**MINISTRY OF EDUCATION AND TRAINING**

**FPT UNIVERSITY**

Capstone Project Document

**Vietnamese Sign Language Recognition**

|  |  |
| --- | --- |
| **Group 05** | |
| **Group members** | Nguyễn Hữu Kỳ Long – Team leader – SE60984  Nguyễn Đình Tân – Team member – SE61115  Nguyễn Xuân Ý – Team member – SE60869  Lê Phương Bình – Team member – SE61049 |
| **Supervisor** | Mr. Đỗ Đức Minh Quân |
| **Ext. Supervisor** | N/A |
| **Capstone Project code** | VSLR |

-Ho Chi Minh City, 24/05/2015-

*This page is intentionally left blank*

# Table of Contents

[Table of Contents 3](#_Toc424806801)

[List of Tables 3](#_Toc424806802)

[List of Figure 3](#_Toc424806803)

[Definitions, Acronyms, and Abbreviations 4](#_Toc424806804)

[A. Report No.2 Software Project Management Plan 5](#_Toc424806805)

[1. Problem Definition 5](#_Toc424806806)

[1.1 Name of this Capstone Project 5](#_Toc424806808)

[1.2 Problem Abstract 5](#_Toc424806809)

[1.3 Project Overview 5](#_Toc424806810)

[2. Project organization 9](#_Toc424806811)

[2.1 Software Process Model 9](#_Toc424806812)

[2.2 Roles and responsibilities 10](#_Toc424806813)

[2.3 Tools and Techniques 11](#_Toc424806814)

[3. Project Management Plan 11](#_Toc424806815)

[2.4 Product Backlog 11](#_Toc424806816)

[2.5 Sprint Backlog 12](#_Toc424806817)

[2.6 All Meeting Minutes 18](#_Toc424806818)

[3. Coding Convention 22](#_Toc424806819)

# List of Tables

[Table 1: Roles and Responsibilities Details 11](#_Toc424806897)

[Table 2: Product Backlog Details 12](#_Toc424806898)

[Table 3: Sprint Backlog Details 16](#_Toc424806899)

# List of Figure

[Figure 1: Boundaries of the System 8](#_Toc424807105)

[Figure 2 : Scrum Development Model 9](#_Toc424807106)

[Figure 3: Chart of Sprint Backlog 17](#_Toc424807107)

# Definitions, Acronyms, and Abbreviations

|  |  |
| --- | --- |
| **Name** | **Definition** |
| VSLR | Vietnamese Sign Language Recognition |
| LCD | Liquid crystal display |
|  |  |
|  |  |

# Report No.2 Software Project Management Plan

# Problem Definition



### Name of this Capstone Project

* Vietnamese Sign Language Recognition

### Problem Abstract

As we know, in the daily life, there is a lot of ways people can understand others such as speech, expression of act, gesture or feelings, etc. However, it is better to express oneself in speech. At the same time, it is an actual matter to mute people to get other people and in the opposite way. The current solution for them is sign language but that means it requires everyone to know sign language of mute people or need someone play as a translator. But these solutions just solve the problem at that time, these are not a long-term strategy. It expects a long time and high cost for preparation from them to solve the problem. In additional, there still are some temporary solutions such as handwriting or using familiar signs, but these way will not produce the desired effect and requires lots of time or effort.

To solve those problems mentioned above, we propose a solution which can help dump person to express themselves in speech or text. That is a device playing a translator and act as intermediary role.

### Project Overview

#### Current Situation and Disadvantages

Below are some current behaviors of user:

* Handwriting:
* People will use something can write on as vehicle for communication.
* They can write out exactly what they want to say to the recipient.
* The recipient can receive and read the content immediately.
* Familiar signs:
* Speakers will describe the word which they want say through action, describe the shape, body language.
* Listeners observe the speaker's actions. They predict information that the speaker shown.
* Interpreters:
* Act as intermediary to translate the content of communication.
* Speakers express words by their language, the interpreter receive information from the speaker and then convey that information by the language of the listener.
* Degree of accuracy of translated content is quite high for both two sides.

