IoT 열선 제어 및 CCTV 모니터링 앱 개발 프로젝트 지원서

지원일: 2025년 12월 18일 **지원업체:** [업체명] **담당자:** [담당자명] / [연락처] / [이메일]

1. 프로젝트 이해

1.1 프로젝트 개요

2022년 개발된 기존 '열선 제어 앱(Android/Kotlin)'을 **전면 리뉴얼**하여, Android/iOS 동시 지원 하이브리드 앱으로 재구축하고 관리자 페이지를 개발하는 프로젝트입니다.

1.2 핵심 목표

목표	내용
플랫폼 확장	Android 전용 → Android + iOS 동시 지원
UI/UX 현대화	노후화된 디자인 → 직관적, 현대적 IoT 앱 스타일
기능 통합	열선 제어 + CCTV 모니터링 통합 환경
관리 효율화	관리자 페이지를 통한 회원/기기/데이터 관리

1.3 요구사항 분석

구분	요구사항	이해도
기획	기존 앱 분석, 신규 UI/UX 설계	☑
디자인	모바일 앱 + 관리자 웹 디자인	☑
앱	하이브리드 앱 (Flutter/RN)	☑
앱	열선 컨트롤러 프로토콜 연동	☑
앱	CCTV 실시간 스트리밍	☑
웹	관리자 페이지 (회원/기기/데이터 관리)	☑
서버	API 개발, DB 구축	☑

2. 기술 제안

2.1 기술 스택 제안

구분	기술	선정 사유
앱	Flutter 3.x	단일 코드베이스, 네이티브 수준 성능, 풍부한 UI 위젯

구분	기술	선정 사유
앱 언어	Dart	Flutter 공식 언어, 빠른 개발 속도
백엔드	Node.js (Express/NestJS)	실시간 통신(WebSocket) 최적화, 빠른 응답 속도
DB	PostgreSQL + Redis	안정적 데이터 저장 + 실시간 캐싱
관리자 웹	React.js + TypeScript	컴포넌트 재사용, 타입 안정성
영상 스트리밍	RTSP → HLS/WebRTC 변환	모바일 호환성, 저지연 스트리밍
인프라	AWS / NCP	확장성, 안정성

2.2 Flutter 선정 사유

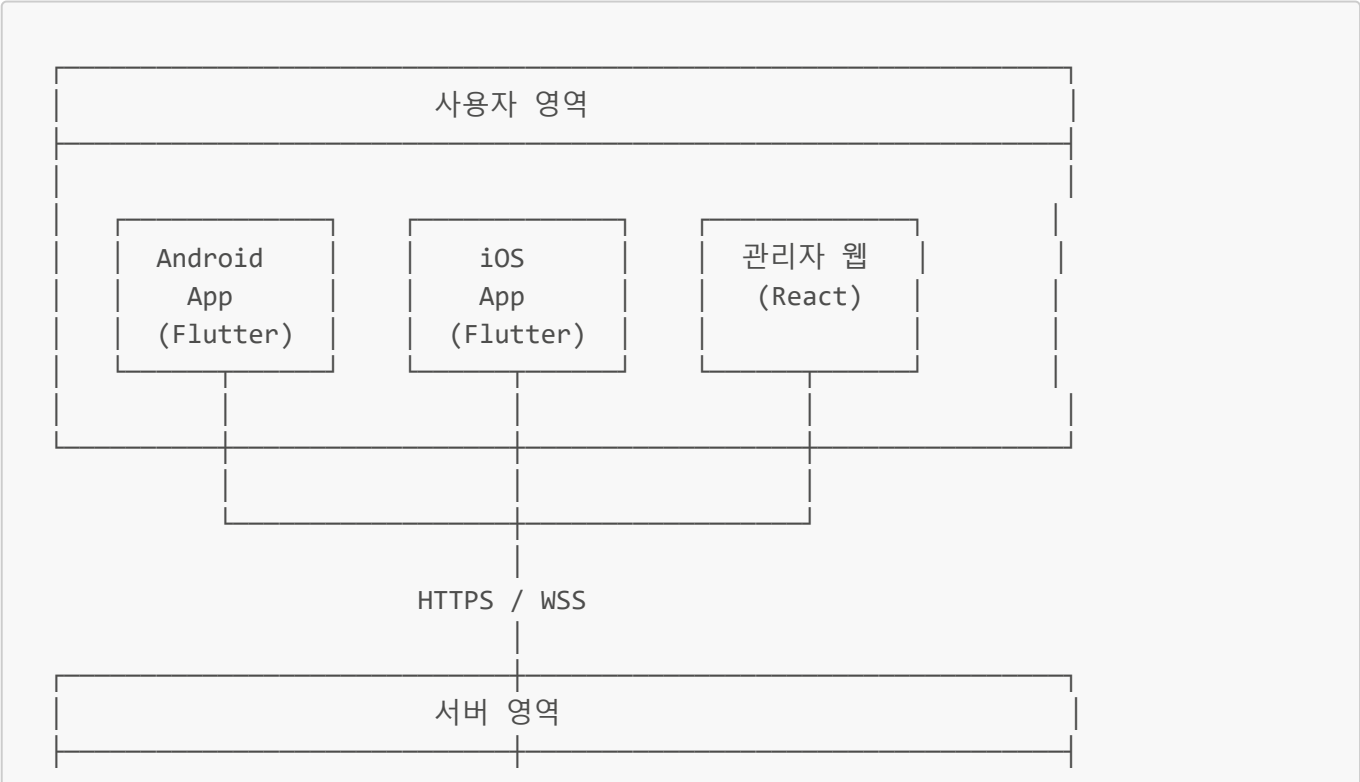
항목	Flutter	React Native
성능	네이티브 수준 (Skia 렌더링)	Bridge 통한 통신 (상대적 느림)
UI 일관성	Android/iOS 동일 UI	플랫폼별 차이 발생 가능
개발 속도	Hot Reload, 풍부한 위젯	Hot Reload 지원
영상 처리	flutter_vlc_player 등 성숙한 패키지	영상 관련 패키지 제한적
커뮤니티	Google 공식 지원, 급성장 중	Meta 지원, 성숙한 생태계

