

# FPT UNIVERSITY

## Capstone Project Document

### Gunpla World Application

Group 15	
<b>Group Members</b>	Hoàng Ngô Minh Tùng – SE62231 Nguyễn Bảo Khánh – SE61978 Lê Nguyễn Ngọc Sang – SE61647
<b>Supervisor</b>	Phạm Công Thành
<b>Ext Supervisor</b>	N/A
<b>Capstone Project code</b>	gwa

– Ho Chi Minh City, 09/2018 –

# Table Of Contents

Table Of Contents .....	2
List of Tables .....	10
List of Figures .....	10
A. Report No.1: Introduction .....	13
1. Project Information .....	13
2. Introduction .....	13
3. Current Situation .....	14
4. Problem Definition.....	14
5. Proposed Solution .....	14
5.1 Feature functions.....	14
5.2 Advantages and disadvantages.....	15
6. Functional Requirements .....	15
7. Role and Responsibility .....	17
B. Report No.2 Software Project Management Plan.....	17
1. Problem Definition.....	17
1.1. Name of this Capstone Project.....	17
1.2. Problem Abstract.....	17
1.3. Project Overview.....	18
1.3.1. Current Situation .....	18
1.3.2. The Proposed System.....	18
1.3.3. Boundaries of the System.....	20
1.3.4. Future Plans.....	20
1.3.5. Development Environment .....	21
1.3.5.1. Hardware requirements .....	21
1.3.5.2. Software requirements .....	21
2. Project Organization.....	22
2.1. Software Process Model.....	22
2.2. Roles and responsibilities.....	23
2.3. Tools and Techniques .....	24
3. Project Management Plan .....	25

3.1. Software development life cycle .....	25
3.2. Phase Detail.....	27
3.2.1. Requirement Analysis and Technical Research .....	27
3.2.2. System and Software Design .....	27
3.3. All Meeting Minutes .....	30
C. Report No.3 Software Requirement Specification .....	31
1. User Requirement Specification .....	31
1.1. Member Requirement.....	31
1.2. Buyer/Seller Requirement.....	32
1.3. Administrator Requirement.....	32
1.4. Authenticated User Requirement .....	32
1.5. Handler Requirement .....	33
2. System Requirement Specification .....	33
2.1. External Interface Requirement .....	33
2.1.1. User Interface.....	33
2.1.2. Hardware Interface.....	33
2.1.3. Software Interface .....	34
2.1.4. Communication Protocol .....	34
2.2. System Overview Usecase .....	35
2.3. List of Usecase .....	36
2.3.1. < Guest > Overview Usecase .....	36
2.3.1.1. < Guest > Register Usecase.....	37
2.3.1.2. < Guest > Get forgot password Usecase .....	40
2.3.2. < Member > Overview Usecase .....	43
2.3.2.1. < Member > Search model Usecase.....	44
2.3.2.2. < Member > Get model's detail Usecase .....	48
2.3.2.3. < Member > Review model Usecase.....	48
2.3.2.4. < Member > Search article Usecase.....	51
2.3.2.5. < Member > Get article's info Usecase.....	53
2.3.2.6. < Member > Create article Usecase .....	54
2.3.2.7. < Member > Edit article Usecase .....	57
2.3.2.8. < Member > Search event Usecase .....	59
2.3.2.9. < Member > Get event's detail Usecase.....	61

2.3.2.10. < Member > Register event Usecase.....	63
2.3.2.11. < Member > Create proposal Usecase .....	65
2.3.2.12. < Member > Search exchange Usecase.....	67
2.3.2.13. < Member > Get exchange detail Usecase .....	71
2.3.2.14. < Member > Request order Usecase .....	73
2.3.3. < Authenticated User > Overview Usecase.....	77
2.3.3.1. < Authenticated User > Get profile Usecase .....	78
2.3.3.2. < Authenticated User > Edit profile Usecase .....	81
2.3.4. < Administrator > Overview Usecase .....	85
2.3.4.1. < Administrator > Crawl model Usecase .....	86
2.3.4.2. < Administrator > Crawl article Usecase .....	88
2.3.4.3. < Administrator > Add new model Usecase .....	91
2.3.4.4. < Administrator > Edit model Usecase .....	95
2.3.4.5. < Administrator > Delete model Usecase .....	100
2.3.4.6. < Administrator > Start/Stop schedule Usecase .....	102
2.3.4.7. < Administrator > Track schedule progress Usecase .....	106
2.3.4.8. < Administrator > Create new event Usecase .....	108
2.3.4.9. < Administrator > Edit event Usecase.....	111
2.3.4.10. < Administrator > Get proposal's detail Usecase.....	114
2.3.5. < Buyer/Seller > Overview Usecase .....	116
2.3.5.1. < Buyer/Seller > Add trade Usecase .....	116
2.3.5.2. < Buyer/Seller > Edit trade Usecase .....	121
2.3.5.3. < Buyer/Seller > Delete trade Usecase.....	126
2.3.5.4. < Buyer/Seller > Confirm trade Usecase.....	128
2.3.5.5. < Buyer/Seller > Confirm trade success Usecase.....	130
2.3.6. < Handler > Overview Usecase.....	133
2.3.6.1. < Handler > Send notification Usecase .....	133
2.3.6.2. < Handler > Disable error model Usecase .....	135
2.3.6.3. < Handler > Authenticate Usecase .....	137
2.3.6.4. < Handler > Update event Usecase .....	138
2.3.6.5. < Handler > Update order request Usecase .....	140
3. Software System Attribute .....	141
3.1. Usability .....	141

3.2. Reliability.....	142
3.2. Availability.....	142
3.3. Security .....	142
3.4. Maintainability .....	142
3.5. Portability.....	142
3.6. Performance .....	142
4. Conceptual Diagram .....	143
D. Report No. 4 Software Design Description.....	145
1. Design Overview.....	145
2. System Architectural Design.....	146
2.1. Web application architecture description.....	146
3. Component Diagram .....	147
4. Detailed Description .....	150
4.1. Class Diagram.....	150
4.2. Class Diagram Explanation.....	153
4.2.1. Model .....	153
4.2.2. SeriesTitle .....	154
4.2.3. ProductSeries.....	154
4.2.4. Manufacturer .....	155
4.2.5. ModelImage .....	155
4.2.6. ImageType .....	155
4.2.7. ModelRating.....	156
4.2.8. Article.....	156
4.2.9. Proposal.....	157
4.2.10. Account .....	157
4.2.11. Role .....	158
4.2.12. Profile.....	158
4.2.13. Event .....	159
4.2.14. EventAttendee .....	160
4.2.15. Notification .....	160
4.2.16. NotificationType .....	161
4.2.18. OrderRequest.....	162
4.2.19. TradePostImage.....	162

4.2.20. TradeReport.....	163
4.2.21. TradeRating.....	163
4.3. Interaction Diagram .....	164
4.3.1. Sequence Diagram .....	164
4.3.1.1. Create model .....	164
4.3.1.2. Edit model .....	166
4.3.1.3. Search Trade Post.....	167
4.3.1.4. Post new trade .....	168
4.3.1.5. Edit trade post .....	169
4.3.1.6. Request order to a trade post .....	170
4.3.1.7. Approve order request.....	171
4.3.1.8. Confirm trade success .....	172
4.3.1.9. Create new article.....	173
4.3.1.10. Edit article .....	174
4.3.1.11. Create event.....	175
4.3.1.12. Edit event .....	176
4.3.1.13. Register event.....	177
4.3.2. Activity Diagram.....	178
4.3.2.1. Search Gundam .....	178
4.3.2.2. Evaluate Gundam.....	179
4.3.2.3. Update profile.....	180
4.3.2.4. Manage order .....	181
5. Interface .....	182
5.1. Member .....	182
5.1.1. Web application .....	182
5.1.1.1. Display search model view .....	182
5.1.1.2. Display model detail .....	184
5.1.1.3. Display TradePost Listing view .....	187
5.1.1.4. Display TradePost details view .....	190
5.1.1.5. Display MyOrder view.....	192
5.1.1.6. Display article listing view.....	193
5.1.1.7. Article detail view .....	195
5.1.1.8. Edit Article view .....	196

5.1.1.9. Create Article view .....	197
5.1.1.10. Event list view.....	199
5.1.1.11. Event detail view.....	200
5.1.1.12. Create proposal view.....	202
5.1.2. Mobile application .....	204
5.1.2.1. Get model information .....	204
5.1.2.2. Get user information .....	205
5.1.2.3. Get user's exchange evaluation.....	206
5.1.2.4. Get notification.....	206
5.2. Buyer/Seller .....	207
5.2.1. Web application .....	207
5.2.1.1. Display Post New TradePost view.....	207
5.2.1.2. Display My trade view .....	211
5.2.2. Mobile Application .....	213
5.3. Guest .....	213
5.3.1. Web application .....	213
5.3.1.1. Register account.....	213
5.3.1.2. Login .....	215
5.3.2. Mobile application .....	217
5.3.2.1. Login .....	217
5.3.2.2. Register account .....	217
5.4. Authenticated User.....	219
5.4.1. Get profile .....	219
5.4.2. Edit profile .....	221
5.5. Administrator .....	223
5.5.1. Create model .....	223
5.5.2. Edit model .....	225
5.5.3. Delete model .....	227
5.5.4. Manage users.....	228
5.5.5. Crawling.....	229
5.5.6. Manage schedule.....	230
5.5.7. Search article.....	231
5.5.8. Edit article .....	232

5.5.9. Create article .....	234
5.5.10. Search event .....	235
5.5.11. Create event.....	236
5.5.12. Edit event .....	239
5.5.13. List proposal.....	241
5.5.14. Proposal detail.....	242
6. Database Design.....	242
6.1. Entity Relationship Diagram (ERD) .....	243
6.2. Data Dictionary .....	244
7. Algorithms .....	245
7.1. As-you-type search with instant results and pagination with sessionStorage .....	245
7.1.1. Definition .....	245
7.1.2. Define problems .....	245
7.1.3. Solution .....	245
7.1.4. Flow chart .....	247
7.2. MD5 hash function.....	248
7.2.1. Definition .....	248
7.2.2. Define problems .....	248
7.2.3. Solution .....	248
7.3. Schedule Tracking.....	248
7.3.1. Definition .....	248
7.3.2. Define problems .....	248
7.3.3. Solution .....	249
7.4. Search nearby location with Haversine formula .....	250
7.4.1. Definition .....	250
7.4.2. Define problems .....	250
7.4.3. Solution .....	250
E. System Implementation & Test .....	251
1. Introduction.....	251
1.1. Overview .....	251
1.2. Test Approach.....	251
2. Database Relationship Diagram.....	251
2.1. Physical Diagram .....	251

2.2. Data dictionary .....	253
3. Performance Measures .....	258
4. Test plan .....	258
4.1. Features to be tested .....	258
4.2. Features not to be tested .....	259
5. System Testing Test Case .....	259
5.1. Communication Diagram .....	259
5.1.1. Manipulate crawling.....	259
5.1.2. Search for model information .....	260
5.1.3. Manipulate trade post.....	260
5.1.4. Request order to trade post for exchange .....	261
5.1.5. Register for event .....	261
5.1.6. Create article .....	261
5.2. Web or Mobile Application Test case.....	261
5.2.1. Manipulate crawling.....	261
5.2.2. Search for model information .....	262
5.2.3. Manipulate trade post.....	263
5.2.4. Order to trade post for exchange .....	263
5.2.5. Register for event .....	264
5.2.6. Create articles.....	265
F. Software User's Manual .....	265
1. Installation Guide .....	265
1.1. Setting up environment at server side .....	265
1.1.1. Hardware Requirement .....	265
1.1.2. Software Requirement.....	266
1.2. Web Application Deployment Process .....	267
1.3. Mobile Application Deployment Process .....	270
2. User Guide .....	270
2.1. Web Application .....	270
2.1.1. Guest, Member and BuyerSeller .....	270
2.1.1.1. Get user information, edit profile, get exchange evaluation .....	270
2.1.1.2. Get gundam model information .....	272
2.1.1.3. Get article information .....	275

2.1.1.4. Get event information .....	277
2.1.2. Administrator .....	277
2.1.2.1. Manipulate GUNDAM model.....	278
2.1.2.2. Manipulate user.....	279
2.1.2.3. Manipulate crawling.....	280
2.1.2.4. Manipulate schedule.....	280
2.1.2.5. Manipulate article.....	281
2.1.2.6. Manipulate event.....	282
2.1.2.7. Manipulate proposal.....	283
2.2. Mobile Application .....	284
2.2.1. Get user information .....	284
2.2.2. Get model information .....	285

## List of Tables

Table 1 - Role and Responsibility .....	17
Table 2 - Hardware requirement for server.....	21
Table 3 - Hardware requirement for mobile .....	21
Table 4 - Software requirements .....	22
Table 5 - Role and Responsibility in detail.....	24
Table 6 - Tool.....	24
Table 7 - Technique .....	25
Table 8 - Software development life cycle .....	26
Table 9 - Requirement Analysis and technical research .....	27
Table 10 - System and software design.....	28
Table 11 - Implementation and Unit testing .....	29
Table 12 - Integration and System Testing .....	29
Table 13 - Write document .....	30
Table 14 - Component Dictionary.....	149
Table 15 - Class Dictionary.....	153
Table 16 - Entity Data Dictionary.....	244
Table 17 - Physical Diagram Dictionary.....	253

## List of Figures

Figure 1 - Software Process Model.....	23
Figure 2 - System Overview Usecase .....	35

Figure 3 - < Guest > Overview Usecase .....	36
Figure 4 - < Guest > Register Usecase.....	37
Figure 5 - < Guest > Get forgot password Usecase .....	40
Figure 6 - < Member > Overview Usecase .....	43
Figure 7 - < Member > Search model Usecase .....	44
Figure 8 - < Member > Get model's detail Usecase .....	48
Figure 9 - < Member > Search article Usecase .....	51
Figure 10 - < Member > Get article's info Usecase .....	53
Figure 11 - < Member > Create article Usecase .....	54
Figure 12 - < Member > Edit article Usecase .....	57
Figure 13 - < Member > Search event Usecase .....	59
Figure 14 - < Member > Get event's detail Usecase .....	61
Figure 15 - < Member > Register event Usecase .....	63
Figure 16 - < Member > Create proposal Usecase.....	65
Figure 17 - < Member > Search exchange Usecase.....	67
Figure 18 - < Member > Get exchange detail Usecase .....	71
Figure 19 - < Member > Request order Usecase.....	73
Figure 20 - < Authenticated User > Overview Usecase.....	77
Figure 21 - < Authenticated User > Get profile Usecase .....	78
Figure 22 - < Authenticated User > Edit profile Usecase .....	81
Figure 23 - < Administrator > Overview Usecase .....	85
Figure 24 - < Administrator > Crawl model Usecase .....	86
Figure 25 - < Administrator > Crawl article Usecase .....	88
Figure 26 - < Administrator > Add new model Usecase.....	91
Figure 27 - < Administrator > Edit model Usecase .....	95
Figure 28 - < Administrator > Delete model Usecase.....	100
Figure 29 - < Administrator > Start/Stop schedule Usecase .....	102
Figure 30 - < Administrator > Track schedule progress Usecase .....	106
Figure 31 - < Administrator > Create new event Usecase .....	108
Figure 32 - < Administrator > Edit event Usecase.....	111
Figure 33 - < Administrator > Get proposal's detail Usecase.....	114
Figure 34 - < Buyer/Seller > Overview Usecase .....	116
Figure 35 - < Buyer/Seller > Add trade Usecase .....	116
Figure 36 - < Buyer/Seller > Edit trade Usecase.....	121
Figure 37 - < Buyer/Seller > Delete trade Usecase.....	126
Figure 38 - < Buyer/Seller > Confirm trade Usecase.....	128
Figure 39 - < Buyer/Seller > Confirm trade success Usecase .....	130
Figure 40 - < Handler > Overview Usecase.....	133
Figure 41 - < Handler > Send notification Usecase .....	133
Figure 42 - < Handler > Disable error model Usecase.....	135
Figure 43 - < Handler > Authenticate Usecase .....	137
Figure 44 - < Handler > Update event Usecase .....	138
Figure 45 - < Handler > Update order request Usecase .....	140
Figure 46 - Conceptual Diagram.....	143

Figure 47 - System Architectural Design.....	146
Figure 48 - Component Diagram .....	148
Figure 49 - Class Diagram .....	151
Figure 50 - < Sequence Diagram > Create model.....	165
Figure 51 - < Sequence Diagram > Edit model.....	166
Figure 52 - < Sequence Diagram > Search Trade Post .....	167
Figure 53 - < Sequence Diagram > Post new trade.....	168
Figure 54 - < Sequence Diagram > Edit trade post.....	169
Figure 55 - < Sequence Diagram > Request order.....	170
Figure 56 - < Sequence Diagram > Approve order request .....	171
Figure 57 - < Sequence Diagram > Confirm trade success.....	172
Figure 58 - < Sequence Diagram > Create new article .....	173
Figure 59 - < Sequence Diagram > Edit article.....	174
Figure 60 - < Sequence Diagram > Create event .....	175
Figure 61 - < Sequence Diagram > Edit event.....	176
Figure 62 - < Sequence Diagram > Register event .....	177
Figure 63 - < Activity Diagram > Search Gundam.....	178
Figure 64 - < Activity Diagram > Evaluate Gundam.....	179
Figure 65 - < Activity Diagram > Update profile .....	180
Figure 66 - < Activity Diagram > Manage order .....	181
Figure 67 - Entity Relationship Diagram .....	243
Figure 68 - As-you-type search workflow .....	247
Figure 69 - Physical Diagram .....	252
Figure 70 - Manipulate crawling test flow .....	259
Figure 71 - Search gundam test flow .....	260
Figure 72 - Manipulate trade post test flow .....	260
Figure 73 - Request order test flow.....	261
Figure 74 - Register event test flow .....	261
Figure 75 - Create article test flow.....	261

# A. Report No.1: Introduction

## 1. Project Information

- Project name: **Gunpla World Application**
- Project code: **gwa**
- 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:

- Manage and search for gundam information, get user's review on gundam models

- Search for article, write article about gunpla tutorial, assembling or painting guide, sharing experience, hobby, lifestyle...
- Proposals to open events, join events , manage event/contests
- Exchange, trading, find nearby exchange, nearby events, get direction
- Evaluate with star ratings and comment
- Manage and get user's information, statistic, exchange evaluation
- Manage system schedule
- 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 nearby location.
- 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

# **6. Functional Requirements**

## **a. Gunpla model:**

### User component:

- Search for gunpla model's information
- Review with star rating and comment

### Administrator component:

- Crawl data, review records, approve pending
- Manipulate gunpla model's information

**b. Information article (tutorial, sharing experience):**

User component:

- Search for information article
- Manipulate the user's own information article

Administrator component:

- Crawl data, review records, approve pending
- Manipulate information article
- Verify user's pending article

**c. Event:**

User component:

- Search for events, find nearby events
- Proposals to open events
- Register events/contests
- Evaluate by star ratings & feedback for events

Administrator component:

- Get user's proposal
- Manage events

**d. Exchange:**

User component:

- Post for exchange (buy/sell), request order
- Evaluate by star ratings & feedback for the exchange
- Manipulate user's own exchange
- Search for exchange, find nearby exchange, get direction
- Report the exchange post

Administrator component:

- Verify user's pending trade post request

**e. User:**

User component:

- Update profile, get user's info with statistic
- Get notification
- Get exchange evaluation

Administrator component:

- Manage account, ban/unban account

**f. Report:**

- List gundam models, articles, events, exchange,crawl records and users with filters
- Count user's proposals, exchange posts, rating evaluation
- Top 10 best rating user, Top 5 best rating model

## 7. Role and Responsibility

No	Full Name	Role	Position	Contact
1	Phạm Công Thành	Project Manager	Supervisor	thanhpcc3@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

Table 1 - Role and Responsibility

# 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.

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 user (customer):
  - ❖ User can register a new account or get forgot password.
  - ❖ User with an account can get any user's info and exchange evaluation
  - ❖ User can get Gundam info with user's review evaluation.
  - ❖ User can read any articles posted.
  - ❖ User can read any events posted, join event
  - ❖ User with an account can write an event proposal.
  - ❖ User with an account can write new articles and post them, the article will need admin approval to be posted.
  - ❖ User can get the exchange posts, request order.
  - ❖ 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 can report an exchange post to the admin
- For administrator:
  - ❖ Administrator can manipulate Gundam info
  - ❖ Administrator can manipulate the crawling (gundam and article), approve pending crawl data, track crawl process
  - ❖ Administrator can manipulate users.
  - ❖ Administrator can manipulate article, verify user' pending article
  - ❖ Administrator can manipulate event, user's proposal
  - ❖ Administrator can verify exchange post request

- ❖ Administrator can manage schedule, track schedule progress

Besides above, the system can also get the location by address to search nearby exchange, events, send notification to user, execute schedule, send email with forgot password.

### **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 against this problem.
  - + System will limit register account by ip address to avoid spamming.

### **1.3.5. Development Environment**

#### **1.3.5.1. Hardware requirements**

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

**Table 2 - Hardware requirement for server**

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

**Table 3 - Hardware requirement for mobile**

#### **1.3.5.2. Software requirements**

<b>Software</b>	<b>Name and version</b>	<b>Description</b>
<b>Operating System</b>	Window 10 1803	Operating system and platform for development
<b>Environment</b>	Java EE 8, Android SDK APIs level 15 or above	Specification for developing web and android application
<b>Modeling Tool</b>	StarUML 2.8.1	Used to implement website and web service

<b>IDE</b>	IntelliJ IDEA 2018.2.3, Android Studio 3.1.3	Programming tools
<b>DBMS</b>	MySQL 5.7.20, MySQL Workbench 6.3.9 CE	Used to create & manage the database for system
<b>Source control</b>	SourceTree 2.6.10	Used for source control
<b>Web browser</b>	Chrome 62 or above	Testing browser

Table 4 - Software requirements

## 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.

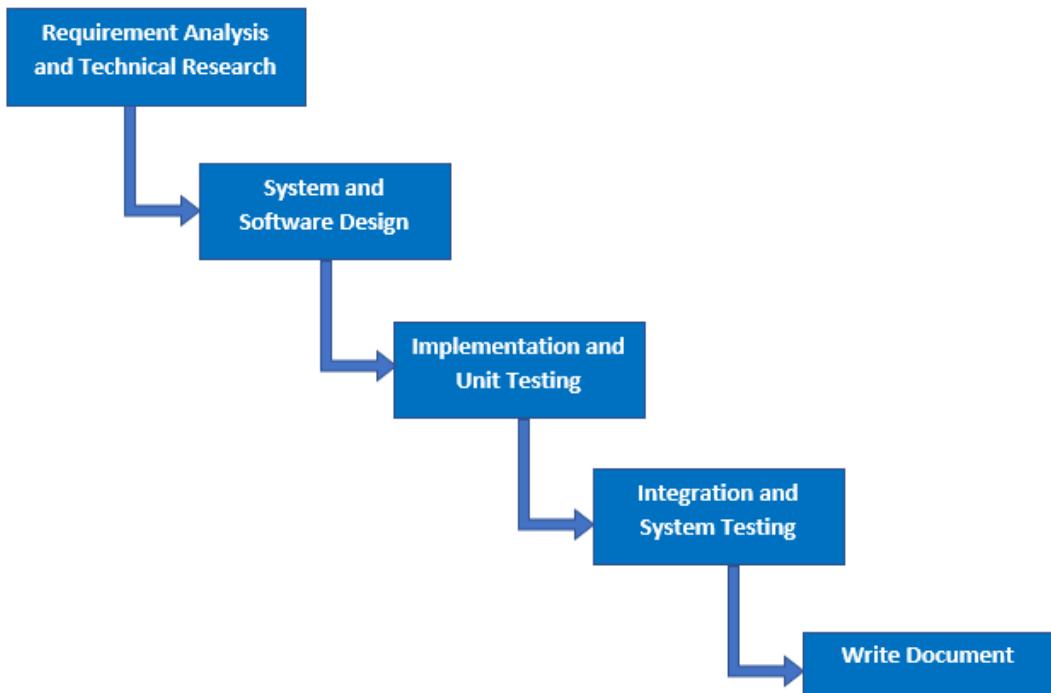


Figure 1 - Software Process Model

## 2.2. Roles and responsibilities

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

			<ul style="list-style-type: none"> <li>Clarify requirements, analyze business</li> <li>Design project architect</li> <li>Coding</li> <li>Testing</li> </ul>
4	Lê Nguyễn Ngọc Sang	Team Member, QC, DEV, Tester	<ul style="list-style-type: none"> <li>Create test plan</li> <li>GUI design</li> <li>Design database</li> <li>Clarify requirements, analyze business</li> <li>Design project architect</li> <li>Coding</li> <li>Testing</li> </ul>

Table 5 - Role and Responsibility in detail

### 2.3. Tools and Techniques

Tool	Name and version
IDE	<ul style="list-style-type: none"> <li>- IntelliJ IDEA 2018.2.3</li> <li>- Android Studio 3.1.3</li> </ul>
Source control	SourceTree 2.6.10
Database Manager	MySQL Workbench 6.3.9 CE

Table 6 - Tool

Technique	Name and version
Front end	<ul style="list-style-type: none"> <li>HTML v5.0</li> <li>JQuery v3.2.1</li> <li>Bootstrap v3.3.7</li> <li>CSS</li> <li>Javascript</li> <li>Ajax</li> </ul>
Back end	<ul style="list-style-type: none"> <li>Java EE 8</li> <li>Spring Boot v2.0.5 RELEASE</li> <li>Hibernate v5.0.12</li> <li>RESTful Web Service</li> </ul>

Google API libraries	<ul style="list-style-type: none"> <li>• Google Maps API</li> <li>• Firebase Cloud Messaging</li> </ul>
Web server	Apache Tomcat v8.5.15
Database system	MySQL 5.7.20

Table 7 - Technique

### 3. Project Management Plan

#### 3.1. Software development life cycle

Phase	Description	Deliverables	Resource needed	Dependencies and Constraints	Risks
<b>Requirement analysis and technical research</b>	<ul style="list-style-type: none"> <li>- Identify and clarify requirements for system</li> <li>- Research for google maps api and crawling technique (StAX), FCM (firebase cloud messaging)</li> </ul>	<ul style="list-style-type: none"> <li>- Software Requirement Specification</li> <li>- Basic knowledge Google Maps API, FCM</li> <li>- Understand about StAX technique</li> <li>- Introduction document</li> <li>- Project Management Plan</li> <li>- Prototypes</li> </ul>	4 weeks	N/A	<ul style="list-style-type: none"> <li>- Lack of member share of understand</li> <li>- SRS does not match user requirement</li> <li>- Misunderstanding of learning</li> </ul>
<b>System and software design</b>	<ul style="list-style-type: none"> <li>- Architect design for the system</li> </ul>	<ul style="list-style-type: none"> <li>- Software Design Description</li> </ul>	2 weeks	Depend on “Requirement Analysis and	- Lack of experience

	<ul style="list-style-type: none"> <li>- Detail design using top-down break down</li> </ul>	<p>Document</p> <ul style="list-style-type: none"> <li>- Base code structure</li> <li>- Technology notes</li> </ul>		<p>Technical Research”</p>	<ul style="list-style-type: none"> <li>- Diagrams don’t match SRS</li> <li>- Not fulfil requirement</li> </ul>
<b>Implementation and unit testing</b>	<ul style="list-style-type: none"> <li>- Implement the system</li> <li>- Crawling data</li> <li>- Building mobile application</li> <li>- Prepare test plan and test cases</li> <li>- Unit testing of the code</li> </ul>	<ul style="list-style-type: none"> <li>- System Implementation and Test Document</li> <li>- Programs (web and mobile app)</li> <li>- Unit test cases and results</li> </ul>	4 weeks	<p>Depend on “System and Software Design”</p>	<ul style="list-style-type: none"> <li>- Delay in implementation</li> <li>- Inaccurate programming</li> </ul>
<b>Integration and system testing</b>	<ul style="list-style-type: none"> <li>- Perform functional and non functional testing</li> <li>- Make sure the system meets the requirements</li> </ul>	<ul style="list-style-type: none"> <li>- Test report</li> <li>- Defect report</li> </ul>	1 week	<p>Depend on “Implementation and Unit Testing”</p>	<ul style="list-style-type: none"> <li>- Missing of functions that need to be tested</li> </ul>
<b>Write document</b>	<ul style="list-style-type: none"> <li>- Write “Software User’s Manual”</li> <li>- Review all documents</li> <li>- Write paper</li> </ul>	<ul style="list-style-type: none"> <li>- Software User’s Manual</li> <li>- A paper</li> </ul>	3 weeks	<p>Depend on “System Design” and “Implementation”</p>	<ul style="list-style-type: none"> <li>- Don’t have enough time to complete paper.</li> </ul>

Table 8 - Software development life cycle

## 3.2. Phase Detail

### 3.2.1. Requirement Analysis and Technical Research

Task	Description	Author
<b>Identify and clarify functions</b>	<p>Identify and clarify user requirements</p> <p>Divide all functions into groups by concept</p>	TungHNM, KhanhNB, SangLNN
<b>Research on Google Maps and Location, Firebase Cloud Messaging, Spring Boot email</b>	<p>Research on google maps api</p> <p>Research on HTML5 Location, Android Location</p> <p>Research on Firebase Cloud Messaging, Spring Boot email</p> <p>Research on StAX crawling technique</p>	TungHNM, KhanhNB, SangLNN
<b>Design application</b>	<p>Design GUI for web and mobile application base on main functions</p> <p>Design Use Case Diagram</p>	TungHNM, KhanhNB, SangLNN
<b>Write document</b>	<p>Write Introduction document</p> <p>Write Project Management Plan</p> <p>Write Software Requirement Specification</p>	TungHNM, KhanhNB, SangLNN

Table 9 - Requirement Analysis and technical research

### 3.2.2. System and Software Design

Task	Description	Author

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

Table 10 - System and software design

### 3.2.3. Implementation and Unit Testing

Task	Description	Author
<b>Implement Web Service</b>	Implement main functions  Implement REST api	TungHNM, KhanhNB, SangLNN
<b>Implement GUI</b>	Implement Web and Mobile UI	TungHNM, KhanhNB, SangLNN
<b>Implement Algorithms</b>	Implement nearby location, auto-complete address input  Implement pagination with sessionStorage  Implement data crawling  Implement as-you-type search with string comparison and localStorage  Implement schedule  Implement MD5 hash function for matching data	TungHNM, KhanhNB, SangLNN

<b>Unit testing</b>	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
<b>Write document</b>	Write System Implementation and Test Document	TungHNM, KhanhNB, SangLNN

Table 11 - Implementation and Unit testing

### 3.2.4. Integration and System Testing

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

Table 12 - Integration and System Testing

### 3.2.5. Write document

Task	Description	Author
<b>Write document</b>	Write Software User's Manual Document	TungHNM, KhanhNB, SangLNN
<b>Review all documents</b>	Review 6 reports	TungHNM, KhanhNB, SangLNN
<b>Write paper</b>	Write abstract and introduction  Write about As-you-type search and pagination with sessionStorage	TungHNM, KhanhNB, SangLNN

	Write comparison  Write conclusion	
--	------------------------------------------	--

Table 13 - Write document

### 3.3. All Meeting Minutes

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

#### 1. 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, get review evaluation
- Search for articles, get articles's information
- Manipulate article
- Search for events, get events's detail

- Search for exchange, get exchange's detail
- Register event
- Proposal to open events
- 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:

- Manage users
- Manage crawling, includes: crawl models and information articles
- Approve pending crawl records
- Manage gunpla model information system
- Manage article
- Approve requests to publish articles, and to post for exchange
- Manage events
- Manage schedule (crawling schedule, update event & update trade schedule)

## **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
- Get profile (includes get exchange evaluation)
- Get notification

## **1.5. Handler Requirement**

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

- Send notifications
- Auto disable error models (error loading 404 image)
- Authenticate and authorize
- Auto update event (which don't have enough attendee after registration date)
- Auto update order request (which don't have any responses from trade post owner in 2 days)

# **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, 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

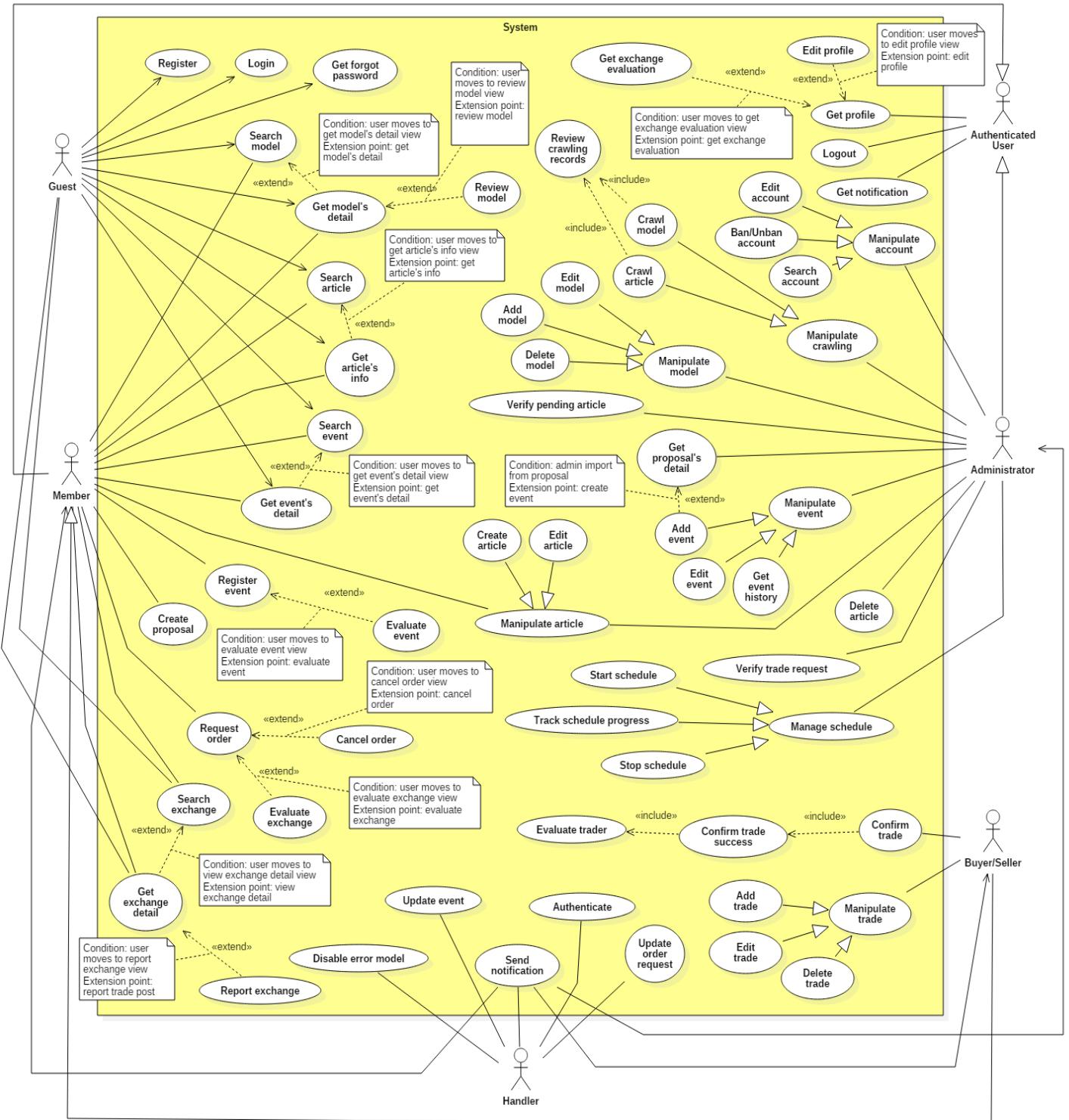


Figure 2 - System Overview Usecase

## 2.3. List of Usecase

### 2.3.1. < Guest > Overview Usecase

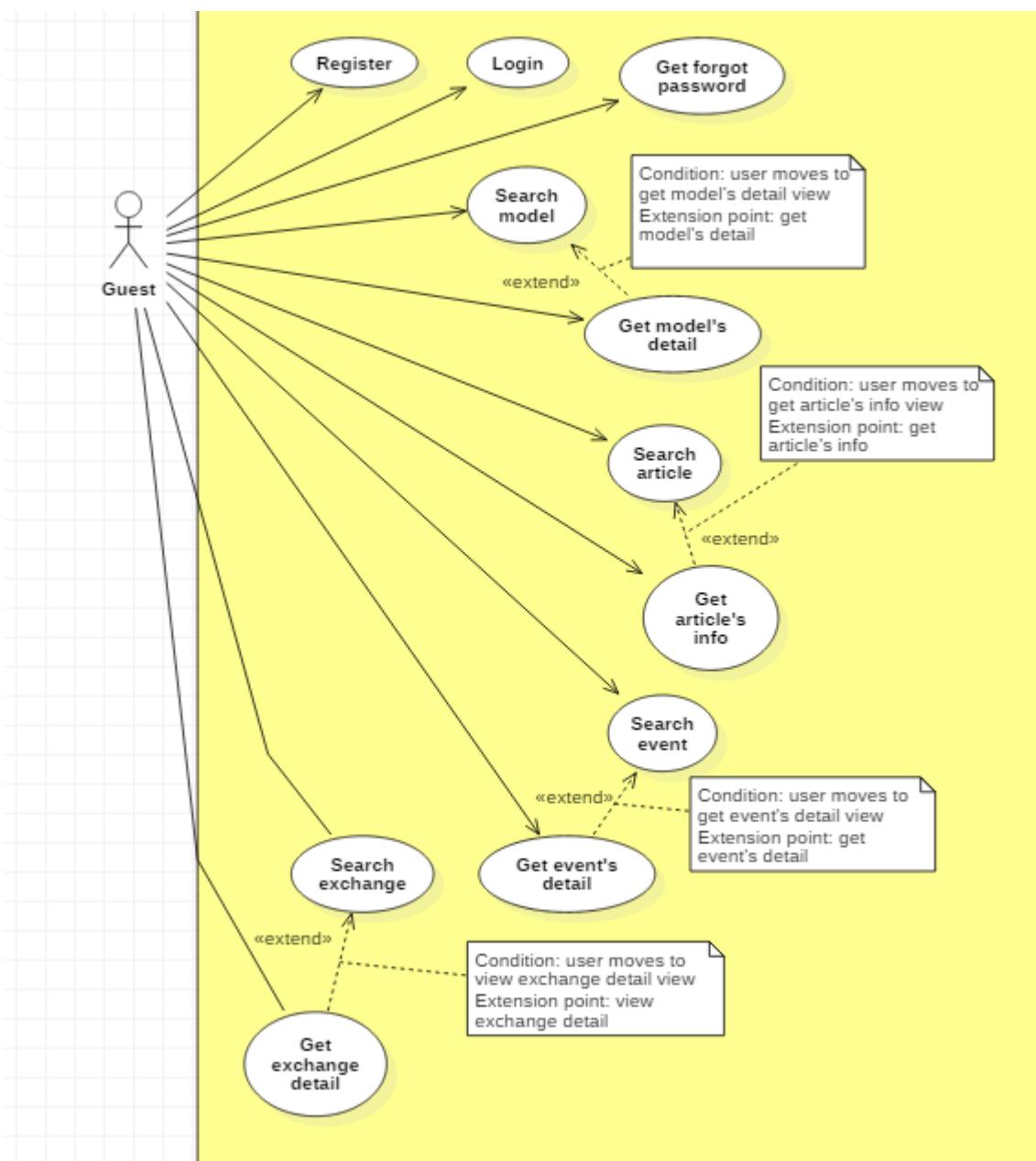


Figure 3 - < Guest > Overview Usecase

### 2.3.1.1. < Guest > Register Usecase



**Figure 4 - < Guest > Register Usecase**

USE CASE – UC_Guest_01			
<b>Usecase No.</b>	Guest_01	<b>Usecase Version</b>	1.0
<b>Usecase Name</b>	Register		
<b>Author</b>	TungHNM		
<b>Date</b>	03/10/2018	<b>Priority</b>	Normal
<b>Actor:</b>	<ul style="list-style-type: none"> <li>- Guest.</li> </ul>		
<b>Summary:</b>	<ul style="list-style-type: none"> <li>- This use case allows guest to register to system.</li> </ul>		
<b>Goal:</b>	<ul style="list-style-type: none"> <li>- System can recognize the new account.</li> <li>- User can login the system with role “Member”</li> </ul>		
<b>Triggers:</b>	<ul style="list-style-type: none"> <li>- Guest sends register command to the system.</li> </ul>		
<b>Preconditions:</b>			

- Register view is displayed.

**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 goes to register view	<p>Register view is shown with following labels and fields:</p> <ul style="list-style-type: none"> <li>• Username: text, required, max length: 50</li> <li>• Email: text, required, max length: 50, match: /^[a-z0-9_.-]+@[\\da-z_.-]+\.(a-z_.){2,6}\$/</li> <li>• Password: free text input, required, max length: 50</li> <li>• Confirm password: free text input, required</li> <li>• First Name: text, max length: 50</li> <li>• Last Name: text, max length: 50</li> </ul>
2	Guest fills out the form	
3	Guest clicks on “SAVE” button	<p>Validate data. [Exception 1, 2, 3, 4, 5, 6]</p> <p>New account is added to system.</p> <p>System displays dialog with 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-zA-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.
- In case of success scenario, the account is created with default role: Member and status: Active

### 2.3.1.2. < Guest > Get forgot password Usecase

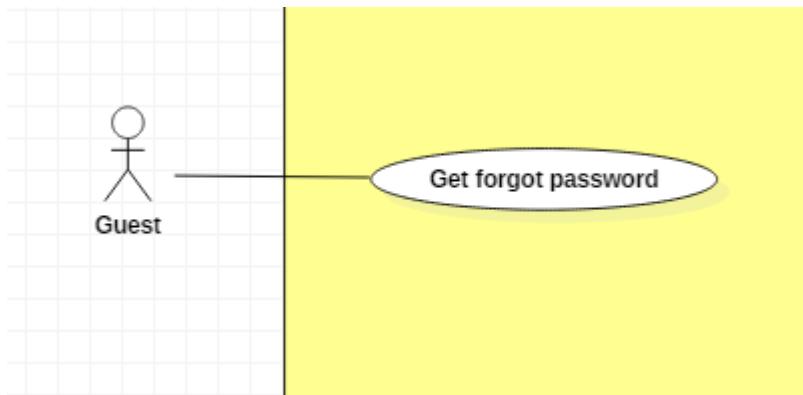


Figure 5 - < Guest > Get forgot password Usecase

USE CASE – UC _Guest_02			
Usecase No.	Guest_02	Usecase Version	1.0
Usecase Name	Get forgot password		
Author	TungHNM		
Date	03/10/2018	Priority	Normal
<b>Actor:</b>			
- Guest.			
<b>Summary:</b>			
- This use case allows guest who has already registered to get forgot password.			
<b>Goal:</b>			
- Registered user is able to get forgot password			
<b>Triggers:</b>			
- Guest sends get forgot password command to the system.			
<b>Preconditions:</b>			
- Get forgot password view is displayed.			
<b>Post Conditions:</b>			
- Success: Registered user receives email with their account's password.			

- **Fail:** Show error message.

#### Main Success Scenario:

Step	Actor Action	System Response
1	Guest input email and submit	<p>Validate data.</p> <p>[Exception 1]</p> <p>Registered user receives email with his/her password</p> <p>System displays message: "We have sent your password to your email. Please check it, thanks !"</p> <p>[Alternative 1]</p>

#### Alternative Scenario:

**Alternative Scenario 1:** The input email does not exist

Step	Actor Action	System Response
1	Guest input email and submit	System displays message: "This email does not exist"

#### Exceptions:

No	Cause	System Response
1	Entered email does not match the required format: /^(a-z0-9_\.]+)@([\\da-zA-Z_\.]+)\.([a-zA-Z_\.]{2,6})\$/	System shows error message: Invalid email !

**Relationships:** N/A

**Business Rules:**

- In case of success scenario, system will send email with password to registered's email

### 2.3.2. < Member > Overview Usecase

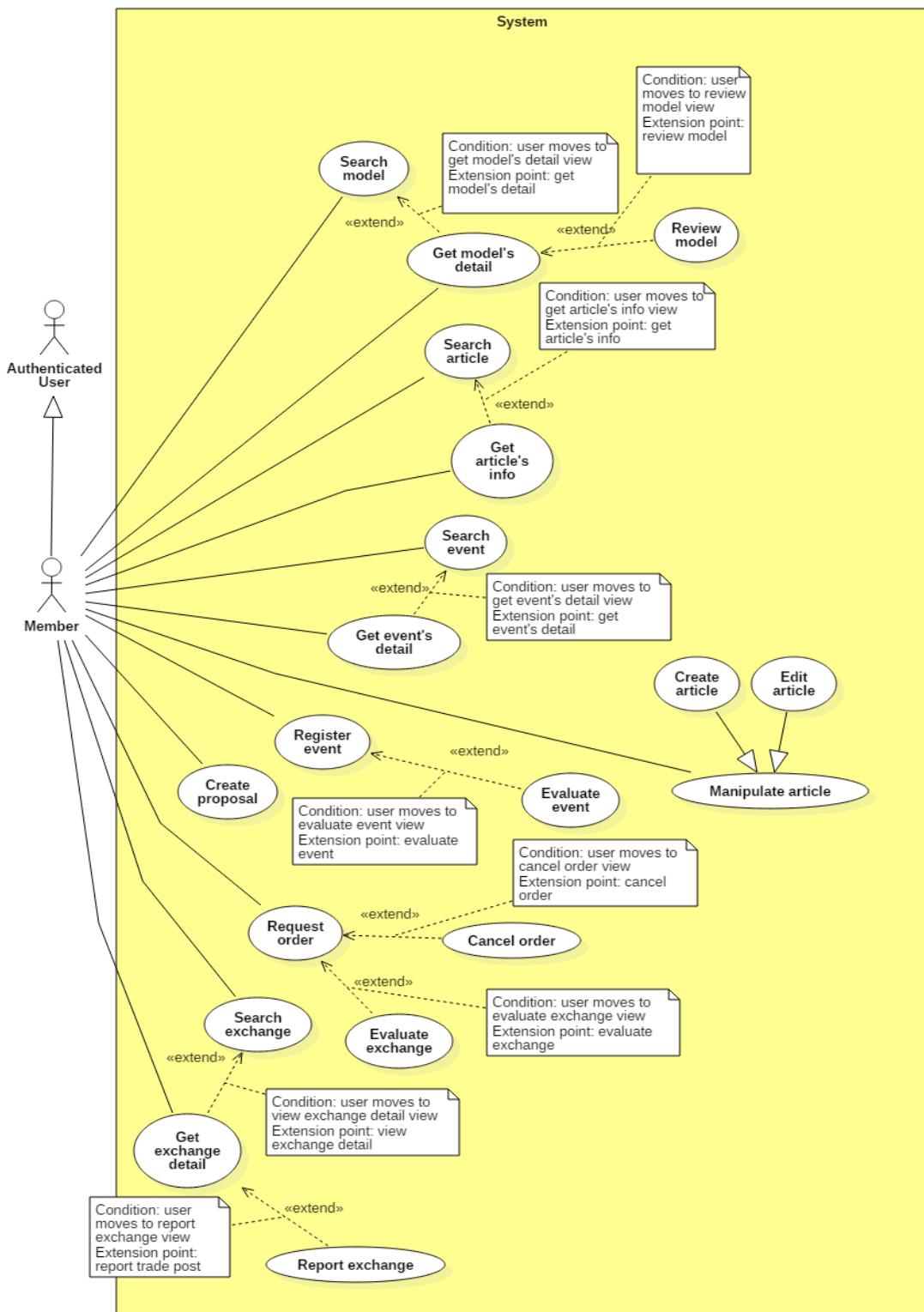


Figure 6 - < Member > Overview Usecase

### 2.3.2.1. < Member > Search model Usecase

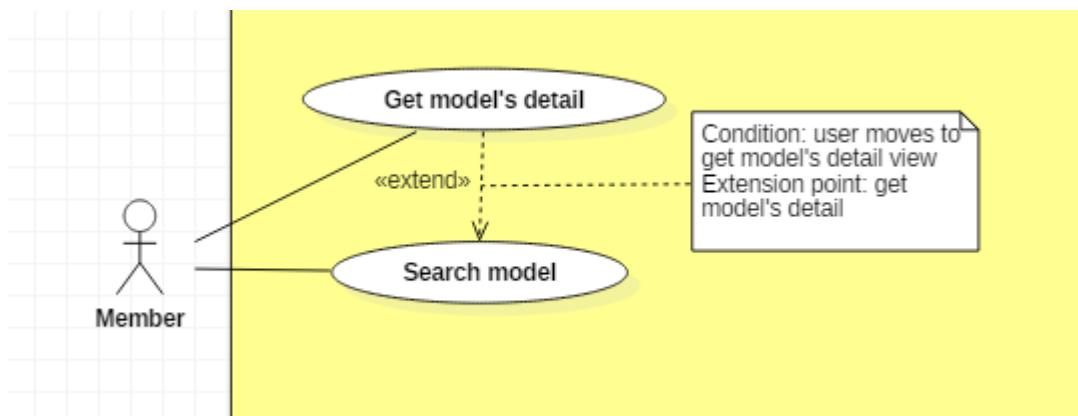


Figure 7 - < Member > Search model Usecase

USE CASE – UC _Member_01			
<b>Usecase No.</b>	Member_01	<b>Usecase Version</b>	1.0
<b>Usecase Name</b>	Search model		
<b>Author</b>	TungHNM		
<b>Date</b>	03/10/2018	<b>Priority</b>	Normal
<b>Actor:</b>	<ul style="list-style-type: none"> <li>- Member.</li> </ul>		
<b>Summary:</b>	<ul style="list-style-type: none"> <li>- This use case allows member to search model.</li> </ul>		
<b>Goal:</b>	<ul style="list-style-type: none"> <li>- Make it more convenient for members to seek for gunpla information.</li> </ul>		
<b>Triggers:</b>	<ul style="list-style-type: none"> <li>- Member sends search model command to the system.</li> </ul>		
<b>Preconditions:</b>	<ul style="list-style-type: none"> <li>- The search model page is displayed.</li> </ul>		
<b>Post Conditions:</b>	<ul style="list-style-type: none"> <li>- <b>Success:</b> A list of matching records are shown with pagination and orderBy filter.</li> <li>- <b>Fail:</b> Show error message.</li> </ul>		

**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"> <li>• Search navbar: free text input</li> <li>• Icon search: button</li> <li>• Filter series title: dropdown list</li> <li>• Filter product series: dropdown list</li> <li>• Filter manufacturer: dropdown list</li> <li>• Filter price: dropdown list, includes following values:           <ul style="list-style-type: none"> <li>○ All</li> <li>○ &lt; 5000 yen</li> <li>○ 5000 yen to 10000 yen</li> <li>○ &gt; 10000 yen</li> </ul> </li> </ul> <p>System displays all available gundam models in database with pagination and filter OrderBy</p> <p>System loads all product series to filter product series, also include option “All”</p> <p>System loads all series title to filter series title, also include option “All”</p> <p>System loads all manufacturers to filter manufacturer, also include option “All”</p>
2	Member clicks on search navbar	An text input is displayed with border blue and placeholder: Please input keyword

3	Member fills out the input text and filters form  [Alternative 1]	System will display a list with instant results which match or nearly match the input keyword, includes thumbImage of gundam model and name
4	Member clicks on search icon button or press Enter	A list of matching gundam models by name, product series, series title, manufacturer, price based on the filters will be displayed on the view with pagination and sorting filter.  [Alternative 2]
5	Member hovers on a record	Brief detail information will be displayed on the view
6	Member clicks on More information button on each records	System redirects to the clicked gundam model's detail page
7	Member chooses an option in OrderBy dropdown list	System will sort the current matching records by OrderBy filter's value (price, createdDate and rating)

#### **Alternative Scenario:**

##### **Alternative Scenario 1: Click on an item in instant results list**

Step	Actor Action	System Response

1	Member clicks on an item in instant results list	System redirects to the clicked gundam model's detail page
---	--------------------------------------------------	------------------------------------------------------------

### Alternative Scenario 2: No matching records

<b>Step</b>	<b>Actor Action</b>	<b>System Response</b>
1	Member clicks on search icon button or press enter	There is no matching records and system displays message No matching records !

