

얼굴인식 스마트 도어락과 애플리케이션



| | | | | |
|-------------|---|---------|---------------|--|
| 팀명 | LockLock | | | |
| 담당 교수 | 김영웅 교수님 | | | |
| 팀원 | 이름 | 학번 | 연락처 | E-mail |
| | 박병민 | 1491051 | 010-3088-0839 | bm941201@naver.com |
| | 임영선 | 1871228 | 010-4087-4871 | dudtjs1021@hansung.ac.kr |
| | 정은주 | 1871246 | 010-9665-0339 | dmswn0210@hansung.ac.kr |
| | 정지현 | 1871248 | 010-4512-0133 | 809jjh@naver.com |
| Project URL | https://github.com/dudtjs1021ej/faceRecognition_doorlock | | | |

목차

1. 프로젝트 수행 목적

- 1.1 프로젝트 정의
- 1.2 프로젝트 배경
- 1.3 프로젝트 목표

2. 프로젝트 개요

- 2.1 프로젝트 구조
- 2.2 프로젝트 결과물
 - 2.2.1 Hardware 구성
 - 2.2.2 Hardware 기능
 - 2.2.3 Software 흐름도
 - 2.2.4 프로그램 사용법
 - 2.2.5 개발환경
 - 2.2.6 개발 프로그램 설명
 - 2.2.7 장애요인과 해결방법
 - 2.2.8 결과물의 차별성

3. 프로젝트 추진 체계 및 일정

- 3.1 각 조원의 조직도
- 3.2 역할 분담
- 3.3 주 단위의 프로젝트 수행 일정

1. 프로젝트 수행 목적

1.1 프로젝트 정의

Raspberry Pi를 이용하여 얼굴 인식 도어락을 제작 및 도어락과 관련된 다양한 기능을 제공하는 애플리케이션 개발

1.2 프로젝트 배경

저희 팀이 얼굴 인식 스마트 도어락을 진행하게 된 이유는 다음 세가지입니다.

첫째, 광범위한 디지털화로 인한 IoT기술의 발달입니다. 최근 현대사회의 가장 큰 화두는 ‘광범위한 디지털화’입니다. 우리의 일상과 산업 현장 전반에 걸쳐 주변의 모든 것이 인터넷과 결합되고 있습니다. 자동차에 인터넷이 결합되어 자율주행 자동차가 등장하였고, 스마트 미터, 스마트 워치 등 사물인터넷(IoT) 기술을 활용한 다양한 제품들이 계속해서 출시되고 있습니다. 그 중 저희는 스마트 도어락에 주목하였습니다. 현재 시중에서 사용하는 스마트 도어락은 스마트폰 모바일키를 이용한 편리한 개폐시스템으로 많은 사람들이 사용하고 있습니다. 그러나 사용자들이 귀가 시 양손에 짐이 많아 모바일키 이용에 불편함을 느끼는 경우가 생길 수 있으며, 비밀번호를 입력하거나 지문인식을 해야 하는 경우에도 이러한 문제가 해결되기 어려운 시스템입니다.

둘째, 고령화의 가속화와 자연인구 감소의 시작입니다. 현재 우리나라는 고령인구가 연평균 4.4%, OECD 평균에 비해 1.7배 빠른 속도로 증가하고 있습니다. 또한, 2020년도부터 처음으로 인구가 자연 감소하는 데드 크로스가 발생하였습니다. 이렇게 고령화와 디지털화가 함께 가속화되는 현대사회에서 우리가 고려해야 하는 사용자는 바로 고령자입니다. 기억력이 감퇴하고 스마트폰 이용이 서툰 고령자에게 시중의 스마트 도어락은 사용에 어려움이 많을 것입니다.

셋째, 주거 침입 범죄의 증가입니다. 최근 여성 1인 가구를 대상으로 한 주거 침입 범죄가 매년 증가하고 있어 사회적으로 문제가 되고 있습니다. 작년 경찰청 데이터에 따르면 2015년 8223건에 비해 2019년 1만 2887건으로 5년 사이에 무려 59.1%가 증가하였습니다. 1인 가구가 지속적으로 증가하고 코로나 인하여 집에 머무는 시간이 많아진 현대사회에서는 홈 보안이 더욱 중요해졌습니다.

그리하여 저희 팀은 현대사회 요구에 맞는 ‘편리함, 실용성, 보안성’을 모두 갖춘 LockLock만의 얼굴인식 스마트도어락을 만드는 프로젝트를 진행하게 되었습니다.

1.3 프로젝트 목표

저희 프로젝트의 목표는 Raspberry Pi를 이용한 얼굴 인식 스마트 도어락으로 제4차 산업혁명의 핵심기술 중 하나인 IOT기술을 활용하여 더욱 높은 ‘편리성, 실용성, 보안성’을 갖춘 얼굴 인식 스마트 도어락의 개발입니다.

현재 시중에 얼굴인식 도어락이 존재하지만, 일반적으로 상용화되기 어려울 정도의 고가로 거래되고 있습니다. 또한, 사진과 사람을 구분하지 못 해서 보안성에 있어 문제의 소지가 있습니다.

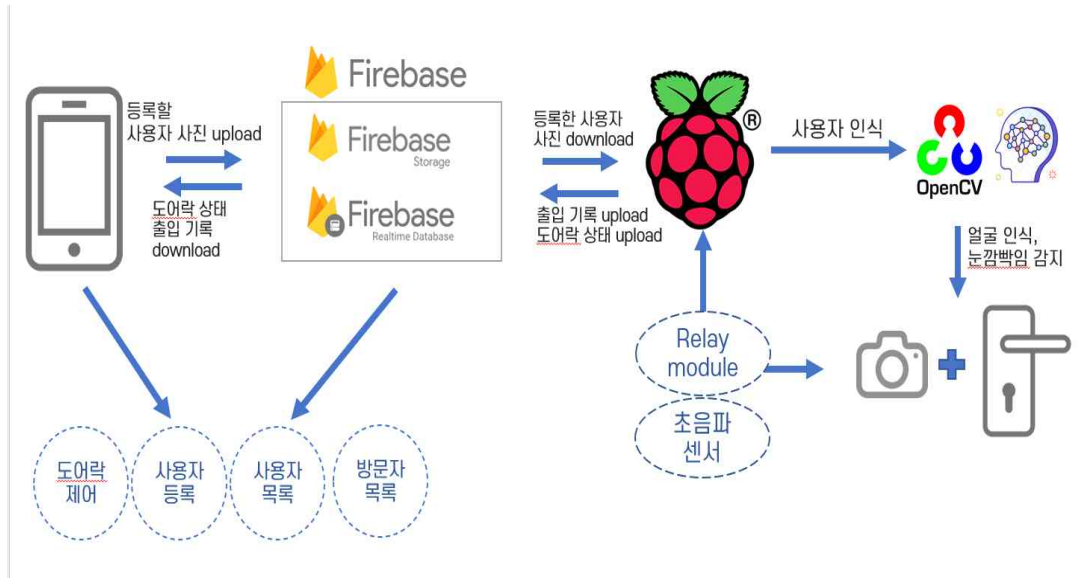
그리하여 저희는 Raspberry Pi를 이용하여 기존의 도어락을 얼굴인식 도어락으로 손쉽게 바꿀 수 있는 시스템을 만듦으로써 가격 접근성을 높일 것입니다.

