

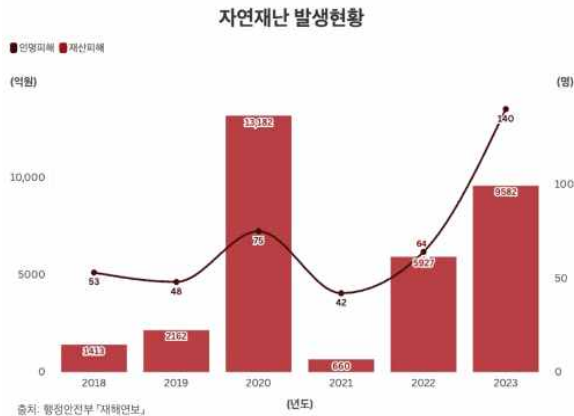
SW공동체 중간보고서

팀명	404Found
프로젝트명	ResQ

1. 프로젝트 소개

가. 개발 배경 및 동기

□ 개발 배경



최근 기후변화로 인해 빈번해진 자연 재난과 미세먼지, 코로나 19 등 다양한 유형의 재난이 일상화됨에 따라 수많은 피해가 발생하고 있습니다. 실제로 2014년부터 2023년까지 연 평균 인명피해(사망·실종)는 43명, 재산피해는 3,980억원에 달한 것으로 나타났습니다. 매년 재난이 반복되며 국민들에게 심각한 피해를 주고 있습니다

(출처: 통계청 지표누리 e-나라지표 - 자연재난 발생)

① 부족한 실시간 대응 정보 및 행동 지침

국회입법조사처 NARS 현안분석 제297호
「재난문자방송시스템운영의 쟁점과 향후 과제」

2023년 5월 31일 새벽 6:41분 서울지역에 대피를 준비하라는 위급재난문자가 발송되었고, ...이번 재난문자는 오발령 관련 논란도 있지만, 왜 대피해야 하는지, 어디로 대피해야 하는지에 대한 이유 없이 대피만 지시함으로써 국민들의 불안과 혼란을 가중시켰다. -1p

지난 여름 발생한 중부지역 집중호우시 피해를 겪었던 지역의 지방자치단체가 발송한 재난문자의 발송시기와 내용을 살펴보면, 하천범람이나 지하철거 침수위험 등에 있어서 지역주민들이 해당 재난상황에 대비할 수 있도록 미리 알려진 것이 아니라 이미 많은 주민들이 피해를 겪은 후에서야 뒤늦게 발송한 경우가 많았다. 특히 해당 지역의 침수상황에 대한 언급 없이 단순히 집중호우에 대비하라는 재난문자만 발송하기도 하였다. -12p

재난문자는 긴급히 전달해야 할 중요한 재난에 대해 행정안전부를 비롯한 여러 정부기관의 결정에 의해 다수의 국민들에게 전송되는 형태이므로, 국민이 직접적으로 그 수행내용을 확인할 수 있는 정부의 정책이라 볼 수 있다. 이번 재난문자의 경우에도 방송 지연, 발령 원인, 대피 장소 등 핵심 정보 부재 등 정부 대응의 미흡한 부분을 여실히 보여주고 있다. -13p

「재난문자방송 기준 및 운영 규정」 개정 사항(23.12.08.)

명칭	표준문안
태풍경보	오늘 00시 00지역 태풍경보 발령, 실내에 머무르세요. 해안가, 급경사지 등 위험지역에는 가지 마시고, 대피 권고를 받으면 즉시 대피하세요. [행정안전부]
호우경보	오늘 00시 00지역 호우경보 발령, 하천 주변, 계곡, 급경사지, 농수로 등 위험지역에는 가지 마시고, 대피 권고를 받으면 즉시 대피하세요. [행정안전부]
홍수경보	오늘 00:00 00강 00시(00교)지점 홍수경보 발령, 저지대 침수 및 하천 범람이 우려되니 해당 지역 주민은 안전한 장소로 대피하세요. [00홍수통제소]

출처: 「재난문자방송 기준 및 운영 규정」, 휴대전화 재난문자방송 표준문안(제10조 관련)(23.12.08.)

- 개인의 위치를 고려한 구체적인 피난 경로
- 대피소의 위치나 진입 방법
- 피해 지역의 실시간 상황 등
정보의 부재로 시민이 즉각적으로 행동으로 옮기기 어려움

재난 상황에서 시민들의 신속한 대처를 어렵게 만드는 주요 원인 중 하나는 실시간 정보의 부족과 전달 방식의 비직관성입니다. 정부나 지자체는 국민재난안전포털, 긴급재난문자 등을 통해 정보를 제공하

고 있지만, 실제 재난에 직면한 시민들에게는 해당 정보가 충분하지 않거나, 활용도가 낮은 경우가 많습니다.

특히 피해 지역의 위치, 규모, 이동 경로 등은 실시간으로 파악하기 어려워 시민들이 위기 상황을 정확히 인식하지 못하고, 이로 인해 적절한 판단과 대처를 하지 못하여 피해가 확대되는 경우가 지속적으로 발생하고 있습니다.

또한, 안전 정보와 행동 지침의 부족도 심각한 문제입니다. 재난 발생 시 개인의 위치를 기반으로 한 대피소 안내가 부족하고, 구체적인 행동 요령 역시 접근성이 떨어집니다. 예를 들어, 국민재난안전포털이나 문자 알림을 통해 대피 정보를 제공하더라도, 시민들은 현재 위치에서 어디로 어떻게 이동해야 하는지, 주변 피해 상황은 어떤지 명확하게 알기 어렵습니다. 이로 인해 많은 시민들이 재난 상황에서 혼란을 겪거나, 잘못된 판단으로 인해 추가 피해를 입는 사례가 반복되고 있습니다.

② 재난 정보 공유 플랫폼의 부재

재난 상황에서 지역 사회의 협력은 매우 중요한 요소입니다.

