**MINISTRY OF EDUCATION AND TRAINING **

**FPT UNIVERSITY**

Capstone Project Document

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Automatic Alternative Image Recognition to Voice**

|  |  |
| --- | --- |
| **Group 04** | |
| **Group members** | Phan Trung Thành – Team Leader – SE61288  Nguyễn Vũ Hoàng Sơn – Team Member – SE61490  Võ Hà Quân – Team Member - SE61254  Nguyễn Cao Duy – Team Member - SE61032 |
| **Supervisor** | Mr. Kiều Trọng Khánh |
| **Ext. Supervisor** | N/A |
| **Capstone Project code** | AAIV |

-Ho Chi Minh City, ***January 05 2016***-

*This page is intentionally left blank*

**Definitions, Acronyms, and Abbreviations**

|  |  |
| --- | --- |
| **Name** | **Definition** |
| AAIV | Automatic Alternative Image Recognition to Voice |
| DEV | Developer |
| MCS | Microsoft Cognitive Service |
|  |  |

**Table 1: Definitions, Acronyms, and Abbreviations**

# Introduction

## Project Information

* **Project Name:** Automatic Alternative Image Recognition to Voice.
* **Project Code:** AAIV
* **Production Type:** Mobile Application
* **Start Date:** 05/01/2017
* **End Date:** 06/04/2017

## Introduction

* In this document, we introduce a solution for blind people to recognize faces and things that happening around them. Nowadays smartphone is very popular, almost everybody can afford one with a reasonable price. They support voice control and shortcut keys for customers. Because of those advances and conveniences, we decide to create a mobile application that can help the blind people or people with a visual impairment to recognize the person or things that stand in front of them and notify them through voices.
* By using Microsoft’s powerful **Cognitive Services**, our customer can take pictures of a person in front of them, and the application will recognize the face(s) base on trained data, which can be imported by the blind user’s family, friends, or caretaker. Furthermore, the application can also detect objects in the picture taken by the user, and then describes it back to them through voice notification.
* Because of the specific purpose and type of customer (blind person), our application supports voice control- which is supported by most of nowadays smartphones – as well as voice notifications.

## Current Situation

Through research and investigation, we found some currently available applications that serve similar functions. After using and testing, we divided them into 2 main categories:

- **Applications that help identifying objects (LookTel, KNFB Reader App, TapTapSee…):** these apps help the user to navigate and identify specific objects or describe scenes through taken picture. Things like money, text, color….

- **Applications that help blind people through sighted volunteers (Be My Eye):** these apps help the blind people by using the help of sighted volunteers. The blind person will request assistance in the app (the challenge can be anything from knowing the expiry date on the milk to navigating new surroundings…). Then the volunteer helper receives a notification for help and a live video connection is established. From the live video, the volunteer can help the blind person by answering the question they need to be answered

## Problem Definition

From that information, we immediately notice some problems:

* **For the identifying objects apps:**
  + Although these apps serve the needed function, that is to help the blind person identify things around them, but they haven’t been able to recognize people face.
  + Some apps do not support voice control.
  + Low performance, take too long (over 20 seconds, with an average bandwidth of 17.3 Mbps) to return the result.
  + Most of these apps are hard to use and not user-friendly.
* **For the volunteer-based apps**:
  + This concept is fairly new and interesting but it doesn’t solve the problem automatically; it always needs the real volunteer to solve the problem and return the result to user.
  + A stable Internet connection is always needed.
  + Security problems.

## Proposed Solution

### Feature functions

* Remote simulator camera:
* Sending images to the system to add new persons to acquaintance list.
* Sending images to the system to detect acquaintances, their features and emotions.
* Sending images to the system to analyze landscape, objects.
* For the blind people:
* Taking photos via voice/shortcut key.
* Getting information about the person in the taken photo.
* Getting information about the landscape or the objects in the taken photo.
* For the trainers:
* Allow training the system through taken images by adding information about new persons such as name, relationship.
* Getting information about photos which the blinds took, where they went, who they met, what they saw.

### Advantages and disadvantages

* Advantages:
* Detect the acquaintances, landscape, objects without helping of other people.
* Storing data as a diary, easy to track what the blinds met anytime.
* The blinds know the people talking to and their attitude that help them communicate effectively.
* Disadvantages:
* Sometimes the detection does not work accurately.

## Functional Requirements

Functional requirements of the system are listed as below:

* Train component:
  + System training: trainer will create blind’s friend information from taken photos.
* Blind person component:
  + Capture a picture: the blind will use the application to capture a picture of people who are talking face to face to them or the landscape/objects in front of them.
  + Save picture: the system can save the picture automatic which system have not been able to recognize as data for later training.
  + Voice control: the blind will use their voice to control the application.
* Camera component:
  + Take picture: the camera will take pictures of people who face to face to the blind or the landscape/objects in front of the blinds.
* System component:
  + Recognize image: The system will recognize the image of blind when they capture a picture and say the name of known people. If the system cannot recognize the person, the taken picture would be saved into database.
  + Store data: The system will store blind’s friend information to the database. Add it to Microsoft Cognitive Service to identify.
  + Voice notification: the system will return blind’s friend information as voice notification when it recognized the person.

## Role and Responsibility

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No** | **Full Name** | **Role** | **Position** | **Contact** |
| 1 | Kiều Trọng Khánh | Project Manager | Supervisor | khanhkt@fpt.edu.vn |
| 2 | Phan Trung Thành | Developer | Leader | thanhptse61288@fpt.edu.vn |
| 3 | Nguyễn Vũ Hoàng Sơn | Developer | Member | sonnvhse61490@fpt.edu.vn |
| 4 | Võ Hà Quân | Developer | Member | quanvhse61254@fpt.edu.vn |
| 5 | Nguyễn Cao Duy | Developer | Member | duyncse61032@fpt.edu.vn |

# Software Project Management Plan

## Problem Definition

### Name of this Capstone Project

* **Official name:** Automatic Alternative Image Recognition to Voice.
* **Vietnamese name:** Nhận dạng hình ảnh hỗ trợ người mù giao tiếp.
* **Abbreviation:** AAIV.

### Problem Abstract

* With the current situation, if the blinds want to get information about the people, the landscape of the objects in front of them, they have to get help from the other people. It’s hard for them to travel alone or communicate effectively. There are some applications that support them to detect things but the effectiveness is not high. So we build an application based on the Image Recognition Technology that helps the blinds detect the acquaintances, the landscape or things in front of them by the photos they took. We have to study new technologies then apply to get the high-accuracy result such as detecting the acquaintances, classifying images into some categories: human features and emotions, landscape, objects…Then we use to speech technology to send the picture information to the blinds via voice. All the captured images are stored in the log file so that the blind’s relatives can use for tracking or training.
* We also provide the function that the blind’s relatives can train images. It means they will provide information for the images in acquaintance list so that it’s easier to detect the people the blind meet. This function can be used via phone application or web application.
* All of these function are used online, but the Internet connection is not always stable so we have to research another algorithm for offline detecting.

