



캠프장님과의 2번째 미팅

Date	@2023/12/14
Tags	캠프장님

- 승용님 설명
 - 팀원별로 개인 의견 정리해주셨음
 - 종준님 의견도 정리해서 알려주심
- 캠프장님: 지금까지 써왔던 게 불편 했다는 의미가 있어서 좋은 것 같다.
 - 캠프에서 잘만들면 직접 써보면서 할 수 있지 않을까
 - 캠프에서 추진 하는 것 → 프로젝트를 클론하다보니 매번 비슷한걸 반복하다 보니, 커뮤니티 내에서 나오는 솔루션이나 기능을 구현할 수 있으면 좋을 것 같다. 내부 오픈소스 화해서 좀 더 발전시키는 방향도 좋을 것 같다. (SOPT MAKERS를 참고하면 되지 않을까?)
 - 프론트, 백엔드 각 2명 인게 좋은 것 같다.
 - 어느 팀 프론트엔드가 와서 어떠한 일을 하든 비슷한 일을 하게 된다.
 - 소스 코드 받아오고 프론트엔드 쪽 기술이 효과적으로 작동하고 돌아가도록 해야 될 필요가 있는 것 같다.
 - 의미도 있었고, 재미도 있었지만, 캠프가 추구하는 방향이랑 말을지 → 향후의 실태에 맞는 방향으로 갈지 / 캠프의 리더가 될 수 있을 정도의 아키텍처, 방법론 까지 다룰 수 있도록 성장 할 수 있으면 좋을 것 같다. → 윈터 2기는 특히나 기본적인 줄프 정도의 느낌이었기 때문에 그런 부분을 좀 더 신경 쓰면 좋을 것 같다.
 - webrtc와 같은 기술을 적용하는 것과 같이 기술적인 도전을 적용하는 것도
 - 소스코드를 어떻게 가지고 올 건지
 - 1차 미팅 때는 파일을 입력하고 들고 오는 쪽으로
 - 코드의 컬러링, syntax 등등의 효과를 넣어주는..?
 - 규민: 노션의 코드 스니펫 같은 라이브러리를 적용할 수 있도록 하면 괜찮지 않을까?
 - 종준: github에서 소스 코드를 가져올 수 있는 api는 따로 없는 것 같다.
 - 캠프장님: 그냥 소스 코드를 끌고 올 수 있도록 하면 어떨까? pull로 땡겨와서 사용할 수 있도록 해야 될 것 같음.
 - github 계정을 연결해서 쓸 수 있도록 하면 좋을 것 같음.
 - commit을 기준으로 읽어올 수 있도록 하는 것도 나쁘지 않을까?
 - git protocol 땡겨오는 거 + 레포지토리 처리를 어떻게 할 것이며
 - 캠프장님: 파일선택을 불러올 수 있도록 작업 정도는 해줄 수 있지 않을까
 - copy&paste는 좀 짜치는 느낌(ㅋㅋㅋ)

→ 외부 소스를 연동을 하거나 불러올 수 있도록 하는게 좋지 않을까

→ file 업로드는 file 관련해서 포맷, 언어에 따라서 달라지는 요소가 매우가 많아서 신경써야 되는 부분이 많다. 대신에 공부하는 꽤나 많이 될 수 있을 것 같다.

→ 캠프장님: copy&paste는 프론트 하면 좋지만 그럼 백엔드가 줄어들지 않을까?

highlight.js

A syntax highlighter written in JavaScript supporting 192 languages for Node.js and the web.

 <https://highlightjs.org/>

```
class MyClass {
  public static myValue: string;
  constructor(init: string) {
    this.myValue = init;
  }
}
import fs = require("fs");
module MyModule {
  export interface MyInterface {
    myProperty: any;
  }
}
declare magicNumber
myArray.forEach
```

highlight.js

→ 캠프장님: 백엔드가 좀 더 할 수 있는 것

→ 좀 더 제품화 된 기획으로 바뀌면 좋을 것 같다.

→ 실시간으로 수정하는 부분에 대한 걸 webrtc, socket으로 다룰 수 있으면 기술적으로 많은 걸 해 볼 수 있지 않을까?

