

간단하고 강력한 UI/UX
소통이 편한 엔지니어



Hong Jae Min
1998. 01. 24
+82 010 5417 5563
dheldh20@gmail.com
<https://github.com/fkeldh2089>

Activity

삼성 청년 소프트웨어 아카데미
(2022.01~2022.12)

Education

아주대학교 (전자공학) 2016.03~2022.02 경기도 수원시
창현고교 2013.03~2016.02 경기도 수원시

Qualification

TOEIC 2023.01.14 870
OPIC 2021.07.11 IM1

Skills & Ability



Projects

Ollenge 22.10.10~22.11.25
갯생을 살기 위한 이용자들이, 도전과제를 이미지를 통해 인증해 나가는 서비스

FastAPI **OpenCV** **R•N**

MaFya 22.08.22~22.10.07
인공지능을 통한 간편한 출석 및 출결을 지원하는 서비스 시스템

Flask **React**

We See Play 22.07.05~22.08.19
OpenVidu를 활용한 웹 화상 게임 서비스

Vue3 **OpenVidu**

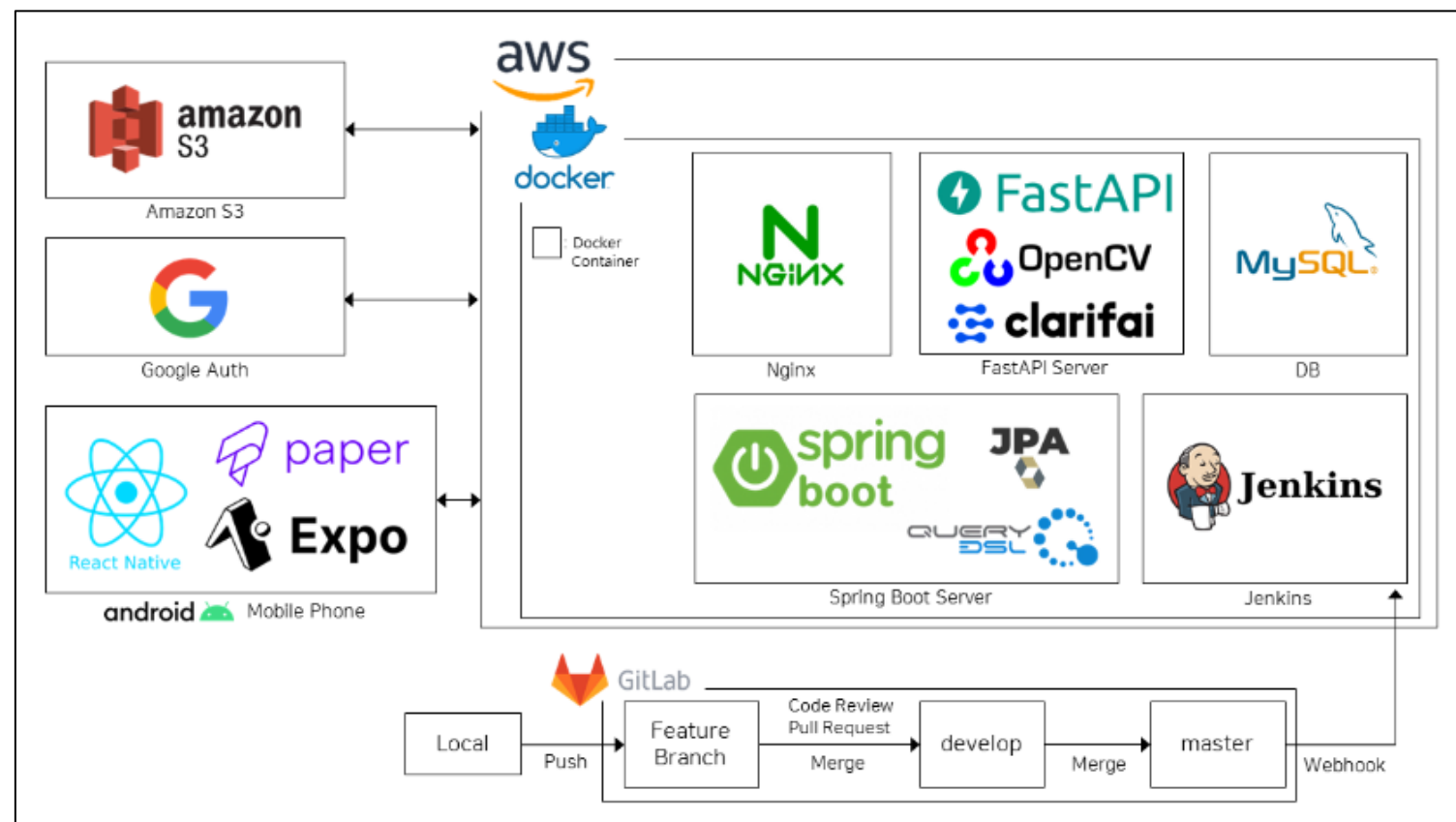
O`llenge

오늘의 챌린지

갓생을 살기 위한 이용자들이 매일매일 도전하고, 이미지를 통해 인증해 나가는 서비스

기간	2022.10.10~2022.11.25
기술	React Native, Expo, Spring Boot, FastAPI, OpenCV(AKAZE), Clarifai, AWS, Nginx, Docker, Jenkins
팀 구성	5명(Front-end 3, Back-end 2)