### Project Overview

#### Current Situation

* Below are the problems encountered in this project:
* **First time using Microsoft Cognitive Services:** this is the first time our team uses the Cognitive Services. Therefore times and efforts are required to investigating and developing.
* **Lack of Mobile development skills:** our team members are quite new to Android development procedure (UI, UX, coding..).
* **Time-consuming to implement Camera and Voice control function:** the application requires a picture taken by user in order to identify faces/objects. Also, the application must support voice control system to fit the user needs.
* **User Interface and User Experience design:** because of the special function and type of customer (blind people), our team has to design our application in a way that blind users can use it conveniently and effectively.
* **Still required an Internet connection:** the Cognitive Service requires an Internet connection in order to request APIs and return results. Our team must find a walk around for this issue.

#### The proposed system

* After research, we concluded that Microsoft’s Cognitive Services matched the requirements for this applications. It provides lots of services for identifying human faces and object.
* Using Cognitive Service, we can identify a person by the taken picture of them. Moreover, the system can also learn new faces input by the user/ user’s partner to enhance performance in the future.
* The results returned by the Cognitive Service will be handled and filter by the system. Then it will be shown to the user through voice notifications.
* User can use voice to control the application.
* We divided our system into:
  + **The back-end system:**
* The back-end system will be implemented in C# .Net framework.
* Receive picture taken by the mobile application, using the Face Recognition / Computer Vision API, to identify person/object that the user needs.
* Storing pictures and training new faces as requested by user in a database.
  + **The mobile application:**
* Used by user/user’s partner to take pictures for the identification process.
* Receives result returned from server and notice user through voice.
* Create and learn new faces through the training process.
  + **The website:**
* Created for the user’s partners, family…to manage the application resources (people faces, pictures).

#### Boundaries of the system

* The system can:
  + Allow user to train the system to learn new faces.
  + Allow the blind to capture a picture by the camera.
  + Allow saving log files if it could not recognize, for later training.
  + Allow voice control.
  + The system can be used online or offline.
  + Get blind’s friend’s attendance.

#### Future plans

* Improve voice control: support new features: save information by voice, search information by voice.
* Describe object should be recognized.
* Repeat last picture identification.
* Support multipla platforms: iOS, Window phone or smart glasses.

#### Development Environment

**Hardware requirement**

+ For server

|  |  |  |
| --- | --- | --- |
| Windows | Minimum Requirements | Recommended |
| Internet Connection | Cable(4 Mbps) | Cable(8 Mbps) |
| Operating System | Window Server 2008 | Window Server 2012 |
| Computer Processor | Intel® Core i3 1.4GHz | Intel® Core i5 2.50 GHz |

**Table …: Hardware requirement for Server**

+ For mobile

|  |  |  |
| --- | --- | --- |
| Windows | Minimum | Recommended |
| Internet Connection | Wi-Fi or 3G (4 Mbps) | Wi-Fi or 3G (14 Mbps) |
| Operating System | Android 6.0.1 | Android 6+ |
| Device | Support camera 5 megapixel  Internet connection (4 Mbps) | Support camera 5+ megapixel  Internet connection (14 Mbps) |

**Table …: Hardware requirement for mobile**

+ For web

|  |  |  |
| --- | --- | --- |
| Windows | Minimum | Recommended |
| Internet Connection | Cable or Wi-Fi or 3G (4 Mbps) | Cable or Wi-Fi or 3G (14 Mbps) |
| Operating System | Window 7 | Window 10 |
| Computer Processor | Intel® Core i3 1.4GHz | Intel® Core i5 2.50 GHz |

**Table …: Hardware requirement for web**

**Software requirement**

|  |  |  |
| --- | --- | --- |
| Software | Name / Version | Description |
| Operating system | Window 7, Window 10 | Operating system and platform for development |
| Environment | .NET | Specification for developing web application |
| IDE |  | Used for implement website and Android Mobile App. |
| Design Model tool | StartUML v5.0 | Used for creating modal and diagrams. |
| DBMS | SQL Server 2008 | Used to create & manage the database for system |
| Document storage | Slack | Used for storing document |
| Store and manage source code | GitHub, SourceTree | Used to store all source code |

## Project organization

### Software Process Model

Our project using Scum model to develops, which is an iterative and incremental agile software development framework. Because four reason:

* We use Microsoft Cognitive Service and Clarify which is new for us, need time to research and practice.
* Project can respond easily to change.
* Reduced risk (spending large amounts of time with no return on investment).
* Shorten the time-release software



**Figure xx: Scrum Development Model**

(Software Engineering 9 th , Sommerville - Fig. 3.7)

For more information: <https://en.wikipedia.org/wiki/Scrum_(software_development)>

### Roles and responsibilities

| No | Full name | Role in Group | Responsibilities |
| --- | --- | --- | --- |
| 1 | Kiều Trọng Khánh | Supervisor | * Specify user requirement * Control the development process * Give out technique and business analysis support |
| 2 | Phan Trung Thành | Team leader, DEV, Tester | * Managing process * Designing database * Clarifying requirements * Prepare documents * Create test plan * Coding * Testing |
| 3 | Nguyễn Vũ Hoàng Sơn | Team member, DEV,  Tester | * Designing database * Clarifying requirements * Prepare documents * GUI design * Create test plan * Coding * Test |
| 4 | Võ Hà Quân | Team member, DEV,  Tester | * Designing database * Clarifying requirements * Prepare documents * Create test plan * Coding * Test |
| 5 | Nguyễn Cao Duy | Team meamber, DEV, Tester | * Designing database * Clarifying requirements * Prepare documents * Create test plan * Coding * Test |

**Table xx: Roles and Responsibilities Details**

### Tools and Techniques

|  | Tools | Techniques |
| --- | --- | --- |
| **Front-end** | Visual Studio | * HTML5 * CSS3 * JavaScript * jQuery * Ajax |
| **Back-end** | Visual Studio, Android Studio | * .Net framework   + ASP.NET MVC4  + Entity Framework 5   * OpenCV library |
| **Web server** | IIS version 1607 | * C# .NET |
| **Mobile application** | Android Studio | * Version 2.2.3 * Java 8 * SDK 23 |
| **Database management system** | MS SQL Server 2008 | N/A |

