**MINISTRY OF EDUCATION AND TRAINING**



**FPT UNIVERSITY**

Capstone Project Document

**Building Material C2B Website**

|  |  |
| --- | --- |
| **Nhóm số** | |
| **Group members** | Cao Minh Thúy Vy – SE61562 (Team Leader)  Đặng Minh Tùng – SE61703  Trần Anh Nguyên – SE61667  Nguyễn Đình Thiện – SE61446 |
| **Supervisor** | Lại Đức Hùng |
| **Ext. Supervisor** | N/A |
| **Capstone Project code** | Mã đề tài |

-Ho Chi Minh City, ***Ngày bắt đầu làm***-

*This page is intentionally left blank*

**Table of Contents**

Table of Contents ................................................................................................................................3

List of Tables........................................................................................................................................4

Definitions, Acronyms, and Abbreviations..........................................................................................6

A. Report No. 1 Introduction ...........................................................................................................7

1. Project Information .................................................................................................................7

2. Introduction ............................................................................................................................7

3. Current Situation.....................................................................................................................7

4. Problem Definition ..................................................................................................................7

5. Proposed Solution ...................................................................................................................8

5.1 Feature functions ............................................................................................................8

5.2 Advantages and disadvantages .......................................................................................8

6. Functional Requirements ........................................................................................................9

6.1 Name Card Management ................................................. **Error! Bookmark not defined.**

6.2 Event ................................................................................ **Error! Bookmark not defined.**

6.3 Searching .......................................................................... **Error! Bookmark not defined.**

6.4 Suggestion ........................................................................ **Error! Bookmark not defined.**

6.5 User Management ........................................................... **Error! Bookmark not defined.**

7. Role and Responsibility ...........................................................................................................9

**List of Tables**

Table 1: Roles and Responsibilities .....................................................................................................9

**List of Figures**

**Definitions, Acronyms, and Abbreviations**

|  |  |
| --- | --- |
| **Name** | **Definition** |
| BMW | Building Material C2B Website |
| C2B | Customer to Business |
| GUI | Graphic User Interface |

**A. Report No. 1 Introduction**

**1. Project Information**

- Project name: **Building Material C2B Website**

- Project Code: **BMW**

- Product Type: **Web Application**

- Start Date: **03/02/2017**

- End Date:

**2. Introduction**

Nowadays, the demand of building a house is rising. Homeowners are looking forward to find the suitable material supplier. Together with the rapid growth of information technology, especially in e-commerce field. The fact that many e-commerce websites are not really meet the customer’s need. Therefore, we hope to create for our customer a new experience in shopping online based on C2B. In C2B model, sometimes known as Consumer to Business, is a business model where consumers create products and services which are consumed by businesses and organization. According to our research and analysis, we introduce a new solution for building material C2B website.

We build a system which help customers to find more suppliers with the most reasonable price. Suppliers also find their own potential customers. Moreover, building material shops can also improve their revenues by helping customer to find the suitable material.

**3. Current Situation**

Recently, when homeowner is looking forward to building a house, they normally choose traditional way that go directly to the material shop

In VietNam, there aren’t any building material website that based on C2B now. However, around the world, especially in America and India, there are many websites that offer this type of transaction, such as: priceline.com, msupply.com…

**4. Problem Definition**

In VietNam, there aren’t any building material website that based on C2B now. When shopping online, customer must combine with other B2C building material website.

**5. Proposed Solution**

Our solution is to build a new website named “Building Material C2B” to solve the current problem. The website supports the customers find suppliers with the most reasonable price. In addition, our solution is also help supplier find more customers.

BMW includes a web application with following functions:

**5.1 Feature functions**

* **Web application:**
* For customer:
* Posting request for find material, suppliers to build a house.
* Selecting suitable supplier via reverse auction.
* For supplier:
* Bidding: bid to find the suitable orders for their own shop.

**5.2 Advantages and disadvantages**

- Advantages:

* Bring new experience about buy building material.
* Help customer to save time and money.
* Help customer to find their own supplier.
* Help supplier to have more customers, orders.
* Support pre-order/ e-payment via nganluong.vn.

- Disadvantages:

* C2B website is uncommon for customer.
* Supplier sell material at price equal or lower than market price.

**6. Functional Requirements**

* For guest:
  + Register, login.
  + View shop.
* For customer:
* Manage auction: create, cancel, post request.
* Edit profile, send feedback.
* View shops, orders, auctions…
* Search/Filter shops, orders, auctions.
* Chat with supplier.
* Review shop: make a review.
* Logout.
* For supplier:
* Bidding: place a bid, retract…
* Manage shop: create, edit shop, add/edit/delete product.
* Chat with customer.
* View shops, orders, auctions…
* Search/Filter orders, auctions, products, categories, others shops.
* Send feedback.
* For admin:
* Manage all accounts in the system: create, edit, delete.
* Manage all categories, products: create, edit, delete.
* Active/ De-active user.

**7. Role and Responsibility**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No** | **Full Name** | **Role** | **Position** | **Contact** |
| 1 | Lại Đức Hùng | Project Manager | Supervisor | [hungld@fpt.edu.vn](mailto:hungld@fpt.edu.vn) |
| 2 | Cao Minh Thúy Vy | Developer | Leader | [vycmtse61562@fpt.edu.vn](mailto:vycmtse61562@fpt.edu.vn) |
| 3 | Đặng Minh Tùng | Developer | Member | tungdmse61703[@fpt.edu.vn](mailto:phucnhse60749@fpt.edu.vn) |
| 4 | Trần Anh Nguyên | Developer | Member | [nguyentase61667@fpt.edu.vn](mailto:nguyentase61667@fpt.edu.vn) |
| 5 | Nguyễn Đình Thiện | Developer | Member | [thienndse61446@fpt.edu.vn](mailto:thienndse61446@fpt.edu.vn) |

**Table 1: Roles and Responsibilities**

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

**1. Problem Definition**

* 1. **Name of this Capstone Project**
* Official name: **Building Material C2B Website**
* Vietnamese name: **Website** **C2B mua hàng vật liệu xây dựng**
* Abbreviation: **BMW**

**1.2 Problem Abstract**

We build the system to support customer to buy building material with reasonable price. Our system also support supplier to sell more materials. For that reason we create a C2B website that achieve their requirements. In order to satisfy customer’s demand, we help them create plans. Customer can also make a review about the shops. For supplier, we help them manage their online shop. More than that, we also provide e-payment through Nganluong.vn.

Our system use Google API technology, this technology is quite useful to customers. They can check which shop is the nearest to their location.

**1.3 Project Overview**

**1.3.1 Current Situation**

Below are the problems encountered in this system.

* **Disadvantages:**
* Customer’s habit: customers are used to buy material at material shop when looking forward to building a house.
* System’s security: system allow customer to cancel bid, this function may become a target for cheating.
* E-commerce’s model: C2B model is unfamiliar with customer.
* Require enormous data about construction.

**1.3.2 The Proposed System**

After doing research on technology, we choose Google API because this technology is useful in determine the location. The basic idea is to use Google API to check how far from the supplier’s shop to the customer’s location.

In task assignment, we assign to member using vertical model 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.

BMW is built as a web based application. It is high availability (24/7) and fast responds with real-time function.

**1.3.2.1 BMW Web Application**

Our web application includes of three parts:

* For customer:
* Manage auction: create, cancel, post request.
* Edit profile, send feedback.
* View shops, orders, auctions…
* Search/Filter shops, orders, auctions.
* Chat with supplier.
* Review shop: make a review.
* Logout.
* For supplier:
* Bidding: place a bid, retract…
* Manage shop: create, edit shop, add/edit/delete product.
* Chat with customer.
* View shops, orders, auctions…
* Search/Filter orders, auctions, products, categories, others shops.
* Send feedback.
* For admin:
* Manage all accounts in the system: create, edit, delete.
* Manage all categories, products: create, edit, delete.
* Active/De-active user.

**1.3.3 Boundaries of the System**

* Our main target is helping customer to shopping with more convenient and efficient in HaNoi and HoChiMinh city
* Language of system is VietNamese
* The completed product includes:

+ Website application

**1.3.4 Future Plans**

Currently, the system only support in Hanoi and Ho Chi Minh City. In further development, the sytem can:

* Expand location in all province of Vietnam
* Support group buying function.
* Deploy the system in multiple platform (IOS, Android)
* Provide more kind of auction such as: English auction, Dutch auction

**1.3.5 Development Environment**

**1.3.5.1 Hardware requirements**

* For web application server

|  |  |  |
| --- | --- | --- |
| Windows | Minimum Requirements | Recommended |
| Internet Connection | Cable, Wi-Fi (4 Mbps) | Cable, Wi-Fi (8 Mbps) |
| Operating System | Window Server 2008 | Window Server 2008 |
| Computer Processor | Intel® Core i7 2.4 GHz | Intel® Core i7 2.4 GHz |
| Computer Memory | 2GB of RAM | 4GB of RAM or more |

**Table 2: Hardware Requirement for Server**

**1.3.5.2 Software requirements**

|  |  |  |
| --- | --- | --- |
| Software | Name / Version | Description |
| Operating system | Window 10 64 bit | Operating system and platform for development |
| Environment | .NET Framework 4.5 | Specification for developing web application |
| IDE | Visual Studio 2015 | Used for implement website |
| Design Model tool | StartUML v2.5.1 | Used for creating modal and diagrams. |
| DBMS | Microsoft SQL Server 2012 | Used to create & manage the database for system |
| Document storage | Github | Used for storing document |
| Store and manage source code | Github & SourceTree | Used to store all source code |

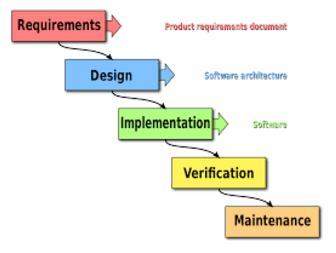
**Table 3: Software Requirement**

**2. Project organization**

**2.1 Software Process Model**

The project is developed under waterfall model. The waterfall model is a [sequential](https://en.wikipedia.org/wiki/Sequence) (non-iterative) [design](https://en.wikipedia.org/wiki/Design) process, used in software development process, in which progress is seen as flowing steadily downwards (like a [waterfall](https://en.wikipedia.org/wiki/Waterfall)) through the phases of conception, initiation, analysis, design, construction, testing, production/implementation and maintenance.

