

# 항공 소프트웨어 프로젝트

## 최종 보고서

[해킹에 의한 사생활 유출 방지 홈 캠]



# SECAM

팀장	함기성	임베디드-총괄	201801043
팀원	박성훈	APP	201801005
팀원	이대호	관제시스템	202000969
팀원	조성진		201801035
제출일	2023.12.04 (월)		
담당 교수님	정준호 교수님		

# 목 차

## 서론

---

1. SECAM 소프트웨어 개발 보고서 개요	1
2. SECAM 개발 소프트웨어에 대한 개략적 설명	1
3. 대상 소프트웨어 개발 선정 이유	2
4. 사용자 요구사항	3
4.1. 사용자 요구사항에 따른 개발 목적	3
4.2. 개발 목표	3
4.3. SECAM 개념도	4
5. SECAM 개발 환경	4
6. SECAM 개발 일정	5

## 본론

---

7. CSCI	6
8. CSC	6
8.1. 카메라(하드웨어)	6
8.2. 서버(소프트웨어)	7
8.3. 관제 시스템(소프트웨어)	7
8.4. 어플리케이션(소프트웨어)	8
9. CSU	9
9.1. 카메라	9
9.2. 서버	12
9.3. 관제 시스템	12
9.4. 어플리케이션	14
10. 검증 결과 기술	15
10.1. 카메라	15
10.2. 서버	16
10.3. 관제 시스템	17
10.4. 어플리케이션	19
11. 통합 시험 결과 기술	20
11.1. 전원 인가	20
11.2. 클라이언트 접속	20
11.2.1. 영상 저장 작동	20
11.2.2. 카메라 제어 작동	20
11.2.3. 로그 기능 작동	20

목 차

결론

---

과정 요약	21
개발 결과	26
기대 효과	27
SECAM 프로젝트를 마치며	28

## 1. SECAM 소프트웨어 개발 보고서 개요

서론에서는 개발 소프트웨어에 대한 개략적 설명과 대상 소프트웨어 개발 선정 이유, 사용자 요구사항, 사용자 요구사항에 따른 개발 목적 및 목표, 파트 별 개발 환경, 사용 프레임 워크, 개발 언어 등을 간략하게 소개할 것이다.

본론은 소프트웨어 구성도(CSCI / CSC / CSU)을 제시하고 소프트웨어 형상항목(CSCI), 소프트웨어 구성품(CSC), 단위 소프트웨어(CSU)의 기능 및 내용을 기술한다.(단, 기술이 가능한 범위까지 기술한다. 또한 호칭의 편의를 위해 소프트웨어 형상항목, 구성품, 단위 소프트웨어를 각 단어의 영문 약자인 CSCI, CSC, CSU로 칭한다.) 다음으로 CSCI / CSC / CSU 검증 환경 및 검증 결과를 기술한 다음 통합 시험 결과를 기술한다.

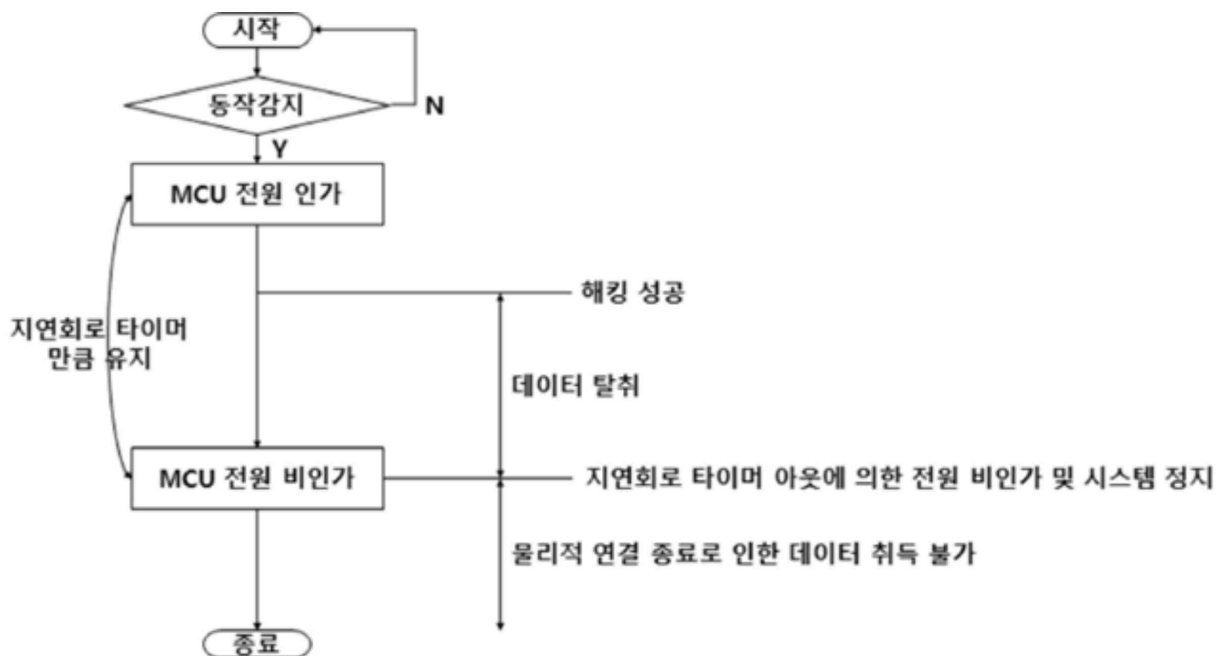
결론은 SECAM의 개발 회고를 기술하고, 항공 소프트웨어 프로젝트 I (캡스톤 디자인) 과목 종료 후 일정표 및 유지보수 및 추가 개발 사항들을 기술한다.

## 2. SECAM 개발 소프트웨어에 대한 개략적 설명

SECAM이 개발한 항목은 ‘해킹에 의한 사생활 유출 방지 홈 캠’이다. 해킹의 근본적 원인인 네트워크 연결을 통제해 해킹의 위협을 최소화하고 해킹을 통해 일어나는 개인 정보 침해 및 범죄 예방이 목표다.

네트워크 연결 통제를 위해 외부센서를 이용해 인체 감지가 되는 경우, 홈 캠의 전원을 인가하는 아키텍처를 활용해 네트워크 연결을 통제한다. 위 아키텍처는 홈 캠 전원 통제 트리거가 MCU(마이크로 컨트롤러 유닛)와 분리돼 있고 물리적으로 작동해 외부에서 악의적 조작이 불가능한 장점이 있다. 전체적인 동작은 아래 그림과 같이 동작한다.

그림 1 SECAM 청사진



### 3. 대상 소프트웨어 개발 선정 이유

4차 산업 시대에 접어들며 스마트 홈 시장은 점차 성장하고 있다. 또한 주거침입 사례가 증가하며 스마트 홈 보안 시장 또한 큰 성장률을 보인다. 한국인터넷진흥원에서 조사한 '사물인터넷 제품 및 서비스 이용 유형 (복수 응답) 중 홈 캠은 27%로 단일 항목 중 가장 큰 비율을 차지하고 있다. 한국인터넷진흥원에서 제시한 IoT기기 해킹 방지법은 제조사 펌웨어 수시로 업데이트, 기기 전원 자주 껐다 켜기 등 사용자가 번거로운 행위를 요구한다. 하지만 SECAM의 아키텍처로 해결이 가능해 주제로 선정했다.

SECAM 아키텍처 개발 전, 동일한 제품 혹은 논문이 존재하는 지 확인 차 홈 캠 시장 및 학회 조사 결과, 외부센서를 이용한 상황 인지 및 카메라 통제 방식이 유사하나 기존 기술들 모두 내부 카메라 관리자에 의해 사생활 노출을 방지하는 방식이라 해킹에 의한 위협을 방지하지 못한다는 점으로 SECAM에서 개발한 방식과 차이점이 있다. 조사한 사례 중 SECAM과 가장 유사한 3가지 사례를 SECAM과 비교하여 간략하게 표로 나타냈다.

표 1 사전 조사 비교

구분	SECAM	특허 출원번호 10-2005- 0030056	특허 출원번호 10-2013- 0059315	특허 출원번호 10-2014- 091380
내용	해킹에 의한 사생활노출 방지를 위한 전원 통제	비상신호가 전달될 경우에만 기록 매체에 접근	사전에 설정된 상황 발생 시 촬영된 영상 관리자에게 전달	사생활 보호를 위한 침입자 감지센서의 신호로 촬영을 시작
공통점	외부 센서를 활용한 상황 인지 및 상황 인지를 통한 카메라 통제 방식			
차이점	1. 3개의 특허 모두 카메라 관리자로부터의 사생활 보호 2. 상시 네트워크와 연결되어 해킹의 위협 존재			

#### 4. 사용자 요구사항

기존 IoT시장에 있는 홈 캠이 제공하는 기능을 이용하면서, 외부 해킹에 의한 사생활 유출을 예방할 수 있는 홈 캠을 원한다.

##### 4.1 사용자 요구사항에 따른 개발 목적

위 언급한 사용자 요구사항과 기존에 없던 SECAM만의 아키텍처를 더하면 외부 해킹에 의한 사생활 유출 예방이 가능해 SECAM아키텍처 홈 캠을 개발하려고 한다. 기존의 홈 캠은 상시 네트워크에 연결되어 있다는 점과 홈 캠에서 촬영중인 영상과 저장한 영상을 사용자가 아닌 홈 캠 제공자가 관리한다는 점이 사생활 유출에 문제점 중 하나이다.

네트워크와의 연결과 촬영중인 영상은 물리적인 요소다. 저장한 영상에 관한 관리 권한을 사용자에게 위임하여 홈 캠을 직접 해킹하거나 서버를 직접 해킹하여 사용자의 영상이 해킹 당하는 경우를 제거하였고, 저장 영상을 해킹하는 것은 더 큰 범죄에 해당하여 SECAM은 해당 범위까지의 보안을 책임지지 않는다. 하지만 사용자가 저장한 영상 자료들을 직접 관리해 녹화 영상 해킹에 관한 상황에 직접적으로 방어 할 수 있다.

##### 4.2 개발 목표

개발 목표는 다음과 같다. 첫 번째 기존 시장에 있는 ‘홈 캠’의 기능을 포함한다. 두 번째 SECAM만의 아키텍처를 추가한다. 두 가지 사항을 모두 포함하는 SECAM의 홈 캠을 제작하는 것이 목표이다. 표로 SECAM이 구현 목표인 기능 및 기능을 간략히 설명하겠다.

표 2 SECAM 홈 캠 기능 및 기능 설명

기능	기능설명
영상 촬영	적외선 센서로 침입자가 식별된 경우, 홈 캠의 전원을 인가해 영상 촬영
영상 정보 실시간 스트리밍	홈 캠이 촬영중인 카메라 웹 서버를 통해, 실시간 스트리밍 서버를 통해 사용자에게 영상정보 제공 (관제시스템 · APP)
영상 저장	사용자가 저장하기 원하는 영상만 사용자의 PC와 같은 물리적 저장 장치에 저장
서버	침입 감지 · 카메라 작동 및 제어 · 오류 로그 DB에 저장(MongoDB), RTSP 실시간 스트리밍, 감지 신호 송신
안드로이드 어플리케이션	QR 코드 : WiFi 정보 카메라로 전달 블루투스 : 사용자 단말과 홈 캠 블루투스 연결을 통해 사용자와의 거리 판단 후, 카메라 전원 통제

### 4.3 SECAM 개념도

4.1 사용자 요구사항에 따른 개발 목적과 4.2 개발 목표를 모두 수렴하여 만든 SECAM의 개념도는 다음 아래 사진들과 같다. SECAM은 해당 개념도를 통해 설계하였으며, 설계를 기반으로 구현했다.