**Exceptions:** N/A

**Relationships:** N/A

**Business Rules:**

- When user input a keyword in search bar, system will display all the match or nearly match model's name on the view with thumb image
- When user goes to search page, all gundam model's name and thumb image url in database response from server will be saved in localStorage with datetime
- Every next time user goes to search page, system will check the datetime of the last saving in localStorage and compares to current datetime, if larger or equal to 4 days, then ajax will request to server and update data
- Each page displays a maximum of 24 gundam records
- When user clicks on a page, data of that page will be saved to sessionStorage, and also 2 pages next by
- When user clicks on a page, system will check data of that page and also 2 pages nextby whether it is saved in sessionStorage or not, if saved then get it from sessionStorage and display to view, if not then request to server and get data

### 2.3.2.2. < Member > Get model's detail Usecase

### 2.3.2.3. < Member > Review model Usecase

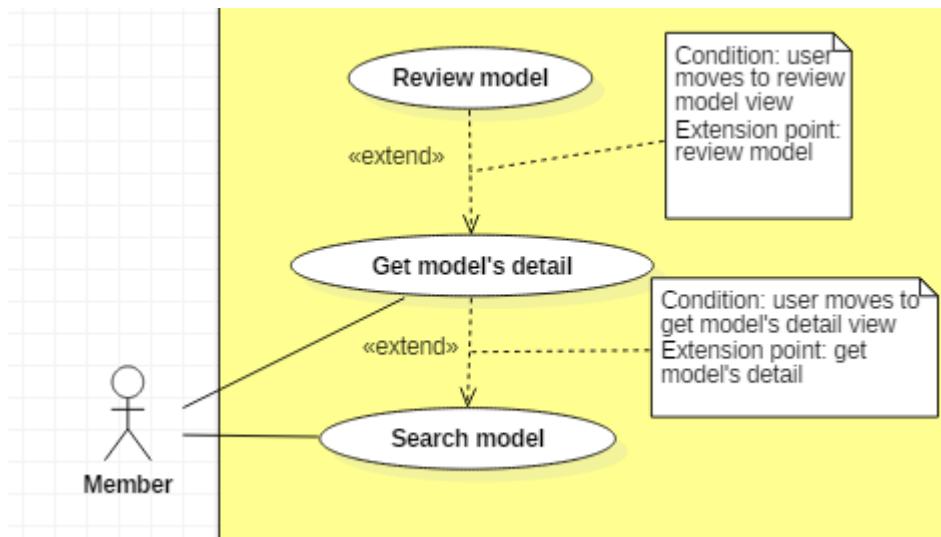


Figure 8 - < Member > Get model's detail Usecase

USE CASE – UC _Member_02, USE CASE – UC _Member_03			
Usecase No.	Member_02, Member_03	Usecase Version	1.0
Usecase Name	Get model's detail, Review model		
Author	TungHNM		
Date	03/10/2018	Priority	Normal
<b>Actor:</b>			
- Member.			
<b>Summary:</b>			
- This use case allows users to get model's information detail.			
<b>Goal:</b>			
- Member can get information from models.			
<b>Triggers:</b>			
- 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"> <li>● Big image: img</li> <li>● Manufacturer: text</li> <li>● Scale: text</li> <li>● Series: text</li> <li>● Released Date: text</li> <li>● Price: text</li> <li>● Rating: stars image</li> <li>● View review evaluation icon: button</li> <li>● Description: text</li> <li>● My rating: button</li> </ul> <p>System displays a slideshow of images divided into following categories:</p> <ul style="list-style-type: none"> <li>● Package</li> <li>● Item picture</li> <li>● Other picture</li> <li>● Contents</li> <li>● About item</li> <li>● Color</li> <li>● Assembly guide</li> </ul> <p>System displays Top 5 rating gundam models on the right navbar</p> <p>System displays gundam model's related</p>

		article and related trade post (exchange) on the right navbar
2	Member clicks on view review evaluation icon button	System displays gundam model's all user's review evaluation with pagination by scrolling to bottom (each page has maximum 10 records)
3	Member clicks on my rating button	System displays evaluation form with star ratings and review comment (required)
4	Member fills out the form and click submit	Validate [Exception 1] System displays message "Submit successfully" and remove my rating button

**Alternative Scenario:** N/A

**Exceptions:**

No	Cause	System Response
1	Missing required fields	System display error message to the view for member to input required fields

**Relationships:** Use case Get model's detail extends to Get Search model use case, Use case Review model extends to Get model's detail use case

**Business Rules:**

- Related article is found by matching article's title to current model's name

- Related trade post is found by percentage string comparison between trade post's model name and current model's name which have to be larger than or equal to 70%
- Pagination in list user's review evaluation will be divided to maximum 10 records for each page and will continue to load if user scroll to bottom
- Each member can only evaluate gundam model one time

#### 2.3.2.4. < Member > Search article Usecase

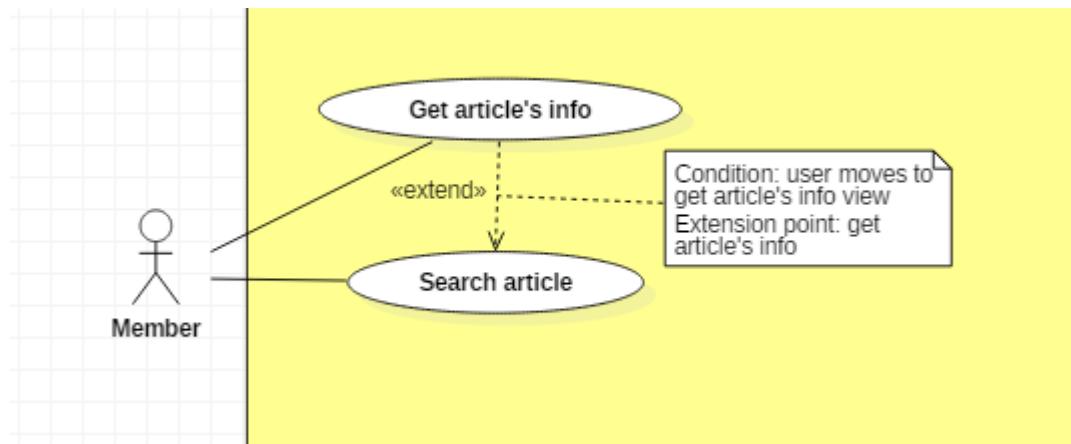


Figure 9 - < Member > Search article Usecase

USE CASE – UC _Member_04			
Usecase No.	Member_04	Usecase Version	1.0
Usecase Name	Search article		
Author	Khanhnb		
Date	03/10/2018	Priority	Normal
<b>Actor:</b>	<ul style="list-style-type: none"> <li>- Member.</li> </ul>		
<b>Summary:</b>	<ul style="list-style-type: none"> <li>- This use case allows member to search articles.</li> </ul>		
<b>Goal:</b>	<ul style="list-style-type: none"> <li>- Make it more convenient for guest to seek for article.</li> </ul>		

**Triggers:**

- Member sends search news command to the system.

**Preconditions:**

- The search article 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 article view	<p>System displays search article view with the following elements:</p> <ul style="list-style-type: none"><li>• Search bar: free text input</li><li>• Dropdown list: select from a list of article's category, default selection is "All category"</li><li>• Dropdown list: select sort by newest or oldest articles</li><li>• SEARCH BY: article's title</li></ul>
2	Member fills out the input text, select category and sort type	
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

### 2.3.2.5. < Member > Get article's info Usecase

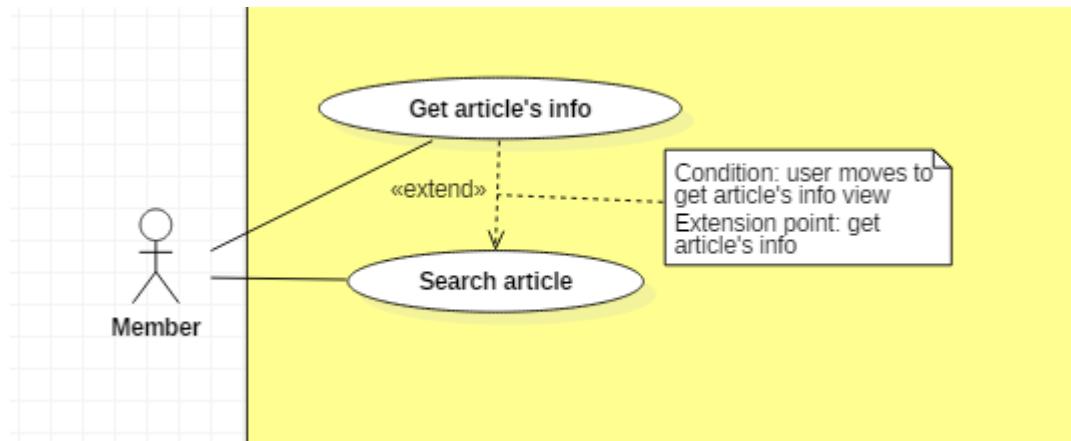


Figure 10 - < Member > Get article's info Usecase

USE CASE – UC _Member_05			
Usecase No.	Member_05	Usecase Version	1.0
Usecase Name	Get article's info		
Author	Khanhnb		
Date	03/10/2018	Priority	Normal
<b>Actor:</b>	<ul style="list-style-type: none"> <li>- Member.</li> </ul>		
<b>Summary:</b>	<ul style="list-style-type: none"> <li>- This use case allows member to get an article's info.</li> </ul>		
<b>Goal:</b>	<ul style="list-style-type: none"> <li>- Display an article's content</li> </ul>		
<b>Triggers:</b>	<ul style="list-style-type: none"> <li>- Member sends get news's detail command to the system.</li> </ul>		

**Preconditions:**

- Member select a article from the list of articles.

**Post Conditions:**

- **Success:** The matching record is shown.
- **Fail:** Show error message.

**Main Success Scenario:**

Step	Actor Action	System Response
1	Member clicks on an article and goes to article detail view	<p>System displays news's detail view with the following elements:</p> <ul style="list-style-type: none"><li>• The title of the article: text</li><li>• The date the article was posted on</li><li>• The username of the article's author</li><li>• The article's content</li><li>• (if the member posted this article) “Edit article” button</li></ul>

**Alternative Scenario:** N/A

**Exceptions:** N/A

**Relationships:** N/A

**Business Rules:** N/A

### 2.3.2.6. < Member > Create article Usecase

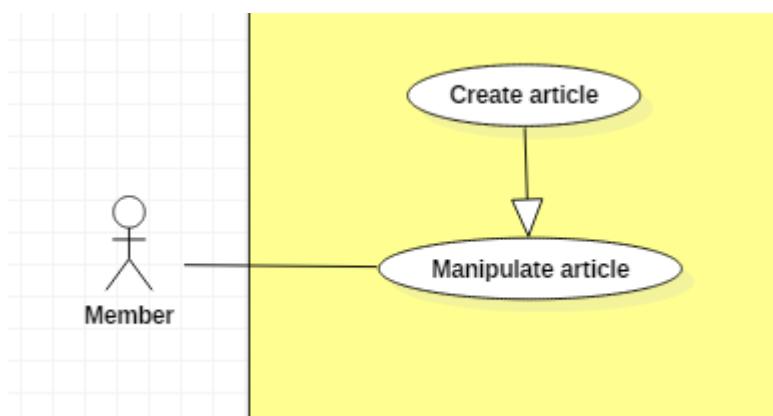


Figure 11 - < Member > Create article Usecase

<b>USE CASE – UC _Member_06</b>			
<b>Usecase No.</b>	Member_06	<b>Usecase Version</b>	1.0
<b>Usecase Name</b>	Create article		
<b>Author</b>	KhanhNB		
<b>Date</b>	25/09/2018	<b>Priority</b>	Normal
<b>Actor:</b> - Member <b>Summary:</b> - This use case allows user to create a new article. <b>Goal:</b> - A new article added to the database system. <b>Triggers:</b> - Member write an articles and submit it. <b>Preconditions:</b> - User must login into system with role Member. <b>Post Conditions:</b> - <b>Success:</b> New article is added to the database system. - <b>Fail:</b> Show error message. <b>Main Success Scenario:</b>			
<b>Step</b>	<b>Actor Action</b>	<b>System Response</b>	
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"> <li>● Input field for article’s Title</li> <li>● Dropdown list to select article’s category</li> <li>● A textarea for the article’s description</li> <li>● An image upload button to upload the article’s thumbnail</li> <li>● An empty image to preview the thumbnail</li> </ul>	

		<ul style="list-style-type: none"> <li>• A textarea using CKEditor library for the article's content</li> <li>• Submit button</li> </ul>
2	Member fills out the form.	
3	Member submit article.	<p>Add new article the database.</p> <p>System shows message saying “Article submitted! Please wait for admin’s approval!”.</p>

### **Alternative Scenario:**

N/A

### **Exceptions:**

No	Cause	System Response
1	Member submit article with empty title	System shows error message to ask the member to enters missing required fields.
2	Member submit article with empty description	System shows error message to ask the member to enters missing required fields.
3	Member submit article with no thumbnail uploaded	System shows error message to ask the member to enters missing required fields.
4	Member submit article with empty content	System shows error message to ask the member to enters missing required fields.
5	Member upload file too large.	System shows error message to ask the member to enters missing required fields.

**Relationships:** N/A

### **Business Rules:**

- In case of success scenarios, a new article would be posted to the database. The article's status will be “Pending”.
- Article id is automatically initialized by the system. File upload must be file type of images: \*.jpeg, \*.jpg, \*.png, \*.gif.

### 2.3.2.7. < Member > Edit article Usecase

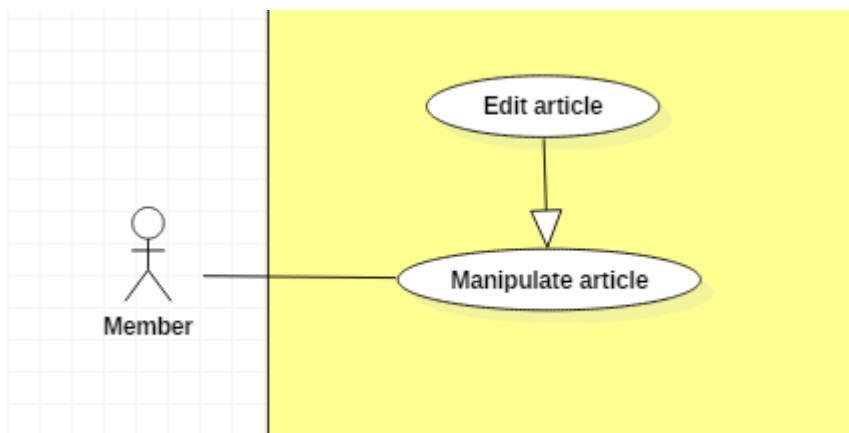


Figure 12 - < Member > Edit article Usecase

USE CASE – UC _Member_07			
Usecase No.	Member_07	Usecase Version	1.0
Usecase Name	Edit article		
Author	KhanhNB		
Date	25/09/2018	Priority	Normal
<b>Actor:</b>			
- Member			
<b>Summary:</b>			
- This use case allows member to edit an article.			
<b>Goal:</b>			
- The articles is updated in database			
<b>Triggers:</b>			
- Member edit an article and submit it.			
<b>Preconditions:</b>			
- User must login into system with role Member.			
- Member viewing an article that they posted			
- Member selects “Edit article”			
<b>Post Conditions:</b>			
- <b>Success:</b> New article is updated in the database system.			
- <b>Fail:</b> Show error message.			

**Main Success Scenario:**

Step	Actor Action	System Response
1	Member go to “Edit article” page	<ul style="list-style-type: none"> <li>• Input field for article’s Title</li> <li>• Dropdown list to select article’s category</li> <li>• A textarea for the article’s description</li> <li>• An image upload button to upload the article’s thumbnail</li> <li>• An empty image to preview the thumbnail</li> <li>• A textarea using CKEditor library for the article’s content</li> <li>• Submit button</li> </ul>
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	Member submit article with empty title	System shows error message to ask the member to enters missing required fields.
2	Member submit article with empty description	System shows error message to ask the member to enters missing required fields.
3	Member submit article with no thumbnail uploaded	System shows error message to ask the member to enters missing required fields.
4	Member submit article with empty content	System shows error message to ask the member to enters missing required fields.

5	Member upload file too large.	System shows error message to ask the member to enter missing required fields.
---	-------------------------------	--------------------------------------------------------------------------------

**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

#### 2.3.2.8. < Member > Search event Usecase

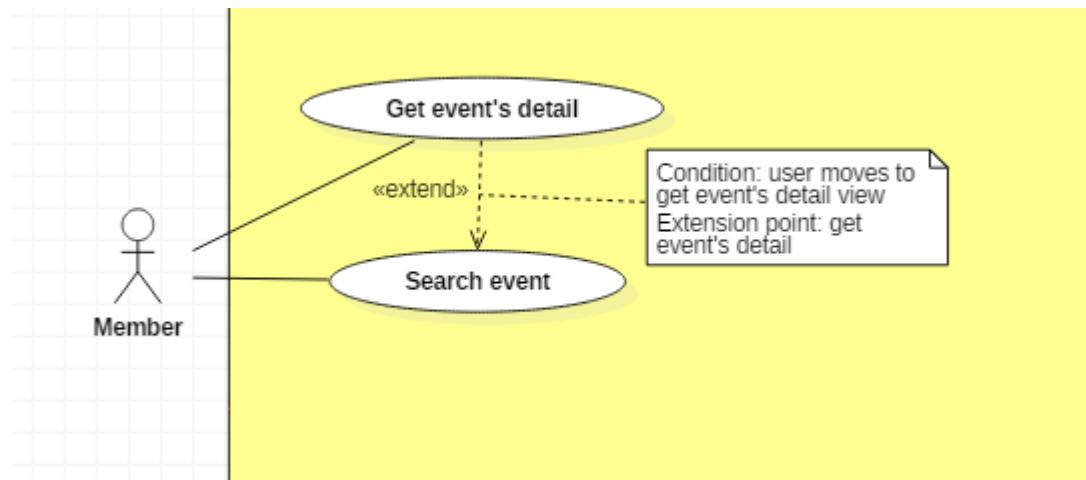


Figure 13 - < Member > Search event Usecase

USE CASE – UC _Member_08			
<b>Usecase No.</b>	Member_08	<b>Usecase Version</b>	1.0
<b>Usecase Name</b>	Search event		
<b>Author</b>	Khanhnb		
<b>Date</b>	03/10/2018	<b>Priority</b>	Normal
<b>Actor:</b>	<ul style="list-style-type: none"> <li>- Member..</li> </ul>		
<b>Summary:</b>			

- This use case allows member to search events.

**Goal:**

- Make it more convenient for member to seek for events.

**Triggers:**

- Member sends search event command to the system.

**Preconditions:**

- The search event 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 event view  [Alternative scenario 1]	System displays search article view with the following elements: <ul style="list-style-type: none"> <li>• Search bar: free text input</li> <li>• Dropdown list: select sort by newest or oldest events, default</li> <li>• SEARCH BY: event's title</li> </ul>
2	Member fills out the input text, select sort type	
3	Member clicks on “SEARCH” button or press Enter	A list of events with matching titles is shown

**Alternative Scenario:**

Step	Actor Action	System Response

1	Member select “Show nearby events” button	A list of events located within 10km of the user’s current location is shown
---	-------------------------------------------	------------------------------------------------------------------------------

**Exceptions:** N/A

**Relationships:** N/A

**Business Rules:**

- System search events by title matching with the keyword input from guest
- A list of records will go with pagination and sorting filters
- System search nearby events using Haversine algorithm to calculate the distance between the user and each event

#### 2.3.2.9. < Member > Get event’s detail Usecase

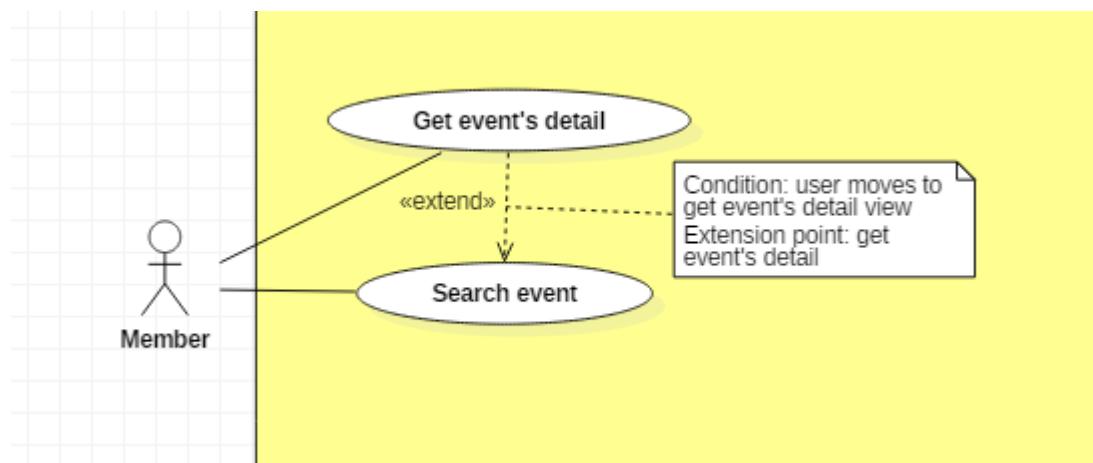


Figure 14 - < Member > Get event's detail Usecase

USE CASE – UC _Member_09			
<b>Usecase No.</b>	Member_09	<b>Usecase Version</b>	1.0
<b>Usecase Name</b>	Get event's info		
<b>Author</b>	Khanhnb		
<b>Date</b>	03/10/2018	<b>Priority</b>	Normal

**Actor:**

- Member.

**Summary:**

- This use case allows member to get an event's info.

**Goal:**

- Display an event's content

**Triggers:**

- Member sends get news's detail command to the system.

**Preconditions:**

- Member select a event from the list of event.

**Post Conditions:**

- **Success:** The matching record is shown.
- **Fail:** Show error message.

**Main Success Scenario:**

Step	Actor Action	System Response
1	Member clicks on an event and goes to event detail view	<p>System displays event's detail view with the following elements:</p> <ul style="list-style-type: none"><li>• The title of the event: text</li><li>• The date when the event starts and ends</li><li>• The date then registration starts and ends</li><li>• The location of the event</li><li>• The number of people registered for this event</li><li>• The event's content</li><li>• (if available) The event's rating and feedback by its attendees</li><li>• (if the event's status is "Finished") Text line saying "This event has finished"</li><li>• (if the event's status is "Cancelled") Text line saying "This event was cancelled"</li><li>• (if today is within the registration period) Register button</li></ul>

		<p>If the current user has registered to this event, system display the following elements:</p> <ul style="list-style-type: none"> <li>• The username of the user</li> <li>• The date when the user registered on</li> <li>• The amount of tickets the user bought</li> <li>• A star rating area to rate the event</li> <li>• A textarea for the feedback</li> <li>• A submit feedback button</li> </ul> <p>If the current user has sent feedback to htis event then system display the following elements:</p> <ul style="list-style-type: none"> <li>• The star rating with the user's rating</li> <li>• A textarea showing the user's written feedback</li> </ul>
<b>Alternative Scenario:</b> N/A		
<b>Exceptions:</b> N/A		
<b>Relationships:</b> N/A		
<b>Business Rules:</b> N/A		

#### 2.3.2.10. < Member > Register event Usecase

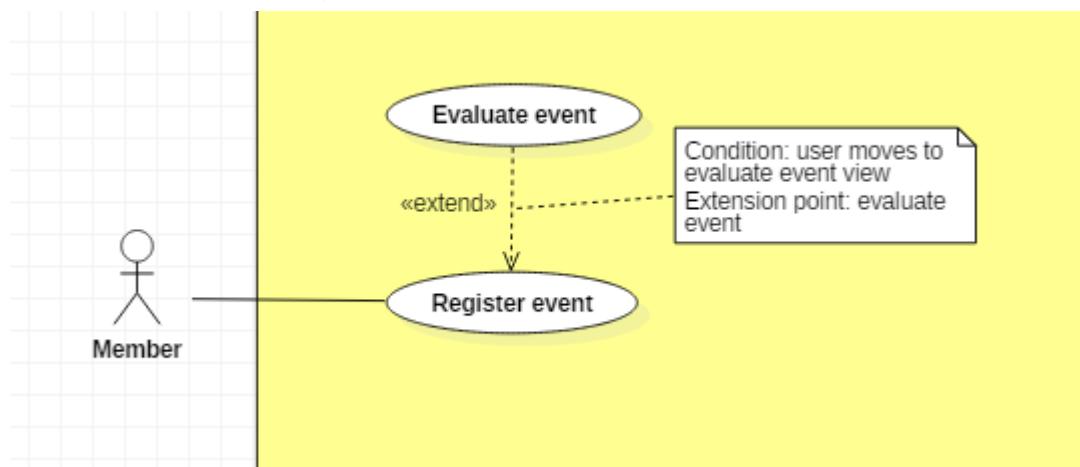


Figure 15 - < Member > Register event Usecase

<b>USE CASE – UC _Member_10</b>			
<b>Usecase No.</b>	Member_10	<b>Usecase Version</b>	1.0
<b>Usecase Name</b>	Register event		
<b>Author</b>	Khanhn		
<b>Date</b>	03/10/2018	<b>Priority</b>	Normal
<b>Actor:</b> <ul style="list-style-type: none"> <li>- Member .</li> </ul> <b>Summary:</b> <ul style="list-style-type: none"> <li>- This use case allows Member to register to an event</li> </ul> <b>Goal:</b> <ul style="list-style-type: none"> <li>- The member is added to the event's attendee list</li> </ul> <b>Triggers:</b> <ul style="list-style-type: none"> <li>- Member send register event command to the system</li> </ul> <b>Preconditions:</b> <ul style="list-style-type: none"> <li>- Member is viewing an event's detail, the current date is within registration time</li> </ul> <b>Post Conditions:</b> <ul style="list-style-type: none"> <li>- <b>Success:</b> Member successfully register to an event, member's id is added to the list of attendees</li> <li>- <b>Fail:</b> Show error message.</li> </ul> <b>Main Success Scenario:</b>			
<b>Step</b>	<b>Actor Action</b>	<b>System Response</b>	
1	Member select “Register” when in event detail view	A new dialog is shown with following labels and fields: <ul style="list-style-type: none"> <li>• The member’s username</li> <li>• Today’s date</li> <li>• Number of remaining tickets</li> <li>• Number of tickets text input</li> <li>• Register button</li> </ul>	
2	Member fill all required form		

3	Member click “Register”	System add member to the list of attendees  A message saying “Registered successful!” shows.
---	-------------------------	----------------------------------------------------------------------------------------------------

### Alternative Scenario:

#### Exceptions:

No	Cause	System Response
1	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 number of tickets the user enter must be lower or equal with the remaining number of tickets the event has

#### 2.3.2.11. < Member > Create proposal Usecase

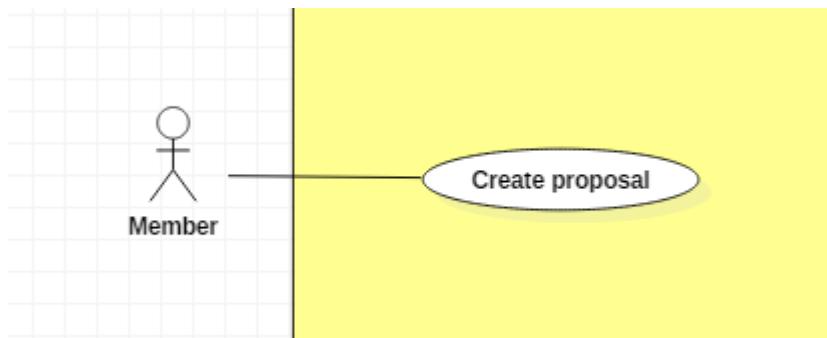


Figure 16 - < Member > Create proposal Usecase

USE CASE – UC _Member_11			
Usecase No.	Member_11	Usecase Version	1.0
Usecase Name	Evaluate event		

<b>Author</b>	Khanhnb		
<b>Date</b>	03/10/2018	<b>Priority</b>	Normal

**Actor:**

- Member .

**Summary:**

- This use case allows Member to create an event proposal

**Goal:**

- The member can send the admin a proposal idea for an event

**Triggers:**

- Member submit the proposal information

**Preconditions:**

- User is logged in with the role “Member”

**Post Conditions:**

- **Success:** Member successfully send the proposal, proposal is saved to the database
- **Fail:** Show error message.

**Main Success Scenario:**

Step	Actor Action	System Response
1	Member go to “Create a new proposal” page	<p>“Create new proposal” view is shown with following labels and fields:</p> <ul style="list-style-type: none"> <li>• Input field for event’s title</li> <li>• Textarea for a short description of the event</li> <li>• Input field for the location of the event</li> <li>• A textarea using CKEditor library for the event’s content</li> <li>• Submit button</li> </ul>
2	Member fills out the form.	

3	Member submit article.	Add new article the database.  System shows message saying “Proposal sent! Thank you for your time”.
<b>Alternative Scenario:</b> N/A		
<b>Exceptions:</b> N/A		
<b>Relationships:</b> N/A		
<b>Business Rules:</b> N/A		

### 2.3.2.12. < Member > Search exchange Usecase

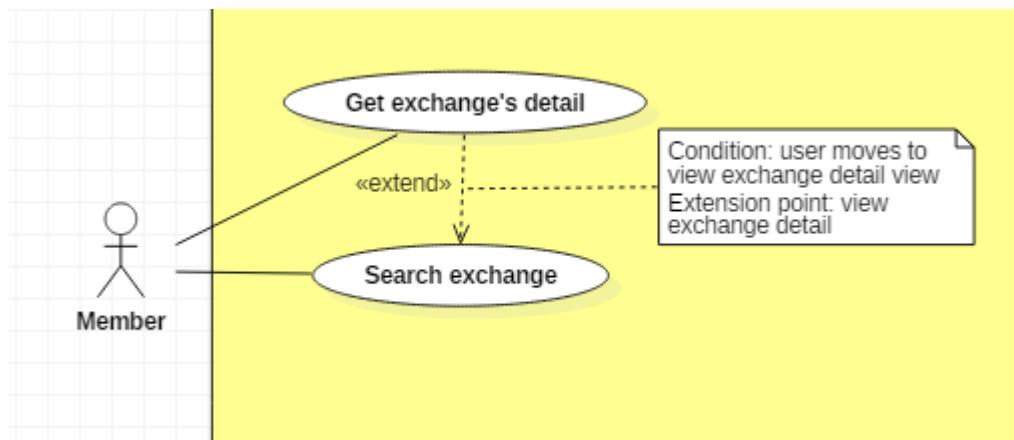


Figure 17 - < Member > Search exchange Usecase

USE CASE – UC _Member_12			
<b>Usecase No.</b>	Member_12	<b>Usecase Version</b>	1.0
<b>Usecase Name</b>	Search exchange		
<b>Author</b>	SangLNN		

<b>Date</b>	26/09/2018	<b>Priority</b>	Normal
-------------	------------	-----------------	--------

**Actor:**

- Member.

**Summary:**

- This use case allows member search exchange by keyword (title,brand,model) or location and range.

**Goal:**

- System create view to show search result.

**Triggers:**

- User sends search a trade post command with keyword, location and range to the system.

**Preconditions:**

- User must on website page.

**Post Conditions:**

- **Success:** System create view to show all result matching keyword and range with location.
- **Fail:** Show error message.

**Main Success Scenario:**

<b>Step</b>	<b>Actor Action</b>	<b>System Response</b>
1	Member click on “Trade market” link from header navbar or “Trade listing” on trade market page to send list all trade command.	<p>System create a view show search form with following fields:</p> <ul style="list-style-type: none"> <li>• Search input: text input , required.</li> <li>• “More” button</li> </ul> <p>[Alternative Scenario 1]</p>

2	<p>Member type keyword and press enter to send search command.</p>	<p>[Exception 1]</p> <ul style="list-style-type: none"> <li>- System create a view show all result.</li> <li>- Filter</li> </ul> <p>[Alternative Scenario 2]</p> <ul style="list-style-type: none"> <li>- Group by</li> </ul> <p>[Alternative Scenario 3]</p>
---	--------------------------------------------------------------------	---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

#### **Alternative Scenario:**

Alternative Scenario 1: Search with location and range

<b>Step</b>	<b>Actor Action</b>	<b>System Response</b>
1	<p>Member click on “more” button.</p>	<p>System will show additional search form with following labels and fields:</p> <ul style="list-style-type: none"> <li>• Search with location and form: radio button.</li> <li>• Location: text input, readonly.</li> <li>• Change your location: button.</li> <li>• Range: range input, required, min value: 500, max value: 20000.</li> </ul>
2	<p>Member fills out the form.</p>	
3	<p>Member type keyword and press enter to send search command.</p>	<p>[Exception 1]</p> <ul style="list-style-type: none"> <li>- System create a view show all result.</li> <li>- Filter</li> </ul>

		[Alternative Scenario 2] - Group by [Alternative Scenario 3]
--	--	--------------------------------------------------------------------

Alternative Scenario 2 : Filter Scenario.

Step	Actor Action	System Response
1	Member 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	Member 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 pagination with 8 trade post per page.
- Have 3 grouping tab: All, Selling , Buying.
- Have 4 filter: Date descending, Date ascending, Price descending, price ascending.

#### 2.3.2.13. < Member > Get exchange detail Usecase

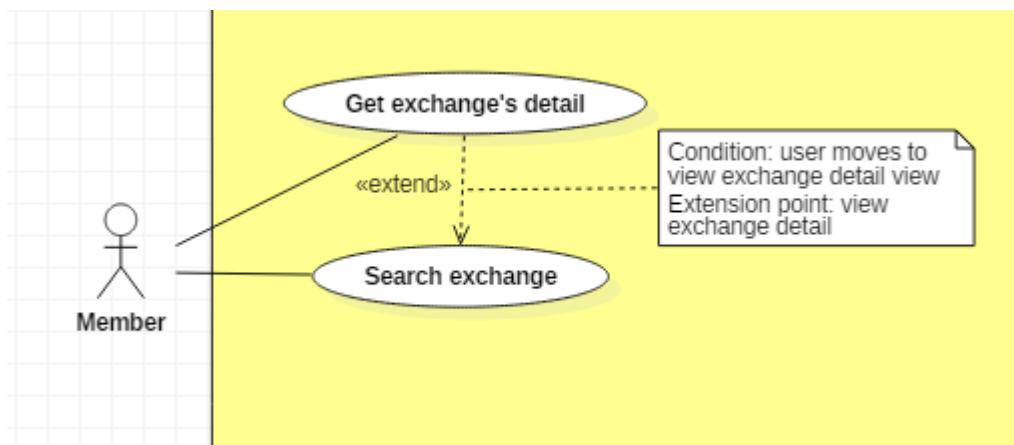


Figure 18 - < Member > Get exchange detail Usecase

#### USE CASE – UC \_Member\_13

<b>Usecase No.</b>	Member_13	<b>Usecase Version</b>	1.0
<b>Usecase Name</b>	Get exchange's detail		
<b>Author</b>	SangLNN		
<b>Date</b>	26/09/2018	<b>Priority</b>	Normal

**Actor:**

- Member.

**Summary:**

- This use case allows member view exchange details.

**Goal:**

- System create view to show all information about exchange.

**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	Member 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"><li>• Photo slideshow</li><li>• Price</li><li>• Title</li><li>• Trade id</li><li>• Date posted</li><li>• Location</li><li>• Type of trade</li><li>• Brand</li><li>• Model</li><li>• Quantity</li><li>• Rating</li><li>• “Send order” button.</li></ul>

		<p>Trader information section:</p> <ul style="list-style-type: none"> <li>• Delivery</li> <li>• Report this trade</li> </ul> <p>Description section:</p> <ul style="list-style-type: none"> <li>• Description</li> </ul>
<b>Alternative Scenario:</b> N/A		
<b>Exceptions:</b> N/A		
<b>Relationships:</b> N/A		
<b>Business Rules:</b> N/A		

#### 2.3.2.14. < Member > Request order Usecase

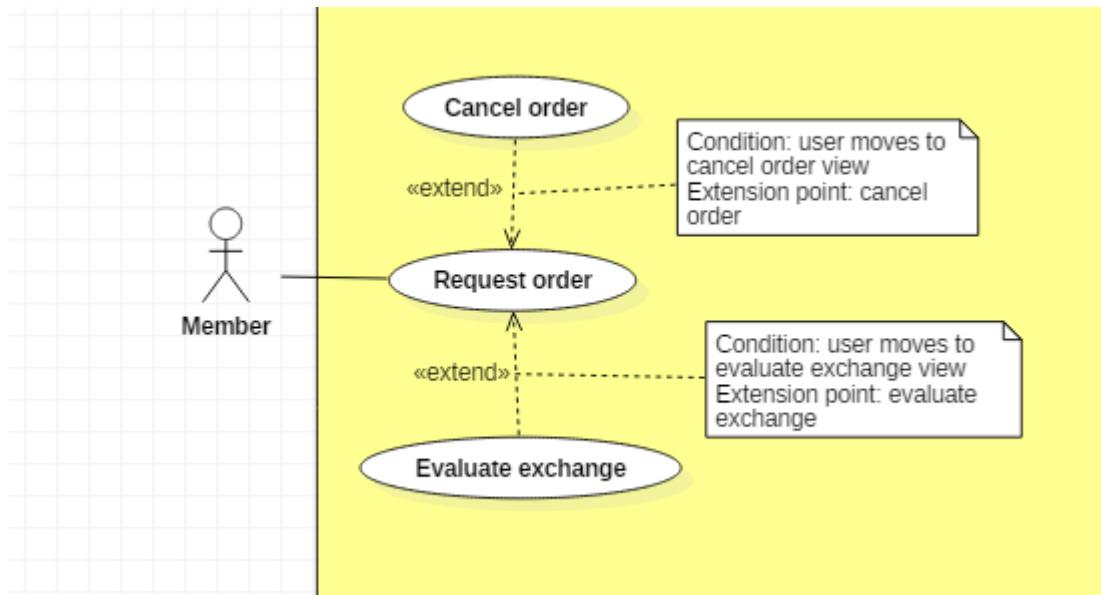


Figure 19 - < Member > Request order Usecase

USE CASE – UC _Member_14			
<b>Usecase No.</b>	Member_14	<b>Usecase Version</b>	1.0
<b>Usecase Name</b>	Request order		
<b>Author</b>	SangLNN		

<b>Date</b>	26/09/2018	<b>Priority</b>	Normal
-------------	------------	-----------------	--------

**Actor:**

- Member.

**Summary:**

- This use case allows member to order a trade.

**Goal:**

- User can get request order 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.

**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"> <li>• Quantity: Number, required, positive integer and below than stock quantity.</li> </ul> <p>Trader information:</p> <ul style="list-style-type: none"> <li>• Full name: Text input, readonly.</li> <li>• Email: Text input, readonly.</li> <li>• Phone: Text input, readonly.</li> </ul>

		<ul style="list-style-type: none"> <li>• Address: Text input, readonly.</li> <li>• Select address: button.</li> </ul> <p>[Alternative Scenario 1]</p> <ul style="list-style-type: none"> <li>• Auto get location: button.</li> </ul> <p>[Alternative Scenario 2]</p>
2	User click “Send order” button.	<p>Validate data.</p> <p>[Exception 1]</p> <p>System save order data to database and send notification to buyer/seller.</p>

#### **Alternative Scenario:**

Alternative Scenario 1: Select address 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"> <li>• Province/City: select option, required.</li> <li>• District: select option, required.</li> <li>• Ward: select option, required.</li> <li>• Street name: text input, required, length 3 - 50.</li> <li>• Apartment number: text input,</li> </ul>

		<p>required, length 3 – 50.</p> <ul style="list-style-type: none"> <li>• Your full address: text input, readonly.</li> </ul> <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 into “Address” field from trader information form.</p>

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.	<p>[Exception 2]</p> <p>System auto get your location and generated to text address then autoform to address field.</p>

#### Exceptions:

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

		System shows error message to ask member enters validate 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.

### 2.3.3. < Authenticated User > Overview Usecase

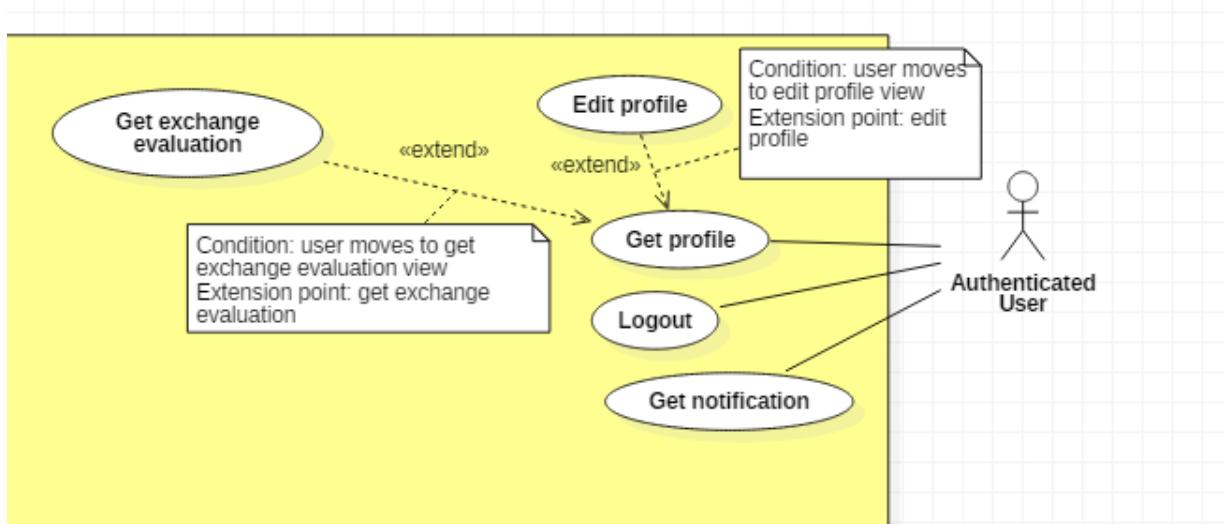


Figure 20 - < Authenticated User > Overview Usecase

### 2.3.3.1. < Authenticated User > Get profile Usecase

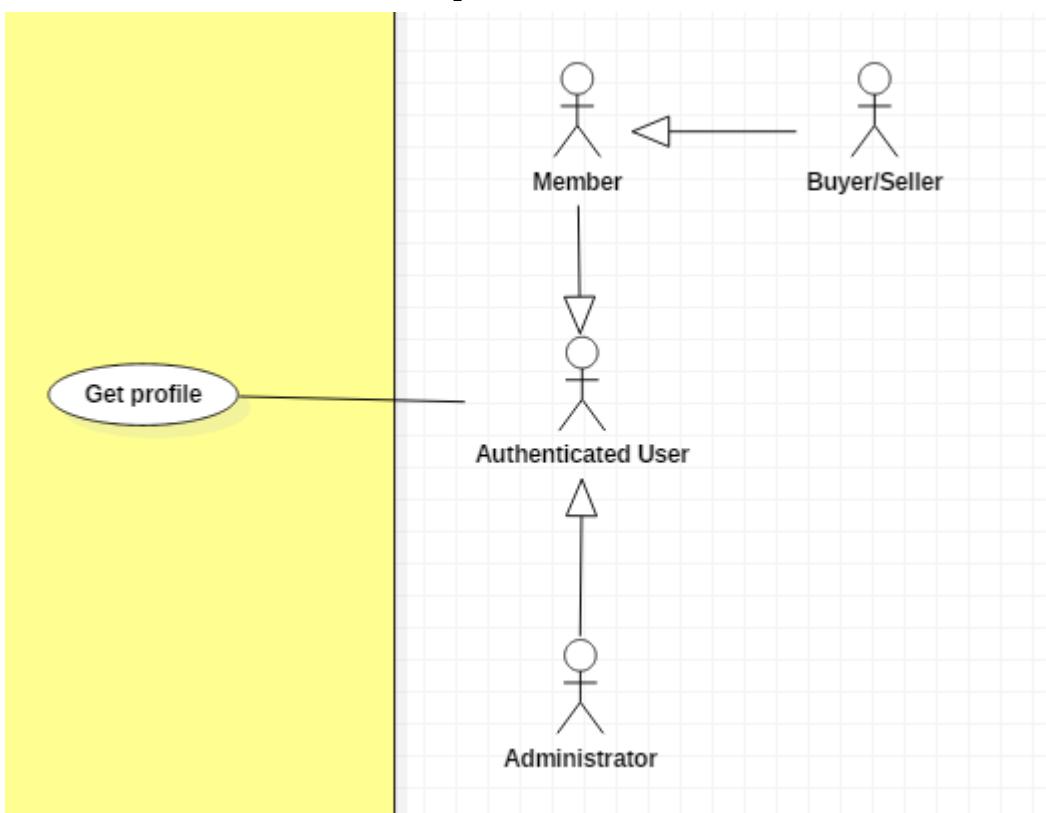


Figure 21 - < Authenticated User > Get profile Usecase

USE CASE – UC _AuthenticatedUser_01			
<b>Usecase No.</b>	AuthenticatedUser_01	<b>Usecase Version</b>	1.0
<b>Usecase Name</b>	Get profile		
<b>Author</b>	TungHNM		
<b>Date</b>	03/10/2018	<b>Priority</b>	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 other user's name  [Alternative 1]	<p>System displays the selected user's profile page with the following labels and text:</p> <ul style="list-style-type: none"><li>• Avatar: img</li><li>• Email: text</li><li>• Fullname: text</li><li>• Role: text</li><li>• Tel: text</li><li>• Address: text</li><li>• Birthday: text</li><li>• Status: text</li></ul> <p>System displays statistic on the left side and TOP 10 ranking user on the right side, icon show exchange evaluation button to show other user's exchange evaluation</p>

**Alternative Scenario:**

**Alternative Scenario 1:** Authenticated User gets his/her own profile

Step	Actor Action	System Response
1	Authenticated user clicks on his/her own profile	<p>System displays his/her own profile view with the following labels and text:</p> <ul style="list-style-type: none"> <li>● Avatar: img</li> <li>● Email: input text</li> <li>● Firstname: input text</li> <li>● Middlename: input text</li> <li>● Lastname: input text</li> <li>● Role: input text, disabled</li> <li>● Tel: input text</li> <li>● Address: input text</li> <li>● Birthday: DateTimePicker</li> <li>● Status: input text, disabled</li> </ul> <p>System also displays EDIT button and icon get exchange evaluation button to show other user's exchange evaluation if any</p>

**Exceptions:** N/A

**Relationships:** N/A

**Business Rules:** N/A

### 2.3.3.2. < Authenticated User > Edit profile Usecase

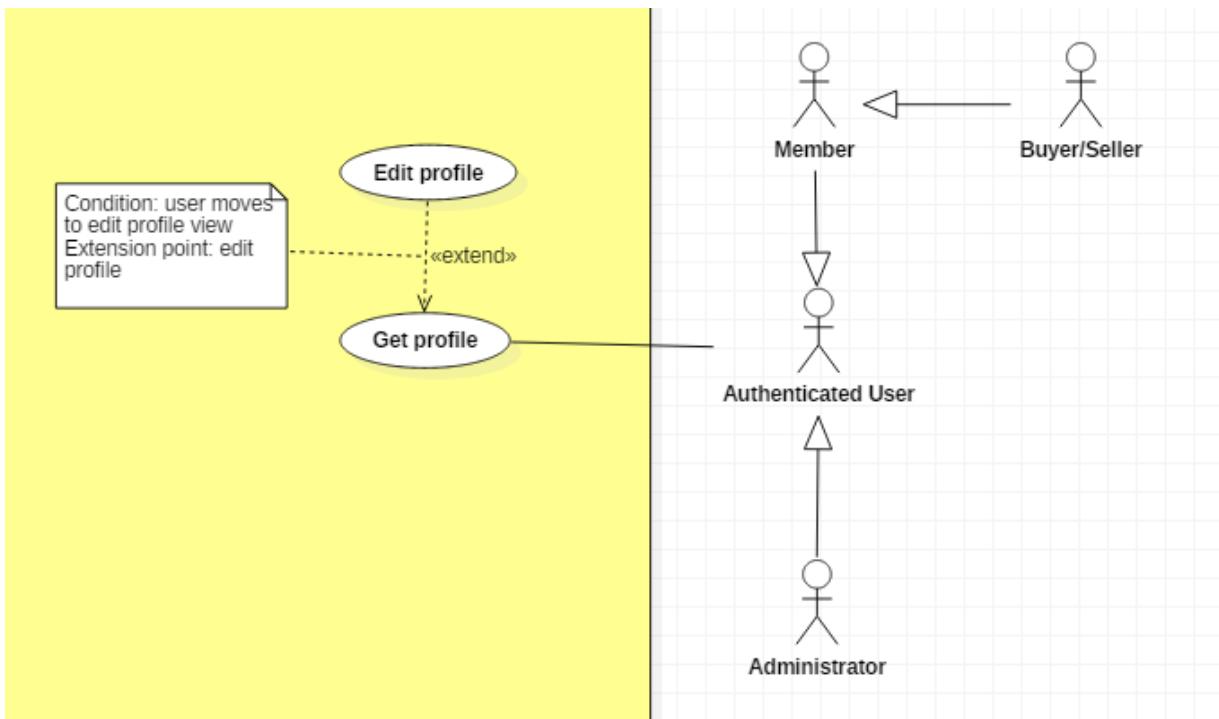


Figure 22 - < Authenticated User > Edit profile Usecase

USE CASE – UC _AuthenticatedUser_02			
<b>Usecase No.</b>	AuthenticatedUser_02	<b>Usecase Version</b>	1.0
<b>Usecase Name</b>	Edit profile		
<b>Author</b>	TungHNM		
<b>Date</b>	03/10/2018	<b>Priority</b>	Normal
<b>Actor:</b>	<ul style="list-style-type: none"> <li>- Authenticated User.</li> </ul>		
<b>Summary:</b>	<ul style="list-style-type: none"> <li>- This use case allows authenticated user to update his/her profile information.</li> </ul>		
<b>Goal:</b>	<ul style="list-style-type: none"> <li>- Authenticated User can edit his/her profile.</li> </ul>		

**Triggers:**

- Authenticated User sends edit profile command to the system.