The waterfall model illustrates the software development process in a linear sequential flow; hence it is also referred to as a linear-sequential life cycle model. This means that any phase in the development process begins only if the previous phase is complete.



**Figure 1: Modified Waterfall Model**

We use Waterfall Development Model for our project development because:

* We need to have output documents following school’s schedule.
* Business is clear and can be implemented.
* If the requirement change or having trouble when designing, implemenating, testing, you can roll back and change the documents.

Reference: <http://www.waterfall-model.com/sashimi-waterfall-model/>

**2.2 Roles and responsibilities**

|  |  |  |  |
| --- | --- | --- | --- |
| No | Full name | Role in Group | Responsibilities |
| 1 | Lại Đức Hùng | Supervisor, Project Manager | * Specify user requirements * Specify business * Control the development process * Give out technique and business analysis support |
| 2 | Cao Minh Thúy Vy | Team leader, B.A, Developer, Tester | * Managing process * Designing database * Clarifying requirements * Prepare documents * GUI design * Create test plan * Coding * Testing |
| 3 | Đặng Minh Tùng | Team member,  B.A, Developer,  Tester | * Designing database * Clarifying requirements * Prepare documents * GUI design * Create test plan * Coding * Test |
| 4 | Trần Anh Nguyên | Team member,  B.A, Developer,  Tester | * Designing database * Clarifying requirements * Prepare documents * Create test plan * Coding * Test |
| 5 | Nguyễn Đình Thiện | Team member,  B.A, Developer,  Tester | * Designing database * Clarifying requirements * Prepare documents * Create test plan * Coding * Test |

|  |  |
| --- | --- |
| Tool | Name / version |
| Web server | IIS |
| Development tool | Visual Studio 2015 |
| DBMS | SQL Server 2012 |
| Source control | GitHub & SourceTree |
| Modeling tool | StarUML 2.7.1 |
| Document tool | Microsoft Word 2013, Microsoft Excel 2013 |

**Table 4: Roles and Responsibilities Details**

**2.3 Tools and Technique**

**Table 5: Tool List**

|  |  |
| --- | --- |
| Technique | Name / version |
| Frontend | HTML5, CSS3, JavaScript, jQuery, Ajax |
| Backend | ASP.Net MVC 5 |

**Table 6: Technique List**

**3. Project Management Plan**

**3.1 Software development life cycle**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Phase | Description | Deliverables | Resouces needed | Dependencies and Constraints | Risks |
| Requirement Analysis | - Collect requirements  - Identify and clarify overall requirements | - Introduction of proposed system  - Software requirement specification  - Project task plan | 20 man – days | N/A | - Unclear project scope, business process  - Lack of sharing understand between members |
| Design | - Create architecture design  - Create detail design by using top-down approach  - Choose architecture style | - Software design documents  - Base code structure | 20 man – days | Requirement Analysis | - Lack of experience  - Not fulfill requirement |
| Implementation | - Code system core functions and other features with GUI  - Create unit test | - Main functions in website | 50 man – days | Design | - Lack of specialized knowledge  - Lack of experience |
| Testing | - Create integration test  - Create alpha test  - Create acceptance test  - Correct bugs | - Test specification | 20 man – days | Implementation | - Lack of experience  - Missing test case |
| Maintenance | - Deploy on web server | - Installation guide  - User mannual | 10 man – days | Testing | - Lack of experience |

**Table 7: Software Development Life Cycle Detail**

**3.2 Phase Detail**

**3.2.1 Phase 1: Requirement Analysis**

|  |  |  |
| --- | --- | --- |
| Task | Description | Author |
| 1. Collect requirements | Define requirement  Find similar website, their strength and weekness | VyCMT, TungDM, NguyenTA, ThienND |
| 1. Identify and clarify main functions | Define main functions that system provide | VyCMT, NguyenTA, ThienND |
| 1. Create system introduction | Complete Introduction Report | VyCMT, TungDM, ThienND |
| 1. Software Management Plan | Prepare Project Management Plan | VyCMT, TungDM, NguyenTA |
| 1. SRS | Create SRS document | VyCMT, ThienND |

**Table 8: Phase 1: Requirement Analysis**

**3.2.2 Phase 2: Design**

|  |  |  |
| --- | --- | --- |
| Task | Description | Author |
| 1. Detail design | Compare new document with existed documents of system | VyCMT, TungDM, NguyenTA, ThienND |
| 1. Database design | Based on parse data to recommendation  Based on other needs to recommendation | TungDM, NguyenTA |
| 1. Technology Research | Study Google API | TungDM, NguyenTA |
| 1. Design document | Create software design document | VyCMT, ThienND |

**Table 9 - Phase 2: Design**

**3.2.3 Phase 3: Implementation**

|  |  |  |
| --- | --- | --- |
| Task | Description | Author |
| 1. Front-end web function | Implement front-end web functions | VyCMT, TungDM, NguyenTA, ThienND |
| 1. Back-end web function | Implement back-end web functions | VyCMT, TungDM, NguyenTA, ThienND |
| 1. Unit testing | Write test case and testing for web functions | VyCMT, TungDM, NguyenTA, ThienND |

**Table 10 - Phase 3: Implementation**

**3.2.4 Phase 4: Test**

|  |  |  |
| --- | --- | --- |
| Task | Description | Author |
| 1. Integration testing | Write test case and testing modules | VyCMT, ThienND |
| 1. Alpha testing | Testing whole system to find bugs that can not be found through other testing | VyCMT, TungDM, NguyenTA, ThienND |

**Table 11 - Phase 4: Test**

* + 1. **Phase 5: Maintenance**

|  |  |  |
| --- | --- | --- |
| Task | Description | Author |
| 1. Installation guide | Write installation guide | VyCMT, TungDM, NguyenTA, ThienND |
| 1. User mannual | Write user mannual | VyCMT, TungDM, NguyenTA, ThienND |

**Table 12 - Phase 5: Maintenance**

**3.3 All Meeting Minutes**

* [https://github.com/Capstone-JS/Documents/Meeting minutes](https://github.com/Capstone-JS/Documents/Meeting%20minutes) (Security: Must be a member of GitHub Repository)

**4. Coding Convention**

* C#: Using to develop website
* Naming Convention:
  + For variable’s name, use Camel Case. Eg: minValue, maxValue…
  + For function name, class name, use Pascal Case. Eg: AddIncome, AddExpense…
* Layout Convention:
  + Indent continuation one tab stop (four spaces).
  + Write only one statement, one declaration per line
  + Add at least one blank line between method definitions and property definitions.
  + Use parentheses to make clauses in an expression apparent.
* 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

+ End comment text with a period

+ Insert one space between comment delimiter (//) and comment text

References:

**C# Coding Conventions (C# Programming Guide)**

Update: July 20, 2015

<https://msdn.microsoft.com/en-us/library/ff926074.aspx>

**C. Report No. 3 Software Requirement Specification**

**1. User Requirement Specification**

* 1. **Guest Requirement**

Guest is a person who does not have access to the system. Guest can use some functions in the system. To use all functions, guest must login. These are some functions that guest can use:

* Register.
* Login*.*
* View shop
  1. **User Requirement**

User is a person who uses their account to login to the system. User include customer and supplier. These are some additional functions that user can use:

* Edit profile
* Send feedback
* View shop
* Logout
  1. **Customer Requirement**

Customer is a person who wants to buy material for their house. Customer account must be actived by admin. These are some funtions that customer can use:

* Manage request
* View order
  1. **Supplier Requirement**

Supplier is a person who wants to sell material to customer. Supplier account must be active by admin. These are some funtions that supplier can use:

* Manage shop
* View bid
* View customer’s order
* Reply request
  1. **Admin Requirement**

Admin is a person who manages whole system. These are some funtions that admin can use

