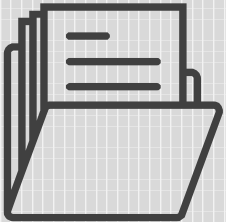


PMP



Team CMD

# ***Team CMD***

## ***Project Management Plan***

FRONT END : 임유빈, 안재현

BACK END : 전용현, 서장준, 박규현



# **0. Project Introduction**



# 프로젝트 정보

“재난대응 가이드 챗봇 개발”은 재난 상황 시 시민 대응 지원을 위한 공공서비스 지원이 필요함을 해소시킬 수 있도록 한다.

### 프로젝트 명

- 재난대응 가이드 챗봇  
개발

### 개요

- 주제 영역 및 기술 분야 - 공공/교통, 모바일
- 성과목표 - 실용화/ 공모전(한이음 공모전)

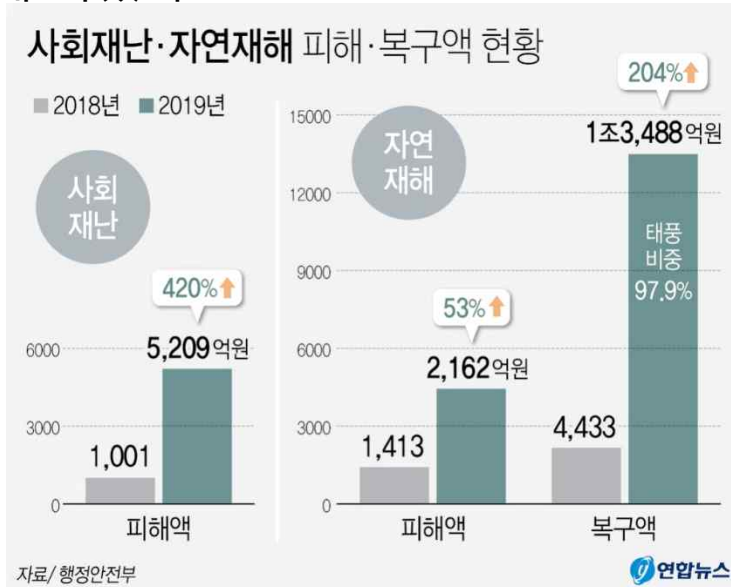
### 기간

- 23. 05.01 ~ 23.11.30



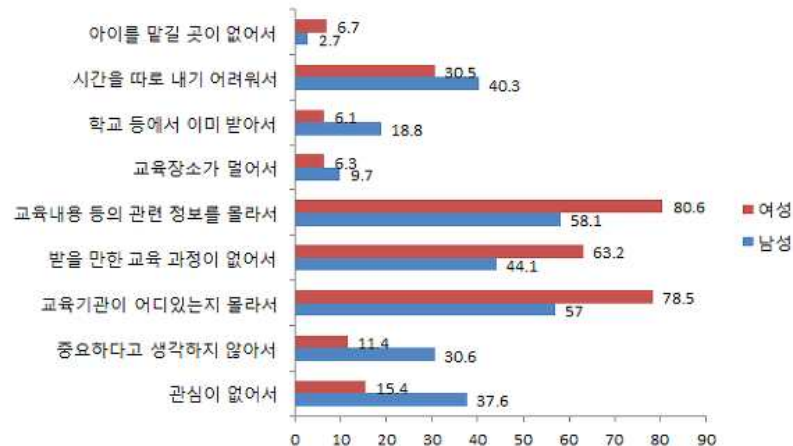
# 프로젝트 소개 및 제안배경

시민들이 사회 재난 및 자연재해로 입는 피해가 기하급수적으로 증가 하고 있다.  
대처법을 몰라서 더 큰 피해를 입는 경우도 적지 않기에 이를 해소 할 수 있는 프로젝트가 필요하다고 생각하게 되었다.



김영은 기자 / 20200812

트위터 @yonhap\_graphics 페이스북 tune.y.kr/LeYN1





# 타겟 사용자

안전불감증으로 인한 재난 상황 시 대피 및 대처법을 모르는 사람이  
증가

정보가 통일되지 않고 방대하기 때문에 직접 찾아보지 않으면 알기  
힘듦

빠르고 간결하게 원하는 대처법을 제공 받고 싶다.



