** MINISTRY OF EDUCATION AND TRAINING**

**FPT UNIVERSITY**

Capstone Project Document

Smart Buy

|  |  |
| --- | --- |
| **Group 1** | |
| **Group member** | Doan Ho Anh Triet – Team Leader – SE60763  Huynh Thanh Viet – Team Member - SE60666  Dang Huu Hoang – Team Member - 60486  Tran Trung Dung – Team Member - 60236 |
| **Supervisor** | Mr. Kieu Trong Khanh |
| **Ext. Supervisor** | N/A |
| **Capstone Project code** | SmartB |

-Ho Chi Minh City, 02/2014-

*This page is intentionally left blank*

***ACKNOWLEDGEMENTS***

We wish to thank various people for their contribution to this project: Our teachers for their advice and participation in the final review, our friends for their 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](#_Toc368775433)

[List of Tables 5](#_Toc368775434)

[List of Figures 7](#_Toc368775435)

[Definitions, Acronyms, and Abbreviations 9](#_Toc368775436)

[Report No.3 Software Requirement Specification 10](#_Toc368775437)

[1. User Requirement Specification 10](#_Toc368775438)

[1.1 Guest Requirement 10](#_Toc368775439)

[1.2 Instructor Requirement 10](#_Toc368775440)

[1.3 Staff Requirement 10](#_Toc368775441)

[1.4 Student Requirement 10](#_Toc368775442)

[1.5 Admin Requirement 10](#_Toc368775443)

[1.6 System Requirement 10](#_Toc368775444)

[2. System Requirement Specification 10](#_Toc368775445)

[2.1 External Interface Requirement 10](#_Toc368775446)

[2.2 System Overview Use Case 11](#_Toc368775447)

[2.3 List of Use Case 12](#_Toc368775448)

[3. Software System Attribute 83](#_Toc368775449)

[3.1 Usability 83](#_Toc368775450)

[3.2 Reliability 83](#_Toc368775451)

[3.3 Availability 83](#_Toc368775452)

[3.4 Security 83](#_Toc368775453)

[3.5 Maintainability 83](#_Toc368775454)

[3.6 Portability 83](#_Toc368775455)

[3.7 Performance 83](#_Toc368775456)

[4. ERD 84](#_Toc368775457)

# List of Tables

[Table 1: <Admin> Add Image for Single Student 13](#_Toc368775542)

[Table 2: <Admin> Add Image for Multi Students 16](#_Toc368775543)

[Table 3: <Admin> Remove Student Images 18](#_Toc368775544)

[Table 4: <Admin> Create Account 20](#_Toc368775545)

[Table 5: <Admin> Active Account 21](#_Toc368775546)

[Table 6: <Admin> Inactive Account 23](#_Toc368775547)

[Table 7: <Admin> Config System 24](#_Toc368775548)

[Table 8: <System> Face Detection 26](#_Toc368775549)

[Table 9: <System> Recognize Student 27](#_Toc368775550)

[Table 10: <System> Auto Free Storage Space 28](#_Toc368775551)

[Table 11: <System> Auto Active Roll Call 29](#_Toc368775552)

[Table 12: <System> Auto Inactive Roll Call 31](#_Toc368775553)

[Table 13: <Staff>Add Roll Call 33](#_Toc368775554)

[Table 14: <Staff>Edit Roll Call 36](#_Toc368775555)

[Table 15: <Staff> Add Subject 37](#_Toc368775556)

[Table 16: <Staff> Edit Subject 39](#_Toc368775557)

[Table 17: <Staff> Active Subject 41](#_Toc368775558)

[Table 18: <Staff> Inactive Subject 42](#_Toc368775559)

[Table 19: <Staff> Add Student 44](#_Toc368775560)

[Table 20: <Staff> Edit Student 47](#_Toc368775561)

[Table 21: <Staff> Import Student List 48](#_Toc368775562)

[Table 22: <Staff>Add Instructor 50](#_Toc368775563)

[Table 23: <Staff> Edit Instructor 52](#_Toc368775564)

[Table 24: <Staff> Add Class 54](#_Toc368775565)

[Table 25 <Staff> Edit Class 55](#_Toc368775566)

[Table 26: <Staff> Report Attendance by Class 57](#_Toc368775567)

[Table 27: <Staff> Report Attendance by Student 58](#_Toc368775568)

[Table 28: <Staff> Report Attendance by Block 60](#_Toc368775569)

[Table 29: <Staff> Report Attendance by Semester 61](#_Toc368775570)

[Table 30: <Instructor> Check Attendance by Taking Picture 65](#_Toc368775571)

[Table 31: <Instructor> Check Attendance Manually 68](#_Toc368775572)

[Table 32: <Instructor> View Roll Call List 70](#_Toc368775573)

[Table 33: <Instructor> Report Attendance by Class 72](#_Toc368775574)

[Table 34: <Student> CheckPresent Rate 74](#_Toc368775575)

[Table 35: <Guest> Login 78](#_Toc368775576)

[Table 36: < Authorized User> Change Password 80](#_Toc368775577)

[Table 37: <User> Logout 82](#_Toc368775578)

# List of Figures

[Figure 1: System Overview Use Case 10](#_Toc368775579)

[Figure 2: <Admin> Overview Use Case 11](#_Toc368775580)

[Figure 3: <Admin> Add Image for Single Student 11](#_Toc368775581)

[Figure 4: <Admin> Add Image for Multi Student 14](#_Toc368775582)

[Figure 5: <Admin> Remove Student Image 16](#_Toc368775583)

[Figure 6: <Admin> Create Account 18](#_Toc368775584)

[Figure 7: <Admin> Active Account 20](#_Toc368775585)

[Figure 8: <Admin> Inactive Account 21](#_Toc368775586)

[Figure 9: <Admin> Config System 23](#_Toc368775587)

[Figure 10: <System> Overview Use Case 24](#_Toc368775588)

[Figure 11: <System> Face Detection 25](#_Toc368775589)

[Figure 12: <System> Recognize Student 26](#_Toc368775590)

[Figure 13: <System> Auto Free Storage Space 28](#_Toc368775591)

[Figure 14: <System> Auto Active Roll Call 29](#_Toc368775592)

[Figure 15: <System> Auto Inactive Roll Call 30](#_Toc368775593)

[Figure 16: <Staff> Overview Use Case 31](#_Toc368775594)

[Figure 17: <Staff>Add Roll Call 31](#_Toc368775595)

[Figure 18: <Staff>Edit Roll Call 34](#_Toc368775596)

[Figure 19: <Staff> Add Subject 36](#_Toc368775597)

[Figure 20: <Staff> Edit Subject 38](#_Toc368775598)

[Figure 21: <Staff> Active Subject 39](#_Toc368775599)

[Figure 22: <Staff> Inactive Subject 41](#_Toc368775600)

[Figure 23: <Staff> Add Student 43](#_Toc368775601)

[Figure 24: <Staff> Edit Student 45](#_Toc368775602)

[Figure 25: <Staff> Import Student List 47](#_Toc368775603)

[Figure 26: <Staff>Add Instructor 49](#_Toc368775604)

[Figure 27: <Staff>Edit Instructor 50](#_Toc368775605)

[Figure 28: <Staff> Add Class 52](#_Toc368775606)

[Figure 29: <Staff> Edit Class 54](#_Toc368775607)

[Figure 30: <Staff> Report Attendance by Class 56](#_Toc368775608)

[Figure 31: <Staff> Report Attendance by Student 57](#_Toc368775609)

[Figure 32: <Staff> Report Attendance by Block 59](#_Toc368775610)

[Figure 33: <Staff> Report Attendance by Semester 60](#_Toc368775611)

[Figure 34: <Staff> Export Report 62](#_Toc368775612)

[Figure 35: <Instructor> Overview Use Case 63](#_Toc368775613)

[Figure 36: <Instructor> Capture and Show result attendance 63](#_Toc368775614)

[Figure 37: <Instructor> Check Attendance Manually 66](#_Toc368775615)

[Figure 38: <Instructor> View Roll Call List 69](#_Toc368775616)

[Figure 39: <Instructor> Report Attendance by Class 71](#_Toc368775617)

[Figure 40: <Student> Overview Use Case 72](#_Toc368775618)

[Figure 41: <Student> CheckPresent Rate 73](#_Toc368775619)

[Figure 42: <Guest> Overview Use Case 77](#_Toc368775620)

[Figure 43: <Guest> Login 77](#_Toc368775621)

[Figure 44: < Authorized User> Overview Use Case 79](#_Toc368775622)

[Figure 45: < Authorized User> Change Password 79](#_Toc368775623)

[Figure 46: < Authorized User> Logout 81](#_Toc368775624)

[Figure 47: ERD - Conceptual Diagram 83](#_Toc368775625)

# Definitions, Acronyms, and Abbreviations

|  |  |
| --- | --- |
| SmartB | Smart Buy |

# Report No.3 Software Requirement Specification

## User Requirement Specification

### 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.
* Ask for system suggestion.
* Search product.
* Add product to cart.

### Member Requirement

Member is guest who uses his account to login to the system. Member can use additional functions, such as:

* Update product price.
* Save product history.

### Staff Requirement

Staff is the person who manages markets and products. Staff can use these functions:

* Manage market.
* Manage product.

### Admin Requirement

Admin is the person who manages the system. Admin can use these functions:

* Configure system.
* Make statistics.
* Manage user.

### System Requirement

System is also an actor, run in the background to keep the system working. System can do functions:

* Parse data.

## System Requirement Specification

### External Interface Requirement

#### User Interface

* + The interface of website is clear, do not annoy customer.
  + The interface of mobile app must be clear, compatible with touch screen. The size of controls must be big enough to touch on smartphone.
  + The error, warning messages must be make clear, easy to understand. Error warning does not disturb customer.

#### Hardware Interface

* + The system will use the standard hardware and data communications resources of a standard computer.

#### Software Interface

* + Firefox Browser, Chromes with Resolution (1280\*800) or bigger and support JavaScript and HTML5
  + Smartphone with Android 4.1 or above. Screen size (70 x 120 mm) or bigger.

#### Communication Protocol

* Website using HTTP protocol for communication between the web browser and the web server.
* Mobile app using HTTP protocol for communicating between app and web service.

