

중간 프로젝트 개발 계획서

[얼굴 인식 로그인 시스템]

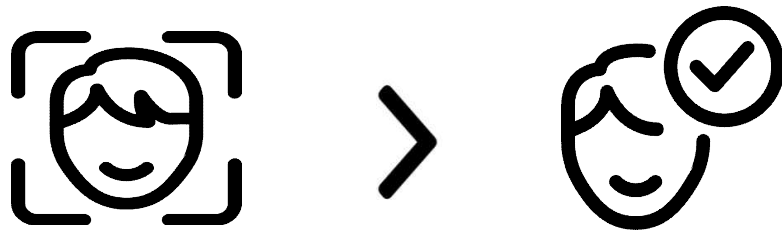
□ 개발 개요

▶ 요약 설명

팀 명	미남미녀조
작품명	OpenCV를 활용한 얼굴인식 로그인, 사용자 정보 관리 시스템

▶ 개발 목적 및 목표

- 별도의 비밀번호를 입력하지 않고 라즈베리파이 카메라(OpenCV)로 사용자의 얼굴을 식별하여 로그인할 수 있는 서비스를 구현하고, 뿐만 아니라 등록된 사용자의 회원정보를 관리자가 관리하는 서비스를 만드는 것을 목표로한다.



▶ 개발 배경 및 동기

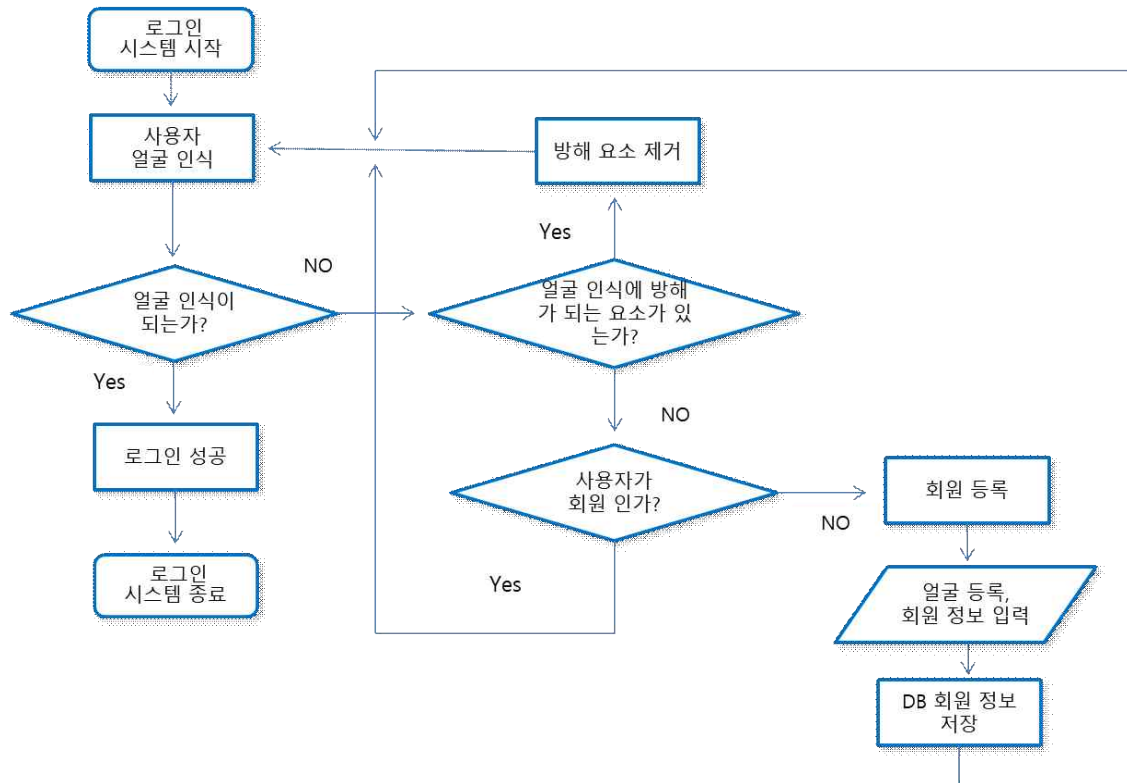
- 요즘에는 결제시스템의 편의성과 보안 문제로 인해 비밀번호의 대안으로 생체인식 기술을 많이 사용하고 있다. 생체 인식 기술은 신체적 특성으로 지문, 홍채, 얼굴, 정맥 등이 있고 행동적 특성으로는 목소리, 서명 등이 있다. 생체인식 기술 중에서도 얼굴 인식 기술의 활용도가 점점 높아지고 있는 추세이다. 얼굴 인식 시스템은 신원 파악뿐 아니라 다양한 마케팅과 인구 통계에도 활용되고 있다. 향후 얼굴 인식 기술은 보안 기능을 보완하여 도입되면 앞으로도 더 많은 분야에 이용 가능할 것이다. 이번 프로젝트에서 구현하고 싶은 얼굴 인식 시스템은 '협력 시스템' 개념을 기반으로 구축한 시스템이다.

