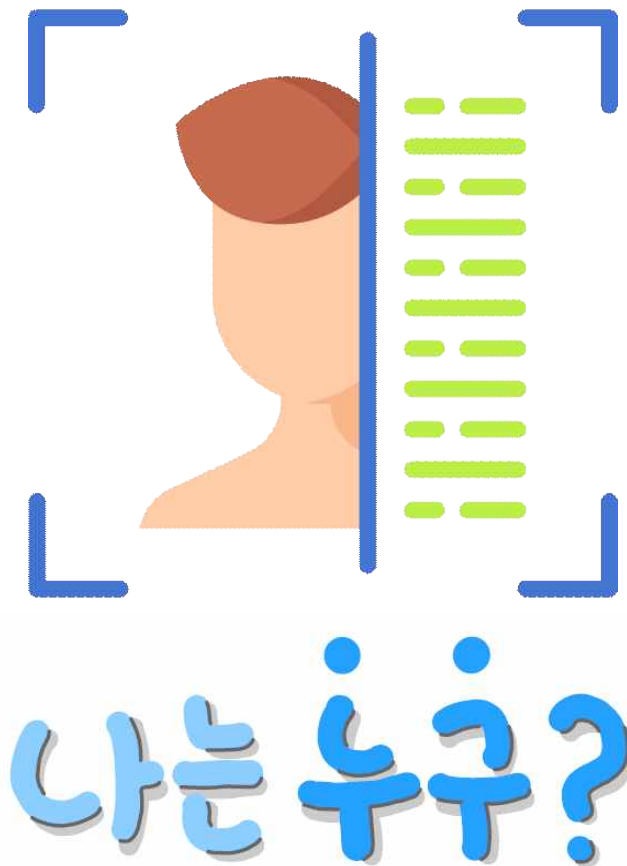


# 나는 누구?

- 사진을 이용한 얼굴분석 시스템 -



당신의 얼굴에 숨겨진 아이돌 얼굴을 찾아드립니다.

## [ Revision history ]

Revision date	Version #	Description	Author
11/05/2021	1.00	First Document	허재윤, 김태환, 조현근, 김예빈

= Contents =

1. Introduction .....	1
2. Use case analysis .....	2
2.1) Use case diagram .....	2
2.2) Use case description .....	3
3. Class diagram .....	18
3.1) Class diagram .....	18
3.2) Class description .....	19
4. User interface prototype .....	37
5. Glossary .....	49
6. References .....	49

## 1. Introduction

---

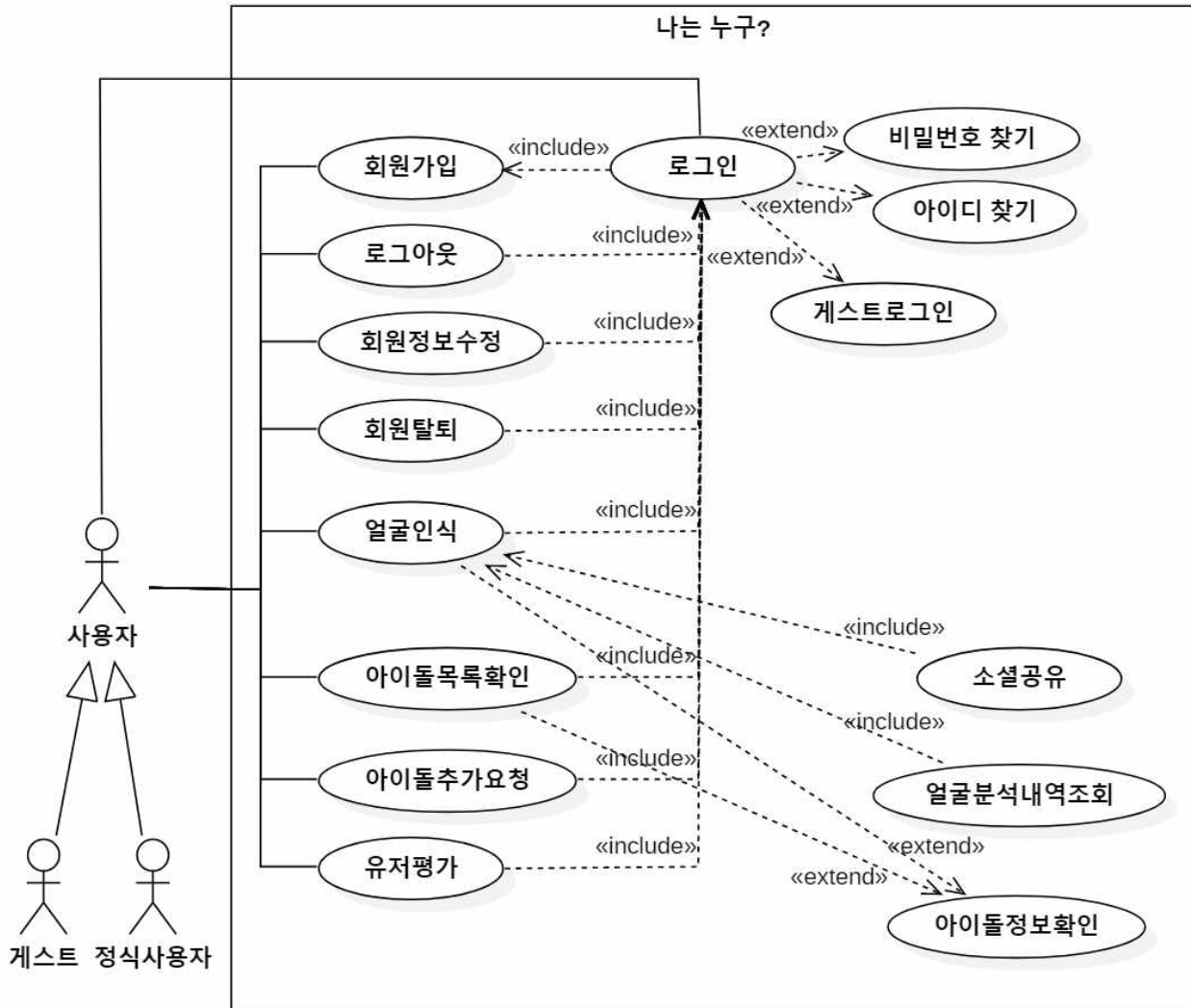
현재 우리는 4차 산업혁명을 맞이하여 컴퓨팅 기술과 인공지능 기술이 크게 발전하였다. 이를 기반으로 하여 다른 여러 기술들이 등장하면서 함께 발전되었는데 그 중 하나가 얼굴인식기술이다. 얼굴인식기술은 얼굴을 포함하는 사진 또는 영상에 대한 얼굴 영역의 자동적인 검출 및 분석을 통해 해당 얼굴이 어떤 인물인지 판별해 내는 기술로 패턴인식 및 컴퓨터 비전 분야에서 오랫동안 연구되어 온 분야이다. 많은 연구결과로부터 감시시스템, 출입국관리, 생체인식 등과 같이 실제 여러 환경에서 적용되고 있다. 특히 최근 코로나 감염질환의 확산을 극복하기 위한 비대면 기술의 일종으로 여겨져 사회적 역할과 관심, 필요성이 더욱 높아지고 있는 추세이다. 우리는 이렇게 각광받고 있는 얼굴인식기술을 대중들이 좀 더 재밌게 접해볼 수 있도록 ‘나는 누구?’라는 애플리케이션을 개발하게 되었다.

‘나는 누구?’ 애플리케이션은 모델에 학습되어 있는 아이돌 사진을 바탕으로 사용자의 얼굴이 담긴 사진을 분석하여 가장 유사한 아이돌을 찾아주는 프로그램이다. 사진은 이미 저장되어 있던 것을 사용하거나 실시간으로 찍어서 올릴 수 있다. 또한 분석결과를 통해 해당 아이돌의 기본적인 정보를 알 수 있고, 소셜공유기능으로 다른 사용자와 결과를 공유할 수도 있다. 얼굴인식기능은 학습된 정보에 한하여 결과를 출력한다는 한계가 있어 사용자의 피드백을 통한 좀 더 폭 넓은 정보 추가 및 보완이 가능하도록 학습되어 있지 않은 아이돌을 추가 요청할 수 있다. 또한 프로그램으로 도출된 결과가 사용자입장에서 얼마나 정확한가를 평가할 수 있다.

위와 같은 기능을 통해 대중들이 흥미롭게 얼굴인식기술을 접하도록 하는 것이 이번 프로젝트의 목표이다.

## 2. Use case analysis

### 2.1 Use case diagram



위 Diagram은 '나는 누구?'에서 수행 가능한 기능과 그들 간의 관계, 사용자와의 관계를 나타낸 Diagram이다. 사용자가 사용 가능한 기능을 실선으로 연결하였고, 기능 간의 관계를 점선으로 연결하였다.

사용자는 크게 게스트와 정식사용자 2가지 타입으로 나뉘는데, 게스트는 로그인 과정을 거치지 않고 해당 프로그램을 이용하는 사용자로 아이돌 추가요청, 유저평가, 얼굴분석내역조회는 이용할 수 없다. 반면에 로그인 과정을 거치고 이용하는 정식사용자의 경우 모든 기능을 이용할 수 있다.

