

AR 기반 원격 협업시스템 구현 중간 보고서



지도교수 : 이명호

분과명 : 소프트웨어/인공지능

팀명 : 꿈깡깡

202055507 강유승

202055525 김준성

202055569 윤민혁

목차

1. 서비스 개발 배경
2. 요구조건 및 제약사항 분석에 대한 수정사항
3. 설계 상세화 및 변경 내역
4. 서비스 핵심 기능
5. 자문 의견 및 해결 방안 모색
6. 역할 분담
7. 진행 현황

1. 서비스 개발 배경

- 새 컴퓨터 부품을 구매하여 이를 조립하는 것은 상당한 난이도가 있어 전문 지식이 부족한 사용자는 직접 전문가를 찾아가 조립하거나 인터넷에 올라와 있는 정보만을 사용하여 조립하여야 한다는 번거로움이 있습니다.
- 저희는 AR 기반 원격 시스템을 통해 3D 모델, 텍스트 등을 활용한 주석을 추가하여 사용자가 컴퓨터 조립을 하는 데 도움을 주는 시스템을 구현해 보고자 하였습니다.

2. 요구조건 및 제약사항 분석에 대한 수정사항

a. 기존 요구조건

본 프로젝트는 AR 기술을 활용하여 원격 협업 시스템을 구현하는 것을 목표로 합니다. 주요 요구사항은 다음과 같습니다:

- **스마트폰 AR 기반 원격 협업 시스템 개발:** 사용자가 스마트폰을 통해 AR 세션을 시작하고 지원을 요청하며, 전문가가 실시간으로 주석을 추가하여 문제 해결을 지원합니다.
- **실시간 음성 및 영상 통화 기능 통합:** 원활한 커뮤니케이션을 위한 실시간 음성 및 영상 통화 기능을 제공합니다.
- **원격 전문가의 시야 공유 및 정보 제공:** 원격 전문가가 스마트폰 AR 사용자의 시야를 공유하고, 필요한 정보를 증강 현실로 제공하는 기능을 구현합니다.
- **효율적인 세션 매칭 및 상태 동기화:** 사용자-전문가 간의 효율적인 매칭 시스템과 실시간 상태 동기화를 지원합니다.

b. 요구 조건 수정사항

초기 착수보고서에서는 Unity AR Foundation을 기반으로 한 개발을 계획했으나, Unity의 WebRTC 패키지 버전 문제 및 웹 기반의 접근성을 고려하여 Next.js (React), Node.js (Express), Firebase, Socket.io를 활용한 웹 애플리케이션 형태로 개발 방향을 전환하였습니다. 이에 따라 AR 기능은 웹 기반 AR 기술(Three.js, WebXR 등) 또는 외부 AR SDK 연동을 통해 구현될 예정이며, 실시간 통신 및 데이터 관리는 웹 환경에 최적화된 방식으로 진행됩니다.

유스케이스 다이어그램

3. 설계 상세화 및 변경 내역

기술 스택

- Next + React
- Node.js(Express) + WebRTC
- WebXR
- Firebase
- AWS EC2

초기 계획은 Unity 기반의 AR 애플리케이션 개발이었으나, 현재 프로젝트는 웹 접근성과 개발 효율성을 고려하여 다음과 같은 아키텍처로 변경되었습니다.

