**Software Capstone Design**

**Final repot**

**오늘의 집사**



**Team. Twenty Six + 1**

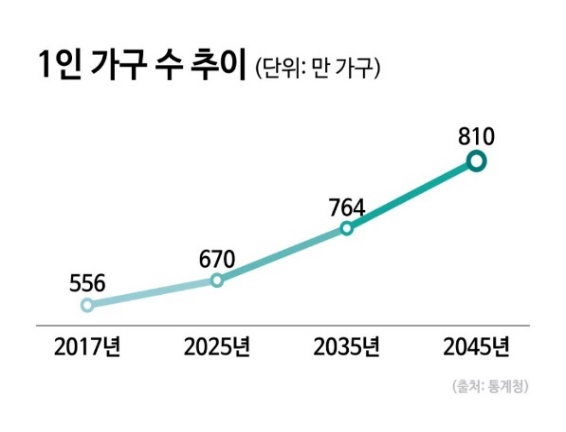
**Team member.**

|  |  |  |
| --- | --- | --- |
| **김형직** | **201120874** | **소프트웨어학과** |
| **권혁정** | **201120960** | **소프트웨어학과** |
| **유진성** | **201120844** | **소프트웨어학과** |
| **오동성** | **201120913** | **소프트웨어학과** |
| **김다미** | **201220951** | **소프트웨어학과** |

**초록(abstract)**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **과제 명** | (국문) 독신 여성을 위한 종합 보안 솔루션 – 오늘의 집사 | | | | |
| **연구기관** | 아주대학교 Twenty Six + 1 | | **총괄 책임자** | 아주대학교 소프트웨어 | |
| **참여기업** | ? | | (성명)윤 대균 | |
| **총 개발비** | 총계 | ??? | **총 연구기간** | ???????? | |
| 재료 구입 | ?? | **총 참여 연구원 수** | 총 인원 | 5 |
| 기타 등등 | ?? | 내부 인원 | 4 |
| 개인 부담금 | ? | 외부 인원 | 1 |
| **# 연구 개발 목표**   * 혼자 사는 여성의 안전을 지킬 수 있는 IOT 기반의 보안 솔루션을 제공하며 이를 원격으로 손 쉽게 이용 가능한 모바일 서비스를 제공한다.   **# 연구 개발 내용**   1. mimic voice : 사용자가 단어를 이야기하면 상황에 맞는 답변을 남성 목소리로 출력한다. 2. Control door lock : 사용자가 외출했을 시, 비밀번호를 노출하지 않고 원격으로 도어락을 열고 닫을 수 있다. 3. Streaming : 하나의 카메라의 각도를 제어해 창문과 현관문의 실시간 상황을 확인할 수 있는 cctv 기능을 지원한다. 4. Door detect : 거수자의 현관 침입을 탐지 하면 사용자에게 푸시 알람 화면이 뜬다. 해당 알람화면에는 현관 쪽 cctv 영상 확인 기능과 침입자가 창문을 바라보면 얼굴을 인식해 사진을 찍고 저장한 뒤 사용자가 확인하는 기능, 위치정보를 포함한 sms로 경찰에게 신고 하는 기능, 경보 음과 최루가스 살포를 포함한 경보 조치 on/off기능을 지원한다. 5. Window detect : 창문에서 거수자가 바라보면 이를 탐지하여 사용자에게 거수자의 얼굴을 인식하여 사용자에게 사진이 담긴 푸시 알람 화면을 제공한다.   **# 핵심 기술 요소**   * 핵심 기술 요소에는 | | | | | |

1. **개요**
   1. **문제 정의**



1인 가구 인구가 증가함에 따라 2045년에는 국내 1인 가구 수가 전체 인구의 36.3%인810만 가구가 될 것으로 전망되고있다.

1인 가구 수의 증가는 독신 여성의 증가로 이어진다. 정부는 독신 여성을 위해 ‘여성안심귀가 스카우트’, ‘여성 안심 택배’, ‘여성 안심 주택’ 등 각종 지원 정책들을 지원하고 있지만 현실적인 대응책으로는 부족하다. 또한 독신 여성을 대상으로 한 범죄가 지능화 및 다양화되고, 사회적 범죄가 늘어남에 따라 홈 시큐리티 시스템에 대한 관심이 증가하고 있다.

홈 시큐리티 시스템은 범죄나 불법적 침입을 막고 거주자의 사생활을 보호하기 위한 안전장치이다. 본 프로젝트에서는 IoT 기술을 이용하여 도어락 , CCTV, 스피커 등 각종 디바이스를 통한 홈 시큐리티 시스템을 제공할 것이며, 모바일 어플리케이션을 통하여 해당 디바이스들을 원격으로 제어할 수 있는 기능을 제공하고자 한다.

* 1. **. 타겟**
     1. **user**

유저는 크게 독신 여성과, 자취를 하고 있는 대학생으로 나눌 수 있다. 독신 여성들에게 집은 외부와 차단된 자신만의 공간이다. 이 공간에서 편안함과 안전함을 동시에 느낄 수 있어야 한다. 하지만 원룸이나 오피스텔 등 공동주택에 거주하는 경우 택배 배달, 노크 소리, 유리창 사이로 훔쳐보는 범죄행위 등 여러 방해요건으로 인해 안전하다고 느끼기 힘들다는 문제점이 있다. 또한 집을 비웠을 때 외부인의 침입이 있을 시 감지할 수 있는 방법이 없어 범죄에 노출될 수 있다. 따라서 독신 여성들은 본 프로젝트를 통해 외부의 여러 방해요소들을 사전에 차단하여 불안감을 해소할 수 있다.

자취생의 경우 본인들이 외부에 있을 때 지인들의 자취방 방문이 있을 경우가 있다. 전화를 통해 도어락 비밀번호를 알려주게 되면 상대방이 되묻는 과정에서 비밀번호 유출의 가능성이 있고, 다른 매체를 통해 알려주게 되더라도 유출 가능성 또는 상대방이 자신의 비밀번호를 기억하고 있기 때문에 비밀번호를 재설정하는 경우가 생길 수 있다. 그런 사용자들은 본 시스템의 도어락 원격 제어 기능을 통해 고민을 해소할 수 있을 것이다. 본 시스템은 플러그인 방식이기 때문에 도어락 제어 기능만을 이용한다면 전체 시스템을 구입하는 것보다 훨씬 저렴한 가격으로 만족도를 얻을 수 있다.

* + 1. **stake holders**

본 시스템을 사용함으로써 사용자 이외에도 보호자 또한 심리적 안정을 얻을 수 있다. 또한 ADT, SECOM 등 보안 업체와의 업무 제휴를 맺어 보안 업체의 신뢰도를 향상시킬 수 있다. 마지막으로 경찰의 경우 CCTV를 통한 영상 확보로 범인을 파악하고 수사하는 데 도움을 얻을 수 있다. 또한 신고로 들어온 문자에 적힌 주소와 시간 통해서 범죄에 대한 분석자료로 사용할 수 있을 것이다.경찰은 분석된 자료를 토대로 우범지역, 범죄가 자주 일어나는 시간에 순찰을 강화하여 치안을 향상시킬 수 있다. **(형직수정)**

(( 교수님 피드백 \_\_ 또한 경찰 등 다른 stake holder들을 본 서비스에서 어떻게 취급하고 운영할지 설명이 되어 있지 않음 (-1)))

* 1. **가치**
     1. **안전한 집이라는 인식 제공**

시스템은 사물인터넷과 보안 기술을 융합하여 사용자의 안전성을 높인다. 외부로부터의 침입이나 염탐, 택배기사, 초인종 등 실시간으로 응답해야 하는 상황에서 솔루션을 제공한다.

* + 1. **스마트폰을 이용한 관리 및 제어**

마이크로 프로세서는 집에 설치된 센서들과 연결되어 실시간으로 데이터를 수집한다. 또한 서버를 통해 스마트폰에 설치된 어플리케이션에 데이터를 전달한다. 따라서 사용자는 스마트폰으로 언제 어디서든 집 내부의 상태를 확인할 수 있고, 카메라, 도어락, 남성 음성시스템을 제어할 수 있다.

* 1. **핵심 기술 요소**
  2. **목표**
     1. 핵심 목표

오늘의 집사의 핵심 목표는 범죄에 노출되기 쉬운 혼자 사는 여성을 보호 및 방지하는 것이다. 그 시작으로 혼자 사는 여성의 집에서 일어나는 각종 범죄 사례를 분석했다. 그 결과 침입에 무방비한 경우가 많다는 점과 범죄자들이 여성이 혼자 사는 걸 알게 되었을 때 범죄가 일어난다는 점을 주목하였다. 따라서 목표를 달성하기 위해 사용자의 집에 침입이 감지되었을 때 사용자에게 즉각적으로 알림과 함께 경찰에게 신고할 수 있고 남자 목소리가 들리게끔 하여 혼자 사는 것처럼 여겨지지 않도록 만들 것이다.

* + 1. 개발 목표 **(형직수정) 추천받음**
       1. 남성 음성 서비스

독신 여성이 혼자 사는 것을 들키지 않기 위해 네이버 Clova 음성 인식 api를 이용해 남자의 음성을 들려주는 기능을 구현한다. 여성이 특정 키워드를 말하면 clova 음성 인식 api가 여자의 말을 분석하고 그에 맞게 저장된 멘트를 남자의 목소리로 들려준다. 특정 키워드는 사용자가 추가할 수도 있고 삭제할 수도 있도록 한다.

* + - 1. 침입자 및 거수자의 얼굴 인식 사진

침입이 탐지되었거나 거수자가 창문을 통해 염탐을 하려고 할 때 Open CV로 구현한 얼굴 인식 기능이 탑재된 카메라를 통해 사진을 찍는다. 이 사진을 사용자의 어플리케이션에 보내고 사용자는 사진을 보고 경찰에 신고할지 말지 여부를 결정한다.

* + - 1. 카메라 제어 및 스트리밍

카메라 한대로 집내부와 창문을 확인하기 위해 카메라가 360도 회전이 가능하도록 만든다. +@

* + - 1. REST한 웹 서버 구현

REST기반으로 하는 웹 서버를 구현하여 라즈베리파이나 안드로이드 어플리케이션이 쉽게 접근이 가능하도록 만든다. 회원가입, 로그인 인증, 푸쉬메세지 요청, 라즈베리파이 등록 및 아이피 수정 등의 기능을 가지며 post방식으로 구현한다. 모든 기능의 응답 메시지 포맷은 json으로 한다.

1. **배경**
   1. **기존 사례(-> 유사제품으로 바꿀까?)**
      1. **삼성 스마트싱스 시큐리티 서비스**

삼성전자는 미국 최대 주택 보안기업인 ADT와 협업하여 smart-thinks home security service(이하 스마트씽스)을 제공하고 있다. 시스템은 각종 센서들과 디바이스, 패널로 이루어져 있으며 패널로 현관문 안전장치, 초인종, 온도조절장치, 전구와 같은 기기를 제어하 고 모니터링이 가능하다. 스마트씽스 모바일 어플리케이션 으로는 해당 디바이스들과 홈 시큐리티 센서를 관리할 수 있으며, 집안 안전을 확인하고 알람을 받을 수 있다. 또한 보안업체인 ADT를 통해 월 25달러의 이용료로 24시간 모니터링 서비스를 제공받을 수 있다. 63만원 정도의 시큐리티 스타터 킷에는 시큐리티 허브와 문, 창문 탐지기, 움직임 탐지기가 들어있고 추가로 23만원을 내고 구입할 수 있는 확장 팩은 불, 일산화탄소, 물 탐지 센서를 통한 안전 시스템을 제공한다.

**한계점 : 높은 가격, 보안에 필요 없는 기능도 제공, 영상감시 체계 미비**

* + 1. **세콤 이지**

 세콤 이지는 IP 카메라에 UWB (초광대역 레이더) 센서를 결합해 영상 확인과 침입자 감지를 동시에 하는 IoT 홈 CCTV 제품이다. 스마트폰 어플리케이션을 통해 모니터링 및 방범 서비스를 이용할 수 있으며, 블루투스 기능을 이용하여 사용자의 휴대폰과 통신하여 외출 시 자동 경비모드 귀가 시 경비모드 자동 해제를 통한 사생활 보호 가능을 가지고있다. 주요 서비스로는 침입감지, 비상통보, 영상 모니터링, 사생활보호, 영상 캡쳐가 있다. 25만원 1회 구입으로 월 이용료가 없이 반영구적으로 사용할 수 있는 제품이다.

**한계점 : CCTV 기능만 수행, 높은 가격**

* 1. **적용 기술((???))**
     1. **기초 지식(보니까 아마존 웹서비스 이런거 넣음)**
        1. **아두 이노**
        2. **라즈 베리 파이((동성 추가))**
        3. **안드로이드**

안드로이드(android)란 휴대폰용 운영체제, middle ware, 응용 프로그램을 한데 묶은 소프트웨어 플랫폼이다. 안드로이드는 개발자들이 자바 언어로 응용 프로그램을 작성할 수 있게 하였으며 compile 된 바이트 코드를 구동할 수 있는 라이브러리를 제공한다. 안드로이드 소프트웨어 개발 키트인 SDK는 웅용 프로그램을 개발할 때 필요한 각종 도구와 API를 제공한다.

* + - 1. **아마존 웹 서비스(형직 수정)**

아마존 웹 서비스(aws)는 온라인 서점으로 유명한 아마존에서 만든 클라우드 서비스 플랫폼으로서, 컴퓨팅 파워, 데이터베이스 스토리지, 콘텐츠 전송 및 기타 기능을 제공하고 있다. EC2(Elastic Computer Cloud)로 서버로 사용할 컴퓨터를 생성하고 S3(Simple Storage Service)라는 파일 저장공간을 이용해 서버에 파일을 저장하고 다운로드 받을 수 있다. 또한 RDS(Relational Database Service)를 통해 서버에서 관계형 데이터베이스를 사용 할 수 있다. 우리 프로젝트에 사용한 Elastic Beanstalk는 PaaS(Platform as a Service)로 웹 어플리케이션을 빌드/배포하는데 필요한 환경을 제공해주는 PaaS이다. EB의 장점으로는 한 번의 명령으로 모든 EC2 인스턴스에 있는 어플리케이션 서버를 빠르게 재 시작할 수 있어 새로운 어플리케이션 버전을 쉽게 배포가 가능하다. 따라서 개발자는 어플리케이션 개발에 집중할 수 있다는 장점이 있다. 또한 어플리케이션 운영에 사용된 AWS 리소스에 대한 비용만 발생하고 Beanstalk 자체에 대한 사용 요금은 없기 때문에 비용적 부담이 적다는 장점이 있다.

