# Report No.4 Software Design Description

## Design Overview

<Kiến trúc hệ thống mà nhóm xây dựng: sử dụng các pattern và reference đến nội dung và xem xét lựa chọn các diagram mang đầy đủ nội dung như concept, không sao chép, vay mượn và chế kí hiệu. Nếu dùng kí hiệu ngoài UML thì ghi chú giải kí hiệu ngay cạnh hình vẽ.> <Mô tả kiến trúc của từng thành phần trong ứng dụng nếu có.>  
*Ví dụ*  
***Figure 9 System architecture design***  
*This diagram is referenced and modified from an original concept from: Chapter 6*  
*Architecture Design, SOFTWARE ENGINEERING 9th Edition, by Ian Sommerville.*

## System Architectural Design

Figure 22: System architecture design

### Embedded application architecture description

<Giải thích lý do tại sao lựa chọn mô hình này dựa trên SRS, Introduction, và project plan đã nêu ra ở các phần trên> <Mô tả các thành phần của kiến trúc theo dạng bảng, và sự tương tác giữa các thành phần theo kiến trúc.>

### Window application architecture description

<Giải thích lý do tại sao lựa chọn mô hình này dựa trên SRS, Introduction, và project plan đã nêu ra ở các phần trên> <Mô tả các thành phần của kiến trúc theo dạng bảng, và sự tương tác giữa các thành phần theo kiến trúc.>

## Component Diagram

<Thể hiện việc chia hệ thống thành các component. Nội dung này dựa trên kiến trúc đã đề ra ở phần trên để chia cho phù hợp và đúng mô hình>  
Ghi chú: Xem lại bộ quy ước kí hiệu của UML 2.0 trước khi vẽ các mối quan h cũng như hiểu rõ thiết kế để vẽ chính xác. Nếu tool không phù hợp thì nhóm nên dùng Paint để vẽ <Mô tả từng thành phần trong hình vẽ theo bảng biểu bên dưới.>

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

Table 26: Component Dictionary

## Detail Description

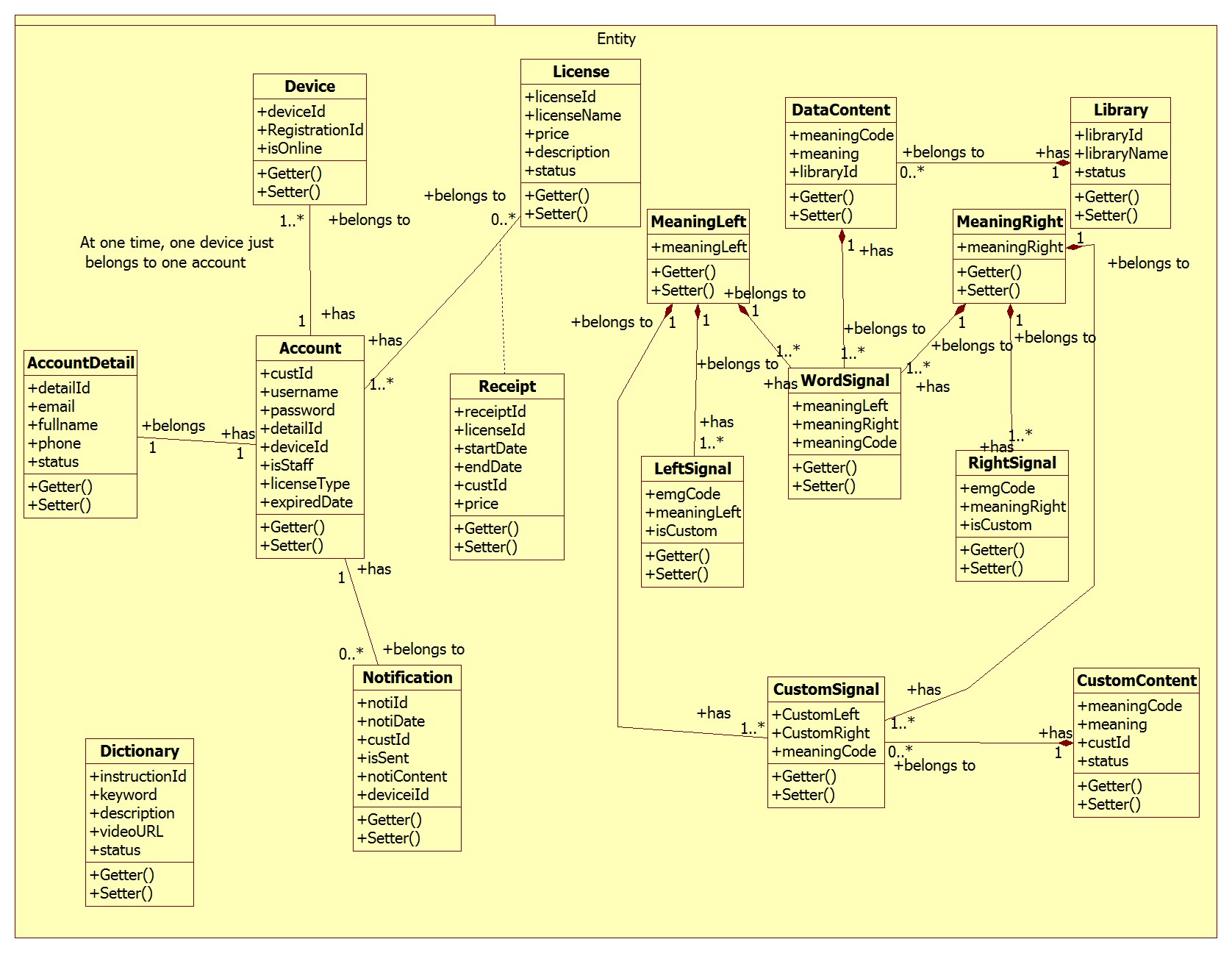
### Hardware

#### Schematic

### Software

#### Class Diagram

<Hình thiết kế class diagram: tham khảo các mối quan hệ giữa các lớp trong đặc tả UML, nắm rõ về dependency, association, composition, aggregation, inheritance. Bên cạnh đó, cần xác định rõ cardinality giữa các quan hệ với nhau. Đây là dạng conceptual class diagram, do vậy, cần căn cứ trên conceptual diagram và nội dung xây dựng object cần thiết khi lập trình và xây dựng ứng dụng trong lúc viết chương trình> <Mô tả từng thành phần class theo bảng biểu bên dưới.>

Figure 25: Class Diagram

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

Table 27: Class dictionary

#### Class Diagram Explanation

##### Account

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

Table 28: Account Attributes

##### AccountDetail

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

Table 29: AccountDetail Attributes

….