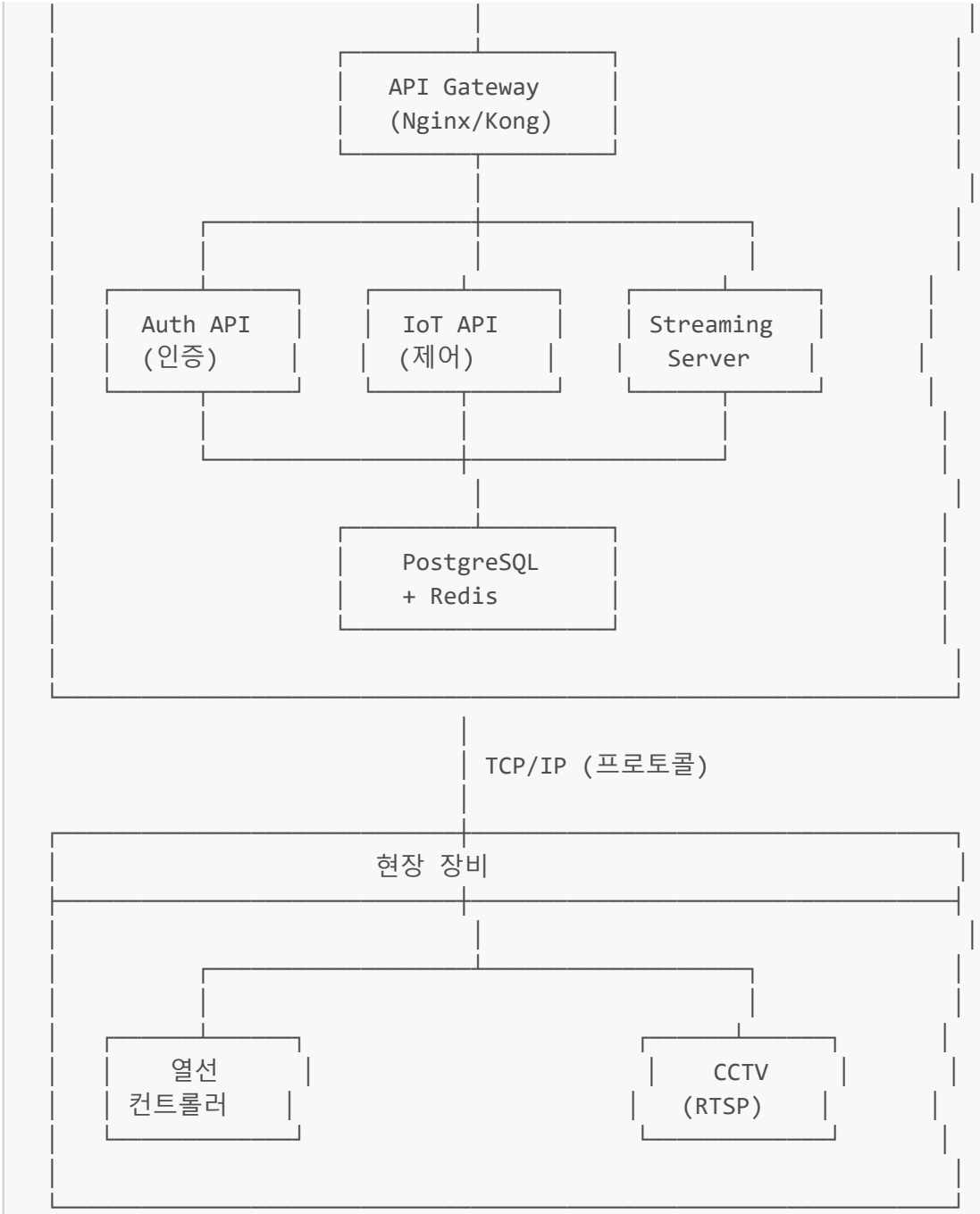
결론: IoT 제어 + 영상 스트리밍 특성상 **Flutter** 권장

---

3. 시스템 아키텍처

3.1 전체 구성도





3.2 앱 화면 구조 (IA)

```
┌─────────────────────────────────────┐
│           열선제어 앱                │
├─────────────────────────────────────┤
│                                     │
│  ├── 로그인/회원가입                  │
│  │   ├── 일반 로그인                 │
│  │   ├── 소셜 로그인 (선택)           │
│  │   └── 회원가입                    │
│  │                                  │
│  ├── 대시보드 (홈)                   │
│  │   ├── 전체 현황 요약              │
│  │   ├── 빠른 제어 버튼              │
```

```
│   │   └── 알림 영역                          │
│   │                                          │
│   ├── 열선 제어                              │
│   │   ├── 컨트롤러 목록                      │
│   │   ├── 개별 제어 (ON/OFF, 온도 설정)      │
│   │   ├── 그룹 제어                          │
│   │   └── 상태 모니터링 (실시간)             │
│   │                                          │
│   ├── CCTV 모니터링                          │
│   │   ├── 카메라 목록                        │
│   │   ├── 실시간 스트리밍                    │
│   │   └── 멀티뷰 (4분할 등)                  │
│   │                                          │
│   ├── 이력/통계                              │
│   │   ├── 제어 이력                          │
│   │   ├── 온도 추이 그래프                   │
│   │   └── 데이터 내보내기                    │
│   │                                          │
