**MINISTRY OF EDUCATION AND TRAINING**

**FPT UNIVERSITY**

Capstone Project Document

The Roll System using Mobile Device

|  |  |
| --- | --- |
| **Group 1** | |
| **Group member** | Pham Huy Hoang – Team Leader – 60740  Nguyen Thanh Binh – Team Member - 60321  Nguyen Quoc Huy – Team Member - 60551  Do Minh Dat – Team Member - 60545 |
| **Supervisor** | Mr. Kieu Trong Khanh |
| **Ext. Supervisor** | N/A |
| **Capstone Project code** | RSM |

-Ho Chi Minh City, 09/2013-

*This page is intentionally left blank*

***ACKNOWLEDGEMENTS***

We wish to thank various people for their contribution to this project: Our teachers for their advices and participant in the final review, our friend for the valuable technical support.

Special thanks should be given to Mr.Kiều Trọng Khánh, our research supervisor for his professional guidance and the useful, constructive recommendations throughout the course of this project.

# Table of Contents

[**Table of Contents 4**](#_Toc366867001)

[**List of Tables 6**](#_Toc366867002)

[**List of Figures 7**](#_Toc366867003)

[A. Introduction 8](#_Toc366867004)

[1. Project Information 8](#_Toc366867005)

[2. Introduction 8](#_Toc366867006)

[3. Current Roll Call System 8](#_Toc366867007)

[4. Problem Definition 8](#_Toc366867008)

[5. Proposed Solution 9](#_Toc366867009)

[5.1 Feature functions 9](#_Toc366867010)

[5.2 Advantage and disadvantage 9](#_Toc366867011)

[6. Functional Requirement 10](#_Toc366867012)

[6.1 Create Account 10](#_Toc366867013)

[6.2 Instructor Management 10](#_Toc366867014)

[6.3 Class Management 10](#_Toc366867015)

[6.4 Student Management 10](#_Toc366867016)

[6.5 Course Management 10](#_Toc366867017)

[6.6 Roll Call Management 10](#_Toc366867018)

[6.7 Attendance Checking 10](#_Toc366867019)

[6.8 Stranger Alert 10](#_Toc366867020)

[6.9 Attendance Report 11](#_Toc366867021)

[7. Role and Responsibility 11](#_Toc366867022)

[B. Software Project Management Plan 12](#_Toc366867023)

[1. Problem Definition 12](#_Toc366867024)

[1.1 Name of this Capstone Project 12](#_Toc366867025)

[1.2 Problem Abstract 12](#_Toc366867026)

[1.3 Project Overview 12](#_Toc366867027)

[2. Project organization 14](#_Toc366867028)

[2.1 Software Process Model 14](#_Toc366867029)

[2.2 Roles and responsibilities 15](#_Toc366867030)

[2.3 Tools and Techniques 16](#_Toc366867031)

[3. Project Management Plan 17](#_Toc366867032)

[3.1 Iteration 17](#_Toc366867033)

[3.2 Iteration Detail 19](#_Toc366867034)

[3.3 All Meeting Minutes 23](#_Toc366867035)

[4. Coding Convention 23](#_Toc366867036)

# List of Tables

[Table 1: Roles and Responsibility 11](#_Toc366867037)

[Table 2: Hardware Requirement for Server 14](#_Toc366867038)

[Table 3: Hardware Requirement for Mobile App 14](#_Toc366867039)

[Table 4: Roles and Responsibility Details 16](#_Toc366867040)

[Table 5: Iteration 18](#_Toc366867041)

[Table 6: Phase 1: Preliminary Investigation or Analysis 19](#_Toc366867042)

[Table 7: Phase 2: Face Detect & Recognize 20](#_Toc366867043)

[Table 8: Phase 3: Student Management 20](#_Toc366867044)

[Table 9: Phase 4: Roll Call Management 21](#_Toc366867045)

[Table 10: Phase 5: Web Service Implement 22](#_Toc366867046)

[Table 11: Phase 6: Attendance Checking 22](#_Toc366867047)

[Table 12: Phase 7: Attendance Report 23](#_Toc366867048)

# List of Figures

[Figure 1: Agile Development Model 15](#_Toc366867050)

# Introduction

## Project Information

* Project name: **The roll system using mobile device**
* Project Code: **RSM**
* Product Type: **Website, Phone Application**
* Start Date: **September 9th, 2013**
* End Date:

## Introduction

Roll system was known with HPLite32, SimplePass of HP fingerpint system; roll system with ID card using by almost corporation or company around the world; the system face identify by Uniqul – Finland publish on 7/15/2013 or LogonSmart by Asus. Today, the trend machines replace humans in the hard work or the work does not require high intelligence became popular. So roll system face recognition becomes ever more necessary. The number of students in a university as well as the number of employees in large corporations in Vietnam is increasing dramatically, which means that the system should have professional roll with accuracy high.

## Current Roll Call System

Below are some current roll call systems:

* By manual: This system is widely used in school, university. The instructor will call the name of each student, check the absent, then submit the result to log system.
* Using ID card: This system is usually used in corporations. Each employee has a card. The card will be read by a card reader to check the attendance of the employee.
* Using fingerprint: This system is currently used in FPT University. 15 minutes before and after a studying session, the student must show his fingerprint to a machine to take attendance.
* Using camera + face recognition: This system is just in experimental stage. Each classroom has a camera. At the beginning of the studying session, the instructor uses the computer, connect to the camera and take picture of classroom. The system will recognize the student in the picture, write to roll call log system.

## Problem Definition

Below are the advantage and disadvantage of current roll call systems:

* By manual

+ Advantage: Simple to implement, cheap.

+ Disadvantage: The roll call take 3-5 minutes, take effort of instructor, wrong roll call.

* Using ID card

+ Advantage: High accuracy, not take much effort to check attendance.

+ Disadvantage: High cost (ID Card, Card Reader). Risk of ID card missing.

* Using fingerprint

+ Advantage: Quickly. No effort must be made from instructor.

+ Disadvantage: Cost of fingerprint reading machine. Risk of machine error. The student checks the attendance but not go to class.

* Using camera + face recognition

+ Advantage: Quickly. The entire class picture can be stored as log.

+ Disadvantage: High cost (Camera cost). Face recognition not too accuracy.

## Proposed Solution

The system is intended for used in only school or university, where the pupils/students sit in a classroom (A small classroom, from 20-30 students). The system must to manage the course, teacher, check attendance…. In detail, the system will enable following function:

### **Feature functions**

* The admin can manage information about course, class, instructor, students.
* The system will provide a method for admin to upload the student’s images, select the students in the images to make training data.
* The system will make a roll call list (Contains: Course, Class, Time, Instructor, Student List) for each class, based on the input information.
* The system will provide a method to assign instructor and student to course. The system must check the availability of the instructor before assigning.
* The system must support a method to change instructor of a course (When instructor is sick or busy).
* The system can output report about the absent rate of a course, or a student.
* The instructor/examiner (fix other place) can view info about: What course they are teaching. Roll Call and student lists of these classes.
* The instructor will use the mobile application. At the beginning of a studying session, a picture of entire class will be taken for checking attendance.
* Base on the student list, the system will recognize the students from the picture. The mobile app will notify attendance result to instructor.
* The instructor can, re-check attendance manually.
* The system will alert the instructor when it detects stranger in classroom.
* Based on the respond from instructor, the system will gather more data to become more accuracy.
* The student can view info about what course they’re studying.
* The student can view their own attendance of the course they participated in.