보안성 향상을 위해서는 얼굴 인식을 통한 도어락 제어기능, 눈 깜박임을 이용한 사람과 사진의 구분기능을 제공할 것입니다.

사용자의 편리성을 위해서 도어락과 관련된 다양한 기능을 사용할 수 있는 관리자용 어플리케이션을 개발할 것입니다. 어플리케이션에서는 지문인식을 통한 도어락 제어기능, 사용자 얼굴등록 및 사용자 목록 확인기능, 출입시도기록 확인기능, 출입시도 실시간 알람 기능을 제공할 것입니다.

2. 프로젝트 개요

2.1 프로젝트 구조



2.2 프로젝트 결과물

2.2.1 Hardware 구성

- Raspberry Pi
- Raspberry Pi 카메라
- Raspberry Pi 초음파센서
- 도어락
- 릴레이모듈
- 휴대폰

2.2.2 Hardware 기능

- Raspberry Pi

얼굴 인식 도어락의 핵심적인 역할을 담당한다. 앱에서 사용자 등록을 요청 시 Firebase에서 등록을 요청한 사용자 사진을 가져와 얼굴을 찾아 정렬 후 얼굴 데이터를 뽑아 저장하고 출입시도가 있을 시 실시간 영상 속 얼굴과 저장된 얼굴 데이터를 비교하여 도어락을 제어한다.

- Raspberry Pi 초음파센서

움직임 여부를 파악하는 역할로서 도어락에 다가오면 Raspberry Pi 카메라를 실행시키고 일정 시간 동안 출입시도가 없다면 Raspberry Pi 카메라를 종료시키고 다시 움직임을 감지한다.

- Raspberry Pi 카메라

실시간 영상을 Raspberry Pi로 전송한다.

- 도어락

Raspberry Pi 카메라를 통한 얼굴인식과 앱을 이용한 지문인식을 통해 도어락을 제어한다.

- 릴레이모듈

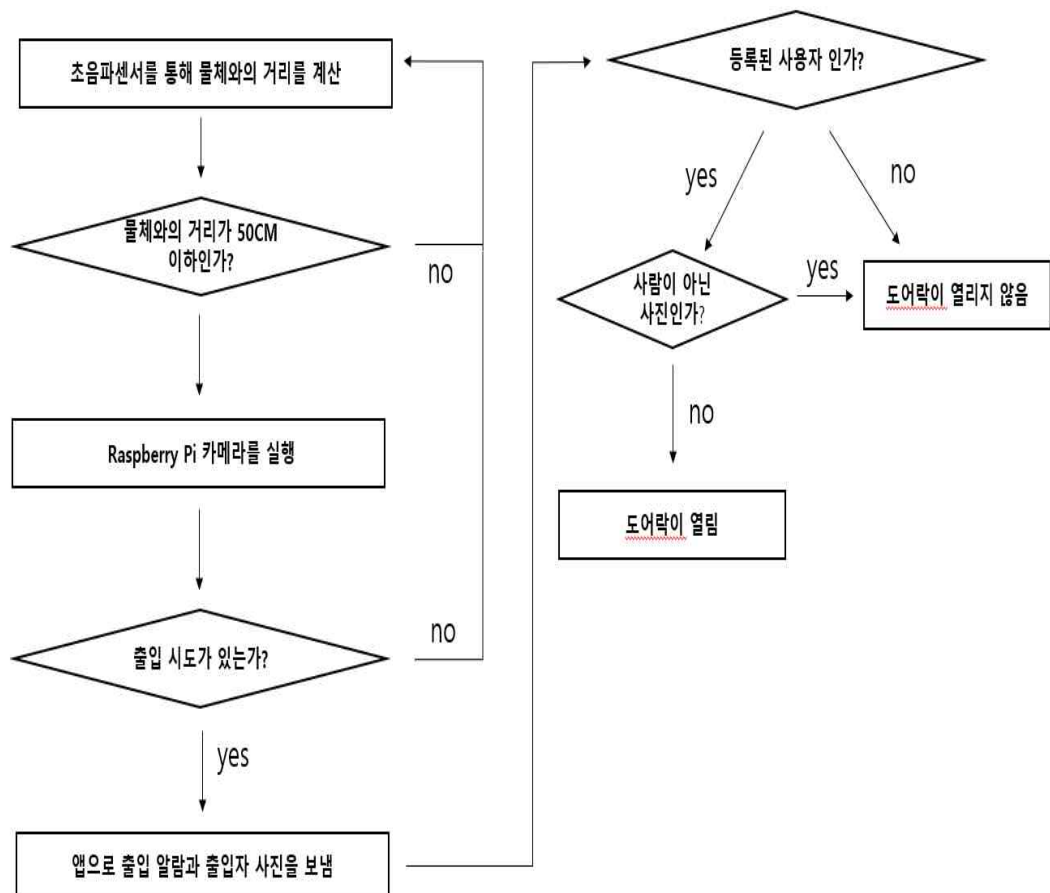
스위치역할로서 Raspberry Pi와 도어락을 연결하여 도어락을 제어할 수 있게 해준다.

- 휴대폰

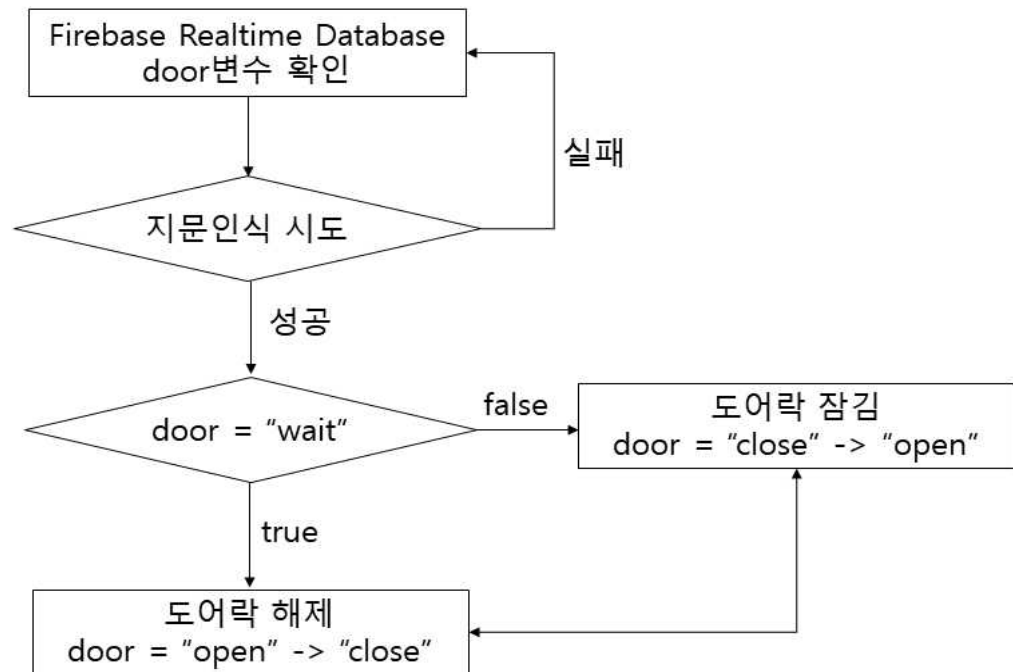
앱을 이용하여 지문인식을 통한 도어락제어, 사용자 등록 및 확인, 출입기록에 대한 알림을 받고 목록을 확인할 수 있다.

