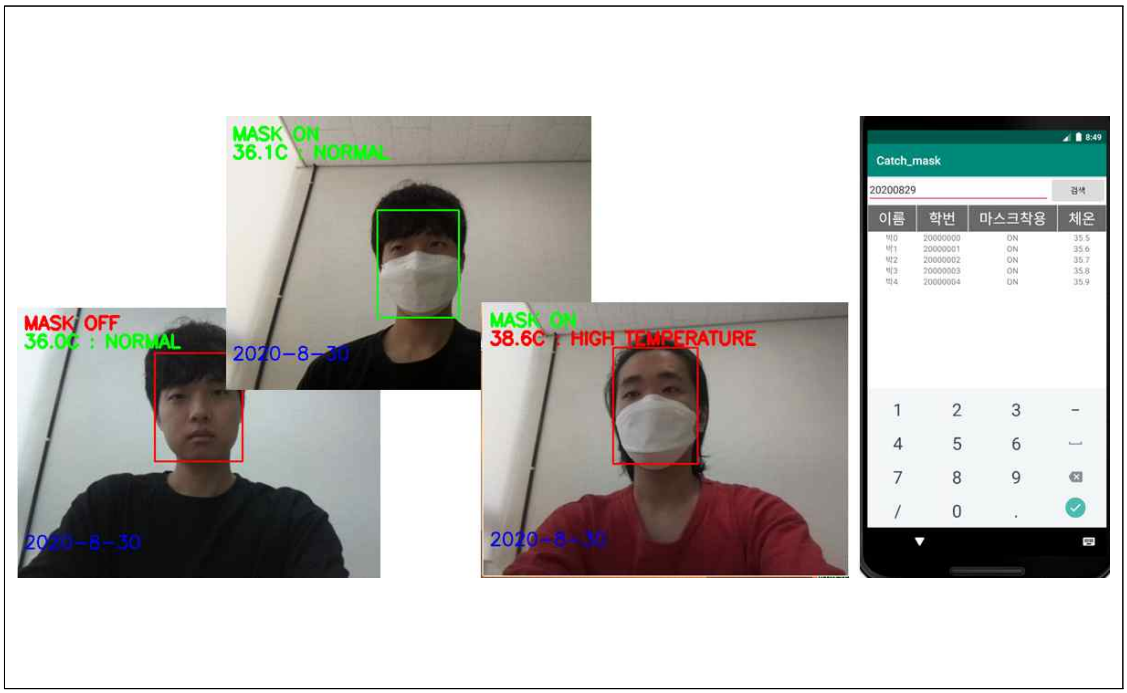


출품 분야 : 임베디드 SW, 윈도우, 모바일SW
기타( )

1. 작품명

< Catch_Corona >

2. 팀 이름

CRA

3. 팀 구성

팀장 - 컴퓨터학부 20152375 박동희
팀원 - 컴퓨터학부 20152379 박종대
전자정보공학부 it융합학과 20170606 안병준

4. 담당교수님

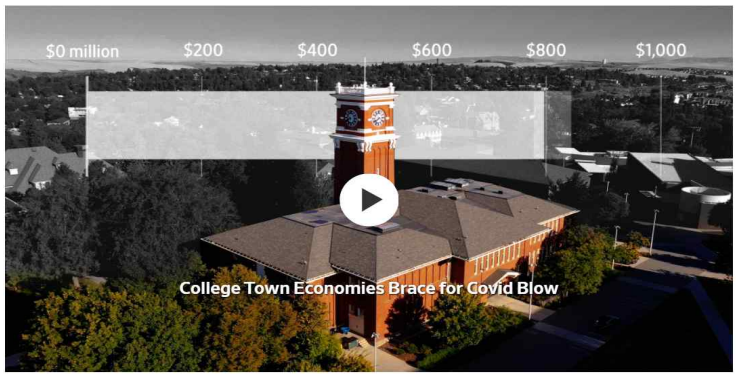
박동주 교수님

5. 기획의도

코로나-19 바이러스 확산에 따라 일상 생활에 전례없는 변화가 찾아 왔다. 대학교 안에서도 마찬가지다. 실제 외국 대학에서는 집단 감염이 발생하기도 했다. ([그림 1])

