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

|  |
| --- |
| **FPT UNIVERSITY** |
| Capstone Project Document |
| **Gunpla World Application** |
|  |
| |  |  | | --- | --- | | **Group 15** | | | **Group Members** | Hoàng Ngô Minh Tùng – SE62231  Nguyễn Bảo Khánh – SE61978  Lê Nguyễn Ngọc Sang – SE61647 | | **Supervisor** | Phạm Công Thành | | **Ext Supervisor** | N/A | | **Capstone Project code** | gwa | |
| – Ho Chi Minh City, **09/2018** –  **CAPSTONE PROJECT REGISTER**  Class: Duration time: From 14/5/2018 To 28/08/2018  (\*) Profession: <Software Engineer> Specialty: <ES> <IS>  x  (\*) Kinds of person make registers: Lecturer Students  x  1. Register information for supervisor (if have)   |  |  |  |  |  | | --- | --- | --- | --- | --- | |  | **Full name** | **Phone** | **E-Mail** | **Title** | | Supervisor 1 | Nguyễn Huy Hùng |  | hungnh@fpt.edu.vn | Mr. |   2. Register information for students (if have)   |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | |  | **Full name** | **Code** | **Phone** | **E-mail** | **Role** | | Student 1 | Lê Hùng Sơn | SE61849 | 01284901098 | sonlhse61849@fpt.edu.vn | Leader | | Student 2 | Võ Hoàng Việt | SE61846 | 0984773711 | vietvhse61846@fpt.edu.vn | Member | | Student 3 | Nguyễn Thanh Nhã | SE61236 | 0965671434 | nhantse61236@fpt.edu.vn | Member | | Student 4 | Nguyễn Kiến Huy | SE61871 | 01228654868 | huynkse61871@fpt.edu.vn | Member |   3. Register content of Capstone Project  (\*) 3.1. Capstone Project name:  **English:** Criminal Face Detection.  **Vietnamese:** Hệ Thống Nhận Diện Khuôn Mặt Tội Phạm.  **Abbreviation:** CFD  (\*) 3.2. Main proposal content (including result and product)   1. Theory and practice (document):    * Student should apply the software development process and the UML 2.0 in modeling the system.    * 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.    * Server-side technique:      + Database design, OOA, OOD, OOP, MVC, Java or .Net technology, …      + Apache Lucene, ElasticSearch.    * Client-side technique      + HTML5, CSS, JavaScript, JQuery, Ajax      + Mobile Platform (iOS, Android) 2. Program:   Build a system to help public organizations/places (Super Markets, Hospitals…) detect faces of suspects/criminals. The following workflow and features should be implemented:   * + - * The system connects with cameras in a particular place.       * The system captures walk by faces and compares with the built-in data.       * If the results are likely matched with the built-in data, alert and notification should be send to administrator via Email and Phone.       * The administrator should be able to add/remove facing data.  1. Other products:  * All of management functions of the system must be implemented to support the operating system.   4. Other comment (propose all relative thing if have)  N/A   |  |  | | --- | --- | | **Supervisor (If have)**  *(Sign and full name)* | HCM city, date 12/12/2017  **On behalf of Registers**  *(Sign and full name)* |   Nguyễn Huy Hùng  Contents  [A. Introduction 4](#_Toc532068359)  [1. Project Information 4](#_Toc532068360)  [2. Introduction 4](#_Toc532068361)  [3. Current Situation 5](#_Toc532068362)  [4. Problem Definition 5](#_Toc532068363)  [5. Proposed Solution 5](#_Toc532068364)  [5.1 Feature functions 5](#_Toc532068365)  [5.2 Advantages and disadvantages 6](#_Toc532068366)  [6. Functional Requirements 6](#_Toc532068367)  [7. Role and Responsibility 8](#_Toc532068368)  [B. Software Project Management Plan 8](#_Toc532068369)  [1. Problem Definition 8](#_Toc532068370)  [1.1. Name of this Capstone Project 8](#_Toc532068371)  [1.2. Problem Abstract 8](#_Toc532068372)  [1.3. Project Overview 9](#_Toc532068373)  [2. Project Organization 13](#_Toc532068374)  [2.1. Software Process Model 13](#_Toc532068375)  [2.2. Roles and responsibilities 14](#_Toc532068376)  [2.3. Tools and Techniques 15](#_Toc532068377)  [3. Project Management Plan 16](#_Toc532068378)  [3.1. Software development life cycle 16](#_Toc532068379)  [3.2. Phase Detail 18](#_Toc532068380)  [3.3. All Meeting Minutes 21](#_Toc532068381)  [C. Software Requirement Specification 22](#_Toc532068382)  [1. User Requirement Specification 22](#_Toc532068383)  [1.1. Member Requirement 22](#_Toc532068384)  [1.2. Buyer/Seller Requirement 23](#_Toc532068385)  [1.3. Administrator Requirement 23](#_Toc532068386)  [1.4. Authenticated User Requirement 23](#_Toc532068387)  [1.5. Handler Requirement 24](#_Toc532068388)  [2. System Requirement Specification 24](#_Toc532068389)  [2.1. External Interface Requirement 24](#_Toc532068390)  [2.2. System Overview Usecase 26](#_Toc532068391)  [3. Software System Attribute 27](#_Toc532068392)  [3.1. Usability 27](#_Toc532068393)  [3.2. Reliability 27](#_Toc532068394)  [3.2. Availability 27](#_Toc532068395)  [3.3. Security 27](#_Toc532068396)  [3.4. Maintainability 27](#_Toc532068397)  [3.5. Portability 28](#_Toc532068398)  [3.6. Performance 28](#_Toc532068399)  [4. Conceptual Diagram 28](#_Toc532068400)  [D. Software Design Description 30](#_Toc532068401)  [1. Design Overview 30](#_Toc532068402)  [2. System Architectural Design 31](#_Toc532068403)  [2.1. Web application architecture description 31](#_Toc532068404)  [3. Component Diagram 32](#_Toc532068405)  [4. Detailed Description 1](#_Toc532068406)  [4.1. Class Diagram 1](#_Toc532068407)  [6. Database Design 4](#_Toc532068408)  [6.1. Entity Relationship Diagram (ERD) 5](#_Toc532068409)  [6.2. Data Dictionary 6](#_Toc532068410)  [7. Algorithms 7](#_Toc532068411)  [7.1. As-you-type search with instant results and pagination with sessionStorage 7](#_Toc532068412)  [7.2. MD5 hash function matching new data 7](#_Toc532068413)  [7.3. Track schedule progress 7](#_Toc532068414)  [7.4. Search nearby location with Haversine formula 7](#_Toc532068415) |

