

캡스톤디자인 중간보고서

제 목	국문	스마트 주차장 검색 및 경로 안내 위치기반 실시간 애플리케이션			
	영문	Smart Parking Lot Search and Navigation Application			
진 행 상 황	중요마일스톤	<ul style="list-style-type: none">- 기초 아키텍처 작성 (시스템 구성도)- Map API 연동- 실시간 주차 차량 대수 및 공유 모빌리티의 사용 현황을 API에 요청하여 업데이트- 주차장 정보 즐겨찾기를 위한 DB 설계 구축- 주차장 정보 API 통합- 지도에 주차장 위치 표시- 주차장 카테고리 별 리스트 구현- 내비게이션 서비스 구현			
	진행상황	<ul style="list-style-type: none">- 기초 아키텍처 작성 완료- Kakao API를 이용해 지도 연동 및 현재 위치 반영 완료- 모바일 앱 프로토타입 제작 완료- 로그인, 회원가입 서버 구축 완료			
산출물	요구사항 정의서(별첨 1), 중간보고서(별첨 2)				
팀 구성원	학년	학 번	이 름	연락처(전화번호/이메일)	
	4	20211906	김민서	010-9724-5578 / 20211885@edu.hanbat.ac.kr	
	4	20211885	윤채원	010-5822-2153 / 20211906@edu.hanbat.ac.kr	
	4	20211918	유진주	010-4405-0838 / 20211918@edu.hanbat.ac.kr	
컴퓨터공학과의 프로젝트 관리규정에 따라 다음과 같이 요구사항 정의서와 중간보고서를 제출합니다					
2024년 5월 일					
책임자 : 김 민 서 김민서 지도교수 : 이 상 금 (인)					

[별첨1]

프로젝트명 : 스마트 주차장 검색 및 경로 안내 위치기반 실시간 애플리케이션

소프트웨어 요구사항 정의서

Version 1.0

개발 팀원 명(팀리더):김민서

윤채원

유진주

대표 연락처: 010-5822-2153

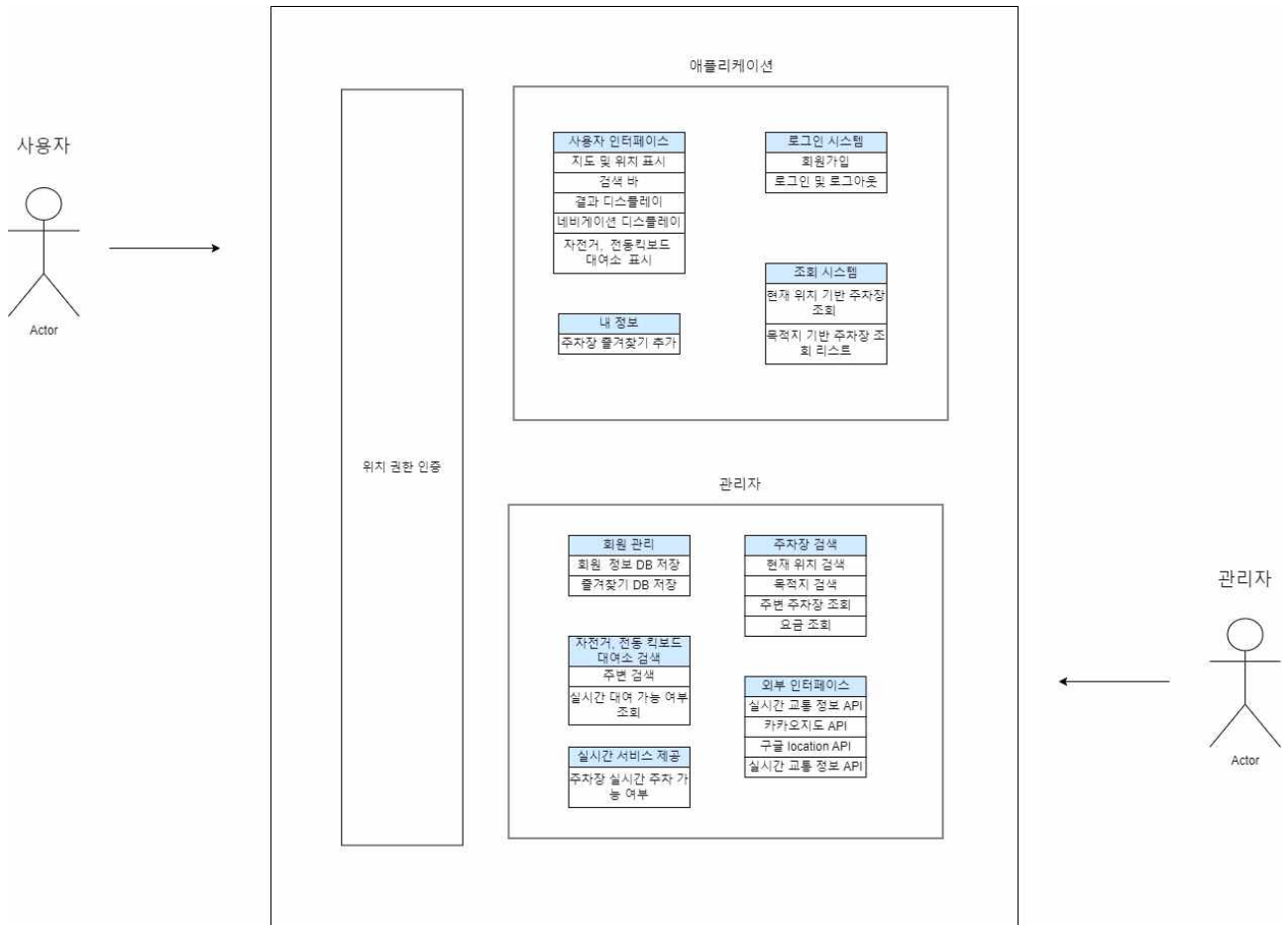
e-mail: 20211906@edu.hanbat.ac.kr

목차

1. 개요
2. 시스템 장비 구성 요구사항
3. 기능 요구사항
4. 성능 요구사항
5. 인터페이스 요구사항
6. 데이터 요구사항
7. 테스트 요구사항
8. 보안 요구사항
9. 품질 요구사항
10. 제약 사항
11. 프로젝트 관리 요구사항

1. 시스템 개요

[시스템 구성도]



2. 시스템 장비 구성요구사항

요구사항 고유번호		ECR-001		
요구사항 명칭		애플리케이션		
요구사항 분류		시스템 장비 구성 요구 사항	응락수준	필수
요구사항 상세설명	정의	- 애플리케이션 UI, 서버 연동		
	세부 내용	- 장비 품목: Android Studio(language:JAVA) - 장비 수량: 2RO - 장비 기능: 시각적 레이아웃 제작, 애플리케이션 테스트 수행 - 장비 성능: 사용자에게 제공하는 UI를 제작하고 서버에게 사용자 정보를 전달		

요구사항 고유번호		ECR-002		
요구사항 명칭		테스트용 기기		
요구사항 분류		시스템 장비 구성 요구 사항	응락수준	필수
요구사항 상세설명	정의	- Android Studio 테스트용 기기		
	세부 내용	- 장비 품목: Galaxy S10 - 장비 수량: 1개 - 장비 기능: Android Studio의 에뮬레이터 대신 시각적 레이아웃과 기능 테스트 수행		

