

# FPT UNIVERSITY

## Capstone Project Document

### Gunpla World Application

<b>Group 15</b>	
<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

## CAPSTONE PROJECT REGISTER

Class:                      Duration time: from ...../20.... To ...../20.....

(\*) Profession: <Software Engineer>                      Specialty:    <ES> ☐                      <IS> ☒

(\*) Kinds of person make registers:                      Lecturer ☒                      Students ☐

### 1. Register information for supervisor (if have)

	Full name	Phone	E-Mail	Title
Supervisor 1	Phạm Công Thành		thanhpc3@fe.edu.vn	Mr.

### 2. Register information for students (if have)

	Full name	Student code	Phone	E-mail	Role in Group
Student 1	Hoàng Ngô Minh Tùng	SE62231	0981875113	TungHNMS E62231@fpt .edu.vn	Leader
Student 2	Nguyễn Bảo Khánh	SE61978	0768275251	KhanhNBS E61978@fpt .edu.vn	Member
Student 3	Lê Nguyễn Ngọc Sang	SE61647	0343777228	SangLNNS E61647@fpt .edu.vn	Member

### 3. Register content of Capstone Project

(\*) 3.1. Capstone Project name:

English: Gunpla World Application

Vietnamese: Thế giới Gunpla

Abbreviation: GWA

Building a web application provides following services:

- Gunpla information system and classification
- Information about events, contests, awards of gunpla.
- Information about gunpla assembly: method, sharing experience...

- Information about paint, painting method, sharing experience...
- Information and sharing experience how to create your own gunpla.
- Search, exchange gunpla by location. (using mobile).

(\*) 3.2. Main proposal content (including result and product)

a) Theory and practice (document):

- Student should apply the software development process and the UML
- Software artifacts include User Requirement, Software Requirement Specification, Architecture Design, Detail Design, System Implementation and Testing Document, Installation Guide, sources code, and deployable software packages
- 3 tiers should be applied
- Server side technique:
  - o Database design, OOA, OOD, OOP, MVC, Java technology, RESTful API ...
- Client side technique
  - o HTML5, CSS, JavaScript, JQuery, Ajax, Androids ...
- Communication technique
  - o Exchange information and transfer data effectively in networks ...

b) Program:

- Admin/Manager:
  - o Add, delete, adjust information about gunpla system.
  - o Approve proposal to create new events, contests, prizes from other users, receive suggestions, and have reasonable rewards.
  - o Manage feedback/ evaluation from users (users who attend events or use the gunpla exchange service).
  - o Manage event calendar, contests to arrange rational calendar, avoid duplication.
  - o Manage the fee collection for providing services to users.
- Users:
  - o Users registration.
  - o Events, contests registration.
  - o Proposals to open events, contests, awards.
  - o Search for other users who need to exchange gunpla.
  - o Evaluate events, contests.
- Report:
  - o Events, contest with good feedback (by time).
  - o Users who win prizes in contests (by time).

c) Other products:

- The system automatically searches for users who need to exchange gunpla within the required or nearest range.
- The system automatically searches for gunpla events within the required or nearest range.

4. Other comment (propose all relative thing if have)

N/A

HCM, date ..... /20 .....

**Supervisor (If have)**

*(Sign and full name)*

**On behalf of Registers**

*(Sign and full name)*

# Table of Contents

Table of Contents.....	5
List of Tables .....	6
List of Figures .....	6
1. Introduction.....	8
1. Project Information .....	8
2. Introduction.....	8
3. Current Situation.....	9
4. Problem Definition .....	9
5. Proposed Solution .....	9
5.1 Feature functions.....	9
5.2 Advantages and disadvantages .....	10
6. Functional Requirements .....	10
7. Role and Responsibility .....	12
2. Software Process Model .....	12
3. Conceptual Diagram .....	14
4. Use-case Diagram .....	16
4.1. System Overview Usecase .....	16
4.2. Use-case Specification .....	17
4.2.1. < Guest, Member > Search model Usecase .....	17
4.2.2. < Guest, Member > Search exchange Usecase .....	21
4.2.3. < Member > Register event Usecase .....	25
4.2.4. < Member > Request order Usecase .....	27
4.2.5. < Member > Create article Usecase .....	31
4.2.6. < Administrator > Crawl model Usecase .....	33
4.2.7. < Administrator > Create new event Usecase .....	36
4.2.8. < Administrator > Start/Stop schedule Usecase.....	39
5. Architectural Diagram .....	42
6. Component Diagram.....	43
7. Class Diagram.....	45
8. Entity Relationship Diagram .....	49
9. Interaction Diagram .....	50
9.1. Sequence Diagram .....	50
9.1.1. Search Trade Post .....	51

9.1.2. Request order to a trade post.....	52
9.1.3. Register event .....	53
9.2. Activity Diagram .....	54
9.2.1. Search Gundam Model .....	54
9.2.2. Manage order .....	55
10. Physical Diagram .....	56
11. Algorithms .....	58
11.1. As-you-type search with instant results and pagination with sessionStorage .....	58
11.1.1. Definition .....	58
11.1.2. Define problems.....	58
11.1.3. Solution.....	58
11.1.4. Flow chart .....	60
11.2. MD5 hash function .....	61
11.2.1. Definition .....	61
11.2.2. Define problems.....	61
11.2.3. Solution.....	61
11.3. Haversine formula.....	62
11.3.1. Definition .....	62
11.3.2. Define problems.....	62
11.3.3. Solution.....	62
12. Future Plan.....	63

## List of Tables

Table 1 - Role and Responsibility.....	12
Table 2 - Conceptual Dictionary .....	15
Table 3 - Component Dictionary .....	45
Table 4 - Class Dictionary .....	48
Table 5 - ERD Dictionary .....	50
Table 6 - Physical Diagram Dictionary .....	57

## List of Figures

Figure 1 - Software Process Model.....	13
Figure 2 - Conceptual Diagram .....	14

Figure 3 - System Overview Usecase .....	16
Figure 4 - < Guest, Member > Search model Usecase .....	17
Figure 5 - < Guest, Member > Search exchange Usecase .....	21
Figure 6 - < Member > Register event Usecase.....	25
Figure 7 - < Member > Request order Usecase .....	27
Figure 8 - < Member > Create article Usecase .....	31
Figure 9 - < Administrator > Crawl model Usecase .....	33
Figure 10 - < Administrator > Create new event Usecase .....	36
Figure 11 - < Administrator > Start/Stop schedule Usecase .....	39
Figure 12 - Architectural Diagram.....	42
Figure 13 - Component Diagram .....	44
Figure 14 - Class Diagram .....	46
Figure 15 - Entity Relationship Diagram.....	49
Figure 16 - < Sequence Diagram > Search Trade Post.....	51
Figure 17 - < Sequence Diagram > Request order.....	52
Figure 18 - < Sequence Diagram > Register event .....	53
Figure 19 - < Activity Diagram > Search Gundam Model .....	54
Figure 20 - < Activity Diagram> Manage order .....	55
Figure 21 - Physical Diagram .....	56
Figure 22 - As-you-type search work flow .....	60

# 1. Introduction

## 1. Project Information

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

## 2. Introduction

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

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

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

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



### **3. Current Situation**

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

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

### **4. Problem Definition**

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

### **5. Proposed Solution**

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

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

#### **5.1 Feature functions**

Web/Mobile Application:

- 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

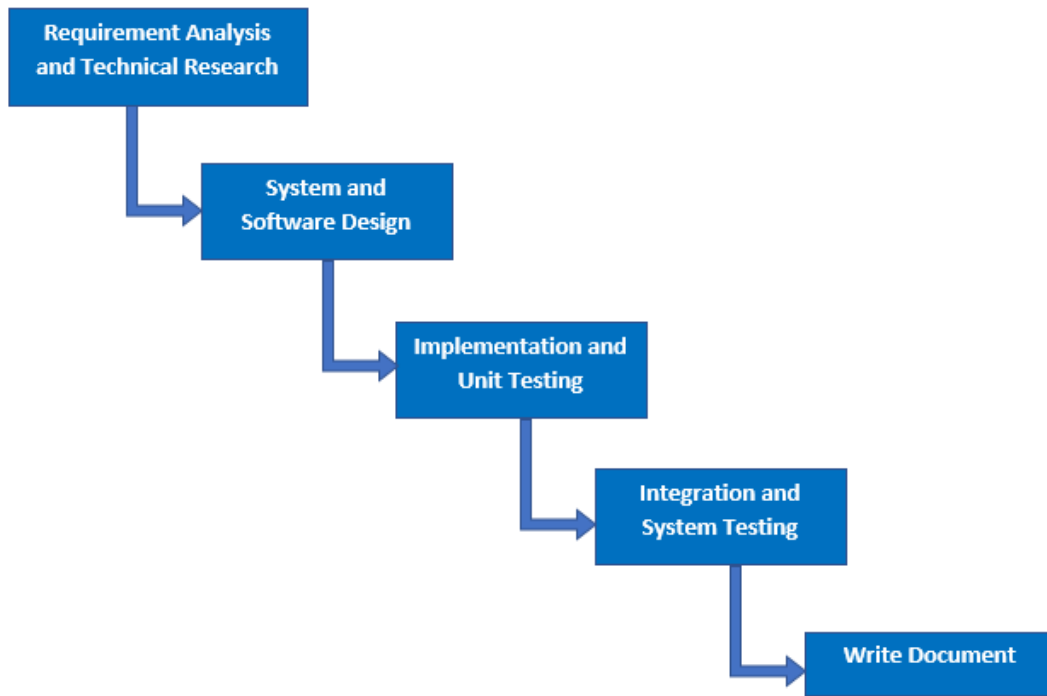
No	Full Name	Role	Position	Contact
1	Phạm Công Thành	Project Manager	Supervisor	thanhpc3@fe.edu.vn
2	Hoàng Ngô Minh Tùng	Developer	Leader	tunghnmse62231@fpt.edu.vn
3	Nguyễn Bảo Khánh	Developer	Member	khanhnbse61978@fpt.edu.vn
4	Lê Nguyễn Ngọc Sang	Developer	Member	sanglnnse61647@fpt.edu.vn

