

# 한이음 공모전 2017 개 발 보 고 서

2017. 10. 17

프로젝트명	국문	얼굴인식을 통한 인공지능 엘리베이터
	영문	Intelligent elevator using face recognition
작 품 명	너의 층수가 보여	
신 청 자	한국외국어대학교 / 김재민	

# 요 약 본

## 팀 정보

팀 명	AdvICE			
팀 원	이 름	소 속	부서/학과	직위/학년
멘 토	권위수	씨에스리컨설팅	아키텍처 3팀	팀장
지도교수	-	-	-	-
멘티 1(팀장)	김재민	한국외국어대학교	정보통신공학과	3
멘티 2	정상수	한국외국어대학교	정보통신공학과	3
멘티 3	이동열	한국외국어대학교	정보통신공학과	3
멘티 4	전진우	한국외국어대학교	정보통신공학과	2
멘티 5	-	-	-	-



작품 정보		
프로젝트명	국문	얼굴인식을 통한 인공지능 엘리베이터
	영문	Intelligent elevator using face recognition
작품명	너의 층수가 보여	
작품 소개	<ul style="list-style-type: none"> <li>○ 버튼을 눌러 작동시키는 기존의 엘리베이터에 얼굴인식 기능을 추가하여 자동으로 원하는 층 수를 눌러주는 인공지능 엘리베이터</li> <li>○ 엘리베이터 작동이 어려운 사람에게 미리 등록된 승객정보를 이용해 얼굴인식을 통한 승객들의 층수를 자동으로 눌러주는 편의 기능 제공</li> <li>○ 음성인식 기능과 SMS 기능을 추가하여 기존의 엘리베이터보다 더욱 편리하게 이용가능하며 불편점 해소</li> <li>○ 관리자 측면에서도 시스템 서버 내에 이용자의 정보 및 승객의 사진이 포함되어있어 웹 서비스를 통한 이용자 관리에 수월</li> </ul>	
작품 구성도	<p>시스템은 엘리베이터 동작부, 제어부, 서버, Camera 영역으로 나눌 수 있다.</p>	
작품의 개발배경 및 필요성	기존의 엘리베이터 이용이 불편한 승객에게 편리한 기능을 접목시켜 불편함을 해소하고 편리성을 제공하기 위해 개발	
작품의 특징점	<ul style="list-style-type: none"> <li>○ 얼굴인식을 이용해 버튼을 자동으로 눌러주어 기존 엘리베이터에 비해 편리함</li> <li>○ 음성인식, SMS기능, 가이드 정보 등 다양한 서비스제공을 통해 효과적이며 효율적인 사용이 가능하며, 추가적인 비용이 없어 경제적인 설치가 가능</li> </ul>	
작품 기능	<p>주요기능에는 Feature Matching 알고리즘을 이용한 얼굴인식을 통해 버튼을 자동으로 눌러주는 자동화버튼 기능, 등록된 층 이외의 층으로 가고 싶은 경우 API를 이용해 음성으로 층수를 눌러주는 음성인식 기능, SMS기능을 통해 관리실로 비상메시지를 전송하여 기존의 비상벨보다 빠른 대처가 가능한 비상메시지 전송 기능, 눌러진 층수에 대한 간략한 정보를 알려주는 각 층별 가이드정보 기능, 승객의 정보를 웹서비스를 통해 DB에 저장하여 이용 승객의 정보를 편리하게 관리할 수 있는 고객 관리 웹서비스 등이 있다.</p>	
작품의 기대효과 및 활용분야	기대효과에는 승객에게 얼굴인식을 통한 자동화버튼으로 편의성 제공과 고층건물에 필수적인 엘리베이터를 통한 넓은 시장성이 있고 활용분야에는 엘리베이터 업체들과의 제휴를 통한 추후 창업가능성과 얼굴인식을 이용한 다양한 분야 접목이 가능	

# 본 문

## I. 작품 개요

※ 평가항목 : 기획력 (필요성, 차별성)

### 1. 작품 소개

#### ○ 엘리베이터 작동이 어려운 사람들을 위한 엘리베이터 기획

- 엘리베이터는 동력을 사용하여 사람이나 화물을 수직 방향으로 이동하기 위해 사용하는 장치로써 고층 건물을 이용하는 승객들에게 필수적인 장치
- 두 손 가득 짐을 든 사람, 몸이 불편한 사람, 어린이와 같이 엘리베이터 작동이 어려운 사람에게 편의성 제공

#### ○ 기존의 엘리베이터에 인공지능을 결합한 서비스

- 미리 등록된 승객정보를 이용해 얼굴인식을 통한 승객들의 층수를 자동으로 눌러주는 기능 제공
- 엘리베이터를 이용하는 승객에게 보다 나은 편의성을 제공하기 위해 기존 엘리베이터에 음성인식 기능과 SMS기능을 더하여 추가 제공
- 음성인식 기능을 이용하여 다수의 사람들이 엘리베이터에 탑승 시에 자신이 원하는 층의 버튼을 누르기 위해서 여러 사람과의 불필요한 접촉 방지
- SMS기능을 이용하여 위급상황 시 관리자에게 즉시 메시지 전송을 통한 신속한 대처 가능

#### ○ 웹 서버스를 통한 승객의 데이터관리 제공

- 관리자 측면에서도 시스템 서버 내에 이용자의 정보 및 승객의 사진이 포함되어있어 웹 서비스를 통한 이용자 관리에 수월

#### ○ 작품내용

- 얼굴 검출(Haar-like Feature), 비교(Feature Matching) 알고리즘을 사용하여 승객의 얼굴을 검출 후 비교하여 등록되어 있는 층수의 버튼을 엘리베이터가 자동으로 눌러줌
- 해당 건물에 등록된 승객이 아닌 경우 직접 버튼을 누르거나 음성인식을 통해 이용 가능
- 엘리베이터의 동작부분을 구현하기 위하여 아두이노, 제어부분을 구현하기 위하여 스마트 폰을 사용

## 2. 작품의 개발 배경 및 필요성

### ○ 승객에게 편리성 제공

- 기존의 엘리베이터는 짐을 든 사람, 몸이 불편한 사람, 엘리베이터 버튼에 손이 잘 닿지 않는 어린이와 같이 특수성이 있는 사람들에게 엘리베이터 사용에 불편함이 있어 여러 기능을 접목시켜 승객의 불편함 해소

### ○ 저비용으로 여러 가지 서비스 제공

- 여러 기업들이 홍채인식, 동작감지와 같은 기술을 접목시킨 엘리베이터를 개발하고 있으나 가격적 부담
- 본 프로젝트는 기존 엘리베이터의 CCTV 카메라에 얼굴인식 기능을 접목시켜 서비스 제공이 가능 하므로 경쟁력 확보

## 3. 작품의 특징 및 장점

### ○ 주요 기능적 측면 차별화

구 분	기 능	설 명
S/W	고객정보 관리	- 등록된 승객들의 데이터를 웹 서버스를 통해 한눈에 파악하는데 용이
	얼굴인식	- 얼굴인식 기능을 통해 버튼입력 없이 등록된 층수 입력 기능 제공
	음성인식	- 음성인식 기능을 통해 실시간으로 층수 자동 입력 기능 제공
	SMS	- SMS 기능을 통해 위급상황시 관리자에게 메시지 전송기능 제공
	각 층별 가이드정보	- 승객들에게 각 층수마다 시설정보를 제공
H/W	저비용	- 기존에 엘리베이터의 CCTV카메라 이용만으로 서비스 제공이 가능

