**MINISTRY OF EDUCATION AND TRAINING**

**FPT UNIVERSITY**

Capstone Project Document

**Communication by Your Hands**

|  |  |
| --- | --- |
| **Group 04** | |
| **Group members** | Trương Công Thái – ThaiTCSE61209  Nguyễn Nhất Nguyên – NguyenNNSE61172  Phạm Hồng Quý – QuyPHSE61130  Nguyễn Duy Anh – AnhNDSE61130 |
| **Supervisor** | Mr. Kiều Trọng Khánh |
| **Ext. Supervisor** | N/A |
| **Capstone Project code** | CBYH |

-Ho Chi Minh City, **05/01/2016**-

*This page is intentionally left blank*

# Table of Contents

[Table of Contents 3](#_Toc442959676)

[List of Tables 4](#_Toc442959677)

[List of Figure 5](#_Toc442959678)

[A. Report No. 1 Introduction 8](#_Toc442959679)

[1. Project Information 8](#_Toc442959680)

[2. Introduction 8](#_Toc442959681)

[3. Current Situation 8](#_Toc442959682)

[4. Problem Definition 9](#_Toc442959683)

[5. Proposed Solution 10](#_Toc442959684)

[5.1 Feature functions 10](#_Toc442959685)

[5.2 Advantages and disadvantages 10](#_Toc442959686)

[6. Functional Requirements 11](#_Toc442959687)

[7. Role and Responsibility 12](#_Toc442959688)

[B. Report No.2 Software Project Management Plan 12](#_Toc442959689)

[1. Problem Definition 12](#_Toc442959690)

[1.1 Name of this Capstone Project 12](#_Toc442959691)

[1.2 Problem Abstract 12](#_Toc442959692)

[1.3 Project Overview 13](#_Toc442959693)

[2. Project organization 17](#_Toc442959694)

[2.1 Software Process Model 17](#_Toc442959695)

[2.2 Roles and responsibilities 19](#_Toc442959696)

[2.3 Tools and Techniques 21](#_Toc442959697)

[3. Project Management Plan 21](#_Toc442959698)

[3.1 Software development life cycle 21](#_Toc442959699)

[3.2 Phase Detail 23](#_Toc442959700)

[3.3 All Meeting Minutes 26](#_Toc442959707)

[4. Coding Convention 26](#_Toc442959708)

[C. Report No.3 Software Requirement Specification 26](#_Toc442959709)

[1. User Requirement Specification 26](#_Toc442959710)

[1.1 Guest Requirement 26](#_Toc442959711)

[1.2 Staff Requirement 27](#_Toc442959712)

[1.3 User Requirement 27](#_Toc442959713)

[1.4 Premium user Requirement 27](#_Toc442959714)

[1.5 Authenticated user Requirement 27](#_Toc442959715)

[1.6 Scheduler Requirement 27](#_Toc442959716)

[2. System Requirement Specification 28](#_Toc442959717)

[2.1 External Interface Requirement 28](#_Toc442959718)

[2.2 System Overview Use Case 28](#_Toc442959719)

[2.3 List of Use Case 29](#_Toc442959720)

[3. Software system attribute 82](#_Toc442959721)

[3.1 Usability 82](#_Toc442959722)

[3.2 Reliability 82](#_Toc442959723)

[3.3 Availability 82](#_Toc442959724)

[3.4 Security 82](#_Toc442959725)

[3.5 Mantainability 82](#_Toc442959726)

[3.6 Portability 82](#_Toc442959727)

[3.7 Performance 82](#_Toc442959728)

[4. Conceptual diagram 82](#_Toc442959729)

# List of Tables

[Table 1: Roles and Responsibilities 12](#_Toc442959636)

[Table 2: Hardware Requirement for Server 16](#_Toc442959637)

[Table 3: Hardware Requirement for Client 16](#_Toc442959638)

[Table 4: Software requirements for develop web site and web service 17](#_Toc442959639)

[Table 5: Software requirements for develop client application 17](#_Toc442959640)

[Table 6: Roles and Responsibilities Details 21](#_Toc442959641)

[Table 7: Tools and Techniques 21](#_Toc442959642)

[Table 8: Software Development Life Cycle Detail 23](#_Toc442959643)

[Table 9: Phase 1: Specification 24](#_Toc442959644)

[Table 10: Phase 2: Implementation 24](#_Toc442959645)

[Table 11: Phase 3: Validation 25](#_Toc442959646)

[Table 12: Register specification 32](#_Toc442959647)

[Table 13: Login specification 34](#_Toc442959648)

[Table 14: Search specification 35](#_Toc442959649)

[Table 15: Translate (Online) specification 38](#_Toc442959650)

[Table 16: Buy license specification 40](#_Toc442959651)

[Table 17: Translate (Offline) specification 43](#_Toc442959652)

[Table 18: Train custom specification 45](#_Toc442959653)

[Table 19: Share content specification 48](#_Toc442959654)

[Table 20: Add license type specification 51](#_Toc442959655)

[Table 21: Change license state specification 52](#_Toc442959656)

[Table 22: Edit license specification 54](#_Toc442959657)

[Table 23: Get license specification 55](#_Toc442959658)

[Table 24: Edit user information specification 57](#_Toc442959659)

[Table 25: Change user state specification 58](#_Toc442959660)

[Table 26: Get user information specification 60](#_Toc442959661)

[Table 27: Add new instruction specification 61](#_Toc442959662)

[Table 28: Change instruction state specification 63](#_Toc442959663)

[Table 29: Edit instruction specification 65](#_Toc442959664)

[Table 30: Get instruction specification 66](#_Toc442959665)

[Table 31: Add new library specification 68](#_Toc442959666)

[Table 32: Change library state specification 69](#_Toc442959667)

[Table 33: Edit library specification 71](#_Toc442959668)

[Table 34: Get library specification 72](#_Toc442959669)

[Table 35: Train sign language specification 75](#_Toc442959670)

[Table 36: Logout specification 76](#_Toc442959671)

[Table 37: Create notification specification 78](#_Toc442959672)

[Table 38: Send notification specification 80](#_Toc442959673)

[Table 39: Sync custom train data specification 82](#_Toc442959674)

[Table 40: Conceptual diagram data dictionary 83](#_Toc442959675)

# List of Figure

[Figure 1: Evolutionary development Model 18](#_Toc442959599)

[Figure 2: System Overview Use Case 29](#_Toc442959600)

[Figure 3: <Guest> Overview Use Case 30](#_Toc442959601)

[Figure 4: <Guest> Register 30](#_Toc442959602)

[Figure 5: <Guest> Login 32](#_Toc442959603)

[Figure 6: <Guest> Search 34](#_Toc442959604)

[Figure 7: <User> Oveview Use Case 35](#_Toc442959605)

[Figure 8: <User> Translate (Online) 36](#_Toc442959606)

[Figure 9: <User> Buy license 38](#_Toc442959607)

[Figure 10: <Premium User> Overview Use case 40](#_Toc442959608)

[Figure 11: <Premium User> Translate (Offline) 41](#_Toc442959609)

[Figure 12: <Premium User> Train Custom 43](#_Toc442959610)

[Figure 13: <Premium User> Share content 46](#_Toc442959611)

[Figure 14: <Staff> Overview Use Case 49](#_Toc442959612)

[Figure 15: <Staff> Add license type 49](#_Toc442959613)

[Figure 16: <Staff> Chage license state 51](#_Toc442959614)

[Figure 17: <Staff> Edit license 52](#_Toc442959615)

[Figure 18: <Staff> Get license 54](#_Toc442959616)

[Figure 19: <Staff> Edit user information 55](#_Toc442959617)

[Figure 20: <Staff> Change user state 57](#_Toc442959618)

[Figure 21: <Staff> Get user information 59](#_Toc442959619)

[Figure 22: <Staff> Add new instruction 60](#_Toc442959620)

[Figure 23: <Staff> Change instruction state 62](#_Toc442959621)

[Figure 24: <Staff> Edit instruction 63](#_Toc442959622)

[Figure 25: <Staff> Get instruction 65](#_Toc442959623)

[Figure 26: <Staff> Add new library 66](#_Toc442959624)

[Figure 27: <Staff> Change library state 68](#_Toc442959625)

[Figure 28: <Staff> Edit library 69](#_Toc442959626)

[Figure 29: <Staff> Get library 71](#_Toc442959627)

[Figure 30: <Staff> Train sign language 72](#_Toc442959628)

[Figure 31: <Authenticated user> Overview Use Case 75](#_Toc442959629)

[Figure 32: <Authenticated user> Logout 75](#_Toc442959630)

[Figure 33: <Scheduler> Overview Use Case 77](#_Toc442959631)

[Figure 34: <Scheduler> Create notification 77](#_Toc442959632)

[Figure 35: <Scheduler> Send notification 79](#_Toc442959633)

[Figure 36: <Scheduler> Sync custom train data 81](#_Toc442959634)

[Figure 37: Conceptual diagram 83](#_Toc442959635)

**Definitions, Acronyms, and Abbreviations**

|  |  |
| --- | --- |
| **Name** | **Definition** |
| CBYH | Communication by Your Hands |
| BOM | Back Office Management |
| EMG | Electromyography |
| SRS | Software Requirement Specification |

# Report No. 1 Introduction

## Project Information

* Project name: **Communication by Yours Hands**
* Project Code: **CBYH**
* Product Type: **Mobile application, BOM Website**
* Start Date: **05/01/2016**
* End Date: **<Ngày kết thúc>**

## Introduction

In communicating, sign language is the best way to communicate between people with deaf and mute. However, its required normal persons must have knowledge on sign language to communicate with the deaf and mute. Moreover, it is impossible for the deaf and mute to communicate with the blind. In this document, we introduce a solution for deaf and mute persons to communicate easily with the others without sign language knowledge on the normal person side.

We build a system, which help communication is easier between normal and deaf / mute persons. In the process of our research, we findout that MYO Gesture Control Armband is the key to solve the problem. By using MYO armband, we can read the electrical activity of person’s muscle and the motion of their arm then map the gesture with the customized data to translate sign language into text or sound with the same meaning. Beside that, we also provide an information system to manage easily the user, license packages and library packages.

This document also describes our working process in 4 months includes our perspective in the system, component designs and detailed core workflows. We all hope the system as so as our solution will help the deaf and mute persons easier to intergrate with the community.

## Current Situation

Currently, there is no official system that to support translate sign language into normal text or sound in VietNam or around the world. There is a project of Microsoft in China since 2013 named “Kinect Sign Language Translator” that use Kinect device to capture sign language movement to translate into spoken language and translate spoken language into sign language in real time. However, the project have not officially released yet.

So far, the most effective method for the deaf / mute persons to communicate is performing a combination of hands movement to describe a word or a phrase of words called “sign languge”.

Process of using signlanguage:

Case 1: (All participants have already known sign language)

- Deaf / mute persons perform sign language to description the message.

- The receivers understand the sign language base on their knowledge.

- The receivers perform sign language to reply the message.

Case2: (Not all participants have already known sign language)

There must be a translator

- Deaf / mute persons perform sign language to description the message.

- The translator translate the message from sign language into spoken language.

- The receivers get the message from the translator then reply the message to the translator.

- The translator translate the reply into sign language for the deaf / mute.

## Problem Definition

Below are advantages / disadvantages of the current situation:

\*Software support behavior:

-Advantages:

* Fully support communication: “Kinect Sign Language Translator” highly support deaf / mute persons to perform communicate with normal person and vice versa with delay is nearly zero.

-Disadvantages:

* Low mobility: To use “Kinect Sign Language Translator”, user has to come to a place where is set up the system.
* Just a concept: However, “Kinect Sign Language Translator” is just at project level, hasn’t officially released yet

\*User behavior:

-Advantages:

* Familiar and ease to use for deaf / mute person: It is the most familiar and effective for the deaf / mute to communicate in deaf / mute community.

-Disadvantages:

* Not use widly in community: In normal life, deaf / mute persons can hardly find someone who has knowledge on sign language to communicate.
* Normal turn into disabilities: It is quite hard for normal person who turn into deaf / mute accidentally to approach to sign language.
* Sign language is hard to learn with normal persons: It takes time and difficult to learn sign language.
* Communicate between the deaf / mute with the blind: Sign language is useless on helping in communication between the deaf / mute with the blind.

