**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 |
| Friend | Same meaning with Person |
| Friend Group | Same meaning with Person Group |

**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.

### 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.
  + 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 member, 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,  Sublime Text 2 | * .Net framework   + ASP.NET MVC4  + Entity Framework 5   * OpenCV library * Flask framework(Python) |
| **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\_Online Realtime 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 realtime 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 | SonNVH |
| 5 | Project management plan | SonNVH | ThanhPT |
| 6 | Coding convention | ThanhPT | SonNVH,  DuyCT,  QuanVH |
| 7 | Review document | ThanhPT | SonNVH |
| 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 | ThanhPT |
| 13 | Studying Computer Vision Api | SonNVH, ThanhPT | ThanhPT |
| 14 | Using common model | DuyNC | ThanhPT |
| 15 | How to do prediction | DuyNC | ThanhPT |
| 16 | How to train model | DuyNC | ThanhPT |
| 17 | Dealing with images and inputs | DuyNC | ThanhPT |
| 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 | SonNVH, QuanVH, DuyNC | ThanhPT |
| 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 realtime mode recognition

**Development:**

|  |  |  |  |
| --- | --- | --- | --- |
| **No** | **Task** | **Implement** | **Reviewer** |
| 1 | Design Overview | ThanhPT | SonNVH, QuanVH, DuyNC |
| 2 | System Architectural Design | ThanhPT | SonNVH, QuanVH, DuyNC |
| 3 | Component Diagram | ThanhPT | SonNVH, QuanVH, DuyNC |
| 4 | Detailed Description of Components | ThanhPT | SonNVH, QuanVH, DuyNC |
| 5 | Class Diagram | ThanhPT | SonNVH, QuanVH, DuyNC |
| 6 | Sequence Diagram | ThanhPT | SonNVH, QuanVH, DuyNC |
| 7 | User Interface Diagram | ThanhPT, QuanVH, DuyNC | SonNVH |
| 8 | Database Design | ThanhPT, SonNVH | QuanVH, DuyNC |
| 9 | Mobile\_Write Log File | SonNVH | ThanhPT, QuanVH, DuyNC |
| 10 | Web\_User Search + View data | QuanVH | ThanhPT, SonNVH, DuyNC |
| 11 | System\_Implement realtime 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\_Online realtime 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\_Online realtime mode Face Recognition | ThanhPT, SonNVH | SonNVH, 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 | ThanhPT | SonNVH, QuanVH, DuyNC |
| 3 | Test Cases | ThanhPT, QuanVH | DuyNC, SonNVH |
| 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 | DuyNC | ThanhPT |
| 2 | User’s Guide | DuyNC | ThanhPT |

### 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. User can be the blinds or their relatives, partners. There are 2 main function-groups:

* For the blind:
  + Get information about the image:
    - Acquaintances
    - 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:
  + Active / Deactivate account
  + Update account
* Manage person 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.

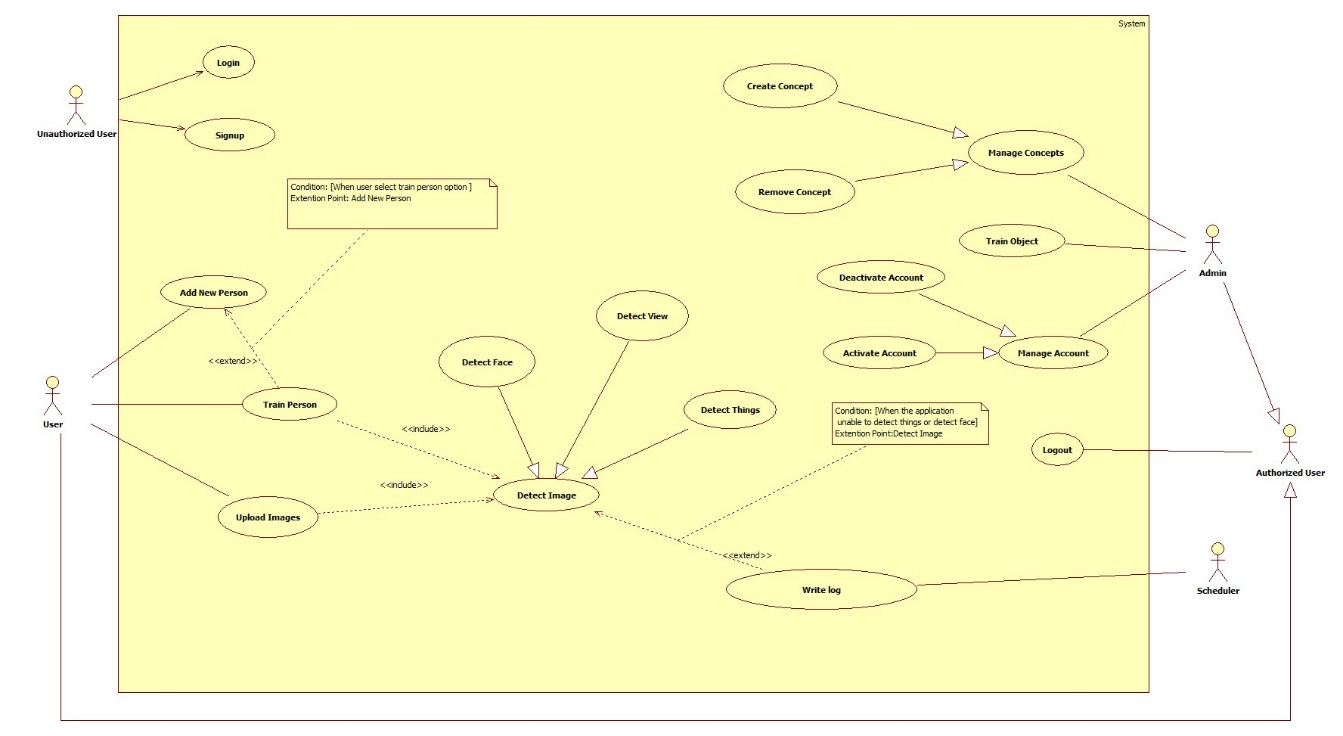
#### Software Interface

* 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.

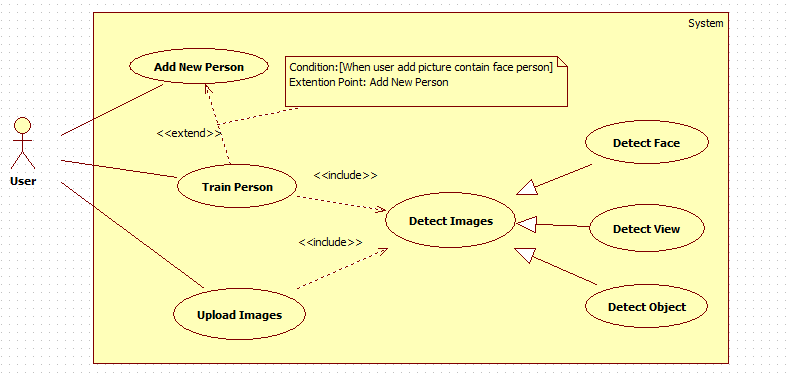
### System Overview Use Case



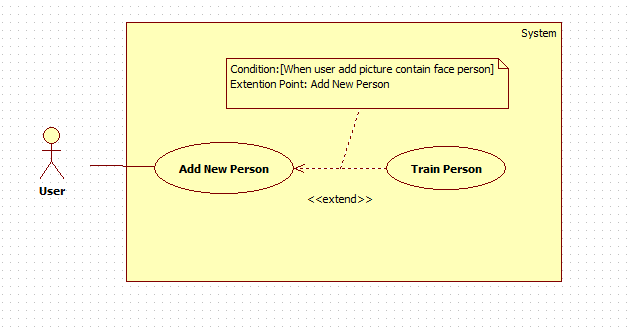
**Figure xxx: System Overview Use Case**

### List of Use Case

#### <User> Overview Use Case



##### <User> Add New Peron



**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** | 03/26/2017 | **Priority** | High |
| **Actor:**   * User   **Summary:**   * This use case allows user to add a new person to the system for later recognition.   **Goal:**   * A new person is added to the system with name and description. The system can be trained later to recognize this person when the user request.   **Triggers:**   * User sends command to add a new person .   **Preconditions:**   * Actor must login at user role.   **Post Conditions:**   * **Success:** New person has been added to the system. * **Fail:** Show error message.   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | User sends command to add new person | System open Add New Person view. | | 2 |  | System requires information from user:   * Name: free text input. Required. length is 3 - 128. * User Description: free text input. The size limit is 128 chracters. * Image: hidden free text input. Valid image size is from 1KB to 4MB. | | 3 | User inputs information. |  | | 4 | User sends command to create new Person.  [Alternative 1, 2] | System validates information.  [Exception 1, 2, 3, 4, 5, 6] | | 5 |  | Adds new person into the system.  Shows successful message.  [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 view. |   **Exceptions:**   |  |  |  | | --- | --- | --- | | No. | Cause | System Response | | 1 | Name is empty. | System responses: “Name không được bỏ trống”. | | 2 | Length of Name is not between 3 - 128 characters. | System responses: “Độ dài của Name trong khoảng 3 -128 kí tự”. | | 3 | Image size exceeded the limit size (4Mb) | System automatically compresses the image but remains its resolution. | | 4 | Invalid subscription Key or user/plan is blocked. | System responses: “Tạo người mới thất bại”. | | 5 | Out of call volume quota | System responses: “Lượng truy cập cho phép đã hết. Truy cập sẽ khả dụng trở lại trong ## ngày”. | | 6 | Rate limit is exceeded | System responses: “Đã đạt lượng truy cập tối đa. Vui lòng chờ trong vào ## giây để tiếp tục”. |   **Relationships:**  N/A.  **Business Rules:**   * New person will be created both in the system database and Microsoft Cognitive Service database. | | | |

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

##### <User> Upload Images

**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 Image | | |
| **Author** | SonNVH | | |
| **Date** | 03/26/2017 | **Priority** | High |
| **Actor:**   * User   **Summary:**   * This use case allows user to send a picture to the system. After received the picture, the system can use it for later functions (detections, create new person).   **Goal:**   * The taken / uploaded picture from user is stored in the system.   **Triggers:**   * When user capture a picture from device’s camera. * When user choose an image(s) to create a new person.   **Preconditions:**   * Actor must login at user role.   **Post Conditions:**   * **Success:** captured / uploaded picture from user is now stored in the system. * **Fail:** No image is stored in the system. Show error message.   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | User captures new picture. [Alternative 1] |  | | 2 |  | System stores captured image, compress the image file. | | 3 |  | System uploads image to server.[Exception 1, 2] |   **Alternative Scenario:**  *[Alternative 1]*   |  |  |  | | --- | --- | --- | | Step. | Actor Action | System Response | | 1 | User choose image(s) from device. | System open image picker. | | 2 | User choose image(s) | System shows preview of chosen image(s) | | 5 | User sends upload command | System stores choosen image(s), compress image file(s). | | 6 |  | System uploads image to server.[Exception 1, 2] |   **Exceptions:**   |  |  |  | | --- | --- | --- | | No. | Cause | System Response | | 1 | No image file is found by the system. | System responses with error message | | 2 | No Internet connection / request denied by upload host. | System responses with upload fail error message. |   **Relationships:** N/A.  **Business Rules:**   * Captured image will be compressed to reduce the size but still remain the resolution. The compressed image size is between 20Kb – 70Kb. * The system must provide option for user to crop out faces in the images. * Captured / Chosen images must have size between 1Kb – 4Mb, with resolution between 36x36 and 4096x4096 pixels. * Support image formats are: JPG, PNG, BMP, GIF (the first frame). * After the upload process, the system will receive a direct link of the image. | | | |

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

##### <User> Train Person

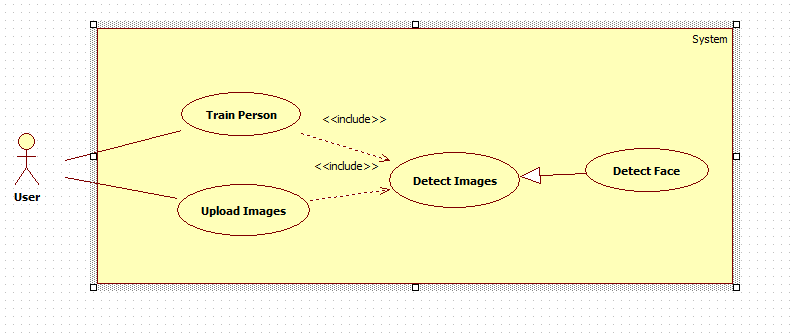


**Figure xxx: <User> Train Person (UC\_US03)**

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE – UC\_US03** | | | |
| **Use Case No.** | UC\_US03 | **Use Case Version** | 1.0 |
| **Use Case Name** | Train Person | | |
| **Author** | SonNVH | | |
| **Date** | 03/26/2017 | **Priority** | High |
| **Actor:**   * User   **Summary:**   * This use case allows user to teach the system how to identify a person that has been added to the system.   **Goal:**   * The system can identify the person that has been added to the system.   **Triggers:**   * After a new person is added to the system successfully. * After user finish updated a person information. * When user sends command to train person.   **Preconditions:**   * Actor must login at user role. * The person group contains the person is not in a training process.   **Post Conditions:**   * **Success:** The system can identify the trained persons . * **Fail:** System cannot identify newly added person. System shows error response.   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | Step. | Actor Action | System Response | | 1 | User sends request to train person. [Alternative 1, 2] | System redirects to Train Person View | | 2 |  | System requires input informations from user:   * personGroupId: hidden free text input. Required. Length between 3 – 64 characters. | | 3 | User input required information | System validate input information from user  [Exception 1, 2, 3] | | 4 | User sends command to train person | System run training process.  Show success message when done.  Redirects to main activity[Exception 4, 5, 6] |   **Alternative Scenario:**  *[Alternative 1]*   |  |  |  | | --- | --- | --- | | Step. | Actor Action | System Response | | 1 | User updated a person’s information. | System gets personGroupId from user, begin training process | | 2 |  | Show success message when done.  Redirect to main activity[Exception 4, 5, 6] |   *[Alternative 2]*   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | User successfully created a new person | System gets personGroupId from user, begin training process | | 2 |  | Show success message when done.  Redirect to main activity[Exception 4, 5, 6] |   **Exceptions:**   |  |  |  | | --- | --- | --- | | No. | Cause | System Response | | 1 | PersonGroupId is not existed | System responses: “Person Group ID không tồn tại”. | | 2 | Person group is in a training process | System responses: “Person group hiện đang trong quá trình training” | | 3 | Out of call volume quota | System responses: “Lượng truy cập cho phép đã hết. Truy cập sẽ khả dụng trở lại trong ## ngày”. | | 4 | Rate limit is exceeded | System responses: “Đã đạt lượng truy cập tối đa. Vui lòng chờ trong vào ## giây để tiếp tục”. |   **Relationships:** N/A.  **Business Rules:**   * Person Group Id is gotten from logged in user. * Image will be compressed when upload to the system. (size) * User can choose to crop out the face in their picture. * If training process is fail, return training status to user. | | | |