그림 2 SECAM 소프트웨어 개념도

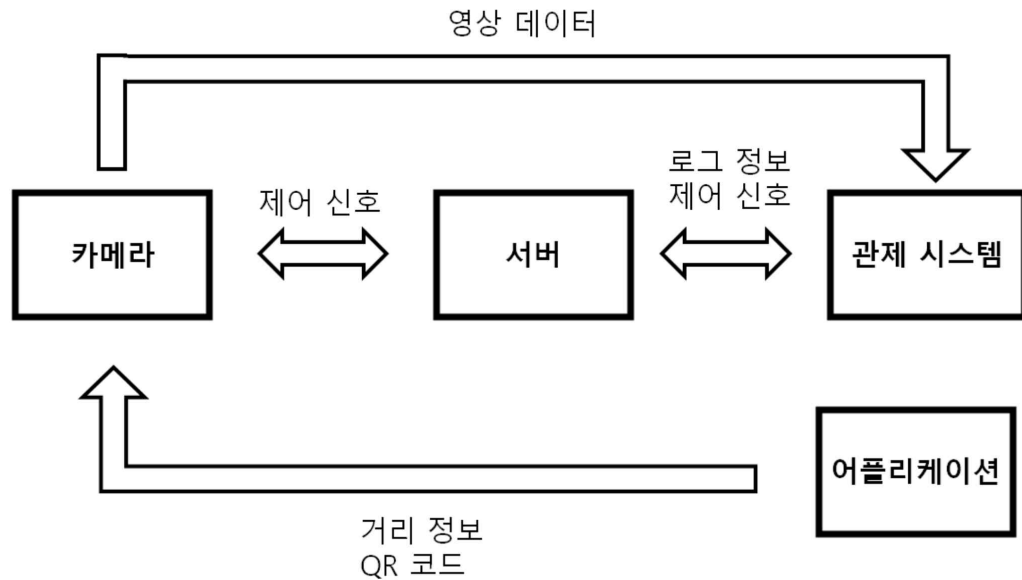
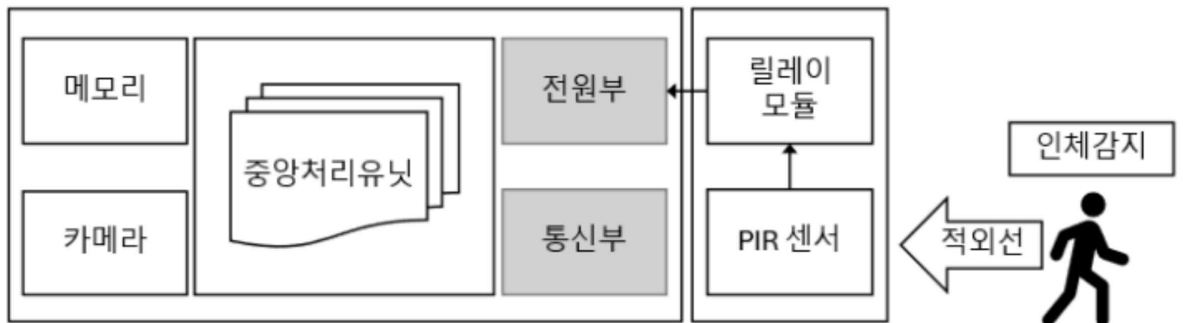


그림 3 SECAM 하드웨어 개념도



## 5. SECAM 개발 환경

개발 환경, 언어 및 프레임 워크를 표로 기술.

표 3 SECAM 개발 환경


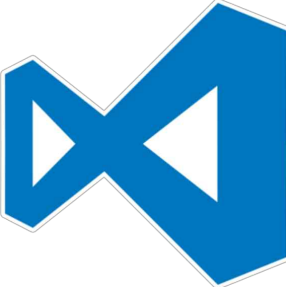
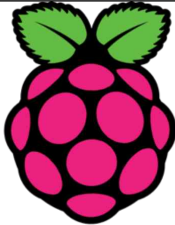







임베디드	관제 시스템, APP	서버
 Arduino IDE	 Visual Studio Code	 Raspbian OS

표 4 SECAM 개발 언어 및 프레임 워크

임베디드	서버, 관제 시스템, APP					
						
C Language	HTML	CSS	Java Script	node.JS	APP Inventer	Mongo DB

\*두꺼운 글씨는 프레임 워크를 의미한다.

## 6. SECAM 개발 일정

SECAM 팀의 개발 일정은 표로 기술한다.

표 5 SECAM 개발 일정

구분	1분기	2분기	3분기	4분기
사전 조사 및 개발 계획 수립				
스트리밍 서버 구축				
서버 및 관제 시스템 기능 구현				
위치 전달, 카메라 각도 조절 구현				
하드웨어 프로토 타입 조립				
구현 기능 테스트				
어플리케이션 구현				
통합 테스트 및 최종 완성				



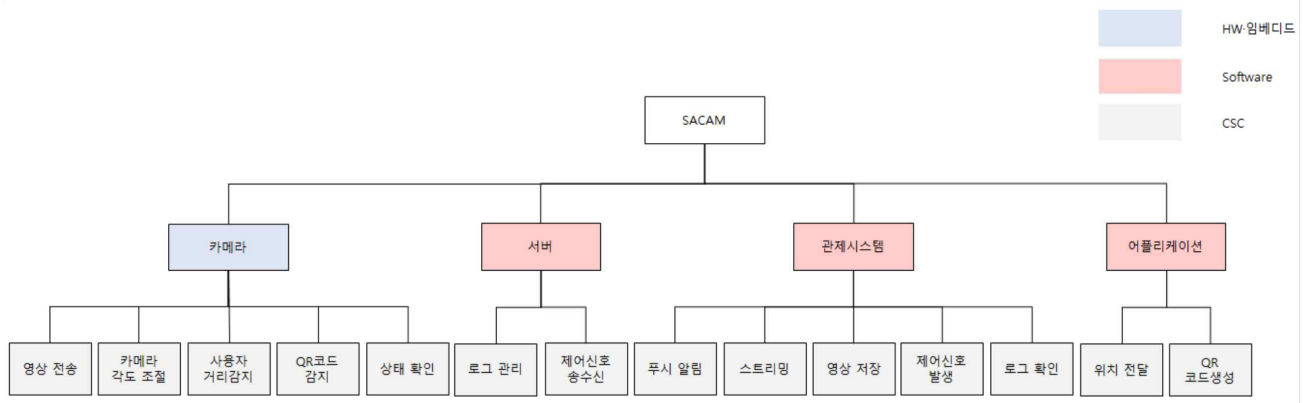
## 본론

본론에서는 CSCI, CSC, CSU 및 검증 환경 및 검증 결과에 대해서 서술한다. SECAM 팀은 하향식으로 소프트웨어 및 하드웨어 개발을 계획했고 해당 과정에서 나온 산출물인 CSCI, CSC, CSU를 기반으로 기능들을 구현했다. 구현 완료 후, 구현 사항들을 검증을 진행했고, 검증 시 사용했던 검증 환경 및 검증 결과에 대해서 서술 할 것 이다.

## 7. CSCI

SECAM 아키텍처의 CSCI을 사진으로 구현했다. 하늘색은 하드웨어, 붉은색은 소프트웨어다. 하드웨어는 카메라가 있고, 소프트웨어에는 서버, 관제시스템 그리고 어플리케이션이 있다. 그 아래 소프트웨어 형상항목을 구성하는 소프트웨어 구성품은 회색으로 표현했다.

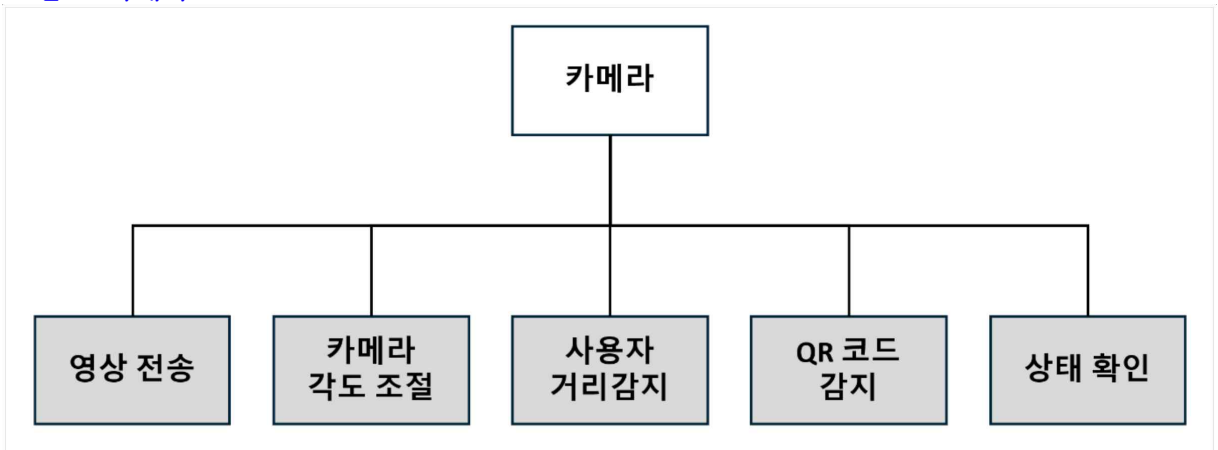
그림4 SECAM 소프트웨어 구성도



## 8. CSC

### 8.1 카메라(하드웨어)

그림 5 카메라 CSC



카메라의 CSC로는 영상 전송, 카메라 각도 조절, 사용자 거리감지, QR 코드 감지 그리고 상태 확인이 있다. 홈 캠의 기본적 기능인 영상 촬영 및 이를 외부에서 확인할 수 있게 하는 영상 전송을 CSC로 선정하였다.

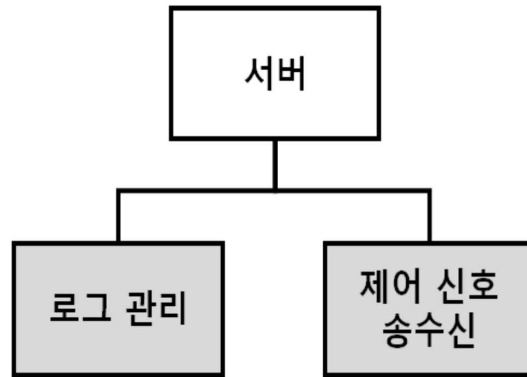
다음으로 카메라의 상하좌우 사각지대를 없애기 위해 카메라 각도 조절을 CSC로 선정하였다. 프로젝트 특정상 카메라의 소유자가 센서에 감지되어도 카메라가 동작 할 수 있기에 이를 구분하기 위해 사용자의 거리감지를 CSC로 선정하였다.

그리고 초기 Wi-Fi의 ssid와 pw 데이터가 카메라에 저장되어 있지 않기에 사용자의 어플로 생성된 QR코드의 감지 및 해석을 CSC로 선정하였다.

마지막으로 현재 카메라의 동작상태를 사용자가 확인할 수 있도록 RGB LED를 활용한 상태확인을 CSC로 선정하였다.

## 8.2 서버(소프트웨어)

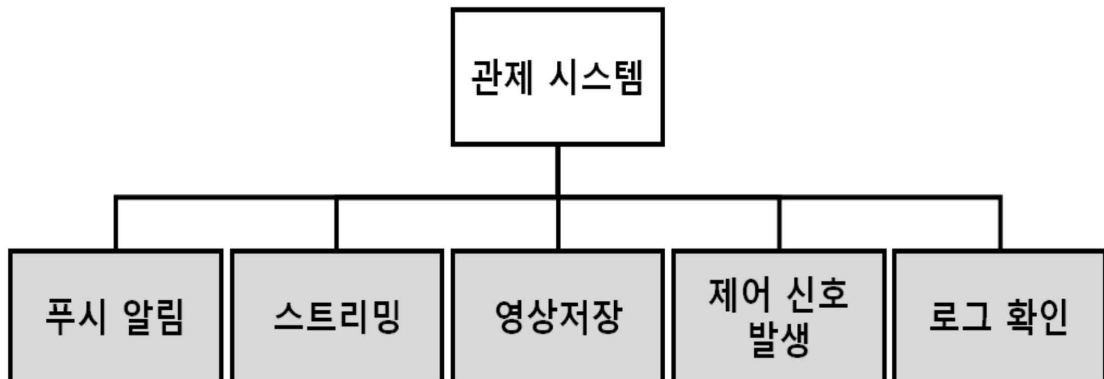
그림 6서버 CSC



서버의 CSC로는 제어 신호 송수신, 로그 관리가 있다. 클라이언트로부터 수신 받은 제어 신호를 카메라로 전달하여 카메라의 각도를 변경하기 위해 제어 신호 송수신 기능을 서버의 CSC로 선정했다. 그리고 로그 저장 기능을 통해 사용자가 이전에 발생한 이벤트, 알림 등의 기록을 확인하고 필요한 경우에 참고할 수 있도록 한다. 또한 홈캠에서 감지된 활동과 이벤트에 대한 로그를 관제 시스템에 제공해야 하고 이를 통해 사용자가 언제 어디서나 특정 시간대에 발생한 이벤트나 활동을 추적하고 분석할 수 있도록 한다. 때문에 로그 관리 기능을 서버의 CSC로 선정했다.