## Proposed Solution

Our proposed solution is to build a system named “CBYH”, which use a pair of MYO armbands and an internet connected mobile device to help deaf / mute persons to communicate with the others easier by translate sign language from users into normal text or sound with the same meaning. We also design the system to be scalable so we can deploy this system on multiple platforms in future plan.

CBYH system includes a web application and a mobile application with following functions:

### **Feature functions**

* + - Web application (for staff):
* Back Office Management: Staff can take manage on any user information, license and library database with this website.
  + - Sign language training (mobile app for staff):
* Training: Staff can add new sign language move and the meaning of it into database. Right after staff perform sign language, the application will receive raw data and the meaning of it, which is inputed by staff, then send and store them on server.
  + - Sign language translator (mobile app for user):
* Translate sign language: User can translate sign language into text or sound with the same meaning. Right after user perform sign language, the application will receive raw data from MYO armband and send to server then reveive translated data and outputs text or sound with the same meaning.

### Advantages and disadvantages

The advantages and disadvantages of the proposed solution:

* Advantages:
* The communication between the deaf / mute with the normal persons: It is easier for deaf and mute persons to express what they want to say to normal persons. There is no need sign language on the normal to understand what the deaf and mute want to say.
* Communicate between the deaf / mute with the blind: With the system, now the deaf and mute can communicate with the blind for the first time, which is impossible before.
* The deaf and mute in there job: It is easier for the deaf and mute in there job especially whose job relate with presentation.
* Mobility: To compare with “Kinect Sign language Translator” system, CBYH system’s user can translate sign language anywhere, anytime with a pair of MYO armbands and an internet connected mobile device.
* Disadvantages:
* The delay of translation: There is delay in translation sign language into normal text or sound.
* Lower accuracy: To compare with “Kinect Sign language Translator” system, the accuracy of sign language detection of CBYH system is quite lower.

## Functional Requirements

Function requirements of the system are listed as below:

* User component:
* Translate sign language (online mode): User can translate sign language into text or sound with internet connection required
* Buy license (for 30 days).
* Switch: After buy license, user has three more functions to switch for personal use.
* Translate sign language (offline mode): After buy license, user can download the meaning resource to device to translate sign language with no internet connection required.
* Train custom hand sign: After buy license, user can create new sign move and meaning of that sign for personal use.
* Share content: After buy license, user can create a group of device to receive translated data and stored in a list for personal use.
* Staff component:
* Train standard sign language: Staff can train new sign language and meaning for system.
* Manage database: Staff can take manage on database (User, license, library, sign language dictionary) with BOM website.

## Role and Responsibility

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| No | Full Name | Role | Position | Contact |
| 1 | Kiều Trọng Khánh | Project Manager | Supervisor | [khanhkt@*f*pt.edu.vn](mailto:khanhkt@fpt.edu.vn) |
| 2 | Trương Công Thái | Developer, Tester | Leader | [thaitcse61209@fpt.edu.vn](mailto:trungdqse60994@fpt.edu.vn) |
| 3 | Nguyễn Nhất Nguyên | Developer, Tester | Member | [nguyennnse61172@fpt.edu.vn](mailto:phucnhse60749@fpt.edu.vn) |
| 4 | Phạm Hồng Quý | Developer, Tester | Member | [quyphse61130@fpt.edu.vn](mailto:tripqmse60746@fpt.edu.vn) |
| 5 | Nguyễn Duy Anh | Developer, Tester | Member | [anhndse61077@fpt.edu.vn](mailto:khanc60351@fpt.edu.vn) |

Table : Roles and Responsibilities

# Report No.2 Software Project Management Plan

## Problem Definition

### Name of this Capstone Project

-Official name: Communication by Your Hands

-Vietnamese name Phiên dịch ngôn ngữ kí hiệu

-Abbreviation: CBYH

### Problem Abstract

In daily life, communication is the one of the most basic needs and tool of humanity. There are many ways for people to communicate such as speech, body language, gesture, feeling, wrtting, texting, etc. However, with the deaf / mute persons, that basic need is hardly satisfied. The most effective current solution for them is sign language – a combination of hands movement that describes a word, or a phrase of words, which they want to express. Nevertheless, that means it requires ervery participants must have knowledge on sign language or there must be someone as a translator. In addition, the sign language cannot help deaf / mute persons communicate with the blind. Because of those, this is not a comprehensive solution and it takes lots of time and cost for training and learning sign language. In additional, there are still some temporary solution to solve the problem such as writing and texting, but these ways require a lot of time and effort. Moreover, those ways are not working in some situations with a group of people such as in a classroom or a presentation, etc.

To solve those problems, which mentioned above, we provide a system that translate sign language into normal text or sound with the same meaning so that deaf / mute persons can use sign lanaguge to communicate with the others who have no knowledge on sign language without a person as a translator. The system includes a pair of MYO armbands, an internet connected mobile device and a web service. The system will plays the translator role to translate sign language into normal text or sound with the same meaning. We also provide a web page as a sign language dictionary to support persons who want to approach to sign language. In addition, we also provide an information system to manage user information, license, library and the sign language dictionary.

### Project Overview

#### Current Situation

Below are the problems encountered in this project:

* Disadvantages:
* High risk: Because the project uses MYO armband, team must study new technology/ API to apply it.
* Not familiar topic: Main topic of this project is highly related with sign language, which is quite hard to approach for all team members
* Advantages:
* Reveive good support from MYO development forum: Because of the development of MYO community, it is easier for team to get support from MYO forum when raise a problem.
* Receive good support from deaf / mute community of HoChiMinh city: Team got support about sign language knowledge from deaf / mute community, especially Ms. Lê Thị Thu Xương – instructor of HoChiMinh City Pedagogical University.

#### The Proposed System

According to the technology researches, MYO armband is the key to solve the current situation about helping deaf / mute persons in communication. We can use the feature of MYO armband to solve the problem about translating sign language. The basic idea is to use the MYO armbands to read user’s muscle electrical activity, which called “EMG” to translate user’s sign language move into text or sound with the same meaning.

To translate sign language, user must wear two MYO armbands, and then connect them via Bluetooth v4.0 with an Android device with internet connected and install our application. While user performing the sign language move, those armbands will read and send user’s raw EMG data to the Android device. Right after receive the data, the Android device will send those data to server via internet. Server will analyze raw EMG data then map with the meaning, which stored in database then return the meaning result to Android device in text. User can choose to display result as text or play sound depence on personal use.

User can also buy our license (for 30 days) to get more feature of the system. After buy our license, user can download library recource into device to translate sign language without internet connection. User can train new custom sign move for the system for personal use. Moreover, User can share in a group the meaning of translated sign language which performed by user for personal use.

##### Web Site

Website is a management tool for staff to take manage on user’s info, license, library and sign language dictionary. If login by user account, the website will be a sign lanaguge dictionary for user to research.

* For staff
* Manage user: Staff can take manage on user as view / edit info, change user state.
* Manage library: Staff can add new or change state of library package.
* Manage dictionary: Staff can add new, vew / edit and change state of records in sign language dictionary.
* Manage license: Staff can add new, view / edit info and change license state.
* For user
* Search sign language: User can search available sign language of the system by meaning. There will be a video to describe the word.
* Buy license: User can buy license to upgrade account to get more feature of the system.
* For guest
* Register: User can register new account to use the system.
* Search sign language: User can search available sign language of the system by meaning. There will be a video to describe the word.

Beside above, the website also provide API interfaces for mobile application to receive or update data from mobile application.

##### Mobile Applicaion

* For staff
* Training: Staff can train new sign language move and the meaning of it for the system.
* For user

This is the official application, which provide to user to collect and send raw data from MYO armands then send to server and receive analyzed data to do the following funtions:

* **Translate sign language (Online mode): User can translate sign language into text or sound with internet connection required.**

The following function only available if user have bought the license:

* Switch: user can switch between four modes (online, offline, train custom hand sign and share content).
* Translate sign language (Offline mode): User can download resource to device to translate sign language without an internet connection.
* Train custom hand sign: User can train new personal sign move and meaning of it for the system for personal use.
* Share content: User can share content of the translated data to a group of devices.

#### Boundaries of the System

* A User who wants to use the funtions of this system have to equip enough device includes:
* A Pair of MYO gesture control armband.
* A mobile device with our application installed and internet connected.
* To do the job, a staff of the system must be equipped the following devices:
* Computer system with internet connection.
* A pair of MYO gesture control armband.
* A mobile device with the training application installed and internet connected.

#### Future Plans

Currently, the system only deploy on a single platform: Android. Besides that, the system just support one side of the communication: from user of the system (the deaf / mute) to the others. We design the system to make it easily to scale to be a bigger model with more functions and run on more platform:

* Run on multiple platform on client side: Android, IOS, Window phone, Window, MacOS.
* Support multiple language library: Currently just support Vietnamese.
* Support translating two sides of a communication: From sign language in to text or sound and from spoken language into sign language.

#### Development Environment

##### Hardware requirements

**For Server**

|  |  |  |
| --- | --- | --- |
| **Windows** | **Minimum Requirements** | **Recommended** |
| **Internet Connection** | Cable, Wi-Fi (4 Mbps) | Cable, Wi-Fi (8 Mbps) |
| **Operating System** | Microsoft Window 7 | Microsoft Window 10 |
| **Computer Processor** | Intel® Xeon ® 1.4GHz | Intel® CORE i7 Quad core 2.4 GHz |
| **Computer Memory** | 1GB RAM | 2GB or more |

Table 2: Hardware Requirement for Server

**For Client**

|  |  |  |
| --- | --- | --- |
| **Android** | **Minimum Requirements** | **Recommended** |
| **Internet Connection** | Wi-Fi (4 Mbps) | Wi-Fi (8 Mbps) |
| **Operating System** | Android 4.4: Kitkat | Android 5.1.1: Lollipop |
| **Computer Processor** | Snapdragon 400 1.7GHz Dual Core | Snapdragon 800 2.3GHz Quad Core or higher |
| **Memory** | 512MB RAM | 2GB |
| **Bluetooth** | Bluetooth 4.0 required | Bluetooth 4.0 required |

Table 3: Hardware Requirement for Client

##### Software requirements

|  |  |  |
| --- | --- | --- |
| Software | Name / Version | Description |
| Operating system | Microsoft Window 7 | Operating system and platform for development |
| Environment | Java EE 7 | Specification for developing web application |
| Modeling tool | Star UML 5.0 | Used to implement website and web service |
| IDE | Netbeans 7.4 | Programming tools |
| DBMS | MS SQL Server 2008 | Used to create & manage the database for system |
| Source control | TortoiseSVN 1.8.11 | Used for source control |
| Web browser | Chrome 47 or above | Testing browser |

Table 4: Software requirements for develop web site and web service

|  |  |  |
| --- | --- | --- |
| Software | Name / Version | Description |
| Operating system | Android 4.4 Kitkat to 5.1.1: Lollipop | Operating system and platform for development |
| Environment | Java EE 5 | Specification for developing web application |
| IDE | Android Studio 1.5.1 | Programming tools |
| DBMS | SQLite 3 | Used to create & manage the database for system |
| Source control | TortoiseSVN 1.8.11 | Used for source control |
| Testing OS | Android 5.1.1: Lollipop | Testing Client Operation System |

Table 5: Software requirements for develop client application

## Project organization

### Software Process Model

#### Overall Description

Agile development are methods allow the development team to focus on the software itself rather on design and documentation. Agile methods universally rely on an incremental approach to software specification, development, and delivery. They are best suited to application development where the system requirements usually change rapidly during the development process.

Evolutionary development model is one the models of Agile method. Evolutionary development is an iterative and incremental approach to software development. Instead of creating a comprehensive antifact, such as a SRS, that is reviewed and accepted before creating a comprehensive design model (and so on) developer instead evolve the critical development antifact over time in an iterative manner. Instead of building and then delivering the system is a single time release, developers deliver it incrementally over times.

References:

- Software Engineering, 9/E -Ian Sommerville.

- <http://www.agiledata.org/essays/evolutionaryDevelopment.html>

#### Agile Development Method – Evolutionary Development Model

Figure 1: Evolutionary development Model

References:

Software Engineering, 8/E -Ian Sommerville.

#### Reasons for Choosing

The project is developed under Evolutionary Development Model to capable with current situation of our team. We chose this model because of the following reasons:

* The project use new technology – the MYO gesture armband, team do not sure about what the device can do so the solution and reality technology may mismatch.
* The team must study MYO API and implement the project at the same time.
* The team have to study sign language then map them into database of system.