### ○ 기존 제품과의 비교

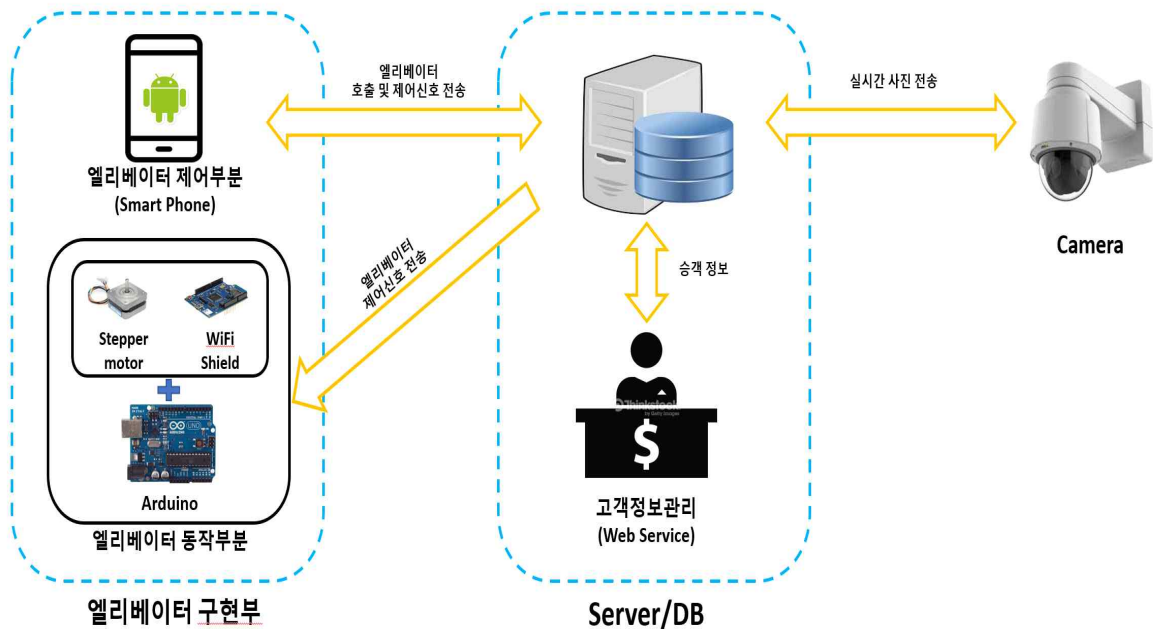
측정지표	기존 엘리베이터	본 프로젝트 엘리베이터 (너의 층수가 보여)
편리성	- 승객이 층수를 직접 입력 시 원하는 층수로 이동	- 승객이 버튼을 직접 누를 필요 없이 원하는 층으로 이동 가능
기능성	- 버튼 입력 시 엘리베이터 작동 - 비상시 호출벨 이용 가능	- 얼굴인식을 통한 엘리베이터 작동 - 음성인식을 통한 엘리베이터 작동 - 버튼 입력 시 엘리베이터 작동 - 비상상황시 sms전송을 통한 대처가능 - 해당 층의 시설 정보메시지 제공 - 웹서버스를 통한 데이터관리 수월
경제성	보편화 되어 있음	- 기존 엘리베이터에 추가적인 비용 없이 접목 가능하므로 보다 경제적
사용성	단순한 엘리베이터 기능 수행	- 다양한 서비스 제공을 통해 기존 엘리베이터보다 편리하고 효과적임

## II. 작품 내용

※ 평가항목 : 기술력 (기능구체성, 난이도, 완성도)

### 1. 작품 구성도

#### 1-1. 전체 시스템 구성도



위의 그림은 본 작품의 전체 시스템 구성도이다. 시스템은 크게 엘리베이터 동작부, 제어부, 서버(DB 포함), Camera 영역으로 나눌 수 있다.

#### 1) 엘리베이터 동작부(아두이노)

- 실제 엘리베이터를 가지고 테스트 및 작품 시연에 제약 사항이 많아 아두이노를 이용해 엘리베이터를 구현하였다. 아두이노에서는 WiFi Shield를 통해 서버와 TCP/IP 통신을 하며 서버 측으로부터 엘리베이터 제어신호, 층수를 받는다. 제어신호에 따라 아두이노는 스텝모터를 이용하여 엘리베이터를 작동시킨다.

#### 2) 엘리베이터 제어부(스마트 폰)

- 엘리베이터 승객에게 엘리베이터를 제어할 수 있게 화면을 보여주는 역할을 한다. 안드로이드 기반의 애플리케이션에서 동작하며 서버와 TCP/IP, JSON 통신을 이용해 정보를 주고받는다.
- 스마트 폰과 서버 간에 엘리베이터 제어신호는 엘리베이터 호출신호와 엘리베이터 층 선택 신호이다. 엘리베이터 호출신호는 엘리베이터를 해당하는

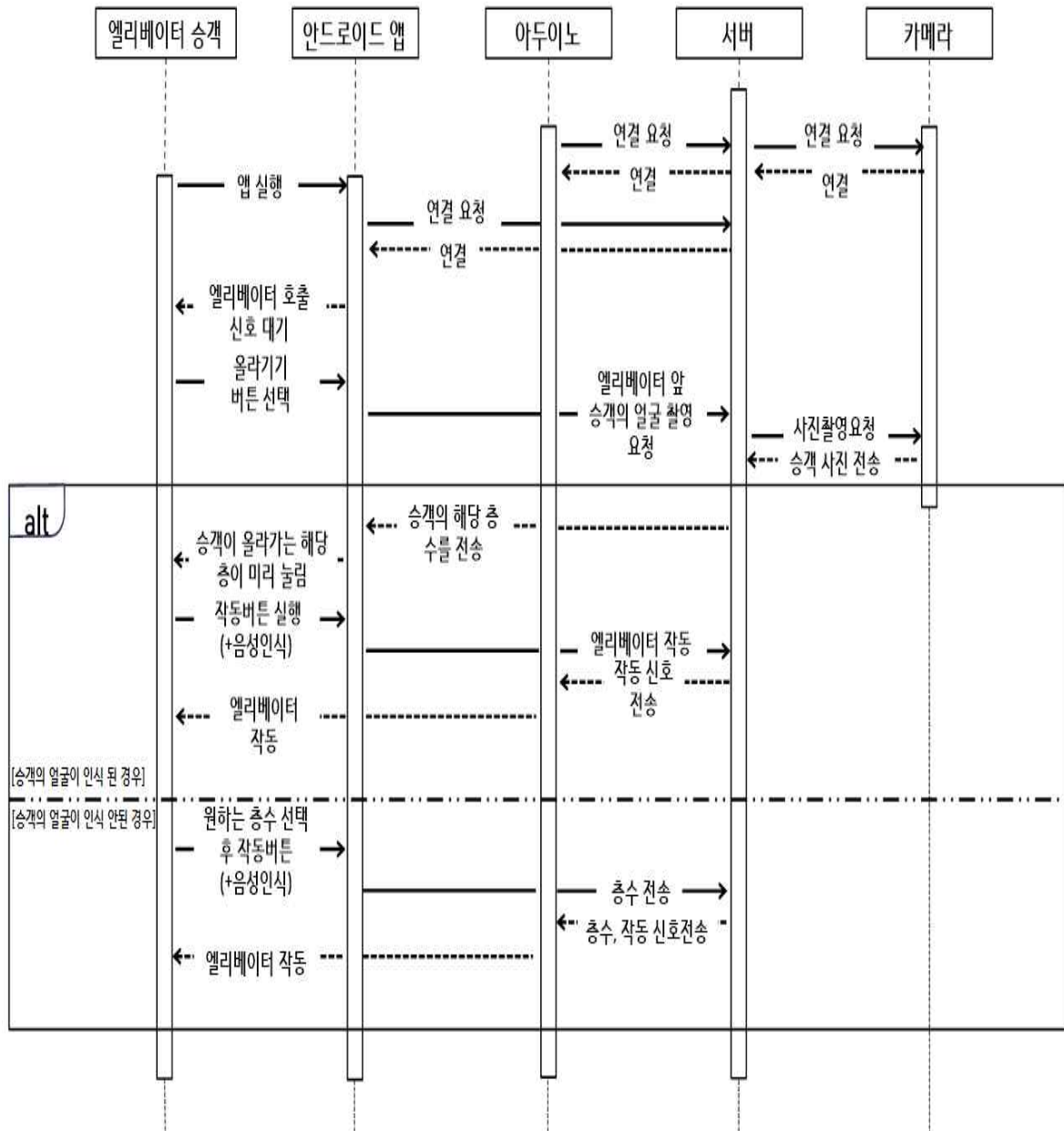
층으로 호출하면서 엘리베이터 앞 승객들의 얼굴을 카메라로 촬영하는 신호를 보낸다. 그 후 서버로부터 받은 인식 결과에 따라 엘리베이터 승객의 해당 층수가 탑승 전 결정되어 버튼이 눌러진다.

- 엘리베이터 층 선택 신호는 등록되지 않은 승객의 경우 가고자 하는 층의 버튼을 누르거나 음성인식을 통해 서버로 층 수를 보낸다.
- 제공하는 서비스로는 해당 층의 버튼이 눌러질 경우 가이드정보를 받아서 승객에게 보여준다. 또한 엘리베이터 내에 응급상황 발생 시 SOS버튼을 눌러 관리실에 메시지를 전송해 응급상황을 알려줄 수 있다.