< 얼굴 인식 시스템의 두 가지 개념>

협력 시스템	<ul style="list-style-type: none">✓ 얼굴 인식 시스템은 대부분 '협력 시스템' 개념을 기반으로 구축✓ 미리 DB에 검증된 저장 이미지를 인식한 얼굴과 대조하는 방식
비협력 시스템	<ul style="list-style-type: none">✓ 조명과 얼굴 각도, 움직이는 사람 등 제각각인 상황에서 얼굴을 인식하는 시스템 ('Imagus Technology' 라는 호주 기업이 개발)✓ 군중 속 얼굴 인식 시스템

▶ 작품 상세 설명

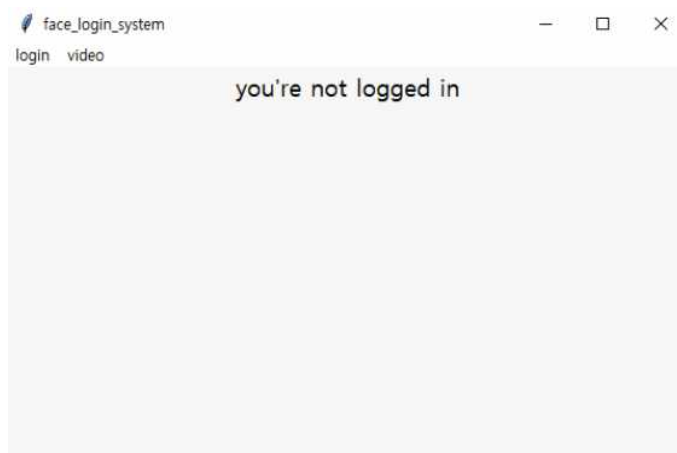
< 로그인 시스템 순서도 >



[로그인 시스템]

- 로그인 시스템을 사용자가 편리하게 사용할 수 있도록 GUI를 사용 (모듈은 tkinter)

