**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 (droped out) |
| **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](#_Toc448578463)

[List of Tables 5](#_Toc448578464)

[List of Figure 7](#_Toc448578465)

[A. Report No. 1 Introduction 11](#_Toc448578466)

[1. Project Information 11](#_Toc448578467)

[2. Introduction 11](#_Toc448578468)

[3. Current Situation 11](#_Toc448578469)

[4. Problem Definition 12](#_Toc448578470)

[5. Proposed Solution 12](#_Toc448578471)

[5.1 Feature functions 13](#_Toc448578472)

[5.2 Advantages and disadvantages 13](#_Toc448578473)

[6. Functional Requirements 14](#_Toc448578474)

[7. Role and Responsibility 14](#_Toc448578475)

[B. Report No.2 Software Project Management Plan 15](#_Toc448578476)

[1. Problem Definition 15](#_Toc448578477)

[1.1 Name of this Capstone Project 15](#_Toc448578478)

[1.2 Problem Abstract 15](#_Toc448578479)

[1.3 Project Overview 16](#_Toc448578480)

[2. Project organization 21](#_Toc448578481)

[2.1 Software Process Model 21](#_Toc448578482)

[2.2 Roles and responsibilities 23](#_Toc448578483)

[2.3 Tools and Techniques 25](#_Toc448578484)

[3. Project Management Plan 25](#_Toc448578485)

[3.1 Software development life cycle 25](#_Toc448578486)

[3.2 Phase Detail 27](#_Toc448578487)

[3.3 Task sheet 29](#_Toc448578488)

[3.4 All Meeting Minutes 30](#_Toc448578495)

[4. Coding Convention 30](#_Toc448578496)

[C. Report No.3 Software Requirement Specification 31](#_Toc448578497)

[1. User Requirement Specification 31](#_Toc448578498)

[1.1 Guest Requirement 31](#_Toc448578499)

[1.2 Staff Requirement 31](#_Toc448578500)

[1.3 User Requirement 31](#_Toc448578501)

[1.4 Premium user Requirement 31](#_Toc448578502)

[1.5 Authenticated user Requirement 31](#_Toc448578503)

[1.6 Scheduler Requirement 31](#_Toc448578504)

[2. System Requirement Specification 32](#_Toc448578505)

[2.1 External Interface Requirement 32](#_Toc448578506)

[2.2 System Overview Use Case 33](#_Toc448578507)

[2.3 List of Use Case 34](#_Toc448578508)

[3. Software system attribute 61](#_Toc448578509)

[3.1 Usability 61](#_Toc448578510)

[3.2 Reliability 61](#_Toc448578511)

[3.3 Availability 61](#_Toc448578512)

[3.4 Security 62](#_Toc448578513)

[3.5 Maintainability 62](#_Toc448578514)

[3.6 Portability 62](#_Toc448578515)

[3.7 Performance 62](#_Toc448578516)

[4. Conceptual diagram 63](#_Toc448578517)

[D. Report No.4 Software Design Description 65](#_Toc448578518)

[1. Design Overview 65](#_Toc448578519)

[2. System Architectural Design 66](#_Toc448578520)

[2.1 Web application architecture description 66](#_Toc448578521)

[2.2 Mobile application architecture description 66](#_Toc448578522)

[3. Component Diagram 67](#_Toc448578523)

[4. Detail Description 69](#_Toc448578524)

[4.1 Class Diagram 69](#_Toc448578525)

[4.2 Class Diagram Explanation 71](#_Toc448578526)

[4.3 Interactive Diagram 76](#_Toc448578527)

[5. Interface 86](#_Toc448578528)

[5.1 Component Interface 86](#_Toc448578529)

[5.2 Web Application Design 89](#_Toc448578530)

[5.3 Mobile Application Design 93](#_Toc448578531)

[6. Database design 98](#_Toc448578532)

[6.1 Entity relationship diagram 98](#_Toc448578533)

[6.2 Entity dictionary 100](#_Toc448578534)

[7. Algorithms 100](#_Toc448578535)

[7.1 Standardize EMG data (Android) 100](#_Toc448578536)

[7.2 Matching 101](#_Toc448578537)

[7.3 Detect 105](#_Toc448578538)

[7.4 Train 110](#_Toc448578539)

[7.5 Standardize EMG data before input to Database (Server) 112](#_Toc448578540)

[E. Report No.5 System implement and Test 113](#_Toc448578541)

[1. Introduction 113](#_Toc448578542)

[1.1 Overview 113](#_Toc448578543)

[1.2 Test Approach 113](#_Toc448578544)

[2. Database relationship diagram 113](#_Toc448578545)

[2.1 Physical diagram 113](#_Toc448578546)

[2.2 Data dictionary 115](#_Toc448578547)

[3. Performance measures 118](#_Toc448578548)

[3.1 Mobile Application API load speed 118](#_Toc448578549)

[4. Test plan 120](#_Toc448578550)

[4.1 Features to be tested 120](#_Toc448578551)

[4.2 Features not to be tested 121](#_Toc448578552)

[5. System testing test case 121](#_Toc448578553)

[5.1 Communication diagram 121](#_Toc448578554)

[5.2 Test cases 123](#_Toc448578555)

[5.3 Test cases results and statistics 146](#_Toc448578556)

[F. Report No.6 Software user’s manual 150](#_Toc448578557)

[1. Installation guide 150](#_Toc448578558)

[1.1 Setting up environment at server side 150](#_Toc448578559)

[1.2 Web application / web service deployment process 151](#_Toc448578560)

[1.3 Mobile application deployment process 153](#_Toc448578561)

[2. User guide 155](#_Toc448578562)

[2.1 MYO armband 155](#_Toc448578563)

[2.2 Mobile application 155](#_Toc448578564)

[2.3 Web application 166](#_Toc448578565)

[G. Appendix 171](#_Toc448578566)

# List of Tables

[Table 1: Roles and Responsibilities 15](#_Toc448578567)

[Table 2: Hardware Requirement for server/ web development 19](#_Toc448578568)

[Table 3: Hardware Requirement for client (Mobile device) development 19](#_Toc448578569)

[Table 4: Hardware Requirement for client (MYO armband) development 20](#_Toc448578570)

[Table 5: Software requirements for develop web site and web service 20](#_Toc448578571)

[Table 6: Software requirements for develop client application 20](#_Toc448578572)

[Table 7: Roles and Responsibilities Details 24](#_Toc448578573)

[Table 8: Tools and Techniques 25](#_Toc448578574)

[Table 9: Software Development Life Cycle Detail 27](#_Toc448578575)

[Table 10: Phase 1: Specification 28](#_Toc448578576)

[Table 11: Phase 2: Implementation 28](#_Toc448578577)

[Table 12: Phase 3: Validation 29](#_Toc448578578)

[Table 13: Register specification 37](#_Toc448578579)

[Table 14: Login specification 39](#_Toc448578580)

[Table 15: Search specification 40](#_Toc448578581)

[Table 16: Translate (Online) specification 43](#_Toc448578582)

[Table 17: Buy license specification 45](#_Toc448578583)

[Table 18: Edit profile specification 48](#_Toc448578584)

[Table 19: Translate (Offline) specification 50](#_Toc448578585)

[Table 20: Train custom sign specification 52](#_Toc448578586)

[Table 21: Train sign language specification 55](#_Toc448578587)

[Table 22: Logout specification 57](#_Toc448578588)

[Table 23: Create notification specification 59](#_Toc448578589)

[Table 24: Send notification specification 61](#_Toc448578590)

[Table 25: Conceptual diagram data dictionary 64](#_Toc448578591)

[Table 26: Component Dictionary 69](#_Toc448578592)

[Table 27: Class dictionary 71](#_Toc448578593)

[Table 28: Account Attributes 72](#_Toc448578594)

[Table 29: AccountDetail Attributes 72](#_Toc448578595)

[Table 30: Dictionary Attributes 72](#_Toc448578596)

[Table 31: Notification Attributes 73](#_Toc448578597)

[Table 32: License Attributes 73](#_Toc448578598)

[Table 33: Library Attributes 73](#_Toc448578599)

[Table 34: Device Attributes 74](#_Toc448578600)

[Table 35: LeftSignal Attributes 74](#_Toc448578601)

[Table 36: RightSignal Attributes 74](#_Toc448578602)

[Table 37: MeaningLeft Attributes 74](#_Toc448578603)

[Table 38: MeaningRight Attributes 75](#_Toc448578604)

[Table 39: WordSignal Attributes 75](#_Toc448578605)

[Table 40: DataContent Attributes 75](#_Toc448578606)

[Table 41: CustomSignal Attributes 75](#_Toc448578607)

[Table 42: CustomContent Attributes 76](#_Toc448578608)

[Table 43: Receipt Attributes 76](#_Toc448578609)

[Table 44: Web Services interface 88](#_Toc448578610)

[Table 45: Exception description 88](#_Toc448578611)

[Table 46: <Guest> Register fields 89](#_Toc448578612)

[Table 47: <Guest> Register buttons/ hyperlinks 90](#_Toc448578613)

[Table 48: <User> Buy license fields 90](#_Toc448578614)

[Table 49: <User> Buy license buttons/ hyperlinks 91](#_Toc448578615)

[Table 50: <User> Search instruction fields 91](#_Toc448578616)

[Table 51: <User> Search instruction buttons/ hyperlinks 91](#_Toc448578617)

[Table 52: <User> Edit user fields 92](#_Toc448578618)

[Table 53: <User> Edit user buttons/hyperlinks 93](#_Toc448578619)

[Table 54: <Staff> Train online fields 94](#_Toc448578620)

[Table 55: <Staff> Train online buttons/hyperlinks 94](#_Toc448578621)

[Table 56: <User> Login fields 95](#_Toc448578622)

[Table 57: <User> Login buttons/hyperlinls 95](#_Toc448578623)

[Table 58: <User> Translate fields 96](#_Toc448578624)

[Table 59: <User> Translate buttons/hyperlinks 97](#_Toc448578625)

[Table 60: <Premium user> Train offline fields 98](#_Toc448578626)

[Table 61: <Premium user> Train offline Buttons/Hyperlinks 98](#_Toc448578627)

[Table 62: Entity dictionary 100](#_Toc448578628)

[Table 63: Data table dictionary 115](#_Toc448578629)

[Table 64: Data table dictionary 118](#_Toc448578630)

[Table 65: Mobile API load speed client specification 119](#_Toc448578631)

[Table 66: Mobile API load speed test cases 119](#_Toc448578632)

[Table 67: Mobile API load speed test result 120](#_Toc448578633)

[Table 68: Test case <Staff>-<User> Train Online – Translate Online 130](#_Toc448578634)

[Table 69: Test case <Staff> Train Online 132](#_Toc448578635)

[Table 70: Test case <User> Translate Online 135](#_Toc448578636)

[Table 71: Test case <User> Buy license 137](#_Toc448578637)

[Table 72: Test case <Premium User> Train Offline – Translate Offline 140](#_Toc448578638)

[Table 73: Test case <Premium User> Train Offline 141](#_Toc448578639)

[Table 74: Test case <Premium User> Translate Offline 143](#_Toc448578640)

[Table 75: Test case <System> Notify schedule 145](#_Toc448578641)

[Table 76: Test case result statistic 150](#_Toc448578642)

[Table 78: Hardware requirements 150](#_Toc448578643)

[Table 79: Software requirements 151](#_Toc448578644)

# List of Figure

[Figure 1: Evolutionary development Model 27](#_Toc448578703)

[Figure 2: System Overview Use Case 38](#_Toc448578704)

[Figure 3: <Guest> Overview Use Case 39](#_Toc448578705)

[Figure 4: <Guest> Register 39](#_Toc448578706)

[Figure 5: <Guest> Login 43](#_Toc448578707)

[Figure 6: <Guest> Search 44](#_Toc448578708)

[Figure 7: <User> Overview Use Case 46](#_Toc448578709)

[Figure 8: <User> Translate (Online) 46](#_Toc448578710)

[Figure 9: <User> Buy license 49](#_Toc448578711)

[Figure 10: <User> Edit profile 50](#_Toc448578712)

[Figure 11: <Premium User> Overview Use case 53](#_Toc448578713)

[Figure 12: <Premium User> Translate (Offline) 54](#_Toc448578714)

[Figure 13: <Premium User> Train (Offline) 56](#_Toc448578715)

[Figure 14: <Staff> Overview Use Case 58](#_Toc448578716)

[Figure 15: <Staff> Train (Online) 58](#_Toc448578717)

[Figure 16: <Authenticated user> Overview Use Case 61](#_Toc448578718)

[Figure 17: <Authenticated user> Logout 61](#_Toc448578719)

[Figure 18: <Scheduler> Overview Use Case 62](#_Toc448578720)

[Figure 19: <Scheduler> Create notification 63](#_Toc448578721)

[Figure 20: <Scheduler> Send notification 64](#_Toc448578722)

[Figure 21: Conceptual diagram 68](#_Toc448578723)

[Figure 22: System architecture design 71](#_Toc448578724)

[Figure 23: Component Diagram (Server side) 73](#_Toc448578725)

[Figure 24: Component Diagram (Mobile side) 73](#_Toc448578726)

[Figure 25: Class Diagram 75](#_Toc448578727)

[Figure 26: Sequence diagram - <User> Buy license 82](#_Toc448578728)

[Figure 27: Sequence diagram - <User> Update profile 83](#_Toc448578729)

[Figure 28: Sequence diagram - <User> Search 84](#_Toc448578730)

[Figure 29: Sequence diagram - <Guest> Register 85](#_Toc448578731)

[Figure 30: Activity diagram - <Staff> Train online 86](#_Toc448578732)

[Figure 31: Activity diagram - <User> Connect Myo armbands 87](#_Toc448578733)

[Figure 32: Activity diagram - <User> Translate online 88](#_Toc448578734)

[Figure 33: Activity diagram - <Premium User> Translate offline 89](#_Toc448578735)

[Figure 34: Activity diagram - <Premium User> Train offline 90](#_Toc448578736)

[Figure 35: Interface - <Guest> Register 94](#_Toc448578737)

[Figure 36: Interface - <User> Buy license 95](#_Toc448578738)

[Figure 37: Interface - <User> Search instruction 96](#_Toc448578739)

[Figure 38: Interface - <User> Edit user profile 97](#_Toc448578740)

[Figure 39: Interface - <Staff> Train online 98](#_Toc448578741)

[Figure 40: Interface - <User> Login 99](#_Toc448578742)

[Figure 41: Interface - <User> Translate – Translating 100](#_Toc448578743)

[Figure 42: Interface - <User> Translate – Translated 101](#_Toc448578744)

[Figure 43: Interface - <Premium user> Train offline 102](#_Toc448578745)

[Figure 44: Entity relationship diagram 104](#_Toc448578746)

[Figure 45: Matching workflow 109](#_Toc448578747)

[Figure 46: Detect workflow 111](#_Toc448578748)

[Figure 47: Find meaning right workflow 112](#_Toc448578750)

[Figure 48: Find meaning left workflow 113](#_Toc448578752)

[Figure 49: Find meaning of each hand workflow 114](#_Toc448578754)

[Figure 50: Train workflow 116](#_Toc448578755)

[Figure 51: Physical diagram 119](#_Toc448578756)

[Figure 52: Mobile application communication diagram 126](#_Toc448578757)

[Figure 53: Web application communication 127](#_Toc448578758)

[Figure 54: Scheduler communication 127](#_Toc448578760)

*This page is intentionally left blank*

**Definitions, Acronyms, and Abbreviations**

|  |  |
| --- | --- |
| **Name** | **Definition** |
| CBYH | Communication by Your Hands |
| EMG | Electromyography |
| SRS | Software Requirement Specification |
| GUI | Graphic User Interface |
| EMG data format | A list with 8 numbers with byte type |
| BLE | Bluetooth less energy |

# Report No. 1 Introduction

## Project Information

* Project name: **Communication by Yours Hands**
* Project Code: **CBYH**
* Product Type: **Mobile application, register – buy license Website**
* Start Date: **05/01/2016**
* End Date: **16/04/2016**

## Introduction

In communicating, sign language is the best way to communicate between people with deaf and mute. However, it requires 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 find out 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 integrate 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 language”.

Process of using sign language:

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 widely 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, a mobile application and web services with following functions:

### **Feature functions**

* + - Web application (for user):
* Buy license: User can buy license to upgrade their account from basic to premium.
* Register new account: To use main functions the system, user has to register an account.
* Search: User can search to view instruction video that describe how to do the sign language.
* Edit profile: Users can edit information of their profile after registered an account.
  + - Mobile application
* Train Online (For staff): Staff can add new sign language gesture 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 inputted by staff, then send and store them on server.
* Translate Online (for user): 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 armbands and send to server then receive translated data and outputs text or sound with the same meaning.
* Translate Offline (for premium user): Premium user can translate sign language into text or sound without the internet connection required. Right after premium user performs sign language; the application will receive the data from MYO armbands then analyzes locally to find the meaning and outputs it.
* Train Offline (for premium user): Premium user can add new their custom sign language. Right after premium user performs sign language, the application will receive the data from MYO armbands and the meaning of it then analyzes locally and store in storage of mobile device.
  + - Web services:
* TranslateAPI: Web services provide API for mobile application to perform the translate function.
* TrainAPI: Web services provide API for mobile application to perform the train function.

### 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.
* MYO armbands work asynchronously: MYO armband is designed for working individually; there is no official method to make to MYO armbands work as a pair.

## Functional Requirements

Function requirements of the system are listed as below:

* User component:
* Translate (online): User can translate sign language into text or sound with internet connection required
* Buy license: User can buy license to upgrade account into premium user for 30 days each payment.
* Premium user component:
* Translate (offline): Premium user can download the meaning resource to device to translate sign language with no internet connection required.
* Train (offline): Premium user can create new sign gesture and meaning of that sign and store in mobile device storage for personal use.
* Guest:
* Search: Guest can search to view sign language instruction video.
* Register: Guest can register account to use main function of the system
* Staff component:
* Train: Staff can train the standard sign language and meaning for the standard library of the system.
* Scheduler component:
* Notify: System will check for near expired license and send notification for premium user.

## Role and Responsibility

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| No | Full Name | Role | Position | Contact |
| 1 | Kiều Trọng Khánh | Project Manager | Supervisor | [khanhkt@fpt.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) |

Table : Roles and Responsibilities

# Report No.2 Software Project Management Plan

## Problem Definition

### Name of this Capstone Project

-Official name: Communication by Your Hands

-Vietnamese name: Giao tiếp sử dụng tay để diễn đạt

-Abbreviation: CBYH

### Problem Abstract

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.

However, at starting point, there is some problem for developing the system. The current official SDK of the MYO armband does not support get EMG data from the armband. The MYO armband is designed to work individually; there is no way to synchronize two MYO armbands. MYO armband just can connect via Bluetooth 4.0 BLE, the team must study about the connect protocol, which device support, which does not.

Team has to set a plan to approach the MYO armband:

* + Test with available application to understand more about how to use the armband.
  + Find out how to connect the MYO armband with the mobile device.
  + Get raw EMG data.
  + Find out the meaning of EMG data.

Although the EMG data between two people are quite alike but the accuracy for translate the sign language still depend on:

* + The longer doing the sign language, the more tired of muscle => the lower accuracy.
  + Condition of the armbands, the longer using time, the lower accuracy.
  + Difference type of people (thin, fat, average) has difference type of muscle => impact to the accuracy.
  + The input from the staff as the base value in the database.

### 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.
* Team has no information about the value receive from the armbands.
* MYO armband is a device that read inner body part: muscle. All team members have no knowledge on anatomy or human body structure.
* No knowledge on sign language: Main topic of this project is highly related with sign language, which is quite hard to approach for all team members. Team knows nothing about sign language.
* Advantages:
* Receive 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.

#### 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 and Android device will get those data actively via BLE. Right after getting the EMG data that defines the rest signal, 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 depend 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 resource into device to translate sign language without internet connection. User can train new custom sign move for the system for personal use.

##### Web Site

Website is a place for guest to search for sign language instruction video or register account to use main function of the system. User also can use this website to buy license to upgrade account into premium user.

* 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.

##### Web Services

Web service provide API for Mobile application to connect to server to perform translate, train, and download database function.

* For user:
* doTranslate: Get EMG data from mobile run the algorithms to translate sign language and return result to mobile.
* doLoginFromMoblie: Login from mobile application.
* For premium user:
* doDownload: Download standard library to mobile storage to use translate offline function.
* For staff:
* doTrain: Get EMG data and meaning of the gesture to create a word and store in the standard library of system storage.

##### Mobile Application

* For staff
* Train (Online): Staff can input new sign language gesture and meaning of it to create a new word and store in standard library of system storage.
* For user

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

* Translate (Online): User can translate sign language into text or sound with internet connection required.
* For premium user
* Translate (Offline): User can download resource to device to translate sign language without an internet connection.
* Train (Offline): User can train new personal sign move and meaning of it for the system for personal use.

#### Boundaries of the System

* A User who wants to use the functions of this system have to equip enough device includes:
* A Pair of MYO gesture control armbands.
* 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:
* A pair of MYO gesture control armband.
* A mobile device with the training application installed and internet connected.
* The complete product includes:
* A mobile application that allow:
* Train sign language and meaning of it (for staff)
* Translate (online) sign language into text / sound (for user)
* Translate (offline) sign language into text/sound (for premium user)
* Train (offline) custom sign and meaning of it for personal use (for premium user)
* A web application that allow:
* Register account to use main function of the system
* Search for sign language instruction video
* Buy license to upgrade account to premium user to use offline function

#### 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: IOS, MacOS, Windows application
* 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/web development**

|  |  |  |
| --- | --- | --- |
| **Windows** | **Minimum Requirements** | **Recommended** |
| **Internet Connection** | Cable, Wi-Fi (4 Mbps) | Cable, Wi-Fi (8 Mbps) |
| **Operating System** | Ubuntu Server 12 LTS | Ubuntu Server 14.04.2 LTS |
| **Computer Processor** | Intel® CORE i3 Quad core 2.1 GHz | Intel® CORE i7 Quad core 2.4 GHz |
| **Computer Memory** | 2GB RAM | 4GB or more |

Table 2: Hardware Requirement for server/ web development

* **For mobile development**

|  |  |  |
| --- | --- | --- |
| **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 600 1.89GHz 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) development

|  |  |
| --- | --- |
| **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) development

##### Software requirements

|  |  |  |
| --- | --- | --- |
|  | Name / Version | Description |
| 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

|  |  |  |
| --- | --- | --- |
|  | Name / Version | Description |
| IDE | Android Studio 1.5.1 | Programming tools |
| 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 artifact, such as a SRS, that is reviewed and accepted before creating a comprehensive design model (and so on) developer instead evolve the critical development artifact 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. Chapter 3: Agile software development page 59

- <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. Chapter4 – Software processes Page 68

#### 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.
* Team has to study the meaning of those data which get from the MYO armband by researching structure hoc human muscle.
* Team has no idea about the algorithms that solve the problem of the project.
* Team cannot predict what will happen during the time using the MYO armband

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 Nhất 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 |

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. |
| **Development** | -Design the current architecture  -Choose technology  - Implement 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 * Test case * 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 |

Table 10: Phase 1: Specification

#### Development

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

Table 11: Phase 2: Implementation

#### Validation

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

Table 12: Phase 3: Validation

### Task sheet

Place at folder “Task sheet” in Github



### All Meeting Minutes

Place at folder “Meeting minute” in Github 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

* Java: Using to develop website and web service.
* Android: Using to develop mobile application. Because team choose android native to develop the mobile application so the coding convention is base on Java.

Summary:

* 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 statements:

-Package statement is the first non-comment line.

-Import statement is after package statement.

* Constants

-Numerical constants should not be coded directly.

* Variable Assignments:

-Avoid assigning several 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.

