국민대학교
전자정보통신대학
컴퓨터공학부

캡스톤 디자인 I
종합설계 프로젝트

프로젝트 명	*WeQuiz*
팀 명	*WeQuiz*
문서 제목	결과보고서

Version	1.4
Date	2024-05-15

팀원	박규연 (조장)
	김우림
	배준형
	심재민
	안금장
지도교수	허대영 교수

All rights are reserved. Reproduction in whole or in parts is prohibited without the written consent of the copyright owner.

CONFIDENTIALITY/SECURITY WARNING

이 문서에 포함되어 있는 정보는 국민대학교 전자정보통신대학 컴퓨터공학부 및 컴퓨터공학부 개설 교과목 캡스톤 디자인I 수강 학생 중 프로젝트 "WeQuiz"를 수행하는 팀 "WeQuiz"의 팀원들의 자산입니다. 국민대학교 컴퓨터공학부 및 팀 "WeQuiz"의 팀원들의 서면 허락없이 사용되거나, 재가공 될 수 없습니다.

문서 정보 / 수정 내역

Filename	수행결과보고서-WeQuiz.docx
원안작성자	박규연, 김우림, 배준형, 심재민, 안금장
수정작업자	박규연, 김우림, 배준형, 심재민, 안금장

수정날짜	대표수정자	Revision	추가/수정 항목	내 용
2024-05-11	박규연	1.0	최초 작성	초안 작성
2024-05-12	배준형	1.1	시퀀스 다이어그램	시퀀스 다이어그램 작성
2024-05-13	김우림	1.2	자기 평가, 테스트 케이스	자기평가 작성, 테스트 케이스 분류 및 서술
2024-05-14	심재민	1.3	연구 및 개발 내용, ERD 설계, 시퀀스 다이어그램	연구 및 개발 내용에 서버 기능 서술 추가, ERD 설계와 설명 작성, 시퀀스 다이어그램 수정
2024-05-15	안금장	1.4	시스템 구조 및 설계도	시스템 아키텍처 및 CI/CD 아키텍쳐 작성

All rights are reserved. Reproduction in whole or in parts is prohibited without the written consent of the copyright owner.

목 차

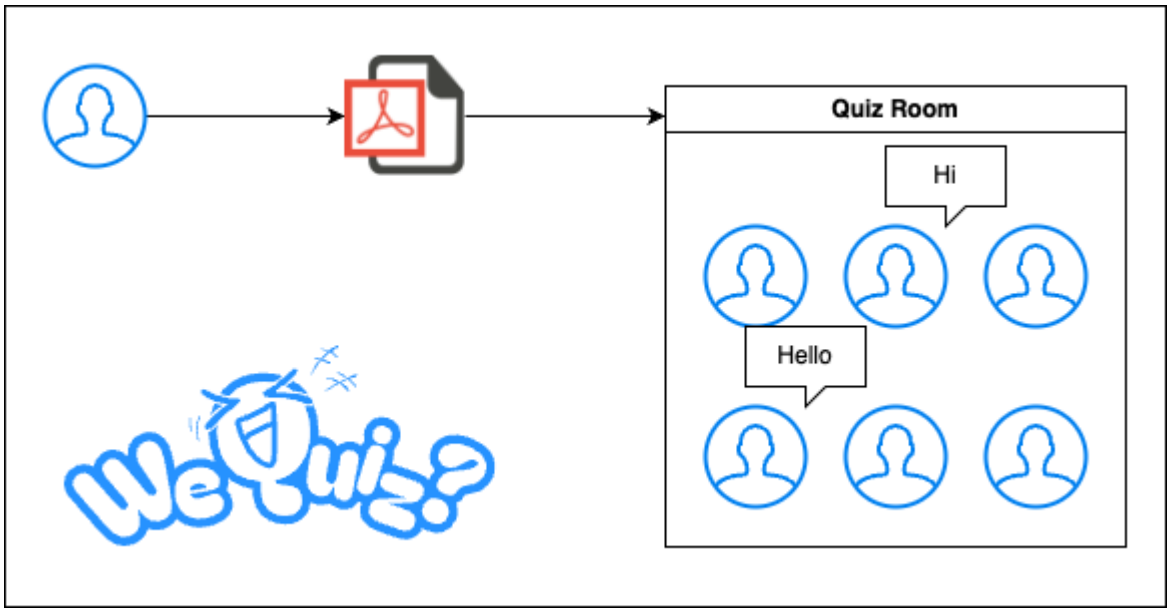
1 개요 4
1.1 프로젝트 개요 4
1.2 추진 배경 및 필요성 4
2 개발 내용 및 결과물 5
2.1 목표 5
2.2 연구/개발 내용 및 결과물 6
2.2.1 연구/개발 내용 6
2.2.2 시스템 기능 및 구조 설계도 6
2.2.3 활용/개발된 기술 6
2.2.4 현실적 제한 요소 및 그 해결 방안 6
2.2.5 결과물 목록 6
2.3 기대효과 및 활용방안 6
3 자기평가 7
4 참고 문헌 7
5 부록 7
5.1 사용자 매뉴얼 7
5.2 운영자 매뉴얼 7
5.3 배포 가이드 7
5.4 WeQuiz 매뉴얼 7
5.5 WeQuiz에 대한 기술 문서 7

All rights are reserved. Reproduction in whole or in parts is prohibited without the written consent of the copyright owner.

1 개요

1.1 프로젝트 개요

AI가 생성한 퀴즈를 친구들과 함께 푸는 서비스, <WeQuiz>를 개발합니다.
WeQuiz는 사용자가 업로드한 PDF 자료를 기반으로 AI가 퀴즈를 생성하고, 이를 친구들과 실시간으로 함께 푸는 AI기반의 멀티 플레이어 서비스입니다. WeQuiz는 사용자 친화적인 게임 형식의 UI를 통해 사용자가 손쉽게 퀴즈를 생성하고 참여할 수 있도록 유도하며, 학습 및 정보 공유 과정을 더욱 재미있고 효과적으로 만듭니다.



[그림 1] 서비스 이미지

WeQuiz는 다음과 같은 기능을 제공합니다.

- **자료 기반 퀴즈 생성 기능:** 사용자가 업로드한 PDF를 분석하여 문서의 내용에 기반한 퀴즈를 자동으로 생성합니다. 이를 통해 사용자는 문서 내용에 대한 이해나 복잡한 과정 없이 퀴즈를 만들고 풀 수 있습니다.
- **친구들과의 퀴즈 대결:** 사용자는 실시간으로 친구들과 함께 퀴즈를 풀면서 재미있게 지식을 공유하고 학습할 수 있는 기회를 갖습니다.
- **게임화된 UI:** 게임에 가까운 UI를 통해 사용자가 퀴즈 생성 및 참여 과정을 쉽고 재미있게 경험할 수 있도록 유도합니다. 이는 사용자 간의 상호작용과 참여도를 높이고, 퀴즈 풀이를

All rights are reserved. Reproduction in whole or in parts is prohibited without the written consent of the copyright owner.

하나의 즐거운 경험으로 만듭니다.

- **다양한 분야에서의 활용:** 사용자가 업로드하는 자료에 따라 학습 뿐만 아니라 단순히 재미와 오락을 위한 퀴즈도 생성할 수 있습니다.

1.2 추진 배경 및 필요성

열품타, 투두메이트 등 친구들과 '함께' 함으로써 서로에게 동기부여를 제공할 수 있는 서비스들이 많지만, 함께 같은 내용을 공부할 수 있는 서비스는 충분하지 않습니다. 이에 따라 친구들과 '함께' '같은 내용'을 공부할 수 있는 서비스를 제공하고자 했습니다.

친구들과 함께 해야하는 서비스이기 때문에 '학습'이 가능한 플랫폼이지만 간단함과 재미가 공존할 필요가 있었습니다. 이에 따라 퀴즈 플랫폼이라는 서비스를 고안하게 되었습니다. 하지만 퀴즈를 내는 당사자는 이미 퀴즈의 내용을 알고 있기 때문에 "같이" 풀 수 없다는 모순이 발생합니다. 이 문제를 해결하기 위해 AI를 도입하게 되었습니다. 이러한 배경에서 AI가 생성한 퀴즈를 친구들과 함께 푸는 서비스, WeQuiz를 개발하게 되었습니다.

All rights are reserved. Reproduction in whole or in parts is prohibited without the written consent of the copyright owner.

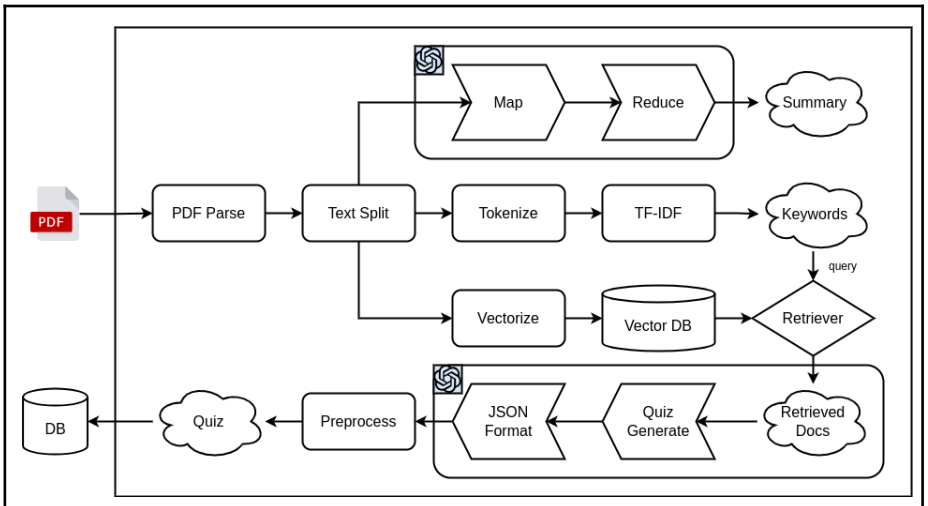
2 개발 내용 및 결과물

2.1 목표

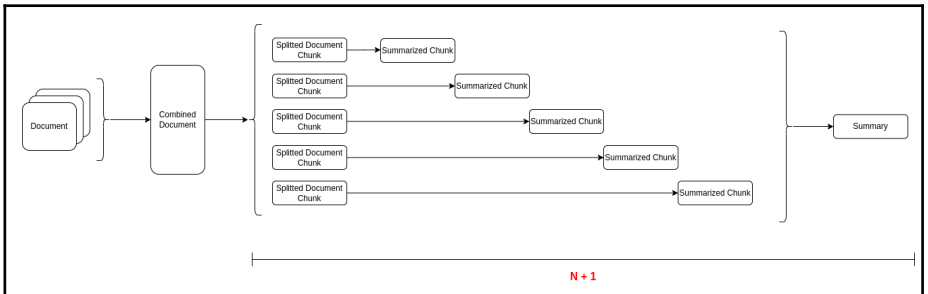
AI가 생성한 퀴즈를 친구들과 함께 푸는 서비스, <WeQuiz>를 개발합니다. WeQuiz는 누구나 간편하고 쉽게 **퀴즈를 자동 생성**할 수 있는 편의성을 제공합니다. 또한 친구들과 **실시간**으로 퀴즈를 풀 수 있는 **멀티 플레이어** 기능을 통해 사용자 간의 상호 동기부여와 즐거움을 제공합니다.