For those reasons, the requirements of the project cannot be stable, clear, fix. Moreover, they can be rapidly changed.

### Roles and responsibilities

|  |  |  |  |
| --- | --- | --- | --- |
| No | Full name | Role in Group | Responsibilities |
| 1 | Mr. Kiều Trọng Khánh | Product Owner – Technical Expert | * Specify user requirement * Specifying the business * Control the development process * Give advices on techniques, solutions and business analysis support |
| 2 | Trương Công Thái | Team Leader, BA, DEV, Tester | * Managing process * Clarifying requirements * Researching solutions and techniques * Assigning task for members * Design architecture * Support team members * Reviewing the task result of members * Creating/ Editing documents and reports * Reviewing documents and reports * Coding Web service * Creating test plan * Creating test case * Testing |
| 3 | Nguyễn Nhát Nguyên | Team Member, BA, DEV, Tester | * Clarifying requirements * Researching solutions and techniques * Design architecture * Designing database * Reviewing documents and reports * Coding Web service * Reviewing test plan * Reviewing test case * Testing |
| 4 | Phạm Hồng Quý | Team Member, BA, DEV, Tester | * Clarifying requirements * Designing Mobile application UI * Reviewing documents and reports * Coding Mobile * Reviewing test plan * Reviewing test case * Testing Coding * Testing |

|  |  |  |  |
| --- | --- | --- | --- |
| 5 | Nguyễn Duy Anh | Team Member, BA, DEV, Tester | * Clarifying requirements * Designing BOM Website UI * Reviewing documents and reports * Coding BOM Web site * Reviewing test plan * Reviewing test case * Testing Coding * Testing |

Table 6: Roles and Responsibilities Details

### Tools and Techniques

|  |  |
| --- | --- |
| Tool / Technique | Name /version |
| Front-end IDE | Android Studio 1.5.1 |
| Back-end IDE | NetBean 7.4 |
| Front-end technology | HTML5, CSS, JavaScript, JQuery, Ajax, Android |
| Back-end technology | MVC, JavaEE, Servlet, JSP |
| Managing database | SQLite 3, MS SQL Server 2008 |
| Managing the project | SVN tortoise version 1.8.11 |
| Managing documents, reports, models and diagrams | Microsoft Office 2013 |

Table 7: Tools and Techniques

## Project Management Plan

### Software development life cycle

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Phase** | **Description** | **Deliverables** | **Resource needed** | **Dependencies and Constrains** | **Risks** |
| **Specification** | -Identify and define system spec in general | -Introduction of proposed system.  -General software requirement specification. | 20 man- days | N/A | * Lack of member share of understand * Lack of experience. |
| **Implementation** | -Design the current architecture  -Choose technology  - Code module | -Task plan  -Software design document  -Technology notes  - Actual software of each module | 60 man- days | Base on specification | * Lack of experience. * Code dose not work. |
| **Validation** | * Integrate modules of system * Release the version * Create test case * Test the version * Note changes. | * Actual software of the whole system * Testcase * Changes log / notes | 20 man days | Depend on software of each module | * Modules can’t connect with others * Test case doesn’t cover all core functions |

Table 8: Software Development Life Cycle Detail

If the result of current version in validation phrase is not satisfied, loop the process for the next version until result of the version is approved.

### Phase Detail

#### Specification

|  |  |  |
| --- | --- | --- |
| **Task** | **Description** | **Author** |
| **1. Identify and define system spec in general.** | Define which main functions system should provide. | ThaiTC, NguyenNN, QuyPH, AnhND |

Table 9: Phase 1: Specification

#### Implementation

|  |  |  |
| --- | --- | --- |
| **Task** | **Description** | **Author** |
| **1. Design the current architecture** | Design the architecture for the current system base on current definition of specification. | ThaiTC, NguyenNN, QuyPH, AnhND |
| **2. Choose technology** | Choose technology to implement the current system | ThaiTC, NguyenNN, QuyPH, AnhND |
| **3. Code modules** | Code modules base the designs and chosen technology | ThaiTC, NguyenNN, QuyPH, AnhND |

Table 10: Phase 2: Implementation

#### Validation

|  |  |  |
| --- | --- | --- |
| **Task** | **Description** | **Author** |
| **1. Integrate all modules of the system** | Integrate all separate modules | ThaiTC, NguyenNN, QuyPH, AnhND |
| **2. Release the version** | Release a version after intergrate all modules into a system | ThaiTC, NguyenNN, QuyPH, AnhND |
| **3. Create test case** | Create test case base current specification which was determinded in Specification phrase | ThaiTC, NguyenNN, QuyPH, AnhND |
| **4. Test the version** | Execute the created test case | ThaiTC, NguyenNN, QuyPH, AnhND |
| **5. Note changes** | Note the changes in changes log for the next version. | ThaiTC, NguyenNN, QuyPH, AnhND |

Table 11: Phase 3: Validation



### All Meeting Minutes

Place at folder “Meeting minute” in SVN with the following URL: [https://github.com/tcthai1994/communicateByYourHands/tree/master/Meeting%20minute](https://github.com/tcthai1994/communicateByYourHands/tree/master/Meeting%20minute%20)

## Coding Convention

General view of JAVA Programming Style put into practice in the project:

* Naming Conventions:

-Variable name should be short yet meaningful. If the name is more than one word, it must be in mixed case, starting word with a lowercase.

-Constants name should be all uppercase with words separated by underscores.

-Methods name should be verbs, in mixed case with the first word lowercase, the first letter of each internal word capitalized.

-Class name should be nouns, in mixed case with the first letter of each internal word capitalized.

* Package and import staments:

-Package statement is the first non-comment line.

-Import statement is after package statement.

* Constants

-Numerical constants should not be coded directly.

* Variable Assignments:

-Advoid assigning serveral variables to the same value in a single statement.

* Comments:

-Using /\* \*/ for block comments

-Using // for line comments

* Return Statements:

-A return statement with a value should not use parentheses.

References:

**Code Conventions for the Java TMProgramming Language**

Revised April 20, 1999

<http://www.oracle.com/technetwork/java/codeconvtoc-136057.html>

# Report No.3 Software Requirement Specification

## User Requirement Specification

### Guest Requirement

Guest is a person who has not accessed the system. Guest can use some functions of the system. To use fully functions, Guest has to login. These are some functions that guest can use:

* Register
* Login
* Search

### Staff Requirement

Staff is a role of system user. Person who has accessed the system with staff role can use the following functions:

* Train sign language
* Manage user
* Manage dictionary
* Manage license
* Manage library

### User Requirement

Person who login with registerd account can access the system with user role. These are functions that user can use:

* Search
* Translate (Online)
* Buy license

### Premium user Requirement

After bought license, user can upgrade account become “premium user”. Besides the functions that user can use, premium user can use more following functions:

* Translate (Offline)
* Share content
* Train custom

### Authenticated user Requirement

Authenticated user is the person who has accessed the system, besides the functions that users can use base on their role, authenticated user also can use the following function:

* Logout

### Scheduler Requirement

Scheduler is a part of the system. it runs automatically when the condition is met. These are the functions that scheduler can do:

* Sync custom train data
* Create notification
* Send notification

## System Requirement Specification

### External Interface Requirement

#### User Interface

#### Hardware Interface

* Android smartphone supports Bluetooth 4.0 low energy
* 2 MYO gesture control armband
* PC: 4GB ram, Intel® CORE i7 Quad core

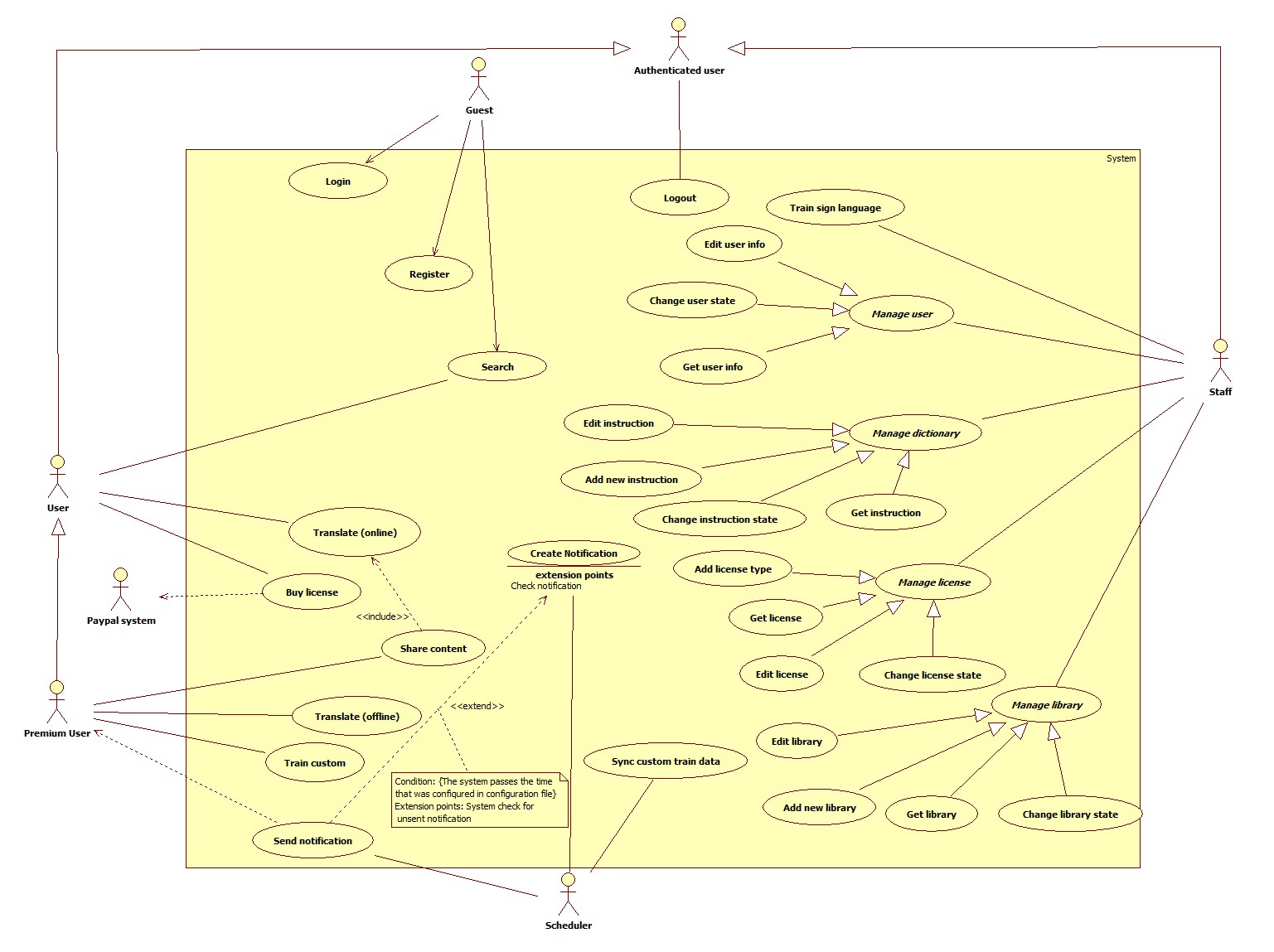
#### Software Interface

* Web application: work with Chrome (v47 or above), Internet Explorer (v10 or above), Firefox (v43 or above)
* Mobile application: Android operation system (v4.3 or above)
* Web server: Window operation system (window 7 or above)

#### Comunication Protocol

* Use HTTP protocol 1.1 for communication between the web browser and the web server
* Use HTTP protocol 1.1 for communication between the mobile application and the web server
* Use Bluetooth 4.0 low energy protocol for communication between the MYO armbands with the mobile application