# A. Introduction

## 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

## 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**, aportmanteau of "*Gun*dam *pla*stic 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 .

## 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.

## 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.

## 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

## Functional Requirements

1. **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

1. **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

1. **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

1. **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

1. **User:**

User component:

* Update profile, get user’s info with statistic
* Get notification
* Get exchange evaluation

Administrator component:

* Manage account, ban/unban account

1. **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

## Role and Responsibility

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

# B. Software Project Management Plan

## Problem Definition

### 1.1. Name of this Capstone Project

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

### 1.2. Problem Abstract

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

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

### 1.3. Project Overview

#### 1.3.1. Current Situation

Below are the problems encountered in this project:

* **Security**: While spamming can be minimized with the use of filters and admin system, an attack can create many accounts to attack the sites with spam.
* **Legal issues:** There maybe legal issues regarding the information we use on our website.
* **Absence of team members**: team members can get sick or unexpected problems.

#### 1.3.2. The Proposed System

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

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

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

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

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

##### 1.3.2.1 Website

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

* For 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 defend against attacks, and it can provide important information like the number of attendees for an event so the company can manage that event effectively.

+ A registered user can propose an event from their idea and the host company can use that information to judge what is the best way to host an event.

+ A post about selling or trading have a locating function that accurately locates the user, the sellers can also enter the address by themselves if they choose to.

+ If you’re looking to buy something, the system can show the nearest sellers that are selling what you’re looking for based on your location.

