**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](#_Toc447884244)

[List of Tables 5](#_Toc447884245)

[List of Figure 7](#_Toc447884246)

[A. Report No. 1 Introduction 10](#_Toc447884247)

[1. Project Information 10](#_Toc447884248)

[2. Introduction 10](#_Toc447884249)

[3. Current Situation 10](#_Toc447884250)

[4. Problem Definition 11](#_Toc447884251)

[5. Proposed Solution 12](#_Toc447884252)

[5.1 Feature functions 12](#_Toc447884253)

[5.2 Advantages and disadvantages 12](#_Toc447884254)

[6. Functional Requirements 13](#_Toc447884255)

[7. Role and Responsibility 14](#_Toc447884256)

[B. Report No.2 Software Project Management Plan 14](#_Toc447884257)

[1. Problem Definition 14](#_Toc447884258)

[1.1 Name of this Capstone Project 14](#_Toc447884259)

[1.2 Problem Abstract 14](#_Toc447884260)

[1.3 Project Overview 15](#_Toc447884261)

[2. Project organization 19](#_Toc447884262)

[2.1 Software Process Model 19](#_Toc447884263)

[2.2 Roles and responsibilities 21](#_Toc447884264)

[2.3 Tools and Techniques 23](#_Toc447884265)

[3. Project Management Plan 23](#_Toc447884266)

[3.1 Software development life cycle 23](#_Toc447884267)

[3.2 Phase Detail 25](#_Toc447884268)

[3.3 All Meeting Minutes 28](#_Toc447884275)

[4. Coding Convention 29](#_Toc447884276)

[C. Report No.3 Software Requirement Specification 29](#_Toc447884277)

[1. User Requirement Specification 29](#_Toc447884278)

[1.1 Guest Requirement 29](#_Toc447884279)

[1.2 Staff Requirement 29](#_Toc447884280)

[1.3 User Requirement 30](#_Toc447884281)

[1.4 Premium user Requirement 30](#_Toc447884282)

[1.5 Authenticated user Requirement 30](#_Toc447884283)

[1.6 Scheduler Requirement 30](#_Toc447884284)

[2. System Requirement Specification 30](#_Toc447884285)

[2.1 External Interface Requirement 30](#_Toc447884286)

[2.2 System Overview Use Case 31](#_Toc447884287)

[2.3 List of Use Case 32](#_Toc447884288)

[3. Software system attribute 86](#_Toc447884289)

[3.1 Usability 86](#_Toc447884290)

[3.2 Reliability 86](#_Toc447884291)

[3.3 Availability 86](#_Toc447884292)

[3.4 Mantainability 86](#_Toc447884293)

[3.5 Portability 86](#_Toc447884294)

[3.6 Performance 86](#_Toc447884295)

[4. Conceptual diagram 87](#_Toc447884296)

[D. Report No.4 Software Design Description 88](#_Toc447884297)

[1. Design Overview 88](#_Toc447884298)

[2. System Architectural Design 89](#_Toc447884299)

[2.1 Web application architecture description 89](#_Toc447884300)

[2.2 Mobile application architecture description 90](#_Toc447884301)

[3. Component Diagram 90](#_Toc447884302)

[4. Detail Description 93](#_Toc447884303)

[4.1 Class Diagram 93](#_Toc447884304)

[4.2 Class Diagram Explanation 95](#_Toc447884305)

[4.3 Interactive Diagram 103](#_Toc447884306)

[5. Interface 119](#_Toc447884307)

[5.1 Component Interface 119](#_Toc447884308)

[5.2 Web Application Design 120](#_Toc447884309)

[5.3 Mobile Application Design 127](#_Toc447884310)

[6. Database design 136](#_Toc447884311)

[6.1 Entity relationship diagram 136](#_Toc447884312)

[6.2 Entity dictionary 138](#_Toc447884313)

[7. Algorithms 138](#_Toc447884314)

[7.1 Standardize EMG data (Android) 138](#_Toc447884315)

[7.2 Matching 139](#_Toc447884316)

[7.3 Detect 142](#_Toc447884317)

# List of Tables

[Table 1: Roles and Responsibilities 14](#_Toc447884318)

[Table 2: Hardware Requirement for Server 18](#_Toc447884319)

[Table 3: Hardware Requirement for Client (Mobile device) 18](#_Toc447884320)

[Table 4: Hardware Requirement for Client (MYO armband) 18](#_Toc447884321)

[Table 5: Software requirements for develop web site and web service 19](#_Toc447884322)

[Table 6: Software requirements for develop client application 19](#_Toc447884323)

[Table 7: Roles and Responsibilities Details 23](#_Toc447884324)

[Table 8: Tools and Techniques 23](#_Toc447884325)

[Table 9: Software Development Life Cycle Detail 25](#_Toc447884326)

[Table 10: Phase 1: Specification 26](#_Toc447884327)

[Table 11: Phase 2: Implementation 26](#_Toc447884328)

[Table 12: Phase 3: Validation 27](#_Toc447884329)

[Table 13: Register specification 35](#_Toc447884330)

[Table 14: Login specification 37](#_Toc447884331)

[Table 15: Search specification 38](#_Toc447884332)

[Table 16: Translate (Online) specification 41](#_Toc447884333)

[Table 17: Buy license specification 43](#_Toc447884334)

[Table 18: Translate (Offline) specification 46](#_Toc447884335)

[Table 19: Train custom sign specification 48](#_Toc447884336)

[Table 20: Share content specification 52](#_Toc447884337)

[Table 21: Add license type specification 54](#_Toc447884338)

[Table 22: Change license state specification 56](#_Toc447884339)

[Table 23: Edit license specification 57](#_Toc447884340)

[Table 24: Get license specification 59](#_Toc447884341)

[Table 25: Edit user information specification 60](#_Toc447884342)

[Table 26: Change user state specification 62](#_Toc447884343)

[Table 27: Get user information specification 63](#_Toc447884344)

[Table 28: Add new instruction specification 65](#_Toc447884345)

[Table 29: Change instruction state specification 66](#_Toc447884346)

[Table 30: Edit instruction specification 68](#_Toc447884347)

[Table 31: Get instruction specification 69](#_Toc447884348)

[Table 32: Add new library specification 71](#_Toc447884349)

[Table 33: Change library state specification 72](#_Toc447884350)

[Table 34: Edit library specification 74](#_Toc447884351)

[Table 35: Get library specification 75](#_Toc447884352)

[Table 36: Train sign language specification 78](#_Toc447884353)

[Table 37: Logout specification 80](#_Toc447884354)

[Table 38: Create notification specification 82](#_Toc447884355)

[Table 39: Send notification specification 84](#_Toc447884356)

[Table 40: Sync custom train data specification 86](#_Toc447884357)

[Table 41: Conceptual diagram data dictionary 88](#_Toc447884358)

[Table 42: Component Dictionary 93](#_Toc447884359)

[Table 43: Class dictionary 95](#_Toc447884360)

[Table 44: Account Attribute 96](#_Toc447884361)

[Table 45: Account Method 96](#_Toc447884362)

[Table 46: AccountDetail Attribute 96](#_Toc447884363)

[Table 47: AccountDetail Method 96](#_Toc447884364)

[Table 48: Dictionary Attribute 97](#_Toc447884365)

[Table 49: Dictionary Method 97](#_Toc447884366)

[Table 50: Notification Attribute 97](#_Toc447884367)

[Table 51: Notification Method 97](#_Toc447884368)

[Table 52: License Attribute 98](#_Toc447884369)

[Table 53: Notification Method 98](#_Toc447884370)

[Table 54: Library Attribute 98](#_Toc447884371)

[Table 55: Notification Method 98](#_Toc447884372)

[Table 56: Device Attribute 98](#_Toc447884373)

[Table 57: Device Method 99](#_Toc447884374)

[Table 58: LeftSignal Attribute 99](#_Toc447884375)

[Table 59: LeftSignal Method 99](#_Toc447884376)

[Table 60: RightSignal Attribute 99](#_Toc447884377)

[Table 61: RightSignal Method 99](#_Toc447884378)

[Table 62: MeaningLeft Attribute 100](#_Toc447884379)

[Table 63: MeaningLeft Method 100](#_Toc447884380)

[Table 64: MeaningRight Attribute 100](#_Toc447884381)

[Table 65: MeaningRight Method 100](#_Toc447884382)

[Table 66: WordSignal Attribute 100](#_Toc447884383)

[Table 67: WordSignal Method 101](#_Toc447884384)

[Table 68: DataContent Attribute 101](#_Toc447884385)

[Table 69: WordSignal Method 101](#_Toc447884386)

[Table 70: CustomSignal Attribute 101](#_Toc447884387)

[Table 71: CustomSignal Method 101](#_Toc447884388)

[Table 72: CustomContent Attribute 102](#_Toc447884389)

[Table 73: CustomContent Method 102](#_Toc447884390)

[Table 74: Recipt Attribute 102](#_Toc447884391)

[Table 75: Recipt Method 102](#_Toc447884392)

[Table 76: Web Services interface 120](#_Toc447884393)

[Table 77: Exception description 120](#_Toc447884394)

[Table 78: <Staff> Manage user fields 121](#_Toc447884395)

[Table 79: <Staff> Manage user buttons/ hyperlinks 121](#_Toc447884396)

[Table 80: <Staff> Update user fields 122](#_Toc447884397)

[Table 81: <Staff> Update user buttons/ hyperlinks 123](#_Toc447884398)

[Table 82: <Guest> Register fields 124](#_Toc447884399)

[Table 83: <Guest> Register buttons/ hyperlinks 124](#_Toc447884400)

[Table 84: <User> Buy license fields 125](#_Toc447884401)

[Table 85: <User> Buy license buttons/ hyperlinks 125](#_Toc447884402)

[Table 86: <User> Search instruction fields 126](#_Toc447884403)

[Table 87: <User> Search instruction buttons/ hyperlinks 126](#_Toc447884404)

[Table 88: <User> Edit user fields 127](#_Toc447884405)

[Table 89: <User> Edit user buttons/hyperlinks 127](#_Toc447884406)

[Table 90: <Staff> Train online fields 129](#_Toc447884407)

[Table 91: <Staff> Train online buttons/hyperlinks 129](#_Toc447884408)

[Table 92: <User> Login fields 131](#_Toc447884409)

[Table 93: <User> Login buttons/hyperlinls 131](#_Toc447884410)

[Table 94: <User> Translate fields 134](#_Toc447884411)

[Table 94: <User> Translate buttons/hyperlinks 134](#_Toc447884412)

[Table 96: <Premium user> Train offline fields 136](#_Toc447884413)

[Table 97: <Premium user> Train offline Buttons/Hyperlinks 136](#_Toc447884414)

[Table 98: Entity dictionary 138](#_Toc447884415)

# List of Figure

[Figure 1: Evolutionary development Model 20](#_Toc447885943)

[Figure 2: System Overview Use Case 32](#_Toc447885944)

[Figure 3: <Guest> Overview Use Case 33](#_Toc447885945)

[Figure 4: <Guest> Register 33](#_Toc447885946)

[Figure 5: <Guest> Login 35](#_Toc447885947)

[Figure 6: <Guest> Search 37](#_Toc447885948)

[Figure 7: <User> Oveview Use Case 38](#_Toc447885949)

[Figure 8: <User> Translate (Online) 39](#_Toc447885950)

[Figure 9: <User> Buy license 41](#_Toc447885951)

[Figure 10: <Premium User> Overview Use case 43](#_Toc447885952)

[Figure 11: <Premium User> Translate (Offline) 44](#_Toc447885953)

[Figure 12: <Premium User> Train custom sign 46](#_Toc447885954)

[Figure 13: <Premium User> Share content 49](#_Toc447885955)

[Figure 14: <Staff> Overview Use Case 52](#_Toc447885956)

[Figure 15: <Staff> Add license type 53](#_Toc447885957)

[Figure 16: <Staff> Chage license state 54](#_Toc447885958)

[Figure 17: <Staff> Edit license 56](#_Toc447885959)

[Figure 18: <Staff> Get license 58](#_Toc447885960)

[Figure 19: <Staff> Edit user information 59](#_Toc447885961)

[Figure 20: <Staff> Change user state 61](#_Toc447885962)

[Figure 21: <Staff> Get user information 62](#_Toc447885963)

[Figure 22: <Staff> Add new instruction 63](#_Toc447885964)

[Figure 23: <Staff> Change instruction state 65](#_Toc447885965)

[Figure 24: <Staff> Edit instruction 67](#_Toc447885966)

[Figure 25: <Staff> Get instruction 69](#_Toc447885967)

[Figure 26: <Staff> Add new library 70](#_Toc447885968)

[Figure 27: <Staff> Change library state 71](#_Toc447885969)

[Figure 28: <Staff> Edit library 73](#_Toc447885970)

[Figure 29: <Staff> Get library 74](#_Toc447885971)

[Figure 30: <Staff> Train sign language 76](#_Toc447885972)

[Figure 31: <Authenticated user> Overview Use Case 79](#_Toc447885973)

[Figure 32: <Authenticated user> Logout 79](#_Toc447885974)

[Figure 33: <Scheduler> Overview Use Case 80](#_Toc447885975)

[Figure 34: <Scheduler> Create notification 81](#_Toc447885976)

[Figure 35: <Scheduler> Send notification 83](#_Toc447885977)

[Figure 36: <Scheduler> Sync custom train data 85](#_Toc447885978)

[Figure 37: Conceptual diagram 87](#_Toc447885979)

[Figure 38: System architecture design 89](#_Toc447885980)

[Figure 39: Component Diagram (Server side) 91](#_Toc447885981)

[Figure 40: Component Diagram (Mobile side) 92](#_Toc447885982)

[Figure 41: Class Diagram 94](#_Toc447885983)

[Figure 42: Sequence diagram - <Web Server> Translate 104](#_Toc447885984)

[Figure 43: Sequence diagram - <Web Server> Train 106](#_Toc447885985)

[Figure 44: Sequence diagram - <Staff> Add dictionary 106](#_Toc447885986)

[Figure 45: Sequence diagram - <Staff> Update dictionary 107](#_Toc447885987)

[Figure 46: Sequence diagram - <Staff> Get dictionary 107](#_Toc447885988)

[Figure 47: Sequence diagram - <Staff> Add library 108](#_Toc447885989)

[Figure 48: Sequence diagram - <Staff> Update library 108](#_Toc447885990)

[Figure 49: Sequence diagram - <Staff> Get library 109](#_Toc447885991)

[Figure 50: Sequence diagram - <Staff> Update license 110](#_Toc447885992)

[Figure 51: Sequence diagram - <Staff> Update user 110](#_Toc447885993)

[Figure 52: Sequence diagram - <Staff> Get user information 111](#_Toc447885994)

[Figure 53: Sequence diagram - <User> Buy license 111](#_Toc447885995)

[Figure 54: Sequence diagram - <User> Update profile 112](#_Toc447885996)

[Figure 55: Sequence diagram - <User> Search 113](#_Toc447885997)

[Figure 56: Sequence diagram - <Guest> Register 114](#_Toc447885998)

[Figure 57: Activity diagram - <Staff> Train online 115](#_Toc447885999)

[Figure 58: Activity diagram - <User> Connect Myo armbands 116](#_Toc447886000)

[Figure 59: Activity diagram - <User> Translate online 117](#_Toc447886001)

[Figure 60: Activity diagram - <Premium User> Translate offline 118](#_Toc447886002)

[Figure 61: Activity diagram - <Premium User> Train offline 119](#_Toc447886003)

[Figure 62: Interface - <Staff> Manage user 121](#_Toc447886004)

[Figure 63: Interface - <Staff> Update user 122](#_Toc447886005)

[Figure 64: Interface - <Guest> Register 123](#_Toc447886006)

[Figure 65: Interface - <User> Buy license 124](#_Toc447886007)

[Figure 66: Interface - <User> Search instruction 125](#_Toc447886008)

[Figure 67: Interface - <User> Edit user profile 126](#_Toc447886009)

[Figure 68: Interface - <Staff> Train online 128](#_Toc447886010)

[Figure 69: Interface - <User> Login 130](#_Toc447886011)

[Figure 70: Interface - <User> Translate – Translating 132](#_Toc447886012)

[Figure 71: Interface - <User> Translate – Translated 133](#_Toc447886013)

[Figure 72: Interface - <Premium user> Train offline 135](#_Toc447886014)

[Figure 73: Entity relationship diagram 137](#_Toc447886015)

[Figure 74: Matching workflow 141](#_Toc447886016)

[Figure 75: Detect workflow 143](#_Toc447886017)

[Figure 76: Find meaning right workflow 144](#_Toc447886018)

[145](#_Toc447886019)

[Figure 77: Find meaning left workflow 145](#_Toc447886020)

[146](#_Toc447886021)

[Figure 78: Find meaning of each hand workflow 146](#_Toc447886022)

**Definitions, Acronyms, and Abbreviations**

|  |  |
| --- | --- |
| **Name** | **Definition** |
| CBYH | Communication by Your Hands |
| BOM | Back Office Management |
| EMG | Electromyography |
| SRS | Software Requirement Specification |
| GUI | Graphic User Interface |
| EMG data format | A list with 8 byte type elements |

# 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 1: 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, writing, 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 every 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 language 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 |
| **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 (Mobile device)

|  |  |
| --- | --- |
| **Sensors** | Medical Grade Stainless Steel EMG sensors, Highly sensitive nine-axis IMU containing three-axis gyroscope, three-axis accelerometer, three-axis magnetometer |
| **LEDs** | Dual Indicator LEDs |
| **Processor** | ARM Cortex M4 |
| **Haptic Feedback** | Short, Medium, Long Vibrations |

Table 4: Hardware Requirement for Client (MYO armband)

##### Software requirements

|  |  |  |
| --- | --- | --- |
| Software | Name / Version | Description |
| Operating system | Microsoft Window 7 | Operating system and platform for development |
| Environment | Java EE 5 | Specification for developing web application |
| Modeling tool | Star UML 5.0 | Used to implement website and web service |
| IDE | Netbeans 8.0.2 | 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 5: 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 |
| SDK | MYO Android SDK 0.10.00 | SDK of MYO armbands for Android that:  -Managing MYO devices  -Locking and Unlocking MYO  -Receiving Events  -Usage Data |
| Source control | TortoiseSVN 1.8.11 | Used for source control |
| Testing OS | Android 5.1.1: Lollipop | Testing Client Operation System |

Table 6: 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 7: Roles and Responsibilities Details

### Tools and Techniques

|  |  |
| --- | --- |
| Tool / Technique | Name /version |
| Front-end IDE | Android Studio 1.5.1 |
| Back-end IDE | NetBean 8.0.2 |
| 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 8: 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 9: 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 10: 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 11: 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 12: 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’s 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 registered account can access the system with user role. These are functions that user can use:

* Search
* Translate (Online)
* Buy license

### Premium user Requirement

After buy license, user can upgrade account become “premium user”. Beside those functions that user can use, premium user can use more following functions:

* Translate (Offline)
* Share content
* Train custom sign

### 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

* General requirement for GUI should be simple, clear, bright color, intuitive and reminiscent.

