**MINISTRY OF EDUCATION AND TRAINING**

**FPT UNIVERSITY**

Capstone Project Document

**Communication by Your Hands**

|  |  |
| --- | --- |
| **Group 04** | |
| **Group members** |  |
| **Supervisor** | Mr. Kiều Trọng Khánh |
| **Ext. Supervisor** | N/A |
| **Capstone Project code** | FFRS |

-Ho Chi Minh City, **????????**-

*This page is intentionally left blank*

# Table of Contents

[Table of Contents 3](#_Toc498187117)

[List of Tables 5](#_Toc498187118)

[List of Figure 7](#_Toc498187119)

[A. Report No. 1 Introduction 11](#_Toc498187120)

[1. Project Information 11](#_Toc498187121)

[2. Introduction 11](#_Toc498187122)

[3. Current Situation 11](#_Toc498187123)

[4. Problem Definition 11](#_Toc498187124)

[5. Proposed Solution 12](#_Toc498187125)

[5.1 Feature functions 12](#_Toc498187126)

[5.2 Values 12](#_Toc498187127)

[6. Functional Requirements 13](#_Toc498187128)

[7. Role and Responsibility 13](#_Toc498187129)

[B. Report No.2 Software Project Management Plan 14](#_Toc498187130)

[1. Problem Definition 14](#_Toc498187131)

[1.1 Name of this Capstone Project 14](#_Toc498187132)

[1.2 Problem Abstract 14](#_Toc498187133)

[1.3 Project Overview 14](#_Toc498187134)

[2. Project organization 17](#_Toc498187135)

[2.1 Software Process Model 17](#_Toc498187136)

[2.2 Roles and responsibilities 17](#_Toc498187137)

[2.3 Tools and Techniques 20](#_Toc498187138)

[3. Project Management Plan 20](#_Toc498187139)

[3.1 Software development life cycle 20](#_Toc498187140)

[3.2 Phase Detail 22](#_Toc498187141)

[3.3 Task sheet 24](#_Toc498187142)

[3.4 All Meeting Minutes 25](#_Toc498187149)

[4. Coding Convention 25](#_Toc498187150)

[C. Report No.3 Software Requirement Specification 26](#_Toc498187151)

[1. User Requirement Specification 26](#_Toc498187152)

[1.1 Guest Requirement 26](#_Toc498187153)

[1.2 Staff Requirement 26](#_Toc498187154)

[1.3 User Requirement 26](#_Toc498187155)

[1.4 Premium user Requirement 26](#_Toc498187156)

[1.5 Authenticated user Requirement 26](#_Toc498187157)

[1.6 Scheduler Requirement 26](#_Toc498187158)

[2. System Requirement Specification 27](#_Toc498187159)

[2.1 External Interface Requirement 27](#_Toc498187160)

[2.2 System Overview Use Case 28](#_Toc498187161)

[2.3 List of Use Case 28](#_Toc498187162)

[3. Software system attribute 56](#_Toc498187163)

[3.1 Usability 56](#_Toc498187164)

[3.2 Reliability 57](#_Toc498187165)

[3.3 Availability 57](#_Toc498187166)

[3.4 Security 57](#_Toc498187167)

[3.5 Maintainability 57](#_Toc498187168)

[3.6 Portability 57](#_Toc498187169)

[3.7 Performance 57](#_Toc498187170)

[4. Conceptual diagram 58](#_Toc498187171)

[D. Report No.4 Software Design Description 62](#_Toc498187172)

[1. Design Overview 62](#_Toc498187173)

[2. System Architectural Design 63](#_Toc498187174)

[2.1 Web application architecture description 63](#_Toc498187175)

[2.2 Mobile application architecture description 63](#_Toc498187176)

[3. Component Diagram 64](#_Toc498187177)

[4. Detail Description 66](#_Toc498187178)

[4.1 Class Diagram 66](#_Toc498187179)

[4.2 Class Diagram Explanation 68](#_Toc498187180)

[4.3 Interactive Diagram 77](#_Toc498187181)

[5. Interface 87](#_Toc498187182)

[5.1 Component Interface 87](#_Toc498187183)

[5.2 Web Application Design 90](#_Toc498187184)

[5.3 Mobile Application Design 94](#_Toc498187185)

[6. Database design 99](#_Toc498187186)

[6.1 Entity relationship diagram 99](#_Toc498187187)

[6.2 Entity dictionary 101](#_Toc498187188)

[7. Algorithms 101](#_Toc498187189)

[7.1 Standardize EMG data (Android) 101](#_Toc498187190)

[7.2 Matching 102](#_Toc498187191)

[7.3 Detect 106](#_Toc498187192)

[7.4 Train 111](#_Toc498187193)

[7.5 Standardize EMG data before input to Database (Server) 113](#_Toc498187194)

[E. Report No.5 System implement and Test 115](#_Toc498187195)

[1. Introduction 115](#_Toc498187196)

[1.1 Overview 115](#_Toc498187197)

[1.2 Test Approach 115](#_Toc498187198)

[2. Database relationship diagram 115](#_Toc498187199)

[2.1 Physical diagram 115](#_Toc498187200)

[2.2 Data dictionary 117](#_Toc498187201)

[3. Performance measures 125](#_Toc498187202)

[3.1 Mobile Application API load speed 125](#_Toc498187203)

[4. Test plan 127](#_Toc498187204)

[4.1 Features to be tested 127](#_Toc498187205)

[4.2 Features not to be tested 127](#_Toc498187206)

[5. System testing test case 128](#_Toc498187207)

[5.1 Communication diagram 128](#_Toc498187208)

[5.2 Test cases 130](#_Toc498187209)

[5.3 Test cases results and statistics 153](#_Toc498187210)

[F. Report No.6 Software user’s manual 157](#_Toc498187211)

[1. Installation guide 157](#_Toc498187212)

[1.1 Setting up environment at server side 157](#_Toc498187213)

[1.2 Web application / web service deployment process 158](#_Toc498187214)

[1.3 Mobile application deployment process 160](#_Toc498187215)

[2. User guide 162](#_Toc498187216)

[2.1 MYO armband 162](#_Toc498187217)

[2.2 Mobile application 162](#_Toc498187218)

[2.3 Web application 173](#_Toc498187219)

[G. Appendix 178](#_Toc498187220)

# List of Tables

[Table 1: Roles and Responsibilities 14](#_Toc499864252)

[Table 2: Roles and Responsibilities 17](#_Toc499864253)

[Table 3: Roles and Responsibilities 17](#_Toc499864254)

[Table 4: Roles and Responsibilities Details 19](#_Toc499864255)

[Table 5: Tools and Techniques 20](#_Toc499864256)

[Table 9: Software Development Life Cycle Detail 22](#_Toc499864257)

[Table 10: Phase 1: Specification 23](#_Toc499864258)

[Table 11: Phase 2: Implementation 23](#_Toc499864259)

[Table 12: Phase 3: Validation 24](#_Toc499864260)

[Table 13: Register specification 32](#_Toc499864261)

[Table 14: Login specification 34](#_Toc499864262)

[Table 15: Search specification 35](#_Toc499864263)

[Table 16: Translate (Online) specification 38](#_Toc499864264)

[Table 17: Buy license specification 40](#_Toc499864265)

[Table 18: Edit profile specification 43](#_Toc499864266)

[Table 19: Translate (Offline) specification 46](#_Toc499864267)

[Table 20: Train custom sign specification 48](#_Toc499864268)

[Table 21: Train sign language specification 51](#_Toc499864269)

[Table 22: Logout specification 52](#_Toc499864270)

[Table 23: Create notification specification 54](#_Toc499864271)

[Table 24: Send notification specification 56](#_Toc499864272)

[Table 25: Conceptual diagram data dictionary 61](#_Toc499864273)

[Table 26: Component Dictionary 66](#_Toc499864274)

[Table 27: Class dictionary 68](#_Toc499864275)

[Table 28: Account Attributes 69](#_Toc499864276)

[Table 29: Profile Attributes 69](#_Toc499864277)

[Table 30: Role Attributes 69](#_Toc499864278)

[Table 31: Promotion Attributes 70](#_Toc499864279)

[Table 32: FavoriteField Attributes 70](#_Toc499864280)

[Table 33: Blacklist Attributes 71](#_Toc499864281)

[Table 34: Voucher Attributes 71](#_Toc499864282)

[Table 35: VoucherRecord Attributes 71](#_Toc499864283)

[Table 36: Field Attributes 72](#_Toc499864284)

[Table 37: FieldType Attributes 72](#_Toc499864285)

[Table 38: TimeEnable Attributes 73](#_Toc499864286)

[Table 39: TimeSlot Attributes 73](#_Toc499864287)

[Table 40: Matching request Attributes 74](#_Toc499864288)

[Table 41: TourMatch Attributes 75](#_Toc499864289)

[Table 42: FriendlyMatch Attributes 75](#_Toc499864290)

[Table 43: Bill Attributes 75](#_Toc499864291)

[Table 43: RattingOpponent Attributes 76](#_Toc499864292)

[Table 43: ReportOpponent Attributes 76](#_Toc499864293)

[Table 44: Web Services interface 88](#_Toc499864294)

[Table 45: Exception description 89](#_Toc499864295)

[Table 46: <Guest> Register fields 90](#_Toc499864296)

[Table 47: <Guest> Register buttons/ hyperlinks 91](#_Toc499864297)

[Table 48: <User> Buy license fields 91](#_Toc499864298)

[Table 49: <User> Buy license buttons/ hyperlinks 92](#_Toc499864299)

[Table 50: <User> Search instruction fields 92](#_Toc499864300)

[Table 51: <User> Search instruction buttons/ hyperlinks 92](#_Toc499864301)

[Table 52: <User> Edit user fields 93](#_Toc499864302)

[Table 53: <User> Edit user buttons/hyperlinks 94](#_Toc499864303)

[Table 54: <Staff> Train online fields 95](#_Toc499864304)

[Table 55: <Staff> Train online buttons/hyperlinks 95](#_Toc499864305)

[Table 56: <User> Login fields 96](#_Toc499864306)

[Table 57: <User> Login buttons/hyperlinls 96](#_Toc499864307)

[Table 58: <User> Translate fields 97](#_Toc499864308)

[Table 59: <User> Translate buttons/hyperlinks 98](#_Toc499864309)

[Table 60: <Premium user> Train offline fields 99](#_Toc499864310)

[Table 61: <Premium user> Train offline Buttons/Hyperlinks 99](#_Toc499864311)

[Table 62: Entity dictionary 101](#_Toc499864312)

[Table 63: Data table dictionary 118](#_Toc499864313)

[Table 64: Data table dictionary 120](#_Toc499864314)

[Table 65: Mobile API load speed client specification 121](#_Toc499864315)

[Table 66: Mobile API load speed test cases 121](#_Toc499864316)

[Table 67: Mobile API load speed test result 122](#_Toc499864317)

[Table 68: Test case <Staff>-<User> Train Online – Translate Online 132](#_Toc499864318)

[Table 69: Test case <Staff> Train Online 134](#_Toc499864319)

[Table 70: Test case <User> Translate Online 137](#_Toc499864320)

[Table 71: Test case <User> Buy license 139](#_Toc499864321)

[Table 72: Test case <Premium User> Train Offline – Translate Offline 142](#_Toc499864322)

[Table 73: Test case <Premium User> Train Offline 143](#_Toc499864323)

[Table 74: Test case <Premium User> Translate Offline 145](#_Toc499864324)

[Table 75: Test case <System> Notify schedule 147](#_Toc499864325)

[Table 76: Test case result statistic 152](#_Toc499864326)

[Table 78: Hardware requirements 152](#_Toc499864327)

[Table 79: Software requirements 153](#_Toc499864328)

# 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: Football Field Reservation System

- Project Code: FFRS

- Product Type: Mobile application, Web application

- Start Date: 5/9/2017

- End Date: /12/2017

## Introduction

Football is one of the most popular sport in the world. It does not attract only professionals but also many new players and fans regardless of gender, age as well as region. As a result, thousands of soccer fields are built to provide playgrounds for everyone. Currently, people who want to reserve a football field will have to come in and set up a rental schedule with the field owner or contact via mobile phone. If the field is not available, they will normally give up because they only know a few fields that are familiar to them. For field owners, they cannot manage field effectively and not have environment to promote to users. We build a system to help users to find other fields around that area.

In this document, we introduce our system as a new solution for both field owners and users to communicate faster, more convenient and easier. In particular, the main purpose of the system is to help field owners manage their fields more effective; help users reserve field online and find opponents have same level based on rating points. Rating point is calculated by rating and comment of users after match finished.

## Current Situation

When users want to reserve a football field:

Case 1: If user already had opponent team (there are 2 teams)

Reservation

Case 2: User needs to find team (there is 1 team)

1. Ask field owner to find out opponent team

2. Field owner checks if it is OK

3. Reservation

## Problem Definition

The disadvantages of the current situation:

Field owner:

Record reservation request on paper makes it is easier to lead to conflict and missing.

Usually not have customers at idle hours.

User:

It takes time to find out and reserve a suitable field at peak times.

If there is not enough people to divide by 2 team before going, users are hard to find opponents.

Finding and matching opponents are not suitable because the number of teams limit to one field owners.

User have to go to the field to reserve.

## Proposed Solution

The solution we proposed is to develop a reservation system for users and field owners which allows users to search and reserve a field. It can also match users with other team having same level to play with. The field owners can manage and promote their fields efficiently by using this system.

Our system includes a web application for field owners and a mobile application for users with following functions:

### **Feature functions**

Web application:

* Provides the ability to manage field status for field owners.
* Manage field timeline.
* Manage profits.
* Manage promotions: provide users discount vouchers.

Mobile application:

* Search fields: users can look for fields by mobile GPS or field name, time…
* View timeline of fields and reserve, checkout online.

System component:

* Matching users and reserve field automatically.
* Suggest fields: the system will suggest suitable fields for users nearby location.
* Calculate user rating score and field score after each match.
* Manage field reservation schedule.

### Values

* Benefits:
  + Users do not have to go to field to reserve because the reservation transaction is proceeded online.
  + Users can find opponents with the matching system.
  + Field owner can manage timeline of field more effective, all things is record in system.
  + With function set discount price, free services for each time frame, field owner can look for users in idle times.
  + With rating function, users will be received a lot of feedbacks about field to make right choice.
* Drawbacks:
  + The application depends heavily on user’s behaviors. If the user does not support assess or assess incorrectly, the calculation ranking of system will be incorrect.

## Functional Requirements

The functional requirements are listed as below:

* Users component:
  + Reserve fields.
  + Manage rewards for using application.
  + View timeline of field to reserve.
  + Manage reservation requests.
  + Rate field and competitors.
  + View notifications about field reservation request, promotion of field...
* Field owners component:
  + Manage their fields: field owners can set field status, timeline and price.
  + Rate teams who have played in their fields after match.
  + Manage promotion.
  + Manage profit.
  + View notifications about reserve field status.
* System component:
  + Suggest fields for user.
  + Match teams have same level to play together.
  + Process payment between players and system; between system and field owners.
  + Calculate team and field rating score, give bonus point to user.
* Admin component:
  + Manage user’s account and profit.

## Role and Responsibility

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No** | **Full Name** | **Role** | **Position** | **Contact** |
| 1 | Kiều Trọng Khánh | Project Manager | Supervisor | khanhkt@fpt.edu.vn |
| 2 | Mai Minh Quý | Developer | Leader | quymmse61610@fpt.edu.vn |
| 3 | Trương Hữu Thành  Trương Hữu Thành | Developer | Member | thanhthse61493@fpt.edu.vn |
| 4 | Phan Minh Huấn  Phan Minh Huấn | Developer | Member | huanpmse61860@fpt.edu.vn |
| 5 | Phạm Trung Hiếu | Developer | Member | hieuptse60874@fpt.edu.vn |
| Table 1: Roles and Responsibilities | | | | |

# Report No.2 Software Project Management Plan

## Problem Definition

### Name of this Capstone Project

* Official name: Football Field Reservation System
* Vietnamese name: Hệ thống đặt chỗ đá bóng
* Abbreviation: FFRS

### Problem Abstract

* The system provides a mobile application for user to support reserve field online. User have 2 options to reserve field:
  + - If user has full team, user can search field, view timeline of field. User chooses field have free time as expected and reserve field.
    - If user wants match opponents, system will suggest opponent have same level with user. Rating point of user is rated by opponent, field owner after match finish. Based on result, system will calculate to identify ability of user.
* Field owner will have web application to manage field status. With system, field owner can provide promotion to attract players. When having field reservation request, system will arrange and choose suitable field automatically.
* In addition, we build web application for admin to manage profit. Besides, the admin will have right to lock account for bad users.

### Project Overview

#### Current Situation

Below are the problems encountered in this project:

* New technique: Some team members are new to the techniques used in the project. The team need an amount of time to get familiar with those techniques.
* Design interface: The system serves users of varying degrees. So, we need an amount of time to analysis and evaluate how to build the easy to use user interface.
* Payment: The system processes payment online so that user must have electronic wallet.

#### The Proposed System

The system will have three sub-systems:

* An API application to serve API for mobile application and web application. API application is a center to process all business logic.
* A mobile application for users to perform find opponents, reserve field, feedback opponent and field, view promotion from field owner, manage and exchange discount voucher from bonus points, view notify from system.
* A web application for field owners and administrators. Field owner perform manage fields and field timeline, promote sale off price or free services, manage profits, view notify about field reservation request. Admin manage profits and lock account of bad users.

We build an algorithm to arrange field reservation request between the fields with other effectively because many field owner own multiple fields.

We use Paypal payments portals to perform transactions in the system, using Google API to get directions for user and find nearby location of field on the map.

##### Mobile Application:

* For user:

+ Manage profile

+ Find field

+ View timeline of field to reserve

+ Find opponent

+ Reserve field

+ Manage field reservation request

+ Payment online

+ View rewards, exchange voucher from bonus points

+ View notifications

+ View reservation history

* For guest

+ Become football team

##### Web Application

* For admin:

+ Manage profit

+ Receive report and lock user’s account

* For field owner:

+ Manage profile

+ Create/View/Delete fields

+ View free field at a specific time

+ View timeline of field

+ View notify about field reservation request

+ Manage time enable

+ Disable time slot of field

+ Manage profits

+ Manage promotion

* For guest:

+ Become a field owner

##### API Application

The server system takes responsibility to respond all the requests and also manages and processes data.

* Provide APIs for Mobile Application, Web Application
* Suggest opponent, field
* Perform scheduled tasks

#### Boundaries of the System

The system does:

* Allow user to find fields.
* Allow user to find opponents.
* Allow user to reserve fields.
* Allow user to create a matching opponents request.
* Allow user to cancel field reservation request.
* Suggest opponents with similar level.
* Suggest field have free time as expected.
* Allow field owner manage their field.
* Notify to field owner when field is reserved.
* Notify to user when request is accepted.
* Allow admin to manage profit.
* Allow admin to view report and lock user account.

#### Future Plans

Current system is concentrated on core business flow. Therefore, some supporting features are restricted for the development team. These features may be expanded in the future:

* Organize tournament to attract more users and based on the results to assess skill of users more exactly.

#### Development Environment

##### Hardware requirement

* **For server:**

|  |  |  |
| --- | --- | --- |
|  | **Minimum Requirements** | **Recommended** |
| Internet Connection | Cable (4 Mbps) | Cable (8 Mbps) |
| Operating System | Ubuntu 12.04 LTS | Ubuntu 16.04 LTS |
| Computer Processor | Intel® Core i3 1.4GHz | Intel® Core i5 2.50GHz |
| Computer Memory | 2 GB RAM | 4 GB RAM or higher |
| Table 2: Roles and Responsibilities | | |

* **For mobile:**

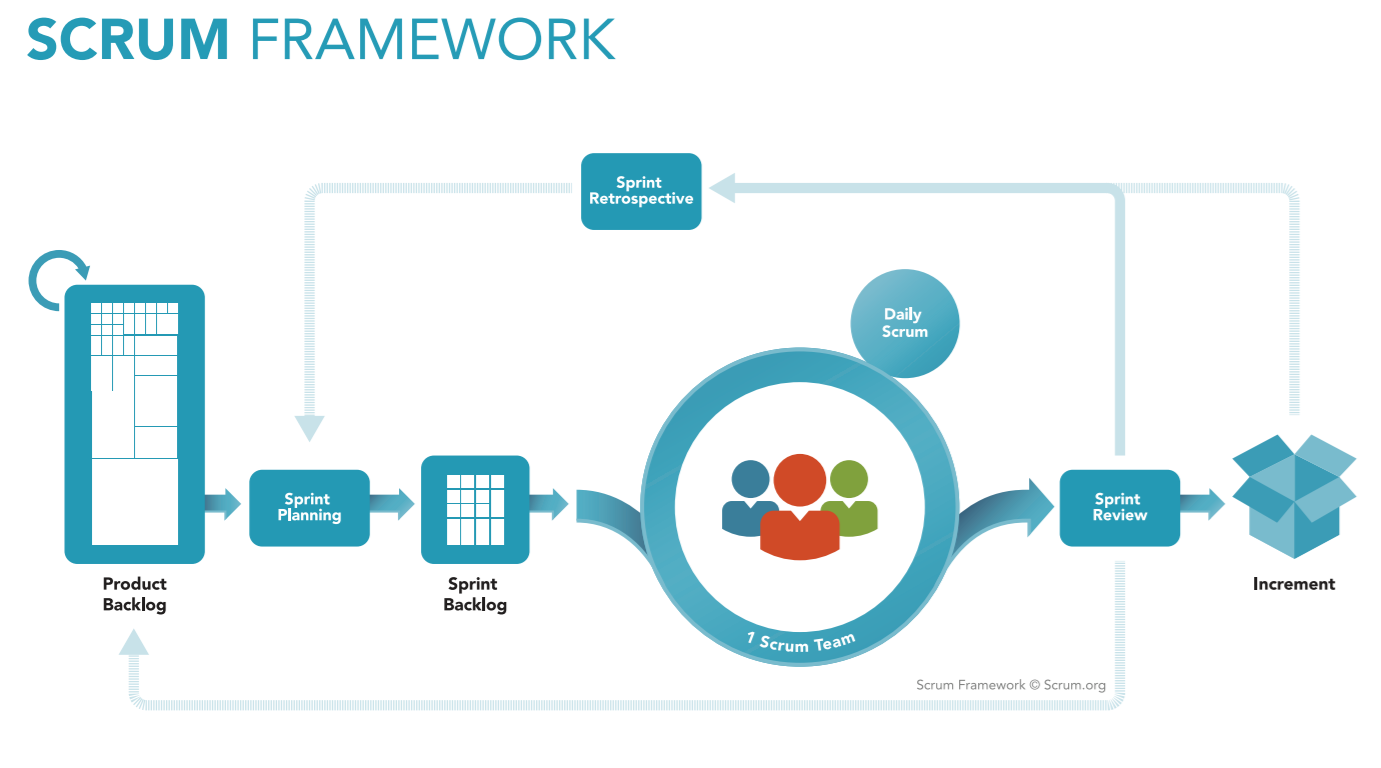
|  |  |  |
| --- | --- | --- |
|  | **Minimum Requirements** | **Recommended** |
| Internet Connection | Wi-Fi or 3G | Wi-fi or 3G, 4G |
| Operating System | Android 6 | Android 7 |
| Mobile Processor | Qualcomm Snapdragon 400 4 Cores 32-bit | [Qualcomm Snapdragon 430 8 Cores 64 bit](https://www.thegioididong.com/hoi-dap/qualcomm-snapdragon-430-844784) |
| Mobile Memory | 1 GB RAM | 2 GB RAM or higher |
| Table 3: Roles and Responsibilities | | |

## Project organization

### Software Process Model

This project is developed using Scrum model – part of an agile framework for Software development project. Our team choose Scrum model because of the following reasons:

* In the team, there are 2 members not playing football. So, all members must work together in order to discuss about business logic and avoid misunderstanding.
* In the project there are many new technologies that need to be learned. With the Scrum model, the team can learn and develop in parallel to meet deadline.
* There is no leader, no hierarchy in team, so team members work cheerfully, stimulating the initiative and creativity of each member.
* The project implements a new idea, so maybe product owner change requirement or extend scope. The team will adapt to change better.