## 8.3 관제 시스템(소프트웨어)

그림 7관제 시스템 CSC



관제 시스템의 CSC로 푸시 알림, 스트리밍, 영상 저장, 저장 신호 송신, 제어 신호 송신, 로그 확인을 선정했다.

홈캠에서 이벤트나 비정상적인 상황이 발생할 때 서버로부터 감지 신호를 수신 받으면 이를 사용자에게 직접적으로 알려야 한다. 그래서 푸시 알림 기능을 관제 시스템의 CSC로 선정하였다.

관제 시스템 CSC에서 스트리밍 기능을 선정한 이유는 다음과 같다. 스트리밍 기능을 통해 사용자는 실시간으로 홈캠에서 촬영되는 영상을 모니터링 할 수 있다. 스트리밍 기능을 사용하면 사용자는 홈캠에서 촬영되는 영상을 실시간으로 보면서 상호작용할 수 있다. 스트리밍 기능은 집의 보안을 강화하는 데 도움을 주기 때문이다. 실시간으로 모니터링하여 도난, 침입 등 발생한 상황에 대해 즉시 파악하고 즉각 조치를 내릴 수 있기 때문이다.

다음으로 선정한 관제 시스템의 CSC에는 영상 저장이 있다. 홈캠의 영상 저장 기능은 영상을 전송하는 것만으로 끝나면 안된다. 사고나 범죄의 증거로 사용되거나 관찰 및 기록을 위하여 저장할 수 있어야 한다. 그래서 우리는 사용자가 원하는 영상을 관제

시스템에 저장하기 위해 영상 저장 기능을 포함했다.

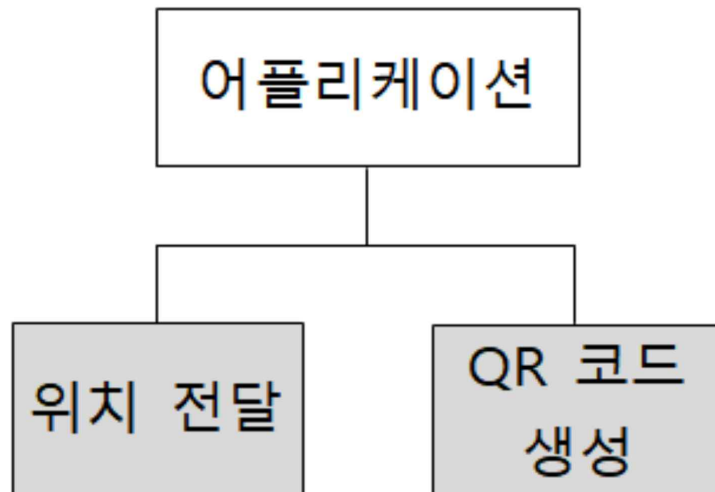
다음으로 저장 신호 송신 기능이다. 실시간 모니터링을 통해 비정상적인 상황이 감지되면 해당 영상을 사용자가 저장을 원할 경우 서버에 저장하기 위해 관제 시스템에서 신호를 송신해야 한다. 또한 증거 보존이나 남기고 싶은 영상을 저장해야 할 때 저장 신호를 관제 시스템에서 전송해야 하기 때문에 이 기능을 관제 시스템의 CSC로 선정하였다.

제어 신호 송신 기능이 필요하다. 그 이유는 원격 제어가 필요하기 때문이다. 카메라의 촬영 각도에서 사각지대를 없애기 위하여 카메라의 움직임을 원격으로 조절해야 한다. 그러기 위해선 제어 신호 송신 기능을 이용하여 사용자가 카메라를 제어할 수 있도록 제어 신호 송신 기능을 관제 시스템의 CSC로 선정하였다.

관제 시스템에서도 로그 확인 기능이 필요하다. 로그는 시스템의 동작 이력 즉, 카메라의 촬영 시작 시간과 같은 것들을 기록한다. 이를 통해 사용자가 특정 시간대의 활동이나 이벤트를 확인하고 상황을 분석할 수 있기 때문이다. 또한 시스템이 제대로 작동하지 않거나 오류가 발생할 경우 로그를 분석하여 조치할 수 있기 때문에 로그 확인 기능을 관제 시스템의 CSC로 선정하였다.

#### 8.4 어플리케이션(소프트웨어)

##### 그림 8어플리케이션 CSC



어플리케이션의 CSC로 위치 전달, QR코드 생성 기능을 선정했다. 위치 전달 기능은 외부의 침입자와 사용자 혹은 그의 가족들을 구분하기 위해 필요한 기능이다. 센서가 침입자와 사용자를 구분하지 못한다면 집에 누군가 들어올 때마다 푸시 알림이 전송될 것이고 그에 따른 좋지 않은 상황이 일어날 수 있기 때문이다.

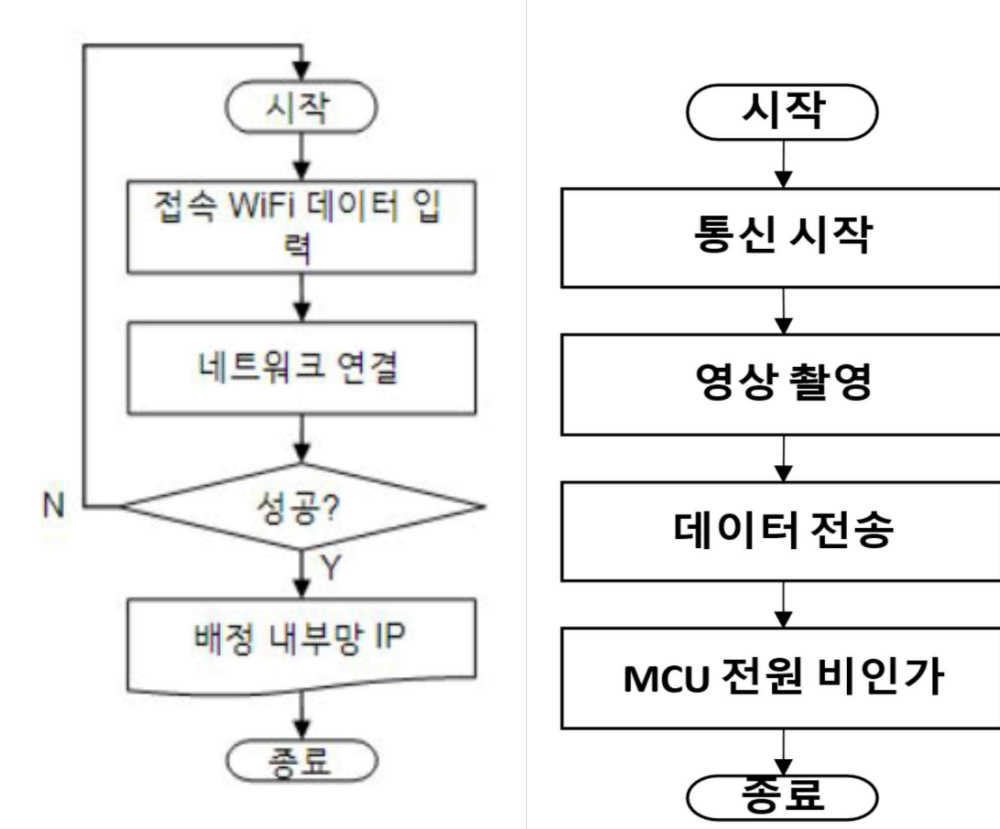
그래서 사용자 핸드폰의 위치를 카메라 아두이노에 전달하여 사용자가 집에 있는 경우에는 외부 센서에 동작이 감지되어도 영상 촬영과 특별한 푸시 알림 기능이 작동되지 않도록 하는 기능이라 어플리케이션의 CSC로 선정하였다.

마지막으로, QR 코드 생성 기능을 선정한 이유는 카메라와 인식 및 연동을 하기 위해선 인증을 해야하는 방법이 필요하기 때문이다. 처음에 생성한 QR 코드가 상시로 출력이 된다면 해킹에 문제가 발생하기 때문에 사용자의 SSID와 PASSWORD를 입력하여 인증에 사용할 때마다 QR코드를 생성하여 해킹을 방지하여 조금 더 안전하게 인증을 하는 방법을 선택하게 됐고 그로 인해 QR 코드를 생성하는 기능을 어플리케이션의 CSC로 선정하였다.

## 9. CSU 기능

### 9.1 카메라

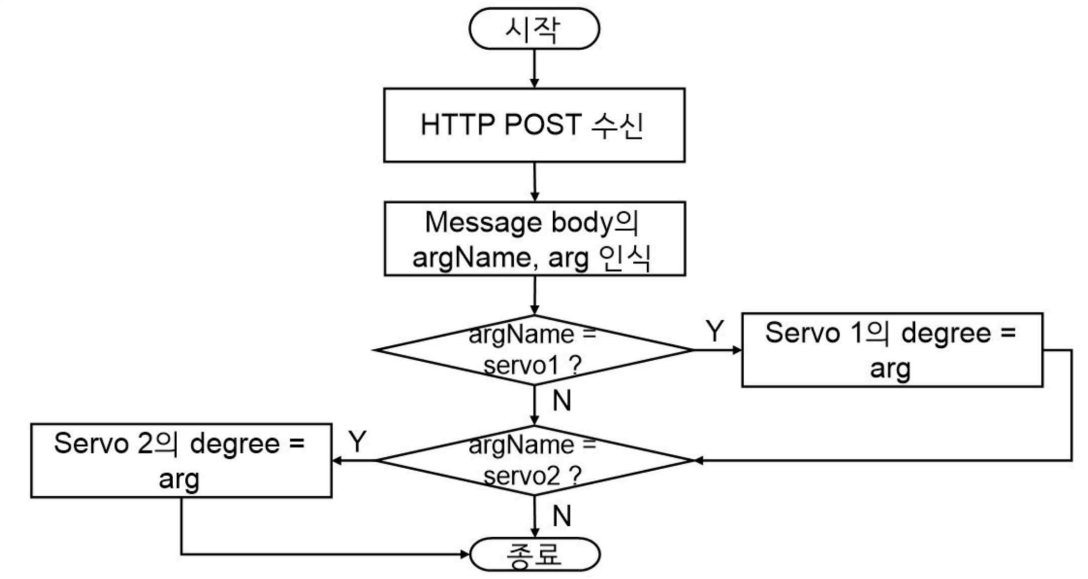
그림 9-10 카메라 영상 전송 Flow Chart Diagram



첫 번째 카메라의 영상 전송 기능이다. 우선 지정된 Wi-Fi 망에 보드를 접속시킨다. 네트워크 연결이 실패한다면 다시 Wi-Fi 망에 보드를 접속시키는 과정으로 돌아가 네트워크 연결 과정을 하게 된다. 이후 MCU 보드를 웹 서버로 하여 촬영되는 영상 프레임을 인터넷을 통해 전송한다.

이 기능을 실현하면 부팅 후 Wi-Fi가 연결되고 웹 서버를 열어 스트리밍을 실행한다. `mjpegCB()` 함수를 통해 `camCB()`, `streamCB()`를 호출하고 웹 서버를 실행시켜 데이터를 전송한다. `camCB()`는 카메라에서 프레임을 읽어와 MJPEG 스트림에 사용될 수 있도록 처리하고 `streamCB()`는 활성 클라이언트에게 MJPEG 스트림을 제공하고, 각 프레임의 크기와 내용을 클라이언트에 전송한다.