* Using Java coding convention from:

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

* Using Android codding convention form:

<https://source.android.com/source/code-style.html>

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 (Online)

### User Requirement

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

* Translate (Online)
* Buy license
* Edit profile

### 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)
* Train (Offline)

### 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:

* Create notification
* Send notification

## System Requirement Specification

### External Interface Requirement

#### User Interface

* The user interface use English language in website and android mobile application.
* The user interface for website display best on 1366x768 pixels – screen size.
* The user interface for android application is designed base on material design and display best on 1920x1080 pixels – screen size

#### Hardware Interface

* Android smartphone supports Bluetooth 4.0 low energy
  + OS: Android 4.4: Kitkat
  + Chipset: Snapdragon 400 1.7GHz Dual Core
  + RAM: 512MB
* Two MYO gesture control armband
* Computer:
* OS: Ubuntu Server 12 LTS
* CPU: Intel® CORE i3 Quad core 2.1 GHz
* RAM: 2GB

#### Software Interface

* Web application: work with Chrome (v47 or above), Internet Explorer (v10 or above), Firefox (v43 or above)
* Mobile application: Android operation system (v4.4 or above)

#### Communication 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

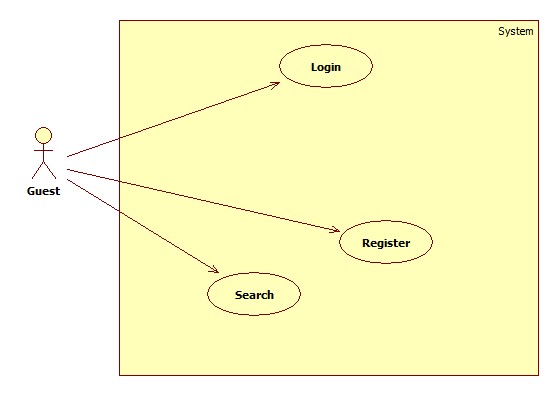


Figure 3: <Guest> Overview Use Case

##### <Guest> Register



Figure 4: <Guest> Register

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE – UC001** | | | |
| **Use Case No.** | 001 | **Use Case Version** | 2.0 |
| **Use Case Name** | Register | | |
| **Author** | NguyenNN | | |
| **Date** | 20/01/2016 | **Priority** | Normal |
| **Actor:**   * Guest   **Summary:**   * This use case allows Guest to register 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:** Systemshows messages   **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, regex [^[\_A-Za-z0-9-\\+]+(\\.[\_A-Za-z0-9-]+)\*@"+"[A-Za-z0-9-]+(\\.[A-Za-z0-9]+)\*(\\.[A-Za-z]{2,})$] * Full name : free text input, required, length(10-50) * 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) * Phone: free number input, required, length(10-12) positive integer, value:[0,9] | | 2 | Guest inputs information |  | | 3 | Guest sends command to register  [Alternative 1]  [Alternative 2] | System shows login view  Account registered  [Exception 1]  [Exception 2]  [Exception 3]  [Exception 4] |   **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 send command to back to login view | System shows login view  Account isn’t created |   **Exceptions 1:**   |  |  |  | | --- | --- | --- | | Step | Cause | System Response | | 1 | Guest input already exist username | System show warning message “User name already exist” |   **Exceptions 2:**   |  |  |  | | --- | --- | --- | | Step | Cause | System Response | | 1 | Guest input already exist email | System shows warning message “Email already exist”. |   **Exceptions 3:**   |  |  |  | | --- | --- | --- | | Step | Cause | System Response | | 1 | Guest does not input required field. | System notices that guest need to input all these field:   * “Email”: System display warning message: “please fill out this field”. * “Full name”: System display warning message: “please fill out this field”. * “Username”: System display warning message: “please fill out this field”. * “Password”: System display warning message: “please fill out this field”. * “Repeat password”: System display warning message: “please fill out this field”. * “Phone”: System display warning message: “please fill out this field”. |   **Exceptions 4:**   |  |  |  | | --- | --- | --- | | Step | Cause | System Response | | 1 | Guest input wrong some fields with requirement. | System notices that guest need to re-input all these field:   * “Email”: System display warning message: “Email invalid! ([me@example.com)](mailto:me@example.com))”. * “Email”: System display warning message: “Email must be 10 - 254 characters”. * “Full name”: System display warning message: “Full name must be 10 - 50 characters”. * “Username”: System display warning message: “Username must be 6 - 20 characters”. * “Password”: System display warning message: “Password must be 6 - 12 characters”. * “Repeat password”: System display warning message: “Repeat password does not match password”. * “Phone”: System display warning message: “Phone must be numbers”. |   **Relationships:** N/A  **Business Rules:**   * 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

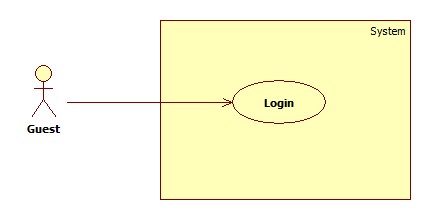


Figure 5: <Guest> Login

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE – UC002** | | | |
| **Use Case No.** | 002 | **Use Case Version** | 2.0 |
| **Use Case Name** | Login | | |
| **Author** | NguyenNN | | |
| **Date** | 20/01/2016 | **Priority** | Normal |
| **Actor:**   * Guest   **Summary:**   * This use case allows Guest login to the system on website and mobile application   **Goal:**   * Guest login successfully with the proper role   **Triggers:**   * Guest send the login command   **Preconditions:**   * Guest has an account   **Post Conditions:**   * **Success:** Guest accesses the system successfully * **Fail:** System shows error message “Invalid username or password”   **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 * Password : free text input | | 2 | Guest inputs information |  | | 3 | Guest sends command to login to system  [Alternative 1] | Guests will login system with their specific role |   **Alternative Scenario:**   |  |  |  | | --- | --- | --- | | Step | Actor action | System Response | | 1 | Guest input invalid username or password | System show error message “Invalid username or password” |   **Exceptions:** N/A  **Relationships:** N/A  **Business Rules:**  - After login to system, guest will be redirected to specific view based on their role on the system: staff or user.   * If role is “User”, the system will display to User view. * If role is “Staff”, the system will display to Staff view. | | | |

Table 14: Login specification

##### <Guest> Search

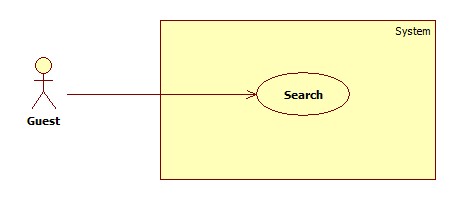


Figure 6: <Guest> Search

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE – UC003** | | | |
| **Use Case No.** | 003 | **Use Case Version** | 2.0 |
| **Use Case Name** | Search | | |
| **Author** | QuyPH | | |
| **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 in video, keyword and description * **Fail:** N/A   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | Actors input keyword in to search textbox |  | | 2 | Actors send Search command  [Alternative 1] | System will find in database any record of dictionary have keyword like input text and show that record as:   * An instruction video * Keyword * Description |   **Alternative Scenario :**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | Guest inputs blank in textbox | System show all record |   **Exceptions :** N/A  **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 as: * An instruction video * Keyword * Description | | | |

Table 15: Search specification

#### <User> Overview use case

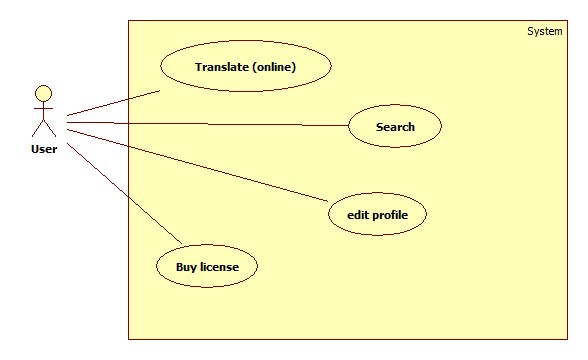


Figure 7: <User> Overview Use Case

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

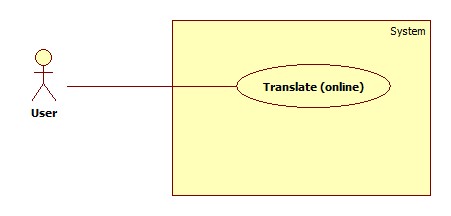


Figure 8: <User> Translate (Online)

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE – UC004** | | | |
| **Use Case No.** | 004 | **Use Case Version** | 2.0 |
| **Use Case Name** | Translate (online) | | |
| **Author** | ThaiTC | | |
| **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. * Two MYO armbands must connect to mobile device successfully   **Post Conditions:**   * **Success:** Text or voice match with the sign language is shown. * **Fail:** Show error message “Server error”.   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | Actor send detect command  [Alternative 1]  [Alternative 2] | Processing dialog is shown  [Exception 2] | | 2 | Actor performs sign language | Processing dialog is shown  [Exception 2] | | 3 | Actor performs end sign command | System returns proper text or voice.  [Exception 1]  [Exception 3] |   **Alternative Scenario1:**     |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | Actor send vibrate command | The MYO armband vibrates |   **Alternative Scenario2:**     |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | Actor send manual command | Processing dialog is shown  [Exception 2] | | 2 | Actor performs sign language | Processing dialog is shown  [Exception 2] | | 3 | Actor send capture EMG manual command | [Exception 2] | | 4 | Actor performs end sign command | System returns proper text or voice.  [Exception 1]  [Exception 3] |   **Exceptions 1:**   |  |  |  | | --- | --- | --- | | Step | Cause | System Response | | 1 | Internet connection is lost | System shows error message “Cannot connect to server”. |   **Exceptions 2:**   |  |  |  | | --- | --- | --- | | Step | Cause | System Response | | 1 | Bluetooth is disconnect | System shows warning message “MYO armbands must be paired”. |   **Exceptions 3:**   |  |  |  | | --- | --- | --- | | Step | Cause | System Response | | 1 | Sever error | System shows error message “Server error”. |   **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 receive 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 vn speak. | | | |

Table 16: Translate (Online) specification

##### <User> Buy license

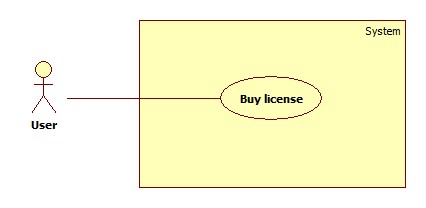


Figure 9: <User> Buy license

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE – UC005** | | | |
| **Use Case No.** | 005 | **Use Case Version** | 2.0 |
| **Use Case Name** | Buy License | | |
| **Author** | QuyPH | | |
| **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:**   * N/A   **Post Conditions:**   * **Success:** User upgrade to Premium user, system shows payment success view * **Fail:** Systemshows payment fail view   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | User sends command to buy license | System to switch to Paypal payment view | | 2 | User sends command to confirm to continue the payment  [Alternative 1] | System switch to Paypal payment process | | 3 | User completes payment process with Paypal  [Alternative 2]  [Alternative 3]  [Alternative 4] | System switch back and show successful view |   **Alternative Scenario 1:**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | User sends command to decline | System switch to payment fail view |   **Alternative Scenario 2:**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | Paypal responses payment process is fail | System switch to payment fail view |   **Alternative Scenario 3:**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | Paypal payment process time out | System switch to payment fail view |   **Alternative Scenario 3:**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | User does the payment with not enough money in account | System switch to payment fail view |   **Exceptions :** N/A  **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” in 30 days. If after 30 days, license isn’t re-bought, account type will be back to “User” | | | |

Table 17: Buy license specification

##### <User> Search

Reference 2.3.1.3 <Guest> Search

##### <User> Edit profile

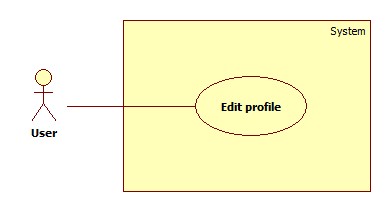


Figure 10: <User> Edit profile

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE – UC006** | | | |
| **Use Case No.** | 006 | **Use Case Version** | 2.0 |
| **Use Case Name** | Edit profile | | |
| **Author** | ThaiTC | | |
| **Date** | 20/01/2016 | **Priority** | Normal |
| **Actor:**   * User   **Summary:**   * This use case allows User to edit account profile   **Goal:**   * Account is edited successfully and store in database of the system   **Triggers:**   * User sends command to edit profile   **Preconditions:**   * Actor already has an account   **Post Conditions:**   * **Success:** New account will be edit * **Fail:** Systemshows messages   **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, regex [^[\_A-Za-z0-9-\\+]+(\\.[\_A-Za-z0-9-]+)\*@"+"[A-Za-z0-9-]+(\\.[A-Za-z0-9]+)\*(\\.[A-Za-z]{2,})$] * Full name : free text input, required, length(10-50) * New Password: free text input, required, length(6-12) * Repeat new password: free text input, required, length(6-12) * Phone: free number input, length(10-12), positive integer, value: [0,9] | | 2 | Guest inputs information |  | | 3 | Guest sends command to save edit  [Alternative 1] | System shows dictionary view  [Exception 1]  [Exception 2]  [Exception 3] |   **Alternative Scenario 1:**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | Guest sends command to reset | System reset all field to blank |   **Exceptions 1:**   |  |  |  | | --- | --- | --- | | Step | Cause | System Response | | 1 | Guest input already exist email | System shows warning message “Email already exist”. |   **Exceptions 2:**   |  |  |  | | --- | --- | --- | | Step | Cause | System Response | | 1 | Guest does not input required field. | System notices that guest need to input all these field:   * “Email”: System display warning message: “please fill out this field”. * “Full name”: System display warning message: “please fill out this field”. * “Password”: System display warning message: “please fill out this field”. * “Repeat password”: System display warning message: “please fill out this field”. * “Phone”: System display warning message: “please fill out this field”. |   **Exceptions 3:**   |  |  |  | | --- | --- | --- | | Step | Cause | System Response | | 1 | Guest input wrong some fields with requirement. | System notices that guest need to re-input all these field:   * “Email”: System display warning message: “Email invalid! ([me@example.com)](mailto:me@example.com))”. * “Email”: System display warning message: “Email must be 10 - 254 characters”. * “Full name”: System display warning message: “Full name must be 10 - 50 characters”. * “Password”: System display warning message: “Password must be 6 - 12 characters”. * “Repeat password”: System display warning message: “Repeat password does not match password”. * “Phone”: System display warning message: “Phone must be numbers”. |   **Relationships:** N/A  **Business Rules:**   * After edited, new information of account will be stored in database of the system. | | | |

Table 18: Edit profile specification

#### <Premium User> Overview use case

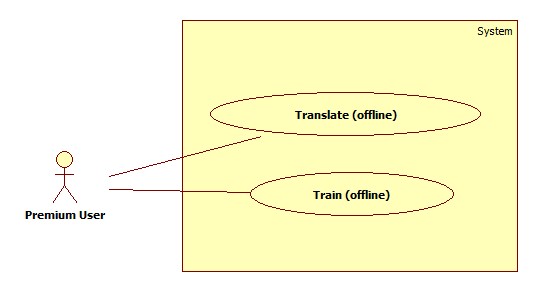


Figure 11: <Premium User> Overview Use case

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

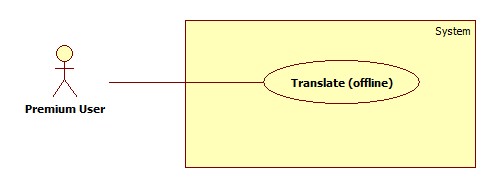


Figure 12: <Premium User> Translate (Offline)

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE – UC007** | | | |
| **Use Case No.** | 007 | **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 * MYO armbands must be paired with mobile device   **Post Conditions:**   * **Success:** Text or voice match with the sign language is shown without internet connection * **Fail:** Show error message “MYO armbands must be paired”.   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | Premium user sends detect command  [Alternative 1]  [Alternative 2] | Processing dialog is shown  [Exception 1]  [Exception 2] | | 2 | Premium user performs sign language | Processing dialog is shown  [Exception 1]  [Exception 2] | | 3 | Premium user performs end sign command | System returns proper text or voice. |   **Alternative Scenario 1:**   |  |  |  | | --- | --- | --- | | Step | Actor action | System Response | | 1 | Premium user sends vibrate command | The MYO armband vibrates |   **Alternative Scenario 2:**   |  |  |  | | --- | --- | --- | | Step | Actor action | System Response | | 1 | Premium user send manual command | Processing dialog is shown  [Exception 1] | | 2 | Premium user performs sign language | Processing dialog is shown  [Exception 1] | | 3 | Premium user send capture EMG manual command | [Exception 1] | | 4 | Premium user performs end sign command | System returns proper text or voice.  [Exception 1] |   **Exceptions 1:**   |  |  |  | | --- | --- | --- | | Step | Cause | System Response | | 1 | Bluetooth is disconnect | System shows warning message “MYO armbands must be paired”. |   **Relationships:** N/A  **Business Rules:**   * After connect the MYO armbands with the smartphone, the application 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, the application 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 premium user want to play sound, the system will use text-to-speech of vn speak. | | | |

Table 19: Translate (Offline) specification

##### <Premium User> Train (Offline)

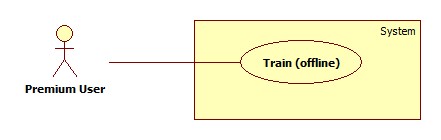


Figure 13: <Premium User> Train (Offline)

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE – UC008** | | | |
| **Use Case No.** | 008 | **Use Case Version** | 2.0 |
| **Use Case Name** | Train (offline) | | |
| **Author** | NguyenNN | | |
| **Date** | 23/01/2016 | **Priority** | High |
| **Actor:**   * Premium user   **Summary:**   * This use case helps premium user train custom sign for personal use   **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 * MYO armbands must be paired with mobile device   **Post Conditions:**   * **Success:** new sign language is trained * **Fail:** Show error message “Save fail”   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | Premium user inputs meaning of new custom sign  [alternative 1]  [alternative 2] |  | | 2 | Premium user send start command  [alternative 1]  [alternative 2] | Processing dialog is shown. | | 3 | Premium user performs sign language | Processing dialog is shown.  [Exception 1] | | 4 | Premium user sends save command  [alternative 1]  [alternative 2]  [alternative 3] | System shows the message “Save successfully” |   **Alternative Scenario 1:**   |  |  |  | | --- | --- | --- | | Step | Actor action | System Response | | 1 | Premium user sends vibrate command | The MYO armband vibrates |   **Alternative Scenario 2:**   |  |  |  | | --- | --- | --- | | Step | Actor action | System Response | | 1 | Premium user sends detect command | System turn to translate offline view |   **Alternative Scenario 3:**   |  |  |  | | --- | --- | --- | | Step | Actor action | System Response | | 1 | Premium user inputs blanks meaning fields | System required to input |   **Exceptions 1:**   |  |  |  | | --- | --- | --- | | Step | Cause | System Response | | 1 | Bluetooth connection is lost | System shows warning message “MYO armbands must be paired”. |   **Relationships:** N/A  **Business Rules:**   * After connected the MYO armbands with the smartphone, the system can get premium user’s EMG data which describes the sign language via BLE * After save, custom sign and meaning of it will be stored in the smartphone * EMG data must follows EMG data format | | | |

Table 20: Train custom sign specification

#### <Staff> Overview use case

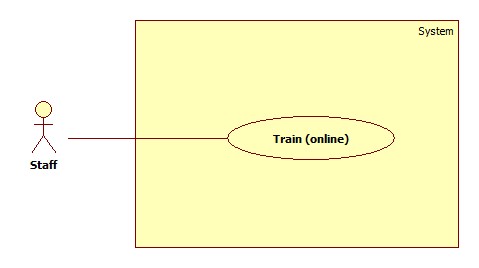


Figure 14: <Staff> Overview Use Case

##### <Staff> Train (Online)

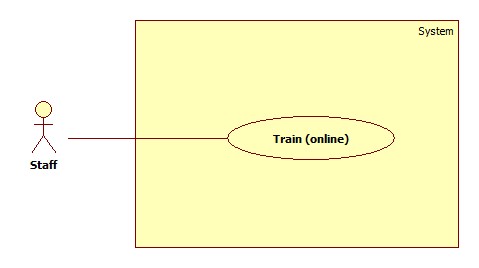


Figure 15: <Staff> Train (Online)

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE – UC09** | | | |
| **Use Case No.** | 009 | **Use Case Version** | 2.0 |
| **Use Case Name** | Train (online) | | |
| **Author** | NguyenNN | | |
| **Date** | 23/01/2016 | **Priority** | High |
| **Actor:**   * Staff   **Summary:**   * This use case helps staff train standard 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 * Myo armbands must be paired with mobile device   **Post Conditions:**   * **Success:** new sign language is trained * **Fail:** Show error message “Server error”   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | Staff inputs three meaning fields of new sign language gesture  [Alternative 2] |  | | 2 | Staff send start command | The processing dialog is shown  [Exception 1] | | 3 | Staff performs sign language | The processing dialog is shown  [Exception 1] | | 4 | Staff sends save command  [Alternative 1]  [Alternative 2] | System shows the message “Save successfully”  [Exception 2]  [Exception 3] |   **Alternative Scenario 1:**   |  |  |  | | --- | --- | --- | | Step | Actor action | System Response | | 1 | Staff send vibrate command | MYO armband vibrates |   **Alternative Scenario 2:**   |  |  |  | | --- | --- | --- | | Step | Actor action | System Response | | 1 | Staff input blank in one of three meaning filed | System require to input to field which blank |   **Exceptions 1:**   |  |  |  | | --- | --- | --- | | Step | Cause | System Response | | 1 | Bluetooth connection is lost | System shows warning message “MYO armbands must be paired”. |   **Exceptions 2:**   |  |  |  | | --- | --- | --- | | Step | Cause | System Response | | 1 | Internet connection is lost | System shows error message “Cannot connect to server”. |   **Exceptions 3:**   |  |  |  | | --- | --- | --- | | Step | Cause | System Response | | 1 | Server error | System shows error message “Server error”. |   **Relationships:** N/A  **Business Rules:**   * Staff must clear about the sign language * The sign language must be based on the standard document * After connected the MYO armbands with the smartphone, the system can get staff’s EMG data which describes the sign language via BLE * 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 | | | |

Table 21: Train sign language specification

#### <Authenticated user> Overview use case

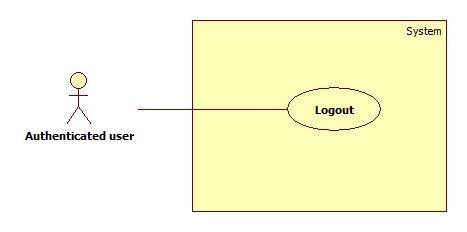


Figure 16: <Authenticated user> Overview Use Case

##### <Authenticated user> Logout

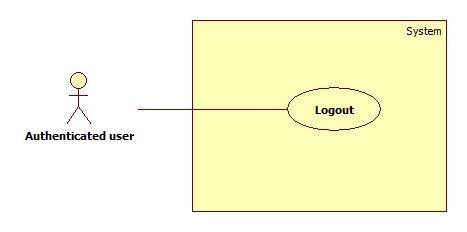


Figure 17: <Authenticated user> Logout

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE – UC010** | | | |
| **Use Case No.** | 010 | **Use Case Version** | 2.0 |
| **Use Case Name** | Logout | | |
| **Author** | NguyenNN | | |
| **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 * Actor send commend after not available time for too long (for web)   **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 | Actor sends command to Logout | System clears session state if any, takes user out of the system.  System displays login view. |   **Alternative Scenario 1:**   |  |  |  | | --- | --- | --- | | Step | Actor action | System Response | | 1 | Actor send commend after not available time for too long (for web) | System clears session state if any, takes user out of the system.  System displays session expired view. |   **Exceptions:** N/A  **Relationships: N/A**  **Business Rules:**   * After logout, role “Authenticated User” will become “Guest” * If actors is not available longer than 30 minutes, they will see session expired view when they are back | | | |

Table 22: Logout specification

#### <Scheduler> Overview use case

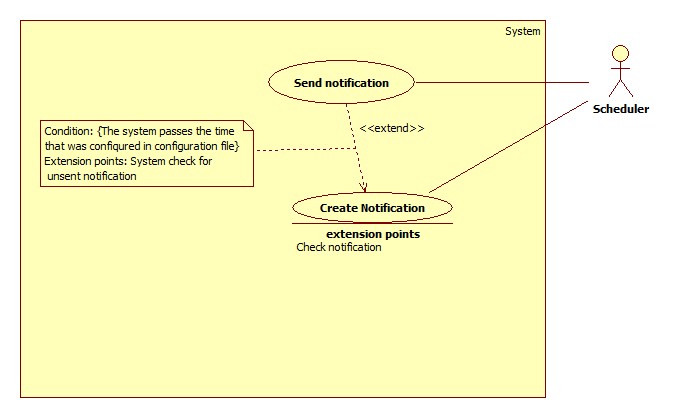


Figure 18: <Scheduler> Overview Use Case

##### <Scheduler> Create notification

Figure 19: <Scheduler> Create notification

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE – UC011** | | | |
| **Use Case No.** | 011 | **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 * **Fail:** No new notification save in the storage   **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 compare with the license expiration date of premium user. If current date < expiration date 5 days   * 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 |   **Relationships:** Extended by send notification  **Business Rules:**   * Daily, at the time that was set in configuration file, the system will check for the license expiration date of premium user * Condition for creating notification * Curent date – License expiration date <= 5 : content of the notification is: Your licence is about to expire in … days. * Curent date – License expiration date = 0 : content of the notification is : Your licence is expired. * The notification will be created with the status is “unsent” | | | |

Table 23: Create notification specification

##### <Scheduler> Send notification

Figure 20: <Scheduler> Send notification

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE – UC012** | | | |
| **Use Case No.** | 012 | **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 * 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 24: Send notification specification

## Software system attribute

### Usability

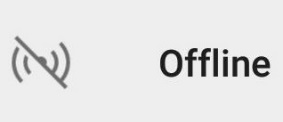
#### Graphic user interface

* All the texts, labels, alerts and messages will be written in English
* GUI of the mobile application is designed base on material design language of Google with the navigation bar contains the main functions on the left of the screen.

#### Usability

* System provides user GUI with instruction step by step.
* Icon with function name aside will help user to recognize the feature easier.

Exp:

* The system is easy to use. It will need about 1 hour for training for staff to use the mobile application to train sign language for the standard library of the system.
* It will take about 1 – 2 hours for user to get used to familiar to all function of the system for user including approach to know how to use the MYO armbands.

#### Installation

User can follow installation and manual guide for installation. If there are any problems, user cans contacts developer for help.

### Reliability

* The system uses electromyography technology to read activity of user’s muscle so the accuracy is higher than other systems.
* Scheduler task run at 00:00 everyday with 100% execution rate.
* Web service API response success rate is less than two failed requests per 10,000 requests.

### Availability

* The system relates to communication so it can be available 24/7.
* Server should have back-up method to make sure data can be restored easily if any problem happens.
* There is a function that allow premium user can use the system offline if there is no internet connection
* System is divided into modules, If a function is down, it will not impact others.

### Security

* N/A

### Maintainability

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

### Portability

* User, guest can use the web application of the system on an OS that support web browser
* Staff, user, premium user can use mobile application on any Android smartphone that support BLE and Android version from 4.4 KitKat

### Performance

* Requests from web application are responded in less than 10 seconds at 8 Mbps bandwidth speed.
* Mobile application can return result of the translation from EMG data into meaning through calling API after a rage from 0.8 to 1.1 second at 8 Mbps bandwidth speed.

## Conceptual diagram

Figure 21: 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 information |
| 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 25: 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 22: System architecture design

### 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 such as support multi language, support more kind of translation: form spoken language into sign language…
* In scope of 3-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.

### Mobile application architecture description

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

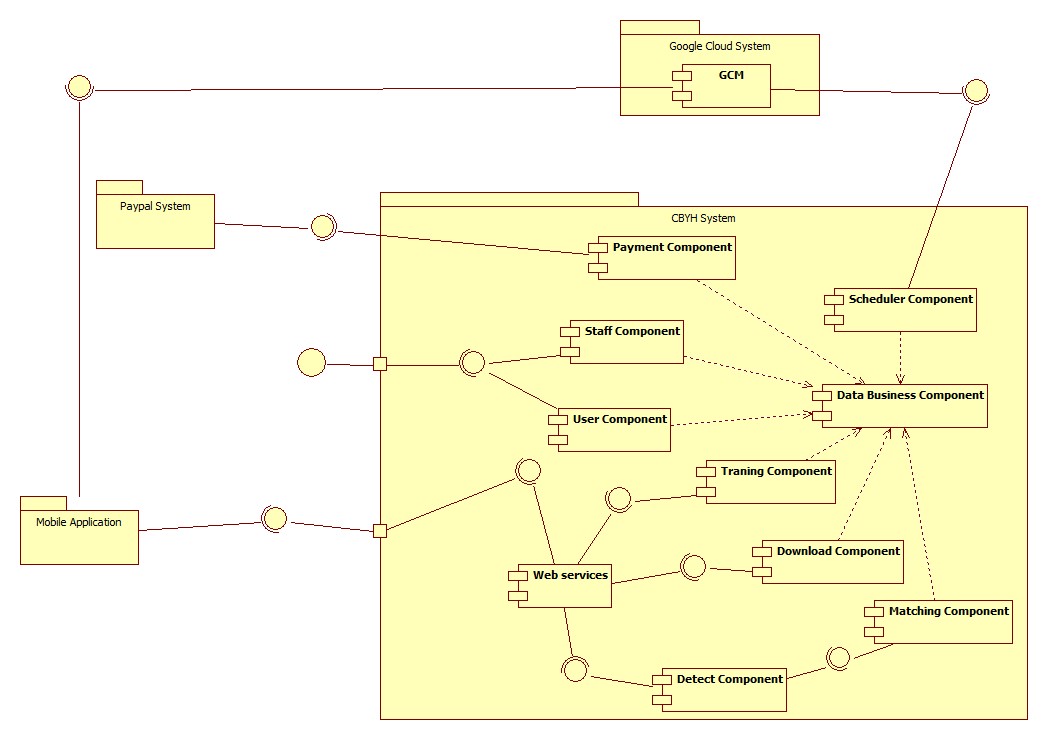
Reason for choosing: Currently, Android is the most common mobile platform in the world. Beside that, Android is the only one mobile platform that team member were taught in university.