### System Overview Use Case

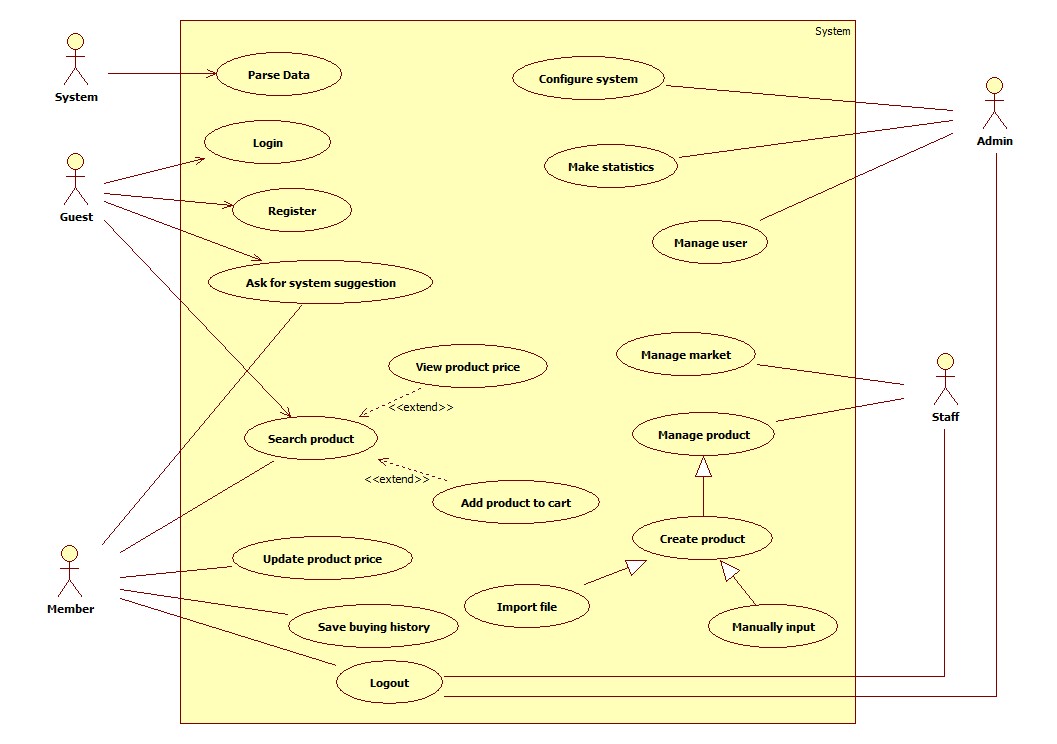


Figure 1: System Overview Use Case

### List of Use Case

#### <Admin> Overview Use Cases

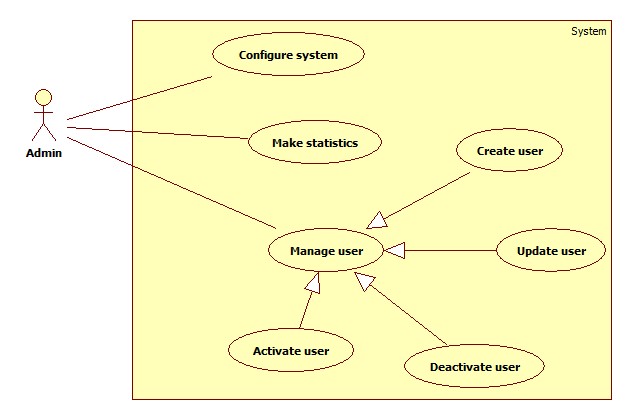


Figure 2 - <Admin> Overview use case

##### <Admin> Configure System

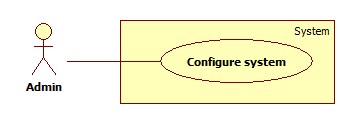


Figure 3: <Admin> Configure System

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE – SB001** | | | |
| **Use Case No.** | SB001 | **Use Case Version** | 2.0 |
| **Use Case Name** | Configure system | | |
| **Author** | Doan Ho Anh Triet | | |
| **Date** | 15/02/2014 | **Priority** | High |
| **Actor:**   * Admin   **Summary:**   * This use case allows admin to configure the system.   **Goal:**   * System parameters will be configured by admin.   **Triggers:**   * Admin wants to change system parameter, such as epsilon number or parse time. * To go to “System Configuration” screen, admin clicks at “System configuration” menu.   **Preconditions:**   * User must log in the system with the admin role.   **Post Conditions:**   * **Success:** New parameter will be saved to configuration file. * **Fail:** Nothing is saved to configuration file.   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | Click at “System Configuration” link in the menu. | Show “System Configuration” page which contains:   * Epsilon: textbox, min length: 1, max length: 3, type: integer number, min value: 1, max value: 100, required. * Thời gian parse: time picker, min value: 0:00, max value: 23:59, required. * Lưu: button. | | 2 | Admin changes system parameters. |  | | 3 | Click on “Lưu” button.  [Exception 1, 2, 3] | * Save data to configuration file. * Show success message. |   **Alternative Scenario:** N/A  **Exceptions:**   |  |  |  | | --- | --- | --- | | No | Actor Action | System Response | | 1 | Epsilon value out of range. | Show error message: “Epsilon phải lớn hơn 0 và nhỏ hơn hoặc bằng 100!” | | 2 | Epsilon is not a number. | Show error message: “Epsilon phải là số!” | | 3 | Time format is not correct. | Show error message: “Thời gian phải đúng định dạng!” |   **Relationships:** N/A  **Business Rules:** N/A | | | |

Table 1: <Admin> Configure System

##### <Admin> Make Statistics

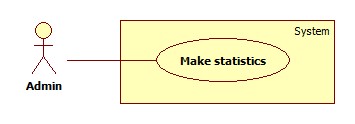


Figure 4: <Admin> Make Statistics

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE – SB002** | | | |
| **Use Case No.** | SB002 | **Use Case Version** | 2.0 |
| **Use Case Name** | Make statistics | | |
| **Author** | Doan Ho Anh Triet | | |
| **Date** | 15/02/2014 | **Priority** | Normal |
| **Actor:** Admin  **Summary:**  **Goal:**  **Triggers:**  **Preconditions:**   * User must log in the system with admin role.   **Post Conditions:**   * **Success:** * **Fail:**   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 |  |  |   **Alternative Scenario:**   |  |  |  | | --- | --- | --- | | No | Actor Action | System Response | | 1 |  |  |   **Exceptions:**   |  |  |  | | --- | --- | --- | | No | Actor Action | System Response | | 1 |  |  |   **Relationships:**  **Business Rules:** | | | |

Table 2: <Admin> Make Statistics

##### <Admin> Create User

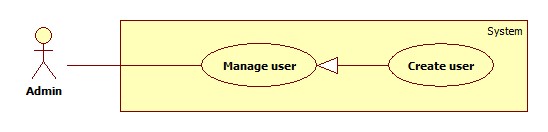


Figure 5: <Admin> Create User

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE – SB003** | | | |
| **Use Case No.** | SB003 | **Use Case Version** | 2.0 |
| **Use Case Name** | Create user | | |
| **Author** | Doan Ho Anh Triet | | |
| **Date** | 15/02/2014 | **Priority** | Normal |
| **Actor:**   * Admin   **Summary:**   * This user case allows admin to create new user account for the system.   **Goal:**   * New user account will be added to the system.   **Triggers:**   * Admin wants to create new user account. * Navigate to “Quản lý tài khoản” and click “Tạo tài khoản” button.   **Preconditions:**   * User must log in the system with admin role.   **Post Conditions:**   * **Success:** New user account will be added to the database. * **Fail:** Nothing is added to the database.   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | Admin clicks “Tạo tài khoản” link. | Navigate to “Tạo tài khoản” page which contains:   * Username: textbox, min length: 6, max length: 30, required. * Password: textbox, min length: 6, max length: 20, required. * Role: drop down list, required. * Tạo tài khoản: button. * Xóa trắng: button. | | 2 | Admin fills data to the form. |  | | 3 | Admin clicks “Tạo tài khoản” button. [Alternative 1], [Exception 1, 2, 3, 4] | Insert new user account to the database and redirect to “Quản lý tài khoản” page. |   **Alternative Scenario:**   |  |  |  | | --- | --- | --- | | No | Actor Action | System Response | | 1 | Admin clicks “Xóa trắng button”. | Clear all input data. |   **Exceptions:**   |  |  |  | | --- | --- | --- | | No | Actor Action | System Response | | 1 | Length of username is not in range. | Show error message: “Độ dài username phải từ 6-30 ký tự!” | | 2 | Length of password is not in range. | Show error message: “Độ dài password phải từ 6-20 ký tự!” | | 3 | Role is not specified. | Show error message: “Phải phân quyền cho user!” | | 4 | Username’s existed already. | Show error message: “Trùng tên đăng nhập! Vui lòng chọn tên khác!” |   **Relationships:** Manage user  **Business Rules:**   * Each created account must be in active state when insert to database. | | | |

Table 3: <Admin> Create User

##### <Admin> Update User

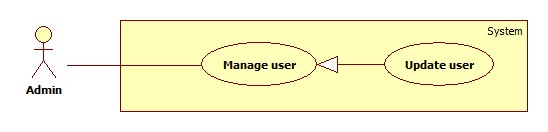


Figure 6: <Admin> Update User

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE – SB004** | | | |
| **Use Case No.** | SB004 | **Use Case Version** | 2.0 |
| **Use Case Name** | Update user | | |
| **Author** | Doan Ho Anh Triet | | |
| **Date** | 16/02/2014 | **Priority** | Normal |
| **Actor:**   * Admin   **Summary:**   * This use case allows admin to update information for user accounts.   **Goal:**   * New information will be updated to the database.   **Triggers:**   * Admin wants to change some information of members or staffs.   **Preconditions:**   * User must log in system with admin role.   **Post Conditions:**   * **Success:** New information will be updated to the database. * **Fail:** Nothing is updated.   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | Click menu item “Quản lý tài khoản”. | Navigate to “Quản lý tài khoản” page which contains:   * A table with 4 columns:   + Số thứ tự.   + Tên đăng nhập.   + Trạng thái.   + Hành động. * Tạo tài khoản: button. | | 2 | Click at username. | Navigate to “Thông tin tài khoản” page, which contains:   * Username: textbox, min length: 6, max length: 30, disabled. * Password: textbox, min length: 6, max length: 20, required. * Email: textbox, min length: 0, max length: 100, regular expression: ^[a-zA-Z0-9.!#$%&'\*+/=?^\_`{|}~-]+@[a-zA-Z0-9](?:[a-zA-Z0-9-]{0,61}[a-zA-Z0-9])?(?:\.[a-zA-Z0-9](?:[a-zA-Z0-9-]{0,61}[a-zA-Z0-9])?)\*$ * Role: drop down list, required. * Cập nhật: button. | | 3 | Admin updates information. |  | | 4 | Click “Cập nhật” button. | New information will be updated to the database. |   **Alternative Scenario:** N/A  **Exceptions:**   |  |  |  | | --- | --- | --- | | No | Actor Action | System Response | | 1 | Length of password is not in range. | Show error message: “Độ dài password phải từ 6-20 ký tự!” | | 2 | Role is not specified. | Show error message: “Phải phân quyền cho user!” | | 3 | Invalid email format. | Show error message: “Email phải đúng định dạng!” |   **Relationships:** Manage User  **Business Rules:**   * Editing username is forbidden. | | | |

Table 4: <Admin> Update User

##### <Admin> Activate User

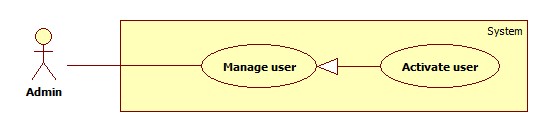


Figure 7: <Admin> Active User

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE – SB005** | | | |
| **Use Case No.** | SB005 | **Use Case Version** | 2.0 |
| **Use Case Name** | Activate User | | |
| **Author** | Doan Ho Anh Triet | | |
| **Date** | 16/02/2014 | **Priority** | Normal |
| **Actor:**   * Admin   **Summary:**   * This user case allows admin to activate a user account.   **Goal:**   * Deactivated accounts will be activated again.   **Triggers:**   * Admin wants to activate an account so that it can be used to log in the system again.   **Preconditions:**   * User must log in the system with admin role.   **Post Conditions:**   * **Success:** The selected account will be activated. * **Fail:** No account is activated.   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | Click menu item “Quản lý tài khoản”. | Navigate to “Quản lý tài khoản” page which contains:   * A table with 4 column:   + Số thứ tự.   + Tên đăng nhập.   + Trạng thái. * Tạo tài khoản: button. | | 2 | Check the checkbox in “Trạng thái” cell of a deactivated account. | Selected account is activated again. [Exception 1] |   **Alternative Scenario:** N/A  **Exceptions:**   |  |  |  | | --- | --- | --- | | No | Actor Action | System Response | | 1 | Cannot activate the account due to database connection. | Show error message: “Không thể kích hoạt! Vui lòng thử lại sau.” |   **Relationships:** Manage User  **Business Rules:**   * Admins can’t activate/deactivate account of themselves or other admins. | | | |

