

## 주간보고서 3

### 32181771 모바일시스템공학과 박용관

이번 주차는 크게 저번 발표 이후 피드백 반영, 개발을 위한 환경 설정(버전 설정), 전체적인 시스템 구조를 정의하고 사용 기술 및 추진 전략 등을 정리하였습니다.

#### ○ 피드백 반영

구현할 명령어를 수정하였고 추가적으로 cli환경, spring서버를 사용하는 이유에 대한 근거를 정리하였습니다. 아래 내용은 근거에 대한 내용입니다.

##### • CLI 환경에서 사용할 수 있는 명령어 개발

저희는 CLI환경에서 chatgpt와 통신을 원활하게 할 수 있는 명령어를 개발할 예정입니다. 위에서 말한 프로젝트 배경에 따르면 현재 chatgpt를 사용하는 사용자는 기하 급수적으로 늘어나고 있고 많은 분야에서 쓰이고 있습니다. 그에 대한 예시로 여러 스타트업에서 chatgpt를 사용한 다양한 서비스를 개발하고 있습니다. 그래서 현재 트렌드에 맞게 chatgpt를 사용한 서비스를 개발하려고 했습니다.

따라서 이러한 chatgpt를 사용하여 저희 강의에 맞게 어떻게하면 오픈소스에 기여할 수 있는지를 생각해보았습니다. 그 결과, "Unix 환경에 기여하기 위해 편리하게 chatgpt를 사용할 수 있는 명령어를 만들면 좋겠다." 라는 생각이 들었습니다. 따라서 "기존 서비스가 UI적인 측면에서 chatgpt를 사용한 서비스였다면, 저희같은 경우에는 UI가 존재하지 않는, CLI환경에서 chatgpt를 편하게 쓸 수 있는 서비스를 만들면 Unix환경에 기여할 수 있겠다." 라는 생각이 들었습니다. 실제로 저 또한 CLI환경에서 작업을 많이 하는데, 이때 chatgpt에 질문을 하려고 하면, "https://chat.openai.com/" 해당 사이트에 들어가서 질문을 자주 합니다. 이 과정이 번거롭고 UI가 없는 CLI환경에서 작업을 한다면, 다른 OS가 없으면 chatgpt에 질문하기 쉽지 않습니다. 따라서 만약 저희 서비스가 개발된다면, 서버 개발을 하면서 Unix환경에서 chatgpt사용을 더 편하게 할 수 있는 기대효과를 볼 수 있습니다.

##### • 서버 DB를 통해 동일 질문 응답 시간 단축(cache)

실제로 chatgpt를 사용하다보면, 같은 질문을 반복적으로 할때가 많습니다. CLI환경에서도 만약 curl명령어를 통해 chatgpt와 통신한다면, 질문하고 후에 같은 질문을 해야할 상황이 온다면 chatgpt에 다시 curl명령어를 통해 질문해야합니다. 이 경우 또다시 chatgpt가 해당 질문을 요청으로 받고, 해당 질문에 대한 답을 응답으로 받을 것입니다. 이 과정에서 시간이 전 질문과 같은 질문인데도 불구하고 많은 시간이 걸릴 것입니다. 따라서 이 부분을 개

선하기 위해서 저희는 저희 서버에 chatgpt질문과 답을 저장할 것입니다. 그래서 이 후, 같은 질문을 한다면 open ai서버에 요청이 가는 것이 아닌 저희 서버 DB에 저장된 답을 사용자에게 보여줄 것입니다. 그렇게 된다면 동일한 질문에 대한 응답이 빠르게 온다는 기대효과를 볼 수 있습니다.