3. 기능 요구사항

요구사항 고유번호		SFR-001		
요구사항 명칭		메인화면 지도		
요구사항 분류		기능 요구사항	응락수준	필수
요구사항 상세설명	정의	- 홈 화면 지도		
	세부 내용	- 사용자의 현재 위치를 마커로 표시함 - 주차장, 자전거, 킥보드의 아이콘을 누르면 해당 주차장만 마커로 위치를 표시함 - 사용자의 위치에서 선택한 주차장으로 경로 안내 제공함		

요구사항 고유번호		SFR-002		
요구사항 명칭		주차장 검색		
요구사항 분류		기능 요구사항	응락수준	필수
요구사항 상세설명	정의	- 사용자가 원하는 주차장 검색		
	세부 내용	- 사용자가 원하는 특정 주소, 관심 영역을 검색함 - 사용자가 위치를 빠르게 찾을 수 있도록 자동완성 구현함 - 빠른 재접속을 위해 최근 검색어 목록 표시함		

요구사항 고유번호		SFR-003		
요구사항 명칭		다양한 카테고리 시각화		
요구사항 분류		기능 요구사항	응락수준	필수
요구사항 상세설명	정의	- 사용자가 원하는 카테고리로 시각화		
	세부 내용	- 사용자의 검색 기준에 따라 주차장 데이터를 주기적으로 API를 요청함 - 데이터를 가져온 후 정렬 알고리즘을 사용하여 가격별로 정렬함 - 상세보기에는 가격, 운영시간, 주차공간 수, 실시간 이용 가능 여부 등 각 주차장에 대한 자세한 보기 제공함		

요구사항 고유번호		SFR-004		
요구사항 명칭		로그인		
요구사항 분류		기능 요구사항	응락수준	필수
요구사항 상세설명	정의	- 앱을 사용하기 위한 로그인		
	세부 내용	- 사용자가 ID, 비밀번호, 이메일을 입력할 수 있는 양식 디자인함 - 비밀번호 분실 시 사용자가 비밀번호를 복구하거나 재설정할 수 있는 안전한 방법 구현함 - 개인정보 보호정책 및 서비스 약관에 대한 링크를 명확하게 표시하여 사용자가 계정을 만들기 전에 이에 동의하는지 확인함		

요구사항 고유번호		SFR-005		
요구사항 명칭		내비게이션		
요구사항 분류		기능 요구사항	응락수준	필수
요구사항 상세설명	정의	- 사용자가 검색한 주차장으로 경로 안내		
	세부 내용	- 지도에 경로를 시각적으로 표시하고 단계별 텍스트 탐색 지침 제공 - 교통상황을 고려하여 이동시간을 추정함 - 사용자가 경로를 따라 진행하면 지도가 실시간으로 업데이트되어 교통량이나 사용자 우회로의 변화로 인해 필요에 따라 경로 다시 계산함 - 현재 속도 제한을 표시하고 사용자가 사용하는 제한 속도를 초과하면 경고 제공함		

요구사항 고유번호		SFR-006		
요구사항 명칭		즐거찾기 시각화		
요구사항 분류		기능 요구사항	응락수준	필수
요구사항 상세설명	정의	- 사용자가 즐겨찾기한 목록 시각화		
	세부 내용	- 별 모양 아이콘이나 북마크 라벨 버튼 등 주차장을 즐겨찾기할 수 있는 옵션 제공함 - 즐겨찾기 아이콘을 탭하면 다시 검색할 필요 없이 빠르게 길찾기를 하거나 주차장에 대한 세부 정보 확인 가능함		

4. 성능 요구사항

요구사항 고유번호		PER-001		
요구사항 명칭		느린 작업에 대한 대처		
요구사항 분류		성능 요구사항	응락수준	필수
요구사항 상세설명	정의	- 데이터 처리		
	세부 내용	- 데이터 처리 기능은 3초 정도 걸릴 것으로 예상함 - 데이터를 받아오는 중 시간이 오래걸릴 경우 로딩창을 띄워 사용자에게 알림		

요구사항 고유번호		PER-002		
요구사항 명칭		앱 용량 최소화		
요구사항 분류		성능 요구사항	응락수준	필수
요구사항 상세설명	정의	- 스마트폰 저장공간 활용		
	세부 내용	- 불필요한 파일을 제거하여 앱의 용량을 최소화함으로써 사용자에게 편의성을 제공함		

5. 인터페이스 요구사항

1) 사용자 인터페이스 요구사항 분석 및 도출

요구사항 고유번호		SIR001		
요구사항 명칭		사용자 위치 표시		
요구사항 분류		인터페이스	응락수준	필수
요구사항 상세 설명	정의	<ul style="list-style-type: none"> - 사용자 위치 제공 		
	세부 내용	<ul style="list-style-type: none"> - 애플리케이션이 사용자의 현재 위치를 정확하게 감지하고 지도 상에 표시할 수 있어야 함 - 사용자의 위치 권한을 요청하는 명확한 방법을 제공해야 함 - 사용자가 위치 권한을 거부할 시 권한을 요청하는 메시지를 띄우거나 애플리케이션을 종료함 		
주석		<ul style="list-style-type: none"> - 애플리케이션이 사용자의 위치를 실시간으로 확인하고, 이를 지도 상에 아이콘으로 표시함 - 사용자의 위치 권한 요청과 관련된 UI/UX 설계가 필요함 		
요구사항 출처				
요구사항 고유번호		SIR002		
요구사항 명칭		주차장(자전거, 전동킥보드 등) 검색 및 표시		
요구사항 분류		인터페이스	응락수준	필수
요구사항 상세 설명	정의	<ul style="list-style-type: none"> - 주차장 검색 및 표시 기능 		
	세부 내용	<ul style="list-style-type: none"> - 사용자의 현재 위치나 설정한 목적지를 기반으로 주변의 주차장을 자동으로 탐색하고 지도 상에 마커로 표시함 - 검색된 목적지를 지도 중심으로 설정하고 해당 지역의 주차장을 새롭게 탐색하여 표시함 - 각 주차장 마커를 누를 시 해당 주차장의 이름, 위치, 요금 등의 상세 정보를 제공함 - 지도상에 이동 수단(자전거, 전동킥보드 등)에 대한 위치를 마커로 표시하고 해당 마커를 누를 시 사용 여부(배터리 등)에 대한 정보를 제공함 - 제공하는 내용들은 사용자가 알아보기 쉽도록 제공해야 함 		
주석		<ul style="list-style-type: none"> - 사용자의 현재 위치 또는 목적지 중심 위치를 기준으로 가까운 주차장을 탐색하고, 이를 마커로 표시함 - 마커 선택 시 주차장의 상세 정보를 팝업으로 나타냄 		
요구사항 출처				
요구사항 고유번호		SIR003		
요구사항 명칭		내비게이션 및 실시간 정보 통합		
요구사항 분류		인터페이스	응락수준	필수
요구사항 상세 설명	정의	<ul style="list-style-type: none"> - 목적지까지 경로 안내 		
	세부 내용	<ul style="list-style-type: none"> - 사용자가 목적지까지의 경로를 제공하여 실시간 도로 상황을 파악하여 사용자에게 최적의 경로를 제공해야 함 		
주석		<ul style="list-style-type: none"> - 사용자가 특정 주차장을 선택하면 그 위치까지의 최적 경로를 계산하여 길 안내를 시작함 - 실시간 교통 정보를 반영하여 최적의 경로를 제공하며 경로 옵션 선택 기능(최소 시간, 최소 거리 등)을 포함함 		
요구사항 출처				