*Figure 1: Scrum Process*

(<https://www.scrum.org/resources/what-is-scrum>)

### Roles and responsibilities

|  |  |  |  |
| --- | --- | --- | --- |
| **No** | **Role in Scrum** | **Name** | **Responsibilities** |
| 1 | Product owner | Kiều Trọng Khánh | * Specify scope and user requirement. * Supervise the development progress. * Provide professional techniques and business analysis support. |
| 2 | Scrum master | Mai Minh Quý | * Create Sprint Backlog and Product Backlog. * Make sure the Scrum teams understand and follow the process. * Always be present to answer questions and give advice when product owner or scrum member needs. * Help the team master scrum artifacts such as: Sprint Backlog, Product Backlog, ... |
| 3 | Scrum team member | Mai Minh Quý  Trương Hữu Thành  Phan Minh Huấn  Phạm Trung Hiếu | * Designing database * Clarifying requirements * Prepare documents * GUI design * Coding * Testing |
|  | | | |

Table 4: Roles and Responsibilities Details

### Tools and Techniques

|  |  |
| --- | --- |
|  | Tool/Technique |
| Mobile Application | Android SDK, Firebase |
| Web Portal | CSS, HTML5, React JS, Java Script, Firebase |
| Back-end | SpringBoot framework, Java, JPA |
| IDE | IntelliJ 2017.2.4, Android Studio 2.3.3 |
| Database | MySQL Server 5.7.19, MySQL Workbench 6.3.9 |
| Modeling tools | Star UML 2.8.0 |

Table 5: 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

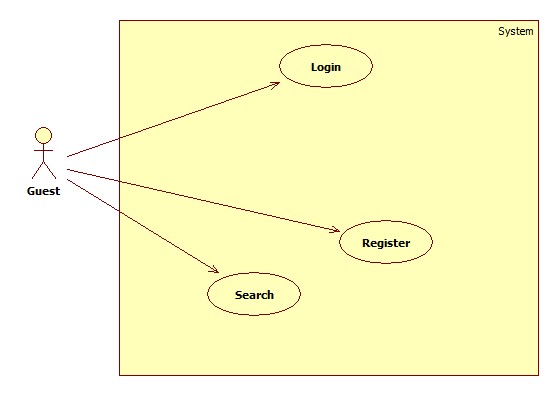


Figure 3: <Guest> Overview Use Case

##### <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

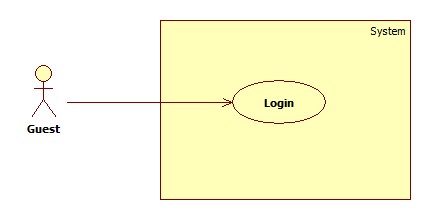


Figure 5: <Guest> Login

|  |  |  |  |
| --- | --- | --- | --- |
| **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

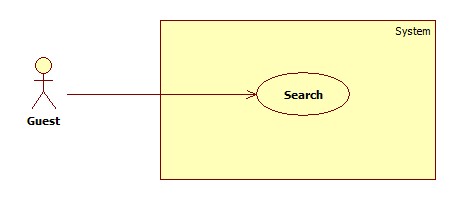


Figure 6: <Guest> Search

|  |  |  |  |
| --- | --- | --- | --- |
| **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

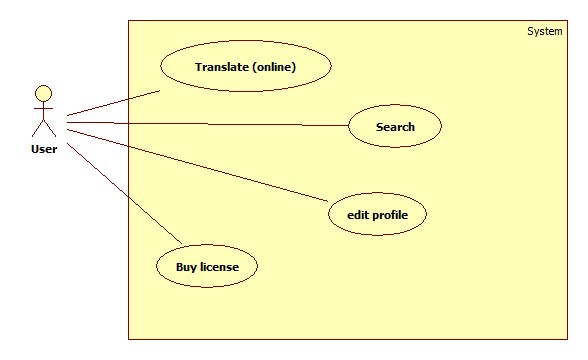


Figure 7: <User> Overview Use Case

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

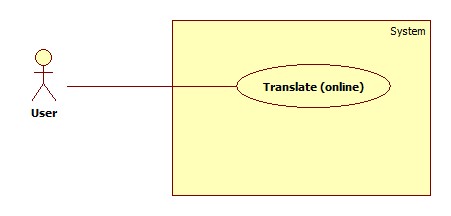


Figure 8: <User> Translate (Online)

|  |  |  |  |
| --- | --- | --- | --- |
| **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

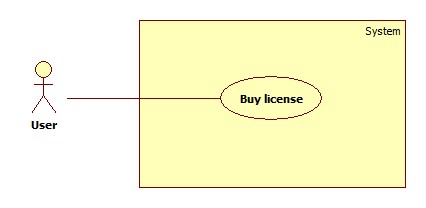


Figure 9: <User> Buy license

|  |  |  |  |
| --- | --- | --- | --- |
| **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

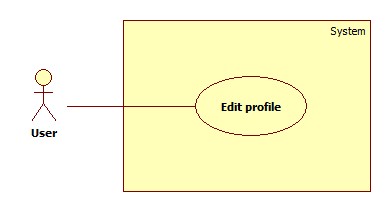


Figure 10: <User> Edit profile

|  |  |  |  |
| --- | --- | --- | --- |
| **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

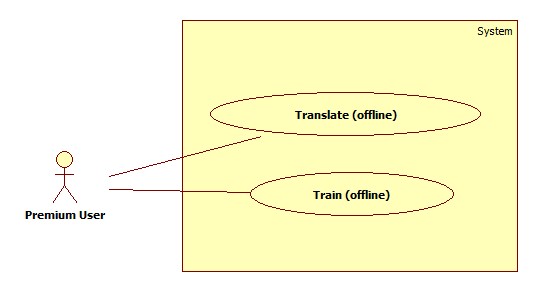


Figure 11: <Premium User> Overview Use case

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

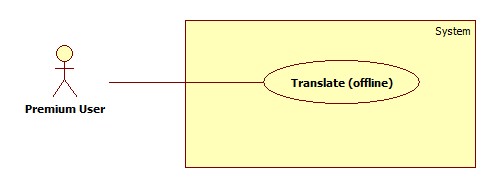


Figure 12: <Premium User> Translate (Offline)

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

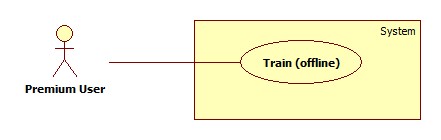


Figure 13: <Premium User> Train (Offline)

|  |  |  |  |
| --- | --- | --- | --- |
| **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

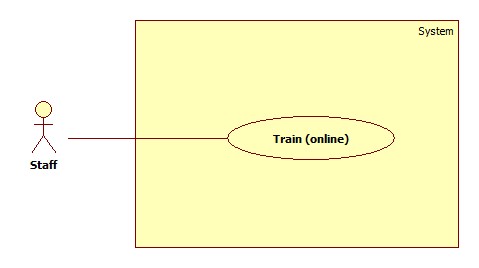


Figure 14: <Staff> Overview Use Case

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

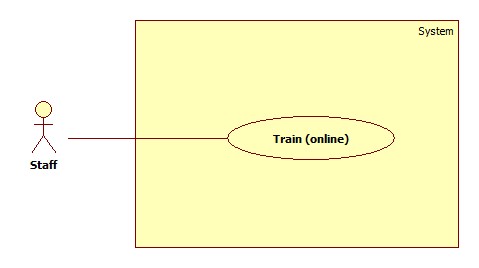


Figure 15: <Staff> Train (Online)

|  |  |  |  |
| --- | --- | --- | --- |
| **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

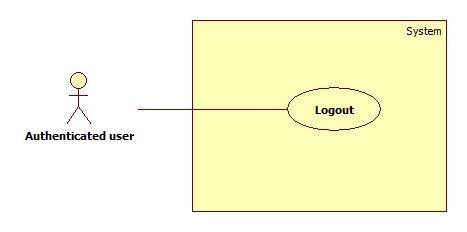


Figure 16: <Authenticated user> Overview Use Case

##### <Authenticated user> Logout

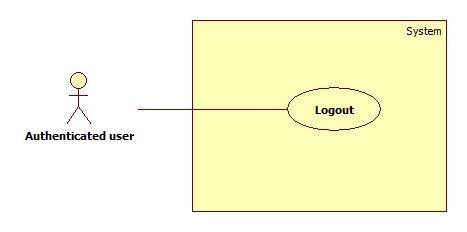


Figure 17: <Authenticated user> Logout

|  |  |  |  |
| --- | --- | --- | --- |
| **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

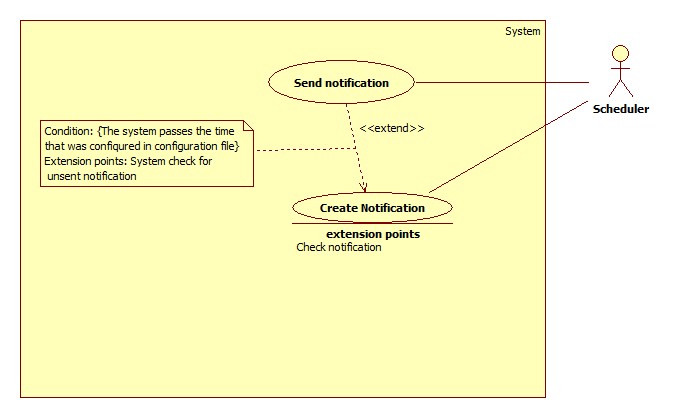


Figure 18: <Scheduler> Overview Use Case

##### <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

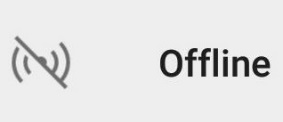
#### Graphic user interface

* 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

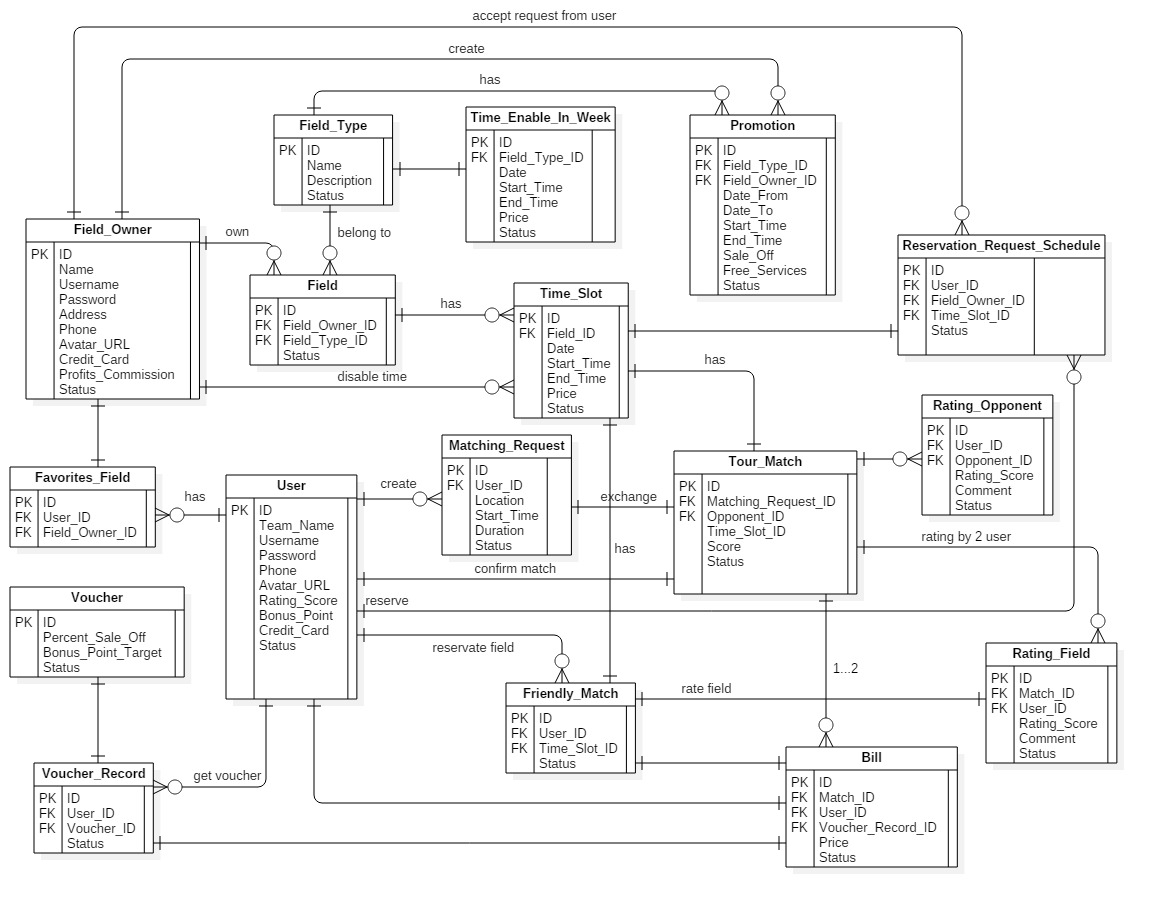


Figure 21: Conceptual diagram

|  |  |
| --- | --- |
| 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 **FFR 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.

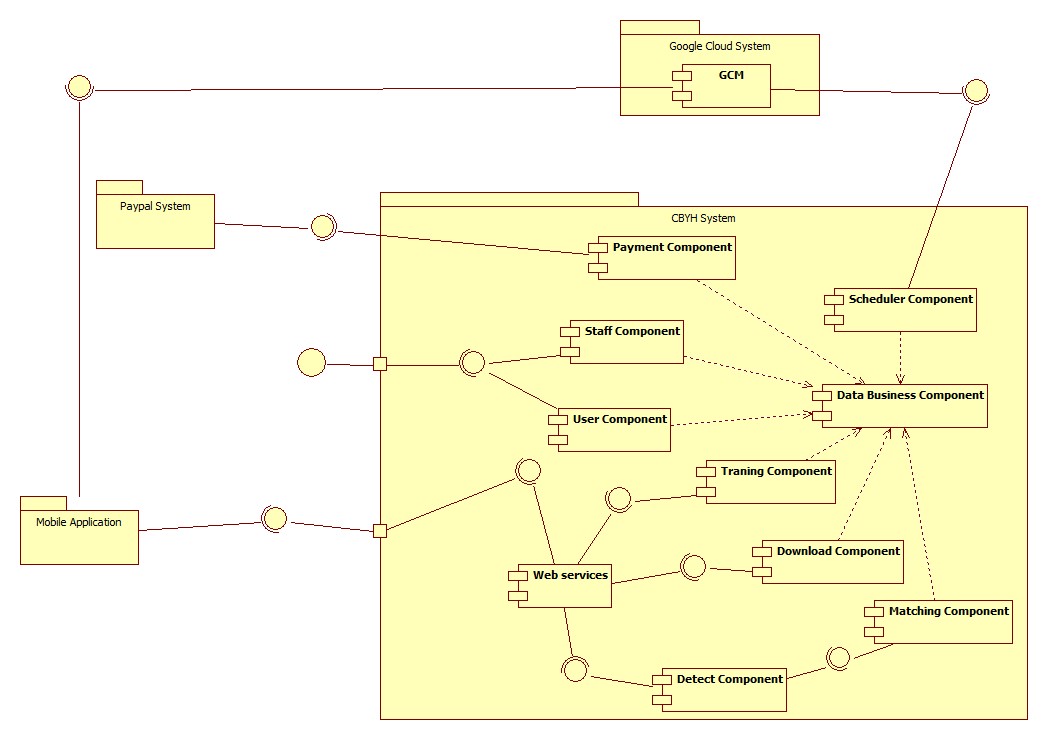
Reason for choosing: Currently, Android is the most common mobile platform in the world. Beside that, Android is the only one mobile platform that team member were taught in university.



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

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

## Component Diagram



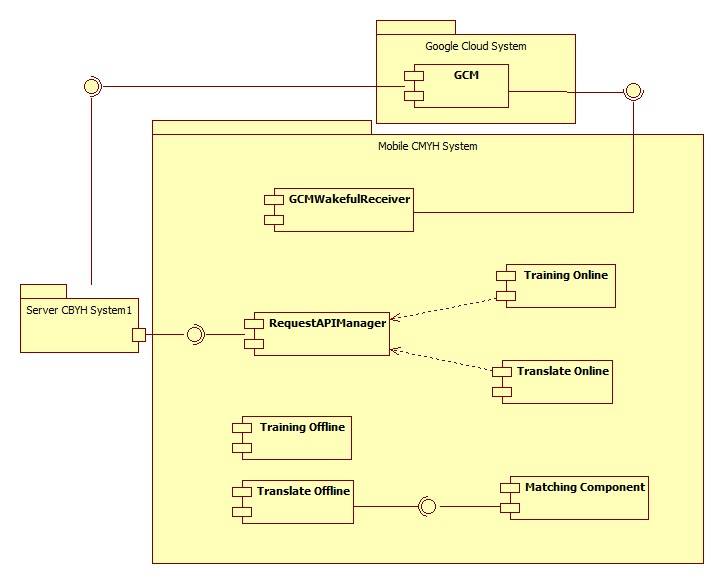
Figure 23: Component Diagram (Server side)

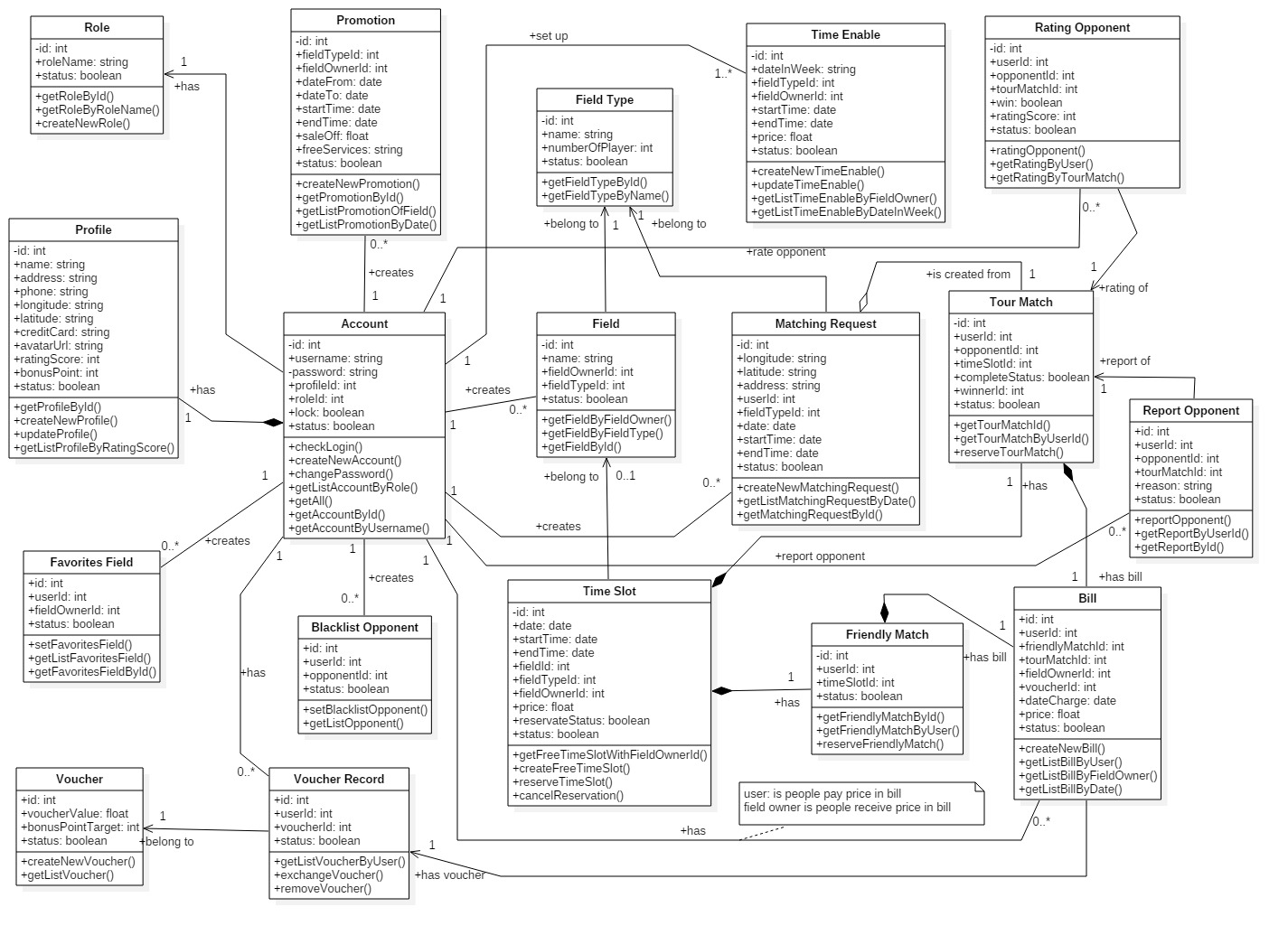
Figure 24: Component Diagram (Mobile side)

|  |  |
| --- | --- |
| Component Dictionary: Describes components | |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

Table 26: Component Dictionary

## Detail Description

### Class Diagram

Figure 25: Class Diagram

|  |  |
| --- | --- |
| Class dictionary: describe Class | |
| Class Name | **Description** |
| *Account* | Contain the account information |
| *Profile* | Contain the profile information |
| *Role* | Contain the role information |
| *Promotion* | Contain the promotion information |
| *FavoriteField* | Contain the favorite field information |
| *BlacklistOpponent* | Contain the blacklist opponent information |
| *Voucher* | Contain the voucher information |
| *VoucherRecord* | Contain the voucher record information |
| *Field* | Contain the field information |
| *FieldType* | Contain the field type information |
| *TimeEnable* | Contain the time enable information |
| *TimeSlot* | Not exists in conceptual diagram but need to contain the time slot when field owner create time enable and user create reservation requests |
| *MatchingRequest* | Contain the matching request information |
| *TourMatch* | Contain the tour match information |
| *FriendlyMatch* | Contain the friendly match information |
| *Bill* | Contain the bill information |
| *RattingOpponent* | Contain the ratting opponent information |
| *ReportOpponent* | Contain the custom content information |

Table 27: Class dictionary

### Class Diagram Explanation

#### Account

|  |  |  |  |
| --- | --- | --- | --- |
| Attribute | Type | Visibility | Description |
| id | Integer | Private | Unique identifier of an account |
| username | String | Private | Username of Account |
| password | String | Private | Password of Account |
| profileId | Integer | Private | Identifier of the profile of Account |
| roleId | Integer | Private | Identifier of the role of Account |
| lock | Boolean | Private | Staff request lock Actor |
| status | Boolean | Private | Active account checker |

Table 28: Account Attributes

#### Profile

|  |  |  |  |
| --- | --- | --- | --- |
| Attribute | Type | Visibility | Description |
| id | Integer | Private | Unique identifier of a profile |
| name | String | Private | Name of Actor |
| address | String | Private | Address of Actor |
| phone | String | Private | Phone of Actor |
| longitude | String | Private | Longitude of Actor |
| latitude | String | Private | Latitude of Actor |
| creditCard | String | Private | Credit card of Actor |
| avatarUrl | String | Private | Avatar url of Actor |
| rattingScore | Integer | Private | Ratting score of Actor |
| bonusPoint | Integer | Private | Bonus point of Actor |
| status | Boolean | Private | Active profile checker |