## 2.2 Use case description

Use case #1: 로그인	
GENERAL CHARACTERISTICS	
Summary	나는누구?에 등록된 계정으로 사용자 구분
Scope	나는누구?
Level	User level
Author	허재윤
Last Update	2021.10.25
Status	Analysis
Primary Actor	anyone
Preconditions	로그인하려고 하는 계정이 나는누구?에 회원가입을 통해 등록되어 있는 상태여야 한다.
Trigger	로그인하지 않은 사용자가 로그인 화면에서 ID 및 PW 정보 입력 후 로그인 버튼을 클릭했을 때
Success Post Condition	사용자는 나는누구?에 정상적으로 로그인하고 이미지 입력 페이지로 이동하며 대부분의 기능을 이용할 수 있다.
Failed Post Condition	로그인에 실패한다.
MAIN SUCCESS SCENARIO	
Step	Action
S	사용자는 나는누구?를 실행한다.
1	이 use case는 사용자가 나는누구?에 로그인할 때 시작된다.
2	사용자는 ID 와 Password를 입력하고 로그인 버튼을 클릭한다.
3	나는누구?에 등록된 사용자인지 체크 해보고 등록된 회원이라면 로그인에 성공한다.
4	이 Use case는 로그인이 성공하면 끝난다.
EXTENSION SCENARIOS	
Step	Branching Action
3	3a. 아이디나 비밀번호가 잘못되어 로그인에 실패한다. ...3a1. 아이디나 비밀번호가 잘못되었다는 메시지를 보여준다. ...3a2. 로그인 UI를 유지한다.
RELATED INFORMATION	
Performance	≤ 2 seconds
Frequency	사용자당 하루에 평균 2 번
<Concurrency>	제한 없음
Due Date	

Use case #2: 회원가입	
GENERAL CHARACTERISTICS	
Summary	나는누구?에 계정을 등록하기 위한 회원가입 기능
Scope	나는누구?
Level	User level
Author	허재윤
Last Update	2021.10.25
Status	Analysis
Primary Actor	anyone
Preconditions	나는누구? 어플리케이션을 정상적으로 실행한 상태여야 한다.
Trigger	로그인하지 않은 사용자가 로그인 화면에서 회원가입 버튼을 클릭했을 때
Success Post Condition	사용자의 회원 정보가 등록되고, 로그인 화면으로 이동한다.
Failed Post Condition	회원 가입에 실패한다.
MAIN SUCCESS SCENARIO	
Step	Action
S	사용자는 나는누구?를 실행한다.
1	이 use case는 사용자가 로그인 화면에서 회원가입을 클릭하면 시작된다.
2	사용자는 ID, PW, 이름, 성별, 나이, 별명, 이메일을 작성한다.
3	시스템은 사용자가 작성한 모든 항목이 정상적으로 작성되었는지 확인한다.
4	시스템은 작성되어있는 항목들을 DB에 저장한다.
5	이 Use case는 회원가입이 성공하면 끝난다.
EXTENSION SCENARIOS	
Step	Branching Action
3	3a. 작성되지 않은 항목이 존재한다. ...3a1. 작성되지 않은 정보가 있다는 오류메시지를 출력한다. ...3a2. 회원가입 UI를 유지한다.
4	4a. 범위를 벗어난 입력값이 존재한다. ...4a1. 잘못 기입했다는 메시지를 출력한다. ...4a2. 회원가입 UI를 유지한다.
RELATED INFORMATION	
Performance	≤ 2 seconds
Frequency	사용자당 평균 2회
<Concurrency>	제한 없음
Due Date	

Use case #3: 아이디 찾기	
GENERAL CHARACTERISTICS	
Summary	나는누구?에 등록된 계정을 확인하는 기능
Scope	나는누구?
Level	User level
Author	허재윤
Last Update	2021.10.25
Status	Analysis
Primary Actor	나는누구? user
Preconditions	나는누구? 어플리케이션을 정상적으로 실행한 상태여야 한다.
Trigger	로그인하지 않은 사용자가 로그인 화면에서 아이디 찾기를 클릭했을 때
Success Post Condition	사용자가 기입한 정보와 일치하는 ID 목록을 보여준다.
Failed Post Condition	아이디 찾기에 실패한다.
MAIN SUCCESS SCENARIO	
Step	Action
S	사용자는 나는누구?를 실행한다.
1	이 use case는 사용자가 로그인 화면에서 아이디 찾기를 클릭하면 시작된다.
2	사용자는 이름, 나이, 이메일을 입력한다.
3	시스템은 사용자가 작성한 모든 항목이 정상적으로 작성되었는지 확인한다.
4	시스템은 작성되어있는 항목들을 DB에 검색한다.
5	검색된 항목들을 화면에 나열한다.
6	이 Use case는 아이디를 찾으면 종료된다.
EXTENSION SCENARIOS	
Step	Branching Action
3	3a. 작성되지 않은 항목이 존재한다. ...3a1. 작성되지 않은 정보가 있다는 오류메시지를 출력한다. ...3a2. 아이디 찾기 UI를 유지한다.
4	4a. 작성한 정보와 일치하는 정보를 DB에서 찾지 못했다. ...4a1. 사용자 정보가 등록되어 있지 않다는 메시지를 출력한다. ...4a2. 아이디 찾기 UI를 유지한다.
RELATED INFORMATION	
Performance	≤ 2 seconds
Frequency	사용자당 하루 평균 1회
<Concurrency>	제한 없음
Due Date	



Use case #4: 비밀번호 찾기	
GENERAL CHARACTERISTICS	
Summary	나는누구?에 등록된 계정의 비밀번호를 확인하는 기능
Scope	나는누구?
Level	User level
Author	허재운
Last Update	2021.10.25
Status	Analysis
Primary Actor	나는누구? user
Preconditions	나는누구? 어플리케이션을 정상적으로 실행한 상태여야 한다.
Trigger	로그인하지 않은 사용자가 로그인 화면에서 비밀번호 찾기를 클릭했을 때
Success Post Condition	사용자가 회원가입 시 입력한 이메일로 비밀번호를 전송한다.
Failed Post Condition	비밀번호 찾기에 실패한다.
MAIN SUCCESS SCENARIO	
Step	Action
S	사용자는 나는누구?를 실행한다.
1	이 Use case는 사용자가 로그인 화면에서 비밀번호 찾기를 클릭하면 시작된다.
2	사용자는 ID, 이름, 나이, 이메일을 입력한다.
3	시스템은 사용자가 작성한 모든 항목이 정상적으로 작성되었는지 확인한다.
4	시스템은 작성되어있는 항목들을 DB에 검색한다.
5	검색된 유저 정보에 저장되어 있는 이메일로 임의로 변경한 이메일을 전송한다.
6	이 Use case는 비밀번호를 찾으면 종료된다.
EXTENSION SCENARIOS	
Step	Branching Action
3	3a. 작성되지 않은 항목이 존재한다. ...3a1. 작성되지 않은 정보가 있다는 오류메시지를 출력한다. ...3a2. 비밀번호 찾기 UI를 유지한다.
4	4a. 작성한 정보와 일치하는 정보를 DB에서 찾지 못했다. ...4a1. 사용자 정보가 등록되어 있지 않다는 메시지를 출력한다. ...4a2. 아이디 찾기 UI를 유지한다.
RELATED INFORMATION	
Performance	≤ 2 seconds
Frequency	사용자당 하루 평균 1회
<Concurrency>	제한 없음
Due Date	

Use case #5: 게스트 로그인	
GENERAL CHARACTERISTICS	
Summary	나는누구?에 계정을 등록하지 않고 사용하기 위한 임의 로그인
Scope	나는누구?
Level	User level
Author	허재윤
Last Update	2021.10.25
Status	Analysis
Primary Actor	anyone
Preconditions	나는누구? 어플리케이션을 정상적으로 실행한 상태여야 한다.
Trigger	로그인하지 않은 사용자가 로그인 화면에서 게스트 로그인을 클릭했을 때
Success Post Condition	얼굴 분석 기능을 이용할 수 있는 화면으로 넘어간다.
Failed Post Condition	게스트 로그인에 실패한다.
MAIN SUCCESS SCENARIO	
Step	Action
S	사용자는 나는누구?를 실행한다.
1	이 Use case는 사용자가 로그인 화면에서 게스트 로그인을 클릭하면 시작된다.
2	시스템은 사용자에게 임의 ID를 부여한다.
3	임의 ID로 얼굴 분석 기능을 이용할 수 있도록 화면이 전환된다.
RELATED INFORMATION	
Performance	≤ 2 seconds
Frequency	사용자당 하루 평균 1회
<Concurrency>	제한 없음
Due Date	