**Reference:** [Android Developer Guide - Application Fundamentals](http://developer.android.com/guide/components/fundamentals.html)

http://developer.android.com/guide/components/fundamentals.html

## Component Diagram



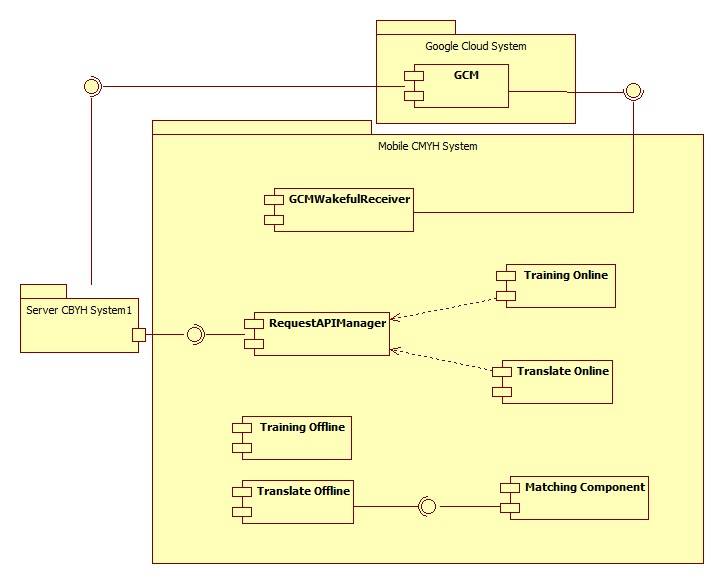
Figure 23: Component Diagram (Server side)

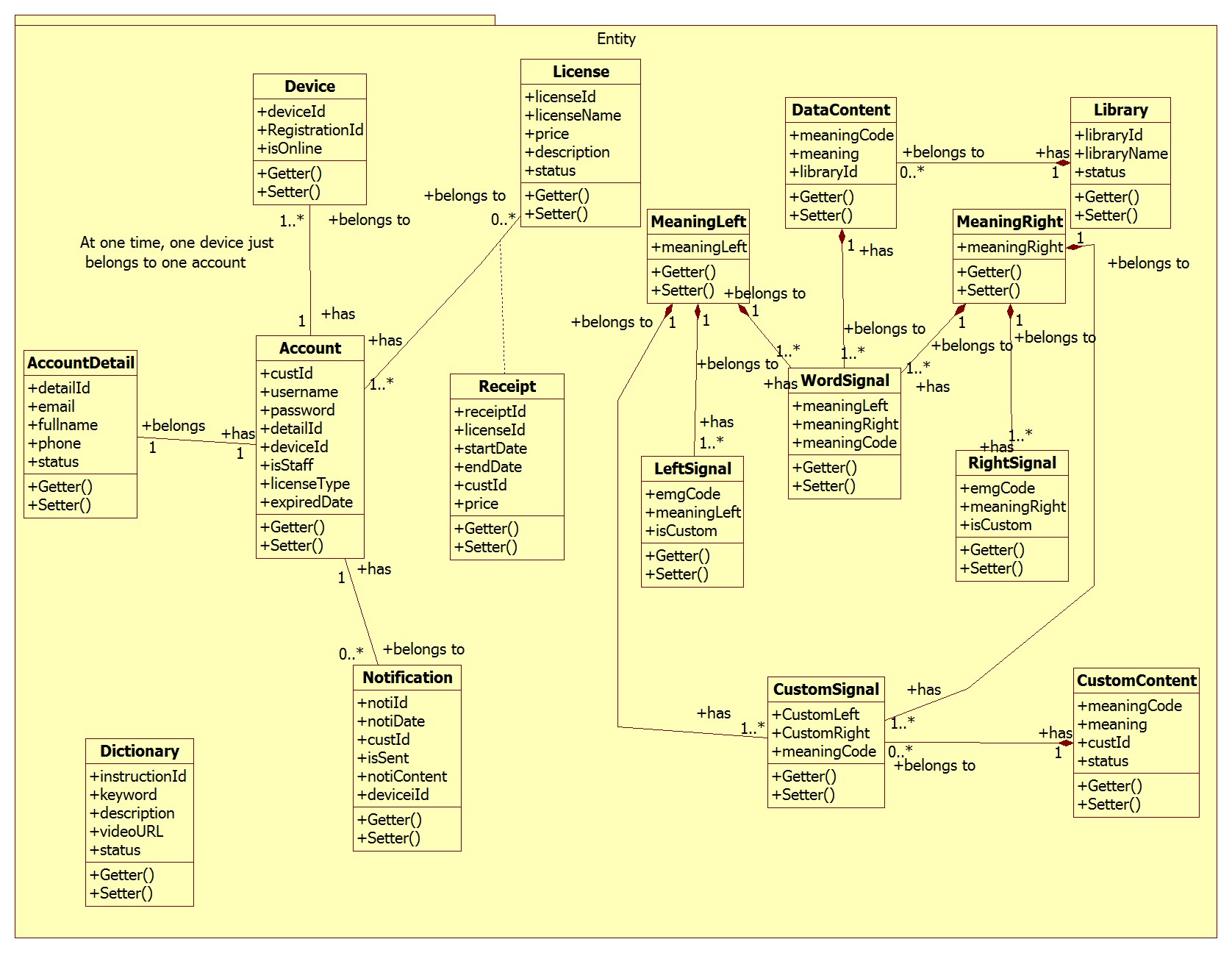
Figure 24: Component Diagram (Mobile side)

|  |  |
| --- | --- |
| Component Dictionary: Describes components | |
| 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 | Component to handle detect process |
| Download Component | Component to handle download data process |
| Web services | Component to handle provide API for mobile application |
| GCMWakefulReceiver | Component to receive notification from Google Cloud Message 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 26: Component Dictionary

## Detail Description

### Class Diagram

Figure 25: Class Diagram

|  |  |  |
| --- | --- | --- |
| Class dictionary: describe Class | | |
| Class Name | **Mapping column with Conceptual diagram** | **Description** |
| Account | user | Contain the account information |
| AccountDetail | N/A | Not exists in conceptual diagram but need to contain the detail information of account |
| Dictionary | dictionary | Contain the dictionary information |
| 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 |
| Receipt | N/A | Not exists in conceptual diagram but need to contain the receipt when user buy a license |

Table 27: 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 28: Account Attributes

#### 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 29: AccountDetail Attributes

#### 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 30: Dictionary Attributes

#### 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 31: Notification Attributes

#### 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 32: License Attributes

#### 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 33: Library Attributes

#### 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 34: Device Attributes

#### 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 35: LeftSignal Attributes

#### 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 36: RightSignal Attributes

#### MeaningLeft

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

Table 37: MeaningLeft Attributes

#### MeaningRight

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

Table 38: MeaningRight Attributes

#### 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 39: WordSignal Attributes

#### 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 40: DataContent Attributes

#### 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 41: CustomSignal Attributes

#### 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 42: CustomContent Attributes

#### Receipt

|  |  |  |  |
| --- | --- | --- | --- |
| Attribute | Type | Visibility | Description |
| receiptId | Integer | Private | Unique identifier of a receipt |
| 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 43: Receipt Attributes

### Interactive Diagram

#### Web Application

##### <User>

###### Buy license

Summary: this diagram show process of buy license

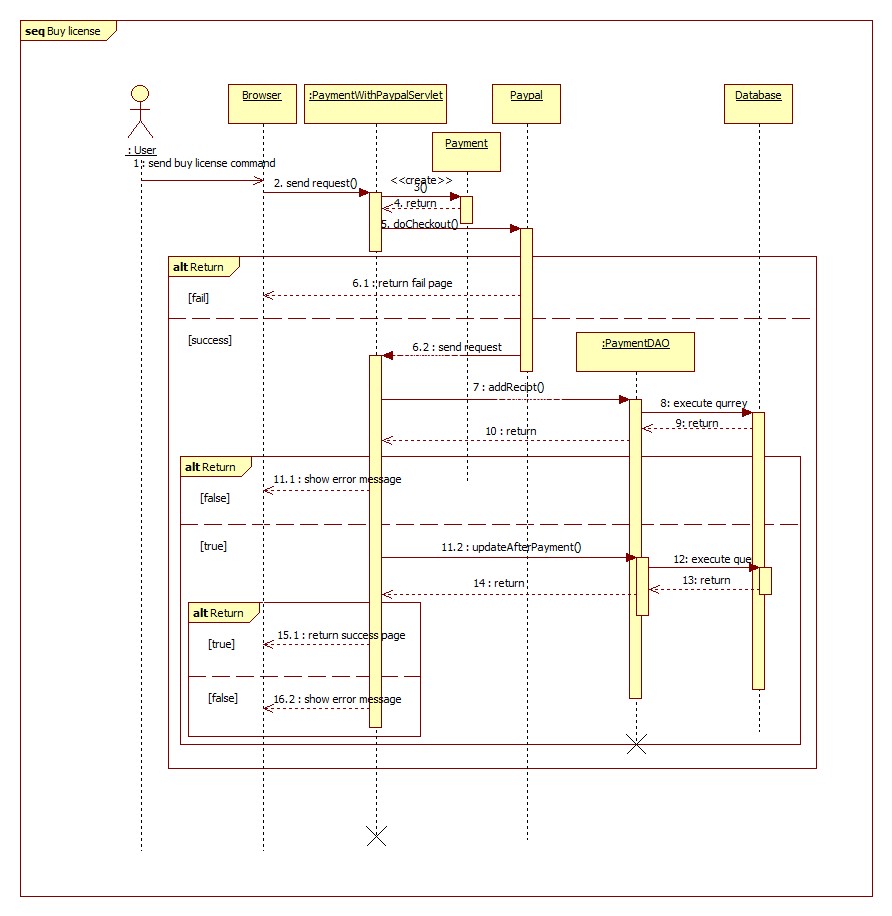


Figure 26: Sequence diagram - <User> Buy license

###### Update profile

Summary: this diagram show process of update profile

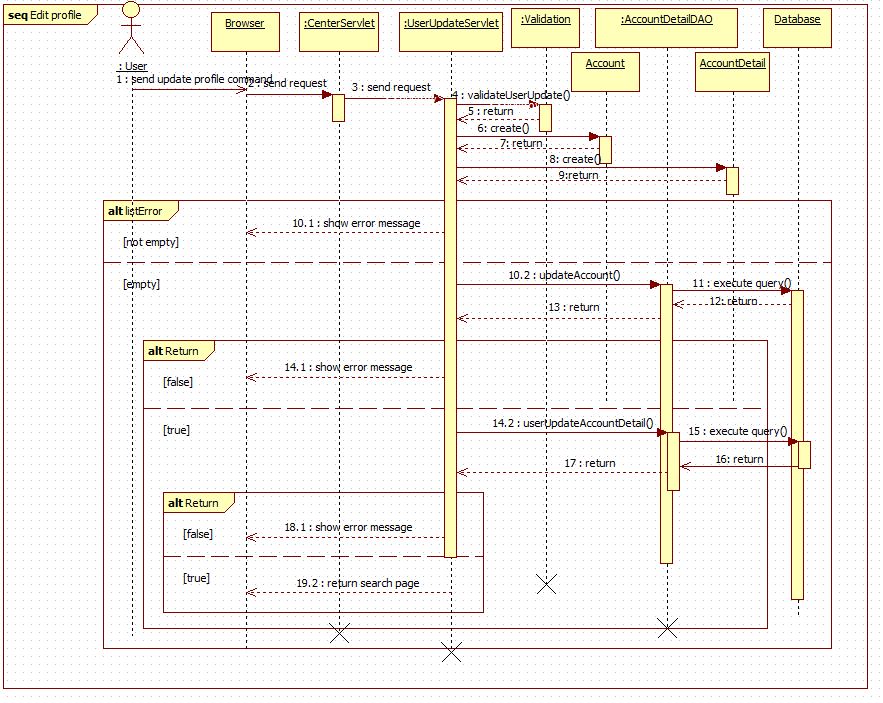


Figure 27: Sequence diagram - <User> Update profile

###### Search

Summary: this diagram show process of search

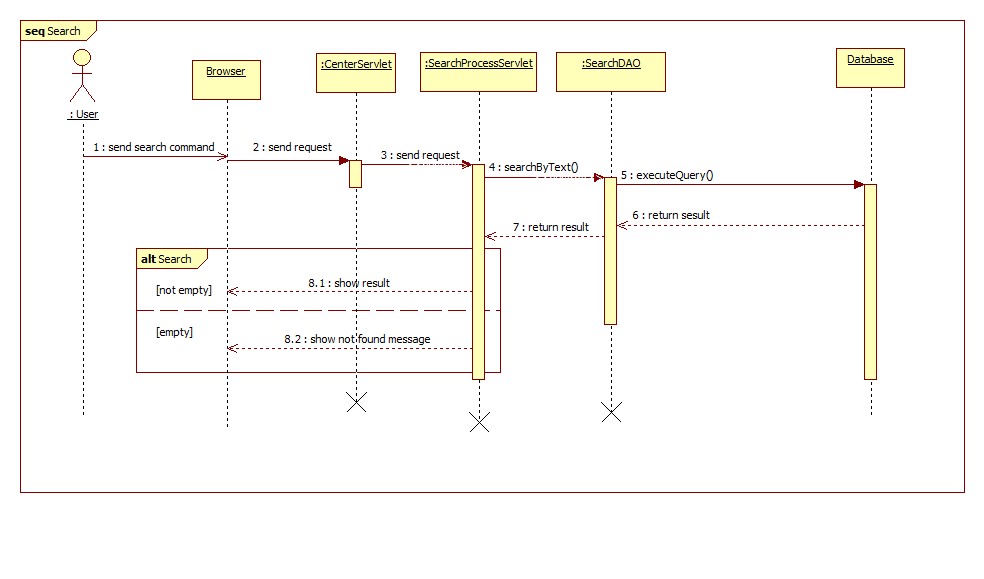


Figure 28: Sequence diagram - <User> Search

##### <Guest>

###### Register

Summary: this diagram show process of register

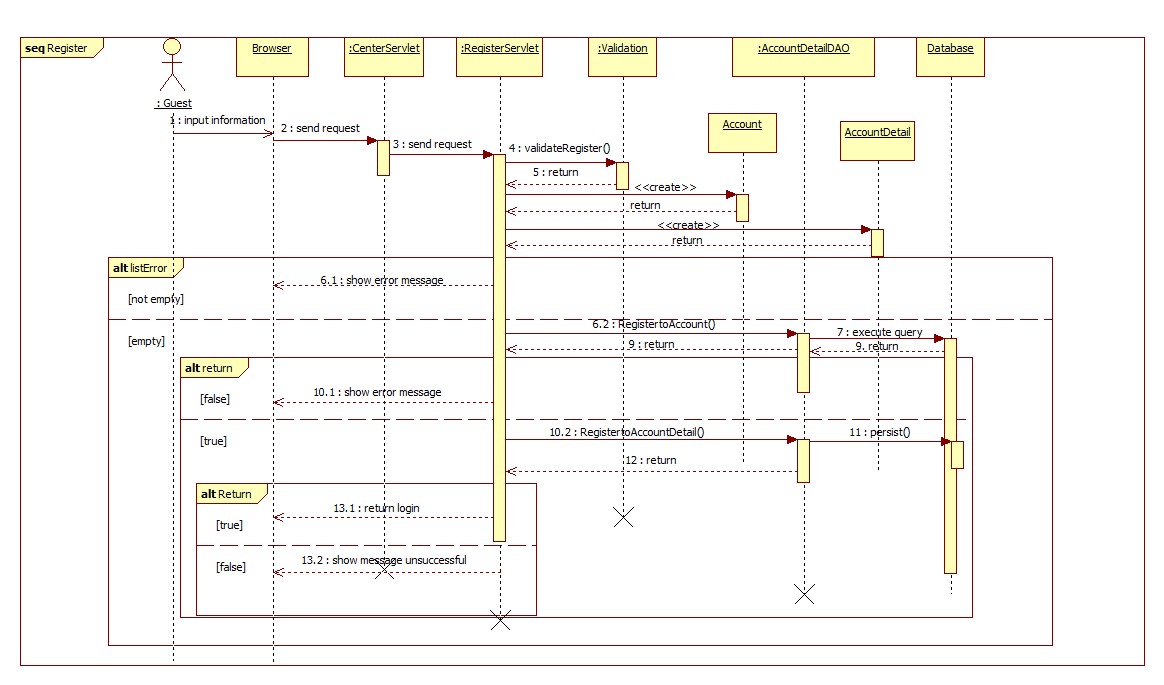


Figure 29: Sequence diagram - <Guest> Register

#### Mobile Application

##### <Staff>

###### Train online

Summary: this diagram show process of Train online

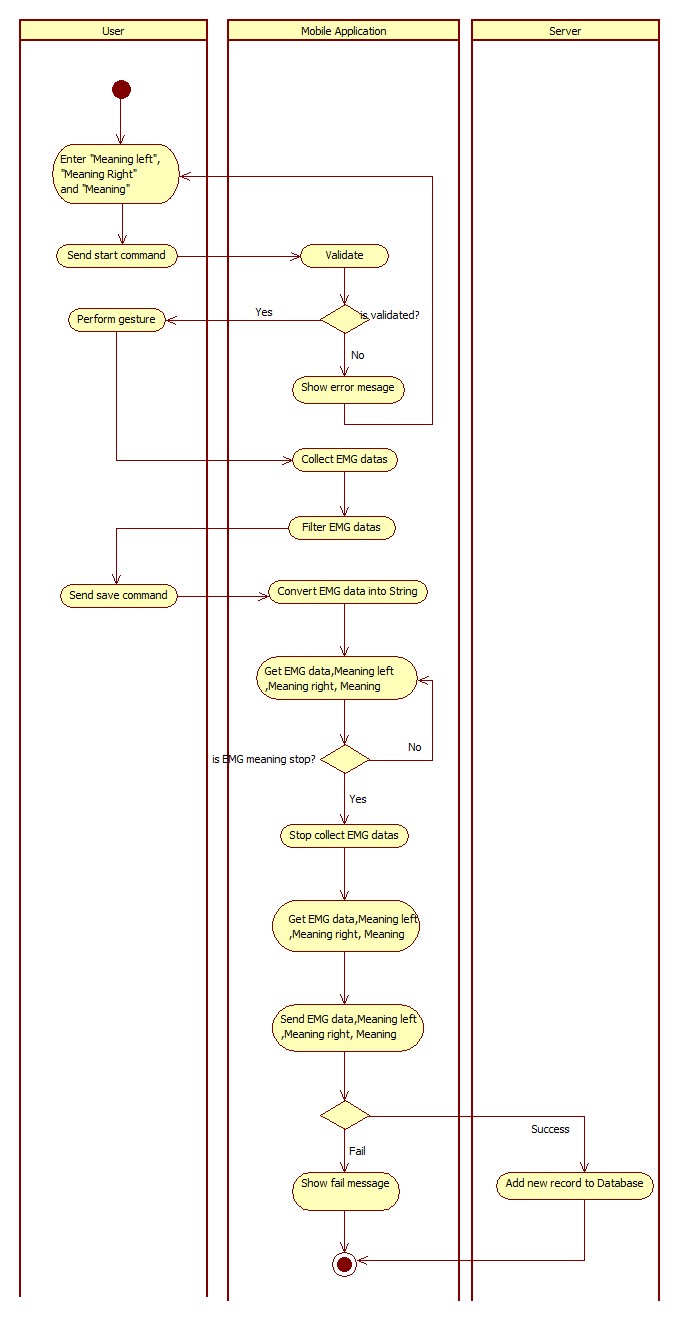


Figure 30: Activity diagram - <Staff> Train online

##### <User>

###### Connect Myo armbands

Summary: this diagram show process of connect Myo armbands

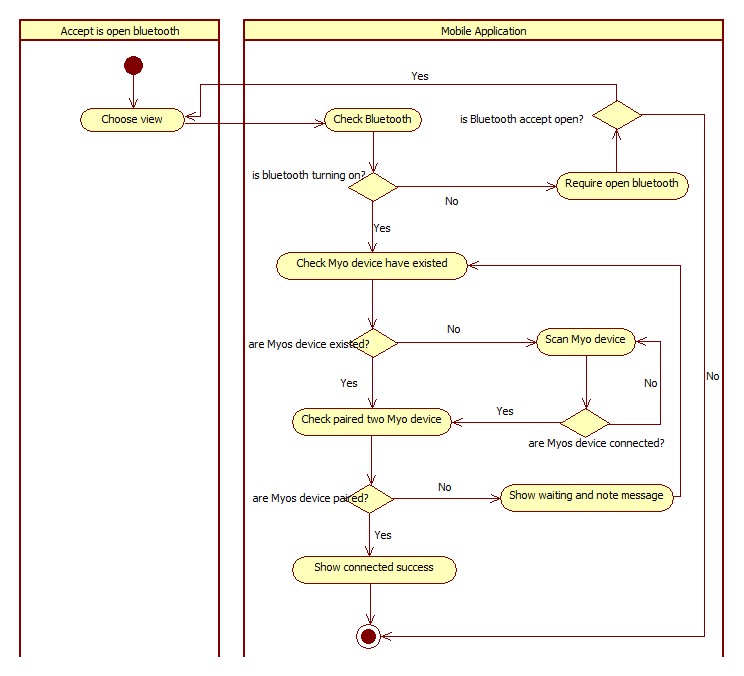


Figure 31: Activity diagram - <User> Connect Myo armbands

###### Translate online

Summary: this diagram show process of translate online

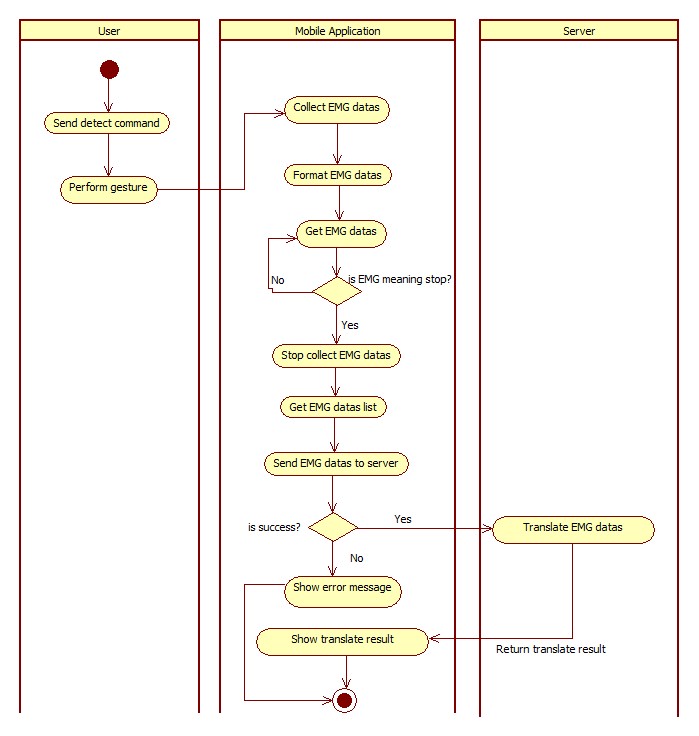


Figure 32: Activity diagram - <User> Translate online

##### <Premium User>

###### Translate offline

Summary: this diagram show process of translate offline

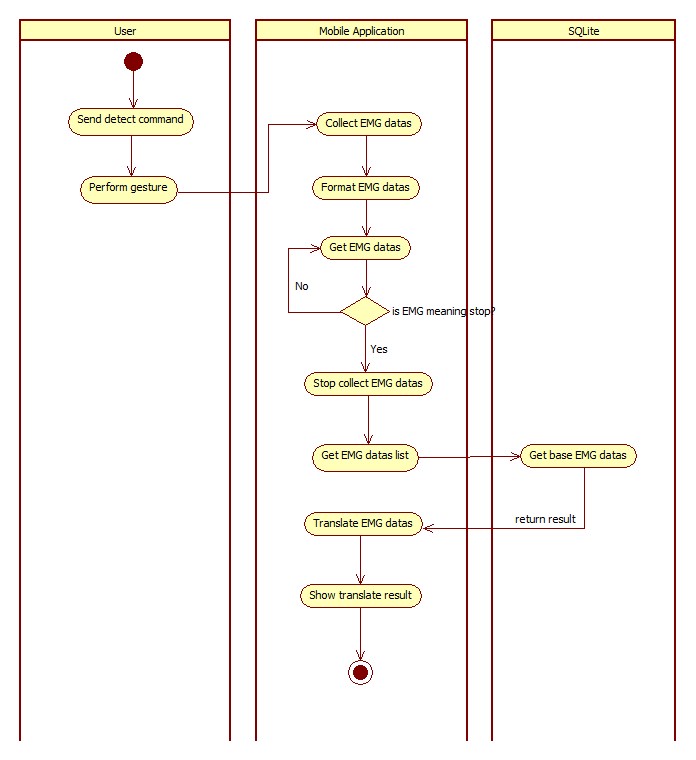


Figure 33: Activity diagram - <Premium User> Translate offline

###### Train offline

Summary: this diagram show process of train offline



Figure 34: 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 |
| public Response doDownload() | Download EMG base data for mobile | N/A | Response (Object) downFIle | Response (Object) | IOEException |

Table 44: 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 |
| IOEException | Thrown when there has been an Input/Output (usually when working with files) error. |

Table 45: Exception description

### Web Application Design

#### Guest interface design

##### Register

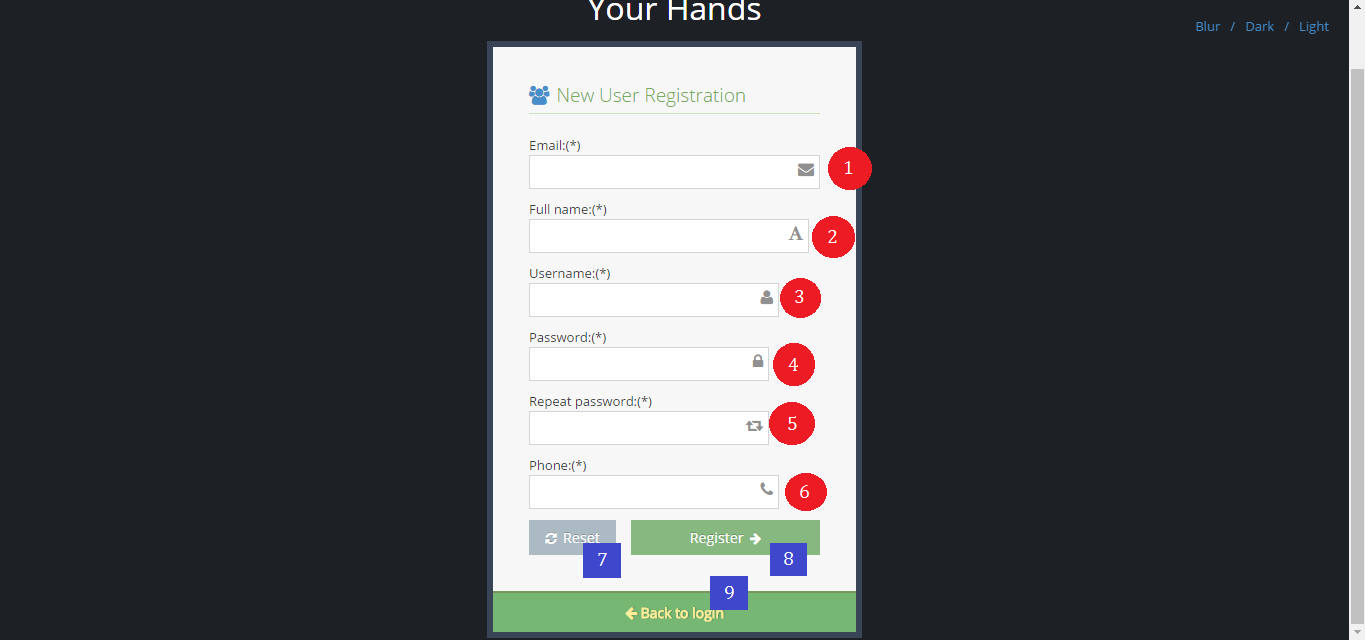


Figure 35: Interface - <Guest> Register

**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 46: <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 47: <Guest> Register buttons/ hyperlinks