#### Hardware Interface

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

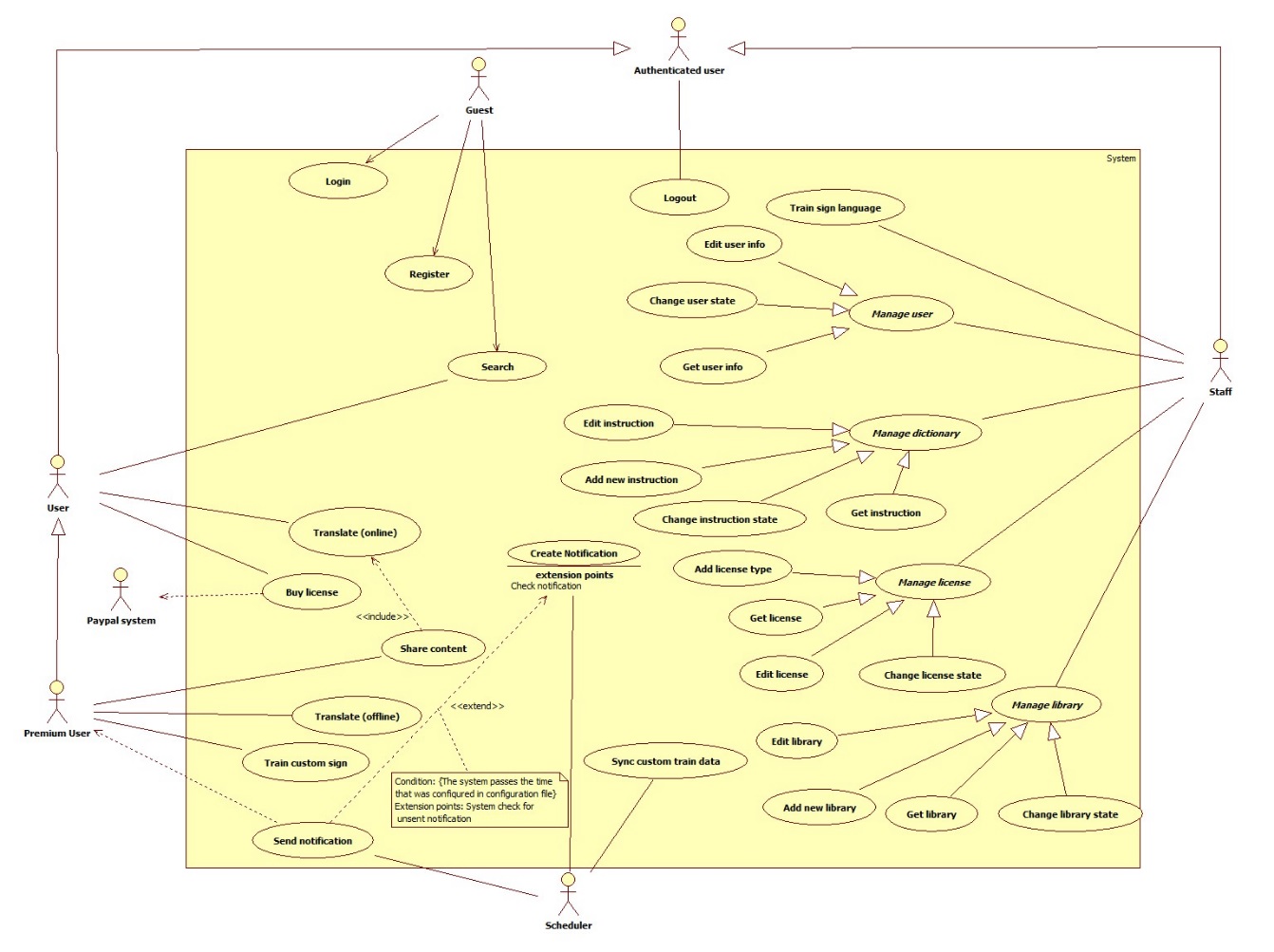
#### Software Interface

* 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

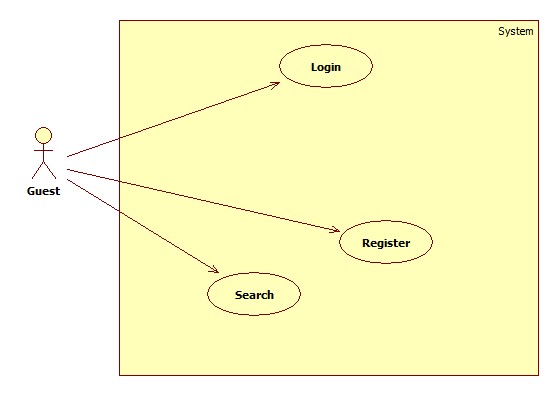


Figure 3: <Guest> Overview Use Case

##### <Guest> Register

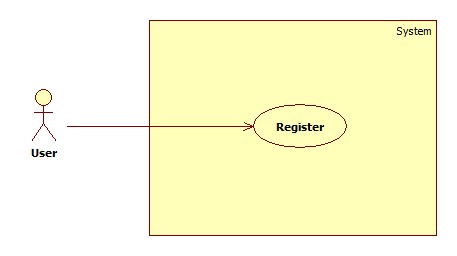


Figure 4: <Guest> Register

|  |  |  |  |
| --- | --- | --- | --- |
| **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 | Internet connection is lost | System shows message “You are offline, please check your connection”. |   **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 13: Register specification

##### <Guest> Login

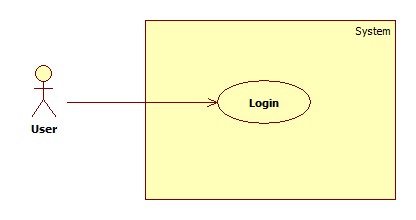


Figure 5: <Guest> Login

|  |  |  |  |
| --- | --- | --- | --- |
| **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 14: Login specification

##### <Guest> Search

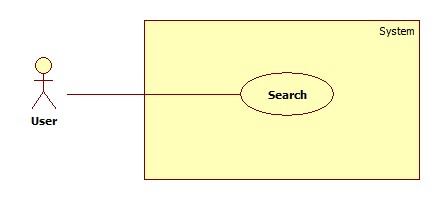


Figure 6: <Guest> Search

|  |  |  |  |
| --- | --- | --- | --- |
| **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 15: Search specification

#### <User> Overview use case

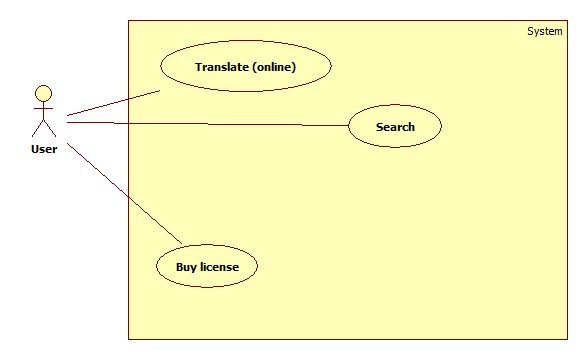


Figure 7: <User> Oveview Use Case

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

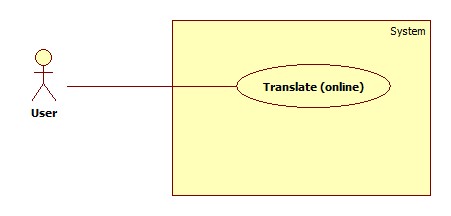


Figure 8: <User> Translate (Online)

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE – UC004** | | | |
| **Use Case No.** | 004 | **Use Case Version** | 2.0 |
| **Use Case Name** | Translate (online) | | |
| **Author** | QuyPH | | |
| **Date** | January 23th,2016 | **Priority** | High |
| **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 two 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 | System shows message “Fail to connect” and requires to connect again |   **Alternative Scenario 2:**   |  |  |  | | --- | --- | --- | | Step | Cause | System Response | | 1 | System read the sign language wrongly | System returns mismatch text or voice |   **Exceptions 1:**   |  |  |  | | --- | --- | --- | | Step | Cause | System Response | | 1 | Internet connection is lost | System shows message “You are offline, please check your connection”. |   **Exceptions 2:**   |  |  |  | | --- | --- | --- | | Step | Cause | System Response | | 1 | Bluetooth is disconnect | System shows message “Bluetooth is disconnected”. |   **Exceptions 3:**   |  |  |  | | --- | --- | --- | | Step | Cause | System Response | | 1 | Actor performs too fast | System shows the message “System can’t recognize your movement”. |   **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. * After recive EMG data from smartphone, server will compare the data with EMG data in database to find the best match then looking for the meaning of matching result to return to smartphone. * EMG data which is sent to server must be follow EMG data format * Result returns to user will be under text format. When user want to play sound, the system will use text-to-speech API of Google. | | | |

Table 16: Translate (Online) specification

##### <User> Buy license

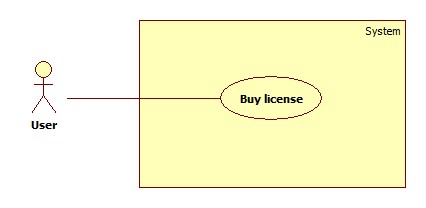


Figure 9: <User> Buy license

|  |  |  |  |
| --- | --- | --- | --- |
| **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 | Internet connection is lost | System shows message “You are offline, please check your connection”. |   **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 17: Buy license specification

##### <User> Search

Referance 2.3.1.3 <Guest> Search

#### <Premium User> Overview use case

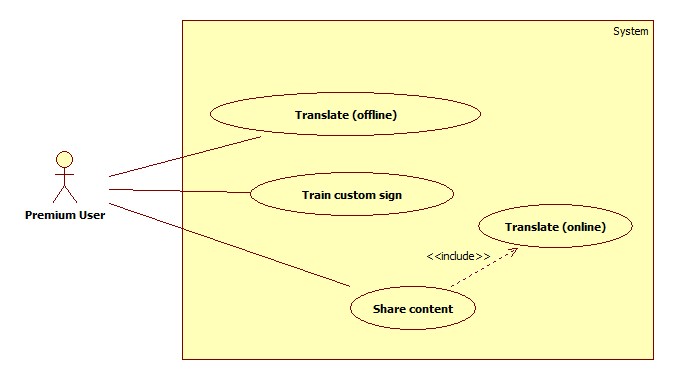


Figure 10: <Premium User> Overview Use case

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

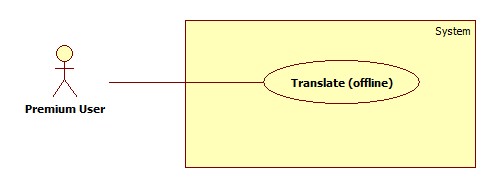


Figure 11: <Premium User> Translate (Offline)

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE – UC006** | | | |
| **Use Case No.** | 006 | **Use Case Version** | 2.0 |
| **Use Case Name** | Translate (offline) | | |
| **Author** | QuyPH | | |
| **Date** | January 23th,2016 | **Priority** | High |
| **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 two 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 | System shows message “Fail to connect” and requires to connect again |   **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 is disconnected”. |   **Exceptions 2:**   |  |  |  | | --- | --- | --- | | Step | Cause | System Response | | 1 | Actor performs too fast | System shows the message “System can’t recognize your movement”. |   **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. * After collect EMG data, server will compare the data with EMG data in local database to find the best match then looking for the meaning of matching result to return to premium user. * EMG data which is collected and use to compare must be follow EMG data format * Result returns to premium user will be under text format. When user want to play sound, the system will use text-to-speech API of Google. | | | |

Table 18: Translate (Offline) specification

##### <Premium User> Train custom sign

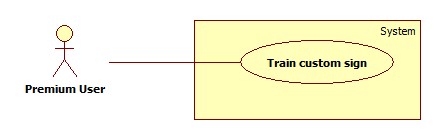
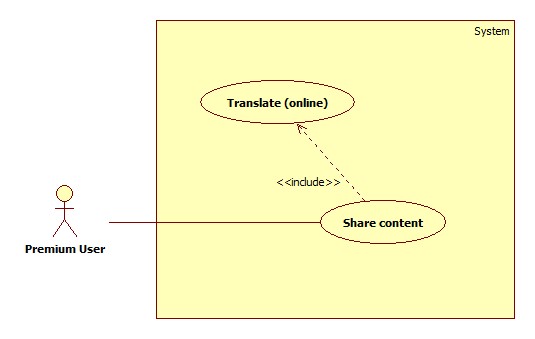


Figure 12: <Premium User> Train custom sign

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE – UC007** | | | |
| **Use Case No.** | 007 | **Use Case Version** | 2.0 |
| **Use Case Name** | Train custom sign | | |
| **Author** | ThaiTC | | |
| **Date** | 23/01/2016 | **Priority** | High |
| **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 two 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 | System shows message “Fail to connect” and requires to connect again |   **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 | Actor performs too fast | System shows the message “System can’t recognize your movement”. |   **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 two 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 * EMG data must follows EMG data format * 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 19: Train custom sign 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** | High |
| **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 two 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 | System shows message “Fail to connect” and requires to connect again |   **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 | Internet connection is lost | System shows message “You are offline, please check your connection”. |   **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 | Actor performs too fast | System shows the message “System can’t recognize your movement”. |   **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 * EMG data must follows EMG data format * 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 20: Share content specification

#### <Staff> Overview use case

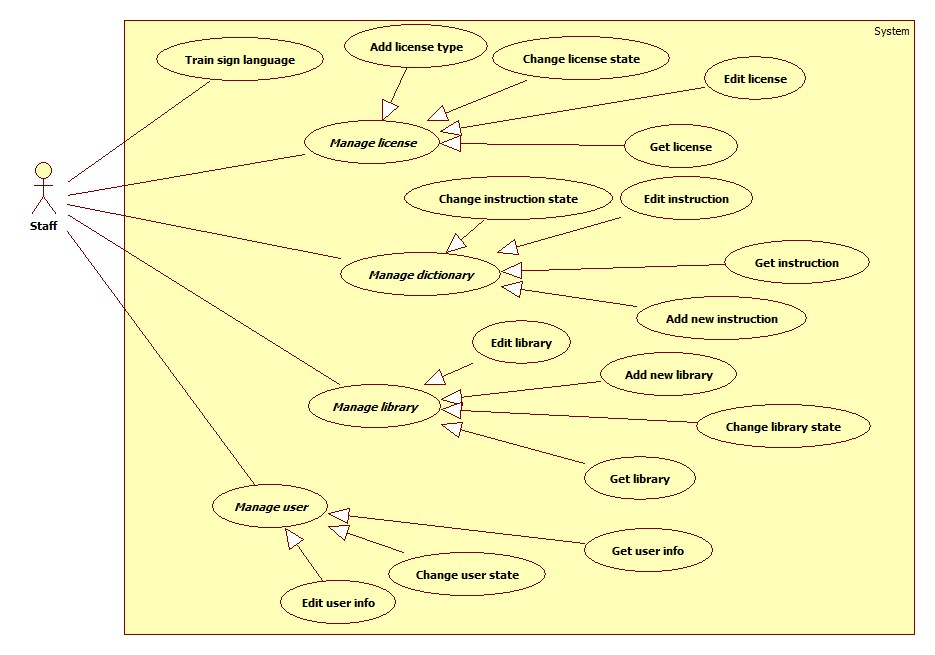
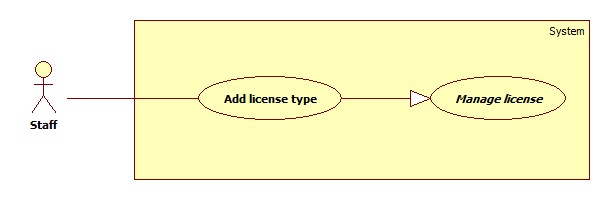


Figure 14: <Staff> Overview Use Case

##### <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 | Internet connection is lost | System shows message “You are offline, please check your connection”. |   **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 21: 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 | Internet connection is lost | System shows message “You are offline, please check your connection”. |   **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 22: Change license state specification

##### <Staff> Edit license

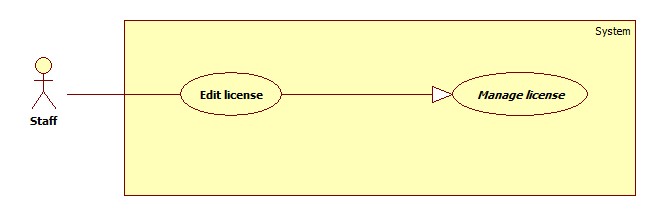


Figure 17: <Staff> Edit license

|  |  |  |  |
| --- | --- | --- | --- |
| **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 | Cause | System Response | | 1 | Internet connection is lost | System shows message “You are offline, please check your connection”. |   **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 23: Edit license specification

##### <Staff> Get license

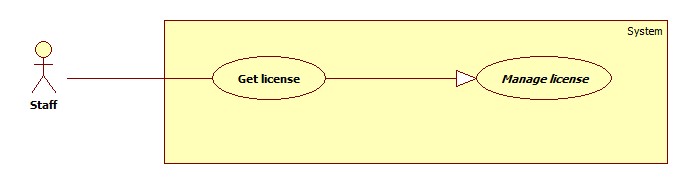


Figure 18: <Staff> Get license

|  |  |  |  |
| --- | --- | --- | --- |
| **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 | Internet connection is lost | System shows message “You are offline, please check your connection”. |   **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 24: 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(6-12) * Email, free text input, format(abc@abc.abc) * Fullname: free text input, length(10-50) * Phone: free text input, length(10-12)[0,9] * 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 | Internet connection is lost | System shows message “You are offline, please check your connection”. |   **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 25: 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 | Internet connection is lost | System shows message “You are offline, please check your connection”. |   **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 26: 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 | Internet connection is lost | System shows message “You are offline, please check your connection”. |   **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 27: Get user information specification

##### <Staff> Add new instruction

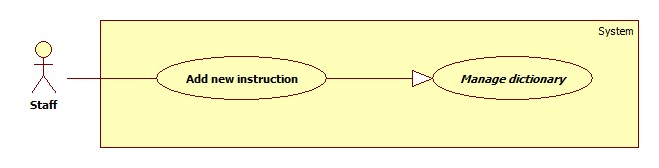


Figure 22: <Staff> Add new instruction

|  |  |  |  |
| --- | --- | --- | --- |
| **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(1-255) | | 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 | Internet connection is lost | System shows message “You are offline, please check your connection”. |   **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: Add new instruction specification

##### <Staff> Change instruction state

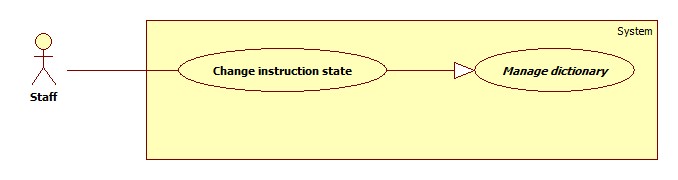


Figure 23: <Staff> Change instruction state

|  |  |  |  |
| --- | --- | --- | --- |
| **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 | Internet connection is lost | System shows message “You are offline, please check your connection”. |   **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 29: Change instruction state specification

##### <Staff> Edit instruction

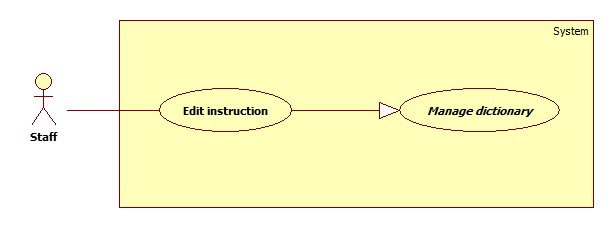


Figure 24: <Staff> Edit instruction

|  |  |  |  |
| --- | --- | --- | --- |
| **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(1-255) | | 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 | Internet connection is lost | System shows message “You are offline, please check your connection”. |   **Relationships:** N/A  **Business Rules:**   * After edit, new information of instruction will be updated in database of system | | | |

Table 30: Edit instruction specification

##### <Staff> Get instruction

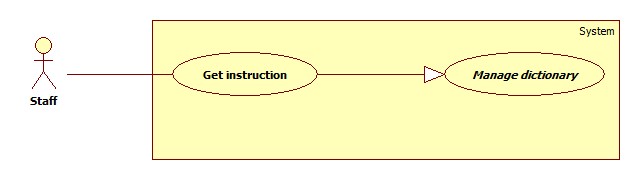


Figure 25: <Staff> Get instruction

|  |  |  |  |
| --- | --- | --- | --- |
| **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 | Internet connection is lost | System shows message “You are offline, please check your connection”. |   **Relationships:** N/A  **Business Rules:**   * After get instruction, staff can view all information which is stored in database of selected instruction. | | | |

Table 31: Get instruction specification

##### <Staff> Add new library

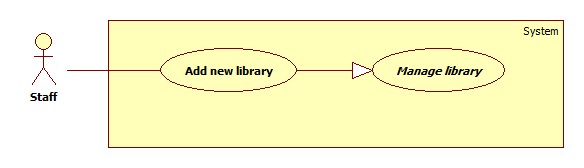


Figure 26: <Staff> Add new library

|  |  |  |  |
| --- | --- | --- | --- |
| **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 | Internet connection is lost | System shows message “You are offline, please check your connection”. |   **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 32: Add new library specification

##### <Staff> Change library state

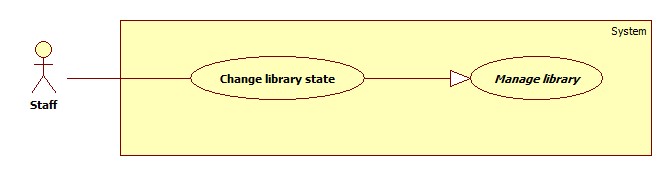


Figure 27: <Staff> Change library state

|  |  |  |  |
| --- | --- | --- | --- |
| **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 | Internet connection is lost | System shows message “You are offline, please check your connection”. |   **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 33: Change library state specification

##### <Staff> Edit library

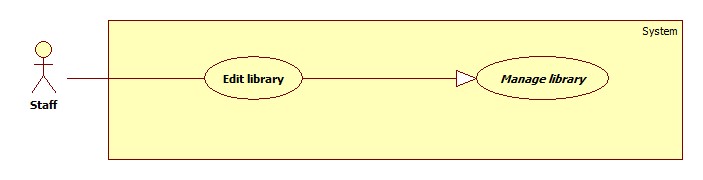


Figure 28: <Staff> Edit library

|  |  |  |  |
| --- | --- | --- | --- |
| **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 | Internet connection is lost | System shows message “You are offline, please check your connection”. |   **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 34: Edit library specification

##### <Staff> Get library

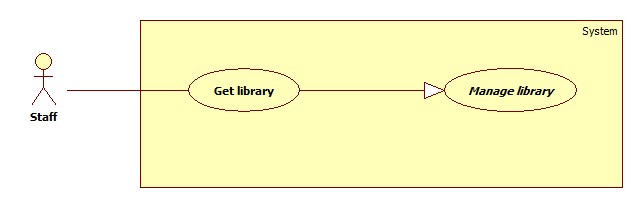


Figure 29: <Staff> Get library

|  |  |  |  |
| --- | --- | --- | --- |
| **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 | Cause | System Response | | 1 | Internet connection is lost | System shows message “You are offline, please check your connection”. |   **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 35: Get library specification