그러나 현재 시민들이 재난 상황에서 사용할 수 있는 정보 공유 수단은 SNS나 일반 커뮤니티 앱에 한정되어 있습니다. 즉, 실시간으로 재난 상황을 공유하고 협력할 수 있는 재난 중점적인 플랫폼은 부재한 상태입니다. 이로 인해 재난 관련 정보가 분산되고, 체계적인 정보 공유와 시민 주도 대응이 이루어지지 못하고 있는 실정입니다.

이러한 문제들은 재난 대응의 골든 타임을 놓치게 만들며, 시민의 생명과 안전을 위협하고 있습니다.

나. 개발목표 및 주요내용

□ 개발 목표

- 위치 기반의 실시간 재난 정보를 제공하고, 사용자 인근의 대피소 위치와 정보를 안내하여 신속한 대피 지원
- 지역 사회의 정보 공유와 협력을 강화하여 재난 대응의 효율성과 정보 신뢰성을 높이는 플랫폼 구축

□ 주요 내용

1) 현재 지역의 재난정보와 주변 대피소 확인

사용자의 위치 정보를 바탕으로, 현재 지역에서 발생한 재난 정보 목록을 표시하고, 각 재난에 대한 상세 정보와 대처 방법을 제공합니다. 또한 현재 위치 근처에 있는 대피소를 지도에 시각적으로 표시하며, 대피소로부터의 거리도 함께 안내합니다. 각 대피소의 상세정보 및 길찾기 안내 버튼을 통해 외부 지도 앱과 연동되어 빠르게 길을 찾을 수 있도록 지원합니다.

2) 전체 재난 상황 확인

전국에서 발생 중인 재난 상황을 실시간으로 요약해 한눈에 볼 수 있도록 지도로 제공하고, 지역별로 발생한 재난의 종류, 건수 등을 리스트 형태로 확인할 수 있도록 합니다. 또한 재난과 관련된 최신 뉴스 목록을 제공하고 클릭 시 상세 페이지로 이동하여 세부 내용을 확인할 수 있습니다.

3) 재난 정보 공유 커뮤니티

사용자는 재난과 관련된 경험, 정보, 주의 사항 등을 자유롭게 게시할 수 있으며, 원하는 지역의 게시글만 모아보거나 키워드로 글을 검색하여 필요한 정보에 빠르게 접근합니다.

또한 댓글, 좋아요 기능으로 사용자들이 서로 소통할 수 있으며, 실시간 재난 대응에 도움이 되는 정보 공

유가 활성화됩니다.

다. 세부내용

○ 요구사항 분석

<기능 요구사항>

1. 내 지역 근처 재난 상황 대비

기능	상세 정보
내 주변 대피소 확인	사용자의 지역 정보를 바탕으로 주변 대피소를 지도에 표시한다.
대피소 상세 정보	대피소의 상세 주소, 이름 등 상세 정보를 사용자에게 보여준다.
대피소 길찾기	선택한 대피소로 향하는 길을 지도에 표시해준다.
내 주변 재난 목록	자신의 지역 근처의 재난 정보를 재난별로 보여준다.
내 주변 재난 상세 정보	주변에 일어난 재난 중 선택한 재난의 상세 정보를 보여준다.
특정 재난 대처 방법	내가 선택한 재난의 대처 방법을 표시하고 상세 동영상이나 글의 링크를 제공한다.
내 위치 표시	지도에 나의 위치를 표시한다.
나와 대피소의 거리 표시	지도에 나타난 대피소와 나의 위치 사이의 거리를 표시한다.
위험 상태메시지	지금 자신이 위험한 상황에 있는지 상태메시지가 표시된다.
재난 알림	재난 상황에 대한 알림을 제공한다.

2. 전체 재난 상황 확인

기능	상세 정보
전체 재난 정보	지역에 상관없이 현재 대한민국에 일어나고 있는 재난 상황 제공한다.
재난 관련 뉴스 제공	재난과 관련된 뉴스 중 몇가지를 화면에 제공한다.
재난 뉴스 상세 페이지 제공	재난뉴스의 상세 내용을 제공한다.
전체 지도 제공	전체 지도에 현재 재난 상황이 일어난 곳을 시각적으로 색이나 아이콘을 통해 확인할 수 있도록 돕는다.
재난 별 대처 방법	재난 별로 대처 방법 페이지를 따로 만들어 이를 모아서 볼 수 있도록 한다.
지역별 재난 건수 제공	지역 별로 어떤 재난이 몇건, 어떤 규모로 일어났는지 제공한다.

3. 재난 커뮤니티

기능	상세 정보
----	-------

재난 관련 글 작성	재난과 관련된 글을 작성할 수 있다.
지역 별 필터링	자신이 원하는 지역에서 올라온 글만 모아볼 수 있다
게시글 검색 기능	자신이 검색한 키워드가 들어간 글을 모아볼 수 있다.
인기글 목록 제공	좋아요 수가 많은 글들을 인기글 페이지에서 모아서 제공한다.
게시글 상세 페이지	한 게시글의 상세 글을 볼 수 있다.
게시글 좋아요 기능	특정 게시글에 좋아요를 표시할 수 있다.
댓글 기능	특정 게시글에 댓글을 남길 수 있다.
댓글 좋아요 기능	특정 댓글에 좋아요를 남길 수 있다.
글 작성 시 지역 설정 기능	게시글을 작성할 때 자신의 지역을 설정하여 글을 작성할 수 있다.
게시글 수정 및 삭제	자신이 작성한 게시글을 수정 및 삭제할 수 있다.
댓글 수정 및 삭제	자신이 작성한 댓글을 수정 및 삭제할 수 있다.

4. 로그인 및 회원가입

기능	상세 정보
회원가입	닉네임, 이메일, 비밀번호 등을 입력하여 회원가입을 할 수 있다.
로그인 기능	닉네임, 비밀번호 등을 통해 로그인을 할 수 있다.

