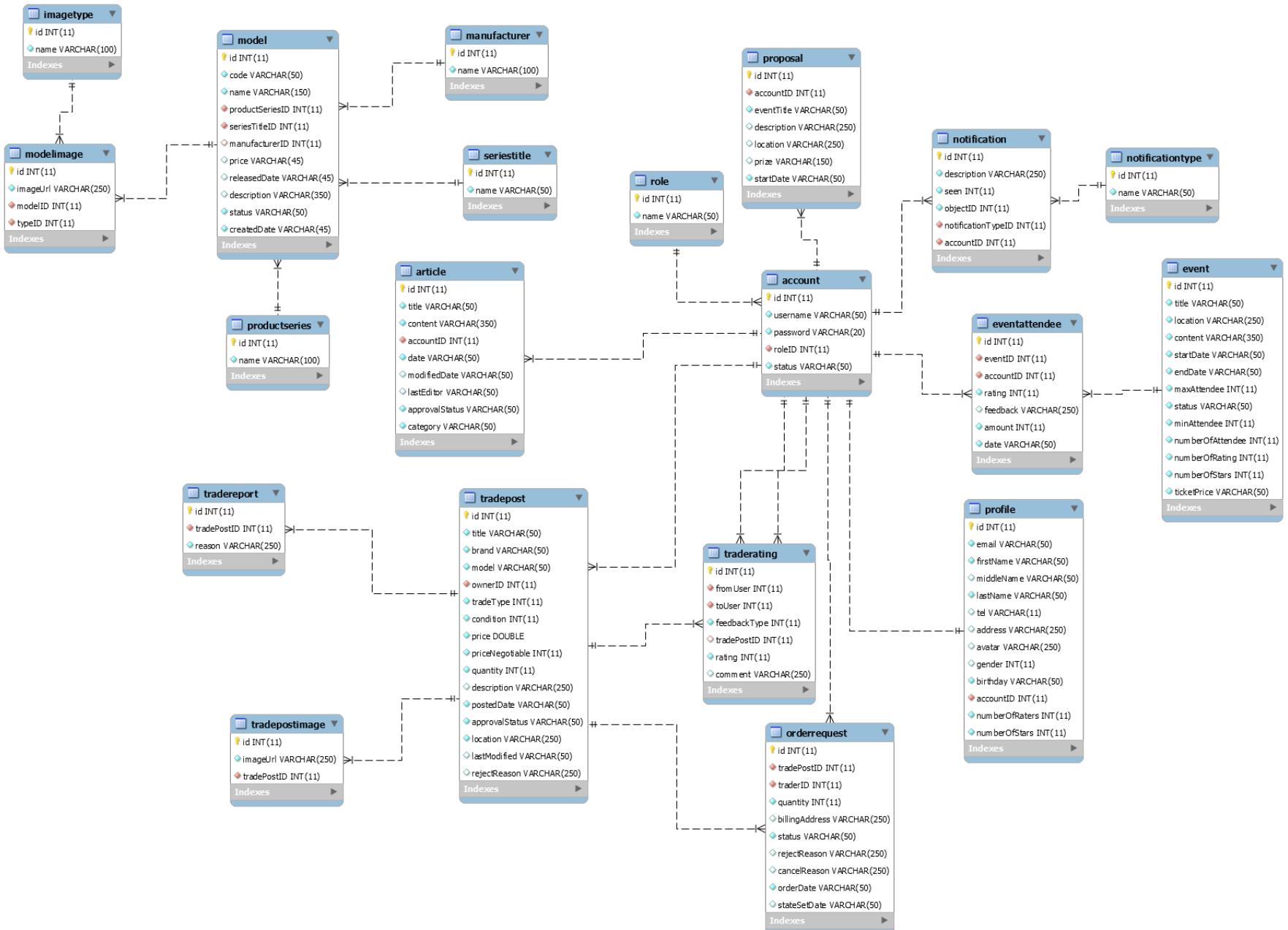
This section provides in detail all necessary information about implementation information of **GWA** which includes physical database design, system state machine diagram and testing procedures which includes test plans, test cases and test result.

1.2. Test Approach

We use black box test approach to estimate our system quality. It's the most appropriate testing method for the system to test all functions from a user's point of view and help in exposing discrepancies in the specifications. Our testers don't need to know how the software has been implemented, tests can be conducted by a body independent from the developers, allowing for an objective perspective and the avoidance of developer-bias.

