

프로젝트1 - 2팀 산출물

팀장: 박청조

팀원: 고민정 이홍비 허선호

회의록

노선 회의록 요약

1차 - 2024.11.26

팀장 선출 및 팀 Github Repository 생성

2차 - 2024.12.9

프로젝트 주제 선정 및 기능 예상

- 주차 관리 시스템
 - 입/출차 관리
 - 차량 등록 및 해제, 블랙리스트 등록 및 해제
 - (기간별 또는 전체) 주차 기록, 차량 조회
 - 일정 관리
- 그 외
 - 일정 관리 서비스, 검색 서비스, 영화 관련 커뮤니티 서비스, 설문조사 서비스, LMS 학습 관리 시스템

회의록

노선 회의록 요약

3차 - 2024.12.10

프로젝트 흐름 계획

- 주제 선정 → 테이블 설계 및 명세서 작성 → DB/더미 데이터 SQL Query → ERD 생성 → 레퍼런스 조회 → API 명세서 작성, 화면 구상 → 개발 → 테스트 → 인프라 구축 → 도커를 통한 자동화 스크립트 작성 → 프로젝트 정리 및 회고
- 인프라 구축이 먼저 되는 게 좋겠다는 강사님의 피드백 ⇒ 개발과 인프라 구축의 진행 순서 변경

테이블 설계

- 블랙리스트 등록 및 해제 기능 구현을 위한 별도의 테이블 생성에 대한 논의

- 핵심 기능 정리
- 테이블 명세서, ERD 작성
- 기능 별 페이지 구분
- 개발 환경 정의

회의록

노선 회의록 요약

4차 - 2024.12.11

- 화면 구상 마무리
- 웹사이트 디자인 및 색상 , 폰트결정
- 총 수입에 대한 의견 논의
- 인프라 구축 완료
- API 명세 관련 논의

5차 - 2024.12.12

API 명세서 작성 완료

개발 진행

- 담당 Entity 구현, DTO 구현
- 담당 기능 결정 후 본격적인 개발

개발 중 논의 사항

- 주차 기록 조회 시 기간 별 조회에 대한 것

회의록

노선 회의록 요약

6차 - 2024.12.13

개발 완료

- 개발 진행 상황 공유
- 각자 담당 기능 개발
 - 박청조 : 로그인, 관리자 주차 현황
 - 고민정 : 입/출차
 - 이홍비 : 기간별 | 전체 주차 기록 및 차량 조회
 - 허선호 : 사용자 주차 현황, 사용자 주차 조회, 인트로