* Manage system’s product
* Manage system’s category
* Manage user
* Manage role

**2. System Requirement Specification**

**2.1 External Interface Requirement**

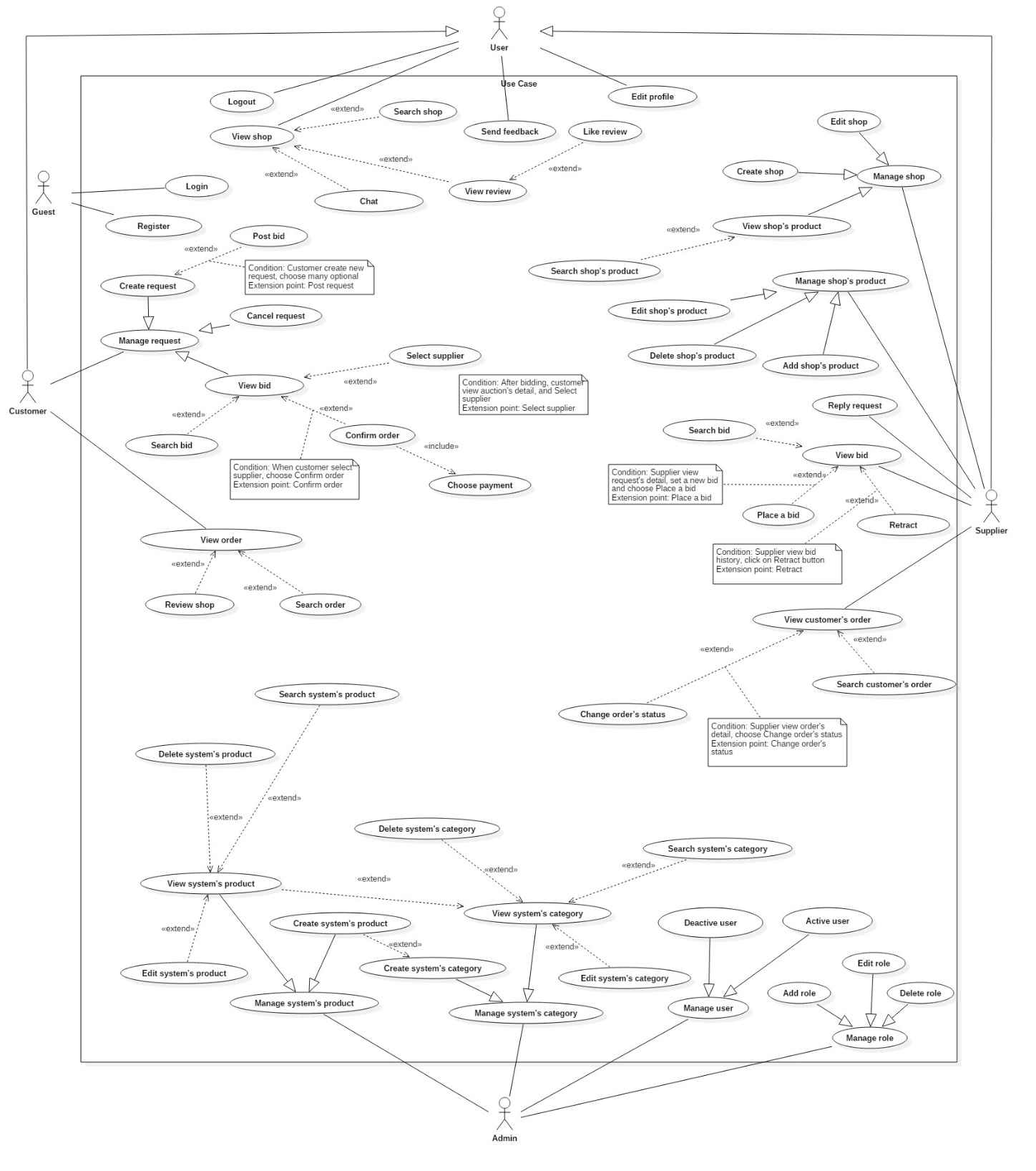
* + 1. **User Interface**
* General requirement for graphics user interface is the GUI should be simple, clear and intuitive
* The user interface uses Vietnamese language in web application
* The user interface displays best on 1366x768 resolutions on desktop
  + 1. **Hardware Interface**
* Desktop:
  + RAM: 4 GB
  + CPU: 2.5Ghz

**2.1.3 Software Interface**

* Web application: Works with Firefox (v30 or later), Google Chrome (v39 or later), Cốc Cốc (v39 or later) web browser and must supports HTML5, CSS3 and JavaScript.
* The screen must bigger than 1024 x 768 to browse the completely part of document.
  + 1. **Communication Protocol**

- Use HTTP 1.1 to communicate between the web browser and the web server

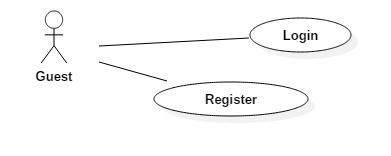
**2.2 System Overview Use Case**



**Figure 2: System Overview Use Case**

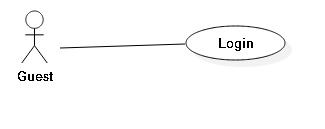
**2.3 List of Use Case**

**2.3.1 <Guest>Overview Use Case**



**Figure 3: <Guest> Overview Use Case**

**2.3.1.1 <Guest> Login**

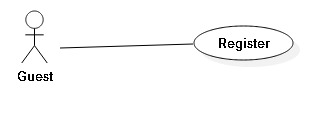


**Figure 4: <Guest>Login**

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE – BMW01** | | | |
| **Use Case No.** | BMW01 | **Use Case Version** | 1.0 |
| **Use Case Name** | Login | | |
| **Author** | VyCMT | | |
| **Date** | February 06, 2017 | **Priority** | Normal |
| **Actor:**   * Guest   **Summary:**   * This use case allows guest to login to the system and gain access to their role.   **Goal:**   * Guest can login to the system.   **Triggers:**   * Guest enter email and password in the login screen * Guest sends the Login command.   **Preconditions:**   * Email and password must existed in the database.   **Post Conditions:**   * **Success:** Guest is login to the system. * **Fail:** Show error messasge   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | In home page, guest click on Login link | System show Login Screen with 2 areas:   * Login normal * Login using Facebook   [Alternative 1] | | 2 | Guest choose Login normal | Login Screen appear with the login form   * Email: free text input * Password: free password input * Login button | | 3 | Guest enter email and password |  | | 4 | Guest click Login button | System check valid input in email and password text input  System check email and verify authorization for account  System redirect to role’s view  [Exception 1, 2] |   **Alternative Scenario:**   |  |  |  | | --- | --- | --- | | No | Actor action | System Response | | 1 | Guest choose Login using Facebook |  |   **Exception:**   |  |  |  | | --- | --- | --- | | No | Actor action | System Response | | 1 | Guest enter wrong email or password | System shows message “Invalid email or password!” | | 2 | Guest do not enter some required fields | System shows message “Field name must be entered” |   **Relationships: N/A**  **Business Rules:**   * When guest send Login command, System authenticate by checking email or password. * Guest enter password in the hidden text field and password must be encrypted before sending to server. * After logged in to system, guest will be redirected to a specific view based on their role on the system: Customer or Supplier   + If the role is “Customer”, the system will display Customer view.  + If the role is “Supplier”, the system will display Supplier view.  + If the role is “Admin”, the system will display Administrator view. | | | |

**Table 2: <Guest>Login**

**2.3.1.2 <Guest>Register**

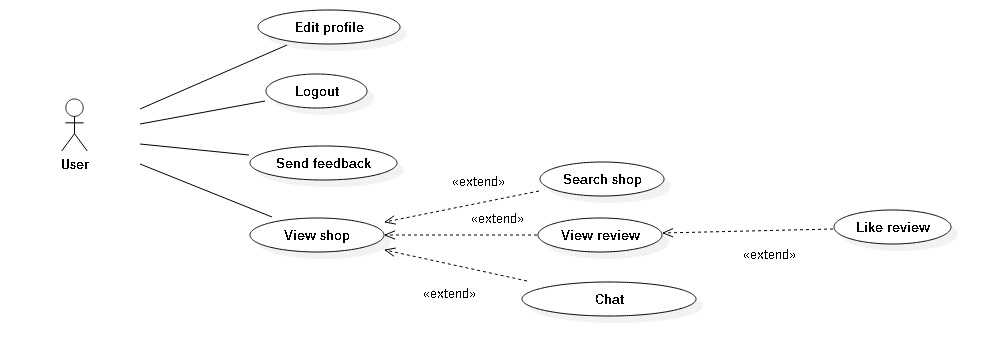
****

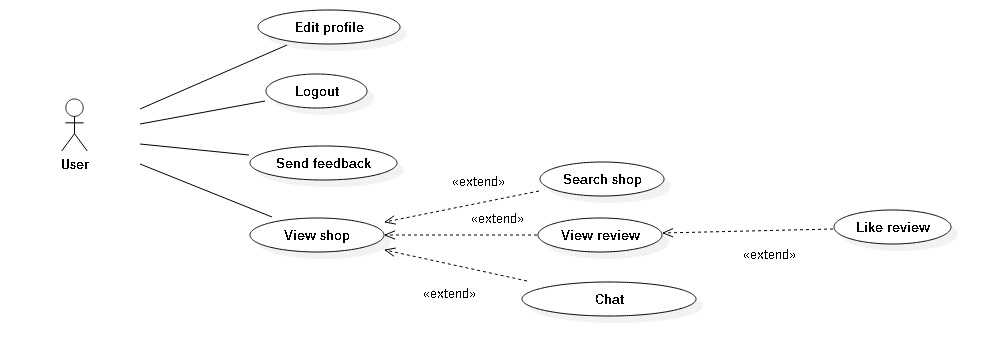
**Figure 5: <Guest> Register**

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE – BMW02** | | | |
| **Use Case No.** | BMW02 | **Use Case Version** | 1.0 |
| **Use Case Name** | Register | | |
| **Author** | VyCMT | | |
| **Date** | February 06, 2017 | **Priority** | Normal |
| **Actor:**   * Guest   **Summary:**   * This use case allows guest to register a new account.   **Goal:**   * Guest can register new account according to their roles.   **Triggers:**   * Guest sends the Register command.   **Preconditions:** N/A  **Post Conditions:**   * **Success:** New account has been added to system * **Fail:** Show error messasge   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | Guest click on Register link in Home Page | System display drop-down list with 2 options:   * Customer * Supplier | | 2 | Guest choose any options and goes to Register Screen | System show Register Screen with 2 areas:   * Register normal * Register using Facebook   [Alternative 1] | | 3 | Guest choose Register normal | Register Screen appear with the register form   * Email: free text input * Username: free text input * Password: free password input * Confirm Password: free password input * Submit button | | 4 | Guest enter the following information |  | | 5 | Guest click on Submit button | System check valid input in email, password and confirm password  [Exception 1, 2, 3, 4]  System send confirmation email | | 6 | Guest check mails and click on Activation link | System confirm and redirect to home page |   **Alternative Scenario:**   |  |  |  | | --- | --- | --- | | No | Actor action | System Response | | 1 | Guest choose Register using Facebook |  |   **Exception:**   |  |  |  | | --- | --- | --- | | No | Actor action | System Response | | 1 | Guest enter invalid email | System shows message “Invalid email” | | 2 | Guest enter password that different from confirm password | System shows message “Password and confirm password do not match” | | 3 | Guest do not enter some required fields | System shows message “Field name must be entered” | | 4 | Guest register with the same email | System shows message “Email is existed in system” |   **Relationships: N/A**  **Business Rules: N/A** | | | |