2.2 연구/개발 내용 및 결과물

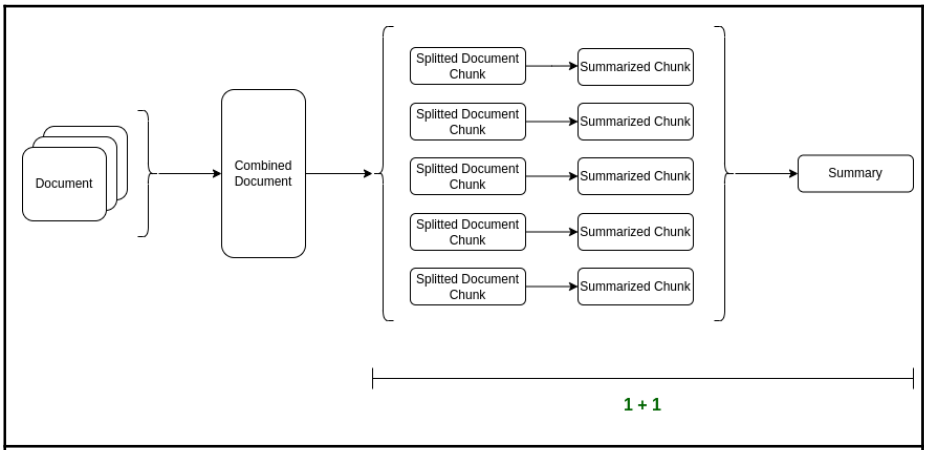
2.2.1 연구/개발 내용

번호	연구 및 개발 내용	
1	퀴즈 및 문서 요약 생성 agent	
	개요 및 결과	 [그림 2] 퀴즈 및 문서 요약 생성 agent 업로드한 PDF파일을 기반으로 자동으로 문서 요약 및 퀴즈를 생성합니다. 하나의 퀴즈 생성 요청에 대해 **4** 개의 **sub task**가 발생합니다. 각각의 상황에 적절하게 chunk 사이즈를 설정하여 text split을 진행합니다.

All rights are reserved. Reproduction in whole or in parts is prohibited without the written consent of the copyright owner.

		1. 문서 요약 문서 조각들을 각각 요약합니다. 문서 조각 요약본들을 합쳐 하나의 최종 요약본을 생성합니다. **2.** 키워드 추출 문서 조각들에 대해서 각각 TF-IDF matrix를 계산하고 matrix들을 합쳐서 하나의 최종 matrix를 생성합니다. 이 중 상위 n개의 keywords를 선정합니다. **3.** 문서 벡터화 문서 조각들을 각각 벡터화하여 FAISS에 인덱싱합니다. **4.** 문항 생성 추출한 키워드를 기반으로 FAISS에서 유사 문서 m개를 검색합니다. 검색된 문서를 프롬프트 템플릿에 추가하고 LLM에 쿼리를 날려 Quiz 문항을 생성합니다. 생성된 문항을 DB에 적재하고, 백엔드에서 사용 가능한 형태로 변형합니다. 예외로 발생할 수 있는 케이스들에 대한 처리를 거친 후 최종 완성됩니다.
2	맵리듀스, 병렬 처리를 이용한 효율적인 문서 요약	
	개요 및 결과	 [그림 3] 병렬처리 적용 전

All rights are reserved. Reproduction in whole or in parts is prohibited without the written consent of the copyright owner.



[그림 4] 병렬처리 적용 후

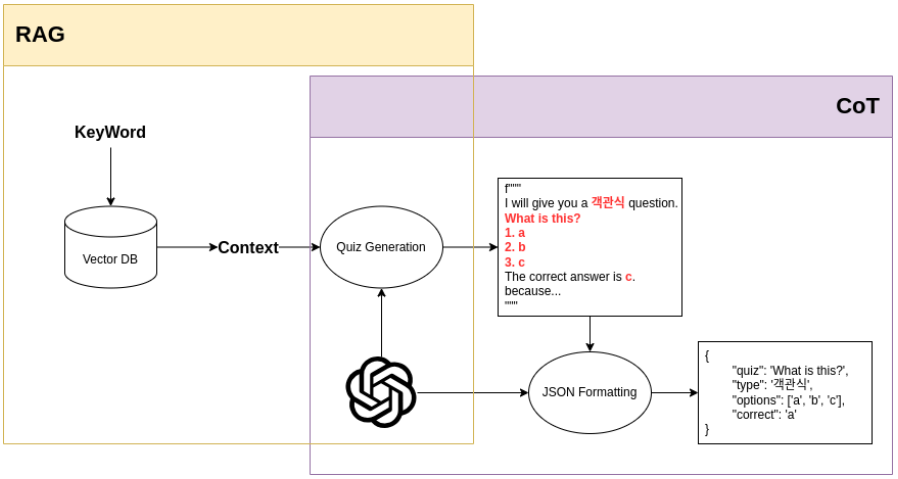
많은 양의 문서를 요약하고, 문서 요약의 질을 높이기 위해 맵리듀스 방식을 채택했습니다. 문서를 작은 조각으로 쪼개어 각각 요약 후 다시 한 번 최종 요약을 진행하는 기존의 방식에서 크게 문제가 되었던 요약 속도 문제는 각각의 요약을 비동기로 처리하여 병렬적으로 실행될 수 있도록 함으로써 해결했습니다.

'기존의 문서 조각 수 + 요약본들의 최종요약' 총 'n+1' 만큼 소요되던 요약 반환 시간을 '하나의 문서 조각 요약 + 요약본들의 최종요약' 총 '1+1' 으로 대폭 감소시켰습니다.

3	RAG + CoT = RAT 방식을 통한 문항 생성	
	개요 및 결과	문서 내용을 기반으로 문항을 생성하기 위해, LLM에게 (1) 문서의 내용을 제공하고, (2) 문항을 생성해달라는 지시를 해야합니다. 하지만 (1)에서는 너무 많은 문서의 내용을 LLM의 input으로 전달할 수 없는 문제, (2)에서는 단순히 문항 생성을 지시했을 때 GPT3.5-turbo모델이 유의미한 문항을 생성하는 경우가 드물다는 문제가 있었습니다. 이러한 문제를 극복하고 문항의 질을 향상시키기 위해서 *RAG와 *CoT를 도입하게 되었습니다. *RAG(Retrieved Augmented Generation): LLM을 이용한 추론 시, 검색 결과를 적용하여 사실적 일관성을 높이고 , 생성된 응답의 신뢰성을 향상시키며, 환각 문제를 완화시키는 방식 *CoT(Chain Of Thought): 단계적인 추론을 사용하여 LLM의 답변 퀄리티를

All rights are reserved. Reproduction in whole or in parts is prohibited without the written consent of the copyright owner.

높이는 프롬프트 엔지니어링 기법



[그림 5] 문항 생성 task 프로세스

키워드를 기반으로 VectorDB에서 해당 Context를 검색해오고,
그 Context를 프롬프트에 주입하여 관련 문항을 생성하도록 지시합니다.
이후 생성된 String을 Json Format에 맞춰 생성하도록 지시합니다.

4	실시간 소켓 통신 시스템 설계	
	개요 및 결과	Web Socket을 위에서 STOMP 프로토콜을 사용하여 메세지의 커맨드, 헤더, 바디를 지정할 수 있습니다. 커맨드가 CONNECT 일 때 Spring에서 인터셉터가 메세지의 헤더에서 JWT 토큰을 가져옵니다. 이를 서버에서 인증한 이후 연결된 세션에 담아 유저를 구분할 수 있습니다. 또한 커맨트가 DISCONNECT 될 때 이벤트 리스너를 두어 퀴즈방에서 나가게 하고 관련된 정보를 삭제하여 서버 부담을 줄일 수 있게 구현했습니다.
5	실시간 퀴즈 풀이를 위한 멀티 플레이어 시스템 개발	
	개요 및 결과	1. 퀴즈 진행 전 참가자 상태 업데이트 및 채팅 서버에서 참가자 상태(입/퇴장, 준비, 채팅메시지)에 변화가 생길 때마다 메시지를 전송합니다. 클라이언트는 참가자 상태 엔드포인트를 구독하여 데이터를 수신하고,

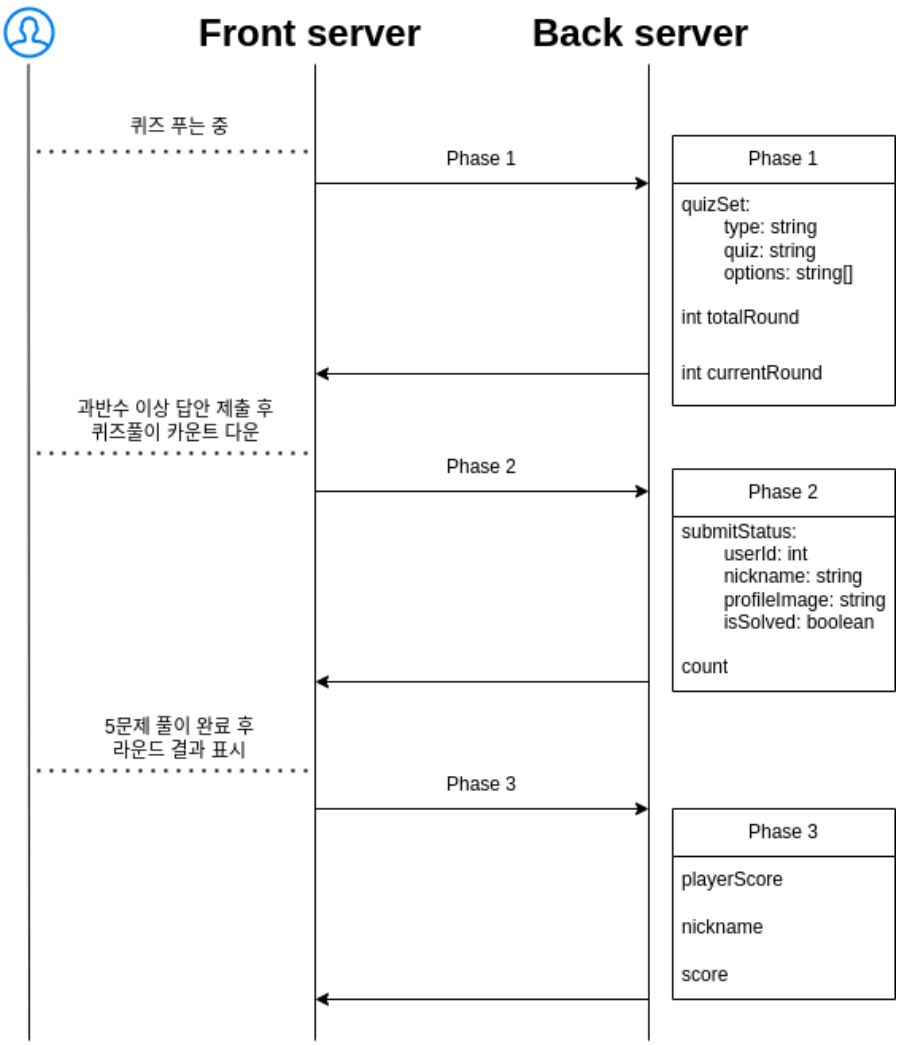
All rights are reserved. Reproduction in whole or in parts is prohibited without the written consent of the copyright owner.