결정 및 수정사항

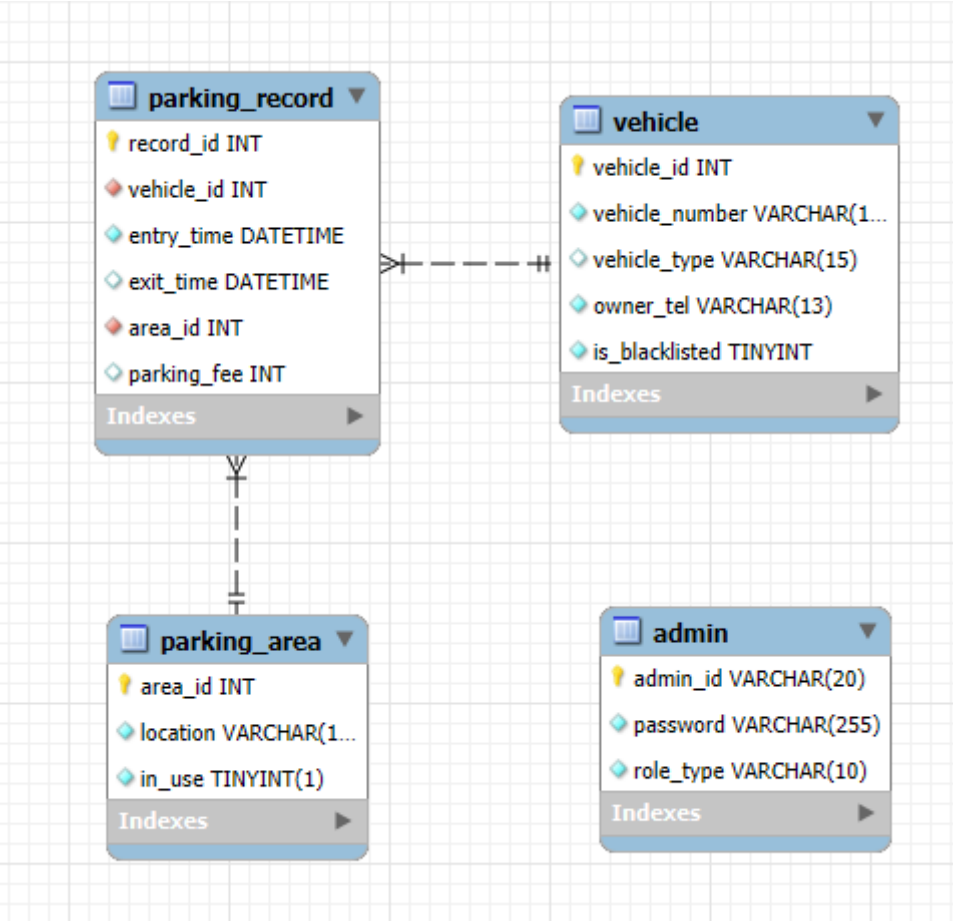
테이블 관련

- 블랙리스트 테이블은 차량 테이블의 컬럼으로 축소
- parking_area 테이블에 vehicle 참조 여부 논의 ⇒ in_use 컬럼만 남겨두기로 결정
- 테이블 네이밍 컨벤션 (소문자 + 언더바(_))
- user의 phone number 필드 Unique 속성 추가, 주차비 필드 추가, 차량 테이블은 user 참조

기능 관련

- 페이지 수 증가 우려로 관리자 페이지에 집중하여 user는 admin 하나로 결정, 차주 연락처는 차량 테이블의 컬럼으로 추가
- 주차 기록 조회 페이지에서 '특정 기간의 수입' 출력 + 주차 현황 페이지에서 '전체 기간'의 수입 출력
- 기존의 '총 수입 페이지'는 삭제
- /search 페이지에서 검색 시 엔드포인트로 데이터 요청하는 대신 /search/{차량번호}로 차량 조회

ERD



테이블 명세서

주제영역명		주차 관리	작성일	2024.12.10	작성자		고민정	
테이블ID		parking_area	테이블명		주차 공간			
테이블설명		주차 공간 관리 테이블						
No.	컬럼ID	컬럼명	타입	길이	NULL	KEY	DEFAULT	비고
1	area_id	주차 구역 ID	INTEGER		N	PK		AUTO INCREMENT
2	location	주차 구역 위치	VARCHAR	10	N		Z-99	
3	in_use	주차 구역 사용 여부	BOOLEAN		N		FALSE	
인덱스		인덱스 키						
area_id		주차 구역 ID						
업무규칙	주차 구역 위치: A~Z + 01~99 (ex. A-01)							

주제영역명		주차 관리	작성일	2024.12.10		작성자		이홍비
테이블ID		vehicle	테이블명		차량 정보			
테이블설명		차량 정보 테이블						
No.	컬럼ID	컬럼명	타입	길이	NULL	KEY	DEFAULT	비고
1	vehicle_id	차량 고유 ID	INTEGER		N	PK		AUTO INCREMENT
2	vehicle_number	차량 번호	VARCHAR	10	N			
3	vehicle_type	차종	VARCHAR	15				
4	owner_tel	차주 전화번호	VARCHAR	13	N			
5	is_blacklisted	블랙 리스트 여부	TINYINT		N		0	0 : 블랙 리스트 x 1 : 불량 리스트 o
인덱스		인덱스 키						
vehicle_id		차량 고유 ID						
업무규칙	블랙 리스트 기준 1. 규칙 위반 2. 요금 미납							

테이블 명세서

주제영역명		주차 관리	작성일	2024.12.10		작성자		허선호
테이블ID		parking_record	테이블명		주차 기록			
테이블설명		입출차 기록 테이블						
No.	컬럼ID	컬럼명	타입	길이	NULL	KEY	DEFAULT	비고
1	record_id	기록 ID	INTEGER		N	PK		AUTO INCREMENT
2	vehicle_id	차량 번호	INTEGER		N	FK		
3	entry_time	입차 시각	DATETIME		N			
4	exit_time	출차 시각	DATETIME					
5	area_id	주차 공간	INTEGER		N	FK		
6	parking_fee	정산된 주차비	INTEGER					
인덱스		인덱스 키						
exit_time		출차 시각						
record_id		기록 ID						
업무규칙	입차 시 차량 번호 확인 후 남은 주차 공간이 있는지 확인. 출차 시 주차비 자동 정산							