백신이 없는 상황속에서 예방책은 '**불필요한 접촉**'을 최대한 줄이는 것이다. 이를 위해 숭실대학교는 비대면 수업을 진행하고 있다. 하지만 성적과 직결되는 시험 등은 대면 진행이 불가피하고 평소 학교 건물에 출입하는 인원이 전혀 없는 것도 아니기 때문에 건물 출입 인원 관리/통제가 필수적이다. 이를 위해서는 또 다른 여러 인원이 투입되어야 하기 때문에 시간적, 공간적 제약이 발생할 뿐만 아니라 인원간의 불필요한 접촉이 발생할 수 밖에 없다.

편리한 건물 출입 인원 관리/통제 프로그램이 'Catch_Corona' 이다.



The coronavirus has pushed nearly half of U.S. colleges and universities into some degree of remote learning, a change that's sending shock waves through small college town economies. WSJ's Carlos Waters explains.

[그림 1]

6. 작품 설명

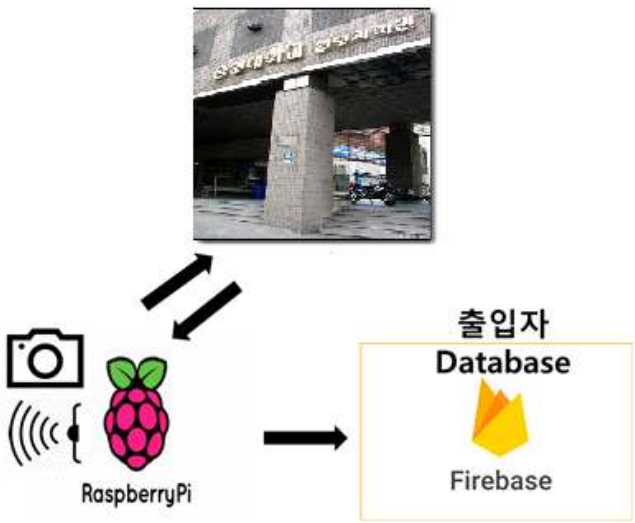
가. 작품 개요

인원이 체온을 직접 측정, 마스크 착용 여부 확인 후 주의 경고를 함으로써 불필요한 접촉이 발생한다. 적외선 카메라를 비치하는 건물도 있지만 마스크 착용 여부는 판단할 수 없고 비용적 한계가 있다.

이와 달리 Catch-Corona는 실시간 스트리밍 영상 프레임을 분석

하여 **인원의 마스크 착용여부**, **인원의 체온측정 정보**를 확인하고 이상이 있을 시 경고음과 함께 알림 메시지를 출력해준다. 그 후 측정한 정보들과 건물에 출입한 총 인원수 등의 **정보를 파이어베이스 실시간 데이터베이스에 즉시 전송**한다. 이는 컴퓨터 하나로 편리한 인원 관리가 가능하게 한다. 마스크 착용여부는 딥러닝을 통해 구현하고(마스크 착용한 사람 사진 500장 가량을 훈련시킨다.), 체온은 정확도가 높은 적외선 온도 센서(DTS-L300-V2)를 이용한다. 불필요한 접촉을 최대한 줄이고 스마트 하게 학교 건물 출입 인원들을 관리할 수 있는 프로그램이 Catch-Corona이다.

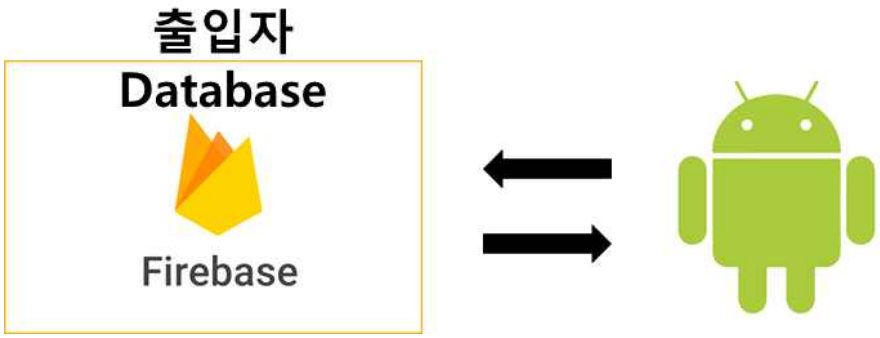
나. 시스템 구성도



[**그림 2. 시스템 구성도**]