2.2.3 Software 흐름도

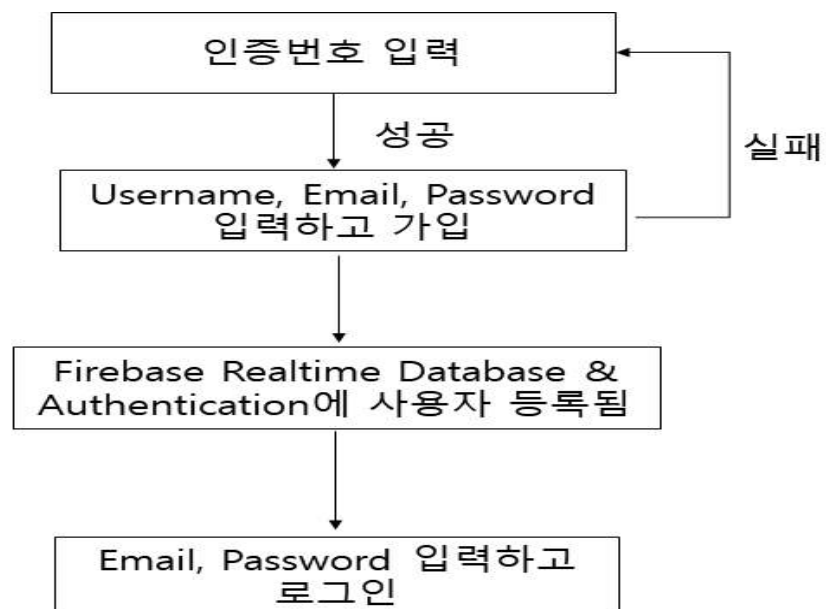
- 얼굴 인식 흐름도



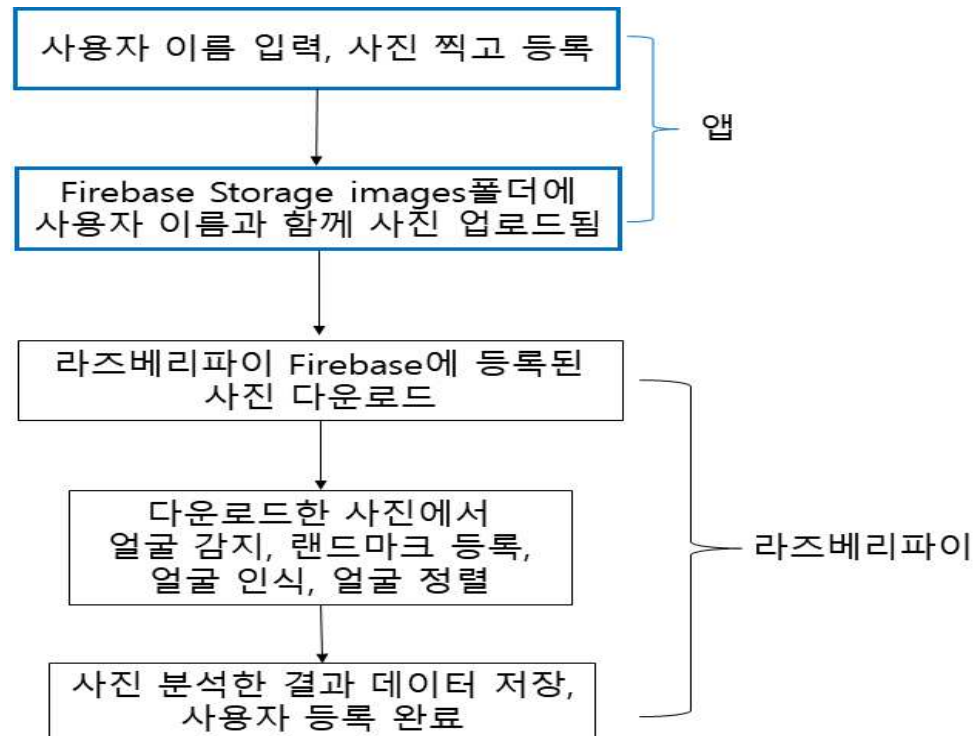
- 지문인식을 통한 도어락 제어



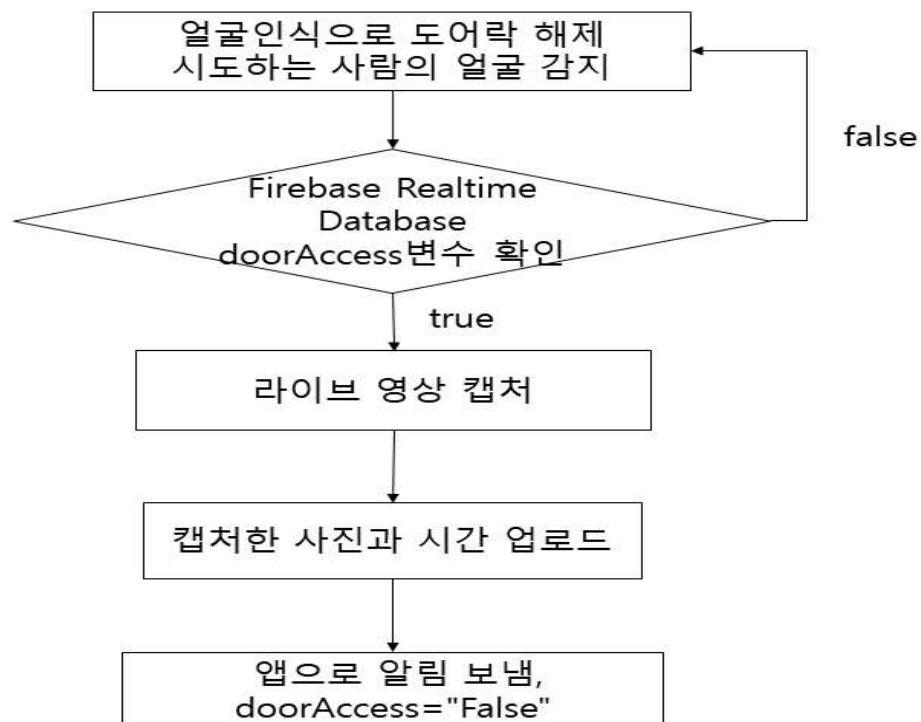
- 앱(회원가입/로그인)