Below are the disadvantages of current situation:

* Hand-writing :
* Users must use an intermediary for communication such as paper, pens. However, these things are not always available.
* Users spend more time to write out all their wishes and read them.
* User can meet difficulties about different languages.
* The error can be caused by user handwriting.
* Using familiar signs :
* Maybe be misleading because the symbols are not standardized.
* It is trending towards personally identifiable user.
* It is difficult to show all wishes of communicator.
* Time consuming for understanding the content is long.
* Translator :
* Hiring a translator must be costly.
* Translator who work only in the fixed time, thus not always can meet user's demands.
* Translator must be a experienced person.
* Number of translator is limited.

Analyzing image is the most common way to solve many problems in the real life. One of those problems is recognition. Today, with growth of support analyzing image library and algorithms provided to process image is widespread, tracking and recognition can be performed more easily. Our project is taking into consideration about it to recognize hand signs to help people can communicate with another people.

* Advantages:
* Can be implemented on many different platforms.
* Operating costs less expensive.
* Implemented quickly by many image processing algorithm diversity.
* Disadvantages:
* Analyzing image still remains restriction on process environment, point of view.
* Recognition have still not covered every cases yet. Within weird characterizes, the result maybe not high accurate.
* Currently, analyzing image and recognition just detect and recognize hand signs without motion.
* To get high degree of accuracy, it requires some accessories from users.

#### The Proposed System

Exploiting the development of embedded technology and the growing of image processing, we put forward a system which can recognize hand sign language to help dumb people can communicate. This system includes a camera which captures hand signs from user, a raspberry board plays role as central processing unit which analyzes these captures, processes some algorithms to recognize them and performs some different functions in the system, and a LCD which shows interfaces of the system and recognition result. Besides that, the system still provides some electronic devices to user can control battery, or devices.

##### Controlling System

* Users can turn on/off the system by a switch button.
* Users can monitor the battery capacity.
* Users use hand gestures to select the functions and move between functions.

##### Hand Sign Language Recognize

* Users express hand gestures which describes the desired content, then they can receive the hand sign recognition result.
* Users can see your hand gestures on LCD.
* Users can check the result of the current hand sign.
* Users can edit the current translated content.
* Users receive the recognition result via text or sound shown from LCD.

##### Learning Hand Sign

* Users can choose words that they want to learn which existed in the system.
* Users can see images which express the hand gesture.
* User's hand signs can be practiced and checked by following some steps of the system.
* Users receive the current recognized result of the hand sign via text or sound.

#### Boundaries of the System

##### The restrictions

* The system language is Vietnamese.
* Hand sign language the system supports is Vietnamese sign language.
* The system requires users must use supported accessories.
* The system requires users must provide a stable environment in room with sufficient light and a background is not complex on color, especially, no color close to skin color.
* The system must be fixed during the working process.

