**MINISTRY OF EDUCATION AND TRAINING**

**FPT UNIVERSITY**

Capstone Project Document

**Office Task Management**

|  |  |
| --- | --- |
| **Group 3** | |
| **Group member** | Nguyen Cu Dat – SE61010  Lu Cam Toan – SE60633  Le Hai Trieu - SE61146  Tran Thi Kim Anh – SE60909 |
| **Supervisor** | Mr. Lai Duc Hung |
| **Ext. Supervisor** | N/A |
| **Capstone Project code** | OTM |

-Ho Chi Minh City, 13/05/2015-

# Table of Contents

[Table of Contents 3](#_Toc420758204)

[List of Tables 6](#_Toc420758205)

[List of Figures 7](#_Toc420758206)

[Definitions, Acronyms, and Abbreviations 8](#_Toc420758207)

[A. Introduction 9](#_Toc420758208)

[1. Project Information 9](#_Toc420758209)

[2. Introduction 9](#_Toc420758210)

[3. Current Situation 9](#_Toc420758211)

[4. Problem Definition 9](#_Toc420758212)

[5. Proposed Solution 9](#_Toc420758213)

[5.1 Feature functions 9](#_Toc420758214)

[5.2 Advantage and disadvantage 10](#_Toc420758215)

[6. Functional Requirement 10](#_Toc420758216)

[6.1 User Management 10](#_Toc420758217)

[6.2 Tracking task 10](#_Toc420758218)

[6.3 Review and approve document 10](#_Toc420758219)

[6.4 Collaborate in real-time 10](#_Toc420758220)

[6.5 Statistic 10](#_Toc420758221)

[7. Role and Responsibility 10](#_Toc420758222)

[B. Software Project Management Plan 12](#_Toc420758223)

[1. Problem Definition 12](#_Toc420758224)

[1.1 Name of this Capstone project 12](#_Toc420758225)

[1.2 Problem Abstract 12](#_Toc420758226)

[1.3 Project Overview 12](#_Toc420758227)

[2. Project organization 13](#_Toc420758228)

[2.1 Software Process Model 13](#_Toc420758229)

[2.2 Roles and responsibilities 13](#_Toc420758230)

[2.3 Tools and Techniques 14](#_Toc420758231)

[3. Project Management Plan 14](#_Toc420758232)

[3.1 Software development life cycle 14](#_Toc420758233)

[3.2 Phase Detail 14](#_Toc420758234)

[3.3 All Meeting Minutes 15](#_Toc420758235)

[4. Coding Convention 15](#_Toc420758236)

[C. Software Requirement Specification 15](#_Toc420758237)

[1. User Requirement Specification 15](#_Toc420758238)

[1.1 Guest Requirement 16](#_Toc420758239)

[1.2 Member Requirement 16](#_Toc420758240)

[1.3 Partner Requirement 16](#_Toc420758241)

[1.4 Staff Requirement 16](#_Toc420758242)

[1.5 Admin Requirement 16](#_Toc420758243)

[2. System Requirement Specification 16](#_Toc420758244)

[2.1 External Interface Requirement 16](#_Toc420758245)

[2.2 System Overview Use Case 17](#_Toc420758246)

[2.3 List of Use Case 19](#_Toc420758247)

[3. Software System Attribute 32](#_Toc420758248)

[3.1 Usability 32](#_Toc420758249)

[3.2 Reliability 32](#_Toc420758250)

[3.3 Availability 32](#_Toc420758251)

[3.4 Security 33](#_Toc420758252)

[3.5 Maintainability 33](#_Toc420758253)

[3.6 Portability 33](#_Toc420758254)

[3.7 Performance 33](#_Toc420758255)

[4. Conceptual Diagram 33](#_Toc420758256)

[D. Software Design Description 33](#_Toc420758257)

[1. Design Overview 33](#_Toc420758258)

[2. System Architectural Design 34](#_Toc420758259)

[2.1 Web application architecture description 34](#_Toc420758260)

[2.2 Mobile application architecture description 34](#_Toc420758261)

[3. Component Diagram 34](#_Toc420758262)

[4. Detailed Description 35](#_Toc420758263)

[4.1 Class Diagram 35](#_Toc420758264)

[4.2 Class Diagram Explanation 35](#_Toc420758265)

[4.3 Interaction Diagram 35](#_Toc420758266)

[5. User Interface Design 37](#_Toc420758267)

[5.1 Guest Interface Design 37](#_Toc420758268)

[6. Database Design 38](#_Toc420758269)

[6.1 Logical Diagram 38](#_Toc420758270)

[6.2 Data Dictionary 38](#_Toc420758271)

[7. Algorithms 39](#_Toc420758272)