- 사용자 등록



- 출입기록 알림



2.2.4 프로그램 설명

- 회원가입기능



회원가입

Username

Email

Password

가입하기

LockLock

jh@gmail.com

로그인

회원가입

Realtime Database

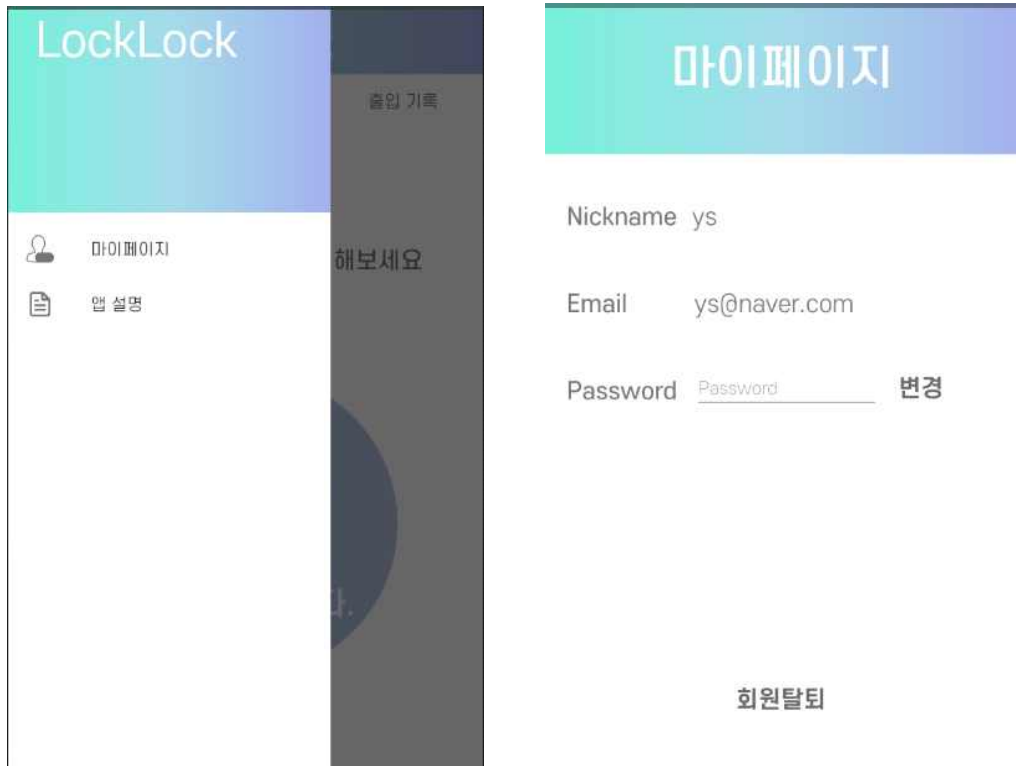
데이터 규칙 백업 사용량

로컬 에뮬레이터 도구 모음을 사용한 프로토타입 제작 및 엔드 투 엔드 테스트에 Firebase 인증 지원 도입 시작하기



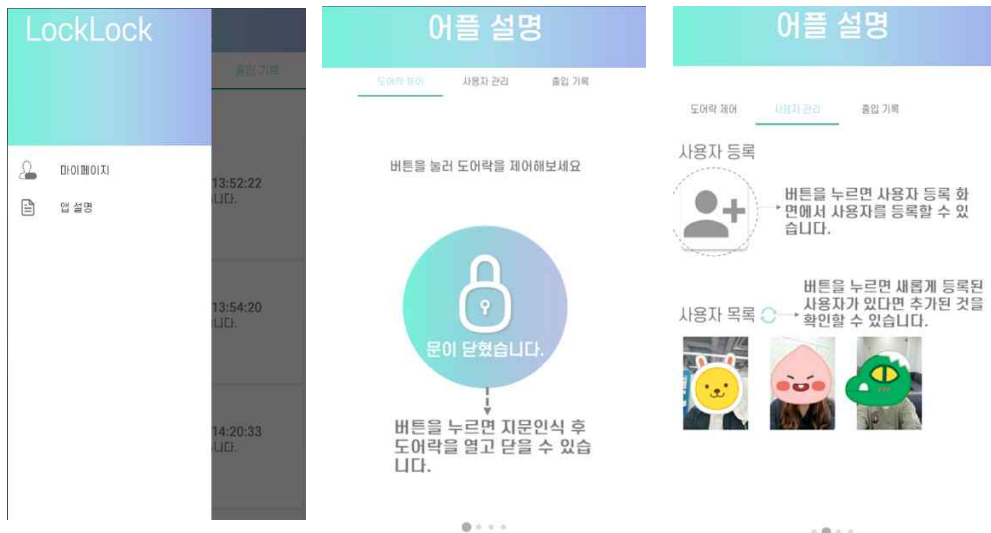
1. 도어락 사용자만 회원 가입할 수 있게 도어락 고유번호를 입력한 후 회원가입을 할 수 있다.
2. Firebase Authentication, 어플리케이션을 통해 회원가입을 한 뒤 로그인한다.
3. Firebase Realtime Database에서 가입한 사용자의 정보를 확인할 수 있다.

- 마이페이지기능



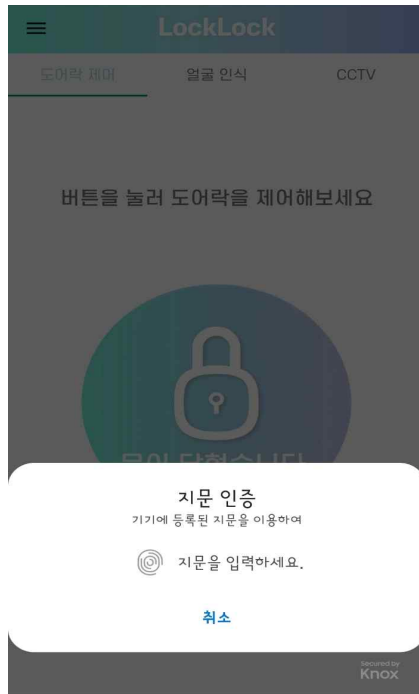
1. 어플리케이션의 마이페이지를 통해 닉네임과 이메일을 확인할 수 있다.
2. 어플리케이션의 마이페이지를 통해 비밀번호를 변경할 수 있고, 회원 탈퇴할 수 있다.

- 앱 설명기능



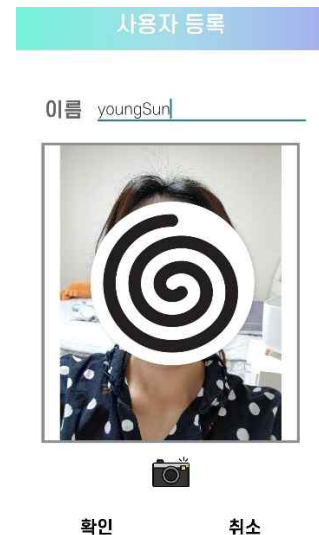
1. 메뉴의 앱 설명 클릭 시, 전반적인 앱의 사용법에 대해 알 수 있다.