### Advantage and disadvantage

The advantage and disadvantage of the proposed solution:

* Advantage: Small cost or no cost (If the instructor has phone). Face Recognition accuracy is about 70~80%, will reach 100% with confirm from instructor. The attendance check is quickly (< 30 seconds). The taken picture can be stored as log.
* Disadvantage: Complex to implement. The accuracy can vary depend on: Noise, brightness, number of people, image solution… Need student’s picture to be used for recognition training.

## Functional Requirement

Function requirement of the system are listed as below:

### Create Account

* The account of instructors and students will be auto-created by the system, basing on the input from admin.
* The instructors and students can change their profile and password.

### Instructor Management

* The staff can add/edit/active/inactive instructor.

### Class Management

* The staff can add/edit/active/inactive class.

### Student Management

* The admin can add/edit/active/inactive student.
* Each student must has a training set, contains 8-20 images for face recognizing. The staff can upload images, select the faces of the students to add. The staff can also remove images from training set.

### Course Management

* The staff can add/edit/active/inactive course.

### Roll Call Management

* The staff can add/edit/active/inactive roll call.
* The staff must input info of a roll call: The course, the instructor who teaches the course, the student list, the time of the course (Begin, End Date).
* The instructor can see what roll call they belong to.
* The instructor can change their roll call to another instructor, in case of sickness or busy.

### Attendance Checking

* At the beginning of each studying session, the instructor takes a picture of entire class for attendance checking.
* The instructor can re-check attendance manually.
* The mobile app will notify the result to the instructor.
* In case of wrong recognition, the student face will be added to training set for higher accuracy.
* The instructor can view the attendance log of the course they are teaching.
* The student can view the attendance log of the course they are studying.

### Stranger Alert

* If stranger is detected in the classroom, the system will alert to teacher. A log file will also be written.
* In case of wrong recognition, the student face will be added to training set for higher accuracy.

### Attendance Report

* The system will make attendance report, which shows the rate of absence in a course, or the rate of absence of a student.
* The report is only available to staff and instructor

## Role and Responsibility

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No** | **Full Name** | **Role** | **Position** | **Contact** |
| 1 | Kiều Trọng Khánh | Project Manager | Instructor | khanhkt@fpt.edu.vn |
| 2 | Phạm Huy Hoàng | Developer | Team Leader | hoangphse60740@fpt.edu.vn |
| 3 | Nguyễn Thanh Bình | Developer | Team Member | [binhnt60321@fpt.edu.vn](mailto:binhnt60321@fpt.edu.vn) |
| 4 | Nguyễn Quốc Huy | Developer | Team Member | [huynq60551@fpt.edu.vn](mailto:huynq60551@fpt.edu.vn) |
| 5 | Đỗ Minh Đạt | Developer | Team Member | datdm60545@fpt.edu.vn |

Table 1: Roles and Responsibility

# Software Project Management Plan

## Problem Definition

### Name of this Capstone Project

The roll system using mobile device (RSM)

### Problem Abstract

Roll system was known with HPLite32, SimplePass of HP fingerpint system; roll system with ID card using by almost corporation or company around the world; the system face identify by Uniqul – Finland publish on 7/15/2013 or LogonSmart by Asus. Today, the trend machines replace humans in the hard work or the work does not require high intelligence became popular. So roll system face recognition becomes ever more necessary. The number of students in a university as well as the number of employees in large corporations in Vietnam is increasing dramatically, which means that the system should have professional roll with accuracy high.

### Project Overview

#### The Current System

Below are some current roll call systems:

* By manual: This system is widely used in school, university. The instructor will call the name of each student, check the absent, then submit the result to log system.

+ Advantage: Simple to implement, cheap.

+ Disadvantage: The roll call take 3-5 minutes, take effort of instructor, wrong roll call.

* Using ID card: This system is usually used in corporations. Each employee has a card. The card will be read by a card reader to check the attendance of the employee.

+ Advantage: High accuracy, not take much effort to check attendance.

+ Disadvantage: High cost (ID Card, Card Reader). Risk of ID card lending, missing.

* Using fingerprint: This system is currently used in FPT University. 15 minutes before and after a studying session, the student must show his fingerprint to a machine to take attendance.

+ Advantage: Quickly. No effort must be made from instructor.

Disadvantage: Cost of fingerprint reading machine. Risk of machine error. The student checks the attendance but not go to class.

* Using camera + face recognition: This system is just in experimental stage. Each classroom has a camera. At the beginning of the studying session, the instructor uses the computer, connect to the camera and take picture of classroom. The system will recognize the student in the picture; write to roll call log system.

+ Advantage: Quickly. The entire class picture can be stored as log.

+ Disadvantage: High cost (Camera cost). Face recognition not to accuracy.

#### The Proposed System

The system is intended for used in only school or university, where the pupils/students sit in a classroom. The system must to manage the course, teacher, check attendance…. In detail, the system will enable following function:

##### Web

* The admin can manage information about course, class, instructor, students.
* The system will provide a method for staff to upload the student’s images, select the students in the images to make training data.
* The system will make a roll call list (Contains: Course, Class, Time, Instructor, Student List) for each class, based on the input information.
* The system will provide a method to assign instructor and student to course. The system must check the availability of the instructor before assigning.
* The system must support a method to change instructor of a course (When instructor is sick or busy).
* The system can output report about the absent rate of a course, or a student.
* Based on the respond from instructor, the system will gather more data to become more accuracy.
* The student can view info about what course they’re studying.
* The student can view their own attendance of the course they participated in.

##### Mobile

* The instructor can view info about: What course they are teaching. Roll Call and student lists of these classes.
* The instructor will use the mobile application to take a picture.
* The mobile shows the list of student present in class, notify the instructor if absent rate is high.
* The instructor can re-check attendance manually.
* The system will alert stranger.

#### Boundaries of the System

* The system is intended for using university, with small classroom, or for the examine room (The testing site will be FPT University).
* The maximum number of a classroom is 30 people. The classroom size is about: 6m x 8m
* The system is not intended for managing these aspect:

+ Managing the teaching calendar of instructor.

+ Managing instructor qualification, salary info.

+ Managing the testing, mark of student of each class.

* The language of the system is English.
* The complete product includes:

+ The website, for admin and students. Instructor can also use the website to change profile, view roll call info.

+ Mobile Application for instructor to check attendance.

+ All the process involved document.

