

SW개발/HW제작 설계서

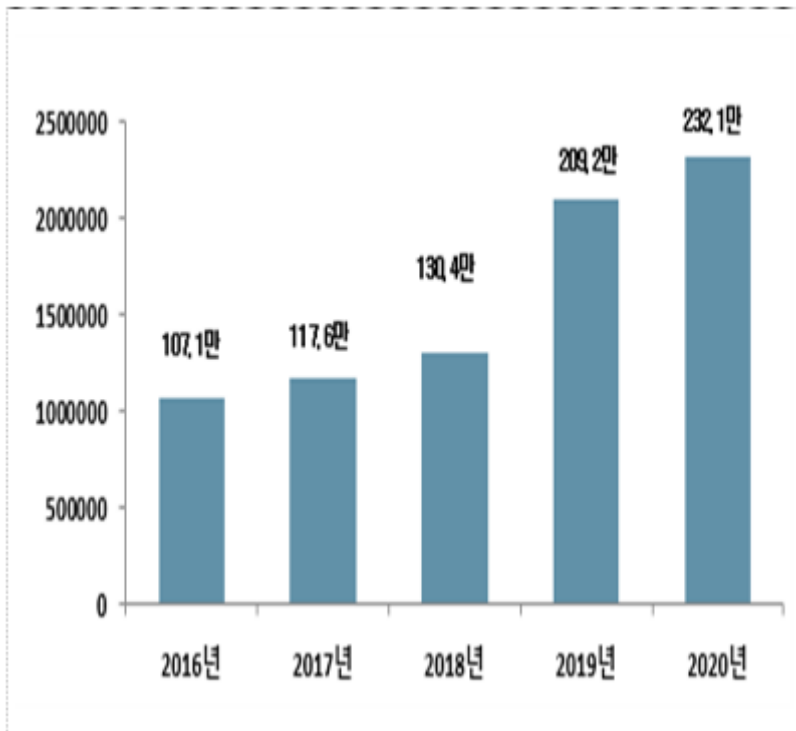
프로젝트 명 : 반려견 목줄 탐지 서비스

2023. 11. 1

| 시장/기술 동향 분석

| 반려견 소유 인구의 현황과 성장 추이 분석 |

출처 : 농림 축산 검역본부, 「반려동물 보호·복지 실태조사 결과」, 2020년



반려견 등록 누계 현황(마리)

- 2020년 신규 등록 반려견은 23만 5,637마리, 등록된 총 반려견의 수는 232만 1,701마리로 집계됨 (전년 대비 11% 증가)
- 비반려인 중 22%가 여력이 된다면 반려동물을 키울 예정이라고 함 (출처 : 한국리서치 여론속의 여론)

→ 반려견 소유 인구 증가 추세

| 시장/기술 동향 분석

| 개 물림 사고 발생 원인 및 통계 자료 분석 |

개 물림 사고 발생 원인 |

출처 : 시사 매거진, 「반려동물 인구 1,500만 시대, 끊이지 않는 개 물림 사고원인과 대책은?」

피해의 원인, 목줄과 입마개 부재에 있어

다른 반려견의 목숨은 물론 사람의 목숨까지도 위협하는 개 물림 사고가 잇따르면서 반려동물 관리에 대한 규제와 책임에 대한 논란이 계속되고 있다.

대다수의 개 물림 사고의 1차적 원인은
목줄 미착용 및 문단속 미비 등
보호자의 관리 부실로 인해 발생

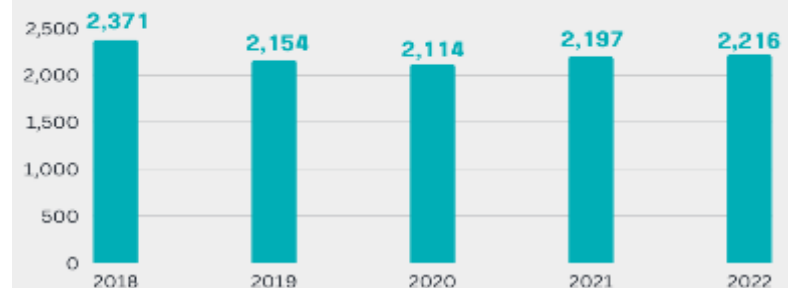
개 물림 사고 발생 원인

| 개 물림 사고 통계 자료

- 매년 2천명 이상, 하루 평균 6명 이상
이 개에게 물리는 사고가 발생
- 해당 자료는 119 출동 횟수이므로
실제로는 이보다 많을 것으로 예상됨

출처 : 단비뉴스 「목줄 안 한 개, 신고하려면 주소까지 알아야」

119구급대 개물림 환자 이송 현황



개 물림 사고 통계

| 시장/기술 동향 분석

| 시장 진입 분석 |

| 반려인 · 비반려인의 인식 차이

'<반려견 소유자 준수사항> 이 잘 지켜지고 있는가' 문항에서

반려견 양육자의 경우 83.1%가 그렇다고 했으나,

비양육자의 경우 33.6%만 그렇다고 답변함

→ 반려인과 비반려인 사이에 큰 인식 차이 有

(출처 : 농림축산식품부 '2022 동물보호에 대한 국민의식조사')

| 현재 시장에서 판매되고 있는 반려견 목줄 탐지 관련 서비스 無

- 기존의 반려견 관련 서비스는 반려견의 건강, 펫캠, 모빌리티 등 반려인만을 위한 서비스가 대부분
- 반려인과 비반려인 모두의 안전과 재산을 보호하기 위한 AI 기술이 필요함
- 반려견 산업의 새로운 사업 영역으로 성장할 것이라 기대

| 인터뷰결과서

| 전화 인터뷰 결과서 |

Keep-In Mind Message

목줄 미착용 신고는 다산콜 센터, 경찰,
지자체별 담당 부서 등에서 접수

반려견 목줄 미착용 신고가 접수되어도
상시 단속이 불가능함
담당자의 현장 출동 중 해결되는
경우가 대부분

반려견 목줄 미착용 신고 관련 통계
자료는 별도로 없음

본 과제 추진 반영 시사점

반려견 목줄 미착용 신고 접수 부서가 지자체별로
달라 혼란 유발. 복잡한 신고 절차

목줄 미착용 반려견이 공공장소에서
발견되어도, 즉각적인 제도 조치가 불가능함

반려견 목줄 미착용 이슈에 대한 사회적 관심에도
불구하고 신고 건수, 현황에 대한
통합적 통계 자료의 부재

| 요구사항 정의서

| 일반 (WEB) |

기능적 요구사항

구분	요구사항 ID	요구사항 명	기능 ID	기능명	상세설명
일반(WEB)	FADM01	실시간 관리감독	FADM01_MNT	실시간 모니터링	상황을 실시간 모니터링 할 수 있어야함
			FADMIN_ALRM	실시간 알림	목줄 미착용 반려견이 탐지되는 상황을 알림을 통해 실시간으로 인지할 수 있어야함
			FADM01_INFO	실시간 정보 확인	목줄 미착용 반려견과 이의 상황에 대한 정보를 실시간으로 확인할 수 있어야함
			FADM01_RSP	실시간 대응 (접수&삭제)	실시간으로 전송되는 정보 확인 후 그에 따른 대응을 할 수 있어야 하며, 대응 방식을 저장할 수 있어야함
	FADM02	하드웨어 설정	FADM02_SET	설정	음량, 알림 반복,주기 등 하드웨어 설정을 컨트롤 할 수 있어야함
			FADM02_MENT	멘트 설정	적발 시, 적발 외시 송출되는 안내멘트를 설정할 수 있어야함
	FADM03	통계	FADM03_STAT	통계 확인	목줄 미착용 반려견의 데이터를 통계로 확인할 수 있어야함
	FAPI01	데이터 관리	FAPI01_UPD	데이터 업데이트	목줄 미착용 반려견 데이터를 실시간으로 API 데이터로 업데이트 할 수 있어야함
			FAPI01_VW	데이터 조회	목줄 미착용 반려견 데이터를 실시간으로 API 데이터로 조회 할 수 있어야함

비기능적 요구사항

구분	요구사항 ID	요구사항 명	기능 ID	기능명	상세설명
일반(WEB)	NFADMI01	편의성	NFADM01_CONV	사용 편의성	웹 페이지가 사용자가 이용하기 편한 상태로 제공되어야함
	NFAPI01	범용성	NFAPI01_UNIV	사용 범용성	구축된 데이터들은 다른 용도 또는 도구로 쉽게 활용될 수 있어야함
	NFWEB	속도	NFAPI01_SPD	사용 속도	많은 데이터에도 빠른 속도로 서비스할 수 있어야함
		안정성	NFAPI01_STBL	사용 안정성	서비스는 어떤 상황에도 항상 안정적이어야함

| 요구사항 정의서

| 응용 소프트웨어 (AI) |

기능적 요구사항

구분	요구사항 ID	요구사항 명	기능 ID	기능명	상세설명
응용 소프트웨어 (AI)	FDETECT01	객체 탐지	FDETECT01_DOG	반려견 탐지	입력되는 이미지로부터 반려견이 존재하는 영역의 경계 상자 값을 제공해야함
	FCLS01	이미지 분류	FCLS01_LSH	목줄 탐지	반려견의 목줄 착용 여부를 분류해야함
	FTRCK01	객체 추적	FTRCK01_DOG	반려견 추적	입력되는 일련의 이미지로부터 동일한 반려견은 같은 객체로 인식해야함
	FEXP01	예외 처리	FEXP01_THETA	임계값 설정	반려견 탐지 AI 모델을 사용하여 반려견 존재 영역의 정확성(accuracy)에 대한 임계값(threshold)을 설정해야함

비기능적 요구사항

구분	요구사항 ID	요구사항 명	기능 ID	기능명	상세설명
응용 소프트웨어 (AI)	NFSPD01	속도	NFSPD01_MDL	모델 경량화	AI 모델의 크기를 줄이고, 연산량을 줄여서 모델의 성능을 유지하면서 속도와 자원 사용을 최적화해야함
	NFACR01	정확성	NFACR01_MDL	모델 정확성	결과에 대한 충분한 정확도를 보장해야함
	NFRLB01	신뢰성	NFRLB01_MDL	모델 신뢰성	AI모델 학습 과정에서 사용되는 데이터의 편향성을 최소화하고 충분한 양과 정확한 라벨링 과정이 요구됨
	NFEXP01	확장성	NFEXP01_MDL	모델 확장성	AI모델의 파이프라인을 클래스로 설계하여 기능별로 모듈을 쉽게 교체하거나 업데이트할 수 있도록 함.

| 요구사항 정의서

| 응용 하드웨어(IoT) |

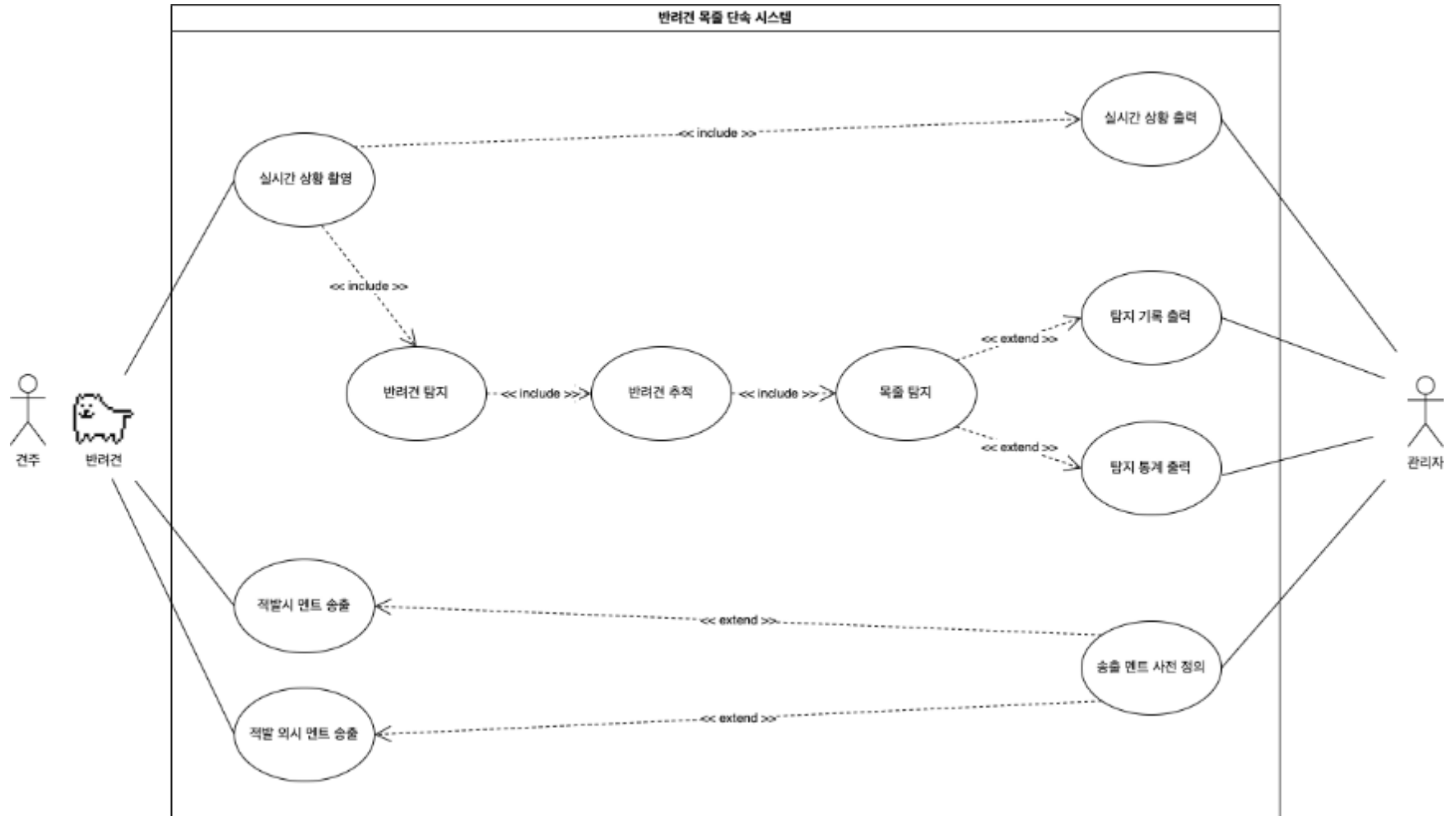
기능적 요구사항

구분	요구사항 ID	요구사항 명	기능 ID	기능명	상세설명
응용 하드웨어 (IoT)	FCAM01	카메라 작동	FCAM01_RTM	실시간 캠 작동	실시간 단속을 하기 위해 카메라는 실시간으로 작동해야함
	FALRM01	소리 송출	FALRM01_CAT	적발 시 소리 송출	목줄 미착용 견주에게 반려견 목줄 착용을 경고하기 위해 소리 송출이 필요함
			FALRM01_NCAT	적발 외 시 소리 송출	적발 외 시 공공장소에서 안내 사항을 송출할 수 있어야함
	FCONT01	외부 연결	FCONT01_WL	무선 연결	모든 하드웨어는 무선으로 상호 통신할 수 있어야함