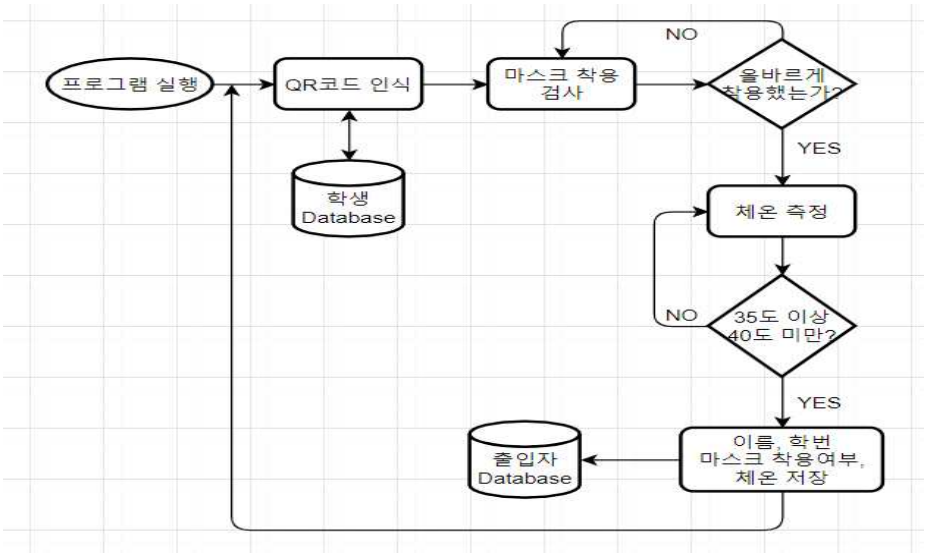
RaspberryPi에서 실시간 스트리밍을 시작한다. 영상 프레임 별로 분석하여 사람이 **특정 거리 내에 접근** 했을 때 (초음파 거리 센서 사용) **사람 얼굴을 인식**하고 얼굴 부분에 직사각형을 그린다. (Opencv4, Keras, Tensorflow 사용) 마스크를 착용한 사람의 얼굴 인식은 딥러닝 기술을 활용한다. 마스크 착용한 사람들의 구글 이미지와 마스크를 착용하지 않은 사람들의 이미지를 훈련 시켜 생성한 딥러닝 모델 파일을 이용한다. 얼굴 인식을 마치고 적외선 온

도 센서에 손목을 갖다 대면 **체온을 측정**한다. 그 후 스트리밍 화면에는 **마스크 착용 여부(MASK ON, MASK OFF)와 체온 측정 이상 여부(NORMAL, HIGH TEMPERATURE)** 텍스트를 출력한다.
한 명 측정이 완료 될 때마다 즉시 출입자 관리 서버에 데이터를 전송하고 Result 정보를 갱신한다. 데이터 전송 시에는 와이파이 무선 연결을 이용한다.



[**그림 3. 시스템 구성도**]

안드로이드 어플에서 파이어베이스 서버의 데이터를 불러올 수 있도록 한다. 관리자 한 명은 스마트폰 어플로도 건물 출입 현황을 확인할 수 있다.