## Project Management Plan

### Product Backlog

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Story ID** | **Features** | **Task ID** | **Task description** | **Sprint** |
| 1 | Create Project Back Log | 1.1 | Create project back log | 1 |
| 2 | Create Introduction Documents | 2.1 | Create Introduction Documents | 1 |
| 3 | Write Project Management Plan | 3.1 | Problem definition | 1 |
| 3.2 | Project organization | 1 |
| 3.3 | Project management plan | 1 |
| 3.4 | Coding convention | 1 |
| 3.5 | Review document | 1 |
| 4 | Create System Structure | 4.1 | Create API for online mode | 1 |
| 4.2 | Create prototype Mobile application | 1 |
| 4.3 | Create Database Structure | 1 |
| 5 | Studying Microsoft Cognitive Services | 5.1 | How it works | 1 |
| 5.2 | Face Recognition Api | 1 |
| 5.3 | Computer Vision Api | 1 |
| 6 | Studying Clarify | 6.1 | Using common model | 1 |
| 6.2 | How to do prediction | 1 |
| 6.3 | How to train model | 1 |
| 6.4 | Dealing with images and inputs | 1 |
| 7 | Write Software Requirements | 7.1 | User Requirement Specification | 2 |
| 7.2 | External Interface Requirement | 2 |
| 7.3 | Use case diagram | 2 |
| 7.4 | Software System Attributes | 2 |
| 8 | Write Software Design Description | 8.1 | Design Overview | 3 |
| 8.2 | System Architectural Design | 3 |
| 8.3 | Component Diagram | 3 |
| 8.4 | Detailed Description of Components | 3 |
| 8.5 | Sequence Diagram | 3 |
| 8.6 | User Interface Diagram | 3 |
| 8.7 | Database Design | 3 |
| 9 | Implementation | 9.1 | Mobile\_Camera Surface View | 1 |
| 9.2 | Mobile\_Online Mode Vision Recognition function | 1 |
| 9.3 | Mobile\_Online Mode Face Recognition function | 2 |
| 9.4 | Mobile\_Voice Notification of returned result | 2 |
| 9.5 | Mobile\_Voice Control | 2 |
| 9.6 | Mobile\_Write Log File | 3 |
| 9.7 | Mobile\_Online mode Face recognition quick check | 4 |
| 9.8 | Mobile\_Offline mode Face Recognition | 4 |
| 9.9 | Mobile\_Record of unrecognized faces | 5 |
| 9.10 | Web\_User Login | 2 |
| 9.11 | Web\_User Logout | 2 |
| 9.12 | Web\_User Search + View data | 3 |
| 9.13 | Web\_User Trains Data by adding images | 4 |
| 9.14 | System\_Create Vision Api | 1 |
| 9.15 | System\_Create Face Api | 1 |
| 9.16 | System\_Implement offline mode recognition | 3 |
| 9.17 | System\_Implement Vision Api with Clarify | 4 |
| 10 | Create Software Test Documentation | 10.1 | Test Plan | 5 |
| 10.2 | Test Cases | 5 |
| 10.3 | Check lists | 5 |
| 11 | Quality Assurance | 11.1 | Quality Assurance for Backend | 5 |
| 11.2 | Quality Assurance for Web | 5 |
| 11.3 | Quality Assurance for Mobile | 5 |
| 12 | Software User’s Manual | 12.1 | Installation Guide | 6 |
| 12.2 | User’s Guide | 6 |

*Table #: Product Backlog*

### Sprint Backlog

#### Sprint 1 (5/1/2017 – 19/1/2017) Project initiation

**Goals**: Sprint 1 must complete the following tasks:

* Create Project Back Log
* Create Introduction Documents
* Write Project Management Plan

+ Problem definition

+ Project organization

+ Project management plan

+ Coding convention

+ Review document

* Create System Structure

+ Create API for online mode

+ Create prototype Mobile application

+ Create Database Structure

* Studying Microsoft Cognitive Services

+ How it works

+ Face Recognition Api

+ Computer Vision Api

* Studying Clarify

+ Using common model

+ How to do prediction

+ How to train model

+ Dealing with images and inputs

* Implementation

+ Mobile\_Camera Surface View

+ Mobile\_Online Mode Vision Recognition function

+ System\_Create Vision Api

+ System\_Create Face Api

**Development:**

|  |  |  |  |
| --- | --- | --- | --- |
| **No** | **Task** | **Implement** | **Reviewer** |
| 1 | Create project back log | SonNVH | ThanhPT |
| 2 | Create Introduction Documents | SonNVH, ThanhPT, QuanVH, DuyNC | ThanhPT |
| 3 | Problem definition | QuanVH | ThanhPT |
| 4 | Project organization | ThanhPT | N/a |
| 5 | Project management plan | SonNVH | ThanhPT |
| 6 | Coding convention | ThanhPT | SonNVH,  DuyCT,  QuanVH |
| 7 | Review document | ThanhPT | N/a |
| 8 | Create API for online mode | ThanhPT | SonNVH |
| 9 | Create prototype Mobile application | SonNVH | ThanhPT |
| 10 | Create Database Structure | QuanVH, DuyNC | ThanhPT |
| 11 | How it works | SonNVH, ThanhPT | ThanhPT |
| 12 | Studying Face Recognition Api | SonNVH, ThanhPT | N/a |
| 13 | Studying Computer Vision Api | SonNVH, ThanhPT | N/a |
| 14 | Using common model | DuyNC | N/a |
| 15 | How to do prediction | DuyNC | N/a |
| 16 | How to train model | DuyNC | N/a |
| 17 | Dealing with images and inputs | DuyNC | N/a |
| 18 | Mobile\_Camera Surface View | SonNVH | ThanhPT |
| 19 | Mobile\_Online Mode Vision Recognition function | SonNVH | ThanhPT |
| 20 | System\_Create Vision Api | ThanhPT | SonNVH. QuanVH, DuyNC |
| 21 | System\_Create Face Api | ThanhPT | SonNVH. QuanVH, DuyNC |