### System Overview Use Case

Figure 2: System Overview Use Case

### List of Use Case

#### <Guest> Overview use case

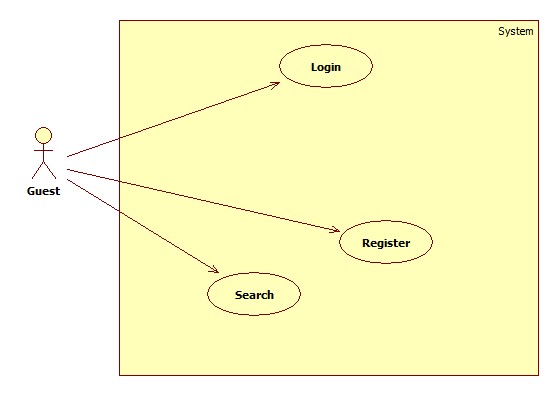


Figure 3: <Guest> Overview Use Case

##### <Guest> Register

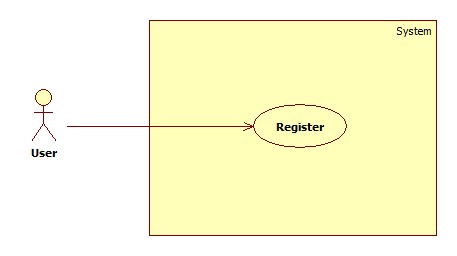


Figure 4: <Guest> Register

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE – UC001** | | | |
| **Use Case No.** | 001 | **Use Case Version** | 2.0 |
| **Use Case Name** | Register | | |
| **Author** | AnhND | | |
| **Date** | 20/01/2016 | **Priority** | Normal |
| **Actor:**   * Guest   **Summary:**   * This use case allows Guest to create new account   **Goal:**   * Account is registered successfully and store in database of the system   **Triggers:**   * Guest sends command to register   **Preconditions:**   * Actor has not accessed in the system   **Post Conditions:**   * **Success:** New account will be created * **Fail:** Show error message   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | Guest sends command to request register | System requires information from Guest:   * Email : free text input, required, (abc@abc.abc) * Username: free text input, required, length(9-20) * Password: free text input, required, length(6-12) * Repeat password: free text input, required, length(6-12) * Full name : free text input, required, length(10-50) * Phone : free number input, length(10-12)[0,9] | | 2 | Guest inputs information |  | | 3 | Guest sends command to register  [Alternative 1] | Guest will receive success message  [Alternative 2]  [Exception 1]  [Exception 2] |   **Alternative Scenario 1:**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | Guest sends command to reset | System reset all field to blank |   **Alternative Scenario 2:**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | Guest inputs invalid information | System show error message and request reenter |   **Exceptions 1:**   |  |  |  | | --- | --- | --- | | Step | Cause | System Response | | 1 | Guest input already exist username | System show error message and request reenter |   **Exceptions 2:**   |  |  |  | | --- | --- | --- | | Step | Cause | System Response | | 1 | Connection is lost | System show the message: “System is busy” when connection is lost. |   **Relationships:** N/A  **Business Rules:**   * Password is encrypted MD5 before being sent to server. * After registered, information of account will be stored in database of the system with role “User” and status is “active” | | | |

Table 12: Register specification

##### <Guest> Login

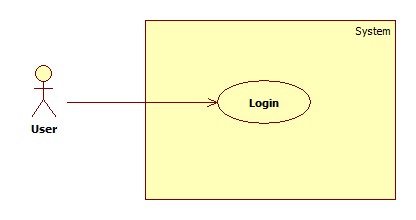


Figure 5: <Guest> Login

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE – UC002** | | | |
| **Use Case No.** | 002 | **Use Case Version** | 2.0 |
| **Use Case Name** | Login | | |
| **Author** | AnhND | | |
| **Date** | 20/01/2016 | **Priority** | Normal |
| **Actor:**   * Guest   **Summary:**   * This use case allows Guest login to the system   **Goal:**   * Guest login successfully with the proper role   **Triggers:**   * Guest send the login command   **Preconditions:**   * Guest has an account   **Post Conditions:**   * **Success:** Guest login the system * **Fail:** N/A   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | Guest sends command to request login | System requires identity information form Guest:   * Username : free text input, required, length(9-20) * Password : free text input, required, length(6-12) | | 2 | Guest inputs information |  | | 3 | Guest sends command to login to system | Guests will login system with their specific role  [Alternative 1] |   **Alternative Scenario 1:**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | Guest enter wrong identity information | System shows error message. |   **Relationships:** N/A  **Business Rules:**  - Password is encrypted MD5 before being sent to server.  - After login to system, guest will be redirected to specific view based on their role on the system: staff or user.  o If role is “User”, the system will display to User view.  o If role is “Staff”, the system will display to Staff view. | | | |

Table 13: Login specification

##### <Guest> Search

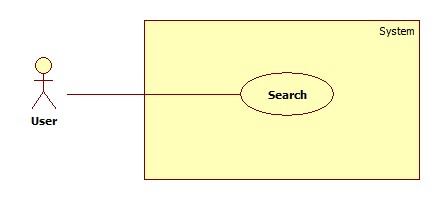


Figure 6: <Guest> Search

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE – UC003** | | | |
| **Use Case No.** | 003 | **Use Case Version** | 2.0 |
| **Use Case Name** | Search | | |
| **Author** | NguyenNN | | |
| **Date** | 24/01/2016 | **Priority** | Normal |
| **Actor:**   * User, Guest   **Summary:**   * This use case allows Actors to search sign language instruction   **Goal:**   * Actors can find available instruction sign language base on keyword   **Triggers:**   * Actors sends search command   **Preconditions:**   * N/A   **Post Conditions:**   * **Success:** records are shown * **Fail:** N/A   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | Actors input keyword in to search textbox |  | | 2 | Actors click Search button  [Alternative 1] | System will find in database any record of dictionary have keyword like input text and show that record. |   **Alternative Scenario:**   |  |  |  | | --- | --- | --- | | No. | Actor Action | System Response | | 1 | Staff input blank in textbox | System show all record |   **Relationships:** N/A  **Business Rules:**   * After get search command, the system will get the search value then looking for the right instruction base on instruction’s keyword then return the result to user | | | |

Table 14: Search specification

#### <User> Overview use case

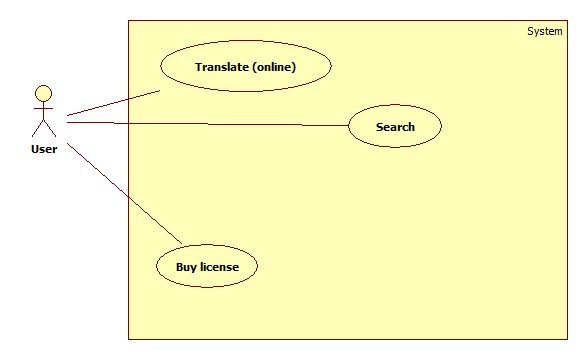


Figure 7: <User> Oveview Use Case

##### <User> Translate (online)

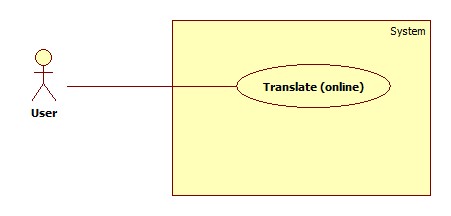


Figure 8: <User> Translate (Online)

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE – UC004** | | | |
| **Use Case No.** | 004 | **Use Case Version** | 2.0 |
| **Use Case Name** | Translate (online) | | |
| **Author** | QuyPH | | |
| **Date** | January 23th,2016 | **Priority** | Normal |
| **Actor:**   * User, Premium User   **Summary:**   * This use case allows user translate sign language into text or voice.   **Goal:**   * Proper text or voice match with the sign language will be shown to actor   **Triggers:**   * Staff sends translate command   **Preconditions:**   * Actor has accessed the system under user or premium user role.   **Post Conditions:**   * **Success:** Text or voice match with the sign language is shown. * **Fail:** Show error message.   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | User sends request to translate sign language | System requires connect to MYO armbands.  System requires sign language for detection | | 2 | User connects smartphone with 2 MYO armbands  [Alternative 1] |  | | 3 | User performs sign language | System shows a graph that describe the EMG data  [Exception 2]  [Exception 4] | | 4 | User performs end sign command | System returns proper text or voice.  [Exception 1]  [Exception 3]  [Alternative 2] |   **Alternative Scenario 1:**   |  |  |  | | --- | --- | --- | | Step | Cause | System Response | | 1 | Staff cannot connect the MYO armband with the smartphone | Nothing changes |   **Alternative Scenario 2:**   |  |  |  | | --- | --- | --- | | Step | Cause | System Response | | 1 | System read the sign language wrongly | System return mismatch text or voice |   **Exceptions 1:**   |  |  |  | | --- | --- | --- | | Step | Cause | System Response | | 1 | Connection is lost | System show message “System is busy” when connection is lost. |   **Exceptions 2:**   |  |  |  | | --- | --- | --- | | Step | Cause | System Response | | 1 | Bluetooth is disconnect | System show message “Bluetooth disconnect” when Bluetooth connection is lost. |   **Exceptions 3:**   |  |  |  | | --- | --- | --- | | Step | Cause | System Response | | 1 | Staff performs too fast | System shows the message “Save unsuccessfully” when the sign language move is too fast |   **Exceptions 4:**   |  |  |  | | --- | --- | --- | | Step | Cause | System Response | | 1 | System cannot get EMG data from the MYO armbands | The graph shows nothing |   **Relationships:** N/A  **Business Rules:**   * After connect the MYO armbands with the smartphone, the system will collect EMG data of user via Bluetooth until user perform the end sign command. After that, the system will send those data to server and looking for the proper result in database and return it to smart phone. * User can choose the return result in text or sound base on personal use | | | |

Table 15: Translate (Online) specification

##### <User> Buy license

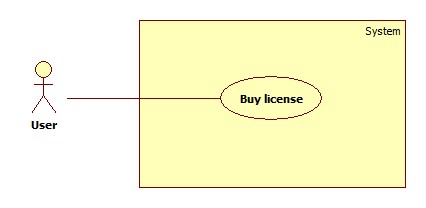


Figure 9: <User> Buy license

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE – UC005** | | | |
| **Use Case No.** | 005 | **Use Case Version** | 2.0 |
| **Use Case Name** | Buy License | | |
| **Author** | AnhND | | |
| **Date** | 23/01/2016 | **Priority** | Normal |
| **Actor:**   * User   **Summary:**   * This use case allows User buy license and upgrade to Premium User   **Goal:**   * User can buy license through Paypal payment   **Triggers:**   * User sends command to buy license   **Preconditions:**   * Actor has accessed the system under user role * Actor had Paypal account   **Post Conditions:**   * **Success:** User upgrade to Premium user * **Fail:** Show error message   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | User sends command to buy license | System requires confirmation to switch to Paypal payment | | 2 | User sends command to confirm  [Alternative 1] | System switch to Paypal payment process | | 3 | User completes payment process with Paypal  [Alternative 2] | System switch back and show successful message  [Alternative 3]  [Exception 1] |   **Alternative Scenario 1:**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | User sends command to decline | System stays still, stop requiring |   **Alternative Scenario 2:**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | Paypal responses payment process is fail | System switch back and show unsuccessful message |   **Alternative Scenario 3:**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | Paypal payment process time out | System switch back and show unsuccessful message |   **Exceptions 1:**   |  |  |  | | --- | --- | --- | | Step | Cause | System Response | | 1 | The connection is lost | System show the message: “System is busy” when connection is lost. |   **Relationships: N/A**  **Business Rules:**   * User must have Paypal account and enough money to buy the package * System will wait for response form Paypal payment process (maximum five minutes) to confirm the result to user * After five minute if the payment process isn’t completed, the system will show unsuccessful message and the staff will work with Paypal and confirm to user manually * After purchased license, user will be upgraded to “Premium User” | | | |

Table 16: Buy license specification

##### <User> Search

Referance 2.3.1.3 <Guest> Search

#### <Premium User> Overview use case

Figure 10: <Premium User> Overview Use case

##### <Premium User> Translate (offline)

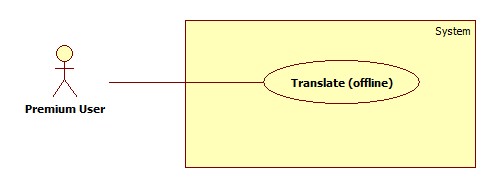


Figure 11: <Premium User> Translate (Offline)