[7.1 Document Breakdown 39](#_Toc420758273)

[7.2 String Comparison 40](#_Toc420758274)

[E. System Implementation & Test 41](#_Toc420758275)

[1. Introduction 41](#_Toc420758276)

[1.1 Overview 41](#_Toc420758277)

[1.2 Test Approach 43](#_Toc420758278)

[2. Database Relationship Diagram 43](#_Toc420758279)

[2.1 Physical Diagram 43](#_Toc420758280)

[2.2 Data Dictionary 43](#_Toc420758281)

[3. Performance Measures 43](#_Toc420758282)

[3.1 Clustering Performance 44](#_Toc420758283)

[4. Test Plan 44](#_Toc420758284)

[4.1 Features to be tested 44](#_Toc420758285)

[4.2 Features not to be tested 44](#_Toc420758286)

[5. System Testing Test Case 44](#_Toc420758287)

[5.1 Guest Test Case 46](#_Toc420758288)

[F. Software User’s Manual 46](#_Toc420758289)

[1. Installation Guide 46](#_Toc420758290)

[1.1 Setting up environment at server side 46](#_Toc420758291)

[1.2 Deployment at server side 47](#_Toc420758292)

[1.3 Setting up the environment at client side 47](#_Toc420758293)

[2. User Guide 47](#_Toc420758294)

[G. Appendix 47](#_Toc420758295)

# List of Tables

[Table 1: Roles and Responsibility 8](#_Toc408512547)

# List of Figures

No table of figures entries found.

# Definitions, Acronyms, and Abbreviations

|  |  |
| --- | --- |
| **Name** | **Definition** |
|  |  |

# Introduction

## Project Information

* Project name: **Office Task Management**
* Project Code: **OTM**
* Product Type: **Website, Mobile app**
* Start Date: **May 13th, 2015**
* End Date: **August 22th, 2015**

## Introduction

Nowadays, business organizations have a very large number of tasks every day, so managers may cause mistakes in tracking work and managing tasks. To solve problems of using handwriting to track tasks and manage works, a solution that will support manager to manage and monitor progress tasks and help members track project’s status in needed.

From actual needs, we give the idea of developing a product that helps manage office tasks for business uses. Besides website version, we will develop one more mobile version. Specifically, this version will run based on Android OS.

With this product, we hope that user will follow up tasks more easily.

## Current Situation

Current systems can support users follow up and manage tasks on website, but don’t have mobile version or vice versa. Besides, current systems have comment function on website, but cannot discuss with real-time collaboration.

All of current systems are web service that allows users create project then add tasks into project simply and freely. This model is not suitable with complex groups like a company.

## Problem Definition

Below are the advantages and disadvantages of current office task management software:

* Advantages:
* Have a nice and simple user-interface, so user can manage tasks more easy.
* Have support review document, download or upload files, and manage groups.
* Disadvantages:
  + - Don’t have support report progress.
    - Lack of productivity assessments.
    - Don’t have discuss function with real-time collaboration on website and mobile.

## Proposed Solution

Our project includes website and mobile app not only let managers track tasks and manage processes, but also help assessing productivity of work and creating convenient communication environment.

### **Feature functions**

* Assign task
* Document review and approval
* Task review and approval
* Group discussion
* Productivity assessment
* Report

### **Advantage and disadvantage**

* Advantages:
  + Easy to use
  + Simple interface
  + Easy to divide task
  + Approve document fast and easily
  + Discuss group in real-time
  + Productivity assessment based on effort and difficult-level of task
  + Report progress of work
* Disadvantages:
  + Only for small companies
  + Report feature is simple
  + Business constraint of this system may not be applicable to some others kind of company.

## Functional Requirement

Function requirements of the system are listed as below:

### **User Management**

* Add, edit, delete account
* Manage user privileges

### Tracking task

* Assign task
* Monitor task
* View, update, delete task
* Search task

### Review and approve document

* Upload/ Download document
* Approve document

### Collaborate in real-time

* Chat group
* See history

### Statistic

* Report
* Productivity assessment

## Role and Responsibility

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| No | Full Name | Role | Position | Contact |
| 1 | Lại Đức Hùng | Project Managers | Instructor | [Hungld@fpt.edu.vn](mailto:Hungld@fpt.edu.vn) |
| 2 | Nguyễn Cự Đạt | Developer | Leader | [Datncse601010@fpt.edu.vn](mailto:Datncse601010@fpt.edu.vn) |
| 3 | Lư Cẩm Toàn | Developer | Member | [Toanlcse60633@fpt.edu.vn](mailto:Toanlcse60633@fpt.edu.vn) |
| 4 | Lê Hải Triều | Developer | Member | [Trieulhse61146@fpt.edu.vn](mailto:Trieulhse61146@fpt.edu.vn) |
| 5 | Trần Thị Kim Anh | Developer | Member | [Anhttkse60909@fpt.edu.vn](mailto:Anhttkse60909@fpt.edu.vn) |

