**Beauty and the Beast**

Design Specification



|  |  |
| --- | --- |
| 학번 | 이름 |
| 2014311323 | 박범진 |
| 2017314888 | 장유진 |
| 2015311099 | 조준희 |
| 2018313592 | 공허 |
| 2016313354 | 위응주 |
| 2019319598 | 류젠웨 |

# Contents

[Contents 2](#_Toc24578732)

[1. Preface 5](#_Toc24578733)

[1. Objective 5](#_Toc24578734)

[2. Readership 5](#_Toc24578735)

[3. Document Structure 5](#_Toc24578736)

[A. Preface 5](#_Toc24578737)

[B. Introduction 5](#_Toc24578738)

[C. Glossary 6](#_Toc24578739)

[D. System Architecture 6](#_Toc24578740)

[E. Similar Celebrities Finding System 6](#_Toc24578741)

[F. Data Collecting System 6](#_Toc24578742)

[G. Makeup Applying System 7](#_Toc24578743)

[H. Testing Plan 7](#_Toc24578744)

[I. Development Plan 7](#_Toc24578745)

[J. Index 7](#_Toc24578746)

[4. Version History 8](#_Toc24578747)

[2. Introduction 9](#_Toc24578748)

[1. Objective 9](#_Toc24578749)

[2. Applied Diagrams 9](#_Toc24578750)

[A. UML 9](#_Toc24578751)

[B. Package Diagram 10](#_Toc24578752)

[C. Deployment Diagram 10](#_Toc24578753)

[D. Class Diagram 10](#_Toc24578754)

[E. State Diagram 11](#_Toc24578755)

[F. Sequence Diagram 11](#_Toc24578756)

[3. Applied Tools 12](#_Toc24578757)

[4. Project Scope 12](#_Toc24578758)

[3. Glossary 13](#_Toc24578759)

[1. Objective 13](#_Toc24578760)

[2. Term Definition 13](#_Toc24578761)

[4. System Architecture 15](#_Toc24578762)

[1. Objective 15](#_Toc24578763)

[2. System Organization 15](#_Toc24578764)

[A. Frontend Application 16](#_Toc24578765)

[B. Backend Application 16](#_Toc24578766)

[C. Similar Celebrities Finding System 17](#_Toc24578767)

[D. Data Collecting System 17](#_Toc24578768)

[E. Makeup Applying System 18](#_Toc24578769)

[3. Package Diagram 19](#_Toc24578770)

[4. Deployment Diagram 19](#_Toc24578771)

[5. Similar Celebrities Finding System 20](#_Toc24578772)

[1. Objective 20](#_Toc24578773)

[2. Class Diagram 20](#_Toc24578774)

[3. Sequence Diagram 21](#_Toc24578775)

[4. State Diagram 21](#_Toc24578776)

[6. Data Collecting System 22](#_Toc24578777)

[1. Objective 22](#_Toc24578778)

[2. Class Diagram 22](#_Toc24578779)

[3. Sequence Diagram 22](#_Toc24578780)

[4. State Diagram 23](#_Toc24578781)

[7. Makeup Applying System 24](#_Toc24578782)

[1. Objective 24](#_Toc24578783)

[2. Class Diagram 24](#_Toc24578784)

[3. Sequence Diagram 25](#_Toc24578785)

[4. State Diagram 25](#_Toc24578786)

[8. Testing Plan 26](#_Toc24578787)

[1. Objective 26](#_Toc24578788)

[2. Testing Policy 26](#_Toc24578789)

[A. Development Testing 26](#_Toc24578790)

[B. Release Testing 28](#_Toc24578791)

[C. User Testing 28](#_Toc24578792)

[3. Test Case 29](#_Toc24578793)

[A. Main System 29](#_Toc24578794)

[B. Similar Celebrities Finding System 32](#_Toc24578795)

[C. Data Collecting System 33](#_Toc24578796)

[D. Makeup Applying System 34](#_Toc24578797)

[9. Development Plan 35](#_Toc24578798)

[1. Objective 35](#_Toc24578799)

[2. Programming Language & IDE 35](#_Toc24578800)

[A. Programming Language 35](#_Toc24578801)

[B. IDE 36](#_Toc24578802)

[3. Schedule 37](#_Toc24578803)

[10. Index 38](#_Toc24578804)

[1. Objective 38](#_Toc24578805)

[2. Table Index 38](#_Toc24578806)

[3. Figure Index 39](#_Toc24578807)

[4. Diagram Index 39](#_Toc24578808)

# 1. Preface

## 1. Objective

본 항목에서는 명세서의 예상 독자를 기술하고 명세서의 항목들을 간략하게 소개한다. 또, 명세서 작성 과정 및 버전 별 변경사항을 기술한다.

## 2. Readership

본 문서는 프로젝트 매니저, 프로젝트 팀 및 개발 팀을 위해 작성되었고, 시스템의 지원, 유지 관리와 관련된 다양한 이해 관계자 및 사용자들을 대상으로 한다. 즉, 본 문서의 독자는 “Beauty and the Beast” 어플리케이션의 개발 및 유지 관리 팀 구성원이다.

## 3. Document Structure

### A. Preface

Preface 항목에서는 명세서의 예상 독자를 기술하고 명세서의 항목들을 간략하게 소개한다. 또, 명세서 작성 과정 및 버전 별 변경사항을 기술한다.

### B. Introduction

Introduction 항목에서는 본 문서에 사용 된 다이어그램과 도구를 기술한다. 또한 본 문서에서 소개하는 “Beauty and the Beast” 어플리케이션의 범위를 기술한다.

### C. Glossary

Glossary 항목에서는 전문적 지식이 없는 독자들이 문서를 이해하는 데에 어려움이 없도록 본 문서에 사용된 기술적 용어들을 정의한다.

### D. System Architecture

System architecture 항목에서는 개발하고자 하는 시스템의 전체적인 설명을 기술한다. 각 시스템의 전체적인 구조는 블록 다이어그램으로 표시하고, 시스템 간의 관계 및 실제 사용 방법은 패키지 다이어그램 및 배포 다이어그램으로 설명한다.

### E. Similar Celebrities Finding System

Similar celebrities finding system 항목에서는 사용자가 제공한 사진과 비슷한 연예인을 찾아 돌려주는 시스템의 구조적인 디자인과 해당 시스템에 사용된 알고리즘을 기술한다. Similar celebrities finding system이 사용하는 기계학습 모델인 CNN 모델은 python과 tensorflow 모듈로 개발된 오픈 소스를 사용하여 조달한다.

### F. Data Collecting System

Data collecting system 항목에서는 Similar celebrities finding system에서 사용할 학습 데이터를 수집하는 시스템의 구조적인 디자인과 데이터 수집 방법을 기술한다. Data collecting system 내부의 이미지 데이터 크롤링 시스템과 이미지 데이터 정렬 시스템은 python으로 개발된 오픈 소스를 사용하여 조달한다.