│   └── 설정                                   │
│       ├── 알림 설정                          │
│       ├── 계정 관리                          │
│       └── 앱 정보                            │
│                                              │
└──────────────────────────────────────────────┘
```

3.3 관리자 페이지 구조

```
┌──────────────────────────────────────────────┐
│              관리자 페이지                   │
├──────────────────────────────────────────────┤
│                                              │
│   ├── 대시보드                               │
│   │   ├── 전체 현황 (회원/기기/알림)         │
│   │   └── 실시간 모니터링                    │
│   │                                          │
│   ├── 회원 관리                              │
│   │   ├── 회원 목록/검색                     │
│   │   ├── 회원 상세/수정                     │
│   │   ├── 승인 대기 목록                     │
│   │   └── 권한 관리                          │
│   │                                          │
│   ├── 기기 관리                              │
│   │   ├── 열선 컨트롤러 목록                 │
│   │   ├── CCTV 목록                          │
│   │   ├── 기기 등록/수정/삭제                │
│   │   └── 회원-기기 매핑                     │
│   │                                          │
│   ├── 데이터 관리                            │
│   │   ├── 센싱 데이터 조회                   │
│   │   ├── 제어 로그 조회                     │
│   │   └── 데이터 내보내기 (Excel/CSV)        │
```

