



# 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(Dropped out)
Supervisor	Mr. Kiều Trọng Khánh
Ext. Supervisor	N/A
Capstone Project code	AAIV



## CAPSTONE PROJECT REGISTER

Class:                      Duration time: from 02/01/2017.... To                      /2017.....

(\*) Profession: <Software Engineer>                      Specialty:    <ES> ☐                      <IS> ☒

(\*) Kinds of person make registers:                      Lecturer ☒                      Students ☐

### 1. Register information for supervisor (if have)

	Full name	Phone	E-Mail	Title
Supervisor 1	Kiều Trọng Khánh		khanhkt@fpt.edu.vn	Mr.

### 2. Register information for students (if have)

	Full name	Student code	Phone	E-mail	Role in Group
Student 1	Phan Trung Thành	SE61288	01689129170	thanhptse61288@fpt.edu.vn	Team Leader
Student 2	Nguyễn Vũ Hoàng Sơn	SE61490	0916398819	sonnvhse61490@fpt.edu.vn	Team Member
Student 3	Võ Hà Quân	SE61254	01234979545	quanhvse61254@fpt.edu.vn	Team Member
Student 4	Nguyễn Cao Duy	SE61032	0976666375	duyncse61032@fpt.edu.vn	Team Member

### 3. Register content of Capstone Project

(\*) 3.1. Capstone Project name:

English: Automatic Alternative Image Recognition to Voice

Vietnamese: Nhận dạng hình ảnh hỗ trợ người mù giao tiếp

Abbreviation: AAIV

## **- Context:**

- + Nowadays, the smartphone is very popular. It supports controlling using voice. Some phones allow users to setup the shortcut keys for their need
- + The Facebook provides useful function supporting the blind feeling the pictures
- + How can the blind know whether the person that is talking face to face with him/her is his/her friend or not? How to the blind can guess his/her age, feeling ... to communicate effectively?
- + How to the blind can know what are the things that he/she is seeing?
- + Microsoft Cognitive Services let you build apps with powerful algorithms using just a few lines of code. They work across devices and platforms such as iOS, Android, and Windows, keep improving and are easy to set up. This API supports some features as vision (From faces to feelings, these APIs allow your apps to understand image and video content), speech (Hear and speak to your users with APIs that filter noise, identify speakers, and more), language (APIs that allow your app to process natural language and learn how to recognize what users want), Knowledge (Tap into rich knowledge amassed from the web, academia, or your own data), and Search (Make your apps more intelligent with the power of Bing APIs, A single call accesses data from billions of web pages, images, videos, and news)  
(<https://www.microsoft.com/cognitive-services> )

## **- Building the application provides following services**

- + Combine smartphone devices and Microsoft Cognitive Services, ...
- + Building the application that uses voice to describe the person who is talking face to face with the blind (whether the person is their friend or not? What is the person's age, gender, feeling...?). The application will run when the blind hears the familiar voice, when he/she take a picture with shortcut key when he/she uses voice control. (The condition is required as people in face to face)
- + The application can also detect anything that had been trained
- + The application allows a regular man to input training data to support the blind.
- + The blind users can use voice control or shortcut keys to interact with the application.
- + ...

## **- Simulator**

- + The man trains the blind's face friend and some fruits
- + The blind uses the application to capture a picture of who are talking face to face to them. The application will tell the blind if there are his/her friend or not. The application also tells some information about the recognized man/woman (such as gender, age, feeling...).
- + The blind also captures the picture of some fruits. The system says the name of known fruits and says "Unknown" if it does not know the fruit. The blind can save that pictures as data for later training.

(\*) 3.2. Main proposal content (including result and product)

a) Theory and practice (document):

- ..Student should apply the software development process and the UML
- ..Software artifacts include User Requirement, Software Requirement Specification, Architecture Design, Detail Design, System Implementation and Testing Document, Installation Guide, sources code, and deployable software packages
- ..3 tiers should be applied
- ..Server side technique:
  - Database design, OOA, OOD, OOP, MVC, Java or .Net technology, Restful API, ...
- ..Client side technique
  - HTML5, CSS, JavaScript, JQuery, Ajax, Android, iOS...
- ..Communication technique
  - Exchange information and transfer data in effective in networks, communicating protocol between mobile devices, ...
- ..Research
  - Algorithms
  - Microsoft Cognitive Services
  - ...

b) Program:

-..Main functions

- The application can allow the user training image
- Web Application for users
- Mobile Application for blind and users
- The application can allow the user searching with image and speaking the voice, ...
- ...

c) Other products:

- ..All of the management functions of the system must be implemented to support the operating system in best

4. Other comment (propose all relative thing if have)

N/A

HCM city, date 14/12/2016

**Supervisor (If have)**

*(Sign and full name)*

**On behalf of Registers**

*(Sign and full name)*

## Table of Contents

<b>A. Introduction.....</b>	<b>1</b>
<b>1. Project Information.....</b>	<b>1</b>
<b>2. Introduction.....</b>	<b>1</b>
<b>3. Current Situation.....</b>	<b>1</b>
<b>4. Problem Definition.....</b>	<b>1</b>
<b>5. Proposed Solution.....</b>	<b>2</b>
5.1 Feature functions.....	2
5.2 Advantages and disadvantages.....	2
<b>6. Functional Requirements.....</b>	<b>3</b>
<b>7. Role and Responsibility.....</b>	<b>3</b>
<b>B. Software Project Management Plan.....</b>	<b>3</b>
<b>1. Problem Definition.....</b>	<b>3</b>
1.1 Name of this Capstone Project.....	3
1.2 Problem Abstract.....	3
1.3 Project Overview.....	4
<b>2. Project organization.....</b>	<b>6</b>
2.1 Software Process Model.....	6
2.2 Roles and responsibilities.....	6
2.3 Tools and Techniques.....	7
<b>3. Project Management Plan.....</b>	<b>7</b>
3.1 Product Backlog.....	7
3.2 Sprint Backlog.....	7
<b>C. Software Requirement Specification.....</b>	<b>8</b>
<b>1. User Requirement Specification.....</b>	<b>8</b>
1.1 Guest Requirement.....	8
1.2 User Requirement.....	8
1.3 Admin Requirement.....	8
<b>2. System Requirement Specification.....</b>	<b>8</b>
2.1 External Interface Requirement.....	8
2.2 System Overview Use Case.....	9
2.3 List of Use Case.....	11
<b>3. Software System Attribute.....</b>	<b>26</b>
3.1 Usability.....	26
3.2 Reliability.....	27
3.3 Availability.....	27
3.4 Security.....	27
3.5 Maintainability.....	27
3.6 Portability.....	27
3.7 Performance.....	27
<b>4. Conceptual Diagram.....</b>	<b>27</b>
<b>D. Software Design Description.....</b>	<b>28</b>
<b>1. System Architectural Design.....</b>	<b>28</b>
1.1 Mobile Application Architecture Description.....	28
1.2 Web Service Architecture Description.....	29
1.3 Web Application Architecture Description.....	30
<b>2. Component Diagram.....</b>	<b>31</b>
<b>3. Detailed Description.....</b>	<b>32</b>
3.1 Class Diagram.....	32

3.2 Class Diagram Explanation.....	33
3.3 Interaction Diagram .....	33
<b>4. Database Design.....</b>	<b>40</b>
4.1 Entity relationship diagram (ERD) .....	40
4.2 Data Dictionary.....	40
<b>5. Algorithms .....</b>	<b>40</b>
5.1 Voice Recognition Algorithms.....	40
5.2 Training Person Algorithms.....	42
5.3 Detect Person Algorithms .....	43
5.4 Person Group For Each User.....	44
5.5 Find Duplicated Person By Faces Algorithms .....	45
5.6 Find Duplicated Person By Name Algorithms.....	47
5.7 Training Object Algorithms.....	48
5.8 Detect Object Algorithms .....	49

## A. Introduction

### 1. 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

### 2. 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.

### 3. 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

### 4. 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.

## **5. Proposed Solution**

### **5.1 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.

### **5.2 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 helps them communicate effectively.
- Disadvantages:
  - Sometimes the detection does not work accurately.



## 6. 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 the 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 the 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.

## 7. 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	quanhse61254@fpt.edu.vn
5	Nguyễn Cao Duy	Developer	Member	duyncse61032@fpt.edu.vn

*Table 1: Role and Responsibility*

## B. Software Project Management Plan

### 1. Problem Definition

#### 1.1 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.

#### 1.2 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.

### 1.3 Project Overview

#### 1.3.1 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.

#### 1.3.2 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 the 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).

### **1.3.3 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.

### **1.3.4 Future plans**

- Improve voice control: Recognize only user voice.
- Available for multiple devices: Smart Glasses, Portable Device, Smaller devices with camera.
- Support multiple platforms: iOS, Window phone.
- Better performance.

### **1.3.5 Development Environment**

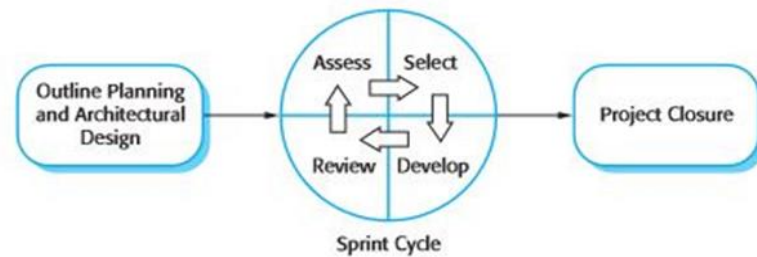
*(References to main document, Section 2.1.3.5)*

## 2. Project organization

### 2.1 Software Process Model

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

- 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 1: Scrum Development Model*

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

For more information:

[https://en.wikipedia.org/wiki/Scrum\\_\(software\\_development\)](https://en.wikipedia.org/wiki/Scrum_(software_development))

### 2.2 Roles and responsibilities

No	Full name	Role in Group	Responsibilities
1	Kiều Trọng Khánh	Supervisor	<ul style="list-style-type: none"> <li>- Specify user requirement</li> <li>- Control the development process</li> <li>- Give out technique and business analysis support</li> </ul>
2	Phan Trung Thành	Team leader, DEV, Tester	<ul style="list-style-type: none"> <li>- Managing process</li> <li>- Designing database</li> <li>- Clarifying requirements</li> <li>- Prepare documents</li> <li>- Create test plan</li> <li>- Coding</li> <li>- Testing</li> </ul>
3	Nguyễn Vũ Hoàng Sơn	Team member, DEV, Tester	<ul style="list-style-type: none"> <li>- Designing database</li> <li>- Clarifying requirements</li> <li>- Prepare documents</li> <li>- GUI design</li> <li>- Create test plan</li> <li>- Coding</li> <li>- Test</li> </ul>

No	Full name	Role in Group	Responsibilities
4	Võ Hà Quân	Team member, DEV, Tester	<ul style="list-style-type: none"> <li>- Designing database</li> <li>- Clarifying requirements</li> <li>- Prepare documents</li> <li>- Create test plan</li> <li>- Coding</li> <li>- Test</li> </ul>
5	Nguyễn Cao Duy	Team member, DEV, Tester	<ul style="list-style-type: none"> <li>- Designing database</li> <li>- Clarifying requirements</li> <li>- Prepare documents</li> <li>- Create test plan</li> <li>- Coding</li> <li>- Test</li> </ul>

*Table 5: Roles and Responsibilities Details*

### 2.3 Tools and Techniques

	Tools	Techniques
<b>Front-end</b>	Visual Studio	<ul style="list-style-type: none"> <li>- HTML5</li> <li>- CSS3</li> <li>- JavaScript</li> <li>- jQuery</li> <li>- Ajax</li> </ul>
<b>Back-end</b>	Visual Studio, Android Studio, Sublime Text 2	<ul style="list-style-type: none"> <li>- .Net framework + ASP.NET MVC4 + Entity Framework 5</li> <li>- Flask framework(Python)</li> </ul>
<b>Web server</b>	IIS version 1607	- C# .NET
<b>Mobile application</b>	Android Studio	<ul style="list-style-type: none"> <li>- Version 2.2.3</li> <li>- Java 7</li> <li>- SDK 23</li> </ul>
<b>Database management system</b>	MS SQL Server 2008	N/A

*Table 6: Tools and Techniques*

## 3. Project Management Plan

### 3.1 Product Backlog

*(References to main document, Section 2.3.1)*

### 3.2 Sprint Backlog

*(References to main document, Section 2.3.2)*

## C. Software Requirement Specification

### 1. User Requirement Specification

#### 1.1 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

#### 1.2 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

#### 1.3 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

## 2. System Requirement Specification

### 2.1 External Interface Requirement

#### 2.1.1 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

#### 2.1.2 Hardware Interface

- Android Smartphone: Android 6.0.1 Marshmallow or later.