#### User interface design

##### Buy license

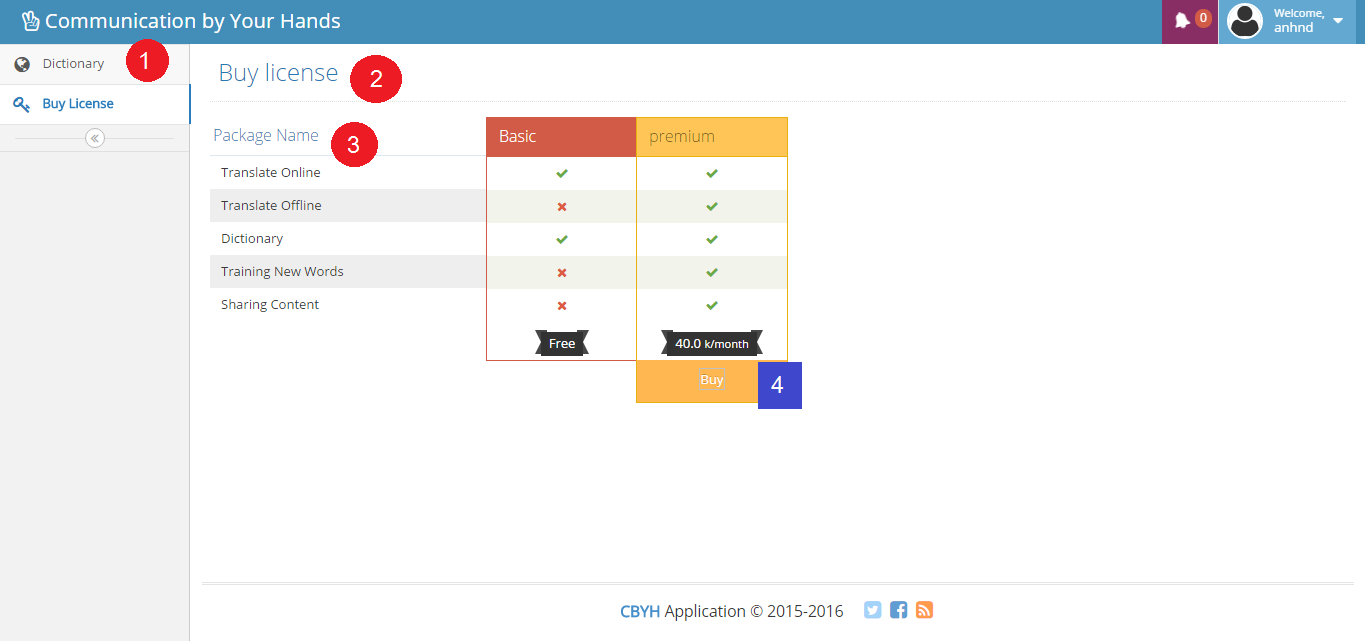


Figure 36: Interface - <User> Buy license

**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 48: <User> Buy license fields

**Buttons/Hyperlinks**

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

Table 49: <User> Buy license buttons/ hyperlinks

##### Search instruction

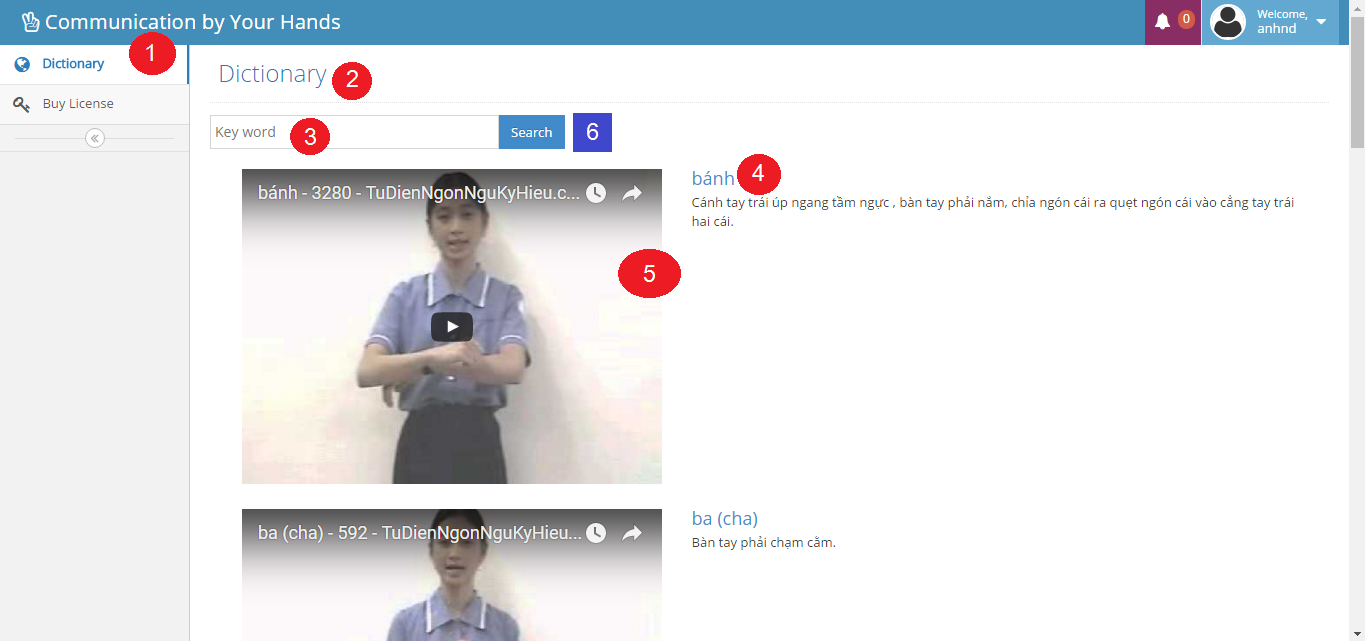


Figure 37: Interface - <User> Search instruction

**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 50: <User> Search instruction fields

**Buttons/Hyperlinks**

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

Table 51: <User> Search instruction buttons/ hyperlinks

##### Edit user profile

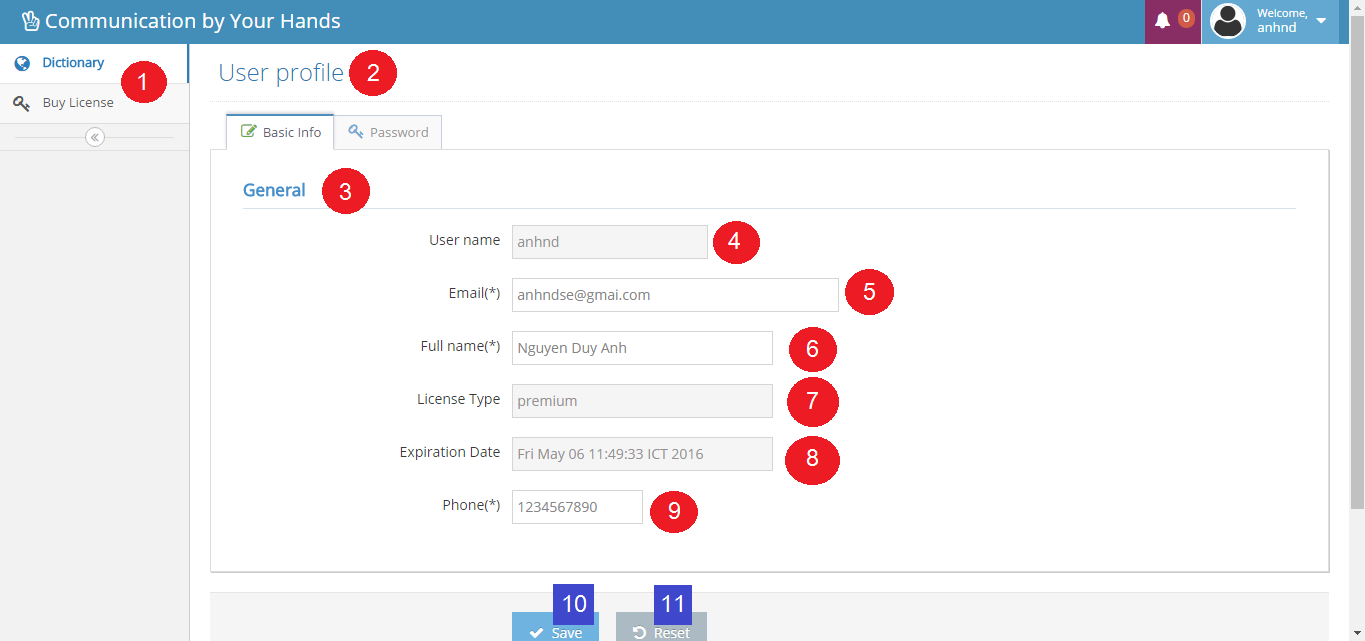


Figure 38: Interface - <User> Edit user profile

**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 52: <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 53: <User> Edit user buttons/hyperlinks

### Mobile Application Design

#### Staff interface design

##### Train online

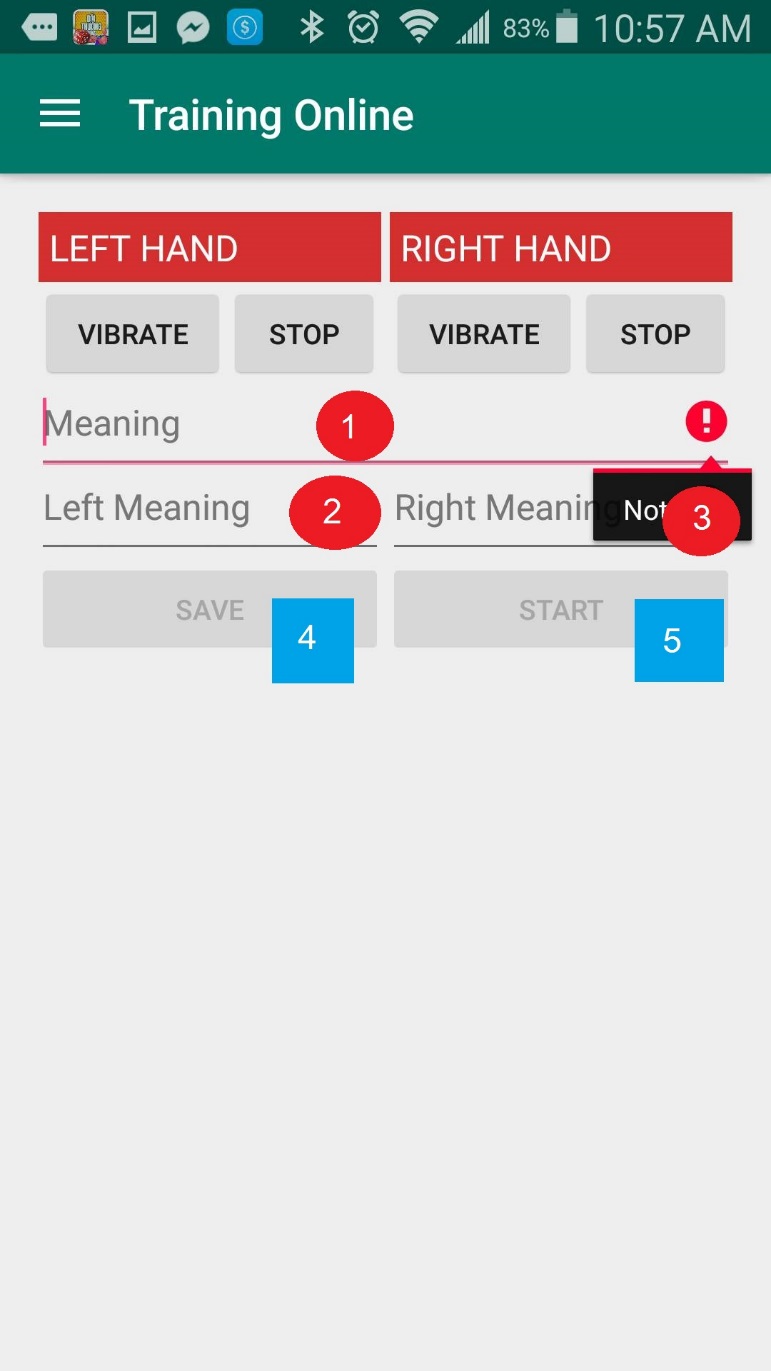


Figure 39: Interface - <Staff> Train online

**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 54: <Staff> Train online fields

**Buttons/Hyperlinks**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No** | **Function** | **Description** | **Validation** | **Outcome** |
| **4** | btnSave | Save new gesture | N/A | N/A |
| **5** | btnStart | Start collecting emg | N/A | N/A |

Table 55: <Staff> Train online buttons/hyperlinks

#### User interface design

##### Login

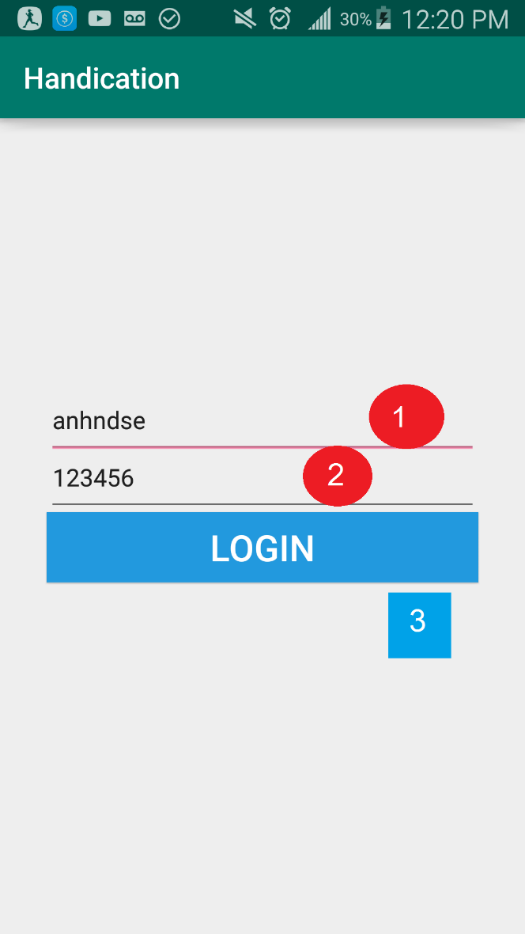


Figure 40: Interface - <User> Login

**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 56: <User> Login fields

**Buttons/Hyperlinks**

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

Table 57: <User> Login buttons/hyperlinls

##### Translate

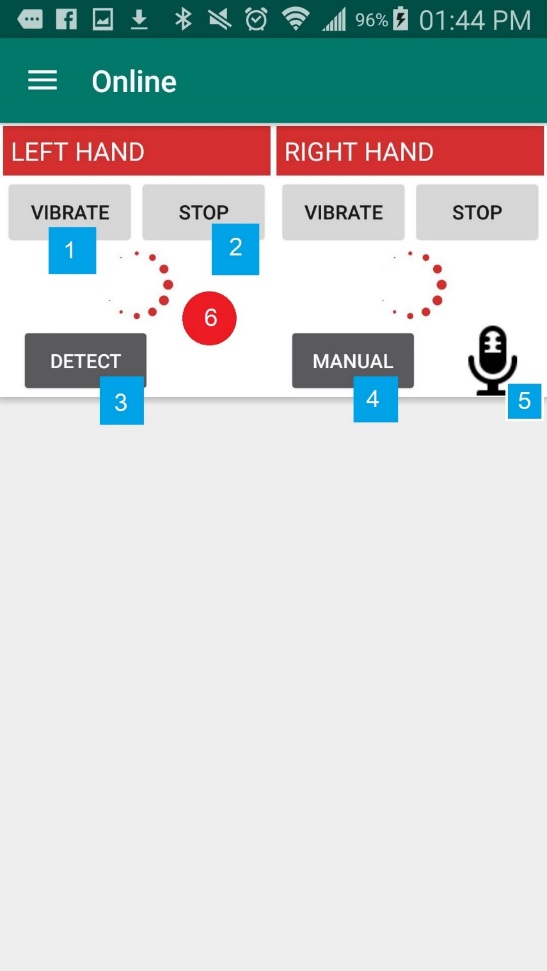


Figure 41: Interface - <User> Translate – Translating

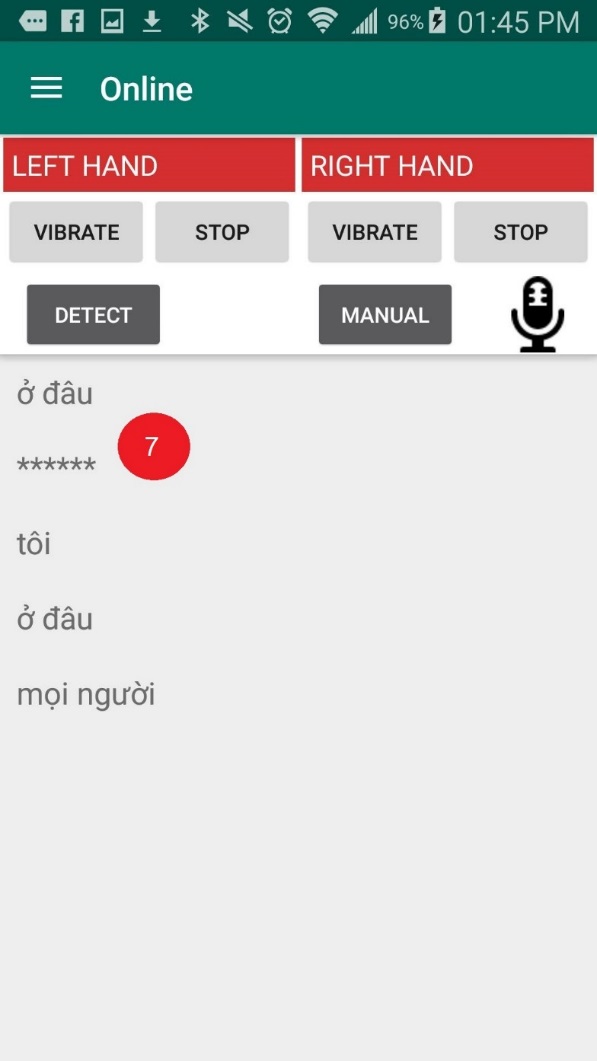


Figure 42: Interface - <User> Translate – Translated

**Fields**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| No | Field Name | Description | Read only | Mandatory | Control Type | Data Type | Length |
| 6 | Process dailog | Processing signal | Yes | Yes | Dailog | N/A | N/A |
| 7 | Result | Translate result | Yes | Yes | List | String | N/A |

Table 58: <User> Translate fields

**Buttons/Hyperlinks**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No** | **Function** | **Description** | **Validation** | **Outcome** |
| **1** | btnVibrate | Active vibrating for the armband | N/A | N/A |
| **2** | btnStop | Stop collect EMG data | N/A | N/A |
| **3** | btnDetect | Start translate automatically | N/A | N/A |
| **4** | btnManual | Start translate manually | N/A | N/A |
| **5** | btnTextToSpeech | Turn the result to speech | N/A | N/A |

Table 59: <User> Translate buttons/hyperlinks

#### Premium user interface design

##### Train offline

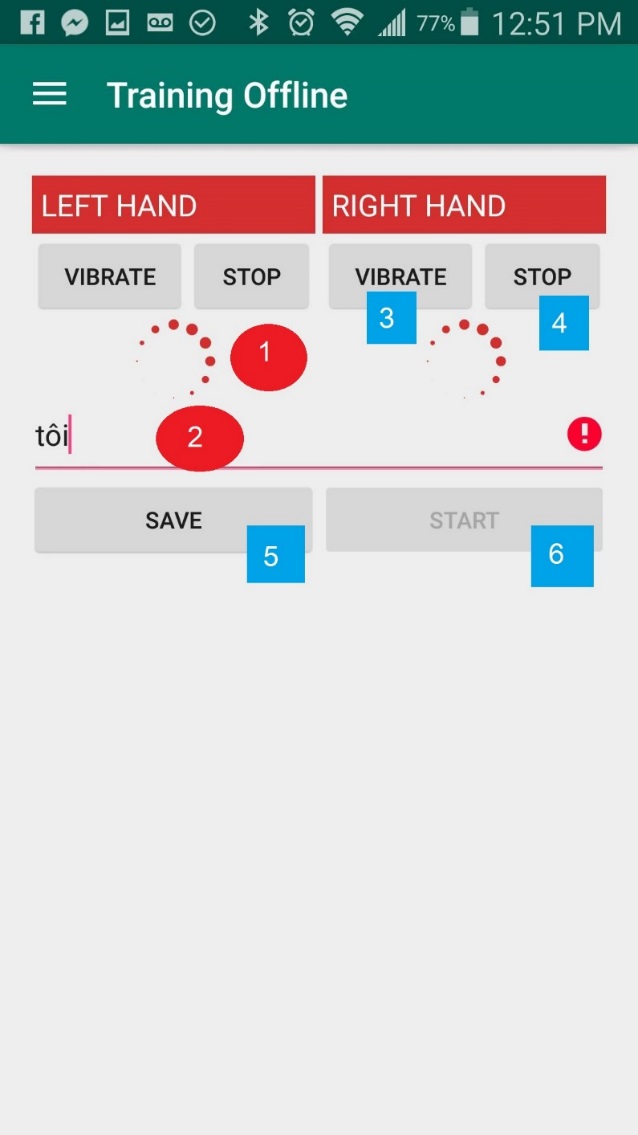


Figure 43: Interface - <Premium user> Train offline

**Fields**

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

Table 60: <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 gesture | N/A | N/A |
| **6** | btnStart | Start collecting emg | N/A | N/A |

Table 61: <Premium user> Train offline Buttons/Hyperlinks

## Database design

### Entity relationship diagram

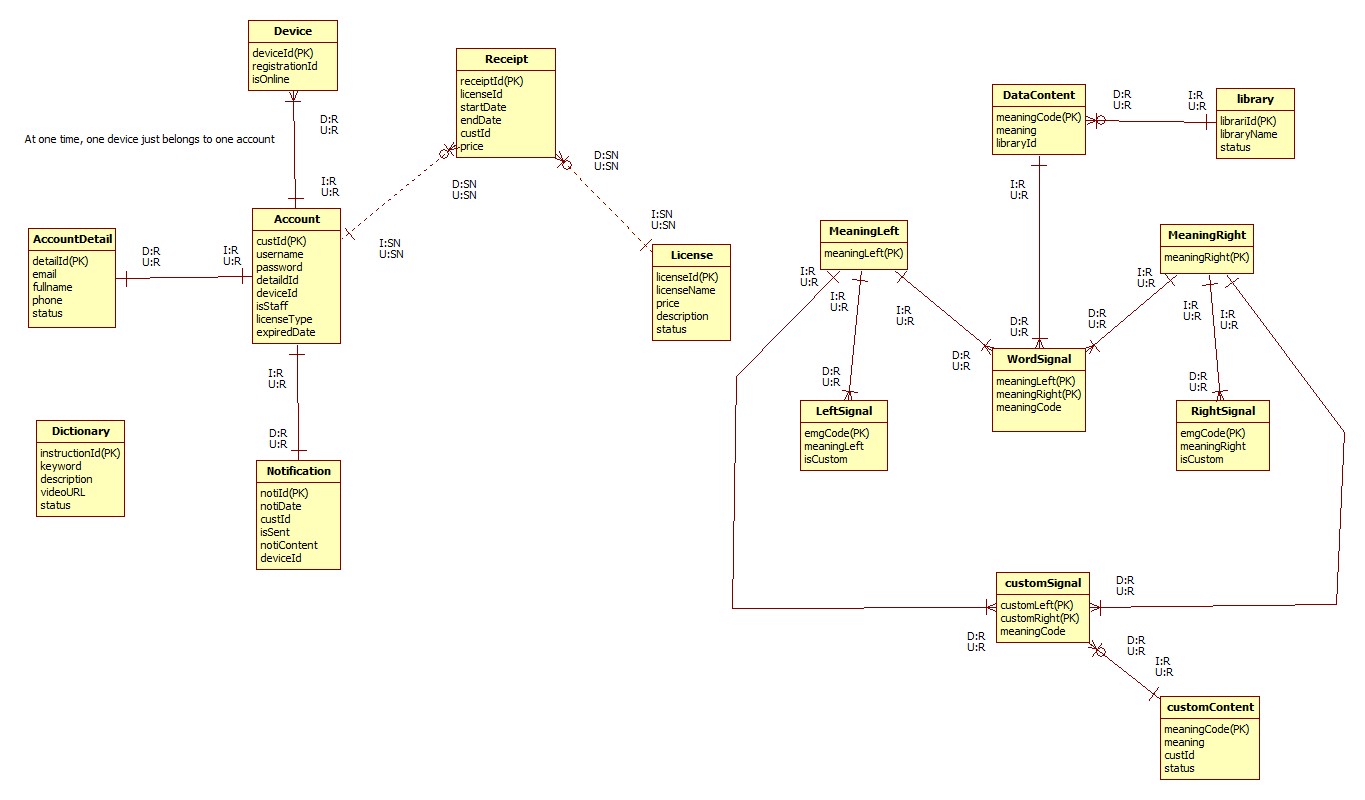


Figure 44: Entity relationship diagram

### Entity dictionary

|  |  |  |
| --- | --- | --- |
| Entity Data Dictionary: describe content of all entities | | |
| Entity name | **Description** | **Mapping column with Conceptual diagram** |
| Account | Contain the account information | user |
| AccountDetail | Not exist in conceptual diagram but necessary for saving detail information of account | N/A |
| Device | Contain the device information | device |
| Notificaton | Contain the notification information | notification |
| License | Contain the license information | license |
| Receipt | Not exist in conceptual diagram but necessary for saving buying history | N/A |
| Dictionary | Contain the dictionary information | dictionary |
| MeaningLeft | Contain the meaning left information | meaningLeft |
| MeaningRight | Contain the meaning right information | meaningRight |
| LeftSignal | Contain the left signal information | leftSignal |
| RightSignal | Contain the right signal information | rightSignal |
| WordSignal | Contain the word signal information | wordSignal |
| DataContent | Contain the data content information | dataContent |
| CustomSignal | Contain the custom signal information | customSignal |
| CustomContent | Contain the custom content information | customContent |
| Library | Contain the library information | library |

Table 62: 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 combine two next EMG datas into one that define the characteristics of a gesture.