주제영역명		주차 관리		작성일	2024.12.10		작성자		박청조	
테이블ID		admin		테이블명		관리자				
테이블설명		주차 서비스 관리자 테이블								
No.	컬럼ID	컬럼명	타입	길이	NULL	KEY	DEFAULT	비고		
1	admin_id	관리자 아이디	VARCHAR2		N	PK				
2	password	비밀번호	VARCHAR2		N					
3	role_type	권한	VARCHAR2		N		ROLE_ADMIN			
인덱스		인덱스 키								
admin_id		관리자 ID								
업무규칙										

DDL, DML SQL scripts.sql

```
# 사용 데이터베이스 지정
use parking;
```

```
# 기존 테이블 데이터 삭제
drop table if exists parking_record;
drop table if exists vehicle;
drop table if exists parking_area;
drop table if exists admin;
```

```
# 관리자 테이블 생성
create table if not exists admin (
    admin_id varchar(20) not null,
    password varchar(255) not null,
    role_type varchar(20) not null,
    primary key (admin_id)
);
```

```
# 차량 테이블 생성
create table if not exists vehicle (
    vehicle_id integer not null auto_increment, # 차량 고유 id
    vehicle_number varchar(10) not null, # 차량 번호
    vehicle_type varchar(15), # 차종
    owner_tel varchar(13) not null, # 차주 전화번호 : 010-1234-5678
    is_blacklisted tinyint not null default 0, # 블랙리스트 여부

    primary key (vehicle_id), # 기본키
    unique (vehicle_number) # 차 번호 - 중복 불가 처리
);
```

```
# 주차 공간 테이블 생성
create table if not exists parking_area (
    area_id int not null auto_increment,
    location varchar(10) not null default 'Z-99',
    in_use boolean not null default false,
    primary key (area_id)
);
```

```
# 주차 기록 테이블 생성
create table if not exists parking_record (
    record_id int not null auto_increment, # 기록 id (프라이머리 키)
    vehicle_id int not null, # 차량 번호(외래키)
    entry_time datetime not null, # 입차 시각
    exit_time datetime, # 출차 시각 (null 가능)
    area_id int not null, # 주차한 공간 (외래키)
    parking_fee int, # 주차 요금 (null 가능)
    primary key (record_id), # 기본키
    foreign key (vehicle_id) references vehicle(vehicle_id), # 외래키 차량 테이블 참조
    foreign key (area_id) references parking_area(area_id) # 외래키 주차 공간 테이블 참조
);
```

```
# 더미 데이터 생성
```

```
# 관리자 생성
insert into admin (admin_id, password, role_type) values ('admin', 'admin1234', 'ADMIN');
```

```
# 차량 생성
insert into vehicle (vehicle_number, vehicle_type, owner_tel, is_blacklisted) values
('23가2017', 'SEDAN', '010-2007-0805', 0), ('53가2937', 'HATCHBACK', '010-2024-1210', 0),
('132라9348', 'CONVERTIBLE', '010-1991-0214', 0), ('54바7226', 'SUV', '010-1950-0625', 0),
('23하7196', 'COUPE', '010-1919-0301', 0), ('92나1653', 'WAGON', '010-1945-0815', 0),
('123다1212', 'RV', '010-2014-0801', 0), ('214라0214', 'VAN', '010-1443-0102', 0);
```

```
# 주차 공간
insert into parking_area (area_id, location, in_use)
values
(1, 'A-01', false), (2, 'A-02', false), (3, 'A-03', false), (4, 'A-04', false), (5, 'A-05', false), (6, 'A-06', false),
(7, 'A-07', false), (8, 'A-08', false), (9, 'A-09', false), (10, 'A-10', false), (11, 'B-01', false), (12, 'B-02', false),
(13, 'B-03', false), (14, 'B-04', false), (15, 'B-05', false), (16, 'B-06', false), (17, 'B-07', false), (18, 'B-08', false),
(19, 'B-09', false), (20, 'B-10', false), (21, 'C-01', false), (22, 'C-02', false), (23, 'C-03', false), (24, 'C-04', false),
(25, 'C-05', false), (26, 'C-06', false), (27, 'C-07', false), (28, 'C-08', false), (29, 'C-09', false), (30, 'C-10', false),
(31, 'D-01', false), (32, 'D-02', false), (33, 'D-03', false), (34, 'D-04', false), (35, 'D-05', false), (36, 'D-06', false),
(37, 'D-07', false), (38, 'D-08', false), (39, 'D-09', false), (40, 'D-10', false);
```

```
# 주차 기록
insert into parking_record (vehicle_id, entry_time, exit_time, area_id, parking_fee) values
(1, '2024-12-05 07:22:00', '2024-12-05 18:33:00', 4, 33500),
(2, '2024-12-07 14:20:00', '2024-12-08 08:13:00', 2, 53500),
(3, '2024-12-8 08:04:21', '2024-12-08 08:27:32', 11, 500), (4, '2024-12-10 21:27:33', '2024-12-10 23:21:35', 7, 5500),
(5, '2024-12-11 11:30:00', '2024-12-11 12:30:00', 29, 3000),
(6, '2024-12-11 11:30:00', '2024-12-11 14:30:00', 20, 9000),
(7, '2024-12-12 09:00:00', '2024-12-12 18:50:00', 9, 29500),
(8, '2024-12-12 20:32:15', '2024-12-13 06:55:09', 15, 31000);
commit;
```