2. Database Relationship Diagram

2.1. Physical Diagram



2.2. Data dictionary

Data dictionary: describe content of all tables	
Table name	Description
model	Table which contains model information
seriesitle	Table which contains series title information, this is a type of model's category
productseries	Table which contains product series information, this is a type of model's category
manufacturer	Table which contains model's manufacturer information
modelimage	Table which contains model's images
imagetype	Table which contains model's image type
article	Table which contains article information which is posted by authenticated users
proposal	Table which contains proposal information which is posted by members
account	Table which contains account information
profile	Table which contains user's profile information
role	Table which contains role information
event	Table which contains event information, which is created by administrator
eventattendee	Table which contains information about event's attendee, we also use this table to contain user's evaluation about the events
notification	Table which contains notification information
notificationtype	Table which contains types of notification
tradepost	Table which contains trade post information, which is created by buyer/seller
traderating	Table which contains user's evaluation about the exchange
tradepostimage	Table which contains trade post's images
tradereport	Table which contains information about user's report on exchange
orderrequest	Table which contains member's order request to trade posts

Entity name	Attributes	Description	Null
model	id {pk}	Auto increment unique identifier	No
	code	Model code	No
	name	Model name	No
	productSeriesID	Model's product series	No

	{fk}		
	seriesTitleID {fk}	Model's series title	No
	manufacturerID	Model's manufacturer	Yes
	price	Model's price	Yes
	releasedDate	Model's released date	Yes
	description	Model's description	Yes
	status	Model's status	No
	createdDate	Model's created date	No
productseries	id {pk}	Auto increment unique identifier	No
	name	Unique name of product series	No
seriesTitle	id {pk}	Auto increment unique identifier	No
	name	Unique name of series title	No
manufacturer	id {pk}	Auto increment unique identifier	No
	name	Unique name of manufacturer	No
modelimage	id {pk}	Auto increment unique identifier	No
	imageUrl	Url of images	No
	modelID {fk}	Model which images belong to	No
	typeID {fk}	Type of images	No
imageType	id {pk}	Auto increment unique identifier	No
	name	Unique name of image type	No
account	id {pk}	Auto increment unique identifier	No
	username	Unique username of user	No
	password	Account password to login	No
	roleID {fk}	Account's role	No
	status	Account status	No
role	id {pk}	Auto increment unique identifier	No
	name	Unique name of role	No
proposal	id {pk}	Auto increment unique identifier	No
	accountID {fk}	Proposal owner	No
	eventTitle	Proposal event's title	No
	description	Proposal description	Yes
	location	Proposal event's location	Yes
	prize	Proposal event's prize	Yes
	startDate	Proposal event start date	No
article	id {pk}	Auto increment unique identifier	No
	title	Article title	No

	content	Article content	No
	accountID {fk}	User who posts the articles	No
	date	Article posted date	No
	modifiedDate	Article modified date	Yes
	lastEditor	Article's last editor	Yes
	approvalStatus	Article's approval status	No
	category	Article's category	No
notification	id {pk}	Auto increment unique identifier	No
	description	Notification description	No
	seen	Status seen	No
	objectID	To save id of content object	No
	notificationTypeID {fk}	Type of notification	No
	accountID {fk}	User whom the notification is sent to	No
notificationtype	id {pk}	Auto increment unique identifier	No
	name	Unique name of notification type	No
event	id {pk}	Auto increment unique identifier	No
	title	Event title	No
	location	Event location	No
	content	Event's content	No
	startDate	Event start date	No
	endDate	Event end date	No
	maxAttendee	Event's max attendee	No
	status	Event's status	No
	minAttendee	Event's min attendee	No
	numberOfAttendee	Event's number of attendee, this attribute is to improve query performance	No
	numberOfRating	Event's number of raters, this attribute is to improve query performance	No
	numberOfStars	Event's number of star ratings, this attribute is to improve query performance	No
	ticketPrice	Event's ticket price	No
eventattendee	id {pk}	Auto increment unique identifier	No
	eventID {fk}	ID of event	No
	accountID {fk}	Event's attendee	No
	rating	Event's rating which is evaluated by user	No
	feedback	Feedback of user about event	Yes
	amount	Number of tickets that a user register	No