#### 1.3.5. Development Environment

##### 1.3.5.1. Hardware requirements

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

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

##### 1.3.5.2. Software requirements

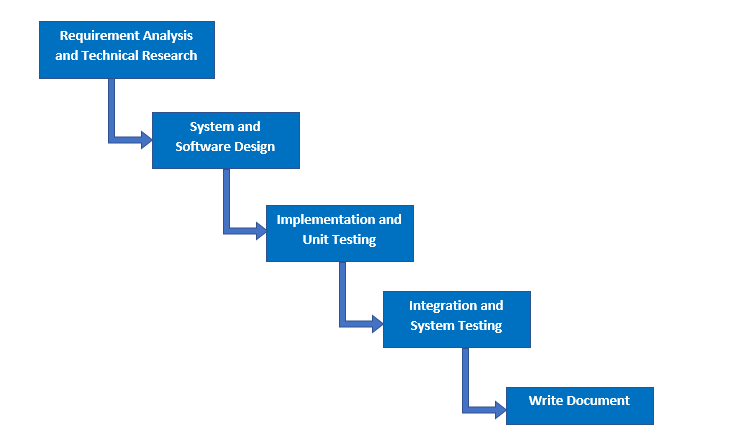
|  |  |  |
| --- | --- | --- |
| **Software** | **Name and version** | **Description** |
| **Operating System** | Window 10 1803 | Operating system and platform for development |
| **Environment** | Java EE 8, Android SDK APIs level 15 or above | Specification for developing web and android application |
| **Modeling Tool** | StarUML 2.8.1 | Used to implement website and web service |
| **IDE** | IntelliJ IDEA 2018.2.3, Android Studio 3.1.3 | Programming tools |
| **DBMS** | MySQL 5.7.20, MySQL Workbench 6.3.9 CE | Used to create & manage the database for system |
| **Source control** | SourceTree 2.6.10 | Used for source control |
| **Web browser** | Chrome 62 or above | Testing browser |

## 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 ambigous.
* 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 seperate 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 | * Specify user requirement * Control the development process * Give out technique and business analysis support |
| **2** | Hoàng Ngô Minh Tùng | Team Leader, QA, DEV, Tester | * Manage the process, quality, risk * GUI design * Design database * Clarify requirements * Prepare documents, reports * Arrange meeting * Design project architect * Coding * Testing |
| **3** | Nguyễn Bảo Khánh | Team Member, BA, DEV, Tester | * GUI design * Design database * Clarify requirements, analyze business * Design project architect * Coding * Testing |
| 4 | Lê Nguyễn Ngọc Sang | Team Member, QC, DEV, Tester | * Create test plan * GUI design * Design database * Clarify requirements, analyze business * Design project architect * Coding * Testing |

### 2.3. Tools and Techniques

|  |  |
| --- | --- |
| **Tool** | **Name and version** |
| IDE | * IntelliJ IDEA 2018.2.3 * Android Studio 3.1.3 |
| Source control | SourceTree 2.6.10 |
| Database Manager | MySQL Workbench 6.3.9 CE |

|  |  |
| --- | --- |
| **Technique** | **Name and version** |
| Front end | * HTML v5.0 * JQuery v3.2.1 * Bootstrap v3.3.7 * CSS * Javascript * Ajax |
| Back end | * Java EE 8 * Spring Boot v2.0.5 RELEASE * Hibernate v5.0.12 * RESTful Web Service |
| Google API libraries | * Google Maps API * Firebase Cloud Messaging |
| Web server | Apache Tomcat v8.5.15 |
| Database system | MySQL 5.7.20 |

## Project Management Plan

### 3.1. Software development life cycle

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Phase** | **Description** | **Deliverables** | **Resource needed** | **Dependencies and Constraints** | **Risks** |
| **Requirement analysis and technical research** | **-** Identify and clarify requirements for system  - Research for google maps api and crawling technique (StAX), FCM (firebase cloud messaging) | - Software Requirement Specification  - Basic knowledge Google Maps API, FCM  - Understand about StAX technique  - Introduction document  - Project Management Plan  - Prototypes | 4 weeks | N/A | **-** Lack of member share of understand  - SRS does not match user requirement  - Misunderstanding of learning |
| **System and software design** | **-** Architect design for the system  - Detail design using top-down break down | - Software Design Description Document  - Base code structure  - Technology notes | 2 weeks | Depend on “Requirement Analysis and Technical Research” | **-** Lack of experience  - Diagrams don’t match SRS  - Not fulfil requirement |
| **Implementation and unit testing** | **-** Implement the system  - Crawling data  - Building mobile application  - Prepare test plan and test cases  - Unit testing of the code | **-** System Implementation and Test Document  - Programs (web and mobile app)  - Unit test cases and results | 4 weeks | Depend on “System and Software Design” | **-** Delay in implementation  - Inaccurate programming |
| **Integration and system testing** | **-** Perform functional and non functional testing  - Make sure the system meets the requirements | **-** Test report  - Defect report | 1 week | Depend on “Implementation and Unit Testing” | - Missing of functions that need to be tested |
| **Write document** | **-** Write “Software User’s Manual”  - Review all documents  - Write paper | **-** Software User’s Manual  - A paper | 3 weeks | Depend on “System Design” and “Implementation” | -Don’t have enough time to complete paper. |