# 주요 기능

- 재난 종류 실시간 파악 및 알림(재난 문자 같은 역할)  
→원하는 종류의 재난 문자만 받을 수 있도록 필터링
- 대피시설 정보제공(이용자 GPS정보 기반, 지정한 지역 기반)
- 대피, 대처 방법 안내 (정적인 정보)
- 재난 예측 서비스 제공(기상정보 바탕으로 산사태나 홍수 정도)



# UX(사용자 경험 전략)

- 너무 길지 않은 문단을 이용한 가독성 향상
- 잘 정리된 카테고리
- 난잡하지 않은 UI
- 시각적인 자료(사진, 영상)와 같은 예제를 통한 습득
- 사용자 친화적 로직 구성 (ex. 입력 완료시 다음 입력칸에 커서 이동)



# 적용 기술

## Spring 서버 개발

- Spring을 통한 빠른 통신이 가능한 서버 구현

## AI챗봇

- Pytorch를 이용한 개발
- Chatter-bot / langchain을 통한 ai 챗봇 모델 구현

## React 프론트 개발

React를 이용하여 웹앱 형식으로 구현, 가장 중요한 기능인 푸시 알람은 웹푸시 알람을 이용하여서 구현.





# 프로젝트 상세개발 내용

No	구분	서비스	기능
1	프론트 엔드	로그인	일반 로그인 / sns 로그인
2		회원가입	일반 로그인
3		내 위치정보 가져오기	사용자의 위치정보 가져오기
4		주변 대피시설 보기	정해진 반경의 사용자 주변 대피 시설 정보 제공
5		경고 push기능	경고 메시지 push
6	백엔드	로그인	일반 로그인 및 sns로그인 구현
7		재난 정보 알림	재난 정보 받아오기 / 원하는 알림 필터링
8		대피시설 정보 api	사용자 위치 기반으로 최단거리 순 대피시설 정보 제공
9		재난 예측 서비스	일기예보 등 날씨정보 기반으로 재난 예측
10	챗봇	재난 대응 매뉴얼 제공	원하는 상황에 따른 대응 및 대피 매뉴얼 제공



## 1. 역할 소개

# CMD 팀 소개



FE / 안재현

Role / 코드 품질 관리



FE / 임유빈

Role / 코드 품질 학습 관리자



팀장 / BE / 전용현

Role / 프로세스 품질 관리



BE / 박규현

Role / 프로젝트 관리 학습 관리자



BE / 서장준

Role / 스크럼 마스터

# CMD 팀 업무 분장표

이름	개발	역할
전용현	백엔드	프로젝트 품질관리
박규현	백엔드	학습 리드(전반적인 스터디)
서장준	AI 챗봇	스크럼 마스터
임유빈	프론트엔드	학습 리드 (코드 부분)
안재현	프론트엔드	코드 품질관리



## 2. 팀 목표

# “TEAM GOAL ”

## 1. 성장에 초점을 두기!

- 진행 기간 동안 중도탈락 없이 성장에 초점을 맞춰 최대한 얻어가기

## 2. 기억보단 기록을 ...

- 문서화 할 수 있는 것은 기록으로 남겨두기

## 3. 기회가 생겼을 때 도전!

- 경험이 없는 아키텍처, 디자인 패턴에 대해서 이해하고 적용하기



### 3. 스프린트 계획

## Step. 1

### 1차 스프린트 🖥

- ✓ 유저 기능(로그인 & 회원가입)

재난 정보 제공

- ✓ 지역별 재난 정보 제공

- ✓ 위치 기반 재난 대피소 정보 제공

- ✓ 사용자 커스텀 알림

## Step. 2

### 2차 스프린트 🖥

- ✓ 채봇 기능

- ✓ 재난 예측 서비스 제공

- ✓ 유저 긴급 연락망 기능

- ✓ 위치 기반 재난 대피소 네비게이션

- ✓ 코드 리팩토링

## Step. 3

### 3차 스프린트 🖥

- ✓ 재난 관련 뉴스 및 업데이트 정보 제공

- ✓ 재난 복구 정보 제공

- ✓ 재난 피해 보상 정보 제공

- ✓ 성능 최적화



[illegible]