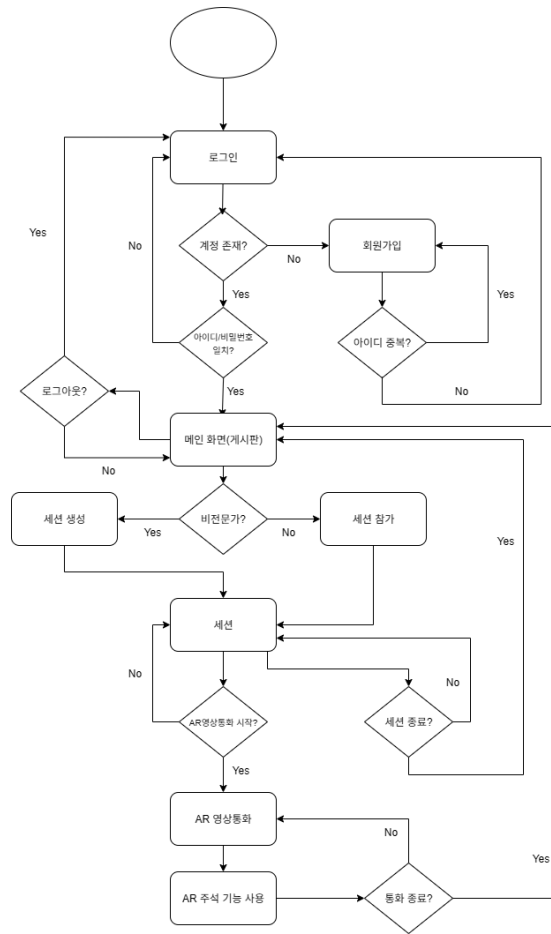
- **프론트엔드:** Next.js (React)를 사용하여 사용자 인터페이스를 구축합니다. app/ 디렉토리 내에 페이지 라우팅 및 컴포넌트들이 구성되어 있습니다.
- **백엔드 API:** Node.js (Express)를 기반으로 RESTful API를 제공합니다. pages/api/ 디렉토리 내에 API 라우트가 정의되어 있으며, server/socketServer.js를 통해 실시간 통신을 위한 [Socket.io](https://socket.io) 서버가 운영됩니다.
- **데이터베이스 및 인증:** Firebase를 활용하여 사용자 인증 및 데이터베이스를 관리합니다. (lib/firebase.ts 참조)
- **실시간 통신:** Socket.io를 사용하여 클라이언트와 서버 간의 실시간 양방향 통신을 구현합니다. (lib/socket.js, server/socketServer.js 참조)
- **AR 기능:** 웹 기반 AR 기술(예: WebXR) 또는 AR 기능을 제공하는 라이브러리를 프론트엔드에 통합하여 구현할 예정입니다.

4. 서비스 핵심 기능

- **회원가입/로그인:** 모든 유저는 회원가입 후 로그인을 통해 시스템을 사용할 수 있습니다.
- **세션 생성 :** 사용자는 게시판에 컴퓨터 조립에 문제를 겪고 있는 부분을 담고 있는 게시물을 등록하여 서버 내에 새로운 세션을 생성합니다.
- **세션 참가 :** 전문가는 게시판에서 사용자가 게시한 게시물을 선택하여 세션에 입장하고 사용자가 원할 시 AR 화상 통화를 시작할 수 있습니다.
- **AR 화상 통화:** 사용자와 전문가는 화상, 음성 통화를 통해 실시간으로 정보를 공유하여 문제 해결에 도움을 줄 수 있습니다.
- **AR 주석:** 사용자와 전문가는 3D 오브젝트(ex. 간소화 된 CPU, RAM 모델)나 화살표, 마커, 텍스트와 같은 시각적 정보를 활용하여 문제를 해결할 수 있습니다.

5. 자문 의견 및 해결 방안 모색

- **연구 배경 및 목표 구체화:** AR 기반 실시간 협업 시스템이 필요한 특정 시장(예: 제품 기반 고객 콜 센터)을 명확히 하고, 이에 따른 목표를 세분화할 필요가 있습니다.
 - 컴퓨터 조립을 힘들어하는 사람(게스트)들과 AR을 활용하여 컴퓨터 조립의 전반적인 과정을 도와주는 전문가의 협업 시스템으로 서비스 목표를 구체화하여 해결하였습니다
- **고객 니즈 및 사용 시나리오 상세화:** 현재 고객들이 겪는 문제점(음성 정보 전달의 한계)을 명확히 제시하고, AR 기반 시스템이 이를 어떻게 해결하는지 구체적인 사용 시나리오를 통해 설명해야 합니다. 기능적인 접근보다는 사용자 관점의 플로우를 고민해야 합니다.
- **서비스 전체 개념 시각화:** 서비스의 전반적인 개념, 플로우, 개념 아키텍처 등을 도식화하여 이해도를 높여야 합니다.
 - 아래의 플로우 차트를 통해 사용자 관점의 구체적인 사용 시나리오를 명시하였습니다.