# Software Project Management Plan

## Problem Definition

### Name of this Capstone project

* **Office Task Management (OTM)**

### Problem Abstract

* Hiện nay, công việc trong 1 công ty, 1 phòng ban có số lượng khá nhiều trong 1 ngày. Và đa số các công ty làm việc trên giấy tờ, việc đó dẫn đến nhiều vấn đề như thất lạc tài liệu, gây khó khan trong việc quản lý công việc và đánh giá sau này. Office Task Management là 1 phần mềm giúp người dùng có thể nắm rõ được công việc đang làm, nó giúp cho nhân viên có thể nắm rõ và theo sát công việc được giao, giúp người quản lý theo dõi thành viên trong nhóm và dự án đang thực hiện, giúp (director) có thể nắm được công việc của nhân viên trong công ty và hỗ trợ đánh giá dự án và nhân viên mang tính chính xác cao.

### Project Overview

#### Current Situation

* Advantages:
  + Có giao diện thân thiện với người dùng.
  + Có phiên bản trên điện thoại, giúp cho người quản lý thuận tiện trong việc giám sát công việc.
  + Có hỗ trợ đánh giá tasks và projects, phục vụ cho việc đánh giá nhân viên theo từng thời kì trong năm.
  + Hỗ trợ forum để nhân viên trong cùng project thảo luận công việc.
* Disadvantages:
  + Chưa hỗ trợ được tính năng chat.
  + Hiện tại trên thực tế đã có nhiều hệ thống quản lý công việc văn phòng.
  + Phiên bản mobile chỉ hỗ trợ giám sát công việc cho người quản lý.

#### The proposed system

Để giải quyết tình hình hiện tại, nhóm chúng tôi đề xuất 1 dự án xây dựng hệ thống quản lý các công việc văn phòng. Hệ thông được xây dựng trên cả 2 phiên bản website và mobile.

##### Web Site

Hệ thống được hiện thực thành 1 trang web với những chức năng chính

* Manage project: director và manager có thể quản lý projects trong công ty.
* Manage task: manager và staff có thể quản lý và theo dõi các tasks trong projects.
* Group discussion: user có thể thảo luận trong forum của từng project hoặc comment vào task.
* Report tasks and projects: hệ thống sẽ thống kê tình trạng dự án hỗ trợ người dùng dễ dàng theo dõi.
* Assess employee in period time: người quản lý có thể đánh giá nhân viên theo từng đợt đánh giá định kỳ trong năm.

##### Mobile Application

* View Project: người cùng có thể theo dõi các projects do họ quản lý.
* View Task: người dùng có thể theo dõi các tasks trong từng project mà họ quản lý.

#### Boundaries of the System

* The system can be used by everybody who have basic knowledge about IT and have an Android smartphone.
* The language of system is English.
* The completed product includes:
  + Website version for all roles of user.
  + Mobile version for director and manager.
  + All the process document involved.

#### Development Environment

##### Hardware requirements

**For Server**

|  |  |  |
| --- | --- | --- |
| **Windows** | **Minimum Requirements** | **Recommended** |
| **Internet Connection** | Cable, Wi-Fi (4 Mbps) | Cable, Wi-Fi (12 Mbps) |
| **Operating System** | Window Server 2008 | Window Server 2008 |
| **Computer Processor** | Intel® Xeon ® 1.4GHz | Intel® Xeon ® Quad Core  (12M Cache, 2.50 GHz) |
| **Computer Memory** | 1GB RAM | 2GB or more |

**For Mobile**

|  |  |  |
| --- | --- | --- |
| **Windows** | **Minimum Requirements** | **Recommended** |
| **Internet Connection** | Cable, Wi-Fi (4 Mbps) | Cable, Wi-Fi (12 Mbps) |
| **Operating System** | Android 4.1.2 | Android 4.4.2 |
| **Hardware** | Intel® Xeon ® 1.4GHz | Intel® Xeon ® Quad Core  (12M Cache, 2.50 GHz) |
| **Computer Memory** | 1GB RAM | 2GB or more |

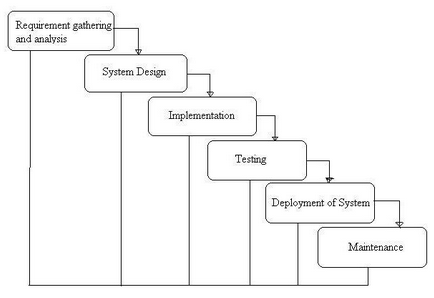
##### Software requirements

* Window Server 2008: operating system and platform for development.
* SQL Server 2008 Enterprise R2: used to create and manage the database for system.
* Eclipse Luna 4.4.2: used to implement website and web service.
* GitHub & TortoiseSVN: used for source control.
* StarUML, [www.lucidchart.com](http://www.lucidchart.com): used to create models and diagrams.
* Justinmind: used to create GUI prototypes.
* Skype, Garena: used for communication and meeting.