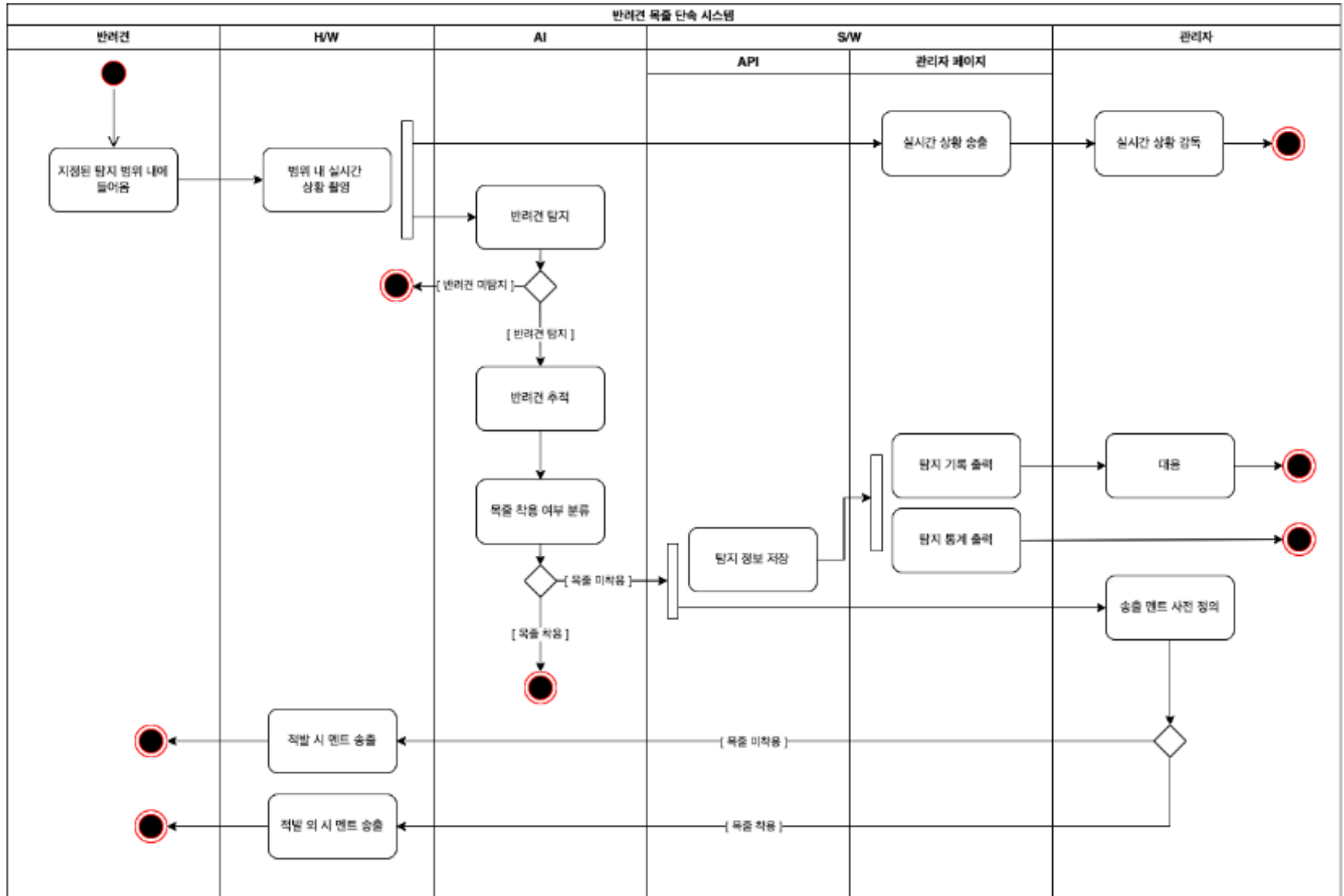
비기능적 요구사항

구분	요구사항 ID	요구사항 명	기능 ID	기능명	상세설명
응용 하드웨어 (IoT)	NFSTAB01	안정성	NFSTAB01_HW	장치 안정성	라즈베리파이 환경 내에서도 충분히 모델을 구현할 수 있는 시스템을 구성하여 하드웨어에 무리가 가지 않도록 함.
	NFPORT01	휴대성	NFPORT01_SIZE	장치 크기	설치의 편리성을 위해 장치의 크기는 작아야함.
	NFMNT01	유지성	NFMNT01_BAT	배터리	라즈베리파이의 구동을 위해 전원이 지속적으로 공급되어야함

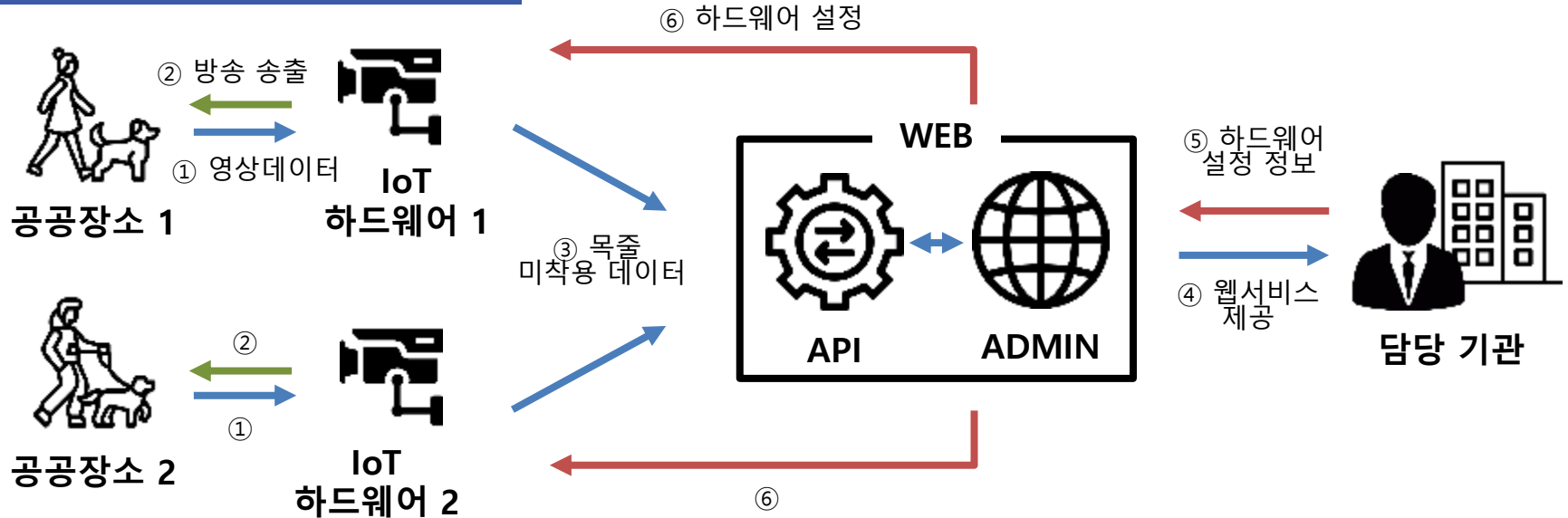
| 유즈케이스



반려견 목줄 탄속 시스템

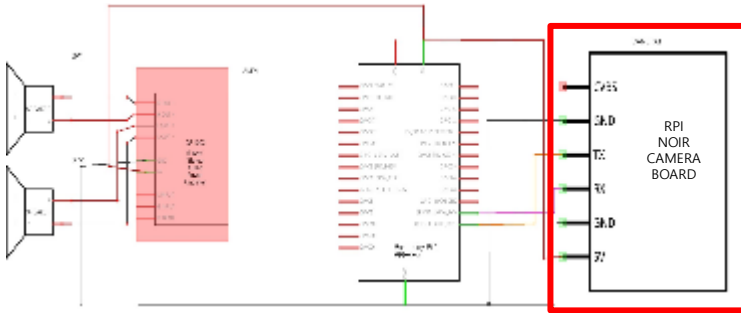


| 서비스 흐름도

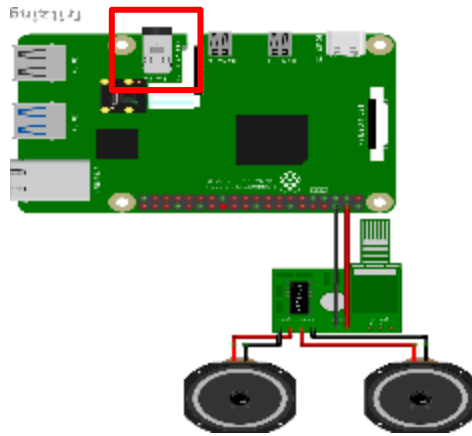


구분	상세 설명
관리자용 웹페이지	① 각각의 IoT 하드웨어는 장치 설치 장소의 실시간 영상 데이터를 입력 받음. ③ IoT 하드웨어 내부의 AI 추론 과정을 통해 목줄 미착용 반려견의 데이터를 서버로 전송. ④ 목줄 미착용 반려견 탐지 장소, 시간, 이미지 정보를 웹페이지를 통해 지자체별 신고 접수 담당 기관에 제공. ⑤ IoT 하드웨어별 음량, 안내멘트 내용 등 하드웨어 설정 정보를 웹사이트에 입력.
실시간 방송 송출	① 각각의 IoT 하드웨어는 장치 설치 장소의 실시간 영상 데이터를 입력. ②-1 목줄 미착용 반려견이 탐지되었을 때, 반려견 목줄 착용을 요구하는 방송 멘트 송출. ②-2 평상시, ⑤에서 설정한 평상시 방송 멘트 송출.