**Preconditions:**

- User must login into system.
- User's own profile view 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” button	<p>Edit Profile page is shown with following labels and fields:</p> <ul style="list-style-type: none"><li>• Email: text, required, max length: 50, RegEx: /^[a-zA-Z_\.]{1,}@[a-zA-Z_\.]+\.[a-zA-Z]{2,6}\$/</li><li>• First Name: text, max length: 50</li><li>• Middle Name: text, max length: 50</li><li>• Last Name: text, max length: 50</li><li>• Tel: text, length: 10-11</li><li>• Address: text, max length: 250</li><li>• Birthday: DateTimePicker</li><li>• Password: text, disabled</li><li>• Status: text, disabled</li><li>• Change image: button, icon</li></ul> <p>System automatically fills out the form with information of the selected user's profile in database .</p>

2	User fills out the form [Alternative 1]	
4	User clicks on “SAVE” button [Alternative 2]	<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:

**Alternative Scenario 1:** Authenticated User clicks on change image icon button

Step	Actor Action	System Response
1	Authenticated User clicks on change image icon button	<p>System displays choose files dialog for user to choose</p>
2	Authenticated User choose files	<p>Validate file type.</p> <p>[Exception 6]</p> <p>System displays the selected image on the view</p>

### Exceptions:

No	Cause	System Response
1	Email does not match the required format: /^[a-zA-Z0-9_\.-]+@[a-zA-Z]+\.[a-zA-Z]{2,3}\$^	

	<code>] + ) @ ( [ \da - z \ ] + ) \ . ( [ a - 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 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
6	Invalid image type	System shows message that notify user the selected file is not type of image

**Relationships:** Extends to Get Profile use case

**Business Rules:**

- User's profile will be automatically filled out when Edit Account page is loaded or whenever an exception occurs.
- The selected avatar from choose file dialog must be type of \*.jpeg/\*.\*.png/\*.\*.jpg/\*.\*.gif
- In case of success scenarios, account's information would be updated to the database and displays the profile page with updated information.
- System will saved user's avatar with file name: username.filetype
- Email must not be duplicate in database.

### 2.3.4. < Administrator > Overview Usecase



Figure 23 - < Administrator > Overview Usecase

### 2.3.4.1. < Administrator > Crawl model Usecase

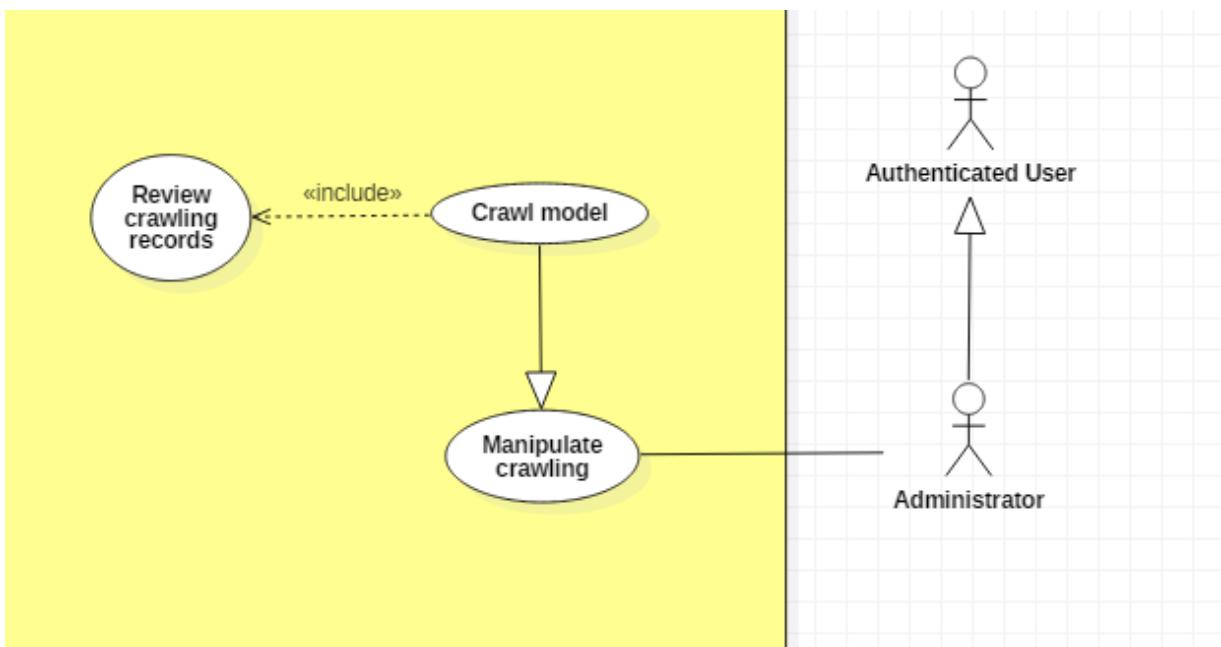


Figure 24 - < Administrator > Crawl model Usecase

USE CASE – UC _Admin_01			
<b>Usecase No.</b>	Admin_01	<b>Usecase Version</b>	1.0
<b>Usecase Name</b>	Crawl model		
<b>Author</b>	TungHNM		
<b>Date</b>	28/09/2018	<b>Priority</b>	Normal
<b>Actor:</b>			
- Administrator.			
<b>Summary:</b>			
<ul style="list-style-type: none"> <li>- This use case allows administrator to crawl model from high domain: <a href="https://www.1999.co.jp/eng/gundam">https://www.1999.co.jp/eng/gundam</a>.</li> </ul>			
<b>Goal:</b>			
<ul style="list-style-type: none"> <li>- Information about models will be crawled and saved to database with status “crawlpending”.</li> </ul>			
<b>Triggers:</b>			

- Administrator sends crawl models command to the system.

**Preconditions:**

- User must login into system with role Administrator.
- Crawl model page is displayed.

**Post Conditions:**

- **Success:** The crawl is running with counting records and new records. Models's crawling records will be saved to database with status “crawlpending” and will display on pending model page for administrator to review.
- **Fail:** Show error message.

**Main Success Scenario:**

Step	Actor Action	System Response
1	Administrator goes to crawl model view	<p>System displays the following items:</p> <ul style="list-style-type: none"> <li>• CRAWL button</li> <li>• List of crawl history records in log “crawlmodel.dat”</li> </ul>
2	Administrator clicks on “CRAWL” button	<p>System displays messages: “System is crawling successfully”.</p> <p>System displays icon “Track crawling process” button</p> <p>Save new models to database, log file.</p>
3	Administrator switches on “Track crawling process” button  [Alternative 1]	<p>System shows counting number of crawl records &amp; number of new records.</p>

**Alternative Scenario:**

**Alternative Scenario 1: Switch off “Track crawling process” button**

Step	Actor Action	System Response
------	--------------	-----------------

1	<p>Administrator switches off “Track crawling process” button</p>	<p>System hide number of crawl records and number of new records.</p>
---	-------------------------------------------------------------------	-----------------------------------------------------------------------

**Relationships:** N/A

**Business Rules:**

- In case of success scenarios, new model crawling data would be saved to database with status “crawlpending”.
- All crawling records are matching with data in database using MD5 hash function (name + product series + series title + manufacturer)
- System log to file “crawlmodel.dat” when start crawl with status “Crawling” and update to “Done” when finish

#### 2.3.4.2. < Administrator > Crawl article Usecase

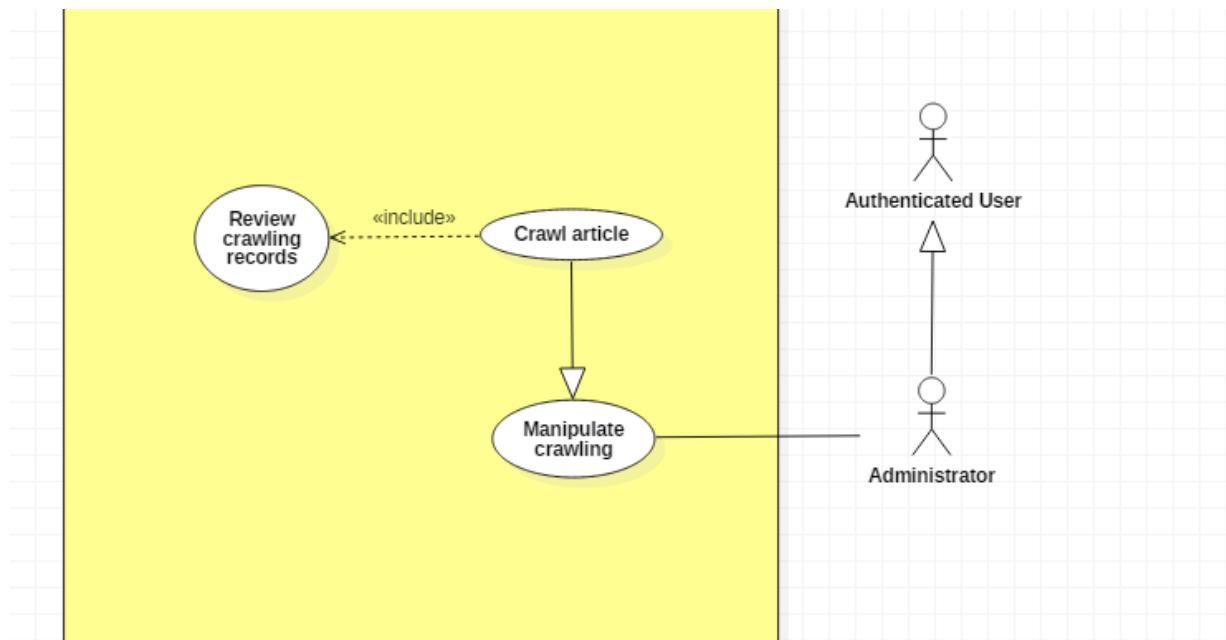


Figure 25 - < Administrator > Crawl article Usecase

USE CASE – UC _Admin_02			
Usecase No.	Admin_02	Usecase Version	1.0

<b>Usecase Name</b>	Crawl article		
<b>Author</b>	TungHNM		
<b>Date</b>	28/09/2018	<b>Priority</b>	Normal

**Actor:**

- Administrator.

**Summary:**

- This use case allows administrator to crawl article from high domain:  
<https://www.gunpla101.com>

**Goal:**

- Information about article will be crawled and saved to database with status “crawlpending”.

**Triggers:**

- Administrator sends crawl article command to the system.

**Preconditions:**

- User must login into system with role Administrator.
- Crawl article page is displayed.

**Post Conditions:**

- **Success:** The crawl is running with counting records and new records. Article's crawling records will be saved to database with status “crawlpending” and will display on pending article page for administrator to review.
- **Fail:** Show error message.

**Main Success Scenario:**

Step	Actor Action	System Response
1	Administrator goes to crawl model view	<p>System displays the following items:</p> <ul style="list-style-type: none"> <li>• CRAWL button</li> <li>• List of crawl history records in log “crawlarticle.dat”</li> </ul>
2	Administrator clicks on “CRAWL” button	

		<p>System display messages: “System is crawling successfully”.</p> <p>System matches crawl article records with existed data in database.</p> <p>Save new articles to database, log file.</p>
3	Administrator switchs on “Track crawling process” button  [Alternative 1]	System shows counting number of crawl records & number of new records.

### **Alternative Scenario:**

#### **Alternative Scenario 1: Switch off “Track crawling process” button**

<b>Step</b>	<b>Actor Action</b>	<b>System Response</b>
1	Administrator switchs off “Track crawling process” button	System hide number of crawl records and number of new records.

**Relationships:** N/A

#### **Business Rules:**

- In case of success scenarios, new article crawling data would be saved to database with status “crawlpending”.
- System log to file “crawlarticle.dat” when start crawl with status “Crawling” and update to “Done” when finish

### 2.3.1.3. < Administrator > Add new model Usecase

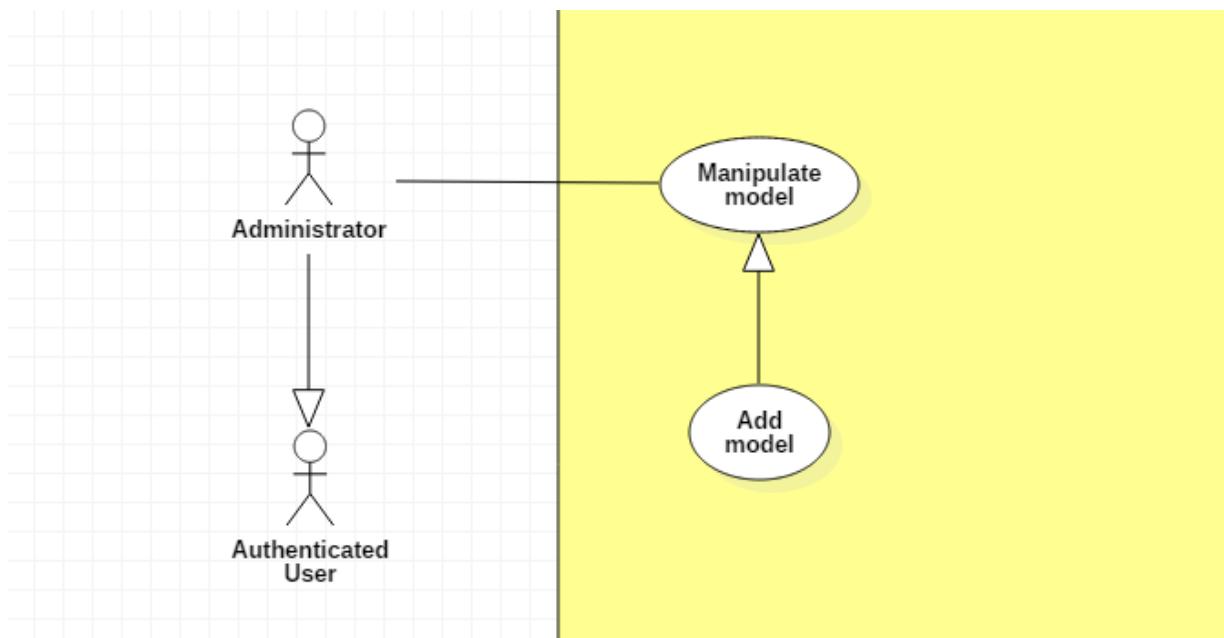


Figure 26 - < Administrator > Add new model Usecase

USE CASE – UC _Admin_03			
<b>Usecase No.</b>	Admin_03	<b>Usecase Version</b>	1.0
<b>Usecase Name</b>	Add new model		
<b>Author</b>	TungHNM		
<b>Date</b>	30/09/2018	<b>Priority</b>	Normal
<b>Actor:</b>	<ul style="list-style-type: none"> <li>- Administrator.</li> </ul>		
<b>Summary:</b>	<ul style="list-style-type: none"> <li>- This use case allows administrator to create a new model.</li> </ul>		
<b>Goal:</b>	<ul style="list-style-type: none"> <li>- To add new model to system.</li> </ul>		
<b>Triggers:</b>	<ul style="list-style-type: none"> <li>- Administrator sends add new model command to the system.</li> </ul>		

**Preconditions:**

- User must login into system with role Administrator.
- The create new model page is displayed.

**Post Conditions:**

- **Success:** A model with information is created and saved to database successfully.
- **Fail:** Show error message.

**Main Success Scenario:**

Step	Actor Action	System Response
1	Administrator goes to create new model view	An add new model form is shown with following labels and fields: <ul style="list-style-type: none"><li>• Code: text, required with regEx: /^[A-z]+\$/</li><li>• Name: text, required, max length: 50</li><li>• Product Series: dropdown list, editable</li><li>• Series Title: dropdown list, editable</li><li>• Manufacturer: dropdown list, editable</li><li>• Price: free text input</li><li>• Released Date: DateTimePicker</li><li>• Description: free text input, max length: 350</li><li>• Import image: input type file</li><li>• Image type: dropdown list</li></ul>
2	Administrator fills out the form  [Alternative 1]	
3	Administrator clicks on “CREATE” button	

		<p>Validate data [Exception 1, 2, 3, 5]</p> <p>New model is created and saved to database.</p> <p>Model's images are saved to database and to folder "uploads"</p> <p>System displays message Model {code} is saved successfully!</p>
--	--	-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

### **Alternative Scenario:**

#### **Alternative Scenario 1: Click on “Add images” button**

<b>Step</b>	<b>Actor Action</b>	<b>System Response</b>
1	Administrator clicks on “Import image” button	<p>System shows dialog for administrator to choose local image</p>
2	Administrator chooses an image or multi images	<p>Validate file type [Exception 4]</p> <p>System shows the selected image on the view with following data: thumb image, image name with image type, image size, image type dropdown list, cancel button and checkbox (to delete multi images)</p>
3	Administrator delete an image or multi images	<p>System deletes images on the view.</p>

**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 and must not be duplicated in database.
- Each of local image which is imported by administrator must go with an image type
- File which is imported by administrator must be type of images: \*.jpeg, \*.jpg, \*.png, \*.gif.
- Input product series, series title and manufacturer are match with records in database. If new, then save to database
- Model's images is saved in folder "uploads" with file name of model's name go with auto generated id

#### 2.3.4.4. < Administrator > Edit model Usecase

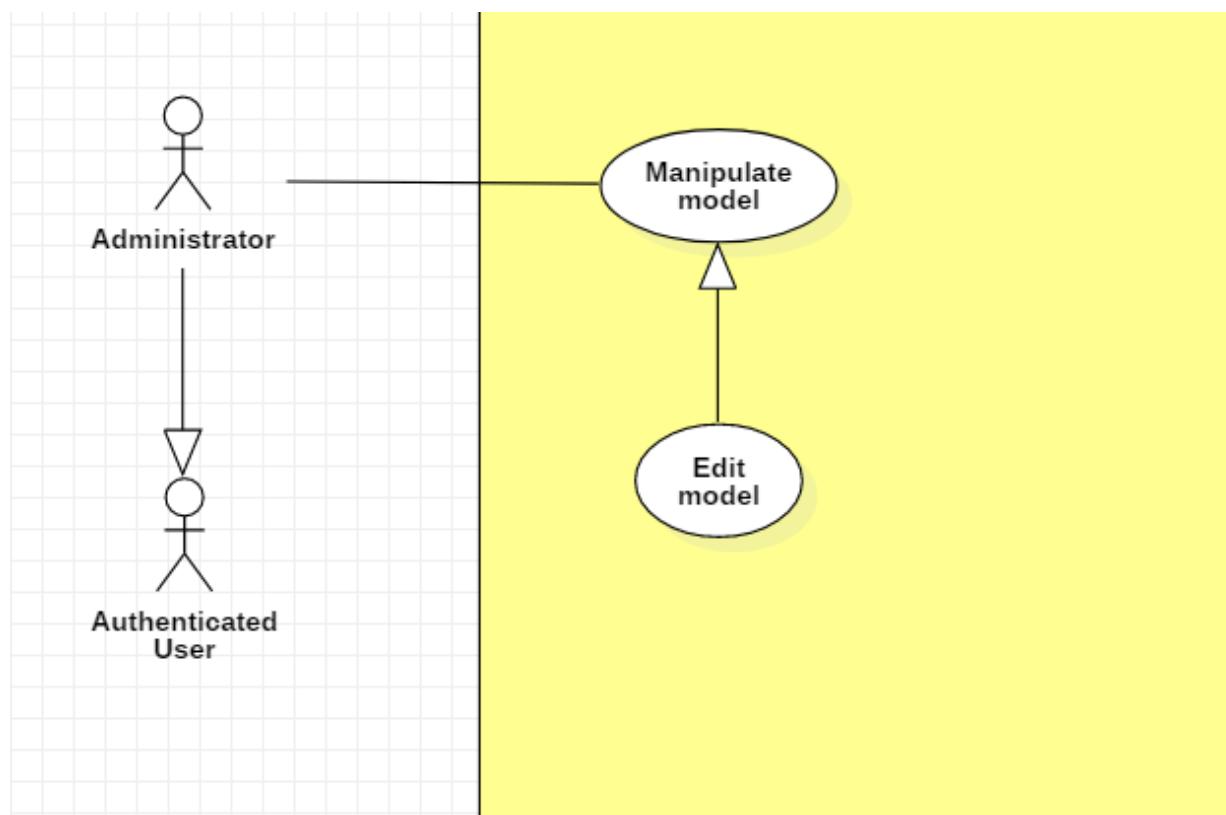


Figure 27 - < Administrator > Edit model Usecase

USE CASE – UC _Admin_04			
Usecase No.	Admin_04	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"><li>• Code: text, required with regEx: /^[A-z]+\$/</li><li>• Name: text, required, max length: 50</li><li>• Product Series: dropdown list, editable</li><li>• Series Title: dropdown list, editable</li><li>• Manufacturer: dropdown list, editable</li><li>• Price: free text input</li><li>• Released Date: DateTimePicker</li><li>• Description: free text input,</li></ul>

		<p>max length: 350</p> <ul style="list-style-type: none"> <li>• Import image: input type file</li> <li>• Image type: dropdown list</li> </ul> <p>System automatically loads the selected gundam model's information to the form.</p> <p>System loads selected model's images with following labels for each image:</p> <ul style="list-style-type: none"> <li>• Image: img</li> <li>• Image's name: label</li> <li>• Image's type: dropdown list</li> <li>• "Cancel": button</li> </ul>
2	<p>Administrator fills out the form</p> <p>[Alternative 1]</p>	
3	<p>Administrator clicks on "SAVE" button</p>	<p>Validate data</p> <p>[Exception 1, 2, 3, 5]</p> <p>Model's information is updated and saved to database.</p> <p>Model's images are updated in database and in "uploads" folder at server.</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	System shows dialog for administrator to choose local image
2	Administrator chooses an image or multi images	<p>Validate file type</p> <p>[Exception 4]</p> <p>System shows the selected image on the view with following data: thumb image, image name with image type, image size, image type dropdown list, cancel button and checkbox</p>
3	Administrator delete an image or multi images	System will delete images on the view

#### **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 and not duplicated in database.
- Each of local image which is imported by administrator goes with an image type: dropdown list, image name with type, image size, cancel button at the same time
- File which is imported by administrator must be type of images: \*.jpeg, \*.jpg, \*.png, \*.gif.
- In case of success scenario, the selected model will be updated. System will create new images record in database and also in “uploads” folder if any, system will update the existed image if any (update image type) and system will delete images in database as well as “uploads” folder if any.
- Input product series, series title and manufacturer are match with records in database. If new, then save to database
- Model's images is saved in folder “uploads” with file name of model's name go with auto generated id

#### 2.3.4.5. < Administrator > Delete model Usecase

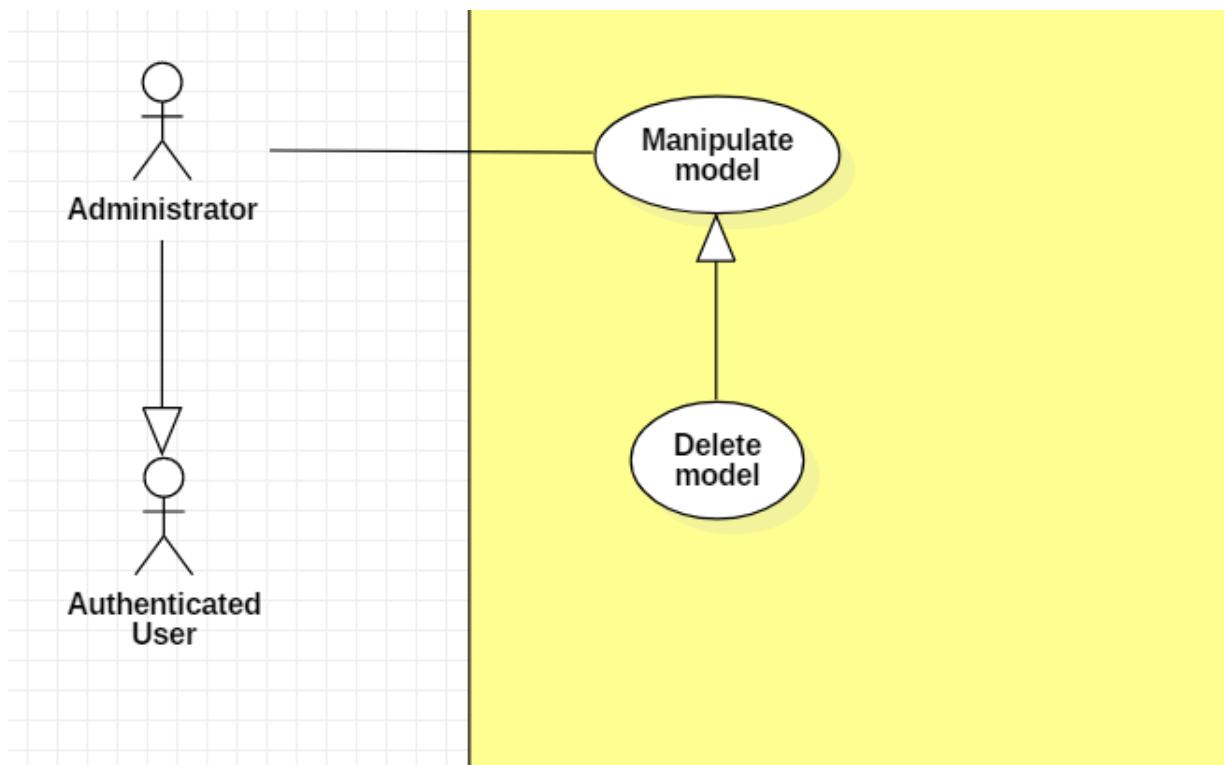


Figure 28 - < Administrator > Delete model Usecase

USE CASE – UC _Admin_05			
<b>Usecase No.</b>	Admin_05	<b>Usecase Version</b>	1.0
<b>Usecase Name</b>	Delete a model		
<b>Author</b>	TungHNM		
<b>Date</b>	30/09/2018	<b>Priority</b>	Normal
<b>Actor:</b>	<ul style="list-style-type: none"> <li>- Administrator.</li> </ul>		
<b>Summary:</b>	<ul style="list-style-type: none"> <li>- This use case allows administrator to delete a gundam model.</li> </ul>		
<b>Goal:</b>	<ul style="list-style-type: none"> <li>- Models which are no longer available will be removed from system and will not be displayed on the user view.</li> </ul>		

**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"><li>• Yes</li><li>• No</li></ul>
2	Administrator clicks on “Yes” button  [Alternative 1]	<p>Model information is deleted in database.</p> <p>Model’s images are deleted in folder “uploads”</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	<p>Administrator clicks on “No” button</p>	<p>The dialog disappears and system views detail page of the selected gunpla model, automatically fills out all fields.</p>
<b>Exceptions:</b> N/A		
<b>Relationships:</b> N/A		
<b>Business Rules:</b> N/A		

#### 2.3.4.6. < Administrator > Start/Stop schedule Usecase

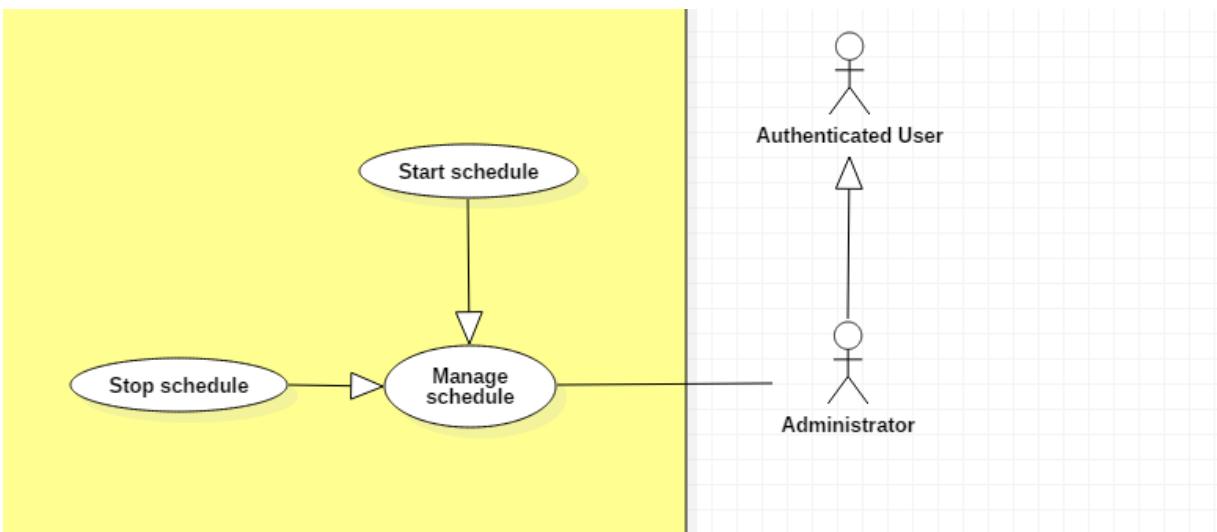


Figure 29 - < Administrator > Start/Stop schedule Usecase

USE CASE – UC _Admin_06			
<b>Usecase No.</b>	Admin_06	<b>Usecase Version</b>	1.0
<b>Usecase Name</b>	Start/Stop schedule		
<b>Author</b>	TungHNM		

<b>Date</b>	30/09/2018	<b>Priority</b>	Normal
-------------	------------	-----------------	--------

**Actor:**

- Administrator.

**Summary:**

- This use case allows administrator to start/stop schedule. There are 3 types of schedules: schedule auto crawl model and article, schedule disable events which don't have enough attendee after end registration date and schedule reject all order requests which don't have any responses from trade post owner in 2 days.

**Goal:**

- Administrator is able to control the schedule with input cycle and schedule will auto update system data after the input interval.

**Triggers:**

- Administrator sends start/stop schedule command to the system.

**Preconditions:**

- User must login into system with role Administrator.
- The manage schedule page is displayed.

**Post Conditions:**

- **Success:** The selected schedule will start with input cycle (hours) or stop
- **Fail:** Show error message.

**Main Success Scenario:**

Step	Actor Action	System Response
1	Administrator clicks on start button	A dialog is displayed with following item: <ul style="list-style-type: none"> <li>• Cycle: text input, required</li> </ul>
2	Administrator input cycle hour with type of Integer	Validate.  [Exception 1, 2, 3]  System displays a dialog for administrator to

		<p>confirm, the dialog is displayed with message: Are you sure ?, with 2 buttons:</p> <ul style="list-style-type: none"> <li>● Yes</li> <li>● No</li> </ul>
3	Administrator clicks on “Yes” button  [Alternative 1]	<p>The selected schedule will start to run at server with the input cycle!</p> <p>System displays message: “Schedule {schedule name} is running successfully”</p>
4	Administrator clicks “Stop” button	<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"> <li>● Yes</li> <li>● No</li> </ul>
5	Administrator clicks on “Yes” button  [Alternative 1]	<p>The selected schedule is interrupted.</p> <p>System displays message: “Schedule {schedule name} is stopped successfully”</p>

#### **Alternative Scenario:**

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

<b>Step</b>	<b>Actor Action</b>	<b>System Response</b>
1	Administrator clicks on “No” button	<p>The dialog disappears and system views the</p>

		manage schedule page
--	--	----------------------

**Exceptions:**

No	Cause	System Response
1	Out of range input cycle	System display error message to the view
2	Is not integer	System display error message to the view
3	Missing required field	System display error message to the view

**Relationships:** N/A

**Business Rules:**

- Schedule will start with thread and auto run on the server with the infinity loop, sleep for input interval (cycle) from administrators before execute update process and will be stopped when admin stops.
- Cycle must be larger than 0 and lower than 480 (480 minutes = 20 days)

#### 2.3.4.7. < Administrator > Track schedule progress Usecase

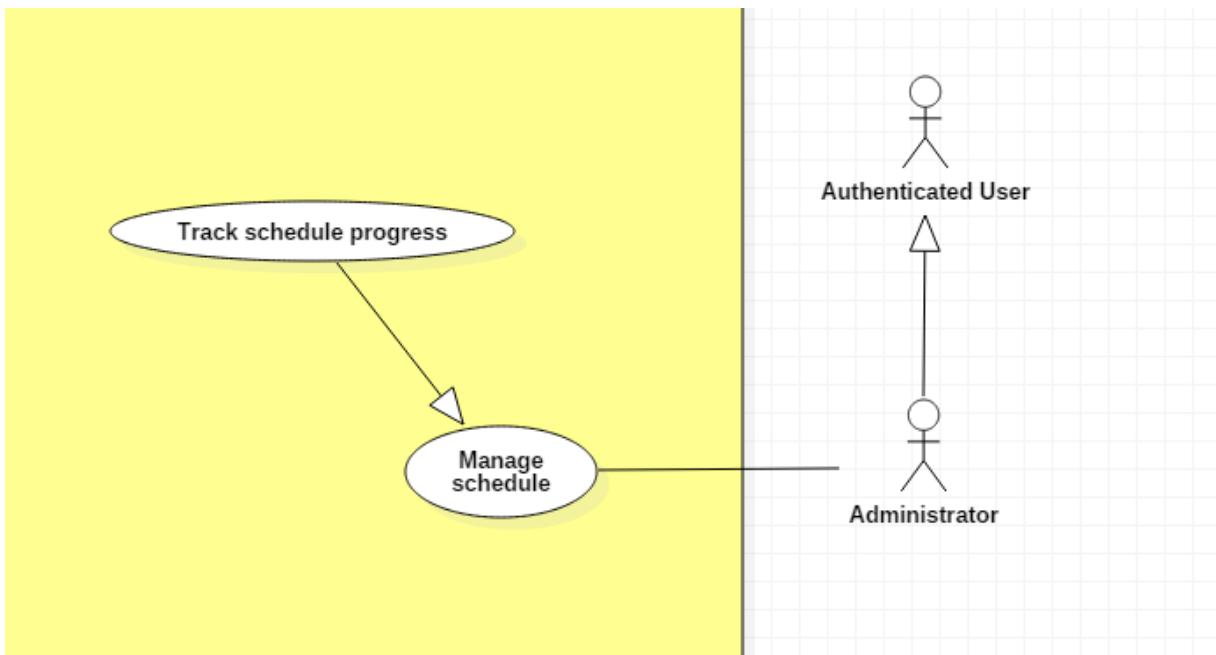


Figure 30 - < Administrator > Track schedule progress Usecase

USE CASE – UC _Admin_07			
<b>Usecase No.</b>	Admin_07	<b>Usecase Version</b>	1.0
<b>Usecase Name</b>	Track schedule progress		
<b>Author</b>	TungHNM		
<b>Date</b>	30/09/2018	<b>Priority</b>	Normal
<b>Actor:</b>	<ul style="list-style-type: none"> <li>- Administrator.</li> </ul>		
<b>Summary:</b>	<ul style="list-style-type: none"> <li>- This use case allows administrator to track the schedules progress with countdown time and data in log file.</li> </ul>		
<b>Goal:</b>	<ul style="list-style-type: none"> <li>- Administrator is able to track the schedule, the exact time that each schedule executes update process.</li> </ul>		
<b>Triggers:</b>			

- Administrator goes to manage schedule view.

**Preconditions:**

- User must login into system with role Administrator.
- The manage schedule page is displayed.

**Post Conditions:**

- **Success:** If the schedule is running, then system will display countdown time with data in log file and schedule's status
- **Fail:** Show error message.

**Main Success Scenario:**

Step	Actor Action	System Response
1	Administrator goes to manage schedule view	<p>System displays schedule name, status, and data in log file</p> <p>If the schedule is running, countdown time will display</p>

**Exceptions:** N/A

**Relationships:** N/A

**Business Rules:**

- Countdown time will reset to the last input cycle of administrator after countdown finishes.

### 2.3.4.8. < Administrator > Create new event Usecase

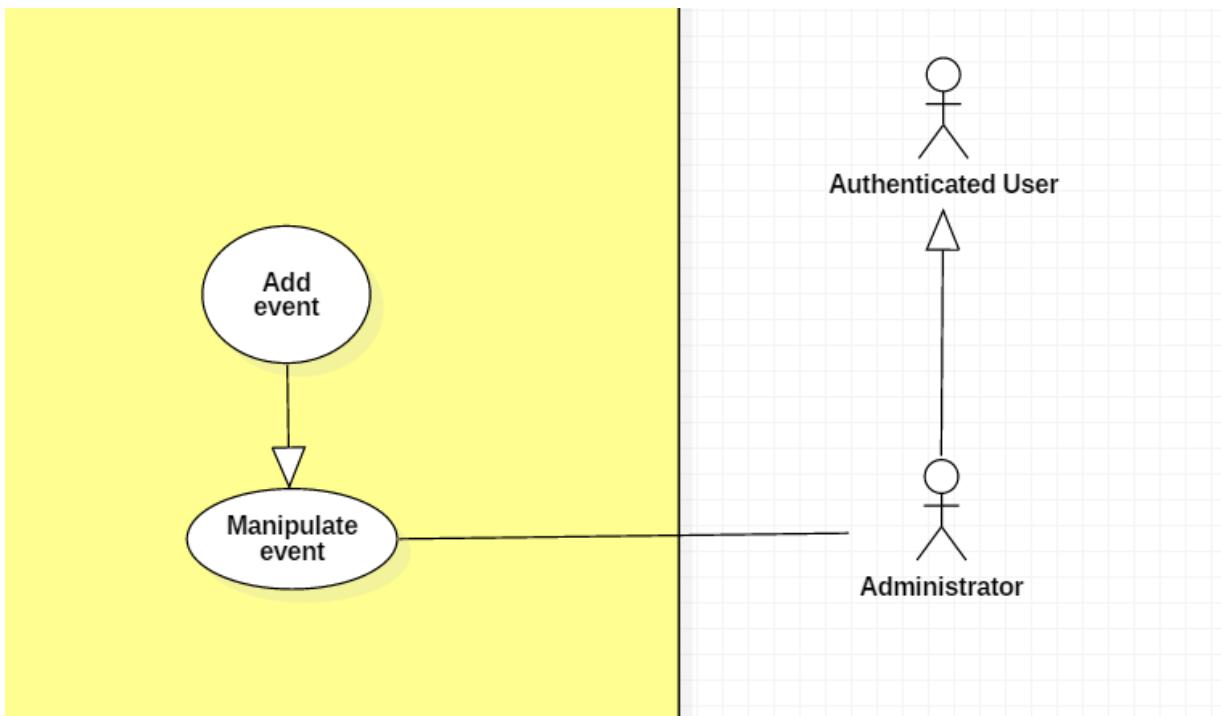


Figure 31 - < Administrator > Create new event Usecase

USE CASE – UC _Admin_08			
<b>Usecase No.</b>	Admin_08	<b>Usecase Version</b>	1.0
<b>Usecase Name</b>	Create new event		
<b>Author</b>	KhanhNB		
<b>Date</b>	25/09/2018	<b>Priority</b>	Normal
<b>Actor:</b>			
- Administrator			
<b>Summary:</b>			
- This use case allows admin to post a new event			
<b>Goal:</b>			
- The event is posted on the website			
<b>Triggers:</b>			
- 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"><li>• Input field for the event’s Title</li><li>• Input field for the event’s location</li><li>• A textarea for the event’s description</li><li>• Input image for the thumbnail</li><li>• Input field for the event’s start date</li><li>• Input field for the event’s end date</li><li>• Input field for the event’s registration start date</li><li>• Input field for the event’s registration end date</li><li>• Input field for the event’s ticket price</li><li>• A textarea using CKEditor library for the event’s content</li><li>• Dropdown list for the event’s status, values “Active”, “Inactive”, “Finished”</li><li>• Submit button</li></ul>
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 title	System shows error message to ask the admin to enter missing required fields.

2	Admin submit with empty location	System shows error message to ask the admin to enters missing required fields.
3	Admin submit with empty description	System shows error message to ask the admin to enters missing required fields.
4	Admin submit with empty max attendee	System shows error message to ask the admin to enters missing required fields.
5	Admin submit with empty min attendee	System shows error message to ask the admin to enters missing required fields.
6	Admin submit with empty content	System shows error message to ask the admin to enters missing required fields.
7	Admin submit with no thumbnail uploaded	System shows error message to ask the admin to enters missing required fields.
8	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.
- The registration dates validation formula:

Registration start date < Registration end date < Event start date < Event end date

### 2.3.4.9. < Administrator > Edit event Usecase

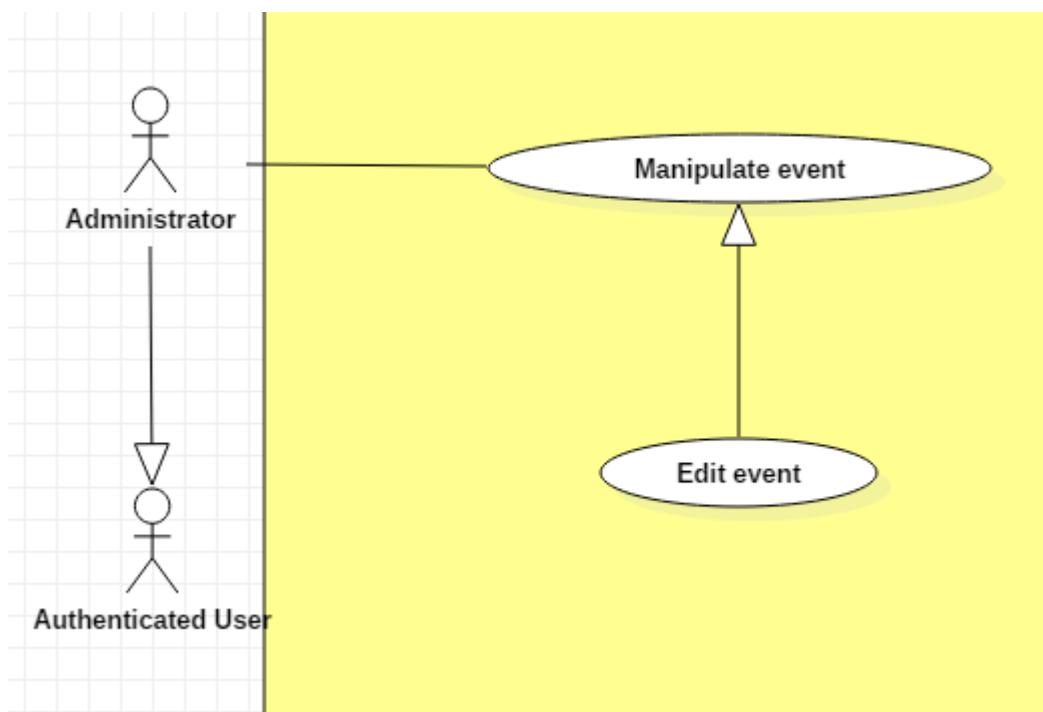


Figure 32 - < Administrator > Edit event Usecase

USE CASE – UC _Admin_09			
Usecase No.	Admin_09	Usecase Version	1.0
Usecase Name	Update event		
Author	KhanhNB		
Date	25/09/2018	Priority	Normal
<b>Actor:</b>	<ul style="list-style-type: none"> <li>- Administrator</li> </ul>		
<b>Summary:</b>	<ul style="list-style-type: none"> <li>- This use case allows admin to update an event's info.</li> </ul>		
<b>Goal:</b>	<ul style="list-style-type: none"> <li>- The event is updated in database</li> </ul>		
<b>Triggers:</b>	<ul style="list-style-type: none"> <li>- Admin edit an event and submit it.</li> </ul>		

**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 existing data:</p> <ul style="list-style-type: none"><li>• Input field for the event’s Title</li><li>• Input field for the event’s location</li><li>• A textarea for the event’s description</li><li>• Input image for the thumbnail</li><li>• Input field for the event’s start date</li><li>• Input field for the event’s end date</li><li>• Input field for the event’s registration start date</li><li>• Input field for the event’s registration end date</li><li>• Input field for the event’s ticket price</li><li>• A textarea using CKEditor library for the event’s content</li><li>• Dropdown list for the event’s status, values “Active”, “Inactive”, “Finished”</li><li>• Submit button</li></ul>
2	Admin fills out the form.	
3	Admin submit event.	<p>Validate data.</p> <p>System update the article in the database</p>

**Alternative Scenario:**

N/A

**Exceptions:**

No	Cause	System Response

1	Admin submit with empty title	System shows error message to ask the admin to enters missing required fields.
2	Admin submit with empty location	System shows error message to ask the admin to enters missing required fields.
3	Admin submit with empty description	System shows error message to ask the admin to enters missing required fields.
4	Admin submit with empty max attendee	System shows error message to ask the admin to enters missing required fields.
5	Admin submit with empty min attendee	System shows error message to ask the admin to enters missing required fields.
6	Admin submit with empty content	System shows error message to ask the admin to enters missing required fields.
7	Admin upload file too large.	System shows error message to ask the admin to upload correct required file size.

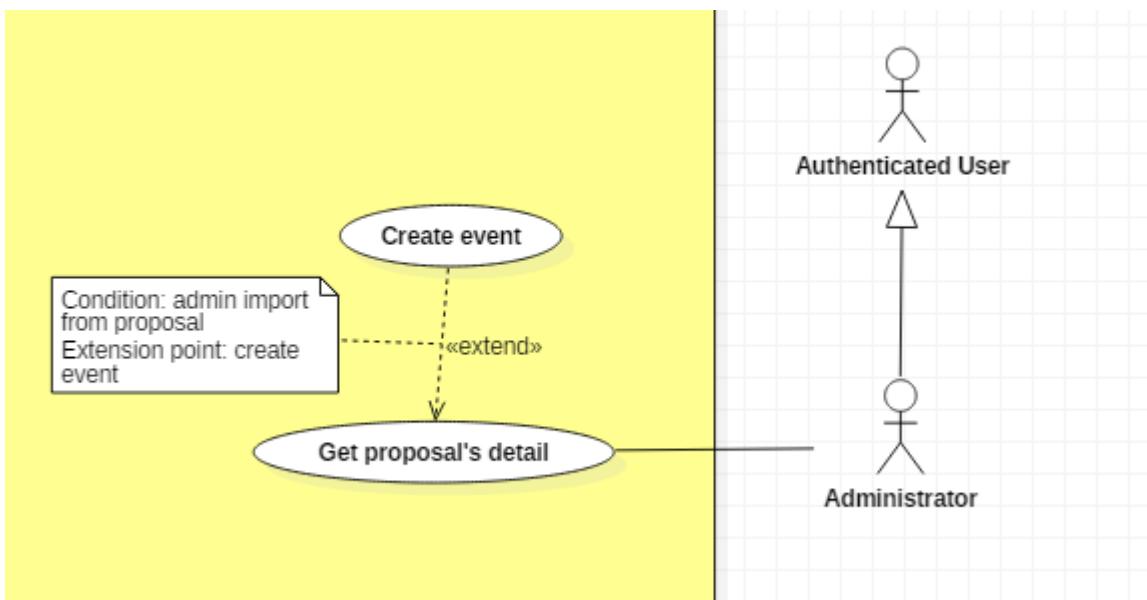
**Relationships:** N/A

**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.
- The registration dates validation formula:

Registration start date < Registration end date < Event start date < Event end date

### 2.3.4.10. < Administrator > Get proposal's detail Usecase



**Figure 33 - < Administrator > Get proposal's detail Usecase**

<b>USE CASE – UC _Admin_10</b>			
<b>Usecase No.</b>	Admin_10	<b>Usecase Version</b>	1.0
<b>Usecase Name</b>	Get proposal's detail		
<b>Author</b>	KhanhNB		
<b>Date</b>	25/09/2018	<b>Priority</b>	Normal
<b>Actor:</b>	<ul style="list-style-type: none"> <li>- Administrator</li> </ul>		
<b>Summary:</b>	<ul style="list-style-type: none"> <li>- This use case allows admin to create an event from a proposal</li> </ul>		
<b>Goal:</b>	<ul style="list-style-type: none"> <li>- The proposal data is loaded to the interface</li> </ul>		
<b>Triggers:</b>	<ul style="list-style-type: none"> <li>- Admin selects “Create event from this proposal” while viewing a proposal</li> </ul>		
<b>Preconditions:</b>	<ul style="list-style-type: none"> <li>- User must login into system with role Administrator.</li> </ul>		

**Post Conditions:**

- **Success:** The proposal data is loaded to the interface
- **Fail:** Show error message.