○ 총평

- 전체적으로 4명에서 하기 좋은 프로젝트 같다.
- 실제로 디테일이 나오면 같이 얘기를 더 해봐야 될 것 같음.
- 진짜로 기술적인 부분을 포함해서 아키텍처를 그려왔고, 이부분이 어떻게 구현 되겠구나라는 생각이 캠프장님과 우리 의견이 유사한 정도는 되어 됨.
- PMP 작성을 좀 해야함.
 - 샘플을 너무 많이 참고하지 말고
 - PMP에는 ERD, 아키텍처는 굳이 적을 필요가 없음
 - PMP에는 목표, 포부가 많이 들어가는 게 좋음.
 - 이사님은 전체적으로 목표나 계획에 대해서 보고 싶은 거니까 그 부분 참고 해서 하면 좋을 것 같다.
- 전체적인 디테일을 좀 더 잡아서 들고 오면 될 것 같음.
- 기술적인 부분에 대한 디테일을 어떻게 잡아갈지에 대해서 고민을 해야 될 것 같음.
 - 백엔드가 소스제어 부분이나 관리부분을 가지고 오면 좀 더 괜찮을 것 같다.
 - 기술적인 기획적인 부분을 정리했으면 좋을 것 같다.

• ? 질문 타임

○ 규민: 기초적인 베이스 기능은 다 잡아줬는데 어떤식으로 확장을 해나가는 게 좋을지

- 노션이나 피그마 처럼 유저의 화면이나 커서 위치를 볼 수 있다거나, 음성을 실시간으로 할 수 있도록 보이 스톱으로 바로 할 수 있도록, 싱글페이지 형식으로 공유해주는 걸 하자.

→ 확장의 기준은 어느정도 잡아가야 할지?

→ 캠프장님: 기능으로 봤을 때는 필요한 기능이 실질적인 기술을 다루는 부분에서는 생각보다 중요하지 않을 수가 있음.