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE – UC006** | | | |
| **Use Case No.** | 006 | **Use Case Version** | 2.0 |
| **Use Case Name** | Translate (offline) | | |
| **Author** | QuyPH | | |
| **Date** | January 23th,2016 | **Priority** | Normal |
| **Actor:**   * Premium User   **Summary:**   * This use case allows user translate sign language into text or voice without internet connection   **Goal:**   * Proper text or voice match with the sign language will be shown to actor   **Triggers:**   * Staff sends translate command   **Preconditions:**   * Actor has accessed the system under premium user role. * Premium user had download the library to device   **Post Conditions:**   * **Success:** Text or voice match with the sign language is shown without internet connection * **Fail:** Show error message.   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | User sends request to translate sign language | System requires connect to MYO armbands.  System requires sign language for detection | | 2 | User connects smartphone with 2 MYO armbands  [Alternative 1] |  | | 3 | User performs sign language | System shows a graph that describe the EMG data  [Exception 1]  [Exception 3] | | 4 | User performs end sign command | System returns proper text or voice.  [Exception 2]  [Alternative 2] |   **Alternative Scenario 1:**   |  |  |  | | --- | --- | --- | | Step | Cause | System Response | | 1 | Staff cannot connect the MYO armband with the smartphone | Nothing changes |   **Alternative Scenario 2:**   |  |  |  | | --- | --- | --- | | Step | Cause | System Response | | 1 | System read the sign language wrongly | System return mismatch text or voice |   **Exceptions 1:**   |  |  |  | | --- | --- | --- | | Step | Cause | System Response | | 1 | Bluetooth is disconnect | System show message “Bluetooth disconnect” when Bluetooth connection is lost. |   **Exceptions 2:**   |  |  |  | | --- | --- | --- | | Step | Cause | System Response | | 1 | Staff performs too fast | System shows the message “Save unsuccessfully” when the sign language move is too fast |   **Exceptions 3:**   |  |  |  | | --- | --- | --- | | Step | Cause | System Response | | 1 | System cannot get EMG data from the MYO armbands | The graph shows nothing |   **Relationships:** N/A  **Business Rules:**   * After connect the MYO armbands with the smartphone, the system will collect EMG data of user via Bluetooth until user perform the end sign command. After that, the system will looking for the proper result in local database in smartphone then return it to premium user | | | |

Table 17: Translate (Offline) specification

##### <Premium User> Train custom

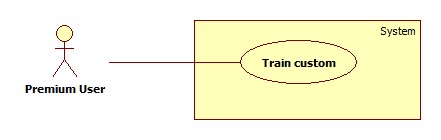
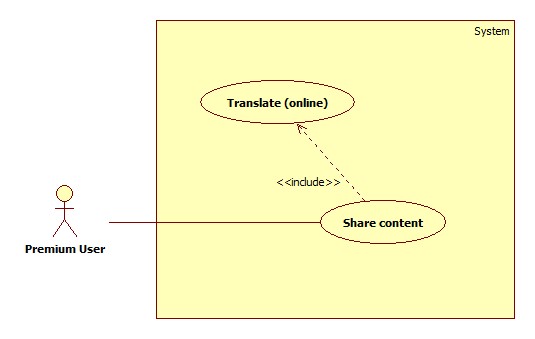


Figure 12: <Premium User> Train Custom

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE – UC007** | | | |
| **Use Case No.** | 007 | **Use Case Version** | 2.0 |
| **Use Case Name** | Train custom | | |
| **Author** | ThaiTC | | |
| **Date** | 23/01/2016 | **Priority** | Normal |
| **Actor:**   * Premium user   **Summary:**   * This use case helps premium user train custom sign for the system   **Goal:**   * Custom sign and its meaning are saved successfully   **Triggers:**   * Premium user sends save command   **Preconditions:**   * Actor has accessed the system under premium user role   **Post Conditions:**   * **Success:** new sign language is trained * **Fail:** Show error message   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | Premium user sends request to train custom sign | System requires connect to MYO armbands.  System requires information of new custom sign:   * Data of sign language move * Meaning of sign language: free text input, required, length (1 - 50) | | 2 | Premium user connects smartphone with 2 MYO armbands  [Alternative 1] |  | | 3 | Premium user inputs meaning of new custom sign |  | | 4 | Premium user performs custom sign | System shows a graph that describe the EMG data  [Exception 1]  [Exception 3] | | 5 | Premium user sends save command | System shows the message “Save successfully”  [Alternative 2]  [Exception 2] |   **Alternative Scenario 1:**   |  |  |  | | --- | --- | --- | | Step | Cause | System Response | | 1 | Premium user cannot connect the MYO armband with the smartphone | Nothing changes |   **Alternative Scenario 2:**   |  |  |  | | --- | --- | --- | | Step | cause | System Response | | 1 | Premium user inputs invalid information | System shows message “Your inputs are invalid” |   **Exceptions 1:**   |  |  |  | | --- | --- | --- | | Step | Cause | System Response | | 1 | Bluetooth connection is lost | System shows the message “Bluetooth disconnected” when the Bluetooth connection between the Android smartphone and the MYO armbands is lost |   **Exceptions 2:**   |  |  |  | | --- | --- | --- | | Step | Cause | System Response | | 1 | Premium user performs too fast | System shows the message “Save unsuccessfully” when the sign language move is too fast |   **Exceptions 3:**   |  |  |  | | --- | --- | --- | | Step | Cause | System Response | | 1 | System cannot get EMG data from the MYO armbands | The graph shows nothing |   **Relationships:** N/A  **Business Rules:**   * Premium user must have 2 MYO armbands * After connected the MYO armbands with the smartphone, the system can get premium user’s EMG data which describes the sign language via Bluetooth * After save, custom sign and meaning of it will be stored in the smartphone * If there is internet connection, the system will check for new custom sign data(EMG data and the meaning) then sync them to database of the system and remove data in the smartphone * After sync, the custom sign is ready for translate function immediately * While collecting EMG data via Bluetooth, the system draws a graph that describes the collected data | | | |

Table 18: Train custom specification

##### <Premium User> Share content

Figure 13: <Premium User> Share content

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE – UC008** | | | |
| **Use Case No.** | 008 | **Use Case Version** | 2.0 |
| **Use Case Name** | Share content | | |
| **Author** | ThaiTC | | |
| **Date** | 24/01/2016 | **Priority** | Normal |
| **Actor:**   * Premium User   **Summary:**   * This use case allows premium user to share a series of sign language translate result in text   **Goal:**   * Translated content are shared for selected device   **Triggers:**   * Premium user send share command   **Preconditions:**   * Actor has accessed the system under premium user role   **Post Conditions:**   * **Success:** Sign language of actor is translated and show on all selected device * **Fail:** Just some of selected device receive sharing data or show error message   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | Premium user sends command to request share content | System request information to detect receive account:   * Username: free text input, required | | 2 | Premium user input username of receive account |  | | 3 | Premium user send search command | System return account base on username that premium user searched  [Alternative 1]  [Exception 1] | | 4 | Premium user select account that want to share the content |  | | 5 | Premium send share command | System requires connect to MYO armbands.  System requires sign language for detection | | 6 | User connects smartphone with 2 MYO armbands  [Alternative 2] |  | | 7 | User performs sign language | System shows a graph that describe the EMG data  [Exception 2]  [Exception 4] | | 8 | User performs end sign command | System returns proper result in text in selected account’s smartphone  [Exception 1]  [Exception 3]  [Exception 5]  [Alternative 3] |   **Alternative Scenario 1:**   |  |  |  | | --- | --- | --- | | Step | Cause | System Response | | 1 | Premium user input nothing | Nothing happens |   **Alternative Scenario 2:**   |  |  |  | | --- | --- | --- | | Step | Cause | System Response | | 1 | Premium user cannot connect the MYO armband with the smartphone | Nothing changes |   **Alternative Scenario 3:**   |  |  |  | | --- | --- | --- | | Step | Cause | System Response | | 1 | System read the sign language wrongly | System return mismatch text or voice |   **Exceptions 1:**   |  |  |  | | --- | --- | --- | | Step | Cause | System Response | | 1 | Connection is lost | System show message “System is busy” when connection is lost. |   **Exceptions 2:**   |  |  |  | | --- | --- | --- | | Step | Cause | System Response | | 1 | Bluetooth is disconnect | System show message “Bluetooth disconnect” when Bluetooth connection is lost. |   **Exceptions 3:**   |  |  |  | | --- | --- | --- | | Step | Cause | System Response | | 1 | Staff performs too fast | System shows the message “Save unsuccessfully” when the sign language move is too fast |   **Exceptions 4:**   |  |  |  | | --- | --- | --- | | Step | Cause | System Response | | 1 | System cannot get EMG data from the MYO armbands | The graph shows nothing |   **Exceptions 5:**   |  |  |  | | --- | --- | --- | | Step | Cause | System Response | | 1 |  | Not all selected system receive sharing content |   **Relationships:** include translate (online)  **Business Rules:**   * System will collect accounts that premium user selected to share the translated content and create a room * The room just exist until the premium user sends command to stop sharing * The system will collect the EMG data from the MYO armbands via Bluetooth and send to server * After receive the EMG data, server will looking for the meaning of the sign language and return to the selected account’s smartphone in text | | | |

Table 19: Share content specification

#### <Staff> Overview use case

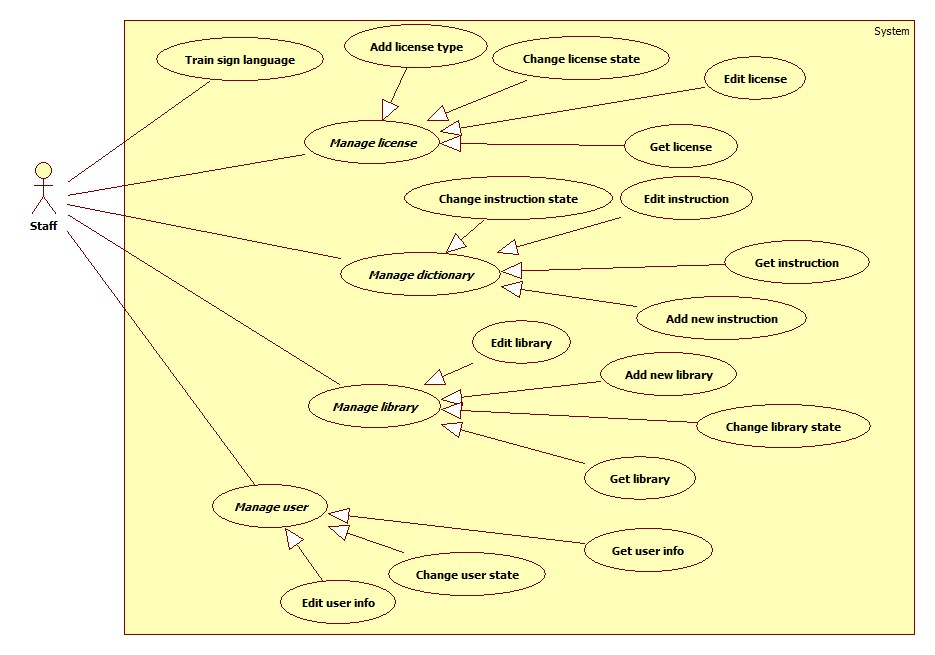
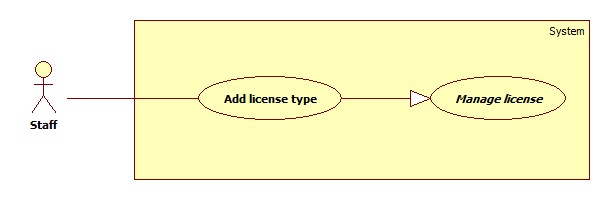


Figure 14: <Staff> Overview Use Case

##### <Staff> Add license type

Figure 15: <Staff> Add license type

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE – UC009** | | | |
| **Use Case No.** | 009 | **Use Case Version** | 2.0 |
| **Use Case Name** | Add license type | | |
| **Author** | ThaiTC | | |
| **Date** | 23/01/2016 | **Priority** | Normal |
| **Actor:**   * Staff   **Summary:**   * This use case helps staff add new license to the system   **Goal:**   * New license added successfully with status is “de-active”   **Triggers:**   * Staff sends add license command   **Preconditions:**   * Actor has already accessed under staff role   **Post Conditions:**   * **Success:** New license is added in system * **Fail:** Show error message   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | Staff sends request to add new license | System require information of new license from staff:   * License name: free text input, required, length(6 - 20) * Price: free text input, required, [0,50](unit: 1000VND) * Description: free text area input, required, length (6 - 250) | | 2 | Staff inserts new license information |  | | 3 | Staff sends add new license command  [Alternative 1] | New license is added to system  [Exception 1]  [Alternative 2] |   **Alternative Scenario 1:**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | Staff sends reset command | All input fields are cleared |   **Alternative Scenario 2:**   |  |  |  | | --- | --- | --- | | Step | Cause | System Response | | 1 | Staff inputs invalid information | System shows message “Your inputs are invalid” |   **Exceptions 1:**   |  |  |  | | --- | --- | --- | | Step | Cause | System Response | | 1 | Connection is lost | System show the message: “System is busy” when connection is lost. |   **Relationships:** N/A  **Business Rules:**   * After added, information of license will be stored in database of the system with status is “de-active”. When the staff wants to public the license to the system, staff will change license state into “active” manually. | | | |