① 사용자가 시스템에 로그인하기 위해 라즈베리파이 카메라에 사용자의 얼굴인식 시도



② 사용자 얼굴을 인식하여 MySQL DB에 검증된 저장 이미지가 있는지 조회

③ 카메라에 인식된 얼굴이 저장된 이미지가 일치할 경우



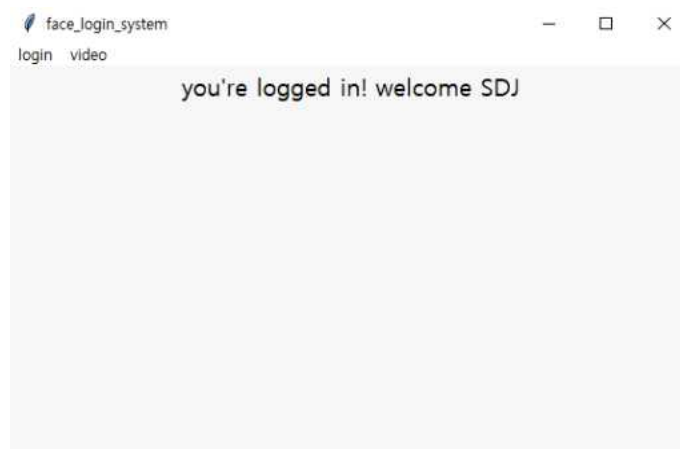
④ 카메라에 인식된 얼굴이 저장된 이미지와 일치하지 않는 경우

4-1) 등록된 사용자의 경우

- 1) 사용자 얼굴이 카메라와 멀거나, 얼굴인식 환경이 어두운 경우
- 2) 마스크나 기타, 얼굴을 인식을 방해하는 물체가 있는 경우

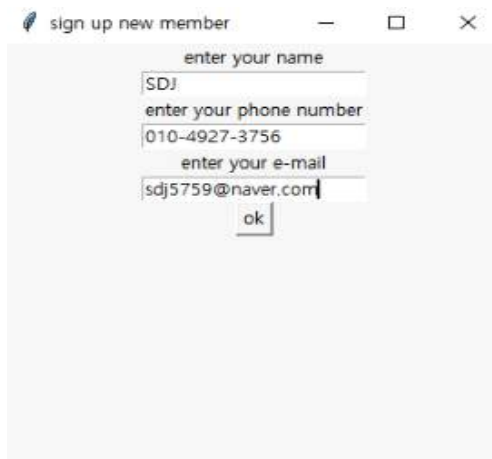


1)과 2) 의 조건을 개선하고 다시 2. 실행하기



4-2) 등록되지 않은 사용자의 경우

- 1) 사용자 정보 입력 (이름, 전화번호, 이메일) → MySQL DB 저장
라즈베리파이 카메라로 얼굴 이미지 저장 ('t' key press)



	usernum	username	userphone	useremail
▶	1	awdawd	wdwdw	wdwdw
	2	SDJ	010-4927-3756	sdj5759@naver.com
•	NULL	NULL	NULL	NULL

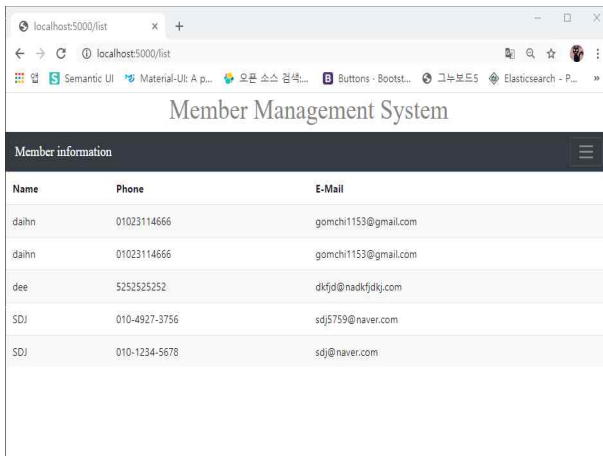
- 2) 사용자 얼굴이 카메라와 멀거나 , 얼굴인식 환경이 어두운 경우
- 3) 마스크나 기타, 얼굴을 인식을 방해하는 물체가 있는 경우

- 1) 2) 3) 조건을 개선하고 다시 2. 실행하기



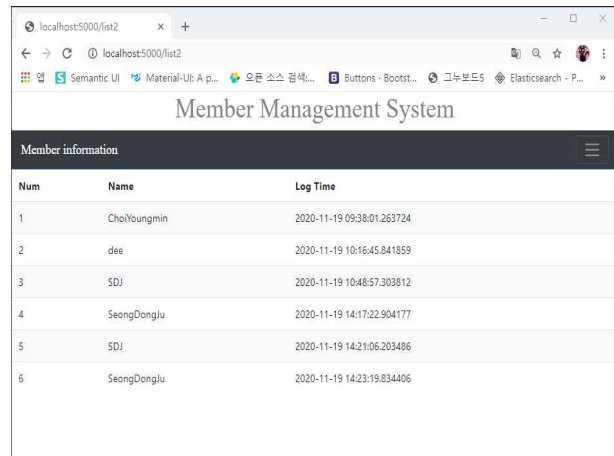
[사용자 정보 관리 시스템]

① 관리자가 사용자 관리 Web 에 접속하여 사용자의 로그인 기록과 정보를 확인, 관리함



Name	Phone	E-Mail
daihn	01023114666	gomchi1153@gmail.com
daihn	01023114666	gomchi1153@gmail.com
dee	5252525252	dkfjd@nadjfkdj.com
SDJ	010-4927-3756	sdj5759@naver.com
SDJ	010-1234-5678	sdj@naver.com