| 하드웨어/센서 구성도



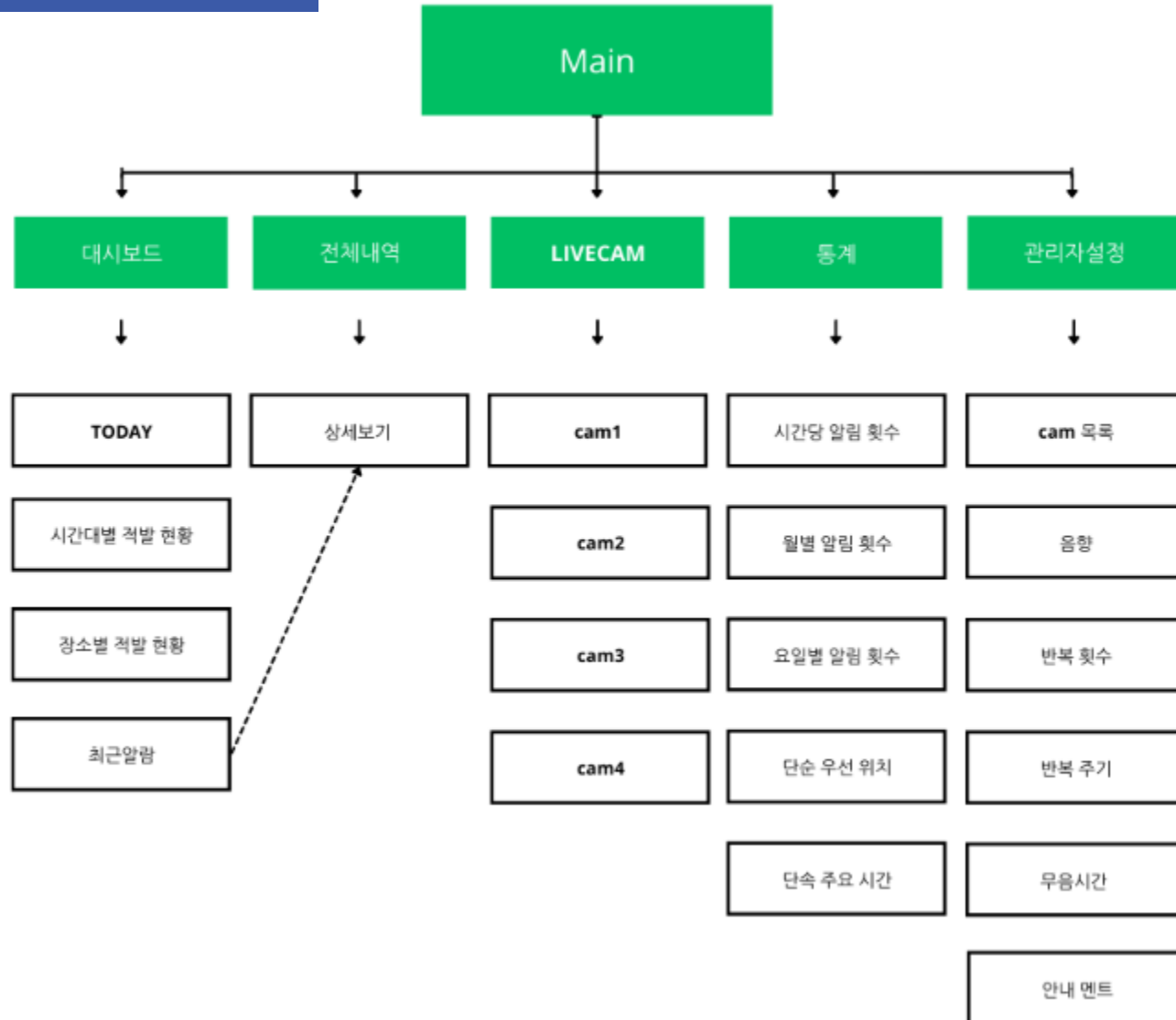
하드웨어 회로도



하드웨어 구성도

센서	연결 핀	설명
RPI NOIR CAMERA BOARD	CSI	라즈베리파이 CSI에 연결 (빨간색 박스)
PAM8403 (오디오 모듈)	GND	라즈베리파이의 GND에 연결
	VCC	라즈베리파이의 5V에 연결
	Output L+	스피커 1의 + 단자에 연결
	Output L-	스피커 1의 - 단자에 연결
	Output R+	스피커 2의 + 단자에 연결
	Output R-	스피커 2의 - 단자에 연결
스피커1	+	PAM8403의 Output L+에 연결
	-	PAM8403의 Output L-에 연결
스피커2	+	PAM8403의 Output R+에 연결
	-	PAM8403의 Output R-에 연결

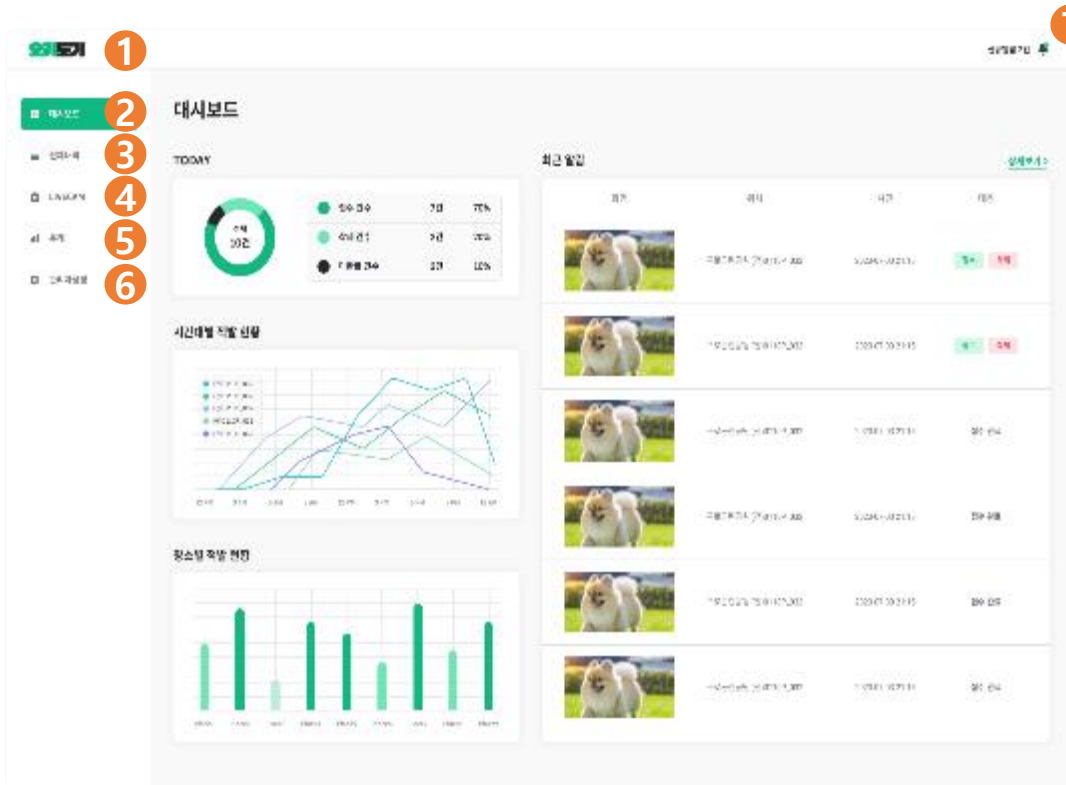
| 메뉴 구성도



| 화면 설계서 — UI/UX 설계

사이드바 & 상단바

기능: URL(페이지 이동)
기능 번호: URL0101 ~ URL0105

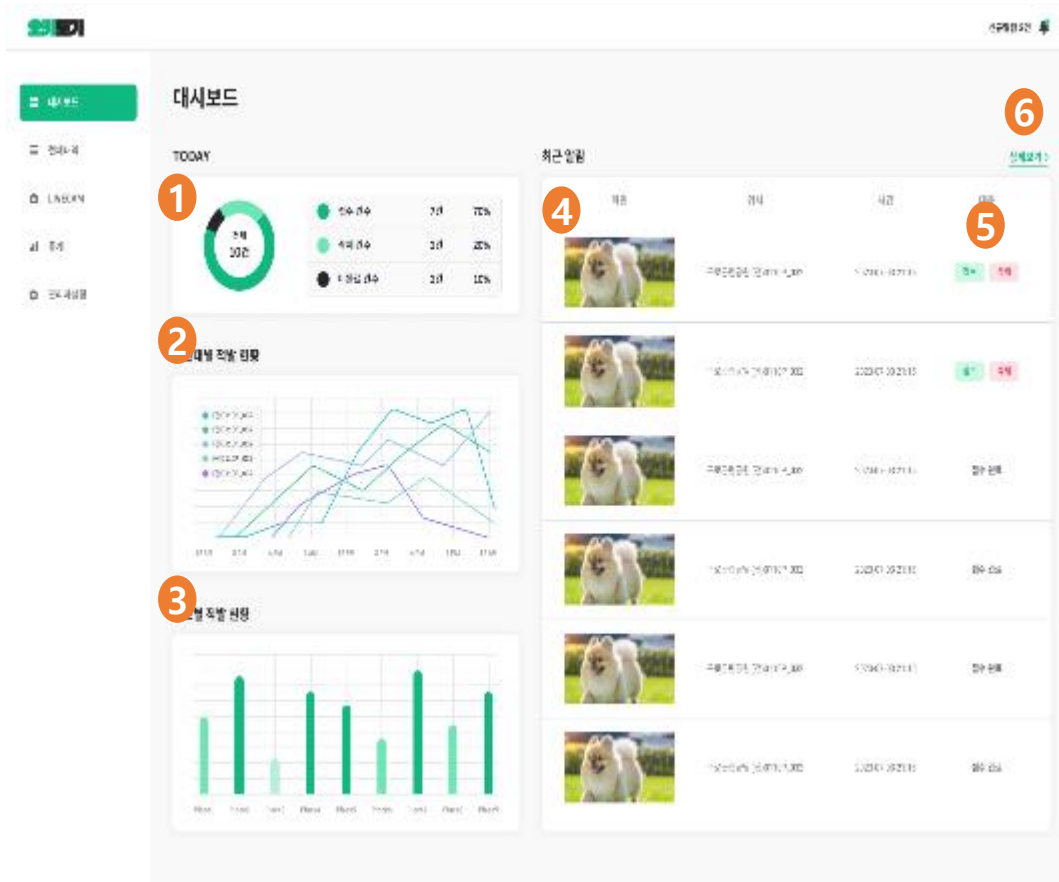


Description	
①	■ 대시보드 페이지로 이동
②	■ 대시보드 페이지로 이동
③	■ 전체내역 페이지로 이동
④	■ LIVECAM 페이지로 이동
⑤	■ 통계 페이지로 이동
⑥	■ 관리자설정 페이지로 이동
⑦	■ 신규 알람 발생 시 발생 건수 및 아이콘 변경.

| 화면 설계서 — UI/UX 설계

대시보드

기능: Declaration(신고), URL(페이지 이동)
기능번호: Decl0101 ~ Decl0103, Url-01-02

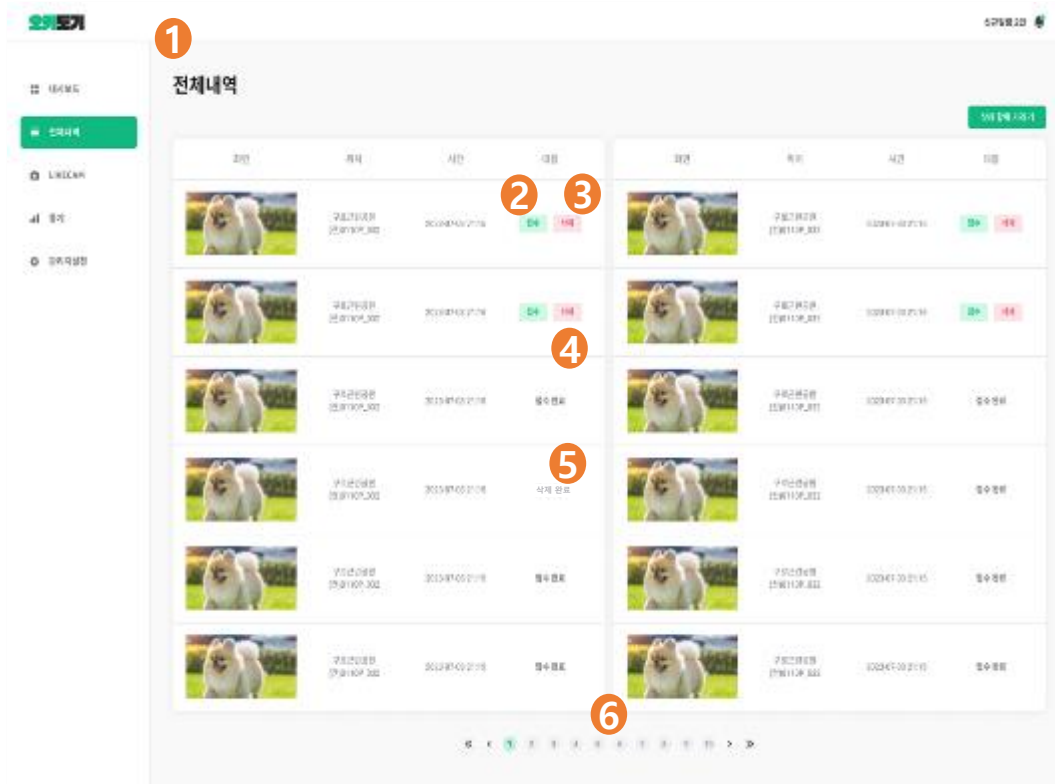


Description	
①	<ul style="list-style-type: none"> 금일 반려견 목줄 미착용 알림 현황 요약 및 시각화. 총 몇 건이 발생했는지, 접수, 삭제, 미완료 건수 및 비율 정보 제공.
②	<ul style="list-style-type: none"> 금일 시간대별 적발 현황을 시각화. 장소별로 다른 색상 부여로 구분.
③	<ul style="list-style-type: none"> 금일 장소 별 적발 현황 시각화.
④	<ul style="list-style-type: none"> 최근 6건에 대해 목줄 미착용 반려견 이미지, 탐지된 위치, 시간에 정보 제공
⑤	<ul style="list-style-type: none"> 대응과 삭제 버튼을 통해 담당자가 유연적으로 선택할 수 있음.(상세 기능은 다음 페이지에 기재)
⑥	<ul style="list-style-type: none"> 상세보기 버튼 클릭 시 전체 내역 페이지로 이동.