Use case #6: 회원정보 수정	
GENERAL CHARACTERISTICS	
Summary	나는누구?에 등록된 사용자 정보를 수정하는 기능
Scope	나는누구?
Level	User level
Author	허재운
Last Update	2021.10.25
Status	Analysis
Primary Actor	나는누구? user
Preconditions	나는누구?에 등록된 계정으로 로그인 되어있어야 한다.
Trigger	나는누구?에 등록된 계정으로 로그인한 사용자가 메뉴에서 회원정보 수정을 클릭했을 때
Success Post Condition	저장되어 있는 사용자 정보를 변경 후 저장한다.
Failed Post Condition	회원정보 수정에 실패한다.
MAIN SUCCESS SCENARIO	
Step	Action
S	사용자는 나는누구?에 로그인한다.
1	이 Use case는 사용자가 메인 메뉴에서 회원정보 수정을 클릭하면 실행된다.
2	시스템은 사용자에게 현재 등록되어 있는 회원정보를 나열한다.
3	사용자는 변경을 원하는 항목을 텍스트 박스를 이용해 수정할 수 있다.
4	시스템은 사용자가 입력한 정보가 유효한지 검사한다.
5	시스템은 입력된 정보를 DB에서 찾아 수정한다.
6	이 Use case는 회원 정보를 변경하면 종료된다.
EXTENSION SCENARIOS	
Step	Branching Action
3	3a. 사용자가 취소 버튼을 클릭한다. ...3a1. 메인 메뉴 UI로 돌아간다.
4	4a. 입력한 정보가 유효하지 않은 입력값을 갖는다. ...4a1. 유효하지 않은 정보라는 메시지를 출력한다. ...4a2. 회원정보 수정 UI를 유지한다.
RELATED INFORMATION	
Performance	≤ 2 seconds
Frequency	
<Concurrency>	제한 없음
Due Date	

Use case #7: 로그아웃	
GENERAL CHARACTERISTICS	
Summary	나는누구?에 로그인 된 상태를 종료하는 기능
Scope	나는누구?
Level	User level
Author	허재윤
Last Update	2021.10.25
Status	Analysis
Primary Actor	나는누구? user
Preconditions	나는누구?에 등록된 계정으로 로그인 되어있어야 한다.
Trigger	나는누구?에 등록된 계정으로 로그인한 사용자가 메뉴에서 로그아웃을 클릭했을 때
Success Post Condition	로그인 화면으로 돌아간다.
MAIN SUCCESS SCENARIO	
Step	Action
S	사용자는 나는누구?에 로그인한다.
1	이 Use case는 사용자가 메인 메뉴에서 로그아웃을 클릭하면 실행된다.
2	시스템은 로그아웃 되었다는 메시지를 출력한다.
3	로그인 UI로 돌아간다.
RELATED INFORMATION	
Performance	≤ 2 seconds
Frequency	
<Concurrency>	제한 없음
Due Date	

Use case #8: 회원 탈퇴	
GENERAL CHARACTERISTICS	
Summary	나는누구?에 등록된 사용자 정보를 삭제하는 기능
Scope	나는누구?
Level	User level
Author	허재운
Last Update	2021.10.25
Status	Analysis
Primary Actor	나는누구? user
Preconditions	나는누구?에 등록된 계정으로 로그인 되어있어야 한다.
Trigger	사용자가 메인 메뉴에서 회원 탈퇴를 클릭했을 때
Success Post Condition	회원 탈퇴가 정상적으로 이루어지고 등록된 사용자 정보가 DB에서 삭제된다.
Failed Post Condition	회원 탈퇴에 실패한다.
MAIN SUCCESS SCENARIO	
Step	Action
S	사용자는 나는누구?에 로그인한다.
1	이 Use case는 사용자가 메인 메뉴에서 회원 탈퇴를 클릭하면 실행된다.
2	시스템은 사용자에게 비밀번호 입력 텍스트 박스를 제공한다.
3	사용자는 비밀번호를 입력한다.
4	시스템은 사용자가 입력한 비밀번호가 일치하는지 확인한다.
5	시스템은 사용자에게 ‘정말 탈퇴하시겠습니까? 회원 탈퇴하면 사용 기록이 모두 삭제됩니다.’ 라는 메시지를 출력한다.
6	사용자가 확인 버튼을 누르면 로그인 UI로 이동한다.
EXTENSION SCENARIOS	
Step	Branching Action
4	4a. 비밀번호가 일치하지 않는다. ...4a1. 유효하지 않은 비밀번호라는 메시지를 출력한다. ...4a2. 비밀번호 입력 UI를 유지한다.
5	5a. 사용자가 취소 버튼을 클릭한다. ...5a1. 메인 메뉴 UI로 돌아간다.
RELATED INFORMATION	
Performance	≤ 2 seconds
Frequency	
<Concurrency>	제한 없음
Due Date	

Use case #9: 얼굴 인식	
GENERAL CHARACTERISTICS	
Summary	나는누구?에서 업로드한 사진과 비슷한 연예인 사진 출력
Scope	나는누구?
Level	User level
Author	김태환
Last Update	2021.10.25
Status	Analysis
Primary Actor	나는누구? user
Preconditions	나는누구?의 사진업로드에서 사진을 업로드 한 상태여야 한다.
Trigger	로그인한 사용자가 사진 업로드 후에 결과 확인을 클릭할 때
Success Post Condition	결과 확인 페이지로 넘어간다.
Failed Post Condition	얼굴 인식 결과를 확인하지 못한다.
MAIN SUCCESS SCENARIO	
Step	Action
S	사용자는 나는누구?에 로그인한다.
1	이 Use case는 사진 업로드를 완료 후 페이지의 결과 확인 버튼을 클릭할 때 시작된다.
2	시스템은 결과 페이지를 출력한다.
3	시스템은 사용자에게 아이돌에 대한 사진과 함께 해당 아이돌과 일치하는 정도를 %로 출력한다.
4	사용자가 정보 확인 버튼을 클릭할시 시스템은 아이돌의 상세 정보를 확인할 수 있게 해준다.
EXTENSION SCENARIOS	
Step	Branching Action
3	3a. 사용자가 이미지를 업로드하지 않는다. ...3a1. 오류 메시지를 출력한다. ...3a2. 이미지 업로드 UI를 유지한다.
RELATED INFORMATION	
Performance	≤ 2 seconds
Frequency	
<Concurrency>	제한 없음
Due Date	

Use case #10: 소셜 공유	
GENERAL CHARACTERISTICS	
Summary	나는누구?에서 자신의 이미지에 대한 결과를 sns로 공유
Scope	나는누구?
Level	User level
Author	김태환
Last Update	2021.10.25
Status	Analysis
Primary Actor	나는누구? user
Preconditions	나는누구?에 등록된 계정으로 로그인 되어있어야 한다. 나는누구?의 사진업로드에서 사진을 업로드 한 상태여야 한다.
Trigger	로그인한 사용자가 사진 업로드 후 소셜 공유를 클릭할 때
Success Post Condition	sns상에 URL 주소를 공유하게 된다.
Failed Post Condition	sns상에 URL 주소를 공유하지 못한다.
MAIN SUCCESS SCENARIO	
Step	Action
S	사용자는 나는누구?에 로그인한다.
1	이 Use case는 아이돌 정보 확인 페이지의 소셜 공유를 클릭하면 실행된다.
2	시스템은 소셜 공유 가능한 어플을 나타낸다.
3	사용자가 시스템이 제공한 어플을 클릭할 시 URL이 복사되며 클릭한 어플로 현재 결과가 전달된다.
4	아이돌 정보확인 UI로 돌아간다.
EXTENSION SCENARIOS	
Step	Branching Action
1	1a. 사용자가 로그인을 하지 않았다. ...1a1. 오류 메시지를 출력한다. ...1a2. 현재 UI(아이돌 정보 확인 UI)를 유지한다.
3	3a. 사용자가 소셜 공유를 하지 않는다. ...3a1. 확인 버튼을 클릭한다. ...3a2. 아이돌 정보 확인 UI 유지한다.
RELATED INFORMATION	
Performance	≤ 2 seconds
Frequency	
<Concurrency>	제한 없음
Due Date	

Use case #11: 얼굴 분석 내용 조회	
GENERAL CHARACTERISTICS	
Summary	나는누구?에서 얼굴을 분석했던 기록들을 조회한다.
Scope	나는누구?
Level	User level
Author	김태환
Last Update	2021.10.25
Status	Analysis
Primary Actor	나는누구? user
Preconditions	나는누구?에 등록된 계정으로 로그인 되어있어야 한다. 나는누구?의 사진업로드에서 사진을 업로드 한 상태여야 한다.
Trigger	로그인한 사용자가 메뉴 화면에서 나의 얼굴분석 기록 버튼을 클릭했을 때
Success Post Condition	분석한 결과를 출력한다.
Failed Post Condition	분석한 결과를 출력하지 못한다.
MAIN SUCCESS SCENARIO	
Step	Action
S	사용자는 나는누구?에 로그인한다.
1	이 Use case는 메인 메뉴 창에서 나의 얼굴분석 기록을 클릭하면 실행된다.
2	시스템은 시간 순서에 따라 분석 결과를 나타낸다.
3	시스템은 사용자가 업로드한 사진과 결과로 나온 연예인 사진, 이름을 출력한다.
4	메인 메뉴 UI로 돌아간다.
RELATED INFORMATION	
Performance	≤ 2 seconds
Frequency	
<Concurrency>	제한 없음
Due Date	