맡은 업무	<ul style="list-style-type: none"> Figma를 사용한 와이어프레임 제작 Fast API에 이미지 인증 시스템 구현 Fast API를 통한 feed CRUD 구현
배운 점	<ul style="list-style-type: none"> Fast API와 React-Native간 이미지 전송 OpenCV를 이용한 feature matching 유연한 업무 분담
관련 링크	

System Architecture





챌린지(도전과제)를 설정하고 도전

- >> 사용자가 원하는 챌린지를 정하고 도전
- >> 점수와 배지를 획득

랭킹 챌린지 피드 목록



인증 사진 업로드 페이지



기준 이미지 확인



인증 피드 작성

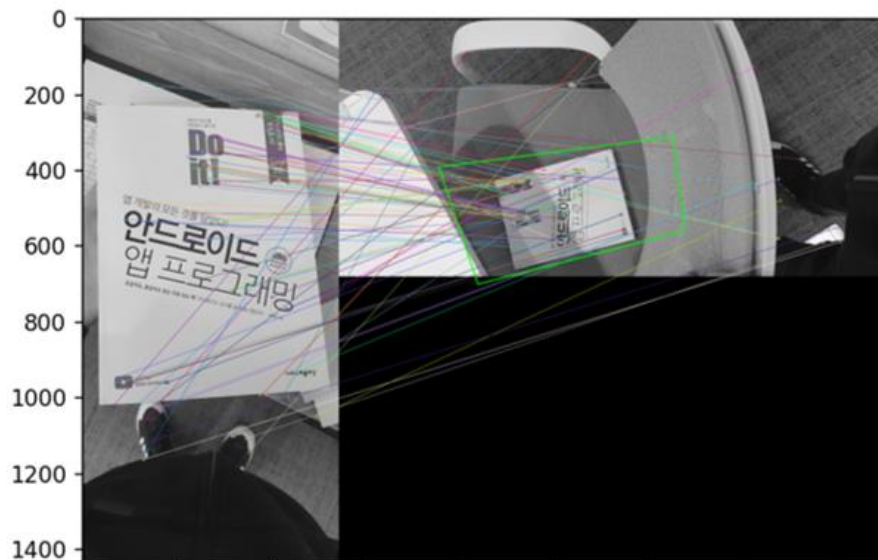


챌린지 인증

- >> 이미지를 통해 자신의 챌린지 수행을 인증
- >> 피드 작성 후 공유

피드를 통한 상호작용

- >> 피드를 통해 서로의 챌린지 수행을 확인
- >> 댓글을 통해 소통



이미지 특징점 비교

- >> OpenCV의 AKAZE 알고리즘을 통한 특징점을 추출
- >> 동일한 객체가 비교되는 사진 내부에 있는지 확인

이미지 인식

- >> Clarifai에서 제공하는 이미지 인식 Api를 활용
- >> 해당 이미지의 속성을 높은 순으로 반환



이미지 인식

```
Result = {
  "cat":0.9944410920143127,
  "cute":0.9894723296165466,
  "animal":0.9860432147979736,
  "pet":0.9856846332550049,
  "kitten":0.9674375057220459,
  "domestic":0.963958203792572,
  "mammal":0.9604588150978088,
}
```