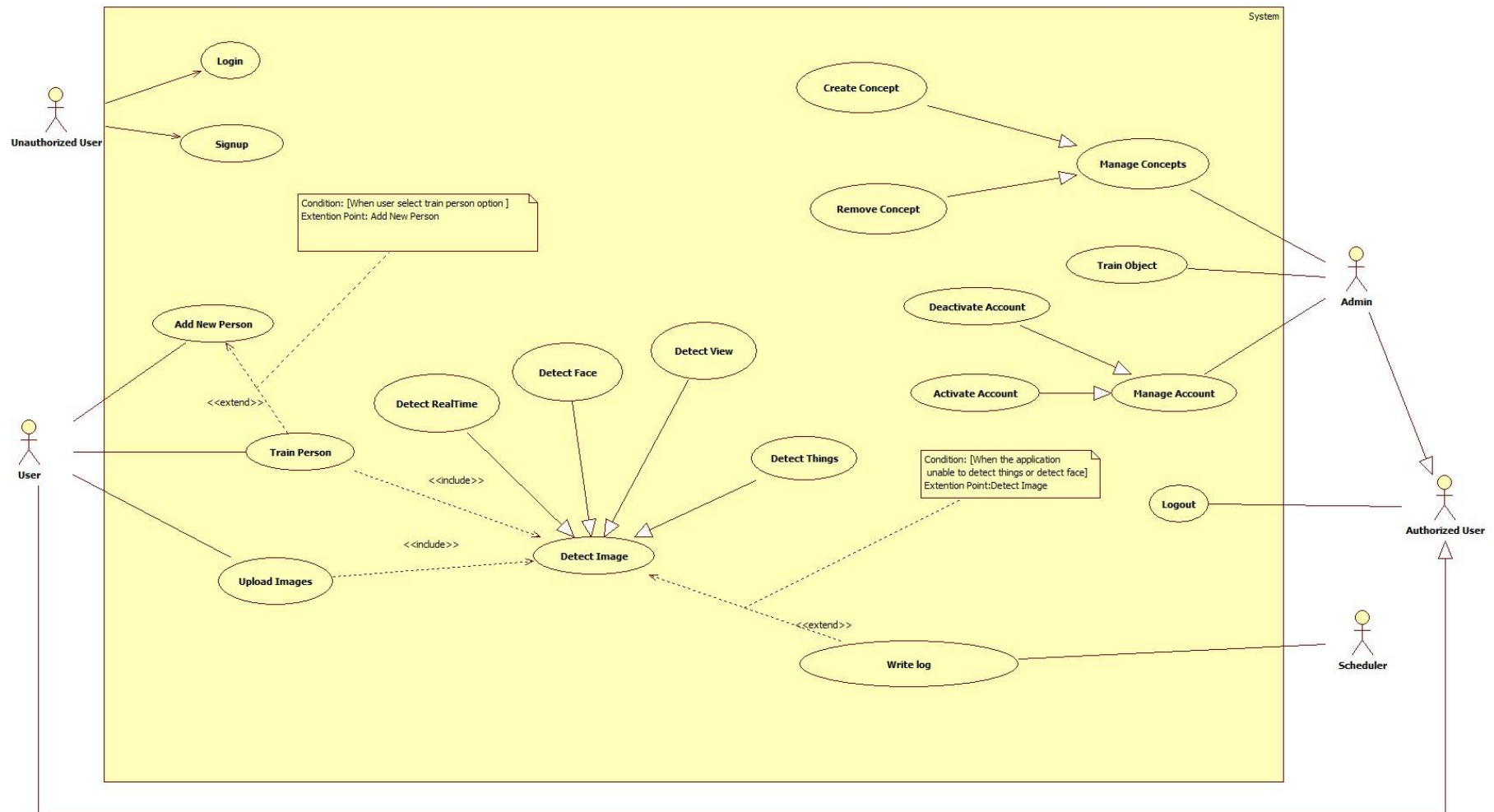
#### 2.1.3 Software Interface

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

#### **2.1.4 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.

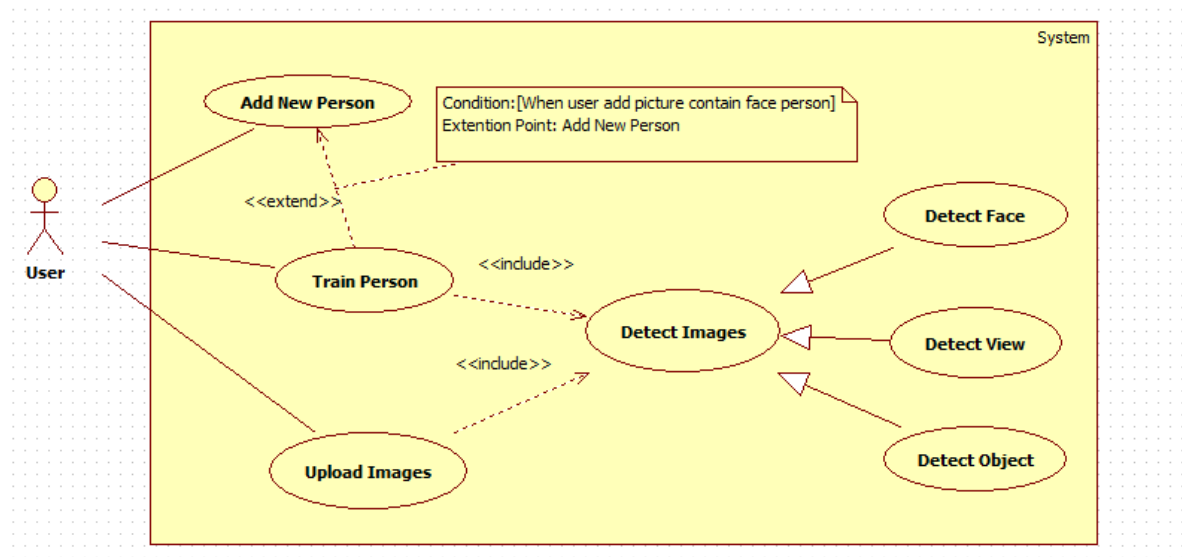
#### **2.2 System Overview Use Case**





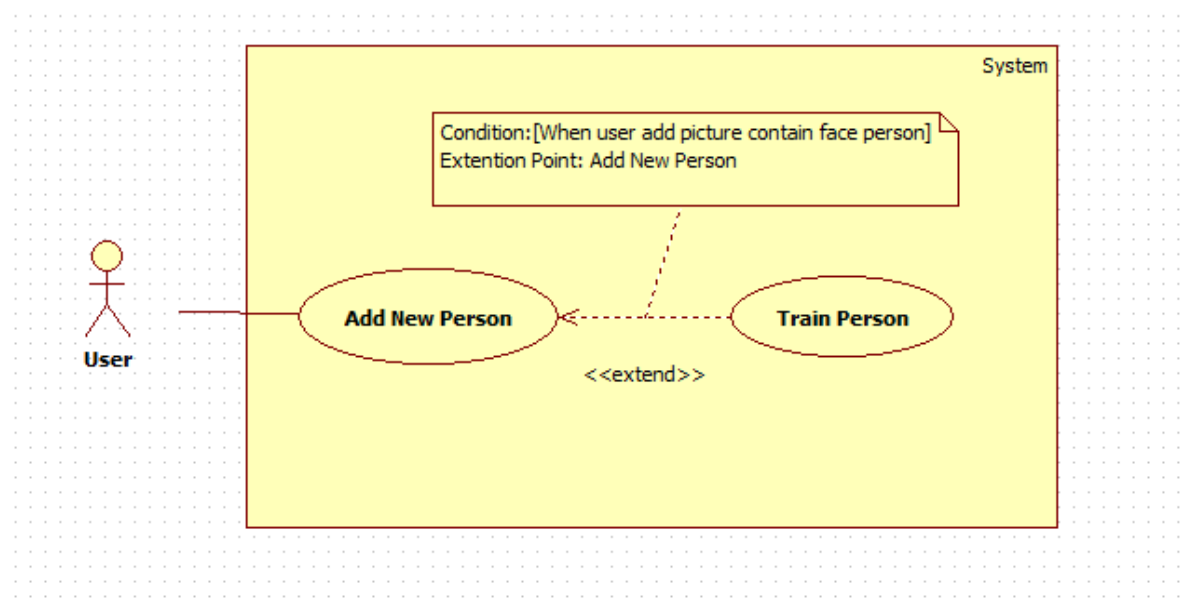
## 2.3 List of Use Case

### 2.3.1 <User> Overview Use Case



*Figure 3: User Overview Use Case*

#### 2.3.1.1 <User> Add New Peron



*Figure 4: <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	04/15/2017	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: <ul style="list-style-type: none"> <li>- Name free text input. Required length is 3 - 128.</li> <li>- User Description: free text input. The size limit is 128 characters.</li> <li>- Image: hidden free text input. Valid image size is from 1KB to 4MB.</li> </ul>
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:**

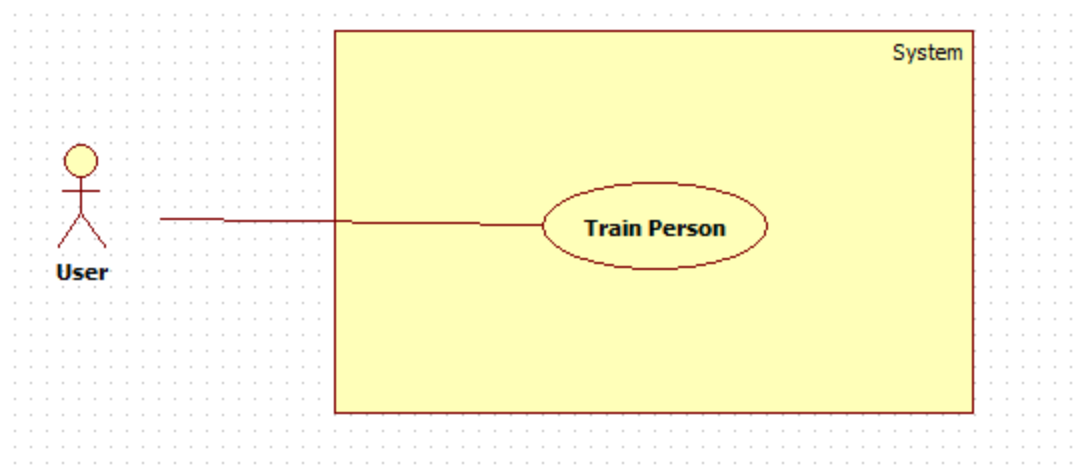
No.	Cause	System Response
1	Name is empty.	System responses: “Name không được bỏ trống”.
2	Length of Name is not between 3 - 128 characters.	System responses: “Độ dài của Name trong khoảng 3 - 128 kí tự”.
3	Image size exceeded the limit size (4Mb)	System automatically compresses the image but remains 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 created person information will be stored at both system Database and Microsoft Cognitive Service.
- The Person ID must be consistent in both system Database and Microsoft Cognitive Service.
- The person group contains the newly created person must be trained after successfully created a new Person.

*Table 15: USE CASE – UC\_AS01 - <User> Add New Person***2.3.1.2 <User> Train Person***Figure 6: <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	04/15/2017	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 information from user: <ul style="list-style-type: none"> <li>- personGroupId: hidden free text input. Required. Length between 3 – 64 characters.</li> </ul>
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]

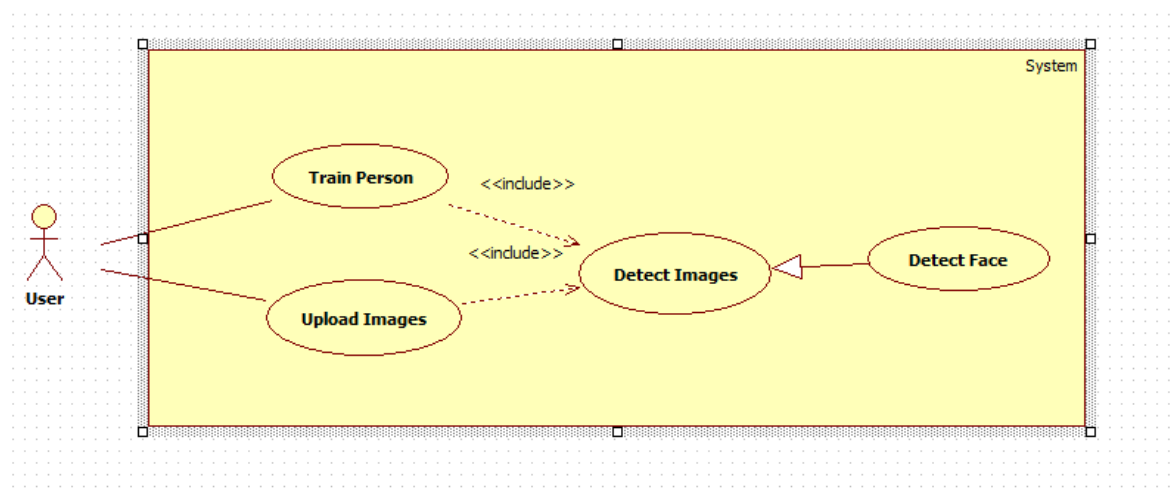
**Alternative Scenario:***[Alternative 1]*

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]

<i>[Alternative 2]</i>		
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]
<b>Exceptions:</b>		
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”.
<b>Relationships:</b> N/A.		
<b>Business Rules:</b>		
<ul style="list-style-type: none"> <li>- Person Group Id is gotten from logged in user.</li> <li>- Image will be compressed when upload to the system, less than 300kb.</li> <li>- User can choose to crop out the face in their picture.</li> <li>- If training process is fail, return training status to user.</li> </ul>		

*Table 17: USE CASE – UC\_US03 - <User> Train Person*

### 2.3.1.3 <User> Detect Face



*Figure 7: <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	04/15/2017	Priority	High
<b>Actor:</b> <ul style="list-style-type: none"><li>- User</li></ul>			
<b>Summary:</b> <ul style="list-style-type: none"><li>- This use case 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.</li></ul>			
<b>Goal:</b> <ul style="list-style-type: none"><li>- User can knows who the person in front of them is.</li></ul>			
<b>Triggers:</b> <ul style="list-style-type: none"><li>- User sends command to detect a person.</li></ul>			
<b>Preconditions:</b> <ul style="list-style-type: none"><li>- Actor must login at user role.</li><li>- Person Group training process is finished.</li></ul>			
<b>Post Conditions:</b> <ul style="list-style-type: none"><li>- <b>Success:</b> The person is identified. His / her information is noticed to the user through voice notification.</li><li>- <b>Fail:</b> System cannot identify the person. Notify user through voice notification.</li></ul>			
<b>Main Success Scenario:</b>			
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 inputted 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]	
<b>Alternative Scenario:</b>			
[Alternative 1]			
Step.	Actor Action	System Response	
1	No face is detected in the image	System response: cannot detect the person. Notify result to user through voice notification.	
[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 notification.
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:

