**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, 17/05/2015-

*This page is intentionally left blank*

# Table of Contents

[Table of Contents 3](#_Toc427025662)

[List of Tables 6](#_Toc427025663)

[List of Figure 6](#_Toc427025664)

[Definitions, Acronyms, and Abbreviations 8](#_Toc427025665)

[A. Report No. 1 Introduction 9](#_Toc427025666)

[1. Project Information 9](#_Toc427025667)

[2. Introduction 9](#_Toc427025668)

[3. Current Situation 9](#_Toc427025669)

[4. Problem Definition 9](#_Toc427025670)

[5. Proposed Solution 9](#_Toc427025671)

[5.1 Feature functions 9](#_Toc427025672)

[5.2 Advantages and disadvantages 10](#_Toc427025673)

[6. Functional Requirements 10](#_Toc427025674)

[6.1 Tracking hand 10](#_Toc427025675)

[6.2 Hand recognition 10](#_Toc427025676)

[6.3 Showing the content 10](#_Toc427025677)

[6.4 Learning hand sign 10](#_Toc427025678)

[6.5 Controlling System 10](#_Toc427025679)

[6.6 Controlling power 10](#_Toc427025680)

[7. Role and Responsibility 11](#_Toc427025681)

[B. Report No.2 Software Project Management Plan 11](#_Toc427025682)

[1. Problem Definition 11](#_Toc427025683)

[1.1 Name of this Capstone Project 11](#_Toc427025684)

[1.2 Problem Abstract 11](#_Toc427025685)

[1.3 Project Overview 11](#_Toc427025686)

[Figure 1: Components of the the system 14](#_Toc427025687)

[2. Project organization 15](#_Toc427025688)

[2.1 Software Process Model 15](#_Toc427025689)

[Figure 2 : Scrum Development Model 16](#_Toc427025690)

[2.2 Roles and responsibilities 16](#_Toc427025691)

[2.3 Tools and Techniques 18](#_Toc427025692)

[3. Project Management Plan 19](#_Toc427025693)

[3.1 Product Backlog 19](#_Toc427025695)

[3.2 Sprint Backlog 20](#_Toc427025696)

[3.3 Sprint Burndown Chart 29](#_Toc427025702)

[Figure 3: Chart of Sprint Backlog 29](#_Toc427025703)

[3.4 All Meeting Minutes 30](#_Toc427025704)

[4. Coding Convention 39](#_Toc427025705)

[C. Report No. 3 Software Requirement Specification 40](#_Toc427025706)

[1. User Requirement Specification 40](#_Toc427025707)

[4. System Requirement Specification 40](#_Toc427025708)

[4.1 External Interface Requirement 40](#_Toc427025709)

[4.2 System Overview Use Case 41](#_Toc427025710)

[Figure 4: System Overview Use Case 41](#_Toc427025711)

[4.3 List of Use Case 42](#_Toc427025712)

[Figure 6: Recognize Hand Sign Language use case diagram 42](#_Toc427025713)

[Figure 6: Recognize Hand Sign Language use case diagram 45](#_Toc427025714)

[Figure 8: Charge Battery use case diagram 51](#_Toc427025715)

[Figure 10: Notify Battery Capacity use case diagram 52](#_Toc427025716)

[5. System Attribute 54](#_Toc427025717)

[5.1 Usability 54](#_Toc427025718)

[5.2 Reliability 54](#_Toc427025719)

[5.3 Availability 55](#_Toc427025720)

[5.4 Security 55](#_Toc427025721)

[5.5 Maintainability 55](#_Toc427025722)

[5.6 Portability 55](#_Toc427025723)

[5.7 Performance 55](#_Toc427025724)

[D. Report No. 4 Software Design Description 55](#_Toc427025725)

[1. Design Overview 55](#_Toc427025726)

[2. System Architectural Design 56](#_Toc427025727)

[Figure 1: VSLR System Architectural 56](#_Toc427025728)

[3. Component Diagram 56](#_Toc427025729)

[Figure 2: Component Diagram 57](#_Toc427025730)

[4. Detailed Description 58](#_Toc427025731)

[4.1 Software Detailed Description 58](#_Toc427025732)

[2. User Interface Design 96](#_Toc427025744)

[2.1 Subtract Background Color 96](#_Toc427025745)

[Figure 5: Subtract Background Color 96](#_Toc427025746)

[2.2 Recognize Hand Sign Language 96](#_Toc427025747)

[Figure 5: Recognize Hand Sign Language 97](#_Toc427025748)

[2.3 Learn Hand Sign Language 97](#_Toc427025749)

[Figure 5: Learn Hand Sign Language 98](#_Toc427025750)

[2.4 Select 98](#_Toc427025751)

[Figure 5: Select 99](#_Toc427025752)

[2.5 Hardware Detailed Description 100](#_Toc427025753)

[2.6 Algorithms 110](#_Toc427025768)

[E. Report No.5 System Implementation & Test 140](#_Toc427025769)

[1. Introduction 140](#_Toc427025770)

[1.1 Overview 140](#_Toc427025771)

[1.2 Test Approach 140](#_Toc427025772)

[2. Database Relationship Diagram 140](#_Toc427025773)

[2.1 Physical Diagram 140](#_Toc427025774)

[2.2 Data Dictionary 140](#_Toc427025775)

[3. Test Plan 141](#_Toc427025776)

[3.1 Test items 141](#_Toc427025777)

[3.2 Features to be tested 141](#_Toc427025778)

[3.3 Features not to be tested 141](#_Toc427025779)

[3.4 Environmental needs 141](#_Toc427025780)

[3.5 Test case pass/fail criteria 141](#_Toc427025781)

[4. Integration Test Specifications 142](#_Toc427025782)

[4.1 “Background Color Subtraction” Test 142](#_Toc427025783)

[4.2 “Hand Sign Language Recognition” Test 148](#_Toc427025784)

[4.3 “Learning Hand Sign Language” Test 154](#_Toc427025785)

[4.4 “Monitor Battery Capacity” Test 160](#_Toc427025786)

[4.5 “Charging Battery” Test 163](#_Toc427025787)

# List of Tables

[Table 1: Roles and Responsibilities 12](#_Toc427032737)

[Table 2: Roles and Responsibilities Details 19](#_Toc427032738)

[Table 3: Product Backlog Details 20](#_Toc427032739)

[Table 4: Sprint 1 Backlog Details 22](#_Toc427032740)

[Table 5: Sprint 2 Backlog Details 24](#_Toc427032741)

[Table 6: Sprint 3 Backlog Details 26](#_Toc427032742)

[Table 7: Sprint 4 Backlog Details 28](#_Toc427032743)

[Table 8: Sprint 5 Backlog Details 29](#_Toc427032744)

[Table 9: Definitions, Acronyms, and Abbreviations 31](#_Toc427032745)

[Table 10: Scrum Meeting Minutes Detail 39](#_Toc427032746)

# List of Figure

[Figure 1: Components of the the system 15](#_Toc427032976)

[Figure 2 : Scrum Development Model 17](#_Toc427032977)

[Figure 3: Chart of Sprint Backlog 30](#_Toc427032978)

[Figure 4: System Overview Use Case 42](#_Toc427032979)

[Figure 5: Recognize Hand Sign Language use case diagram 43](#_Toc427032980)

[Figure 6: Recognize Hand Sign Language use case diagram 46](#_Toc427032981)

[Figure 7: Charge Battery use case diagram 52](#_Toc427032982)

[Figure 8: Notify Battery Capacity use case diagram 52](#_Toc427032983)

[Figure 9: VSLR System Architectural 56](#_Toc427032984)

[Figure 10: Component Diagram 57](#_Toc427032985)

[Figure 12: Background color subtraction Activity Diagram 86](#_Toc427032986)

[Figure 13: Recognize Hand Sign Language Activity Diagram 87](#_Toc427032987)

[Figure 14: Learn Hand Sing Language Activity Diagram 88](#_Toc427032988)

[Figure 15: Notify Battery Capacity Activity Diagram 89](#_Toc427032989)

[Figure 16: Charge Battery Activity Diagram 90](#_Toc427032990)

[Figure 17: Subtract Background Color Sequence Diagram 91](#_Toc427032991)

[Figure 18: Recognize Hand Sign Language Sequence Diagram 92](#_Toc427032992)

[Figure 19 : Learn Hand Sign Language Sequence Diagram 93](#_Toc427032993)

[Figure 20: Notify Battery Capacity Sequence Diagram 94](#_Toc427032994)

[Figure 21: Charge Battery Sequence Diagram 95](#_Toc427032995)

[Figure 22: Subtract Background Color 96](#_Toc427032996)

[Figure 23: Recognize Hand Sign Language 97](#_Toc427032997)

[Figure 24: Learn Hand Sign Language 98](#_Toc427032998)

[Figure 25: Select 99](#_Toc427032999)

[Figure 26: Raspberry Pi B2 Kit 100](#_Toc427033000)

[Figure 27: Lipo Battery 101](#_Toc427033001)

[Figure 28: LM2576ADJ - 3A UNI REG Board 101](#_Toc427033002)

[Figure 29 : ??? 102](#_Toc427033003)

[Figure 30 : IC LM2576HV-ADJ 102](#_Toc427033004)

[Figure 31 : LT084 103](#_Toc427033005)

[Figure 32 : ??? 104](#_Toc427033006)

[Figure 33 : ??? 104](#_Toc427033007)

[Figure 34 : ??? 105](#_Toc427033008)

[Figure 35 : ??? 105](#_Toc427033009)

[Figure 36 : ??? 106](#_Toc427033010)

[Figure 37 : Connecting Components 107](#_Toc427033011)

[Figure 38 : ??? 108](#_Toc427033012)

[Figure 39 : ??? 109](#_Toc427033013)

[Figure 40 : The first goal of background subtraction algorithm 110](#_Toc427033014)

[Figure 41 : The second goal of background subtraction algorithm 110](#_Toc427033015)

[Figure 42 : Three spaces of RGB image 113](#_Toc427033016)

[Figure 43 : Three-dimensional model of LAB image 115](#_Toc427033017)

[Figure 44 : Example of binary image 115](#_Toc427033018)

[Figure 45 : Examle of contour 116](#_Toc427033019)

[Figure 46 : kernel size 3x3 116](#_Toc427033020)

[Figure 47 : kernel size 5x5 117](#_Toc427033021)

[Figure 48 : kernel size 3x5 117](#_Toc427033022)

[Figure 49 : Kernel size movements 117](#_Toc427033023)

[Figure 50 : Value matrix taken from the kernel size 3x3 117](#_Toc427033024)

[Figure 51 : Value matrix after applying Gaussian blur algorithm 118](#_Toc427033025)

[Figure 52 : Original RGB Image 118](#_Toc427033026)

[Figure 53 : RGB Image applied Guassian Blur with kernel size 7x7 118](#_Toc427033027)

[Figure 54 : RGB Image applied Guassian Blur with kernel size 21x21 118](#_Toc427033028)

[Figure 55 : the coordinate of the center point with matrix 3x3 119](#_Toc427033029)

[Figure 56 : the weighted 3x3 matrix before divide the sum of the weight 119](#_Toc427033030)

[Figure 57 : the final weighted 3x3 matrix 119](#_Toc427033031)

[Figure 58 : The 3x3 matrix value of grayscale image 120](#_Toc427033032)

[Figure 59 : Multiplying the matrix value with the weighted matrix 120](#_Toc427033033)

[Figure 60 : Matrix values after applying Gaussian blur 120](#_Toc427033034)

[Figure 61 : Example of converting RGB image to LAB image 121](#_Toc427033035)

[Figure 62 : Example of Morphological operations – Opening 122](#_Toc427033036)

[Figure 63 : Illustration of the dilation process 123](#_Toc427033037)

[Figure 64 : Illustration of the erosion process 124](#_Toc427033038)

[Figure 65 : starting pixel of find contour 125](#_Toc427033039)

[Figure 66 : demonstrating the path to P1 125](#_Toc427033040)

[Figure 67 : demonstrating the path to P2 126](#_Toc427033041)

[Figure 68 : demonstrating the path to P3 126](#_Toc427033042)

[Figure 69 : flow of background subtraction algorithm 127](#_Toc427033043)

[Figure 70 : The goal of features extraction 127](#_Toc427033044)

[Figure 71 : example of convex hull 129](#_Toc427033045)

[Figure 72 : Example of 4x4 zones 130](#_Toc427033046)

[Figure 73 : RGB image containing hand with black background 130](#_Toc427033047)

[Figure 74 : binary image after converting to binary image by adaptive gaussian 131](#_Toc427033048)

[Figure 75 : example of the steps finding convex hull 131](#_Toc427033049)

[Figure 76 : example of the steps finding convex hull 132](#_Toc427033050)

[Figure 77 : example of finding the convexity defects 133](#_Toc427033051)

[Figure 78 : example of finding the convexity defects 133](#_Toc427033052)

[Figure 79 : Illustration of the resizing 200% 134](#_Toc427033053)

[Figure 80 : Example of Image Scaling 134](#_Toc427033054)

[Figure 81 : The example of degree features extraction 135](#_Toc427033055)

[Figure 82 : The example of degree features extraction 136](#_Toc427033056)

[Figure 83 : the flow of features extraction algorithm 136](#_Toc427033057)

[Figure 84 : Example of linear SVM technique 137](#_Toc427033058)

[Figure 85 : Example of linear SVM technique 138](#_Toc427033059)

[Figure 86 : Example of non-linear SVM technique 139](#_Toc427033060)

[Figure 87 : Example of non-linear SVM technique 139](#_Toc427033061)

[Figure 88 : Black-box testing 140](#_Toc427033062)

[Figure 89 : Physical Database Diagram 140](#_Toc427033063)

[Figure 90 : Components of the Background Color Subtraction 142](#_Toc427033064)

[Figure 91 : Components of the Hand Sign Language Recognition 148](#_Toc427033065)

[Figure 92: Components of the Learning Hand Sign Language 154](#_Toc427033066)

[Figure 93: Components of the Monitoring Battery Capacity 160](#_Toc427033067)

[Figure 94: Components of the charging battery 163](#_Toc427033068)

# Definitions, Acronyms, and Abbreviations

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

# Report No. 1 Introduction

## Project Information

* Project name: **Vietnamese Sign Language Recognition**
* Project Code: **VSLR**
* Product Type: **Embedded system**
* Start Date: **May 11th, 2015**
* End Date:

## Introduction

Nowadays, the communication is the way people can understand each other, is the way people can express their ideas, their thoughts to others. As we know, speaking is the most common way to communicate in life. However, to dumb person, they still need to communicate with others so they have a different way to expose themselves, it is called hand sign language or dumb language.

In this project, we want to develop a device that can help dumb person communicate with not only another mute but also everyone. The device can capture hand signs and then recognize them into text or sound with the same meaning.

## Current Situation

When you want to talk to a dumb person or when a mute wants to present his / her ideas, presentations in a meeting but you are not able to get their signs. Furthermore, when two dumb persons talk to each other but they are from different countries, they have distinct hand sign language, which way can they understand each other? Obviously, there are some ways, they can write out what they want or they can use some signs that are familiar to the daily life, and they can even hire a translator to interpret.

## Problem Definition

*The following disadvantages of current situation:*

* Handwritten: Time consuming to write out all content is very high.
* Using familiar signs: Without time consuming, the accuracy of the content is not high.
* Hand sign language translator can not respond the instant needs of communication. Moreover, the price for hiring a translator is very costly.

## Proposed Solution

To meet the needs of users we offer a solution based on translating hand signs into content and then show them.

Our system is a small device with a camera to capture hand signs and then translate them.

*In more detail, our system has the following functions:*

### **Feature functions**

* The system detects your hands, keeps track them and then analyzes the captured images into content.
* Showing the translated content for users on text and sound.
* Learning sign language hand for people who want to know about the language in order to better communicate with dumb people.

### Advantages and disadvantages

*The advantages and disadvantages of the proposed solution:*

* Advantages:
  + Quick and easy communicate for dumb person.
  + Train for person who don’t know about mute language.
  + Standardized for hand sign language.
  + People get used to the dumb language easily.
* Disadvantages:
  + In some cases, this solution does not work really exactly with the hands have weird characterize.
  + This solution needs stable environment (light, background) and some accessories.
  + This solution can not solve the problem about hand motion language.

## Functional Requirements

*Function requirements of the system are listed as below:*

### Tracking hand

* Allow users can move the hand in range area but the system still works correctly.

### Hand recognition

* The system analyzes the images which is captured by camera, then detects and recognizes the hand sign on these images into content.

### Showing the content

* The translated content is shown not only on text but also on sound.

### Learning hand sign

* Users select and learn words existed in the system with images express the hand gesture.

### Controlling System

* Allow users can turn on / off the system by the power button.
* Users can select functions by hand signs.
* Users can perform operations of function by hand signs.

### Controlling power

* System uses battery power gives users more flexibility in using.
* Combining with controlling the battery capacity that helps users to use the most effective.

## Role and Responsibility

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| No | Full Name | Role | Position | Contact |
| 1 | Đỗ Đức Minh Quân | Scrum Master/Product Owner | Instructor | [m](mailto:Khanhkt@fpt.edu.vn)inhquandd@fpt.edu.vn |
| 2 | Nguyễn Hữu Kỳ Long | Developer | Leader | [longnhkse60984@fpt.edu.vn](mailto:longnhkse60984@fpt.edu.vn) |
| 3 | Nguyễn Đình Tân | Developer | Member | [tanndse61115@fpt.edu.vn](mailto:tanndse61115@fpt.edu.vn) |
| 4 | Nguyễn Xuân Ý | Developer | Member | [ynxse60896@fpt.edu.vn](mailto:ynxse60896@fpt.edu.vn) |
| 5 | Lê Phương Bình | Developer | Member | [binhlpse61049@fpt.edu.vn](mailto:binhlpse61049@fpt.edu.vn) |

Table : Roles and Responsibilities

# 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. Nevertheless, these solutions just solve the problem at that time and 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 ways 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 an 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 supported 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:
* The system can be implemented on many different platforms.
* Operating costs are less expensive.
* Recognition is implemented quickly by many image-processing algorithms.
* Disadvantages:
* Analyzing image still remains restriction on process environment, point of view.
* Recognition has still not covered every case 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

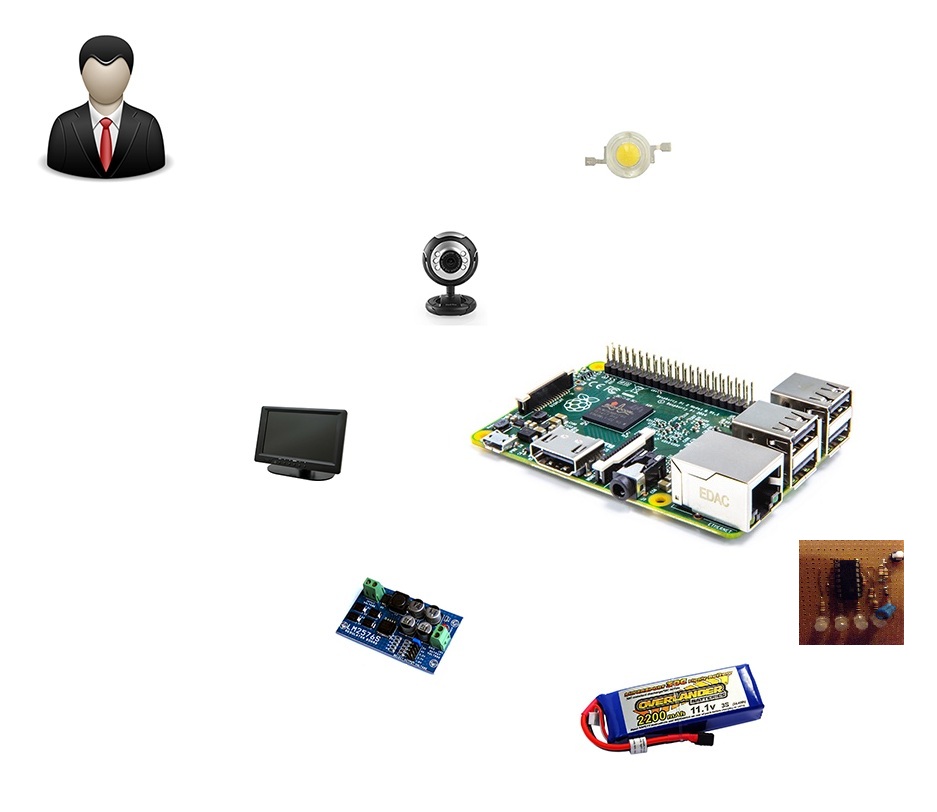
* Users can choose words that they want to learn which existed in the system.
* Users can see images which express the hand gesture.
* User's hand signs can be practiced and checked by following some steps of the system.
* Users receive the current recognized result of the hand sign via text or sound.

#### Boundaries of the System

##### The restrictions

* The system language is Vietnamese.
* Hand sign language the system supports is Vietnamese sign language.
* The system just recognizes no motion hand signs.
* 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: Components 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 similar 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 interface of functions and the recognized results.
    - 2 Led (1W): is used to balance light.
    - LM2576ADJ-Board: UNI Regulator Board.
    - LT084 + zener 5.1v is used to monitor battery capacity.
    - XL6009 DC-DC Voltage Boost Module is used for increasing voltage.