## Project organization

### Software Process Model

Project is developed under modified waterfall model.



For more information: <http://istqbexamcertification.com/what-is-waterfall-model-advantages-disadvantages-and-when-to-use-it/>

### Roles and responsibilities

|  |  |  |  |
| --- | --- | --- | --- |
| **No** | **Full name** | **Role in Group** | **Responsibilities** |
| **1** | Lại Đức Hùng | Project manager | * Specify user requirement * Control the development process * Give out technique and business analysis |
| **2** | Nguyễn Cự Đạt | Team Leader,  BA, Dev, Tester | * Managing process * Clarifying requirements * Prepare documents * GUI Design * Designing database * Coding * Create test plan * Testing |
| **3** | Lư Cẩm Toàn | Team Member,  BA, Dev, Tester | * Clarifying requirements * Prepare documents * GUI Design * Designing database * Coding * Create test plan * Testing |
| **4** | Lê Hải Triều | Team Member,  BA, Dev, Tester | * Clarifying requirements * Prepare documents * GUI Design * Designing database * Coding * Create test plan * Testing |
| **5** | Trần Thị Kim Anh | Team Member,  BA, Dev, Tester | * Clarifying requirements * Prepare documents * GUI Design * Designing database * Coding * Create test plan * Testing |

### Tools and Techniques

* Front-end technologies: HTML5, CSS3, JavaScript, jQuery, AJAX.
* Back-end technologies:
  + Website: Java EE
  + Web service: Java + Jersey
  + Third-party libraries:
* Mobile: Android JellyBean 4.1.2 & Android KitKat 4.4.2 –Java 7
* WebServer: Apacha Tomcat 7.0.61
* Database Management System: MSSQL Server 2008 R2 Enterprise

## Project Management Plan

### Software development life cycle

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Phase** | **Description** | **Deliverables** | **Resource needed** | **Dependencies and Constrains** | **Risks** |
| **Requirement gathering and Analysis** | - Gather requirement from customers.  - Study business in reality to determine business rules.  - Identify and clarify requirement base on real business. | - Introduction of system.  - Software project management plan.  - Software requirement spectification.  - Business rules.  - Protoypes. |  | N/A | - Các thành viên trong nhóm không đồng nhất về cách hiểu requirement.  - Hiểu nhầm về business. |
| **System Design** | - Determine architecture style which will be used.  - Design architecture for the system.  - Solve problems and plan solutions. | - Software design document.  - Architectual style for code.  - Technologies for the system. |  | Depend on “Requirement gathering and analysis”. | - Lack of knowledge and experience.  - Lack of understanding requirement |
| **Implementation** | - Code GUI.  - Code core functions and other features.  - Unit test and debug. | - Web application.  - Mobile application on Android. |  | Depend on “System design”. | - Mistake while implementing |
| **Testing** | - Unit testing.  - Integration testing.  - System testing.  - Fix bugs. | - Test document. |  | Depend on “Implemention”. | - Lack of experience.  - Lack of test cases. |
| **Deployment of System** | - Deploy on web server and mobile. | - Installation guide.  - User Manual |  | Depend on “Testing”. | - Lack of experience. |
| **Maintenance** | - Maintain and update the system. | N/A |  | Depend on “Deployment of system”. | N/A |

### Phase Detail

#### 10.2.1 Phase 1: Requirement gathering and analysis

|  |  |  |
| --- | --- | --- |
| **Task** | **Description** | **Author** |
| 1. Gather requirement | - Collect requirement from customer.  - Study from current similar system. | DatNC, ToanLC, TrieuLH, AnhTTK |
| 2. Study business | - Define strengths and weakness of current similar system.  - Interview some real PMs about business. | DatNC, ToanLC, TrieuLH, AnhTTK |
| 3. Identify and clarify requirement | - Define main scope of system. | DatNC, ToanLC, TrieuLH, AnhTTK |
| 4. Create system introduction, SRS document | - Complete introduction of system  - Write SRS document | DatNC, TrieuLH |
| 5. Design prototypes | - Design GUI prototypes by tools | ToanLC, AnhTTK |