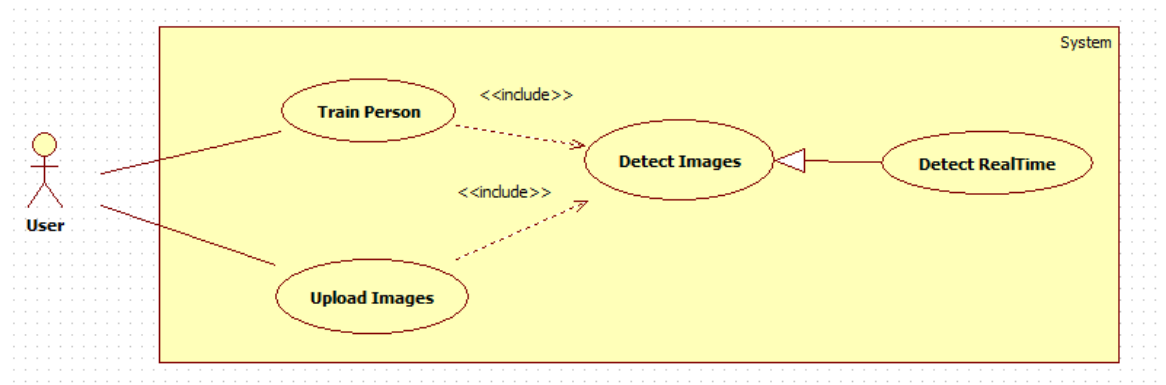
ID	USER_ID	IMAGE_URL	IMAGE_TYPE	DATETIME	ACTIVE

- The detecting face process:
  - 1) Face Detection:
    - Microsoft Cognitive Service receives image url, turn it into black and white image.
    - Using algorithms, scanning 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 Process 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 18: USE CASE – UC\_US04 - <User> Detect Face*

#### 2.3.1.4 <User> Detect Real-time



*Figure 10: <User> Detect Real-time (UC\_US07)*

USE CASE – UC_US07			
Use Case No.	UC_US07	Use Case Version	1.0
Use Case Name	Detect Real-time		
Author	SonNVH		
Date	04/15/2017	Priority	High
<b>Actor:</b> <ul style="list-style-type: none"> <li>- User</li> </ul>			
<b>Summary:</b> <ul style="list-style-type: none"> <li>- This use case allows user to detect person continuously in real time.</li> </ul>			
<b>Goal:</b> <ul style="list-style-type: none"> <li>- The application can automatically detect person without the user command to capture picture.</li> </ul>			
<b>Triggers:</b>			



- User sends command to begin streaming detecting process.

**Preconditions:**

- Actor must login at user role.

**Post Conditions:**

- **Success:** The application automatically identify any detected person.
- **Fail:** Show response message.

**Main Success Scenario:**

Step	Actor Action	System Response
1	User sends command to begin streaming detecting process. [Alternative 1]	System open device camera.
2		System notify begin the streaming detecting process by voice.
3	The device detected a human face.	System automatically capture the human face and begin identifying process.
4		System notify the person information to the user by voice. [Alternative 2]

**Alternative Scenario:***[Alternative 1]*

Step.	Actor Action	System Response
1	User send command to quit streaming detecting process	System close camera, back to main screen.

*[Alternative 2]*

Step.	Actor Action	System Response
1	A person is detected but cannot be identified.	System continue the streaming process.

**Exceptions:**

No.	Cause	System Response
1	Invalid subscription Key or user/plan is blocked.	System returns response message.
2	Out of call volume quota	System returns response message.
3	Rate limit is exceeded	System returns response message.

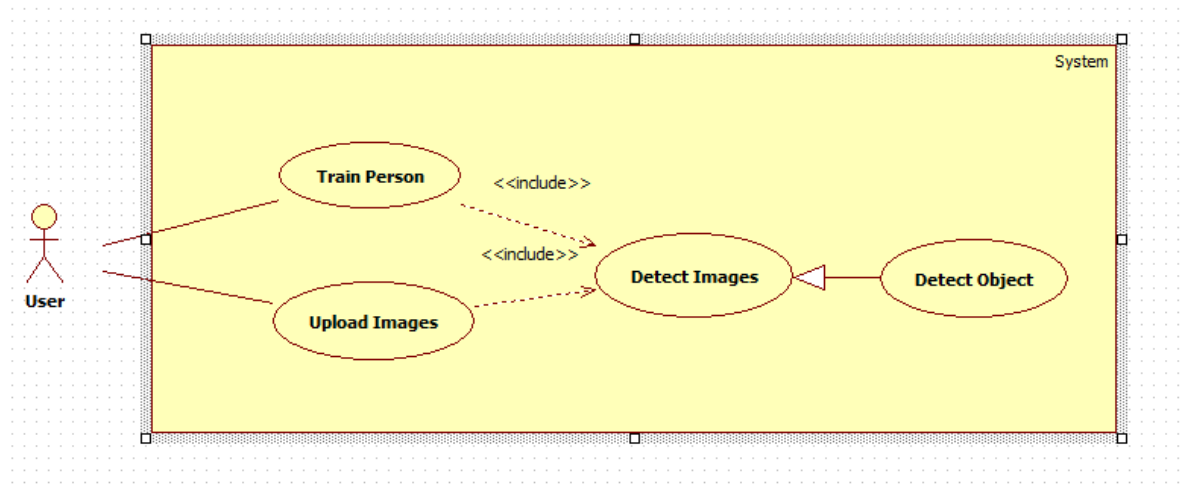
**Relationships:** N/A.**Business Rules:**

- The application must notifies user the beginning of the streaming process through voice.
- The application will continuously detect for human faces in real time.
- When a human faces is detected, the application captures a picture of the human face and begin the identifying process.
- When a human face is detected, the application has 3 seconds to identify the person.
- After 3 seconds, the application continues to detect human faces automatically in real time.

- After 3 seconds, if the human face is identified successfully, the application will notify the person information to user by voice.
- If the human face cannot be identified, the application will not notify any response message.

*Table 21: USE CASE – UC\_US07 - <User> Detect Real-time*

### 2.3.1.5 <User> Detect Object



*Figure 9: <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	04/15/2017	Priority	High
<b>Actor:</b> <ul style="list-style-type: none"> <li>- User</li> </ul>			
<b>Summary:</b> <ul style="list-style-type: none"> <li>- This use case allows user to identify what object is in front of them by captured image.</li> </ul>			
<b>Goal:</b> <ul style="list-style-type: none"> <li>- System can describes what the object is to user through voice notification.</li> </ul>			
<b>Triggers:</b> <ul style="list-style-type: none"> <li>- User sends command to detect an object.</li> </ul>			
<b>Preconditions:</b> <ul style="list-style-type: none"> <li>- Actor must login at user role.</li> </ul>			
<b>Post Conditions:</b> <ul style="list-style-type: none"> <li>- <b>Success:</b> System describes the object back to the user through voice notification.</li> <li>- <b>Fail:</b> System cannot describes the object. Notify result to user through voice notification.</li> </ul>			

**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 inputted 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 notification.
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:**

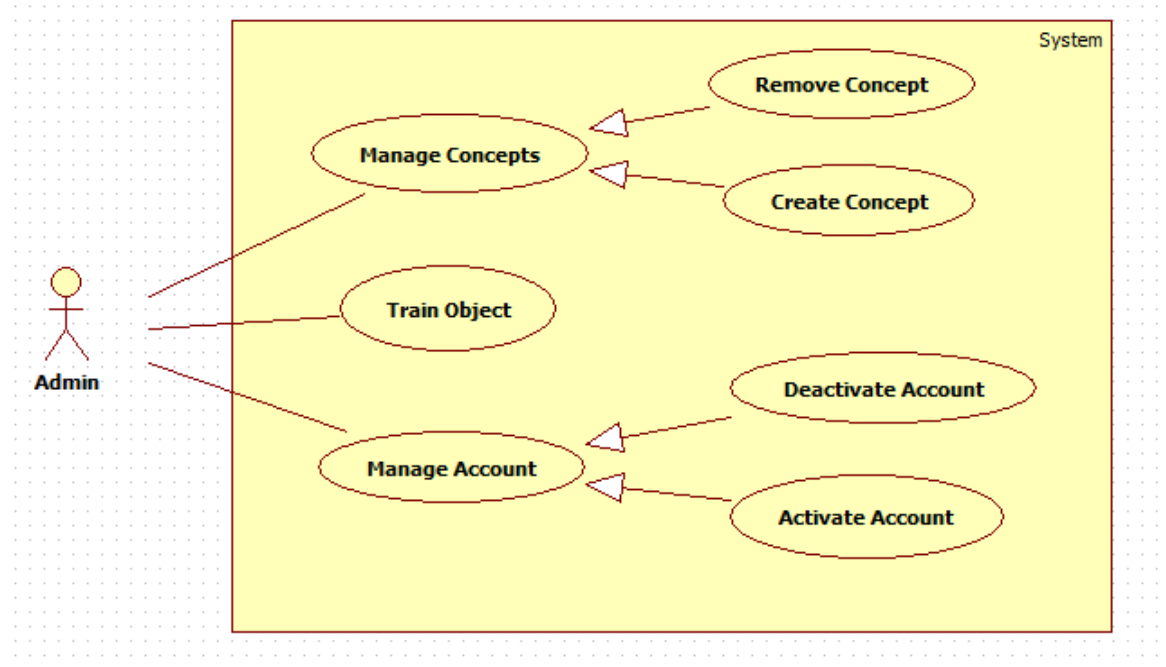
- Image will be compressed when uploaded to the system. The compressed image file size is between 20Kb – 70Kb.
- Identification result will be noticed to the user through voice notification.
- If the application cannot identify the object, a Log file will be created in the system and system Admin will be noticed.
- Log file structure in database:

ID	CREATED_DATE	IMAGE_URL	USER_ID	ACTIVE

- **ID:** Log File ID: integer
- **User\_ID:** Id of authorized user that requested the identification of object: String
- **Image\_URL:** URL of the object image: String
- **Created\_Date:** Date and Time when the log file is created: Datetime dd/MM/yyyy hh:mm:ss
- **Active:** activation state of the log file: Bit

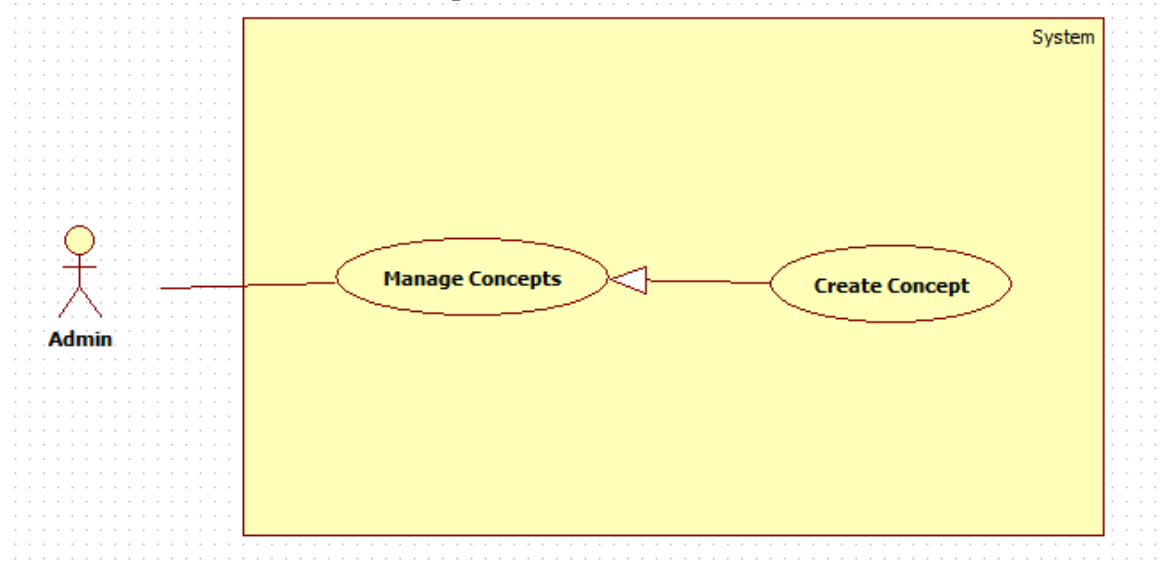
*Table 20: USE CASE – UC\_US06 - <User> Detect Object*

### 2.3.2 <Admin> Overview Use Case



*Figure 14: <Admin> Overview Use Case*

#### 2.3.2.1 <Admin> Create Concept



*Figure 18: <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
<b>Actor:</b> - Admin			
<b>Summary:</b>			

- This use case allows admin create new concept to the system predict object and can recognize object.

**Goal:**

- A new concept is created into the system.

**Triggers:**

- Admin sends create new concept command.

**Preconditions:**

- Account must be login with admin role.

**Post Conditions:**

- **Success:** System will send success message when concept is created successfully.
- **Fail:** System will show error message.

**Main Success Scenario:**

Step	Actor Action	System Response
1	Admin goes to create new concept view.	System require information from actor: <ul style="list-style-type: none"> <li>- Concept Name: free text input, required. The size limit is 100 characters.</li> <li>- Description: free text input, required. The size limit is 100 characters.</li> <li>- Images: image type input, required.</li> </ul>
3	Admin inputs information.	
3	Admin sends command to create new concept.	System validates inputted information. [Exception 1,2,3,4]
4		System create new concept and redirects to view all concept view.

**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”.
s2	Images are empty.	System shows error message: “Vui lòng chọn hình ảnh”.
3	Invalid image.	System shows error message: “Vui lòng chỉ chọn hình”.