- webrtc 등으로 실시간으로 커서를 공유한다거나 그런 부분은 기술적으로 공을 들여야 되지만 생각보다 접근 권한 같은 부분은 기술적으로는 나가지 않는 부분일 수 있음.
 - 캠프는 기능적인 거의 완성도를 바라보는게 아님. 기술적인 보완 까지는 바랄 수 있지만
 - 우선순위를 정해서 팀 협업 적으로 어떻게 할 지에 대해서
 - 설계가 연동되는 부분에 대해서 놓치고 갈 수도 있지 않을까?
 - 각자 기술적으로 말아서 해결을 했는데 각자 연관성이 없는 걸 만들어오는 것도 문제가 있음.
- 규민: 기술과 기능의 선을 타고 있는 그 라인이라서 애매한 거 같긴 하지만, 산출물을 드리면 결과물에 대해서 캠프장님께서 피드백을 주시면 제일 좋을 것 같은..?
- 캠프장님: 항상 비슷한 걸 보다 보니 새롭게 관심있게 볼 수 있는 프로젝트를 하니까 같이 의견을 많이 나눌 수 있으면 좋을 것 같음.
- 당장은 추가적인 미팅이 필요하면 추가적으로 미팅을 요청 주시면 됨!
- 종준: 캠프에서는 왜 제품화에 가까운 그런 걸 추구하지 않는 부분을 추구하지 않는 이유? 기술적인 기능을 추가하고 구현하는 건 혼자서도 할 수 있는 부분, 자료들도 많지만 실질적으로 접근하기 어려운 건 실제로 만들어서 제품화 했을 때 내는 부분에 대해서 얻는 것도 많지 않을까?
- 캠프장님: 우리 프로젝트가 현업 처럼 되어야 되는 부분도 있음. 스케일 아웃해야 되는 특정 부분이 있을 것. 현업에서 하는 것 처럼 담당에 따라서 팀을 맡는 수준으로 MSA 구조로 개발을 하고 거기서 MSA 정도로 진행을 하는 거는 안정성, 성능 부분 까지 하는..?
- 필요 없는 부분, 보안과 같이 실질적으로 그런 분야를 갈을 때만 필요로 하는 것
 - 예를 들어서 webRTC를 다뤘을 때, 멤버십에 따라서 화질을 다르게 가져간다고 했을 때 720, 1080p 처럼 전문적인 분야까지 해볼 수 있지만, 그쪽분야로 가지 않는다면 생각보다 의미가 없는 영역일 수도 있음.
 - 스터디에서도 그렇지만, 필요없는 거 공부하는 것 만큼 도움이 안되는 게 없음. 운영체제 등과 같이 base에 들어와 있어야 되는 부분.
 - 요즘 러스트가 유명하다고 해서 러스트를 한다면? 도움이이 되겠지만 몇 달안에 프로젝트나 업무로써 쓰지 않으면 다른데 써야될 시간과 공간을 낭비하는 일이 될 수도 있음.
 - 공부도 선택을 하면서 공부를 해야 될 필요가 있음. 깊이 파는 것이 안중요한 것이 아니라 본인이 생각했을 때 어떤 분야가 본인에게 필요한지를 파악하고 걸러내서 하는 것이 중요하다는 점.
 - 데이터베이스 튜닝을 해야 되나? 사실 DBA가 해주는데 알아야 될 필요가 있을까?
 - 그래서 인터넷에도 있는 정보 수준으로 못하는 걸 도전하는 건 괜찮은데 그게 정말 도움이 될지는 생각을 해봐야 될 것 같다.
- 종준: 운영이나 필요한 부분 → 운영툴 같은 부분을 현업자가 되게 많이 강조를 하시는 데, 모니터링 툴 같은 것도 쓰는 게 좋을지?
- 캠프장님→ CI/CD나 모니터링은 안하는 쪽을 권유를 하는, 굳이 직접 하는게 아니라 취미 생활 처럼 해보는 것도 나쁘지 않음. 거의 간단하고 개발자가 실제로 하는 일은 아님, 누가 세팅해서 만들어주면 보는 경우가 많고 직접하는 경우는 거의 없는 것 같음.
- 엔지니어가 직접해주는 거라 굳이 알 필요는 없지만 IT 업계에서는 알아서 안좋은 건 없지만 굳..이?

