



MINISTRY OF EDUCATION AND TRAINING



# FPT UNIVERSITY

## Capstone Project Document

---

### Automatic Alternative Image Recognition to Voice

Group 04	
<b>Group members</b>	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)
<b>Supervisor</b>	Mr. Kiều Trọng Khánh
<b>Ext. Supervisor</b>	N/A
<b>Capstone Project code</b>	AAIV

*This page is intentionally left blank*

### **ACKNOWLEDGEMENTS**

We wish to thank various people for their contribution to this project: Our teachers for their advice and participation in the final review, our friends for their valuable technical support.

Special thanks should be given to Mr. Kiều Trọng Khánh, our research supervisor for his professional guidance and the useful, constructive recommendations throughout the course of this project.

## Table of Contents

1	Introduction .....	1
1.1	Project Information .....	1
1.2	Introduction .....	1
1.3	Current Situation .....	1
1.4	Problem Definition.....	1
1.5	Proposed Solution.....	2
1.5.1	Feature functions .....	2
1.5.2	Advantages and disadvantages.....	2
1.6	Functional Requirements.....	2
1.7	Role and Responsibility .....	3
2	Software Project Management Plan .....	4
2.1	Problem Definition.....	4
2.1.1	Name of this Capstone Project.....	4
2.1.2	Problem Abstract.....	4
2.1.3	Project Overview.....	4
2.2	Project organization.....	8
2.2.1	Software Process Model.....	8
2.2.2	Roles and responsibilities .....	8
2.2.3	Tools and Techniques.....	10
2.3	Project Management Plan.....	10
2.3.1	Product Backlog.....	10
2.3.2	Sprint Backlog.....	12
2.3.3	Deliverables.....	17
2.3.4	All Meeting Minutes .....	18
2.4	Coding Convention.....	18
3	Software Requirement Specification .....	19
3.1	User Requirement Specification .....	19
3.1.1	Guest Requirement .....	19
3.1.2	User Requirement.....	19
3.1.3	Admin Requirement.....	19
3.2	System Requirement Specification .....	20
3.2.1	External Interface Requirement.....	20
3.2.2	System Overview Use Case.....	20
3.2.3	List of Use Case.....	22

3.3	Software System Attribute.....	53
3.3.1	Usability .....	53
3.3.2	Reliability .....	53
3.3.3	Availability .....	53
3.3.4	Security.....	53
3.3.5	Maintainability.....	53
3.3.6	Portability .....	53
3.3.7	Performance.....	53
3.4	Conceptual Diagram.....	54
4	Software Design Description .....	55
4.1	Design Overview .....	55
4.2	System Architectural Design.....	55
4.2.1	Mobile Application Architecture Description.....	56
4.2.2	Web Service Architecture Description.....	57
4.2.3	Web Application Architecture Description .....	58
4.3	Component Diagram.....	59
4.4	Detailed Description.....	60
4.4.1	Class Diagram.....	60
4.4.2	Class Diagram Explanation .....	61
4.4.3	Interaction Diagram.....	64
4.5	Interface.....	72
4.5.1	Component interface .....	72
4.5.2	User Interface Design .....	76
4.6	Database Design .....	112
4.6.1	Entity relationship diagram (ERD) .....	112
4.6.2	Data Dictionary.....	112
4.7	Algorithms.....	113
4.7.1	Person Group For Each User.....	113
4.7.2	Voice Recognition Algorithms.....	114
4.7.3	Image Recognition Algorithms .....	116
4.7.4	Detect Person Algorithms .....	117
4.7.5	Training Person Algorithms .....	119
4.7.6	Find Duplicated Person By Faces Algorithms .....	122
4.7.7	Find Duplicated Person By Name Algorithms .....	125
4.7.8	Detect Object Algorithms .....	127

4.7.9	Training Object Algorithms .....	130
5	System Implementation & Test.....	132
5.1	Introduction.....	132
5.1.1	Overview .....	132
5.1.2	Test Approach.....	132
5.2	Database Relationship Diagram .....	132
5.2.1	Physical Diagram.....	132
5.2.2	Data Dictionary.....	134
5.3	Performance Measures.....	137
5.4	Test Plan.....	137
5.5	System Testing Test Case.....	137
5.5.1	Communication Diagram .....	137
5.5.2	Test Cases .....	137
6	Software User's Manual.....	144
6.1	Installation Guide.....	144
6.1.1	Setting up environment at server side.....	144
6.1.2	Web Services Application Deployment Process.....	145
6.1.3	Mobile Application Deployment Process.....	155
6.2	User Guide.....	158
6.2.1	Mobile Application Guide.....	158
6.2.2	Web Application Guide.....	176
7	Appendix.....	192

## **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
Confident Rate	The percentage of accuracy of detecting result

*Table 0: Definitions, Acronyms, and Abbreviations*

## 1 Introduction

### 1.1 Project Information

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

### 1.2 Introduction

- In this document, we introduce a solution for blind people to recognize faces and things that happen 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.

### 1.3 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

### 1.4 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.

## 1.5 Proposed Solution

### 1.5.1 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.

### 1.5.2 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.

## 1.6 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 which the blind captures 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.

### 1.7 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	thanhphtse61288@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*

## 2 Software Project Management Plan

### 2.1 Problem Definition

#### 2.1.1 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:** AAV.

#### 2.1.2 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 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.

#### 2.1.3 Project Overview

##### 2.1.3.1 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.

### 2.1.3.2 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 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).

### 2.1.3.3 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.

### 2.1.3.4 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.

#### 2.1.3.5 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 2: Hardware requirement for Server*

+ For mobile

<b>Windows</b>	<b>Minimum</b>	<b>Recommended</b>
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 3: Hardware requirement for mobile*

+ For web

<b>Windows</b>	<b>Minimum</b>	<b>Recommended</b>
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 4: Hardware requirement for web*

### Software requirement

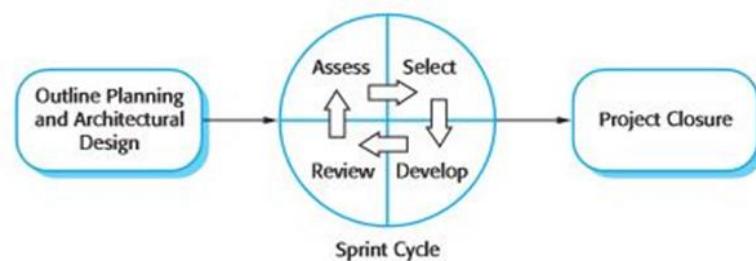
<b>Software</b>	<b>Name / Version</b>	<b>Description</b>
Operating system	Window 7, Window 10	Operating system and platform for development
Environment	.NET	Specification for developing web application
IDE	Android Studio 2.3, Visual Studio 2015	Used for implement website and Android Mobile App.
Design Model tool	StarUML v5.0	Used for creating model 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

## 2.2 Project organization

### 2.2.1 Software Process Model

Our project using Scrum 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.
- The 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))

### 2.2.2 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>- Coding</li> </ul>

No	Full name	Role in Group	Responsibilities
			<ul style="list-style-type: none"> <li>- Test</li> </ul>
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>- 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>- Coding</li> <li>- Test</li> </ul>

*Table 5: Roles and Responsibilities Details*

### 2.2.3 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</li> <li>+ Entity Framework 5</li> <li>- Flask framework(Python)</li> </ul>
<b>Web server</b>	IIS version 1607	<ul style="list-style-type: none"> <li>- C# .NET</li> </ul>
<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*

## 2.3 Project Management Plan

### 2.3.1 Product Backlog

Story ID	Features	Task ID	Task description	Sprint
1	Create Project Back Log	1.1	Create project backlog	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
		9.1	Mobile_Camera Surface View	1
9	Implementation	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
10	Create Software Test Documentation	9.17	System_Implement Vision API with Clarify	4
		10.1	Test Plan	5
		10.2	Test Cases	5
11	Quality Assurance	10.3	Checklists	5
		11.1	Quality Assurance for Backend	5
		11.2	Quality Assurance for Web	5
12	Software User's Manual	11.3	Quality Assurance for Mobile	5
		12.1	Installation Guide	6
		12.2	User's Guide	6

Table 7: Product Backlog

### 2.3.2 Sprint Backlog

#### 2.3.2.1 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 8: Sprint 1 Development*

### 2.3.2.2 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
6	Mobile_Voice Notification of returned result	SonNVH	ThanhPT
7	Mobile_Voice Control	SonNVH, QuanVH	ThanhPT
8	Web_User Login	QuanVH	ThanhPT
9	Web_User Logout	QuanVH	ThanhPT

*Table 9: Sprint 2 Development*

### 2.3.2.3 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	QuanVH	ThanhPT
3	Component Diagram	SonNVH	ThanhPT
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 Design	QuanVH, DuyNC	ThanhPT
8	Database Design	ThanhPT, SonNVH	QuanVH, DuyNC
9	Mobile_Write Log Person File	SonNVH	ThanhPT, QuanVH, DuyNC
10	Mobile_Write Log Object File	ThanhPT	SonNVH, QuanVH, DuyNC
11	Web_User Search + View data	QuanVH	ThanhPT, SonNVH, DuyNC
12	System_Implement realtime mode recognition	ThanhPT, SonNVH	QuanVH, DuyNC

*Table 10: Sprint 3 Development*

#### 2.3.2.4 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
5	Algorithm	SonNVH, QuanVH, ThanhPT	ThanhPT

**Table 11: Sprint 4 Development**

### 2.3.2.5 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
  - + Checklists
- 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	DuyNC, SonNVH, QuanVH
4	Checklists	ThanhPT, SonNVH	ThanhPT
5	Quality Assurance for Backend	ThanhPT	SonNVH, QuanVH, DuyNC
6	Quality Assurance for Web	ThanhPT	SonNVH, QuanVH, DuyNC

7	Quality Assurance for Mobile	ThanhPT	SonNVH, QuanVH, DuyNC
---	------------------------------	---------	-----------------------------

**Table 12: Sprint 5 Development****2.3.2.6 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	SonNVH	ThanhPT
2	User's Guide	SonNVH	ThanhPT

**Table 13: Sprint 6 Development****2.3.3 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

**Table 14: Deliverable**

### 2.3.4 All Meeting Minutes

All meeting minutes can be found at

<https://drive.google.com/drive/folders/0BzzIdYJIXc4cTERKUVFPQmxHRnM>

## 2.4 Coding Convention

### Python: Using for Web API

Naming Convention:

1. 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.
2. Methods should be verbs,

### Java: Using to develop mobile app.

Naming Convention:

1. 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.
2. Methods should be verbs, in mixed case with the first letter lowercase, with the first letter of each internal word capitalized.

Indentation:

1. One declaration per line is recommended since it encourages commenting.
2. In absolutely no case should variables and functions be declared on the same line.
3. Do not put different types on the same line.

Declarations Convention:

1. 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 and Web API

Naming Convention:

1. 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.
2. Methods should be verbs, in mixed case with the first letter uppercase, with the first letter of each internal word capitalized.

Indentation:

1. One declaration per line is recommended since it encourages commenting.
2. In absolutely no case should variables and functions be declared on the same line.
3. Do not put different types on the same line.

Declarations Convention:

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

Using C# Code Convention from:

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

### 3 Software Requirement Specification

#### 3.1 User Requirement Specification

##### 3.1.1 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

##### 3.1.2 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

##### 3.1.3 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

### **3.2 System Requirement Specification**

#### **3.2.1 External Interface Requirement**

##### **3.2.1.1 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

##### **3.2.1.2 Hardware Interface**

- Android Smartphone: Android 6.0.1 Marshmallow or later.

##### **3.2.1.3 Software Interface**

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

##### **3.2.1.4 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.

### **3.2.2 System Overview Use Case**

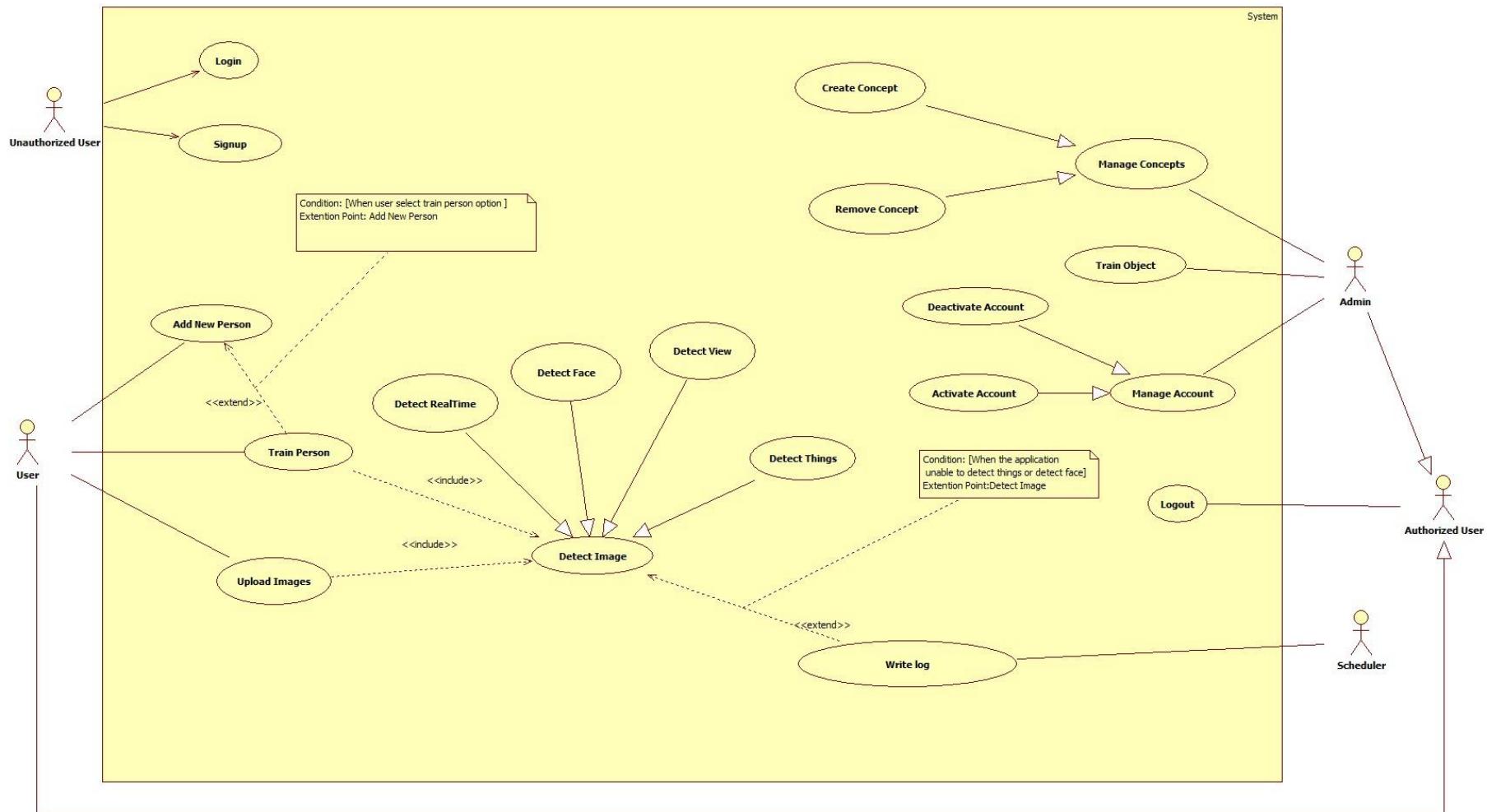
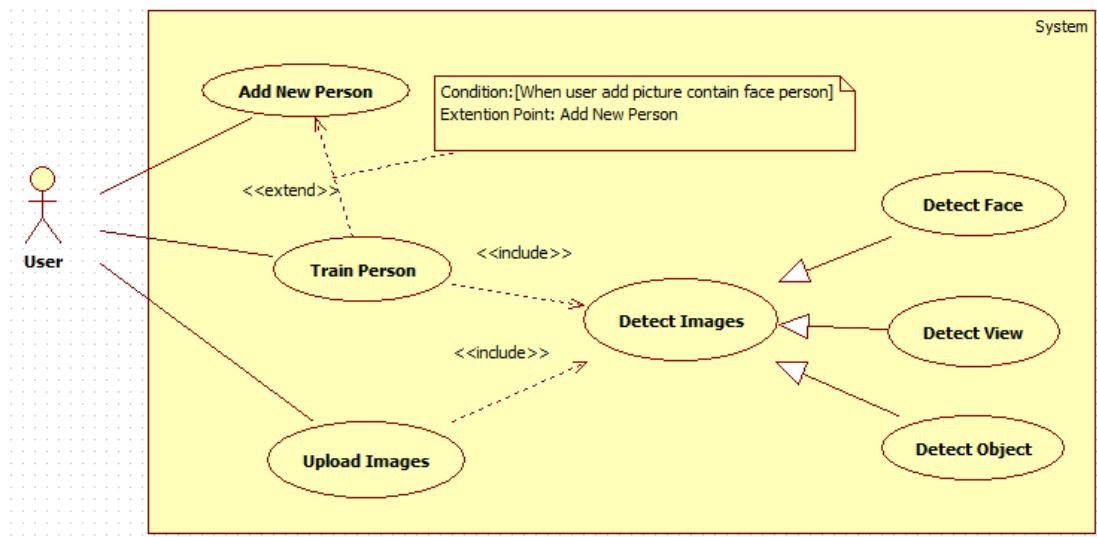


Figure 2: System Overview Use Case

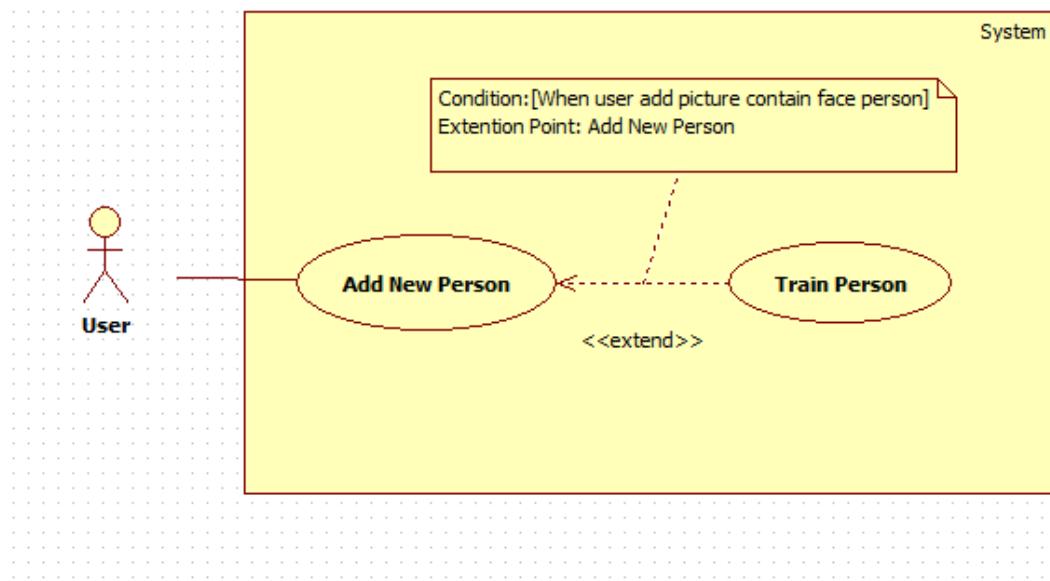
### 3.2.3 List of Use Case

#### 3.2.3.1 <User> Overview Use Case



*Figure 3: User Overview Use Case*

##### 3.2.3.1.1 <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: - Name: free text input. Required. Length is 1-128 - User Description: free text input. The size limit is 128 characters. Length is 0 - 128 - 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:**

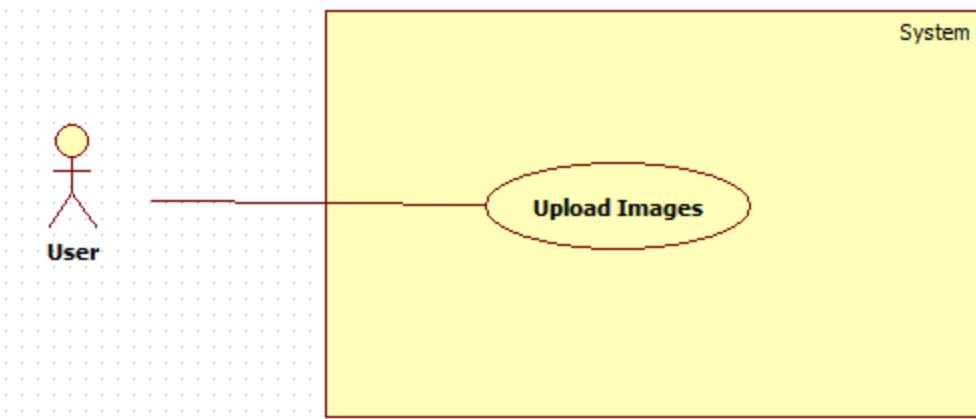
No.	Cause	System Response
1	Name is empty.	System responses: "Name không được bỏ trống".
2	Length of Name is not between 3 - 128 characters.	System responses: "Độ dài của Name trong khoảng 3 -128 ký tự".
3	Image size exceeded the limit size (4Mb)	System automatically compresses the image but remains 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***3.2.3.1.2 <User> Upload Images***Figure 5: <User> Upload Images (UC\_US02)*

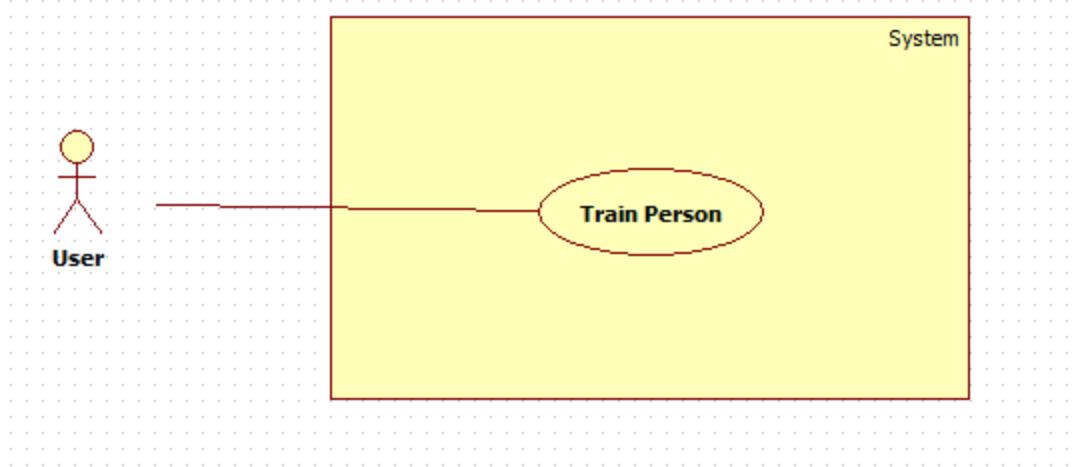
<b>USE CASE – UC_US02</b>					
<b>Use Case No.</b>	UC_US02	<b>Use Case Version</b>	1.0		
<b>Use Case Name</b>	Upload Image				
<b>Author</b>	SonNVH				
<b>Date</b>	04/15/2017	<b>Priority</b>	High		
<b>Actor:</b>	<ul style="list-style-type: none"> <li>- User</li> </ul>				
<b>Summary:</b>	<ul style="list-style-type: none"> <li>- This use case allows user to send a picture to the system. After received the picture, the system can use it for later functions (detections, create new person).</li> </ul>				
<b>Goal:</b>	<ul style="list-style-type: none"> <li>- The taken / uploaded picture from user is stored in the system.</li> </ul>				
<b>Triggers:</b>	<ul style="list-style-type: none"> <li>- When user capture a picture from device's camera.</li> <li>- When user choose an image(s) to create a new person.</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> captured/uploaded picture from user is now stored in the system.</li> <li>- <b>Fail:</b> No image is stored in the system. Show error message.</li> </ul>				
<b>Main Success Scenario:</b>					
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]			
<b>Alternative Scenario:</b>					
[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 chosen 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 responds with error message
2	No Internet connection/request denied by upload host.	System responds 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 16: USE CASE - UC\_US02 - <User> Upload Images***3.2.3.1.3 <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: - personGroupId: hidden free text input. Required. Length is 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]*

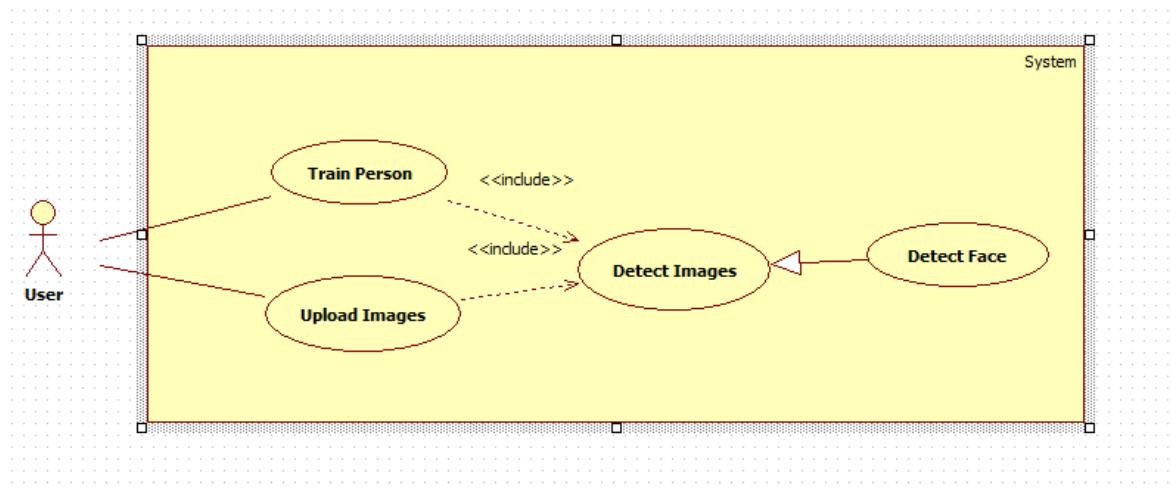
Step	Actor Action	System Response
1	User successfully created a new person	System gets personGroupId from user, begin training process
2		Show success message when done. Redirect to main activity[Exception 4]

**Exceptions:**

No.	Cause	System Response
1	PersonGroupId does not exist	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, less than 300kb.
- User can choose to crop out the face in their picture.
- If training process is fail, return training status to user.

*Table 17: USE CASE - UC\_US03 - <User> Train Person***3.2.3.1.4 <User> Detect Face***Figure 7: <User> Detect Face (UC\_US04)*

<b>USE CASE – UC_US04</b>																								
<b>Use Case No.</b>	UC_US04	<b>Use Case Version</b>	1.0																					
<b>Use Case Name</b>	Detect Face																							
<b>Author</b>	SonNVH																							
<b>Date</b>	04/15/2017	<b>Priority</b>	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 is the person in front of them.</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>	<table border="1"> <thead> <tr> <th>Step</th><th>Actor Action</th><th>System Response</th></tr> </thead> <tbody> <tr> <td>1</td><td>User sends request to detect a person</td><td>System opens camera.</td></tr> <tr> <td>2</td><td>User takes a picture of the person.</td><td>System uploads picture to server. [Exception 1]</td></tr> <tr> <td>3</td><td></td><td>System validate inputted information</td></tr> <tr> <td>4</td><td></td><td>System calls Microsoft Cognitive Service to detect faces in the uploaded image. [Alternative 1] [Exception 4, 5, 6]</td></tr> <tr> <td>5</td><td></td><td>System call Microsoft Cognitive Service to identify detected face(s) in the person group and return candidate(s) [Exception 2, 3, 6] [Alternative 2]</td></tr> <tr> <td>6</td><td></td><td>System gets detected person's information and notify it to the user. [Exception 6]</td></tr> </tbody> </table>			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 gets detected person's information and notify it to the user. [Exception 6]
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 gets detected person's information and notify it to the user. [Exception 6]																						
<b>Alternative Scenario:</b>	<p>[Alternative 1]</p> <table border="1"> <thead> <tr> <th>Step</th><th>Actor Action</th><th>System Response</th></tr> </thead> </table>			Step	Actor Action	System Response																		
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 are 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 asks if user wants to add this person.
3	User sends confirm command [Alternative 3]	System asks user to input the person name.
4	User input person name.	System writes 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 an 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".

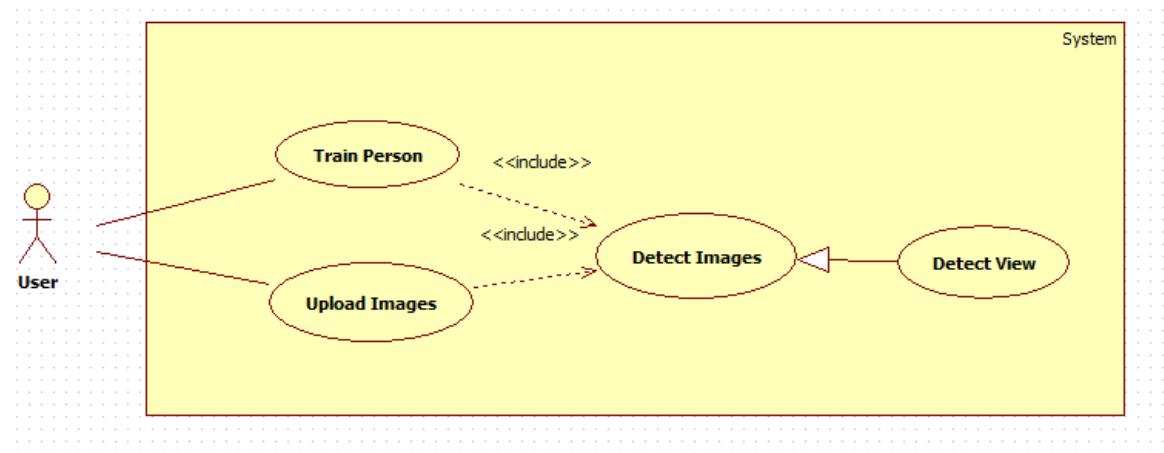
**Relationships:** N/A.**Business Rules:**

- Person Group Id is gotten from logged in user.
- Image will be compressed when uploaded to the system.
- System can only detect 10 persons 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 the 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 returns 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 looks 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****3.2.3.1.5 <User> Detect View****Figure 8: <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	04/15/2017	Priority	High

**Actor:**

- User

**Summary:**

- This use case allows the application to describe the view/landscape in the taken image by the user through 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 validates inputted information.
4		System calls Microsoft Cognitive Service to detect what is going on in the uploaded image. [Alternative 1] [Exception 2, 3]
5		System notifies 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 notification.

**Exceptions:**

No.	Cause	System Response
1	Unable to upload image	System response with an 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".

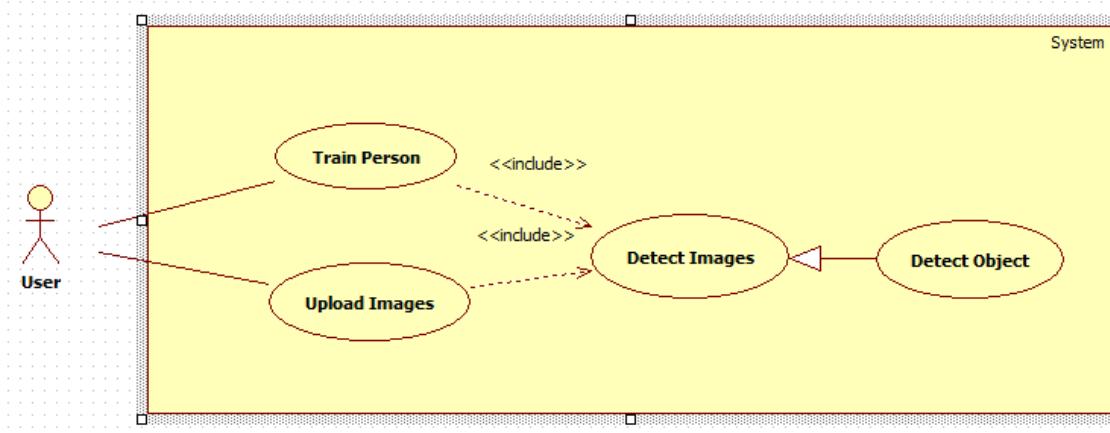
3	Rate limit is exceeded	System responses: “Đã đạt lượng truy cập tối đa. Vui lòng chờ trong vào ## giây để tiếp tục”.
---	------------------------	---

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

### 3.2.3.1.6 <User> Detect Object

**Figure 9: <User> Detect Object (UC\_US06)**

<b>USE CASE – UC_US06</b>			
<b>Use Case No.</b>	UC_US07	<b>Use Case Version</b>	1.0
<b>Use Case Name</b>	Detect Object		
<b>Author</b>	SonNVH		
<b>Date</b>	04/15/2017	<b>Priority</b>	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 describe 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> </ul>		

- **Fail:** System cannot describe 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 validates inputted information. [Exception 1]
4		System call service to identify object. [Alternative 1]
5		System notifies 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:

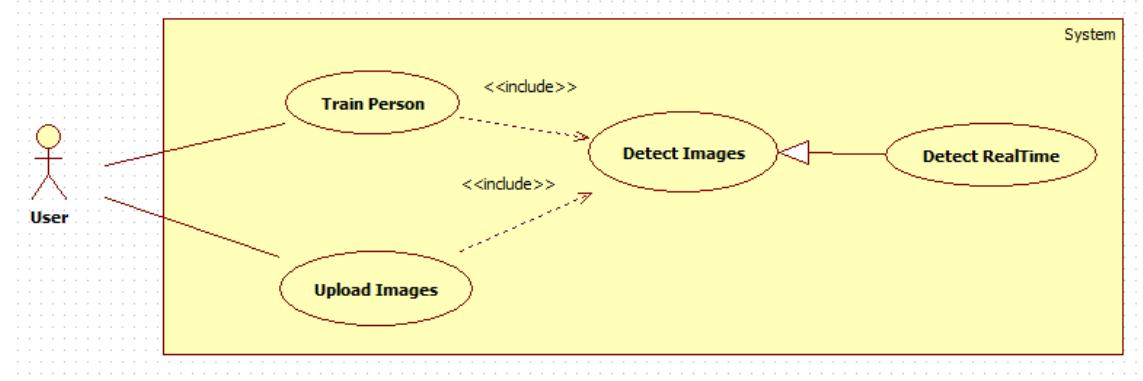
- Image will be compressed when uploaded to the system. The compressed image file size is between 20Kb – 70Kb.
- Identification result will be noticed to the user through voice notification.
- If the application cannot identify the object, a Log file will be created in the system and system Admin will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

ID	CREATED_DATE	IMAGE_URL	USER_ID	ACTIVE

- o **ID:** Log File 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 file is created: Datetime dd/MM/yyyy hh:mm:ss
- o **Active:** activation state of the log file: Bit

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

### 3.2.3.1.7 <User> Detect Real-time



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

USE CASE – UC_US07									
<b>Use Case No.</b>	UC_US07	<b>Use Case Version</b>	1.0						
<b>Use Case Name</b>	Detect Real-time								
<b>Author</b>	SonNVH								
<b>Date</b>	04/15/2017	<b>Priority</b>	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>	<ul style="list-style-type: none"> <li>- User sends command to begin streaming detecting process .</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> The application automatically identify any detected person.</li> <li>- <b>Fail:</b> Show response message.</li> </ul>								
<b>Main Success Scenario:</b>	<table border="1"> <thead> <tr> <th>Step</th><th>Actor Action</th><th>System Response</th></tr> </thead> <tbody> <tr> <td>1</td><td>User sends command to begin streaming detecting process. [Alternative 1]</td><td>System open device camera.</td></tr> </tbody> </table>			Step	Actor Action	System Response	1	User sends command to begin streaming detecting process. [Alternative 1]	System open device camera.
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 captures the human face and begin identifying process.
4		System notifies 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 continues 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:**

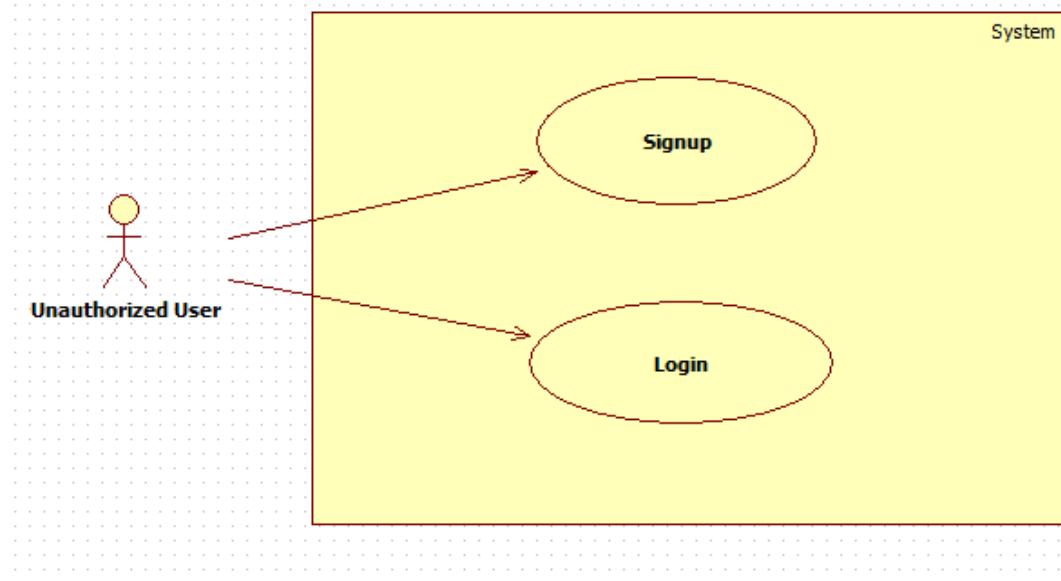
N/A.

**Business Rules:**

- The application must notify user the beginning of the streaming process through voice.
- The application will continuously detect for human faces in real time.
- When a human face 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.

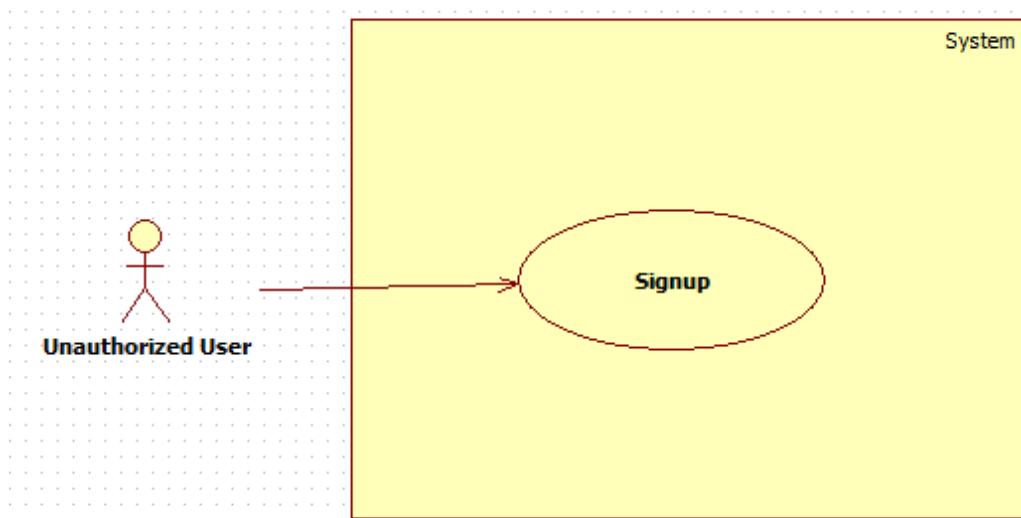
**Table 21: USE CASE – UC\_US07 - <User> Detect Real-time**

### 3.2.3.2 <Unauthorized User> Overview Use Case



*Figure 11: <Unauthorized User> Overview Use Case*

#### 3.2.3.2.1 <Unauthorized User> Signup



*Figure 12: <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 can 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.	<p>System requires identity information from Unauthorized user:</p> <ul style="list-style-type: none"> <li>- Full Name: free text input, required, length 5–50 characters.</li> <li>- Username: free text input, required, length 5–30 characters.</li> <li>- Password: free text input, required, length 5–30 characters.</li> <li>- Gender: select one of the options (Male, Female), required.</li> <li>- Phone number: free text input, required, length 10–11 digits.</li> <li>- Address: free text input, length 10–100 characters.</li> </ul>
2	Actor inputs information.	
3	Actor sends command to signup new account.	<p>System will sign up with validated inputted information.</p> <p>[Alternative 1]</p> <p>[Exception 1,2,3,4,5,6,7,8,9,10,11,12]</p>

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

Step.	Actor Action	System Response
1	Unauthorized user sends the cancel command.	System closes signup new account form.