Table 1 - Role and Responsibility

## 2. 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**

### 3. Conceptual Diagram

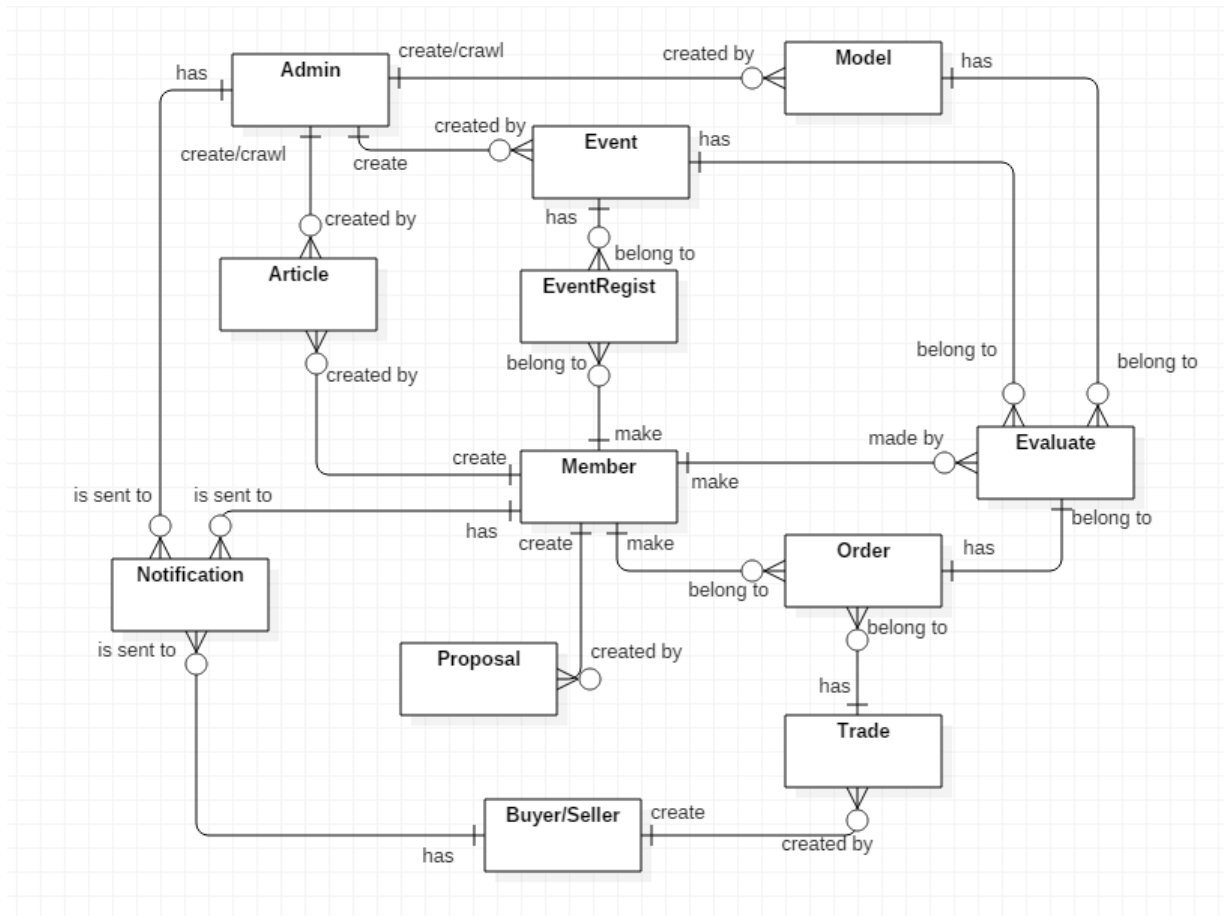


Figure 2 - Conceptual Diagram

### 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

**Table 2 - Conceptual Dictionary**

## 4. Use-case Diagram

### 4.1. System Overview Usecase

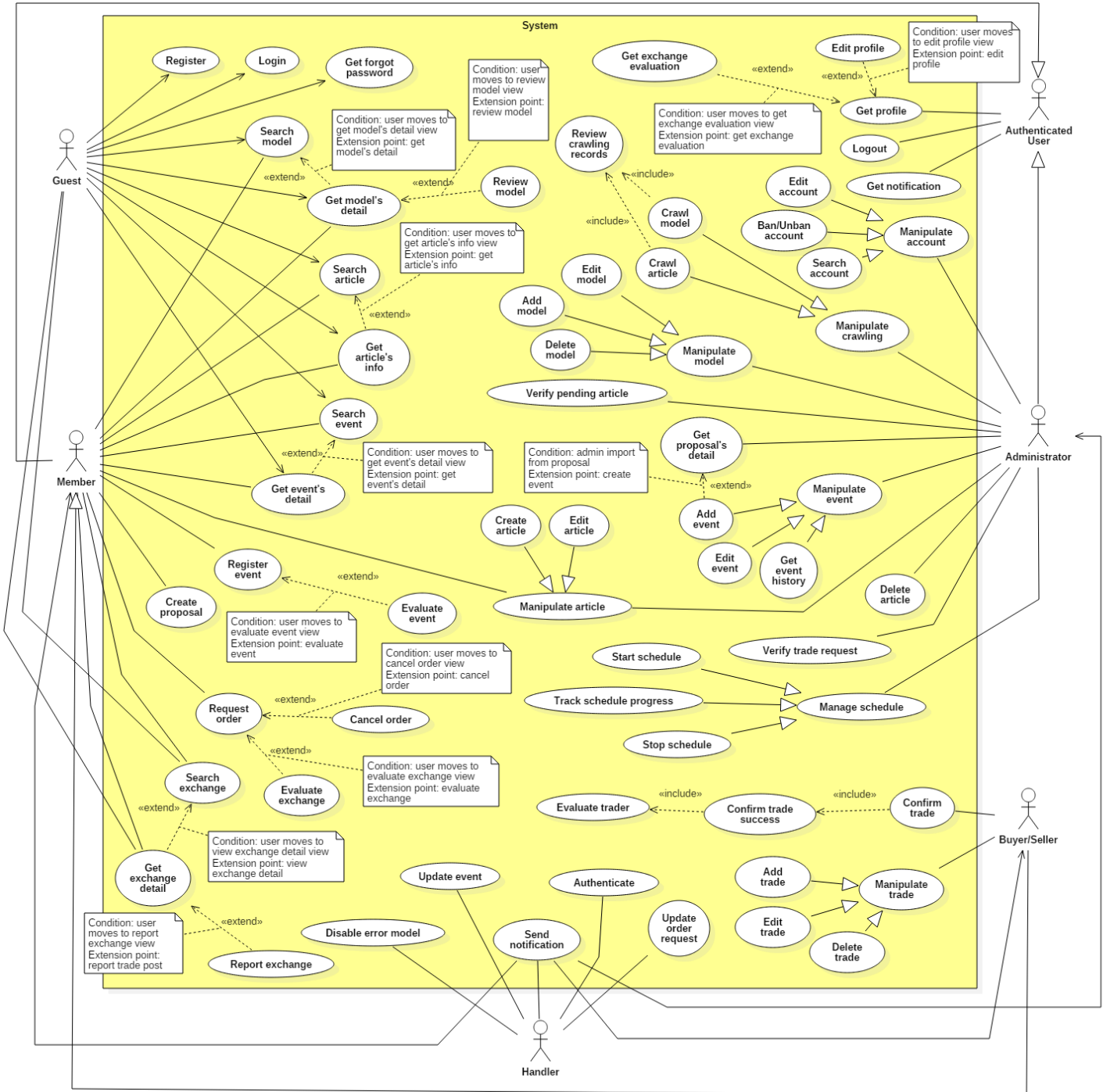


Figure 3 - System Overview Usecase



## 4.2. Use-case Specification

### 4.2.1. < Guest, Member > Search model Usecase

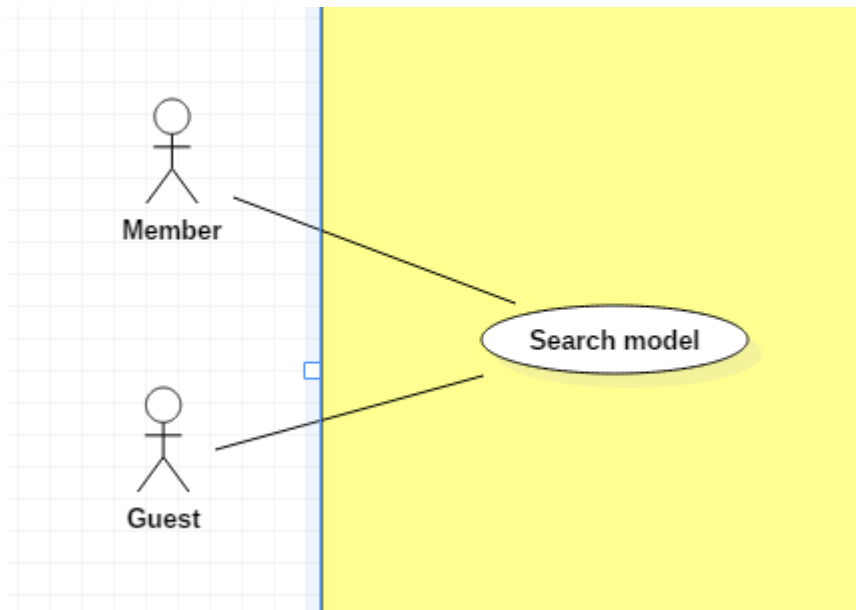


Figure 4 - < Guest, Member > Search model Usecase

USE CASE – UC _Guest,Member_01			
Usecase No.	Guest,Member_01	Usecase Version	1.0
Usecase Name	Search model		
Author	TungHNM		
Date	03/10/2018	Priority	Normal
<b>Actor:</b> <ul style="list-style-type: none"><li>- Guest, Member.</li></ul> <b>Summary:</b> <ul style="list-style-type: none"><li>- This use case allows guest, member to search model.</li></ul> <b>Goal:</b> <ul style="list-style-type: none"><li>- Make it more convenient for user to seek for gunpla information.</li></ul> <b>Triggers:</b>			

- User sends search model command to the system.

**Preconditions:**

- The search model page is displayed.

**Post Conditions:**

- **Success:** A list of matching records are shown with pagination and orderBy filter.
- **Fail:** Show error message.