### G. Makeup Applying System

Makeup applying system 항목에서는 사용자가 제공한 사진에 사용자가 선택한 연예인의 화장법을 적용하는 시스템의 구조적인 디자인과 해당 시스템에 사용된 알고리즘을 기술한다. Makeup applying system은 얼굴 인식 시스템과 화장 적용 시스템을 포함하고 있으며, OpenCV 라이브러리를 이용하여 개발한다.

### H. Testing Plan

Testing plan 항목에서는 테스트 정책 및 테스트 사례를 기술한다. 본 항목의 목적은 시스템 개발을 완료 한 후 시스템의 결함을 식별하고 시스템이 작동 중인지 확인 및 분석하기 위한 예비 계획을 수립하는 것이다. 테스트 정책 장과 테스트 사례 장으로 구분되며, 테스트 정책은 테스트 기준을 설정하고 테스트 사례는 구체적인 입력 및 출력 세트를 기술한다.

### I. Development Plan

Development plan 항목에서는 시스템 개발에 사용할 프로그래밍 언어와 개발 IDE를 기술한다. 또한, 사용할 프로그래밍 언어 및 관리 도구의 버전을 명시한다.

### J. Index

Index항목에서는 본 문서에 사용된 표, 그림, 다이어그램의 색인을 기술한다.

## 4. Version History

|  |  |  |
| --- | --- | --- |
| Version | Modified Date | Description |
| 0.1 | 2019.11.02 | 문서 목차 작성 |
| 0.2 | 2019.11.03 | Preface 및 Introduction작성 |
| 0.3 | 2019.11.09 | System Architecture 작성 |
| 0.4 | 2019.11.10 | 세부 system 작성 |
| 1.0 | 2019.11.11 | Plan 포함 모든 항목 작성 |
| 1.1 | 2019.11.12 | 검토 및 오타 수정 |

[Table 1] Version History

# 2. Introduction

## 1. Objective

본 항목에서는 본 문서에 사용 된 다이어그램과 도구를 기술한다. 또한 본 문서에서 소개하는 “Beauty and the Beast” 어플리케이션의 범위를 기술한다.

## 2. Applied Diagrams

### A. UML

As a unified software modeling language, UML has a wide range of modeling capabilities. UML is the process of digesting, absorbing, and refining all software that exists today

Proposed on the basis of model language, it is a master of software modeling language. UML also breaks through the limitation of software, widely absorbs modeling methods in other fields, and combines the characteristics of software according to the general principles of modeling, so it has a solid theoretical foundation and universality. UML can be used not only for software modeling, but also for modeling work in other areas since it provides different modeling techniques and a handful subset of diagrams. It can be efficiently used to provide means of communication between developers and users as it covers wide range of symbols and definitions and it consists of the following diagrams: Package Diagram, Deployment Diagram, Class Diagram, State Diagram, Sequence Diagram and ER Diagram.

### B. Package Diagram

Package diagrams depict the dependencies between the packages that make up a model. A package is a general-purpose element for organizing model elements and diagrams into groups. Elements can be classes, interfaces, objects, tables or other packages.

### C. Deployment Diagram

A deployment diagram is a type of structure diagram that shows the architecture of the system. For example, how Artifacts from a software system constitute deployment targets (nodes). An Artifact represents a specific element in the real world. Entities are usually the result of the development process, such as executable files, libraries, archives, database schemas, configuration files, and so on. Deployment target: typically expressed as a Node, it represents a hardware device or some software runtime environment. Nodes can form arbitrarily complex, network-connected systems through communication paths.

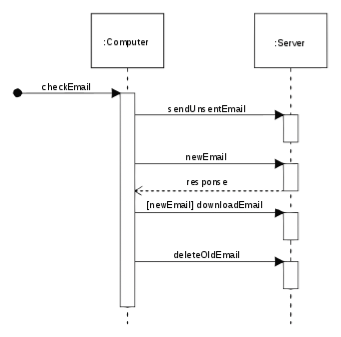
### D. Class Diagram

The Class diagram shows the static structure of the model, especially the classes that exist in the model, their internal structure, and their relationship to other classes. Class diagrams do not show transient information. Class diagrams are a major part of object-oriented modeling. It is used for both general conceptual modeling of the system classification of the application and detailed modeling, converting the model into programming code. Class diagrams can also be used for data modeling.

### E. State Diagram

The State Diagram describes the dynamic behavior of an entity based on event reactions, showing how the entity reacts to different events based on its current state. Typically, we create a UML state diagram for the following research purposes: to study the complex behavior of a class, role, subsystem, or component.

### F. Sequence Diagram



[Figure 1] Sequence Diagram Example

Sequence Diagram is a UML interaction Diagram. It shows dynamic collaboration between multiple objects by describing the time sequence in which messages are sent between them. Seven elements are involved when drawing a sequence diagram: roles, objects, lifelines, control focus, messages, self-associated messages, and composite fragments.

## 3. Applied Tools



[Figure 2] Power Point

상기한 다이어그램을 작성하기 위해 Power Point 툴을 사용하였다. 프레젠테이션 발표를 위해 개발된 툴이지만 내장된 도형 및 순서도 작성 기능이 강력하고 좋은 효율을 보이기 때문에 다이어그램 작성에 사용하였다.

## 4. Project Scope

Frontend Application은 사진 촬영 및 사진 불러오기, 마음에 드는 연예인 찾기 등 사용자와의 상호작용을 담당하며 Makeup Applying System을 포함한다.

Backend Application은 사용자가 요구한 기능의 연산을 담당하며 Data Collecting System과 Similar Celebrities Finding System을 포함한다.

Data Collecting System은 Similar Celebrities Finding System에 사용할 학습 데이터를 수집한다.

Similar Celebrities Finding System은 사용자가 제공한 사진과 비슷한 연예인을 찾아 반환한다.

Makeup Applying System은 사용자가 제공한 사진에 연예인의 화장을 덧입힌다.

# 3. Glossary

## 1. Objective

본 항목에서는 전문적 지식이 없는 독자들이 문서를 이해하는 데에 어려움이 없도록 본 문서에 사용된 기술적 용어들을 정의한다.