*Table #: Sprint 1 Development*

#### Sprint 2 (20/1/2017 – 21/1/2017; 31/1/2017 – 14/2/2017) Software Requirement and Core Features

**Goals:** Sprint 2 must complete the following tasks.

* Write Software Requirements

+ User Requirement Specification

+ External Interface Requirement

+ Use case diagram

+ Software System Attributes

* Implementation

+ Mobile\_Online Mode Face Recognition function

+ Mobile\_Voice Notification of returned result

+ Mobile\_Voice Control

+ Web\_User Login

+ Web\_User Logout

**Development:**

|  |  |  |  |
| --- | --- | --- | --- |
| **No** | **Task** | **Implement** | **Reviewer** |
| 1 | User Requirement Specification | QuanVH | ThanhPT |
| 2 | External Interface Requirement | DuyNC | ThanhPT |
| 3 | Use case diagram |  |  |
| 4 | Software System Attributes | DuyNC | ThanhPT |
| 5 | Mobile\_Online Mode Face Recognition function | SonNVH | ThanhPT, QuanVH, DuyNC |
| 6 | Mobile\_Voice Notification of returned result | SonNVH | ThanhPT, QuanVH, DuyNC |
| 7 | Mobile\_Voice Control | SonNVH | ThanhPT, QuanVH, DuyNC |
| 8 | Web\_User Login | QuanVH | ThanhPT, SonNVH, DuyNC |
| 9 | Web\_User Logout | QuanVH | ThanhPT, SonNVH, DuyNC |

*Table #: Sprint 2 Development*

#### Sprint 3 (15/2/2017 – 1/3/2017) Software Design Description and Main Features

**Goals**: Sprint 3 must complete the following tasks:

* Write Software Design Description

+ Design Overview

+ System Architectural Design

+ Component Diagram

+ Detailed Description of Components

+ Sequence Diagram

+ User Interface Diagram

+ Database Design

* Implementation

+ Mobile\_Write Log File

+ Web\_User Search + View data

+ System\_Implement offline mode recognition

**Development:**

|  |  |  |  |
| --- | --- | --- | --- |
| **No** | **Task** | **Implement** | **Reviewer** |
| 1 | Design Overview |  |  |
| 2 | System Architectural Design |  |  |
| 3 | Component Diagram |  |  |
| 4 | Detailed Description of Components |  |  |
| 5 | Sequence Diagram |  |  |
| 6 | User Interface Diagram |  |  |
| 7 | Database Design |  |  |
| 8 | Mobile\_Write Log File | SonNVH | ThanhPT, QuanVH, DuyNC |
| 9 | Web\_User Search + View data | QuanVH | ThanhPT, SonNVH, DuyNC |
| 10 | System\_Implement offline mode recognition | ThanhPT | SonNVH QuanVH, DuyNC |

#### Sprint 4 (2/3/2017 – 16/3/2017) Finish Coding

**Goals:** Sprint 4 must complete the following tasks:

* Implementation

+ Mobile\_Online mode Face recognition quick check

+ Mobile\_Offline mode Face Recognition

+ Web\_User Trains Data by adding images

+ System\_Implement Vision Api with Clarify

**Development:**

|  |  |  |  |
| --- | --- | --- | --- |
| **No** | **Task** | **Implement** | **Reviewer** |
| 1 | Mobile\_Online mode Face recognition quick check | SonNVH | ThanhPT, QuanVH, DuyNC |
| 2 | Mobile\_Offline mode Face Recognition | SonNVH | ThanhPT, QuanVH, DuyNC |
| 3 | Web\_User Trains Data by adding images | QuanVH | SonNVH, QuanVH, DuyNC |
| 4 | System\_Implement Vision Api with Clarify | ThanhPT | SonNVH, QuanVH, DuyNC |

*Table #: Sprint 4 Development*

#### Sprint 5 (17/3/2017 – 31/3/2017) Testing Document and Quality Assurance

**Goals:** Sprint 5 must complete the following tasks:

* Implementation

+ Mobile\_Record of unrecognized faces

* Create Software Test Documentation

+ Test Plan

+ Test Cases

+ Check lists

* Quality Assurance

+ Quality Assurance for Backend

+ Quality Assurance for Web

+ Quality Assurance for Mobile

**Development:**

|  |  |  |  |
| --- | --- | --- | --- |
| **No** | **Task** | **Implement** | **Reviewer** |
| 1 | Mobile\_Record of unrecognized faces | SonNVH | ThanhPT, QuanVH, DuyNC |
| 2 | Test Plan |  |  |
| 3 | Test Cases |  |  |
| 4 | Check lists |  |  |
| 5 | Quality Assurance for Backend |  |  |
| 6 | Quality Assurance for Web |  |  |
| 7 | Quality Assurance for Mobile |  |  |

*Table #: Sprint 5 Development*

#### Sprint 6 (31/3/2017 – 6/4/2017) Software User’s Manual

**Goals:** Sprint 6 must complete the following tasks

* Software User’s Manual

+ Installation Guide

+ User’s Guide

**Development:**

|  |  |  |  |
| --- | --- | --- | --- |
| **No** | **Task** | **Implement** | **Reviewer** |
| 1 | Installation Guide |  |  |
| 2 | User’s Guide |  |  |

### Deliverables

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No** | **Deliverable** | **Deliverable date** | **Deliverable location** | **Note** |
| 1 | Introduction Document, Task list | 10/01/2017 | FU - LMS | Report No. 1 |
| 2 | Software Project Management Plan | 12/01/2017 | FU – LMS | Report No. 2 |
| 3 | Software Requirements Specification | 20/1/2017 | FU – LMS | Report No. 3 |
| 4 | Software Design Description | 16/2/2017 | FU – LMS | Report No. 4 |
| 5 | Software Test Documentation Guide Implementation (Coding) | 11/03/2017 | FU – LMS | Report No. 5 |
| 6 | Software User’s Manual | 30/3/2017 | FU - LMS | Report No. 6 |

### All Meeting Minutes

All meeting minutes can be found at:

https://github.com/capstonbaby/imagetovoice/tree/master/Meetting

## Coding Convention

Java: Using to develop mobile app.

Summary:

- Naming Convention:

+ Variable names should be short yet meaningful. The choice of a variable name should be designed to indicate to the casual observer the intent of its use.

+ Methods should be verbs, in mixed case with the first letter lowercase, with the first letter of each internal word capitalized.

- Indentation:

+ One declaration per line is recommended since it encourages commenting.

+ In absolutely no case should variables and functions be declared on the same line.

+ Do not put different types on the same line.

- Declarations Convention:

+ One declaration per line is recommended since it encourages commenting.

Using Java Code Convention from:

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

C#: using for web application

Summary:

- Naming Convention:

+ Variable names should be short yet meaningful. The choice of a variable name should be designed to indicate to the casual observer the intent of its use.

+ Methods should be verbs, in mixed case with the first letter uppercase, with the first letter of each internal word capitalized.

- Indentation:

+ One declaration per line is recommended since it encourages commenting.

+ In absolutely no case should variables and functions be declared on the same line.

+ Do not put different types on the same line.

- Declarations Convention:

+ One declaration per line is recommended since it encourages commenting.