- 지문인식을 통한 도어락 제어기능

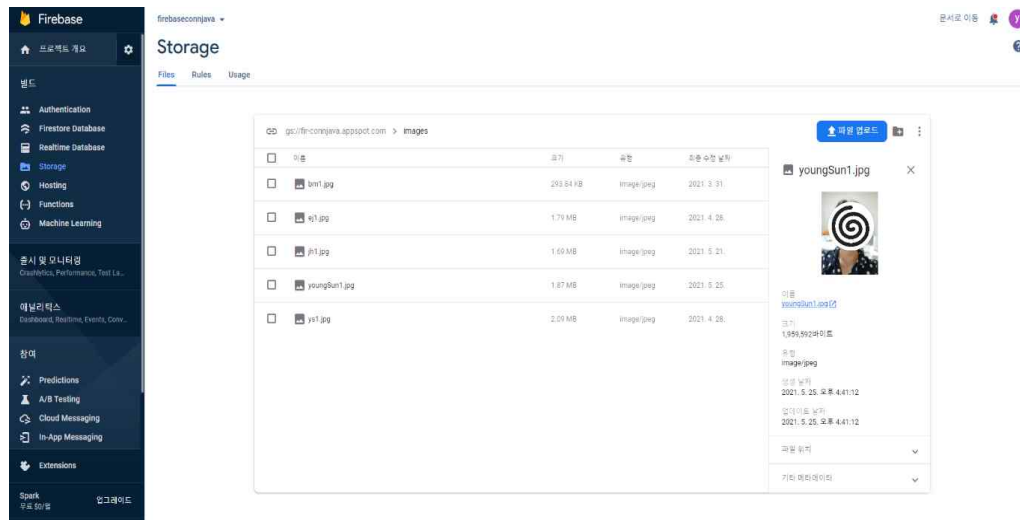


1. 어플리케이션에서 로그인한 뒤 도어락 제어 부문에서 지문인식을 한 뒤에 Firebase Realtime database를 이용하여 도어락을 열거나 닫을 수 있다.

- 사용자 얼굴 등록 및 사용자 목록 확인기능



1. 어플리케이션의 카메라를 사용해 사진을 촬영하여 사용자의 사진과 이름을 등록한다.



2. 촬영된 사진을 Firebase Storage를 이용하여 사용자의 얼굴을 저장한다.

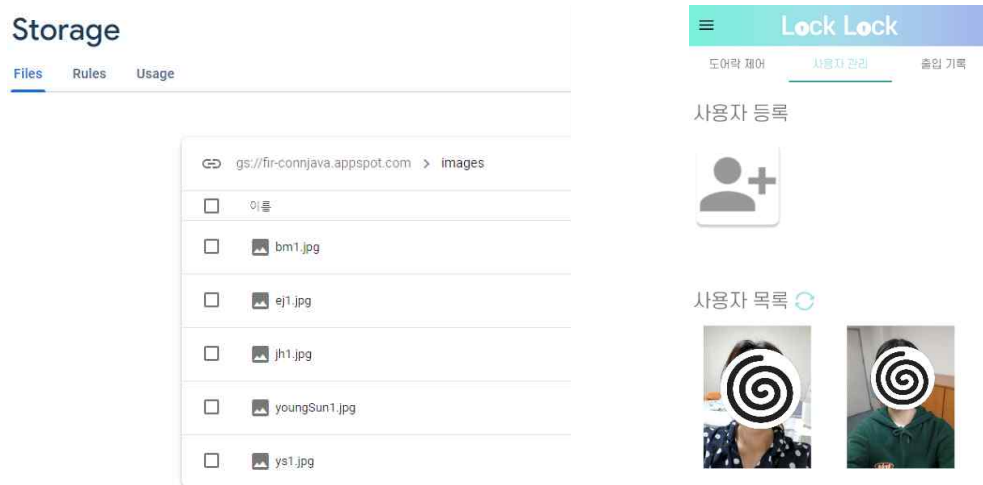
Realtime Database

데이터 규칙 백업 사용량

로컬 에뮬레이터 도구 모음을 사용한 프로토타입 제작 및 엔드 투 엔드 테스트에 Firebase 인증 지원 도입 시작하기

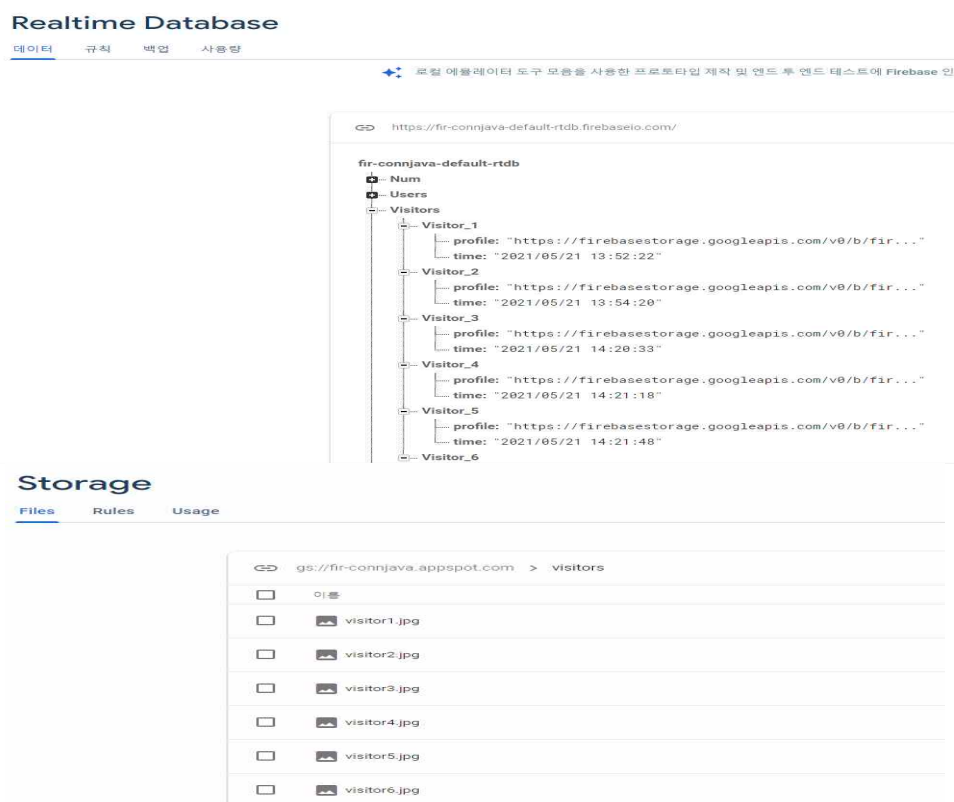


3. 입력한 이름을 Firebase Realtime Database를 이용하여 저장한다.