##### Software requirements

* + - Operating system and platform for deployment and development: Ubuntu 14.04 for laptop and Raspbian for Raspberry PI.
    - Remote Desktop: application for remoting to work on raspberry.
    - QT 5.4 Creator: is to develop C++ application and Linux GUI.
    - OpenCV 2.4.9 library: supporting image processing.
    - SQLite 3: 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 2: Roles and Responsibilities Details

### Tools and Techniques

* + - Front-end and back-end IDE:
* QT 5.4 Creator
  + - Front-end technology:
* QT 5.4 Linux GUI
  + - Back-end library:
* OPENCV 2.4.9 library
* LIBSVM 3.20 library
* Espeak 1.48.04 library
  + - Managing database:
* SQLite 3
  + - Connecting to Raspberry PI 2:
* Remote Desktop Connection Program of Ubuntu 14.04
  + - Managing the project:
* SVNtortoise version 1.8.11
* 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 | Medium | 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 | Medium | 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 | High | 4 |
| 9 | Device | User | turn on/off the system | can turn on/off the device according to the demand | Medium | 4 |
| 10 | Device | User | the system is boxed firm, compact | the component are protected against bumps | Medium | 5 |
| 11 | Recognition | User | the system reliable operation | no error occurs when using | High | 5 |
| 12 | User manual | User | know how to install and use the system | easy to use, repair | High | 5 |

Table 3: Product Backlog Details

### Sprint Backlog

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Sprint** | **User Story** | **Time est** | **Time spent** | **Time**  **left** | **Task** | **Time (est)** | **Who** | **Status** | **Work Done per day per task [Hours]** | | | | | | | | | | | | | | |
| **Week 1** | | | | | **Week 2** | | | | | **Week 3** | | | | |
| *Project Name: "Vietnamese Language Sign Recognition" Started: 12-05-2015* | | | | | | | | **1** | **2** | **3** | **4** | **5** | **6** | **7** | **8** | **9** | **10** | **11** | **12** | **13** | **14** | **15** |
| 1 | As a user, I want to keep track my hand gesture | 75 | 75 | 0 | Setup Ubuntu 14.04.2 LTS Operating System for laptops | **5** | All Team | Done | 5 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Setup Ubuntu Debian Operating System for raspberry | **5** | All Team | Done |  | 5 |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Setup necessary software for laptop | **5** | All Team | Done |  |  | 5 |  |  |  |  |  |  |  |  |  |  |  |  |
| Setup necessary software for raspberry | **5** | All Team | Done |  |  |  | 5 |  |  |  |  |  |  |  |  |  |  |  |
| Writing document introduce about project | **5** | All Team | Done |  |  |  |  | 5 |  |  |  |  |  |  |  |  |  |  |
| Capture images from camera | **5** | TanND + BinhLP | Done |  |  |  |  |  | 5 |  |  |  |  |  |  |  |  |  |
| Save image to directory | **5** | TanND + BinhLP | Done |  |  |  |  |  |  | 5 |  |  |  |  |  |  |  |  |
| Showing image to user | **5** | TanND + BinhLP | Done |  |  |  |  |  |  |  | 5 |  |  |  |  |  |  |  |
| Writing document about Software Project Management Plan | **10** | All Team | Done |  |  |  |  |  |  |  |  | 5 | 5 |  |  |  |  |  |
| Convert BGR background sample image to LAB sample image | **5** | LongNHK | Done |  |  |  |  |  | 5 |  |  |  |  |  |  |  |  |  |
| Define color range of LAB sample color | **5** | LongNHK | Done |  |  |  |  |  |  | 5 |  |  |  |  |  |  |  |  |
| Create a binary image with black background | **5** | LongNHK | Done |  |  |  |  |  |  |  | 5 |  |  |  |  |  |  |  |
| Remove the color pixels is between color range: | **5** | LongNHK | Done |  |  |  |  |  |  |  |  |  |  | 5 |  |  |  |  |
| Remove noises and smooth the contou | **5** | LongNHK | Done |  |  |  |  |  |  |  |  |  |  |  | 5 |  |  |  |
| Find the hand in binary image | **10** | LongNHK | Done |  |  |  |  |  |  |  |  |  |  |  |  | 5 | 5 |  |
| Testing function extracting color, subtracting color | **5** | LongNHK | Done |  |  |  |  |  |  |  |  |  |  |  |  |  |  | 5 |
| Finding the contours on the binary image | **5** | TanND | Done |  |  |  |  |  |  |  |  |  |  | 5 |  |  |  |  |
| Draws contours outlines | **5** | TanND | Done |  |  |  |  |  |  |  |  |  |  |  | 5 |  |  |  |
| Testing function | **5** | TanND | Done |  |  |  |  |  |  |  |  |  |  |  |  | 5 |  |  |
| As a user, I want the system is a portable system | 75 | 75 | 0 | Learning about the power supply for devices | **5** | YNX | Done |  |  |  |  |  | 5 |  |  |  |  |  |  |  |  |  |
| Learning about the types of batteries | **2** | YNX | Done |  |  |  |  |  |  | 2 |  |  |  |  |  |  |  |  |
| Try different types of batteries | **3** | YNX | Done |  |  |  |  |  |  | 3 |  |  |  |  |  |  |  |  |
| Learning about the voltage regulator circuit | **5** | YNX | Done |  |  |  |  |  |  |  | 3 |  |  |  |  |  |  |  |
| Finding suitable type voltage regulator | **5** | BinhLP + YNX | Done |  |  |  |  |  |  |  |  | 5 |  |  |  |  |  |  |
| Check the voltage regulator | **5** | BinhLP + YNX | Done |  |  |  |  |  |  |  |  |  | 5 |  |  |  |  |  |
| Do testing circuit components | **5** | BinhLP + YNX | Done |  |  |  |  |  |  |  |  |  |  | 5 |  |  |  |  |
| Devices running on the test circuit to test stability. | **5** | BinhLP + YNX | Done |  |  |  |  |  |  |  |  |  |  |  | 5 |  |  |  |
| Welding the components together. | **5** | TanND + YNX | Done |  |  |  |  |  |  |  |  |  |  |  |  |  | 5 |  |
| Testing system after welding the components together. | **5** | TanND + BinhLP | Done |  |  |  |  |  |  |  |  |  |  |  |  |  |  | 5 |

Table 4: Sprint 1 Backlog Details

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Sprint** | **User Story** | **Time est** | **Time**  **spent** | **Time (left)** | **Task** | **Time (est)** | **Who's working** | **Status** | **Work Done per day per task [Hours]** | | | | | | | | | | | | |
| **Week 4** | | | | | **Week 5** | | | | | **Week 6** | | |
| *Project Name: "Vietnamese Language Sign Recognition" Started: 12-05-2015* | | | | | | | | **1** | **2** | **3** | **4** | **5** | **6** | **7** | **8** | **9** | **10** | **11** | **12** | **13** |
| 2 |  | 65 | 65 | 65 | Capture hand sign to create database | **5** | All Team | Done | 5 |  |  |  |  |  |  |  |  |  |  |  |  |
| Writing document about Software Requirement Specification | **15** | All Team | Done |  | 5 | 5 | 5 |  |  |  |  |  |  |  |  |  |
| Setup SQLite on laptop, raspberry | **2** | TanND | Done |  |  |  |  | 2 |  |  |  |  |  |  |  |  |
| Design database | **4** | TanND | Done |  |  |  |  | 3 | 1 |  |  |  |  |  |  |  |
| Input data for database | **4** | TanND | Done |  |  |  |  |  | 4 |  |  |  |  |  |  |  |
| Convert the BGR image containing the hand to binary image containing features | **5** | LongNHK | Done |  |  |  |  | 5 |  |  |  |  |  |  |  |  |
| Produce the binary image containing the hand palm from the binary image containing contour | **5** | LongNHK | Done |  |  |  |  |  | 5 |  |  |  |  |  |  |  |
| Produce the binary image containing the finger lines from the binary image containing contour | **5** | LongNHK | Done |  |  |  |  |  |  | 5 |  |  |  |  |  |  |
| Defining features to recognize | **5** | LongNHK | Done |  |  |  |  |  |  |  | 5 |  |  |  |  |  |
| Calculate the features from these binary images | **5** | LongNHK | Done |  |  |  |  |  |  |  |  | 5 |  |  |  |  |
| Training SVM | **5** | LongNHK | Done |  |  |  |  |  |  |  |  |  | 5 |  |  |  |
| Recognizing the hand sign by SVM | **10** | LongNHK | Done |  |  |  |  |  |  |  |  |  |  | 5 | 5 |  |
| Testing function recognize | **5** | LongNHK | Done |  |  |  |  |  |  |  |  |  |  |  |  | 5 |
| Update database | **5** | TanND + YNX | Done |  |  |  |  |  | 5 |  |  |  |  |  |  |  |
| Defining the meaning word of the SVM result from database | **5** | TanND + YNX | Done |  |  |  |  |  |  | 5 |  |  |  |  |  |  |
| Designing QT Linux GU | **3** | BinhLP + YNX | Done |  |  |  |  | 3 |  |  |  |  |  |  |  |  |
| Testing design when display in LCD | **2** | BinhLP + YNX | Done |  |  |  |  | 2 |  |  |  |  |  |  |  |  |
|  | 50 | 50 | 0 | Learn some text to speech opensource | **5** | BinhLP | Done |  |  |  |  |  | 5 |  |  |  |  |  |  |  |
| Testing and compare some text to speech opensource | **5** | BinhLP | Done |  |  |  |  |  |  | 5 |  |  |  |  |  |  |
| Setup espeak software for laptop,  raspberry | **5** | BinhLP + YNX | Done |  |  |  |  |  |  |  |  |  | 5 |  |  |  |
| Changing pronunciation of alphabet | **10** | TanND + BinhLP | Done |  |  |  |  |  |  |  | 5 | 5 |  |  |  |  |
| Showing the recognition result of each alphabet | **10** | YNX | Done |  |  |  |  |  |  |  | 5 | 5 |  |  |  |  |
| Join recognition result become syllable and a complete word | **5** | TanND + YNX | Done |  |  |  |  |  |  |  |  |  |  | 5 |  |  |
| Showing the recognition result via sound | **10** | TanND + BinhLP | Done |  |  |  |  |  |  |  |  |  |  |  | 5 | 5 |
| Testing, changing pronunciation of syllable | **10** | YNX | Done |  |  |  |  |  |  |  |  |  |  |  | 5 | 5 |

Table 5: Sprint 2 Backlog Details

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Sprint** | **User Story** | **Time (est)** | **Time (spent)** | **Time (left)** | **Task** | **Time (est)** | **Who's working** | **Status** | **Work Done per day per task [Hours]** | | | | | | | | | | | | | | | |
| **Week6** | | **Week 7** | | | | | | **Week 8** | | | | | **Week 9** | | |
| *Project Name: "Vietnamese Language Sign Recognition" Started: 12-05-2015* | | | | | | | | **1** | **2** | | **3** | **4** | **5** | **6** | **7** | **8** | **9** | **10** | **11** | **12** | **13** | **14** | **15** |
| 3 | As a user, I want to control the system functions by hand gesture | 75 | 75 | 0 | Writing Design Description document | **20** | All Team | Done | 5 | 5 | | 5 | 5 |  |  |  |  |  |  |  |  |  |  |  |
| Defining operations of function | **10** | LongNHK + TanND | Done |  |  | |  |  | 5 | 5 |  |  |  |  |  |  |  |  |  |
| Defining hand sign to control the system | **5** | LongNHK + TanND | Done |  |  | |  |  |  |  | 5 |  |  |  |  |  |  |  |  |
| Recognizing the control hand sign | **10** | LongNHK + TanND | Done |  |  | |  |  |  |  |  | 5 | 5 |  |  |  |  |  |  |
| Implementing operations of function | **10** | LongNHK + TanND | Done |  |  | |  |  |  |  |  |  |  | 5 | 5 |  |  |  |  |
| Designing QT Linux GUI | **5** | LongNHK + TanND | Done |  |  | |  |  |  |  |  |  |  |  |  | 5 |  |  |  |
| Testing GUI on LCD , change GUI | **5** | TanND | Done |  |  | |  |  |  |  |  |  |  |  |  |  |  |  | 5 |
| Synchronize between hand gestures and operations of function | **10** | LongNHK + TanND | Done |  |  | |  |  |  |  |  |  |  |  |  |  | 5 | 5 |  |
| Testing control the system function by hand gesture | **5** | LongNHK | Done |  |  | |  |  |  |  |  |  |  |  |  |  |  |  | 5 |
| As a user, I want to know remaining of battery capacity | 75 | 75 | 0 | Learning about the types of chip voltage comparator | **5** | BinhLP + YNX | Done |  |  | |  |  | 5 |  |  |  |  |  |  |  |  |  |  |
| Choosing chip voltage comparator | **5** | BinhLP + YNX | Done |  |  | |  |  |  | 5 |  |  |  |  |  |  |  |  |  |
| Learning about the types of zener | **5** | BinhLP + YNX | Done |  |  | |  |  |  |  | 5 |  |  |  |  |  |  |  |  |
| Circuit design | **5** | BinhLP + YNX | Done |  |  | |  |  |  |  |  | 5 |  |  |  |  |  |  |  |
| Circuit tested by software proteus | **5** | BinhLP + YNX | Done |  |  | |  |  |  |  |  |  | 5 |  |  |  |  |  |  |
| Constructing circuit | **10** | BinhLP + YNX | Done |  |  | |  |  |  |  |  |  |  | 5 | 5 |  |  |  |  |
| Testing circuit with battery | **5** | BinhLP + YNX | Done |  |  | |  |  |  |  |  |  |  |  |  | 5 |  |  |  |
| Adjusting the device components | **5** | BinhLP + YNX | Done |  |  | |  |  |  |  |  |  |  |  |  |  | 5 |  |  |
| Connecting to the system | **5** | BinhLP + YNX | Done |  |  | |  |  |  |  |  |  |  |  |  |  |  | 5 |  |
| Run system to check stability with this circuit. | **5** | BinhLP + YNX | Done |  |  | |  |  |  |  |  |  |  |  |  |  |  |  | 5 |

Table 6: Sprint 3 Backlog Details

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Sprint** | **User Story** | **Time est** | **Time spent** | **Time left** | **Task** | **Time est** | **Who's working** | **Status** | **Work Done per day per task [Hours]** | | | | | | | | | | | | | | | | |
| **Week 9** | | **Week 10** | | | | | **Week 11** | | | | | **Week 12** | | | | |
| *Project Name: "Vietnamese Language Sign Recognition" Started: 12-05-2015* | | | | | | | | **1** | **2** | **3** | **4** | **5** | **6** | **7** | **8** | **9** | **10** | **11** | **12** | **13** | **14** | **15** | **16** | **17** |
| 4 | As a user, I want to increase the accuracy of the recognition result | 45 | 45 | 45 | Defining more features to recognize | **10** | LongNHK + TanND | Done | 5 | 5 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Extracting features | **10** | LongNHK + TanND | Done |  |  | 5 | 5 |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Training SVM | **5** | LongNHK + TanND | Done |  |  |  |  | 5 |  |  |  |  |  |  |  |  |  |  |  |  |
| Providing two LEDs to balance light | 10 | BinhLP + YNX | Done | 5 | 5 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Providing background | 5 | BinhLP + YNX | Done |  |  | 5 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Improving camera setting | 10 | BinhLP + YNX | Done |  |  |  | 5 | 5 |  |  |  |  |  |  |  |  |  |  |  |  |
| Improving background - hand color subtraction | 5 | LongNHK + TanND | Done |  |  |  |  |  | 5 |  |  |  |  |  |  |  |  |  |  |  |
| Capturing new database | **5** | All Team | Done |  |  |  |  |  |  | 5 |  |  |  |  |  |  |  |  |  |  |
| Testing recognition hand sign | **5** | BinhLP + YNX | Done |  |  |  |  |  |  |  | 5 |  |  |  |  |  |  |  |  |  |
| Writing document System Implementation & Test | **15** | All Team | Done |  |  |  |  |  |  |  |  |  |  |  |  |  |  | 5 | 5 | 5 |
| As a user, I want to learn hand sign language | 40 | 40 | 0 | Defining operations of function | **10** | LongNHK + TanND | Done |  |  |  |  |  |  |  | 5 | 5 |  |  |  |  |  |  |  |  |
| Creating database | **5** | LongNHK + TanND | Done |  |  |  |  |  |  |  |  |  | 5 |  |  |  |  |  |  |  |
| Managing database | **10** | TanND | Done |  |  |  |  |  |  |  |  |  |  | 5 | 5 |  |  |  |  |  |
| Implementing the practice function of learning | **15** | LongNHK | Done |  |  |  |  |  |  |  |  |  |  | 5 | 5 | 5 |  |  |  |  |
| Testing function learn hand sign | **5** | LongNHK + BinhLP | Done |  |  |  |  |  |  |  |  |  |  |  |  |  | 5 |  |  |  |
| Designing GUI | **5** | TanND + YNX | Done |  |  |  |  |  |  |  |  |  |  |  |  | 5 |  |  |  |  |
| Testing GUI in LCD | **5** | TanND | Done |  |  |  |  |  |  |  |  |  |  |  |  |  | 5 |  |  |  |
| As a user, I want to turn on/off the system | 35 | 35 | 0 | Learning about the types of switch button | **5** | BinhLP + YNX | Done |  |  |  |  |  | 5 |  |  |  |  |  |  |  |  |  |  |  |
| Choosing switch button | **5** | BinhLP + YNX | Done |  |  |  |  |  |  | 5 |  |  |  |  |  |  |  |  |  |  |
| Testing system with switch button | **5** | BinhLP + YNX | Done |  |  |  |  |  |  |  | 5 |  |  |  |  |  |  |  |  |  |
| Constructing circuit with switch button | **10** | BinhLP + YNX | Done |  |  |  |  |  |  |  |  | 5 | 5 |  |  |  |  |  |  |  |
| Run system to check stability with switch button. | **5** | BinhLP + YNX | Done |  |  |  |  |  |  |  |  |  |  | 5 |  |  |  |  |  |  |
| Adjusting the device components | **5** | BinhLP + YNX | Done |  |  |  |  |  |  |  |  |  |  |  | 5 |  |  |  |  |  |

Table 7: Sprint 4 Backlog Details

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Sprint** | **User Story** | **Time est** | **Time spent** | **Time left** | **Task** | **Time**  **est** | **Who's working** | **Status** | **Work Done per day per task [Hours]** | | | | | | | | | |
| **Week 13** | | | | | **Week 14** | | | | |
| *Project Name: "Vietnamese Language Sign Recognition" Started: 12-05-2015* | | | | | | | | **1** | **2** | **3** | **4** | **5** | **6** | **7** | **8** | **9** | **10** |
| 5 | As a user, I want to the system is boxed firm, compact | 20 | 20 | 0 | Measuring, designing the box for system | **5** | LongNHK + YNX |  | 5 |  |  |  |  |  |  |  |  |  |
| Compare, choose materials for box | **5** | TanND + BinhLP |  | 5 |  |  |  |  |  |  |  |  |  |
| Constructing box for system | **10** | All Team |  |  | 5 | 5 |  |  |  |  |  |  |  |
| Assembling of component | **5** | All Team |  |  |  |  | 5 |  |  |  |  |  |  |
| As a user, I want the system reliable operation | 15 | 15 | 15 | Perform system testing | **10** | All Team |  |  |  |  |  | 5 | 5 |  |  |  |  |
| Fix some recognition bug | **5** | LongNHK + YNX |  |  |  |  |  |  |  | 5 |  |  |  |
| Fix pronunciation of some word | **5** | TanND + BinhLP |  |  |  |  |  |  |  | 5 |  |  |  |
| As a user, I want to know how to install and use the system | 15 | 15 | 15 | Writing install guide | **3** | All Team |  |  |  |  |  |  |  |  | 3 |  |  |
| Writing recognize hand sign manual | **3** | All Team |  |  |  |  |  |  |  |  | 2 | 1 |  |
| Writing learn hand sign manual | **3** | All Team |  |  |  |  |  |  |  |  |  | 3 |  |
| Writing some solutions when system has problems | **3** | All Team |  |  |  |  |  |  |  |  |  | 1 | 2 |
| Writing complete user manual document | **3** | All Team |  |  |  |  |  |  |  |  |  |  | 3 |

Table 8: Sprint 5 Backlog Details



### All Meeting Minutes

|  |  |
| --- | --- |
| **Name** | **Definition** |
| x | Selected Person |
| VH | Very High Priority |
| H | High Priority |
| M | Medium Priority |
| A | Approved |
| Y | Yes |
| N | No |