요구사항 고유번호		SIR004		
요구사항 명칭		사용자 설정 및 개인화		
요구사항 분류		인터페이스	응락수준	필수
요구사항 상세 설명	정의	- 사용자 설정에 따른 개인화		
	세부 내용	- 사용자가 로그인하여 자주 방문하는 위치를 저장하여 즐겨찾기 해놓는 등 사용자 개인화를 해야 함		
주석		- 사용자가 자주 가는 목적지를 즐겨찾기로 관리할 수 있는 기능을 구현함 - 사용자의 선호에 따라 주차장을 필터링하고 추천하는 기능을 포함		
요구사항 출처				
요구사항 고유번호		SIR005		
요구사항 명칭		인터페이스 디자인		
요구사항 분류		인터페이스	응락수준	필수
요구사항 상세 설명	정의	- 간결하고 명확한 사용자 인터페이스		
	세부 내용	- 간결하고 명확한 사용자 인터페이스를 통해 정보를 쉽게 이해하고 조작할 수 있도록 해야 함 - 터치스크린 최적화 디자인을 고려하여 적절한 크기와 영역으로 사용자의 편의성을 보장함		
주석		- 모바일 환경에 최적화된 디자인을 구현함 - 사용자가 쉽게 조작할 수 있도록 명확한 버튼, 한눈에 기능을 알아볼 수 있는 아이콘, 충분한 터치 영역을 제공함		
요구사항 출처				

2) 시스템 인터페이스 요구사항 분석 및 도출

요구사항 고유번호		SIR006		
요구사항 명칭		지리 정보 시스템 연동		
요구사항 분류		인터페이스	응락수준	필수
요구사항 상세 설명	정의	- 위치기반 데이터와 정보 수집 및 표시		
	세부 내용	- 애플리케이션이 정확한 지리적 데이터를 표시하고 처리할 수 있도록 지리 정보 시스템과의 인터페이스가 필요함		
기타 고려 사항		- 정확성 및 업데이트 주기: 데이터는 최신 상태를 유지해야 하며 정확해야 함. 데이터의 정기적인 업데이트와 정확성 확인 절차가 필요함		
산출정보		- 통합 설계 문서, 업데이트 및 유지보수 계획		
요구사항 출처				

요구사항 고유번호		SIR007		
요구사항 명칭		지리 정보 시스템 연동		
요구사항 분류		인터페이스	응락수준	필수
요구사항 상세 설명	정의	- 서비스가 제공하는 데이터에 접근할 수 있는 수단 제공		
	세부 내용	- 주차장 정보, 주차장 요금 등을 제공하는 외부 API와의 연동이 필요함 - 이동 수단(자전거, 전동킥보드 등)의 위치, 현재 이용 가능 여부 등을 제공하는 외부 API와의 연동이 필요함 - 사용자 개인화(즐거찾기)를 위하여 외부 데이터베이스와의 연동이 필요함		
기타 고려 사항		- 데이터 일관성 및 신뢰성: 외부 API를 통해 접근하는 데이터는 신뢰할 수 있는 출처에서 가져와야 함 - API 의존성 관리: 외부 API의 변경이나 중단이 애플리케이션의 성능에 영향을 미치지 않도록 관리가 중요함		
산출정보		- API 통합 설계 문서		
요구사항 출처				

요구사항 고유번호		SIR008		
요구사항 명칭		실시간 교통 정보 통합		
요구사항 분류		인터페이스	응락수준	필수
요구사항 상세 설명	정의	- 실시간 교통 정보를 제공		
	세부 내용	<ul style="list-style-type: none"> - 도로 상황, 교통 체증 등 실시간 교통 정보를 제공하는 외부 서비스와의 통합이 필요함 - 이 정보는 경로 안내 시 사용자에게 최적의 경로를 제공하는 데 사용함 		
기타 고려 사항		<ul style="list-style-type: none"> - 시간적 정확성: 교통 정보는 실시간성이 매우 중요하므로 정보의 즉각적인 업데이트와 전송이 필수적임 		
산출정보		<ul style="list-style-type: none"> - 실시간 데이터 통합 설계 문서 - 성능 및 안정성 평가 보고서 		
요구사항 출처				

요구사항 고유번호		SIR009		
요구사항 명칭		사용자 인증 및 보안		
요구사항 분류		인터페이스	응락수준	필수
요구사항 상세 설명	정의	- 사용자 확인 인증		
	세부 내용	<ul style="list-style-type: none"> - 사용자 데이터를 보호하기 위해 사용자 인증을 통한 보안 시스템 구축이 필요함 - 사용자의 개인 정보 보호와 데이터의 안전한 처리를 보장함 		
기타 고려 사항		<ul style="list-style-type: none"> - 데이터 암호화: 사용자 데이터, 특히 위치 정보와 같은 민감한 정보는 전송 및 저장 단계에서 암호화되어야 함 		
산출정보		<ul style="list-style-type: none"> - 보안 및 인증 설계 문서 - 인증 실패 및 보안 사고 대응 계획 		
요구사항 출처				

6. 데이터 요구사항

요구사항 고유번호		DAR-001		
요구사항 명칭		데이터 관리		
요구사항 분류		데이터	응락수준	필수
요구사항 상세 설명	정의	- 시스템에서 사용되는 모든 데이터의 저장, 액세스 요구사항		
	세부 내용	<ul style="list-style-type: none"> - 회원 정보 데이터를 DB에 저장함 - 각 회원별로 즐겨찾기한 주차장 데이터를 DB에 저장함 - 사용자에게 표시되는 모든 정보는 최신 상태로 유지되어야 하며, 데이터는 정기적으로 갱신함 		

7. 테스트 요구사항

요구사항 고유번호		TER001		
요구사항 명칭		테스트 방안	응락수준	필수
요구사항 분류		테스트		
요구사항 세부내용		<ul style="list-style-type: none"> - 단위 테스트: 개별 모듈 또는 함수의 기능을 테스트로 모든 함수와 메소드에 대해 테스트 케이스를 작성하고 예상되는 입력과 출력을 통해 코드가 정확하게 동작하는지 확인함 - 통합 테스트: API 연동, 데이터베이스 접근 등의 기능이 요구사항에 맞게 정상적으로 작동하는지 확인함 - 시스템 테스트: 완성된 시스템이 전체적으로 요구사항을 충족하는지, 사용자 요구사항, 기능성, 시스템의 동작 등 전체적인 측면을 테스트함 - 성능 테스트: 시스템의 반응시간, 처리량 등을 측정하고 과부하 상황에서 안정성과 최대 성능 한계를 파악함 - (테스트 환경) Wi-Fi, 4G, 5G 등 다양한 네트워크 환경에서의 테스트를 지원해야 하며 네트워크 속도 및 연결 안정성 변화에 따른 앱의 반응을 테스트함. 실제 운전 경로를 모의하여 앱의 내비게이션 기능(경로 변경, 교통상황 반영 등)을 테스트함. 다양한 테스트 케이스를 지원하기 위한 테스트용 데이터베이스를 준비하고 현실적인 데이터로 구성해야 함, 외부 시스템과의 통신을 검증하기 위해 안전한 API 호출을 시뮬레이션하고 테스트함 		
		[테스트 방법 및 절차] <ul style="list-style-type: none"> - 테스트 계획 수립: 기능 테스트, 성능 테스트를 포함하여 테스트의 목적, 범위, 목표를 명확히 해야 함 - 테스트 환경 설정: 지도 및 내비게이션 애플리케이션에 필요한 환경을 준비하고 구성하여 테스트에 필요한 데이터를 준비함 - 테스트 케이스 및 시나리오 설계: 실제 사용자의 사용 패턴을 모방한 시나리오를 생성하여 애플리케이션을 평가함 - 테스트 실행: 발견된 모든 버그를 기록하고 우선순위를 매겨 해결함 - 결과 분석 및 보고: 테스트를 분석하여 애플리케이션의 성능, 안정성을 평가하고 테스트 결과를 문서화하여 상세한 테스트 보고서를 작성함 - 최종 검증: 최종 사용성 및 성능 테스트를 실시함 		