인트로 영역

첫 화면

주차장 상태 확인

- 사용 중 : 이미 주차된 자리
- 사용 가능 : 주차할 수 있는 자리

차량 번호를 입력해 주십시오.

조회

A 구역

B 구역

C 구역

D 구역

A-01 사용 중	A-02 사용 가능	A-03 사용 가능	A-04 사용 가능	A-05 사용 중
A-01 사용 중	A-01 사용 중	A-01 사용 중	A-01 사용 가능	A-10 사용 중
A-01 사용 중	A-01 사용 중	A-01 사용 가능	A-01 사용 중	A-15 사용 중
A-01 사용 중	A-01 사용 가능	A-01 사용 가능	A-01 사용 가능	A-20 사용 중

주차 공간 조회 : /

24바1234

조회

A-01
사용 가능

A-02
사용 가능

A-03
사용 가능

A-04
사용 가능

A-05
사용 가능

A-06
사용 가능

A-07
사용 가능

A-08
사용 가능

A-09
사용 가능

A-10
사용 가능

차량 번호 : 24바1234

차주 번호 : 010-1234-1234

주차 구역 : A-01

입차 시각 : 2024-12-11-11:50:29

예상 주차비 : 3000원

돌아가기

주차 차량 조회 페이지 :

관리자 영역

로그인 시 토큰 받아서
주차현황 페이지로 이동

주차 관리 서비스

관리자 로그인

ID :

PW :

로그인

관리자 로그인

주차 현황

주차 기록

차량 조회

로그아웃

주차 현황(관리자)

현재 수입: 25,000원

여유 공간: 30면/40면

A구역

B구역

C구역

D구역

A-01

20가2037

010-2736-3927

25시간 10분 5000원

A-02

51가2733

010-2736-3927

25시간 10분 5000원

A-03

88가2512

010-2736-3927

25시간 10분 5000원

A-04

미사용

A-05

미사용

A-06

미사용

A-07

미사용

A-08

미사용

A-09

52가5875

010-2736-3927

25시간 10분 5000원

A-10

64가7255

010-2736-3927

25시간 10분 5000원

입차: 미사용 공간을 클릭

출차: 사용 중인 공간을 클릭

주차 현황

- 1. 각 구역 선택 시 입출차
관리 페이지로 이동
- 2. 주차 기록 / 차량 조회
클릭 시 해당하는 각
페이지로 이동

화면 구성

입차 시 필요한 정보
1. 차량 번호
2. 차주 폰번호
3. 차종

주차 현황 주차 기록 차량 조회 로그아웃

입출차 관리

A-01 입차

차종 :

전화번호 :

차량 번호 :

입차

입차 (등록)

주차 현황 주차 기록 차량 조회 로그아웃

입출차 관리

A-01 출차

차종 :

전화번호 :

차량 번호 :

출차

출차 (정산)