#### Development Environment

##### Hardware requirements

**For server**

|  |  |  |
| --- | --- | --- |
| Windows | Minimum Requirements | Recommended |
| Internet Connection | Cable, Wifi (4 Mbps) | Cable, Wifi (8 Mbps) |
| Operating System | XP, Vista, 7, 8 | XP, Vista, 7, 8 |
| Computer Processor | Intel® Core 2 Duo | Intel® Core(TM) i5 CPU , M 460 @ 2.53GHz |
| Computer Memory | 1GB RAM | 3GB or more |

Table 2: Hardware Requirement for Server

**For Mobile Application**

|  |  |  |
| --- | --- | --- |
| Mobile | Minimum Requirements | Recommended |
| Internet Connection | Wifi (2Mbps) | Wifi (4Mbps) |
| Operating System | Android 4.0 or later version | Android 4.4 |
| Hardware | Touchscreen, Camera 2.0 MP or above | Touchscreen, Camera 4.0 MP or above |
| Memory | 512 MB or more | 1 GB or more |

Table 3: Hardware Requirement for Mobile App

##### Software requirements

* Microsoft Windows 7 Service Pack 1: operating system and platform for development.
* SQL Server 2008 Express:used to create and manage the database for system.
* StarUML: used to create models and diagrams
* Skype: used for communication and meeting
* Visual Studio 2010: used to implement website and web service.
* Eclipse Juno 4.4, Android SDK 22.0.5, ADT 22.0.5 & JDK 7u25: used to implement mobile application.
* Google Code & TortoiseSVN: used for source control.

## Project organization

### Software Process Model

Project is developed under agile model.



Figure 1: Agile Development Model

For more information: <http://www.indicthreads.com/1439/quick-introduction-to-agile-software-development/>

(Owner: IndicThreads.com. Online Software Developer Magazine and Conferences)

### Roles and responsibilities

|  |  |  |  |
| --- | --- | --- | --- |
| **No** | **Full name** | **Role in Group** | **Responsibilities** |
| **1** | Kiều Trọng Khánh | Project manager | * Specify user requirement * Control the development process * Give out technique and business analysis support |
| **2** | Phạm Huy Hoàng | Team Leader, BA, DEV, Tester | * Managing process * Designing database * Clarifying requirements * Prepare documents * GUI Design * Create test plan * Coding * Testing |
| **3** | Nguyễn Thanh Bình | Team Member, BA, DEV, Tester | * Designing database * Clarifying requirements * Prepare documents * GUI Design * Create test plan * Coding * Testing |
| **4** | Nguyễn Quốc Huy | Team Member, BA, DEV, Tester | * Designing database * Clarifying requirements * Prepare documents * GUI Design * Create test plan * Coding * Testing |
| **5** | Đỗ Minh Đạt | Team Member, BA, DEV, Tester | * Designing database * Clarifying requirements * Prepare documents * GUI Design * Create test plan * Coding * Testing |

Table 4: Roles and Responsibility Details

### Tools and Techniques

- Front-end technologies: HTML5, CSS3, JavaScript, jQuery, AJAX.

- Back-end: Website: ASP.NET MVC3 + Entity Framework.

Web Service: WCF. Mobile App: Android - Java.

- Web Server: Microsoft IIS.

- Database Management System: MS SQL Server 2008 Express

## Project Management Plan

### Iteration

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Phase**  **/Iteration** | **Description** | **Deliverables** | **Resource needed** | **Dependencies and Constrains** | **Risks** |
| **Preliminary Investigation or Analysis** | - Study similar existing systems.  -Identify and clarify requirements for the system in general. | -Introduction of proposed system.  -Main functions.  -Project Iteration Plan. | 30 man-days | N/A | Project may  not be feasible  for developing  because lack of technologies  and/or data |
| **Face Detect & Recognize** | -Studying face detect, face recognize algorithm & library.  -Find a solution, optimize for higher performance and accuracy. | -Face Detect and Recognize System (On Web Site and Web Service) | 25 man-days | N/A | Lack of experience.  The implemented algorithm is not the best.  Lack of test data |
| **Student management** | -Manage subject, student images/information. | -Student management function | 20 man-days | Depend on “**Face Detect & Recognize”** |  |
| **Roll call management** | -Create the roll call list, based on the information of student, instructor, class and course . | -Roll call management function | 30 man-days | N/A | Lack of experience.  Not have a clear understanding about business process. |
| **Web Service Implement** | -Create and deploy the web service on server. Provide RestFul service. | -Running WCF Web Service | 25 man-days | Depend on “**Roll call management”** | Lack of experience on making and deploying web service. |
| **Attendance checking** | -Instructor use mobile app to take picture, use picture for attendance checking. | -Android App with attendance checking function | 20 man-days | Depend on “**Web Service Implement”** | Lack of experience on Android development, transfer file between Android and WCF.  No Android device available for testing. |
| **Attendance report** | -Students can view their own attendance rate.  -Instructor and admin can view reports. | -Attendance report function. | 15 man-days | N/A | Lack of experience of making report. |

Table 5: Iteration

### Iteration Detail

#### Phase 1: Preliminary Investigation or Analysis

|  |  |  |
| --- | --- | --- |
| **Task** | **Description** | **Author** |
| **1. Identifying and studying existing systems** | Find which systems currently provide similar service, their strengths and weakness. | HuyNQ, HoangPH, BinhNT |
| **2. Identifying and clarifying main functions.** | Define which main functions system should provide. | HuyNQ, HoangPH, BinhNT |
| **3. Introduction.** | Complete Introduction Report. | HoangPH |
| **4. Project Management**  **Plan.** | Prepare Project  Management Plan. | HoangPH |
| **5. Website Prototype.** | Build a prototype of proposed system (Website). | HuyNQ, HoangPH |
| **6. Mobile Prototype.** | Build a prototype of proposed system (Mobile App). | BinhNT, HoangPH |
| **7. Design ER diagram.** | Design ER diagram. | HoangPH, HuyNQ, BinhNT |

Table 6: Phase 1: Preliminary Investigation or Analysis

#### Phase 2: Face Detect & Recognize

|  |  |  |
| --- | --- | --- |
| **Task** | **Description** | **Author** |
| **1. Identifying Requirement and Planning** | Which feature this function  should have and how to  implement. | HoangPH |
| **2. Studying Face Detection & Recognition Algorithm** | Studying algorithm, implement by using library EmguCV. | HoangPH |
| **3. Extract Face from Image** | Find the faces in images, extract them for later use | HoangPH |
| **4. Recognize Face** | From the input face, find out who the face belong to | HoangPH |
| **5. Optimize** | Optimize the implement for more performance and accuracy | HoangPH |
| **6. Implement GUI** | Create the interface for extracting and storing face | HoangPH |
| **7. Testing** | Test system behavior and  performance  Test user behavior and  performance | HoangPH, HuyNQ, BinhNT, DatDM |
| **8. Document** | Adding SRS, SDD,  Installation Guide, Manual  Guide | HoangPH, HuyNQ, BinhNT, DatDM |

Table 7: Phase 2: Face Detect & Recognize

#### Phase 3: Student Management

|  |  |  |
| --- | --- | --- |
| **Task** | **Description** | **Author** |
| **1. Identifying Requirement and Planning** | Which feature this function  should have and how to  implement. | HoangPH, HuyNQ, BinhNT, DatDM |
| **2. Design ER Diagram** | Design ER Diagram | HoangPH, HuyNQ, BinhNT, DatDM |
| **3. Manage Student** | Allow admin to add/edit/active/inactive student | HuyNQ |
| **4. Manage Student Face** | Each student has a training set, which contains 8-20 faces. Allow admin to add, edit images in this training set | HuyNQ, HoangPH |
| **5. Implement GUI** | Create the interface for managing student info | HuyNQ, HoangNQ |
| **6. Testing** | Test system behavior and  performance  Test user behavior and  performance | HoangPH, HuyNQ, BinhNT, DatDM |
| **7. Document** | Adding SRS, SDD,  Installation Guide, Manual  Guide | HoangPH, HuyNQ, BinhNT, DatDM |