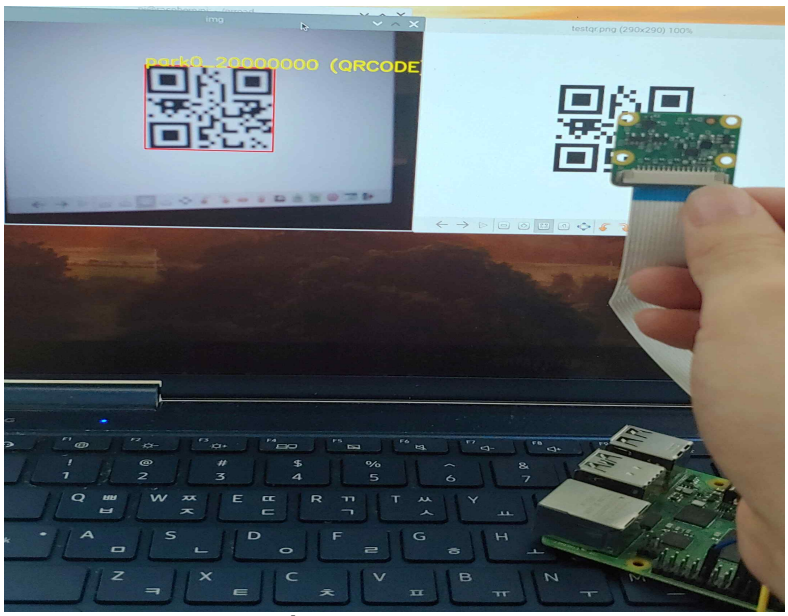


[**그림 4. 흐름도**]

프로그램은 ESC 키 입력 시 종료되도록 설계한다.

다. 구현기능

다-1. QR코드 인식



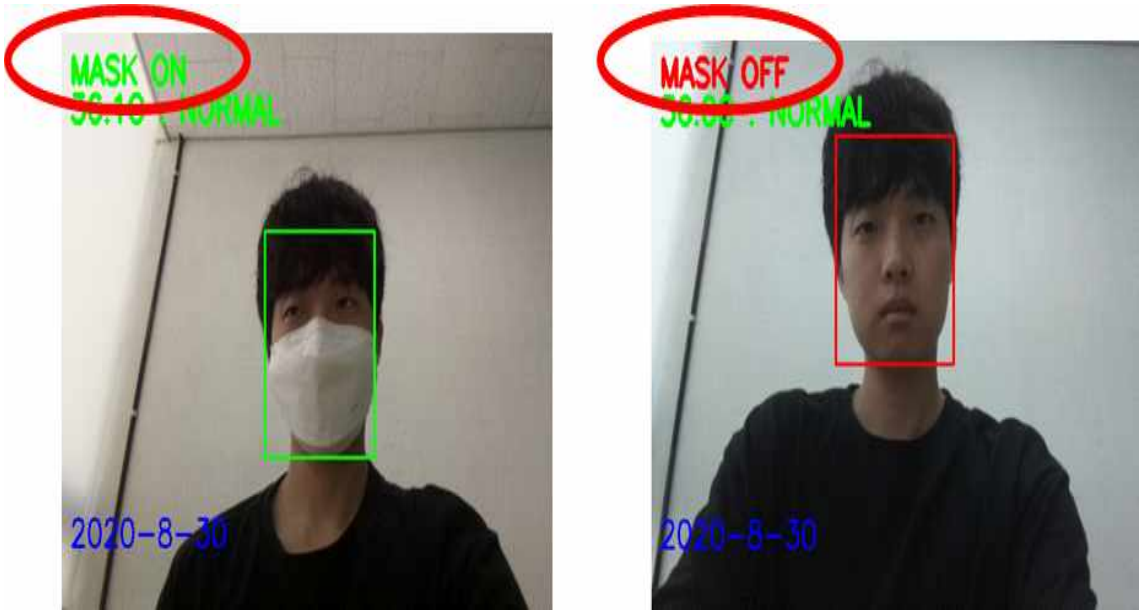
[**그림 5. QR코드 인식**]

RaspberryPi 카메라를 통해 학생증 QR코드를 인식하여 해당 학생의 이름, 학번정보를 학교 database로부터 받아온다. 현재 실제 학교 database에 접근할 수 없기에 임시로 data를 만들어 QR코드 인식, 정보를 획득하게 된다.

언어는 Python을 사용하였으며 OpenCV를 이용해 QR코드를 이미지로 임시 저장한 후 pyzbar를 이용하여 QR코드를 읽어 들이는 방식을 사용하였다.