Table 9: Definitions, Acronyms, and Abbreviations

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Scrum Meeting Minutes** | | | | | Target Personas | | | | Status | | Approval | |
| LongNHK | TanND | BinhLP | YNX | Degree of priority | Task completion | LongNHK | Mr. QuanDDM |
|
|
|
|
|
| **Date: 12/05/2015** | | | | |  | | | |  | |  | |
| *Raspberry PI 2* | | | | |  | | | |  | |  |  |
| Development environment for Raspberry | | | | | x | x | x | x | H | Y | A | A |
| *Report 1* | | | | |  | | | |  | | | |
| Orienting the way writing report 1 | | | | | x | x | x | x | VH | Y | A | A |
| **Date: 16/05/2015** | | | | |  | | | |  | |  | |
| *Hand Detection* | | | | |  | | | |  | |  |  |
| Extracting background color | | | | | x |  |  |  | VH | Y | A | A |
| Capturing images from camera and showing them to user | | | | |  | x |  |  | VH | Y | A | A |
| *Portable System* | | | | |  | | | |  | |  |  |
| Choosing type of battery | | | | |  |  |  | x | H | Y | A | A |
| Choosing voltage regulator circuit | | | | |  |  | x |  | H | Y | A | A |
| *Report 1* | | | | |  | | | |  | | | |
| Reviewing report 1 | | | | | x | x | x | x | H | Y | A | A |
| **Date: 19/05/2015** | | | | |  | | | |  | | | |
| *Hand Detection* | | | | |  | | | |  | |  |  |
| Demo extracting background color | | | | | x |  |  |  | H | Y | A | A |
| Demo capturing images from camera and showing them to user | | | | |  | x |  |  | H | Y | A | A |
| *Portable System* | | | | |  | | | |  | |  |  |
| Reviewing battery | | | | | x | x | x | x | M | Y | A | A |
| Reviewing voltage regulator circuit | | |  |  | x | x | x | x | M | Y | A | A |
| *Report 2* | | | | |  | | | |  | | | |
| Orienting the way writing report 2 | | | | | x | x | x | x | VH | Y | A | A |
| **Date: 23/05/2015** | | | | |  | | | |  | | | |
| *Hand Detection* | | | | |  | | | |  | |  |  |
| Extracting hand color | | | | | x |  |  |  | VH | Y | A | A |
| Finding the hand contours on the hand binary image | | | | |  | x |  |  | VH | Y | A | A |
| *Portable System* | | | | |  | | | |  | |  |  |
| Connecting components | | | | |  |  | x | x | H | Y | A | A |
| *Report 2* | | | | |  | | | |  | | | |
| Reviewing report 2 | | | | | x | x | x | x | H | Y | A | A |
| **Date: 26/05/2015** | | | | |  | | | |  | | | |
| *Hand Detection* | | | | |  | | | |  | |  |  |
| Extracting hand color | | | | | x |  |  |  | H | Y | A | A |
| Designing QT Linux GUI which of subtracting color | | | | |  | x |  |  | H | Y | A | A |
| *Portable System* | | | | |  | | | |  | |  |  |
| Demo fully made portable system | | |  |  |  |  | x | x | H | Y | A | A |
| *Report 3* | | | | |  | | | |  | | | |
| Orienting the way writing report 3 | | | | | X | x | x | x | VH | Y | A | A |
| **Date: 30/05/2015** | | | | |  | | | |  | | | |
| *Hand Detection* | | | | |  | | | |  | |  |  |
| Subtracting color to get hand binary image | | | | | x |  |  |  | VH | Y | A | A |
| Finding the hand contours on the hand binary image | | | | |  | x |  |  | VH | Y | A | A |
| *Portable System* | | | | |  | | | |  | |  |  |
| Demo adjusted portable system | | | | |  |  | x | x | M | Y | A | A |
| *Report 3* | | | | |  | | | |  | | | |
| Orienting the way writing report 3 | | | | | x | x | x | x | VH | Y | A | A |
| **Date: 02/06/2015** | | | | |  | | | |  | | | |
| *Hand Detection* | | | | |  | | | |  | |  |  |
| Training SVM | | | | | x |  |  |  | VH | Y | A | A |
| Creating SQLite database | | |  |  |  | x |  |  | VH | Y | A | A |
| Designing QT Linux GUI which of steps recognition | | | | |  |  | x | x | M | Y | A | A |
| *Report 3* | | | | |  | | | |  | | | |
| Reviewing report 3 | | | | | x | x | x | x | VH | Y | A | A |
| **Date: 06/06/2015** | | | | |  | | | |  | | | |
| *Hand Detection* | | | | |  | | | |  | |  |  |
| Defining features to recognize | | | | | x |  |  |  | VH | Y | A | A |
| Defining the meaning word of the SVM result from database | | | | |  | x |  |  | VH | Y | A | A |
| *Portable System* | | | | |  | | | |  | |  |  |
| Designing QT Linux GUI which of the recognition flow | | | | |  |  | x | x | M | Y | A | A |
| *Report 3* | | | | |  | | | |  | | | |
| Reviewing report 3 | | | | | x | x | x | x | VH | Y | A | A |
| **Date: 09/06/2015** | | | | |  | | | |  | | | |
| *Hand Detection* | | | | |  | | | |  | |  |  |
| Defining features to recognize | | | | | x |  |  |  | VH | Y | A | A |
| Defining the meaning word of the SVM result from database | | | | |  |  |  | x | M | Y | A | A |
| *Result via sound* | | | | |  | | | |  | |  |  |
| Choosing text to speech opensource | | | |  |  | x | x |  | H | Y | A | A |
| *Report 4* | | | | |  | | | |  | | | |
| Orienting the way writing report 4 | | | | | x | x | x | x | VH | Y | A | A |
| **Date: 13/06/2015** | | | | |  | | | |  | | | |
| *Hand Detection* | | | | |  | | | |  | |  |  |
| Features extraction | | | | | x |  |  |  | VH | Y | A | A |
| *Result via sound* | | | | |  | | | |  | |  |  |
| Changing pronunciation of word | | | | |  | x |  | x | H | Y | A | A |
| Implementing text to speech opensource | | | |  |  | x | x |  | M | Y | A | A |
| *Report 4* | | | | |  | | | |  | | | |
| Orienting the way writing report 4 | | | | | x | x | x | x | VH | Y | A | A |
| **Date: 16/06/2015** | | | | |  | | | |  | | | |
| *Hand Detection* | | | | |  | | | |  | |  |  |
| Recognizing the hand sign by SVM | | | | | x |  |  |  | VH | Y | A | A |
| *Result via sound* | | | | |  | | | |  | |  |  |
| Changing pronunciation of word | | | | |  | x |  | x | H | Y | A | A |
| Showing the recognition result via sound | | | |  |  |  | x |  | M | Y | A | A |
| *Report 4* | | | | |  | | | |  | | | |
| Reviewing report 4 | | | | | x | x | x | x | VH | Y | A | A |
| **Date: 20/06/2015** | | | | |  | | | |  | | | |
| *Hand Detection* | | | | |  | | | |  | |  |  |
| Showing the recognition result via text | | | | | x |  |  |  | H | Y | A | A |
| *Result via sound* | | | | |  | | | |  | |  |  |
| Changing pronunciation of word | | | | |  |  |  | x | H | Y | A | A |
| Showing the recognition result via sound | | | |  |  |  | x |  | M | Y | A | A |
| *Report 4* | | | | |  | | | |  | | | |
| Reviewing report 4 | | | | | x | x | x | x | H | Y | A | A |
| **Date: 30/06/2015** | | | | |  | | | |  | | | |
| *Controlling function* | | | | |  | | | |  | |  |  |
| Defining operations of function | | | | | x | x |  |  | VH | Y | A | A |
| *Monitor the battery capacity* | | | | |  | | | |  | |  |  |
| Choosing chip voltage comparator | | | | |  |  | x | x | H | Y | A | A |
| *Report 5* | | | | |  | | | |  | | | |
| Orienting the way writing report 5 | | | | | x | x | x | x | VH | Y | A | A |
| **Date: 04/07/2015** | | | | |  | | | |  | | | |
| *Controlling function* | | | | |  | | | |  | |  |  |
| Implementing operations of function | | | | | x | x |  |  | VH | Y | A | A |
| *Monitor the battery capacity* | | | | |  | | | |  | |  |  |
| Choosing zener | | | | |  |  | x | x | H | Y | A | A |
| *Report 5* | | | | |  | | | |  | | | |
| Orienting the way writing report 5 | | | | | x | x | x | x | VH | Y | A | A |
| **Date: 07/07/2015** | | | | |  | | | |  | | | |
| *Controlling function* | | | | |  | | | |  | |  |  |
| Demo implementing operations of function | | | | | x | x |  |  | H | Y | A | A |
| *Monitor the battery capacity* | | | | |  | | | |  | |  |  |
| Reviewing the chip voltage comparator and zener | | | | | x | x | x | x | VH | Y | A | A |
| *Report 5* | | | | |  | | | |  | | | |
| Orienting the way writing report 5 | | | | | x | x | x | x | VH | Y | A | A |
| **Date: 11/07/2015** | | | | |  | | | |  | | | |
| *Controlling function* | | | | |  | | | |  | |  |  |
| Designing QT Linux GUI which of the flow of operations | | | | | x | x |  |  | H | Y | A | A |
| *Monitor the battery capacity* | | | | |  | | | |  | |  |  |
| Constructing circuit | | | | |  |  | x | x | VH | Y | A | A |
| *Report 5* | | | | |  | | | |  | | | |
| Reviewing report 5 | | | | | x | x | x | x | VH | Y | A | A |
| **Date: 14/07/2015** | | | | |  | | | |  | | | |
| *Controlling function* | | | | |  | | | |  | |  |  |
| Synchronizing between hand gestures and operations of function | | | | | x | x |  |  | VH | Y | A | A |
| *Monitor the battery capacity* | | | | |  | | | |  | |  |  |
| Adjusting the device components | | | | |  |  | x | x | H | Y | A | A |
| *Report 5* | | | | |  | | | |  | | | |
| Reviewing report 5 | | | | | x | x | x | x | VH | Y | A | A |
| **Date: 18/07/2015** | | | | |  | | | |  | | | |
| *Controlling function* | | | | |  | | | |  | |  |  |
| Demo synchronizing between hand gestures and operations of function | | | | | x | x |  |  | M | Y | A | A |
| *Portable System* | | | | |  | | | |  | |  |  |
| Connecting to the system | | | | |  |  | x | x | VH | Y | A | A |
| *Report 5* | | | | |  | | | |  | | | |
| Reviewing report 5 | | | | | x | x | x | x | VH | Y | A | A |
| **Date: 21/07/2015** | | | | |  | | | |  | | | |
| *Hand Detection* | | | | |  | | | |  | |  |  |
| Defining more features to recognize | | | | | x | x | x | x | VH | Y | A | A |
| *Report 5* | | | | |  | | | |  | | | |
| Reviewing report 5 | | | | | x | x | x | x | H | Y | A | A |
| **Date: 25/07/2015** | | | | |  | | | |  | | | |
| *Hand Detection* | | | | |  | | | |  | |  |  |
| Extracting features | | | | | x |  |  |  | VH | Y | A | A |
| Improving camera setting | | |  |  |  | x |  | x | M | Y | A | A |
| *Stable Environment* | | | | |  | | | |  | |  |  |
| Providing two LEDs to balance light | | | | |  |  | x | x | H | Y | A | A |
| *Report 5* | | | | |  | | | |  | | | |
| Reviewing report 5 | | | | | x | x | x | x | H | Y | A | A |
| **Date: 28/07/2015** | | | | |  | | | |  | | | |
| *Hand Detection* | | | | |  | | | |  | |  |  |
| Demo extracting features | | | | | x |  |  |  | H | Y | A | A |
| Improving background - hand color subtraction | | | | | x | x |  |  | VH | Y | A | A |
| Demo improving camera setting | | | | |  | x |  | x | M | Y | A | A |
| *Stable Environment* | | | | |  | | | |  | |  |  |
| Demo balancing light | |  |  |  |  |  | x | x | M | Y | A | A |
| *Report 6* | | | | |  | | | |  | | | |
| Orienting the way writing report 6 | | | | | x | x | x | x | VH | Y | A | A |
| **Date: 01/08/2015** | | | | |  | | | |  | | | |
| *Hand Detection* | | | | |  | | | |  | |  |  |
| Improving background - hand color subtraction | | | | | x | x |  |  | H | Y | A | A |
| *Learning Function* | | | | |  | | | |  | |  |  |
| Creating database | |  |  |  |  | x |  |  | H | Y | A | A |
| Designing GUI | | | | |  | x |  | x | M | Y | A | A |
| Implementing the practice function of learning | | | | | x |  | x |  | M | Y | A | A |
| *Report 6* | | | | |  | | | |  | | | |
| Orienting the way writing report 6 | | | | | x | x | x | x | VH | Y | A | A |
| **Date: 04/08/2015** | | | | |  | | | |  | | | |
| *Hand Detection* | | | | |  | | | |  | |  |  |
| Training SVM | | | | | x |  |  |  | VH | Y | A | A |
| *Learning Function* | | | | |  | | | |  | |  |  |
| Managing database | |  |  |  |  | x |  |  | M | Y | A | A |
| Designing GUI | | | | |  | x |  | x | H | Y | A | A |
| *ON/OFF the system* | | | | |  | | | |  | |  |  |
| Choosing switch button | |  |  |  |  |  | x |  | H | Y | A | A |
| *Report 6* | | | | |  | | | |  | | | |
| Orienting the way writing report 6 | | | | | x | x | x | x | VH | Y | A | A |
| **Date: 08/08/2015** | | | | |  | | | |  | | | |
| *Learning Function* | | | | |  | | | |  | |  |  |
| Implementing the practice function of learning | | | | | x |  | x |  | VH | Y | A | A |
| Demo GUI |  |  |  |  |  | x |  | x | H | Y | A | A |
| *ON/OFF the system* | | | | |  | | | |  | |  |  |
| Constructing circuit | | | | |  |  | x | x | VH | Y | A | A |
| *Report 6* | | | | |  | | | |  | | | |
| Reviewing report 6 | | | | | x | x | x | x | VH | Y | A | A |
| **Date: 11/08/2015** | | | | |  | | | |  | | | |
| *Learning Function* | | | | |  | | | |  | |  |  |
| Demo learning function | | | | | x |  | x |  | H | Y | A | A |
| Demo GUI |  |  |  |  |  | x |  | x | M | Y | A | A |
| *ON/OFF the system* | | | | |  | | | |  | |  |  |
| Demo constructed circuit | | | | |  |  | x | x | H | Y | A | A |
| Connecting to the system | | |  |  |  |  | x | x | VH | Y | A | A |
| *Report 6* | | | | |  | | | |  | | | |
| Reviewing report 6 | | | | | x | x | x | x | VH | Y | A | A |

Table 10: Scrum Meeting Minutes Detail

## 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>

# Report No. 3 Software Requirement Specification

## User Requirement Specification

*The system is not only reserved for mute person but also everyone who wants to learn sign language. Therefore, we have determined the requirement from these users:*

* Recognize his or her hand signs to text and sound: users want devices that can recognize exactly their hand signs. Then, the device must show recognition results via text on screen and emit pronunciation of this word via speaker.
* Learn the way expressing hand signs: there still are a lot of hand signs that users do not know exactly, they want a device that can help them practice these signs. The system should have images which can describe clearly the way expressing hand sign for user can follow. In addition, the system should have practice function for user practice.
* Controlling the system by hand gesture: users want to perform the operations of the system through his or her hand gesture without electricity devices.
* The system is portable: Users can easily move the system. They expect the system can work at many places, and it still works during a power outage.
* System's power must be controlled: Users can know the remaining battery capacity to monitor the use of equipment. Moreover, they can charge the battery when the battery is low.
* System should be easy to use as the electricity systems people use in daily live: Users can turn on/off the system safely without prejudice to the durability of the equipment.

## 2. System Requirement Specification

### 2.1 External Interface Requirement

External interface is concerned with designing interactive products to support the way people communicate and interact in their everyday and working lives. The products must be usability means easy to learn, effective to use and provide an enjoyable experience.

#### 2.1.1 User Interface

* The GUI should be simple, clear, intuitive, and reminiscent.
* The interface is accessible, easy to use, and efficient.
* The interface should meet some criteria such as direct manipulation, device actions, information processing approach, visual features, …
* Each screen has fully instructions of the function implementation. Besides that, it still provides error, success, or implementation notification.

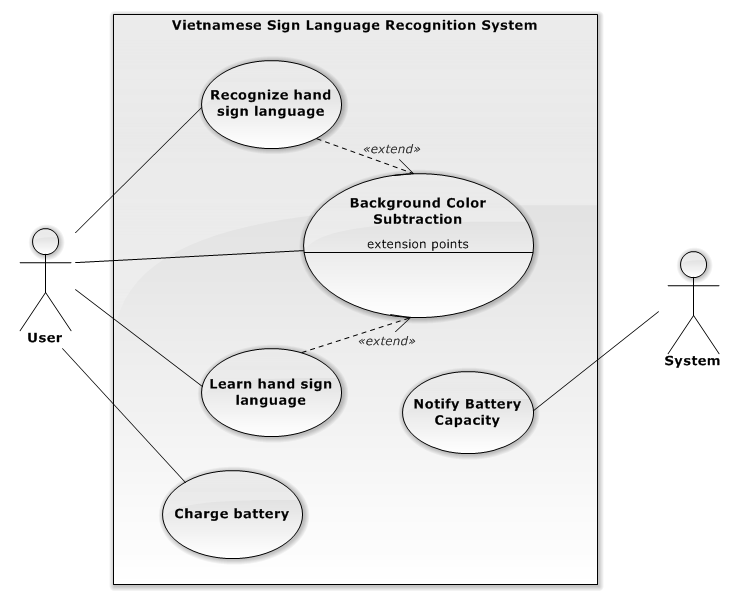
#### 2.1.2 Hardware Interface

* The system must design hardware interface similar to the standard electricity system for anyone can use.
* Provide fully devices of a portable system.
* The system needs to be designed suitable for capturing the hands with an appropriate height, and a width for people can watch the LCD.
* The provided devices should be easy to replace.
* Electricity devices should be packaged in the safety way.

#### 2.1.3 Software Interface

* Linux GUI for Raspbian Operating System.
* The interface must be responsive for LCD 7-inch with the resolution 1024 \* 600.

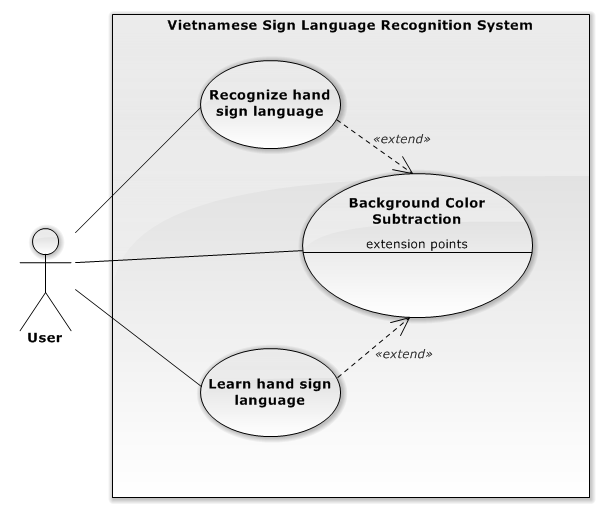
### 2.2 System Overview Use Case



**Figure 4: System Overview Use Case**

### 2.3 List of Use Case

#### 2.3.1 Background Color Subtraction

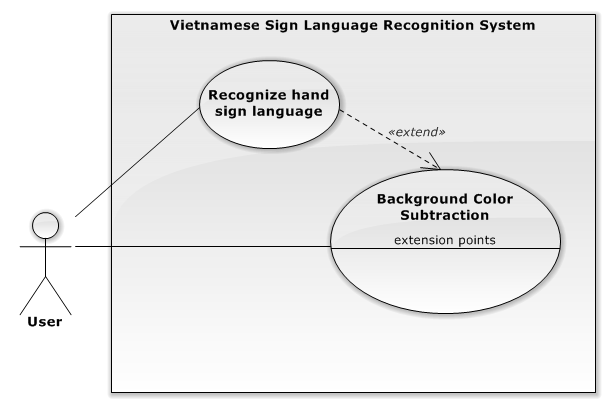


**Figure 5: Recognize Hand Sign Language use case diagram**

**Use Case Specification**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **USE CASE - 1 SPECIFICATION** | | | | | |
| **Use-case No.** | VSLR001 | **Use-case Version** | | | 2.0 |
| **Use-case Name** | Background Color Subtraction | | | | |
| **Author** | Nguyễn Hữu Kỳ Long | | | | |
| **Date** | 31/05/2015 | | **Priority** | High | |
| **Actor**   * User   **Summary**   * The use case describes the way subtracting background color and checking the result of background color subtraction.   **Goal**   * The system can detect and keep track of the hands.   **Triggers**   * User turns on the switch button on the system hardware.   **Preconditions**   * The system is turned on.   **Post Conditions**   * **On Success**: The system navigates to the interface to select function.   **Main Success Scenario**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | - User turns on the switch button on the system hardware. | - The system displays the notify requiring users move out of the camera area.  - The system shows the images captured from camera on the interface for users.  - The system shows the count down time and counts down from 5 by a second.  - The system shows the message “Đang tiến hành"  [Exception No.1] | | 2 | User keeps the background fixed and waits for the count down time is 0. | - The system displays the notify requiring users show the “testing” hand sign inside the camera area.  - The system shows the images subtracting background color on the interface for users.  - The system shows the count down time and counts down from 5 by a second.  - The system shows the message “Đang tiến hành" | | 3 | User shows the right “testing” sign inside the camera. | - The system continues counting down.  - The system shows a message “Thành Công”.  [Alternative No.1] | | 4 | User waits for count down time is 0. | - The system navigates to the interface for user selects function.  [Alternative No.2] |   **Alternative Scenario**   |  |  |  | | --- | --- | --- | | No | Actor Action | System Response | | 1 | User shows the wrong “testing” signs or the background user selected is not good. | - The system continues counting down.  - The system shows a message “Đang tiến hành”. | | 2 | The user “testing” sign is not recognized. | - The system shows message “Thất bại”  - The system backs to the Step No.1 |   **Exceptions**   |  |  |  | | --- | --- | --- | | No | Actor Action | System Response | |  |  |  |   **Relationships**   * “Background Color Subtraction” use case is conditionally extended by “Recognize Hand Sign Language” use case and “Learn Hand Sign Language” in extension points “Hand Sign Recognition” and “Learning Hand Sign” respectively.   **Business Rules**   * The background color subtraction will executes immediately when the application starts working. * The images captured from camera will shows continuously for users can follow. * The count down time informs users of the system is working and will finish in five seconds. * The recognizing “testing” hand sign step is to check whether background is good to recognize and the recognition will execute continuously in five seconds. * Users can get the “Thành công” message immediately when the “testing” hand sign recognition is successful and the recognition will stop. However, the count down time still continues down to 0 and then the system will move to the function interface. * After five seconds, if the “testing” sign recognition is unsuccessful, the system will backs to the first step in five seconds. | | | | | |

#### 2.3.2 Recognize Hand Sign Language

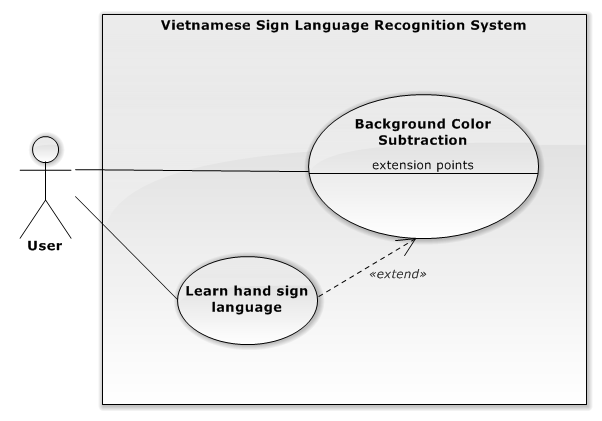
****

**Figure 6: Recognize Hand Sign Language use case diagram**

**Use Case Specification**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **USE CASE - 2 SPECIFICATION** | | | | | |
| **Use-case No.** | VSLR002 | **Use-case Version** | | | 2.0 |
| **Use-case Name** | Recognize Hand Sign Language | | | | |
| **Author** | Nguyễn Hữu Kỳ Long | | | | |
| **Date** | 31/05/2015 | | **Priority** | High | |
| **Actor**   * User   **Summary**   * The use case describes the way recognizing hand signs captured by camera.   **Goal**   * Recognize hand signs and translate them to the same meaning content with the kind of sound and text.   **Triggers**   * User shows the specific “select” hand sign on the “Nhận dạng” function area that is drawn on the images shown continuously to user.   **Preconditions**   * Background color subtraction is successful.   **Post Conditions**   * **On Success**: The translated content shows on the screen and speaker of LCD.   **Main Success Scenario**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | - User completes background color subtraction step | - The system shows a notify “Hãy chọn chức năng mong muốn bằng cách đưa ký hiệu hình bên vào vùng chức năng đó!” in the groupbox “Thông Báo”  - The analyzed images show on the interface continuously.  - Two white “Nhận Dạng” and “Học” area are drawn inside analyzed images.  - The system shows an image guiding users to select function. | | 2 | - User shows “select” hand sign inside “Nhận dạng” area. | - The system shows the hand sign recognition interface  - The analyzed images show on the interface continuously.  - A notify “Hệ thống sẽ lưu lại kết quả nhận dạng sau 3 giây” is shown in groupbox “Thông Báo”  - Countdown time is shown from 3 in groupbox “Thời Gian”  - The system shows two groupbox “Nội dung toàn bộ” and “Kết Quả Hiện Tại” with empty content. | | 3 | User shows the hand sign through camera | - Countdown time counts down by second.  - The system shows messages containing the recognition result of the current hand sign continuously in the group “Kết Quả Hiện Tại”.  [Alternative No.1] | | 4 | User waits for the counting down counts to 0. | - The entire translated content will be updated and shown in the group “Nội Dung Toàn Bộ”  - The system backs to step No.1.  [Alternative No.2]  [Alternative No.3]  [Alternative No.4] |   **Alternative Scenario**   |  |  |  | | --- | --- | --- | | No | Actor Action | System Response | | 1 | Can not detect the hand inside the camera area. | - Countdown time counts down by second.  - The system shows a message “Không tìm thấy bàn tay!” in the group “Kết Quả Hiện Tại”. | | 2 | Detect the “end” hand sign inside the camera area. | - The system navigates to the interface at the step No.1. | | 3 | Detect the “speak” hand sign inside the camera area. | - The system reads the whole content result via LCD speaker.  - After that, the whole content will be cleared.  - The system backs to step No.2 | | 4 | No hand inside the camera area | - The system remains the whole content without updating.  - The system backs to step No.2 |   **Exceptions**   |  |  |  | | --- | --- | --- | | No | Actor Action | System Response | |  |  |  |   **Relationships**   * “Recognize hand sign language” use case is an optional extension of “background color subtraction”   **Business Rules**   * The images captured from camera will be analyzed be system and then shown on the interface for users can keep track their hands. * The recognition result of the current hand sign will be shown as soon as possible for users can check the hand sign. * The last recognition result of the hand sign will update to the entire translated content every 3 seconds. * Every 3 seconds, if the hand sign is recognized with “speak” hand sign, the system will speak the whole content via speaker before clearing the content. * When the message “Không tìm thấy bàn tay!” is shown, that means the system does not detect user’s hands on the captured image. * Every 3 seconds, if the system does not detect user’s hands, it will not update the whole content. * Recognizing works continuously until the user shows “end” hand sign through camera. * The recognition result is always gotten from the system database. | | | | | |