Table 20: Add license type specification

##### <Staff> Change license state

Figure 16: <Staff> Chage license state

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE – UC010** | | | |
| **Use Case No.** | 010 | **Use Case Version** | 2.0 |
| **Use Case Name** | Change license state | | |
| **Author** | ThaiTC | | |
| **Date** | 23/01/2016 | **Priority** | Normal |
| **Actor:**   * Staff   **Summary:**   * This use case helps staff change state of license   **Goal:**   * Status of license is changed and updated in database of the system   **Triggers:**   * Staff sends change license state command   **Preconditions:**   * Actor has accessed the system under staff role * There must be at least one license in database of system   **Post Conditions:**   * **Success:** State of license is changed * **Fail:** Show error message or state cannot be changed   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | Staff sends change state command of the selected license | State of license is changed  [Exception 1]  [Exception 2] |   **Exceptions 1:**   |  |  |  | | --- | --- | --- | | Step | Cause | System Response | | 1 | Connection is lost | System show the message: “System is busy” when connection is lost. |   **Exceptions 2:**   |  |  |  | | --- | --- | --- | | Step | Cause | System Response | | 1 | Request cannot be sent | License still keep the old state |   **Relationships:** N/A  **Business Rules:**   * After change state, status of selected license will be updated in database immediately then return new state to staff. | | | |

Table 21: Change license state specification

##### <Staff> Edit license

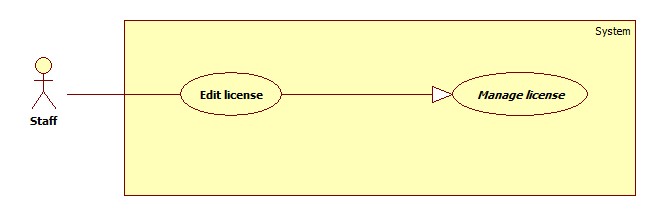


Figure 17: <Staff> Edit license

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE – UC011** | | | |
| **Use Case No.** | 011 | **Use Case Version** | 2.0 |
| **Use Case Name** | Edit license | | |
| **Author** | ThaiTC | | |
| **Date** | 23/01/2016 | **Priority** | Normal |
| **Actor:**   * Staff   **Summary:**   * This use case helps staff edit a license   **Goal:**   * New information of license will be update in the database of the system after being edited   **Triggers:**   * Staff sends save command   **Preconditions:**   * Actor has accessed the system under staff role * There must be at least one license in database of system   **Post Conditions:**   * **Success:** license information is edited * **Fail:** Show error message   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | Staff sends request to edit license | System show editable field contain information of the license:   * License name: free text input, required, length(6 -20) * Price: free text input, required, [0,50](unit: 1000VND) * Description: free text area input, required, length(6-250) | | 2 | Staff edit information of license |  | | 3 | Staff sends save command  [Alternative 1] | New license information is edited  [Exception 1]  [Alternative 2] |   **Alternative Scenario 1:**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | Staff sends cancel command | System shows previous page |   **Alternative Scenario 2:**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | Staff inputs invalid information | System shows message “Your inputs are invalid” |   **Exceptions:**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | Connection is lost | System show the message: “System is busy” when connection is lost. |   **Relationships:** N/A  **Business Rules:**   * After edit, new information of license will be updated in database of system then return new information to staff | | | |

Table 22: Edit license specification

##### <Staff> Get license

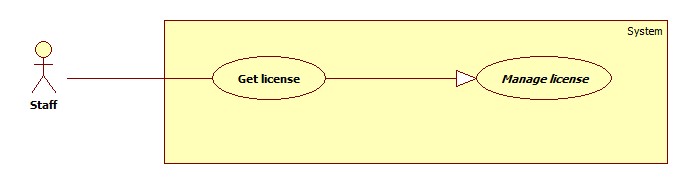


Figure 18: <Staff> Get license

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE – UC013** | | | |
| **Use Case No.** | 013 | **Use Case Version** | 2.0 |
| **Use Case Name** | Get license | | |
| **Author** | ThaiTC | | |
| **Date** | 23/01/2016 | **Priority** | Normal |
| **Actor:**   * Staff   **Summary:**   * This use case helps staff get information of a specific license   **Goal:**   * Information of the selected license will be fully shown to staff   **Triggers:**   * Staff sends get license command   **Preconditions:**   * Actor has accessed the system under staff role * There must be at least one license in database of system   **Post Conditions:**   * **Success:** information of license is get * **Fail:** Show error message   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | Staff sends request to get information of a license | System show detail information page of the selected license:   * License name * Price * Description * Status   [Exception 1] |   **Exceptions 1:**   |  |  |  | | --- | --- | --- | | Step | Cause | System Response | | 1 | Connection is lost | System show the message: “System is busy” when connection is lost. |   **Relationships:** N/A  **Business Rules:**   * After sends get license command, the system will collect fully information of selected license then return them to staff | | | |

Table 23: Get license specification

##### <Staff> Edit user information

Figure 19: <Staff> Edit user information

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE – UC014** | | | |
| **Use Case No.** | 014 | **Use Case Version** | 2.0 |
| **Use Case Name** | Edit User Information | | |
| **Author** | QuyPH | | |
| **Date** | January 23th,2016 | **Priority** | Normal |
| **Actor:**   * Staff   **Summary:**   * This use case allows staff edit information of user in system.   **Goal:**   * New information of user will be updated in the database of the system after being edited.   **Triggers:**   * Staff sends edit user command   **Preconditions:**   * Actor has accessed the system under staff role * There must be at least one user in database of system   **Post Conditions:**   * **Success:** User information is edited. * **Fail:** Show error message.   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | Staff sends request to edit user. | System show editable field contain information of user:   * Password, length(6,20) * Email, free text input, format(abc@abc.abc) * Fullname: free text input, length(6,20) * Phone: free text input, length(10,12) * LicenseType * ExpiredDate * Status. | | 2 | Staff edits user information at field staff wants to edit. | System verifies information of user has changed and then notify to staff.  [Alternative 2] | | 3 | Staff sends save command.  [Alternative 1] | New user information is edited  [Exception 1] |   **Alternative Scenario 1:**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | Staff sends cancel command | System shows previous page. |   **Alternative Scenario 2:**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | If staff enter invalid information | System shows message “Your inputs are invalid” |   **Exceptions 1:**   |  |  |  | | --- | --- | --- | | No | Cause | System Response | | 1 | Connection is lost | System show the message: “System is busy” when connection is lost. |   **Relationships:** N/A  **Business Rules:**   * After edited, new information of user will be updated in database of system then return new information to staff | | | |

Table 24: Edit user information specification

##### <Staff> Change user state

Figure 20: <Staff> Change user state

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE – UC015** | | | |
| **Use Case No.** | 015 | **Use Case Version** | 2.0 |
| **Use Case Name** | Change User State | | |
| **Author** | QuyPH | | |
| **Date** | January 23th,2016 | **Priority** | Normal |
| **Actor:**   * Staff   **Summary:**   * This use case allows staff change user state.   **Goal:**   * Status of user is changed and update in database of the system   **Triggers:**   * Staff sends change state user command   **Preconditions:**   * Actor has accessed the system under staff role * There must be at least one user in database of system   **Post Conditions:**   * **Success:** User state is changed. * **Fail:** Show error message or state cannot be changed**.**   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | Staff sends change state command of the selected user. | New status of selected user is edited  [Exception 1] |   **Exceptions:**   |  |  |  | | --- | --- | --- | | No | Cause | System Response | | 1 | Connection is lost | System show message “System is busy” when connection is lost. |   **Relationships:** N/A  **Business Rules:**   * After change state, status of selected user will be updated in database immediately then return new state to staff. | | | |

Table 25: Change user state specification

##### <Staff> Get user information

Figure 21: <Staff> Get user information

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE – UC016** | | | |
| **Use Case No.** | 016 | **Use Case Version** | 2.0 |
| **Use Case Name** | Get User Information | | |
| **Author** | QuyPH | | |
| **Date** | January 23th,2016 | **Priority** | Normal |
| **Actor:**   * Staff   **Summary:**   * This use case helps staff view information of a specific user.   **Goal:**   * Information of the selected user will be fully shown to staff.   **Triggers:**   * Staff sends get user information command   **Preconditions:**   * Actor has accessed the system under staff role * There must be at least one user in database of system   **Post Conditions:**   * **Success:** User information is get. * **Fail:** Show error message   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | Staff sends request to get information of a user | System show detail information page of the selected user:   * Password * Email * Fullname * Phone * LicenseType * ExpiredDate * Status.   [Exception 1] |   **Exceptions 1:**   |  |  |  | | --- | --- | --- | | No | Cause | System Response | | 1 | Connection is lost | System show the message: “System is busy” when connection is lost. |   **Relationships:** N/A  **Business Rules:**   * After sends get user information command, the system will collect fully information of selected user then return them to staff | | | |

Table 26: Get user information specification

##### <Staff> Add new instruction

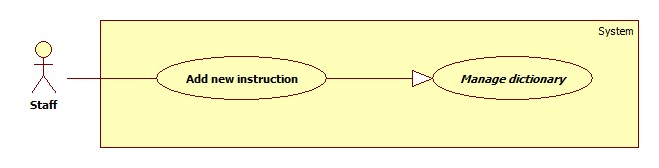


Figure 22: <Staff> Add new instruction

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE – UC017** | | | |
| **Use Case No.** | 017 | **Use Case Version** | 2.0 |
| **Use Case Name** | Add new instruction | | |
| **Author** | AnhND | | |
| **Date** | 23/01/2016 | **Priority** | Normal |
| **Actor:**   * Staff   **Summary:**   * This use case allows Staff to create new instruction   **Goal:**   * New instruction is created with status is “de-active”   **Triggers:**   * Staff sends command to add new instruction   **Preconditions:**   * Actor has accessed the system under staff role   **Post Conditions:**   * **Success:** New instruction will be created * **Fail:** Show error message   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | Staff send command to request add new instruction | System requires information:   * Key word: free text input, required, length(1-20) * Description: free text input, required, length(1-250) * Video URL: free text input, required, length(1-50) | | 2 | Staff input information |  | | 3 | Staff sends command to add new instruction request  [Alternative 1] | System will add new instruction to the system, show successful message  [Alternative 2]  [Exception 1] |   **Alternative Scenario 1:**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | Staff sends command to reset | System reset all field to blank |   **Alternative Scenario 2:**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | Staff inputs invalid information | System show error message and request to reenter |   **Exceptions 1:**   |  |  |  | | --- | --- | --- | | Step | Cause | System Response | | 1 | The connection is lost | System show the message: “System is busy” when connection is lost. |   **Relationships:** N/A  **Business Rules:**   * After added, information of instruction will be stored in database of the system with status is “de-active”. To public the instruction, staff has to change state of it manually | | | |