* + - 1. **근거리 무선 통신( 소켓통신으로 변경?((동성))-우리 안드로이드랑 라즈베리파이 그리고 라즈베리 파이랑 아두이노랑 어떻게 통신하지 정확히..? )**
      2. **음성인식 및 자연어 처리 기술**

과거 음성인식의 역할은 소리를 문자로 바꾸어주는 것에 한정되어 있었으나 현재는 사람과 기계간의 대화를 가능하게 하는 자연어 처리(natural language processing) 기술 까지도 그 기술적 영역 안에 포함한다. 현재 음성 인식 기술은 단순히 사용자의 음성을 알아듣는 정도가 아니라 목소리를 분석하는 과정에서 주변 환경, 상황 등을 모두 파악하는 단계까지 진화했다. 특히 ‘집’이라는 환경에서는 스마트폰이나 키보드 같은 인터페이스보다는 음성인식 방식이 더 자연스러운 인터렉션 방식이므로 음성인식기술은 스마트 홈 확산을 위한 핵심 기술이 될 것으로 보여진다. 클라우드 기반의 음성인식 기술인 ‘클라우드 스피치 API, 지능형 개인 비서 역할을 수행하는 애플 iOS전용 소프트웨어 ‘Siri’가 대표적이다.

* + - 1. **이미지 인식 기술**

이미지를 인식하고 그 안에 담긴 정보를 분석함으로써 사용자들이 원하는 정보를 빠르고 정확하게 찾을 수 있는 기술이다. 이미지 인식 기술은 사람의 눈과 같은 역할을 하는 기술로서 글을 읽고, 사람을 구분하고, 색을 추출, 상품을 파악하는 등 다양한 정보 습득을 위해 필요한 기술이며 이미지가 내재하고 있는 의미 있는 정보들을 빠르고 효율적으로 찾아내는 것이 핵심이다. 사람의 얼굴을 통해 나이, 성별, 감정 등을 파악하는 기술, 동작인식 기술, 텍스트 인식 기술 등이 있다. 현재 보안, 개인화, 정보 분류 등의 분야에 활용되고 있다. 구글의 클라우드 비전, 웹 캠을 통해 얼굴을 인식하고 실시간으로 감정을 분석할 수 있는 CARA 등이 있다.

* + 1. **프로젝트 관련 학과목 (형직수정) 출처 요람2017으로 하면 될 듯**
       1. **컴퓨터 네트워크**

본 과목은 TCP/IP 프로토콜을 중심으로 데이터링크, 네트워크, 트랜스포트, 어플리케이션 계층의 표준 인터넷 프로토콜을 이해한다. 여기에는 ARP, IP, RIP, ICMP, TCP, UDP에, TELNET, FTP, HTTP, SMTP, 및, DNS 등의 프로토콜이 포함되며, 이들 프로토콜 각각의 동작과 함께 상호 간의 작용을 종합적으로 공부한다. 이 과목의 목적은 인터넷 구조, 프로토콜 및 관련 어플리케이션에 대한 전반적인 이해를 도울 것이다. 계층적 구조를 기반으로 컴퓨터 네트워킹을 이해하고 지금까지 인터넷을 사용자 입장에서 사용하였지만 실제적으로 어떻게 인터넷이 동작되는가를 배울 것이다.

* + - 1. **시스템 프로그래밍**

본 교과목에서는 시스템 소프트웨어를 구성하는 어셈블러, 링커, 로더, 운영체제 디바이스 드라이버에 관한 이론적 사항을 공부하고 이들을 조합하여 설계할 수 있는 능력을 키운다. 즉, 어셈블리 언어를 익힌 후 이를 이용한 여러 프로그래밍 기법을 터득하고, 링커 및 로더에 대한 기초적인 개념 및 설계방법을 파악하도록 한다. 본 과목을 수강한 학생들은 시스템 프로그램 설계에 필요한 목표설정, 분석, 제작, 시험평가 능력을 배양한다. 본 과목은 체계적이고 실제적인 실습 교육을 병행한다. 어셈블러, 로더, Cross compiler, 운영체제 디바이스드라이버 등 다양한 시스템 소프트웨어를 임베디드 리눅스 시스 템 환경에서 직접 구현해 봄으로써 임베디드 시스템 환경에 대한 기초 지식과 실무 경험을 갖추도록 한다.

* + - 1. **오픈소스 SW 입문**

본 교과목은 오픈소스 SW 개발자로의 기초 역량 양성을 목표로 한다. 이를 위해 오픈소스 SW의 정의 및 철학에 대해 학습하고, 오픈소스 SW를 활용한 다양한 서비스와 플랫폼 사례들을 조사하며, 이를 통해 오픈소스 SW 개발의 의미를 이해한다. 오픈소스 SW 라이센스와 소스 코드 버전 관리, 버그 트래킹과 같은 오픈소스 SW 개발에 필요한 지식을 학습하며, 이슈등록, 코드리뷰 등 오픈소스 SW 개발 프로세스에 따라 오픈소스 SW에 기여하는 활동에 대해 경험해 봄으로 오픈소스 SW 개발자로의 기초 역량을 배양한다. 본 교과목을 시점으로 하여 학생들은 추후 개설되는 교과목에서 오픈소스 SW 프로젝트를 추진하게 되고 기여자로 서의 역량을 배양하게 될 것이다.

* + - 1. **객체 지향 프로그래밍**

객체지향 프로그래밍 언어의 기본 구성 요소인 객체와 클래스, 상속 등의 개념과 이들을 활용하여 프로그래밍 하는 기법을 배운다. 객체지향프로그래밍의 핵심 개념인 데이터 추상화, 캡슐화, 다형성 등을 공부하고, 이들로 하여금 객체지향 프로그램이 어떻게 소프트웨어의 재사용과 확장성을 제공하는지에 대해서도 공부한다. 이 과목에서는 가장 널리 사용되는 객체지향 언어 중의 하나인 C++ 혹은 Java 언어를 이용하여 이러한 객체지향적인 개념을 공부한다. 학생들은 선호도에 따라 언어를 선택하여 수강할 수 있다.

* + - 1. **웹 시스템 설계**

본 과목에서는 분산시스템의 대표적인 사례인 인터넷을 기반으로 소프트웨어 시스템에 대한 이해를 넓히고 가장 보편적인 표준인 웹 기술을 이용하여 원하는 어플리케이션을 “쉽게”, “빨리”, 그리고 “많은 사람이 사용”할 수 있는 형태로 제작할 수 있는 능력을 함양하는 것을 목표로 한다. 이를 위해 웹 구축을 위한 클라이언트 사이드 프로그래밍 언어로 HTML5 및 스크립트 언어의 개념과 문법, 활용지식들을 습득하며, 동적인 웹페이지 설계를 위한 Servlet, JSP 등의 서버사이드 프로그래밍 활용에 대해 학습한다. 학습 된 세부 지식을 기반으로 Ajax 스타일등의 최신 인터넷 어플리케이션 제작 및 클라우드 컴퓨팅과 관련된 실습 프로 젝트를 수행한다. 본 과목에서 학습된 서버 사이드 프로그래밍 기술은 독립적인 앱 형태의 어플리케이션 제작에 응 용하는 형태 등으로 다양하게 활용이 된다.

* + - 1. **데이터베이스**

데이터베이스는 방대한 크기의 데이터를 컴퓨터에 저장하고 필요한 데이터를 얻어오는 기법에 대한 것으로 컴퓨터 응용프로그램에서 컴퓨터 시스템에 이르기까지 영향을 미치는 전산학에서 매우 중요한 분야이다. 본 교과에서는 데이터베이스 전반에 대한 이해를 목표로 하며 특히 사용자 입장에서 데이터베이스 시스템을 다루는 것에 중점을 둔다.

* + 1. **활용 플랫폼/오픈 소스**

|  |  |  |  |
| --- | --- | --- | --- |
| client | server | tool | Language |
| -  Android | - AWS  - Linux ubuntu | -Eclipse  -Android studio  -Arduino sketch | -Java  -C  -Python |

|  |  |
| --- | --- |
| Platform / Open source | Description **(형직수정)** |
| MjpgStreamer | 오픈소스 IP기반 JPEG파일 스트리머 |
| OpenCV | 오픈소스 컴퓨터 비전 라이브러리 |
| Naver Clova Speech Synthesis | 한국어 음성인식 API |
| Ruby on Rails | Ruby 기반 웹 어플리케이션 개발 프레임워크 |
| Devise | Rails 기반 인증 솔루션 오픈 소스 |
| Carrierwave | Rails 기반 파일 업로드 솔루션 오픈소스 |
| Firebase Cloud Messaging | 교차 플랫폼 메시징 솔루션 |
| AWS Elastic Beanstalk | 빌드/배포 환경을 제공해주는 PaaS (Platform as a Service) |
|  |  |

Framework 활용

**((형직 – 데이터 베이스, 서버 부분 수정 부탁))**

* 1. **제한 요소 분석(없는 조도 있음, 길게 안써도 될 듯?)**
     1. **기술적 제약(일단은 기존의 것 들 다 넣어봤음)**

‘오늘의 집사'에서 제공하려는 기능을 구현하기 위해서는 몇 가지 제약사항이 발생 할 수 있다. 프로젝트를 설계하는데 나타나는 제약 사항은 다음과 같다.

* + - 1. **카메라 기능 – Always On**

창문에 설치된 카메라는 거수자의 얼굴을 인식하여 촬영하고 출입문의 카메라는 예기치 못한 상황에서 외부 침입자로부터 침입이 발생했을 경우 외부 침입자를 촬영한다. 창문이나 출입문에 설치된 카메라의 기능 발생 조건은 항시 가 아니라 간헐적으로 발생한다. 즉, 기능 발생 조건이 언제 인지 예측 할 수 없기 때문에 카메라가 항상 촬영하여 데이터를 User Interface에 전송하지는 않으나 카메라는 항상 켜져 있어야 한다.

* + - 1. **카메라 기능 – Face Recognition**

창문에 설치된 카메라는 거수자의 얼굴을 인식하고 일정 시간 이상 얼굴이 인식되는 조건을 만족 해야만 촬영을 하게 된다. 낮일 경우에는 문제가 발생하지 않지만 밤일 경우에는 Noir 파이카메라 사용에도 불구하고 거수자의 얼굴을 인식하는데 어려움이 존재했다. 또한 얼굴 전체를 드러내지 않거나 얼굴을 인식하기 힘들게 하는 도구를 사용하여 얼굴 인식 기능이 잘 작동하지 못하게 하는 경우가 발생한다.

* + - 1. **도어락 제어 기능 – Not Perfect**

User가 외부에서 User Interface을 통해서 을 완전 잠금 했을 경우 발생하는 제약 조건으로 User의 스마트폰 배터리가 모두 소진되었을 경우 충전되기 전까지는 도어락을 열 수 없다는 것이다. 이는 일정 시간이상 User Interface와 Server가 통신하지 못 할 경우 Server에서 자동으로 도어락을 잠금 해제하는 것으로 설계상 제약 사항을 보완 할 수 있다.

또한, 외부 침입자 발생시 도어락 제어를 통해 감금을 시도 하는데 사용자가 도어락을 부수고 나가는 경우가 발생 할 수도 있다. 이 경우 도어락 만으로 문제를 해결하기 힘들기 때문에 출입문 외부 장치와 최루 가스 살포를 통해서 감금 기능을 보완한다

* + - 1. **센서 값 – Sensitive**

User가 외부에서 User Interface를 통해서 도어락을 완전 잠금 했을 경우 발생하는 제약 조건으로 User의 스마트폰 배터리가 모두 소진되었을 경우 충전되기 전까지는 도어락을 열 수 없다는 것이다. 이는 일정 시간이상 User Interface와 Server가 통신하지 못 할 경우 Server에서 자동으로 도어락을 잠금 해제하는 것으로 설계상 제약 사항을 보완 할 수 있다 외부 침입자를 감지하기 위해서 적외선 거리 센서를 사용한다. 적외선 거리 센서는 주로 창문과 출입문에 설치하여 침입을 감지하게 된다. 센서의 민감도가 낮을 경우에는 외부의 출입을 잘 감지하지 못하는 경우가 발생 할 수도 있고, 센서의 값이 너무 민감할 경우 작은 움직임도 센서가 감지하여 User를 골치아프게 할 수 있다. 또한, User가 반려 동물을 기를 경우 외부 침입자가 아닌 반려 동물이 적외선 거리 센서에 감지되는 경우가 발생 할 수 있다.

