# Software Project Management Plan

## Problem Definition

### Name of this Capstone Project

* **Official name**: Build an Information Management System for a datacenter
* **Vietnamese name**: Hệ thống quản lý thông tin của một trung tâm dữ liệu
* **Abbreviation**: IMS

### Problem Abstract

At current, the information management of the data center is facing many troubles. All of processes are still handled manually with Excel, Word, etc. by staff. Therefore, the staff of data center has to spend a lot of time and effort to manage customer information, server information, server location, IP address. In addition, this is easy to cause a loss of data, affect to customer satisfaction and reduce work performance.

To overcome above problems, we provide a convenient system for data center. Firstly, this system helps staff to manage easier, work faster and enhance the accuracy. Secondly, it helps the admin of data center to manage staff, control all of activities of data center. Finally, it helps customers to have comfortable channel to control information of their servers and communicate with data center.

### Project Overview

#### 1.3.1. Current Situation

Below are the problems encountered in this project:

* **Depend on an existed data center**: In developing time, we must contact to this data center to learn about ways which they are working and rules in data center. From that, understand their problems and find the best solution for them.
* **Knowledge about network configuration:** To deal with function named “Add network configuration of server”, the members of teams have to spend time to research about network configuration.
* **Testing:** hardly to test cover all of the cases which can happen, must list as many as possible test cases.
* **Absent of team members:** team members get sick or unexpected problems. Working time and learning time are different among members.
* **Ability of team members:** working abilities are different among members. Therefore, the leader must know about the ability of each member and assign task appropriately.

#### 1.3.2. The Proposed System

From current issues, combined with research results in data center at QTSC, we create a system which is user-friendly and strictly designed to avoid as many errors as possible. It must support the most needs in information management and provide reports and logs of all basic activities or changes which take place at data center.

##### Customer information

* Staff can add/update customer information.
* Staff can active/ deactivate customer account.
* Customer can send request to go into data center and staff can record time customers go into data center.
  + - * 1. **Server information**
* Staff can add/edit/search/delete configuration of a server.
* Staff can comply with the necessary formalities when a server was carried on/out server room by customer.
* Customers can search and view the information and history of their servers.
* The system can create report of server delivery.
* Staff can record content when server was upgraded configuration.

##### Server placement

* Staff can add/update/search/delete location of server on a rack.
* Staff can add network configuration of server.
* The system can record time when server is moved from a rack to a rack.
* The system can create report of server moving.

##### IP address location

* Staff can add new IP addresses.
* Staff can update usage status of IP address.
* Staff can comply with the necessary formalities when IP address assigned to server.
* Staff can search available/ unavailable IP addresses.

##### Usage history of IP address

* The system can report blocked IP address.
* The system can report free IP address.
* The system can create statistic IP addresses being used by customers.

#### 1.3.3. Boundaries of the System

* The system could be used by staff and customer with a laptop, PC or MAC.
* The used language of the system is English.
* The complete product includes:

+ The website

+ All the process document involved.

#### 1.3.4. Future plans

In the future, the function named “Record time customer go into data center” can be enhanced by the customer can use the electronic card or fingerprint to check the time they go into data center. By this way, the system will become more convenient and saving time is more accurate.

#### Development Environment

##### Hardware requirement

**For client**

|  |  |  |
| --- | --- | --- |
| Windows | Minimum Requirements | Recommended |
| Operating System | Windows 7 | Windows 7, 8, 10 |
| Computer Processor | Intel® Core 2(TM) i3 CPU M370 @2.4GHz 3.39GHz | Intel® Core(TM) i5-2410M CPU @ 2.30GHz |
| Computer Memory | 2GB RAM | 4GB or more |
| Internet Connection | Cable, Wi-Fi (2 Mbps) | Cable, Wi-Fi (12 Mbps) |

Table 1: Hardware Requirement for client

**For server**

|  |  |  |
| --- | --- | --- |
| Windows | Minimum Requirements | Recommended |
| Operating System | Window Server 2008 | Window Server 2008 |
| Computer Processor | CPU Intel Xeon E3 2.0GHz | 2.0 GHz CPU 6 core E5-2620 |
| Computer Memory | 2 GB RAM DDR3 | 4 GB RAM DDR2 |
| HDD | 100 GB HDD Raid 0,1 | 500 GB HDD Raid 0,1 |
| Bandwidth | 300Mbps | 500Mbps |
| International bandwidth | 2Mbps | 10Mbps |

Table 2: Hardware Requirement for Server

##### Software requirement

|  |  |
| --- | --- |
| Software | Name / Version |
| Operating system | Windows 7 or above |
| Modeling tool | StartUML, Gliffy |
| DBMS | SQL Server 2012 |
| Source control | GitHub, Source Tree 1.7.0.32509 |
| Web browser | Google Chrome |
| Team Collaboration | Slack |
| Issues and Task Management | GitHub and Trello |
| Implement web-application | Visual Studio 2015 |

Table 3: Software requirement

## Project organization

## Software Process Model

Project is developed under Sashimi Waterfall Model.