**Table 3: <Guest>Register**

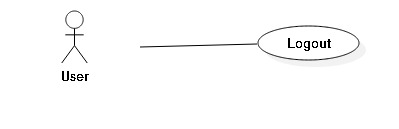
**2.3.2 <User>Overview Use Case**

****

****

**Figure 6: <User>Overview Use Case**

**2.3.2.1 <User>Logout**

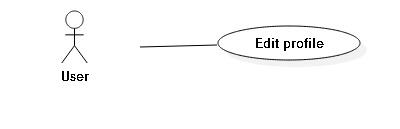
****

**Figure 7: <User>Logout**

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE – BMW03** | | | |
| **Use Case No.** | BMW03 | **Use Case Version** | 1.0 |
| **Use Case Name** | Logout | | |
| **Author** | VyCMT | | |
| **Date** | February 06, 2017 | **Priority** | Normal |
| **Actor:**   * User   **Summary:**   * This use case allows user to log out of the system.   **Goal:**   * System remove user’s session   **Triggers:**   * User send Logout command   **Preconditions:**   * User already logged in the system with role “Customer” or “Supplier”   **Post Conditions:**   * **Success:** User’s session was removed * **Fail: N/A**   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | User send Logout comand | Check if user’s session existed  Remove current session of specified user  Redirect to home page |   **Alternative Scenario: N/A**  **Exception: N/A**  **Relationships: N/A**  **Business Rules:**   * After log out of system, user is no longer has permission to access any system’s function and need to log in again to use it. | | | |

**Table 4: <User>Logout**

**2.3.2.2 <User>Edit Profile**

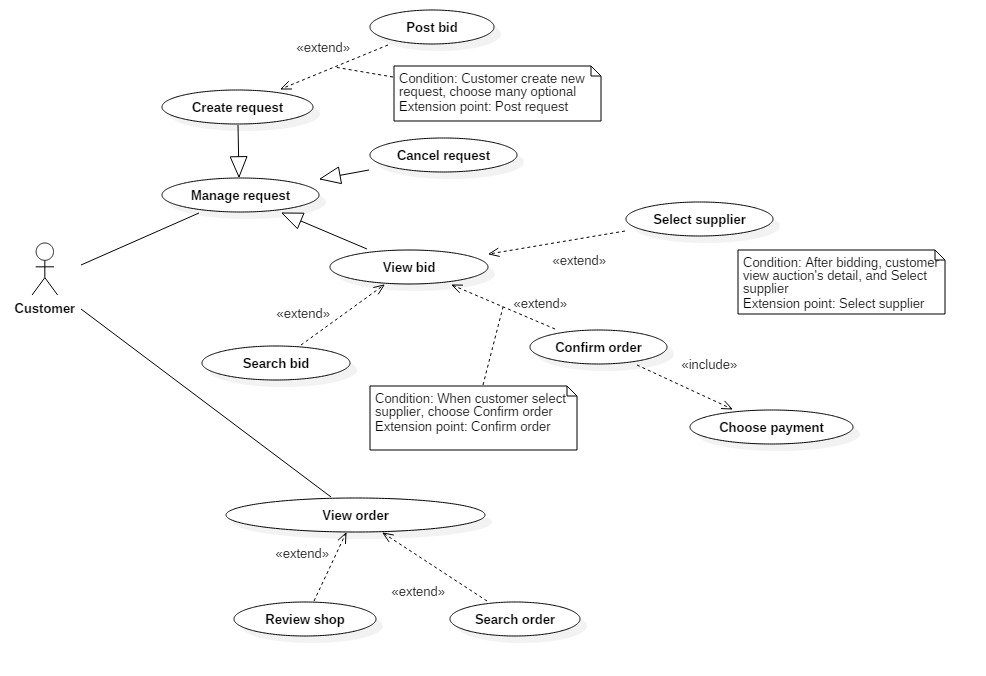
****

**Figure 8: <User>Edit Profile**

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE – BMW04** | | | |
| **Use Case No.** | BMW04 | **Use Case Version** | 1.0 |
| **Use Case Name** | Edit Profile | | |
| **Author** | VyCMT | | |
| **Date** | February 06, 2017 | **Priority** | Normal |
| **Actor:**   * User   **Summary:**   * This use case allows user to edit their personal information.   **Goal:**   * User’s personal information is updated when they want to change.   **Triggers:**   * User sends the Edit Profile command.   **Preconditions:**   * User must have logged in   **Post Conditions:**   * **Success:** User’s personal information is updated in database * **Fail:** Show error message   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | User click on Edit Profile link | System show Edit Profile Screen:   * Address: free text input * Full Name: free text input * Password: free password input, length 6 – 50 * Update button | | 2 | User update the following information |  | | 3 | User click on Update button | System validate input then update to profile  System display successful message “Update successfully”  [Exception 1] |   **Alternative Scenario: N/A**  **Exception:**   |  |  |  | | --- | --- | --- | | No | Actor action | System Response | | 1 | User enter password field with wrong format | System displays error message: “Password length from 6 – 50 characters” |   **Relationships: N/A**  **Business Rules:**   * User update account profile in case of changing full name or password * Password must be encrypted before saving to the system. * User cannot update email. | | | |

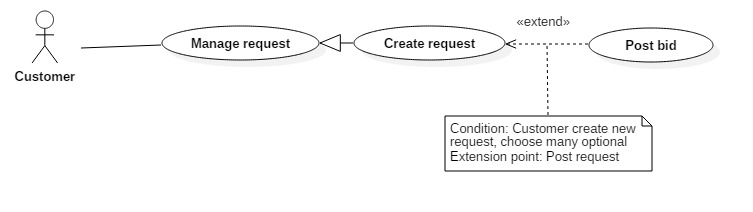
**Table 5: <User>Edit Profile**

**2.3.3 <Customer>Overview Use Case**



**Figure 9: <Customer>Overview Use Case**

**2.3.3.1 <Customer>Create Request**

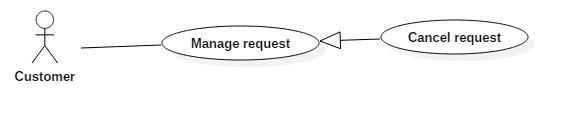


**Figure 10: <Customer>Create Request**

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE – BMW0** | | | |
| **Use Case No.** | BMW0 | **Use Case Version** | 1.0 |
| **Use Case Name** | Create Request | | |
| **Author** | ThienND | | |
| **Date** | February 07, 2017 | **Priority** | Normal |
| **Actor:**   * Customer   **Summary:**   * This use case allows Customer to create request.   **Goal:**   * Customer adds shop information into system.   **Triggers:**   * Customer sends the Create Request command.   **Preconditions:**   * User already logged in Customer role   **Post Conditions:**   * **Success:** New request has been added to storage. * **Fail:** Show error message   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | Supplier goes to Manage Request Screen and click on Create Request | System show Create Request form to enter new product:   * Customer Price: number * Delivery Address: free text input * Delivery Date: date time * Start Date: date time * Due Date: date time * Payment Type: free text input * Name: free text input * Create button | | 2 | Customer enter the following information |  | | 3 | Customer click on “Create” button | System inserts request into storage  [Exception 1] |   **Alternative Scenario: N/A**  **Exception:**   |  |  |  | | --- | --- | --- | | No | Actor action | System Response | | 1 | Customer do not enter some require fields | System shows message “Field name must be entered” |   **Relationships: N/A**  **Business Rules: N/A** | | | |