##### <Staff> Train sign language

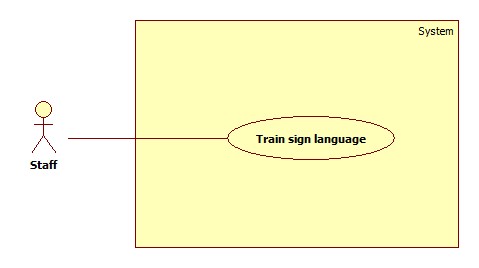


Figure 30: <Staff> Train sign language

|  |  |  |  |
| --- | --- | --- | --- |
| **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** | High |
| **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 move * Meaning of sign language: free text input, required, length (1 - 50) | | 2 | Staff connects smartphone with tow 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 | System shows message “Fail to connect” and requires to connect again |   **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 is disconnected” when the Bluetooth connection between the Android smartphone and the MYO armbands is lost |   **Exceptions 2:**   |  |  |  | | --- | --- | --- | | Step | Cause | System Response | | 1 | Actor performs too fast | System shows the message “System can’t recognize your movement”. |   **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 two 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 * EMG data must follow EMG data format * 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 36: Train sign language specification

#### <Authenticated user> Overview use case

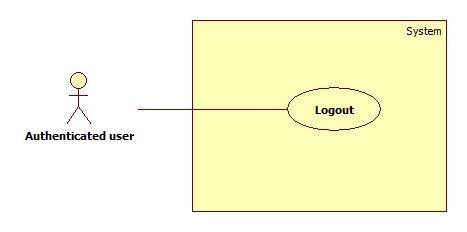


Figure 31: <Authenticated user> Overview Use Case

##### <Authenticated user> Logout

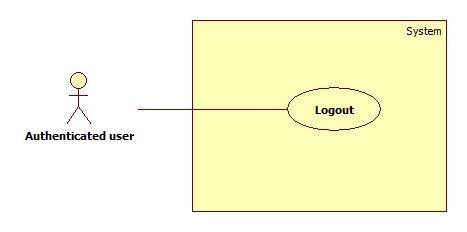


Figure 32: <Authenticated user> Logout

|  |  |  |  |
| --- | --- | --- | --- |
| **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 37: Logout specification

#### <Scheduler> Overview use case

Figure 33: <Scheduler> Overview Use Case

##### <Scheduler> Create notification

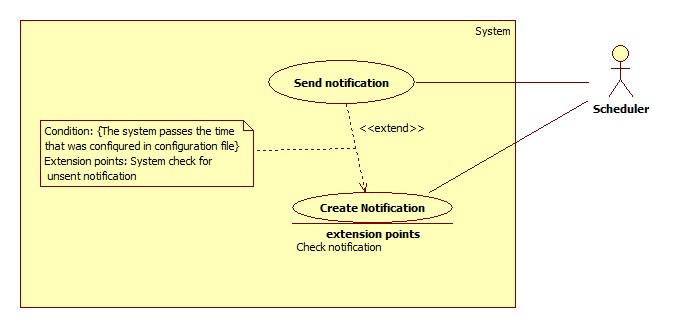
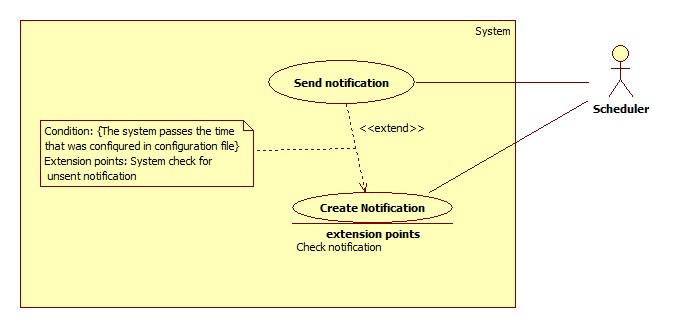


Figure 34: <Scheduler> Create notification

|  |  |  |  |
| --- | --- | --- | --- |
| **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:** No new notification save in the storage, 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 | Cause | System Response | | 1 | Condition does not meet | Notification isn’t created, 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 * The notification will be created with the status is “unsent” * Log file is generate: * id: {Created date + Created time} * Content: {notification id}, {account notification belongs}, {total crated notification} | | | |

Table 38: 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 from “unsent” to “sent” * **Fail:** Notification is not sent to premium user, status of notification is change from “unsent” 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 from”unsent” into “sent”   [Exception 1] |   **Exceptions 1:**   |  |  |  | | --- | --- | --- | | Step | Cause | System Response | | 1 | Notification sent fail | System retries send notification after 30 minutes |   **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” * When the condition is met, the notification will be sent * After sent, the status of notification will be changed into “sent” | | | |

Table 39: Send notification specification

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

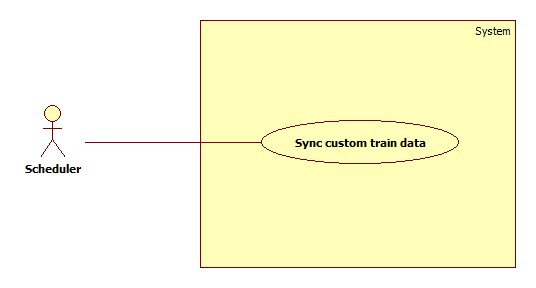


Figure 36: <Scheduler> Sync custom train data

|  |  |  |  |
| --- | --- | --- | --- |
| **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 | Internet connection is lost | System shows message “You are offline, please check your connection”. |   **Exceptions 1:**   |  |  |  | | --- | --- | --- | | Step | Cause | 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 40: Sync custom train data specification

## Software system attribute

### Usability

* System provides user friendly GUI with instruction

### Reliability

* The system uses electromyography technology to read activity of user’s muscle so the accuracy is higher than other systems

### Availability

* The system relates to communication so it can be available 24/7. The only case that makes users cannot use the system is they are not connected to the internet and they are not premium user.
* When a function is down, it will not impact other functions

### Mantainability

* When a module of a function is down, it is easy to take it down to fix without impact other functions

### Portability

* With required device (an internet connected android smartphone with system application installed, two MYO armbands), user can use the system anywhere, anytime

### Performance

* System can translate user sign language into text or speech after user do the stop sign in the range of 3 to 5 secconds.

## Conceptual diagram

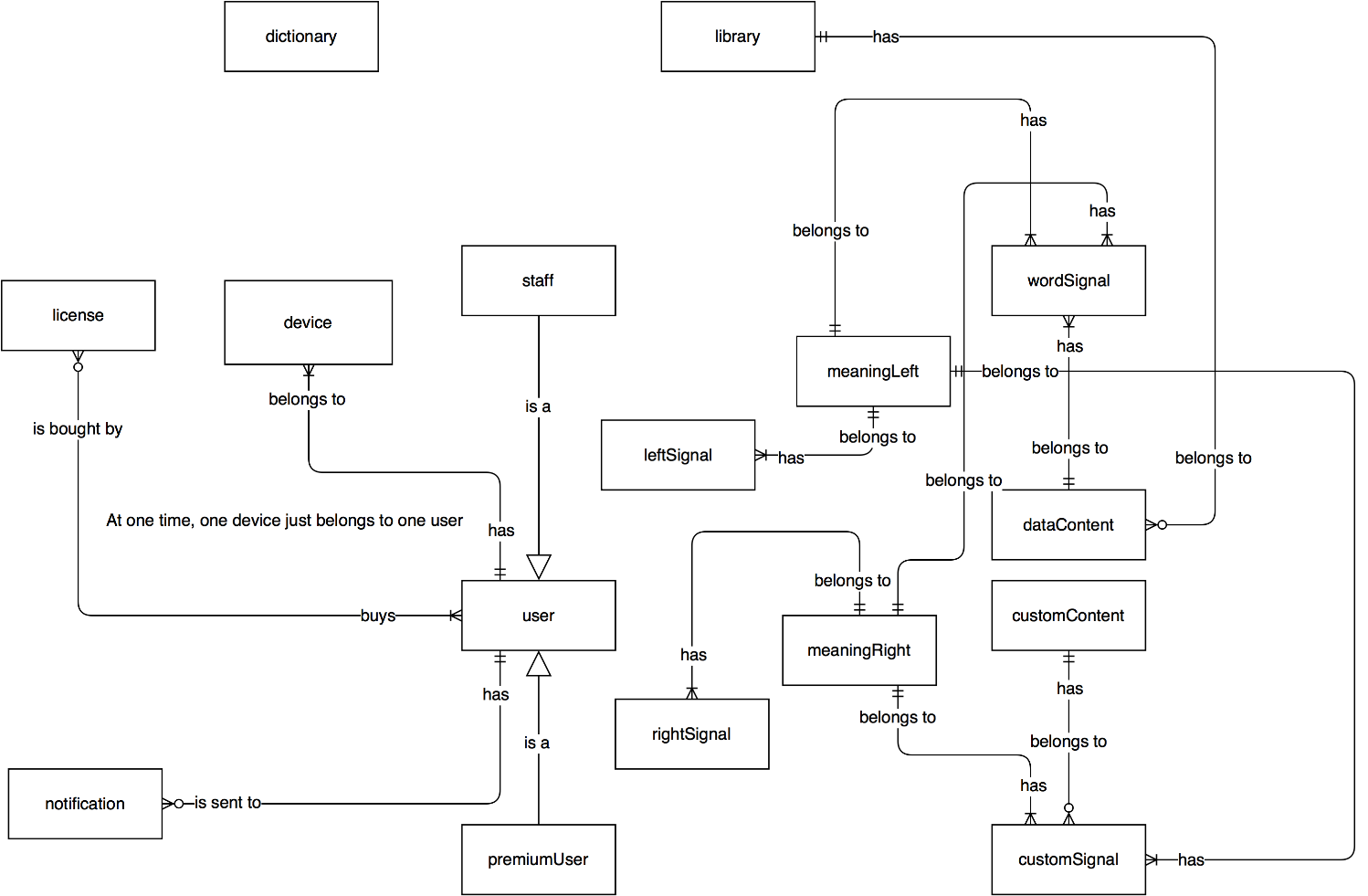


Figure 37: Conceptual diagram

|  |  |
| --- | --- |
| 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 |
| Dictionary | Contain the dictionary 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 |
| meaningLeft | Contain the meaningLeft information |
| meaningRight | Contain the meaningRight information |

Table 41: Conceptual diagram data dictionary

# Report No.4 Software Design Description

## Design Overview

* This document describes the technical and user interface design of **CBYH System**. It includes the architectural design, the detailed design of common functions and business functions and the design of database model.
* The architectural design describes the overall architecture of the system and the architecture of each main component and subsystem.
* The detailed design describes static and dynamic structure for each component and functions. It includes class diagrams, class explanations and sequence diagrams for each use cases.
* The database design describes the relationships between entities and details of each entity.
* Document overview:
  + Section 2: gives an overall description of the system architecture design.
  + Section 3: gives component diagrams that describe the connection and integration of the system.
  + Section 4: gives the detail design description, which includes class diagram, class explanation, and sequence diagram to details the application functions.
  + Section 5: describe screens design.
  + Section 6: describe a fully attributed ERD.
  + Section 7: describe algorithms***.***

## System Architectural Design

Figure 38: System architecture design

This diagram is referenced and modified from an original concept from: Chapter 6 Architecture Design, SOFTWARE ENGINEERING 9th Edition, by Ian Sommerville.

### Web application architecture description

In Web Application, the system is developed under J2EE MVC architecture style. We choose this architecture for Web application because of following advantages:

* Web app contains a Web service (public API for mobile app), with MVC architecture, we can separate business code with Controller and View, so we can use the business code in web service without repeat the code.
* Current system only supports motor insurance card, with MVC architecture, we can organize the code better for maintainability, extensibility, reusability so we can expand the scope to other kind of insurance services such as Heath insurance or Asset insurance…
* In scope of 4-member team, MVC architecture make it easier to split the big project into small modules and make it easier to assign each module for members in our team.

This project follows MVC architecture with following components:

* **Controller (Servelet)** is the parts of the application that acts like event handler to handles user interaction. Typically, controller read data from a request and calls appropriate Business’s method then selects view to return to user.
* **JSP/HTML (View)** is the parts of the application that handles the display of the data. The selection of View is under control of Controller.
* **Model** is the parts of the application that do business processing to solve domain problems.
* **Entity** is the parts of the application that acts like a data transfer object between the system and database.
* **Web Service** is the parts of the application that acts like event handler for web and mobile communication via REST method.

### Mobile application architecture description

The application is developed as an Android native application. In general, the application architecture conforms to Android architecture.

Reason for choosing: Currently, Android is the most common mobile flatform in the world.

This project follows Android application architecture with following component:

* **Activity** is the basic core of an android application that handles user input,create thread to run asynchronous tasks, send request and receive data from server via servlet and web services ...
* **Model** is the part of the application that handles the management of fragment and database.
* **View** is a part of the application that represents a behavior or a portion of user interface in an Activity.

## Component Diagram

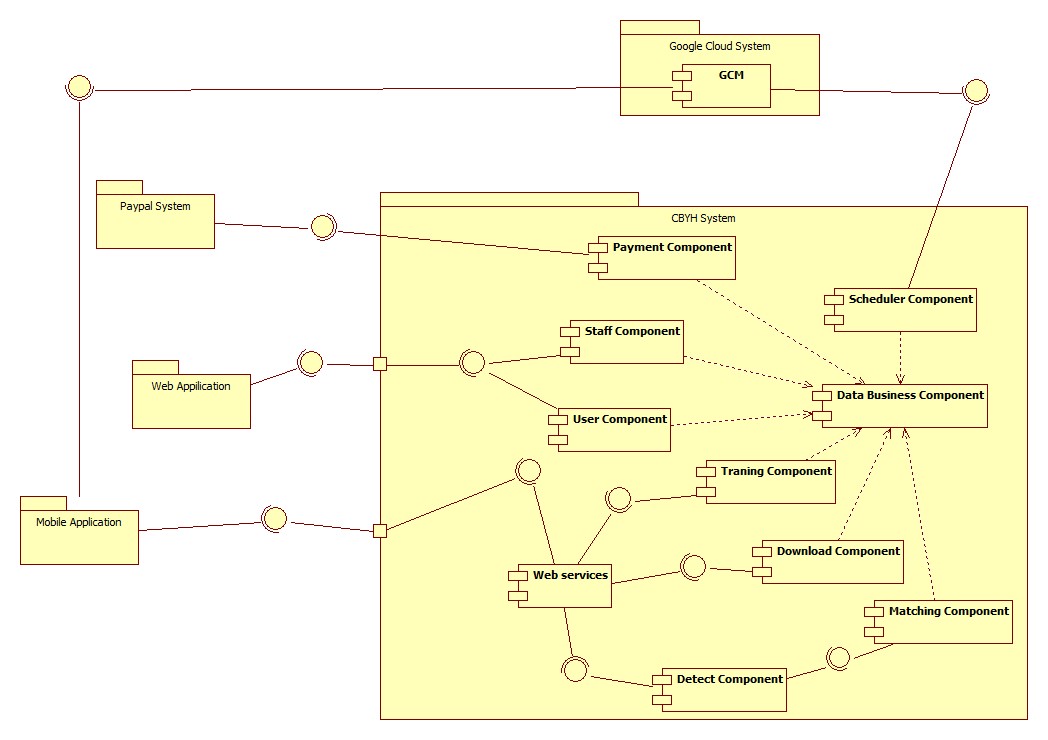
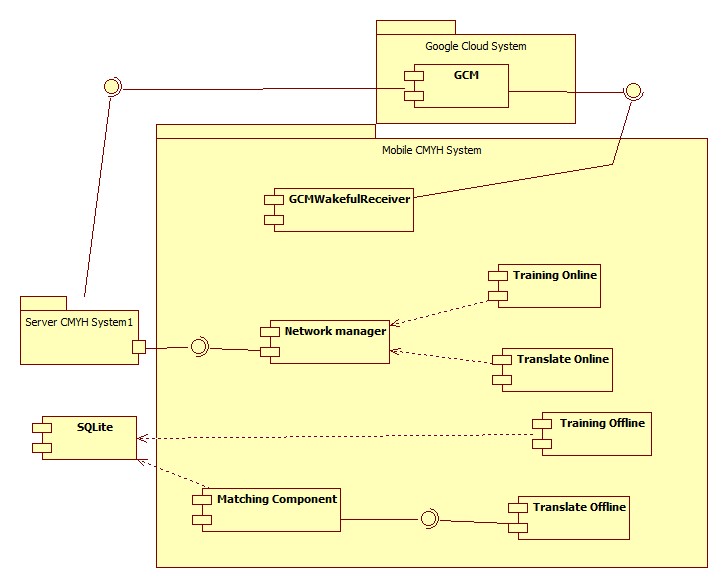


Figure 39: Component Diagram (Server side)

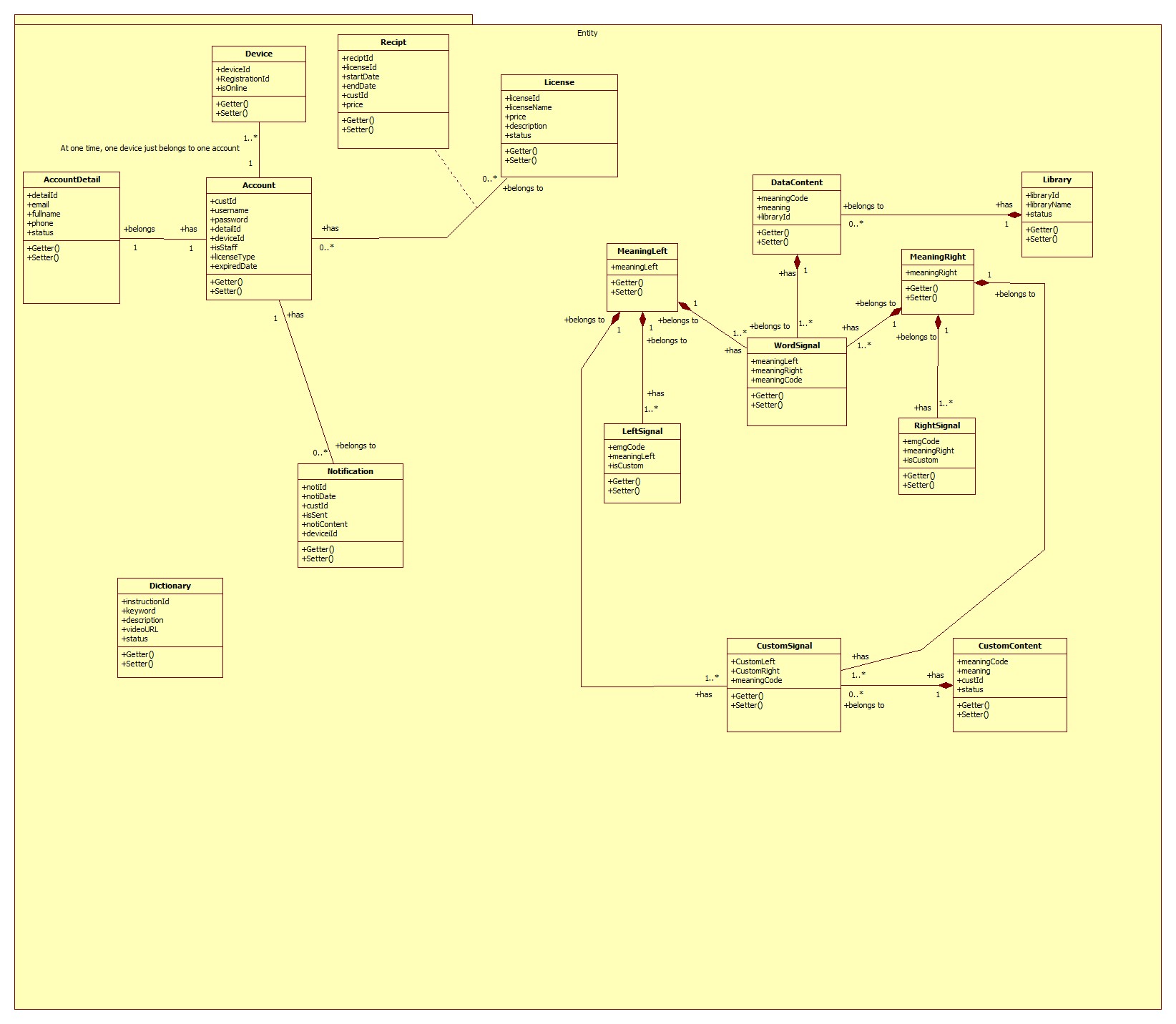
Figure 40: Component Diagram (Mobile side)

|  |  |
| --- | --- |
| Component Dictionary: Describes components | |
| Web Application | Web application package: View, Controller |
| Mobile Application | Mobile application package |
| PayPal | Handle payment process with PayPal API |
| GCM (Google Cloud Message) | Handle sending notification to mobile device |
| Payment Component | Component to handle payment process |
| Scheduler Component | Component to handle scheduler in the system |
| Staff Component | Component to handle Staff’s functions in the system |
| User Component | Component to handle User’s functions in the system |
| Data Business Component | Common objects to handle domain business operations for each components |
| Training Component | Component to handle training new word process |
| Matching Component | Component to handle matching process |
| Detect Component | Componet to handle detect process |
| Download Component | Component to handle download data process |
| Web services | Component to handle provide API for mobile application |
| GCMWalkfulReceiver | Component to receive notification from Google Cloud Message on mobile |
| SQLite | Component that manage database on mobile |
| Network Manager | Component that call API |
| Training Online | Component to handle training online new word process on mobile |
| Translate Online | Component to handle translate online process on mobile |
| Training Offline | Component to handle training offline process on mobile |
| Translate Offline | Component to handle translate offline process on mobile |