Table 29: Profile Attributes

#### Role

|  |  |  |  |
| --- | --- | --- | --- |
| Attribute | Type | Visibility | Description |
| id | Integer | Private | Unique identifier of a role |
| roleName | String | Private | Name of the role |
| status | Boolean | Private | Active role checker |

Table 30: Role Attributes

#### Promotion

|  |  |  |  |
| --- | --- | --- | --- |
| Attribute | Type | Visibility | Description |
| id | Integer | Private | Unique identifier of a promotion |
| fieldTypeId | Integer | Private | Id of field type that promotion belongs |
| fieldOwnerId | Integer | Private | Id of field owner that promotion belongs |
| dateFrom | Date | Private | Sent notification tracker |
| dateTo | Date | Private | Notification content |
| startTime | Date | Private | Time to start apply promotion (in a day) |
| endTime | Date | Private | Time to end apply promotion (in a day) |
| saleOff | Float | Private | Sale off percent of promotion |
| freeServices | String | Private | Free services content |
| status | Boolean | Private | Active promotion checker |

Table 31: Promotion Attributes

#### FavoriteField

|  |  |  |  |
| --- | --- | --- | --- |
| Attribute | Type | Visibility | Description |
| id | Integer | Private | Unique identifier of a favorite field |
| userId | Integer | Private | Id of the user that the favorite field belongs |
| fieldOwnerId | Integer | Private | Id of the field owner that the favorite field belongs |
| status | Boolean | Private | Active favorite field checker |

Table 32: FavoriteField Attributes

#### BlacklistOpponent

|  |  |  |  |
| --- | --- | --- | --- |
| Attribute | Type | Visibility | Description |
| id | Integer | Private | Unique identifier of a black list opponent |
| userId | Integer | Private | Id of the user that the black list opponent belongs |
| opponentId | Integer | Private | Id of the opponent that the black list opponent belongs |
| status | Boolean | Private | Active black list checker |

Table 33: Blacklist Attributes

#### Voucher

|  |  |  |  |
| --- | --- | --- | --- |
| Attribute | Type | Visibility | Description |
| id | Integer | Private | Unique identifier of a voucher |
| voucherValue | Float | Private | Value of the voucher |
| bonusPointTarget | Integer | Private | Number of bonus point to exchange the voucher |
| status | Boolean | Private | Active voucher checker |

Table 34: Voucher Attributes

#### VoucherRecord

|  |  |  |  |
| --- | --- | --- | --- |
| Attribute | Type | Visibility | Description |
| id | Integer | Private | Unique identifier of a voucher record |
| userId | Integer | Private | Id of the user that the voucher record belongs |
| voucherId | Integer | Private | Id of the voucher that the voucher record belongs |
| status | Boolean | Private | Active voucher checker |

Table 35: VoucherRecord Attributes

#### Field

|  |  |  |  |
| --- | --- | --- | --- |
| Attribute | Type | Visibility | Description |
| id | Integer | Private | Unique identifier of a field |
| name | String | Private | Name of a field |
| fieldOwnerId | Integer | Private | Id of the field owner that the field belongs |
| fieldTypeId | Integer | Private | Id of the field type that the field belongs |
| status | Boolean | Private | Active field checker |

Table 36: Field Attributes

#### FieldType

|  |  |  |  |
| --- | --- | --- | --- |
| Attribute | Type | Visibility | Description |
| id | Integer | Private | Unique identifier of a field type |
| name | String | Private | Name of a field type |
| numberOfPlayer | Integer | Private | Number of player of the field type |
| status | Boolean | Private | Active field type checker |

Table 37: FieldType Attributes

#### TimeEnable

|  |  |  |  |
| --- | --- | --- | --- |
| Attribute | Type | Visibility | Description |
| id | Integer | Private | Unique identifier of a time enable |
| dateInWeek | String | Private | Date in week that the time enable belongs |
| fieldTypeId | Integer | Private | Id of the field type that the time enable belongs |
| fieldOwnerId | Integer | Private | Id of the field owner that the time enable belongs |
| startTime | Date | Private | Start time of the time enable |
| endTime | Date | Private | End time of the time enable |
| price | Float | Private | Price (per hour) of the time enable |
| status | Boolean | Private | Active time enable checker |

Table 38: TimeEnable Attributes

#### TimeSlot

|  |  |  |  |
| --- | --- | --- | --- |
| Attribute | Type | Visibility | Description |
| id | Integer | Private | Unique identifier of a time slot |
| dateInWeek | String | Private | Date in week that the time slot belongs |
| startTime | Date | Private | Start time of the time slot |
| endTime | Date | Private | End time of the time slot |
| fieldId | Integer | Private | Id of the field that the time slot belongs |
| fieldTypeId | Integer | Private | Id of the field type that the time slot belongs |
| fieldOwnerId | Integer | Private | Id of the field owner that the time slot belongs |
| price | Float | Private | Price (per hour) of the time slot |
| reservateStatus | Boolean | Private | Active reserve checker |
| status | Boolean | Private | Active time slot checker |

Table 39: TimeSlot Attributes

#### MatchingRequest

|  |  |  |  |
| --- | --- | --- | --- |
| Attribute | Type | Visibility | Description |
| id | Integer | Private | Unique identifier of the matching request |
| longitude | String | Private | Longitude of the matching request |
| latitude | String | Private | Latitude of the matching request |
| address | String | Private | Address of the matching request |
| userId | Integer | Private | Id of the user that the matching request belongs |
| fieldTypeId | Integer | Private | Id of the field type that the matching request belongs |
| date | Date | Private | Date of the matching request |
| startTime | Date | Private | Start time of the matching request |
| endTime | Date | Private | End time of the matching request |
| status | Boolean | Private | Active matching request checker |

Table 40: Matching request Attributes

#### TourMatch

|  |  |  |  |
| --- | --- | --- | --- |
| Attribute | Type | Visibility | Description |
| id | Integer | Private | Unique identifier of the tour match |
| userId | Integer | Private | Id of the user that the tour match belongs |
| opponentId | Integer | Private | Id of the opponent that the tour match belongs |
| timeSlotId | Integer | Private | Id of the time slot that the tour match belongs |
| completeStatus | Boolean | Private | Active complete checker |
| winnerId | Integer | Private | Id of the winner that the tour match belongs |
| status | Boolean | Private | Active tour match checker |

Table 41: TourMatch Attributes

#### FriendlyMatch

|  |  |  |  |
| --- | --- | --- | --- |
| Attribute | Type | Visibility | Description |
| id | Integer | Private | Unique identifier of the friendly match |
| userId | Integer | Private | Id of the user that friendly match belongs |
| timeSlotId | Integer | Private | Id of the time slot that friendly match belongs |
| status | Boolean | Private | Active friendly match checker |

Table 42: FriendlyMatch Attributes

#### Bill

|  |  |  |  |
| --- | --- | --- | --- |
| Attribute | Type | Visibility | Description |
| id | Integer | Private | Unique identifier of a bill |
| userId | Integer | Private | Id of the user that the bill belongs |
| friendlyMatchId | Integer | Private | Id of the friendly match that the bill belongs |
| tourMatchId | Integer | Private | Id of the tour match that the bill belongs |
| fieldOwnerId | Integer | Private | Id of the field owner that the bill belongs |
| voucherId | Integer | Private | Id of the voucher that the bill belongs |
| dateCharge | Date | Private | Date charge the bill |
| price | Float | Private | Price of the bill |
| status | Boolean | Private | Active bill checker |

Table 43: Bill Attributes

#### RattingOpponent

|  |  |  |  |
| --- | --- | --- | --- |
| Attribute | Type | Visibility | Description |
| id | Integer | Private | Unique identifier of a ratting opponent |
| userId | Integer | Private | Id of the user that the ratting belongs |
| opponentId | Integer | Private | Id of the opponent that the ratting belongs |
| tourMatchId | Integer | Private | Id of the tour match that the ratting belongs |
| win | Boolean | Private | Active win checker |
| rattingScore | Integer | Private | Ratting score of the ratting |
| status | Boolean | Private | Active ratting checker |

Table 43: RattingOpponent Attributes

#### ReportOpponent

|  |  |  |  |
| --- | --- | --- | --- |
| Attribute | Type | Visibility | Description |
| id | Integer | Private | Unique identifier of a report opponent |
| userId | Integer | Private | Id of the user that the report belongs |
| opponentId | Integer | Private | Id of the opponent that the report belongs |
| tourMatchId | Integer | Private | Id of the tour match that the report belongs |
| reason | String | Private | Reason of the report |
| status | Boolean | Private | Active report checker |

