**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 |

**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 |

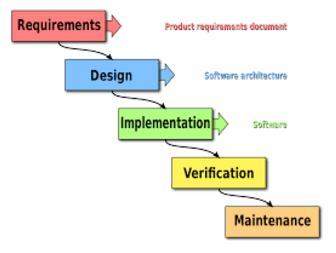
**Table 3: Software Requirement**

**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 processes](https://en.wikipedia.org/wiki/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: 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.
* Documentation (outcomes or products) is produced at each phase and it fits with other engineering process models (means the documents are approved).
* Reflects the type of process model used in other engineering project.

Reference: Based on I2SE subject in FPT University.

**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**

<Liệt kê các yêu cầu về tính năng theo vai trò trong dự án>

*Ví dụ*

***1.1 Guest Requirement***

*Guest is a person who doesn’t have access to the system. Guest can use some*

*functions in the system. To use all functions, guest must login. These are some functions guest can use:*

*Register.*

*Login.*

*...*

***1.2 Member Requirement***

*...*

***1.3 ...***

**2. System Requirement Specification**

**2.1 External Interface Requirement**

**2.1.1 User Interface**

<Liệt kê các yêu cầu về trình bày cho người sử dụng>

*Ví dụ*

*General requirement for graphics user interface is the GUI should be simple, clear, intuitive, and reminiscent.*

*The interface design is an iterate process includes: design, sketching, prototyping, user assessment.*

*Some design principles will be taken into consideration:*

o *UI for businesss web applications - Janko Jovanovic [Ref:* [*http://www.smashingmagazine.com/2010/02/25/designing-user-interfaces-for- business-web-applications/]*](http://www.smashingmagazine.com/2010/02/25/designing-user-interfaces-for-business-web-applications/)

o *Ten principles of effective web design – Vitaly Friedman [Ref:* [*http://www.smashingmagazine.com/2008/01/31/10-principles-of-effective-web- design/]*](http://www.smashingmagazine.com/2008/01/31/10-principles-of-effective-web-design/)

o *Principles of mobile interface design – Jonathan Stark [Ref:*

[*http://www.oreilly.com/pub/e/214*](http://www.oreilly.com/pub/e/2144)*4]*

**2.1.2 Hardware Interface**

<Liệt kê các yêu cầu phần cứng sử dụng trong dự án>

*Ví dụ*

*Smartphone with NFC support.*

**2.1.3 Software Interface**

<Liệt kê các yêu cầu về phần mềm chú ý ghi rõ phiên bản cũng như kích thước

màn hình>

*Ví dụ*

*Web application: work with Firefox (v30 or above), Chromes (v14 or above), Internet*

*Explorer (v10 or above) browse.*

*Mobile application: Android operating system (v 4.0 or above).*

**2.1.4 Communication Protocol**

<Yêu cầu về giao tiếp giữa các thành phần trong ứng dụng>

*Ví dụ*

*Use HTTP protocol 1.1 for communication between the web browser and the web server.*

**2.2 System Overview Use Case**

<Hình Overall Use case của hệ thống: chú ý sử dụng bộ kí hiệu phù hợp ý

nghĩa và phiên bản UML sử dụng để ghi trong mô tả use case>

*Ví dụ*

*Thông tin mô tả về đặc tả UML tham khảo tại* [*http://www.omg.org/spec/UML/2.0/*](http://www.omg.org/spec/UML/2.0/)

*Ví dụ*

**Chú ý**

- Các quan hệ giữa các use case và khi dùng **extend** phải ghi rõ **<extension point> và condition**

- Overview usercase phải thể hiện ràng buộc giữa các usecase trong hệ thống, tuyệt đối **không được liệt kê usecase**

- Nên sử dụng abstract usecase với nhóm chức năng có liên quan. Không nên sử dụng dạng **abstract usecase chỉ có một usecase**, **không sử dụng dạng abstract usecase có chứa thành phần abstract usecase**

- Khi mô tả usecase nên **chú ý tập trung chức năng**, **view** là các **thành phần phụ trợ (có thể nói là extend) không phải** là **chức năng chính** của hệ thống

- Cần phân biệt rõ **usecase là chức năng, qui trình**. Usecase **không phải là**

**màn hình**, hay các **bước - step - trong quá** trình xử lý



PAGE \\* MERGEFORMAT 1

**Figure 2: System Overview Use Case**

*Ví dụ*

**2.3 List of Use Case**

<Đặc tả chi tiêt Use case theo từng role>

<Tách nhỏ thành phần usecase trong overview thành từng nhóm theo vai trò actor trong hệ thống đã được phân tích. Hình vẽ phải bao gồm luôn các usecase có quan hệ>