***Table 1: Software development life cycle***

### 3.2. Phase Detail

#### 3.2.1. Requirement Analysis and Technical Research

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

#### 3.2.2. System and Software Design

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

**3.2.3. Implementation and Unit Testing**

|  |  |  |
| --- | --- | --- |
| **Task** | **Description** | **Author** |
| **Implement Web Service** | Implement main functions  Implement REST api | TungHNM, KhanhNB, SangLNN |
| **Implement GUI** | Implement Web and Mobile UI | TungHNM, KhanhNB, SangLNN |
| **Implement Algorithms** | Implement 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 |
| **Unit testing** | Prepare test cases for all functions  Test each functions and provide check list of defects  Fix bugs base on the check list | TungHNM, KhanhNB, SangLNN |
| **Write document** | Write System Implementation and Test Document | TungHNM, KhanhNB, SangLNN |

**3.2.4. Integration and System Testing**

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

**3.2.5. Write document**

|  |  |  |
| --- | --- | --- |
| **Task** | **Description** | **Author** |
| **Write document** | Write Software User’s Manual Document | TungHNM, KhanhNB, SangLNN |
| **Review all documents** | Review 6 reports | TungHNM, KhanhNB, SangLNN |
| **Write paper** | Write abstract and introduction  Write about As-you-type search and pagination with sessionStorage  Write comparison  Write conclusion | TungHNM, KhanhNB, SangLNN |

### 3.3. All Meeting Minutes

All meeting minutes could be found at [here](https://drive.google.com/open?id=1ACCC2tjce8m-4AyyhLiwlV_43ZZlKM96).

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. 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 logined 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