4	Description is empty	System shows error message: “Mô tả không được bỏ trống”
---	----------------------	---

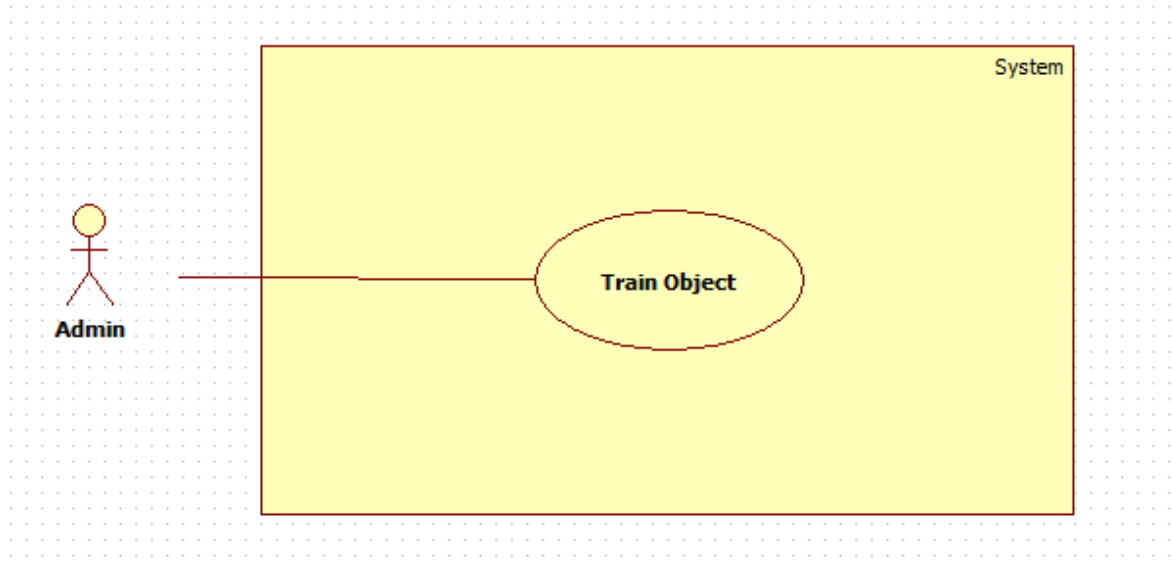
**Relationships:** N/A.

**Business Rules:**

- System must validate user inputted information before creating a Concept.
- Concept ID must be unique.
- Each concept must have at least one image of it (input).
- If user does not input Concept Name, the Concept ID will be used as Concept Name by default.
- System will redirect to all concept list view after successfully creating a Concept.
- Support image formats are: JPG, PNG, JPEG.

*Table 27: USE CASE – UC\_AD04 - <Admin> Create Concept*

### 2.3.2.2 <Admin> Train Object



*Figure 19: <Admin> Train Object (UC\_AD05)*

USE CASE – UC_AD05			
<b>Use Case No.</b>	UC_ AD05	<b>Use Case Version</b>	1.0
<b>Use Case Name</b>	Train Object		
<b>Author</b>	ThanhPT		
<b>Date</b>	20/03/2017	<b>Priority</b>	Normal
<b>Actor:</b>			
- Admin			
<b>Summary:</b>			
- This use case allows admin train object by adding image into concept from log file.			
<b>Goal:</b>			
- The system can recognize object has been added to the system.			

**Triggers:**

- Admin sends trains object command.

**Preconditions:**

- Account must be login with admin role.
- Log must be existed.

**Post Conditions:**

- **Success:** The system can recognize the trained object.
- **Fail:** System cannot recognize newly added object. System shows error response.

**Main Success Scenario:**

Step	Actor Action	System Response
1	Admin goes to logs view.	System shows all logs.
	Admin choose log to train.	System show table include: image, all name of concept had existed.
2	Admin choose name of concept had existed to update. [Alternative 1]	
3	Admin sends command to train object.	System run training process. Show success message when done. Redirects to logs view.

**Alternative Scenario:***[Alternative 1]*

Step	Actor Action	System Response
1	Admin sends command to create new concept.	System redirect to create new concept page.
2	Admin goes to create new concept view.	System require information from actor: <ul style="list-style-type: none"> <li>- Object Name: free text input, required. The size is [2,100] characters.</li> <li>- Description: free text input, required. The size is [2,100]</li> </ul>
3	Admin inputs information.	
4	Admin sends command to create new object.	

		System validates inputted information. [Exception 1,3]
5		System creates new object and redirect to view concept page. [Exception 2]

**Exceptions:**

No.	Cause	System Response
1	Object name is empty.	System shows error message: “Tên đồ vật không được để trống”.
2	Rate limit is exceeded	System responses: “Tài khoản đã đạt đến giới hạn tạo concept”.
3	Description is empty	System responses: “Mô tả đồ vật không được để trống”

**Relationships:** N/A.**Business Rules:**

- Support image formats are JPG, PNG and JPEG.
- After object is trained successfully, Active state will be updated to “False”.
- Log structure in database:

ID	CREATED_DATE	IMAGE_URL	USER_ID	ACTIVE

- o **ID:** Log ID: integer
- o **User\_ID:** Id of authorized user that requested the identification of object: String
- o **Image\_URL:** URL of the object image: String
- o **Created\_Date:** Date and Time when the log is created: Datetime Format: dd/MM/yyyy hh:mm:ss.
- o **Active:** activation state of the log file: Bit
- Concept must have at least 1 image.
- If Concept is already existed, the inputted image will be update to that Concept.

*Table 28: USE CASE – UC\_AD05 - <Admin > Train Object***3. Software System Attribute****3.1 Usability****3.1.1 Graphic User Interface**

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



**3.1.2 Usability**

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

**3.2 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.

**3.3 Availability**

- Server has back up method to protect and restore necessary data as soon as possible whenever there is a problem.

**3.4 Security**

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

**3.5 Maintainability**

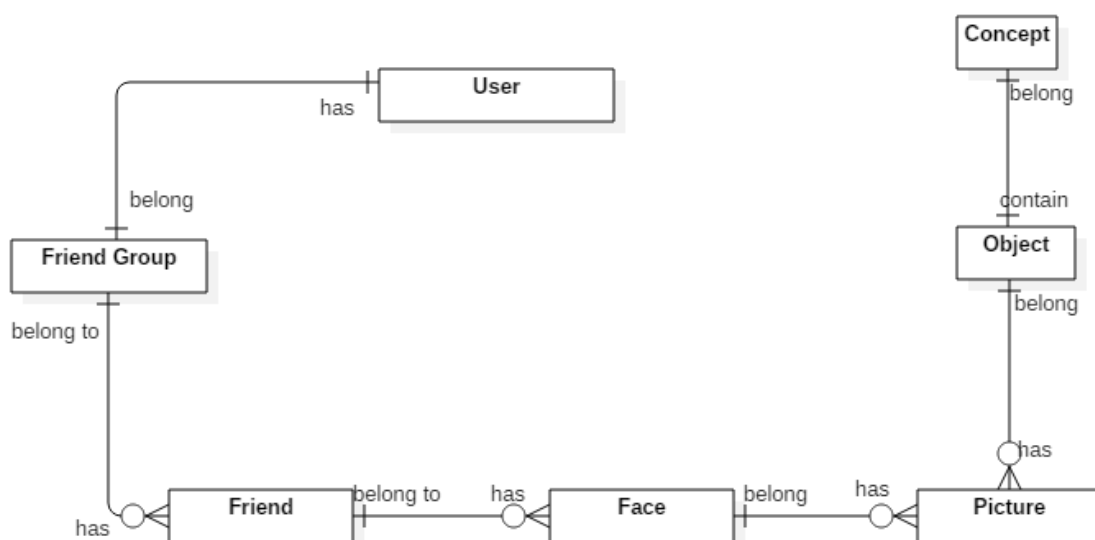
- The system is divided into separated modules.

**3.6 Portability**

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

**3.7 Performance**

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

**4. Conceptual Diagram***Figure 21: Conceptual Diagram***Data Dictionary**

Entity Name	Description
-------------	-------------

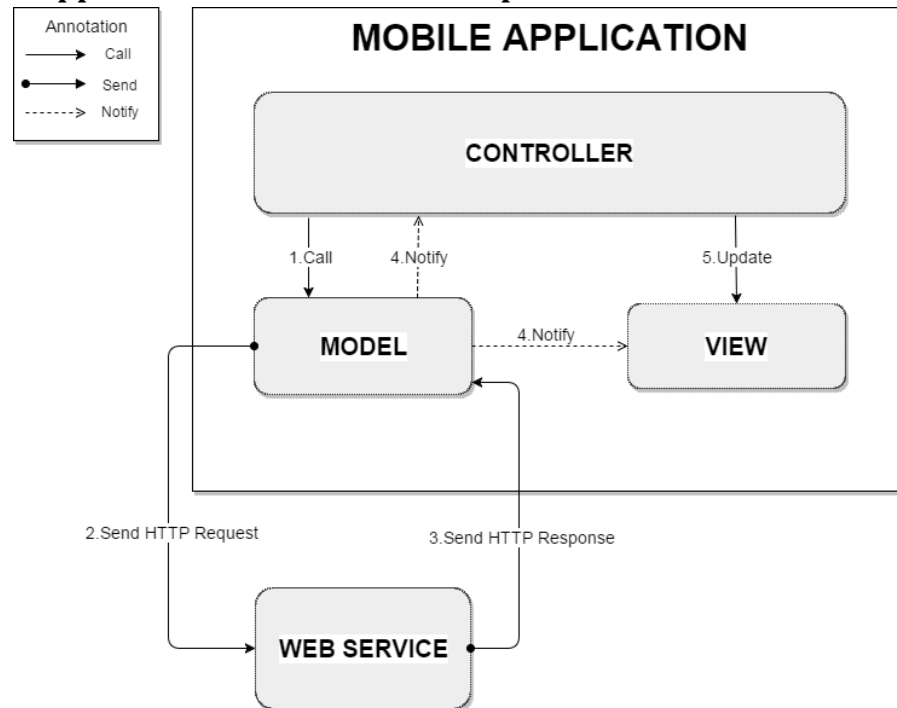
User	The person who using the system
Friend Group	Contain all properties about friend group
Friend	Contain all properties about friend of the blind
Face	Contain all properties about face of Friend
Picture	Store all picture of object
Object	Contain all properties of object
Concept	Contain all properties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

*Table 30: Data Dictionary*

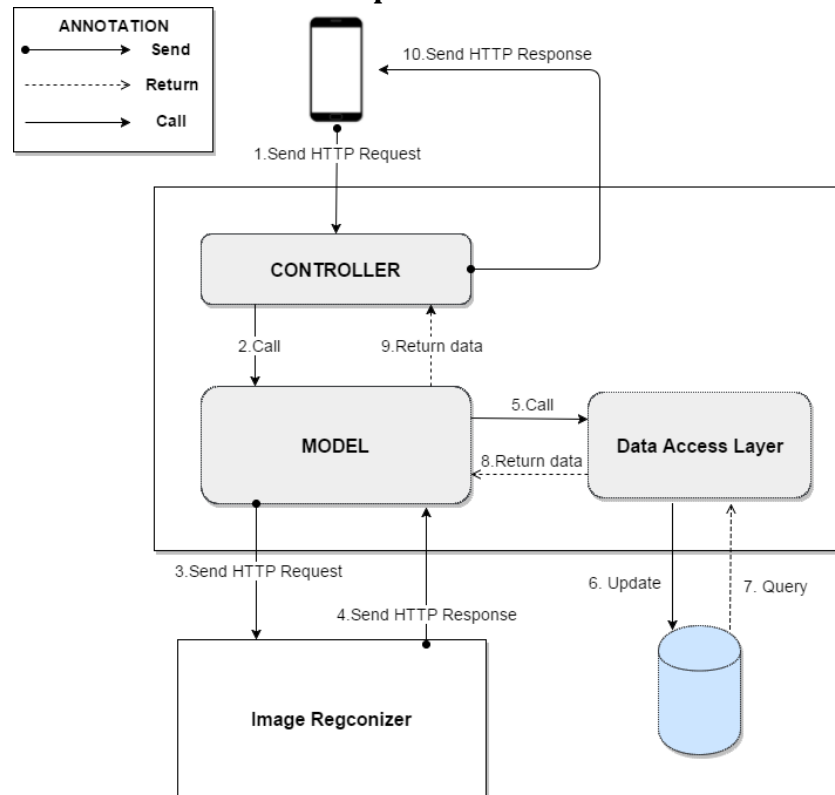
## D. Software Design Description

### 1. System Architectural Design

#### 1.1 Mobile Application Architecture Description

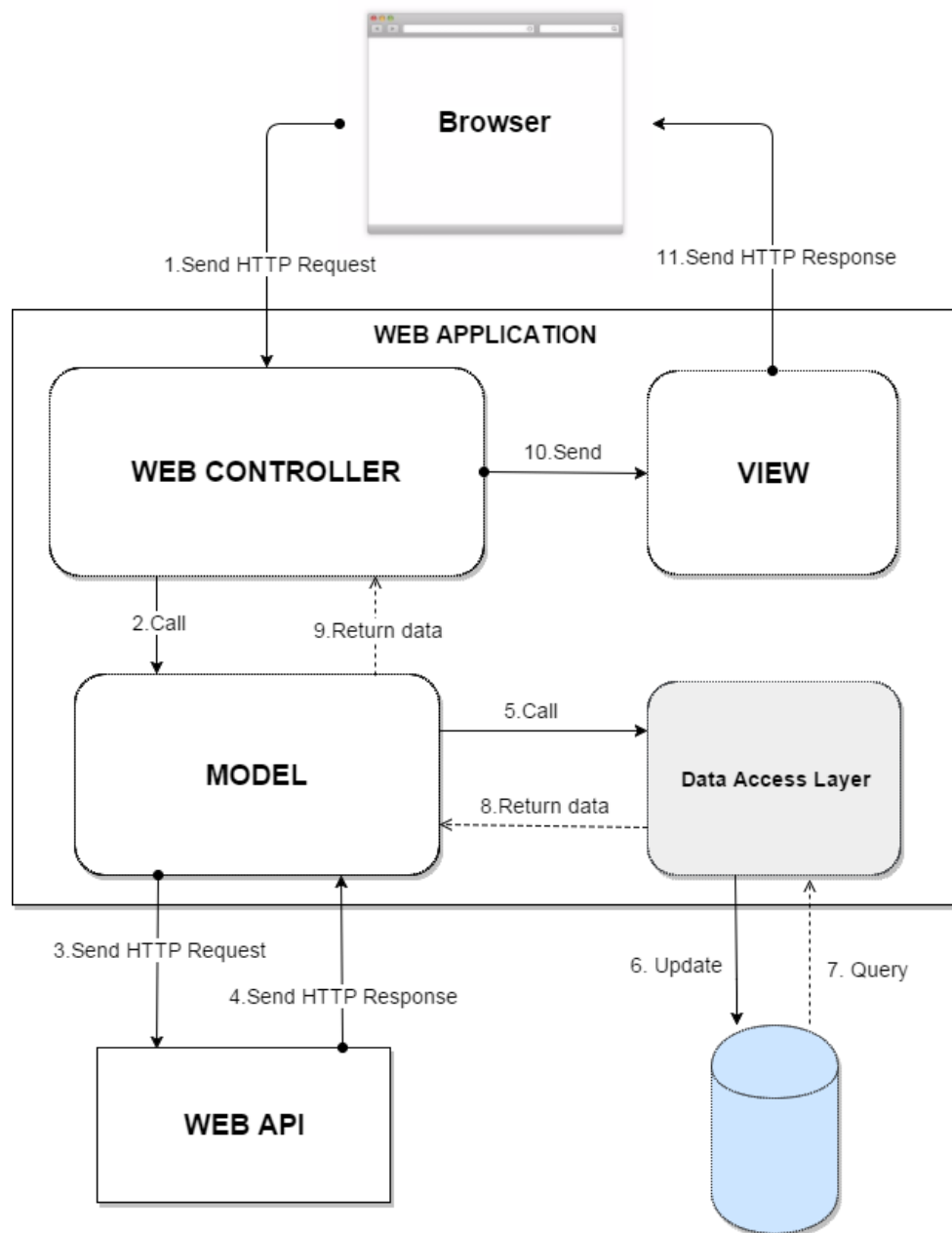
*Figure 22: Mobile Application Architecture Description*