다-2. 마스크 착용 여부 검사

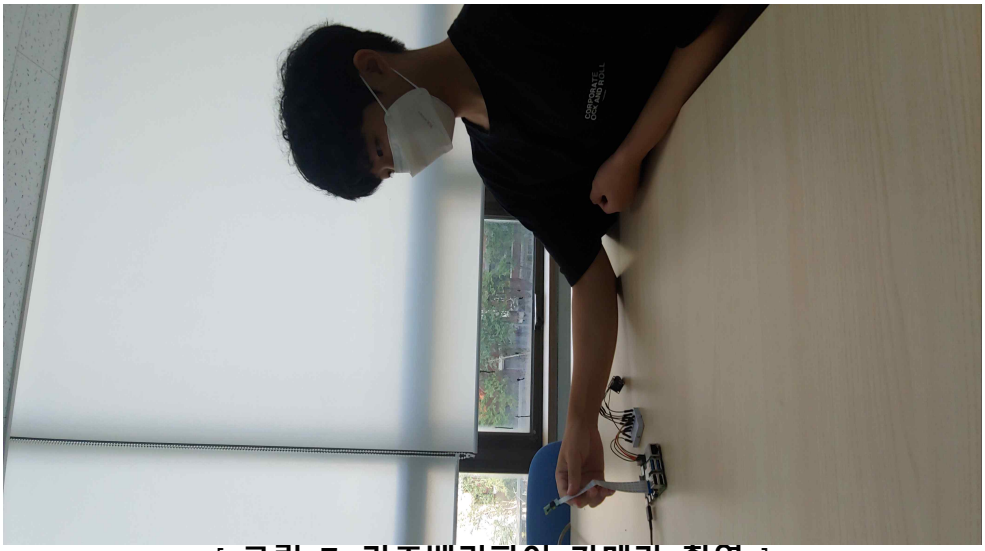
보통 얼굴인식은 OpenCV의 Haar cascade 알고리즘을 이용하지만 본 프로젝트에서는 마스크 착용 시의 얼굴도 인식해야 하기 때문에 haar cascade 알고리즘 대신 **OpenCV**와 **Keras**, **Tensorflow** 및 **딥러닝 모델**을 사용한다.



[**그림 6. 마스크 착용 여부 검사**]

마스크를 착용한 사람의 얼굴을 인식하려면 **먼저 마스크를 착용하지 않은 사람의 얼굴 이미지부터 분석**해야 한다. 마스크를 착용하지 않은 사람의 얼굴 이미지에서 **눈, 코, 입의 위치**를 찾아내고 그에 따라 적절한 위치에 임의의 마스크를 인위적으로 생성한다. 그렇게 마스크의 위치를 훈련시키는 것이 첫 번째이다. 인식의 정확도를 높이기 위해서는 마스크를 쓴 사람의 얼굴 이미지를 훈련 시켜 얼굴 내에서 마스크의 위치를 훈련시킨다. 이를 통해 생성된 mask 모델 파일을 이용한다. 이제 컴퓨터는 **얼굴 내에서 마스크의 위치를 '추론'** 할 수 있게 되고 최종적으로 마스크를 쓴 사람의 얼굴을 인식할 수 있게 된다. 구체적인 훈련 과정은 Pyimage search라는 컴퓨터 비전 관련 해외 개발자 홈페이지의 논문을 참고하여 설계한다. 훈련 이미지 파일 세트는 다음 github에서 가져왔다.

https://github.com/prajnasb/observations



[**그림 7. 라즈베리파이 카메라 촬영**]