**Exceptions:**

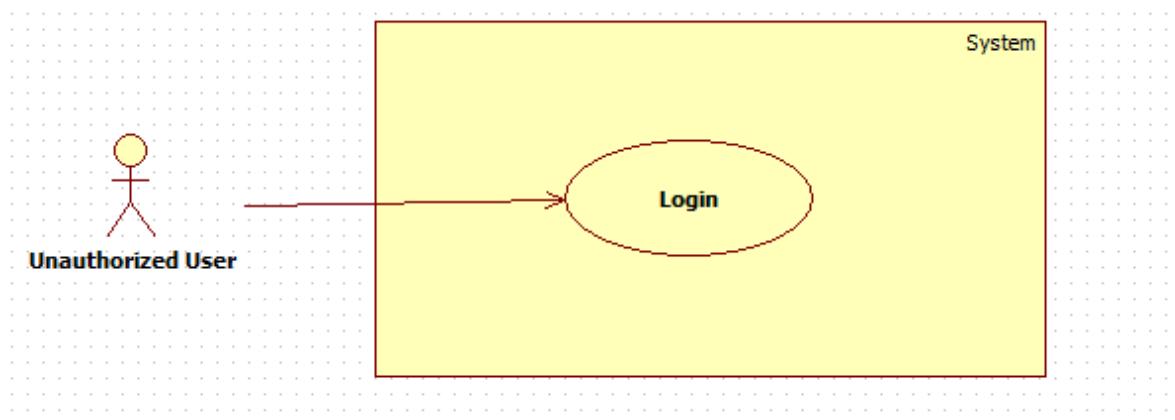
No.	Cause	System Response
1	Full name is empty.	System shows error message the “Tên không được để trống”.

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 “Email”.
- After signup in the system, unauthorized user will be redirected to login view.
- All newly signed up account will be assigned to “User” role.
- Each user will have a unique Person Group created for them after signed up successfully, with the Person Group ID is the same as the User ID.

**Table 22: USE CASE – UC\_UU01 - <Unauthorized User> Signup****3.2.3.2.2 <Unauthorized User> Login****Figure 13: <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						
<b>Actor:</b>	<ul style="list-style-type: none"> <li>- Unauthorized User</li> </ul>								
<b>Summary:</b>	<ul style="list-style-type: none"> <li>- This use case allows unauthorized user to login in the system and uses permission function.</li> </ul>								
<b>Goal:</b>	<ul style="list-style-type: none"> <li>- Actor can login in the system.</li> </ul>								
<b>Triggers:</b>	<ul style="list-style-type: none"> <li>- Actor sends the login command.</li> </ul>								
<b>Preconditions:</b>	<ul style="list-style-type: none"> <li>- Account must exist in the system.</li> </ul>								
<b>Post Conditions:</b>	<ul style="list-style-type: none"> <li>- <b>Success:</b> Actor login in the system.</li> <li>- <b>Fail:</b> Show error message.</li> </ul>								
<b>Main Success Scenario:</b>	<table border="1"> <thead> <tr> <th>Step</th><th>Actor Action</th><th>System Response</th></tr> </thead> <tbody> <tr> <td>1</td><td>Actor goes to login view.</td><td>System requires identity information from Actor:</td></tr> </tbody> </table>			Step	Actor Action	System Response	1	Actor goes to login view.	System requires identity information from Actor:
Step	Actor Action	System Response							
1	Actor goes to login view.	System requires identity information from Actor:							

		<ul style="list-style-type: none"> <li>- Username: free text input, required. Length 3- 100</li> <li>- Password: free text input, required. Length 6 - 100</li> </ul>
2	Actor inputs information.	
3	Actor sends command to login in the system.	Actor will login with their 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:**

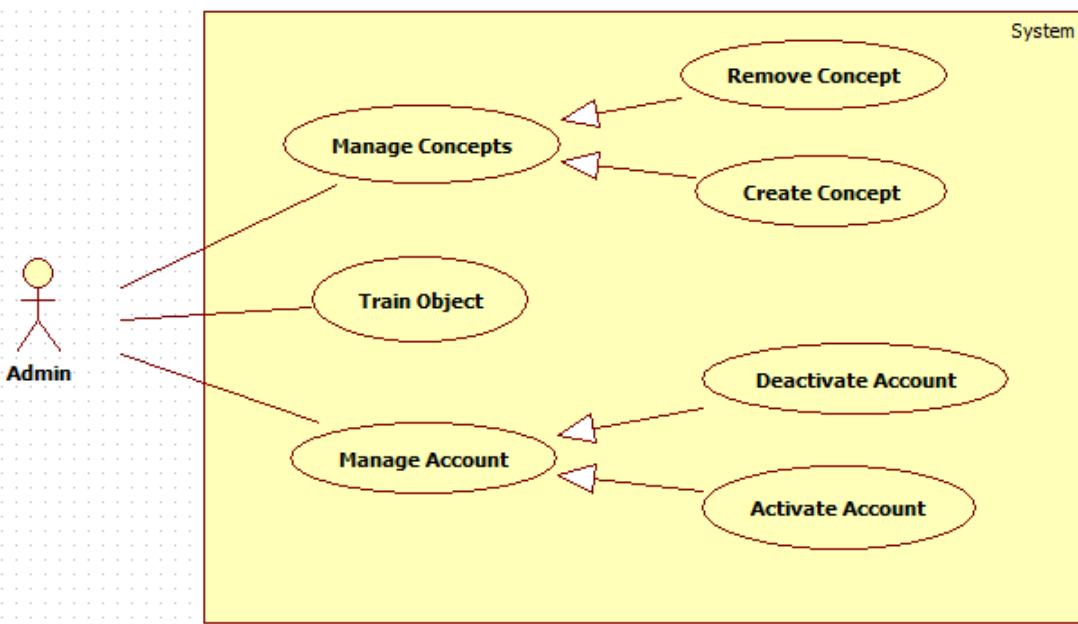
No.	Cause	System Response
1		System shows error message the “Hệ thống hiện đang bận. Vui lòng thử lại sau.” when the internet is lost.

**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.
  - o If role is “Admin”, the system will display to admin page.
  - o If role is “User”, the system will display to user view.
- If the account is Deactivated, user cannot login and a response message will be shown to user.

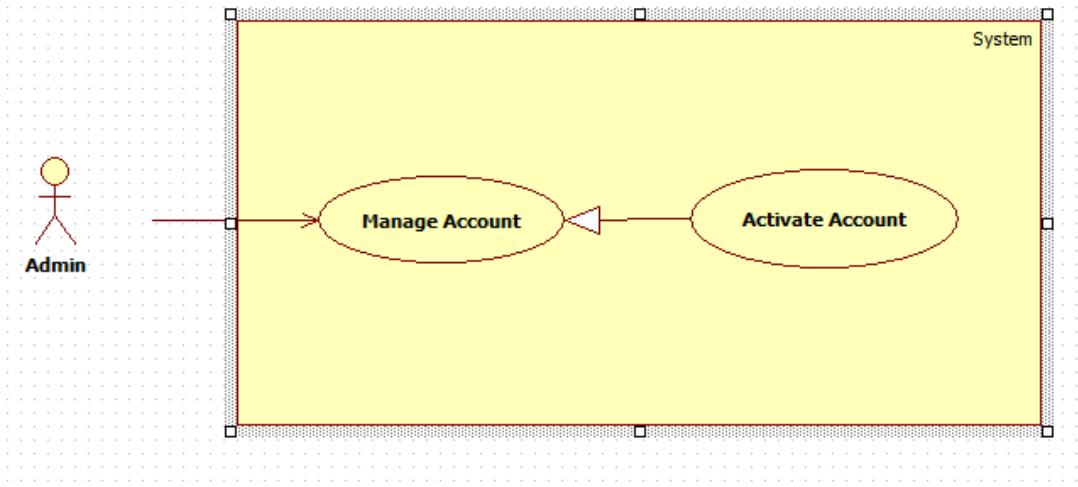
**Table 23: USE CASE - UC\_UU02 - <Unauthorized User >Login**

### 3.2.3.3 <Admin> Overview Use Case



*Figure 14: <Admin> Overview Use Case*

#### 3.2.3.3.1 <Admin> Activate Account



*Figure 15: <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	04/15/2017	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:**

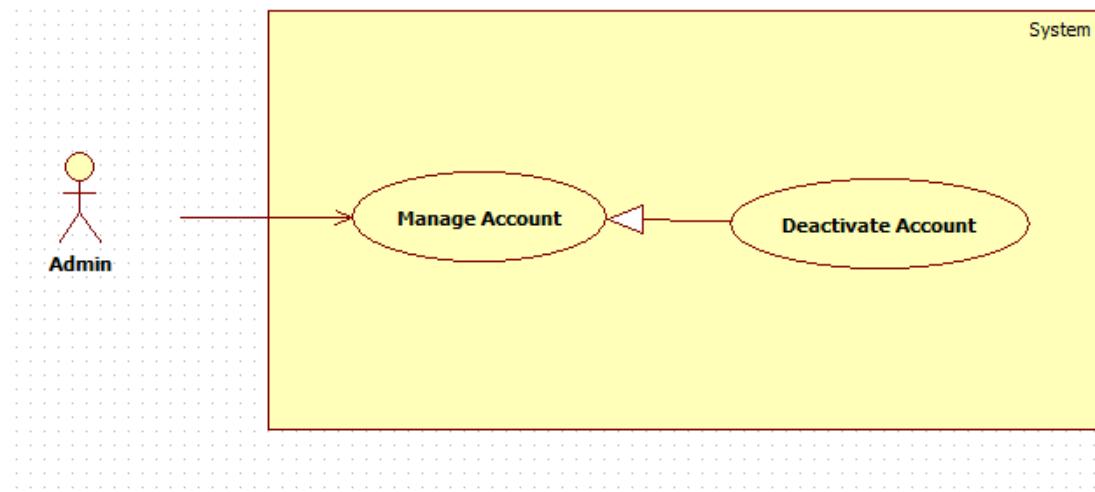
Step	Actor Action	System Response
1	Admin sends “Activate” command on a specific account	System activates the account and saves the account status to database.

**Alternative Scenario: N/A****Exceptions: N/A****Relationships: N/A.****Business Rules:**

- An account’s status can be “Activated” or “Deactivated”.
  - o “Activated” means account is activated and can be used to access the system.
  - o “Deactivated” means account is disabled and can’t be used to access the system.
- After being activated, the “Active” record of the user account in the system will be updated to “True”.

*Table 24: USE CASE - UC\_AD01 - <Admin >Activate Account*

### 3.2.3.3.2 <Admin> Deactivate Account



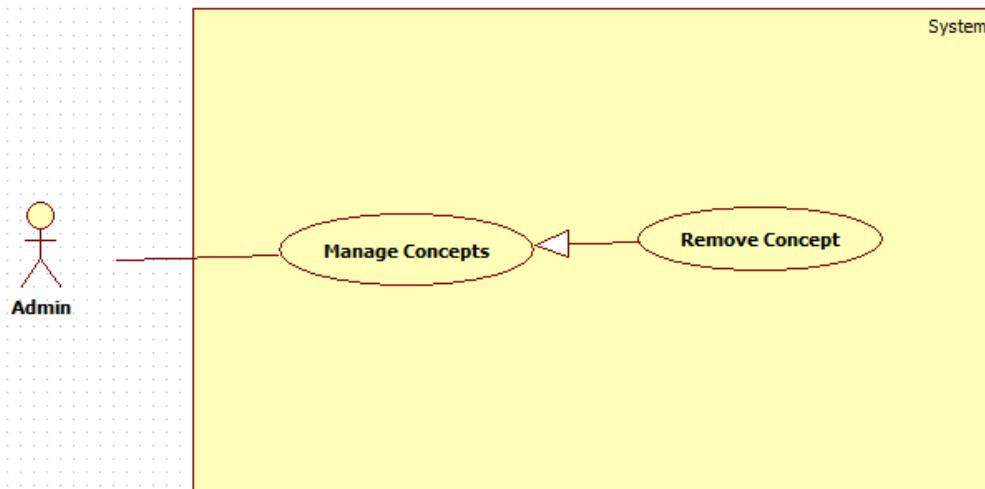
*Figure 16: <Admin> Deactivate Account (UC\_AD02)*

USE CASE - UC_AD02									
<b>Use Case No.</b>	UC_AD02	<b>Use Case Version</b>	1.0						
<b>Use Case Name</b>	Deactivate Account								
<b>Author</b>	QuanVH								
<b>Date</b>	04/15/2017	<b>Priority</b>	High						
<b>Actor:</b>	<ul style="list-style-type: none"> <li>- Admin</li> </ul>								
<b>Summary:</b>	<ul style="list-style-type: none"> <li>- This use case allows admin to deactivate account.</li> </ul>								
<b>Goal:</b>	<ul style="list-style-type: none"> <li>- An account is deactivated and can't access the system.</li> </ul>								
<b>Triggers:</b>	<ul style="list-style-type: none"> <li>- Admin send Deactivate account command</li> </ul>								
<b>Preconditions:</b>	<ul style="list-style-type: none"> <li>- Actor must log in the system with "Admin" role.</li> <li>- The selected account is activated.</li> </ul>								
<b>Post Conditions:</b>	<ul style="list-style-type: none"> <li>- <b>Success:</b> An account's status is updated in the database.</li> <li>- <b>Fail:</b> System displays the error message.</li> </ul>								
<b>Main Success Scenario:</b>	<table border="1"> <thead> <tr> <th>Step</th><th>Actor Action</th><th>System Response</th></tr> </thead> <tbody> <tr> <td>1</td><td>Admin sends "Deactivate" command on a specific account</td><td></td></tr> </tbody> </table>			Step	Actor Action	System Response	1	Admin sends "Deactivate" command on a specific account	
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.
<b>Alternative Scenario:</b> N/A		
<b>Exceptions:</b> N/A		
<b>Relationships:</b> N/A.		
<b>Business Rules:</b> <ul style="list-style-type: none"> <li>- An account's status can be "Activated" or "Deactivated".             <ul style="list-style-type: none"> <li>o "Activated" means account is activated and can be used to access the system.</li> <li>o "Deactivated" means account is disabled and can't be used to access the system.</li> </ul> </li> <li>- After being activated, the "Active" record of the user account in the system will be updated to "False".</li> </ul>		

*Table 25: USE CASE – UC\_AD02 - <Admin>Deactivate Account*

### 3.2.3.3.3 <Admin> Remove Concept

*Figure 17: <Admin> Remove Concept (UC\_AD03)*

USE CASE – UC_AD03			
<b>Use Case No.</b>	UC_AD03	<b>Use Case Version</b>	1.0
<b>Use Case Name</b>	Remove Concept		
<b>Author</b>	ThanhPT		
<b>Date</b>	20/03/2017	<b>Priority</b>	Normal
<b>Actor:</b>	<ul style="list-style-type: none"> <li>- Admin</li> </ul>		
<b>Summary:</b>	<ul style="list-style-type: none"> <li>- This use case allows admin remove concept in the system.</li> <li>- </li> </ul>		

**Goal:**

- Concept is removed out off 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:** System will send success message when concept is removed successfully.
- **Fail:** System will show error message.

**Main Success Scenario:**

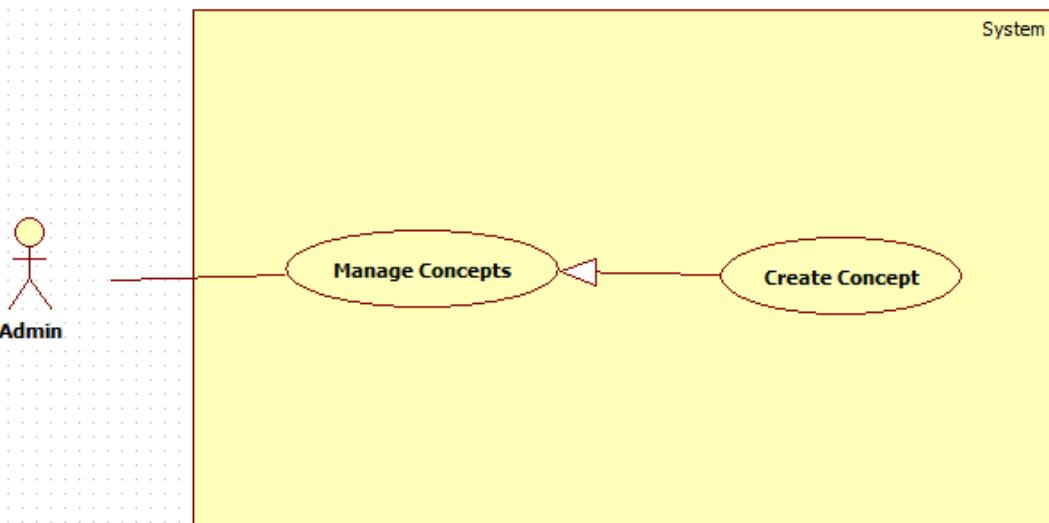
Step	Actor Action	System Response
1	Admin goes to concept view.	System show list of concept includes image, name, description.
2	Admin chooses a concept.	
3	Admin sends command to remove concept.	System removes chosen concept.

**Alternative Scenario:** N/A**Exceptions:** N/A**Relationships:** N/A.**Business Rules:**

- The Active record of the chosen Concept will be updated to False after successfully Deactivated.
- Concept ID must be unique.
- The Concept's Images must be also deactivated when removing the Concept.

*Table 26: USE CASE - UC\_AD03 - <Admin> Remove Concept*

### 3.2.3.3.4 <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>	<ul style="list-style-type: none"> <li>- Admin</li> </ul>								
<b>Summary:</b>	<ul style="list-style-type: none"> <li>- This use case allows admin create new concept to the system predict object and can recognize object.</li> </ul>								
<b>Goal:</b>	<ul style="list-style-type: none"> <li>- A new concept is created into the system.</li> </ul>								
<b>Triggers:</b>	<ul style="list-style-type: none"> <li>- Admin sends create new concept command.</li> </ul>								
<b>Preconditions:</b>	<ul style="list-style-type: none"> <li>- Account must be login with admin role.</li> </ul>								
<b>Post Conditions:</b>	<ul style="list-style-type: none"> <li>- <b>Success:</b> System will send success message when concept is created successfully.</li> <li>- <b>Fail:</b> System will show error message.</li> </ul>								
<b>Main Success Scenario:</b>	<table border="1"> <thead> <tr> <th>Step</th><th>Actor Action</th><th>System Response</th></tr> </thead> <tbody> <tr> <td>1</td><td>Admin goes to create new concept view.</td><td>           System requires information from actor:           <ul style="list-style-type: none"> <li>- Concept Name: free text input, required. The size limit is 100 characters.</li> </ul> </td></tr> </tbody> </table>			Step	Actor Action	System Response	1	Admin goes to create new concept view.	System requires information from actor: <ul style="list-style-type: none"> <li>- Concept Name: free text input, required. The size limit is 100 characters.</li> </ul>
Step	Actor Action	System Response							
1	Admin goes to create new concept view.	System requires information from actor: <ul style="list-style-type: none"> <li>- Concept Name: free text input, required. The size limit is 100 characters.</li> </ul>							

		<ul style="list-style-type: none"> <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 creates new concept and redirects to view all concept view.

**Alternative Scenario:** N/A**Exceptions:**

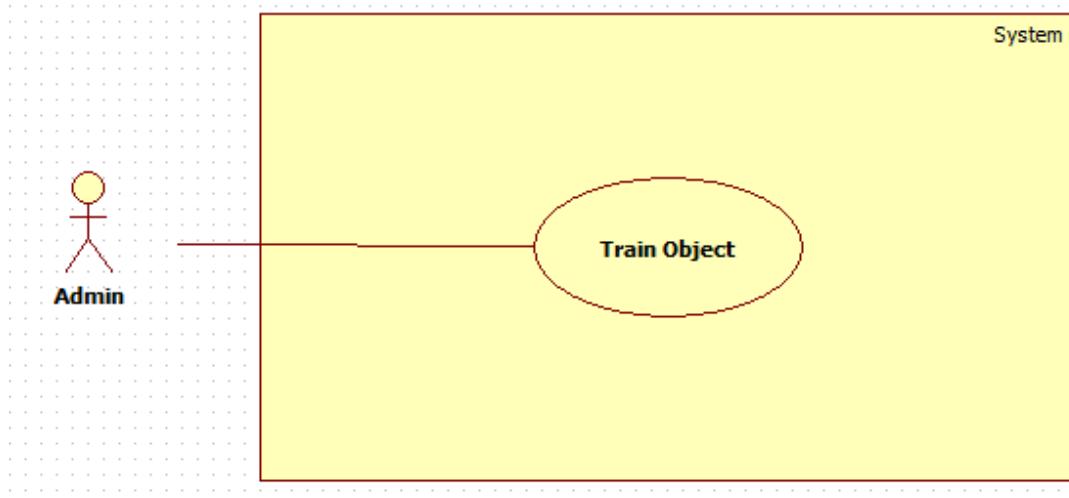
No.	Cause	System Response
1	Concept name is empty.	System shows error message: "Tên đồ vật không được để trống".
2	Images are empty.	System shows error message: "Vui lòng chọn hình ảnh".
3	Invalid image.	System shows error message: "Vui lòng chỉ chọn hình".
4	Description is empty	System shows error message: "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 created Concept.
- Support image formats are: JPG, PNG, JPEG.

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

### 3.2.3.3.5 <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>	<ul style="list-style-type: none"> <li>- Admin</li> </ul>											
<b>Summary:</b>	<ul style="list-style-type: none"> <li>- This use case allows admin train object by adding an image into concept from the log file.</li> </ul>											
<b>Goal:</b>	<ul style="list-style-type: none"> <li>- The system can recognize object has been added to the system.</li> </ul>											
<b>Triggers:</b>	<ul style="list-style-type: none"> <li>- Admin sends trains object command.</li> </ul>											
<b>Preconditions:</b>	<ul style="list-style-type: none"> <li>- Account must be login with admin role.</li> <li>- The log must exist.</li> </ul>											
<b>Post Conditions:</b>	<ul style="list-style-type: none"> <li>- <b>Success:</b> The system can recognize the trained object.</li> <li>- <b>Fail:</b> System cannot recognize the newly added object. System shows error response.</li> </ul>											
<b>Main Success Scenario:</b>	<table border="1"> <thead> <tr> <th>Step</th><th>Actor Action</th><th>System Response</th></tr> </thead> <tbody> <tr> <td>1</td><td>Admin goes to logs view.</td><td>System shows all logs.</td></tr> <tr> <td></td><td>Admin chooses log to train.</td><td></td></tr> </tbody> </table>			Step	Actor Action	System Response	1	Admin goes to logs view.	System shows all logs.		Admin chooses log to train.	
Step	Actor Action	System Response										
1	Admin goes to logs view.	System shows all logs.										
	Admin chooses log to train.											

		System show table includes: image, all name of concept had existed.
2	Admin chooses 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 redirects to create new concept page.
2	Admin goes to create new concept view.	System requires 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 redirects 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:**

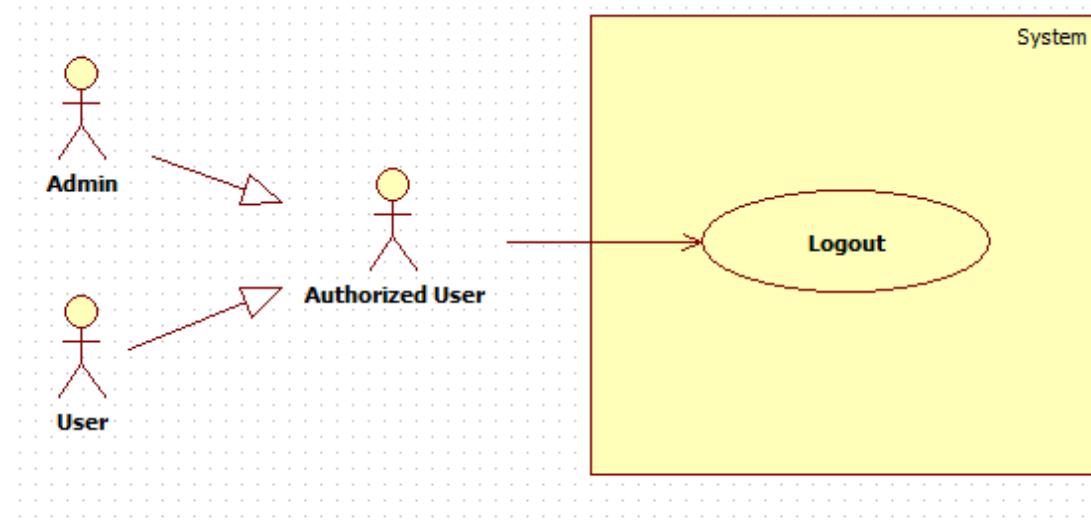
- Support image formats are JPG, PNG, JPEG.
- After object is trained successfully, Active state will be updated to “False”.
- Log structure in database:

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 already exists, the inputted image will be updated to that Concept.

*Table 28: USE CASE – UC\_AD05 - <Admin > Train Object*

### 3.2.3.4 <Authorized User> Log out



*Figure 20: <Authorized User> Logout (UC\_AU01)*

<b>USE CASE – UC_AU01</b>					
<b>Use Case No.</b>	<b>UC_AU01</b>	<b>Use Case Version</b>	1.0		
<b>Use Case Name</b>	Log out				
<b>Author</b>	QuanVH				
<b>Date</b>	04/15/2017	<b>Priority</b>	Normal		
<b>Actor:</b>	<ul style="list-style-type: none"> <li>- Authorized User</li> </ul>				
<b>Summary:</b>	<ul style="list-style-type: none"> <li>- This use case helps Authorized User to sign out of their account.</li> </ul>				
<b>Goal:</b>	<ul style="list-style-type: none"> <li>- Authorized User can sign out of the system.</li> </ul>				
<b>Triggers:</b>	<ul style="list-style-type: none"> <li>- Authorized User sends Logout command.</li> </ul>				
<b>Preconditions:</b>	<ul style="list-style-type: none"> <li>- Actor logged in system with role “Authorized User”.</li> </ul>				
<b>Post Conditions:</b>	<ul style="list-style-type: none"> <li>- <b>Success:</b> Authorized User is logged out.</li> <li>- <b>Fail:</b> System will return error response message.</li> </ul>				
<b>Main Success Scenario:</b>					
Step	Actor Action	System Response			
1	Authorized User sends logout command.	System clears the current session of the user then redirects user to Login page.			
<b>Alternative Scenario: N/A</b>					
<b>Exceptions: N/A</b>					
<b>Relationships: N/A.</b>					
<b>Business Rules:</b>					
<ul style="list-style-type: none"> <li>- After logging out of the system, user is no longer has permission to access any authorized function of the system.</li> <li>- After login, user will be redirected to Login Page.</li> </ul>					

*Table 29: USE CASE – UC\_AU01 - <Authorized User >Logout*

### **3.3 Software System Attribute**

#### **3.3.1 Usability**

##### **3.3.1.1 Graphic User Interface**

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

##### **3.3.1.2 Usability**

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

#### **3.3.2 Reliability**

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

#### **3.3.3 Availability**

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

#### **3.3.4 Security**

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

#### **3.3.5 Maintainability**

- The system is divided into separated modules.

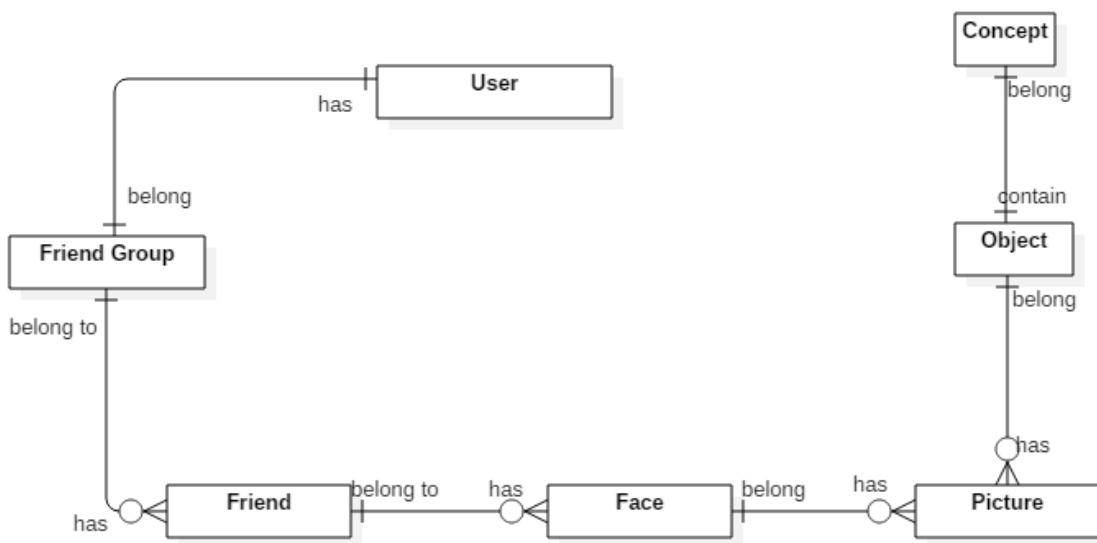
#### **3.3.6 Portability**

- 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.

#### **3.3.7 Performance**

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

### 3.4 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*

## 4 Software Design Description

### 4.1 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 a 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

### 4.2 System Architectural Design

Our application is designed based on MVC Model with 3 interconnected parts:

- Controller: 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.

Advantages and Disadvantages:

- Advantages: The MVC model demonstrates professionalism in programming and design analysis. It's divided into independent components to help develop applications faster, simpler, easier upgrades and maintenance.
- Disadvantages: Knowledge on multiple technologies is required

#### 4.2.1 Mobile Application Architecture Description

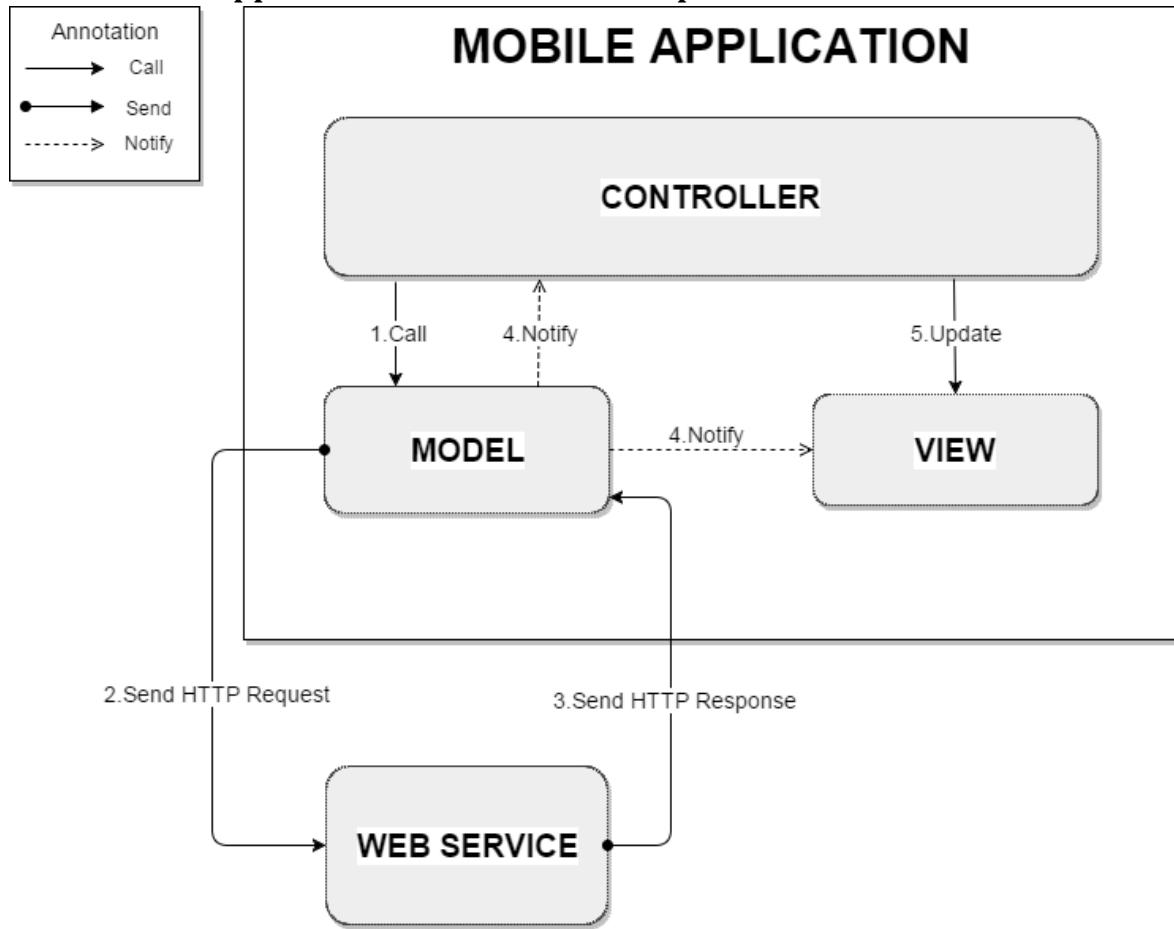


Figure 22: Mobile Application Architecture Description

#### 4.2.2 Web Service Architecture Description

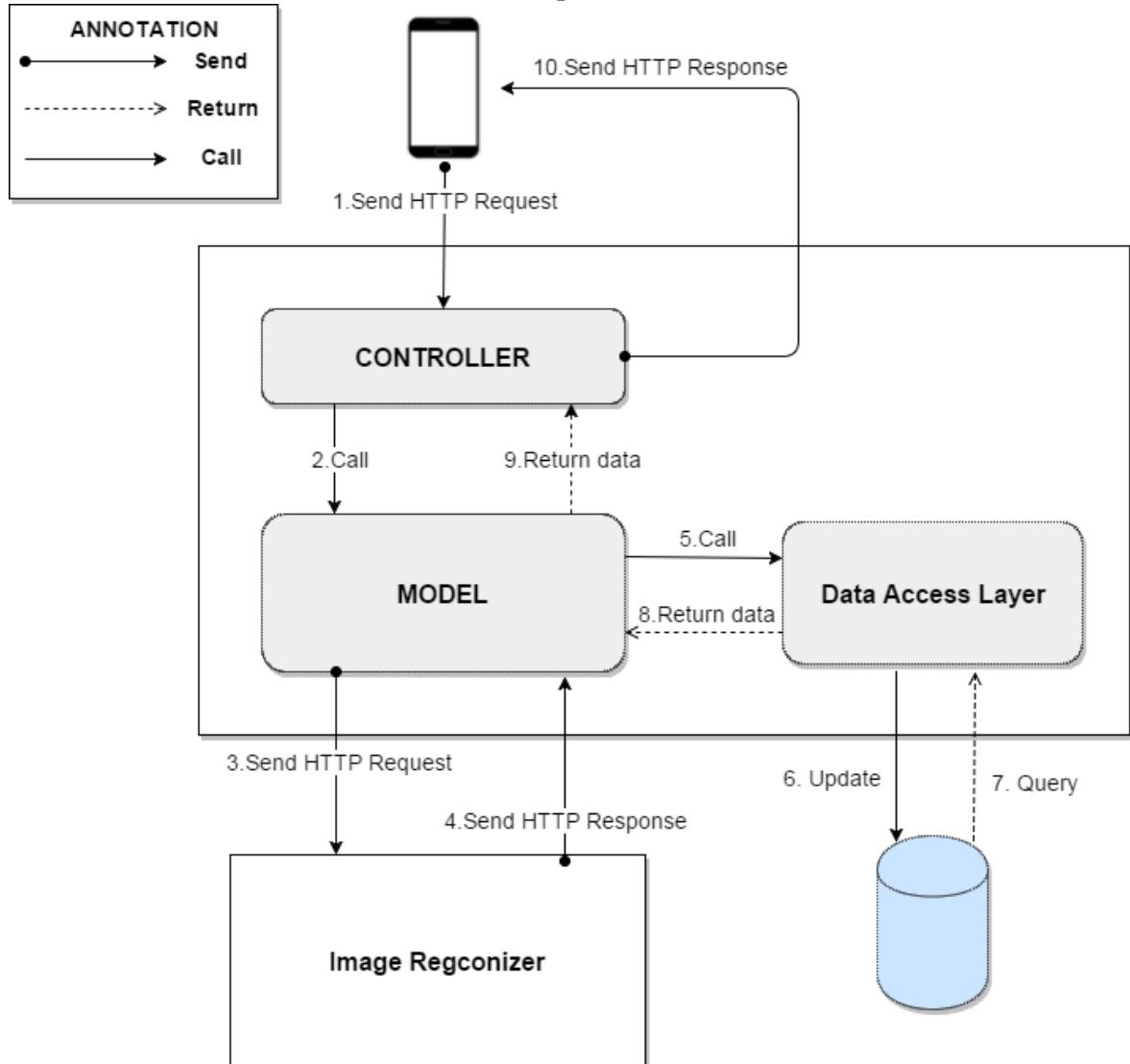


Figure 23: Web Service Architecture Description

#### 4.2.3 Web Application Architecture Description

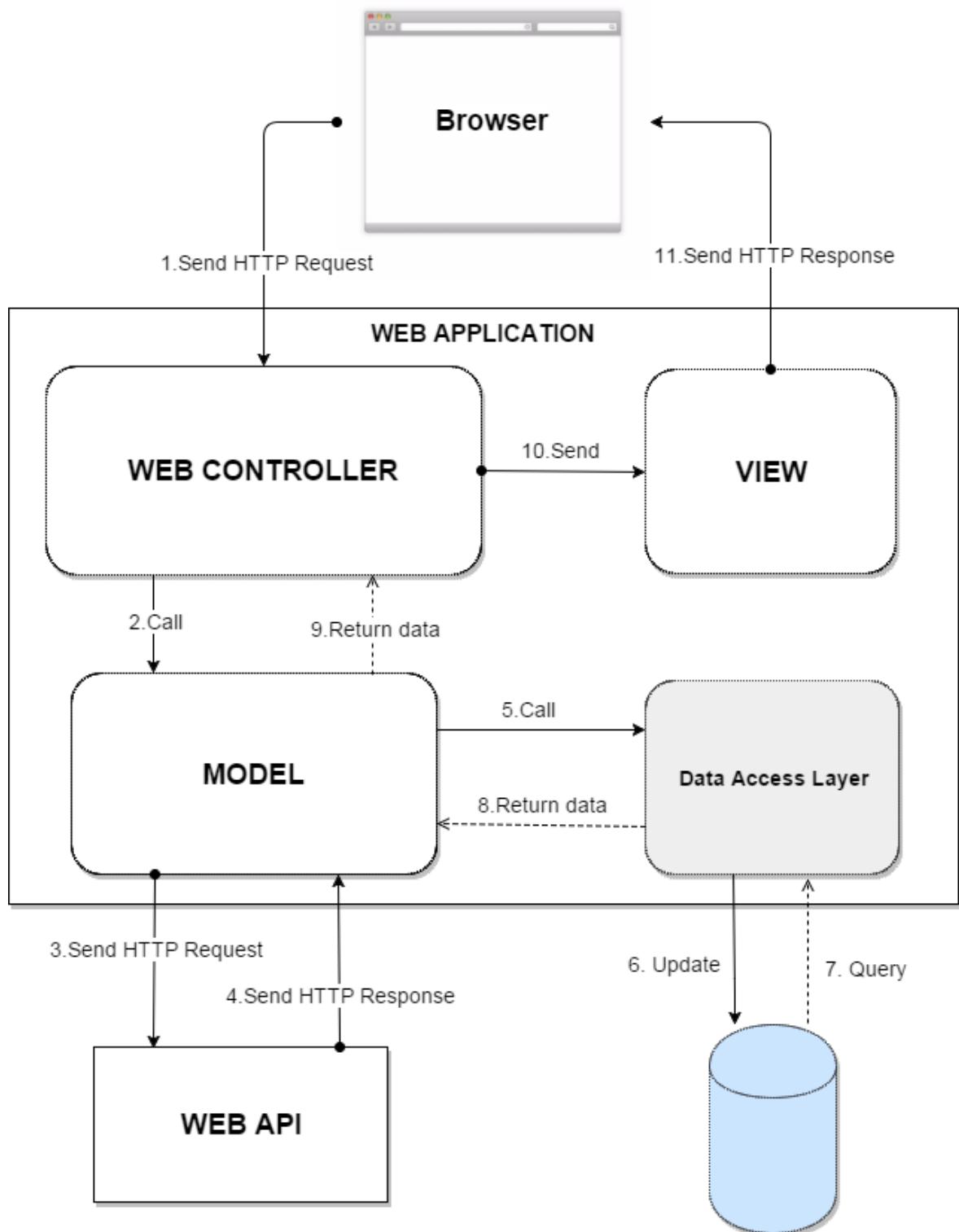


Figure 24: Web Application Architecture Description

### 4.3 Component Diagram

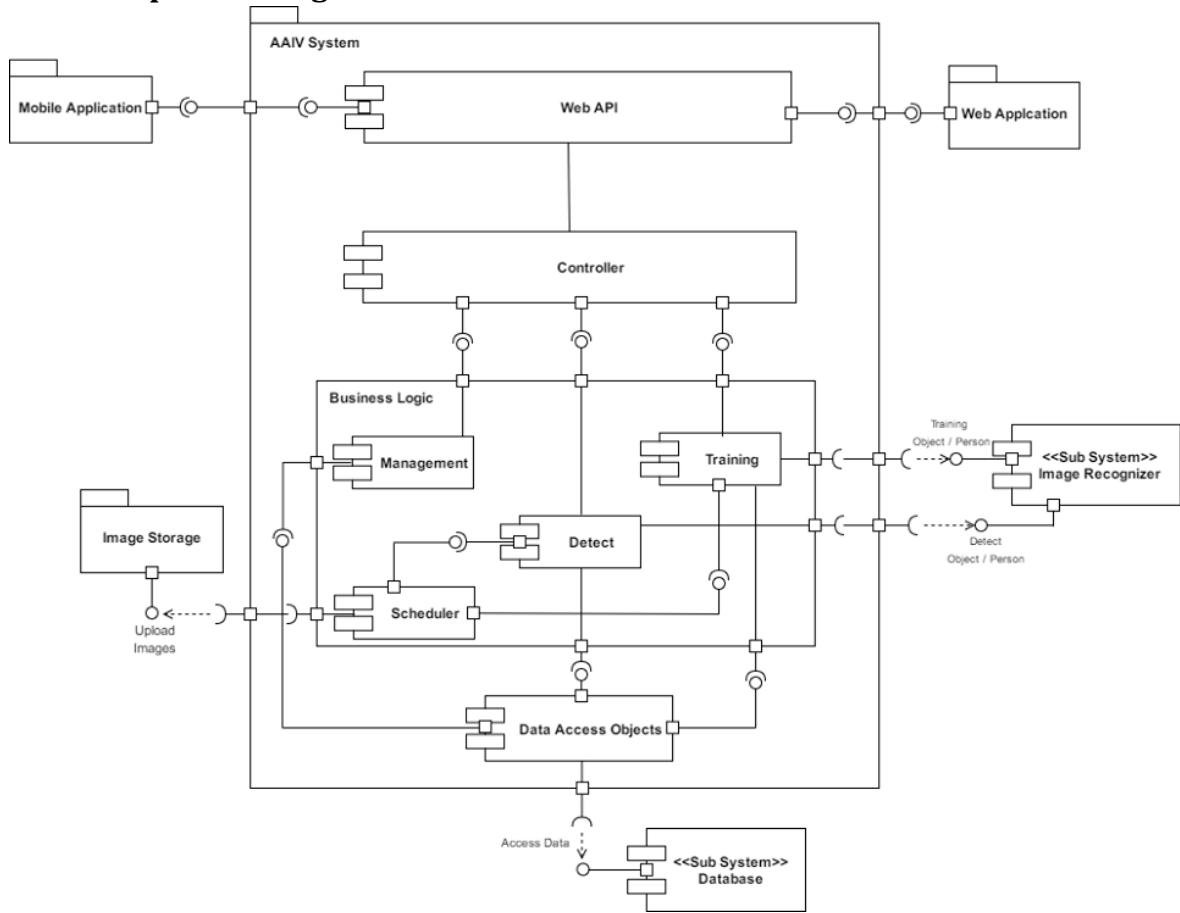


Figure 25: Component Diagram

Name	Description
AAIV System	Automatic Alternative Image Recognition to Voice System
Web API	Component is used to provide API for the 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