#### 10.2.1 Phase 2: System design

|  |  |  |
| --- | --- | --- |
| **Task** | **Description** | **Author** |
| 1. Research technologies | - Research third-party’s APIs support real-time collaboration and push notification.  - Study Android application development.  - Study web services using Jersey | DatNC, ToanLC, TrieuLH, AnhTTK |
| 2. Design ERD | - Create ER diagram. | DatNC, ToanLC, TrieuLH, AnhTTK |
| 3. Design database | - Design logical and create physical database | DatNC, TrieuLH |
| 4. Design Class diagram | - Create Class diagram | DatNC, ToanLC, TrieuLH, AnhTTK |
| 5. Detail design and create design document | - Top-down design.  - Create and update software design document | DatNC, ToanLC, TrieuLH, AnhTTK |

#### 10.2.1 Phase 3: Implemention

|  |  |  |
| --- | --- | --- |
| **Task** | **Description** | **Author** |
| 1. Implement web front-end | - Code GUI for web application | DatNC, ToanLC, TrieuLH, AnhTTK |
| 2. Implement web back-end | - Code functions for web application | DatNC, ToanLC, TrieuLH, AnhTTK |
| 3. Implement mobile front-end | - Code GUI for mobile application |  |
| 4. Implement mobile back-end | - Code functions for mobile application |  |
| 5. Unit testing | - Unit test  - Fix bugs |  |

#### 10.2.1 Phase 4: Testing

|  |  |  |
| --- | --- | --- |
| **Task** | **Description** | **Author** |
| 1. Unit testing | - Write test cases  - Test and fix bugs. |  |
| 2. Integration testing | - Write test cases  - Test and fix bugs. |  |
| 3. System testing | - Write test cases  - Test and fix bugs. |  |
| 3. Create test document | - Create and update test document |  |

#### 10.2.1 Phase 5: Deployment of System

|  |  |  |
| --- | --- | --- |
| **Task** | **Description** | **Author** |
| 1. Deployment | - Deploy system on server and mobile |  |
| 2. Create installation guide | - Write installation guide |  |
| 3. Create user manual | - Write user manual |  |

#### 10.2.1 Phase 5: Maintenance

|  |  |  |
| --- | --- | --- |
| **Task** | **Description** | **Author** |
| 1. Maintain and update system | - Maintain current system  - Update new features |  |

### All Meeting Minutes

Place at Meeting Minutes folder.

## Coding Convention

Use Java coding convention to develop website, web service and mobile application.

Covention:

* Naming convention:
  + Use camel case for name of variables and functions.
  + Use Pascal case for class
* Indentation:
  + Avoid lines longer than 80 characters, since they are not handled well by many terminals and tools.
* Declaration:
  + In absolutely no case should variables and functions be declared on the same line.
  + Do not put different types on the same line.

Follow “Code Conventions for the Java TM Programming Language, by Sun Microsystems, rev April 20, 1999”.

<http://www.oracle.com/technetwork/java/codeconventions-150003.pdf>

# Software Requirement Specification

## User Requirement Specification

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

**Ví dụ**

### 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.
* Search event.
* View event detail.
* Booking and payment for event tickets.

### Member Requirement

### Partner Requirement

### Staff Requirement

### Admin Requirement

## System Requirement Specification

### External Interface Requirement

#### User Interface

<Các yêu cầu về giao diện>

**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:
  + UI for businesss web applications - Janko Jovanovic [Ref: [http://www.smashingmagazine.com/2010/02/25/designing-](http://www.smashingmagazine.com/2010/02/25/designing-user-interfaces-for-business-web-applications/) [user-interfaces-for-business-web-applications/]](http://www.smashingmagazine.com/2010/02/25/designing-user-interfaces-for-business-web-applications/)
  + Ten principles of effective web design – Vitaly Friedman [Ref: [http://www.smashingmagazine.com/2008/01/31/10-](http://www.smashingmagazine.com/2008/01/31/10-principles-of-effective-web-design/) [principles-of-effective-web-design/]](http://www.smashingmagazine.com/2008/01/31/10-principles-of-effective-web-design/)
  + Principles of mobile interface design – Jonathan Stark [Ref: <http://www.oreilly.com/pub/e/2144>]

#### Hardware Interface

#### Software Interface

<Yêu cầu về phần mềm chú ý ghi rõ phiên bản cũng như kích thước màn

hình>

#### Communication Protocol

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

### 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/>

**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**

**Ví dụ**



PAGE \\* MERGEFORMAT 1

**Figure 1: System Overview Use Case**

### 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

**Ví dụ**

#### <Guest>Overview Use Case



**Figure 2: <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

**Ví dụ**

#### <Guest> Register

**Use Case Diagram**



**Figure 3: <Guest>Register**

**Use Case Specification**

GuideLine: Đây là giai đoạn lấy requirement nên đừng quá nghiên về phía

giao diện ứng dụng sau khi làm xong. 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

