

---

# FACE AGE RECOGNITION SPECIFICATION

for

Final Project

Version 1.0 approved

Prepared by Group7

June 18, 2019

# Contents

<b>1</b>	<b>Introduction</b>	<b>3</b>
1.1	Purpose . . . . .	3
1.2	Intended Audience and Reading Suggestions . . . . .	3
1.3	Project Scope . . . . .	3
<b>2</b>	<b>Overall Description</b>	<b>4</b>
2.1	Product Perspective . . . . .	4
2.2	Product Functions . . . . .	4
2.3	User Classes and Characteristics . . . . .	4
2.4	Operating Environment . . . . .	4
2.5	Design and Implementation Constraints . . . . .	5
2.6	Assumptions and Dependencies . . . . .	5
<b>3</b>	<b>External Interface Requirements</b>	<b>6</b>
3.1	User Interfaces . . . . .	6
3.2	Hardware Interfaces . . . . .	6
3.3	Software Interfaces . . . . .	6
<b>4</b>	<b>System Features</b>	<b>7</b>
4.1	Description and Priority . . . . .	7
4.2	Stimulus/Response Sequences . . . . .	7
4.3	Functional Requirements . . . . .	7
<b>5</b>	<b>Other Nonfunctional Requirements</b>	<b>8</b>
5.1	Performance Requirements . . . . .	8
5.2	Safety Requirements . . . . .	8
5.3	Security Requirements . . . . .	8

# 1 Introduction

## 1.1 Purpose

專案目標:

藉由UI介面選擇一張人物大頭照圖，經過系統判別該照片中人物年齡約是多少。

系統介面:

- 能讓使用者操作選擇圖片的UI介面
- 接收圖片並分析人物年齡的後端程式

## 1.2 Intended Audience and Reading Suggestions

此系統為人臉年齡辨識，本規格書提供專案開發人員做為參考，包括專案概述、功能說明、UI操作及環境架設。

## 1.3 Project Scope

此系統包含了能選擇圖片的UI介面以及後端處理分析人物年紀的程式，再傳回結果至UI顯示。

## 2 Overall Description

### 2.1 Product Perspective

本系統分為兩個部分，分別為UI前端，和人臉圖片辨識系統，如下圖所示。

### 2.2 Product Functions

### 2.3 User Classes and Characteristics

<Identify the various user classes that you anticipate will use this product. User classes may be differentiated based on frequency of use, subset of product functions used, technical expertise, security or privilege levels, educational level, or experience. Describe the pertinent characteristics of each user class. Certain requirements may pertain only to certain user classes. Distinguish the most important user classes for this product from those who are less important to satisfy.>

### 2.4 Operating Environment

<Describe the environment in which the software will operate, including the hardware platform, operating system and versions, and any other software components or applications with which it must peacefully coexist.>

### 2.5 Design and Implementation Constraints

<Describe any items or issues that will limit the options available to the developers. These might include: corporate or regulatory policies; hardware limitations (timing requirements, memory requirements); interfaces to other applications; specific technologies, tools, and databases to be used; parallel operations; language requirements; communications protocols; security considerations; design conventions or programming standards (for example, if the customer's organization will be responsible for maintaining the delivered software).>

### 2.6 Assumptions and Dependencies

<List any assumed factors (as opposed to known facts) that could affect the requirements stated in the SRS. These could include third-party or commercial components that you

plan to use, issues around the development or operating environment, or constraints. The project could be affected if these assumptions are incorrect, are not shared, or change. Also identify any dependencies the project has on external factors, such as software components that you intend to reuse from another project, unless they are already documented elsewhere (for example, in the vision and scope document or the project plan).>

## **3 External Interface Requirements**

### **3.1 User Interfaces**

<Describe the logical characteristics of each interface between the software product and the users. This may include sample screen images, any GUI standards or product family style guides that are to be followed, screen layout constraints, standard buttons and functions (e.g., help) that will appear on every screen, keyboard shortcuts, error message display standards, and so on. Define the software components for which a user interface is needed. Details of the user interface design should be documented in a separate user interface specification.>

### **3.2 Hardware Interfaces**

<Describe the logical and physical characteristics of each interface between the software product and the hardware components of the system. This may include the supported device types, the nature of the data and control interactions between the software and the hardware, and communication protocols to be used.>

### **3.3 Software Interfaces**

<Describe the connections between this product and other specific software components (name and version), including databases, operating systems, tools, libraries, and integrated commercial components. Identify the data items or messages coming into the system and going out and describe the purpose of each. Describe the services needed and the nature of communications. Refer to documents that describe detailed application programming interface protocols. Identify data that will be shared across software components. If the data sharing mechanism must be implemented in a specific way (for example, use of a global data area in a multitasking operating system), specify this as an implementation constraint.>

## 4 System Features

<This template illustrates organizing the functional requirements for the product by system features, the major services provided by the product. You may prefer to organize this section by use case, mode of operation, user class, object class, functional hierarchy, or combinations of these, whatever makes the most logical sense for your product.>

### 4.1 Description and Priority

<Provide a short description of the feature and indicate whether it is of High, Medium, or Low priority. You could also include specific priority component ratings, such as benefit, penalty, cost, and risk (each rated on a relative scale from a low of 1 to a high of 9).>

### 4.2 Stimulus/Response Sequences

<List the sequences of user actions and system responses that stimulate the behavior defined for this feature. These will correspond to the dialog elements associated with use cases.>

### 4.3 Functional Requirements

<Itemize the detailed functional requirements associated with this feature. These are the software capabilities that must be present in order for the user to carry out the services provided by the feature, or to execute the use case. Include how the product should respond to anticipated error conditions or invalid inputs. Requirements should be concise, complete, unambiguous, verifiable, and necessary. Use “TBD” as a placeholder to indicate when necessary information is not yet available.>

<Each requirement should be uniquely identified with a sequence number or a meaningful tag of some kind.>

REQ-1: REQ-2:

## 5 Other Nonfunctional Requirements

### 5.1 Performance Requirements

<If there are performance requirements for the product under various circumstances, state them here and explain their rationale, to help the developers understand the intent and make suitable design choices. Specify the timing relationships for real time systems. Make such requirements as specific as possible. You may need to state performance requirements for individual functional requirements or features.>

### 5.2 Safety Requirements

<Specify those requirements that are concerned with possible loss, damage, or harm that could result from the use of the product. Define any safeguards or actions that must be taken, as well as actions that must be prevented. Refer to any external policies or regulations that state safety issues that affect the product's design or use. Define any safety certifications that must be satisfied.>

### 5.3 Security Requirements

<Specify any requirements regarding security or privacy issues surrounding use of the product or protection of the data used or created by the product. Define any user identity authentication requirements. Refer to any external policies or regulations containing security issues that affect the product. Define any security or privacy certifications that must be satisfied.>