***2.3.1 <Guest>Overview Use Case***



*Ví dụ*

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

<Tách riêng từng usecase để đặc tả trong usecase specification, lưu ý nều có quan hệ thì phải vẽ hình có luôn quan hệ>

***2.3.1.1 <Guest> Register***

***Use Case Diagram***



***Figure 4: <Guest>Register***

***Use Case Specification***

**GuideLine**: Đây là giai đoạn **lấy requirement** nên các mô tả phải được diễn đạt theo ngôn ngữ của khách hàng, **không phải là nơi mô tả màn hình giao diện khi ứng dụng đã hoàn tất**. Ngoài ra, đây chính là **nơi thể hiện rõ vai trò lấy requirement với phương pháp ethnography - observate** để chuẩn bị thông tin cho thiết kế và thực hiện sản phẩm. Các **nội dung trong phần này** chính là phần **thông tin để hình thành** nên các **thực thể trong conceptual diagram**

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

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE – <UC number>** | | | |
| **Use Case No.** | Đánh số UC | **Use Case Version** | 2.0 |
| **Use Case Name** | Tên UC | | |
| **Author** | Người thiết kế, hiện thực | | |
| **Date** | Ngày viết | **Priority** | Mức độ quan trọng trong dự án. Core  flow thì đánh là  High và giảm dần  đến Normal |
| **Actor:**  - <Actor sẽ thực hiện use case>  **Summary:**  - <Tóm tắt về tính năng của use case>  **Goal:**  - <Mục đích của use case: kết quả khi usecase kết thúc thành công>  **Triggers:**  - <Bước làm use case được kích hoạt>  **Preconditions:**  - <Xác định các ràng buộc phải đạt được trước khi chức năng được thực hiện, thông thường là role của actor, trạng thái yêu cầu của dữ liệu, các ràng buộc về toàn vẹn dữ liệu hay qui trình>  - *Ví dụ: để cancel một hóa đơn thì precondition là*  o ***User phải là một customer***  o ***Hóa đơn vẫn đang trong tình trạng chưa hết thời hạn hủy của hệ thống***  ***là 3 ngày***  **Post Conditions:**  - < Trạng thái sau khi tiến hành bắt buộc phải có 2 trạng thái cho success và fail.  Vì vậy khi ghi phải có đủ và phần fail bắt buộc xuất hiện trong exception scenario>  - **Success: Khi thành công thì tình trạng hệ thống thế nào đối với hệ thống**  **và đối với người dùng**  - **Fail: Khi có lỗi xảy ra thì hệ thống sẽ xử lý thế nào để đảm bảo usability**  **cho người dùng và toàn vẹn dữ liệu cho hệ thống**  **Main Success Scenario: <Hướng xử lý chính của hệ thống>**  Step Actor Action System Response  1 -  2  **Alternative Scenario: <Hướng xử lý khác trong tình huống dữ liệu cụ thể như** | | | |

**mệnh đề if hoặc lựa chọn khác của người dùng trong quá trình main flow được diễn ra>**

|  |  |  |
| --- | --- | --- |
| No | Actor Action | System Response |
| 1 |  |  |

**Exceptions: Gồm các tình huống xử lý ngoại lệ cũng như xử lý các exception do**

**người dùng gây ra khi nhập liệu**

|  |  |  |
| --- | --- | --- |
| No | Actor Action | System Response |
|  |  |  |

**Relationships:** Mối quan hệ với các Use case khác nếu có trong quá trình xử lý, tuy nhiên nó không phải là abstract usecase

**Business Rules:**

- Thành phần mô tả các yêu cầu về mặt nghiệp vụ của use case.

- Tất cả các giả định về nghiệp vụ nếu có phải được ghi vào

- Chú ý tới sự chuyển đổi về trạng thái của dữ liệu cũng phải được ghi tại đây

- Các định nghĩa cũng cần làm rõ (sản phẩm nổi bật, sản phẩm sắp có là sản phẩm thế nào trong hệ thống)