| 화면 설계서 - UI/UX 설계

전체내역

기능: Declaration(신고)
기능번호: Decl0101 ~ Decl0103



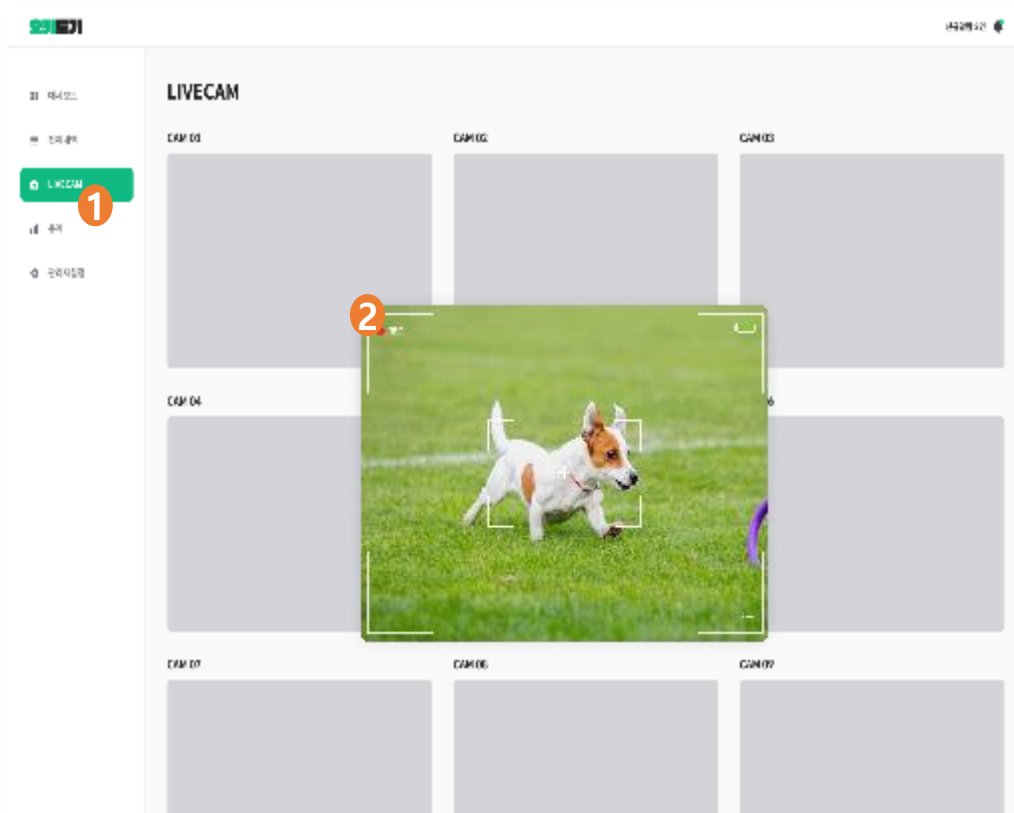
Description	
①	<ul style="list-style-type: none"> 전체 목줄 미착용 탐지 내역(감지된 사진, 위치, 시간)을 최신순으로 테이블로 제공
②	<ul style="list-style-type: none"> 미착용 내역 접수 버튼. 클릭 시 담당자를 선택, 취소할 수 있는 창 제공. 취소 시 해당 내용이 그대로 유지.
③	<ul style="list-style-type: none"> 미착용 내역 삭제 버튼 클릭 시 삭제를 하는 게 맞는지 되묻는 창 제공. 취소 시 해당 내용이 그대로 유지.
④	<ul style="list-style-type: none"> 2번에서 담당자를 선택하면, 내용이 '선택 완료'로 변경됨.
⑤	<ul style="list-style-type: none"> 3번에서 삭제하는 것이 다시 확인되면, '삭제 완료'로 변경됨.
⑥	<ul style="list-style-type: none"> 전체 목줄 미착용 내용 페이지 선택 기능.

| 화면 설계서 — UI/UX 설계

LIVECAM

기능: Camera (카메라)

기능번호: Cam0101 ~ Cam0103



Description

①

- IoT 하드웨어 장비가 설치된 장소의 실시간 영상을 보여줌.(CCTV)

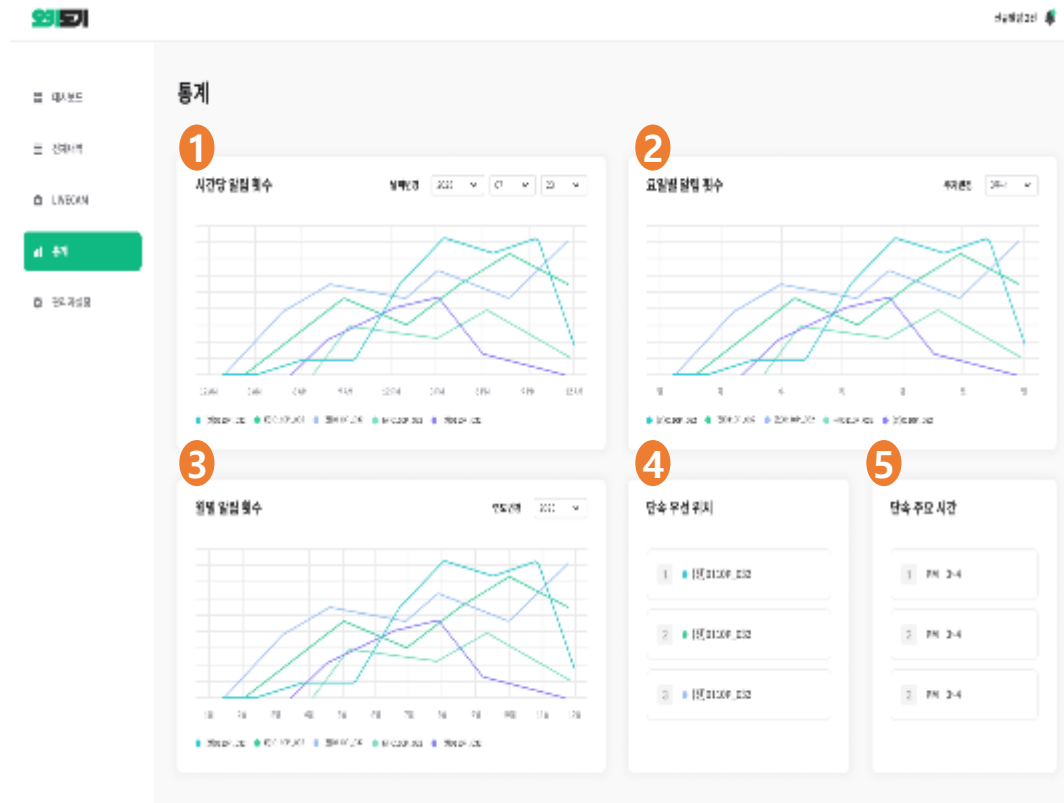
②

- 1에서 특정 카메라를 클릭 시 해당 화면이 새 창으로 확대되어 나타남

| 화면 설계서 — UI/UX 설계

통계

기능: Stochastic (통계)
기능번호: Str0101 ~ Str0103



Description	
①	<ul style="list-style-type: none"> 시간 당 알림 횟수를 장소별로 시각화. 날짜 변경 기능을 이용해 이전 데이터 확인 가능.
②	<ul style="list-style-type: none"> 일주일 간 알림 횟수를 장소별로 시각화. 주차 변경 기능을 통해 이전 데이터 확인 가능.
③	<ul style="list-style-type: none"> 월별 알림 횟수를 장소별로 시각화. 연도 변경 기능을 통해 이전 데이터 확인 가능.
④	<ul style="list-style-type: none"> 어제의 데이터를 기반으로 가장 많이 목줄 미착용 반려견이 접수된 장소 Top3 정보 제공.
⑤	<ul style="list-style-type: none"> 어제의 데이터를 기반으로 가장 많이 목줄 미착용 반려견이 접수된 시간대 Top3 정보 제공.

| 화면 설계서 — UI/UX 설계

관리자 설정

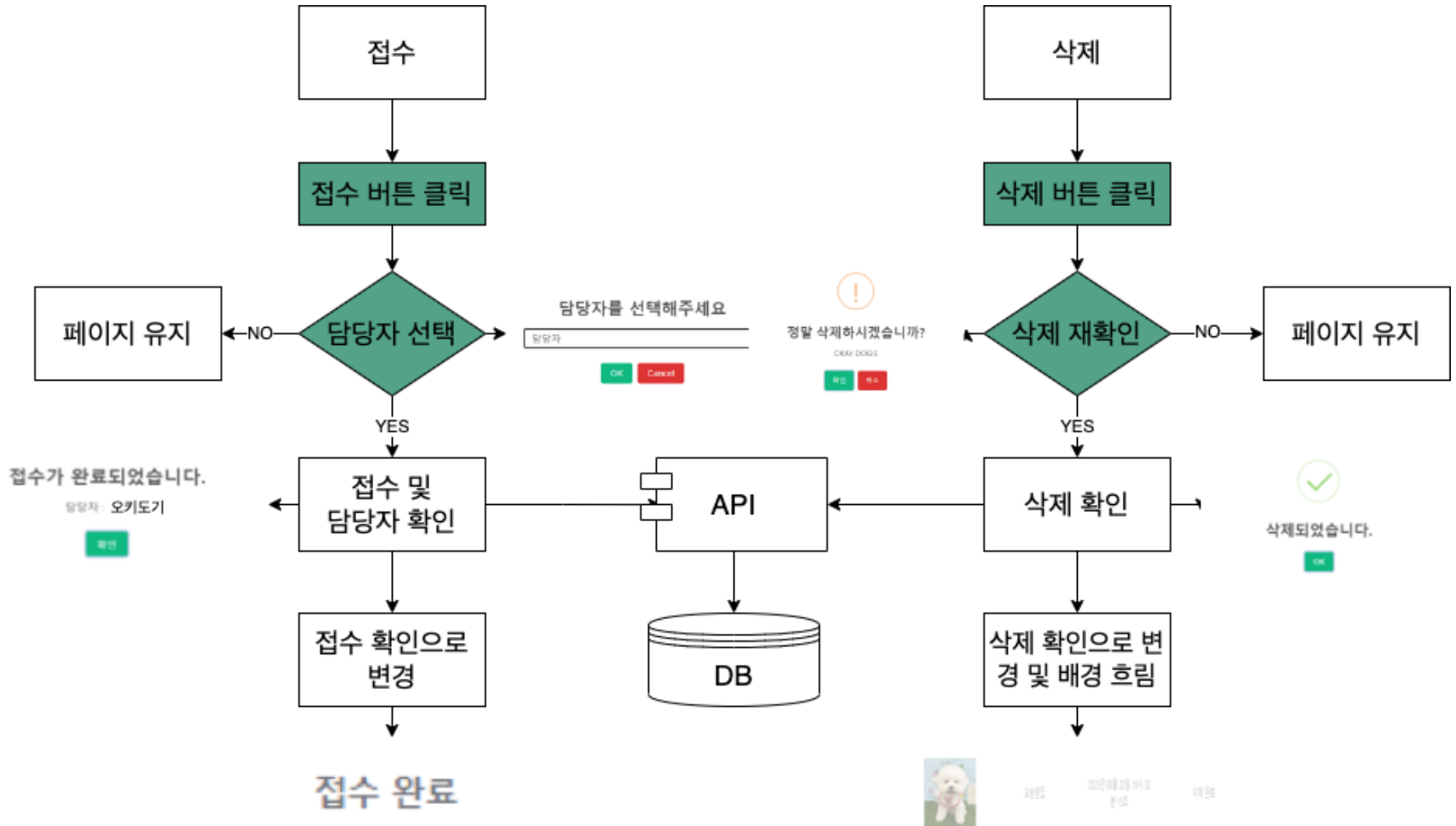
기능: Sound(소리), Camera(카메라)
기능번호: Snd0101 ~ Snd0105, Cam0103

The screenshot shows the '관리자설정' (Admin Settings) page for the 'CAM' section. The page includes a sidebar with navigation links like '대시보드', '전체관리', 'LIVECAM', and '설정'. The main content area is titled '관리자설정' and 'CAM'. It features a search bar (1), a volume slider (2), a selection of camera devices (3), a selection of a device (4), a selection of a device (5), and a '저장' (Save) button (6).

Description	
①	<ul style="list-style-type: none"> 설정을 원하는 카메라(IoT 하드웨어) 검색 설정을 원하는 카메라 선택. 전체 선택 시 관리 담당 카메라 전체 선택.
②	<ul style="list-style-type: none"> 선택된 장비의 음향 크기를 조절하는 기능.
③	<ul style="list-style-type: none"> 반려견 목줄 미착용 미탐지 시 (정상 시) 안내 방송의 반복 횟수와 반복 주기 설정.
④	<ul style="list-style-type: none"> 공공장소에 설치되므로 정상시의 안내 방송 및 목줄 미착용 반려견 탐지 시 송출되는 안내 방송 중단 시간 설정.
⑤	<ul style="list-style-type: none"> 반려견 목줄 미착용 미 탐지 시 (정상 시) 송출할 안내 멘트 입력창.
⑥	<ul style="list-style-type: none"> 설정 내용을 저장.

