

캡스톤 디자인 I

종합설계 프로젝트

프로젝트 명	커리어 한 잔
팀 명	17조
문서 제목	중간보고서

Version	2.1
Date	2024-03-31

팀원	이 하람 (조장)
	공 채연
	권 해담
	김 승언
	김 혜은
	홍 지민
지도교수	허 대영 교수

 국민대학교 소프트웨어학부 캡스톤 디자인 I	중간보고서		
	프로젝트 명	커리어 한잔	
	팀 명	17조	
	Confidential Restricted	Version 2.1	2024-03-31

문서 정보 / 수정 내역

Filename	중간보고서-커리어한잔.doc
원안작성자	이하람, 공채연, 권해담, 김혜은
수정작업자	김승언, 홍지민

CONFIDENTIALITY/SECURITY WARNING

이 문서에 포함되어 있는 정보는 국민대학교 소프트웨어융합대학 소프트웨어학부 및 소프트웨어학부 개설 교과목 캡스톤 디자인 수강 학생 중 프로젝트 “커리어 한잔”을 수행하는 팀 “17조”의 팀원들의 자산입니다. 국민대학교 소프트웨어학부 및 팀 “17조”의 팀원들의 서면 허락없이 사용되거나, 재가공 될 수 없습니다.

수정날짜	대표수정자	Revision	추가/수정 항목	내 용
2024-03-28	이하람	1.0	최초 작성	프로젝트 수행 내용 작성
2024-03-28	공채연	1.1	최초 작성	프로젝트 수행 내용 작성
2024-03-29	권해담	1.2	최초 작성	구체화된 프로젝트 내용 작성
2024-03-29	김혜은	1.3	최초 작성	구체화된 프로젝트 내용 작성
2024-03-30	김승언	2.0	내용 수정	프로젝트 수정 내용 추가
2024-03-31	홍지민	2.1	내용 수정	프로젝트 수정 내용 추가

 국민대학교 소프트웨어학부 캡스톤 디자인 I	중간보고서		
	프로젝트 명	커리어 한잔	
	팀 명	17조	
	Confidential Restricted	Version 2.1	2024-03-31

목 차

1. 프로젝트 목표	4
1.1. 개발 동기	4
1.2. 문제해결 방법 및 기대효과	5
2. 수행 내용 및 중간결과	6
2.1. 계획서 상의 연구내용	6
요구사항 분석	6
수행 일정 계획	6
2.2. 수행내용	6
현재 수행 내용	7
비교분석 : 수행 과정 잘한점 고칠점 뽑아보기	7
3. 수정된 연구내용 및 추진 방향	8
3.1. 수정사항	8
4. 향후 추진계획	10
4.1. 향후 계획의 세부 내용	10
5. 고충 및 건의사항	11

 국민대학교 소프트웨어학부 캡스톤 디자인 I	중간보고서		
	프로젝트 명	커리어 한잔	
	팀 명	17조	
	Confidential Restricted	Version 2.1	2024-03-31

1. 프로젝트 목표

1.1. 개발 동기

요즘 직장인들 간에는, 특히나 IT 분야에서는 이직을 위한 정보를 얻거나 서로가 가진 업무 경험과 인사이트를 공유하고 채용을 제안하는 등, 매우 다양한 목적으로 ‘커피챗’이라는 문화가 점차 발달하고 있다. 커피챗이란, 비즈니스 관계에서 커피 한잔하듯 가볍게 만나 인맥을 쌓고 정보를 공유하는 것을 말한다. 그러나 아직 우리나라에서는 이 문화가 서구권에 비해 활발하지 않고, 실제 커피챗이 이뤄지기까지의 절차가 다소 번거로우며 접근성이 떨어진다. 이러한 점에서 가볍고 캐주얼한 커피챗을 위한 새로운 접근 방식의 필요성을 느끼게 되었다.

이에 모든 IT 분야의 현직자들 간의 부담없는 커리어 대화를 위한 커피챗 매칭 서비스를 만들어보자는 의견으로 뜻을 모았다.

물론 커피챗 매칭에 활용할 수 있는 혹은 커피챗 매칭만을 위한 플랫폼들이 이미 존재하지만, 모두 온라인을 통한 대화를 지향한다는 한계가 있다. 그러나 코로나 팬데믹을 지나며 모두가 경험했듯이 온라인의 편리함만으로는 대체할 수 없는 오프라인 만남의 이점들이 있다. 이는 커피챗이라는 개방적이고 자유로움을 지향하는 문화에도 예외없이 적용될 것이다. 또한 우리에게 진짜 필요한 커피챗 문화는 당장 나에게 필요한 정보만을 얻기 위한 일회성 연락이 아닌, 서로 도움을 주고받으며 함께 발전해나가는 지속적 관계 형성이며 이를 위해서는 대면 만남이 필수적이라고 판단했다.

이러한 동기로 ‘주변 현직자들과 대면 만남이 가능한 실시간 커피챗 매칭 서비스, 커리어 한잔’ 프로젝트를 시작하게 되었다.

 국민대학교 소프트웨어학부 캡스톤 디자인 I	중간보고서		
	프로젝트 명	커리어 한잔	
	팀 명	17조	
	Confidential Restricted	Version 2.1	2024-03-31

1.2. 문제해결 방법 및 기대효과

1) 실시간 커피챗 매칭

다른 사용자에게 직접 연락을 취해 약속 날짜와 장소를 잡는 대신, 현재 가까운 곳에 위치한 사용자에게 커피챗 요청을 보내 바로 커피챗이 이루어질 수 있도록 한다. 이러한 실시간 매칭 방식은 사용자들의 편의성을 증가시키고 현재 우리나라의 딱딱하고 재미없는 커피챗의 이미지를 보다 가볍고 캐주얼하게 만들어 커피챗 문화 활성화에 기여할 수 있으리라 기대된다.