## 4. 프로젝트 백로그

## 프로덕트 백로그

### 1차 스프린트 백로그

1. 유저 기능
2. 재난 통보 기능 개발
3. 실시간 재난 정보 제공 기능 개발
4. 안전 지침 제공 기능 개발
5. 재난 대피소 정보 제공 기능 개발
6. 사용자 커스텀 알림 설정 기능 개발
7. 챗봇
8. 재난 예측 정보 기능

### 2차 스프린트 백로그

1. 다국어 지원 기능 개발
2. SOS 신호 기능 개발
3. 긴급 연락망 기능 개발

1차 스프린트 : 마일스톤 1단계로서 최소 필수 기능을 기반으로 작성

2차 스프린트 : 마일스톤 2단계로서 배포 전 사용자 경험을 기반으로 코드 리팩토링과 함께 이후 추가할 기능

3차 스프린트 : 마일스톤 3단계로서 추가적으로 업데이트 할 내용으로 크롤링과 같은 있으면 좋은 기능

### 3차 스프린트 백로그

1. 재난 관련 뉴스 및 업데이트 제공 기능 개발
2. 재난 복구 정보 제공 기능 개발
3. 재난 피해 보상에 대한 정보 제공

# Planning Poker

## Planning game중 planning poker 진행

← CMD

스프린트 MD측정

TASKS

UI/UX 디자인	5
실시간 재난 정보 제공	10
챗봇 기능 제공	5
안전지침 제공	5
재난 대피소 정보 제공 (위치 정보 기반)	6
재난 예측 서비스 제공	7
로그인 / 마이페이지	5

← CMD-back

TASKS

유저 기능 (로그인 회원가입)	3
재난 정보 제공	6
지역별 재난 정보 제공	7
위치 기반 재난 대피소 정보 제공	5
사용자 커스텀 알림 설정	5

← CMD -chat bot

TASKS

SOS 신호 기능 개발	17
긴급 연락망 기능 개발	0
다국어 지원 기능	5



## **5. Work Flow**

# Meeting

Weekly Planning

- 일요일 1시 ~ 3시 정기세미나
- 일주일 간 공부한 내용, 결과물 공유 & 발표
- 다음 세미나까지 공부할 내용 계획



# Meeting

Daily Planning

- Discord Bot을 활용한 데일리 스크럼
- 평일 11시까지 스크럼 보고 완료
- 팀원들의 일간 작업 진행 정도 파악 가능
- 팀원들의 현재 상태(멘탈, 코드 등) 파악 가능