## 1.2 Web Service Architecture Description



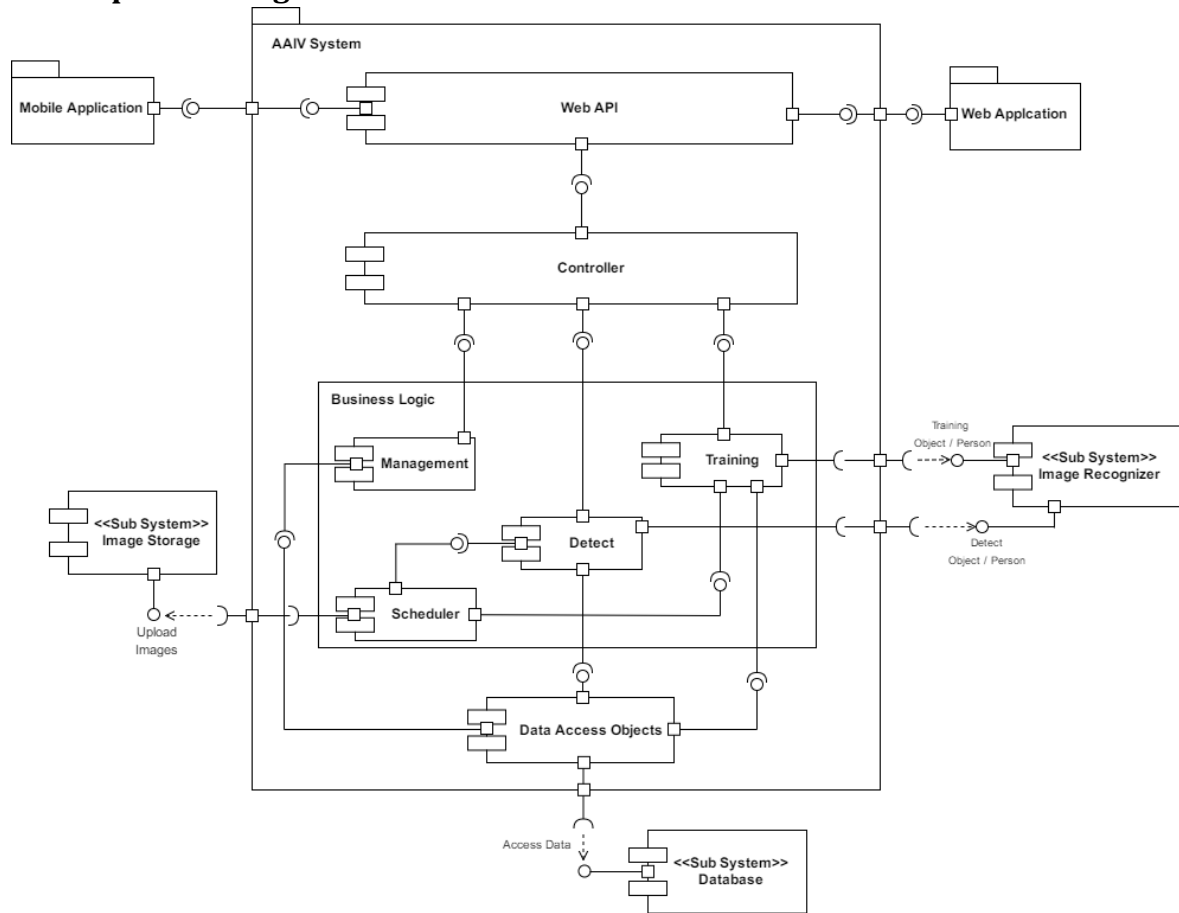
*Figure 23: Web Service Architecture Description*

### 1.3 Web Application Architecture Description



*Figure 24: Web Application Architecture Description*

## 2. Component Diagram



*Figure 25: Component Diagram*

Name	Description
AAIV System	Automatic Alternative Image Recognition to Voice System
Web Api	Component is used to provide API for application to interact with the system.
Business Logic	Common services are used to handle system's business operations
Controller	Component to handle HTTP request
Management	Component to manage system data
Image Recognizer	External component for doing image recognizer
Image Storage	External component for storing images.
Detect	Component to handle detecting requests

Training	Component to train new Person and Object to be recognized by system
Data Access Object	Abstract interface to interact with database system
Database System	Component for system to store and access Data
Scheduler	Run background job and send message to other system

### 3. Detailed Description

#### 3.1 Class Diagram

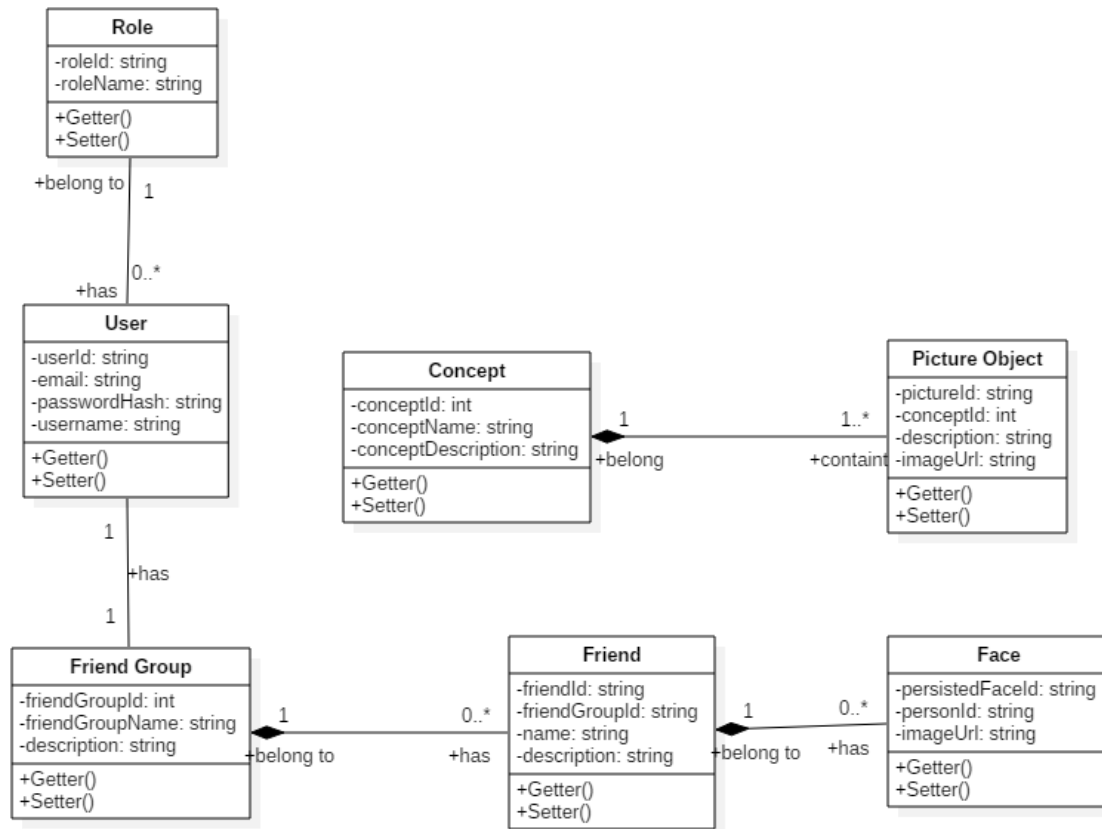


Figure 26: Class Diagram

Class dictionary: Describe Class		
Class Name	Mapping class with conceptual diagram	Description
Role	N/A	Not exist in conceptual diagram. But contains in class diagram to store the permission of entities (Admin, User) after being combined into group "User"

User	User	Contains the information of users in the system
Friend Group	Friend Group	Contain the information of friend group in the system
Friend	Friend	Contains the friends of user in the system
Concept	Concept	Contains the information of object
Picture Object	N/A	Contains the image of object
Face	Face	Contains the information of face

*Table 32: Class dictionary – Describe Class*

### 3.2 Class Diagram Explanation

*(References to main document, Section 4.4.2)*

### 3.3 Interaction Diagram

#### 3.3.1 Mobile Application

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

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

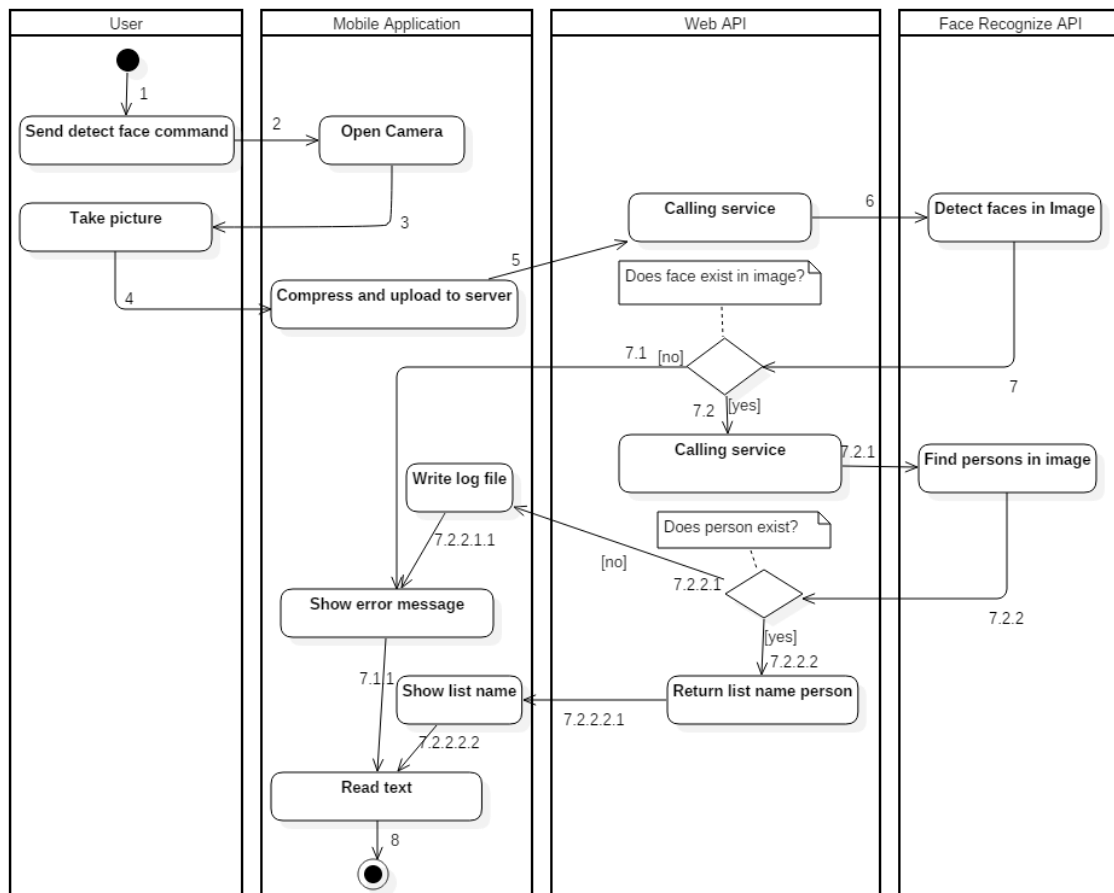


Figure 29: Activity Diagram - <Authorized User> Detect Face

1. Start	2. Send detect face command.	3. Open camera.
4. User sends 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.