Table 5: <Admin> Activate User

##### <Admin> Deactivate User

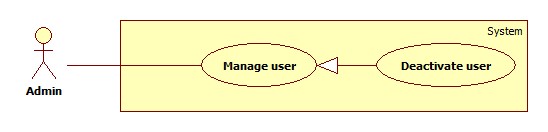


Figure 8: <Admin> Deactivate User

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE – SB006** | | | |
| **Use Case No.** | SB006 | **Use Case Version** | 2.0 |
| **Use Case Name** | Deactivate User | | |
| **Author** | Doan Ho Anh Triet | | |
| **Date** | 16/02/2014 | **Priority** | Normal |
| **Actor:**   * Admin   **Summary:**   * This user case allows admin to deactivate an account.   **Goal:**   * An active account will be deactivated.   **Triggers:**   * In our system, when we do not want a user to login, we deactivate his/her account. That account can be activated later.   **Preconditions:**   * User must log in the system with admin role.   **Post Conditions:**   * **Success:** The selected account will be deactivated. * **Fail:** No account is deactivated.   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | Click menu item “Quản lý tài khoản”. | Navigate to “Quản lý tài khoản” page which contains:   * A table with 4 column:   + Số thứ tự.   + Tên đăng nhập.   + Trạng thái. * Tạo tài khoản: button. | | 2 | Uncheck the checkbox in “Trạng thái” cell of an active account. | The selected account will be deactivated.  [Exception 1] |   **Alternative Scenario:** N/A  **Exceptions:**   |  |  |  | | --- | --- | --- | | No | Actor Action | System Response | | 1 | Cannot deactivate the account due to database connection. | Show error message: “Không thể kích hoạt! Vui lòng thử lại sau.” |   **Relationships:** Manage Account  **Business Rules:**   * Admins can’t activate/deactivate account of themselves or other admins. | | | |

Table 6: <Admin> Deactivate User

##### <Admin> Force Parse Data

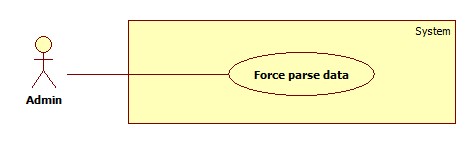


Figure 12: <Admin> Force Parse Data

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE – SB007** | | | |
| **Use Case No.** | SB007 | **Use Case Version** | 2.0 |
| **Use Case Name** | Force Parse Data | | |
| **Author** | Doan Ho Anh Triet | | |
| **Date** | 16/02/2014 | **Priority** | High |
| **Actor:**   * Admin.   **Summary:**   * Admin can force system to parse data at any time.   **Goal:**   * Get data from many websites.   **Triggers:**   * Admin wants to run the parser at that time. * Admin clicks “Parse” button.   **Preconditions:**   * User logged in with “Admin” role.   **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 | Staff clicks “Parse” button. | * Disable “Parse” button. Change its text into “Parsing…” * 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. * Enable “Parse” button. Change its text into “Parse”. |   **Alternative Scenario:**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | Staff clicks “Parse” button. | * If fetched product is already in the database, update its information. * Generate log file. |   **Exceptions:**   |  |  |  | | --- | --- | --- | | No | Actor Action | System Response | | 1 | Data is not valid. | * Don’t insert to database. * Generate log file. |   **Relationships:** N/A  **Business Rules:**   * Refer to “Auto Parse Data” use case. | | | |

Table 9: <Admin> Force Parse Data

#### <System> Overview Use Case

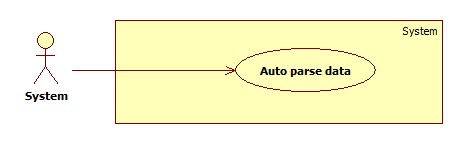


Figure 10: <System> Overview Use Case

##### <System> Auto Parse Data

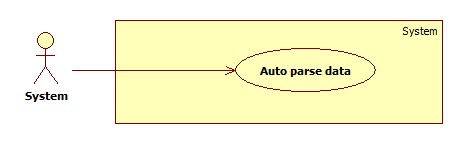


Figure 11: <System> Auto Parse Data

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE – SB008** | | | |
| **Use Case No.** | SB008 | **Use Case Version** | 2.0 |
| **Use Case Name** | Auto Parse Data | | |
| **Author** | Doan Ho Anh Triet | | |
| **Date** | 16/02/2014 | **Priority** | High |
| **Actor:**   * System.   **Summary:**   * System can parse data automatically from many websites at specified time.   **Goal:**   * Get data 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 product 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 price is negative or not a number, consider it invalid. * Log file structure:   SMART BUY LOG FILE  Tạo file lúc: {Created date}, {Created time}   |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | | STT | Link | Thời gian parse | Dạng dữ liệu | Tổng sản phẩm | Insert thành công | Insert thất bại | |  |  |  |  |  |  |  | |  |  |  |  |  |  |  |   Tổng thời gian parse dạng {Data type}: {Elapsed time}  Tổng thời gian parse: {Total elapsed time}  Tổng sản phẩm parse được: {Total parsed product} | | | |

Table 8: <System> Auto Parse Data

#### <Staff >Overview Use Case

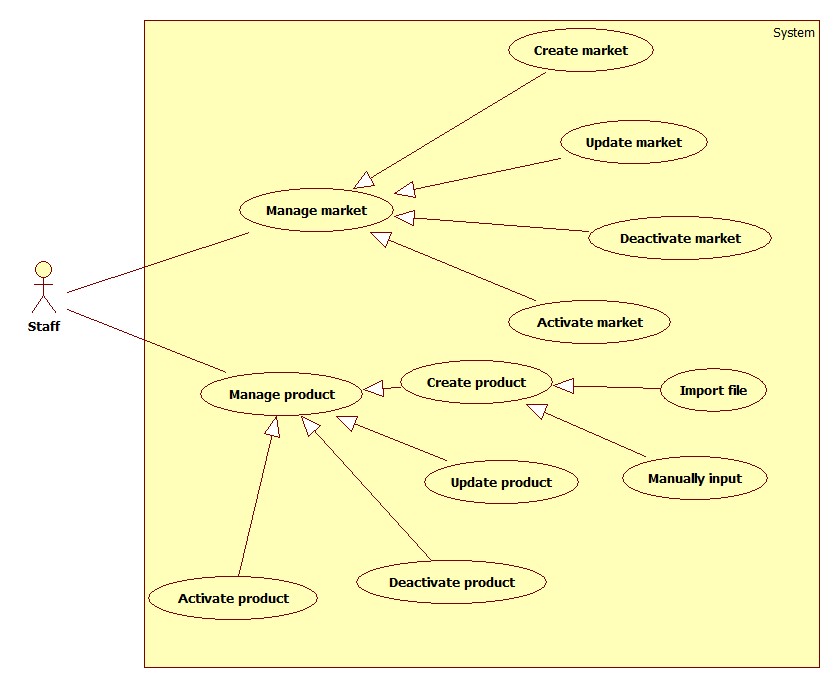


Figure 16: <Staff> Overview Use Case

##### <Staff> Create Market

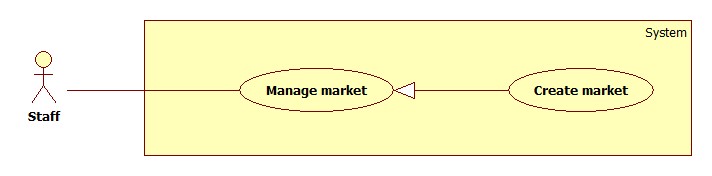


Figure 17: <Staff> Create Market

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE – SB009** | | | |
| **Use Case No.** | SB009 | **Use Case Version** | 2.0 |
| **Use Case Name** | Create Market | | |
| **Author** | Doan Ho Anh Triet | | |
| **Date** | 19/02/2014 | **Priority** | Normal |
| **Actor:**   * Staff.   **Summary:**   * This use case allows staff to create a new market.   **Goal:**   * A new market added to the database.   **Triggers:**   * Staff wants to create a new market. * To create new market, staff must do the following steps:   + Click “Quản lý chợ” on the menu.   + Click “Tạo chợ mới” button.   **Preconditions:**   * User must log in the system with staff role.   **Post Conditions:**   * **Success**: New market will be added to the database. * **Fail:** Nothing is added to the database. Show error message.   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | Click “Quản lý chợ” in menu bar. | Navigate to “Quản lý chợ” page which contains:   * A table has 4 columns:   + Số thứ tự   + Tên chợ   + Địa chỉ   + Trạng thái * Tạo chợ mới: button | | 2 | Click “Tạo chợ mới” button. | Navigate to “Tạo chợ mới” page, which contains:   * Tên chợ: textbox, min length: 6, max length: 50, required. * Địa chỉ: textbox, min length: 6, max length: 100. * Vĩ độ: textbox, decimal number only, min value: -90, max value: 90, min length: 1, max length: 10 * Kinh độ: textbox, decimal number only, min value: -180, max value: 180, min lengh: 1, max length: 11. * Tạo mới: button * Xóa trắng: button | | 3 | Input market information, click “Tạo mới” button. [Alternative 1] | Add new market to the database. Redirect to “Quản lý chợ” page.  [Exception 1, 2, 3, 4, 5, 6, 7, 8, 9] |   **Alternative Scenario:**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | Click “Xóa trắng” button | Clear all input fields. |   **Exceptions:**   |  |  |  | | --- | --- | --- | | No | Actor Action | System Response | | 1 | Duplicate “Tên chợ”. | Show error message: “Tên này đã có trong hệ thống. Vui lòng chọn tên khác!” | | 2 | Length of “Tên chợ” is not in range. | Show error message: “Tên chợ phải từ 6 đến 50 ký tự!” | | 3 | Length of “Địa chỉ” is not in range. | Show error message: “Địa chỉ phải từ 6 đến 100 ký tự!” | | 4 | “Vĩ độ” is not a number. | Show error message: “Vĩ độ phải là số!” | | 5 | Length of “Vĩ độ” is not in range. | Show error message: “Vĩ độ có độ dài từ 1 đến 10 ký tự!” | | 6 | Value of “Vĩ độ” is not in range. | Show error message: “Giá trị của vĩ độ phải từ 0 đến 90!” | | 7 | “Kinh độ” is not a number. | Show error message: “Kinh độ phải là số!” | | 8 | Length of “Kinh độ” is not in range. | Show error message: “Kinh độ có độ dài từ 1 đến 11 ký tự!” | | 9 | Value of “Kinh độ” is not in range. | Show error message: “Giá trị của kinh độ phải từ -180 đến 180!” |   **Relationships:** Manage market.  **Business Rules:**   * We should apply Google Map auto complete to “Địa chỉ” textbox. * User can only input “Địa chỉ” or “Vĩ độ”, “Kinh độ” pair, not both. | | | |