**Table 6: <Customer>Create Request**

**2.3.3.2 <Customer>Cancel Request**

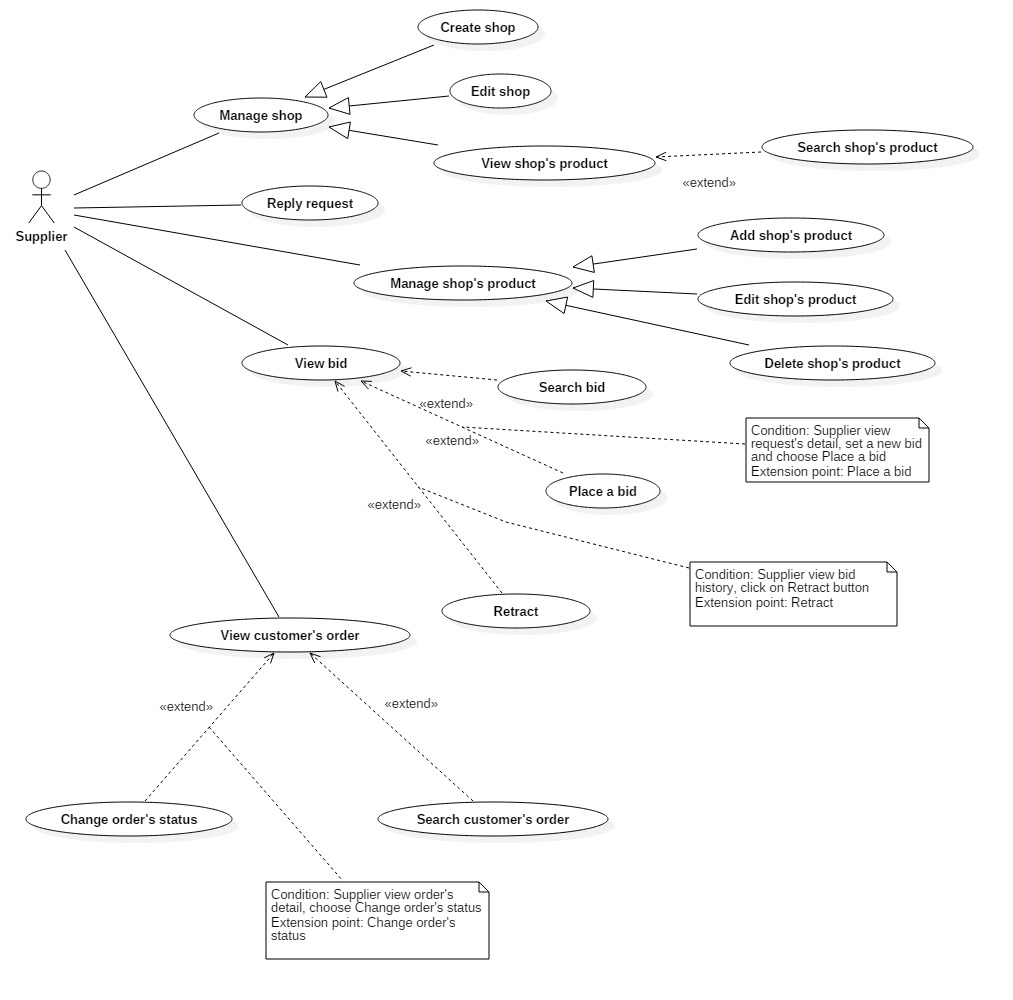


**Figure 11: <Customer>Cancel Request**

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE – BMW0** | | | |
| **Use Case No.** | BMW0 | **Use Case Version** | 1.0 |
| **Use Case Name** | Cancel Request | | |
| **Author** | ThienND | | |
| **Date** | February 07, 2017 | **Priority** | Normal |
| **Actor:**   * Customer   **Summary:**   * This use case allows Customer to cancel request.   **Goal:**   * Customer won’t see the requests in page anymore..   **Triggers:**   * Customer sends the Cancel Request command.   **Preconditions:**   * User already logged in Customer role * Request has been created and available in storage.   **Post Conditions:**   * **Success:** Request has been deleted from storage. * **Fail:** Show error message   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | Supplier goes to Manage Request Screen and click on Cancel Request | System show Cancel Request form to ensure that you want to cancel the request. | | 3 | Customer click on “Cancel” button | System deletes request from storage |   **Alternative Scenario: N/A**  **Exception: N/A**  **Relationships: N/A**  **Business Rules: N/A** | | | |

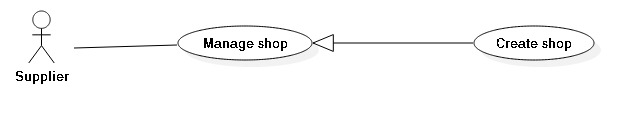
**Table 7: <Customer>Cancle Request**

**2.3.4 <Supplier>Overview Use Case**



**Figure 12: <Supplier>Overview Use Case**

**2.3.4.1 <Supplier>Create Shop**



**Figure 13: <Supplier>Create Shop**

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE – BMW0** | | | |
| **Use Case No.** | BMW0 | **Use Case Version** | 1.0 |
| **Use Case Name** | Create Shop | | |
| **Author** | ThienND | | |
| **Date** | February 07, 2017 | **Priority** | Normal |
| **Actor:**   * Supplier   **Summary:**   * This use case allows Supplier to create shop.   **Goal:**   * Supplier adds shop information into system.   **Triggers:**   * Supplier sends the Create Shop command.   **Preconditions:**   * User already logged in Supplier role   **Post Conditions:**   * **Success:** New shop has been added to storage. * **Fail:** Show error message   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | Supplier goes to Manage Shop Screen and click on Create Shop | System show Create Shop form to enter new product:   * Address: free text input * Create button | | 2 | Supplier enter the following information |  | | 3 | Supplier click on “Create” button | System inserts product into storage  [Exception 1] |   **Alternative Scenario: N/A**  **Exception:**   |  |  |  | | --- | --- | --- | | No | Actor action | System Response | | 1 | Supplier do not enter some require fields | System shows message “Field name must be entered” |   **Relationships: N/A**  **Business Rules: N/A** | | | |

**Table 8: <Supplier>Create Shop**

**2.3.4.2 <Supplier>Edit Shop**



**Figure 13: <Supplier>Edit Shop**

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE – BMW0** | | | |
| **Use Case No.** | BMW0 | **Use Case Version** | 1.0 |
| **Use Case Name** | Edit Shop | | |
| **Author** | ThienND | | |
| **Date** | February 07, 2017 | **Priority** | Normal |
| **Actor:**   * Supplier   **Summary:**   * This use case allows Supplier to edit shop.   **Goal:**   * Supplier edits shop information into system.   **Triggers:**   * Supplier sends the Edit Shop command.   **Preconditions:**   * User already logged in Supplier role   **Post Conditions:**   * **Success:** New shop has been updated to storage. * **Fail:** Show error message   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | Supplier goes to Manage Shop Screen and click on Edit Shop | System show Edit Shop form to enter new product:   * Address: free text input * Edit button | | 2 | Supplier enter the following information |  | | 3 | Supplier click on “Edit” button | System updated product into storage  [Exception 1] |   **Alternative Scenario: N/A**  **Exception:**   |  |  |  | | --- | --- | --- | | No | Actor action | System Response | | 1 | Supplier do not enter some require fields | System shows message “Field name must be entered” |   **Relationships: N/A**  **Business Rules: N/A** | | | |

**Table 9: <Supplier>Edit Shop**

**2.3.5.2 <Supplier>View Shop’s Product**



**Figure 14: <Supplier>View Shop’s Product**

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE – BMW0** | | | |
| **Use Case No.** | BMW0 | **Use Case Version** | 1.0 |
| **Use Case Name** | View Shop’s Product | | |
| **Author** | ThienND | | |
| **Date** | February 07, 2017 | **Priority** | Normal |
| **Actor:**   * Supplier   **Summary:**   * This use case allows Supplier to view list of products.   **Goal:**   * Supplier can view all products.   **Triggers:**   * Supplier sends the View Shop’s Product command.   **Preconditions:**   * User already logged in Supplier role   **Post Conditions:**   * **Success:** System show all products. * **Fail:** Show error message   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | Supplier send command “View Shop’s Product” | System get and show product list. Each product includes:   * Quantity * PricePerUnit * Manufacture |   **Alternative Scenario: N/A**  **Exception: N/A**  **Relationships:**   * Extend by “Search Shop’s Product”   **Business Rules:**   * Product list will be sorted by Product Name ascending order by default | | | |

**Table 10: <Supplier>View Shop’s Product**

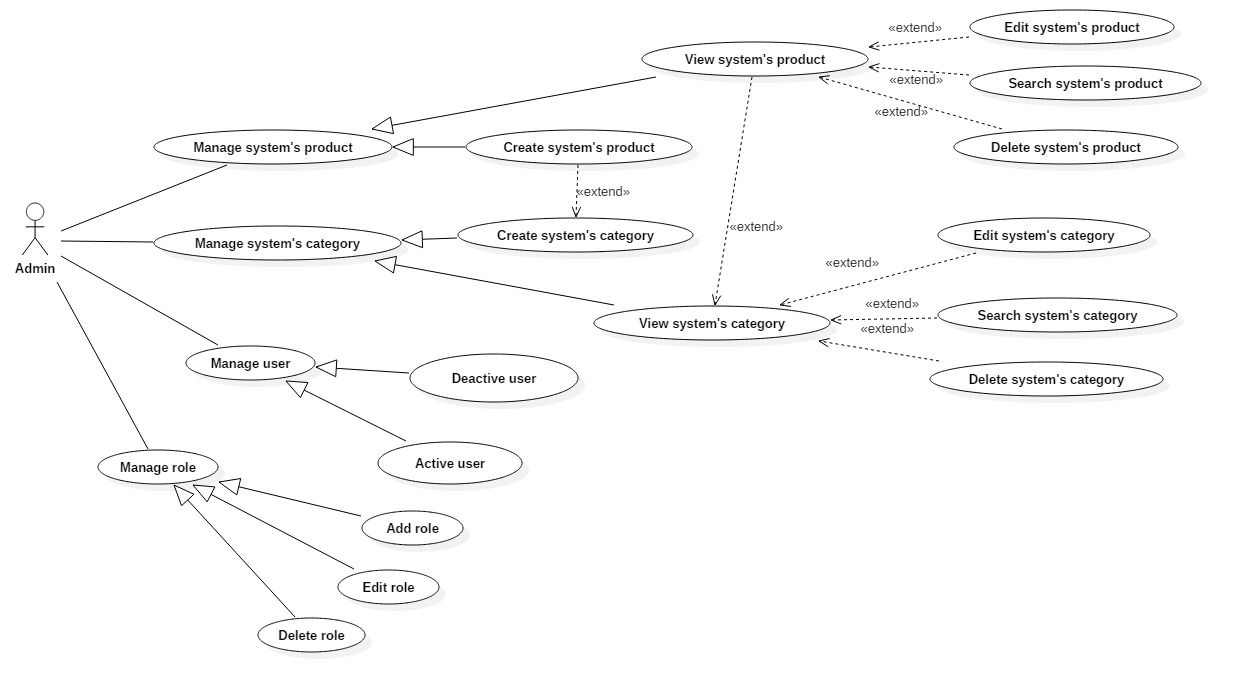
**2.3.5.2 <Supplier>Search Shop’s Product**