Table 8: Phase 3: Student Management

#### Phase 4: Roll Call Management

|  |  |  |
| --- | --- | --- |
| **Task** | **Description** | **Author** |
| **1. Identifying Requirement and Planning** | Which feature this function  should have and how to  implement. | HoangPH, HuyNQ, BinhNT, DatDM |
| **2. Manage Instructor** | Allow admin to add/edit/active/inactive instructor | DatDM |
| **3. Manage Course** | Allow admin to add/edit/active/inactive course | DatDM |
| **4. Manage Class** | Allow admin to add/edit/active/inactive course.  Assign student to class | HuyNQ |
| **5. Manage Roll Call** | Allow admin to create/edit/delete roll call.  Each roll call contain info: The instructor, the student list, the course, time, begin – end date | HuyNQ, HoangPH |
| **6. User Profile** | Instructor or student accounts will be created by system.  Instructor or student can log in to change their profile, password | HuyNQ |
| **7. Implement GUI** | Create the interface for managing roll call | HuyNQ, BinhNT, HoangPH |
| **8. Testing** | Test system behavior and  performance  Test user behavior and  performance | HoangPH, HuyNQ, BinhNT, DatDM |
| **9. Document** | Adding SRS, SDD,  Installation Guide, Manual  Guide | HoangPH, HuyNQ, BinhNT, DatDM |

Table 9: Phase 4: Roll Call Management

#### Phase 5: Web Service Implement

|  |  |  |
| --- | --- | --- |
| **Task** | **Description** | **Author** |
| **1. Identifying Requirement and Planning** | Which feature this function  should have and how to  implement. | HoangPH, HuyNQ, BinhNT, DatDM |
| **2. Studying WCF** | Studying the create and using of WCF | HuyNQ |
| **3. Instructor Login** | Check the input id and password from mobile to login instructor | HuyNQ |
| **4. Get Instructor Info, Roll Call List** | Based on the instructor’s id, show to current roll call list | HuyNQ |
| **5. Face Recognize From Android Camera** | Studying about transferring image files between Android and WCF  Transfer the result back to Mobile for showing | HuyNQ, BinhNT |
| **6. Testing** | Test system behavior and  performance  Test user behavior and  performance | HoangPH, HuyNQ, BinhNT, DatDM |
| **7. Document** | Adding SRS, SDD,  Installation Guide, Manual  Guide | HoangPH, HuyNQ, BinhNT, DatDM |

Table 10: Phase 5: Web Service Implement

#### Phase 6: Attendance Checking

|  |  |  |
| --- | --- | --- |
| **Task** | **Description** | **Author** |
| **1. Identifying Requirement and Planning** | Which feature this function  should have and how to  implement. | HoangPH, HuyNQ, BinhNT, DatDM |
| **2. Getting Image From Mobile, sending to Web Service** | Allow instructor to take picture of class, use picture for checking attendance | BinhNT |
| **3. Confirm result, re-check attendance manually** | Allow instructor to confirm and re-check attendance | BinhNT |
| **4. Stranger Alert** | Alert when detect stranger | BinhNT |
| **5. Implement Mobile App** | Create the Android App for instructor to User | BinhNT, HuyNQ, HoangNQ |
| **6. Testing** | Test system behavior and  performance  Test user behavior and  performance | HoangPH, HuyNQ, BinhNT, DatDM |
| **7. Document** | Adding SRS, SDD,  Installation Guide, Manual  Guide | HoangPH, HuyNQ, BinhNT, DatDM |

Table 11: Phase 6: Attendance Checking

#### Phase 7: Attendance Report

|  |  |  |
| --- | --- | --- |
| **Task** | **Description** | **Author** |
| **1. Identifying Requirement and Planning.** | Which feature this function  should have and how to  implement. | HoangPH, HuyNQ, BinhNT, DatDM |
| **2. Report about absent rate of a course** | System will make report about the absent rate of a course by month/year | HoangPH |
| **3. Report about absent rate of a student** | Allow instructor to confirm and re-check attendance | HoangPH |
| **4. Export Report** | Admin can export report | HuyNQ |
| **6. Testing** | Test system behavior and  performance  Test user behavior and  performance | HoangPH, HuyNQ, BinhNT, DatDM |
| **7. Document** | Adding SRS, SDD,  Installation Guide, Manual  Guide | HoangPH, HuyNQ, BinhNT, DatDM |

Table 12: Phase 7: Attendance Report

### All Meeting Minutes

Refer to Meeting Minutes folder.

## Coding Convention

Java: Using to develop Android App.

Summary:

* Naming Convention.
* Indentation.
* Declaration.
* Code Examples

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

C#: Using to develop website and web service.

Summary:

* Naming Convention.
* Layout Convention.
* Commenting Convention.
* Language Guidelines

Using C# Code Convention From:

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

# Software Requirement Specification

## User Requirement Specification

### Guest Requirement

### Instructor Requirement

### Staff Requirement

### Student Requirement

### Admin Requirement

### System Requirement

## System Requirement Specification

### External Interface Requirement

#### User Interface

#### Hardware Interface

#### Software Interface

#### Communication Protocol

### System Overview Use Case



Figure System Overview Use Case

### List of Use Case

#### <Guest>Overview Use Case



Figure : <Guest> Overview Use Case

##### <Guest> Login

Use Case Diagram



Figure : <Guest> Login

Use Case Specification

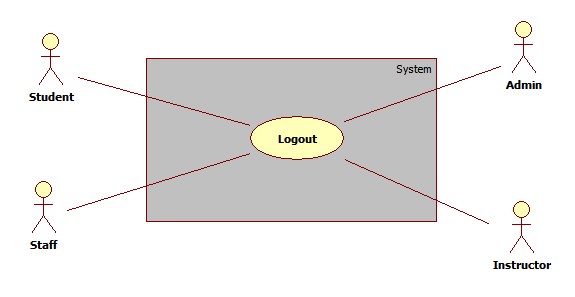
|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE – GU001** | | | |
| **Use Case No.** | GU001 | **Use Case Version** | 1.0 |
| **Use Case Name** | Login | | |
| **Author** | Pham Huy Hoang | | |
| **Date** | 19/09/2013 | **Priority** | Normal |
| **Actor:**Guest.  **Summary:**  Customer uses this case to view shopping cart.  **Goal:**  Allow customer views list of product on their shopping cart.  **Triggers:**  - Customer can view all products that they added to shopping cart. They also can  update quantity of product, delete product or choose product’s size.  - Customer clicks on icon shopping cart on header of page to view shopping cart.  **Preconditions:** N/A.  **Post Conditions:**  - Success: System will transfer to view cart page and show all products in shopping  cart.  **Fail:** System will transfer to error page or still in current page.  **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | |  |  |  | |  |  |  |   **Alternative Scenario:** N/A  **Exceptions:** N/A  **Relationships:** Manage Order  **Business Rules:** | | | |