입출차 관리에서 출차 시
해당되는 API 호출 후 주차 현황
페이지 리다이렉트

화면 구성

짙은 파랑 : 입차 (주차 중)
엷은 파랑 : 입차 → 출차 완료

주차 현황

주차 기록

차량 조회

로그아웃

주차 기록 조회

Vehicle Number

24/12/09 ~ 24/12/13

총수입 ₩ 23,000

▼ Filter

차량 번호	입차 시각	출차 시각	주차 요금
23가 2017	2024.12.05 07:22:00	2024.12.05 18:33:00	33500
53가 2937	2024.12.05 07:22:00		
132라 9348	2024.12.05 07:22:00	2024.12.05 18:33:00	
54바 7226	2024.12.05 07:22:00	2024.12.05 18:33:00	
23하 7196	2024.12.05 07:22:00	2024.12.05 18:33:00	
92나 1653	2024.12.05 07:22:00		
123다 1212	2024.12.05 07:22:00		
214라 0214	2024.12.05 07:22:00	2024.12.05 18:33:00	
...	

주차 기록

주차 현황

주차 기록

차량 조회

로그아웃

등록 차량 조회(관리자)

24/12/09 ~ 24/12/13

▼ Filter

총 20대
블랙리스트 2대

차량 번호	전화 번호	차종	블랙리스트
23가 2017	010-2736-3017	SONATA	등록
23가 2017	010-2736-3017	GRANDUER	해제
23가 2017	010-2736-3017	SORENTO	등록
23가 2017	010-2736-3017	PRIDE	등록
23가 2017	010-2736-3017	PRIDE	등록
23가 2017	010-2736-3017	PRIDE	해제
23가 2017	010-2736-3017	PRIDE	등록
23가 2017	010-2736-3017	PRIDE	등록
23가 2017	010-2736-3017	PRIDE	등록

차량 조회 + 블랙 리스트 등록

주차 관리 API 명세서

기능	HTTP 메서드	EndPoiont	request	response	토큰 필요	설명
주차 공간 조회 (사용자)	GET	?area={구역이름}		{ "zone01": "used", "zone02": "free", "zone03": "free", "zone04": "free", "zone05": "used", "zone06": "used", "zone07": "free", "zone08": "free", "zone09": "free", "zone10": "free" }	NO	area 기본값은 'A'
주차 차량 조회(사용자)	GET	/search/{차량번호}	없음	{ "vehicleNumber": "24바1234", "ownerTel": "010-1234-1234", "location": "A-01", "entryTime": "2024-12-11 11:50:29", "parkingFee": "3000" }	NO	차량을 찾지 못하면 메시지 출력 { "msg": "차량을 찾지 못했습니다." }
관리자 로그인 페이지	GET	/admin/login	없음		NO	
관리자 입출차 페이지	GET	/admin/entry-exit/{구역이름}	없음		YES	
관리자 주차 현황	GET	/admin/parking?area={구역이름}	없음	{ "status": "used", "vehicleNumber": "20가2037", "ownerTel": "010-1234-1234", "entryTime": "2024-12-11 11:50:29", "parkingFee": "3000" }	YES	area 기본값은 'A' 비어 있으면 { "status" : "free" }

주차 관리 API 명세서

기능	HTTP 메서드	EndPoiont	request	response	토큰 필요	설명
관리자 주차 기록 페이지	GET	/admin/parking/history	없음		YES	
관리자 차량 관리 페이지	GET	/admin/vehicles	없음		YES	
관리자 로그인	POST	/admin/login	{ "username" : "admin", "password" : "password" }	로그인 성공 : http status code 200 { "message" : "success" } redirect "/admin/parking"	NO	로그인 실패 : http status code
입차	POST	/admin/entry	{ "vehicleNumber": "24바1234", "ownerTel": "010-1234-1234", "location": "A-01" }	입차 성공: http status code 200 { "entryStatus": "success" } redirect "/admin/parking"	YES	입차 실패: http status code 500 { "entryStatus": "이미 입차한 차량의 번호입니다." } { "entryStatus": "요청하신 공간에 다른 차량이 주차되어 있습니다." }
출차	POST	/admin/exit/{장소이름}	없음	출차 성공: http status code 200 { "exitStatus": "success" } redirect "/admin/parking"	YES	출차 실패: http status code 500 { "exitStatus": "요청하신 공간에 주차된 차량이 없습니다." }
차량 블랙리스트 등록/해제	PUT	/admin/vehicles/{차량번호}/black		redirect "/admin/vehicles"	YES	