참가자 목록과 상태를 업데이트해 UI에 반영합니다.

2. 퀴즈 정보 및 요약 표시
서버는 방 생성 시 ML과의 통신에서 저장한 퀴즈문서ID를 통해 DynamoDB에서 문서 요약과 퀴즈 데이터를 폴링합니다. DB에 데이터가 업데이트 되었는지 5초마다 확인하여 데이터를 읽어오면 소켓으로 클라이언트에 전달합니다. 기존에 생성한 문제 리스트는 user Id를 기준으로 DynamoDB에 쿼리를 통해 가져옵니다. 해당 유저가 만든 퀴즈의 문서 ID와 요약, 퀴즈 개수를 클라이언트에 전달해 유저가 기존에 만든 문제를 다시 풀 수 있도록 합니다.
클라이언트는 퀴즈 준비 상태와 요약 엔드포인트를 구독하고 실시간으로 생성된 퀴즈와 요약 데이터를 수신할 준비를 합니다. 이미 생성된 퀴즈 정보는 REST API 호출을 수행해 별도의 응답을 받을 수 있도록 합니다.

3. 실시간 퀴즈 풀이 진행
클라이언트는 웹소켓으로 퀴즈 데이터를 수신하고, 화면에 퀴즈 문제와 선택지를 유형에 알맞게 렌더링합니다. 사용자는 퀴즈 답안을 작성하고 제출 버튼을 클릭합니다. 서버는 모든 참가자의 답안 제출 상황을 취합하여 실시간으로 제출 현황 데이터를 브로드캐스트 합니다. 또, 제출 현황에 맞춰 풀이 상태(phase)를 세가지로 분류하고 각 상태에 따라 새로운 퀴즈, 카운트다운, 중간결과 데이터를 해당 엔드 포인트로 전송합니다. 서버에서 퀴즈 방이 완전히 종료될 때까지 퀴즈 진행 상태를 관리하고 있기 때문에 개별 참여자의 네트워크 연결이 일시적으로 끊기더라도 재접속시 일관된 데이터를 제공할 수 있습니다. 이렇게 함으로써 개별 클라이언트의 연결 끊김이 전체 게임 진행에 영향을 주지 않고 모든 참가자에게 안정적인 퀴즈 게임 환경을 제공할 수 있습니다.

All rights are reserved. Reproduction in whole or in parts is prohibited without the written consent of the copyright owner.



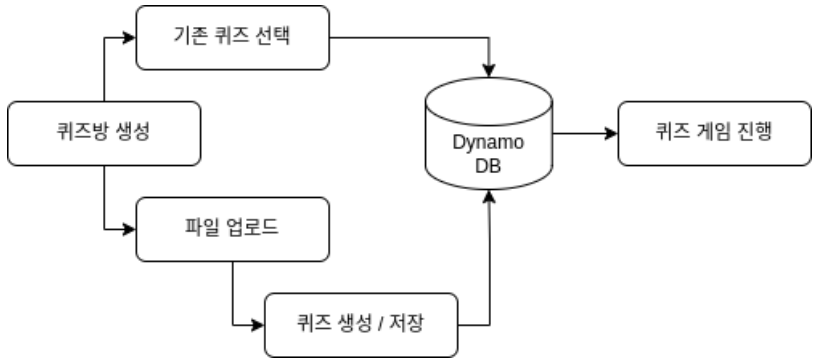
[그림 6] 새로고침 시점에 따른 phase 구분 및 phase 별 작동 매뉴얼

4. 라운드 단위 퀴즈 진행 및 채점

5문제를 한 라운드로 정하고 라운드마다 중간 점수를 보여줍니다. 중간 점수를 보여줌으로써 문제를 만들거나 채점하는 시간을 확보할 수 있고, 사용자들에게도 경쟁심을 더 불어넣을 수 있습니다.

서버에서는 라운드마다 DynamoDB에서 5문제씩 폴링합니다. 해당 라운드에 필요한 5문제가 아직 업데이트 되지 않았으면 5초마다 폴링하고, 준비되었으면 퀴즈 큐에 해당 라운드의 5문제만 담습니다.

All rights are reserved. Reproduction in whole or in parts is prohibited without the written consent of the copyright owner.

6	분기처리를 통한 서비스 사용성 확장	
	개요 및 결과	 [그림 7] 퀴즈방 생성 분기처리 파일로 새로 퀴즈를 생성하는 옵션 뿐만 아니라 기존에 생성했던 퀴즈를 재사용 가능하도록 옵션을 추가하여 서비스 사용성을 확장했습니다.
7	middleware를 활용한 로그인 가드	
	개요 및 결과	Next.js 환경에서, 특정 페이지에 로그인하지 않고 접근할 경우를 차단하는 기능이 필요했습니다. middleware를 활용하여 서버사이드에서 페이지가 렌더링 되기 전에 로그인 여부를 판단하였고, 불필요한 렌더링과 api 호출을 줄일 수 있었습니다.
8	커스텀 예외 처리 핸들러	
	개요 및 결과	서버에서 에러가 나는 경우 프론트에 예외 메세지를 정해진 규격에 맞추어 보내기 위해 기능을 구현했습니다. 서버에서는 예외 발생 시 정확한 에러 발생 사유를 추적할 수 있고, 프론트는 정해진 규격에 맞는 메세지를 받을 수 있기에 에러의 정확한 내용과 상태코드 등을 받으며 유저들에게 적절한 에러를 반환할 수 있습니다.

All rights are reserved. Reproduction in whole or in parts is prohibited without the written consent of the copyright owner.

2.2.2 시스템 기능 요구사항

번호	시스템 기능 요구사항	여부(완료/변경/미완료)
1	PDF를 업로드하면 문서의 내용을 기반으로 자동으로 퀴즈를 생성합니다.	완료
2	문서의 내용을 자동으로 요약합니다.	완료
3	메인 로비에서 실시간으로 활성화 되어있는 퀴즈 대기방 리스트가 반영됩니다.	완료
4	퀴즈 대기방에서 실시간으로 참여자들이 채팅할 수 있습니다.	완료
5	퀴즈 대기방에서 실시간으로 참여자들의 준비 상태가 반영됩니다.	완료
6	퀴즈 진행 시, 실시간으로 모두가, 동시에 같은 문항을 제공받으며, 답안을 제출합니다.	완료
7	퀴즈 진행 시, 실시간으로 참여자들의 답안 제출 상태가 반영됩니다.	완료
8	퀴즈 진행 시, 각 라운드마다 중간 성적표가 산출됩니다.	완료
9	퀴즈 종료 시, 각 참여자들은 각각 자신의 최종 성적표 페이지로 이동하여 채점 결과를 확인합니다.	완료
10	마이페이지에서 내가 참여한 퀴즈 리스트를 확인하고, 클릭시 채점 결과를 보여줍니다.	완료

All rights are reserved. Reproduction in whole or in parts is prohibited without the written consent of the copyright owner.

2.2.3 시스템 비기능(품질) 요구사항

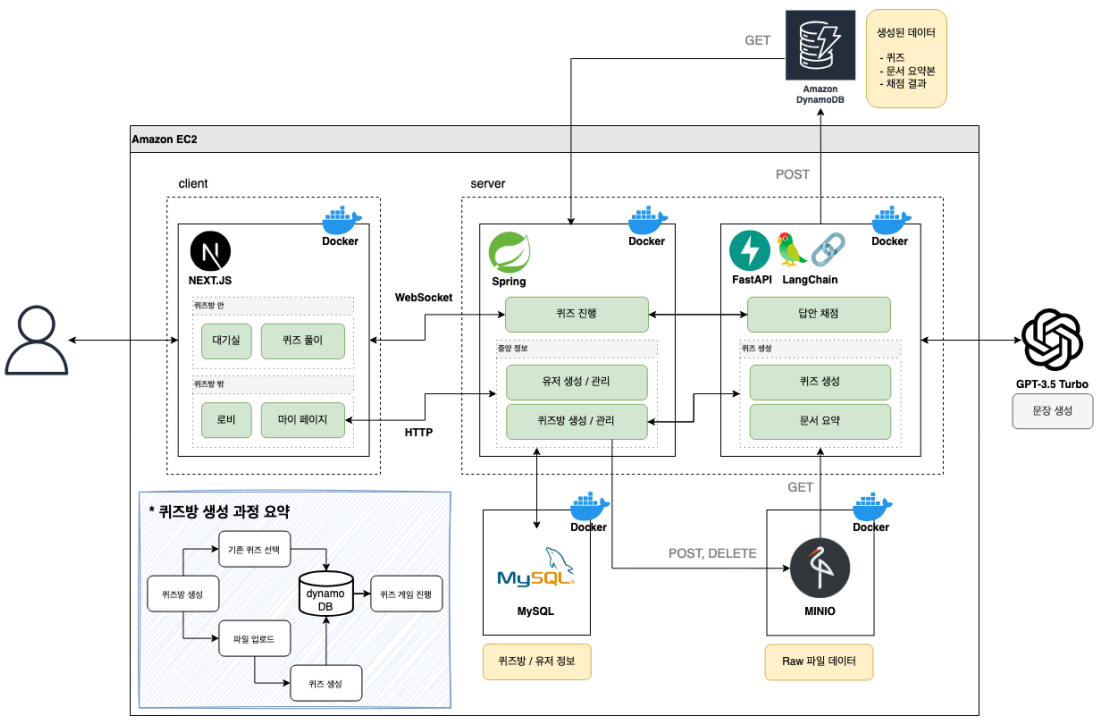
번호	시스템 비기능(품질) 요구사항	여부(완료/변경/미완료)
1	문서의 내용을 판단할 수 있는 수준의 품질로 문서 요약을 생성합니다.	완료
2	퀴즈 대기방에서 대기하는 중에 문서에 대한 내용을 파악할 수 있도록 빠른 속도로 문서 요약을 생성합니다.	완료
3	퀴즈 시작 및 퀴즈 진행 중간에 대기 시간이 발생하지 않도록 빠르게 퀴즈를 생성합니다.	완료
4	LLM의 환각현상이 개입하지 않고, 문서의 내용을 기반으로 퀴즈를 생성합니다.	완료
5	한 문서에 대해서 최소 5개, 최대 15개의 문항을 생성하면서 퀴즈의 중복이 최소화 되도록합니다.	완료
6	퀴즈의 형태가 정해진 형태를 준수하도록 합니다.	완료
7	퀴즈를 푸는 의미가 있는 수준의 품질로 퀴즈를 생성합니다.	완료
8	태블릿, 모바일에 대응하도록 반응형 UI를 제작합니다.	미완료
9	CI 과정에서 Git Actions에서 캐시를 이용해 도커 빌드 시간을 최적화 합니다.	완료

2.2.4 시스템 구조 및 설계도

- Front-end 서버: 클라이언트, 백엔드와 http, socket 통신하며 각 페이지에 필요한 정보 Back-end에 요청
- Back-end 서버: 퀴즈방/유저 정보 조회 및 관리, 퀴즈 진행, Front-end 서버와 ML 서버와 통신
- ML 서버: 퀴즈 생성, 문서 요약, 답안 채점 기능 제공, 백엔드와 소통

2.2.4.1. 시스템 아키텍처

All rights are reserved. Reproduction in whole or in parts is prohibited without the written consent of the copyright owner.