- 경쟁 서비스와의 비교

vs. 링크드인(LinkedIn): 직접적인 커피챗 매칭보다는 다른 사용자의 정보를 습득하고 연락을 취하는 용도로 사용된다. 커피챗 요청 시 적당히 예의와 양식을 갖춰 메시지를 보내야 하고, 상대방부터 답변이 오지 않거나, 답변이 오더라도 서로 가능한 일정을 맞춰야 하므로 커피챗이 성사되기까지 여러 절차와 번거로움을 요한다.

2) 대면 만남

대면 만남을 통한 커피챗은 온라인 커피챗에 비해 친밀감과 유대감을 형성하기 좋고, 깊은 대화를 이어나가기에 더욱 적합하다. 따라서 휴지처럼 뽑아 쓰고 버리는 단순 일회성 관계가 아닌, 지속적인 관계 및 비즈니스 인맥으로의 발전을 기대해볼 수 있다.

커피챗 목적을 기준으로 구분해서 보자면, 가볍게 회사에서 점심식사 후 남는 시간을 의미있게 쓰고자 하는 개발자는 대면 커피챗을 통해 산책 메이트이자 소중한 개발자 인맥을 얻을 수 있다. 또한 다른 직무로의 이직을 준비하는 이들에게 대면 커피챗은 즉각적인 소통으로 실무의 생생한 이야기를 들을 수 있어 더욱 실질적인 도움을 줄 수 있다. 주니어 개발자로부터 참신하고 기발한 젊은 세대의 시선을 배우고자 하는 시니어 개발자의 경우라면, 편안한 분위기의 일대일 만남을 통해 상대를 더 면밀히 살펴 회사에 필요한 인재를 찾을 수도 있을 것이다.

 국민대학교 소프트웨어학부 캡스톤 디자인 I	중간보고서		
	프로젝트 명	커리어 한잔	
	팀 명	17조	
	Confidential Restricted	Version 2.1	2024-03-31

- 경쟁 서비스와의 비교

vs. 커피챗(Coffeechat): 음성 커피챗 플랫폼. 커피챗 매칭을 해주는 플랫폼이지만, 음성 대화로 진행하며 멘티가 멘토에게 비용을 지불하는 멘토링 성격의 커피챗이다. 서로 도움을 주고받는 게 아니라 한 쪽이 일방적으로 도움을 준다는 점에서 커피챗의 본래 취지와는 동떨어진다. 또한 지속적인 관계 형성이 아닌 취업이나 이직 준비를 위한 일회성 전문가 컨설팅에 가깝다.

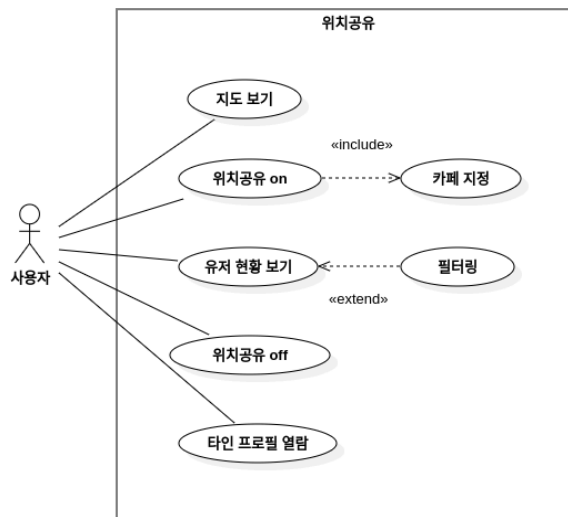
 국민대학교 소프트웨어학부 캡스톤 디자인 I	중간보고서		
	프로젝트 명	커리어 한잔	
	팀 명	17조	
	Confidential Restricted	Version 2.1	2024-03-31

2. 수행 내용 및 중간결과

2.1. 계획서 상의 연구내용

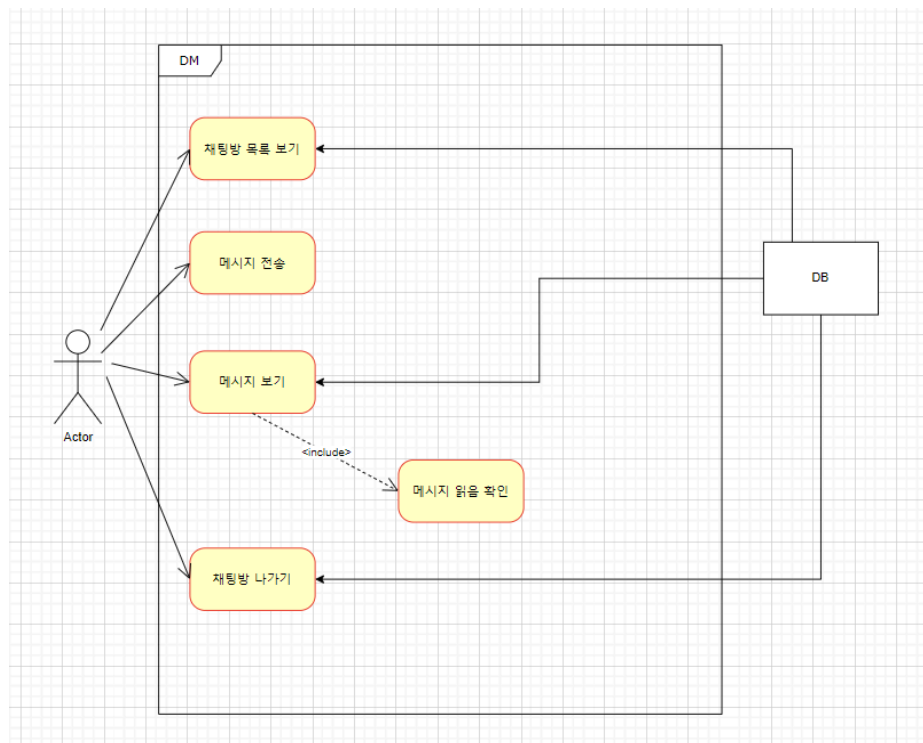
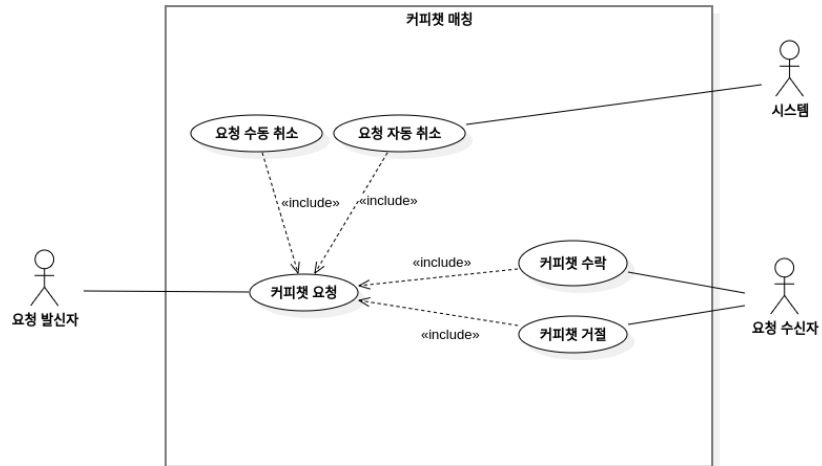
요구사항 분석