Use case #12: 아이돌 정보 확인	
GENERAL CHARACTERISTICS	
Summary	나는누구?에서 결과로 나온 아이돌에 대한 정보 확인
Scope	나는누구?
Level	User level
Author	김태환
Last Update	2021.10.25
Status	Analysis
Primary Actor	나는누구? user
Preconditions	나는누구?의 사진업로드에서 사진을 업로드 한 상태여야 한다.
Trigger	로그인한 사용자가 사진 업로드 후 결과 확인 페이지 안에 있는 아이돌 정보 확인을 클릭할때
Success Post Condition	아이돌의 사진, 기본적인 생년월일과 그룹 이름을 확인한다.
Failed Post Condition	아이돌의 사진, 기본적인 생년월일과 그룹 이름을 확인하지 못한다.
MAIN SUCCESS SCENARIO	
Step	Action
S	사용자는 나는누구?에 로그인한다.
1	이 Use case는 결과 확인의 아이돌 정보 확인 버튼을 클릭하면 시작된다.
2	시스템은 업로드 된 사진에 대한 정보를 불러온다.
3	시스템은 아이돌의 사진, 기본적인 생년월일과 그룹 이름을 출력한다.
4	결과 확인 UI로 돌아간다.
EXTENSION SCENARIOS	
Step	Branching Action
1	1a. 사용자가 이미지를 업로드하지 않는다. ...1a1. 오류 메시지를 출력한다. ...1a2. 현재 UI(결과확인 UI)를 유지한다.
RELATED INFORMATION	
Performance	≤ 2 seconds
Frequency	
<Concurrency>	제한 없음
Due Date	

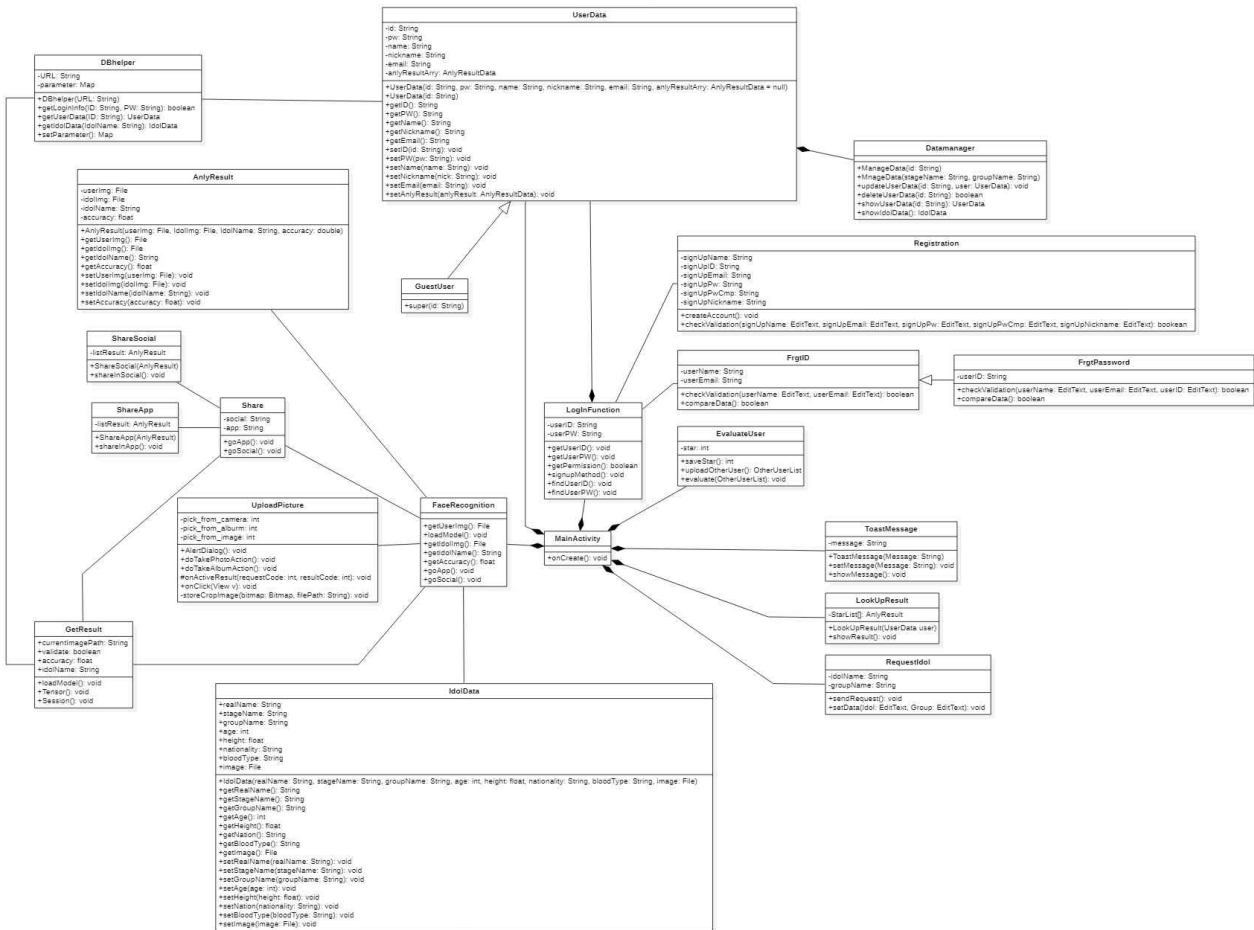
Use case #13: 아이돌 목록 확인	
GENERAL CHARACTERISTICS	
Summary	나는누구?에서 결과로 출력 가능한 아이돌의 목록을 보여준다.
Scope	나는누구?
Level	User level
Author	김태환
Last Update	2021.10.25
Status	Analysis
Primary Actor	나는누구? user
Preconditions	
Trigger	사용자가 메인 메뉴에서 아이돌 목록 확인을 클릭할 때
Success Post Condition	결과로 출력 가능한 아이돌의 목록을 나타낸다.
MAIN SUCCESS SCENARIO	
Step	Action
S	
1	이 Use case는 메인 메뉴의 아이돌 목록 확인 버튼을 클릭하면 실행된다.
2	시스템은 그룹명, 개인별로 아이돌의 그룹명, 활동명을 출력한다.
3	사용자가 그룹명, 아이돌 개인에 대한 선택을 하며 아이돌에 대한 정보를 출력한다.
4	메인 메뉴 UI로 돌아간다.
RELATED INFORMATION	
Performance	≤ 2 seconds
Frequency	
<Concurrency>	제한 없음
Due Date	

Use case #14: 아이돌 추가 요청	
GENERAL CHARACTERISTICS	
Summary	나는누구?에서 사용자가 추가를 원하는 아이돌이 있을 시 요청
Scope	나는누구?
Level	User level
Author	김태환
Last Update	2021.10.25
Status	Analysis
Primary Actor	나는누구? user
Preconditions	
Trigger	사용자가 메인 메뉴에서 아이돌 추가 요청을 클릭할 때
Success Post Condition	사용자가 원하는 아이돌 추가 목록을 시스템이 전달 받는다.
MAIN SUCCESS SCENARIO	
Step	Action
S	
1	이 Use case는 메인 메뉴의 아이돌 추가 요청을 클릭하면 실행된다.
2	시스템은 그룹명과 활동명을 입력할 수 있는 텍스트 입력란과 확인 버튼을 제공한다.
3	사용자가 그룹명과 활동명을 입력한 후 확인 버튼을 클릭할 경우 시스템으로 데이터가 저장된다.
4	메인 메뉴 UI로 돌아간다.
RELATED INFORMATION	
Performance	≤ 2 seconds
Frequency	
<Concurrency>	제한 없음
Due Date	

Use case #15: 유저평가	
GENERAL CHARACTERISTICS	
Summary	나는누구?에서 사용자가 다른 사용자가 공유해놓은 결과를 보고 평가
Scope	나는누구?
Level	User level
Author	김태환
Last Update	2021.10.25
Status	Analysis
Primary Actor	나는누구? user
Preconditions	나는누구?에 등록된 계정으로 로그인 되어있어야 한다.
Trigger	사용자가 아이돌 정보 확인 페이지에서 유저 평가 기능을 클릭할 때
Success Post Condition	사용자가 현재 결과에 대한 평가를 남긴다.
MAIN SUCCESS SCENARIO	
Step	Action
S	사용자는 나는누구?에 로그인한다.
1	이 Use case는 아이돌 정보 확인 페이지에서 유저 평가 기능을 클릭하면 실행된다.
2	시스템은 사용자에게 최소 1개에서 최대 5개로 된 별점을 출력한다.
3	사용자는 원하는 별점을 남기고 시스템이 저장한다.
4	아이돌 정보 확인 UI로 돌아간다.
RELATED INFORMATION	
Performance	≤ 2 seconds
Frequency	
<Concurrency>	제한 없음
Due Date	

## 3. Class diagram

### 3.1 Class diagram



## 3.2 Class description

### 1) UserData

<b>UserData</b>
-id: String -pw: String -name: String -nickname: String -email: String -anlyResultArry: AnlyResultData
+UserData(id: String, pw: String, name: String, nickname: String, email: String) +UserData(id: String) +getID(): String +getPW(): String +getName(): String +getNickname(): String +getEmail(): String +setID(id: String): void +setPW(pw: String): void +setName(name: String): void +setNickname(nick: String): void +setEmail(email: String): void +setAnlyResult(AnlyResult: AnlyResultData): void