[그림 8] 시스템 아키텍처

1. 하나의 EC2 서버 사용

   언제나 접속 가능한 서비스여야 하기에 서버의 상시 구동을 위해 EC2 상에 서버를 배포했습니다. 또한 사용자가 몰리거나 서버의 용량이 더 필요할 때도 쉽게 스펙을 올릴 수 있습니다. 비용 절감을 위해 가격이 합리적인 EC2 인스턴스를 하나만 사용했습니다.

2. Docker Container로 각 서버의 환경 분리

   하나의 Instance에 프론트, 백, ML, DB 등을 모두 띄워야하는 상황이었기 때문에 각각 독립적인 환경이 필요했습니다. 이를 위해 프론트, 백, ML, MySQL, MinIO 서버를 각각의 Docker Container로 독립 환경을 구성했습니다. 이를 통해 하나의 인스턴스 안에서도 각각 독립적인 개발과 배포를 진행할 수 있었습니다.

3. Credential keys 관리의 중앙집중화

   프론트, 백, ML 서버 모두 사용하는 개발 언어와 프레임워크가 다르기 때문에 각각의 방식으로 credential keys를 관리했습니다. 이는 각 서버를 통합하고 배포하는 과정에서 생산성을 크게 해치는 부분이었습니다. 이를 해결하기 위해서 AWS Secret Managers를 이용하여 보안이 필요한 모든 키를 중앙에서 관리할 수 있도록 변경하였습니다. 결과적으로 시스템에서 사용 중인 모든 credential keys 중앙에서 관리함으로써 추적이 용이해지고, 보안이 철저해졌으며, 시스템 통합 과정에서 생산성을 극대화 할 수 있었습니다.

4. OpenAI의 GPT3.5 모델 사용

All rights are reserved. Reproduction in whole or in parts is prohibited without the written consent of the copyright owner.

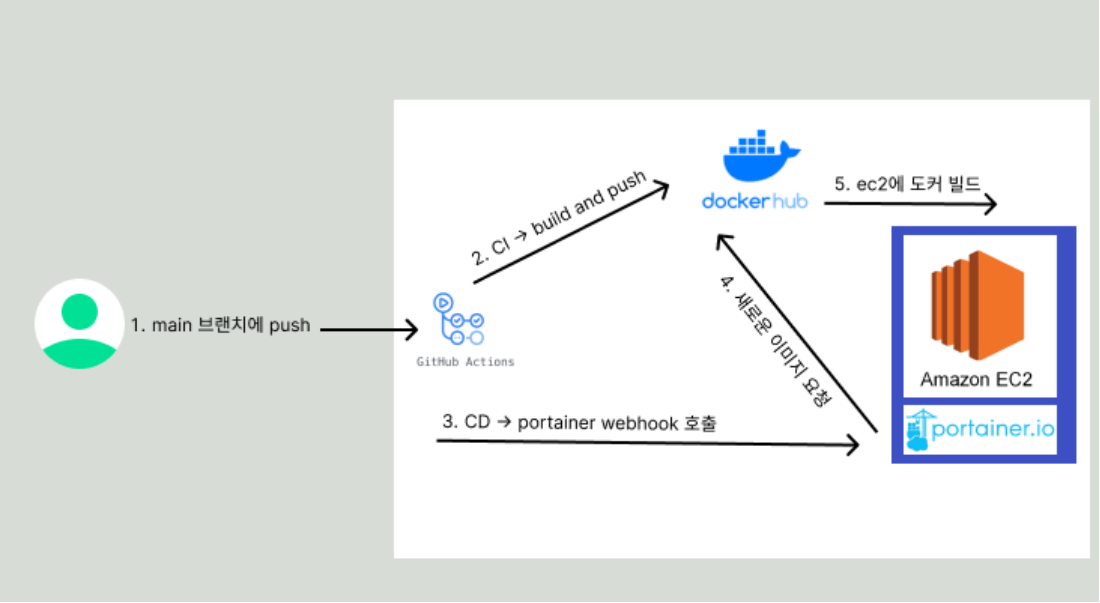
해당 서비스는 AI를 활용하여 퀴즈를 생성하는 서비스이기 때문에 퀴즈를 생성할 수 없으면 서비스를 운영할 수 없는 Single Failure Point가 있었습니다. 이를 극복하기 위해서는 문장을 생성하는 안정적인 모델을 필요로 했고 따라서 OpenAI의 GPT3.5-turbo 모델을 사용하게 되었습니다. GPT4 모델은 비용 문제로 사용할 수 없었습니다.

5. DB, 스토리지 접근 권한에 대한 역할 분리

기존에는 클라이언트에서 스토리지로 파일을 업로드하는 방식이었으나, 클라이언트에서는 DB, 스토리지에 접근하지 않고, 반드시 서버에서만 접근이 가능하도록 접근을 제한했습니다. 이를 통해 장애 발생 시 일관성 있게 백엔드에서 대응할 수 있게 되었습니다.

서버는 내부적으로 웹 백엔드 서버와 ML 서버로 분리되기 때문에 서버 내에서도 웹 백엔드 서버와 ML 서버의 db 접근 권한을 확실하게 분리하여 장애 대응, 동시성 문제를 예방하였습니다. ML서버에서는 DynamoDB에 push, MinIO에서는 get 요청만 하며, 웹 백엔드 서버에서는 DynamoDB에서는 get, MinIO에는 push와 delete만을 진행합니다. 또한 MySQL의 접근은 웹 백엔드 서버에서만으로 제한합니다.

2.2.4.2. CI/CD



[그림 9] CI/CD 아키텍처

CI/CD를 적용하여 소프트웨어의 빌드 및 배포를 자동화합니다. main 브랜치에 새로운 변경사항이 push되면 도커 이미지를 빌드하고 docker hub에 이미지를 업로드하는 ci git action이 발생합니다. ci가 완료되면 cd git action에서는 portainer webhook을 호출합니다. portainer는 docker hub에 새로운 이미지를 요청합니다. 이 새로운 이미지는 ec2 상에 컨테이너로 배포됩니다.

All rights are reserved. Reproduction in whole or in parts is prohibited without the written consent of the copyright owner.

2.2.4.3. ERD 설계

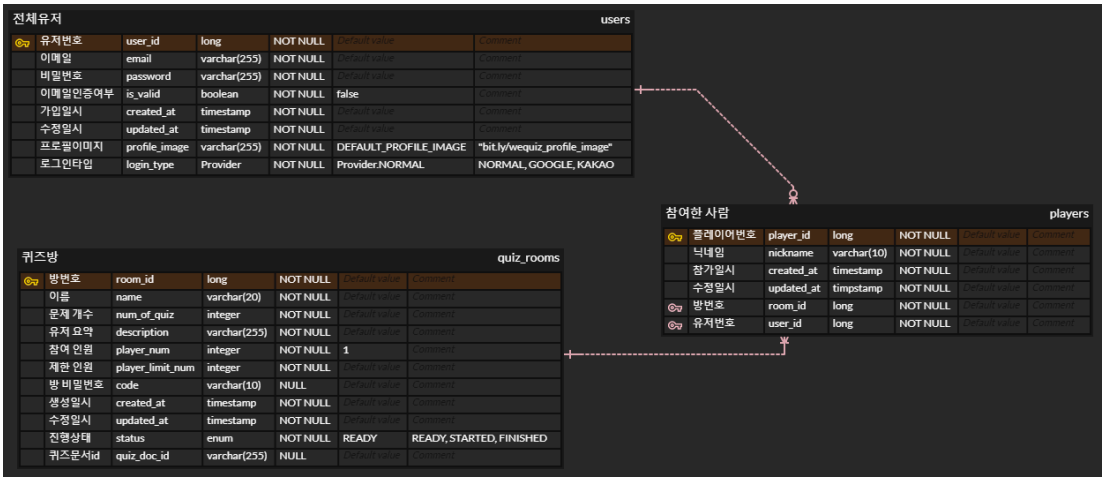
MySQL에는 유저를 관리하는 users 테이블, 퀴즈룸을 관리하는 quiz_rooms 테이블, 게임에 참가한 모든 플레이어 정보를 관리하는 players 테이블이 있습니다.

모든 테이블은 JPA의 @GeneratedValue를 사용해 기본키를 생성했고, @EnableJpaAuditing을 사용해 생성일자와 수정일자를 자동으로 저장할 수 있게 하였습니다.

users 테이블에 저장되는 정보는 이메일과 비밀번호, 이메일 인증 여부, 프로필이미지 링크(또는 파일 이름), 일반 회원인지 소셜 회원인지를 나타내는 로그인 타입이 있습니다.

quiz_rooms 테이블에 저장되는 정보는 방 이름, 문제 개수, 유저가 작성한 방 설명, 플레이어 수, 최대 플레이어 수, 방 코드, 준비 중/게임 중/게임 완료를 나타내는 방 상태, 문서ID가 있습니다.

players 테이블은 users 테이블의 기본키인 user_id와 quiz_rooms 테이블의 기본키인 room_id를 외래키로 받습니다. players 테이블에는 방에 입장할 때 입력한 닉네임이 저장됩니다.