Table 13: <Staff> Create Market

##### <Staff> Update Market

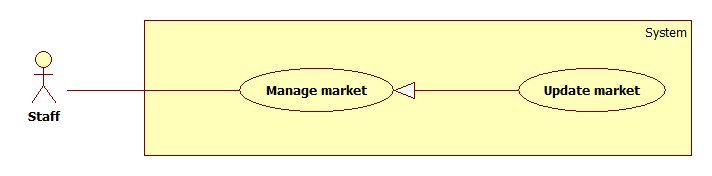


Figure 18: <Staff> Update Market

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE – SB010** | | | |
| **Use Case No.** | SB010 | **Use Case Version** | 2.0 |
| **Use Case Name** | Update Market | | |
| **Author** | Doan Ho Anh Triet | | |
| **Date** | 19/02/2014 | **Priority** | Normal |
| **Actor:**   * Staff.   **Summary:**   * This use case allows staff to update market information.   **Goal:**   * Update new information to the market.   **Triggers:**   * Staff wants to update information of markets. * To update information of markets, staff must do the following steps:   + Click “Quản lý chợ” in the menu.   + Click at the name of the market which staff wants to change information.   **Preconditions:**   * User must log in the system with staff role.   **Post Conditions:**   * **Success**: Market information will be updated. * **Fail:** Nothing is updated. Show error message.   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | Click “Quản lý chợ” in menu bar. | Navigate to “Quản lý chợ” page which contains:   * A table has 4 columns:   + Số thứ tự   + Tên chợ   + Địa chỉ   + Trạng thái * Tạo chợ mới: button | | 2 | Click at the market name. | Navigate to “Thông tin chợ” page, which contains:   * Tên chợ: textbox, min length: 6, max length: 50, required. * Địa chỉ: textbox, min length: 6, max length: 100. * Vĩ độ: textbox, decimal number only, min value: -90, max value: 90, min length: 1, max length: 10 * Kinh độ: textbox, decimal number only, min value: -180, max value: 180, min lengh: 1, max length: 11. * Cập nhật: button | | 3 | Edit market information, click “Cập nhật” button. | The system will update new market information to the database. Redirect to “Quản lý chợ” page.  [Exception 1, 2, 3, 4, 5, 6, 7, 8, 9] |   **Alternative Scenario:** N/A  **Exceptions:**   |  |  |  | | --- | --- | --- | | No | Actor Action | System Response | | 1 | Duplicate “Tên chợ”. | Show error message: “Tên này đã có trong hệ thống. Vui lòng chọn tên khác!” | | 2 | Length of “Tên chợ” is not in range. | Show error message: “Tên chợ phải từ 6 đến 50 ký tự!” | | 3 | Length of “Địa chỉ” is not in range. | Show error message: “Địa chỉ phải từ 6 đến 100 ký tự!” | | 4 | “Vĩ độ” is not a number. | Show error message: “Vĩ độ phải là số!” | | 5 | Length of “Vĩ độ” is not in range. | Show error message: “Vĩ độ có độ dài từ 1 đến 10 ký tự!” | | 6 | Value of “Vĩ độ” is not in range. | Show error message: “Giá trị của vĩ độ phải từ 0 đến 90!” | | 7 | “Kinh độ” is not a number. | Show error message: “Kinh độ phải là số!” | | 8 | Length of “Kinh độ” is not in range. | Show error message: “Kinh độ có độ dài từ 1 đến 11 ký tự!” | | 9 | Value of “Kinh độ” is not in range. | Show error message: “Giá trị của kinh độ phải từ -180 đến 180!” |   **Relationships:** Manage market.  **Business Rules:**   * We should apply Google Map auto complete to “Địa chỉ” textbox. * User can only input “Địa chỉ” or “Vĩ độ”, “Kinh độ” pair, not both. | | | |

Table 14: <Staff> Update Market

##### <Staff> Deactivate Market

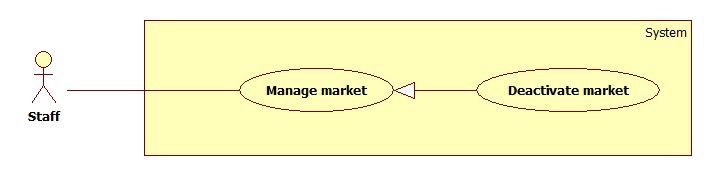


Figure 19: <Staff> Deactivate Market

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE – SB011** | | | |
| **Use Case No.** | SU011 | **Use Case Version** | 2.0 |
| **Use Case Name** | Deactivate Market | | |
| **Author** | Doan Ho Anh Triet | | |
| **Date** | 20/02/2014 | **Priority** | Normal |
| **Actor:**   * Staff.   **Summary:**   * This use case allows staff to deactivate market in the system.   **Goal:**   * Markets will be deactivated.   **Triggers:**   * Staff wants to deactivate a market so that it is no longer available in the system. * To deactivate a market, staff must do the following steps:   + Click “Quản lý chợ” item in the menu.   + Uncheck the status checkbox of each active market.   **Preconditions:**   * User must log in the system with staff role.   **Post Conditions:**   * **Success**: The selected market will be deactivated. * **Fail:** Nothing is changed. Show error message.   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | Click “Quản lý chợ” in menu bar. | Navigate to “Quản lý chợ” page which contains:   * A table has 4 columns:   + Số thứ tự   + Tên chợ   + Địa chỉ   + Trạng thái   Tạo chợ mới: button | | 2 | Uncheck the checkbox in “Trạng thái” cell of an active market. | The selected market will be deactivated. [Exception 1]  Show success message: “Thao tác thành công!” |   **Alternative Scenario:** N/A  **Exceptions:**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | Cannot deactivate the market due to database connection problem. | Show error message: “Thao tác không thể thực hiện. Vui lòng thử lại sau!” |   **Relationships:** Manage Market.  **Business Rules:**   * Cannot deactivate virtual market. | | | |

Table 15: <Staff> Deactivate Market

##### <Staff> Activate Market

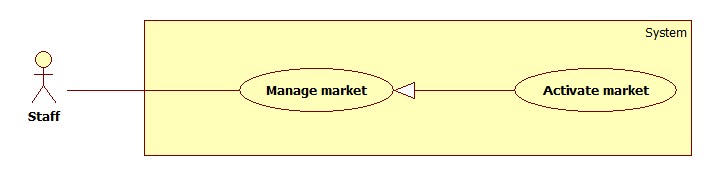


Figure 20: <Staff> Activate Market

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE – SB012** | | | |
| **Use Case No.** | SB012 | **Use Case Version** | 2.0 |
| **Use Case Name** | Activate Market | | |
| **Author** | Doan Ho Anh Triet | | |
| **Date** | 20/02/2014 | **Priority** | Normal |
| **Actor:**   * Staff.   **Summary:**   * This use case allows staff to activate the market.   **Goal:**   * Selected market will be activated.   **Triggers:**   * Staff wants to activate a market so that it can be accessed from the system again. * To activate the market, staff must do the following steps:   + Click “Quản lý chợ” item in the menu.   + Check the status checkbox of each deactivated market   **Preconditions:**   * User must log in the system with staff role.   **Post Conditions:**   * **Success**: Market will be activated again. * **Fail:** Nothing is changed. Show error message.   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | Click “Quản lý chợ” in menu bar. | Navigate to “Quản lý chợ” page which contains:   * A table has 4 columns:   + Số thứ tự   + Tên chợ   + Địa chỉ   + Trạng thái   Tạo chợ mới: button | | 2 | Check the checkbox in “Trạng thái” cell of an deactivated market. | The selected market will be deactivated. [Exception 1]  Show success message: “Thao tác thành công!” |   **Alternative Scenario:** N/A  **Exceptions:**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | Cannot activate the market due to database connection problem. | Show error message: “Thao tác không thể thực hiện. Vui lòng thử lại sau!” |   **Relationships:** Manage Market.  **Business Rules:** N/A | | | |

Table 16: <Staff> Activate Market

##### <Staff> Active Subject

Use Case Diagram



Figure 21: <Staff> Active Subject

Use Case Specification

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE – SU005** | | | |
| **Use Case No.** | SU005 | **Use Case Version** | 2.0 |
| **Use Case Name** | Active Subject. | | |
| **Author** | Nguyen Quoc Huy. | | |
| **Date** | 19/09/2013 | **Priority** | Normal |
| **Actor:** Staff.  **Summary:**  Staff can use this case to active subject.  **Goal:**  Active a subject for new block.  **Triggers:**   * Staff can active a subject have exit. * Staff must do these step to Active subject: * In Home Page, click on link “Manage subject”. * Click on button “Inactive” in row, which staff wants to edit. * The system will show a message box confirm. * Click on button “Yes”.   **Preconditions:**   * User must logged in the system with the role is staff.   **Post Conditions:**   * **Success**: A Subject will be active. The system will still in current manage subject page. * **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   Currnt Tab is Subject List. its include table subject ,for each row:   * No. : label * Major: label * Name: label * Slot (/day): label * Detail: button * Active/Inactive: button | | 2 | Click button “Inactive” in row, which staff wants to edit. | The system will show confirm message box:   * Text: “Do you want to active this object?” * Button “Yes” * Button “No” | | 3 | Click on button “Yes”  [Alternative 1] | The system will change status subject to active and still in current page. |   **Alternative Scenario:**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | In confirm message box, staff click on button “No” | The system will still in current page with no value is changed. |   **Exceptions:** N/A  **Relationships:** Manage Subject.  **Business Rules:**   * The button active has color blue and button inactive has color orange. | | | |

Table 17: <Staff> Active Subject

##### <Staff> Inactive Subject

Use Case Diagram



Figure 22: <Staff> Inactive Subject

Use Case Specification

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE – SU006** | | | |
| **Use Case No.** | SU006 | **Use Case Version** | 2.0 |
| **Use Case Name** | Inactive Subject. | | |
| **Author** | Nguyen Quoc Huy. | | |
| **Date** | 19/09/2013 | **Priority** | Normal |
| **Actor:** Staff.  **Summary:**  Staff can use this case to inactive subject.  **Goal:**  Inactive a subject.  **Triggers:**   * Staff can inactive a subject have exit. * Staff must do these step to Inactive subject: * In Home Page, click on link “Manage subject”. * Click on button “Active” in row, which staff wants to edit. * The system will show a message box confirm. * Click on button “Yes”.   **Preconditions:**   * User must logged in the system with the role is staff.   **Post Conditions:**   * **Success**: A Subject will be inactive. The system will still in current manage subject page. * **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   Currnt Tab is Subject List. its include table subject ,for each row:   * No. : label * Major: label * Name: label * Slot (/day): label * Detail: button * Active/Inactive: button | | 2 | Click button “Active” in row, which staff wants to edit. | The system will show confirm message box:   * Text: “Do you want to inactive this object?” * Button “Yes”. * Button “No”. | | 3 | Click on button “Yes”  [Alternative 1] | The system will change status subject to inactive and still in current page. |   **Alternative Scenario:**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | In confirm message box, staff click on button “No” | The system will still in current page with no value is changed. |   **Exceptions:** N/A  **Relationships:** Manage Subject.  **Business Rules:**   * The button active has color blue and button inactive has color orange. | | | |