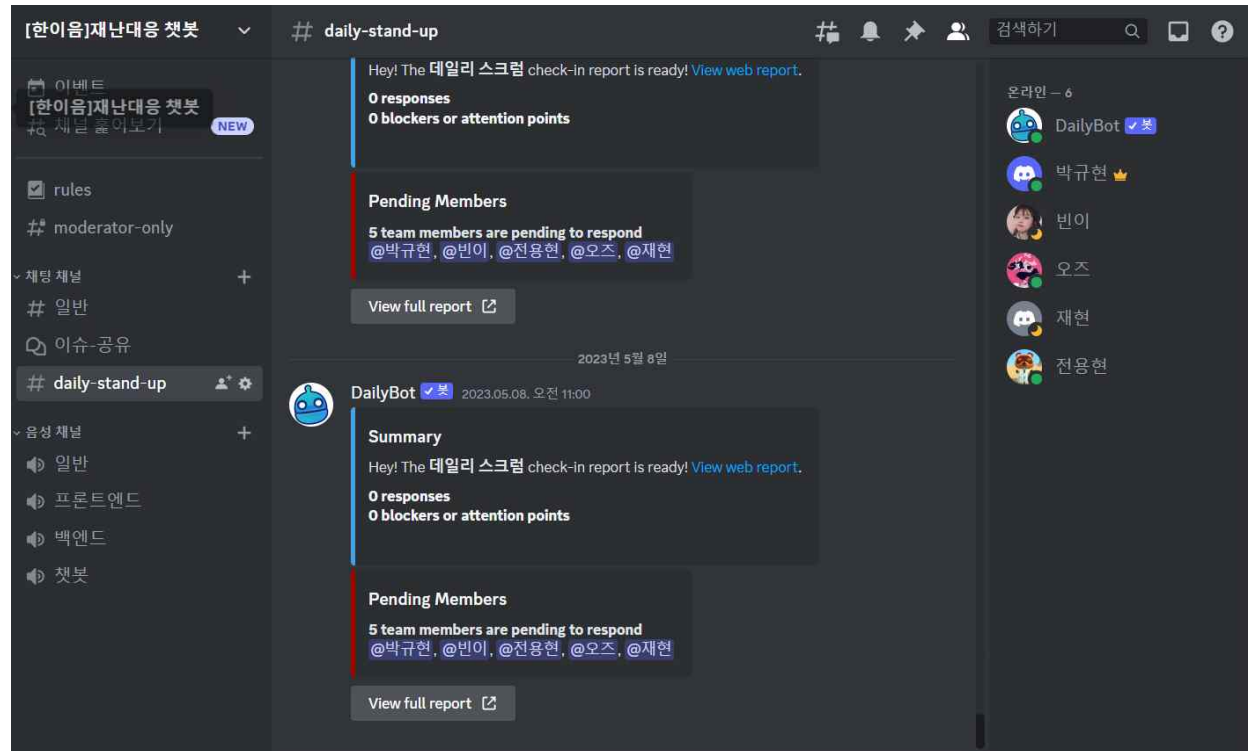
## Question

- 오늘 기분은 어떠시나요?
- 오늘 할 일 적기
- 어제 하던 일 중 남은 일 적기
- 팀원과 공유하고 싶은 내용이 있다면?
- 현재 가장 직면한 큰 문제는 무엇인가요?
- 팀원들에게 하고 싶은 말



# Trouble Shooting

- 데일리 스크럼에 지속적으로 이슈 공유
- 디스코드 폼을 통해 이슈 공유
- 이슈 있으면 팀원들과의 미팅



# Development Rules

• Commit convention 통일

• 클린 코드 작성 원칙 적용

• 소스 코드 최적화 적용

Feat	새로운 기능 추가
Fix	버그 수정
Docs	문서 관련
Style	스타일 변경
Refactor	코드 리팩토링

## <클린 코드 작성 원칙>

### 1. 가독성

이해하기 쉬운 용어를 사용, 코드 작성 시 들여쓰기 기능을 사용

### 2. 단위성

한 번에 한 가지 처리만 수행, 클래스/메소드/함수를 최소 단위로 분리

### 3. 의존성

영향도를 최소화, 코드의 변경이 다른 부분에 영향이 없게 작성

### 4. 중복성

중복된 코드를 제거, 공통된 코드를 사용

### 5. 추상화

클래스/메소드/함수에 대해 동일한 수준의 추상화 구현, 상세 내용은 하위 클래스/메소드/함수에서 구현

## <소스 코드 최적화 기법>

1. 변수나 클래스, 메서드 명을 의미가 분명한 이름(사용용도, 작업명)으로 사용한다.

2. 클래스는 행위의 주체로 명사나 명사구로 표현하고 함수 이름은 클래스가 행하는 행위로 동사 또는 동사구로 사용한다.

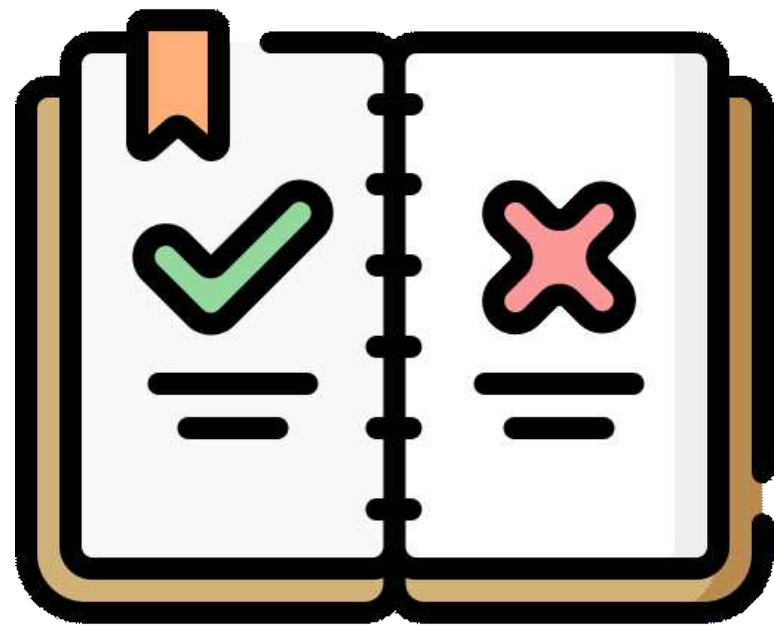
3. 클래스는 하나의 역할, 책임만 수행할 수 있도록 응집도를 높이고, 크기를 작게 작성한다.