## 2. Term Definition

|  |  |
| --- | --- |
| Term | Definition |
| 자연어 | 사람들이 일상적으로 쓰는 언어. 인공적인 언어와 구분. |
| 머신러닝 | 인공지능의 연구 분야 중 하나. 인간의 학습 능력과 같은 기능을 컴퓨터에서 실현하고자 하는 기술 및 기법 |
| CNN모델 | 이미지의 공간 정보를 유지한 상태로 학습이 가능한 기계학습 신경망 모델 |
| Texture | 3D 그래픽에서 폴리곤으로 제작된 오브젝트에 덧씌워 다양한 색상이나 질감 등을 표현하는데 쓰이는 2D이미지 |
| Plan-driven | SW분야에서 시스템 요구사항, 소프트웨어 요구사항 등을 순차적으로 구성하고 실행하는 접근법 |
| 프론트 엔드 | 사용자로부터 다양한 형태의 입력을 받아 ‘백 엔드’가 사용할 수 있는 규격을 따라 처리한다. 일반적으로 클라이언트-서버 모델에서 클라이언트를 말함 |
| 백 엔드 | 클라이언트와 직접 대면하지 않으나, 프론트 엔드와 연동되어 기술적인 기능을 하는 부분. 일반적으로 클라이언트- 서버 모델에서 서버를 말함. |
| 오픈 소스 | 무상으로 공개된 소스코드 또는 소프트웨어 |
| 웹 크롤링 | 무수히 많은 컴퓨터에 분산 저장되어 있는 문서를 수집하여 검색 대상의 색인으로 포함시키는 기술. |
| 라이브러리 | 데이터의 관점에서는 관련한 파일의 집합을, 프로그램의 관점에서는 프로그램의 조직화된 집합을 말한다. |
| AWS | 아마존 웹 서비스, 웹사이트나 클라이언트 측 응용프로그램에 대해 온라인 서비스를 제공하고 있다. |
| Retrofit | 안드로이드 앱 및 JAVA언어에서 편리하게 사용할 수 있는 HTTP 클라이언트 라이브러리 |
| 컴포넌트 | 소프트웨어 개발을 마치 레고 블록을 쌓듯이 쉽게 할 수 있도록 하는 기술. 즉 소프트웨어 구성단위를 미리 만든 뒤 이를 조립하는 기술. |

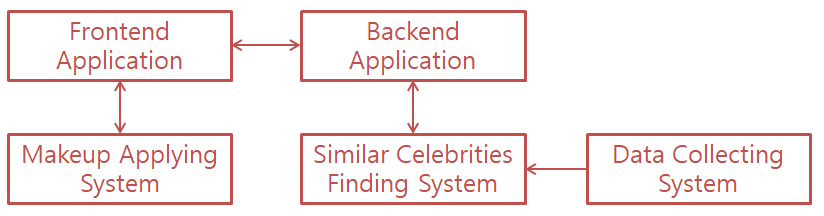
[Table 2] Term Definition

# 4. System Architecture

## 1. Objective

본 항목에서는 개발하고자 하는 시스템의 전체적인 설명을 기술한다. 각 시스템의 전체적인 구조는 블록 다이어그램으로 표시하고, 시스템 간의 관계 및 실제 사용 방법은 패키지 다이어그램 및 배포 다이어그램으로 설명한다.

## 2. System Organization

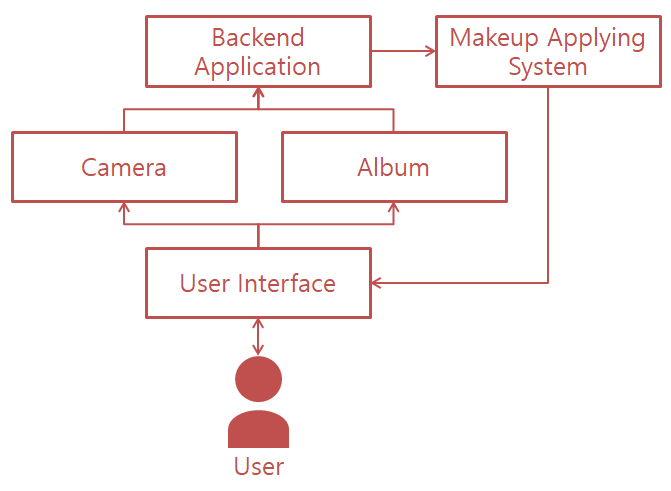


[Diagram 1] System organization

본 서비스는 클라이언트-서버 모델을 적용해 설계하였다. Frontend Application 이 사용자와의 상호작용을 담당하며 Backend Application에서 닮은 연예인을 찾아주는 기능을 담당한다.

Frontend Application에서 사용자의 사진을 받아 Backend Application으로 넘겨주면, Backend Application의 Similar Celebrities Finding System에서 Data Collecting System의 학습 데이터를 이용해 사용자가 제공한 사진과 비슷한 연예인을 찾아 다시 Frontend Application으로 넘겨준다. 사용자가 제공한 사진과 비슷한 연예인 리스트를 받은 Frontend Application은 사용자로부터 마음에 드는 연예인 정보를 받고, Makeup Applying System에서 사용자가 선택한 연예인의 화장을 사용자가 제공한 사진에 덧입혀준다.

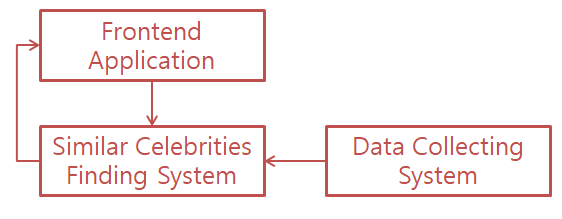
### A. Frontend Application



[Diagram 2] Frontend Application Architecture

사용자는 카메라 또는 앨범 어플리케이션을 통해 본인의 사진을 Backend Application으로 넘겨준다. Backend Application으로부터 사용자가 제공한 사진과 비슷한 연예인 리스트를 받은 Frontend Application은 사용자로부터 마음에 드는 연예인 정보를 받고, Makeup Applying System에서 사용자가 선택한 연예인의 화장을 사용자가 제공한 사진에 덧입혀준다.

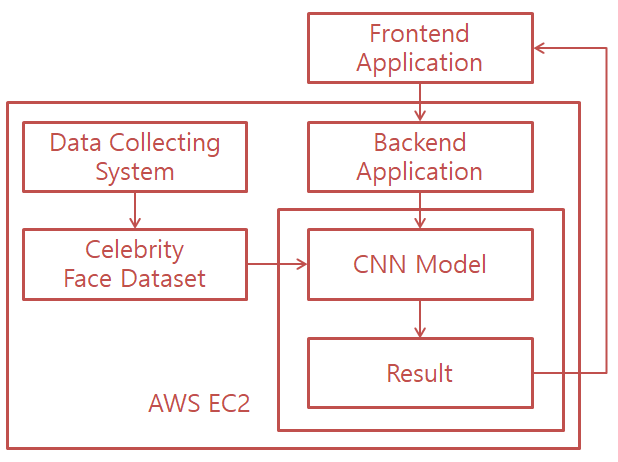
### B. Backend Application



[Diagram 3] Application Architecture

Frontend Application에서 사용자의 사진을 받아 Backend Application으로 넘겨주면, Backend Application의 Similar Celebrities Finding System에서 Data Collecting System의 학습 데이터를 이용해 사용자가 제공한 사진과 비슷한 연예인을 찾아 다시 Frontend Application으로 넘겨준다.