Table 42: Component Dictionary

## Detail Description

### Class Diagram

Figure 41: Class Diagram

|  |  |  |
| --- | --- | --- |
| Class dictionary: describe Class | | |
| Class Name | **Mapping column with Conceptual diagram** | **Description** |
| Account | user | Contain the account information |
| AccountDetail | staff, premiumUser | Contain the account detail information |
| Dictionary | dictionary | Contain the dictionary infromation |
| Notification | notification | Contain the notification information |
| License | license | Contain the license information |
| Library | library | Contain the library information |
| Device | device | Contain the device information |
| LeftSignal | leftSignal | Contain the left signal information |
| RightSignal | rightSignal | Contain the right signal information |
| MeaningLeft | meaningLeft | Contain the meaning left information |
| MeaingRight | meaningRight | Contain the meaning right information |
| WordSignal | wordSignal | Contain the word signal information |
| DataContent | dataContent | Contain the data content information |
| CustomSignal | customSignal | Contain the custom signal information |
| CustomContent | customContent | Contain the custom content information |
| Recipt | N/A | Not exists in conceptual diagram but need to contain the recipt when user buy a license |

Table 43: Class dictionary

### Class Diagram Explanation

#### Account

|  |  |  |  |
| --- | --- | --- | --- |
| Attribute | Type | Visibility | Description |
| custId | Integer | Private | Unique identifier of an account |
| username | String | Private | User’s username |
| password | String | Private | User’s password |
| detailId | Integer | Private | The id of detail of user |
| deviceId | String | Private | The user’s device ID |
| isStaff | Boolean | Private | Staff checker |
| licenseType | String | Private | User’s license type |
| expiredDate | Date | Private | User’s expiredDate of license |

Table 44: Account Attribute

|  |  |  |  |
| --- | --- | --- | --- |
| Method | Return Type | Visibility | Description |
| Getter | Attribute type | Public | Get attribute value |
| Setter | Void | Public | Set value of attribute |

Table 45: Account Method

#### AccountDetail

|  |  |  |  |
| --- | --- | --- | --- |
| Attribute | Type | Visibility | Description |
| detailId | Integer | Private | Unique identifier of an account detail |
| email | String | Private | User’s email |
| fullname | String | Private | User’s fullname |
| phone | String | Private | User’s phone |
| deviceId | Integer | Private | The device ID |
| status | Boolean | Private | Active account checker |

Table 46: AccountDetail Attribute

|  |  |  |  |
| --- | --- | --- | --- |
| Method | Return Type | Visibility | Description |
| Getter | Attribute type | Public | Get attribute value |
| Setter | Void | Public | Set value of attribute |

Table 47: AccountDetail Method

#### Dictionary

|  |  |  |  |
| --- | --- | --- | --- |
| Attribute | Type | Visibility | Description |
| instructionId | Integer | Private | Unique identifier of an instruction |
| keyword | String | Private | Instruction’s keyword |
| description | String | Private | Instruction’s description |
| videoURL | String | Private | Instruction’s videoURL |
| status | Boolean | Private | Active instruction checker |

Table 48: Dictionary Attribute

|  |  |  |  |
| --- | --- | --- | --- |
| Method | Return Type | Visibility | Description |
| Getter | Attribute type | Public | Get attribute value |
| Setter | Void | Public | Set value of attribute |

Table 49: Dictionary Method

#### Notification

|  |  |  |  |
| --- | --- | --- | --- |
| Attribute | Type | Visibility | Description |
| notiId | Integer | Private | Unique identifier of an notification |
| notiDate | Date | Private | Notification date |
| custId | Integer | Private | Id of notification’s owner |
| isSent | Boolean | Private | Sent notification tracker |
| notiContent | String | Private | Notification content |
| deviceId | String | Private | Id of device that notification is sent to |

Table 50: Notification Attribute

|  |  |  |  |
| --- | --- | --- | --- |
| Method | Return Type | Visibility | Description |
| Getter | Attribute type | Public | Get attribute value |
| Setter | Void | Public | Set value of attribute |

Table 51: Notification Method

#### License

|  |  |  |  |
| --- | --- | --- | --- |
| Attribute | Type | Visibility | Description |
| licenseId | Integer | Private | Unique identifier of a license |
| licenseName | String | Private | License name |
| price | Double | Private | License price |
| description | String | Private | Sent notification tracker |
| status | Boolean | Private | Active license checker |

Table 52: License Attribute

|  |  |  |  |
| --- | --- | --- | --- |
| Method | Return Type | Visibility | Description |
| Getter | Attribute type | Public | Get attribute value |
| Setter | Void | Public | Set value of attribute |

Table 53: Notification Method

#### Library

|  |  |  |  |
| --- | --- | --- | --- |
| Attribute | Type | Visibility | Description |
| libraryId | Integer | Private | Unique identifier of a library |
| libraryName | String | Private | Library name |
| status | Boolean | Private | Active library checker |

Table 54: Library Attribute

|  |  |  |  |
| --- | --- | --- | --- |
| Method | Return Type | Visibility | Description |
| Getter | Attribute type | Public | Get attribute value |
| Setter | Void | Public | Set value of attribute |

Table 55: Notification Method

#### Device

|  |  |  |  |
| --- | --- | --- | --- |
| Attribute | Type | Visibility | Description |
| deviceId | String | Private | Unique identifier of a device |
| registrationId | String | Private | Identifier of a application on a device |
| isOnline | boolean | Private | Is online device checker |

Table 56: Device Attribute

|  |  |  |  |
| --- | --- | --- | --- |
| Method | Return Type | Visibility | Description |
| Getter | Attribute type | Public | Get attribute value |
| Setter | Void | Public | Set value of attribute |

Table 57: Device Method

#### LeftSignal

|  |  |  |  |
| --- | --- | --- | --- |
| Attribute | Type | Visibility | Description |
| emgCode | String | Private | Unique identifier of a left signal |
| meaningLeft | Integer | Private | Meaning code of a left signal |
| isCustom | Boolean | Private | Is custom word checker |

Table 58: LeftSignal Attribute

|  |  |  |  |
| --- | --- | --- | --- |
| Method | Return Type | Visibility | Description |
| Getter | Attribute type | Public | Get attribute value |
| Setter | Void | Public | Set value of attribute |

Table 59: LeftSignal Method

#### RightSignal

|  |  |  |  |
| --- | --- | --- | --- |
| Attribute | Type | Visibility | Description |
| emgCode | String | Private | Unique identifier of a right signal |
| meaningRight | Integer | Private | Meaning code of a right signal |
| isCustom | Boolean | Private | Is custom word checker |

Table 60: RightSignal Attribute

|  |  |  |  |
| --- | --- | --- | --- |
| Method | Return Type | Visibility | Description |
| Getter | Attribute type | Public | Get attribute value |
| Setter | Void | Public | Set value of attribute |

Table 61: RightSignal Method

#### MeaningLeft

|  |  |  |  |
| --- | --- | --- | --- |
| Attribute | Type | Visibility | Description |
| meaningLeft | String | Private | Unique identifier of a meaning left |

Table 62: MeaningLeft Attribute

|  |  |  |  |
| --- | --- | --- | --- |
| Method | Return Type | Visibility | Description |
| Getter | Attribute type | Public | Get attribute value |
| Setter | Void | Public | Set value of attribute |

Table 63: MeaningLeft Method

#### MeaningRight

|  |  |  |  |
| --- | --- | --- | --- |
| Attribute | Type | Visibility | Description |
| meaningRight | String | Private | Unique identifier of a meaning right |

Table 64: MeaningRight Attribute

|  |  |  |  |
| --- | --- | --- | --- |
| Method | Return Type | Visibility | Description |
| Getter | Attribute type | Public | Get attribute value |
| Setter | Void | Public | Set value of attribute |

Table 65: MeaningRight Method

#### WordSignal

|  |  |  |  |
| --- | --- | --- | --- |
| Attribute | Type | Visibility | Description |
| meaningLeft | Integer | Private | Meaning left of a word |
| meaningRight | Integer | Private | Meaning right of a word |
| meaningCode | Integer | Private | Meaning code of a word |

Table 66: WordSignal Attribute

|  |  |  |  |
| --- | --- | --- | --- |
| Method | Return Type | Visibility | Description |
| Getter | Attribute type | Public | Get attribute value |
| Setter | Void | Public | Set value of attribute |

Table 67: WordSignal Method

#### DataContent

|  |  |  |  |
| --- | --- | --- | --- |
| Attribute | Type | Visibility | Description |
| meaningCode | Integer | Private | Unique identifier of a data content |
| meaning | String | Private | Meaning of a word |
| libraryId | Integer | Private | Id of the library that the word belongs |

Table 68: DataContent Attribute

|  |  |  |  |
| --- | --- | --- | --- |
| Method | Return Type | Visibility | Description |
| Getter | Attribute type | Public | Get attribute value |
| Setter | Void | Public | Set value of attribute |

Table 69: WordSignal Method

#### CustomSignal

|  |  |  |  |
| --- | --- | --- | --- |
| Attribute | Type | Visibility | Description |
| customLeft | Integer | Private | Meaning left of a custom word |
| customRight | Integer | Private | Meaning right of a custom word |
| meaningCode | Integer | Private | Meaning code of a custom word |

Table 70: CustomSignal Attribute

|  |  |  |  |
| --- | --- | --- | --- |
| Method | Return Type | Visibility | Description |
| Getter | Attribute type | Public | Get attribute value |
| Setter | Void | Public | Set value of attribute |

Table 71: CustomSignal Method

#### CustomContent

|  |  |  |  |
| --- | --- | --- | --- |
| Attribute | Type | Visibility | Description |
| meaningCode | Integer | Private | Meaning code of a custom word |
| meaning | String | Private | Meaning of a custom word |
| custId | Integer | Private | Id of the user that create the custom word |
| status | Boolean | Private | Active custom word checker |

Table 72: CustomContent Attribute

|  |  |  |  |
| --- | --- | --- | --- |
| Method | Return Type | Visibility | Description |
| Getter | Attribute type | Public | Get attribute value |
| Setter | Void | Public | Set value of attribute |

Table 73: CustomContent Method

#### Recipt

|  |  |  |  |
| --- | --- | --- | --- |
| Attribute | Type | Visibility | Description |
| reciptId | Integer | Private | Unique identifier of a recipt |
| licenseId | Integer | Private | The id of the license that user bought |
| startDate | Date | Private | The start date of license usage |
| endDate | Date | Private | The end date of license usage |
| custId | Integer | Private | Id of user bought the license |
| price | Double | Private | Price of the license |

Table 74: Recipt Attribute

|  |  |  |  |
| --- | --- | --- | --- |
| Method | Return Type | Visibility | Description |
| Getter | Attribute type | Public | Get attribute value |
| Setter | Void | Public | Set value of attribute |

Table 75: Recipt Method

### Interactive Diagram

#### Web Server

##### <Web Server> Translate

Summary: this diagram show process of translating on server

Precondition: MYO armbands must be paired with mobile device

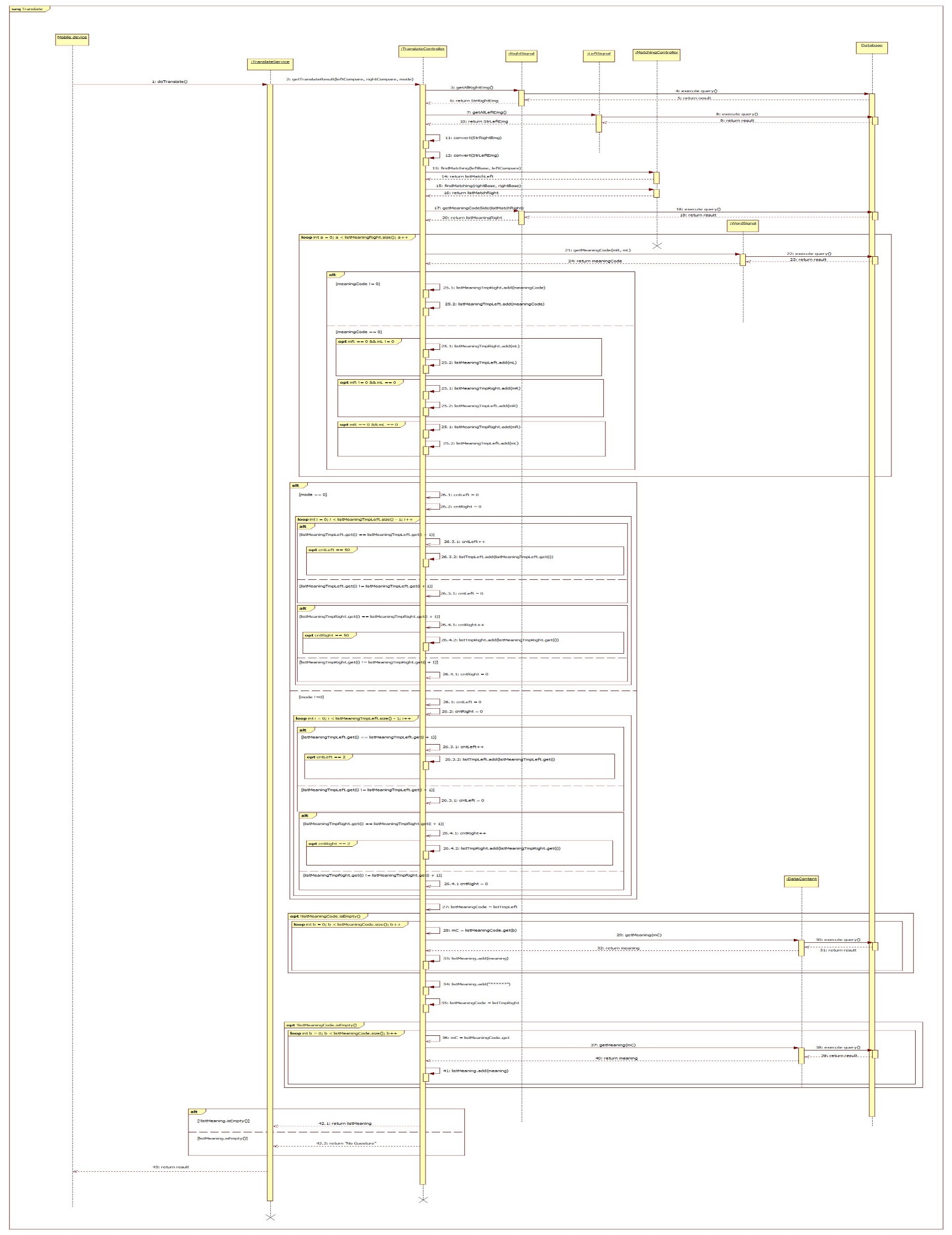


Figure 42: Sequence diagram - <Web Server> Translate

##### <Web Server> Train

Summary: this diagram show process of training new word on server

Precondition: MYO armbands must be paired with mobile device

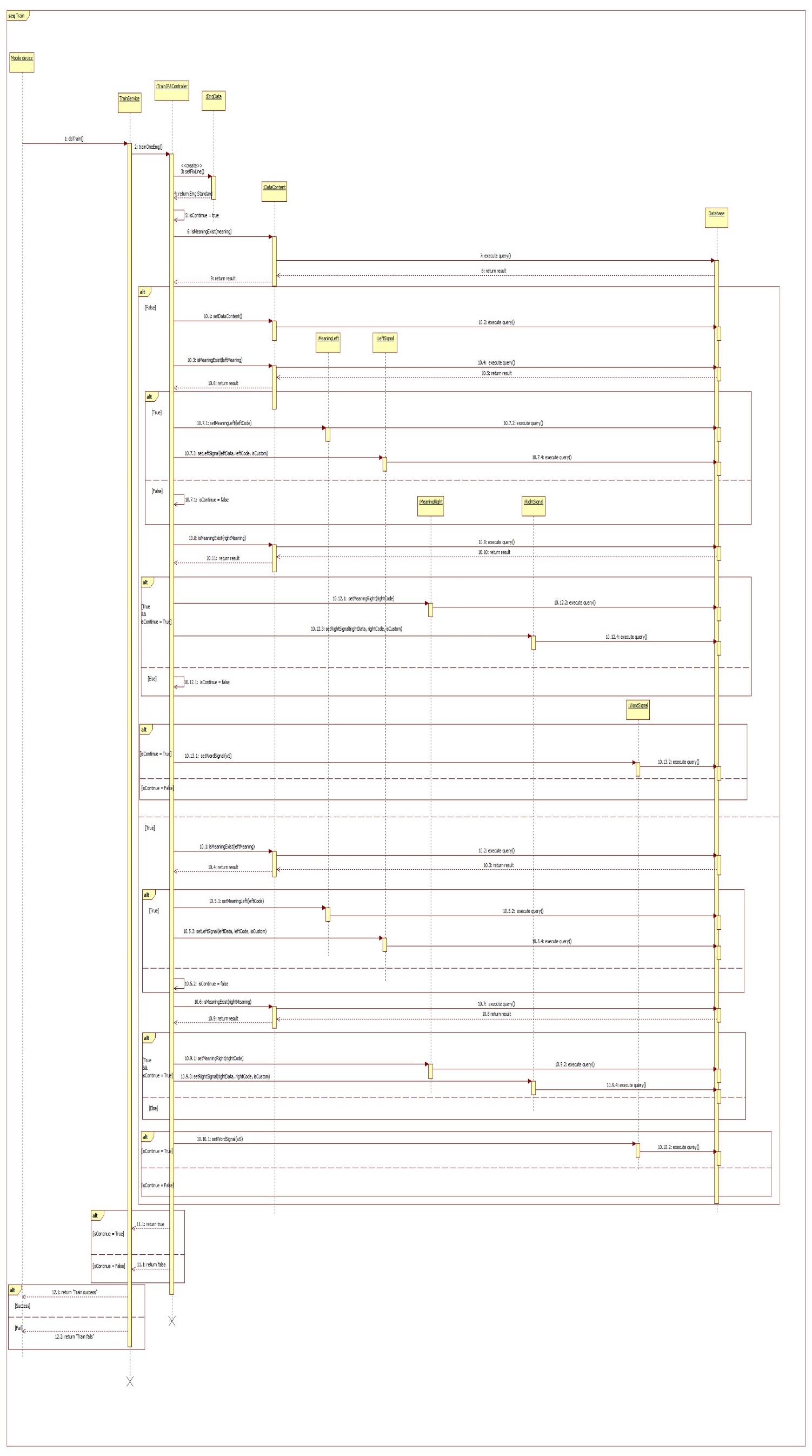


Figure 43: Sequence diagram - <Web Server> Train

#### Web Aplication

##### <Staff>

###### Add dictionary

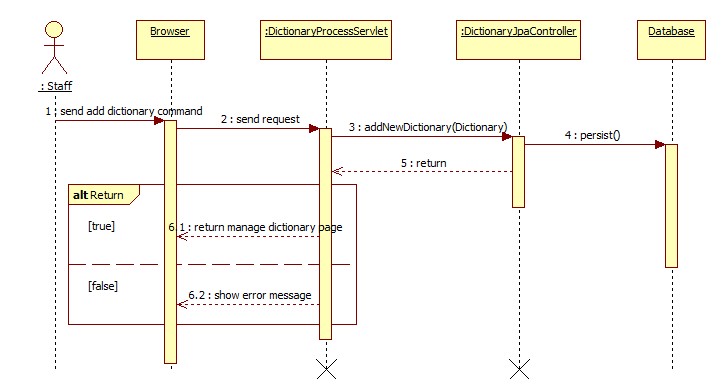
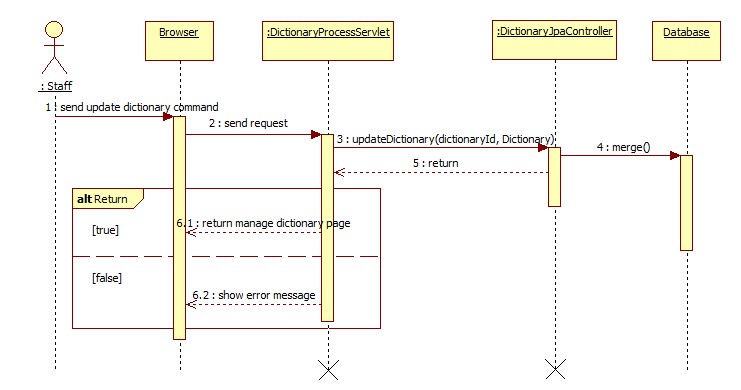
Summary: this diagram show process of Add new dictionary