Table 18: <Staff> Inactive Subject

##### <Staff>Add Student

Use Case Diagram



Figure 23: <Staff> Add Student

Use Case Specification

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE – SU007** | | | |
| **Use Case No.** | SU007 | **Use Case Version** | 2.0 |
| **Use Case Name** | Add Student. | | |
| **Author** | Nguyen Quoc Huy. | | |
| **Date** | 19/09/2013 | **Priority** | Normal |
| **Actor:** Staff.  **Summary:**  Staff can create a new student.  **Goal:**  A new student include: name, email, student id, birthday, citizen id, address will be added to database.  **Triggers:**   * Staff can create new student with all information: name, email, student id, birthday, citizen id, address... * Staff must do these step to create new student:   + On Home page, click on link “Manage student” in menu sidebar and Manage student page will be showed.   + Then, choose “New student” tab.   + Then, choose the class, input name, id, email, citizen id, address for student.   + Click on button “Add”.   **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 2 tabs:   * Student List * New Student * Import Student List   Current Tab is Student List, its include table student , for each row:   * No: label * Name: label * Current Classes: label * Email: label * Student ID: label * Detail: button * Active/Inactive: button | | 2 | Click “New Student” tab. | Display “New Student” tab with :   * Name: textbox (min length: 1 max length: 30, required). * Student ID: textbox (min length: 1 max length: 30, required). * Birthday: date picker (value: current date.) * Address: textbox (min length: 1 max length: 50, required). * Email: textbox (match regular expression:   ^([\w\.])+@([\w])+\.(\w){2,6}(\.([\w]){2,4})\*$  , required).   * Citizen ID: (match regular expression:^\d{8}$   , required).   * Button “Add” * Button “Cancel” | | 3 | Input Student information, click “Add” button. | Create new student to database. Redirect student list tab.  [Exception 1,2,3,4] |   **Alternative Scenario:**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | Click “Cancel” button. | All control in “New Student” tab is cleared. |   **Exceptions:**   |  |  |  | | --- | --- | --- | | No | Actor Action | System Response | | 1 | Name’s length is not in range [1,30] | Display “Name is invalid”. | | 2 | Address’s length is not in range [1,50] | Display “Address is invalid”. | | 3 | Student ID’s length is not in range [1,30] | Display “Student ID is invalid”. | | 4 | Email is not matched regular expression. | Display “Invalid email address”. |   **Relationships:** Manage Student.  **Business Rules:** N/A | | | |

Table 19: <Staff> Add Student

##### <Staff>Edit Student

Use Case Diagram



Figure 24: <Staff> Edit Student

Use Case Specification

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE – SU008** | | | |
| **Use Case No.** | SU008 | **Use Case Version** | 2.0 |
| **Use Case Name** | Edit Student. | | |
| **Author** | Nguyen Quoc Huy. | | |
| **Date** | 19/09/2013 | **Priority** | Normal |
| **Actor:** Staff.  **Summary:**  Staff can use this case to edit a student.  **Goal:**  Update new information student includes: name, birthday, citizen ID, email…  **Triggers:**   * Staff can update new information for student to database. * Staff must do these step to edit student: * In Home Page, click on link “Manage Student”. * Click on button “Detail” in row, which staff wants to edit. * In “Student detail” pop-up window, edit information. * Click on button “Update”.   **Preconditions:**   * User must logged in the system with the role is staff.   **Post Conditions:**   * **Success**: A student will be updated. The system will transfer to student list page. * **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 2 tabs:   * Student List * New Student * Import Student List   Current Tab is Student List, its include table student , for each row:   * No: label * Name: label * Current Classes: label * Email: label * Student ID: label * Detail: button * Active/Inactive: button | | 2 | Click button “Detail” in row, which staff wants to edit. | The system will show “Student Detail” pop-up:   * Name: textbox (min length: 1 max length: 30, required). * Student ID: textbox (min length: 1 max length: 30, required, disabled). * Birthday: date picker (value: current date.) * Address: textbox (min length: 1 max length: 50, required). * Email: textbox (match regular expression:   ^([\w\.])+@([\w])+\.(\w){2,6}(\.([\w]){2,4})\*$  , required).   * Citizen ID: (match regular expression:^\d{8}$   , required).   * Update: button * Cancel: button | | 3 | Edit student information, click “Update” button.  [Alternative 1] | The system will update student to database. The system transfer to student list page.  [Exception 1,2,3,4] |   **Alternative Scenario:**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | Click “Cancel” button. | The pop-up is closed. |   **Exceptions:**   |  |  |  | | --- | --- | --- | | No | Actor Action | System Response | | 1 | Name’s length is not in range [1,30] | Display “Name is invalid”. | | 2 | Address’s length is not in range [1,50] | Display “Address is invalid”. | | 3 | Student ID’s length is not in range [1,30] | Display “Student ID is invalid”. | | 4 | Email is not matched regular expression. | Display “Invalid email address”. |   **Relationships:** Manage Student.  **Business Rules:**   * In pop-up window, the value of name, current class, address, citizen id, and birthday will be auto set by student was chosen. | | | |

Table 20: <Staff> Edit Student

##### <Staff>Import Student List

Use Case Diagram



Figure 25: <Staff> Import Student List

Use Case Specification

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE – SU009** | | | |
| **Use Case No.** | SU009 | **Use Case Version** | 2.0 |
| **Use Case Name** | Import student list | | |
| **Author** | Nguyen Quoc Huy. | | |
| **Date** | 19/09/2013 | **Priority** | Normal |
| **Actor:** Staff.  **Summary:**  Staff can use this case to import student list.  **Goal:**  Create multiple student by import from excel file.  **Triggers:**   * User can import an excel file student list. * Staff must do these step to import student list: * In Home Page, click on link “Manage Student”. * Click on tab “Import Student List” * Click on button “Import” * Then, choose the excel file in browser dialog.   **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 2 tabs:   * Student List * New Student * Import Student List   Current Tab is Student List, its include table student ,for each row:   * No: label * Name: label * Current Classes: label * Email: label * Student ID: label * Detail: button * Active/Inactive: button | Click “Manage Student” in menu bar. | Redirect to Manage Student Page, include 2 tabs:   * Student List * New Student   Student List is tab default, its include table student ,for each row:   * “No.” * “Name”. * “Current Classes”. * “Email”. * “Student ID”. * Button “Detail” * Button “Active/Inactive”. | | 2 | Click “Import Student List” tab | Display “Import Student List” tab with :   * Import file: File upload control. * Button “Import” | | 3 | Click on textbox Import file, choose file. Click “Ok” | The system will get path file and show on text box (only allow excel file.) | | 4 | Click on button “Import” | Import student list to database and show a message box:   * Text: “Import is successful!   There are [number of student in file] student and [number of class in file] class were imported to database”   * Button “Ok”.   [Exception 1] |   **Alternative Scenario:** N/A  **Exceptions:**   |  |  |  | | --- | --- | --- | | No | Actor Action | System Response | | 1 | Import file get error. | Show error message, the sheet, line, row where error occur. |   **Relationships:** Manage Student.  **Business Rules:**   * The input excel file must follow a student list excel template. | | | |

Table 21: <Staff> Import Student List

##### <Staff>Add Instructor

Use Case Diagram



Figure 26: <Staff>Add Instructor

Use Case Specification

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE – SU010** | | | |
| **Use Case No.** | SU010 | **Use Case Version** | 2.0 |
| **Use Case Name** | Add instructor. | | |
| **Author** | Nguyen Quoc Huy. | | |
| **Date** | 19/09/2013 | **Priority** | Normal |
| **Actor:** Staff.  **Summary:**  Staff can create a new instructor.  **Goal:**  A new instructor include: name, current subject, current class, and phone number will be added to database.  **Triggers:**   * Staff can create new instructor with all information: name, current subject , current class, phone number … * Staff must do these step to create new instructor:   + On Home page, click on link “Manage Instructor” in menu sidebar and Manage Instructor page will be showed.   + Then, choose “New Instructor” tab.   + Then, choose the Classes, subject, input name, phone number for instructor.   + Click on button “Add”.   **Preconditions:**   * User must logged in the system with the role is staff.   **Post Conditions:**   * **Success**: A new instructor will be created. The instructor will be displayed on instructor 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   Current Tab is Instructor List, include table instructor list, for each instructor:   * No: label * Name: label * Teaching Subject: label * Teaching Class: label * Detail: button * Active/Inactive: button | | 2 | Click “New Instructor” tab. | Display “New Instructor” tab with :   * Instructor name: Text box (min length:1 max length:30, required) * Email: textbox (match regular expression:   ^([\w\.])+@([\w])+\.(\w){2,6}(\.([\w]){2,4})\*$  , required).   * Phone number: text box (min length: 8, max length: 11, match regular expression: ^\s\*\d+\s\*$ ) | | 3 | Input instructor information, click “Add” button. | Create new instructor to database. Redirect instructor list tab.  [Exception 1, 2, 3] |   **Alternative Scenario:** N/A  **Exceptions:**   |  |  |  | | --- | --- | --- | | No | Actor Action | System Response | | 1 | Name’s length is not in range [1,30] | Display “Name is invalid”. | | 2 | Email not matched regular expresion | Display “Email is invalid”. | | 3 | Phone number is not matched regular expresion. | Display “Phone number is invalid”. |   **Relationships:** Manage instructor.  **Business Rules:** N/A | | | |

Table 22: <Staff>Add Instructor

##### <Staff>Edit Instructor

Use Case Diagram



Figure 27: <Staff>Edit Instructor

Use Case Specification

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE – SU011** | | | |
| **Use Case No.** | SU011 | **Use Case Version** | 2.0 |
| **Use Case Name** | Edit instructor. | | |
| **Author** | Nguyen Quoc Huy. | | |
| **Date** | 19/09/2013 | **Priority** | Normal |
| **Actor:** Staff.  **Summary:**  Staff can use this case to edit an instructor.  **Goal:**  Update new information instructor includes: name, current subjects, current classes, phone…  **Triggers:**   * Staff can update new information for instructor to database. * Staff must do these step to edit instructor: * In Home Page, click on link “Manage Instructor”. * Click on button “Detail” in row, which staff wants to edit. * In “Instructor detail” pop-up window, edit information. * Click on button “Update”.   **Preconditions:**   * User must logged in the system with the role is staff.   **Post Conditions:**   * **Success**: An instructor will be updated. The system will transfer to instructor list page. * **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   Current Tab is Instructor List, include table instructor list, for each instructor:   * No: label * Name: label * Teaching Subject: label * Teaching Class: label * Detail: button * Active/Inactive: button | | 2 | Click button “Detail” in row, which staff wants to edit. | The system will show “Instructor detail” pop-up:   * Instructor name: Text box (min length:1 max length:30, required) * Email: textbox (match regular expression:   ^([\w\.])+@([\w])+\.(\w){2,6}(\.([\w]){2,4})\*$  , required).   * Phone number: text box (min length: 8, max length: 11, match regular expression: ^\s\*\d+\s\*$ ) * Update: button * Cancel: button | | 3 | Edit instructor information, click “Update” button.  [Alternative 1] | The system will update instructor to database. The system transfer to instructor list page.  [Exception 1, 2] |   **Alternative Scenario:**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | Click “Cancel” button. | The pop-up is closed. |   **Exceptions:**   |  |  |  | | --- | --- | --- | | No | Actor Action | System Response | | 1 | Name’s length is not in range [1,30] | Display “Name is invalid”. | | 2 | Phone number is not matched regular exception. | Display “Phone number is invalid”. |   **Relationships:** Manage Instructor.  **Business Rules:**   * In pop-up window, the value of name, current class, current subject, phone number will be auto set by instructor was chosen. | | | |