##### The components of the system:

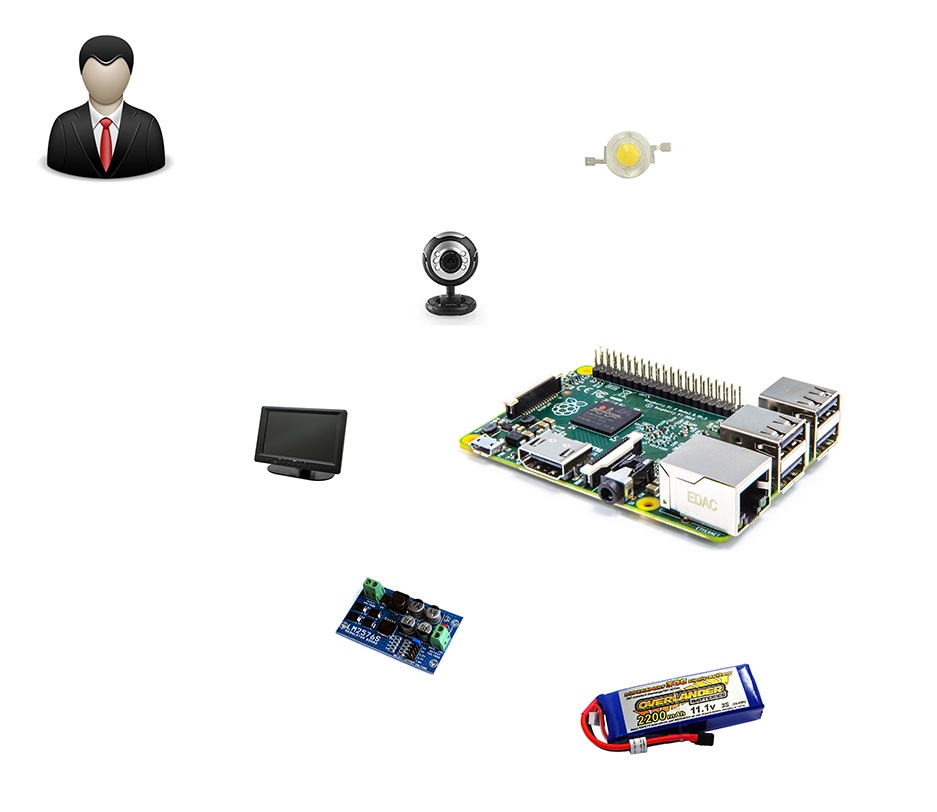


Figure 1: Boundaries of the System

#### Development Environment

##### Hardware requirements

* 4 laptops is used for development the system. These are setup Ubuntu 14.04 operating system.
* Raspberry Pi B2 is used to process as central processing unit.
* Cable is connection between laptop and raspberry pi 2.
* Keyboard, mouse, and usb wifi are used to setup operating system and necessary environments for raspberry pi 2.
* Backup flash memory: a backup solution when problems with operating system. This memory must be setup similiar to main flash memory.
* LIPO battery (12V – 3A): power for the system can works.
* Camera module of raspberry kit: is used to capture images.
* LCD 7 inch is used to show the inteface of functions and the recognized results.
* 2 Led (1W): is used to balance light.
* LM2576ADJ-Board connect between LIPO battery and raspberry pi 2

##### Software requirements

* Linux: operating system and platform for deploy
* Remote Desktop: application for remoting to work on raspberry
* QT Creator: is to develop c++ application and Linux GUI
* OpenCV 2.4.9 library: supporting image processing.
* SQLite: software creates and manages the system database.
* Software Ideas Modeler: application for creating models and diagrams.
* Microsoft Office 2010: is used to write documents and assign tasks.
* Githup and TortoiseSVN and Rabbit VCS: used for source control
* Skype: used for communication and meeting

## Project organization

### Software Process Model

#### Overall Description

Scrum is an agile methodology that can be applied to nearly any project; however, the Scrum methodology is most commonly used in software development. The Scrum process is suited for projects with rapidly changing or highly emergent requirements. Scrum software development progresses via a series of iterations called sprints, which last from one to four weeks. In the agile Scrum world, a sprint planning meeting is described in terms of the desired outcome (a commitment to a set of features to be developed in the next sprint) instead of a set of Entry criteria, Task definitions, Validation criteria, Exit criteria. The Scrum model suggests each sprint begins with a brief planning meeting and concludes with a review. These are the basics of Scrum project management.

#### Scrum Development Model



Figure 2 : Scrum Development Model

#### Reasons for Choosing

Project is developed under scrum model. We choose this model because the scope of the project is not fixed when the requirement changes day by day. Products are created quickly. Therefore ,the development team can easy to change if the wrong direction. Degree of cooperation between the members is set to high.

### Roles and responsibilities

|  |  |  |  |
| --- | --- | --- | --- |
| **No** | **Full name** | **Role in Group** | **Responsibilities** |
| **1** | Đỗ Đức Minh Quân | Scrum Master/Product Owner | * Defining user requirements * Specifying business * Control the development process * Give advices on techniques, solutions and business analysis support |
| **2** | Nguyễn Hữu Kỳ Long | Team Leader, BA, DEV, Tester | * Managing process * Clarifying requirements * Researching solutions and techniques * Assigning task for members * Reviewing the result of task of members. * Editing documents and reports * Reviewing documents and reports * Developing the system software * Reviewing the system hardware * Coding * Creating test plan. * Testing |
| **3** | Nguyễn Đình Tân | Team Member, BA, DEV, Tester | * Clarifying requirements * Researching solutions and techniques * Designing database * Preparing documents and reports * Reviewing documents and reports * Developing the system software * Reviewing the system hardware * Coding * Testing |
| **4** | Lê Phương Bình | Team Member, BA, DEV, Tester | * Clarifying requirements * Preparing documents and reports * Reviewing documents and reports * Developing the system hardware * Reviewing the system software * Coding * Testing |
| **5** | Nguyễn Xuân Ý | Team Member, BA, DEV, Tester | * Clarifying requirements * Editing documents and reports * Reviewing documents and reports * Developing the system hardware * Coding * Testing |