Class #1: UserData	
Attribute	
id: String	사용자의 ID가 저장됨
pw: String	사용자의 비밀번호가 저장됨
name: String	사용자의 이름이 저장됨
nickname: String	사용자의 별명이 저장됨
email: String	사용자의 이메일이 저장됨
anlyResultArray: AnlyResultData[]	사용자의 얼굴분석결과 리스트(5개)
Operation	
UserData(id:String, pw:String, name:String, nickname:String, email:String, anlyResultArray:AnlyResultData[]=null)	User객체를 생성하는 생성자(GeneralUser)
UserData(id:String)	User객체를 생성하는 생성자(GuestUser)
getID(): String	id의 getter
getPW(): String	pw의 getter
getName(): String	name의 getter
getNickname(): String	nickname의 getter
getEmail(): String	email의 getter
getAnlyResult(): AnlyResultData[]	anlyResultArray의 getter
setID(id:String): void	id의 setter
setPW(pw:String): void	pw의 setter
setName(name:String): void	name의 setter
setNickname(nick:String): void	nickname의 setter
setEmail(email:String): void	email의 setter
setAnlyResult(AnlyResult:AnlyResultData): void	anlyResultArray의 setter

## 2) GuestUser

GuestUser
+super(id: String)

Class #2: GuestUser	
Attribute	
Operation	
super(id:String)	GuestUser객체를 생성하는 생성자



### 3) AnlyResultData

AnlyResultData
-userImg: File -idollImg: File -idolName: String -accuracy: double
+AnlyResult(userImg: File, IdollImg: File, IdolName: String, accuracy: double) +getUserImg(): File +getIdollImg(): File +getIdolName(): String +getAccuracy(): float +setUserImg(userImg: File): void +setIdollImg(idollImg: File): void +setIdolName(idolName: String): void +setAccuracy(accuracy: float): void

Class #3: AnlyResultData	
Attribute	
userImg: File	사용자의 사진을 저장하는 변수
idollImg: File	아이돌의 사진을 저장하는 변수
IdolName: String	아이돌의 이름을 저장하는 변수
accuracy: float	얼굴분석 정확도를 저장하는 변수
Operation	
AnlyResultData(userImg: File, idolData: IdolData, accuracy: float)	AnlyResultData객체를 생성하는 생성자
getUserImg(): File	userImg의 getter
getIdollImg(): File	idollImg의 getter
getIdolName(): String	idolName의 getter
getAccuracy(): float	accuracy의 getter
setUserImg(userImg: File): void	userImg의 setter
setIdollImg(idollImg: File): void	idollImg의 setter
setIdolName(idolName: String): void	idolName의 setter
setAccuracy(accuracy: float): void	accuracy의 setter

#### 4) IdolData

IdolData
+realName: String +stageName: String +groupName: String +age: int +height: float +nationality: String +bloodType: String +image: File
+IdolData(realName: String, stageName: String, groupName: String, age: int, height: float, nationality: String, bloodType: String, image: File) +getRealName(): String +getStageName(): String +getGroupName(): String +getAge(): int +getHeight(): float +getNation(): String +getBloodType(): String +getImage(): File +setRealName(realName: String): void +setStageName(stageName: String): void +setGroupName(groupName: String): void +setAge(age: int): void +setHeight(height: float): void +setNation(nationality: String): void +setBloodType(bloodType: String): void +setImage(image: File): void

Class #4: IdolData	
Attribute	
realName: String	아이돌의 실명을 저장하는 변수
stageName: String	아이돌의 활동명을 저장하는 변수
groupName: String	아이돌의 그룹이름을 저장하는 변수
age: int	아이돌의 나이를 저장하는 변수
height: float	아이돌의 키를 저장하는 변수
nationality: String	아이돌의 국적을 저장하는 변수
bloodType: String	아이돌의 혈액형을 저장하는 변수
image: File	아이돌의 대표사진을 저장하는 변수
Operation	
IdolData(realName: String, stageName: String, groupName: String, age: int, height: float, nationality: String, bloodType: String, image: File)	IdolData객체를 생성하는 생성자
getrealName(): String	realName의 getter
getstageName(): String	stageName의 getter
getGroupName(): String	groupName의 getter
getAge(): int	age의 getter
getHeight(): float	height의 getter
getNation(): String	nationality의 getter
getBloodType(): String	bloodType의 getter
getImage(): File	image의 getter
setRealName(realName: String): void	realName의 setter
setStageName(stageName: String): void	stageName의 setter
setGroupName(groupName: String): void	groupName의 setter
setAge(age: int): void	age의 setter
setHeight(height: float): void	height의 setter
setNation(nationality: String): void	nationality의 setter
setBloodType(bloodType: String): void	bloodType의 setter
setImage(image: File): void	image의 setter

## 5) DataManager

<b>DataManager</b>
+ManageData(id: String) +MnageData(stageName: String, groupName: String) +updateUserData(id: String, user: UserData): void +deleteUserData(id: String): boolean +showUserData(id: String): UserData +showIdolData(): IdolData

<b>Class #5: DataManager</b>	
Attribute	
Operation	
ManageData(id:String)	DataManager객체를 생성하는 생성자(UserData)
ManageData(stageName:String, groupName:String)	DataManager객체를 생성하는 생성자(IdolData)
updateUserData(id:String, user:UserData): void	회원정보를 수정하는 함수
deleteUserData(id:String): boolean	회원정보를 삭제(탈퇴)하는 함수
showUserData(id:String): UserData	회원정보를 조회하는 함수
showIdolData(): IdolData[]	아이돌리스트를 조회하는 함수

## 6) MainActivity

MainActivity
+onCreate(): void

Class #6: MainActivity	
Attribute	
Operation	
onCreate(): void	UI 기능을 연결해주는 함수

## 7) LogInFunction

LogInFunction
-userID: String -userPW: String
+getUserID(): void +getUserPW(): void +getPermission(): boolean +signupMethod(): void +findUserID(): void +findUserPW(): void

Class #7: LogInFunction	
Attribute	
userID: String	사용자의 ID가 저장됨
userPW: String	사용자의 비밀번호가 저장됨
Operation	
getUserID(): void	EditText를 통해 입력받은 사용자 ID를 멤버 변수에 저장
getUserPW(): void	EditText를 통해 입력받은 사용자 PW를 멤버 변수에 저장
getPermission(): UserData	입력받은 ID, PW가 DB에 존재하는 지 확인하고 해당하는 UserData 클래스 반환
signupMethod(): void	회원가입 버튼을 클릭했을 때 실행
findUserID(): void	아이디 찾기 버튼을 클릭했을 때 실행
findUserPW(): void	비밀번호 찾기 버튼을 클릭했을 때 실행

## 8) Registration

Registration
-signUpName: String -signUpID: String -signUpEmail: String -signUpPw: String -signUpPwCmp: String -signUpNickname: String +createAccount(): void +checkValidation(EditText signUpName, EditText signUpEmail, EditText signUpPw, EditText signUpPwCmp, EditText signUpNickname): boolean

Class #8: Registration	
Attribute	
signUpName: String	입력된 사용자 이름을 저장하는 변수
signUpID: String	입력된 사용자 ID를 저장하는 변수
signUpEmail: String	입력된 사용자 Email을 저장하는 변수
signUpPw: String	입력된 사용자 PW를 저장하는 변수
signUpPwCmp: String	입력된 사용자 PW 재확인을 저장하는 변수
signUpNickname: String	입력된 사용자 별명을 저장하는 변수
Operation	
createAccount(): void	입력된 데이터를 토대로 새로운 Application 사용 권한을 생성
checkValidation(signUpName: EditText, signUpID: EditText, signUpEmail: EditText, signUpPw: EditText, signUpPwCmp: EditText, signUpNickname: EditText): boolean	모든 데이터가 입력되었는지, 올바른 입력형을 가지고 있는지 확인

## 9) FrgtID

FrgtID
-userName: String -userEmail: String
+checkValidation(EditText userName, EditText userEmail): boolean +compareData(): boolean

Class #9: FrgtID	
Attribute	
userName: String	입력된 사용자 이름을 저장하는 변수
userEmail: String	입력된 사용자 Email을 저장하는 변수
Operation	
checkValidation(signUpName: EditText, signUpEmail: EditText): boolean	모든 데이터가 입력되었는지, 올바른 입력형을 가지고 있는지 확인
compareData(): boolean	입력된 데이터를 DB에서 검색한 후 UI에 출력

## 10) FrgtPassword

FrgtID
-userName: String -userEmail: String
+checkValidation(EditText userName, EditText userEmail): boolean +compareData(): boolean

Class #10: FrgtPassword	
Attribute	
userID: String	입력된 사용자 ID를 저장하는 변수
Operation	
checkValidation(signUpName: EditText, signUpEmail: EditText, signUpID: EditText): boolean	모든 데이터가 입력되었는지, 올바른 입력형을 가지고 있는지 확인
compareData(): boolean	입력된 데이터를 DB에서 검색한 후 UI에 출력

## 11) FaceRecognition

FaceRecognition
+getUserImg(): File +loadModel(): void +getIdollImg(): File +getIdolName(): String +getAccuracy(): float +goApp(): void +goSocial(): void

Class #11: FaceRecognition	
Attribute	
Operation	
getUserImg(): File	분석할 이미지를 얻어옴
loadModel(): void	학습된 모델을 불러옴
getIdollImg(): File	아이돌 대표 이미지를 얻어옴
getIdolName(): String	아이돌 이름을 얻어옴
getAccuracy(): float	아이돌 예측값을 얻어옴
goApp(): void	예측된 아이돌 App공유
goSocial(): void	예측된 아이돌 Social공유



## 12) UploadPicture

UploadPicture
-pick_from_camera: int -pick_from_alburm: int -pick_from_image: int
+AlertDialog(): void +doTakePhotoAction(): void +doTakeAlbumAction(): void #onActivityResult(requestCode: int, resultCode: int): void +onClick(View v): void -storeCropImage(bitmap: Bitmap, filePath: String): void

Class #12: UploudPicture	
Attribute	
pick_from_camera: int	카메라촬영으로 이미지 만드는 변수
pick_from_alburm: int	앨범에서 이미지 선택하는 변수
crop_from_image: int	crop해서 이미지 저장하는 변수
Operation	
AlertDialog(): void	'사진불러오기' 버튼 누르면 생성되는 팝업창
doTakePhotoAction(): void	카메라 촬영 후 이미지 가져오기 (임시경로 포함)
doTakeAlbumAction(): void	앨범에서 이미지 가져오기 (앨범호출)
onActivityResult(requestCode: int, resultCode: int): void	액티비티 결과반환 (주요기능에 포함되진 않음)
onClick(View v): void	오버라이딩을 통한 촬영, 선택, 취소
storeCropImage(bitmap: Bitmap, filePath: String): void	새로운 폴더를 생성하여 crop한 image를 저장.