Table 23: <Staff> Edit Instructor

##### <Staff>Add Class

Use Case Diagram



Figure 28: <Staff> Add Class

Use Case Specification

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE – SU012** | | | |
| **Use Case No.** | SU012 | **Use Case Version** | 2.0 |
| **Use Case Name** | Add Class. | | |
| **Author** | Nguyen Quoc Huy. | | |
| **Date** | 19/09/2013 | **Priority** | Normal |
| **Actor:** Staff.  **Summary:**  Staff can create a new Class.  **Goal:**  A new class includes: major, name will be added to database.  **Triggers:**   * Staff can create new class with for every block/semester/year… * Staff must do these step to create new class:   + On Home page, click on link “Manage Class” in menu sidebar and Manage subject page will be showed.   + Then, choose “New Class” tab.   + Then, choose the major, input name for class.   + Click on button “Add”.   **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   Current Tab is Class List, include table class list, for each class:   * No: label * Major: label * Name: label * Show Student List: button * Detail: button * Active/Inactive: button | | 2 | Click “New Class” tab. | Display “New Class” tab with :   * Major: Drop down list (value: Software Engineering, Financial Banking, Business Analysis ) * Name: textbox (min length: 1 max length: 30, required). * Add: button * Cancel: button | | 3 | Input class information, click “Add” button. | Create new class to database. Redirect Class list tab.  [Exception 1, 2] |   **Alternative Scenario:**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | Click “Cancel” button. | The input form is reseted. |   **Exceptions:**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | Name’s length is not in range [1,30] | Display “Name is invalid”. | | 2 | The name class has existed. | Display “The class has existed”. |   **Relationships:** Manage Class.  **Business Rules:** N/A | | | |

Table 24: <Staff> Add Class

##### <Staff>Edit Class

Use Case Diagram



Figure 29: <Staff> Edit Class

Use Case Specification

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE – SU013** | | | |
| **Use Case No.** | SU013 | **Use Case Version** | 2.0 |
| **Use Case Name** | Edit Class. | | |
| **Author** | Nguyen Quoc Huy. | | |
| **Date** | 19/09/2013 | **Priority** | Normal |
| **Actor:** Staff.  **Summary:**  Staff can use this case to edit a class.  **Goal:**  Update new information class includes: major, name …  **Triggers:**   * User can update new information for class to database. * Staff must do these step to Edit class: * In Home Page, click on link “Manage Class”. * Click on button “Detail” in row, which staff wants to edit. * In “Class detail” pop-up window, edit information. * Click on button “Update”.   **Preconditions:**   * User must logged in the system with the role is staff.   **Post Conditions:**   * **Success**: A class will be updated. The system will transfer to class list page. * **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   Current Tab is Class List, include table class list, for each class:   * No: label * Major: label * Name: label * Show Student List: button * Detail: button * Active/Inactive: button | | 2 | Click button “Detail” in row, which staff wants to edit. | The system will show “Class detail” pop-up:   * Major: Drop down list (value: Software Engineering, Financial Banking, Business Analysis ) * Name: textbox (min length: 1 max length: 30, required). * Table: Student List of class. * Update: button * Cancel: button | | 3 | Edit Class information, click “Update” button.  [Alternative 1] | The system will update class to database. The system transfer to class list page. |   **Alternative Scenario:**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | Click “Cancel” button. | The pop-up is closed. |   **Exceptions:**   |  |  |  | | --- | --- | --- | | No | Actor Action | System Response | | 1 | Name’s length is not in range [1,30] | Display “Name is invalid”. |   **Relationships:** Manage Class.  **Business Rules:**   * In pop-up win down, the value of major, name will be auto set by class was chosen. | | | |

Table 25 <Staff> Edit Class

##### <Staff> Report Attendance by Class

Use Case Diagram



Figure 30: <Staff> Report Attendance by Class

Use Case Specification

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE – SU014** | | | |
| **Use Case No.** | SU014 | **Use Case Version** | 2.0 |
| **Use Case Name** | Report Attendance by Class | | |
| **Author** | Nguyen Quoc Huy. | | |
| **Date** | 19/09/2013 | **Priority** | Normal |
| **Actor:** Staff.  **Summary:**  This use case allows staff report attendance of all students by class.  **Goal:**  Staff can report attendance of class.  **Triggers:**   * Staff can report information of class: subject, student, attendance… * Staff must do these step to view report result:   + On Home page, click on link “Report” in menu bar and Report page will be showed.   + Then, in select box “Report by” choose type: Class.   + Then, in drop down list “Name” chose the class name.   + Click on button “Statistic”.   **Preconditions:**   * User must logged in the system with the role is staff.   **Post Conditions:**   * **Success**: A report attendance by class will be showed on screen. * **Fail:** The system will transfer to error page or still in current page.   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | Click “Report” link in menu bar. | Redirect to Report Page, include 4 part:   * Drop down list “Report by” (value: Class, Student, Block, and Semester). * Drop down list “Name” (value: auto set based on drop down list Report by) * Button “Statistic” * Button “Export”. | | 2 | Choose “Class” in drop down list Report by. | System will auto set value based on type Class (just active class) for drop down list “Name”. | | 3 | Chose the class in drop down list “Name”. Click button “Statistic” | System will show result attendance by class on screen.  System still in current “Report” page. |   **Alternative Scenario:** N/A  **Exceptions:** N/A  **Relationships:** Manage Report Attendance.  **Business Rules:**   * System will show attendance result in chosen conditions. * Default value when manager clicks on tab “Report” is: Class. | | | |

Table 26: <Staff> Report Attendance by Class

##### <Staff> Report Attendance by Student

Use Case Diagram



Figure 31: <Staff> Report Attendance by Student

Use Case Specification

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE – SU015** | | | |
| **Use Case No.** | SU015 | **Use Case Version** | 2.0 |
| **Use Case Name** | Report Attendance by Student | | |
| **Author** | Nguyen Quoc Huy. | | |
| **Date** | 19/09/2013 | **Priority** | Normal |
| **Actor:** Staff.  **Summary:**  This use case allows staff report attendance of student.  **Goal:**  Staff can report attendance of student.  **Triggers:**   * Staff can report information of student: student name, class, subject, attendance… * Staff must do these step to view report result:   + On Home page, click on link “Report” in menu bar and Report page will be showed.   + Then, in drop down list “Report by” choose type: Student.   + Then, in drop down list “Name” choose the student name.   + Click on button “Statistic”.   **Preconditions:**   * User must logged in the system with the role is staff.   **Post Conditions:**   * **Success**: A report attendance by student will be showed on screen. * **Fail:** System will not return any value. Transfer to error page or still in current page.   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | Click “Report” link in menu bar. | Redirect to Report Page, include 4 part:   * Drop down list “Report by” (value: Class, Student, Block, and Semester). * Drop down list “Name” (value: auto set based on drop down list Report by) * Button “Statistic” * Button “Export”. | | 2 | Choose “Student” in drop down list Report by. | System will auto set value based on type Student (just active student) for drop down list “Name”. | | 3 | Chose the student in drop down list “Name”. Click button “Statistic”. | System will show result attendance by student on screen.  System still in current “Report” page. |   **Alternative Scenario:** N/A  **Exceptions:** N/A  **Relationships:** Manage Report Attendance.  **Business Rules:**   * System will show attendance result in chosen conditions. * Default value when manager clicks on tab “Report” is: Class. | | | |

Table 27: <Staff> Report Attendance by Student

##### <Staff> Report Attendance by Block

Use Case Diagram



Figure 32: <Staff> Report Attendance by Block

Use Case Specification

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE – SU016** | | | |
| **Use Case No.** | SU016 | **Use Case Version** | 2.0 |
| **Use Case Name** | Report Attendance by Block | | |
| **Author** | Nguyen Quoc Huy. | | |
| **Date** | 19/09/2013 | **Priority** | Normal |
| **Actor:** Staff.  **Summary:**  This use case allows staff report attendance all students by block.  **Goal:**  Staff can report attendance of student.  **Triggers:**   * Staff can report information of student: Block, class, subject, attendance… * Staff must do these step to view report result:   + On Home page, click on link “Report” in menu bar and Report page will be showed.   + Then, in drop down list “Report by” choose type: Block.   + Then, in drop down list “Name” choose the block.   + Click on button “Statistic”.   **Preconditions:**   * User must logged in the system with the role is staff.   **Post Conditions:**   * **Success**: A report attendance by block will be showed on screen. * **Fail:** System will not return any value. Transfer to error page or still in current page.   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | Click “Report” link in menu bar. | Redirect to Report Page, include 4 part:   * Drop down list “Report by” (value: Class, Student, Block, and Semester). * Drop down list “Name” (value: auto set based on drop down list Report by) * Button “Statistic” * Button “Export”. | | 2 | Choose “Block” in drop down list Report by. | System will auto set value based on type Block (just active block) for drop down list “Name”. | | 3 | Chose the Block in drop down list “Name”. Click button “Statistic”. | System will show result attendance by block on screen.  System still in current “Report” page. |   **Alternative Scenario:** N/A  **Exceptions:** N/A  **Relationships:** Manage Report Attendance.  **Business Rules:**   * System will show attendance result in chosen conditions. * Default value when manager clicks on tab “Report” is: Class. | | | |

Table 28: <Staff> Report Attendance by Block

##### <Staff> Report Attendance by Semester

Use Case Diagram



Figure 33: <Staff> Report Attendance by Semester

Use Case Specification

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE – SU017** | | | |
| **Use Case No.** | SU017 | **Use Case Version** | 2.0 |
| **Use Case Name** | Report by Semester. | | |
| **Author** | Nguyen Quoc Huy. | | |
| **Date** | 19/09/2013 | **Priority** | Normal |
| **Actor:** Staff.  **Summary:**  This use case allows staff report attendance by semester.  **Goal:**  Staff can report attendance by semester.  **Triggers:**   * Staff can report information of student: Block, class, subject … * Staff must do these step to view report result:   + On Home page, click on link “Report” in menu bar and Report page will be showed.   + Then, in drop down list “Report by” choose type: Semester.   + Then, in drop down list “Name” choose the semester.   + Click on button “Statistic”.   **Preconditions:**   * User must logged in the system with the role is staff.   **Post Conditions:**   * **Success**: A report attendance by semester will be showed on screen. * **Fail:** System will not return any value. Transfer to error page or still in current page.   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | Click “Report” link in menu bar. | Redirect to Report Page, include 4 part:   * Drop down list “Report by” (value: Class, Student, Block, and Semester). * Drop down list “Name” (value: auto set based on drop down list Report by) * Button “Statistic” * Button “Export”. | | 2 | Choose “Semester” in drop down list Report by. | System will auto set value based on type Semester (just active semester) for drop down list “Name”. | | 3 | Chose the semester in drop down list “Name”. Click button “Statistic”. | System will show result attendance by semester on screen.  System still in current “Report” page. |   **Alternative Scenario:** N/A  **Exceptions:** N/A  **Relationships:** Manage Report Attendance.  **Business Rules:**   * System will show attendance result in chosen conditions. * Default value when manager clicks on tab “Report” is: Class. | | | |