Using C# Code Convention from:

https://msdn.microsoft.com/en-us/library/ms229042.aspx

# Software Requirement Specification

## User Requirement Specification

### Guest Requirement

Guest is a person who doesn’t have access to the system. Guest can use some functions in the system. To use these functions, guest must log in. These are some function guest can use:

• Register

• Login

### User Requirement

User is a guest who logged in the system. Guest can be the blinds or their relatives, partners. There are 2 main function-groups:

* For the blind:
  + Get information about the image:
    - Acquaintances and their emotion
    - Landscape
    - Objects
* For their relatives, partners:
  + Train new acquaintances
  + Manage acquaintances:
    - Update information
    - Delete person
  + View log

### Admin Requirement

Admin is the person who manages accounts and person group. Admin can use some following function:

* Manage accounts:
  + Add new account
  + Active / Deactivate account
  + Update account
* Manage person group:
  + Add new group
  + Update existed group’s information
  + Delete group

## System Requirement Specification

### External Interface Requirement

#### User Interface

* The user interface uses Vietnamese in android application and English language in web application.
* The user interface displays best on 1024x768-screen size

#### Hardware Interface

* Android Smartphone: Android 6.0.1 Marshmallow or later.

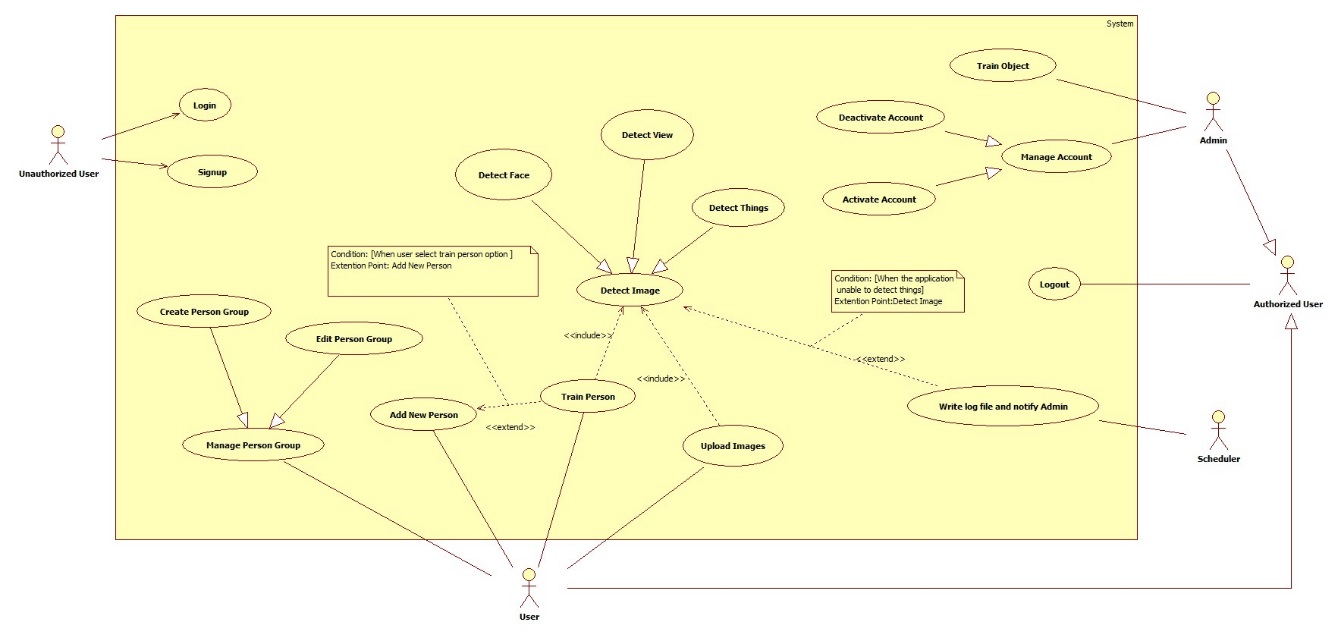
#### Software Interface

* Web application: work with Firefox (v30 or above), Chromes (v25 or above).
* Mobile Application: Android SDK Platform API 23 (or later).

#### Communication Protocol

* Use HTTP protocol 1.1 for communication between the web browser and the web server.
* Use HTTP protocol 1.1 for communication between the mobile application and the web service.
* Use HTTP protocol 1.1 for communication between the server and the Microsoft service.