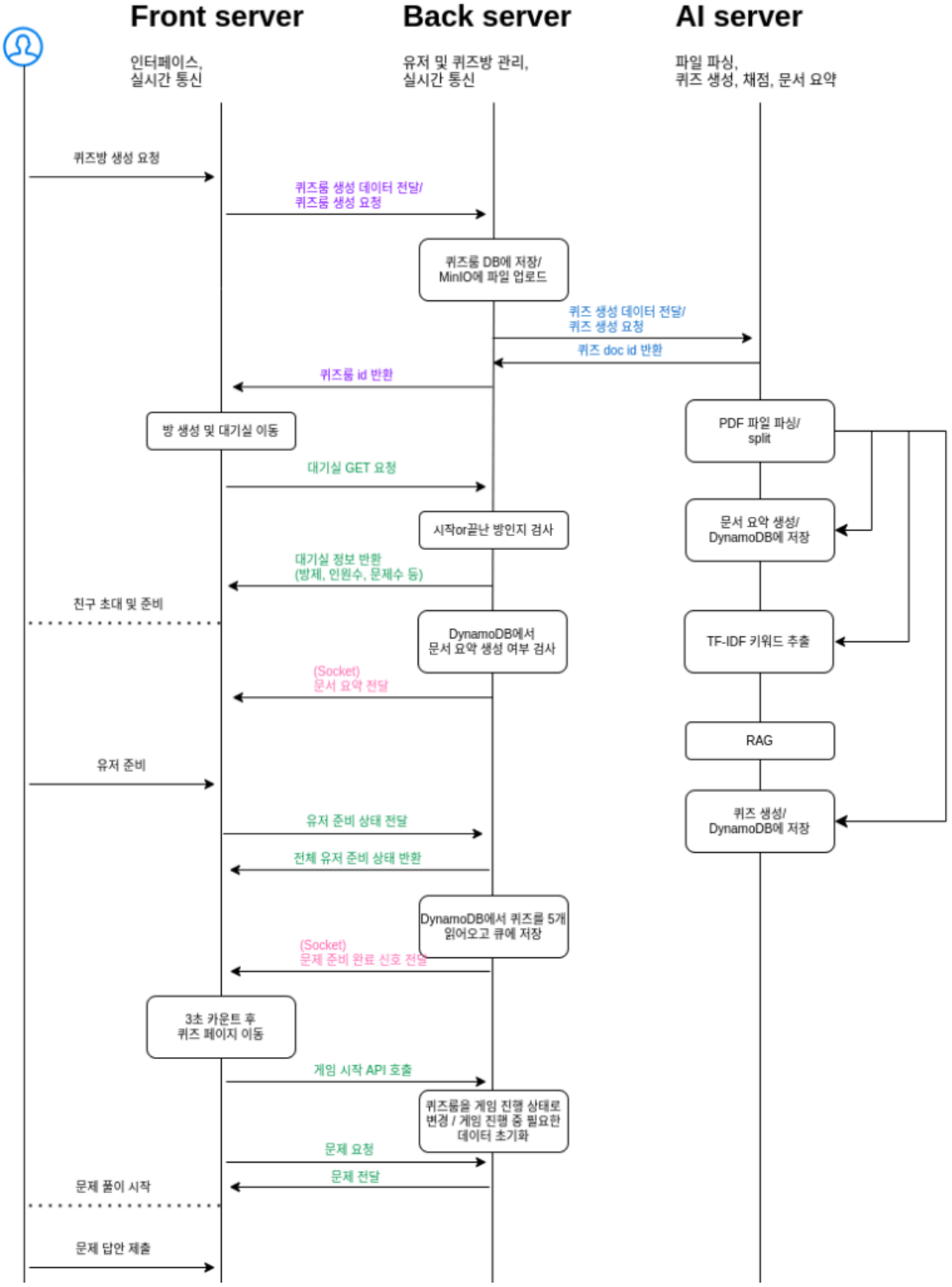


[그림 10] ERD 다이어그램 (MySQL)

2.2.4.4. 시퀀스 다이어그램

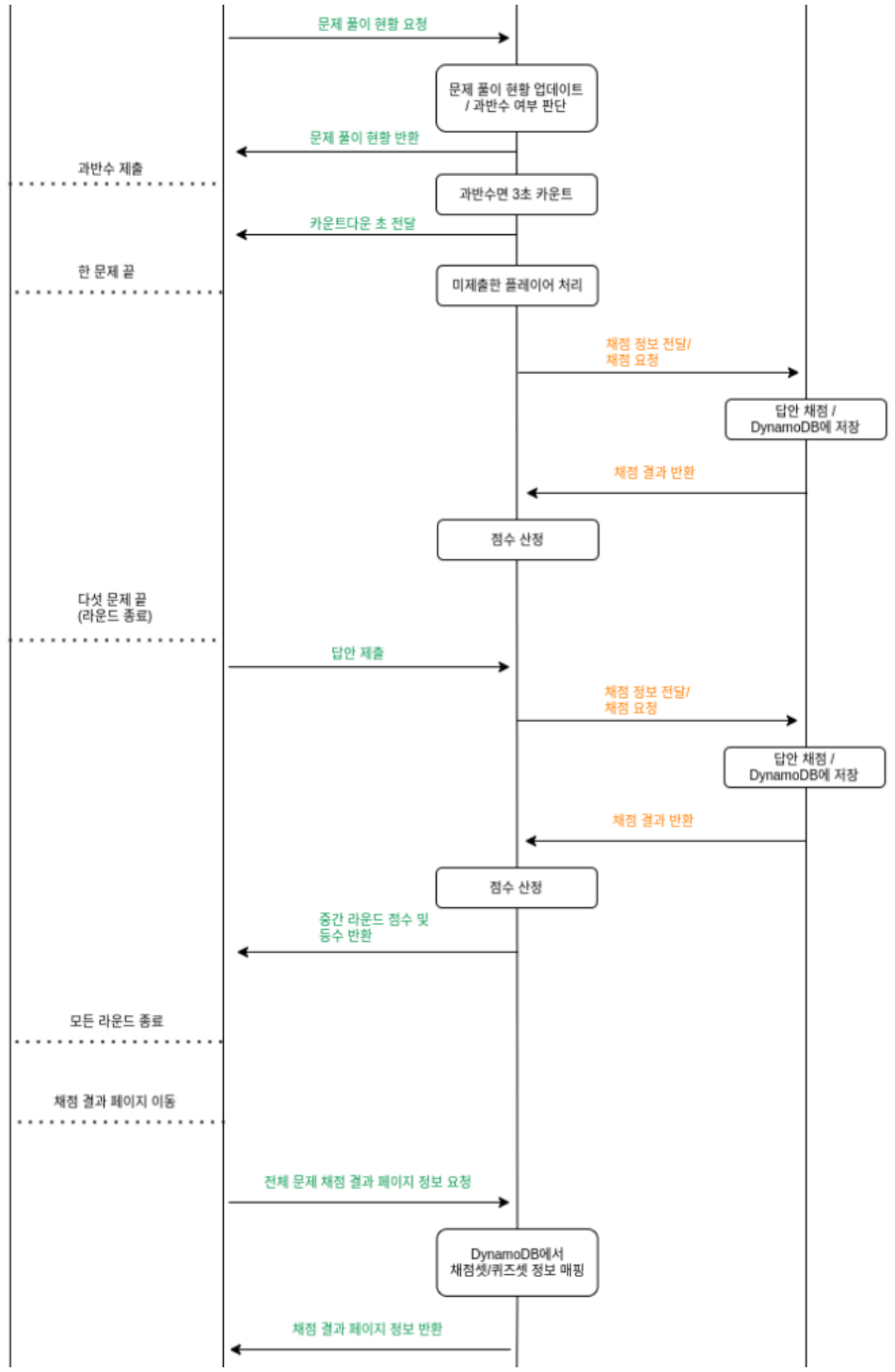
해당 서비스의 작동 방식은 아래 시퀀스 다이어그램을 통해 확인할 수 있습니다.

All rights are reserved. Reproduction in whole or in parts is prohibited without the written consent of the copyright owner.



[그림 11] 시퀀스 다이어그램(1/2)

All rights are reserved. Reproduction in whole or in parts is prohibited without the written consent of the copyright owner.



[그림 12] 시퀀스 다이어그램(2/2)

All rights are reserved. Reproduction in whole or in parts is prohibited without the written consent of the copyright owner.

2.2.5 활용/개발된 기술

2.2.5.1. ML Server

- **FastAPI:** 최근 가장 고성능(Node, Go랑 비슷한 성능)이면서 비동기 프로그래밍을 지원하며, 대규모 트래픽 처리에 유용한 파이썬 백엔드 프레임워크입니다. Django 대비 가볍고, Flask 대비 기능이 많아 ML 서버로 많이 활용됩니다. 자동 API 문서화 기능을 제공합니다.
- **LangChain:** LLM을 사용하여 애플리케이션 생성을 단순화하도록 설계된 프레임워크입니다. CoT(ChainOfThought) 방식의 프롬프트 엔지니어링과 RAG(Retrieval-Augmented-Generation)을 구현하기 용이합니다.

2.2.5.2. Server

- **Spring Boot:** 자바 기반의 프레임워크로 백엔드 애플리케이션을 구축할 수 있습니다. 국내에서 가장 많이 쓰이는 프레임워크 중 하나이며, 국내뿐 아니라 세계적으로 스프링 생태계가 크게 형성되어 있어 다양한 라이브러리도 쓸 수 있습니다. DI(Dependency Injection), IOC(Inversion Of Control), AOP(Aspect Oriented Programming), PSA(Portable Service Abstraction) 의 특징을 가지며 객체지향프로그래밍을 기반으로 합니다. 또한 복잡한 설정 없이 미리 구성된 스타터 키트를 통해 모듈을 쉽게 관리할 수 있습니다. 또한 내장된 서버를 사용하여 쉽게 배포하고 테스트할 수 있어 효율성을 향상시킬 수 있습니다.
- **Spring Data JPA:** Spring Data JPA는 Spring Framework에서 제공하는 모듈 중 하나로, Java Persistence API (JPA)를 사용하여 ORM 기능을 이용할 수 있어서 데이터베이스 액세스를 간소화하고 추상화하는 기능을 제공합니다.

2.2.5.3. Client

- **Next.js:** 최근 프론트엔드 생태계에서 가장 주목받고 있는 프레임워크입니다. Server-Side Rendering을 활용할 수 있으며, App router 등 편리한 기능을 활용할 수 있습니다. 저희는 SSR을 했을 때 이점으로 가져올 수 있는 SEO 최적화를 목표로 해당 프레임워크를 도입하였습니다.
- **Zustand:** 로그인 여부, Access Token을 클라이언트 메모리, 전역 상태로 관리하기 위해 도입하였습니다. 현업에서 많이 사용되는 Redux와 동일한 FLUX 패턴을 사용하며 강력한 기능을 제공하지만 러닝커브가 비교적 낮습니다.
- **Storybook:** 컴포넌트 단위로 테스트해볼 수 있는 라이브러리 입니다. 컴포넌트를 테스트하기 위해 의미없는 유즈케이스를 로컬에서 반복하며 체크해보아야 하지만, 이 라이브러리 도입으로 개별 컴포넌트를 바로 테스트 해볼 수 있고, 이를 통해 생산성을 크게 향상시킬 수 있었습니다.
- **mkcert:** 가상인증서를 발급해 로컬에서 https 환경에서 테스트할 수 있게 해주는 패키지입니다.

All rights are reserved. Reproduction in whole or in parts is prohibited without the written consent of the copyright owner.

실제 배포 환경과 동일한 네트워크에서의 동작을 테스트해볼 수 있기 때문에 도입하였습니다.

2.2.5.4. DB

- **MySQL:** 가장 널리 사용되는 오픈 소스 RDBMS 입니다. 웹 어플리케이션에서 강력한 성능을 제공하며, ACID를 보장합니다. PostgreSQL에 비해 가볍고 빠르기 때문에 도입했습니다.
- **DynamoDB:** AWS에서 제공하는 완전 관리형 NoSQL 데이터베이스 서비스입니다. 빠른 고성능과 무한한 확장성을 자랑합니다. ML 서버에서 퀴즈 및 디스크립션을 저장, 퀴즈 채점 결과를 저장하는데 쓰입니다. BE 서버에서는 DynamoDB에 접근하여 ML서버에서 저장한 정보를 가져와 데이터를 처리합니다.