< 사용자 개인 정보 >

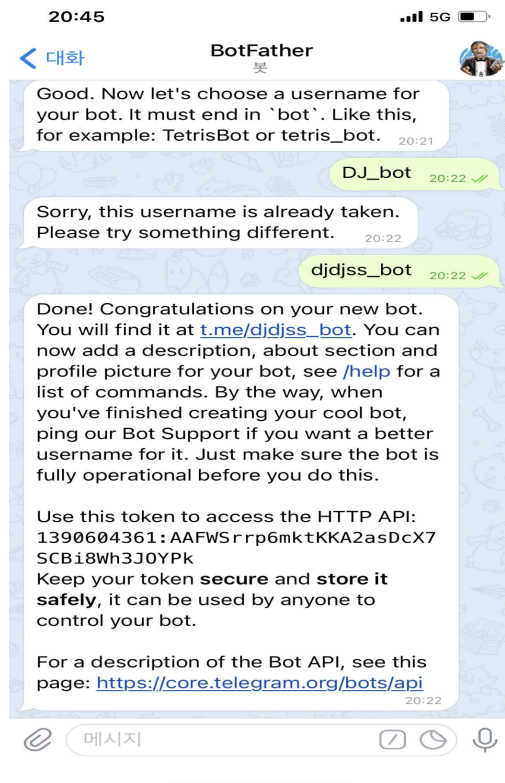


Num	Name	Log Time
1	ChoiYoungmin	2020-11-19 09:38:01.263724
2	dee	2020-11-19 10:16:45.841859
3	SDJ	2020-11-19 10:48:57.303812
4	SeongDonglu	2020-11-19 14:17:22.904177
5	SDJ	2020-11-19 14:21:06.203486
6	SeongDonglu	2020-11-19 14:23:19.834406

< 사용자 이름과 로그 기록 >

② 관리자가 텔레그램 봇을 활용해 사용자 로그인 기록과 정보를 확인

- 관리자의 스마트폰에 '텔레그램'을 설치
- 텔레그램 봇은 유저와 메시지나 명령/응답을 주고받을 수 있는 대화형 앱이다.
- 이번 프로젝트 구현하고 싶은 기능은 인식된 사람의 리스트를 조회하고 정보를 조회 기능이다.



① 관리자가 텔레그램 봇을 실행한다.

② 관리자가 해당되는 요청 번호를 입력하여 DB에 저장된 정보를 불러온다.

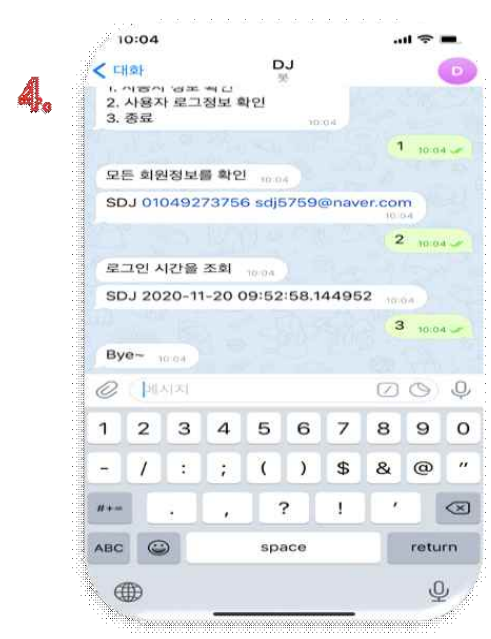
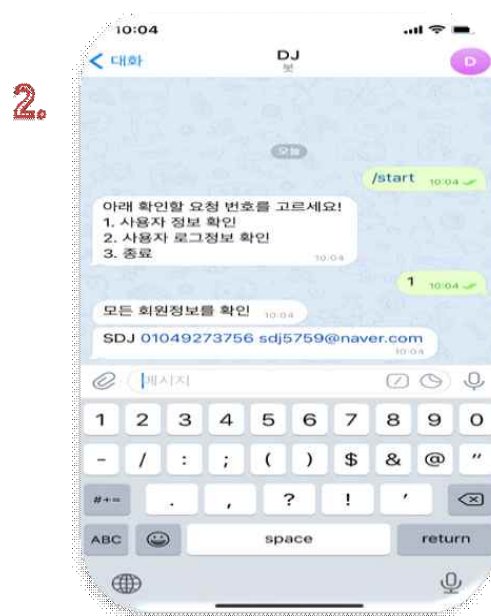
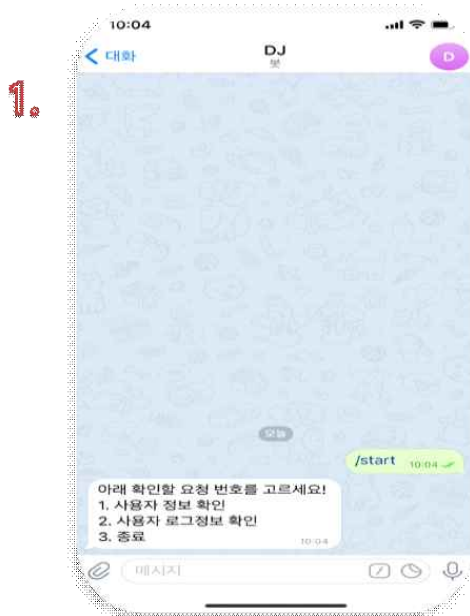
⇒ 1번 DB에 저장된 모든 사용자 정보를 불러온다