#### 2.3.3 Learn Hand Sign



**Figure 7: Learn sign use case diagram**

**Use Case Specification**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **USE CASE - 3 SPECIFICATION** | | | | | |
| **Use-case No.** | VSLR003 | **Use-case Version** | | | 2.0 |
| **Use-case Name** | Learn Hand Sign | | | | |
| **Author** | Nguyễn Hữu Kỳ Long | | | | |
| **Date** | 31/05/2015 | | **Priority** | Medium | |
| **Actor**   * User   **Summary**   * The use case describes the way practicing a hand sign.   **Goal**   * It is to help user training his or her hand gesture more accurately.   **Triggers**   * User shows the specific “select” hand sign on the “Học” function area that is drawn on the images shown continuously to user.   **Preconditions**   * Background color subtraction is successful.   **Post Conditions**   * **On Success**: The system shows the image describing the selected word and the mean of the hand sign which is captured.   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | - User completes background color subtraction step. | - The system shows a notify “Hãy chọn chức năng mong muốn bằng cách đưa ký hiệu hình bên vào vùng chức năng đó!” in the groupbox “Thông Báo”  - The analyzed images show on the interface continuously.  - Two white “Nhận Dạng” and “Học” area are drawn inside analyzed images.  - The system shows an image guiding users to select function. | | 2 | - User shows “select” hand sign inside “Học” area. | - The system shows the list of words on the interface in the group box “Hướng Dẫn”.  - The analyzed images show on the interface continuously.  - A notify “Hãy đưa kí hiệu trong hướng dẫn vào vùng mũi tên lên xuống để thay đổi từ được chọn ” is shown in the groupbox “Thông Báo”  - Two white “Lên” and “Xuống” area were drawn on these images showing on the interface. | | 3 | User shows the hand gesture through camera. | - The system returns the recognition result on text of the current hand sign continuously in the group “Kết Quả Hiện Tại”.  [Alternative No.1]  [Alternative No.2]  [Alternative No.3]  [Alternative No.4] |   **Alternative Scenario**   |  |  |  | | --- | --- | --- | | No | Actor Action | System Response | | 1 | Detect the “select” hand sign inside the “Lên” area drawn on the interface. | - The system will move the selection up to upper word in the list of words.  - The system shows the image describing the selected word. | | 2 | Detect the “select” hand sign inside the “Xuống” area drawn on the interface. | - The system will move the selection down to lower word in the list of words.  - The system shows the image describing the selected word. | | 3 | Detect the “end” hand sign through camera. | - The system navigates to the interface at the step No.1. | | 4 | Detect no hand inside the camera area. | - The system shows a message “Không tìm thấy bàn tay!” in the group box “Kết Quả Hiện Tại” |   **Exceptions:**   |  |  |  | | --- | --- | --- | | No | Actor Action | System Response | |  |  |  |   **Relationships**   * “Learn hand sign language” use case is an optional extension of “background color subtraction”   **Business Rules**   * List of words is always loaded from the system database. * The word “A” will be selected first. * The image corresponding to the selected word will be loaded and shown on the interface. * Recognizing processes continuously until the “end” hand sign is recognized. * Users just select one of two functions. * When the message “Không tìm thấy bàn tay!” is shown, that means the system does not detect user’s hands on the captured image. * The “Lên” rectangle is to move selecting up one-step in the list of words. * The “Xuống” rectangle is to move selecting down one-step in the list of words. * The recognition result of the current hand sign will be shown as soon as possible for users can check the hand sign. | | | | | |

#### 2.3.4 Charge Battery



**Figure 7: Charge Battery use case diagram**

#### Notify Battery Capacity

****

**Figure 8: Notify Battery Capacity use case diagram**

**Use Case Specification**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **USE CASE - 6 SPECIFICATION** | | | | | |
| **Use-case No.** | VSLR001 | **Use-case Version** | | | 2.0 |
| **Use-case Name** | Notify Battery Capacity | | | | |
| **Author** | Nguyễn Hữu Kỳ Long | | | | |
| **Date** | 31/05/2015 | | **Priority** | High | |
| **Actor**   * System   **Summary**   * The use case can notify user the battery capacity when system is working.   **Goal**   * Notify users of the battery capacity can use the system in an appropriate way.   **Triggers**   * Users switch on the system.   **Preconditions**   * The system is ON.   **Post Conditions**   * **On Success**: The battery capacity will shows continuously at the LEDs indicator on the system box and on the interface.   **Main Success Scenario**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | System checks the battery capacity is higher than 75%.  [Alternative No.1]  [Alternative No.2]  [Alternative No.3]  [Alternative No.4] | - System shows four bright leds on the box.  - System shows the “full” battery image on the top right of the application interface. |   **Alternative Scenario**   |  |  |  | | --- | --- | --- | | No | Actor Action | System Response | | 1 | System checks the battery capacity is between 75% and 50%. | - System shows three bright leds on the box.  - System shows the “75%” battery image on the interface. | | 2 | System checks the battery capacity is between 50% and 25%. | - System shows two bright leds on the box.  - System shows the “50%” battery image on the interface. | | 3 | System checks the battery capacity is under 25% | - System shows one bright leds on the box.  - System shows the “25%” battery image on the interface.  - System shows a dialog message “Bin yếu vui lòng tắt hệ thống và cắm sạc. Thông báo sẽ được tự động tắt.” within a countdown time from 5 on the system application interface in five seconds. | | 4 | System checks the battery capacity is empty. | - System shows no bright leds on the box.  - System shows the “empty” battery image on the interface.  - System shows a dialog message “Ứng dụng sẽ tắt trong vòng 5 giây. Thông báo sẽ được tự động tắt.” on the system application interface in five seconds. |   **Exceptions**   |  |  |  | | --- | --- | --- | | No | Actor Action | System Response | |  |  |  |   **Relationships**   * N/A   **Business Rules**   * System will check how much the battery capacity is every five minutes. * System shows the battery capacity by the electricity devices such as LEDs on the system hardware. * The notify dialog will show every five minutes if the system checks that battery capacity is under 25% or empty. * The notify dialog closes automatically after five seconds. * A countdown time will be shown on the notify dialog. * Actually, checking battery capacity is to check the voltage level of battery. Battery Capacity Display circuit will check the voltage level of battery continuously and notify on LEDs indicator and it returns 5 levels of battery capacity to the system application every five minutes. * The battery capacity is higher than 75% if the voltage is higher than 12V. * The battery capacity is between 75% and 50% if the voltage is lower than or equal 12V and higher than 11.3V. * The battery capacity is between 50% and 25% if the voltage is lower than or equal 11.3V and higher than 10.8V. * The battery capacity is lower than 25% if the voltage is lower than or equal 10.8V and 9.9V. * The battery capacity is empty if the voltage is lower than 9.9V. | | | | | |

## System Attribute

### Usability

The system should be designed for everyone can use easily in controlling and GUI operations.

#### Graphic User Interface

* The system musts show all instructions, notifications and operations in Vietnamese.

#### Usability

* User just needs to read the user manual which is enclosed with the system for using in the first time. The attached manual guide must be clear. User can read and do by themselves.

#### Hardware controlling

* User can control the device very easily as well as using any electronic device in the daily live.

### Reliability

* The database should be constructed on Vietnamese sign language.
* The system uses “Support Vector Machine” library to recognize hand sign language and OpenCV library to process image.
* The system is using Raspberry PI 2 to process which is popular board in the world.

### Availability

* The system runs continuously about 3 hours with LIPO 2700mAh battery. That means it is safe to user.

### Security

N/A

### Maintainability

* Electronic devices in the system are common so when any electronic equipment, which is attached with the system, is out of ordered, it is so easy to change or to fix at any electronic store.
* The system can be extended in the future.

### Portability

* The system supplies the LIPO battery as power source in which user can use for 3 hours without charging. In addition, the system also provides LIPO B3AC charger for users.
* The system provides a circuit monitoring LIPO battery for users.

### Performance

The system uses Raspberry PI 2 with RAM 1GB as central unit processing, so that the system can recognize one hand sign in 1 to 3 seconds and hand sign recognize can be performed continuously.

# Report No. 4 Software Design Description

## Design Overview

* This document describes the technical and user interface design of VSLR 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 a fully attributed ERD.

## System Architectural Design

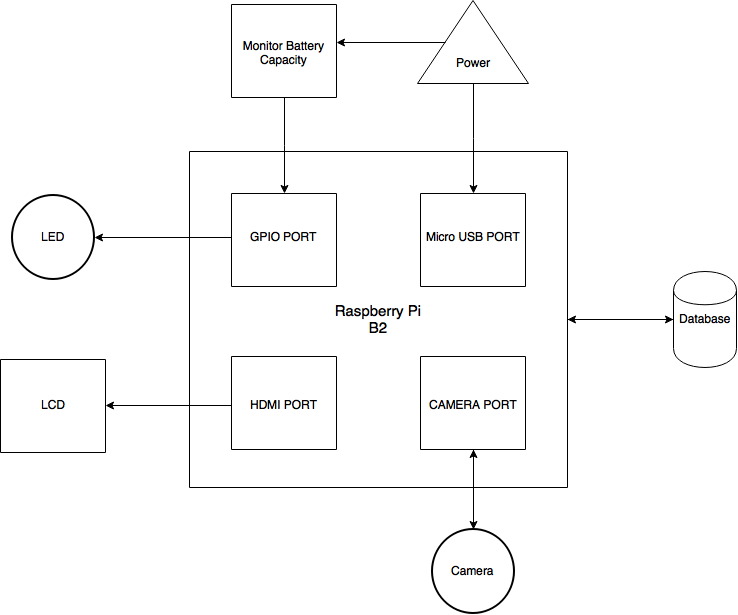


Figure 9: VSLR System Architectural



## Component Diagram

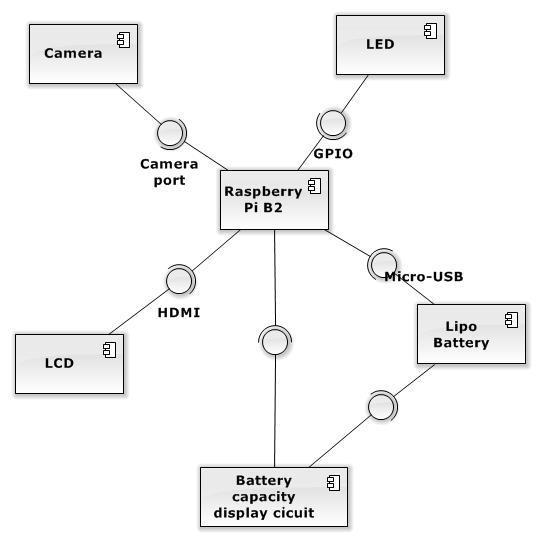


Figure 10: Component Diagram

## Detailed Description

### **Software Detailed Description**

#### Class Diagram

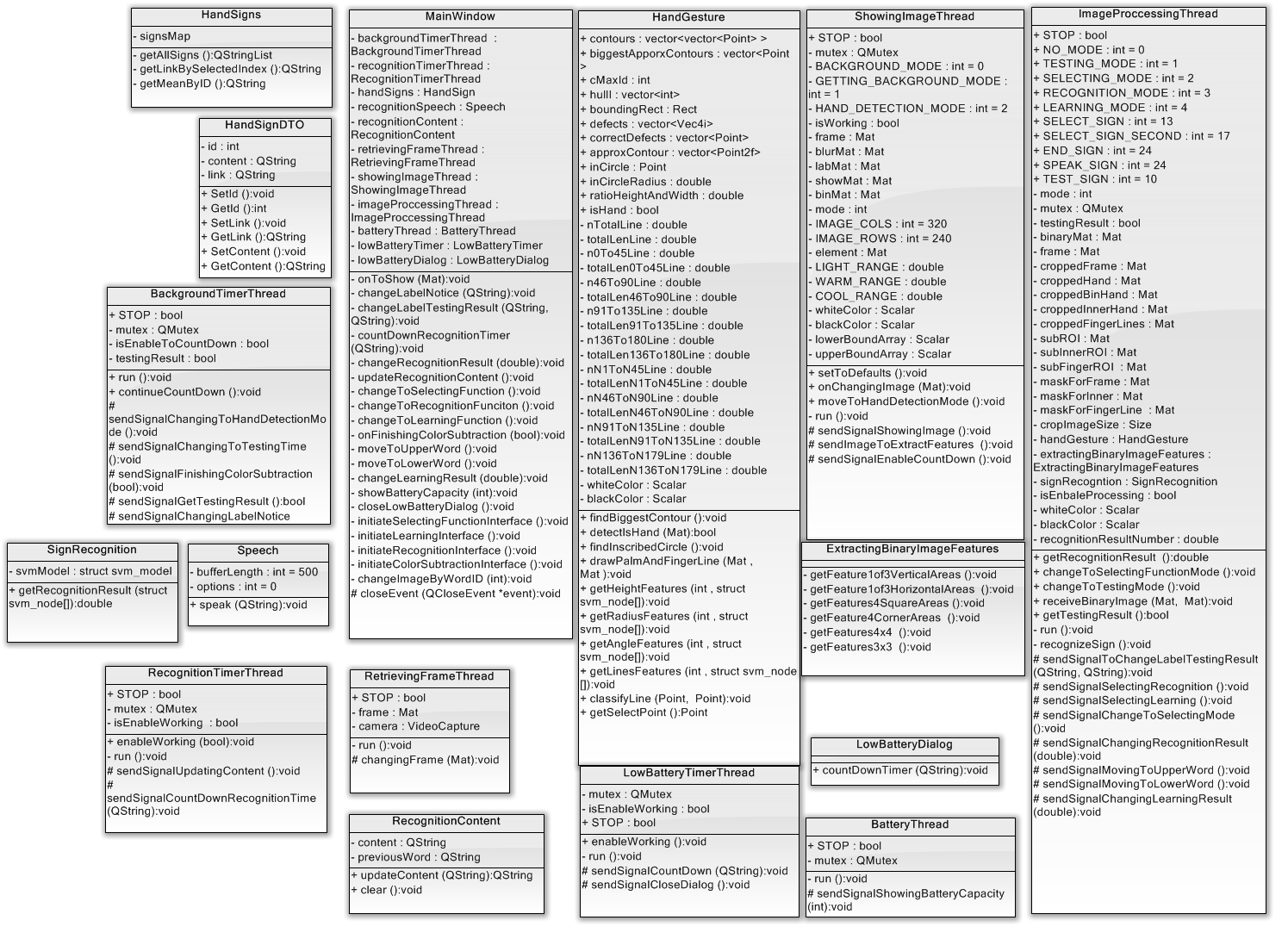


Figure 11: Class Diagram

|  |  |
| --- | --- |
| Class Dictionary : Class Description | |
| Class Name | Description |
| HandSignDTO | This is data transfer object class of table HandSign encapsulating HandSign attributes. |
| HandSigns | This class contains list of hand sign loaded from database and methods for getting attributes of a specific hand sign. |
| BackgroundTimerThread | This class inherits QThread class that is used for sending signals to notify, count down timer and move between steps in background color subtraction phase.  This thread will stop if the phase testing background color subtraction succeeds. |
| RetreivingFrameThread | This class inherits QThread class that is used for activing camera and then retrieving images captured from camera continuously.  This thread will run until the application stops. |
| ShowingImageThread | This class inherits QThread class that is used for getting image from RetreivingFrameThread when it finish , processing these image to subtract background color which will show on the interface.  This thread will run until the application stops. |
| ImageProcessingThread | This class inherits QThread class that receives image subtracted background color to detect hands, create the binary image containing hand features and then extract the features from these image.  This thread receives signals asking to perform image processing for testing background subtraction, learning function and recognition function.  This thread just is enable to process when it receives image from ShowingImageThread.  This thread will run until the application stops. |
| HandGesture | This class encapsulates attributes, methods to create binary images containing hand features and methods to output features related to height, hand palm and finger lines. |
| ExtractingBinaryImage | This class provides methods which outputs histogram features of a binary image. |
| SignRecognition | This class encapsulates attributes and methods to perform SVM algorithm for hand sign recognition. |
| Speech | This class encapsulates attributes and methods to initiate some parameters for using espeak library to read a content in sound via LCD speaker. |
| RecognitionContent | This class encapsulates attributes and methods to manage the whole recognition content for hand sign language recognition function. |
| RecognitionTimerThread | This class inherits QThread class that is used for implementing the real time timer to manage hand sign recognition function.  This thread will start when the application runs but it is just enable to process when the recognition function is selected.  This thread will stop until the application stops. |
| BatteryThread | This class inherits QThread class that will run during the application works to check battery capacity and send signals notifying user. |
| LowBatteryDialog | This class is used for initiating dialog interface which shows low battery announcement. |
| LowBatteryTimerThread | This class inherits QThread processes as real time timer to manage the time showing LowBatteryDialog. |
| MainWindow | This class is the main UI thread which manages the application interfaces and creates communications between thread objects. |

#### Class Diagram Explanation

##### HandSigns

Attribute

|  |  |  |  |
| --- | --- | --- | --- |
| **Attribute** | **Type** | **Visibility** | **Description** |
| signsMap | QMap<int, HandSignDTO> | Private | This map contains hand sign records loaded from HandSign table. |

Method

|  |  |  |  |
| --- | --- | --- | --- |
| **Method** | **Return type** | **Visibility** | **Description** |
| getAllSigns | QStringList | Private | Returns all records signsMap hold. |
| getLinkBySelectedIndex | QString | Private | Return record by index user select in combobox. |
| getMeanByID | QString | Private | Return record by recognition result returned from SVM. |

##### HandSignDTO

##### Attribute

|  |  |  |  |
| --- | --- | --- | --- |
| **Attribute** | **Type** | **Visibility** | **Description** |
| id | int | Private | Unique identifier of a word and recognition result number. |
| content | QString | Private | Content of a particular sign. |
| link | QString | Private | Path to directory contain image of hand sign. |

##### Method

|  |  |  |  |
| --- | --- | --- | --- |
| **Method** | **Return type** | **Visibility** | **Description** |
| SetId | void | Public | Set value of attribute id |
| GetId | int | Public | Get attribute id value |
| SetLink | void | Public | Set value of attribute link |
| GetLink | QString | Public | Get attribute link value |
| SetContent | void | Public | Set value of attribute content |
| GetContent | QString | Public | Get attribute content value |

##### 

##### BackgroundTimerThread

##### Attribute

|  |  |  |  |
| --- | --- | --- | --- |
| **Attribute** | **Type** | **Visibility** | **Description** |
| STOP | bool | Public | This variable is used for to stop the thread before the application is closed. |
| mutex | QMutex | Private | It is to protect section of code so that only one thread can access at a time. |
| isEnableToCountDown | bool | Private | This variable is to check whether this thread continues counting down timer. |
| testingResult | bool | Private | This variable is to held background subtraction testing result and it is condition to stop this thread. |

##### Method

|  |  |  |  |
| --- | --- | --- | --- |
| **Method** | **Return type** | **Visibility** | **Description** |
| run | void | Private | This is starting point for the thread and this method implements functions of the thread. |
| sendSignalChangingToHandDetectionMode | void | Private | This is signal which is sent Image Processing thread to move to hand detection step. |
| sendSignalChangingToTestingTime | void | Private | This is signal which is sent Image Processing thread to move to background subtraction testing step. |
| sendSignalFinishingColorSubtraction | void | Private | This is signal which is sent with testing result to notify main thread. |
| sendSignalGetTestingResult | bool | Private | This is signal which is sent Image Processing thread to get background subtraction testing result and stop testing step. |
| sendSignalChangingLabelNotice | void | Private | This is signal which is sent with notify content to main thread. |
| sendSignalFailTesingResult | void | Private | This is signal which is sent to notify main thread of testing result. |
| continueCountDown | void | Private | This is function that is called in response to signal which enables to continue counting down timer. |

##### 

##### SignRecognition

##### Attribute

|  |  |  |  |
| --- | --- | --- | --- |
| **Attribute** | **Type** | **Visibility** | **Description** |
| svmModel | struct svm\_model | Private | SVM will load the model file to this struct. |

##### Method

|  |  |  |  |
| --- | --- | --- | --- |
| **Method** | **Return type** | **Visibility** | **Description** |
| getRecognitionResult | double | Public | This method receives the hand sign features, then uses SVM library to predict recognition result and return it. |

##### Speech

##### Attribute

|  |  |  |  |
| --- | --- | --- | --- |
| **Attribute** | **Type** | **Visibility** | **Description** |
| bufferLength | int | Private |  |
| options | int | Private |  |

##### Method

|  |  |  |  |
| --- | --- | --- | --- |
| **Method** | **Return type** | **Visibility** | **Description** |
| speak | void | Public |  |

##### RecognitionTimerThread

##### Attribute

|  |  |  |  |
| --- | --- | --- | --- |
| **Attribute** | **Type** | **Visibility** | **Description** |
| STOP | bool | Public | This variable is used for to stop the thread before the application is closed. |
| mutex | QMutex | Private | It is to protect section of code so that only one thread can access at a time. |
| isEnableWorking | bool | Private | This variable is to check whether the thread can process functions. |

##### Method

|  |  |  |  |
| --- | --- | --- | --- |
| **Method** | **Return type** | **Visibility** | **Description** |
| enableWorking | void | Public | This is function that is called in response to signal which enables to perform functions. |
| run | void | Private | This is starting point for the thread and this method implements function counting down timer. |
| sendSignalUpdatingContent | void | Protected | This is signal which is sent to notify main thread of performing updating recognition content. |
| sendSignalCountDownRecognitionTime | void | Protected | This is signal which is sent main thread the real time timer. |

#### 

##### RetrievingFrameThread

##### Attribute

|  |  |  |  |
| --- | --- | --- | --- |
| **Attribute** | **Type** | **Visibility** | **Description** |
| STOP | bool | Public | This variable is used for to stop the thread before the application is closed. |
| frame | Mat | Private | This is object holding image captured from camera. |
| camera | VideoCapture | Private | This is object which actives camera working and retrieves image captured from camera to frame object. |