Table 1: Roles and Responsibilities Details

### Tools and Techniques

* Front-end and back-end IDE:
* QT 5.4 Creator
* Front-end technology:
* QT Linux GUI
* Back-end library:
* OPENCV 2.4.9 library
* LIBSVM library
* Espeak 1.48.04 library
* Managing database:
* SQLite 3
* Connecting to Raspberry PI 2:
* Remote Desktop Connection of Ubuntu 14.04
* Managing the project:
* SVNtortoise
* Rabbit VCS
* Managing documents, reports, models and diagrams:
* Software Ideas Modeler version 7.70.5385.38708
* Microsoft Office 2010

## Project Management Plan

### Product Backlog

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **ID** | **Theme** | **User Type** | **Wants to...** | **So that...** | **Priority** | **Sprint** |
| 1 | Detection | User | keep track their hand gesture | can see his/her hand in the screen | Very High | 1 |
| 2 | Device | User | the system is a portable system | move the device easily and use it more flexibly | High | 1 |
| 3 | Recognition | User | recognize the hand signs | express the same meaning to the partner can understand | Very High | 2 |
| 4 | Recognition | User | receive the recognition result via text and sound | express the translated content in a clearly way | High | 2 |
| 5 | Detection | User | control the system functions by hand gesture | perform and move beetween the system functions | High | 3 |
| 6 | Power | User | know remaining of battery capacity | can monitor the use of device | High | 3 |
| 7 | Recognition | User | increase the accuracy of the recognition result | raise the reliability of the translated content | Very High | 4 |
| 8 | Learning | User | learn the hand sign language | learn new signs or pratice his/her signs | Medium | 4 |
| 9 | Device | User | turn on/off the system | can turn on/off the device according to the demand | Medium | 4 |