8. 보안 요구사항

요구사항 고유번호		SER-001		
요구사항 명칭		시스템 및 데이터 보안		
요구사항 분류		보안	응락수준	필수
요구사항 상세설명	정의	<ul style="list-style-type: none"> - 시스템과 데이터 보안을 확보하여 사용자 정보와 시스템 데이터를 보호 		
	세부 내용	<ul style="list-style-type: none"> - 토큰 기반 인증 시스템을 사용하여 사용자 세션을 관리함 - 사용자 패스워드와 같은 민감한 데이터는 암호화함 - 사용자의 패스워드는 해시 알고리즘을 사용하여 안전하게 저장되어야 함 		

9. 품질 요구사항

요구사항 고유번호		QUR-001		
요구사항 명칭		사용성		
요구사항 분류		품질	응락수준	필수
요구사항 상세설명	정의	- 사용자 인터페이스		
	세부 내용	- 사용자 인터페이스는 직관적이고 사용자 친화적이어야 하며, 모든 주요 기능은 쉽게 접근하고 사용할 수 있어야 함		
요구사항 고유번호		QUR-002		
요구사항 명칭		유지보수성		
요구사항 분류		품질	응락수준	필수
요구사항 상세설명	정의	- 코드 유지보수성		
	세부 내용	- 시스템은 변경 요구사항에 신속하게 대응할 수 있어야 하며, 새로운 기능 추가나 기존 기능 수정이 용이해야 함		
	내용	- 코드는 명확하고, 문서화되며, 모듈화된 구조를 가져야 하며, 유지보수를 용이하게 하는 설계 원칙을 따름		
요구사항 고유번호		QUR-003		
요구사항 명칭		성능		
요구사항 분류		품질	응락수준	필수
요구사항 상세설명	정의	- 프로그램 성능		
	세부 내용	- 시스템은 부하 조건 하에서 최소한의 자원을 사용하여 최대의 성능을 제공함		
	내용	- 응답시간과 처리량은 사전에 정의된 벤치마크를 충족하거나 초과해야 함		

10. 제약 사항

요구사항 고유번호		COR-001		
요구사항 명칭		시스템 개발		
요구사항 분류		제약 사항	응락수준	필수
요구사항 상세설명		<ul style="list-style-type: none"> - 프로그래밍 언어: Java - 프레임 워크: Spring Boot - 개발 환경: IntelliJ, Android Studio - 데이터베이스 관리 시스템: MySQL - Android 기반 스마트폰에서만 운영되며, iso나 다른 모바일 OS 지원은 현재 범위 밖임 		

11. 프로젝트 관리 요구사항

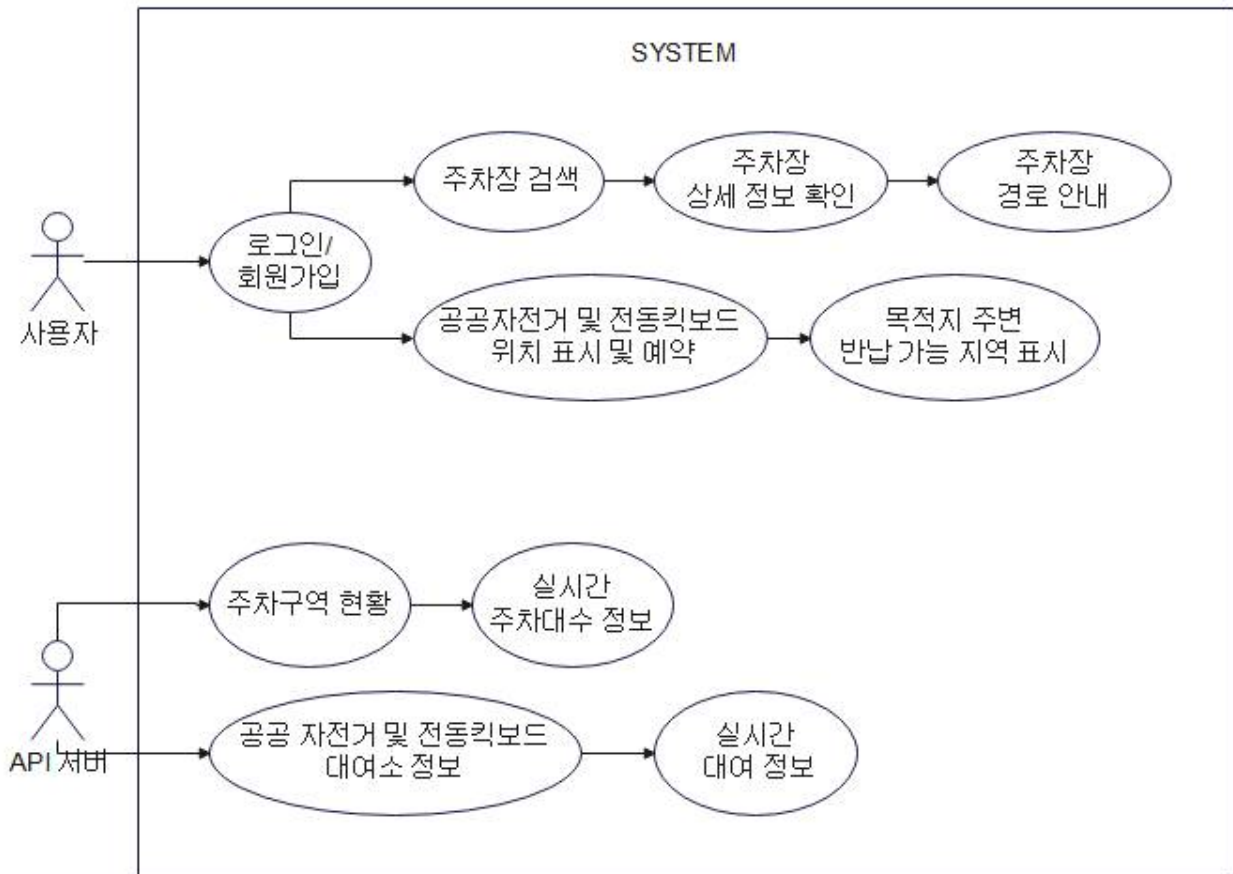
요구사항 고유번호	PMR-001		
요구사항 명칭	프로젝트 조직		
요구사항 분류	프로젝트 관리	응락수준	필수
요구사항 상세설명	<ul style="list-style-type: none"> - 김민서: 팀장, 프론트엔드/ UXUI - 유진주: 팀원, 프론트엔드/ 산출물, 자료조사 - 윤채원: 팀원, 백엔드/ 회의록, 품질 		
요구사항 고유번호	PMR-002		
요구사항 명칭	프로젝트 일정 계획		
요구사항 분류	계약 사항	응락수준	필수
요구사항 상세설명	<ul style="list-style-type: none"> - 로그인 및 회원가입 기능: 5월 이전 - 목적지 검색 서비스: 6월 이전 - 주차장 정보 API 통합: 7월 이전 - 주차장 위치 표시 및 카테고리별 리스트: 8월 이전 - 즐겨찾기 서비스: 8월 이전 - 네비게이션 서비스: 9월 이전 		
요구사항 고유번호	PMR-003		
요구사항 명칭	품질		
요구사항 분류	프로젝트 관리	응락수준	필수
요구사항 상세설명	<ul style="list-style-type: none"> - 프로젝트의 품질 기준은 명확히 정의되며, 이는 코드 품질, 시스템 성능, 사용자 경험 등을 포함함 - 모든 개발 단계에서 코드 리뷰, 단위 테스트, 통합 테스트를 실시하고 알파 테스트를 통해 최종 제품의 품질을 검증함 		
요구사항 고유번호	PMR-004		
요구사항 명칭	커뮤니케이션 계획		
요구사항 분류	프로젝트 관리	응락수준	필수
요구사항 상세설명	<ul style="list-style-type: none"> - 주간 및 월간 미팅을 통해 프로젝트 진행 상황을 공유하고, 모든 팀원이 프로젝트 목표와 진행 상황을 명확히 인지하도록 함 		

[별첨2]

중간보고서

1. 요구사항 정의서에 명시된 기능에 대하여 현재까지 분석, 설계, 구현(소스코드 작성) 및 테스트한 내용을 기술하시오.