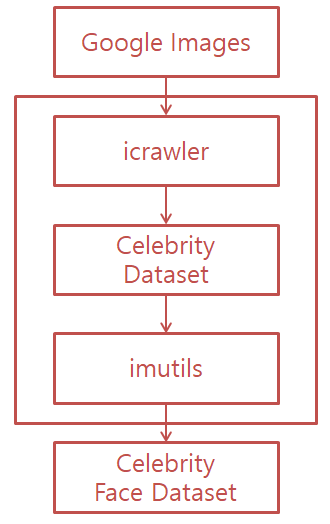
### C. Similar Celebrities Finding System



[Diagram 4] Similar Celebrities Finding System Architecture

사용자가 어플리케이션을 통해 사진을 제공하면 Retrofit 라이브러리를 이용해AWS EC2 서버에 python으로 구현된 CNN Model에 데이터를 넘겨준다. CNN Model에서는 제공받은 데이터를 이용해 닮은 연예인 리스트를 찾아내고, 결과 데이터는 다시 Retrofit 라이브러리를 통해 Application으로 넘어가며 사용자에게 돌아간다.

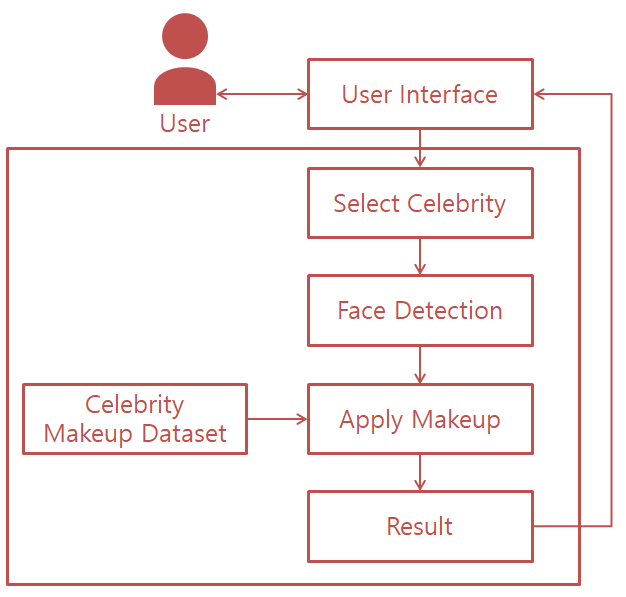
### D. Data Collecting System



[Diagram 5] Data Collecting System Architecture

Google image 검색을 통해 연예인의 사진 데이터를 구하고, 웹 크롤링 오픈 소스 라이브러리 icrawler를 이용해 데이터를 추출한다. 이미지 추출 오픈 소스 라이브러리 imutils를 이용해 연예인 이미지에서 얼굴을 추출한 뒤 64x64 크기로 정렬해 학습시키기 좋은 데이터로 만든다.

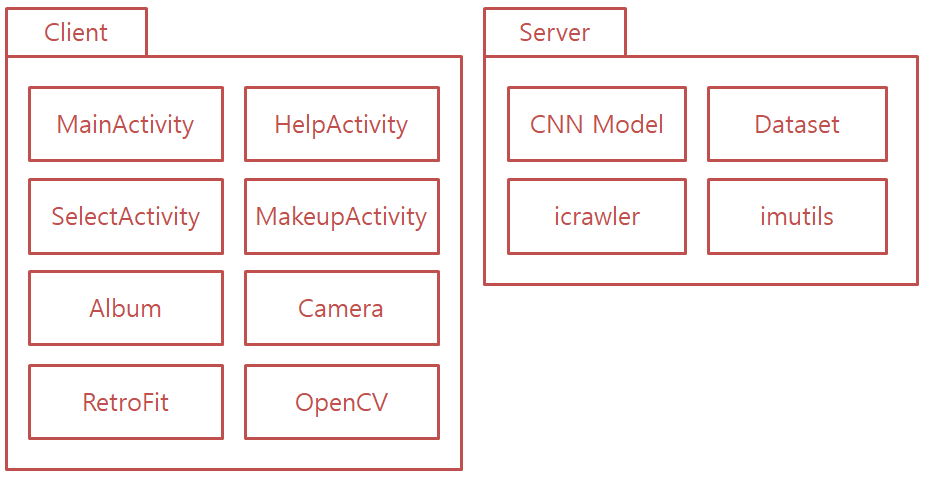
### E. Makeup Applying System



[Diagram 6] Makeup Applying System Architecture

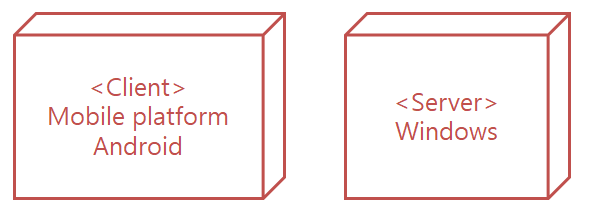
사용자가 원하는 연예인을 선택하면 OpenCV 라이브러리를 이용해 사용자가 제공한 사진에서 얼굴을 찾는다. 연예인 화장 texture를 OpenCV 라이브러리를 이용해 찾은 얼굴 영역에 덧입힌 뒤 결과 이미지를 사용자에게 보여준다.