5. 내 계정 관리

기능	상세 정보
로그아웃	로그인 된 상태에서 로그아웃을 할 수 있다.
회원 탈퇴	로그인 된 계정을 삭제할 수 있다.
비밀번호 변경	자신의 계정 비밀번호를 변경할 수 있다.
프로필 사진 변경	자신의 프로필 사진을 변경할 수 있다.
닉네임 변경	자신의 닉네임을 변경할 수 있다.
작성글 확인	자신이 작성한 글에 조회할 수 있다.
아이디와 이메일 확인	마이 페이지에서 자신의 계정 이메일과 아이디를 확인할 수 있다.
작성 댓글 확인	자신이 작성한 댓글에 조회할 수 있다.

<사용자 요구사항>

요구사항	상세 설명
빠른 재난 정보 제공	재난 상황이 일어났을 때 빠른 정보의 제공이 중요하다.
재난 상황 시 대피 유용성	재난 상황 대피에 도움이 될 수 있어야 한다.
직관적인 ui	지도에 간단하면서도 직관적으로 상황을 파악할 수 있는 ui 디자인을 제공해야한다
지역 별 정보	자신의 근처 재난 상황을 확인하기 쉽도록 지역별로 세분화된 정보를 제공해야한다.

다. 사회적 가치 도입 계획

1. 지역 사회 재난 대응력 강화

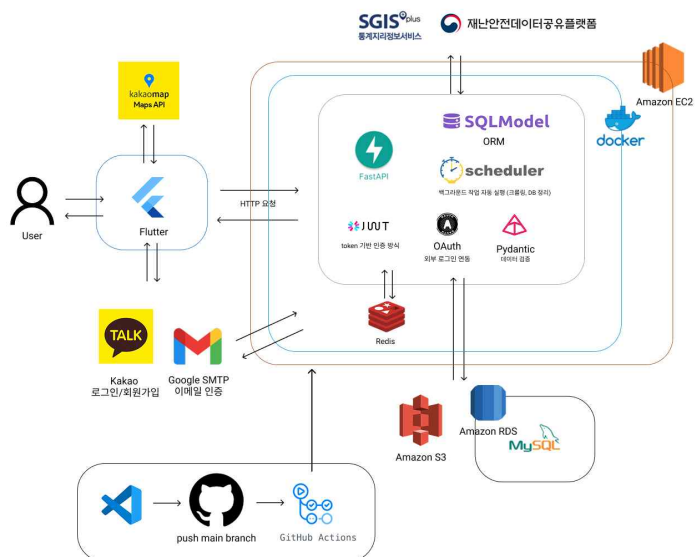
본 앱은 사용자의 위치를 기반으로 실시간 재난 정보와 대피소 안내를 제공함으로써, 지역 주민이 신속하고 효율적으로 대응할 수 있는 환경을 조성합니다. 특히 부산대학교처럼 산불·산사태에 취약한 녹지 밀집 지역이나 인구 밀도가 높은 생활권에서 유용하게 활용될 수 있으며, 재난 발생 시 혼란을 최소화하고 생명과 재산 피해를 줄이는 데 기여합니다.

2. 재난 정보 공유 커뮤니티를 통한 시민 협력

앱 내 커뮤니티 기능을 통해 사용자가 재난 상황을 직접 공유하고 실시간 피드백을 주고받으며, 기반의 협력 체계와 공동 대응 문화 형성에 기여할 수 있습니다.

2. 상세 설계

가. 시스템 구성도



나. 플로우차트

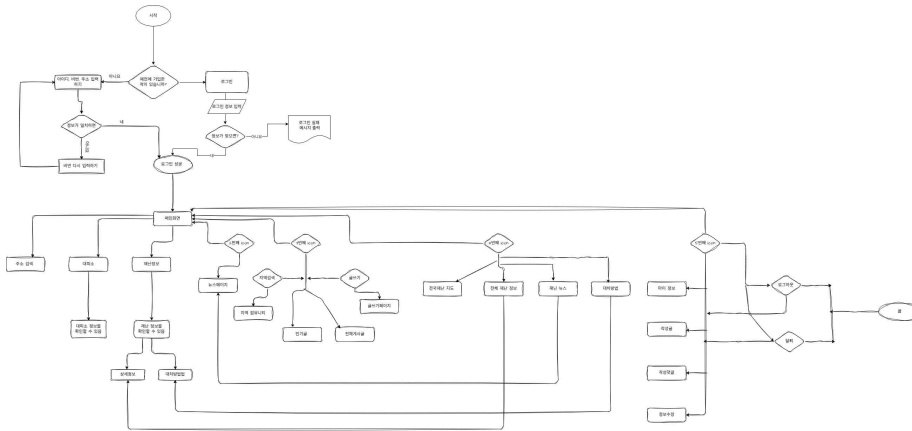


그림 5 [User flowchart](#)

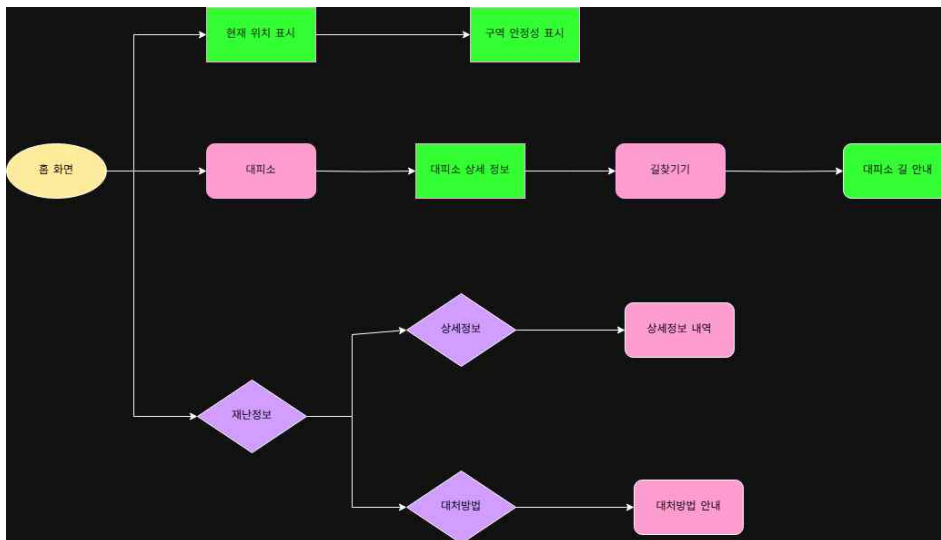


그림 6 [Home flow chart](#)