Table 43: ReportOpponent Attributes

### Interactive Diagram

#### Web Application

##### <FieldOwner>

###### Buy license

Summary: this diagram show process of buy license

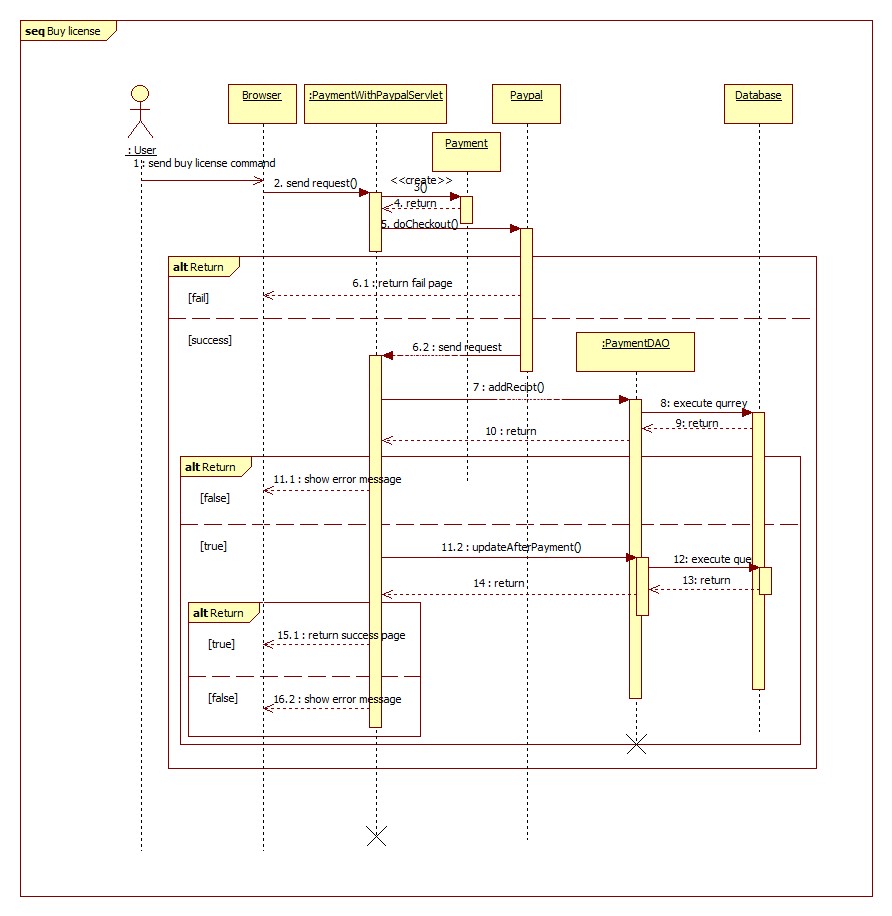


Figure 26: Sequence diagram - <User> Buy license

###### Update profile

Summary: this diagram show process of update profile

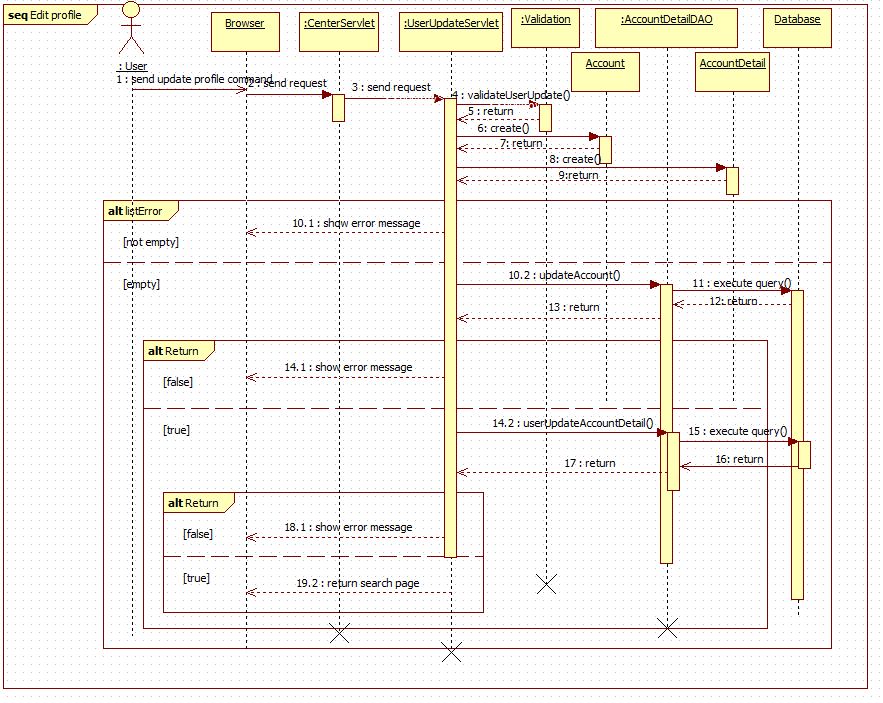


Figure 27: Sequence diagram - <User> Update profile

###### Search

Summary: this diagram show process of search

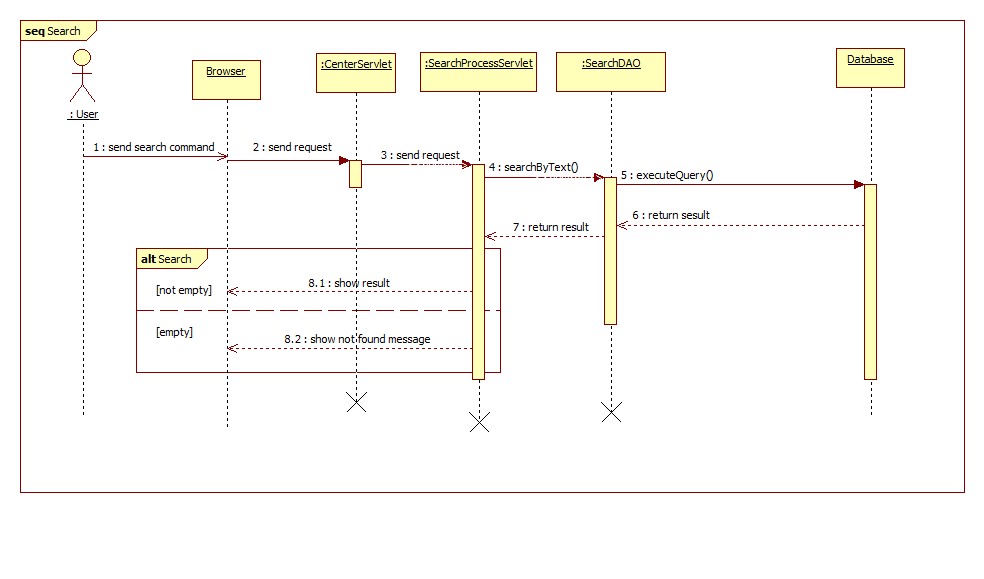


Figure 28: Sequence diagram - <User> Search

##### <Guest>

###### Register

Summary: this diagram show process of register

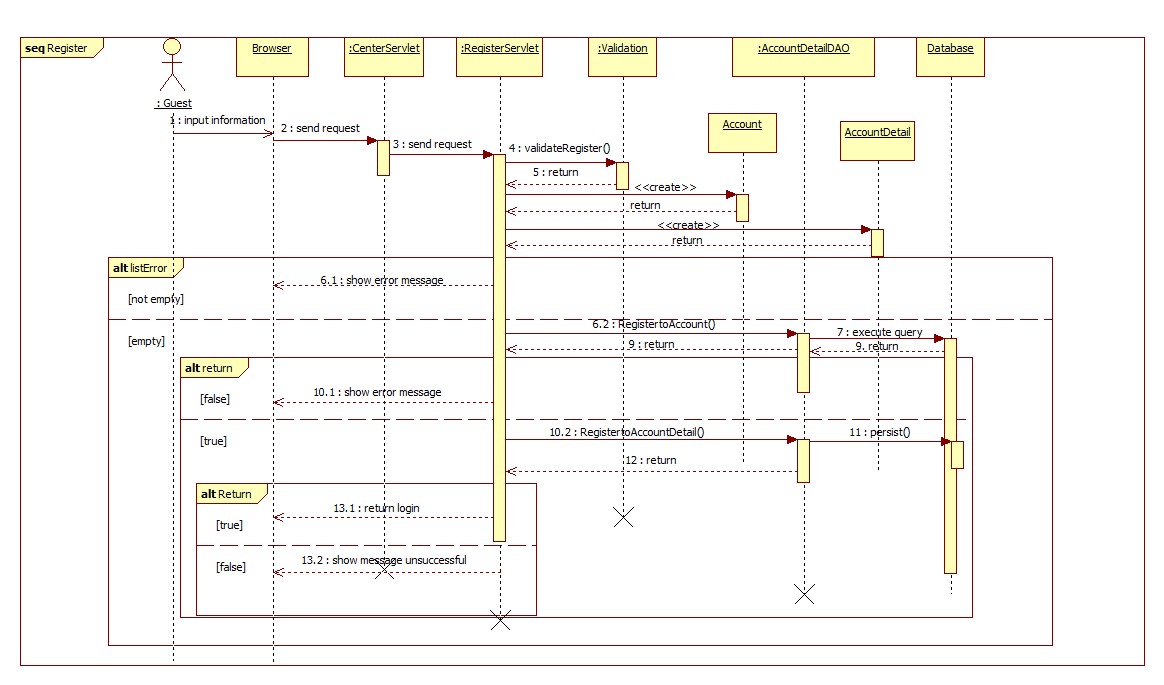


Figure 29: Sequence diagram - <Guest> Register

#### Mobile Application

##### <Staff>

###### Train online

Summary: this diagram show process of Train online

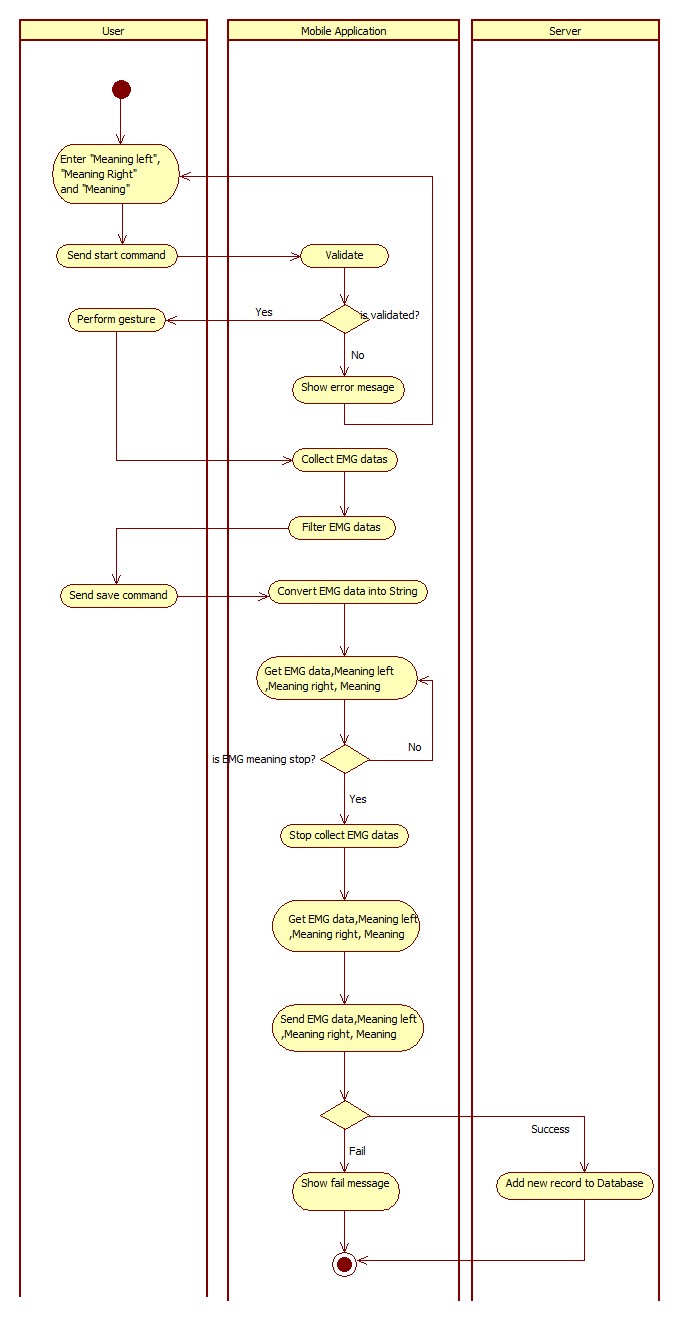


Figure 30: Activity diagram - <Staff> Train online

##### <User>

###### Connect Myo armbands

Summary: this diagram show process of connect Myo armbands

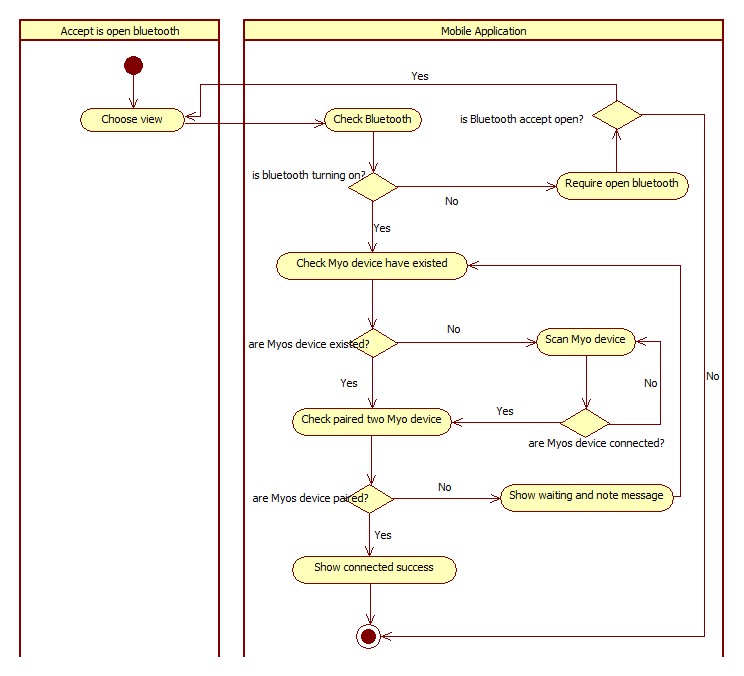


Figure 31: Activity diagram - <User> Connect Myo armbands

###### Translate online

Summary: this diagram show process of translate online

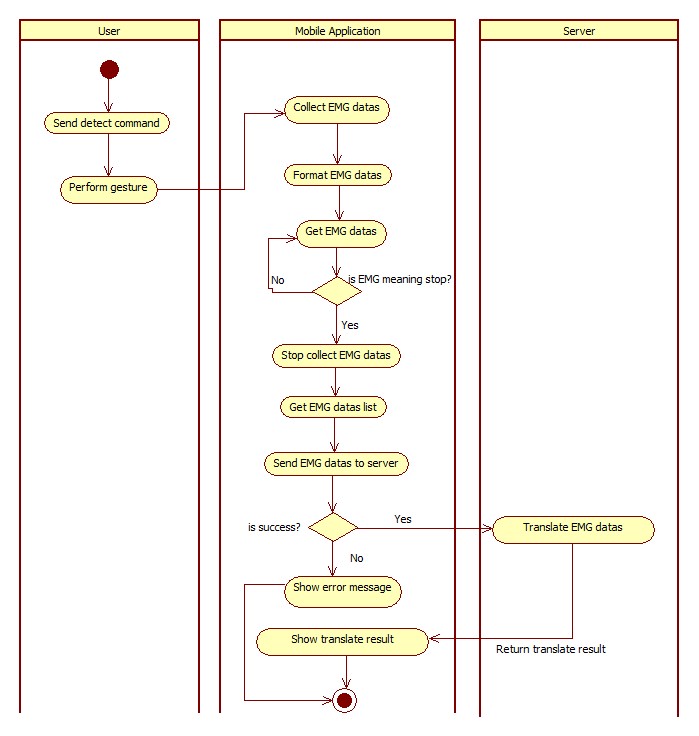


Figure 32: Activity diagram - <User> Translate online

##### <Premium User>

###### Translate offline

Summary: this diagram show process of translate offline

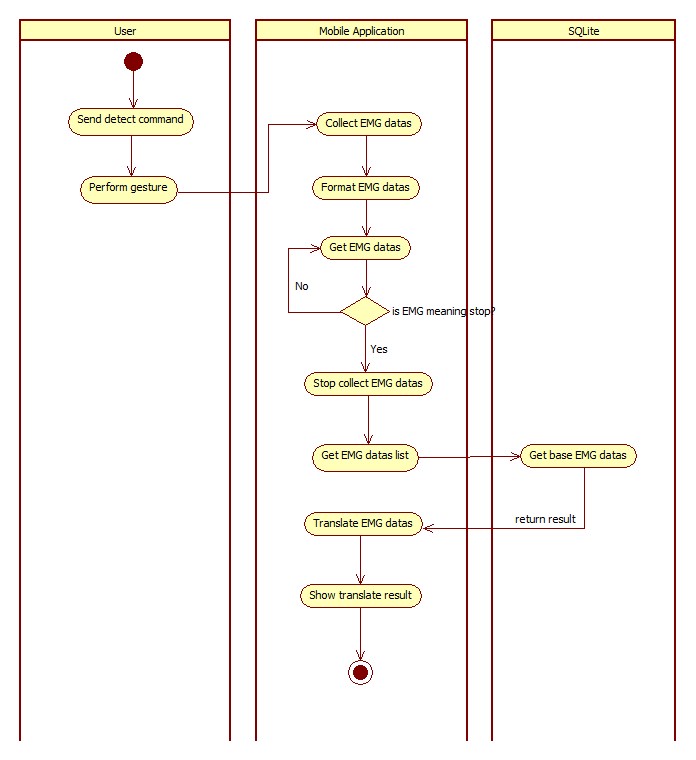


Figure 33: Activity diagram - <Premium User> Translate offline

###### 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

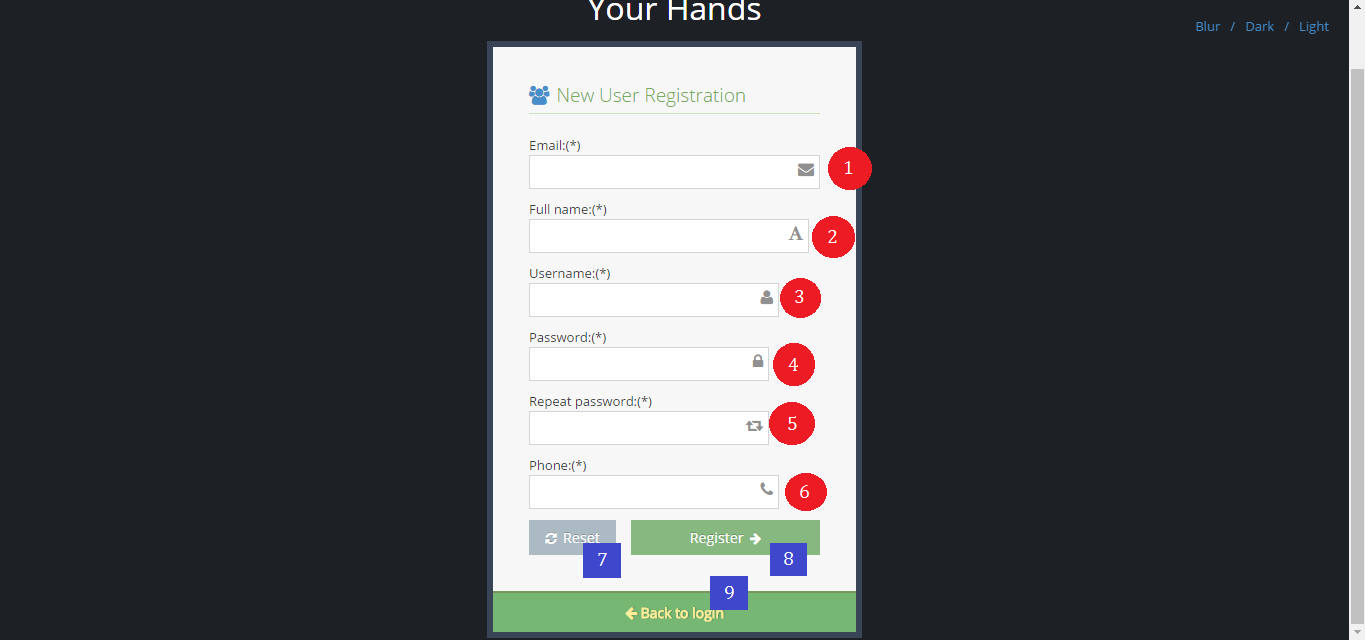


Figure 35: Interface - <Guest> Register

**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

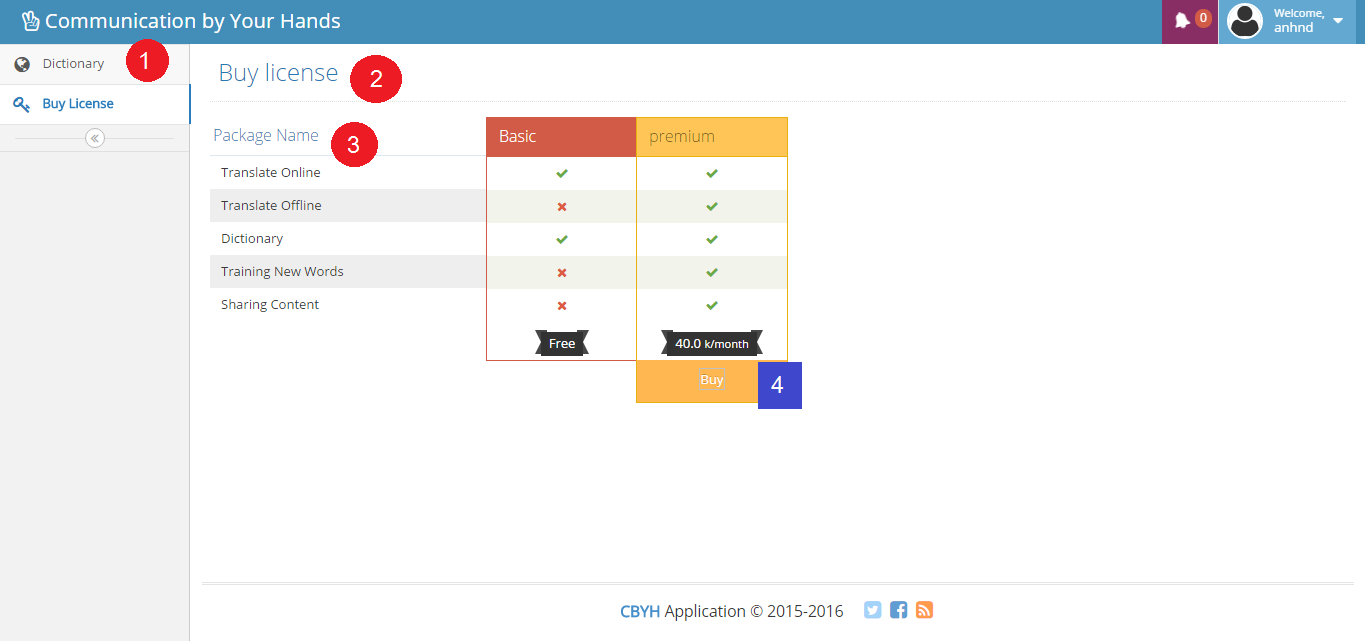


Figure 36: Interface - <User> Buy license

**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

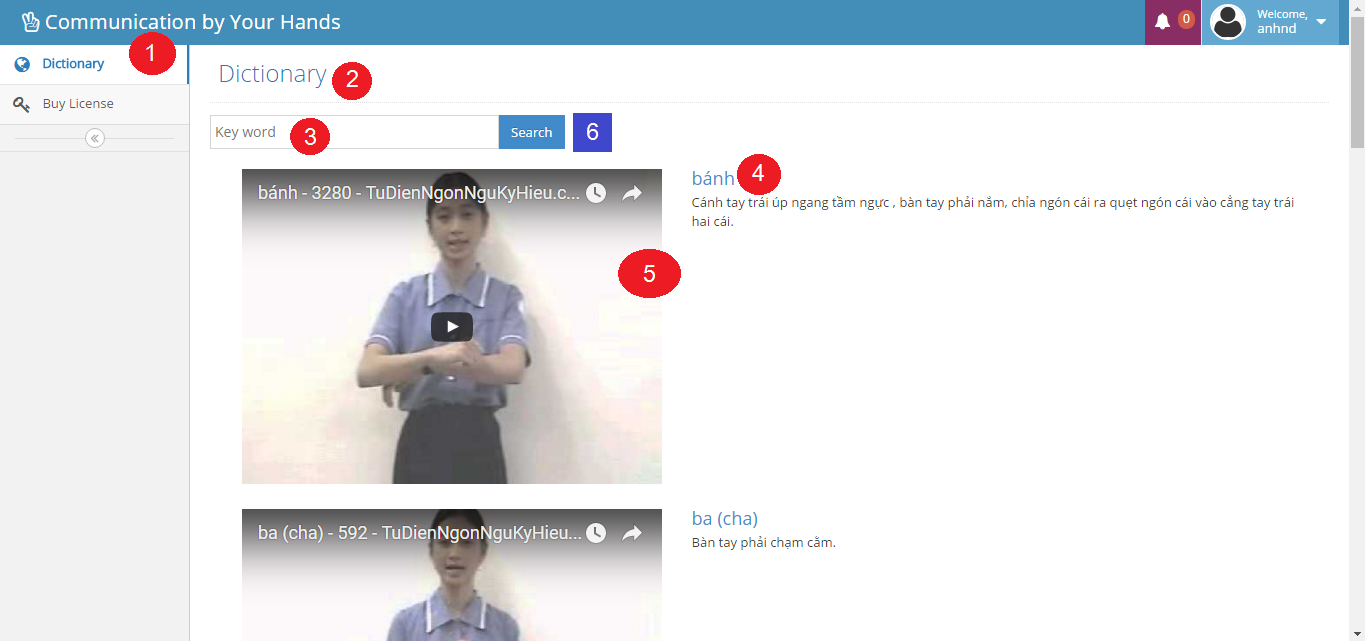


Figure 37: Interface - <User> Search instruction

**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

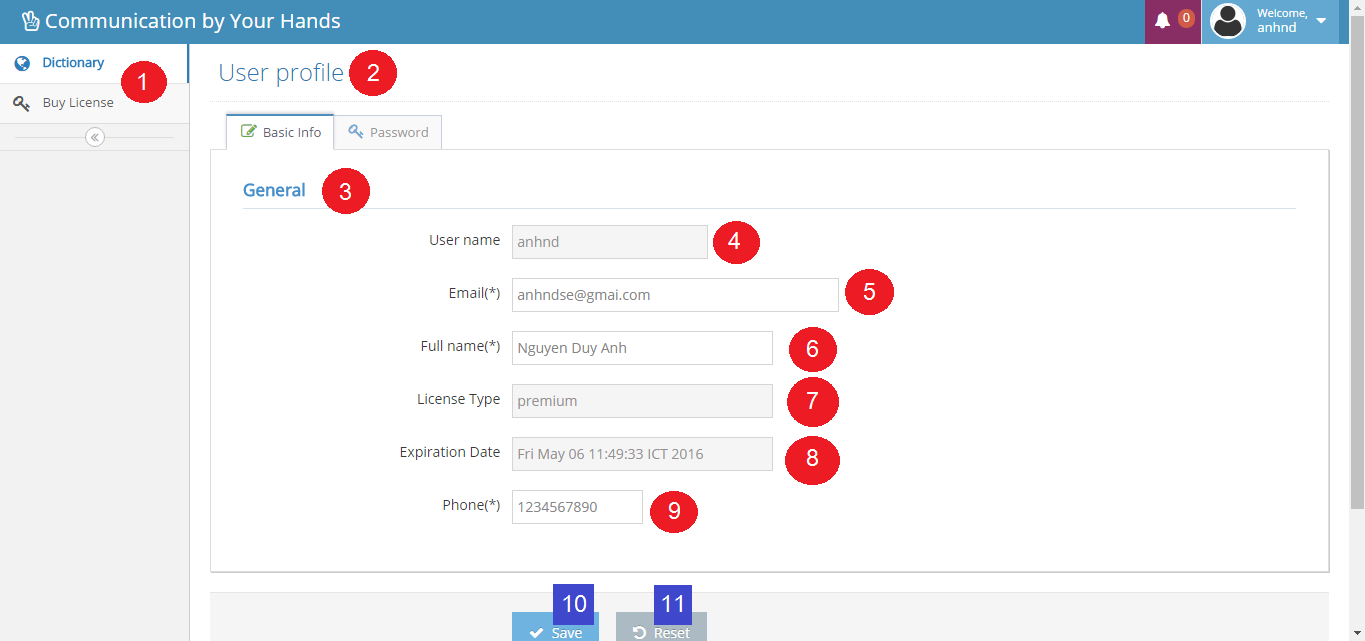


Figure 38: Interface - <User> Edit user profile

**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

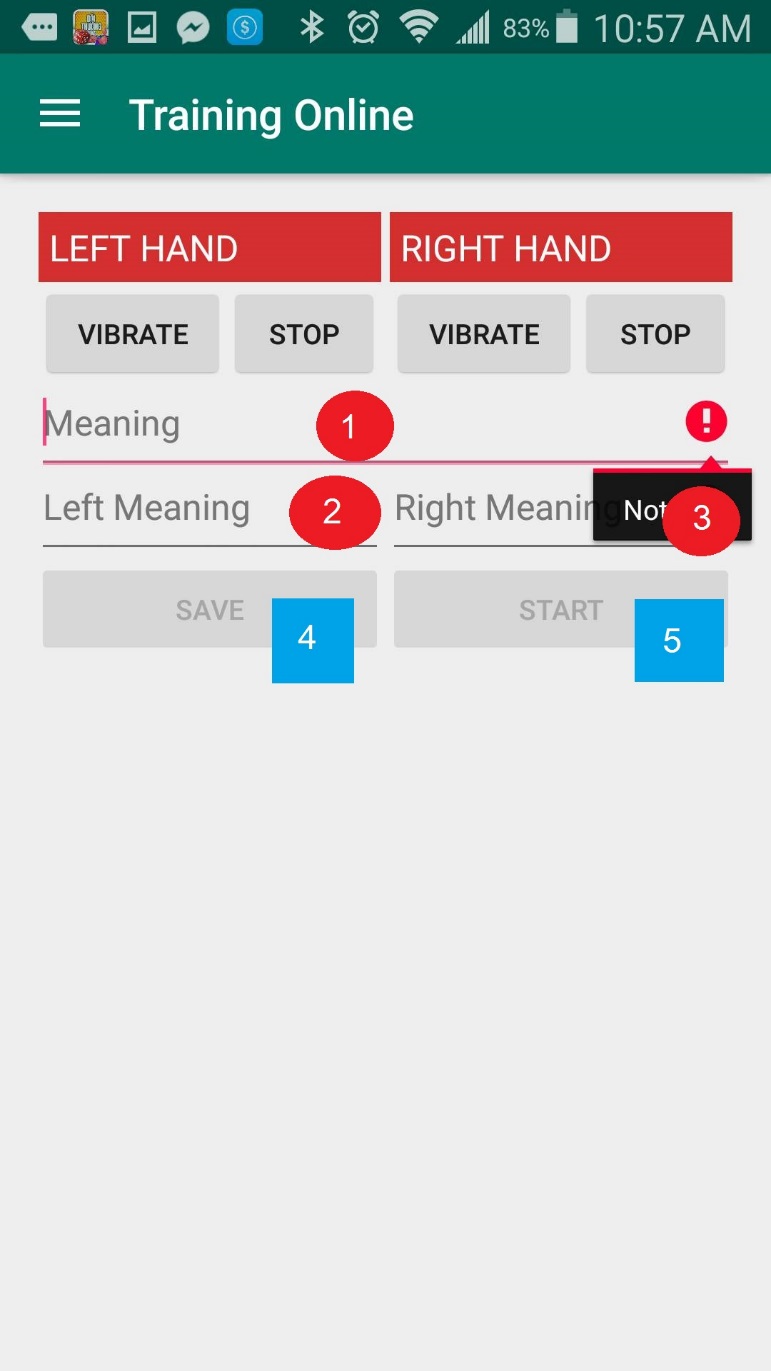


Figure 39: Interface - <Staff> Train online

**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

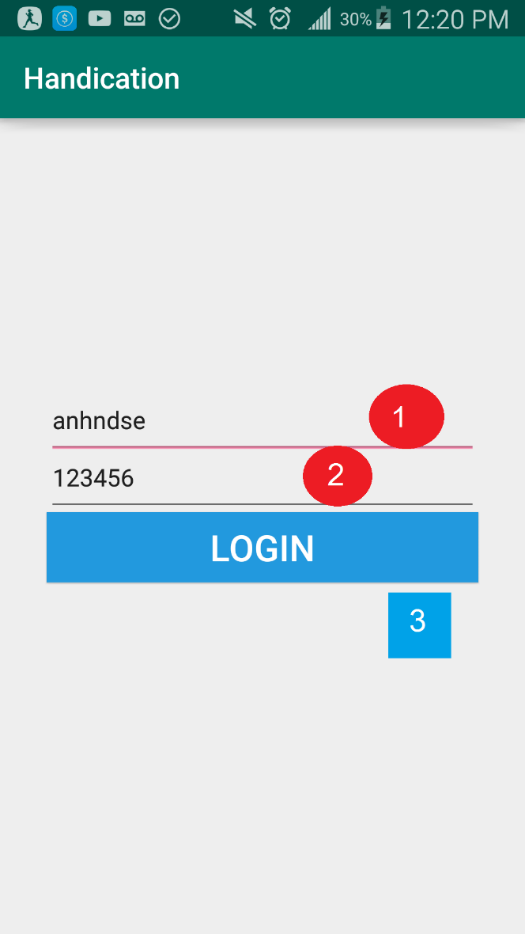


Figure 40: Interface - <User> Login

**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

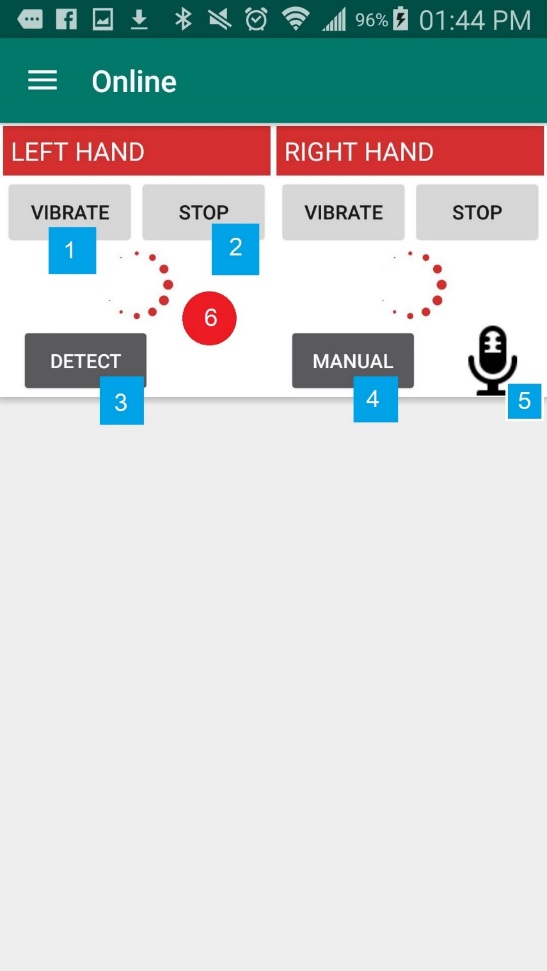


Figure 41: Interface - <User> Translate – Translating

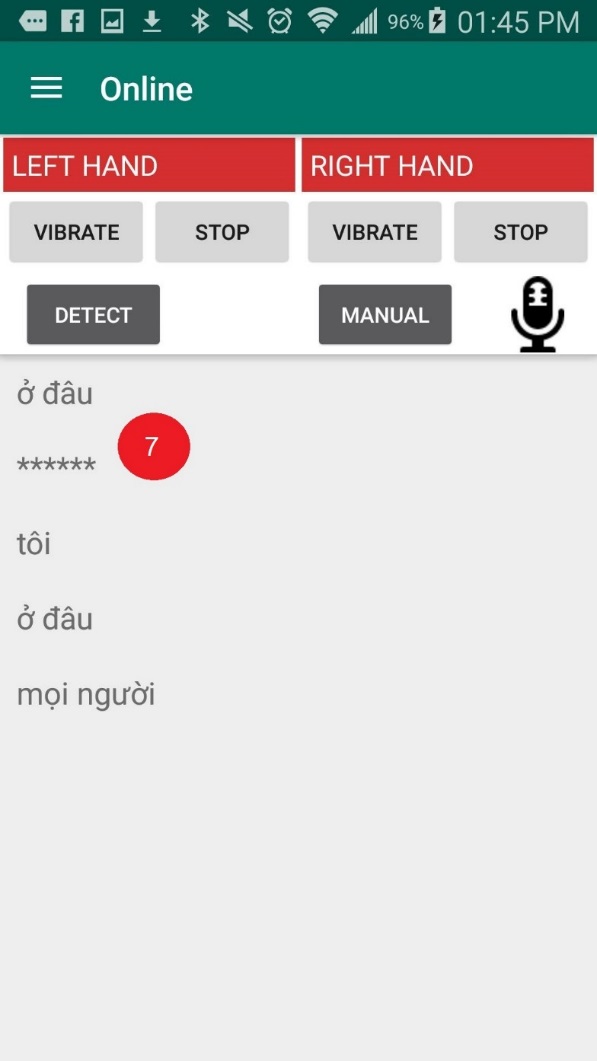


Figure 42: Interface - <User> Translate – Translated

**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

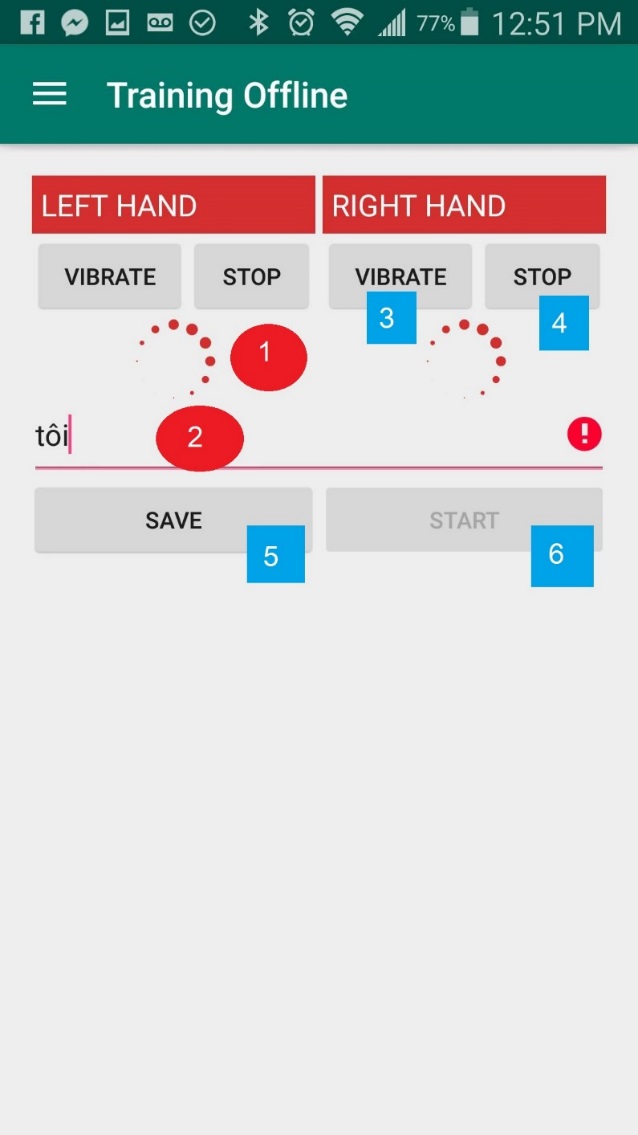


Figure 43: Interface - <Premium user> Train offline

**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

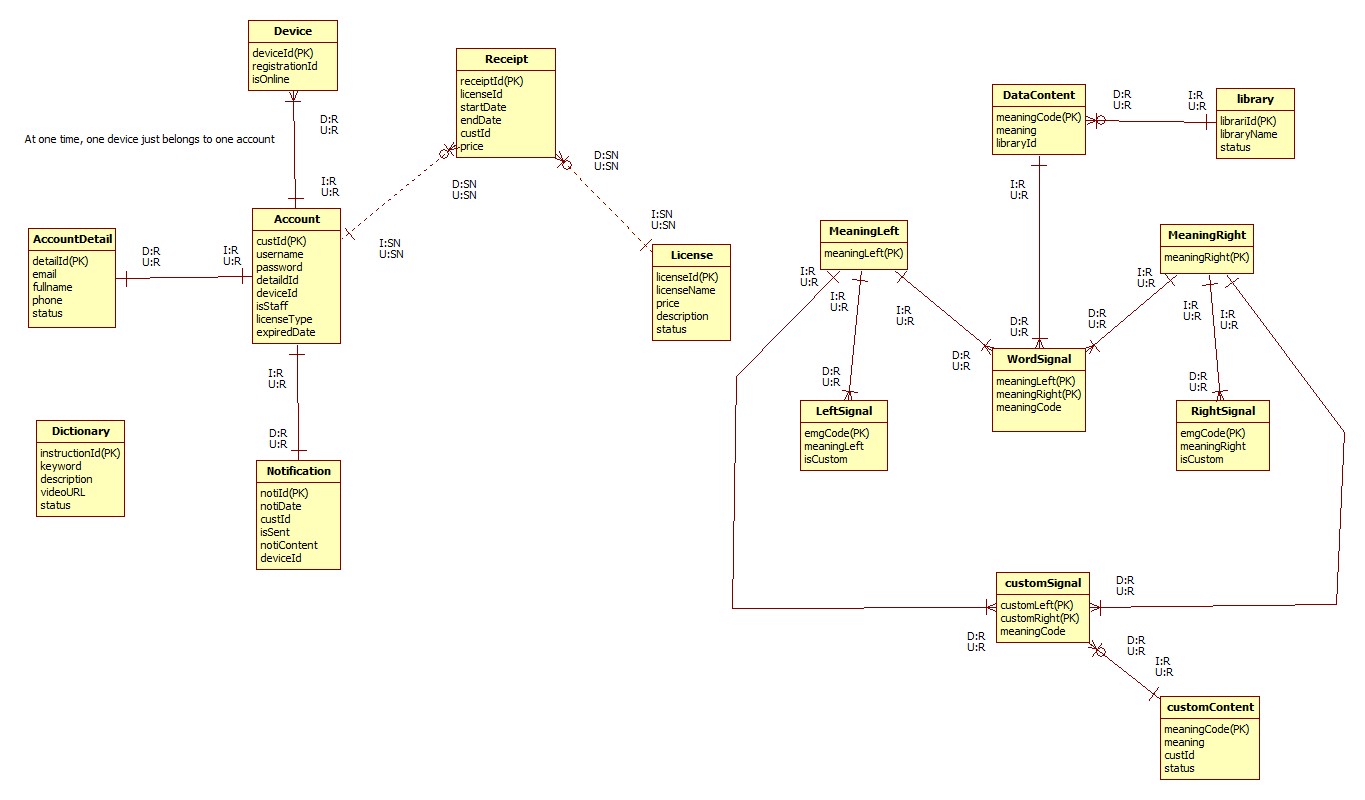


Figure 44: Entity relationship diagram

### 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

### Manage time

#### Definition

The field management system is based on free time of the field, so that the System can quickly know which field owner can satisfy the user’s request.

#### Define problem

The User may select the time frame arbitrarily and then reserves the field. The problem is how system can manage time of fields. System has to calculate free time quickly and effectively.

When a large number of users reserve field, the field’s timeline may be broken into fragments. Therefore, system has to choose field owners who received matching requests.

#### Solution

The field management system is based on free time of the field, so that the system can decide which field owner can satisfy the user’s request. If H (0, ..., k, ..., 23) is the set of vacant fields for one hour from k (H (k), the number of vacant fields between k and k + 1 hour).

Let F(1, .., i, n) is the set of fields of a field owner. User A reserve field from x to y hour(s).

Step 1: The System will check H (k) with x <= k <= y-1 with all H (k)> 0, there exists at least 1 vacant field from x to y.

Step 2: Suppose F(i) is the field that Field owner choose for user A.

For (j = x, j = j +1, j<=y)

If (F(i) is free at j hour) then

User can reserve this field as normal till j hour

If (F(i) is reserved by other user) then

1. Field owner find other field F(ii) which has free time at j hour
2. Swap all reserve request of F(i) and F(ii) from j hour
3. After swap F(i) is free at j hour

#### Example

Two fields of field owner “Cây Trâm”:

+ Field 1: 06AM to 22PM

+ Field 2: 06AM to 22PM

Reserve request:

+ User A reserved Field 1, time frame is: 06AM to 08AM

+ User B reserved Field 2, time frame is: 08AM to 10AM

New reserve request:

+User C reserves time frame 07AM to 09AM

**Field owner set field for Use**r:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Field** | **06AM** | **07AM** | **08AM** | **09AM** | **10AM** | **11AM** |
| Field 1 | User A | User A | 0 | 0 | 0 | 0 |
| Field 2 | 0 | 0 | User B | User B | 0 | 0 |

**After system apply algorithm:**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Field** | **06AM** | **07AM** | **08AM** | **09AM** | **10AM** | **11AM** |
| Field 1 | User A | User A | User B | User B | 0 | 0 |
| Field 2 | 0 | User C | User C | 0 | 0 | 0 |

#### Complexity

The complexity is: O(n2)

### Distance

#### Definition

The situation is: when system finds fields around 1 location A (latitude1, longitude1), with k(km) radius.

If location B (latitude2, longtitude2) is within a radius of k(km) from A, the precondition is:

x, y are constants.

The main purpose is we calculate the approximately x, y then input to the system. Based on x, y, system can quickly find out which fields is the most suitable. X, y constants help system early removal fields can not afford.

#### Define problem

Calculating the distance between 2 points is one of the most used functions in our system (user reserves field, system finds opponent, system matches 2 users).

Calculating the distance between 2 points takes so many steps.

#### Solution

We calculate x, y as 2 constants. System based on x, y can calculate faster.

For any two points on a sphere, the haversine of the central angle between them is given by:

hav is the Havesin fuction:

* *d* is the distance between the two points
* *r* is the radius of the sphere,
* *φ*1, *φ*2: latitude of point 1 and latitude of point 2, in radians
* *λ*1, *λ*2: longitude of point 1 and longitude of point 2, in radians

Note that φ and λ can be converted from radians to degrees by multiplying by  as usual.

A (latitude1, longitude1), B (latitude2, longtitude2)

The latitude of Ho Chi Minh City, Vietnam is 10.762622, and the longitude is 106.660172.

* Let latitude1 = latitude2 = 10.76, d=5(km), r= 6.367.449(m)
  + - longitude2 = longitude1±0.0455 (in decimal)
* Let longitude1 = longitude2, d=5(km), r=6.367.449(m)
  + - latitude2 = latitude1±0.045 (in decimal)

After testing, the erroneous is less than 5%.

#### Example

A(10.8547079,106.6257702) (Google maps define as Công viên phần mềm quang trung)

B (latitude, longitude) has longitude the same as A, latitude as A+0.0455

B(10.9002079,106.6257702)

The distance between A and B is 5.04km (using google maps calculator)

C (latitude, longitude) has latitude the same as A, longitude as A+0.045

C (10.8547079, 106.6707702)

The distance between A and C is 4.98km (using google maps calculator)

#### Complexity

The complexity is: O(n)

### Matching 2 users

#### Definition

After 2 users create matching request, if 2 requests are matched, system will suggest list of fields for user.

#### Define problem

Usually, 2 requests will be matched by comparing time. However, just comparing time is not enough. We don’t know if they are as the same level (for example, a team is skilled and the other is not so good at playing football).

There are many other reasons why we have to build different criterions.

#### Solution

The system not only match users based on time but also rely on the user's rank, distance between two users and the list of their favorite fields to make a list of fields suitable for both.

Set favorite field and blacklist functions make the matching system more accurate and preferable to users. After completing a match, two users will rate the match, which enables the system to calculate 2 users’ rank more accurately.

The favorite field has high priority on the suitable list of fields. If a user is in the blacklist of other user, they can not match each other.

# 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 FFRS 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 FFRS 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

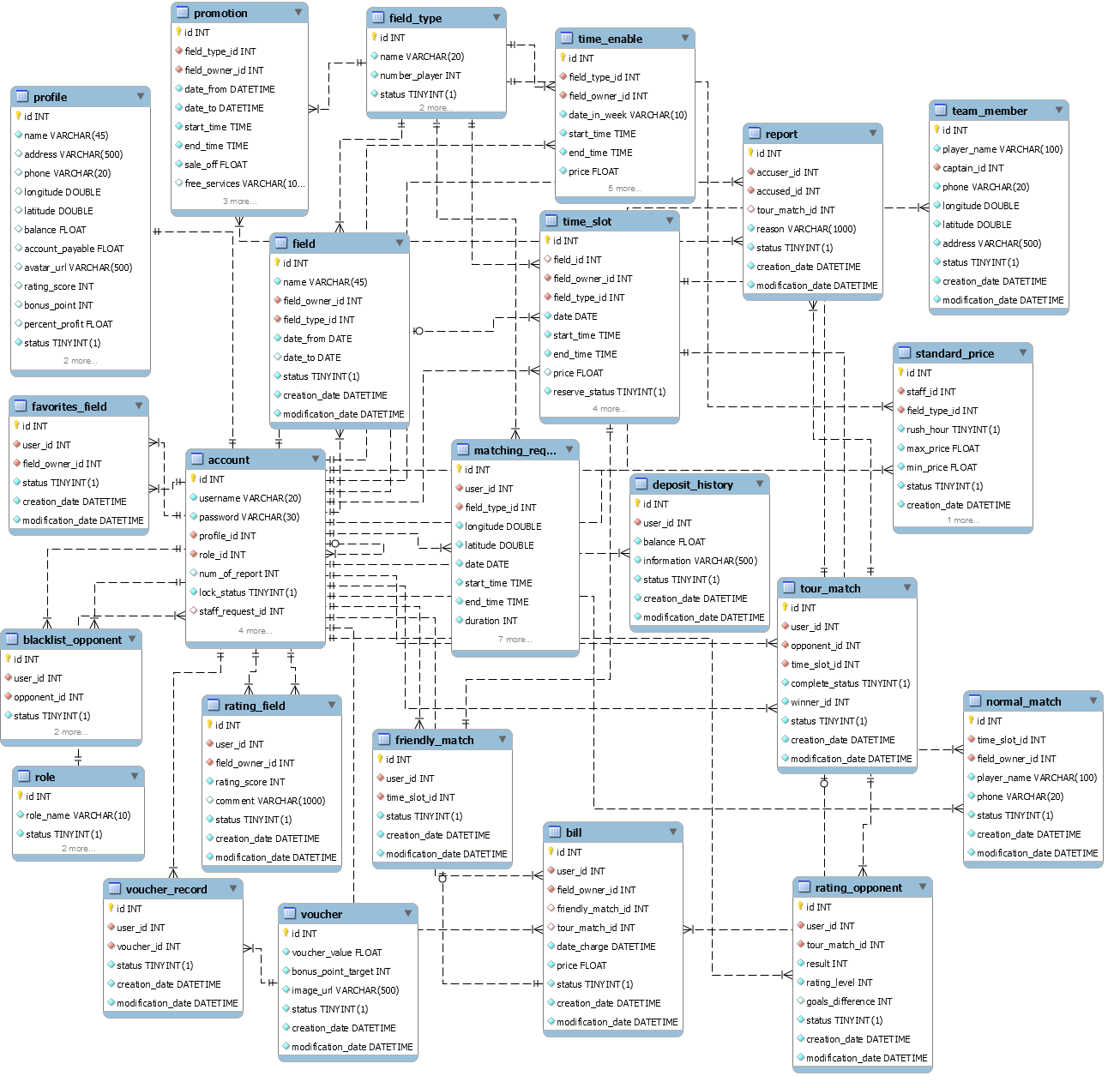


Figure 51: Physical diagram

|  |  |  |
| --- | --- | --- |
| Data dictionary: describe content of all tables | | |
| Table Name | **Mapping with Conceptual diagram** | **Description** |
| Account | account | Contain the account information |
| Profile | profile | Contain the profile information |
| Role | role | Contain the role information |
| Promotion | promotion | Contain the promotion information |
| FavoriteField | favoriteField | Contain the favorite field information |
| BlacklistOpponent | blacklistOpponent | Contain the blacklist opponent information |
| Voucher | voucher | Contain the voucher information |
| VoucherRecord | voucherRecord | Contain the voucher record information |
| Field | field | Contain the field information |
| FieldType | fieldType | Contain the field type information |
| TimeEnable | timeEnable | Contain the time enable information |
| TimeSlot | timeSlot | Not exists in conceptual diagram but need to contain the time slot when field owner create time enable and user create reservation requests |
| MatchingRequest | matchingRequest | Contain the matching request information |
| TourMatch | tourMatch | Contain the tour match information |
| FriendlyMatch | friendlyMatch | Contain the friendly match information |
| Bill | bill | Contain the bill information |
| RattingOpponent | rattingOpponent | Contain the ratting opponent information |
| ReportOpponent | reportOpponent | Contain the report opponent information |

### Data dictionary

Table 63: Data table dictionary

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Table Name | Attributes | Description | Domain | Null |