API 호출을 통해 해당 이미지의 속성을 높은 순으로 출력

FastAPI를 활용한 이미지 처리 서버 구현

- >> 위의 이미지 처리 방식을 사용
- >> 사용자가 인증 사진을 알맞게 촬영하였는지 판단
- >> 피드를 생성하는 서버를 구현

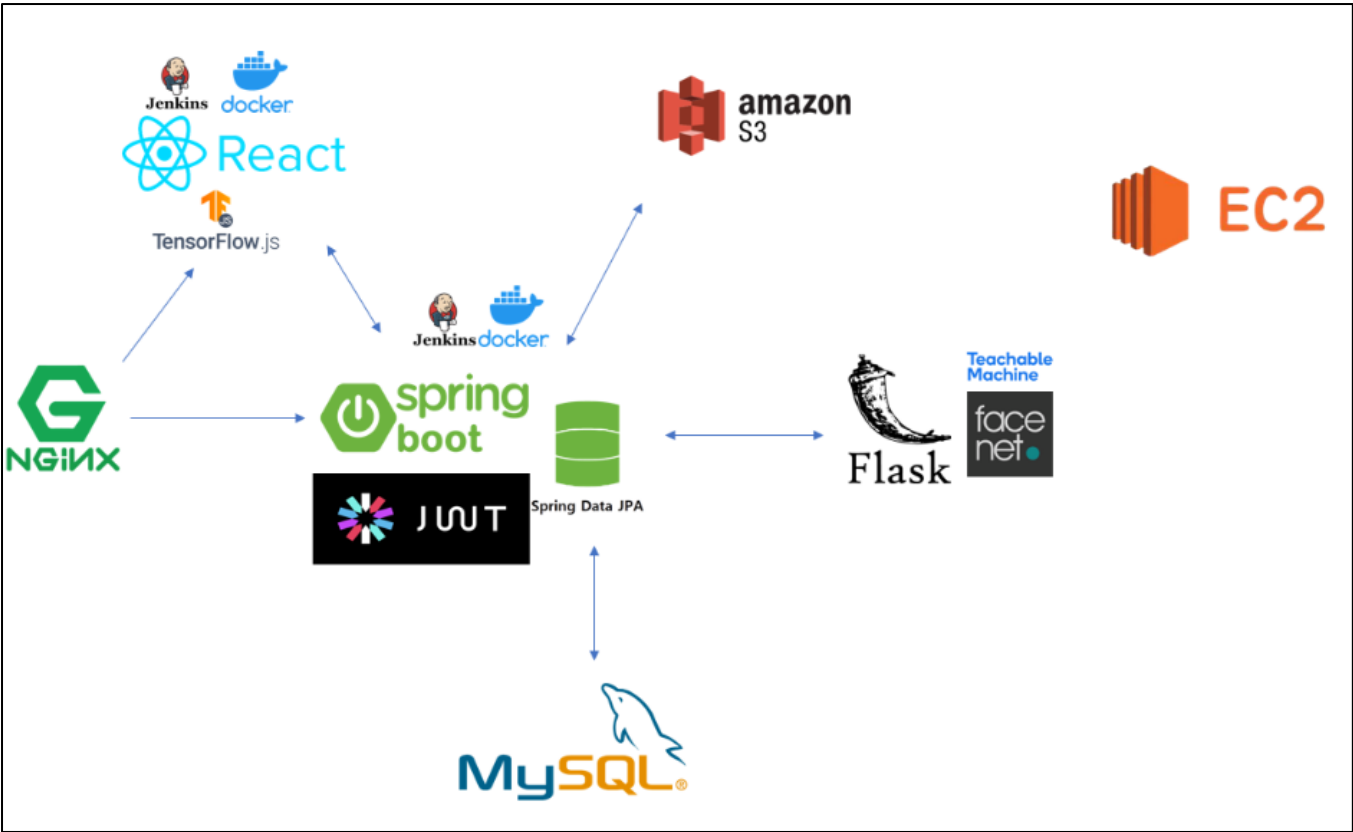
MaFya

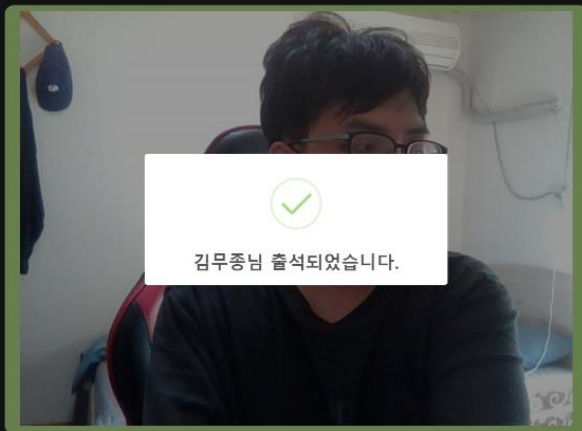
Masked ssaFYcian Attendance System

캠을 이용한 인공지능 얼굴 인식 및 마스크 여부 판단
후 간편한 출석 체크 기능,
관리자와 학생의 출결을 지원하는 서비스 시스템

기간	2022.08.22~2022.10.07
기술	React, Spring Boot, Gitlab, Jira, Docker MySQL, Node AWS, jenkins,
팀 구성	6명(Front-end 2, Back-end 4)
맡은 업무	<ul style="list-style-type: none">• Figma를 사용한 와이어프레임 제작• 마스크 착용 여부 판단 모델 구현• 전반적인 css 스타일링• 데이터 수집, Flask 서버 구축
배운 점	<ul style="list-style-type: none">• Python을 통한 데이터 분류• Agile한 업무 진행• 사용자를 위한 서비스 설계
관련 링크	https://www.notion.so/MaFya-a62ec8e0452f488fb4f031435261f113