##### Method

|  |  |  |  |
| --- | --- | --- | --- |
| **Method** | **Return type** | **Visibility** | **Description** |
| run | void | Private | This is starting point for the thread and this method implements retrieving images continuously. |
| changingFrame | void | Protected | This is signal which is sent to notify Showing Image thread that new image is retrieved. |

##### 

##### RecognitionContent

##### Attribute

|  |  |  |  |
| --- | --- | --- | --- |
| **Attribute** | **Type** | **Visibility** | **Description** |
| content | QString | Private | This is the whole recognition content will be updated every 3 seconds. |
| previousWord | QString | Private | This object holds the last updated hand sign. |

##### Method

|  |  |  |  |
| --- | --- | --- | --- |
| **Method** | **Return type** | **Visibility** | **Description** |
| updateContent | QString | Public | This method receives new hand sign recognition result and then performs updating content function. |
| clear | void | Public | This method is used for clear the current content. |

##### LowBatteryTimerThread

##### Attribute

|  |  |  |  |
| --- | --- | --- | --- |
| **Attribute** | **Type** | **Visibility** | **Description** |
| STOP | bool | Public | This variable is used for to stop the thread before the application is closed. |
| mutex | QMutex | Private | It is to protect section of code so that only one thread can access at a time. |
| isEnableWorking | bool | Private | This variable is to check whether the thread can process functions. |

##### Method

|  |  |  |  |
| --- | --- | --- | --- |
| **Method** | **Return type** | **Visibility** | **Description** |
| enableWorking | void | Public | This is function that is called in response to signal which enables thread can perform functions. |
| run | void | Private | This is starting point for the thread and this method implements retrieving images continuously. |
| sendSignalCountingDown | void | Protected | This is signal which is sent the real time timer to low battery dialog. |
| sendSignalClosingDialog | void | Protected | This is signal which is to notify main thread of closing current low battery dialog. |

##### BatteryThread

##### Attribute

|  |  |  |  |
| --- | --- | --- | --- |
| **Attribute** | **Type** | **Visibility** | **Description** |
| STOP | bool | Public | This variable is used for to stop the thread before the application is closed. |
| mutex | QMutex | Private | It is to protect section of code so that only one thread can access at a time. |

##### Method

|  |  |  |  |
| --- | --- | --- | --- |
| **Method** | **Return type** | **Visibility** | **Description** |
| run | void | Private | This is starting point for the thread and this method implements retrieving images continuously. |
| sendSignalShowingBatteryCapacity | void | Protected | This is signal which is to notify main thread of current battery capacity every 5 minutes. |

##### 

##### LowBatteryDialog

##### Attribute

|  |  |  |  |
| --- | --- | --- | --- |
| **Attribute** | **Type** | **Visibility** | **Description** |
|  |  |  |  |

##### Method

|  |  |  |  |
| --- | --- | --- | --- |
| **Method** | **Return type** | **Visibility** | **Description** |
| countDownTimer | void | Public | This method is used for changing the timer which is shown on the low battery dialog interface. |

##### HandGesture

##### Attribute

|  |  |  |  |
| --- | --- | --- | --- |
| **Attribute** | **Type** | **Visibility** | **Description** |
| contours | vector<vector<Point> > | Public | This holds sets of points as the contours of possible hands. |
| cMaxId | int | Public | This variable is to determines which contour is of right hand. |
| boundingRect | Rect | Public | This is rectangle bounding the contour of the right hand. |
| approxContour | vector<Point2f> | Public | This is contour of the hand in the form of set of float points after it is approximated polygonal curves. |
| biggestApporxContour | vector<Point > | Public | This is contour of the hand in the form of set of integer points after it is approximated polygonal curves. |
| hullI | vector<int> | Public | This is convex hull output. |
| defects | vector<Vec4i> | Public | This is convexity defects output. |
| correctDefects | vector<Point> | Public | This is convexity defects output after verifying which is correct. |
| inCircle | Point | Public | This is center location of hand palm. |
| inCircleRadius | double | Public | This is radius length of hand palm. |
| ratioHeightAndWidth | double | Public | This is height feature output of the hand sign. |
| isHand | bool | Public | This is output of detecting the hands. |
| nTotalLine | double | Private | This variable holds the number of finger lines. |
| totalLenLine | double | Private | This variable hold the total length of all finger lines. |
| n0To45Line | double | Private | This variable holds the number of finger lines belongs to 0 to 45 degree category. |
| totalLen0To45Line | double | Private | This variable hold the total length of all finger lines belongs to 0 to 45 degree category. |
| n46To90Line | double | Private | This variable holds the number of finger lines belongs to 46 to 90 degree category. |
| totalLen46To90Line | double | Private | This variable hold the total length of all finger lines belongs to 46 to 90 degree category. |
| n91To135Line | double | Private | This variable holds the number of finger lines belongs to 91 to 135 degree category. |
| totalLen91To135Line | double | Private | This variable hold the total length of all finger lines belongs to 91 to 135 degree category. |
| n136To180Line | double | Private | This variable holds the number of finger lines belongs to 136 to 180 degree category. |
| totalLen136To180Line | double | Private | This variable hold the total length of all finger lines belongs to 136 to 180 degree category. |
| nN1ToN45Line | double | Private | This variable holds the number of finger lines belongs to -45 to -1 degree category. |
| totalLenN1ToN45Line | double | Private | This variable hold the total length of all finger lines belongs to -45 to -1 degree category. |
| nN46ToN90Line | double | Private | This variable holds the number of finger lines belongs to -46 to -90 degree category. |
| totalLenN46ToN90Line | double | Private | This variable hold the total length of all finger lines belongs to -46 to -90 degree category. |
| nN91ToN135Line | double | Private | This variable holds the number of finger lines belongs to -91 to -135 degree category. |
| totalLenN91ToN135Line | double | Private | This variable hold the total length of all finger lines belongs to -91 to -135 degree category. |
| nN136ToN179Line | double | Private | This variable holds the number of finger lines belongs to -136 to -179 degree category. |
| totalLenN136ToN179Line | double | Private | This variable hold the total length of all finger lines belongs to -136 to -179 degree category. |
| whiteColor | Scalar | Private | White color is used to draw binary images. |
| blackColor | Scalar | Private | Black color is used to draw binary images. |

##### Method

|  |  |  |  |
| --- | --- | --- | --- |
| **Method** | **Return type** | **Visibility** | **Description** |
| findBiggestContour | void | Public | This method is to find which contour is right. |
| detectIsHand | bool | Public | This method is to detect the biggest contour is contour of hand or not. |
| findInscribedCircle | void | Public | This method is to specify the hand palm. |
| drawPalmAndFingerLine | void | Public | This method is to draw two binary images such as hand palm and finger line. |
| getHeightFeatures | void | Public | This method returns the height feature of hand. |
| getRadiusFeatures | void | Public | This method calculates and returns radius features of hand palm. |
| getAngleFeatures | void | Public | This method calculates and returns angle features of finger lines. |
| getLinesFeatures | void | Public | This method calculate and returns classifier features of finger lines. |
| classifyLine | void | Public | This method classifies finger lines to 8 degree categories. |
| getSelectPoint | Point | Public | This method is to find the highest point of select hand sign and return it. |

#### 

##### ShowingImageThread

##### Attribute

|  |  |  |  |
| --- | --- | --- | --- |
| **Attribute** | **Type** | **Visibility** | **Description** |
| STOP | bool | Public | This variable is used for to stop the thread before the application is closed. |
| mutex | QMutex | Private | It is to protect section of code so that only one thread can access at a time. |
| BACKGROUND\_MODE | int | Private | This is a constant static variable holds value of background mode. |
| GETTING\_BACKGROUND\_MODE | int | Private | This is a constant static variable holds value of step sampling background. |
| HAND\_DETECTION\_MODE | int | Private | This is a constant static variable holds value of mode detecting hand. |
| isWorking | bool | Private | This variable is to check whether this thread is working. |
| frame | Mat | Private | This is image received from Retrieving Image thread. |
| blurMat | Mat | Private | This is image after blurring image process. |
| labMat | Mat | Private | This is image after converting BGR image into LAB image. |
| showMat | Mat | Private | This is image subtracted background and it is used for showing on the interface. |
| binMat | Mat | Private | This is binary image subtracted background. |
| mode | int | Private | This variable holds value of current mode. |
| IMAGE\_COLS | int | Private | This is constant static variable holds image’s width value. |
| IMAGE\_ROWS | int | Private | This is constant static variable holds image’s height value. |
| element | Mat | Private | This is structuring element is for morphological transformations. |
| LIGHT\_RANGE | double | Private | This is constant static variable holds range value of lightness. |
| WARM\_RANGE | double | Private | This is constant static variable holds range value of warm color-opponent. |
| COOL\_RANGE | double | Private | This is constant static variable holds range value of cool color-opponent. |
| whiteColor | Scalar | Private | White color is used to draw binary images. |
| blackColor | Scalar | Private | Black color is used to draw binary images. |
| lowerBoundArray | Scalar | Private | This array holds lower boundary of every single pixels. |
| upperBoundArray | Scalar | Private | This array holds upper boundary of every single pixels. |

##### Method

|  |  |  |  |
| --- | --- | --- | --- |
| **Method** | **Return type** | **Visibility** | **Description** |
| setToDefaults | void | Public | This method sets this thread back to first step. |
| onChangingImage | void | Public | This is function that is called in response to signal, which receives new image captured from camera. |
| moveToHandDetectionMode | void | Public | This is function that is called in response to signal, which changes current mode to hand detection mode. |
| run | void | Private | This is starting point for the thread and this method implements processing these image to subtract background color which will show on the interface. |
| sendSignalShowingImage | void | Protected | This is signal which is to send images subtracted background color to main thread can show on the interface. |
| sendImageToExtractFeatures | void | Protected | This is signal which is to send images subtracted background color to Image Processing thread can extract features. |
| sendSignalEnableCountDown | void | Protected | This is signal which is to enable Timer thread to continue working. |

##### 

##### ExtractingBinaryImageFeatures

##### Attribute

|  |  |  |  |
| --- | --- | --- | --- |
| **Attribute** | **Type** | **Visibility** | **Description** |
|  |  |  |  |

Method

|  |  |  |  |
| --- | --- | --- | --- |
| **Method** | **Return Type** | **Visibility** | **Descrition** |
| getFeature1of3VerticalAreas | void | Private | This method extracts 3 vertical area features of binary image. |
| getFeature1of3HorizontalAreas | void | Private | This method extracts 3 horizontal area features of binary image. |
| getFeatures4SquareAreas | void | Private | This method extracts 4 square area features of binary image. |
| getFeature4CornerAreas | void | Private | This method extracts 4 triangle area features of binary image. |
| getFeatures4x4 | void | Private | This method extract 16 square area features of binary image. |
| getFeatures3x3 | void | Private | This method extract 9 square area features of binary image. |

##### ImageProcessingThread

##### Attribute

|  |  |  |  |
| --- | --- | --- | --- |
| **Attribute** | **Type** | **Visibility** | **Description** |
| STOP | bool | Public | This variable is used for to stop the thread before the application is closed. |
| NO\_MODE | int | Public | This is a constant static variable holds value of no mode. |
| TESTING\_MODE | int | Public | This is a constant static variable holds value of testing background subtraction mode. |
| SELECTING\_MODE | int | Public | This is a constant static variable holds value of selecting function mode. |
| RECOGNITION\_MODE | int | Public | This is a constant static variable holds value of hand sign language recognition mode. |
| LEARNING\_MODE | int | Public | This is a constant static variable holds value of hand sign language learning mode. |
| SELECT\_SIGN | int | Public | This is a constant static variable holds recognition result of “select” hand sign. |
| SELECT\_SIGN\_SECOND | int | Public | This is a constant static variable holds recognition result of “select” hand sign. |
| END\_SIGN | int | Public | This is a constant static variable holds recognition result of “end” hand sign. |
| SPEAK\_SIGN | int | Public | This is a constant static variable holds recognition result of “speak” hand sign. |
| TEST\_SIGN | int | Public | This is a constant static variable holds recognition result of “test” hand sign. |
| mode | int | Private | This variable holds value of current mode. |
| testingResult | bool | Private | This is testing background color subtraction result at testing mode. |
| binaryMat | Mat | Private | This is hand binary image received from Showing Image thread. |
| frame | Mat | Private | This is image received from Showing Image thread which is subtracted background color. |
| croppedFrame | Mat | Private | This is hand images cropped from images subtracted background. |
| croppedHand | Mat | Private | This is cropped images which contains hand after adjusting size. |
| croppedBinHand | Mat | Private | This is cropped binary images which contains hand after adjusting size. |
| croppedInnerHand | Mat | Private | This is cropped binary images which contains hand palm after adjusting size. |
| croppedFingerLines | Mat | Private | This is cropped binary images which contains finger lines after adjusting size. |
| subROI | Mat | Private | This is region of hand image is to hold hand images cropped. |
| subInnerROI | Mat | Private | This is region of hand palm image is to hold hand palm cropped. |
| subFingerROI | Mat | Private | This is region of finger lines image is to hold finger line images cropped. |
| maskForFrame | Mat | Private | This is a mask which is used for cropping hand images. |
| maskForInner | Mat | Private | This is a mask which is used for cropping hand palm images. |
| maskForFingerLine | Mat | Private | This is a mask which is used for cropping finger line images. |
| cropImageSize | Size | Private | This is common size for every cropped images. |
| handGesture | HandGesture | Private | This is object processing images to create binary images and output features related to height, hand palm and finger lines. |
| extractingBinaryImageFeatures | ExtractingBinaryImageFeatures | Private | This is object processing binary images to output histogram features. |
| signRecogntion | SignRecognition | Private | This object is used for recognize hand sign. |
| isEnbaleProcessing | bool | Private | This variable is to check whether this thread can process images. |
| whiteColor | Scalar | Private | White color is to draw binary images containing features. |
| blackColor | Scalar | Private | Black color is to draw binary images containing features. |
| recognitionResultNumber | double | Private | This is recognition result predicted by SVM. |

##### Method

|  |  |  |  |
| --- | --- | --- | --- |
| **Method** | **Return type** | **Visibility** | **Description** |
| getRecognitionResult | double | Public | This method return hand sign recognition result. |
| recognizeSign | void | Private | This method implements steps to recognize hand sign. |
| changeToSelectingFunctionMode | void | Public | This method is to change current mode to selecting function mode. |
| changeToTestingMode | void | Public | This method is to change current mode to selecting function mode. |
| receiveBinaryImage | void | Public | This is function that is called in response to signal, which receives new image subtracted background color. |
| getTestingResult | bool | Public | This is function that is called in response to signal, which return testing background subtraction result. |
| run | void | Private | This is starting point for the thread and this method perform image processing for testing background subtraction, learning function and recognition function. |
| sendSignalToChangeLabelTestingResult | void | Protected | This is signal which is sent testing background color subtraction to main thread can update the interface. |
| sendSignalSelectingRecognition | void | Protected | This is signal which is sent notify main thread of changing current mode to recognition mode. |
| sendSignalSelectingLearning | void | Protected | This is signal which is sent notify main thread of changing current mode to learning mode. |
| sendSignalChangeToSelectingMode | void | Protected | This is signal which is sent notify main thread of changing current mode to selecting function mode. |
| sendSignalChangingRecognitionResult | void | Protected | This is signal which is sent notify main thread of outputting recognition result at recognition mode. |
| sendSignalMovingToUpperWord | void | Protected | This is signal which is sent notify main thread of “Lên” area is selected at learning mode. |
| sendSignalMovingToLowerWord | void | Protected | This is signal which is sent notify main thread of “Xuống” area is selected atlearning mode. |
| sendSignalChangingLearningResult | void | Protected | This is signal which is sent notify main thread of outputting recognition result at learning mode. |

#### 

##### MainWindow

##### Attribute

|  |  |  |  |
| --- | --- | --- | --- |
| **Attribute** | **Type** | **Visibility** | **Description** |
| backgroundTimerThread | BackgroundTimerThread | Private | This object is to manage background color subtraction phase. |
| recognitionTimerThread | RecognitionTimerThread | Private | This object is to manage timer at recognition mode. |
| handSigns | HandSign | Private | This object is to output hand signs content. |
| recognitionSpeech | Speech | Private | This object is to speak recognition content. |
| recognitionContent | RecognitionContent | Private | This object is to manage the whole recognition content at recognition mode. |
| retrievingFrameThread | RetrievingFrameThread | Private | This is thread object which retrieves images captured from camera continuously. |
| showingImageThread | ShowingImageThread | Private | This is thread object which subtracts background color. |
| imageProccessingThread | ImageProccessingThread | Private | This is thread object which processes image to recognize hand sign during the application runs. |
| batteryThread | BatteryThread | Private | This is thread object which outputs battery capacity every 5 minutes. |
| lowBatteryTimerThread | LowBatteryTimerThread | Private | This object is to manage timer of Low Battery Dialog. |
| lowBatteryDialog | LowBatteryDialog | Private | This is UI thread of Low Battery Dialog. |

##### Method

|  |  |  |  |
| --- | --- | --- | --- |
| **Method** | **Return type** | **Visibility** | **Description** |
| onToShow | void | Private | This is function that is called in response to signal, which shows images on the interface. |
| changeLabelNotice | void | Private | This is function that is called in response to signal, which updates notify on the interface. |
| changeLabelTestingResult | void | Private | This is function that is called in response to signal, which updates testing background color subtraction on the interface. |
| countDownRecognitionTimer | void | Private | This is function that is called in response to signal, which updates real time timer on the interface at recognition mode. |
| changeRecognitionResult | void | Private | This is function that is called in response to signal, which updates recognition result on the interface at recognition mode. |
| updateRecognitionContent | void | Private | This is function that is called in response to signal, which updates new recognition result to the whole content and shows it on the interface at recognition mode. |
| changeToSelectingFunction | void | Private | This is function that is called in response to signal, which implements steps to change to selecting function mode. |
| changeToRecognitionFunciton | void | Private | This is function that is called in response to signal, which implements steps to change to recognition mode. |
| changeToLearningFunction | void | Private | This is function that is called in response to signal, which implements steps to change to learning mode. |
| onFinishingColorSubtraction | void | Private | This is function that is called in response to signal, which checks the testing background subtraction result. |
| moveToUpperWord | void | Private | This is function that is called in response to signal, which moves selection to upper word at learning mode. |
| moveToLowerWord | void | Private | This is function that is called in response to signal, which moves selection to lower word at learning mode. |
| changeLearningResult | void | Private | This is function that is called in response to signal, which updates recognition result on the interface at laerning mode. |
| showBatteryCapacity | void | Private | This is function that is called in response to signal, which shows battery capacity images on the interfaces. |
| closeLowBatteryDialog | void | Private | This is function that is called in response to signal, which closes low battery dialog. |
| initiateSelectingFunctionInterface | void | Private | This methods initiates components of selecting function mode on the interface. |
| initiateLearningInterface | void | Private | This methods initiates components of learning mode on the interface. |
| initiateRecognitionInterface | void | Private | This methods initiates components of recognition mode on the interface. |
| initiateColorSubtractionInterface | void | Private | This methods initiates components of selecting function mode on the interface. |
| changeImageByWordID | void | Private | This is function that is called in response to signal, which updates image of selected word on the interface at learning mode. |
| closeEvent | void | Protected | This is an override method which stops threads which is still working before closing application. |