Table 2: Product Backlog Details

### Sprint Backlog

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Sprint** | **User Story** | **Time (est)** | **Time (spent)** | **Time (left)** | **Task** | **Time (est)** | **Who's working** | **Status** | **Work Done per week per task [Days]** | | | | | | | | | | | |
|  | *Project Name: "Vietnamese Language Sign Recognition" Started: 12-05-2015* | | | | | | | | **1** | **2** | **3** | **4** | **5** | **6** | **7** | **8** | **9** | **10** | **11** | **12** |
| 1 | As a user, I want to keep track my hand gesture | 15 | 15 | 0 | Setup development environment for Raspberry PI and laptops | **3** | All Team | Done | 3 |  |  |  |  |  |  |  |  |  |  |  |
| Capturing images from camera and showing them to user | **4** | TanND | Done | 1 | 3 |  |  |  |  |  |  |  |  |  |  |
| Extracting background color | **4** | LongNHK | Done | 1 | 3 |  |  |  |  |  |  |  |  |  |  |
| Extracting hand color | **4** | LongNHK | Done |  | 2 | 2 |  |  |  |  |  |  |  |  |  |
| Subtracting color to get hand binary image | **4** | LongNHK | Done |  |  |  | 1 |  |  |  |  |  |  |  |  |
| Finding the hand contours on the hand binary image | **4** | TanND | Done |  | 2 | 2 |  |  |  |  |  |  |  |  |  |
| Designing QT Linux GUI | **4** | TanND | Done |  |  | 3 | 1 |  |  |  |  |  |  |  |  |
| As a user, I want the system is a portable system | 15 | 15 | 0 | Choose type of battery | **3** | YNX | Done | 1 | 2 |  |  |  |  |  |  |  |  |  |  |
| Choose voltage regulator circuit | **3** | BinhLP | Done | 1 | 2 |  |  |  |  |  |  |  |  |  |  |
| Connecting components | **5** | BinhLP + YNX | Done |  | 3 | 2 |  |  |  |  |  |  |  |  |  |
| Adjusting the device components | **4** | BinhLP + YNX | Done |  |  | 3 | 1 |  |  |  |  |  |  |  |  |
| 2 | As a user, I want to recognize the hand signs | 13 | 13 | 0 | Creating SQLite database | **2** | TanND | Done |  |  |  | 2 |  |  |  |  |  |  |  |  |
| Training SVM | **3** | LongNHK | Done |  |  |  | 3 |  |  |  |  |  |  |  |  |
| Defining features to recognize | **3** | LongNHK | Done |  |  |  | 1 | 2 |  |  |  |  |  |  |  |
| Features extraction | **4** | LongNHK | Done |  |  |  |  | 3 | 1 |  |  |  |  |  |  |
| Recognizing the hand sign by SVM | **3** | LongNHK | Done |  |  |  |  |  | 3 |  |  |  |  |  |  |
| Defining the meaning word of the SVM result from database | **3** | TanND + YNX | Done |  |  |  | 2 | 1 |  |  |  |  |  |  |  |
| Designing QT Linux GUI | **4** | BinhLP + YNX | Done |  |  |  | 4 |  |  |  |  |  |  |  |  |
| As a user, I want to receive the recognition result via text and sound | 10 | 10 | 0 | Choosing text to speech opensource | **3** | TanND + BinhLP | Done |  |  |  |  | 3 |  |  |  |  |  |  |  |
| Implementing text to speech opensource | **3** | TanND + BinhLP | Done |  |  |  |  | 1 | 2 |  |  |  |  |  |  |
| Changing pronunciation of word | **4** | TanND + YNX | Done |  |  |  |  |  | 3 | 1 |  |  |  |  |  |
| Showing the recognition result via text | **2** | LongNHK | Done |  |  |  |  |  | 1 | 1 |  |  |  |  |  |