System Architecture



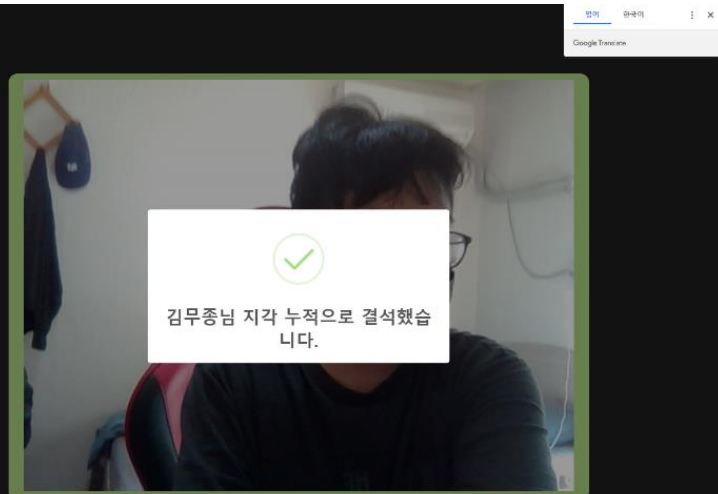


간편한 출석 체크

- » 기존의 간단하지 않은 출석 체크 과정을 개선
- » 카메라에 얼굴을 노출해 간단하게 출석 가능

인공지능 활용

- » Object Detection, Face Recognition, 마스크 인식 모델
- » 학생들이 빠르게 출석하는 것을 도움



시간과 상황에 따른 출결 처리

- » (1) 정상 출석 (2) 지각 (3) 지각 누적으로 인한 결석 (3) 조퇴 (4) 정상 퇴실을 분기 처리

음성과 텍스트를 이용한 안내

- » 음성과 텍스트를 통해 출결 과정과 결과 안내

Student Page



한 눈에 들어오는 출결 현황

- >> 본인의 출결 현황 확인
- >> 팀원의 출결을 확인, SMS 전송 가능

Admin Page



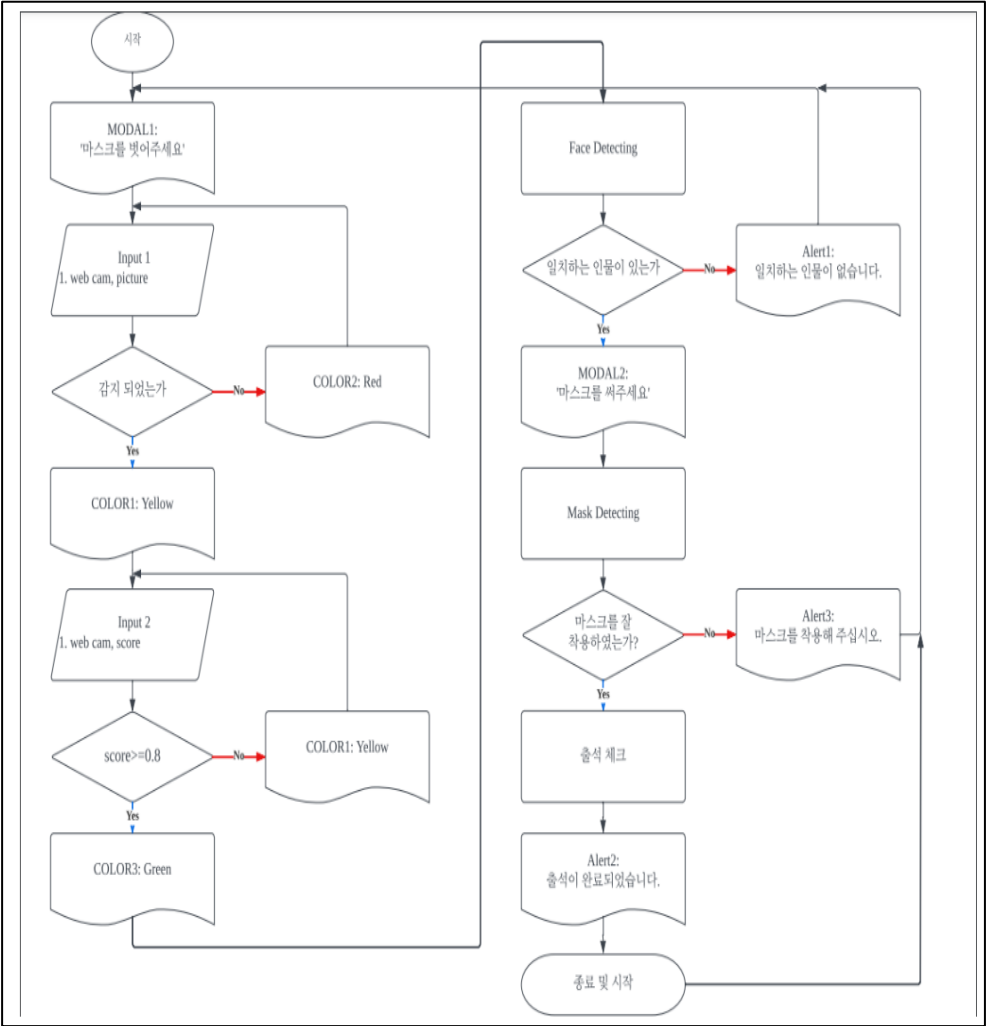
관리자용 페이지

- >> 직접 사용하게 되는 관리자들의 의견 반영

다양한 키워드를 통한 검색

- >> (1)날짜 (2)팀 코드 (3)반 (4)이름 (5)학번을 조합하여 원하는 기록 검색 가능

Flow Chart



마스크 착용 여부 판단 모델 생성

- >> 데이터 셋을 수집
- (1) 정상 착용 (2) 미착용 (3) 불량 착용으로 라벨링
- >> Teachable Machine을 활용하여 모델 생성