**Figure 15: <Supplier>Search Shop’s Product**

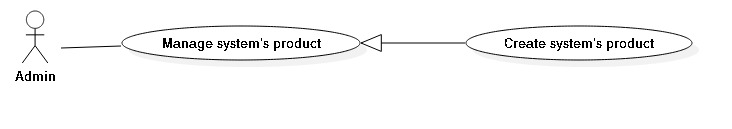
|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE – BMW0** | | | |
| **Use Case No.** | BMW0 | **Use Case Version** | 1.0 |
| **Use Case Name** | Search Shop’s Product | | |
| **Author** | ThienND | | |
| **Date** | February 07, 2017 | **Priority** | Normal |
| **Actor:**   * Supplier   **Summary:**   * This use case allows Supplier to search products.   **Goal:**   * Supplier can search products.   **Triggers:**   * Supplier sends the Search Shop’s Product command.   **Preconditions:**   * User already logged in Supplier role   **Post Conditions:**   * **Success:** System show all products. * **Fail:** Show error message   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | Supplier send command “Search Shop’s Product” | System get and show product list. Each product includes:   * Quantity * PricePerUnit * Manufacture |   **Alternative Scenario: N/A**  **Exception: N/A**  **Relationships:**  **Business Rules:**   * Product list will be sorted by Product Name ascending order by default | | | |

**Table 11: <Supplier>Search Shop’s Product**

**2.3.5 <Admin>Overview Use Case**

**Figure 16: <Admin>Overview Use Case**

**2.3.5.1 <Admin>Create System’s Product**

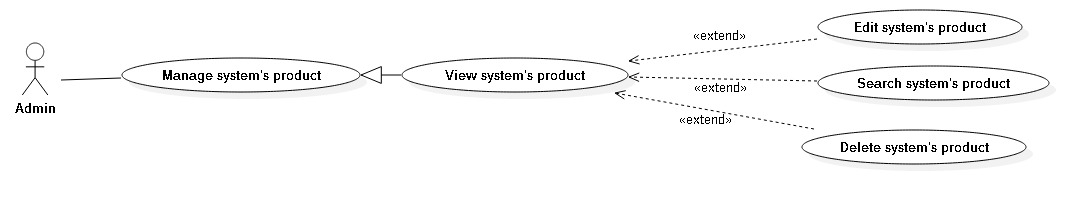


**Figure 17: <Admin>Create System’s Product**

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE – BMW0** | | | |
| **Use Case No.** | BMW0 | **Use Case Version** | 1.0 |
| **Use Case Name** | Create System’s Product | | |
| **Author** | VyCMT | | |
| **Date** | February 07, 2017 | **Priority** | Normal |
| **Actor:**   * Admin   **Summary:**   * This use case allows Admin to create new product.   **Goal:**   * Admin adds product information into system.   **Triggers:**   * Admin sends the Create System’s Product command.   **Preconditions:**   * User already logged in Admin role   **Post Conditions:**   * **Success:** New product has been added to storage. * **Fail:** Show error message   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | Admin goes to Manage Product Screen and click on Create Product | System show Create Product form to enter new product:   * Name: free text input * Manufacturer: drop down list able to choose * Category: drop down list able to choose * Price: free number input * Image: image file * Create button | | 2 | Admin enter the following information |  | | 3 | Admin click on “Create” button | System inserts product into storage  [Exception 1] |   **Alternative Scenario: N/A**  **Exception:**   |  |  |  | | --- | --- | --- | | No | Actor action | System Response | | 1 | Admin do not enter some require fields | System shows message “Field name must be entered” |   **Relationships: N/A**  **Business Rules:**   * Product must belong to one category * Product must belong to one manufacturer * Image type: should be JPG or PNG | | | |

**Table 12: <Admin> Create System’s Product**

**2.3.5.2 <Admin>View System’s Product**

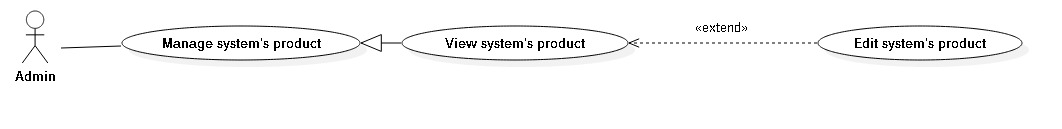


**Figure 18: <Admin>View System’s Product**

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE – BMW0** | | | |
| **Use Case No.** | BMW0 | **Use Case Version** | 1.0 |
| **Use Case Name** | View System’s Product | | |
| **Author** | VyCMT | | |
| **Date** | February 07, 2017 | **Priority** | Normal |
| **Actor:**   * Admin   **Summary:**   * This use case allows Admin to view list of products.   **Goal:**   * Admin can view all products.   **Triggers:**   * Admin sends the View System’s Product command.   **Preconditions:**   * User already logged in Admin role   **Post Conditions:**   * **Success:** System show all products. * **Fail:** Show error message   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | Admin send command “View System’s Product” | System get and show product list. Each product include:   * Product Name * Price * Manufacture * Category * Image |   **Alternative Scenario: N/A**  **Exception: N/A**  **Relationships:**   * Extend by “Edit System’s Product” * Extend by “Delete System’s Product” * Extend by “Search System’s Product”   **Business Rules:**   * Product list will be sorted by Product Name ascending order by default | | | |

**Table 13: <Admin> View System’s Product**

**2.3.5.3 <Admin>Edit System’s Product**

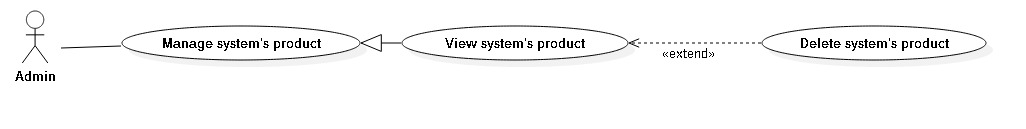
****

**Figure 19: <Admin>Edit System’s Product**

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE – BMW0** | | | |
| **Use Case No.** | BMW0 | **Use Case Version** | 1.0 |
| **Use Case Name** | Edit System’s Product | | |
| **Author** | VyCMT | | |
| **Date** | February 07, 2017 | **Priority** | Normal |
| **Actor:**   * Admin   **Summary:**   * This use case allows Admin to update product.   **Goal:**   * Admin can update product information.   **Triggers:**   * Admin sends the Edit System’s Product command.   **Preconditions:**   * User already logged in Admin role   **Post Conditions:**   * **Success:** Product information has been updated to the storage. * **Fail:** Show error message   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | Admin send command “Edit System’s Product” | System show Edit Product form. It includes:   * Product Name: free text input * Price: free number input * Manufacture: drop down list able to choose * Category: drop down list able to choose * Edit button | | 2 | Admin enter the following information |  | | 3 | Admin click on Edit button | System updates product into storage  [Exception 1] |   **Alternative Scenario: N/A**  **Exception:**   |  |  |  | | --- | --- | --- | | No | Actor action | System Response | | 1 | Admin do not enter some require fields | System shows message “Field name must be entered” |   **Relationships: N/A**  **Business Rules:**   * Product must belong to one category * Product must belong to one manufacturer | | | |

**Table 14: <Admin> Edit System’s Product**

**2.3.5.4 <Admin>Delete System’s Product**



**Figure 20: <Admin>Delete System’s Product**

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE – BMW0** | | | |
| **Use Case No.** | BMW0 | **Use Case Version** | 1.0 |
| **Use Case Name** | Delete System’s Product | | |
| **Author** | VyCMT | | |
| **Date** | February 07, 2017 | **Priority** | Normal |
| **Actor:**   * Admin   **Summary:**   * This use case allows Admin to delete product.   **Goal:**   * Admin can delete products.   **Triggers:**   * Admin sends the Delete System’s Product command.   **Preconditions:**   * User already logged in Admin role   **Post Conditions:**   * **Success:** Product information has been deleted to the storage. * **Fail:** Show error message   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | Admin send command “Delete System’s Product” | System show dialogue: Are you sure want to delete?   * Yes button * No button   [Alternative 1] | | 2 | Admin choose Yes button | System delete products and their relationships |   **Alternative Scenario:**   |  |  |  | | --- | --- | --- | | No | Actor action | System Response | | 1 | Admin choose No button | Dialogue disappear and nothing change |   **Exception: N/A**  **Relationships: N/A**  **Business Rules: N/A** | | | |

**Table 15: <Admin> Delete System’s Product**

**3. Software System Attribute**

**3.1 Usability**

* All the texts, labels should be written in Vietnamese
* Admin, Supplier, Customer, Guest need less than one week of training
* The system should be easy to deploy and easy to use

**3.2 Reliability**

* The number of sending notification failure is 1 time per 1000 notifications.
* Timer tasks run at configured time with 100% execution rate.

**3.3 Availability**

* + - Server have back-up method to make sure that if it has problems while running then all necessary data will be protected and restore as soon as possible.
    - System should take at most 5 hours per month for backup or repairing
    - System is available 24 hours per day and 7 days per week.

**3.4 Security**

* + - Privacy: Each role of user has a specific permission to interact with system.
    - Only admin can grant permission to other roles.
    - System requires SMS verification for all users.
    - System always checks authorization and authenticated before doing anything.

**3.5 Maintainability**

* + - The system is divided into separated modules.
    - The code is easy to maintain and upgrade.