- 제너럴 한 영역을 좀 더 다뤄보는 게 좋은 것 같다. 모니터링 하다가 시간을 좀 더 잡아 먹을 수도 있을 것 같다.
- 캠프장님: 메이저급 회사를 희망하고 있으면 그런 회사들에서는 아마 직접 쿼리를 날린다거나 모니터링 시스템을 직접 짜는 경우는 없을 것. 도커도 직접 올리는 경우도 없을 거고 설치해둔 서비스를 봄, 작은 회사로 가면 콕티..? 같은 걸 써서 직접 해야 되는 경우가 있을 것.
 - 원래 중소기업 다니시다 넘어오심 → 당시에만 해도 전부다 직접 해보셔야 됐어야 되서 오히려 회사 생활에는 도움이 덜 났지만 캠프에 도움이 됨 😊
 - 당시에 스토브 캠프를 시작할 때, 캠프장님이 모바일, 인프라, 작가, 디자이너, 게임 등등 다 하시다 보니까 캠프를 맡아서 들고 오셨는데 → 창업 하면 도움 될지도(순수 개발자나 연구직에는 크게 도움이 안될 것 같음)
- PMP 준비 마저 잘 해주시고, 어딜 가든 잡캐는 별로 안좋은 것 같음.
- 지금 하고 있는 그 일들을 잘하면서 가면 좋을 것 같다.
- 약간 얇은 거리로 엮여 있어서 그런 디테일을 서로 잘 맞춰가면 좋을 것 같음.
- 현우: 프론트 아키텍처나 디자인 패턴을 잘 공부하는..? 적용하는
→ 캠프장님
 - 한번 짜고 나서 한달 뒤에 리팩토링만 해봐도 코드 퀄리티가 올라갈 수 있음
 - 각각의 수준 차이가 엄청나게 보임.
 - 기술적인 게 코드의 예쁨이나 그런 부분이 아니라 코드의 질이나 성능을 올리기 위해서는 기본적인 지식을 많이 올려야 될 필요가 있다.
 - 데코레이터 패턴 같은 디자인 패턴들이 뭔지를 이해하고 짜봤을 수는 있지만 생각보다 이해도는 낮을 수도 있음.
 - 깊이 있게 무언가를 볼 정도가 되면 이해도가 높을 수 있음
 - 어떤 걸 참조하면 좋을지? → 오픈소스 코드를 보면 좋을 것 같음, 여러 commiter가 붙고 Constructor가 붙기 때문에 보면 좋다.
 - 미드의 CSI라는 드라마만 해도 작가가 400명 있고, 24화가 감독이 모두 다른 경우도 있지만 전체적인 색깔에 대해서 어색함을 느끼지는 않았음.
 - 메인 작가나 기획자가 퀄리티를 맞춰 나가주는 방식이 있음
 - 오픈소스를 뜯어보면서 그런 부분을 채워나갈 수 있음, 조금 쉬운 부분에 대해서 자주 썼던 요소들에 이해도가 높은 것들을 보는 게 좋을 것 같음.
 - 전체적인 부분을 보는 것 보다는 자주 사용하는 몇가지를 우선 순위로 삼아서 보는게 좋을 것 같다.
 - 개발자가 한계에 도달하는 경우는 이런식으로 난이도가 높은 코드를 볼 때 피하거나 회피하고 놓는 경우, 그걸 안피하고 보면 실력이 확 성장하는..?
 - 보면서 공부할 걸 쌓아갈 필요가 있음. 오픈소스를 사용한다고 하면 사용은 어떻게 한다고 하면, 그 상태에서 어떻게 돌아가는지 뜯어보는 필요가 있음. 이해가 안되면 이해가 안되는 그 부분에 대해서 확실하게 이해를 해야 될 필요가 있음.

- 깊음을 느끼는 단계가 올 것임. 이해했을 때 오는 이해의 단계가 있음
 - 종준 → 연합동아리의 코드를 보는 것도 도움되는 것 같음
 - 종준: 디테일을 잡아가는 부분에 대해서
 - 캠프장님: 디테일을 잡아가는 부분
 - 기획과 아이디어가 나왔다면 스토리보드를 짠다거나 WBS를 구축한다거나 그렇게 까지는 아니더라도 전체적인 설계도 처럼 돌아가는 구상을 보여줄 수 있도록 하는게 좋을 것 같음.
 - 그걸 따라 프로그램을 만들 수 있도록 전체적인 설계도를 작성하는 IA, 기능 명세서
 - 기능명세는 진짜 기능명세일 뿐이고 화면 구성이나 기술 스펙 까지 종합하는 부분까지 고려를 한 거를
 - 조금 더 진행하고 다시 캠프장님께 여쭙보는 방식으로 진행을 해도 될 것 같음.
 - 와이어 프레임이 있으면 어떤 페이지에서 어떤 기술을 쓰고 어떤 기능을 제공할지가 좋을 거 같다.
 - 여기서 확장을 더 해나가는 거는 점선으로 그려놓아도 될 것 같음. 앞으로 이런 걸하면 좋을 것 같다. 나중에 가지치기 하면 되니까!
 - 다음회의
 - 유저 플로우
 - 확장하는 기술을 생각해서 가지치기를 하는게 좋을 것 같음.
 - 들어가면 좋을 것 같은 기술들도 적어오기
 - git에서 extension 확장하기
- 주제 : "지난번에 못했던 유저 플로우(기본기능에 중점으로) 짜오고 + 확장기능(가볍게 아이디어 + 왜 필요 한지)
- 유저 플로우 짜는 과정에서 기본기능을 구체적으로 작성해오기
 - 📎 필요한 라이브러리나 레퍼런스 정리해오기
 - 😊 확장기능은 가볍게 아이디어 정도로 + 만약에 아이디어를 던진다면 필요한 이유도 있으면 좋을 것 같음