**Main Success Scenario:**

Step	Actor Action	System Response
1	Admin go to “Proposal detail” page	“Proposal detail” view is shown with following labels and fields: <ul style="list-style-type: none"><li>• Title of the event</li><li>• Description of the event</li><li>• Location of the event</li><li>• Content of the event</li></ul>
2	Admin select “Create event from this proposal”	System redirect the admin to “Create new event” view with the following fields with proposal’s data: <ul style="list-style-type: none"><li>• Title of the event</li><li>• Description of the event</li><li>• Location of the event</li><li>• Content of the event</li></ul>

**Alternative Scenario:**

N/A

**Exceptions:** N/A**Relationships:** N/A**Business Rules:** N/A

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

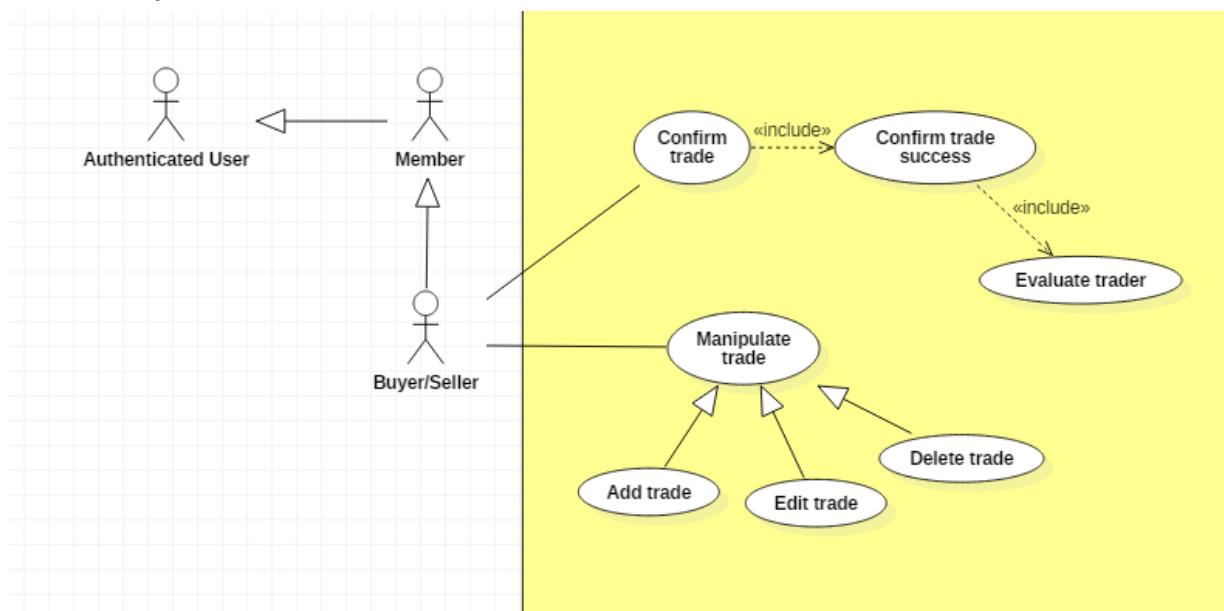


Figure 34 - < Buyer/Seller > Overview Usecase

#### 2.3.5.1. < Buyer/Seller > Add trade Usecase

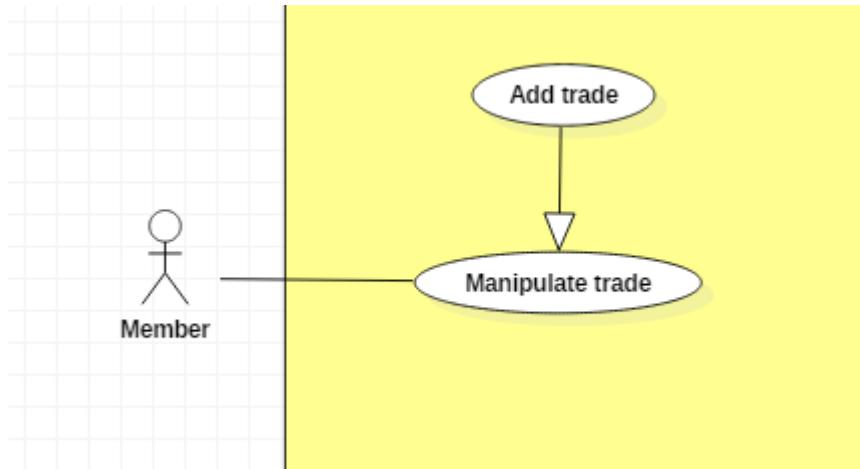


Figure 35 - < Buyer/Seller > Add trade Usecase

USE CASE – UC _BuyerSeller_01			
Usecase No.	BuyerSeller_01	Usecase Version	1.0
Usecase Name	Add trade		
Author	SangLNN		

<b>Date</b>	25/09/2018	<b>Priority</b>	Normal
-------------	------------	-----------------	--------

**Actor:**

- Buyer/Seller

**Summary:**

- This use case allows buyer/seller 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 Buyer/Seller.

**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"> <li>• Type of Trade: radio button , required, 2 options: “I want to sell” and “I want to buy”.</li> <li>• Title: text input, required, length 7 – 250.</li> <li>• Photos: file input, required, number of file 1 – 5, maximum size per file is 2.5mb.</li> <li>• Condition : radio button , required, 2 options: “New” and “Used”.</li> <li>• Price: text, required, length 1</li> </ul>

		<p>– 7.</p> <ul style="list-style-type: none"> <li>• Quantity: number, required, positive integer and below than 1000.</li> <li>• Brand: text input, required, length 3 – 50.</li> <li>• Model: text input, required, length 3 – 50.</li> <li>• Description : text input, required, length 20 – 5000.</li> </ul> <p>Create trade post owner's information view is shown with following labels and fields:</p> <ul style="list-style-type: none"> <li>• Full name: text input, required, length 3 – 50.</li> <li>• Email: text input, readonly.</li> <li>• Phone Number: text input, required, length 7 – 12.</li> <li>• Address: text input, readonly.</li> <li>• Auto get your location: Button</li> </ul> <p>[Alternative Scenario 1]</p> <ul style="list-style-type: none"> <li>• Change your location: Button</li> </ul> <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>

		System shows message trade post has been created successfully and waiting to administrators approve.
--	--	------------------------------------------------------------------------------------------------------

**Alternative Scenario:**

**Alternative Scenario 1: Auto get your location Scenario.**

Step	Actor Action	System Response
1	User click “Auto get your location” button to send get your location command.	<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"> <li>• Province/City: select option, required.</li> <li>• District: select option, required.</li> <li>• Ward: select option, required.</li> <li>• Street name: text input, required, length 3 - 50.</li> <li>• Apartment number: text input, required, length 3 – 50.</li> <li>• Your full address: text input,</li> </ul>

		readonly.  And “submit” modal button.
2	User fills out the form.	
3	User click “submit” button.	System close modal and get text data from “Your full address” field and fill to “Address” field on trade post owner’s information form.

**Exceptions:**

No	Cause	System Response
1	User sends post new trade command by click “Post your trade” without filling out full form.	System shows error message to ask member enters missing required fields.
2	User upload file too large.	System shows error message to ask member upload correct required 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 permission 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

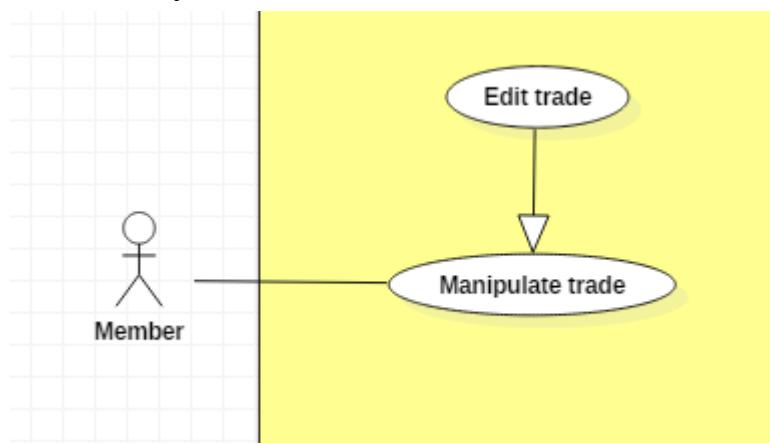


Figure 36 - < Buyer/Seller > Edit trade Usecase

USE CASE – UC _BuyerSeller_02			
Usecase No.	BuyerSeller_02	Usecase Version	1.0
Usecase Name	Edit trade		
Author	SangLNN		
Date	26/09/2018	Priority	Normal

**Actor:**

- Buyer/Seller

**Summary:**

- This use case allows buyer/seller 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"><li>• Type of Trade: radio button , required, 2 option: “I want to sell” and “I want to buy”.</li><li>• Title: text input, required, length 7 – 250.</li><li>• Photos: file input, required, number of file 1 – 5, maximum size per file is 2.5mb.</li><li>• Condition : radio button , required, 2 option: “New” and “Used”.</li><li>• Price: text, required, length 1 – 7.</li><li>• Quantity: number, required, positive integer and below than 1000.</li></ul>

		<ul style="list-style-type: none"> <li>• Brand: text input, required, length 3 – 50.</li> <li>• Model: text input, required, length 3 – 50.</li> <li>• Description : text input, required, length 20 – 5000.</li> </ul> <p>Create trader information view is show with following labels and fields:</p> <ul style="list-style-type: none"> <li>• Full name: text input, required, length 3 – 50.</li> <li>• Email: text input, readonly.</li> <li>• Phone Number: text input, required, length 7 – 12.</li> <li>• Address: text input, readonly.</li> <li>• Auto get your location: Button</li> </ul> <p>[Alternative Scenario 1]</p> <ul style="list-style-type: none"> <li>• Change your location: Button</li> </ul> <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>

		System shows message trade post has been updated successfully and waiting to administrators approve.
--	--	------------------------------------------------------------------------------------------------------

**Alternative Scenario: N/A**

**Alternative Scenario:**

**Alternative Scenario 1: Auto get your location Scenario.**

Step	Actor Action	System Response
1	User click “Auto get your location” button to send get your location command.	<p>[Exception 3]</p> <p>System auto get your location and generated to text address then autoform 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"> <li>• Province/City: select option, required.</li> <li>• District: select option, required.</li> <li>• Ward: select option, required.</li> <li>• Street name: text input, required, length 3 - 50.</li> <li>• Apartment number: text input, required, length 3 – 50.</li> <li>• Your full address: text input,</li> </ul>

		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 required 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

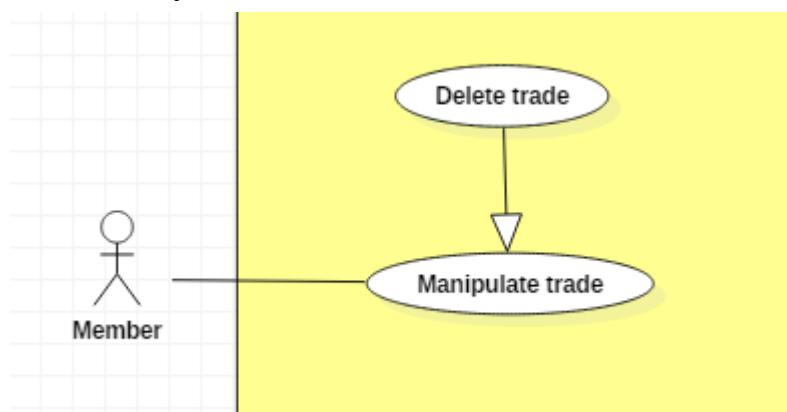


Figure 37 - < Buyer/Seller > Delete trade Usecase

USE CASE – UC _BuyerSeller_03			
Usecase No.	BuyerSeller_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.	The trade post has selected is deleted from database. System close modal and show message trade post has been deleted successfully.

	User cancel delete by click “No”.	System close modal and nothing has changed.
--	-----------------------------------	---------------------------------------------

**Alternative Scenario:** N/A

**Exceptions:** N/A

**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

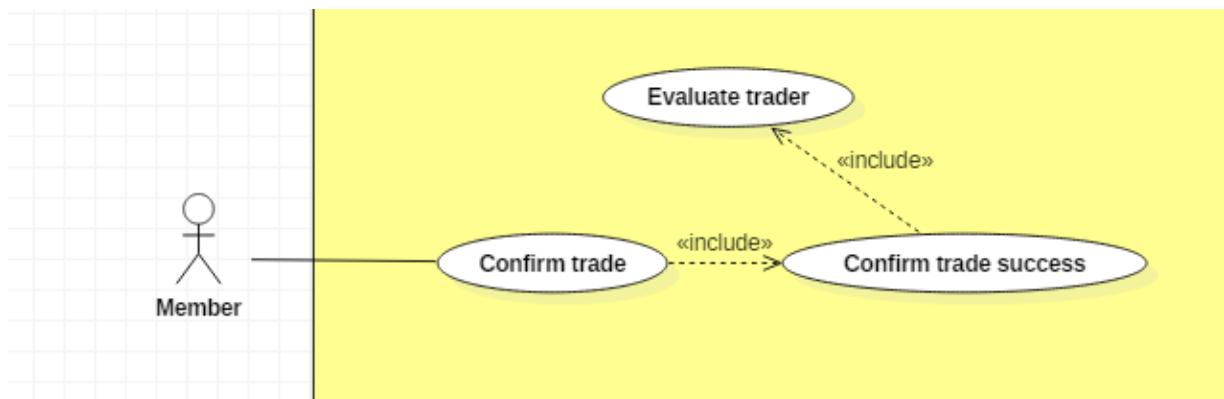


Figure 38 - < Buyer/Seller > Confirm trade Usecase

USE CASE – UC _BuyerSeller_04			
<b>Usecase No.</b>	BuyerSeller_04	<b>Usecase Version</b>	1.0
<b>Usecase Name</b>	Confirm trade		
<b>Author</b>	SangLNN		
<b>Date</b>	26/09/2018	<b>Priority</b>	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 buyer/seller.
- 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:**

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

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

<b>Step</b>	<b>Actor Action</b>	<b>System Response</b>
-------------	---------------------	------------------------

1	If buyer/seller decline request by click “decline” button.	<p>System create and show modal with following labels and fields:</p> <ul style="list-style-type: none"> <li>• Reason: text input, length 10 - 250.</li> </ul> <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

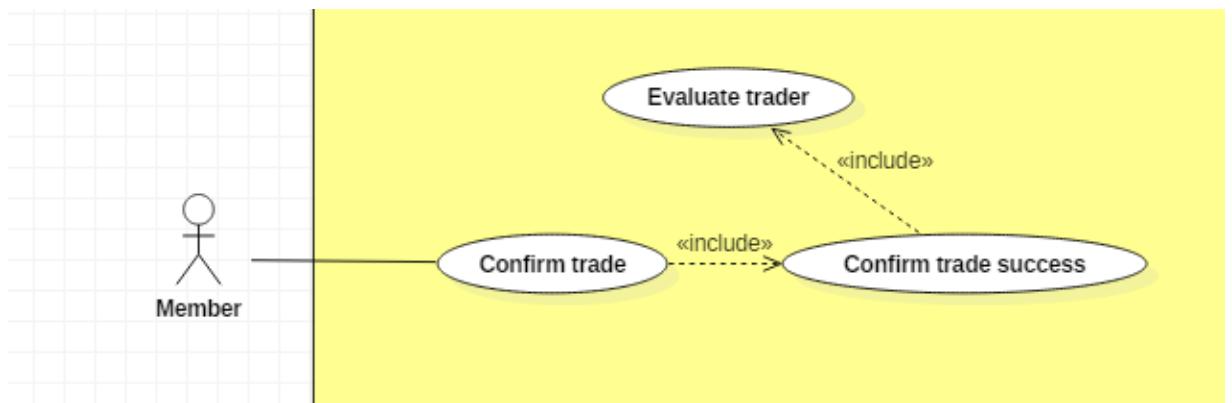


Figure 39 - < Buyer/Seller > Confirm trade success Usecase

**USE CASE – UC \_BuyerSeller\_05**

<b>Usecase No.</b>	BuyerSeller_05	<b>Usecase Version</b>	1.0
<b>Usecase Name</b>	Confirm trade success		
<b>Author</b>	SangLNN		
<b>Date</b>	26/09/2018	<b>Priority</b>	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 buyer/seller.
- User have at least one request order accepted.

**Post Conditions:**

- **Success:** System show message traded successfully and give evaluate 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..  [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” to send confirm trade success command.  [Alternative Scenario 2]	System close dialog and set order state to succeed 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” button on payment list.	<p>System create and show modal with following labels and fields:</p> <ul style="list-style-type: none"> <li>• Reason: text input, length 10 - 250.</li> </ul> <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 order has been cancelled with reason.</p>

Alternative Scenario 2: Confirm by click “no”

Step	Actor Action	System Response

1	User confirm by click “No”	System close dialog and nothing has changed.
<b>Exceptions:</b> N/A		
<b>Relationships:</b> N/A		
<b>Business Rules:</b> N/A		

### 2.3.6. < Handler > Overview Usecase

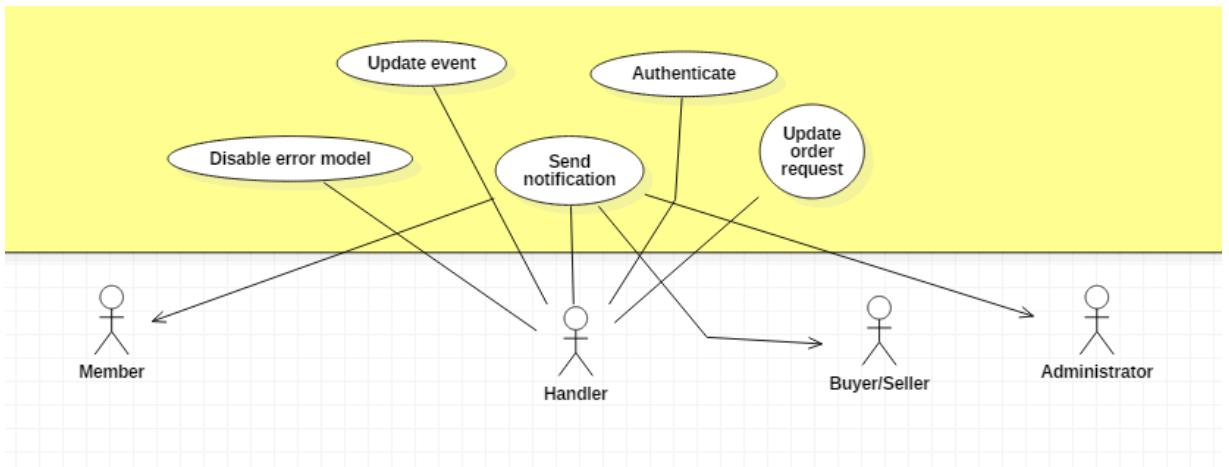


Figure 40 - < Handler > Overview Usecase

#### 2.3.6.1. < Handler > Send notification Usecase

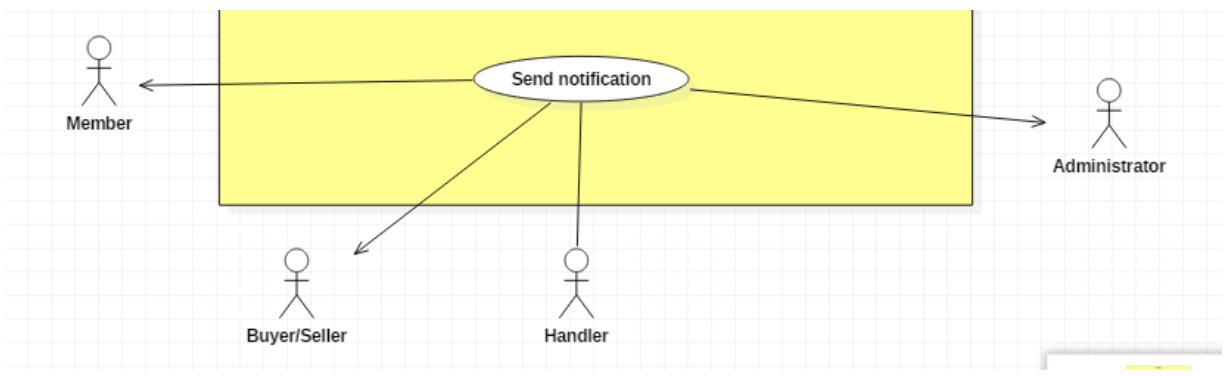


Figure 41 - < Handler > Send notification Usecase

USE CASE – UC _Handler_01			
Usecase No.	Handler_01	Usecase Version	1.0

<b>Usecase Name</b>	Send notification								
<b>Author</b>	TungHNM								
<b>Date</b>	20/11/2018	<b>Priority</b>	Normal						
<b>Actor:</b>									
<ul style="list-style-type: none"> <li>- Handler.</li> </ul>									
<b>Summary:</b>									
<ul style="list-style-type: none"> <li>- Handler will notify notifications to users through Firebase Cloud Messaging service.</li> </ul>									
<b>Goal:</b>									
<ul style="list-style-type: none"> <li>- Make it more convenient for user to have information.</li> </ul>									
<b>Triggers:</b>									
<ul style="list-style-type: none"> <li>- To profile's owner: Administrator update user's profile with reason</li> <li>- To administrator: Error 404 loading model's images</li> <li>- To article's author: Administrator approves user's article</li> <li>- To event's attendees: Event is cancelled (notification to event's attendee)</li> <li>- To trader: approve/decline/cancel/rating/confirm success order</li> <li>- To trade post owner: Administrator approve/decline trade post request</li> </ul>									
<b>Preconditions:</b> N/A									
<b>Post Conditions:</b>									
<ul style="list-style-type: none"> <li>- <b>Success:</b> Notification is sent to user.</li> <li>- <b>Fail:</b> Log error to console.</li> </ul>									
<b>Main Success Scenario:</b>									
<table border="1"> <thead> <tr> <th>Step</th><th>Actor Action</th><th>System Response</th></tr> </thead> <tbody> <tr> <td>1</td><td>Handler sends send notification command</td><td>System sends notification to user</td></tr> </tbody> </table>				Step	Actor Action	System Response	1	Handler sends send notification command	System sends notification to user
Step	Actor Action	System Response							
1	Handler sends send notification command	System sends notification to user							
<b>Alternative Scenario:</b> N/A									
<b>Exceptions:</b> N/A									

**Relationships:** N/A

**Business Rules:**

- In case of success scenario, system will create new notification in database with status “not seen” and push notification to user’s devices which include web browser and mobile through Firebase Cloud Messaging service

#### 2.3.6.2. < Handler > Disable error model Usecase

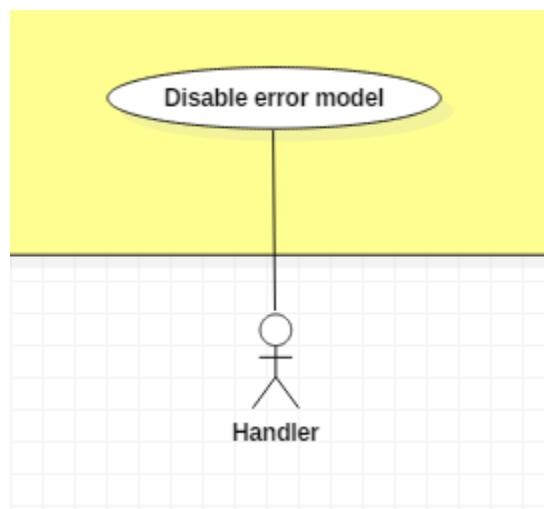


Figure 42 - < Handler > Disable error model Usecase

USE CASE – UC _Handler_02			
<b>Usecase No.</b>	Handler_02	<b>Usecase Version</b>	1.0
<b>Usecase Name</b>	Disable error model		
<b>Author</b>	TungHNM		
<b>Date</b>	25/10/2018	<b>Priority</b>	Normal

**Actor:**

- Handler.

**Summary:**

- Handler will notify notifications to administrator when any model loads with error 404 image.

**Goal:**

- Change status of error loading images model to “Unavailable” so as to avoid displaying error model to user’s view and send notification to administrator to update

**Triggers:**

- Member view model’s detail, model’s images are loading with error 404 url.

**Preconditions:** N/A**Post Conditions:**

- **Success:** Notification is sent to administrator and the error model’s status is auto changed to “Unavailable”.
- **Fail:** Log error to console.

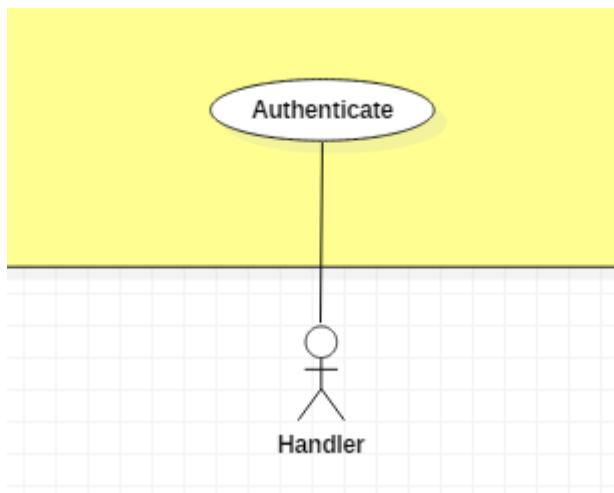
**Main Success Scenario:**

Step	Actor Action	System Response
1	Handler sends error model command	System auto changes the model’s status to “Unavailable” and redirects user back to search model page

**Alternative Scenario:** N/A**Exceptions:** N/A**Relationships:** N/A**Business Rules:**

- Model is called error model when the number of 404 error loading images is greater than 30% of the total model’s images
- System will disable error model to avoid displaying on user’s view

### 2.3.6.3. < Handler > Authenticate Usecase



**Figure 43 - < Handler > Authenticate Usecase**

USE CASE – UC _Handler_03			
<b>Usecase No.</b>	Handler_03	<b>Usecase Version</b>	1.0
<b>Usecase Name</b>	Authenticate		
<b>Author</b>	TungHNM		
<b>Date</b>	30/09/2018	<b>Priority</b>	Normal
<b>Actor:</b>			
- Handler.			
<b>Summary:</b>			
- Authentication and authorization			
<b>Goal:</b>			
- Authentication and authorization			
<b>Triggers:</b>			
- User logins to system.			
<b>Preconditions:</b> N/A			
<b>Post Conditions:</b>			
- <b>Success:</b> System redirects user to the appropriate view depends on user's role			
- <b>Fail:</b> Log error to console.			

**Main Success Scenario:**

Step	Actor Action	System Response
1	Handler sends authentication and authorization command	System redirects user to appropriate view depends on user's role

**Alternative Scenario:** N/A

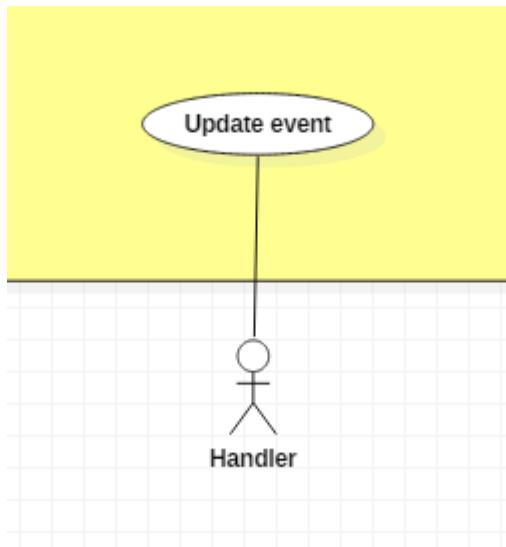
**Exceptions:** N/A

**Relationships:** N/A

**Business Rules:**

- If Admin, redirects to Web Admin
- If Member, redirects to Search Model Page
- If Buyer/Seller, redirects to My Trade Page

**2.3.6.4. < Handler > Update event Usecase**



**Figure 44 - < Handler > Update event Usecase**

**USE CASE – UC \_Handler\_04**

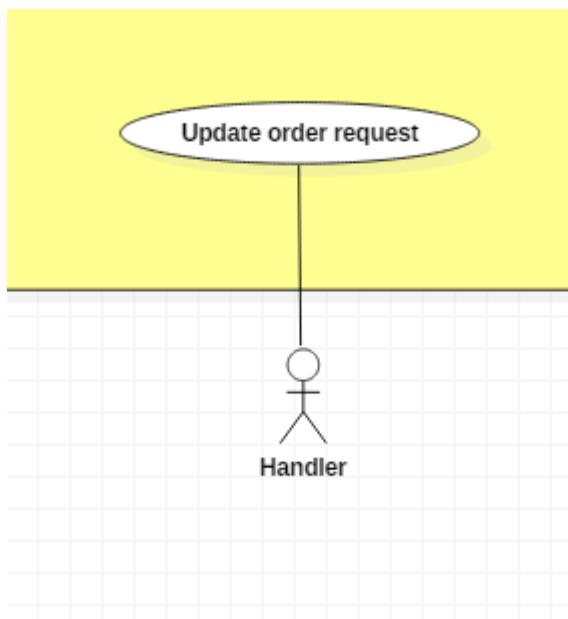
<b>Usecase No.</b>	Handler_04	<b>Usecase Version</b>	1.0
--------------------	------------	------------------------	-----

<b>Usecase Name</b>	Update event								
<b>Author</b>	TungHNM								
<b>Date</b>	30/09/2018	<b>Priority</b>	Normal						
<b>Actor:</b>									
<ul style="list-style-type: none"> <li>- Handler.</li> </ul>									
<b>Summary:</b>									
<ul style="list-style-type: none"> <li>- Events will be cancel if there is not enough registered attendee after end-registration date</li> </ul>									
<b>Goal:</b>									
<ul style="list-style-type: none"> <li>- Auto disable events which don't have enough attendee, make it more convenient for administrator to manage events.</li> </ul>									
<b>Triggers:</b>									
<ul style="list-style-type: none"> <li>- Schedule executes update event process</li> </ul>									
<b>Preconditions:</b> N/A									
<b>Post Conditions:</b>									
<ul style="list-style-type: none"> <li>- <b>Success:</b> Notification is sent to event's attendees if it is cancelled, event's status is changed to "Inactive"</li> <li>- <b>Fail:</b> Log error to console.</li> </ul>									
<b>Main Success Scenario:</b>									
<table border="1"> <thead> <tr> <th>Step</th><th>Actor Action</th><th>System Response</th></tr> </thead> <tbody> <tr> <td>1</td><td>Handler sends update event command</td><td>System auto changes the event's status which don't have enough attendee after registration date to "Inactive" and send notification to that event's attendees</td></tr> </tbody> </table>				Step	Actor Action	System Response	1	Handler sends update event command	System auto changes the event's status which don't have enough attendee after registration date to "Inactive" and send notification to that event's attendees
Step	Actor Action	System Response							
1	Handler sends update event command	System auto changes the event's status which don't have enough attendee after registration date to "Inactive" and send notification to that event's attendees							
<b>Alternative Scenario:</b> N/A									
<b>Exceptions:</b> N/A									

**Relationships:** N/A

**Business Rules:** N/A

#### 2.3.6.5. < Handler > Update order request Usecase



**Figure 45 - < Handler > Update order request Usecase**

USE CASE – UC _Handler_05			
<b>Usecase No.</b>	Handler_05	<b>Usecase Version</b>	1.0
<b>Usecase Name</b>	Update order request		
<b>Author</b>	TungHNM		
<b>Date</b>	30/09/2018	<b>Priority</b>	Normal

**Actor:**

- Handler.

**Summary:**

- All order requests which don't have any responses in 2 days will be rejected

**Goal:**

- Auto reject all order request which don't have any response in 2 days.

**Triggers:**

- Schedule executes update order request process

**Preconditions:** N/A**Post Conditions:**

- **Success:** Notification is sent to user whose order request is auto rejected by system.
- **Fail:** Log error to console.

**Main Success Scenario:**

Step	Actor Action	System Response
1	Handler sends update order request command	System auto changes the order request status which don't have any response after 2 days to rejected and sends notification to user.

**Alternative Scenario:** N/A**Exceptions:** N/A**Relationships:** N/A**Business Rules:** N/A

### 3. Software System Attribute

#### 3.1. Usability

- UI website is fit for each browser in each device

- o Font style: Helvetica, Arial, Helvetica Neue, Roboto, Arial, Droid Sans, sans-serif
  - o Font size: 12px - 26px
  - o Color: green, black, red, white, blue, Light Slate Grey, Fuego, Honeysuckle...
  - o Background: Whitesmoke, white
- UI mobile is scalable with each monitor of smart phone:
  - o Font size: 13-30pt
  - o Color: darkblue, black, ...
  - o Background: white, black...

### **3.2. Reliability**

- System crawls information from official international gunpla's websites, so data will be high accuracy.

### **3.2. Availability**

- System replies in maximum 2 seconds

### **3.3. Security**

- Each role of user has a specific permission to interact with the system
- System always checks for authorization and authentication before doing anything.
- Input data is validated before saving to database

### **3.4. Maintainability**

- The system is divided into separated layers with interface and modules for easy maintaining

### **3.5. Portability**

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

### **3.6. Performance**

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

- System sends email to the user in 10 seconds max
- User can view information with pagination which are saved at client-side

## 4. Conceptual Diagram

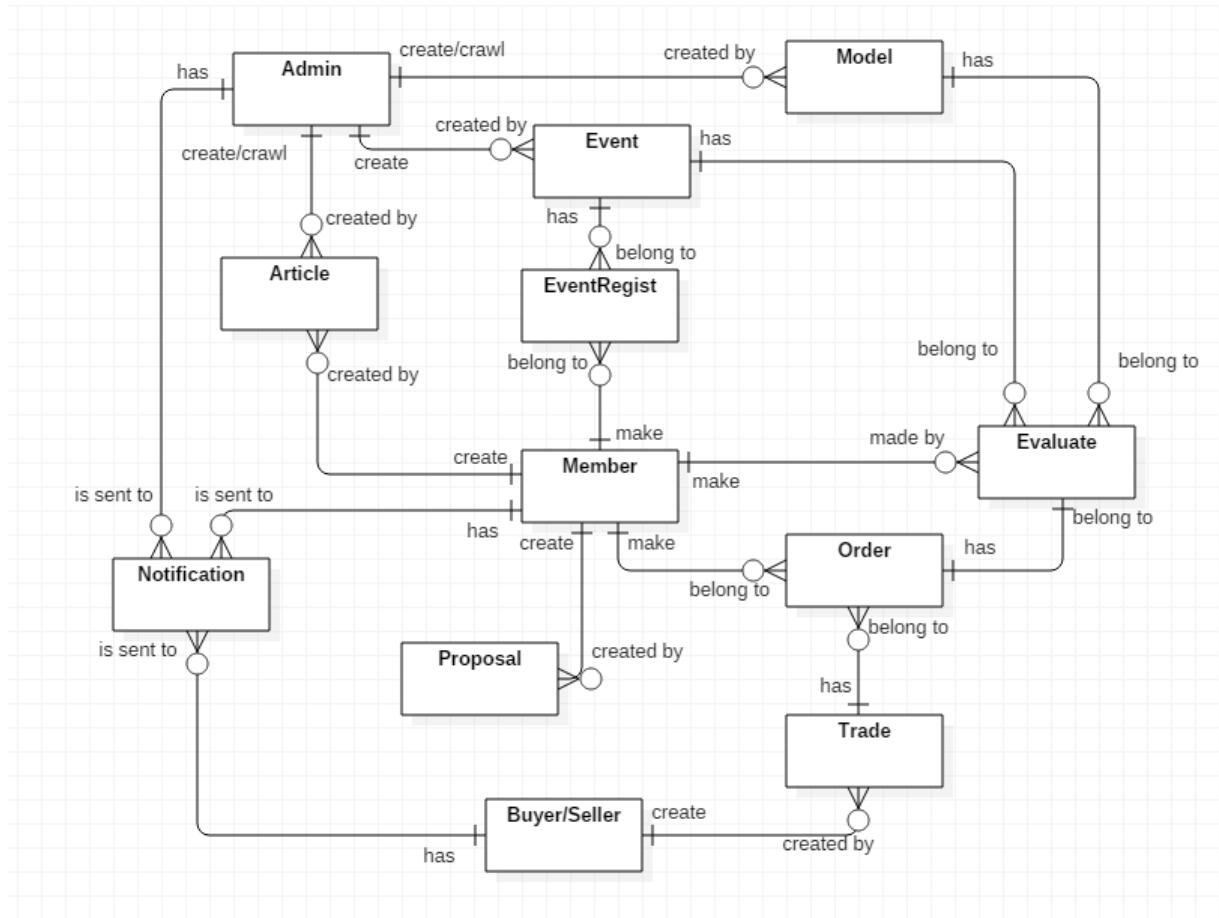


Figure 46 - Conceptual Diagram

## Data Dictionary

Entity Data Dictionary: describe all contents of all entities	
Entity Name	Description
Article	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/review comment about models, events and exchange, made by member
Notification	<p>Is sent to administrator when models have error 404 loading images</p> <p>Is sent to member when administrator update member's profile</p> <p>Is sent to users when their order request is accepted/declined/cancel/confirm exchange success or rating from trade post owner</p> <p>Is sent to buyer/seller when their exchange post is approved/decline by administrator</p> <p>Is sent to article's author when their article is approved/disapproved by administrator</p> <p>Is sent to event's attendees when the event is cancelled because of not enough attendee to start</p>
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, activity diagrams.

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

Document overview:

- Section 1: gives an overview design of the system
- 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

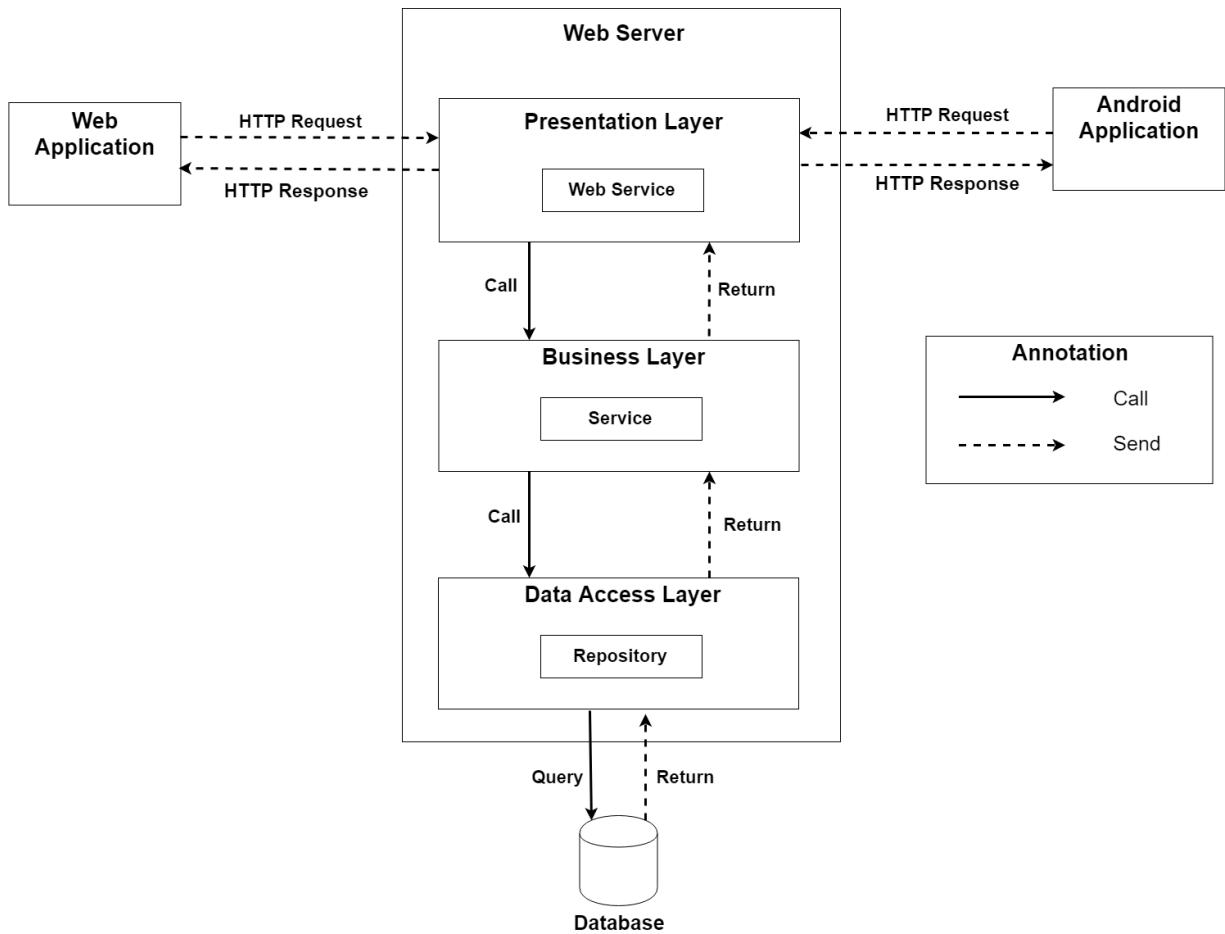


Figure 47 - System Architectural Design

### 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 anything 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 on web application. 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

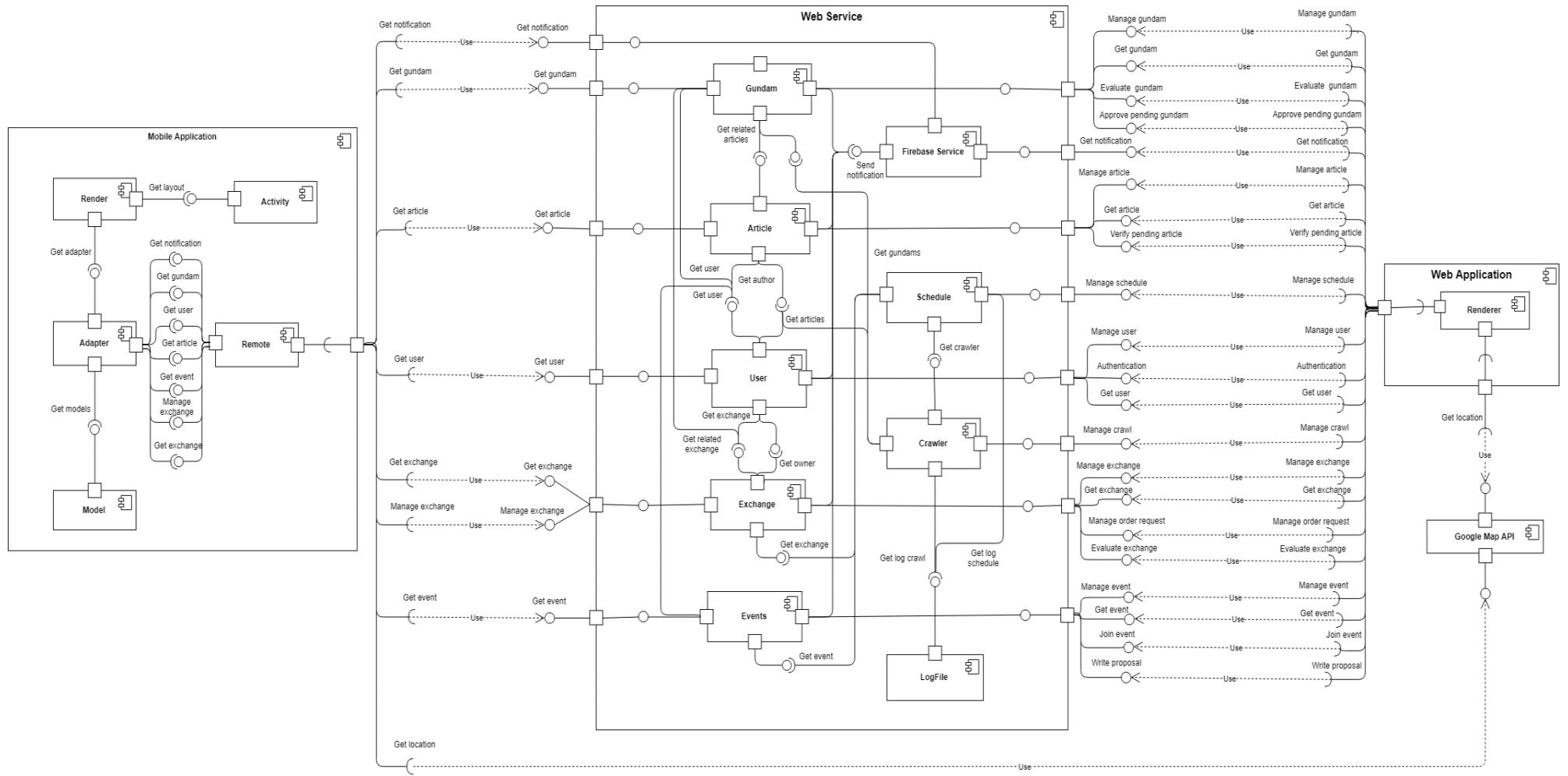


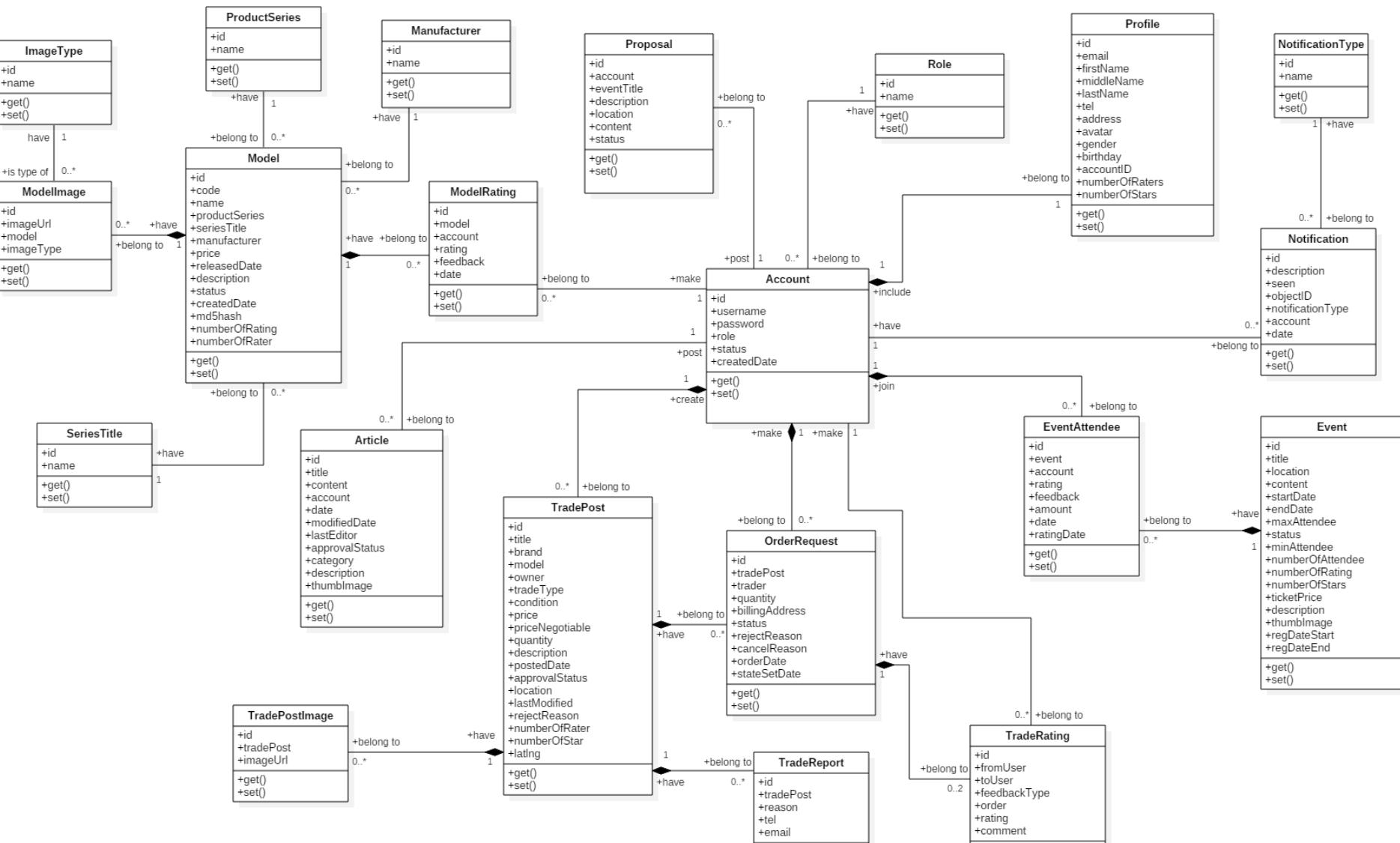
Figure 48 - Component Diagram

Component Dictionary: Describes components	
Web Service	Subsystem that contains business components and provides API for client to interact with the system
Web Application	Subsystem that contains client view in web application
Mobile Application	Subsystem to handle client activities in mobile application
Gundams Component	Component about manipulating gundams
Notification Component	Component about manipulating notification
Articles Component	Component about manipulating articles
Schedule Component	Component about manipulating schedule
Users Component	Component about manipulating users
Crawler Component	Component about manipulating crawling
Exchange Component	Component about manipulating exchange
Events Component	Component about manipulating events
LogFile Component	Component about reading from and writing to file log
Renderer Component	Component to render client web view with json data response from web service
Google Map API Component	Component about google map api which response location
Remote Component	Component that handles web service request and response
Model Component	Component that contains data access objects
Adapter Component	Component that contains view custom adapter
Activity Component	Component that contains activities in mobile application
Render Component	Component to render data from adapter to activity's layout view

Table 14 - Component Dictionary

## **4. Detailed Description**

### **4.1. Class Diagram**



**Figure 49 - Class Diagram**

<b>Class Name</b>	<b>Mapping column with Conceptual diagram</b>	<b>Description</b>
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
ModelRating	Evaluate	Contain model review rating information
Article	Article	Contain article 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

Table 15 - Class Dictionary

## 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
md5hash	String	Private	Model's md5 hash value
numberOfRating	int	Private	Model's number of star ratings
numberOfrater	int	Private	Model's number of raters

### 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
imageType	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. ModelRating

##### Attribute

Attribute	Type	Visibility	Description
id	int	Private	Unique identifier of image type
model	Model	Private	Model which is evaluated
account	Account	Private	User who does evaluation
rating	int	Private	Number of stars rating
date	String	Private	Rating date

##### Method

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

#### 4.2.8. 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
date	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
description	String	Private	Article's description
thumbImage	String	Priate	Article's thumb image url

### Method

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

#### 4.2.9. 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
content	String	Private	Proposal content
status	String	Private	Proposal status

### Method

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

#### 4.2.10. 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
createdDate	String	Private	Account created date

### Method

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

#### 4.2.11. Role

##### Attribute

<b>Attribute</b>	<b>Type</b>	<b>Visibility</b>	<b>Description</b>
id	int	Private	Unique identifier of role
name	String	Private	Role name

##### Method

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

#### 4.2.12. Profile

##### Attribute

<b>Attribute</b>	<b>Type</b>	<b>Visibility</b>	<b>Description</b>
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
numberOfRaters	int	Private	User's number of raters
numberOfStars	int	Private	User's number of star rating

##### Method

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

#### 4.2.13. Event

##### Attribute

<b>Attribute</b>	<b>Type</b>	<b>Visibility</b>	<b>Description</b>
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
description	String	Private	Event description
thumbImage	String	Private	Event's thumb image url
regDateStart	String	Private	Event's start registration date
regDateEnd	String	Private	Event's end registration date

##### Method

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

#### 4.2.14. 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
ratingDate	String	Private	User's rating date

##### Method

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

#### 4.2.15. 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	NoticationType	Private	Notification type object
account	Account	Private	User whom notification is sent to
date	String	Private	Created date

##### Method

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

#### 4.2.16. 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

#### 4.2.17. 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	int	Private	Trade type
condition	int	Private	Trade condition
price	double	Private	Products's price
priceNegotiable	int	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	int	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
numberOfRater	int	Private	Number of users who evaluate this trade
numberOfStars	int	Private	Number of stars rating
latlng	String	Private	Trade post's location latitude and longitude

### Method

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

#### 4.2.18. 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
status	String	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

#### 4.2.19. 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

## 4.2.20. 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
tel	String	Private	Reporter's mobile phone
email	String	Private	Reporter's email

### Method

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

## 4.2.21. 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
order	OrderRequest	Private	OrderRequest that is evaluated

rating	int	Private	Number of stars rating
comment	String	Private	Feedback about exchange