* + - 1. **보이스 기능 – Not Perfect**

User가 외부에서 User Interface를 통해서 도어락을 완전 잠금 했을 경우 발생하는 제약 조건으로 User의 스마트폰 배터리가 모두 소진되었을 경우 충전되기 전까지는 도어락을 열 수 없다는 것이다. 이는 일정 시간이상 User Interface와 Server가 통신하지 못 할 경우 Server에서 자동으로 도어락을 잠금 해제하는 것으로 설계상 제약 사항을 보완 할 수 있다. Mimic Voice기능을 수행하기 위해서는 User의 음성을 Naver CSR을 통해서 인식해야 한다. User가 반려 동물을 기르는 경우 반려동물의 소리나 주변 소음으로 음성 인식이 제대로 되지 않을 수도 있다. 이는 거수자에 대한 대처가 늦어져 범죄로 이어질 가능성이 존재한다.

|  |  |  |
| --- | --- | --- |
| Risk Type | Risk | Solution |
| Technology | 적외선 거리 센서를 이용해서 침입 여부를 판단할 때, 외부 요인으로 오탐의 가능성이 있다. | 적외선 거리 센서 뿐만 아니라 카메라와 얼굴인식 기능을 함께 사용하여 탐지의 정확성을 높인다. 또한 적외선 거리 센서를 탐지에 가장 적합한 위치에 배치한다. |
| 가정마다 도어락 기종이 다르기 때문에 호환성 문제가 발생할 수 있다. | 전선을 사용하여 도어락 기종에 관계없이 탈 부착 가능한 형태로 도어락 제어 모듈을 만든다. |
| 집 주인의 말에 적합한 대답을 mapping 하여 출력하는데 어려움이 있다. | 독신 여성이 자주 쓰는 말을 통계적으로 분석하여 이에 적합한 응답을 풀로 만들어 둔다. |
| Google Cloud Platform 사용 경험이 없기 때문에 서버 구축에 어려움이 있다. | 팀원 중 1명을 배치하여 기술 습득 및 교육을 진행하도록 한다. |
| Mimic voice에서 text를 받아 TTS service를 통해 음성으로 변환할 시 지연이 생길 수 있다.. | 해당 서비스를 이용 할 시 매크로로 기본적인 음성을 저장해 놓을 수 있게 하여 지연을 최소화 하며  TTS(Text To Speech) Service인 Naver Clova Speech Synthesis를 통해 다양한 상황 시뮬레이션으로 신속히 이루어지는지 확인한다. |
| Integration | 기능 통합과정에서 마이크로프로세서가 영상처리, 원격제어, 음성서비스 기능을 모두 커버하지 못할 수 있다. | 값싼 마이크로프로세서를 병렬 배치하거나 성능이 좋은 마이크로 프로세스를 사용한다. |
| Team Work | 팀원들의 개인 사정으로 조모임에 불참 할 수 있다. | Git hub로 작업하며 전달사항을 전해 이 후 조모임에 신속히 참여 가능하도록 한다. |
| Testing | 추후 데모 시나리오 시연을 어떻게 보여 줄 것인가? | 간단하게 축소된 크기의 집 모형을 만들어 데모 시나리오대로 시연 할 수 있도록 한다. |

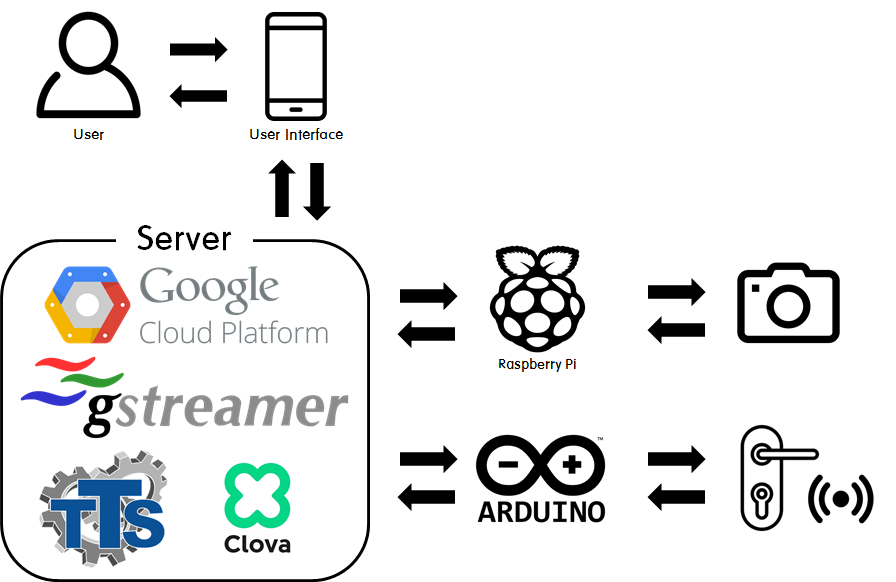
* + 1. **사회적 영향**

혼자 사는 여성의 안전이 위협받고 있고 이를 방지해야 한다는 의식이 낮다. ((다미-뭐라고 하지…)((전체))

* + 1. **비용 / 경제성**

해당 서비스를 이용하기 위해서는 별도의 설치를 요구한다. 따라서 설치 비용이 추가로 들며 설치 후, 카메라, 라즈베리 파이, 아두이노, 최루가스와 같은 하드웨어 장비가 망가질 경우 교체가 필요하다. 또한 최루 가스의 경우 영속적인 것이 아니기 때문에 최루 가스가 고갈됐을 경우 충전을 해야 한다. 따라서 부가적인 비용이 든다.

1. **시스템 구성 및 기능**
   1. **시스템 설명**
      1. **Overall시스템 구조도(그림+내용 수정해야함)**



전체 시스템 구조는 User, User Interface(Mobile App), Server(Google Cloud Platform), Raspberry Pi, Arduino 와 Sensor 및 H/W로 구성된다.

User는 ‘오늘의 집사’를 이용해 홈 시큐리티 서비스를 제공받고자 하는 사용자이다.

User Interface는 Mobile App으로 구성되어 있다. User Interface는 User에게 침입자 감지 알림 기능, 남성 음성 서비스, 도어락 제어 기능, 외부 침입자 및 거수자의 모습 촬영 서비스 등을 제공한다.

Server는 크게 ?가지로 나뉜다. 첫번째인 웹 서버는 Ruby on Rails를 사용하여 구축하고 AWS의 Elastic Beanstalk에 빌드/배포를 한다. Server는 User Interface인 Mobile App와 Middle Ware인 Raspberry Pi 와의 통신, 회원가입, 로그인 인증, 푸쉬메세지 요청, Raspberry Pi 등록 및 아이피 수정을 위해 사용된다. 회원가입시 Raspberry Pi의 고유번호를 이용, 회원과 Raspberry Pi를 1:1로 연결하며, Pi가 부팅 할때 마다 IP address가 담긴 데이터베이스 테이블을 업데이트한다. 침입이 탐지시 Raspberry Pi가 웹 서버에 Post요청을 보내고 서버는 FCM으로 push 요청을 한다. Mobile App에서 사용자 인증이나 회원 가입 요청이 들어오면 웹 서버는 데이터베이스에 접근하여 유저 정보를 확인하거나 저장한다.

두번째인 push 서버는 구글의 Firebase Cloud Messaging이다. 서버에서 push 요청을 할 때 보낸 디바이스의 고유의 토근을 통해서 그에 해당하는 디바이스로 push메시지를 보낸다.

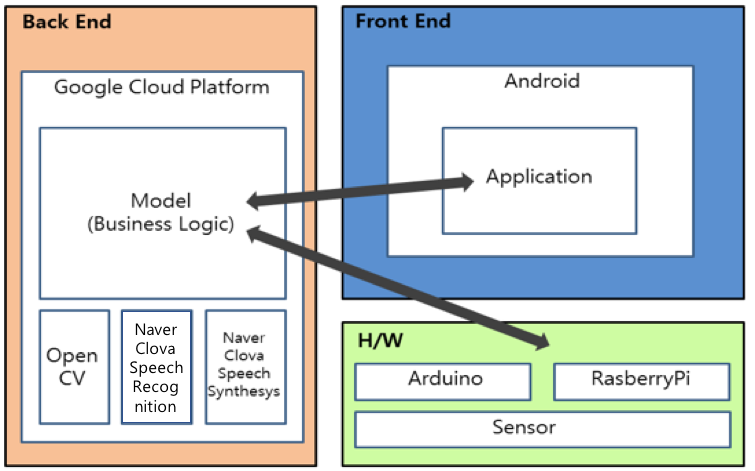
Stt tts 이거 서버 따로 있으면 언급 ㄱ (진성)

// 또한 TTS(Text To Speach) Service와 Naver의 CSR(Clova Speach Recognition)을 이용하여 User의 음성을 인식하고 이를 Text로 전환해 Text에 맞게 맵핑된 변조 음성을 User Interface로 전송해 출력해 준다.//

Raspberry Pi는 Server, Moblie App, Arduino와 통신하며 서로를 이어주는 미들웨어 역할을 한다. 또한 카메라와 Mjpg 스트리머를 사용하여 스트리밍 서버를 만듦과 동시에 얼굴인식을 한다.

Arduino는 침입감지 및 부착된 디바이스 제어를 담당한다.

* + 1. **소프트웨어 구조도**



본 프로젝트는 Front-End, Back-End 그리고 Sensor 및 H/W로 구성되어 있다.

* + - 1. **Front-End**

Front-End는 유저 입장에서 언제 어디서든지 집의 정보를 제공 받고 도어락을 제어할 수 있어야 하기 때문에 스마트폰을 이용한 Mobile App를 사용한다. Mobile App는 Android OS를 기반으로 한다. Android App는 카메라와 적외선 거리 센서로부터 발생한 모든 정보를 수신하고 도어락을 제어하는 기능을 갖는다.

Android App는 적외선 거리 센서가 감지한 외부 침입자에 대한 데이터를 수신한다. 적외선 거리 센서에서 발생한 데이터는 Arduino와 Server를 거쳐 가공된 데이터를 제공 받는다. 또한, 카메라로부터 거수자 및 외부 침입자의 촬영 데이터를 제공 받는데 카메라 데이터는 Raspberry Pi와 Server를 통해서 User Interface로 전송되어 User가 확인 할 수 있도록 한다.

Front-End에서 제공하는 Mimic Voice란 기능을 수행하기 위해서 Naver CSR(Clova Speech Recognition)과 Naver CSS(Clova Speech Synthesys)를 사용한다. User는 User Interface를 통해서 변조를 원하는 문장을 말하고 Naver CSR을 이용해 음성을 인식하고 Text로 변환한다. 변환된 Text는 Naver CSS를 이용해 맵핑된 남성 음성을 출력한다.

원격 도어락 제어 기능은 User가 User Interface를 통해서 원하는 도어락의 잠금 상태를 설정하면 Server와 Arduino를 거쳐 도어락을 제어한다.

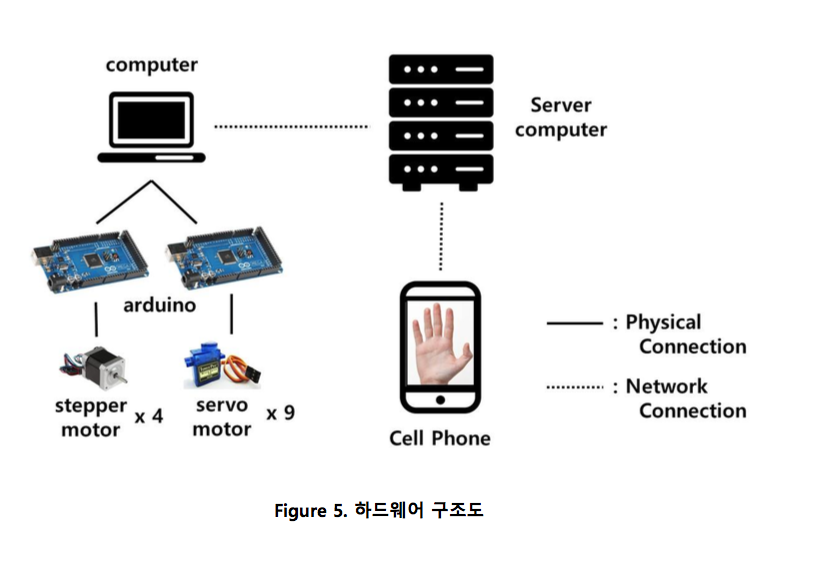
* + - 1. **Back-End**

**Server**

AWS Elastic-Beanstalk를 사용하여 서버를 구축하였다. User와 Raspberry Pi 데이터 저장, Raspberry Pi의 중요 로그(거수자 사진, 침입감지) 저장, UI의 Mimic voice 기능에 필요한 Text 데이터 저장을 위한 DB Table을 가진다. UI로 Push알림을 전송하기위해 Firebase를 사용하였다.

- ((교수님 피드백))백엔드에서의 모델(비스니스로직) 부분의 역할이 매우 중요하며 여기서 처리해야 할 일들이 많이 있는데 이에 대한 구체적인 구조/설계가 생략되어 있어 이를 기준으로 개발일정을 잡기는 매우 어려울 것으로 보임 (-1)

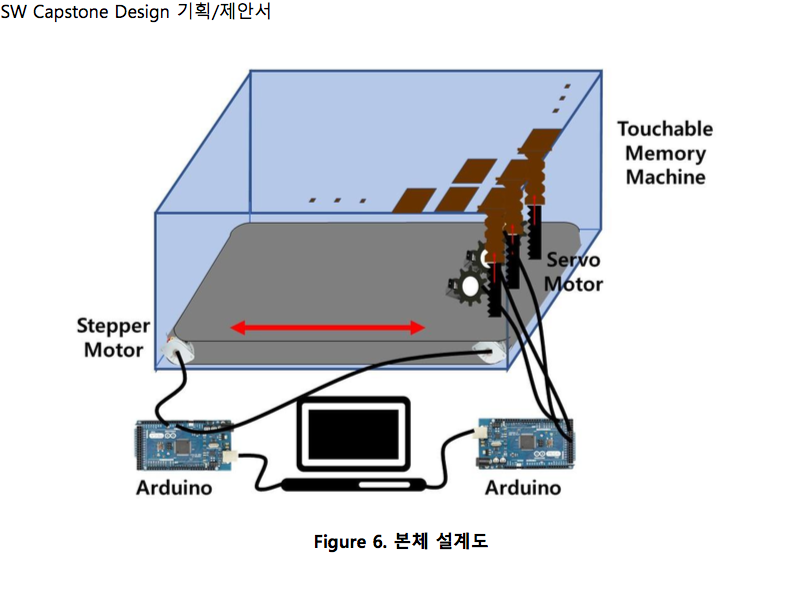
* + 1. **하드웨어 구조도((동성,혁정))**



**Raspberry Pi**

Server, Moblie App, Arduino를 이어주는 미들웨어 역할을 한다. Arduino, App과의 통신에 스레드를 사용하며, 안정성을 위하여 Arduino로부터 읽은 값, Moblie App으로부터 전송된 제어정보를 command Queue에 넣고 순차적으로 처리한다. Server와는 HTTP 통신을 사용하여 거수자 사진, 침입자 감지여부를 전송하며, Mobile App과는 TCP Socket통신으로 제어정보를 받는다. Arduino와는 Serial 통신을 사용하여 침입 여부를 판단한다. 또한 파이카메라를 이용, 연속된 사진을 촬영하여 MJpg Streamer로 스트리밍하고, 동시에 촬영된 사진을 OpenCV 라이브러리와 얼굴인식 데이터파일로 얼굴인식하여 Server로 전송하는 역할을 수행한다.