4. 클래스의 자료 구조, 메소드를 추상화할 수 있는 인터페이스 클래스를 이용하여, 클래스 간의 의존성을 최소화해야 한다.

# Ground Rules

주의할 점

- 일정을 지속적으로 공유하기
- 노선에 회의록 작성 (전체, 트랙별 동일)
- 데일리 스크럼을 통해 본인의 상태 알리기
- 눈치 보지 말고 본인의 생각(조언, 비평 등) 말하기





### 3. 개인 목표

# 개인목표 - 임유빈

## Current

- React의 기초단계
- Hook에 대한 이해도 부족
- 상태 관리 라이브러리에 대한 이해 부족
- Docker와 github action을 이용한 ci/cd 아주 간단한 구현
- Docker-compose에 대한 지식이 없음

# 개인목표 - 임유빈

## GOAL

- React with TypeScript
- React Hook에 대한 자세한 이해, custom hook를 만들고 활용
- Javascript의 심화, 동기 / 비동기처리등의 이해
- Zustand, react-redux와 같은 상태관리 라이브러리
- Docker와 github action을 이용한 CI/CD
- Docker, Docker-compose등 docker에 관련한 사용법

## 개인목표 - 안재현

### Current

- Readct의 기초단계
- CI/CD 구현해 본적 X
- 구현 로직과 UI 로직 분리 등 효율적인 코드 관리 미흡

# 개인목표 - 안재현

## GOAL

- TS를 사용하여 React코드 작성
- Github actions를 사용하여 CI/CD 경험
- 코드 분리 신경써서 컴포넌트 효율적으로 관리
- Github 사용법 익히기
- Emotion 학습해서 적용하기



## 개인목표 - 전용현

### Current

- Django, JDBC 사용 경험
- 자바와 spring을 공부 중
- 다양한 디비를 사용한 경험이 없음

# 개인목표 - 전용현

## GOAL

- 자바 & Spring을 이용하여 구현하기
- 데이터와 DB와의 관계를 좀 더 생각하고 관리하기(데이터 정규화)
- 작성된 기능 단위 개발 및 테스트 코드 시도해보기
- 배포에 대해 정확하게 익히고 활용하기(Docker / Kubernetes등..)
- 문제가 발생했을 때 고민하고 회고록 작성 -> 구글링을 하는 것에서 그치는 것이 아닌 문서화 시켜, 복습, 정리 능력까지 갖추는 것이 목표
- 내가 아는 것과 모르는 것에 대한 확실한 구분을 두고 이슈 / 활동 문서화하는 습관 형성

## 개인목표 - 서장준

### Current

- Django, Fast-API 사용 경험
- NLP 학습 경험은 google-mT5

## 개인목표 - 서장준

### GOAL

- 애자일 방식 이해 및 숙련
- BentoML 을 이용하여 model serving 해보기
- Docker 공부 및 활용
- 판단의 근거를 기록하여, Feed-back 받기

## 개인목표 - 박규현

### Current

- Node.js와 express 경험만 있음
- 자바와 spring을 공부 중
- 다양한 디비를 사용한 경험이 없음

# 개인목표 - 박규현

## GOAL

- 자바 & Spring을 이용하여 구현하기
- 다양한 DB 를 사용해서 판단해보기(noe4j, MongoDB, Redis와 같은 NoSQL도 고려)
- 데이터와 DB와의 관계를 좀 더 생각하고 관리하기(데이터 정규화)
- DDD 아키텍처를 바탕으로 생각하기
- 작성된 기능 단위 개발 및 테스트 코드 시도해보기
- 문제가 발생했을 때 고민하고 회고록 작성 -> 구글링을 하는 것에서 그치는 것이 아닌 문서화 시켜, 복습, 정리 능력까지 갖추는 것이 목표

END

감사합니다.