주차 관리 서비스 (Parking Lot Management Service)

- 주차에 대한 전반적인 서비스 제공
- 사용자 = 고객 + 관리자
- 관리자 - 로그인, 주차 현황 조회, 주차 기록 조회, 입출차 처리, 차량 조회
- 고객 - 주차 공간 조회, 주차 중인 차량 조회

- 박청조 : 관리자 로그인, 관리자 주차 현황 조회 기능 구현
- 이홍비 : 관리자 주차 기록 조회 기능 구현
- 고민정 : 입차 / 출차 기능 구현
- 허선호 : 사용자 주차 공간 조회, 차량 조회 기능 구현

주차 공간 조회

A~D 4개의 구역으로 각 10개의 공간으로 이루어진 주차장에서 현재 사용 중인 공간과 그렇지 않은 공간을 사용자가 한눈에 파악할 수 있도록 한다.

주차 차량 조회

입차해서 차량을 등록하고 난 후 차주가 주차 중인 자신의 차량 번호를 조회하여 차량 번호, 차주 번호, 주차 위치, 입차 시각, 예상 주차비 등 상세 정보를 볼 수 있다.

관리자 로그인 페이지

관리자가 차량의 입출차 처리를 수행하고 입출차 시각 및 규칙 등을 고려하여 특정 차량을 블랙리스트로 설정한다.

이러한 권한을 가지는 유저(관리자)가 로그인함으로써 각 기능에 접근할 수 있도록 처리한다.

관리자 주차 현황

사용자가 주차 현황을 파악할 수 있는 페이지와 달리, 관리자의 주차 현황은 주차 요금으로 인한 전체 수입, 구역별 여유 공간 파악, 해당 위치에 주차한 차량 정보, 주차 시간, 현재 시각까지의 주차비를 한눈에 파악할 수 있다.

입차/출차

관리자가 주차 현황 페이지에서 미사용 구역을 클릭하면 차량과 차주 정보를 등록한다.
사용 중인 구역을 클릭하면 차량 정보, 주차 구역, 주차 요금을 확인한다. 이렇게 입출차 처리를 진행한다.