**Arduino**  
Raspberry Pi와 시리얼 통신하며 반복적으로 제어정보를 주고받는다. 적외선 거리 센서로부터 데이터를 읽고 침입여부를 판단한다. 침입이 감지되었을 시 Raspberry Pi에 침입 감지 메시지를 전송한다. UI에서 선택 된 제어정보를 Raspberry Pi를 통해 전달받고, 해당 제어신호에 맞는 Device(도어락, 버저, 스프레이)를 동작한다.



* + - 1. **Sensor 및 h/w ((혁정-센서부분 + 하드웨어 부분 ))**
    1. **소프트웨어 컴포넌트 목록 (형직수정)**

|  |  |  |
| --- | --- | --- |
| **Type** | **Component Name** | **Description** |
| 재사용 | Naver Clova Speech Recognition API | User의 음성 멘트를 인식하고 Text로 변환시켜 주기위해 사용되는 API이다. User는 User Interface를 통해서 음성 멘트를 입력하고 API를 통해서 Text로 변환하는 구조를 갖는다. |
| 재사용 | Naver Clova Speech Synthesis API | 텍스트로 입력된 멘트를 남자의 목소리로 변환시키기 위해서 사용되는 TTS API이다. 추후에 머신 러닝을 적용시킨다면 사용자가 텍스트를 입력할 필요 없이 자연스러운 대화가 되도록 만들 수 있을 것이다. |
| 재사용 | Face Recognition | 침입자의 얼굴을 찍거나 창가에 거수자를 확인하기위해 사용되는 open CV라이브러리이다. 촬영된 사진으로 얼굴을 인식하여 서버로 전송 시킨다. |
| 신규 | 웹 서버 | 웹 서버는 유저data와 라즈베리파이data에 대한 데이터베이스 접근, 안드로이드와 라즈베리 파이와의 통신, fcm에 push 메시지 요청, 사진을 AWS s3에 업로드 하는 역할을 한다. |
| 신규 | 어플로 제어가능한 도어락 | 아두이노와 도어락을 연결하고 안드로이드 어플리케이션과 아두이노가 통신을 하면서 도어락을 제어할 것이다. 또한 잠금 모드가 설정되면 하드웨어에서 제어가 되지 않도록 설정할 예정이다. 추후에 NFC를 응용할 수도 있을 것이다. |
| 재사용 | Mjpg Streamer | Pi camera를 통해 촬영한 jpg파일을 읽어 Streaming서버로 전송한다. |
| 재사용 | Firebase Cloud Messaging | 안드로이드에 push 메시지를 보내기 위해 사용되는 플랫폼이다. Server가 라즈베리파이로부터 신호를 받으면 push 요청 메시지를 fcm에 보낸다. fcm은 받은 요청 메시지에 따라 안드로이드 유저에게 push메시지를 보낸다. |

Table ? Software Components

* + 1. **하드웨어 컴포넌트 목록**

|  |  |  |
| --- | --- | --- |
| **Type** | **Component Name** | **Description** |
| 신규 | 어플로 제어가능한 도어락 | 아두이노와 도어락을 연결하고 안드로이드 어플리케이션과 아두이노가 통신을 하면서 도어락을 제어할 것이다. 또한 잠금 모드가 설정되면 하드웨어에서 제어가 되지 않도록 설정할 예정이다. 추후에 NFC를 응용할 수도 있을 것이다. |
|  | Arduino Mega 2560 |  |
|  | Raspberry Pi |  |
|  | 최루탄 발사기 |  |
|  | Samsung galaxy note 4 |  |

Table ? Hardware Components

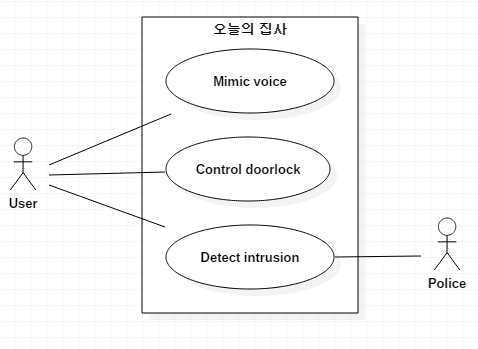
* 1. **기능 설명**
     1. **제안서 대비 기능 설명** 
        1. **사용자 관점 features**

|  |  |  |
| --- | --- | --- |
| Feature | 제안서 | 최종 |
| 사용자 정보 입력 | 제안서에서는 별도의 feature로 다루지 않음 | 사용자가 e-mail, 비밀번호, 라즈베리 파이 고유번호를 입력.  각 사용자의 데이터는 e-mail로 구분되어 저장. |
| 경찰에 신고 | 침입이 감지되어 사용자의 안드로이드 어플에 알림이 오면 사용자는 경찰에 신고할지 여부를 선택할 수 있다. 경찰에 신고하기를 누르면 미리 입력된 사용자의 집주소가 적힌 문자를 112번으로 전송한다. | 변동 사항 없음. |
| 남성 음성  서비스 | 해당 서비스를 개발하기 위해서 Never Clova Speech Synthesis API와 Never Clova Speech Recognition API를 사용한다. 처음 사용자가 특정 말을 할때 그에 맞는 대답이 나오도록 설정한다. 사용자가 안드로이드 어플에서 남성 음성 서비스를 활성화 시킨 후 특정 말을 말하면 음성이 텍스트로 전환되고 전환된 텍스트는 다시 한번 남성 음성으로 변조되어 출력되게 된다. | 변동 사항 없음. |
| 도어락 제어 | 사용자가 안드로이드 어플을 통해  도어락을 열고 닫을 수 있다. | 변동 사항 없음. |
| 도어락 잠금 모드 | 사용자가 안드로이드 어플에서 도어락 잠금 모드를 설정하면 다시 해제하기 전까지 도어락이 열리지 않게 된다. | 해당 기능은 불필요하다 생각하여 빼도록 함. |
| 최루가스 살포 | 사용자에게 침입 감지 알림이 오면 최루가스를 살포할지 선택하여 살포를 선택할 시 최루가스가 살포된다. | 변동 사항 없음. |
| 스트리밍 | 제안서에서는 별도의 feature로 다루지 않음 | 하나의 카메라의 각도를 제어하여 창문과 현관 쪽 영상을 제공. |

* + - 1. **관리자 관점 features**

|  |  |  |
| --- | --- | --- |
| Feature | 제안서 | 최종 |
| 침입 감지 | 집 내부에 설치된 적외선 거리 센서를 통해서 센서가 동작하면 침입이 감지된 것으로 간주한다. 침입이 감지되면 사용자의 Mobile App에 알림을 보낸다. | 변동 사항 없음. |
| 실시간 감시 | 적외선 거리 센서와 카메라를 항시 구동하여 실시간 감시 체계를 활성화 한다. | 변동 사항 없음. |
| 범죄자 캡쳐 | 적외선 거리 센서를 통해서 외부 침입자가 감지 될 경우 출입문에 설치된 카메라를 통해 외부 침입자를 캡쳐한다. 또한, 창문에 설치된 카메라는 OpenCV를 통해서 거수자의 얼굴이 일정 시간이상 감지되면 거수자의 얼굴을 캡쳐한다. 캡쳐된 내용은 사용자에게 전송되고 저장할 수 있다. | 적외선 거리 센서를 통해 감지된 외부 침입자의 경우 캡쳐가 아닌 실시간 스트리밍 영상을 사용자에게 제공. |
| 데이터 관리 | 제안서에서는 별도의 feature로 다루지 않음 | 관리자가 직접 AWS의 RDB에 접속해서 관리한다. 추후에 관리자용 웹페이지를 제작해서 쉽게 DB를 다룰 수 있도록 할 예정.(근데 예정을 적는게 맞는지 모르겠다) |
| 푸시 메시지 | 제안서에서는 별도의 feature로 다루지 않음 | 라즈베리 파이가 Post 방식으로 웹 서버에 메시지를 보내면 웹 서버는 FCM으로 push 메시지를 요청한다. 이때 웹서버는 database에 저장된 라즈베리파이 등록번호로 유저의 FCM 등록 토큰을 찾아서 메시지와 함께 보낸다. 만약 라즈베리파이가 사진을 첨부한다면 AWS s3에 사진을 저장하고 사진의 URL 주소를 FCM 메시지와 함께 보낸다. FCM은 서버가 보낸 FCM 등록 토큰에 해당하는 유저의 디바이스로 push 메시지를 보낸다. 사진 주소가 첨부 된다면 사진도 함께 보낸다. |

* + 1. **Use-Case**
       1. **Use Case Diagram(다시 그려야 함-안그린데도 많던디)**



* + - 1. **Use Case Specification**

|  |  |
| --- | --- |
| **Use Case 1** | |
| Use Case Title | Register User |
| Actor Name | 사용자 |
| Text Description | 사용자는 어플리케이션에 본인을 등록한다. |
| Preconditions | 사용자는 자신의 라즈베리 파이 시리얼 번호를 알아야 한다.  사용자는 개인 이메일 주소를 갖고 있어야 한다. |
| Basic Path | 1. 사용자는 로그인 화면에서 [sign up]기능을 선택한다.  2. 사용자는 각 Textbox에 정보(ID, PW, Serial Number)를 입력한다.  3. 사용자는 [sign up]요청을 한다.  4. 어플리케이션은 사용자의 정보를 서버로 전송한다.  5. 서버는 어플리케이션으로부터 받은 Serial Number를 테이블에서 검색한다.  6. 사용자 등록이 완료된다. |
| Alternative Path | 2a) 입력하지 않은 경우   1. 정보를 입력해달라는 Toast 메시지가 발생한다.   2b) 시리얼 번호가 DB에 미등록된 경우   1. 사용자 정보가 Database에 등록되지 않는다. |
| Post Conditions | 사용자가 서버에 등록된다. |

|  |  |
| --- | --- |
| **Use Case 2** | |
| Use Case Title | Mimic Voice |
| Actor Name | 사용자 |
| Text Description | 사용자가 남성 음성 서비스를 활성화 시킨다.  지정된 말에 맞는 말이 스피커를 통해 남자 목소리로 나온다. |
| Preconditions | 사용자는 로그인 되어있다. |
| Basic Path | 1. 사용자가 안드로이드 어플을 이용해 음성 서비스 기능을 활성화시킨다.  2. 사용자가 지정된 말을 하면 그에 맞는  음성이 스피커를 통해 출력된다. |
| Alternative Path | 2a) 사용자가 데이터베이스에 등록되지 않은 말을 했다.   1. 다시 입력해달라는 음성이 출력된다. |
| Post Conditions | 스피커에서 남성 음성이 출력된다. |

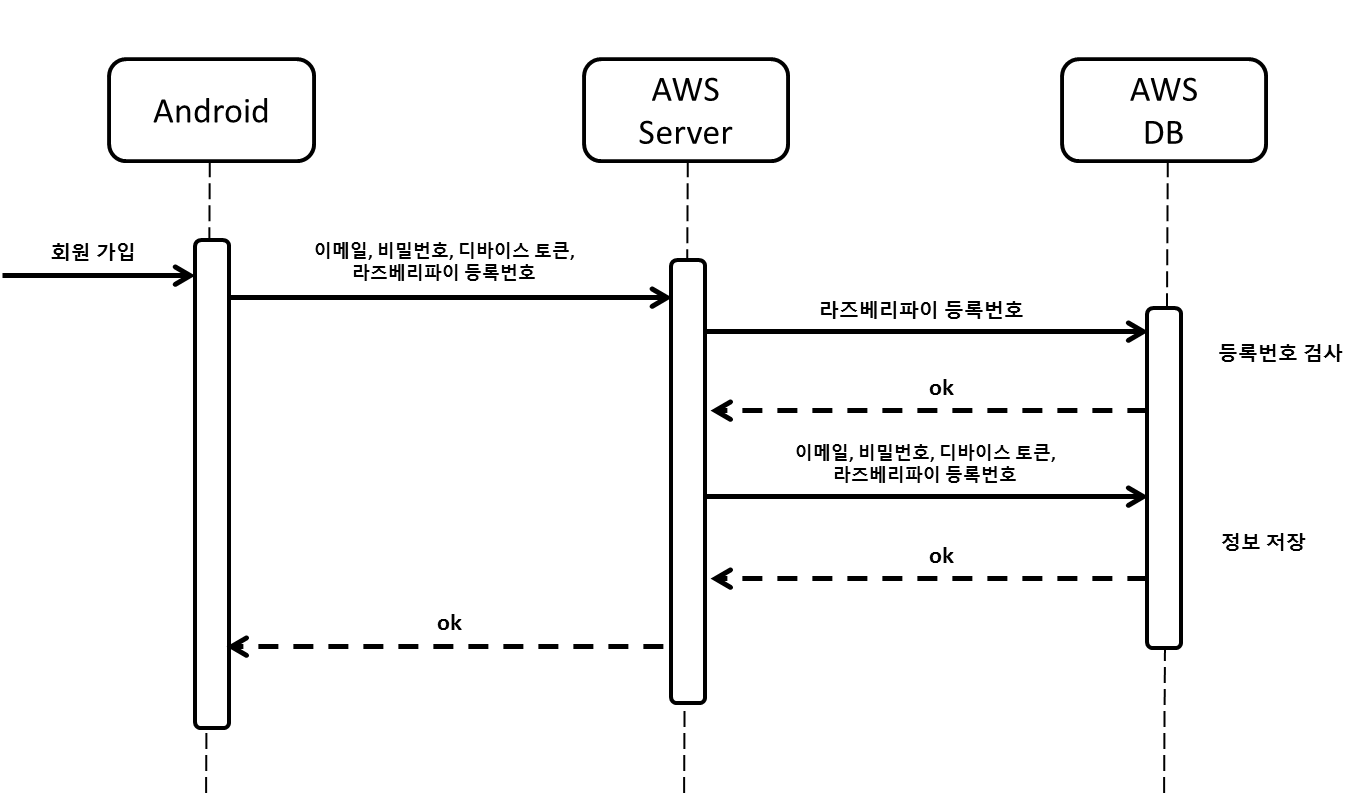
|  |  |
| --- | --- |
| **Use Case 3** | |
| Use Case Title | Control Doorlock |
| Actor Name | 사용자 |
| Text Description | 사용자가 어플리케이션으로 도어락을 제어한다. |
| Preconditions | 사용자가 로그인 되어있다. |
| Basic Path | 1. 사용자가 안드로이드 어플 메뉴화면에 [control doorlock]기능을 선택한다.  2. 사용자가 [open] 을 요청한다.  3. 어플리케이션은 제어정보를 라즈베리파이에 전송한다.  4. 라즈베리파이는 제어정보를 아두이노로 전송한다.  5. 아두이노는 제어정보를 파싱하여 도어락을 제어한다.  6. 도어락 잠금이 해제된다. |
| Alternative Path | 3a) 어플리케이션과 라즈베리파이의 통신이 실패했을 경우.   1. 사용자에게 오류를 알린다.. 2. 초기 메뉴 페이지로 이동한다. 3. 라즈베리파이는 재부팅하여 서버로 IP를 재전송한다. 4. 사용자는 다시 로그인하여 라즈베리파이와 연결한다. |
| Post Conditions | 도어락 잠금이 해제된다. |