Flask를 활용한 이미지 처리 서버 구현

- >> 학생 출결 확인 과정 중 얼굴 인식과, 마스크 착용 여부를 판단하는 서버를 Flask를 통해 구현

React를 활용한 화면 구성

- >> 간단하고 이해하기 쉬운 UI/UX 설계

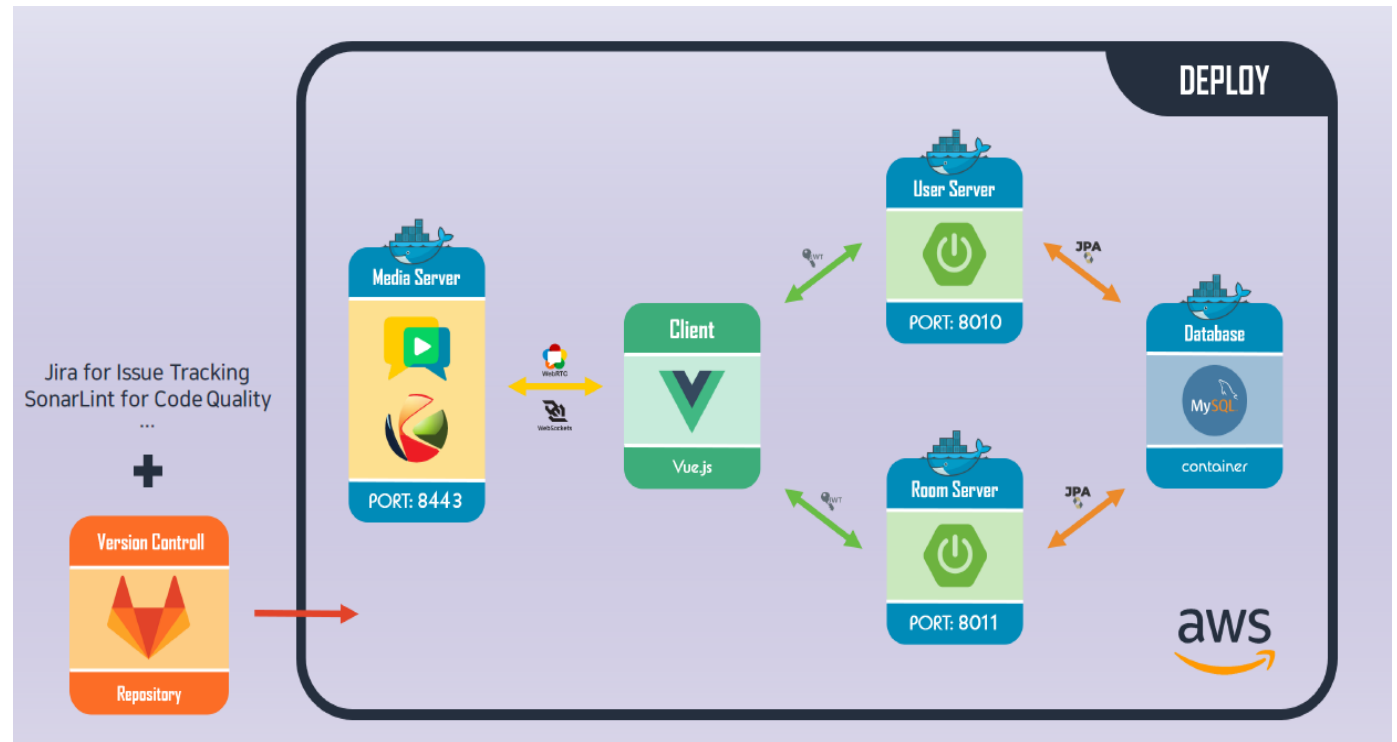
We See Play

웹 화상 게임 서비스

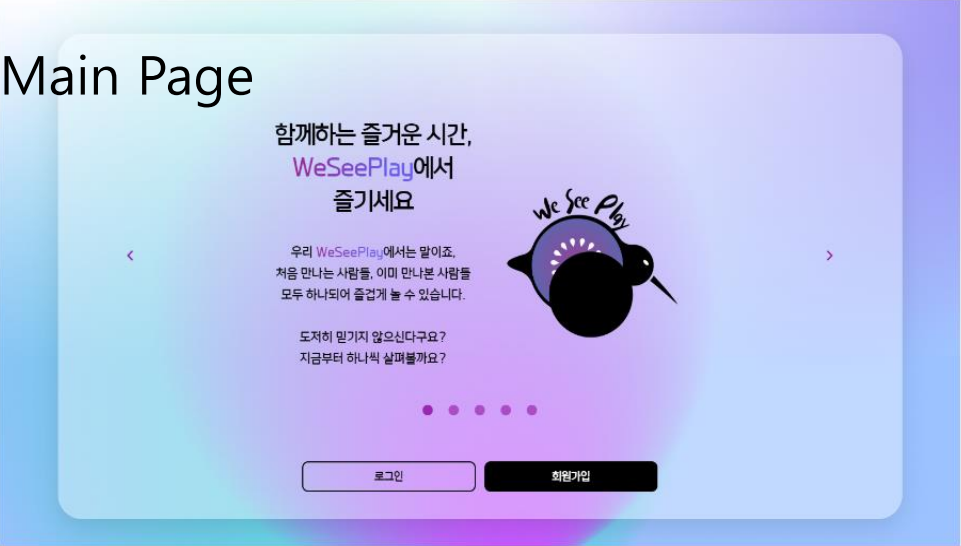
언제 어디서든 즐길 수 있는
웹 화상 게임 서비스

기간	2022.07.05~2022.08.19
기술	Vue3, Spring Boot, Gitlab, Jira, Docker, MySQL, Node, AWS, OpenVidu
팀 구성	6명 (Front-end 4, Back-end 2)
맡은 업무	<ul style="list-style-type: none"> • Front-end 중점 개발 • OpenVidu를 활용한 화상 채팅 • 게임 로직 구현
배운 점	<ul style="list-style-type: none"> • 협업을 위한 코드 관리 • 소켓을 활용한 통신 • Flow Chart 기반 로직 설계
관련 링크	https://chwon03.notion.site/9f6413568e1d48848ebf80e86d35eac3