2.2.5.5. Storage

- **MinIO:** 고성능, 오픈 소스 객체 스토리지 서비스로 분산 스토리지 아키텍처를 구현하기 위해 도입했습니다. AWS S3와 비교했을 때 외부의 스토리지를 이용하는 것이 내부에서 이용할 수 있게 합니다. 또한 무료로 이용할 수 있습니다.

2.2.5.6. Infra

- **EC2:** AWS에서 제공하는 확장 가능한 컴퓨팅을 제공하는 서비스입니다. 가상 서버 인스턴스를 쉽게 구성하고 관리할 수 있어서 도입했습니다. 클라우드 스펙을 쉽게 확장할 수 있기에 서비스 요구사항에 따라 스펙을 변경할 수 있다는 점에서 이용했습니다.
- **Docker:** 애플리케이션을 컨테이너화하여, 의존성과 설정 문제 없이 일관되게 동작하게 만드는 오픈 소스 플랫폼입니다. 팀원들의 다양한 개발 환경에서 테스트가 용이하기 위해, 배포를 편리하게하며 자동화까지 하도록 구성하기 위해 사용했습니다.
- **AWS Secret Managers:** AWS 에서 제공하는 서비스로, 팀원들 간 환경변수를 따로 공유하지 않고도 편리하고 안전하게 관리할 수 있도록 이용했습니다. API 키, DB 소스 등 민감한 정보를 저장합니다.

2.2.5.7. CI/CD

- **Git Actions:** Github에서 제공하는 CI/CD 도구로, 소프트웨어 개발 워크플로우를 자동화하여 빌드, 테스트, 배포 과정을 간소화합니다. 클라이언트, 서버, ML 서버의 워크플로우 파일을 개별 작성하여 각 디렉토리에 변화를 감지했을 때 실행됩니다.
- **Portainer:** Docker 환경 관리를 단순화하는 오픈 소스 툴입니다. Docker 를 위한 그래픽 관리 인터페이스를 제공하며, 컨테이너를 쉽게 관리할 수 있도록 돕습니다. 또한 제공하는 webhook을 이용하여 CD(Continuous Deployment)를 구성했습니다.
- **Docker Hub:** Git Actions에서 CI를 하여 Docker hub에 빌드된 이미지를 업로드 합니다. 추후 Portainer를 통해 CD 과정이 진행될 때 Docker hub에서 이미지를 가져와 실행합니다.

2.2.6 현실적 제한 요소 및 그 해결 방안

- PDF 파일에 텍스트문서(교과서, 논문)등을 촬영한 이미지가 들어있는 경우의 텍스트 처리는

All rights are reserved. Reproduction in whole or in parts is prohibited without the written consent of the copyright owner.

어떻게 하는가?

  - 먼저 이미지 내의 텍스트도 처리할 수 있는 기능이 구현은 되어 있으나, 이미지 처리는 계산량이 많기 때문에 반드시 GPU 서버를 사용해야합니다. 하지만 비용 문제로 GPU 사용을 할 수 없었고, 현재는 이미지 속 텍스트를 처리하는 옵션을 임시 제거해두었습니다.

- 그렇다면 위의 경우에는 아무 텍스트도 나오지 않을 텐데 이런 경우는 어떻게 되는가?
  - 텍스트의 양이 너무 적거나 없어 퀴즈를 생성할 수 없는 경우 예외를 발생시키고, 퀴즈 생성을 제한합니다.
  - 텍스트 양의 측정은 퀴즈 생성 요청 시, 추출된 전체 텍스트의 토큰수를 계산하고, 토큰수가 임계치 미만이면 예외를 발생시킵니다.
- GPT3.5 사용 시, 비용 소모가 크지 않은가?
  - 24년 2월 GPT3.5-turbo 기존 모델에서 가격이 약 2-3배 가량 저렴한 버전의 모델이 출시 되었습니다. 따라서 사용자가 많지 않은 현재는 많은 비용이 발생하고 있지 않습니다.
- 사용자가 늘어난다면 비용 소모가 커지지 않는가?
  - 위에서 언급한 GPT3.5-turbo 모델 뿐만 아니라 GPT4 모델도 지난 5월 13일 4월 초 기준의 GPT4 모델보다 12배 가량 저렴하게 출시 되었습니다. LLM 모델 API는 2년 사이에 240배 가량 비용이 감소되었습니다. 이러한 감소 추세에 따라 추후 사용자가 늘어나는 속도 보다 LLM 비용 감소의 추세가 더 클 것으로 기대하고 있습니다.
  - 최근 llama3가 매우 합리적인 가격에 모델을 출시했고, LLM inference API를 제공하는 플랫폼들에서 해당 모델의 유료 API 제공을 앞두고 있습니다. 출시 후 LLM API를 수정하여 비용절감을 기대할 수 있습니다.
- 입력하는 파일의 언어가 한글이 아닐 경우에는 어떻게 하는가? 성능에 문제가 생기지 않는가?
  - multi-lingual text embedding 모델을 활용하여 임베딩 과정에서 발생할 수 있는 언어문제를 극복했습니다.
  - 언어의 문서를 감지하고, 언어에 따라 다른 토크나이저를 사용하여 키워드 추출과정에서 발생할 수 있는 언어 문제를 해결했습니다. 현재 한글이면 konlpy를, 그 외의 언어라면 nltk를 사용하고 있습니다.
- 데스크탑에서만 사용 가능한 서비스라면 접근성이 떨어지지 않나?
  - 원래는 모바일과 데스크탑, 태블릿 환경까지 반응형으로 구현하려고 하였으나, 서비스 특성 상 단순 스태틱 웹페이지가 아니기 때문에 각 뷰포트 환경 별로 페이지를 각각 구현해야 했습니다. 디자인에 들어가는 공수가 너무 많이든다고 판단하여, 꼭 필요한 부분만 우선적으로 반응형을 구현하여 해결하고 있습니다.

All rights are reserved. Reproduction in whole or in parts is prohibited without the written consent of the copyright owner.

2.2.7 결과물 목록

번호	결과물 목록	상세
1	WeQuiz 서비스	최종 서비스 배포 사이트 - https://wequiz.kr/
2	소스코드	GitHub - https://github.com/kookmin-sw/capstone-2024-01

2.3 기대효과 및 활용방안

본 서비스는 상호 간 동기부여를 제공하는 것에 그치던 기존의 서비스들의 한계에서 '재미'와 '학습'을 더했습니다. '학습'의 불편함을 **AI**를 이용한 퀴즈 자동 생성을 통해 극복하고, '학습'의 지루함을 실시간 통신이 가능한 멀티 플레이어 기능을 제공하여 극복합니다.

본 서비스는 학습 목적 뿐만 아니라 단순 재미를 목적으로 활용도 가능하기 때문에, 다양한 사용자들이 다양한 목적으로 활용할 수 있습니다. 아래 세 명의 가상인물로 사용 예시를 제안합니다.

1. 인물: 최준숙(23세)
   직업: 대학생(컴퓨터공학과 3학년)
   특징: 혼자 공부하기에 지루하고 동기부여가 안 됨, 친구들과 함께 공부하면서 재미와 경쟁 심리를 느끼고 싶어함.
   활용 및 기대효과: 친구들과 함께 강의안을 기반으로 퀴즈를 만들어 풀고, 정답을 공유하면서 게임처럼 재미있게 공부할 수 있음.
2. 인물: 호대영(40세)
   직업: 중학교 과학 선생님
   특징: 학생들의 수업 이해도와 참여도를 높이는 새로운 방법을 찾고 있음.
   활용 및 기대효과: 수업 때 다룬 내용을 WeQuiz에 업로드 후 학생들과 함께 퀴즈를 풀면서 학습 효과를 높임, 게임처럼 재미있는 UI로 학생들의 흥미와 집중력을 이끌어 냄.
3. 인물: 박하멍(25세)
   직업: 게임 스트리머(유튜브/트위치 채널 운영)
   특징: 뛰어난 게임 실력과 재치있는 입담으로 많은 구독자와 팬들을 확보하고 있음. 시청자들과 활발히 소통하여 흥미로운 컨텐츠를 만들어내는 것을 좋아함,

All rights are reserved. Reproduction in whole or in parts is prohibited without the written consent of the copyright owner.

활용 및 기대효과**:** 게임 방송 중 WeQuiz를 활용하여 실시간으로 시청자들과 게임 관련 퀴즈를 푸는 이벤트를 진행함. 퀴즈를 이용해서 시청자들과 재미있게 소통하며 방송 컨텐츠로도 활용함.

All rights are reserved. Reproduction in whole or in parts is prohibited without the written consent of the copyright owner.

3 자기평가

아래 평가 요소로 본 서비스의 사용 가능 여부를 평가합니다. 평가 지표는 세 가지(높음, 보통, 낮음)으로 나타냅니다.

- 기능성**(Functionality):** 서비스가 사용자의 요구사항을 얼마나 잘 충족하는지 평가합니다. 서비스의 기능이 사용자의 니즈를 만족시키는지, 기능이 완전하고 정확한지 등을 고려합니다.
- 사용성**(Usability):** 서비스의 사용자 인터페이스가 직관적이고 사용하기 쉬운지 평가합니다. 사용자가 서비스를 쉽게 배우고 효과적으로 사용할 수 있는지가 중요합니다.
- 성능**(Performance):** 서비스의 응답 시간, 처리량, 리소스 사용량 등을 평가합니다. 서비스가 사용자의 기대에 부응하는 성능을 제공하는지 확인합니다.
- 신뢰성**(Reliability):** 서비스가 안정적으로 작동하는지, 얼마나 자주 다운타임이 발생하는지 등을 평가합니다. 서비스의 가용성과 오류 복구 능력이 중요한 고려사항입니다.
- 보안성**(Security):** 서비스가 데이터를 안전하게 보호하는지, 인증 및 권한 관리가 적절한지 등을 평가합니다. 서비스가 산업 표준과 규제 요구사항을 준수하는지도 고려해야 합니다.
- 확장성**(Scalability):** 서비스가 사용자 수와 데이터 볼륨의 증가에 따라 원활하게 확장될 수 있는지 평가합니다. 서비스 아키텍처의 확장성과 비용 효율성이 고려사항입니다.
- 유지보수성**(Maintainability):** 서비스가 쉽게 유지보수되고 업데이트될 수 있는지 평가합니다. 코드의 품질, 문서화 수준, 지원 체계 등이 유지보수성에 영향을 미칩니다.
- 비용**(Cost):** 서비스의 가격 모델과 총소유비용(TCO)을 평가합니다. 서비스의 가격이 제공하는 가치에 상응하는지, 장기적인 비용 효율성이 있는지 등을 고려합니다.