플로우 차트

- 3D 모델 공유 시나리오 심층 검토:** 3D 모델 공유 시 모바일 CPU 자원 한계, 정합의 어려움, 사용자 경험 저해 등의 문제가 발생할 수 있으므로, 이에 대한 현실적인 대안(예: 3D 모델 자체 렌더링 대신 2D 이미지 위에 AR 주석 오버레이)을 모색해야 합니다.
 - 2D 이미지를 통해 컴퓨터 조립의 이해를 돕는 것은 한계가 있다고 판단하여 3D 모델 사용이 불가피할 것 같습니다.
 - 3D 오브젝트에 의한 CPU 부하는 오브젝트의 폴리곤 수와 비례한다고 알고 있습니다. 따라서 정밀한 컴퓨터 부품 모델링을 사용하기보다는 단순 컴퓨터 부품의 규격만 맞춘 직육면체로 3D 모델 공유를 지원할 예정입니다.
- 소켓 통신 재연결 및 예외 처리:** 네트워크 불안정으로 인한 소켓 통신 단절 및 재연결 상황에 대한 예외 처리를 설계에 반영해야 합니다.
 - 재연결 후에도 세션 상태를 유지할 수 있도록 예외 처리 로직을 강화하겠습니다.

6. 역할 분담 및 진척도

강유승

- DB 구축 및 관리
 - Firebase Realtime DB를 통해 로그인/회원가입 기능 구현 완료
 - 비밀번호는 salting 암호화를 통해 보안됨
 - 후에 게시물과 관련된 클래스가 필요할 시 추가할 예정
- AR 기능 구현
 - AR 환경을 카메라에 연동시키고 전문가와 사용자가 서로 화면 공유되도록 할 예정
 - AR 주식에 사용할 모델, 텍스트, 마커 등을 화면에 표시하고 자유롭게 위치시킬 수 있는 기능 추가 예정

김준성

AR 기능 구현 계획 (예정)

1. 핵심 AR 기능 (고객 측):

- * 전문가로부터 AR 주식 데이터를 수신하여 파싱.
- * 스마트폰 카메라 화면에 주석을 3D 객체로 렌더링하고, 이미지 트래킹을 통해 실제 제품 위치에 고정.

2. 주식 시스템 (전문가 측):

- * 고객 영상 스트림의 특정 프레임을 캡처.
- * 캡처 화면에 다양한 형태의 주식(화살표, 텍스트 등)을 작성하는 UI 구현.
- * 생성된 주식 데이터를 WebSocket을 통해 고객에게 전송할 수 있도록 준비.

3. 데이터 전송 및 동기화:

- * 주식 '전송', '삭제' 등을 위한 WebSocket 이벤트를 정의.
- * 전문가가 원할 때, 작성된 주식 데이터를 고객의 화면에 전송하여 동기화하는 기능 구현.

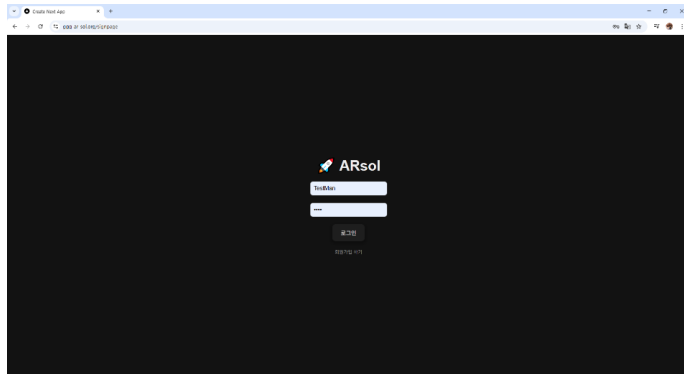
윤민혁

- 게시판 서버 구축 및 관리
 - Socket.io를 활용한 상담 요청 게시판 구축 완료
 - 방 입장 및 퇴장 시 이벤트 처리 로직 보완 필요
- 화상 통화 환경 구축 및 관리
 - WebRTC를 활용한 원활한 화상 통화 환경 구축 완료
 - 화상통화 방에 입장 시, 가끔 시그널을 받아오지 못하는 현상 수정 필요
 - 화상통화 관련 더욱 편리한 UX 기능 제공 예정
- 서비스 배포
 - <https://ggg-ar-sol.org/> 도메인 설정 완료
 - AWS 인스턴스에 중간 버전 프로젝트 업로드 및 배포 완료
 - 프로젝트가 진행될 때마다 주기적으로 빌드 예정