**Main Success Scenario:**

Step	Actor Action	System Response
1	User 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	User clicks on search navbar	A text input is displayed with border blue and placeholder: Please input keyword
3	User 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	User 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	User hovers on a record	Brief detail information will be displayed on the view
6	User clicks on “More information” button on each records	System redirects to the clicked gundam model’s detail page
7	User chooses an option in OrderBy dropdown list	System will sort the current matching records by OrderBy filter’s value (price, createDate and rating)

**Alternative Scenario:**

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

Step	Actor Action	System Response
1	User clicks on an item in instant results list	System redirects to the clicked gundam model's detail page

**Alternative Scenario 2: No matching records**

Step	Actor Action	System Response
1	User 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

#### 4.2.2. < Guest, Member > Search exchange Usecase

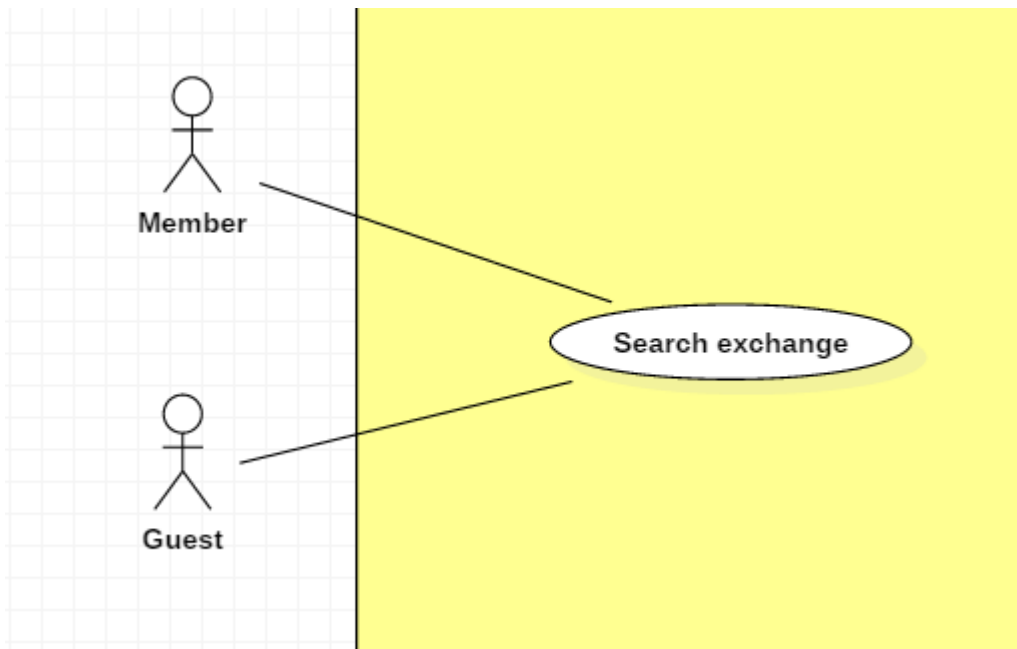


Figure 5 - < Guest, Member > Search exchange Usecase

USE CASE – UC _Guest,Member_02			
Usecase No.	Guest,Member_02	Usecase Version	1.0
Usecase Name	Search exchange		
Author	SangLNN		
Date	26/09/2018	Priority	Normal
<b>Actor:</b> <ul style="list-style-type: none"><li>- Guest, Member.</li></ul> <b>Summary:</b> <ul style="list-style-type: none"><li>- This use case allows guest and member to search exchange by keyword (title,brand,model) or location and range.</li></ul> <b>Goal:</b> <ul style="list-style-type: none"><li>- System create view to show search result.</li></ul> <b>Triggers:</b>			

- 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:**

Step	Actor Action	System Response
1	User clicks 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	User types keyword and press enter to send search command.	<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**

Step	Actor Action	System Response
1	User clicks on “more” button.	System will show additional search form with following labels and fields: <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	User fills out the form.	
3	User types keyword and press enter to send search command.	[Exception 1] - System create a view show all result. - Filter [Alternative Scenario 2] - Group by [Alternative Scenario 3]

**Alternative Scenario 2 : Filter Scenario.**

Step	Actor Action	System Response
1	User selects on filter option send filter command.	System will list trade post sorting by user selected option.

**Alternative Scenario 3 : Group by Scenario.**

Step	Actor Action	System Response
1	User selects 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.



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

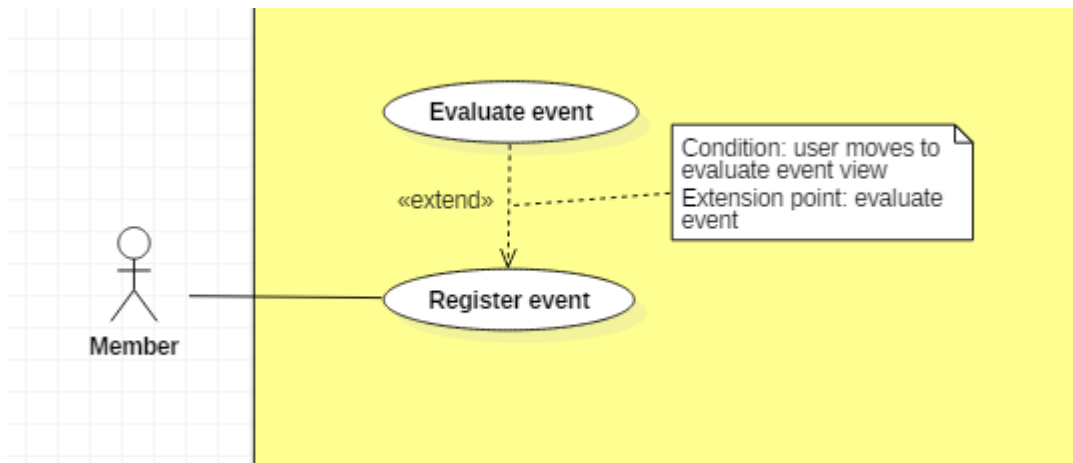


Figure 6 - < Member > Register event Usecase

USE CASE – UC_Member_03			
Usecase No.	Member_03	Usecase Version	1.0
Usecase Name	Register event		
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 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></ul>			

- **Fail:** Show error message.

**Main Success Scenario:**

Step	Actor Action	System Response
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

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

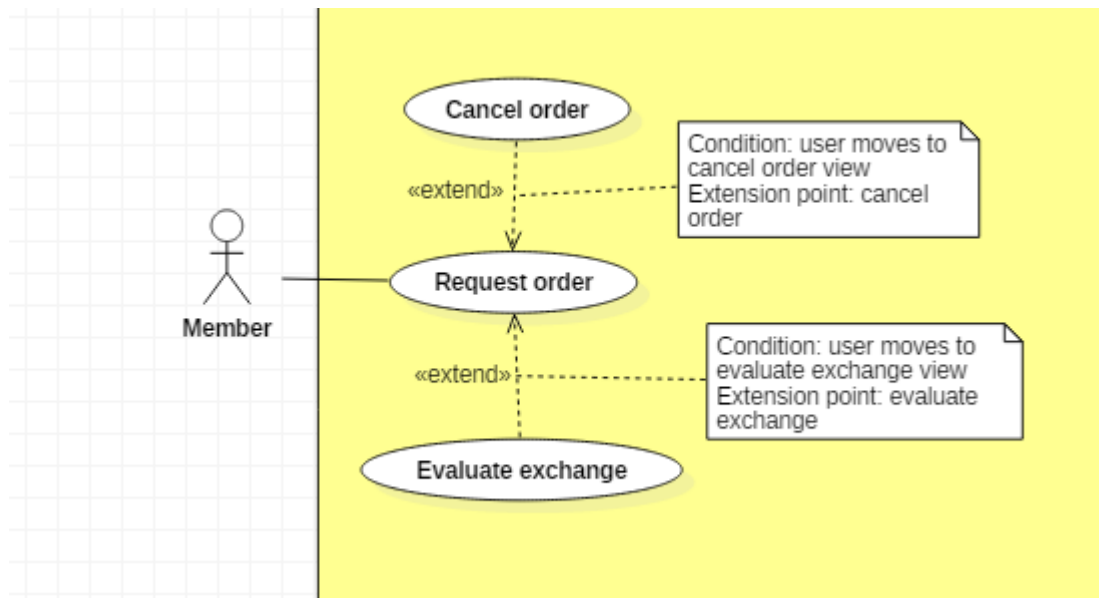


Figure 7 - < Member > Request order Usecase

USE CASE – UC_Member_04			
Usecase No.	Member_04	Usecase Version	1.0
Usecase Name	Request order		
Author	SangLNN		
Date	26/09/2018	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 order a trade.</li></ul> <b>Goal:</b> <ul style="list-style-type: none"><li>- User can get request order to trade post owner and waiting them to accept request.</li></ul> <b>Triggers:</b> <ul style="list-style-type: none"><li>- User sends order trade command with trade post id to the system.</li></ul>			

**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><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>

		System save order data to database and send notification to buyer/seller.
--	--	---

**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, required, length 3 – 50.</li> <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 autofill to address field.</p>									
<b>Exceptions:</b> <table> <tr> <th>No</th><th>Cause</th><th>System Response</th></tr> <tr> <td>1</td><td>User input quantity is not validate.</td><td>System shows error message to ask member enters validate quantity.</td></tr> <tr> <td>2</td><td>User do not allows location permission.</td><td>System shows error message to ask member provide required permission.</td></tr> </table>			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.
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.									
<b>Relationships:</b> N/A <b>Business Rules:</b> <ul style="list-style-type: none"> <li>- System auto fill billing information from profile database.</li> <li>- Order quantity must be lower or equal stock quantity.</li> <li>- Order id is automatic initialized by the system.</li> </ul>											