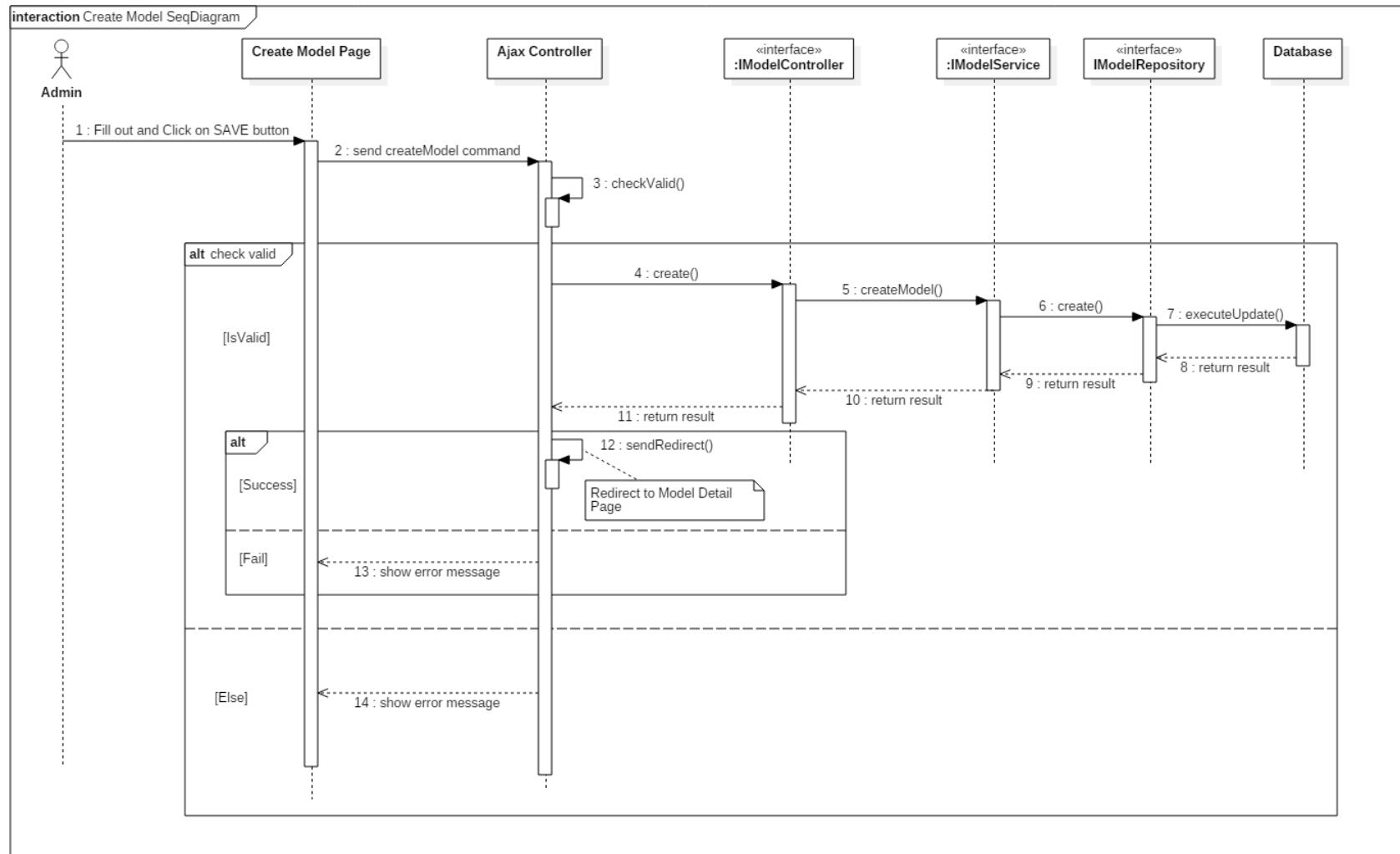
### Method

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. Create model



**Figure 50 - < Sequence Diagram > Create model**

#### 4.3.1.2. Edit model

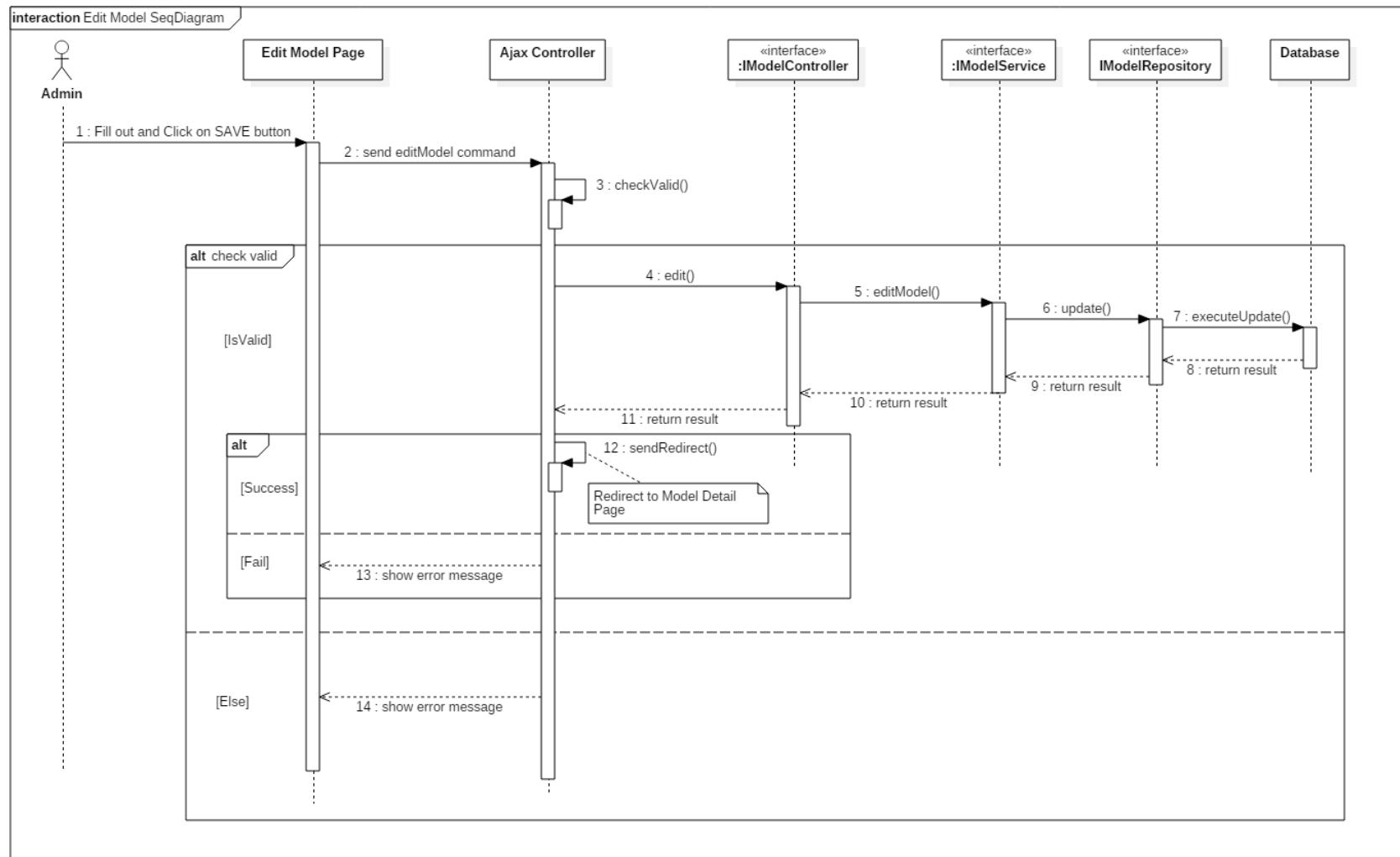


Figure 51 - < Sequence Diagram > Edit model

#### 4.3.1.3. Search Trade Post

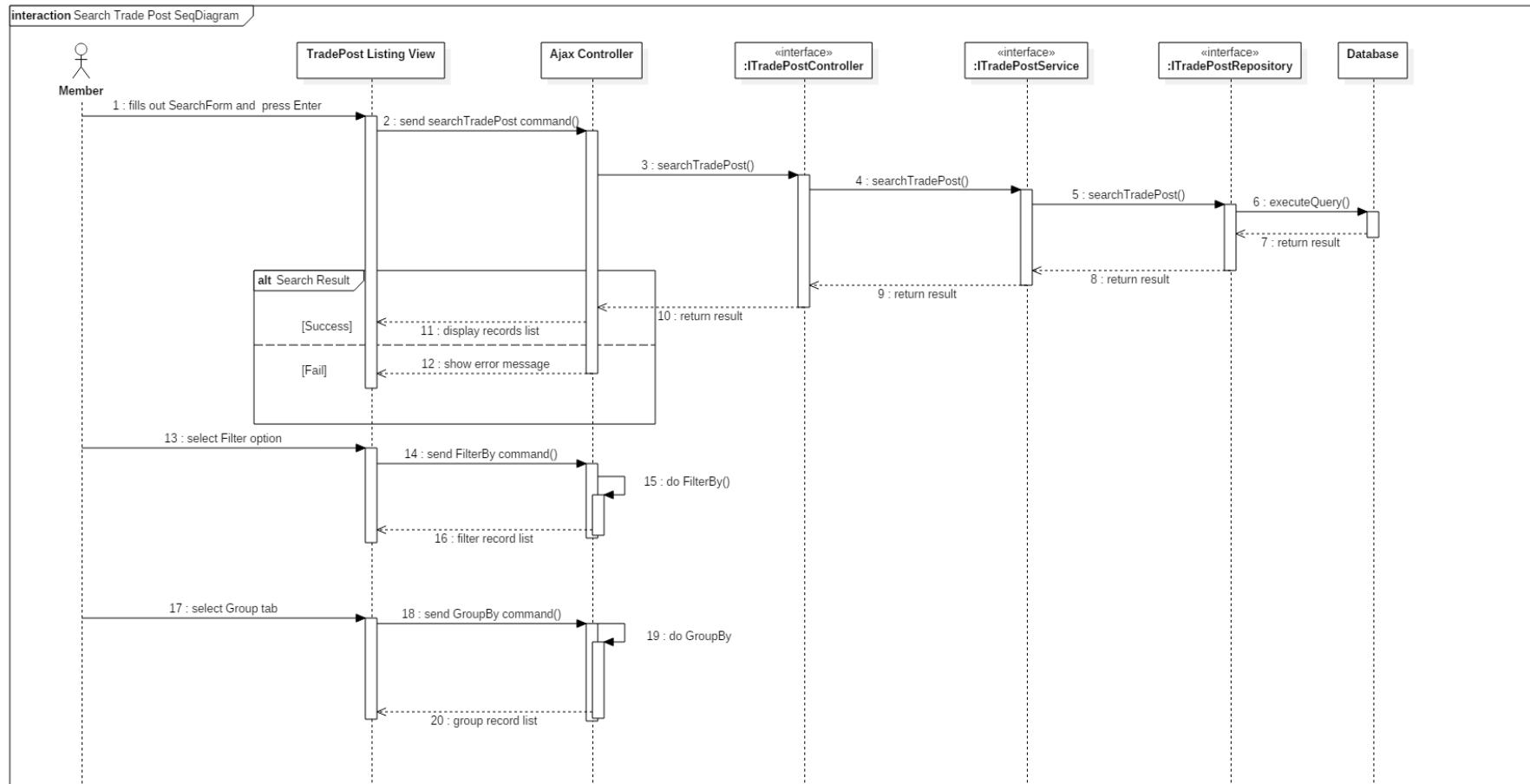


Figure 52 - < Sequence Diagram > Search Trade Post

#### 4.3.1.4. Post new trade

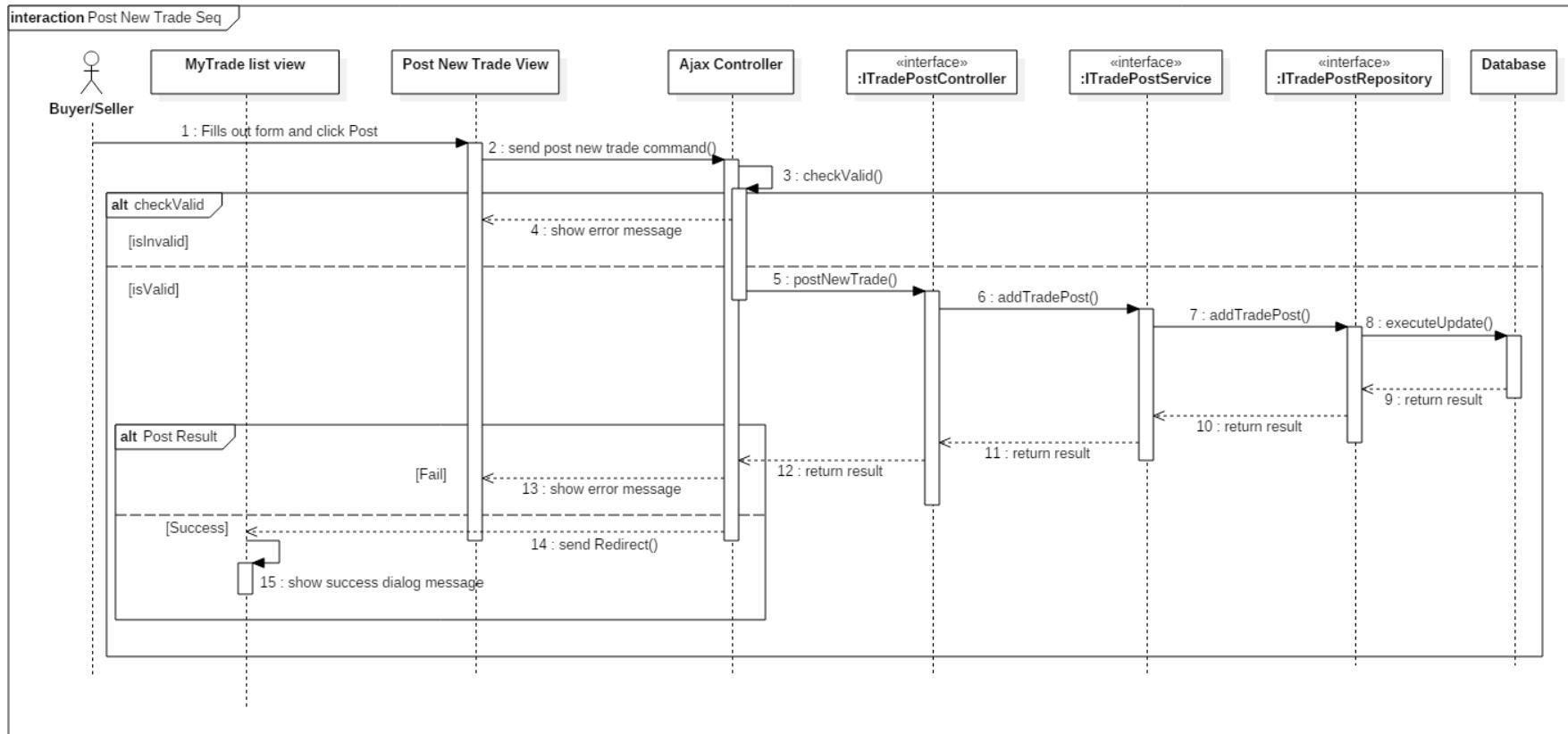


Figure 53 - < Sequence Diagram > Post new trade

#### 4.3.1.5. Edit trade post

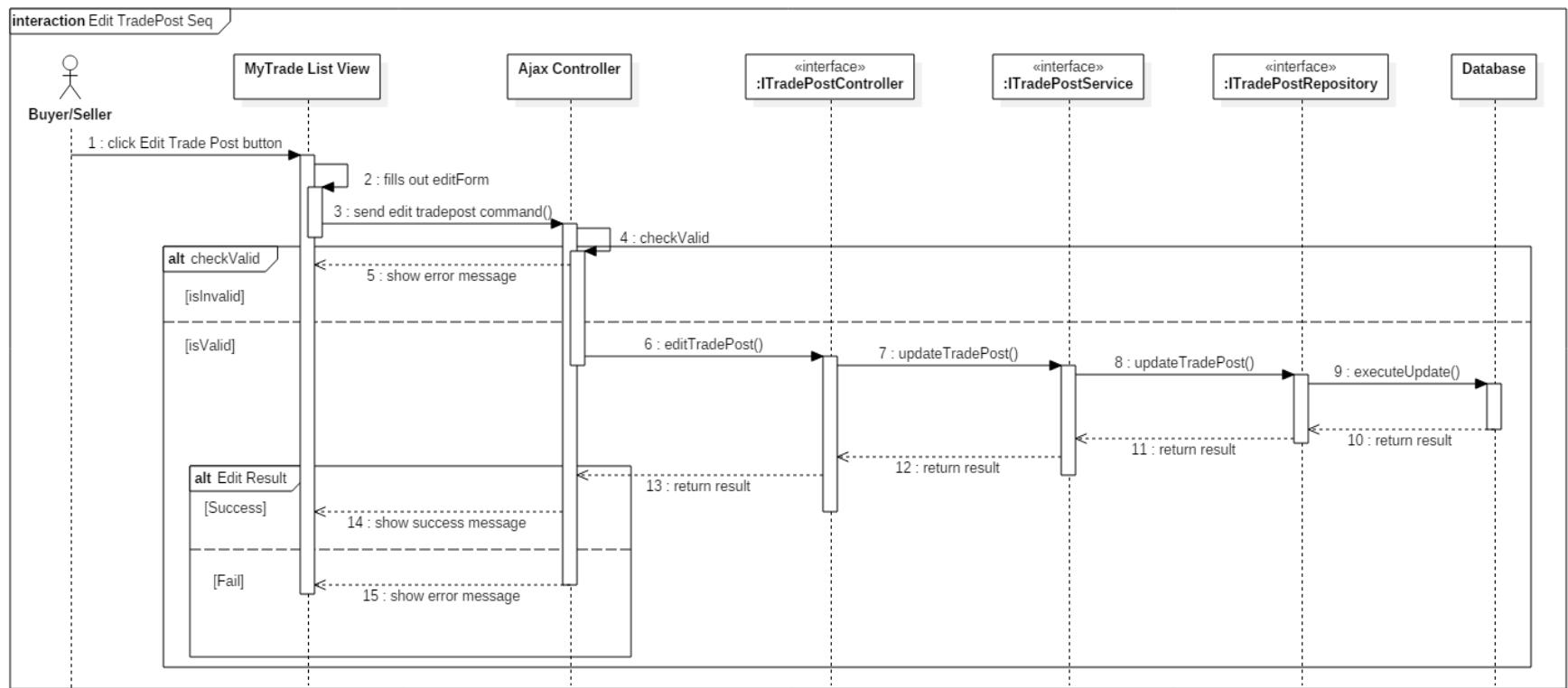


Figure 54 - < Sequence Diagram > Edit trade post

#### 4.3.1.6. Request order to a trade post

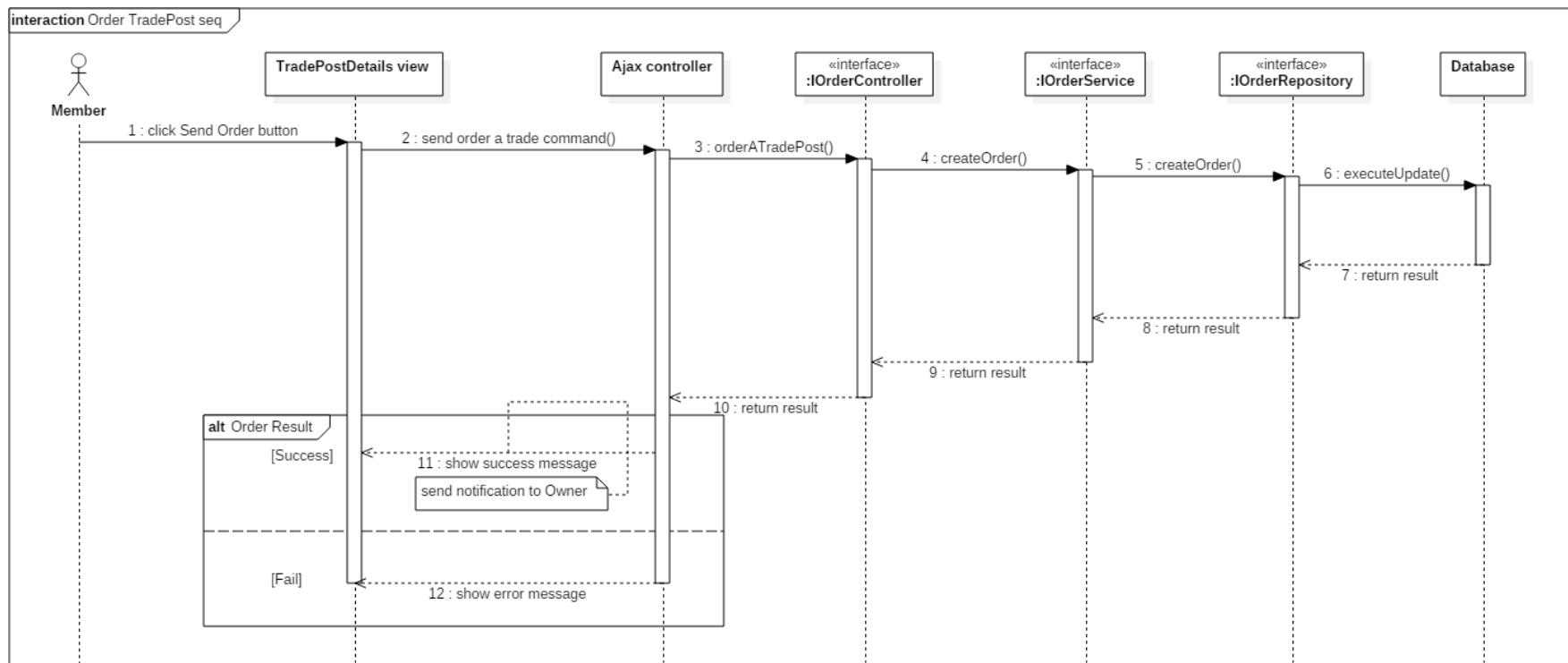


Figure 55 - < Sequence Diagram > Request order

#### 4.3.1.7. Approve order request

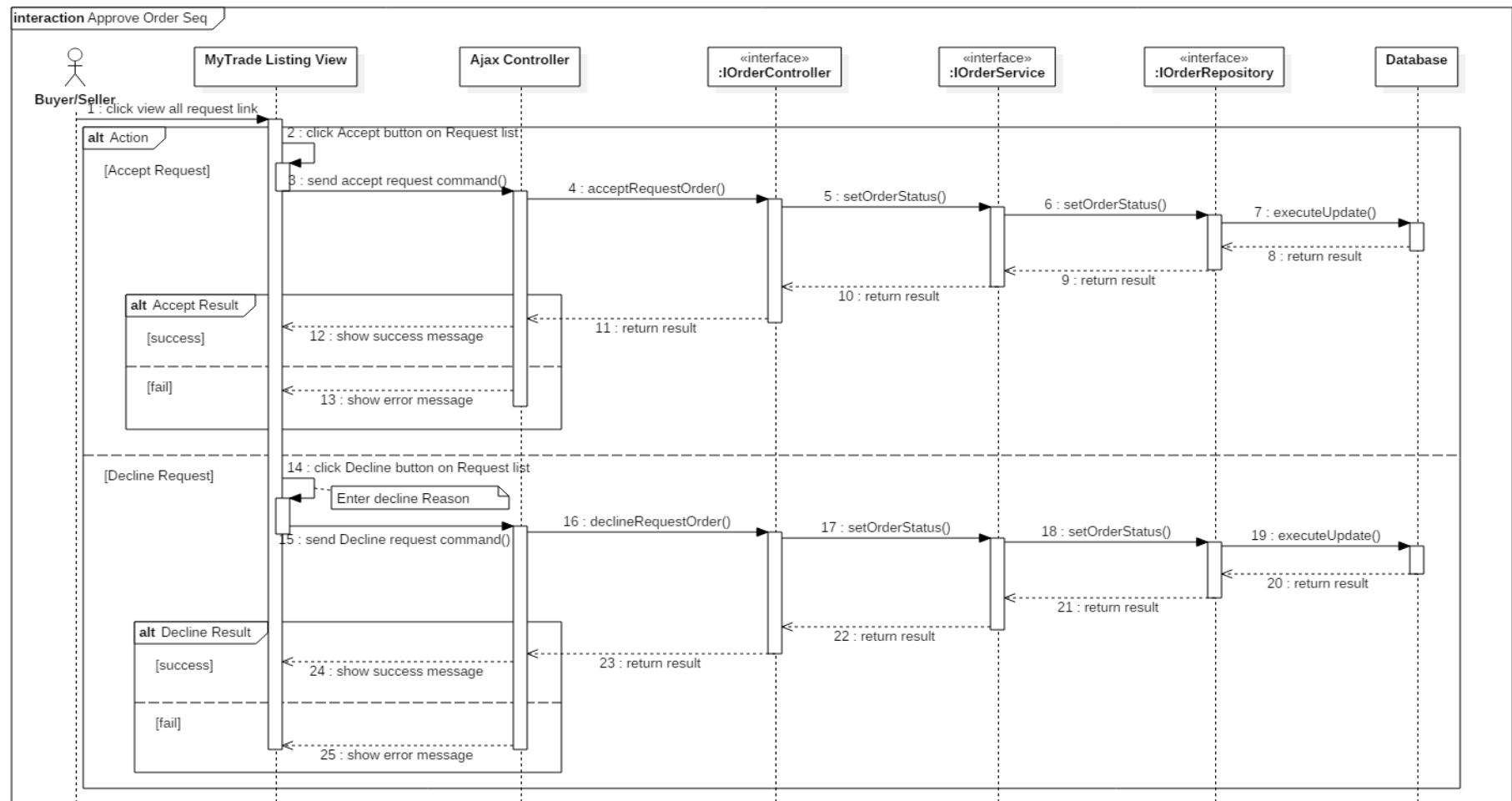


Figure 56 - < Sequence Diagram > Approve order request

#### 4.3.1.8. Confirm trade success

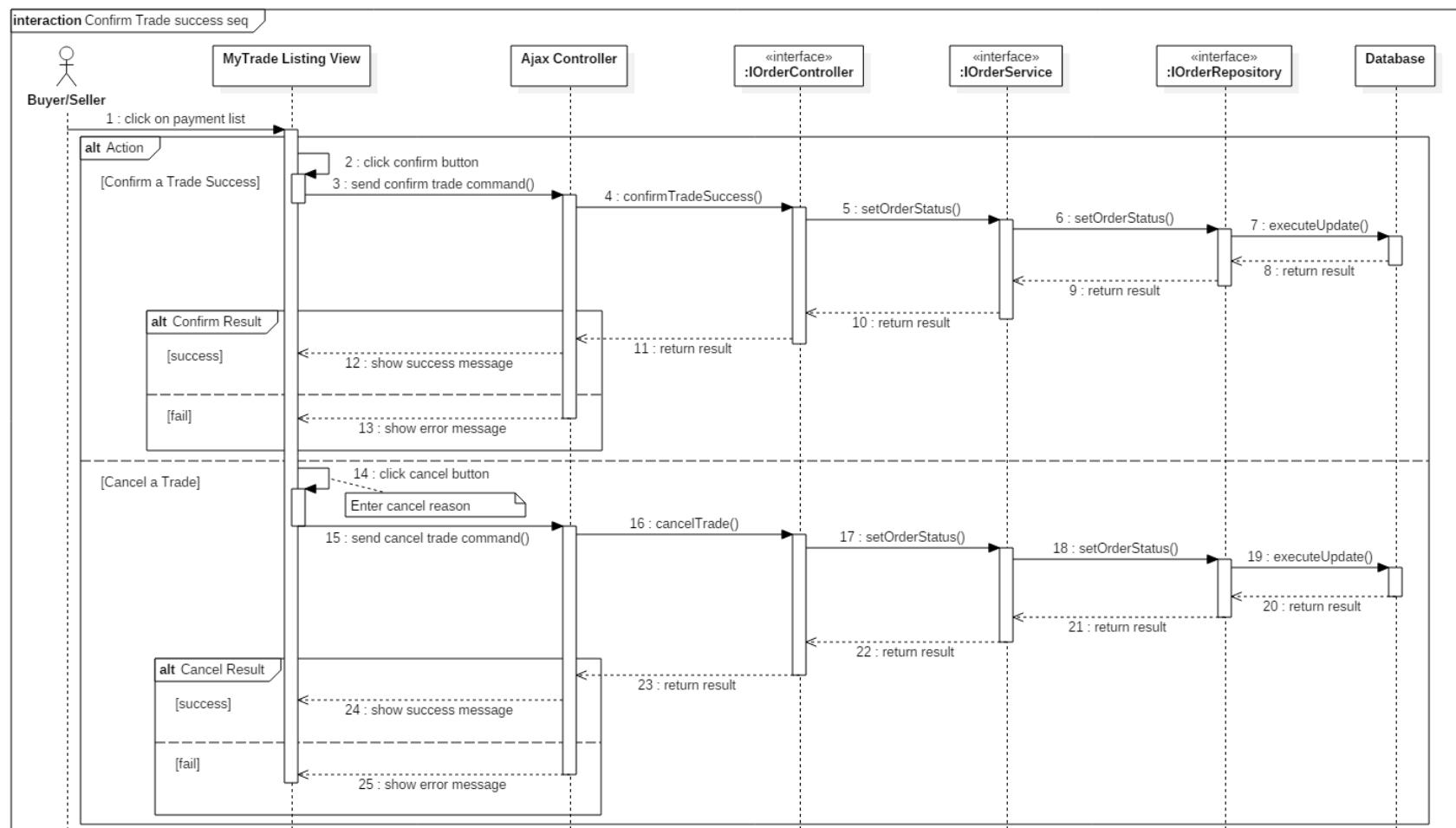


Figure 57 - < Sequence Diagram > Confirm trade success

#### 4.3.1.9. Create new article

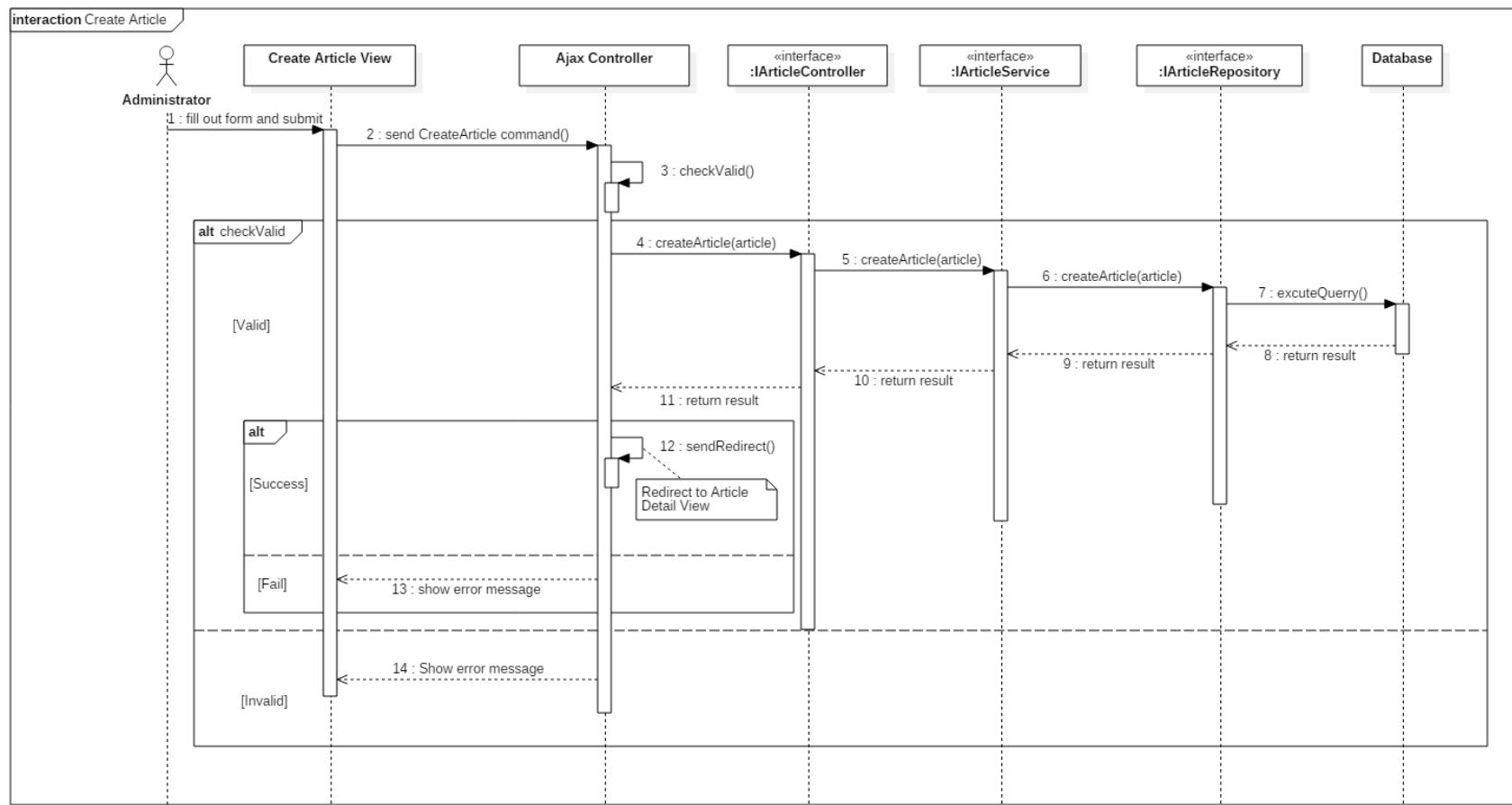


Figure 58 - < Sequence Diagram > Create new article

#### 4.3.1.10. Edit article

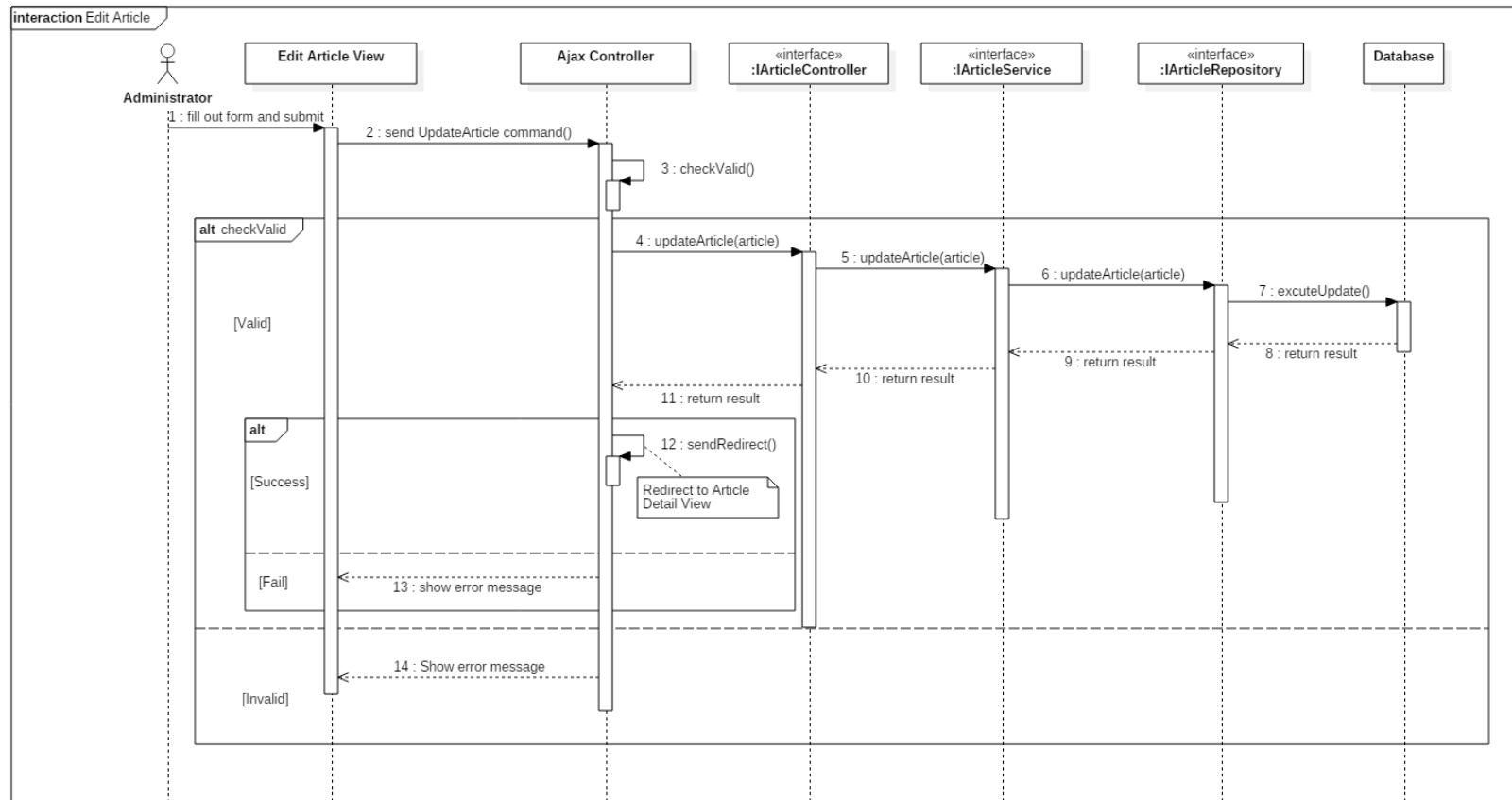


Figure 59 - < Sequence Diagram > Edit article

#### 4.3.1.11. Create event

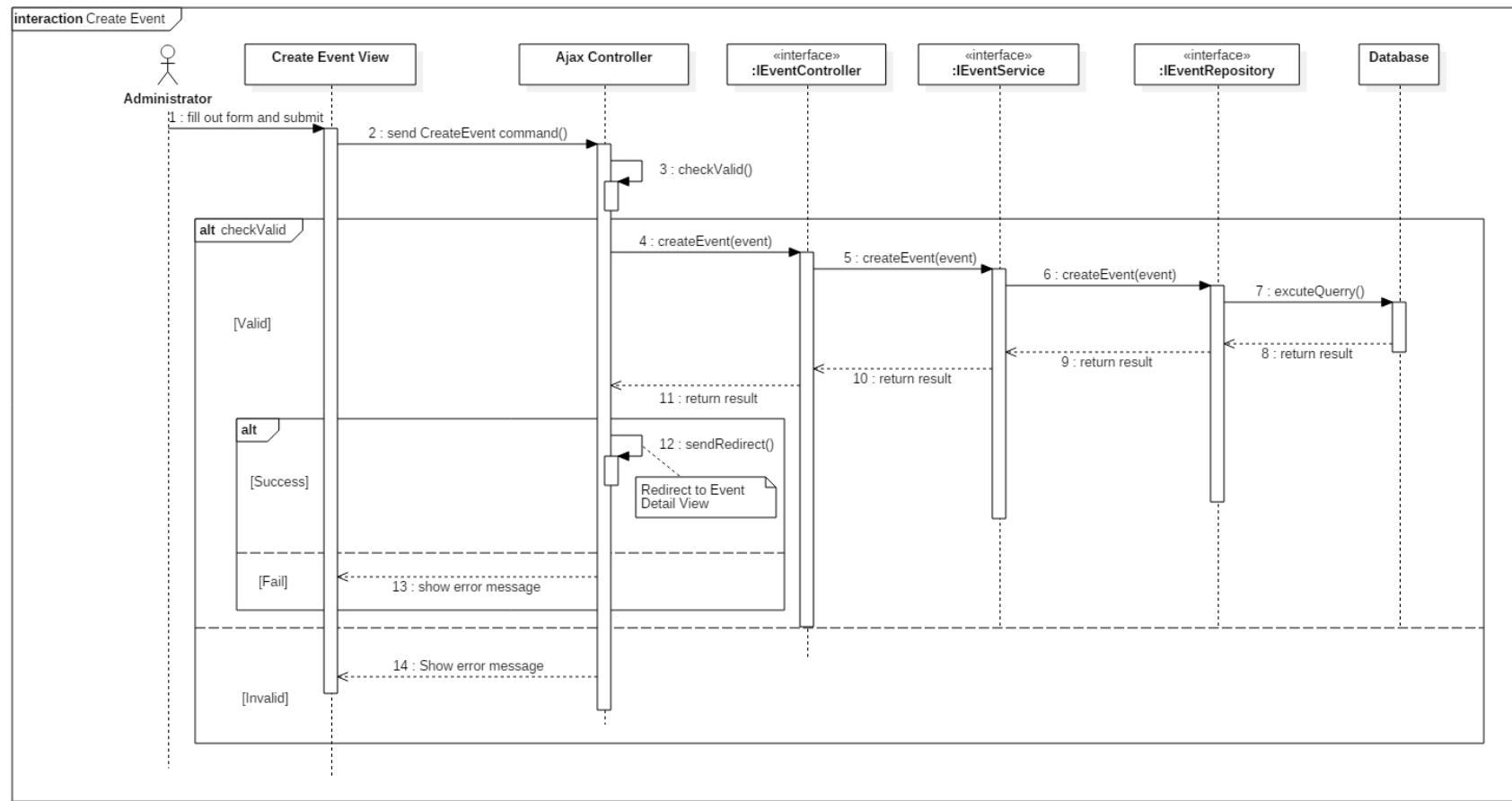


Figure 60 - < Sequence Diagram > Create event

#### 4.3.1.12. Edit event

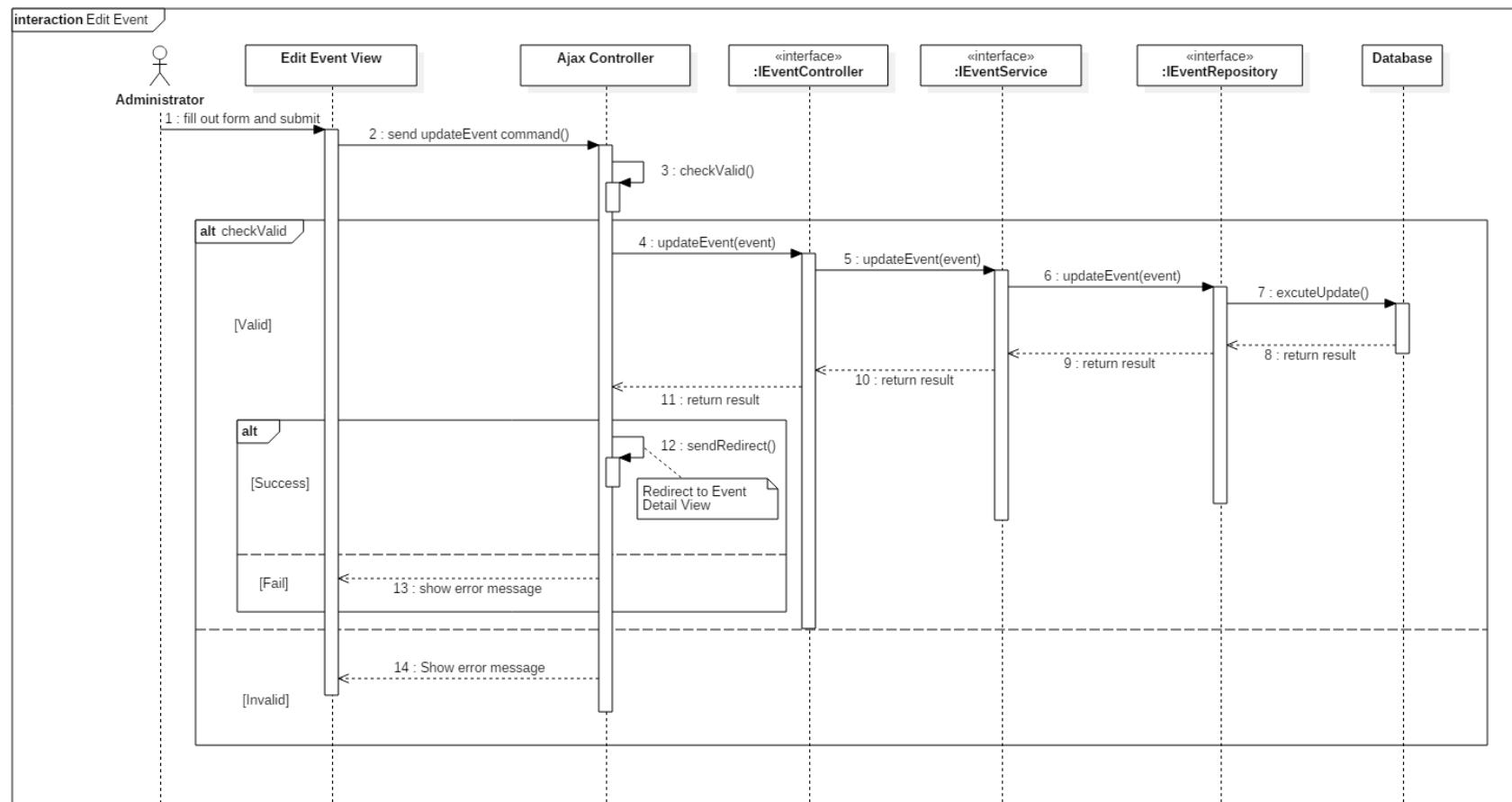


Figure 61 - < Sequence Diagram > Edit event

#### 4.3.1.13. Register event

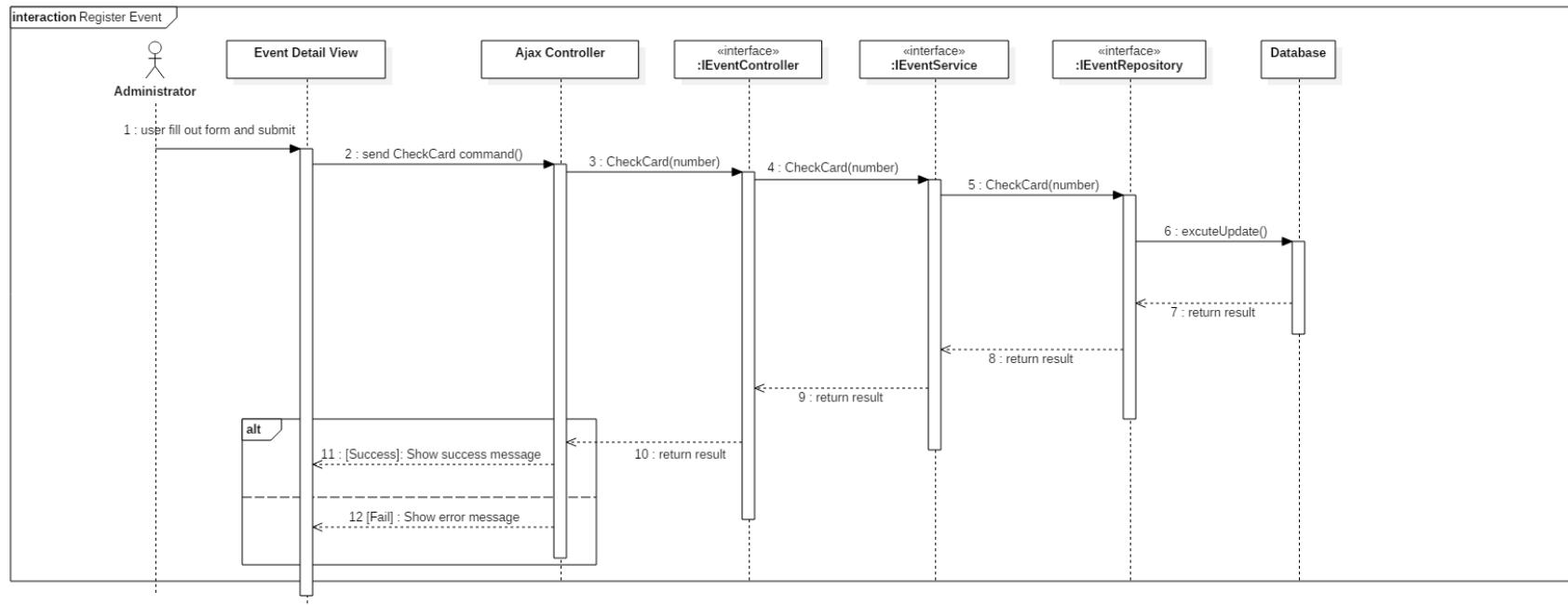


Figure 62 - < Sequence Diagram > Register event

#### 4.3.2. Activity Diagram

##### 4.3.2.1. Search Gundam

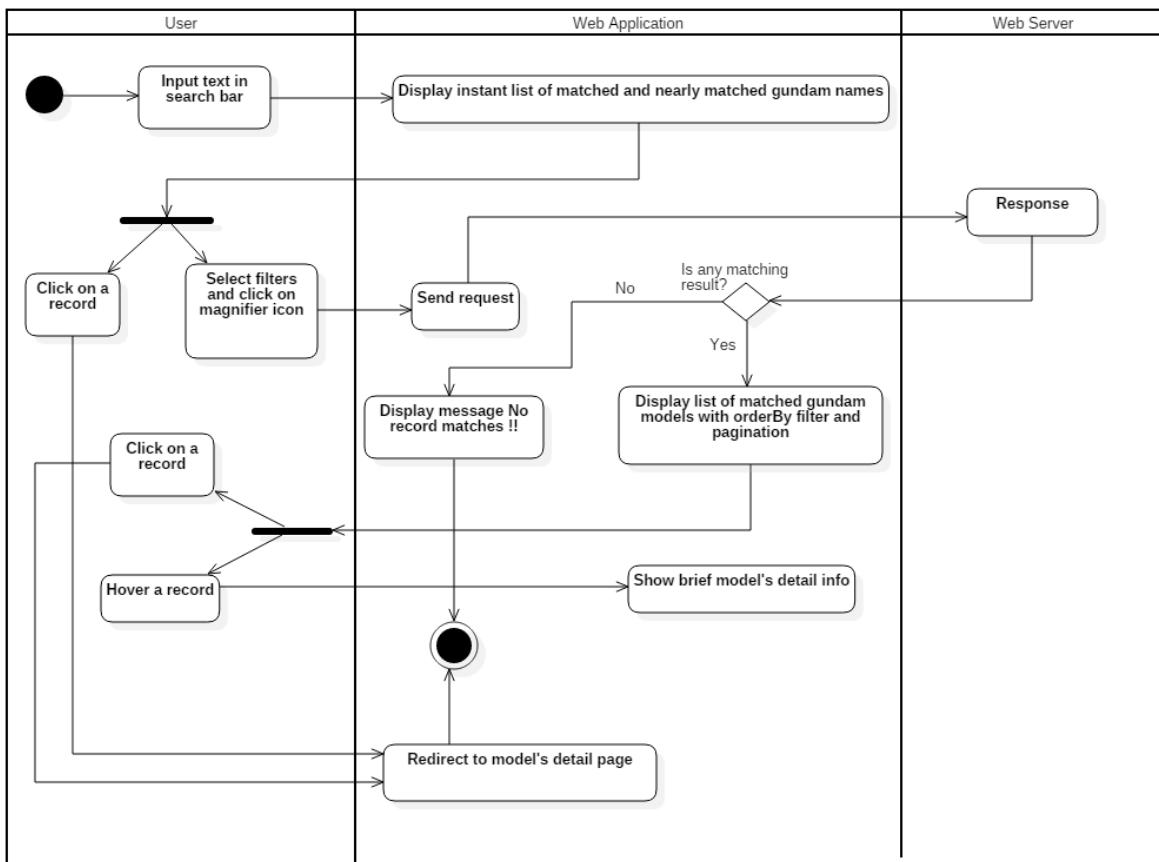


Figure 63 - < Activity Diagram > Search Gundam

#### 4.3.2.2. Evaluate Gundam

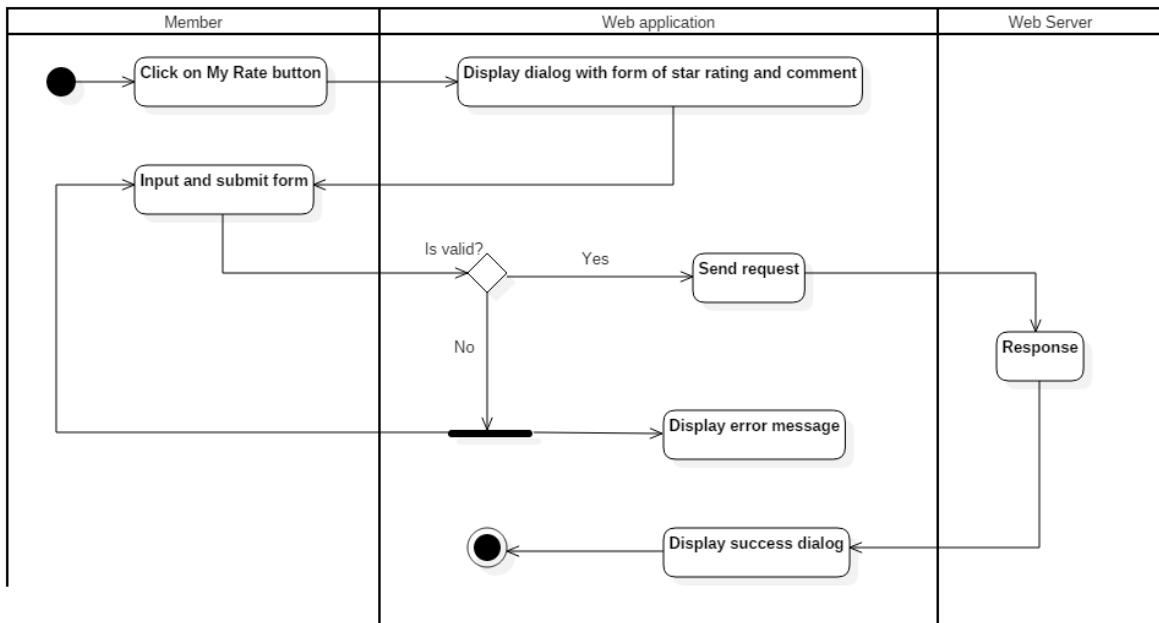


Figure 64 - < Activity Diagram > Evaluate Gundam

#### 4.3.2.3. Update profile

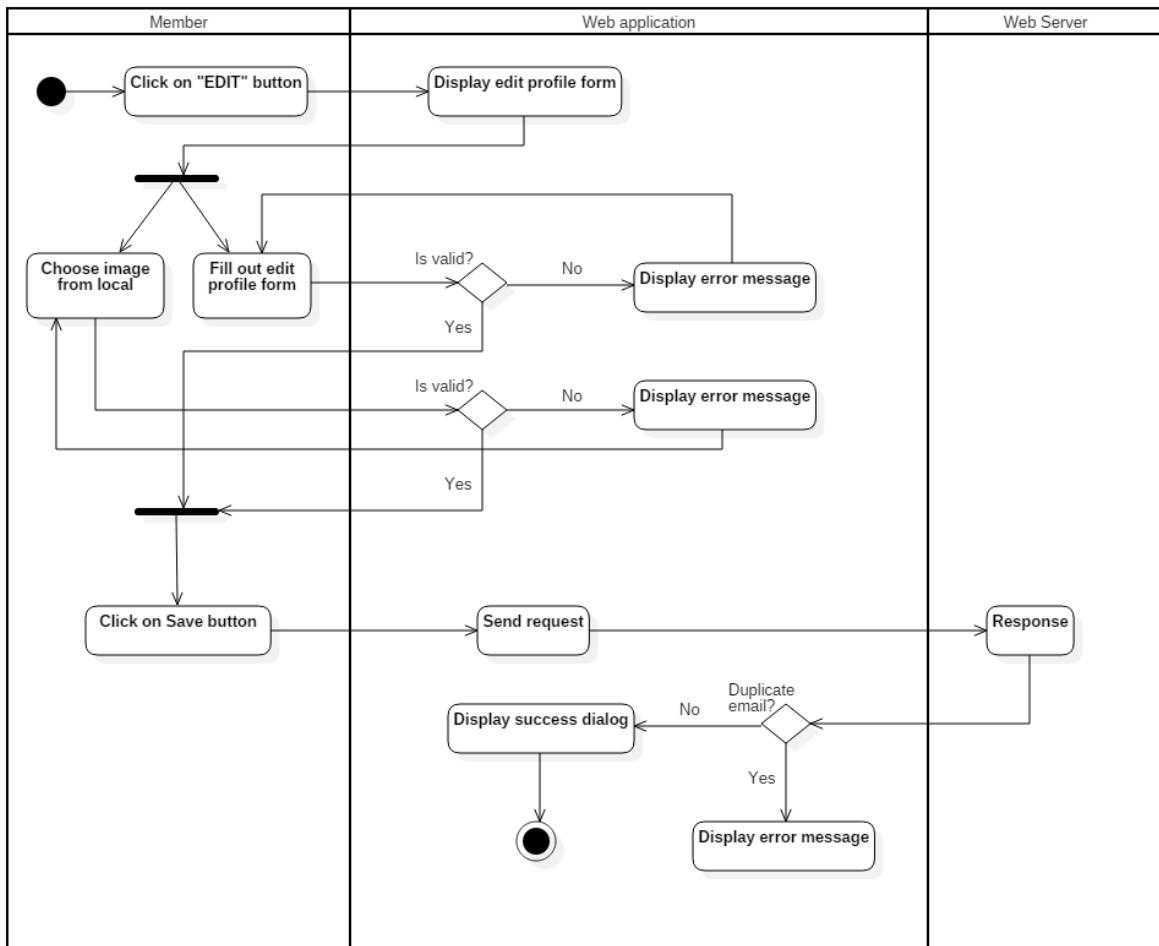


Figure 65 - < Activity Diagram > Update profile

#### 4.3.2.4. Manage order

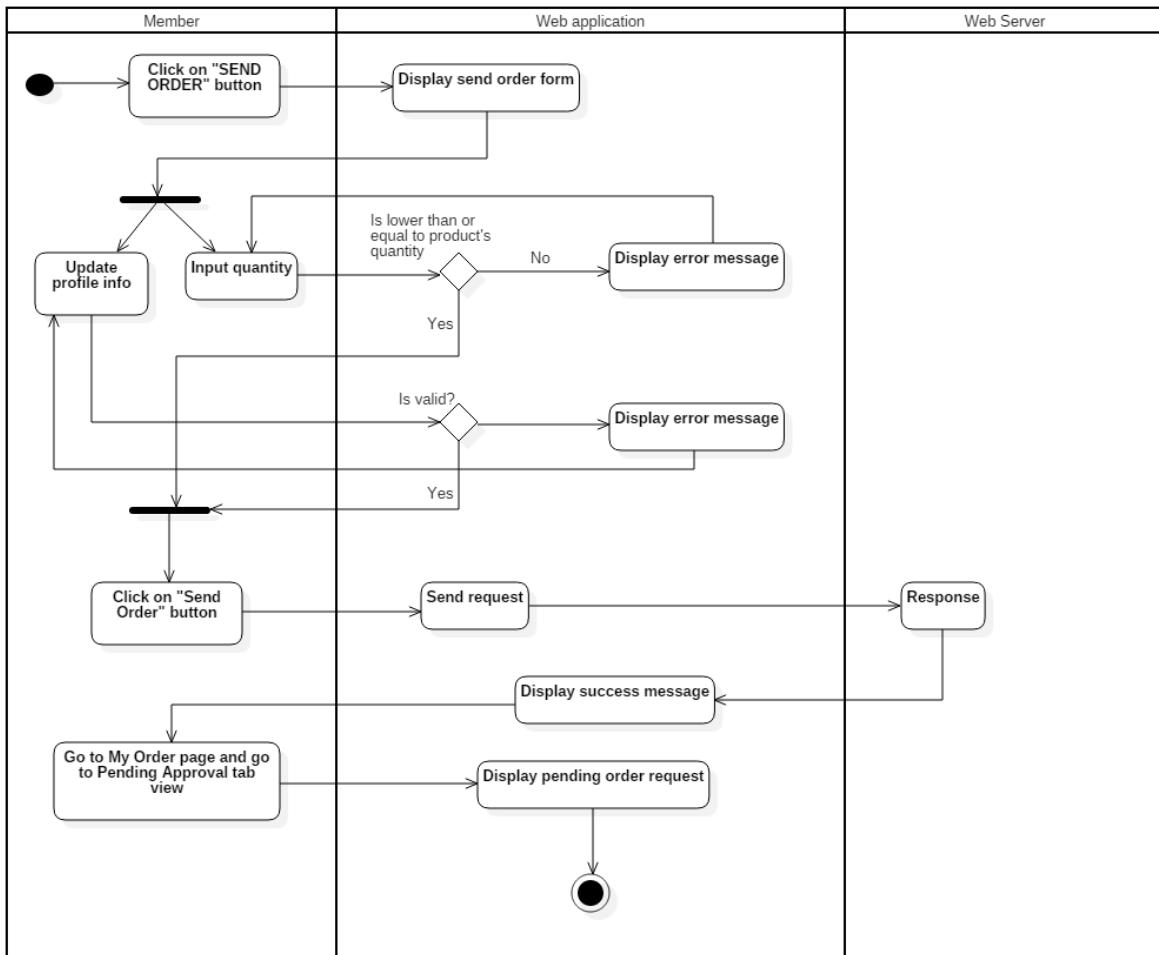


Figure 66 - < Activity Diagram > Manage order

# 5. Interface

## 5.1. Member

### 5.1.1. Web application

#### 5.1.1.1. Display search model view

The screenshot shows a search interface for a Gundam model. At the top left is a large image of a red and blue Unicorn Gundam. Below it is a sidebar with two sections: "CATEGORY" and "OPTIONS". The "CATEGORY" section has a red header "Series Title" and lists "High Grade", "Love Phantom", "Master Grade", "Perfect Grade", "Real Grade", "Remaster Ver.", and "Thunderbolt". The "OPTIONS" section has a red header "Product Series" and shows "1/100" and "1/144". The main content area has a header "Search Model Information" with dropdowns for "Product Series" (All, 3), "Price" (All, 4), "Series Title" (All, 5), and "Manufacturer" (All, 6). There are also "Order by" dropdowns for "Rating" (Descending, 9) and "13" (page number). A search bar (1) with placeholder "Leave blank to search all" and a magnifying glass icon (2) are at the top right. Below the search bar is a scroll arrow (7). The results section (11) shows four items: "Unicorn Gundam (Red / Gree...", "Unicorn Gundam 03 Phe...", "RX-0 Unicorn Gundam 02...", and "RX-78 GP01Fb Gundam G...". Each item has a thumbnail, a title, a price (e.g., 10,800 yen), a release date (e.g., Late Dec, 2016), a manufacturer (e.g., Bandai), a product series (e.g., 1/100), and a "More Information" button. A scroll arrow (12) is located below the first item. A scroll arrow (13) is located below the page number. A scroll arrow (14) is located at the bottom right of the results section. A scroll arrow (15) is located at the bottom right of the sidebar.

This screenshot shows another search interface for a different set of Gundam models. The layout is identical to the previous one, with a sidebar, search bar, and results section. The sidebar has the same "CATEGORY" and "OPTIONS" sections. The main content area has a header "Search Model Information" with dropdowns for "Product Series" (All) and "Price" (All). It also has "Order by" dropdowns for "Rating" and "15" (page number). The search bar and scroll arrows are also present. The results section (11) shows four items: "Unicorn", "RX-0 Full Armor Unicorn Gundam Ver.Ka", "RX-0 Unicorn Gundam", and "Unicorn Gundam HD Color +MS CAGE". A scroll arrow (12) is located below the first item. A scroll arrow (13) is located below the page number. A scroll arrow (14) is located at the bottom right of the results section. A scroll arrow (15) is located at the bottom right of the sidebar.

## 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	Product Series	Choose product series filter	No	No	Dropdown list	String	N/A
4	Price	Choose price filter	No	No	Dropdown list	String	N/A
5	Series Title	Choose series title filter	No	No	Dropdown list	Float	N/A
6	Manufacturer	Choose manufacturer filter	No	No	Dropdown list	String	N/A
8	Order By options	Choose order by options	No	No	Dropdown list	String	N/A
9	Ascending or Descending options	This dropdown list follows with order by dropdown list	No	No	Dropdwon list	String	N/A
11	Model detail info	Display model detail info when hover on a record	Yes	Yes	N/A	String	N/A
13	Search record	Search record	Yes	Yes	N/A	N/A	N/A
14	Model name	Display model's name	Yes	Yes	N/A	String	N/A
15	As-you-type list	This list appears when user types any text in search input bar	Yes	No	N/A	N/A	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
7	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
10	Pagination	For paging with first, next, page numbers, next and last	N/A	The list records is updated match with page numbers
12	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



CATEGORY  
OPTIONS

- Series Title
- High Grade
- Love Phantom
- Master Grade
- Perfect Grade
- Real Grade
- Remaster Ver.
- Thunderbolt

**MODEL INFORMATION**



**Name:** RX-0 Unicorn Gundam 02 Banshee  
**Titanium Finish Ver.**  
**Manufacturer:** Bandai  
**Scale:** 1/100  
**Series:** Master Grade  
**Release Date:** Late Dec., 2012  
**Sales Price:** 10,800 yen  
**Item Code:** 2203063

Rating: ★★★★★ (16)  5

**IMAGES**

Package	Item picture
	

**TOP FIVE**

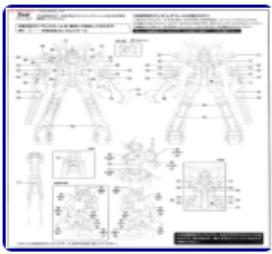
Model

-  4
- 
- 
- 

**About item**



**Color**



**Assembly guide**



3

**DESCRIPTION**

---

Copyright Sotsu Agency / Sunrise

6

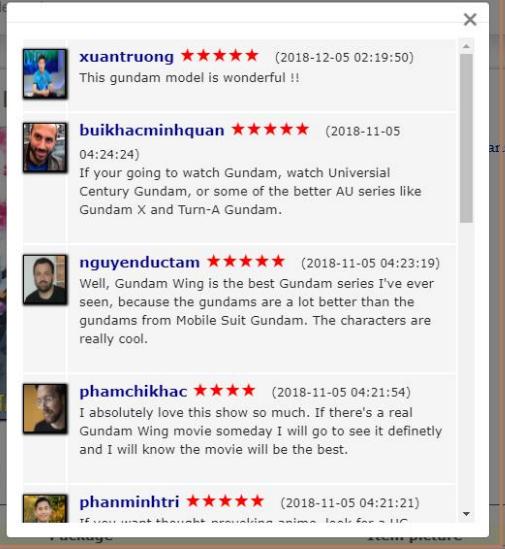
*Note: Can only fit one copy per SAL Small Packet/Registered SAL shipment, and size limitations may be exceeded if combined with other items.*

**Related Trade Post**

- [Bán thân ra đĩ con RX-05 Unicorn Gundam](#)
- [Bán số lượng lớn cảng con RX-0](#)
- [Muốn mua cái khiên mẫu RX-0 Unicorn Gundam](#)
- [Cần mua gấp thủ cấp con RX-0 Unicorn Gundam](#)
- [Cần bán gấp con RX-0 Unicorn Gundam](#)

**Related Article**

- [RX-0 Unicorn Gundam 02 Banshee Titanium Finish Ver. Manual and Color Guide](#)
- [Review model RX-0 Unicorn Gundam 02 Banshee Titanium Finish Ver.](#)
- [DV 0 Unicorn](#)



**XUANTRUONG** ★★★★★ (2018-12-05 02:19:50)  
This gundam model is wonderful !!

**BUIKHACMINHQUAN** ★★★★★ (2018-11-05 04:24:24)  
If your going to watch Gundam, watch Universal Century Gundam, or some of the better AU series like Gundam X and Turn-A Gundam.

**NGUYENDUCTHAM** ★★★★★ (2018-11-05 04:23:19)  
Well, Gundam Wing is the best Gundam series I've ever seen, because the gundams are a lot better than the gundams from Mobile Suit Gundam. The characters are really cool.

**PHAMCHIKHAC** ★★★★★ (2018-11-05 04:21:54)  
I absolutely love this show so much. If there's a real Gundam Wing movie someday I will go to see it definitely and I will know the movie will be the best.

**PHANMINHTRI** ★★★★★ (2018-11-05 04:21:21)  
If you want thought provoking anime, look for a HG

**TOP FIVE**

Model

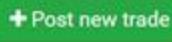
## Fields

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
2	Model rating	Average rating of model	Yes	Yes	N/A	N/A	N/A
3	Model images	Model images	Yes	Yes	N/A	Img	N/A
4	TOP FIVE	Top 5 rating models	Yes	Yes	N/A	N/A	N/A
6	Model description	Model's description	Yes	Yes	N/A	String	N/A
7	Related trade post	Related exchange	Yes	Yes	N/A	N/A	N/A
8	Related article	Related article	Yes	Yes	N/A	N/A	N/A
9	Review evaluation	User's review evaluation	Yes	Yes	N/A	N/A	N/A

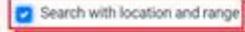
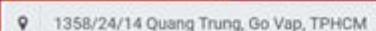
### Buttons/Hyperlinks

No	Function	Description	Validation	Outcome
5	Display review evaluation dialog	Display other user's review evaluation on current model	N/A	A dialog with user's review evaluation

### 5.1.1.3. Display TradePost Listing view

1 

2  3 

4  5  6  7 

8  9  10  11  12 

**10** **11**

13    

**10** **11**

**12** **13**

1 

## 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

[Home](#) > [Trade market](#) > [Gundam abc - Title here](#)



**\$950.00**  
Gundam abc - Title here

Trade ID: 251716763  
🕒 7 Jan, 16 10:10 pm 🗺 Los Angeles, USA  
 Yury Corporation need to sell

**Product Information**

Condition: Used  
Brand: Brand here  
Model: Model here

Quantity: 999  
Rating: ★★★★☆

**SEND ORDER**

Share this ad

**Description**

Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est.

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

**Trade Information**

- Delivery: Meet in person
- More trades by [Yury Corporation](#)
- Contact: Send order to see
- Report this trade

**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

The screenshot displays the 'My Order' view with the following elements:

- Header:** Trade Listing, My Trade, My Order (highlighted in red).
- Header Buttons:** On Payment, Pending Approval, Rejected, History.
- Order Item 1:**
  - Image:** Red robot model.
  - Order From:** Title of Trade Post, Phone: 0343777228, Email: tksangit@gmail.com, Quantity: 1 - Need to pay: 50\$.
  - Status:** Ordered On: 7 Jan, 16 10:10 pm, Accepted On: 7 Jan, 16 10:10 pm.
  - Actions:** Location icon (4), Refresh icon (5).
- Order Item 2:** Similar to Item 1, showing a red robot model, order details, and status.
- Order Item 3:** Similar to Item 1, showing a red robot model, order details, and status.
- Order Item 4:** Similar to Item 1, showing a red robot model, order details, and status.
- Order Item 5:** Similar to Item 1, showing a red robot model, order details, and status.
- Navigation:** Page numbers 1 through 10 at the bottom.

**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

The screenshot shows the 'Article Listing' view of the Gunpla World website. The interface includes a navigation bar with Home, GUNDAM, Articles (1), Events, and Trade Market. Below the navigation is a red-bordered 'Article Listing' section containing 'My articles' and 'Create article' buttons. A search bar with 'Search article' input and 'Search' button is also red-bordered. A 'Sort' dropdown menu is shown with 'Newest' selected. The main content area displays two article cards. The first card is for a 'SHOPPING GUIDES' article titled 'Gunpla Pre-orders and New Releases for December-January 2018' from 2018/11/26 by author khanh. The second card is for a 'TUTORIALS' article titled 'How to Build a Mesa Diorama For Gunpla Display' from 2018/11/26 by author khanh. Both cards include a note: 'This article is crawled by system !!!'.

## 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	Yes	Text input	String	N/A
5	Search record	Result record from searching	No	Yes	N/A	N/A	N/A
3	Sort by date	Sort by newest first or oldest first	No	Yes	Dropdown list	String	N/A
4	Category type	Select article's category	Yes	Yes	Dropdown list	String	N/A
6	Article category	The category that this article belongs to	Yes	Yes	N/A	String	N/A
8	Article post date and author	Date when article was posted and the user who posted it	Yes	Yes	N/A	String	N/A
9	Article description	Short description of the article	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
7	Article's title	Title of the article	N/A	Redirects to the article's detail view
10	Article listing	Tab for searching articles	N/A	Redirect to article listing tab
11	My article	Tab for showing article posted by the logged in user	N/A	Redirect to my article tab
12	Create article	Link to the create article page	N/A	Redirects to the create article view

### 5.1.1.7. Article detail view

The screenshot shows a red header bar with the text "Gunpla World". Below it is a navigation menu with links to "Home", "GUNDAM", "Articles", "Events", and "Trade Market". The main content area has a white background. At the top of the content area, there is a title "Gunpla Pre-orders and New Releases for December-January 2018" and a timestamp "Posted on: 2018/11/26 / Author: khanh". To the right of the title is a small "Edit this article" button with a red border and the number "2" above it. Below the title, there is a paragraph of text followed by a heading "New Releases for December 2017". Underneath this heading, there is a link "PG Gundam Exia" and a short description. A photograph of two green and blue bear-like figures holding signs is displayed below the text. The signs read "I'm gonna shoot'em up!" and "I am a Gundam.". The entire screenshot is framed by a red border.

#### Fields

No	Field Name	Description	Read only	Mandatory	Control Type	Data Type	Length
1	Article's information	All the information of an article	Yes	Yes	N/A	N/A	N/A

#### Buttons/Hyperlinks

No	Function	Description	Validation	Outcome
2	Edit this article	Let the user edit the article, only visible if the logged user is the author of this article	Yes	Redirects the user to the edit article view

### 5.1.1.8. Edit Article view

Home GUNDAM Articles Events Trade Market

**Edit article**

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

**Article title:** 1  
Gunpla Pre-orders and New Releases for December-January 2018

Posted on: 2016/11/26 | Author: Manh

**Thumbnail image** 4  


**Category** 2  
Shopping Guides

**Description** 3  
This article is crawled by system !!!

5

Did you get any Gunpla for your winter holiday of choice? If you wanted something you didn't get, we still have our [2017 Gunpla 101 Holiday Guide](#) up for some late month shopping. (But who am I kidding? We keep the holiday guide up all year and just update it each November.)

With no current Gundam shows airing, things are looking a little sparse on the Gunpla front. But the selection makes up in quality what it lacks in quantity. Check it out:

New Releases for December 2017