| Showing the recognition result via sound | **4** | BinhLP + YNX | Done |  |  |  |  |  | 3 | 1 |  |  |  |  |  |
| 3 | As a user, I want to control the system functions by hand gesture | 15 | 15 | 0 | Defining operations of function | **3** | LongNHK + TanND | Done |  |  |  |  |  |  | 3 |  |  |  |  |  |
| Implementing operations of function | **3** | LongNHK + TanND | Done |  |  |  |  |  |  | 1 | 2 |  |  |  |  |
| Designing QT Linux GUI | **4** | LongNHK + TanND | Done |  |  |  |  |  |  |  | 3 | 1 |  |  |  |
| Synchronize between hand gestures and operations of function | **5** | LongNHK + TanND | Done |  |  |  |  |  |  |  |  | 4 | 1 |  |  |
| As a user, I want to know remaining of battery capacity | 15 | 15 | 0 | Choosing chip voltage comparator | **3** | BinhLP + YNX | Done |  |  |  |  |  |  | 3 |  |  |  |  |  |
| Choosing zener | **3** | BinhLP + YNX | Done |  |  |  |  |  |  | 1 | 2 |  |  |  |  |
| Constructing circuit | **4** | BinhLP + YNX | Done |  |  |  |  |  |  |  | 3 | 1 |  |  |  |
| Adjusting the device components | **2** | BinhLP + YNX | Done |  |  |  |  |  |  |  |  | 2 |  |  |  |
| Connecting to the system | **3** | BinhLP + YNX | Done |  |  |  |  |  |  |  |  | 2 | 1 |  |  |
| 4 | As a user, I want to increase the accuracy of the recognition result | 11 | 11 | 0 | Defining more features to recognize | **3** | All Team | Done |  |  |  |  |  |  |  |  |  | 3 |  |  |
| Extracting features | **3** | LongNHK | Done |  |  |  |  |  |  |  |  |  | 1 | 2 |  |
| Providing two LEDs to balance light | 3 | BinhLP + YNX | Done |  |  |  |  |  |  |  |  |  | 1 | 2 |  |
| Improving camera setting | 2 | TanND + YNX | Done |  |  |  |  |  |  |  |  |  | 1 | 1 |  |
| Improving background - hand color subtraction | 2 | LongNHK + TanND | Done |  |  |  |  |  |  |  |  |  |  | 3 |  |
| Training SVM | **3** | LongNHK | Done |  |  |  |  |  |  |  |  |  |  |  | 2 |
| As a user, I want to learn hand sign language | 8 | 8 | 0 | Creating database | **2** | TanND | Done |  |  |  |  |  |  |  |  |  |  | 1 |  |
| Managing database | **2** | TanND | Done |  |  |  |  |  |  |  |  |  |  |  | 2 |
| Implementing the practice function of learning | **4** | LongNHK + BinhLP | Done |  |  |  |  |  |  |  |  |  |  |  | 4 |
| Designing GUI | **4** | TanND + YNX | Done |  |  |  |  |  |  |  |  |  |  |  | 4 |
| As a user, I want to turn on/off the system | 8 | 8 | 0 | Choosing switch button | **2** | BinhLP | Done |  |  |  |  |  |  |  |  |  |  | 2 |  |
| Constructing circuit | **3** | BinhLP + YNX | Done |  |  |  |  |  |  |  |  |  |  |  | 3 |
| Connecting to the system | **3** | BinhLP | Done |  |  |  |  |  |  |  |  |  |  |  | 3 |
|  | TOTAL | 60 | 60 | 0 |  |  |  |  | 5 | 5 | 5 | 5 | 5 | 5 | 5 | 5 | 5 | 5 | 5 | 5 |
|  | Daily burnout | 0 |  |  |  |  |  |  | **5** | **5** | **5** | **5** | **5** | **5** | **5** | **5** | **5** | **5** | **5** | **5** |
|  | **Total time left (from estimate)** |  |  |  |  |  | Estimate | 60 | 55 | 50 | 45 | 40 | 35 | 30 | 25 | 20 | 15 | 10 | 5 | 0 |
|  | **Total time left (from spent)** |  |  |  |  |  | Burnout | 60 | 55 | 50 | 45 | 40 | 35 | 30 | 25 | 20 | 15 | 10 | 5 | 0 |