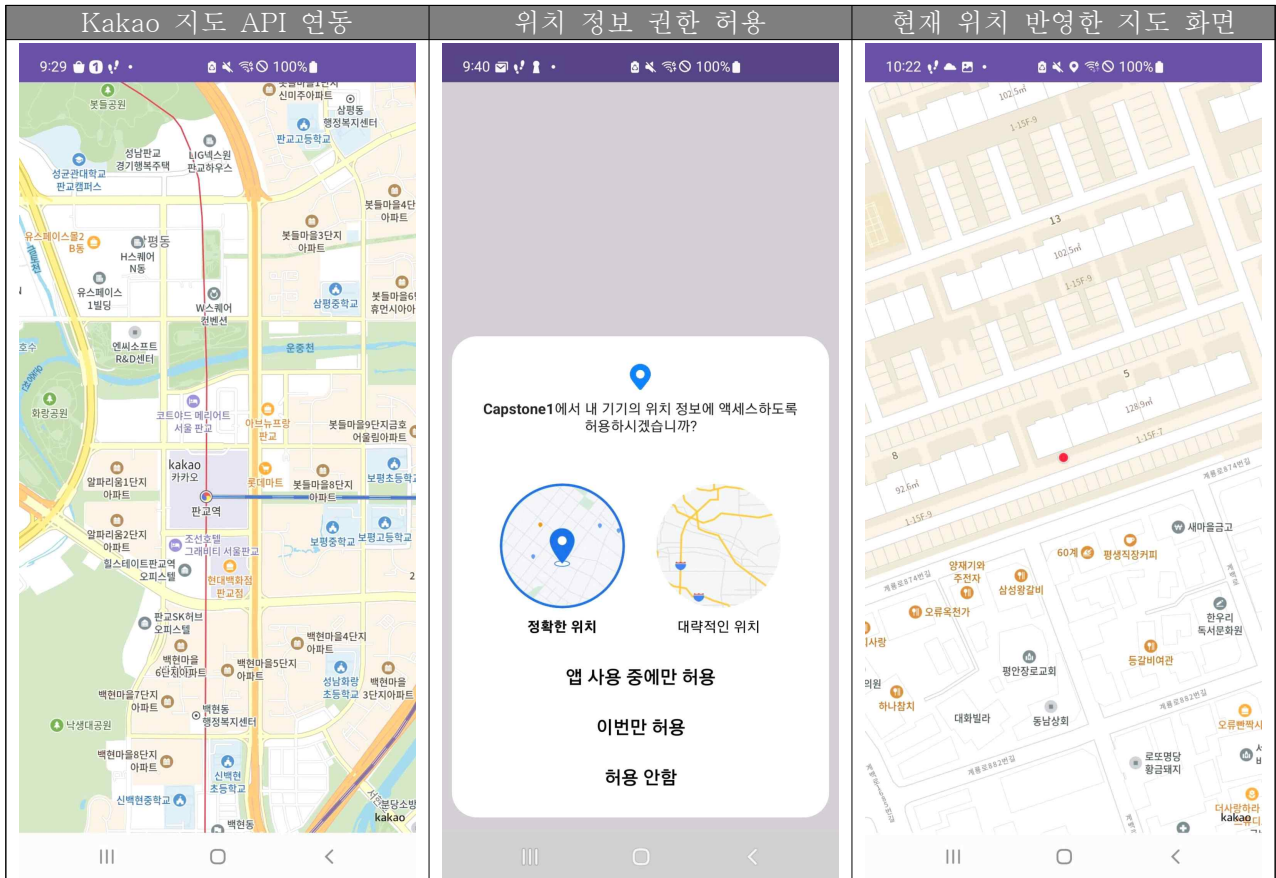
- 시스템(하드웨어, 시스템)구성도, 또는 소프트웨어 아키텍처
- 기능별 상세 요구사항(또는 유스케이스)



[유스케이스]

- 지도 표시: 이는 사용자가 앱을 통해 위치 정보를 시각적으로 파악할 수 있게 함
- 현재 위치 추적 및 업데이트: 사용자의 실시간 위치를 지도에 표시하고, 위치 변경이 있을 때마다 지도 상의 위치를 업데이트함
- 위치 서비스 권한 관리: 사용자의 위치 정보를 활용하기 위해 필요한 권한을 요청하고, 사용자가 이를 허용하거나 거부할 수 있도록 함

- 현재까지 구현한 소스코드 및 캡처
- UI 분석/설계 모델/프로토타입



- UI 분석: 현재 카카오맵 API v2를 활용하여 메인화면에 지도를 표시한 후 FusedLocationProviderClient를 활용하여 현재 위치를 받아 지도 중심에 표시함
- 프로토타입: 안드로이드 스튜디오를 이용하여 XML 레이아웃 편집기로 기본 레이아웃을 구성하고 JAVA를 통해 기능을 구현함. 레이아웃 인스펙터로 UI를 실시간으로 확인하여 각 구성요소를 디버깅하고 조정함. 또한 에뮬레이터를 통하여 테스트를 하고, 에뮬레이터로 실행이 안될 시 휴대하고 있는 안드로이드 기기로 테스트함

User Entity 클래스

```
@Entity
@Getter
@Setter
@NoArgsConstructor(access = AccessLevel.PROTECTED)
@Table(name = "users")
public class User {
    @Id
    @GeneratedValue(strategy = GenerationType.IDENTITY)
    private Long id;
    @Column(unique = true, nullable = false)
    private String username;
    @Column(nullable = false)
    private String password;
    @Column(nullable = false)
    private String name;
    @Column(unique = true, nullable = false)
    private String email;
    @OneToMany(mappedBy = "user", cascade = CascadeType.ALL, orphanRemoval = true)
    private Set<Favorite> favorites = new HashSet<>();
    public User(String username, String password, String name, String email) {
        this.username = username;
        this.password = password;
        this.name = name;
        this.email = email;
    }
}
```

AuthController

```
@RestController
@RequestMapping("/api/auth")
public class AuthController {
    @Autowired
    AuthenticationManager authenticationManager;
    @Autowired
    UserRepository userRepository;
    @Autowired
    PasswordEncoder encoder;
    @Autowired
    JwtUtils jwtUtils;
    @PostMapping(value = "/signin", produces = MediaType.APPLICATION_JSON_VALUE)
    public ResponseEntity<?> authenticateUser(@RequestBody LoginRequest loginRequest) {
        try {
            Authentication authentication = authenticationManager.authenticate(
                new UsernamePasswordAuthenticationToken(loginRequest.getUsername(),
                    loginRequest.getPassword()));
            SecurityContextHolder.getContext().setAuthentication(authentication);
            String jwt = jwtUtils.generateJwtToken(authentication);
            UserDetailsImpl userDetails = (UserDetailsImpl) authentication.getPrincipal();
            return ResponseEntity.ok(new JwtResponse(jwt,
                userDetails.getId(),
                userDetails.getUsername(),
                userDetails.getEmail()));
        } catch (AuthenticationException e) {
            // 로그인 실패 시 예외 처리
            return ResponseEntity
                .status(HttpStatus.UNAUTHORIZED)
        }
    }
}
```