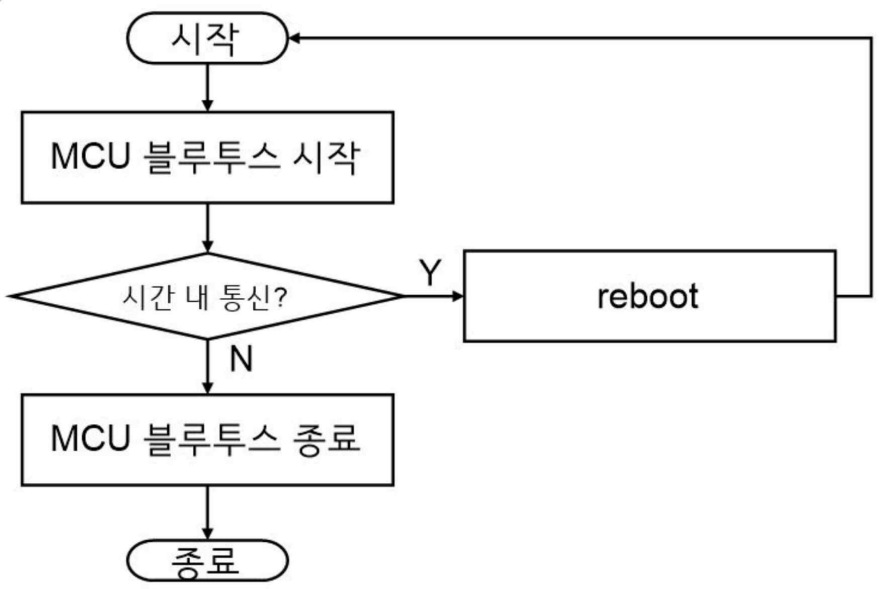
그림 11카메라 각도 조절 기능 Flow Chart Diagram



서버를 보드의 판에 배정하고 펄스 값을 통해 모터의 각도를 조정한다. 그리고 ESP32의 기능을 통해 웹 서버를 열고 이를 통해 URL로 모터 조작에 대한 명령을 이에 맞게 동작하기 위해서 server.on에서 웹서버를 열어 POST 명령을 받을 수 있게 했다. HTTP POST를 수신하고 Message body의 argName, arg 인식한다. 이후에 모터 확인, 각도를 확인 후에 각도를 조절한다.

서버에서는 key-value 데이터를 인식하여 적절한 처리를 수행한다. querystring 모듈을 사용하여 servo2Angle 값을 문자열로 변환하고 이 값을 나중에 서버로 보내기 위한 데이터로 사용한다. servoWrite() 함수에서 PWM 신호로 제어되며 신호가 어떤 주기로 들어왔는지를 인식해 각도를 조절한다. 따라서 모터를 선택하여 얼마나 움직일지 인수로 받아 각도 값을 펄스 값으로 치환해 모터를 제어하도록 한다. argName이 servo1이면 Servo1이라는 서보 모터의 각도를 나타내는 값을 arg로 설정한다. argName이 servo2이면 Servo2이라는 서보 모터의 각도를 나타내는 값을 arg로 설정하여 서보 모터의 움직임을 제어하고 설정한다.

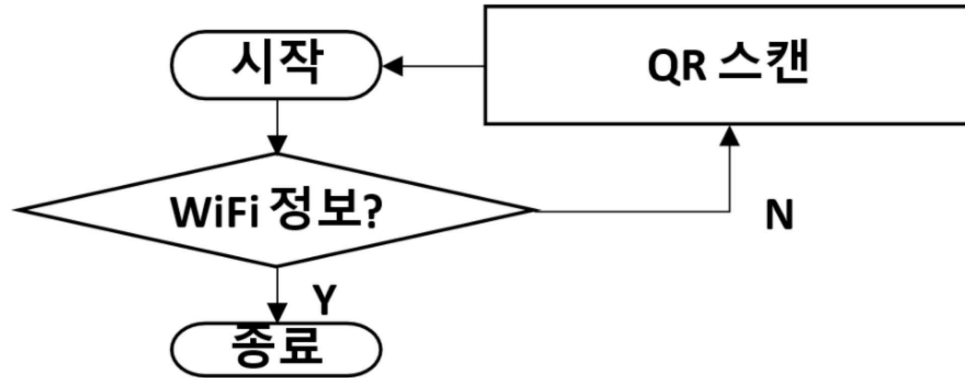
그림 12카메라 사용자 거리 감지 기능 Flow Chart Diagram



사용자 디바이스와 카메라 MCU의 거리가 일정 거리 이하가 되면 블루투스 통신을 시작한다. 카메라 MCU 보드가 어플리케이션을 설치하고 있는 디바이스와 거리가

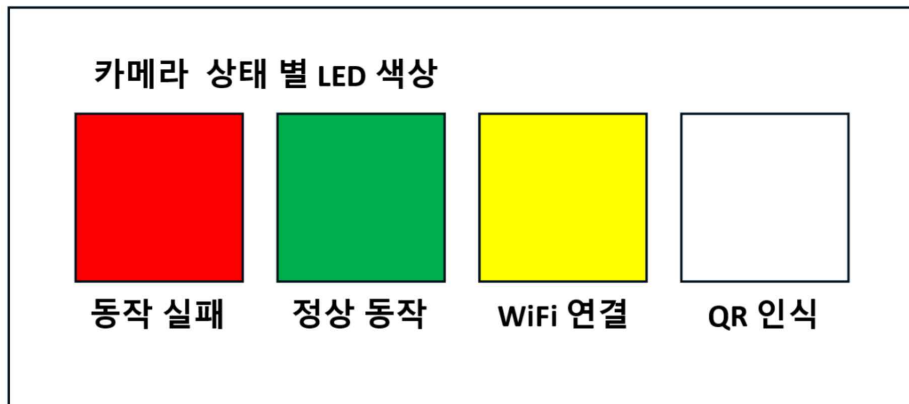
일정 거리에 만족하게 될 때 블루투스 통신이 일정 시간내에 연결되면 카메라로부터 일정 거리 내에 사용자가 있는 것으로 간주하고 MCU를 REBOOT 한다. 블루투스가 켜져 있는 동안 디바이스와 연결이 된다면 MCU 보드의 블루투스는 종료가 되고 연결이 끊긴다. 이 과정을 통해 영상 촬영을 하지 않고 푸시 알림 기능이 작동하지 않는다. 부트 이후 와이파이 망을 이용하여 사용자 디바이스의 어플리케이션과 통신을 한다.

그림 13 QR 코드 감지 Flow Chart Diagram



카메라에 Wi-Fi 초기 정보가 없다면 어플에서 생성된 QR코드를 인식해 해당 데이터를 읽어온다. quirc 라이브러리를 이용해 QR코드 처리를 위한 데이터 구조체 및 변수들을 정의한다. dumpData 함수는 디코딩된 QR 코드 데이터에서 SSID와 Password를 추출해 EEPROM에 저장한다.

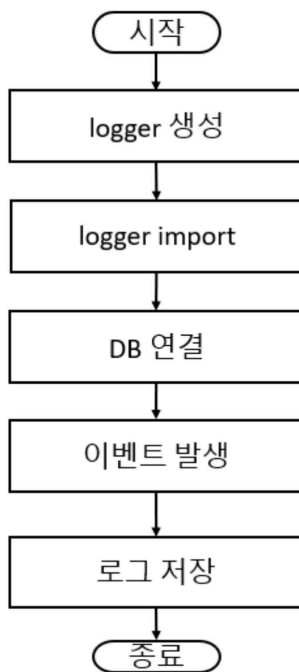
그림 14 상태 별 LED 색상



사용자가 카메라의 동작상태를 확인할 수 있도록 RGB LED의 색상을 이용해 상태를 표시한다. 각 동작 단계마다 LED의 값을 초기화 하고 단계에 할당된 RGB 값을 출력한다.

## 9.2 서버 로그관리

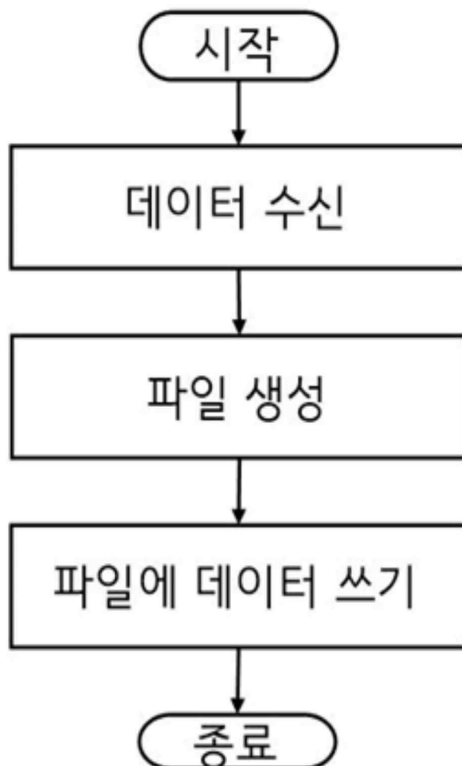
그림 15 서버 로그 관리 기능 Flow Chart Diagram



우선 사용자가 푸시 알림을 받고 핸드폰이나 노트북을 통해 서버와 연결이 되면 로그를 기록한다. 기존의 파일 형태로 로그를 저장했던 방식과 달리 MongoDB를 사용하여 로그 파일 전용 데이터베이스에 저장하는 방식으로 변경하였다. MongoDB에 저장하기 위해서는 2개의 파일(connect.js / ClientLog.js)이 필요하다. ClientLog.js는 데이터베이스의 스키마를 설정하는 파일로, 연도(YMD), 시간(time), 상태(status), 로그 레벨(level) 그리고 로그 발생 횟수 시각화를 위한 로그 요청 횟수(pushCnt)를 스키마로 설정했다. connect.js는 서버와 MongoDB를 직접 연결하는 파일로, 서버에서 connect.js를 호출하면 MongoDB의 도메인 주소와 비밀번호를 가지고 연결을 시도한다. 서버와 DB가 연결된 후, 클라이언트에서 침입 감지, 녹화 시작, 녹화 중지의 3가지 상황이 발생한 경우 fetch() 함수를 통해 로그 내용을 서버로 전달한다. 서버에서는 ClientLog를 사용하여 전달된 로그 내용을 해당 스키마에 대입하여 DB에 저장한다.

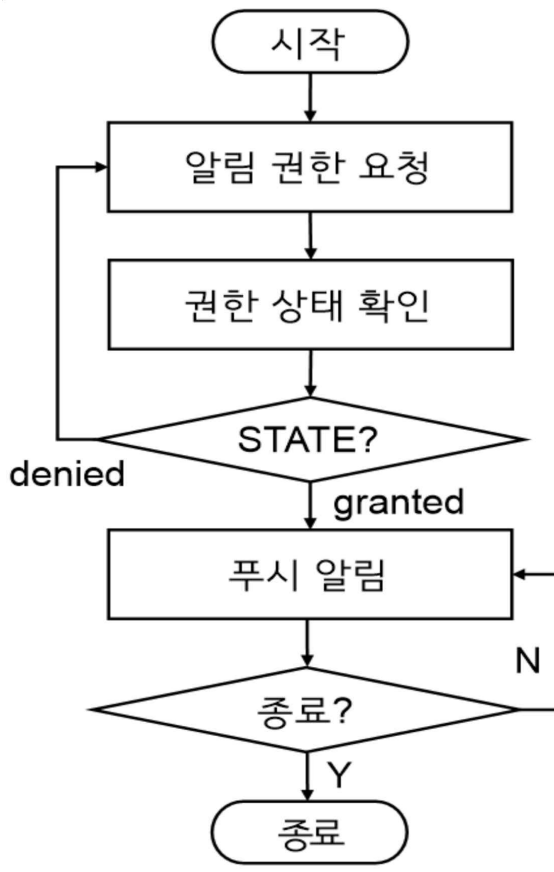
## 9.3 관제 시스템

그림 16 관제 시스템 영상 저장 기능 Flow Chart Diagram



클라이언트의 영상프레임을 전달받아 mp.4 파일로 저장하는 방식을 사용했다. socket.on 함수는 Socket.IO의 이벤트 핸들링을 위한 메서드로, 클라이언트에서 특정 이벤트가 발생했을 때 해당 이벤트에 대한 처리를 정의한다. video-stop은 클라이언트에서 video-stop 이벤트를 전송할 때 호출되고 비디오 녹화를 중지하고자 할 사용된다. Video 이벤트를 사용하여 비디오 데이터를 전달하고 핸들러는 데이터를 받아와서 현재 날짜와 시간 정보를 이용하여 파일명을 생성한 다음, 해당 파일 데이터를 저장한다. socket.emit('save') 서버로 save 이벤트를 전송하고 이벤트는 서버에서 비디오 저장을 시작하도록 알린다. 이후에 MediaRecorder 객체를 생성하여 stream을 녹화하기 위한 녹화기를 만들고 mediaRecorder.start() 함수를 통해 비디오 녹화를 시작한다. 이것을 통해 파일에 데이터를 쓰는 과정이다.