Table : <Guest> Login

#### <Instructor>Overview Use Case



##### <Instructor> Logout

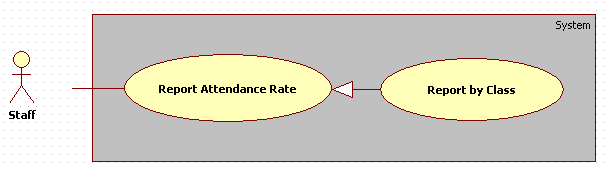


#### <Staff >Overview Use Case



##### <Staff> Report by Class

Use Case Diagram

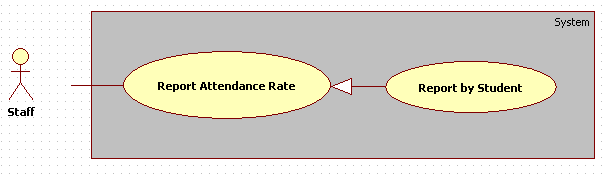


Use Case Specification

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE – SU001** | | | |
| **Use Case No.** | SU001 | **Use Case Version** | 1.0 |
| **Use Case Name** | Report by Class. | | |
| **Author** | Nguyen Quoc Huy. | | |
| **Date** | 19/09/2013 | **Priority** | Normal |
| **Actor:** Staff.  **Summary:**  User uses this case to report attendance rate by class.  **Goal:**  Allow user views rate of attendance by class.  **Triggers:**   * User can view all students with rate attendance in class. * They also can export to excel file by click button “Export”.   **Preconditions:**   * User must logged in the system with the role is staff.   **Post Conditions:**   * **Success**: System will transfer to view report page and show all students with attendance rate. * **Fail:** System will transfer to error page or still in current page.   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | Click “Report” in menu bar. | Redirect to Report Page, include 3 part:   * Select box report. * Data grid view with data default. * Button “Export”. | | 2 | Type or choose class. | Filter dataset of data grid view by class. |   **Alternative Scenario:** N/A  **Exceptions:** N/A  **Relationships:** Manage Report Attendance Rate.  **Business Rules:** | | | |

##### <Staff> Report by Student

Use Case Diagram

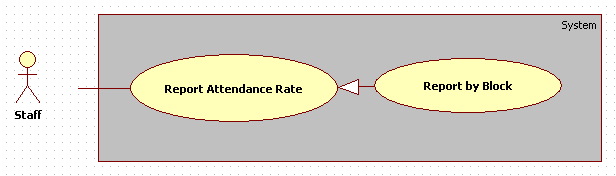


Use Case Specification

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE – SU001** | | | |
| **Use Case No.** | SU002 | **Use Case Version** | 1.0 |
| **Use Case Name** | Report by Student. | | |
| **Author** | Nguyen Quoc Huy. | | |
| **Date** | 19/09/2013 | **Priority** | Normal |
| **Actor:** Staff.  **Summary:**  User uses this case to report attendance rate by student.  **Goal:**  Allow user views rate of attendance by student.  **Triggers:**   * User can view student with rate attendance. * They also can export to excel file by click button “Export”.   **Preconditions:**   * User must logged in the system with the role is staff.   **Post Conditions:**   * **Success**: System will transfer to view report page and show student with attendance rate. * **Fail:** System will transfer to error page or still in current page.   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | Click “Report” in menu bar. | Redirect to Report Page, include 3 part:   * Select box report type. * Data grid view with data default. * Button “Export”. | | 2 | Type or choose the student. | Filter dataset of data grid view. |   **Alternative Scenario:** N/A  **Exceptions:** N/A  **Relationships:** Manage Report Attendance Rate.  **Business Rules:** | | | |

##### <Staff> Report by Block

Use Case Diagram

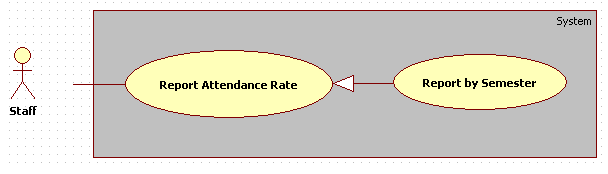


Use Case Specification

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE – SU003** | | | |
| **Use Case No.** | SU003 | **Use Case Version** | 1.0 |
| **Use Case Name** | Report by Block. | | |
| **Author** | Nguyen Quoc Huy. | | |
| **Date** | 19/09/2013 | **Priority** | Normal |
| **Actor:** Staff.  **Summary:**  User uses this case to report attendance rate by Block.  **Goal:**  Allow user views rate of attendance by Block.  **Triggers:**   * User can view all students with rate attendance. * They also can export to excel file by click button “Export”.   **Preconditions:**   * User must logged in the system with the role is staff.   **Post Conditions:**   * **Success**: System will transfer to view report page and show all students with attendance rate. * **Fail:** System will transfer to error page or still in current page.   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | Click “Report” in menu bar. | Redirect to Report Page, include 3 part:   * Select box report type. * Data grid view with data default. * Button “Export”. | | 2 | Type or choose the block. | Filter dataset of data grid view. |   **Alternative Scenario:** N/A  **Exceptions:** N/A  **Relationships:** Manage Report Attendance Rate.  **Business Rules:** | | | |

##### <Staff> Report by Semester

Use Case Diagram

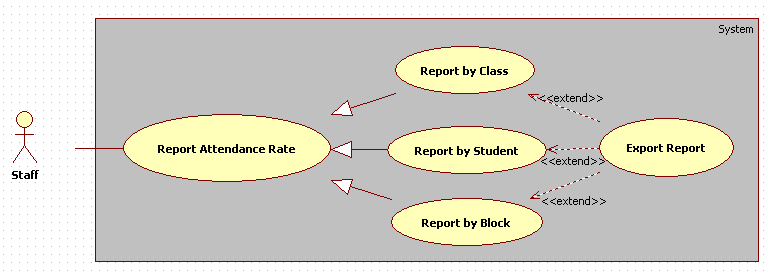


Use Case Specification

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE – SU004** | | | |
| **Use Case No.** | SU004 | **Use Case Version** | 1.0 |
| **Use Case Name** | Report by Semester. | | |
| **Author** | Nguyen Quoc Huy. | | |
| **Date** | 19/09/2013 | **Priority** | Normal |
| **Actor:** Staff.  **Summary:**  User uses this case to report attendance rate by Semester.  **Goal:**  Allow user views rate of attendance by Semester.  **Triggers:**   * User can view all students with rate attendance. * They also can export to excel file by click button “Export”.   **Preconditions:**   * User must logged in the system with the role is staff.   **Post Conditions:**   * **Success**: System will transfer to view report page and show all students with attendance rate. * **Fail:** System will transfer to error page or still in current page.   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | Click “Report” in menu bar. | Redirect to Report Page, include 3 part:   * Select box report type. * Data grid view with data default. * Button “Export”. | | 2 | Type or choose the Semester. | Filter dataset of data grid view. |   **Alternative Scenario:** N/A  **Exceptions:** N/A  **Relationships:** Manage Report Attendance Rate.  **Business Rules:** | | | |