### 3.3.1.2 <Authorized User> Detect View

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

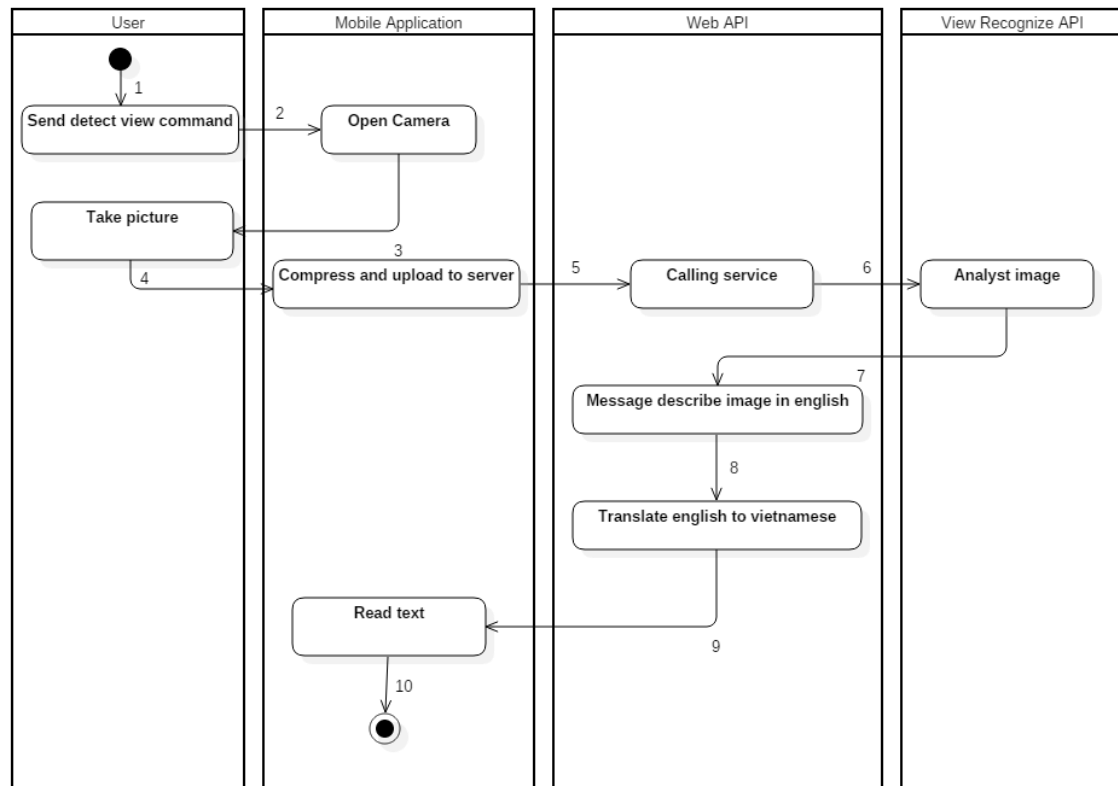
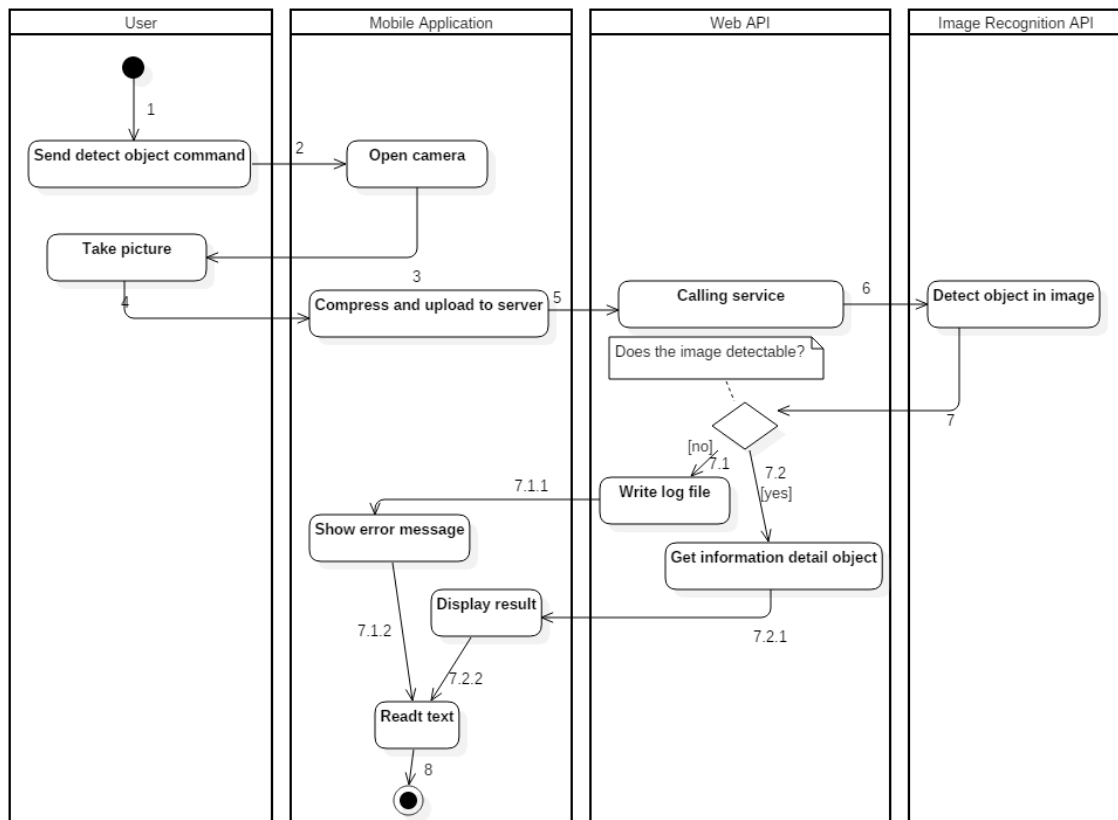


Figure 30: Activity Diagram - <Authorized User> Detect View

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		

### 3.3.1.3 <Authorized User> Detect Object

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

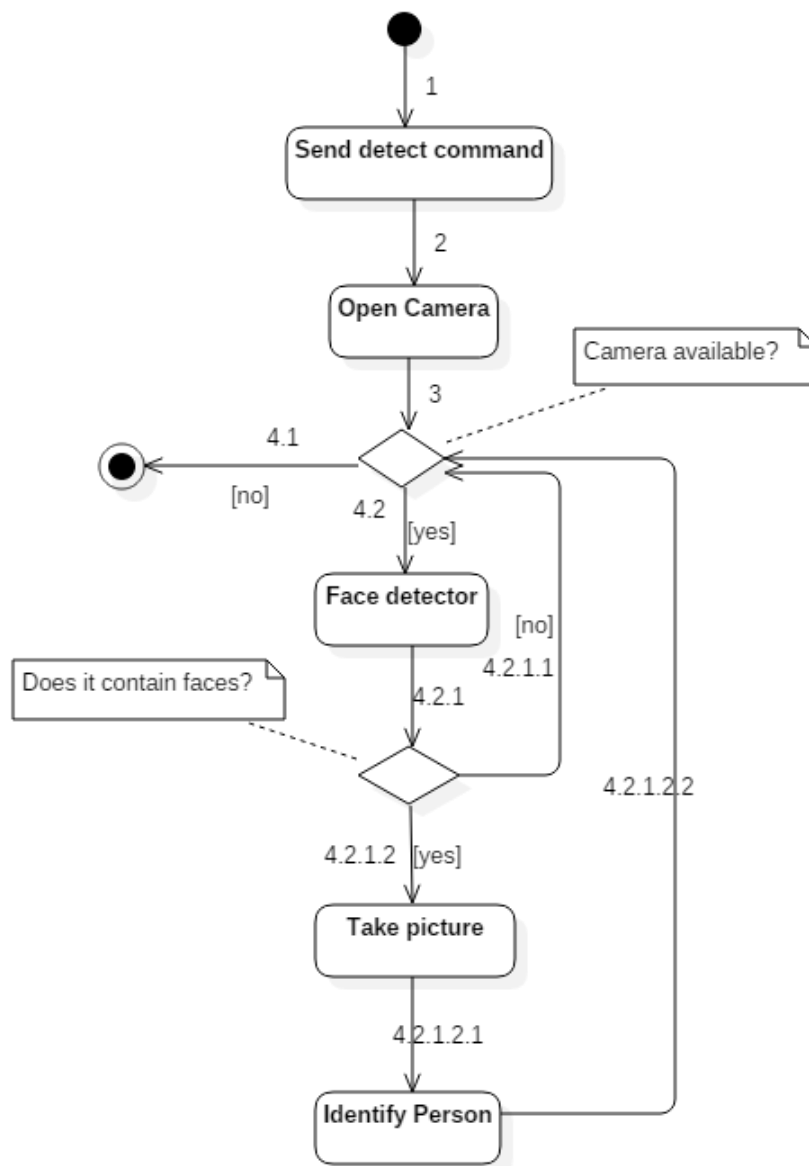


*Figure 31: Activity Diagram - <Authorized User> Detect Object*

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.	

### 3.3.1.4 <Authorized User> Detect Person Real-time

Summary: This diagram show the process of Authorized User detect person real-time using mobile application