### System Overview Use Case

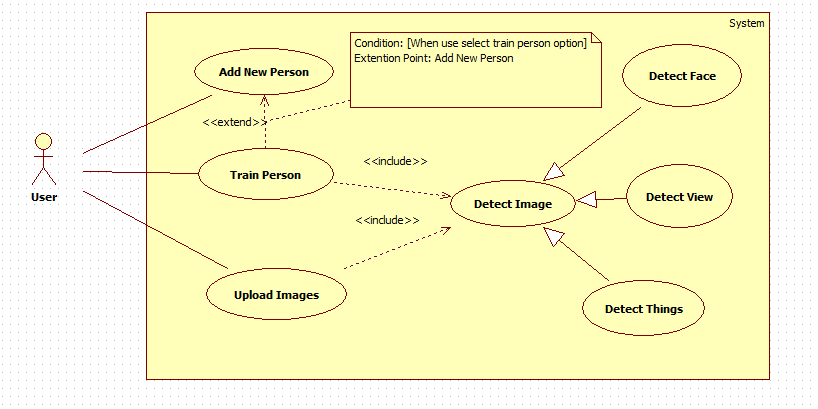


**Figure xxx: System Overview Use Case**

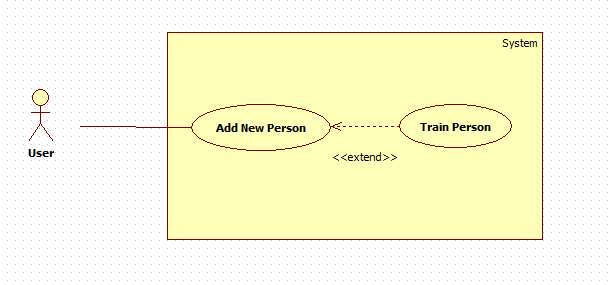
//Thieu dictionary

### List of Use Case

#### <User> Overview Use Case



##### <User> Add New Peron (UC\_AS01)



**Figure xxx: <User> Add New Person (UC\_US01)**

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE – UC\_US01** | | | |
| **Use Case No.** | UC\_US01 | **Use Case Version** | 1.0 |
| **Use Case Name** | Add New Person | | |
| **Author** | SonNVH | | |
| **Date** | 02/16/2017 | **Priority** | High |
| **Actor:**   * User   **Summary:**   * This use case allows user to add new person into a selected person group.   **Goal:**   * This function helps a user to add a new Person to a selected person group with name and descriptions.   **Triggers:**   * User sends command to add new person .   **Preconditions:**   * Actor must login at user role.   **Post Conditions:**   * **Success:** New person has been added to the system. Show success message: “Tạo người mới thành công”. * **Fail:** Show error message: “Tạo người mới thất bại”.   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | User send Create new Person command. | System requires information from user:   * personGroupId: User-provided personGroupId as a string. The valid characters include numbers, English letters in lower case, '-' and '\_'. The maximum length of the personGroupId is 64. * Name: Person display name. The maximum length is 128. * userData (optional): User-provided data attached to the person. The size limit is 16KB. * ImageUrl (optional): User-provided list of images of the new person.Valid image size is from 1KB to 4MB. Only one face is allowed per image. Maximum amount is 3 images. | | 2 | User inputs information. |  | | 3 | User sends command to create new Person.  [Alternative 1, 2] | System validates information.  [Exception 1, 2, 3, 4, 5, 6, 7] | | 4 |  | System adds new Person to selected Person Group and shows success message: “Tạo người mới thành công”.  [Exception 8, 9, 10] |   **Alternative Scenario:**  *[Alternative 1]*   |  |  |  | | --- | --- | --- | | Step. | Actor Action | System Response | | 1 | User sends command to reset. | System resets all required information fields. |   *[Alternative 2]*   |  |  |  | | --- | --- | --- | | Step. | Actor Action | System Response | | 1 | User sends command to cancel. | System closes adding new Person Group form. |   **Exceptions:**   |  |  |  | | --- | --- | --- | | No. | Cause | System Response | | 1 | personGroupId is empty. | System shows error message: “personGroupId không được bỏ trống”. | | 2 | Person group ID is invalid. Valid format should be a string composed by numbers, English letters in lower case, '-', '\_', and no longer than 64 characters. | System shows error message: “Độ dài của personGroupId tối đa 64 kí tự”. | | 3 | Person Group Id does not exist | System shows error message: “Person Group Id không tồn tại” | | 4 | Name is empty. | System shows error message: “Name không được bỏ trống”. | | 5 | Length of Name out of range 128 characters. | System shows error message: “Độ dài của Name tối đa 128 kí tự”. | | 6 | Image size under / exceeded the limit size (1Kb – 4Mb) | System shows error message: “Hình ảnh quá lớn hoặc quá nhỏ”. | | 7 | Image contains more than 1 person face. | System shows error message: “Hình ảnh chứa nhiều hơn một khuôn mặt” | | 8 | Invalid subscription Key or user/plan is blocked. | System shows error message: “Tạo người mới thất bại”. | | 9 | Out of call volume quota | System shows error message: “Lượng truy cập cho phép đã hết. Truy cập sẽ khả dụng trở lại trong ## ngày”. | | 10 | Rate limit is exceeded | System shows error message: “Đã đạt lượng truy cập tối đa. Vui lòng chờ trong vào ## giây để tiếp tục”. |   **Relationships:** N/A.  #extend trainperson???  **Business Rules:**   * System creates new person with provided Name and Descriptions and return a personId of new person. Returned personId and person information will be added to the database. * Selected images will all be compressed to reduce their size but still remain the resolution. * Compress images will be uploaded to Cloudinary Image Server and images URLs will be returned. * With each of the provided image URL, a face will be detected and system returns a faceId which will exist for 24 hours after the detection. * With each of the faceId, the system will create a persistedFaceId for the person. A person’s persistedFaceId will not expired. * The selected person group will have to be trained after successfully created a new person. | | | |

***Table #: USE CASE – UC\_AS01 - <User> Add New Person***

##### <User> Upload Images (UC\_AS02)

**Figure xxx: <User> Upload Images (UC\_US02)**

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE – UC\_US02** | | | |
| **Use Case No.** | UC\_US02 | **Use Case Version** | 1.0 |
| **Use Case Name** | Upload Images | | |
| **Author** | SonNVH | | |
| **Date** | 02/16/2017 | **Priority** | High |
| **Actor:**   * User   **Summary:**   * This use case allows user to upload an captured image from device camera to the system.   **Goal:**   * System receives the captured image(s).   **Triggers:**   * User sends command to capture image.   **Preconditions:**   * Actor must login at user role.   **Post Conditions:**   * **Success:** System receives the captured image URL. * **Fail:** No image URL is received by the system. Show error message.   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | User sends capture command. | - System receives the file path of the captured image. [Exception 1]  - Send request to Cloudinary for upload the image.[Exception 2]  - Receives return image URL.  - Continue detecting process.[Alternative 1] |   **Alternative Scenario:**  *[Alternative 1]*   |  |  |  | | --- | --- | --- | | Step. | Actor Action | System Response | | 1 |  | System receives no image URL, return error message. Waiting for next command from user. |   **Exceptions:**   |  |  |  | | --- | --- | --- | | No. | Cause | System Response | | 1 | Image Path not found | System show error message: “Chụp ảnh thất bại”. | | 2 | No Internet connection / request denied by Cloudinary | System show error message: “Upload hình ảnh không thành công”. |   **Relationships:** N/A.  **Business Rules:**   * Captured image will be compressed to reduce the size but still remain the resolution. | | | |

***Table #: USE CASE – UC\_US02 - <User> Upload Images***

##### <User> Train Person (UC\_US03)



**Figure xxx: <User> Train Person (UC\_US03)**

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE – UC\_US03** | | | |
| **Use Case No.** | UC\_US03 | **Use Case Version** | 1.0 |
| **Use Case Name** | Train Person | | |
| **Author** | SonNVH | | |
| **Date** | 02/16/2017 | **Priority** | High |
| **Actor:**   * User   **Summary:**   * This use case allows user to train a person group after create a new person or update a person face/information.(#usecase nay la hanh dong them nhieu face vo 1 person de cai thien chat luong detect)   **Goal:**   * The selected person group is trained.   **Triggers:**   * After created a new person or updated a person’s face, information.   **Preconditions:**   * Actor must login at user role. * The selected person group is not in a training process.   **Post Conditions:**   * **Success:** Show message: “Train person group thành công”. * **Fail:** Show message: “Train person group thất bại”.   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | A Train Person command is sent. | System requires information from user:  - personGroupId: Target person group to be trained. The valid characters include numbers, English letters in lower case, '-' and '\_'. The maximum length of the personGroupId is 64. | | 2 | User input required information |  | | 3 | User sends train person command | System validates input information [Exception 1, 2] | | 4 |  | System train the selected person group and show message when training process is completed: “Quá trình training đã hoàn tất” [Exception 3, 4, 5] |   **Alternative Scenario:**  *N/a*  **Exceptions:**   |  |  |  | | --- | --- | --- | | No. | Cause | System Response | | 1 | PersonGroupId is exceeded the limit, invalid | System show error message: “PersonGroupId không hợp lệ”. | | 2 | PersonGroupId is not existed | System shows error message: “Person Group ID không tồn tại”. | | 3 | Selected person group is in a training process | System shows error message: “Person group hiện đang trong quá trình training” | | 4 | Out of call volume quota | System shows error message: “Lượng truy cập cho phép đã hết. Truy cập sẽ khả dụng trở lại trong ## ngày”. | | 5 | Rate limit is exceeded | System shows error message: “Đã đạt lượng truy cập tối đa. Vui lòng chờ trong vào ## giây để tiếp tục”. |   **Relationships:** N/A.  **Business Rules:**   * Training process must be completed before continuing any action on the selected person group. * If training process is fail, a dialog must be shown to ask user for re-run the training process. | | | |