Table 31: Component Dictionary

## 4.4 Detailed Description

### 4.4.1 Class Diagram

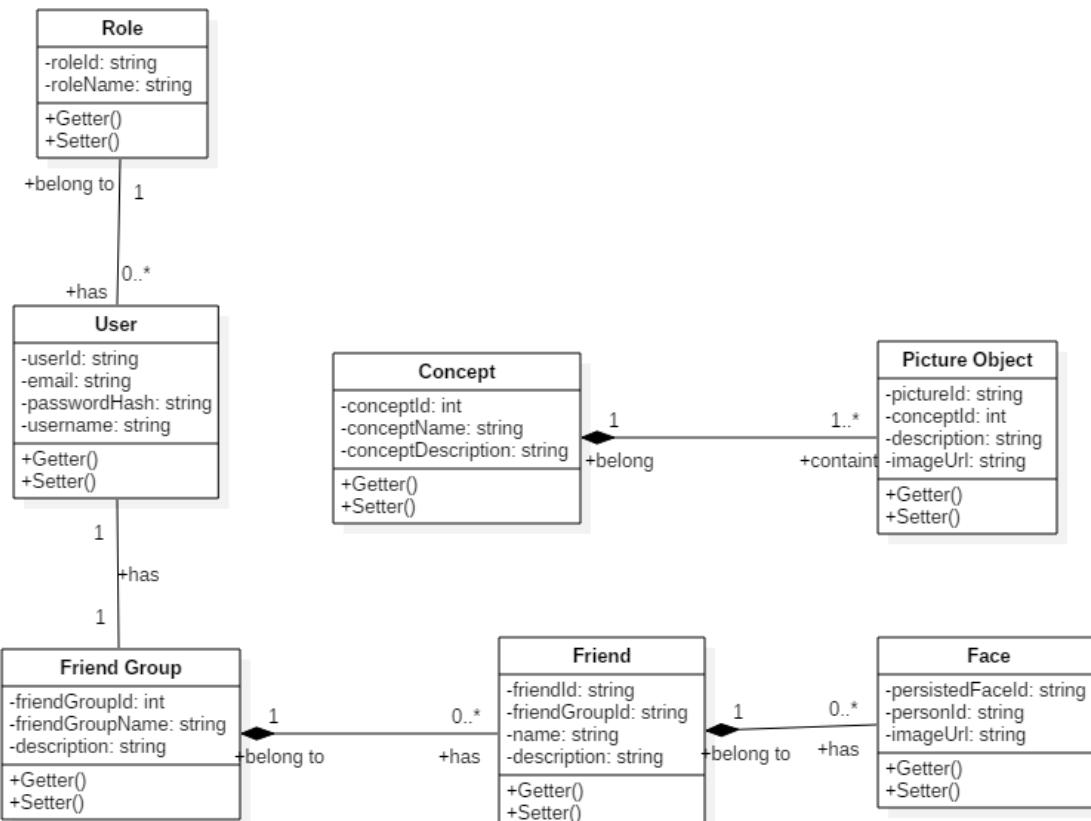


Figure 26: Class Diagram

Class dictionary: Describe Class		
Class Name	Mapping class with conceptual diagram	Description
Role	N/A	Not exist in the 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*

#### 4.4.2 Class Diagram Explanation

##### 4.4.2.1 Role

###### Attributes

Attribute	Type	Visibility	Description
roleId	String	Private	Unique id of role
roleName	String	Private	Name of role

*Table 33: Role class Attributes explanation*

###### Methods:

Method	Type	Visibility	Description
Getter	Attribute type	Public	Get attribute value
Setter	Void	Public	Set value of attribute

*Table 34: Role Class methods explanation*

##### 4.4.2.2 User

###### Attributes:

Attribute	Type	Visibility	Description
userId	String	Private	Unique id of user
Email	String	Private	Email of user
passwordHash	String	Private	Password of user
Username	String	Private	Username of user

*Table 35: User Class attributes explanation*

###### Methods:

Method	Type	Visibility	Description
Getter	Attribute type	Public	Get attribute value
Setter	Void	Public	Set value of attribute

*Table 36: User class methods explanation*

#### 4.4.2.3 Friend Group

**Attributes:**

Attribute	Type	Visibility	Description
friendGroupId	Int	Private	Unique id of friend group
friendGroupName	String	Private	Name of friend group
Description	string	Private	Description of friend group

*Table 37: Friend Group class attributes explanation*

**Methods:**

Method	Type	Visibility	Description
Getter	Attribute type	Public	Get attribute value
Setter	Void	Public	Set value of attribute

*Table 38: Friend Group class methods explanation*

#### 4.4.2.4 Concept

**Attributes:**

Attribute	Type	Visibility	Description
conceptId	Int	Private	Unique id of concept
conceptName	String	Private	Name of concept
conceptDescription	String	Private	Description of concept

*Table 39: Concept class attributes explanation*

**Methods:**

Method	Type	Visibility	Description
Getter	Attribute type	Public	Get attribute value
Setter	Void	Public	Set value of attribute

*Table 40: Concept class methods explanation*

#### 4.4.2.5 Picture

**Attributes:**

Attribute	Type	Visibility	Description
pictureId	String	Private	Unique id of picture
conceptId	Int	Private	Id of concept contain picture
Description	String	Private	Description of picture
imageUrl	string	Private	URL of the image

*Table 41: Picture class attributes explanation***Methods:**

<b>Method</b>	<b>Type</b>	<b>Visibility</b>	<b>Description</b>
Getter	Attribute type	Public	Get attribute value
Setter	Void	Public	Set value of attribute

*Table 42: Picture class methods explanation***4.4.2.6 Friend****Attributes:**

<b>Attribute</b>	<b>Type</b>	<b>Visibility</b>	<b>Description</b>
friendId	String	Private	Unique id of friend
friendGroupId	String	Private	Id of friend group
Name	String	Private	Name of friend
Description	string	Private	Description of friend

*Table 43: Friend class attributes explanation***Methods:**

<b>Method</b>	<b>Type</b>	<b>Visibility</b>	<b>Description</b>
Getter	Attribute type	Public	Get attribute value
Setter	Void	Public	Set value of attribute

*Table 44: Friend class methods explanation***4.4.2.7 Face****Attributes:**

<b>Attribute</b>	<b>Type</b>	<b>Visibility</b>	<b>Description</b>
persistedFaceId	String	Private	Unique id of face
personId	String	Private	If of person contain this face
imageUrl	String	Private	Image url contain this face

*Table 45: Face class attributes explanation***Methods:**

<b>Method</b>	<b>Type</b>	<b>Visibility</b>	<b>Description</b>
Getter	Attribute type	Public	Get attribute value
Setter	Void	Public	Set value of attribute

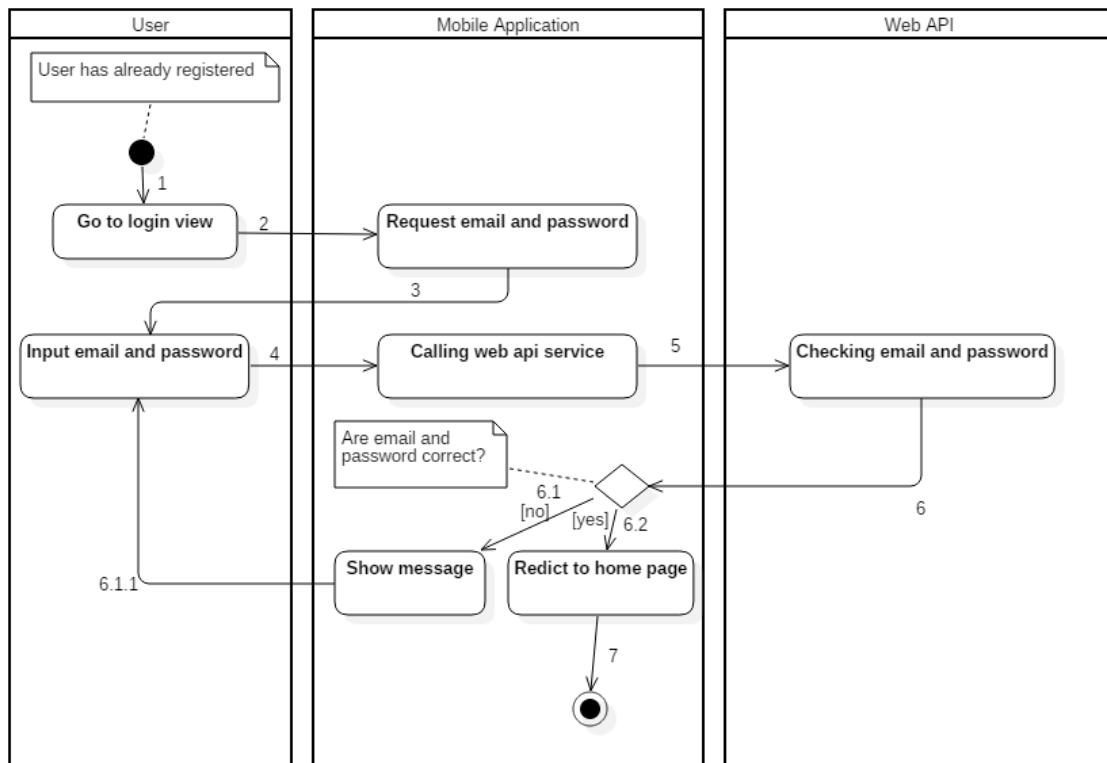
*Table 46: Face class methods explanation*

### 4.4.3 Interaction Diagram

#### 4.4.3.1 Mobile Application

##### 4.4.3.1.1 <Unauthorized User> Login

Summary: This diagram shows the process of unauthorized user login into the system using mobile application

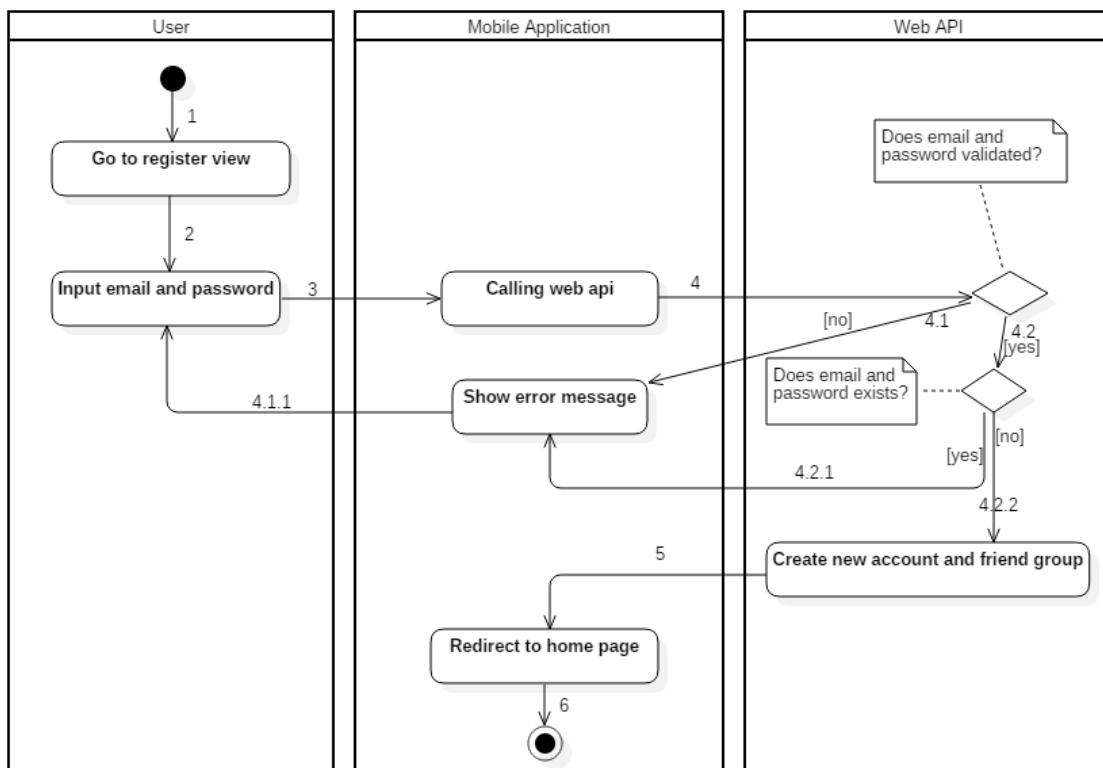


*Figure 27: Activity Diagram - <Unauthorized User> Login*

1. Start	2. Go to Login view	3. Request email and password
4. Input email and password	5. Calling web API service	6. Check email 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.

##### 4.4.3.1.2 <Unauthorized User> Register

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

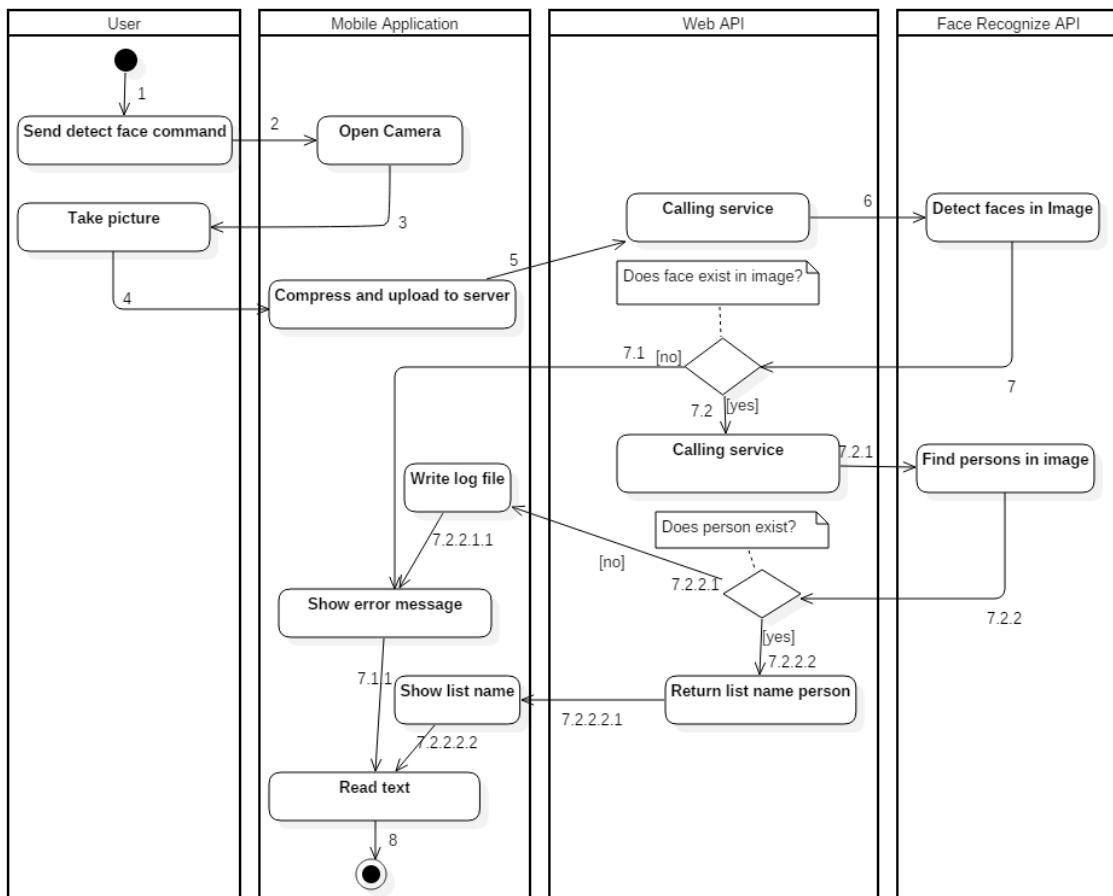


**Figure 28: Activity Diagram - <Unauthorized User> Register**

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 a new account with data user input also create friend group with id is userid.	6. Redirect to home page	

#### 4.4.3.1.3 <Authorized User> Detect Face

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

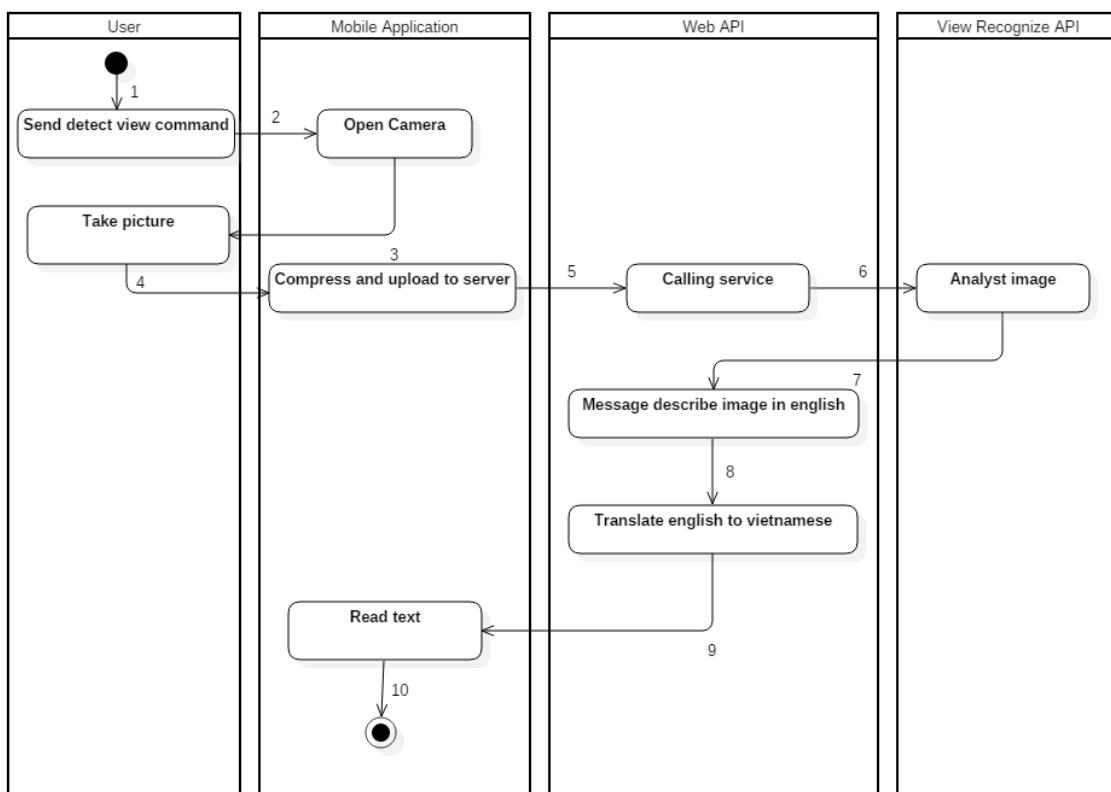


*Figure 29: Activity Diagram - <Authorized User> Detect Face*

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.

#### 4.4.3.1.4 <Authorized User> Detect View

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

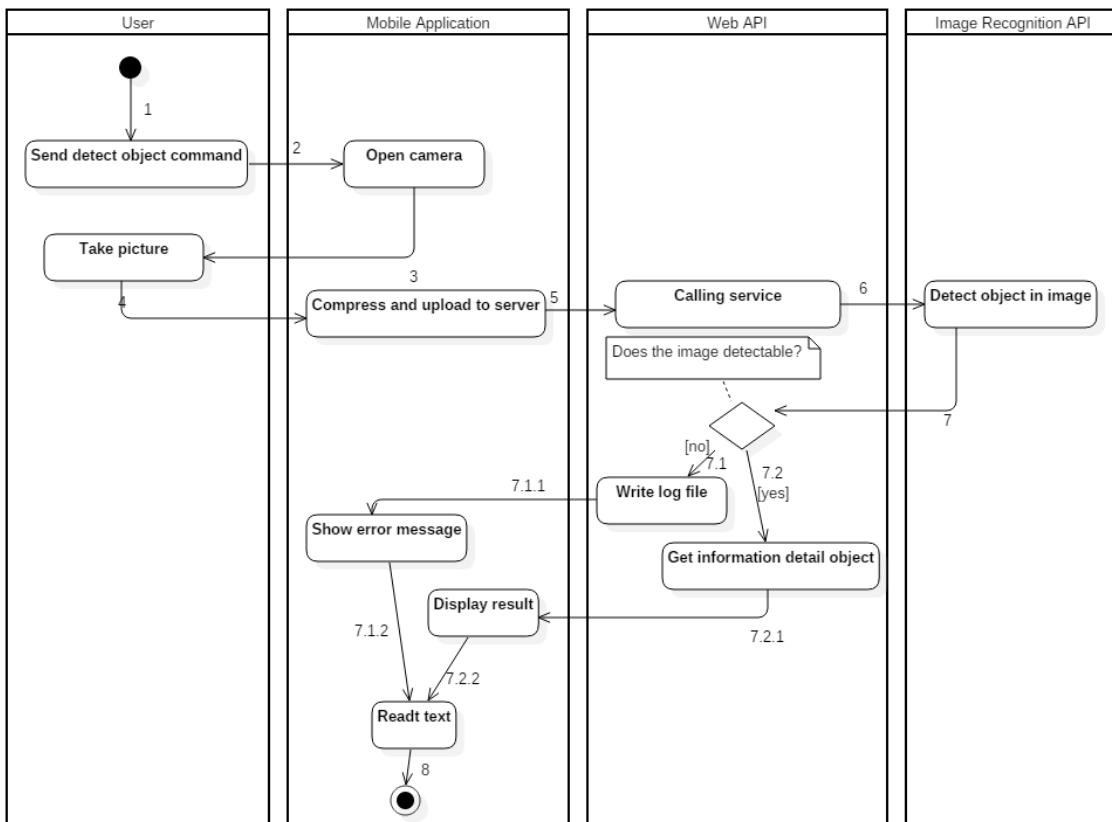


*Figure 30: Activity Diagram - <Authorized User> Detect View*

1. Start.	2. User sends detect view command.	3. Open camera.
4. User sends command to take picture.	5. Compress image and upload to server.	6. Call service to detect view.
7. Analyst image	8. Return message describes image in English.	9. Translate English to Vietnamese.
10. Text to speech		

#### 4.4.3.1.5 <Authorized User> Detect Object

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

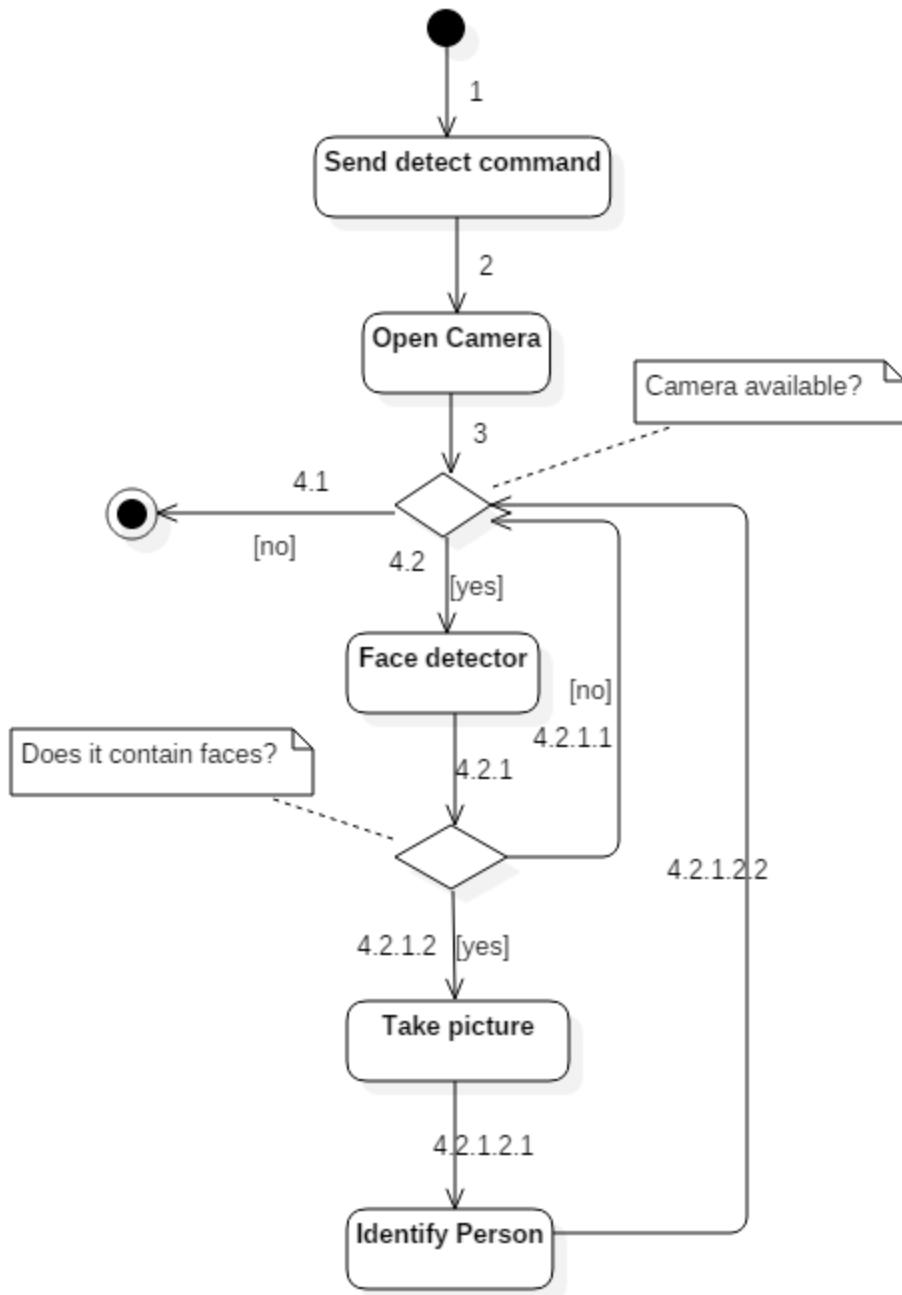


**Figure 31: Activity Diagram - <Authorized User> Detect Object**

1. Start.	2. User send detect object command.	3. Open camera.
4. User sends 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.	

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

Summary: This diagram shows 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		
-------------------------------------	--	--

#### 4.4.3.2 Web Application

##### 4.4.3.2.1 <Admin> Create Concept

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

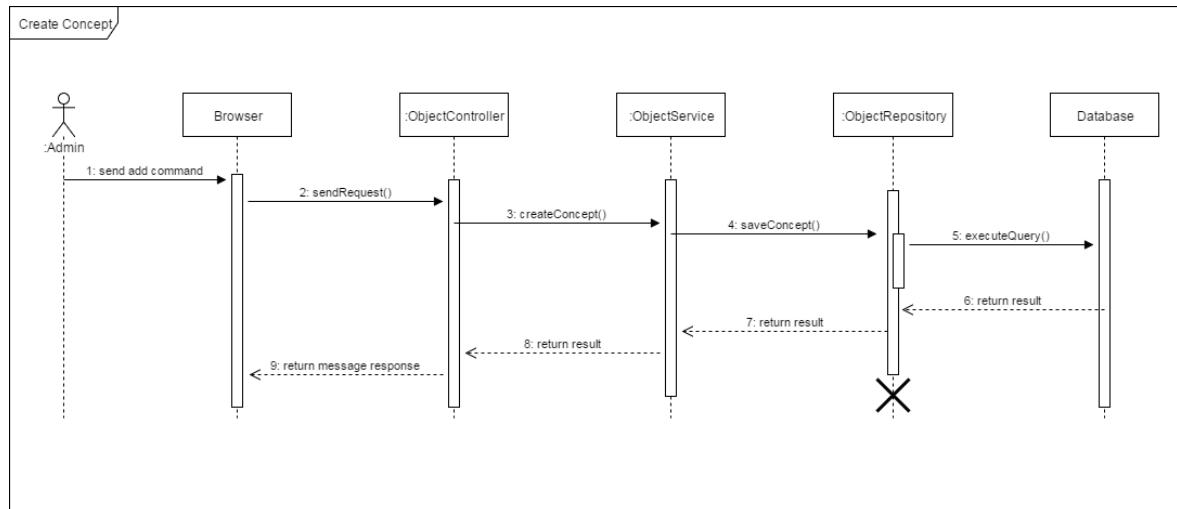
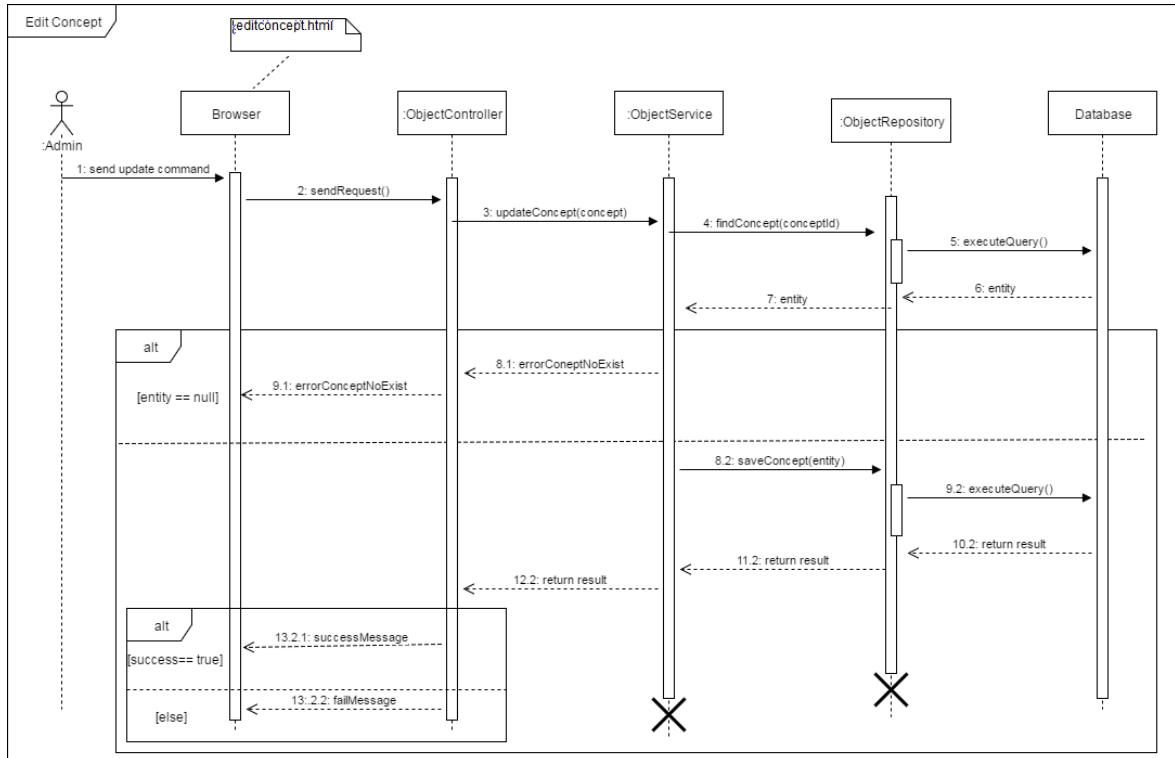


Figure 33: Sequence Diagram - <Admin > Create Concept

##### 4.4.3.2.2 <Admin> Edit Concept

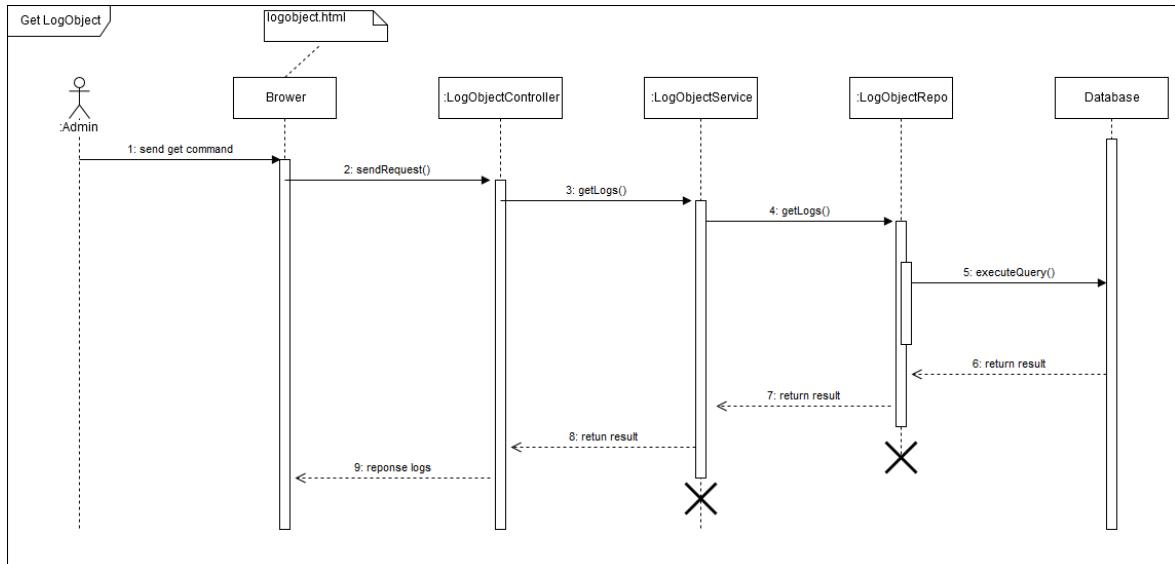
Summary: This diagram show the process of Admin edit concept using the web application.



**Figure 34: Sequence Diagram - <Admin >Edit Concept**

#### 4.4.3.2.3 <Admin> Get Log Object

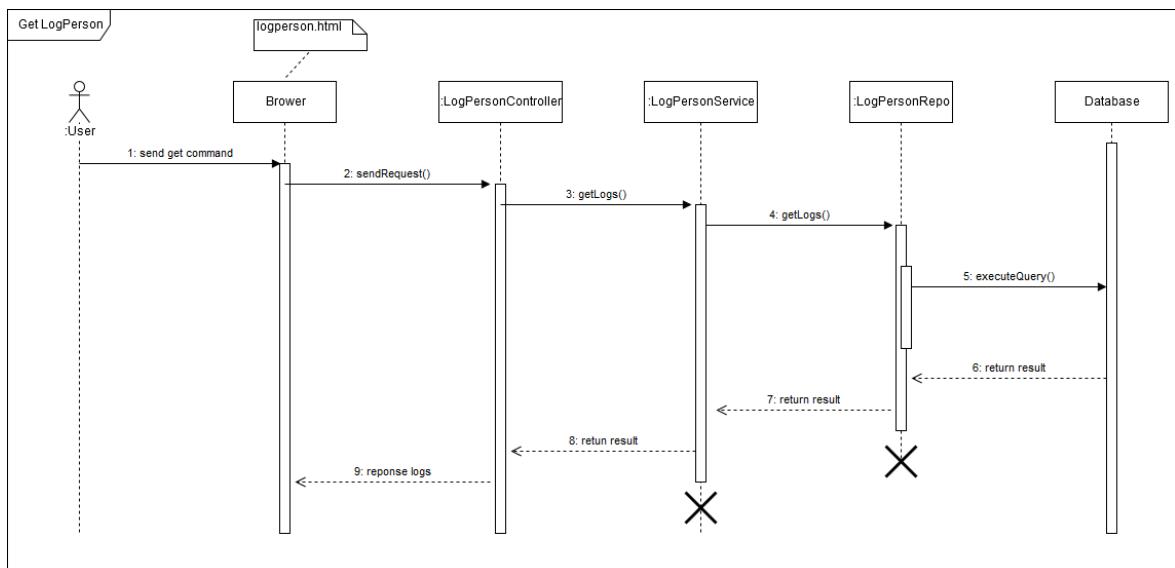
Summary: This diagram shows the process of Admin getting logs using the web application.