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

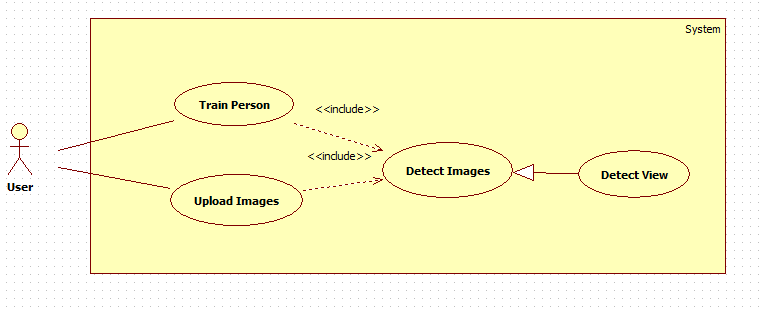
##### <User> Detect Face

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

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE – UC\_US04** | | | |
| **Use Case No.** | UC\_US04 | **Use Case Version** | 1.0 |
| **Use Case Name** | Detect Face | | |
| **Author** | SonNVH | | |
| **Date** | 03/26/2017 | **Priority** | High |
| **Actor:**   * User   **Summary:**   * This usecase allows user to identify a person by taking a picture of him/her. The system will detect and notify back the person identity to user by voice.   **Goal:**   * User can knows who is the person infront of them.   **Triggers:**   * User sends command to detect a person.   **Preconditions:**   * Actor must login at user role. * Person Group training process is finished.   **Post Conditions:**   * **Success:** The person is identified. His / her information is noticed to the user through voice notification. * **Fail:** System cannot identify the person. Notify user through voice notification.   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | User sends request to detect a person | System opens camera. | | 2 | User takes a picture of the person. | System uploads picture to server. [Exception 1] | | 3 |  | System validate inputed information | | 4 |  | System calls Microsoft Cognitive Service to detect faces in the uploaded image.  [Alternative 1]  [Exception 4, 5, 6] | | 5 |  | System call Microsoft Cognitive Service to identify detected face(s) in the person group and return candidate(s)  [Exception 2, 3, 6]  [Alternative 2] | | 6 |  | System get detected person’s information and notify it to the user.  [Exception 6] |   **Alternative Scenario:**  *[Alternative 1]*   |  |  |  | | --- | --- | --- | | Step. | Actor Action | System Response | | 1 | No face is detected in the the image | System response: cannot detect the person. Notify result to user through voice notificaiton. |   *[Alternative 2]*   |  |  |  | | --- | --- | --- | | Step. | Actor Action | System Response | | 1 | No candidates is found in the person group that matched the detected faces. | System response: cannot detect the person. Notify result to user through voice notificaiton. | | 2 |  | System ask if user want to add this person. | | 3 | User sends confirm command  [Alternative 3] | System ask user to input the person name. | | 4 | User input person name. | System write log file of new person.  Return to main screen |   *[Alternative 3]*   |  |  |  | | --- | --- | --- | | Step. | Actor Action | System Response | | 1 | User sends denied command | System returns to main screen waiting for new command. |   **Exceptions:**   |  |  |  | | --- | --- | --- | | No. | Cause | System Response | | 1 | Unable to upload image | System response with error message. | | 2 | Person Group Id not found | System response: “Person group không tồn tại” | | 3 | Person group is in a training process | System responses: “Person group hiện đang trong quá trình training” | | 4 | System detects more than 10 faces in the uploaded image. | System response: “Ứng dụng chỉ nhận biết được tối đa 10 khuôn mặt cùng một lúc” | | 5 | Out of call volume quota | System responses: “Lượng truy cập cho phép đã hết. Truy cập sẽ khả dụng trở lại trong ## ngày”. | | 6 | Rate limit is exceeded | System responses: “Đã đạt lượng truy cập tối đa. Vui lòng chờ trong vào ## giây để tiếp tục”. |   **Relationships:** N/A.  **Business Rules:**   * Person Group Id is gotten from logged in user. * Image will be compressed when uploaded to the system. * System can only detect 10 person at a time. * Identification works well for frontal faces and near-frontal faces. * If no person is identified, the system will describe the gender of the person back to the user through voice notification. * Identification result will be noticed to the user through voice notification. * Log file structure in database:  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | | **ID** | **USER\_ID** | **IMAGE\_URL** | **IMAGE\_TYPE** | **DATETIME** | **ACTIVE** | |  |  |  |  |  |  |  * The detecting face process:  1. Face Detection:    * Microsoft Cognitive Service receives image url, turn it into black and white image.    * Using algorithms, scaning every pixel from left to right to detect the human face.    * When a face is detected, the service return a unique FaceId, along with facial details (age, gender) for each of the detected faces. This faceid will be expired in 24 hours. 2. Face Identification:    * With the return FaceId from detecting process, the system compares and identify with trained faces.    * If there is a match, the system will return a person Id of a candidate which have the highest accuracy percentage.    * If there is no match, the system will return the facial details returned from Detecting Processs back to the user. 3. Get Person info:    * Using the returned Person Id from Identifying process, the system look for that person in the person group.    * If a matched person is found, personal information of that person will be returned to user.    * If there is no match of the person is found, the system will return the facial details back to the user. | | | |

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

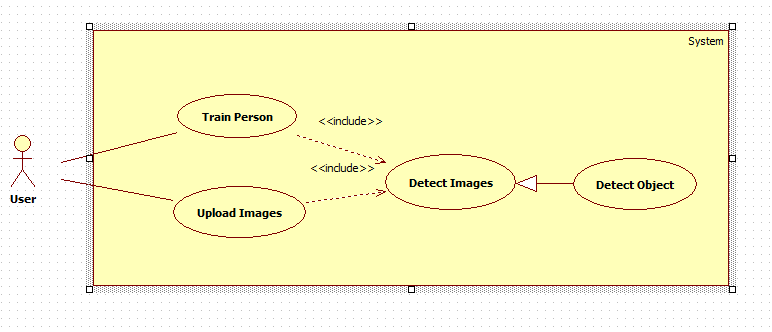
##### <User> Detect View

**Figure xxx: <User> Detect View (UC\_US05)**

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE – UC\_US05** | | | |
| **Use Case No.** | UC\_US05 | **Use Case Version** | 1.0 |
| **Use Case Name** | Detect View | | |
| **Author** | SonNVH | | |
| **Date** | 03/26/2017 | **Priority** | High |
| **Actor:**   * User   **Summary:**   * This usecase allows the application to describe the view / landscape in the taken image by the user though voice notification.   **Goal:**   * User knows what is in the view they took picture of.   **Triggers:**   * User sends command to detect view.   **Preconditions:**   * Actor must login at user role.   **Post Conditions:**   * **Success:**  System describes back to the user the view they took picture of and notify it through voice notification. * **Fail:** System cannot describe what is going on in the view. Notify result to user through voice notification.   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | User sends request to detect view | System opens camera. | | 2 | User takes a picture of the view. | System uploads image to server. [Exception 1] | | 3 |  | System validate inputed information. | | 4 |  | System calls Microsoft Cognitive Service to detect what is going on in the uploaded image.  [Alternative 1]  [Exception 2, 3] | | 5 |  | System notify detection result back to user through voice notification. |   **Alternative Scenario:**  *[Alternative 1]*   |  |  |  | | --- | --- | --- | | Step. | Actor Action | System Response | | 1 | Microsoft Cognitive Service cannot describe the image. | System response: Cannot describe the image. Notify result to user through voice notificaiton. |   **Exceptions:**   |  |  |  | | --- | --- | --- | | No. | Cause | System Response | | 1 | Unable to upload image | System response with error message. | | 2 | Out of call volume quota | System responses: “Lượng truy cập cho phép đã hết. Truy cập sẽ khả dụng trở lại trong ## ngày”. | | 3 | Rate limit is exceeded | System responses: “Đã đạt lượng truy cập tối đa. Vui lòng chờ trong vào ## giây để tiếp tục”. |   **Relationships:** N/A.  **Business Rules:**   * Image will be compressed when uploaded to the system. * Identification result will be noticed to the user through voice notification. | | | |

***Table #: USE CASE – UC\_US05 - <User> Detect View***

##### <User> Detect Object

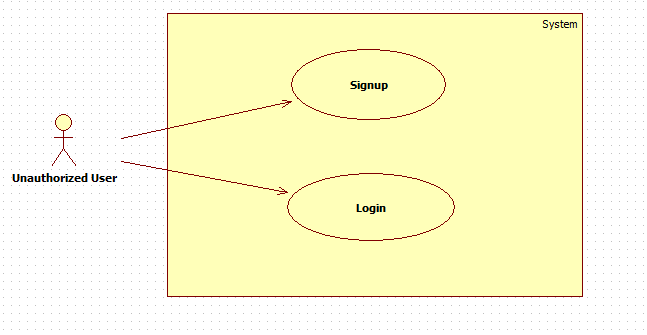
**Figure xxx: <User> Detect Object (UC\_US06)**

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE – UC\_US06** | | | |
| **Use Case No.** | UC\_US07 | **Use Case Version** | 1.0 |
| **Use Case Name** | Detect Object | | |
| **Author** | SonNVH | | |
| **Date** | 03/26/2017 | **Priority** | High |
| **Actor:**   * User   **Summary:**   * This use case allows user to identify what object is infront of them by captured image.   **Goal:**   * System can describes what the object is to user through voice notification.   **Triggers:**   * User sends command to detect an object.   **Preconditions:**   * Actor must login at user role.   **Post Conditions:**   * **Success:**  System describes the object back to the user through voice notification. * **Fail:** System cannot describes the object. Notify result to user through voice notification.   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | User sends request to detect object | System opens camera. | | 2 | User takes a picture of the object. | System upload image to server.[Exception 1] | | 3 |  | System validate inputed information.  [Exception 1] | | 4 |  | System call service to identify object.  [Alternative 1] | | 5 |  | System notify detection result back to user through voice notification. |   **Alternative Scenario:**  *[Alternative 1]*   |  |  |  | | --- | --- | --- | | Step. | Actor Action | System Response | | 1 | System cannot detect object. | System response: Cannot describe the image. Notify result to user through voice notificaiton. | | 2 |  | System generates log file in database. | | 3 |  | Redirect to main activity. |   **Exceptions:**   |  |  |  | | --- | --- | --- | | No. | Cause | System Response | | 1 | No image is upload to the system | System response with error message. | | 2 | No Internet connection | System response with error message. |   **Relationships:** N/A.  **Business Rules:**   * Image will be compressed when uploaded to the system. The compressed image file size is between 20Kb – 70Kb. * Identification result will be noticed to the user through voice notification. * Log file structure in database:  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | | **ID** | **USER\_ID** | **IMAGE\_URL** | **IMAGE\_TYPE** | **DATETIME** | **ACTIVE** | |  |  |  |  |  |  | | | | |