### 3) 서버 / DB

- 서버는 앱과 TCP-IP 통신을 하며 앱에서 프로토콜에 맞게 요청메세지가 오기를 기다린다. 요청 메시지의 종류는 엘리베이터 호출제어, 엘리베이터 층수 선택 제어, 가이드정보 선택 등이 있으며 요청 메시지가 오면 요청 메시지에 따른 기능을 수행한다.
- 서버에서는 카메라로부터 받은 사진을 통해서 승객의 얼굴을 검출하고, 데이터베이스에 저장된 사진과 비교하여 동일인물인지를 판별해낸다. 동일 인물이 있을 시 엘리베이터 구현부분으로 정보를 전송한다.

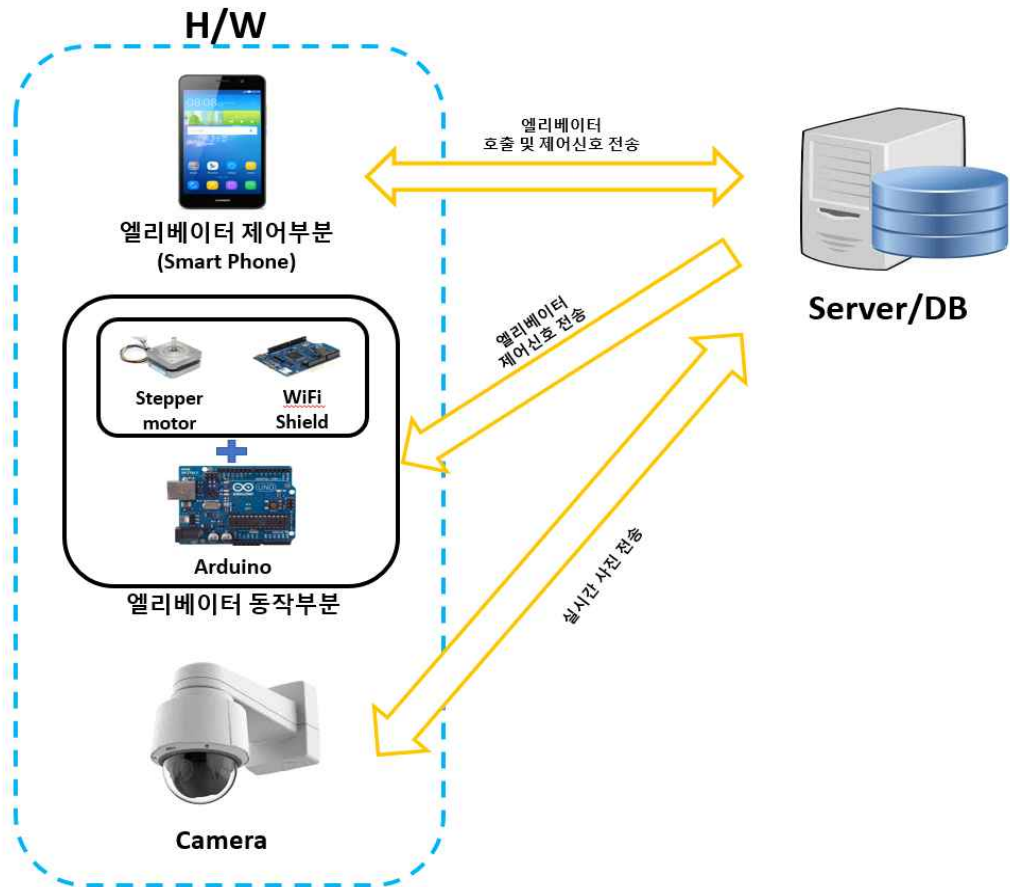
## 1-2. S/W 구성도



- 서버와 아두이노, 카메라를 연결하여 시스템 기반을 구축한 후 사용자가 스마트폰에서 앱을 실행하여 서버와 연결을 한다. 사용자가 엘리베이터 호출 버튼을 누르면 엘리베이터는 해당 층수로 호출되며, 문이 열리기전에 엘리베이터 앞에 기다리는 승객의 얼굴을 촬영한다.
- 기존에 등록된 승객이 맞으면 저장된 층수가 탑승 전에 미리 버튼이 눌러져 있으며, 그렇지 않으면 승객이 직접 버튼을 누르거나 음성 인식을 통해 엘리베이터를 작동해야 한다.

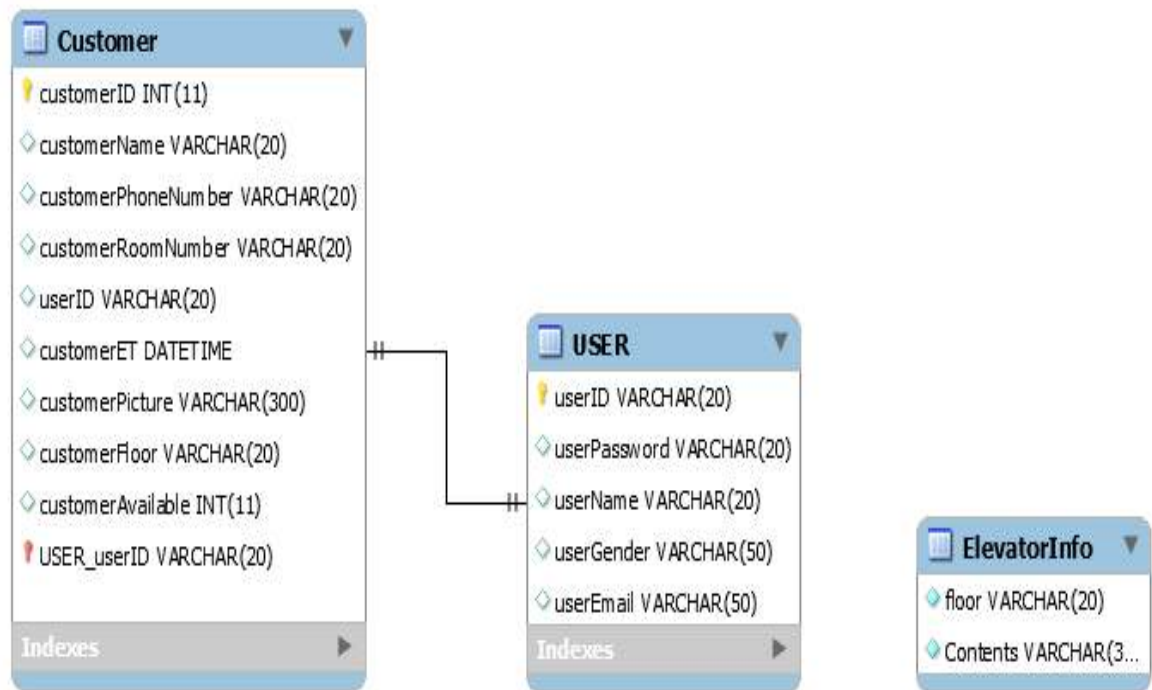


### 1-3. H/W 구성도



- 스마트폰은 승객이 엘리베이터를 제어하기 위한 서버TCP/IP 통신을 하며 아두이노 작동을 제어하기 위한 정보들을 통신한다.
- 실제로 엘리베이터를 가지고 구현하는데 제한점이 있어 아두이노를 이용해 구현을 하였다. 아두이노에 스텝모터를 부착하여 엘리베이터의 작동이 가능하게 만들었고, 동작을 하기 위한 제어신호를 와이파이 쉴드를 이용하여 서버로부터 결과 값을 받는다.
- 카메라는 엘리베이터 앞에 승객의 얼굴 촬영하기 위하여 사용되었다. 서버에서 사진촬영 신호가 오면 사진촬영 후 서버로 제어 값을 전송한다.