	date	Date when user registers for event	No
profile	id {pk}	Auto increment unique identifier	No
	email	User's email	No
	firstName	User's firstname	No
	middleName	User's middleName	Yes
	lastName	User's lastname	No
	tel	User's mobile number	Yes
	address	User's address	Yes
	avatar	User's avatar image	Yes
	gender	User's gender	Yes
	birthday	User's birthday	No
	accountID {fk}	Profile owner	No
	numberOfRaters	Number of users who rate this user, this attribute is to improve query performance	No
	numberOfStars	Number of star ratings from users, this attribute is to improve query performance	No
tradepost	id {pk}	Auto increment unique identifier	No
	title	Trade post title	No
	brand	Sale product's brand	No
	model	Sale product name	No
	ownerID {fk}	User who posts this trade post	No
	tradeType	Type of trade post: buyer/seller	No
	condition	Condition of trade post	No
	price	Sale product's price	No
	priceNegotiable	Price that may be negotiable between traders	No
	quantity	Sale product quantity	No
	description	Trade post description	Yes
	postedDate	Date when trade post is created	No
	approvalStatus	Trade post approval status by administrator	No
	location	Trade post location	No
tradereport	lastModified	Trade post's last modified	Yes
	rejectReason	Trade post's reject reason by admin	Yes
	id {pk}	Auto increment unique identifier	No
tradepostimage	tradePostID {fk}	ID of the trade post which is reported by user	No
	reason	Report reason	No
tradepostimage	id {pk}	Auto increment unique identifier	No

	imageUrl	Url of image	No
	tradePostID {fk}	Trade post which images belong to	No
traderating	id {pk}	Auto increment unique identifier	No
	fromUser {fk}	User who evaluates	No
	toUser {fk}	User who is evaluated	No
	feedbackType	To check whether the evaluations is from trade owner to user or users to trade owner	No
	tradePostID {fk}	This attribute is used when user evaluates a trade post	Yes
	rating	Trade post rating	No
	comment	User's feedback about trade post	Yes
orderrequest	id {pk}	Auto increment unique identifier	No
	tradePostID {fk}	The trade post which user request order to	No
	traderID {fk}	User who requests order to trade post	No
	quantity	Product order quantity	No
	billingAddress	Address that is negotiable for exchange	Yes
	status	Order request status	No
	rejectReason	This attribute is used when trade post owner reject traders after accepted their order request	Yes
	cancelReason	This attribute is used when trade post owner cancel user's order request to trade post	Yes
	orderDate	Date of order request	No
	stateSetDate	This attribute is used when trade post owner reject or cancel order request, if reject, this attribute will belong to reject concept. If cancel, this attribute will belong to cancel concept	Yes

3. Performance Measures

Website can locate the user's location in 2 seconds max

4. Test plan

The purpose of this section is to verify and ensure that GWA meets its requirements and design specification. The following part will describe which features to be tested and which will not.

4.1. Features to be tested

Web and mobile application:

- Search models and view detail
- Crawl models, articles
- Proposal to open event
- Register for event
- Post for trade/exchange
- Request order to a trade post

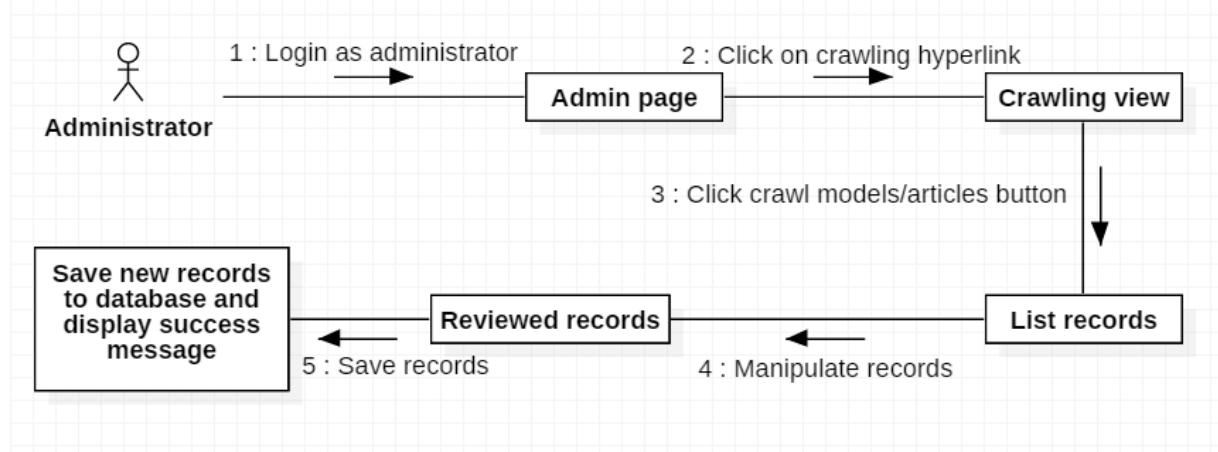
4.2. Features not to be tested

Functions are not listed above won't be tested.