*Figure 32: Activity Diagram - <Authorized User> Detect Person Real-time*

1. Start.	2. Send detect person real-time 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		

### 3.3.2 Web Application

#### 3.3.2.1 <Admin> Create Concept

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

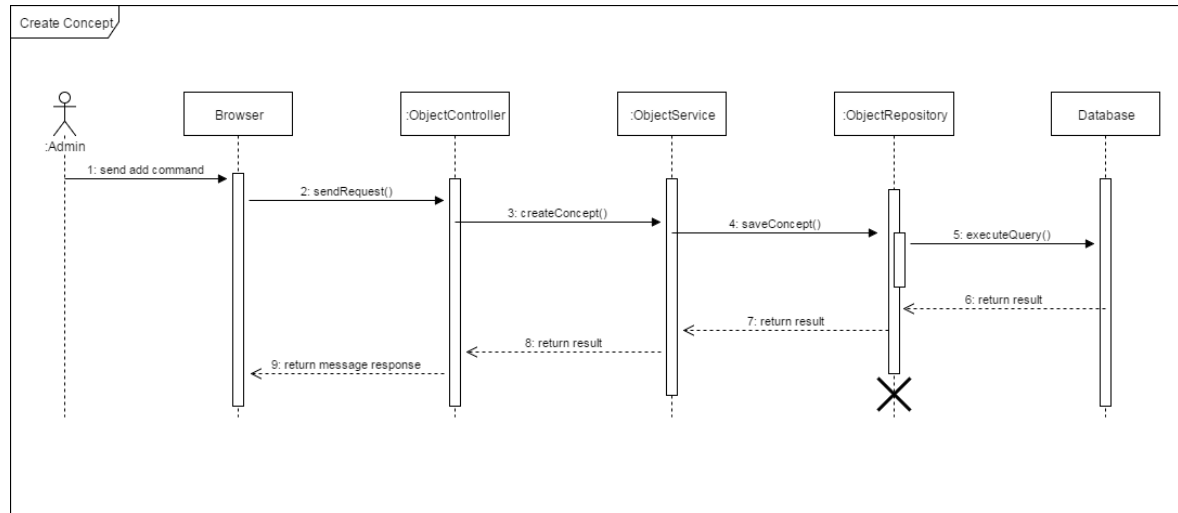


Figure 33: Sequence Diagram - <Admin > Create Concept

#### 3.3.2.2 <Admin> Edit Concept

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

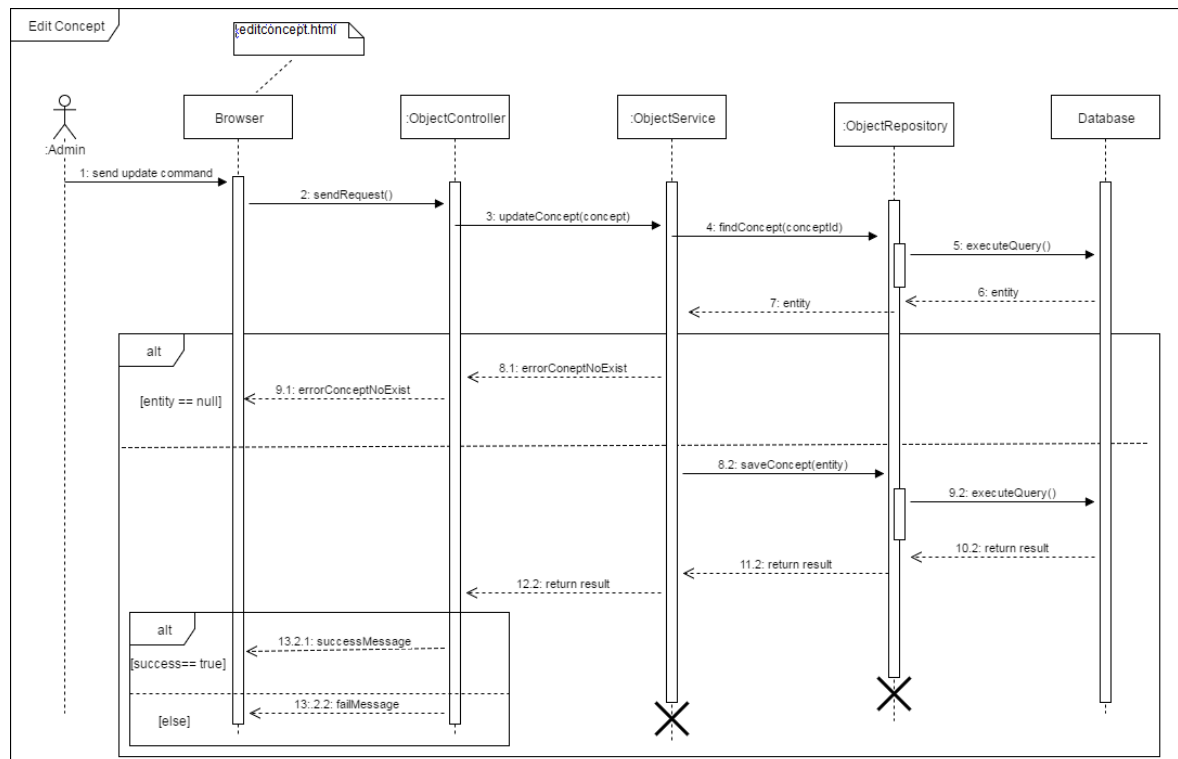


Figure 34: Sequence Diagram - <Admin >Edit Concept

### 3.3.2.3 <Admin> Get Log Object

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

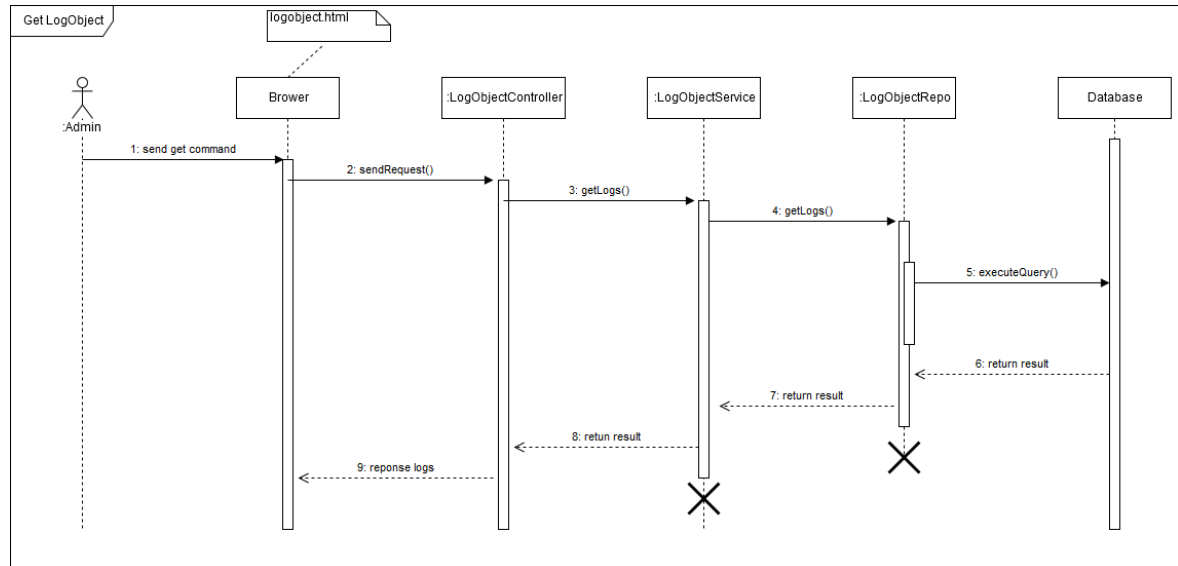


Figure 35: Sequence Diagram - <Admin > Get Log Object

### 3.3.2.4 <User> Get Log Person

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

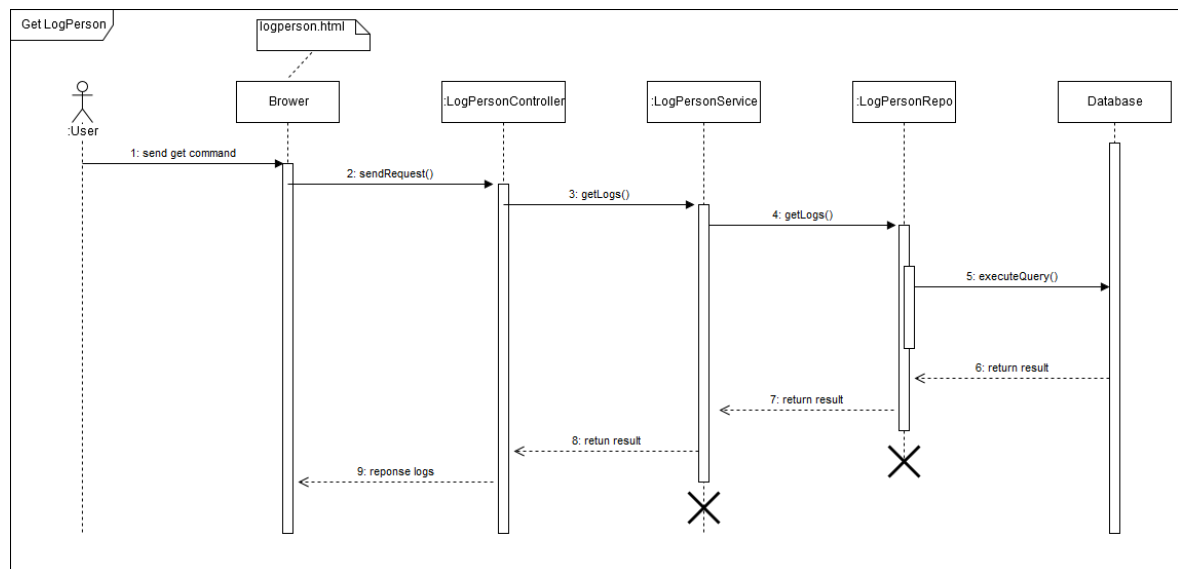
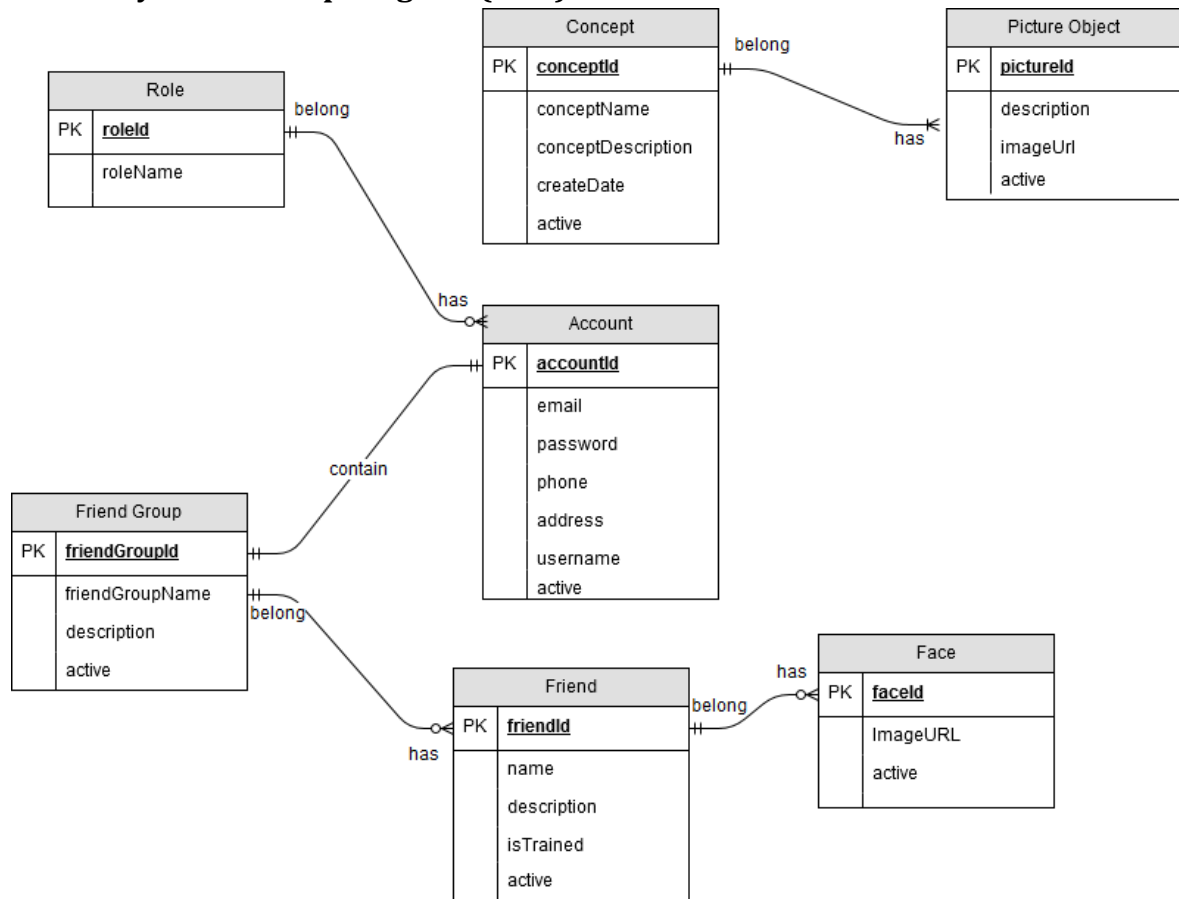


Figure 36: Sequence Diagram - <Admin > Get Log Person

## 4. Database Design

### 4.1 Entity relationship diagram (ERD)



*Figure 67: <Admin > Entity relationship diagram (ERD)*

### 4.2 Data Dictionary

Entity Data Dictionary: describe contents of all entities	
Entity Name	Description
Account	Contains all user in the system
Role	Contains all role in the system
Friend Group	Contains all friend group in the system
Friend	Contains all friend of users in the system
Face	Contains all face of users in the system
Concept	Contains all concept in the system
Picture Object	Contain all picture of object in the system

*Table 83: Data Dictionary*

## 5. Algorithms

### 5.1 Voice Recognition Algorithms

#### 5.1.1 Definition

- This algorithms is used to recognize user voice command when interact with the application.

### 5.1.2 Define Problem

- The main user of the application is a blind person, so the application needs to provide a way that users can use to communicate with the application.

### 5.1.3 Solution

- The Voice Recognition process can be broken down into followed steps:
  - Step1: User input voice data
  - Step2: Analog-to-Digital converter transfer analog wave into digital data that computer can understand. The process contains followed smaller steps:
    - Digitizes the input sound by taking precise measurement of the wave at the frequent intervals.
    - Filter out unwanted sounds.
    - Adjust the input voice to a constant volume level.
    - Adjust the input voice's speed to match the samples stored in system memories.
  - Step3: Using language models and statistical analysis to reduce mistakes, analyze language pattern and speed up the process.
  - Step4: Artificial neural networks is used to recognize the words in the inputted voice after exhaustive training.
- Android Operating System provides a Voice Recognition Module that can analyze user input voice and transfer them into text.
- Using the return text, the application can determine which function the user want to trigger.
- Since the Voice Recognition Module of Android Operating System has similar procedure and pattern, we decided to use this module to help user communicate with the application.

### 5.1.4 Complexity

- Since the complexity of the algorithms is not described or mentioned in the provider documents, we spent times testing and measuring the speed of the Voice Recognition process.
- The result is presented as bellow:

<b>Input Voice \ Network Type</b>	<b>3G Network</b> (Ping: 24ms, Download: 15.31 Mbps, Upload: 1.36 Mbps)	<b>Wi-Fi Network</b> (Ping: 3ms, Download: 30.55Mbps, Upload: 22.17 Mbps)
“Nhận diện hình ảnh”	0.512	0.312
“Thêm người mới”	0.482	0.315
“Cái gì đây”	0.465	0.3

### 5.1.5 References

- <http://www.explainthatstuff.com/voicerecognition.html>
- <https://www.youtube.com/watch?v=5Gn0soUYmLM>
- <https://developer.android.com/reference/android/speech/SpeechRecognizer.html>

## 5.2 Training Person Algorithms

### 5.2.1 Definition

- The training algorithms allow the system to learn and identify a person face by their images provided by the user.

Training process has two meaning:

- Train the system to identify a completely new Person
- Train an existed Person to increase the system's accuracy.

### 5.2.2 Define Problem

- User uses the application to identify Person by their image, so we need an algorithms to train the system to recognize the Person and improve the detecting accuracy.

### 5.2.3 Image Recognizer Concepts

- Person Group: Each person will belong to a Person Group. A Person Group can have up to 1000 person. Each User will have 1 Person Group.
- Person Face: The face of the Person. Each Person can have up to 248 faces.
- Person ID: Each person when created will have a unique Person ID. After the Identifying process, Image Recognizer will return the candidates as their Person ID.
- Face ID: When the image recognizer detects a human face in the image, it will return the Face ID of the detected human face, along with face attributes (age, gender...). Each Face ID will exist within 24 hours.
- Persisted Face ID: When the adding a new face to a person, the Image Recognizer will return a Persisted Face ID of that person face, which will not be expired.

### 5.2.4 Solution

The training person process includes these steps:

- Step 1: User creates a new Person with basic information: Name, Description.
- Step 2: User input images of the person to Image Recognizer. The frontal and near-frontal face will give best result. Each image shall contain one face only. User can resize or crop the person face where they see fit.
- Step 3: Image Recognizer detect the human face in the uploaded images, add the face from image into the person then return a Persisted Face Id of the person face.
- Step 4: Image Recognizer trains the Person Group that contains the Person.
- Step 5: After successfully trained, the system can now identify the person.

### 5.2.5 Complexity

For the complexity of this algorithms is depend on the Image Recognition Service, which is not described in their documents. Therefore we spent times testing the performance of the Training Process:



<b>Images Amount</b>	<b>Wi-Fi Network</b> (Ping: 3ms, Download: 30.55Mbps, Upload: 22.17 Mbps)	<b>3G Network</b> (Ping 24ms, Download: 15.31 Mbps, Upload: 1.36 Mbps)
2	8,951 ms	6,893 ms
3	5,643 ms	7,072 ms
4	7,134 ms	8,382 ms
5	11,325 ms	9,438 ms

### 5.2.6 Accuracy

The accuracy of the algorithms is tested and the result is presented as bellow:

<b>Image Amount</b>	<b>Confident Percentage</b>
1	0.76543
2	0.776
3	0.80058
4	0.80287
5	0.80815

## 5.3 Detect Person Algorithms

### 5.3.1 Definition

- This algorithm is used to identify person that matched the captured picture from user.

### 5.3.2 Define Problem

- The user needs to use the application to identify unknown person. The user can command the application to capture a picture of the person, after the identification process, the person's information will be returned.
- If the person is failed to be identified, the application will return the gender of the person, and ask if user wants to add this person into contact.

### 5.3.3 Solution

The Identifying process content these followed steps:

- Step 1: User open Detect Person mode by voice command.
- Step 2: User capture picture of the person.
- Step 3: System stores captured picture and send image URL to Image Recognizer to detect human face(s) as Face IDs and their Gender (Male or Female). If no human faces are detected in the image, the system will return response message to user by voice.
- Step 4: System receives Face IDs and sends them to Image Recognizer with the Person Group ID of the user to identify the person.
- Step 5: If Image Recognizer is unable to identify the person (no candidates returned or confident rate is  $\leq 65\%$ ), the application will describe the person Gender back to the user by voice.
- Step 6: Otherwise the application will describe the person information back to the user by voice.

### 5.3.4 Complexity

- For the complexity of this algorithms is depend on the Image Recognition Service, which is not described in their documents. Therefore we spent times testing the performance of the Identifying Process:

Network Type	Speed (Voice Recognition + Image Recognition)
Wi-Fi Network (Ping: 3ms, Download: 30.55Mbps, Upload: 22.17 Mbps)	1,699 ms
<b>3G Network</b> (Ping 24ms, Download: 15.31 Mbps, Upload: 1.36 Mbps)	2,372 ms

## 5.4 Person Group For Each User

### 5.4.1 Definition

- This document will explain why each user will have only one unique Person Group but not many Person Group.