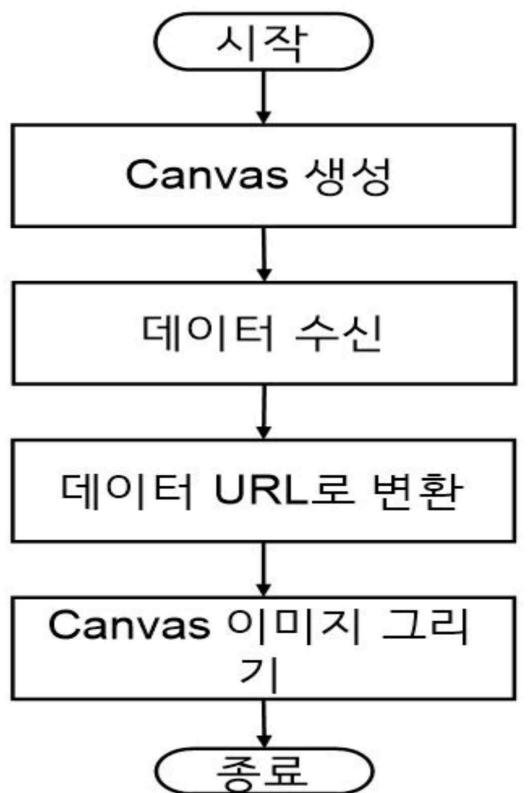
그림 17 관제 시스템 푸시 알림 기능 Flow Chart Diagram



Date 객체를 생성하고, toLocaleString() 메서드를 사용하여 로컬 시간대 및 형식 규칙에 맞는 문자열로 변환하여 date 변수에 할당하고 notification 변수를 선언하여 이 변수에 이후 알림을 저장할 수 있도록 기능을 구현했다. 그리고 Notification을 사용하여 사용자에게 보여주는 알림창인 푸시알림을 뜨게 하여 영상 촬영이 시작됐다는 알림을 제공하기 위해 사용했다. 브라우저에서의 알림 권한을 확인하기 위해 notification 변수에 할당 이후에 알림 권한을 확인한다. 여기에서 푸시알림 권한을 사용자가 결정할 수 있게 되는데 권한 허용으로 설정되어 있으면 푸시 알림이 생성되고 권한이 차단으로 설정되어 있으면 알림 권한 요청 과정으로 돌아가 푸시 알림 권한을 결정하게 되는 동작을 다시 하게 된다. 마지막으로 푸시 알림이 생성이 된 후 닫기 버튼을 누르지 않으면 푸시 알림 창이 꺼지지 않고 닫으면 종료가 된다. 웹 오프라인 상태에서도 푸시

알림 기능이 실행되도록 service\_works.js를 사용하여 이미지나 데이터를 캐시에 저장해서 캐싱 되어있는 데이터를 사용하여 오프라인에서도 실행 가능하도록 했다.

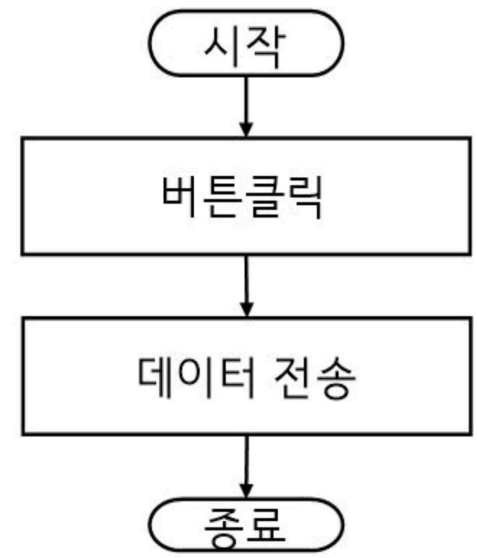
그림 18 관제 시스템 스트리밍 기능 Flow Chart Diagram



카메라의 RTSP 서버에 접속하여 setTimeout(function()) 함수를 사용하여 일정 시간이 지난 후에 새로고침을 수행하기 위한 타이머를 설정한다. delay 변수에 설정된 시간이 지난 후에 location.reload()를 호출하여 페이지를 새로고침 한다. socket.emit함수에서 소켓을 통해 data를 전송하고 데이터는 Buffer로 변환 후에 Base64 형식으로 인코딩되어 전송된다. pipeStream 함수를 통해 데이터를 스트림으로 전달하고 Data를 매개변수로 받게 되면 데이터를 URL로 변환 후에 img.onload 이벤트 핸들러를 사용해서 이미지가 로드되었을 때 실행되도록 하였다. 또한 이미지가 로드되었을 때 그림을 그리는 캔버스 요소에 해당 이미지를 그린다.



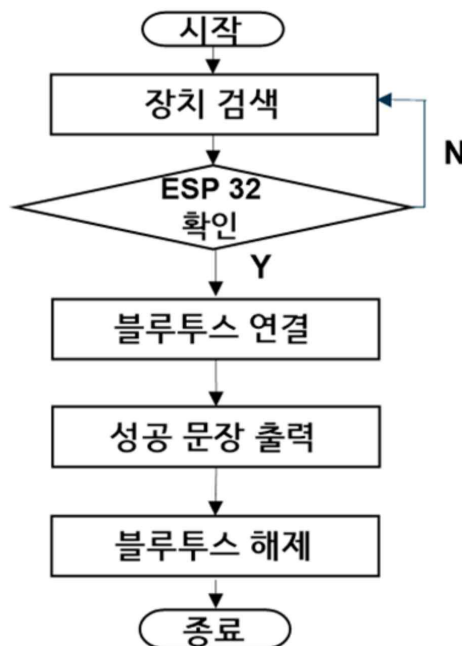
그림 19 관제 시스템 제어 신호 발생 기능 Flow Chart Diagram



socket.on 함수는 Socket.IO의 이벤트 핸들링을 위한 메서드로, 클라이언트에서 특정 이벤트가 발생했을 때 해당 이벤트에 대한 처리가 가능하다. 그래서 인터페이스에서 각도를 조절할 수 있는 버튼을 생성하여 클릭하고 socket.on 이벤트 핸들러를 사용하여 이벤트가 발생하면 서버 제어 관련 함수를 호출하여 서버 모터 제어를 시작한다.

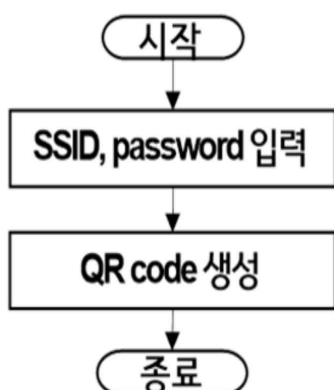
#### 9.4 어플리케이션

그림 20 어플리케이션 사용자 위치 전달 기능 Flow Chart Diagram



사용자 디바이스와 ESP 32와 사용자 거리를 주고 받기 위해 Web Bluetooth API를 사용하여 아두이노 ESP32와 블루투스 연결하는 방식을 이용했다. 메서드를 사용하여 장치를 검색하여 연결이 가능한 블루투스 장치의 목록을 나열한다. 이후 navigator.bluetooth.getDevices() 메소드를 사용하여 이미 페어링된 디바이스 목록을 비동기로 수행한다. 이미 페어링된 장치가 아니라면 해당 장치를 클릭 후에 연결을 할지 말지 정한다. 연결을 진행하고 싶으면 API를 이용하여 연결 후에 GATT 프로토콜을 사용하여 Bluetooth LE 장치 간 통신을 가능하게 하기 때문에 이를 이용하여 거리 정보를 전달하고 연결을 끊고 싶으면 disconnect() 함수를 통해 특성이 존재하거나 연결을 해제하는 경우 연결된 디바이스의 GATT 연결을 해제한다. 최종적으로 해당 동작을 종료한다.

그림 21 QR 코드 생성 Flow Chart Diagram





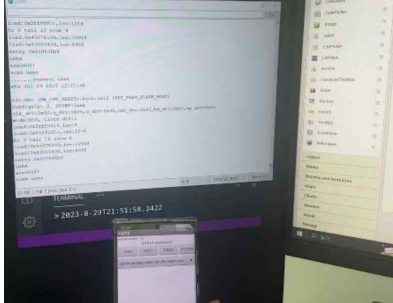
사용자 카메라에서 QR 코드를 촬영하기 위해 앱인벤터에서 SSID와 password를 입력한다. 이후에 프로그램에서 Express 미들웨어를 사용하여 POST 요청의 본문 데이터 파싱하고 처리하여 해석하고 JavaScript 객체로 변환시킨다. 그리고 Express 라우터가 클라이언트로부터 POST 요청을 받아들이고 해당 요청의 데이터를 기반으로 Wi-Fi 정보를 포함하는 QR 코드를 생성하여 클라이언트에게 응답하고 종료하게 된다.

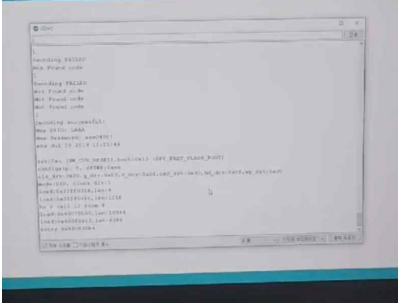
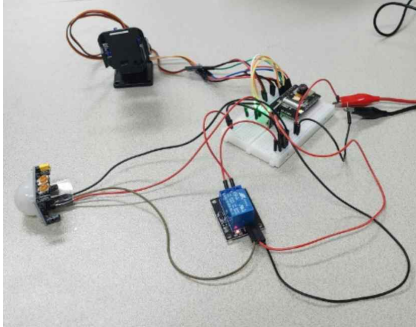
## 10. 검증 결과 기술

### 10.1 카메라

카메라의 검증 목록으로는 영상 전송과, 카메라 각도 조절, 사용자 거리 감지, QR코드 인식, 상태확인 등이 있고, 모든 기능들의 검증이 완료되었다. 영상전송의 경우 카메라에 전원 인가 후, EEPROM 저장된 Wi-Fi 정보를 활용해 웹 서버를 열어 스트리밍이 진행됨을 확인했다. 카메라 각도 조절의 경우 관제 시스템에서 카메라 조작 버튼을 누르면 소켓 통신을 통해 특정 값이 서버로 전송되어 알맞은 함수를 호출함으로써 모터를 조작하는 방식으로 구현했다. 검증 결과 각 버튼에 해당하는 방향으로 카메라 각도가 변하는 것을 확인했다. 사용자 거리 감지는 블루투스 통신을 이용해 사용자의 스마트폰과 카메라가 통신이 연결되면 일정 거리 내에 사용자가 위치하였다 판단해 카메라의 전원을 차단하도록 구현했고 실제 블루투스 연동 시 이 기능이 작동함을 확인했다. QR코드 인식은 초기 Wi-Fi 정보가 카메라 내부에 저장되어 있지 않을 때 QR을 통해 해당 데이터를 전달받도록 구현하였고 어플리케이션을 통해 생성된 QR을 인식해 Wi-Fi에 연결됨을 확인했다. 마지막으로 상태확인인 카메라의 동작 상태를 RGB LED를 활용해 구분할 수 있도록 했다. 각 동작별로 지정한 색상이 표시됨을 확인했다.

표 6 카메라 검증 결과 기술

검증 내용	검증 결과	성공 여부
영상 전송		O
카메라 각도 조절		O
사용자 거리 감지		O

QR 코드 감지		○
상태 확인		○

## 10.2 서버

서버의 검증 목록으로는 제어 신호 송수신, 로그 관리가 있고, 모든 기능들의 검증이 완료되었다. 제어 신호 송수신의 경우 관제 시스템에서 카메라 제어 버튼을 누르면 실제 카메라 각도가 변경되는 것을 확인하였다.

로그 관리의 경우 특정 이벤트 발생 시, 데이터베이스에 해당 로그가 기록되는 것을 확인하였다. 또한 관제 시스템에서 로그 정보를 요청할 경우, 데이터베이스에서 관제 시스템에 로그 정보를 제공하여 관제 시스템 상에 로그 기록이 출력되는 것을 확인하였다.