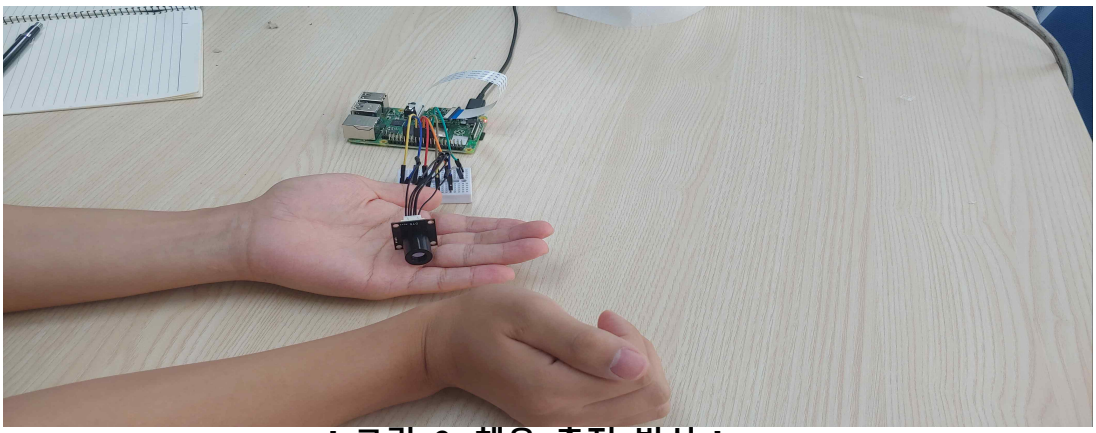
작품 외관은 추후에 보완하기로 한다. 이제 마스크 착용 여부를 인식할 수 있게 되었다. 마스크 착용 여부는 label 변수를 MASK OFF, MASK ON의 string 데이터로 갱신해준다. 그것을 스트리밍 화면에 출력해주고 색깔은 경고성을 표시하기 위해 MASK OFF일 때는 빨간색으로 출력한다. 스트리밍 화면에 출력할 때에는 cv2 라이브러리의 puttext 함수를 이용한다.

다-3. 체온 측정

[**그림 8. 체온 측정 후 상태 출력**]

wiringPi 라이브러리를 사용하여 GPIO를 제어한다. 이를 통해 적외선 센서가 라즈베리파이와 **SPI 통신**을 이용해 데이터를 받아온다. 그 데이터를 통해 최종 온도를 계산한다. 체온은 실시간 스트

리밍 항시 측정되고 있기 때문에 정상적인 사람 체온 (프로젝트에서는 **35.5도에서 42도 이내**로 정의하고 있다) 이 측정 될 때에만 **5번의 측정 평균값**을 real_temp 데이터에 저장한다.



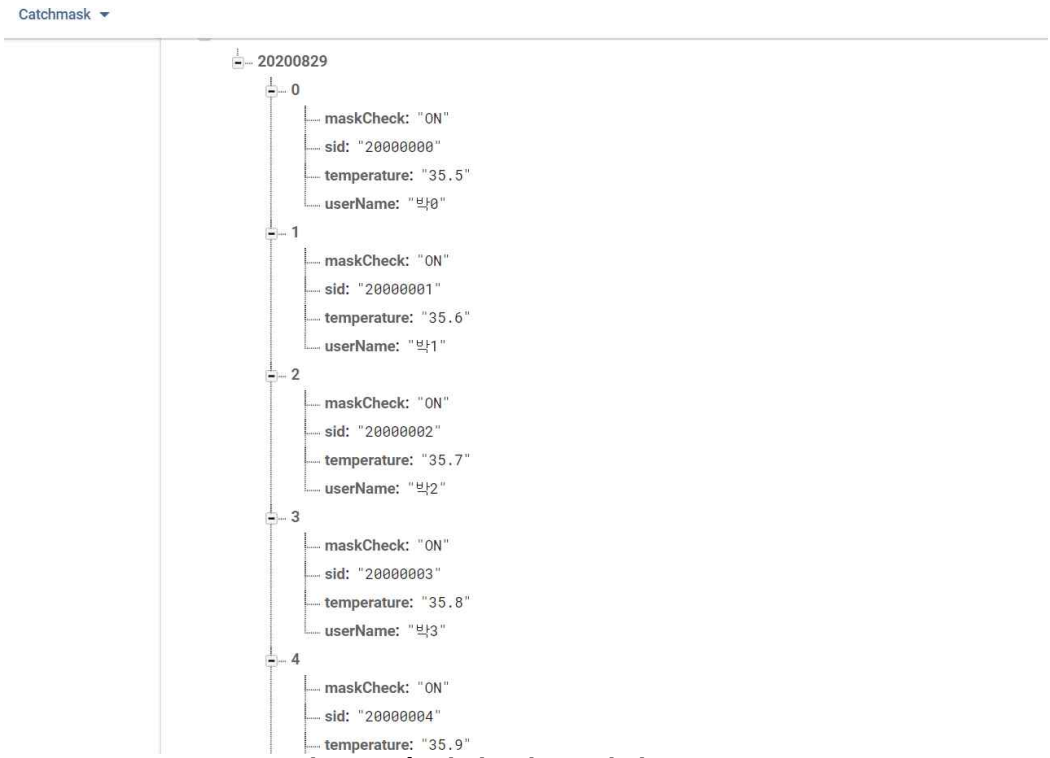
[**그림 9. 체온 측정 방식**]