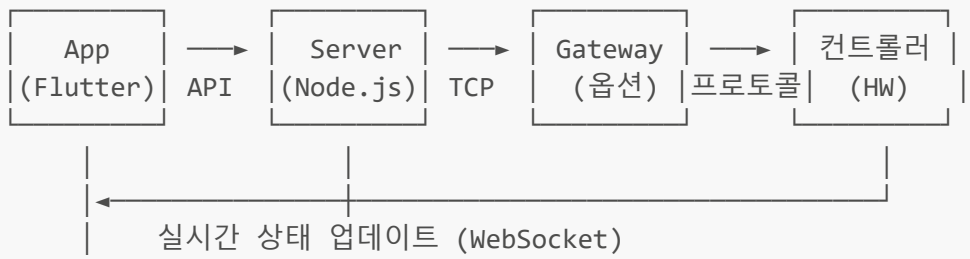
```
│  │                                    │
│  └── 시스템 설정                      │
│      ├── 알림 설정                    │
│      └── 시스템 로그                  │
│                                       │
└───────────────────────────────────────┘
```

---

4. 상세 기능 구현 계획

4.1 열선 컨트롤러 연동

통신 흐름



주요 기능

기능	설명	구현 방식
원격 제어	ON/OFF, 온도 설정	프로토콜 명령 전송
상태 조회	현재 온도, 작동 상태	주기적 폴링 or 이벤트 기반
실시간 업데이트	상태 변경 즉시 반영	WebSocket
그룹 제어	다수 컨트롤러 일괄 제어	순차/병렬 명령 전송

프로토콜 연동 (예시)

```dart
// Flutter - 열선 제어 예시
class HeaterController {
  final WebSocketChannel _channel;

  // 열선 ON/OFF 제어
  Future<void> setPower(String deviceId, bool isOn) async {
    final command = {
      'type': 'CONTROL',
      'deviceId': deviceId,
      'action': isOn ? 'POWER_ON' : 'POWER_OFF',
      'timestamp': DateTime.now().toIso8601String(),
    };
    _channel.sink.add(jsonEncode(command));
```

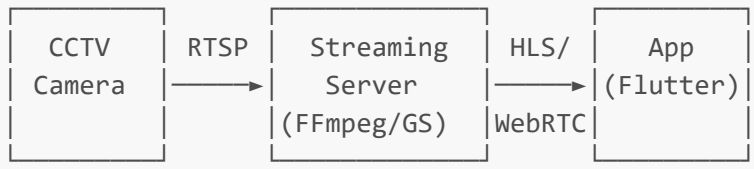
```
  }

  // 온도 설정
  Future<void> setTemperature(String deviceId, double temp) async {
    final command = {
      'type': 'CONTROL',
      'deviceId': deviceId,
      'action': 'SET_TEMP',
      'value': temp,
    };
    _channel.sink.add(jsonEncode(command));
  }

  // 상태 수신 스트림
  Stream<DeviceStatus> get statusStream => _channel.stream
      .map((data) => DeviceStatus.fromJson(jsonDecode(data)));
}
```

4.2 CCTV 영상 스트리밍

스트리밍 아키텍처



스트리밍 방식 비교

방식	지연 시간	호환성	구현 난이도
HLS	10~30초	높음 (브라우저/앱)	낮음
WebRTC	0.5~2초	중간	높음
RTMP→HLS	5~15초	높음	중간