## 3. Package Diagram



[Diagram 7] Package Diagram

## 4. Deployment Diagram



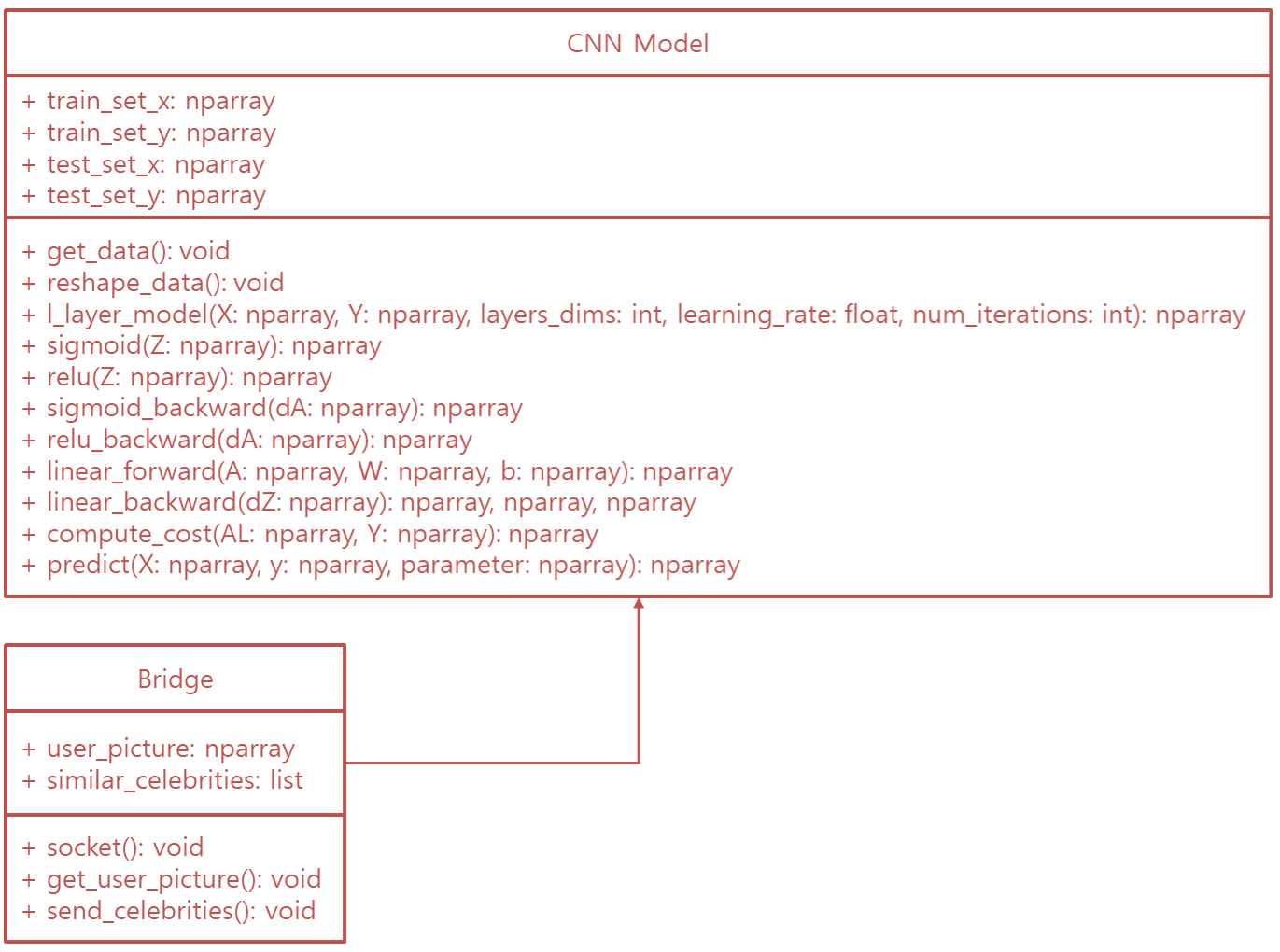
[Diagram 8] Deployment Diagram

# 5. Similar Celebrities Finding System

## 1. Objective

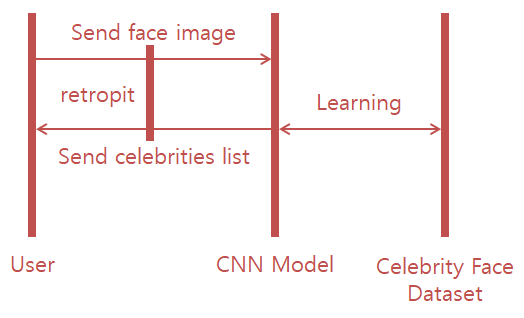
본 항목에서는 사용자가 제공한 사진과 비슷한 연예인을 찾아 돌려주는 시스템의 구조적인 디자인과 해당 시스템에 사용된 알고리즘을 기술한다. 본 시스템이 사용하는 기계학습 모델인 CNN 모델은 python과 tensorflow 모듈로 개발된 오픈 소스를 사용하여 조달한다.

## 2. Class Diagram



[Diagram 9] Similar Celebrities Finding System Class Diagram

## 3. Sequence Diagram



[Diagram 10] Similar Celebrities Finding System Sequence Diagram

## 4. State Diagram



[Diagram 11] Similar Celebrities Finding System Sequence Diagram

# 6. Data Collecting System

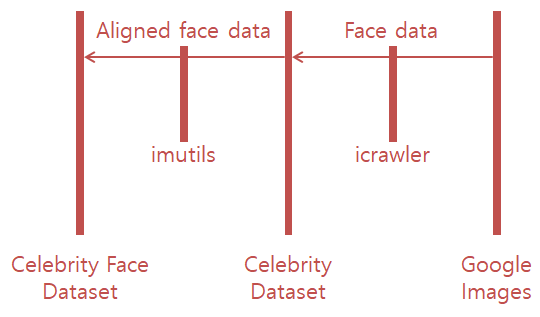
## 1. Objective

본 항목에서는 Similar celebrities finding system에서 사용할 학습 데이터를 수집하는 시스템의 구조적인 디자인과 데이터 수집 방법을 기술한다. Data collecting system 내부의 이미지 데이터 크롤링 시스템과 이미지 데이터 정렬 시스템은 python으로 개발된 오픈 소스를 사용하여 조달한다.

## 2. Class Diagram

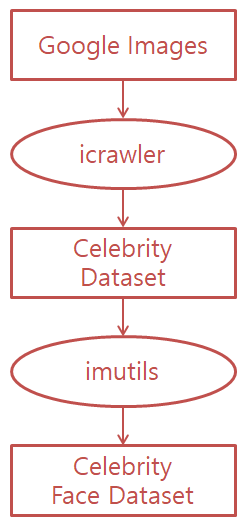
Data Collecting System의 imutils와 icrawler는 조달을 통해 구한 오픈 소스 라이브러리이며, 두 모듈에 의존을 갖는 class는 따로 존재하지 않는다.

## 3. Sequence Diagram



[Diagram 12] Data Collecting System Sequence Diagram

## 4. State Diagram



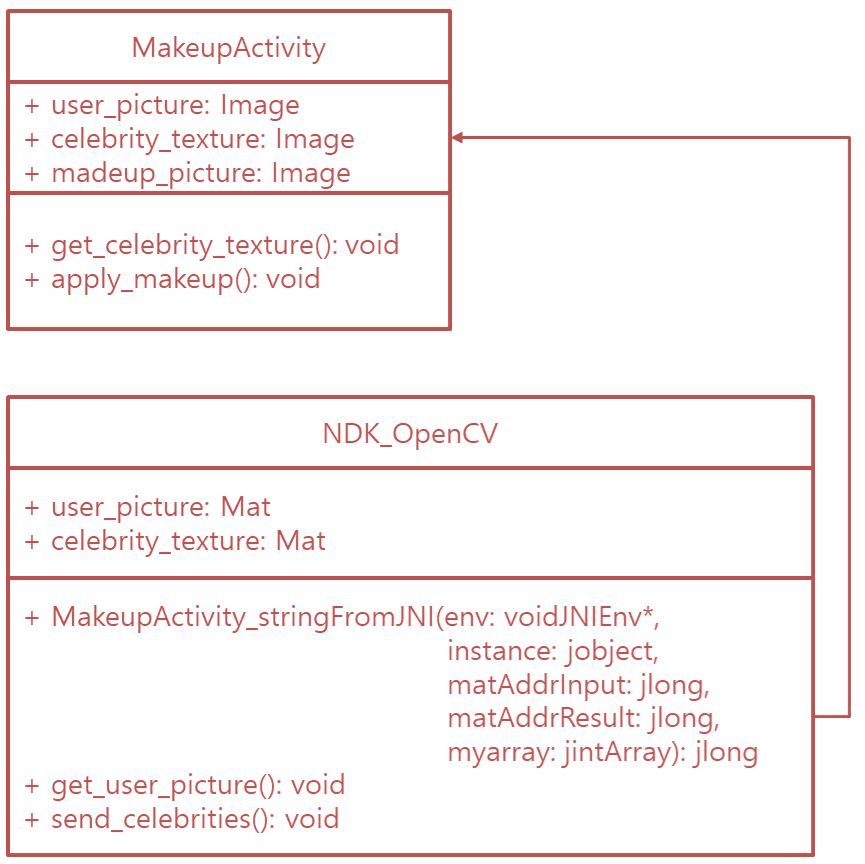
[Diagram 13] Data Collecting System State Diagram