Figure 44: Sequence diagram - <Staff> Add dictionary

###### Update dictionary

Summary: this diagram show process of update dictionary

Figure 45: Sequence diagram - <Staff> Update dictionary

###### Get dictionary

Summary: this diagram show process of get dictionary

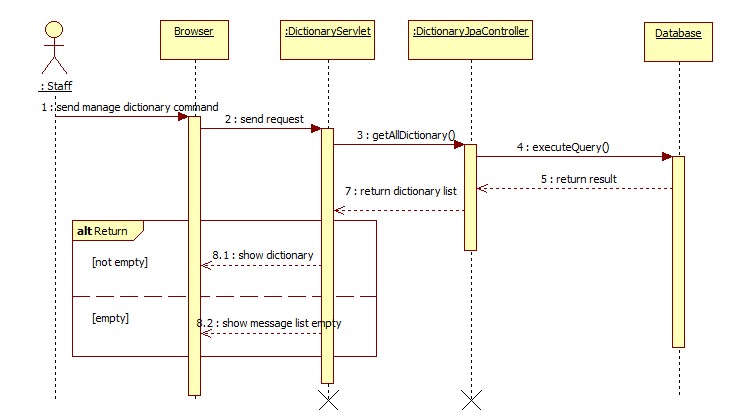


Figure 46: Sequence diagram - <Staff> Get dictionary

###### Add library

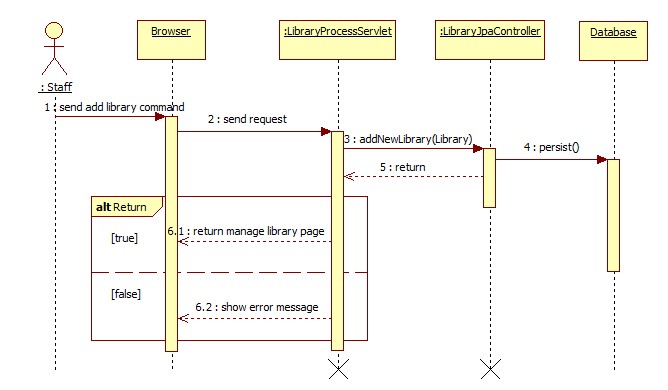
Summary: this diagram show process of add library

Figure 47: Sequence diagram - <Staff> Add library

###### Update library

Summary: this diagram show process of update library

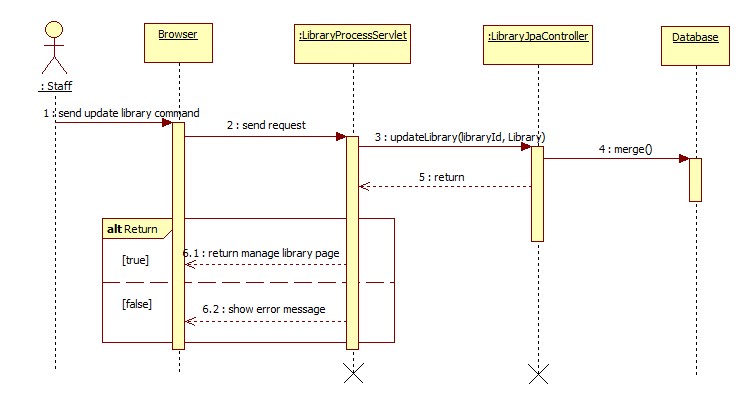


Figure 48: Sequence diagram - <Staff> Update library

###### Get library

Summary: this diagram show process of get library

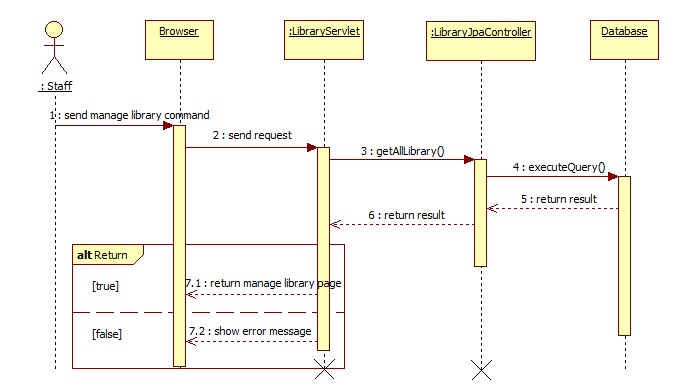


Figure 49: Sequence diagram - <Staff> Get library

###### Update license

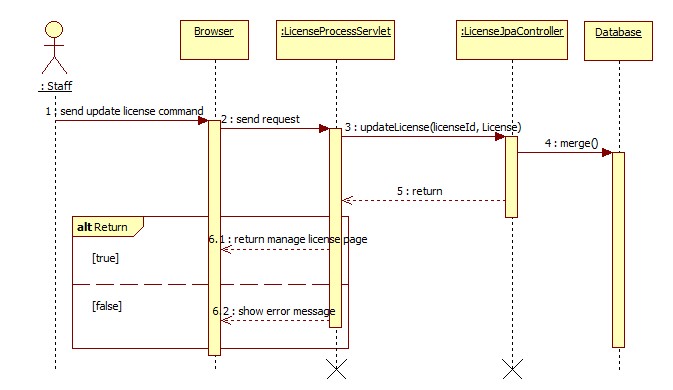
Summary: this diagram show process of update license

Figure 50: Sequence diagram - <Staff> Update license

###### Update user

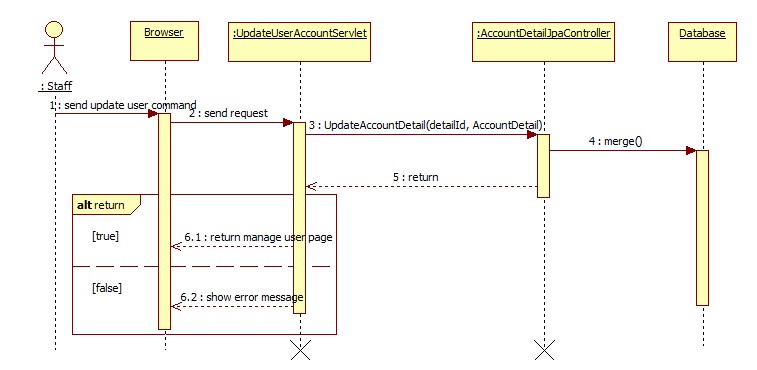
Summary: this diagram show process of update user

Figure 51: Sequence diagram - <Staff> Update user

###### Get user information

Summary: this diagram show process of get user information

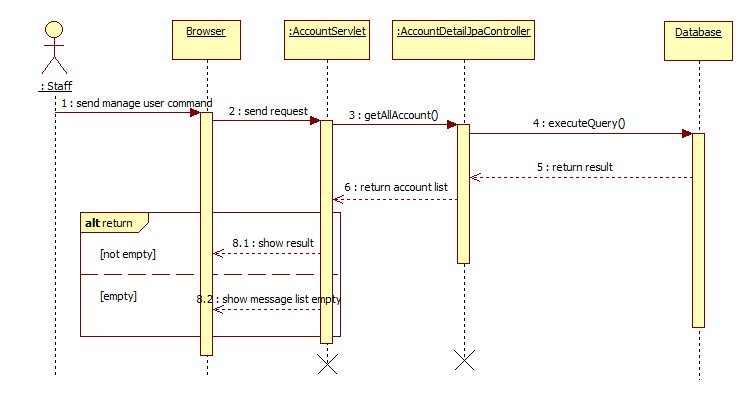


Figure 52: Sequence diagram - <Staff> Get user information

##### <User>

###### Buy license

Summary: this diagram show process of buy license

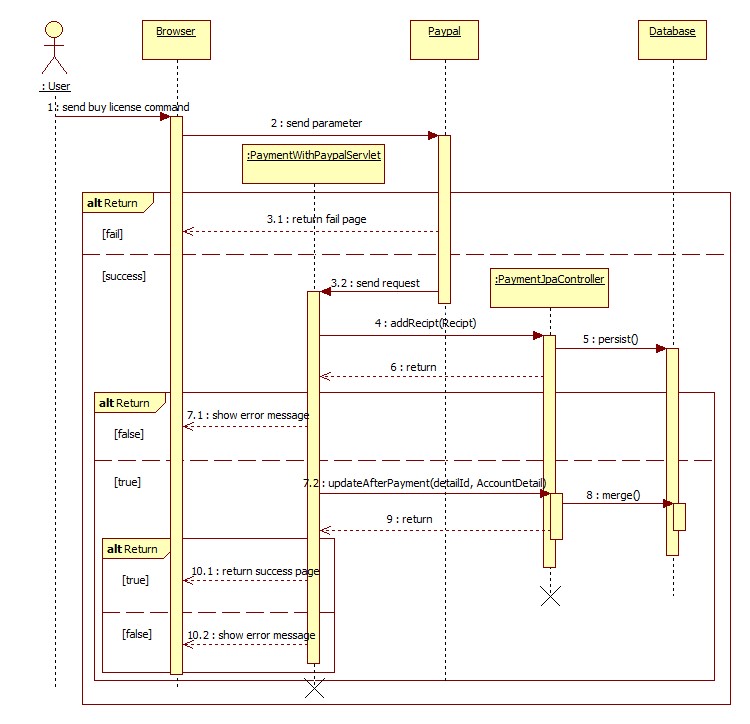


Figure 53: Sequence diagram - <User> Buy license

###### Update profile

Summary: this diagram show process of update profile

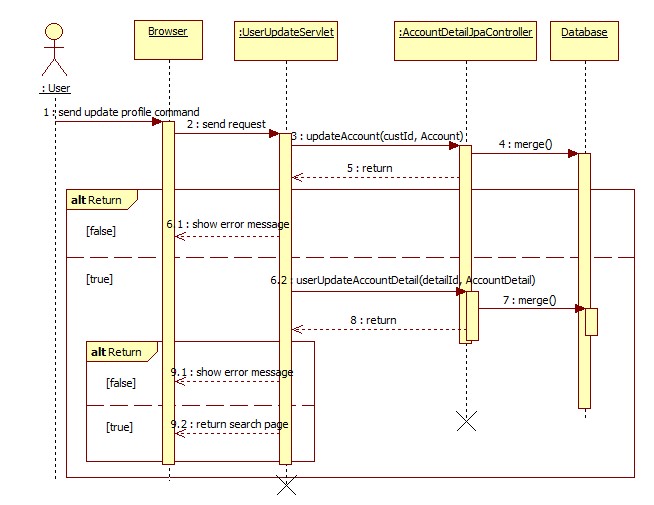


Figure 54: Sequence diagram - <User> Update profile

###### Search

Summary: this diagram show process of search

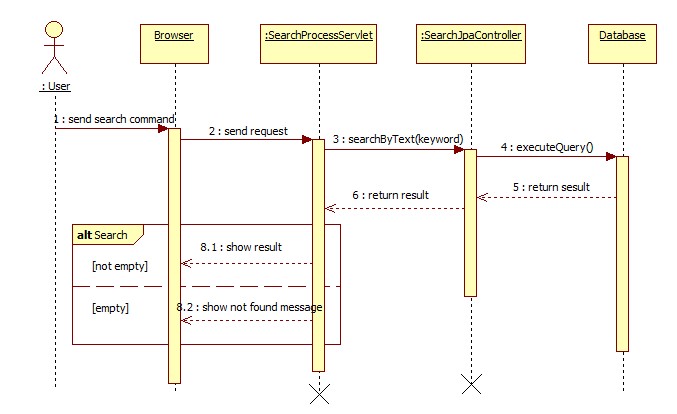


Figure 55: Sequence diagram - <User> Search

##### <Guest>

###### Register

Summary: this diagram show process of register

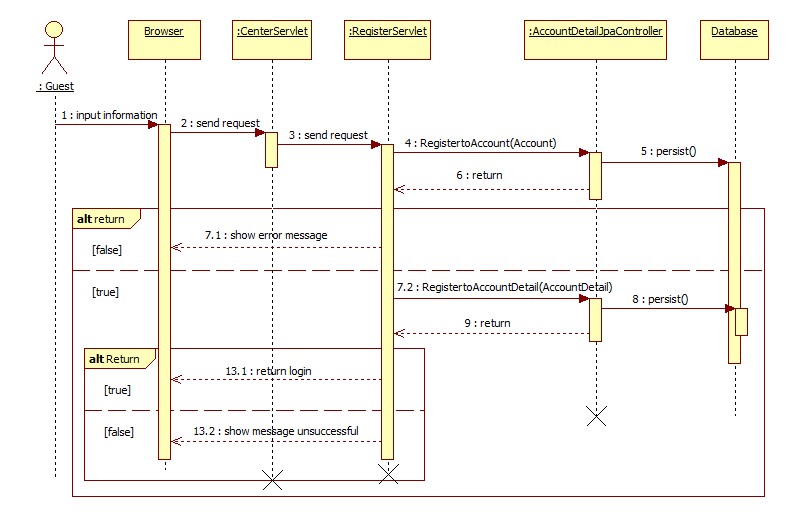


Figure 56: Sequence diagram - <Guest> Register

#### Mobile Aplication

##### <Staff>

###### Train online

Summary: this diagram show process of Train online

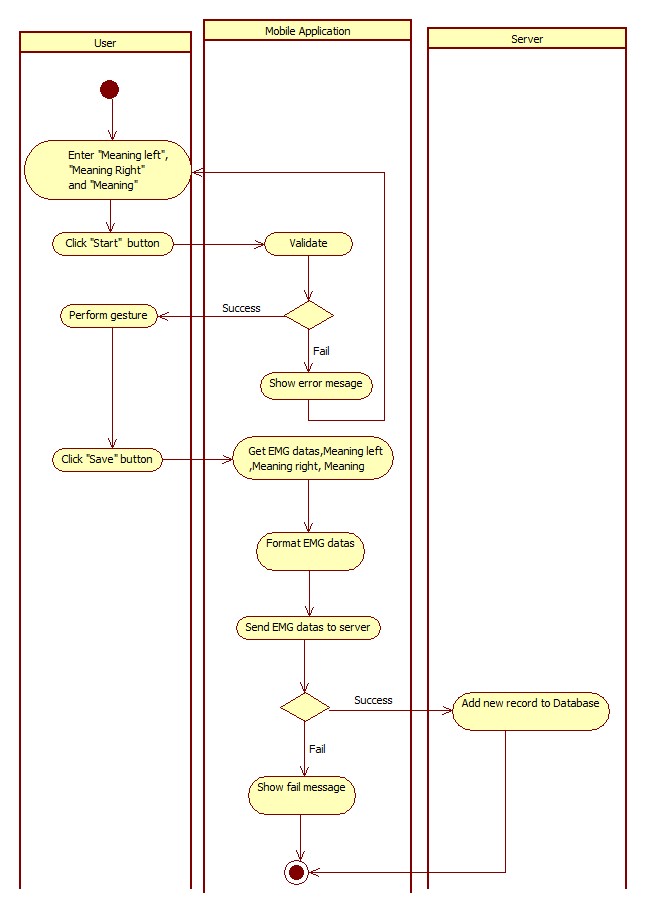


Figure 57: Activity diagram - <Staff> Train online

##### <User>

###### Connect Myo armbands

Summary: this diagram show process of connect Myo armbands

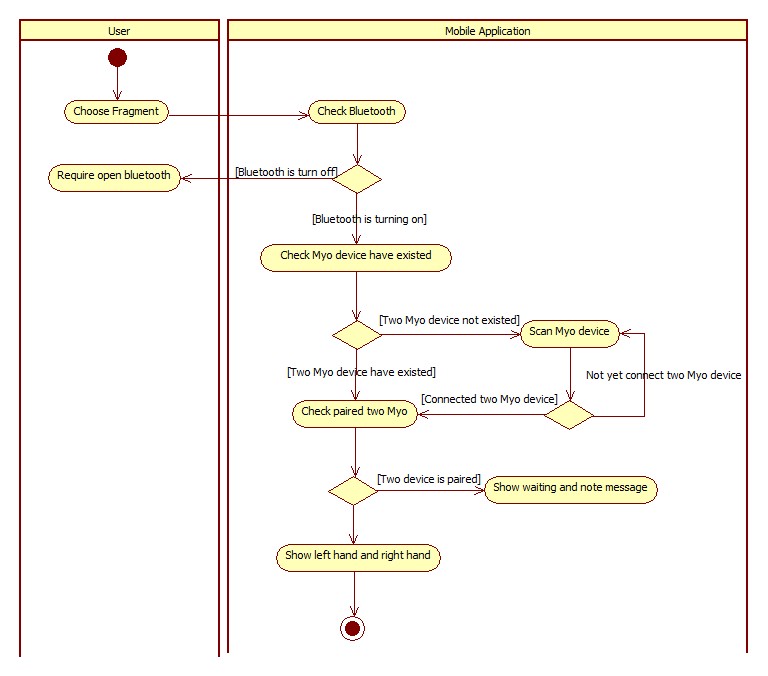


Figure 58: Activity diagram - <User> Connect Myo armbands

###### Translate online

Summary: this diagram show process of translate online

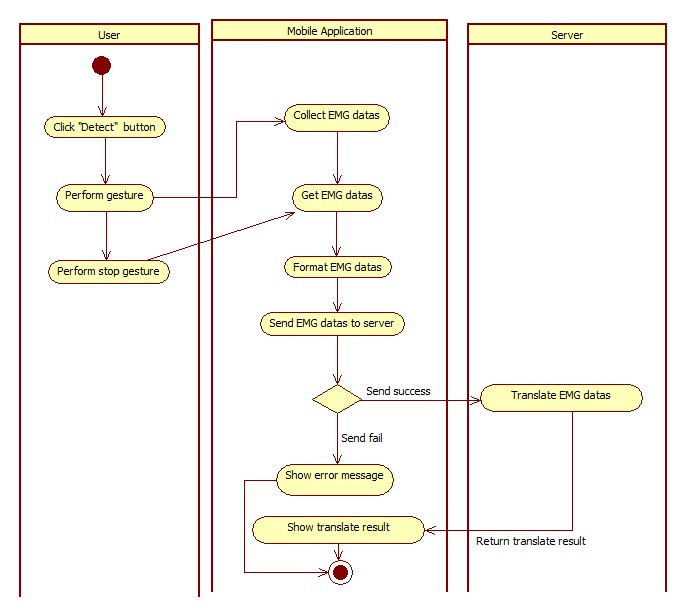


Figure 59: Activity diagram - <User> Translate online

##### <Premium User>

###### Translate offline

Summary: this diagram show process of translate offline

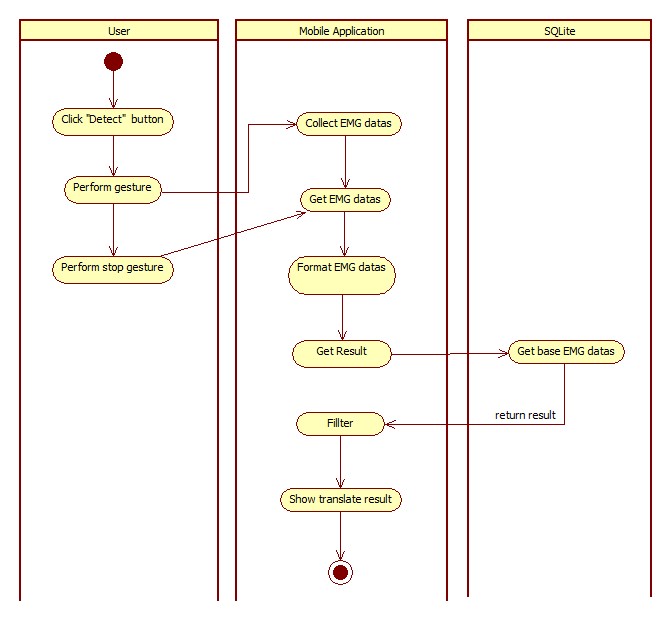


Figure 60: Activity diagram - <Premium User> Translate offline

###### Train offline

Summary: this diagram show process of train offline

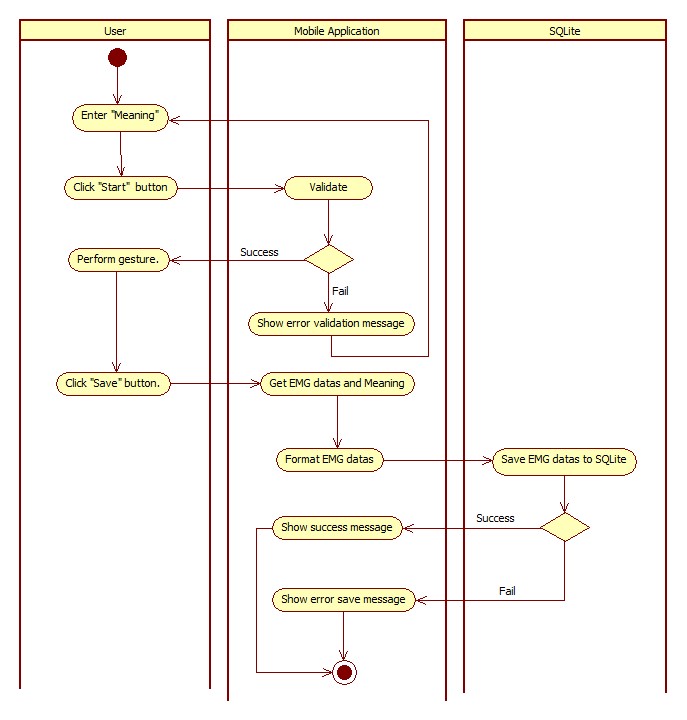


Figure 61: Activity diagram - <Premium User> Train offline

## Interface

### Component Interface

#### Web Services Interface

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Signature | Description | Input | Output | Output Format | Exception |
| public String doTranslate(String inputData) | Translate EMG code into text | inputData: Json String | Json String List of result | String | JsonProcessingException  NoResultException |
| Contractlean if the update is successntract public String doTrain(@QueryParam("meaning") String meaning,  @QueryParam("leftData") String leftData,  @QueryParam("rightData") String rightData,  @QueryParam("leftMeaning") String leftMeaning,  @QueryParam("rightMeaning") String rightMeaning)public String doTrain(@QueryParam("meaning") String meaning,  @QueryParam("leftData") String leftData,  @QueryParam("rightData") String rightData,  @QueryParam("leftMeaning") String leftMeaning,  @QueryParam("rightMeaning") String rightMeaning) | Train new guesture for the system | meaning: String  leftData: String  rightData: String  leftMeaning: String  rightMeaning: String | String response | String | N/A |