***Table #: USE CASE – UC\_US03 - <User> Train Person***

##### <User> Detect Image (UC\_US04)

#Cai usecase nay la abstract

**Figure xxx: <User> Detect Image (UC\_US04)**

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE – UC\_US04** | | | |
| **Use Case No.** | UC\_US04 | **Use Case Version** | 1.0 |
| **Use Case Name** | Detect Image | | |
| **Author** | SonNVH | | |
| **Date** | 02/16/2017 | **Priority** | High |
| **Actor:**   * User   **Summary:**   * This use case allows user to know what is inside of an image. A captured image can be described as who is in it, what is in it.   **Goal:**   * User knows what / who is in an image.   **Triggers:**   * User sends command to detect an image.   **Preconditions:**   * Actor must login at user role. * Person Groups must be trained completely.   **Post Conditions:**   * **Success:** Detected result will be returned to the user. * **Fail:** Show error message: “Nhận dạng thất bại”.   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | User sends Detect Image command with Face Detecting Mode. [Alternative 1, 2] | System requires information from user:   * ImageUrl: User-provided a image to be detected.Valid image size is from 1KB to 4MB. Each image can contains a maximum of 10 faces. | | 2 | User inputs information |  | | 3 |  | System validates inputed informations. [Exception 1] | | 4 |  | System detects the inserted image and returns the recognized person.  [Exception 2, 3, 4, 5, 6]  [Alternative 3] |   **Alternative Scenario:**  *[Alternative 1]*   |  |  |  | | --- | --- | --- | | Step. | Actor Action | System Response | | 1 | User sends Detect Image command with Object Detecting Mode . | System requires information from user:   * ImageUrl: User-provided a image to be detected.Valid image size is from 1KB to 4MB. Each image can contains a maximum of 10 faces. | |  |  | System validates inputed informations. [Exception 1] | |  |  | System detects the inserted image and returns the detected object descriptions.  [Exception 2, 3, 4, 5, 6]  [Alternative 4] |   *[Alternative 2]*   |  |  |  | | --- | --- | --- | | Step. | Actor Action | System Response | | 1 | User sends Detect Image command with View Detecting Mode . | System requires information from user:   * ImageUrl: User-provided a image to be detected.Valid image size is from 1KB to 4MB. Each image can contains a maximum of 10 faces. | |  |  | System validates inputed informations. [Exception 1] | |  |  | System detects the inserted image and returns the description of the image.  [Exception 2, 3, 4, 5, 6] |   *[Alternative 3]*   |  |  |  | | --- | --- | --- | | Step. | Actor Action | System Response | | 1 |  | System cannot detect any known person in the picture and return result: “Không nhận diện được người này”. |   *[Alternative 4]*   |  |  |  | | --- | --- | --- | | Step. | Actor Action | System Response | | 1 |  | System cannot detect the object in the image and return result message: “Không nhận diện được vật này” |   **Exceptions:**   |  |  |  | | --- | --- | --- | | No. | Cause | System Response | | 1 | Image size under / exceeded the limit size (1Kb – 4Mb) | System shows error message: “Hình ảnh quá lớn hoặc quá nhỏ”. | | 2 | Invalid subscription Key or user/plan is blocked. | System shows error message: “Nhận diện thất bại”. | | 3 | Person Group is not trained | System shows error message: “Person Group chưa được train” | | 4 | Person Group training process is not completed. | System show error message: “Person Group chưa train xong” | | 5 | Out of call volume quota | System shows error message: “Lượng truy cập cho phép đã hết. Truy cập sẽ khả dụng trở lại trong ## ngày”. | | 6 | Rate limit is exceeded | System shows error message: “Đã đạt lượng truy cập tối đa. Vui lòng chờ trong vào ## giây để tiếp tục”. |   **Relationships:** N/A.  **Business Rules:**   * User will use device’s camera to capture image. * Captured image will be compressed to smaller size but still remain its resolution. * Compressed image will be uploaded to Cloudinary Image Server, and a image direct URL will be returned. * With the returned image URL, user can choose each of followed detecting mode:   + Face Detecting Mode:     - System detects face(s) in the provided image URL, then return faceId(s) of detected faces. Maximum amount of faces for each image is 10 faces. This faceId(s) will exist for 24 hours after detection.     - With each of the faceId, system will identify the faceId with faces in person groups. If system found a match, a list candidates for that faceId will be returned. A candidate has a personId and confidence percent.     - With the returned personId, system can get the person of that personId and return person information to the user.   + Object Detecting Mode:(phan cua anh Duy)   + View Detecting Mode:     - System will detect and describe what it see from the provided image. | | | |