***Table #: USE CASE – UC\_US06 - <User> Detect Object***

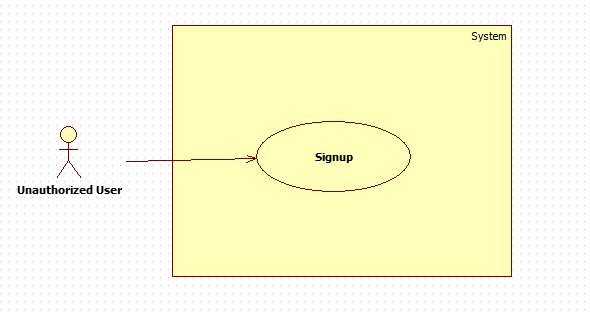
##### <User> Detect Object Realtime

#### <Unauthorized User> Overview Use Case



**Figure xxx: <Unauthorized User> Overview Use Case**

##### <Unauthorized User> Signup

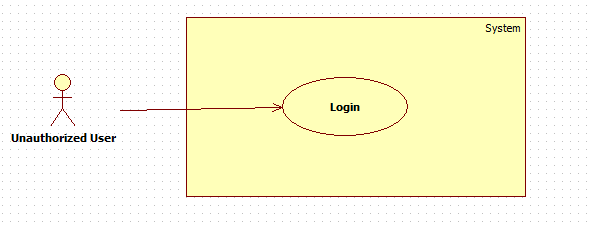


**Figure xxx: <Unauthorized User> Signup (UC\_UU01)**

|  |  |  |  |
| --- | --- | --- | --- |
| **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 register new account in the system for login in the next time.   **Goal:**   * Actor cans signup new account in the system.   **Triggers:**   * Actor 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 | Actor 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), required. * Phone number: free text input, required, length 10-11 digits. * Address: free text input, length 10-100 characters. | | 2 | Actor inputs information. |  | | 3 | Actor sends command to signup new account. | System will sign up with validates inputted 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 “Tên không được để trống”. | | 2 | Length of full name out of range 5-50 characters. | System shows error message the “Độ dài của tên phải trong khoảng từ 5 đến 50 ký tự”. | | 3 | Username is empty. | System shows error message the “Tên đăng nhập không được để trống”. | | 4 | Length of username out of range 5-30 characters. | System shows error message the “Độ dài của tên đăng nhập phải trong khoảng từ 5 đến 30 ký tự”. | | 5 | Password is empty. | System shows error message the “Mật khẩu không được để trống”. | | 6 | Length of password out of range 5-30 characters. | System shows error message the “Độ dài của mật khẩu phải trong khoảng từ 5 đến 30 ký tự”. | | 7 | Gender is not selected. | System shows error message the “Giới tính phải được chọn”. | | 8 | Phone number is empty. | System shows error message the “Số điện thoại không được để trống”. | | 9 | Length of phone number out of range 10-11 digits. | System shows error message the “Độ dài của số điện thoại phải trong khoảng từ 10 đến 11 số”. | | 10 | Phone number contains alphabetical characters. | System shows error message the “Số điện thoại chỉ chứa số”. | | 11 | Username had already existed in the system. | System shows error message the “Tên đăng nhập đã tồn tại”. | | 12 |  | System shows error message the “Hệ thống hiện đang bận. Vui lòng thử lại sau.” 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. | | | |

***Table #: USE CASE – UC\_UU01 - <Unauthorized User > Signup***

##### <Unauthorized User> Login

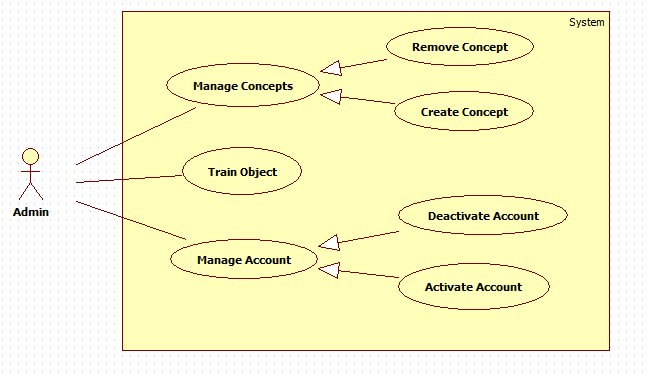


**Figure xxx: <Unauthorized User> Login (UC\_UU02)**

|  |  |  |  |
| --- | --- | --- | --- |
| **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 and use permission function.   **Goal:**   * Actor cans login in the system.   **Triggers:**   * Actor sends the login command.   **Preconditions:**   * Account must be existed in the system.   **Post Conditions:**   * **Success:** Actor login in the system. * **Fail:** Show error message.   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | Actor goes to login view. | System requires identity information from Actor:   * Username: free text input, required. * Password: free text input, required. | | 2 | Actor inputs information. |  | | 3 | Actor sends command to login in the system. | Actor 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 “Hệ thống hiện đang bận. Vui lòng thử lại sau.” 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. | | | |

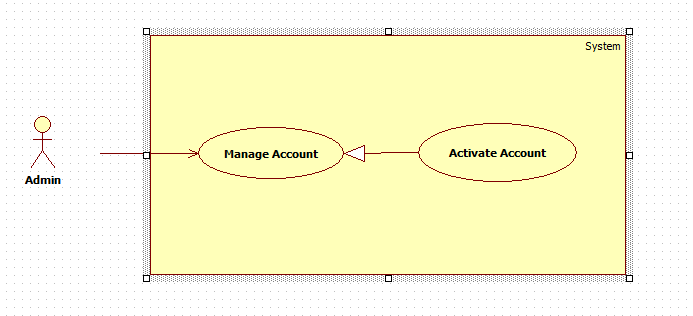
***Table #: USE CASE – UC\_UU02 - <Unauthorized User >Login***

#### <Admin> Overview Use Case



**Figure xxx: <Admin> Overview Use Case**

##### <Admin> Activate Account

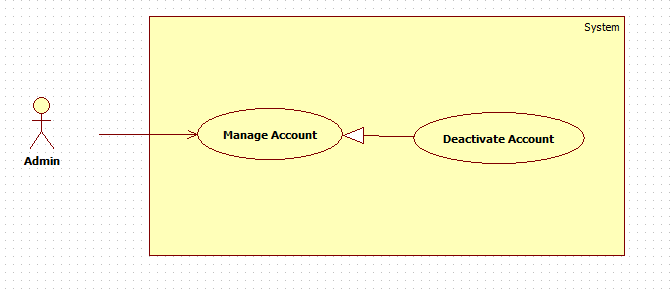


**Figure xxx: <Admin> Activate Account (UC\_AD01)**

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE – UC\_AD01** | | | |
| **Use Case No.** | UC\_AD01 | **Use Case Version** | 1.0 |
| **Use Case Name** | Activate Account | | |
| **Author** | QuanVH | | |
| **Date** | 03/26/2017 | **Priority** | High |
| **Actor:**   * Admin   **Summary:**   * This use case allows admin to activate account.   **Goal:**   * An account is activated and allowed to access the system.   **Triggers:**   * Admin send Activate account command   **Preconditions:**   * Actor must log in the system with “Admin” role. * The account is deactivated.   **Post Conditions:**   * **Success:** An account’s status is updated in the database. * **Fail:** System displays the error message.   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | Admin sends “Activate” command on a specific account | System activates the account and saves the account status to database. |   **Alternative Scenario: N/A**  **Exceptions: N/A**  **Relationships:** N/A.  **Business Rules:**   * An account’s status can be “Đang hoạt động” or “Đã khóa”. “Đang hoạt động” means account is activated and can be used to access the system. “Đã khóa” means account is disabled and can’t be used to access the system. | | | |

***Table #: USE CASE – UC\_AD01 - <Admin >Activate Account***

##### <Admin> Deactivate Account

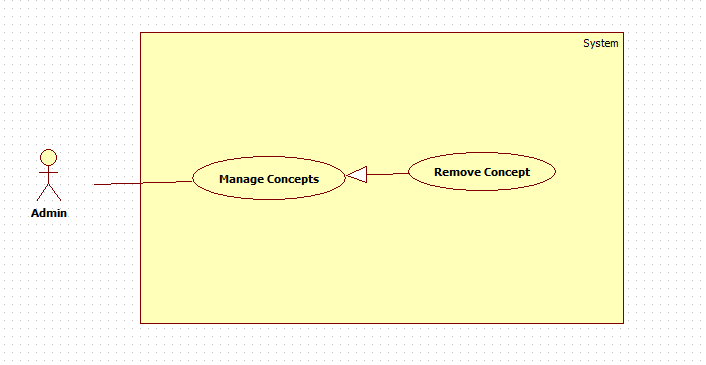


**Figure xxx: <Admin> Deactivate Account (UC\_AD02)**

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE – UC\_AD02** | | | |
| **Use Case No.** | UC\_AD02 | **Use Case Version** | 1.0 |
| **Use Case Name** | Deactivate Account | | |
| **Author** | QuanVH | | |
| **Date** | 03/26/2017 | **Priority** | High |
| **Actor:**   * Admin   **Summary:**   * This use case allows admin to deactivate account.   **Goal:**   * An account is deactivated and can’t access the system.   **Triggers:**   * Admin send Deactivate account command   **Preconditions:**   * Actor must log in the system with “Admin” role. * The selected account is activated.   **Post Conditions:**   * **Success:** An account’s status is updated in the database. * **Fail:** System displays the error message.   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | Admin sends “Deactivate” command on a specific account | System deactivate the account and saves the account status to database. |   **Alternative Scenario: N/A**  **Exceptions: N/A**  **Relationships:** N/A.  **Business Rules:**   * An account’s status can be “Đang hoạt động” or “Đã khóa”. “Đang hoạt động” means account is activated and can be used to access the system. “Đã khóa” means account is disabled and can’t be used to access the system. | | | |

***Table #: USE CASE – UC\_AD02 - <Admin >Deactivate Account***

##### <Admin> Remove Concept

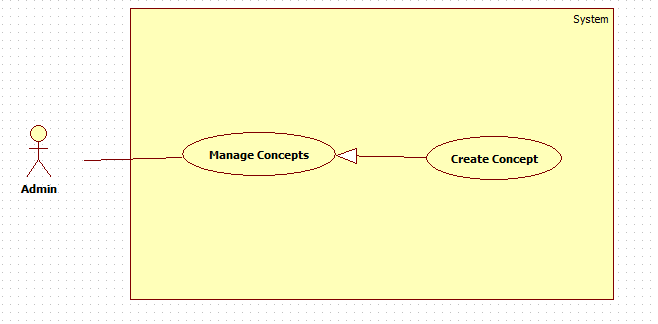


**Figure xxx: <Admin> Remove Concept (UC\_AD03)**

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE – UC\_AD03** | | | |
| **Use Case No.** | UC\_ AD03 | **Use Case Version** | 1.0 |
| **Use Case Name** | Remove Concept | | |
| **Author** | DuyNC | | |
| **Date** | 20/03/2017 | **Priority** | Normal |
| **Actor:**   * Admin   **Summary:**   * This use case allows admin remove concept in the system.   **Goal:**   * Admin cans remove concept in the system.   **Triggers:**   * Admin sends the remove command.   **Preconditions:**   * Account must be login with admin role. * Concept must exist in the system.   **Post Conditions:**   * **Success:** Concept is removed successfully. * **Fail:** Show error message.   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | Admin goes to concept view. | System show list of concept includes: image, name, description. | | 2 | Admin choose a concept. |  | | 3 | Admin sends command to remove concept. | System removes chosen concept. |   **Alternative Scenario:** N/A  **Exceptions:** N/A  **Relationships:** N/A.  **Business Rules:**   * When admin remove concept, system will check: * If images have existed, system will be all images in concept. * System will remove concept and train. * A notification will be sent admin after the process is completed. | | | |

***Table #: USE CASE – UC\_AD03 - <Admin >Remove Concept***

##### <Admin> Create Concept

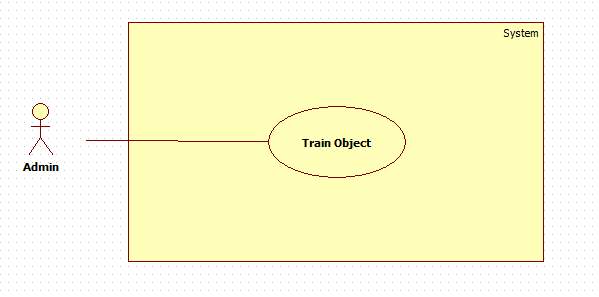


**Figure xxx: <Admin> Create Concept (UC\_AD04)**

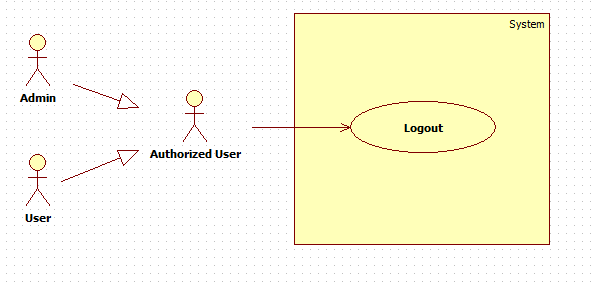
|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE – UC\_AD04** | | | |
| **Use Case No.** | UC\_ AD04 | **Use Case Version** | 1.0 |
| **Use Case Name** | Create Concept | | |
| **Author** | DuyNC | | |
| **Date** | 20/03/2017 | **Priority** | Normal |
| **Actor:**   * Admin   **Summary:**   * This use case allows admin create new concept to the system predict object.   **Goal:**   * The system cans recognize object after being trained.   **Triggers:**   * Admin sends create new concept command.   **Preconditions:**   * Account must be login with admin role.   **Post Conditions:**   * **Success:** Concept is created successfully. * **Fail:** Show error message.   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | Admin goes to create new concept view. | System require information from actor:   * Concept Name: free text input, required. * Description: free text input. * Images: image type input, required. | | 3 | Admin inputs information. |  | | 3 | Admin sends command to create new concept. | System validates inputted information.  [Exception 1,2] | | 4 |  | System create new concept. |   **Alternative Scenario:** N/A  **Exceptions:**   |  |  |  | | --- | --- | --- | | No. | Cause | System Response | | 1 | Concept name is empty. | System shows error message: “Tên đồ vật không được để trống”. | | 2 | Images are empty. | System shows error message: “Vui lòng chọn hình ảnh”. |   **Relationships:** N/A.  **Business Rules:**   * Concept will create after system check input information. * A notification will be sent admin after the process is completed. * System will redirect to all concept list view. * Support image formats are: JPG, PNG, BMP, GIF (the first frame). | | | |

***Table #: USE CASE – UC\_AD04 - <Admin > Create Concept***

##### <Admin> Train Object



#### <Authorized User> Log out



**Figure xxx: <Authorized User> Logout (UC\_AU01)**

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE – UC\_AU01** | | | |
| **Use Case No.** | **UC\_AU01** | **Use Case Version** | 1.0 |
| **Use Case Name** | Log out | | |
| **Author** | QuanVH | | |
| **Date** | 03/26/2017 | **Priority** | Normal |
| **Actor:**   * Authorized User   **Summary:**   * This use case helps Authorized User sign out of their account.   **Goal:**   * Authorized User can sign out of the system.   **Triggers:**   * Authorized User sends Logout command.   **Preconditions:**   * Actor logged in system with role “Authorized User”.   **Post Conditions:**   * **Success:** Authorized User is logged out. * **Fail:** N/A   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | Authorized User sends logout command. | System clears the current session of the user then redirects to home page. |   **Alternative Scenario: N/A**  **Exceptions: N/A**  **Relationships:** N/A.  **Business Rules:**   * After logging out of the system, user is no longer has permission to access any authorized function of the system. | | | |

***Table #: USE CASE – UC\_AU01 - <Authorized User >Logout***

## Software System Attribute

### Usability

#### Graphic User Interface

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

#### 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 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

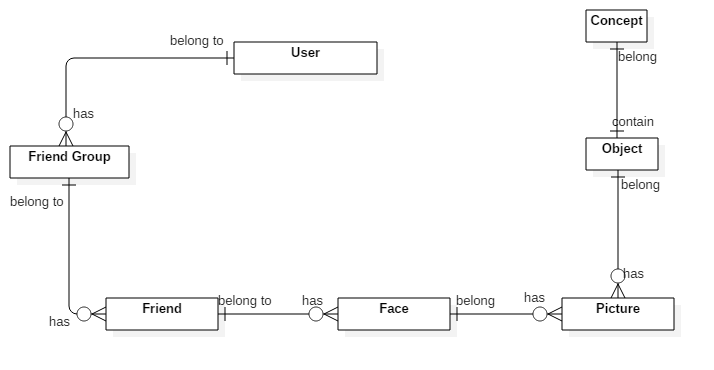
* Admin, the trainer can use application on Google Chrome, version 42 or above.
* Employee can use mobile application on Android that support 6.0.1 Marshmallow or later.