Table 76: Web Services interface

|  |  |
| --- | --- |
| Exception | Description |
| JsonProcessingException | encountered when processing (parsing, generating) JSON content that are not pure I/O problems |
| NoResultException | Thrown by the persistence provider when getSingleResult() is executed on a query and there is no result to return |

Table 77: Exception description

### Web Application Design

#### Staff interface design

##### Manage user

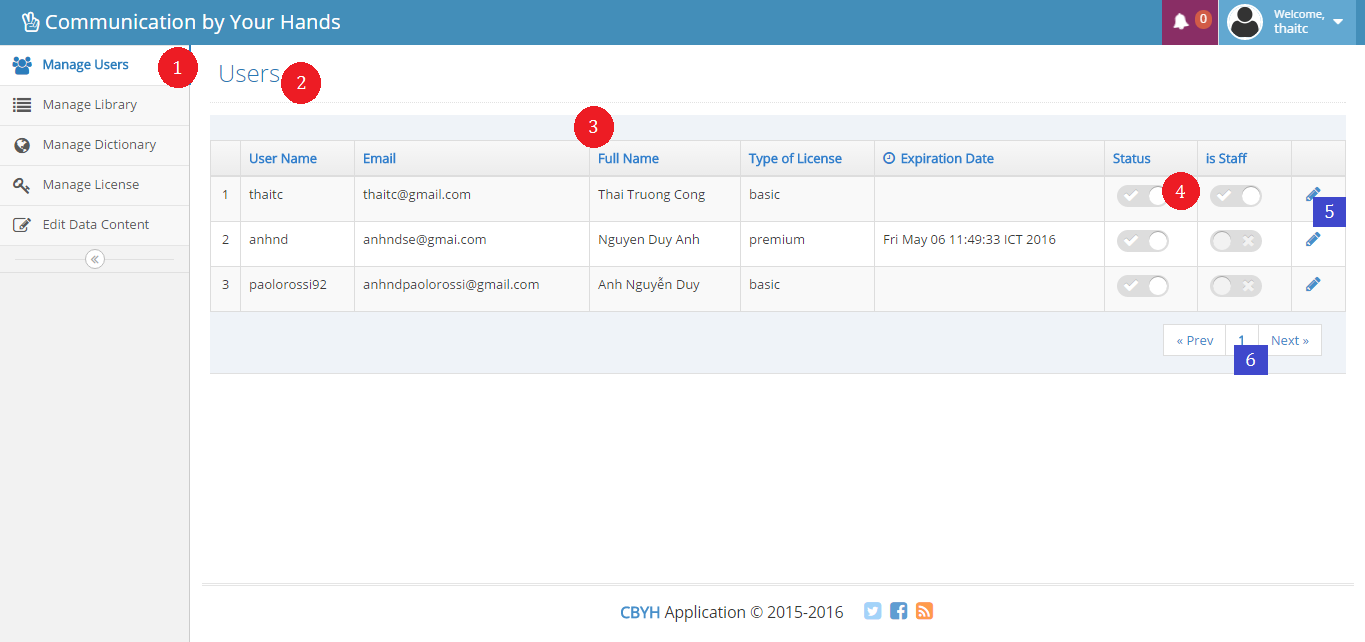


Figure 62: Interface - <Staff> Manage user

**Fields**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| No | Field Name | Description | Read only | Mandatory | Control Type | Data Type | Length |
| 1 | Menu | Navigation bar | Yes | Yes | Menu bar | N/A | N/A |
| 2 | Title | Title of the page | Yes | Yes | Label | N/A | N/A |
| 3 | userTable | Table of all user | Yes | Yes | Table | N/A | N/A |
| 4 | chbStatus | Status of user | Yes | Yes | Checkbox | Boolean | N/A |

Table 78: <Staff> Manage user fields

**Buttons/Hyperlinks**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No** | **Function** | **Description** | **Validation** | **Outcome** |
| **5** | linkUpdateUser | View user detail to update | N/A | Transfer to user update page |
| **6** | linkNextPage | View next page if table | N/A | Transfer to next page |

Table 79: <Staff> Manage user buttons/ hyperlinks

##### Update user



Figure 63: Interface - <Staff> Update user

**Fields**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| No | Field Name | Description | Read only | Mandatory | Control Type | Data Type | Length |
| 1 | Menu | Navigation bar | Yes | Yes | Menu bar | N/A | N/A |
| 2 | Title | Title of the page | Yes | Yes | Label | N/A | N/A |
| 3 | txtDetailId | Fill detail Id | Yes | Yes | Textbox | String | N/A |
| 4 | txtUsername | Fill username | Yes | Yes | Textbox | String | 6 - 20 |
| 5 | txtEmail | Fill email | No | Yes | Textbox | String | 10 - 254 |
| 6 | txtFullname | Fill full name | No | Yes | Textbox | String | 10 - 50 |
| 7 | txtPhone | Fill phone | No | Yes | Textbox | String | 10 - 12 |
| 8 | chbIsStaff | Authorize user is staff or not | No | Yes | Checkbox | Boolean | N/A |
| 9 | ddlLicenseType | Select license type | No | Yes | Select | String | N/A |
| 10 | dtExpiredDate | Fill expiration date | Yes | Yes | Textbox | Date | N/A |
| 11 | chbStatus | Status of user | No | Yes | Checkbox | Boolean | N/A |

Table 80: <Staff> Update user fields

**Buttons/Hyperlinks**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No** | **Function** | **Description** | **Validation** | **Outcome** |
| **12** | btnUpdateUser | Update user | N/A | Transfer to manage user page |
| **13** | btnReset | Reset fill | N/A | N/A |

Table 81: <Staff> Update user buttons/ hyperlinks

#### Guest interface design

##### Register

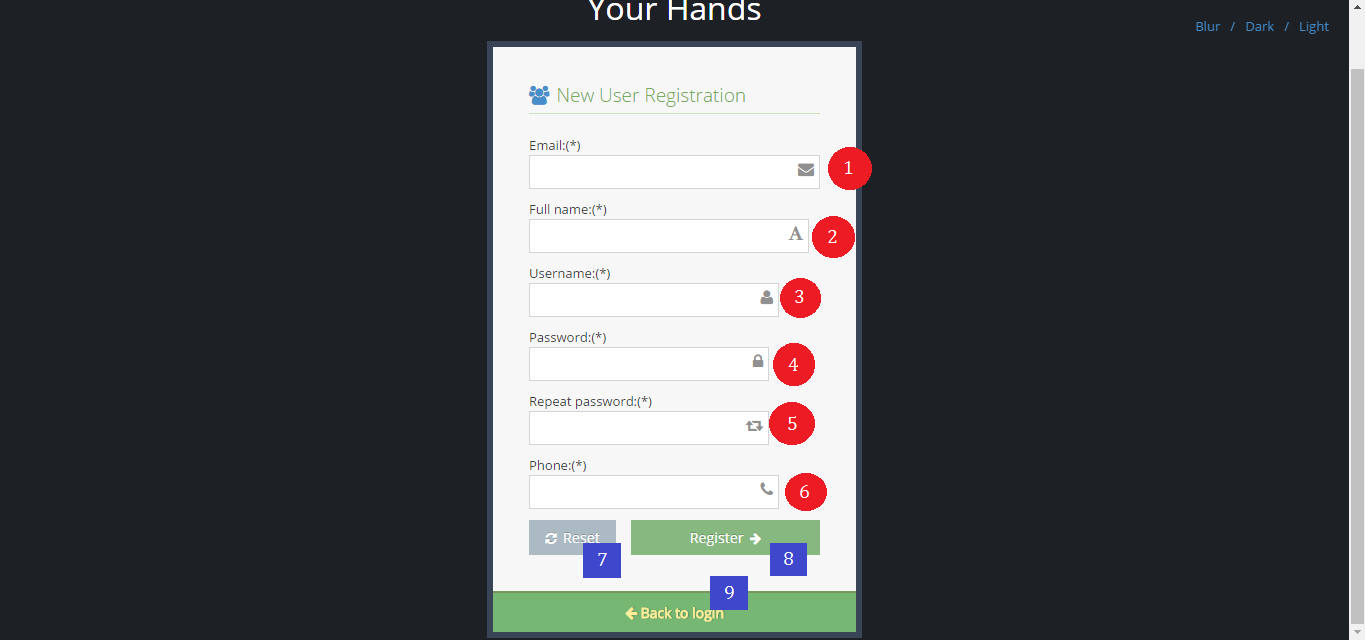


Figure 64: Interface - <Guest> Register

**Fields**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| No | Field Name | Description | Read only | Mandatory | Control Type | Data Type | Length |
| 1 | txtEmail | Fill email | No | Yes | Textbox | String | 10 -254 |
| 2 | txtFullname | Fill full name | No | Yes | Textbox | String | 10 - 50 |
| 3 | txtUsername | Fill username | No | Yes | Textbox | String | 6 – 20 |
| 4 | txtPassword | Fill password | No | Yes | Textbox | String | 6 – 12 |
| 5 | txtRepeatPassword | Fill repeat password | No | Yes | Textbox | String | 6 – 12 |
| 6 | txtPhone | Fill phone | No | Yes | Textbox | String | 10 – 12 |

Table 82: <Guest> Register fields

**Buttons/Hyperlinks**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No** | **Function** | **Description** | **Validation** | **Outcome** |
| **7** | btnReset | Reset fill | N/A | N/A |
| **8** | btnRegister | Register new account | N/A | Transfer to login page |
| **9** | linkToLogin | View login page | N/A | Transfer to login page |

Table 83: <Guest> Register buttons/ hyperlinks

#### User interface design

##### Buy license

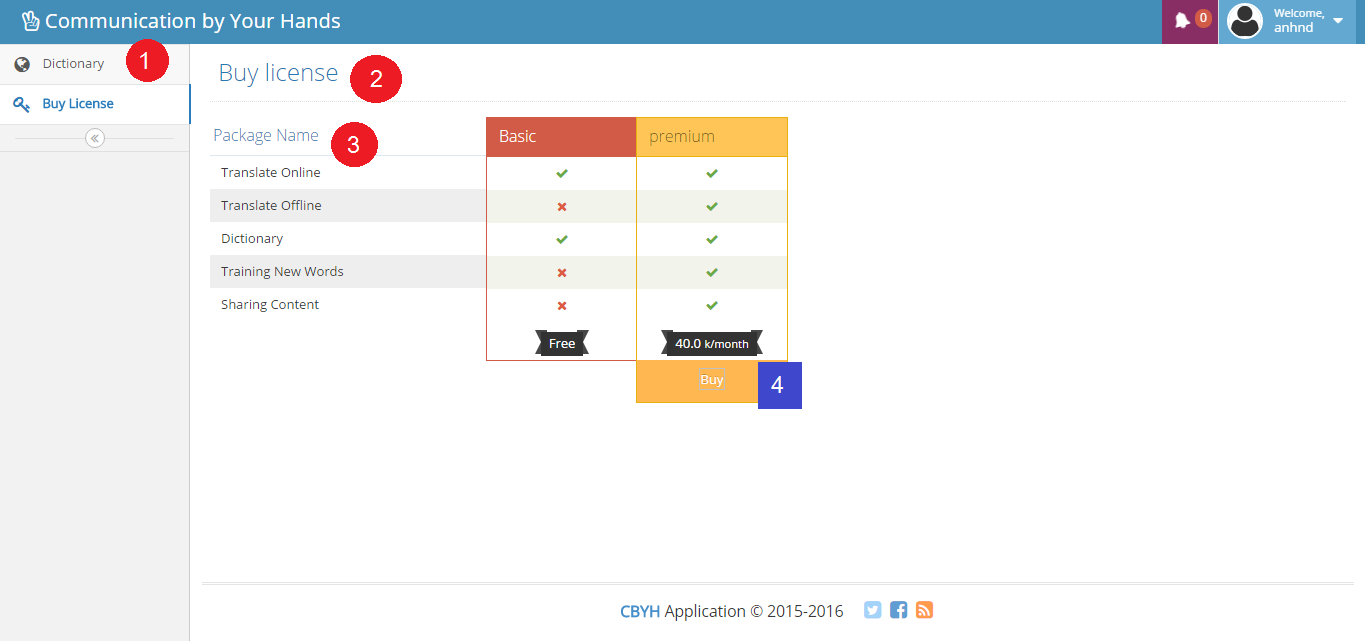


Figure 65: Interface - <User> Buy license

**Fields**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| No | Field Name | Description | Read only | Mandatory | Control Type | Data Type | Length |
| 1 | Menu | Navigation bar | Yes | Yes | Menu bar | N/A | N/A |
| 2 | Title | Title of the page | Yes | Yes | Label | N/A | N/A |
| 3 | lbDescription | Function description | Yes | Yes | Label | N/A | N/A |

Table 84: <User> Buy license fields

**Buttons/Hyperlinks**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No** | **Function** | **Description** | **Validation** | **Outcome** |
| **4** | btnBuy | Buy license with PayPal | N/A | Transfer to PayPal page |

Table 85: <User> Buy license buttons/ hyperlinks

##### Search instruction

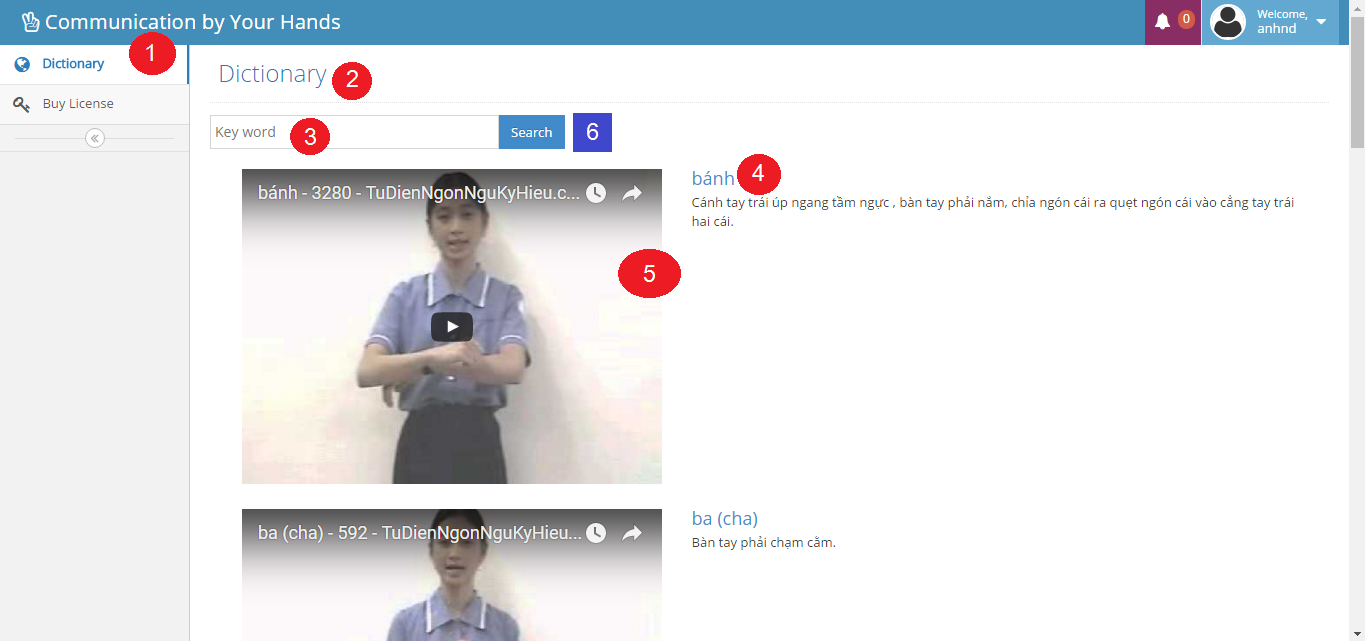


Figure 66: Interface - <User> Search instruction

**Fields**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| No | Field Name | Description | Read only | Mandatory | Control Type | Data Type | Length |
| 1 | Menu | Navigation bar | Yes | Yes | Menu bar | N/A | N/A |
| 2 | Title | Title of the page | Yes | Yes | Label | N/A | N/A |
| 3 | txtKeyword | Fill keyword | No | Yes | Textbox | String | N/A |
| 4 | lbDescription | Description of keyword | Yes | Yes | Label | N/A | N/A |
| 5 | txtVideoURL | Embedded video URL | Yes | Yes | Link | N/A | N/A |

Table 86: <User> Search instruction fields

**Buttons/Hyperlinks**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No** | **Function** | **Description** | **Validation** | **Outcome** |
| **6** | btnSearch | Search by keyword | N/A | N/A |

Table 87: <User> Search instruction buttons/ hyperlinks

##### Edit user profile

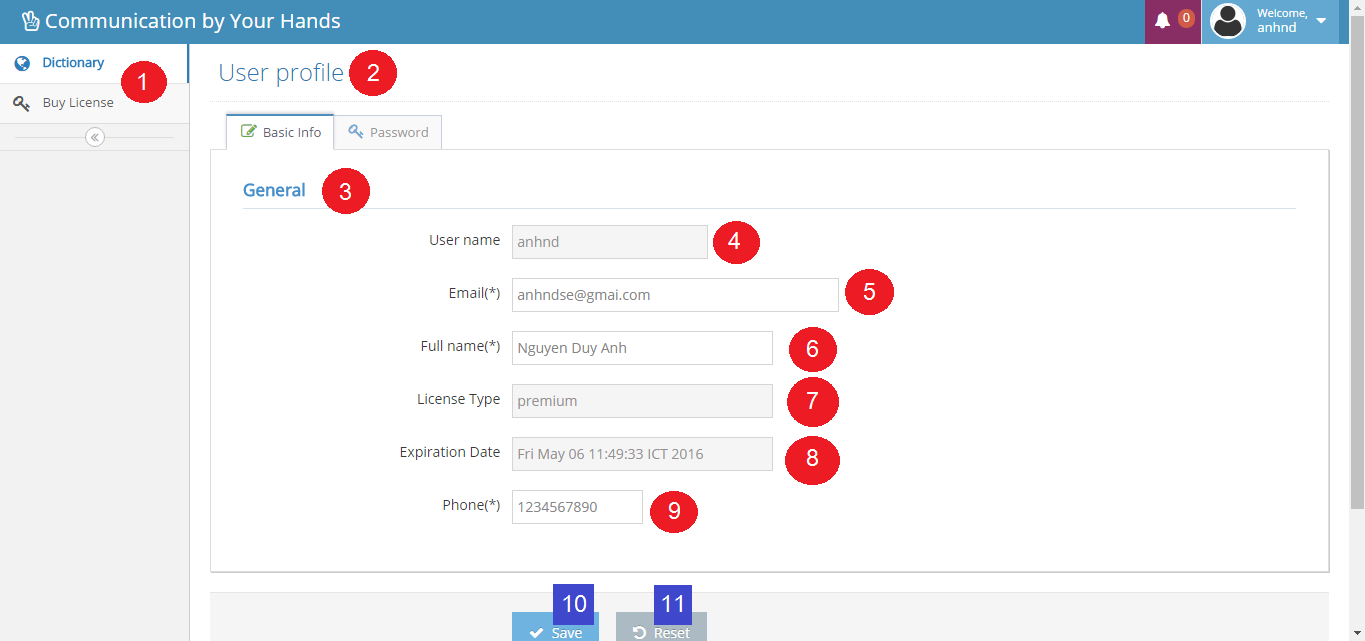


Figure 67: Interface - <User> Edit user profile

**Fields**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| No | Field Name | Description | Read only | Mandatory | Control Type | Data Type | Length |
| 1 | Menu | Navigation bar | Yes | Yes | Menu bar | N/A | N/A |
| 2 | Title | Title of the page | Yes | Yes | Label | N/A | N/A |
| 3 | Title | Title of the page | Yes | Yes | Label | N/A | N/A |
| 4 | txtUsername | Fill username | Yes | Yes | Textbox | String | 6 – 20 |
| 5 | txtEmail | Fill email | No | Yes | Textbox | String | 10 – 254 |
| 6 | txtFullname | Fill full name | No | Yes | Textbox | String | 10 – 50 |
| 7 | txtLicenseType | Fill license type | Yes | Yes | Textbox | String | N/A |
| 8 | txtExpiredDate | Fill expired date | Yes | Yes | Textbox | String | N/A |
| 9 | txtPhone | Fill phone | No | Yes | Textbox | String | 10 – 12 |

Table 88: <User> Edit user fields

**Buttons/Hyperlinks**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No** | **Function** | **Description** | **Validation** | **Outcome** |
| **10** | btnSave | Save updated fill | N/A | Transfer to dictionary page |
| **11** | btnReset | Reset all fill | N/A | N/A |