	김우림	박규연	배준형	심재민	안금장
기능성	높음	높음	높음	높음	높음
	사용자가 기대할 법한 주요 기능들이 모두 포함되어있고 정상적으로 작동됨	실시간으로 소통이 가능하며, 실시간 진행이 가능하도록 퀴즈 생성 속도가 충분히 빠름	사용자가 실시간 퀴즈 플랫폼에 원하는 요소들을 고루 갖추고 있다고 판단함	실시간 게임형 퀴즈 서비스에 대한 주변의 수요가 높았고 서비스가 그에 대응하는 적절한 기능을 제공하고 있음	내가 가진 파일로 퀴즈를 만들어 실시간으로 친구들과 풀 수 있다는 기능을 제공하고 정상 작동함

All rights are reserved. Reproduction in whole or in parts is prohibited without the written consent of the copyright owner.

사용성	높음	높음	높음	높음	높음
	직관적이고 사용하기 쉬운 UI로 설계되어있고 조작이 간편함	UI가 사용자 친화적이고, 사용법이 매우 간편함	사용자 친화적이고 게임과 같은 쉬운 사용법으로 높은 사용성을 지녔다고 평가함	사용자 친화적으로 개발했고, 핵심 기능이 직관적으로 표현되어 있음	누구나 사용하기 쉽고 UI가 직관적이기 때문에 쉽게 접근 가능할 것이라고 생각함
성능	높음	높음	높음	높음	보통
	빠른 문제 생성과 요약, 실시간 퀴즈 풀이 및 채점이 가능함	빠르게 문제 생성 및 요약 생성이 가능함	빠르고 정확한 문제 생성 및 채점, 소켓 통신이 가능함	퀴즈 생성 속도와 서비스 속도가 충분 빠름	퀴즈와 요약이 잘 생성되지만, 아직 오류가 있는 퀴즈가 생성되는 경우가 있음
신뢰성	보통	낮음	낮음	보통	보통
	트래픽 증가로 인한 장애 상황에 대비할 수 있는 시스템이 따로 이루어지지 않아 가용성이 제한적임	아직 서버 트래픽에 따라 서버를 자동으로 스케일링할 수 있는 인프라가 구축되어있지 않고, 장애에 대응할 수 있도록 다중화가 되어있지	ML파트가 고장날 경우 매우 제한적인 기능만 사용이 가능함	아직 예상하지 못한 곳에서 버그가 많이 발생함	트래픽 증가로 인해 생길 수 있는 장애를 극복하기 위한 오토 스케일링 및 로드밸런싱이 구축되어 있지 않음

All rights are reserved. Reproduction in whole or in parts is prohibited without the written consent of the copyright owner.

		않아서 가용성이 떨어짐			
보안성	보통	높음	높음.	높음	높음
	토큰을 통해 사용자 인증 정보를 관리하고 있지만 탈퇴 기능이 따로 없고 사용자가 업로드하는 파일을 따로 검증하지 않음	중요한 정보를 저장하고 있지 않음	토큰의 분리, 클라이언트에서 각 세션과 클라이언트 메모리에 토큰을 분리하여 보관하고, 서버사이드에서 체크함으로써 보안성을 제고하였음	토큰을 통해 인증 정보를 관리하고 있음	Rest API의 특징 중 하나인 Stateless에 맞춰 JWT 토큰을 이용한 회원 인가 작업 및 FE, BE, ML 모두 SSL 인증을 통한 보안강화를 진행
확장성	높음	높음	높음	보통	높음
	공통 컴포넌트들을 재사용할 수 있도록 구축했기 때문에 향후 기능 추가나 변경 시에 용이할 것이라고 생각함	모듈화가 되어있기 때문에 기능을 확장하기 유리함	추후 미리 제작된 퀴즈, 솔로플레이 등의 활성화가 추가될 경우 퀴즈 플랫폼으로서의 역할을 할 수 있는 여지가 있음	지금은 대량의 사용자가 접속하면 감당 가능할지 잘 모르겠지만 로드 밸런싱을 적용하면 충분히 극복 가능할 것이라 생각됨	객체지향적인 설계를 통해 코드 재사용성을 높이고 SOLID 원칙을 고려하여 개발을 하였기에 기능 추가 혹은 변경 시에 용이할 것이라 생각함.

All rights are reserved. Reproduction in whole or in parts is prohibited without the written consent of the copyright owner.

유지보수성	높음	높음	높음	높음	높음
	기능별 컴포넌트 분리를 적절히 해 놓았기 때문에 효율적인 유지보수가 가능함	클래스 객체가 논리적으로 정의되어있어 유지보수하기에 용이함	UI의 컴포넌트 분리로 추후	기능에 따라 모듈화하였고, 스웨거 등 api 문서를 이용하여 유지보수하기 용이함	기능에 따른 모듈화 및 SRP를 생각하며 클래스와 메소드의 분리가 잘 이루어져 있어서 유지보수 하기 용이함
비용	보통	보통	보통	보통	보통
	AI 모델 api를 사용하는데에 비용이 들겠지만 후에 광고, 구독 서비스 등을 통해 이를 충분히 충당할 수 있을 것으로 보임	최근 GPT3.5-turbo의 저렴한 버전이 출시되기도 했고, 앞으로도 빠른 추세로 비용이 감소할 것으로 보이기 때문에 비용 문제가 심하지 않을 것이라고 예상됨	기본적으로 api 통신을 최소화하였으나, 소켓 통신에서 비정상적으로 많은 통신이 발생할 경우 문제가 생길 수 있음	모델 API 비용이 좀 들지만 후에 서비스에 광고를 건다면 유지비는 충분할 거라고 생각함	GPT api를 사용하고 있어 비용이 적지 않게 들고 있음. 추후 자체 모델을 개발하여 api 비용을 절감할 수 있을 것이라고 생각함

All rights are reserved. Reproduction in whole or in parts is prohibited without the written consent of the copyright owner.

4 참고 문헌

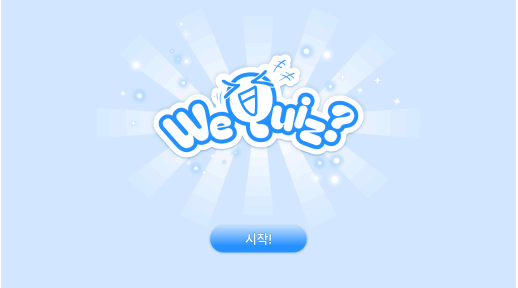
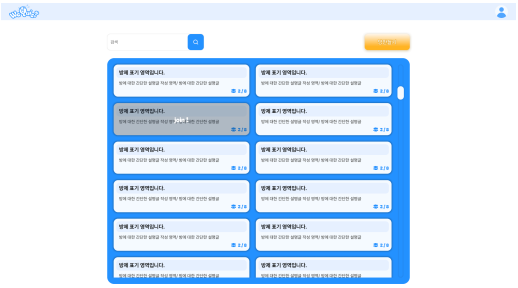
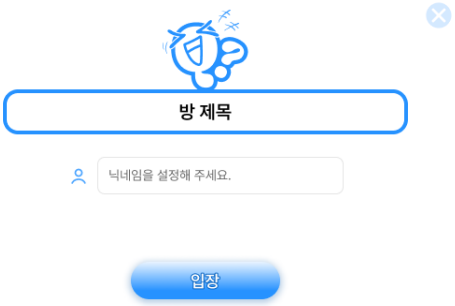
번호	종류	제목	출처	발행년도	저자
1	웹페이지	LangChain 공식문서	https://python.langchain.com/v0.1/docs	2024	LangChain
2	웹페이지	Next.js 공식문서	https://nextjs.org/docs	2024	Vercel, Inc.
3	웹페이지	AWS DynamoDB 공식문서	https://docs.aws.amazon.com/dynamodb/	2024	AWS
4	웹페이지	Spring 공식문서	https://docs.spring.io/spring-framework	2024	Spring
5	웹페이지	MinIO 공식문서	https://min.io/docs/minio/kubernetes/upstream/index.html	2020	Minio
6	웹페이지	Portainer 공식문서	https://docs.portainer.io/	2024	Portainer

5 부록

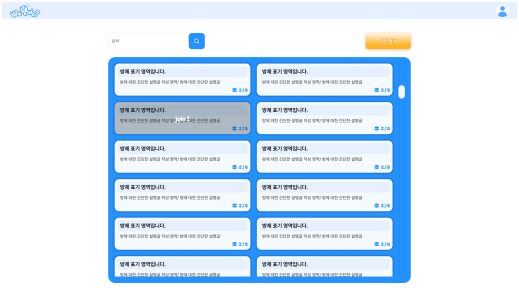
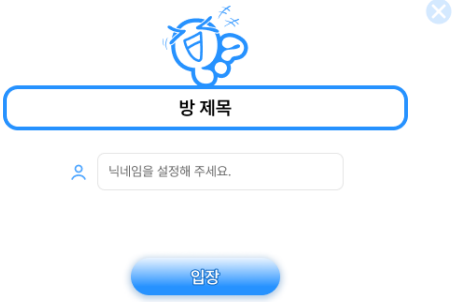
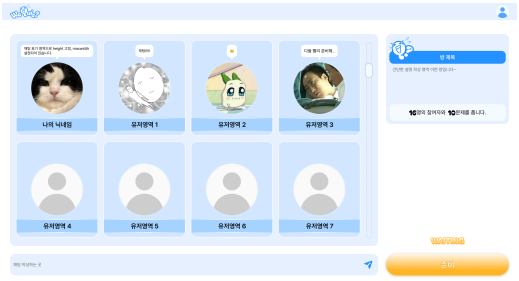
5.1 WeQuiz 매뉴얼

방 접속하기

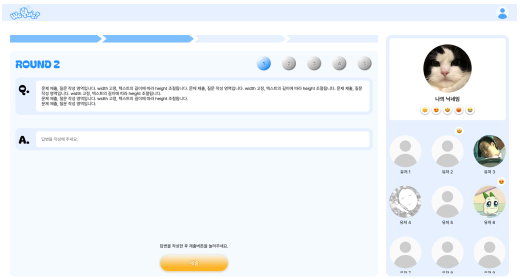
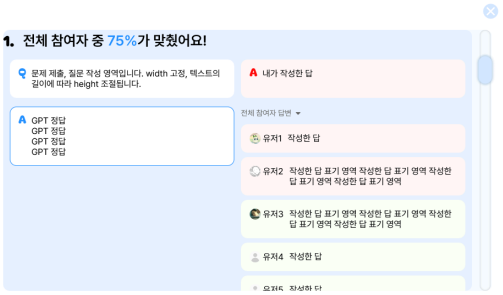
All rights are reserved. Reproduction in whole or in parts is prohibited without the written consent of the copyright owner.

	
[시작] 시작 버튼을 클릭합니다.	[로그인] 로그인 및 회원가입 합니다.
방 생성하기	
	
[메인로비] 오른쪽 상단의 방생성하기 버튼을 클릭합니다.	[방생성하기] 방 정보를 입력 후 생성하기 버튼을 클릭합니다. 문제 제작 옵션 - 리스트 선택: 자신이 생성한 퀴즈 리스트 중에서 선택합니다. - 직접 제작: PDF파일을 업로드합니다.
	