#### 4.2.5. < Member > Create article Usecase

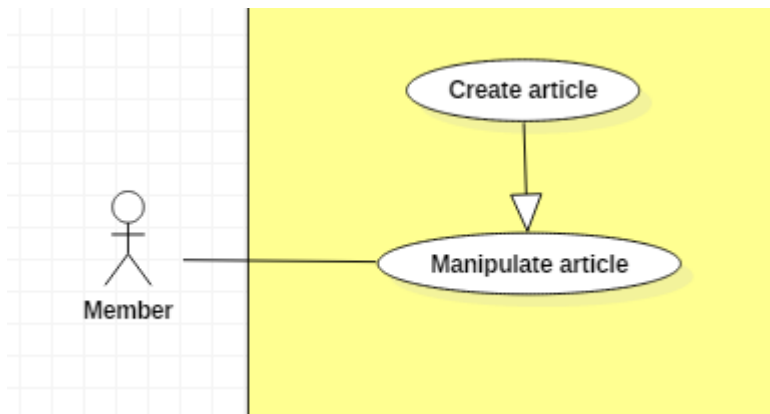


Figure 8 - < Member > Create article Usecase

USE CASE – UC_Member_05			
Usecase No.	Member_05	Usecase Version	1.0
Usecase Name	Create article		
Author	KhanhNB		
Date	25/09/2018	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 user to create a new article.</li></ul> <b>Goal:</b> <ul style="list-style-type: none"><li>- A new article added to the database system.</li></ul> <b>Triggers:</b> <ul style="list-style-type: none"><li>- Member write an articles and submit it.</li></ul> <b>Preconditions:</b> <ul style="list-style-type: none"><li>- User must login into system with role Member.</li></ul> <b>Post Conditions:</b> <ul style="list-style-type: none"><li>- <b>Success:</b> New article is added to the database system.</li><li>- <b>Fail:</b> Show error message.</li></ul> <b>Main Success Scenario:</b>			

Step	Actor Action	System Response
1	Member go to “Create a new article” page	<p>“Create new article” view is shown with following labels and fields:</p> <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.	<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.



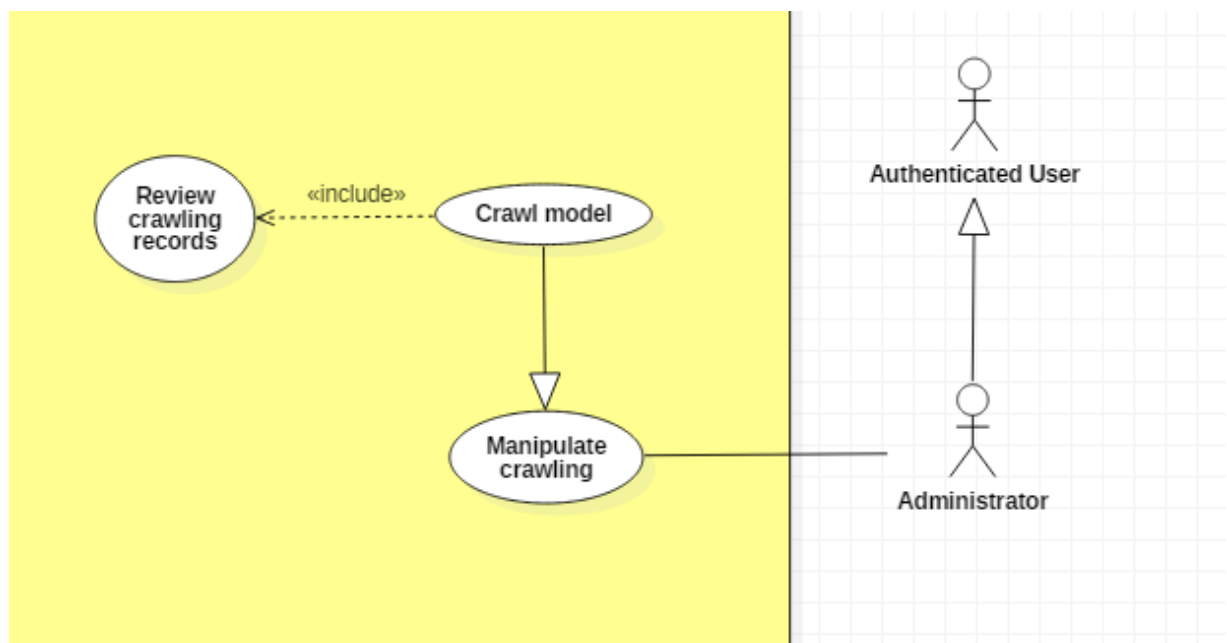
4	Member submit article with empty content	System shows error message to ask the member to enters missing required fields.
5	Member upload file too large.	System shows error message to ask the member to enters missing required fields.

**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.

#### 4.2.6. < Administrator > Crawl model Usecase



**Figure 9 - < Administrator > Crawl model Usecase**

USE CASE – UC_Admin_06			
Usecase No.	Admin_06	Usecase Version	1.0
Usecase Name	Crawl model		
Author	TungHNM		

Date	28/09/2018	Priority	Normal
------	------------	----------	--------

**Actor:**

- Administrator.

**Summary:**

- This use case allows administrator to crawl model from high domain:  
<https://www.1999.co.jp/eng/gundam>.

**Goal:**

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

**Triggers:**

- 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	System displays the following items: <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	System displays messages: “System is crawling successfully”.

		System displays icon “Track crawling process” button  Save new models to database, log file.
3	Administrator switches 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**

Step	Actor Action	System Response
1	Administrator switches 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 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

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

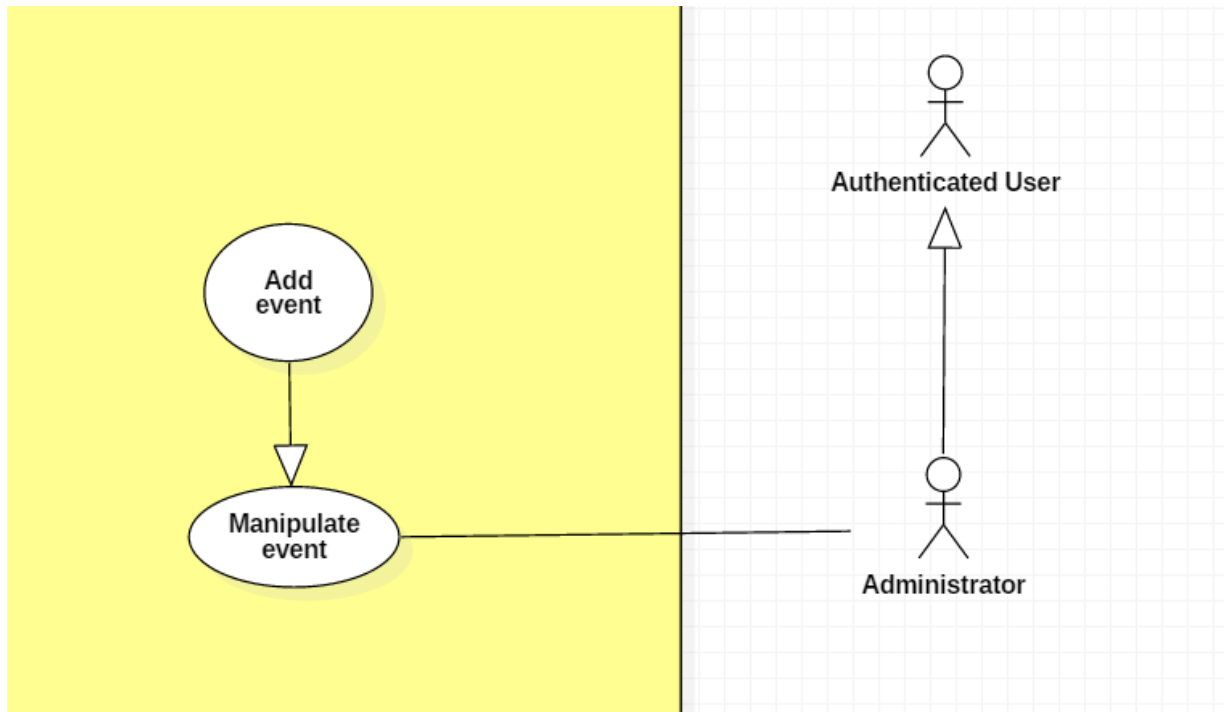


Figure 10 - < Administrator > Create new event Usecase

USE CASE – UC _Admin_07			
Usecase No.	Admin_07	Usecase Version	1.0
Usecase Name	Create new 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 post a new event</li> </ul> <b>Goal:</b> <ul style="list-style-type: none"> <li>- The event is posted on the website</li> </ul> <b>Triggers:</b> <ul style="list-style-type: none"> <li>- Admin fill in the event's info and submit</li> </ul>			

**Preconditions:**

- User must login into system with role Administrator.

**Post Conditions:**

- **Success:** The event is added to the database
- **Fail:** Show error message.

**Main Success Scenario:**

Step	Actor Action	System Response
1	Admin go to “Create new event” page	“Create new event” view is shown with following labels and fields: <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 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 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

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

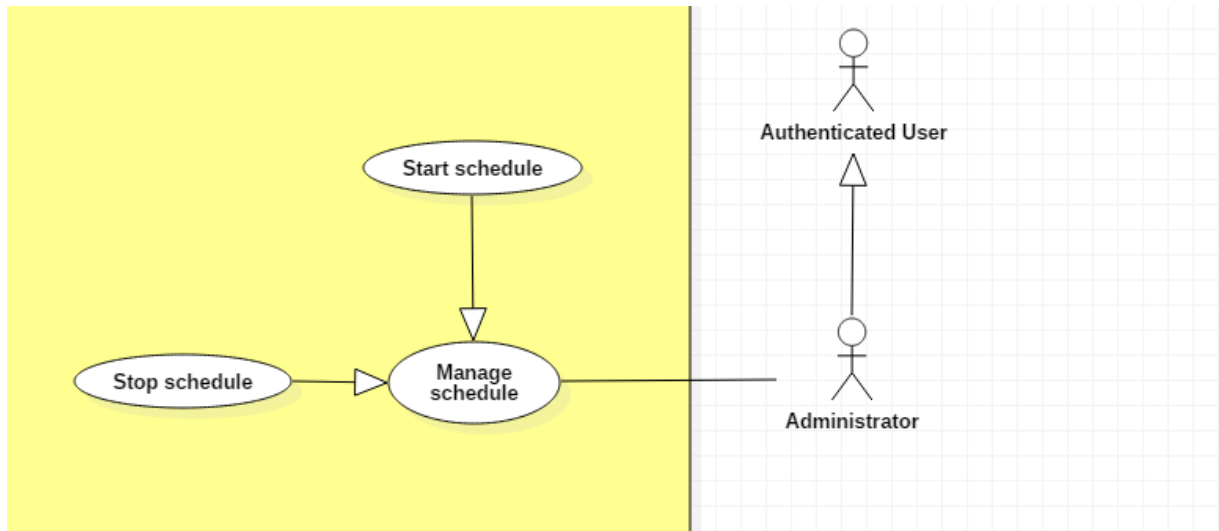


Figure 11 - < Administrator > Start/Stop schedule Usecase

USE CASE – UC _Admin_08			
Usecase No.	Admin_08	Usecase Version	1.0
Usecase Name	Start/Stop schedule		
Author	TungHNM		
Date	30/09/2018	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 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.</li> </ul> <b>Goal:</b> <ul style="list-style-type: none"> <li>- Administrator is able to control the schedule with input cycle and schedule will auto update system data after the input interval.</li> </ul> <b>Triggers:</b> <ul style="list-style-type: none"> <li>- Administrator sends start/stop schedule command to the system.</li> </ul>			