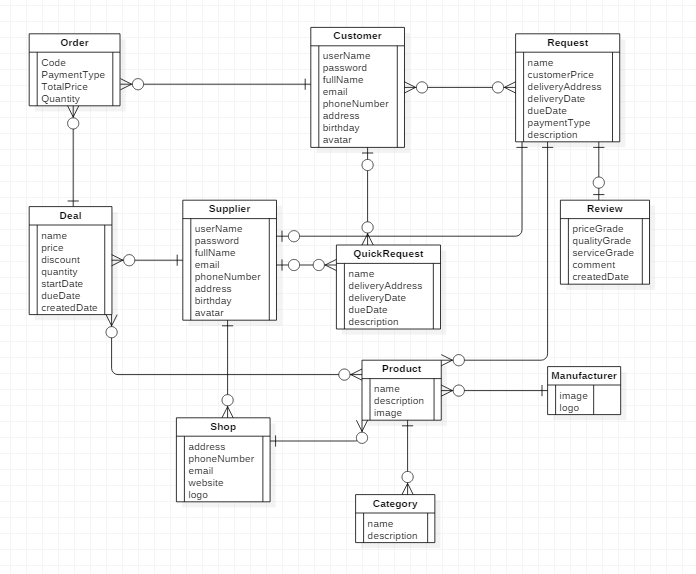
**3.6 Portability**

* The system can be deployed into many type of servers those have IIS8.
* Admin, Supplier, Customer and Guest can use application on Google Chrome version 42 or above.

**3.7 Performance**

* Requests from web application are responded in less than 10 seconds at 8 Mbps bandwidth speed and 1GHz Processing Speed of CPU.

**4. Conceptual Diagram**



**Figure 21: Conceptual Diagram**

**Data Dictionary**

|  |  |
| --- | --- |
| Entity Name | Description |
| Request | Contain the request information |
| Customer | Contain the customer information. |
| Supplier | Contain the supplier information. |
| Order | Contain the order information |
| Deal | Contain the deal information |
| Quick Request | Contain the quick request information. |
| Shop | Contain the shop information. |
| Product | Contain the product information. |
| Category | Contain the category information. |
| Manufacturer | Contain the manufacturer information. |
| Review | Contain the review information. |
| Request | Contain the request information |
| Customer | Contain the customer information. |
| Supplier | Contain the supplier information. |
| Order | Contain the order information |

**Table 16: Conceptual Diagram Data Dictionary**

**D. Report No. 4 Software Design Description**

**1. Design Overview**

<Nội dung này tham khảo và có thể giữ nguyên và chỉ thay thế các phần phù hợp với đồ án của nhóm. Nhóm có thể viết lại cho hay hơn>

- *This document describes the technical and user interface design of* ***MSSC 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 component and functions. It includes class diagrams, class explanations and sequence diagrams for each use cases.*

- *The database design describes the relationships between entities and details of*

*each entity.*

- *Document overview:*

*Section 2: gives an overall description of the system architecture design.*

*Section 3: gives component diagrams that describe the connection and integration of the system.*

*Section 4: gives the detail design description which includes class diagram, class explanation, and sequence diagram to details the application functions.*

*Section 5: describe screens design.*

*Section 6: describe a fully attributed ERD.*

*Section 7: describe algorithms****.***

**2. System Architectural Design**

<Kiến trúc hệ thống mà nhóm xây dựng: sử dụng các pattern và reference đến

nội dung và xem xét lựa chọn các diagram mang đầy đủ nội dung như concept, không sao chép, vay mượn và chế kí hiệu. Nếu dùng kí hiệu ngoài UML thì ghi chú giải kí hiệu ngay cạnh hình vẽ.>

<Mô tả kiến trúc của từng thành phần trong ứng dụng nếu có.>

*Ví dụ*



***Figure 9 System architecture design***

*This diagram is referenced and modified from an original concept from: Chapter 6*

*Architecture Design, SOFTWARE ENGINEERING 9th Edition, by Ian Sommerville.*

*Ví dụ*

**2.1 Web application architecture description**

<Giải thích lý do tại sao lựa chọn mô hình này dựa trên SRS, Introduction, và

project plan đã nêu ra ở các phần trên>

<Mô tả các thành phần của kiến trúc theo dạng bảng, và sự tương tác giữa các thành phần theo kiến trúc.>

*In Web Application, the system is developed under J2EE MVC architecture style. We choose this architecture for Web application because of following advantages:*

*Web app contains a Web service (public API for mobile app), with MVC architecture, we can separate business code with Controller and View, so we can use the business code in web service without repeat the code.*

*...*

*This project follows MVC architecture with following components:*

*Servlet (Controller) is the parts of the application that acts like event handler to handles user interaction. Typically, controller read data from a request and calls appropriate Business’s method then selects view to return to user.*

*...*

**2.2 ...**

**3. Component Diagram**

<Thể hiện việc chia hệ thống thành các component. Nội dung này dựa trên kiến trúc đã đề ra ở phần trên để chia cho phù hợp và đúng mô hình>

**Ghi chú:** Xem lại bộ quy ước kí hiệu của UML 2.0 trước khi vẽ các mối quan hệ cũng như hiểu rõ thiết kế để vẽ chính xác. Nếu tool không phù hợp thì nhóm nên dùng Paint để vẽ

<Mô tả từng thành phần trong hình vẽ theo bảng biểu bên dưới.>

|  |  |
| --- | --- |
| **Component dictionary: describe component** | |
| **Component Name** | **Description** |
|  |  |

*Ví dụ*



**Figure 10 Component Diagram**

|  |  |
| --- | --- |
| *Component Dictionary: Describes components* | |
| *Web Application* | *Web application package: View, Controller* |
| *Mobile Application* | *Mobile application package* |
| *PayPal* | *Handle payment process with PayPal API* |
| *Payment Component* | *Component to handle payment process* |
| *Web Service* | *Provide API for mobile applications to interact with the*  *system.* |
| *Staff Component* | *Component to handle staff activities in the system* |
| *Customer Component* | *Component to handle customer activities in the system* |
| *Public Component* | *Component to handle guest activities in the system* |
| *Admin Component* | *Component to handle admin activities in the system* |
| *Schedule Component* | *Component to handle scheduler in the system* |
| *Business Objects* | *Common objects to handle domain business operations for*  *each components* |
| *Data Access Objects* | *Component to handle interaction between the system and*  *database* |

***Table 10 Component Dictionary***

**4. Detailed Description**

**4.1 Class Diagram**

<Hình thiết kế class diagram: tham khảo các mối quan hệ giữa các lớp trong đặc tả UML, nắm rõ về dependency, association, composition, aggregation, inheritance. Bên cạnh đó, cần xác định rõ cardinality giữa các quan hệ với nhau. Đây là dạng conceptual class diagram, do vậy, cần căn cứ trên conceptual diagram và nội dung xây dựng object cần thiết khi lập trình và xây dựng ứng dụng trong lúc viết chương trình>

<Mô tả từng thành phần class theo bảng biểu bên dưới.>

|  |  |
| --- | --- |
| **Class dictionary: describe Class** | |
| **Class Name** | **Description** |
|  |  |

*Ví dụ*



*Figure 11 Class Diagram*

|  |  |  |
| --- | --- | --- |
| *Class dictionary: describe Class* | | |
| *Class Name* | ***Mapping column with Conceptual diagram*** | ***Description*** |
| *PaymentEntity* | *Payment* | *Contain the payment information.* |
| *CardEntity* | *Card* | *Contain the card information.* |
| *CardInstanceEntity* | *CardInstance* | *Contain the card instance information* |
| *CustomerEntity* | *Customer* | *Contain the customer information.* |
| *ContractEntity* | *Contract* | *Contain the contract information.* |
| *StaffEntity* | *Staff* | *Contain the staff information.* |
| *CompensationEntity* | *Compensation* | *Contain the compensation information.* |
| *PunishmentEntity* | *Punishment* | *Contain the punishment information.* |
| *AccidentEntity* | *Accident* | *Contain the accident information.* |
| *ContractTypeEntity* | *ContractType* | *Contain the contract type information.* |
| *NewCardRequestEntity* | *NewCardRequest* | *Contain the new card request information.* |
| *CardAccessLogEntity* | *N/A* | *Not exist in conceptual diagram. But needed*  *in class diagram to contain the card access log information.* |
| *NotificationEntity* | *N/A* | *Not exist in conceptual diagram. But needed*  *in class diagram to contain the notification information.* |
| *NotificationReadEntity* | *N/A* | *Not exist in conceptual diagram. But needed*  *in class diagram to know what notifications is read.* |

*Ví dụ*

***Table 11 Class dictionary***

**4.2 Class Diagram Explanation**

<Mô tả các thành phần cụ thể cho các lớp đã được vẽ ra ở phần trên>

*Attribute*

***4.2.1 Role***

|  |  |  |  |
| --- | --- | --- | --- |
| ***Attribute*** | ***Type*** | ***Visibility*** | ***Description*** |
| *RoleID* | *int* | *Private* | *Unique identifier of a role* |
| *Name* | *string* | *Private* | *Role name* |

*Method*

|  |  |  |  |
| --- | --- | --- | --- |
| ***Method*** | ***Return type*** | ***Visibility*** | ***Description*** |
| *Getter* | *Attribute type* | *Public* | *Get attribute value* |
| *Setter* | *Void* | *Public* | *Set value of attribute* |

***4.2.2 ...***

**4.3 Interaction Diagram**

**4.3.x Tên Interaction Diagram**

<Sử dụng **sequence diagram là chủ yếu để trình bày nội này**. Sequence diagram cần kết hợp giữa các class đã trình bày ở trên kết hợp với các kiến trúc đã được thuyết minh để có mô hình phù hợp. Đối với ứng **dụng điện thoại di động thì nên sử dụng activity diagram**>

**Summary:** <Nên có phần tóm tắt trước diagram để trình bày về

mục đích của diagram trước khi thể hiện hình vẽ>.