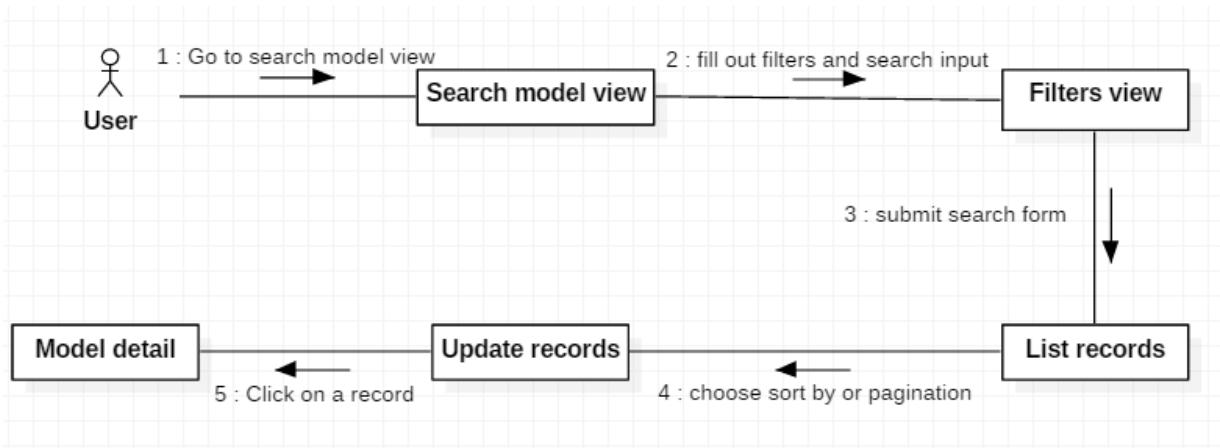
5. System Testing Test Case

5.1. Communication Diagram

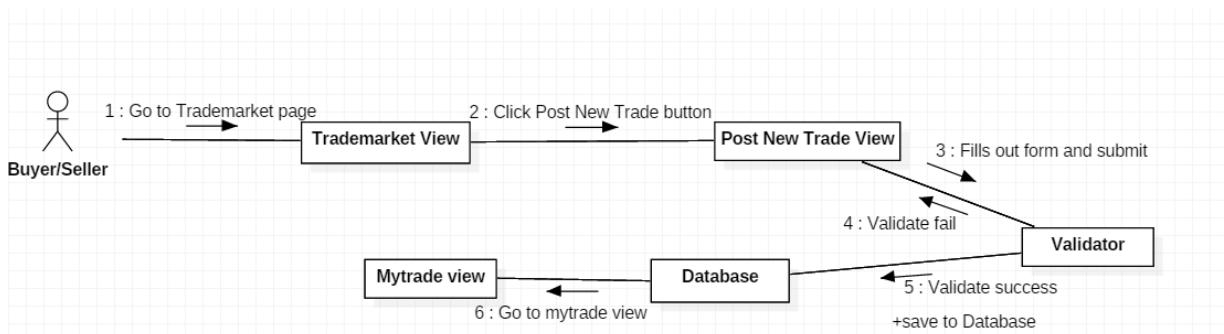
5.1.1. Manipulate crawling



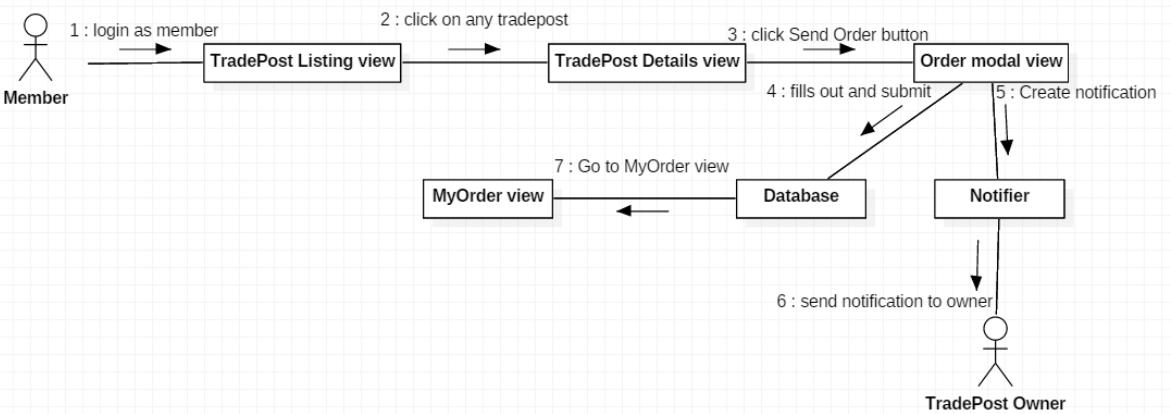
5.1.2. Search for model information



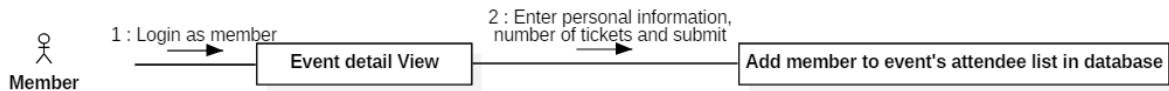
5.1.3. Manipulate trade post



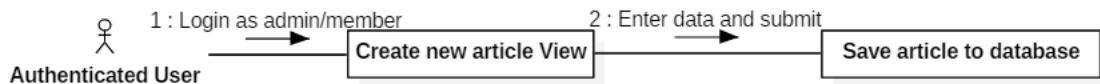
5.1.4. Request order to trade post for exchange



5.1.5. Register for event



5.1.6. Create articles



5.2. Web and Mobile Application Test case

5.2.1. Manipulate crawling

ID	Test Case Description	Test Case Procedure	Expected output	Result	Test Date
AMC 1	Crawl models	1. Login with role “Admin” 2. Go to “Crawling” view. 3. Click “crawl model” or “crawl articles” button.	List of crawling models/articles records is displayed		
AMC 2	Review records	4. CRUD records	The selected record information is updated when admin submit		
AMC 3	Save records	5. Click “SAVE” button	Save new records to database and display success message on		

			the view		
--	--	--	----------	--	--

5.2.2. Search for model information

ID	Test Case Description	Test Case Procedure	Expected output	Result	Test Date
SFM1	Search for model information with filters	<ol style="list-style-type: none"> 1. Go to “Search Model” view. 2. Fill out the search form with filters 3. Click search button 	List of matching records are shown with filters sort by and pagination		
SFM2	Choose filter sorts by and pagination	<ol style="list-style-type: none"> 4. Choose sort by options or pagination 	Updated list records follow the filters and pagination		
SFM3	Show model detail information	<ol style="list-style-type: none"> 6. Click on a record’s “More information” button 	The selected model detail page is shown		

5.2.3. Manipulate trade post

ID	Test Case Description	Test Case Procedure	Expected output	Result	Test Date
MTP1	Post New Trade with valid data.	<ol style="list-style-type: none"> 1. Click “Post new trade” button. 2. Fills out form with valid data and submit 3. Click “Post your trade” button. 	Go to My trade view.		