권장: 일반 모니터링 목적이면 **HLS**, 실시간성 중요하면 **WebRTC**

Flutter 영상 플레이어

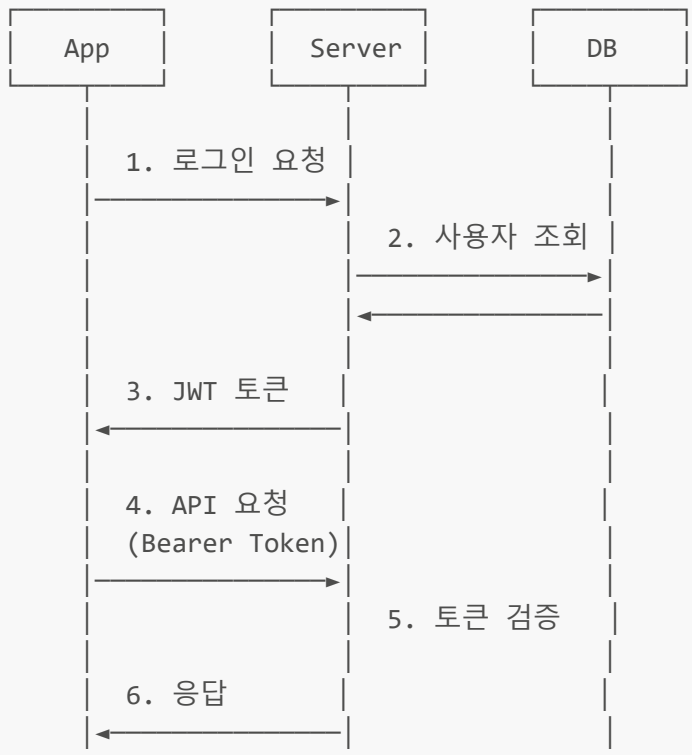
```
// Flutter - CCTV 스트리밍 예시
class CCTVPlayer extends StatefulWidget {
  final String streamUrl;

  @override
  Widget build(BuildContext context) {
```

```
      return VlcPlayer(
        controller: VlcPlayerController.network(
          streamUrl,
          hwAcc: HwAcc.full,
          autoPlay: true,
          options: VlcPlayerOptions(
            advanced: VlcAdvancedOptions([
              VlcAdvancedOptions.networkCaching(1000),
            ]),
          ),
        ),
        aspectRatio: 16 / 9,
      );
    }
  }
```

4.3 회원 및 권한 관리

인증 플로우



권한 체계

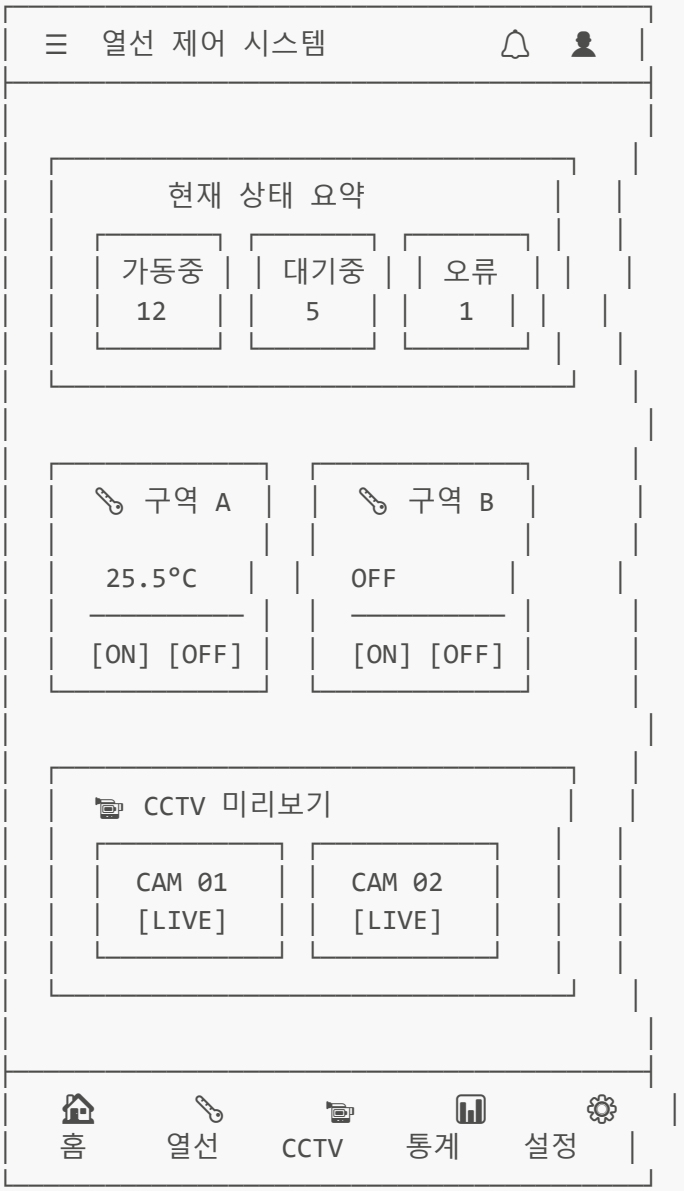
권한	설명	기능 범위
일반 사용자	앱 사용자	할당된 기기 제어/모니터링
관리자	웹 관리자	회원/기기/데이터 관리
슈퍼 관리자	최고 권한	시스템 설정, 관리자 관리