### Performance

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

## Conceptual Diagram

#User->fri group 1-n nhung duoi la 1-1



**Figure xxx: Conceptual Diagram**

**Data Dictionary**

|  |  |
| --- | --- |
| **Entity Name** | **Description** |
| User | The person who using the system |
| Friend Group | Contain all properties about friend group |
| Friend | Contain all properties about friend of the blind |
| Face | Contain all properties about face of Friend |
| Picture | Store all picture of object |
| Object | Contain all properties of object |
| Concept | Contain all properties of concept |

# Software Design Description

## Design Overview

* 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 1: gives an overall description of the system architecture design.
* Section 2: gives component diagrams that describe the connection and integration of the system.
* Section 3: gives the detail design description, which includes class diagram, class explanation, and sequence diagram to details the application functions.
* Section 4: gives the interface design description, which includes component interface, web application interface and mobile application design.
* Section 5: describe a fully attributed Entity Relationship Diagram.
* Section 6: describe the algorithms that apply to the system

## System Architectural Design

### Mobile Application Architecture Description

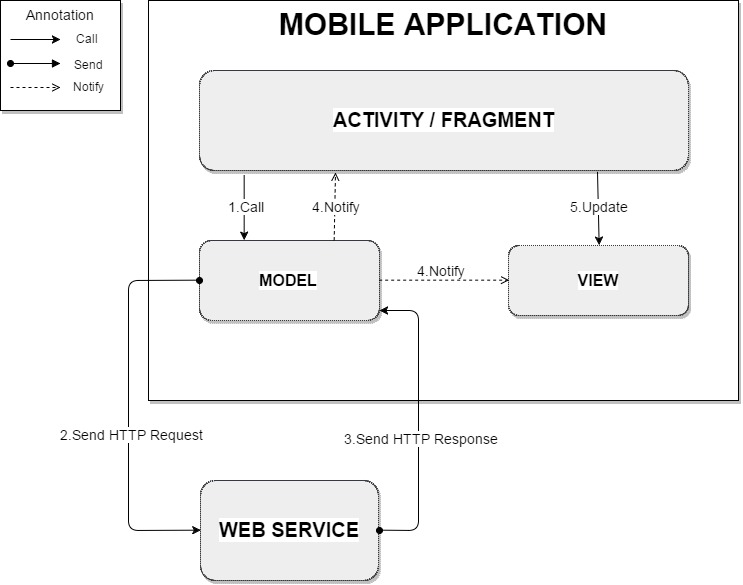
****

Figure xx: Mobile Application Architecture Description

The mobile application is developed in android platform and the application architecture conforms Android architecture.

+ Controller(Activity/Fragment): the logic layer, gets notified of the user’s behavior and updates the Model as needed.

+ View: the UI layer - a visualization of the data from the Model.

+ Model: the data layer, responsible for managing the business logic and handling network or database API.

### Web Service Architecture Description

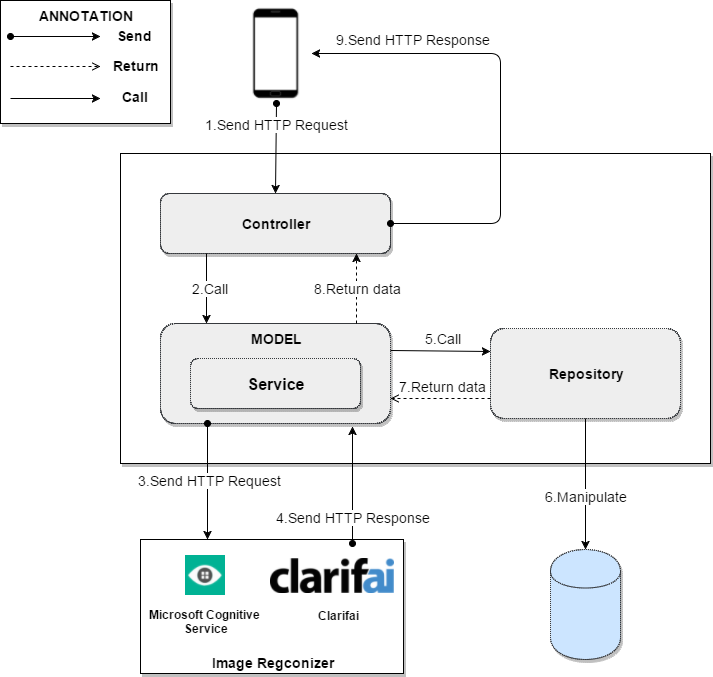


Figure xx: Web Service Architecture Description

### Web Application Architecture Description

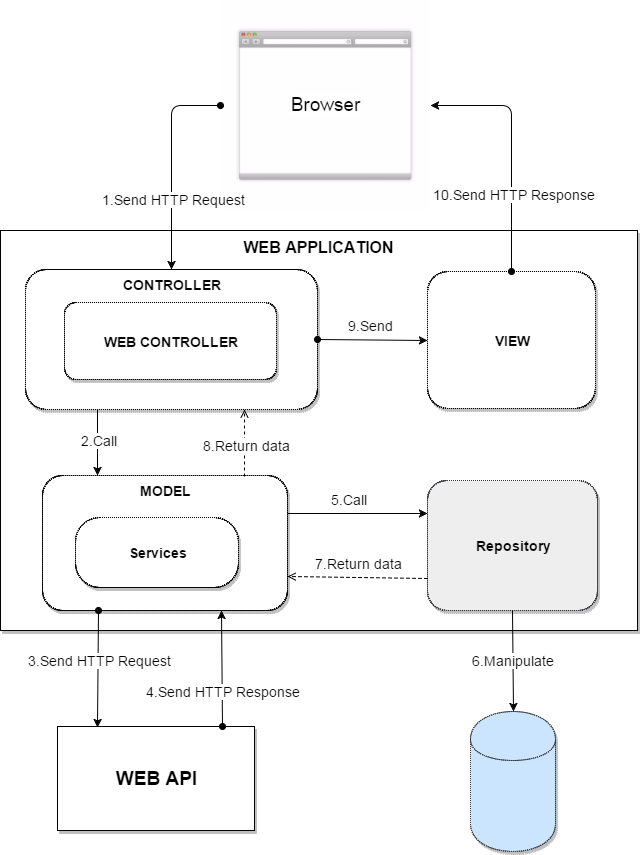
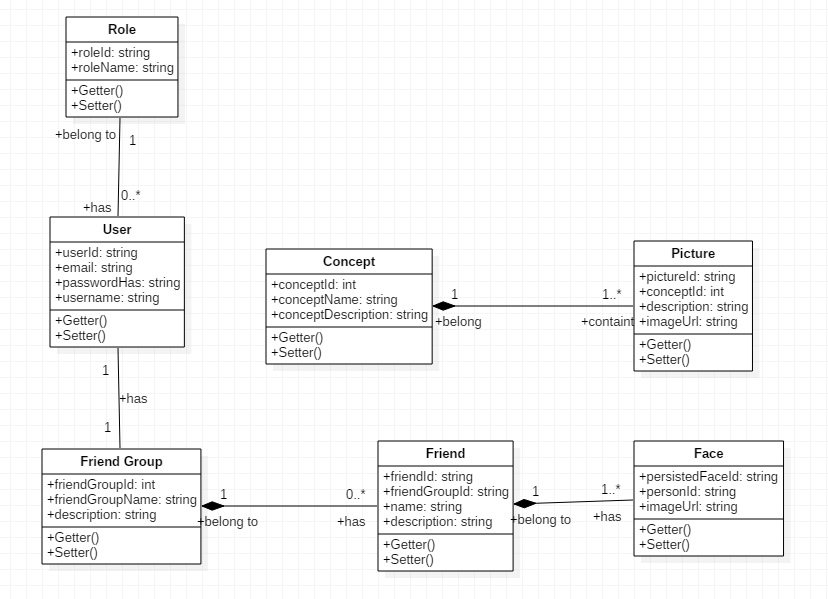


Figure xx: Web Application Architecture Description

## Component Diagram

## Detailed Description

### Class Diagram



**Figure xxx: Class Diagram**

### Class Diagram Explanation

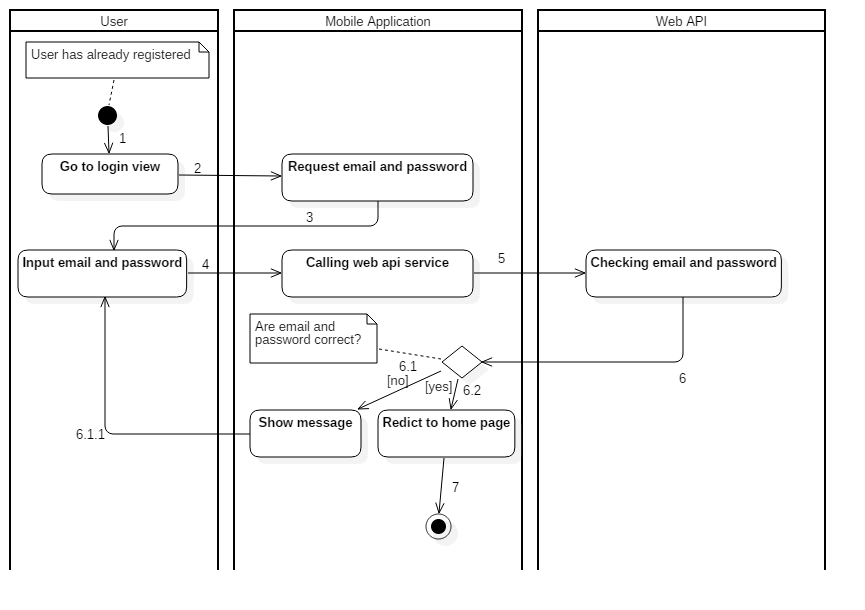
### Interaction Diagram

#### Mobile Application

##### Unauthorized User

###### <Unauthorized User> Login

Summary: This diagrams show the process of unauthorized user login into the system using mobile application

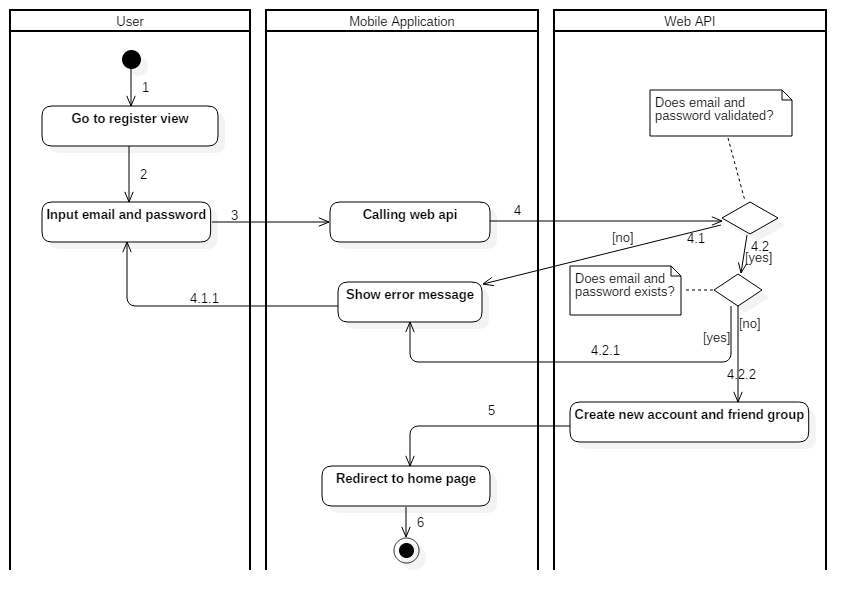


**Figure xxx: Activity Diagram - <Unauthorized User> Login**

|  |  |  |
| --- | --- | --- |
| 1. Start | 2. Go to Login view | 3. Request email and password |
| 4. Input email and password | 5. Calling web api service | 6. Check emal and password in database |
| 6.1. Invalid email or password | 6.2. Valid email and password. Go to home page | 6.1.1. Show error message and go to login page. |

###### <Unauthorized User> Register

Summary: This diagram show the process of unauthorized user register new account to login in the system.