- Use Case Diagram

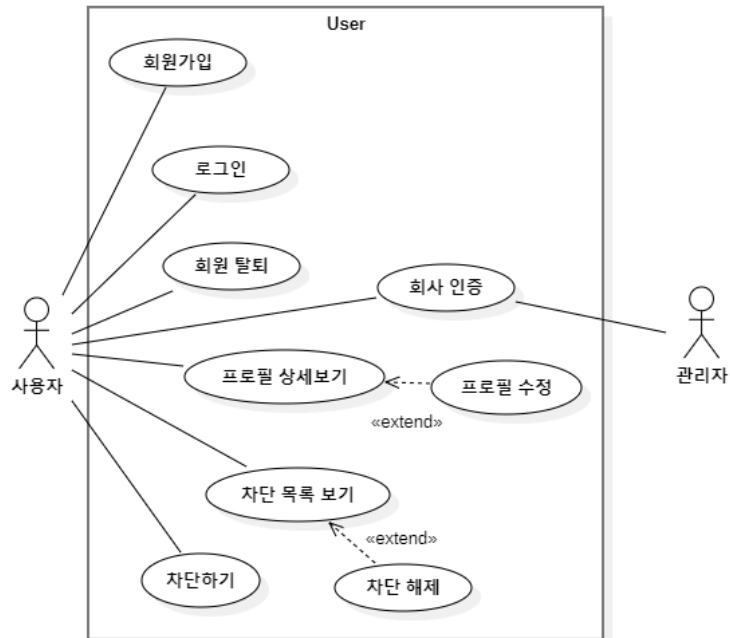




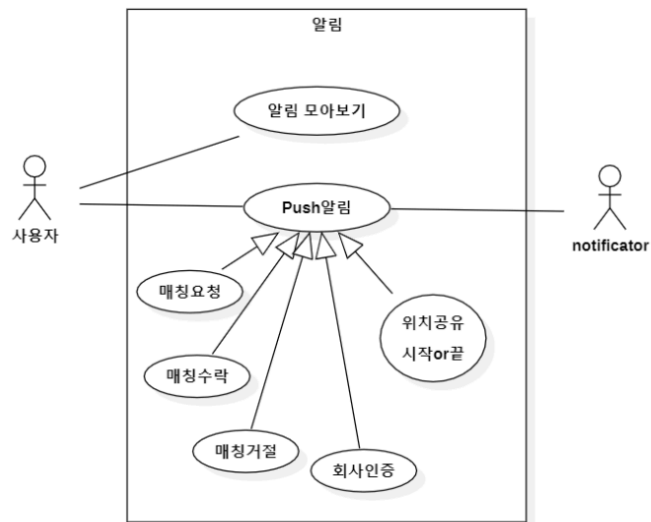
중간보고서		
프로젝트 명	커리어 한잔	
팀 명	17조	
Confidential Restricted	Version 2.1	2024-03-31



 국민대학교 소프트웨어학부 캡스톤 디자인 I	중간보고서		
	프로젝트 명	커리어 한잔	
	팀 명	17조	
	Confidential Restricted	Version 2.1	2024-03-31



[Notification]



 국민대학교 소프트웨어학부 캡스톤 디자인 I	중간보고서		
	프로젝트 명	커리어 한잔	
	팀 명	17조	
	Confidential Restricted	Version 2.1	2024-03-31

기술적 해결 방안 연구

Push Notification

푸시 알림 기능은 FCM(Firebase Cloud Messaging)을 사용하여 구현한다.

Polling 기법은 높은 실시간성을 유지하기 위해서는 짧은 주기로 요청과 응답 작업을 반복하기 때문에 불필요한 트래픽이 많이 발생하고, 클라이언트와의 커넥션을 유지하고 있어야 하기 때문에 클라이언트가 포그라운드에서 실행 중이어야 한다. 이 점은 배터리 소모량이 심하면 안되는 어플리케이션 특성에 적합하지 않은 방법이다.

그에 반해, **FCM**은 클라이언트가 포그라운드로 실행중이지 않더라도 **FCM** 서버에서 알림을 줄 수 있다. 이 점은 위치 공유 기능을 켜놓은 채 어플리케이션을 백그라운드에 실행중인 상태로도 있을 수 있어야 하는 점에 부합하다.

또한, **FCM**은 무료로 메시지를 안정적으로 전송할 수 있는 교차 플랫폼 메시징 솔루션이다. 커리어 한잔 프론트엔드는 **Flutter** 언어를 사용해 확장성을 고려한 개발을 진행할 예정이므로 푸시 알림 기능 또한 교차 플랫폼 메시징을 염두에 두고 구현하기로 결정했다.