## • 사용자들의 질문 유형 저장

요즘 개발자들은 해당 키워드에 대한 어떤 질문을 하는지 궁금할 수도 있다고 생각합니다. 따라서 저희 서비스에서는 사용자들의 질문을 저장하고 만약, 한 사용자가 다른 사용자들이 “JAVA”에 대한 키워드를 가지고 어떤 질문을 했는지 궁금할때, 해당 키워드에 대한 다른 사용자들의 질문 내용을 보여줍니다. 물론, 사용자들이 다른 사용자들에게 자신의 질문을 보여주기 싫을 때도 있을 것이라고 생각합니다. 이때는 private한 질문 형식이라고 명시함으로써, 해당 질문은 따로 서버 DB에 저장하지 않도록 처리할 예정입니다. 이로 인해 사용자들은 다른 사용자들이 어떻게 질문하는지에 대해 배우고 자신의 질문 스타일을 조정할 수 있다는 기대효과를 볼 수 있습니다.

## ○ 전체적인 시스템 구조



## ○제공되는 리눅스 명령어

	형식	설명
1	chatAsk -m “~~” -pri	질문하기 - DB 저장 X (질문 노출 X)
2	chatAsk -m “~~”	질문하기 - DB 저장 O(키워드 추출 후 키워드 저장)
3	chatAsk -gc -m “~~”	문장 맞춤법 검사
4	chatAsk -t -m “~~~”	문장 번역(한국어)
5	chatAsk f —m “~~~”	문장 감정 분석
7	chatAsk -s -m “~~~”	문장 요약
8	chatAsk -q -k “~~~”	키워드를 사용해 사용자들 질문 추출

명령어 형식은 바뀔 수 있습니다.

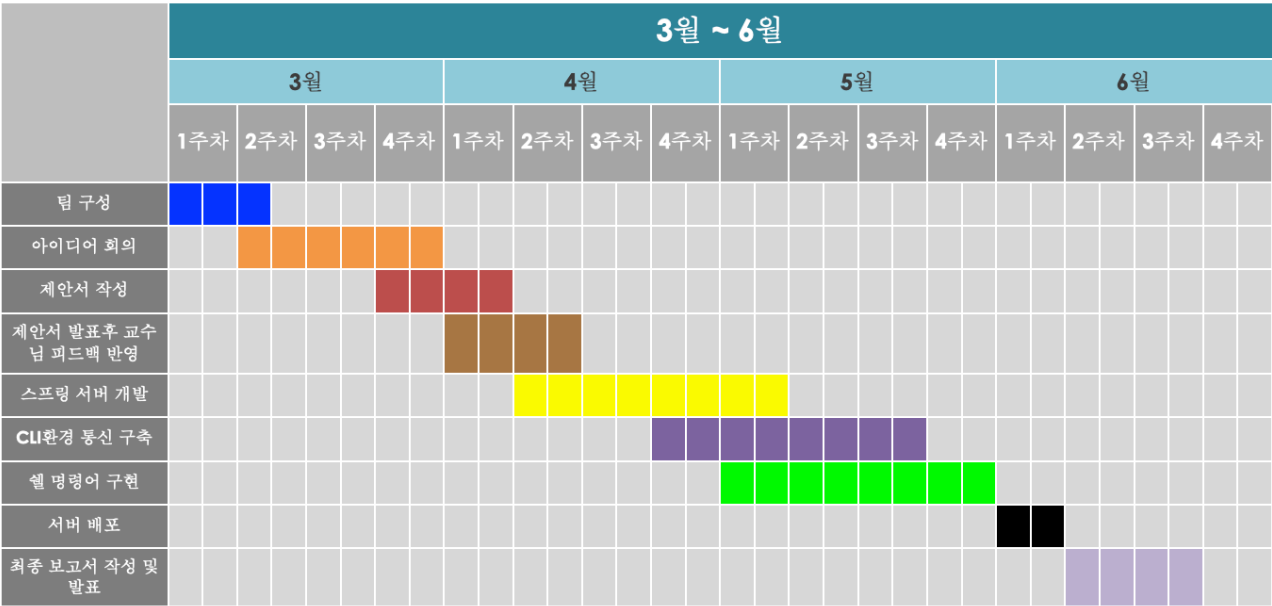
• 사용기술

이름	역할	사용기술
박용관	백엔드 개발	1. Spring 버전 : 2.7.10 2. 자바 버전 : 11 3. MySQL 데이터베이스 버전 : 8.0.21 4. 쉘 스크립트
서지현	백엔드 개발	1. Spring 버전 : 2.7.10 2. 자바 버전 : 11 3. MySQL 데이터베이스 버전 : 8.0.21 4. 도커 버전 : 20.10.21