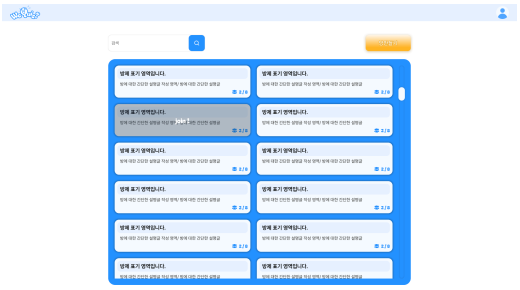
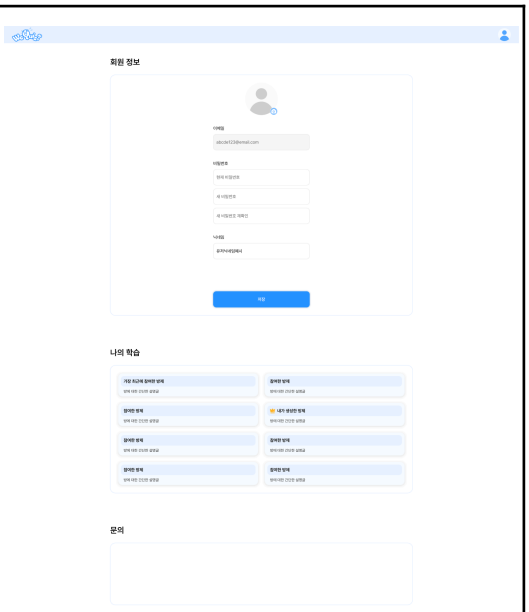
All rights are reserved. Reproduction in whole or in parts is prohibited without the written consent of the copyright owner.

[닉네임정하기] 닉네임 작성 후 입장 버튼을 클릭합니다.	
방 접속하기	
	
[메인로비] 리스트에서 방을 클릭합니다.	[닉네임정하기] 닉네임 작성 후 입장 버튼을 클릭합니다.
퀴즈 시작 전 기다리기	
	
[퀴즈대기방] 준비 버튼을 클릭하여 준비합니다. 모든 참여자들, AI봇까지 준비가 완료되면 퀴즈가 시작됩니다. 하단에 채팅창을 이용하여 말풍선을 띄울 수 있습니다.	
퀴즈 풀기	

All rights are reserved. Reproduction in whole or in parts is prohibited without the written consent of the copyright owner.

	
[퀴즈풀이] 답안을 작성하고 제출 버튼을 클릭합니다.과반수가 답안을 제출하면 3초의 카운트 다운이 시작됩니다. 이미 풀었다면 다음 문제가 넘어갈 때까지 기다립니다. 아직 못 풀었다면 3초 내로 답안을 제출하지 않으면 오답 처리가 됩니다. 오른쪽의 이모지 버튼을 클릭하여 이모지 말풍선을 띄울 수 있습니다.	[라운드종료] 하나의 라운드가 끝나면 중간 점수가 발표됩니다. 다음 라운드까지 7초 간 대기하며 본인의 점수, 참여자들의 점수를 확인합니다.
	
[최종성적표] 마지막 라운드가 종료되면 자신의 최종 성적표 페이지로 이동합니다. 퀴즈 본문과 나의 채점 결과를 확인합니다. 오른쪽 상세보기 버튼을 클릭하여 상세 답안을 확인합니다.	[상세보기] 정답과 나의 답안, 그리고 다른 참여자들의 답안을 비교합니다.
마이페이지	

All rights are reserved. Reproduction in whole or in parts is prohibited without the written consent of the copyright owner.

	
[메인로비] 오른쪽 상단의 프로필 이미지를 클릭합니다. 마이페이지 버튼을 클릭합니다.	[마이페이지] 프로필 이미지 및 비밀번호 재설정 을 할 수 있습니다. 나의 학습에서 내가 푼 퀴즈 리스트를 확인합니다. 리스트에서 하나를 클릭하면 해당 퀴즈의 최종 성적표를 열람할 수 있습니다.
[최종성적표] 풀었던 퀴즈의 최종 성적표를 열람합니다.	

All rights are reserved. Reproduction in whole or in parts is prohibited without the written consent of the copyright owner.

5.2 테스트 케이스

대분류	소분류	기능	테스트 방법	기대 결과	테스트 결과
유저 정보	회원가입	이메일을 직접 입력하여 회원 가입을 한다.	로그인 페이지에서 [회원가입] 버튼을 누르면, 1) 이메일과 비밀번호, 비밀번호 재확인을 입력한다. 2) 이미 사용 중인 이메일이면 alert을 띄우고 별도의 유저 정보 추가 액션이 일어나지 않는다.	유저 정보가 db에 추가된다.	성공
		카카오, 구글의 로그인 정보를 이용하여 회원 가입을 한다.	로그인 페이지에서 [카카오 로그인]이나 [구글 로그인] 버튼을 누르면, 1) 카카오나 구글 정보를 이용하여 계속하기 창이 뜬다. 2) 이미 같은 정보로 회원가입한 유저라면 바로 로그인이 된다.	유저 정보가 db에 추가된다.	성공
	로그인	이메일을 직접 입력하여 로그인 한다.	로그인 페이지에서 이메일과 비밀번호를 입력한다. 1) 회원가입하지 않은 정보의 경우 alert을 띄우고 별도의 로그인 액션이 일어나지 않는다.	로그인 된다.	성공
		카카오, 구글의 로그인 정보를 이용하여 로그인한다.	로그인 페이지에서 [카카오 로그인]이나 [구글 로그인] 버튼을 누른다.	로그인이 되고 이메일, 프로필 사진을 가져온다.	성공
	로그아웃	로그아웃 한다.	메인로비 오른쪽 상단의 프로필 아이콘을 클릭하면, 1) [로그아웃] 버튼을 클릭한다.	로그아웃 된다. 이후 재접속시 로그인을 요구한다.	성공
	유저 정보 변경	비밀번호를 변경한다.	메인로비 오른쪽 상단의 프로필 아이콘을 클릭하면, 1) [마이페이지] 버튼을 클릭한다. 2) 비밀번호 변경, 재확인 칸을 입력하고 변경버튼을 클릭한다.	비밀번호가 변경되고, 이후 재로그인 시 새로운 비밀번호을	성공

All rights are reserved. Reproduction in whole or in parts is prohibited without the written consent of the copyright owner.

				이용해 로그인해야한다.	
		프로필 사진을 변경한다.	메인로비 오른쪽 상단의 프로필 아이콘을 클릭하면, 1) [마이페이지] 버튼을 클릭한다. 2) 프로필 이미지 오른쪽 하단의 카메라 아이콘을 클릭한다. 3) jpg, jpeg, png 이미지를 선택하여 이미지를 변경한다.	변경된 이미지가 적용된다.	성공
	기타 정보	내가 푼 퀴즈 리스트 및 최종 성적표를 조회한다.	메인로비 오른쪽 상단의 프로필 아이콘을 클릭하면, 1) [마이페이지] 버튼을 클릭한다. 2) 내가 푼 퀴즈 리스트에서 퀴즈id 하나를 클릭한다.	해당 퀴즈에 대한 최종 성적표 페이지로 이동한다.	성공
퀴즈방	퀴즈방 생성	*문제 직접 제작하여 퀴즈방 생성*	메인 로비 오른쪽 상단의 [방만들기] 버튼을 클릭하면, 1) 방생성하기 페이지가 열린다. 2) 문제만들기에서 [직접제작] 옵션을 클릭한다. 3) 파일을 드래그앤드롭하고 [방생성하기] 버튼을 클릭한다.	닉네임 작성 모달이 뜨고, 닉네임 작성 후 [다음] 버튼을 클릭하면, 메인 로비에 방이 생성된다.	성공
		기존 문제 리스트에서 선택하여 퀴즈방 생성	메인 로비 오른쪽 상단의 [방만들기] 버튼을 클릭하면, 1) 방생성하기 페이지가 열린다. 2) 문제만들기에서 [리스트선택] 옵션을 클릭한다. 3) 내가 생성한 퀴즈 리스트가 조회되고, 이 중 선택 후 [방생성하기] 버튼을 클릭한다.	닉네임 작성 모달이 뜨고, 닉네임 작성 후 [다음] 버튼을 클릭하면, 메인 로비에 방이 생성된다.	성공
	퀴즈 대기방 입장	메인 로비에서 입장	메인 로비에서 방을 클릭하면, 1) 닉네임 작성 모달이 뜬다. 2) 닉네임 작성 후 [다음] 버튼을 클릭한다.	퀴즈 대기방으로 이동한다.	성공
		공유 코드로 입장	메인 로비의 [코드로 입장하기] 입력칸에 공유받은 코드를 입력하면, 1) 닉네임 작성 모달이 뜬다. 2) 닉네임 작성 후 [다음] 버튼을 클릭한다.	퀴즈 대기방으로 이동한다.	성공
	퀴즈 대기방	퀴즈방 정보 불러오기	퀴즈 대기방에 접속한다.	퀴즈방 제목, 설명 접속한 인원 및	성공

All rights are reserved. Reproduction in whole or in parts is prohibited without the written consent of the copyright owner.

				준비 상태를 불러온다.	
		채팅	채팅창에 메세지를 작성 및 전송한다.	프로필 이미지 위에 말풍선과 함께 채팅 내용이 나타난다. 다른 참여자들이 실시간으로 채팅 말풍선을 볼 수 있다.	성공
		문서요약	퀴즈 대기방에 접속한다.	문서 요약이 스트리밍 형태로 나타난다.	성공
		준비하기	[준비하기] 버튼을 클릭한다.	나의 프로필이 준비상태로 변경된다.	성공
		시작하기	모두가 준비하고 AI봇도 준비를 완료한다.	3초 카운트 후 퀴즈 시작페이지로 넘어간다.	성공
	퀴즈 진행	문제 출제	퀴즈 풀이 페이지로 이동한다.	모두가 동시에 같은 문제를 제공받는다.	성공
		답안 제출	답안 표기 후 [제출하기] 버튼을 클릭한다.	오른쪽 나의 프로필에 제출 상태가 실시간으로 표시되며, 다른 참여자들도 실시간으로 이를 확인할 수 있다.	성공
		다음 문제	참여자의 과반수가 답안을 제출했다고 표시가 되면, 3초의 카운트가 시작되고 다음 문제로 넘어간다.	모두가 동시에 다음 문제를 제공받는다.	성공
		중간 점수	다섯번 째 문제가 종료된다.	모두가 동시에 중간 라운드결과 페이지를 제공받는다. 이 페이지에서 7초간 대기하며 중간 점수를 확인한다.	성공

All rights are reserved. Reproduction in whole or in parts is prohibited without the written consent of the copyright owner.

	퀴즈 종료	최종 성적표	모든 퀴즈풀이가 완료되고, 마지막 라운드의 중간 성적표가 7초간 조회되며, 7초 후 최종 성적표 페이지로 이동한다.	최종 성적표에서 퀴즈 본문과 AI의 답안, 나의 답안 그리고 다른 참여자들의 답안을 확인한다.	성공

All rights are reserved. Reproduction in whole or in parts is prohibited without the written consent of the copyright owner.