System Architecture



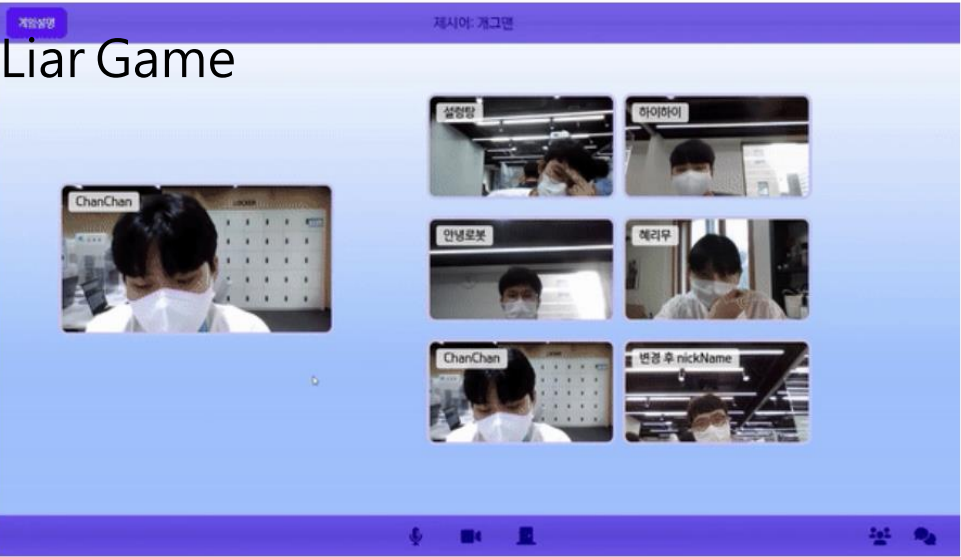
Main Page



비대면 환경

>> 여러 요인으로 직접 보지 못하는 사용자끼리 얼굴을 보며 이야기 할 수 있는 서비스 제작

Liar Game



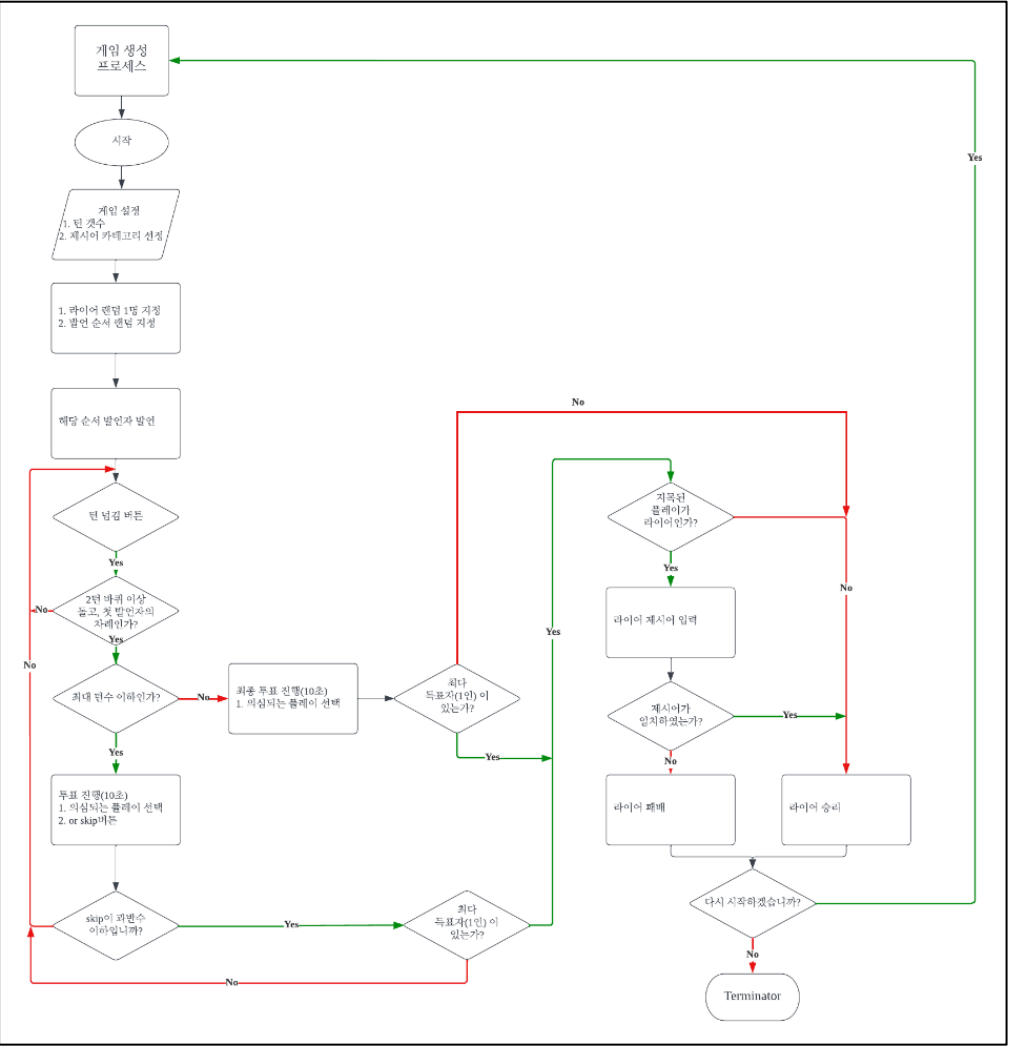
화상 정보를 이용한 게임

>> Liar Game, Call My Name과 같은 게임
>> 문자, 음성 뿐만 아니라 화상 정보들을 통해 같은 공간에서 플레이 하는 것과 같은 경험을 제공

한 눈에 들어오는 UI

>> 게임에 익숙하지 않는 사람들도 쉽게 참여할 수 있도록 설계한 UI/UX

Liar Game Flow Chart



OpenVidu를 활용한 화상 채팅구현

>> 간편하게 화상 채팅을 구현

게임 로직 설계 및 구현

- >> 각 게임의 룰에 따라 Flow Chart 제작
- >> OpenVidu의 Socket을 활용하여 게임을 구현

Vue3 를 활용한 화면 구성

- >> Angular, React에 비해 가벼운 Vue3 선택
- >> Composition Api로 script를 구성