#### Interactive Diagram

##### Web Application

###### <User>

Buy license

Summary: this diagram show process of buy license

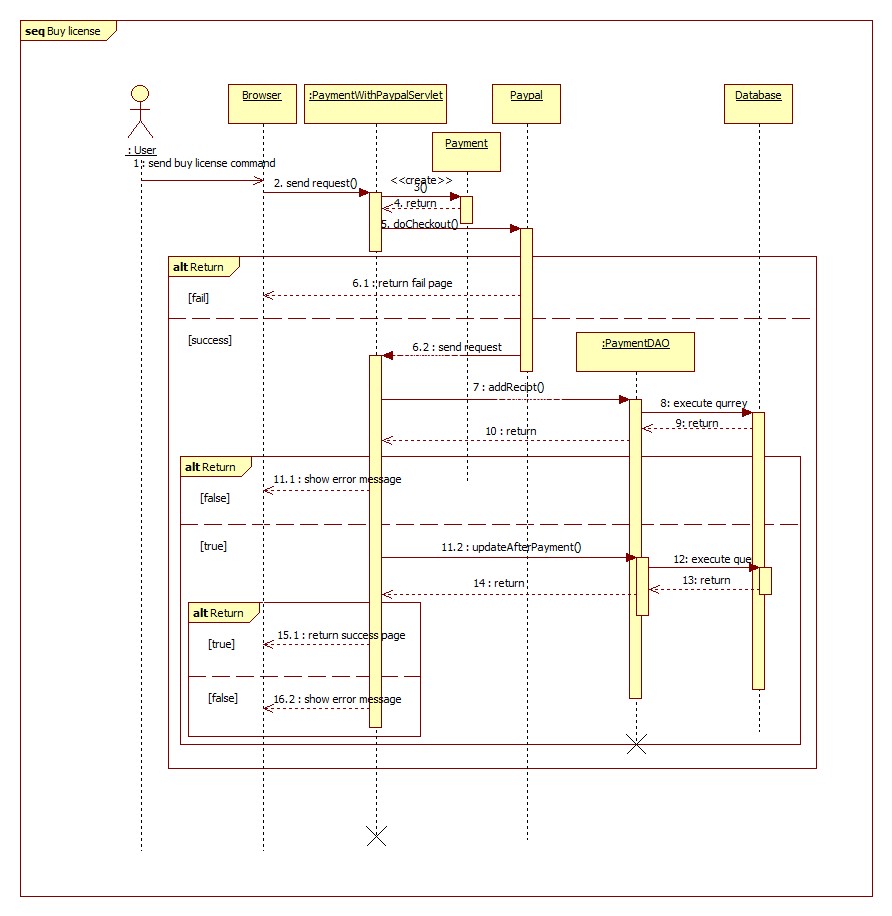


Figure 26: Sequence diagram - <User> Buy license

Update profile

Summary: this diagram show process of update profile

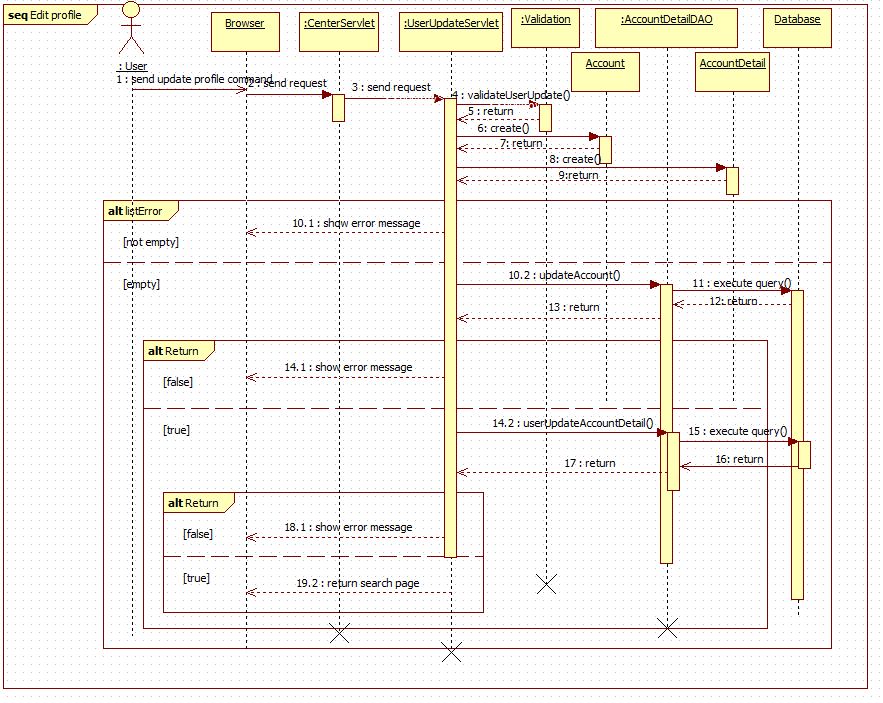


Figure 27: Sequence diagram - <User> Update profile

Search

Summary: this diagram show process of search

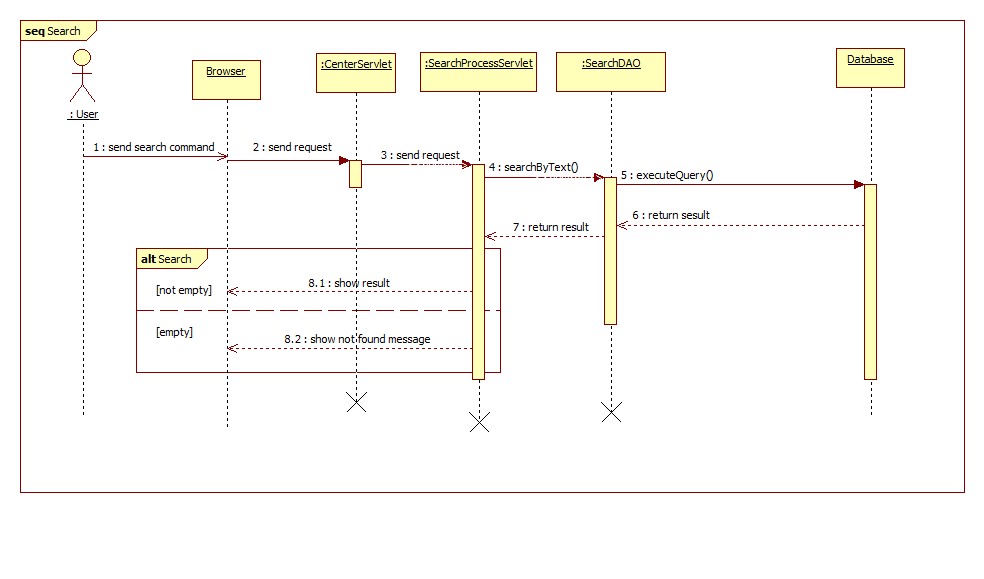


Figure 28: Sequence diagram - <User> Search

###### <Guest>

Register