Table 89: <User> Edit user buttons/hyperlinks

### Mobile Application Design

#### Staff interface design

##### Train online

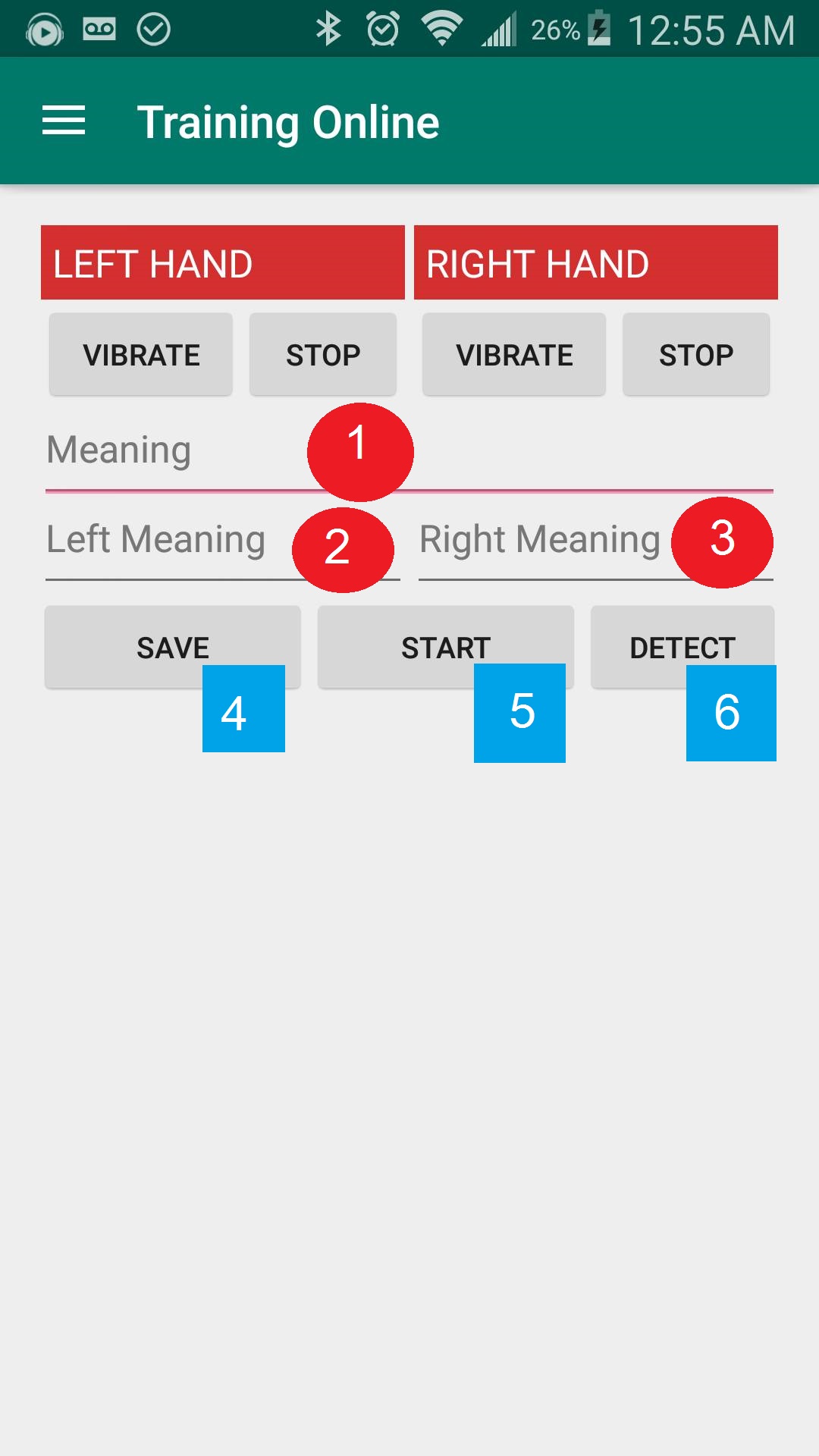


Figure 68: Interface - <Staff> Train online

**Fields**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| No | Field Name | Description | Read only | Mandatory | Control Type | Data Type | Length |
| 1 | Meaning | Fill meaning | No | Yes | Textbox | String | 50 |
| 2 | Left Meaning | Fill left meaning | No | Yes | Textbox | String | 50 |
| 3 | Right Meaning | Fill right meaning | No | Yes | Textbox | String | 50 |

Table 90: <Staff> Train online fields

**Buttons/Hyperlinks**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No** | **Function** | **Description** | **Validation** | **Outcome** |
| **4** | btnSave | Save new guesture | N/A | N/A |
| **5** | btnStart | Start collecting emg | N/A | N/A |
| **6** | btnDetect | Switch to translate screen | N/A | Translate screen |

Table 91: <Staff> Train online buttons/hyperlinks

#### User interface design

##### Login

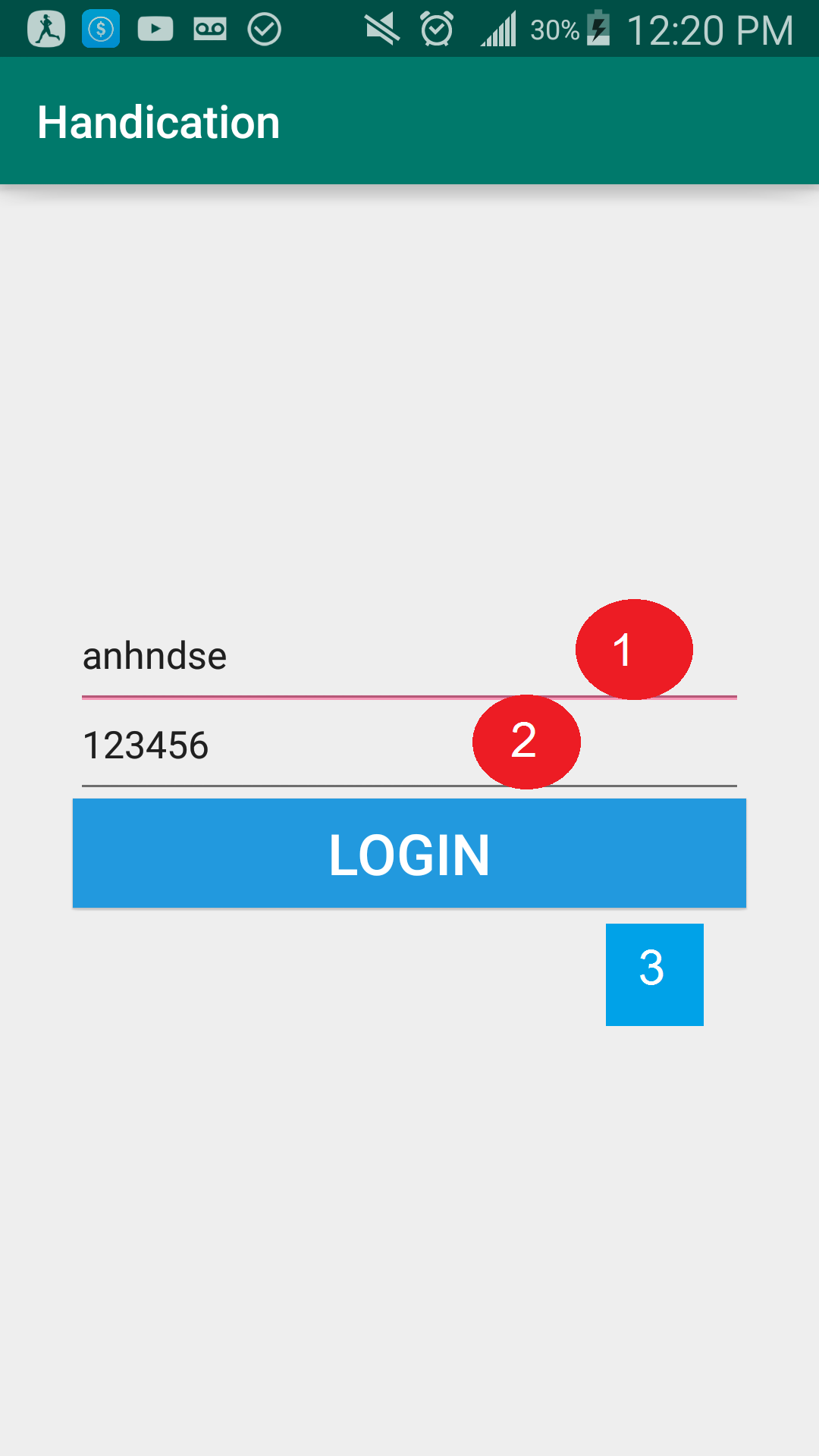


Figure 69: Interface - <User> Login

**Fields**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| No | Field Name | Description | Read only | Mandatory | Control Type | Data Type | Length |
| 1 | Username | Fill username | No | Yes | Textbox | String | 6-20 |
| 2 | Password | Fill left password | No | Yes | Textbox | String | 6-12 |

Table 92: <User> Login fields

**Buttons/Hyperlinks**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No** | **Function** | **Description** | **Validation** | **Outcome** |
| **3** | btnLogin | Login to app | N/A | Home screen |

Table 93: <User> Login buttons/hyperlinls

##### Translate

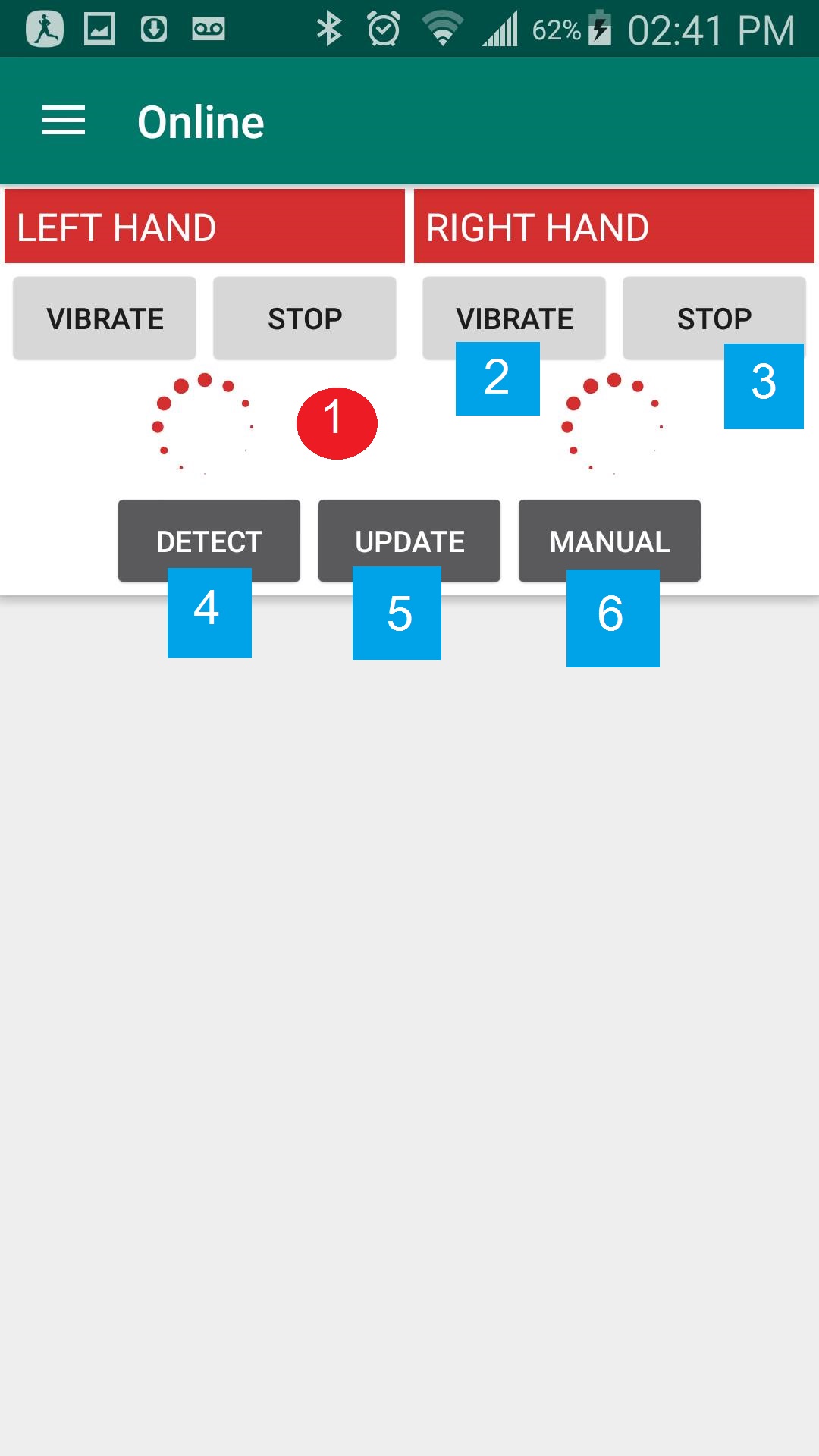


Figure 70: Interface - <User> Translate – Translating

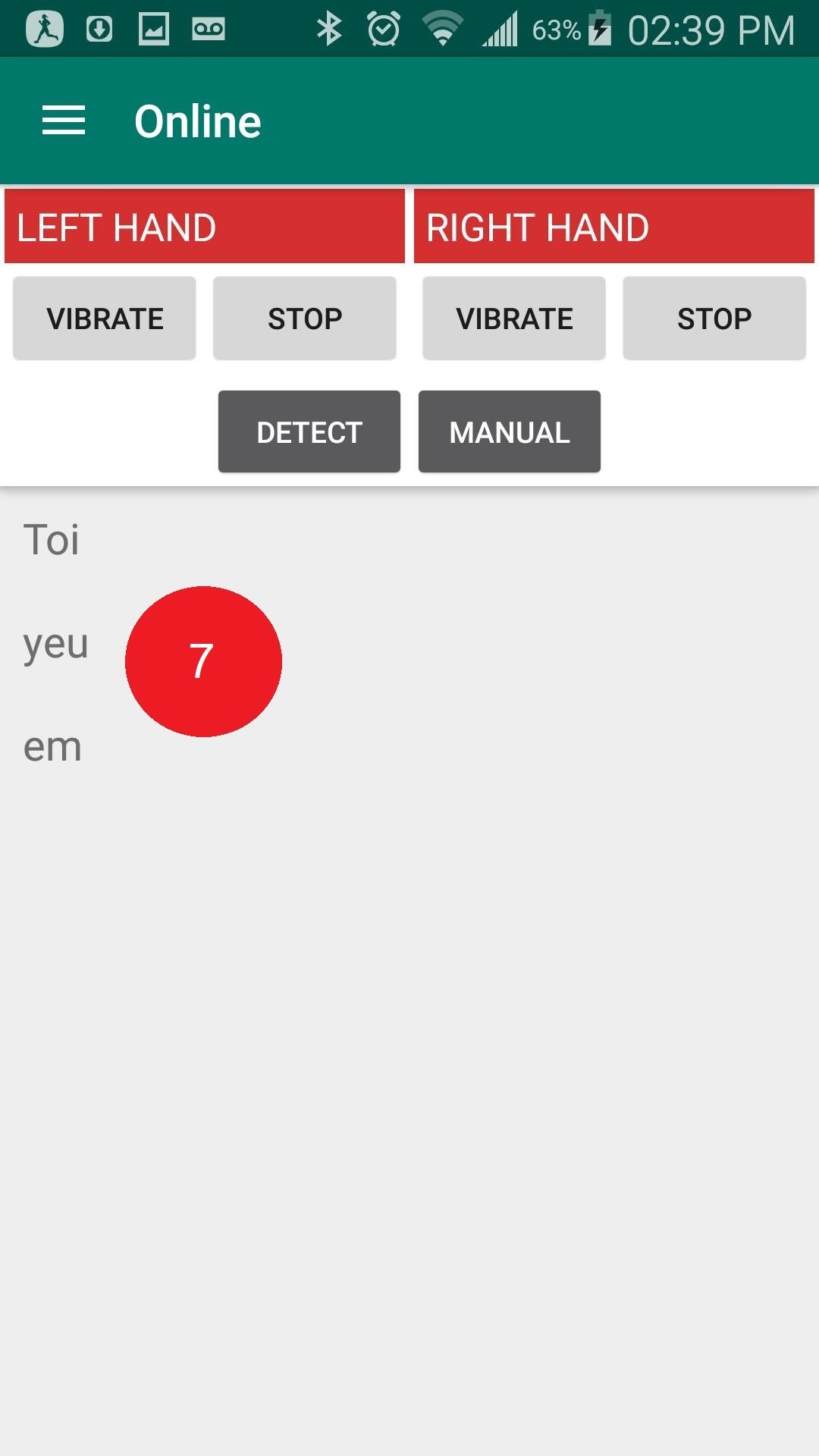


Figure 71: Interface - <User> Translate – Translated

**Fields**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| No | Field Name | Description | Read only | Mandatory | Control Type | Data Type | Length |
| 1 | Process dailog | Processing signal | Yes | Yes | Dailog | N/A | N/A |
| 7 | Result | Translate result | Yes | Yes | List | String | N/A |

Table 94: <User> Translate fields

**Buttons/Hyperlinks**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No** | **Function** | **Description** | **Validation** | **Outcome** |
| **2** | btnVibrate | Active vibrating for the armband | N/A | N/A |
| **3** | btnStop | Stop collect EMG data | N/A | N/A |
| **4** | btnDetect | Start translate automatically | N/A | N/A |
| **5** | btnUpdate | Send update meaning | N/A | N/A |
| **6** | btnManual | Start translate manually | N/A | N/A |

Table 94: <User> Translate buttons/hyperlinks

#### Premium user interface design

##### Train offline

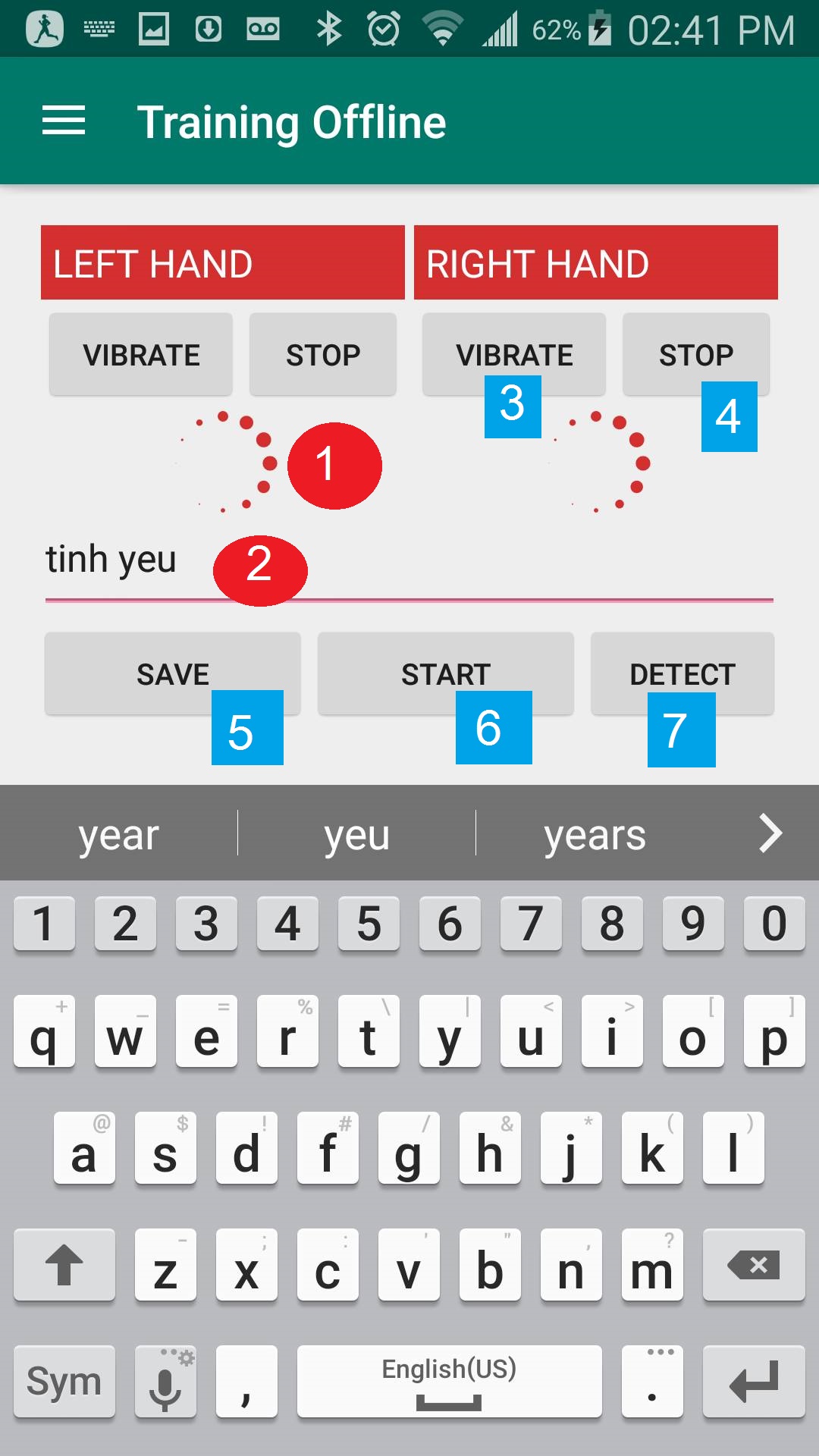


Figure 72: Interface - <Premium user> Train offline

**Fields**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| No | Field Name | Description | Read only | Mandatory | Control Type | Data Type | Length |
| 1 | Process dailog | Processing signal | Yes | Yes | Dailog | N/A | N/A |
| 2 | Meaning | Fill meaning | Yes | Yes | Textbox | String | 50 |