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

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

|  |  |  |  |
| --- | --- | --- | --- |
| **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 hoặc lựa chọn của người dùng>**  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 – MSSC001** | | | | | | | |
| **Use Case No.** | | | MSSC001 | | **Use Case Version** | 2.0 | |
| **Use Case Name** | | | View Friend List | | | | |
| **Author** | | | HuyDN | | | | |
| **Date** | | | 12/09/2014 | | **Priority** | High | |
| **Actor:**  - Member.  **Summary:**  - This use case allows member to view all contacts that includes MSSC contacts and android cellphone contacts.  **Goal:**  - Member can view all of their contacts.  **Triggers:**  - Member touches tab “Tất Cả Danh Bạ” to view the list of contacts.  - Member touches tab “Danh Bạ MSSC” to view the list of MSSC contacts.  - Member touches tab “Danh Bạ Android” to view the list of Android contacts.  **Preconditions:**  - Member must login into the system.  **Post Conditions:**  - **Success:** Show list of all contacts.  - **Fail:** Show error message.  **Main Success Scenario:** | | | | | | | |
|  | Step | Actor Action | | System Response | | |  |
| 1 | Member touches tab “Tất Cả Danh Bạ” to view all contacts list.  [Alternative 1] [Alternative 2] | | Show list of contacts as a grid which shows the following information:  - “Tên liên lạc”: label  - “Số điện thoại”: label  - “update” : image | | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  |  | - “friendship” : image  [Alternative 3] [Exception 1] |  |
| **Alternative Scenario:**  No Actor Action System Response  1 Member touches tab “Danh  Bạ MSSC” to view all MSSC  contacts. Show list of contacts as a grid which shows the following information:  - “Tên liên lạc”: label  - “Số điện thoại”: label  - “update” : image  - “friendship” : image  [Alternative 3] [Exception 1]  2 Member touches contact tab  “Danh Bạ Điện Thoại” to view  all android cellphone Show list of contacts as a grid which contacts. shows the following information:  - “Tên liên lạc”: label  - “Số điện thoại”: label  [Alternative 3] [Exception 1]  3 Member doesn’t have any  contact in contact list. Show message: “Danh sách liên hệ trống”.  **Exceptions:**  No Actor Action System Response  1 Connect to mobile database Show message “khởi động lại ứng dụng”.  fail  **Relationships:** extended by Save To Phone (Select save phone), extended by Remove Contact (Select a contact to remove), extended by Update Friend Contact (Select update), and extended by View Contact Detail (Select contact).  **Business Rules:**  - List of contacts are alway getted from mobile database.  - When having internet connection, Sync happen to synchronize mobile database and server database.  - List of contacts are sorted by alphabet based on alias.  - Search bar on the top and fast scroll bar on the right help user finding contact faster.  - “friendship” and “update” image are only shown on contacts which are an  MSSC contact.  - Every time member login into system in online mode, application will use web  service to check and show “update” image and “friendship” image on contacts.  - “friendship” image is only shown on a contact which is having friendship.  - Friendship contacts can update information when the owner of it update their | | | | |

PAGE \\* MERGEFORMAT 1

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

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

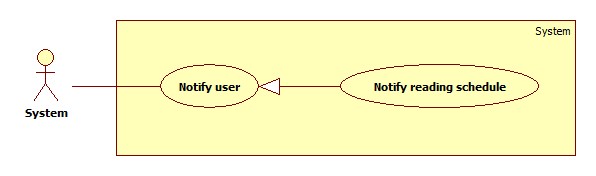
profile.

- “update” image is only shown on a friendship contacts in online mode, when

current information are different from database.

- Member can keep old information of a contact.

- Member can call or make text message to a contact by touch and hold it.