Table 27: Add new instruction specification

##### <Staff> Change instruction state

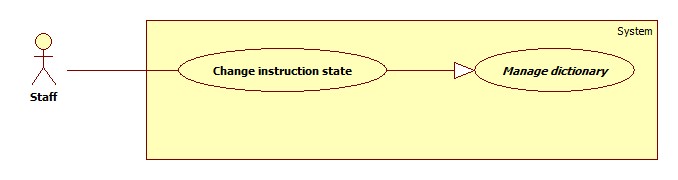


Figure 23: <Staff> Change instruction state

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE – UC018** | | | |
| **Use Case No.** | 018 | **Use Case Version** | 2.0 |
| **Use Case Name** | Add new instruction | | |
| **Author** | AnhND | | |
| **Date** | 23/01/2016 | **Priority** | Normal |
| **Actor:**   * Staff   **Summary:**   * This use case allows Staff to create new instruction   **Goal:**   * New instruction is created with status is “de-active”   **Triggers:**   * Staff sends command to add new instruction   **Preconditions:**   * Actor has accessed the system under staff role   **Post Conditions:**   * **Success:** New instruction will be created * **Fail:** Show error message   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | Staff send command to request add new instruction | System requires information:   * Key word: free text input, required, length(1-20) * Description: free text input, required, length(1-250) * Video URL: free text input, required, length(1-50) | | 2 | Staff input information |  | | 3 | Staff sends command to add new instruction request  [Alternative 1] | System will add new instruction to the system, show successful message  [Alternative 2]  [Exception 1] |   **Alternative Scenario 1:**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | Staff sends command to reset | System reset all field to blank |   **Alternative Scenario 2:**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | Staff inputs invalid information | System show error message and request to reenter |   **Exceptions 1:**   |  |  |  | | --- | --- | --- | | Step | Cause | System Response | | 1 | The connection is lost | System show the message: “System is busy” when connection is lost. |   **Relationships:** N/A  **Business Rules:**   * After added, information of instruction will be stored in database of the system with status is “de-active”. To public the instruction, staff has to change state of it manually | | | |

Table 28: Change instruction state specification

##### <Staff> Edit instruction

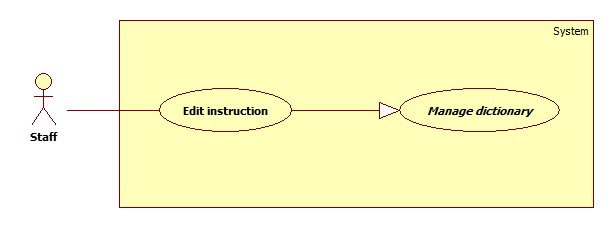


Figure 24: <Staff> Edit instruction

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE – UC019** | | | |
| **Use Case No.** | 019 | **Use Case Version** | 2.0 |
| **Use Case Name** | Edit Instruction | | |
| **Author** | AnhND | | |
| **Date** | 23/01/2016 | **Priority** | Normal |
| **Actor:**   * Staff   **Summary:**   * This use case allows Staff to edit instruction information   **Goal:**   * New information of instruction will be update in the database of the system after being edited   **Triggers:**   * Staff sends command to edit instruction information   **Preconditions:**   * There must be at least one instruction in database of system * Actor has accessed the system under staff role   **Post Conditions:**   * **Success:** Instruction information will be edited * **Fail:** Show error message   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | Staff sends command to request edit a specific instruction | System show editable field:   * Key word: free text input, required, length(1-20) * Description: free text input, required, length(1-250) * Video URL: free text input, required, length(1-50) | | 2 | Staff changes information |  | | 3 | Staff sends command to save change  [Alternative 1] | System change information and show successful message  [Alternative 2]  [Exception 1] |   **Alternative Scenario 1:**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | Staff sends cancel command | System shows previous page |   **Alternative Scenario 2:**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | Staff input wrong type of information | System show error message and request to reenter |   **Exceptions :**   |  |  |  | | --- | --- | --- | | Step | Cause | System Response | | 1 |  | System show the message: “System is busy” when connection is lost. |   **Relationships:** N/A  **Business Rules:**   * After edit, new information of instruction will be updated in database of system | | | |

Table 29: Edit instruction specification

##### <Staff> Get instruction

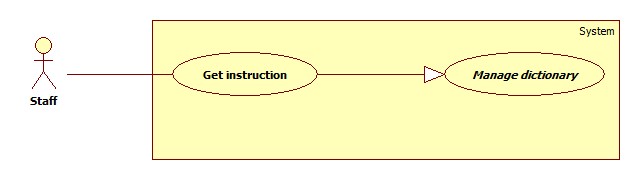


Figure 25: <Staff> Get instruction

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE – UC020** | | | |
| **Use Case No.** | 020 | **Use Case Version** | 2.0 |
| **Use Case Name** | Get Instruction | | |
| **Author** | AnhND | | |
| **Date** | 23/01/2016 | **Priority** | Normal |
| **Actor:**   * Staff   **Summary:**   * This use case allows Staff to view information of specific instruction   **Goal:**   * Information of the selected instruction will be fully shown to staff   **Triggers:**   * Staff sends command to get instruction detail   **Preconditions:**   * Actor has accessed the system under staff role   **Post Conditions:**   * **Success:** System will show instruction of available sign language * **Fail:** Show error message   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | Staff sends command to get instruction | System show available instructions | | 2 | Staff sends get command of a specific instruction | System show detail information page of the selected instruction:   * Key word * Description * Video URL   [Exception 1] |   **Exceptions :**   |  |  |  | | --- | --- | --- | | Step | Cause | System Response | | 1 |  | System show the message: “System is busy” when connection is lost. |   **Relationships:** N/A  **Business Rules:**   * After get instruction, staff can view all information which is stored in database of selected instruction. | | | |

Table 30: Get instruction specification

##### <Staff> Add new library

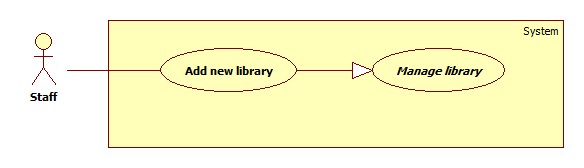


Figure 26: <Staff> Add new library

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE – UC021** | | | |
| **Use Case No.** | 021 | **Use Case Version** | 2.0 |
| **Use Case Name** | Add new library | | |
| **Author** | NguyenNN | | |
| **Date** | 24/01/2016 | **Priority** | Normal |
| **Actor:**   * Staff   **Summary:**   * This use case allows Staff to add new library   **Goal:**   * A new record of library is inserted to database successfully.   **Triggers:**   * Staff sends command to add new library   **Preconditions:**   * Actors has accessed the system under Staff role   **Post Conditions:**   * **Success:** New library will be created * **Fail:** Show error message   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | Staff sends command to add new library | System requires information:   * Name: free text input, required, length(3-50) | | 2 | Staff input information |  | | 3 | Staff sends command to add new library request | System will add new record of library to the database and show successful message  [Alternative 1]  [Exception 1] |   **Alternative Scenario 1:**   |  |  |  | | --- | --- | --- | | Step | cause | System Response | | 1 | Staff input wrong type of information | System shows error message and request to reenter |   **Exceptions 1:**   |  |  |  | | --- | --- | --- | | Step | Cause | System Response | | 1 | Connection is lost | System show the message: “System is busy” when connection is lost. |   **Relationships:** N/A  **Business Rules:**   * After added, information of library will be stored in database of the system with status is “de-active”. When the staff wants to public the library to the system, staff will change library state into “active” manually. | | | |

Table 31: Add new library specification

##### <Staff> Change library state

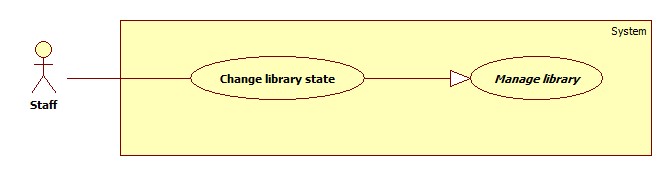


Figure 27: <Staff> Change library state

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE – UC022** | | | |
| **Use Case No.** | 022 | **Use Case Version** | 2.0 |
| **Use Case Name** | Change Library State | | |
| **Author** | NguyenNN | | |
| **Date** | 24/01/2016 | **Priority** | Normal |
| **Actor:**   * Staff   **Summary:**   * This use case allows Staff to change library state   **Goal:**   * Status of library is changed and updated in database of the system   **Triggers:**   * Staff sends command to change library state   **Preconditions:**   * There must be at least one instruction in database of system * Actor has accessed the system under staff role   **Post Conditions:**   * **Success:** State of library will be changed * **Fail:** Show error message   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | Staff sends command to change | System change status of library in DB and show success message  [Exception 1]  [Exception 2] |   **Exceptions 1:**   |  |  |  | | --- | --- | --- | | Step | Cause | System Response | | 1 | The connection is lost | System show the message: “System is busy” when connection is lost. |   **Exceptions 2:**   |  |  |  | | --- | --- | --- | | Step | Cause | System Response | | 1 | Request cannot be sent | Library still keep the old state |   **Relationships:** N/A  **Business Rules:**   * After change state, status of selected library will be updated in database immediately then return new state to staff. | | | |

Table 32: Change library state specification

##### <Staff> Edit library

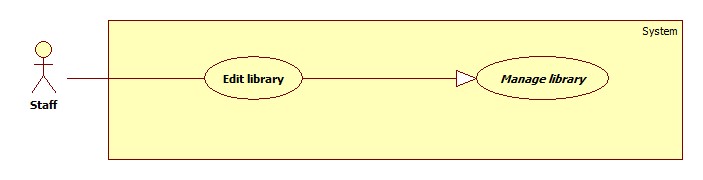


Figure 28: <Staff> Edit library

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE – UC023** | | | |
| **Use Case No.** | 023 | **Use Case Version** | 2.0 |
| **Use Case Name** | Edit Library | | |
| **Author** | NguyenNN | | |
| **Date** | 24/01/2016 | **Priority** | Normal |
| **Actor:**   * Staff   **Summary:**   * This use case allows Staff to edit library information   **Goal:**   * New information of library will be update in the database of the system after being edited   **Triggers:**   * Staff sends command to edit library information   **Preconditions:**   * Actor has accessed the system under staff role * There must be at least one library in database of system   **Post Conditions:**   * **Success:** library information will be edited * **Fail:** Show error message   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | Staff sends request to edit library information | System display fields to edit:   * Name: free text input, required, length(3 - 50) | | 2 | Staff changes information |  | | 3 | Staff sends command to save change | System change information in System’s DB has change and show successful message  [Exception 1]  [Alternative 1] |   **Alternative Scenario 1:**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | Staff input wrong type of information | System show error message and request to reenter |   **Exceptions 1:**   |  |  |  | | --- | --- | --- | | Step | Cause | System Response | | 1 | The connection is lost | System show the message: “System is busy” when connection is lost. |   **Relationships:** N/A  **Business Rules:**   * After edit, new information of library will be updated in database of system then return new information to staff | | | |

Table 33: Edit library specification

##### <Staff> Get library

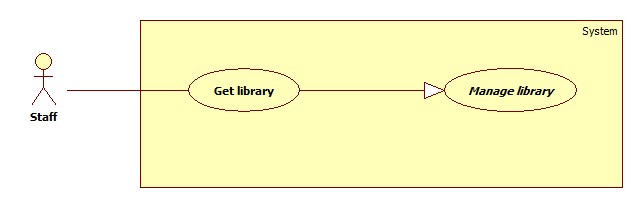


Figure 29: <Staff> Get library

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE – UC024** | | | |
| **Use Case No.** | 024 | **Use Case Version** | 2.0 |
| **Use Case Name** | Get Library | | |
| **Author** | NguyenNN | | |
| **Date** | 24/01/2016 | **Priority** | Normal |
| **Actor:**   * Staff   **Summary:**   * This use case helps staff get information of a specific library   **Goal:**   * Information include name and status and number of record belong to selected library will be shown for staff   **Triggers:**   * Staff sends command to view library detail   **Preconditions:**   * Actor has accessed the system under staff role   **Post Conditions:**   * **Success:** System will show detail of selected library * **Fail:** Show error message   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | Staff sends command to show library | System show available library | | 2 | Staff sends get command of a specific library | System display library information include:   * Name * Number of data record belong to selected library * Status   [Exception 1] |   **Exceptions 1:**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | Connection is lost | System show the message: “System is busy” when connection is lost. |   **Relationships:** N/A  **Business Rules:**   * After sends get license command, the system will collect fully information of selected library then return them to staff | | | |