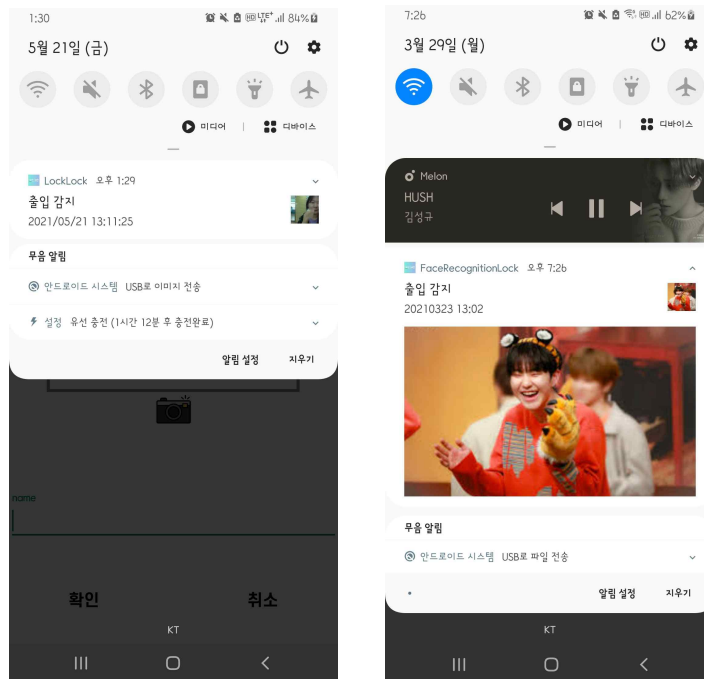


4. Firebase Storage에 저장된 사용자들의 사진을 다운받아 어플리케이션에 사용자 목록을 보여준다.

- 출입시도기록 확인 및 출입 알람기능



1. 얼굴인식을 통해 도어락 해제를 시도한 경우, 인식 유무에 관계없이 출입을 시도한 시간은 firebase realtime database의 Visitors에 저장하고 출입을 시도한 사람의 사진은 firebase storage의 Vistors폴더에 저장하여 관리한다.

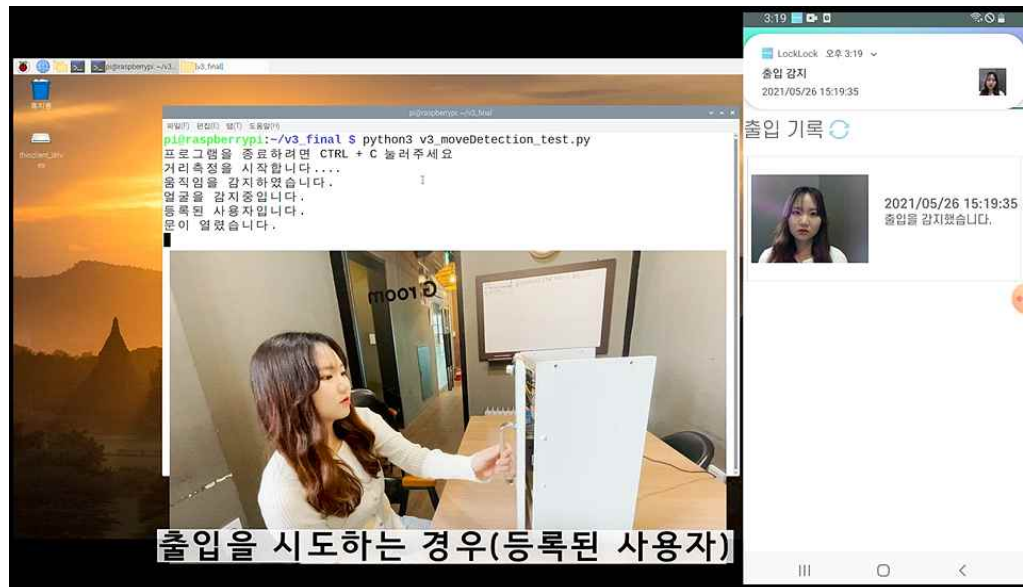


2. 도어락 해제를 시도할 때마다 앱을 통해 누군가 출입을 시도한다는 사실을 그 사람의 사진, 시간을 Notification을 통해 사용자에게 알린다.

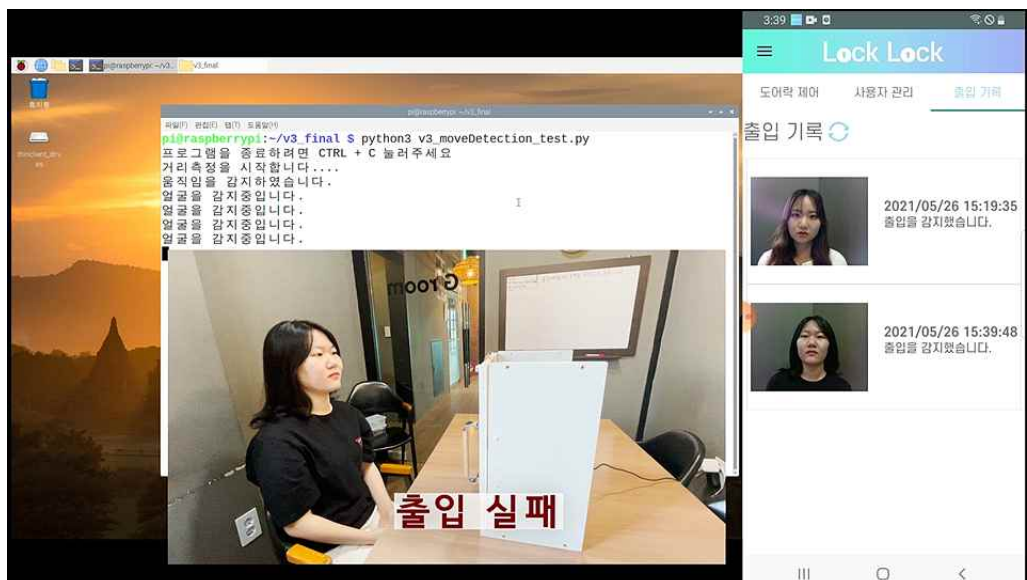


3. 출입을 시도한 사람의 사진과 시간을 앱을 통해 확인할 수 있다.

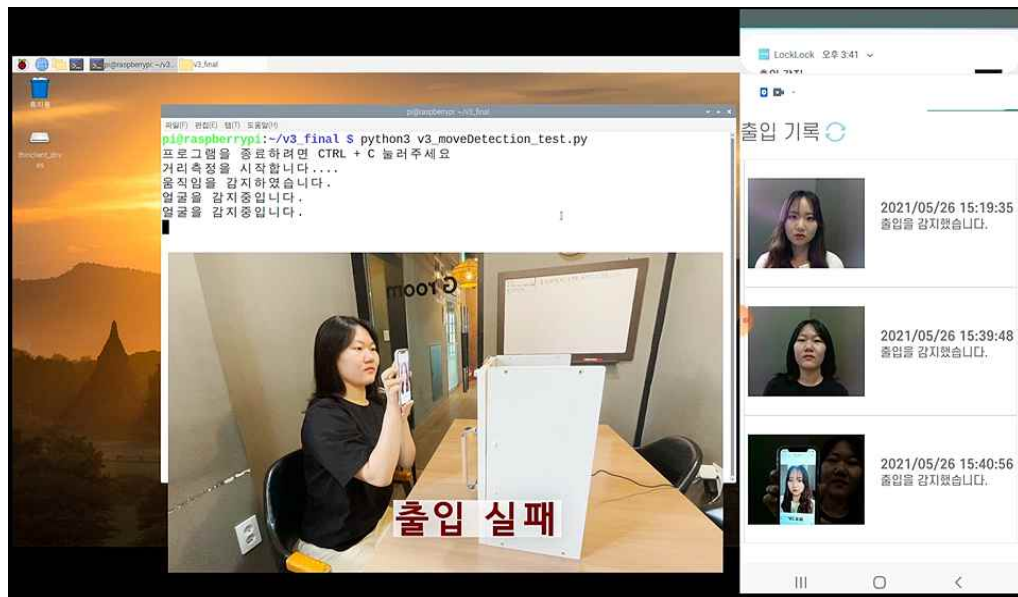
- 얼굴 인식을 통한 도어락 제어



1. 사용자가 도어락에 가까워지면 초음파 센서를 이용해 움직임을 감지하여 카메라를 실행시키고 얼굴을 감지한다.
2. 얼굴 감지를 성공하면 사용자에게 알람을 보내주고 앱에서 출입기록을 확인할 수 있다.
3. 출입을 시도한 사람이 등록된 사람이고 눈 깜박임을 감지한다면 문을 열어준다.