표 7 서버 검증 결과 기술

검증 내용	검증 결과	성공 여부
감지 신호 송신	2023-05-25 00:18:38 [SECAM] info: 클라이언트 연결	○
로그 관리	<pre>const mongoose=require('mongoose');  const { Schema } =mongoose  const ClientLog = new Schema({   YMD: {     type:String,     required:true,   },   time:{     type:String,     required:true,   },   status:{     type:String,     required:true,   },   level:{     type:String,     required:true,   },   pushCnt:{     type:Number,     default:0,   }, })  module.exports=mongoose.model('ClientLog',ClientLog);</pre>	○

### 10.3 관제 시스템

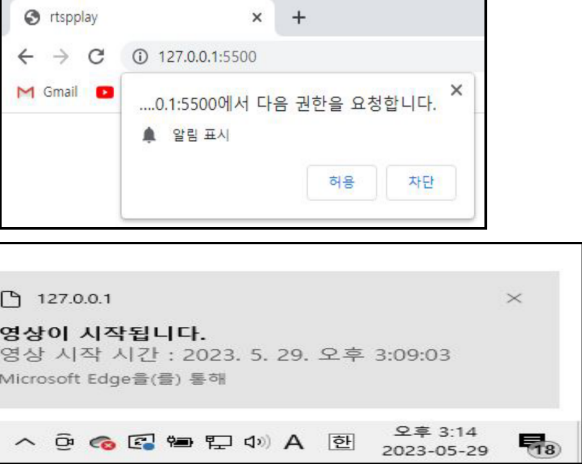
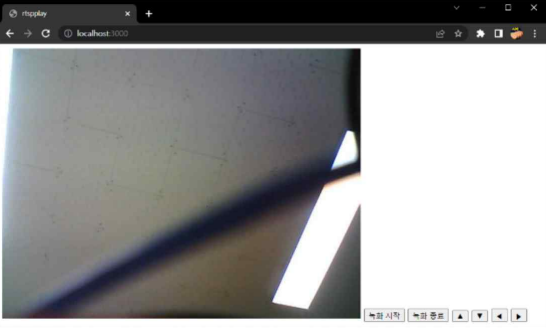
관제 시스템의 검증 목록으로는 푸시 알림, 스트리밍, 영상 저장, 저장·제어 신호 송신, 로그 확인이 있고, 모든 기능들의 검증이 완료되었다. 푸시 알림의 경우 우선 브라우저에서 알림 권한을 허용해야 한다. 일단 브라우저 상에서 알림 권한을 허용하라는 알림 창이 잘 떴고, 허용을 누른 경우 해당 값이 잘 전달되어 푸시 알림이 발생했을 때 알림 창이 뜨는 것을 확인했다. 스트리밍의 경우 우선 카메라의 RTSP 서버에 접속해 영상 데이터 프레임 수신 받고, 이를 소켓통신을 통해 클라이언트로 전송한다.

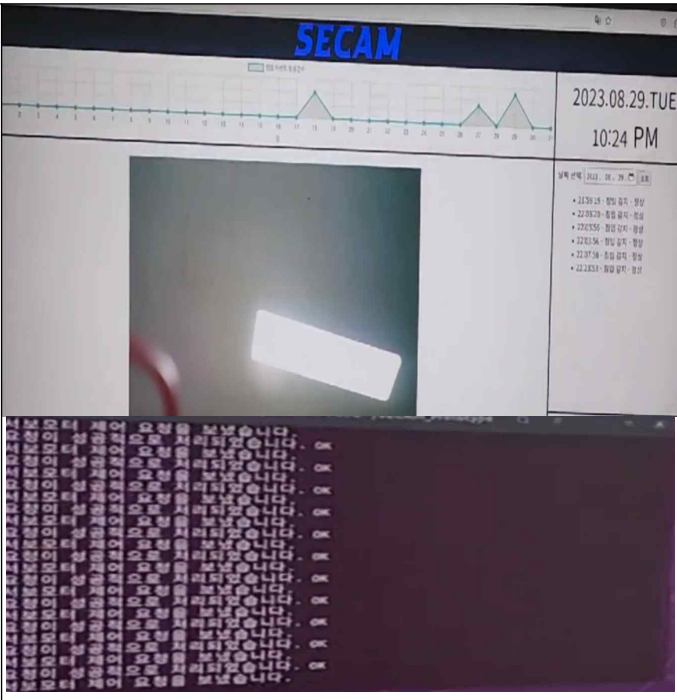
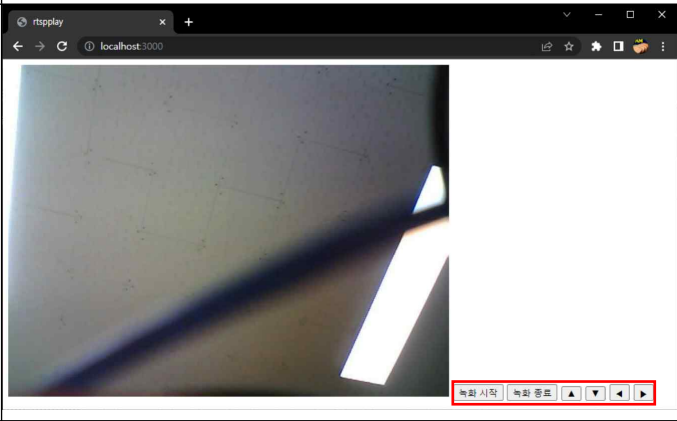
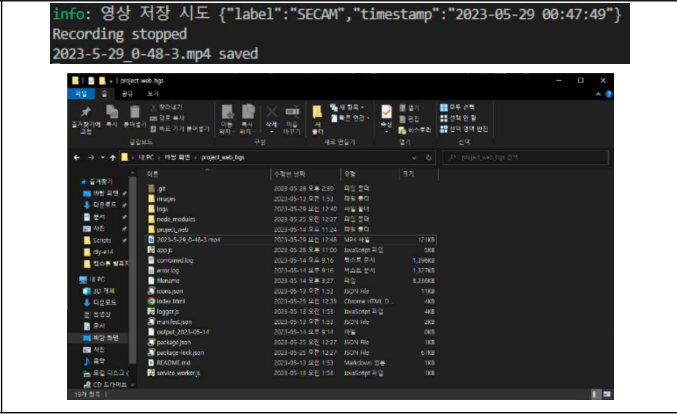
데이터를 수신 받은 클라이언트는 해당 데이터를 캔버스에 출력함으로써 구현된다. 검증 결과 관제 시스템에 녹화되는 영상 데이터가 실시간으로 스트리밍 되는 것을 확인했다.

영상 저장의 경우 관제 시스템에서 영상 저장 버튼을 누르면 관제 시스템 내에 파일을 생성하여 영상을 저장하는 방식으로 구현했다. 검증 결과 웹페이지에서 영상 저장 버튼을 클릭하면 관제 시스템 내의 downloads폴더에 영상이 저장된 것을 확인했다.

로그 확인의 경우 관제 시스템에서 원하는 날짜를 입력하면, 서버에서 데이터베이스 내에 존재하는 해당 날짜의 로그들을 읽어서 관제 시스템에 출력하는 방식으로 구현했다. 검증 결과 원하는 날짜의 로그 파일이 출력되는 것을 확인했다. 저장·제어 신호 송신의 경우 관제 시스템의 저장 버튼과 각종 제어 버튼을 클릭하면 로그가 기록되고, 카메라의 각도가 조절되는 것을 확인했다.

표 8 관제 시스템 검증 결과 기술

검증 내용	검증 결과	검증 여부
푸시 알림		O
스트리밍		O

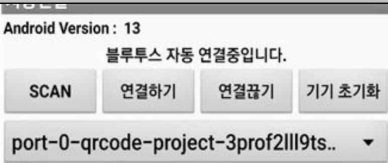
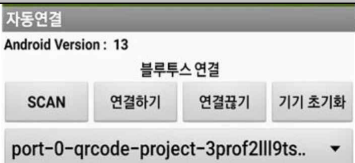


<p>로그 확인</p>	 <p>The screenshot shows the SECAM control system interface. At the top, there is a graph with a blue line and the SECAM logo. Below the graph, there is a list of logs. The logs are in Korean and show the status of the control system. The date and time are 2023.08.29.TUE 10:24 PM.</p>	<p>0</p>
<p>제어 신호 송신</p>	 <p>The screenshot shows a video player window titled 'rtspplay'. The video content shows a control signal being transmitted. The player controls at the bottom are highlighted with a red box.</p>	<p>0</p>
<p>영상 저장</p>	 <p>The screenshot shows a terminal window with the following text:     <pre>info: 영상 저장 시도 {"label":"SECAM","timestamp":"2023-05-29 00:47:49"} Recording stopped 2023-5-29_0-48-3.mp4 saved</pre>     Below the terminal window, there is a file explorer window showing the contents of the 'project-web-logs' directory. The file '2023-5-29_0-48-3.mp4' is highlighted.   </p>	<p>0</p>

## 10.4 어플리케이션

어플리케이션의 검증 목록으로는 푸시 알림, 스트리밍, 로그 확인, 저장·제어 신호 송신, 사용자 위치 전달이 있고, 모든 기능들의 검증이 완료되었다.

사용자 위치 전달의 경우 블루투스를 통해 페어링 대기 중인 장치를 검색해 연결하면 해당 장치와 서버의 거리 정보를 전달하는 방식으로 구현했다. 검증 결과 페이지에 페어링 안내 알림창이 뜨는 것을 확인했다.

표 9 어플리케이션 검증 결과 기술

검증 내용	검증 결과		검증 여부
블루투스 자동 연결			O
QR 코드 생성			O
사용자 위치 전달			O

## 11. 통합 시험 결과 기술

본 절에서는 각 CSC별로 검증한 결과들을 합하여 SECAM 이 운용되는 순서를 따라 전체적인 통합 시험 결과를 기술했다.

### 1. 전원 인가

우선 평상시에 서버는 사용자의 위치 정보를 전달받는다. 사용자가 특정 거리 내에 있는 경우에는 센서에 움직임이 감지되어도 카메라에 전원이 인가되지 않는다. 그러나 사용자가 특정 거리 밖에 있는 경우, 센서에 움직임이 감지되면 카메라에 전원이 인가됨과 동시에 카메라는 영상 녹화를 시작하고, 녹화되는 영상을 실시간으로 서버에 전송한다. 서버에서는 첫 번째 영상 데이터 패킷이 들어오는 순간, 감지 신호를 송신하고 로그를 기록한다. 또한 사용자에게 푸시 알림을 보내 이 사실을 알린다.

이 과정에서 사용자 위치 전달, 영상 전송, 감지 신호 송신, 로그 저장, 푸시 알림 기능이 사용되었으며 모두 검증이 완료되었다.

### 2. 클라이언트 접속

사용자가 클라이언트에 접속하면 카메라는 클라이언트로 영상 데이터를 전송하고, 영상 데이터를 받은 클라이언트는 해당 영상 데이터 패킷들을 관제 시스템에 스트리밍한다.

이 과정에서 영상 전송, 스트리밍 기능이 사용되었으며 모두 검증이 완료되었다.

#### 2.1. 영상 저장 작동

해당 영상을 보는 사용자가 영상 저장의 필요성을 느껴 영상 저장 버튼을 클릭하면 관제 시스템 내의 downloads 폴더 내에 영상이 저장된다.

이 과정에서 영상 저장 기능이 사용되었으며 검증이 완료되었다.

#### 2.2. 카메라 제어 작동

사용자가 카메라의 촬영 각도를 변경하고 싶어 제어 버튼을 클릭하면 서버로 제어 신호가 송신된다. 제어 신호를 수신 받은 서버는 해당 신호에 따라 카메라 각도 조절을 수행한다.

이 과정에서 제어 신호 송신, 카메라 각도 조절 기능이 사용되었으며 모두 검증이 완료되었다.

#### 2.3. 로그 기능 작동

사용자가 이전에 기록된 로그를 확인하고 싶어 원하는 날짜를 입력하고 로그 조회 버튼을 클릭하면, 서버에서 데이터베이스에 접속하여 해당 날짜의 로그를 불러와 관제 시스템에 출력한다. 또한 스키마(테이블)의 요소인 pushCnt(침입 횟수)를 통해 상단에 그래프에 정상적으로 출력되는 것을 확인했다.

이 과정에서 로그 저장, 확인 기능이 사용되었으며 검증이 완료되었다.

## 과정요약

### 1. 대상 소프트웨어 개발 선정 이유

현재 대한민국의 스마트 홈 시장 규모가 커지고 있고, 주거 침입 검거 인원도 증가하는 추세를 보인다. 사물인터넷 제품 및 서비스 이용 유형에서 홈 캠을 제일 많이 사용하는 것을 확인하였고, 기존에 있는 해킹 방지법은 번거롭고 소프트웨어 요소만으로 해결하지 못하는 것을 확인했다. 물리적으로 전원을 차단하는 요소와 녹화한 영상을 사용자가 관리하는 방법으로 필요한 순간에만 영상을 촬영하고 영상이 외부에 있는 것이 아닌 사용자의 물리적 저장장치에 있으므로 외부의 해킹으로부터 비교적 안전하게 지킬 수 있어 해당 소프트웨어를 주제로 선정하였다.