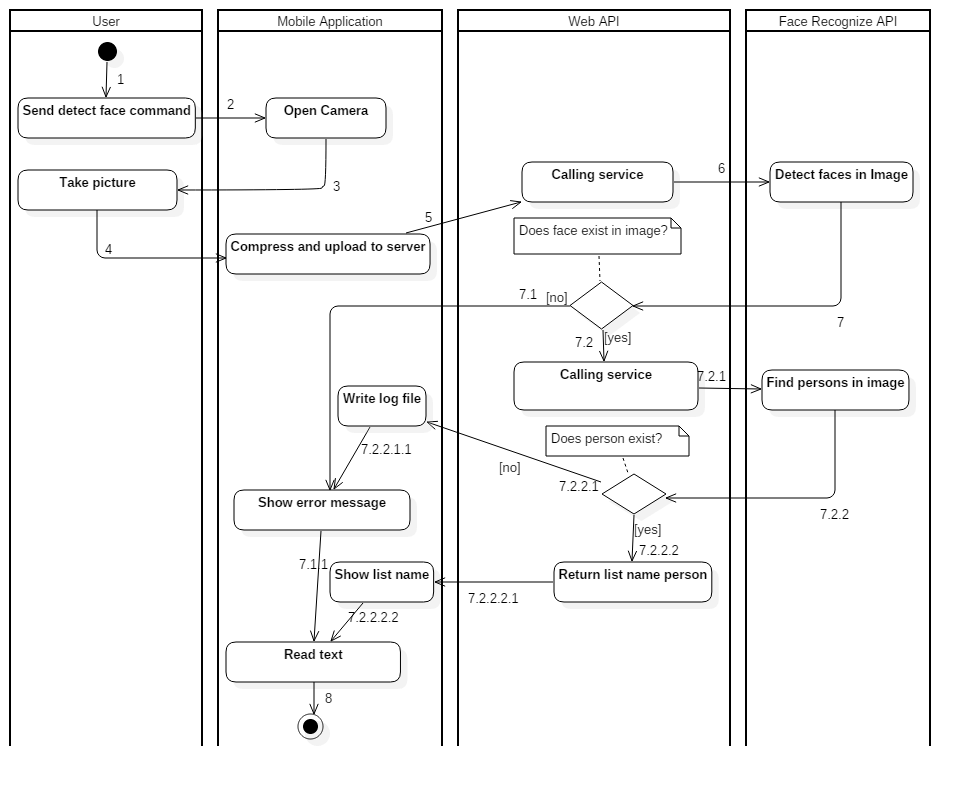
**Figure xxx: Activity Diagram - <Unauthorized User> Register**

|  |  |  |
| --- | --- | --- |
| 1. Start | 2. Go to register view | 3. User input email and password |
| 4. Call web api to check. | 4.1. Email or password invalid. | 4.2 Email and password is validated. |
| 4.1.1. Show error message to tell user input again. | 4.2.1. Email exists in database. | 4.2.2. Email not exist in database. |
| 5. Create new account with data user input also create friend group with id is userid. | 6. Redirect to home page |  |

##### Authorized User

###### <Authorized User> Detect Face

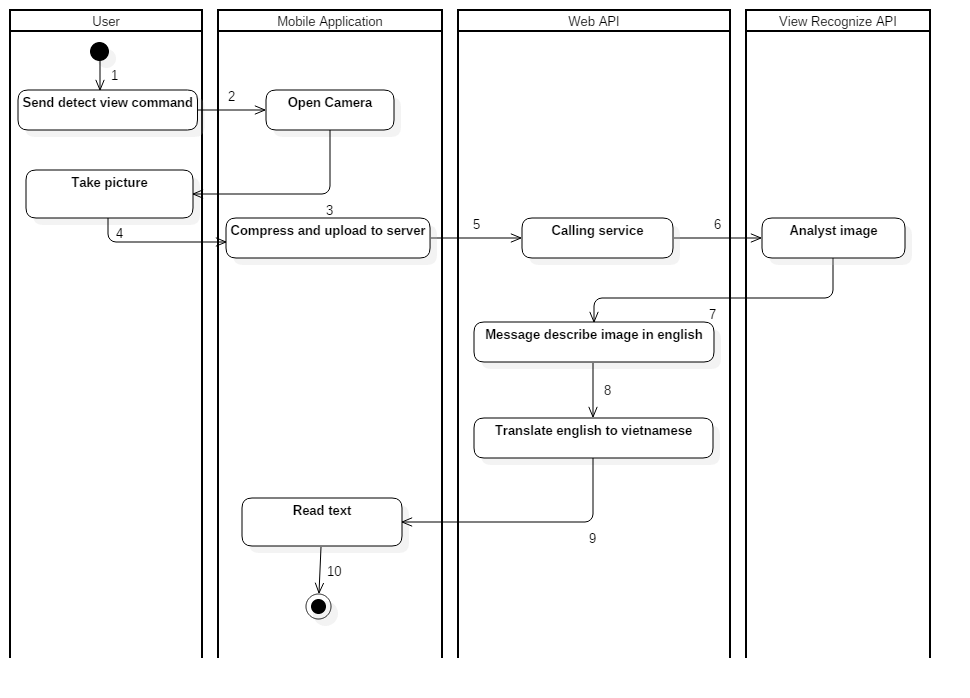
Summary: This diagrams show the process of authorized user detect face to know who stand before with picture using mobile application.



**Figure xxx: Activity Diagram - <Authorized User> Detect Face**

|  |  |  |
| --- | --- | --- |
| 1. Start | 2. Send detect face commnd. | 3. Open camera. |
| 4. User send command to check picture. | 5. Compress image and upload to server. | 6. Call service. |
| 7. Detect faces in image. | 7.1. Image do not contain any face. | 7.2. Image contain face. |
| 7.1.1. Show error message. | 7.2.1. Calling service. | 7.2.2. Find person in image |
| 7.2.2.1. Image do not contain any person. | 7.2.2.1.1. Write to log file. | 7.2.2.2. Image contain person. |
| 7.2.2.2.1. Return list name of person was detected. | 7.2.2.2.2. Show list name of person was detected. | 8. Text to speech. |

###### <Authorized User> Detect View

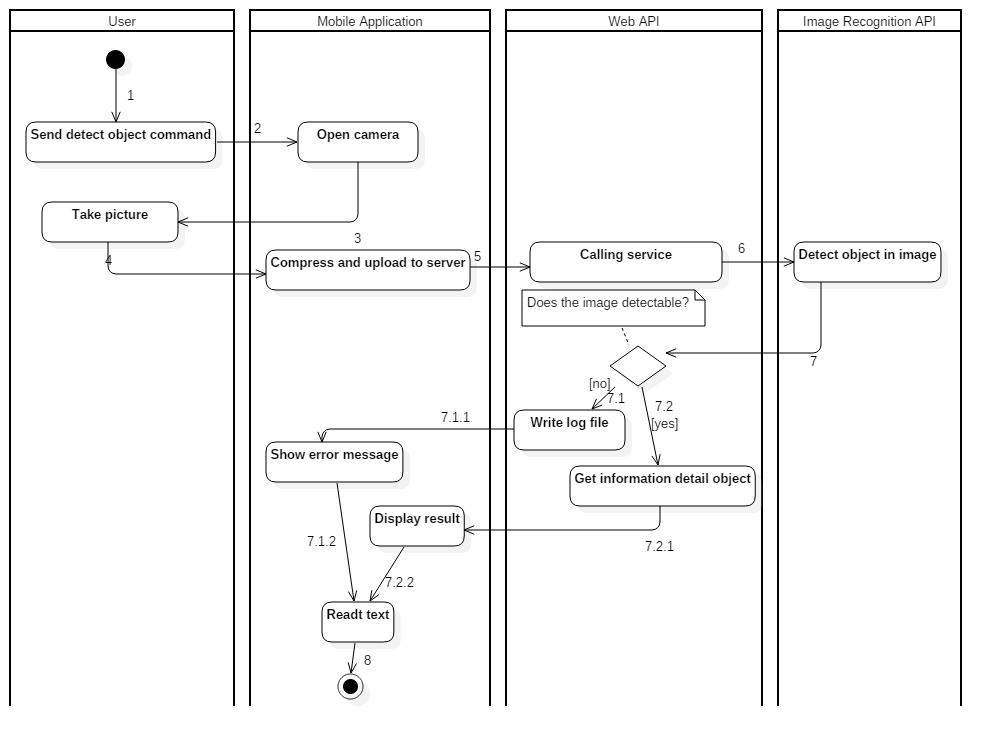
Summary: This diagrams show the process of authorized user detect view by capture picture using mobile application. 

**Figure xxx: Activity Diagram - <Authorized User> Detect View**

|  |  |  |
| --- | --- | --- |
| 1. Start. | 2. User send detect view command. | 3. Open camera. |
| 4. User send command to take picture. | 5. Compress image and upload to server. | 6. Call service to detect view. |
| 7. Analyst image | 8. Return message describe image in english. | 9. Translate english to vietnamese. |
| 10. Text to speech |  |  |

###### <Authorized User> Detect Object

Summary: This diagram show the process detect things in the front of the Authorized User by using Mobile Application.

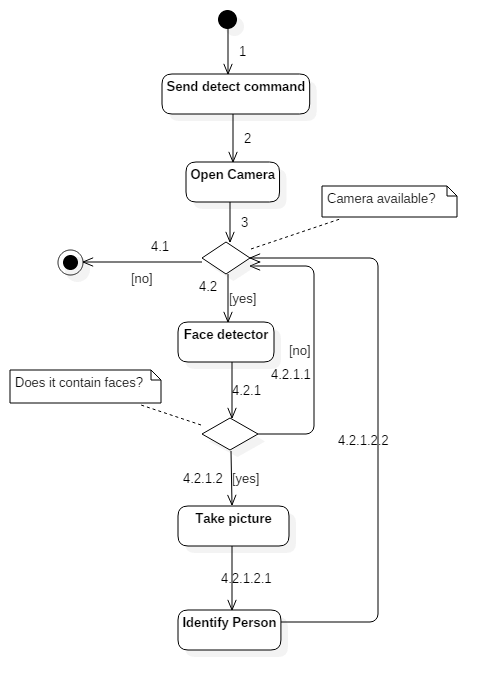


**Figure xxx: Activity Diagram - <Authorized User> Detect Object**

|  |  |  |
| --- | --- | --- |
| 1. Start. | 2. User send detect object command. | 3. Open camera. |
| 4. User send command to take picture. | 5. Compress image and update to server. | 6. Call service to detect object. |
| 7. Detect object in image. | 7.1. Do not know object in image. | 7.2. Know the object in the image. |
| 7.1.1. Show the error message. | 7.1.2. Show the error message. | 7.2.1. Get information detail of the object. |
| 7.2.2. Display result. | 8. Text to speech. |  |

###### <Authorized User> Detect Person Realtime

Summary: This diagram show the process of Authorized User detect person realtime using mobile application



**Figure xxx: Activity Diagram - <Authorized User> Detect Person Realtime**

|  |  |  |
| --- | --- | --- |
| 1. Start. | 2. Send detect person realtime command. | 3. Open camera. |
| 4.1. Camera not available. | 4.2. Camera is available. | 4.2.1. Detect face in frame. |
| 4.2.1.1. Do not have face in frame. | 4.2.1.2. Frame contain face | 4.2.1.2.1. Take picture when frame contain face. |
| 4.2.1.2.2. Identify person in image |  |  |

#### Web Application

##### Admin

###### <Admin> Create Concept

Summary: This diagram show the process of admin add new object for detect using web application.

D:\CN9\CapstoneProject\Report\Report4\Interaction\sequencediagram_createconcept.png

**Figure xxx: Sequence Diagram - <Admin > Create Concept**

###### <Admin> Edit Concept

Summary: This diagram show the process of Admin edit concept using the web application.

D:\CN9\CapstoneProject\Report\Report4\Interaction\sequencediagram_editconcept.png

**Figure xxx: Sequence Diagram - <Admin >Edit Concept**

###### <Admin> Get Log Object

Summary: This diagram show the process of Admin getting logs using the web application.

D:\CN9\CapstoneProject\Report\Report4\Interaction\sequendiagram_getLogObject.png

**Figure xxx: Sequence Diagram - <Admin > Get Log Object**

###### <Admin> Train Object

##### User

###### <User> Get Log Person

Summary: This diagram show the process of User getting logs using web application.