③ 관리자가 요청 번호 2번 입력

⇒ 2번 DB에 저장된 사용자의 로그 기록을 불러온다.

④ 관리자가 요청 번호 3번 입력

⇒ 텔레그램 봇 종료



▶ 예상되는 장애요인 및 해결방안

※ 성능 개선

face_recognition package를 사용하기 위해서 모듈 설치

```
$ conda install -c conda-forge dlib
```

→ 얼굴 인식의 오차를 줄이고 인식률을 높이기 위함

```
for face_encoding in face_encodings:
    # See if the face is a match for the known face(s)
    matches = face_recognition.compare_faces(
        known_face_encodings, face_encoding, tolerance=0.45
    )
```

매개변수 : known_face_encoding → 저장된 얼굴 이미지 인코딩 목록
face_encoding → 저장된 얼굴 이미지와 비교할 인식된 얼굴 인코딩
tolerance (허용 오차) → 비교한 얼굴 이미지의 허용 오차 값, 낮을수록 더 엄격함

결과 : tolerance = 0.45 변경 후 얼굴인식 오차 감소 !

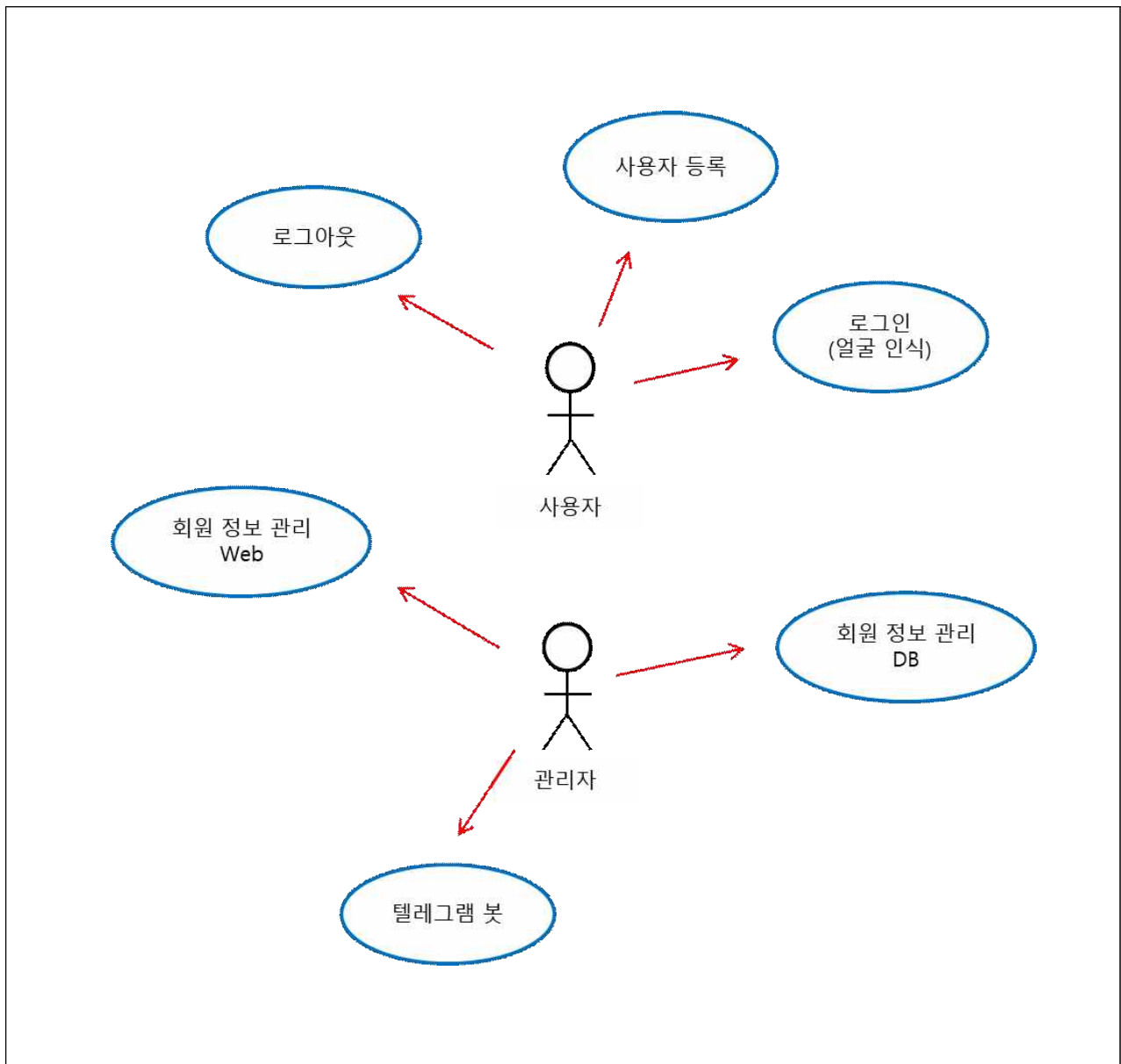
▶ 프로젝트 수행 일정



□ 팀 구성 및 업무 분담

No	구분	성명	팀 내 담당 업무
1	팀장	양예승	- OpenCV 얼굴 인식 알고리즘 - GUI(tkinter) 프로그램 - 사용자 관리 DB
2	팀원	최영민	- OpenCV 얼굴 인식 알고리즘 - 사용자 관리 Web - 사용자 관리 DB
3	팀원	안다인	- OpenCV 얼굴 인식 알고리즘 - 텔레그램 봇 알고리즘
4	팀원	성동주	- OpenCV 얼굴 인식 알고리즘 - 중간 프로젝트 계획서 작성

□ 유즈케이스 다이어그램