##### <Staff> Export Report

Use Case Diagram

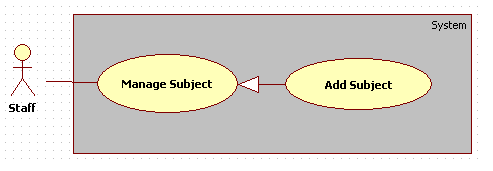


Use Case Specification

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE – SU005** | | | |
| **Use Case No.** | SU005 | **Use Case Version** | 1.0 |
| **Use Case Name** | Export Report. | | |
| **Author** | Nguyen Quoc Huy. | | |
| **Date** | 19/09/2013 | **Priority** | Normal |
| **Actor:** Staff.  **Summary:**  User uses this case to report attendance rate by student.  **Goal:**  Allow user views rate of attendance by student.  **Triggers:**   * Export report student list with attendance rate (by class/block or semester) to excel file.   **Preconditions:**   * User must logged in the system with the role is staff.   **Post Conditions:**   * **Success**: Export a excel file with student list.. * **Fail:** System will transfer to error page or still in current page.   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | Click “Report” in menu bar. | Redirect to Report Page, include 3 part:   * Select box report type. * Data grid view with data default. * Button “Export”. | | 2 | Type or choose class/block/student or semester. | Filter dataset of data grid view. | | 3 | Click button “Export” | Create a excel file with student list. |   **Alternative Scenario:** N/A  **Exceptions:** N/A  **Relationships:** Manage Report Attendance Rate.  **Business Rules:** | | | |

##### <Staff> Add Subject

Use Case Diagram

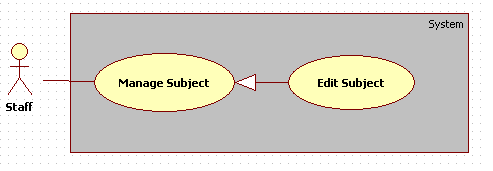


Use Case Specification

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE – SU006** | | | |
| **Use Case No.** | SU006 | **Use Case Version** | 1.0 |
| **Use Case Name** | Add Subject. | | |
| **Author** | Nguyen Quoc Huy. | | |
| **Date** | 19/09/2013 | **Priority** | Normal |
| **Actor:** Staff.  **Summary:**  Create a new Subject.  **Goal:**  A new Subject include: specialized, name, block will be added to database.  **Triggers:**   * User can create a new subject with information.   **Preconditions:**   * User must logged in the system with the role is staff.   **Post Conditions:**   * **Success**: A new Subject will be created. The Subject will be displayed on Subject list. * **Fail:** System will transfer to error page or still in current page.   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | Click “Manage Subject” in menu bar. | Redirect to Manage Subject Page, include 2 tabs:   * Subject List * New Subject | | 2 | Click “New Subject” tab. | Display “New Subject” tab with :   * Specialized: textbox (max length: 30), required. * Name: textbox (max length: 30), required. * Slot (/day): drop down list (1,2). | | 3 | Input Subject information, click “Add” button. | Create new Subject to database. Redirect Subject list tab. |   **Alternative Scenario:** N/A  **Exceptions:** N/A  **Relationships:** Manage Subject.  **Business Rules:** | | | |

##### <Staff> Edit Subject

Use Case Diagram

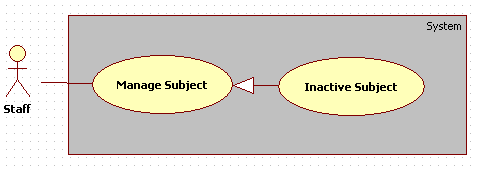


Use Case Specification

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE – SU007** | | | |
| **Use Case No.** | SU007 | **Use Case Version** | 1.0 |
| **Use Case Name** | Edit Subject. | | |
| **Author** | Nguyen Quoc Huy. | | |
| **Date** | 19/09/2013 | **Priority** | Normal |
| **Actor:** Staff.  **Summary:**  Edit a Subject.  **Goal:**  Update Subject includes: specialized, name, block will be added to database.  **Triggers:**   * User can update new information for subject.   **Preconditions:**   * User must logged in the system with the role is staff.   **Post Conditions:**   * **Success**: A Subject will be updated. The Subject will be displayed on Subject list. * **Fail:** System will transfer to error page or still in current page.   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | Click “Manage Subject” in menu bar. | Redirect to Manage Subject Page, include 2 tabs:   * Subject List * New Subject | | 2 | In “Subject List” , click button “Edit” | Display “Updated Subject” popup with :   * Specialized: textbox (max length: 30), required. * Name: textbox (max length: 30), required. * Slot (/day): drop down list (1,2). | | 3 | Edit Subject information, click “Save” button. | Update Subject to database. Show Subject list . |   **Alternative Scenario:** N/A  **Exceptions:** N/A  **Relationships:** Manage Subject.  **Business Rules:** | | | |

##### <Staff> Inactive Subject

Use Case Diagram

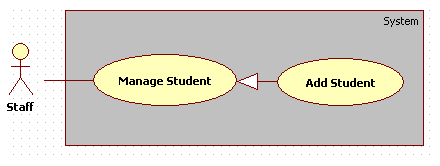


Use Case Specification

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE – SU008** | | | |
| **Use Case No.** | SU008 | **Use Case Version** | 1.0 |
| **Use Case Name** | Inactive subject. | | |
| **Author** | Nguyen Quoc Huy. | | |
| **Date** | 19/09/2013 | **Priority** | Normal |
| **Actor:** Staff.  **Summary:**  Edit a Subject.  **Goal:**  Update Subject includes: specialized, name, block will be added to database.  **Triggers:**   * User can set active/ inactive for subject.   **Preconditions:**   * User must logged in the system with the role is staff.   **Post Conditions:**   * **Success**: The status Subject will be changed to “inactive”. * **Fail:** System will transfer to error page or still in current page.   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | Click “Manage Subject” in menu bar. | Redirect to Manage Subject Page, include 2 tabs:   * Subject List * New Subject | | 2 | In “Subject List” , click button “Active” | Show a message box: “Do you want to inactive this subject?”   * Yes button * No button | | 3 | Click “Yes” button | Change status “Active” to “Inactive” |   **Alternative Scenario:** N/A  **Exceptions:** N/A  **Relationships:** Manage Subject.  **Business Rules:** | | | |