| Account | id(pk) | Unique identifier of an account | int | No |
| username | Username of Account | varchar(20) | No |
| password | Password of Account | varchar(30) | No |
| profile\_id | Identifier of the profile of Account | int | No |
| role\_id | Identifier of the role of Account | int | No |
| num\_of\_report | Number of report of Account | int | No |
| lock\_status | Staff request lock Account checker | bit | No |
| staff\_request\_id | Identifier of staff requested | int | Yes |
| request\_lock | Request lock account checker | bit | No |
| status | Active account checker | bit | No |
| Profile | id(pk) | Unique identifier of a profile | int | No |
| name | Name of profile | varchar(45) | No |
| address | Address of profile | varchar(500) | Yes |
| phone | Phone of profile | varchar(20) | Yes |
| longitude | Longitude of profile | double | Yes |
| latitude | Latitude of profile | double | Yes |
| balance | Balance of profile | float | Yes |
| account\_payable | Account payable of profile | float | Yes |
| avatar\_url | Avatar url of profile | varchar(500) | Yes |
| ratting\_score | Ratting score of profile | int | Yes |
| bonus\_point | Bonus point of profile | int | Yes |
| percent\_profit | Percent profit of profile | float | Yes |
| status | Active profile checker | bit | No |
| Role | id(pk) | Unique identifier of a role | int | No |
| role\_name | Name of the role | varchar(10) | No |
| status | Active role checker | bit |  |
| Promotion | id | Unique identifier of a promotion | int | No |
| field\_type\_id | Id of field type that promotion belongs | datetime | No |
| field\_owner\_id | Id of field owner that promotion belongs | int | No |
| date\_from | Sent notification tracker | datetime | No |
| date\_to | Notification content | datetime | No |
| start\_time | Time to start apply promotion (in a day) | time | No |
| end\_time | Time to end apply promotion (in a day) | time | No |
| sale\_off | Sale off percent of promotion | float | No |
| free\_services | Free services content | varchar(1000) | Yes |
| status | Active promotion checker | bit | No |
| BlacklistOpponent | id | Unique identifier of a black list opponent | int | No |
| user\_id | Id of the user that the back list opponent belongs | int | No |
| opponent\_id | Id of the user that the black list opponent belongs | int | No |
| status | Active black list checker | bit | No |
| Voucher | id(PK) | Unique Identifier of License | int | No |
| voucher\_value | Name of License | float | No |
| bonus\_point\_target | Price of License | int | No |
| image\_url | Description of License | varchar(500) | No |
| status | Active checker of License | bit | No |
| VoucherRecord | id(PK) | Unique identifier of a voucher record | int | No |
| user\_id | Id of the user that the voucher record belongs | int | No |
| voucher\_id | Id of the voucher that the voucher record belongs | int | No |
| status | Active voucher checker | bit | No |
| Field | id(PK) | Unique identifier of a field | int | No |
| name | Name of a field | VARCHAR(45) | No |
| field\_owner\_id | Id of the field owner that the field belongs | INT | No |
| field\_type\_id | Id of the field type that the field belongs | INT | No |
| date\_from | The date when the field available | DATE | No |
| date\_to | The date when the field not available | DATE | Yes |
| status | Active field checker | bit | No |
| FieldType | id | Unique identifier of a field type | int | No |
| name | Name of a field type | VARCHAR(20) | No |
| numberOfPlayer | Number of player of the field type | Int | No |
| status | Active field type checker | bit | No |
| TimeEnable | id (PK) | Unique identifier of a time enable | INT | No |
| field\_type\_id | Id of the field type that the time enable belongs | int | No |
| field\_owner\_id | Id of the field owner that the time enable belongs | INT | No |
| date\_in\_week | Date in week that the time enable belongs | VARCHAR(10) | No |
| start\_time | Start time of the time enable | TIME | No |
| end\_time | End time of the time enable | TIME | No |
| price | Price (per hour) of the time enable | FLOAT | No |
| optimal | Optimal time enable checker | bit | No |
| effective\_date | Effective date of time enable | DATE | No |
| status | Active time enable checker | bit | No |
| TimeSlot | id (PK) | Unique identifier of a time slot | INT | No |
| field\_id | Id of the field that the time slot belongs | int | No |
| field\_owner\_id | Id of the field owner that the time slot belongs | INT | No |
| field\_type\_id | Id of the field type that the time slot belongs | INT | No |
| date | Date in week that the time slot belongs | DATE | No |
| start\_time | Start time of the time slot | TIME | No |
| end\_time | End time of the time slot | TIME | No |
| price | Price (per hour) of the time slot | FLOAT | Yes |
| reserve\_status | Active reserve checker | Bit | No |
| optimal | Is optimal checker | bit | No |
| status | Active time slot checker | bit | No |
| MatchingRequest | id | Unique identifier of the matching request | int | No |
| user\_id | Id of the user that the matching request belongs | int | No |
| field\_type\_id | Id of the field type that the matching request belongs | int | No |
| longitude | Longitude of the matching request | DOUBLE | No |
| latitude | Latitude of the matching request | DOUBLE | No |
| date | Date of the matching request | DATE | No |
| start\_time | Start time of the matching request | TIME | No |
| end\_time | End time of the matching request | TIME | No |
| duration | Duration of the matching request | INT | No |
| address | Address of the matching request | VARCHAR(500) | No |
| expected\_price | Expected price of matching request | FLOAT | No |
| expected\_distance | Expected distance of matching request | INT | No |
| priority\_field | Is high priority field | Bit | No |
| status | Active matching request checker | bit | No |
| TourMatch | id(PK) | Unique identifier of the tour match | Int | No |
| user\_id | Id of the user that the tour match belongs | INT | No |
| opponent\_id | Id of the opponent that the tour match belongs | int | No |
| time\_slot\_id | Id of the time slot that the tour match belongs | INT | No |
| complete\_status | Active complete checker | bit | No |
| winner\_id | Id of the winner that the tour match belongs | INT | No |
| status | Active tour match checker | bit | No |
| FriendlyMatch | id | Unique identifier of the friendly match | int | No |
| user\_id | Id of the user that friendly match belongs | int | No |
| time\_slot\_id | Id of the time slot that friendly match belongs | int | No |
| status | Active friendly match checker | bit | No |
| NormalMatch | id(PK) | Unique identifier of the normal match | int | No |
| time\_slot\_id | Id of the time slot that normal match belongs | int | No |
| field\_owner\_id | Id of the field owner that normal match belongs | int | No |
| player\_name | Player name of normal match | varchar(100) | No |
| phone | Phone of player of normal match | varchar(20) | No |
| status | Active normal match checker | bit | No |
| Bill | id(PK) | Unique identifier of a bill | int | No |
| user\_id | Id of the user that the bill belongs | int | No |
| field\_owner\_id | Id of the field owner that the bill belongs | int | No |
| friendly\_match\_id | Id of the friendly match that the bill belongs | int | Yes |
| tour\_match\_id | Id of the tour match that the bill belongs | int | Yes |
| date\_charge | Date charge the bill | datetime | No |
| price | Price of the bill | float | No |
| status | Active bill checker | bit |  |
| RattingOpponent | id(PK) | Unique identifier of a ratting opponent | int | No |
| user\_id | Id of the user that the ratting belongs | nvarchar(50) | No |
| tour\_match\_id | Id of the tour match that the ratting belongs | int | No |
| result | Result of the ratting opponent | int | No |
| rating\_level | Ratting level of the ratting opponent | int | No |
| goals\_difference | Goal difference checker | int | Yes |
| status | Active ratting checker | bit | No |
| RattingField | id(PK) | Unique identifier of a ratting field | int | No |
| user\_id | Id of the user that the ratting belongs | int | No |
| field\_owner\_id | Id of the tour match that the ratting belongs | int | No |
| rating\_score | Ratting score of the ratting field | int | No |
| comment | Commend of ratting | varchar(1000) | No |
| status | Active ratting checker | bit | No |
| Report | Id(PK) | Unique identifier of a report opponent | int | No |
| accuser\_id | Id of the accuser that the report belongs | int | No |
| accused\_id | Id of the accused that the report belongs | int | No |
| tour\_match\_id | Id of the tour match that the report belongs | int | Yes |
| reason | Reason of the report | varchar(1000) | No |
| status | Active report checker | bit | No |
| StandardPrice | id | Unique identifier of a standard price | int | No |
| staff\_id | Id of the staff that the standard price belongs | int | No |
| field\_type\_id | Id of the field type that the standard price belongs | int | No |
| rush\_hour | Rush hour checker | bit | No |
| max\_price | Max price of standard price | float | No |
| min\_price | Min price of standard price | float | No |
| status | Active standard price checker | bit | No |
| TeamMember | id(PK) | Unique identifier of a team member | int | No |
| player\_name | Name of team member | varchar(100) | No |
| captain\_id | Identifier of captain of team member | int | No |
| phone | Phone of team member | varchar(20) | No |
| longitude | Longitude of team member | double | No |
| latitude | Latitude of team member | double | No |
| address | Address of team member | varchar(500) | No |
| status | Unique identifier of a team member | bit | No |
| DepositHistory | id(PK) | Unique identifier of a team member | int | No |
| user\_id | Identifier of user of deposit history | int | No |
| balance | Balance of deposit history | float | No |
| information | Information of deposit history | varchar(500) | No |
| status | Unique identifier of a deposit history | bit | 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 5.0 | Android 7: 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 | top-10-field-owner | Search 10 fields near user |
| 2 | name | Search field by field’s name |
| 3 | friendly-match | Reserve a field |
| 4 | matching-request | Create new matching request |
| 5 | tour-match | Reserve a tour match |
| 6 | Manage-favorites | Add new favorite field |