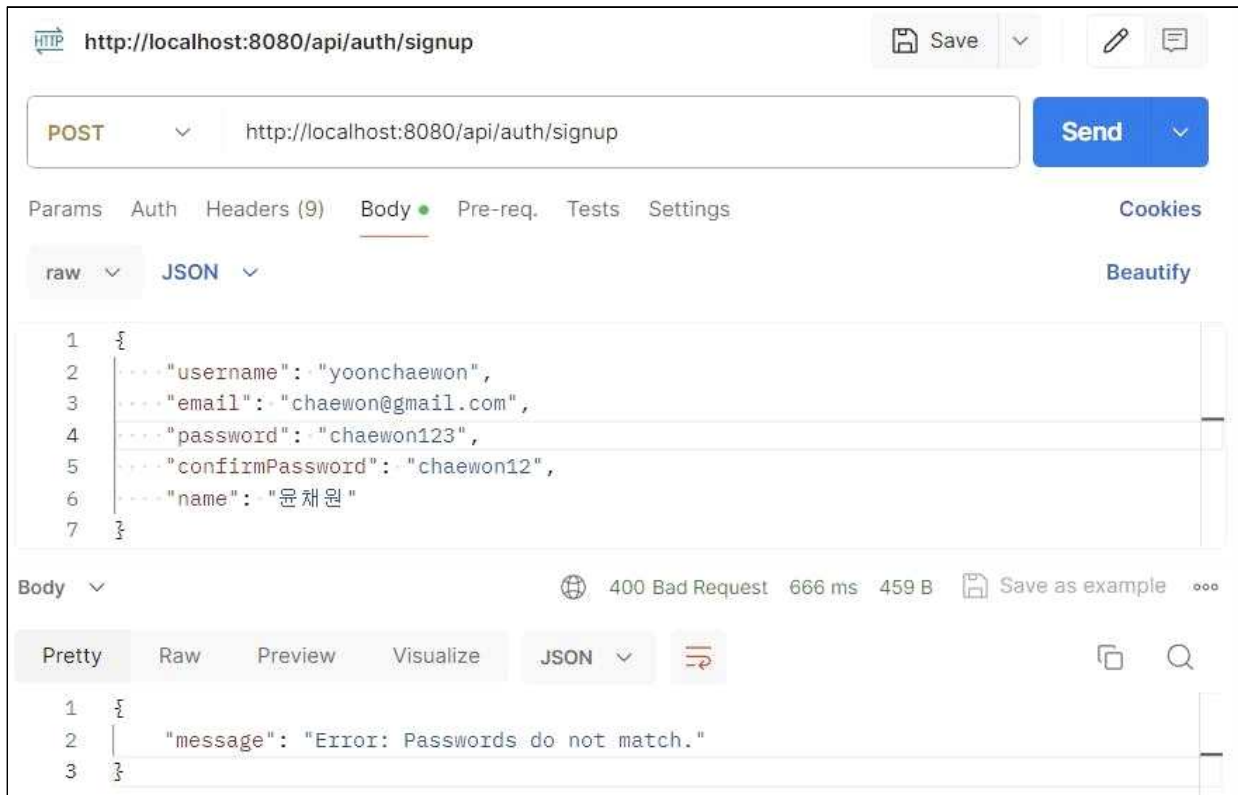
```

        .body(new MessageResponse("Error: Invalid username or password."));
    }
}
@PostMapping(value = "/signup", produces = MediaType.APPLICATION_JSON_VALUE)
public ResponseEntity<?> registerUser(@RequestBody SignupRequest signUpRequest) {
    if (userRepository.existsByUsername(signUpRequest.getUsername())) {
        return ResponseEntity
            .badRequest()
            .body(new MessageResponse("Error: Username is already taken!"));
    }
    if (userRepository.existsByEmail(signUpRequest.getEmail())) {
        return ResponseEntity
            .badRequest()
            .body(new MessageResponse("Error: Email is already in use!"));
    }
    if (!signUpRequest.getPassword().equals(signUpRequest.getConfirmPassword())) {
        return ResponseEntity
            .badRequest()
            .body(new MessageResponse("Error: Passwords do not match."));
    }
    // Create new user's account
    User user = new User(signUpRequest.getUsername(),
        encoder.encode(signUpRequest.getPassword()),
        signUpRequest.getName(),
        signUpRequest.getEmail());
    userRepository.save(user);
    return ResponseEntity.ok(new MessageResponse("User registered successfully!"));
}
}

```


실행결과

[회원가입 - 비밀번호 불일치]



HTTP <http://localhost:8080/api/auth/signup> Save

POST <http://localhost:8080/api/auth/signup> Send

Params Auth Headers (9) Body Pre-req. Tests Settings Cookies

raw JSON Beautify

```
1 {
2   ... "username": "yoonchaewon",
3   ... "email": "chaewon@gmail.com",
4   ... "password": "chaewon123",
5   ... "confirmPassword": "chaewon12",
6   ... "name": "윤채원"
7 }
```


Body 400 Bad Request 666 ms 459 B Save as example

Pretty Raw Preview Visualize JSON

```
1 {
2   "message": "Error: Passwords do not match."
3 }
```

- 회원가입에서 비밀번호가 불일치시 "Error: Passwords do not match" 메시지를 띄움.

[회원가입 - 성공]



HTTP <http://localhost:8080/api/auth/signup> Save

POST <http://localhost:8080/api/auth/signup> Send

Params Auth Headers (9) Body Pre-req. Tests Settings Cookies

raw JSON Beautify

```
1 {
2   ... "username": "yoonchaewon",
3   ... "email": "chaewon@gmail.com",
4   ... "password": "chaewon123",
5   ... "confirmPassword": "chaewon123",
6   ... "name": "윤채원"
7 }
```