- Các ràng buộc dữ liệu dưới hệ thống, các rule liên quan đến toàn vẹn dữ liệu

- Các qui trình, activities, quá trình chuyển đổi trạng thái của hệ thống

*Ví dụ*



|  |  |  |  |
| --- | --- | --- | --- |
| *USE CASE – WG01* | | | |
| *Use Case No.* | *WG01* | ***Use Case Version*** | *2.0* |
| *Use Case Name* | *Login* | | |
| *Author* | *TrungDQ* | | |
| *Date* | *27/05/2015* | ***Priority*** | *Normal* |
| *Actor:*  - *Guest*  *Summary:*  - *This use case allows guest to log in the system. Goal:*  - *Guest can log in the system. Triggers:*  - *Guest sends the login command. Preconditions:*  - *N/A*  *Post Conditions:* | | | |

- *Success: Guest login the system.*

- *Fail: Show error message.*

*Main Success Scenario:*

|  |  |  |
| --- | --- | --- |
| *Step* | *Actor Action* | *System Response* |
| *1* | *Guest goes to login view.* | *System requires identity information from*  *Guest:*  - *Email or customer code: free text input*  - *Password: free text input* |
| *2* | *Guest inputs information.* |  |
| *3* | *Guest sends command to login*  *to system* | *Guest will login system with their specific*  *role*  *[Alternative 1] [Exception 1]* |

*Alternative Scenario:*

|  |  |  |
| --- | --- | --- |
| *Step* | *Actor Action* | *System Response* |
| *1* | *Guest enter wrong identity*  *information.* | *Wrong identity information, System shows*  *error message.* |

*Exceptions:*

|  |  |  |
| --- | --- | --- |
| *Step* | *Actor Action* | *System Response* |
| *1* |  | *System show message the "System is busy"*  *when the internet is lost* |

*Relationships: N/A*

*Business Rules:*

- *Password are encrypted before being sent to server.*

- *After login to system, guest will be redirected to specific view based on their role on the*

*system: staff or customer.*

o *If role is “Customer”, the system will display to Customer view.*

o *If role is “Staff”, the system will display to Staff Dashboard view.*

*Ví dụ*

***<Guest> Create new contract request***



***Figure 5 <Guest> Create new contract request***

|  |  |  |  |
| --- | --- | --- | --- |
| *USE CASE – WG02* | | | |
| *Use Case No.* | *WG02* | ***Use Case Version*** | *2.0* |
| *Use Case Name* | *Create new contract request* | | |
| *Author* | *TrungDQ* | | |
| *Date* | *27/05/2015* | ***Priority*** | *Normal* |
| *Actor:*  - *Guest* | | | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| *Summary:*  - *This use case allows guest to create new contract request. Goal:*  - *Guest can create new contract request. Triggers:*  - *Guest sends command to create contract request. Preconditions:*  - *N/A*  *Post Conditions:*  - *Success: New account and new contract will be created for guest.*  - *Fail: Show error message.*  *Main Success Scenario:* | | | | |
|  | *Step* | *Actor Action* | *System Response* |  |
| *1* | *Guest goes to new*  *contract view.* | *System requires information from guest:*  ***Personal information***  - *Name: free text input, required, length 3 – 80.*  - *Address: free text input, required, length 3 – 250.*  - *Email: free text input, required, length 3 – 250.*  - *Phone number: free text input, required, length 8 –*  *15.*  - *Personal ID: free text input, length 8 – 15.*  ***Contract information*** *(all information below are required)*  - *Contract’s type: select one of the options.*  - *Start date: date time input, required.*  - *Contract term: text*  - *Contract’s fee: text*  ***Vehicle information***  - *Plate: free text input, required, length 4 – 15.*  - *Brand: free text input, required, length 2 – 20.*  - *Model code: free text input, length 2 – 20.*  - *Vehicle type: free text input, length 2 – 20.*  - *Color: free text input, length 2 – 20.*  - *Engine: free text input, required, length 2 – 20.*  - *Chassis: free text input, required, length 2 – 20.*  - *Capacity: free text input, required, length 2 – 20.*  - *Year of manufacture: number text input, value from 1900 to current year.*  - *Weight: free text input, value from 1 – 1000, unit:*  *kilogram*  - *Seat capacity: free text input, value from 1 – 100.*  ***Security question***  - *Answer: free text input, required, length 1 -*  *10* |
| *2* | *Guest inputs*  *information.* |  |
| *3* | *Guest sends command*  *to create new contract request.* | *System validate information, display contract details*  *and request for confirmation. [Exception 1, 2, 3]* |
| *4* | *Guest sends command*  *to create new contract request.* | *Add new account and new contract information to the*  *system. Show successful message and ask user to process payment.* |
| *5* | *Guest sends command* | *Display new view let user select one of following* |