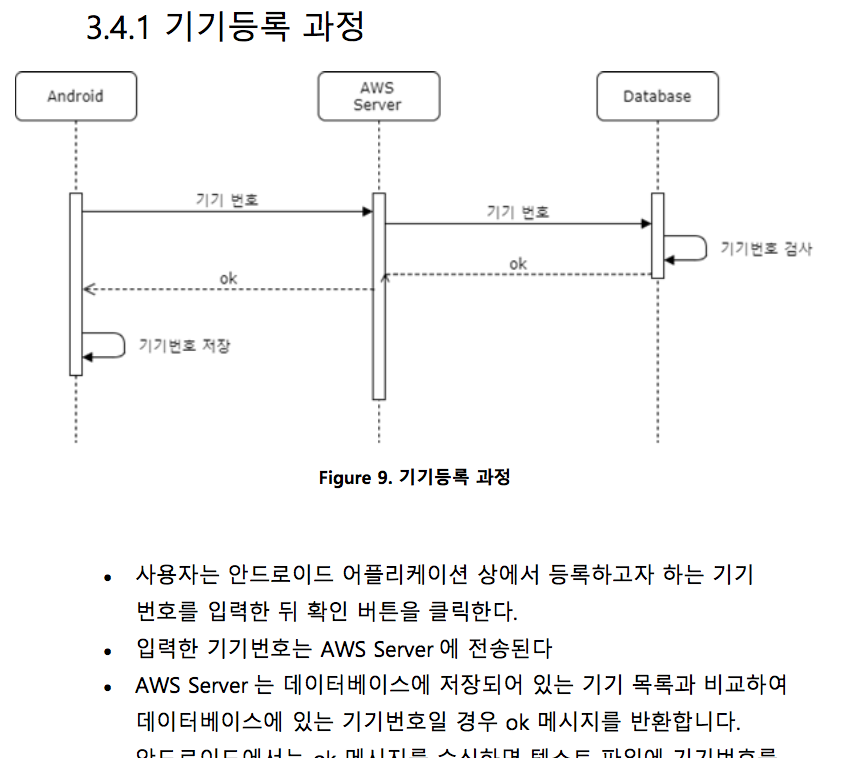
|  |  |
| --- | --- |
| **Use Case 4** | |
| Use Case Title | Streaming |
| Actor Name | 사용자 |
| Text Description | 사용자는 어플리케이션으로 창문과 현관의 실시간 스트리밍영상을 확인한다. |
| Preconditions | 사용자가 로그인 되어있다.  라즈베리파이의 Streaming Server가 동작하고있다. |
| Basic Path | 1. 사용자는 [streaming]기능을 선택한다.  2. 어플리케이션은 Streaming Server로부터 영상정보를 수신한다.  3. 사용자는 [창문을 보여줘], [현관을 보여줘] 중 하나를 선택한다.  4. 어플리케이션은 제어정보를 라즈베리파이로 전송한다.  5. 라즈베리파이는 제어정보를 카메라에 부착된 모터를 제어한다.  6. 카메라 각도가 회전한다. |
| Alternative Path | 2a) Streaming Server로부터 영상정보 수신이 실패할 경우   1. 어플리케이션은 라즈베리파이로 수신 실패 메시지를 전송한다. 2. 라즈베리파이는 Streaming 서버를 재가동시킨다.   4a) 어플리케이션과 라즈베리파이의 통신이 실패했을 경우.   1. 사용자에게 오류를 알린다. 2. 초기 메뉴 페이지로 이동한다. 3. 라즈베리파이는 재부팅하여 서버로 IP를 재전송한다. 4. 사용자는 다시 로그인하여 라즈베리파이와 연결한다. |
| Post Conditions | 사용자는 현관과 창문의 실시간 영상을 확인 할 수 있다. |

|  |  |
| --- | --- |
| **Use Case 5** | |
| Use Case Title | Detect Intrusion |
| Actor Name | 사용자, 경찰 |
| Text Description | 사용자는 탐지 알림을 받고 침입자의 영상을 확인하고 경찰에 신고하며 경보 조치를 취한다. |
| Preconditions | 사용자가 로그인 되어있다.  침입자가 집 안에 있다. |
| Basic Path | 1. 적외선 감지센서에서 침입을 감지한다.  2. 아두이노는 라즈베리파이로 칩임감지 정보를 전송한다.  3. 라즈베리파이는 서버로 침입감지 정보를 전송한다.  4. 서버는 로그를 저장하고, 사용자에게 push 메시지를 전송한다.  5. 사용자는 push 메시지를 선택한다  6. 사용자는 [카메라 확인] 기능을 선택한다.  7. 어플리케이션은 Streaming Server로부터 영상정보를 수신한다.  8. 사용자는 [SMS 보내기] 기능을 선택한다.  9. 어플리케이션은 사용자의 주소가 적혀있는 문자를 경찰에게 전송한다.  10. 사용자는 [경보 조치 켜기] 기능을 선택한다.  11. 어플리케이션은 라즈베리파이로 제어 정보를 전송한다.  12. 라즈베리파이는 아두이노로 제어 정보를 전송한다.  13. 아두이노는 버저를 최루가스 살포 기능을 동작시킨다. |
| Alternative Path | 1a) 사용자가 집에 있을 경우   1. 사용자의 모바일과 라즈베리파이가 블루투스 연결된다. 2. 라즈베리파이는 침입감지 정보를 무시한다.   2a) Streaming Server로부터 영상정보 수신이 실패할 경우   1. 어플리케이션은 라즈베리파이로 수신 실패 메시지를 전송한다. 2. 라즈베리파이는 Streaming 서버를 재가동시킨다.   9a) SMS 문자 전송에 실패 할 시   1. 사용자에게 전송 실패를 알린다.   11a) 어플리케이션과 라즈베리파이의 통신이 실패했을 경우.   1. 사용자에게 오류를 알린다. 2. 초기 메뉴 페이지로 이동한다. 3. 라즈베리파이는 재부팅하여 서버로 IP를 재전송한다.   사용자는 다시 로그인하여 라즈베리파이와 연결한다. |
| Post Conditions |  |

|  |  |
| --- | --- |
| **Use Case 6** | |
| Use Case Title | Detect Window |
| Actor Name | 사용자 |
| Text Description | 사용자는 창문 거수자의 얼굴 사진을 확인한다. |
| Preconditions | 창문에 거수자가 일정시간 집안을 바라본다. |
| Basic Path | 1, 라즈베리파이는 얼굴을 인식하여 사진을 저장한 후 서버로 전송한다.  2. 서버는 사진을 저장한 후 어플리케이션에 push 메시지(사진)를 전달한다.  3. 어플리케이션에 push알림이 뜬다.  4. 사용자는 사진을 확인한다. |
| Alternative Path |  |
| Post Conditions | 서버에 거수자의 사진이 저장된다. |

* 1. **핵심 알고리즘 / 기술에 대한 상세 설명 (api와 알고리즘 설명 들어가야함)**
  2. **동작 과정 설명 (사진 전송 동작 과정, 아두이노 제어 과정 등등)**

****

그림 그려야 함

((교수님 피드백))- 다양한 데이터들이 활용되고 서버와 안드로이드 기기간의 인터페이가 있음에도, 데이터 구조에 대한 설명이 없음 (-1)

- 사용자 계정 등을 운영하기 위한 부분이 반영되어 있지 않음. 내용에서도 빠진 부분 (-0.5)

1. **구현 결과**
   1. **구현 환경 (형직수정)**

|  |  |  |
| --- | --- | --- |
|  | **Factor** | **Enviroment** |
| **Server** | **IDE** | **RubyMine 2017.03** |
| **OS** | **Ubutu 16.04.3 LTS** |
| **Platform** | **AWS ElasticBeanstalk** |
| **Language** | **Ruby** |
| **Database** | **AWS RDB MySQL** |
| **Framework** | **Ruby on Rails 5.1.4** |
| **Push Message** | **Firebase Cloud Messaging** |
| **Virtual Computing** | **AWS EC2** |
| **Virtual Storge** | **AWS S3** |
| **Arduino** | **IDE** | **Arduino Sketch** |
| **OS** | **Windows 10** |
| **Language** | **C** |
| **Raspberry Pi** | **IDE ?** |  |
| **OS ?** |  |
| **Language** | **Python** |
| **??** |  |
| **Application** | **IDE** | **Android Studio 3.0.0** |
| **Compile SDK, Tartget SDK version** | **V26** |
| **MIN SDK version** | **V23** |
| **Language** | **Java** |
| **Test Device** | **Galaxy Note 4 (size: 5.7 inch, Resolution : 2560 x 1440)** |
|  |  |  |
|  |  |  |
|  |  |  |

**Table ?. 구현 환경**

* 1. **테스트 시나리오 (kpi기준으로)**
     1. **사용자 측면 (형직수정)**

1)거수자가 테스트 사용자의 집 창문을 통해 내부를 염탐한다.

2) 설치된 카메라에 거수자의 얼굴이 인식되고 사진을 찍는다.

3) 라즈베리 파이가 찍은 사진을 서버에 전송한다.

4) 서버가 전송받은 사진을 가상 저장소에 저장한다.

5) 저장한 사진의 URL주소를 FCM을 통해 테스트 사용자의 어플리케이션에 push 메시지를 보낸다.

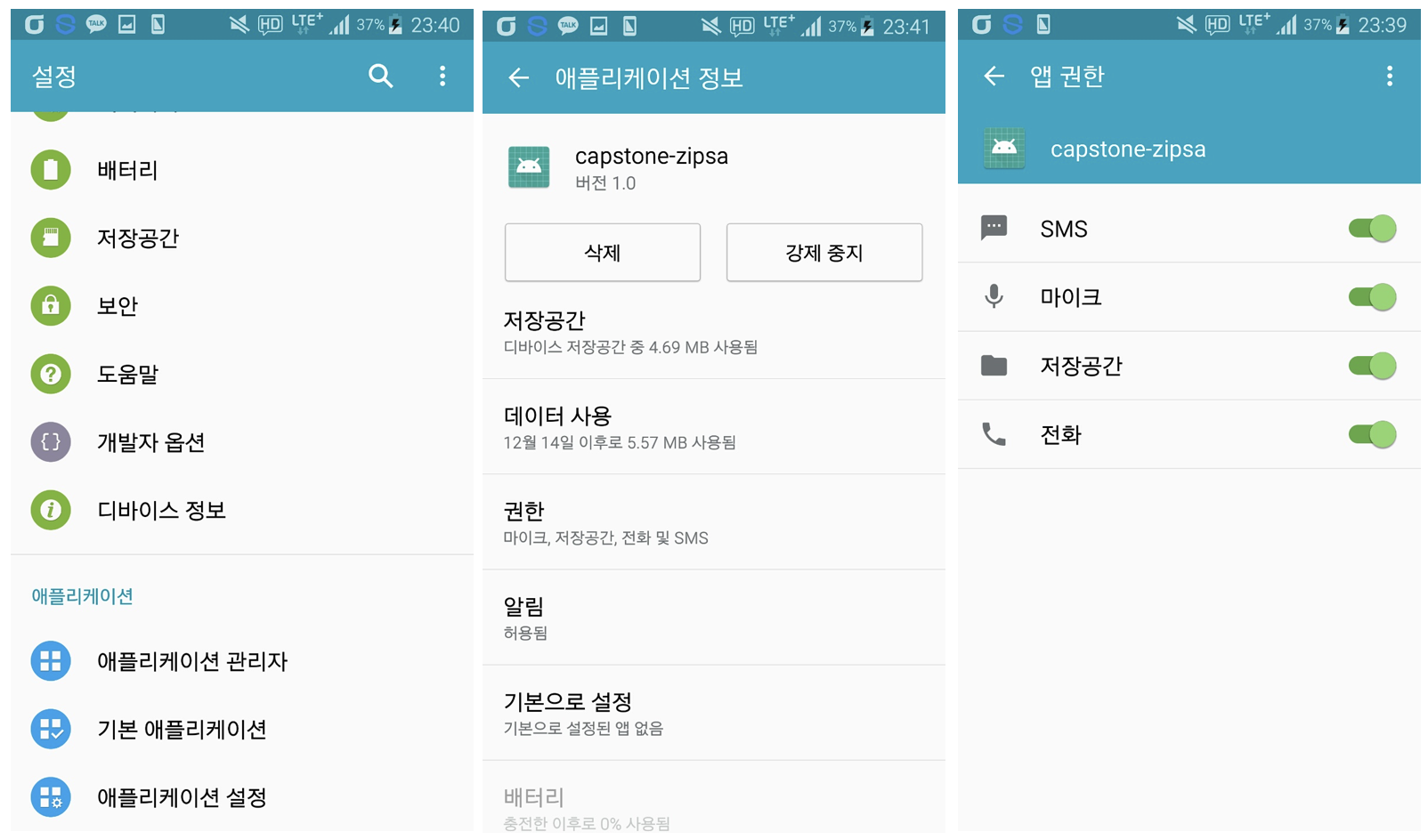
6) 테스트 사용자가 거수자의 얼굴이 찍힌 push 메시지를 받는다.