#### 

#### Activity Diagram

##### Background color subtraction



Figure 12: Background color subtraction Activity Diagram

##### Recognize Hand Sign Language



Figure 13: Recognize Hand Sign Language Activity Diagram

##### Learn Hand Sing Language



Figure 14: Learn Hand Sing Language Activity Diagram

##### Notify Battery Capacity



Figure 15: Notify Battery Capacity Activity Diagram

##### Charge Battery



Figure 16: Charge Battery Activity Diagram

#### Sequence Diagram

#### Subtract Background Color

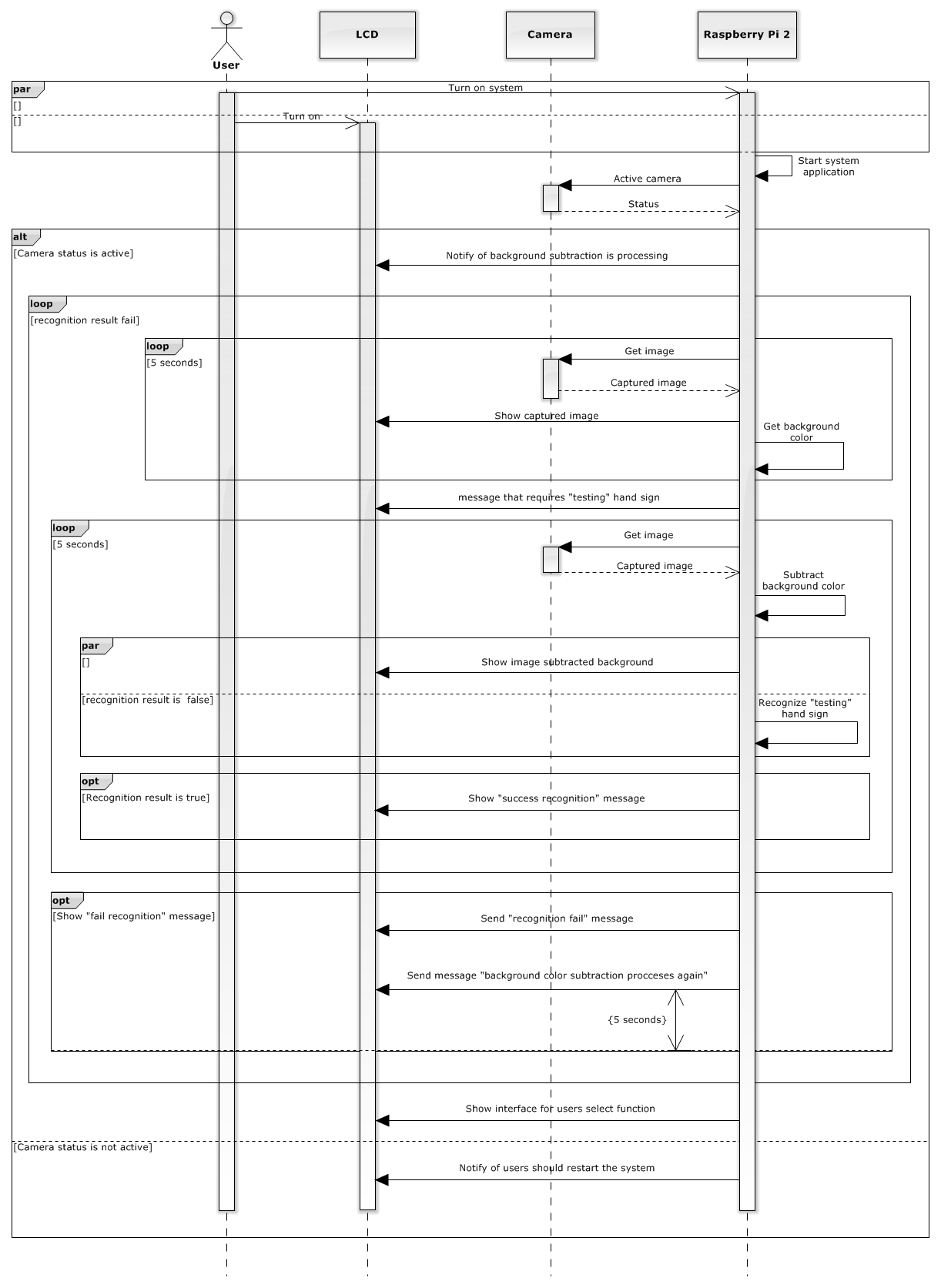


Figure 17: Subtract Background Color Sequence Diagram

#### Recognize Hand Sign Language

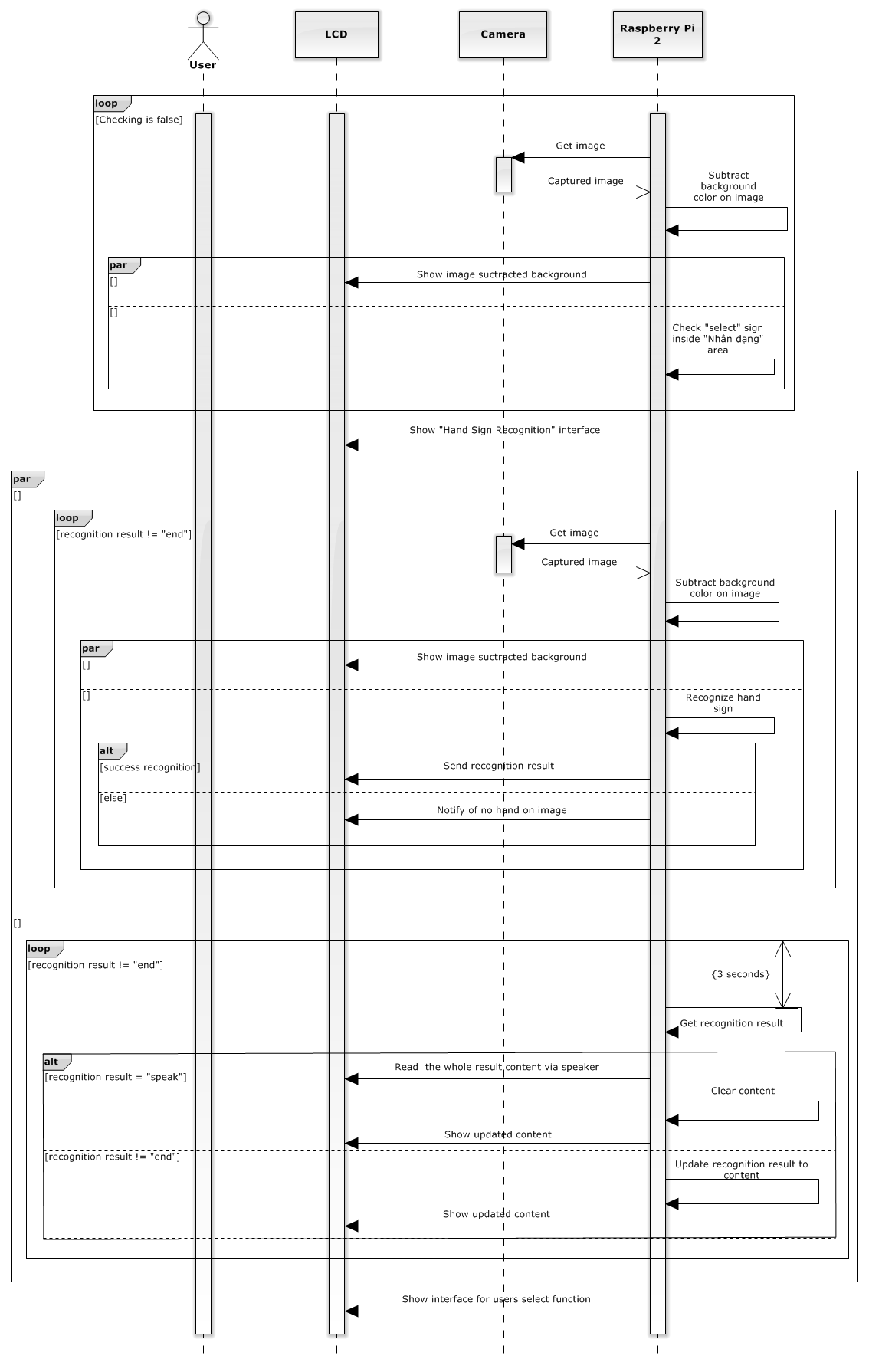


Figure 18: Recognize Hand Sign Language Sequence Diagram

#### Learn Hand Sign Language

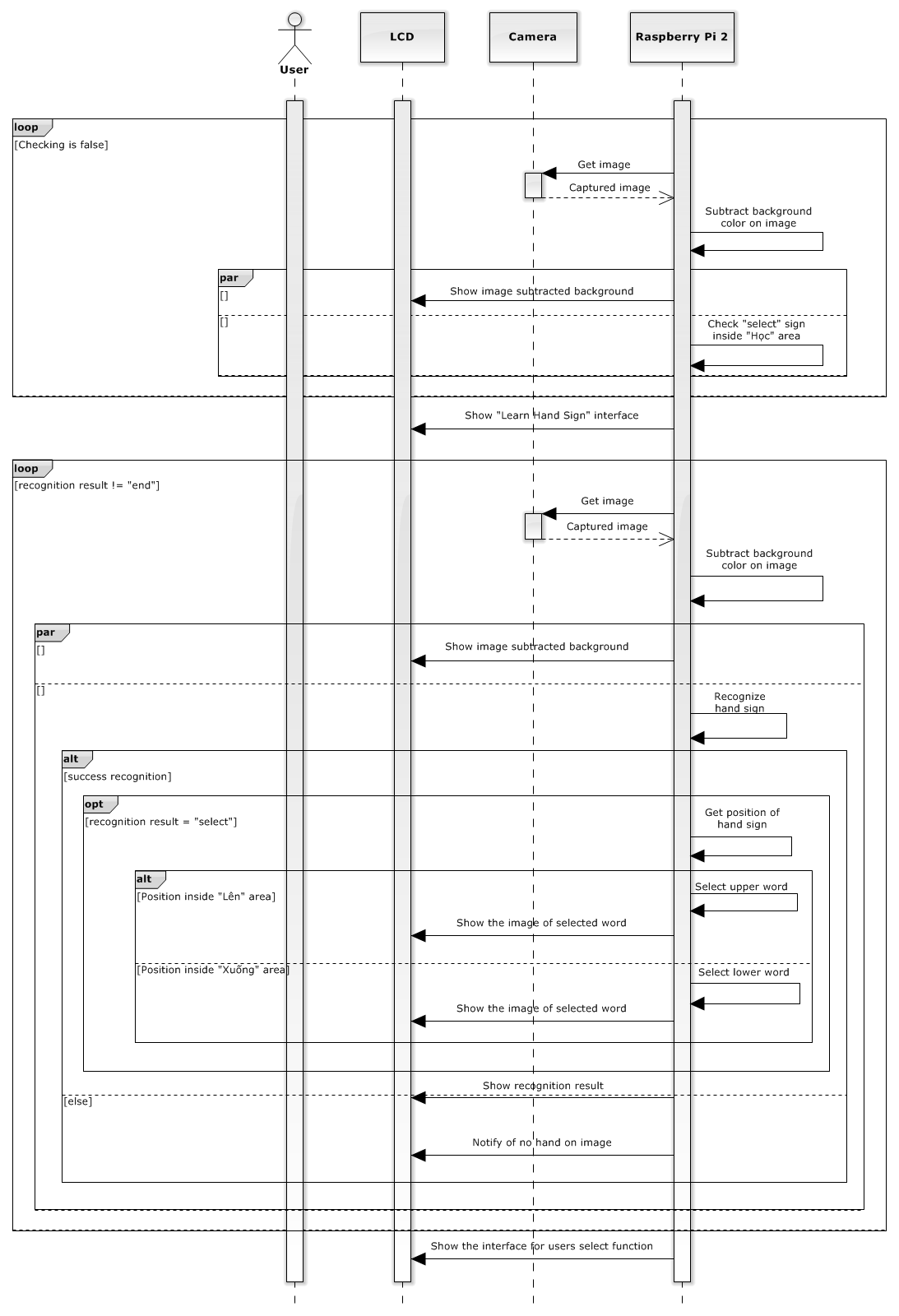


Figure 19 : Learn Hand Sign Language Sequence Diagram

#### Notify Battery Capacity



Figure 20: Notify Battery Capacity Sequence Diagram

#### Charge Battery



Figure 21: Charge Battery Sequence Diagram

## User Interface Design

### Subtract Background Color



Figure 22: Subtract Background Color

**Fields**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **No** | **Field Name** | **Description** | **Read only** | **Mandatory** | **Control Type** | **Data Type** | **Length** |
| 1 | BatteryCapacity | Image describing current battery capacity | Yes | Yes | QLabel | QImage | N/A |
| 2 | NotifyMessage | Notify of current situation | Yes | Yes | QLabel | QString | N/A |
| 3 | Timer | Countdown timer of current function | Yes | Yes | QLabel | QString | N/A |
| 4 | CapturedImage | Image captured from camera | Yes | Yes | QLabel | QImage | N/A |
| 5 | TestingStatus | Current status of background subtraction step | Yes | Yes | QLabel | QString | N/A |

**Buttons/Hyperlinks**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No** | **Function** | **Description** | **Validation** | **Outcome** |
|  |  |  |  |  |

### Recognize Hand Sign Language



Figure 23: Recognize Hand Sign Language

**Fields**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **No** | **Field Name** | **Description** | **Read only** | **Mandatory** | **Control Type** | **Data Type** | **Length** |
| 1 | BatteryCapacity | Image describing current battery capacity | Yes | Yes | QLabel | QImage | N/A |
| 2 | NotifyMessage | Notify of current situation | Yes | Yes | QLabel | QString | N/A |
| 3 | Timer | Countdown timer of current function | Yes | Yes | QLabel | QString | N/A |
| 4 | CapturedImage | Image captured from camera | Yes | Yes | QLabel | QImage | N/A |
| 5 | RecognitionResult | The result of current hand sign captured from camera | Yes | Yes | QLabel | QString | N/A |
| 6 | EntireContent | The whole content of recognition process | Yes | Yes | QLabel | QString | N/A |

**Buttons/Hyperlinks**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No** | **Function** | **Description** | **Validation** | **Outcome** |
|  |  |  |  |  |

### Learn Hand Sign Language



Figure 24: Learn Hand Sign Language

**Fields**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **No** | **Field Name** | **Description** | **Read only** | **Mandatory** | **Control Type** | **Data Type** | **Length** |
| 1 | BatteryCapacity | Image describing current battery capacity | Yes | Yes | QLabel | QImage | N/A |
| 2 | NotifyMessage | Notify of current situation | Yes | Yes | QLabel | QString | N/A |
| 3 | RecognitionResult | The result of current hand sign captured from camera | Yes | Yes | QLabel | QString | N/A |
| 6 | CapturedImage | Image captured from camera | Yes | Yes | QLabel | QImage | N/A |
| 7 | ListHandSigns | List of hand signs which is loaded from database | Yes | Yes | QListWidget | QStringList | N/A |
| 8 | HandSignImage | Image describing gesture of selected hand sign | Yes | Yes | QLabel | QImage | N/A |

**Buttons/Hyperlinks**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No** | **Function** | **Description** | **Validation** | **Outcome** |
| 4 | Select Hand Sign | Move the selection to the upper hand sign. | N/A | Upper hand sign is selected and its image is shown. |
| 5 | Select Hand Sign | Move the selection to the lower hand sign. | N/A | Lower hand sign is selected and its image is shown. |

### Select



Figure 25: Select

**Fields**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **No** | **Field Name** | **Description** | **Read only** | **Mandatory** | **Control Type** | **Data Type** | **Length** |
| 1 | BatteryCapacity | Image describing current battery capacity | Yes | Yes | QLabel | QImage | N/A |
| 2 | NotifyMessage | Notify of current situation | Yes | Yes | QLabel | QString | N/A |
| 5 | CapturedImage | Image captured from camera | Yes | Yes | QLabel | QImage | N/A |
| 6 | InstructionImage | Image guides the way selecting function | Yes | Yes | QLabel | QImage | N/A |

**Buttons/Hyperlinks**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No** | **Function** | **Description** | **Validation** | **Outcome** |
| 3 | Select Function | Select hand sign recognition function. | N/A | Interface of hand sign recognition function is initiated |
| 4 | Select Function | Select learning hand sign language function | N/A | Interface of learning hand sign language function is initiated |

1. **Database Design**
   1. **Logical Diagram**



* 1. **Data Dictionary**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Entity name** | **Attributes** | **Description** | **Domain** | **Null** |
| HandSign | id | Unique identifier of the hand sign. | Integer | No |
| content | The translated content of the handsign | Text | No |
| link | The image link lead to images describing the hand sign. | Text | No |

### Hardware Detailed Description

#### Raspberry Pi B2



Figure 26: Raspberry Pi B2 Kit

The Raspberry Pi is a low cost, **credit-card sized computer** that plugs into a computer monitor or TV, and uses a standard keyboard and mouse. It is a capable little device that enables people of all ages to explore computing, and to learn how to program in languages like Scratch and Python. It’s capable of doing everything you’d expect a desktop computer to do, from browsing the internet and playing high-definition video, to making spreadsheets, word-processing, and playing games.

***FOR***

* Quad-core CPU
* Backwards compatible
* More RAM
* Will fit existing cases

***AGAINST***

* Could prove intimidating for Linux newcomers
* No micro-USB adapter included

*Raspberry Pi B2 specification:*

* SoC: Broadcom 2836
* CPU: Quad-core ARM7 800MHz
* GPU: Videocore IV 250MHz  
  Memory: 1GB
* GPIO: 40 pin
* Ports: 4x USB 2.0, 100BaseT Ethernet, HDMI and MicroSD card
* Size: 85.60 × 56.5mm (about 3.2 x 2.1-inch)

#### Create a Portable System

Raspberry use input is 5v and 1A so; we do not use battery AA or AAA for Raspberry. We choose lipo battery use for Raspberry Pi B2 kit because lipo battery supplies 11.1v and 1A.



Figure 27: Lipo Battery

But Raspberry use input is 5v and 1A so, we need a circuit transformer convert from 11.1v to 5v.

That reason why, we choose LM2576ADJ - 3A UNI REG Board.



Figure 28: LM2576ADJ - 3A UNI REG Board

**Overview:**

* UNI-REG board allows changing voltages from 7-23V AC (or 9-32V DC) to 5V, 4V, 3.3V, 2.7V or 1.8V.
* Circuit Board using LM2576 - Step-Down Voltage Regulator.
* On-board screw-terminals are available for easy connection.

**Description:**

LM2576ADJ - UNI REG Board uses voltage regulator IC provides functional step-down (buck) switching regulator, capable of responding and changing load voltage lines are excellent. This is the ideal motherboard for projects requiring high voltage switching from lower to AC (DC).

Circuit Board accepts 7-23V AC input voltage (9-32V DC or), and stable output 5V, 4V, 3.3V, 2.7V or 1.8V DC, suitable for most electronics projects. The output voltage is selected via a jumper on the board. Compact and affordable, this board is perfect for use when switching power supplies are needed for your embedded project.



Figure 29 : ???



Figure 30 : IC LM2576HV-ADJ

VOUT = VREF (1+ R2/R1)

R2 = R1(VOUT/VREF  - 1)

Where VREF = 1.23v, R1 between 1k and 5k

#### The Battery Capacity Display Circuit

**Components of the circuit:**

- Battery lipo

- LT084

- 1 diode zener 5.1v (1N4733)

- 3 resistors 220 ohm

- 2 resistors 3.3k ohm

- 2 resistors 2.2k ohm

- 5 resistors 1k ohm

- 4 LED

##### LT084

The TL084 JFET-input operational amplifier family is designed to offer a wider selection than any previously developed operational amplifier family. Each of these JFET-input operational amplifiers incorporates well-matched, high-voltage JFET and bipolar transistors in a monolithic integrated circuit. The devices feature high slew rates, low input bias and offset currents, and low offset voltage temperature coefficient.



Figure 31 : LT084



Figure 32 : ???

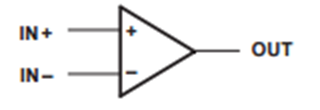


Figure 33 : ???

##### Diode zener

Zener diode, also known as voltage regulator diode, is a semiconductor diode work in reverse polarity mode on the breakdown voltage (breakdown). This voltage is also called Zener voltage avalanche or cascade (avalanche). At that voltage value little changed. It was built so that the reverse polarity, the Zener diodes will pin a fixed voltage level nearly equal to the value indicated on the diode, do Stabilizers of circuit.

When biased diodes Zener diodes operate like normal. When the polarity invert, at first only a small electric current through the diode truth. But if the voltage is increased to a value inversely adaptation: Vnguoc = Vz (Vz: Zener voltage), the current through diodes increase, but the voltage between the two ends of the diodes hardly change, so-called Zener effect.



Figure 34 : ???



Figure 35 : ???

##### Principles of circuit



Figure 36 : ???

#### Connecting Components In System



Figure 37 : Connecting Components



Figure 38 : ???



Figure 39 : ???

## Algorithms

### Background Subtraction

#### Definition

Background subtraction is the way that the system keeps only the hand on images captured from camera, and then the system can detect and keep track the hand through camera.

#### Define Problem

Images captured from camera that is quite complex in color and these maybe contains the user’s hand or not so we need to remove the colors not belong to the hand for detecting and recognizing.

#### Goal

* This solution must produce an image containing only the hands within black background and a binary image.
* Example:
  + Image containing the hands



Figure 40 : The first goal of background subtraction algorithm

* + Binary Image containing contour



Figure 41 : The second goal of background subtraction algorithm

#### Solution

To solve this problem, we need to have a background image as a sample, then we will remove the colors belongs to the sample from the following images captured from camera.

We should follow these steps to process background subtraction:

\* Sampling background image

* Specify sample background color:
  + Capture an image as background sample.



* + Blur background image to smooth and balance color.



* + Convert BGR background sample image to LAB sample image.



* + Get all color pixels from lab sample image.
* Define color range of LAB sample color:
  + With lightness dimension of LAB color space, the range is

[value – 40, value + 20]

* + With A and B for the color-opponent dimensions, the range is

[value – 15, value + 15]

\* Process the following image to subtract background color

* Create a binary image with black background.
* Preprocess image
  + Blur image



* + Convert to LAB image



* Check color pixels of the following images is between sample color range:
  + With every single color pixel, if lightness dimension (L) and color-opponent dimensions (A,B) is in color range of background, that means this pixel does not belong to hand.
* Remove the color pixels is between color range:
  + Change all pixels not belong to hand to black color.



* Remove noises and smooth the contour:
  + After the previous step, we have a binary image containing some noises and the contour is not smooth so we need to remove them by morphology transformations.
* Find the hand in binary image:
  + Now we have a binary image containing not only hand but also arm or something else so we need to find out the hand by finding the biggest contour.



### Definitions in image processing

#### RGB Image

RGBimage is an image in which each color pixel is described based on the RGB color model. A color in the RGB color model is described by indicating how much of each of the red, green, and blue is included. The color is expressed as an RGB triplet (r,g,b), each component of which can vary from zero to a defined maximum value. If all the components are at zero the result is black; if all are at maximum, the result is the brightest representable white.



Figure 42 : Three spaces of RGB image

These ranges may be quantified in several different ways:

* From 0 to 1, with any fractional value in between. This representation is used in theoretical analyses, and in systems that use floating point representations.
* Each color component value can also be written as a percentage, from 0% to 100%.
* In computers, the component values are often stored as integer numbers in the range 0 to 255, the range that a single 8-bit byte can offer. These are often represented as either decimal orhexadecimal numbers.
* High-end digital image equipment are often able to deal with larger integer ranges for each primary color, such as 0..1023 (10 bits), 0..65535 (16 bits) or even larger, by extending the 24-bits (three 8-bit values) to [32-bit](https://en.wikipedia.org/wiki/32-bit), [48-bit](https://en.wikipedia.org/wiki/48-bit), or 64-bit units (more or less independent from the particular computer's word size).

For example, brightest saturated **red** is written in the different RGB notations as:

|  |  |
| --- | --- |
| Notation | RGB triplet |
| Arithmetic | (1.0, 0.0, 0.0) |
| Percentage | (100%, 0%, 0%) |
| Digital 8-bit per channel | (255, 0, 0) or sometimes  #FF0000 (hexadecimal) |
| Digital 16-bit per channel | (65535, 0, 0) |

( [https://en.wikipedia.org/wiki/RGB\_color\_model](http:///h) )

#### LAB Image

LAB image is an image in which each color pixel is described based on the Lab color space model.

In this model, color space have one channel for Luminance (lightness) (L) and two color channels (a and b). The a axis extends from green (-a) to red (+a) and the b axis from blue (-b) to yellow (+b). The brightness (L) increases from the bottom to the top of the three-dimensional model.



Figure 43 : Three-dimensional model of LAB image

One of the most important attributes of the Lab image is device independence. This means that the colors are defined independent of their nature of creation or the device they are displayed on.

#### Binary Image

Binary image is a digital image that has only two possible values for each pixel. Typically, the two colors used for a binary image are black represented by 0 and white represented by 1 , though any two colors can be used.



Figure 44 : Example of binary image

Binary images are used in many applications because they are the simplest to process.

#### Contour

Contours can be explained simply as a curve joining all the continuous points (along the boundary), having same color or intensity.

The contour pixels are generally a small subset of the total number of pixels representing a pattern. Therefore, the amount of computation is greatly reduced when we run feature extracting algorithms on the contour instead of on the whole pattern. The contours are a useful tool for shape analysis and object detection and recognition.

For example, below picture contains three contours which is red color.



Figure 45 : Examle of contour

### Used Image Processing Algorithm

#### Blur Image – Gaussian Blur Algorithm

##### Gaussian Blur common mechanic

First, we choose a kernel size

* A kernel size is a matrix with width and height can differ but they must be positive and odd.
* Why kernel width and height both must be odd? Because we can get a center location in the matrix.
* For example:



Figure 46 : kernel size 3x3



Figure 47 : kernel size 5x5



Figure 48 : kernel size 3x5

Move the kernel size on the image matrix all over pixels. At single movement, the kernel’s center point will take the average value of its surrounding points.

* Movements

Figure 49 : Kernel size movements

* At single movement

Assume we have a value matrix taken from the kernel size 3x3



Figure 50 : Value matrix taken from the kernel size 3x3

On the above matrix, the center point value is 2 and its surrounding points are 1. The center point will take the average value of its neighbor, it will be 1.