Table 3: Sprint Backlog Details

Figure 3: Chart of Sprint Backlog

### All Meeting Minutes

***Scrum Meeting Report***

* 12/05/2015 : SUMMARY OF MEETING
  + LongNHK
    - Setup development environment for Raspberry PI
  + TanND
    - Setup development environment for Raspberry PI and laptops
  + YNX
    - Setup development environment for Raspberry PI and laptops
  + BinhLP
    - Setup development environment for Raspberry PI and laptops
* 16/05/2015: SUMMARY OF MEETING
  + LongNHK
    - Extracting background color
  + TanND
    - Capturing images from camera and showing them to user
  + YNX
    - Choose type of battery
  + BinhLP
    - Choose voltage regulator circuit
* 19/05/2015: SUMMARY OF MEETING
  + LongNHK
    - Extracting background color
  + TanND
    - Capturing images from camera and showing them to user
  + YNX
    - Choose type of battery
  + BinhLP
    - Choose voltage regulator circuit
* 23/5/2015: SUMMARY OF MEETING
  + LongNHK
    - Extracting hand color
  + TanND
    - Finding the hand contours on the hand binary image
  + YNX
    - Connecting components
  + BinhLP
    - Connecting components
* 26/5/2015: SUMMARY OF MEETING
  + LongNHK
    - Extracting hand color
  + TanND
    - Finding the hand contours on the hand binary image
  + YNX
    - Connecting components
  + BinhLP
    - Connecting components
* 30/5/2015: SUMMARY OF MEETING
  + LongNHK
    - Subtracting color to get hand binary image
  + TanND
    - Designing QT Linux GUI
  + YNX
    - Adjusting the device components
  + BinhLP
    - Adjusting the device components
* 2/6/2015: SUMMARY OF MEETING
  + LongNHK
    - Training SVM
  + TanND
    - Creating SQLite database
  + YNX
    - Designing QT Linux GUI
  + BinhLP
    - Designing QT Linux GUI
* 6/6/2015: SUMMARY OF MEETING
  + LongNHK
    - Defining features to recognize
  + TanND
    - Defining the meaning word of the SVM result from database
  + YNX
    - Designing QT Linux GUI
  + BinhLP
    - Designing QT Linux GUI
* 9/6/2015: SUMMARY OF MEETING
  + LongNHK
    - Defining features to recognize
  + TanND
    - Choosing text to speech opensource
  + YNX
    - Defining the meaning word of the SVM result from database
  + BinhLP
    - Choosing text to speech opensource
* 13/6/2015: SUMMARY OF MEETING
  + LongNHK
    - Features extraction
  + TanND
    - Implementing text to speech opensource
  + YNX
    - Changing pronunciation of word
  + BinhLP
    - Implementing text to speech opensource
* 16/6/2015: SUMMARY OF MEETING
  + LongNHK
    - Recognizing the hand sign by SVM
  + TanND
    - Changing pronunciation of word
  + YNX
    - Changing pronunciation of word
  + BinhLP
    - Showing the recognition result via sound
* 20/6/2015: SUMMARY OF MEETING
  + LongNHK
    - Showing the recognition result via text
  + TanND
    - Changing pronunciation of word
  + YNX
    - Showing the recognition result via sound
  + BinhLP
    - Showing the recognition result via sound
* 30/6/2015: SUMMARY OF MEETING
  + LongNHK
    - Defining operations of function
  + TanND
    - Defining operations of function
  + YNX
    - Choosing chip voltage comparator
  + BinhLP
    - Choosing chip voltage comparator
* 4/7/2015: SUMMARY OF MEETING
  + LongNHK
    - Implementing operations of function
  + TanND
    - Implementing operations of function
  + YNX
    - Choosing zener
  + BinhLP
    - Choosing zener
* 7/7/2015: SUMMARY OF MEETING
  + LongNHK
    - Implementing operations of function
  + TanND
    - Implementing operations of function
  + YNX
    - Constructing circuit
  + BinhLP
    - Constructing circuit
* 11/7/2015: SUMMARY OF MEETING
  + LongNHK
    - Designing QT Linux GUI
  + TanND
    - Designing QT Linux GUI
  + YNX
    - Constructing circuit
  + BinhLP
    - Constructing circuit
* 14/7/2015: SUMMARY OF MEETING
  + LongNHK
    - Synchronize between hand gestures and operations of function
  + TanND
    - Synchronize between hand gestures and operations of function
  + YNX
    - Adjusting the device components
  + BinhLP
    - Adjusting the device components
* 18/7/2015: SUMMARY OF MEETING
  + LongNHK
    - Synchronize between hand gestures and operations of function
  + TanND
    - Synchronize between hand gestures and operations of function
  + YNX
    - Connecting to the system
  + BinhLP
    - Connecting to the system
* 21/7/2015: SUMMARY OF MEETING
  + LongNHK
    - Defining more features to recognize
  + TanND
    - Defining more features to recognize
  + YNX
    - Defining more features to recognize
  + BinhLP
    - Defining more features to recognize
* 25/7/2015: SUMMARY OF MEETING
  + LongNHK
    - Extracting features
  + TanND
    - Improving camera setting
  + YNX
    - Providing two LEDs to balance light
  + BinhLP
    - Providing two LEDs to balance light
* 28/7/2015: SUMMARY OF MEETING
  + LongNHK
    - Extracting features
  + TanND
    - Improving background - hand color subtraction
  + YNX
    - Improving camera setting
  + BinhLP
    - Providing two LEDs to balance light
* 1/8/2015: SUMMARY OF MEETING
  + LongNHK
    - Improving background - hand color subtraction
  + TanND
    - Creating database
  + YNX
    - Designing GUI
  + BinhLP
    - Implementing the practice function of learning
* 4/8/2015: SUMMARY OF MEETING
  + LongNHK
    - Training SVM
  + TanND
    - Managing database
  + YNX
    - Designing GUI
  + BinhLP
    - Choosing switch button
* 8/8/2015: SUMMARY OF MEETING
  + LongNHK
    - Implementing the practice function of learning
  + TanND
    - Designing GUI
  + YNX
    - Constructing circuit
  + BinhLP
    - Constructing circuit
* 11/8/2015: SUMMARY OF MEETING
  + LongNHK
    - Implementing the practice function of learning
  + TanND
    - Designing GUI
  + YNX
    - Constructing circuit
  + BinhLP
    - Connecting to the system

## Coding Convention

*General view of C++ Programming Style put into practice in the project*

* Naming Conventions
* Variable names must be in mixed case starting with lower case.
* Named constants must be all uppercase using underscore to separate words.
* Names representing methods or functions must be verbs and written in mixed case starting with lower case.
* Plural form should be used on names representing a collection of objects
* The prefix is should be used for boolean variables and methods
* Include Files and Include Statements
* Header files must contain an include guard
* Include statements should be sorted and grouped
* Include statements must be located at the top of a file only
* Variables
* Class variables should never be declared public
* C++ pointers and references should have their reference symbol next to the type rather than to the name
* Conditionals
* Complex conditional expressions must be avoided
* The conditional should be put on a separate line
* Executable statements in conditionals must be avoided
* Comments
* Use // for all comments, including multi-line comments
* Comments should be included relative to their position in the code
* Class and method header comments should follow the JavaDoc conventions

*References*

C++ Programming Style Guidelines, Version 4.9, January 2011, Geotechnical Software Services, Copyright © 1996 – 2011

<http://geosoft.no/development/cppstyle.html>