PAGE \\* MERGEFORMAT 1

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  | *to process payment* | *payment gateways:*  *- PayPal payment gateway.*  *- Direct payment.*  *And show guest the fee:*  *Contract’s fee: text.* |  |
| *6* | *If user chooses PayPal*  *gateway and sends confirm command. [Alternative 1]* | *Forward to PayPal payment view to process the*  *payment.* |
| *7* | *User process the*  *PayPal payment* | *If payment succeed:*  *Show message created successful. [Exception 4]* |
| *Alternative Scenario:*  *No Actor Action System Response*  *1 If user chooses direct payment Show company address map. method*  *Exceptions:*  *No Actor Action System Response*  *1 Guest sends command to create System shows error message to ask user new contract request input missing required fields.*  *2 Guest’s email is existed in the Show message to notify guest that their email system is existed in the system.*  *3 Guest’s vehicle plate is existed Show message to notify guest that their in the system vehicle is existed in the system.*  *4 If payment failed Show message to notify user that payment failed and the renew request has been aborted.*  *Relationships: Payment*  *Business Rules:*  - *New customer account and new contract will be created in the system with inputted information.*  - *The initial status of contract will be set to “Pending”.*  - *When customer completed payment process:*  *+ if the contract’s start date has come, contract’s status would change from “Pending”*  *to “No Card”.*  *+ If start date is not come yet, the contract status is not changed.*  - *Staff will receive a notification about new contract request, they verify contract’s information and issue a card for this contract, in this case, contract’s status would*  *change from “No Card” to “Ready”.*  - *System must ensure has no duplicate customer or vehicle.*  - *An email contains customer code and password will be sent to user, user can use this information to login to the system later.*  - *Start date must not be earlier than the current date.*  - *Contract term is specified by the system.*  - *Contract types are loaded from system, contract type can be managed by system administrator.*  - *Contract price would be calculated from contract type and contract term.* | | | | |

***Table 6 Use case WG02 - <Guest> Create new contract request***

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

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

*Ví dụ*

***<Customer> Cancel contract***



***Figure 6 <Customer> Cancel contract***

|  |  |  |  |
| --- | --- | --- | --- |
| *USE CASE – WC03* | | | |
| *Use Case No.* | *WC03* | ***Use Case Version*** | *2.0* |
| *Use Case Name* | *Cancel contract* | | |
| *Author* | *TriPQM* | | |
| *Date* | *27/05/2015* | ***Priority*** | *High* |
| *Actor:*  - *Customer. Summary:*  - *This use case helps user cancel their contract. Goal:*  - *Customer can cancel the contract. Triggers:*  - *Customer sends cancel contract request. Preconditions:*  - *User must login into the system with role Customer.*  - *User’s contract has not expired.*  - *Customer's contract status must not be “Expired”, "Cancelled" or “Request cancel”.*  *Post Conditions:*  - *Success: Send to the staff the cancel contract request.*  - *Fail: Show error message. Main Success Scenario:*  *Step Actor Action System Response*  *1 User goes to cancel contract Display new view require user input some view. information:*  - *Reason to cancel the contract: can be optional selected from these values:*  o *“Xe cơ giới bị thu hồi đăng ký và biển số theo quy định của pháp luật”*  o *“Xe cơ giới hết niên hạn sử dụng theo*  *quy định của pháp luật”*  o *“Xe cơ giới bị mất được cơ quan công an*  *xác nhận”*  o *“Xe cơ giới hỏng không sử dụng được hoặc bị phá huỷ do tai nạn giao thông được cơ quan công an xác nhận”*  o *Other reason: free text input, required, length 1-250.*  *2 User inputs information*  *3 User sends cancel contract* - *Change contract status. request command.* - *Send request to the Staff.*  *[Exception 1]*  *Alternative Scenario: N/A* | | | |

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

*Exceptions:*

|  |  |  |
| --- | --- | --- |
| *No* | *Actor Action* | *System Response* |
| *1* | *If user didn't check any reason*  *to cancel contract* | *Show message to notify user that they have to*  *choose the reason for cancel contract.* |

*Relationships: N/A*

*Business Rules:*

- *Cancel contract request will be sent to the system with inputted information.*