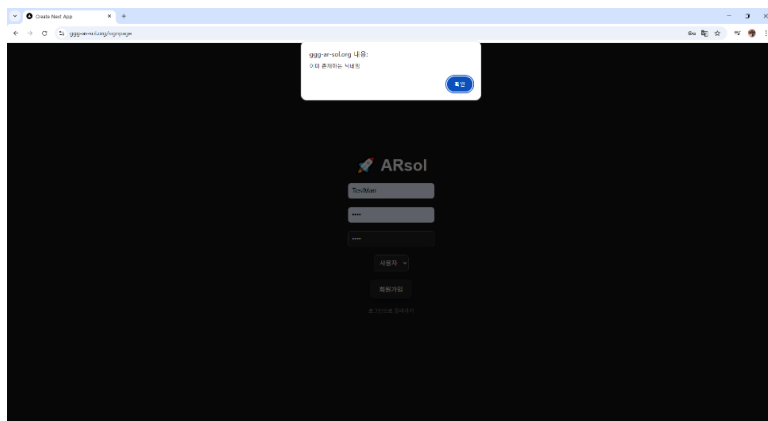
7. 진행 현황

a) 프로젝트 예시 사진

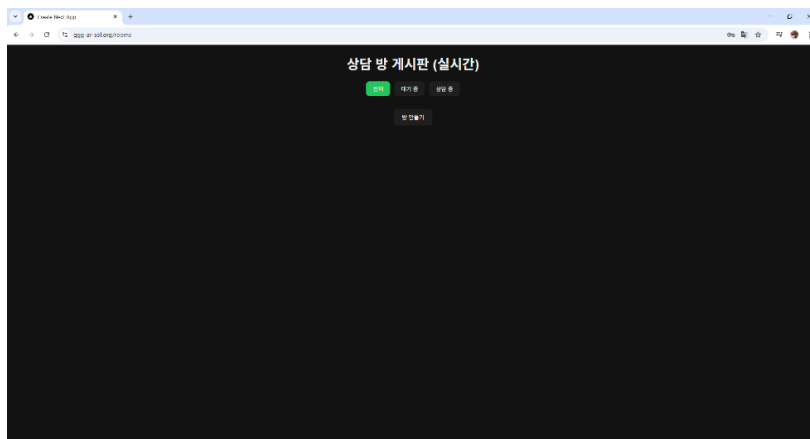
1: 로그인 화면



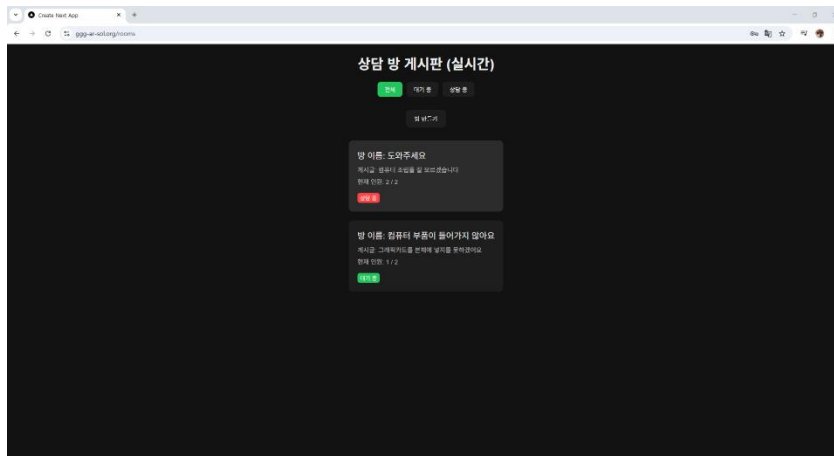
2: 회원가입 및 유저 이름 중복 확인



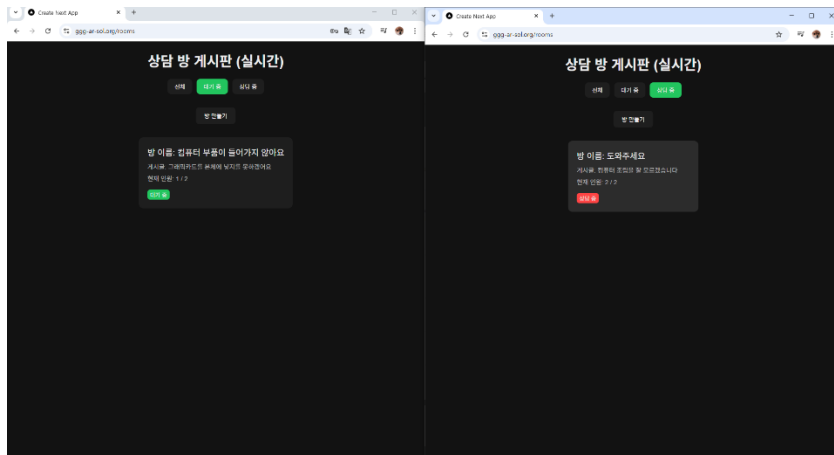
3: 로그인 성공 후 메인 (상담 게시판)