Figure 51 : Value matrix after applying Gaussian blur algorithm

The kernel size is large, the blur effect is strong.



Figure 52 : Original RGB Image



Figure 53 : RGB Image applied Guassian Blur with kernel size 7x7



Figure 54 : RGB Image applied Guassian Blur with kernel size 21x21

##### Weight Matrix

To preserve boundaries and and edges better, we have a rule, the closer the points in distance, the larger the weight. In other words, the original’s pixel must receive the heaviest weight and its neighboring pixels receive smaller weights as their distance increases.

Gaussian provides a formula to produce weighted matrix:

where x, y is the distance from the center point in x-axis and y-axis respectively, and is standard deviation. Standard deviation is computed from size of kernel size as sigma = 0.3 \* ( ( ksize - 1 ) \* 0.5 -1 ) + 0.8 or we can choose a specific standard deviation.

For example

* we will calculate the weight matrix 3x3 with standard deviation = 1.5
  + The coordinate of the center point



Figure 55 : the coordinate of the center point with matrix 3x3

* + Apply the above formula for each point



Figure 56 : the weighted 3x3 matrix before divide the sum of the weight

* + The above 9 values will divide the sum of the weight of these 9 points which is 0.4787147



Figure 57 : the final weighted 3x3 matrix

* Apply this weight matrix to calculate the gaussian blur
  + Assume we have a matrix value (3x3) of grayscale image



Figure 58 : The 3x3 matrix value of grayscale image

* + Each point multiplies its weight value (above weighted matrix)



Figure 59 : Multiplying the matrix value with the weighted matrix

* + The final result



Figure 60 : Matrix values after applying Gaussian blur

##### Convert RGB image to LAB image

In case of 8-bit and 16-bit images, R, G, and B are converted to the floating-point format and scaled to fit the 0 to 1 range.RGB values in a particular set of primaries can be transformed to CIE via a 3x3 matrix transform



CIE LAB is based directly on CIE XYZ and is an attempt to linear the perceptibility of color differences. The non-linear relations for L\*, a\*, and b\* are intended to mimic the logarithmic response of the eye. Coloring information is referred to the color of the white point of the system, subscript n

X  \leftarrow X/X_n,  \text{where} X_n = 0.950456

Z  \leftarrow Z/Z_n,  \text{where} Z_n = 1.088754

L  \leftarrow \fork{116*Y^{1/3}-16}{for $Y>0.008856$}{903.3*Y}{for $Y \le 0.008856$}

a  \leftarrow 500 (f(X)-f(Y)) + delta

b  \leftarrow 200 (f(Y)-f(Z)) + delta

f(t)= \fork{t^{1/3}}{for $t>0.008856$}{7.787 t+16/116}{for $t\leq 0.008856$}

delta =  \fork{128}{for 8-bit images}{0}{for floating-point images}

This outputs 0 \leq L \leq 100, -127 \leq a \leq 127, -127 \leq b \leq 127. The values are then converted to the destination data type 8-bit images

L  \leftarrow L*255/100, \; a  \leftarrow a + 128, \; b  \leftarrow b + 128

Example:



Figure 61 : Example of converting RGB image to LAB image

##### Morphological operations - Opening on binary images

Morphological operations are affecting the form, structure or shape of an object. A morphological operation on a binary image creates a new binary image in which the pixel has a non-zero value only if the test is successful at that location in the input image.



Figure 62 : Example of Morphological operations – Opening

Morphological operations have 2 principal: *dilation* and *erosion.* These operations can be customized by the selection of the *structuring element* which determines exactly small binary image will be dilated or eroded.

Structuring element can be a small matrix of pixels, each with a value of zero or one:

* The matrix dimensions specify the *size* of the structuring element.
* The pattern of ones and zeros specifies the *shape* of the structuring element.
* An *origin* of the structuring element is usually one of its pixels, although generally the origin can be outside the structuring element.



Dilation allows objects to expand, thus potentially filling in small holes and connecting disjoint objects. The dilation process is performed by laying the structuring element B on the image A **(Notation A⊕ B*)***and move it across each pixel of binary image. When move to a pixel, this pixel is origin. Dilation operation will perform:

**1.** If the origin of the structuring element coincides with a 'white' pixel in the image, there is no change; move to the next pixel.

**2.** If the origin of the structuring element coincides with a 'black' in the image, make black all pixels from the image covered by the structuring element.





Original Image *structuring element* Image after dilation



Figure 63 : Illustration of the dilation process

Erosion shrinks objects by etching away (eroding) their boundaries. The *erosion* process is similar to dilation**(Notation *A* Θ *B*)** but it perform these steps:

**1.** If the origin of the structuring element coincides with a 'white' pixel in the image, there is no change; move to the next pixel.

**2.** If the origin of the structuring element coincides with a 'black' pixel in the image, and at least one of the 'black' pixels in the structuring element falls over a white pixel in the image, then change the 'black' pixel in the image (corresponding to the position on which the center of the structuring element falls) from ‘black’ to a 'white'.



Original Image *structuring element* Image after Erosion

Figure 64 : Illustration of the erosion process

These two basic operations, dilation and erosion, can be combined into more complex sequences. *Opening* consists of an erosion followed by a dilation and can be used to eliminate all pixels in regions that are too small to contain the structuring element. **Notation: A◦B = (AΘB) ⊕ B**

##### Find Contour

Apply Theo Pavlidis' Algorithm :

Given a image digital have group of black pixels, on a background of white pixels locate a black pixel and declare it as your "start" pixel.

You can choose any black boundary pixel to be your start pixel as long as when you're initially standing on it, your left adjacent pixel is NOT black. In other words, you should make sure that you enter the start pixel in a direction which ensures that the left adjacent pixel to it will be white ("left" here is taken with respect to the direction in which you enter the start pixel).

We have are a bug (ladybird) standing on the **start** pixel as in Figure below.

Throughout the algorithm, the pixels which interest you at any time are the 3 pixels in front of you i.e. **P1, P2** and **P3** shown in ***Figure 1***. (We will define **P2** to be the pixel right **in front** of you , **P1** is the pixel adjacent to **P2** from the left and **P3** is the right adjacent pixel to **P2**).



Figure 65 : starting pixel of find contour

The most important thing in Pavlidis' algorithm is your "sense of direction". The left and right turns you make are with respect to your current positioning, which depends on the way you entered the pixel you are standing on. In example , we choose direction from the bottom up which you enter the **start** pixel.

First, check pixel P1. If P1 is black, then declare P1 to be your current boundary pixel and move one step forward followed by one step to your current left to land on P1.

(the order in which you make your moves is very important)

Figure 2 below demonstrates this case. The path you should follow in order to land on P1 is drawn in blue.



Figure 66 : demonstrating the path to P1

Only if P1 is white proceed to check P2...

If **P2** is black, then declare **P2** to be your current boundary pixel and **move one step forward** to land on **P2**.

***Figure 3*** below demonstrates this case. The path you should follow in order to land in **P2** is drawn in blue.



Figure 67 : demonstrating the path to P2

Only if both P1 and P2 are white proceed to check P3...

If **P3** is black, then declare **P3** to be your current boundary pixel and move one step to your right followed by one step to your current leftas demonstrated in Figure 4 below



Figure 68 : demonstrating the path to P3

If all 3 pixels in front of you are white: Then, you rotate (while standing on the current boundary pixel) 90 degrees clockwise to face a new set of 3 pixels in front of you. Afterwards you do the same check on these new pixels as you've done before.

If all of these 3 pixels are still white: then rotate again through 90 degrees clockwise while standing on the same pixel.

You can rotate 3 times (each through 90 degrees clockwise) before checking out the whole Moore neighborhood of the pixel. If you rotate 3 times without finding any black pixels, this means that you are standing on an isolated pixel i.e. not connected to any other black pixel.

The algorithm terminate when :

- As mentioned above, the algorithm will allow you to rotate 3 times (each through 90 degrees clockwise) after which it will terminate and declare the pixel an isolated one, OR

- It visiting the start pixel a third time

##### Flow chart



Figure 69 : flow of background subtraction algorithm

### Features Extraction

#### Definition

Features extraction is the way to specify 81 features of hand for recognition.

#### Define Problem

At this time, we just have an image containing only hand and a binary image containing the hand contour and we need to find the best features from these two images.

#### Goal

* This solution must produce three binary images containing 81 features for hand sign recognition.
* Example:



Figure 70 : The goal of features extraction

#### Solution

To solve this problem, we need to convert the BGR image containing hand which is produced from the background color subtraction into binary images which can describe most of features.

We should follow the following steps to extract the features of hand:

* Convert the BGR image containing the hand to binary image containing features:
  + To process it we will apply the Adaptive Gaussian Threshold algorithm.

* Produce the binary image containing the hand palm from the binary image containing contour:
  + We will find the point inside the contour and calculate the nearest distance from the point to the contour, and then we find the farthest point from those nearest distance.
  + From the farthest point and the distance, we can draw a circle similar to the hand palm



* Produce the binary image containing the finger lines from the binary image containing contour:
  + First, we find the convex hull of the contour
  + After that, we find the convexity defects of the contour from the convex hull.
  + From these convexity defects we can draw the finger lines



* Calculate the features from these binary images:
  + Calculate zoning features





* + Calculate the ratio between height and width of hand



* + Calculate the ratio between radius of palm and size of hand



* + Calculate the degree features of hand fingers

#### Definitions in image processing

##### Convex Hull

Convex is a shape or set if for any two points that are part of the shape, the whole connecting line segment is also part of the shape. For any subset of the plane (set of points, rectangle, simple polygon), its convex hull is the smallest convex set that contains that subset



Figure 71 : example of convex hull

##### Zoning features

Zoning features are efficient statistical features that provides low complexity for different characters of an object. They are defined by the density of black pixels in several zones we divided in an image containing object.

Zoning features is good that depends on zones we selected to calculate the density of black pixels.

For an example of 4x4 zones:



Figure 72 : Example of 4x4 zones

#### Used Image Processing Algorithm

##### Convert to binary image - Adaptive Gaussian Threshold

This algorithm applies an adaptive threshold to a matrix for transforming a grayscale image to binary image. According to the formulae:

where x, y is the position of a pixel in the matrix and T(x,y) is a threshold calculated individually for every single pixel.

A transformation is good that depends on how the threshold is calculated.

We apply the method using Weight Matrix (explained in Gaussian Blur Algorithm), the threshold is a weighted sum of the MATRIX\_SIZE x MATRIX\_SIZE neighboring pixel of the origin pixel.

This algorithm is a useful technique to draw features of images. For example, we have an RGB image containing hand with black background.



Figure 73 : RGB image containing hand with black background

After image processing, we have a binary image containing features of the hand.



Figure 74 : binary image after converting to binary image by adaptive gaussian

##### Find the convex hull

Implement Melkman's algorithm 1987

It takes the first three vertices (starting anywhere) and sets up the current convex hull, i.e. the triangle formed. In the picture below, they happen to form a right turn (clockwise). Vertices are stored in a double ended queue (deque) : (bottom) 3-1-2-3 (top). Notice that if we take the last three vertices of the bottom in the order 2,1,3 we get a left turn. If we take the last three vertices from the top, in the order 1,2,3 we get a right turn. This is a property that we wish to maintain: in general as we read the deque from bottom to top, we get the hull in clockwise order, and as we read from top to bottom we get a counterclockwise order.

Now the next vertex, V, could be in the red/green/blue/yellow regions



Figure 75 : example of the steps finding convex hull

If V is in the yellow region, ignore it and all following vertices until one emerges into the other regions. Call the emerging vertex V. If V is not yellow, we must add it to the deque on both sides, because it will be on the current hull. However we must ensure that we preserve our clockwise/counterclockwise property if this is to be done:

If V is in the red region, then 2,3,V form a left turn. Backtrack/delete vertices from the top of the deque (i.e. 3, then maybe 2, etc), until a right turn is formed by the last 3 vertices.

If V is in the green region, then 1,3,V form a right turn. Backtrack/delete vertices from the bottom of the deque (i.e. 3, then maybe 1, etc), until a left turn is formed by the last 3 vertices. Notice that this case is symmetric to the red region.

If blue, follow the instructions for both red and green.

Now the deque structure is correct and we can process the next point.

The figure below shows these regions in a more general case. Vertices on the hull have circles on them. (N) is the last vertex added. The next vertex could be anywhere in the colored regions. Note that you can find these regions by looking at N, and its neighbors on the hull, which are conveniently represented at the two ends of the deque



Figure 76 : example of the steps finding convex hull

##### Find the convexity defects

After find convex hull of shape, we have blue points as result.



Figure 77 : example of finding the convexity defects

Then, we can find the convexity defects (red point) of this shape by:

Move each point (red point ) on contour of shape and measure distance from this point to green line connect 2 blue points.



Figure 78 : example of finding the convexity defects

The convexity defect is the farthest point to green line.

With two blue points, we just find out one convexity defect point.

##### Resize Image - Nearest Neighbor Image Scaling

The principle in image scaling is to have a reference image and using this image as the base to construct a new scaled image. The constructed image will be smaller, larger, or equal in size depending on the scaling ratio.

We say: w1 and h1 are the width and height of an origin image, whereas w2 and h2 are the width and height of new image . Calculating the ratio for both horizontal and vertical plane is given by,



In new image, value of each pixel Y(x2, y2) determined by the

value of the pixel X(x1,y1) where :

x1 = rounded down of x2. x\_ratio

y1 = rounded down of y2. y\_ratio





Origin image 3X3 New image 6X6

Figure 79 : Illustration of the resizing 200%

 Original image Scale 200%

Figure 80 : Example of Image Scaling

##### Calculate the degree features of hand fingers

The goal of this algorithm is to define the degree category of each finger line. The degree category is the degree between the line and the x-axis and we will have eight categories:

* Category 1: The degree is between 0 and 45
* Category 2: The degree is between 46 and 90
* Category 3: The degree is between 91 and 135
* Category 4: The degree is between 136 and 180
* Category 5: The degree is between -45 and -1
* Category 6: The degree is between -90 and -46
* Category 7: The degree is between -135 and -91
* Category 8: The degree is between -179 and -136

For example, below picture we have four lines:



Figure 81 : The example of degree features extraction

where the red point is the origin point and the blue point is the distance point.

Then, calculate the degree and classify them.



Figure 82 : The example of degree features extraction

After classify the line into these eight categories, the feature is the number of lines of each category.

##### Flow chart



Figure 83 : the flow of features extraction algorithm

### Support Vector Machine

#### Definition

Support Vector Machine is a useful technique for data classification that analyzes data and recognizes patterns.

#### Define problem

Now we have sets of features of the hand signs and we must rely on those sets to recognize them. Therefore, we use library for Support Vector Machine (LIBSVM) to produce a model by basing on the training hand sign feature data and then rely on the model to predict the target values of the other hand sign features.

#### Goal

This library for Support Vector Machine algorithm builds a model that assigns new examples into specify category and when new hand sign are mapped into that same space and predicted to belong to a category.

#### Support Vector Machine Algorithm

A Support Vector Machine is a discriminative classifier formally defined by constructing set of hyperplanes in a space has high dimension or infinite dimension. Furthermore, the algorithm should output a hyperplane which is optimal to categorize new examples. An optimal hyperplane is a hyperplane has the largest distance to the nearest training data point of classes.

For example:



Figure 84 : Example of linear SVM technique

* In the above picture, we see that there are a lot of lines can resolve the problem but which is better than others?
* The algorithm defines a criterion to estimate the worth of these lines that is the one that represents the largest separation between two set of points, so we choose the optimal separating hyperplane maximizes the distance from it to the nearest data point on each side.



Figure 85 : Example of linear SVM technique

The above example is one technique of Linear SVM. Besides that, there is another technique supported by SVM that is nonlinear classification. This is the way to create nonlinear hyperplanes by applying the kernel trick to margin hyperplanes.

We chose the nonlinear SVM technique because it can map samples into a higher dimensional space and the number of features and sample is quite small. The following image is an example of nonlinear SVM technique:



Figure 86 : Example of non-linear SVM technique

With multiclass, SVM algorithm aims a common method that is reducing the single multiclass problem into binary classifiers which separate between one class and the remains. That is called one-versus-all. The following example:



Figure 87 : Example of non-linear SVM technique

Besides that, to have a good SVM model that relies on the number of sample, how many cases the samples can cover and the most important condition is the features we select.

# Report No.5 System Implementation & Test

## Introduction

### Overview

This section provides in detail all necessary information about implementation information and testing procedure of VSLR includes test plans, test cases, test procedures and test result.

### Test Approach

#### Method

* *Black-box testing*: We examines the functionality of the system without peering into its internal structures or workings. This testing can dominate integration testing as well.



Figure 88 : Black-box testing

#### Goal

* To validate that the application works as the user will be operating it, then find out incorrect or missing functions, interface errors, behavior and performance errors.

## Database Relationship Diagram

### Physical Diagram



Figure 89 : Physical Database Diagram

### Data Dictionary

|  |  |
| --- | --- |
| **Entity Data dictionary: describe content of all entities** | |
| **Entity Name** | **Description** |
| HandSign | Describe the hand sign words in the system. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Entity name** | **Attributes** | **Description** | **Domain** | **Null** |
| HandSign | id | Unique identifier of the hand sign. | Integer | No |
| content | The translated content of the handsign | Text | No |
| link | The image link lead to images describing the hand sign. | Text | No |

## Test Plan

### Test items

*We have a main test phase : Integration test:*

* Integration Testing: We test the integration of the code modules developed and interaction with hardware. The integration testing starts at the bottom level. Each component at lower hierarchy is tested individually; then the components that rely upon these are tested.

### Features to be tested

*Integration Test includes the following features:*

* Background Color Subtraction
* “Selecting Function” function
* “Hand Sign Language Recognition” function
* “Learning Hand Sign Language” function
* “Charging Battery” function
* “Monitoring Battery Capacity” function

### Features not to be tested

N/A

### Environmental needs

* A complete system with fully devices and functions.
* An environment with stable light, and background is not complex in color.

### Test case pass/fail criteria

* Every test case must describe what expected output are to pass that specific test.
* Test coverage must be at least 90%.
* All test case must pass.

## 

## Integration Test Specifications

### “Background Color Subtraction” Test



Figure 90 : Components of the Background Color Subtraction

#### Integration test case

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **ID** | **Test Item(s)** | **Input specification** | **Expected Output** | **Condition** |
| BCSTC01 | Display Screen to Sampling Background Color | Switch on | Displays notify “Người dùng vui lòng di chuyển ra khỏi vùng camera đang theo dõi”. | N/A |
| Shows the images captured from camera on the interface for users. |
| Shows the count down time and counts down from 5 by a second. |
| Show the message “Đang tiến hành". |
| BCSTC02 | Display Screen to Sampling Background Color | Switch on | Displays notify dialog “”. | Camera is not active. |
| BCSTC03 | Background Color -> Testing Color Subtraction | The background is fixed and user waiting for the countdown time counts to 0. | A notify “Vui lòng điều chỉnh bàn tay của bạn theo kí hiệu “kiểm tra” trong hướng dẫn” is shown. | Test case BCSTC01 is executed |
| Shows the images subtracting background color on the interface for users |
| Countdown time is shown by seconds from 5. |
| Shows the message “Đang tiến hành" |
| BCSTC04 | Background Color -> Testing Color Subtraction | The background is continuously changing and user waiting for the countdown time counts to 0. | A notify “Vui lòng điều chỉnh bàn tay của bạn theo kí hiệu “kiểm tra” trong hướng dẫn” is shown. | Test case BCSTC01 is executed |
| Shows the images subtracting background color on the interface for users |
| Countdown time is shown by seconds from 5. |
| Shows the message “Đang tiến hành" |
| BCSTC05 | Testing Color Subtraction | Showing right “testing” hand sign inside camera area | Counting down continues. | Test case BCSTC03 or BCSTC04  is executed |
| Show a message “Thành Công”. |
| BCSTC06 | Testing Color Subtraction | Showing wrong “testing” hand sign inside camera area | Counting down continues | Test case BCSTC03 or BCSTC04  is executed |
| Show a message “Đang tiến hành”. |
| BCSTC07 | Testing Color Subtraction | Don’t show hand sign | Counting down continues | Test case BCSTC03 or BCSTC04  is executed |
| Show a message “Đang tiến hành”. |
| BCSTC08 | Testing Color Subtraction | Showing right “testing” hand sign outside camera area | Counting down continues | Test case BCSTC03 or BCSTC04  is executed |
| Show a message “Đang tiến hành”. |
| BCSTC09 | Testing Color Subtraction | Showing wrong “testing” hand sign outside camera area | Counting down continues | Test case BCSTC03 or BCSTC04  is executed |
| Show a message “Đang tiến hành”. |
| BCSTC10 | Testing Color Subtraction -> Sampling Background Color | Waiting for the countdown time counts to 0. | Shows message “Thất bại” | Test case BCSTC06 or BCSTC07 or BCSTC08 or BCSTC09  is executed |
| A notify “Người dùng vui lòng di chuyển ra khỏi vùng camera đang theo dõi” is shown. |
| Show the images captured from camera on the interface for users . |
| Shows the count down time and counts down from 5 by a second |
| Show the message “Đang tiến hành". |
| BCSTC11 | Testing Color Subtraction -> Selecting Function Interface | Waiting for the countdown time counts to 0. | Selecting Function Interface is shown | Test case BCSTC05 is executed |