#### 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:

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| EMG1 | 4 | 5 | 7 | 6 | 6 | 7 | 2 | 10 |
| EMG2 | 1 | 4 | 10 | 4 | 5 | 7 | 8 | 9 |
| The new EMG | 4 | 5 | 10 | 6 | 6 | 7 | 8 | 10 |

#### Complexity

The Complexity is: O(n)

### 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 gesture 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.

Distance formula to calculate distance between 2 point A (Xa, Ya) and B (Xb, Yb): Distance AB =

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 formula 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)

Threshold = 0.01

Calculate distance AB:

AB = =

Calculate distance OA:

OA= =

Calculate distance OB:

OB = =

CalculateDistance(A,B) = = < threshold number => A is not related with B

#### Complexity

The complexity is : O(n)

#### Flowchart



Figure 45: 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 gesture 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 gesture.

#### 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

#### Complexity

* The complexity of Find meaning left is: O()
* The complexity of Find meaning right is : O()
* The complexity of Find meaning of each hand is: 2O

#### Flowchart



Figure 46: Detect workflow



Figure 47: Find meaning right workflow



Figure 48: Find meaning left workflow

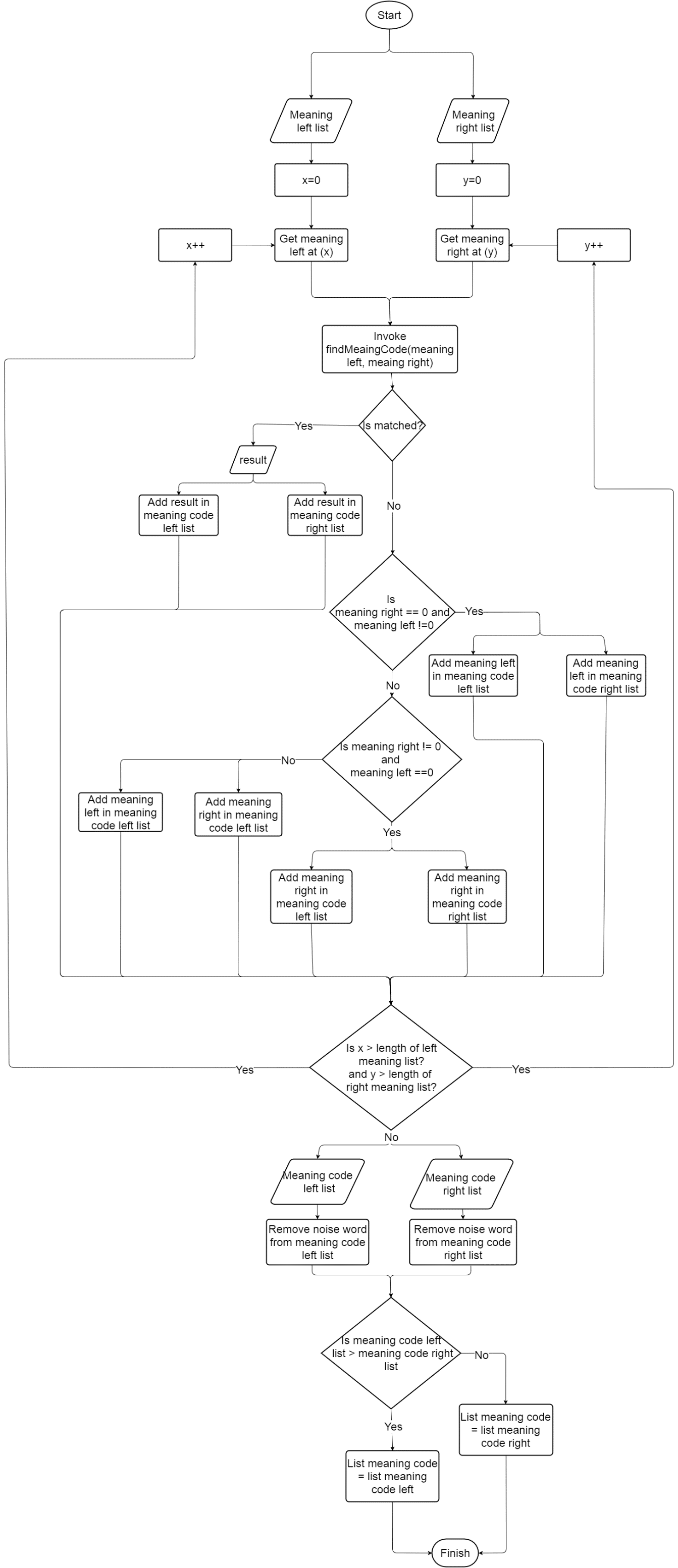


Figure 49: Find meaning of each hand workflow

### Train

#### Definition

Train is the method that teach for the system learn the sign language and meaning of it to prepare for translate function.

#### Define problem

The system needs to know the EMG data and the meaning of it to recognize it as a gesture for translate function. However, a gesture is describe by many different EMG data, Every pair of EMG datas will be matched with a meaning code that match with a meaning. So there must be a flow to input data correctly.

#### Complexity

The complexity is: O(1)

#### Flowchart

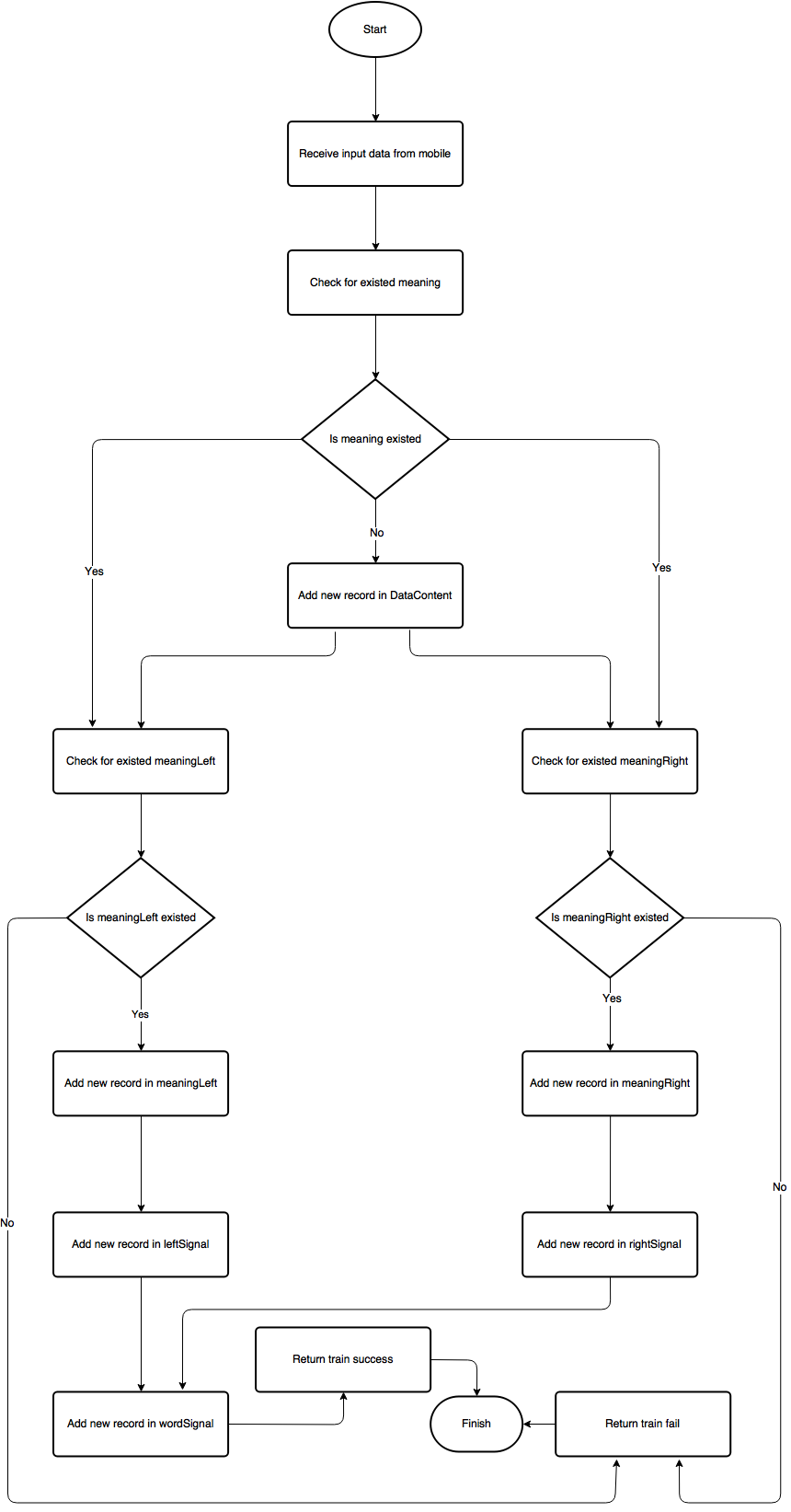


Figure 50: Train workflow

### Standardize EMG data before input to Database (Server)

#### Definition

This is the method to specify all EMG data input to database will be in a specific area, to support matching algorithm will not misunderstand about the force of gesture.

#### Define problem

When staff perform sign language with too much or less force in train process will decrease the accuracy of translate function.

#### Solution

We set an EMG data as point A in eight dimensions space with the Origin is O.

A(a1, a2, a3, a4, a5, a6, a7, a8)

We calculate the norm of vector OA by the following formula:

Norm(OA) =

We use a number , call K, to standardize each number in a EMG data to create a new EMG data with different value but still describe a same gesture.

K = 50/Norm()

a1’ = ka1, a2’ = ka2, a3’ = ka3, a4’ = ka4, a5’ = ka5, a6’ = ka6, a7’ = ka7, a8’ = ka8

=> The new EMG data: (a1’, a2’, a3’, a4’, a5’, a6’, a7’, a8’)

#### Example

We have an EMG:

A (5,6,7,8,4,5,6,7)

Norm(OA) = = 10

K = = = 2.8867513459481

New EMG = (14.4337567297405, 17.3205080756886, 20.2072594216367, 23.0940107675848, 11.5470053837924, 14.4337567297405, 17.3205080756886, 20.2072594216367)

# Report No.5 System implement and Test

## Introduction

### Overview

This section describes the approach and methodologies used by group to plan, organize and manage the testing of CBYH system. It provides in the detail all necessary information about the implementation and testing procedure of the system included test plans, test cases, test result, test environments, pass/fail criteria and risks estimations as well as a checklist to cover all possible cases.

### Test Approach

* Goal: Test all features in the whole CBYH system based on the core flow.
* Method: black-box testing
* Technique: check list

The testing for this project will consists of Integration System test level. Testing the program which was integrated and as a complete system to ensure that the software requirements have been met.

* Integration testing would be performed by all member of team and approved by team leader.
* System testing is focused on assessing the system’s reliability. This process is concerned with finding errors that result from unanticipated interactions between components and component interface problems.

## Database relationship diagram

### Physical diagram

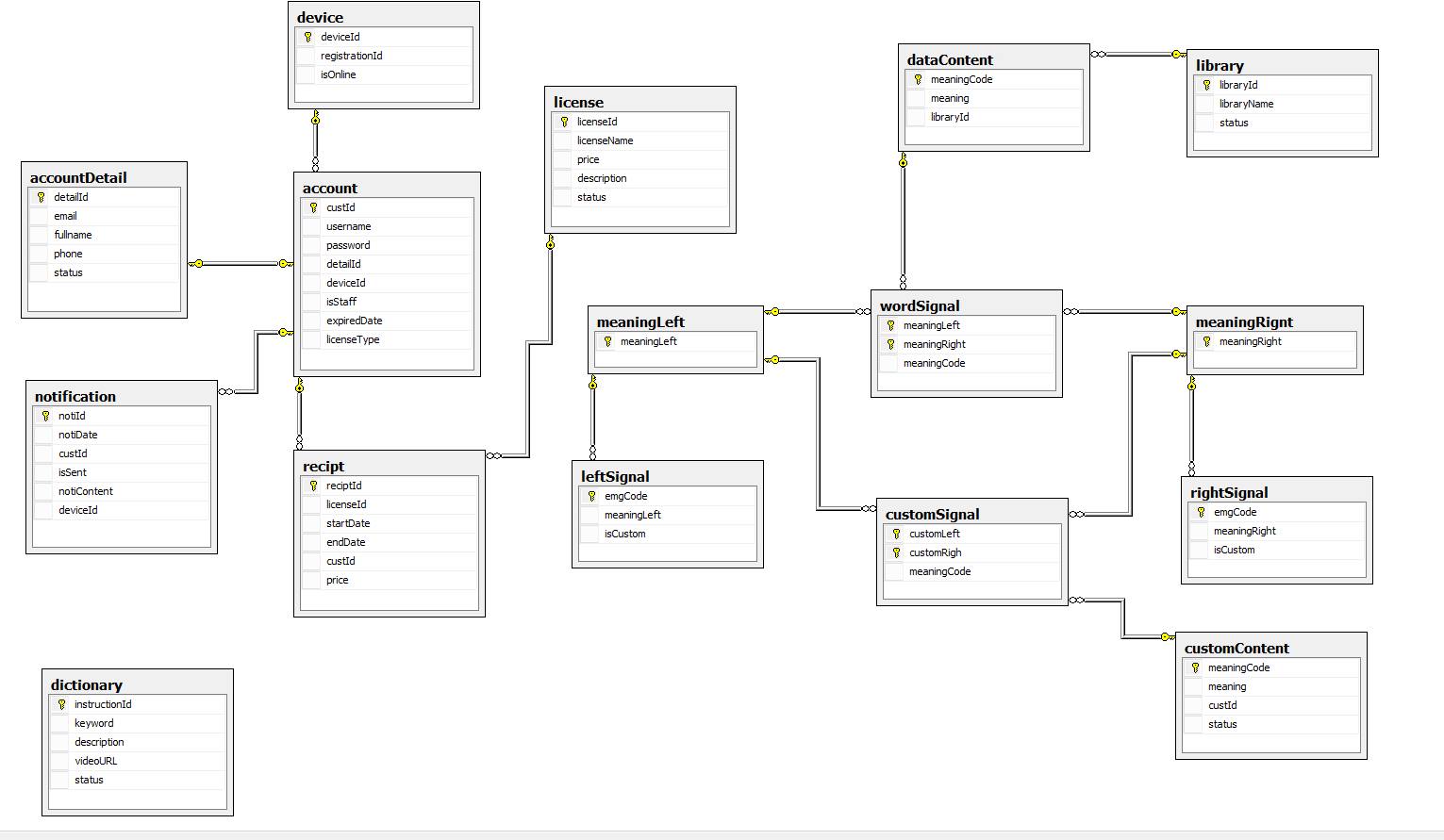


Figure 51: Physical diagram

|  |  |  |
| --- | --- | --- |
| Data dictionary: describe content of all tables | | |
| Table Name | **Mapping with Conceptual diagram** | **Description** |
| Account | user | Contain information of account |
| Account detail | N/A | Not exist in conceptual diagram but need to store detail information of account |
| Device | device | Contain information of device |
| License | license | Contain information of license |
| Receipt | N/A | Not exist in conceptual diagram but need to store receipt when user buy license |
| Notification | notification | Contain information of notification |
| Dictionary | dictionary | Contain information of dictionary |
| Notification | notification | Contain information of notification |
| WordSignal | wordSignal | Contain information of word signal |
| LeftSignal | leftSignal | Contain information of left signal |
| RightSignal | rightSignal | Contain information of right signal |
| MeaningLeft | meaningLeft | Contain information of meaning left |
| MeaningRight | meaningRight | Contain information of meaning right |
| DataContent | dataContent | Contain information of data content |
| Library | library | Contain information of library |
| CustomSignal | customSignal | Contain information of custom signal |
| CustomContent | customContent | Contain information of custom content |

### Data dictionary

Table 63: Data table dictionary

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Table Name | Attributes | Description | Domain | Null |
| Account | custId {PK} | Unique identifier of Account | int | No |
| username | Username of Account | nvarchar(20) | No |
| password | Password of Account | nvarchar(12) | No |
| detailId | Identifier of the detail of Account | int | No |
| deviceId | Identifier of the device of Account | int | Yes |
| isStaff | Staff checker for authority | bit | No |
| licenseType | Type of license that Account is owning | varchar(20) | No |
| expiredDate | Expired date of license that Account is owning | datetime | Yes |
| AccountDetail | deatilId(PK) | Unique Identifier of AccountDetail | int | No |
| email | Email of Account | varchar(254) | No |
| fullname | Fullname of user of Account | nvarchar(50) | No |
| phone | Phone number of user of Account | varchar(12) | Yes |
| status | Active checker of Account | bit | No |
| Device | deviceId(PK) | Unique identifider of Device | nvarchar(50) | No |
| registrationId | Registration identifier of Device | nvarchar(250) | No |
| Notification | notiId(PK) | Unique Identifier of Notification | int | No |
| notiDate | Date the notification is created | datetime | No |
| custId | Identifier of owner of Notification | int | No |
| isSent | Sent checker of Notification | bit | No |
| notiContent | Content of Notification | nvarchar(250) | No |
| deviceId | Identifier of device that notification will be sent to | nvarchar(50) | No |
| Dictionary | intructionId(PK) | Unique Identifier of instruction | int | No |
| keyword | Keyword of instruction | nvarchar(20) | No |
| description | Description of instruction | nvarchar(250) | Yes |
| videoURL | URL of video that describe the instruction | nvarchar(225) | No |
| status | Active checker of instruction | bit | No |
| License | licenseId(PK) | Unique Identifier of License | int | No |
| licenseName | Name of License | nvarchar(20) | No |
| price | Price of License | float | No |
| description | Description of License | nvarchar(250) | No |
| status | Active checker of License | bit | No |
| Receipt | receiptId(PK) | Unique Identifier of Receipt | int | No |
| licenseId | Identifier of license that is bought by receipt owner | int | No |
| startDate | The date that license usage time starts count down | datetime | No |
| endDate | The date that license will be expired | datetime | No |
| custId | Identifier of account own the receipt | int | No |
| price | Price of the license which is bought | float | No |
| MeaningLeft | meaningLeft(PK) | Unique identifier of MeaningLeft | int | No |
| MeaningRight | meaningRight(PK) | Unique identifier of MeaningRight | int | No |
| LeftSignal | emgCode(PK) | Emg code of left hand signal and also unique identifier of LeftSignal | varchar(250) | No |
| meaningLeft | Meaning code for LeftSignal | int | No |
| isCustom | Custom guesture checker | bit | No |
| RightSignal | emgCode(PK) | Emg code of right hand signal and also unique identifier of rightSignal | varchar(250) | No |
| meaningRight | Meaning code for RightSignal | int | No |
| isCustom | Custom guesture checker | bit | No |
| WordSignal | meaningLeft(PK) | Half of Unique identifier of WordSignal | int | No |
| meaningRight(PK) | Half of Unique identifier of WordSignal | int | No |
| meaningCode | Meaning code of WordSignal | int | No |
| DataContent | meaningCode(PK) | Unique identifier of DataContent | int | No |
| meaning | Meaning of a guesture | nvarchar(50) | No |
| libraryId | Identifier of library that the meaning belongs | int | No |
| Library | libraryId | Unique identifier of library | int | No |
| libraryName | Name of library | nvarchar(50) | No |
| status | Active checker of library | bit | No |
| CustomSignal | customLeft(PK) | Half of Unique identifier of CustomSignal | int | No |
| customRight(PK) | Half of Unique identifier of CustomSignal | int | No |
| meaningCode | Meaning code of CustomSignal | int | No |
| CustomContent | meaningCode | Unique identifier of CustomContent | int | No |
| meaning | Meaning of a custom guesture | nvarchar(50) | No |
| custId | Identifier of Account that create the custom guesture | int | No |
| status | Active checker of CustomContent | int | No |

Table 64: Data table dictionary

## Performance measures

### Mobile Application API load speed

#### Definition

This sections tests the load speed of mobile app when connect to server through API

#### Test environment

**Server**

* Operating System: Ubuntu 14.04 Server 32 bit
* RAM: 4GB
* Storage: 50GB
* Processor: Intel® CORE i7 Quad core 2.4 GHz
* Network: Wi-Fi 1Mbps

**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 600 1.89GHz Quad Core or higher |
| **Memory** | 512MB RAM | 2GB |
| **Bluetooth** | Bluetooth 4.0 required | Bluetooth 4.0 required |

Table 65: Mobile API load speed client specification

#### Test cases

Using mobile application to send entire API request to web server

List of APIs:

|  |  |  |
| --- | --- | --- |
| No. | API | Description |
| 1 | doTrain | Check server status |
| 2 | doTranslate | Get contract information |
| 3 | doLoginFromMobile | Login from mobile |

Table 66: Mobile API load speed test cases

#### Test result

The test is run 20 times, each time we run all the test cases listed above and log down the average API response time.

|  |  |  |
| --- | --- | --- |
| Test No. | Average page load time (second) | Execute date |
| 1 | 0.85 | 12 April 2016 |
| 2 | 1.1 | 12 April 2016 |
| 3 | 0.7 | 12 April 2016 |
| 4 | 0.9 | 12 April 2016 |
| 5 | 1.3 | 12 April 2016 |
| 6 | 0.75 | 12 April 2016 |
| 7 | 0.8 | 12 April 2016 |
| 8 | 1.2 | 12 April 2016 |
| 9 | 1 | 12 April 2016 |
| 10 | 0.93 | 12 April 2016 |
| 11 | 0.89 | 12 April 2016 |
| 12 | 0.8 | 12 April 2016 |
| 13 | 0.83 | 12 April 2016 |
| 14 | 0.93 | 12 April 2016 |
| 15 | 1.1 | 12 April 2016 |
| 16 | 1 | 12 April 2016 |
| 17 | 0.9 | 12 April 2016 |
| 18 | 0.85 | 12 April 2016 |
| 19 | 0.83 | 12 April 2016 |
| 20 | 0.93 | 12 April 2016 |
|  | **Average: 0.93(seconds)** |  |

Table 67: Mobile API load speed test result

## Test plan

The overall purpose of testing is to ensure CBYH system meets its entire technical, functional and business requirement. The purpose of this document is to describe the overall test plan and strategy for testing the CBYH system. The approach described in this document provides the framework for all testing related to this application. Individual test cases will be written for each version of the application that is released. This document will also be updated as required for each release.

### Features to be tested

* Staff: Train online
* User: Translate online, buy license
* System: Notify schedule
* Premium user: Train offline, translate offline

### Features not to be tested

* Guest: Search, login, logout, edit profile
* User: Search instruction

## System testing test case

### Communication diagram

Test cases are created from below communication diagram



Figure 52: Mobile application communication diagram



Figure 53: Web application communication



Figure 54: Scheduler communication

### Test cases

#### <Staff>-<User>Train Online-Translate Online

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| ID | Test Case Description | Precondition | Test case procedure | Expected output | Inter-test case dependence | Result | Test Date |
| TO-TO 1 | Test Staff train online first time | Database for translate online function is blank | 1.Staff input new sign language gesture and with meaning “Tôi” five times | 1.System shows “Save successfully” | N/A | Pass | 4/14/16 |
| TO-TO 2 | Test User translate online with one word in database | There is at least one record in database for translate online function | 1.User performs sign language: “Tôi”  2.User performs end sign command | 2.System show result “Tôi” | TO – TO 1 | Pass | 4/14/16 |
| TO- TO 3 | Test Staff train online second time | N/A | 1.Staff input new sign language gesture and with meaning “Thích” five times | 1.System shows “Save successfully” | TO – TO 1 | Pass | 4/14/16 |
| TO- TO 4 | Test User translate online one word with two words in database | There is at least two records in database for translate online function | 1.User performs sign language: “Thích”  2.User performs end sign command  3.User performs sign language: “Tôi”  4.User performs end sign command | 2.System show result “Thích”  4.System show result “Tôi” | TO – TO 3 | Pass | 4/14/16 |
| TO- TO 5 | Test User translate online two words right order with two words in database | There is at least two records in database for translate online function | 1.User performs sign language: “Tôi”  2. User performs sign language: “Thích”  3. User performs end sign command | 3.System show result “Tôi Thích” | TO – TO 3 | Pass | 4/14/16 |
| TO- TO 6 | Test User translate online two words reverse order with two words in database | There is at least two records in database for translate online function | 1.User performs sign language: “Thích”  2. User performs sign language: “Tôi”  3. User performs end sign command | 3.System show result “Thích Tôi” | TO – TO 3 | Pass | 4/14/16 |
| TO- TO 7 | Test Staff train online third time | N/A | 1.Staff input new sign language gesture and with meaning “Bạn” five times | 1.System shows “Save successfully” | TO – TO 3 | Pass | 4/14/16 |
| TO- TO 8 | Test User translate online one word with three words in database | There is at least three records in database for translate online function | 1.User performs sign language: “Thích”  2.User performs end sign command  3.User performs sign language: “Tôi”  4.User performs end sign command  5.User performs sign language: “Bạn”  6.User performs end sign command | 2.System show result “Thích”  4.System show result “Tôi”  6.System show result “Bạn” | TO-TO 7 | Pass | 4/14/16 |
| TO- TO 9 | Test User translate online three words right order with three words in database | There is at least three records in database for translate online function | 1.User performs sign language: “Tôi”  2. User performs sign language: “Thích”  3. User performs sign language: “Bạn”  4. User performs end sign command | 4.System show result “Tôi Thích Bạn” | TO-TO 7 | Pass | 4/14/16 |
| TO- TO 10 | Test User translate online three words mix order with three words in database | There is at least three records in database for translate online function | 1.User performs sign language: “Thích”  2. User performs sign language: “Tôi”  3. User performs sign language: “Bạn”  4. User performs end sign command | 4.System show result “Thích Tôi Bạn” | TO-TO 7 | Pass | 4/14/16 |
| TO- TO 11 | Test User translate online three words mix order with three words in database | There is at least three records in database for translate online function | 1.User performs sign language: “Thích”  2. User performs sign language: “Bạn”  3. User performs sign language: “Tôi”  4. User performs end sign command | 4.System show result “Thích Bạn Tôi” | TO-TO 7 | Pass | 4/14/16 |
| TO- TO 12 | Test User translate online three words mix order with three words in database | There is at least three records in database for translate online function | 1.User performs sign language: “Tôi”  2. User performs sign language: “Bạn”  3. User performs sign language: “Thích”  4. User performs end sign command | 4.System show result “Tôi Bạn Thích” | TO-TO 7 | Pass | 4/14/16 |
| TO- TO 13 | Test User translate online three words mix order with three words in database | There is at least three records in database for translate online function | 1.User performs sign language: “Bạn”  2. User performs sign language: “Tôi”  3. User performs sign language: “Thích”  4. User performs end sign command | 4.System show result “Bạn Tôi Thích” | TO-TO 7 | Pass | 4/14/16 |
| TO- TO 14 | Test User translate online three words reverse order with three words in database | There is at least three records in database for translate online function | 1.User performs sign language: “Bạn”  2. User performs sign language: “Thích”  3. User performs sign language: “Tôi”  4. User performs end sign command | 4.System show result “Bạn Thích Tôi” | TO-TO 7 | Pass | 4/14/16 |
| TO- TO 15 | Test Staff train online the forth time | N/A | 1.Staff input new sign language gesture and with meaning “Lắm” five times | 1.System shows “Save successfully” | TO-TO 7 | Pass | 4/14/16 |
| TO- TO 16 | Test User translate online one words with four words in database | There is at least four records in database for translate online function | 1.User performs sign language: “Thích”  2.User performs end sign command  3.User performs sign language: “Tôi”  4.User performs end sign command  5.User performs sign language: “Bạn”  6.User performs end sign command  7.User performs sign language: “Lắm”  8.User performs end sign command | 2.System show result “Thích”  4.System show result “Tôi”  6.System show result “Bạn”  8.System show result “Lắm” | TO-TO 15 | Pass | 4/14/16 |
| TO- TO 17 | Test User translate online two words with four words in database | There is at least four records in database for translate online function | 1.User performs sign language: “Bạn”  2. User performs sign language: “Tôi”  3.User performs end sign command | 3.System show result contains these words: “Bạn Tôi” | TO-TO 15 | Pass | 4/14/16 |
| TO- TO 18 | Test User translate online two words with four words in database | There is at least four records in database for translate online function | 1.User performs sign language: “Thích”  2. User performs sign language: “Lắm”  3.User performs end sign command | 3.System show result contains these words: “Thích Lắm” | TO-TO 15 | Pass | 4/14/16 |
| TO- TO 19 | Test User translate online three words with four words in database | There is at least four records in database for translate online function | 1.User performs sign language: “Tôi”  2. User performs sign language: “Thích”  3. User performs sign language: “Lắm”  4.User performs end sign command | 4.System show result contains these words: “Tôi Thích Lắm” | TO-TO 15 | Pass | 4/14/16 |
| TO- TO 20 | Test User translate online three words with four words in database | There is at least four records in database for translate online function | 1.User performs sign language: “Bạn”  2. User performs sign language: “Tôi”  3. User performs sign language: “Thích”  4.User performs end sign command | 4.System show result contains these words: “Bạn Tôi Thích” | TO-TO 15 | Pass | 4/14/16 |
| TO- TO 21 | Test User translate online four words with four words in database | There is at least four records in database for translate online function | 1.User performs sign language: “Tôi”  2. User performs sign language: “Thích”  3. User performs sign language: “Bạn”  4. User performs sign language: “Lắm”  5.User performs end sign command | 5.System show result contains these words: “Tôi Thích Bạn Lắm” | TO-TO 15 | Pass | 4/14/16 |
| TO- TO 22 | Test User translate online four words with four words in database | There is at least four records in database for translate online function | 1.User performs sign language: “Bạn”  2. User performs sign language: “Tôi”  3. User performs sign language: “Thích”  4. User performs sign language: “Lắm”  5.User performs end sign command | 5.System show result contains these words: “Bạn Tôi Thích Lắm” | TO-TO 15 | Pass | 4/14/16 |