Table 34: Get library specification

##### <Staff> Train sign language

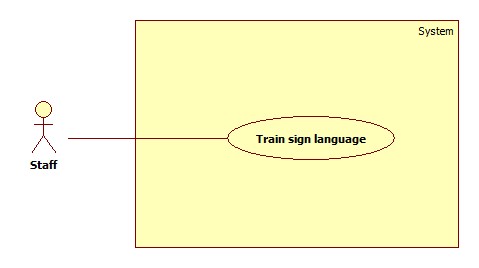


Figure 30: <Staff> Train sign language

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE – UC025** | | | |
| **Use Case No.** | 025 | **Use Case Version** | 2.0 |
| **Use Case Name** | Train sign language | | |
| **Author** | ThaiTC | | |
| **Date** | 23/01/2016 | **Priority** | Normal |
| **Actor:**   * Staff   **Summary:**   * This use case helps staff train sign language for the system   **Goal:**   * Sign language and its meaning are saved successfully   **Triggers:**   * Staff sends save command   **Preconditions:**   * Actor has accessed the system under staff role   **Post Conditions:**   * **Success:** new sign language is trained * **Fail:** Show error message   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | Staff sends request to train sign language | System requires connect to MYO armbands.  System requires information of new sign language:   * Data of sign language movement * Meaning of sign language: free text input, required, length (1 - 50) | | 2 | Staff connects smartphone with 2 MYO armbands  [Alternative 1] |  | | 3 | Staff inputs meaning of new sign language move |  | | 4 | Staff performs sign language | System shows a graph that describe the EMG data  [Exception 1]  [Exception 3] | | 5 | Staff sends save command | System shows the message “Save successfully”  [Alternative 2]  [Exception 2] |   **Alternative Scenario 1:**   |  |  |  | | --- | --- | --- | | Step | Cause | System Response | | 1 | Staff cannot connect the MYO armband with the smartphone | Nothing changes |   **Alternative Scenario 2:**   |  |  |  | | --- | --- | --- | | Step | Cause | System Response | | 1 | Staff input invalid information | System shows message “Your inputs are invalid” |   **Exceptions 1:**   |  |  |  | | --- | --- | --- | | Step | Cause | System Response | | 1 | Bluetooth connection is lost | System shows the message “Bluetooth disconnected” when the Bluetooth connection between the Android smartphone and the MYO armbands is lost |   **Exceptions 2:**   |  |  |  | | --- | --- | --- | | Step | Cause | System Response | | 1 | Staff performs too fast | System shows the message “Save unsuccessfully” when the sign language move is too fast |   **Exceptions 3:**   |  |  |  | | --- | --- | --- | | Step | Cause | System Response | | 1 | System cannot get EMG data from the MYO armbands | The graph shows nothing |   **Relationships:** N/A  **Business Rules:**   * Staff must clear about the sign language * The sign language must be based on the standard document * Staff must have 2 MYO armbands * After connected the MYO armbands with the smartphone, the system can get staff’s EMG data which describes the sign language via Bluetooth * After save, the system will store EMG data and meaning of it in database of the system and ready for translate function immediately * In case of lost connect while training, the data will be stored in device and waiting to be synced when then connection is ready * While collecting EMG data via Bluetooth, the system draws a graph that describes the collected data | | | |

Table 35: Train sign language specification

#### <Authenticated user> Overview use case

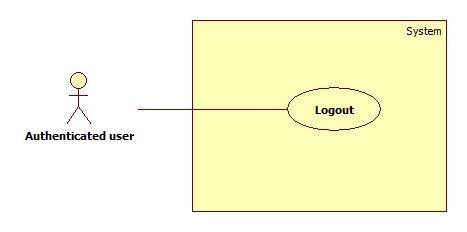


Figure 31: <Authenticated user> Overview Use Case

##### <Authenticated user> Logout

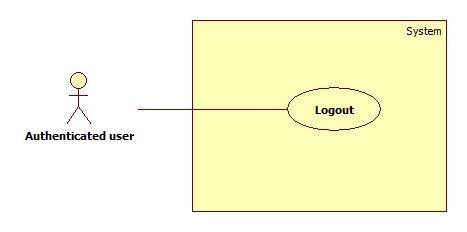


Figure 32: <Authenticated user> Logout

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE – UC026** | | | |
| **Use Case No.** | 026 | **Use Case Version** | 2.0 |
| **Use Case Name** | Logout | | |
| **Author** | AnhND | | |
| **Date** | 20/01/2016 | **Priority** | Normal |
| **Actor:**   * Authenticated user   **Summary:**   * This use case allows Authenticated user logouts the system   **Goal:**   * Authenticated user logouts the system successfully, the session is killed   **Triggers:**   * Authenticated User send request to logout   **Preconditions:**   * Actors has accessed the system   **Post Conditions:**   * **Success:** Authenticated user logouts successfully * **Fail:** N/A   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | Staff sends command to Logout | System return successful message |   **Relationships: N/A**  **Business Rules:**   * After logout, role “Authenticated User” will become “Guest” | | | |

Table 36: Logout specification

#### <Scheduler> Overview use case

Figure 33: <Scheduler> Overview Use Case

##### <Scheduler> Create notification

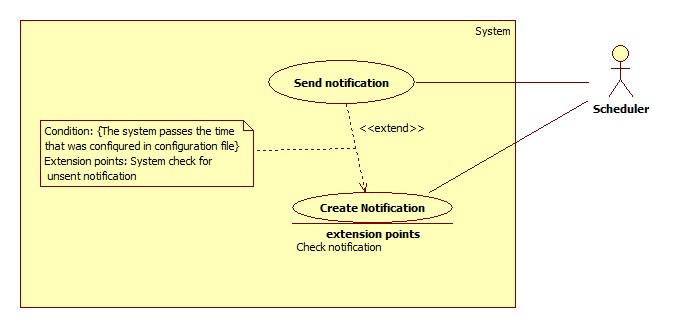
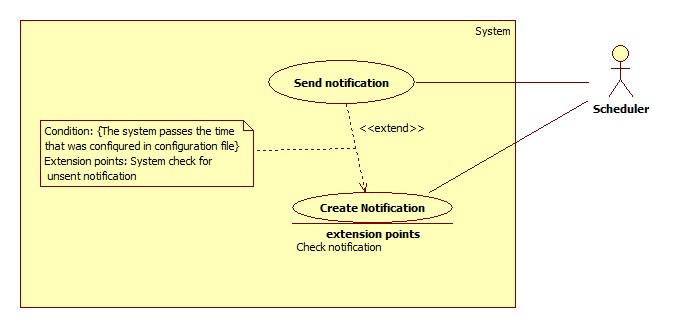


Figure 34: <Scheduler> Create notification

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE – UC027** | | | |
| **Use Case No.** | 027 | **Use Case Version** | 2.0 |
| **Use Case Name** | Create notification | | |
| **Author** | ThaiTC | | |
| **Date** | 24/01/2016 | **Priority** | Normal |
| **Actor:**   * Scheduler   **Summary:**   * This use case allows scheduler to create notification for Premium User   **Goal:**   * Scheduler creates successfully notification on time   **Triggers:**   * The system passes the time which is configured in configuration file   **Preconditions:**   * There must be a configuration file that set the time to create notification   **Post Conditions:**   * **Success:** Notification is created on time, log file generated * **Fail:** Nothing is created, log file is generated   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | The system check for the current time. If it passes the time that was set in configuration file the creating process will be started | System get current date then minus the license expiration date of premium user. If the result <= 5 then   * Create the notification * Generate the log file * Store notification in database with status is “unsent”   [Exception 1] |   **Exceptions 1:**   |  |  |  | | --- | --- | --- | | Step | Actor action | System Response | | 1 |  | Notification isn’t created but log file still being generated |   **Relationships:** N/A  **Business Rules:**   * Daily, at the time that was set in configuration file, the system will check for the license expiration date of premium user * There will be a flag to check the create notification process * When the system passes the time in configuration file, scheduler will check the flag. If the flag is off, the process will be started set the flag to on * When the process is finished, the flag will be set to off * Condition for creating notification * Current date – License expiration date <= 5 * If the condition is met, the notification will be created with the status is “unsent” * Log file is generate with the content is date and time that generate the notification and the id of notification, account of notification belongs to | | | |

Table 37: Create notification specification

##### <Scheduler> Send notification

Figure 35: <Scheduler> Send notification

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE – UC028** | | | |
| **Use Case No.** | 028 | **Use Case Version** | 2.0 |
| **Use Case Name** | Send notification | | |
| **Author** | ThaiTC | | |
| **Date** | 24/01/2016 | **Priority** | Normal |
| **Actor:**   * Scheduler   **Summary:**   * This use case allows scheduler to send notification to Premium User   **Goal:**   * Scheduler sends successfully notification on time   **Triggers:**   * The system passes the time which is configured in configuration file   **Preconditions:**   * There must be a configuration file that set the time to create notification   **Post Conditions:**   * **Success:** Notification is sent on time, status of notification is change to “sent” * **Fail:** Nothing is sent, status of notification is change to “sent”   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | The system check for the current time. If it passes the time that was set in configuration file the creating process will be started | System checks for notification storage in database of the system. If the status of the notification is “unsent”   * Send the notification to the account it belongs to * Change status of notification into “sent”   [Exception 1] |   **Exceptions 1:**   |  |  |  | | --- | --- | --- | | Step | Actor action | System Response | | 1 |  | Notification isn’t sent but status of notification is changed into “sent” |   **Relationships:** Extend from create notification  **Business Rules:**   * Daily, at the time that was set in configuration file, the system will check for the “unsent” notification in database of the system * There will be a flag to check the send notification process * When the system passes the time in configuration file, scheduler will check the flag. If the flag is off, the process will be started set the flag to on * When the process is finished, the flag will be set to off * Condition for sending notification * Status of notification is “unsent” * If the condition is met, the notification will be sent * After sent, the status of notification will be changed into “sent” | | | |

Table 38: Send notification specification

##### <Scheduler> Sync custom train data

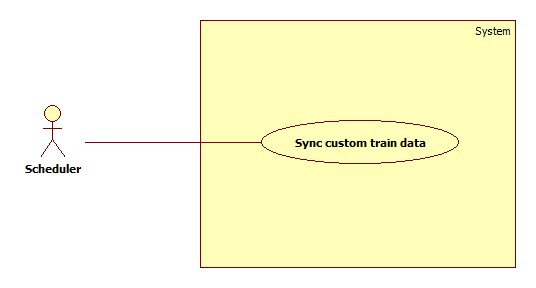


Figure 36: <Scheduler> Sync custom train data

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE – UC029** | | | |
| **Use Case No.** | 029 | **Use Case Version** | 2.0 |
| **Use Case Name** | Sync custom train data | | |
| **Author** | NguyenNN | | |
| **Date** | 24/01/2016 | **Priority** | Normal |
| **Actor:**   * Scheduler   **Summary:**   * This use case allows scheduler to sync custom train data to database   **Goal:**   * Custom train data are added to system   **Triggers:**  User go online  **Preconditions:**   * Custom data is in local device and not added to database   **Post Conditions:**   * **Success:** Custom train data will be synced into database of the system and the data in local device is deleted * **Fail:** Custom train data is synced successfully but the data in local device isn’t deleted   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | Scheduler check for new custom train data |  | | 2 | If there is new custom train data in local device, scheduler will sync the data into database of system and remove the local data  If there is no custom train data, scheduler does nothing  [Alternative 1] | [Exception 1] |   **Alternative Scenario 1:**   |  |  |  | | --- | --- | --- | | Step | cause | System Response | | 1 | The connection is lost | The function will stop sync and wait for next connection |   **Exceptions 1:**   |  |  |  | | --- | --- | --- | | Step | Actor action | System Response | | 1 |  | After sync process is complete, the scheduler doesn’t deleted the local data |   **Relationships:** N/A  **Business Rules:**   * When the device goes online, scheduler will check for new custom train data in local database * If there are any custom train data, scheduler will sync them into database of the system * After the sync process is complete, the data in local device will be deleted * The sync process will be stopped if the connection is lost and wait for the next connection, the local data isn’t deleted | | | |

Table 39: Sync custom train data specification

## Software system attribute

### Usability

### Reliability

### Availability

### Security

### Mantainability

### Portability

### Performance

## Conceptual diagram



Figure 37: Conceptual diagram

|  |  |
| --- | --- |
| Entity Data dictionary: describe all content of all entities | |
| Entity Name | Description |
| User | Abstract entity describes a user in system |
| Staff | Contain the staff information |
| Premium user | Contain the premium user information |
| Notification | Contain the notification infromation |
| License | Contain the license information |
| Instruction | Contain the instruction information |
| Library | Contain the library information |
| dataContent | Contain the dataContent information |
| customContent | Contain the customContent information |
| wordSignal | Contain the wordSignal information |
| customSignal | Contain the customSignal information |
| leftSignal | Contain the leftSignal information |
| rightSignal | Contain the rightSignal information |

Table 40: Conceptual diagram data dictionary