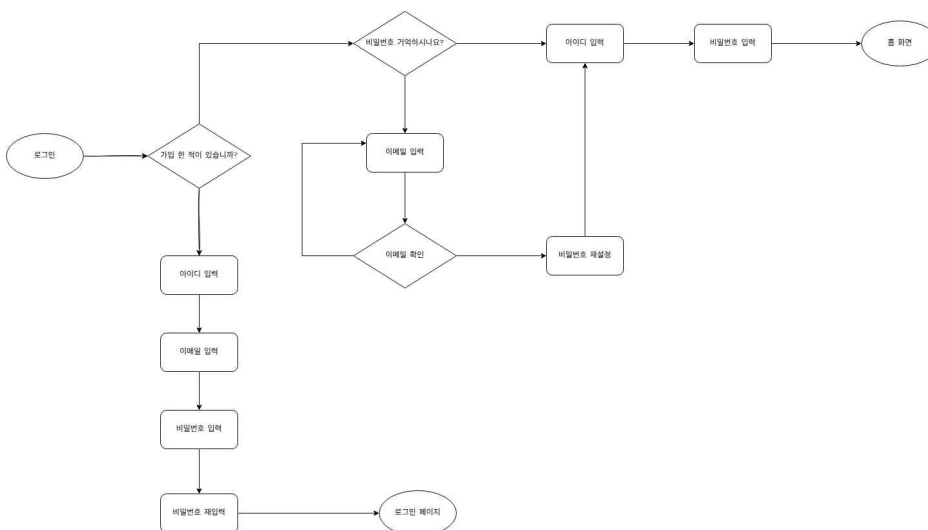


그림 7 [login flow chart](#)