[PG Gundam Exia](#)

It took until the holidays, but 2017 has finally gifted us with a Perfect Grade Gunpla. In fact, this is our first Perfect Grade in two years—since 2015's [PG Banshee Norma](#).

[PG Gundam Exia \(Lightning Model\) \(pictured at top\)](#)

I especially like this version, which comes with Bandai's latest LED unit for lighting. It's probably better than buying the LED separately: when I bought my PG Unicorn, I spent an extra \$100 on the LED kit. That explains the higher price here. Notice that the LED is built into the Gunpla frame and can be adjusted to glow either blue, green, or pink.

6

**Submit**

Gunpla World Quick Links Follow us  
gunplaworld@gmail.com Home Facebook GUNDAM

### Fields

No	Field Name	Description	Read only	Mandatory	Control Type	Data Type	Length
1	Article's title	The title of the article	No	Yes	Input field	String	<=250
2	Article's category	The category of the article	No	Yes	Drop-down list	String	N/A
3	Article's description	Short description of the article	No	Yes	Textarea	String	<=250

4	Article's thumbnail	Thumbnail of the article	No	Yes	Image	String	N/A
5	Article's content	The content of the article	No	Yes	Textarea	String	N/A

### Buttons/Hyperlinks

No	Function	Description	Validation	Outcome
6	Submit article	Let the user submit the article	Yes	Validate all fields and submit article.

#### 5.1.1.9. Create Article view

The screenshot shows the 'Create article' page of the Gunpla World website. The page has a red header bar with the site name 'Gunpla World'. Below the header, there is a navigation menu with links to Home, GUNDAM, Articles, Events, and Trade Market. The main content area is titled 'Create article' and contains the following fields:

- Title:** A text input field labeled 'Title' with a red border and a red number '1' in the top right corner.
- Category:** A dropdown menu labeled 'Category' with a red border and a red number '2' in the top right corner.
- News:** A dropdown menu labeled 'News' with a red border and a red number '2' in the top right corner.
- Description:** A large text area labeled 'Description' with a red border and a red number '3' in the top right corner.
- Thumbnail image:** A file upload input labeled 'Thumbnail image' with a red border and a red number '4' in the top right corner. It shows a small preview of an image and a file selection button.
- Rich Text Editor:** A WYSIWYG editor with various toolbar icons, labeled with a red border and a red number '5' in the top right corner.

Below the form, there is a note: 'Your article will be reviewed by the administrator before being posted on the website!'. The overall layout is clean and modern, typical of a web-based CMS.

Did you get any Gunpla for your winter holiday of choice? If you wanted something you didn't get, we still have our [2017 Gunpla 101 Holiday Guide](#) up for some late month shopping. (But who am I kidding? We keep the holiday guide up all year and just update it each November.)

With no current Gundam shows airing, things are looking a little sparse on the Gunpla front. But the selection makes up in quality what it lacks in quantity. Check it out:

**New Releases for December 2017**

**PG Gundam Exia**

It took until the holidays, but 2017 has finally gifted us with a Perfect Grade Gunpla. In fact, this is our first Perfect Grade in two years—since 2015's [PG Banshee Norma](#).

**PG Gundam Exia (Lightning Model) (pictured at top)**

I especially like this version, which comes with Bandai's latest LED unit for lighting. It's probably better than buying the LED separately: when I bought my PG Unicorn, I spent an extra \$100 on the LED kit. That explains the higher price here. Notice that the LED is built into the Gunpla frame and can be adjusted to glow either blue, green, or pink.

**Submit** 6

## Gunpla World

Quick Links

Follow us

gunplaworld@gmail.com

Home

Facebook

GUNPLA

## Fields

No	Field Name	Description	Read only	Mandatory	Control Type	Data Type	Length
1	Article's title	The title of the article	No	Yes	Input field	String	<=250
2	Article's category	The category of the article	No	Yes	Drop-down list	String	N/A
3	Article's description	Short description of the article	No	Yes	Textarea	String	<=250
4	Article's thumbnail	Thumbnail of the article	No	Yes	Image	String	N/A
5	Article's content	The content of the article	No	Yes	Textarea	String	N/A

## Buttons/Hyperlinks

No	Function	Description	Validation	Outcome
6	Submit article	Let the user submit the article	Yes	Validate all fields and submit article.

### 5.1.1.10. Event list view

The screenshot shows a web-based event listing interface. At the top, there's a navigation bar with 'Event Listing' (highlighted with a red box and labeled 1), 'Registered events' (labeled 2), and 'Write an event proposal!' (labeled 3). Below this is a search bar with 'Search event' input (labeled 1), a 'Search' button (labeled 2), and a 'Show nearby events' link (labeled 3). A 'Sort' section follows, with 'Newest' (labeled 4) and 'Ongoing/future events' (labeled 5) dropdown menus. The main content area displays event cards. The first card, for 'Event mol', is highlighted with a red box and labeled 6. It shows a photo of a bull, the event title, date range (From: 2020-12-12 to 2030-12-12), ticket price (100.000 VND), and a short description. Below it is another event card for 'GUNPLA BUILDERS WORLD CUP 2018 REGULATION'.

### 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
4	Sort by date	Sort by date newest or oldest first	No	Yes	Drop-down list	String	N/A
5	Status of event	Filter status of the event, active or finished events	No	Yes	Drop-down list	String	N/A
6	Search result	Event records from searching	Yes	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

3	Show nearby events	Search events within 10km of the user	N/A	List records match with location within 10km of the user
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

### 5.1.1.11. Event detail view

The screenshot shows a detailed view of an event on the Gunpla World website. The event is titled "GUNPLA BUILDERS WORLD CUP 2018 REGULATION". The page includes sections for "Information", "Quốc gia và vùng lãnh thổ tổ chức", "Thời gian diễn ra cuộc thi", "Các bảng dự thi", and "Đơn vị tổ chức". The "Information" section contains details like start date (2018-12-12), location (Tô Ký, District 12, Ho Chi Minh City, Vietnam), price (10,000 VND), and attendance (0 people). The "Quốc gia và vùng lãnh thổ tổ chức" section lists countries and regions. The "Thời gian diễn ra cuộc thi" section specifies age ranges for different categories. The "Các bảng dự thi" section provides details about the competition tables. The "Đơn vị tổ chức" section lists the organizing body as BANDAI Hobby products department. At the bottom, there are sections for "Your registered information", "Rate the event", "Write your feedback", and a "SUBMIT FEEDBACK" button.

## 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's information	Information of the event	Yes	Yes	N/A	String	N/A
3	Event's content	The content of the event	Yes	Yes	N/A	String	N/A
4	Registration dates	The time when user can register to this event	Yes	Yes	N/A	String	N/A
5	Registered information	The registered information (visible only after user registered)	Yes	No	N/A	String	N/A
6	Star ratings	Star ratings for the user to rate the event from 1-5	No	No	Star ratings	int	N/A
7	Feedback text	Feedback for the event	No	No	Textarea	String	<=250

10	Register Information	The register information	No	Yes	N/A	N/A	N/A
11	Ticket amount	The number of tickets the user want to buy	No	Yes	Input field	String	N/A

### Buttons/Hyperlinks

No	Function	Description	Validation	Outcome
8	Submit feedback	Button that submit feedback of the user	N/A	Submit the ratings and the feedback text
9	Show register information modal	Register button (visible only if the user hasn't registered and the day is within register dates)	Yes	Make the Register Information modal visible
12	Submit register information	Submit register information	Yes	Validate data and submit the information

#### 5.1.1.12. Create proposal view

Gunpla World

Nguyen Bao Khanh

Home GUNDAM Articles Events Trade Market

Write a proposal for an event.  
Tell us what kind of event would you want to happen! We might make it happen!

**The event's title \*** 1  
What should the event be called?

**Description \*** 2  
A short description of the event

**Location \*** 3  
Where do you want the event to happen?

**Content**  
Please write a detailed description of the event  
4

B I S Tx Source

## Fields

No	Field Name	Description	Read only	Mandatory	Control Type	Data Type	Length
1	Event title	The title of the proposed event	No	Yes	Input field	String	<=250
2	Event description	Description of the proposed event	No	Yes	Input field	String	<=250
3	Location	Proposed location	No	Yes	Input field	String	<=250
4	Event content	Content of the proposed event	No	Yes	Textarea	String	N/A

## Buttons/Hyperlinks

No	Function	Description	Validation	Outcome
5	Submit proposal	Submit proposal button	N/A	Submit the proposal to the server

## 5.1.2. Mobile application

### 5.1.2.1. Get model information

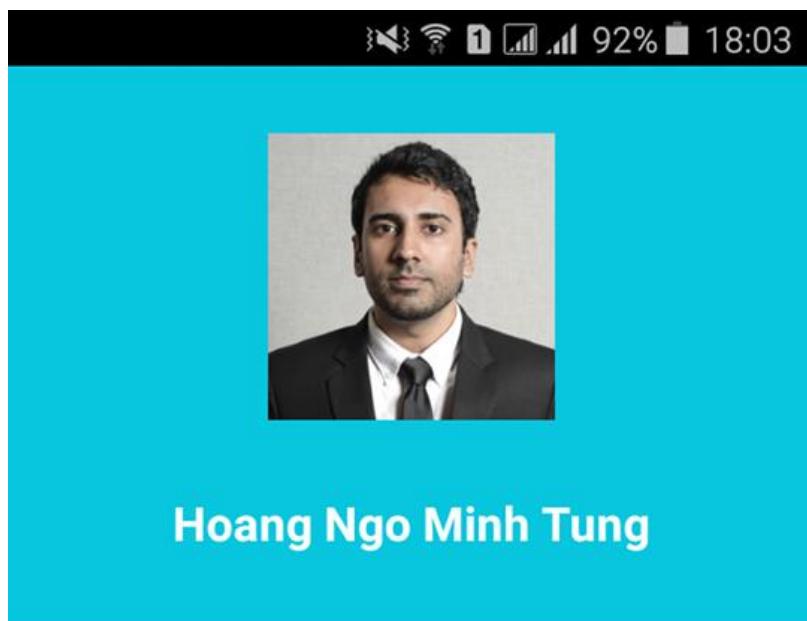
The screenshot shows a mobile application interface titled "Gunpla World". At the top, there is a status bar with icons for signal strength, battery level (92%), and time (18:03). Below the status bar, the app title "Gunpla World" is displayed next to a menu icon (three horizontal lines).

The main content area displays a list of five Gundam models, each with a thumbnail image, the model name, series title, scale, and rating.

- Unicorn Gundam (Red / Green Twin...)**  
Series title: Master Grade  
Scale: 1/100  
Rating: ★★★★★
- Unicorn Gundam 03 Phenex**  
Series title: Master Grade  
Scale: 1/100  
Rating: ★★★★★
- RX-0 Unicorn Gundam 02 Banshee...**  
Series title: Master Grade  
Scale: 1/100  
Rating: ★★★★★
- RX-78 GP01Fb Gundam GP01 Full...**  
Series title: Master Grade  
Scale: 1/100  
Rating: ★★
- RX-78 GP01 GUNDAM GP01...**  
Series title: Master Grade  
Scale: 1/100  
Rating: N/A
- MS-14B/C Gelgoog Cannon**  
Series title: Master Grade

For the MS-14B/C Gelgoog Cannon entry, there are two small thumbnail images labeled "Front view" and "Rear view".

### 5.1.2.2. Get user information



tung



02-12-1997



tunghnmse62231@fpt.edu.vn



1

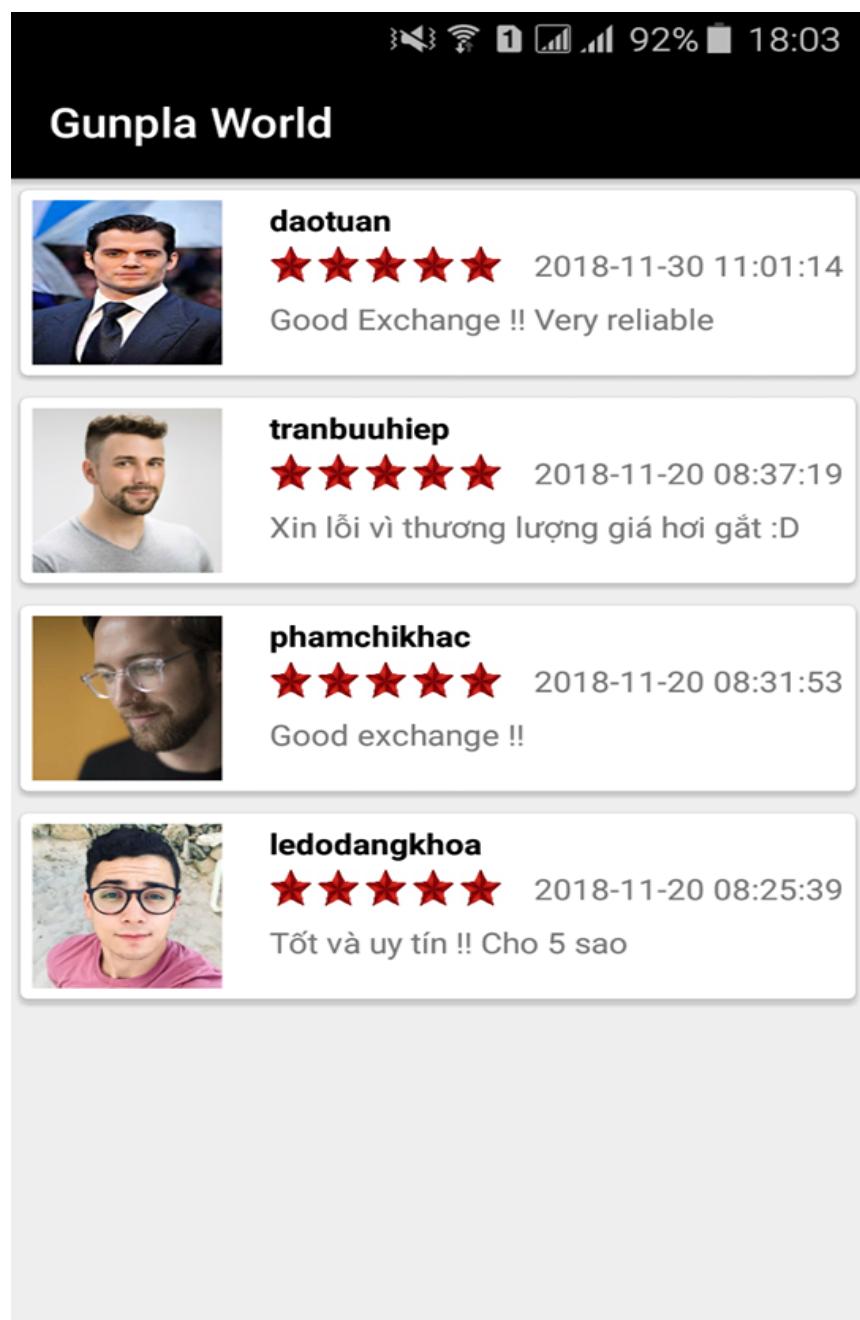


4

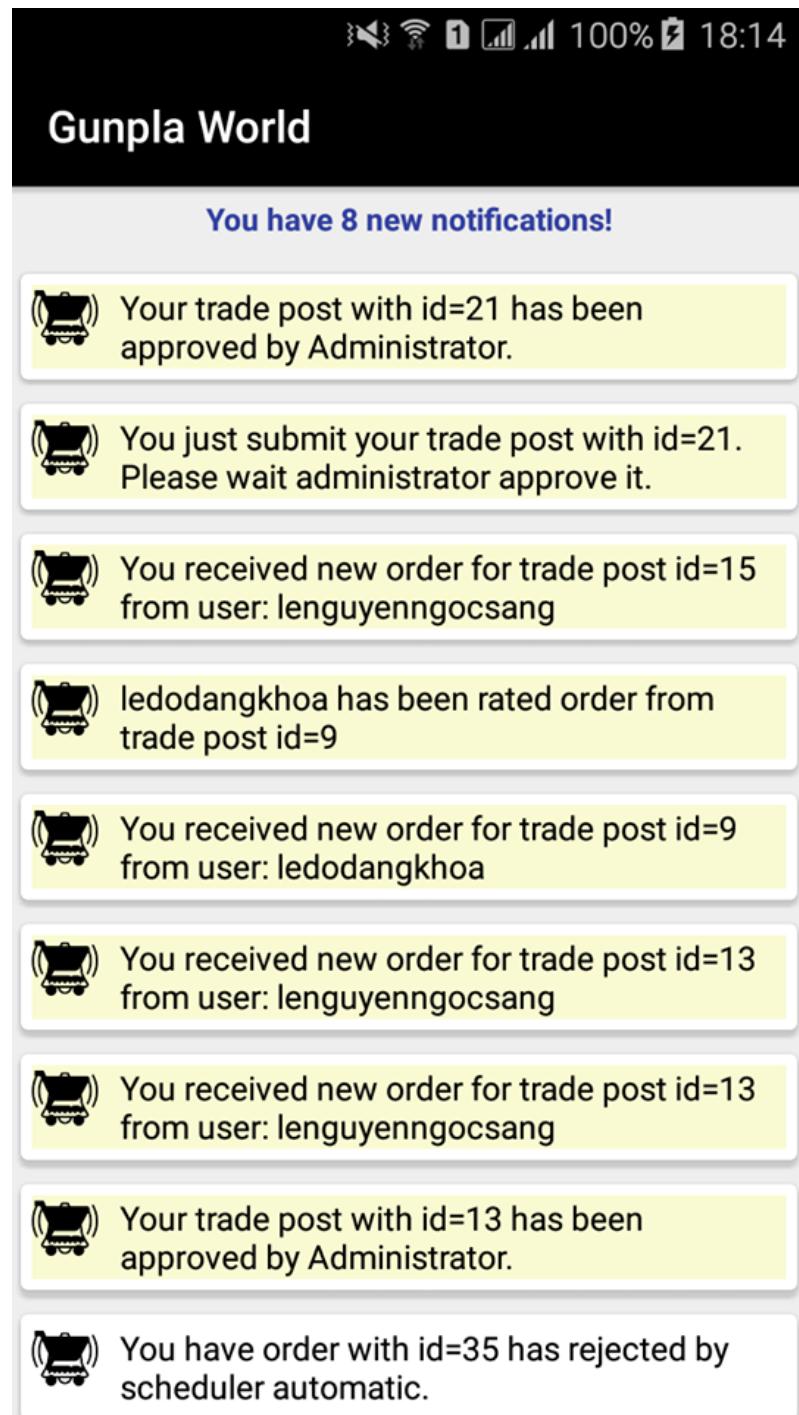


★ ★ ★ ★

### 5.1.2.3. Get user's exchange evaluation



### 5.1.2.4. Get notification



## 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\*  I want to sell  want to buy 1

Title for your trade\* ex, P-bandai xxx brandnew. 2

Photos for your trade (This will be your cover photo ) 3

Select Files to Upload / Drag and Drop Files  
You can add multiple images.

Condition\*  New  Used 4

Price\* SUSD 5 Negotiable 6

Quantity\* 7

Brand Name\* ex, P-bandai 8

Model ex, P150 9

Description\* Write few lines about your products 10

5000 characters left

## Quick rules

Posting an ad on [gundamworld.com](http://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 16 17

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

**Post Your Trade**

## 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

Trade Listing **My Trade** My Order **Post new trade**

Search in my trade...

Enter your keyword and press enter.

Sorting by: Date Descending

**On Trading** Pending Approval Rejected

**1** **2** **3** **4** **5** **6** **7** **8** **9** **10** **11**

**1** Search bar  
**2** Sorting dropdown  
**3** Filter buttons  
**4** Item preview image  
**5** Price and title  
**6** Brand/Model links  
**7** Posting date  
**8** No trading activity message  
**9** Edit and delete icons  
**10** Request Orders and On Payment counts  
**11** Navigation bar

**Item 1:** \$800.00, Gundam abc - Title here!, Brand here / Model here, Posted On: 7 Jan, 16 10:10 pm, Request Orders: 21, On Payment: 12

**Item 2:** \$800.00, Gundam abc - Title here!, Brand here / Model here, Posted On: 7 Jan, 16 10:10 pm, Request Orders: 21, On Payment: 12

**Item 3:** \$800.00, Gundam abc - Title here!, Brand here / Model here, Posted On: 7 Jan, 16 10:10 pm, Request Orders: 21, On Payment: 12

**Item 4:** \$800.00, Gundam abc - Title here!, Brand here / Model here, Posted On: 7 Jan, 16 10:10 pm, Request Orders: 21, On Payment: 12

**Item 5:** \$800.00, Gundam abc - Title here!, Brand here / Model here, Posted On: 7 Jan, 16 10:10 pm, Request Orders: 21, On Payment: 12

1 2 3 4 5 ... 10

### Fields

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

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 mobile application's registration screen. The title 'Register' is at the top. There are six input fields arranged in two columns of three. The first column contains 'First Name' (1) and 'Last Name' (3). The second column contains 'Username' (2) and 'Email' (4). The third column contains 'Password' (5) and 'Confirm Password' (6). Below the fields is a link 'Use your username to login' and a 'Register' button. At the bottom, there is a link 'You have an account already? Log in here'.

#### Fields

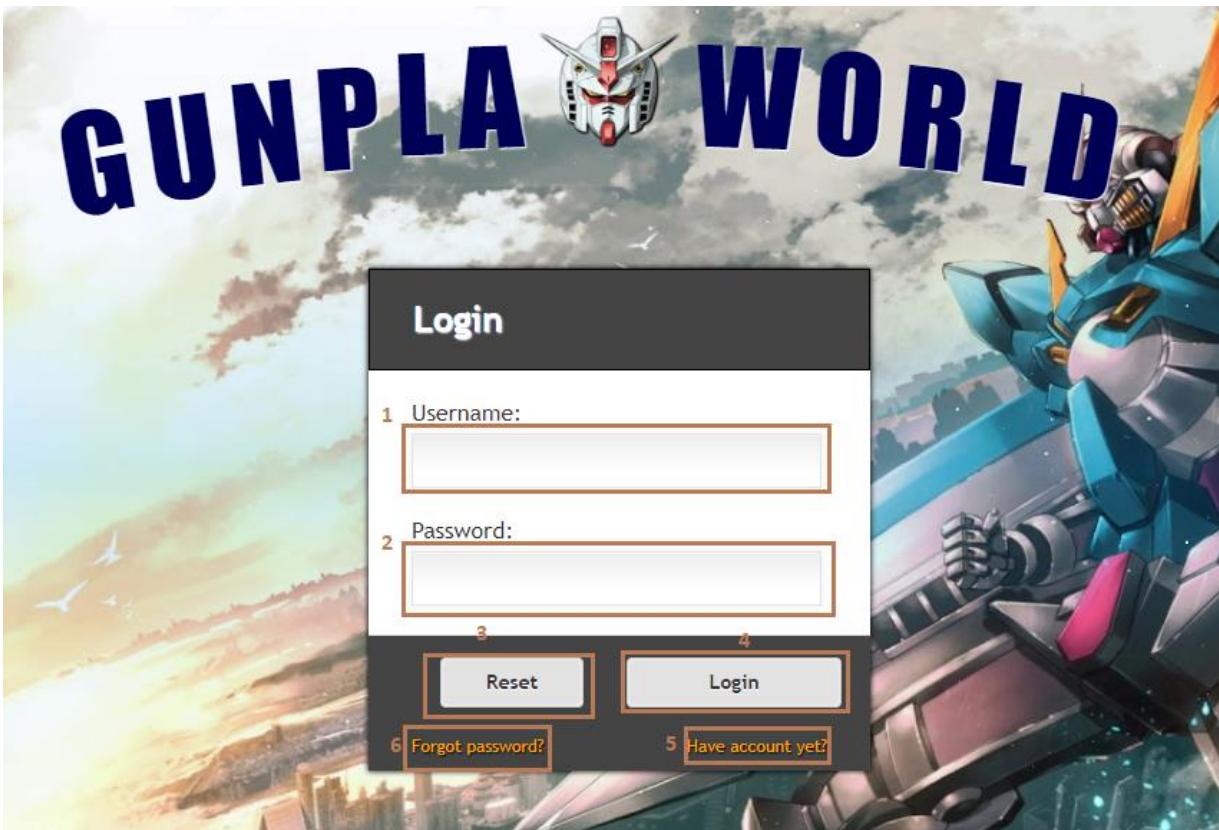
No	Field Name	Description	Read only	Mandatory	Control Type	Data Type	Length
1	Firstname	User's first name	No	Yes	Text input	String	<= 50

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

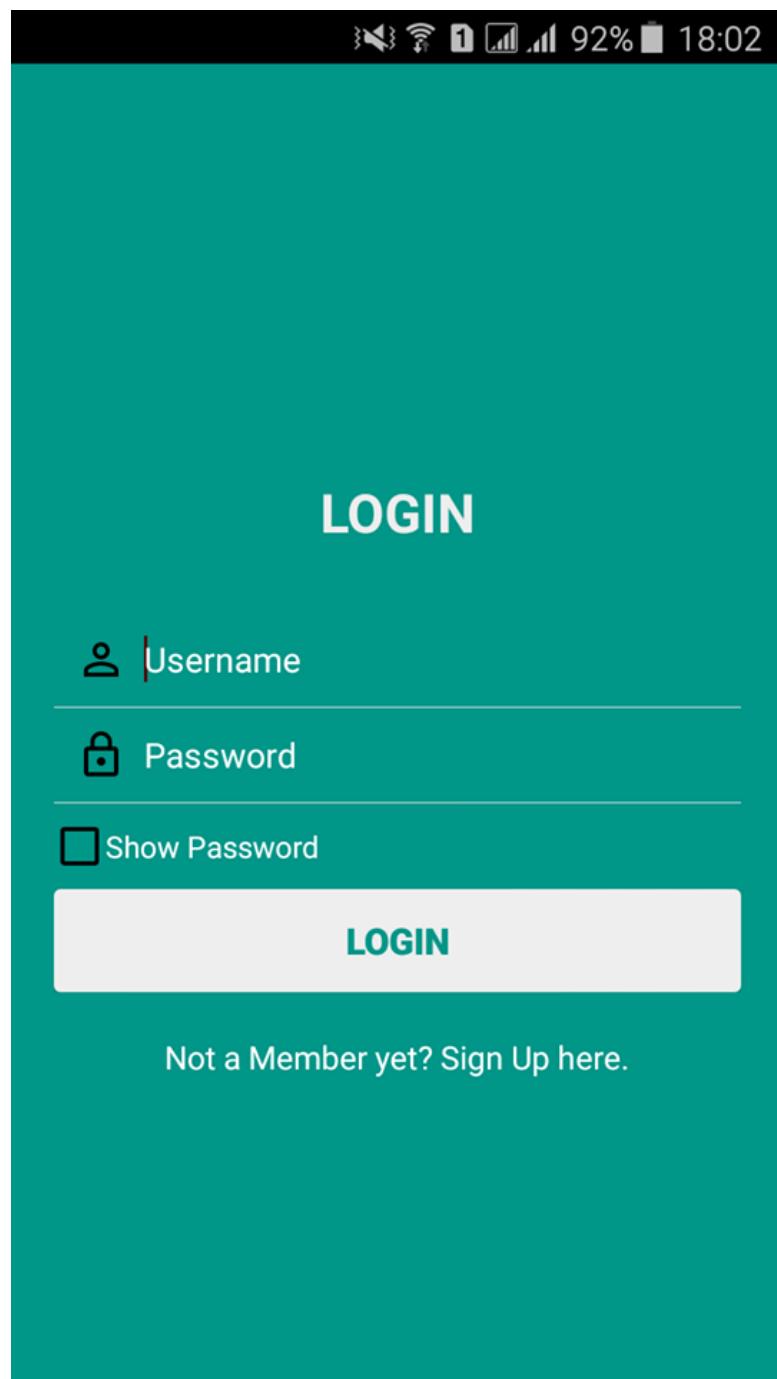
#### Buttons/Hyperlinks

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
6	Redirect to get forgot password page	For who forgot password	N/A	Get forgot password is shown

### **5.3.2. Mobile application**

#### **5.3.2.1. Login**



#### **5.3.2.2. Register account**

92% 18:02

## SIGN UP

 Username

 Email

 First Name

 Last Name

 Password

 Confirm Password

**SIGN UP**

Already have Account? [Login here.](#)

## 5.4. Authenticated User

### 5.4.1. Get profile

The screenshot shows a user profile page with the following sections:

- About:** Includes a profile picture (1), title "BuyerSeller" (3), full name "Le Do Dang Khoa" (4), email "N/A" (5), birthday "21-08-1997" (6), and a button "Exchange evaluation" (7).
- Address:** Shows address "268 Cach Mang Thang 8, phuong 3, quan Tan Binh" (8), tel "N/A" (9), join date "2018-10-29 04:31:47" (10), and status "Active" (11).
- Activity:** Displays proposal (0), buy (0), sell (1), and reputation (3) with 5 stars.
- TOP Ranking:** A sidebar listing top users with their names and profile pictures, ending at index 12.