4. 등록된 사용자가 아니라면 움직임 감지와 얼굴 감지를 통해 사용자에게 출입 알람은 보내지만 문을 열어주지는 않는다.



5. 누군가 사진으로 출입을 시도하였을 때 움직임 감지와 얼굴 감지를 통해 사용자에게 출입 알람을 보내지만 문을 열어주지는 않는다.

2.2.5 개발환경

- 하드웨어 및 소프트웨어 개발환경

| | | | | |
|---|---|---|--|---|
|  |  |  |  |  |
| Raspberry Pi 4 | Raspberry Pi 카메라 | 릴레이 모듈 | 초음파 센서 | 도어락 |

| 제품명 | 주요정보 | |
|------------------|---------------------------|---------------------------------------|
| Raspberry Pi 4 | CPU | 28nm 1.5GHz Quad Cortex A72 BCM2711 |
| | RAM | 8GB LPDDR4-3200 SDRAM |
| | Network | 2.4 GHz and 5.0 GHz Gigabit LAN BT5.0 |
| | GPIO | 40pin |
| | Resolution | UHD(3840x2160) 60fps |
| | USB | USB3.0 x2 , USB2.0 x2 |
| | Extend | Display , Camera Port |
| | Power | C-type(15W) |
| Raspberry Pi 카메라 | 크기 | 25 × 20 × 9 mm |
| | 화질 | 1080p |
| | 해상도 | 2592 × 1944 (pixel) |
| 릴레이모듈 | Drive current | 20mA |
| | Control signal | 5V/12V/24V TTL level |
| | Maximum switching voltage | 250VAC 30VDC |
| | Dimension | 50mm + 70mm + 15mm |
| 초음파센서 | 전압 | DC5V |
| | 거리측정거리 | 20~5000mm |
| | 측정 유효 각도 | 15도 이내 |
| | 측정 해상도 | 3mm |
| 도어락 | 제품명 | 해강 솔리티 SRB100 실버 |

| 구분 | 내용 |
|-------------|------------------------|
| 개발환경 | Linux, Windows |
| 개발언어 | Java, Python |
| 개발 Tool | Android Studio, Python |
| 개발 Platform | Firebase |

2.2.7 장애요인과 해결방법

- 얼굴을 인식할 때, 이미지를 흑백으로 만들어 각 픽셀 영역의 검은색 영역과 흰색 영역을 계산해 얼굴을 구분하는 haar cascade를 사용하였다. 하지만 명암 차이에 큰 영향을 받아 밝기에 의해 인식도 차이가 커 어려움을 겪었다.

-> 얼굴에 68개의 특징점을 찾아 얼굴을 구분하는 landmark모델을 사용하여 밝기에 영향을 받지 않도록 모델을 수정했다.

- 어플리케이션의 출입 기록 관리 기능에서 사진은 storage에, 시간은 realtime database에 있어 어떤 사진이 어느 시간에 찍힌지 구분을 못시켜 어려움을 겪었다.

Realtime Database

데이터 규칙 백업 사용량



-> 먼저 얼굴을 감지했을 때 사진을 찍어 storage에 저장시키고 realtime database에서 visitor별로 child를 만들어 각 visitor마다 시간과 storage에 저장된 사진의 토큰을 get해 url을 받아 저장시키게 해 해결하였다.

- 얼굴을 인식하기 위해 카메라가 계속 켜져 있어 라즈베리파이의 발열이 심해지고 성능이 저하됨을 발견했다.

-> 초음파 센서를 설치해 distance 측정을 하며 distance가 50cm이하면 카메라가 켜지고 얼굴인식을 하게하였다. 또한 distance가 50cm이하가 되었지만 출입을 시도하지 않는 경우도 고려해 일정 시간 얼굴을 감지하지 못하면 다시 카메라가 꺼지게 기능을 추가했다.

- 보안을 위하여 얼굴의 유사도 기준을 너무 높게 잡아 모욕률이 올라가는 문제가 발생하였다.

| 기준치 | 시도 횟수 | 성공 횟수 | 실패 횟수 | 기준치 | 시도 횟수 | 성공 횟수 | 실패 횟수 |
|------|-------|-------|-------|------|-------|-------|-------|
| 0.25 | 50 | 0 | 50 | 0.25 | 50 | 0 | 50 |
| 0.3 | 50 | 10 | 40 | 0.3 | 50 | 0 | 50 |
| 0.35 | 50 | 27 | 23 | 0.35 | 50 | 0 | 50 |
| 0.4 | 50 | 50 | 0 | 0.4 | 50 | 0 | 50 |
| 0.45 | 50 | 50 | 0 | 0.45 | 50 | 0 | 50 |
| 0.5 | 50 | 50 | 0 | 0.5 | 50 | 0 | 50 |
| 0.55 | 50 | 50 | 0 | 0.55 | 50 | 8 | 42 |
| 0.6 | 50 | 50 | 0 | 0.6 | 50 | 27 | 23 |

등록된 사용자가

출입을 시도한 경우

등록되지 않은 사용자가

출입을 시도한 경우

- > 기준치별로 다양하게 테스트를 해본 후 모욕률과 기만율이 같은 기준치 범위 중 가장 높은 정확도인 0.4를 기준치로 채택하였다.

- 눈 깜박임 기능에서 특정 기준치에서는 사진 또한 눈을 깜박였다고 판단하는 문제가 발생하였다.

| <사람이 시도한 경우> | | | | <사진이 시도한 경우> | | | |
|--------------|-------|-------|-------|--------------|-------|-------|-------|
| 기준치 | 시도 횟수 | 성공 횟수 | 실패 횟수 | 기준치 | 시도 횟수 | 성공 횟수 | 실패 횟수 |
| 2 | 50 | 25 | 25 | 2 | 50 | 25 | 25 |
| 3 | 50 | 24 | 26 | 3 | 50 | 24 | 26 |
| 4 | 50 | 11 | 39 | 4 | 50 | 0 | 50 |
| 5 | 50 | 3 | 47 | 5 | 50 | 0 | 50 |
| 6 | 50 | 1 | 49 | 6 | 50 | 0 | 50 |