# 7. Makeup Applying System

## 1. Objective

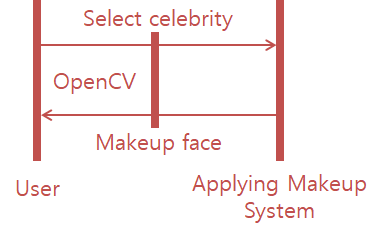
본 항목에서는 사용자가 제공한 사진에 사용자가 선택한 연예인의 화장법을 적용하는 시스템의 구조적인 디자인과 해당 시스템에 사용된 알고리즘을 기술한다. Makeup applying system은 얼굴 인식 시스템과 화장 적용 시스템을 포함하고 있으며, OpenCV 라이브러리를 이용하여 개발한다.

## 2. Class Diagram



[Diagram 14] Makeup Applying System Class Diagram

## 3. Sequence Diagram



[Diagram 15] Makeup Applying System Sequence Diagram

## 4. State Diagram



[Diagram 16] Makeup Applying System State Diagram

# 8. Testing Plan

## 1. Objective

본 항목에서는 테스트 정책 및 테스트 사례를 기술한다. 본 항목의 목적은 시스템 개발을 완료 한 후 시스템의 결함을 식별하고 시스템이 작동 중인지 확인 및 분석하기 위한 예비 계획을 수립하는 것이다. 테스트 정책 장과 테스트 사례 장으로 구분되며, 테스트 정책은 테스트 기준을 설정하고 테스트 사례는 구체적인 입력 및 출력 세트를 기술한다.

## 2. Testing Policy

### A. Development Testing

Development testing은 소프트웨어 개발의 전반적인 과정에 테스트를 실행하는 것을 의미한다. 본 테스트는 개발 도중 발생할 수 있는 시스템의 오류를 감지하고 방지할 뿐만 아니라, 시간과 비용 측면에서 시스템의 위험이 발생하지 않도록 하는 역할을 한다. 본 단계의 주된 목표는 시스템이 주어진 요구를 얼마나 잘 만족시키는지 평가하는 것이며, 테스트의 대상에 따라 시스템의 성능을 확인하는 것을 돕기 위해 여러 단계의 테스트, 즉 component testing, integration testing, system testing, acceptance testing으로 분리된다.

#### A. 1. Component Testing

Component는 테스트와 실행이 가능한 가장 작은 시스템의 단위를 의미하며 특정 기능이나 객체, 또는 이들의 그룹이 component가 될 수 있다. 본 단계에서는 개발자들에 의해 각각의 component들이 개별적으로 테스트를 수행하게 되며, 해당 component가 기능을 알맞게 구현하고 있는지 아닌지를 확인한다.

#### A. 2. Integration Testing

Integration testing은 component testing과 system testing의 중간 단계라고 할 수 있다. 관련 있는 시스템 구성요소들을 통합하여 하나의 그룹으로 묶고, 각각의 그룹이 통합되고 난 이후에도 기능을 잘 구현하는지를 확인한다. 본 단계는 테스터들에 의해 실행되며 만약 테스트를 통과하지 못한다면 system testing 단계로 넘어갈 수 없다.

#### A. 3. System Testing

System testing은 전체적으로 통합된 시스템에 테스트를 진행하는 것을 말한다. Component 사이의 모든 상호작용을 검사하고 특히 통합된 시스템으로서 가지는 emergent property를 테스트하는 것이 중요하다. 본 단계에서는 시스템이 specification을 만족하는지 아닌지를 확인하며, functional requirement와 non-functional requirement의 필요성을 평가한다.

#### A. 4. Acceptance Testing

Acceptance testing은 해당 시스템이 고객들의 요구나 specification을 충족시키는지 아닌지를 테스트한다. 기본적으로 사용자나 고객에 의해 본 단계의 테스트가 실행되지만, 다른 이해당사자들이 테스트를 수행하는 것 또한 가능하다.

### B. Release Testing

Release testing은 소프트웨어 시스템이 사용자들에게 배포될 수 있을지를 확인하기 위해 해당 시스템의 새로운 버전을 테스트하는 것이다. 명세서에 제시된 requirement들을 적절하게 만족시키는지 확인하고, 의도하지 않은 결함이 있는지 없는지 또한 확인해야 한다.

### C. User Testing

User testing은 사용자 혹은 고객이 시스템에 만족할 수 있는지를 확인하는 테스트 단계이다. 사용자와 개발자의 환경이 다를 수밖에 없기 때문에 user testing은 빠뜨려서는 안 되는 필수적인 단계이다. 사용자의 환경 속에서 사용자가 직접 시스템을 테스트하는 방식으로 진행된다.

## 3. Test Case

### A. Main System

#### A. 1. Select a Photo

**Test 1**

|  |  |
| --- | --- |
| 사용자가 앨범에서 선택한 사진 파일의 크기가 4MB이상인 경우 | |
| 성공 | 오류 메시지를 출력하여 사용자가 다른 사진을 선택하도록 한다. |
| 실패 | 선택한 사진이 화면에 출력되지 않는다. |

[Table 3] Test A. 1. 1

**Test 2**

|  |  |
| --- | --- |
| 사용자가 선택한 사진 파일의 이름에 한글이나 특수기호가 존재할 경우 | |
| 성공 | 복사본을 만들어 적합한 이름으로 변경하여 사진을 불러온다. |
| 실패 | 선택한 사진이 화면에 출력되지 않는다. |

[Table 4] Test A. 1. 2

**Test 3**

|  |  |
| --- | --- |
| 사용자가 선택한 사진에서 눈, 코, 입, 눈썹, 얼굴형 등의 얼굴 인식이  제대로 되지 않는 경우 | |
| 성공 | 오류 메시지를 출력한다. |
| 실패 | 현재 단계에서는 문제가 발생하지 않지만 similar celebrities finding system에서 오류가 발생한다. |