#### 1-4. DB Modeling



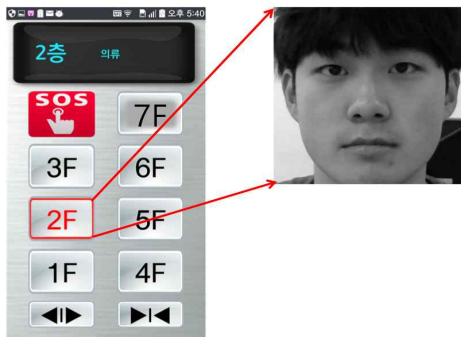

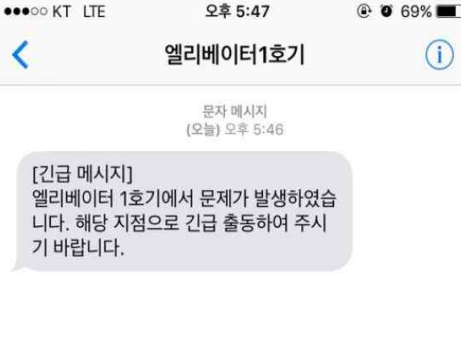

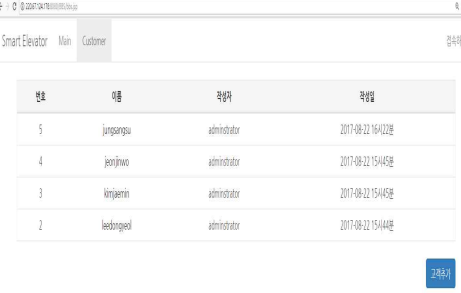
데이터베이스 정보는 고객(승객)정보, 건물가이드정보, 웹 서비스 계정정보 3가지로 구성되어 있다. 데이터베이스에 저장된 정보에는 이름, 전화번호, 호실 등 개인 신상정보와 얼굴데이터 사진경로, 승객을 관리할 수 있는 관리자정보가 들어있다. 웹서비스 계정정보는 고객을 관리할 수 있는 관리자 정보를 뜻한다. 마지막으로 건물가이드 정보는 각 건물 층마다 승객이 필요한 가이드 정보를 저장한다.

## 2. 작품 기능

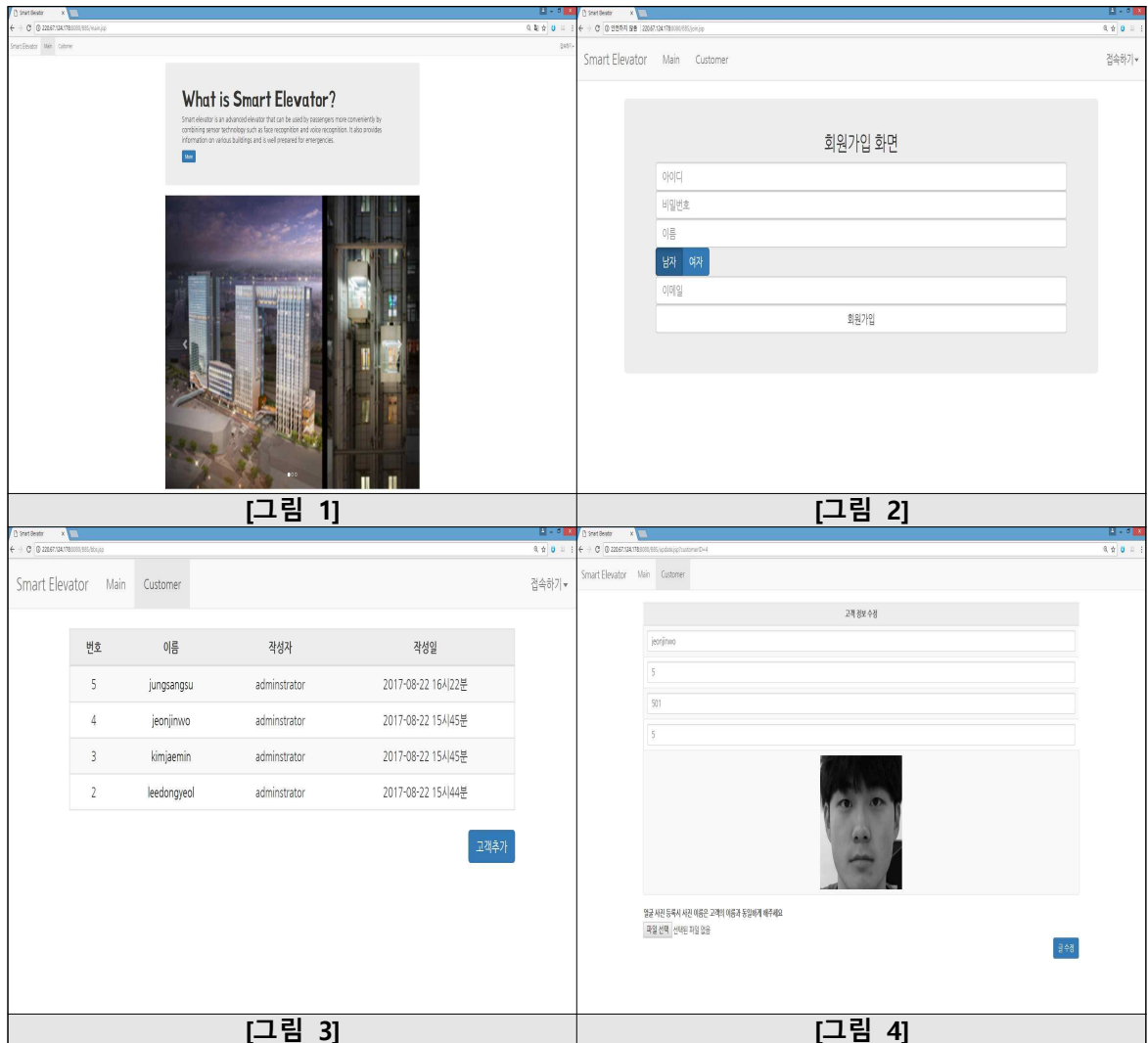
### 2-1. 전체 기능 목록

구 분	기 능	설 명	현재진척도(%)
S/W	얼굴인식을 통한 자동화버튼	사용자의 얼굴을 촬영한 후 등록된 데이터와 비교하여 층수를 자동으로 눌러준다.	100%
	음성인식	등록된 층 이외의 층으로 가고 싶은 경우 음성인식을 통하여 원하는 층의 버튼을 눌러준다.	100%
	비상메시지 전송	응급상황 발생 시 SMS기능을 통해 관리실로 비상메시지를 전송해준다.	100%
	각 층별 가이드정보	엘리베이터에서 층수 버튼을 누를 시 승객에게 해당 층에 대한 간략한 시설정보를 알려준다.	100%
	고객 관리 웹서비스	승객의 정보를 웹서비스를 통해 DB에 저장하여 이용 승객의 정보를 편리하게 관리할 수 있다.	100%
H/W	엘리베이터 모형	실제 엘리베이터를 대신할 모형 엘리베이터를 구현한다.	100%
	소켓 통신	서버와 아두이노, 애플리케이션에서 데이터를 주고받을 때 사용된다.	100%
	영상 촬영	얼굴 인식에 사용되는 영상사진을 촬영한다. 서버와 통신을 통해 촬영정보를 전송 후 얼굴비교에 사용된다.	100%

## 2-2. S/W 주요 기능

기 능	설 명	작품실물사진
얼굴인식을 통한 자동화버튼	엘리베이터 앞 상승버튼을 누르면 사용자의 얼굴을 촬영한 후 기존에 등록된 데이터와 비교하여 일치하는 사용자정보를 찾아 등록되어있는 층수를 자동으로 눌러준다.	
음성인식	등록되어 있지 않은 승객이나 등록된 층 이외의 층으로 가고 싶은 경우 버튼을 누르는 대신 음성인식을 통하여 원하는 층이 자동으로 눌러지는 편의성을 제공한다.	
비상메시지 전송	엘리베이터에서 사용하는 비상버튼을 구현하였다. 응급상황 발생 시 엘리베이터 내의 버튼을 누르면 SMS기능을 통해 관리실로 비상메시지를 전송하여 기존의 비상버튼보다 빠른 대처가 가능하다.	
각 층별 가이드정보	엘리베이터에서 층수 버튼을 누를 시 해당 층에 대한 정보를 알려준다. 여러 층을 입력 시에 마지막으로 입력된 층에 대한 정보가 나타나며 승객에게 해당 층에 대한 간략한 시설정보를 알려준다.	
고객 관리 웹서비스	엘리베이터 이용 승객의 정보를 저장한다. 승객의 정보를 웹서비스를 통해 DB에 저장 및 수정할 수 있으며 이용 승객의 정보(이름, 전화번호, 호실번호, 층수, 승객사진 등)를 편리하게 관리할 수 있다.	

## ○ 사용자 인터페이스




[그림1] 웹서비스를 제공하는 메인페이지로써 이용자에게 제공하는 서비스가 무엇인지 간략하게 소개한다.