##### <Staff>Add Student

Use Case Diagram

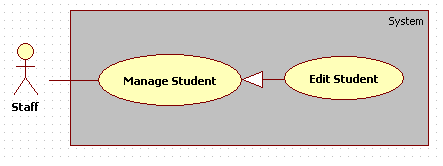


Use Case Specification

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE – SU009** | | | |
| **Use Case No.** | SU009 | **Use Case Version** | 1.0 |
| **Use Case Name** | Add Student. | | |
| **Author** | Nguyen Quoc Huy. | | |
| **Date** | 19/09/2013 | **Priority** | Normal |
| **Actor:** Staff.  **Summary:**  Create a new Student.  **Goal:**  A new Student include: Course, class, full name, birthday, Citizen card id, Email, address and parents information (name, job, address and phone number).  **Triggers:**   * User can add new student manual or import by excel.   **Preconditions:**   * User must logged in the system with the role is staff.   **Post Conditions:**   * **Success**: A new student will be created. The student will be displayed on student list. * **Fail:** System will transfer to error page or still in current page.   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | Click “Manage Student” in menu bar. | Redirect to Manage Student Page, include 3 tabs:   * Subject List * New Subject * Searched Student | | 2 | Click “New Student” tab. | Display “New Student” tab with :   * Course and class: drop down list, required. * Full name: textbox (max length: 30), required. * Birthday: textbox (max length: 30), required. * Citizen card id: textbox (max length: 30), required. * Email: textbox (max length: 30). * Address: textbox (max length: 30), required. * Father’s Name: textbox (max length: 30). * Father’s job: textbox (max length: 30). * Father’s Address: textbox (max length: 30). * Father’s Phone number: textbox (max length: 30). * Mother’s Name: textbox (max length: 30). * Mother’s job: textbox (max length: 30). * Mother’s Address: textbox (max length: 30). * Mother’s Phone number: textbox (max length: 30). | | 3 | Input student information, click “Add” button. | Create new student to database. Redirect student list tab. |   **Alternative Scenario:** N/A  **Exceptions:** N/A  **Relationships:** Manage Student.  **Business Rules:** | | | |

##### <Staff>Edit Student

Use Case Diagram

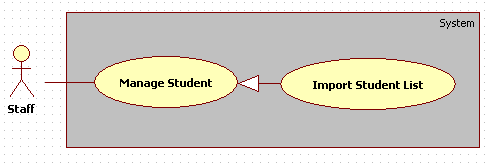


Use Case Specification

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE – SU010** | | | |
| **Use Case No.** | SU010 | **Use Case Version** | 1.0 |
| **Use Case Name** | Edit student. | | |
| **Author** | Nguyen Quoc Huy. | | |
| **Date** | 19/09/2013 | **Priority** | Normal |
| **Actor:** Staff.  **Summary:**  Edit a student.  **Goal:**  Update student includes: Course, class, full name, birthday, Citizen card id, Email, address and parents information (name, job, address and phone number) will be updated to database.  **Triggers:**   * User can update new information for student.   **Preconditions:**   * User must logged in the system with the role is staff.   **Post Conditions:**   * **Success**: A student will be updated. The student will be displayed on Subject list. * **Fail:** System will transfer to error page or still in current page.   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | Click “Manage Student” in menu bar. | Redirect to Student manage Page, include 3 tabs:   * Student List * New Student * Searched Student | | 2 | In “Student List” , click button “Detail” | Display “Updated student” popup with :   * Course and class: drop down list, required. * Full name: textbox (max length: 30), required. * Birthday: textbox (max length: 30), required. * Citizen card id: textbox (max length: 30), required. * Email: textbox (max length: 30). * Address: textbox (max length: 30), required. * Father’s Name: textbox (max length: 30). * Father’s job: textbox (max length: 30). * Father’s Address: textbox (max length: 30). * Father’s Phone number: textbox (max length: 30). * Mother’s Name: textbox (max length: 30). * Mother’s job: textbox (max length: 30). * Mother’s Address: textbox (max length: 30). * Mother’s Phone number: textbox (max length: 30). (1,2). | | 3 | Edit student information, click “Save” button. | Update student to database. Show student list. |   **Alternative Scenario:** N/A  **Exceptions:** N/A  **Relationships:** Manage Student.  **Business Rules:** | | | |

##### <Staff>Import Student List

Use Case Diagram

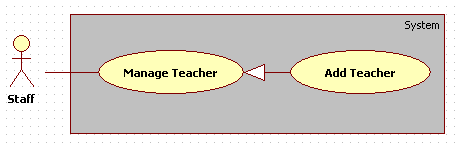


Use Case Specification

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE – SU011** | | | |
| **Use Case No.** | SU011 | **Use Case Version** | 1.0 |
| **Use Case Name** | Import student list | | |
| **Author** | Nguyen Quoc Huy. | | |
| **Date** | 19/09/2013 | **Priority** | Normal |
| **Actor:** Staff.  **Summary:**  Import student list.  **Goal:**  Create multi student by export from excel file.  **Triggers:**   * User can import an excel file student list.   **Preconditions:**   * User must logged in the system with the role is staff.   **Post Conditions:**   * **Success**: Student list will be added to database. * **Fail:** System will transfer to error page or still in current page.   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | Click “Manage Student” in menu bar. | Redirect to Manage student Page, include 3 tabs:   * Student List * New Student * Searched Student | | 2 | Click “New Student” tab | Redirect to New student tab include:   * Textbox (max length: 30), required. * Browser button. * Import button | | 3 | Input path or choose from browser. Click “Import” button. | Import student list to database, redirect to Student list tab. |   **Alternative Scenario:** N/A  **Exceptions:** N/A  **Relationships:** Manage Student.  **Business Rules:** | | | |

##### <Staff>Add Teacher

Use Case Diagram

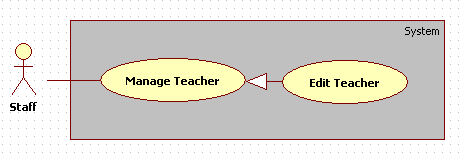


Use Case Specification

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE – SU012** | | | |
| **Use Case No.** | SU012 | **Use Case Version** | 1.0 |
| **Use Case Name** | Add Teacher. | | |
| **Author** | Nguyen Quoc Huy. | | |
| **Date** | 19/09/2013 | **Priority** | Normal |
| **Actor:** Staff.  **Summary:**  Create a new teacher.  **Goal:**  A new teacher include: Name …  **Triggers:**   * User can add new teacher.   **Preconditions:**   * User must logged in the system with the role is staff.   **Post Conditions:**   * **Success**: A new teacher will be created. The teacher will be displayed on teacher list. * **Fail:** System will transfer to error page or still in current page.   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | Click “Manage Instructor” in menu bar. | Redirect to Manage Instructor Page, include 2 tabs:   * Instructor List * New Instructor | | 2 | Click “New Instructor” tab. | Display “New Instructor” tab with :   * Name: textbox (max length: 30), required. | | 3 | Input Subject information, click “Create” button. | Create new instructor to database. Redirect instructor list tab. |   **Alternative Scenario:** N/A  **Exceptions:** N/A  **Relationships:** Manage Teacher.  **Business Rules:** | | | |