[Table 5] Test A. 1. 3

#### A. 2. Take a Photo

**Test 1**

|  |  |
| --- | --- |
| 사용자가 촬영한 사진 파일의 크기가 4MB이상인 경우 | |
| 성공 | 오류 메시지를 출력한다. |
| 실패 | 선택한 사진이 화면에 출력되지 않는다. |

[Table 6] Test A. 2. 1

**Test 2**

|  |  |
| --- | --- |
| 사용자가 촬영한 사진에서 눈, 코, 입, 눈썹, 얼굴형 등의 얼굴 인식이  제대로 되지 않는 경우 | |
| 성공 | 오류 메시지를 출력한다. |
| 실패 | 현재 단계에서는 문제가 발생하지 않지만 similar celebrities finding system에서 오류가 발생하게 된다. |

[Table 7] Test A. 2. 2

**Test 3**

|  |  |
| --- | --- |
| 단말 기기에서 카메라 기능을 지원하지 않는 경우 | |
| 성공 | 오류 메시지를 출력하고 앨범에서 사진 선택하는 화면으로 전환한다. |
| 실패 | 어플리케이션 작동이 중단된다. |

[Table 8] Test A. 2. 3

#### A. 3. Save the Result Photo

**Test 1**

|  |  |
| --- | --- |
| 결과 사진을 저장할 경로를 찾을 수 없는 경우 | |
| 성공 | 결과 사진을 저장할 새로운 경로를 찾거나 새로운 폴더를 생성하여 저장한다. |
| 실패 | 결과 사진을 저장하지 못하고 사용자의 기기에서 사진을 찾을 수 없다. |

[Table 9] Test A. 3. 1

**Test 2**

|  |  |
| --- | --- |
| 파일 저장에 대한 접근 권한이 없을 경우 | |
| 성공 | 오류 메시지를 출력하고 접근 권한을 허용 받는 팝업을 출력한다. |
| 실패 | 결과 사진을 저장하지 못한다. |

[Table 10] Test A. 3. 2

**Test 3**

|  |  |
| --- | --- |
| 저장 공간이 부족할 경우 | |
| 성공 | 오류 메시지를 출력하여 사용자가 저장 공간을 확보하도록 유도한다. |
| 실패 | 결과 사진을 저장하지 못한다. |

[Table 11] Test A. 3. 3

### B. Similar Celebrities Finding System

**Test 1**

|  |  |
| --- | --- |
| 데이터 전송 중 인터넷이 끊겼을 경우 | |
| 성공 | 데이터 전송을 중단하고 오류 메시지를 출력한다. 인터넷이 다시 연결되면 데이터 전송을 다시 시작한다. |
| 실패 | 데이터가 전부 제대로 전송되지 않는다. |

[Table 12] Test B. 1

**Test 2**

|  |  |
| --- | --- |
| 일정 비율 이상으로 비슷한 얼굴 특징을 가진 연예인이  단 한 명도 존재하지 않을 경우 | |
| 성공 | 오류 메시지를 출력하고 사용자가 다른 사진을 통해 다시 시도하도록 유도한다. |
| 실패 | 텅 빈 목록을 화면에 출력한다. |

[Table 13] Test B. 2

**Test 3**

|  |  |
| --- | --- |
| 머신 러닝을 수행할 서버에서 오류가 발생할 경우 | |
| 성공 | 서버 오류를 해결한 뒤 처음부터 다시 실행한다. |
| 실패 | 서버로 사진을 전송했지만 서버에서 오류가 발생했으므로 결과 사진을 받지 못한다. |

[Table 14] Test B. 3

### C. Data Collecting System

**Test 1**

|  |  |
| --- | --- |
| 수집한 사진에서 눈, 코, 입, 눈썹, 얼굴형 등의 얼굴 인식이 제대로 되지 않을 경우 | |
| 성공 | 해당 사진을 수집된 데이터 목록에 추가하지 않는다. |
| 실패 | 해당 사진을 수집된 데이터 목록에 추가한다. 얼굴 인식이 되지 않으므로 데이터 목록에 추가되기만 할 뿐 기타 정보는 잘못 설정된다. |

[Table 15] Test C. 1

**Test 2**

|  |  |
| --- | --- |
| 해당 인물 사진이 특정 유명인으로 인식되지 않을 경우 | |
| 성공 | 해당 사진을 수집된 데이터 목록에 추가하지 않는다. |
| 실패 | 해당 사진의 유명인이 잘못 설정된 상태로 데이터 목록에 추가된다. |

[Table 16] Test C. 2

**Test 3**

|  |  |
| --- | --- |
| 사진 파일의 크기가 4MB 이상일 경우 | |
| 성공 | 해당 사진을 축소하여 데이터 목록에 추가한다. |
| 실패 | 사진 파일이 깨진 상태로 데이터 목록에 추가된다 |

[Table 17] Test C. 3

**Test 4**

|  |  |
| --- | --- |
| 화장법을 추출해낼 수 없을 정도로 얼굴이 비대칭적으로 사진에 나와있을 경우 | |
| 성공 | 해당 사진을 데이터 목록에 추가하지 않는다. |
| 실패 | 얼굴 정보의 일부분만을 인식하게 되므로 해당 사진의 유명인을 잘못 판단할 가능성이 존재한다. |

[Table 18] Test C. 4

### D. Makeup Applying System

#### D. 1. Extracting Makeup

**Test 1**

|  |  |
| --- | --- |
| 화장만을 추출해 가져오는 것이 아닌 얼굴 구성요소도 함께 추출될 경우 | |
| 성공 | 잘못된 부분을 수정하여 화장 추출을 다시 시도한다. |
| 실패 | 현재 단계에서는 겉으로 보이는 오류가 드러나지 않지만 applying makeup 단계에서 사용자의 이목구비 자체가 달라질 가능성이 존재한다. |

[Table 19] Test D. 1. 1

#### D. 2. Applying Makeup

**Test 1**

|  |  |
| --- | --- |
| 화장을 적용하는 과정에서 사진의 화질이 크게 저하될 경우 | |
| 성공 | 설정을 변경하여 화질이 저하되지 않도록 다시 화장을 적용한다. |
| 실패 | 화질이 크게 저하된 상태 그대로 사용자에게 결과 사진을 보여준다. |

[Table 20] Test D. 2. 1

**Test 2**

|  |  |
| --- | --- |
| 화장이 적용된 결과 사진을 서버에서 받아오는 도중 인터넷이 중단될 경우 | |
| 성공 | 오류 메시지를 출력하고 인터넷이 다시 연결되면 결과 사진을 서버에서 받아온다. |
| 실패 | 결과 화면에 아무것도 출력되지 않는다. |

[Table 21] Test D. 2. 2

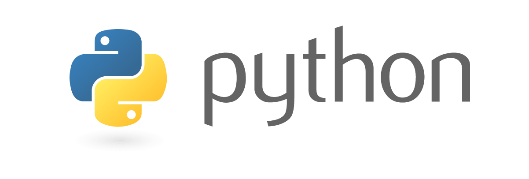
# 9. Development Plan

## 1. Objective

본 항목에서는 시스템 개발에 사용할 프로그래밍 언어와 개발 IDE를 기술하고, 개발 일정을 서술한다.