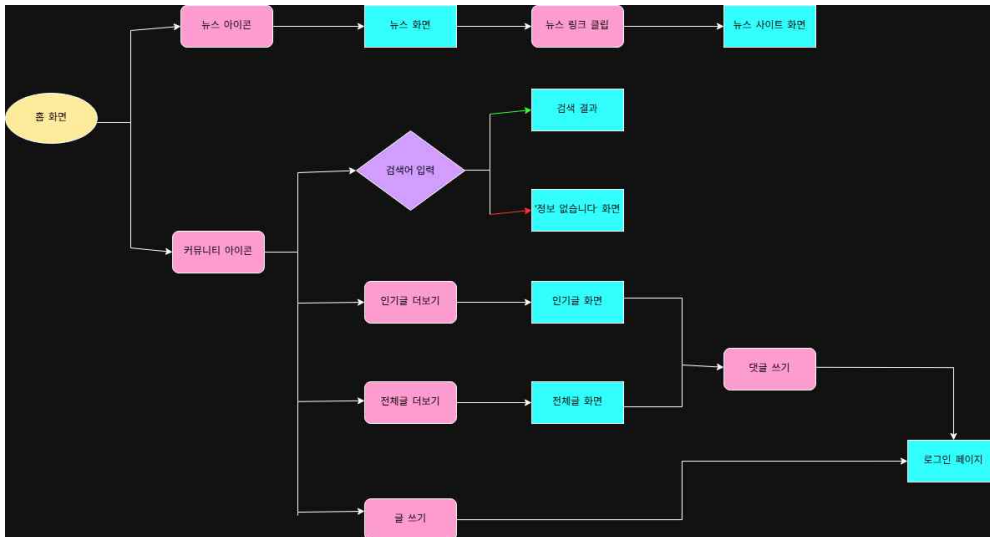


그림 8 news & 커뮤니티 flow chart

다. 테스트 플로우차트

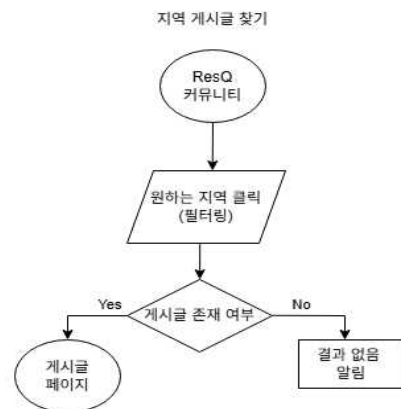


그림 9 게시물 검색 테스트 플로우 차트

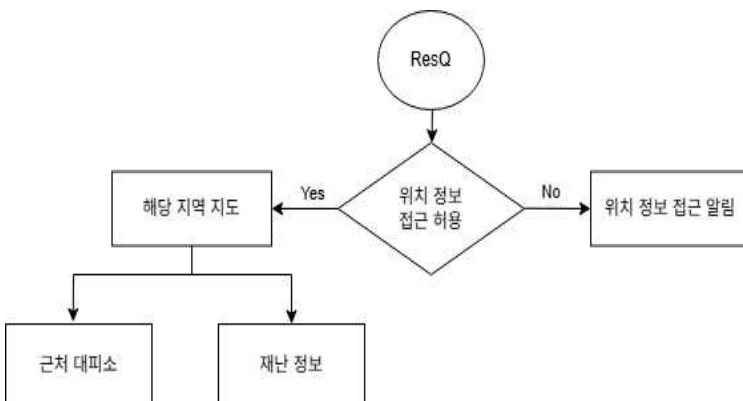


그림 10 지도 테스트 플로우 차트

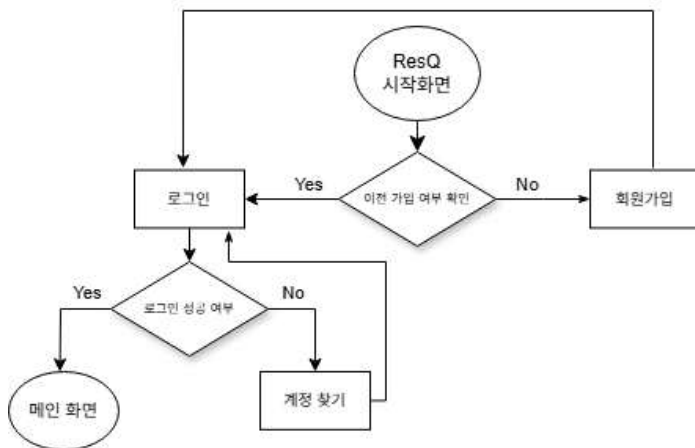


그림 11 로그인 및 회원가입 테스트 플로우 차트

라. I.A



그림 12 I.A

마. 시스템 플로우

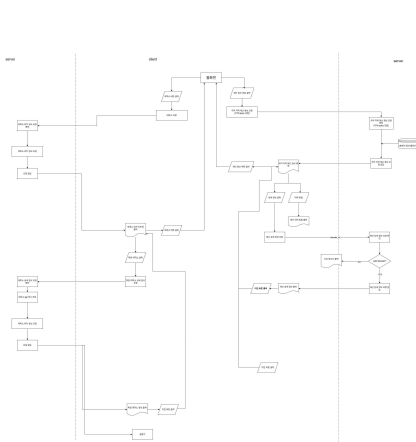


그림 13 [홈화면 시스템 플로우](#)

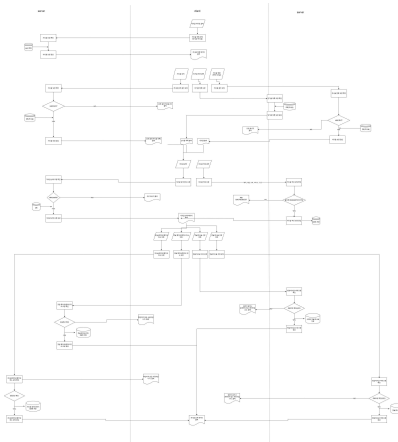


그림 14 [post&comment 시스템 플로우](#)

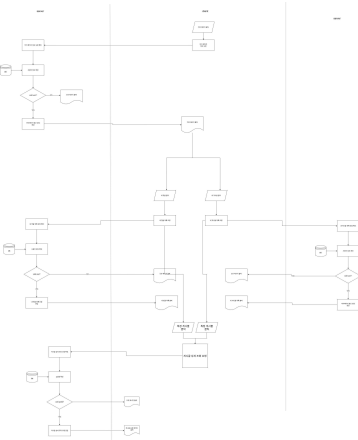


그림 15 [My 시스템 플로우](#)

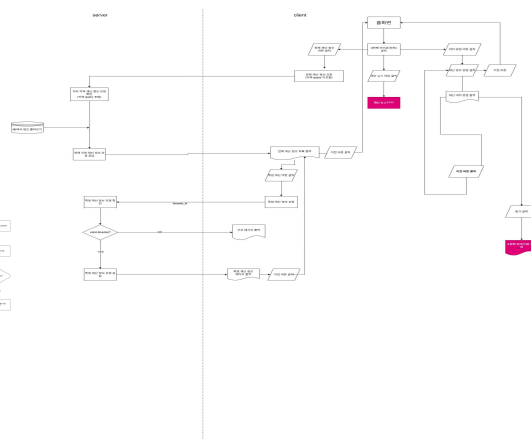


그림 16 [전체 재난 정보 시스템 플로우](#)

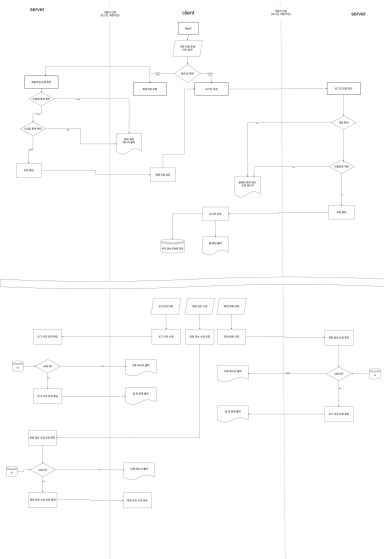


그림 17 [User 시스템 플로우](#)

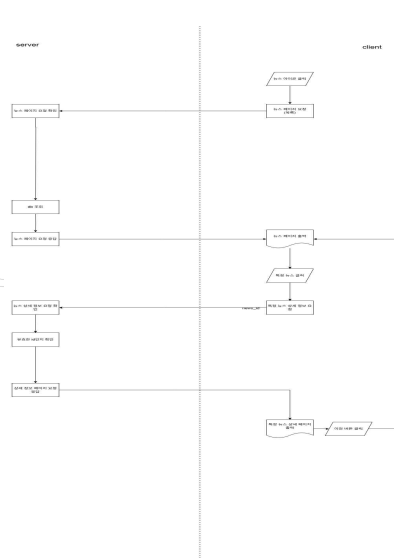
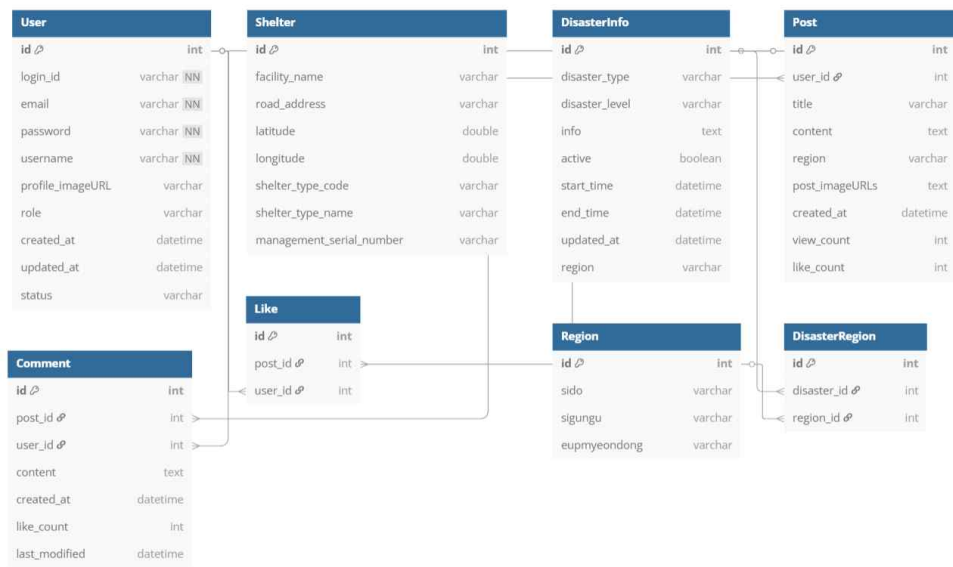