* + 1. **관리자 측면?**
  1. **테스트 결과**
     1. **기능/성능/안정성 및 기타 kpi대비**

|  |  |  |  |
| --- | --- | --- | --- |
| **영역** | | **성과지표** | **비중(%)** |
| 신뢰도 | 얼굴인식 |  | 30% |
| 센서탐지 |  | 30% |
| 성능 | | 탐지 후 사용자에게 알림이 오는 반응속도가 3초 이내 | 30% |

* 1. **메뉴얼** 
     1. **접근성 설정 (Android)**

본 서비스를 이용하기 앞서 접근성 설정을 해야 한다. 과정은 다음과 같다. ((진성-설명))

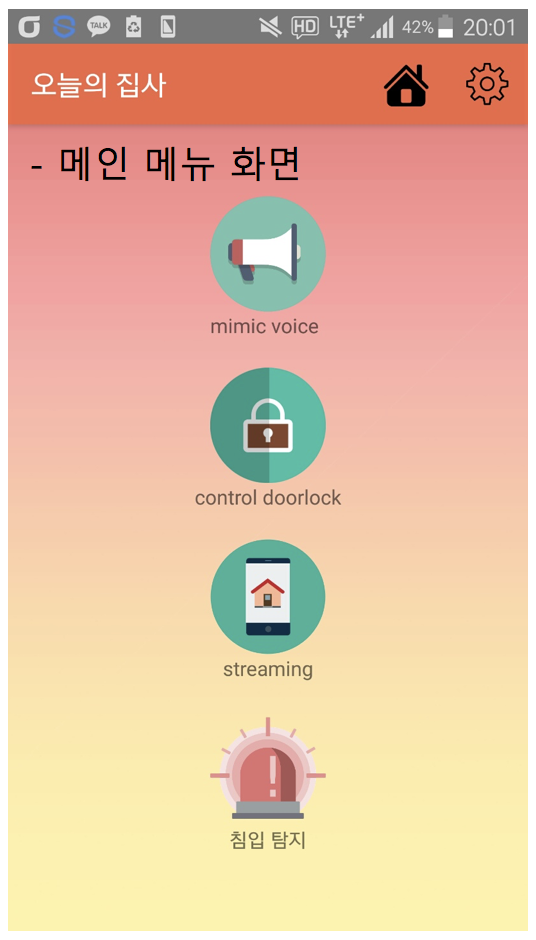


* + 1. **로그인, 사용자 등록 화면**



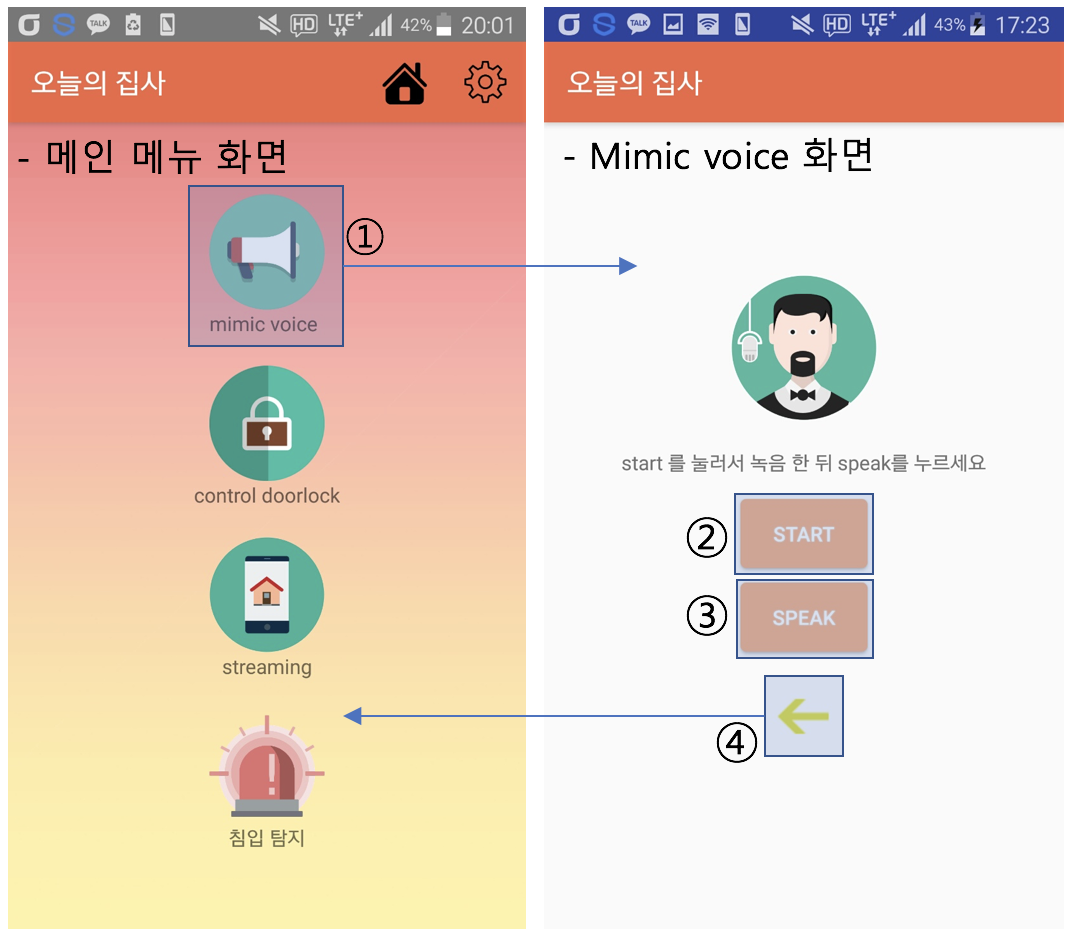
사용자는 오늘의 집사 서비스를 사용하기 위해서는 log in을 해야 한다. 만약 처음 사용한다면 log in 화면 하단에 있는 ① [Sign up] 버튼을 누르면 나오는 사용자 등록 화면에서 e-mail, 비밀번호, 라즈베리 파이 시리얼 번호를 입력 한 뒤 하단 왼쪽에 있는 ② [Sign up] 버튼을 눌러 사용자 등록을 해야한다. 사용자 등록을 하면 다시 log in 화면으로 돌아간다. 사용자 등록할 필요가 없다면 ③ [Before]버튼을 누르면 log in 화면으로 돌아간다. 이 후 log in 화면으로 돌아가 e-mail 과 비밀번호를 입력하고 ④ [Login] 버튼을 누르면 본 서비스를 이용 할 수 있다.

* + 1. **메인 메뉴 화면**

****

사용자는 log in 후에 다음과 같은 메인 메뉴 화면을 볼 수 있다. 메뉴에는 [mimic voice], [control doorlock], [streaming], [침입 탐지]가 있으며사용자가 원하는 기능을 선택하여 해당 기능을 이용할 수 있다.

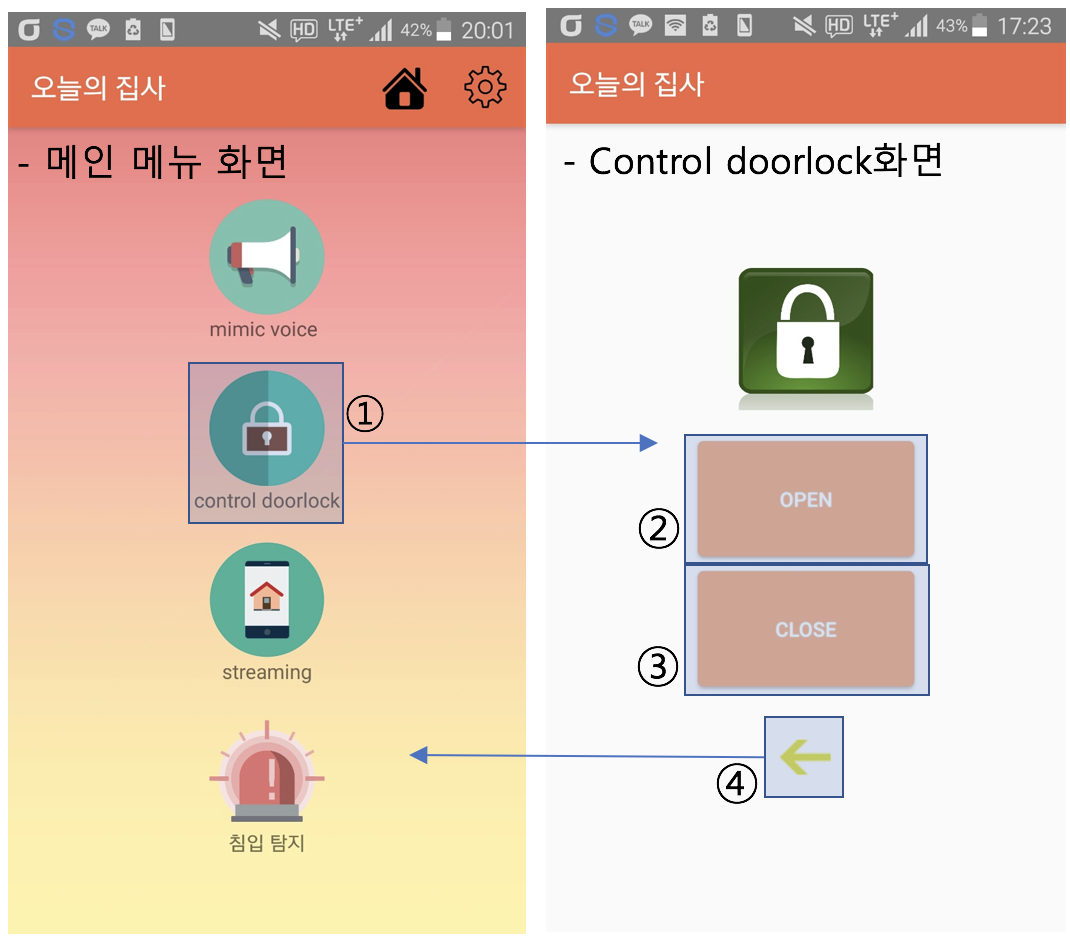
* + 1. **mimic voice**



((진성- 맞는지?)) 메인 메뉴 화면에서 ① [mimic voice] 버튼을 누르면 다음과 같은 화면이 나온다. 해당 기능을 이용하기 위해서 우선 [START] 버튼을 누른 뒤 핸드폰의 마이크에 본인이 원하는 단어를 말한다. 그 후 ② [START]버튼이 [STOP] 버튼으로 바뀌었을 때 다시 한번 버튼을 누른 뒤, 그 아래 ③ [SPEAK] 버튼을 누른다. 버튼을 누르면 사용자가 말한 단어에 알맞는 남성의 대답이 나온다. 메인 메뉴 화면으로 돌아가고 싶으면 최 하단의 ④ [노란색 화살표] 버튼을 누르면 된다.

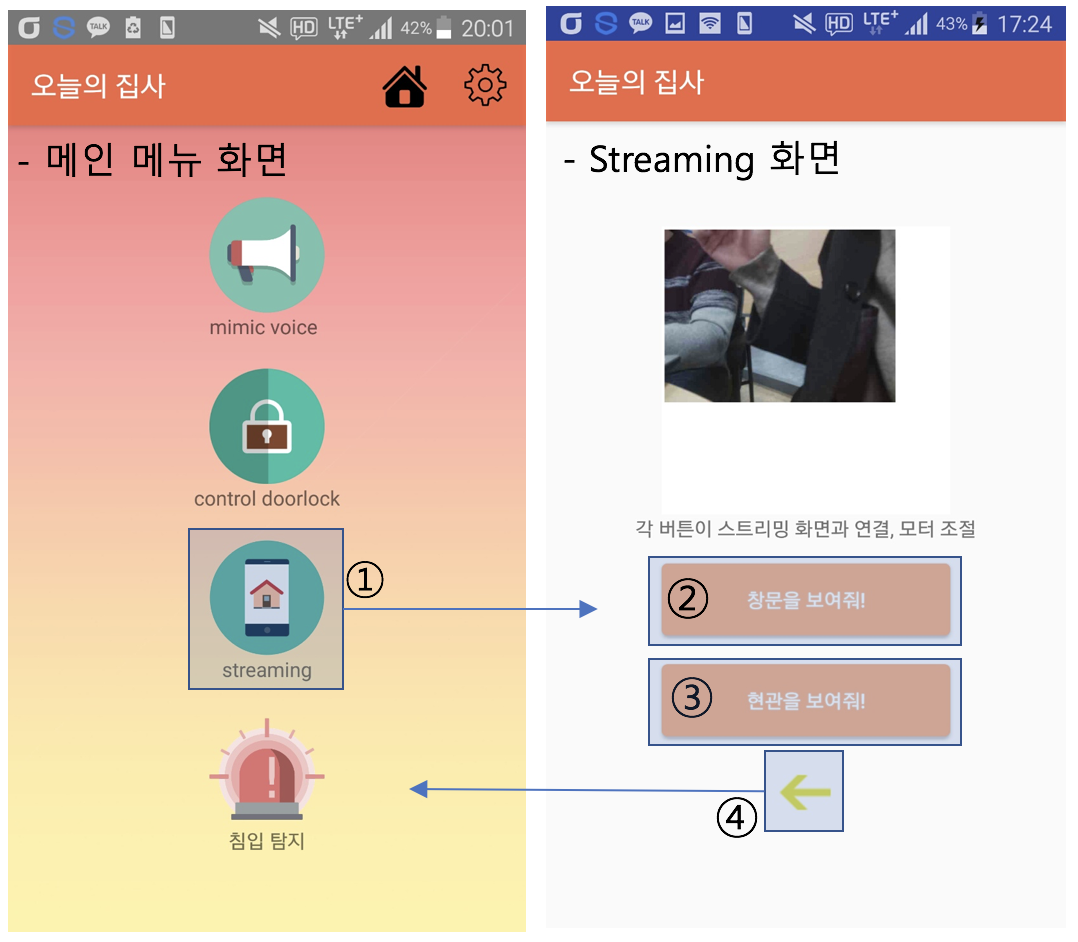
Ex) 사용자 “ 택배 “ , 어플리케이션 “집 앞에 놔주세요” .