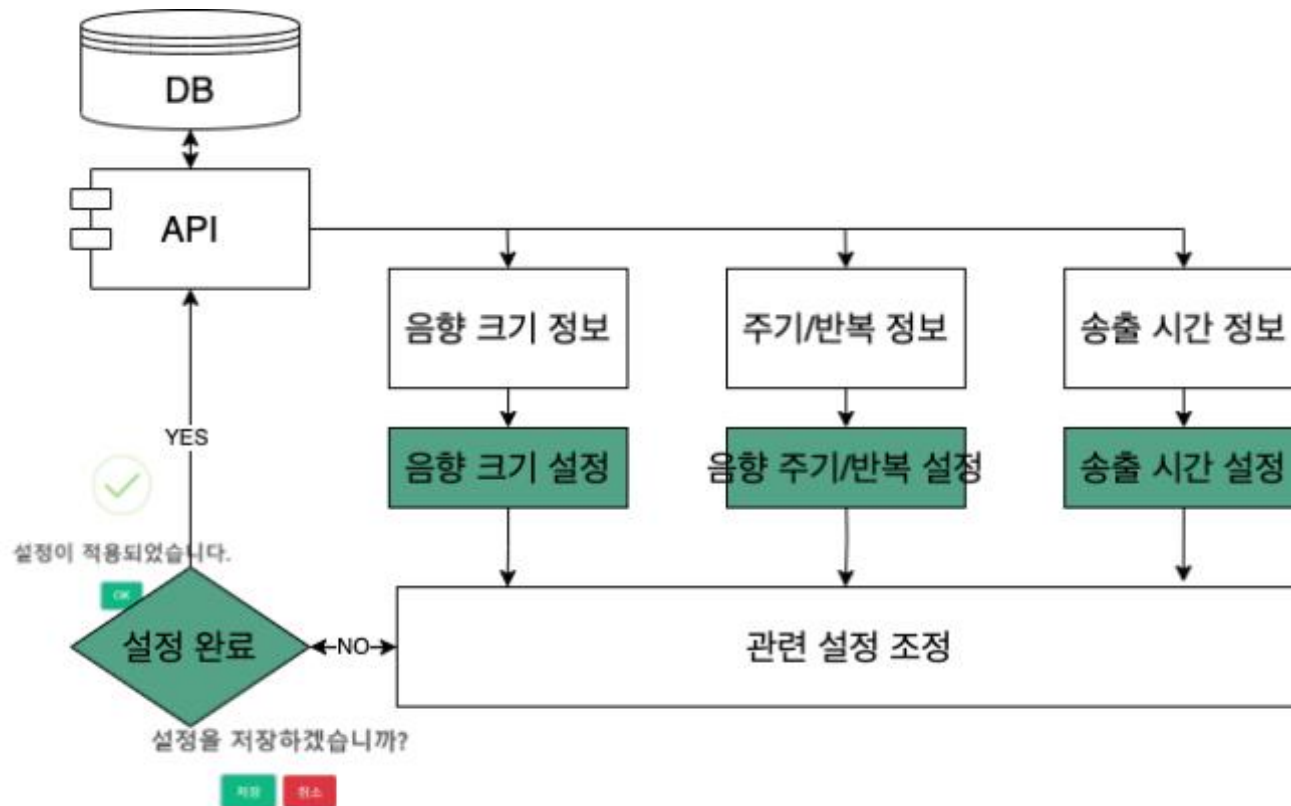
| 화면 설계서 — UI/UX 설계

분류	삭제/접수 버튼
기능	Declaration(신고)



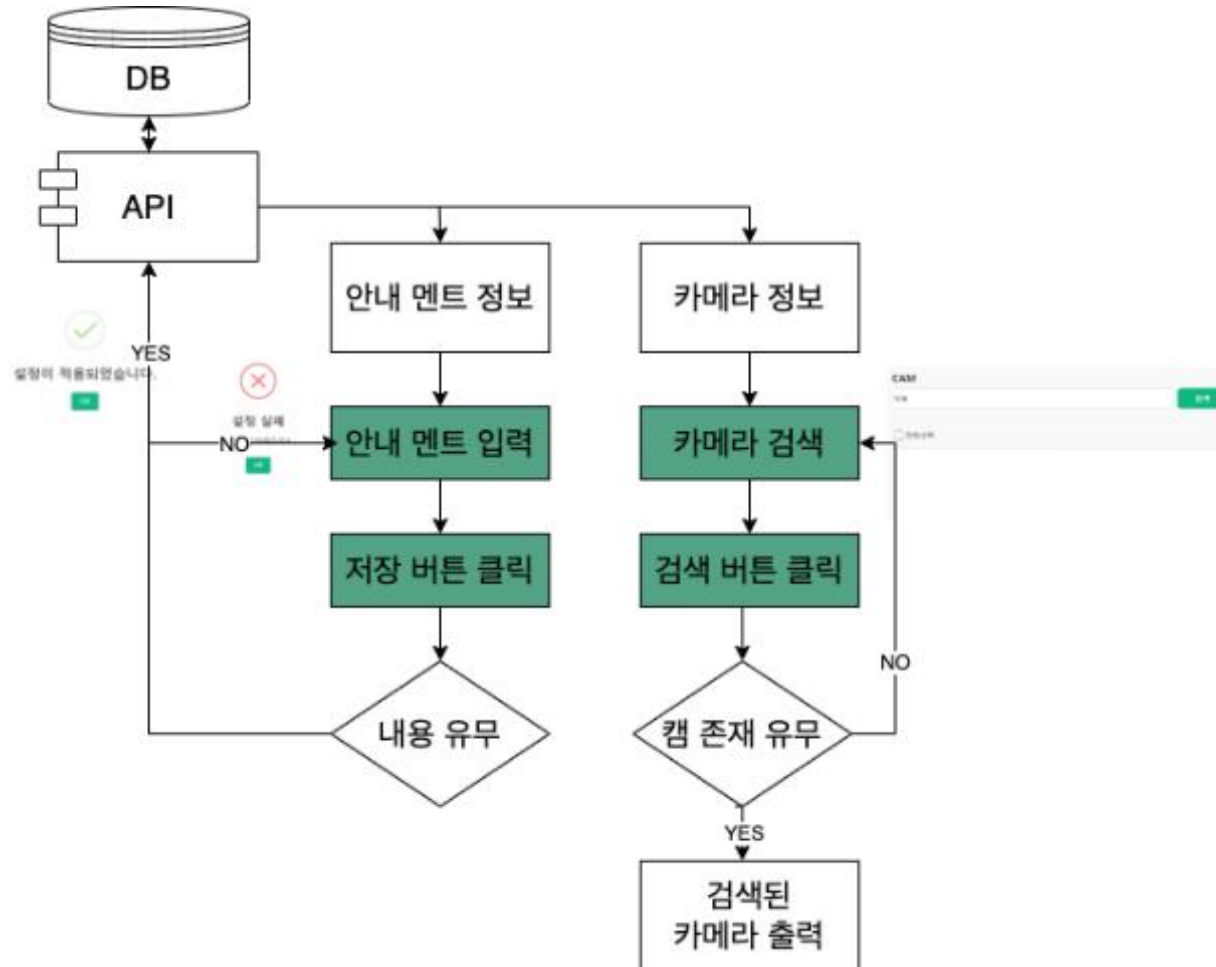
| 화면 설계서 - UI/UX 설계

분류	관리자 설정
기능	Sound(소리)



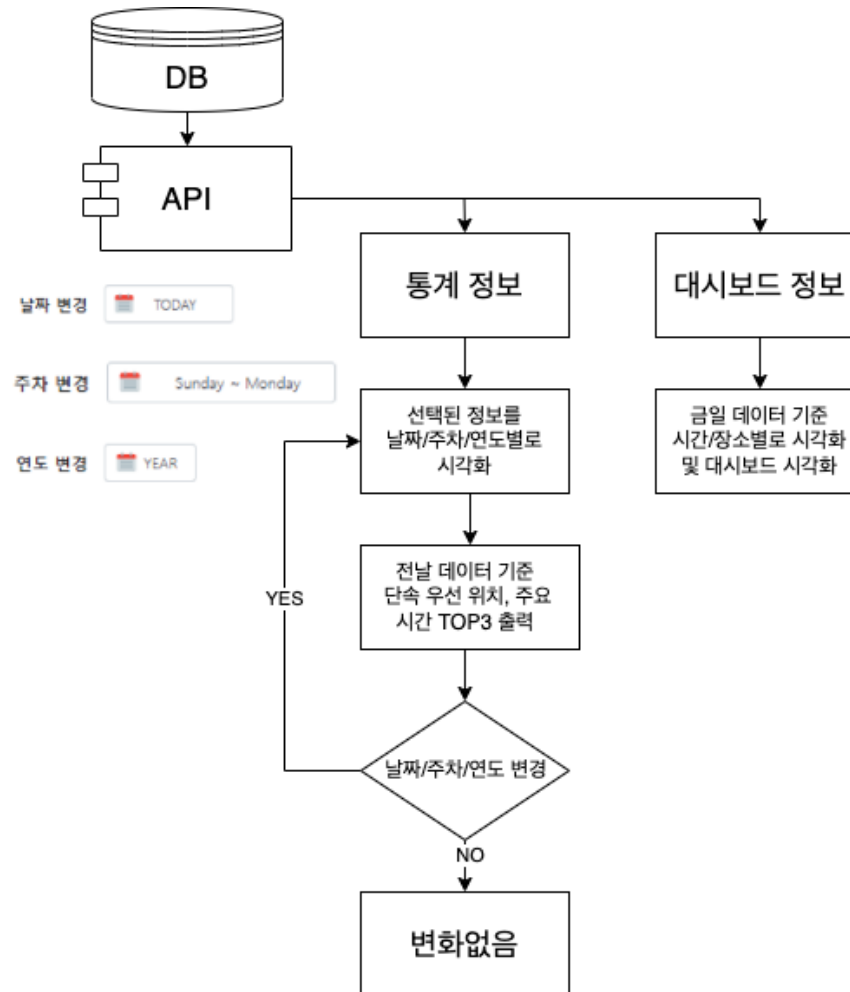
| 화면 설계서 — UI/UX 설계

분류	관리자 설정
기능	Sound(소리), Camera(카메라)

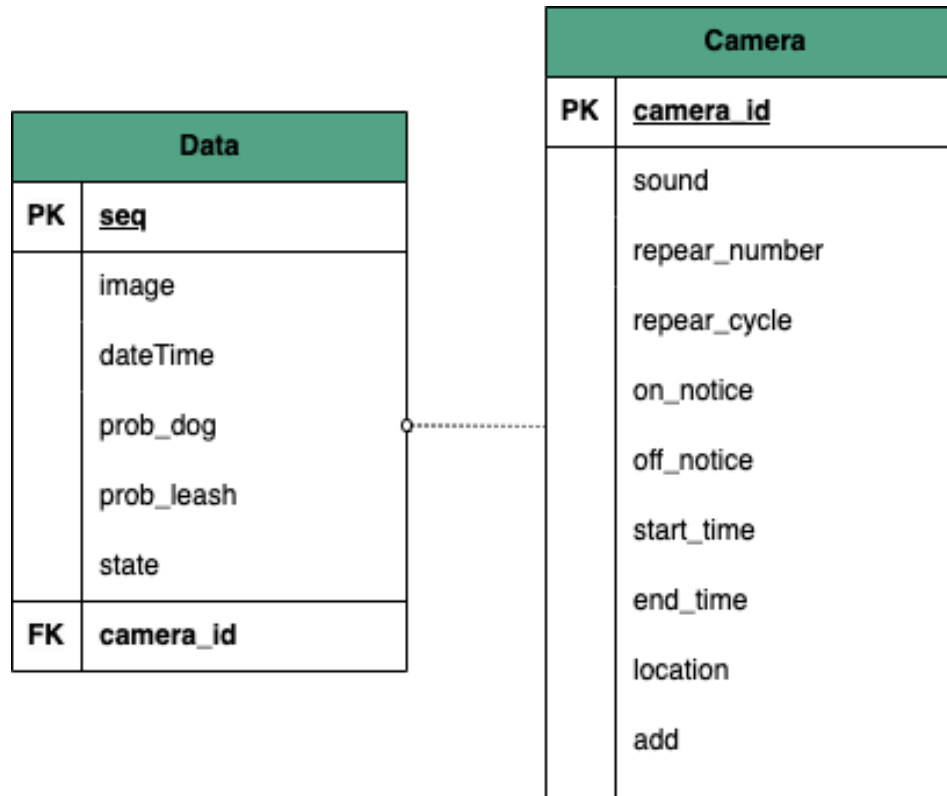


| 화면 설계서 — UI/UX 설계

분류	통계 페이지, 대시보드
기능	Stochastic(통계)