### 13) GetResult

GetResult
+currentImagePath: String +validate: boolean +accuracy: float +idolName: String
+loadModel(): void +Tensor(): void +Session(): void

Class #13: GetResult	
Attribute	
currentImagePath: String	사진의 경로를 가리키는 변수
validate: boolean	사진등록유무를 확인하기 위한 변수, 기본적으로 false를 설정.
accuracy: float	예측의 정확도를 백분율로 나타내는 변수
idolName: String	예측된 아이돌 이름을 저장하는 변수
Operation	
loadModel(): void	학습된 model load하는 생성자
Tensor(): void	Tensorflow model 생성자
Session(): void	Tensorflow Session 생성자

#### 14) LookUpResult

<b>LookUpResult</b>
-StarList[]: AnyResult
+LookUpResult(UserData user)
+showResult(): void

<b>Class #14: LookUpResult</b>	
Attribute	
starList[]: AnyResult	얼굴 분석 기록 중 최근 5회의 정보를 저장하는 배열
Operation	
LookUpResult(user: UserData)	사용자 객체를 입력받아서 멤버 변수 저장
showResult(): void	UI에 user의 분석 기록을 최대 5개까지 출력해서 보여줌

#### 15) EvaluateUser

<b>EvaluateUser</b>
-Star: int
+saveStar(): int
+uploadOtherUser(): OtherUserList
+evaluate(OtherUserList): void

<b>Class #15: EvaluateUser</b>	
Attribute	
star: int	별점을 저장하는 변수
Operation	
saveStar(): int	별점에 대한 결과를 저장하는 함수
uploadOtherUser(): AnyResultData[]	다른 유저의 결과를 불러오는 함수
evaluate(OtherUserList) : void	다른 유저의 결과를 보고 평가를 하는 함수

## 16) Share

Share
-Social: String -App: String
+goApp(): void +goSocial(): void

Class #16: Share	
Attribute	
social: String	공유를 할 플랫폼중 Social을 선택하는 변수
app: String	공유를 할 플랫폼중 App을 선택하는 변수
Operation	
goApp(): void	App에서 공유를 하게 해주는 함수
goSocial(): void	Social에서 공유를 하게 해주는 함수

## 17) ShareApp

ShareApp
-ListResult: AnyResult
+ShareApp(AnyResult) +shareInApp(): void

Class #17: ShareApp	
Attribute	
listResult: AnyResult	불러온 결과를 저장하는 변수
Operation	
ShareApp(rst: anyResult)	ShareApp class 생성자로 얼굴 분석 결과를 매개변수로 사용한다.
shareInApp(): void	App에 현재 URL을 저장하는 함수

## 18) ShareSocial

ShareSocial
-ListResult: AnyResult
+ShareSocial(AnyResult)
+shareInSocial(): void

Class #18: ShareSocial	
Attribute	
listResult: AnyResult	불러온 결과를 저장하는 변수
Operation	
ShareSocial(rst: AnyResult)	ShareSocial class 생성자로 얼굴 분석 결과를 매개변수로 사용한다.
shareInSocial(): void	Social에 현재 URL을 저장하는 함수

## 19) RequestIdol

RequestIdol
-IdolName: String
-GroupName: String
+sendRequest(): void
+setData(Idol: EditText, Group: EditText): void

Class #19: RequestIdol	
Attribute	
idolName: String	추가할 아이돌 이름을 저장하는 변수
groupName: String	추가할 그룹 이름을 저장하는 변수
Operation	
sendRequest(): void	저장된 정보를 전달하는 함수
setData(Idol: EditText, Group: EditText): void	TextEditor로부터 멤버 변수에 저장할 값을 얻어옴

## 20) AnlyResult

AnlyResult
-UserImg: File -Idollmg: File -IdolName: String -accuracy: float
+AnlyResult(userImg: File, Idollmg: File, IdolName: String, accuracy: double) +getUserImg(): File +getIdollmg(): File +getIdolName(): String +getAccuracy(): float +setUserImg(userImg: File): void +setIdollmg(idollmg: File): void +setIdolName(idolName: String): void +setAccuracy(accuracy: float): void

Class #20: AnlyResult	
Attribute	
userImg: File	사용자가 입력한 이미지를 저장하는 변수
idollmg: File	매칭된 아이돌 이미지를 저장하는 변수
idolName: String	매칭된 아이돌 이름을 저장하는 변수
accuracy: double	얼굴 분석 결과 매칭 비율을 저장하는 변수
Operation	
AnlyResult(UserImg: File, Idollmg: File, IdolName: String, accuracy: double)	멤버 변수들을 모두 매개변수로 사용하여 생성자 정의
getUserImg(): File	사용자가 입력한 사진의 getter
getIdollmg(): File	분석된 아이돌 사진의 getter
getIdolName(): String	분석된 아이돌 이름의 getter
getAccuracy(): float	분석 결과 confidence 값의 getter
setUserImg(userImg: File): void	사용자가 입력한 사진의 setter
setIdollmg(idollmg: File): void	분석된 아이돌 사진의 setter
setIdolName(idolName: String): void	분석된 아이돌 이름의 setter
setAccuracy(accuracy: float): void	분석 결과 confidence 값의 setter

## 21) DBHelper

DBHelper
-URL: String -Parameter: Map
+DBHelper(URL: String) +getLoginInfo(ID: String, PW: String): boolean +getUserData(ID: String): UserData +getIdolData(IdolName: String): IdolData +setParameter(): Map

Class #21: DBHelper	
Attribute	
URL: String	DB를 가리키는 URL을 저장하는 변수
parameter: Map	DB에서 검색하기 위한 Map Parameter를 저장하는 변수
Operation	
DBHelper(URL: String)	URL을 정의하면서 생성자 실행
getLoginInfo(ID: String, PW: String): boolean	로그인 정보를 DB에서 확인하고 boolean 값으로 반환
getUserData(ID: String): UserData	UserData 객체를 DB에서 얻어온 정보로 반환
getIdolData(IdolName: String): IdolData	Idoldata 객체를 DB에서 얻어온 정보로 반환
setParameter(): Map	Map Parameter를 setting하는 메소드

### 3. User Interfaces

---

#### 1) Main 화면



앱 시작했을 때 초기화면 UI



## 2) Login 화면



The login interface is enclosed in a black rectangular border. At the top center, the text '나는 누구?' is displayed in a large, blue, rounded font with a slight shadow. Below this, there are two input fields: the first is labeled 'ID' and the second is labeled 'Password', both in a grey sans-serif font. Each label is positioned to the left of a horizontal line representing the input field. Below the input fields, there are two dark blue buttons with white text. The left button is labeled '회원가입' (Sign Up) and the right button is labeled '로그인' (Login). At the bottom of the form, there are two links: '아이디찾기' (Find ID) and '비밀번호찾기' (Find Password), both in a grey sans-serif font.

Login UI

### 3) 회원가입 화면

## 회원가입

ID

Password

Password 확인

이름

Email

Nickname

취소

확인

회원가입 UI이며, ID, PW, 이름, email, 별명을 입력받는다.