* + 1. **control door lock**



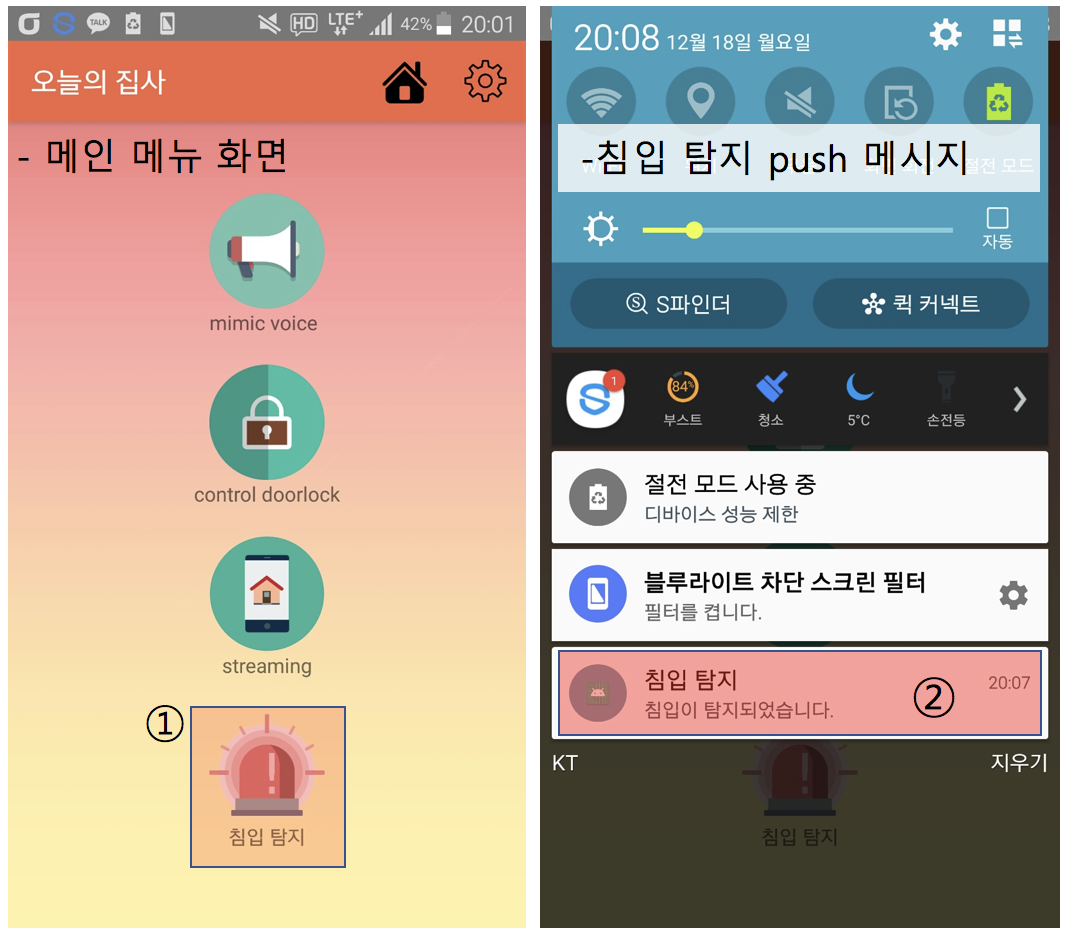
메인 메뉴 화면에서 ① [control doorlock] 버튼을 누르면 다음과 같은 화면이 나온다. 사용자가 ② [OPEN] 버튼을 누르면 원격으로 집에 설치된 도어락이 열린다. ③ [CLOSE] 버튼을 누르면 ((혁정- 닫는거 뭐라고 말할지.) 메인 메뉴 화면으로 돌아가고 싶으면 최 하단의 ④ [노란색 화살표] 버튼을 누르면 된다.

* + 1. **streaming**

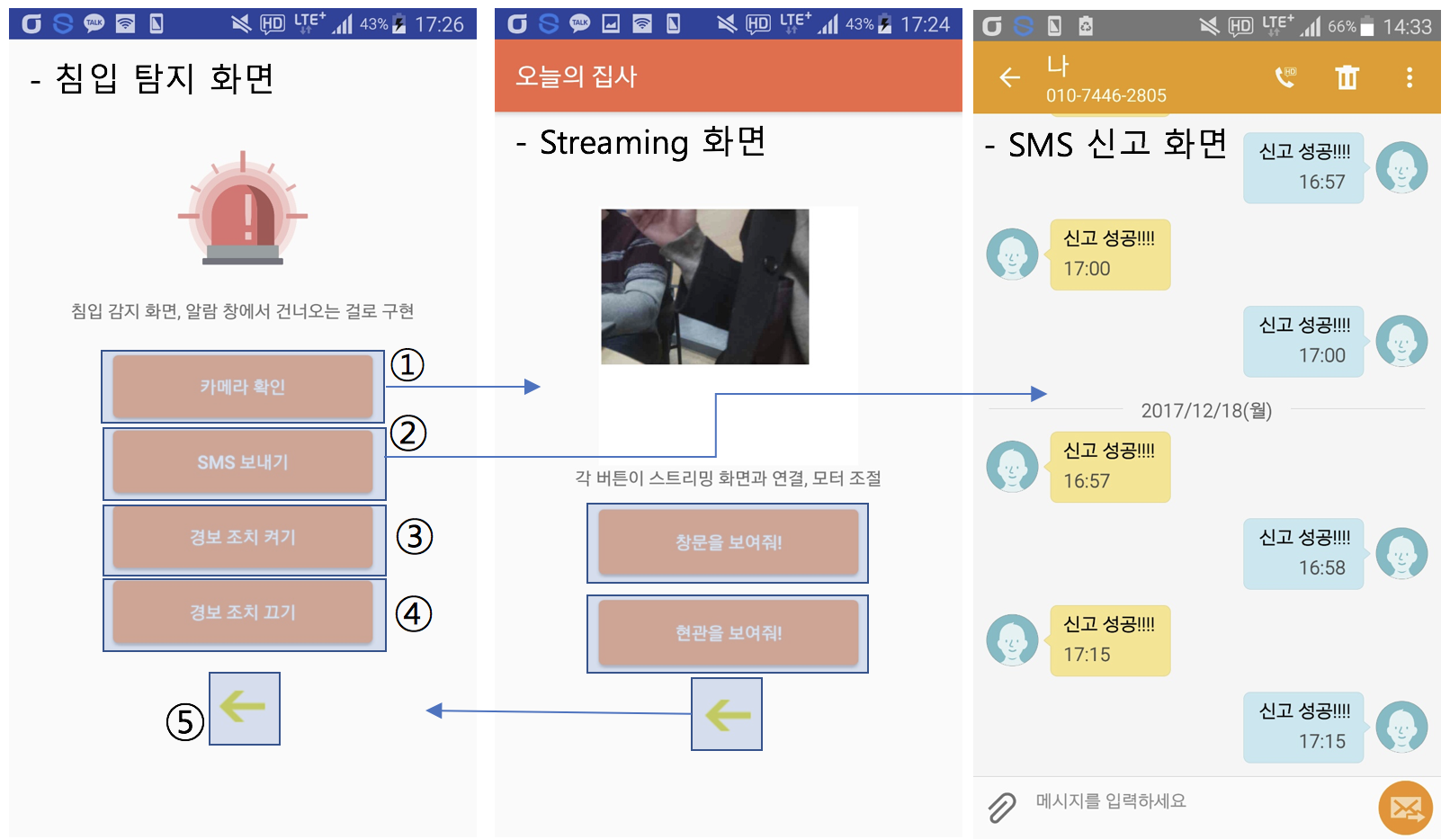


메인 메뉴 화면에서 ① [streaming] 버튼을 누르면 다음과 같은 화면이 나온다. 사용자가 ② [창문을 보여줘!] 버튼을 누르면 집안에 설치된 카메라가 창문 쪽으로 돌아가 스트리밍 화면에 창문 쪽 실시간 스트리밍 영상을 보여준다. 사용자가 ③ [현관을 보여줘!] 버튼을 누르면 설치된 카메라가 현관 쪽으로 돌아가 현관 쪽 실시간 스트리밍 영상을 보여준다. 메인 메뉴 화면으로 돌아가고 싶으면 최 하단의 ④ [노란색 화살표] 버튼을 누르면 된다.

* + 1. **침입 탐지 (현관)**

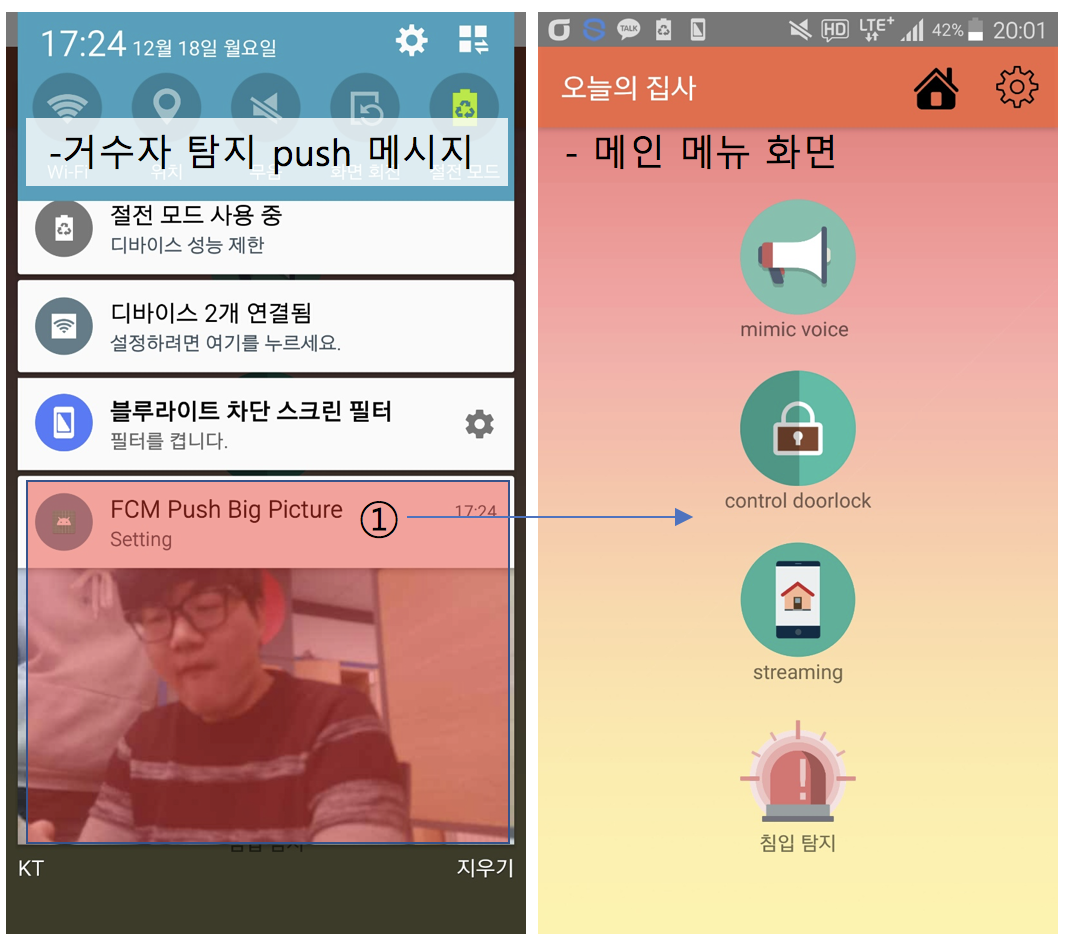


침입 탐지 화면은 메인 메뉴에서의 ① [침입 탐지] 버튼을 눌렀을 시와 침입이 탐지 되었을 시로 나뉜다. 현관에서 침입이 탐지 되면 사용자의 핸드폰에 진동과 함께 “침입 탐지” push 메시지가 뜨고, 사용자가 ② [push 메시지]를 누르면 침입 탐지 화면이 나타난다.



침입 탐지 화면의 첫번째 메뉴인 ① [카메라 확인] 버튼을 누르면 사용자는 현재 집안의 스트리밍 영상을 확인 할 수 있다. (streaming 화면의 사용법과 이전과 동일하다.) 두번째 메뉴인 ② [SMS 보내기] 버튼을 누르면 사용자는 경찰에게 자신의 집 주소가 담긴 신고 문자 메시지를 보낼 수 있다. (실제 신고를 할 수 없기 때문에 신고 성공으로 대체 하였다) 세번째 메뉴인 ③ [경보 조치 켜기] 버튼을 누르면 사용자는 도어락에 설치된 경보음과 최루 가스를 킬 수 있다. 네번째 메뉴인 ④ [경보 조치 끄기] 버튼을 누르면 경보음과 최루 가스가 해제 된다. 각 화면 하단의 ⑤ [노란색 화살표] 버튼을 누르면 이전 화면으로 갈 수 있다.

* + 1. **거수자 탐지 (창문)**



창문에서 거수자가 탐지 되면 얼굴을 인식한 뒤 사진을 찍는다. 사용자에게 사진과 함께 push 메시지 화면으로 이를 알려준다. 사용자는 해당 ① [push 메시지] 를 누르면 오늘의 집사 메인 메뉴로 이동한다.

1. **과제 관리**
   1. **과제 관리 도구 및 활용 설명**
      1. **프로젝트 통합 관리**
      2. **소스 및 변경 관리**

소스코드의 공유와 변경 관리에는 Github를 사용하였다. Github는 분산 버전 관리 툴인 깃(Git)을 사용하는 프로젝트를 지원하는 웹호스팅 서비스이다.