5. UI/UX 디자인 방향

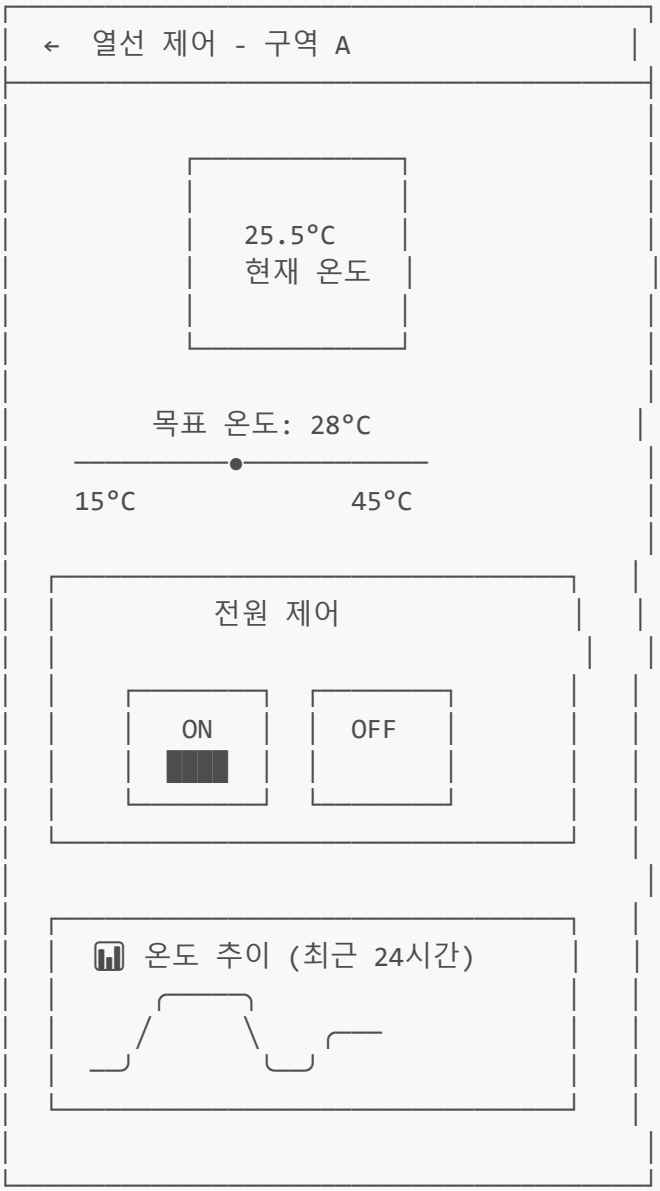
5.1 디자인 컨셉

항목	방향
스타일	모던 IoT / 스마트홈 스타일
컬러	다크 모드 기본 + 라이트 모드 지원
아이콘	직관적 아이콘, 애니메이션 효과
레이아웃	카드 기반 대시보드, 그리드 구성

5.2 대시보드 화면 (예시)

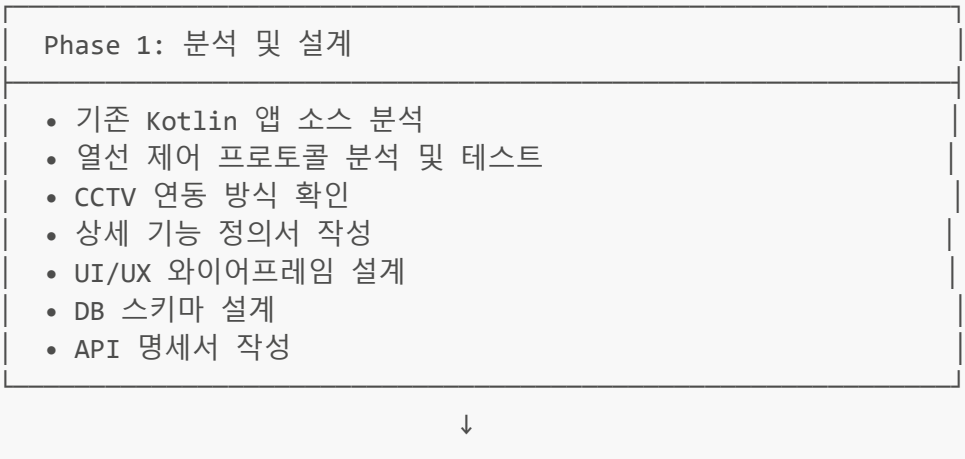


5.3 열선 제어 상세 화면 (예시)