#### 4) 아이디 찾기 화면

## 아이디 찾기

이름

Email

취소

확인

ID를 찾는 UI

## 아이디 찾기

ID : user1

취소

확인

ID찾기로 찾은 결과 UI

## 5) 비밀번호 찾기 화면

### 비밀번호 찾기

ID \_\_\_\_\_

이름 \_\_\_\_\_

Email \_\_\_\_\_

**취소** **확인**

비밀번호 찾는 UI

### 비밀번호 재설정

Password \_\_\_\_\_

Password 확인 \_\_\_\_\_

**취소** **확인**

비밀번호 찾기로 실행되는 재설정 UI

## 6) 사진 업로드 화면

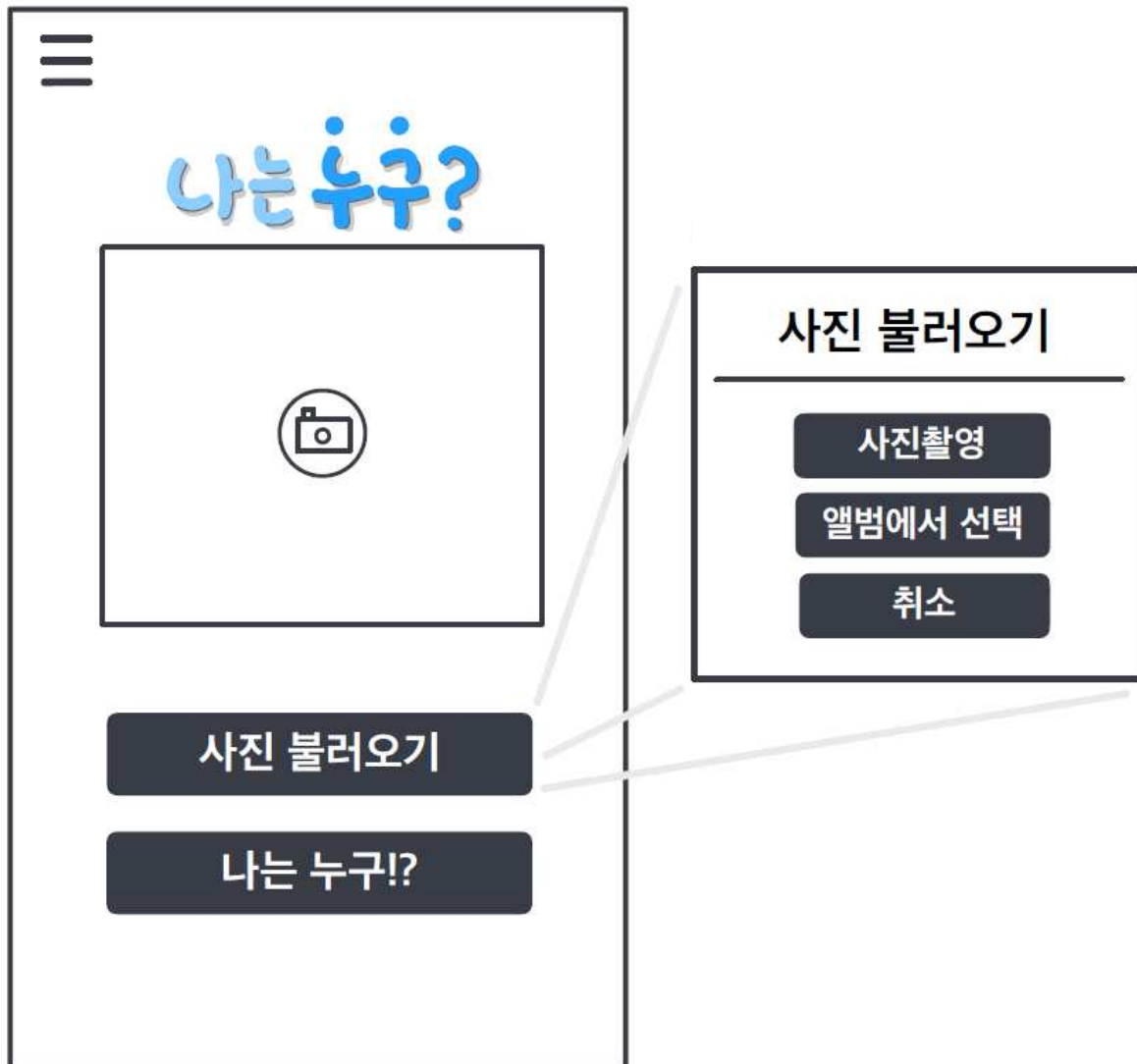
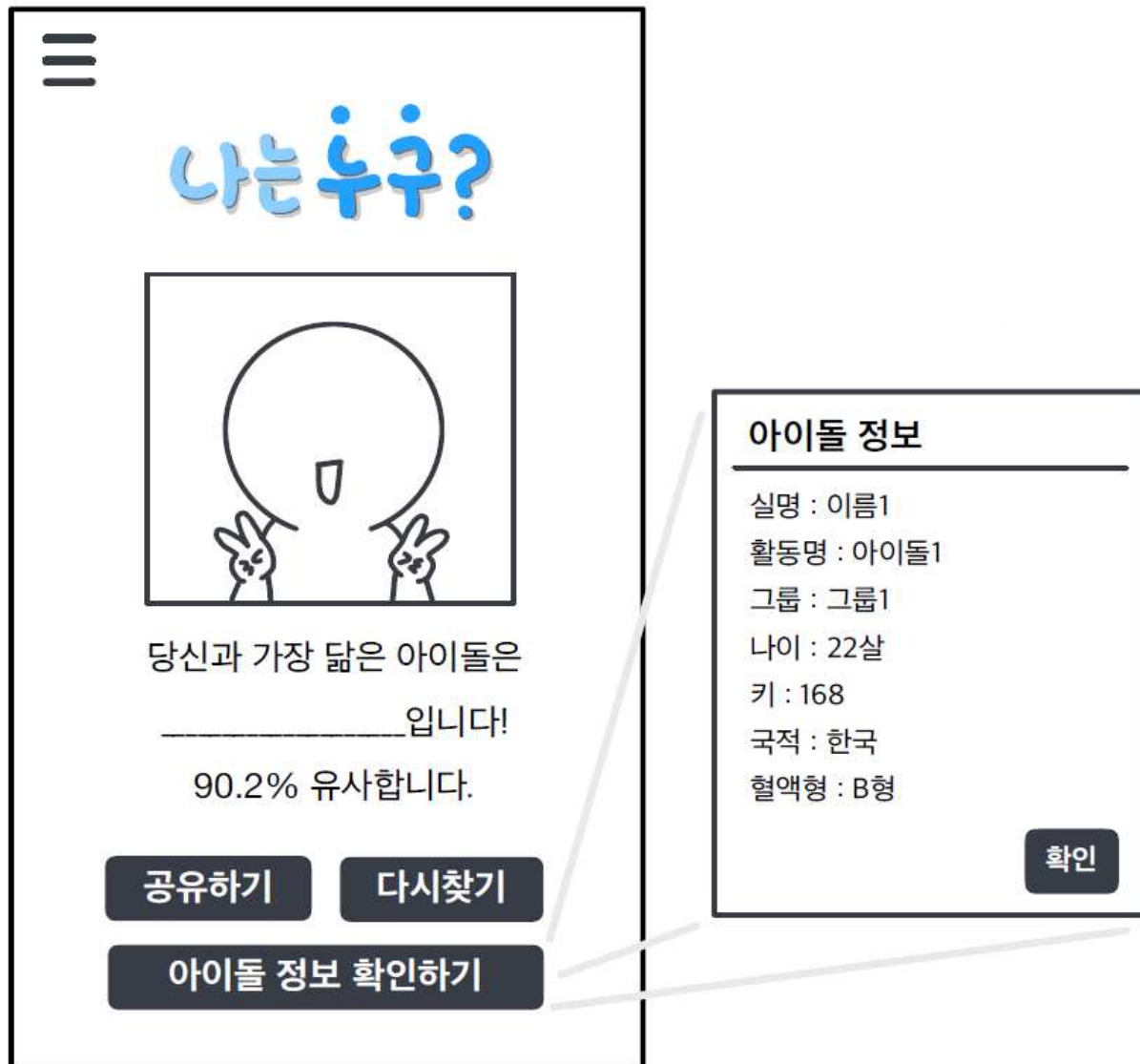