그림 18 [News 시스템 플로우](#)

바. ERD



사. 사용 기술

범주	이름	버전
Frontend	Flutter	3.29.3
Frontend	Dart	3.7.2
Backend	FastAPI	0.115.12
Backend	Python	3.11.9
ORM	SQLModel	0.0.24
Auth	PyJWT	2.10.1
Database	MySQL	8.0
Infra	AWS EC2	Ubuntu 22.04
Infra	Redis	6.1.0
CI/CD	GitHub Actions	-
Container	Docker	27.5.1
Storage	Amazon S3	-

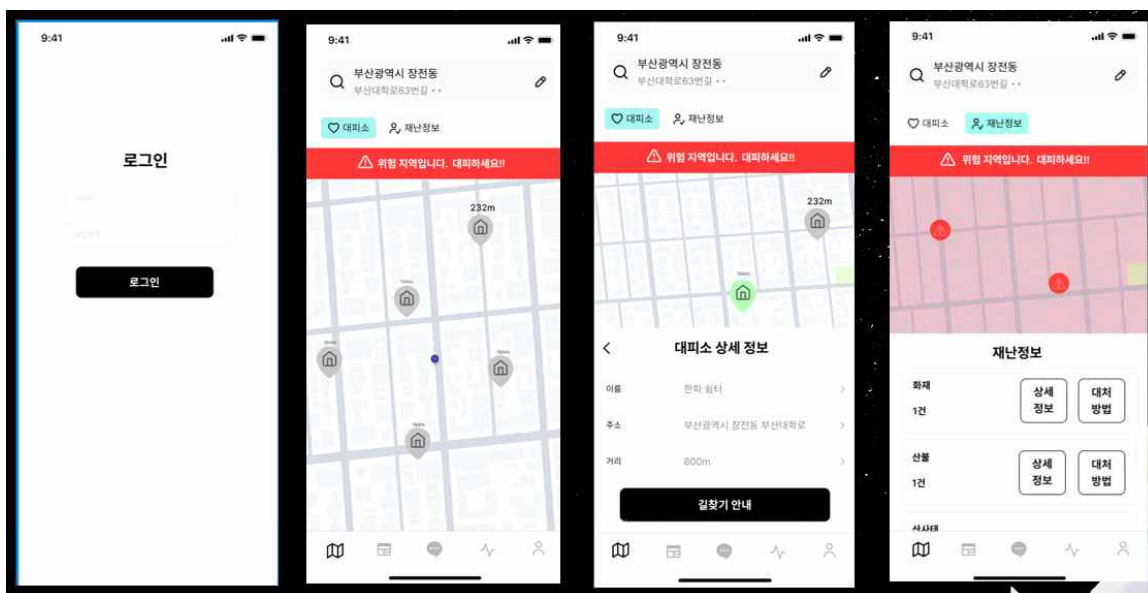
3. 개발현황

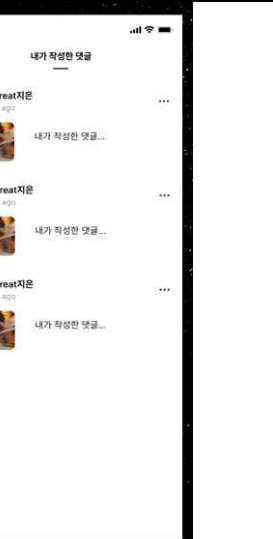
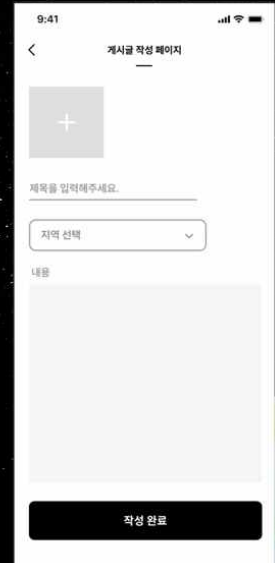
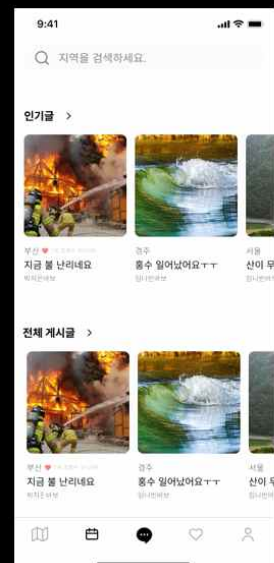
활동 주차	날짜	주요 업무	회의 내용
1주차	3/28	기획, 아이디어	- 계획서, 신청서 작성 - 팀명, 아이디어 정하기 - 노선 정리 - 규칙 정하기
2주차	4/5	ERD 설계, API 명세서, 코드 컨벤션	- 화면 구성 회의 - 디자인/구성 피드백
3~4주차	중간고사	시험공부.....	X
5주차	4/27	API 명세서 피드백	- sw 창의융합 해커톤 신청 회의 - 플로우 차트 - API 명세서, ERD 피드백
6주차	5/4	집중 개발 & 피드백	- 개발한 내용 직군 별로 발표
7주차	5/11	집중 개발 & 피드백	- 개발한 내용 직군 별로 발표
8주차	5/18	집중 개발 & 피드백	- 개발한 내용 직군 별로 발표
9주차	5/25	집중 개발 & 피드백	- 개발한 내용 직군 별로 발표

현재 활동 주차 별로 진행한 내용입니다.