Summary: this diagram show process of register

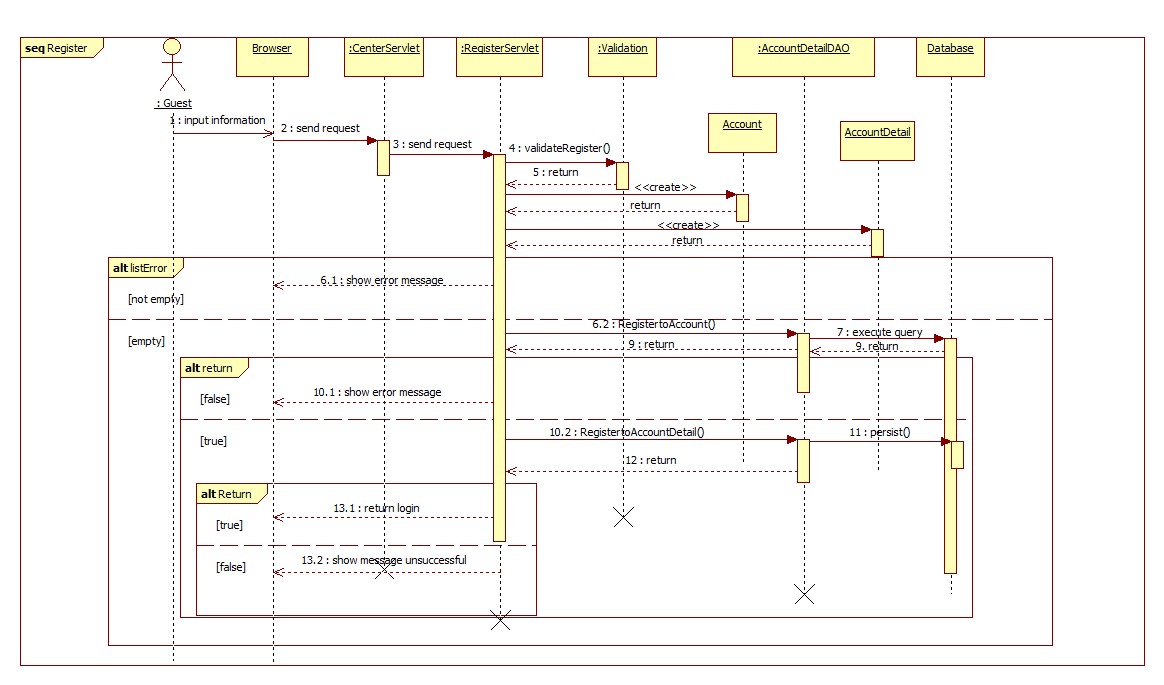


Figure 29: Sequence diagram - <Guest> Register

##### Mobile Application

###### <Staff>

Train online

Summary: this diagram show process of Train online

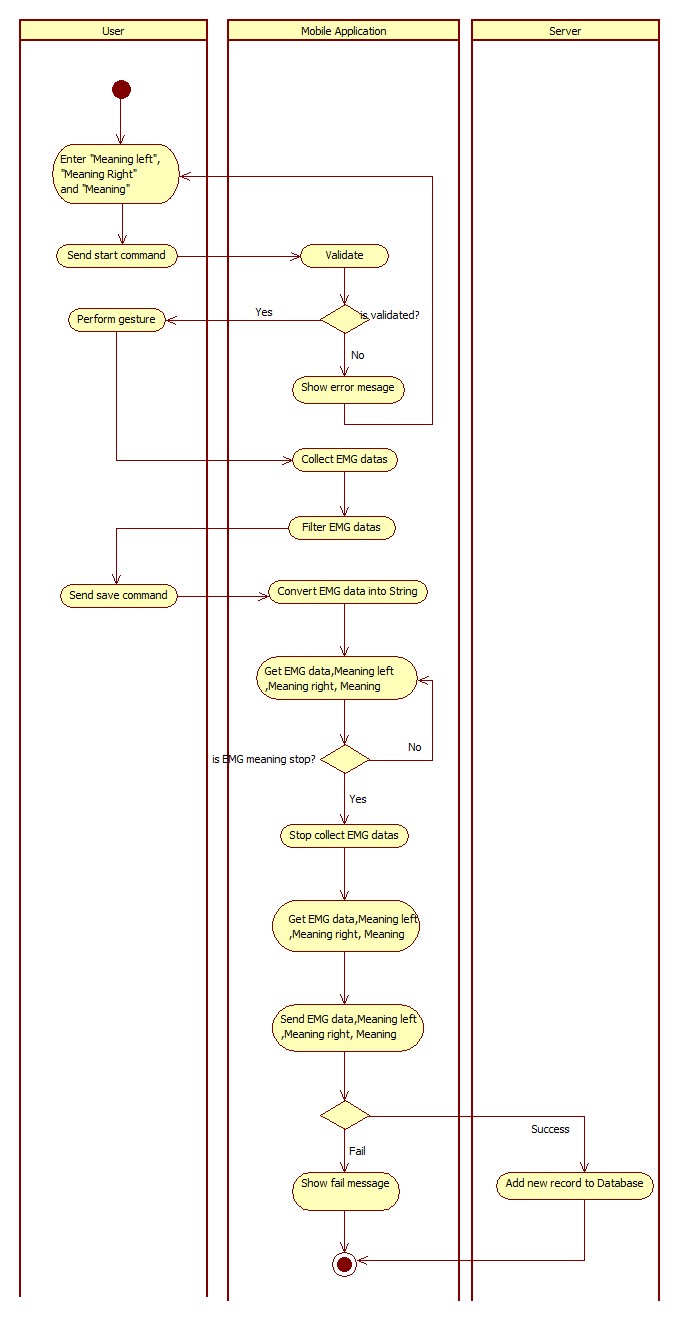


Figure 30: Activity diagram - <Staff> Train online

###### <User>

Connect Myo armbands

Summary: this diagram show process of connect Myo armbands

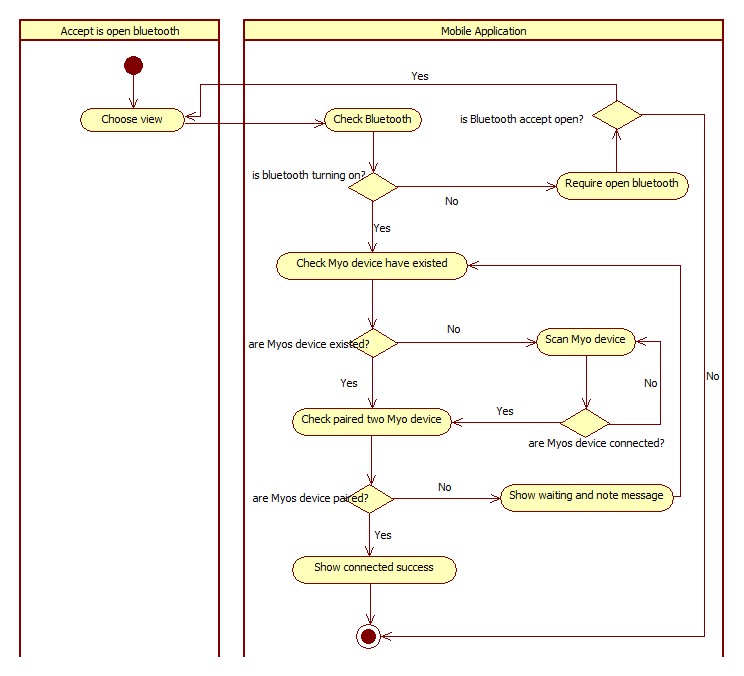


Figure 31: Activity diagram - <User> Connect Myo armbands

Translate online