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



		message: Are you sure ?, with 2 buttons: <ul style="list-style-type: none"> <li>• Yes</li> <li>• No</li> </ul>
5	Administrator clicks on “Yes” button  [Alternative 1]	The selected schedule is interrupted.  System displays message: “Schedule {schedule name} is stopped successfully”

**Alternative Scenario:**

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

Step	Actor Action	System Response
1	Administrator clicks on “No” button	The dialog disappears and system views the manage schedule page

**Exceptions:**

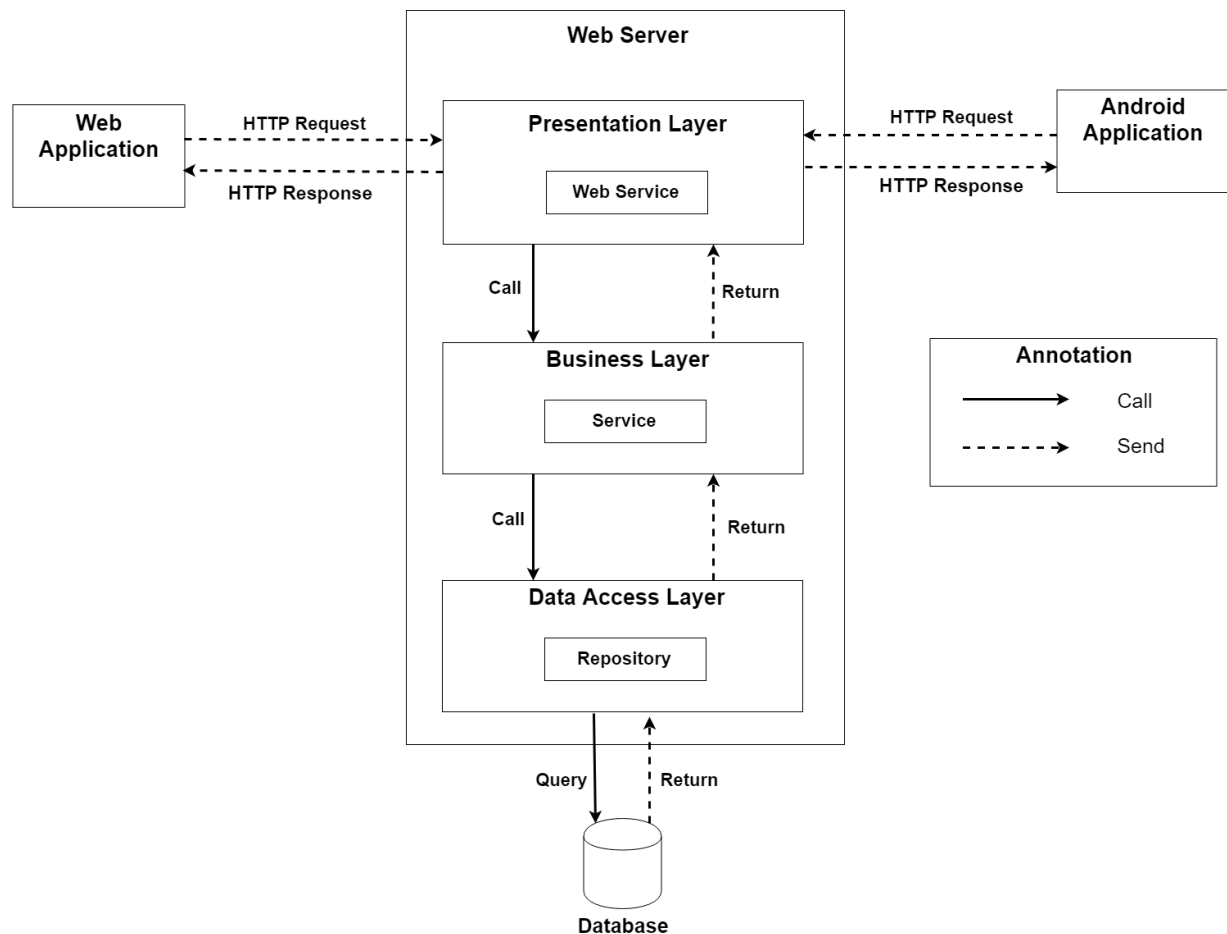
No	Cause	System Response
1	Out of range input cycle	System display error message to the view
2	Is not integer	System display error message to the view
3	Missing required field	System display error message to the view

**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)

## 5. Architectural Diagram



**Figure 12 - Architectural Diagram**

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

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

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

- **WebController:** is the part of presentation layer that acts like event handler to handle user interaction 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.