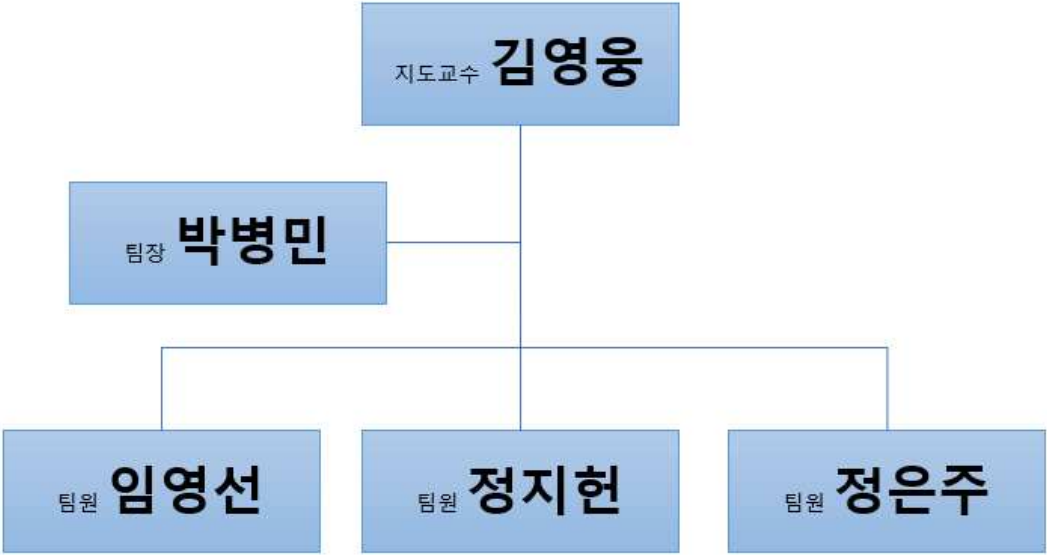
- > 기준치별로 다양하게 테스트를 해본 후 보안성에 있어 사진을 구별해 내는 능력이 중요했기 때문에 사진을 100% 잡아낼 수 있는 기준치인 4를 최종 기준치로 채택하였다.

2.2.8 결과물의 차별성

- 앱에서 도어락을 제어할 수 있도록 하여, 집에 사람이 없어도 앱을 통해 문을 열고 닫을 수 있게 하여 사용자의 편리성을 높였다.
- 앱을 통해 도어락을 제어할 때 지문인식을 이용해서 휴대폰의 주인이 아닌 다른 사람이 시도했을 때 실패하도록 하여 보안성을 높였다.
- 카메라로 출입을 시도한 사람의 사진을 촬영하여 휴대폰의 푸시알림으로 전송하고 출입을 시도한 사람의 사진을 관리자가 확인할 수 있도록 하여 사용자의 편리성과 보안성을 높였다.
- 기존 발표된 스마트 도어락은 가격이 40만원 정도로 고가인 경우가 많은데, 우리가 만든 얼굴 인식 도어락은 기존 도어락에 라즈베리파이 도어락을 추가한 것이므로 저렴하고, 기존에 집에 있는 도어락에 라즈베리파이만 연결하면 사용할 수 있다.
- 미등록 사용자가 등록된 사용자의 사진을 이용하여 출입을 시도하는 경우를 방지하기 위해 눈깜박임을 감지하여 사람과 사진을 구분할 수 있게 하여 보안성을 높였다.
- 사용자를 등록할 때 도어락과 연결된 카메라가 아닌 앱을 이용해 얼굴을 등록하게 하여 사용자의 편리성을 높였다.
- 앱에서 사용자를 등록할 때 보안성을 높이기 위해 기존 갤러리에 사진을 선택하여 업로드하는 방식이 아닌, 카메라로 촬영을 하여 바로 업로드하는 방식을 사용하였다.
- 초음파센서를 이용하여 움직임 감지 기능을 만듦으로써 라즈베리파이가 계속해서 카메라를 실행시키지 않게 하여 라즈베리파이의 발열로 인한 성능 저하를 줄였다.
- 얼굴데이터를 뺄 때 사진을 정면으로 정렬한 후에 데이터를 뺄게 만들어 얼굴 인식의 정확도를 높였고 얼굴 인식 시 랜드마크 방식을 사용하여 사용자가 안경을 쓰고 있거나 불빛이 약해 어두운 상황에서도 정확히 얼굴 인식을 할 수 있다.

3. 프로젝트 추진 체계 및 일정

3.1 각 조원의 조직도



3.2 역할 분담

| | |
|-----|--|
| 박병민 | 얼굴인식코드 개발, 눈 깜빡임 감지 기능 개발, 초음파센서를 이용한 움직임 감지 기능 개발, 최종보고서 작성, 영상편집 및 발표 |
| 임영선 | 앱 디자인, 앱 도어락제어 기능 개발, 앱 사용자 목록 기능 개발, 앱 출입 기록 기능 개발, 주간보고서 작성, ppt 제작 |
| 정지현 | 얼굴사진 정렬 기능 개발, 앱 로그인 및 기능설명 개발, 앱 사용자 목록 기능 개발, 앱 출입 시도 알림 기능 개발, 주간보고서 작성, 최종보고서 작성 |
| 정은주 | 초음파센서를 이용한 움직임 감지 기능 개발, 앱 사용자 등록 기능 개발, 주간 보고서 작성, 얼굴인식 데이터 분석 |

3.3 주 단위의 프로젝트 수행 일정

| 주차 | 수행 일정 |
|------|--|
| 2주차 | 주제 회의, 관련 자료 수집, 필요한 장비 주문 |
| 3주차 | 라즈베리파이 프로그래밍 환경구축, haarcascade기반 얼굴 인식 코드 코딩 및 테스트 |
| 4주차 | 소켓통신 서버, 클라이언트 구축 및 테스트, haarcascade기반 얼굴 인식 코드 코딩 및 얼굴 인식 정확도 높이기, 프로젝트 타켓 결정 |
| 5주차 | 랜드마크 코드분석, 도어락 주문 및 라즈베리파이와 연결 후 제어 확인, 파이어베이스 연결 및 환경구축 |
| 6주차 | 도어락 연결 및 도어락 어플제어기능 코딩, 파이어베이스 연결 및 서버 구축, 랜드마크 얼굴 인식 코드 코딩 |
| 7주차 | 어플 디자인, 사용자 등록 기능 구현, 랜드마크 얼굴 인식 코드 정확도 개선 |
| 8주차 | 얼굴 인식 코드 속도 개선, 도어락 보완, 어플 사진경로 수정 |
| 9주차 | 눈 깜박임 코드 자료수집 및 코딩, 마스크 탐지코드 자료수집 및 코딩 |
| 10주차 | 출입 기록 관리 기능, 마스크 감지 기능 구현, 얼굴 등록 코드 통합 |
| 11주차 | 초음파 및 적외선 센서 코드 분석 및 실시간 얼굴인식 코드와 통합, 출입 기록 관리 기능 및 등록된 사용자 불러오기 구현, 도어락 설치 |
| 12주차 | 초음파 코드와 얼굴인식 코드 통합 및 오류 수정, 등록된 사용자 불러오는 코드 완성, 사용자 목록 새로고침 기능 추가 및 앱 디자인 |
| 13주차 | 최종 통합코드 작성, 앱 아이콘 및 인트로 화면, 마이페이지 추가, 앱 디자인 수정 |
| 14주차 | 데모 영상 촬영, 보고서 작성 및 수정, 회원가입 시 인증번호 입력 기능 추가 |