MTP2	Post New Trade with invalid data.	<ol style="list-style-type: none"> 1. Click “Post new trade” button. 2. Fills out form with invalid data and submit 3. Click “Post your trade” button. 	Stay at Post new trade view and show error message for invalid data.		
------	-----------------------------------	---	--	--	--

5.2.4. Order to trade post for exchange

ID	Test Case Description	Test Case Procedure	Expected output	Result	Test Date
OTT1	Send order with valid quantity	<ol style="list-style-type: none"> 1. Click “Send Order” button on trade post details view. 2. Enter valid quantity 3. fills out Trader information form 4. Click submit 	Go to My order view		
OTT2	Send order with invalid quantity	<ol style="list-style-type: none"> 1. Click “Send Order” button on trade post details view. 2. Enter invalid quantity 3. fills out Trader information form 4. Click submit 	Stay at send order modal view and show error message for invalid quantity..		

5.2.5. Register for event

ID	Test Case Description	Test Case Procedure	Expected output	Result	Test Date
RE1	Register with a valid card	<ol style="list-style-type: none"> 1. Enter card number and all required information 2. Click on “Register” 	Save information to database and a Popup		

			shows up saying “Registration complete”		
RE2	Register with a invalid card	5. Enter card number and all required information 6. Click on “Register”	Popup shows saying “Card is invalid!”		
RE3	Register with the amount of tickets exceeding the remaining tickets available	7. Enter all required information with the number of tickets higher than the amount of remaining tickets for that event 8. Click on “Register”	Popup shows saying The number of tickets you’ve entered exceeds the remaining tickets amount! ”		

5.2.6. Create articles

ID	Test Case Description	Test Case Procedure	Expected output	Result	Test Date
PA1	Post an article	1. Enter data in all fields 2. Click on “Submit”	Save information to database and a Popup shows up saying “Article submitted! Please wait for admin’s approval!”		