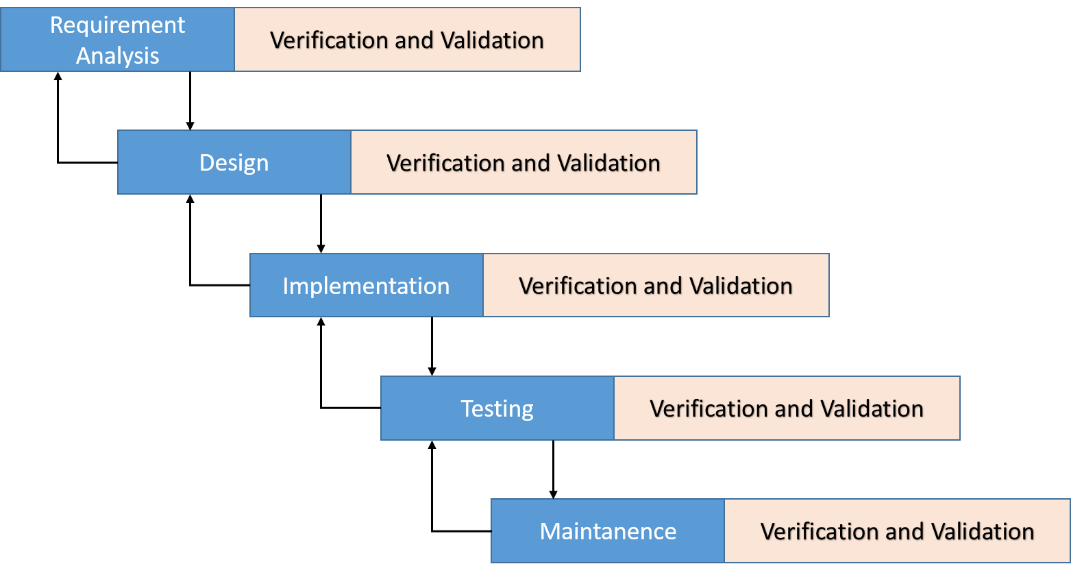


Figure 1: Sashimi Waterfall Model

For more information: <http://www.waterfall-model.com/sashimi-waterfall-model/>

The purpose of applying Sashimi Waterfall model for IMS-datacenter is because of:

* Requirements of the project may be made clearly and difficult to change.
* In current phrase, if the error which in previous phrase or the update occur, we can return and fix it.

## Roles and responsibilities

|  |  |  |  |
| --- | --- | --- | --- |
| **No** | **Full name** | **Role in Group** | **Responsibilities** |
| **1** | Ngo Dang Ha An | Project manager | * Specify user requirement * Control the development process * Give out technique and business analysis support |
| **2** | Le Thi Thu Ha | Team Leader, BA, DEV, Tester | * Managing process * Designing database * Clarifying requirements * Prepare documents * GUI Design * Create test plan * Coding * Testing |
| **3** | Huynh Lam Ha Tien | Team Member, BA, DEV, Tester | * Designing database * Clarifying requirements * Prepare documents * GUI Design * Create test plan * Coding * Testing |
| **4** | Cao Hong Nam | Team Member, BA, DEV, Tester | * Designing database * Clarifying requirements * Prepare documents * GUI Design * Create test plan * Coding * Testing |

Table 4: Roles and Responsibilities Details

## Tools and Techniques

* Front-end technologies: HTML5, CSS3, JavaScript, jQuery
* Application is built on .NET platform.
* Web Server: Microsoft IIS version 7.
* Database Management System: MS SQL Server 2012