### 5.4.2 Define Problem

- Microsoft Cognitive Service provides the **Person Group** terms. This is one of the most important parameters in the Identifying process. Each Person Group can have up to 1000 person.
- If device many person into smaller groups, the identifying process can be faster.
- But because the user is blind, so if all the user's person is divided into smaller groups, the Identifying process will have to look in all of the Person Group until the person is found.

### 5.4.3 Solution

- We spent time testing the detecting process's speed, relative to the amount of Person in a Person Group, with a different type of the Internet Connection. The result is presented as bellow:

Amount of Person	Speed – Using 3G Internet connection 10 Mbps	Speed – Using Fiber Optic Internet connection 36 Mbps
2	1,6 seconds	0,73 second
50	1,7 seconds	0,79 second
300	2 seconds	1,1-1,6 second

- The result shows little differences between Person Group with 2 people and Person Group with 300 people. From that, we decided that each user will only have one unique Person Group.

### 5.4.4 References

- <https://www.microsoft.com/cognitive-services/en-us/face-api/documentation/overview>

- <https://westus.dev.cognitive.microsoft.com/docs/services/563879b61984550e40cbbe8d/operations/563879b61984550f30395239>

## **5.5 Find Duplicated Person By Faces Algorithms**

### **5.5.1 Definition**

- This function helps the user to find out persons which have the same faces, whether they have the same name or not.

### **5.5.2 Define Problem**

- When a blind user creates a new person, the application asks the user what is the person name, the user will speak the name of the person they want to add. But sometimes, that person has already existed in the list, so that we will have the same persons with different names.
- There has to be an option to allow user to merge those persons into one person.

### **5.5.3 Solution**

To solve the problem, follow these step:

- Given a person list of the user, each person will have a list of Faces.
- Sort this list ascending by the number of faces of each person.
- Compare each face of each unmatched person to the rest of the persons in the list, if the confidence rate (the number represents the matching percentage of the face and the compared person) is not less than 0.7, we accept the face and the compared person are matched.
- If a person has at least 3 faces or 50% the number of faces are matched with another person, we will mark these persons are the same.
- If no matched person found, we continue to the next person.
- Continue the process until the list has less than 2 people.
- After the process, we will have a list of person list with the same faces.

- Pseudo code:

**Start**

**Sort** the person list by a number of faces **ascending**.

**For** (each unmatched person [A] in the person list) **do**

**For** (each unmatched person [B] in the rest list) **do**

**For** (each face [F] of the person [A]) **do**

**Compare** each face [F] with person [B] using Microsoft Cognitive Service Verify API

**If** (returned confidence number  $\geq$  average\_matched\_confidence)

**Mark** face [F] and person [B] as matched

**If** (at least 3 faces or 50% number of faces 're matched)

**Add** matched persons to the result list

**Return** result list

Ex:

- Given person list: [P1, P2, P3, P4, P5, P6]
- Compare faces of P1 to P2, P3, P4, P5, P6
- P1 matched with P2, P6, put them into a new list [P1, P2, P6]
- Continue with the unchecked person
- P3 doesn't match P4 or P5
- P4 matched with P5, put them into a new list [P4, P5]
- Results: [ [P1, P2, P6], [P4, P5] ]

#### 5.5.4 Complexity

- Given N is the number of persons in the list.
- Given M is the average number of faces per person.
- The sort algorithm takes  $O(N^2)$  in the worst case.
- The complexity of comparing each person's face to the others is  $O(M * N^2)$
- ⇒ The complexity of this algorithm is:  $O(M * N^2)$

#### 5.5.5 Accuracy

- From testing, the average confidence rate from Image Recognizer Service is more than 72%.
- Test result:  
<https://docs.google.com/spreadsheets/d/1YnqJRSUiCeizhDTHoaJgMHrDWH3k4D5ou80jy2vLOA/edit?usp=sharing>

## 5.6 Find Duplicated Person By Name Algorithms

### 5.6.1 Definition

- This function helps the user to find out persons which have the same name in the Person contact list.

### 5.6.2 Define Problem

- When the blind user adds a new person, they cannot determine whether there is already an existed person with the same name or not. The application will automatically create new the new person, result in there will be people in the person group that have the same name.

### 5.6.3 Solution

To solve this problem, we follow these steps:

- Find all Person in the contact list of the user, then group persons which have the same name.
- Ex:
  - + We have the person list: [Nguyen Van A, Nguyen Van B, Nguyen Van C, Nguyen Van B, Nguyen Van D, Nguyen Van C].
  - + After grouping, we have [Nguyen Van B, Nguyen Van B], [Nguyen Van C, Nguyen Van C].
- In each group, we use the “Find duplicated person by faces” which has been described in [1.1] to group persons which have the same name and may represent the same person in real life.
- After the process, we will have a list which contains groups of persons with the same name.

### 5.6.4 Complexity

- Given  $N$  is the number of persons in the list
  - Given  $M$  is the average number of faces per person.
  - The “Group persons by same name” process takes  $O(N)$
  - The “Find duplicated persons by faces” take  $O(M * N^2)$  (reference 4.7.4.4)
- ⇒ The complexity of this algorithm is:  $O(M * N^2)$

## 5.7 Training Object Algorithms

### 5.7.1 Definition

- The training algorithms allow the system to learn and identify an object images provided by the user.

Training process has two meaning:

- Train the system to identify a completely new Object
- Train an existed Object to increase the system's accuracy

### 5.7.2 Define Problem

- User use the application to identify Object by their image, so we need an algorithms to train the system to recognize the Object and improve the detecting accuracy

### 5.7.3 Image Recognizer Concepts

- **Concept** is a definition that is attached to an image. It could be understood as tags of an image. A concept belongs to a model.
- **Input** is an image used to feed a model. It could be an URL or a raw file.
- **Model**: is a set of concepts (or tags), it could be customized to predict a given image. An output of a model is the percent of each concept in that model.

### 5.7.4 Solution

The training object process includes these steps:

- Step 1: User creates a new object with basic information: Name, Description.
- Step 2: Admin can input one image or multiple images and concept that best describes the new object to Input.
- Step3: Admin creates object with Concept contain Input and name concept. After the system finish success with data (image, concept). The system will start training process in Model.
- Step 4: After successful training, the system then notifies admin training success.

### 5.7.5 Complexity

For the complexity of this algorithms is depend on the Image Recognition Service, which is not described in their documents. Therefore we spent times testing the performance of the Training Process

Network Type	Speed
Wi-Fi Network (Ping: 3ms, Download: 30.55Mbps, Upload: 22.17 Mbps)	12,572 ms
<b>3G Network</b> (Ping 24ms, Download: 15.31 Mbps, Upload: 1.36 Mbps)	13,742 ms

## 5.8 Detect Object Algorithms

### 5.8.1 Definition

- This algorithm is used to identify object that matched the captured picture from user.

### 5.8.2 Define Problem

- The user needs to use the application to identify an unknown object. The user can command the application to capture a picture of the object, after the identification process, the object information will be returned by voice.

### 5.8.3 Solution

The Identifying process content these followed steps:

- Step 1: User open Detect Object mode by voice command.
- Step 2: User capture picture of the object.
- Step 3: System stores captured picture and send image URL to Image Recognizer to detect object.
- Step 4: If Image Recognizer is unable to identify the object (object do not have in the system or confident rate is  $\leq 60\%$ ), a Log file will be created in the system, and the application will return a response message back to the user by voice.
- Step 5: Otherwise the application will describe the object information back to the user by voice.

### 5.8.4 Complexity

- For the complexity of this algorithms is depend on the Image Recognition Service, which is not described in their documents. Therefore we spent times testing the performance of the Identifying Process:

Network Type	Speed (Voice Recognition + Image Recognition)
Wi-Fi Network (Ping: 3ms, Download: 30.55Mbps, Upload: 22.17 Mbps)	5,731 ms
<b>3G Network</b> (Ping 24ms, Download: 15.31 Mbps, Upload: 1.36 Mbps)	6,182 ms

#### 5.8.4.1 Accuracy

The accuracy of the algorithms is tested and the result is presented as bellow:

Image Amount In Concept	Confident Rate				
	> 90%	90% - 80%	80% - 70%	70% - 60%	60% - 50%
1	7	25	23	14	20
2	9	28	19	17	22
3	8	16	33	20	10
4	5	17	35	22	7



**FPT University – Capstone Project Spring 2017 – Group 4 - AAIIV**

**E. Task Sheet**

No.	Product Deliverables	Task	ThanhPT	SonNVH	QuanVH	DuyNC	Size
1	Report1 - Introduction	Project Overview		0			
		Introduction		0			
		Current Situation		0			
		Problem Definition			0		
		Proposed Solution			0		
		Functional Requirements				0	
		Roles and Responsibilities	0				
2	Report2- Software Project Management Plan	<b>Problem Definition</b>					
		Name of this Capstone Project			0		
		Problem Abstract			0		
		Project Overview		0		0	
		<b>Project organization</b>					
		Software Project Model	0				
		Roles and Responsibilities	0				
		Tools and Techniques	0				
		<b>Project Management Plan</b>					
		Product Backlog		0			
		Sprint Backlog		0			
		Deliverables		0			
		All Meeting Minutes			0		
		<b>Coding Convention</b>	0				
3	Report 3- Software Requirement Specification	<b>User Requirement Specification</b>		0			
		<b>Software Requirement Specification</b>					
		External Interface Requirement				0	
		System Overview Use Case	0				

**FPT University – Capstone Project Spring 2017 – Group 4 - AAIIV**

		<b>List of use case</b>					
		<User> Add new Person		0			
		<User> Upload Images		0			
		<User> Train Persons		0			
		<User> Detect Face		0			
		<User> Detect Object		0			
		<User> Detect Real-time		0			
		<Unauthorized User> Sign Up				0	
		<Unauthorized User> Login				0	
		<Admin> Activate Account			0		
		<Admin> Deactivate Account			0		
		<Admin> Remove Concept	0				
		<Admin> Create Concept	0				
		<Admin> Train Object	0				
		<b>Software System Attribute</b>				0	
		<b>Conceptual Diagram</b>	0				
4	Report 4- Software Design Description	<b>Design Overview</b>	0				
		<b>System Architectural Design</b>					
		Web Application Architecture Description			0		
		Web Service Architecture Description			0		
		Mobile Application Architecture Description			0		
		<b>Component Diagram</b>		0			
		<b>Detailed Description</b>					
		Class Diagram	0				
		Class Diagram Explanation	0				
		Interaction Diagram	0				
		<b>Interface</b>					
		Component interface	0				

**FPT University – Capstone Project Spring 2017 – Group 4 - AAIIV**

		User Interface Design		0	0		
		<b>Database Design</b>					
		Entity Relationship Diagram (ERD)	0				
		Data Dictionary	0				
		<b>Algorithms</b>					
		Person Group For Each User		0			
		Voice Recognition Algorithms		0			
		Detect Person Algorithms		0			
		Training Person Algorithms		0			
		Find Duplicated Person By Faces Algorithms			0		
		Find Duplicated Person By Name Algorithms			0		
		Detect Object Algorithms	0				
		Training Object Algorithms	0				
5	Report 5 - Software Implementation and Test Document	<b>Introduction</b>					
		Overview	0				
		Test Approach	0				
		<b>Database Relationship Diagram</b>					
		Physical Diagram	0				
		Data Dictionary	0				
		<b>Implementation</b>					
		Server Architecture Implementation	0	0			
		Mobile Application Architecture Implementation	0	0			
		Web Architecture Implementation		0	0	0	
		Test Driven Development	0				
		<b>Test Plan</b>					
		Communication Diagram	0				
		<b>System Testing Test Case</b>					

**FPT University – Capstone Project Spring 2017 – Group 4 - AAIIV**

		Web Application Test Cases	0				
		Mobile Application Test Cases	0				
		<b>Algorithm</b>					
		Implement Person Group For Each User		0			2
		Implement Voice Recognition Algorithms		0			5
		Implement Detect Person Algorithms	0	0			5
		Implement Training Person Algorithms		0			5
		Implement Find Duplicated Person By Faces Algorithms		0	0		3
		Implement Find Duplicated Person By Name Algorithms			0		2
		Implement Detect Object Algorithms	0				5
		Implement Training Object Algorithms	0				5
		<b>Server</b>					
		Implement manage person		0			1
		Implement manage object	0				1
		Implement train	0				5
		Implement identity	0				5
		<b>Web</b>					
		Implement manage accounts		0			1
		Implement manage person			0		2
		Implement manage concepts				0	2
		Implement train person			0		3
		Implement manage logs			0		3
		Implement train object				0	3
		<b>Mobile</b>					
		Implement detect person		0			5
		Implement detect person real time	0	0			5

## FPT University – Capstone Project Spring 2017 – Group 4 - AAIV

		Implement detect object	0				5
		Implement resize picture		0			2
		Implement option menu		0			1
		Implement notification		0			2
		Implement voice control		0	0		5
		Implement write log person		0			1
		Implement write log object	0				1
		Implement upload image		0			1
6	Report 6 - Software User's Manual	<b>Installation Guide</b>					
		Setting up environment at server side		0			
		Web Services Application Deployment Process		0			
		<b>Installation Guide</b>					
		Mobile application		0			
		Web application		0			

### F. Appendix

1. UML standard diagrams:  
[https://www.tutorialspoint.com/uml/uml\\_standard\\_diagrams.htm](https://www.tutorialspoint.com/uml/uml_standard_diagrams.htm)
2. Component diagram notation:  
<http://www.uml-diagrams.org/component-diagrams.html>
3. UML Diagrams:  
<http://www.uml-diagrams.org/>
4. Clarifai documents:  
<https://developer.clarifai.com/docs/>
5. Microsoft documents:  
<https://www.microsoft.com/cognitive-services/en-us/documentation>
6. Algorithms document:  
<https://medium.com/@ageitgey/machine-learning-is-fun-part-3-deep-learning-and-convolutional-neural-networks-f40359318721>

7. Detect face Google vision document:  
<https://developers.google.com/vision/face-detection-concepts>