

「2025 IA x AI 해커톤」

개발 완료 보고서

팀 명 : SIREN

프로젝트명 : SENSE(Safety ENgine for Situational Escape)

* 유의사항

1. 본 보고서의 내용은 최대 2 page이내로 작성 (본 표지 제외)
2. 보고서의 설명을 보충하기 위해 필요한 사진 또는 그래프 첨부 가능
3. 제출 서류는 일체 반환을 하지 않음
4. 제출 파일명 작성 요령
 - 파일명: [2025 IA x AI 해커톤]_팀명
5. 서체: 맑은고딕, 크기: 12p, 줄간격: 160%
6. 제출처: 깃허브에 업로드

「2025 IA x AI 해커톤」

프로젝트명	SENSE(Safety ENgine for Situational Escape)
프로젝트 목표	<ul style="list-style-type: none"> 단계형 행동 지침 제시: “즉시 해야 할 일 → 다음 단계 → 주의사항”을 정리해 제공합니다. 개인화 판단: GPS와 공공 데이터를 융합해 사용자의 실제 위치·시간·지형 맥락에 맞춘 대응을 도출합니다. 대피 후보 추천: 재난이 흉수/폭우라면 가까운 대피소나 고지대 등을 거리·예상 이동 시간과 함께 제안합니다. 대화형 Q&A: 재난 상황에서 떠오르는 의문점을 경량 LLM이 실시간으로 답변해 줍니다. 오프라인 지원: 비행기 모드에서도 핵심 기능이 작동하도록 규칙과 FAQ를 로컬에 캐시합니다. 타겟: 대상 재난은 흉수/폭우, 지진(여진·쓰나미), 전시(민방위/공습·CBRN) 등입니다.
개발 환경	<p>[백엔드]</p> <ul style="list-style-type: none"> Python 3.13+ / FastAPI 로컬: Ollama (예: gemma3:4b, qwen3:4b) 클라우드: Google Gemini <p>Retrieval (Hybrid RAG), Graph RAG, Vector RAG, db</p> <ul style="list-style-type: none"> Neo4j (공간/관계형 지식 그래프: 예, 대피소 위치, 시설 속성) Chroma (로컬 벡터 DB: 재난 대응 매뉴얼, 행동 수칙 등) <p>[모바일]</p> <p>Flutter SDK 3.9.2, Riverpod, go_router</p> <ul style="list-style-type: none"> Android (SDK 21 이상) iOS는 설계 대상에는 포함되어 있으나, 현재 핵심 기능(재난 문자 자동 수신)은 Android 우선 구현 <p>네이티브 모듈 (Android / Kotlin)</p> <ul style="list-style-type: none"> BroadcastReceiver를 통한 SMS 수신 Flutter와의 통신은 EventChannel 등 플랫폼 채널 기반
구현 기능	붙임 1에 작성
코드 주요 설명	SmsReceiver.kt: 안드로이드에서 재난 문자(SMS)를 감지하고 Flutter로

	<p>전달</p> <p>DisasterAlertNotifier / room_list_provider.dart: 전달된 재난 문자를 파싱해 상황 룸(대화방)을 생성·관리</p> <p>room_messages_provider.dart + chat_tab.dart: 룸별 메시지(사용자 질문, 서버 답변, 행동 지침·대피소 안내)를 채팅 UI로 보여줌</p> <p>api.py: 모바일에서 온 /chat 요청(위치, 층수, 질문 등)을 받고 백엔드 파이프라인을 호출</p> <p>orchestrator.py: LangGraph로 Profile → Plan → Analyze → Advise 순서대로 에이전트를 실행</p> <p>rag.py + db.py: Neo4j(대피소·위치 그래프)와 Chroma(재난 매뉴얼 임베딩)에서 근거를 조회해 최종 행동 지침을 생성</p>
개발 내용	붙임 1에 작성
시연 영상	-
실행 파일 (선택)	-
기타 (선택)	<p>김종석: 모바일 클라이언트, 모바일 파트는 Flutter UI/상태관리, 안드로이드 네이티브(SMS 수신, 센서 수집), 위치/층수 전송 로직을 담당</p> <p>임시은: 백엔드, FastAPI 기반 API, LangGraph 파이프라인(Profile→Plan→Analyze→Advise), Hybrid RAG(Neo4j+Chroma) 설계를 담당</p> <p>윤석현, 권덕현, 김현서: 데이터·온톨로지 파트, 재난 행동 규칙, 데이터 수집, 데이터 전처리</p> <p>협업 방식: 깃허브, 디스코드를 활용, 해커톤 준비 기간 중 7회 이상 만나 논문, 자료 등을 통해 개발에 필요한 내용 학습</p>

불임1

구현기능

구현 기능

Flutter 기반 Android 앱은 재난 문자를 자동 감지·파싱해 상황별 대화방(상황 룸)을 만들고, 사용자 위치(GPS), 기압 기반 층수, 사용자 질문 등을 서버에 전송하며, 채팅 UI에서 즉시 행동 지침·다음 단계·주의사항·대피소 정보를 보여주고 119/가족 연락 등 즉시 실행 버튼을 제공.

백엔드는 FastAPI로 /chat 요청을 받아 사용자의 위치·층수·재난 유형·긴급도를 해석하고, LangGraph 파이프라인(Profile → Plan → Analyze → Advise)을 통해 상황 분석 → 필요한 정보 계획 → Hybrid RAG 조회(Neo4j의 구조화된 대피소·위치 관계 + Chroma의 재난 대응 매뉴얼) → 행동 가이드를 생성

온톨로지는 “즉시 해야 할 행동 / 다음 판단을 위한 맥락 / 주의·금지 사항”을 표준화된 블록으로 정의하고, 예를 들어 ‘지진 시 엘리베이터 사용 금지’ 같은 안전 규칙과 이동 가능 조건을 강제로 포함시켜 개인 상황에 맞는 조언만 선택하도록 제어

개발 내용

모바일 쪽에서는 Flutter 기반 구조를 잡고, 안드로이드 네이티브(Kotlin)로 재난 문자 수신 모듈을 붙여서 자동 감지 → 파싱 → 앱 내부 상태 업데이트까지 흐름을 완성

재난이 감지되면 자동으로 “상황 룸”이 생성되고, 위치(GPS), 층수(기압 센서 추정), 사용자 질문 등이 서버로 전송

UI는 단순 채팅 형태가 아니라 즉시 행동 / 다음 행동 / 주의사항 / 대피소 / 비상 통화(119, 가족)까지 한 화면에서 바로 실행 가능하게 설계·구현

백엔드는 FastAPI로 /chat API를 만들고, 들어온 상황 정보를 LangGraph 파이프라인 (Profile → Plan → Analyze → Advise)로 처리해 행동 가이드를 생성

Neo4j(대피소 등 위치/관계 정보)와 Chroma(재난 매뉴얼 임베딩)를 결합한 Hybrid RAG을 붙여 실제 근거 기반으로 답하도록 했고, 응답을 “즉시 행동 / 다음 행동 / 주의사항” 블록으로 표준화했다.

전체적으로 “모든 사람에게 같은 재난문자”가 아니라, 현재 조건(어디에 있는지, 몇 층인지, 움직일 수 있는지)에 맞춘 개인화된 대응 안내를 주는 구조를 구현