D:\CN9\CapstoneProject\Report\Report4\Interaction\sequendiagram_getLogPerson.png

**Figure xxx: Sequence Diagram - <Admin > Get Log Person**

## Interface

### Component interface

### User Interface Design

#### Mobile Application Design

##### Unauthorized User

###### <Unauthorized User> Register

###### <Unauthorized User> Login

##### Authorized User

###### <Authorized User> Detect Face

###### <Authorized User> Detect View

###### <Authorized User> Detect Object

###### <Authorized User> Add Person

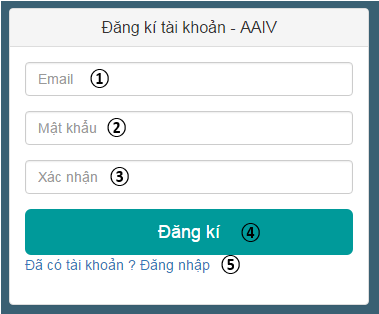
###### <Authorized User> View Log

###### <Authorized User> Update Command Voice

#### Web Application Design

##### Unauthorized User

###### <Unauthorized User> Register



**Figure xxx: Web Application Interface - <Unauthorized> Register**

**Fields:**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **No** | **Field Name** | **Description** | **Read Only** | **Mandatory** | **Control Type** | **Data Type** | **Length** |
| **1** | Email | Fill in email | No | Yes | Textbox | String | N/A |
| **2** | Mật khẩu | Fill in password | No | Yes | Password | String | N/A |
| **3** | Xác nhận | Fill in confirmation password | No | Yes | Password | String | N/A |

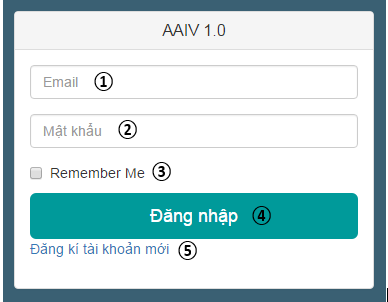
***Table #: Web Application Interface – Register - Fields***

**Buttons/Hyperlinks:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No** | **Function** | **Description** | **Validation** | **Outcome** |
| **4** | Đăng ký | Sign up a new account |  | Redirect to home page for user |
| **5** | Đã có tài khoản? Đăng nhập | Go back to login page | N/A | Redirect to login page |

***Table #: Web Application Interface – Register – Buttons/Hyperlinks:***

###### <Unauthorized User> Login



**Figure xxx: Web Application Interface - <Unauthorized> Login**

**Fields:**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **No** | **Field Name** | **Description** | **Read Only** | **Mandatory** | **Control Type** | **Data Type** | **Length** |
| **1** | Email | Fill in email | No | Yes | Textbox | String | N/A |
| **2** | Mật khẩu | Fill in password | No | Yes | Password | String | N/A |
| **3** | Remember me | Check for saving account | No | No | Checkbox |  | N/A |

***Table #: Web Application Interface – Login - Fields***

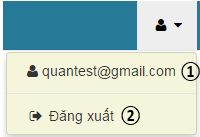
**Buttons/Hyperlinks:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No** | **Function** | **Description** | **Validation** | **Outcome** |
| **4** | Đăng nhập | Login to system | N/A | Redirect to home page |
| **5** | Đăng ký tài khoản mới | Sign up a new account | N/A | Redirect to signup page |

***Table #: Web Application Interface – Login – Buttons/Hyperlinks***

##### Authorized User

###### <Authorized User> Logout



**Figure xxx: Web Application Interface - <Authorized> Logout**

**Fields:**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **No** | **Field Name** | **Description** | **Read Only** | **Mandatory** | **Control Type** | **Data Type** | **Length** |
| **1** | User | User account | No | Yes | Label | String | N/A |

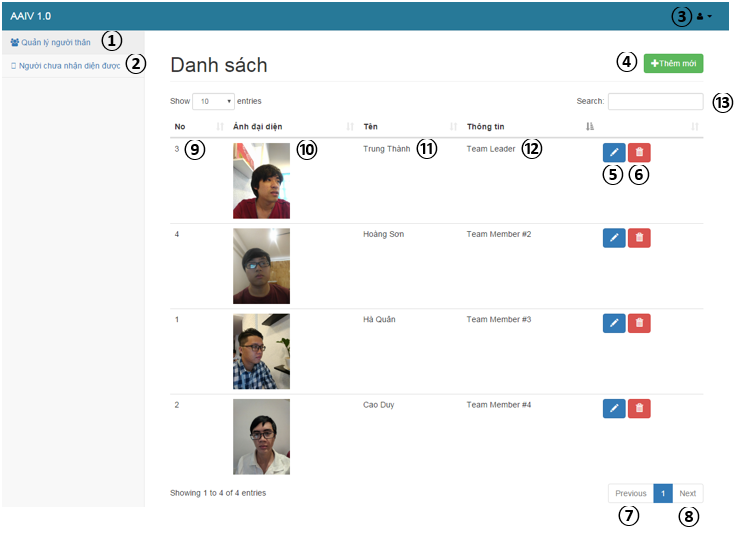
***Table #: Web Application Interface – Logout - Fields***

**Buttons/Hyperlinks:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No** | **Function** | **Description** | **Validation** | **Outcome** |
| **2** | Đăng xuất | Send logout command | N/A | Destroy session and redirect to login page |

***Table #: Web Application Interface – Logout - Buttons/Hyperlinks***

###### <Authorized User> View Persons



**Figure xxx: Web Application Interface - <Authorized> View Perons**

**Buttons/Hyperlinks:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No** | **Function** | **Description** | **Validation** | **Outcome** |
| **1** | getPersons | Get all persons of user | N/A | Show list of all persons |
| **2** | showLogs | Get all logs (unidentified persons) of user | N/A | Show list of all logs |
| **3** | openContextMenu | Open context menu | N/A | Context menu is opened |
| **4** | addNewPerson | Add new person to system | N/A | Redirect to new person view |
| **5** | updatePerson | Modify information and photos of person | N/A | Redirect to person’s info view |
| **6** | deletePerson | Delete a person | N/A | Person is deleted out of the system |
| **7** | btnPrevious | Go back to previous person page | N/A | Show previous page |
| **8** | btnNext | Go to next person page | N/A | Show next page |

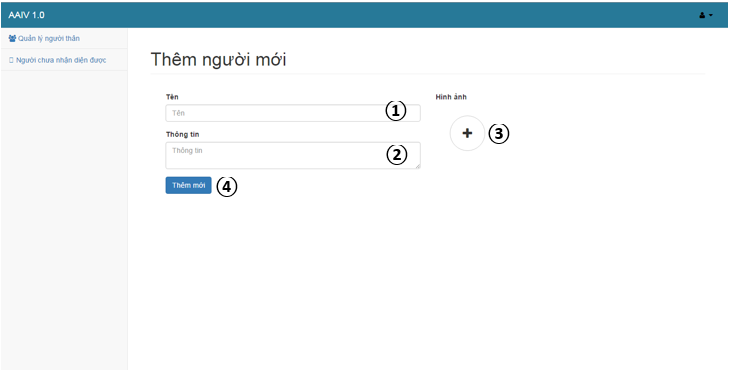
***Table #: Web Application Interface –View Persons - Buttons/Hyperlinks***

**Fields:**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **No** | **Field Name** | **Description** | **Read Only** | **Mandatory** | **Control Type** | **Data Type** | **Length** |
| **9** | Thứ tự | The position number | N/A | Yes | Label | String | N/A |
| **10** | Ảnh đại diện | The avatar of person | N/A | Yes |  | Image | N/A |
| **11** | Tên | The name of person | N/A | Yes | Label | String | N/A |
| **12** | Thông tin | The description of person | N/A | No | Label | String | N/A |
| **13** | Tìm kiếm |  |  |  |  |  |  |

***Table #: Web Application Interface – View Persons - Fields***

###### <Authorized User> Add New Person



**Figure xxx: Web Application Interface - <Authorized> Add New Person**

**Fields:**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **No** | **Field Name** | **Description** | **Read Only** | **Mandatory** | **Control Type** | **Data Type** | **Length** |
| **1** | Tên | Fill in person’s name | No | Yes | Textbox | String | N/A |
| **2** | Thông tin | Fill in person’s information | No | No | Textbox | String | N/A |

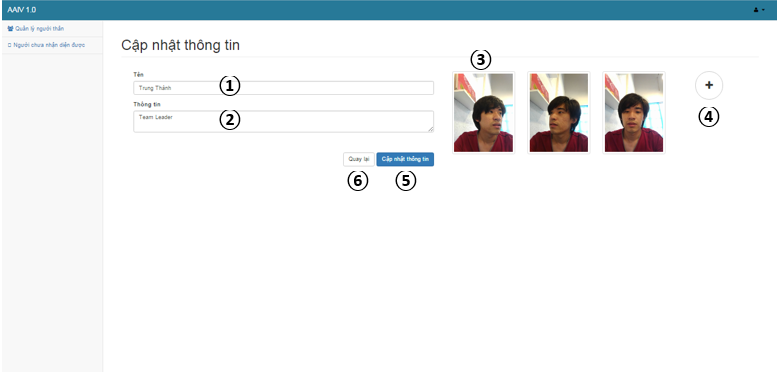
***Table #: Web Application Interface – Add New Person - Fields***

**Buttons/Hyperlinks:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No** | **Function** | **Description** | **Validation** | **Outcome** |
| **3** | addImage | Add new images to current person | N/A | Selected images preview is displayed |
| **4** | addPerson | A new person with selected image will be added to system | N/A | Redirect to Index page |

***Table #: Web Application Interface – Add New Person - Buttons/Hyperlinks***

###### <Authorized User> Update Person



**Figure xxx: Web Application Interface - <Authorized> Update Person**

**Fields:**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **No** | **Field Name** | **Description** | **Read Only** | **Mandatory** | **Control Type** | **Data Type** | **Length** |
| **1** | Tên | Fill in person’s name | No | Yes | Textbox | String | N/A |
| **2** | Thông tin | Fill in person’s information | No | No | Textbox | String | N/A |

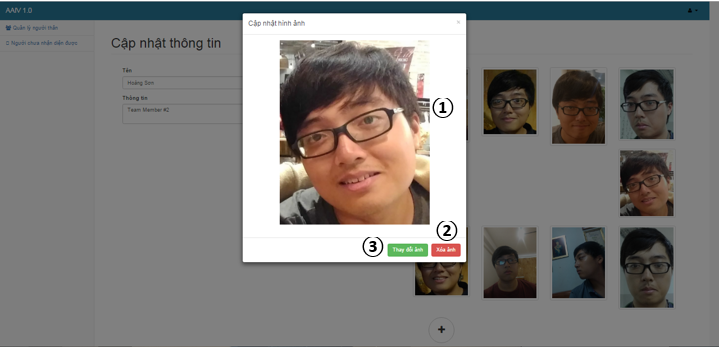
***Table #: Web Application Interface – Update Person - Fields***

**Buttons/Hyperlinks:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No** | **Function** | **Description** | **Validation** | **Outcome** |
| **3** | updateFace | Show selected image information that can be replaced or deleted | N/A | A dialog contain selected image is displayed |
| **4** | addNewFace | Add new face image to current person | N/A | A file choosing dialog is displayed; selected face images will be added |
| **5** | updatePersonInfo | Inputed person’s information will be updated to current person | “Tên” cannot be blank | Update current person’s info and redirect to Index page |
| **6** | Quay lại | Go back to the previous page | N/A | Redirect to Index page |

***Table #: Web Application Interface – Update Person - Buttons/Hyperlinks***

###### <Authorized User> Update Person’s Face



**Figure xxx: Web Application Interface - <Authorized> Update Person’s Face**

**Fields:**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **No** | **Field Name** | **Description** | **Read Only** | **Mandatory** | **Control Type** | **Data Type** | **Length** |
| **1** | Hình ảnh | Person’s face image | Yes | Yes |  | Image | N/A |

***Table #: Web Application Interface – Update Persons’s Face - Fields***

**Buttons/Hyperlinks:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No** | **Function** | **Description** | **Validation** | **Outcome** |
| **2** | deleteFace | Delete selected face image | N/A | Selected image is deleted and redirect to update person view |
| **3** | updateFace | Replace the current image by the selected face image | N/A | A new face image is updated and redirect to update person view |

***Table #: Web Application Interface – Update Person’s Face - Buttons/Hyperlinks***

###### <Authorized User> Show Logs Person



**Figure xxx: Web Application Interface - <Authorized> Show Logs Person**

**Fields:**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **No** | **Field Name** | **Description** | **Read Only** | **Mandatory** | **Control Type** | **Data Type** | **Length** |
| **1** | Hình ảnh | Log image | No | Yes |  | Image | N/A |
| **2** | Tên | Name of the person in log | No | Yes | Label | String | N/A |
| **3** | Ngày tạo | The created date of log | No | Yes | Label | DateTime | N/A |

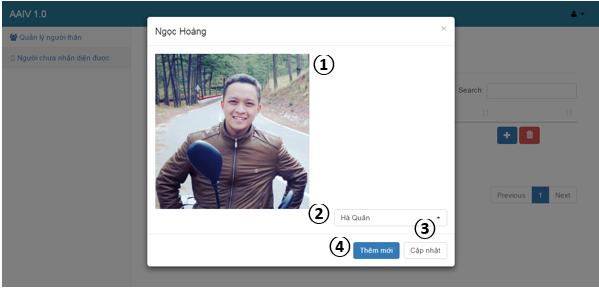
***Table #: Web Application Interface – Show Logs From Person - Fields***

**Buttons/Hyperlinks:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No** | **Function** | **Description** | **Validation** | **Outcome** |
| **4** | addFaceFromLog | Add face from log to person | N/A | The dialog with selected log image is displayed for adding |
| **5** | deleteLog | Delete the current log | N/A | The log is deleted and redirect to the Log view |

***Table #: Web Application Interface – Show Logs From Person - Buttons/Hyperlinks***

###### <Authorized User> Create/Update From Log Person



**Figure xxx: Web Application Interface - <Authorized> Create/Update From Log Person**

**Fields:**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **No** | **Field Name** | **Description** | **Read Only** | **Mandatory** | **Control Type** | **Data Type** | **Length** |
| **1** | Hình ảnh | Log image | No | Yes |  | Image | N/A |
| **2** | Tên | Choose an existed person from list | No | Yes | Dropdown | String | N/A |