1주차에 개발 배경 및 동기, 개발 목표 및 주요내용, 세부 내용, 사회적 가치 도입 계획에 대한 기획 회의를 진행하였습니다. 이후 Figma에서 디자인을 진행하며 2주차에 피드백을 하였고 초기 디자인을 완성하였습니다. 3~4주차에는 백엔드에서 DB 설계 및 API 명세서 작성을 진행하였으며 5주차 회의에서 피드백을 하고 시스템 구성도, 유저 플로우 차트, 테스트 플로우 차트, 시스템 플로우 차트, IA를 작성하며 프로젝트에 대한 이해를 높였습니다. 6주차부터는 프론트엔드, 백엔드가 작업을 진행하며 실시간으로 개발 완료 및 상황 공유를 진행하였습니다.

가. 화면 디자인





나. API 명세서 작성

index	Aa req/resp	기능	HTTP meth...	API Path	토론	백엔드 개발 현황	프론트 개발 현황
User		비밀번호 재설정 (비로그인 상태)	POST	/reset-password	O	Done	
User		비밀번호 재설정 토큰 유효성 확인	POST	/verify-password-reset-token	O	Done	
User		비밀번호 재설정 이메일 보내기	POST	/request-password-reset?email=0	X	Done	Done
User		이메일 인증 토큰 유효성 확인	POST	/verify-email-token	O	Done	Done
User		access token 재발급 (자동 로그인)	POST	/refresh	X	Done	
User		이메일 인증 요청 보내기	POST	/request-verification-email?email=0	X	Done	Done
User		로그인 (로그아웃은 클라이언트 처리)	POST	/users/signin	X	Done	Done
	(보류)	신고하기					
User		회원가입	POST	/users/signup	X	Done	Done
User		카카오 회원가입/로그인					
User		마이페이지 정보 조회	GET	/mypage	O	Done	Done
User		회원 탈퇴	PATCH	/users/delete	O	Done	
User		회원 정보 수정	PATCH	/users/update	O	Done	
Shelter		내 주변 대피소 정보 조회	GET	/shelters?latitude={latitude}&longitude={longitude}&limit={num}	X	Done	
Shelter		특정 대피소 정보 조회	GET	/shelters/{shelter_id}	X	Done	
DisasterInfo		지역 재난 목록 얻기(헬터링)	GET	/disasters?ido=부산광역시&sigungu=금정구&upmyeon=경전동	X	In Progress	
DisasterInfo		특정 재난 정보 얻기	GET	/disasters/{disaster_id}	X	In Progress	
DisasterInfo		재난 해제 (테스트용)	PATCH	/disasters/{disaster_id}/deactivate	X	In Progress	
Like		게시글 좋아요 누르기	POST	/posts/{post_id}/like	O		
Like		게시글 좋아요 취소	DELETE	/posts/{post_id}/like	O		
Like		댓글 좋아요 누르기	POST	/comments/{comment_id}/like	O		
Like		댓글 좋아요 취소	DELETE	/comments/{comment_id}/like	O		
Post		내가 쓴 글 목록 조회	GET	/posts/me	O		
Post		게시글 목록 조회	GET	/posts	X		
Post		게시글 검색	GET	/posts?term={term} (optional)	X		
Post		게시글 필터링	GET	/posts?region=부산광역시	X		
Post		게시글 정렬	GET	/posts?sort={option} (optional)	X		
Post		게시글 상세 조회	GET	/posts/{post_id}			
Post		게시글 작성	POST	/posts	O		
Post		게시글 수정	PATCH	/posts/{post_id}	O		
Post		게시글 삭제	DELETE	/posts/{post_id}	O		
Comment		댓글 목록 조회	GET	/posts/{post_id}/comments	X		
Comment		댓글 작성하기	POST	/posts/{post_id}/comments	O		
Comment		댓글 수정하기	PATCH	/comments/{comment_id}	O		
Comment		댓글 삭제하기	DELETE	/comments/{comment_id}	O		
Comment		내가 쓴 댓글 조회	GET	/comments/me	O		
Region	(삭제 예정)	지역 목록 조회	GET	/regions	X		
Region	(삭제 예정)	상세 지역 목록 조회	GET	/regions/{region_id}	X		
NEWS	(삭제 예정)	뉴스 전체 조회	GET	/news	X		

다. 개발 진행 상황 및 프로젝트 구조

1) 백엔드 프로젝트 구조

```

project-root/
├── README.md
├── .env
├── .gitignore
├── main.py
├── requirements.txt
├── Dockerfile
├── docker-compose.yml
├── .dockerignore
├── .github/
│   └── workflows/
│       └── deploy.yml
├── data/
│   └── RegionCategory.csv

```

```

├─ app/
│  ├─ core/
│  │  └─ redis.py
│  ├─ db/
│  │  ├─ session.py
│  │  └─ init_db.py
│  ├─ handlers/
│  │  ├─ email_handler.py
│  │  ├─ user_handler.py
│  │  ├─ disaster_handler.py
│  │  └─ shelter_handler.py
│  ├─ models/
│  │  ├─ region_model.py
│  │  ├─ user_model.py
│  │  ├─ disaster_model.py
│  │  └─ shelter_models.py
│  ├─ schemas/
│  │  ├─ common_schema.py
│  │  ├─ user_schema.py
│  │  └─ shelter_schema.py
│  ├─ services/
│  │  ├─ region_service.py
│  │  ├─ email_service.py
│  │  ├─ user_service.py
│  │  ├─ disaster_service.py
│  │  └─ shelter_service.py
│  └─ utils/
│     ├─ jwt_util.py
│     └─ redis_util.py

```

(현재 개발된 파일만 정리한 구조입니다. 아직 미완성이므로 변경이나 추가될 수 있습니다.)