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

#### 4.4.3.2.4 <User> Get Log Person

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



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

## 4.5 Interface

### 4.5.1 Component interface

#### 4.5.1.1 Modula of Object API

Signature	Description	Input	Output	Output Format	Exception
createConcept(String name, String description)	Create new Concept	- Name: the name of the new concept. - Description : the display concept information	Newly created concept information displayed	Formatted string	Error Message displayed as formatted string
addImage(Integer conceptId, MultipartFile fileImage)	Add image to Concept	- ConceptId: Id of concept - FileImage: list image	Json new list image	Formatted string	JsonProcessingException
removeImage(Integer conceptId, String pictureId)	Remove image from concept	- ConceptId: Id of concept - PictureId: Id of picture	Json information image removed	Formatted string	JsonProcessingException

detectObject(String imageUrl)	Get information of object in the picture	- ImageUrl: url of image	Json string	Formated string	JsonProcessingException

**Table 47: Object API**

#### 4.5.1.2 Modula of Person API

Signature	Description	Input	Output	Output Format	Exception
AddFaceToPerson (MultipartFile imageFile, String personId)	Add face into Person on MCS to training	- ImageFile: list image face - PersonId: id of person	Json String of faceid	Formated string	JsonProcessingException
RemoveFaceFromPerson (String personId, String faceId)	Remove face from person on MCS	- PersonId: id of person - faceId: id of face	Json boolean of status	Formated string	JsonProcessingException
ListFace( String personId)	List face of person	- PersonId: id of person	Json string of List FaceModel contains	Formated string	sonProcessingException
CreatePersonGroup(String personGroupId, String groupName)	Create new person group	- PersonGroupId: id of person group - GroupName: name of	Json String of person group.	Formated string	JsonProcessingException

		person group			
GetPeopleInGroup (String personGroupId)	Get all person in specify group	- PersonGro upId: id of person group	Json list of person	Formatte d string	JsonProce ssingExcep tion
GetGroups()	Get all person group in system		Json list of person groups	Formatte d string	JsonProce ssingExcep tion

Table 48: Person Api

#### 4.5.1.3 Modula of View API

Signature	Description	Input	Output	Output Format	Exception
GetDetailPicture (String imageUrl)	Describe the picture	- ImageUrl: URL of image	Json String describ e the picture	Formatte d string	JsonProces singException

Table 49: View Api

#### 4.5.1.4 Modula of Data API

Signature	Description	Input	Output	Output Format	Exception
CreateLogPerson(String userId, String imageUrl)	Write log when can not detect person	- UserId: id of user - ImageUrl: URL of person can not detect	Json String of log	Formatte d string	JsonProces singException
GetAllLogFromUser(String userId)	Get all log from specify user	- UserId: id of user	Json String all log of user	Formatte d string	JsonProces singException

getAllLogObject()	Get all logs		Json list of logs	Formatted string	JsonProcessingException
CreateLogObject(String userId, String imageUrl)	Create Log Object when can not detect.	- UserId: id of user - ImageUrl: URL of Image	Json string of log	Formatted string	JsonProcessingException
GetDescriptionConcept(int conceptId)	Get description of concept	- ConceptId : Id of concept	Json String description of concept	Formatted string	JsonProcessingException
GetNameConcept(int conceptId)	Get name of concept	- ConceptId : Id of concept	Json String name of concept	Formatted string	JsonProcessingException

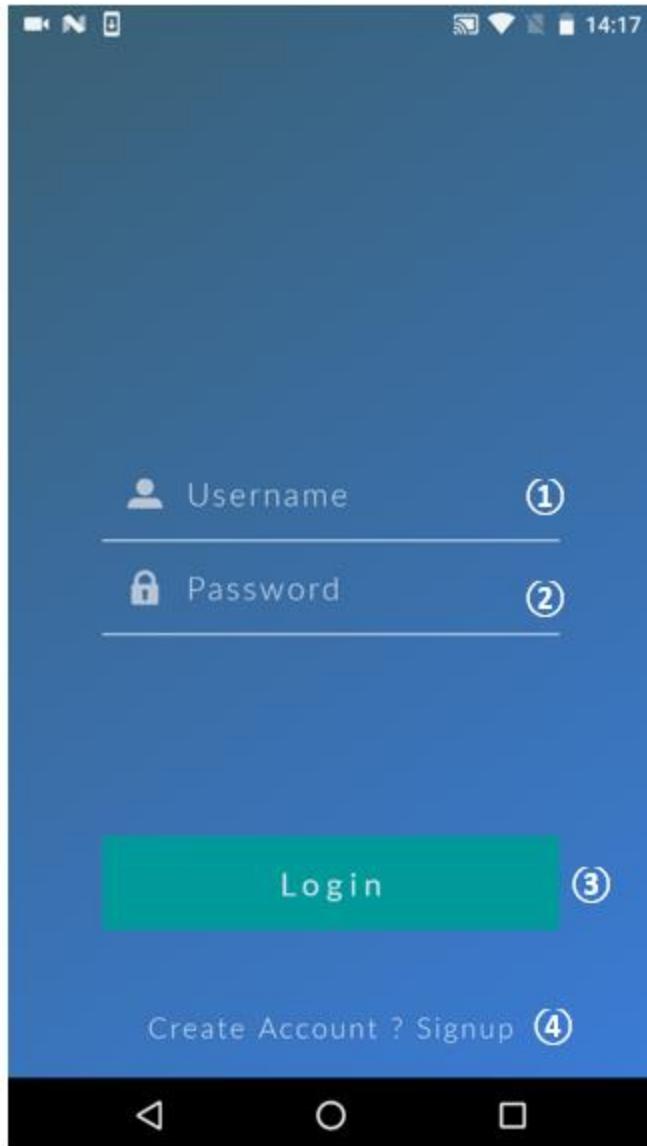
*Table 50: Data Api*

#### 4.5.2 User Interface Design

##### 4.5.2.1 Mobile Application Design

###### 4.5.2.1.1 Unauthorized User

###### 4.5.2.1.1.1 <Unauthorized User> Login



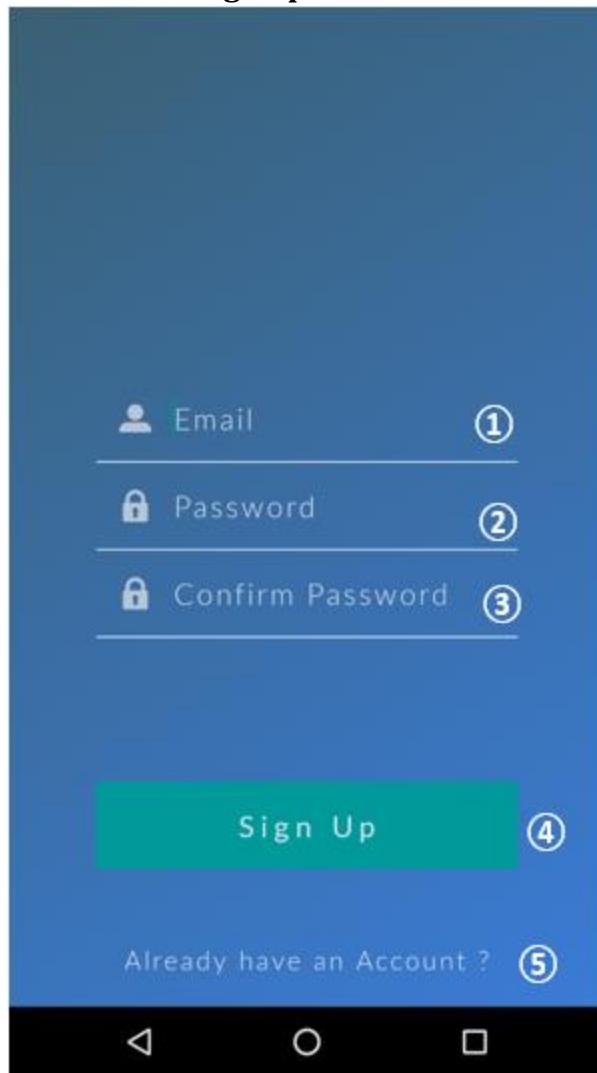
*Figure 37: Mobile Application UI - <Unauthorized User> Login*

**Fields:**

No	Field name	Description	Read only	Mandatory	Control type	Data type	Length
1	Username	Fill in username	No	Yes	Textbox	String	N/A
2	Password	Fill in password	No	Yes	Password	String	N/A

**Buttons/Hyperlinks:**

No	Function	Description	Validation	Outcome
3	Login	Log-in to the system	Email and Password cannot be empty	Transfer to specific page with specific account
4	Create account? Sign up	Sign up a new account	N/A	Redirect user to Sign up page

**4.5.2.1.1.2 <Unauthorized User> Sign up****Figure 38: Mobile Application UI - <Unauthorized User> Sign Up**

**Fields:**

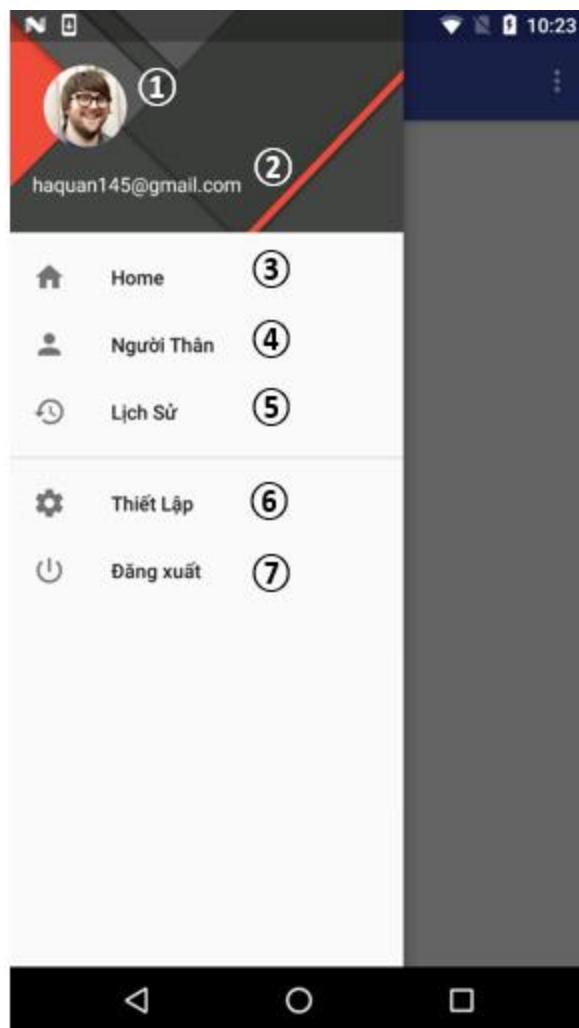
No	Field name	Description	Read only	Mandatory	Control type	Data type	Length
1	Username	Fill in username	No	Yes	Textbox	String	N/A
2	Password	Fill in password	No	Yes	Password	String	N/A
3	Confirm password	Fill in password again	No	Yes	Password	String	N/A

**Buttons/Hyperlinks:**

No	Function	Description	Validation	Outcome
4	Sign up	Create a new account in database	Email, Password and Confirm Password cannot be empty	Transfer to homepage for user
5	Already have an account?	User already has an account	N/A	Redirect user to Login page

#### 4.5.2.1.2 Authorized User

##### 4.5.2.1.2.1 <Authorized User> Menu



*Figure 39: Mobile Application UI - <Authorized User> Menu*

##### Fields:

No	Field name	Description	Read only	Mandatory	Control type	Data type	Length
1	Avatar	User's avatar	No	No		Image	N/A
2	Username	Username of current user	No	Yes	Label	String	N/A

##### Buttons/Hyperlinks:

No	Function	Description	Validation	Outcome

3	Home	Homepage of the application;	N/A	Transfer to Homepage
4	Người thân	Person list of current user	N/A	Transfer to Person list page
5	Lịch sử	List of unidentified person	N/A	Transfer to Log page
6	Thiết lập	The configuration page (Voice commands, ...)	N/A	Transfer to Setup page
8	Đăng xuất	Log out user from the system	N/A	Transfer to Login page

#### 4.5.2.1.2.2 <Authorized User> View Person List

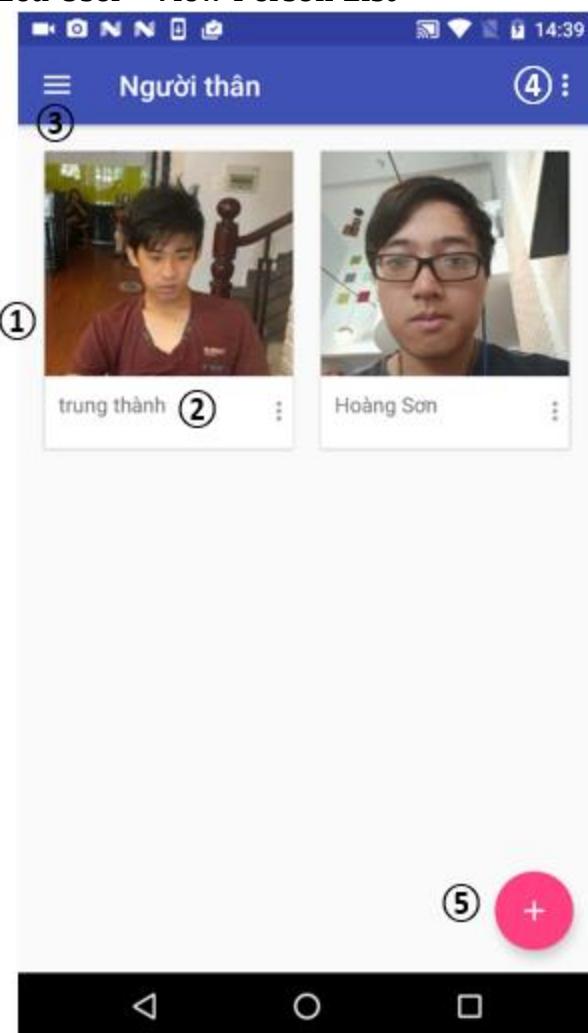


Figure 40: Mobile Application UI - <Authorized User> View Person List

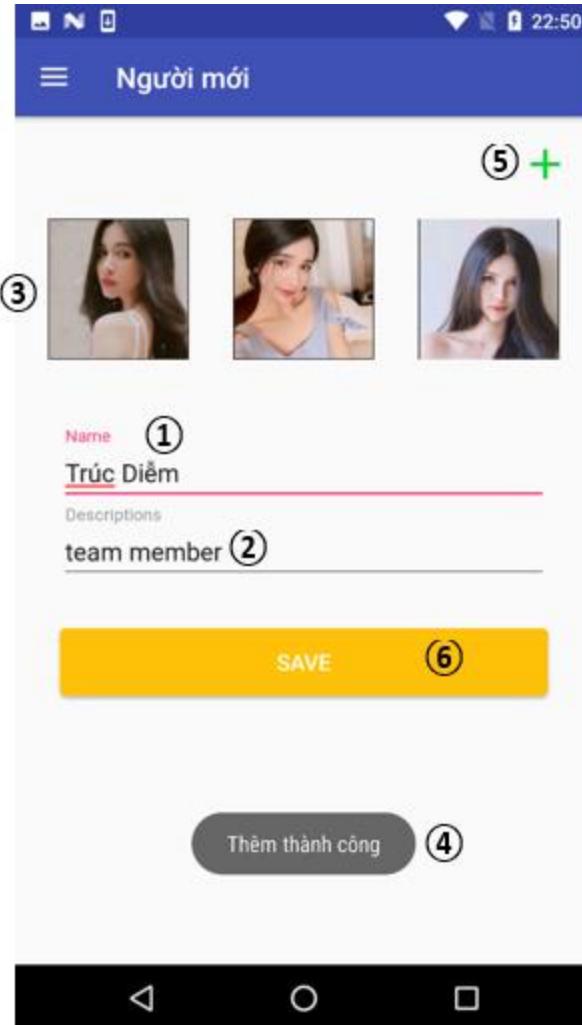
**Fields:**

No	Field name	Description	Read only	Mandatory	Control type	Data type	Length
1	Person's Image	The avatar of the person	No	Yes		Image	N/A
2	Name	Name of the person	No	Yes	Label	String	N/A

**Buttons/Hyperlinks:**

No	Function	Description	Validation	Outcome
3	Menu	Menu trigger button	N/A	Show the menu bar
4			N/A	
5	Add new person	Add a new person to the list	N/A	Transfer to new person page

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



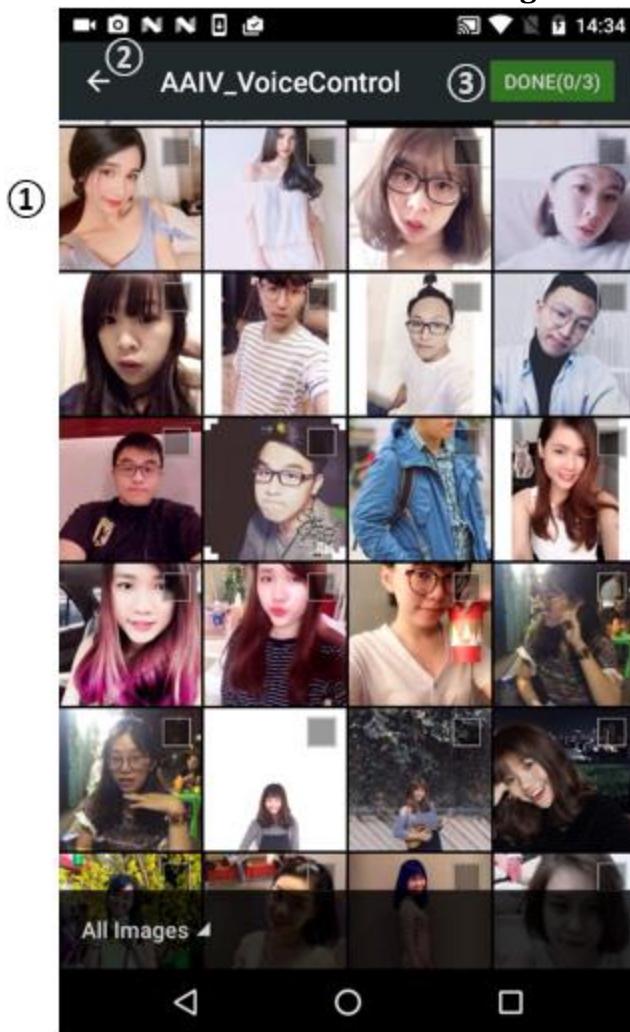
*Figure 41: Mobile Application UI - <Authorized User> Add New Person*

#### Fields:

No	Field name	Description	Read only	Mandatory	Control type	Data type	Length
1	Name	Fill in the person's name	No	Yes	Textbox	String	N/A
2	Description	Fill in the person's description	No	No	Textbox	String	N/A
3	Image preview	Show the preview of selected face images	No	Yes		Image	N/A
4	Notification	Show the status after clicking "Save" button	No	Yes	Label	String	N/A

**Buttons/Hyperlinks:**

No	Function	Description	Validation	Outcome
5	Add photos	Add face photos of person	N/A	Show the image browser for choosing
6	Save	Save person's information	Person's name cannot be blank	Create a new person with inputted information in the system

**4.5.2.1.2.4 <Authorized User> Select Person's face image**

*Figure 42: Mobile Application UI - <Authorized User> Select Person Face Image*

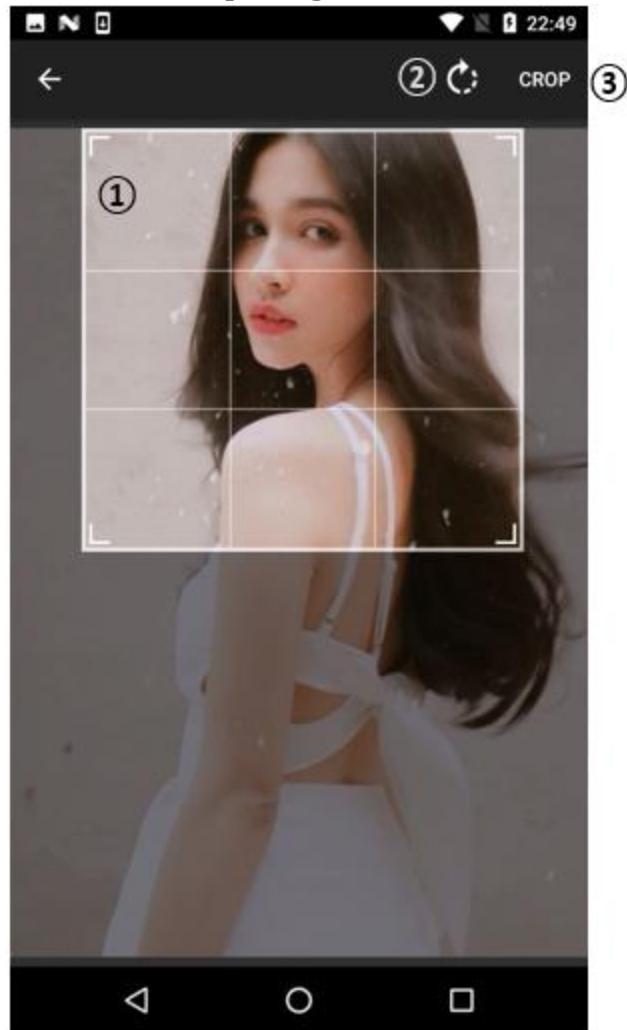
**Fields:**

No	Field name	Description	Read only	Mandatory	Control type	Data type	Length
1	Image	Image for choosing	No	Yes		Image	N/A

**Buttons/Hyperlinks:**

No	Function	Description	Validation	Outcome
2	Back	Return to the previous step	N/A	Return to the New Person page
3	Done	Confirm image selection	N/A	Return to the New Person page with selected images

#### 4.5.2.1.2.5 <Authorized User> Crop Image



*Figure 43: Mobile Application UI - <Authorized User> Crop Image*

#### Fields:

No	Field name	Description	Read only	Mandatory	Control type	Data type	Length
1	Crop zone	Selected zone for cropping	No	Yes		Image	N/A

#### Buttons/Hyperlinks:

No	Function	Description	Validation	Outcome
2	Rotate	Rotate the image	N/A	

3	Crop	Confirm the selected zone for cropping	N/A	Return to the New Person page with cropped image
---	------	--	-----	--

#### 4.5.2.1.2.6 <Authorized User> Update Person's Information

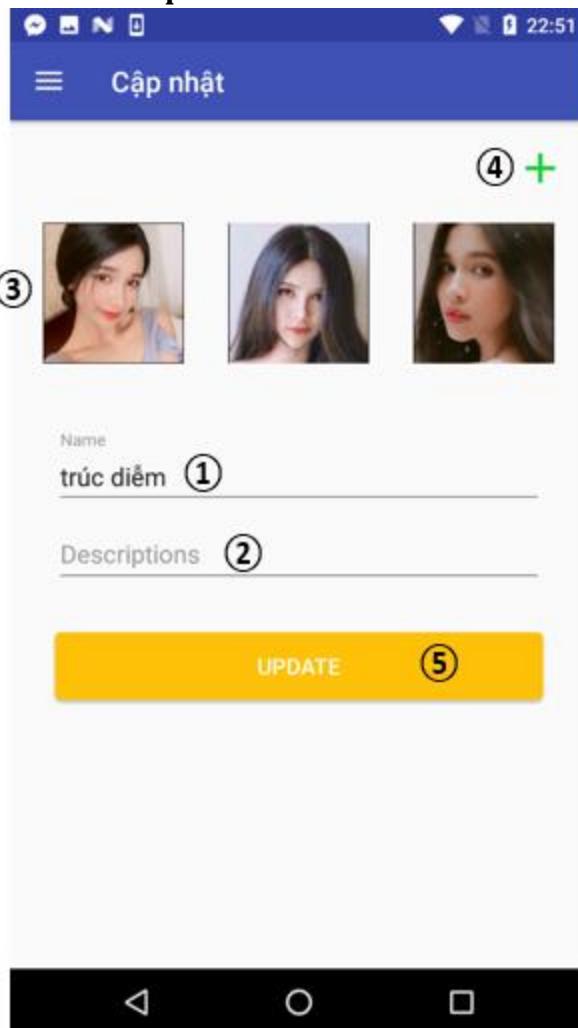


Figure 44: Mobile Application UI - <Authorized User> Update Person Information

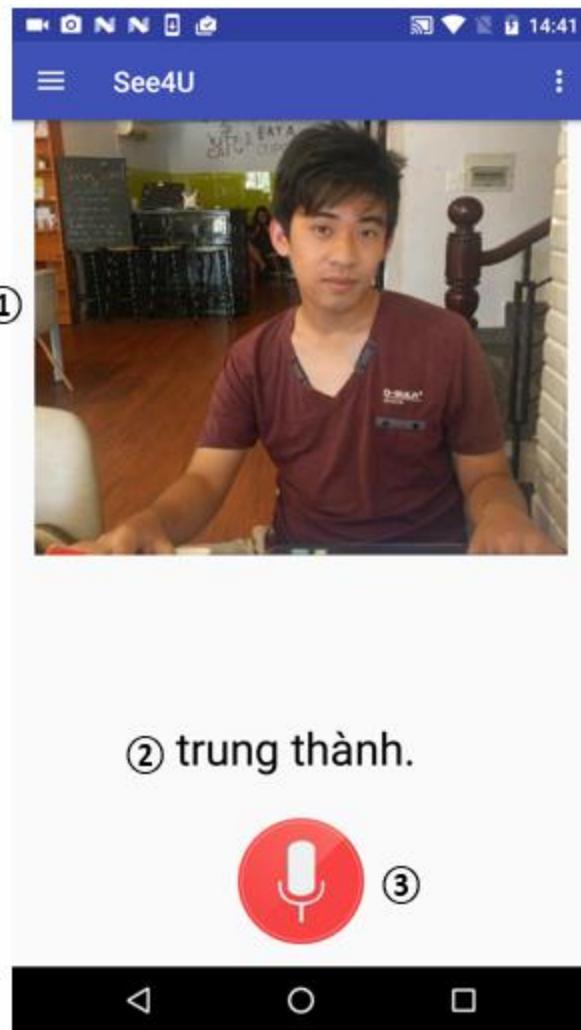
#### Fields:

No	Field name	Description	Read only	Mandatory	Control type	Data type	Length
1	Name	Fill in the person's name	No	Yes	Textbox	String	N/A
2	Description	Fill in the person's description	No	No	Textbox	String	N/A

3	Image preview	Show the preview of selected face images	No	Yes		Image	N/A
---	---------------	--	----	-----	--	-------	-----

**Buttons/Hyperlinks:**

No	Function	Description	Validation	Outcome
4	Add photos	Add face photos of person	N/A	Show the image browser for choosing
5	Update	Save person's information	Person's name cannot be blank	Update person with inputted information in the system

**4.5.2.1.2.7 <Authorized User> Detect Person****Figure 45: Mobile Application UI - <Authorized User> Detect Person**

**Fields:**

No	Field name	Description	Read only	Mandatory	Control type	Data type	Length
1	Image	Show the captured image	No	Yes		Image	N/A
2	Name	Name of the person in the image	No	Yes	Label	String	N/A

**Buttons/Hyperlinks:**

No	Function	Description	Validation	Outcome
3	Voice control	Get the voice command	N/A	Transfer to the specific page follows the inputted command

#### 4.5.2.1.2.8 <Authorized User> Detect Objects



*Figure 46: Mobile Application UI - <Authorized User> Detect Object*

**Fields:**

No	Field name	Description	Read only	Mandatory	Control type	Data type	Length
1	Image	Show the captured image	No	Yes		Image	N/A
2	Name	Show the name of the object in image	No	Yes	Label	String	N/A

**Buttons/Hyperlinks:**

No	Function	Description	Validation	Outcome

3	Voice control	Get the voice command	N/A	Transfer to the specific page follows the inputted command
---	---------------	-----------------------	-----	--

#### 4.5.2.1.2.9 <Authorized User> Detect View

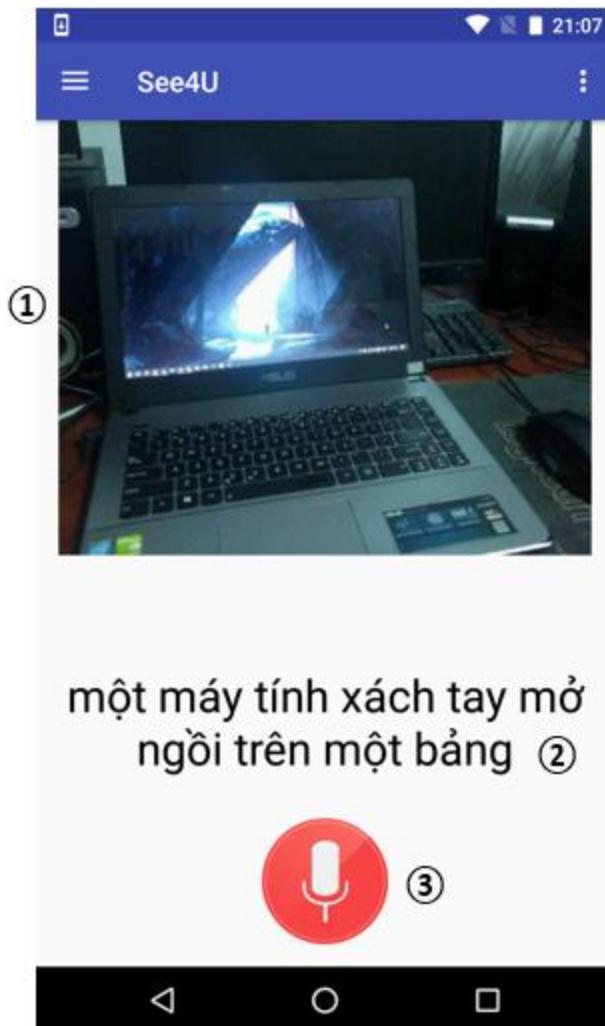


Figure 47: Mobile Application UI - <Authorized User> Detect View

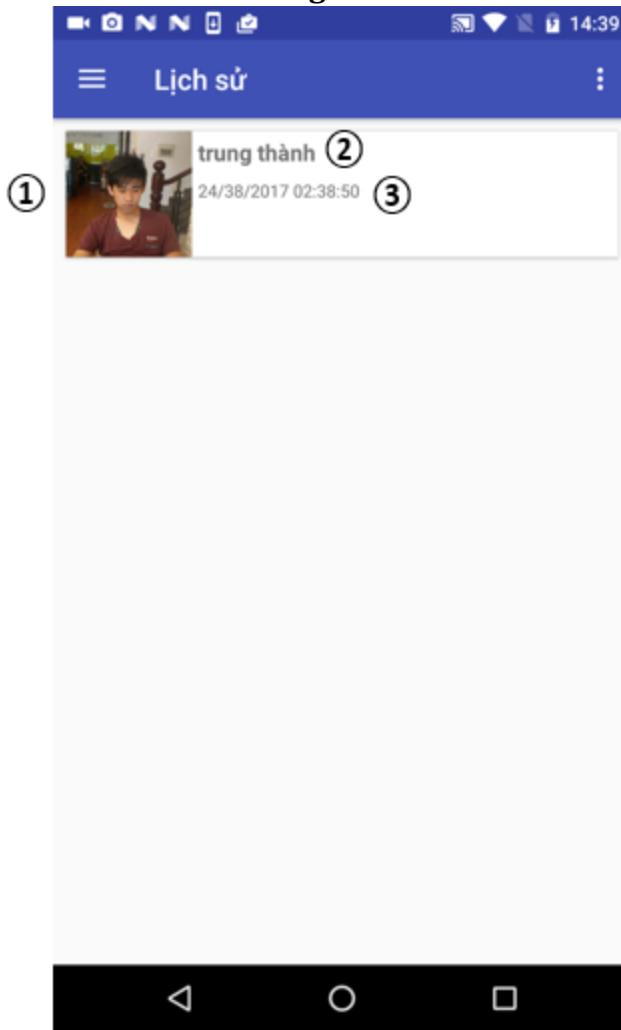
#### Fields:

No	Field name	Description	Read only	Mandatory	Control type	Data type	Length
1	Image	Show the captured image	No	Yes		Image	N/A
2	Description	The information	No	Yes	Label	String	N/A

		about the view in the image					
--	--	-----------------------------	--	--	--	--	--

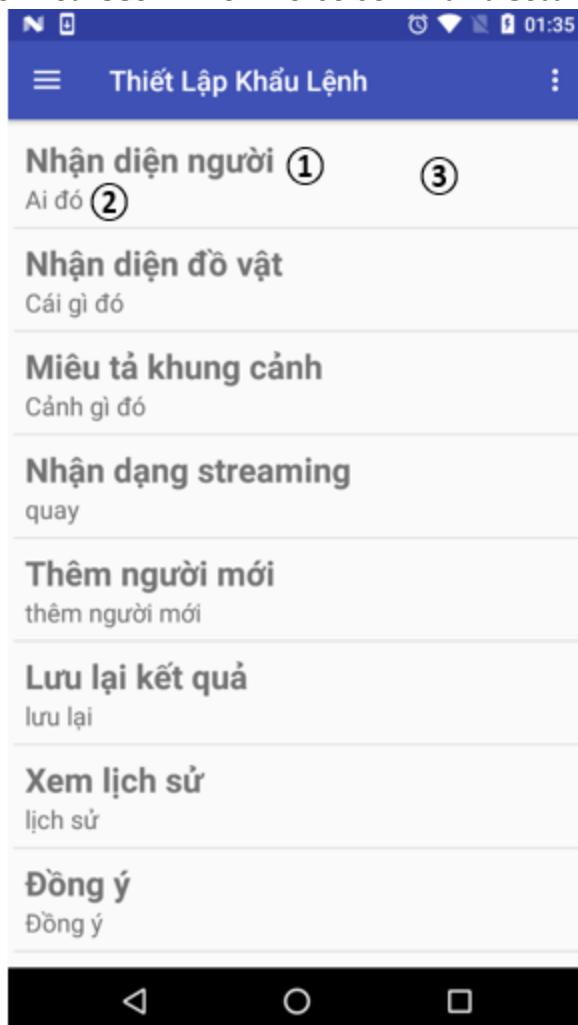
**Buttons/Hyperlinks:**

No	Function	Description	Validation	Outcome
3	Voice control	Get the voice command	N/A	Transfer to the specific page follows the inputted command

**4.5.2.1.2.10 <Authorized User> View log***Figure 48: Mobile Application UI - <Authorized User> View Logs*

**Fields:**

No	Field name	Description	Read only	Mandatory	Control type	Data type	Length
1	Image	Show the log image	No	Yes		Image	N/A
2	Name	Name of the person in the image	No	Yes	Label	String	N/A
3	Date	Created date of the log	No	Yes	Label	Date time	

**4.5.2.1.2.11 <Authorized User> View voice command setting****Figure 49: Mobile Application UI - <Authorized User> Voice Commands Setting**

**Fields:**

No	Field name	Description	Read only	Mandatory	Control type	Data type	Length
1	Command name	The name of the command	No	Yes	Label	String	N/A
2	Command	The command string to activate the specific function	No	Yes	Label	String	N/A

**Buttons/Hyperlinks:**

No	Function	Description	Validation	Outcome
3	Modify command	Change the command string	N/A	Display the textbox to get the command input

**4.5.2.2 Web Application Design****4.5.2.2.1 Unauthorized User****4.5.2.2.1.1 <Unauthorized User> Register**

Đăng kí tài khoản - AAV

Email ①

Mật khẩu ②

Xác nhận ③

Đăng kí ④

Đã có tài khoản ? Đăng nhập ⑤

*Figure 50: 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 51: 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 52: Web Application Interface - Register - Buttons/Hyperlinks:***4.5.2.2.1.2 <Unauthorized User> Login**

The screenshot shows a login form titled "AAIV 1.0". It includes fields for "Email" (1) and "Mật khẩu" (2), a "Remember Me" checkbox (3), a large blue button labeled "Đăng nhập" (4), and a link "Đăng kí tài khoản mới" (5).

*Figure 51: 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 53: Web Application Interface – Login - Fields*

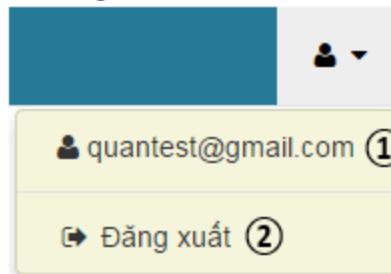
**Buttons/Hyperlinks:**

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 54: Web Application Interface – Login – Buttons/Hyperlinks*

#### 4.5.2.2.2 Authorized User

##### 4.5.2.2.2.1 <Authorized User> Logout

*Figure 52: 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 55: 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 56: Web Application Interface - Logout - Buttons/Hyperlinks****4.5.2.2.2 <Authorized User> View Persons**

The screenshot shows a list of persons with the following data:

No	Ánh đại diện	Tên	Thông tin	Action
3		Trung Thành	Team Leader	(5) Edit (6) Delete
4		Hoàng Sơn	Team Member #2	(5) Edit (6) Delete
1		Hà Quân	Team Member #3	(5) Edit (6) Delete
2		Cao Duy	Team Member #4	(5) Edit (6) Delete

**Figure 53: Web Application Interface - <Authorized> View Persons****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 the previous page
8	btnNext	Go to next person page	N/A	Show next page

*Table 57: 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	Ánh 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 58: Web Application Interface - View Persons - Fields***4.5.2.2.3 <Authorized User> Add New Person**

The screenshot shows a user interface for adding a new person. At the top, there's a header bar with the text 'AAV 1.0'. Below it, a sidebar on the left lists 'Quản lý người thân' and 'Người chưa nhận diện được'. The main area has a title 'Thêm người mới'. It contains two input fields: 'Tên' (Name) and 'Thông tin' (Description). To the right of these fields is a circular placeholder for an image, labeled 'Hình ảnh' (Image). Below the fields is a blue 'Thêm mới' (Add) button, which is labeled with a circled '4'. The entire interface is in Vietnamese.

*Figure 54: 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 59: 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 60: Web Application Interface – Add New Person - Buttons/Hyperlinks***4.5.2.2.2.4 <Authorized User> Update Person**
*Figure 55: Web Application Interface - <Authorized> Update Person*

**Fields:**

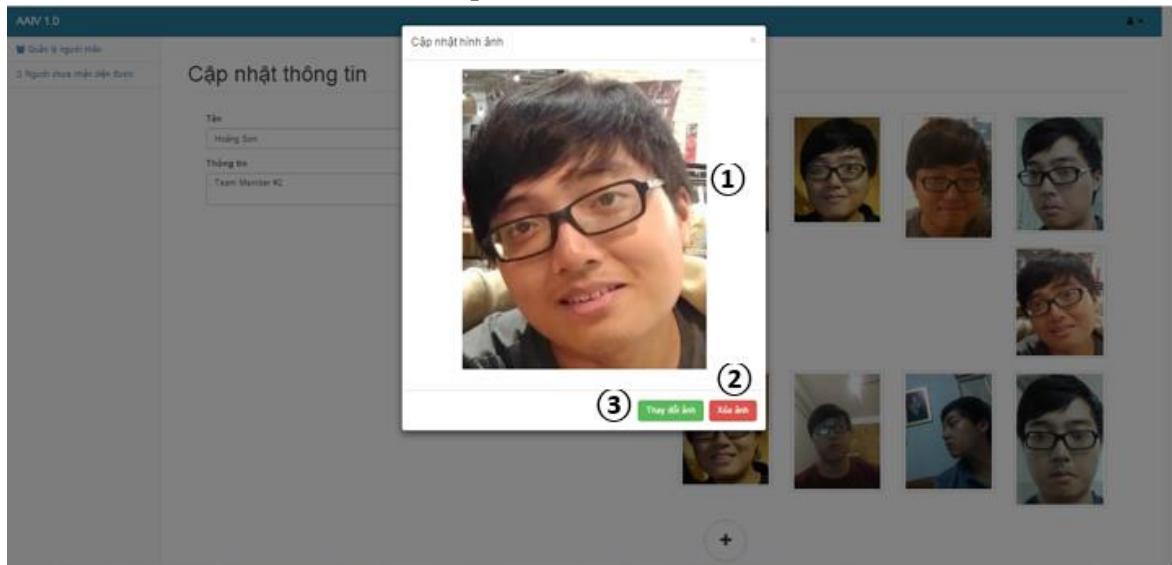
No	Field Name	Description	Read Only	Mandatory	Control Type	Data Type	Length
1	Tên	Fill in person's name	No	Yes	Textbox	String	N/A
2	Thông tin	Fill in person's information	No	No	Textbox	String	N/A

*Table 61: 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	Inputted 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 62: Web Application Interface – Update Person - Buttons/Hyperlinks*

#### 4.5.2.2.2.5 <Authorized User> Update Person's Face



*Figure 56: 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 63: Web Application Interface – Update Person'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 64: Web Application Interface – Update Person's Face - Buttons/Hyperlinks*

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

*Figure 57: 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

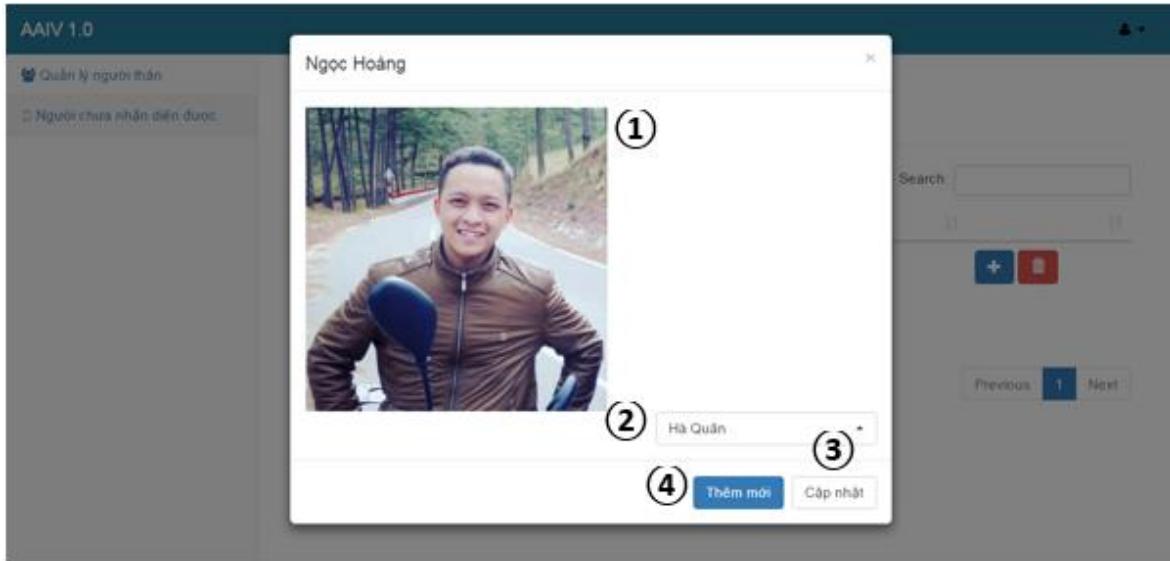
*Table 65: Web Application Interface - Show Logs From Person - Fields*

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

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



*Figure 58: 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 67: 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 68: Web Application Interface – Create/Update From Log Person - Buttons/Hyperlinks*

#### 4.5.2.2.3 Admin

##### 4.5.2.2.3.1 <Admin> Manage User

The screenshot shows a web application interface titled 'Danh sách tài khoản'. The left sidebar has links for 'Quản lý tài khoản' (10), 'Quản lý Concept' (11), and 'Nhận dạng đồ vật' (12). The main area displays a table of users with the following columns: ID (2), Tài khoản (3), Email (4), Số điện thoại (5), and Status (6). The table contains three entries:

ID	Tài khoản	Email	Số điện thoại	Status
661d39a3-ad8f-45e3-8c35-05728d16da6	quanha@gmail.com	quanha@gmail.com	On	
cab8f571-3b22-431d-a872-2de96165a36e	test@gmail.com	test@gmail.com	Off	
f6575bcc-e47b-4b6c-be91-b76a2af3d587	son@gmail.com	son@gmail.com	On	

At the bottom, there is a search bar (8), a 'Cập nhật' button, and navigation buttons for 'Previous' (13), 'Next' (14), and page number '1'.