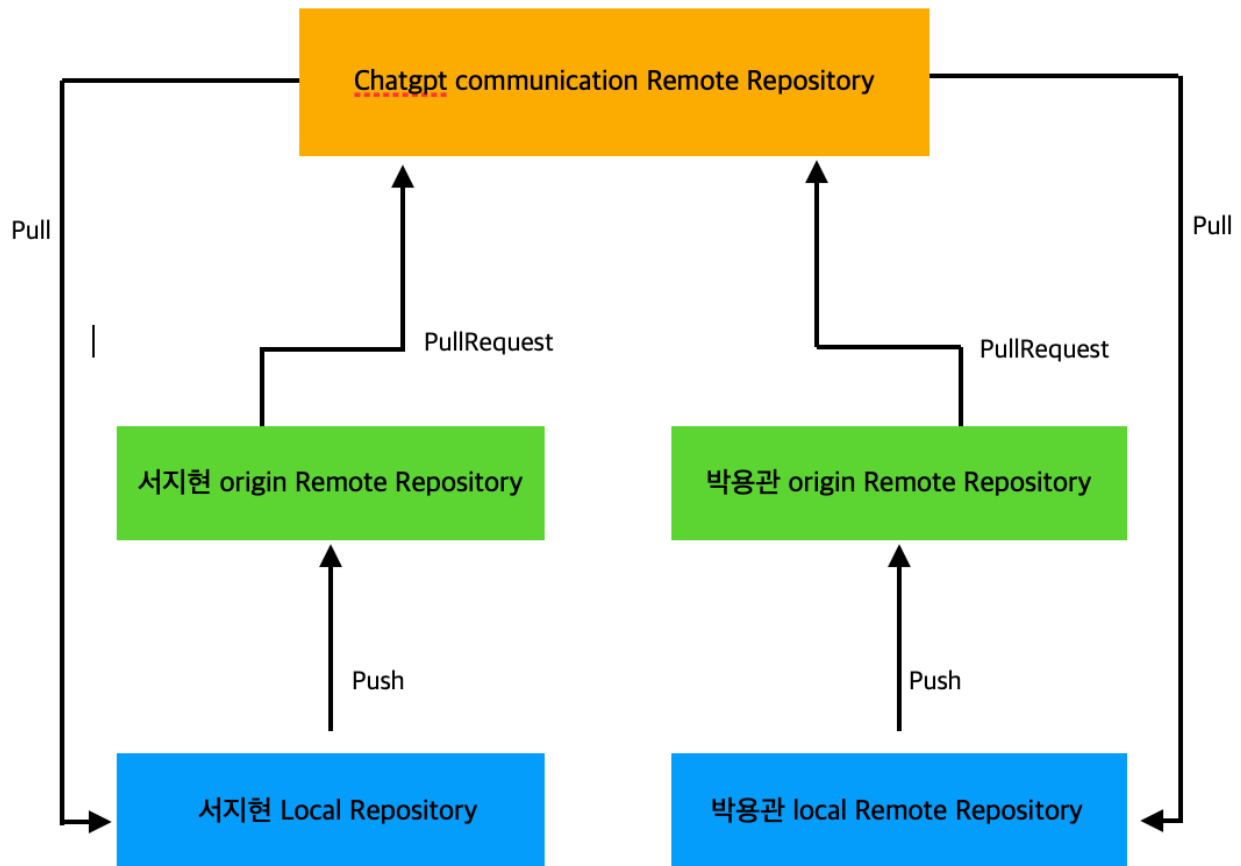
○ 추진 전략

1. 간트 차트

> GANTT CHART



## 2. Git Repository 구성



Git Repository 구성은 다음과 같습니다.