적외선센서는 **'DTS-L300-V2'**를 사용하였으며 시중에서 판매하는 적외선 온도계에 사용되는 센서와 같은 성능을 가진다. 따라서 측정방법은 적외선 온도계 측정방식과 비슷하게 **일정거리에 손목을 갖다 댄다**.

이제 체온을 측정할 수 있게 되었다. 실시간 스트리밍 화면에 체온과 함께 현재 상태를 출력한다. 현재 공식적인 기준으로 37.5도 이상을 열이 있는 사람으로 분류한다. 본 프로젝트에서도 **37.5도 이상은 HIGH TEMPERATURE, 그 미만은 NORMAL 상태**로 분류한다.

다-4. 출입자 정보 저장

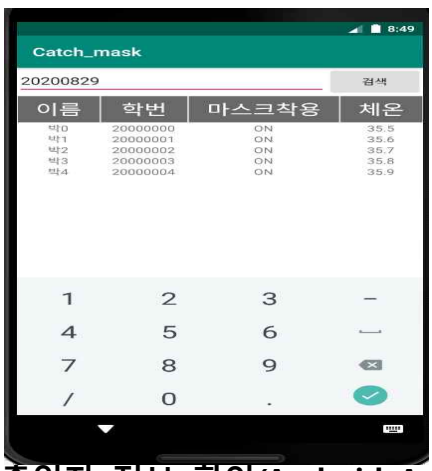
QR코드 인식, 마스크 인식, 체온 측정 과정을 통해 모든 정보를 얻었다면 그 데이터들을 **파이어베이스 실시간 데이터베이스에 전송하고 결과 데이터들을 수정**한다.



[**그림 10. 출입자 정보 저장(Firebase)**]

서버에 전송하는 시기의 기준은 측정을 완료한 사람의 얼굴이 실시간 스트리밍 화면에서 사라졌을 때이다. 이 때 이전에 측정했던 정보들을 (label 변수, real_temp 변수 값 등) 서버로 전송한다.

다-5. 출입자 정보 확인



[**그림 11. 출입자 정보 확인(Andorid Application)**]

컴퓨터 한 대로 서버 상태를 확인해도 되지만 **좀 더 편리하고 높은 접근성**을 위해 관리자에게는 관련 어플리케이션을 제공한다. 해당 어플리케이션은 android studio를 이용하여 만들었으며, firebase와 연동하여 ValueEventListener와 DataSnapshot을 이용해 데이터를 받아온다. 이 안드로이드 어플리케이션을 통해 건물 출입자 Database에 접근하여 원하는 날짜의 출입자 명단과 상태를 확인할 수 있다. 사용되는 database는 구글에서 제공하는 firebase를 이용 하였으며, realtime database를 이용하여 실시간으로 출입하고 있는 명단 또한 확인할 수 있다.
위의 [그림 11]은 안드로이드 어플리케이션을 이용한 [그림 10]의 firebase data를 확인한 결과이다.

7. 개발환경

구분		상세내용
S/W 개발환경	OS	Android 5.0이상, Raspbian, Window, Linux
	개발환경(IDE)	Android Studio, Python
	개발언어	Python – 영상처리(OpenCV4), 체온 측정(wiringPi), Deep Learning(Opencv4, keras, Tensorflow) JAVA – Android Application
	기타사항	Database – Google Firebase