**Figure 4: <System> Notify reading schedule use case diagram**

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

**Use Case Specification**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **USE CASE – ARB06** | | | | | | | |
| **Use Case No.** | | | ARB06 | | **Use Case Version** | 2.0 | |
| **Use Case Name** | | | Notify reading schedule | | | | |
| **Author** | | | Trinh Van Vu | | | | |
| **Date** | | | 30/05/2014 | | **Priority** | High | |
| **Actor:**  - System  **Summary:**  - System can notify reading schedule to user when user login  **Goal:**  - Notify user when reading time start and end.  **Triggers:**  - User log in the system  **Preconditions:**  - User bought book  - User configured schedule  - User log in the system  **Post Conditions:**  - **Success:** Notify user reading time at start reading time and end reading time.  - **Fail:** Log error to administrator.  **Main Success Scenario:**  Step Actor Action System Response  1 Server checks user reading schedule.  Send response when the time reach start and end time [Alternative 1][Exception 1]  **Alternative Scenario:** | | | | | | | |
|  | No | Actor Action | | System Response | | |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | 1 | Time is out of start time and  end time. | Show notification “Bạn đã hết giờ đọc sách” |  |
| **Exceptions:**  No Actor Action System Response  1 Cannot send notify from server Log error message to administrator  **Relationships:** N/A  **Business Rules:**  - System compares current time and end time in schedule which is configured by user. (from server)  + Current time is more than end time in schedule: Response reading time at start reading time and end reading time which is configured by user.  + Current time is less than end time in schedule: Response to user is time’s out of  start time and end time  - User can receive notifications at the time when system sends notification. | | | | |

**Table 6: <System> Notify reading schedule use case specification table**

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

System

**Auto parse**

**System**

**Figure 5: <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 database. Log file is generated.  - **Fail:** Nothing is changed in the database. 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 database  [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 database, update its information.   Generate log file.  **Exceptions:**  No Actor Action System Response  1 Data is invalid.  Don’t insert to database.   Generate log file.  **Relationships:** N/A  **Business Rules:**  - If link resource exists in database, do not insert to database.  - If link resource is not active, do not insert to database.  - 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 7: Auto parse use case specification table**

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

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

## Software System Attribute

<Mô tả nonfunctional, biết thì ghi, ghi thì phải có dẫn chứng về việc đã đo đạc,

có định lượng như thế nào>

### Usability

### Reliability

### Availability

### Security

### Maintainability

### Portability

### Performance

…..

## Conceptual Diagram

<Hình mô tả về conceptual diagram của nhóm: chú ý chỉ sử dụng một tập kí

hiệu và cần tham khảo 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

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

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

# Software Design Description

## Design Overview

<Tham khảo>

- 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 **(HOW)**

## System Architectural Design

<Kiến trúc hệ thống mà nhóm xây dựng: chú ý hình trên mạng là không chính

xác tự vẽ sẽ là chính xác nhất, không sao chép, vay mượn và chế kí hiệu. Nếu dùng kí hiệu ngoài UML thì vui lòng ghi chú giải kí hiệu ngay cạnh hình vẽ.>

<Giải thích lý do tại sao lựa chọn mô hình này dựa trên SRS và Introduction của dự án đang thực hiện.>

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

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

**Ví dụ**

Nếu ứng dụng làm web và mobile thì việc đặc tả theo mục 2.1 và 2.2

### Web application architecture description

<Mô tả cụ thể>

### Mobile application architecture description

<Mô tả cụ thể>

## Component Diagram

<Thể hiện góc nhìn hệ thống dưới sự tương tác của các thành phần component. 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** |
|  |  |

## Detailed Description

### 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>

Lời khuyên là tự vẽ, tránh lệ thuộc công nghệ

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

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

### Class Diagram Explanation

<Mô tả cụ thể cho các lớp>

Ví dụ

#### Role

Attribute

|  |  |  |  |
| --- | --- | --- | --- |
| **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 |

### Interaction Diagram

#### Tên Interaction Diagram

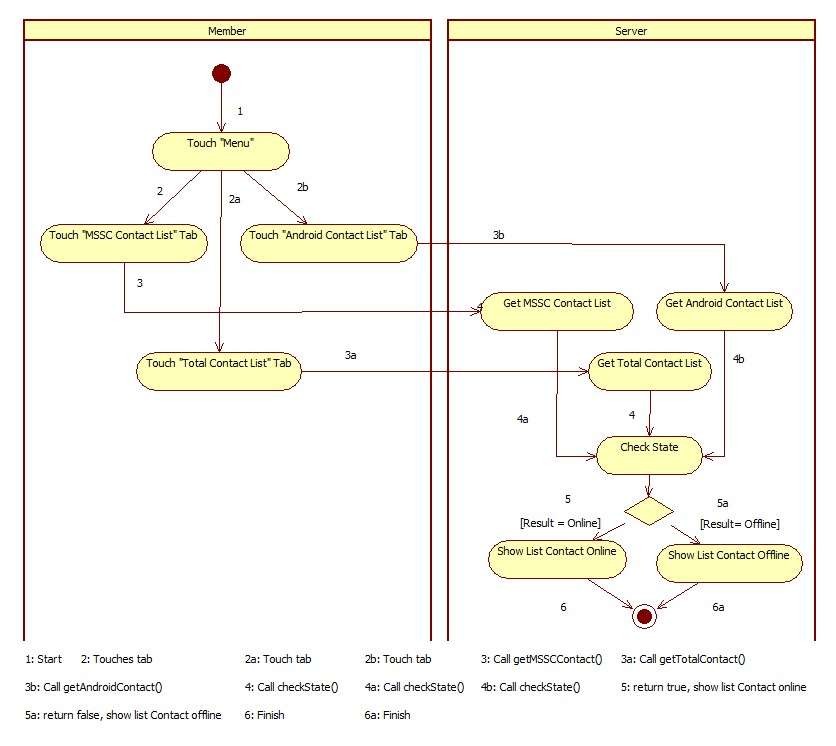
**Summary:** <Nên có phần tóm tắt trước 1 diagram nói về mục