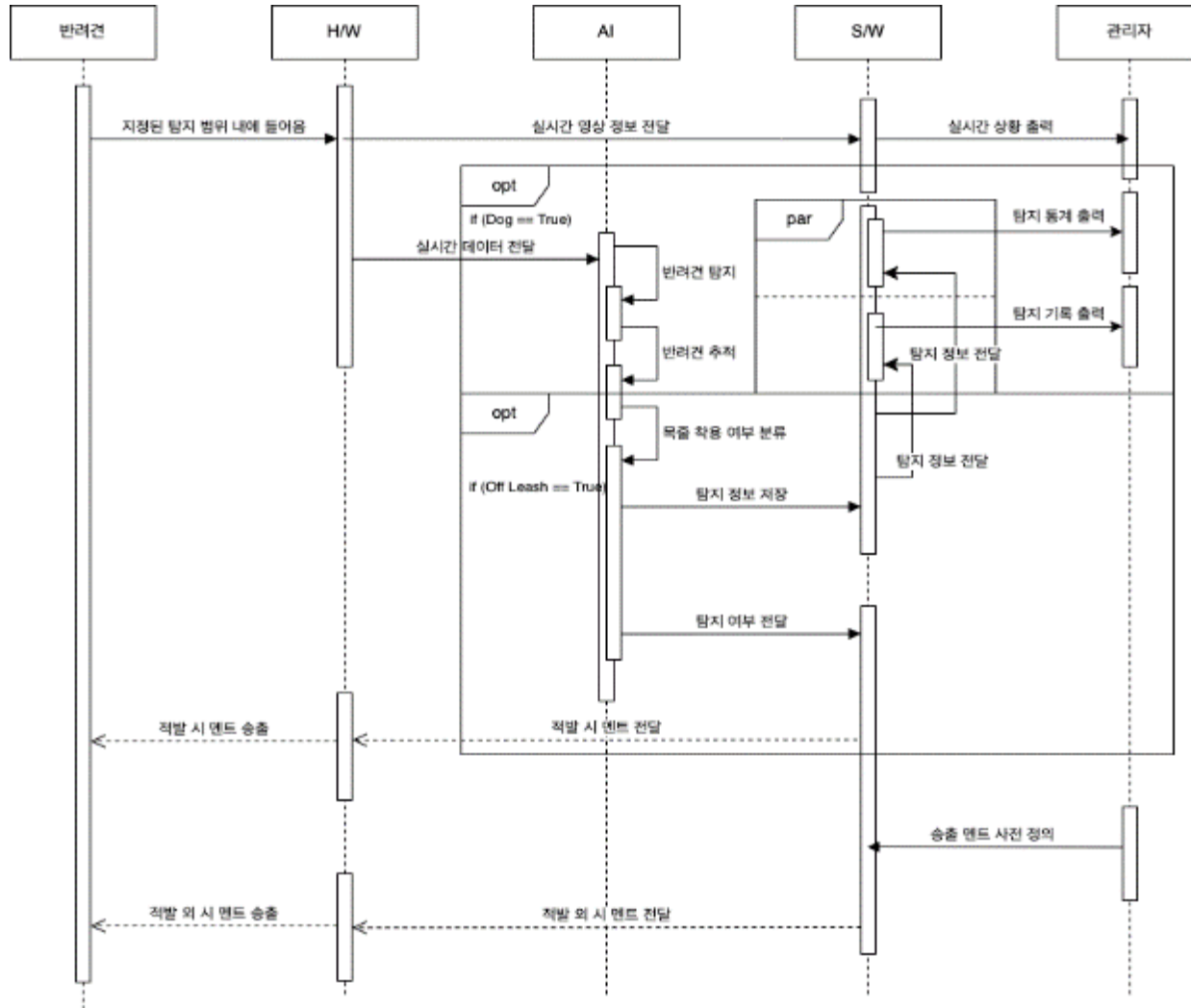
| 엔티티 관계도 - ERD



| 엔티티 관계도 - API 명세서

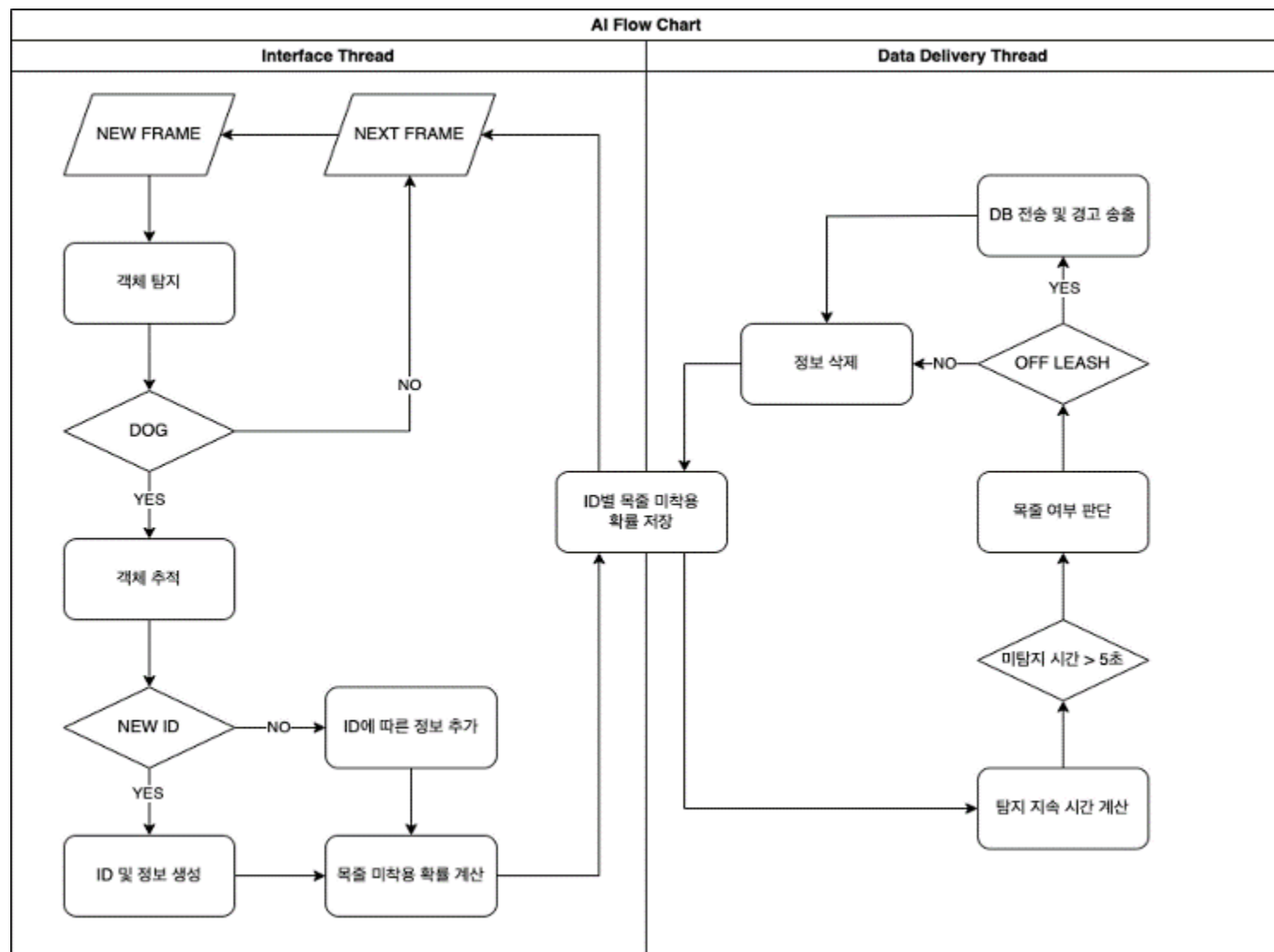
Index	Method	URL	Description
1	POST	/site_admin/	대시보드
2	POST	/site_admin/detail/	전체내역
3	GET	/site_admin/liveCam/	실시간 카메라 화면 조회
4	GET	/site_admin/statistics/	알림 통계 조회
5	PUT	/site_admin/setting/	카메라 시스템 설정
6	GET	/api/informations	전체내역 관련 이미지, 적발 위치, 시간 및 처리 현황 관련 데이터 처리.
7	GET	/api/dounut	금일 처리 현황 관련 데이터.
8	GET	/api/main_time	금일 시간별 적발 현황 관련 데이터
9	GET	/api/main_location	금일 장소별 적발 현황 관련 데이터
10	GET	/api/main_state/<seq>/<state>	처리 완료 시 관련 정보 DB로 전송
11	GET	/api/hour_location_graph/	시간별 통계 관련 데이터
12	GET	/api/day_location_graph/	일별 통계 관련 데이터
13	GET	/api/month_location_graph/	월별 통계 관련 데이터
14	GET	/api/place_three_graph	적발 우선 위치 관련 데이터
15	GET	/api/time_three_graph	적발 우선 시간 관련 데이터

| 기능 처리도(기능 흐름도)



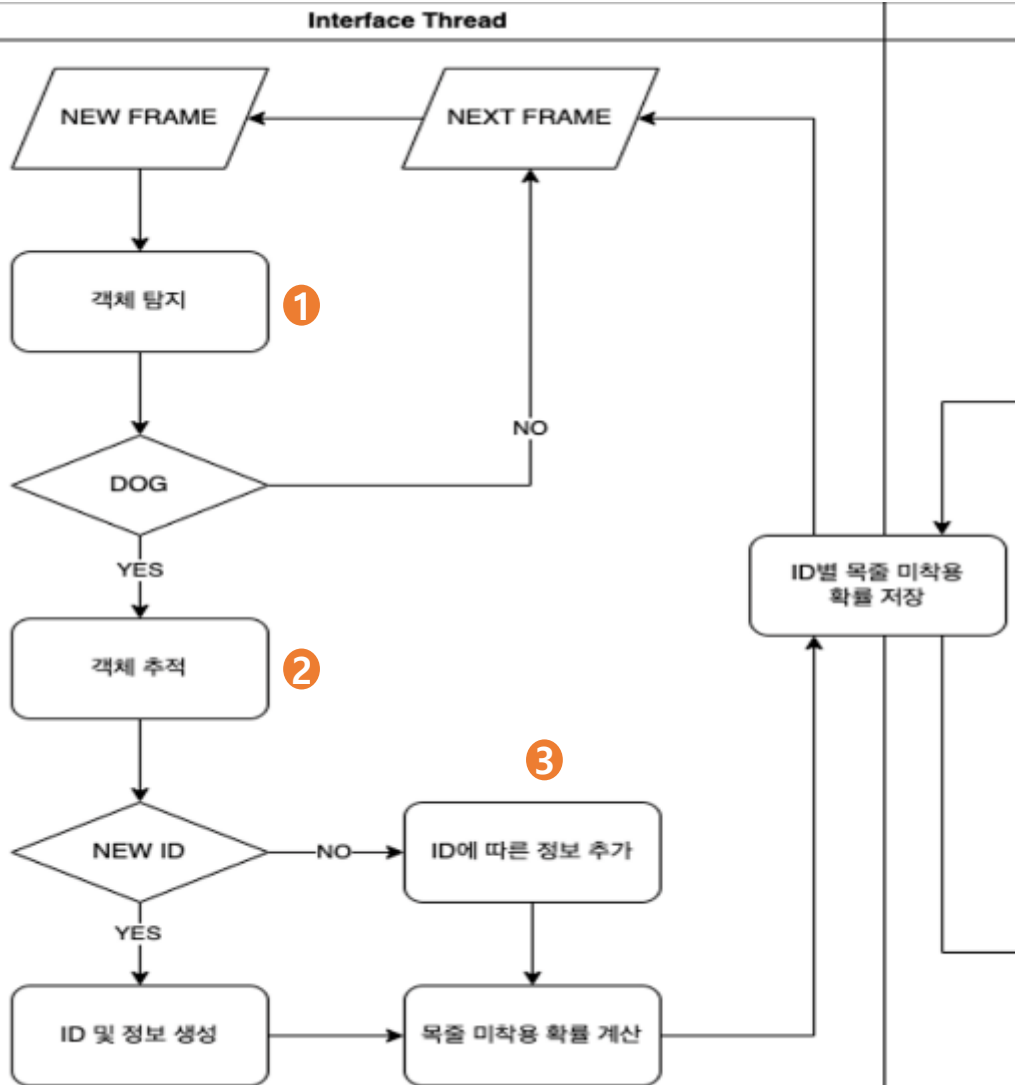
| 알고리즘 명세서 - Flowchart

AI 전체 알고리즘 명세서



| 알고리즘 명세서 — Flowchart

Interface Thread



Description

①

- 영상의 프레임이 입력되고 반려견 객체 탐지 모델의 추론 후 반려견이 존재하면 존재 영역의 경계 상자 좌표를 반환한다.

②

- 객체 추적 알고리즘을 통해 탐지된 반려견에 ID를 부여하고 관련 정보를 저장한다. 이전에 탐지된 반려견과 동일한 반려견일 경우 같은 ID에 정보를 추가한다.

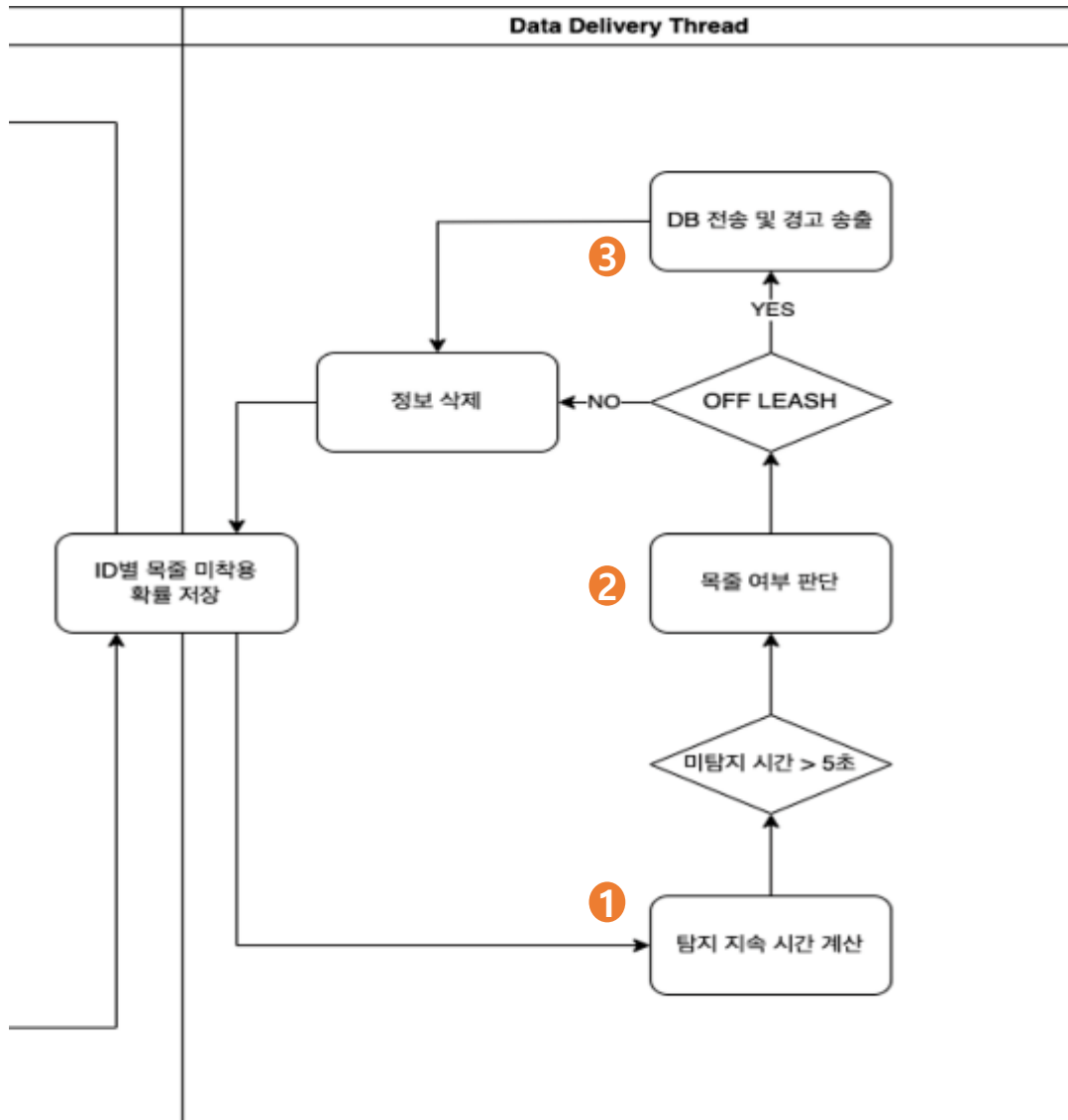
③

- 목줄 분류 모델을 통해 ID가 부여된 반려견의 목줄 미착용 확률을 계산하여 ID와 관련 정보와 함께 저장한다.