Table 29: <Staff> Report Attendance by Semester

##### <Staff> Export Report

Use Case Diagram

**

Figure 34: <Staff> Export Report

Use Case Specification

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE – SU018** | | | |
| **Use Case No.** | SU018 | **Use Case Version** | 2.0 |
| **Use Case Name** | Export report. | | |
| **Author** | Nguyen Quoc Huy. | | |
| **Date** | 19/09/2013 | **Priority** | Normal |
| **Actor:** Staff.  **Summary:**  Staff uses this case to Export report.  **Goal:**  Allow staff to export report.  **Triggers:**   * Staff can export report. * On report page, click on button “Export” to export report.   **Preconditions:**   * User must logged in the system with the role is staff.   **Post Conditions:**   * **Success**: Staff can export report. The system still in current page. * **Fail:** N/A.   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | Click on “Export” to request export report on Report page | System will export report to file excel |   **Alternative Scenario:** N/A  **Exceptions:** N/A  **Relationships:** Manage Report Attendance.  **Business Rules:**   * Staff export report to file excel and attendance: class, block, student, semester … | | | |

#### < Instructor>Overview Use Case



Figure 35: <Instructor> Overview Use Case

##### <Instructor> Capture and Show result attendance



Figure 36: <Instructor> Capture and Show result attendance

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE – IU001** | | | |
| **Use Case No.** | IU001 | **Use Case Version** | 2.0 |
| **Use Case Name** | Capture and Show result attendance | | |
| **Author** | Nguyen Thanh Binh | | |
| **Date** | 23/09/2013 | **Priority** | High |
| **Actor:** Instructor.  **Summary:**  Instructor uses this case to check attendance.  **Goal:**  Allow instructor capture image, send image to web service and check attendance.  **Triggers:**   * Instructor can use mobile to take picture. This picture will be used for attendance checking ... To check attendance using picture, instructor must do: * Instructor login application and click button “Take Attendance” in Roll Call page. System transfer to “Taking Attendance” Page. * On “Taking Attendance” page, Instructor capture image, then send image to web service for detected face of member. * Instructor receive result from Web service. Instructor view result and check attendance for members.   **Preconditions:**   * User must logged in the system with the role is Instructor * Instructor is assign to class   **Post Conditions:**   * **Success**: Instructor can check attendance for student. The attendance log will be stored. * **Fail:** System will alert error.   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | Instructor open mobile app.  Instructor type username, password, press “Login” to login into application. | System will show the Roll call page with information includes:   * Instructor name: label * No: label * Subject: label * Class name: label * Class Time: Calendar * Class date: Calendar   **Main menu:**   * Detail: Button (follow each course) * Take Attendance: Button | | 2 | Instructor click button “Take Attendance” to capture picture.  [Alternative 1] | System transfer to Taking Attendance page with:   * Camera screen view: Frame layout * Capture: Button | | 3 | Instructor look at screen and click “Capture” button to capture image and send image to web service. | System will save image on real time when instructor click button “Capture”.  Image is sent to web service.  Then, system create show a popup “Sending success” with button “OK”.  [Exception 1,2] | | 4 | Instructor click “OK” button to return to Course page. | System will notify result to Instructor.  Result contains the students detected from image.  [Exception 3, 4, 5] |   **Alternative Scenario:**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | Instructor click button “Detail” to check attendance with use case “IU002”. | System will transfer to Roll Call Detail page. |   **Exceptions:**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | Error with camera (No camera, can’t connect to camera) | System will show popup “Camera is not available”. | | 2 | Can’t send image to service | System will show a popup “Failed to connect server, please check you internet connection and try it again”. Instructor click “OK” button to retake an image. | | 3 | No respond from service | System will show popup “Request timeout. Please try again”. | | 4 | Only some student detected (<30% number of class students) | System will vibrate and notify “Only X student detect. You can take more picture, or check attendance manually (See Use Case IU002)”. | | 5 | Stranger detected | System will vibrate and notify “Stranger Detected”. |   **Relationships:** N/A  **Business Rules:**   * Image : min 1500px \* 1000px max 3552px \* 2000px or size smaller than 3mb; resolution 72dpi (recommend) * Cable: 3mb ~ 6mb, Wifi: 3mb ~ 6mb * The average time for attendance taking is 10 seconds. * In success case, the app will notify normally, without sound or vibration. In exception case, the notification is with sound and vibration. | | | |

Table 30: <Instructor> Check Attendance by Taking Picture

##### <Instructor> Check Attendance Manually



Figure 37: <Instructor> Check Attendance Manually

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE – IU002** | | | |
| **Use Case No.** | IU002 | **Use Case Version** | 2.0 |
| **Use Case Name** | Check Attendance Manually | | |
| **Author** | Nguyen Thanh Binh | | |
| **Date** | 23/09/2013 | **Priority** | High |
| **Actor:** Instructor.  **Summary:**   * Instructor uses this case to check attendance manually.   **Goal:**   * Allow instructor check attendance again for some special reasons.   **Triggers:**   * Sometimes, the auto attendance checking is not accuracy, the instructor must check attendance manually. * To check attendance again manually, instructor must do: * Instructor login application and click button “Detail” in “Roll Call” page. System transfer to “Roll Call Detail” Page. * Instructor review all members and check attendance for member have special reason on this day or previous day.   **Preconditions:**   * User must logged in the system with the role is Instructor. * Instructor is assign to class.   **Post Conditions:**   * **Success**: Instructor can check attendance for member on this day or previous day. * **Fail:** System will alert error.   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | Instructor open mobile app.  Instructor type username, password, press “Login” to login into application. | System will show the Roll call page with information includes:   * Instructor name: label * No: label * Subject: label * Class name: label * Class Time: Calendar * Class date: Calendar   **Main menu:**   * Detail: Button (follow each course) * Take Attendance: Button | | 2 | Instructor click button “Detail” to view detail of that Subject.  [Alternative 1] | System will show Roll Call Detail:   * Roll call: label * Instructor Name: label * Class Name: label * Subject: label * Date & Time: Calendar * Student List: link show(No., Name, Code) * Attendance Log: * No: label * Date: Calendar * Present: Number * Log Detail: button | | 3 | Instructor click button “Log Detail” with correlative Date.  [Alternative 2] | System will show Log detail page on correlative with three tab:   * Roll Call info (Roll Call, Date, Subject, Class) * Log image * Attendance: * No: label * Student name: label * Present & Absent : Ratio button * Note: textbox * Present All & Absent All: Ration button * Submit Attendance: Button | | 4 | Click “Present” ratio button with correlative student.  [Alternative 3, 4, 5]  Write info on Note text box (recommend).  Click button “Submit Attendance” send request to check attendance for present member. | System will send request check attendance for present member to server.  System will show popup “Save changed!” with button “OK” | | 5 | Instructor click button “OK” redirect to Roll call page. |  |   **Alternative Scenario:**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | Instructor click button “Take Attendance” to check attendance with Use case “IU001”. | System will transfer to Taking Attendance page. | | 2 | Instructor click “Student list” link. | System will show popup all student with correlative class and course. | | 3 | Instructor click “Absent” ratio button with correlative student. | System will set absent for correlative student. | | 4 | Instructor click “Absent” ratio button with correlative student. | System will set present for all student. | | 5 | Instructor click “Absent All” ratio button with correlative student. | System will set absent for all student. |   **Exceptions:**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | Can’t connection to service | System will show a popup “Failed to connect server, please check you internet connection and try it again”. Instructor click “OK” button to transfer Roll Call page. | | 2 | No respond from service | System will show popup “Request timeout. Please try again”. |   **Relationships:** N/A  **Business Rules:**   * Instructor can check present member again for student have special reason. | | | |

Table 31: <Instructor> Check Attendance Manually

##### <Instructor> View Roll Call List



Figure 38: <Instructor> View Roll Call List

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE – IU003** | | | |
| **Use Case No.** | IU003 | **Use Case Version** | 2.0 |
| **Use Case Name** | View Roll Call List | | |
| **Author** | Nguyen Thanh Binh | | |
| **Date** | 23/09/2013 | **Priority** | Normal |
| **Actor:** Instructor.  **Summary:**  Instructor uses this case to view all roll call list.  **Goal:**  Allow instructor view roll call list with current day or previous day to check attendance again for member have special reason.  **Triggers:**   * Instructor login application and click button “Report” in Course page. System transfer to “Roll Call” Page. * Instructor review all members with course, class and selected day.   **Preconditions:**   * User must logged in the system with the role is Instructor. * Instructor is assign to class.   **Post Conditions:**   * **Success**: Instructor can view list of members on current day or previous day with selected course, class. * **Fail:** System will transfer to error page or still in current page.   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | Instructor open mobile app.  Instructor type username, password, press “Login” to login into application. | System will show the Roll call page with information includes:   * Instructor name: label * No: label * Subject: label * Class name: label * Class Time: Calendar * Class date: Calendar   **Main menu:**   * Detail: Button (follow each course) * Take Attendance: Button | | 2 | Instructor click button “Detail” to view detail of that Subject.  [Alternative 1] | System will show Roll Call Detail:   * Roll call: label * Instructor Name: label * Class Name: label * Subject: label * Date & Time: Calendar * Student List: link show(No., Name, Code) * Attendance Log: * No: label * Date: Calendar * Present: Number * Log Detail: button |   **Alternative Scenario:**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | Instructor click button “Take Attendance” to check attendance with Use case “IU001”. | System will transfer to Taking Attendance page. |   **Exceptions:**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | Can’t connection to service | System will show a popup “Failed to connect server, please check you internet connection and try it again”. Instructor click “OK” button to transfer Roll Call page. |   **Relationships:** Check Attendance Manually  **Business Rules:**   * Instructor can view all roll call list, help to review status of student and check attendance again for students have special reason. | | | |

Table 32: <Instructor> View Roll Call List

##### <Instructor> Report Attendance by Class



Figure 39: <Instructor> Report Attendance by Class

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE – IU004** | | | |
| **Use Case No.** | IU004 | **Use Case Version** | 2.0 |
| **Use Case Name** | Report Attendance by Class | | |
| **Author** | Nguyen Thanh Binh. | | |
| **Date** | 23/09/2013 | **Priority** | Normal |
| **Actor:** Instructor.  **Summary:**  This use case allows Instructor report attendance of all students by class.  **Goal:**  Instructor can report attendance of class.  **Triggers:**   * Instructor can report information of class: subject, student, attendance… * Instructor must do these step to view report result: * On Home page, click on link ”Report” in menu bar and Report page will be showed. * Then, in select box “Report by” choose type: Class. * Then, in drop down list “Name” chose the class name. * Click on button “Statistic”.   **Preconditions:**   * User must logged in the system with the role is Instructor.   **Post Conditions:**   * **Success**: A report attendance by class will be showed on screen. * **Fail:** System will not return any value. Transfer to error page or still in current page.   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | Click “Report” link in menu bar. | Redirect to Report Page, include 4 part:   * Drop down list “Report by” (value: Class, Student, Block, and Semester). * Drop down list “Name” (value: auto set based on drop down list Report by) * Button “Statistic” * Button “Export”. | | 2 | Choose “Class” in drop down list Report by. | System will auto set value based on type Class. | | 3 | Chose the class in drop down list “Name”. Click button “Statistic”. | System will show result attendance by class on screen.  System still in current “Report” page. |   **Alternative Scenario:** N/A  **Exceptions:** N/A  **Relationships:** Manage Report Attendance.  **Business Rules:**   * System will show attendance result in chosen conditions. * Default value when manager clicks on tab “Report” is: Class. | | | |