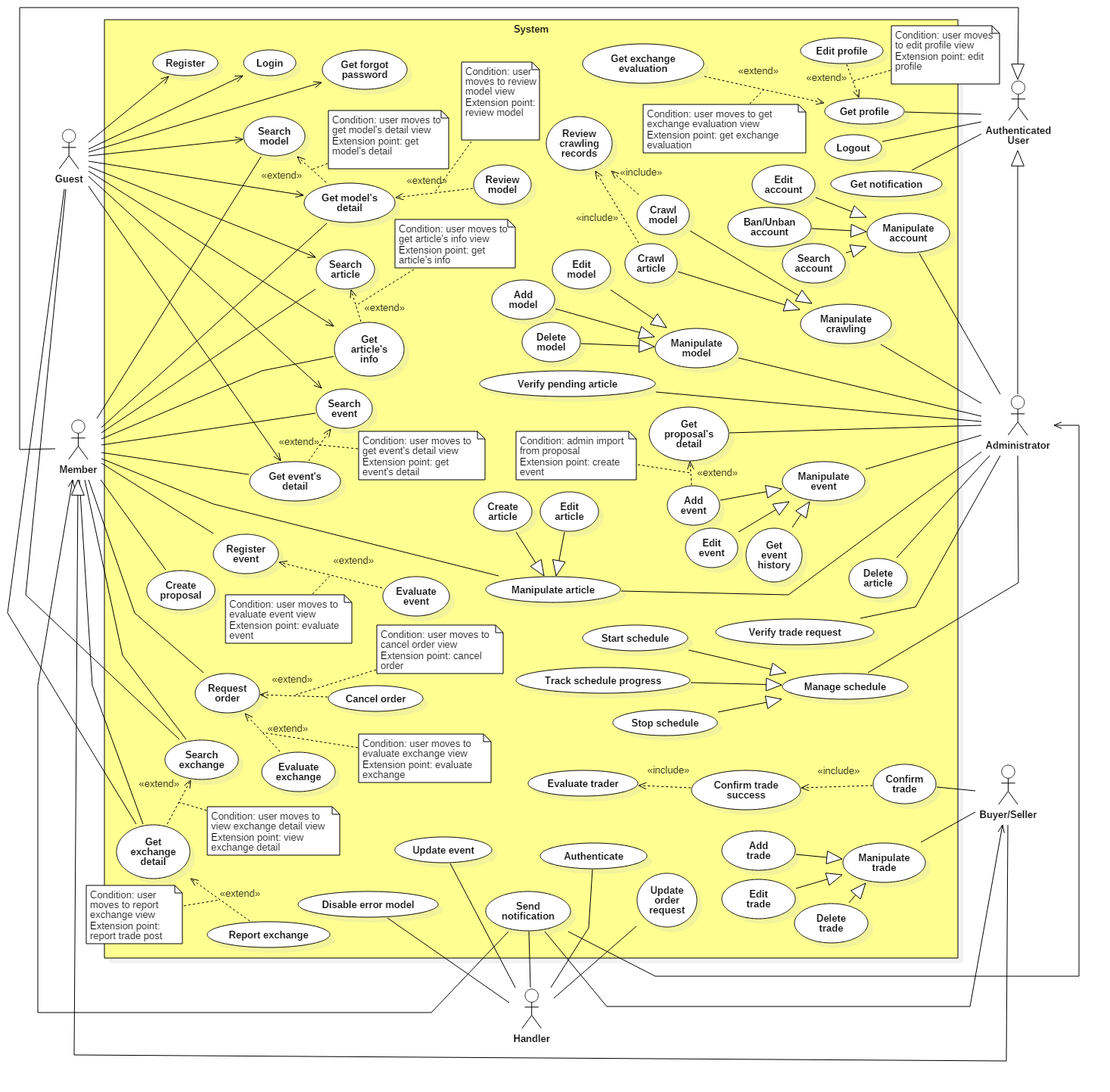
#### 2.1.3. Software Interface

* 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



## 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:
  + Font size: 13-30pt
  + Color: darkblue, black, …
  + 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

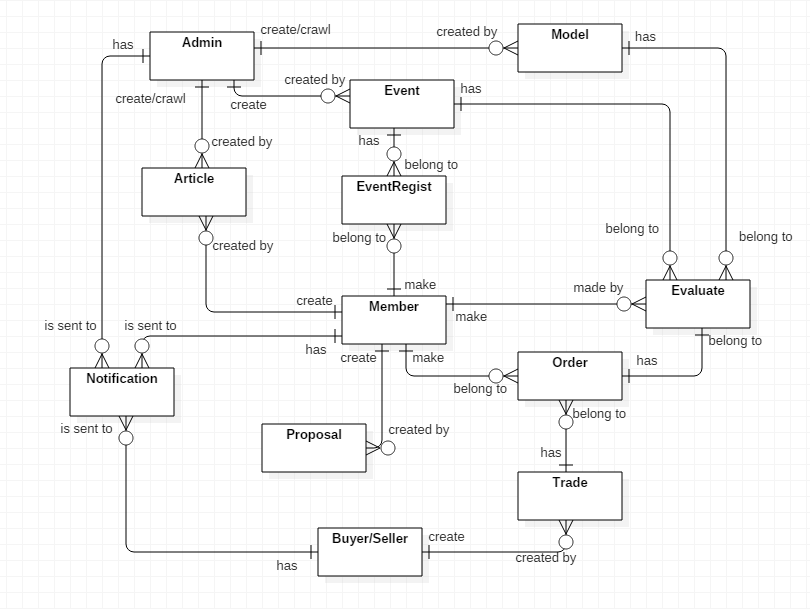
### 3.5. Portability

* 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



**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 | Is sent to administrator when models have error 404 loading images  Is sent to member when administrator update member’s profile  Is sent to users when their order request is accepted/declined/cancel/confirm exchange success or rating from trade post owner  Is sent to buyer/seller when their exchange post is approved/decline by administrator  Is sent to article’s author when their article is approved/disapproved by administrator  Is sent to event’s attendees when the event is cancelled because of not enough attendee to start |
| User (Administrator, Member, Buyer/Seller) | Abstract entities that describe a role in the system |

