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

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

– TP.HCM, 09/2018 –

**Task Assignment:**

** :** Hoàng Ngô Minh Tùng

** :** Nguyễn Bảo Khánh

** :** Lê Nguyễn Ngọc Sang

**B. Report No.2 Software Project Management Plan**

1. **Problem Definition**

**1.1. Name of this Capstone Project**

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

**1.2. Problem Abstract**

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

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

**1.3. Project Overview**

**1.3.1. Current Situation**

Below are the problems encountered in this project:

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

**1.3.2. The Proposed System**

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

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

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

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

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

**1.3.2.1 Website**

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

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

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

**1.3.3. Boundaries of the System**

The system is designed to serve:

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

The system is designed with the following main functions:

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

The system is designed to run on these platforms:

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

**1.3.4. Future Plans**

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

* Gunpla players won’t have to go to different pages and groups to find information or to trade. With Gunpla World Application, everything is concentrated in one place. The user only needs to bookmark the website address or download the mobile application to access all the information and news they need on Gunpla, and all the functions they need to trade Gunpla with other players. Everything is combined on one application.
* Professional aspects:

+ Event host won’t have to worry about false information that may happen from spam or other kind of attacks. The system will have a fee system to defend against attacks, and it can provide important information like the number of attendees for an event so the company can manage that event effectively.

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

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

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

**1.3.5. Development Environment**

**1.3.5.1. Hardware requirements**

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

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

**1.3.5.2. Software requirements**

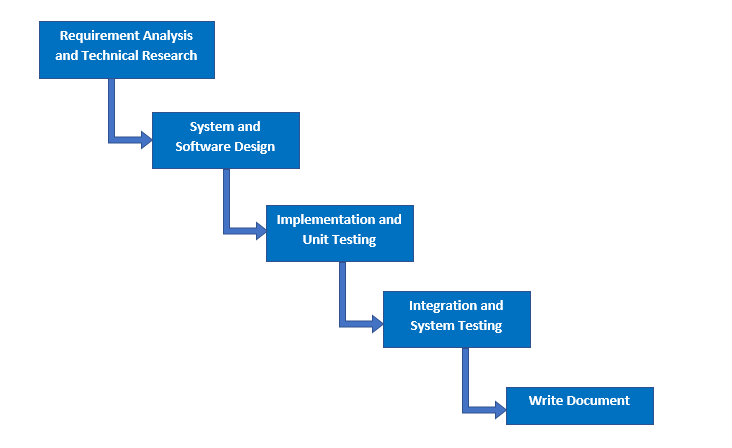
|  |  |  |
| --- | --- | --- |
| **Software** | **Name and version** | **Description** |
| **Operating System** | Window 10 1803 | Operating system and platform for development |
| **Environment** | Java EE 8, Android SDK APIs level 15 or above | Specification for developing web and android application |
| **Modeling Tool** | StarUML 2.8.1 | Used to implement website and web service |
| **IDE** | IntelliJ IDEA 2018.2.3, Netbeans 8.0.2, Android Studio 3.1.3 | Programming tools |
| **DBMS** | MySQL 5.7.20, MySQL Workbench 6.3.9 CE | Used to create & manage the database for system |
| **Source control** | SourceTree 2.6.10 | Used for source control |
| **Web browser** | Chrome 62 or above | Testing browser |

1. **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 * 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 * Review documents, reports * 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 * Netbeans 8.0.2 * Android Studio 3.1.3 |
| Source control | SourceTree 2.6.10 |
| Database Manager | MySQL Workbench 6.3.9 CE |

|  |  |
| --- | --- |
| **Technique** | **Name and version** |
| Front end | * HTML v5.0 * JQuery v3.2.1 * Bootstrap v3.3.7 * CSS * Javascript * Ajax * ReactJS |
| Back end | * Java EE 8 * Spring Boot v2.0.5 RELEASE * Hibernate v5.0.12 * Log4J v1.2.17 * JUnit v3.8.1 |
| Google Map API libraries | google maps |
| Web server | Apache Tomcat v8.5.15 |
| Database system | MySQL 5.7.20 |

1. **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) | - Software Requirement Specification  - Basic knowledge Google Maps API  - 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 <number>: 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** | Research on google maps api  Research on HTML5 Location, Android Location  Research on ReactJS, 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 location suggestion  Implement pagination  Implement data crawling | 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 Pagination and Location Suggestion algorithm  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.