Figure 59: 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 Khoán	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 deactivate	Yes	Yes	Checkbox	bit	N/A

Table 69: 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 the previous page	N/A	Go to the previous page contain list data
14	Next	Go to next page	N/A	Go to next page contain list data

**Table 70: Web Application Interface - Manage User - Buttons/Hyperlinks****4.5.2.2.3.2 <Admin> View Concepts**

Danh sách đồ vật ①

⑧ Tạo mới đồ vật ⑨ Kiểm tra đồ vật

Show 10 entries ⑦

Search: ⑥

② No.	③ Image	④ Tên	⑤ Chú thích	
1		đồng hồ casio	bộ máy chiếu	⑩ <span style="color: blue;">Edit</span> ⑪ <span style="color: red;">Delete</span>
2		giày	giày nike trắng	<span style="color: blue;">Edit</span> <span style="color: red;">Delete</span>

**Figure 60: 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 71: Web Application Interface – View Concepts - Field.***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 the system.	N/A	Redirect admin to create new object page.
9	testConcept	Test object has existed 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 72: Web Application Interface – View Concepts – Buttons/Hyperlinks*

#### 4.5.2.2.3.3 <Admin> Create Concept

#### Tạo mới đồ vật ①

The screenshot shows a web application interface for creating a new object. At the top left is the title 'Tạo mới đồ vật ①'. Below it are two input fields: 'Tên Đồ Vật' (Object Name) with placeholder 'Nhập tên đồ vật!' (Enter object name!) and 'Mô Tả Đồ Vật' (Object Description) with placeholder 'Nhập mô tả đồ vật!'. To the right is an 'Hình Ảnh' (Image) section featuring a yellow Pikachu icon. Below the image is a circular button with a plus sign. At the bottom are three buttons: a grey 'Quay Lại' (Back) button, a green 'Tạo Mới' (Create New) button with a checkmark, and a grey 'Làm Mới' (Reset) button.

*Figure 61: 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 73: 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 74: Web Application Interface - Create Concepts - Buttons/Hyperlinks*

#### 4.5.2.2.3.4 <Admin> Update Concept

Cập nhật đồ vật ①

Tên Đồ Vật  
đỗ bấm casio ②

Mô Tả Đồ Vật  
bấm máy chiếu ③

Thêm hình ④

⑤

⑥

⑦

⑧

⑨

← Quay Lại Cập Nhật

Figure 62: 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 75: 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 updated into system.
9	viewAllConcept	Go back another page	N/A	Redirect view all object page.

*Table 76: Web Application Interface – Update Concepts – Buttons/Hyperlinks*

#### 4.5.2.2.3.5 <Admin> View Logs

*Danh Sách Vật Thể Chưa Nhận Điện Được ①*

The screenshot shows a table with columns: No. (2), Hình ảnh (3), Ngày tạo (4). Row 1: No. 1, Image of two phones, Date 2017-03-14 00:08:14, Buttons (7) +, (8) delete. Row 2: No. 2, Image of a backpack, Date 2017-03-14 00:08:33, Buttons (7) +, (8) delete. Row 3: No. 3, Image of a hand holding a phone, Date 2017-03-16 09:24:27, Buttons (7) +, (8) delete. Top navigation includes Show 10 entries (6), Search (5).

No. (2)	Hình ảnh (3)	Ngày tạo (4)	(7) + (8)
1		2017-03-14 00:08:14	
2		2017-03-14 00:08:33	
3		2017-03-16 09:24:27	

*Figure 63: 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 Điện 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 77: 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 78: Web Application Interface – View Logs – Buttons/Hyperlinks****4.5.2.2.3.6 <Admin> Update Concept From Log Object**

Thêm hình ảnh ①

**Figure 64: 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 79: 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 80: Web Application Interface – Update Concept From Log Object – Buttons/Hyperlinks*

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

#### Tạo mới đồ vật ①

The screenshot shows a web application interface for creating a new concept from a log object. The interface is in Vietnamese. At the top right is a 'Cancel' button (⑥) and a green 'Save' button (⑤) labeled '✓ Tạo Mới'. To the left of the buttons are two input fields: one for 'Name' (Nhập tên đồ vật! ②) and one for 'Description' (Nhập mô tả đồ vật! ③). Above these fields is a placeholder 'Hình Ảnh' (Image). To the right of the image placeholder is a small image of a Sony Xperia smartphone. The entire interface is contained within a light gray box.

*Figure 65: 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 81: 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 82: Web Application Interface – Create Concept From Log Object – Buttons/Hyperlinks*

## 4.6 Database Design

### 4.6.1 Entity relationship diagram (ERD)

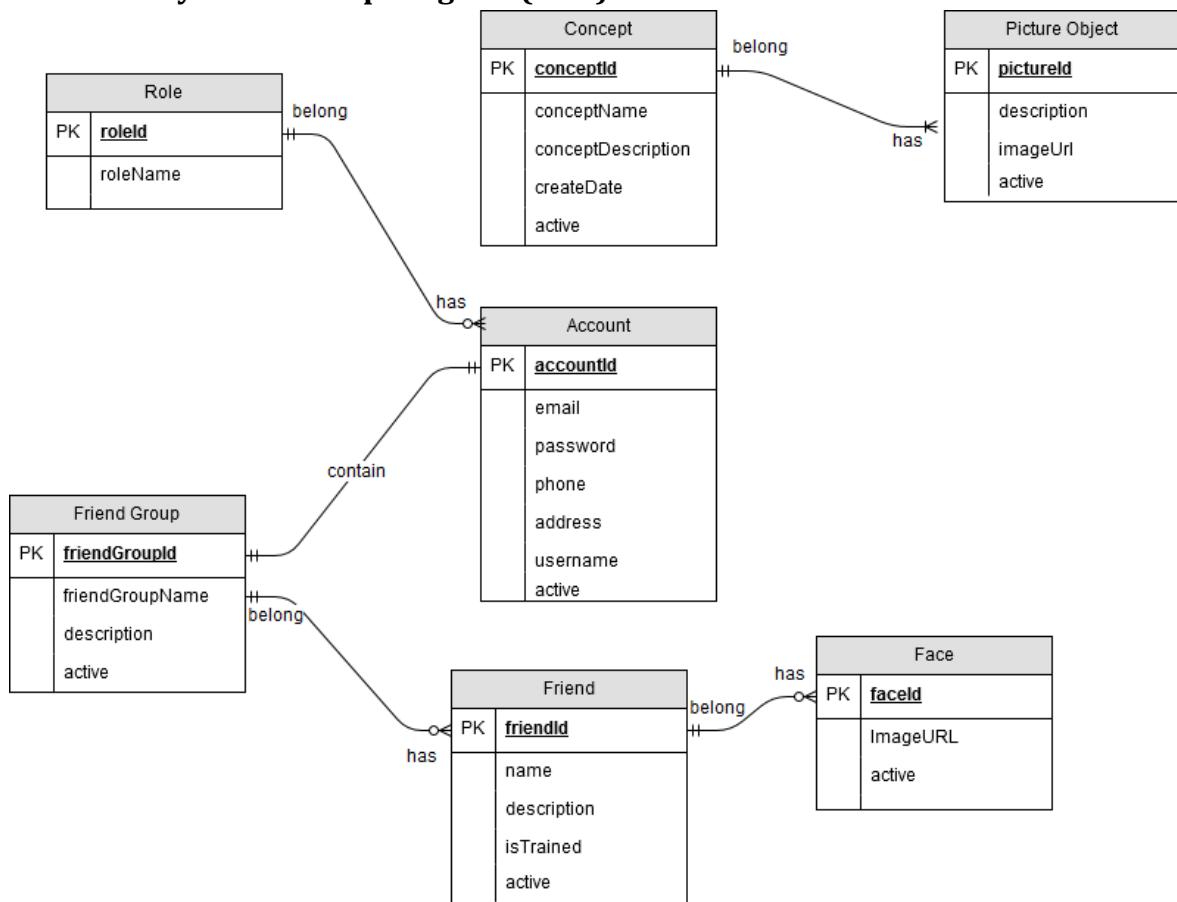


Figure 66: <Admin > Entity relationship diagram (ERD)

### 4.6.2 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

## 4.7 Algorithms

### 4.7.1 Person Group For Each User

#### 4.7.1.1 Definition

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

#### 4.7.1.2 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 persons into the smaller group, 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.

#### 4.7.1.3 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 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.

#### 4.7.1.4 References

- <https://www.microsoft.com/cognitive-services/en-us/face-api/documentation/overview>
- <https://westus.dev.cognitive.microsoft.com/docs/services/563879b61984550e40cbbe8d/operations/563879b61984550f30395239>

### 4.7.2 Voice Recognition Algorithms

#### 4.7.2.1 Definition

- This algorithm is used to recognize user voice command when interacting with the application.

#### 4.7.2.2 Define Problem

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

#### 4.7.2.3 Solution

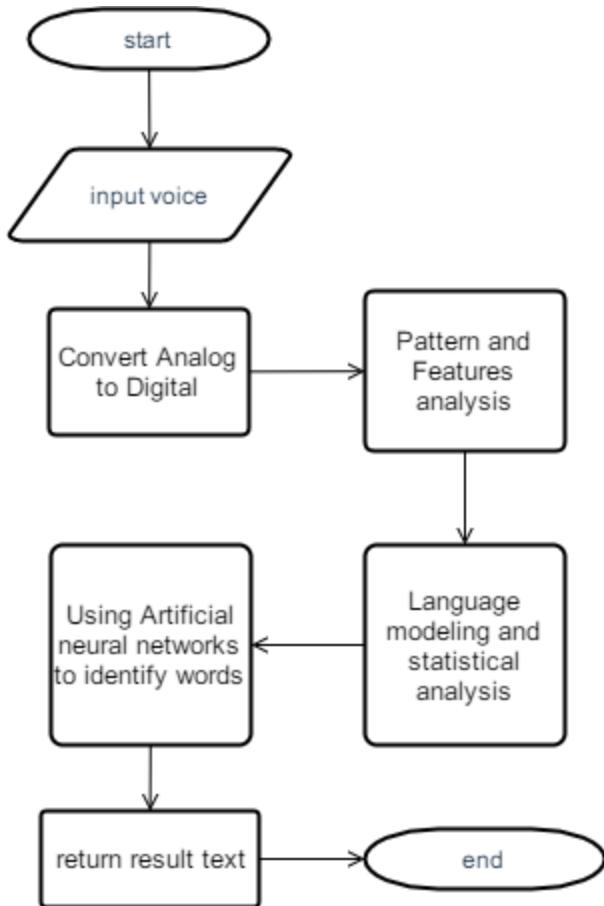
- The Voice Recognition process can be broken down into followed steps:
  - o Step1: User input voice data
  - o Step2: Analog-to-Digital converter transfer analog wave into digital data that computer can understand. The process contains followed smaller steps:
    - Digitizes the input sound by taking a 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.
  - o Step3: Using language models and statistical analysis to reduce mistakes, analyze language pattern and speed up the process.
  - o 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 application.

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

#### 4.7.2.5 Flow Chart



#### 4.7.2.6 References

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

### 4.7.3 Image Recognition Algorithms

#### 4.7.3.1 Definition

- This algorithm allows the system to analyze images to identify object and recognize human faces and their features.

#### 4.7.3.2 Define Problems

- The application provides two main features:
  - o Facial Recognition
  - o Object Recognition
- In order to fulfill its features, the system must use an algorithm to analyze the provided image from user, to filter out human facial details or object details.

#### 4.7.3.3 Solution

- Image Recognition process can be broken down into followed steps:
  - o Step 1: turn inputted image into black and white image.
  - o Step 2: divide inputted image into multiple overlapping parts
  - o Step 3: Using small neural networks to process each of the divided parts of the image and return output as digital numbers.
  - o Step 4: Downsampling process using Max-pooling algorithms to filter most important parts of the output in step 3.
  - o Step 5: Making final guess using Fully-Connected Neural Network and return prediction.
- After examining the basic concept image recognition, we decided to use **Microsoft Cognitive Service** and **Clarifai** to be our Image Recognizer Service, because of followed reasons:
  - o Both required inputted data as image to return predictions.
  - o Both required training process to be able to make predictions.
  - o Microsoft Cognitive Service provides a friendly and fully developed Face Recognition API, which can be used to store, train and identify human faces with high accuracy and good performance (refer to **4.7.4 Detect Person Algorithms**).
  - o Clarifai provides a training service for training and recognizes objects, with high accuracy depend on the amount of training data.

#### 4.7.3.4 Complexity

- The complexity of the algorithms might varies depend on the size of inputted data or amount of Convolution layers.
- Refers to **4.7.4 Detect Person Algorithms** and **4.7.8 Detect Object Algorithms** for more information.

#### 4.7.3.5 References

- <https://medium.com/@ageitgey/machine-learning-is-fun-part-3-deep-learning-and-convolutional-neural-networks-f40359318721>
- <https://www.cs.nyu.edu/~fergus/papers/zeilerECCV2014.pdf>

#### 4.7.4 Detect Person Algorithms

##### 4.7.4.1 Definition

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

##### 4.7.4.2 Define Problem

- The user needs to use the application to identify the 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 the contact.

##### 4.7.4.3 Solution

The Identifying process content these followed steps:

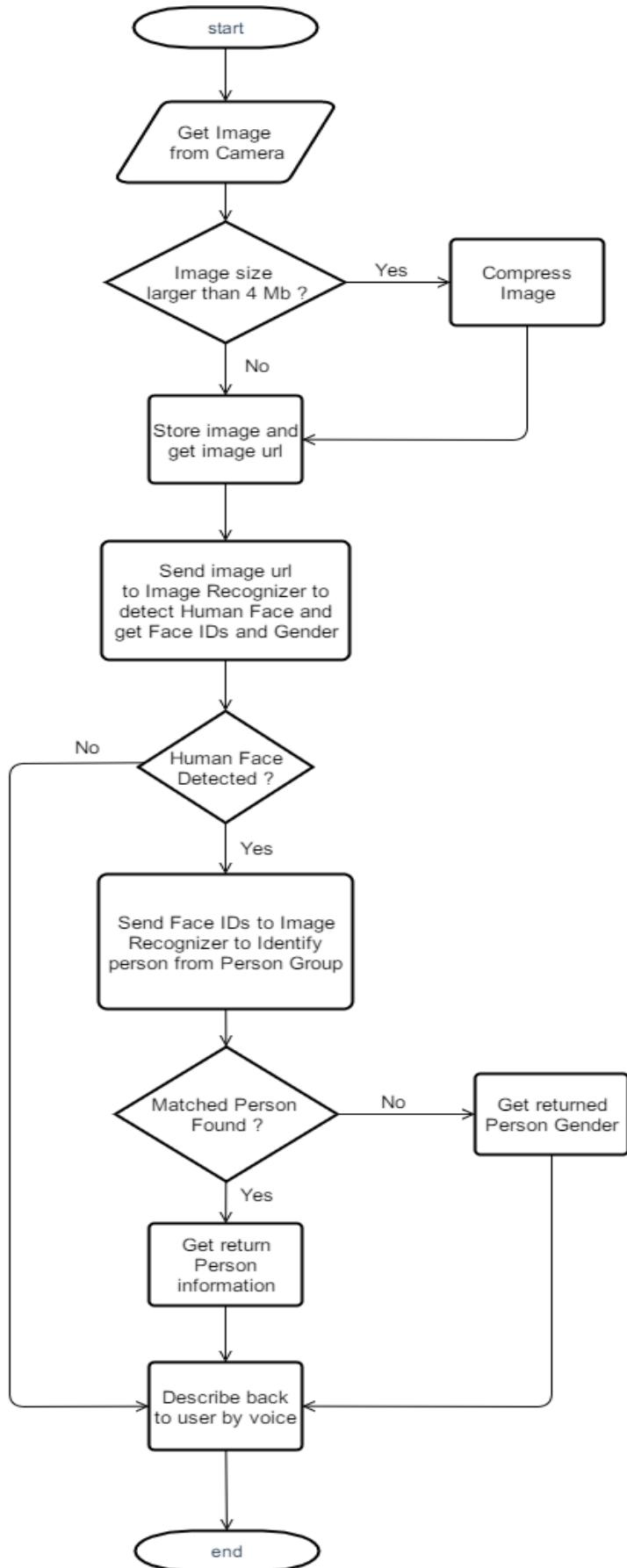
- Step 1: User open Detect Person mode by voice command.
- Step 2: User capture picture of the person.
- Step 3: System stores captured picture and send image URL to Image Recognizer to detect human face(s) as Face IDs and their Gender (Male or Female). If no human faces is detected in the image, the 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 <= 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.

##### 4.7.4.4 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

#### 4.7.4.5 Flow Chart



#### 4.7.5 Training Person Algorithms

##### 4.7.5.1 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.

##### 4.7.5.2 Define Problem

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

##### 4.7.5.3 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's face, which will not be expired.

##### 4.7.5.4 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 the 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's 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.

##### 4.7.5.5 Complexity

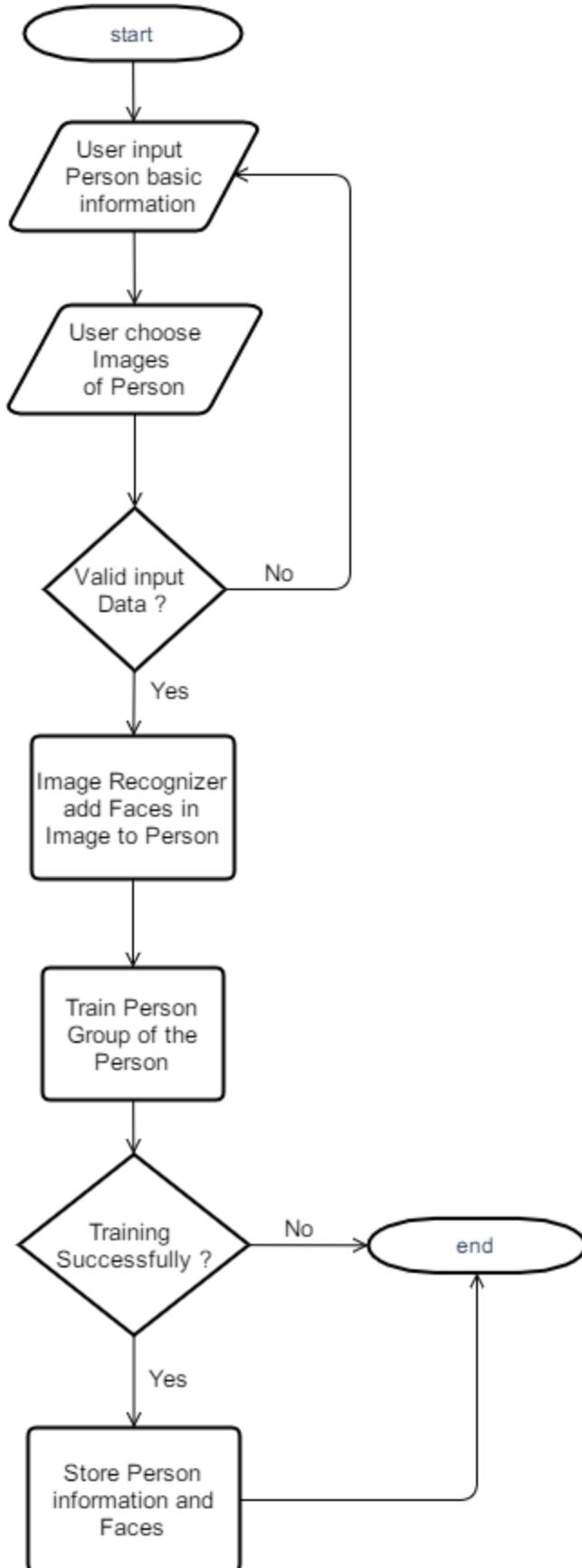
For the complexity of this algorithms is depend on the Image Recognition Service, which is not described in their documents. Therefore we spent times testing the performance of the Training Process:

<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

#### 4.7.5.6 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

**4.7.5.7 Flow Chart**

#### 4.7.6 Find Duplicated Person By Faces Algorithms

##### 4.7.6.1 Definition

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

##### 4.7.6.2 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.

##### 4.7.6.3 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 >=
                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:

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

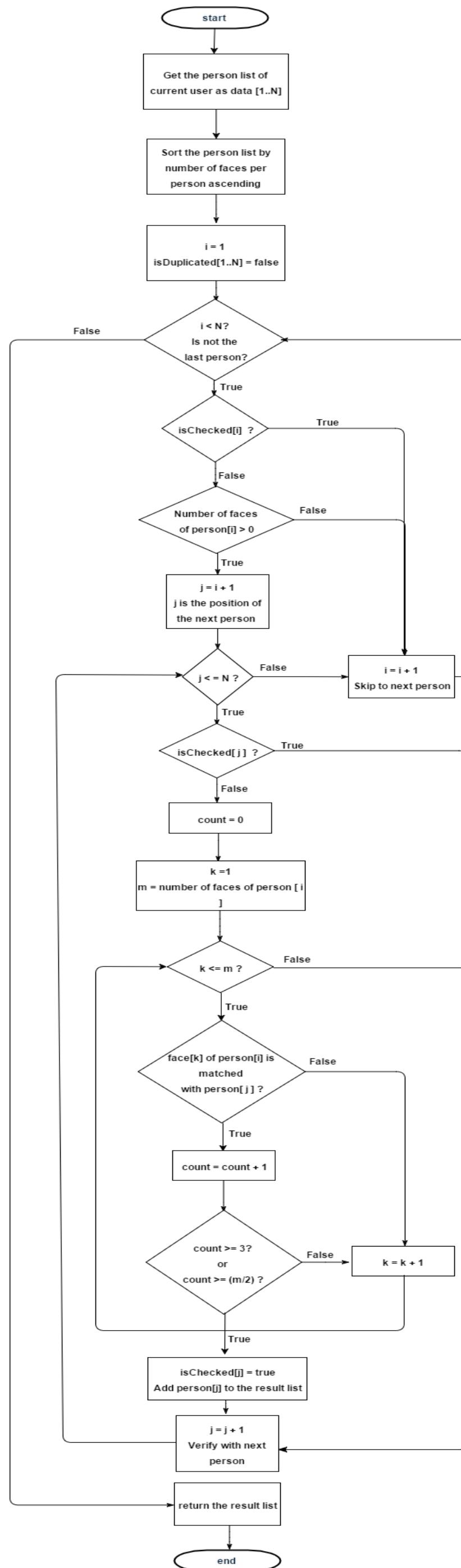
#### 4.7.6.4 Complexity

- 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  $M * O(n^2)$
- ⇒ The complexity of this algorithm is:  $M * O(n^2)$

#### 4.7.6.5 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>

#### 4.7.6.6 Flowchart



#### 4.7.7 Find Duplicated Person By Name Algorithms

##### 4.7.7.1 Definition

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

##### 4.7.7.2 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.

##### 4.7.7.3 Solution

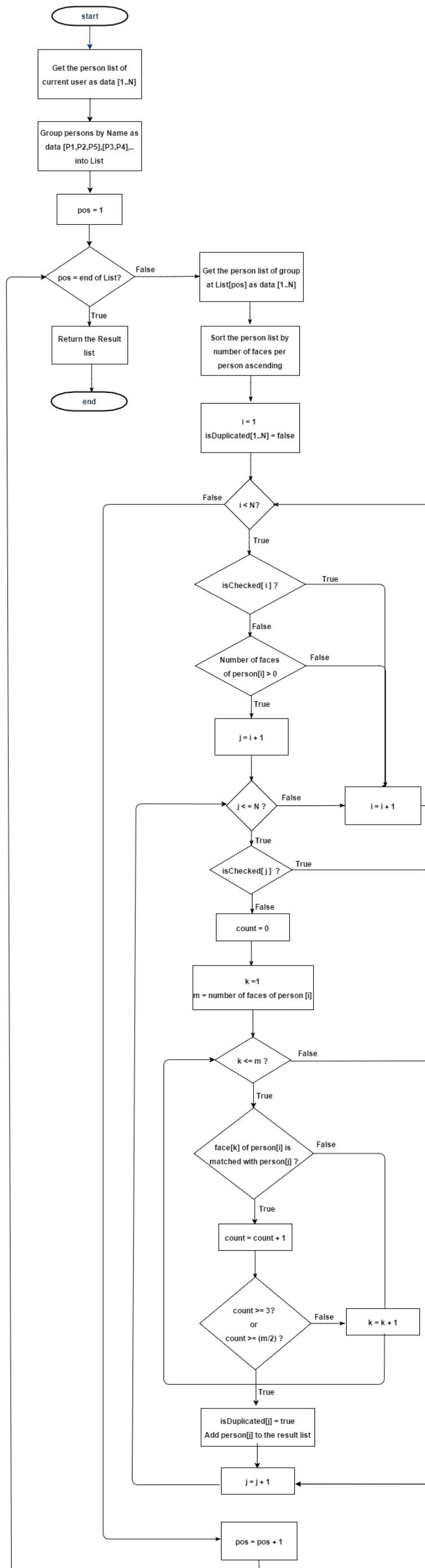
To solve this problem, we follow these steps:

- 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.

##### 4.7.7.4 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  $M * O(n^2)$  (reference 4.7.4.4)
  - ⇒ The complexity of this algorithm is:  $M * O(n^2)$

##### 4.7.7.5 Flowchart



#### 4.7.8 Detect Object Algorithms

##### 4.7.8.1 Definition

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

##### 4.7.8.2 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.

##### 4.7.8.3 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 <= 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.

##### 4.7.8.4 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:

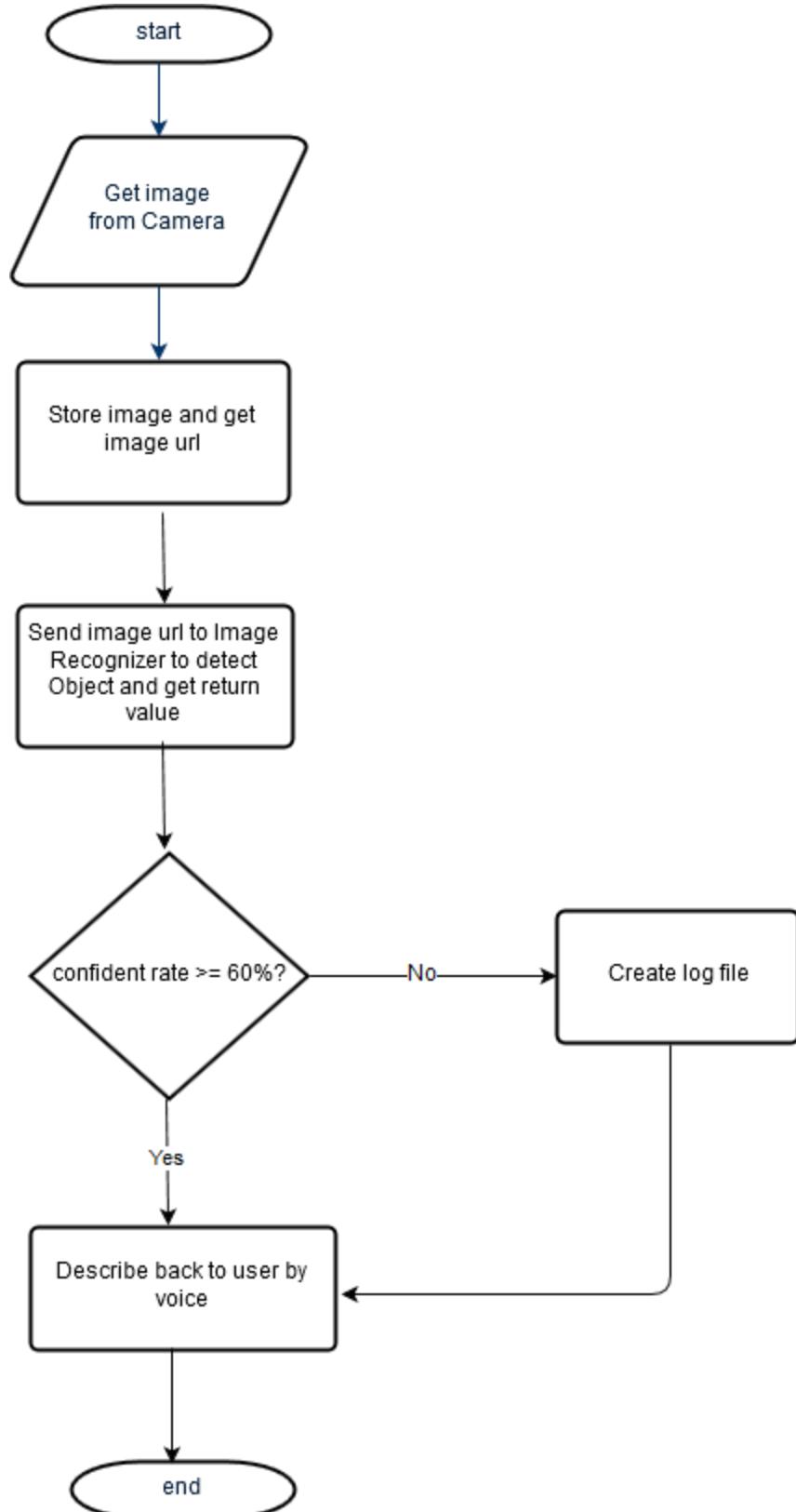
<b>Network Type</b>	<b>Speed (Voice Recognition + Image Recognition)</b>
Wi-Fi Network (Ping: 3ms, Download: 30.55Mbps, Upload: 22.17 Mbps)	5,731 ms
<b>3G Network</b> (Ping 24ms, Download: 15.31 Mbps, Upload: 1.36 Mbps)	6,182 ms

##### 4.7.8.5 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

#### 4.7.8.6 Flowchart



#### 4.7.9 Training Object Algorithms

##### 4.7.9.1 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

##### 4.7.9.2 Define Problem

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

##### 4.7.9.3 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.

##### 4.7.9.4 Solution

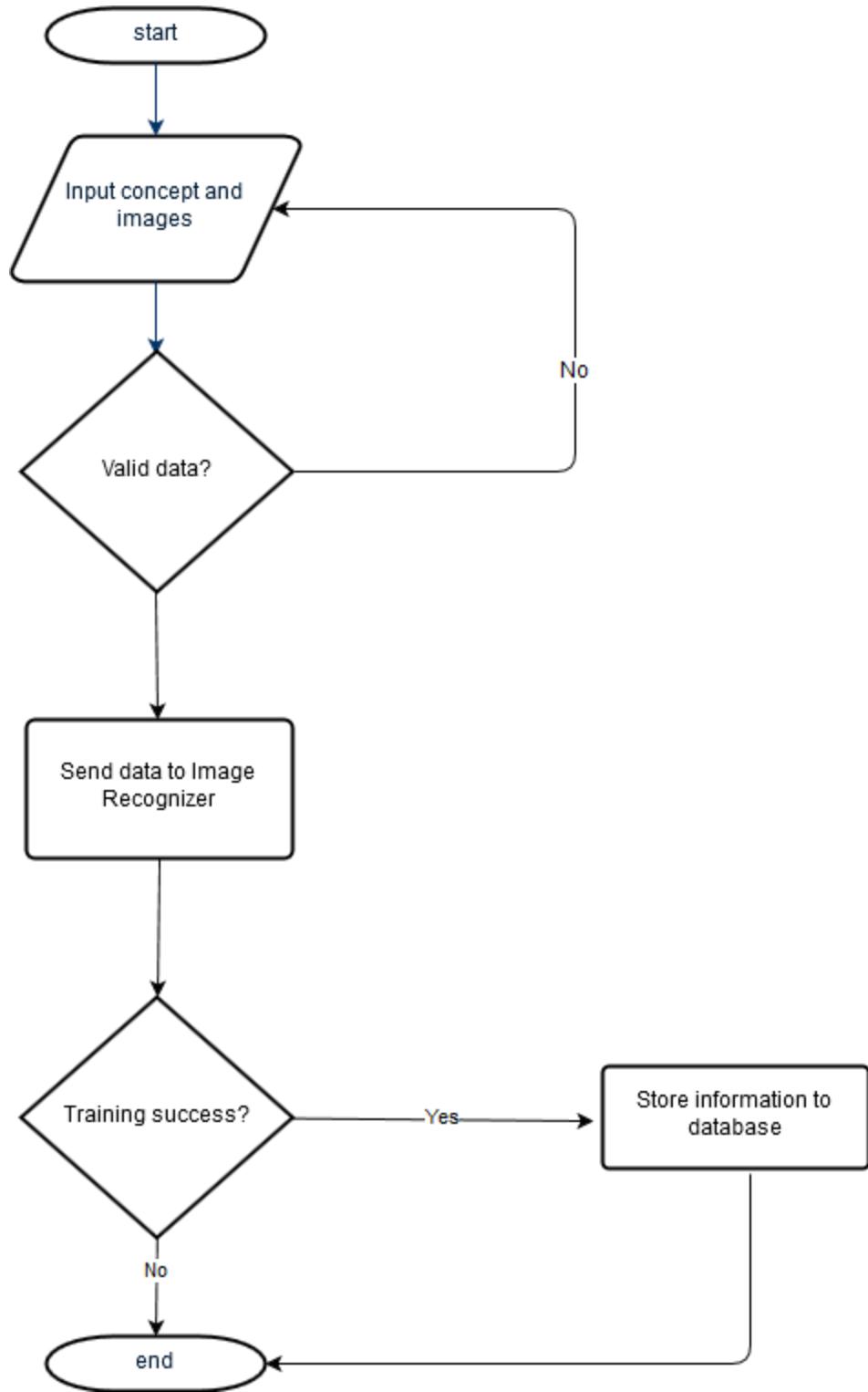
The training object process includes these steps:

- 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.
- Step 3: 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.

##### 4.7.9.5 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

**4.7.9.6 Flow chart**

## 5 System Implementation & Test

### 5.1 Introduction

#### 5.1.1 Overview

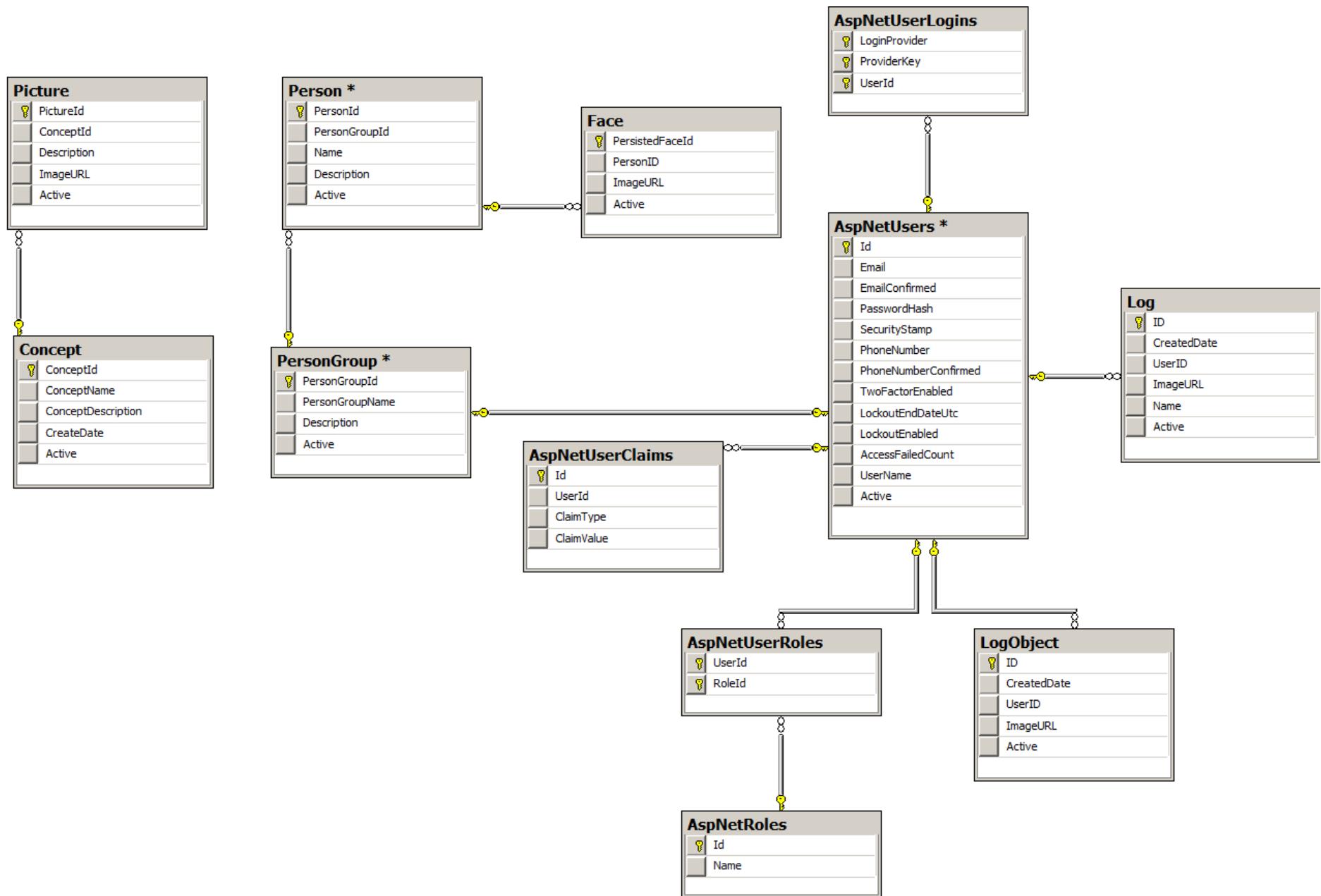
This section describes the approach and methodologies used by group to plan, organize and manage the testing of AAV system. It provides in the detail all necessary information about the implementation and testing procedure of the system included test plans, test cases, test result, test environments,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

#### 5.1.2 Test Approach

- Goal: Test all features in the whole AAV system based on the core flow.
- Method: black-box testing
- Technique: checklist

### 5.2 Database Relationship Diagram

#### 5.2.1 Physical Diagram



**Figure 67: Physical Diagram****5.2.2 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 information 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)	No
	ImageURL	Link of image	Nvarchar(MAX)	No
	Active	The status of Department: False: DE-ACTIVE True: ACTIVE	Bit	No

Concept	ConceptId{PK}	Unique identifier for Concept	Int	No
	ConceptName	Name of the concept	Nvarchar(100)	No
	ConceptDescription	Description of concept	Nvarchar(100)	No
	CreateDate	Created date	Datetime	Yes
	Active	The status of Concept: False: DE-ACTIVE True: ACTIVE	Bit	No
Person	PersonId{PK}	Unique identifier for Person	Nvarchar(36)	No
	PersonGroupId{FK}	FK reference to 'PersonGroup' table	Nvarchar(128)	No
	Name	Name of the person	Nvarchar(128)	No
	Description	Description of the person	Nvarchar(250)	Yes
	Active	The status of Person: False: DE-ACTIVE True: ACTIVE	Bit	No
PersonGroup	PersonGroupId{PK}	Unique identifier for PersonGroup	Nvarchar(128)	No
	PersonGroupName	Name of the person group	Nvarchar(128)	No
	Description	Description of the person group	Nvarchar(250)	Yes
	Active	The status of PersonGroup: False: DE-ACTIVE True: ACTIVE	Bit	No
Face	PersistedFaceId{PK}	Unique identifier for Face	Nvarchar(36)	No
	PersonID{FK}	FK reference to 'Person' table	Nvarchar(36)	No
	ImageURL	Url of image contain this face	Nvarchar(MAX)	No
	Active	The status of Face: False: DE-ACTIVE True: ACTIVE	Bit	No
AspNetUsers	Id{PK}	Unique identifier for User	Nvarchar(128)	No
	Email	Email of User	Nvarchar(256)	Yes
	EmailConfirmed	The status email confirmed or not: False: Not confirm Yes: Confirmed	Bit	No