Summary: this diagram show process of translate online

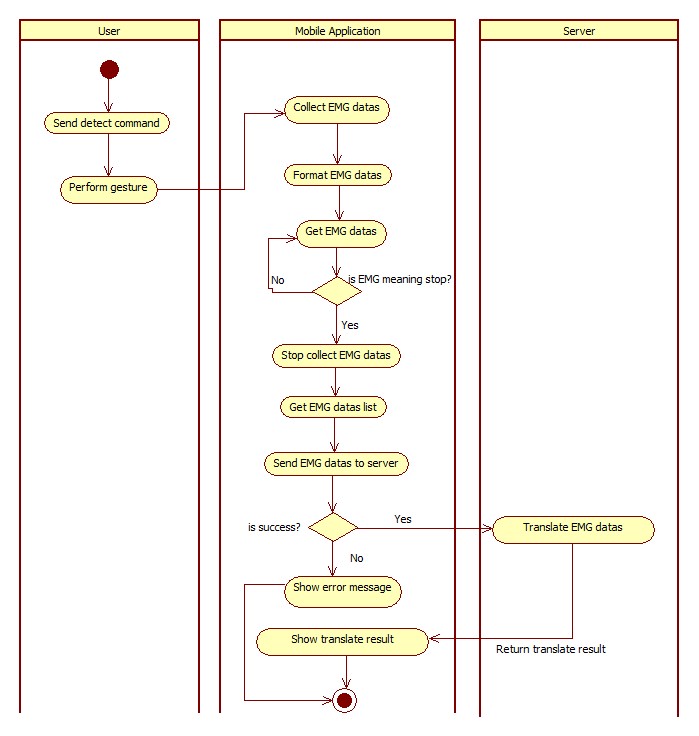


Figure 32: Activity diagram - <User> Translate online

###### <Premium User>

Translate offline

Summary: this diagram show process of translate offline

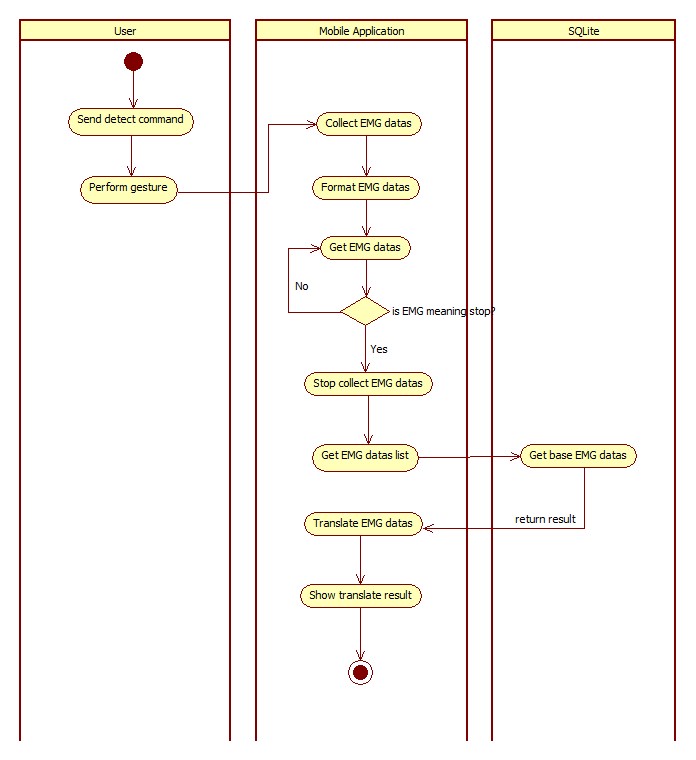


Figure 33: Activity diagram - <Premium User> Translate offline

Train offline

Summary: this diagram show process of train offline

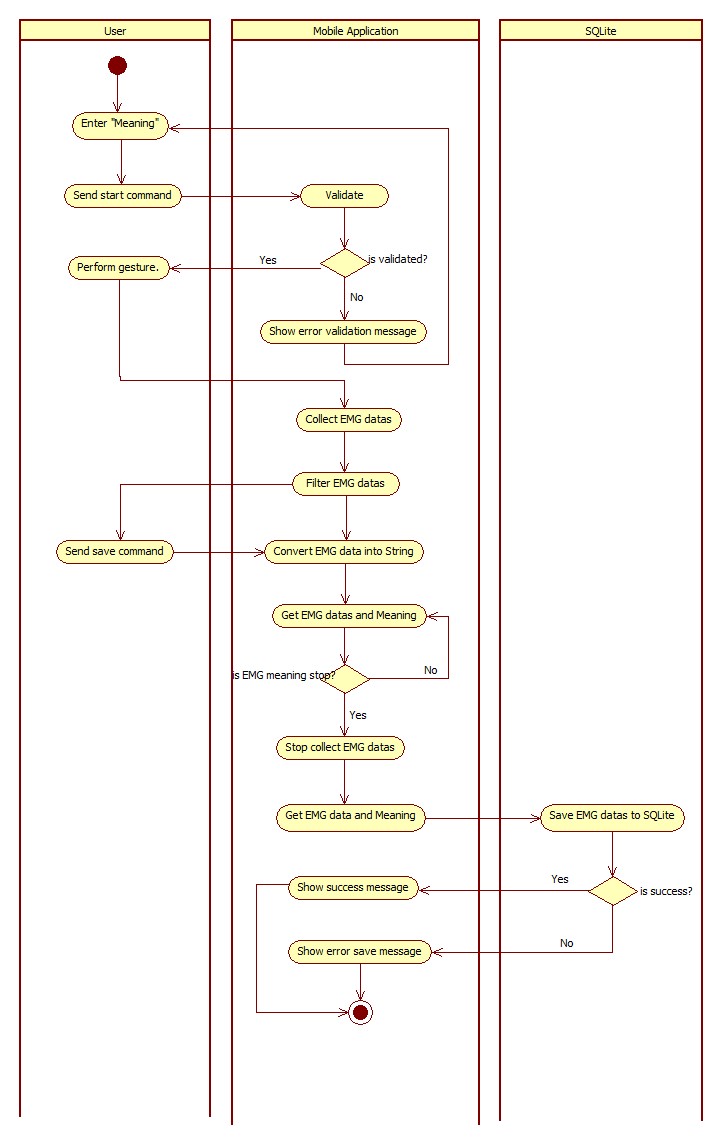


Figure 34: Activity diagram - <Premium User> Train offline

## Interface

### Component Interface

#### Web Services Interface

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

Table 44: Web Services interface

|  |  |
| --- | --- |
| Exception | Description |
| JsonProcessingException | Encountered when processing (parsing, generating) JSON content that are not pure I/O problems |
| NoResultException | Thrown by the persistence provider when getSingleResult() is executed on a query and there is no result to return |
| IOEException | Thrown when there has been an Input/Output (usually when working with files) error. |