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 | 12/03/2017 |
| 2 | 1.1 | 12/03/2017 |
| 3 | 0.7 | 12/03/2017 |
| 4 | 0.9 | 12/03/2017 |
| 5 | 1.3 | 12/03/2017 |
| 6 | 0.75 | 12/03/2017 |
| 7 | 0.8 | 12/03/2017 |
| 8 | 1.2 | 12/03/2017 |
| 9 | 1 | 12/03/2017 |
| 10 | 0.93 | 12/03/2017 |
| 11 | 0.89 | 12/03/2017 |
| 12 | 0.8 | 12/03/2017 |
| 13 | 0.83 | 12/03/2017 |
| 14 | 0.93 | 12/03/2017 |
| 15 | 1.1 | 12/03/2017 |
| 16 | 1 | 12/03/2017 |
| 17 | 0.9 | 12/03/2017 |
| 18 | 0.85 | 12/03/2017 |
| 19 | 0.83 | 12/03/2017 |
| 20 | 0.93 | 12/03/2017 |
|  | **Average: 0.93(seconds)** |  |

Table 67: Mobile API load speed test result

## Test plan

The overall purpose of testing is to ensure FFRS 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 FFRS. 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

* User: Search field, Reserve match, create matching request, reserve tour match, set favorite field
* Field owner: create active time, disable field, cancel match

### Features not to be tested

* Guest: Sign in, sign out, edit profile
* User: Edit profile, cancel matching request, cancel reservation request, add opponent to blacklist, deposit
* Field Owner: create field
* Staff: View report, request lock account, accept the field owner’s account creation request, manage price of field

## System testing test case

### Communication diagram

Test cases are created from below communication diagram

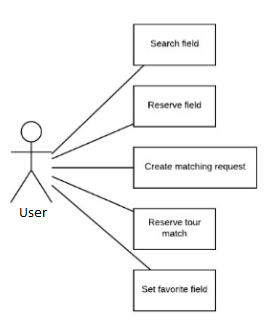


Figure 52: Mobile application communication diagram

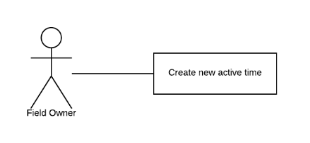


Figure 53: Web application communication

### Test cases

#### <User>Search Field

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| ID | Test Case Description | Precondition | Test case procedure | Expected output | Inter-test case dependence | Result | Test Date |
| SEARCH 1 | Test user search a field not in database | N/A | 1.User input: “zz”  2.User send search command | 1.System shows “Không tìm thấy sân” | N/A | Pass | 12/02/17 |
| SEARCH 2 | Test user input a blank in search box | There is at least one record in database for translate online function | 1.User input blank  2.User send search command | 2.System shows result as a list of 10 field near user | N/A | Pass | 12/02/17 |
| SEARCH 3 | Test user input a field’s name | Field owner named “Cây trâm” is already exist in database. | 1.User inputs: “cay ” | 1.System shows field owner “Cây trâm” on the screen. | N/A | Pass | 12/02/17 |

Table 68: Test case <User> Search Field

#### <User> Reserve field

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| ID | Test Case Description | Precondition | Test case procedure | Expected output | Inter-test case dependence | Result | Test Date |
| ReserveF 1 | Test User reserves a field | N/A | 1.User sets time frame  2.User sends reserve command | 1.System shows “Đặt sân thành công” | N/A | Pass | 12/02/17 |
| ReserveF 2 | Test User reserves a field but not enough money | N/A | 1.User sets time frame  2.User sends reserve command | 1.System shows “Tài khoản không đủ tiền”  2.System displays deposit screen | N/A | Pass | 12/02/17 |
| ReserveF  3 | Test User reserves in the busy time frame | There is at least 1 field has free time in database | 1.User 1 and User 2 set the same time frame  2.User 1 sends reserve command  3.User 2 sends reserve command after User 1 | 1.System shows message “Đặt sân thành công” to the User 1  2.System shows mesage “Sân không còn thời gian trống” to other User 2 | N/A | Pass | 12/02/17 |

Table 69: Test case <User> Reserve field

#### <User> Create matching request

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| ID | Test Case Description | Precondition | Test case procedure | Expected output | Inter-test case dependence | Result | Test Date |
| MatchingR 1 | Test User create matching request is working ok | N/A | 1.User inputs required field  2.User send creates matching request | 1.System shows result: “Đã khởi tạo yêu cầu tìm kiếm” | N/A | Pass | 12/02/16 |
| MatchingR 2 | Test User create matching request without money in account balance | N/A | 1.User inputs required field  2.User send creates matching request | 1.System shows result: “Không đủ tiền”.  2.System displays deposit screen | N/A | Pass | 12/02/16 |

Table 70: Test case <User> Create matching request

#### <User> Reserve tour match

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| ID | Test Case Description | Precondition | Test case procedure | Expected output | Inter-test case dependence | Result | Test Date |
| ResM 1 | Test user reserve tour match is working fine | Matching request: 15/02/2017, start at 8AM end at 10AM, duration 2 hours, location at “Công viên phần mềm quang trung Q12”, 5km area, rank 2000 was created in database.  User’s rank is 2000 | 1.User inputs “15/02/2017, start at 6AM end at 12AM, duration 2 hours, location at “Chợ cầu Gò vấp”, 5km  2.User sends find matching request command | 1.System shows matching request in database | N/A | Pass | 12/02/17 |
| ResM 2 | Test user reserve tour match with not in range of the matching request in database | Matching request: 15/02/2017, start at 8AM end at 10AM, duration 2 hours, location at “Công viên phần mềm quang trung Q12”, 5km area, rank 2000 was created in database.  User’s rank is 2000 | 1.User inputs “15/02/2017, start at 6AM end at 12AM, duration 2 hours, location at “Chợ bến thành”, 5km  2.User sends find matching request command | 1. System shows message “Không tìm thấy đối thủ phù hợp” | N/A | Pass | 12/02/17 |
| ResM 3 | Test user reserve tour match with not in time frame of the matching request in database | Matching request: 15/02/2017, start at 8AM end at 10AM, duration 2 hours, location at “Công viên phần mềm quang trung Q12”, 5km area, rank 2000 was created in database.  User’s rank is 2000 | 1.User inputs “15/02/2017, start at 16PM end at 21AM, duration 2 hours, location at “Công viên phần mềm quang trung Q12”, 5km  2.User sends find matching request command | 1. System shows message “Không tìm thấy đối thủ phù hợp” | N/A | Pass | 12/02/17 |
| ResM 4 | Test user reserve tour match with not in time frame of the matching request in database | Matching request: 15/02/2017, start at 8AM end at 14PM, duration 2 hours, location at “Công viên phần mềm quang trung Q12”, 5km area, rank 2000 was created in database.  User’s rank is 2000 | 1.User inputs “15/02/2017, start at 8AM end at 14PM, duration 1.5 hours, location at “Công viên phần mềm quang trung Q12”, 5km  2.User sends find matching request command | 1. System shows message “Không tìm thấy đối thủ phù hợp” | N/A | Pass | 12/02/17 |
| ResM 5 | Test user reserve tour match with not the same day as the matching request in database | Matching request: 15/02/2017, start at 8AM end at 10AM, duration 2 hours, location at “Công viên phần mềm quang trung Q12”, 5km area, rank 2000 was created in database.  User’s rank is 2000 | 1.User inputs 16/02/2017, start at 8AM end at 10AM, location at “Công viên phần mềm quang trung Q12”, 5km  2.User sends find matching request command | 1. System shows message “Không tìm thấy đối thủ phù hợp” | N/A | Pass | 12/02/17 |
| ResM 6 | Test user reserve tour match with a higher level rank matching request in database | Matching request: 15/02/2017, start at 8AM end at 10AM, duration 2 hours, location at “Công viên phần mềm quang trung Q12”, 5km area, rank 3000 was created in database.  User’s rank is 2000 | 1.User inputs 16/02/2017, start at 8AM end at 10AM, location at “Công viên phần mềm quang trung Q12”, 5km  2.User sends find matching request command | 1. System shows message “Không tìm thấy đối thủ phù hợp” | N/A | Pass | 12/02/17 |
| ResM 7 | Test user reserve tour match. User was in other user blacklist. | Matching request: 15/02/2017, start at 8AM end at 10AM, duration 2 hours, location at “Công viên phần mềm quang trung Q12”, 5km area, rank 2000  was created in database.  User’s rank is 2000.  User in the blacklist of whom created matching request | 1.User inputs 16/02/2017, start at 8AM end at 10AM, location at “Công viên phần mềm quang trung Q12”, 5km  2.User sends find matching request command | 1. System shows message “Không tìm thấy đối thủ phù hợp” | N/A | Pass | 12/02/17 |
| ResM 8 | Test User find matching request without money in account balance | N/A | 1.User inputs 16/02/2017, start at 8AM end at 10AM, location at “Công viên phần mềm quang trung Q12”, 5km  2.User sends find matching request command | 1.System shows result: “Không đủ tiền”.  2.System displays deposit screen | N/A | Pass | 12/02/16 |
| ResM 9 | Test user reserve tour match, but the request was accepted by other user (user later than other user in accepting request) | Matching request: 15/02/2017, start at 8AM end at 10AM, duration 2 hours, location at “Công viên phần mềm quang trung Q12”, 5km area, rank 2000 was created in database.  User’s rank is 2000 | 1.User inputs “15/02/2017, start at 6AM end at 12AM, duration 2 hours, location at “Chợ cầu Gò vấp”, 5km  2.User sends find command  2.User chooses field, then sends reserve match command | 1.System shows message “Yêu cầu tìm kiếm không còn tồn tại” | N/A | Pass | 12/02/17 |

Table 71: Test case <User> Reserve tour match

#### <User> Set favorite field

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| ID | Test Case Description | Precondition | Test case procedure | Expected output | Inter-test case dependence | Result | Test Date |
| FavField 1 | Test set favorite field is working | User played at “Cây Trâm” field before. | 1.User chooses Cây Trâm field  2.Set favorite field | 1.System add field to user’s favorite field list | N/A | Pass | 12/02/17 |
| FavField 2 | Test set favorite field affects on reserve field | “Cây Trâm” field is in favorite field list.  User near “Cây Trâm” field. | 1.User search field | 1.System shows list of field near user.  “Cây Trâm” displays at the first as a high priority field. | N/A | Pass | 12/02/17 |
| FavField 3 | Test set favorite field affects on reserve tour match | “Cây Trâm” field is in favorite field list.  User near “Cây Trâm” field.  User who created matching request is near “Cây Trâm” field. | 1.User sends reserve tour match commands | 1.After chooses matching request, “Cây Trâm” field displayed at the first as a high priority field. | N/A | Pass | 12/02/17 |

Table 72: Test case <User> Set favorite field

#### <Field Owner> Create new active time

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| ID | Test Case Description | Precondition | Test case procedure | Expected output | Inter-test case dependence | Result | Test Date |
| ActiT 1 | Test create active time is working ok | Field Owner has at least 1 field | 1.Field Owner inputs required information.  2.User sends create new active time. | 1.System shows successfully message. | N/A | Pass | 12/02/17 |
| ActiT 2 | Test create active time with a field has time frame reserved | Field Owner has at least 1 field.  User reserve a field at the next day. | 1.User sends create new active time | 1.System shows “Đã có người đặt sân của bạn” | N/A | Pass | 12/02/17 |

Table 73: Test case <Premium User> Train Offline

### 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

### sSetting 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

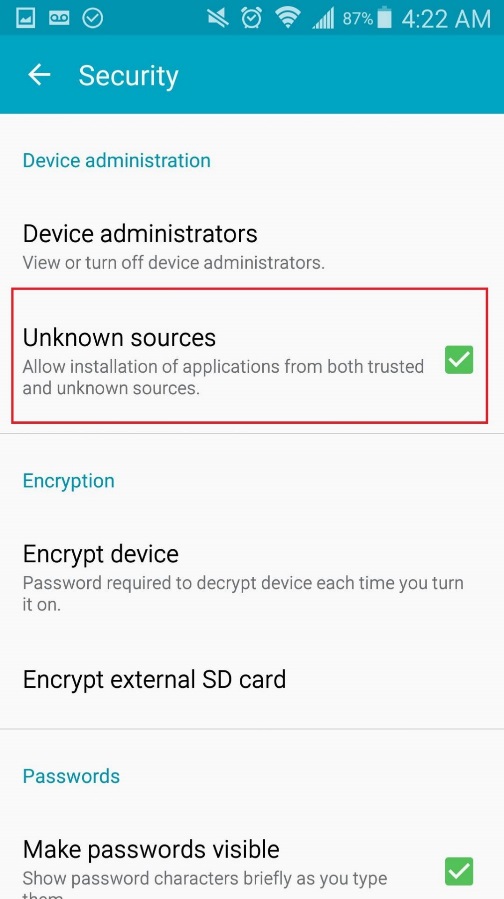
Using file **war** built from previous step.

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

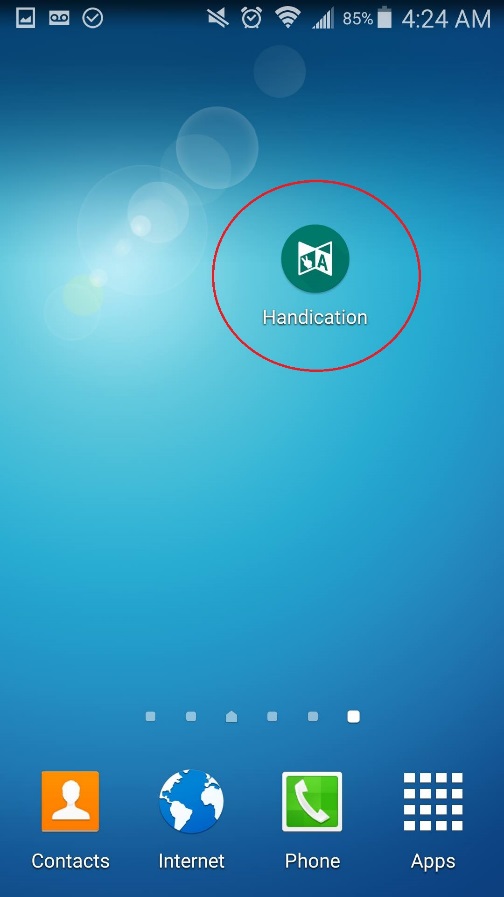
Web application is now available at [**http://server-ip-address**](http://server-ip-address).

### 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

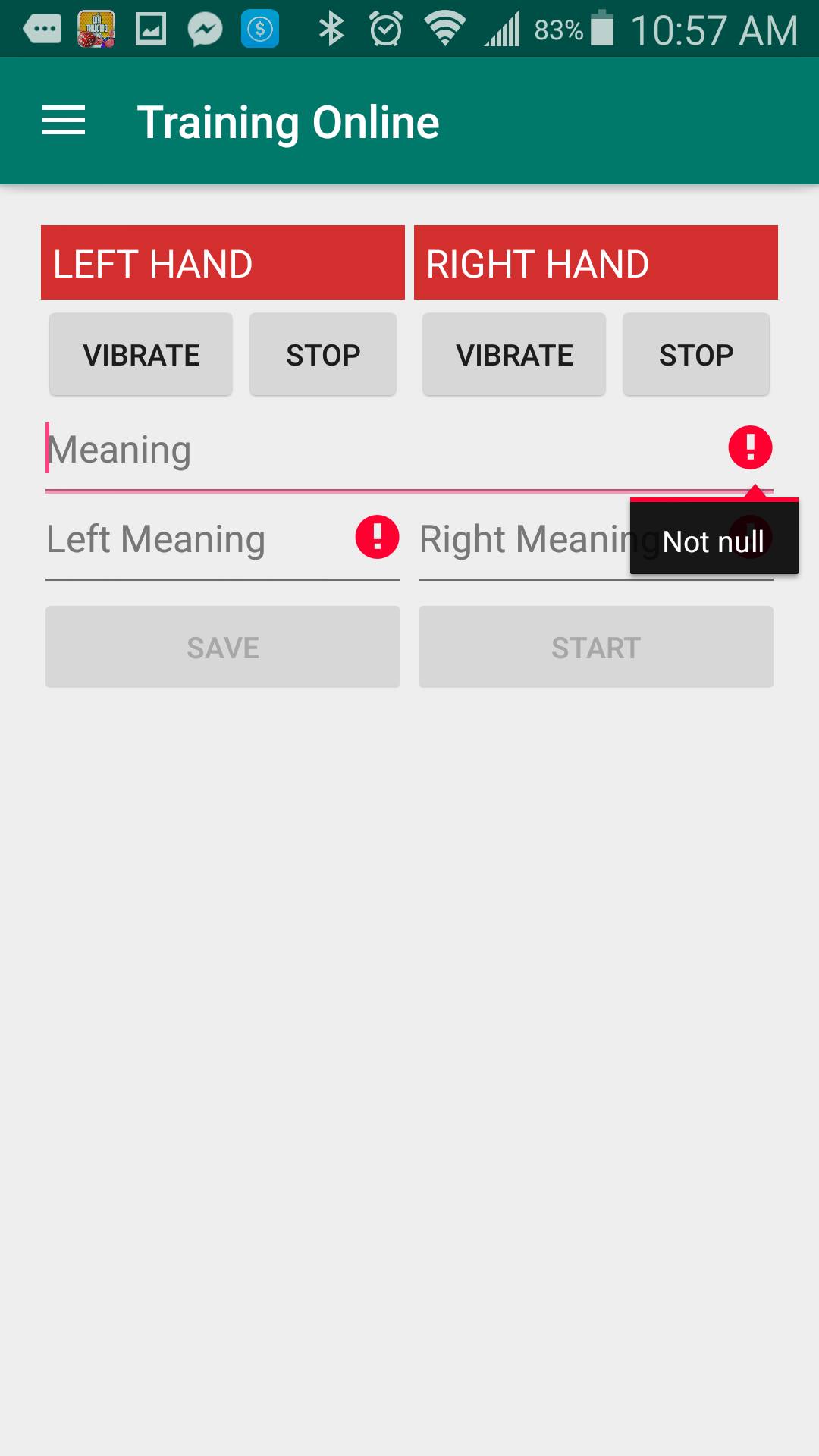
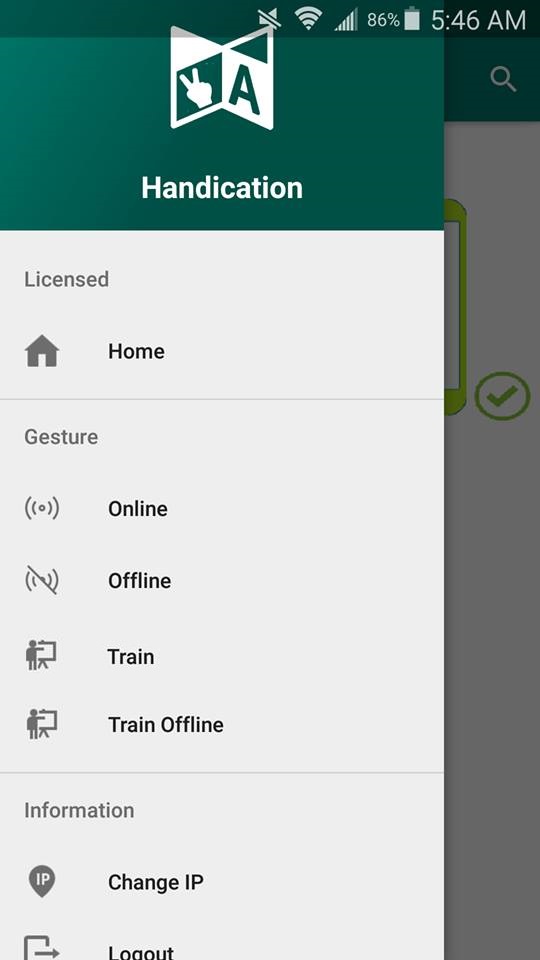
The armband is work best for about first 10 minutes, after that. Users should let their arms rest for 1 minute before the next use for better performance.