Table 96: <Premium user> Train offline fields

**Buttons/Hyperlinks**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No** | **Function** | **Description** | **Validation** | **Outcome** |
| **3** | btnVibrate | Active vibrating for the armband | N/A | N/A |
| **4** | btnStop | Stop collect EMG data | N/A | N/A |
| **5** | btnSave | Save new guesture | N/A | N/A |
| **6** | btnStart | Start collecting emg | N/A | N/A |
| **7** | btnDetect | Switch to translate screen | N/A | Translate screen |

Table 97: <Premium user> Train offline Buttons/Hyperlinks

## Database design

### Entity relationship diagram

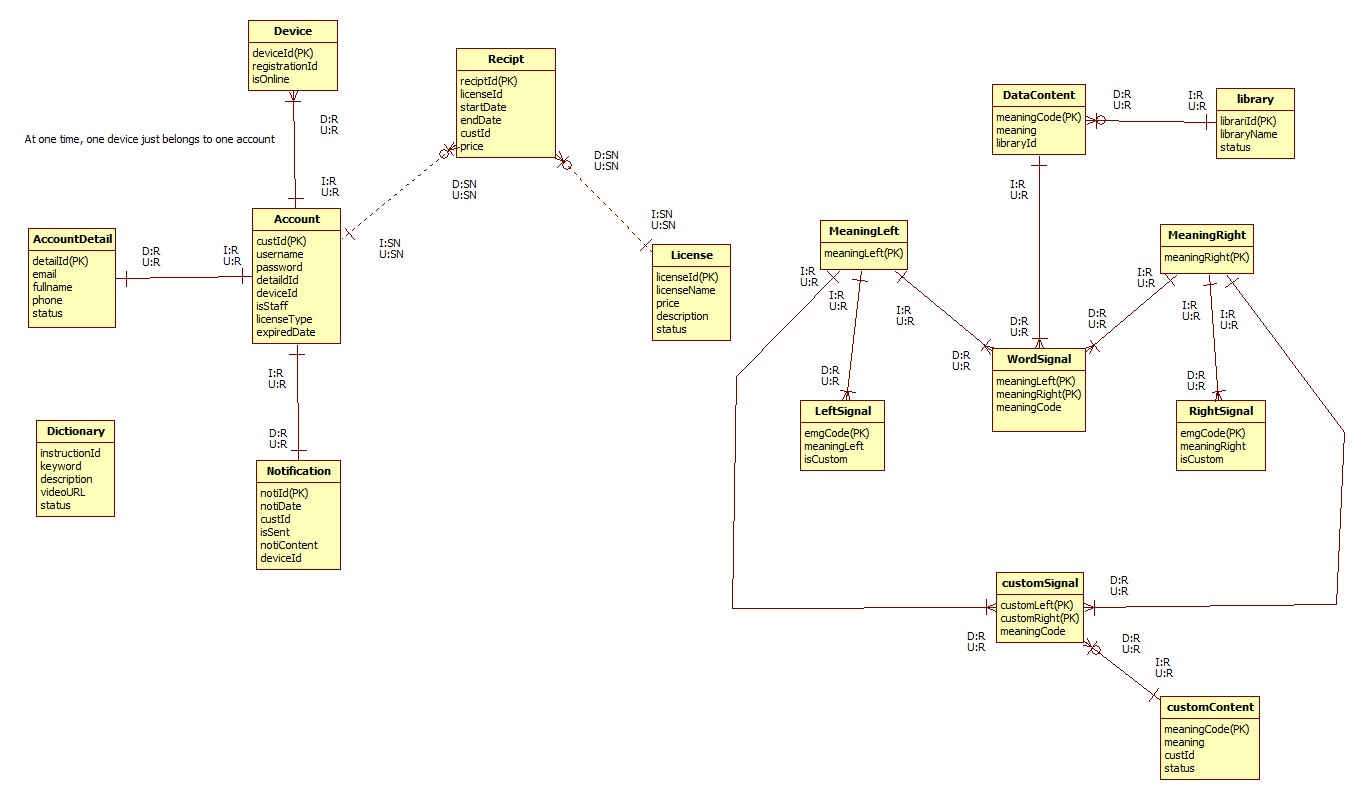


Figure 73: Entity relationship diagram

### Entity dictionary

|  |  |
| --- | --- |
| Entity Data Dictionary: describe content of all entities | |
| Entity name | **Description** |
| Account | Contain the account information |
| AccountDetail | Contain the account detail information |
| Device | Contain the device information |
| Notificaton | Contain the notification information |
| License | Contain the license information |
| Recipt | Contain the recipt information |
| Dictionary | Contain the dictionary information |
| MeaningLeft | Contain the meaning left information |
| MeaningRight | Contain the meaning right information |
| LeftSignal | Contain the left signal information |
| RightSignal | Contain the right signal information |
| WordSignal | Contain the word signal information |
| DataContent | Contain the data content information |
| CustomSignal | Contain the custom signal information |
| CustomContent | Contain the custom content information |
| Library | Contain the library information |

Table 98: Entity dictionary

## Algorithms

### Standardize EMG data (Android)

#### Definition

Standardize EMG data is the method that standardize the input EMG, actually, it is a filter that comibine two next EMG datas into one that define the characteristics of a guesture.

#### Define problem

Currently, the official SDK of MYO armband for android does not support get EMG data from the MYO armband, what the mobile receive are just raw EMG data (numbers in byte type). Those numbers not define any characteristic of EMG data.

#### Solution

We create a filter on mobile application when receiving datas from MYO armbands that:

+ select two adjacent EMG data

+Absolutize those datas

+For each order number in two EMG data, the filter will choose the larger to make a number with the same order for a new EMG data.

#### Example

Two EMG data:

+ EMG1: -4, 5, 7, -6, 6, 7, 2, -10

+ EMG 2: 1, -4, 10, 4, 5, 7, 8, 9

Absolutize:

+ EMG1: -4, 5, 7, -6, 6, 7, 2, -10 => EMG1: 4, 5, 7, 6, 6, 7, 2, 10

+ EMG 2: 1, -4, 10, 4, 5, 7, 8, 9 => EMG 2: 1, 4, 10, 4, 5, 7, 8, 9

The new EMG created:

+ EMG: 4, 5, 10, 6, 6, 7, 8, 10

### Matching

#### Definition

Matching is the method that calculate the Distance number

Distance number is the number that show how related between two EMG datas bases on these two characteristics:

+ Distance between 2 points (each EMG data is seem as a point in eight dimension space)

+ The linear rate of 3 point: The Origin and the two EMG datas.

#### Define problem

To specify a guesture that already defined in DataBase, we need to specify the relation between the base EMG data in DataBase and the input EMG data.

#### Solution

Each EMG data is presented by a series of eight numbers, so we define each EMG data is a point in eight dimensions space.

To know the relation between two EMG datas, we calculate the distance number. We call point A is the input EMG data. Point B is the base EMG data in DataBase. Point O is the Origin.

We calculate the distance between A and B call AB.

We calculate the distance between A and O call OA.

We calculate the distance between B and O call OB.

Then we use the formular below to calculate the distance number:

In developing process, we found a threshold number that equals 0.01, if the distance number is less than 0.01, we seem the A is the most related point with B.

#### Example

Calculate the matching of two EMG datas:

A: 5, 5, 5, 5, 6, 6, 6, 6

B: 3, 4, 5, 3, 5, 6, 7, 8

O: 0, 0, 0, 0, 0, 0, 0, 0

Calculate distance AB:

AB =

Calculate distance OA:

OA =

Calculate distance OB:

OB =

CalculateDistance(A,B) = =

#### Flowchart

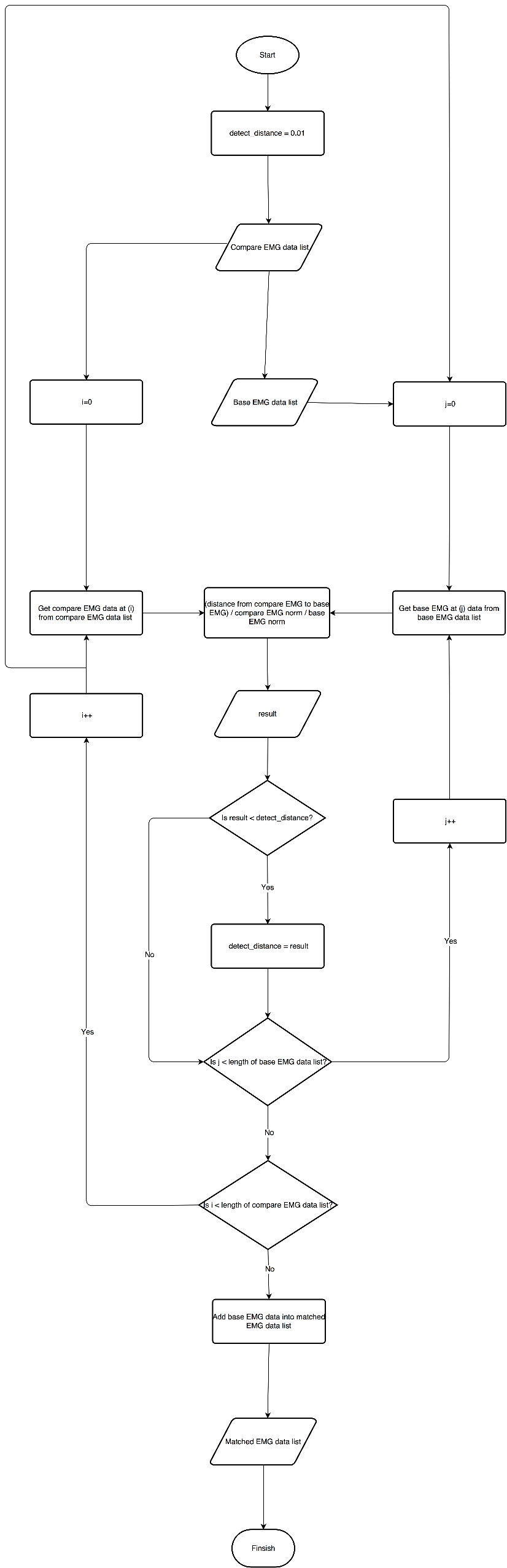


Figure 74: Matching workflow

### Detect

#### Definition

After find the EMG data that relates with the EMG in DataBase. Detect is the function that find the meaning of each hand to find the guesture meaning.

#### Define problem

After do the Matching method, we can find the most related EMG data input with the base EMG data in database of each hand. Therefore, we can find the meaning of each hands. However, two MYO armbands is working separately, we must find the general meaning of both hands to find the meaning of the guesture.

#### Solution

After find the related EMG data, we can easily find the meaning code of each hands. After that, we can find the general meaning code from meaning code from each hands. If there is the result, we will add the result to a result list. Incase of we just can find the meaning code of one hand; we will add it to the result list. The result list will contain the meanings of translated EMG datas

#### Flowchart

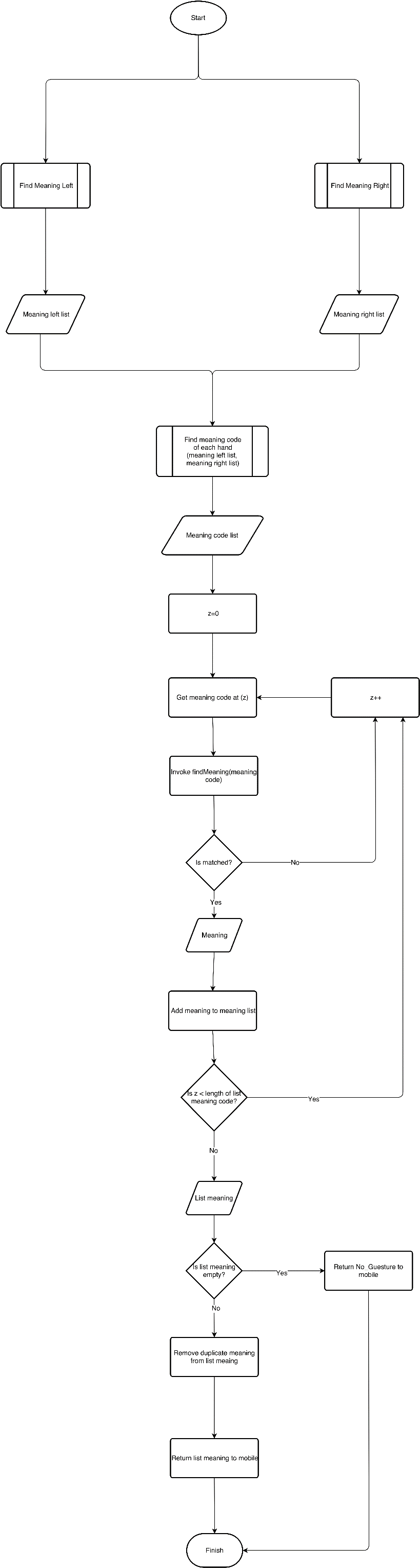
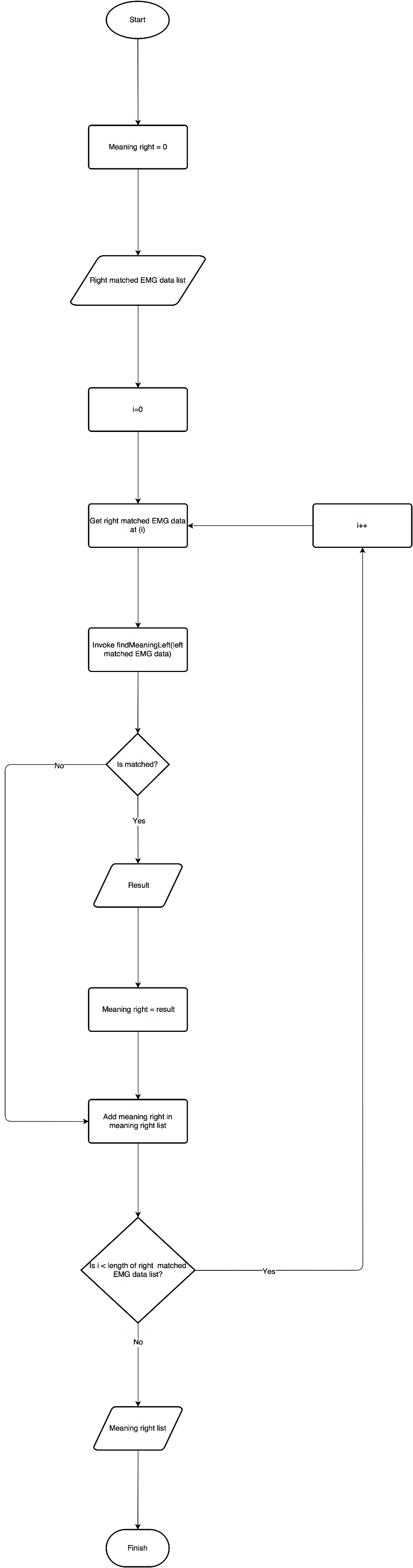


Figure 75: Detect workflow

  
Figure 76: Find meaning right workflow

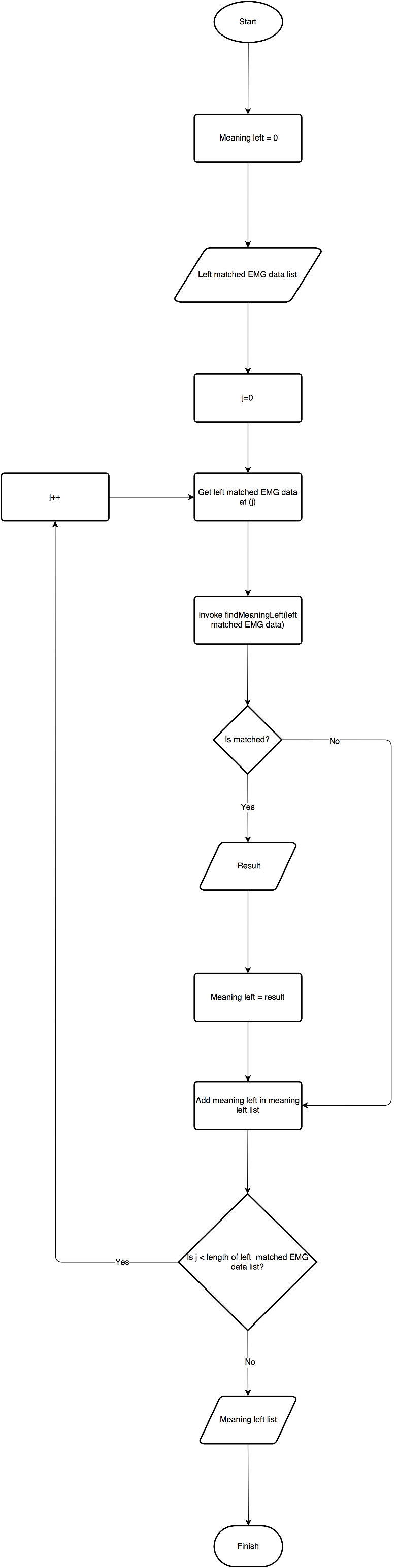


Figure 77: Find meaning left workflow

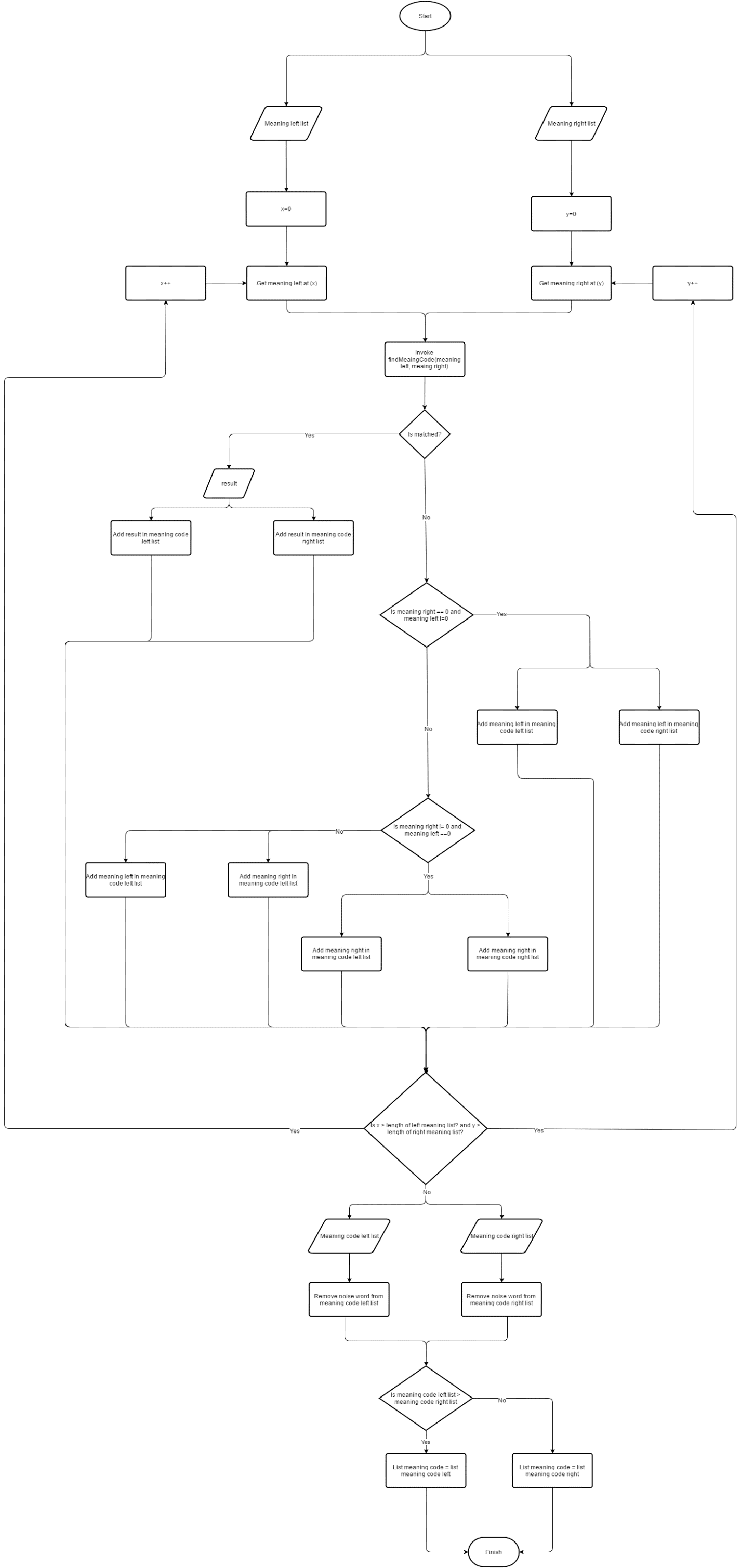


Figure 78: Find meaning of each hand workflow