위치 공유 기능

실시간성을 보장하면서 위치를 공유하기 위해서 서버는 **stomp** 프로토콜을 사용해 클라이언트와 통신한다.

- **http** 프로토콜은 단방향 통신이라 지속적인 채팅 연결에는 적합하지 않다고 판단했다.
- **WebSocket** : HTTP를 기반으로 동작하고, HTTP header에 **ws**를 넣음으로서 **websocket** 프로토콜로 전환하는 요청으로 시작된다. 웹소켓은 메시지를 주고 받는 형식이 정해져있지 않아서, 메시지 형식을 따로 지정해줘야하고, 프론트, 백 측에서 각자 파싱을 해줘야하는

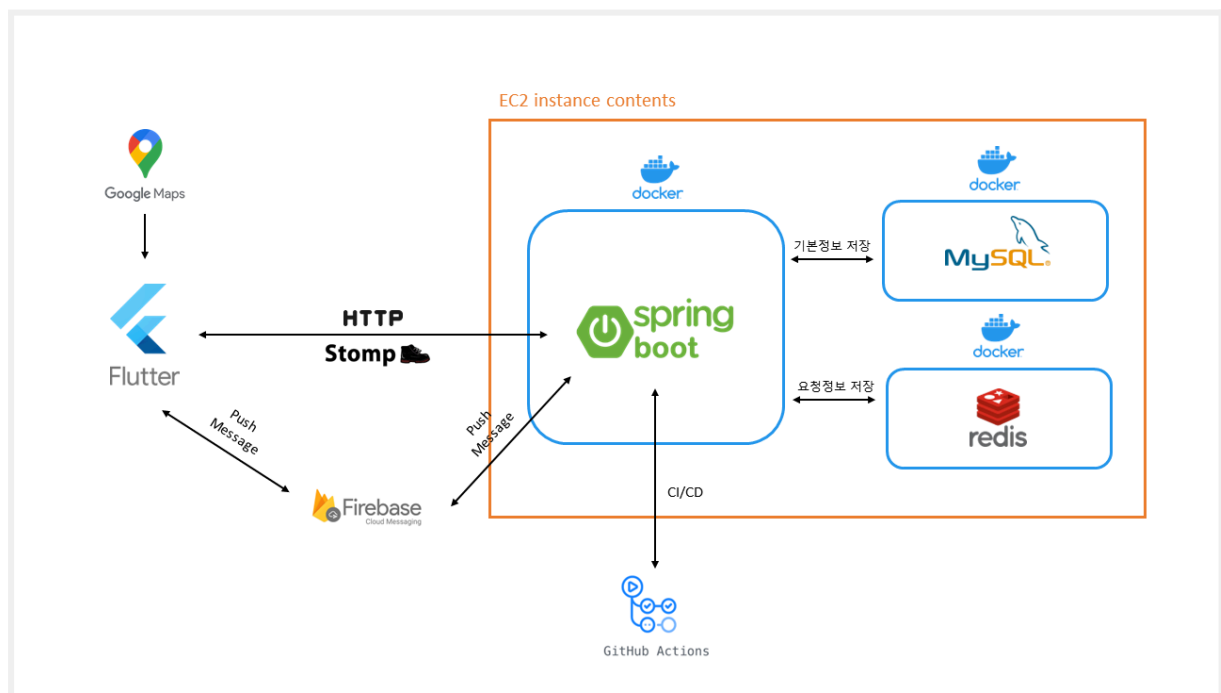
 국민대학교 소프트웨어학부 캡스톤 디자인 I	중간보고서		
	프로젝트 명	커리어 한잔	
	팀 명	17조	
	Confidential Restricted	Version 2.1	2024-03-31

불편함이 있다.

- **stomp** : Simple Text Oriented Messaging Protocol로, 메시지 전송을 효율적으로 하기 위한 프로토콜이다. **websocket** 기반으로 동작하고, **pub/sub** 구조로 되어있다. 메시지 기반의 통신을 지원하기 때문에 메시징 시스템과의 통합이 가능하다.

서버는 **online** 상태인 사용자의 목록을 클라이언트에게 전송한다. 이때 클라이언트가 **polling** 하지 않더라도 변경사항을 전송해줄 수 있어야 한다. 따라서 실시간성을 보장하면서 위치를 공유하기 위해서 서버는 **stomp** 프로토콜을 사용해 클라이언트와 통신한다.

Service Architecture와 ERD



Client-Server 아키텍처

UI와 비즈니스로직의 책임을 분리하고 업무 분담으로 효율성을 증가시킬 수 있는

 국민대학교 소프트웨어학부 캡스톤 디자인 I	중간보고서		
	프로젝트 명	커리어 한잔	
	팀 명	17조	
	Confidential Restricted	Version 2.1	2024-03-31

클라이언트-서버 아키텍처를 채택했고, 플러터 **application** 클라이언트와 스프링 서버를 개발한다.

플러터

ios, 안드로이드 모바일 앱을 제한된 시간 내에 개발하기 위해 크로스 플랫폼 프레임워크인 **flutter**를 선택했다. 크로스 플랫폼 프레임워크들 중에서도 **flutter**에는 내장된 기능이 많다는 장점이 있다. 내장된 기능들을 최대한 활용하여 외부 라이브러리 같은 서드 파티 의존성을 줄이고, 보다 안정적으로 서비스를 개발하기 위해 선택했다. 또한 **flutter**의 튜토리얼 자료나 문서가 체계적으로 작성되어 있어 학습하는 데 용이했고 필요한 내용을 빠르게 찾을 수 있었다.

구글 맵 API

앱에 지도를 띄워 사용자의 위치를 표시하고, 반경 내 카페들의 정보를 가져오는 데에 플러터의 구글 맵 플러그인을 사용했다.