A modal window displays three reviews:

- xuantruong ★★★★★ (2018-12-05 03:16:58)  
Reliable and very friendly ❤ I give you 5 stars
- tung ★★★★★ (2018-11-20 08:25:54)  
Nhiệt tình (y)
- lenguyenngocsang ★★★★★ (2018-11-20 08:18:50)  
Good seller !! Hope to see you again

### Fields

No	Field Name	Description	Read only	Mandatory	Control Type	Data Type	Length
1	Avatar	User's image	Yes	Yes	Img	Img	N/A
2	Statistic	User's statistics and ratings	Yes	Yes	N/A	String	N/A
3	Role	User's role	Yes	Yes	Label	String	N/A
4	Fullscreen	User's fullname	Yes	Yes	Label	String	N/A
5	Email	User's email	Yes	Yes	Label	String	N/A
6	Birthday	User's birthday	Yes	No	Label	String	N/A
8	Address	User's address	Yes	No	Label	String	N/A
9	Tel	User's mobile number	Yes	No	Label	String	N/A
10	Join Date	User's register datetime	Yes	Yes	Label	String	N/A
11	Status	User's status	Yes	Yes	Label	String	N/A
12	Top ranking	Top ranking user	Yes	Yes	N/A	N/A	N/A
13	Exchange evaluation	User's exchange evaluation	Yes	Yes	N/A	N/A	N/A

### Buttons/Hyperlinks

No	Function	Description	Validation	Outcome
7	View exchange evaluation	Get other user's exchange evaluation	N/A	Exchange evaluation dialog is shown

### 5.4.2. Edit profile

The screenshot shows a user profile editing interface. At the top left is the user's role and name: [BuyerSeller] tung. To the right is an 'EDIT' button. Below this is a circular profile picture of a man in a suit, with a placeholder 'Upload a different photo...' and a camera icon below it. Numbered callouts point to various elements: 1 points to the name; 2 points to the EDIT button; 3 points to the First name field containing 'Tung'; 4 points to the Email field containing 'tunghnmse62231@fpt.edu.vn'; 5 points to the Middle name field containing 'Minh'; 6 points to the Address field containing '15 Thống Nhất, Phường Tân Thành, Quận Tân Phú, TP.HCM'; 7 points to the Last name field containing 'Hoang Ngo'; 8 points to the Tel field containing '0985672459'; 9 points to the Birthday field containing '02-12-1997'; 10 points to the Status field showing 'Active'; 11 points to the 'SAVE' button; 12 points to the 'CANCEL' button; 13 points to a sidebar listing activity history: 'Proposal' (1), 'Buy' (1), 'Sell' (4); 14 points to the 'Exchange evaluation' button; 15 points to the profile picture; 16 points to the Reputation section showing '(4) ★★★★☆'.

### Fields

No	Field Name	Description	Read only	Mandatory	Control Type	Data Type	Length
1	Username and role	User's username for login and role	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	Status	User's status	Yes	Yes	Text input	String	N/A
13	Statistic	User's statistics and ratings	Yes	Yes	N/A	String	N/A
15	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
11	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
12	Cancel	All fields are reset to the beginning values	N/A	All fields are reset to the beginning values
14	Choose file	Update avatar	Check valid type of file image	Success: avatar is updated with the selected image from user

				Fail: show error message
16	View exchange evaluation	Display dialog with user's exchange evaluation	N/A	Display dialog with user's exchange evaluation

## 5.5. Administrator

### 5.5.1. Create model

#### 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
11	Check all	Check all images displaying on the view	No	No	Checkbox	N/A	N/A

### Buttons/Hyperlinks

No	Function	Description	Validation	Outcome
9	Add images	Choose model's image	Check valid type of file image	Image name is shown next by
10	Delete images	Delete model's image	N/A	Images are removed from view
12	Save	Send create 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
13	Cancel	Cancel creating model	N/A	Search model page is displayed

## 5.5.2. Edit model

The screenshot shows the 'Edit Model' interface for item 2426168. The left panel displays 'Model Information' with fields: Code (1), Name (2), Product Series (3), Series Title (4), Manufacturer (5), Price (6), and Status (7). The right panel shows the 'Description' (8) which includes HTML code describing the model, and 'Model's Images' (9) which lists three images (10) with options to add files (11) or delete them (12). Buttons for 'SAVE' (13) and 'CANCEL' (14) are at the top right.

### 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

11	Check all	Check all images displaying on the view	No	No	Checkbox	N/A	N/A
12	Model's images	Model's images	No	Yes	N/A	N/A	N/A

### Buttons/Hyperlinks

No	Function	Description	Validation	Outcome
9	Add images	Choose model's image	Check valid type of file image	Image name is shown next by
10	Delete images	Delete model's image	N/A	Images are removed from view
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 edit model	N/A	Selected model detail page is displayed

### 5.5.3. Delete model

The screenshot shows a mobile application interface for managing model kits. At the top, there's a search bar with the placeholder "Enter a search term" and a magnifying glass icon. To the right of the search bar is a blue "EDIT" button and a red-bordered "DELETE" button with a small red "1" indicating a pending action. Below the search bar is a "TOP FIVE" section with a "Model" heading and two thumbnail images of other model kits. On the left side of the main content area, there are navigation buttons for "CATEGORY", "OPTIONS", and "Product Series" (which is currently selected, shown in red). The main content area displays "MODEL INFORMATION" for the "AMX-103 Hamma-Hamma" model. It includes a large image of the model kit box art, which features a blue and white Gundam-like robot. To the right of the image are various details: Name: AMX-103 Hamma-Hamma, Manufacturer: Bandai, Scale: Master Grade, 1/100, Series: MG Other Gundam, Release Date: Mid May, 2018, Sales Price: 3,600yen, Item Code: 257516, and Rating: ★★★★★. A red box highlights the "DELETE" button.

### 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 shown

#### 5.5.4. Manage users

Latest Accounts						
<input type="text" value="Search..."/> <span style="float: right;">🔍</span>						
Order by <input type="button" value="Register Date"/> <span style="margin-left: 10px;">Descending</span> <span style="float: right;">2</span>						
4						
ID	Fullname	Username	Email	Date Time	Status	Address
31	Doan Van Hau	vanhau	doanvanhau@fpt.edu.vn	2018-12-05 11:06:14	Active	178 Nguyễn Văn Trỗi, Phường 8, Phú Nhuận, Hồ Chí Minh, Vietnam
30	Luong Xuan Truong	xuantruong	luongxuantruong@gmail.com	2018-12-05 02:12:53	Active	2 Thống Nhất, Tân Thành, Tân Phú, Hồ Chí Minh, Vietnam
29	Bui Khac Minh Quan	buikhacminhquan	quanbkm@fpt.edu.vn	2018-11-05 00:45:01	Active	19 Phan Dang Luu, phuong 2, quan Bình Tân
28	Nguyen Duc Tam	nguyenductam	tamnd@fpt.edu.vn	2018-11-05 00:43:33	Active	156 Phan Xich Long, phuong 12, quan Bình Thạnh
27	Pham Chi Khac	phamchikhac	khacpc@fpt.edu.vn	2018-11-05 00:42:14	Active	51 Nguyen Cuu Dam, phuong Tan Thanh, quan Tan Phu
26	Phan Minh Tri	phanminhtri	tripm@fpt.edu.vn	2018-11-05 00:40:40	Active	14 Cong Hoa, phuong 5, quan Tan Binh
25	Tran Trong Nghia	trantrongnghia	nghiatt@fpt.edu.vn	2018-11-05 00:39:07	Active	258 Thanh Cong, phuong Tan Thanh, quan Tan Phu
24	Mai Vu Cuong	mai vu cuong	cuongmv@fpt.edu.vn	2018-11-05 00:36:57	Active	169 Bac Hai, phuong Phu Tho Hoa, quan 10

#### 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	Order by filter	Order by filter	No	No	Dropdown list	String	N/A
3	Pagination	Pagination	No	No	N/A	String	N/A
4	List accounts	List accounts	No	Yes	N/A	String	N/A

### 5.5.5. Crawling

The screenshot shows the GUNDAM Crawl model interface. On the left is a sidebar with navigation links like Users, Models, Article, Event, Trade post, Schedule, and Labels (Important, Warning, Information). The main area is titled 'CRAWL' (1) and displays a table of crawl records. The table has columns: ID, Date Time, Crawl Records, New Records, and Status. The status column uses color coding: red for 'Crawling' (row 8), green for 'Done' (rows 7, 6, 5, 4, 3, 2, 1). A message at the top right says 'Crawl records: 14 New records: 14' (3). A large orange-bordered box highlights the table. On the far right, there is a toggle switch (2) with a green circle indicating it is turned on.

ID	Date Time	Crawl Records	New Records	Status
8	2018-12-09 15:42:25	N/A	N/A	Crawling
7	2018-12-04 19:54:42	62	59	Done
6	2018-10-30 01:49:02	253	253	Done
5	2018-10-25 11:12:11	253	0	Done
4	2018-10-25 00:35:25	253	249	Done
3	2018-10-25 00:17:47	34	34	Done
2	2018-10-23 09:26:12	721	69	Done
1	2018-10-23 04:11:21	894	894	Done

### Fields

No	Field Name	Description	Read only	Mandatory	Control Type	Data Type	Length
3	Count records	Counting crawl records and new records which are saved in database	Yes	No	Label	String	N/A
4	Log data	History crawling	Yes	Yes	Label	String	N/A

### Buttons/Hyperlinks

No	Function	Description	Validation	Outcome
1	Start crawl	Send crawl command to server	N/A	System displays message: "System is crawling successfully!"
2	Switch tracking	Switch on/off tracking crawl process	N/A	Number of crawl records and

				number of new records are displayed on the view
--	--	--	--	-------------------------------------------------

### 5.5.6. Manage schedule

The screenshot shows a management interface for three scheduled tasks:

- Gundam Crawl Schedule**: Status: *(Is Running)*, Cycle: 50 hours. Timer: 2 Days, 1 Hours, 59 Minutes, 39 Seconds.
- Trade Market Schedule**: Status: *(Is Stopped)*, Cycle: N/A. Timer: 00 Days, 00 Hours, 00 Minutes, 00 Seconds. Includes a START button.
- Event Update Schedule**: Status: *(Is Running)*, Cycle: 24 hours. Timer: 0 Days, 23 Hours, 59 Minutes, 50 Seconds.

### Fields

No	Field Name	Description	Read only	Mandatory	Control Type	Data Type	Length
1	Schedule name	Schedule name	Yes	Yes	Label	String	N/A
2	Schedule's status	Status of schedule	Yes	Yes	Label	String	N/A
3	Update cycle	Update cycle of schedule	Yes	Yes	Label	String	N/A
4	Countdown	Countdown time	Yes	Yes	Label	String	N/A

### Buttons/Hyperlinks

No	Function	Description	Validation	Outcome
5	Stop schedule	Send stop schedule command to server	N/A	System displays message: "Schedule is stopped successfully!"
6	Start schedule	Send start schedule command to server	N/A	System displays message: "Schedule starts successfully!"

### 5.5.7. Search article

Article Manage article

Latest Articles

ID	Title	Author	Date posted	Category	Approval status	Action
256	Gunpla Pre-orders and New Releases for December-January 2018	khanh	2018/11/26	Shopping Guides	Pending	<a href="#">Link</a> <a href="#">Edit</a>
207	Article 11/25	tung	2018/11/25	News	Pending	<a href="#">Link / Edit</a>

### Fields

No	Field Name	Description	Read only	Mandatory	Control Type	Data Type	Length
1	Search article	Search article by title	No	No	Text input	String	N/A

3	Sort by date	Sort by date newest or oldest first	No	Yes	Drop-down list	String	N/A
4	Article's category	Category to search article	No	Yes	Dropdown list	String	N/A
5	Article's status	Status to search article	No	Yes	Dropdown list	String	N/A
6	Search record	Result of searching	No	Yes	N/A	N/A	N/A

### Buttons/Hyperlinks

No	Function	Description	Validation	Outcome
2	Search	Search articles by title	No	Search articles by title and filters
7	Link to the article	Link to the article	No	Redirects to the article
8	Link to edit page	Link to edit page	No	Redirect to the edit page

#### 5.5.8. Edit article

The screenshot shows the AdminGWA application interface for editing an article. The left sidebar contains navigation links for Users, Models, Article (selected), Event, Trade post, and Schedule. Labels include Important, Warning, and Information. The main area shows the 'Edit article' form with the following fields:

- Title:** Gunpla Pre-orders and New Releases for December-January 2018 (marked with 1)
- Description:** This article is crawled by system !!! (marked with 2)
- Thumbnail image:** An image of a white and blue Gundam model (marked with 3)
- Category:** Shopping Guides (marked with 4)
- Status:** Pending (marked with 5)
- Content:** A rich text editor with various formatting options (marked with 6)



## Fields

No	Field Name	Description	Read only	Mandatory	Control Type	Data Type	Length
1	Article's title	The title of the article	No	Yes	Input field	String	<=250
4	Article's category	The category of the article	No	Yes	Drop-down list	String	N/A
2	Article's description	Short description of the article	No	Yes	Textarea	String	<=250
3	Article's thumbnail	Thumbnail of the article	No	Yes	Image	String	N/A
5	Article's status	Status of the article	No	Yes	Drop-down list	String	N/A
6	Article's content	The content of the article	No	Yes	Textarea	String	N/A

## Buttons/Hyperlinks

No	Function	Description	Validation	Outcome
7	Submit article	Let the user submit the article	Yes	Validate all fields and submit article.
8	Delete article	Let the user delete the article	Yes	Prompts user "Are you sure you want to delete this article?"

				Deletes the article if the user select Yes, cancel if No
--	--	--	--	----------------------------------------------------------

### 5.5.9. Create article

The screenshot shows the 'Create article' page in the AdminGWA application. The left sidebar contains navigation links for Users, Models, Article, Event, Trade post, and Schedule. Below that is a section for LABELS with categories: Important, Warning, and Information. The main content area has the following fields:

- Title**: A text input field with a red border, labeled 1.
- Description**: A large text area with a red border, labeled 2.
- Thumbnail image**: A file upload input field with a red border, labeled 3.
- Category**: A dropdown menu set to 'News' with a red border, labeled 4.
- Status**: A dropdown menu set to 'Approved' with a red border, labeled 5.
- Content**: A rich text editor with a red border, labeled 6.
- Submit**: A button at the bottom with a red border, labeled 7.

### Fields

No	Field Name	Description	Read only	Mandatory	Control Type	Data Type	Length
1	Article's title	The title of the article	No	Yes	Input field	String	<=250
4	Article's category	The category of the article	No	Yes	Drop-down list	String	N/A

2	Article's description	Short description of the article	No	Yes	Textarea	String	<=250
3	Article's thumbnail	Thumbnail of the article	No	Yes	Image	String	N/A
5	Article's status	Status of the article	No	Yes	Drop-down list	String	N/A
6	Article's content	The content of the article	No	Yes	Textarea	String	N/A

### Buttons/Hyperlinks

No	Function	Description	Validation	Outcome
7	Submit article	Let the user submit the article	Yes	Validate all fields and submit article.

### 5.5.10. Search event

Title	Description	Start date	End date	Ticket price	Status	Action
new event for mobile	sasas	2020-12-12	2030-12-12	deobiet	Active	<a href="#">Link</a> <a href="#">Edit</a>
GUNPLA BUILDERS WORLD CUP 2018 REGULATION	Quốc gia và vùng lãnh thổ tổ chức: 16 quốc gia và vùng lãnh thổ Nhật Bản, Trung Quốc, Hàn Quốc, Đài Loan, Hồng Kông, Thái Lan, Singapore, Malaysia, Philippines, Indonesia, Úc, Ý, Pháp, Việt Nam, Mỹ, Canada. ■ Thời gian diễn ra cuộc thi GBWC 2017 sẽ đư	2018-12-12	2019-11-11	10.000 VND	Active	<a href="#">Link</a> <a href="#">Edit</a>

### Fields

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

1	Search event	Search event by title	No	No	Text input	String	N/A
3	Sort by date	Sort by date newest or oldest first	No	Yes	Drop-down list	String	N/A
4	Event's status	Status to search article	No	Yes	Dropdown list	String	N/A
6	Search record	Result of searching	No	Yes	N/A	N/A	N/A

### Buttons/Hyperlinks

No	Function	Description	Validation	Outcome
2	Search	Search articles by title	No	Search articles by title and filters
6	Link to the article	Link to the event	No	Redirects to the event
7	Link to edit page	Link to edit page	No	Redirect to the edit page

#### 5.5.11. Create event

The screenshot shows the AdminGWA application interface. On the left is a dark sidebar with navigation links for 'Users', 'Models', 'Article', 'Event', 'Trade post', 'Schedule', and labels 'Important', 'Warning', and 'Information'. The main area has a blue header bar with the title 'Event Create Event' and a user profile icon. Below the header is a form titled 'Create a new event'. The form fields are: 'Title' (highlighted with a red box and number 1), 'Location' (highlighted with a red box and number 2), 'Description' (highlighted with a red box and number 3), 'Thumbnail image' (highlighted with a red box and number 4, showing a small thumbnail icon), 'Start date' (highlighted with a red box and number 5), 'End date' (highlighted with a red box and number 6), and 'Start registration date' (highlighted with a red box and number 7). The bottom right corner of the screenshot shows a scroll bar.

The form consists of the following fields:

- Start registration date (7)
- End registration date (8)
- Maximum number of attendee (9)
- Min number of attendee (10)
- Ticket price (11)
- Status (12)
  - Active
- Content (13)
  - Rich text editor toolbar
- Submit (14)

## Fields

No	Field Name	Description	Read only	Mandatory	Control Type	Data Type	Length
1	Event's title	Title of the event	No	Yes	Text input	String	<= 250
2	Event's location	Location of the event	No	Yes	Text input	String	<=250
3	Event's description	Description of the event	No	Yes	Text input	String	<=250
4	Event's thumbnail	Thumbnail of the event	No	Yes	Image	N/A	N/A
5	Event's start date	Event's start date	No	Yes	Date input	String	N/A
6	Event's end date	Event's end date	No	Yes	Date input	String	N/A

7	Event's start registration date	Date when users can register to the event	No	Yes	Date input	String	N/A
8	Event's start registration date	End date of when users can register to the event	No	Yes	Date input	String	N/A
9	Max number of attendee	Maximum number of tickets available of the event	No	Yes	Text input	int	N/A
10	Min number of attendee	Minimum number of attendee required	No	Yes	Text input	int	N/A
11	Ticket price	Ticket price of the event	No	Yes	Text input	String	<=20
12	Event status	Status of the event	No	Yes	Drop-down list	String	N/A
13	Event content	Content of the event	No	Yes	Textarea	String	N/A

### Buttons/Hyperlinks

No	Function	Description	Validation	Outcome
14	Submit event	Submit the event	Yes	Validate data and submit the event

### **5.5.12. Edit event**

Fields

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

1	Event's title	Title of the event	No	Yes	Text input	String	<= 250
2	Event's location	Location of the event	No	Yes	Text input	String	<=250
3	Event's description	Description of the event	No	Yes	Text input	String	<=250
4	Event's thumbnail	Thumbnail of the event	No	Yes	Image	N/A	N/A
5	Event's start date	Event's start date	No	Yes	Date input	String	N/A
6	Event's end date	Event's end date	No	Yes	Date input	String	N/A
7	Event's start registration date	Date when users can register to the event	No	Yes	Date input	String	N/A
8	Event's start registration date	End date of when users can register to the event	No	Yes	Date input	String	N/A
9	Max number of attendee	Maximum number of tickets available of the event	No	Yes	Text input	int	N/A
10	Min number of attendee	Minimum number of attendee required	No	Yes	Text input	int	N/A
11	Ticket price	Ticket price of the event	No	Yes	Text input	String	<=20
12	Event status	Status of the event	No	Yes	Drop-down list	String	N/A
13	Event content	Content of the event	No	Yes	Textarea	String	N/A

## Buttons/Hyperlinks

No	Function	Description	Validation	Outcome
14	Submit event	Submit the event	Yes	Validate data and submit the event

### 5.5.13. List proposal

The screenshot shows a web interface for managing proposals. At the top, there's a header bar with 'Proposal' and 'Manage Proposal'. Below it, a navigation bar has 'Events' selected. The main content area displays a table of proposals. A red box surrounds the entire table. Above the table, a red number '1' is placed. To the right of the table, a red number '2' is placed next to a blue 'Detail' link. The table columns are 'Title', 'User', and 'Action'. The data rows show three proposals: 'first proposal ever' by 'tung', 'send proposal' by 'khanh', and 'third proposal' by 'khanh'. The 'Action' column contains 'Detail' and 'Action' links.

## Fields

No	Field Name	Description	Read only	Mandatory	Control Type	Data Type	Length
1	Proposal list	List of the proposals posted by the users	No	Yes	N/A	N/A	N/A

## Buttons/Hyperlinks

No	Function	Description	Validation	Outcome
2	Proposal detail link	Link to the proposal's detail	No	Redirects to the proposal's detail

#### 5.5.14. Proposal detail

Proposal Manage proposal

1

Title: send proposal
Description: description
Location: oolocation
Content Event o nha toi
<a href="#">Create a new event from this data</a>

2

#### Fields

No	Field Name	Description	Read only	Mandatory	Control Type	Data Type	Length
1	Proposal detail	Detail of the proposal	Yes	Yes	N/A	N/A	N/A

#### Buttons/Hyperlinks

No	Function	Description	Validation	Outcome
2	Create event with proposal	Link to the event create view	No	Redirects to the event create view with the proposal's data loaded

## 6. Database Design

## 6.1. Entity Relationship Diagram (ERD)

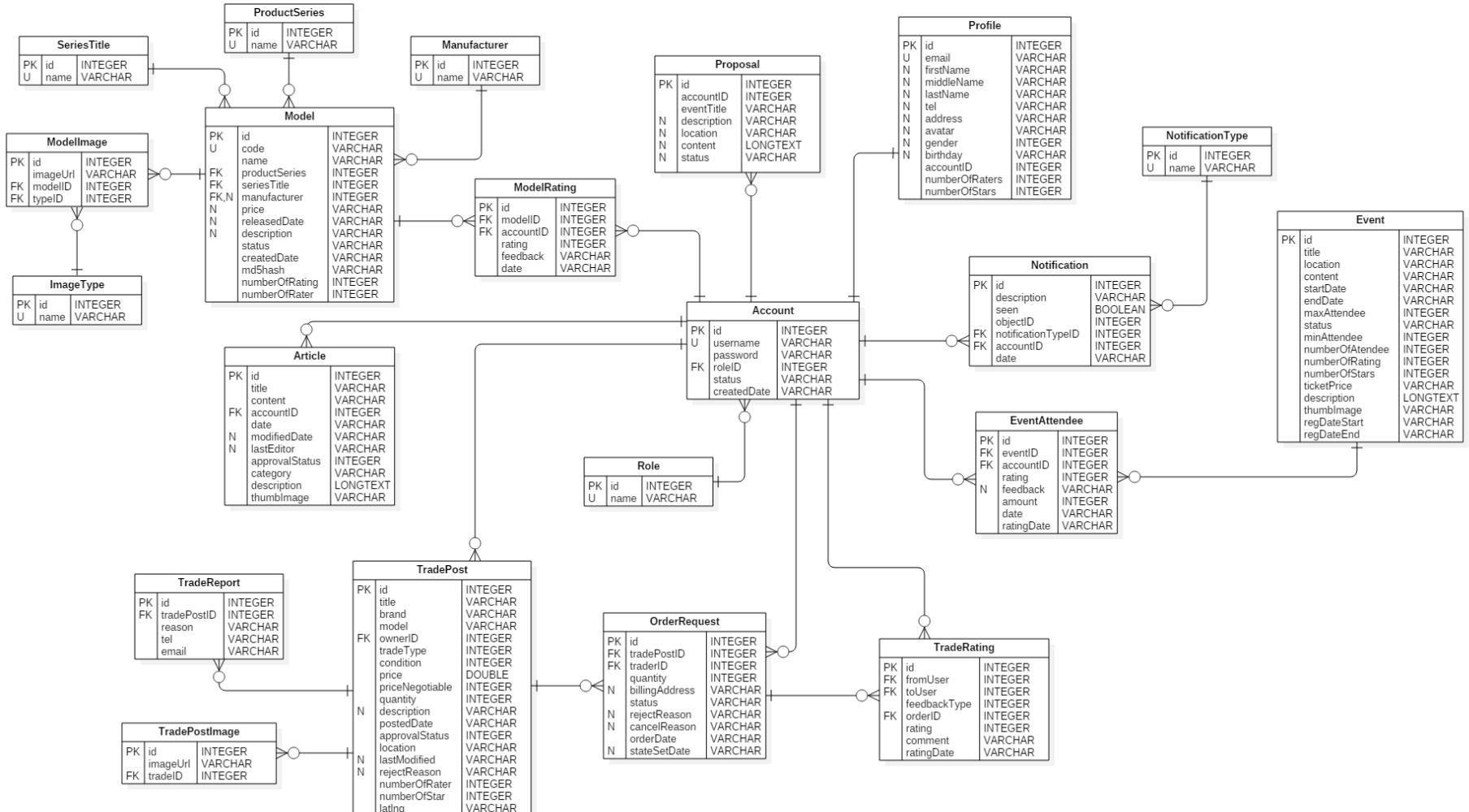


Figure 67 - Entity Relationship Diagram

## 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
ModelRating	Contain model's user review evaluation
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

Table 16 - Entity Data Dictionary

## **7. Algorithms**

### **7.1. As-you-type search with instant results and pagination with sessionStorage**

#### **7.1.1. Definition**

"As-you-type" search is a user interface interaction method to progressively search for and filter through text. As the user types text, one or more possible matches or nearly matches for the text are found and immediately presented to the user.

"Instant results" make as-you-type search offer actual results instead of just query reformulations, provide shortcut to search results.

"Pagination with sessionStorage" is a pagination method that saves records of user's each selected page and nextby pages to sessionStorage and retrieves when user reselected.

#### **7.1.2. Define problems**

User sometimes have to type the entire word or phrase to find models they were looking for and may not know the exact name of gundam models, this lead to no result matches situation. We also want to reduce HTTP requests from client to speed up the system, the fewer requests send to server, the faster client will load.

#### **7.1.3. Solution**

The processing model for search gundam models follow these steps:

1. Save all gundam model names and thumb images to user's localStorage
  - When user goes to search model page, system check if data is in localStorage or not
    - + If data not exist in localStorage, system request to get all model names and thumb images from server and save to localStorage with current datetime.
    - + If data existed in localStorage, system will check the last saved datetime compare to present, request to get data from server if larger than 2 days
2. Find gundam models which match or nearly match with user's input keyword

- When user types any character in search input, system find models which have name contains the text input
  - Next step, use Levenshtein edit distance algorithm to calculate percentage string comparison between models name and text input from user
    - Append all the models which have larger than or equal to 50% string comparison to result list
    - Display results with thumb image and name on the view
- 3. If user selects one of those, redirect user to model specific detail page
- 4. Pagination with sessionStorage, reduce request send to server, system doesn't have to send request to get data when user re-clicks on a page
  - When user clicks on a page, check if data of that page is in sessionStorage or not
    - + If existed, system get current page's data from sessionStorage and render to view
    - + If not, request to server to get current page's data and render to view
  - System also prepares data of two pages nextby for user

#### 7.1.4. Flow chart

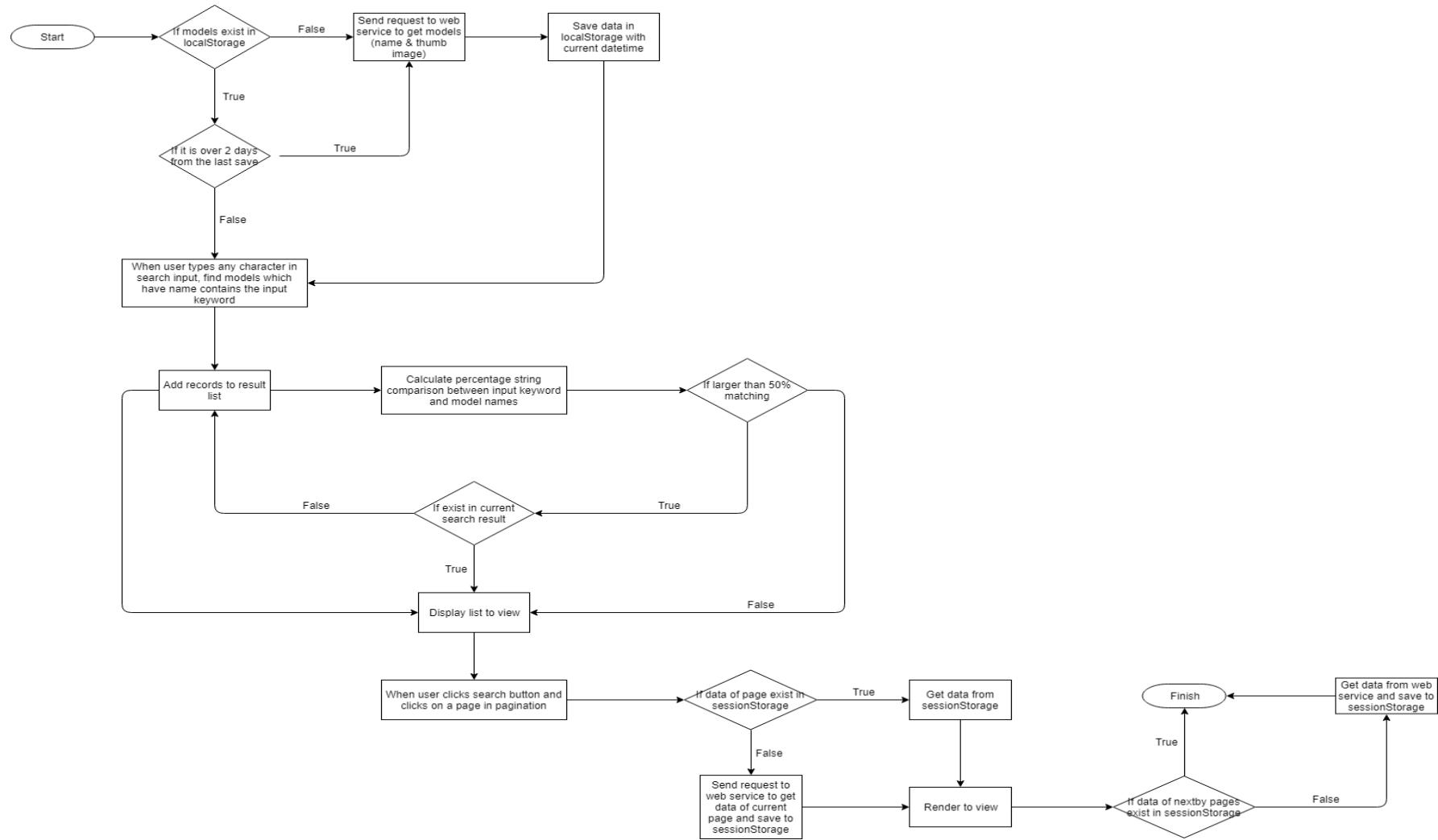


Figure 68 - As-you-type search workflow

## **7.2. MD5 hash function**

### **7.2.1. Definition**

"MD5" is a widely used one-way cryptographic hash function, which produces a hash of 128-bit hash value. It returns a 32 character string of hexadecimal digits 0-9 & a-f and returns null if the input is null.

### **7.2.2. Define problems**

When execute crawling data from sites, crawl records must be matching with data in database. Matching by traditional way waste time and memory when need to compare with multi attributes of a model.

### **7.2.3. Solution**

- When crawling a model, combine following attributes of a gundam model to a string:
  - + name
  - + product series
  - + series title
  - + manufacturer
- Use MessageDigest to find MD5 hash value of string
- Compare to all other model's MD5 hash value

## **7.3. Schedule Tracking**

### **7.3.1. Definition**

"Schedule Tracking" is the process which allows administrator to track the system schedules with countdown time, status and also history execute updates.

### **7.3.2. Define problems**

Our system needs to have some schedules to make it more convenient for administrator and also users to interact with system such as schedule about crawling data, schedule about auto disable all events which don't have enough minimum attendee after end registration date... Administrator also needs to track the schedule for information

### **7.3.3. Solution**

- Use Thread to implement Schedule
- When administrator starts a schedule, system starts Thread with input update cycle (hour), datetime when execute update, status...
- When administrator stops a schedule, system interrupts the selected Schedule thread and reset all fields (cycle, status,...)
- When administrator goes to Manage Schedule view:
  - + System get status, update cycle, log data from all Schedules
  - + System calculate countdown time of each schedule with the following formula (if the schedule is running):

$$\text{Countdown time} = \text{last execute update time} + \text{cycle} - \text{current time}$$

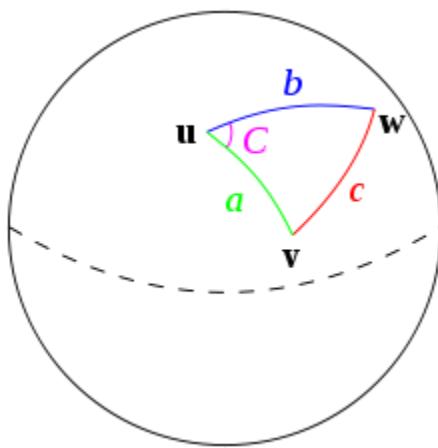
With “last execute update time” is datetime when the schedule executed update process, “cycle” is the interval input from administrator and “current time” is now.

- + Next step, system responses all above data to admin web app
- + System calculate counting down time and render to view with javascript at client-side

## 7.4. Search nearby location with Haversine formula

### 7.4.1. Definition

The haversine formula determines the great-circle distance between two points on a sphere given their longitudes and latitudes. Important in navigation, it is a special case of a more general formula in spherical trigonometry, the law of haversines, that relates the sides and angles of spherical triangles.



$$a = \sin^2\left(\frac{\Delta\varphi}{2}\right) + \cos \varphi_1 \cdot \cos \varphi_2 \cdot \sin^2\left(\frac{\Delta\lambda}{2}\right)$$

$$c = 2 \cdot \text{atan2}(\sqrt{a}, \sqrt{1-a})$$

$$d = R \cdot c$$

### 7.4.2. Define problems

When user want to find an exchange around them with specific range, system will help them find with location and range by using haversin formula.

### 7.4.3. Solution

- When search an exchange user can setting up their location by auto or manual select address.

-

- System will geocode this address from user and all address from all exchange to latlng position.
- System will calculate distance from user position and exchange position using haversin formula and compare with range inputted. If distance is lower than range inputted, system will select this exchange to search result list.

## **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**

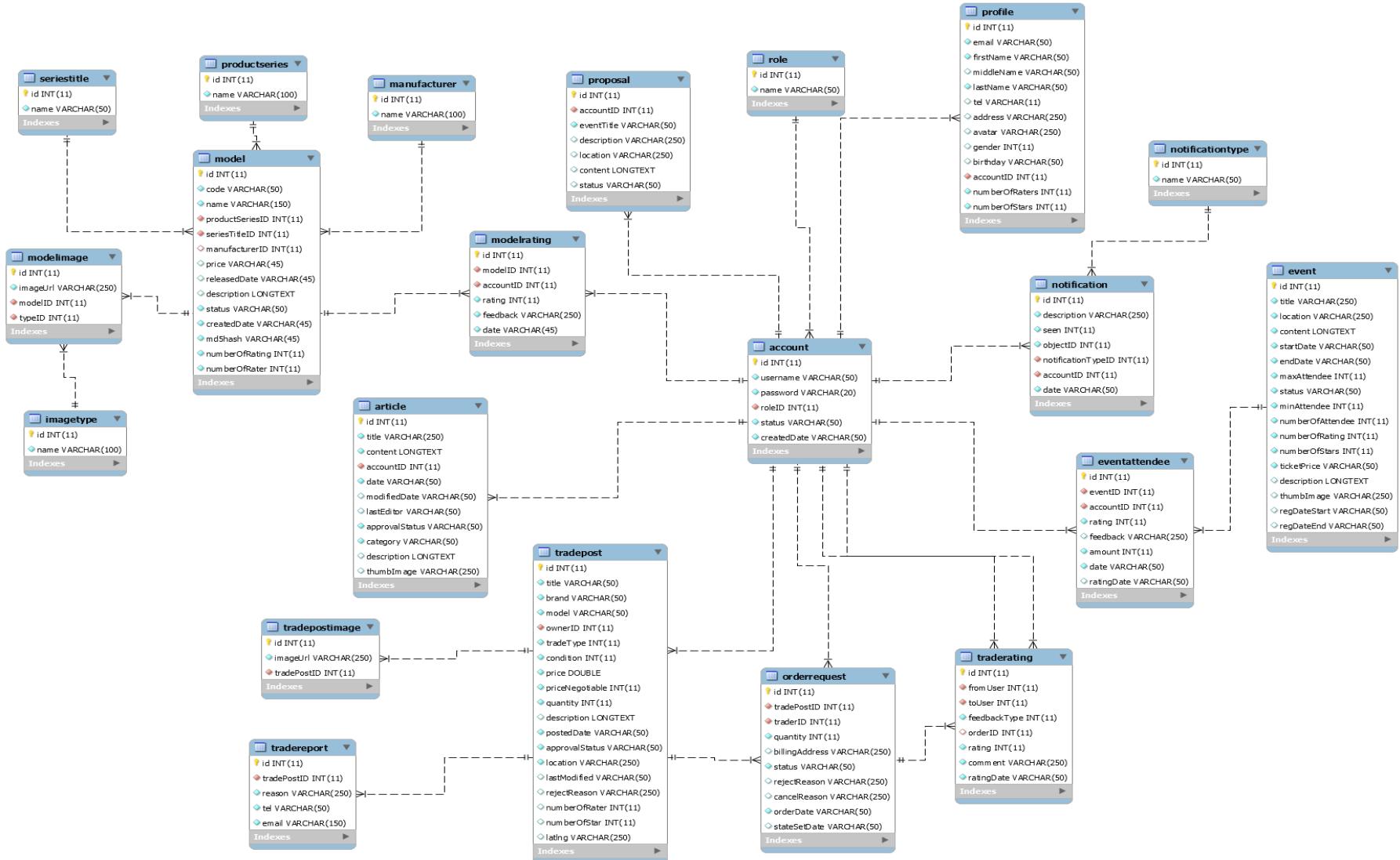


Figure 69 - Physical Diagram

## 2.2. Data dictionary

Data dictionary: describe content of all tables	
Table name	Description
model	Table which contains model information
seriestitle	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
modelrating	Table which contains user review evaluation on models
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

Table 17 - Physical Diagram Dictionary

Entity name	Attributes	Description	Null
model	id {pk}	Auto increment unique identifier	No
	code	Model code	No
	name	Model name	No

	productSeriesID {fk}	Model's product series	No
	seriesTitleID {fk}	Model's series title	No
	manufacturerID {fk}	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
	md5hash	Model's md5 hash value	No
	numberOfRating	Number of stars rating	No
	numberOfRater	Number of users who make review evaluation	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
modelrating	id {pk}	Auto increment unique identifier	No
	modelID {fk}	Model which is evaluated	No
	accountID {fk}	User who makes evaluation	No
	rating	Number of stars rating	No
	feedback	User's feedback	No
	date	Rating date	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

	createdDate	Account's created datetime	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
	content	Proposal event's content	Yes
	status	Proposal event status	Yes
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
	description	Article's description	Yes
notification	thumbImage	Article's thumb image url	Yes
	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	date	Notification sent datetime	No
	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
	description	Event's description	Yes
	thumbImage	Event's thumb image url	Yes
	regDateStart	Event's start registration date	Yes
	regDateEnd	Event's end registration date	Yes
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
	ratingDate	Date when user evaluates event	Yes
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
tradepost	numberOfStars	Number of star ratings from users, this attribute is to improve query performance	No
	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
	lastModified	Trade post's last modified	Yes
	rejectReason	Trade post's reject reason by admin	Yes
	numberOfRater	Number of users who make rating	Yes
	numberOfStar	Number of stars rating	Yes
	latlng	Trade post's location latitude and longitude	Yes
tradereport	id {pk}	Auto increment unique identifier	No
	tradePostID {fk}	ID of the trade post which is reported by user	No
	reason	Report reason	No
	tel	Reporter's mobile number	No
	email	Reporter's email	No
tradeimage	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 evaluation is from trade owner to user or user to trade owner	No
	orderID {fk}	Each order request have at most 2 ratings	Yes
	rating	Number of stars rating	No
	comment	User's feedback about trade post	No
	ratingDate	Rating date	No
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
- System sends email to the user in 10 seconds max
- User can view information with pagination which are saved at client-side
- Crawling can take at least 30 minutes
- Notification is sent in 1 second 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

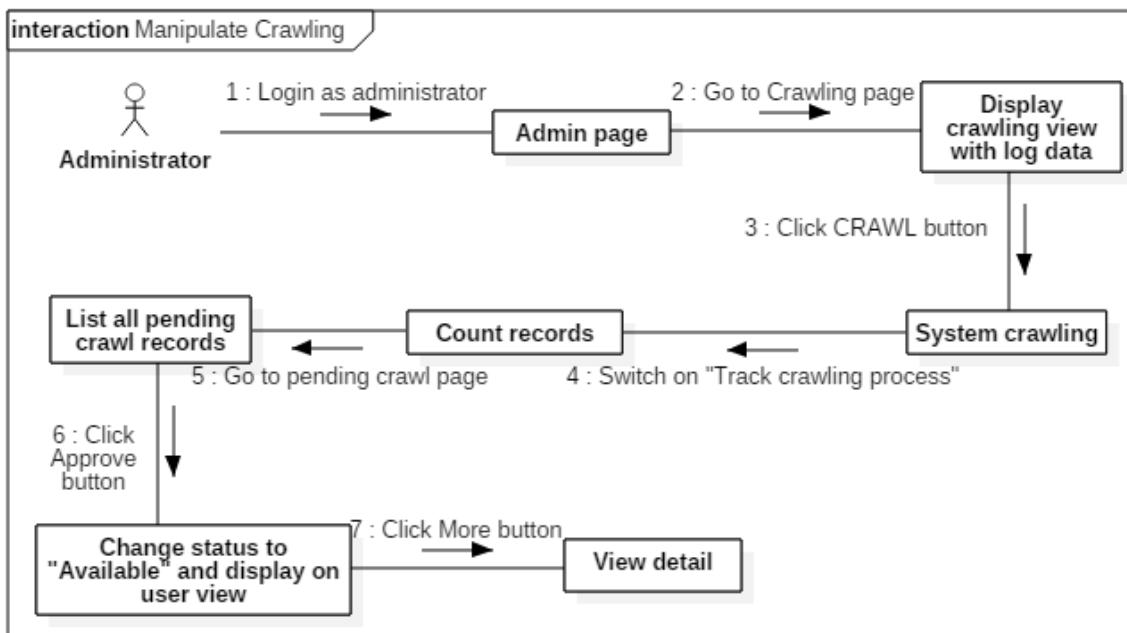


Figure 70 - Manipulate crawling test flow

### 5.1.2. Search for model information

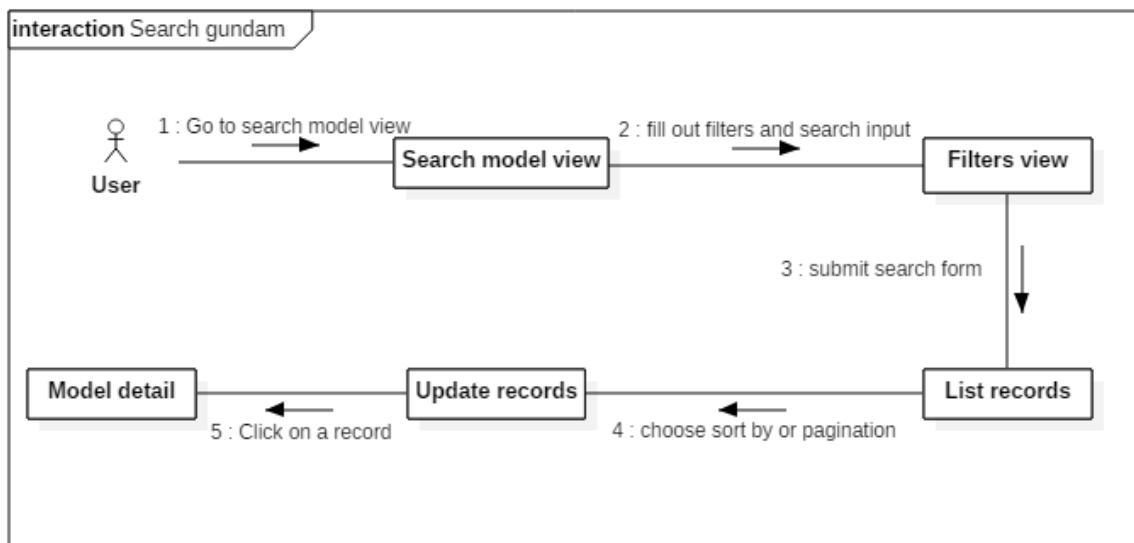


Figure 71 - Search gundam test flow

### 5.1.3. Manipulate trade post

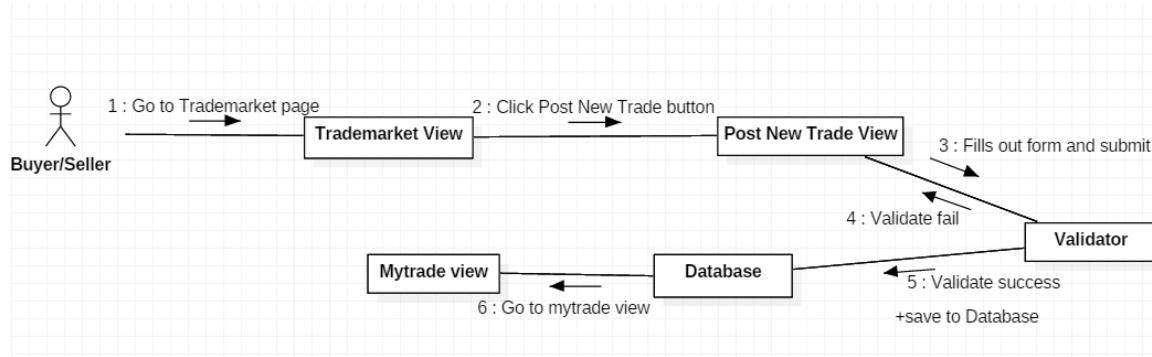


Figure 72 - Manipulate trade post test flow

#### 5.1.4. Request order to trade post for exchange

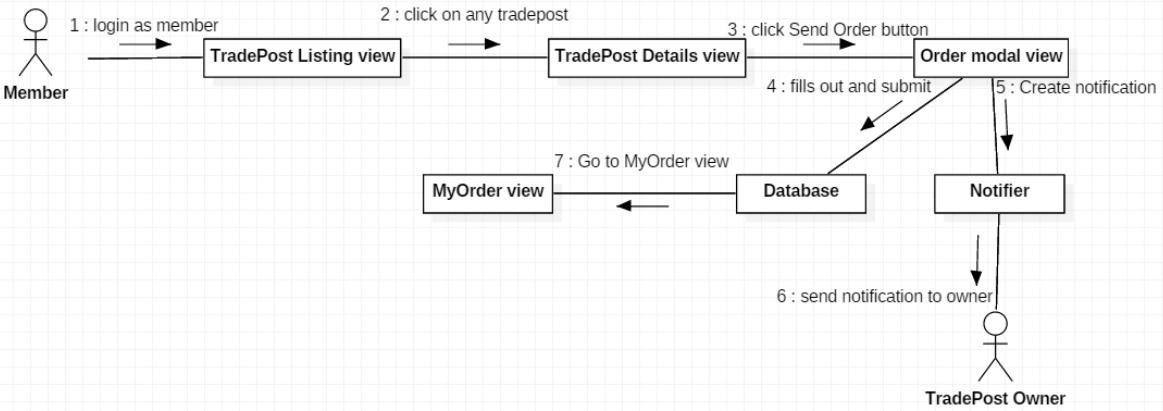


Figure 73 - Request order test flow

#### 5.1.5. Register for event

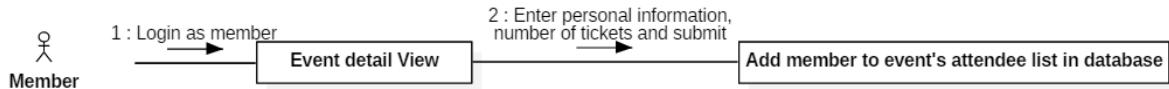


Figure 74 - Register event test flow

#### 5.1.6. Create article

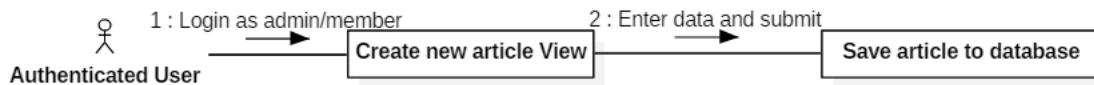


Figure 75 - Create article test flow

### 5.2. Web or Mobile Application Test case

#### 5.2.1. Manipulate crawling

ID	Test Case Description	Test Case Procedure	Expected output	Result	Test Date

AMC 1	Start crawl	1. Login with role “Admin” 2. Go to “Crawling” view. 3. Click CRAWL button.	Display log data with status “Crawling”  Display system is crawling message	Success	2018-12-04 2:22 PM
AMC 2	Track crawling process	4. Switch on icon “Track crawling process”	Counting crawl records and new records (which is saved to database after matching)	Success	2018-12-04 2:22 PM
AMC 3	Review pending crawl records	5. Go to pending crawl records page	Display records with status “crawlpending”	Success	2018-12-04 2:22 PM
AMC 4	Approve pending crawl records	6. Click approve button	Selected record’s status is changed available and display on user view	Success	2018-12-04 2:22 PM

### 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	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	Success	2018-12-04 2:45 PM
SFM2	Choose filter sorts by and pagination	4. Choose sort by options or pagination	Updated list records follow the filters and pagination	Success	2018-12-04 2:45 PM
SFM3	Show model detail information	7. Click on a record’s “More information” button	The selected model detail page is shown	Success	2018-12-04 2:45 PM

### 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.	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.	Success	2018-12-04 3:05 PM
MTP2	Post New Trade with invalid data.	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.	Success	2018-12-04 3:05 PM

### 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"> <li>Click “Send Order” button on trade post details view.</li> <li>Enter valid quantity</li> <li>fills out Trader information form</li> <li>Click submit</li> </ol>	Go to My order view	Success	2018-12-04 3:05 PM
OTT2	Send order with invalid quantity	<ol style="list-style-type: none"> <li>Click “Send Order” button on trade post details view.</li> <li>Enter invalid quantity</li> <li>fills out Trader information form</li> <li>Click submit</li> </ol>	Stay at send order modal view and show error message for invalid quantity..	Success	2018-12-04 3:05 PM

### 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"> <li>Enter card number and all required information</li> <li>Click on “Register”</li> </ol>	Save information to database and a Popup shows up saying “Registration complete”	Success	2018-12-04 3:15 PM
RE2	Register with a invalid card	<ol style="list-style-type: none"> <li>Enter card number and all required information</li> <li>Click on “Register”</li> </ol>	Popup shows saying “Card is invalid!”	Success	2018-12-04 3:15 PM
RE3	Register with the amount of tickets exceeding	<ol style="list-style-type: none"> <li>Enter all required information with the number of tickets higher than the amount of</li> </ol>	Popup shows saying “There aren’t enough	Success	2018-12-04

	the remaining tickets available	remaining tickets for that event 9. Click on “Register”	tickets! Please reenter the amount of tickets.”		3:15 PM
--	---------------------------------	------------------------------------------------------------	----------------------------------------------------	--	---------

#### 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!”	Success	2018-12-04 3:30 PM

## F. Software User's Manual

### 1. Installation Guide

#### 1.1. Setting up environment at server side

- The specifications are based on the dependencies requirements and performance test result from previous section of this document.