* + 1. **코드 테스트**
    2. **code convention(이대로 써도 되는지?)**

|  |  |
| --- | --- |
| Category | Description |
| file | 확장자   |  |  | | --- | --- | | 파일 형태 | 확장자 | | 자바 소스 | .java | | 자바 바이트 코드 | .class | | 파이썬 소스 | .py | | C 소스 | .c | | C 헤더 파일 | .h | | C 바이트 코드 | .obj | |
| Space | 4개의 빈 칸(space)를 들여쓰기 단위로 사용한다.  코드 그룹 간 빈 줄은 1줄로 한다. |
| comment | 1.시작주석은 아래와 같이 사용한다.   |  | | --- | | /\*  \* 클래스 이름  \*  \* 버전 정보  \*  \* 날짜  \*  \* 저작권  \*/ |   2. 시작 주석을 제외하고는 모두 라인 주석을 사용한다.  3. 라인 주석은 변수 선언 시, 블록 주석은 각각의 파일이 시작될 때와 메서드 전에 사용된다. |
| variable and function declarations | 한 줄에 하나의 선언문을 쓴다. |
| 선언은 블록의 시작에 위치해야 한다(보통 블록은 중괄호 "{ }" 로 둘러싸인 코드를 의미한다). |
| 타입과 변수 이름사이에 하나의 space를 둔다. |
|
| 변수 이름은 되도록 알파벳 순서로 정렬한다. |
| 전역변수 사용 금지 |
| 함수 이름의 시작은 소문자로 하며 2단어 이상인 경우 두 번째 단어의 첫 글자를 대문자로 한다. |
| 변수 이름의 시작은 소문자로 하며 2단어 이상인 경우 두 번째 단어의 첫 글자를 대문자로 한다. |
| 상수 이름은 항상 대문자로 하며 2단어 이상인 경우 단어 사이에 ‘\_’를 넣는다 |
| 가능한 한 선언과 정의 후 사용한다. |
| Line length | 일반적으로 한 줄에 70자 이상을 가지지 않는다.  연산자 앞(,)콤마 후에 두 줄로 나눈다 |
|
|
|
|

* 1. **구성원간 업무 분장(좀더 상세하게)**

|  |  |
| --- | --- |
| 이름 | 업무 분담 |
| 권혁정 | 여성 거주자와 남성의 음성 맵핑 및 출력 |
| 김다미 | 침입 탐지 및 조치 기능 |
| 김형직 | 도어락 모듈 원격 제어 |
| 유진성 | 영상 처리 및 얼굴인식 |
| 오동성 | Android Application UI & UX 구현 |

* 1. **비용(수정 할 것인지) (형직수정) (?? 적당히 붙일 말좀…)**

|  |  |  |  |
| --- | --- | --- | --- |
| **이름** | **설명** | **개수** | **총 비용** |
| 아두이노 메가 2560 | 도어락 제어 및 센서 값을 받기 위한 장치 | 1 개 | 32,000 원 |
| 라즈베리파이3  bisic kit | 카메라 제어 및 서버와 안드로이드, 아두이노 통신을 위한 장치 | 1 개 | 63,000 원 |
| 부저(경보기) | 침입 탐지시 경보를 울리기 위한 부저 | 1 개 | 6,500 원 |
| 솔레노이드 푸시풀 | 최루탄 발사를 제어하기 위한 모터 | 1 개 | 8,200 원 |
| 도어락 | 원격으로 문을 제어하기 위해  아두이노와 연결 될 도어락 | 1 개 | 50,700 원 |
| 파이 카메라 | 스트리밍 및 사진 촬영을 위한 카메라 | 1 개 | 30,000 원 |
| 인체감지센서 | 침입을 탐지 하기 위한 센서 | 2 개 | 17,800 원 |
| 브레드보드 400홀 | ?? | 10 개 | 6,000 원 |
| 점퍼 케이블(암수) | ?? | 1 개 | 3,500 원 |
| 점퍼 케이블(수수) | ?? | 1 개 | 3,500 원 |
| 점퍼 케이블(암암) | ?? | 1 개 | 3,500 원 |
| 블루투스 통신모듈 | ?? | 1 개 | 11,000 원 |
| 적외선 거리 센서 | 민감한 인체 감지 센서를 대신해서 침입을 탐지 하기 위한 센서 | 1 개 | ?? 원 |
| 데모 환경 구현 | 데모용 집 | 1 개 |  |
| 합계 | 개집 적외선거리센서 제외 262,600원 | | |

1. **성과**
   1. **특허, 소프트웨어 등록((전체- 이거 안했는데 생략함??))**
   2. **학술 논문 발표, 정량화 될 수 있는 기타 성과**
   3. **교내 공모전 및 수상(이거 사진좀 다시 찍어 줄 수 있는지)**

****

1. **결론 – post mortem**
   1. **프로젝트 진행 과정에서 잘 된 점과 그 이유**

프로젝트 진행 과정에서 잘된 부분은 첫번째로 하드웨어 간의 통합이다. 초반에 application과 아두이노, 라즈베리 파이 간의 데이터 교환에 어려움을 겪었는데 우린 web server를 통해 각 데이터 교환을 하고자 하였다. 하지만 실제로 팀원 중 아무도 서버를 다뤄 본 적이 없었기 때문에 이를 해결하기 위해 많은 시간을 보내야 했다. 와이파이, 블루투스 등의 시도를 했으나 소켓 통신으로 이를 해결 할 수 있었다.

두번째로 데모를 보여 주기 위한 시설물 제작이다. 실제 집 모형과 똑같은 시설물을 구비하여 필요한 공간인 문과 창문에 도어락 과 최루 가스, 카메라 등을 설치 할 수 있었다. 이를 통해 사람들에게 우리 프로젝트의 각 기능을 이해 시키는데 큰 도움이 되었다. 실제 SW Greative Concert에서도 사람들의 좋은 반응을 받을 수 있었다. 눈에 보이는 구체적인 결과물을 만들었다는 것이 잘한 부분이라고 생각한다.

* 1. **프로젝트 진행 과정에서 아쉬운 점**

본 프로젝트의 기능 중 하나인 침입 탐지를 하기 위해 적외선 거리 센서를 이용했다. 우리가 산 센서의 민감도는 크지 않았기 때문에 침입 탐지를 하는 과정에서 많은 시행착오가 있었다. 따라서 하드웨어 장비를 살 때 하드웨어의 성능을 잘 고려했다면 지체 시간을 좀 더 줄일 수 있었을 것이다. 또한 중간중간 프로그램을 테스트 하는 과정에서 이유 모를 오류가 계속 났는데 원인을 찾지 못해 시간이 많이 지체 되었다. 나중에 소프트웨어의 문제가 아닌 하드웨어의 문제임을 알고서 하드웨어 설치 시, 정비를 철저히 해야 함을 깨달았다.

그 외에도 스케줄 관리에서 서버 부분을 제일 먼저 설계를 해야 한다는 것을 몰랐기 때문에 서버 부분을 먼저 설계를 했다면 스케줄 관리를 좀더 잘했을 것이다.

* 1. **개선 아이디어(수정 할 것)((전체 수정해주씨오))**

사용자 관리, 등록, 인증 – 중요하게 생각하지 않았던 부분이 였지만 ((형직,진성))아마존 웹서비스의 firebase를 통해 관리 하기로 하였다. 또한 사용자 등록 시 라즈베리 파이 시리얼 번호를 입력 받아 기기 인증을 하였고, 이를 바탕으로 라즈베리 파이와 서버간의 통신을 하였다.

라즈베리 파이, 아두이노 통신 부분 – ((혁정- 잘 모르겠음 )) 비교적 전달하는 데이터가 적은 아두이노를 라즈베리 파이와 직접 통신 하게 하였으며 결국에는 서버와 라즈베리 파이 간의 ~~~~한 통신으로 개선하였다.

스트리밍 – 기존에는 탐지가 되었을 시에만 확인 할 수 있도록 했으나 별도의 메뉴로 해당 기능을 추가해서 사용자가 언제든지 집안의 실시간 스트리밍 영상은 확인 할 수 있게 였다.

메뉴 ui 수정 – navigation drawer을 통해 사용자가 메뉴를 확인 하는데 불편함이 있다고 판단하여 메인 화면에 각 기능에 접근하기 쉽도록 해주는 아이콘들을 배치함

* 1. **발전 방향**
     1. **본 과제의 관련 분야에 대한 기여도**

ㅇㅇㅇㅇ

* + 1. **기술적 관점에서의 발전 방향**

ㅇㅇㅇㅇㅇㅇ

* + 1. **사업적 관점에서의 발전 방향 : 향후 상용화 가능성**

ㅇㅇㅇㅇㅇ

* 1. **기타 소감 (각자 소감 쓰기)**
* **권 혁정 :**

한 학기 동안 SW캡스톤디자인 강의를 들으며 프로젝트 기획, 요구 분석, 아키텍쳐 및 상세 설계, 구현, 테스팅, 유지보수 등 개발 프로세스 전 범위를 경험해볼 수 있었다. 특히 마이크로프로세서, 센서 등의 하드웨어를 제어해보며 임베디드 소프트웨어와 사물 인터넷 분야에 적용 가능한 기술을 습득했다. 특히 개발 프로세스뿐만 아니라 프로젝트 관리, 품질보증이라는 중요한 부분도 접해본 소중한 경험이었다. 특히 좋은 동료들이 누구 하나 빠지지않고 열심히 참여해준 점이 고맙다.

* **김 다미 :**

이번 학기SW캡스톤 디자인 강의를 통해 프로젝트의 기획부터 완성을 수행했다. 처음 기획 단계에서 주제를 정하기 위한 의견을 나눌 때, 한 학기 내내 끌고 가야할 프로젝트인 만큼 신중하게 정하고자 하였다. 핵심 기술이 이미 많이 상용화한 IOT였기 때문에 과연 우리의 프로젝트를 기존에 존재하는 시스템과 어떻게 차별성을 줘야 할지 고심해야 했다. 우리 팀의 기능이 다른 팀들에 비해 간단해 보였기 때문에 부가 기능을 계속 추가하려 했다. 이 과정에서 방향성을 잃어 갔다. 하지만 교수님과 멘토 분 들의 조언으로 정말 필요한 기능만을 추가할 수 있었다. 특히 초반에 들은 강의에서 들은 내용을 바탕으로 ‘타겟’에 집중하기로 해 이 어려움을 헤쳐나갈 수 있었다고 본다. 프로젝트 구현과정에선 학교를 다니면서 배운 것들을 넘어서는 지식이 필요했다. 이 과정에서 많은 시행착오를 겪었다. 각 모듈의 데이터 교환이 관건이 였는데, 해당 과정을 미리 설계하지 않은 점이 우리의 실수였다. 이때 하나의 프로젝트를 완수 하기 위해서는 고려해야 할 사항들이 매우 많으며, 시행 착오를 줄이기 위해서는 기획 단계부터 섬세 하게 따져봐야 한다는 것을 깨달았다. 첫 데모를 앞뒀을 땐 정말 막막했지만 팀원 모두 힘을 모아 의견을 나누며 완성도를 높이고 결과적으로 데모 발표에서 기획한 기능의 대부분을 선보일 수 있었다. 또한 SW concert 에서 우리 프로젝트를 소개할 때 매우 자랑스러웠다. 기대도 하지 못했던 장려상을 받아 매우 행복했다. 이 강의를 통해서구현을 위한 개발 지식 습득 외에도 프로젝트에서 배운 일정 관리, 커뮤니케이션은 졸업 후 현업에서도 유용하게 것이다. 팀 프로젝트 외에도 다양한 분야의 현업 멘토 분들의 조언을 들을 수 있어서 좋았다. 특히 초반에 Agile SW development 에 대한 강의는 인턴 시절 실무에서 겪었던 것이라 많은 공감이 되었고, 그 외에 많은 특강들이 매우 유용했다. 더 잘할 수 있었는데 하는 아쉬움도 남지만 학교를 다니면서 수행한 프로젝트 중 제일 보람을 느꼈다. 끝으로 졸업을 앞두고 너무 고생한 지금의 좋은 팀원들과 이런 좋은 강의를 함께 듣고 학교 생활을 잘 마무리 할 수 있어 감사하다.

* **김 형직 :**

이번 학기 SW 캡스톤 디자인을 들으면서 팀원들과 약 3개월간 프로젝트를 수행했다. 프로젝트를 진행하면서 아쉬웠던 점은 첫번째로 나의 준비 상태였다. 주제를 선정할 때 교수님도 호평할 만큼 좋은 아이디어도 나왔지만 지금 우리의 수준으로는 구현이 힘들 것 같아 포기를 했었다. 그래서 아쉬웠고 좋은 아이디어를 놓치기 않는 준비된 사람이 되기 위해 좀 더 공부를 열심히 해야겠단 생각이 들었다. 두번째로 커뮤니케이션이다. 기획 보고서를 작성하던 중 의견이 종합이 덜 되어서 같은 내용을 적는데 a가 쓴 부분은 1이고 b가 쓴 부분은 2가 되는 경우가 있었다. 협업을 할 땐 좀 더 명확하고 의견을 종합하고 결정 해야겠다는 생각이 들었었다. 아쉬운 점과는 반대로 좋았던 점은 첫번째로 좋은 팀원을 만나서 프로젝트를 진행한 점이다. 내가 뒤늦게 서버 개발을 시작했음에도 조급해하지 않고 기다려 주고 다독여준 점이 좋았었다. 두번째로는 두번째로 상을 받았다는 점이다. 처음에는 상을 기대하지도 않았고 중간 평가전 교수님에게 지금 프로젝트 진행상황이 굉장히 안좋다는 말까지 들었었다. 하지만 팀원들과 단합해서 열심히 데모부터 준비한 결과 좋은 평가를 받았고 이에 탄력을 받아 전시회 때는 상까지 받을 수 있었다. 비록 작은 상이지만 우리가 열심히 노력한 결과라고 생각해서 좋았었다. 이처럼 아쉬운 점과 좋은 점을 적어 봤는데 아무튼 해피엔딩으로 끝나 너무 좋았고 또 실제 기획부터 결과물까지 만들어본 프로젝트 경험은 사회에 나가 일을 진행하는데 큰 도움이 될 것 같다.

* **오 동성 :**
* **유 진성 :**

1. **참고 문헌**
2. **해당 프로젝트 GitHub 링크**