# D. 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.

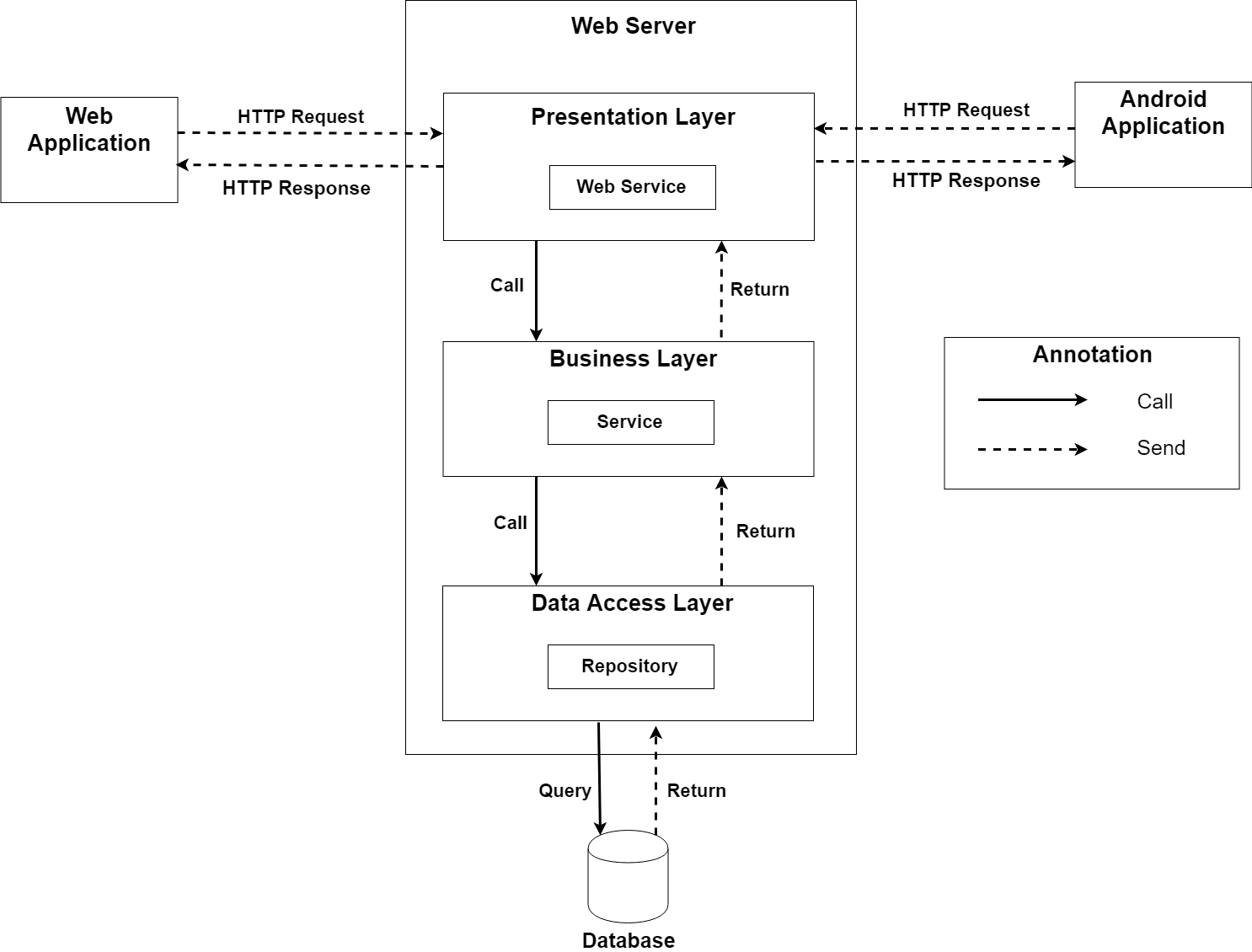
The detailed design describes static and dynamic structure for each components and functions. It includes class diagram, class explanations and sequence diagrams, activity diagrams.

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

Document overview:

* Section 1: gives an overview design of the sytem
* Section 2: gives an overall description of the system architecture design.
* Section 3: gives component diagrams that describe the connection and integration of the system.
* Section 4: gives the detail design description which includes class diagram, class explanation, and sequence diagram to details the application functions.
* Section 5: describe screens design.
* Section 6: describe a fully attributed ERD.
* Section 7: describe algorithms**.**