현재 로그인, 회원가입 및 대피소, 재난 정보 API까지 개발 완료 및 테스트 완료되었으며 게시글, 댓글 관련하여 개발을 진행하고 있으며 재난정보와 지역명 연결 중에 있습니다. 현재 API의 2/3 정도 개발이 완료되어 남은 개발 진행 후 배포 및 코드 리팩토링을 진행할 예정입니다.

2) 프론트엔드 프로젝트 구조

```

project-root/
├─ lib/
│  ├─ main.dart
│  ├─ routes.dart
│  └─ pages/
│     ├─ community_main_page.dart
│     ├─ news_page.dart
│     ├─ login_page.dart
│     └─ signup_page.dart
├─ pubspec.yaml
├─ analysis_options.yaml
├─ .gitignore
└─ README.md

```

(현재 개발된 파일만 정리한 구조입니다. 아직 미완성이므로 변경이나 추가될 수 있습니다.)

현재 비밀번호 재설정 이메일, 이메일 인증 요청 UI 개발 및 테스트를 완료하였으며 이메일 인증 토큰 유효성 확인, 회원가입, 마이페이지 정보 조회는 UI 개발 완료 후 테스트를 진행하고 있습니다.

지도 API는 UI 구현 완료 후 API 연동 진행 중 및 해당 지역의 안전 여부 시각적 표시 기능 개발 중입니다. 향후 내 주변 대피소 정보 조회, 특정 대피소 정보 조회, 커뮤니티 페이지 개발을 진행할 예정입니다.

현재까지 모든 프론트엔드 UI 화면 개발은 착수 되었으며, 앞으로의 남은 작업은 각 기능과 화면에 연결될 API 연동 처리와 이에 필요한 백엔드의 추가 개발 및 수정 작업이 중심이 될 예정입니다.

라. 배포 현황

The image displays two screenshots related to deployment. The top screenshot shows the AWS Management Console for an EC2 instance named 'ResQ'. The instance is in a 'running' state, using the 't2.micro' instance type. The 'Details' tab is selected, showing various configuration details like the instance ID, IP addresses, and DNS settings. The bottom screenshot shows the GitHub Actions interface, specifically the 'Actions' tab. It lists two workflow runs: 'v1.2.5: CORS 설정 추가' (v1.2.5: Add CORS settings) and '배포 설정' (Deployment settings), both of which are completed successfully. The 'Deploy to EC2' workflow is highlighted.

현재 EC2와 Docker를 활용하여 1차 배포를 완료하였으며 Github Actions로 main branch에 push하면 배포가 이루어지도록 CI/CD 환경을 구축하였습니다. 배포된 서버의 /docs 경로를 통해 FastAPI의 Swagger UI에 접근 가능하며, 현재 프론트엔드 측에서는 이를 기반으로 API 연결 및 테스트를 진행하고 있습니다. 향후 재난정보와 지역 연결 관련 개발 후 2차 배포 예정이며 게시글, 댓글 작업 후 3차 배포 및 최종 배포를 계획하고 있습니다.

4. 설치 및 사용방법

가. 프론트엔드 설치 및 사용방법

Flutter 설치

- 1) Flutter SDK 다운로드
- 2) 적당한 경로에 압축 해제 (ex) C:\src\flutter)
- 3) 시스템 환경변수 > Path > C:\src\flutter\bin 추가
- 4) cmd에 flutter doctor 명령어 입력
- 5) Android Studio 설치 및 Android SDK 설정

패키지 설치

```
flutter pub get
```

에뮬레이터 실행

```
flutter devices
```

```
flutter run
```


나. 백엔드 설치 및 사용방법

```
### 개발환경 설치
...

가상환경 설정
python3 -m venv .venv

가상환경 활성화
.venv/Scripts/activate.bat

패키지 설치
pip install "fastapi[standard]"
pip install sqlmodel
pip install pymysql
pip install redis
pip install passlib[bcrypt]
pip install PyJWT
pip install requests
pip install pandas
...

### .env file
DB_HOST=127.0.0.1
DB_PORT=3306
DB_USER=root
DB_PASSWORD=your_password
DB_NAME=your_database
REDIS_HOST=127.0.0.1
REDIS_PORT=6379
GOOGLE_SMTP_EMAIL=your_email
GOOGLE_SMTP_PASSWORD=your_smtp_password
SHELTER_API_SERVICE_KEY = 대피소 API 키
DISASTER_API_SERVICE_KEY = 재난 API 키

### Redis (Ubuntu,Linux)
sudo apt install redis redis-tools -y

Redis CLI 실행
redis-cli

### FastAPI 서버 실행
fastapi dev main.py
```

5. 팀 소개

임나빈	박지은	판킨킨자우	황수민
			
팀장 백엔드 개발	백엔드 개발	프론트엔드 개발	프론트엔드 개발
Github	Github	Github	Github
ddoed2201@pusan.ac.kr	phlox22@pusan.ac.kr	pankhin12@pusan.ac.kr	hhy9706@naver.com

[백엔드]

- 임나빈 (부산대학교 정보컴퓨터공학부 컴퓨터공학과 2학년)
- 박지은 (부산대학교 정보컴퓨터공학부 컴퓨터공학과 2학년)

[프론트엔드]

- 황수민 (부산대학교 정보컴퓨터공학부 컴퓨터공학과 2학년)
- 판킨킨자우 (부산대학교 정보컴퓨터공학부 컴퓨터공학과 2학년)