##### <Staff>Edit Teacher

Use Case Diagram

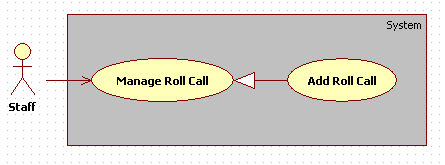


Use Case Specification

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE – SU013** | | | |
| **Use Case No.** | SU013 | **Use Case Version** | 1.0 |
| **Use Case Name** | Edit teacher. | | |
| **Author** | Nguyen Quoc Huy. | | |
| **Date** | 19/09/2013 | **Priority** | Normal |
| **Actor:** Staff.  **Summary:**  Edit a teacher.  **Goal:**  Update Teacher includes: name will be update to database.  **Triggers:**   * User can update new information for teacher.   **Preconditions:**   * User must logged in the system with the role is staff.   **Post Conditions:**   * **Success**: A teacher will be updated. The teacher will be displayed on teacher list. * **Fail:** System will transfer to error page or still in current page.   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | Click “Manage Instructor” in menu bar. | Redirect to Manage Instructor Page, include 2 tabs:   * Instructor List * New Instructor. | | 2 | In “Instructor List” , click button “Edit” | Display “Updated Subject” popup with :   * Name: textbox (max length: 30), required. | | 3 | Edit Subject information, click “Save” button. | Update teacher to database. Show teacher list. |   **Alternative Scenario:** N/A  **Exceptions:** N/A  **Relationships:** Manage teacher.  **Business Rules:** | | | |

##### <Staff>Add Roll Call

Use Case Diagram

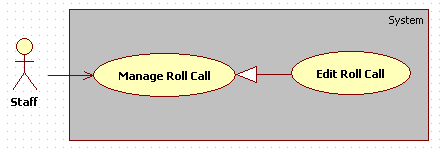


Use Case Specification

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE – SU014** | | | |
| **Use Case No.** | SU014 | **Use Case Version** | 1.0 |
| **Use Case Name** | Add Roll Call. | | |
| **Author** | Nguyen Quoc Huy. | | |
| **Date** | 19/09/2013 | **Priority** | Normal |
| **Actor:** Staff.  **Summary:**  Create a new Roll Call.  **Goal:**  A new Roll Call include: subject, instructor, student list, class, date, time will be added to database.  **Triggers:**   * User can create new roll call.   **Preconditions:**   * User must logged in the system with the role is staff.   **Post Conditions:**   * **Success**: A new Roll Call will be created. The Roll Call will be displayed on Roll Call list. * **Fail:** System will transfer to error page or still in current page.   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | Click “Manage Roll Call” in menu bar. | Redirect to Roll Call Page, include 2 tabs:   * Roll Call List * New Roll Call | | 2 | Click “New Roll Call” tab. | Display “New Roll Call” tab with :   * Instructor: drop down list, required. * Subject: drop down list, required. * Day: drop down list, required. * Class: drop down list, required. * Date: date picker, required. * Time: drop down list, required. * “Add” button * “Cancel” button | | 3 | Input Roll Call information, click “Add” button. | Create new Roll Call to database. Redirect Roll Call list tab. |   **Alternative Scenario:** N/A  **Exceptions:** N/A  **Relationships:** Manage Roll Call.  **Business Rules:** | | | |

##### <Staff>Edit Roll Call

Use Case Diagram

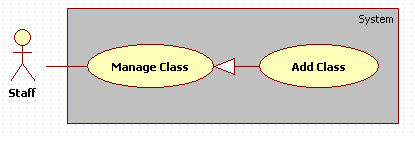


Use Case Specification

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE – SU015** | | | |
| **Use Case No.** | SU015 | **Use Case Version** | 1.0 |
| **Use Case Name** | Edit Roll Call. | | |
| **Author** | Nguyen Quoc Huy. | | |
| **Date** | 19/09/2013 | **Priority** | Normal |
| **Actor:** Staff.  **Summary:**  Edit a Roll Call.  **Goal:**  Update roll call includes: subject, instructor, class, student list, date, time will be update to database.  **Triggers:**   * User can update information for roll call.   **Preconditions:**   * User must logged in the system with the role is staff.   **Post Conditions:**   * **Success**: A roll call will be updated. The roll call will be displayed on roll call list. * **Fail:** System will transfer to error page or still in current page.   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | Click “Manage Roll Call” in menu bar. | Redirect to Manage Roll Call Page, include 2 tabs:   * Roll Call List * New Roll Call. | | 2 | In “Roll Call List” , click button “Edit” | Display “Updated Roll Call” popup with :   * Instructor: drop down list, required. * Subject: drop down list, required. * Day: drop down list, required. * Class: drop down list, required. * Date: date picker, required. * Time: drop down list, required. * “Update” button * “Cancel” button | | 3 | Edit roll call information, click “Save” button. | Update roll call to database. Show roll call list. |   **Alternative Scenario:** N/A  **Exceptions:** N/A  **Relationships:** Manage roll call.  **Business Rules:** | | | |

##### <Staff>Add Class

Use Case Diagram

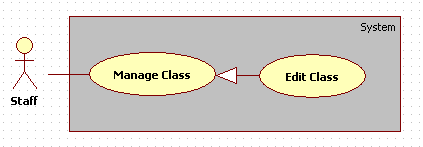


Use Case Specification

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE – SU016** | | | |
| **Use Case No.** | SU016 | **Use Case Version** | 1.0 |
| **Use Case Name** | Add Class. | | |
| **Author** | Nguyen Quoc Huy. | | |
| **Date** | 19/09/2013 | **Priority** | Normal |
| **Actor:** Staff.  **Summary:**  Create a new class.  **Goal:**  A new Class include: specialized, name will be added to database.  **Triggers:**   * User can create new class.   **Preconditions:**   * User must logged in the system with the role is staff.   **Post Conditions:**   * **Success**: A new class will be created. The class will be displayed on class list. * **Fail:** System will transfer to error page or still in current page.   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | Click “Manage class” in menu bar. | Redirect to Manage class Page, include 2 tabs:   * Class List * New Class | | 2 | Click “New Class” tab. | Display “New Subject” tab with :   * Specialized: Drop down list, required. * Name: textbox (max length: 30), required. | | 3 | Input Subject information, click “Add” button. | Create new class to database. Redirect class list tab. |   **Alternative Scenario:** N/A  **Exceptions:** N/A  **Relationships:** Manage Class.  **Business Rules:** | | | |

##### <Staff>Edit Class

Use Case Diagram



Use Case Specification

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE – SU017** | | | |
| **Use Case No.** | SU017 | **Use Case Version** | 1.0 |
| **Use Case Name** | Edit Class. | | |
| **Author** | Nguyen Quoc Huy. | | |
| **Date** | 19/09/2013 | **Priority** | Normal |
| **Actor:** Staff.  **Summary:**  Edit a Class.  **Goal:**  Update class includes: specialized, name will be update to database.  **Triggers:**   * User can update information for class.   **Preconditions:**   * User must logged in the system with the role is staff.   **Post Conditions:**   * **Success**: A class will be updated. The class will be displayed on class list. * **Fail:** System will transfer to error page or still in current page.   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | Click “Manage Class” in menu bar. | Redirect to Manage Class Page, include 2 tabs:   * Class List * New Class. | | 2 | In “Class List” , click button “Edit” | * Display “Updated Class” popup with : Specialized: Drop down list, required. * Name: textbox (max length: 30), required. | | 3 | Edit class information, click “Save” button. | Update class to database. Show class list. |   **Alternative Scenario:** N/A  **Exceptions:** N/A  **Relationships:** Manage Class.  **Business Rules:** | | | |

#### <Student>Overview Use Case



#### <Admin>Overview Use Case



#### <System>Overview Use Case