## 2. System Architectural Design



### 2.1. Web application architecture description

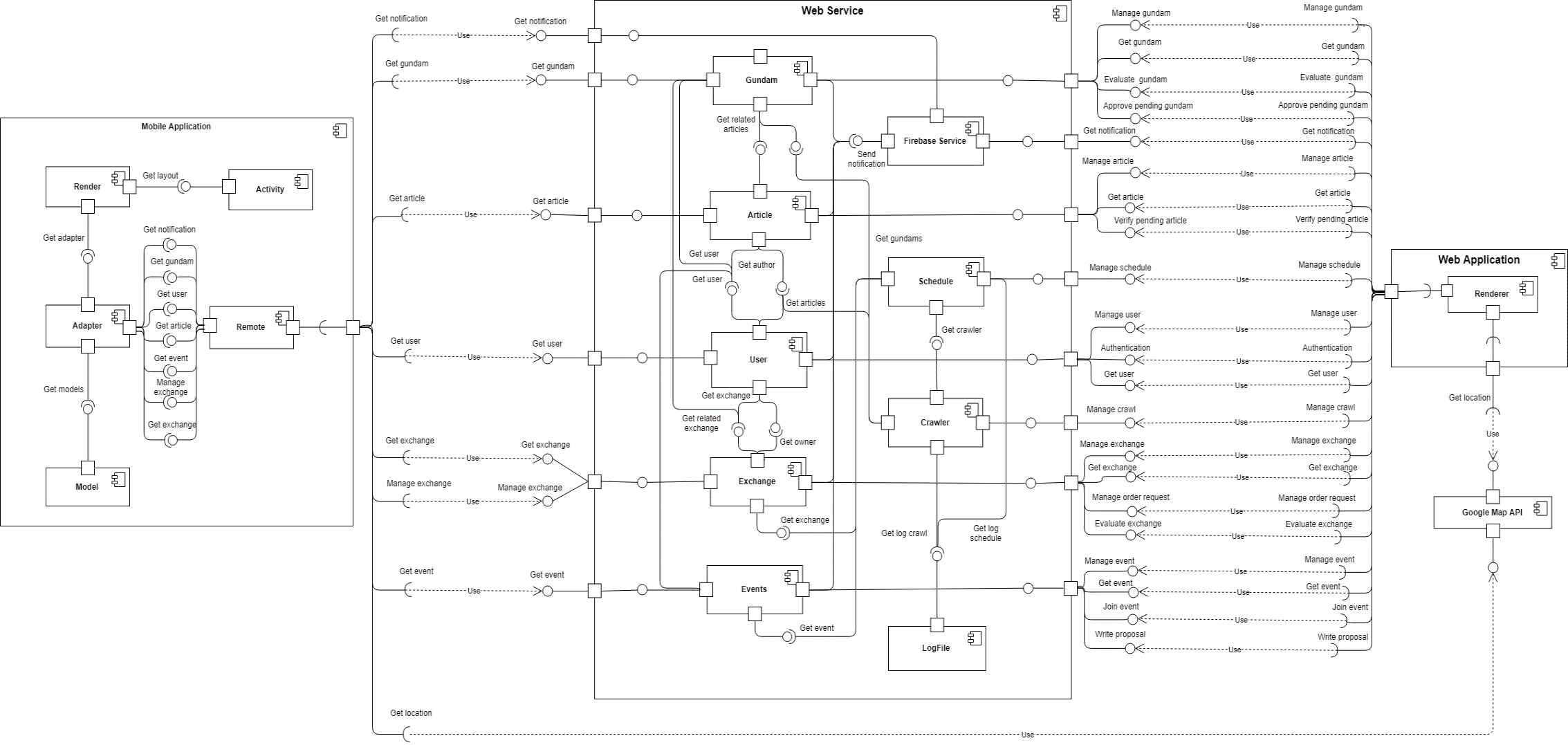
In Web Application, the system is developed under 3-layers architecture style, with Spring Boot technology. We choose this architecture for Web application because of the following advantages:

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

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

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

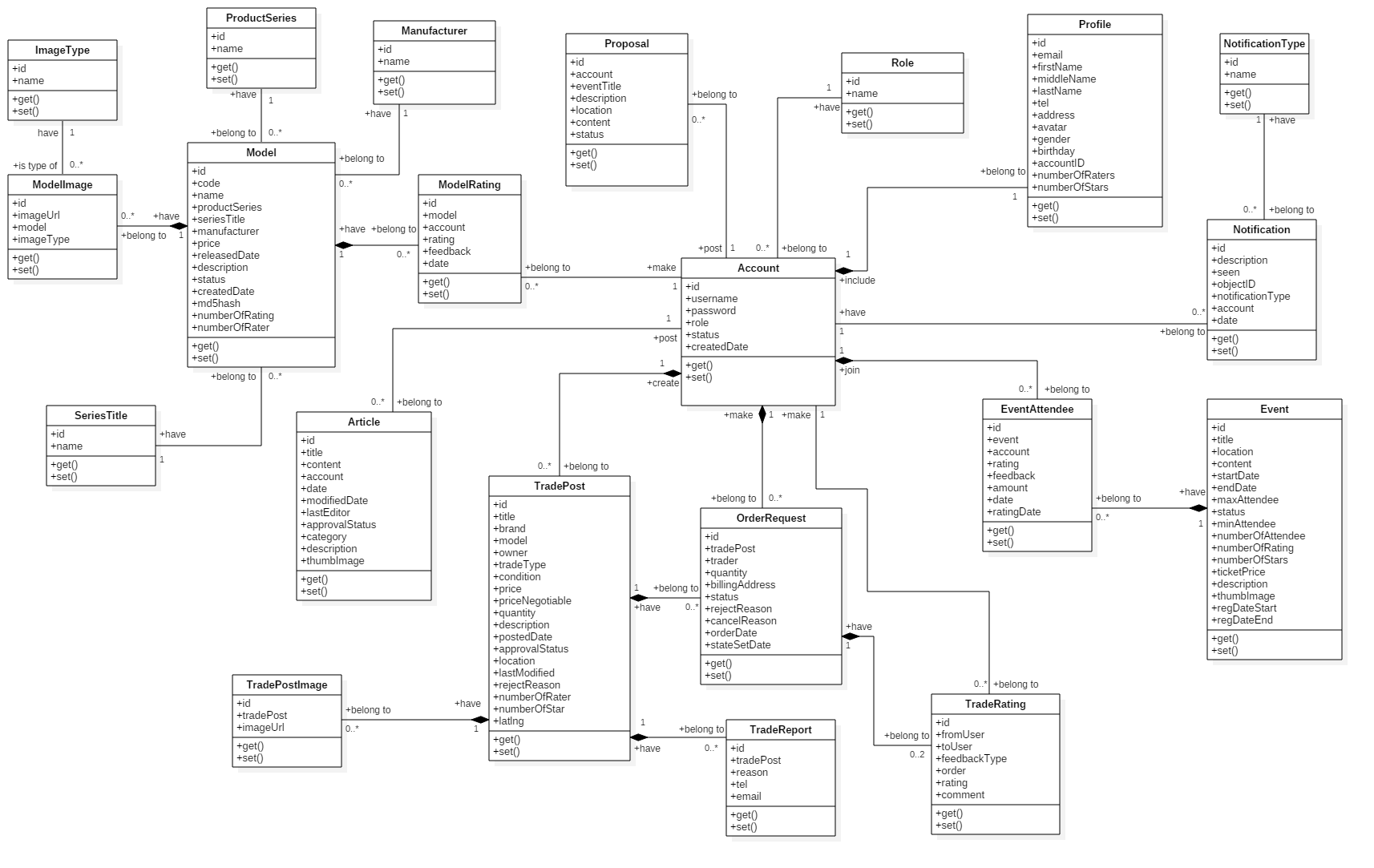
## 3. Component Diagram

****

|  |  |
| --- | --- |
| 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 |

## 4. Detailed Description

### 4.1. Class Diagram

****

|  |  |  |
| --- | --- | --- |
| **Class Name** | **Mapping column with Conceptual diagram** | **Description** |
| Model | Model | Contain model’s information |
| ModelImage | N/A | Not exist in conceptual diagram. It’s used to contain models’s images |
| ImageType | N/A | Not exist in conceptual diagram. It’s used to contain types of images |
| ProductSeries | N/A | Not exist in conceptual diagram. It’s used to contain product series information |
| SeriesTitle | N/A | Not exist in conceptual diagram. It’s used to contain series title information |
| Manufacturer | N/A | Not exist in conceptual diagram. It’s used to contain manufacturers information |
| 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 |

## 6. Database Design

### 6.1. Entity Relationship Diagram (ERD)

### 6.2. Data Dictionary

|  |  |
| --- | --- |
| **Entity Data Dictionary** | |
| **Entity name** | **Description** |
| Model | Contain model information |
| SeriesTitle | Contain series title information, this is a type of model’s category |
| ProductSeries | Contain product series information, this is a type of model’s category |
| Manufacturer | Contain model’s manufacturer information |
| ModelImage | Contain model’s images |
| ImageType | Contain model’s image type |
| 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 |

## 7. Algorithms

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

### 7.2. MD5 hash function matching new data

### 7.3. Track schedule progress

### 7.4. Search nearby location with Haversine formula