#### Integration test procedure TP1

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **ID** | **Purpose** | **Procedure Steps** | **Excuted By** | **Result** | **Test Date** | **Note** |
| BCSITP01 | Test the success background color subtraction flow can work correctly | 1. Execute test case BCSTC01  2. Execute test case BCSTC03  3. Execute test case BCSTC05  4. Execute test case BCSTC11 |  | Pass |  |  |
| BCSITP02 | Test the fail background color subtraction work flow correctly: show wrong “test” hand sign inside camera area, background is fixed | 1. Execute test case BCSTC01  2. Execute test case BCSTC03  3. Execute test case BCSTC06  4. Execute test case BCSTC10 |  | Pass |  |  |
| BCSITP03 | Test the fail background color subtraction work flow correctly: show right “testing” hand sign inside camera area, background is continuously changing | 1. Execute test case BCSTC01  2. Execute test case BCSTC04  3. Execute test case BCSTC05  4. Execute test case BCSTC10 |  | Pass |  |  |
| BCSITP04 | Test the fail background color subtraction work flow correctly : background is continuously changing, show wrong “test” hand sign inside camera area | 1. Execute test case BCSTC01  2. Execute test case BCSTC04  3. Execute test case BCSTC06  4. Execute test case BCSTC10 |  | Pass |  |  |
| BCSITP05 | Test the fail background color subtraction work flow correctly: don’t show hand sign, background is fixed | 1. Execute test case BCSTC01  2. Execute test case BCSTC03  3. Execute test case BCSTC07  4. Execute test case BCSTC10 |  | Pass |  |  |
| BCSITP06 | Test the fail background color subtraction work flow correctly: don’t show hand sign, background is continuously changing | 1. Execute test case BCSTC01  2. Execute test case BCSTC04  3. Execute test case BCSTC07  4. Execute test case BCSTC10 |  | Pass |  |  |
| BCSITP07 | Test the fail background color subtraction work flow correctly: show right “testing” hand sign outside camera area, background is fixed | 1. Execute test case BCSTC01  2. Execute test case BCSTC03  3. Execute test case BCSTC08  4. Execute test case BCSTC10 |  | Pass |  |  |
| BCSITP08 | Test the fail background color subtraction work flow correctly: show right “testing” hand sign outside camera area, background is continuously changing | 1. Execute test case BCSTC01  2. Execute test case BCSTC04  3. Execute test case BCSTC08  4. Execute test case BCSTC10 |  | Pass |  |  |
| BCSITP09 | Test the fail background color subtraction work flow correctly: show wrong “testing” hand sign outside camera area, background is fixed | 1. Execute test case BCSTC01  2. Execute test case BCSTC03  3. Execute test case BCSTC09  4. Execute test case BCSTC10 |  | Pass |  |  |
| BCSITP10 | Test the fail background color subtraction work flow correctly: show wrong “testing” hand sign outside camera area, background is continuously changing | 1. Execute test case BCSTC01  2. Execute test case BCSTC04  3. Execute test case BCSTC09  4. Execute test case BCSTC10 |  | Pass |  |  |
| BCSITP11 | Test “Background Color Subtraction” can handle when camera not active. | 1. Execute test case BCSTC02 |  | Pass |  |  |

### “Hand Sign Language Recognition” Test



Figure 91 : Components of the Hand Sign Language Recognition

#### Test Case Specification

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **ID** | **Test Item(s)** | **Input Specification** | **Output Specification** | **Condition** |
| HSRTC01 | Testing Color Subtraction -> Selecting Function | N/A | Show a notify “” in the groupbox “Thông Báo” | Background Color Subtraction is executed |
| The analyzed images show on the interface continuously. |
| Two white “Nhận Dạng” and “Học” area are drawn inside analyzed images |
| System shows an image guiding users to select function |
| HSRTC02 | Selecting Function | Show “select” hand sign outside “Nhận dạng” area | Nothing change | Test case HSRTC01 is executed |
| HSRTC03 | Selecting Function | Show hand sign different “select” hand sign inside “Nhận dạng” area | Nothing change | Test case HSRTC01 is executed |
| HSRTC04 | Selecting Function | Show hand sign different “select” hand sign outside “Nhận dạng” area | Nothing change | Test case HSRTC01 is executed |
| HSRTC05 | Selecting Function -> Hand Sign Recognition | Show “select” hand sign inside “Nhận dạng” area. | The system shows the hand sign recognition interface | Test case HSRTC01 is executed |
| The analyzed images show on the interface continuously |
| A notify “Hệ thống sẽ lưu lại kết quả nhận dạng sau 3 giây” is shown in groupbox “Thông Báo” |
| Countdown time is shown from 3 in groupbox “Thời Gian” |
| The system shows two groupbox “Nội dung toàn bộ” and “Kết Quả Hiện Tại” with empty content. |
| HSRTC06 | Hand Sign Language Recognition | Showing the “A” hand sign through camera | Countdown time counts down by second. | Test case HSRTC05 is executed |
| Groupbox “ Kết Quả Hiện Tại ” is shown with the result content “A” below. |
| HSRTC07 | Hand Sign Language Recognition | Don’t show hand sign through camera | Countdown time counts down by second. | Test case HSRTC05 is executed |
| Show a message “Không tìm thấy bàn tay!” in the group “Kết Quả Hiện Tại”. |
| HSRTC08 | Hand Sign Language Recognition | User keeps the “A” hand sign through camera and waiting for the countdown time counts to 0. | Groupbox “ Nội Dung Toàn Bộ ” is shown with the result content “A” | Test case HSRTC06 is executed |
| HSRTC09 | Hand Sign Language Recognition | Showing the “C” hand sign through camera and keeps when the countdown time counts to 0. | Groupbox “ Nội Dung Toàn Bộ ” is shown with the result content “C” | Test case HSRTC06 is executed |
| HSRTC010 | Hand Sign Language Recognition | Showing the “speak” hand sign through camera and keeps when the countdown time counts to 0. | Groupbox “ Nội Dung Toàn Bộ ” is still empty | Test case HSRTC06 is executed |
| HSRTC11 | Hand Sign Language Recognition | Showing the “B” hand sign through camera | Countdown time counts down by second. | Test case HSRTC08 is executed |
| Groupbox “ Kết Quả Hiện Tại ” is shown with the result content “B” |
| HSRTC12 | Hand Sign Language Recognition | User keeps the “B” hand sign through camera and waiting for the countdown time counts to 0. | Groupbox “ Nội Dung Toàn Bộ ” is shown with the result content “AB” | Test case HSRTC10 is executed |
| HSRTC13 | Hand Sign Language Recognition | Show “speak” hand sign inside the camera area and waiting for the countdown time counts to 0. | Reads “A” via LCD speaker | Test case HSRTC08 is executed |
| Groupbox “ Nội Dung Toàn Bộ ” is clear |
| HSRTC14 | Hand Sign Language Recognition | No hand inside the camera area and waiting for the countdown time counts to 0 | Countdown time counts down by second. | Test case HSRTC06 is executed |
| Show a message “Không tìm thấy bàn tay!” in the group “Kết Quả Hiện Tại”. |
| HSRTC15 | Hand Sign Language Recognition | Show “end” hand sign then show “speak” hand sign inside the camera area and waiting for the countdown time counts to 0 | Reads “A” via LCD speaker | Test case HSRTC08 is executed |
| Groupbox “ Nội Dung Toàn Bộ ” is clear |  |
| HSRTC16 | Hand Sign Language Recognition -> Selecting Function | Show “speak” hand sign then show “end” hand sign inside the camera area and waiting for the countdown time counts to 0 | Selecting Function interface is displayed | Test case HSRTC08 is executed |
| HSRTC17 | Hand Sign Language Recognition -> Selecting Function | Showing the “end” hand sign through camera and waiting for the countdown time counts to 0. | Selecting Function interface is displayed | Test case HSRTC08 is executed |

#### Integration Test Procedure TP3

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **ID** | **Purpose** | **Procedure Steps** | **Excuted By** | **Result** | **Test Date** | **Note** |
| HSRTP01 | Test fail “Selection Function”: “select” hand sign outside “Nhận dạng” area | 1. Execute test case HSRTC01  2. Execute test case HSRTC02 |  | **Pass** |  |  |
| HSRTP02 | Test fail “Selection Function”: hand sign different “select” hand sign inside “Nhận dạng” area | 1. Execute test case HSRTC01  2. Execute test case HSRTC03 |  | **Pass** |  |  |
| HSRTP03 | Test fail “Selection Function”: different “select” hand sign outside “Nhận dạng” area | 1. Execute test case HSRTC01  2. Execute test case HSRTC04 |  | **Pass** |  |  |
| HSRTP04 | Test the flow Hand Sign Recognition can work correctly show the “AB” content in groupbox “ Nội Dung Toàn Bộ ” | 1. Execute test case HSRTC01  2. Execute test case HSRTC05  3. Execute test case HSRTC06  4. Execute test case HSRTC08  5. Execute test case HSRTC12 |  | **Pass** |  |  |
| HSRTP05 | Test the flow Hand Sign Recognition can work correctly show and speak the “A” content | 1. Execute test case HSRTC01  2. Execute test case HSRTC05  3. Execute test case HSRTC06  4. Execute test case HSRTC08  5. Execute test case HSRTC13 |  | **Pass** |  |  |
| HSRTP06 | Test the flow Hand Sign Recognition can work correctly show and speak the “A” content | 1. Execute test case HSRTC01  2. Execute test case HSRTC05  3. Execute test case HSRTC06  4. Execute test case HSRTC08  5. Execute test case HSRTC15 |  | **Pass** |  |  |
| HSRTP07 | Test the flow Hand Sign Recognition can work correctly show the “C” content in groupbox “ Nội Dung Toàn Bộ ” | 1. Execute test case HSRTC01  2. Execute test case HSRTC05  3. Execute test case HSRTC06  4. Execute test case HSRTC09 |  | **Pass** |  |  |
| HSRTP08 | Test the flow Hand Sign Recognition can work correctly : show Selecting Function interface | 1. Execute test case HSRTC01  2. Execute test case HSRTC05  3. Execute test case HSRTC17 |  | **Pass** |  |  |
| HSRTP09 | Test the flow Hand Sign Recognition can work correctly : show Selecting Function interface | 1. Execute test case HSRTC01  2. Execute test case HSRTC05  3. Execute test case HSRTC16 |  | **Pass** |  |  |
| HSRTP10 | Test the flow Hand Sign Recognition can work correctly : show Selecting Function interface | 1. Execute test case HSRTC01  2. Execute test case HSRTC05  3. Execute test case HSRTC06  4. Execute test case HSRTC17 |  | **Pass** |  |  |
| HSRTP11 | Test the flow Hand Sign Recognition can work correctly : show Selecting Function interface | 1. Execute test case HSRTC01  2. Execute test case HSRTC05  3. Execute test case HSRTC06  4. Execute test case HSRTC16 |  | **Pass** |  |  |
| HSRTP12 | Test fail “Hand Sign Recognition” with No hand inside the camera area | 1. Execute test case HSRTC01  2. Execute test case HSRTC05  3. Execute test case HSRTC06  4. Execute test case HSRTC14 |  | **Pass** |  |  |
| HSRTP13 | Test fail “Hand Sign Recognition” with No hand inside the camera area | 1. Execute test case HSRTC01  2. Execute test case HSRTC05  3. Execute test case HSRTC07 |  | **Pass** |  |  |

### “Learning Hand Sign Language” Test



Figure 92: Components of the Learning Hand Sign Language

#### Test case specification

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **ID** | **Test Item(s)** | **Input Specification** | **Output Specification** | **Condition** |
| LHSTC01 | Testing Color Subtraction -> Selecting Function | N/A | Show a notify “” in the groupbox “Thông Báo” | Background Color Subtraction is executed |
| The analyzed images show on the interface continuously. |
| Two white “Nhận Dạng” and “Học” area are drawn inside analyzed images |
| System shows an image guiding users to select function |
| LHSTC02 | Selecting Function | Show “select” hand sign outside “ Học ” area | Nothing change | Test case LHSTC01is executed |
| LHSTC03 | Selecting Function | Show hand sign different “select” hand sign inside “ Học ” area | Nothing change | Test case LHSTC01is executed |
| LHSTC04 | Selecting Function | Show hand sign different “select” hand sign outside “ Học ” area | Nothing change | Test case LHSTC01is executed |
| LHSTC05 | Selecting Function -> Learning Hand Sign Language | N/A | Show the list of words on the interface in the group box “Hướng Dẫn” | Test case LHSTC04 is executed |
| The analyzed images show on the interface continuously |
| The “A” word is selected first |
| Image describing “A” hand sign is displayed |
| A notify “Hãy đưa kí hiệu trong hướng dẫn vào vùng mũi tên lên xuống để thay đổi từ được chọn ” is shown in the groupbox “Thông Báo” |
| Two white “Lên” and “Xuống” area were drawn on these images showing on the interface. |
| LHSTC06 | Learning Hand Sign Language | Move the “select” hand sign into the “Lên” square area. | The “A” word in the list is still selected | Test case LHSTC05 is executed |
| Image describing “A” hand sign is still displayed |
| Group box “Kết Quả Hiện Tại” with no result content below |
| LHSTC07 | Learning Hand Sign Language | Move the “select” hand sign into the “Xuống” square area. | The “B” word in the list is selected | Test case LHSTC05 or LHSTC06 is executed |
| Image describing “B” hand sign is displayed |
| Group box “Kết Quả Hiện Tại” with no result content below |
| LHSTC08 | Learning Hand Sign Language | Move the “select” hand sign outside the “Lên” and “Xuống” square area. | The “B” word in the list is still selected | Test case LHSTC07 is executed |
| Image describing “B” hand sign is still displayed |
| Group box “Kết Quả Hiện Tại” with result “N” content below |
| LHSTC09 | Learning Hand Sign Language | Show “E” hand sign outside the “Lên” and “Xuống” square area. | The “B” word in the list is still selected | Test case LHSTC07 is executed |
| Image describing “B” hand sign is still displayed |
| Group box “Kết Quả Hiện Tại” with result “E” content below |
| LHSTC10 | Learning Hand Sign Language | Show “E” hand sign inside the “Lên” square area. | The “B” word in the list is still selected | Test case LHSTC07 is executed |
| Image describing “B” hand sign is still displayed |
| Group box “Kết Quả Hiện Tại” with result “E” content below |
| LHSTC11 | Learning Hand Sign Language | Show “E” hand sign inside the “Xuống” square area. | The images containing only the hand on the The “B” word in the list is still selected | Test case LHSTC07 is executed |
| Image describing “B” hand sign is still displayed |
| Group box “Kết Quả Hiện Tại” with result “E” content below |
| LHSTC12 | Learning Hand Sign Language | Move the “select” hand sign into the “Lên” square area. | The “A” word in the list is selected | Test case LHSTC07 is executed |
| Image describing “A” hand sign is displayed |
| Group box “Kết Quả Hiện Tại” is empty |
| LHSTC13 | Learning Hand Sign Language | Showing the “end” hand sign outside of the two square areas “Lên ” and “Xuống” | Selecting Function interface is displayed | Test case LHSTC05 is executed |
| LHSTC14 | Learning Hand Sign Language -> Selecting Function | Showing the “end” hand sign inside of the two square areas “Lên” | Selecting Function interface is displayed | Test case LHSTC05 is executed |
| LHSTC15 | Learning Hand Sign Language -> Selecting Function | Showing the “end” hand sign inside of the two square areas “Xuống” | Selecting Function interface is displayed | Test case LHSTC05 is executed |
| LHSTC16 | Learning Hand Sign Language | Don’t show hand inside the camera area | Shows a message “Không tìm thấy bàn tay!” in the group box “Kết Quả Hiện Tại” |  |

#### Integration Test Procedure TP4

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **ID** | **Purpose** | **Procedure Steps** | **Executed By** | **Result** | **Test Date** | **Note** |
| LHSTP01 | Test fail “Selection Function”: “select” hand sign outside “Học” area | 1. Execute test case LHSTC01  2. Execute test case LHSTC02 |  | **Pass** |  |  |
| LHSTP02 | Test fail “Selection Function”: hand sign different “select” hand sign inside “Nhận dạng” area | 1. Execute test case LHSTC01  2. Execute test case LHSTC03 |  | **Pass** |  |  |
| LHSTP03 | Test fail “Selection Function”: different “select” hand sign outside “Nhận dạng” area | 1. Execute test case LHSTC01  2. Execute test case LHSTC04 |  | **Pass** |  |  |
| LHSTP04 | Test success Learning Hand Sign Language: select “B” word below the “A” word | 1. Execute test case LHSTC01  2. Execute test case LHSTC05  3. Execute test case LHSTC07 |  | **Pass** |  |  |
| LHSTP05 | Test success Learning Hand Sign Language: select “A” word above the “B” word | 1. Execute test case LHSTC01  2. Execute test case LHSTC05  3. Execute test case LHSTC07  4. Execute test case LHSTC12 |  | **Pass** |  |  |
| LHSTP06 | Test success Learning Hand Sign Language: select “Lên” when  “A” word is selected in the top | 1. Execute test case LHSTC01  2. Execute test case LHSTC05  3. Execute test case LHSTC06 |  | **Pass** |  |  |
| LHSTP07 | Test success Learning Hand Sign Language: checking “E” hand sign with result “E” | 1. Execute test case LHSTC01  2. Execute test case LHSTC05  3. Execute test case LHSTC07  4. Execute test case LHSTC09 |  | **Pass** |  |  |
| LHSTP08 | Test success Learning Hand Sign Language: checking “E” hand sign with result “E” | 1. Execute test case LHSTC01  2. Execute test case LHSTC05  3. Execute test case LHSTC07  4. Execute test case LHSTC10 |  | **Pass** |  |  |
| LHSTP10 | Test success Learning Hand Sign Language: checking “E” hand sign with result “E” | 1. Execute test case LHSTC01  2. Execute test case LHSTC05  3. Execute test case LHSTC07  4. Execute test case LHSTC11 |  | **Pass** |  |  |
| LHSTP11 | Test success finish Learning Hand Sign Language | 1. Execute test case LHSTC01  2. Execute test case LHSTC05  3. Execute test case LHSTC13 |  | **Pass** |  |  |
| LHSTP12 | Test success finish Learning Hand Sign Language | 1. Execute test case LHSTC01  2. Execute test case LHSTC05  3. Execute test case LHSTC07  4. Execute test case LHSTC15 |  | **Pass** |  |  |
| LHSTP13 | Test success finish Learning Hand Sign Language | 1. Execute test case LHSTC01  2. Execute test case LHSTC05  3. Execute test case LHSTC07  4. Execute test case LHSTC14 |  | **Pass** |  |  |
| LHSTP14 | Test fail Learning Hand Sign Language | 1. Execute test case LHSTC01  2. Execute test case LHSTC05  3. Execute test case LHSTC16 |  | **Pass** |  |  |

### “Monitor Battery Capacity” Test



Figure 93: Components of the Monitoring Battery Capacity

#### Integration test case

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **ID** | **Test Item(s)** | **Input Specification** | **Output Specification** | **Condition** |
| MBCTC01 | Battery -> Battery Capacity Display Circuit -> LEDs showing battery | Switch on | LED on circuit is bright. | The battery voltage > 9.8V |
| MBCTC02 | Battery -> Battery Capacity Display Circuit -> LEDs showing battery | The battery voltage is higher 12V | 4 LEDs on circuit is bright. | Test case MBCTC01 is exectuted |
| MBCTC03 | Battery -> Battery Capacity Display Circuit -> LEDs showing battery | The battery voltage is between 11.3V and 12V | 3 LEDs on circuit is bright. | Test case MBCTC01 is exectuted |
| MBCTC04 | Battery -> Battery Capacity Display Circuit -> LEDs showing battery | The battery voltage is between 11.3V and 10.8V | 2 LEDs on circuit is bright. | Test case MBCTC01 is exectuted |
| MBCTC05 | Battery -> Battery Capacity Display Circuit -> LEDs showing battery | The battery voltage is between 10.8V and 9.9V | 1 LEDs on circuit is bright. | Test case MBCTC01 is exectuted |
| MBCTC06 | Battery -> Battery Capacity Display Circuit -> LEDs showing battery | The battery voltage is lower 9.9V | 4 LEDs on circuit is off. | Test case MBCTC01 is exectuted |
| MBCTC07 | Battery -> Battery Capacity Display Circuit -> Raspbbery PI 2 | N/A | LED on Raspberry is bright. | Test case MBCTC01 is executed |
| MBCTC08 | Raspbbery PI 2 -> Low Battery Notify | The battery voltage is between 9.8V and 10.5V | Low Battery Notify “Bin yếu vui lòng tắt hệ thống và cắm sạc. Thông báo sẽ được tự động tắt.” is shown on the system interfaces. | Test case MBCTC02 is executed |
| Countdown time is shown in Low Battery Notify by seconds from 3. |
| MBCTC09 | Raspbbery PI 2 -> Low Battery Notify | The battery voltage is between ..V and ..V | Low Battery Notify is not displayed. |  |
| MBCTC10 | Raspbbery PI 2 -> Low Battery Notify | Waiting for the countdown time counts to 0. | Low Battery Notify is hide. | Test case MBCTC03 is executed |

#### Integration test procedure

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **ID** | **Purpose** | **Procedure Steps** | **Executed By** | **Result** | **Test Date** | **Note** |
| MBCTP01 | Test the flow Monitoring Battery Capacity works correctly with low battery. | 1. Execute test case MBCTC01  2. Execute test case MBCTC07  3. Execute test case MBCTC05  4. Execute test case MBCTC08  5. Execute test case MBCTC10 |  | **Pass** |  |  |
| MBCTP02 | Test the flow Monitoring Battery Capacity works correctly with normal battery. | 1. Execute test case MBCTC01  2. Execute test case MBCTC07  3. Execute test case MBCTC04  4. Execute test case MBCTC09  5. Execute test case MBCTC10 |  | **Pass** |  |  |
| MBCTP01 | Test the displaying battery capacity on LEDs. | 1. Execute test case MBCTC01  2. Execute test case MBCTC02 |  | **Pass** |  |  |
| MBCTP01 | Test the displaying battery capacity on LEDS. | 1. Execute test case MBCTC01  2. Execute test case MBCTC03 |  | **Pass** |  |  |
| MBCTP01 | Test the displaying battery capacity on LEDs. | 1. Execute test case MBCTC01  2. Execute test case MBCTC04 |  | **Pass** |  |  |

### 4.5 “Charging Battery” Test



Figure 94: Components of the charging battery

#### 4.5.1 Integration test case

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **ID** | **Test Item(s)** | **Input Specification** | **Output Specification** | **Condition** |
| CBTC01 | 220V power source -> LIPO B3AC charger -> Led Indicator | Connect LIPO B3AC charger to power source | The charger’s LEDs indicator is bright with red color. | The system is OFF. |
| CBTC02 | LIPO B3AC charger -> LIPO Battery | Connect LIPO B3AC charger to LIPO Battery has the voltage lower 9.8V. | The charger’s LEDs indicator is bright with red color. | Test case CBTC01 is executed. |
| CBTC03 | LIPO B3AC charger -> LIPO Battery | Waiting for about ?? minutes. | One Charger’s LED is bright with green color. | Test case CBTC02 is executed. |
| CBTC04 | LIPO B3AC charger -> LIPO Battery | Waiting for about ?? minutes. | Two Charger’s LEDs are bright with green color. | Test case CBTC02 is executed. |
| CBTC05 | LIPO B3AC charger -> LIPO Battery | Waiting for about ?? minutes. | Three Charger’s LEDs are bright with green color. | Test case CBTC02 is executed. |

#### Integration test procedure

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **ID** | **Purpose** | **Procedure Steps** | **Excuted By** | **Result** | **Test Date** | **Note** |
| CBIPT01 | Test the charging battery flow can succeed and work stability. | 1. Execute test case CBTC01  2. Execute test case CBTC02  3. Execute test case CBTC03 |  | Pass |  |  |
| CBIPT02 | Test the charging battery flow can succeed and work stability. | 1. Execute test case CBTC01  2. Execute test case CBTC02  3. Execute test case CBTC04 |  | Pass |  |  |
| CBIPT03 | Test the charging battery flow can succeed and work stability. | 1. Execute test case CBTC01  2. Execute test case CBTC02  3. Execute test case CBTC05 |  | Pass |  |  |