□ 유즈케이스 명세서

1. 로그인 시스템

유즈케이스 명	얼굴인식 로그인 시스템
개요	OpenCV 얼굴인식을 활용해 등록된 사용자만 로그인하게 한다.
관련 액터	사용자
우선순위	높음
선행조건	사용자 등록이 된 사용자만 로그인 할 수 있다.
이벤트 흐름	<ol style="list-style-type: none"> 1. 사용자는 로그인 시스템에 로그인하기 위해 라즈베리파이 카메라에 얼굴을 인식하게 한다. 2. 로그인 시스템은 DB에 저장된 회원정보를 조회해 등록된 사용자를 찾는다. 3. 등록된 사용자 정보를 찾아 카메라에 인식된 얼굴을 비교하여 일치하면 로그인 한다. 4. 등록되지 않은 사용자는 얼굴 이미지를 저장하고 사용자 개인정보를 입력한다.
후행조건	등록된 사용자 정보(얼굴이미지, 개인정보)가 바뀌지 않아야 한다.
비 기능적 요구사항	얼굴인식이 가능한 환경(카메라와 사용자의 적당한 거리, 적절한 밝기)이 갖추어져야 한다.

2. 텔레그램 봇 시스템

유즈케이스 명	텔레그램 봇 사용자 정보 관리
개요	관리자가 텔레그램 봇을 사용하여 DB에 저장된 사용자 정보를 관리한다.
관련 액터	관리자
우선순위	높음
선행조건	검증된 사용자 정보가 DB에 저장되어 있어야 한다.
이벤트 흐름	<ol style="list-style-type: none"> 1. 관리자가 사용자 정보를 확인하기 위해 텔레그램 봇을 실행한다. 2. 요청사항을 확인하기 위해 '시작' 입력 3. 텔레그램 봇에 저장된 요청 사항을 요청 번호를 입력해 필요한 사용자 정보를 불러온다. 4. 관리자 '1' 입력 → DB에 저장된 사용자의 개인 정보를 불러온다. 5. 관리자 '2' 입력 → DB에 저장된 사용자의 로그 정보를 불러온다. 6. 관리자 '3' 입력 → 텔레그램 봇을 종료한다.
후행조건	사용자 정보가 바뀌거나 손실되지 않아야 한다.
비 기능적 요구사항	관리자의 스마트폰 보안관리가 매우 중요하다