[그림2] 서비스 이용을 위한 회원가입기능을 구현하였다. 아이디, 비밀번호, 이름, 성별, 이메일 주소를 통해 회원가입 절차가 진행되며 완료된 이용자에 한해서만 서비스 이용이 가능하다.

[그림3] 등록된 회원의 정보들을 보여주는 리스트화면이다. 고객추가버튼을 통해 엘리베이터를 이용할 승객의 이름, 호실, 사진을 입력받아 데이터베이스에 저장한다. 등록이 완료되면 작성자 이름에는 로그인한 이용자의 아이디가 입력되고 작성일을 통해 등록된 시간을 확인할 수 있어 회원관리가 수월하다.

[그림4] 이미 등록되어 있는 회원의 정보를 수정하는 화면이다. 만일 회원의 정보 수정이 필요할 경우 수정버튼을 통해 원하는 정보를 수정이 가능하다.

## 2-3. H/W 주요 기능

기능/부품	설명	작품실물사진
엘리베이터 모형	아두이노 우노에 NEMA17 스텝모터를 연결하고 모터는 모형 엘리베이터의 위쪽에 설치한다. Arduino Sketch를 이용하여 엘리베이터 동작 프로그래밍을 Arduino Uno에 입력해서 서버에서 층수 값을 받아올 시 원하는 층수로 이동할 수 있다.	
소켓 통신	Arduino Uno에 Wifi 실드를 추가 설치하여 서버와 소켓통신을 함으로서 엘리베이터 모형이 작동할 수 있다.	
영상 촬영	엘리베이터와 카운터에서 승객의 얼굴을 촬영한다. 촬영한 사진은 얼굴 인식을 위해 사용되며 서버로 전송된다.	

## 3. 주요 적용 기술

### 3-1. 적용기술

구 분	세부기술	설 명
적용기술	Socket	네트워크 통신 인터페이스의 일종으로 한 쪽에서 서버가 되고 반대편이 클라이언트가 되며 클라이언트가 서버 쪽에 연결을 요청 하면 서로 연결이 성립되고 데이터를 주고받는다. '너의층수가보여'에서는 안드로이드 앱, 아두이노와 윈도우 서버통신 하는데 사용 된다
	다중 쓰레드	안드로이드 음성인식과 서버와 소켓통신 간에 쓰레드 충돌로 인해 다중 쓰레드를 사용하여 음성인식과 서버 간에 흐름제어를 원활하게 이루었다
	음성인식 API	프로그램 또는 애플리케이션이 운영 체제에 어떤 처리를 위해서 호출할 수 있는 서브루틴 또는 함수 집합이다. 그 중 Naver에서 제공하는 음성인식 API를 이용하여 음성인식 부분을 구현하였다
	SMS 전송	단문 메시지 서비스의 약어로 휴대전화를 이용하는 사람들이 별도의 다른 장비를 사용하지 않고 휴대전화로 짧은 메시지를 주고받을 수 있는 서비스이다
	JSON	안드로이드 어플리케이션에서 서버와 연결 된 데이터베이스에 접근하기 위해 웹을 사용해 JSON 통신을 하였다
	Tomcat server	웹 서버를 구축하기 위해서 사용하였다. 웹 프로그램에서 HTML뿐만 아니라 JSP 및 CSS를 사용하여 웹서비스를 구축하는데 사용되었다

### 3-2. 알고리즘

#### ○ 얼굴인식(Haar-like Feature)

- 최초로 실시간 얼굴 인식에 사용된 알고리즘으로써, 이미지에서 얼굴의 특징을 찾는 기법중 하나인 Haar-like Feature는 그림과 같이 흑백 영역에 대한 픽셀 값의 평균 차에 의한 임계치 구분에 의해 특징을 판단, 파악하는 기법이다.
- 다른 알고리즘들에 비해 연산과정이 간단하여 실시간 얼굴 검출에 적합하다.
- 영상에서의 영역과 영역의 밝기 차를 이용한 것으로 그림과 같이 사각형의 흰색 부분 영상 픽셀들의 밝기 합에서 검은색 부분의 밝기 합을 뺀 차로 계산된 각각의 기본 feature들의 특징들에 따라 그것들을 다수 조합하여 다양한 위치 및 크기에서 물체에 대한 특징을 추출하는 방법이다.
- 물체를 추출하는데 있어 대상물체인지 아닌지를 판단하기 위해서는 의미 있는 feature들을 파악하는 것이 중요하다.

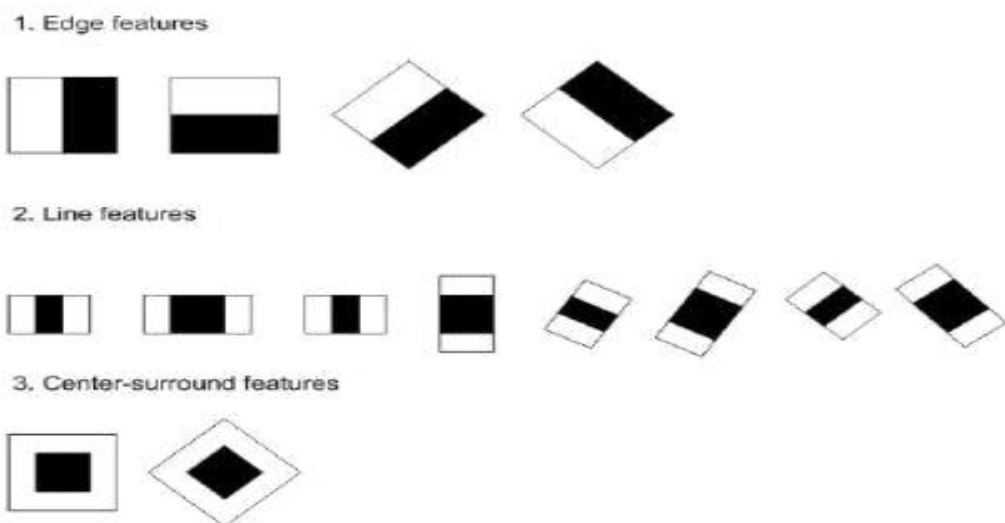


그림 5. Haar-like 특징  
Fig 5. Haar-like feature

### 3-2. 비교 알고리즘 (Feature matching)

종류	Comparing Histograms	Template Matching	Feature Matching
사진			
특징	숲 이미지는 녹색 계열로 되어있음을 인식해 녹색을 많이 가지고 있으면 숲로 인식	큰 이미지에서 주어진 작은 이미지를 검색하는데 유용	이미지로부터 수많은 특징점을 추출하여, 해당부분이 회전, 확대/축소되어도 동일한 특징점으로 인식
단점	녹색을 가진 물체는 모두 숲으로 인식	두 이미지 유사도 비교는 어려움	결과도출 속도가 느림

- 이미지의 유사도를 알아보는 대표적인 알고리즘 중 하나로써, 가장 효과적인 방식으로 판단된다.
- 이미지로부터 수많은 특징점들을 추출하여 비교하는 알고리즘으로써, 비교하고자 하는 해당 부분이 회전, 확대, 축소되더라도 동일한 특징점으로 인식한다.
- 추출된 특징점들을 활용하여 다른 이미지의 피쳐셋과 비교하면서 유사성을 검사한다.
- 두 개의 이미지에서 추출된 특징점들이 높은 비율로 일치한다면 동일하거나 유사한 이미지로 판별된다.
- 비록 결과도출까지의 시간이 다른 알고리즘에 비해 오래 걸리지만 이미지를 비교하는데 있어 사람이 느낄 수 있는 차이가 아니다.