đích của diagram trước khi thể hiện hình vẽ>.

**Ví dụ**

##### <Member> View Friend List

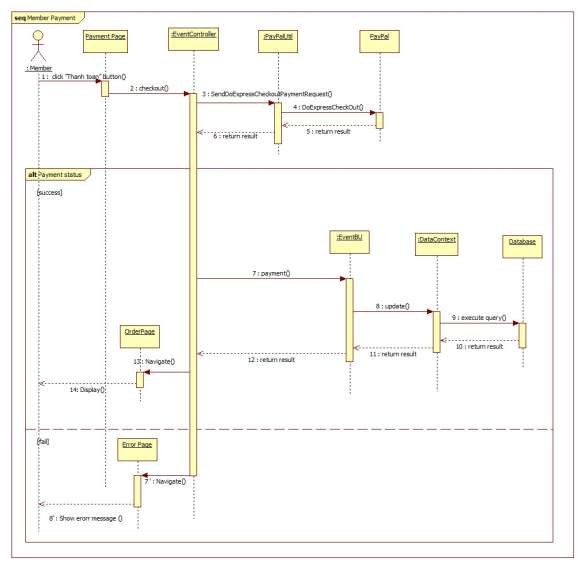
**Summary:** This diagram shows how member views all contacts that include MSSC contacts and android cell phone contacts.



**Figure 6: <Member> View Friend List**

#### <Member> Payment

**Summary:** This diagram shows how members make payment for a booking.



**Figure 7: <Member>Payment**

## User 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

Ví dụ

### Guest Interface Design

#### Login



**Figure 8: Login**

**Fields**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **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 |

## Database Design

### Logical Diagram

<Thiết kế logical>

### Data Dictionary

<Mô tả về các thực thể>

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

**Table 8: Entity Data Dictionary**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Entity name** | **Attributes** | **Description** | **Domain** | **Null** |
| Role | RoleID {PK} | Unique identifier of role, auto increment. | int | No |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Name | Role name. | varchar(20) | No |

**Table 9: Detail Data Dictionary**

\* Business integrity constraint:

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

## Algorithms

<Các thành phần thuật toán 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**

### Document Breakdown

#### 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.

#### Define Problem

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

#### 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

#### Complexity

- In total, the complexity of this algorithm is 

#### Flowchart

**Figure 9: Breakdown document flow chart**

### String Comparison

#### Define Problem

Given two strings. Calculate their matching percent.

#### 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.

#### 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.



#### 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.



# System Implementation & Test

## Introduction

### 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.

### Test Approach

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

## Database Relationship Diagram

### Physical Diagram

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

hiện tôn trọng với người xem tài liệu>

### Data Dictionary

Mô tả thành phần theo bảng biểu bên dưới

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

**Table 10: Data Dictionary**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Table name** | **Attributes** | **Description** | **Domain** | **Null** |
| Role | RoleID {PK} | Unique identifier of role, auto increment. | int | No |
| Name | Role name. | varchar(20) | No |

**Table 11: Attribute Data Dictionary**

## Performance Measures

<Cách nhóm ước lượng cũng nhu đo đạc hệ thống>

**Ví dụ**

### 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.

## 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.

### Features to be tested

<Tính năng sẽ kiểm thử>

### Features not to be tested

<Tính năng sẽ không kiểm thử>

## 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 10: Guest, Member Core Flow**

MSSC - Introduction

### Guest Test Case

#### Search Event

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

MSSC - Introduction

# Software User’s Manual

## Installation Guide

### Setting up environment at server side

The following software must be installed into the server machine:

#### Hardware requirements

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

#### Software requirements

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

### 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>

#### Prepare deployment package

#### Configure Server before deploy

#### Deploy web application on server

### Setting up the environment at client side

#### Setting up for computer

<Ghi rõ phiên bản tối thiểu để sử dụng>

## User Guide

<Viết hướng dẫn sử dụng cho người dùng>

# 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)