Spring프레임워크

오픈소스 프레임워크이면서 안정적인 개발과 개선이 보장된다. 스프링의 가장 큰 특징인 **POJO(Plain Old Java Object)** 프로그래밍을 지향한다는 점을 주목했고, 이를 통해 객체지향 설계가 용의하고 코드가 단순하기 때문에 테스트와 디버깅이 쉬워지는 등의 장점, 서비스를 개발할 때 기술보다 비즈니스 로직을 구현하는데에 집중할 수 있다는 장점 때문에 해당 프레임워크를 선택했다.

인메모리데이터베이스

매칭 요청, 온라인인 사용자의 상태 정보와 같이 **DB**에 영속적으로 남길 필요가 없는 휘발성 데이터를 서버 메모리에 캐싱해두는 목적으로 **Redis** 서버를 사용한다.

매칭 요청은 제한시간(**10분**)이 주어져있는데, **Redis**의 **expire** 옵션을 사용해 제한시간(**10분**)이 지났을 때 별도의 서버 로직을 추가하지 않아도 자동으로 **key**를 메모리 해제할 수 있다. 이를

 국민대학교 소프트웨어학부 캡스톤 디자인 I	중간보고서		
	프로젝트 명	커리어 한잔	
	팀 명	17조	
	Confidential Restricted	Version 2.1	2024-03-31

통해 메모리 효율성을 높이고자 한다.

MySQL

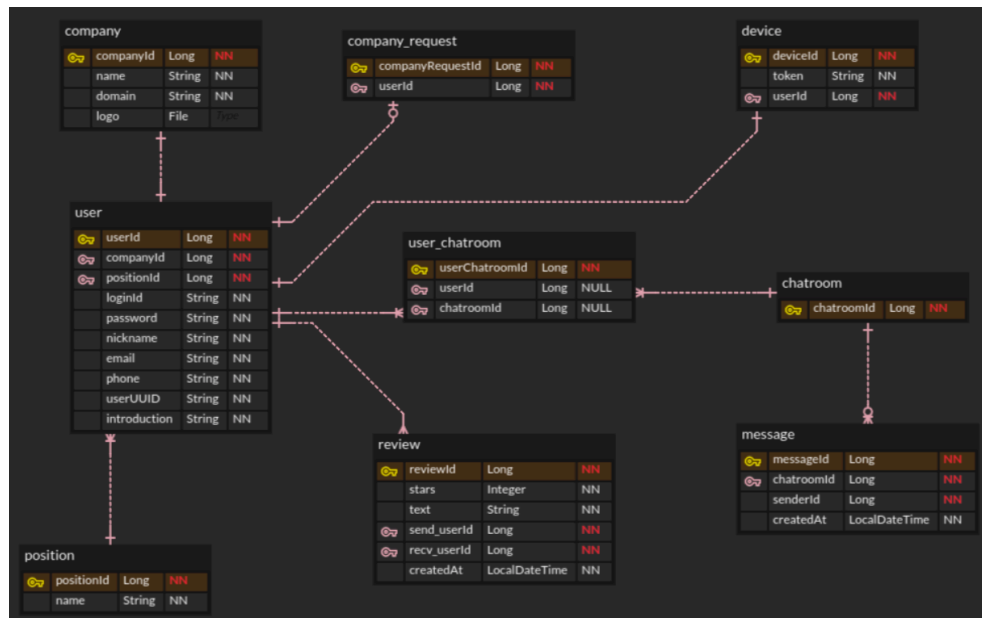
MySQL은 오픈 소스 데이터베이스 관리 시스템으로, 무료로 사용할 수 있기 때문에 초기 비용이나 라이선스 비용 없이 소프트웨어를 사용할 수 있다. 또한 오픈 소스이므로 커뮤니티 지원이 활발하며, 개발 및 유지보수가 용이하므로 프로젝트에 적합하다고 판단했다.

Stomp

HTTP, WebSocket, Stomp 등 다양한 통신 프로토콜이 있었고, 그 중 '실시간 채팅' 기능에는 Stomp 프로토콜이 적합하다고 판단하였다. HTTP는 단방향 통신이라 지속적인 채팅 연결에는 적합하지 않고, WebSocket은 메시지를 주고 받는 형식이 정해져있지 않아서 메시지 형식을 따로 지정해줘서, 프론트/백 측에서 각자 파싱을 해줘야하는 불편함이 있다. 반면, Stomp는 Websocket 기반이지만 Websocket과 달리 pub/sub 구조로 되어있으며, 클라이언트 간에 직접적인 메시지를 주고 받지 않고, 메시지 브로커가 발행된 메시지를 구독자에게 대신 전달해주는 구조이다. 이 덕분에 복잡한 네트워크 환경에서도 유연한 통신을 할 수 있고, 또한 메시지 브로커가 Websocket 연결 및 세션 관리와 같은 저수준의 세부사항을 대신 처리해주는 등 개발 측면에서 편리함이 있어 Stomp를 선택했다.

EC2, Docker, Github Actions

서버 배포를 위해서 AWS의 EC2를 선택했다. OS 호환 문제 때문에 Docker로 Spring-boot Application, Mysql, Redis 각각을 Docker 이미지로 맡아서, EC2 인스턴스에 컨테이너로 띄워서 통신하도록 구현할 예정이다. 또한 서버 배포를 할 때마다 복잡하고 반복되는 위 과정을 자동화하기 위해 Github Actions 을 사용한다. Git Flow 브랜치 전략에 따라 master, develop, feature 브랜치로 나뉘어져 있는데, master 브랜치에 변동사항이 있을 때, Github Actions을 통해 서버가 배포되도록 할 예정이다.



Company

회사에 대한 정보를 저장하는 테이블이다. **company** 테이블의 정보를 이용하여 회사 인증을 진행하고 사용자가 어떤 회사에 소속되어있는지 파악할 수 있다.