#### 4. 작품 개발 환경

구분		상세내용
S/W 개발환경	OS	Windows
	개발환경(IDE)	eclipse EE/SE, Android Studio, Arduino sketch
	개발도구	Arduino tool, Android SDK, Logitech Webcam S/W, apache(Tomcat), Mysql(RDS), openCV library(ver2412)
	개발언어	Java, JSP, C, XML
H/W 구성장비	디바이스	Smart Phone, Logitech Webcam, Arduino (NEMA-17 Stepper Motor 42 motor, L298N moter driver ,WiFi shield)
	통신	TCP/IP , JSON, WIFI
	개발언어	Android Java, Arduino C
프로젝트 관리환경	형상관리	GitHub
	의사소통관리	Goto meeting, kakao Talk
	기타사항	project Web page(Smart Elevator)

### III. 프로젝트 수행 내용

※ 평가항목 : 수행능력 (문제해결능력, 수행충실성)

## 1. 멘티(참여학생) 업무분장

번 호	이 름	대 학	학 과	학 년	역 할	담당업무	
1	김재민	한국외국어대학교	정보통신공학과	3학년	팀장	정	웹 서비스 구현 이미지 프로세싱
						부	앱 구현(Android)
2	정상수	한국외국어대학교	정보통신공학과	3학년	팀원	정	웹 서비스 구현 서버 구현 이미지 프로세싱
						부	앱 구현(Android)
						정	데이터베이스 설계 엘리베이터 구현(Arduino)
3	이동열	한국외국어대학교	정보통신공학과	3학년	팀원	부	앱 구현(Android)
						정	앱 구현(Android) 엘리베이터 구현(Arduino)
4	전진우	한국외국어대학교	정보통신공학과	2학년	팀원	정	앱 구현(Android) 엘리베이터 구현(Arduino)
						부	엘리베이터 구현(Arduino)

## 2. 프로젝트 수행일정

[illegible]

### 3. 프로젝트 추진 과정에서의 문제점 및 해결방안

#### 3-1. 프로젝트 관리 측면

##### 1) 프로젝트 초기

- 프로젝트 초기에는 팀원 전체가 학기 중이었기에 잦은 모임을 가지기 어려울 때가 많았다. 따라서 모임은 일주일에 1~2번 정도로 조정하고 만날 때마다 각자 파트를 정하여 다음 모임 전까지 준비해오는 식으로 진행하였다. 또한 중간중간 메신저를 이용해 잦은 회의를 가져 관련 기술과 정보를 조사하여 프로젝트를 설계하였다.

##### 2) 프로젝트 중반

- 프로젝트 중반인 방학기간에는 매주 월~금 오전 10시부터 오후 4시까지로 시간을 정하고 개인적인 일이 있는 날은 최소한 2주 전에는 말할 수 있도록 해서 가능한 다 같이 모여서 프로젝트를 진행할 수 있게 하였다.
- 프로젝트가 진행됨에 따라 각자 맡은 파트에서 문제점이 하나 둘 씩 보이기 시작했다. 이를 해결하기 위하여 모임이 끝날 시간인 4시 전 후로는 각자 메신저에 그날의 진행상황을 올려 서로의 상황 및 문제점을 알 수 있게 하였다.
- 일주일에 적어도 한번은 모여서 회의 진행을 함으로서 부족한 부분과 문제점을 인식하고 해결방안을 토의하였다. 여기서 나온 문제점과 해결책은 2~3주에 한번 씩 멘토와의 미팅에서 논의해 더욱 좋은 방법을 추구해나갔다.

대분류	중분류	상세내역	구현 현황	진행률	문제점	우선순위	비고
이미지 프로세싱	얼굴 검출 Algorithm 정: 정상수 부: 김재민	lbpcascade frontalface 적용	완료	100%		1	
		얼굴검출 수에 따른 얼굴이미지 저장	완료	100%		3	
		전체 알고리즘 작성	진행중	90%		2	
		1차 테스트	진행중	90%			
	얼굴 비교 Algorithm 정: 정상수 부: 김재민	feature matching 적용	완료	100%		1	
		인식률 상승 필터 적용	완료	100%		2	
		얼굴 유사율 조정	완료	100%		3	
		1차 테스트	진행중	70%			
	2차 테스트		진행 예정	0%			
Elevator	Arduino 정: 전진우 부: 이동열	엘리베이터 모형 구현	진행중	40%		2	
		카메라와 연동	진행 예정	0%		5	
		1차 테스트	진행중	30%			
	Application (Smart Phone) 정: 이동열 부: 전진우	전체 UI 설계	완료	100%		2	
		버튼 시각화	완료	100%		3	
		Server와 연동	완료	100%		1	
		얼굴인식	진행중	80%		1	
		음성인식	진행중	50%		3	
		가이드 정보	완료	100%		4	
		비상 메시지 전송	완료	100%		3	
		1차 테스트	진행중	50%			
2차테스트		진행중	50%				
Server /DB	Server 정: 정상수 부 :김재민	Application과 연동	완료	100%		1	
		Arduino와 연동	완료	100%		1	
		Web이랑 연동	완료	100%		2	
		실시간 동기화	완료	100%		3	
		1차 테스트	진행중	80%			
	DB 정: 이동열 부: 김재민	승객 인적사항 저장	완료	100%		2	
		승객 얼굴 데이터 저장	완료	100%		2	
		해당 층 가이드 정보 저장	완료	100%		4	
		1차 테스트	진행중	100%			
		2차테스트		진행중	50%		
최종테스트		진행예정	0%				

### 3-2. 작품 개발 측면

#### ○ SW 문제점 및 해결방안

##### 1) 얼굴 인식을 저조

문제점	데이터베이스에 저장된 사진과 엘리베이터에서 찍은 사진의 환경이 다르면 인식을 저하
해결방법 및 기술	비교 사진들에 회색 필터를 적용하고, 검출된 얼굴 특징점의 유사 개수를 조정하여 해결

##### 2) 음성인식 자체 종료

문제점	한번 음성을 인식받게 되면 그 이후 데시벨이 낮아질 경우 자동적으로 종료되는 현상 발생
해결방법 및 기술	음성인식 후 데시벨이 낮아져 EndPoint가 잡히게 되면 EndPoint를 제거 후 재실행

##### 3) 소켓 통신 시 문자열 객체전달

문제점	JAVA로 작성된 클라이언트가 서버에 Stream객체로 문자열 전송 시 C로 구현한 서버에서 받은 문자열 깨짐 현상이 일어남
해결방법 및 기술	문자열을 Byte로 변환하여 Buffer로 보내고 다시 서버는 Byte로 받은 내용을 문자열로 변환하여 사용 함

##### 4) 웹 서비스 동기화

문제점	저장 된 승객의 얼굴이미지 파일을 업데이트 시, 사진이미지가 바로 동기화가 되지 않는 문제 발생
해결방법 및 기술	웹페이지 검색기록 및 쿠키제거 후 웹 서비스 동기화문제 해결

#### ○ HW 문제점 및 해결방안

##### 1) 스텝모터 연동 문제

장애요인	스텝모터(28BYJ-48)는 서버와 연동 시 흔들림이 있고, 회전속도가 현저히 느려지는 문제점 발생
해결방법 및 기술	성능이 더 좋은 스텝모터(NEMA-17)로 교체

##### 2) 아두이노 전력 공급 문제

장애요인	정상적인 코드 컴파일 후 스텝모터 실행 시 전력문제로 인해 모터의 작동이 안되는 문제점 발생
해결방법 및 기술	기존 UNO의 경우 5V 이상 사용할 수 없지만 6-12V사용 가능한 외부전압 사용하여 문제 해결



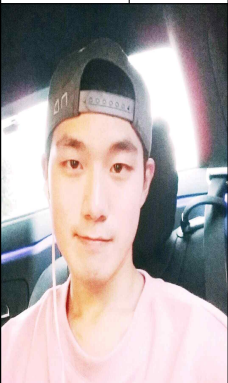
### 3) 카메라 연동 문제

장애요인	IP카메라 사용시 스트리밍으로 화면을 받기 때문에 불러올지 끊김 현상이 발생하며, 이미지 처리하는데에 문제점 발생
해결방법 및 기술	WEB 카메라로 장비 교체후 서버와 카메라간 이미지 처리 직접하게 함으로써 문제 해결