6. 개발 계획

6.1 개발 단계



```
┌─────────────────────────────────────────────────────────┐
│ Phase 2: 디자인                                         │
├─────────────────────────────────────────────────────────┤
│ • 앱 UI 디자인 (Figma)                                  │
│ • 관리자 웹 UI 디자인                                   │
│ • 디자인 시스템 구축 (컬러, 타이포, 컴포넌트)           │
│ • 디자인 리뷰 및 수정                                   │
└─────────────────────────────────────────────────────────┘
                          ↓
┌─────────────────────────────────────────────────────────┐
│ Phase 3: 백엔드 개발                                    │
├─────────────────────────────────────────────────────────┤
│ • 서버 환경 구축 (AWS/NCP)                              │
│ • DB 구축 (PostgreSQL)                                  │
│ • 인증 API 개발 (JWT)                                   │
│ • 열선 제어 API 개발                                    │
│ • CCTV 스트리밍 서버 구축                               │
│ • WebSocket 실시간 통신 구현                            │
└─────────────────────────────────────────────────────────┘
                          ↓
┌─────────────────────────────────────────────────────────┐
│ Phase 4: 앱 개발                                        │
├─────────────────────────────────────────────────────────┤
│ • Flutter 프로젝트 셋업                                 │
│ • 공통 컴포넌트 개발                                    │
│ • 로그인/회원가입 화면                                  │
│ • 대시보드 화면                                         │
│ • 열선 제어 화면 (목록, 상세, 제어)                    │
│ • CCTV 모니터링 화면                                    │
│ • 이력/통계 화면                                        │
│ • 설정 화면                                             │
│ • Push 알림 연동                                        │
└─────────────────────────────────────────────────────────┘
                          ↓
┌─────────────────────────────────────────────────────────┐
│ Phase 5: 관리자 웹 개발                                 │
├─────────────────────────────────────────────────────────┤
│ • React 프로젝트 셋업                                   │
│ • 대시보드 페이지                                       │
│ • 회원 관리 페이지                                      │
│ • 기기 관리 페이지                                      │
│ • 데이터 관리 페이지                                    │
│ • 시스템 설정 페이지                                    │
└─────────────────────────────────────────────────────────┘
                          ↓
┌─────────────────────────────────────────────────────────┐
│ Phase 6: 통합 테스트 및 배포                            │
├─────────────────────────────────────────────────────────┤
│ • 앱 ↔ 서버 ↔ 하드웨어 통합 테스트                      │
│ • QA 테스트 (기능, 성능, 보안)                          │
│ • 버그 수정 및 안정화                                   │
│ • Android/iOS 스토어 배포                               │
│ • 관리자 웹 배포                                        │
│ • 사용자 매뉴얼 작성                                    │
```

- 인수인계 및 교육

6.2 산출물 목록

구분	산출물	형식
기획	기능 정의서	PDF/Word
기획	화면 설계서 (와이어프레임)	Figma/PDF
디자인	앱 디자인 원본	Figma
디자인	관리자 웹 디자인 원본	Figma
디자인	디자인 가이드	PDF
개발	앱 소스 코드 (Flutter)	Git Repository
개발	서버 소스 코드 (Node.js)	Git Repository
개발	관리자 웹 소스 코드 (React)	Git Repository
개발	API 명세서	Swagger/PDF
개발	DB 설계서	ERD/PDF
문서	관리자 매뉴얼	PDF
문서	완료 보고서	PDF

---

7. 기술 역량

7.1 관련 경험

프로젝트	내용	기술 스택
스마트홈 제어 앱	IoT 기기 제어 + 실시간 모니터링	Flutter, Node.js
산업용 관제 시스템	CCTV 영상 스트리밍 + 센서 모니터링	React Native, WebRTC
빌딩 에너지 관리 앱	냉난방 제어 + 에너지 사용량 분석	Flutter, AWS
물류 관제 시스템	실시간 위치 추적 + CCTV 연동	React, HLS Streaming

7.2 기술 역량

분야	기술	숙련도
앱 개발	Flutter, React Native	★★★★★
백엔드	Node.js, NestJS, Express	★★★★★
프론트엔드	React, TypeScript	★★★★★