User

사용자 정보를 저장하는 테이블이다. 사용자 프로필에서 직무를 표시하기 위해 **position** 테이블과 연결한다. **userUUID**는 **loginID**를 유추하여 특정 URL에 접근하는 문제를 방지하는데 사용된다.

User_Chatroom

사용자와 채팅방 간의 관계를 나타내는 중간 테이블이다. 어떤 사용자가 어떤 채팅방에 있는지에 대한 정보를 나타낸다.

Chatroom

채팅방 정보를 저장하는 테이블이다. 메시징 기능이 구현된 서비스에서 채팅방을 만들고

 국민대학교 소프트웨어학부 캡스톤 디자인 I	중간보고서		
	프로젝트 명	커리어 한잔	
	팀 명	17조	
	Confidential Restricted	Version 2.1	2024-03-31

관리하기 위해 사용된다. 채팅방별 대화 기록을 유지하는 데에도 사용된다.

Message

채팅방 메시지를 저장하는 테이블이다. 이전 채팅 기록을 확인하고 싶을 때 사용된다.

Position

직무 테이블은 회사 내의 다양한 직무를 나타낸다. 백엔드, 프론트엔드, DevOps 등 직무를 설정할 수 있다. 현직자가 아닌 사람을 위한 학생, 무직 카테고리도 존재한다.

Company_Request

회사 요청 테이블은 회사 인증 시 소속된 회사가 없는 경우에 대한 요청 사항을 저장한다. 각 요청은 사용자 ID를 통해 특정 사용자와 연결된다.

Device

사용자의 디바이스 정보를 저장하는 테이블이다. 각 디바이스는 고유한 ID와 토큰을 가지고 있으며, 사용자 ID를 통해 특정 사용자에게 속해 있음을 나타낸다. 디바이스 테이블을 통해 여러 기기에 대한 접속을 관리한다.

Review

커피콩 매너온도를 설정하기 위한 정보를 저장하는 테이블이다. 사용자 간의 상호작용과 평가를 기록하여 서비스 개선에 기여할 수 있다.

 국민대학교 소프트웨어학부 캡스톤 디자인 I	중간보고서		
	프로젝트 명	커리어 한잔	
	팀 명	17조	
	Confidential Restricted	Version 2.1	2024-03-31

2.2. 수행 내용

캡스톤 17팀			3월				4월				5월			
		진행 여부	1주차	2주차	3주차	4주차	1주차	2주차	3주차	4주차	1주차	2주차	3주차	4주차
기 획	프로젝트 일정													
	기술 스택 조사	Done												
	시장 조사 & 분석	Done												
	와이어프레임 제작	Done												
	아이디어 발표	Done												
	아이디어 피드백	Done												
	기획안 구체화	Done												
개 발	개발환경 세팅	Done												
	유저 회원가입 구현	Done												
	카페 상세 정보 페이지	Done												
	유저 로그인/로그아웃	Done												
	유저 탈퇴	Done												
	Swagger 연동	Done												
	Google Map 연동	Done												
	지도 불러오기	Done												
	Google Place API 연동	In Progress												
	webSocket 초기화	In Progress												
	카카오로그인 API 연동	Done												
	카카오로그인 기능 구현	In Progress												
	반경 내의 카페 불러오기	In Progress												
	커피챗 요청 화면 제작	In Progress												
	회사 인증	Todo												
	회사 로고 매칭	Todo												
	AWS 배포	Todo												
	커피챗 off 기능 구현	Todo												
	FCM 초기화	Todo												
	CI/CD 설정	Todo												
테 스 트	Redis 사용자 위치값 저장	Todo												
	Redis 요청 정보 저장	Todo												
	.													
	.													
	스프링 프로젝트 초기화	Done												
	유저 회원가입	In Progress												
현 시 점	유저 탈퇴	In Progress												
	Swagger 연동	In Progress												
	Google API Test	In Progress												
	Redis 초기화	Todo												
	Redis 요청 정보 저장	Todo												

회원관리 기능

- 프론트엔드

로그인 ViewModel은 Provider 형태로 구성하여 각 위젯이 로그인 관련 정보를 공유할 수 있도록 했다. ViewModel 내부의 user 인스턴스 존재 여부에 따라 로그인 상태가 관리된다. 로그인 또는 로그아웃이 이루어지면 ViewModel에 등록된 메소드에서 Listener 위젯들에게 해당 정보를 notify한다. 이 설계는 소셜 로그인 기능으로의 확장성을 고려하여 구성되었다.

 국민대학교 소프트웨어학부 캡스톤 디자인 I	중간보고서		
	프로젝트 명	커리어 한잔	
	팀 명	17조	
	Confidential Restricted	Version 2.1	2024-03-31

서버에서 클라이언트로 넘어온 토큰은 flutter의 flutter_secure_storage 패키지를 활용하여 저장했다. 시스템의 내부 저장소인 keystore 영역에 토큰을 저장하여, 앱을 루팅하더라도 토큰에 접근할 수 없도록 했다.

- 백엔드

JWT 토큰을 사용하여 로그인 인증 기능을 구현했다. 사용자가 로그인할 때 사용자의 UUID와 JWT 토큰을 저장한다. 생성된 JWT 토큰은 클라이언트에게 반환되고, 클라이언트는 이 토큰을 인증된 요청에 포함하여 서버로 전송한다.

클라이언트는 JWT 토큰을 포함해서 요청을 보낸다. 이때, 서버에서 토큰의 유효성을 검증한다. 토큰이 유효한 경우, 요청을 계속 진행하고, 유효하지 않은 경우 예외를 발생시켜 요청을 거부한다. 토큰의 유효성은 토큰의 서명을 확인하고, 만료 시간을 확인하여 수행된다.

컨트롤러 메서드 파라미터로 '@AuthenticationPrincipal' 어노테이션을 추가하여 JWT 토큰을 추출한다. 이를 바탕으로 사용자를 식별하고, 필요한 경우 해당 사용자 객체를 반환하여 컨트롤러 메서드에서 사용할 수 있도록 한다.