	PasswordHash	Password with Hash	nvarchar(MAX)	Yes
	SecurityStamp	Security Stamp	nvarchar(MAX)	Yes
	PhoneNumber	Phone number of User	nvarchar(MAX)	Yes
	PhoneNumberConfirmed	The status phone confirmed or not: False: Not confirm True: Confirmed	bit	No
	TwoFactorEnabled	Two-factor authentication	bit	No
	LockoutEndDateTimeUtc	Enddate of lockout user	DateTime	Yes
	LockoutEnabled	User can be locked out, not that the user is locked out	bit	No
	AccessFailedCount	Access failed count	int	No
	UserName	Username of user	nvarchar(256)	No
	Active	The status of AspNetUsers: False: DE-ACTIVE True: ACTIVE	bit	No
Log	Id{PK}	Unique identifier for Log Person	Int	No
	CreatedDate	Created date	DateTime	No
	UserID{FK}	FK reference to 'AspNetUsers' table	Nvarchar(128)	No
	ImageURL	Image URL of person was not detected	Nvarchar(MAX)	No
	Name	Name of that person	Nvarchar(128)	Yes
	Active	The status of Log: False: DE-ACTIVE True: ACTIVE	Bit	No
LogObject	ID{PK}	Unique identifier for LogObject	Int	No
	CreatedDate	Created date	DateTime	No
	UserID{FK}	FK reference to 'AspNetUsers' table	nvarchar(128)	No
	ImageURL	Image url of object detect fail	nvarchar(MAX)	No
	Active	The status of LogObject: False: DE-ACTIVE True: ACTIVE	bit	No
	AspNetUserRoles	UserId{PK}	PK reference to 'AspNetUsers' table	nvarchar(128)

	RoleId{PK}	PK reference to 'AspNetRoles' table	nvarchar(128)	No
AspNetRole s	Id{PK}	Unique identifier for AspNetRoles	Nvarchar(128)	No
	Name	Name of Role	Nvarchar(256)	No

### 5.3 Performance Measures

#### 5.4 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.

#### 5.5 System Testing Test Case

##### 5.5.1 Communication Diagram

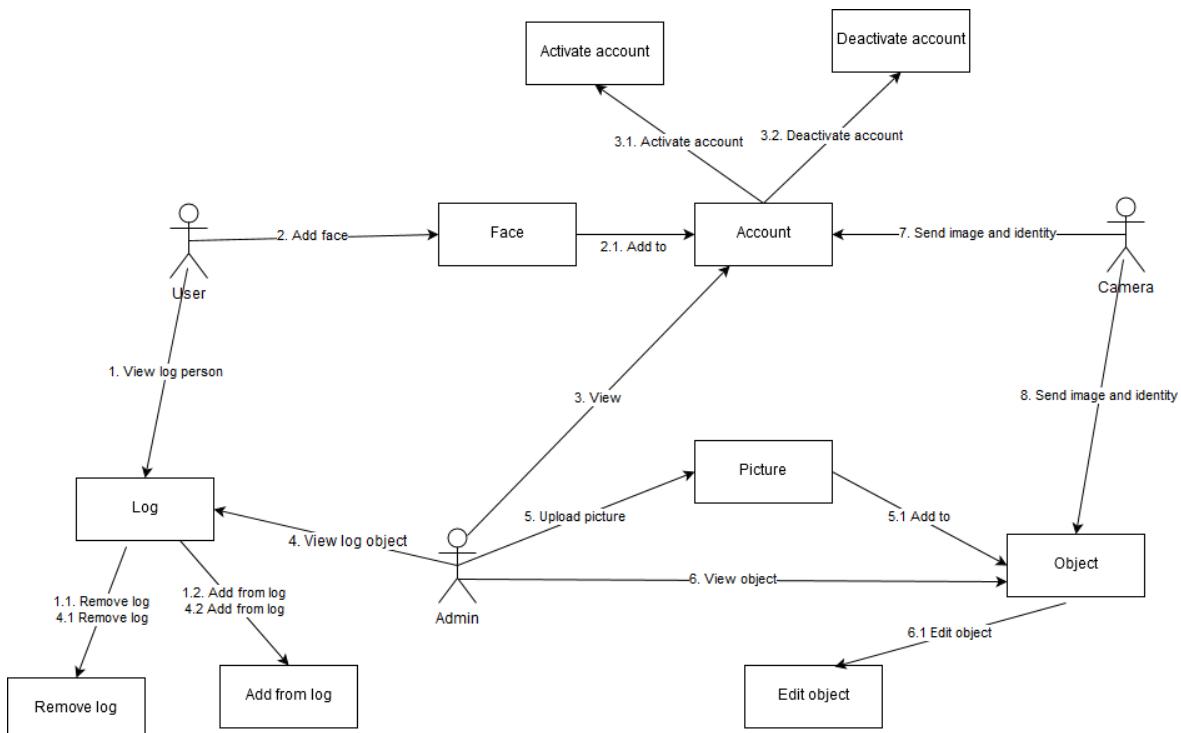


Figure 68: Communication Diagram

##### 5.5.2 Test Cases

###### 5.5.2.1 Mobile Application Test Cases

###### 5.5.2.1.1 <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 shows the	Pass: 30 Fail: 2	01/04/2017 to 13/04/2017

		mobile application. 2. Say detect command. 3. Take picture.	name of the person was trained.		
UDP02	Detect person was not trained.	1. Login successfully with role “User” in mobile application. 2. Send detect command. 3. Take picture.	Mobile application shows the gender of the person.	Pass:27 Fail: 1	01/04/2017 to 13/04/2017
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 shows the name of the person.	Pass:20 Fail:10	01/04/2017 to 13/04/2017
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 shows the name of the person.	Pass:3 Fail:26	01/04/2017 to 13/04/2017
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 shows the name of the person.	Pass:0 Fail:28	01/04/2017 to 13/04/2017
UDP06	Detect person while having no one in picture.	1. Login successfully with role “User” in mobile application. 2. Send detect command. 3. Take picture.	Mobile application show does not have any person message.	Pass:30 Fail:0	01/04/2017 to 13/04/2017

UDP07	Detect person while the person wears glasses.	1. Login successfully with role “User” in mobile application. 2. Send detect command. 3. Take picture.	Mobile application shows the name of the person.	Pass:20 Fail:3	01/04/2017 to 13/04/2017
UDP08	Detect person from behind.	1. Login successfully with role “User” in mobile application. 2. Send detect command. 3. Take picture.	Mobile application shows the name of the person.	Pass:0 Fail:18	01/04/2017 to 13/04/2017
UDP09	Detect person while picture contains ½ face.	1. Login successfully with role “User” in mobile application. 2. Send detect command. 3. Take picture.	Mobile application shows the name of the person.	Pass:0 Fail:17	01/04/2017 to 13/04/2017
UDP10	Detect person while picture contains 1/3 face.	1. Login successfully with role “User” in mobile application. 2. Send detect command. 3. Take picture.	Mobile application shows the name of the person.	Pass:0 Fail:18	01/04/2017 to 13/04/2017
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 shows the name of the person.	Pass:28 Fail:1	01/04/2017 to 13/04/2017
UDP12	Detect person from a distance of 1.5 meters.	1. Login successfully with role “User” in mobile application.	Mobile application shows the name of the person.	Pass:27 Fail:2	01/04/2017 to 13/04/2017

		2. Send detect command. 3. Take picture.			
UDP13	Detect person from a distance of 2 meters.	1. Login successfully with role “User” in mobile application. 2. Send detect command. 3. Take picture.	Mobile application shows the name of the person.	Pass:20 Fail:5	01/04/2017 to 13/04/2017
UDP14	Detect person from a distance of 2.5 meters.	1. Login successfully with role “User” in mobile application. 2. Send detect command. 3. Take picture.	Mobile application shows the name of the person.	Pass:6 Fail:17	01/04/2017 to 13/04/2017
UDP15	Detect person from a distance of 3 meters.	1. Login successfully with role “User” in mobile application. 2. Send detect command. 3. Take picture.	Mobile application shows the name of the person.	Pass:0 Fail:18	01/04/2017 to 13/04/2017
UDP16	Take picture contain 2 people(1 trained and 1 not trained)	1. Login successfully with role “User” in mobile application. 2. Send detect command. 3. Take picture.	Mobile application shows the name of the person was trained.	Pass:24 Fail:0	01/04/2017 to 13/04/2017
UDP17	Take picture contain 2 persons was trained.	1. Login successfully with role “User” in mobile application. 2. Send detect command. 3. Take picture.	Mobile application shows the name of 2 persons.	Pass:24 Fail:2	01/04/2017 to 13/04/2017
UDP18	Take picture contain 3	1. Login successfully with role “User” in	Mobile application shows the	Pass:14 Fail:2	01/04/2017 to 13/04/2017

	persons was trained.	mobile application. 2. Send detect command. 3. Take picture.	name of 3 persons.		
UDP19	Take picture contain 4 persons was trained.	1. Login successfully with role “User” in mobile application. 2. Send detect command. 3. Take picture.	Mobile application shows the name of 4 persons.	Pass:5 Fail:0	01/04/2017 to 13/04/2017

**Table 84: Mobile Application Test Cases - <User> Detect Person****5.5.2.1.2 <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 shows the name of the object.	Pass:27 Fail: 3	01/04/2017 to 13/04/2017
UDO02	Detect object was not trained.	1. Login successfully with role “User”. 2. Send detect object command.	Mobile application shows the name of the object.	Pass:3 Fail: 17	01/04/2017 to 13/04/2017
UDO03	Detect object from a distance of 1 meter.	1. Login successfully with role “User”. 2. Send detect object command.	Mobile application shows the name of the object.	Pass:6 Fail: 2	01/04/2017 to 13/04/2017
UDO04	Detect object from a distance of 1.5 meters.	1. Login successfully with role “User”. 2. Send detect object command.	Mobile application shows the name of the object.	Pass:3 Fail: 3	01/04/2017 to 13/04/2017
UDO05	Detect object from a distance of 2 meters.	1. Login successfully with role “User”. 2. Send detect object command.	Mobile application shows the name of the object.	Pass:5 Fail: 8	01/04/2017 to 13/04/2017
UDO06	Detect object from	1. Login successfully with role “User”.	Mobile application shows the	Pass:0 Fail: 5	01/04/2017 to 13/04/2017

	a distance of 2.5 meters.	2. Send detect object command.	name of the object.		
UDO07	Detect object from a distance of 3 meters.	1. Login successfully with role “User”. 2. Send detect object command.	Mobile application shows the name of the object.	Pass:0 Fail: 5	01/04/2017 to 13/04/2017
UDO08	Take picture contain 2 objects(1 trained and 1 not trained).	1. Login successfully with role “User”. 2. Send detect object command.	Mobile application shows the name of the object was trained.	Pass:1 Fail: 6	01/04/2017 to 13/04/2017
UDO09	Take picture contain 2 objects was trained.	1. Login successfully with role “User”. 2. Send detect object command.	Mobile application shows the name of 2 objects.	Pass:0 Fail: 6	01/04/2017 to 13/04/2017
UDO10	Take picture contain 3 objects (1 trained and 2 not trained).	1. Login successfully with role “User”. 2. Send detect object command.	Mobile application shows the name of the object was trained.	Pass:0 Fail: 5	01/04/2017 to 13/04/2017

*Table 85: Mobile Application Test Cases - <User> Detect Object*

#### 5.5.2.1.3 <User> Detect Person Streaming

ID	Test Case Description	Test case procedure	Expected output	Result	Test Date
UDPS01	Streaming contains no face in 30 seconds.	1. Login successfully with role “User”. 2. Send streaming command.	Mobile application not takes picture or detect.	Pass:10 Fail: 0	01/04/2017 to 13/04/2017

*Table 86: Mobile Application Test Cases - <User> Detect Person Streaming*

#### 5.5.2.2 Web Application Test Cases

##### 5.5.2.2.1 <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.	Pass:8 Fail:0	01/04/2017 to 13/04/2017

**Table 87: Web Application Test Case - <Admin> View List Accounts****5.5.2.2.2 <Admin> Activate/Deactivate Account**

<b>ID</b>	<b>Test Case Description</b>	<b>Test case procedure</b>	<b>Expected output</b>	<b>Result</b>	<b>Test Date</b>
AADA01	Activate specific account success	1. Login successfully with role “Admin”. 2. In “Quản lý tài khoản” view, select an account in list and click “Activate” button. 3. Click “Cập nhật” button.	Web application shows “Cập nhật thành công” message.	Pass:7 Fail:0	01/04/2017 to 13/04/2017
AADA02	Activate specific account but cancels	1. Login successfully with role “Admin”. 2. In “Quản lý tài khoản” view, select a deactivated account in the list and click “Deactivate” button. 3. Refresh page without click “Cập nhật” button.	Web application does nothing, account still locked.	Pass:5 Fail:0	01/04/2017 to 13/04/2017
AADA03	Deactivate specific account success	1. Login successfully with role “Admin”. 2. In “Quản lý tài khoản” view, select an activated account in the list and click “Deactivate” button. 3. Click “Cập nhật” button when it shows up.	Web application shows pop up with message: “Khóa tài khoản thành công”.	Pass:6 Fail:0	01/04/2017 to 13/04/2017

**Table 88: Web Application Test Case - <Admin> Activate/Deactivate Account**

## 6 Software User's Manual

### 6.1 Installation Guide

#### 6.1.1 Setting up environment at server side

The specifications are based on the dependencies requirements and performance test result from the previous section of this document.

#### 6.1.1.1 Hardware Requirements

Hardware	Specification
Internet Connection	4 Mbps
Computer Processor	Intel® Core i3 1.4GHz or more
Computer Memory	2GB of RAM or more
Hard Drive	30GB or more

*Table 89: Installation Guide - Hardware Requirements*

#### 6.1.1.2 Software Requirements

Software	Application name/version
Operating System	Windows 7 Professional or later
.Net	4.5
DBMS	Microsoft SQL Server 2014

*Table 90: Installation Guide - Software Requirement*

#### 6.1.1.3 Mobile Requirements

Hardware	Specification
CPU	At least 1.8 GHz
Memory	At least 512 MB. Recommended: 1GB
Storage	At least 40Mb. Recommended: 1 GB
Camera	At least 3.0 MP Recommended: 5.0 MP
Model	All Android devices with android version 5.0 or higher.

*Table 91: Installation Guide - Mobile Requirement*

## 6.1.2 Web Services Application Deployment Process

### 6.1.2.1 Check Environment

- Check .Net Version

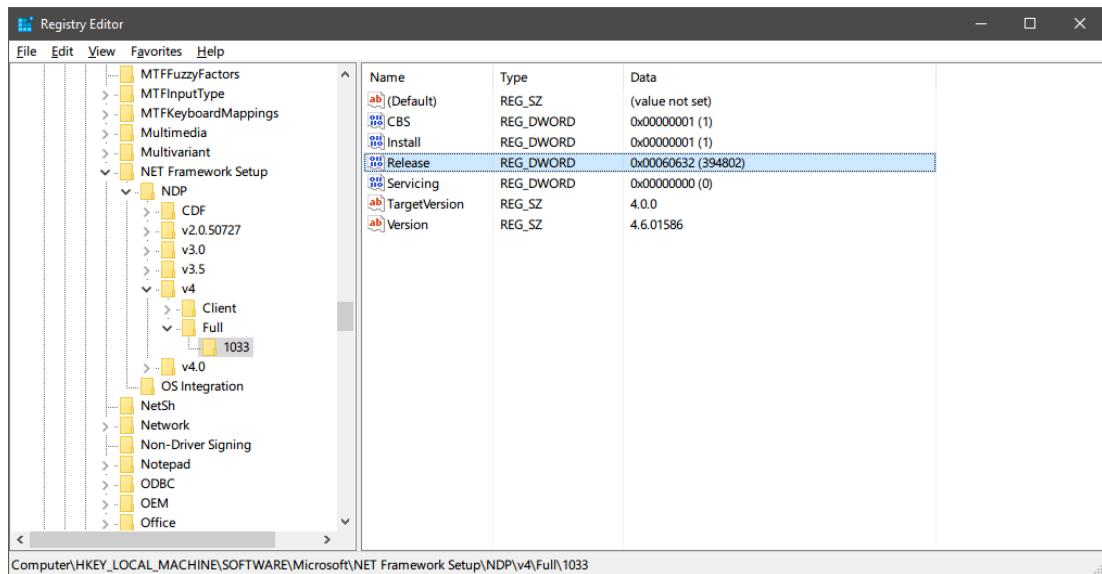
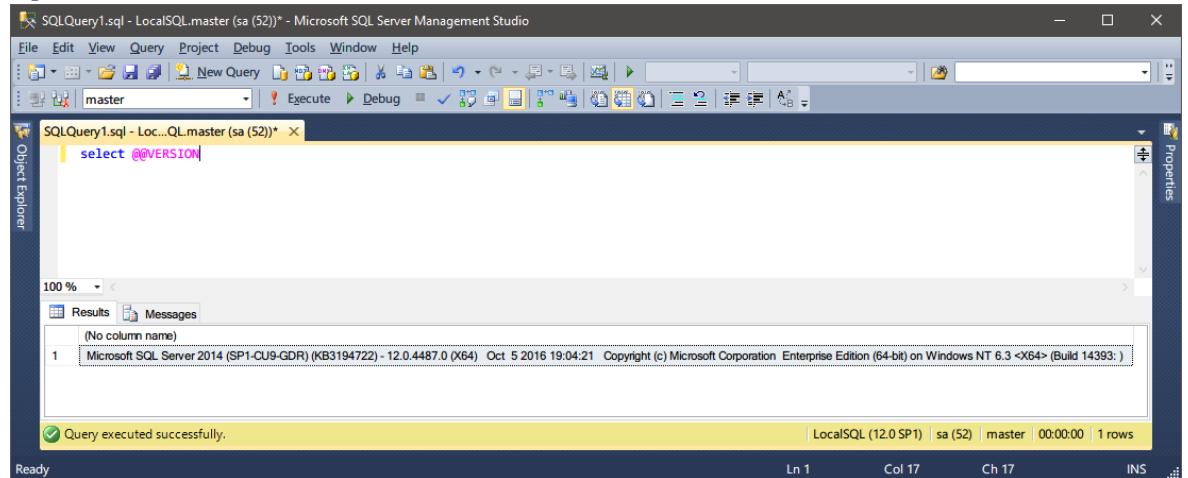


Figure 69: Checking .Net Version

Version	Value of the Release Data Word
.NET Framework 4.5	378389
.NET Framework 4.5.1 installed with Windows 8.1	378675
.NET Framework 4.5.1 installed on Windows 8, Windows 7 SP1, or Windows Vista SP2	378758
.NET Framework 4.5.2	379893
.NET Framework 4.6 installed with Windows 10	393295
.NET Framework 4.6 installed on all other Windows OS versions	393297
.NET Framework 4.6.1 installed on Windows 10	394254
.NET Framework 4.6.1 installed on all other Windows OS versions	394271
.NET Framework 4.6.2 installed on Windows 10 Anniversary Update	394802
.NET Framework 4.6.2 installed on all other Windows OS versions	394806
.NET Framework 4.7 installed on Windows 10 Creators Update	460798

Table 92: .Net Version Release Code

- Open Microsoft SQL Server and check Microsoft SQL Server version



The screenshot shows the Microsoft SQL Server Management Studio interface. A query window titled 'SQLQuery1.sql - Loc...QL.master (sa (52))' contains the command 'select @@VERSION'. The results pane shows a single row of output:

```
Microsoft SQL Server 2014 (SP1-CU9-GDR) (KB3194722) - 12.0.4487.0 (X64) Oct 5 2016 19:04:21 Copyright (c) Microsoft Corporation Enterprise Edition (64-bit) on Windows NT 6.3 <X64> (Build 14393:)
```

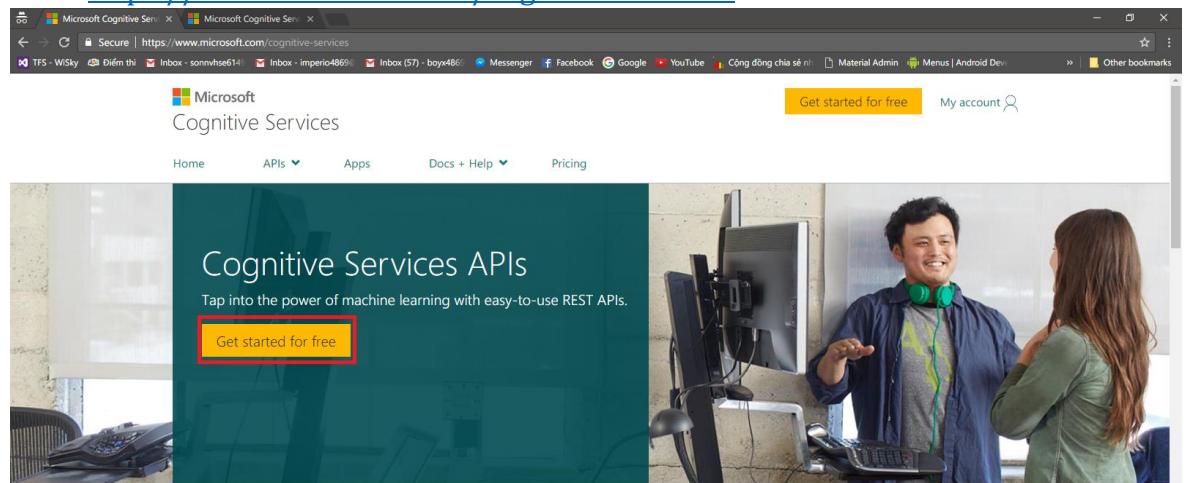
Below the results, a message bar indicates 'Query executed successfully.' The status bar at the bottom right shows 'Ln 1 Col 17 Ch 17 INS'.

*Figure 70: Checking Microsoft SQL Server Version*

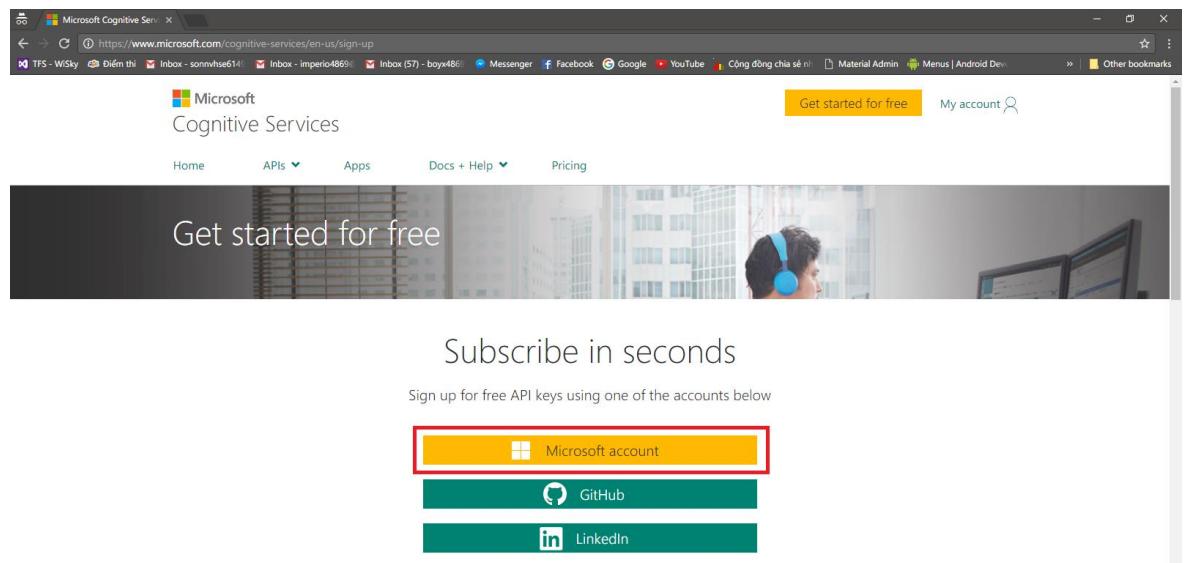
### 6.1.2.2 Prepare Image Recognize API

#### 6.1.2.2.1 Prepare Microsoft Cognitive Service API

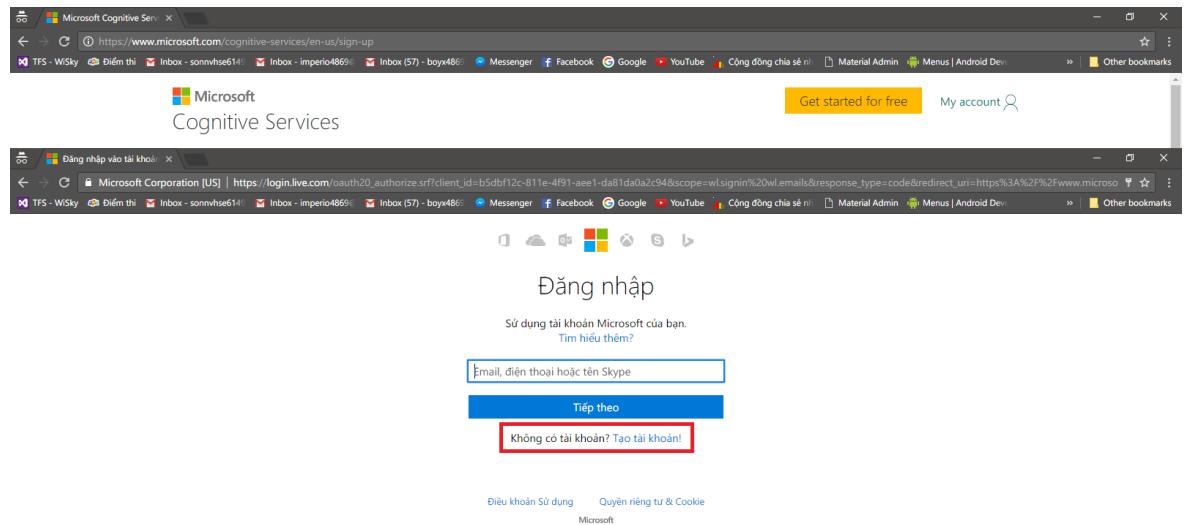
- Go to <https://www.microsoft.com/cognitive-services> and create an account.



*Figure 71: Microsoft Cognitive Service Home Page*



Click “Tạo tài khoản” to create new Microsoft Account



- Fill all necessary information and click “Tạo Tài khoản”

**Tạo một tài khoản**

Bạn có thể sử dụng địa chỉ email bất kỳ làm tên người dùng cho tài khoản Microsoft mới của mình, bao gồm các địa chỉ từ Outlook.com, Yahoo! hoặc Gmail. Nếu bạn đã đăng nhập vào một Windows PC, máy tính bảng hoặc điện thoại, Xbox Live, Outlook.com hoặc OneDrive, hãy sử dụng tài khoản đó để **đăng nhập**.

Họ	Tên
Nguyễn	Sơn

Tên đăng nhập

Nhận địa chỉ email mới

Sau khi bạn đăng ký, chúng tôi sẽ gửi thông báo cho bạn với một liên kết để xác nhận tên người dùng này.

Mật khẩu

ít nhất là 8 ký tự; có phân biệt chữ hoa, chữ thường

Nhập lại mật khẩu

Quốc gia/Vùng lãnh thổ

Ngày sinh

18	Tháng Một	1999
----	-----------	------

Giới tính

Giúp chúng tôi bảo vệ thông tin của bạn

Số điện thoại của bạn giúp chúng tôi giữ an toàn cho tài khoản của bạn.

Mã quốc gia

Số điện thoại

Trước khi tiếp tục, chúng tôi muốn chắc chắn rằng người đang tạo tài khoản là người thực.

Mới
Âm thanh

Nhập các ký tự bạn nhìn thấy

Gửi cho tôi các ưu đãi quảng cáo từ Microsoft. Bạn có thể bỏ đăng ký bất kỳ lúc nào.

Bấm Tạo tài khoản nghĩa là bạn đồng ý với [Thỏa thuận Dịch vụ của Microsoft](#) và điều [kiện và quyền riêng tư và cookie](#).

**Tạo tài khoản**

*Figure 72: Register Page*

- After successfully created an Microsoft Account, go to <https://www.microsoft.com/cognitive-services/en-us/face-api> to get an Face Review API key.

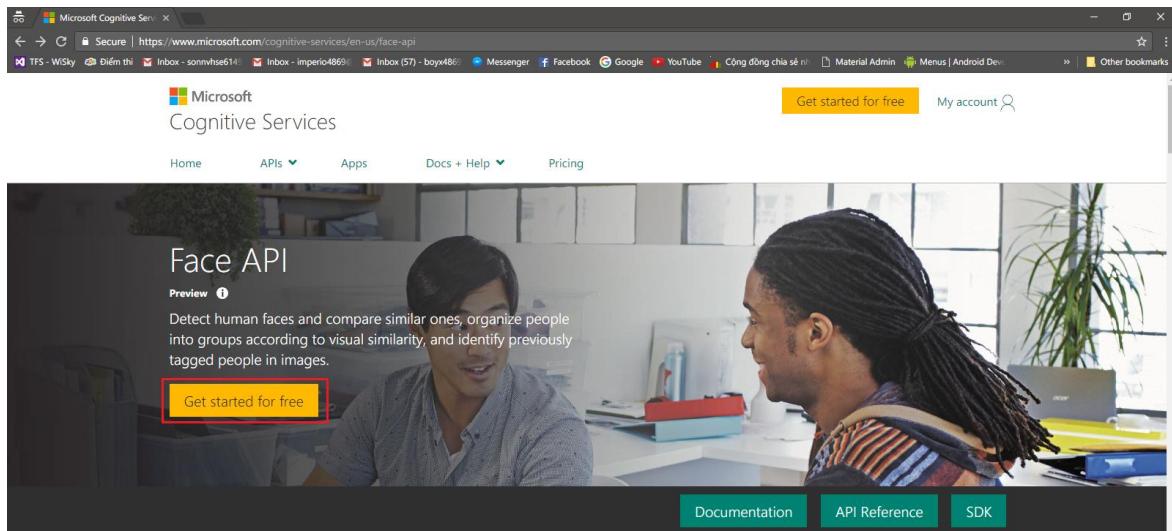


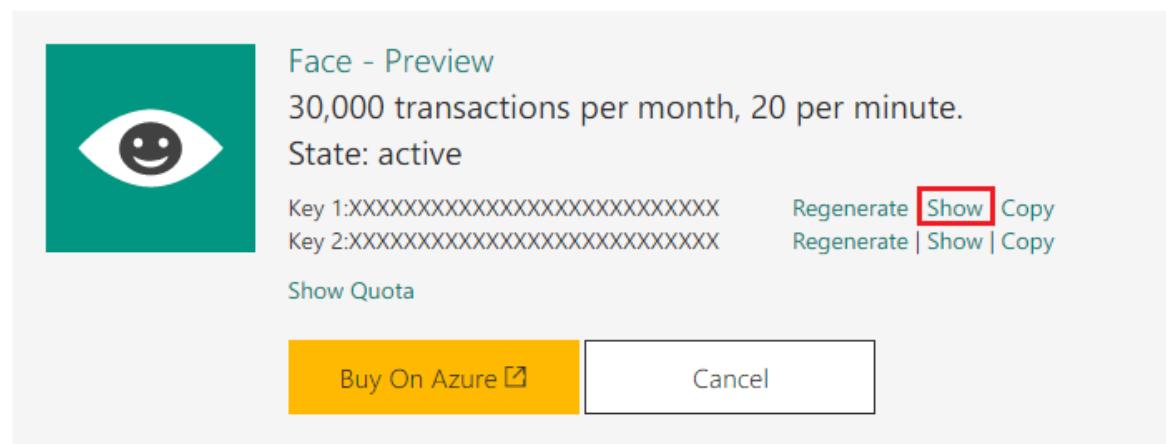
Figure 73: Microsoft Face Review Home Page

- Face - Preview      30,000 transactions per month, 20 per minute.
- I agree to the [Microsoft Cognitive Services Terms](#) and [Microsoft Privacy Statement](#).
- Contact me with promotional offers and updates about Microsoft Cognitive Services.



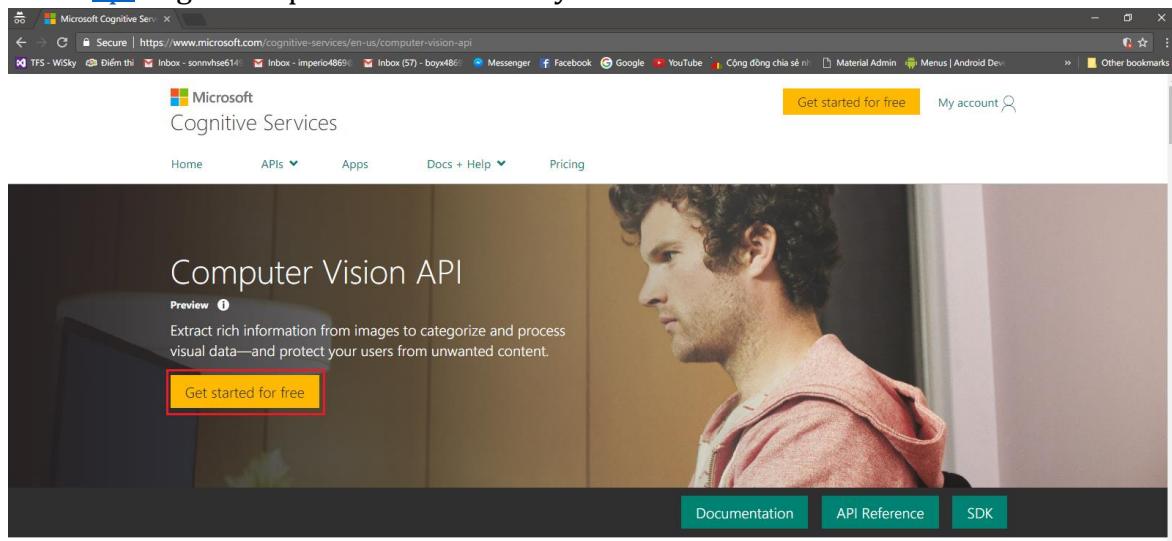
Figure 74: Subscribe for Face Review API

- Click Show to get API key



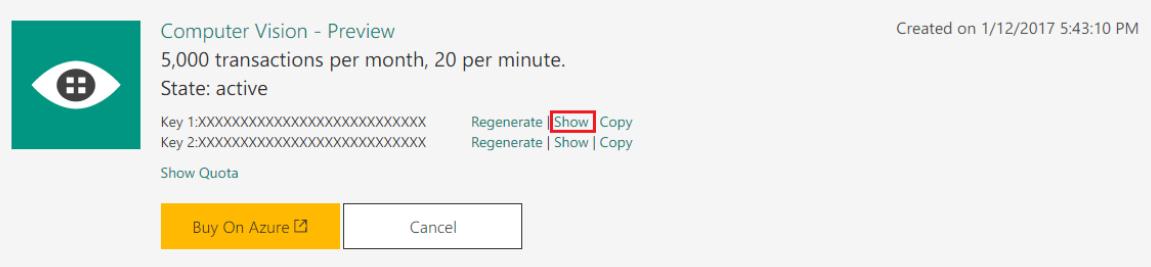
*Figure 75: Face Review API Key*

- Go to <https://www.microsoft.com/cognitive-services/en-us/computer-vision-api> to get Computer-Vision API key



*Figure 76: Microsoft Computer Vision Home Page*

- Click Show to review the API key



*Figure 77: Computer Vision API Key*

- Open web service project and edit file Assets.cs

```
public const string KEY_COMPUTER_VISION = "XXXXXXXXXXXXXXXXXXXXXX";  
public const string KEY_FACE_REVIEW = "XXXXXXXXXXXXXXXXXXXXXX";
```

*Figure 78: Update Application configurations for Microsoft Cognitive Service*

#### 6.1.2.2.2 Prepare Clarifai API

- Go to Home page Clarifai <https://developer.Clarifai.com/>

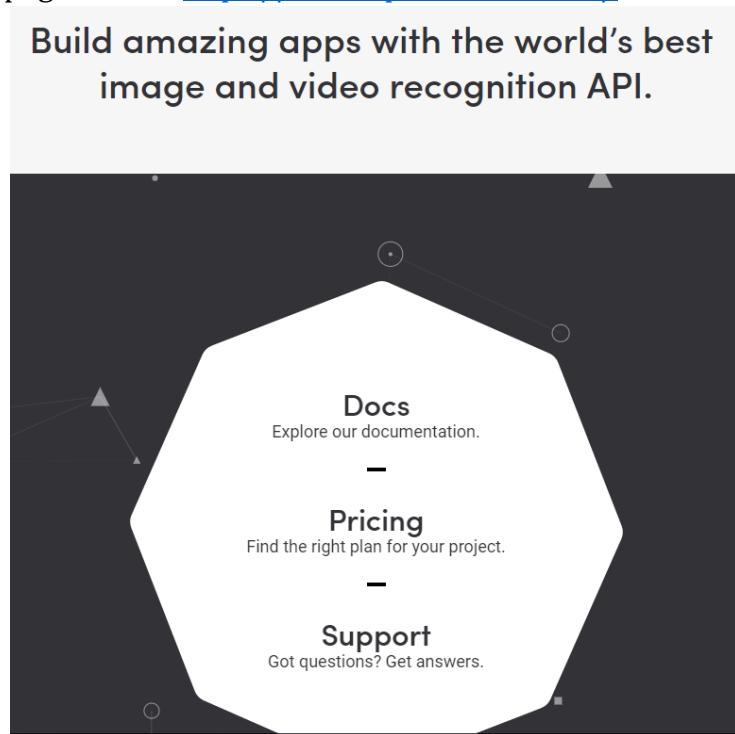


Figure 79: Clarifai Home Page

- Click [Sing Up] button to register account and fill information and check box

The image shows the sign-up form for a Clarifai account. It includes fields for First name (duy), Last name (nguyen), Email address (duynguyen@gmail.com), Password (redacted), Company (Company (optional)), and Signup Code (Code (optional)). Below these fields is a reCAPTCHA box with a checked checkbox labeled 'I'm not a robot'. At the bottom, there is a checkbox for accepting the Terms of service and a 'SIGN UP' button. A link to 'Log in' is also present at the bottom.

Figure 80: Clarifai Account Sign Up Page

- Click account to go to Manage Your Applications page



- Click [Create a New Applications] to create new app Clarifai

**Manage Your Applications**  
Click your application's name below to view Client Id, Secret, and other details.

**Create a New Application**

*Figure 81: Clarifai Manage Applications Page*

- Fill name application, then click [Create Application] button

**New Application**

Name\*

Default Model

Default Language

**Create Application**

*Figure 82: Create New Application Form*

- Get Client Id and Client Secret

**Manage Your Applications**

Name  
Capstone

Client Id  
[REDACTED]

Client Secret  
[REDACTED]

*Figure 83: Client ID and Secrets*

- Go to preview Clarifai page <https://preview.Clarifai.com/#/> and login

**Log in**

Email Address

Password

Remember me

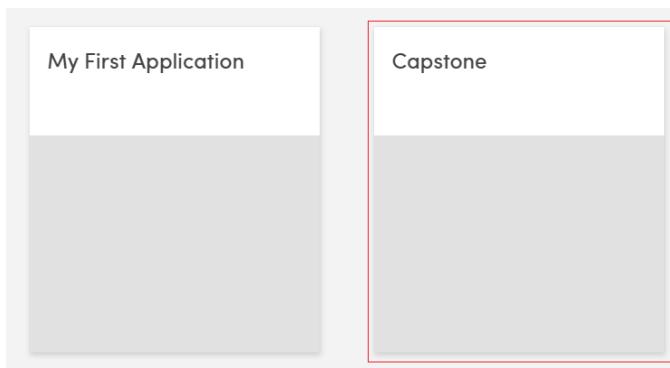
**LOG IN**

Forgot your password?

Don't yet have an account? [Sign up now](#)

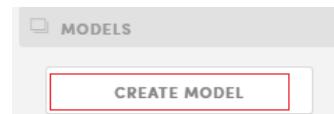
*Figure 84: Clarifai preview page*

- Click the previously created application



*Figure 85: Created Application*

- Click [CREATE MODEL] button



*Figure 86: Create Model button*

- Fill name of model after that click [Done] button to create new Model

A screenshot of a "Create new model" form. At the top, there is a text input field containing the text "Capstone". Below the input field are two toggle switches: "CONCEPTS MUTUALLY EXCLUSIVE" (OFF) and "CLOSED ENVIRONMENT" (OFF). At the bottom of the form is a large blue button with the text "Done" in white, which is also highlighted with a red border.