## 2. Programming Language & IDE

### A. Programming Language

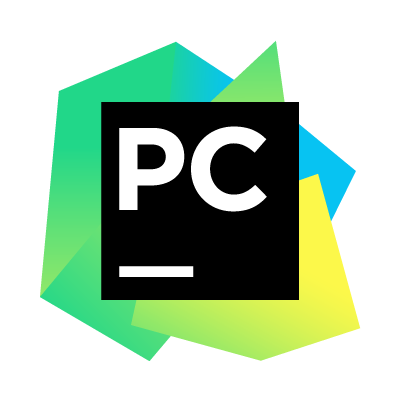
 

[Figure 3] Programming Language

본 어플리케이션은 기계학습 모델인 CNN 모델을 사용하며, tensorflow 모듈로 개발된 오픈 소스를 사용할 예정이기 때문에 Python을 주 언어로 사용한다. Python을 통해 서버사이드에서 이루어지는 모든 작업(기계학습, 웹 크롤링 등)을 진행한다. Python의 버전은 최신 버전인 3.8.0으로 통일한다.

모바일 어플리케이션으로 서비스를 하기 때문에 안드로이드 개발에 필요한 JAVA를 사용한다. 호환성문제나 오픈 소스 사용 여부 및 참조의 문제로 최신버전의 사용이 아닌 jdk1.8 버전을 사용한다. 또한 이는 자바의 표준안인 JavaSE중 JavaSE8을 사용함을 의미한다.

### B. IDE

[Figure 4] IDE

Python 개발 툴 중 가장 높은 완성도를 지녔다고 알려진 JetBrains에서 제작한 Python용 IDE인 Pycharm을 사용한다. 또한 무료버젼인 Community에디션으로 통일한다. Pycharm은 통합 개발환경으로써 코드 자동완성 기능을 제공한다.

모바일 어플리케이션을 개발하기 위해서 안드로이드 스튜디오를 사용한다. 안드로이드 스튜디오의 버전은 가장 최신버전인 3.5.2버전으로 통일한다. Android SDK Tools의 버전 또한 가장 최신인 26.1.1버전으로 통일한다. 단 안드로이드 OS의 버전은 호환성문제의 개선을 위해 가장 최신버전인 Android 10.0(Q)를 사용하지 않고 Android 9.0(Pie)를 사용한다. 또한 해당버전에서 Neural Networks API가 확장되고 개선되어 그 사용이 더욱 용이할 것이다.

## 3. Schedule

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | 수행내용 | 일정(주) | | | | | | | | | |
| 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
| 1 | **아이디어 제안**  **및 컨셉 결정** |  |  |  |  |  |  |  |  |  |  |
| 2 | **Requirement Specification 작성** |  |  |  |  |  |  |  |  |  |  |
| 3 | **Design**  **Specification 작성** |  |  |  |  |  |  |  |  |  |  |
| 4 | **개발환경 세팅** |  |  |  |  |  |  |  |  |  |  |
| 5 | **개발 - front end** |  |  |  |  |  |  |  |  |  |  |
| 6 | **개발 - back end** |  |  |  |  |  |  |  |  |  |  |
| 7 | **Test** |  |  |  |  |  |  |  |  |  |  |

[Table 22] Schedule

# 10. Index

## 1. Objective

본 항목에서는 본 문서에 사용된 표, 그림, 다이어그램의 색인을 기술한다.

## 2. Table Index

[[Table 1] Version History 8](#_Toc24578809)

[[Table 2] Term Definition 14](#_Toc24578810)

[[Table 3] Test A. 1. 1 29](#_Toc24578811)

[[Table 4] Test A. 1. 2 29](#_Toc24578812)

[[Table 5] Test A. 1. 3 29](#_Toc24578813)

[[Table 6] Test A. 2. 1 30](#_Toc24578814)

[[Table 7] Test A. 2. 2 30](#_Toc24578815)

[[Table 8] Test A. 2. 3 30](#_Toc24578816)

[[Table 9] Test A. 3. 1 31](#_Toc24578817)

[[Table 10] Test A. 3. 2 31](#_Toc24578818)

[[Table 11] Test A. 3. 3 31](#_Toc24578819)

[[Table 12] Test B. 1 32](#_Toc24578820)

[[Table 13] Test B. 2 32](#_Toc24578821)

[[Table 14] Test B. 3 32](#_Toc24578822)

[[Table 15] Test C. 1 33](#_Toc24578823)

[[Table 16] Test C. 2 33](#_Toc24578824)

[[Table 17] Test C. 3 33](#_Toc24578825)

[[Table 18] Test C. 4 33](#_Toc24578826)

[[Table 19] Test D. 1. 1 34](#_Toc24578827)

[[Table 20] Test D. 2. 1 34](#_Toc24578828)

[[Table 21] Test D. 2. 2 34](#_Toc24578829)

[[Table 22] Schedule 37](#_Toc24578830)

## 3. Figure Index

[[Figure 1] Sequence Diagram Example 11](#_Toc24578831)

[[Figure 2] Power Point 12](#_Toc24578832)

[[Figure 3] Programming Language 35](#_Toc24578833)

[[Figure 4] IDE 36](#_Toc24578834)

## 4. Diagram Index

[[Diagram 1] System organization 15](#_Toc24578835)

[[Diagram 2] Frontend Application Architecture 16](#_Toc24578836)

[[Diagram 3] Application Architecture 16](#_Toc24578837)

[[Diagram 4] Similar Celebrities Finding System Architecture 17](#_Toc24578838)

[[Diagram 5] Data Collecting System Architecture 17](#_Toc24578839)

[[Diagram 6] Makeup Applying System Architecture 18](#_Toc24578840)

[[Diagram 7] Package Diagram 19](#_Toc24578841)

[[Diagram 8] Deployment Diagram 19](#_Toc24578842)

[[Diagram 9] Similar Celebrities Finding System Class Diagram 20](#_Toc24578843)

[[Diagram 10] Similar Celebrities Finding System Sequence Diagram 21](#_Toc24578844)

[[Diagram 11] Similar Celebrities Finding System Sequence Diagram 21](#_Toc24578845)

[[Diagram 12] Data Collecting System Sequence Diagram 22](#_Toc24578846)

[[Diagram 13] Data Collecting System State Diagram 23](#_Toc24578847)

[[Diagram 14] Makeup Applying System Class Diagram 24](#_Toc24578848)

[[Diagram 15] Makeup Applying System Sequence Diagram 25](#_Toc24578849)

[[Diagram 16] Makeup Applying System State Diagram 25](#_Toc24578850)