***Table #: Web Application Interface – Create/Update From Log Person - Fields***

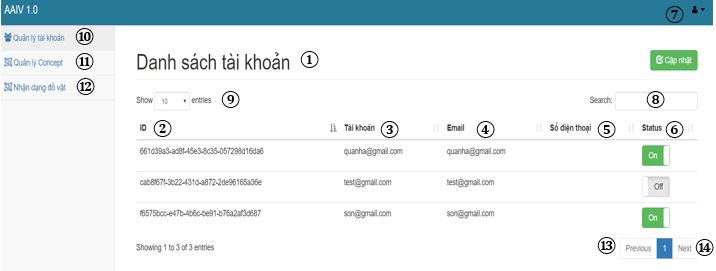
**Buttons/Hyperlinks:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No** | **Function** | **Description** | **Validation** | **Outcome** |
| **3** | updateFromLog | Update the current face image to selected person from the list | N/A | Log face image is updated to selected person; current log is deleted and redirect to the Show Logs view |
| **4** | createNewFromLog | The current face image is added to new person with saved information | N/A | Redirect to Add New Person view with saved information |

***Table #: Web Application Interface – Create/Update From Log Person - Buttons/Hyperlinks***

##### Admin

###### <Admin> Manage User



**Figure xxx: Web Application Interface - < Admin > Manager User**

**Fields:**

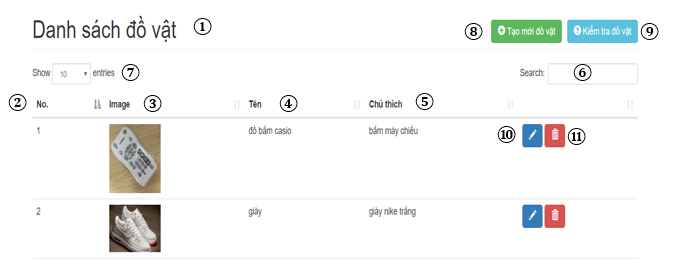
|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **No** | **Field name** | **Description** | **Read only** | **Mandatory** | **Control type** | **Data type** | **Length** |
| 1 | Danh sách tài khoản | Show list user | Yes | Yes | Label | String | N/A |
| 2 | ID | ID of user | Yes | Yes | Table | String | N/A |
| 3 | Tài Khoản | Username of user | Yes | Yes | Table | String | N/A |
| 4 | Email | Email of user to log in into system | Yes | Yes | Table | String | N/A |
| 5 | Số điện thoại | Phone number of user | Yes | Yes | Table | String | N/A |
| 6 | Status | Status of user to know user is active or deactive | Yes | Yes | Checkbox | bit | N/A |
| ***Table #: Web Application Interface – Manage User - Fields*** | | | | | |  |  |

**Buttons/Hyperlinks:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No** | **Function** | **Description** | **Validation** | **Outcome** |
| 7 | Account | Information of account | N/A | Show information of account. |
| 8 | Search | Search data in table | N/A | Show data that are search |
| 9 | Show entries | Show the number of entries in table | N/A | Show the number of entries that are selected |
| 10 | getUser | Get to list all user view | N/A | Redirect to list of all user view |
| 11 | getAlslconcept | Get to list all object view | N/A | Redirect to list of all object view |
| 12 | getLog | Get to list all user view | N/A | Redirect to User view |
| 13 | Previous | Go to previous page | N/A | Go to previous page contain list data |
| 14 | Next | Go to next page | N/A | Go to next page contain list data |

***Table #: Web Application Interface – Manage User - Buttons/Hyperlinks***

###### <Admin> View Concepts



**Figure xxx: Web Application Interface - < Admin > View Concepts**

**Fields:**

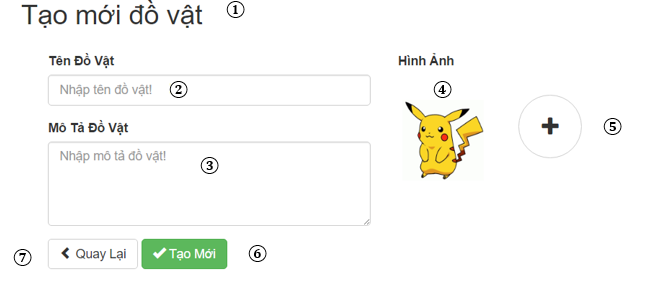
|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **No** | **Field name** | **Description** | **Read only** | **Mandatory** | **Control type** | **Data type** | **Length** |
| 1 | Danh sách đồ vật | Show list object | Yes | Yes | Label | String | N/A |
| 2 | No. | No. of object in table | Yes | Yes | Table | int | N/A |
| 3 | Image | Avatar of object | Yes | Yes | Table | Image | N/A |
| 4 | Tên | Name of object | Yes | Yes | Table | String | N/A |
| 5 | Chú thích | Description of object | Yes | Yes | Table | String | N/A |
| ***Table #: Web Application Interface – View Concepts - Fields*** | | | | | |  |  |

**Buttons/Hyperlinks:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No** | **Function** | **Description** | **Validation** | **Outcome** |
| 6 | Search | Search data in table | N/A | Show data that are search |
| 7 | Show entries | Show the number of entries in table | N/A | Show the number of entries that are selected |
| 8 | createNewConcept | Create new object into system. | N/A | Redirect admin to create new object page. |
| 9 | testConcept | Test object have exist in the system. | N/A | Redirect admin to test object page. |
| 10 | editConcept | Update information of object in the system. | N/A | Redirect admin to update object page. |
| 11 | deleteConcept | Remove object in the system. | N/A | Remove object that is selected out of system. |

***Table #: Web Application Interface – View Concepts – Buttons/Hyperlinks***

###### <Admin> Create Concept



**Figure xxx: Web Application Interface - <Admin> Create Concept**

**Fields:**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **No** | **Field name** | **Description** | **Read only** | **Mandatory** | **Control type** | **Data type** | **Length** |
| 1 | Tạo mới đồ vật | Create new object view | Yes | Yes | Label | String | N/A |
| 2 | Tên đồ vật | Fill in name object | No | Yes | Textbox | String | N/A |
| 3 | Mô tả đồ vật | Fill in description object | No | No | Textarea | String | N/A |
| 4 | Hình ảnh | Fill in file image | No | Yes | Img | Image | N/A |
| ***Table #: Web Application Interface – Create Concepts - Fields*** | | | | | |  |  |

**Buttons/Hyperlinks:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No** | **Function** | **Description** | **Validation** | **Outcome** |
| 5 | AddImage | Upload image | N/A | Show images that are selected |
| 6 | createConcept | Create new object | Tên đồ vật and Hình ảnh cannot be empty | Object is created into system. |
| 7 | viewAllConcept | Go back another page | N/A | Redirect view all object page. |

***Table #: Web Application Interface – Create Concepts – Buttons/Hyperlinks***

###### <Admin> Update Concept



**Figure xxx: Web Application Interface - <Admin> Update Concept**

**Fields:**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **No** | **Field name** | **Description** | **Read only** | **Mandatory** | **Control type** | **Data type** | **Length** |
| 1 | Cập Nhật Đồ Vật | Update object view | Yes | Yes | Label | String | N/A |
| 2 | Tên đồ vật | Fill in name object | No | Yes | Textbox | String | N/A |
| 3 | Mô tả đồ vật | Fill in description object | No | No | Textarea | String | N/A |
| 4 | Thêm Hình | Fill in file image | No | Yes | Img | Image | N/A |
| ***Table #: Web Application Interface – Update Concepts - Fields*** | | | | | |  |  |

**Buttons/Hyperlinks:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No** | **Function** | **Description** | **Validation** | **Outcome** |
| 5 | AddImage | Upload image | N/A | Show images that are selected |
| 6 | selectImage | Selected image in system | N/A | The image will be selected. |
| 7 | DeleteImageBySelect | Remove images that are selected | Image must be selected | Images that are selected will remove out of system. |
| 8 | updateConcept | Update object exist in the system | Tên đồ vật cannot be empty | Object is update into system. |
| 9 | viewAllConcept | Go back another page | N/A | Redirect view all object page. |

***Table #: Web Application Interface – Update Concepts – Buttons/Hyperlinks***

###### <Admin> View Logs



**Figure xxx: Web Application Interface - <Admin> View Logs**

**Fields:**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **No** | **Field name** | **Description** | **Read only** | **Mandatory** | **Control type** | **Data type** | **Length** |
| 1 | Danh Sách Vật Thể Chưa Nhận Diện Được | Show all log view | Yes | Yes | Label | String | N/A |
| 2 | No. | No. of log in table | Yes | Yes | Table | int | N/A |
| 3 | Image | Avatar of log | Yes | Yes | Table | Image | N/A |
| 4 | Ngày tạo | Create Date of log | Yes | Yes | Table | String | N/A |

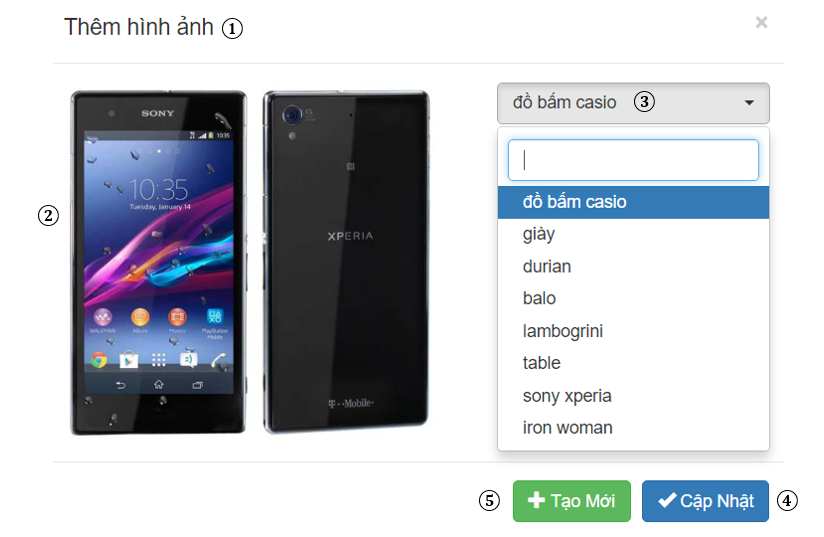
***Table #: Web Application Interface – View Logs - Fields***

**Buttons/Hyperlinks:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No** | **Function** | **Description** | **Validation** | **Outcome** |
| 5 | Search | Search data in table | N/A | Show data that are search |
| 6 | Show entries | Show the number of entries in table | N/A | Show the number of entries that are selected |
| 7 | Cập nhật log | Update information of object from log. | N/A | Show updates log view. |
| 8 | DeleteLog | Remove log in the system. | N/A | Remove log that is selected out of system. |

***Table #: Web Application Interface – View Logs – Buttons/Hyperlinks***

###### <Admin> Update Concept From Log Object



**Figure xxx: Web Application Interface - <Admin> Create/Update Concept From Log Object**

**Fields:**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **No** | **Field name** | **Description** | **Read only** | **Mandatory** | **Control type** | **Data type** | **Length** |
| 1 | Thêm hình ảnh | Update image to concept view | Yes | Yes | Label | String | N/A |
| 2 | Image | Avatar of log | Yes | Yes | Table | Image | N/A |
| 3 | Tên đồ vật | Fill in name object | No | Yes | Dropdown | String | N/A |

***Table #: Web Application Interface – Update Concept From Log Object - Fields***

**Buttons/Hyperlinks:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No** | **Function** | **Description** | **Validation** | **Outcome** |
| 4 | UpdateImagetoConcept | Update information of object from log. | Tên đồ vật cannot be empty | Object that is selected from log will be updated into system |
| 5 | CreateConcept | Create new object from log. | N/A | Redirect to create new object from log page. |

***Table #: Web Application Interface – Update Concept From Log Object – Buttons/Hyperlinks***

###### <Admin> Create New Concept From Log Object



**Figure xxx: Web Application Interface - <Admin> Create New Concept From Log Object**

**Fields:**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **No** | **Field name** | **Description** | **Read only** | **Mandatory** | **Control type** | **Data type** | **Length** |
| 1 | Tạo mới đồ vật | Create concept with image from log view | Yes | Yes | Label | String | N/A |
| 2 | Tên đồ vật | Fill in name object | No | Yes | Textbox | String | N/A |
| 3 | Mô tả đồ vật | Fill in description object | No | No | Textarea | String | N/A |
| 4 | Image | Avatar of log | Yes | Yes | Table | Image | N/A |
| ***Table #: Web Application Interface – Create Concept From Log Object - Fields*** | | | | | |  |  |

**Buttons/Hyperlinks:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No** | **Function** | **Description** | **Validation** | **Outcome** |
| 5 | AddConcept | Create new object | Tên đồ vật cannot be empty | Object is created into system. |
| 6 | viewAllLog | Go back another page | N/A | Redirect view all log page. |