*Figure 87: Create new Model Form*

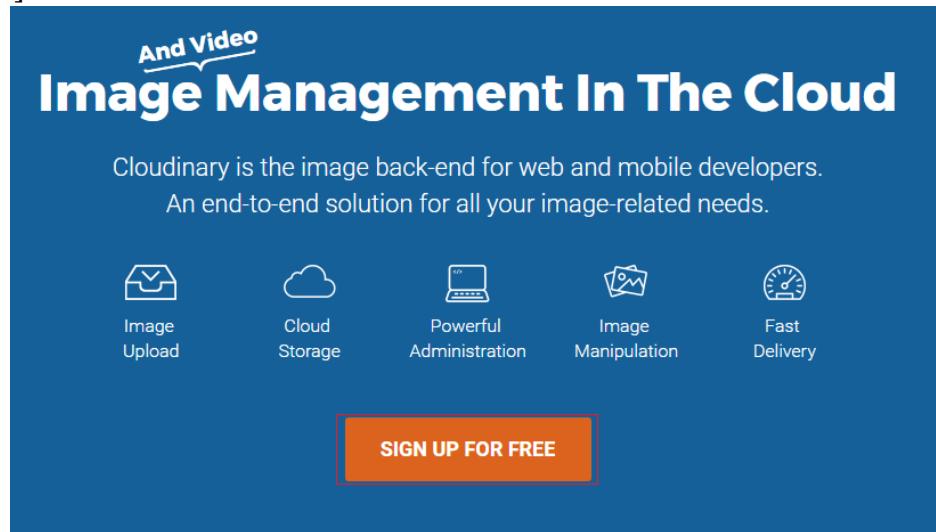
- Open and edit file ClarifaiAPI.py

```
17 CLARIFAI_CLIENT_ID = 'REDACTED'
18 CLARIFAI_CLIENT_SECRET = 'REDACTED'
```

*Figure 88: Update application configuration for Clarifai API*

### 6.1.2.3 Prepare Image Storage

- Go to Cloudinary home page: <http://cloudinary.com/> and click [SIGN UP FOR FREE] button



*Figure 89: Cloudinary Home Page*

- Fill in information and click button [Create Account] to create new account

 The image shows the sign-up form for Cloudinary. It includes fields for "Your name" (duynguyen), "E-mail" (duynguyen@gmail.com), "Password" (redacted), "Country" (Viet Nam), "Phone: (Optional)" (redacted), and "Company or site name: (Optional)" (redacted). Below the form, it says "Assigned cloud name: capstone" with an "Edit" link. A note at the bottom states "By clicking Create Account you agree to Cloudinary's Terms of Service and Privacy Policy". At the bottom is a large orange "CREATE ACCOUNT" button.

*Figure 90: Cloudinary Sign Up form*

- Click [Reveal] to show API Secret key

Cloud name:	<b>capstone</b>
API Key:	<b>923224617222248</b>
API Secret:	***** <a href="#">Reveal</a>

*Figure 91: Cloudinary API Key and Secrets*

- Open and edit file Assets.cs file

```
public const string CLOUDINARY_NAME = "████████";  
public const string CLOUDINARY_API_KEY = "████████";  
public const string CLOUDINARY_API_SECRET = "████████";
```

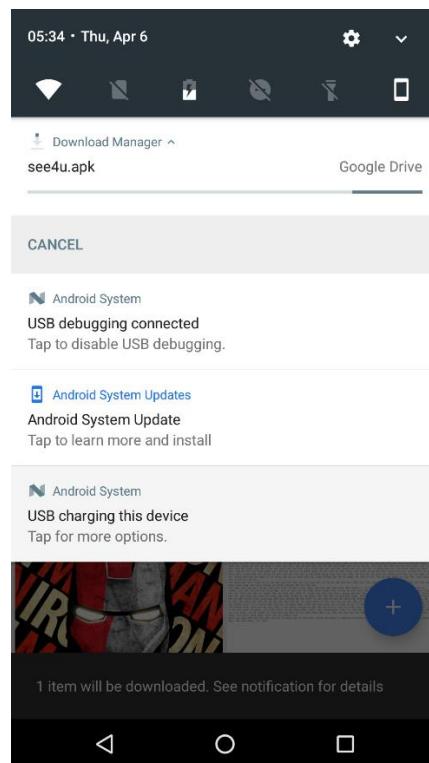
*Figure 92: Update application configuration for Image Storage*

#### 6.1.2.4 Deploying web application

#### 6.1.3 Mobile Application Deployment Process

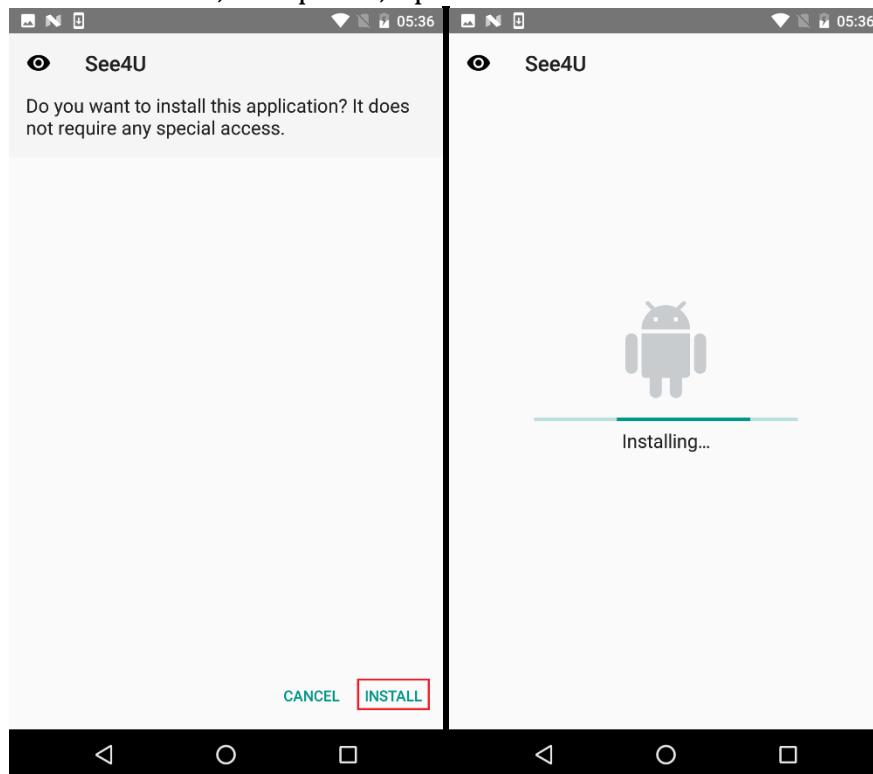
- Download the APK file from followed link:

[https://drive.google.com/file/d/0B2eM0JypmzdDcGZxQzJyZHvneWM/view?  
usp=sharing](https://drive.google.com/file/d/0B2eM0JypmzdDcGZxQzJyZHvneWM/view?usp=sharing)



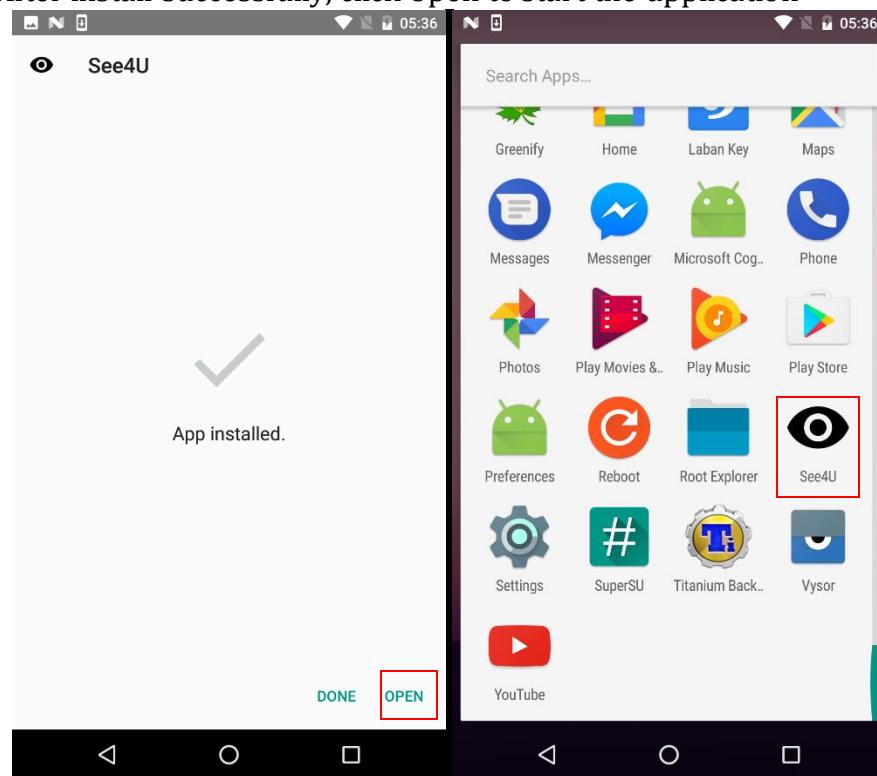
*Figure 93: Download APK file*

- After download, the apk file, open it to install.



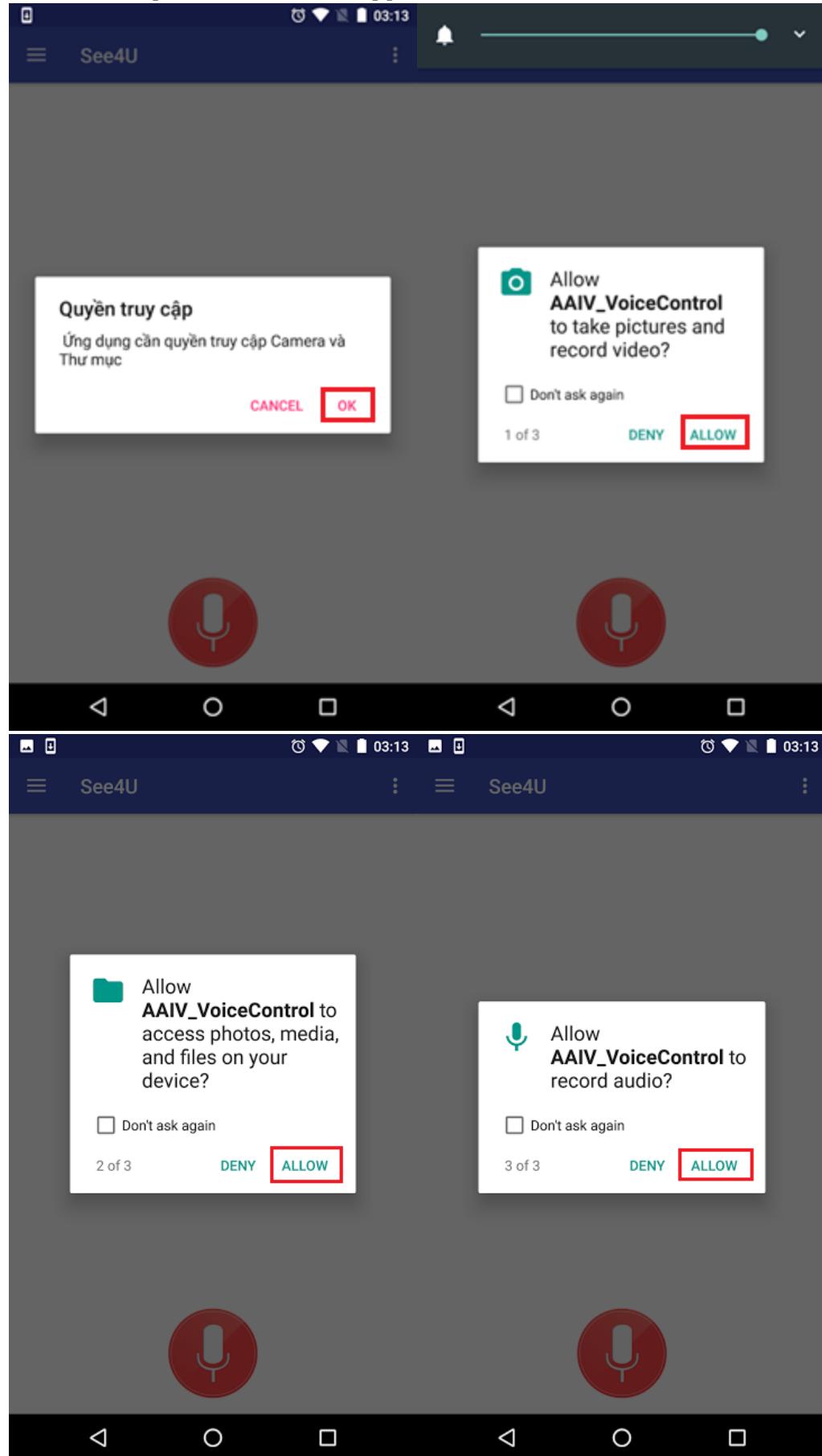
*Figure 94: Install APK file*

- After install successfully, click Open to start the application



*Figure 95: Finish mobile application installation*

- Provide access permission for the application:



## 6.2 User Guide

### 6.2.1 Mobile Application Guide

#### 6.2.1.1 Unauthorized User

##### 6.2.1.1.1 <Unauthorized User> Login

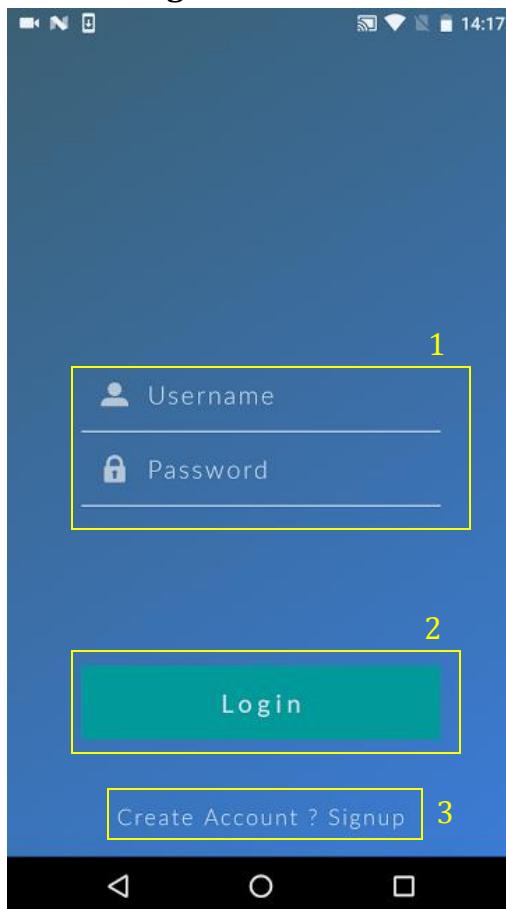
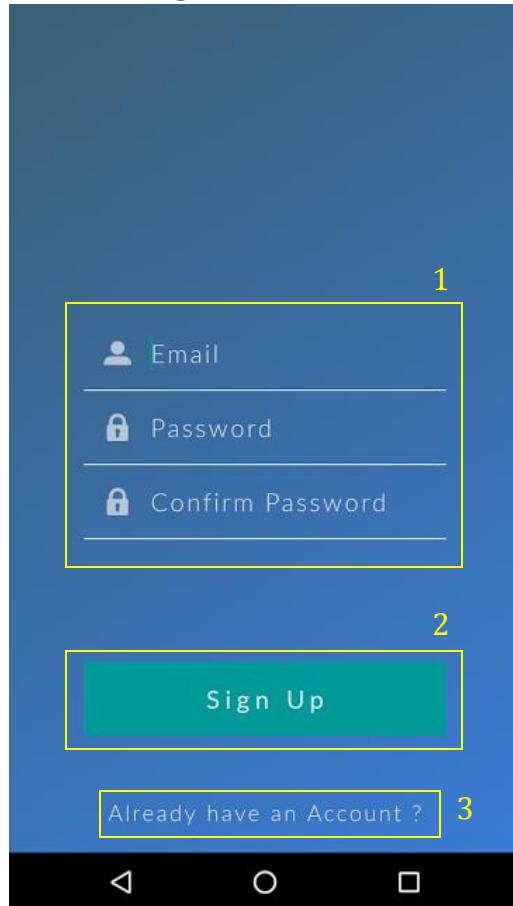


Figure 96: Mobile Application Login Screen

Step	Description
1	Fill in Email and Password
2	Touch on “Login” to login
3	Touch on “Create Account? Signup” to create new account

### 6.2.1.1.2 <Unauthorized User> Register



*Figure 97: Mobile Application Register screen*

Step	Description
1	Fill in Email and Password and Confirm Password
2	Touch on “Sign Up” to create new account
3	Touch on “Already have an Account?” to redirect to Login Screen

### 6.2.1.2 Authorized User

#### 6.2.1.2.1 <Authorized User> Detect Person

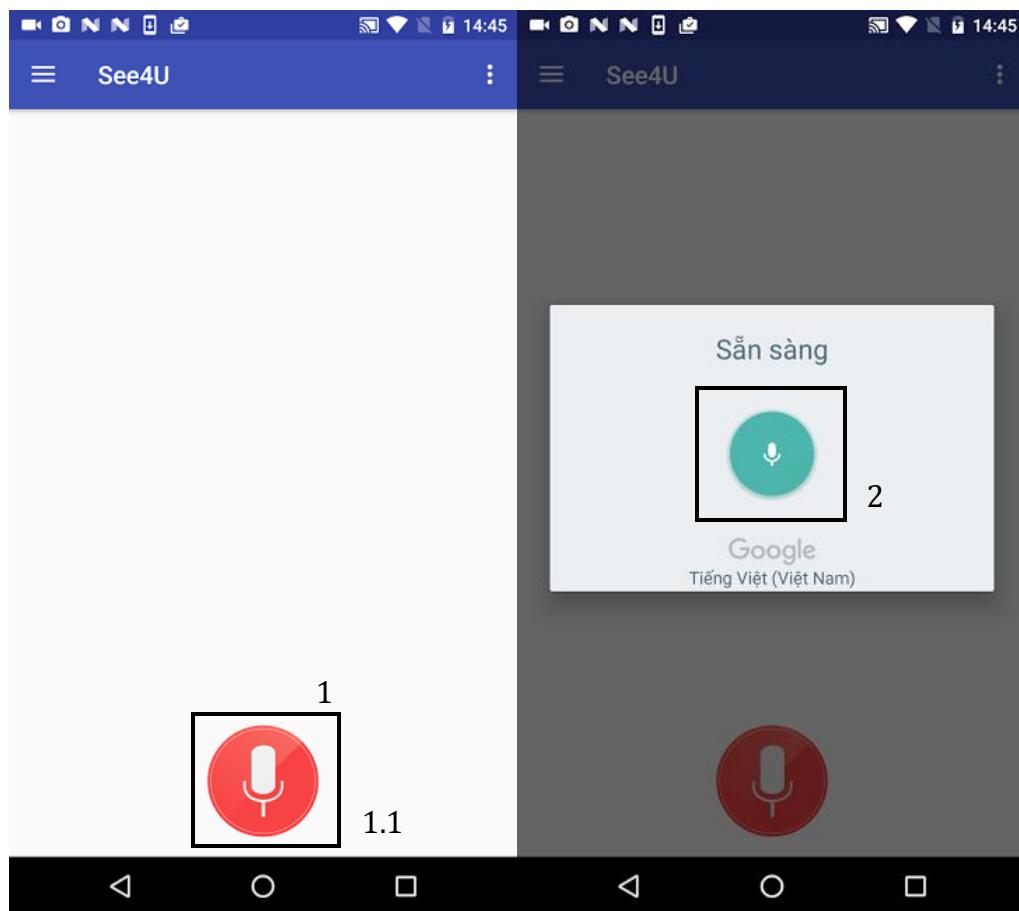
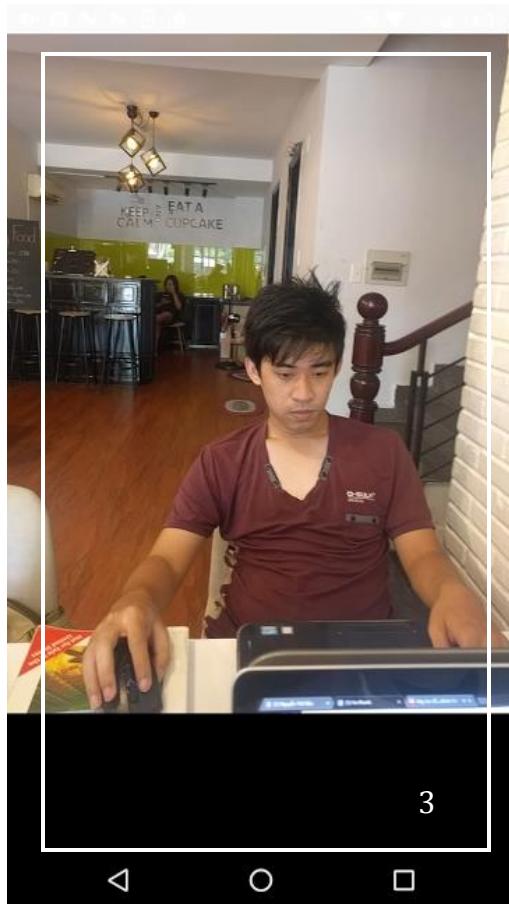


Figure 98: Mobile Application Main screen

Step	Description
1	Touch the Voice Command button to begin speaking command
1.1	Shake device 2 times to begin speaking command
2	Speak Detect Person Command (edit command in “Thiết Lập” menu)



*Figure 99: Mobile Application Camera screen*

Step	Description
3	Touch screen to capture image.

- Identify Result of an Unknown Person



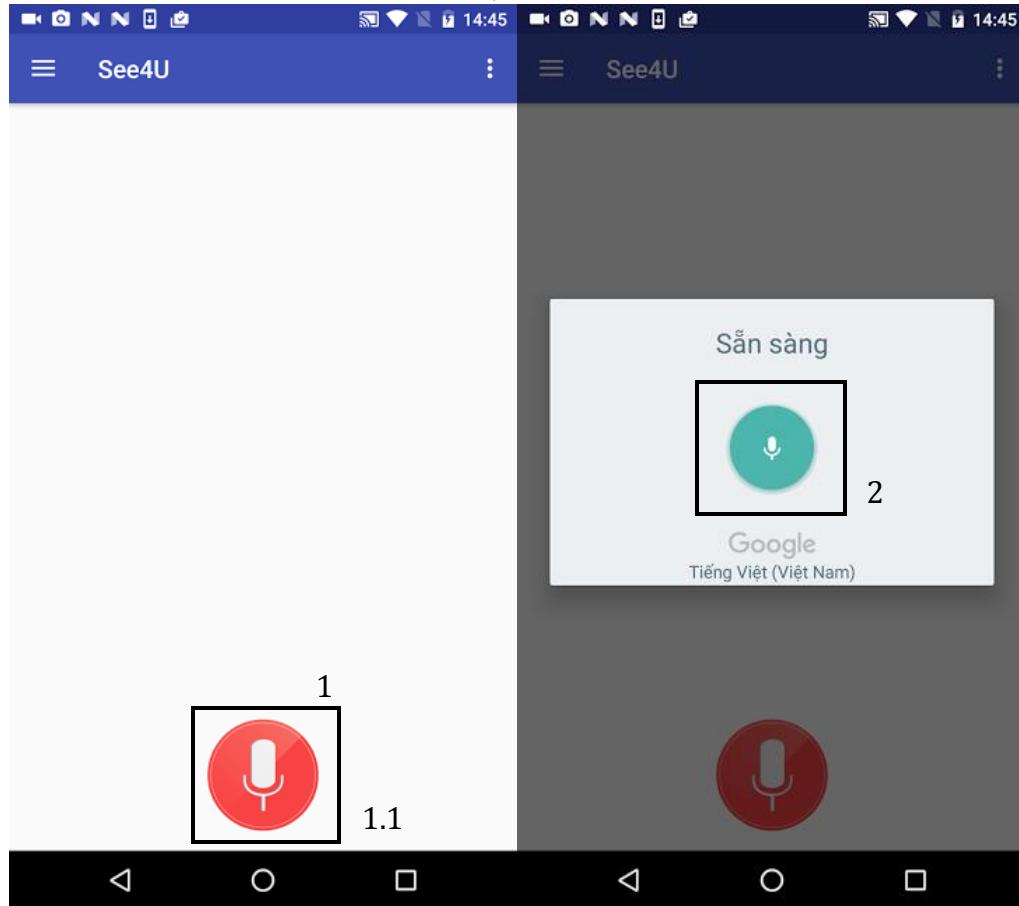
Figure 100: Mobile Application identify result of unknown person

- Identify Result of a known Person



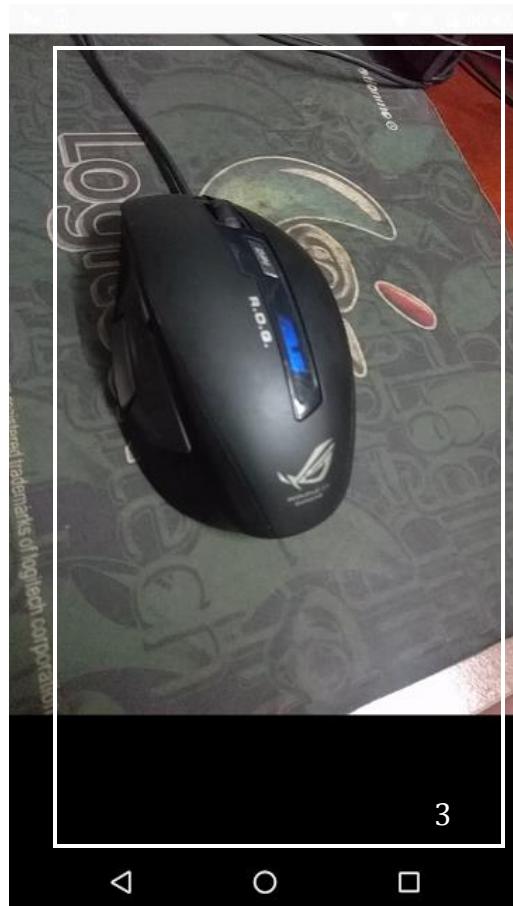
*Figure 101: Mobile Application identify result a known person*

### 6.2.1.2.2 <Authorized User> Detect Object



*Figure 102: Mobile Application Main Screen*

Step	Description
1	Touch the Voice Command button to begin speaking command
1.1	Shake device 2 times to begin speaking command
2	Speak Detect Object Command (edit command in “Thiết Lập” menu)



*Figure 103: Mobile Application camera screen*

Step	Description
3	Touch device screen to capture image

- Identify Result of an Unknown Object

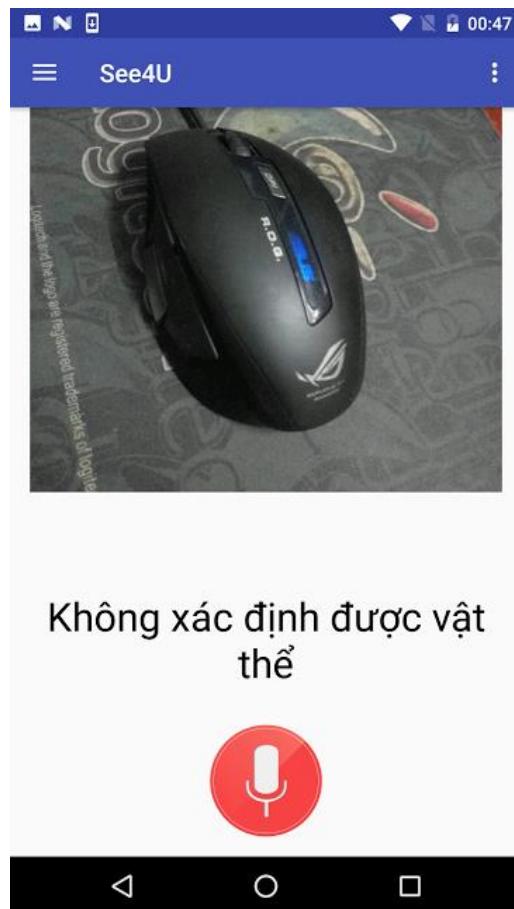


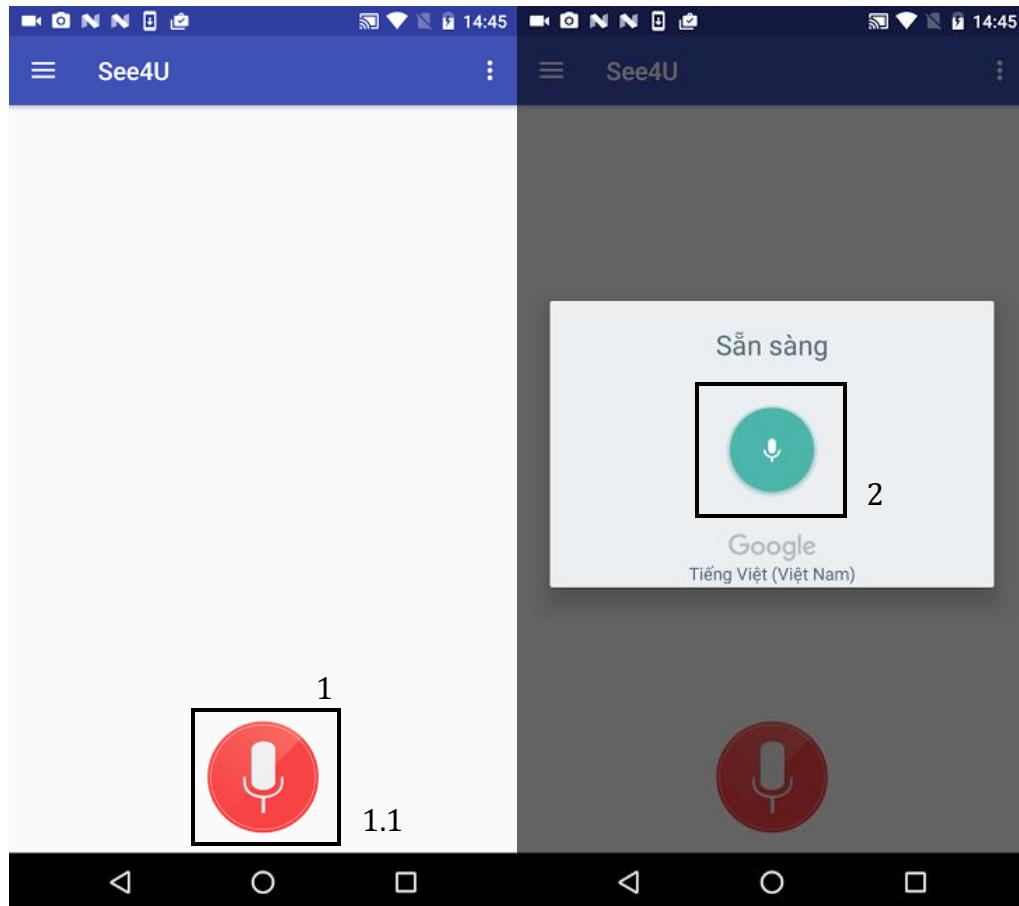
Figure 104: Mobile Application identify result of an unknown object

- Identify Result of a Known Object



*Figure 105: Mobile Application identify result of a known object*

### 6.2.1.2.3 <Authorized User> Detect View



*Figure 106: Mobile Application Main Screen*

Step	Description
1	Touch the Voice Command button to begin speaking command
1.1	Shake device 2 times to begin speaking command
2	Speak Detect View Command (edit command in “Thiết Lập” menu)



*Figure 107: Mobile Application camera screen*

Step	Description
3	Touch device screen to capture image

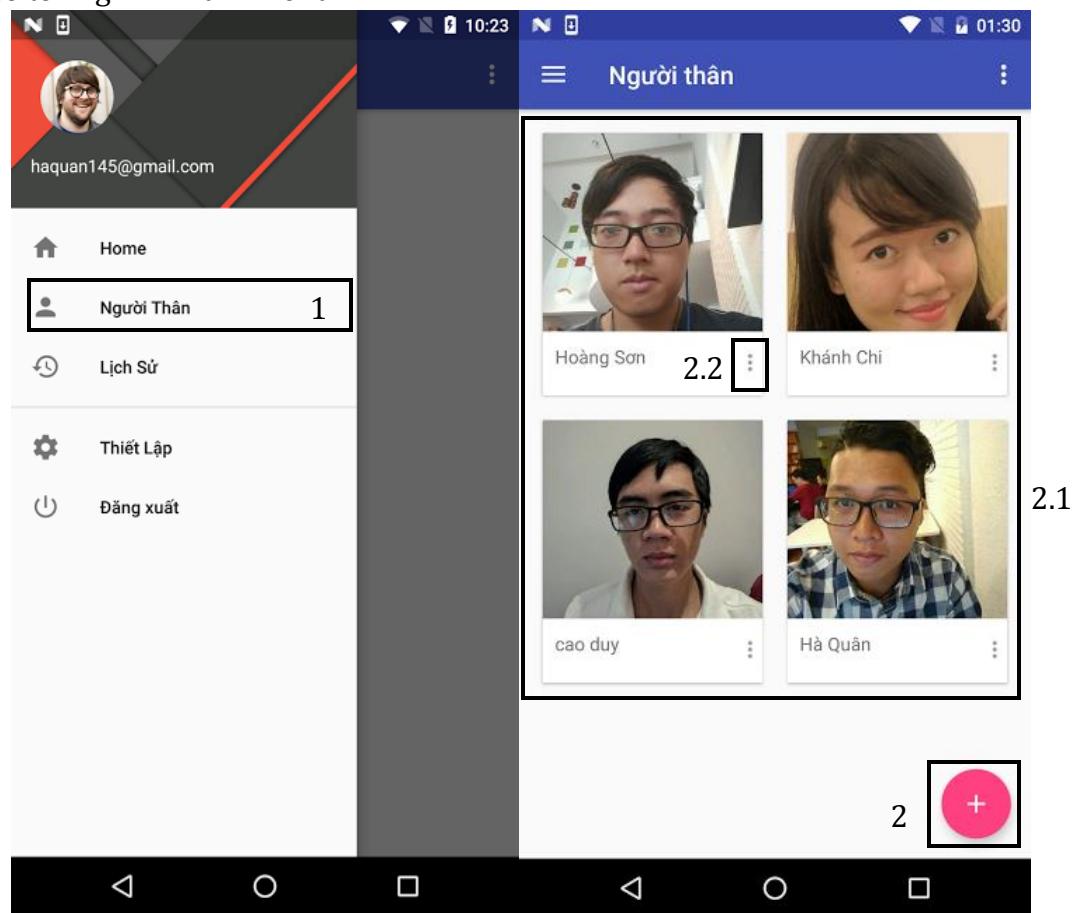
- Detect View Result



*Figure 108: Mobile Application detect view result*

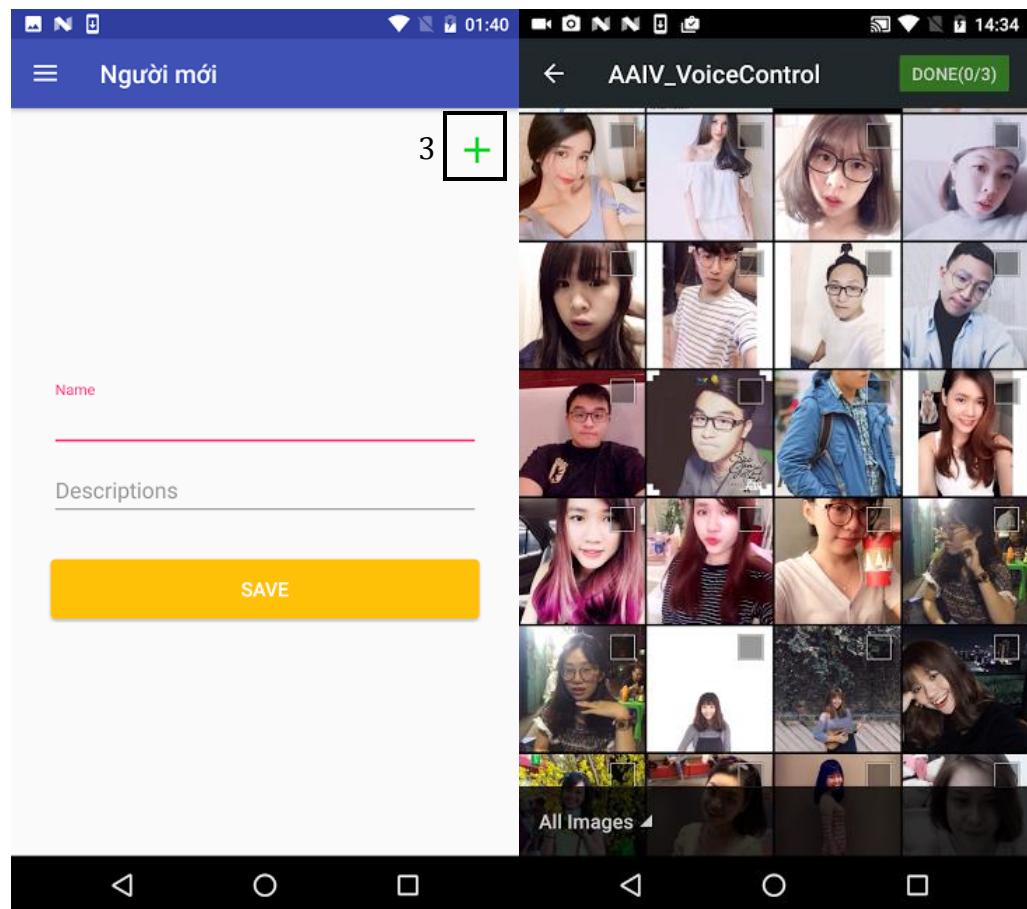
#### 6.2.1.2.4 <Authorized User> Create New Person

- Go to “Người Thân” Menu



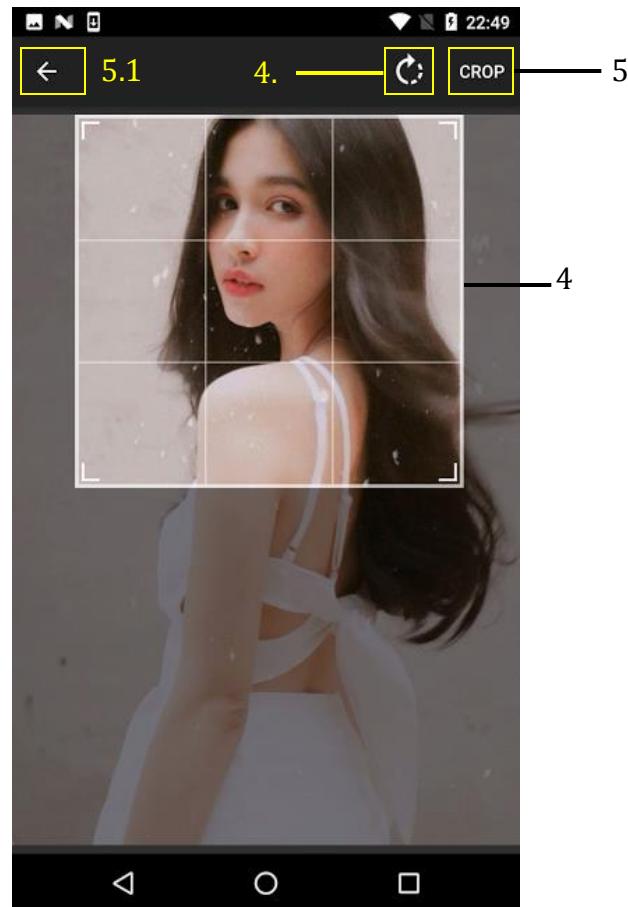
*Figure 109: Mobile Application Person List Screen*

Step	Description
1	Touch “Người Thân” Menu to view Person List
2	Touch “+” button to create a new Person
2.1	Touch a Person card to view and update Person information
2.2	Touch icon to show menu



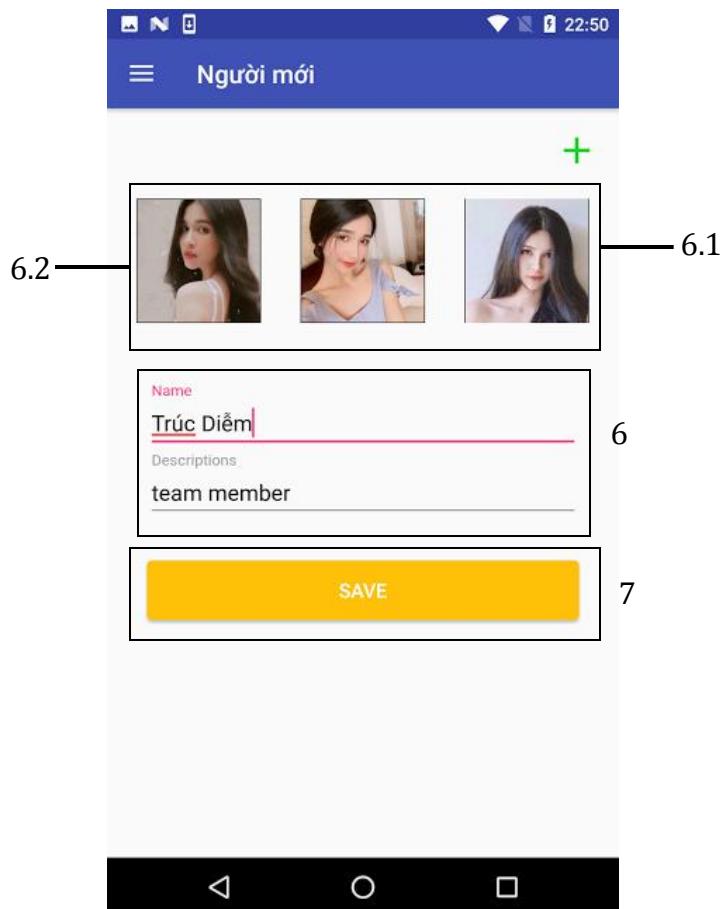
*Figure 110: Mobile Application Create New Person screen*