## 6. 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 3 - Component Dictionary**

## 7. Class Diagram

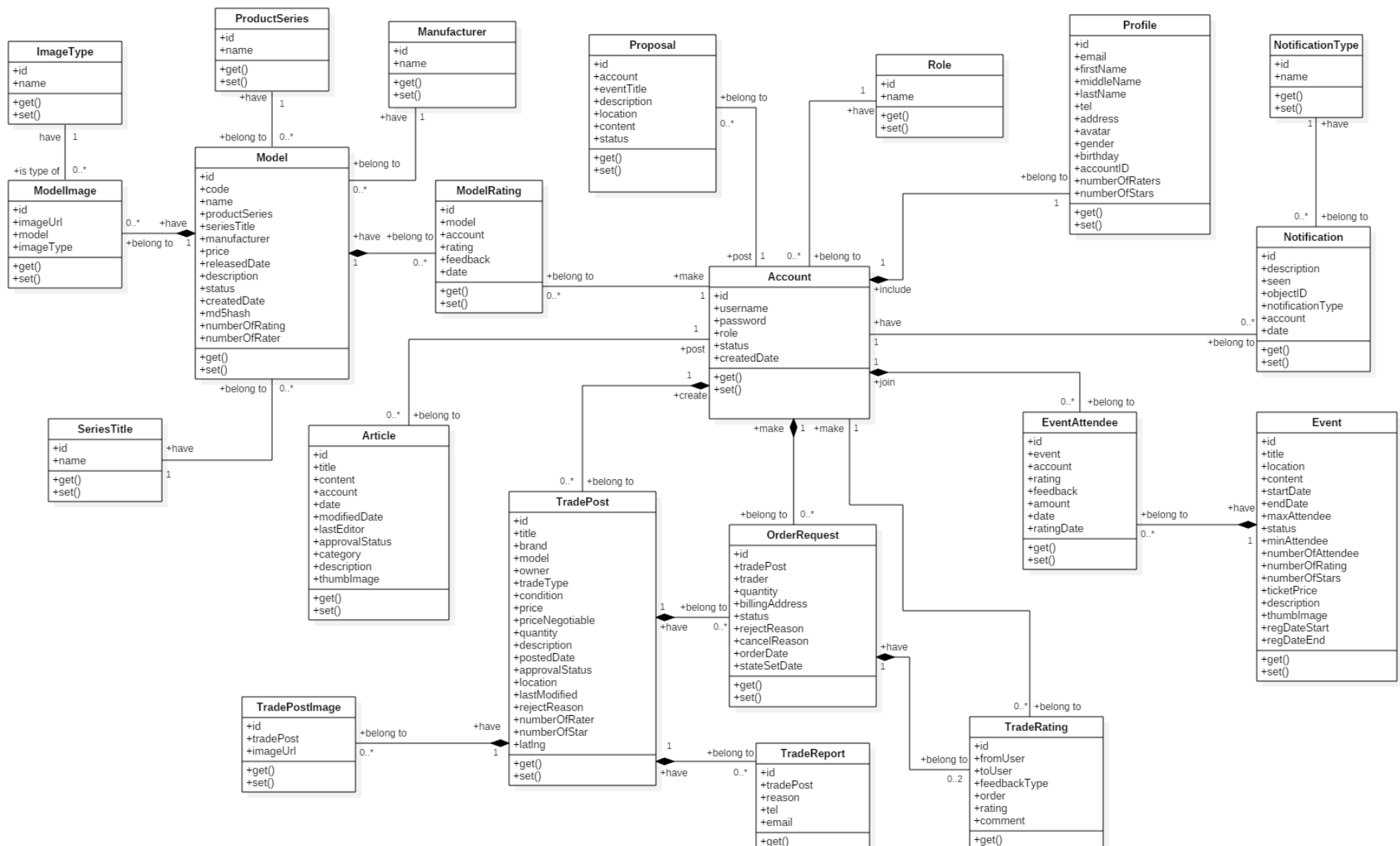


Figure 14 - Class Diagram

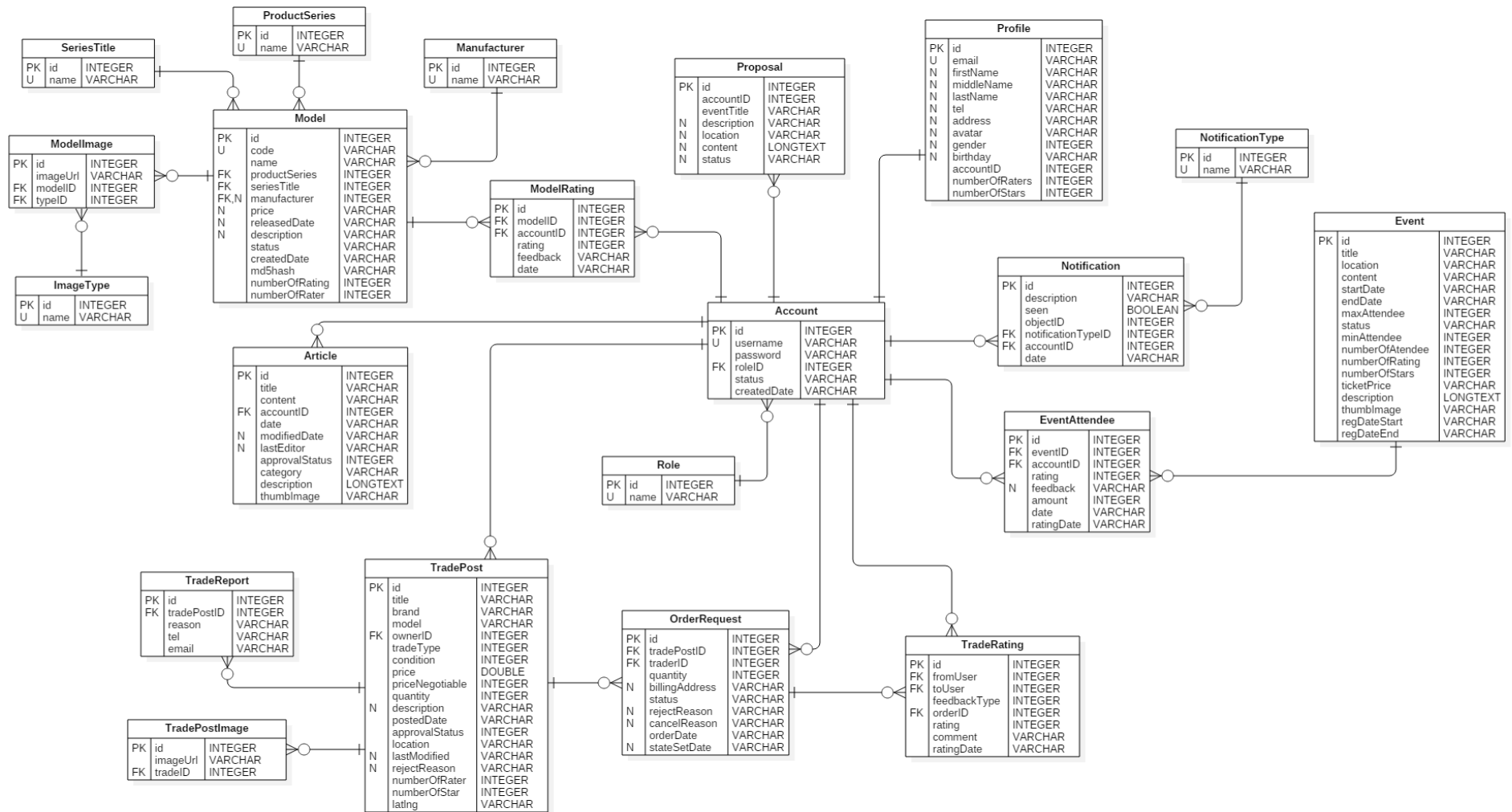
<b>Class Name</b>	<b>Mapping column with Conceptual diagram</b>	<b>Description</b>
Model	Model	Contain model's information
ModelImage	N/A	Not exist in conceptual diagram. It's used to contain models's images
ImageType	N/A	Not exist in conceptual diagram. It's used to contain types of images
ProductSeries	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 4 - Class Dictionary**