커피챗 요청

- 프론트엔드

커피챗 요청하기까지의 UI(유저 선택 -> 유저 상세 프로필 -> 커피챗 요청)를 구현하다, 다이얼로그 전환과 관련해 문제가 발생했다. 보통 플러터에서 페이지를 전환할 때는 route와 Navigator를 사용하는데, 이는 화면에 빌드되는 전체 페이지에 해당되는 것이다. 따라서 다이얼로그(전체 화면이 아닌 팝업과 같은 형식)를 전환할 때 해당 방법을 사용할 경우, 뒤에 깔려 있는 페이지마저 함께 전환돼 사라져버리는 문제가 발생했다.

 국민대학교 소프트웨어학부 캡스톤 디자인 I	중간보고서		
	프로젝트 명	커리어 한잔	
	팀 명	17조	
	Confidential Restricted	Version 2.1	2024-03-31

정확히 다이얼로그 전환에 대한 자료는 찾지 못했으나, 다이얼로그의 내용을 바꾸는 것에 대한 글을 발견하였고 거기서 힌트를 얻었다. 두 개의 다이얼로그를 하나의 상태로 관리하여, 버튼을 클릭할 때마다 상태를 업데이트해서 전환을 하도록 하여 구현해볼 예정이다.

- Google Map 관련

화면에 지도를 띄우는 과정에서 지도가 하얗게 나타나는 현상이 발생했다. **android**와 **IOS**의 **API Key**가 분리가 되지 않음을 파악했고 **Key**를 동일하게 수정하여 해결할 수 있었다.

다음은 **Google Map** 현재 위치 가져오는 과정에서의 문제이다.

flutter의 **emulator**에서는 **GPS** 기능을 지원하지 않기 때문에 자신의 위치를 자동으로 읽을 수 없다. 따라서 현재 위치를 불러오는 기능을 수행할 때 엉뚱한 위치로 이동한다. 해당 기능을 **android** 휴대폰과 연동하여 테스트가 필요한 상태이다.

현재는 내 위치와 주변 카페의 위치를 임시의 데이터로 만들어두고 개발을 진행중이다. 카페의 위치를 **list**에서 불러와 지도에 마커 및 이름을 띄우는 작업을 완료하였다. **Google Map Places Api**를 사용하여 주변의 카페 정보를 불러 오는 부분을 구현중에 있다.

- 에러들

- flutter의 navigate 관련 에러

이미 로그인되어있는 상태에서 로그인 화면에 접근했을 때 **user** 화면으로 **navigate**되도록 설정했는데, 이 과정에서 **'_debuglocked' is not true** 에러가 발생했다. 이 오류는 **push**나 **pop** 등의 화면이동 함수가 중첩되어 화면이동 함수에 재진입(**re-entrant**) 현상이 일어나서 발생한 오류였다. 이 문제를 해결하기 위해, **user** 화면으로 **navigate**되는 코드를 **delay**를 0으로 설정하여 지연시켰다. 이를 통해 해당 호출 스택이 리턴되고 나면 그때 즉시 스케줄링되도록 했다.

 국민대학교 소프트웨어학부 캡스톤 디자인 I	중간보고서		
	프로젝트 명	커리어 한잔	
	팀 명	17조	
	Confidential Restricted	Version 2.1	2024-03-31

3. 수정된 연구내용 및 추진 방향

3.1. 수정 사항

- 위치 공유 방식의 변경

가까운 위치의 사용자들끼리 커피챗 만남을 가질 수 있도록 위치를 공유하게 되는데, 그 위치 공유 방식이 기획 단계에서 크게 바뀌었다. 초반 기획에서는 실시간으로 사용자의 실제 위치를 공유하여 더욱 생생한 실시간 커피챗 매칭 앱을 만들고자 했으나, 특정 사용자의 위치를 악의를 가진 다른 사용자가 추적할 수 있게 된다는 점에서 스토킹 범죄에 악용될 우려가 있었다.

이러한 문제점을 해소하고자 차단 기능, 프로필 사진 제거 등 여러 가지 방안들이 나왔으나 실시간으로 위치가 공유된다는 위험을 완전히 제거할 수 없었다. 그래서 우리 서비스에 있어 가장 핵심적이라고 생각했던 실시간 위치 공유 기능을 버리는 큰 결정을 하게 되었다.

여전히 서비스가 추구하는 목표를 달성하기 위해, 실시간 위치 공유 대신 간접적으로 위치를 공유하는 방안을 채택하게 되었다. 즉 사용자는 실제 자신의 위치를 실시간으로 공유하는 대신, 현재 위치를 기반으로 반경 **500m** 내에 있는 카페 중 자신이 선택한 곳을 자신의 위치로 공유하게 된다. 또한 선택한 카페가 사용자의 위치 반경에서 벗어나는 경우에만 다시 카페를 선택하여 위치가 변경된다. 이렇게 되면 위치를 추적당할 위험이 없어져 스토킹 범죄에 악용될 우려가 해소되면서도, 내 위치를 기반으로 가까운 사람들과 커피챗을 할 수 있다는 점에서 기존의 목표를 달성할 수 있다.

또한 실시간 위치 공유는 서로 매칭된 이후 만날 장소를 일일이 찾아보고 약속을 잡아야 하는 번거로움이 있는 반면, 이 방식은 카페를 자신의 위치로 보여줌으로써 같은 카페를 선택한 사람에게 커피챗 요청을 할 경우 바로 해당 카페로 커피챗 장소를 정할 수 있다는 점에서 사전 약속 절차를 간소화한다는 사용자 관점의 이점이 있다.