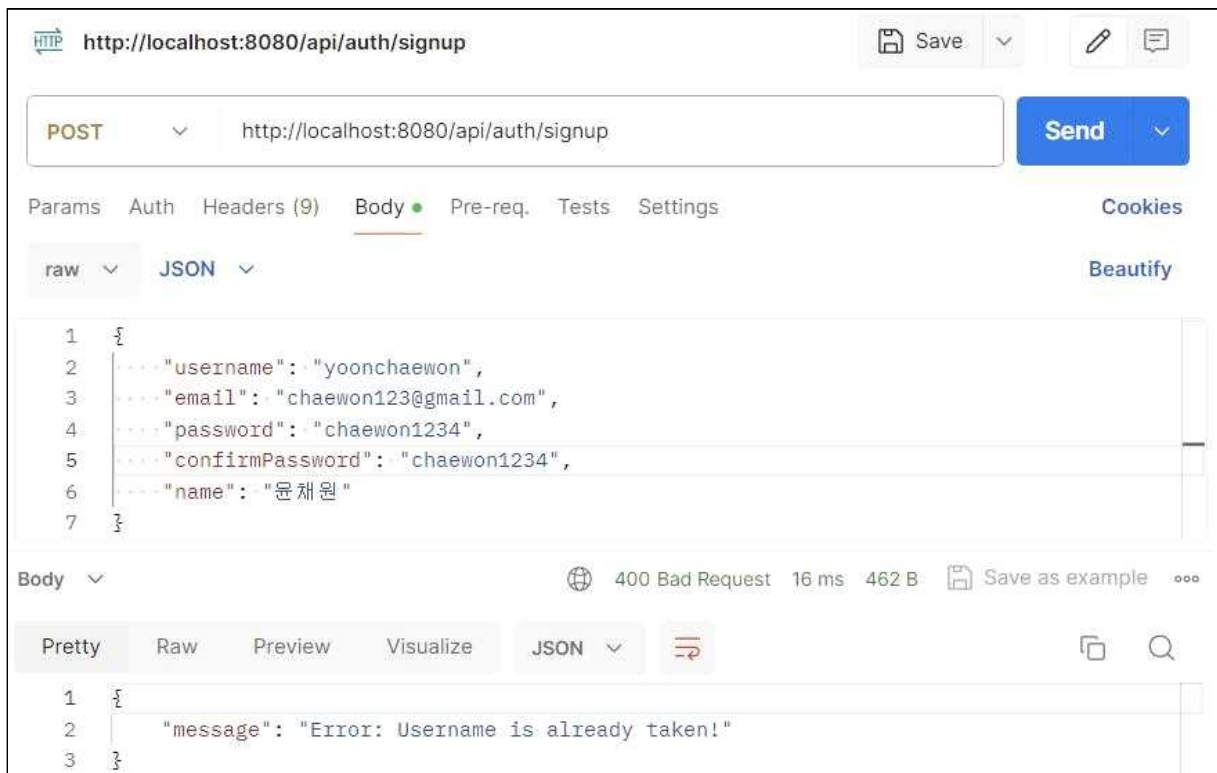
Body 200 OK 284 ms 478 B Save as example

Pretty Raw Preview Visualize JSON

```
1 {
2   "message": "User registered successfully!"
3 }
```

- 회원가입 성공 시 "User registered successfully!" 메시지를 띄움.

[회원가입 - 아이디 중복]



HTTP <http://localhost:8080/api/auth/signup> Save

POST <http://localhost:8080/api/auth/signup> Send

Params Auth Headers (9) Body Pre-req. Tests Settings Cookies

raw JSON Beautify

```
1 {
2   ... "username": "yoonchaewon",
3   ... "email": "chaewon123@gmail.com",
4   ... "password": "chaewon1234",
5   ... "confirmPassword": "chaewon1234",
6   ... "name": "윤채원"
7 }
```

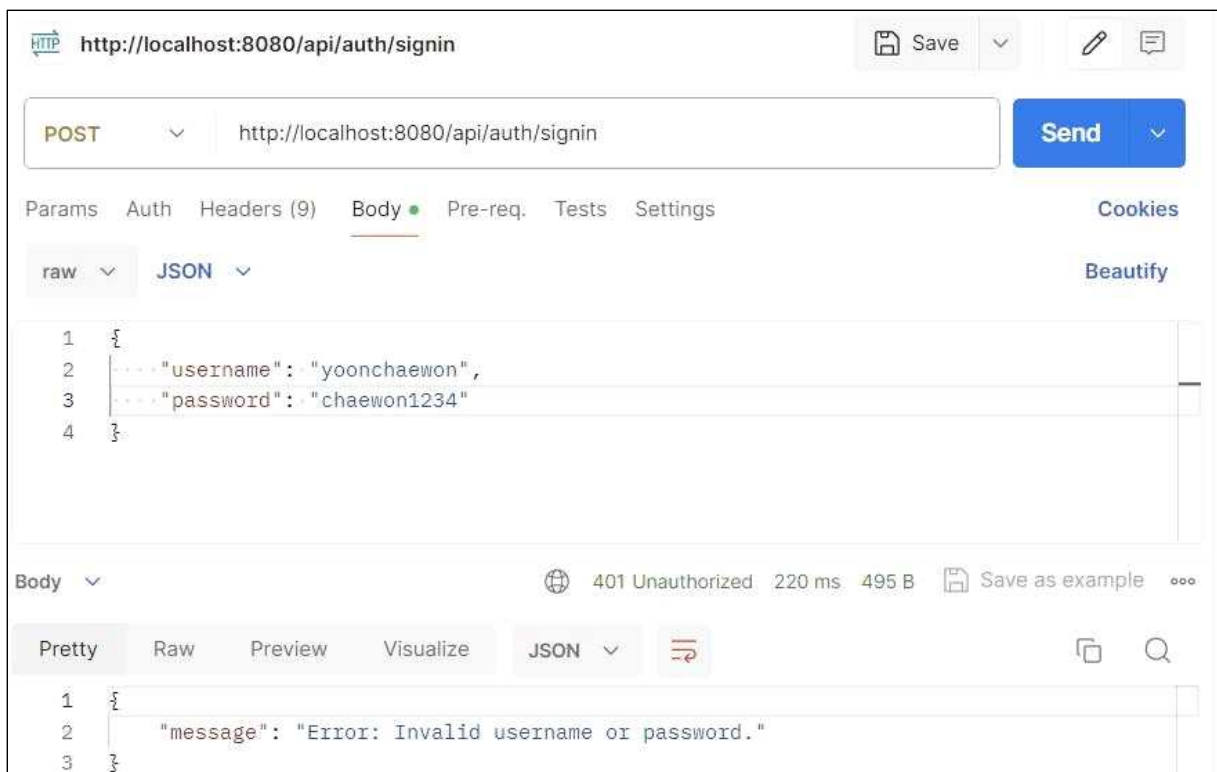
Body 400 Bad Request 16 ms 462 B Save as example

Pretty Raw Preview Visualize JSON

```
1 {
2   "message": "Error: Username is already taken!"
3 }
```

- 회원가입에서 이미 존재하는 아이디면 “Error: Username is already taken!” 메시지를 띄움.

[로그인 - 아이디 또는 비밀번호 불일치]



HTTP <http://localhost:8080/api/auth/signin> Save

POST <http://localhost:8080/api/auth/signin> Send

Params Auth Headers (9) Body Pre-req. Tests Settings Cookies

raw JSON Beautify

```
1 {
2   ... "username": "yoonchaewon",
3   ... "password": "chaewon1234"
4 }
```

Body 401 Unauthorized 220 ms 495 B Save as example

Pretty Raw Preview Visualize JSON

```
1 {
2   "message": "Error: Invalid username or password."
3 }
```

- 로그인에서 아이디 또는 비밀번호가 DB에 저장된 것과 일치하지 않으면 “Error: Invalid username or password.” 메시지를 띄움.