사진 업로드 UI 및 불러올 사진 유형을 선택하는 팝업 UI

## 7) 얼굴분석 결과 화면




얼굴분석 결과 UI 및 아이돌 정보를 확인할 수 있는 팝업 UI

## 8) 메뉴 화면

 로그인 필요합니다.	
회원정보 수정	
얼굴분석하기	
나의 얼굴분석 기록	
분석결과 평가하기	
결과 아이돌 리스트	

로그인을 하지 않은 Guest의 메뉴 UI

 user1	
회원정보 수정	
얼굴분석하기	
나의 얼굴분석 기록	
분석결과 평가하기	
결과 아이돌 리스트	

로그인을 한 정식사용자의 메뉴 UI

## 9) 회원정보 화면

회원 정보

---

ID

user1

Password

●●●●●●

Password 확인

이름

이름1

Email

example0@naver.com

Nickname

별명1

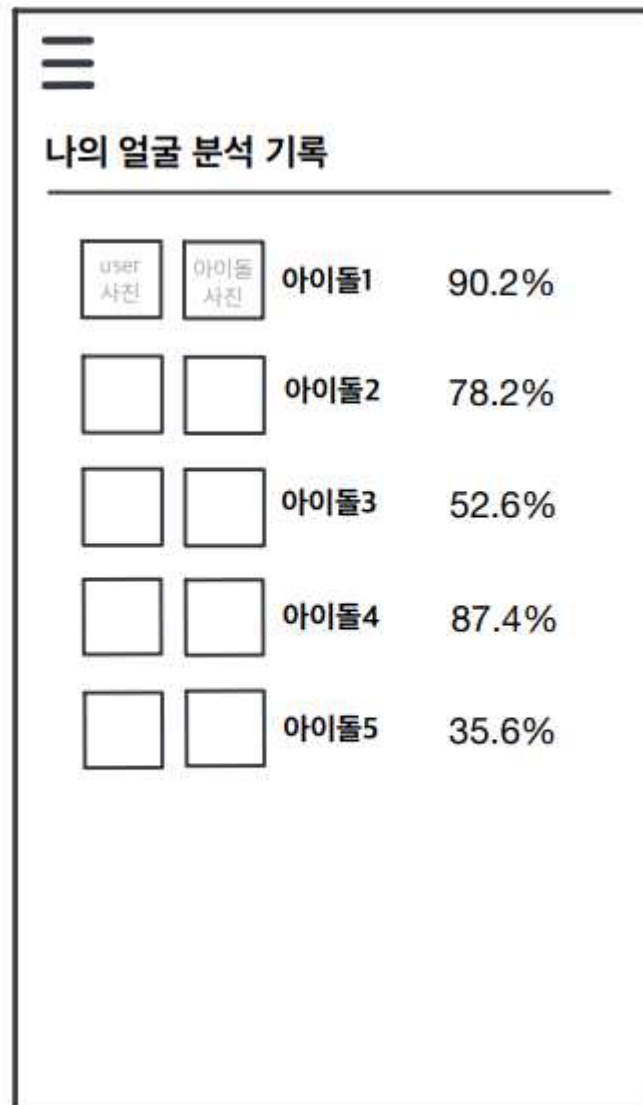
수정하기

탈퇴하기

회원정보를 조회, 수정 및 탈퇴할 수 있는 UI



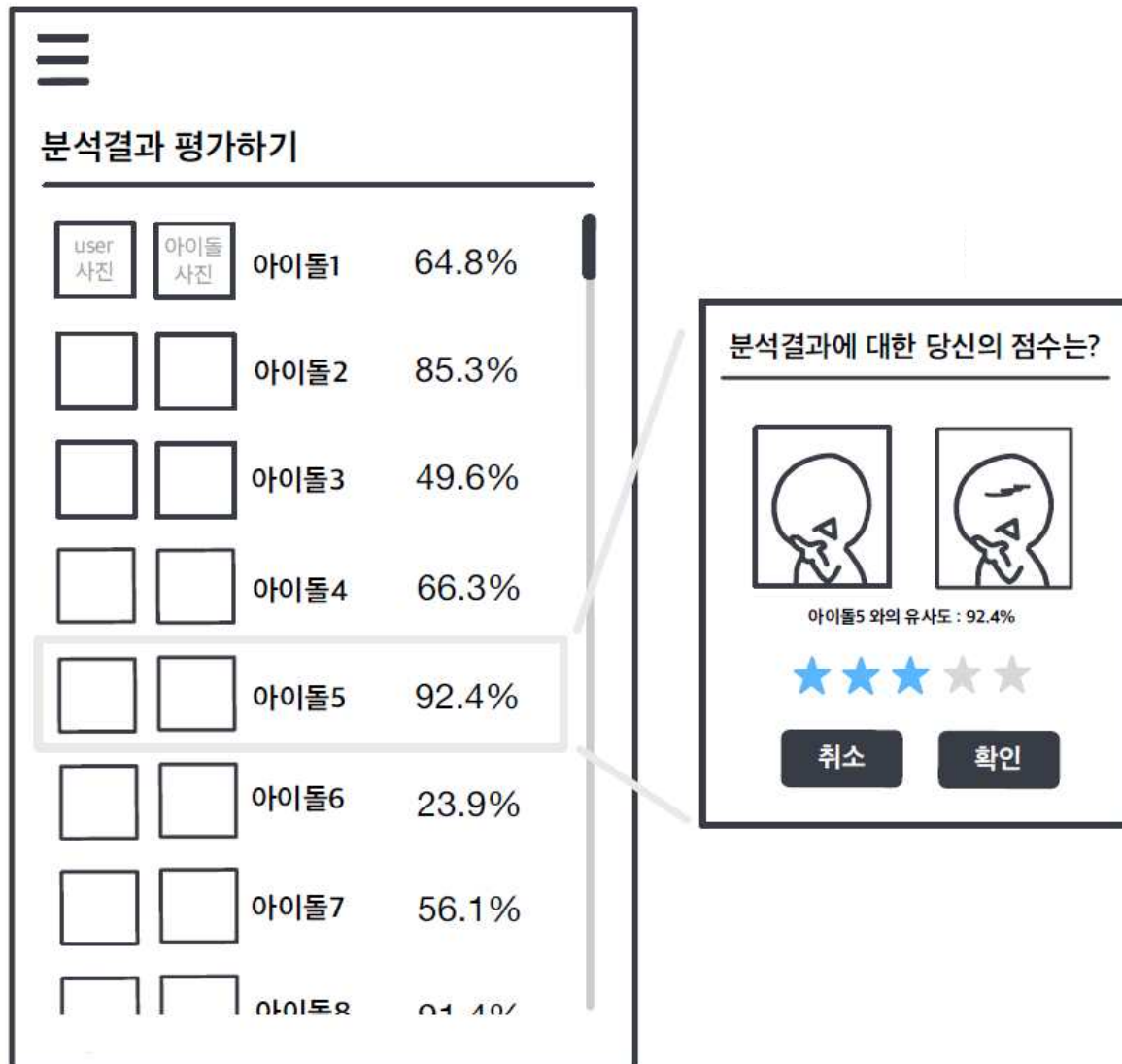
## 10) 얼굴분석 기록 화면



user 사진	아이돌 사진	아이돌	분석률
		아이돌1	90.2%
		아이돌2	78.2%
		아이돌3	52.6%
		아이돌4	87.4%
		아이돌5	35.6%

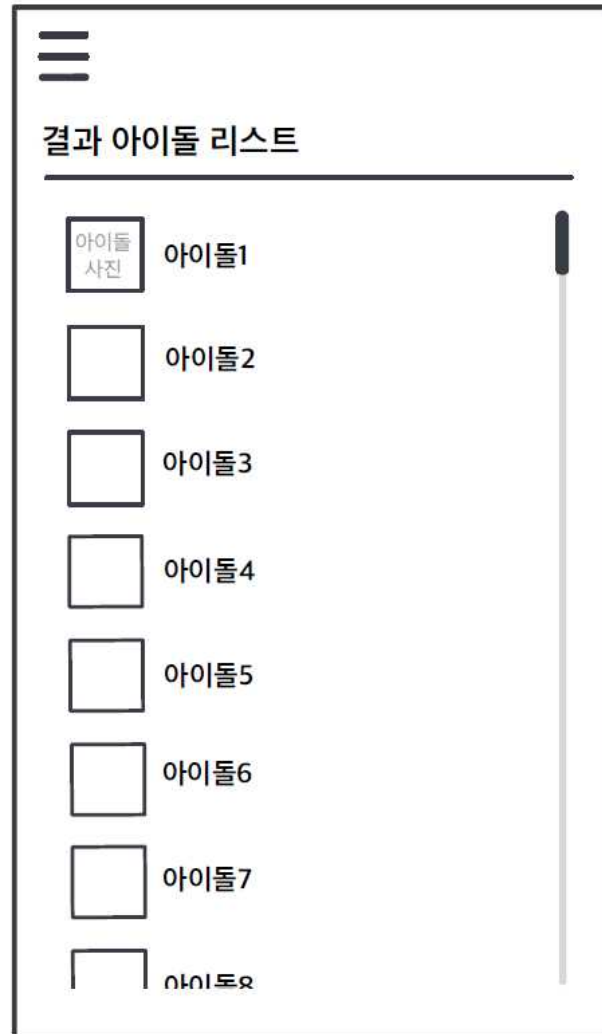
얼굴분석한 기록을 확인할 수 있는 UI

## 11) 다른 유저 분석결과와 평가 화면



다른 유저의 분석결과를 확인하는 UI 및 그에 대한 평점을 남기는 팝업 UI

## 12) 결과 아이돌 리스트 화면



결과로 나올 수 있는 아이돌목록 UI

## 5. Glossary

용어 정리	
가	
게스트	로그인과정을 거치지 않은 사용자
아	
아이돌 추가요청	사용자가 얼굴분석결과로 원하는 아이돌이 나오도록 요청할 수 있는 기능.
자	
정식사용자	로그인과정을 거친 사용자
A	
AnlyResult	게스트, 정식사용자의 이미지를 분석한 결과 Class
D	
DBhelper	데이터베이스에서 UserData, IdolData를 가져오는 Class
I	
IdolData	아이돌의 실명, 활동명, 그룹이름 등의 데이터를 모아둔 객체
S	
ShareApp	App내에서만 공유하는 Class
ShareSocial	카카오톡 등 소셜미디어로 공유하는 Class
U	
UserData	정식사용자의 id, pw, name 등의 데이터를 모아둔 객체
I	
IdolData	아이돌의 실명, 활동명, 그룹이름 등의 데이터를 모아둔 객체

## 6. References

- ▶ 참고 보고서 1: "2. [SE] SDS example 1" , 소프트웨어설계 강의자료
- ▶ 참고 보고서 2: "2. [SE] SDS example 2" , 소프트웨어설계 강의자료
- ▶ 참고 보고서 3: "2. [SE] SDS example 3" , 소프트웨어설계 강의자료
- ▶ 참고 보고서 3: "2. [SE] SDS example 4" , 소프트웨어설계 강의자료
- ▶ 프로젝트 로고 사용: <https://www.flaticon.com/>
- ▶ YOLOV4: <https://arxiv.org/abs/2004.10934/>
- ▶ 안드로이드 스튜디오에서 머신러닝 사용하기: [https://euzl.github.io/hackday\\_1/](https://euzl.github.io/hackday_1/)
- ▶ 클래스 다이어그램 사용법: <https://thinking-jmini.tistory.com/26>
- ▶ 클래스다이어그램사용법:

<https://gmlwjd9405.github.io/2018/07/04/class-diagram.html>