*Ví dụ*

***4.3.1.1 Create new contract***

*Summary: this diagram show process of staff creates new contract*



***Figure 12 Sequence diagram - <Staff> Create new contract***

***4.3.1.2 <Member> View Friend List***

***Summary:*** *This diagram shows how member views all contacts that include MSSC contacts and android cell phone contacts.*



**5. Interface**

***Figure 13: <Member> View Friend List***

**5.1 Component interface**

<Mô tả các interface như của web service hay các signature của core flow được sử

dụng trong hệ thống>

Nội dung được đặc tả theo dạng bảng như sau

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Signature | Description | Input | Output | Output  Format | Exception |
| Tên hàm | Mô tả mục  đích | Tham số  truyền | Kết xuất khi  hàm xử lý xong | Kiểu dữ  liệu | Xử lý lỗi |

*Ví dụ*

***Web Service Interface***

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| *Signature* | *Description* | *Input* | *Output* | *Output*  *Format* | *Exception* |
| *public ResponseObject*  *getCheckConnection(R r)* | *Check*  *server status* | *Request object r* | *Json Boolean*  *the status of server* | *Boolean* | *JsonProcessi*  *ngException* |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| *...* |  |  |  |  |  |

*Ví dụ*

**5.2User Interface Design**

<Chụp và mô tả màn hình>.

**Lưu ý phải đánh số đặc tả các control trên giao diện cùng với các thành phần trong ràng buộc**

***5.3Guest Interface Design***

***5.3.1 Login***



***Fields***

***Figure 14: Login***

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| ***No*** | ***Field***  ***Name*** | ***Description*** | ***Read***  ***only*** | ***Mandatory*** | ***Control***  ***Type*** | ***Data***  ***Type*** | ***Length*** |
| *1* | *Username* | *Fill user*  *name* | *No* | *Yes* | *Textbox* | *String* | *N/A* |
| *2* | *Password* | *Fill*  *password* | *No* | *Yes* | *Password* | *String* | *N/A* |

***Buttons/Hyperlinks***

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ***No*** | ***Function*** | ***Description*** | ***Validation*** | ***Outcome*** |
| *3* | *Signin* | *Log-in into the system* | *N/A* | *Transfer to home page* |

**6. Database Design**

**6.1 Entity relationship diagram (ERD)**

<Thiết kế ERD. Được suy ra và hình thành từ conceptual diagram, class diagram và quá trình hình thành architectural>

**6.2Data Dictionary**

<Mô tả về các thực thể>

|  |  |
| --- | --- |
| **Entity Data dictionary: describe content of all entities** | |
| **Entity Name** | **Description** |
|  |  |

<Mô tả các thành phần bên trong thực thể>

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Entity name** | **Attributes** | **Description** | **Domain** | **Null** |
| Tên | Thuộc tính 1 {PK} | Mô tả | Kiểu dữ liệu | Y/N |
| ... | ... | ... | ... |

**Table 12: Detail Data Dictionary**

\* Business integrity constraint:

<Mô tả các ràng buộc về toàn vẹn dữ liệu để đảm bảo nghiệp vụ>

**7. Algorithms**

<Các thành phần thuật toán - các giải pháp để giải quyết phần core flow mà nhóm

đã áp dụng>

**Chú ý**

Không nhất thiết phải là thuật toán nổi tiếng mà có thể là cách tổ chức dữ liệu cũng như giải thuật do nhóm đang thực hiện ở bên trong hệ thống: ghi rõ bản chất, phân tích về độ phức tạp, nếu tham khảo phải ghi rõ nguồn

Cách giải quyết hay cách áp dụng các qui trình nghiệp vụ hay cách chuyển đổi bài toán khi làm bằng tay - chưa áp dụng máy tính và chương trình để cho thấy việc áp dụng giải bài toán hay giải quyết vấn đề rồi chuyển đổi

cách đó sang thành chương trình máy tính

*Ví dụ*

***7.1Document Breakdown***

***7.1.1 Definition***

*Document breakdown is the way to break the document into many small parts. Each part has it own title and contents of it. And the final data has tree structure.*

***7.1.2 Define Problem***

*All content of document is quite difficute for manage so we must re-construc structure of document for using.*

***7.1.3 Solution***

*To solve this problem, we should follow these steps:*

- *Convert (save) document DOCX file as html type by using Microsoft*

*Word save as Web Filtered.*

- *Import both html file and directory that incluses all pictures of document.*

- *Using xpath to get data of html file as we need, include h1, h2, h3,…,*

*image, text content,..*

- *Save them with structure as below:*

*-TitleA: contentA*

*---TitleA1: contentA1*

*------TitleA1.1: contentA1.1*

*------TitleA1.2: contentA1.2*

*---TitleA2: contentA2*

***7.1.4 Complexity***

- *In total, the complexity of this algorithm is*



***7.1.5 Flowchart***



PAGE \\* MERGEFORMAT 1

**Figure 15: Breakdown document flow chart**

***7.2 String Comparison***

***7.2.1 Define Problem***

*Given two strings. Calculate their matching percent.*

***7.2.2 Requirement***

- *Robustness to changes of word order: two strings which contain the same words, but in a different order, should be recognised as being similar.*

- *Language independence: the algorithm should work not only in English,*

*but in many different languages.*

***7.2.3 Solution***

- *If a string contains many words, break it into a list of words.*

- *For each word, we find out how many adjacent character pairs are contained in it.*

- *Create a function pairs(s) which returns a list of adjacent character*

*pairs of string s.*

- *Then, we use below formula to calculate matching percent.*



***7.2.4 Example***

*Calculate the matching percent of 2 strings: France and French.*

- *Upper case 2 strings:*

+ *France FRANCE.*



+ *French FRENCH.*

- *Break string into list of adjacent character pairs:*

+ *FRANCE*



+ *FRENCH*

- *Calculate its matching percent.*



**E. System Implementation & Test**

**1. Introduction**

**1.1 Overview**

<Mô tả tống quát mục đích test chủ yếu với thời gian và scope và số lượng nhân lực thì nhóm áp dụng phương pháp gì cho việc test>

*Ví dụ*

*This section provides in detail all necessary information about implementation information and testing procedure of MSSC includes test plans, test cases, test result and risks estimations.*

**1.2 Test Approach**

<Phương pháp kiểm thử của nhóm : black box, white box ...>

**2. Database Relationship Diagram**

**2.1 Physical Diagram**

<Vẽ database khi cài đặt vật lý trên các RDBMW: chú ý bố cục cũng nhu kích thước cho dễ đọc>

**2.2 Data Dictionary**

<Mô tả thành phần theo bảng biểu bên dưới>

|  |  |
| --- | --- |
| **Data dictionary: describe content of all tables** | |
| **Table Name** | **Description** |
| Tên | Explanation |

<Mô tả thành phần chi tiết>

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Entity name** | **Attributes** | **Description** | **Domain** | **Null** |
| Tên | Thuộc tính 1 {PK} | Mô tả | Kiểu dữ liệu | Y/N |
| ... | ... | ... | ... |

**Table 13: Attribute Data Dictionary**

**3. Performance Measures**

<Cách nhóm ước lượng việc đo đạc hệ thống>

*Ví dụ*

***3.1 Clustering Performance***

*Clustering is performed by running K Mean Algorithm which has complexity*

*of : O(n \* k \* I \* d)*

o *n : number of points*

o *k : number of cluster*

o *I : number of iteration*

o *d : number of attributes (3)*

*Clustering take almost the time of process that we can ignore the time needed to load data from database, digitalize data.*

*The speed of clustering will vary and increase dramatically when n increase. The purpose of this project is not about optimizing K-Mean Algorithm so it is accepted to let the process run till it completes. Moreover, the clustering is designed to run by staff, wait time is acceptable.*

**4. Test Plan**

<Đưa ra kế hoạch test>

*Ví dụ*

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

**4.1 Features to be tested**

<Tính năng sẽ kiểm thử>

**4.2 Features not to be tested**

<Tính năng sẽ không kiểm thử>

**5. System Testing Test Case**

**<Nên vẽ các workflow tính năng sẽ test để dể hình dung, chú ý dàn trang in**

**ngang, chú ý đánh số, ngày tháng, kết quả, không sao chép>**

*Ví dụ*



***Figure 16: Guest, Member Core Flow***

MSSC - Introduction

***5.1 Guest Test Case***

***5.1.1 Search Event***

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| ***ID*** | ***Test Case***  ***Description*** | ***Test Case Procedure*** | ***Expected output*** | ***Inter-test Case***  ***Dependence*** | ***Result*** | ***Test Date*** | ***Note*** |
|  |  |  |  |  |  |  |  |

MSSC - Introduction

**F. Software User’s Manual**

**1. Installation Guide**

**1.1 Setting up environment at server side**

The following software must be installed into the server machine:

**1.1.1 Hardware requirements**

<Yêu cầu phần cứng server, chú ý xem lại các report trước để nhất quán>

**1.1.2 Software requirements**

<Yêu cầu phần mềm server, chú ý xem lại các report trước để nhất quán>

**1.2 Deployment at server side**

<Mô tả quá trình triển khai lên server thực tế, gợi ý có thể gồm các

bước sau, chú ý khi làm phải chụp hình cụ thể để hướng dẫn cũng như so sánh kết quả thành công>

**1.2.1 Prepare deployment package**

**1.2.2 Configure Server before deploy**

**1.2.3 Deploy web application on server**

**1.3 Setting up the environment at client side**

**1.3.1 Setting up for computer**

<Ghi rõ phiên bản tối thiểu để sử dụng>

**2. User Guide**

<Viết hướng dẫn sử dụng cho người dùng>

**G. Appendix**

<Các thành phần tham khảo của tài liệu chú ý tham khảo thêm cách ghi tại

t

47