[로그인 - 성공]

HTTP <http://localhost:8080/api/auth/signin> Save Send

POST <http://localhost:8080/api/auth/signin> Send

Params Auth Headers (9) **Body** Pre-req. Tests Settings Cookies

raw **JSON** Beautify

```

1 {
2   "username": "yoonchaewon",
3   "password": "chaewon123"
4 }

```

Body 200 OK 214 ms 705 B Save as example

Pretty Raw Preview Visualize **JSON**

```

1 {
2   "token": "eyJhbGciOiJIUzUxMiJ9.eyJzdWIiOiJ5b29uY2hhZXdvbiIsIm1hdCI6MTcxNDQ5NDAwMywiZXhwIjoxNzE0NTgwNDZfQy5CTniaz2P9yFySDxoz1j9DqXFNLz_YeX3cgDDoiuu00ZHpV0oFbapl_zfKKJd_SRqYvgHA9ZM1YF3BxnVHTF4Q",
3   "type": "Bearer",
4   "id": 1,
5   "username": "yoonchaewon",
6   "email": "chaewon@gmail.com"
7 }

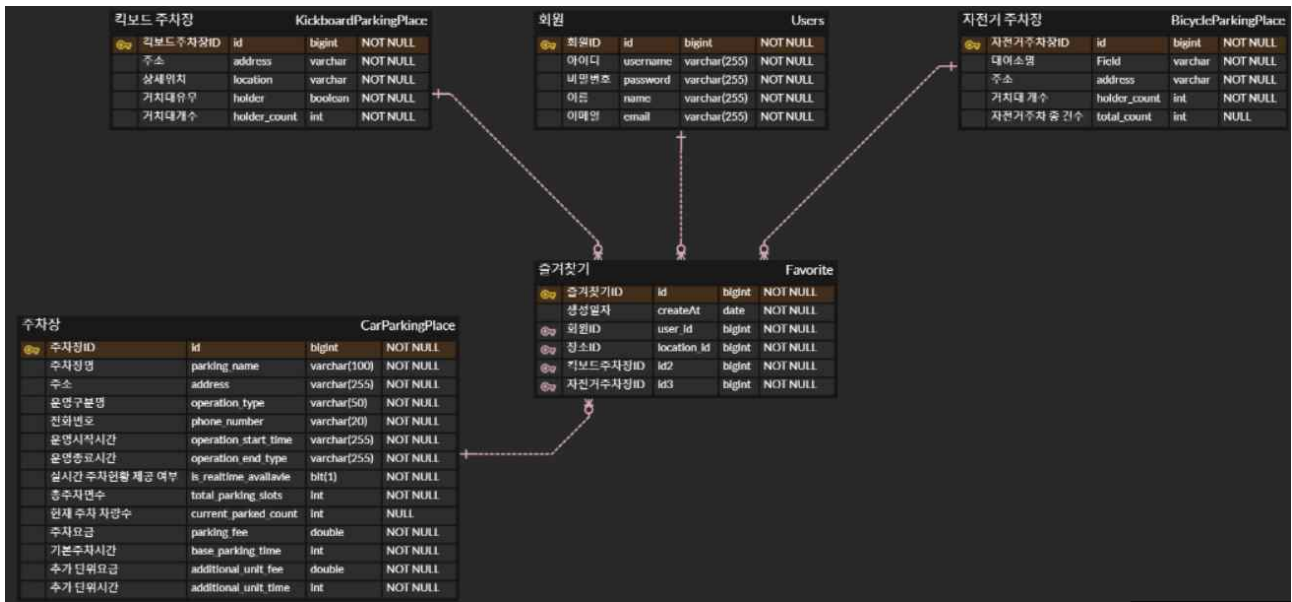
```

[DB 저장 모습]

	id	username	password	name	email
▶	1	yoonchaewon	\$2a\$10\$hbJEKpp1l0zwJqt5PF8KdOvJi0n2fJqjd...	윤채원	chaewon@gmail.com
•	NULL	NULL	NULL	NULL	NULL

- 회원정보가 DB에 저장되면 password는 암호화 돼서 저장됨.

- E-R 다이어그램/DB 설계 모델(테이블 구조)



- 테스트 계획서, 테스트 케이스 기술서 등등.

- 프로젝트 최종 구현을 위하여 크게 프론트엔드와 백엔드로 역할을 분담하여, 외부 API를 이용한 지도 구현 및 주차장, 자전거, 전동킥보드 등에 대한 정보를 추출하여 정보를 제공하고 해당 위치까지의 경로를 안내받기를 하는 경우 내비게이션 기능을 통하여 안내하는 것이 목표이다.

2. 프로젝트 수행을 위해 적용된 추진전략, 수행 방법의 결과를 작성하고, 만일 적용과정에서 문제점이 도출되었다면 그 문제를 분석하고 해결방안을 기술하시오.

- 문제해결을 위해 적용한 방법(또는 기법) 결과, (문제점, 해결방안)

[문제점]

- 카카오 지도 API 버전이 바뀌므로 인하여 SDK에 관한 여러 가지 오류와 API 사용 방법에 대해 어려움이 있었음.

[해결방안]

- 카카오 지도 API v2에 대한 가이드를 참고하여 환경설정 함

[문제점]

- 카카오 Developer 가이드를 참고하여 Map을 띄우려고 하였지만 화면이 켜지지 않고 앱이 실행되지 않음

[해결방안]

- `KakaoMapSdk.init(thins, "앱키");` 코드를 onCreate 안에 작성하였더니 화면에 카카오지도가 띄어짐

[문제점]

- 지도를 띄우기까지 하였으나 중심화면이 현재 위치가 아닌 다른 위치로 뜸

[해결방안]

- Google Play 서비스의 Location API 중 하나인 `FusedLocationProviderClient`(통합위치 정보제공자)를 사용하여 기기의 위치 정보를 얻음

- 팀원의 책임 및 역할 수행에 대한 결과, (문제점, 해결방안)

[문제점]

- 팀원 간의 개인 역량이 다름

[해결방안]

- 관련 서적 및 웹서핑을 참고하여 공부

[문제점]

- 팀원 전원이 API를 사용하여 프로젝트를 진행해 본 경험이 없음

[해결방안]

- 주기적으로 진행 상황을 서로 소통하며 목표를 유연하게 수정

- 같은 개발 영역을 맡은 팀원끼리 필요한 부분을 서로 피드백

- 프로젝트 일정계획에 맞추지 못한 경우의 문제점, 해결 방안

[문제점]

- 계획하였던 일정보다 개발 속도가 늦다는 문제점

[해결방안]

- 팀원끼리 자주 소통하여 개발 현황 확인

- 개발 현황에 따라 일정을 수정하여 꾸준히 개발하고 공유

- 요구사항 변경관리의 결과, (문제점, 해결 방안) 등등.

[문제점]

- 프로젝트 경험이 부족하여 수정하고 해결해야 하는 방법을 잘 알지 못함

[해결방안]

- 웹서핑을 통해 관련 서적 및 외부 자료를 활용하여 수정하고 적용

캡스톤 디자인 | 중간보고서 채점표

평가도구	평 가 항 목	평 가 점 수				
		1	2	3	4	5
중간 보고서 및 실행 결과	1. 요구사항 정의서(기능, 성능, 인터페이스 등)가 구체적으로 작성되었는가?					
	2. 요구분석, 설계 산출물(모델, 프로토타입 등)의 내용이 충실한가?					
	3. 설계 및 구현 문제를 위해 적용한 이론, 문제해결 방법이 제시되었으며 그 적용이 적합한가?					
	4. 구현된 소프트웨어(또는 이와 동등한 하드웨어 시스템)가 버그없이 실행되었는가?					
	5. 구현된 소프트웨어(또는 이와 동등한 하드웨어 시스템)의 성능 요구사항은 충족되었는가?					
도구활용	6. 설계 및 구현을 위해 도구가 적절히 활용되었는가?					
	7. 도구의 활용수준(능숙도)은 프로젝트 수행에 적합한가?					
팀원의 업무 및 역할	8. 팀원의 업무분담에 따른 역할 및 협력이 충실히 이루어졌는가? (평가자에 의한 질의)					
	9. 프로젝트 중간 진척상황에 대해 팀원이 충분히 인지하고 있는가?(평가자에 의한 질의)					
합계						
*검토 의견(최종완료 때까지 보완해야할 점에 대해 작성 요망)						
심사위원(소속):		(이름)			(인)	