## Project Management Plan

#### Software development life cycle

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Phase** | **Description** | **Deliverables** | **Resource needed** | **Dependencies and Constrains** | **Risks** |
| **Requirement Analysis** | - Collect requirements base on user’s behaviours  - Identify and clarify requirements for the system in general | - Introduction of project  - Software requirement specification  - Project task plan  - Prototypes | 40 man-days | N/A | - Missing requirement  - Unclear scope  - Not have a clear understanding about business process |
| **Design** | - Design system Architecture  - Design component Diagram, conceptual diagram  - Detail design | - Software design description  - Technology notes | 60 man-days | Depend on  “Requirement  Analysis” | - Lack of experience  - Not full fill requirements |
| **Implementation** | - Implement physical database  - Coding  system core  functions and  other feature  with GUI  - Unit test | - Physical database diagram  - Main user’s  functions on  website | 80 man-days | Depend on  “Design” | - Lack of experience  - Human mistake |
| **Testing** | - Write test case  - Do Integration  test the system test  - Do Alpha test  - Correct bugs  - Beta test  - Acceptance  test | - Test  document | 40 man-days | Depend on  “Implementation” | - Lack of experience  - Missing  test case |
| **Maintenance** | - Deploy on  sever | - Installation guide  - User Manual | 20 man-days | Depend on  “Testing” | - Lack of  experience |

Table 5: Software Development Life Cycle Detail

#### Phase Detail

#### Phase 1: Requirement Analysis

|  |  |  |
| --- | --- | --- |
| **Task** | **Description** | **Author** |
| 1. Collect requirements | - Analysis real user’s behaviours  - Find which systems currently  provide similar service, their  strengths and weaknesses | HaLTT, TienHLH, NamCH |
| 2. Identify and clarify  main functions | - Define which main functions system  should provide | HaLTT, TienHLH, NamCH |
| 3. Create System  Introduction | - Create Introduction Report. | HaLTT, TienHLH, NamCH |
| 4. Software Project  Management Plan | - Create Project Management Plan | HaLTT, TienHLH, NamCH |
| 5. Prototype | - Build a prototype of propose  website | HaLTT |
| 6. SRS | - Create SRS document | HaLTT, TienHLH, NamCH |

Table 6: Phase 1 - Requirement Analysis

#### Phase 2: Design

|  |  |  |
| --- | --- | --- |
| **Task** | **Description** | **Author** |
| 1.Design system Architecture | - Design system architecture  - Design component diagram | HaLTT, TienHLH, NamCH |
| 2. Database Design | - Based on parsed data to  recommendation  - Based on other needs to  recommendation | HaLTT, TienHLH, NamCH |
| 3.Detail design | - Design for each function | HaLTT, TienHLH, NamCH |
| 4. Technology  research | - Note some technology will be applied in project | TienHLH |
| 5. Design Document | - Create software design document | HaLTT, TienHLH, NamCH |

Table 7: Phase 2 - Design

#### Phase 3: Implementation

|  |  |  |
| --- | --- | --- |
| **Task** | **Description** | **Author** |
| 1. Physical database | - Implement physical database | HaLTT |
| 2. Front-end web  functions | - Implement front-end functions  on web | HaLTT, TienHLH, NamCH |
| 3. Back-end web  functions | - Implement back-end functions on  web | HaLTT, TienHLH, NamCH |
| 4. Unit testing | - Write test case and testing for  web functions | HaLTT, TienHLH, NamCH |

Table 8: Phase 3 - Implementation

#### Phase 4: Testing

|  |  |  |
| --- | --- | --- |
| **Task** | **Description** | **Author** |
| 1. Integration testing | - Write test case and testing  Modules | HaLTT, TienHLH, NamCH |
| 2. System testing | - Write test case and testing  System | HaLTT, TienHLH, NamCH |
| 2. Alpha testing | - Do alpha test with customer | HaLTT, TienHLH, NamCH |

Table 9: Phase 4 - Testing

#### Phase 5: Maintenance

|  |  |  |
| --- | --- | --- |
| **Task** | **Description** | **Author** |
| 1. Installation guide | - Write installation guide | HaLTT, TienHLH, NamCH |
| 2. User manual | - Write user manual | HaLTT, TienHLH, NamCH |

Table 10: Phase 5 - Maintenance

#### All Meeting Minutes

Refer to Meeting Minutes folder in CD.

## Coding Convention

Using C# language to develop website:

- Naming Convention:

* Using Camel Case for method arguments and local variables.
* Using Pascal Case for class names and method names.

- Layout Convention:

* Write only one statement per line.
* Write only one declaration per line.
* If continuation lines are not indented automatically, indent them one tab stop (four spaces).
* Add at least one blank line between method definitions and property definitions.
* Use parentheses to make clauses in an expression apparent, as shown in the following code

- 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 the comment delimiter (//) and the comment text, as shown in the following example.

Ex: // Here is your comment.

* Do not create formatted blocks of asterisks around comments.

Using C# Coding Convention from:

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