### 4) 안드로이드 SMS 처리문제

장애요인	USIM 카드 없이 SMS 서비스 사용 시 서비스 사용 불가
해결방법 및 기술	등록되어있는 USIM칩 삽입 후 실행 시 정상 작동함

#### 4. 프로젝트를 통해 배우거나 느낀 점

팀원	배우거나 느낀점
 <div data-bbox="209 712 437 752"> <div>김재민</div> <div>팀장</div> </div>	<p>이번 프로젝트의 팀장을 맡게 되어 처음에는 부담감이 컸지만 프로젝트가 진행됨에 따라 점점 배우고 느낄 수 있는 점이 많아졌다. 특히 각 팀원들의 의견을 잘 조율하고 계획을 짜는 것에서 약간이나마 리더십을 배울 수 있었고, 각자의 파트를 나누는 부분에서 책임감과 협동심을 배울 수 있었다. 앞으로 해 나갈 일들에 이번 경험이 많은 도움이 될 것이다.</p>
 <div data-bbox="209 1133 437 1173"> <div>정상수</div> <div>팀원</div> </div>	<p>평소에 개인의 역량을 중시하며 다른 사람들에게 맡기기 보다는 스스로 하는 편을 좋아했지만 프로젝트를 진행함에 따라 팀원간의 협력과 조화가 굉장히 중요하다는 것을 알게되었다. 팀 전체의 발전을 좀 더 생각하게 되었고 자연스럽게 협동심을 배웠다. 팀원들과 깊이 친해질 수 있는 계기가 되었고, 또한 나의 공부도 더욱 깊이있게 할 수 있어 좋았다.</p>
 <div data-bbox="209 1554 437 1594"> <div>이동열</div> <div>팀원</div> </div>	<p>프로젝트를 진행하면서 팀원들과의 협동심과 분위기 조성이 팀 전체에게 얼마나 많은 영향을 미치는지 알게 되었다. 의견을 제시하고 아이디어를 내는 과정에서 강한 팀은 어떤 말도 허투루 듣지 않고 서로 시너지 효과를 내며 커 나간다는 것을 알게 되었다. 한이음 프로젝트와 같은 좋은 공모전을 만나 좋은 사람들을 만날 수 있어 좋았고 나 자신에게도 이번 공모전이 많은 도움이 되었고, 지금까지 내가 대학에서 배운 내용들을 종합적으로 사용해 볼 수 있었던 좋은 기회가 되었다. 나에게는 잊지 못할 경험이 된 것 같다.</p>
 <div data-bbox="209 1975 437 2011"> <div>전진우</div> <div>팀원</div> </div>	<p>프로젝트 진행이 처음이어서 '내가 과연 할수 있을까'라는 생각이 몇 번이나 들었지만 팀원들과 여러번의 회의를 거치며 그러한 생각이 자연스럽게 없어졌다. 팀원들과 각자 파트를 분담하면서 내가 맡은 부분을 열심히 하다보면 어느새 목표에 한걸음씩 다가가고 있음을 느꼈다. 모르는 것은 물어보며, 내가 도와줄 수 있는 부분은 있는 힘껏 도와주며 모두가 함께 노력하는 모습을 보며 이번 프로젝트 뿐만 아니라 다른 일을 할때에도 이번 경험이 크나큰 도움이 될 것임을 깨달았다. 또한 멘토님과의 주기적인 만남을 통해 여러 가지로 건문을 넓힐 수 있어서 좋은 기회가 되었다.</p>

## IV. 작품의 기대효과 및 활용분야

### ※ 평가항목 : 기획력 (활용가능성)

#### 1. 작품의 기대효과

##### ○ 승객에게 편의성 제공

- 얼굴인식을 통해서 미리 데이터베이스에 등록된 이용자의 얼굴과 비교하여 직접 층수를 누르지 않아도 일치하는 승객의 층수를 자동적으로 눌러준다. 이로 인해 손에 무거운 짐을 들고 있거나 키가 작은 아이들, 몸이 불편한 사람의 경우 직접 버튼을 누르지 않아도 엘리베이터를 작동시킬 수 있다. 또한, 다수의 사람이 동시에 엘리베이터를 탑승했을 시에 불필요한 접촉 없이 쾌적한 엘리베이터 사용이 가능하다.

##### ○ 시장성

- 최근 엘리베이터 시장에는 홍채인식이나 동작감지 기술을 접목시킨 다양한 스마트 엘리베이터가 등장하고 있다. 그러나 이러한 엘리베이터의 경우 값비싼 센서들을 사용하기 때문에 실용화되기 어렵다는 문제점이 있다. 하지만 이 제품의 경우, 기존 엘리베이터에서 사용하는 CCTV의 기능을 수행함과 동시에 본 프로젝트에서 제공되는 얼굴인식, 음성인식을 활용한 서비스의 이용이 가능하다.
- 글로벌 엘리베이터 시장전문 조사기관은 2015년부터 2021년까지 엘리베이터 연평균 시장성장률이 6.01%로, 2021년에 약 1,252.2 억 달러 (한화로 약 150조 원)에 이를 것으로 전망했다.
- 현재 국내 엘리베이터의 신규설치 수는 매년 꾸준한 증가세를 보이고 있다. 따라서, 본 프로젝트는 기존 엘리베이터에 많은 추가 비용없이 설치가 가능 하므로 충분히 시장성이 있다고 생각한다.

■ 국내 승강기 신규설치 추이 (꾸준한 증가세)



\* 자료 = 한국승강기안전관리원, 현대엘리베이터, 한양증권 리서치센터

## 2. 작품의 활용분야

### ○ 추후 창업가능성

- 엘리베이터 업체들과의 제휴를 통해 창업가능성을 열어둔다. 업체와의 협력을 통해 기존 엘리베이터의 CCTV 카메라에 본 프로젝트의 서비스를 접목시켜 본 프로젝트의 서비스를 사용하는 업체의 경쟁력을 높임과 수익적인 증가도 기대할 수 있을 것이다. 동시에 프로젝트 개발자는 서비스를 제공 후 서버와 데이터베이스 관리 등 유지보수를 해주며 수익을 올릴 수 있다.

### ○ 다양한 분야 접목

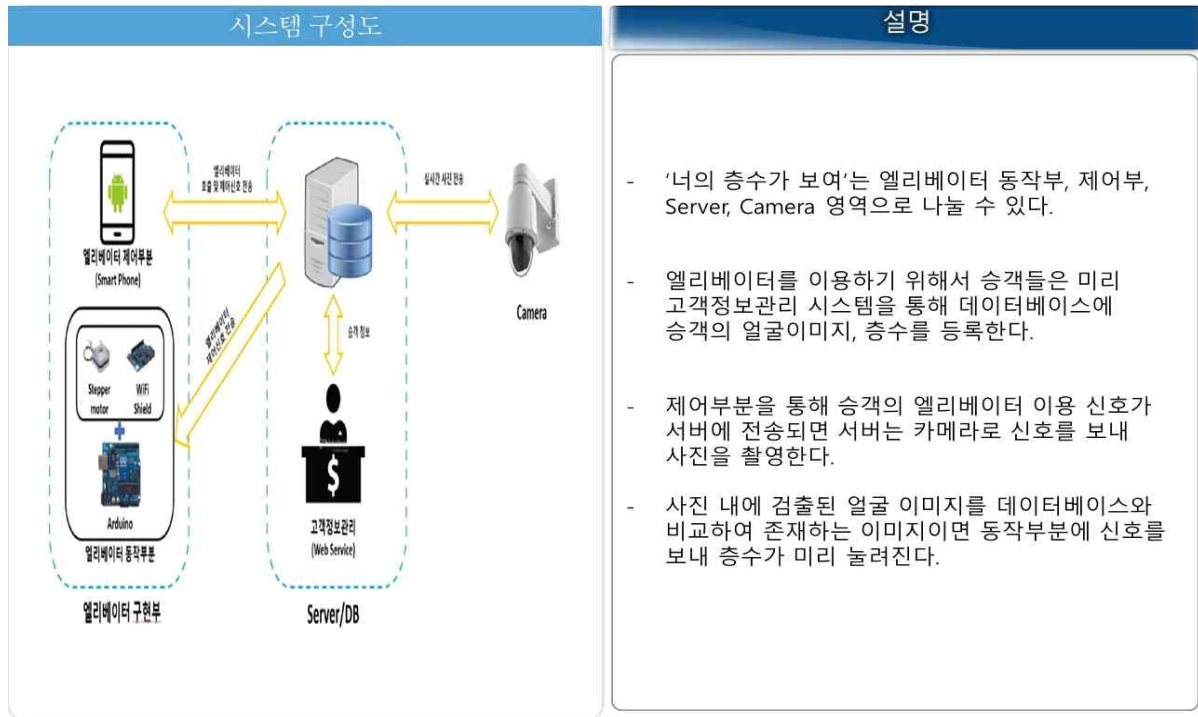
- 엘리베이터 뿐만 아니라 얼굴인식 기술을 활용하여 여러 분야에 접목시킨다. 예를 들어 학교나 회사 내에서 인식된 인물이 미리 등록된 인물인지 아닌지를 구분하여 출퇴근 시간을 확인하는 등 여러 가지 분야에 활용할 수 있다.