##### 1.1.1. Hardware Requirement

For Server	
Criteria	Minimum Requirements

<b>Internet Connection</b>	Cable, Wi-Fi (8 Mbps)
<b>Computer Processor</b>	Intel® Core™ i3 @ 2.0Ghz
<b>Computer Memory</b>	4 GB RAM
<b>Operating system</b>	Window 7

### **For smartphone**

<b>Criteria</b>	<b>Minimum Requirements</b>
<b>Operating System</b>	Android 4.0.3 Ice Cream Sandwich
<b>Processor</b>	CPU dual core 1.4 GHz
<b>Memory</b>	1GB RAM
<b>Operating System</b>	Android 4.0.3 Ice Cream Sandwich

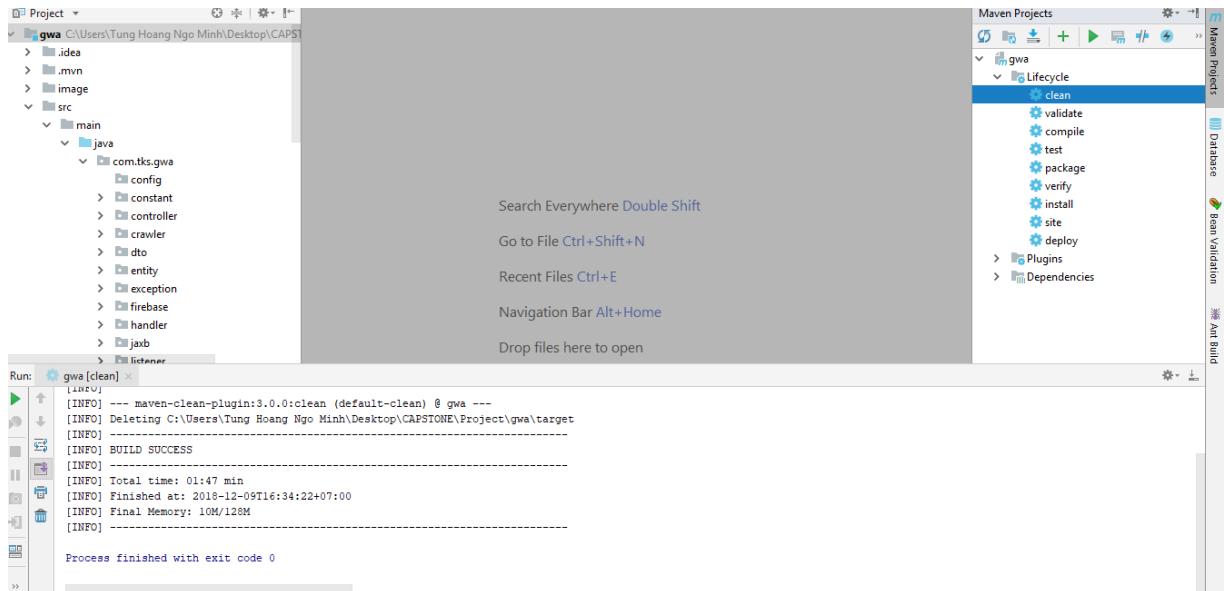
#### **1.1.2. Software Requirement**

<b>Software</b>	<b>Name and version</b>
<b>Operating System</b>	Window 10 1803
<b>Environment</b>	Java EE 8, Android SDK APIs level 15 or above
<b>Modeling Tool</b>	StarUML 2.8.1
<b>IDE</b>	IntelliJ IDEA 2018.2.3, Android Studio 3.1.3
<b>DBMS</b>	MySQL 5.7.20, MySQL Workbench 6.3.9 CE

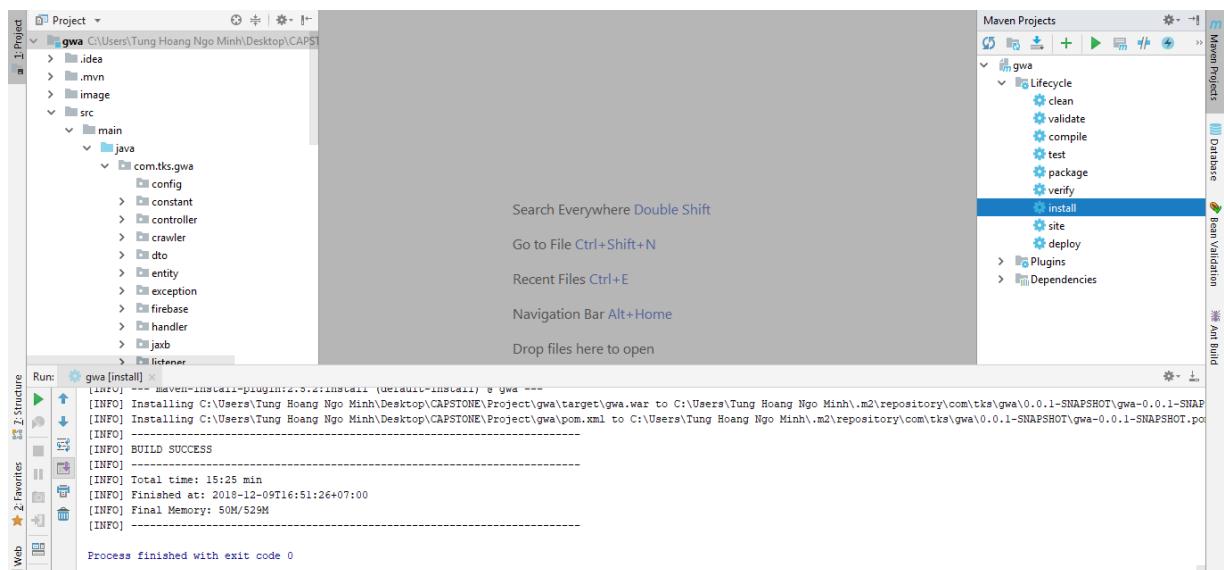
<b>Source control</b>	SourceTree 2.6.10
<b>Web browser</b>	Chrome 62 or above

## 1.2. Web Application Deployment Process

Open IntelliJ IDEA application. Click on tab “Maven Projects”, then click “clean” option



Continue to click on “install” option



After that, go to directory follow this path

```
[INFO] Installing C:\Users\Tung Hoang Ngo Minh\Desktop\CAPSTONE\Project\gwa\target\gwa.war to C:\Users\Tung Hoang Ngo Minh\.m2\repository\com\tks\gwa\0.0.1-SNAPSHOT\gwa-0.0.1-SNAPSHOT.jar
[INFO] Installing C:\Users\Tung Hoang Ngo Minh\Desktop\CAPSTONE\Project\gwa\pom.xml to C:\Users\Tung Hoang Ngo Minh\.m2\repository\com\tks\gwa\0.0.1-SNAPSHOT\gwa-0.0.1-SNAPSHOT.pom
[INFO]
```

Copy file “gwa.war” to folder “webapps” of Tomcat

Name	Date modified	Type	Size
classes	12/9/2018 4:45 PM	File folder	
generated-sources	12/9/2018 4:45 PM	File folder	
generated-test-sources	12/9/2018 4:45 PM	File folder	
gwa	12/9/2018 4:46 PM	File folder	
maven-archiver	12/9/2018 4:50 PM	File folder	
maven-status	12/9/2018 4:45 PM	File folder	
surefire-reports	12/9/2018 4:46 PM	File folder	
test-classes	12/9/2018 4:45 PM	File folder	
gwa.war	12/9/2018 4:51 PM	WAR File	63,532 KB
gwa.war.original	12/9/2018 4:50 PM	ORIGINAL File	57,017 KB

This is folder “webapps” of Tomcat 8.5.15

Name	Date modified	Type	Size
docs	5/5/2017 12:03 PM	File folder	
examples	5/5/2017 12:03 PM	File folder	
host-manager	5/5/2017 12:03 PM	File folder	
manager	5/5/2017 12:03 PM	File folder	
ROOT	5/5/2017 12:03 PM	File folder	
gwa.war	12/9/2018 4:51 PM	WAR File	63,532 KB

Then go to folder “bin” of Tomcat and click on “startup.bat” file

Name	Date modified	Type	Size
configtest.sh	5/5/2017 12:03 PM	SH File	2 KB
crawlarticle.dat	12/8/2018 2:06 PM	DAT File	1 KB
crawlmodel.dat	12/9/2018 4:13 PM	DAT File	1 KB
daemon.sh	5/5/2017 12:03 PM	SH File	8 KB
digest.bat	5/5/2017 12:03 PM	Windows Batch File	3 KB
digest.sh	5/5/2017 12:03 PM	SH File	2 KB
logschedule.dat	12/18/2018 11:28 ...	DAT File	1 KB
service.bat	5/5/2017 12:03 PM	Windows Batch File	7 KB
setclasspath.bat	5/5/2017 12:03 PM	Windows Batch File	4 KB
setclasspath.sh	5/5/2017 12:03 PM	SH File	4 KB
shutdown.bat	5/5/2017 12:03 PM	Windows Batch File	2 KB
shutdown.sh	5/5/2017 12:03 PM	SH File	2 KB
startup.bat	5/5/2017 12:03 PM	Windows Batch File	2 KB
startup.sh	5/5/2017 12:03 PM	SH File	2 KB
tcnative-1.dll	5/5/2017 12:03 PM	Application extens...	2,058 KB
tomcat8.exe	5/5/2017 12:03 PM	Application	108 KB
tomcat8w.exe	5/5/2017 12:03 PM	Application	108 KB
tomcat-juli.jar	5/5/2017 12:03 PM	Executable Jar File	41 KB
tomcat-native.tar.gz	5/5/2017 12:03 PM	WinRAR archive	391 KB
tool-wrapper.bat	5/5/2017 12:03 PM	Windows Batch File	4 KB
tool-wrapper.sh	5/5/2017 12:03 PM	SH File	5 KB

Web application is deployed on Tomcat standalone successfully

```

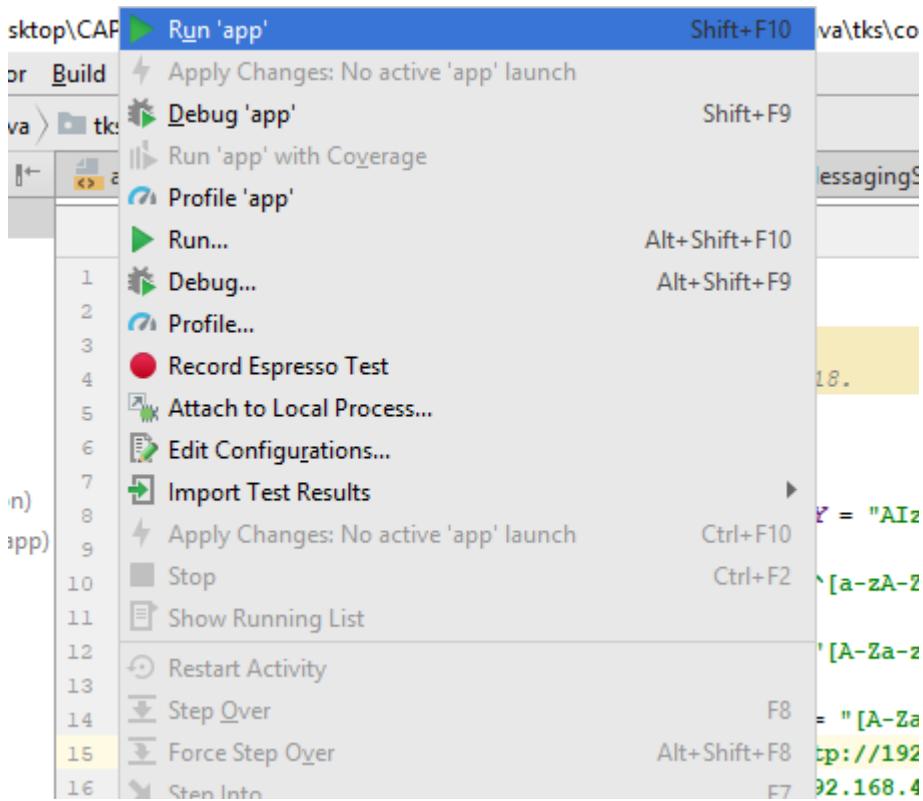
Tomcat
09-Dec-2018 17:10:16.324 INFO [localhost-startStop-1] org.apache.catalina.startup.HostConfig.deployDirectory Deploying w
eb application directory [C:\Users\Tung Hoang Ngo Minh\Documents\apache-tomcat-8.5.15\webapps\docs]
09-Dec-2018 17:10:16.402 INFO [localhost-startStop-1] org.apache.catalina.startup.HostConfig.deployDirectory Deployment
of web application directory [C:\Users\Tung Hoang Ngo Minh\Documents\apache-tomcat-8.5.15\webapps\docs] has finished in
[78] ms
09-Dec-2018 17:10:16.403 INFO [localhost-startStop-1] org.apache.catalina.startup.HostConfig.deployDirectory Deploying w
eb application directory [C:\Users\Tung Hoang Ngo Minh\Documents\apache-tomcat-8.5.15\webapps\examples]
09-Dec-2018 17:10:23.557 INFO [localhost-startStop-1] org.apache.catalina.startup.HostConfig.deployDirectory Deployment
of web application directory [C:\Users\Tung Hoang Ngo Minh\Documents\apache-tomcat-8.5.15\webapps\examples] has finished
in [7,154] ms
09-Dec-2018 17:10:23.557 INFO [localhost-startStop-1] org.apache.catalina.startup.HostConfig.deployDirectory Deploying w
eb application directory [C:\Users\Tung Hoang Ngo Minh\Documents\apache-tomcat-8.5.15\webapps\host-manager]
09-Dec-2018 17:10:23.753 INFO [localhost-startStop-1] org.apache.catalina.startup.HostConfig.deployDirectory Deployment
of web application directory [C:\Users\Tung Hoang Ngo Minh\Documents\apache-tomcat-8.5.15\webapps\host-manager] has fini
shed in [196] ms
09-Dec-2018 17:10:23.755 INFO [localhost-startStop-1] org.apache.catalina.startup.HostConfig.deployDirectory Deploying w
eb application directory [C:\Users\Tung Hoang Ngo Minh\Documents\apache-tomcat-8.5.15\webapps\manager]
09-Dec-2018 17:10:23.820 INFO [localhost-startStop-1] org.apache.catalina.startup.HostConfig.deployDirectory Deployment
of web application directory [C:\Users\Tung Hoang Ngo Minh\Documents\apache-tomcat-8.5.15\webapps\manager] has finished
in [65] ms
09-Dec-2018 17:10:23.821 INFO [localhost-startStop-1] org.apache.catalina.startup.HostConfig.deployDirectory Deploying w
eb application directory [C:\Users\Tung Hoang Ngo Minh\Documents\apache-tomcat-8.5.15\webapps\ROOT]
09-Dec-2018 17:10:23.867 INFO [localhost-startStop-1] org.apache.catalina.startup.HostConfig.deployDirectory Deployment
of web application directory [C:\Users\Tung Hoang Ngo Minh\Documents\apache-tomcat-8.5.15\webapps\ROOT] has finished in
[45] ms
09-Dec-2018 17:10:23.873 INFO [main] org.apache.coyote.AbstractProtocol.start Starting ProtocolHandler ["http-nio-8080"]

09-Dec-2018 17:10:23.890 INFO [main] org.apache.coyote.AbstractProtocol.start Starting ProtocolHandler ["ajp-nio-8009"]
09-Dec-2018 17:10:23.892 INFO [main] org.apache.catalina.startup.Catalina.start Server startup in 305977 ms

```

### 1.3. Mobile Application Deployment Process

Open Android Studio application. Click on “Run” tab, then choose “Run ‘app’”



## 2. User Guide

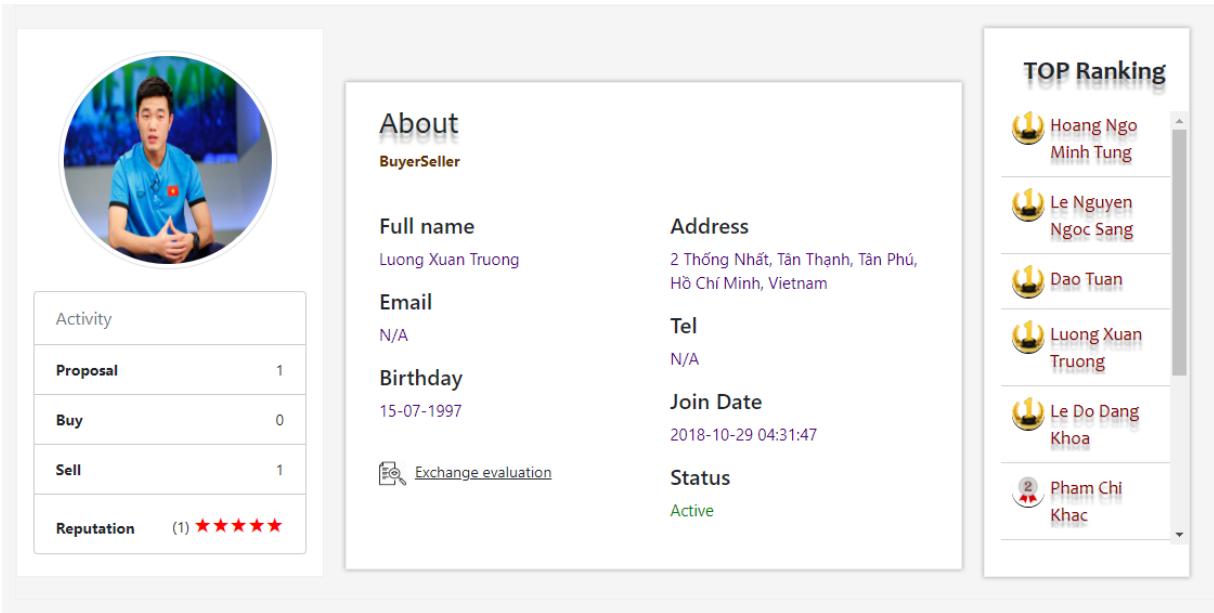
### 2.1. Web Application

#### 2.1.1. Guest, Member and BuyerSeller

##### 2.1.1.1. Get user information, edit profile, get exchange evaluation

*Login to Web Application via <http://localhost:8080/gwa/login>, then click on a username or own profile*

*This is for other user's information*



This screenshot shows a user profile page. At the top left is a circular profile picture of a man in a blue shirt. Below it is a table showing activity history:

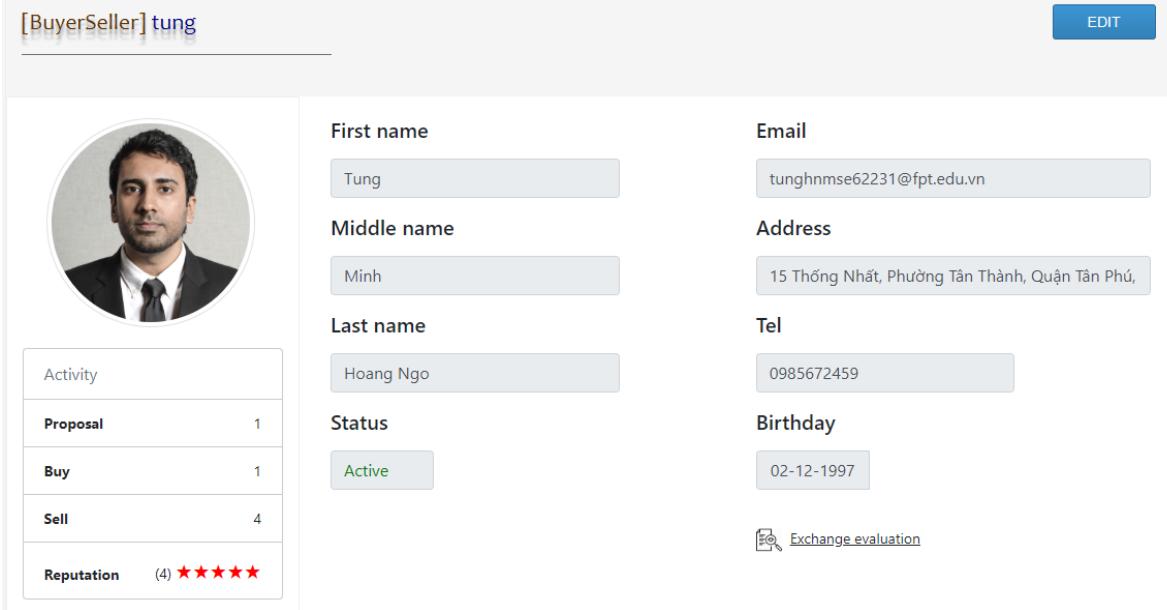
Activity	
Proposal	1
Buy	0
Sell	1
Reputation	(1) ★★★★☆

The main content area contains sections for "About" and "BuyerSeller". The "About" section includes fields for Full name (Luong Xuan Truong), Address (2 Thống Nhất, Tân Thạnh, Tân Phú, Hồ Chí Minh, Vietnam), Email (N/A), Tel (N/A), Birthday (15-07-1997), Join Date (2018-10-29 04:31:47), and Status (Active). There is also a link to "Exchange evaluation".

To the right is a "TOP Ranking" sidebar listing users:

- 1 Hoang Ngo Minh Tung
- 1 Le Nguyen Ngoc Sang
- 1 Dao Tuan
- 1 Luong Xuan Truong
- 1 Le Do Dang Khoa
- 2 Pham Chi Khac

*This is for own profile*



This screenshot shows a pop-up dialog for exchange evaluation. At the top left is a circular profile picture of a man in a suit. Below it is a table showing activity history:

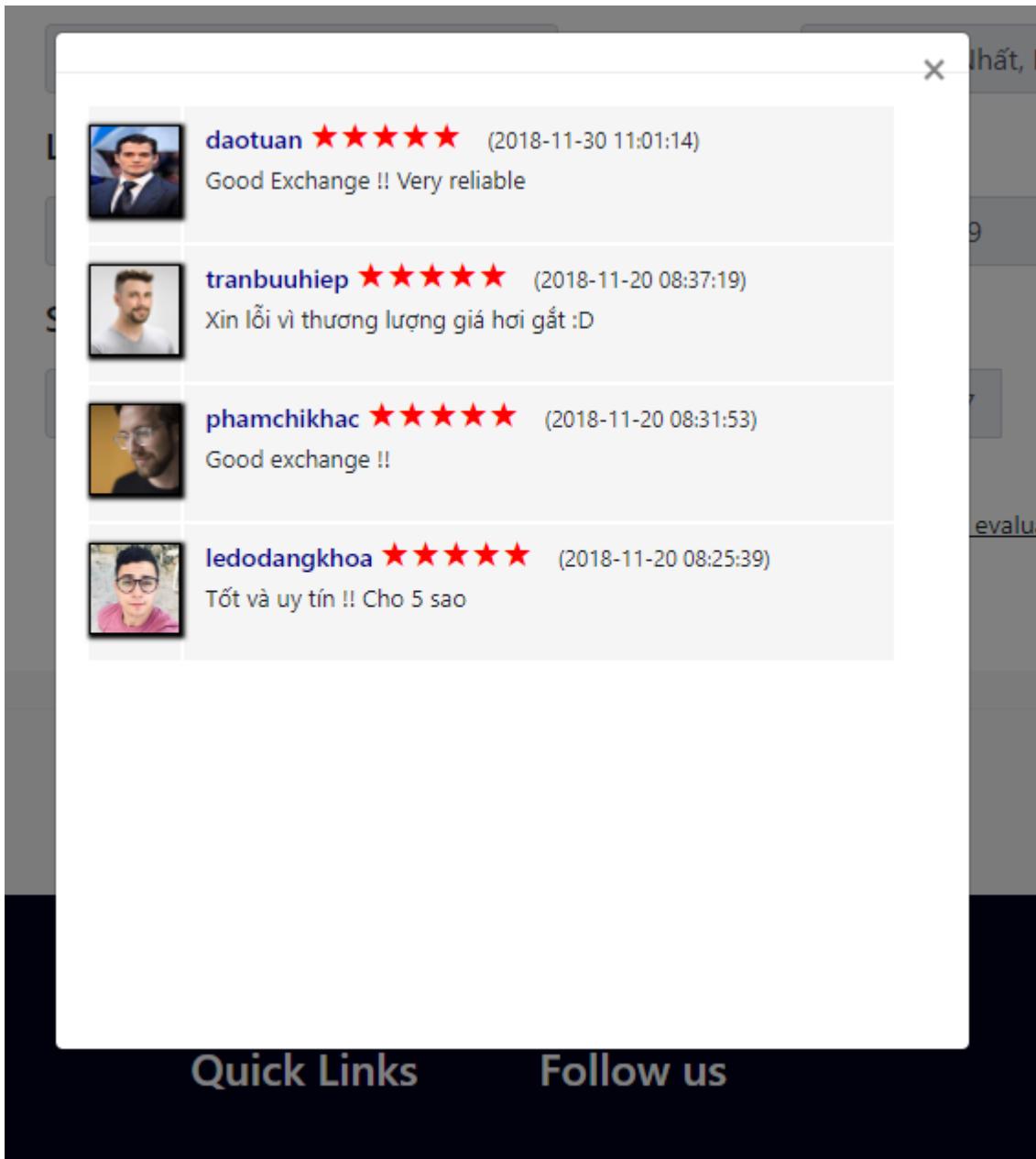
Activity	
Proposal	1
Buy	1
Sell	4
Reputation	(4) ★★★★☆

The main content area contains fields for personal information:

First name	Email
Tung	tunghnmse62231@fpt.edu.vn
Middle name	Address
Minh	15 Thống Nhất, Phường Tân Thành, Quận Tân Phú,
Last name	Tel
Hoang Ngo	0985672459
Status	Birthday
Active	02-12-1997

At the bottom is a link to "Exchange evaluation".

*This pop-up dialog show user's exchange evaluation*



#### 2.1.1.2. Get gundam model information

Click on “GUNDAM” option on the top menu, then type text in search input bar, this floating list shows instant matching models

*Choose filter options and click on magnifier icon, choose order by options*

*Hover on a record and click on “More information” button*

*This page shows model's detail information with top 5 rating model, related exchange post and related article*



CATEGORY  
OPTIONS

Series Title

- High Grade
- Love Phantom
- Master Grade
- Perfect Grade
- Real Grade
- Remaster Ver.
- Thunderbolt

MODEL INFORMATION



Name: RX-0 Unicorn Gundam 02 Banshee Titanium Finish Ver.

Manufacturer: Bandai

Scale: 1/100

Series: Master Grade

Release Date: Late Dec., 2012

Sales Price: 10,800 yen

Item Code: 2203063

Rating: ★★★★★ (16)

IMAGES

Package	Item picture

TOP FIVE

Model



Trade

 **xuantruong** ★★★★★ (2018-12-05 02:19:50)  
This gundam model is wonderful !!

 **buikhacminhquan** ★★★★★ (2018-11-05 04:24:24)  
If your going to watch Gundam, watch Universal Century Gundam, or some of the better AU series like Gundam X and Turn-A Gundam.

 **nguyenductam** ★★★★★ (2018-11-05 04:23:19)  
Well, Gundam Wing is the best Gundam series I've ever seen, because the gundams are a lot better than the gundams from Mobile Suit Gundam. The characters are really cool.

 **phamchikhac** ★★★★ (2018-11-05 04:21:54)  
I absolutely love this show so much. If there's a real Gundam Wing movie someday I will go to see it definitely and I will know the movie will be the best.

 **phanminhtri** ★★★★★ (2018-11-05 04:21:21)  
If you want thought provoking anime, look for a UC

### 2.1.1.3. Get article information

Search article by title and filters

[Home](#) [GUNDAM](#) [Articles](#) [Events](#) [Trade Market](#)

## Article Listing

[Create article](#)

### Search:

[Search](#)

**Sort:**

Newest

All category

#### NEWS

##### [Article cua tung](#)

2018/11/29 | Author: tung

[article cua tung ne](#)

#### SHOPPING GUIDES

##### [Gunpla Pre-orders and New Releases for December-January 2018](#)

2018/11/26 | Author: khanh

This article is crawled by system !!!

#### TUTORIALS

##### [How to Build a Mesa Diorama For Gunpla Display](#)

2018/11/26 | Author: khanh

This is search article page

Create article by inputting all fields and submit

**Create article**

Your article will be reviewed by the administrator before being posted on the website!

**Article title:**

**Category**  News ▼

**Description**

**Thumbnail image**   


**Editor Tools**

X                                                                                                                                                                                                                                                                                                                                                                                                                                            <img alt

## Create article page

#### 2.1.1.4. Get event information

The screenshot shows a web application interface for event listing. At the top, there are navigation links: GUNDAM, Articles, Events, and Trade Market. Below that, a red box highlights the 'Event Listing' tab. To its right are links for 'Registered events' and 'Write an event proposal!'. A search bar with placeholder 'Search event' and a blue 'Search' button is followed by a link 'Show nearby events'. A dropdown menu for sorting events is set to 'Newest' on the left and 'Ongoing/future events' on the right. The main content area displays three event cards:

- Event moi**  
From: 2020-12-12 to 2030-12-12  
Ticket price: 100.000 VND  
sasas
- GUNPLA BUILDERS WORLD CUP 2018 REGULATION 33**  
From: 2018-12-12 to 2019-11-11  
Ticket price: 10.000 VND  
Quốc gia và vùng lãnh thổ tổ chức 16 quốc gia và vùng lãnh thổ Nhật Bản, Trung Quốc, Hàn Quốc, Đài Loan, Hồng Kông, Thái Lan, Singapore, Malaysia, Philippines, Indonesia, Úc, Ý, Pháp, Việt Nam, Mỹ, Canada. Thời gian diễn ra cuộc thi GBWC 2017 sẽ đư
- GUNPLA BUILDERS WORLD CUP 2018 REGULATION 34**  
From: 2018-12-12 to 2019-11-11  
Ticket price: 10.000 VND

*Search event page*

Register event by entering your credit card number and tickets amount

The screenshot shows the Gunpla World website. At the top, there is a red header with the site name 'Gunpla World' and a user profile for 'Nguyen Bao Khanh'. The main content area features a large banner for the 'GUNPLA BUILDERS WORLD CUP' with details about the event's location and date. Below the banner, a modal window titled 'Register information' is open. It contains fields for Name (khanh), Date (2018-12-09), Remaining tickets (100), Your tickets (Ticket amount dropdown), and Credit card number (Card number input field). A blue 'REGISTER' button is at the bottom right of the modal.

*Register event page*

#### 2.1.2. Administrator

Login to Web Application via <http://localhost:8080/gwa/login> with role “Admin”

### 2.1.2.1. Manipulate GUNDAM model

*Manage gundam models*

Code	Name	Product Series	Series Title	Manufacturer	Date/Time	Status	Price
2303444	RX-0 [N] Unicorn Gundam 02 Banshee Norn	1/60	Perfect Grade	Bandai	2018-10-31 23:40:12	Available	19,800 yen
2103482	Strike Freedom Gundam	1/60	Perfect Grade	Bandai	2018-10-31 23:29:10	Available	22,500 yen
0071690	MSZ-010 ZZ Gundam	1/100	Master Grade	Bandai	2018-10-30 02:11:50	Available	3,600 yen
0071692	RGM-79Q GM Quel	1/100	Master Grade	Bandai	2018-10-30 02:11:47	Available	2,700 yen
0071691	RGM-79N GM Custom	1/100	Master Grade	Bandai	2018-10-30 02:11:45	Available	2,700 yen
0074440	MS-09R Rick Dom	1/100	Master Grade	Bandai	2018-10-30 02:11:42	Available	3,600 yen

*Create new gundam model*

**Create Model**

**Model Information**

- Code:
- Name:
- Product Series:
- Series Title:
- Manufacturer:
- Price:
- Status:

**Description**

Enter model's description

**Model's Images**

*Edit gundam model*

Navigation menu (partial view):

- S
- Models
- De
- it
- re post
- chedule
- arrant
- ching
- formation

**Edit Model 2303444**

**Model Information**

Code: 2303444

Name: RX-0 [N] Unicorn Gundam 02 Banshi

Product Series: 1/60

Series Title: Perfect Grade

Manufacturer: Bandai

Price: 19,800 yen

Status: Available

**Description**

<div>This item is limited to 1 per household.<br></div>Sunrise</div><br><br>[Kuroki lion] final battle form of Banshi-I - Banshi-I-Norns appeared !! to PG<br><br><br>[Mobile Suit Gundam UC] than Banshi-I-Norns appeared become a Gundam highest grade PG. Fully reproduce the transformation from Unicorn mode to Destroy mode. Various equipment comes as the final decisive battle form.<br><br><br><br> Armed Armor DE as a new accessory armed, comes with parts to reproduce the revolving launcher./ Existing beam Maximum new parts added to shield both form

**Model's Images**

+ Add files... Delete

	10334864p.jpg	N/A	Package	Delete
	10334864a.jpg	N/A	Item picture	Delete
	10334864a2.jpg	N/A	Item picture	Delete

### 2.1.2.2. Manipulate user

#### Manage user

Navigation menu (partial view):

- AdminGWA
- Nguyen Bao Khanh
- Online
- ACCOUNT

**ACCOUNT** Search account

Latest Accounts

Search...

Order by: Register Date Descending

First Prev 1 Next Last

ID	Fullname	Username	Email	Date Time	Status	Address
31	Doan Van Hau	vanhau	doanvanhau@fpt.edu.vn	2018-12-05 11:06:14	Active	178 Nguyễn Văn Trỗi, Phường 8, Phù Nhuân, Hồ Chí Minh, Vietnam
30	Luong Xuan Truong	xuantruong	luongxuantruong@gmail.com	2018-12-05 02:12:53	Active	2 Tống Nhất, Tân Thành, Tân Phú, Hồ Chí Minh, Vietnam
29	Bui Khac Minh Quan	bukhacminhquan	quankbm@fpt.edu.vn	2018-11-05 00:45:01	Active	19 Phan Dang Luu, phuong 2, quan Bình Tân
28	Nguyen Duc Tam	nguyenductam	tamnd@fpt.edu.vn	2018-11-05 00:43:33	Active	156 Phan Xich Long, phuong 12, quan Bình Thạnh
27	Pham Chi Khac	phamchikhac	khacpc@fpt.edu.vn	2018-11-05 00:42:14	Active	51 Nguyễn Cửu Đàm, phuong Tân Thành, quan Tân Phú
26	Phan Minh Tri	phanminhtri	tripm@fpt.edu.vn	2018-11-05 00:40:40	Active	14 Công Hòa, phuong 5, quan Tân Bình
25	Tran Trang Nhat	trantrangnhat	nhat@fpt.edu.vn	2018-11-05 00:39:58	Active	560 Thủ Đức, Phường Thủ Đức

### 2.1.2.3. Manipulate crawling

The screenshot shows a web-based application for managing crawl models. At the top, it says "GUNDAM Crawl model". Below that is a navigation bar with "Home > Crawl Mode". A large button labeled "CRAWL" is visible. The main content is a table titled "Crawl Records" with columns: ID, Date Time, Crawl Records, New Records, and Status. The table contains 8 rows of data.

ID	Date Time	Crawl Records	New Records	Status
7	2018-12-04 19:54:42	62	59	Done
6	2018-10-30 01:49:02	253	253	Done
5	2018-10-25 11:12:11	253	0	Done
4	2018-10-25 00:35:25	253	249	Done
3	2018-10-25 00:17:47	34	34	Done
2	2018-10-23 09:26:12	721	69	Done
1	2018-10-23 04:11:21	894	894	Done

### 2.1.2.4. Manipulate schedule

The screenshot shows a "SCHEDULE" section with three separate schedule entries. Each entry includes a "STOP" button.

Gundam Crawl Schedule (Is Running) Cycle: 258 hours			
10	17	59	37
Days	Hours	Minutes	Seconds
Trade Market Schedule (Is Running) Cycle: 480 hours			
19	23	59	45
Days	Hours	Minutes	Seconds
Event Update Schedule (Is Running) Cycle: 25 hours			
1	0	59	55
Days	Hours	Minutes	Seconds

## 2.1.2.5. Manipulate article

Article Manage article

Latest Articles

Search...

Newest	All category	Approved				
ID	Title	Author	Date posted	Category	Approval status	Action
512	Gunpla Pre-orders and New Releases for July 2015	khanh	2018/12/01	Shopping Guides	Approved	<a href="#">Link / Edit</a>
513	Gunpla Pre-orders and New Releases for June 2015	khanh	2018/12/01	Shopping Guides	Approved	<a href="#">Link / Edit</a>
514	Gunpla Pre-orders and New Releases for May 2015	khanh	2018/12/01	Shopping Guides	Approved	<a href="#">Link / Edit</a>
515	Ask Gunpla 101: Too young for Gunpla?	khanh	2018/12/01	Advice	Approved	<a href="#">Link / Edit</a>
516	Gunpla Pre-orders and New Releases for April 2015	khanh	2018/12/01	Shopping Guides	Approved	<a href="#">Link / Edit</a>
517	Gunpla Pre-orders and New Releases for March 2015	khanh	2018/12/01	Shopping Guides	Approved	<a href="#">Link / Edit</a>
518	Vintage Gunpla: How They Looked in the '80s	khanh	2018/12/01	Lifestyle	Approved	<a href="#">Link / Edit</a>
519	Gunpla Pre-orders and New Releases for February 2015	khanh	2018/12/01	Shopping Guides	Approved	<a href="#">Link / Edit</a>
520	Some of Our Favorite Gunpla 101 Reader Builds, Part 2	khanh	2018/12/01	Lifestyle	Approved	<a href="#">Link / Edit</a>
521	How we take photos of our Gunpla	khanh	2018/12/01	Tutorials	Approved	<a href="#">Link / Edit</a>

[«](#) [‹](#) [1](#) [›](#) [»](#)

## Manipulate article

Article Create article

Create a new article

**Title**

**Description**

**Thumbnail image**

**Category**

**Status**

**Content**

Create article by inputting all fields and submit

Article Edit article

Edit article

**Title**

new article mang  
Last edited on 2018/12/05 by khanh  
Posted on: 2018/12/05 / Author: khanh

**Description**

asdad

**Thumbnail image**

**Category**

News

**Status**

Pending

**Content**

*(Rich text editor interface with various toolbar icons)*

//pending

Edit article by inputting all fields and submit

### 2.1.2.6. Manipulate event

Event Manage Events

Events

Search...						Search
Newest	Ongoing/future events	Start date	End date	Ticket price	Status	Action
Event moi	sasas	2020-12-12	2030-12-12	100,000 VND	Active	<a href="#">Link / Edit</a>
GUNPLA BUILDERS WORLD CUP 2018 REGULATION 33	Quốc gia và vùng lãnh thổ tổ chức 16 quốc gia và vùng lãnh thổ Nhật Bản, Trung Quốc, Hàn Quốc, Đài Loan, Hồng Kông, Thái Lan, Singapore, Malaysia, Philippines, Indonesia, Úc, Ý, Pháp, Việt Nam, Mỹ, Canada. Thời gian diễn ra cuộc thi GBWC 2017 sẽ đư	2018-12-12	2019-11-11	10,000 VND	Active	<a href="#">Link / Edit</a>
GUNPLA BUILDERS WORLD CUP 2018 REGULATION 34	Quốc gia và vùng lãnh thổ tổ chức 16 quốc gia và vùng lãnh thổ Nhật Bản, Trung Quốc, Hàn Quốc, Đài Loan, Hồng Kông, Thái Lan, Singapore, Malaysia, Philippines, Indonesia, Úc, Ý, Pháp, Việt Nam, Mỹ, Canada. Th...	2018-12-12	2019-11-11	10,000 VND	Active	<a href="#">Link / Edit</a>
GUNPLA BUILDERS WORLD CUP 2018 REGULATION 37	Quốc gia và vùng lãnh thổ tổ chức 16 quốc gia và vùng lãnh thổ Nhật Bản, Trung Quốc, Hàn Quốc, Đài Loan, Hồng Kông, Thái Lan, Singapore, Malaysia, Philippines, Indonesia, Úc, Ý, Pháp, Việt Nam, Mỹ, Canada. Th...	2018-12-12	2019-11-11	10,000 VND	Active	<a href="#">Link / Edit</a>
GUNPLA BUILDERS WORLD CUP 2018 REGULATION 35.2	Quốc gia và vùng lãnh thổ tổ chức 16 quốc gia và vùng lãnh thổ Nhật Bản, Trung Quốc, Hàn Quốc, Đài Loan, Hồng Kông, Thái Lan, Singapore, Malaysia, Philippines, Indonesia, Úc, Ý, Pháp, Việt Nam, Mỹ, Canada. Th!!!	2018-12-05	2019-11-12	10,000 VND	Active	<a href="#">Link / Edit</a>
GUNPLA BUILDERS WORLD CUP 2018 REGULATION 36	Quốc gia và vùng lãnh thổ tổ chức 16 quốc gia và vùng lãnh thổ Nhật Bản, Trung Quốc, Hàn Quốc, Đài Loan, Hồng Kông, Thái Lan, Singapore, Malaysia, Philippines, Indonesia, Úc, Ý, Pháp, Việt Nam, Mỹ, Canada. Th!!!	2018-11-29	2019-11-11	10,000 VND	Active	<a href="#">Link / Edit</a>
Title	Description	Start date	End date	Ticket price	Status	Action

Manipulate events

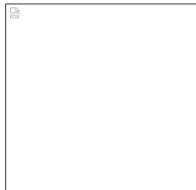
Event [Create Event](#)

Create a new event

**Title**

**Location**  
 Enter your address

**Description**

**Thumbnail image**  


**Start date**  
 mm / dd / yyyy  
- + - +

Create event by inputting all fields and submit

### 2.1.2.7. Manipulate proposal

Proposal [Manage Proposal](#)

[Home](#) > Manage proposal

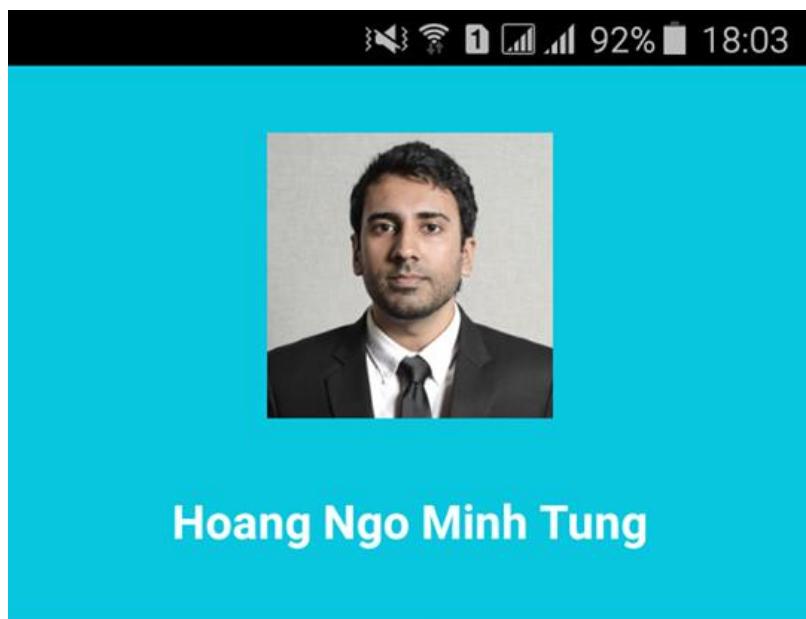
Events

Title	User	Action
first proposal ever	tung	<a href="#">Detail</a>
send proposal	khanh	<a href="#">Detail</a>
third proposal	khanh	<a href="#">Detail</a>
event proposal lmao	banhthibuo1	<a href="#">Detail</a>
Title	User	Action

Manipulate proposal

## 2.2. Mobile Application

### 2.2.1. Get user information



 tung

 02-12-1997

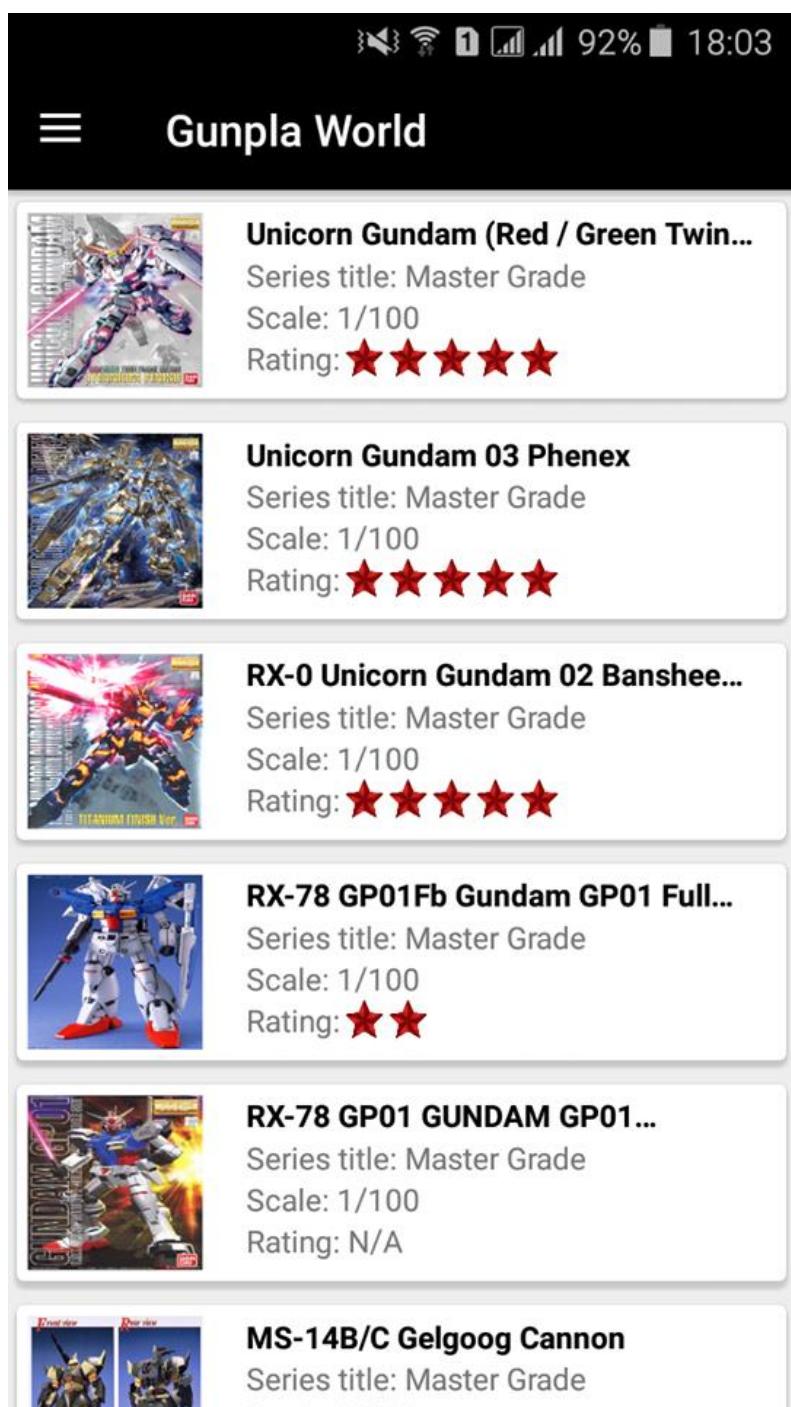
 tunghnmse62231@fpt.edu.vn

 1

 4

## 2.2.2. Get model information



The screenshot shows a mobile application interface for "Gunpla World". At the top, there is a black header bar with a signal strength icon, a battery level of 92%, and the time 18:03. Below the header is a navigation bar with three horizontal lines and the text "Gunpla World". The main content area displays six model kit entries, each in its own card:

- Unicorn Gundam (Red / Green Twin...)**  
Series title: Master Grade  
Scale: 1/100  
Rating: ★★★★★
- Unicorn Gundam 03 Phenex**  
Series title: Master Grade  
Scale: 1/100  
Rating: ★★★★★
- RX-0 Unicorn Gundam 02 Banshee...**  
Series title: Master Grade  
Scale: 1/100  
Rating: ★★★★★
- RX-78 GP01Fb Gundam GP01 Full...**  
Series title: Master Grade  
Scale: 1/100  
Rating: ★★
- RX-78 GP01 GUNDAM GP01...**  
Series title: Master Grade  
Scale: 1/100  
Rating: N/A
- MS-14B/C Gelgoog Cannon**  
Series title: Master Grade

Each card includes a small thumbnail image of the model kit. The last card, "MS-14B/C Gelgoog Cannon", has two thumbnail images labeled "Front view" and "Rear view".