***Table #: USE CASE – UC\_US04 - <User> Detect Image***

#### <Unauthorized User> Overview Use Case

#thieu hinh

##### <Unauthorized User> Signup (UC\_UU01)

#thieu hinh

#sua lai cac cau message thanh tieng viet

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE – UC\_UU01** | | | |
| **Use Case No.** | UC\_UU01 | **Use Case Version** | 2.0 |
| **Use Case Name** | Signup | | |
| **Author** | DuyNC | | |
| **Date** | 13/02/2017 | **Priority** | Normal |
| **Actor:**   * Unauthorized User   **Summary:**   * This use case allows unauthorized user to signup new account in the system.   **Goal:**   * Unauthorized user cans signup new account in the system.   **Triggers:**   * Unauthorized user sends the signup command.   **Preconditions:**   * **N/A**   **Post Conditions:**   * **Success:** New account has been created in the system. * **Fail:** Show error message.   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | Unauthorized user goes to signup view. | System requires identity information from Unauthorized user:   * Full Name: free text input, required, length 5–50 characters. * Username: free text input, required, length 5–30 characters. * Password: free text input, required, length 5–30 characters. * Gender: select one of the options (Male, Female). * Phone number: free text input, required, length 10-11 digits. * Address: free text input, length 10-100 characters. | | 2 | Unauthorized user inputs information. |  | | 3 | Unauthorized user sends command to signup new account. | Unauthorized user will signup with theirs validates information.  [Alternative 1]  [Exception 1,2,3,4,5,6,7,8,9,10,11,12] |   **Alternative Scenario:**  *[Alternative 1]*   |  |  |  | | --- | --- | --- | | Step. | Actor Action | System Response | | 1 | Unauthorized user sends the cancel command. | System closes signup new account form. |   **Exceptions:**   |  |  |  | | --- | --- | --- | | No. | Cause | System Response | | 1 | Full name is empty. | System shows error message the “Full name must be not empty”. | | 2 | Length of full name out of range 5-50 characters. | System shows error message the “Length of full name must be 5-50 characters”. | | 3 | Username is empty. | System shows error message the “Username must be not empty”. | | 4 | Length of username out of range 5-30 characters. | System shows error message the “Length of username must be 5-30 characters”. | | 5 | Password is empty. | System shows error message the “Password must be not empty”. | | 6 | Length of password out of range 5-30 characters. | System shows error message the “Length of password must be 5-30 characters”. | | 7 | Gender is not selected. | System shows error message the “Gender must be selected”. | | 8 | Phone number is empty. | System shows error message the “Phone number must be not empty”. | | 9 | Length of phone number out of range 10-11 digits. | System shows error message the “Length of phone must be 10-11 digits”. | | 10 | Phone number contains alphabetical characters. | System shows error message the “Phone contains only digits”. | | 11 | Username had already existed in the system. | System shows error message the “Username had already existed”. | | 12 |  | System shows error message the “System is busy” when the internet is lost. |   **Relationships:** N/A.  **Business Rules:**   * Password is encrypted before being sent to server. * System does not allow duplicate “Username”. * After signup in the system, unauthorized user will be redirected to login view. | | | |

##### <Unauthorized User> Login (UC\_UU02)

#thieu hinh

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE – UC\_UU02** | | | |
| **Use Case No.** | UC\_UU02 | **Use Case Version** | 2.0 |
| **Use Case Name** | Login | | |
| **Author** | DuyNC | | |
| **Date** | 13/02/2017 | **Priority** | Normal |
| **Actor:**   * Unauthorized User   **Summary:**   * This use case allows unauthorized user to login in the system.   **Goal:**   * Unauthorized user cans login in the system.   **Triggers:**   * Unauthorized user sends the login command.   **Preconditions:**   * **N/A**   **Post Conditions:**   * **Success:** Unauthorized user login in the system. * **Fail:** Show error message.   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | Unauthorized user goes to login view. | System requires identity information from Unauthorized user:   * Username: free text input, required. * Password: free text input, required. | | 2 | Unauthorized user inputs information. |  | | 3 | Unauthorized user sends command to login in the system. | Unauthorized user will login with theirs specific role.  [Alternative 1]  [Exception 1] |   **Alternative Scenario:**  *[Alternative 1]*   |  |  |  | | --- | --- | --- | | Step. | Actor Action | System Response | | 1 | Unauthorized user enters wrong identity information. | Wrong identity information, system shows error message. |   **Exceptions:**   |  |  |  | | --- | --- | --- | | No. | Cause | System Response | | 1 |  | System shows error message the “System is busy” when the internet is lost. |   **Relationships:** N/A.  **Business Rules:**   * Password is encrypted before being sent to server. * After login in the system, unauthorized user will be redirected to specific view based on their role in the system: admin or user. * If role is “Admin”, the system will display to admin page. * If role is “User”, the system will display to user view. | | | |

##### <Unauthorized User> Train Object (UC\_)

#thieu hinh

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE – UC\_** | | | |
| **Use Case No.** | UC\_ | **Use Case Version** | 2.0 |
| **Use Case Name** | Train Object | | |
| **Author** | DuyNC | | |
| **Date** | 13/02/2017 | **Priority** | Normal |
| **Actor:**   * Admin   **Summary:**   * This use case allows admin train object in the system.   (#viet ro nghia ra https://www.clarifai.com/custom-training)  **Goal:**   * Admin cans train object in the system.   **Triggers:**   * Admin sends the train command.   **Preconditions:**   * Account must be login with admin role.   **Post Conditions:**   * **Success:** New object will be trained in the system. * **Fail:** Show error message.   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | Admin goes to train object view. | System requires object information from admin.   * Image: select options or free image input, required. * Concept: select options or free text input, required. * Model: select options or free text input, required. | | 2 | Admin input information. |  | | 3 | Admin sends command to train object. | System validates information, display object details and requests for confirmation.  [Exception 1,2,3] | | 4 | Admin sends command to train object. | Add new object information in the system. Show successful message and refresh train object view. |   **Alternative Scenario:**   * N/A   **Exceptions:**   |  |  |  | | --- | --- | --- | | No. | Cause | System Response | | 1 | Admin sends command to train object. | System shows error message and ask user input missing required field. | | 2 | Concept is existed in the system. | System show message to notify admin that “Concept” is existed in the system. | | 3 | Model is existed in the system. | System show message to notify admin that “Model” is existed in the system. |   **Relationships:** N/A.  **Business Rules:**   * New object will be trained in the system with inputted information. * When admin trained object: * If images have existed, Admin will select image. * If images have not existed, Admin will upload new image. * If concepts or models have existed, Admin will select concept or model. * If concepts or models have not existed, Admin will input new concepts or model. * System must ensure has no duplicate concept or model. * A notification will be sent admin after the process is completed. | | | |

## Software System Attribute

### Usability

#### Graphic User Interface

* All the texts, labels must be written in English.
* The content of alert, log file can be written in Vietnamese.

#### Usability

* Admin, the trainer should need less than 2 hours of training to be used with the system.

### Reliability

* The log file will be written when application detect face and things unsuccessfully.
* Server have back-up method to make sure that if it has problem while running then all necessary data must be protected and restore as soon as.

### Availability

* Application can use online or offline.
* Application is easy to used.

### Security

* Privacy: Each role of user has a specific permission to interact with system.

### Maintainability

* The system is divided into separated modules.

### Portability

* Admin, the trainer can use application on Google Chrome, version 42 or above.
* Employee can use mobile application on Android that support 6.0.1 Marshmallow or later.

### Performance

* Request from web application are responded in less 8 seconds at 4Mbps bandwidth speed and 2.2GHz processing Speed of CPU.

## Conceptual Diagram

// Duy lam lai.