## 8. Entity Relationship Diagram



**Figure 15 - Entity Relationship Diagram**

## 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 5 - ERD Dictionary

## 9. Interaction Diagram

### 9.1. Sequence Diagram

### 9.1.1. Search Trade Post

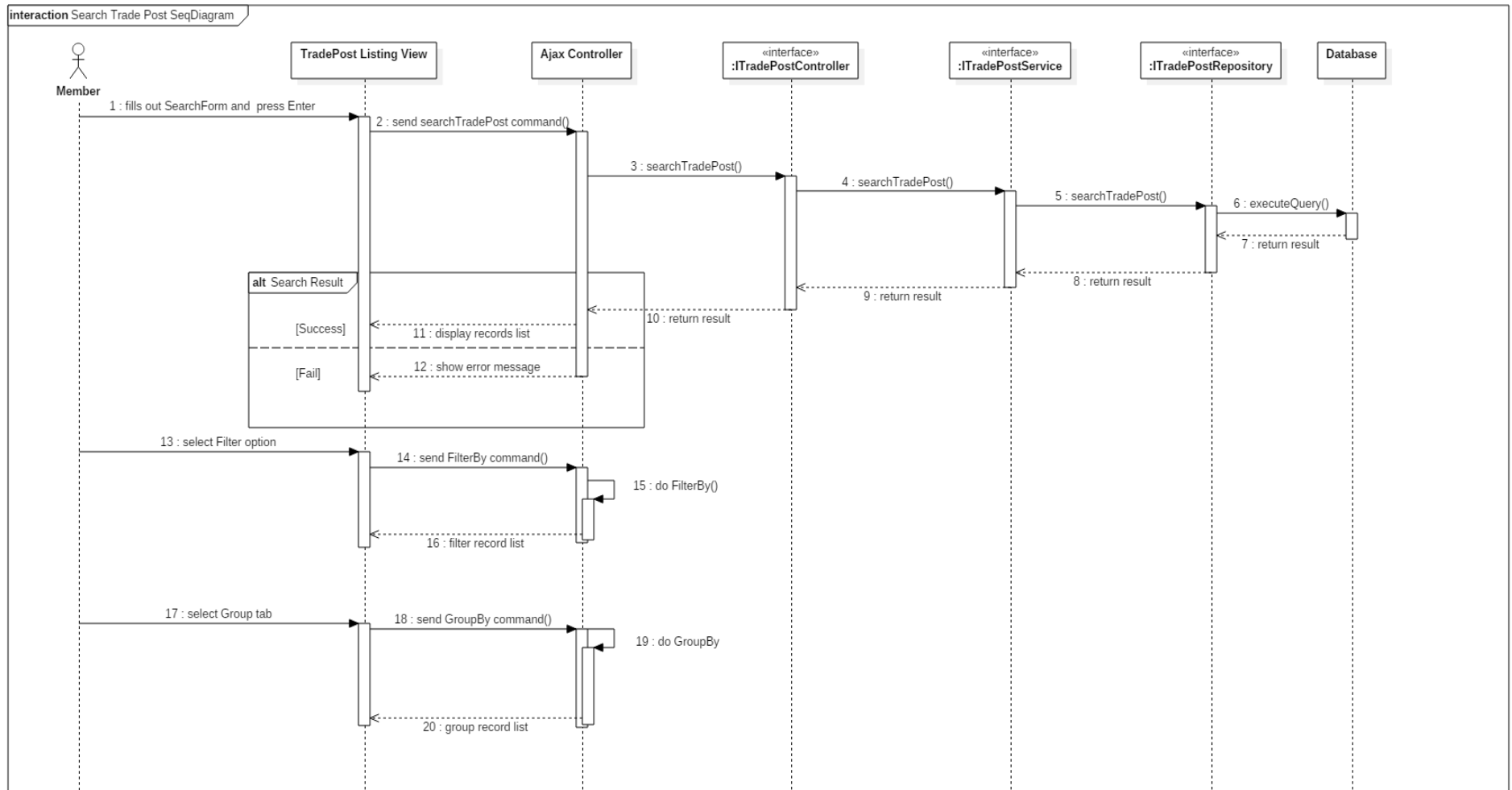


Figure 16 - < Sequence Diagram > Search Trade Post

### 9.1.2. Request order to a trade post

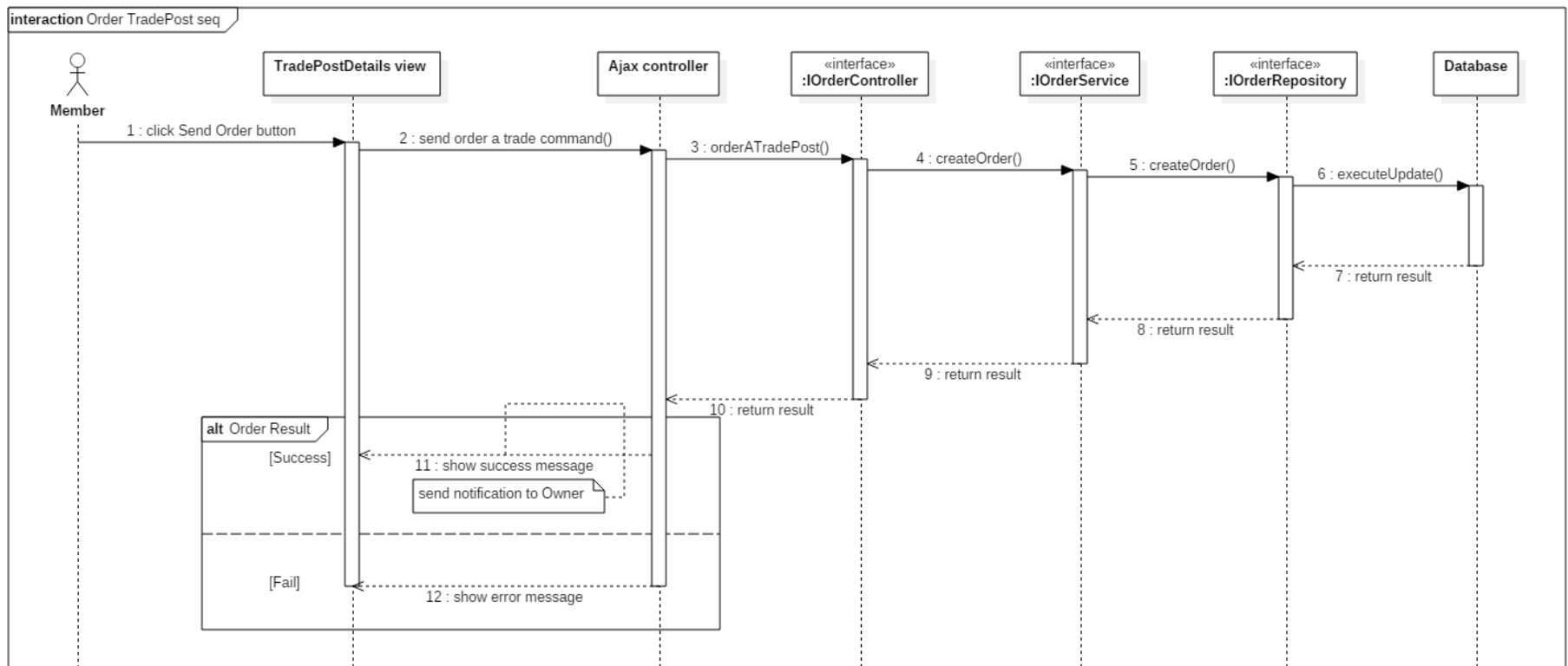


Figure 17 - < Sequence Diagram > Request order

### 9.1.3. Register event

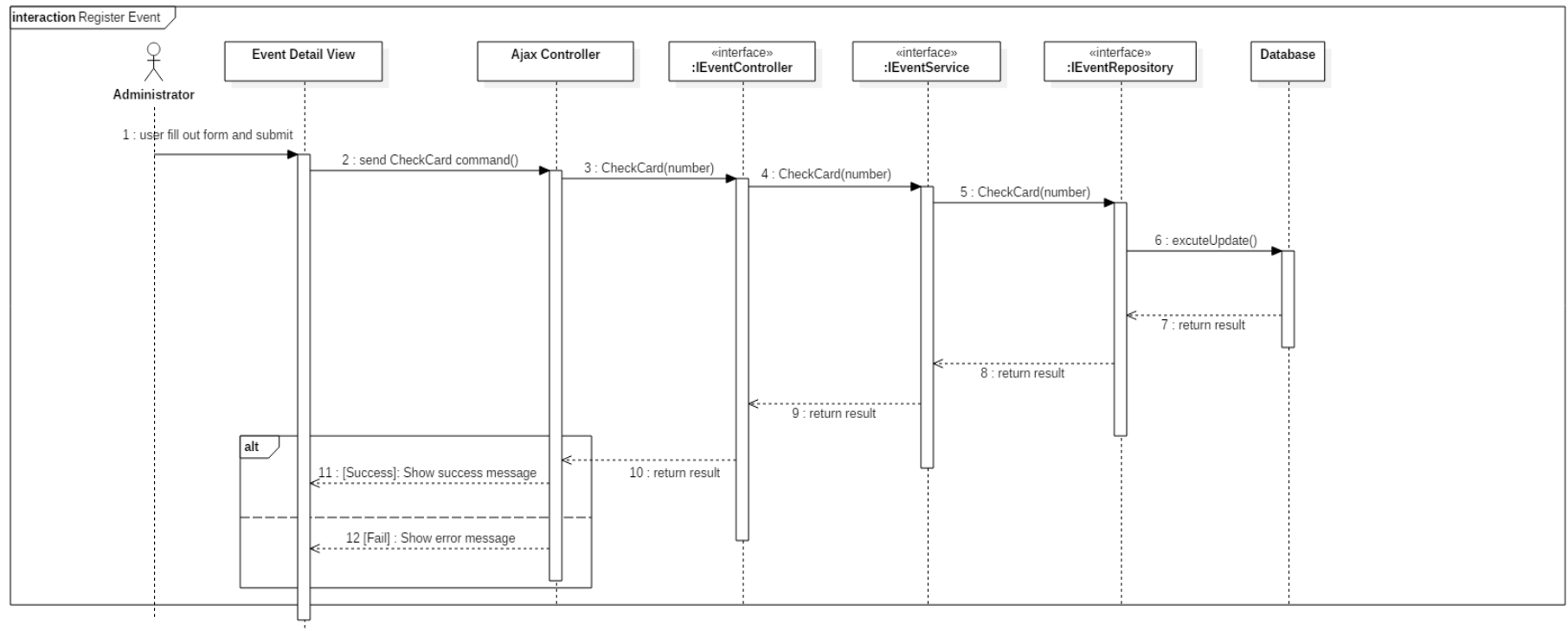


Figure 18 - < Sequence Diagram > Register event

## 9.2. Activity Diagram

### 9.2.1. Search Gundam Model

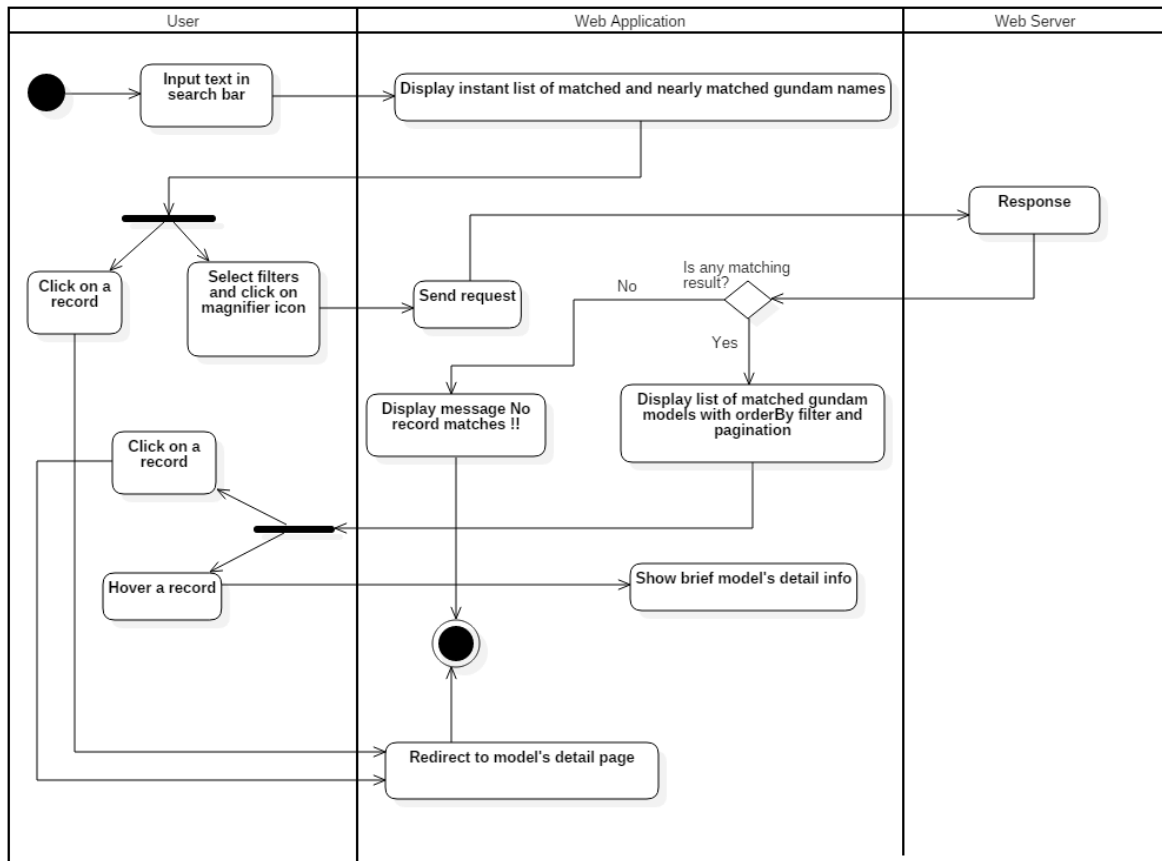


Figure 19 - < Activity Diagram > Search Gundam Model

### 9.2.2. Manage order

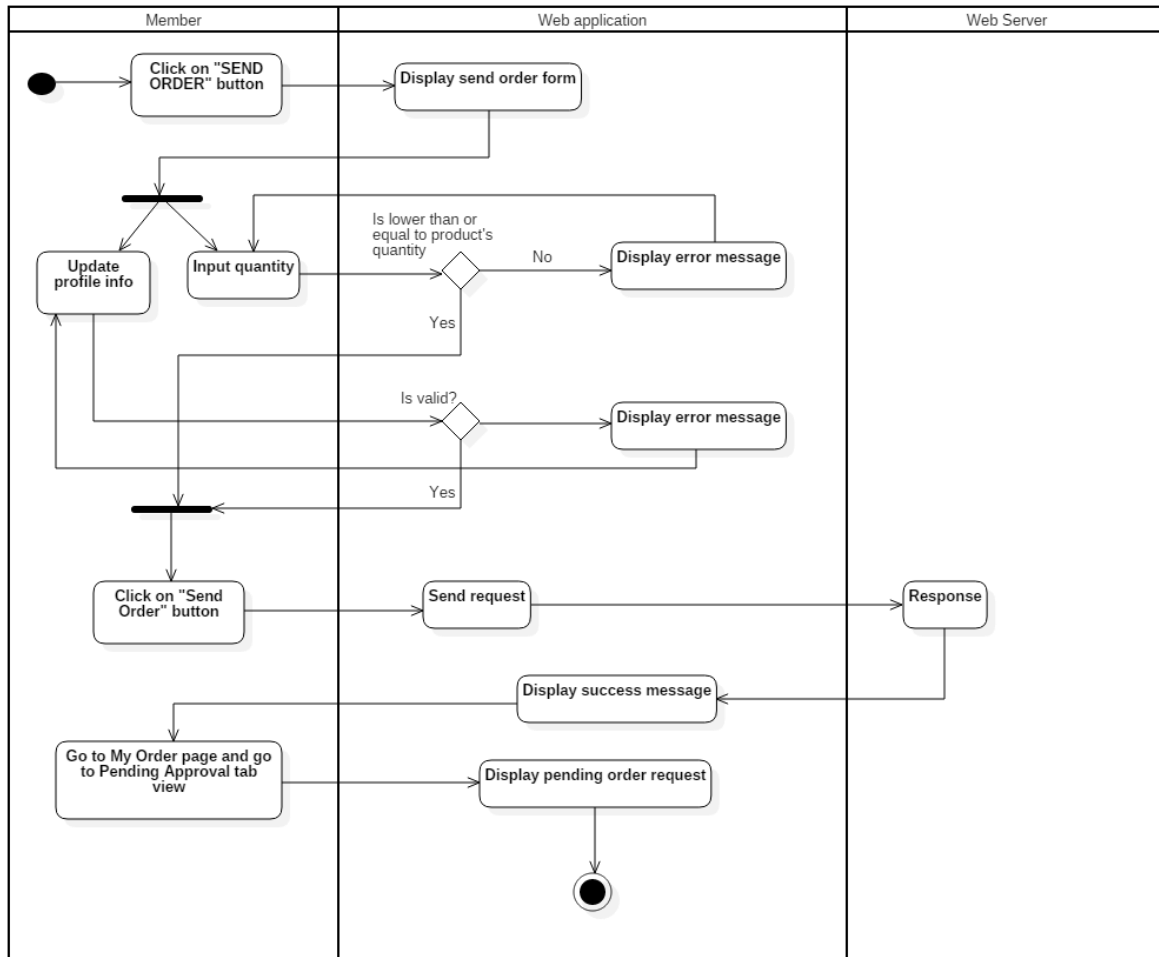


Figure 20 - < Activity Diagram> Manage order

## 10. Physical Diagram

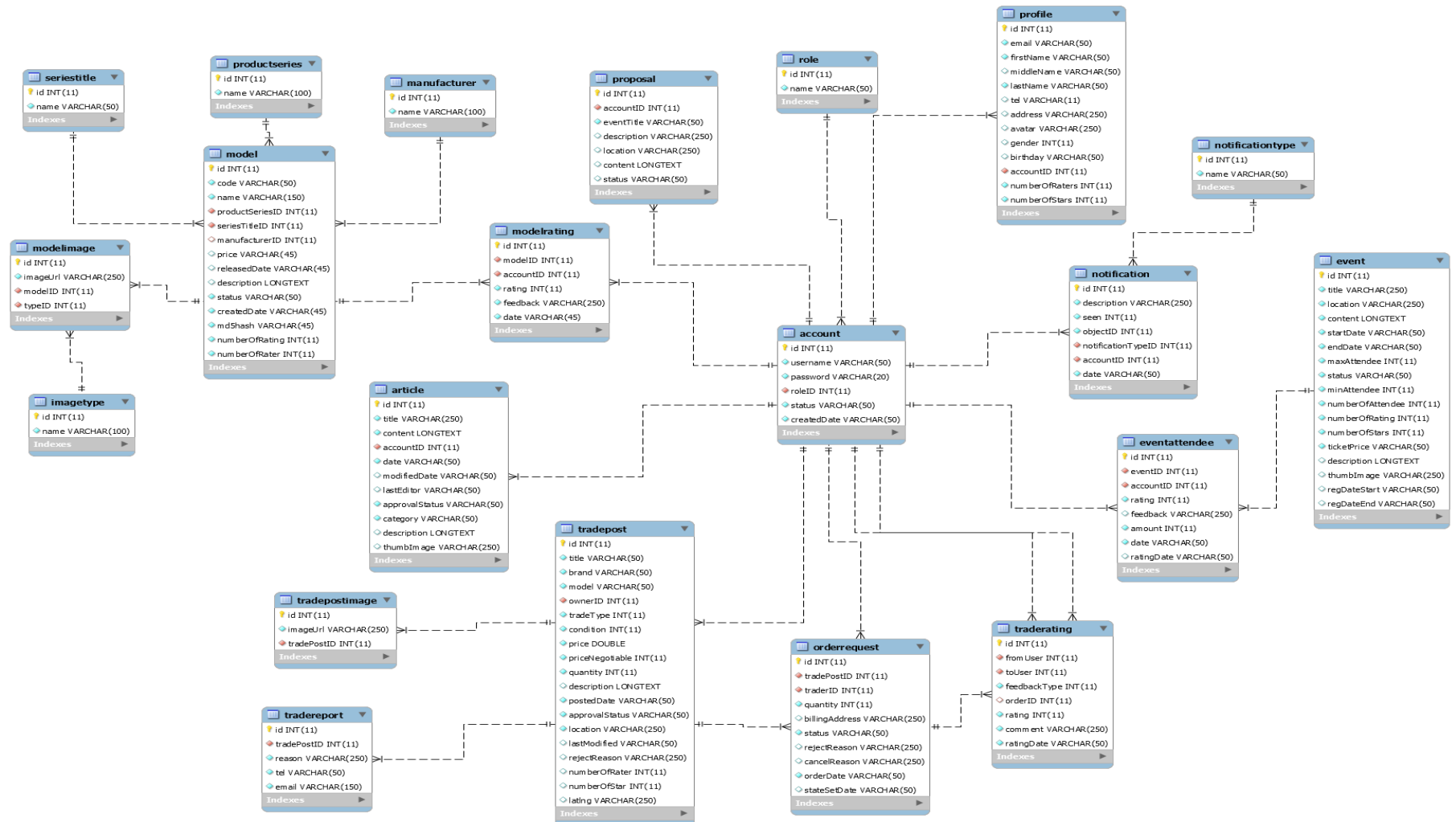


Figure 21 - Physical Diagram



## 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
eventattende	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
trade	Table which contains trade post information, which is created by buyer/seller
traderating	Table which contains user's evaluation about the exchange
trade	Table which contains trade post's images
trade	Table which contains information about user's report on exchange