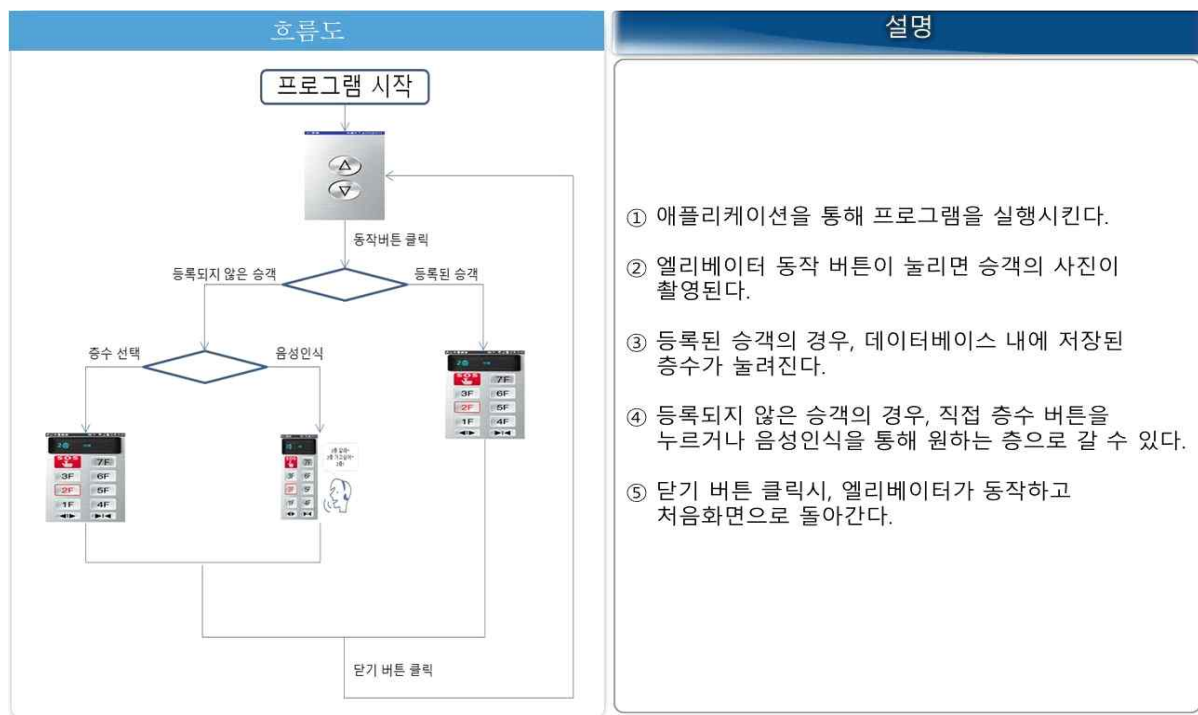
## V. 개발산출물

### ※ 평가항목 : 평가 전반에 참고

#### ○ 시스템 구성도



#### ○ 흐름도



○ 메뉴 구성도



## ○ 주요 소스코드

### <얼굴 검출코드>

프로그램 ID	pictureTOface	프로그램 명	얼굴 검출	작성일	2017. 08. 30	Page	1/4
개요	이미지에서 얼굴을 검출하기 위해 사용하기 위한 코드이다.					작성자	AdvICE

#### 상세 로직

```
public static int pictureTOface(String address) { //전체사진을 얼굴사진만 추출해서 저장
    int detectingFace = 0; // 사진에서 잡힌 얼굴 갯수정수값.
    System.loadLibrary("opencv_java2412"); // 라이브러리 로드.
    CascadeClassifier faceDetector = new CascadeClassifier(
        "C:\\Users\\Wice305\\Downloads\\opencv\\build\\x64\\share\\opencv\\data\\haarcascade_frontalface.xml");
    int imageNumber = 0;
    while (true) {
        Mat Mat_image = Highgui.imread(address + "\\PicturesTakenFromElevator.png");
        // 이미지에서 얼굴검출
        MatOfRect faceDetections = new MatOfRect();
        faceDetector.detectMultiScale(Mat_image, faceDetections);
        detectingFace = faceDetections.toArray().length;
        Mat temp = new Mat();
        // 얼굴만 검출
        for (Rect rect : faceDetections.toArray()) {
            temp = new Mat(Mat_image, rect);
            imageNumber++; // 디텍팅 갯수만큼 저장하기 위해서.
            String filename = "detectingFace" + imageNumber + ".png";
            Highgui.imwrite("C:\\Users\\Wice305\\Desktop\\server\\" + filename, temp);
        }
        break;
    }
    return detectingFace; // 검출된 얼굴의 수를 리턴한다.
}
```

### <회색필터 적용>

프로그램 ID	grayscale	프로그램 명	회색필터 적용	작성일	2017. 08. 30	Page	2/4
개요	인식률을 높이기 위해 회색필터를 적용하는 코드이다.					작성자	AdvICE

#### 상세 로직

```
public BufferedImage grayscale(BufferedImage img) { // 회색필터 적용
    for (int i = 0; i < img.getHeight(); i++) {
        for (int j = 0; j < img.getWidth(); j++) {
            Color c = new Color(img.getRGB(j, i));

            int red = (int) (c.getRed() * 0.299);
            int green = (int) (c.getGreen() * 0.587);
            int blue = (int) (c.getBlue() * 0.114);

            Color newColor = new Color(red + green + blue, red + green + blue, red + green + blue);

            img.setRGB(j, i, newColor.getRGB());
        }
    }
    return img;
}
```

## <얼굴 비교 코드>

프로그램 ID	ComparePicture	프로그램 명	얼굴 이미지 비교	작성일	2017. 08. 30	Page	3/4
개요	얼굴 이미지를 비교하기 위해 사용하는 코드이다.					작성자	AdvICE

### 상세 로직

```
public static int ComparePicture(String address1, String address2) { //얼굴 이미지 비교분석
    int resultRatio = 0;
    System.loadLibrary("opencv_java2412"); // 라이브러리 로드.

    String picture = address1; // 비교대상
    String basePicture = address2; // 비교기준

    Mat objectPictureImage = Highgui.imread(picture, Highgui.CV_LOAD_IMAGE_COLOR); // 이미지 로드실행
    Mat objectBasePictureImage = Highgui.imread(basePicture, Highgui.CV_LOAD_IMAGE_COLOR); // 이미지를 행렬로 만든다.

    MatOfKeyPoint objectKeyPoints = new MatOfKeyPoint(); // 행렬에 키포인트를 저장할 객체생성
    FeatureDetector featureDetector = FeatureDetector.create(FeatureDetector.SURF); // 특징점 알고리즘

    //키포인트 감지 실시!!
    featureDetector.detect(objectPictureImage, objectKeyPoints); // 비교대상 이미지 키포인트 감지(행렬이용)

    MatOfKeyPoint objectDescriptors = new MatOfKeyPoint();
    DescriptorExtractor descriptorExtractor = DescriptorExtractor.create(DescriptorExtractor.SURF);
    descriptorExtractor.compute(objectPictureImage, objectKeyPoints, objectDescriptors);

    // 비교이미지와 기준이미지를 비교 실시.
    MatOfKeyPoint sceneKeyPoints = new MatOfKeyPoint();
    MatOfKeyPoint sceneDescriptors = new MatOfKeyPoint();
```

프로그램 ID	ComparePicture	프로그램 명	얼굴 이미지 비교	작성일	2017. 08. 30	Page	4/4
개요	얼굴 이미지를 비교하기 위해 사용하는 코드이다.					작성자	AdvICE

### 상세 로직

```
featureDetector.detect(objectBasePictureImage, sceneKeyPoints); // 기준 이미지의 주요 지점 감지
descriptorExtractor.compute(objectBasePictureImage, sceneKeyPoints, sceneDescriptors); // 기준 이미지의 설명자 계산

List<MatOfDMatch> matches = new LinkedList<MatOfDMatch>();
DescriptorMatcher descriptorMatcher = DescriptorMatcher.create(DescriptorMatcher.FLANNBASED);
descriptorMatcher.knnMatch(objectDescriptors, sceneDescriptors, matches, 2); // 비교대상 및 기준 이미지 매칭시키기

// 매칭 계산
LinkedList<DMatch> goodMatchesList = new LinkedList<DMatch>();

float nndrRatio = 0.7f;

for (int i = 0; i < matches.size(); i++) {
    MatOfDMatch matofDMatch = matches.get(i);
    DMatch[] dmatcharray = matofDMatch.toArray();
    DMatch m1 = dmatcharray[0];
    DMatch m2 = dmatcharray[1];

    if (m1.distance <= m2.distance * nndrRatio) {
        goodMatchesList.addLast(m1);
    }
}
return resultRatio;
}
```