Repository는 Chatgpt communication Remote Repository, Origin Remote Repository, Local Repository 이렇게 3부분으로 구성됩니다.

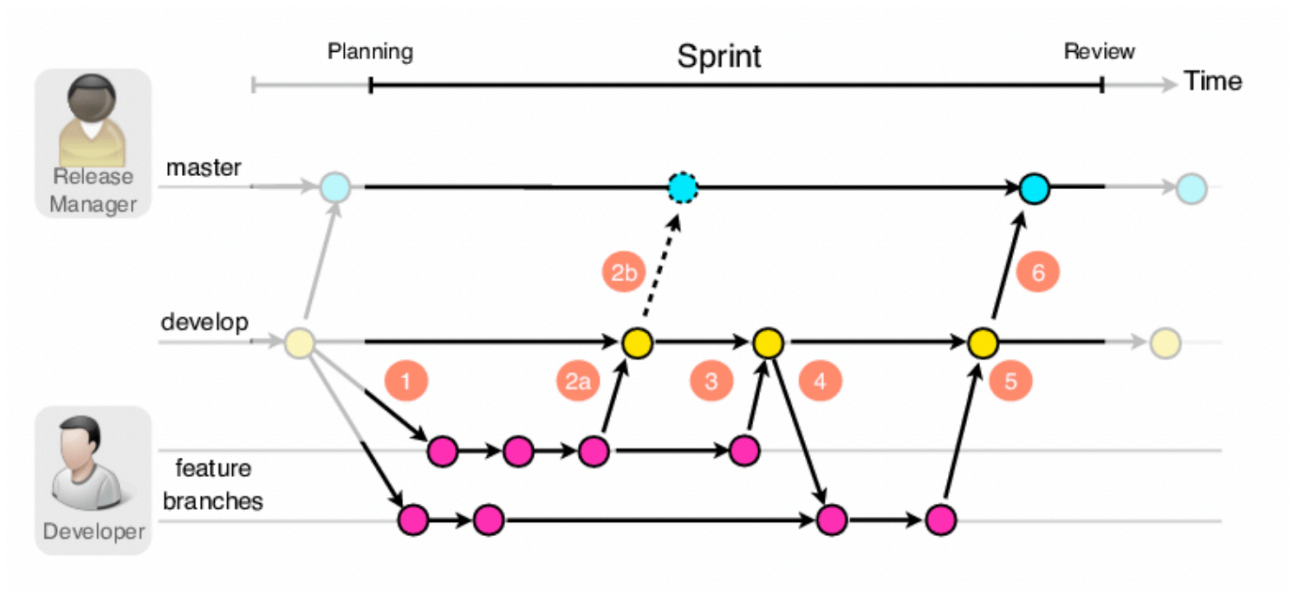
Chatgpt communication Repository는 개발자들이 공유하는 저장소로 최신 소스코드가 저장되어 있는 원격 저장소입니다.

Origin Repository는 Chatgpt communication Repository를 **Fork**한 원격 개인 저장소입니다.

Local Repository는 내 컴퓨터에 저장되어 있는 개인 저장소입니다.

작업시 지켜야 할 서로간의 약속은 다음과 같습니다.

1. Pull request 후, 서로의 코드를 리뷰하며 최종적인 검수 작업이 끝나면 머지합니다.
2. 서로가 서로의 코드 리뷰를 해줍니다.
3. 커밋의 단위는 최소한으로 가져갑니다.
4. 서로 공유하는 브랜치의 커밋 그래프는 합부로 변경하지 않습니다.
5. 사전에 정한 git - flow 전략을 통해 버전을 관리합니다.



Chatgpt communication Repository에서 git-flow 사용시 사용되는 브랜치는 총 3가지 입니다.

- main : 도커허브 이미지 업로드, 오픈소스로 제공하는 브랜치
- Develop : 프로젝트에서 여러가지 도메인을 관리하는 브랜치
- Feature : 프로젝트에서 하나의 도메인의 기능을 개발하는 브랜치