Table 68: Test case <Staff>-<User> Train Online – Translate Online

#### <Staff> Train Online

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| ID | Test Case Description | Precondition | Test case procedure | Expected output | Inter-test case dependence | Result | Test Date |
| TrainO 1 | Test Staff train online server is running ok | N/A | 1.Staff input new sign language gesture and with meaning “Tôi” | 1.System shows “Save successfully” | N/A | Pass | 4/14/16 |
| TrainO 2 | Test Staff train online with server is error | N/A | 1.Staff input new sign language gesture and with meaning “Tôi” | 1.System shows “Server Error” | N/A | Pass | 4/14/16 |
| TrainO 3 | Test Staff train online with no internet connection | N/A | 1.Staff input new sign language gesture and with meaning “Tôi” | 1.System shows “Cannot connect to server” | N/A | Pass | 4/14/16 |

Table 69: Test case <Staff> Train Online

#### <User> Translate Online

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| ID | Test Case Description | Precondition | Test case procedure | Expected output | Inter-test case dependence | Result | Test Date |
| TranslateO 1 | Test User translate online server is running ok | N/A | 1.User performs sign language: “Bạn”  2. User performs sign language: “Tôi”  3. User performs sign language: “Thích”  4. User performs sign language: “Lắm”  5.User performs end sign command | 5.System show result contains these words: “Bạn Tôi Thích Lắm” | N/A | Pass | 4/14/16 |
| TranslateO 2 | Test User translate online with the word is not in database | Database does not have record “Yêu” | 1.User performs sign language: “Yêu”  2.User performs end sign command | 2.System show result: “No Gesture” | N/A | Pass | 4/14/16 |
| TranslateO 3 | Test User translate online with the word is not in database | Database does not have record “Yêu” | 1.User performs sign language: “Tôi”  2.User performs sign language: “Yêu”  3.User performs sign language: “Bạn”  4.User performs end sign command | 2.System show result contains: “Tôi Bạn” | N/A | Pass | 4/14/16 |
| TranslateO 4 | Test User translate with server is error | N/A | 1.User performs sign language: “Bạn”  2. User performs sign language: “Tôi”  3. User performs sign language: “Thích”  4. User performs sign language: “Lắm”  5.User performs end sign command | 5.System shows “Server Error | N/A | Pass | 4/14/16 |
| TranslateO 5 | Test User translate with no internet connection | N/A | 1.User performs sign language: “Bạn”  2. User performs sign language: “Tôi”  3. User performs sign language: “Thích”  4. User performs sign language: “Lắm”  5.User performs end sign command | 5.System shows “Cannot connect to server” | N/A | Pass | 4/14/16 |
| TranslateO 6 | Test User translate one word with database cantains many data | There is at least 30 word in database | 1.User performs sign language: “Bạn”  2.User performs end sign command | 2.System show result contains: “Bạn” | N/A | Pass | 4/14/16 |
| TranslateO 7 | Test User translate a senctene word with database contains many data | There is at least 30 word in database | 1.User performs sign language: “Bạn”  2. User performs sign language: “Tôi”  3. User performs sign language: “Thích”  4. User performs sign language: “Lắm”  5.User performs end sign command | 5.System show result contains these words: “Bạn Tôi Thích Lắm” | N/A | Pass | 4/14/16 |
| TranslateO 8 | Test User translate and play sound with internet connection | Smartphone is connected to the internet | 1.User performs sign language: “Bạn”  2. User performs sign language: “Tôi”  3. User performs sign language: “Thích”  4. User performs sign language: “Lắm”  5.User performs end sign command  6.User remove wrong words  7.User press play sound button | 5.System show result contains these words: “Bạn Tôi Thích Lắm”  7. System plays sound:  “Bạn Tôi Thích Lắm” | N/A | Pass | 4/14/16 |
| TranslateO 9 | Test User translate and play sound with no internet connection | Smartphone is not connected to the internet | 1.User performs sign language: “Bạn”  2. User performs sign language: “Tôi”  3. User performs sign language: “Thích”  4. User performs sign language: “Lắm”  5.User performs end sign command  6.User remove wrong words  7.User press play sound button | 5.System show result contains these words: “Bạn Tôi Thích Lắm”  7. System plays sound:  “Bạn Tôi Thích Lắm” | N/A | Pass | 4/14/16 |

Table 70: Test case <User> Translate Online

#### <User> Buy license

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| ID | Test Case Description | Precondition | Test case procedure | Expected output | Inter-test case dependence | Result | Test Date |
| BL1 | Test User buy lincese first time | current system date is 1/4/2016  License type of user is basic | 1.User goes to buy license page  2.User clicks “Buy” button  3.User login Paypal with paypal account  4.User complete Paypal payment process | 5. System shows payment success page. License type turn into premium with expired date is 1/5/2016 | N/A | Pass | 4/14/16 |
| BL2 | Test User buy license second time | current system date is 1/4/2016  License type of user is premium | 1.User goes to buy license page  2.User clicks “Buy” button  3.User login Paypal with paypal account  4.User complete Paypal payment process | 5. System shows payment success page. License type turn into premium with expired date is 30/5/2016 | BL1 | Pass | 4/14/16 |
| BL3 | Test User buy license third time | current system date is 1/4/2016  License type of user is premium | 1.User goes to buy license page  2.User clicks “Buy” button  3.User login Paypal with paypal account  4.User complete Paypal payment process | 5. System shows payment success page. License type turn into premium with expired date is 29/6/2016 | BL2 | Pass | 4/14/16 |
| BL4 | Test User cancel payment | N/A | 1.User goes to buy license page  2.User clicks “Buy” button  3.User cancel Payment | 3. System shows payment fail page. | N/A | Pass | 4/14/16 |
| BL5 | Test User cancel payment | N/A | 1.User goes to buy license page  2.User clicks “Buy” button  3.User login Paypal with paypal account  4.User cancel Payment | 4. System shows payment fail page. | N/A | Pass | 4/14/16 |

Table 71: Test case <User> Buy license

#### <Premium User> Train Offline – Translate Offline

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| ID | Test Case Description | Precondition | Test case procedure | Expected output | Inter-test case dependence | Result | Test Date |
| TOFF-TOFF 1 | Test Premium trains offline | Smartphone is not connected to the internet | 1.Premium input new sign language gesture and with meaning “Tôi xin chào” five times | 1.System shows “Save successfully” | N/A | Pass | 4/14/16 |
| TOFF-TOFF 2 | Test Premium user translates online with one record in database | Smartphone is not connected to the internet  There is at least one record in database for translate online function | 1.User performs sign language: “Tôi xin chào”  2.User performs end sign command | 2.System show result “Tôi xin chào” | TO – TO 1 | Pass | 4/14/16 |
| TOFF-TOFF 3 | Test Premium trains offline second time | Smartphone is not connected to the internet | 1.Premium input new sign language gesture and with meaning “Rất vui được gặp bạn” five times | 1.System shows “Save successfully” | N/A | Pass | 4/14/16 |
| TOFF-TOFF 4 | Test Premium user translates offline with two record word in database | Smartphone is not connected to the internet  There is at least one record in database for translate online function | 1.User performs sign language: “Tôi xin chào”  2.User performs sign language: “Rất vui được gặp bạn”  3.User performs end sign command | 2.System show result “Tôi xin chào Rất vui được gặp bạn” | TO – TO 3 | Pass | 4/14/16 |
| TOFF-TOFF 5 | Test Premium user translates online with two record in database | Smartphone is not connected to the internet  There is at least one record in database for translate online function | 1.User performs sign language: “Tôi xin chào”  2.User performs end sign command | 2.System show result “Tôi xin chào” | TO – TO 1 | Pass | 4/14/16 |
| TOFF-TOFF 6 | Test Premium user translates online with two record in database | Smartphone is not connected to the internet  There is at least one record in database for translate online function | 1.User performs sign language: “Rất vui được gặp bạn”  2.User performs end sign command | 2.System show result “Rất vui được gặp bạn” | TO – TO 1 | Pass | 4/14/16 |

Table 72: Test case <Premium User> Train Offline – Translate Offline

#### <Premium User> Train Offline

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| ID | Test Case Description | Precondition | Test case procedure | Expected output | Inter-test case dependence | Result | Test Date |
| TrainOff 1 | Test Premium trains offline  With no internet connection | Smartphone is not connected to the internet | 1.Premium input new sign language gesture and with meaning “Tôi xin chào” | 1.System shows “Save successfully” | N/A | Pass | 4/14/16 |
| TrainOff 2 | Test Premium trains offline  With internet connection | Smartphone is connected to the internet | 1.Premium input new sign language gesture and with meaning “Tôi xin chào” | 1.System shows “Save successfully” | N/A | Pass | 4/14/16 |
| TrainOff 3 | Test Premium User access train offline function after license is expired | Premium user’s license is expired | 1.Premium user press train offline button | 1.Train offline button is disable | N/A | Pass | 4/14/16 |

Table 73: Test case <Premium User> Train Offline

#### <Premium User> Translate Offline

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| ID | Test Case Description | Precondition | Test case procedure | Expected output | Inter-test case dependence | Result | Test Date |
| TranslateOff 1 | Test Premium translate offline  with no internet connection | Smartphone is not connected to the internet | 1.Premium user performs sign language “Tôi xin chào”  2.Premium user performs end sign command | 2.System show result “Tôi xin chào” | N/A | Pass | 4/14/16 |
| TranslateOff 2 | Test Premium translate offline  with internet connection | Smartphone is connected to the internet | 1.Premium user performs sign language “Tôi xin chào”  2.Premium user performs end sign command | 2.System show result “Tôi xin chào” | N/A | Pass | 4/14/16 |
| TranslateOff 3 | Test Premium translate offline and play sound  with no internet connection | Smartphone is not connected to the internet | 1.Premium user performs sign language “Tôi xin chào”  2.Premium user performs end sign command  3. Premium user press play sound button | 2.System show result “Tôi xin chào”  3.System play sound “Tôi in chào” | N/A | Pass | 4/14/16 |
| TranslateOff 4 | Test Premium translate offline and play sound  with internet connection | Smartphone is connected to the internet | 1.Premium user performs sign language “Tôi xin chào”  2.Premium user performs end sign command  3. Premium user press play sound button | 2.System show result “Tôi xin chào”  3.System play sound “Tôi in chào” | N/A | Pass | 4/14/16 |
| TranslateOff 5 | Test Premium User access translate offline function after license is expired | Premium user’s license is expired | 1.Premium user press offline button | 1. Offline button is disable | N/A | Pass | 4/14/16 |

Table 74: Test case <Premium User> Translate Offline

#### <System> Notify schedule

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| ID | Test Case Description | Precondition | Test case procedure | Expected output | Inter-test case dependence | Result | Test Date |
| NS 1 | Test System creates notification | There is a configuration file that set time for notification creating at 00:00  There is at least one user has license will be expired in under five days | 1.Set system time to 00:00 | 1.Notification for users whom license will be expired in under five days is create | N/A | Pass | 4/14/16 |
| NS 2 | Test System send notification | There is a configuration file that set time for notification sending at 07:00  There is at least one unsent notification in database | 1.Set system time to 07:00 | 1.Unsent notification is sent to mobile on time | N/A | Pass | 4/14/16 |
| NS3 | Test create notification with time does not met the configuration | There is a configuration file that set time for notification creating at 00:00  There is at least one user has license will be expired in under five days | 1.Set system time to 3:00 | 1.Nothing happens, no notification is created | N/A | Pass | 4/14/16 |
| NS4 | Test System send notification with time does not met the configuration | There is a configuration file that set time for notification sending at 07:00  There is at least one unsent notification in database | 1.Set system time to 3:00 | 1.Nothing happens, no notification is sent to mobile device | N/A | Pass | 4/14/16 |
| NS5 | Test System send notification after change mobile device local time | There is a configuration file that set time for notification sending at 07:00  There is at least one unsent notification in database | 1.Set mobile time to 7:00 | 1.Nothing happens, no notification is sent to mobile device | N/A | Pass | 4/14/16 |

Table 75: Test case <System> Notify schedule

### Test cases results and statistics

|  |  |  |
| --- | --- | --- |
| Test case | Runs | Pass rate (%) |
| TO- TO 1 | 50 | 100 |
| TO- TO 2 | 50 | 100 |
| TO- TO 3 | 50 | 100 |
| TO- TO 4 | 50 | 90 |
| TO- TO 5 | 50 | 80 |
| TO- TO 6 | 50 | 80 |
| TO- TO 7 | 50 | 100 |
| TO- TO 8 | 50 | 90 |
| TO- TO 9 | 50 | 80 |
| TO- TO 10 | 50 | 70 |
| TO- TO 11 | 50 | 75 |
| TO- TO 12 | 50 | 60 |
| TO- TO 13 | 50 | 60 |
| TO- TO 14 | 50 | 65 |
| TO- TO 15 | 50 | 100 |
| TO- TO 16 | 50 | 70 |
| TO- TO 17 | 50 | 60 |
| TO- TO 18 | 50 | 65 |
| TO- TO 19 | 50 | 55 |
| TO- TO 20 | 50 | 58 |
| TO- TO 21 | 50 | 40 |
| TO- TO 22 | 50 | 50 |
| TrainO 1 | 50 | 100 |
| TrainO 2 | 50 | 100 |
| TrainO 3 | 50 | 100 |
| TranslateO 1 | 50 | 45 |
| TranslateO 2 | 50 | 65 |
| TranslateO 3 | 50 | 60 |
| TranslateO 4 | 50 | 100 |
| TranslateO 5 | 50 | 100 |
| TranslateO 6 | 50 | 100 |
| TranslateO 7 | 50 | 100 |
| TranslateO 8 | 50 | 100 |
| TranslateO 9 | 50 | 100 |
| BL1 | 50 | 100 |
| BL2 | 50 | 100 |
| BL3 | 50 | 100 |
| BL4 | 50 | 100 |
| BL5 | 50 | 100 |
| TOFF-TOFF 1 | 50 | 100 |
| TOFF-TOFF 2 | 50 | 100 |
| TOFF-TOFF 3 | 50 | 100 |
| TOFF-TOFF 4 | 50 | 85 |
| TOFF-TOFF 5 | 50 | 85 |
| TOFF-TOFF 6 | 50 | 85 |
| TrainOff 1 | 50 | 50 |
| TrainOff 2 | 50 | 50 |
| TrainOff 3 | 50 | 50 |
| TranslateOff 1 | 50 | 50 |
| TranslateOff 2 | 50 | 50 |
| TranslateOff 3 | 50 | 50 |
| TranslateOff 4 | 50 | 50 |
| TranslateOff 5 | 50 | 50 |
| NS 1 | 50 | 100 |
| NS 2 | 50 | 100 |
| NS 3 | 50 | 100 |
| NS 4 | 50 | 100 |
| NS 5 | 50 | 100 |

Table 76: Test case result statistic

# Report No.6 Software user’s manual

## Installation guide

### Setting up environment at server side

Bellows are requirements for hardware and software environment to run CBYH system in 10 years. The specifications are based on the dependencies requirements and performance test result from previous section of this document.

#### Hardware requirements

|  |  |
| --- | --- |
| Hardware | Specification |
| Internet Connection | 8 Mbps |
| Computer Processor | Intel® CORE i7 Quad core 2.4 GHz |
| Computer Memory | 4GB of RAM or more |
| Hard Disk Drive | 50GB or more |

Table 78: Hardware requirements

#### Software requirements

|  |  |
| --- | --- |
| Software | Application name / version |
| Operating System | Ubuntu Server 14.04.2 LTS |
| Java | 1.7.0\_79 |
| Web Server | Apache Tomcat 8.0.15 |
| Database | MySQL 5.6 |

Table 79: Software requirements

### Web application / web service deployment process

#### Check environment

Check Ubuntu version: 14.04.2 LTS

|  |
| --- |
| root@CBYH:~# **lsb\_release -a**  No LSB modules are available.  Distributor ID: Ubuntu  Description: Ubuntu 14.04.2 LTS  Release: 14.04  Codename: trusty |

Check Java version: 1.7.0\_79

|  |
| --- |
| root@CBYH:~# **java -version**  java version "1.7.0\_79"  OpenJDK Runtime Environment (IcedTea 2.5.6) (7u79-2.5.6-0ubuntu1.14.04.1)  OpenJDK 64-Bit Server VM (build 24.79-b02, mixed mode) |

Check MySQL version: 5.6

|  |
| --- |
| root@CBYH:~# **mysql --version**  mysql Ver 14.14 Distrib 5.6.19, for debian-linux-gnu (x86\_64) using EditLine wrapper |

Check Apache Tomcat version: 7.0.52

|  |
| --- |
| root@CBYH:~# **/usr/share/tomcat8/bin/version.sh**  Using CATALINA\_BASE: /usr/share/tomcat8  Using CATALINA\_HOME: /usr/share/tomcat8  Using CATALINA\_TMPDIR: /usr/share/tomcat7/temp  Using JRE\_HOME: /usr/lib/jvm/java-8-openjdk-amd64  Using CLASSPATH: /usr/share/tomcat7/bin/bootstrap.jar:/usr/share/tomcat7/bin/tomcat-juli.jar  Server version: Apache Tomcat/8.0.15 (Ubuntu)  Server built: Apr 14 2016 06:59:46  Server number: 8.0.15.0  OS Name: Linux  OS Version: 3.13.0-57-generic  Architecture: amd64  JVM Version: 1.7.0\_79-b14  JVM Vendor: Oracle Corporation |

#### Import database

Using file **Myo01.sql** located under **Deployment** directory from this document.

|  |
| --- |
| root@CBYH:~# **mysql -u root -p cbyh\_data < Myo01.sql** |

#### Build war artifact

Using Netbean 8.0.2 to build the project into **MYO-1.war**

#### Deploy war artifact

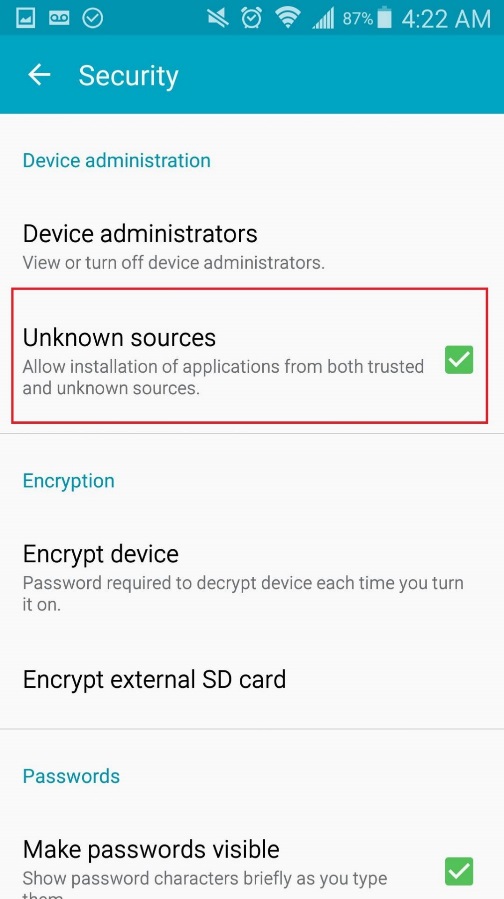
Using file **war** built from previous step.

|  |
| --- |
| root@CBYH:~# **mv MYO-1.war /var/lib/tomcat8/webapps/ROOT.war** |

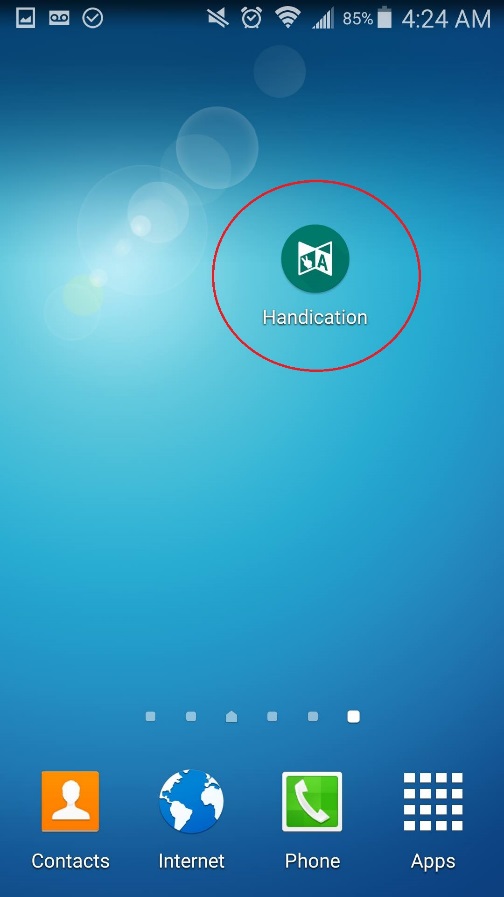
Web application is now available at [**http://server-ip-address**](http://server-ip-address).

### Mobile application deployment process

At mobile device, go to Setting/Security, check checkbox Unknown sources.



Using apk files from Deployment directory under this document. Click to install application.