분야	기술	숙련도
DB	PostgreSQL, MySQL, MongoDB	★★★★☆
영상 스트리밍	RTSP, HLS, WebRTC	★★★★☆
IoT 프로토콜	MQTT, TCP/IP, Modbus	★★★★☆
인프라	AWS, Docker, CI/CD	★★★★☆

7.3 레거시 코드 분석 역량

- **Kotlin/Android** 프로젝트 분석 및 마이그레이션 경험 다수
- 기존 프로토콜 역분석 및 재구현 경험
- 점진적 기능 이관 방법론 적용

---

8. 품질 보증

8.1 개발 프로세스

항목	방법
버전 관리	Git (GitHub/GitLab)
코드 리뷰	PR 기반 리뷰 필수
테스트	단위/통합/E2E 테스트
CI/CD	자동 빌드 및 배포 파이프라인
문서화	코드 주석, API 문서 자동화

8.2 품질 체크리스트

- ☐ 기능 테스트 (모든 기능 정상 동작)
- ☐ 성능 테스트 (제어 응답 < 1초, 영상 지연 < 5초)
- ☐ 보안 테스트 (인증, 암호화, 취약점 점검)
- ☐ 호환성 테스트 (Android 8.0+, iOS 13.0+)
- ☐ 사용성 테스트 (UI/UX 검증)

---

9. 협업 및 커뮤니케이션

9.1 협업 도구

용도	도구
프로젝트 관리	Notion / Jira
실시간 소통	Slack / 카카오톡
화상 회의	Zoom / Google Meet

용도	도구
파일 공유	Google Drive
소스 관리	GitHub / GitLab

9.2 미팅 계획

유형	주기	내용
킥오프 미팅	착수 시	프로젝트 범위 및 일정 확정
정기 미팅	주 1회	진행 상황 공유, 이슈 논의
디자인 리뷰	디자인 완료 시	디자인 검토 및 피드백
중간 데모	개발 50% 시점	주요 기능 시연
최종 검수	개발 완료 시	전체 기능 검수

---

10. 견적 (협의 필요)

구분	항목	비고
기획	분석, 기능 정의, 화면 설계	
디자인	앱 + 관리자 웹 UI/UX	
앱 개발	Flutter (Android + iOS)	
백엔드	API 서버 + DB + 스트리밍 서버	
관리자 웹	React 기반 관리자 페이지	
인프라	서버 구축 및 배포	호스팅 비용 별도
PM	프로젝트 관리	
총 예상 비용	**협의 후 확정**	

※ 상세 견적은 기존 앱 소스 및 프로토콜 분석 후 산정 예정

---

11. 질의 사항

프로젝트 착수 전 확인이 필요한 사항입니다:

11.1 기존 앱 관련

1. 기존 Kotlin 앱 소스 코드 공유가 가능한가요?
2. 현재 앱의 주요 이슈나 개선 요청 사항이 있나요?
3. 기존 서버/DB가 있다면 유지할 예정인가요, 새로 구축하나요?

11.2 열선 컨트롤러 관련

4. 열선 제어 프로토콜 정의서를 검토할 수 있을까요?
5. 테스트용 열선 컨트롤러 장비 지원이 가능한가요?
6. 컨트롤러와의 통신 방식은 무엇인가요? (TCP/IP, MQTT 등)

11.3 CCTV 관련

7. CCTV 영상 프로토콜은 RTSP인가요?
8. CCTV 제조사/모델 정보가 있을까요?
9. NVR/DVR 사용 여부 및 연동 방식은?

11.4 일정 관련

10. 희망 완료일 또는 주요 마일스톤이 있나요?
11. 우선순위가 높은 기능이 있다면 무엇인가요?

11.5 기타

12. 회원가입 승인 프로세스의 상세 요구사항이 있나요?
13. Push 알림이 필요한 이벤트 목록이 있나요?
14. 소셜 로그인 (카카오/네이버/구글 등) 지원이 필요한가요?

---

12. 연락처

항목	내용
업체명	[업체명]
담당자	[담당자명]
직책	[직책]
연락처	[전화번호]
이메일	[이메일]
주소	[주소]

본 지원서는 프로젝트 요구사항을 기반으로 작성되었으며, 상세 협의를 통해 조정될 수 있습니다.

작성일: 2025 년 12 월 18 일