Table 33: <Instructor> Report Attendance by Class

#### <Student>Overview Use Case



Figure 40: <Student> Overview Use Case

##### <Student>Check Present Rate by Searched Course



Figure 41: <Student> CheckPresent Rate

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE – STU001** | | | |
| **Use Case No.** | STU001 | **Use Case Version** | 2.0 |
| **Use Case Name** | Check Present Rate by Searched Course | | |
| **Author** | Nguyen Thanh Binh | | |
| **Date** | 23/09/2013 | **Priority** | Normal |
| **Actor:** Student.  **Summary:**   * Student uses this case to search learned/learning course.   **Goal:**   * Allow student views learned/learning course info. The search condition is course name.   **Triggers:**   * Student can search course name on list course page. The searching results follow course name. * On list courses page, student choose search condition and clicks search button. System will show list search in list course table at current page. * On list search page, student choose the course. System will show information detail about that course.   **Preconditions:**   * User must logged in the system with the role is Student.   **Post Conditions:**   * **Success**: The search course will be found. System will show list search in lists course page. * **Fail:** System will transfer to error page or still in current page.   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | At lists course page, student choose search condition. | System will show the search condition include:  Course Name: text box | | 2 | After chose search condition, student clicks on search button (submit button) finish. | System will show list search of courses that student searched by condition in List Course table.  List Courses table includes 5 column: No (number), Course Name, Semester, Time, Present Rate. | | 3 | Ai list course table, student choose the course they want to view and click on that course name. | System will show information about that course in current page.  Information course includes: Roll call, Course Name, Semester, Time, Student Name & Code, Present, Present Rate and Detail (about exactly day present or absent) |   **Alternative Scenario:** N/A  **Exceptions:**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | Search not found | System will show “No course was found!” |   **Relationships:** Check Present Rate  **Business Rules:**   * The courses will be search by course name. * Student can use search course to view their course easily. | | | |

Table 34: <Student> CheckPresent Rate

##### <Student>Check Present Rate by Learned Course



|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE – STU002** | | | |
| **Use Case No.** | STU002 | **Use Case Version** | 2.0 |
| **Use Case Name** | Check Present Rate by Learned Course | | |
| **Author** | Nguyen Thanh Binh | | |
| **Date** | 23/09/2013 | **Priority** | Normal |
| **Actor:** Student.  **Summary:**   * Student uses this case to search learned course.   **Goal:**   * Allow student views learned course info. The search condition is semester.   **Triggers:**   * On list learned courses page, student choose search condition. System will show list search in list course table at current page. * On list search page, student choose the course. System will show information detail about that course.   **Preconditions:**   * User must logged in the system with the role is Student.   **Post Conditions:**   * **Success**: The search course will be found. System will show list search in lists course page. * **Fail:** System will transfer to error page or still in current page.   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | At lists course page, student choose search condition. | System will show the search condition include:  Semester: drop down list | | 2 | After chose search condition, student view list learned course will be show on current page. | System will show list search of courses that student searched by condition in List Course table.  List Courses table includes 5 column: No (number), Course Name, Semester, Time, Present Rate. | | 3 | Ai list course table, student choose the course they want to view and click on that course name. | System will show information about that course in current page.  Information course includes: Roll call, Course Name, Semester, Time, Student Name & Code, Present, Present Rate and Detail (about exactly day present or absent) |   **Alternative Scenario:** N/A  **Exceptions:** N/A  **Relationships:** Check Present Rate  **Business Rules:**   * The learned courses will be search by semester. * Student can use search course to view their course easily. | | | |

##### <Student>Check Present Rate by Learning Course



|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE – STU003** | | | |
| **Use Case No.** | STU003 | **Use Case Version** | 2.0 |
| **Use Case Name** | Check Present Rate by Learning Course | | |
| **Author** | Nguyen Thanh Binh | | |
| **Date** | 23/09/2013 | **Priority** | Normal |
| **Actor:** Student.  **Summary:**   * Student uses this case to search learning course.   **Goal:**   * Allow student views learning course info.   **Triggers:**   * On list learning courses page student choose the course. System will show information detail about that course.   **Preconditions:**   * User must logged in the system with the role is Student.   **Post Conditions:**   * **Success**: The search course will be found. System will show list search in lists course page. * **Fail:** System will transfer to error page or still in current page.   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | Student choose view Learning Course page. | System will show list learning courses. List Courses table includes 5 column: No (number), Course Name, Semester, Time, Present Rate. | | 2 | Ai list course table, student choose the course they want to view and click on that course name. | System will show information about that course in current page.  Information course includes: Roll call, Course Name, Semester, Time, Student Name & Code, Present, Present Rate and Detail (about exactly day present or absent) |   **Alternative Scenario:** N/A  **Exceptions:** N/A  **Relationships:** Check Present Rate  **Business Rules:**   * Student can use search course to view their course easily. | | | |

#### <Guest>Overview Use Case



Figure 42: <Guest> Overview Use Case

##### <Guest> Login

Use Case Diagram



Figure 43: <Guest> Login

Use Case Specification

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE – GU001** | | | |
| **Use Case No.** | GU001 | **Use Case Version** | 2.0 |
| **Use Case Name** | Login | | |
| **Author** | Pham Huy Hoang | | |
| **Date** | 19/09/2013 | **Priority** | Normal |
| **Actor:** Guest.  **Summary:**  Guest use this case to login into system.  **Goal:**  Allow authentication and authorization of the system.  **Triggers:**   * Guest want to login into system. * To login: Guest go to the login page, guest enter username and password, then click on “Log in” button to login.   **Preconditions:** N/A.  **Post Conditions:**   * **Success:** Guest is authorized, redirect to corresponding page. * **Fail:** System will show error on current page   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | Guest go to login page. | System show a login form, includes:   * Username: textbox (min length: 5, max length: 30, required). * Password: textbox (password box, min length:5, max length: 30, required). * Login: button. | | 1 | Guess enter username and password into textboxs.  Press “Log In” button. | Guest is logged into system, authorized with corresponding role. Redirect to corresponding page..  [Exception 1,2] |   **Alternative Scenario:** N/A  **Exceptions:**   |  |  |  | | --- | --- | --- | | No | Actor Action | System Response | | 1 | No input in “Username” or “Password” textboxs | Show error message: “Please enter username/password” below Username/Password textbox. | | 2 | Username/Password not in range [5,30] | Show error message: “Username/Password must be from 5 to 30 characters” textbox. | | 3 | Input invalid “Username” and “Password” | Show error message: “Invalid username or password”. |   **Relationships:** N/A  **Business Rules:**   * Each student, instructor, staff, admin has an account. * Only active account can log in. * Only instructor’s account can log in into mobile app. | | | |

Table 35: <Guest> Login

#### <Authorized User> Overview Use case

##### < Authorized User> Overview Use Case



Figure 44: < Authorized User> Overview Use Case

##### < Authorized User> Change Password



Figure 45: < Authorized User> Change Password

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE – UU001** | | | |
| **Use Case No.** | UU001 | **Use Case Version** | 2.0 |
| **Use Case Name** | Change Password | | |
| **Author** | Nguyen Thanh Binh | | |
| **Date** | 23/09/2013 | **Priority** | Normal |
| **Actor:** User (Instructor, Staff, Student, Admin).  **Summary:**   * User users this case to change their password.   **Goal:**   * Allow users can change their password.   **Triggers:**   * User use this use case to help improve their account security. * User login their account, click “Change Password” link. Go to “Change Password” page.   **Preconditions:**   * User must logged in the system with the role is User (Staff, Instructor, Student, Admin).   **Post Conditions:**   * **Success**: The password be changed. * **Fail:** System will transfer to error page or still in current page.   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | User selects the “Change Password” command in the top-right column of Home page. | System will show “Change Password” page includes:   * Username – Text box * Current Password – Text box * New Password – Text box * Confirm New Password – Text box * Submit – Button * Reset - Button | | 2 | - User enters the following information (Username, Current Password, New password, Confirm new password).  - Click “Submit” button to send request. | * System changes the user’s password. * System automatically re-authenticates the user with the new credentials. * System will show “Your password was changed successfully!” |   **Alternative Scenario:** N/A  **Exceptions:**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | System verifies that the Username and Current Password entries are correct and correspond to a current user. | System will show “Wrong username/ password” and reset all. | | 2 | New password same current password | System will show “New password was used! Please choose new password” and reset all. | | 3 | New password and confirm new password does not match. | System will show “Password does not match” and reset all. |   **Relationships:**  **Business Rules: N/A** | | | |

Table 36: < Authorized User> Change Password

##### < Authorized User>Logout



Figure 46: < Authorized User> Logout

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE – UU002** | | | |
| **Use Case No.** | UU002 | **Use Case Version** | 2.0 |
| **Use Case Name** | Change Password | | |
| **Author** | Nguyen Thanh Binh | | |
| **Date** | 23/09/2013 | **Priority** | Normal |
| **Actor:** User (Instructor, Staff, Student, Admin).  **Summary:**   * User use this use case to logout system.   **Goal:**   * The user selects the option to log off of the HCPT website and is then redirected back to the Login page.   **Triggers:**   * The use case starts when the user selects the option to logout. * The user selects the option to logout. * The system informs the user of logging out. * The user confirms logging out. * The system redirects user back to the Login page.   **Preconditions:**   * User must logged in the system with the role is User (Staff, Instructor, Student, Admin).   **Post Conditions:**   * **Success**: User logout system and redirect to login page. * **Fail:** System will transfer to error page or still in current page.   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | The user selects log out. | The system will show a pop-up: Do you really want to logout”   * Yes: button * No: button | | 2 | User select “Yes”.  [Alternative 1] | The system will clear user session and the user is logged out |   **Alternative Scenario:**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | User selects No. | User stay at current page. |   **Exceptions:** N/A  **Relationships:** N/A  **Business Rules:** N/A | | | |

Table 37: <User> Logout

## Software System Attribute

### Usability

#### Graphic User Interface

* All the text, label and image in staff, instructor and student page should be English.
* All the text, label and image in admin page should be English.

#### Usability

* Website admin, staff and instructor should need more than one week of training to be productive with the system.
* The student can use the system easily without training.

#### Installation

* The system must be easy to deploy. Customer can deploy successfully and learn to configure, maintain the system within one day of training.
* The mobile app must be easy to install. Compatible with almost Android Phone.
* The attached manual guide must be clear. User can read and do themselves without developer’s help.

### Reliability

* N/A

### Availability

* N/A

### Security

* Privacy: Each role of user has a specific permission to interact with system.
* System always checks authorization and authenticated before doing anything.
* Only admin can grant permission to staff.

### Maintainability

* N/A

### Portability

* N/A

### Performance

* Detect Faces From an Image: 3~4 seconds/image.
* Recognize Faces: 20~30 seconds/100 faces. With training set of 30 students, 600 images.
* The image upload speed depend on the speed of the network.

## ERD



Figure 47: ERD - Conceptual Diagram