- *System update status of the contract from “Pending”, “No Card” or “Ready” to*

*“Request cancel”.*

- *A notification will be sent to staff after the process is completed.*

***Table 7 Use case WC03 - <Customer> Cancel contract***

*Ví dụ*

System

**Auto parse**

**System**

***Figure 7: <System> Auto parse use case diagram***

***Use Case Specification***

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| ***USE CASE – ARB08*** | | | | | | | |
| ***Use Case No.*** | | | *ARB08* | | ***Use Case Version*** | *2.0* | |
| ***Use Case Name*** | | | *Auto parse* | | | | |
| ***Author*** | | | *Pham Nguyen Bich Hien* | | | | |
| ***Date*** | | | *30/05/2014* | | ***Priority*** | *Normal* | |
| ***Actor:***  - *System.*  ***Summary:***  - *System can parse resource automatically from many websites at specified time.*  ***Goal:***  - *Get resource from many websites.*  ***Triggers:***  - *The time hits configured time.*  ***Preconditions:***  - *Parse time has been configured.*  ***Post Conditions:***  - ***Success:*** *New data is inserted to storage. Log file is generated.*  - ***Fail:*** *Nothing is changed in the storage. Log file is generated.*  ***Main Success Scenario:*** | | | | | | | |
|  | *Step* | *Actor Action* | | *System Response* | | |  |
| *1* | *Server checks the current time.*  *If it hits configured time, parse process starts.* | | *Send request to the parsed link.*  *Fetch data from the response based on the inputted XPaths.* | | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  |  | *Validate data [Exception 1].*  *If data is valid, insert to storage*  *[Alternative 1].*  *Generate log file.* |  |
| ***Alternative Scenario:***  *Step Actor Action System Response*  *1 Server checks the current time. If it hits configured time, parse process starts.*  *If fetched link resource is already in the storage, update its information.*  *Generate log file.*  ***Exceptions:***  *No Actor Action System Response*  *1 Data is invalid.*  *Generate log file.*  ***Relationships:*** *N/A*  ***Business Rules:***  - *If link resource exists in storage, do nothing.*  - *If link resource is not active, do nothing.*  - *Log file structure: ARB LOG FILE*  *Tạo file lúc: {Created date}, {Created time}*  *STT Link Thời gian Dạng dữ Tổng số sách Insert thành Insert thất parse liệu nhận được công bại*  *Tổng thời gian parse dạng {Data type}: {Elapsed time} Tổng thời gian parse: {Total elapsed time}*  *Tổng sản phẩm parse được: {Total parsed books}*  - | | | | |

***Table 8: Auto parse use case specification table***

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

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

**3. Software System Attribute**

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

<Mô tả non-functional requirement, các nội dung phải có dẫn chứng về việc

đã đo đạc, có định lượng bằng các phương pháp, công cụ và phải hiểu về các nội dung đã ghi ra.>

**3.1 Usability**

**3.2 Reliability**

**3.3 Availability**

**3.4 Security**

**3.5 Maintainability**

**3.6 Portability**

**3.7 Performance**

…..

**4. Conceptual Diagram**

<Xác định các **thực thể - không cần có thuộc tính** - và **mối quan hệ** giữa chúng với nhau **thông qua các business rule**, **actor**, các **thành phần có mối quan hệ** để hình thành nên các thực thể thông qua các **mô tả trong usecase diagram và usecase specification** đã nêu ra ở trên>

**Chú ý**

Chỉ sử dụng một tập kí hiệu và cần reference đến địa chỉ mô tả tập kí hiệu

để sử dụng cho chính xác

Các Diagram cần lớn rõ ràng, phải dàn trang cho phù hợp và nên dùng

trang A3 để in

Các thành phần trong diagram phải được thể hiện thông qua dictionary

**Data Dictionary <Đặc tả các thực thể có trong hình>**

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

*Ví dụ*



*Figure 8 Conceptual diagram*

***Data Dictionary***

|  |  |
| --- | --- |
| *Entity Data dictionary: describe all content of all entities* | |
| *Entity Name* | ***Description*** |
| *User* | *Abstract entity describes a user in system* |
| *Customer* | *Contain the customer information.* |
| *Contract* | *Contain the contract information.* |
| *Card* | *Contain the card information* |
| *CardInstance* | *Represent a card assigned to a contract* |
| *Payment* | *Contain the payment information.* |
| *Staff* | *Contain the staff information.* |
| *Compensation* | *Contain the compensation information.* |
| *Punishment* | *Contain the punishment information.* |
| *Accident* | *Contain the accident information.* |
| *ContractType* | *Contain the contract type information.* |
| *NewCardRequest* | *Contain the new card request information.* |
| *Notification* | *Contain the notification information* |

***Table 9 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

[http://www.khoahocviet.info/meresci/vi/meresci03d4.html>](http://www.khoahocviet.info/meresci/vi/meresci03d4.html)

47