## User guide

### MYO armband



To use the system, user has to wear a pair of MYO armbands with the following instruction:

* The indicator LED must point to user’s hand.
* The MYO armband should be at 1/3 from elbow to user’s wrist
* User should not wear MYO armbands when the arms are tired for better performance

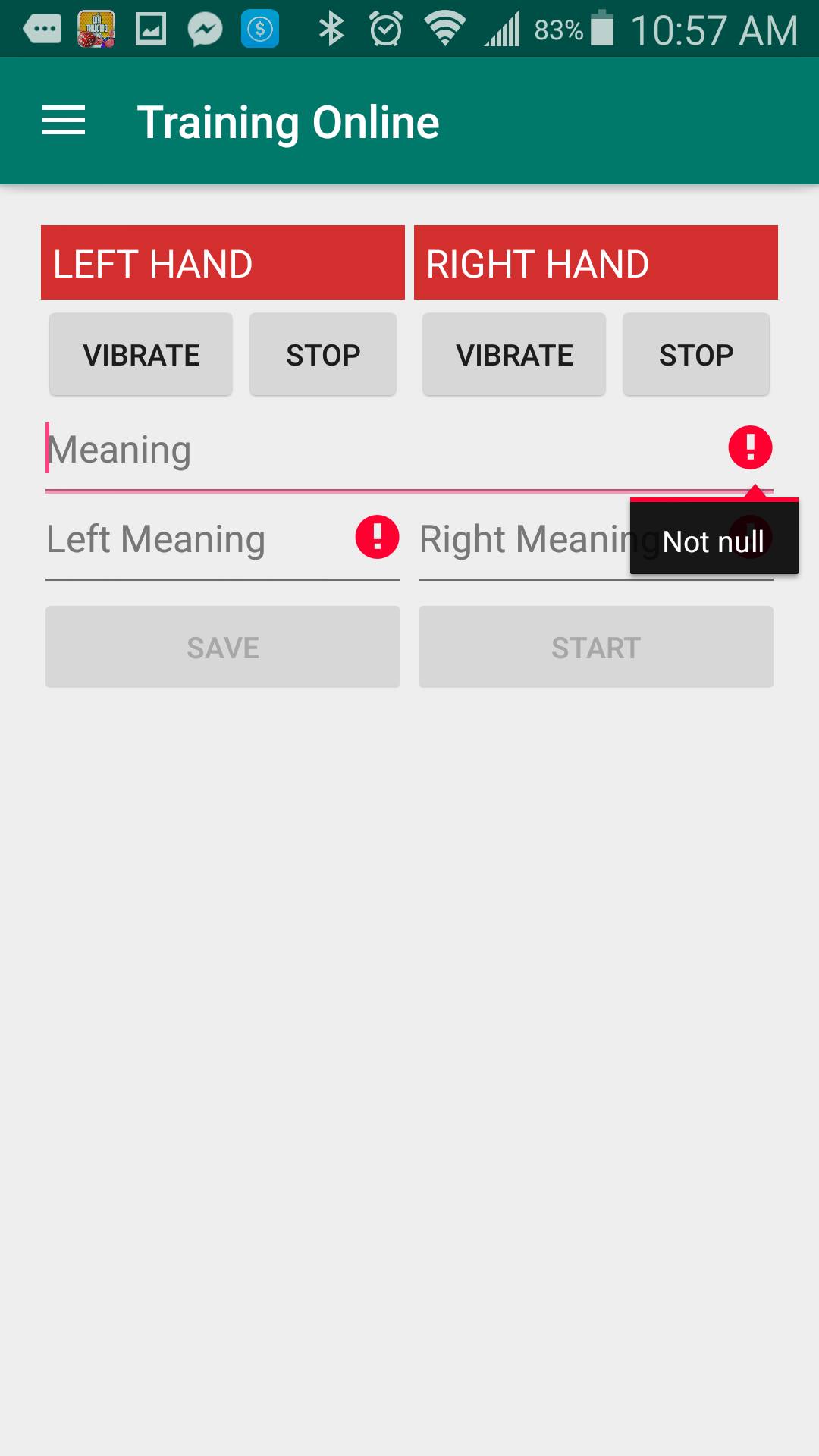
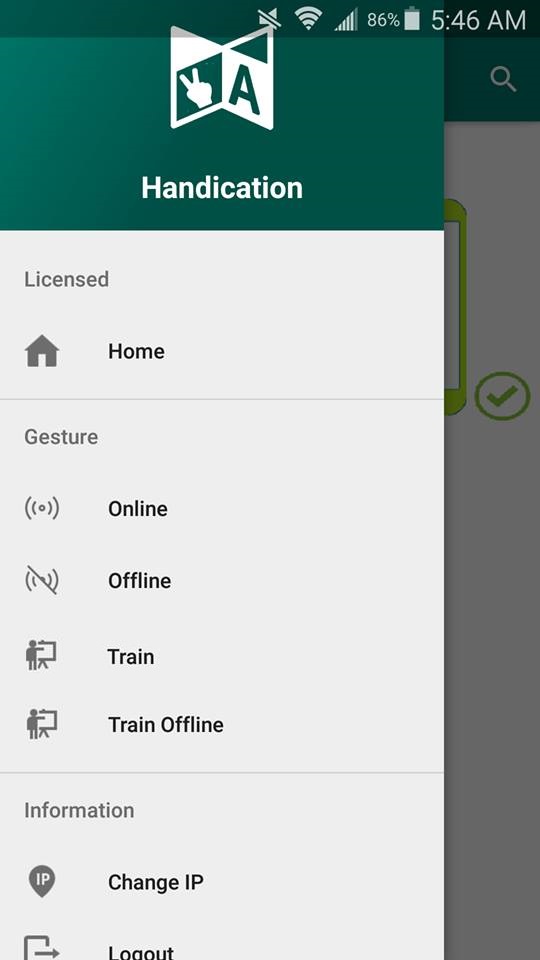
The armband is work best for about first 10 minutes, after that. Users should let their arms rest for 1 minute before the next use for better performance.

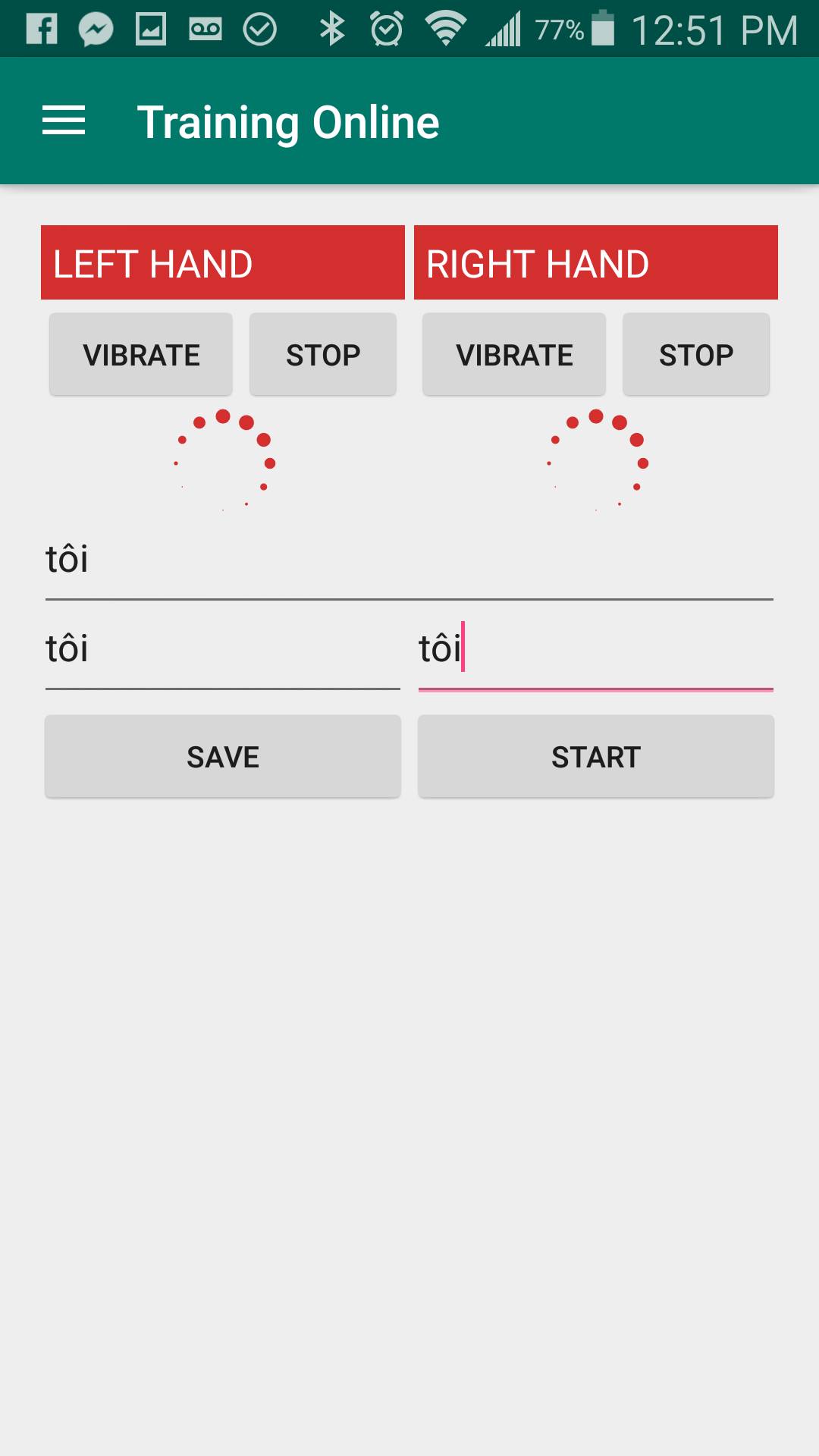
### Mobile application

#### Train online

Precondition:

* The two MYO armbands are paired with the Android mobile device
* The Android mobile device is connected to the internet





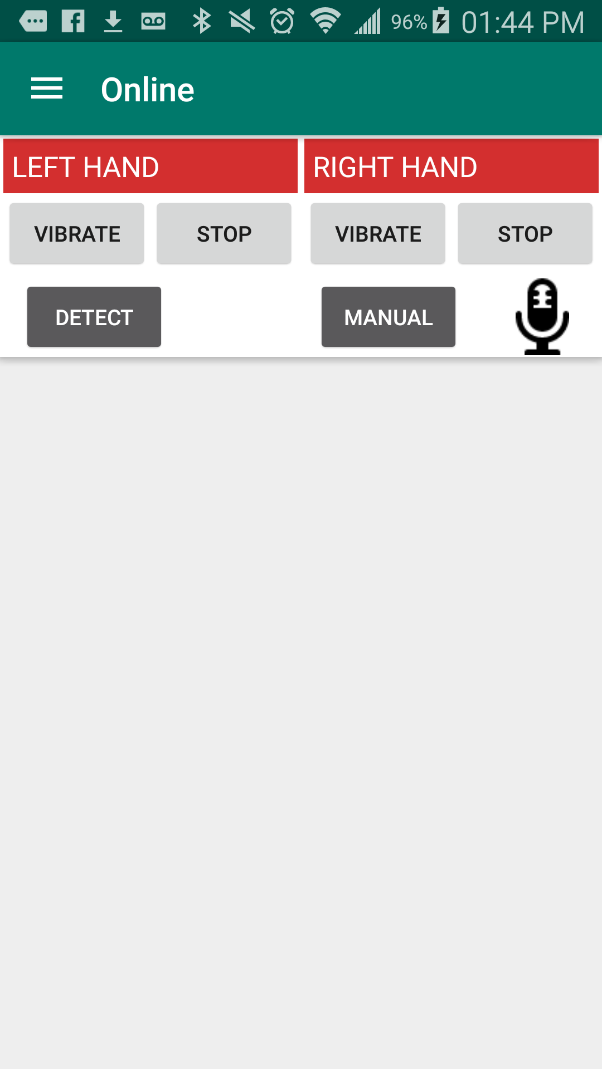
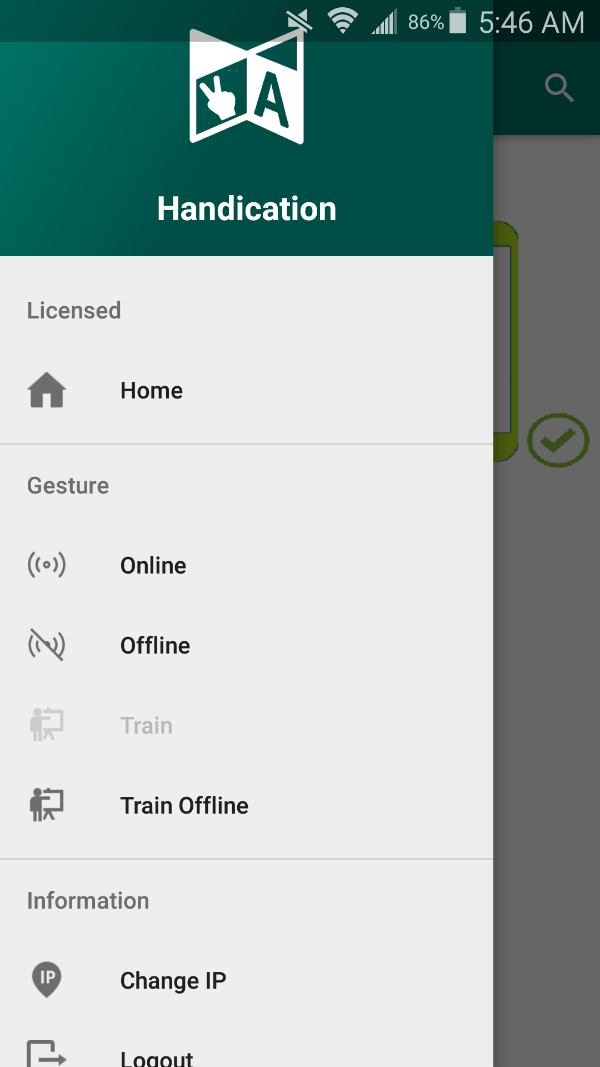
|  |  |
| --- | --- |
| Step | Description |
| 1 | Choose “Train” at navigation panel |
| 2 | Input these fields:  “Meaning”: Tôi (for example)  “Left Meaning”: Tôi (for example)  “Right Meaning”: Tôi (for example) |
| 3 | Press “Start” button |
| 4 | Perform the sign language |
| 5 | Press “Save” Button |

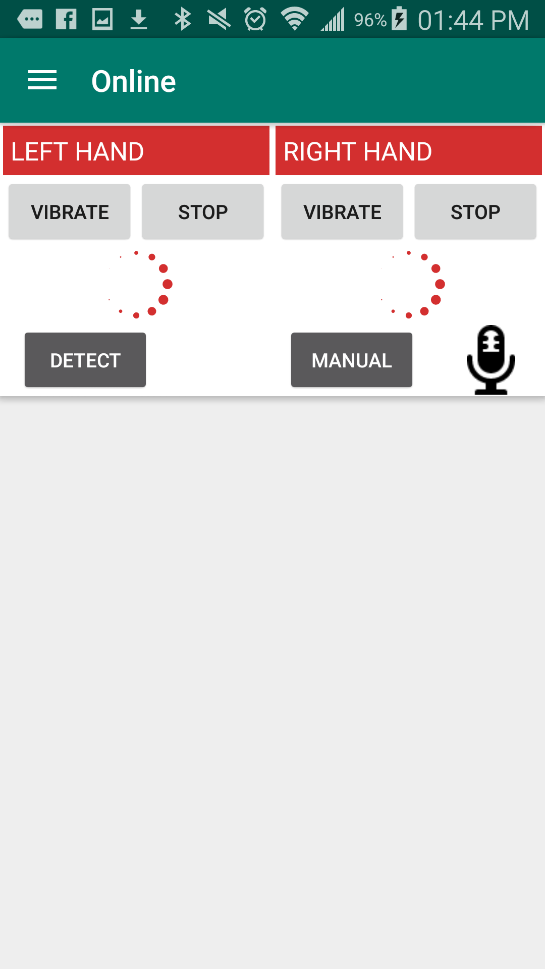
#### Translate online

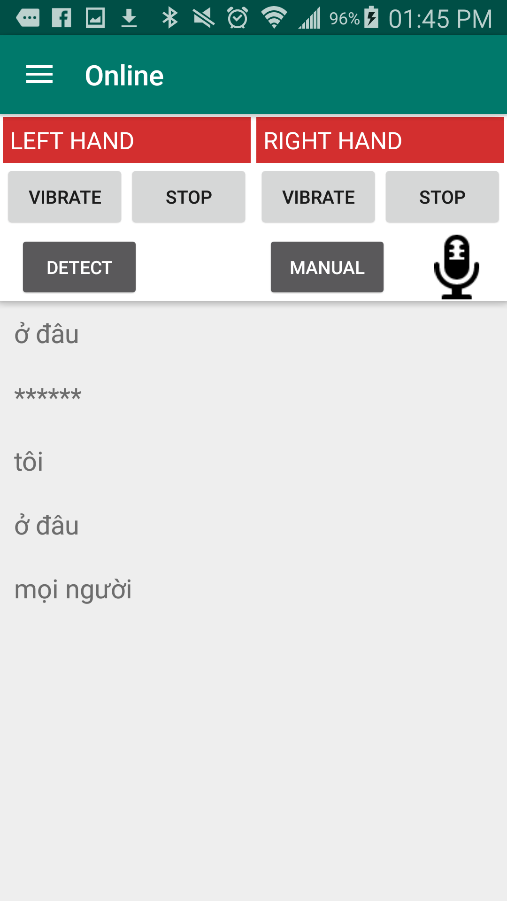
##### Automatic

Precondition:

* The two MYO armbands are paired with the Android mobile device
* The Android mobile device is connected to the internet





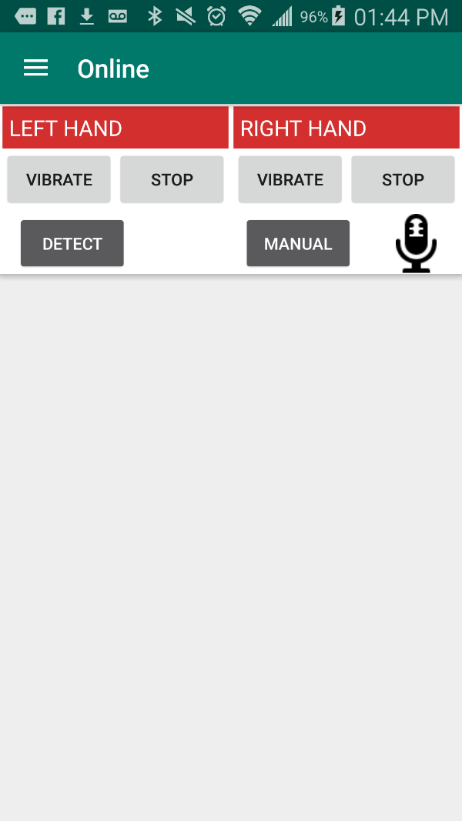
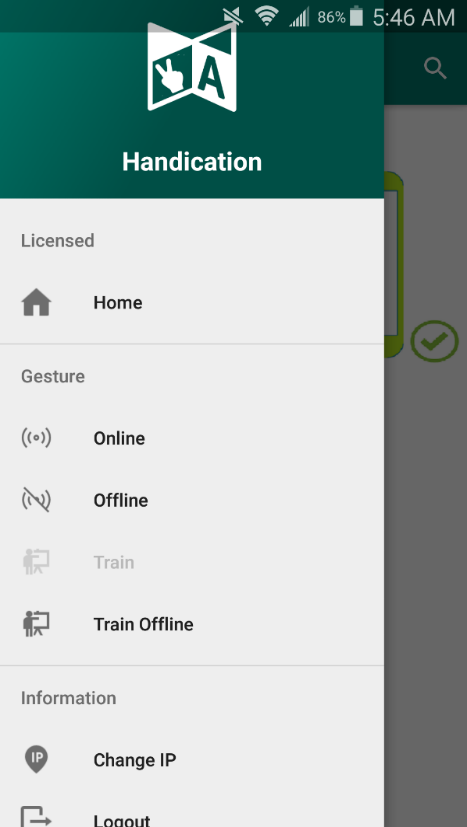


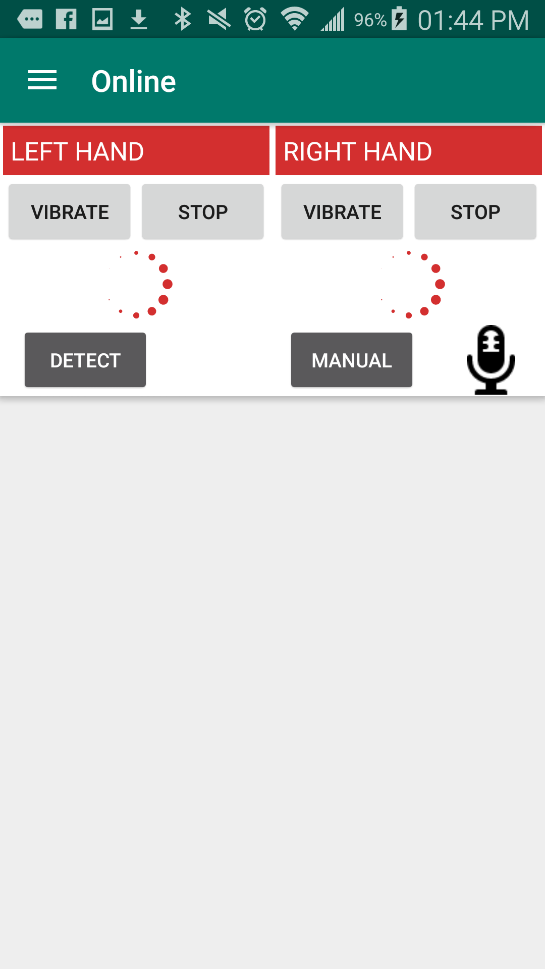
|  |  |
| --- | --- |
| Step | Description |
| 1 | Choose “Online” at navigation panel |
| 2 | Press “Detect” button |
| 3 | Perform the sign language |
| 4 | Perform the rest sign |
| 5 | Check the output information on the output screen |
| 6 | Press  for play sound (optional) |

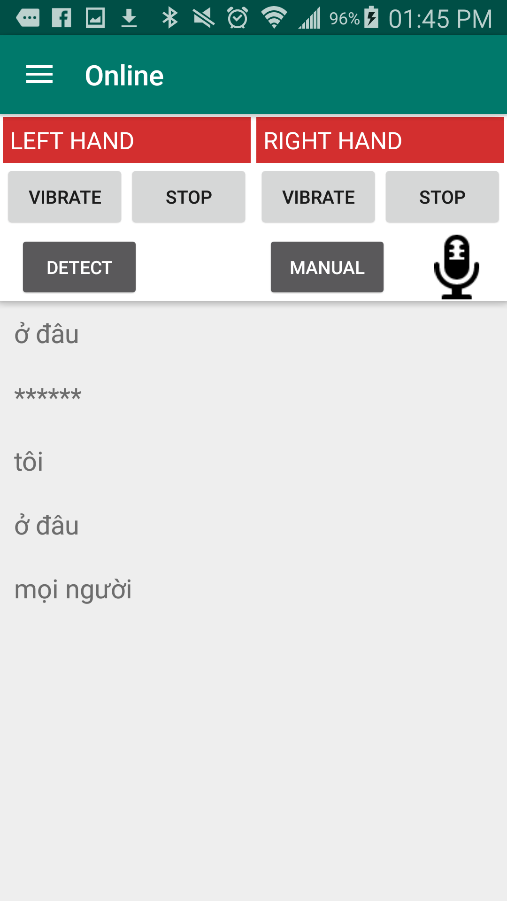
##### Manual

Precondition:

* The two MYO armbands are paired with the Android mobile device





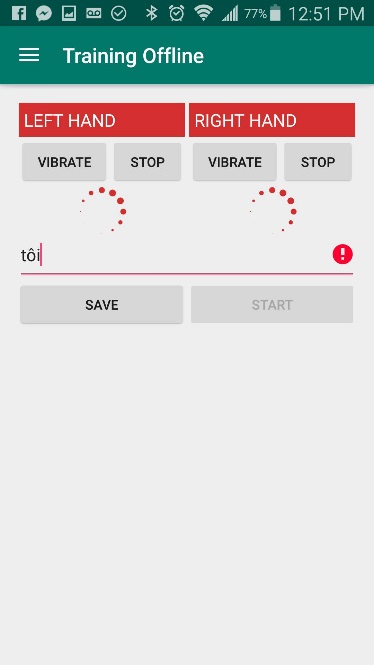


|  |  |
| --- | --- |
| Step | Description |
| 1 | Choose “Online” at navigation panel |
| 2 | Press “Manual” button |
| 3 | Perform the sign language |
| 4 | Press the “Manual” button |
| 5 | Perform the rest sign |
| 6 | Check the output information on the output screen |
| 7 | Press  for play sound (optional) |

#### Train offline

Precondition:

* The two MYO armbands are paired with the Android mobile device





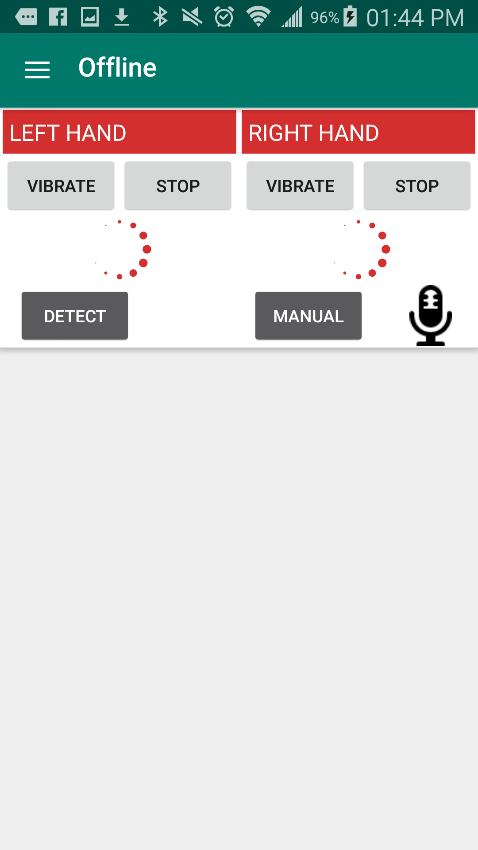
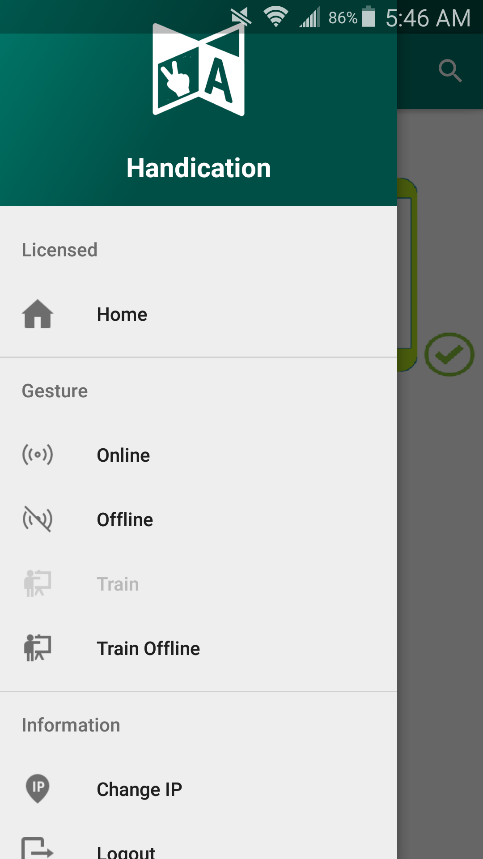
|  |  |
| --- | --- |
| Step | Description |
| 1 | Choose “Train offline” at navigation panel |
| 2 | Input these fields:  “Meaning”: Tôi (for example) |
| 3 | Press “Start” button |
| 4 | Perform the sign language |
| 5 | Press “Save” Button |

#### Translate offline

##### Automatic

Precondition:

* The two MYO armbands are paired with the Android mobile device



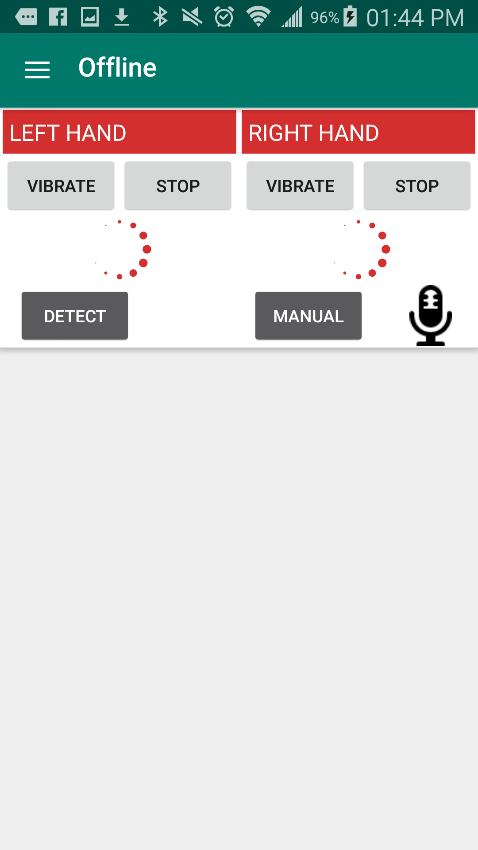
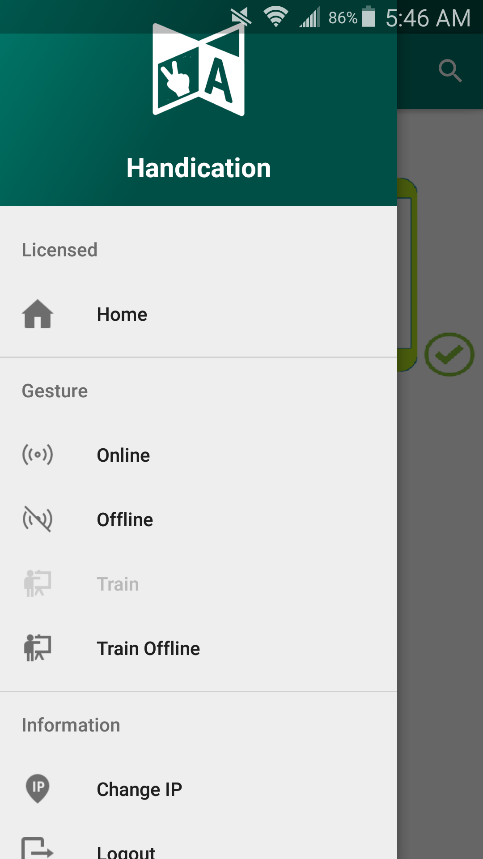


|  |  |
| --- | --- |
| Step | Description |
| 1 | Choose “Offline” at navigation panel |
| 2 | Press “Detect” button |
| 3 | Perform the sign language |
| 4 | Perform the rest sign |
| 5 | Check the output information on the output screen |
| 6 | Press  for play sound (optional) |

##### Manual

Precondition:

* The two MYO armbands are paired with the Android mobile device

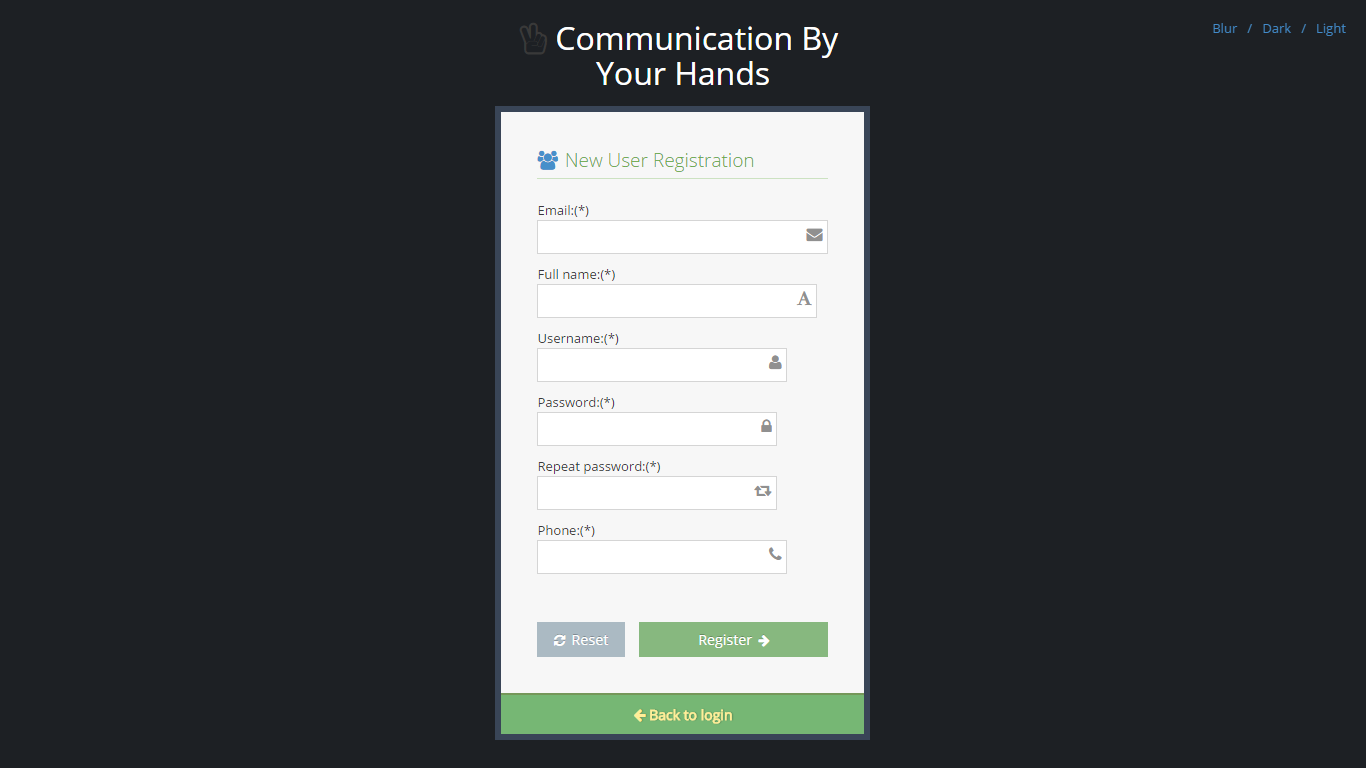




|  |  |
| --- | --- |
| Step | Description |
| 1 | Choose “Offline” at navigation panel |
| 2 | Press “Manual” button |
| 3 | Perform the sign language |
| 4 | Perform the rest sign |
| 5 | Check the output information on the output screen |
| 6 | Press  for play sound (optional) |

### Web application

#### Register

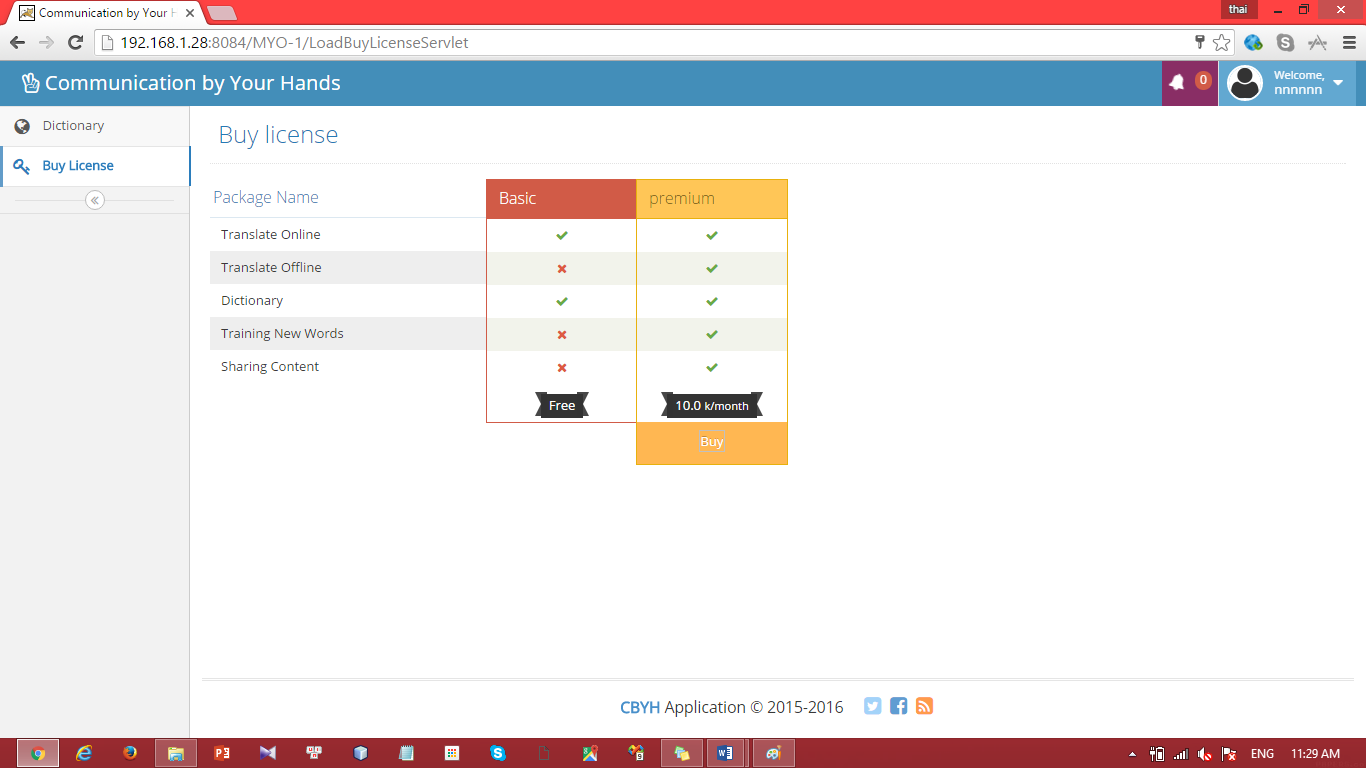


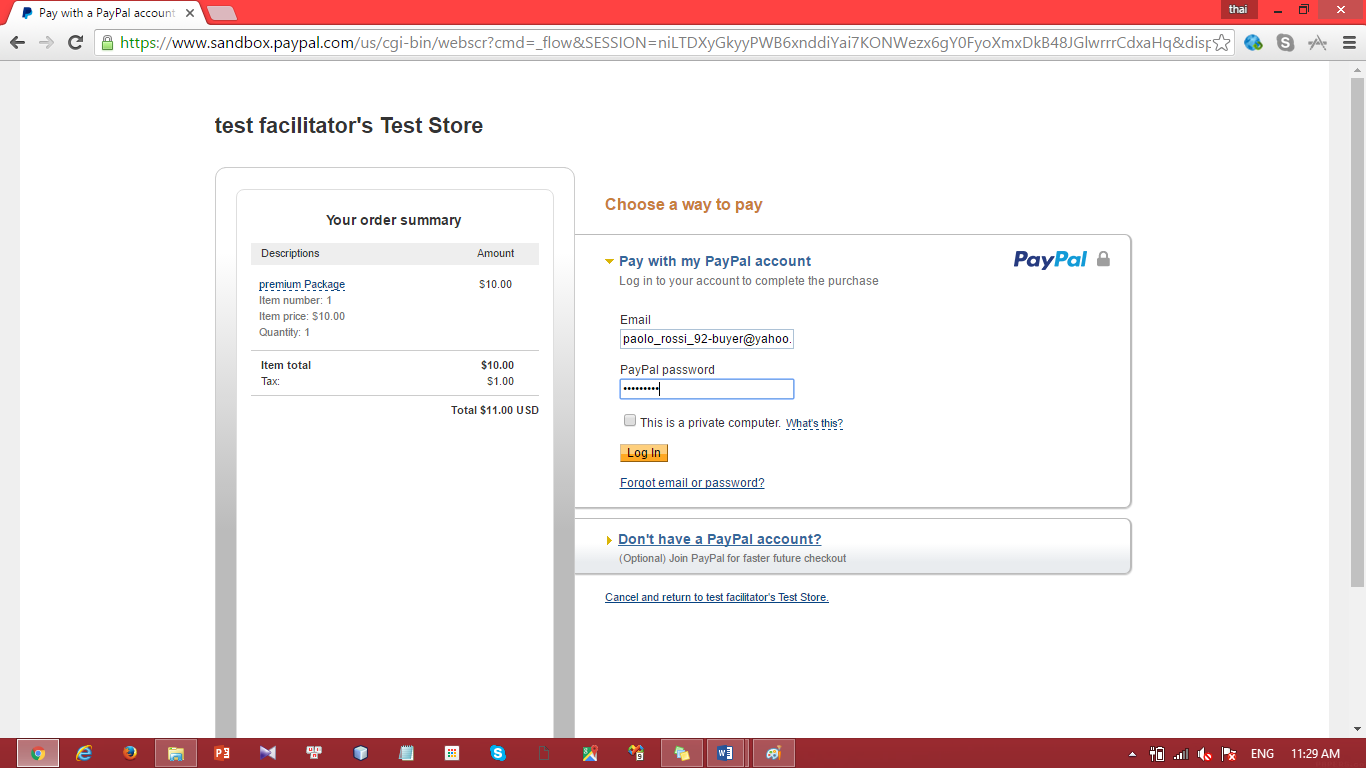
|  |  |
| --- | --- |
| Step | Description |
| 1 | Fill in these fields:  “Email” : [ac@xyz.vn](mailto:ac@xyz.vn) (for example)  “Fullname” : nnnnnn (for example)  “Password” : 123456 (for example)  “Confirm password” : 123456 (for example)  “Phone” : 0908123451 (for example) |
| 2 | Press “Register” button |

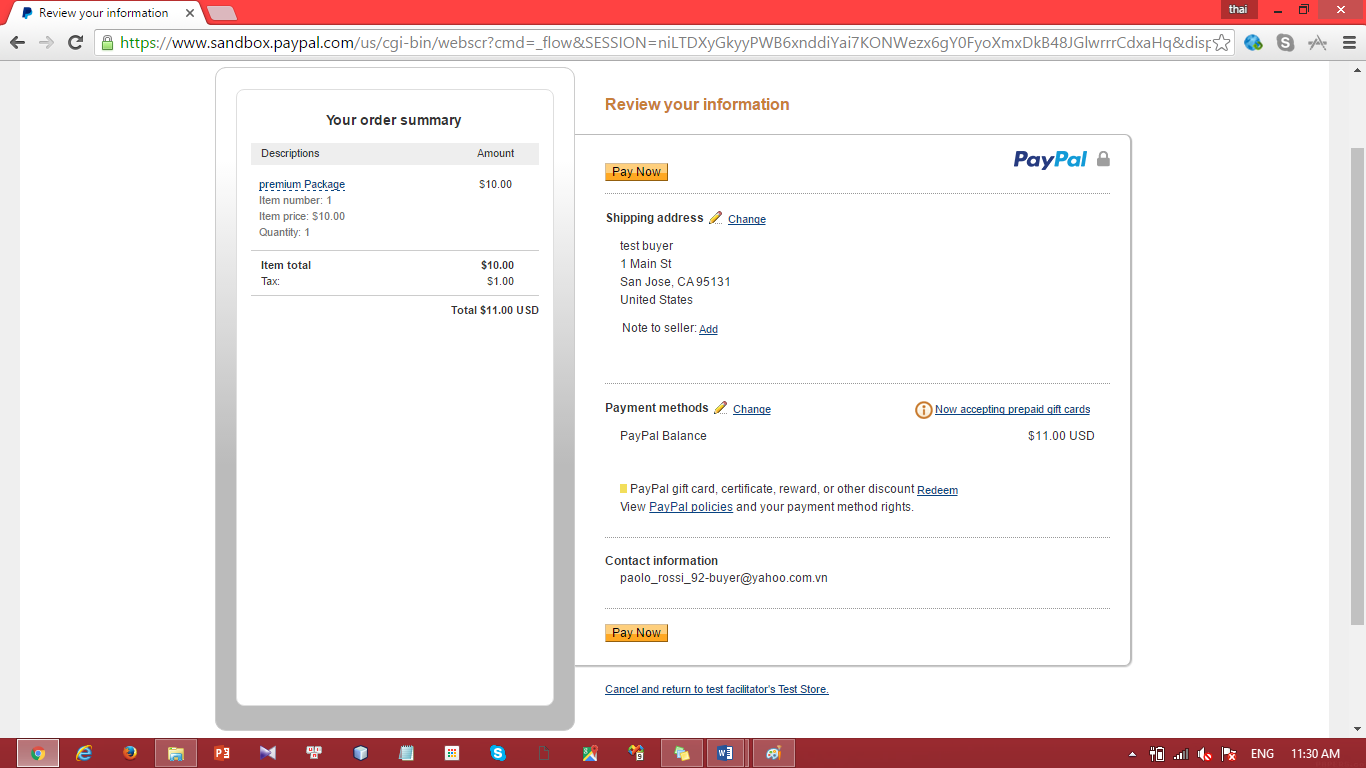
#### Buy license

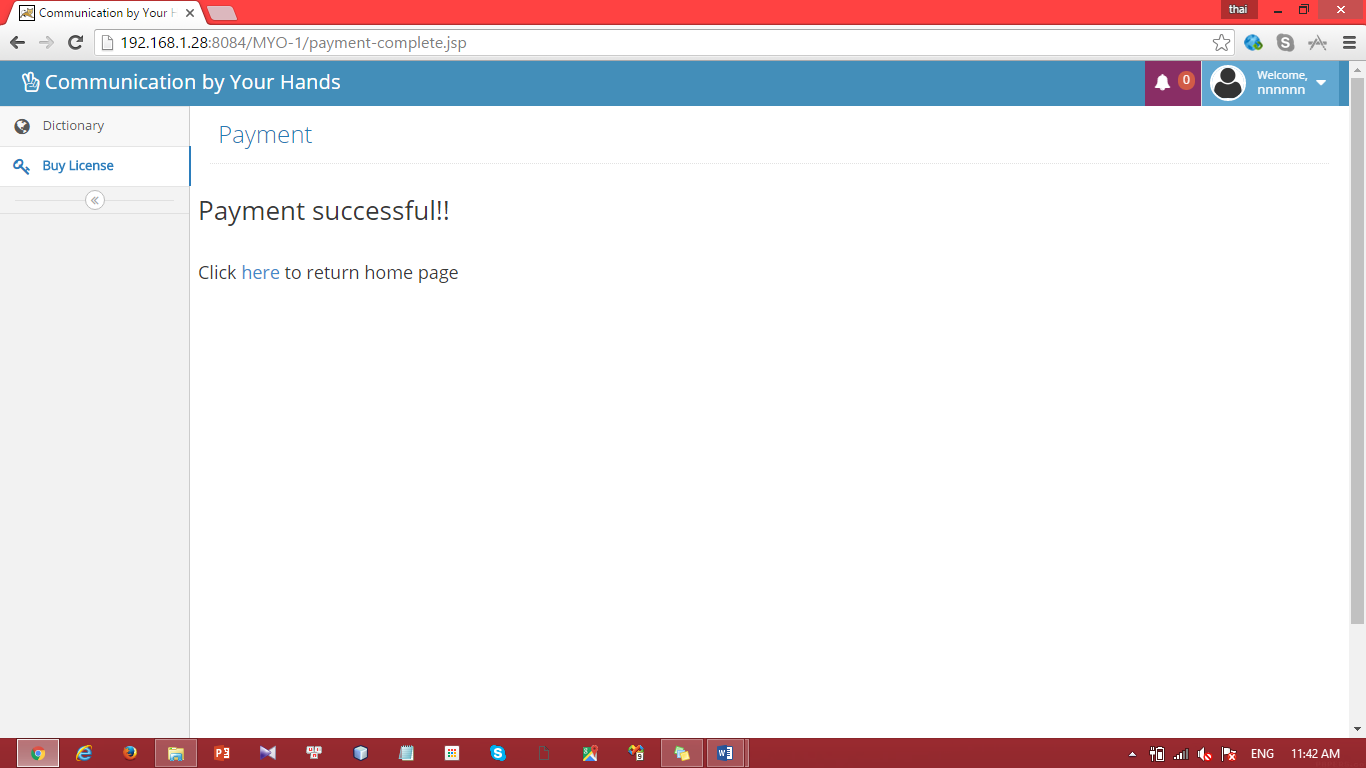
Precondition:

* User already had an Paypal account



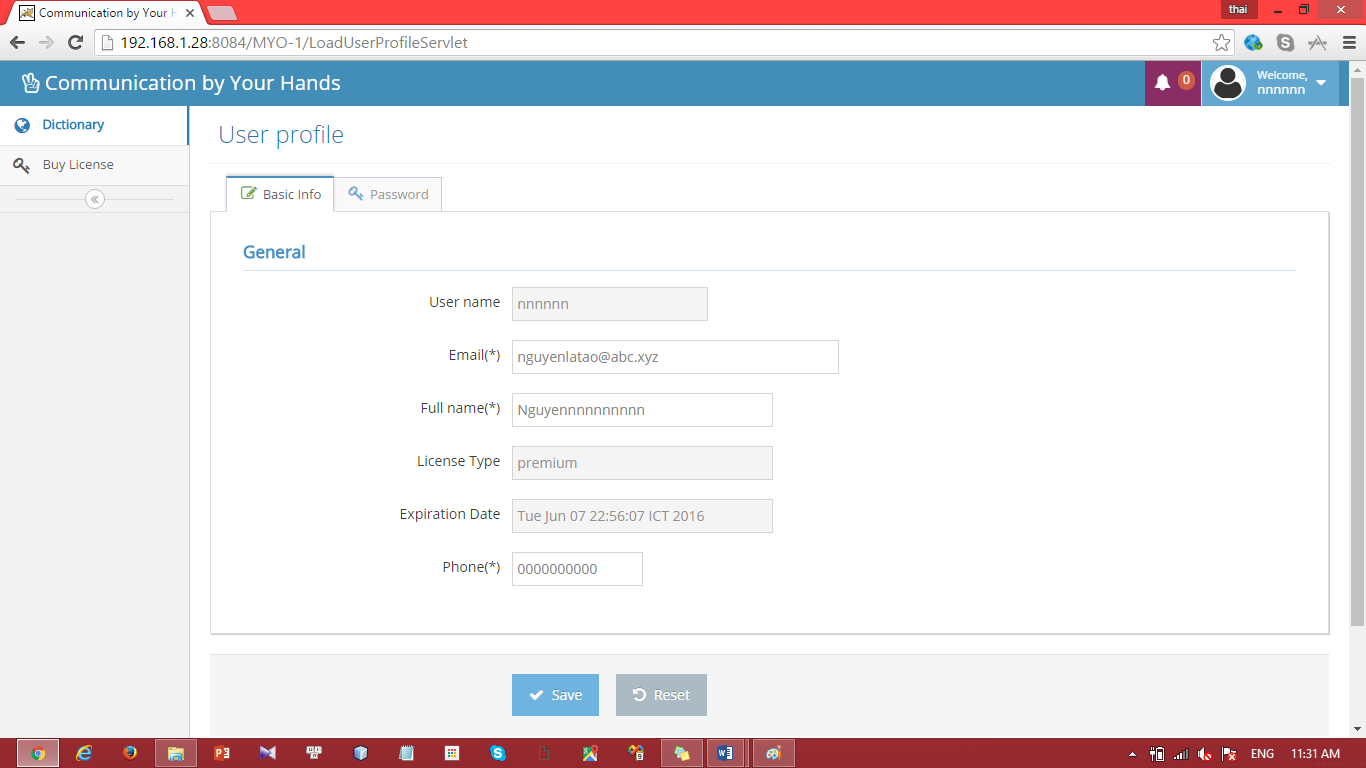


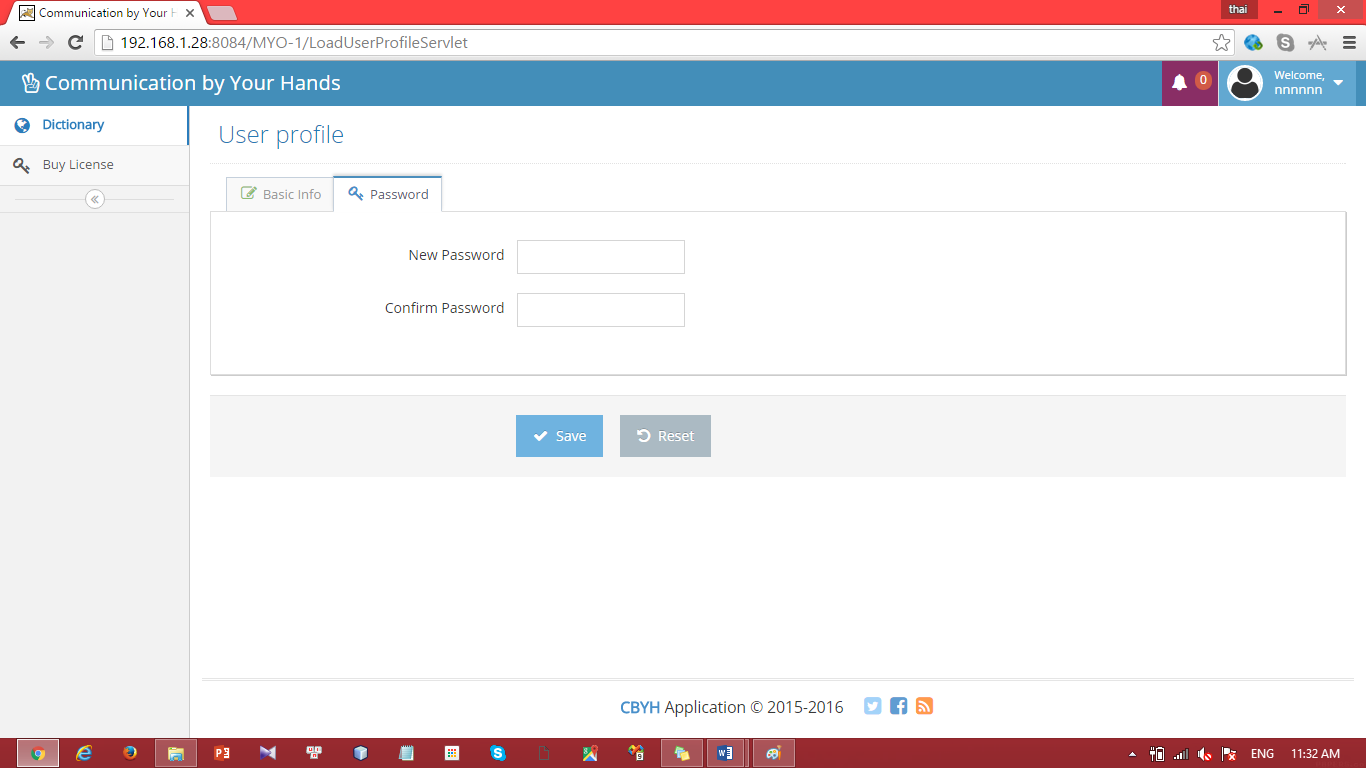




|  |  |
| --- | --- |
| Step | Description |
| 1 | Press “Buy” button |
| 2 | Fill these fields with Paypal account |
| 3 | Press “Buy now” button |
| 4 | Check result information |

#### Update profile





|  |  |
| --- | --- |
| Step | Description |
| 1 | Update these fields:  “Email”: from “[abc@xyz.vn](mailto:abc@xyz.vn)” to “bvc@asd.vn” (for example)  “Fullname”: from “nnnnnn” to “mmmmmm” (for example)  “Phone”: from “0938346538” to “0909090909” (for example) |
| 2 | Press “Password” button |
| 3 | Fill these field:  “New Password” : “654321” (for example)  “Confirm Password” : ”654321” (for example) |
| 4 | Press “Save” button |

# Appendix

1. SOFTWARE ENGINEERING 9th Edition, by Ian Sommerville.
2. SOFTWARE ENGINEERING 8th Edition, by Ian Sommerville.
3. Code Conventions for the Java TM Programming Language, by Sun Microsystems, rev April 20, 1999.
4. Android Developer Guide - Application Fundamentals

http://developer.android.com/guide/components/fundamentals.html

1. http://www.agiledata.org/essays/evolutionaryDevelopment.html