- 다음 프레임을 입력으로 알고리즘 반복

| 알고리즘 명세서 — Flowchart

Data Delivery Thread



Description

①

- ID별로 저장된 반려견의 탐지 지속 시간을 계산한다.

②

- 만약 미탐지시간이 5초 이상일 경우 저장되어있는 정보들을 바탕으로 최종적으로 목줄 착용 여부를 판단한다.

③

- 만약 목줄 미착용의 결과가 나오면 관리자 웹페이지로 데이터를 전송하고 하드웨어로 경고를 송출한 후 정보를 삭제하며, 목줄 착용의 결과가 나올 경우 ID와 관련 정보를 삭제한다.

- 다음 프레임을 입력으로 알고리즘 반복

| 알고리즘 상세 설명서

추론 스레드(Inference Thread) 알고리즘 상세 설명

1. 반려견 탐지(객체 탐지)

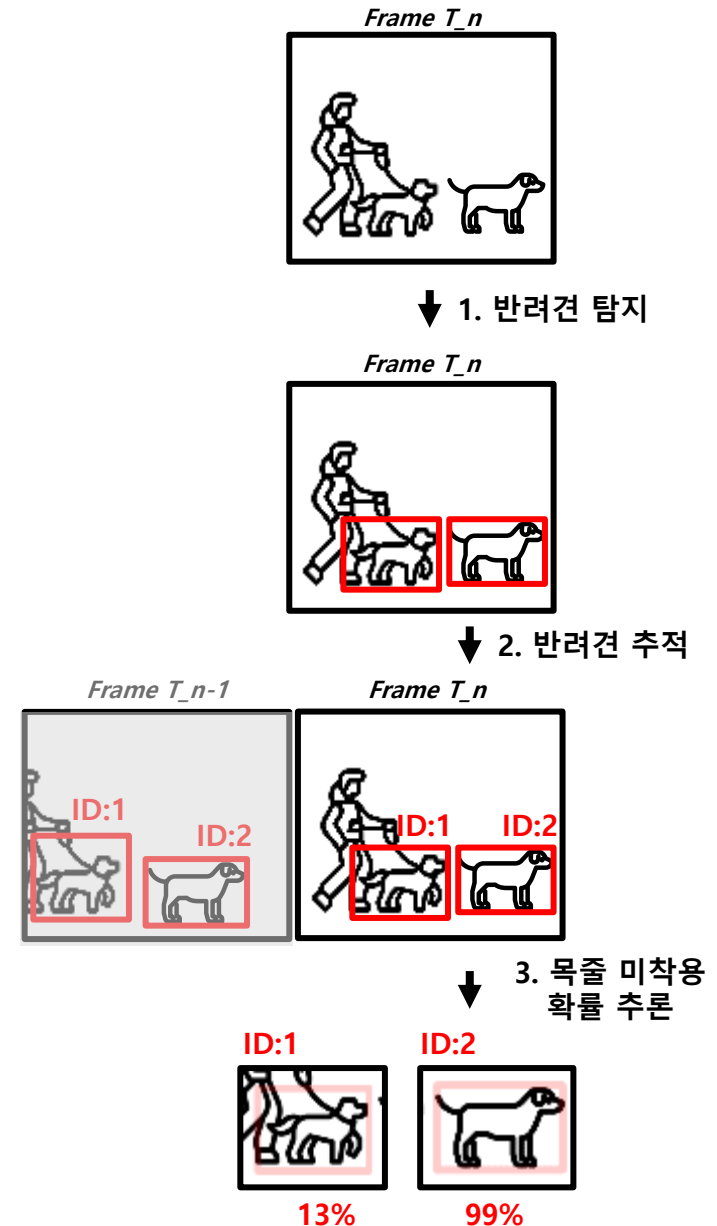
- 입력되는 영상 프레임에서 반려견이 존재하는 영역의 경계 상자 값, 경계 상자가 객체를 포함하고 있을 확률 반환.
- 사전 학습된 YOLOv8_n을 커스텀 데이터셋으로 전이 학습시킨 객체 탐지 모델을 반려견 탐지 모델로 사용.

2. 반려견 추적(객체 추적)

- 객체 탐지 결과로 과거 프레임과 현재 프레임에서 탐지된 반려견이 동일 객체인지 판단하여, 동일 객체일 경우 같은 ID 부여.
- ByteTrack 객체 추적 알고리즘을 활용하여 단일 프레임 기준의 목줄 착용 여부 판단에서 벗어나, 동일한 반려견을 연속된 여러 프레임에서 추적하여 목줄 착용 여부를 보다 정확하게 판단.

3. 목줄 미착용 확률 추론(이미지 분류)

- 탐지되는 반려견의 경계 상자의 높이와 넓이가 상이함을 고려해 경계 상자의 30% 확장한 영역 이미지를 입력으로 하여 목줄 미착용 확률을 추론.



| 알고리즘 상세 설명서

데이터 전송 스레드(Data Delivery Thread) 알고리즘 상세 설명

1. 반려견 별 탐지 지속 시간 측정

- 모델 추론 스레드에서 추가되는 반려견 별 정보 데이터의 업데이트 여부를 확인.
- 더이상 정보가 업데이트되지 않는다면, 해당 반려견이 프레임 밖으로 벗어났다고 판단.

2. 목줄 착용 여부 최종 판단

- 특정 반려견이 더 이상 탐지 되지 않을 때, 해당 반려견이 탐지된 모든 프레임을 고려하여 반려견의 목줄 착용 여부를 최종 판단 (주변 사물이나 사람에 의해 목줄이 가려져 잘못된 데이터를 전송하는 것을 방지하기 위한 과정임.)

3. 데이터 전송

- 목줄 미착용 반려견 정보를 서버로 전송.

Frame 별 목줄 미착용 확률

Id/frame num	Frame Tn-1	Frame Tn
ID 1	10%	13%
ID 2	30%	99%



다음 프레임 추론 결과 append

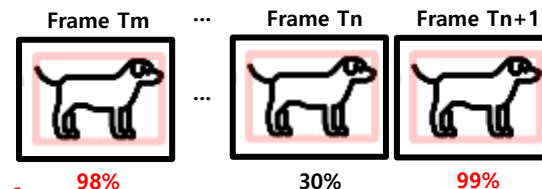
Id/frame num	Frame Tn-1	Frame Tn	Frame Tn+1
ID 1	10%	13%	80%
ID 2	30%	99%	-



Id/frame num	Frame Tn-1	Frame Tn	Frame Tn+1	Frame Tn+2
ID 1	10%	13%	80%	20%
ID 2	30%	99%	-	-