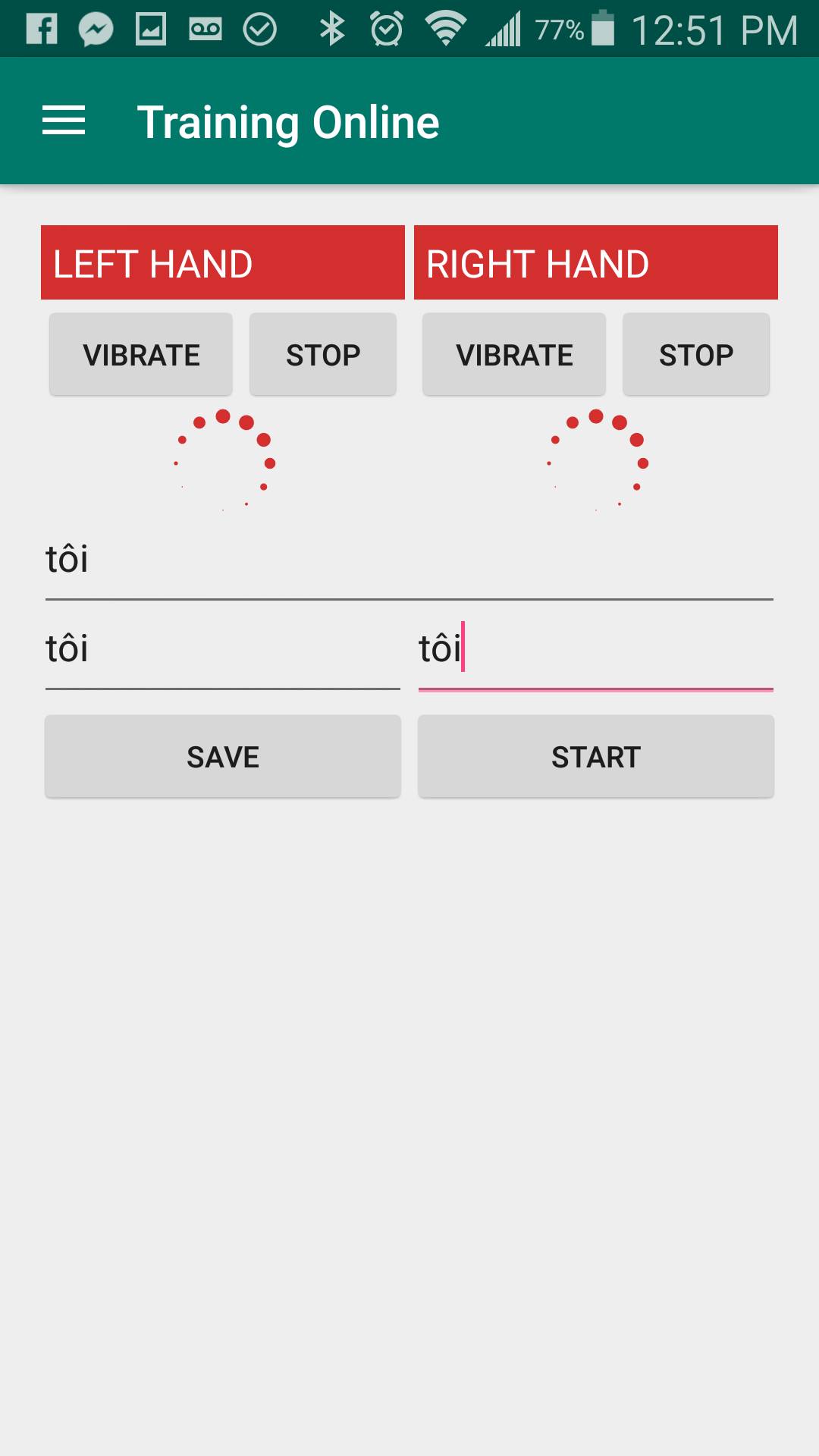
### Mobile application

#### Train online

Precondition:

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





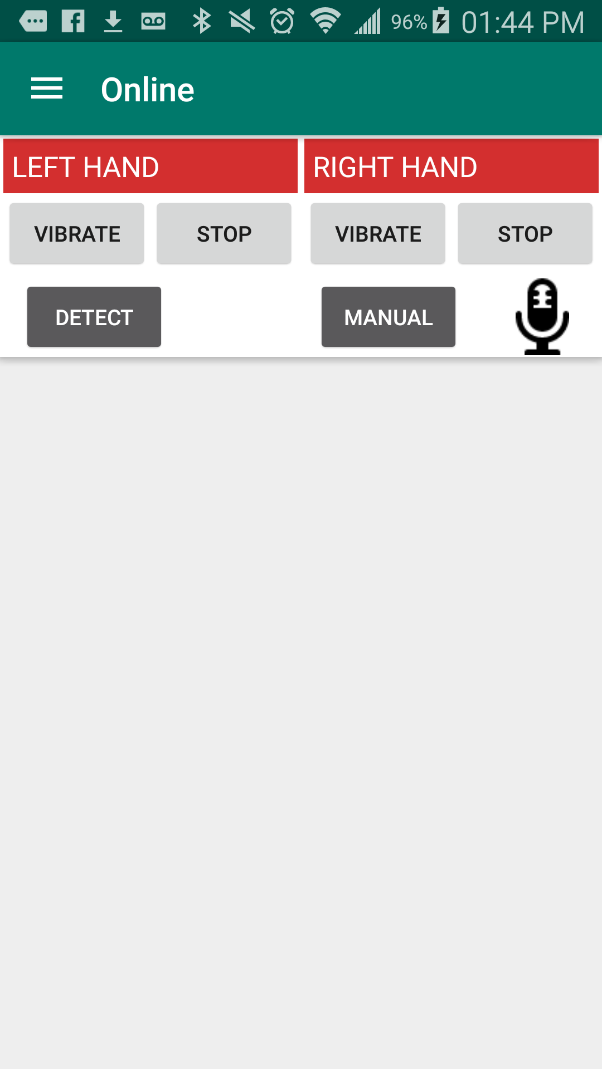
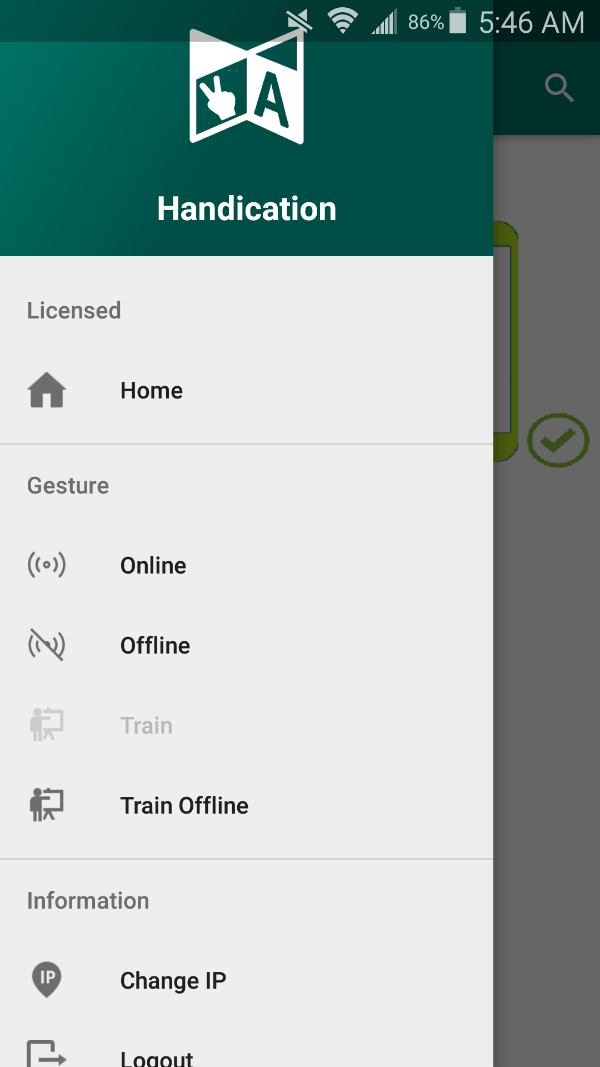
|  |  |
| --- | --- |
| Step | Description |
| 1 | Choose “Train” at navigation panel |
| 2 | Input these fields:  “Meaning”: Tôi (for example)  “Left Meaning”: Tôi (for example)  “Right Meaning”: Tôi (for example) |
| 3 | Press “Start” button |
| 4 | Perform the sign language |
| 5 | Press “Save” Button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

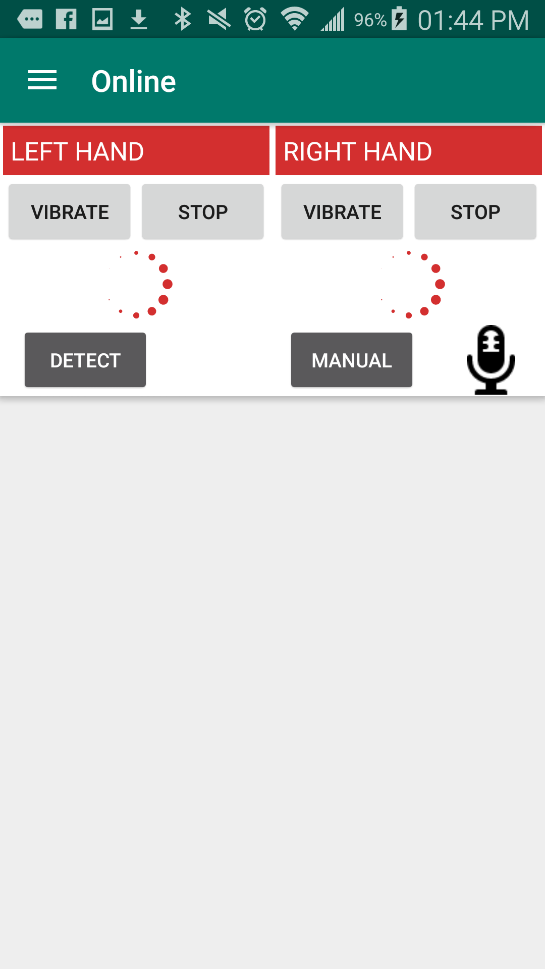
#### Translate online

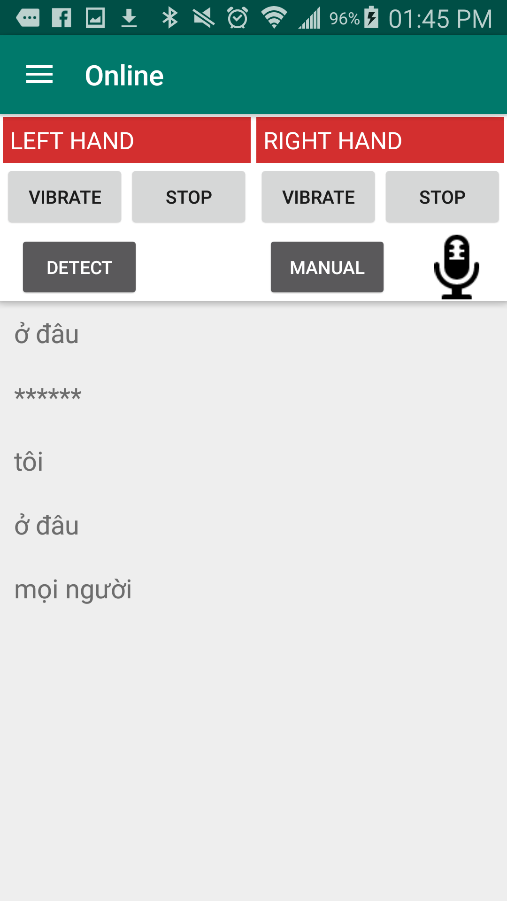
##### Automatic

Precondition:

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





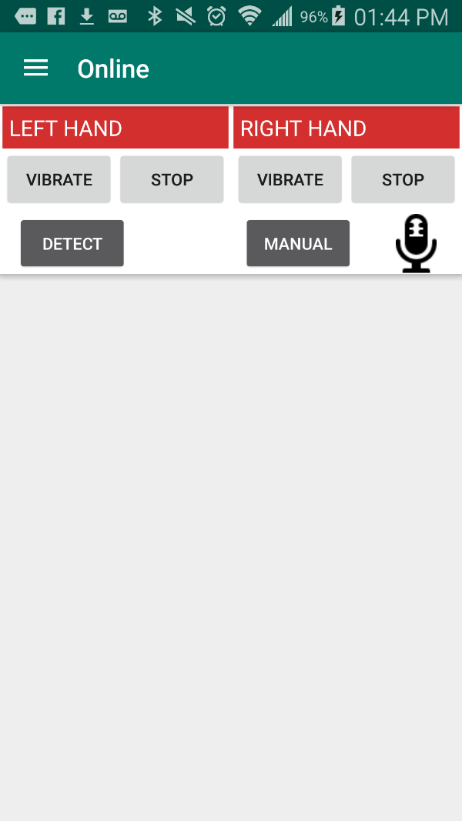
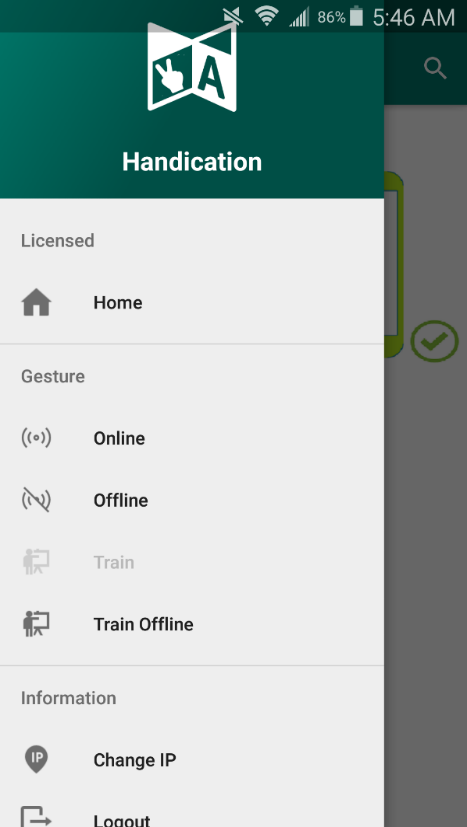


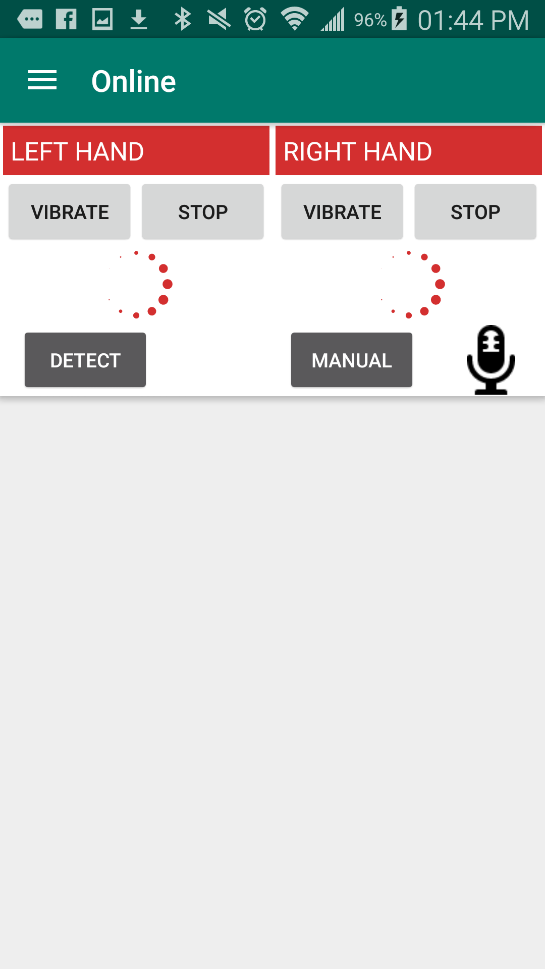
|  |  |
| --- | --- |
| Step | Description |
| 1 | Choose “Online” at navigation panel |
| 2 | Press “Detect” button |
| 3 | Perform the sign language |
| 4 | Perform the rest sign |
| 5 | Check the output information on the output screen |
| 6 | Press  for play sound (optional) |

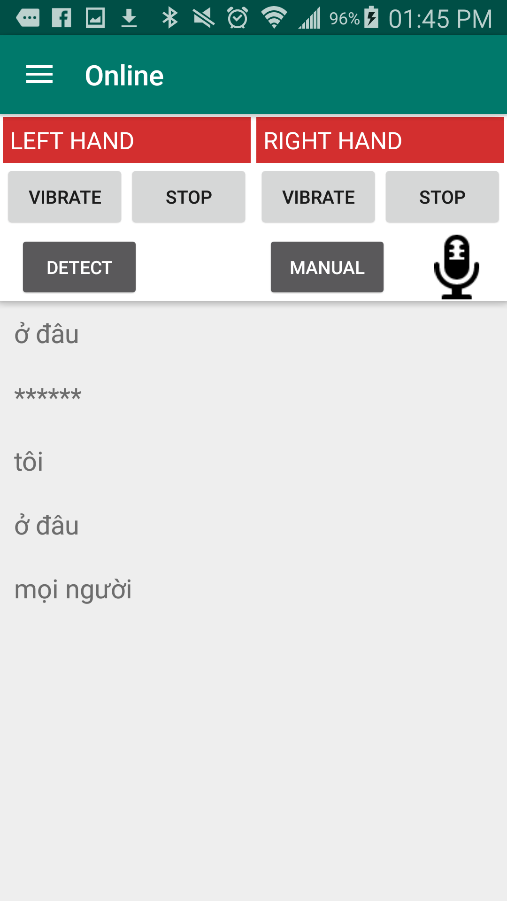
##### Manual

Precondition:

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





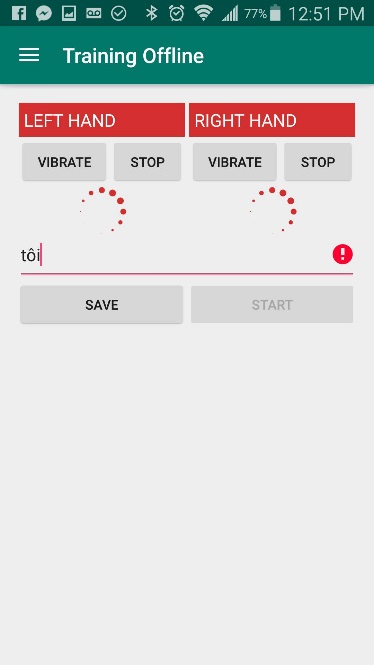


|  |  |
| --- | --- |
| Step | Description |
| 1 | Choose “Online” at navigation panel |
| 2 | Press “Manual” button |
| 3 | Perform the sign language |
| 4 | Press the “Manual” button |
| 5 | Perform the rest sign |
| 6 | Check the output information on the output screen |
| 7 | Press  for play sound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

#### Train offline

Precondition:

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





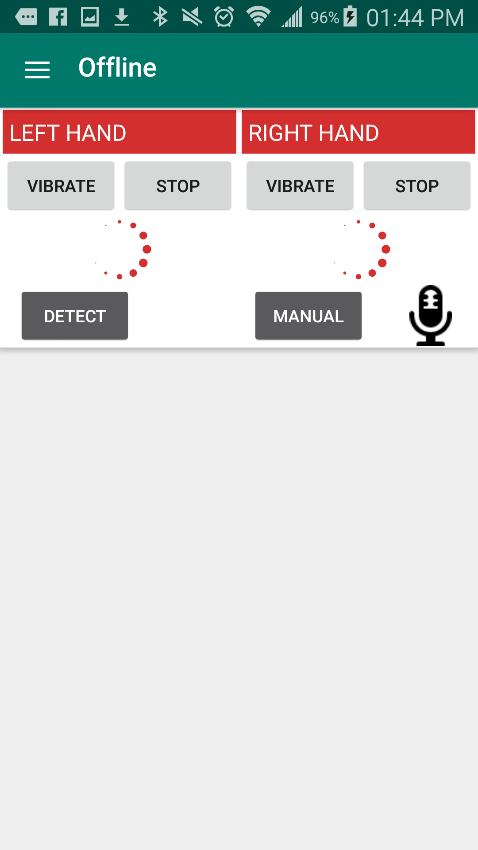
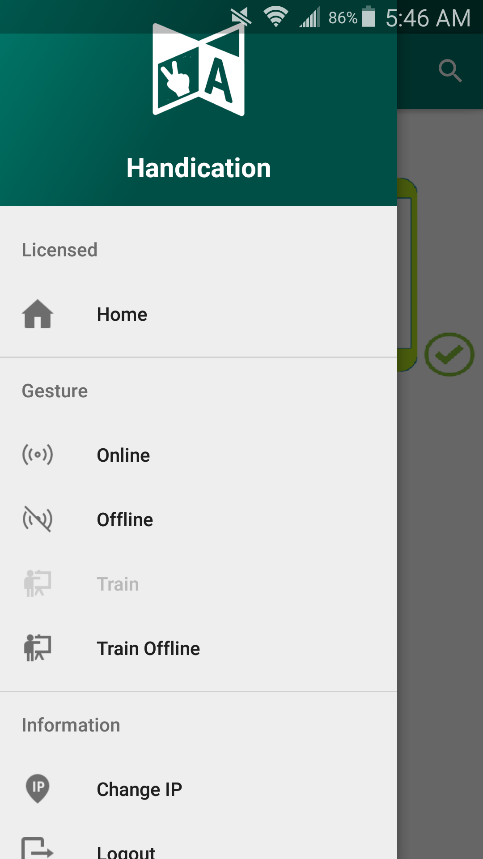
|  |  |
| --- | --- |
| Step | Description |
| 1 | Choose “Train offline” at navigation panel |
| 2 | Input these fields:  “Meaning”: Tôi (for example) |
| 3 | Press “Start” button |
| 4 | Perform the sign language |
| 5 | Press “Save” Button |

#### Translate offline

##### Automatic

Precondition:

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



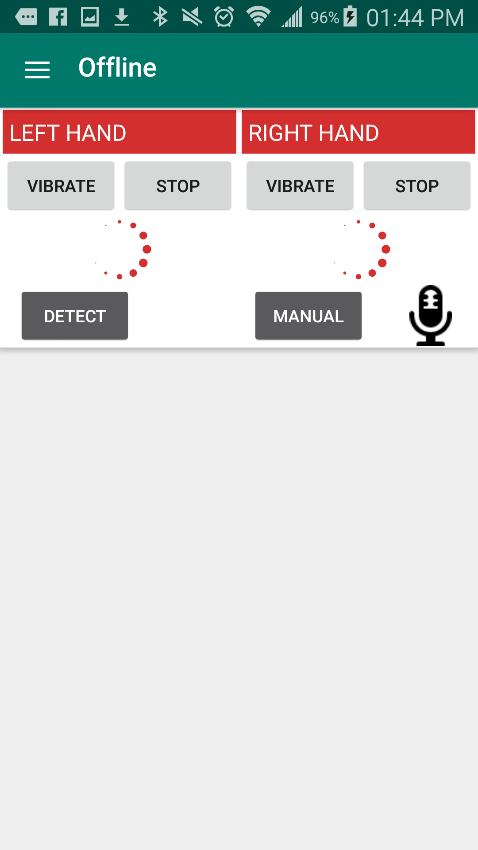
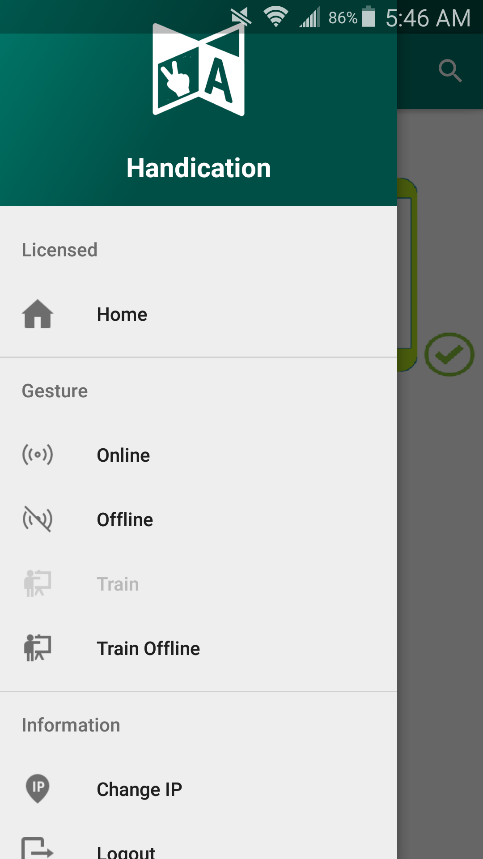


|  |  |
| --- | --- |
| Step | Description |
| 1 | Choose “Offline” at navigation panel |
| 2 | Press “Detect” button |
| 3 | Perform the sign language |
| 4 | Perform the rest sign |
| 5 | Check the output information on the output screen |
| 6 | Press  for play sound (optional) |

##### Manual

Precondition:

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

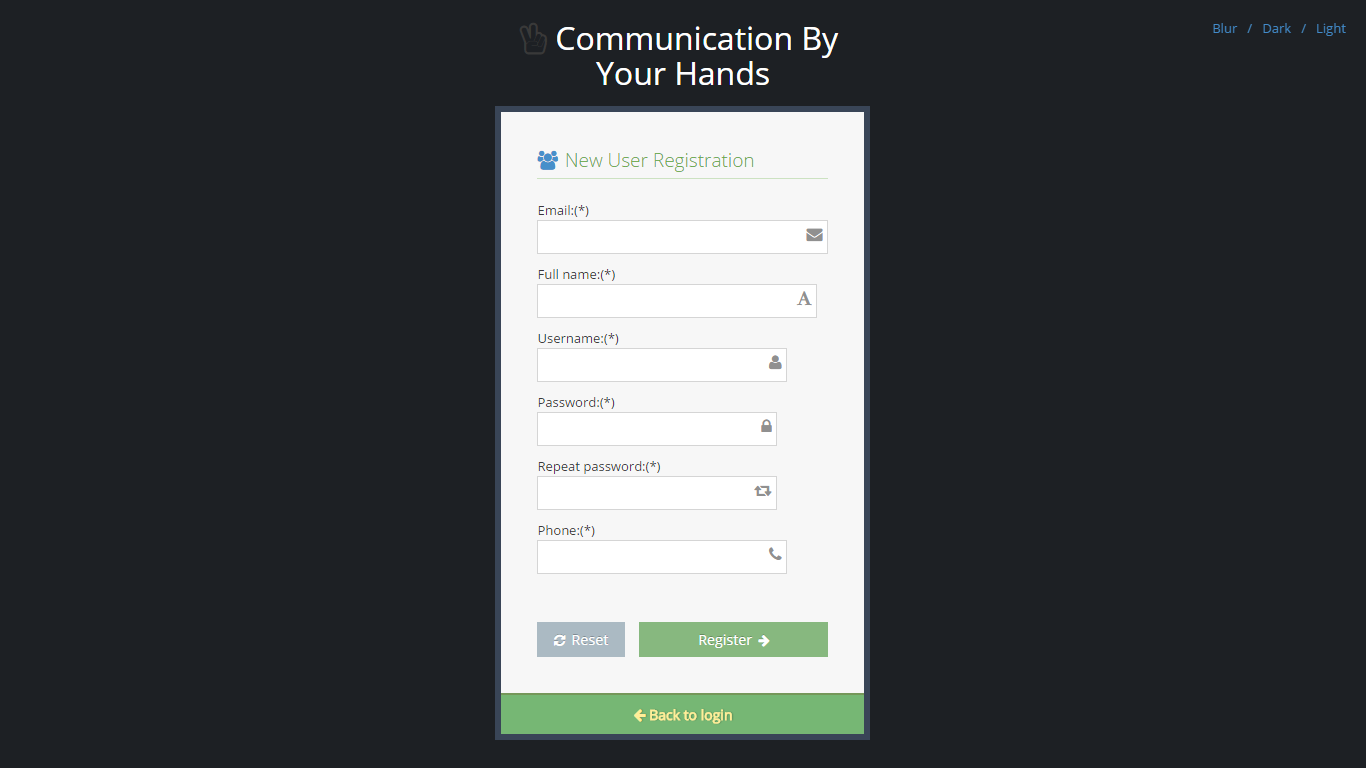




|  |  |
| --- | --- |
| Step | Description |
| 1 | Choose “Offline” at navigation panel |
| 2 | Press “Manual” button |
| 3 | Perform the sign language |
| 4 | Perform the rest sign |
| 5 | Check the output information on the output screen |
| 6 | Press  for play sound (optional) |

### Web application

#### Register

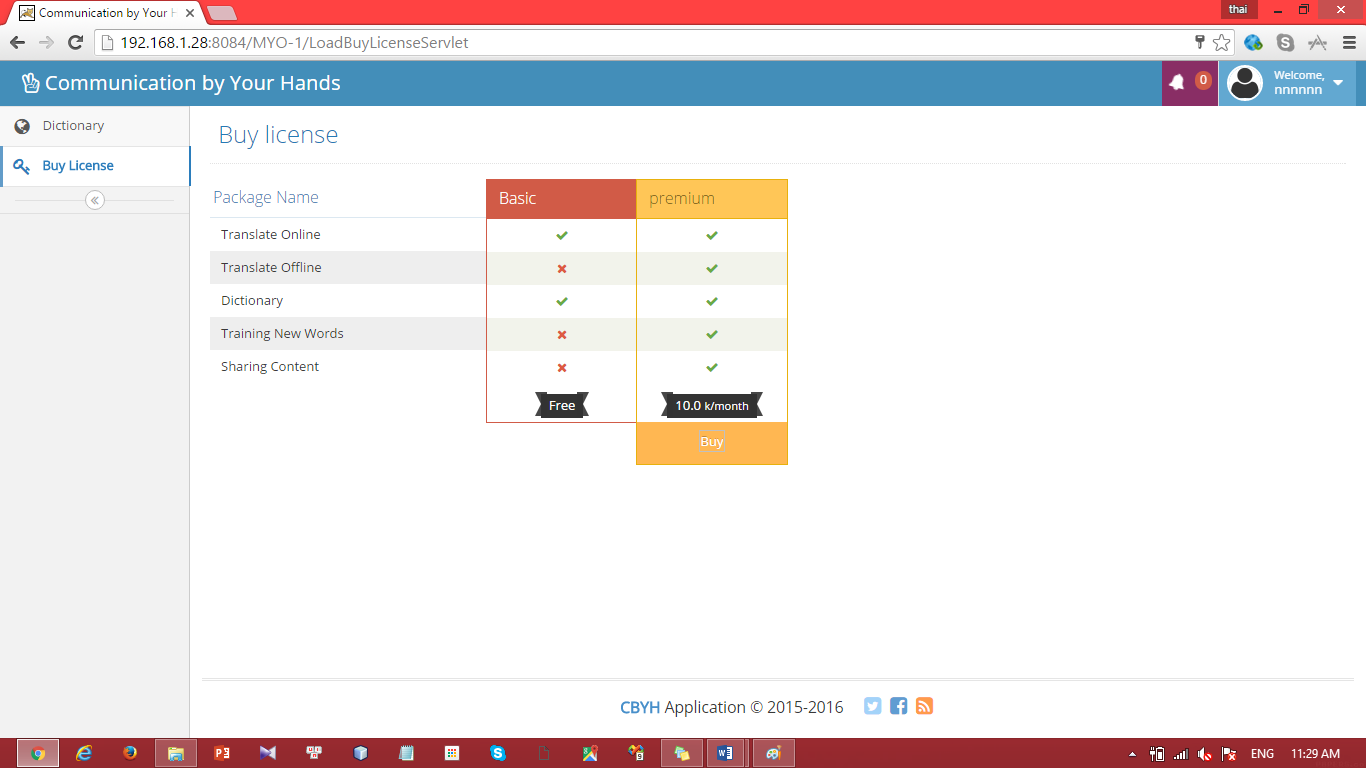


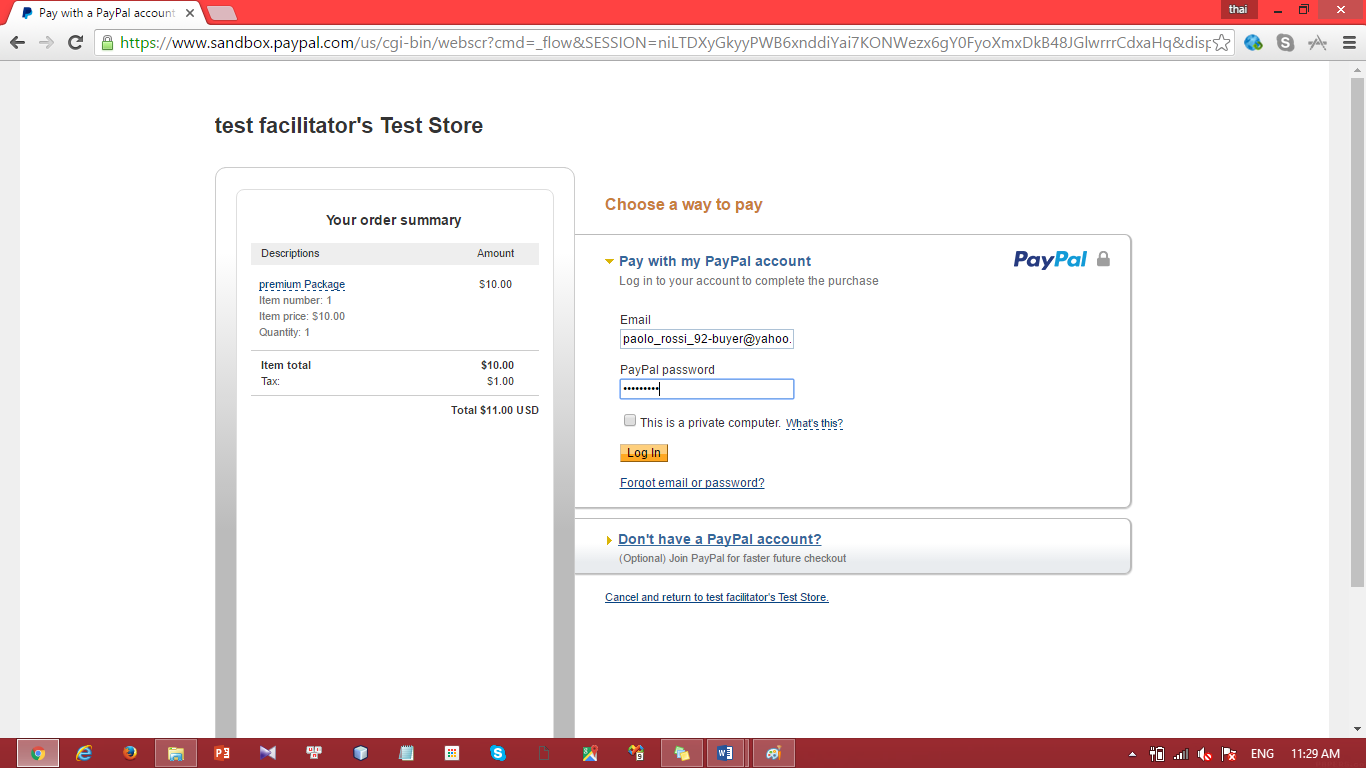
|  |  |
| --- | --- |
| Step | Description |
| 1 | Fill in these fields:  “Email” : [ac@xyz.vn](mailto:ac@xyz.vn) (for example)  “Fullname” : nnnnnn (for example)  “Password” : 123456 (for example)  “Confirm password” : 123456 (for example)  “Phone” : 0908123451 (for example) |
| 2 | Press “Register”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

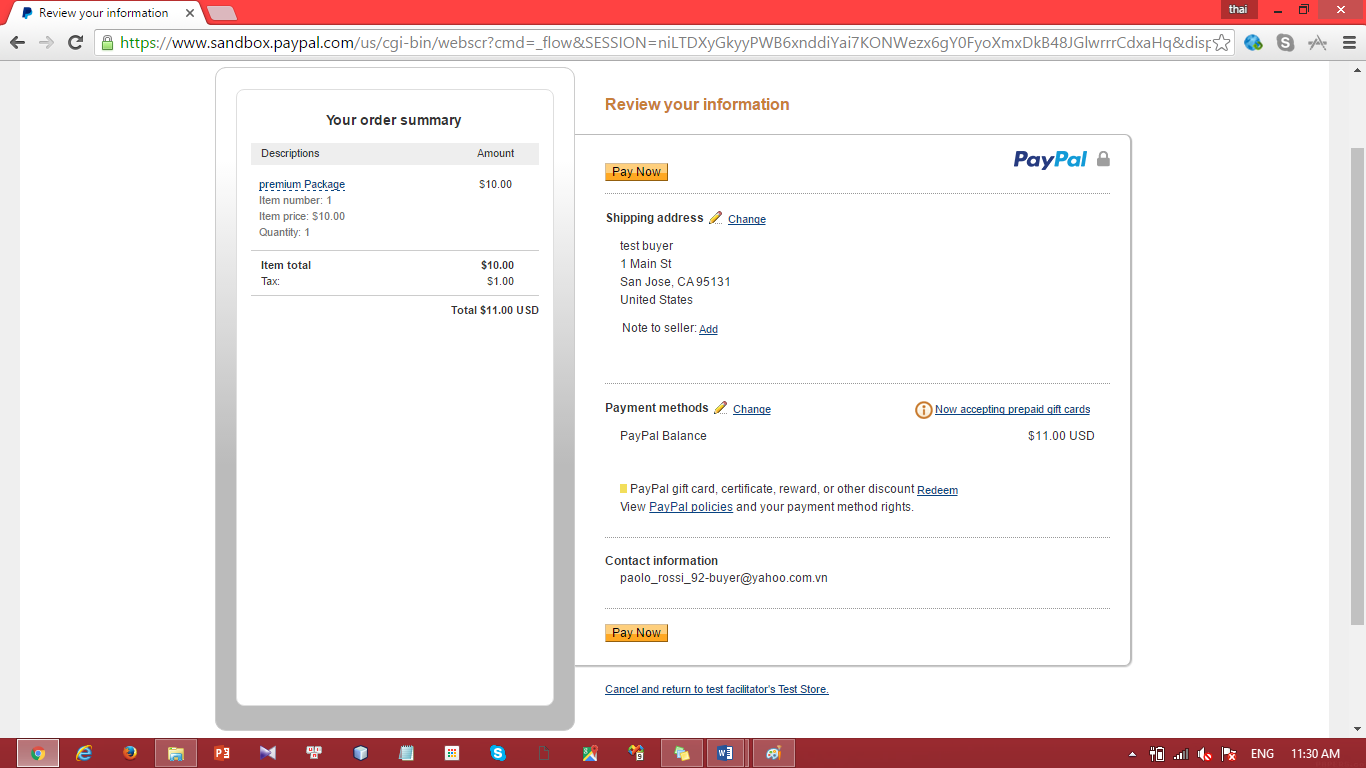
#### Buy license

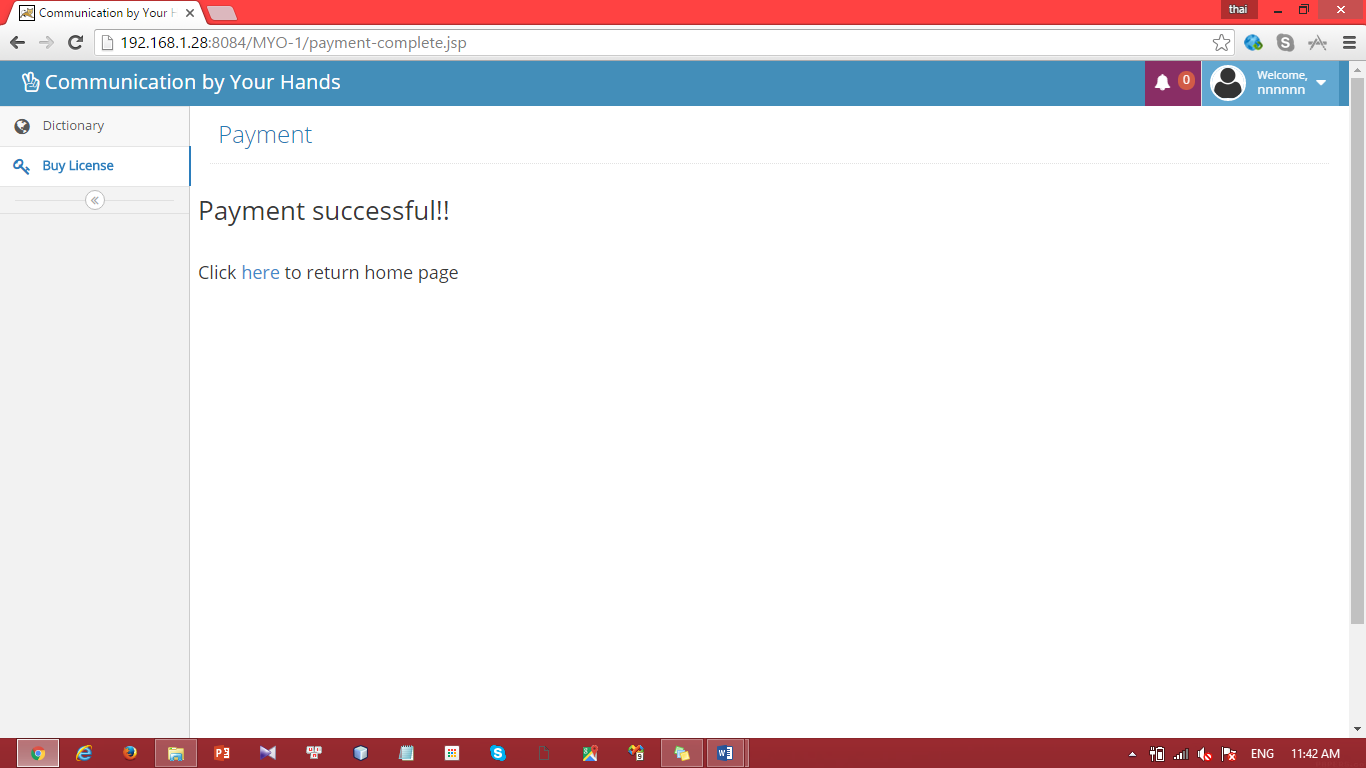
Precondition:

* User already had an Paypal account



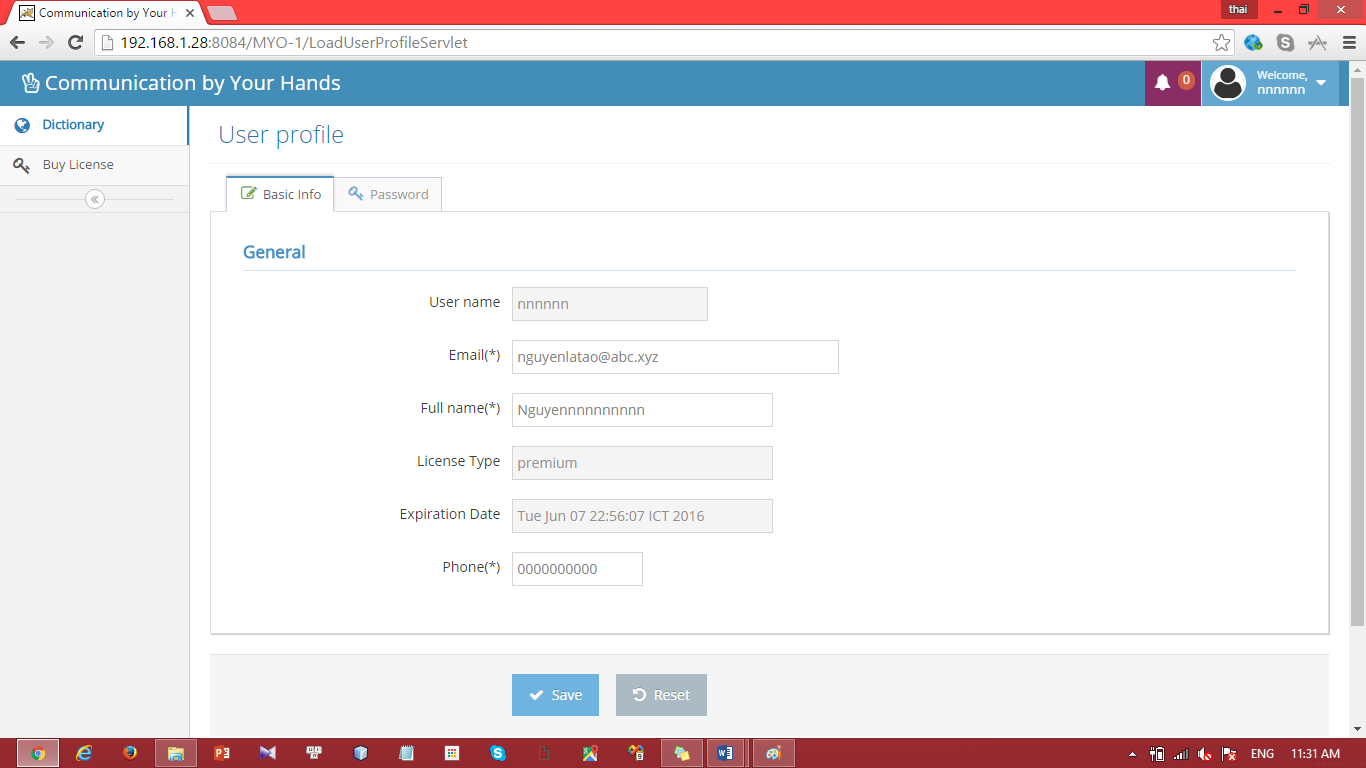


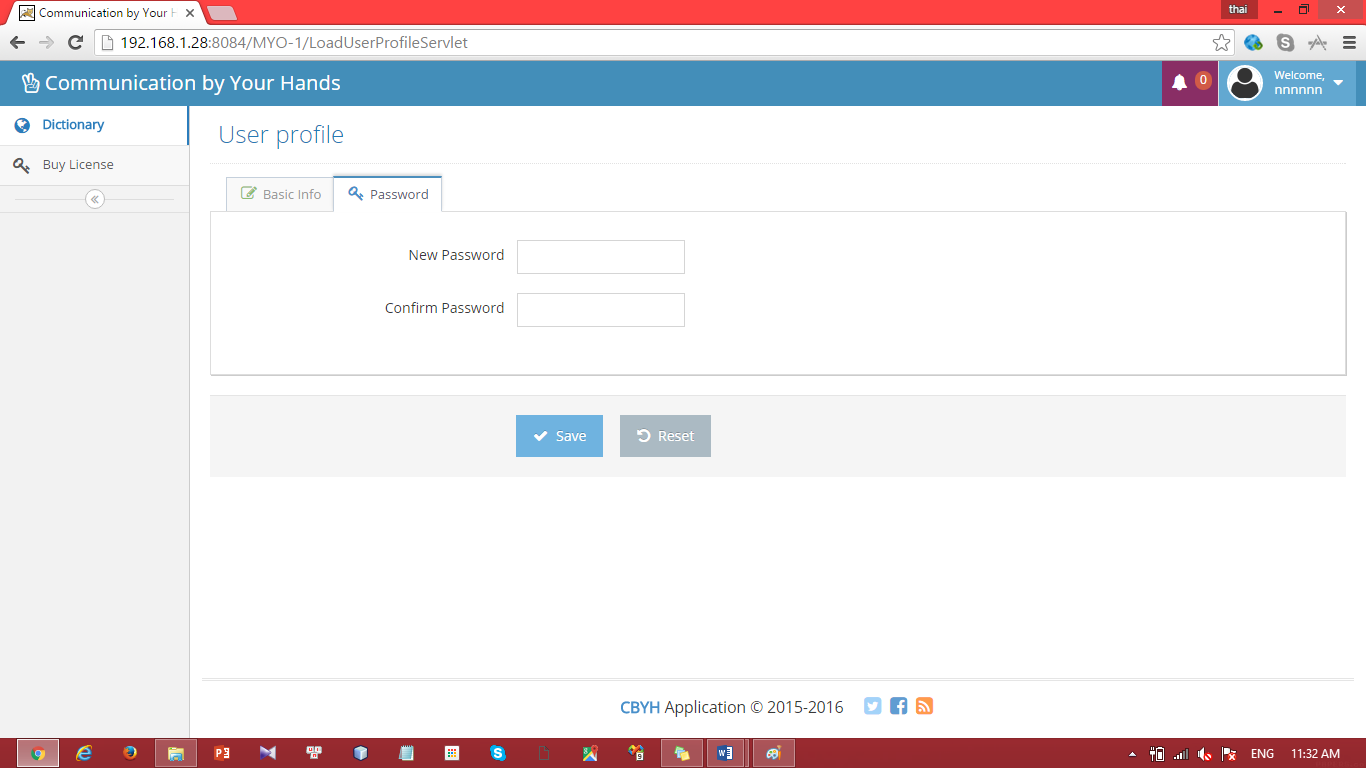




|  |  |
| --- | --- |
| Step | Description |
| 1 | Press “Buy” button |
| 2 | Fill these fields with Paypal account |
| 3 | Press “Buy now” button |
| 4 | Check result information |

#### 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