***Table #: Web Application Interface – Create Concept From Log Object – Buttons/Hyperlinks***

## Database Design

### Entity relationship diagram (ERD)

D:\CN9\CapstoneProject\Report\Report4\Interaction\ERD-Diagram.png

**Figure xxx: <Admin > Entity relationship diagram (ERD)**

### Data Dictionary

## Algorithms

# System Implementation & Test

## Introduction

### Overview

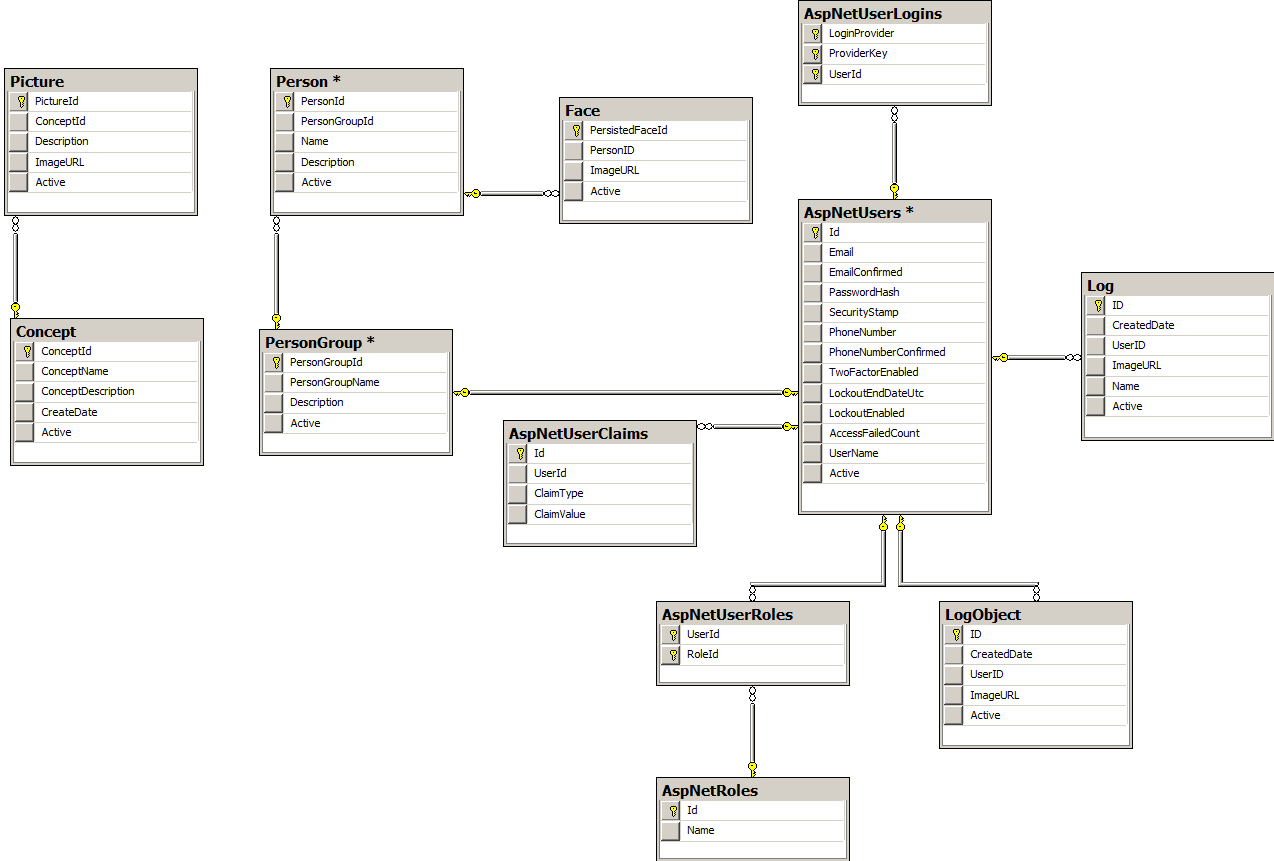
This section describes the approach and methodologies used by group to plan, organize and manage the testing of AAIV system. It provides in the detail all necessary information about the implementation and testing procedure of the system included test plans, test cases, test result, test environments, pass/fail criteria and risks estimations as well as a checklist to cover all possible cases.

### Test Approach

* Goal: Test all features in the whole AAIV system based on the core flow.
* Method: black-box testing
* Technique: check list

## Database Relationship Diagram

### Physical Diagram



### Data Dictionary

|  |  |  |
| --- | --- | --- |
| **Class Dictionary: Describe content of all table** | | |
| **Class name** | **Description** | **Mapping with ERD diagram** |
| Picture | Contain the information of picture object. | Concept |
| Concept | Contain the information of Concept. | Picture, Face |
| Person | Contain the information of Person. | PersonGroup |
| PersonGroup | Contain the information of PersonGroup. | AspNetUsers |
| Face | Contain the information of Face. | Person |
| AspNetUserClaims | Contain the information of Claims. | AspNetUsers |
| AspNetUserLogins | Contain the information of UserLogins. | AspNetUsers |
| AspNetUsers | Contain the information of all User in system. | AspNetUserClaims, AspNetUserLogins, AspNetUserRoles, LogObject, Log |
| AspNetUserRoles | Contain the informaion of user and which role of that user. | AspNetUsers |
| AspNetRoles | Contain the information of all Role in system. | AspNetUserRoles |
| LogObject | Contain the information of all Object can not detect. | AspNetUsers |
| Log | Contain the information of all Person can not detect. | AspNetUsers |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Table Name** | **Attributes** | **Description** | **Domain** | **Allow Null** |
| Picture | PictureId{PK) | Unique identifier for Picture | Varchar(32) | No |
| ConceptId{FK} | FK reference to Concept' table | Int | No |
| Description | Description of picture | Nvarchar(50) | Yes |
| ImageURL | Link of image | Nvarchar(MAX) | No |
| Active | The status of Department:  False: DE-ACTIVE  True: ACTIVE | Bit | No |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

## Performance Measures

## Test Plan

The purpose of this section is to verify and ensure that SCP meets its design specification and other requirements from user. The following part will describe which features to be tested and which will not.

## System Testing Test Case

### Communication Diagram

D:\CN9\CapstoneProject\Report\Report4\Testing\Communication.png

**Figure xx: Communication Diagram**

### Test Cases

#### Mobile Application Test Cases

##### <User> Detect Person

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **ID** | **Test Case Description** | **Test case procedure** | **Expected output** | **Result** | **Test Date** |
| UDP01 | Detect person was trained | 1. Login successfully with role “User” in mobile application.  2. Say detect command.  3. Take picture. | Mobile application show the name of the person was trained. | Pass:  Fail: |  |
| UDP02 | Detect person was not trained. | 1. Login successfully with role “User” in mobile application.  2. Send detect command.  3. Take picture. | Mobile application show the gender of the person. | Pass:  Fail: |  |
| UDP03 | Detect person while the face inclined 45 degrees to the camera. | 1. Login successfully with role “User” in mobile application.  2. Send detect command.  3. Take picture. | Mobile application show the name of the person. | Pass:  Fail: |  |
| UDP04 | Detect person while the face inclined 30 degrees to the camera. | 1. Login successfully with role “User” in mobile application.  2. Send detect command.  3. Take picture. | Mobile application show the name of the person. | Pass:  Fail: |  |
| UDP05 | Detect person while the face inclined 20 degrees to the camera. | 1. Login successfully with role “User” in mobile application.  2. Send detect command.  3. Take picture. | Mobile application show the name of the person. | Pass:  Fail: |  |
| UDP06 | Detect person while have noone in picture. | 1. Login successfully with role “User” in mobile application.  2. Send detect command.  3. Take picture. | Mobile application show do not have any person message. | Pass:  Fail: |  |
| UDP07 | Detect person while the person wear glasses. | 1. Login successfully with role “User” in mobile application.  2. Send detect command.  3. Take picture. | Mobile application show the name of the person. | Pass:  Fail: |  |
| UDP08 | Detect person from behind. | 1. Login successfully with role “User” in mobile application.  2. Send detect command.  3. Take picture. | Mobile application show the name of the person. | Pass:  Fail: |  |
| UDP09 | Detect person while picture contrain ½ face. | 1. Login successfully with role “User” in mobile application.  2. Send detect command.  3. Take picture. | Mobile application show the name of the person. | Pass:  Fail: |  |
| UDP09 | Detect person while picture contrain ½ face. | 1. Login successfully with role “User” in mobile application.  2. Send detect command.  3. Take picture. | Mobile application show the name of the person. | Pass:  Fail: |  |
| UDP10 | Detect person while picture contrain ½ face. | 1. Login successfully with role “User” in mobile application.  2. Send detect command.  3. Take picture. | Mobile application show the name of the person. | Pass:  Fail: |  |
| UDP11 | Detect person from a distance of 1 meter. | 1. Login successfully with role “User” in mobile application.  2. Send detect command.  3. Take picture. | Mobile application show the name of the person. | Pass:  Fail: |  |
| UDP12 | Detect person from a distance of 1.5 meter. | 1. Login successfully with role “User” in mobile application.  2. Send detect command.  3. Take picture. | Mobile application show the name of the person. | Pass:  Fail: |  |
| UDP13 | Detect person from a distance of 2 meter. | 1. Login successfully with role “User” in mobile application.  2. Send detect command.  3. Take picture. | Mobile application show the name of the person. | Pass:  Fail: |  |
| UDP14 | Detect person from a distance of 2.5 meter. | 1. Login successfully with role “User” in mobile application.  2. Send detect command.  3. Take picture. | Mobile application show the name of the person. | Pass:  Fail: |  |
| UDP15 | Detect person from a distance of 3 meter. | 1. Login successfully with role “User” in mobile application.  2. Send detect command.  3. Take picture. | Mobile application show the name of the person. | Pass:  Fail: |  |
| UDP16 | Take picture contain 2 person(1 trained and 1 not trained) | 1. Login successfully with role “User” in mobile application.  2. Send detect command.  3. Take picture. | Mobile application show the name of the person was trained. | Pass:  Fail: |  |
| UDP17 | Take picture contain 2 person was trained. | 1. Login successfully with role “User” in mobile application.  2. Send detect command.  3. Take picture. | Mobile application show the name of 2 person. | Pass:  Fail: |  |
| UDP18 | Take picture contain 3 person was trained. | 1. Login successfully with role “User” in mobile application.  2. Send detect command.  3. Take picture. | Mobile application show the name of 3 person. | Pass:  Fail: |  |
| UDP19 | Take picture contain 4 person was trained. | 1. Login successfully with role “User” in mobile application.  2. Send detect command.  3. Take picture. | Mobile application show the name of 4 person. | Pass:  Fail: |  |

##### <User> Detect Object

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **ID** | **Test Case Description** | **Test case procedure** | **Expected output** | **Result** | **Test Date** |
| UDO01 | Detect object was trained. | 1. Login successfully with role “User”.  2. Send detect object command. | Mobile application show the name of the object. | Pass:  Fail: |  |
| UDO02 | Detect object was not trained. | 1. Login successfully with role “User”.  2. Send detect object command. | Mobile application show the name of the object. | Pass:  Fail: |  |
| UDO03 | Detect object from a distance of 1 meter. | 1. Login successfully with role “User”.  2. Send detect object command. | Mobile application show the name of the object. | Pass:  Fail: |  |
| UDO04 | Detect object from a distance of 1.5 meter. | 1. Login successfully with role “User”.  2. Send detect object command. | Mobile application show the name of the object. | Pass:  Fail: |  |
| UDO05 | Detect object from a distance of 2 meter. | 1. Login successfully with role “User”.  2. Send detect object command. | Mobile application show the name of the object. | Pass:  Fail: |  |
| UDO06 | Detect object from a distance of 2.5 meter. | 1. Login successfully with role “User”.  2. Send detect object command. | Mobile application show the name of the object. | Pass:  Fail: |  |
| UDO07 | Detect object from a distance of 3 meter. | 1. Login successfully with role “User”.  2. Send detect object command. | Mobile application show the name of the object. | Pass:  Fail: |  |
| UDO08 | Take picture contain 2 object(1 trained and 1 not trained). | 1. Login successfully with role “User”.  2. Send detect object command. | Mobile application show the name of the object was trained. | Pass:  Fail: |  |
| UDO09 | Take picture contrain 2 object was trained. | 1. Login successfully with role “User”.  2. Send detect object command. | Mobile application show the name of 2 object. | Pass:  Fail: |  |
| UDO10 | Take picture contrain 3 object (1 trained and 2 not trained). | 1. Login successfully with role “User”.  2. Send detect object command. | Mobile application show the name of the object was trained. | Pass:  Fail: |  |

##### <User> Detect Person Streaming

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **ID** | **Test Case Description** | **Test case procedure** | **Expected output** | **Result** | **Test Date** |
| UDPS01 | Streaming contain no face in 30 seconds. | 1. Login successfully with role “User”.  2. Send streaming command. | Mobile application not take picture or detect. | Pass:  Fail: |  |

##### <User> Add Person

##### <User> Delete Person

##### <User> Add Person From Log

##### <User> Delete Logs

#### Web Application Test Cases

##### <Admin>View List Account

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **ID** | **Test Case Description** | **Test case procedure** | **Expected output** | **Result** | **Test Date** |
| AVLA01 | View list accounts | 1. Login successfully with role “Admin”.  2. Click “Quản lý tài khoản” button | Web application shows list accounts in “Quản lý tài khoản” view. |  |  |

**Table xxx: Web Application Test Case - <Admin> View List Accounts**

##### <Admin> Activate/Deactivate Account

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **ID** | **Test Case Description** | **Test case procedure** | **Expected output** | **Result** | **Test Date** |
| AADA01 | Activate specific account success | 1. Login successfully with role “Admin”.  2. In “Quản lý tài khoản” view, select a account in list and click “Activate” button.  3. Click “Cập nhậ” button. | Web application shows “Cập nhật thành công” message. |  |  |
| AADA02 | Activate specific account but cancels | 1. Login successfully with role “Admin”.  2. In “Quản lý tài khoản” view, select a deactivated account in list and click “Deactivate” button.  3. Refresh page without click “Cập nhật” button. | Web application does nothing, account still locked. |  |  |
| AADA03 | Deactivate specific account success | 1. Login successfully with role “Admin”.  2. In “Quản lý tài khoản” view, select a activated account in list and click “Deactivate” button.  3. Click “Cập nhật” button when it show up. | Web application shows pop up with message: “Khóa tài khoản thành công”. |  |  |

**Table xxx: Web Application Test Case - <Admin> Activate/Deactivate Account**

##### <Admin> Add New Concept

##### <Admin> Delete Concept

##### <User> Add New Person

##### <User> Update Person

##### <User> Delete Person

##### <User> Add New Person From Log

##### <User> Delete Log Person

##### <User> Add New Object From Log

##### <User> Delete Log Object

# Software User’s Manual

## Installation Guide

## User Guide