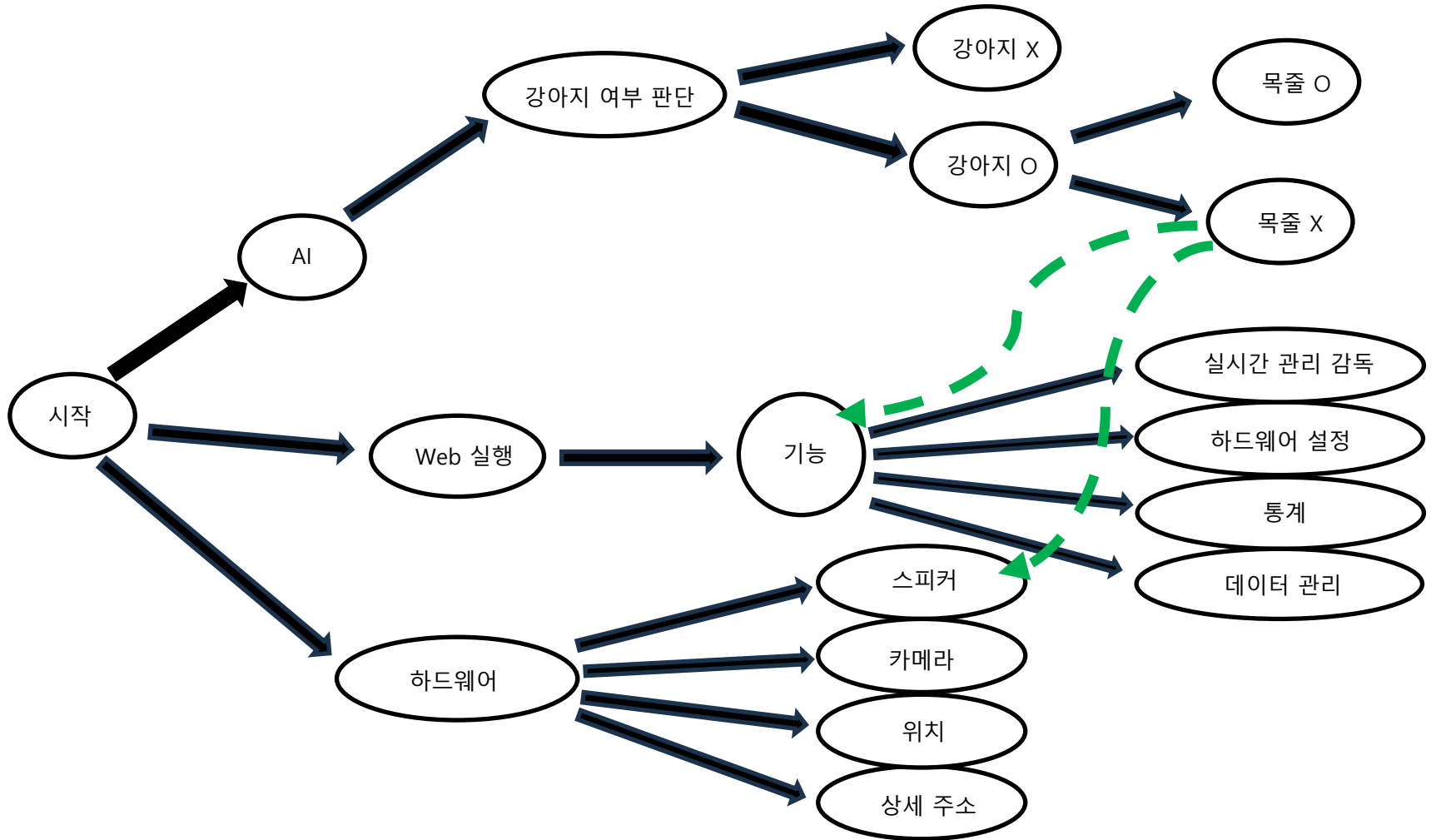


정보 업데이트 멈춤 → 해당 반려견이 더 이상 탐지되지 않음

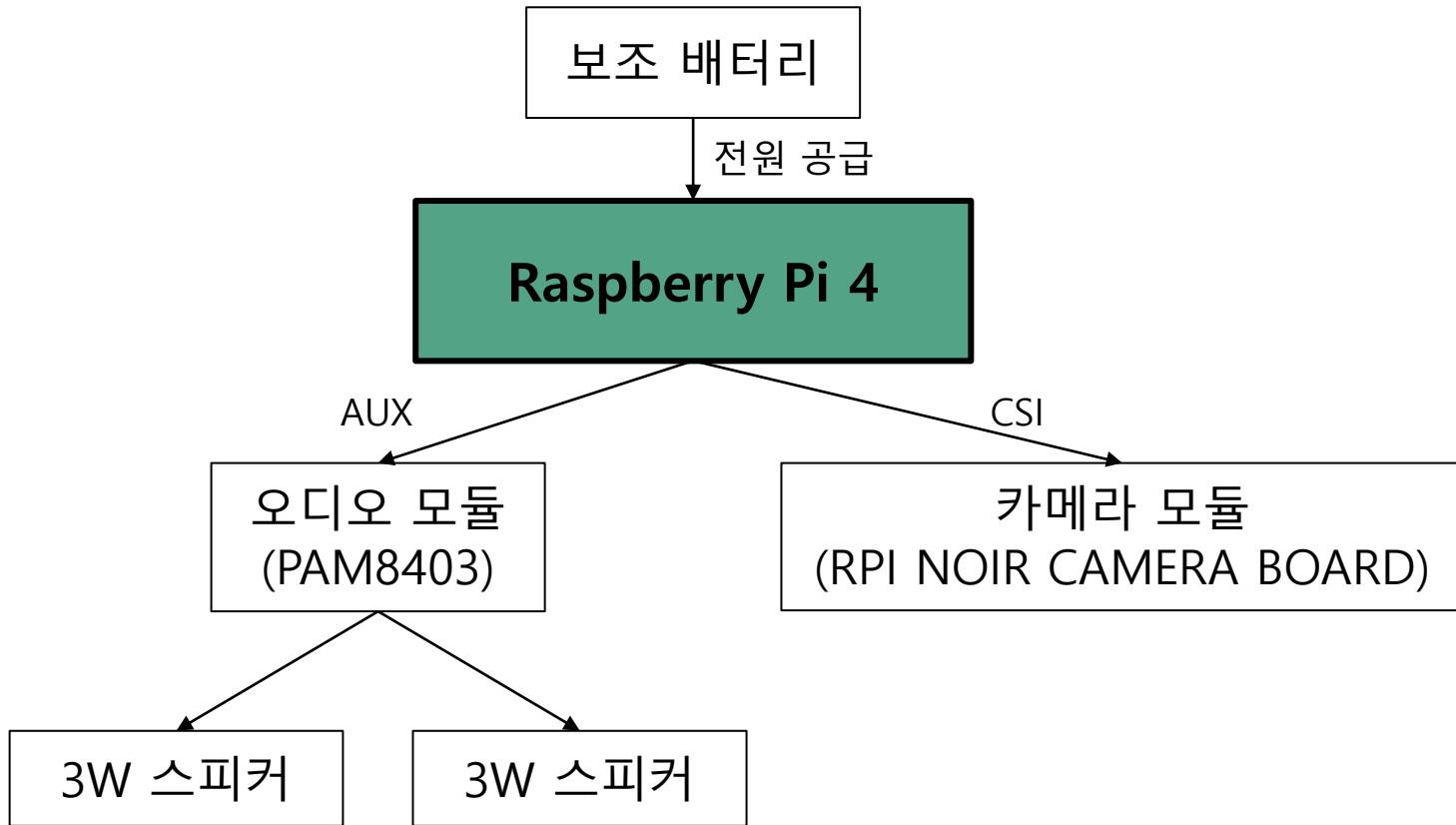


반려견 ID2 : 목줄 미착용

| 데이터 수집처리 명세서

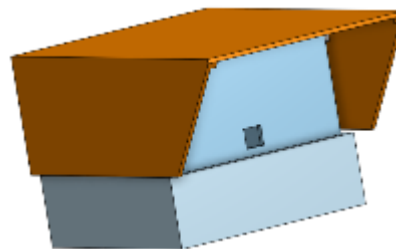
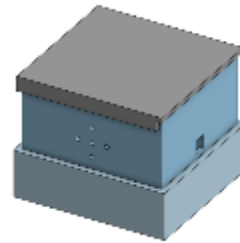
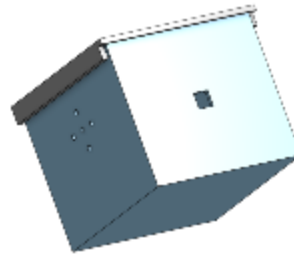


| 하드웨어 설계도 - 구성도



| 하드웨어 설계도 - 구성도

하드웨어 외관 설계도



| 프로그램 - 목록

기능 분류	기능 번호	기능
Declaration (신고)	Decl-01-01	목줄 미착용 건 신고 기능.
	Decl-01-02	목줄 미착용 신고 시 담당자 선택 기능.
	Decl-01-03	잘못 탐지 시 삭제 기능.
Camera (카메라)	Cam-01-01	기본적인 CCTV 기능.
	Cam-01-02	CCTV 확대 기능.
	Cam-01-03	카메라 검색 기능.
Sound (소리)	Snd-01-01	안내 멘트 설정.
	Snd-01-02	안내 멘트 주기 설정
	Snd-01-03	안내 멘트 반복 횟수 설정
	Snd-01-04	안내 멘트 음향 조절
	Snd-01-05	안내 멘트 송출 시간 설정.
Stochastic (통계)	Stc-01-01	금일 내역 시간, 장소 별 요약 및 총 내역 정리 기능.
	Stc-01-02	통계 산출을 통해 단속 우선 위치 및 시간 정리 기능.
	Stc-01-03	특정 년도, 주차, 일 선택 시 관련 통계 확인 기능.
URL (페이지 이동)	Url-01-01	대시보드 페이지로 이동.
	Url-01-02	전체내역 페이지로 이동.
	Url-01-03	LIVECAM 페이지로 이동.
	Url-01-04	통계 페이지로 이동.
	Url-01-05	관리자 설정 페이지로 이동.

| 테이블 정의서 - ERD

Table	Column	Type	PK/FK	Null	설명
camera	camera_id	Int	FK	NOT NULL	카메라
	sound	Int		NOT NULL	카메라 음량
	repeat_number	Int		NOT NULL	안내멘트 반복 횟수
	repeat_cycle	Int		NOT NULL	안내멘트 반복 주기
	on_notice	Char		NOT NULL	평상시 안내 멘트
	off_notice	Char		NOT NULL	적발시 안내 멘트
	start_time	Int		NOT NULL	멘트 시작 시간
	end_time	Int		NOT NULL	멘트 종료 시간
	location	Char		NOT NULL	카메라 설치 위치
	add	Char		NOT NULL	상세 주소

| 테이블 정의서 - ERD

Table	Column	Type	PK/FK	Null	설명
data	seq	Int	PK	NOT NULL	시퀀스
	camera_id	Int	FK	NOT NULL	카메라
	image	Text		NOT NULL	목줄 미착용 반려견 이미지
	datetime	Datetime		NOT NULL	적발 날짜 및 시간
	Prob_dog	Float		NOT NULL	객체가 개일 확률
	Prob_leash	Float		NOT NULL	목줄 착용 확률
	state	Int		NOT NULL	대응

| 핵심소스코드(1)

주요 모델 로드 및 설정 코드

```

1  def yolo_action(self, frame):
2      """ EXECUTE YOLO """
3      self.results = self.yolo.track(frame,conf=0.6,tracker="bytetrack.yaml",persist=True)
4      self.annotated_frame = self.results[0].plot()
5
6      # id가 지정되지 않은 경우 패스
7      if self.results[0].boxes.id == None:
8          self.xyxy=[]
9          self.conf=[]
10         self.id=[]
11     else:
12         self.xyxy=self.results[0].boxes.xyxy.int().tolist()
13         self.conf=self.results[0].boxes.conf.float().tolist()
14         self.id=self.results[0].boxes.id.int().tolist()

```

```

1  def cls_action(self,img_array,probs=True):
2      """ EXECUTE CLS """
3      with torch.no_grad():
4          input_tensor = self.preprocess(Image.fromarray(img_array))
5          input_batch = input_tensor.unsqueeze(0)
6          output=self.cls_model(input_batch)
7          if probs:
8              return torch.argmax(output[0]).item(),output[0]
9          else:
10             return torch.argmax(output[0]).item()

```

| 핵심소스코드(2)

AI 알고리즘 파이프라인 실행 코드

```

1  # 모델 추론 파이프라인
2  def execute(self, cap_source=0):
3      cap=cv2.VideoCapture(cap_source)
4      while cap.isOpened():
5          success, frame = cap.read()
6          if success:
7              start_t = timeit.default_timer() # FPS 연산 시작
8              self.yolo_action(frame) # 플로 수행
9              if self.id==None: continue
10             for key in self.id:
11                 if key not in self.dogs_dict.keys(): # 새로운 강아지가 탐지될 때
12                     self.dogs_dict[key]=([], [], [], []) # 정보저장 공간 초기화
13             for i in range(len(self.id)): # 각 id별로 dogs_dict에 정보 저장
14                 xmin,ymin,xmax,ymax=tuple(map(int,self.crop(self.xyxy[i])))
15                 img_array=frame[max(0,ymin):min(frame.shape[0],ymax),max(0,xmin):min(frame.shape[1],xmax),:] # crop 이미지 배열
16                 ret=self.cls_action(img_array,probs=True) # Cls 결과
17                 print(self.idx_to_classes[ret[0]])
18                 self.dogs_dict[self.id[i]][0].append(img_array) # crop된 프레임 정보 추가
19                 self.dogs_dict[self.id[i]][1].append(self.conf[i]) # 개 확률 정보 추가
20                 self.dogs_dict[self.id[i]][2].append(ret[1][1]) # 미착용 확률 정보 추가
21                 self.dogs_dict[self.id[i]][3].append(ret[0]) # 0(착용) or 1(미착용) 정보 추가
22             terminate_t = timeit.default_timer() # FPS 연산 끝
23             FPS = 1./(terminate_t - start_t)
24             print(f'FPS : {FPS}, SPF : {terminate_t - start_t}')
25             if cv2.waitKey(1) & 0xFF == ord("q"):
26                 break
27         else:
28             break
29     cap.release()
30     cv2.destroyAllWindows()

```

| 핵심소스코드(3)

웹페이지 데이터 전송 파이프라인 코드

```

1 # 데이터 전송 파이프라인
2 def check_and_post(self):
3     last_lengths = {}
4     '''
5     예시
6     {
7         dog1: (남지된 횟수, 현재시간), # 현재시간까지 남지된 횟수를 저장
8         dog2: (남지된 횟수, 현재시간),
9     }
10    '''
11    while True:
12        for key in list(self.dogs_dict.keys()): # 각 반려견별 활동 시간 측정
13            current_length = len(self.dogs_dict[key][0]) # 특정 반려견 당시 횟수
14            if key not in last_lengths:
15                last_lengths[key] = (current_length, time.time()) # {독점반려견: (1회, 15시30분00초)}
16            else:
17                if current_length == last_lengths[key][0] and time.time() - last_lengths[key][1] >= 5: # 현재 당시 횟수와 마지막 당시 횟수가 동일하고 길이가 5초 이상 발견되지 않으면
18                    print(f'Dog number {key} hasn't been detected during 5 seconds .")
19                    if len(self.dogs_dict[key][3]) <= 3: # 길아지로 인식된 횟수가 3보다 작으면
20                        print(f'dog id{key}은 길아지가 아닌거 같습니다.')
21                        del self.dogs_dict[key] # 정보공간에서 해당 데이터 삭제
22                        del last_lengths[key]
23                    elif np.mean(self.dogs_dict[key][3]) >= 0.5: # 길아지로 인식된 횟수가 3보다 특정반려견 정보를 숨김하여 복습 미작동으로 판단되면
24                        name=datetime.now().strftime('%y%m%d%H%M%S')+'_without'
25                        prob_off_leash=max(self.dogs_dict[key][2]) # 미작동 치를
26                        choice=self.dogs_dict[key][2].index(prob_off_leash)
27                        image=self.dogs_dict[key][0][choice]
28                        conf=self.dogs_dict[key][1][choice] # 개 이름
29                        cv2.imwrite(f'D:/track_ver3/bins/{name}_dogie{key}.jpg', image)
30                        image_file=f'D:/track_ver3/bins/{name}_dogie{key}.jpg'
31                        with open(image_file, 'rb') as input_image:
32                            image_data = input_image.read()
33
34                        # DB에 정보 전달
35                        try:
36                            mycursor = self.mydb.cursor()
37                            sql = f"INSERT INTO data (image,dataTime, prob_dog, prob_leash,state,camera_id) \
38                                VALUES (%s, '{datetime.now().replace(microsecond=0)}', '{conf}', '{prob_off_leash}', '{1}', '{2}')"
39                            mycursor.execute(sql, (pymysql.Binary(image_data),))
40                            self.mydb.commit()
41                        except Exception as ex:
42                            print('Error 발생')
43                            print(ex)
44                        finally:
45                            mycursor.close()
46                            del self.dogs_dict[key]
47                            del last_lengths[key]
48                    else:
49                        del self.dogs_dict[key]
50                        del last_lengths[key]
51                elif current_length != last_lengths[key][0]:
52                    print(f'dog id{key} : (current_length)번 입사됨')
53                    last_lengths[key] = (current_length, time.time())
54                elif current_length != last_lengths[key][0]:
55                    print(f'dog id{key} : (current_length)번 항지됨')
56                    last_lengths[key] = (current_length, time.time())
57            time.sleep(0.5)
58

```


참고 자료 - 오키도기 웹사이트

로고 클릭 시 웹사이트 링크로 이동

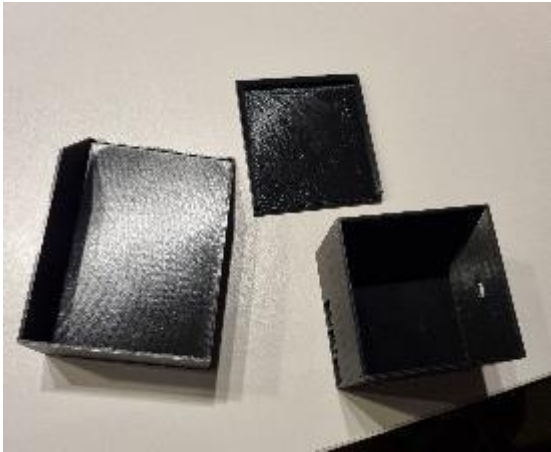
오키도기

참고 자료 - AI 시연 영상

사진 클릭 시 영상 링크로 이동



참고 자료 - 외관



3D 출력물



내부 구성



구동 외관