orderrequest	Table which contains member's order request to trade posts

**Table 6 - Physical Diagram Dictionary**

# 11. Algorithms

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

### 11.1.1. 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.

### 11.1.2. 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 requests from client to speed up the system, the fewer requests send to server, the faster client will load.

### 11.1.3. 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

#### 11.1.4. Flow chart

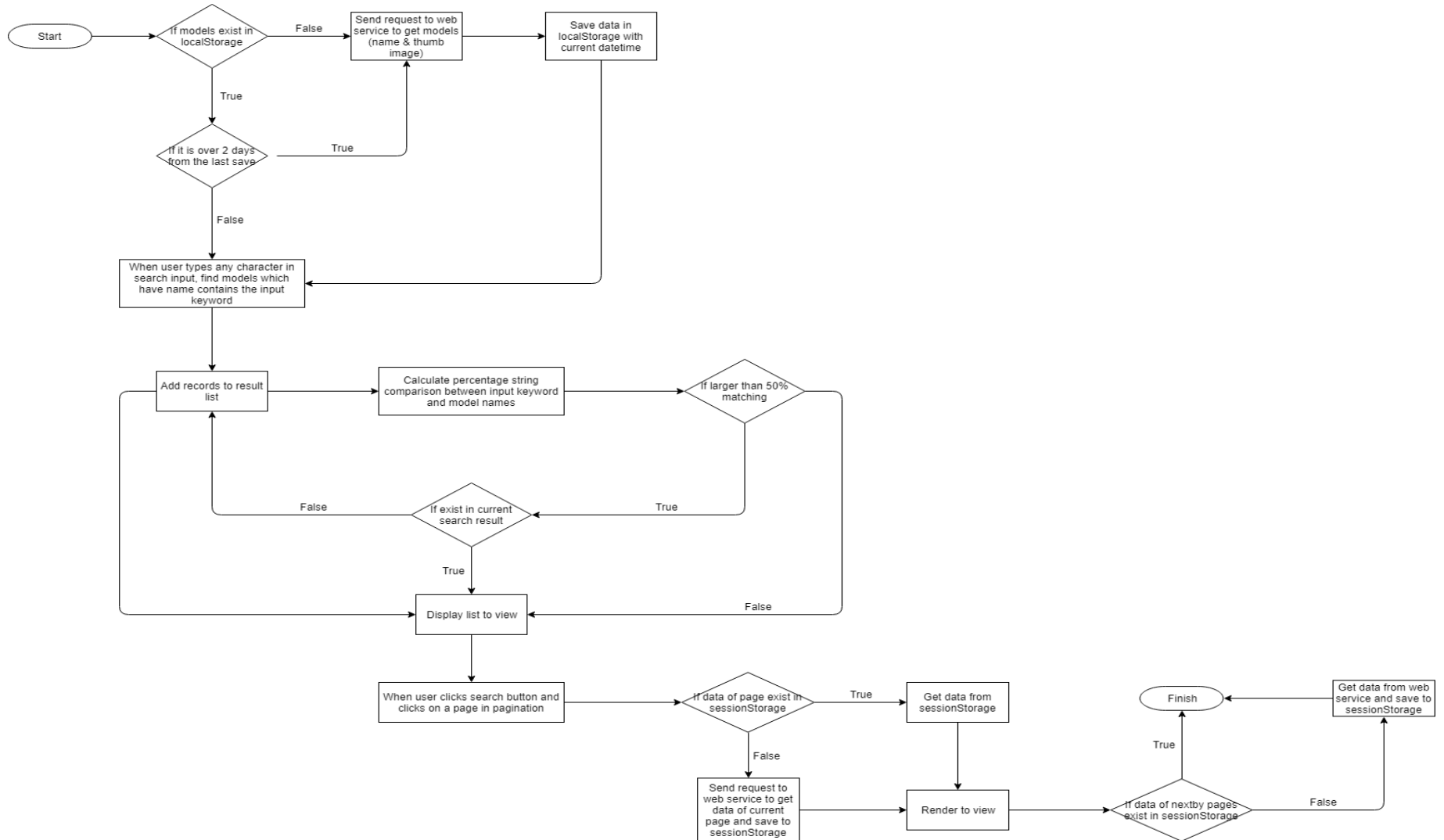


Figure 22 - As-you-type search work flow

## **11.2. MD5 hash function**

### **11.2.1. 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.

### **11.2.2. 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.

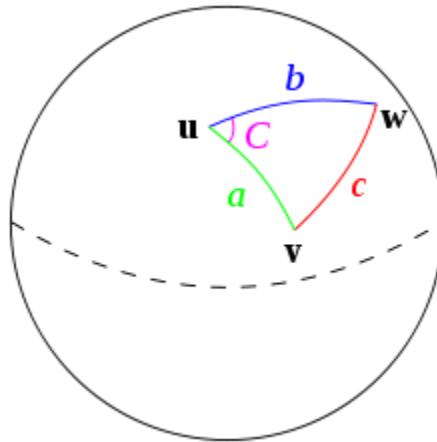
### **11.2.3. Solution**

- 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

## 11.3. Haversine formula

### 11.3.1. 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.



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

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

$$d = R \cdot c$$

### 11.3.2. 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.

### 11.3.3. 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 selected this exchange to search result list.

## 12. Future Plan

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.