Step	Description
3	Touch “+” sign to add new Image of person



*Figure 111: Mobile Application Crop Image Screen*

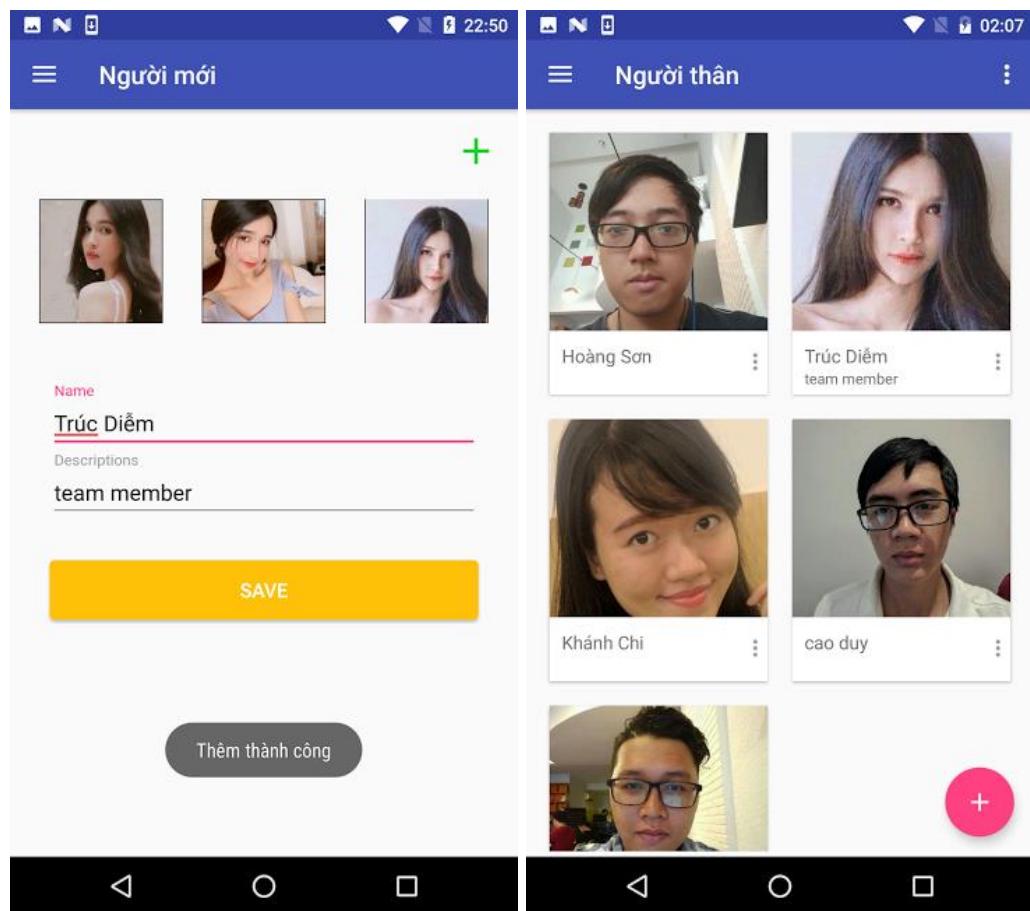
Step	Description
4	Adjust the crop box to fit the desire Person face. If there are more than one person in the picture, crop out only the person which you want to create.
4.1	Rotate the photo
5	Crop the image and proceed to next step
5.1	Back to the previous step



*Figure 112: Mobile Application Create New Person Screen*

Step	Description
6	Fill Person Name and Description
6.1	Touch image to select another image
6.2	Long touch the image to crop the image
7	Touch "Save" button to Create new Person

- Create new Person result



*Figure 113: Mobile Application Create New Person Successfully*

## 6.2.2 Web Application Guide

### 6.2.2.1 Unauthorized User

#### 6.2.2.1.1 <Unauthorized User> Login

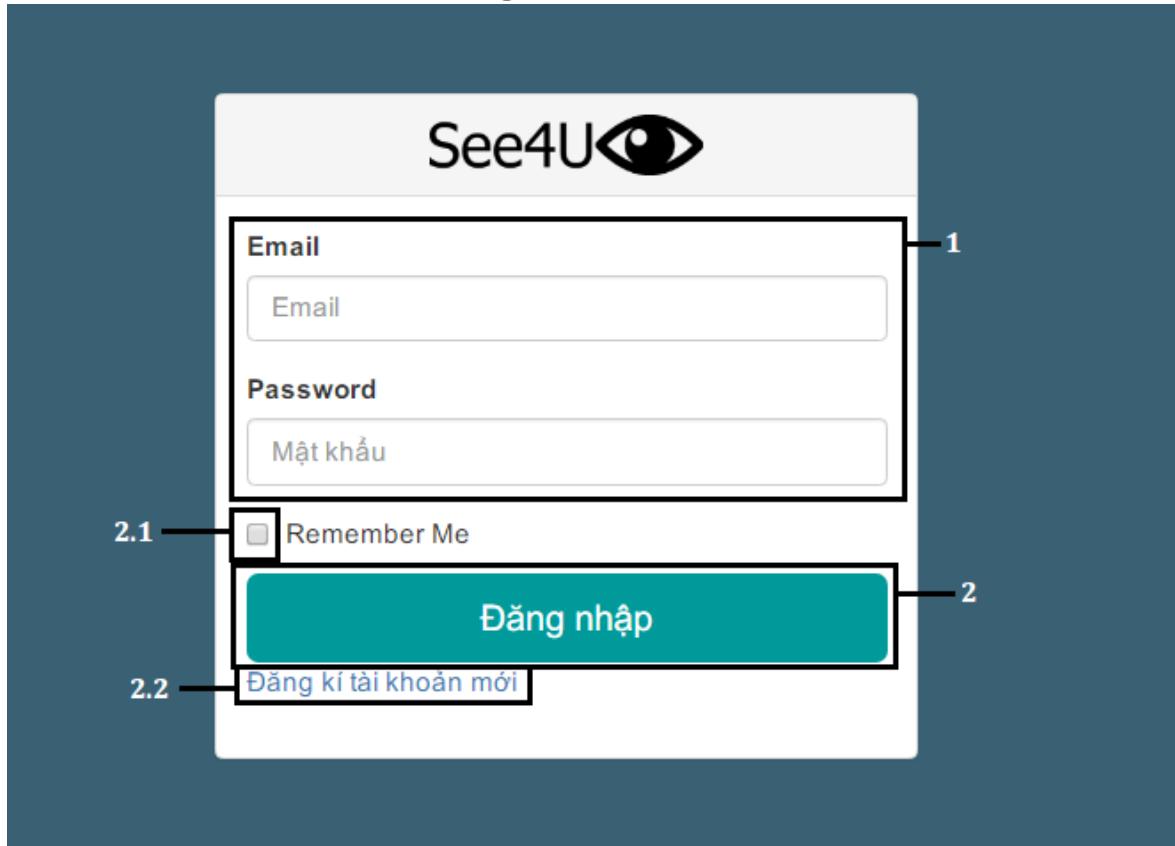


Figure 114: Web Application Login page

Step	Description
1	Fill Email and Password
2	Click “Đăng nhập” button to Login to the website
2.1	Check “Remember Me” checkbox to automatically login next time
2.2	Click “Đăng ký tài khoản mới” to redirect to Register Page

### 6.2.2.1.2 <Unauthorized User> Register

The screenshot shows the registration page for the See4U application. The page features a dark blue header with the 'See4U' logo and a large eye icon. Below the header is a white registration form. The form contains three input fields: 'Email' (placeholder 'Email'), 'Password' (placeholder 'Mật khẩu'), and 'Confirm Password' (placeholder 'Xác nhận'). To the right of the 'Email' field is a small number '1'. Below the form is a teal button labeled 'Đăng ký' (Register) with a small number '2' to its right. At the bottom left of the page is a link 'Đã có tài khoản ? Đăng nhập' (Already have an account? Log in). A small number '2.1' is positioned next to this link.

Figure 115: Web Application Register page

Step	Description
1	Fill Email and Password and Confirm Password
2	Click “Đăng ký” button to create new account.
2.1	Check “Đã có tài khoản ? Đăng nhập” to redirect to Login Page

### 6.2.2.2 Authorized User

#### 6.2.2.2.1 <Authorized User> Admin

##### 6.2.2.2.1.1 <Authorized User><Admin> Create New Object

The screenshot shows a web application interface for managing objects. At the top, there's a header with the logo 'See4U' and a user account section showing 'admin@gmail.com'. Below the header is a navigation menu with items like 'Quản lý tài khoản' and 'Quản lý Concept'. The main content area is titled 'Danh sách đồ vật' (List of objects). It displays a table with columns: No., Image, Tên (Name), and Chú thích (Description). There are two entries: '1' with an image of a black laptop bag, labeled 'balo' and 'ba lô puma đen'; and '2' with an image of a black computer mouse, labeled 'Chuột máy tính' and 'Chuột máy tính màu đen'. At the bottom of the table, it says 'Showing 1 to 2 of 2 entries'. On the right side of the table, there are edit and delete icons. Above the table, there are buttons for 'Tạo mới đồ vật' (Create new object) and 'Kiểm tra đồ vật' (Check object). The 'Tạo mới đồ vật' button is highlighted with a red box and labeled '1'. The 'Nhận dạng đồ vật' button is also visible.

Figure 116: Web Application Create New Object Page

Step	Description
1	Click “Quản lý Concept” menu
2	Click “Tạo mới đồ vật” button

The screenshot shows a 'Create new object' form. The title is 'Tạo mới đồ vật'. The form contains two main sections: 'Tên Đồ Vật' (Object Name) with the value 'Giày' and 'Mô Tả Đồ Vật' (Object Description) with the value 'Giày Nike Trắng'. To the right, there is a preview area labeled 'Hình Ảnh' which shows a white Nike shoe. Below the form, there are two buttons: 'Quay Lại' (Back) and 'Tạo Mới' (Create New). The 'Tạo Mới' button is highlighted with a red box and labeled '3'. To the right of the preview area, there is a button with a plus sign '+' inside a circle, labeled '2'.

Figure 117: Web Application Create New Concept form

Step	Description
1	Fill in Concept information: - Tên Đồ Vật: Object Name - Mô tả đồ vật: Object Description
2	Click “+” button to add new picture of object
3	Click “Tạo mới” button to proceed to next step

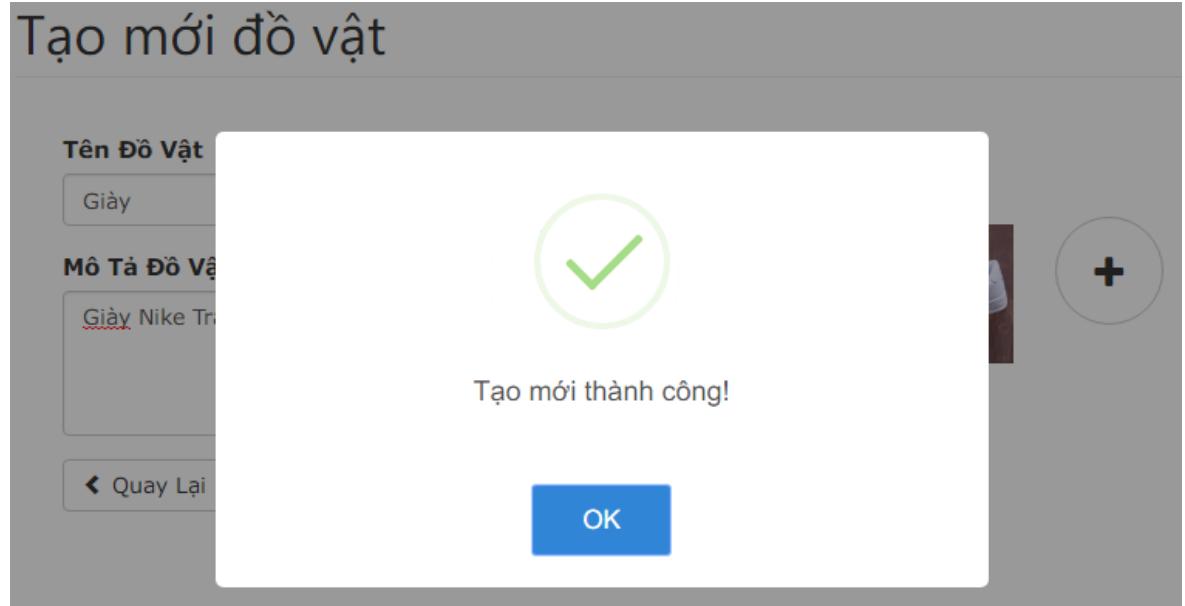
3.1	Click “Quay lại” to back to the previous page
-----	---



*Figure 118: Web Application Create New Concept confirmation*

Step	Description
4	Click “Có” button to create new Object
4.1	Click “Không” button to back to the previous step

- Create Object Result



### 6.2.2.2.1.2 <Authorized User ><Admin> Manage Accounts

*Figure 119: Web Application Manage Accounts*

Step	Description
1	Click “Quản lý tài khoản” menu
2	Click checkbox button to Activate / Deactivate an Account
3	Click “Cập Nhật” button to update accounts state.

- Deactivated account:

### 6.2.2.2.2 <Authorized User> User

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

Figure 120: Web Application Create New Person Page

Step	Description
1	Click “Quản lý người thân” menu
2	Click “Thêm Mới” button

### Thêm người mới

Figure 121: Web Application Create New Person form

Step	Description
3	Fill in Person information: - Tên: Input person name - Thông tin: input person description
4	Click “+” button to add new Person faces
5	Click “Thêm mới” button to proceed to create new person

- Create new person result:

## Thêm người mới

Tên: Trung Thành

Thông tin: Team Leader

Hình ảnh:

Thêm mới

Complete !

*Figure 122: Web Application Create New Person Successfully*

### 6.2.2.2.2 <Authorized User><User> Add New Person from Logs

See4U

Quản lý người thân

Người chưa nhận diện được

Người chưa nhận diện được

1

2

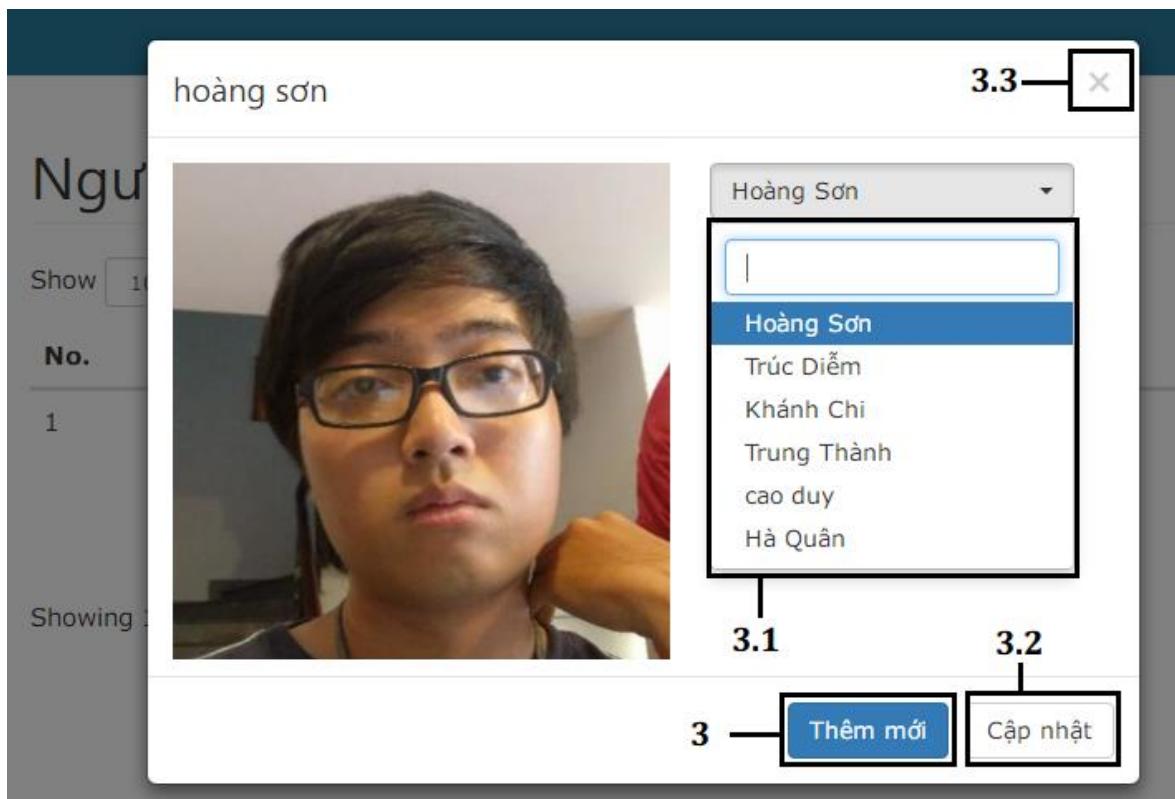
+

No.	Hình ảnh	Tên	Ngày tạo
1		hoàng sơn	03/31/2017 10:37:44 AM

Showing 1 to 1 of 1 entries

*Figure 123: Web Application Create New Person form Log page*

Step	Description
1	Click “Người chưa nhận diện được” menu
2	Click “+” button to proceed to next step



*Figure 124: Web Application Create new Person from log pop-up*

Step	Description
3	Click “Thêm mới” button to proceed to next step
3.1	Choose existed Person to update
3.2	Click “Cập nhật” to add new image to chosen existed person in step 3.1
3.3	Click “X” button to close pop-up panel, back to the previous step

The screenshot shows a form for creating a new person. The title is 'Thêm người mới'. The form has two main sections: 'Tên' (Name) and 'Thông tin' (Information). The 'Tên' section contains a text input field with the value 'hoàng sơn'. The 'Thông tin' section contains a text input field with the value 'Thông tin'. To the right of the form is a 'Hình ảnh' (Image) section. It shows a preview image of a person with glasses and a plus sign button for adding a new image. At the bottom left is a 'Thêm mới' button.

*Figure 125: Web Application Create new Person from log form*

Step	Description
4	Fill in new person information: - Tên: input person name. Default: person name from Log - Thông tin: input person description
5	Click “+” button to choose person image. Default: person picture from Log
6	Click “Thêm mới” to create new person from Log

- Create new Person from Log result:

## Thêm người mới

Tên

Hình ảnh

Thông tin

Thêm mới

Complete!

Figure 126: Web Application Create new Person from log successfully

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

See4U

Quản lý người thân 1

Danh sách

Người chưa nhận diện được

Kiểm tra trùng lặp

Thêm mới

Show 10 entries

Search:

No	Ảnh đại diện	Tên	Thông tin
1		Hoàng Sơn	

Figure 127: Web Application Update a Person Information and Face

Step	Description
1	Click “Quản lý người thân” menu
2	Click “Edit” button to proceed to next step

## Cập nhật thông tin

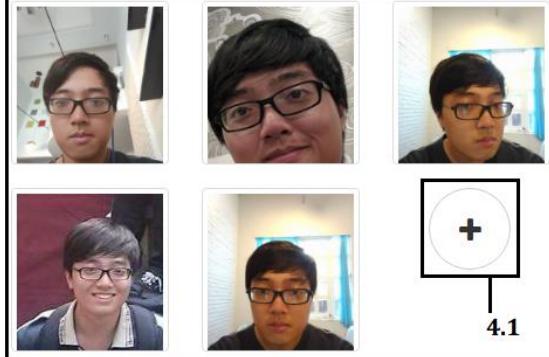
3

Tên  
Hoàng Sơn

Thông tin  
Team mate #2

[Quay lại](#) [Cập nhật thông tin](#)

4



**Figure 128: Web Application Update Person Information form**

Step	Description
3	Update person information
4	Click an image to update person's face
4.1	Click “+” button to add new image of person's face

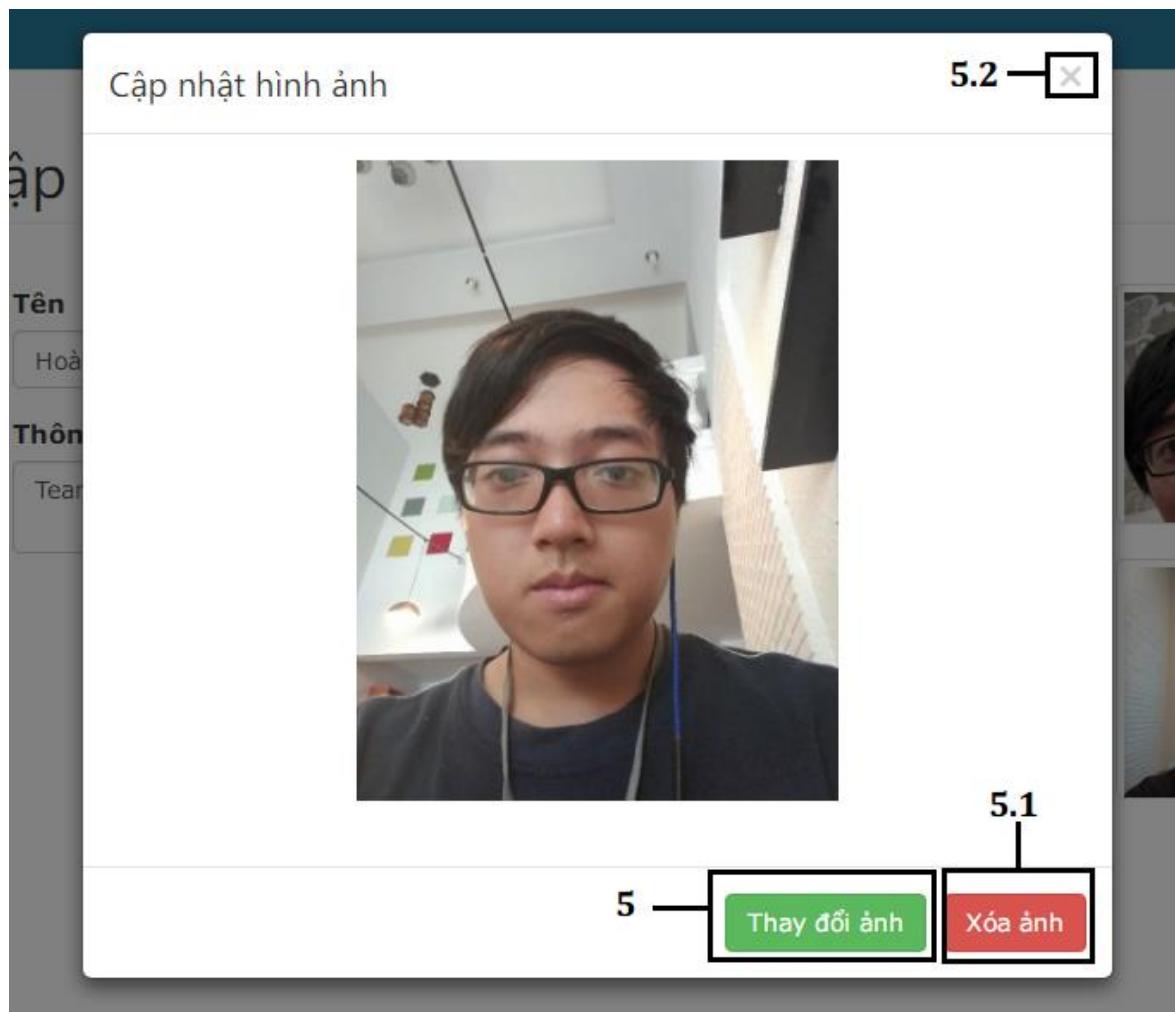
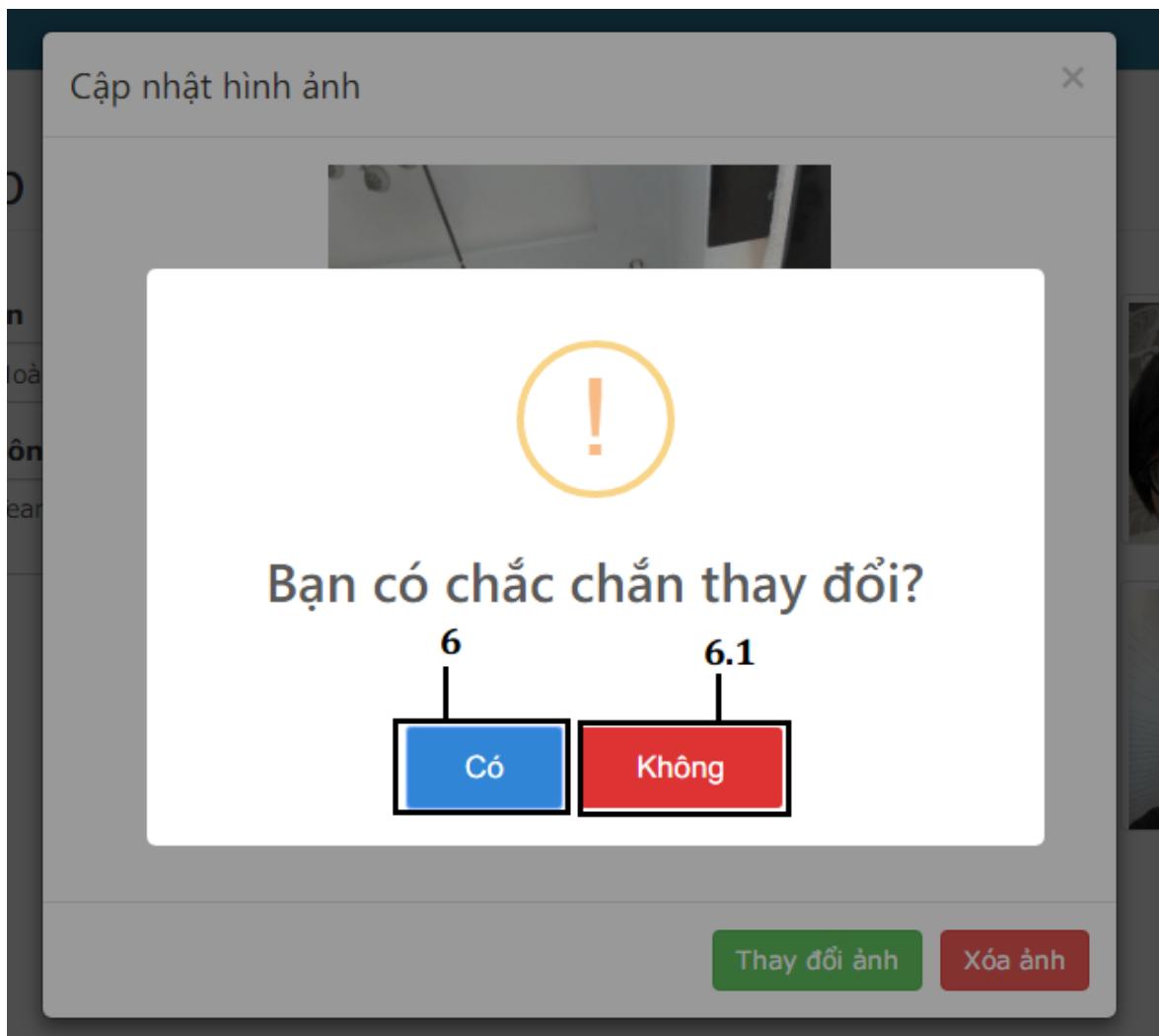


Figure 129: Web Application Update Person Face Pop-up

Step	Description
5	Click “Thay đổi ảnh” button to choose another image for this image.
5.1	Click “Xóa ảnh” button to delete the chosen image.
5.2	Click “x” button to close Update person face pop-up.



*Figure 130: Web Application Update Person Face confirmation*

Step	Description
6	Click “Có” button to change to update the chosen image.
6.1	Click “Không” button to back to the previous step

### Cập nhật thông tin

Tên  
Hoàng Sơn

Thông tin  
team member #2

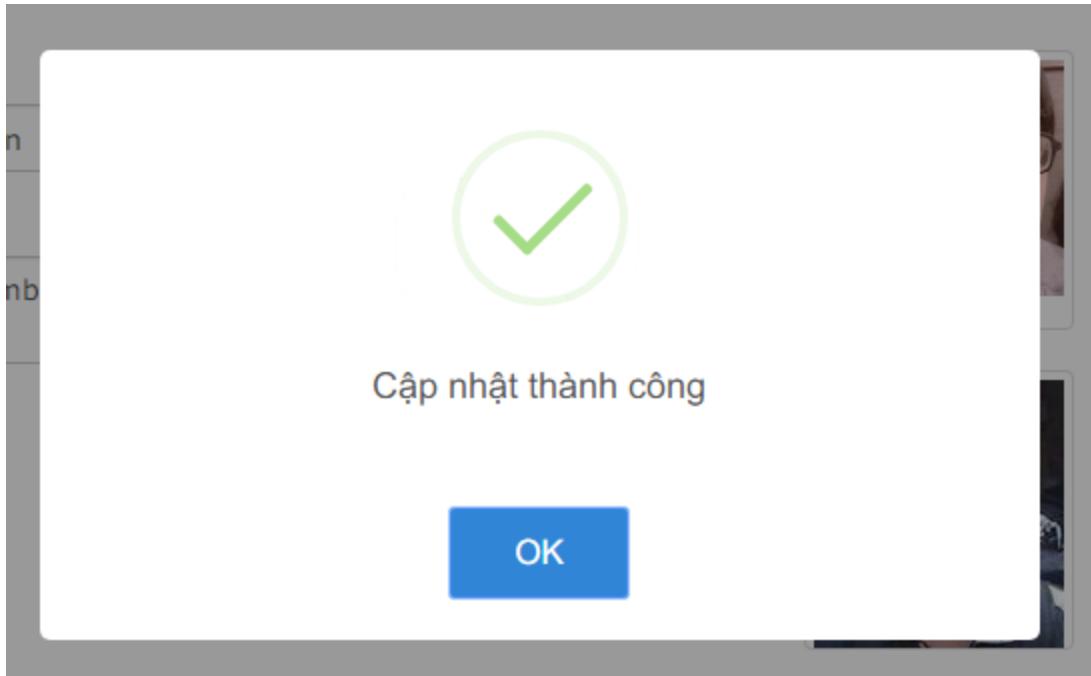
7

Quay lại Cập nhật thông tin

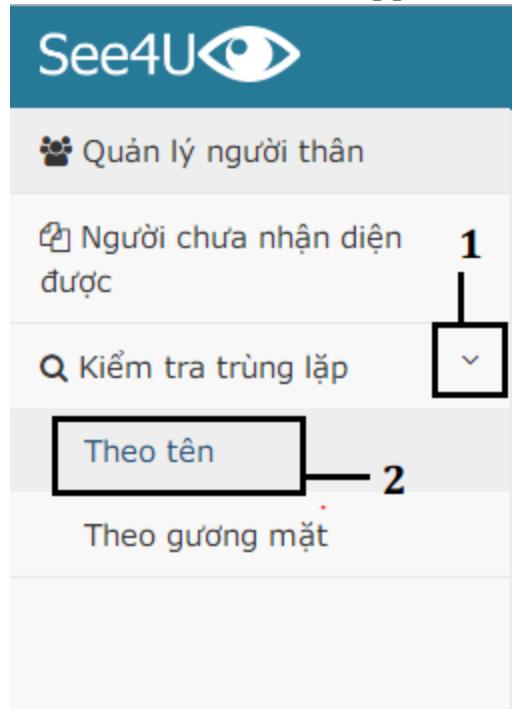
*Figure 131: Web Application Update Person*

Step	Description
7	Click “Cập nhật thông tin” button to update person information.

- Update person information result:

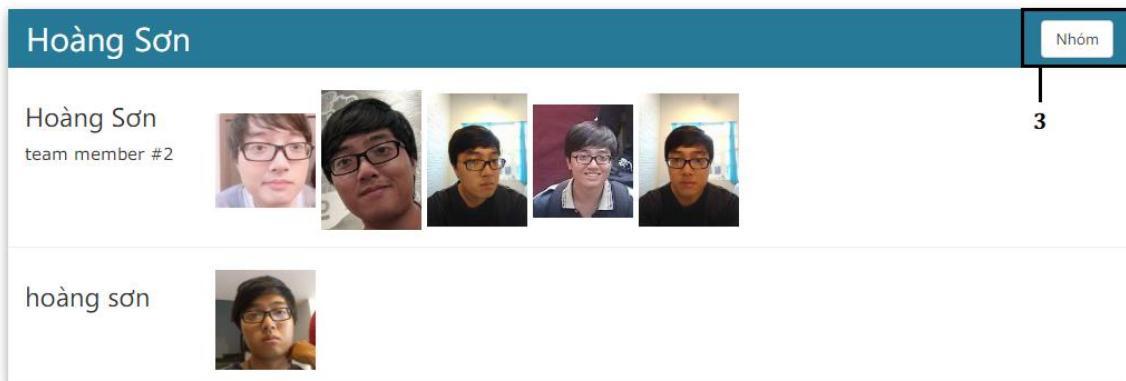


#### 6.2.2.2.2.4 <Authorized User><User> Check Duplicated People by Name



Step	Description
1	Click “Kiểm tra trùng lặp” to show dropdown menu
2	Click “Theo tên” menu to proceed to next step

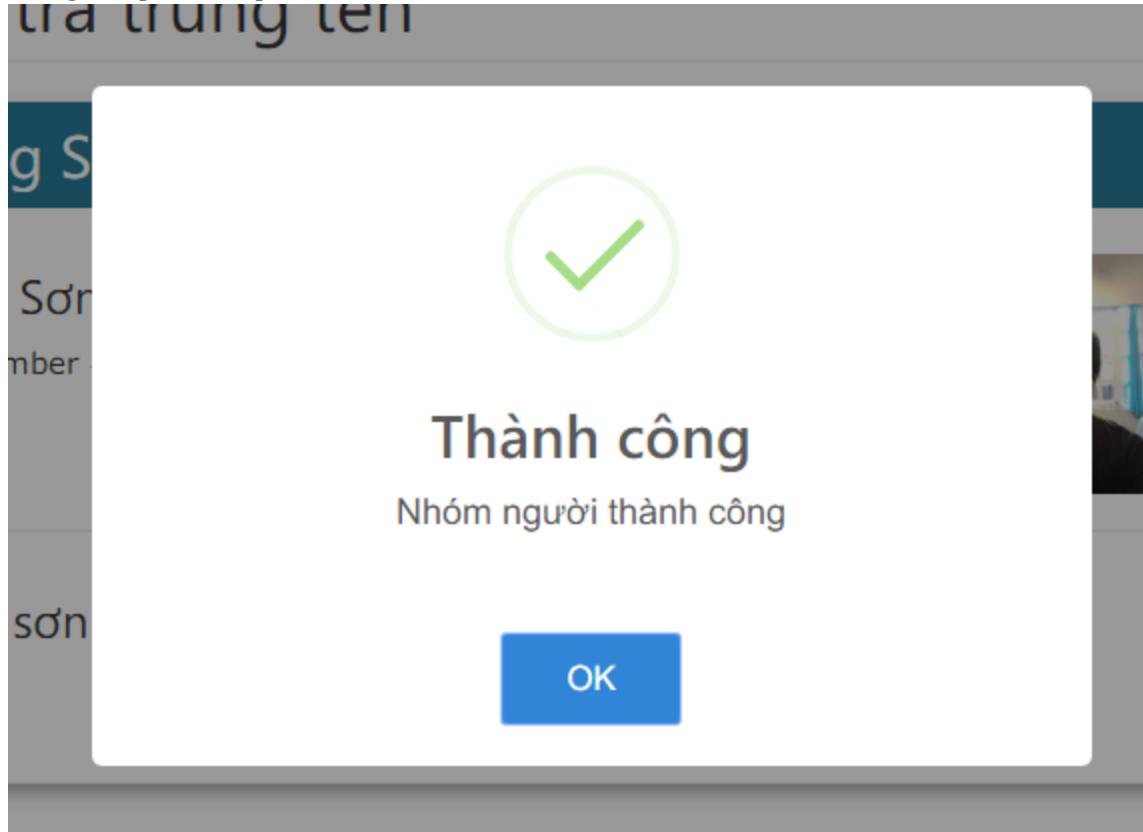
#### Kiểm tra trùng tên



*Figure 132: Web Application Check Duplication Person by Name*

Step	Description
3	Click “Nhóm” button to merge all duplicated person

- Merge duplicated person result:



#### 6.2.2.2.5 <Authorized User><User> Check Duplicated People by Faces

Quản lý người thân

Người chưa nhận diện được

Kiểm tra trùng lặp

Theo tên

Theo gương mặt

Danh sách

1 Show 10 entries

No	Ảnh đại diện	Tên	Thông tin
1		Hoàng Sơn	team member #2
2		Sơn Nguyễn	team member

Figure 133: Web Application Check Duplication Person by Faces

Step	Description
1	Click “Kiểm tra trùng lặp” menu
2	Click “Theo gương mặt” menu to proceed to next step

## Kiểm tra trùng gương mặt

The screenshot shows a user interface for merging people. At the top, there's a blue header bar with the number '#1'. Below it, two sections are listed: 'Hoàng Sơn' (team member #2) and 'Sơn Nguyễn' (team member). Each section contains a name, a title ('team member'), and a grid of small profile pictures. In the top right corner of the 'Nhóm' button, there is a small callout box labeled '3' with an arrow pointing to it. The entire interface is contained within a white box.

*Figure 134: Web Application Check Duplication Person by Faces*

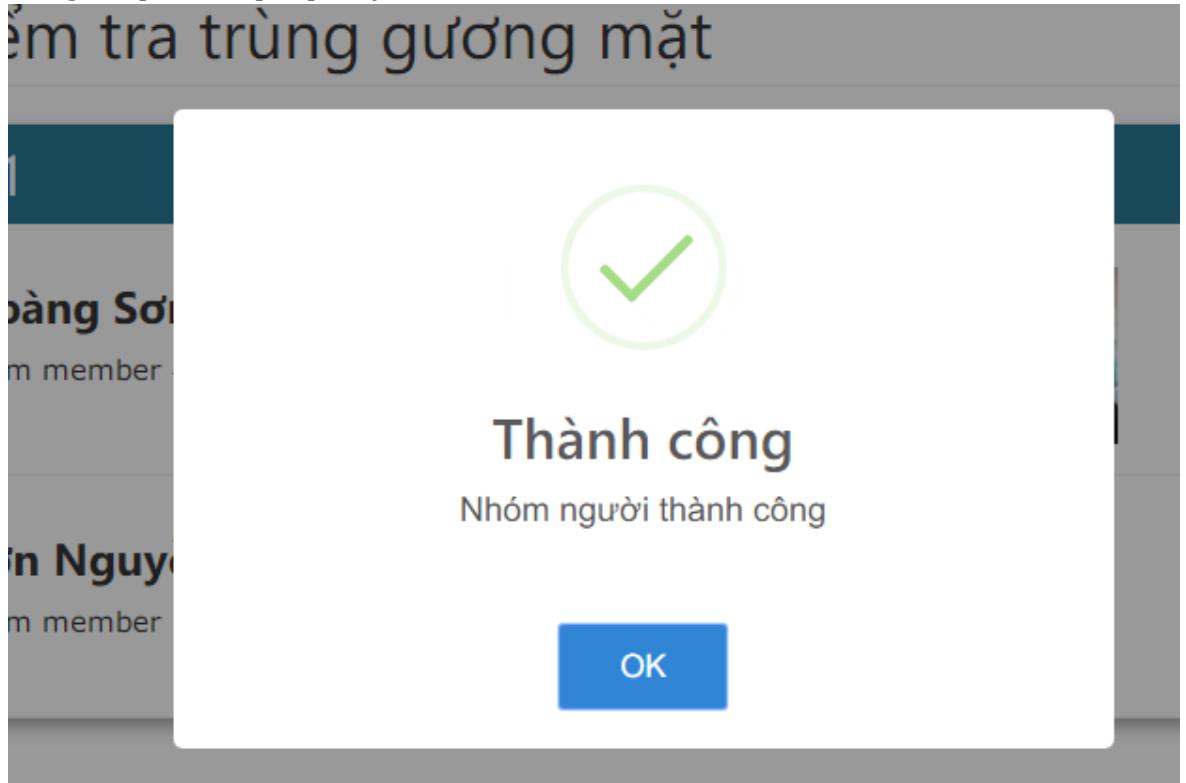
Step	Description
3	Click “Nhóm” to proceed to next step

This screenshot shows a modal dialog box titled 'Nhập tên' (Enter name). Inside the dialog, there is a text input field containing a single vertical bar character '|'. Below the input field are two buttons: 'OK' (blue) and 'Cancel' (gray). An arrow labeled '4' points to the 'Nhập tên' title. An arrow labeled '5' points to the 'OK' button. An arrow labeled '5.1' points to the 'Cancel' button. The background of the dialog is white, while the main application interface is gray.

*Figure 135: Web Application Check Duplication Person by Faces – Choose New Person Name*

Step	Description
4	Fill new person name when merging
5	Click “Ok” button to merge duplicated people with new name
5.1	Click “Cancel” button to back to the previous step

- Merge duplicated people by faces result:



## 7 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>