 국민대학교 소프트웨어학부 캡스톤 디자인 I	중간보고서		
	프로젝트 명	커리어 한잔	
	팀 명	17조	
	Confidential Restricted	Version 2.1	2024-03-31

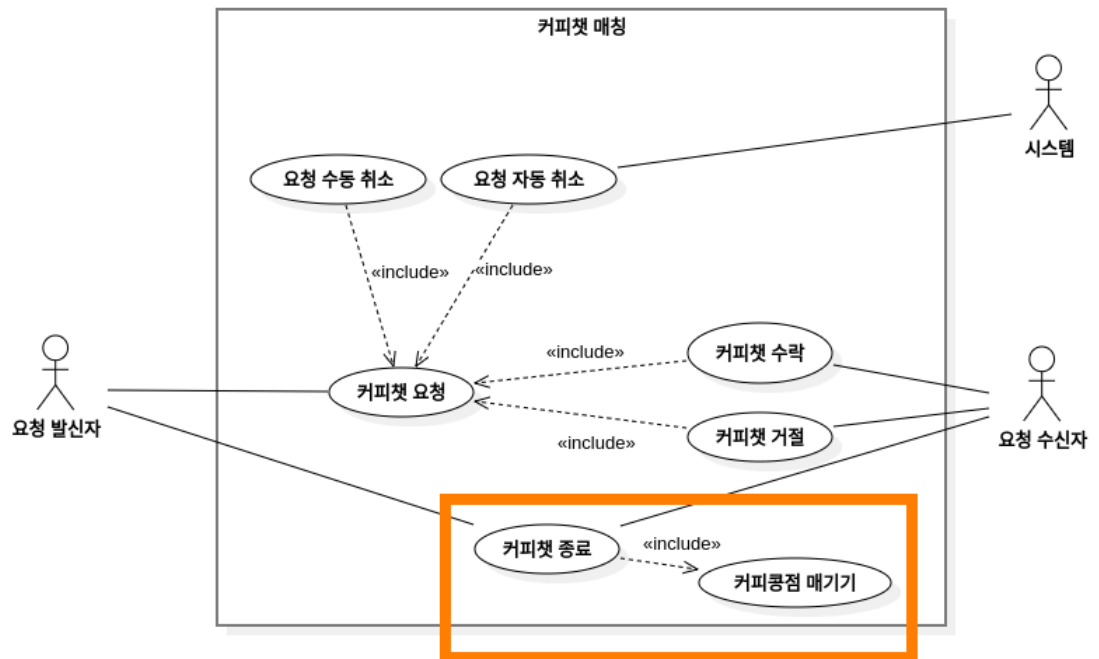
현재 구글 맵 API를 불러와 지도를 띄우고, GPS로 사용자의 위치를 가져오는 기능을 구현했다. 사용자 반경 내의 카페 정보를 가져와 지도에 띄우고, 사용자가 카페의 상세정보와 사용자 목록을 보고 자신의 위치로 표시될 카페를 선택할 수 있도록 구현할 예정이다.

- '커피 온도' 시스템 도입

앱을 통해 서로 모르는 사람끼리 오프라인 커피챗 약속을 잡는 경우, 만나기로 한 장소에 나타나지 않는 '노쇼' 등 약속을 제대로 지키지 않거나, 만났을 때 예의를 벗어난 행동을 하는 사용자가 있을 수 있다. 이러한 부작용을 해소하고자 '당근마켓' 서비스의 매너 온도 시스템에서 힌트를 얻어 '커피 온도' 시스템을 도입하였다.

모든 사용자는 자신만의 커피챗 평가 점수 혹은 평판과도 같은 '커피 온도계'를 가지며, 이는 다른 사용자들과 공유되어 커피챗 요청을 보내거나 받은 요청을 수락할 때 참고할 수 있는 자료가 된다. 따라서 사용자들은 자연스럽게 예의를 지키기 위해 스스로 노력하게 되고, 만약 그렇지 못한 사용자는 커피챗 성사가 어려워지므로 앱을 사용하는 사용자들의 불편을 덜 수 있을 것이라 기대한다.

'커피 온도'는 커피챗이 종료된 후에 상대방이 나와의 커피챗에 대해 매긴 '커피콩점(=별점)'을 일정 비율씩 반영해 올라가거나 내려가도록 구현할 예정이다.



 국민대학교 소프트웨어학부 캡스톤 디자인 I	중간보고서		
	프로젝트 명	커리어 한잔	
	팀 명	17조	
	Confidential Restricted	Version 2.1	2024-03-31

4. 향후 추진계획

4.1. 향후 계획의 세부 내용

향후 계획으로는 완성도 있는 최종 결과물을 위해 개발 규칙을 강화할 예정이다. 또한, 아키텍처를 구체화 하여 프로젝트가 빠르게 진행되도록 할 예정이다.

중간 평가 마일스톤에 계획했던 기능 중 아직 끝내지 못한 기능들을 빠르게 마무리하고, 커피 온도 기능을 추가하여 프로젝트 계획서를 수정한다. 이어서 나머지 기능들을 구현한다.

중간 발표 이후에는 구글 설문지를 활용해 사용자 피드백을 수렴하여 최종 프로젝트가 사용자들의 요구와 기대를 충족시킬 수 있도록 개선할 예정이다.

협업 측면에서는 GitHub에서 Todo에 추가할 내용의 단위를 Commit 혹은 PR 중에 확정하여 유지하거나 수정할 계획이다.

 국민대학교 소프트웨어학부 캡스톤 디자인 I	중간보고서		
	프로젝트 명	커리어 한잔	
	팀 명	17조	
	Confidential Restricted	Version 2.1	2024-03-31

5. 고충 및 건의사항

- 잘한점

일주일에 두 번 정기적으로 회의를 진행하였고, **figma**와 노선을 활용하여 회의 내용을 상세하게 정리해두었다.



- 고칠점

팀 구성원들 간의 피드백은 충분했지만, 초기에 요구 사항과 기술적인 부분을 명확하게 하지 않아 예상하지 못했던 제약사항이 발생했다. 이로 인해 기획이 자주 변경되어 개발을 다소 늦게 시작하게 되었다.