Table 45: Exception description

### Window Application Design

#### Guest interface design

##### Register

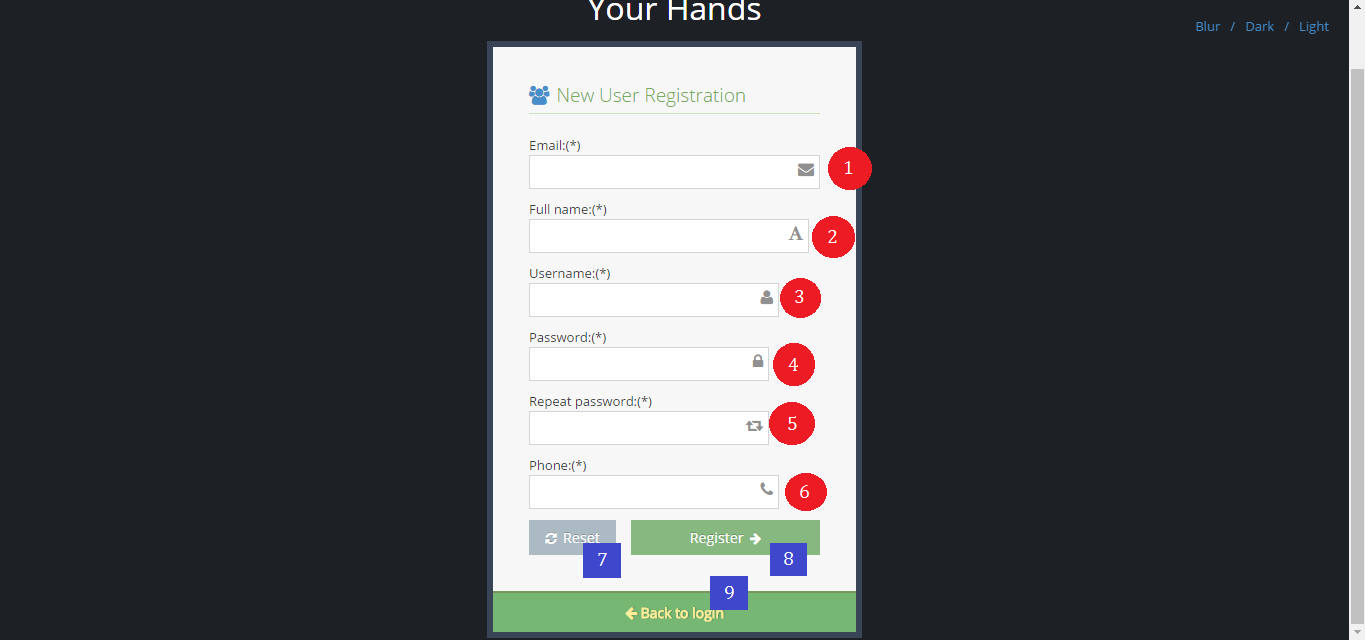


Figure 35: Interface - <Guest> Register

**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

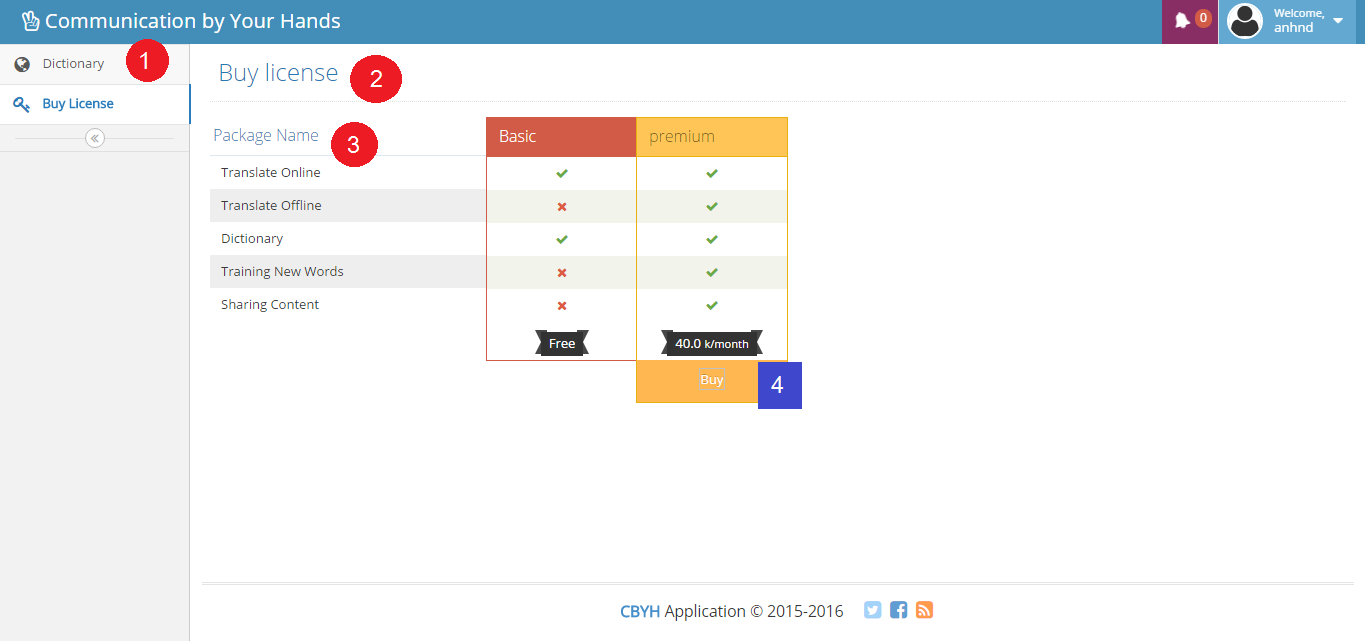


Figure 36: Interface - <User> Buy license

**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

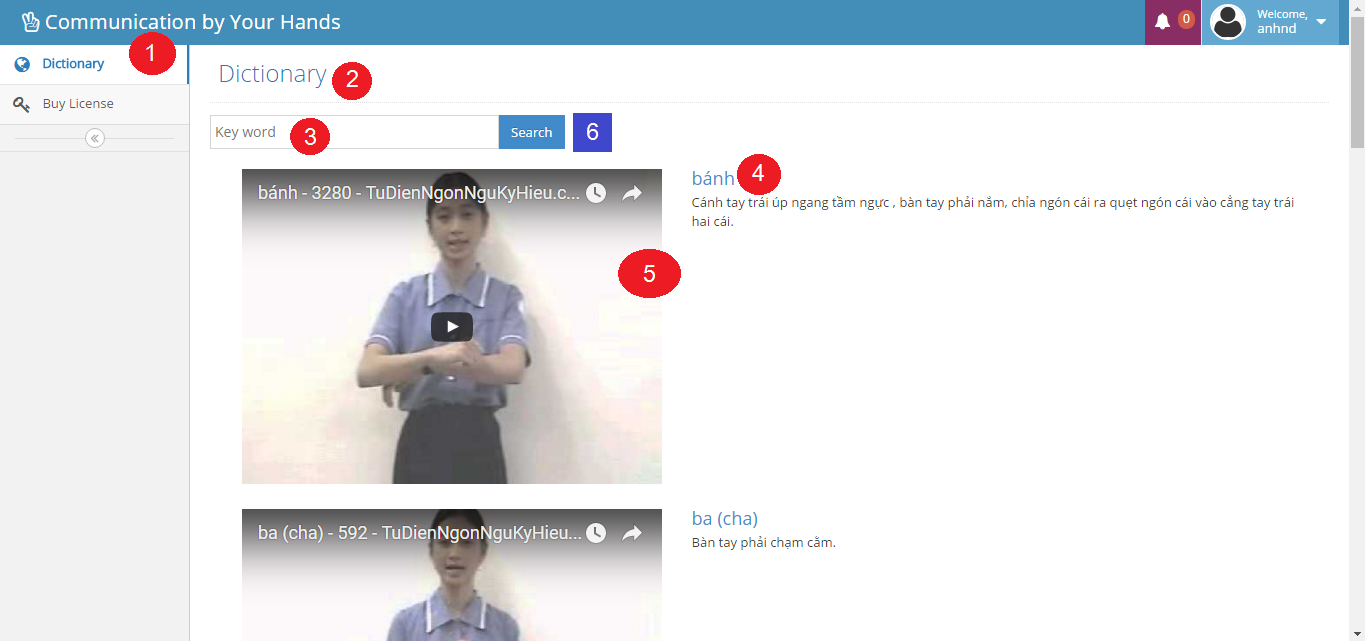


Figure 37: Interface - <User> Search instruction

**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

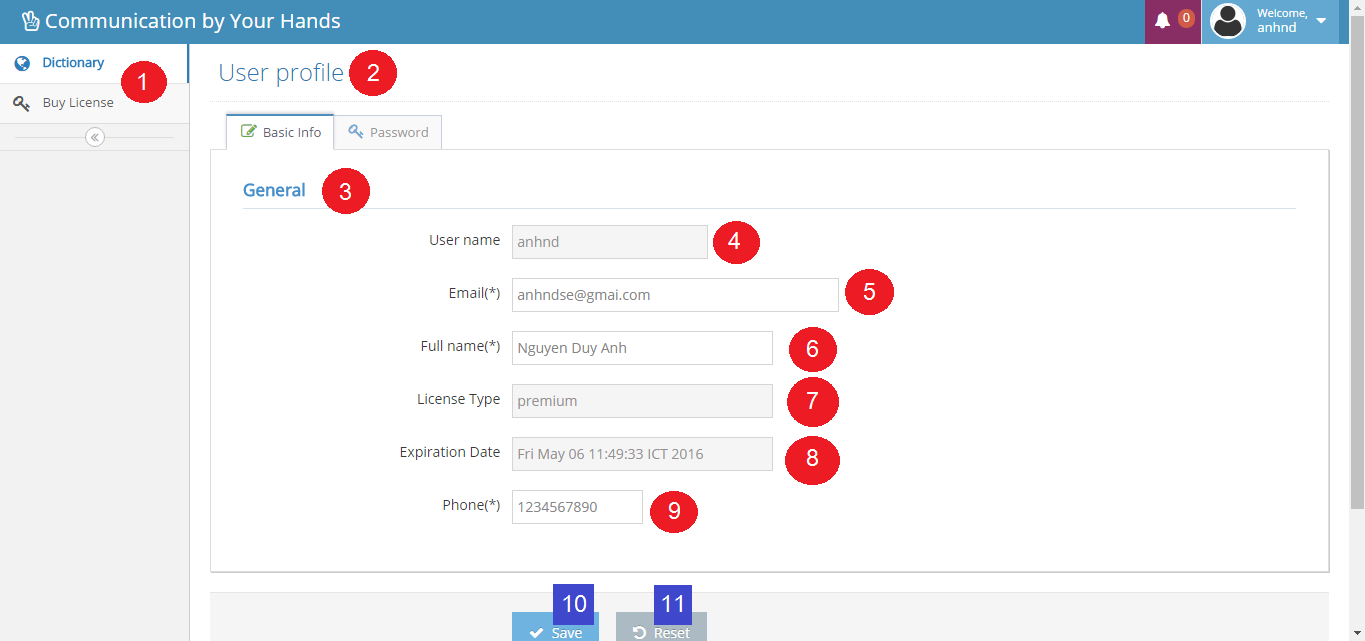


Figure 38: Interface - <User> Edit user profile

**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

## Database design

### Entity relationship diagram

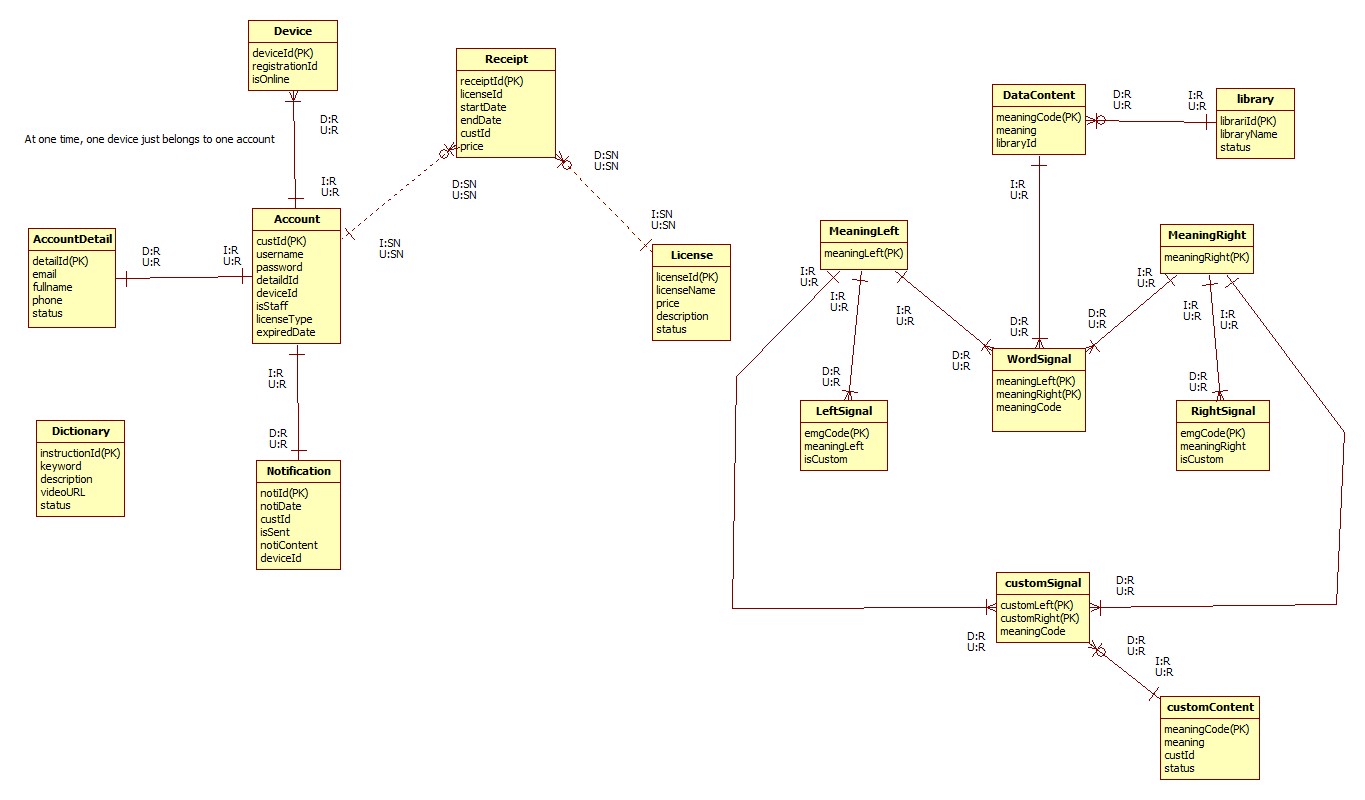


Figure 44: Entity relationship diagram

### Entity dictionary

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

Table 62: Entity dictionary

## Algorithms

### Standardize EMG data (Android)

#### Definition

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

#### Define problem

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

#### Solution

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

+ select two adjacent EMG data

+Absolutize those datas

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

#### Example

Two EMG data:

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

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

Absolutize:

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

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

The new EMG created:

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

#### Complexity

The Complexity is: O(n)

### Matching

#### Definition

Matching is the method that calculate the Distance number

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

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

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

#### Define problem

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

#### Solution

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

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

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

We calculate the distance between A and B call AB.

We calculate the distance between A and O call OA.

We calculate the distance between B and O call OB.

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

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

#### Example

Calculate the matching of two EMG datas:

A: (5, 5, 5, 5, 6, 6, 6, 6)

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

O: (0, 0, 0, 0, 0, 0, 0, 0)

Threshold = 0.01

Calculate distance AB:

AB = =

Calculate distance OA:

OA= =

Calculate distance OB:

OB = =

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

#### Complexity

The complexity is : O(n)

#### Flowchart



Figure 45: Matching workflow

### Detect

#### Definition

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

#### Define problem

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

#### Solution

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

#### Complexity

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

#### Flowchart



Figure 46: Detect workflow



Figure 47: Find meaning right workflow



Figure 48: Find meaning left workflow

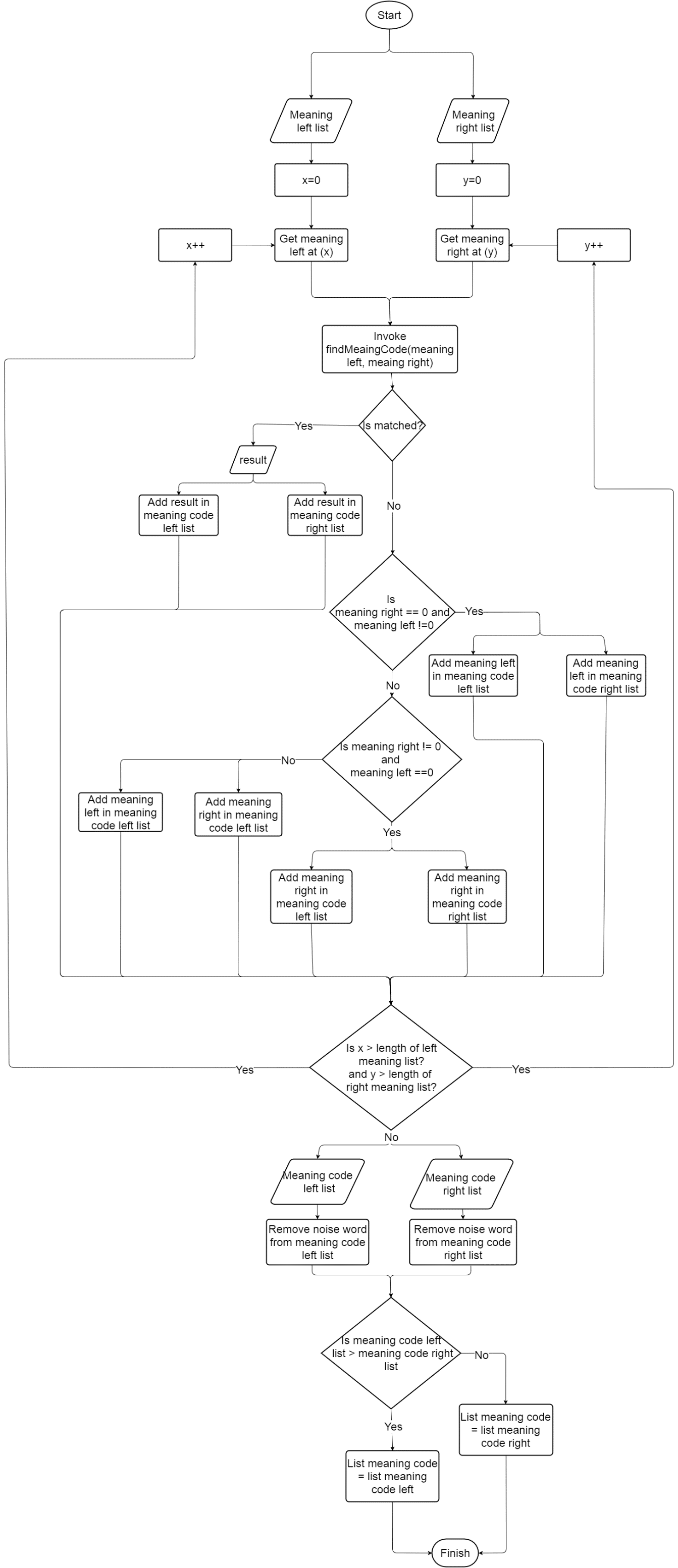


Figure 49: Find meaning of each hand workflow

### Train

#### Definition

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

#### Define problem

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

#### Complexity

The complexity is: O(1)

#### Flowchart



Figure 50: Train workflow

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

#### Definition

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

#### Define problem

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

#### Solution

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

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

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

Norm(OA) =

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

K = 50/Norm()

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

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

#### Example

We have an EMG:

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

Norm(OA) = = 10

K = = = 2.8867513459481

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