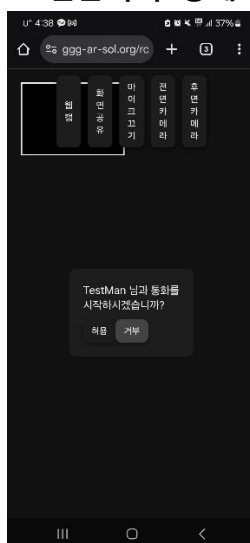
4: 상담 요청 게시물이 등록된 상황에서의 게시판



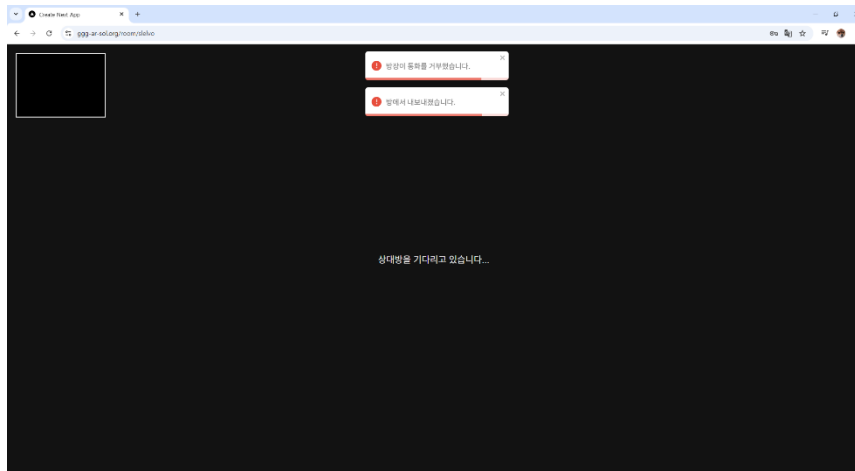
5: 대기 중, 상담 중 태그 검색 기능



6: 전문가가 방에 들어왔을 때, 사용자에게 표시되는 통화 수신/수신거부

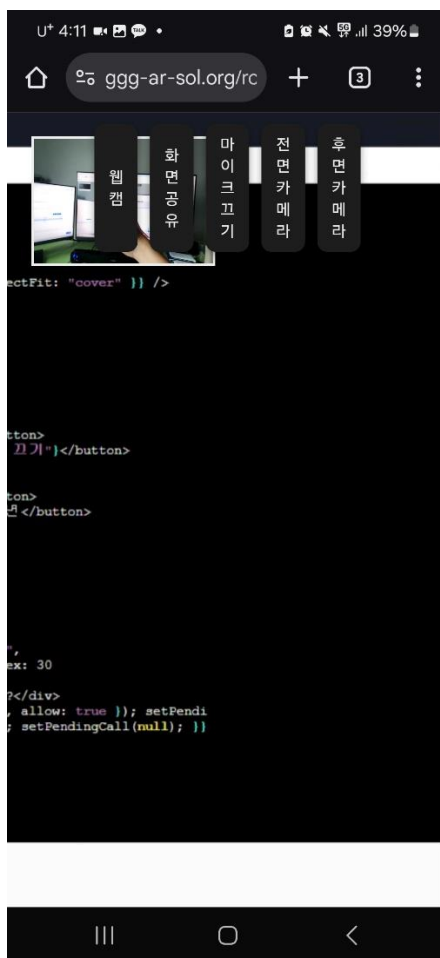


7: 사용자가 전문가와 통화를 거부했을 때



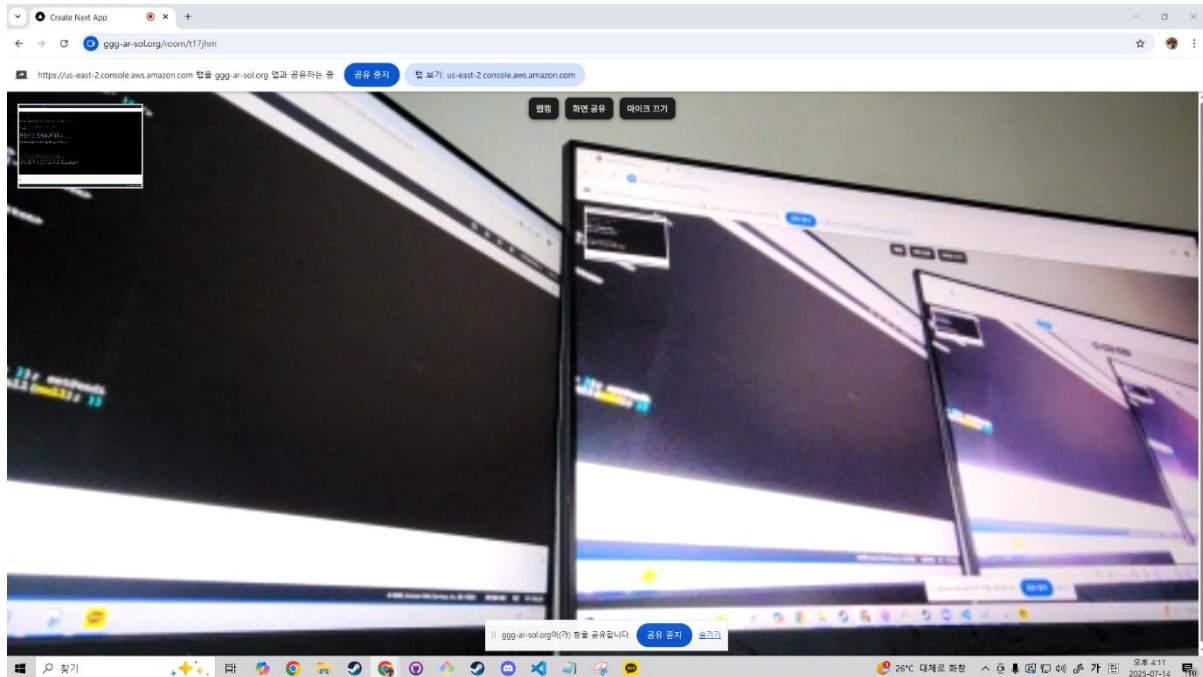
8: 통화 연결 후 사용자(모바일)화면

(전문가의 화면이 크게, 사용자의 화면은 위에 플로팅)



9: 통화 연결 후 전문가(PC) 화면

(사용자의 화면이 크게, 전문가의 화면 플로팅)



b) 일정 계획

강유승	[7/9~7/13] FirebaseDB 회원가입/로그인/게시판 구현 [7/14~7/31] AR 기능 개발 [8/1 ~] DB로직 예외처리 및 부가 기능 구현
김준성	[7/15 ~ 7/31] AR 기능 개발 [8/1 ~] UI/UX 디자인 및 구현
윤민혁	[~7/9] 게시판 서버 구축 [~7/11] 영상통화 환경(비디오,오디오) 구축 [7/14 ~ 7/31] 영상통화 관련 버그 수정 및 부가 기능 구현, 예외처리 강화 [8/1 ~] UI 로직 연결