개발 전, 기존의 시장 및 학회를 조사한 결과 외부 센서를 활용한 상황 인지를 통한 카메라 통제 방식은 이미 존재하지만, 카메라 관리자에 의해 사생활 노출을 방지하는 방식과 상시 네트워크에 연결되어 있어 해킹의 위협이 존재하는 약점을 갖고 있어 최종적으로 위 소프트웨어를 개발하기로 했다.

### 2. 사용자 요구 사항

기존 IoT(Internet of Things)시장에 있는 홈 캠의 기본적인 기능(영상 촬영·영상 저장·실시간 스트리밍)을 이용하고, 외부 해킹에 의한 사생활 유출을 예방할 수 있는 홈 캠을 사용하는 것이다.

#### 2.1 사용자 요구사항에 따른 개발

사용자 요구사항과 기존에 없던 SECAM만의 아키텍처를 더하면 외부 해킹에 의한 사생활 유출 예방이 가능해 SECAM 아키텍처 홈 캠을 개발하려고 한다. 침입자 발생 시, 전원을 인가하고 촬영한 영상을 모두 저장하는 것이 아닌 사용자가 저장을 하는 영상만 사용자의 기기(PC, 휴대폰 등)에 영상을 저장하여 사용자가 직접 영상 정보를 관리한다.

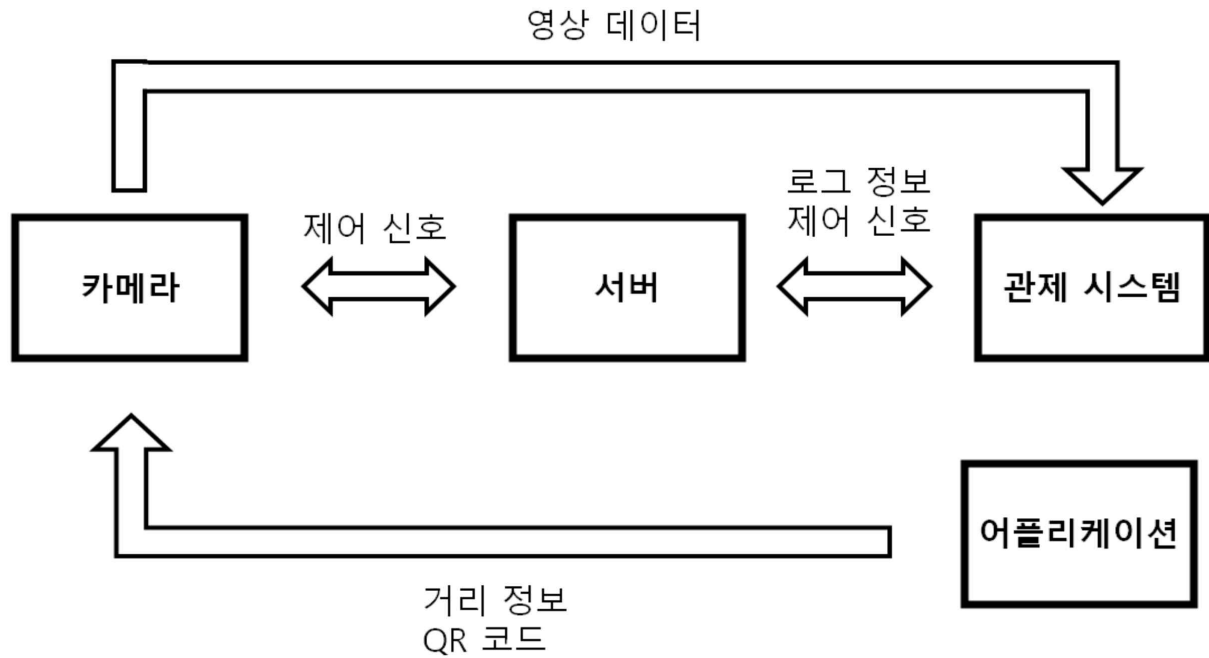
- SECAM은 사용자의 기기에 저장된 영상정보까지 보호하지 않는다.

#### 2.2 개발 목표

기존 시장에 있는 ‘홈 캠’의 기능을 포함하면서 SECAM의 아키텍처를 추가한 홈 캠을 개발 하는 것이다. SECAM 아키텍처 홈 캠이 제공하는 기능은 영상 촬영, 영상 정보 실시간 스트리밍, 영상 저장, 서버, 관제시스템 및 APP이다.



### 3. SECAM 소프트웨어 개념도



### 4. SECAM 개발 환경

개발 환경, 언어 및 프레임 워크는 다음과 같다.

개발 환경

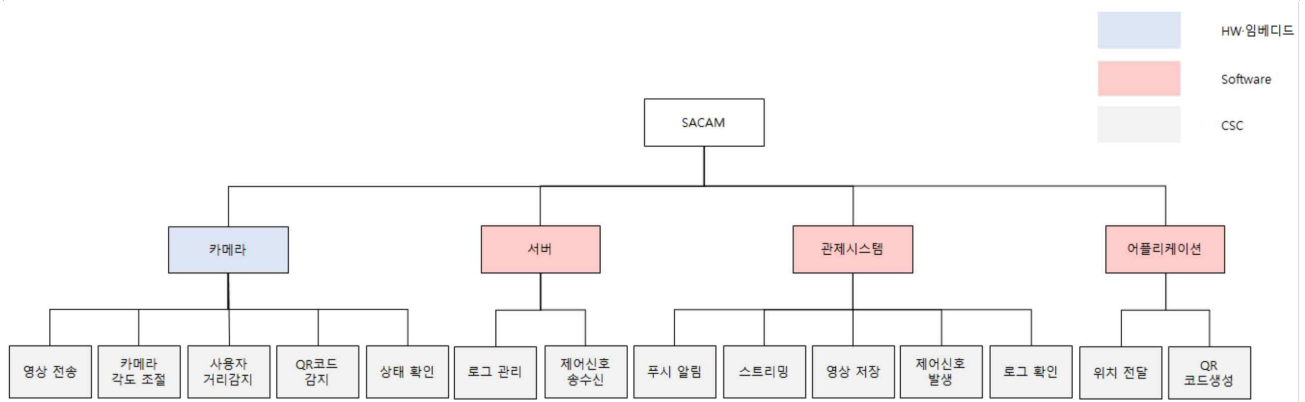
임베디드	관제 시스템, APP	서버
Arduino IDE	Visual Studio Code	Raspbian OS

개발 언어 및 프레임 워크

임베디드	서버, 관제 시스템, APP					
C Language	HTML	CSS	Java Script	node.JS	APP Inventer	Mongo DB

\* 두꺼운 글씨는 프레임 워크를 의미한다.

### 5. 소프트웨어 구성요소 분석



Item(소프트웨어항목) : SECAM (해킹에 의한 사생활 유출 방지 홈 캠)

CSCI(소프트웨어 형상항목) : 카메라(하드웨어), 서버, 관제시스템, 어플리케이션

CSC(소프트웨어구성품)

카메라 : 영상 촬영, 영상 전송, 카메라 각도 조절

서버 : 감시 신호 송신, 영상 전송, 라우팅, 영상 저장, 로그 저장

관제시스템 및 어플리케이션(APP)

공통요소 : 푸시 알림, 스트리밍, 저장·제어 신호 송신, 로그확인

어플리케이션(APP) : 위치 전달

## CSU(단위 소프트웨어)

### 카메라

영상전송 : WiFi 통신, 웹 서버 이용 스트리밍

각도 조절 : 서브 모터 PWM 제어, 제어 신호 수신

거리감지 : 블루투스 기능을 이용한 무선 통신 기능

QR인식 : QR 감지, QR 해석, EEPROM 데이터 저장

상태확인 : 동작 단계별 색상 출력

### 서버

제어신호 송수신 : 관제 시스템으로부터 수신 받은 제어 신호 카메라에 송신

로그 관리 : MongoDB를 이용한 로그 저장 및 로그 전송

### 관제시스템

스트리밍 : 카메라 웹서버 URL로 접속해 영상 프레임 송출

영상 저장 : Socket과 mediaRecorder를 이용한 데이터 전송

제어 신호 발생 : Socket을 이용한 데이터 전송

푸시 알림 확인 : 웹 푸시 API를 이용하여 유저에게 푸시 권한 요청 및 푸시 객체 생성 기능

로그 확인 : MongoDB를 이용한 로그 전송 및 출력

### 어플리케이션(APP)

위치 전달 : 블루투스를 이용한 무선 통신 기능

## 6. 검증 환경 및 검증 결과 기술

### 6.1 검증 환경

임베디드의 경우 Arduino IDE를 사용하여 개발 후, 테스트 보드인 Esp32 cam 을 사용하여 카메라 녹화 및 동작을 검증했다. 관제 시스템과 APP, 서버의 경우 프레임 워크인 NodeJS 사용했기 때문에 NodeJS의 Express를 사용하여 구현한 서버에서 영상 스트리밍 및 저장, 로그 기록 등의 기능들을 검증했다.

임베디드	관제 시스템, APP, 서버
Esp32 cam	Express

### 6.2 검증 결과 기술

## 카메라

검증 내용	검증 여부
영상 전송	0
카메라 각도 조절	0
사용자 거리 감지	0
QR 코드 감지	0
상태 확인	0

## 서버

검증 내용	검증 여부
제어 신호 송수신	0
로그 관리	0

## 관제 시스템

검증 내용	검증 여부
푸시 알림	0
스트리밍	0
로그 확인	0
제어 신호 발생	0
영상 저장	0

## 어플리케이션

검증 내용	검증 여부
QR 코드 생성	0
사용자 위치 전달	0

## 7. 통합 시험 결과 기술

각 CSC별로 검증한 결과들을 합하여 SECAM 이 운용되는 순서를 따라 통합 시험을 진행했다.

### 전원 인가

평상 시, 서버는 사용자의 위치 정보를 받음. 사용자가 특정 거리 내에 있는 경우, 센서에 움직임이 감지되어도 카메라에 전원이 인가되지 않는다. 반대의 경우 센서에 움직임이 감지되면 카메라에 전원이 인가됨과 동시에 카메라는 영상 녹화를 시작하고, 녹화되는 영상을 실시간으로 서버에 전송한다.

서버에서는 첫 번째 영상 데이터 패킷이 들어오는 순간, 감지 신호를 송신하고 로그를 기록한다. 또한 사용자에게 푸시 알림을 보내 이 사실을 알린다.

이 과정을 실행하였을 때 사용자 위치 전달, 영상 전송, 감지 신호 송신, 로그 저장, 푸시 알림 기능이 사용, 모두 검증 완료.

### 클라이언트 접속

클라이언트에 접속하면 클라이언트에 연결되었다는 로그 기록. 서버는 클라이언트로 영상 데이터를 전송, 영상 데이터를 받은 클라이언트는 해당 영상 데이터 패킷들을 관제 시스템에 스트리밍.

이 과정을 실행하였을 때 로그 저장, 영상 전송, 스트리밍 기능을 사용, 모두 검증 완료.

### 영상 저장 확인

영상 저장 버튼을 클릭하면 서버로 저장 신호가 송신. 저장 신호를 수신 받은 서버는 로그를 기록하고, 영상을 저장.

이 과정을 실행하였을 때 저장 신호 송신, 로그 저장, 영상 저장 기능을 사용, 모두 검증 완료.

### 카메라 제어 확인

사용자가 카메라의 촬영 각도를 변경 → 제어 버튼 클릭 → 서버로 제어 신호가 송신. 제어 신호를 수신 받은 서버는 해당 신호에 따라 카메라 각도 조절을 수행.

이 과정을 실행하였을 때 제어 신호 송신, 카메라 각도 조절 기능을 사용, 모두 검증 완료.

### 로그 기능 작동

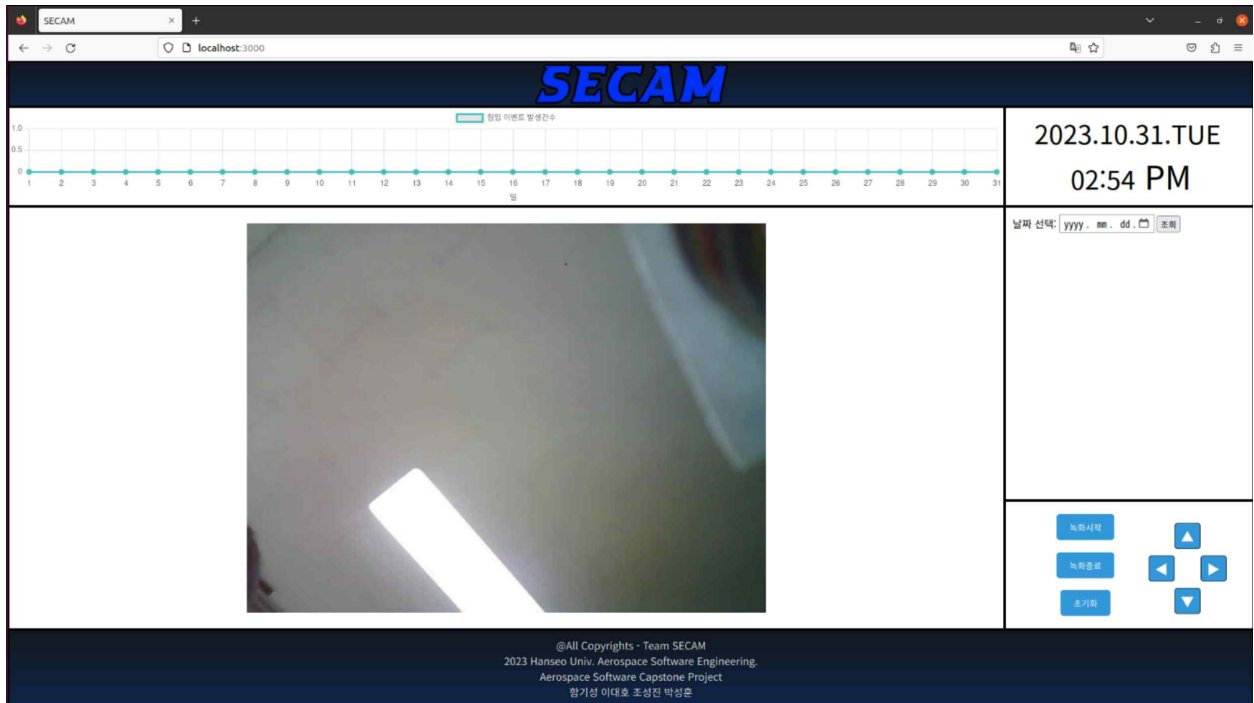
사용자가 이전에 기록된 로그를 확인하고 싶어 원하는 날짜를 입력하고 로그 조회 버튼을 클릭하면, 서버에서 데이터베이스에 접속하여 해당 날짜의 로그를 불러와 관제 시스템에 출력한다.

이 과정에서 로그 확인 기능이 사용되었으며 검증이 완료되었다.

## 결론

## 개발 결과

### 1. 관제 시스템 UI



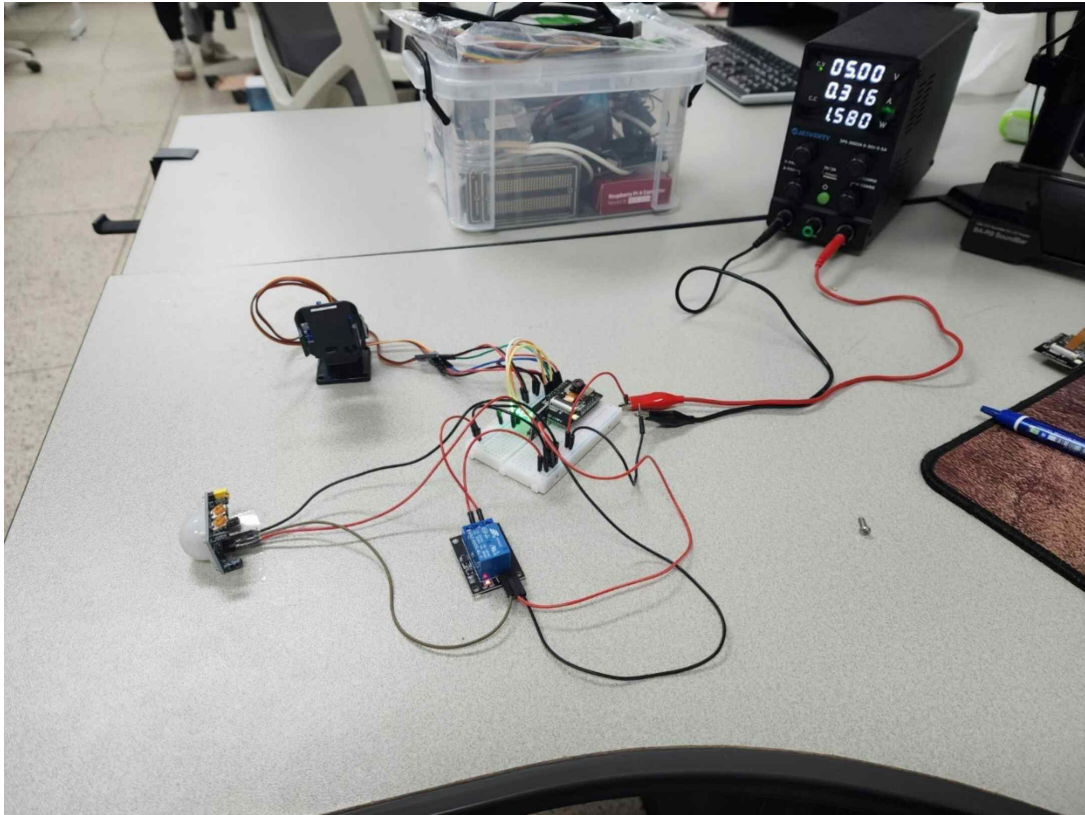
위에서 언급한 관제 시스템과 서버의 검증과정을 통해 사진과 같은 결과물을 완성했다. 각 버튼을 누르면 socket 통신을 통해 기존 설계 단계에서 구성한 동작들을 정상적으로 이행한다. 기존 로그 저장시스템에서 MongoDB를 활용한 로그 관리를 통해 사용자가 기존 로그 조회 방식보다 편한 조회를 할 수 있게 됐고, 해당 달에 발생한 침입 건수를 상단의 그래프를 통해 몇 번 발생했는지 알 수 있게 구현했다.

### 2. 안드로이드 어플리케이션



기존 웹앱이 아닌 사용자의 단말기의 자원들을 사용하기 위해 네이티브 앱으로 변경하였다. 어플리케이션 또한 검증 및 통합 구현을 통해 QR 코드 생성과 사용자와 홈 캠 간의 거리를 홈 캠 전송을 하여 홈 캠이 촬영 진행의 여부를 판단할 수 있게 해준다. 위 사진은 구현 결과물이다.

### 3. 카메라



홈 캠의 경우 이번 프로젝트가 소프트웨어 중심의 프로젝트이기 때문에 HW에 할애할 수 있는 시간이 부족하여 테스트 보드 형식으로 제작하게 되었다. 테스트 보드 형식이지만, SECAM 팀의 검증 과정을 통해 구현했으며, 구현된 테스트 보드는 홈 캠의 역할을 모두 수행하는 것을 확인했다. 사진은 SECAM 팀의 홈 캠 HW이다.

#### 기대 효과

##### 1. 개인 정보 보호 향상

개인 정보 보호 강화 가능 -> 웹 캠의 보안 취약점을 통한 외부 공격으로부터 사용자 보호 -> 개인 정보 유출로 인한 피해 감소

##### 2. 온라인 개인정보 및 금융 거래의 안정성 향상

개인 사용자에게 안정성 제공 -> 해킹에 의한 사용자 화면 유출 방지 -> 온라인 개인 정보 및 금융 거래 보안 강화 기능

##### 3. 사이버 스파이 및 침입자 탐지

얼굴인식, 움직임 및 환경 모니터링 -> 악의적 침입자 or 사이버 스파이 탐지 가능

##### 4. 비대면 보안 강화

카메라 해킹 방지 -> 비대면 통신 및 회의 보안 강화 -> 개인 및 비즈니스 사용자의 비대면 통신 이용률 증가 가능

##### 5. 사회적 영향

개인 및 조직의 보안 인식 증가 가능 -> 개인 사용자의 사이버 보안에 대한 주의와 적절한 보호조치 가능 -> 기업, 정부 기관의 보안 인프라 강화.

##### 6. 사이버 범죄 예방

웹 캠의 보안 강화 -> 사이버 범죄의 일종인 해킹 저지 -> 사회적 안전성 향상

## SECAM 프로젝트를 마치며

이번 SECAM 프로젝트를 마무리 하면서 팀원들이 프로젝트를 하면서 느낀 점을 마지막으로 보고서를 마치겠다.

### 팀장 함기성

이번 프로젝트를 진행하며 팀 프로젝트 운영과 소프트웨어 개발 주기에 맞춘 개발을 진행해 의미있는 경험을 한 것 같다. 또한 일정관리를 하며 프로젝트를 진행해 취업 이후 프로젝트에 참여하게 되면 어떻게 계획해야 할지 알 수 있었다. 혼자 코드를 작성하고 프로젝트를 진행할 때와는 다르게 서로 진행사항을 점검하고 일정을 조율하는 것, 문서화의 중요성을 깨달을 수 있었다.

### 어플리케이션 박성훈

프로젝트를 진행하면서 SSID와 비밀번호를 입력하여 QR코드를 생성하고 블루투스를 통한 자동 연결 기능을 개발한 경험은 매우 유익하고 의미있었습니다. 사용자 편의성을 강조하고자 한 점에서, QR코드를 통한 무선 네트워크 설정은 기존의 수동 입력 방식보다 훨씬 간편하면서도 보안 측면에서도 높은 수준의 안전성을 제공합니다. 사용자들이 매우 간편하게 카메라를 설정하고 사용할 수 있는 이 기능은 사용자 경험을 현저히 향상시켰습니다. 블루투스를 자동으로 연결하는 부분은 기술적으로도 흥미로웠습니다. 이로써 사용자는 번거로운 연결 프로세스를 피하고, 어플과 홈캠 간의 신속하고 원활한 소통을 경험할 수 있게 되었습니다. 프로젝트를 통해 많은 성과를 이뤘지만, 아쉽게도 iOS 플랫폼에서의 구동이 제약되는 부분이 있었습니다. 현재까지의 구현은 안드로이드 플랫폼에 중점을 두고 진행되었으며, iOS 환경에서의 완벽한 호환성은 아직 달성되지 못한 상태입니다. iOS에서의 구동이 어려운 이유에 대해서는 특정 기술적인 제약, 또는 리소스 부족 등 여러 가지 이유가 있습니다. 향후 프로젝트의 발전을 고려할 때는 iOS 플랫폼에 대한 지원을 강화하는 것이 중요할 것으로 생각합니다.

### 서버 & 관제시스템 조성진

이번 프로젝트를 진행하며 전체적인 개발 프로세스를 경험해볼 수 있었다. 또한 개인이 아닌 팀 단위로 프로젝트를 진행하면서 팀워크, 일정 관리의 중요성을 깨달았다. 그리고 여러 번의 수정을 거치면서 코딩도 중요하지만, 그것들을 꼼꼼히 문서화하는 것도 중요하다는 것을 알게 되었다.

### 서버 & 관제시스템 이대호

기존 웹 개발 경험이 있지만, 사용자 UI를 담당하는 Client 쪽은 다루지 않고 주로 Back-end 단 개발을 진행했었다. 그 당시는 Django, Springboot만 사용해봤다. 이번 졸업 프로젝트를 진행하면서 JS 기반의 NodeJS를 사용했다. 아무래도 배움이라는 명목하게 진행했지만, 졸업 프로젝트에선 '배움'이라는 것보다 '결과'라는게 더 중요한 걸 깨달았다. 처음 해보는 영역들에서 완성도가 떨어지고, 시간도 많이 소요가 되었기 때문이다. 하지만 No SQL을 다뤄보고 비동기적 프로그래밍을 사용해보았다는 것에 의의를 두고 있다. 이번 처음으로 HW와 SW를 융합한 코드를 직접 작성해보았다. 만족도 보단 아쉬움이 많이 남은 결과물이다.

배움이라는 측면에선 SW설계부터 구현 서류화 까지 전체적인 프로세스를 지키면서 개발하는 것이 중요하며 첫 요구분석 단계와 설계 단계에 시간을 많이 투자해야한다는 걸 느꼈다.

자기자 제일 자신이 있는 것으로 스킬로 구현해야 하며, 그 과정에서 팀원들과의 소통 아이디어를 실체화 하는 과정이 중요하다는 것을 깨달았다.

이상 SECAM의 항공소프트웨어 프로젝트 최종 보고서를 마무리 한다.

- 끝 -