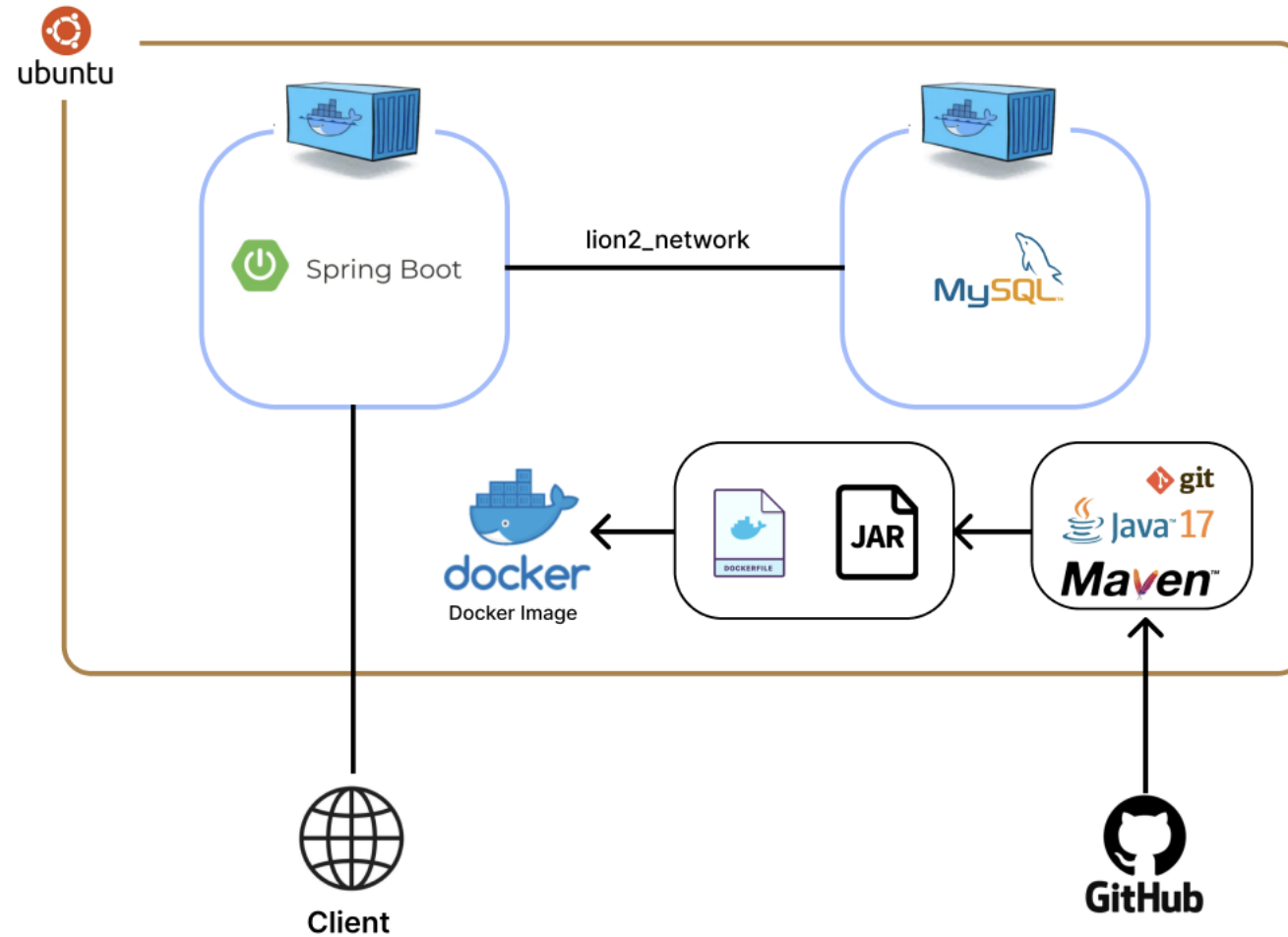
관리자 주차 기록

관리자는 모든 주차 기록(차량 번호, 입출차 시각, 주차 요금)을 확인할 수 있다.
차량 번호와 날짜를 설정하여 검색하면 해당 기간 동안 그 차량의 주차 기록만을 조회한다.
시작일, 마지막 일자를 설정하여 기간별로 입출차한 차량의 주차 기록과 그 기간 동안의 총수입을 확인한다.

관리자 차량 조회 및 블랙 리스트 관리 → 추후 개발 예정

관리자는 등록된 차량과 차주의 정보를 확인할 수 있다.
입출차 시각의 차이가 큰 차량은 관리자가 블랙리스트에 등록하여 특별 관리 하며 필요 시 해제한다.
이 주차장에 등록된 차량의 총 수를 확인할 수 있다.

인프라 구축 Flow



결과 및 성과

이번 프로젝트의 주요 목표는 Git을 활용한 협업 경험 축적과 개발 주제 선정, 테이블 설계, API 명세서 작성, 화면 구상, 인프라 구축까지 전 과정을 직접 경험하며 개발과 인프라 작업의 전반적인 흐름을 복습하는 것이었다고 생각합니다.

프로젝트를 진행하며 스스로 부족한 부분과 잘 학습된 부분을 판별할 수 있었고, 부족했던 영역을 학습을 통해 보완할 수 있었습니다.

특히 Git에 대해 겹핍기로 작업하던게 많았는데 이번 기회에 여러 커맨드를 다시 확인하며 사용했고 개인적으로도 큰 성과를 얻었다고 느낍니다.

후기 및 배운 점

프로젝트 과정

진행하면서 어려움이 있었지만, 그만큼 집중하여 효율적으로 작업을 진행할 수 있었습니다. 특히, 프로젝트 초기에 설계부터 인프라 구축까지 각 단계의 작업 계획을 세우고 이를 기반으로 진행한 점은 큰 도움이 되었습니다. 비록 완벽히 계획대로 진행되지는 않았지만, 팀원들과 지속적으로 질문하고 소통하며 일정을 잘 맞춰 프로젝트를 성공적으로 완료할 수 있었습니다.

팀장 경험

항상 팀장을 맡기 꺼려했지만, 이번에 팀장을 맡아 짧았지만 주도적으로 프로젝트를 이끌어가는 경험을 할 수 있었습니다. 팀원들이 적극적으로 따라주고 열심히 참여해준 덕분에, 좋은 협업 경험을 쌓을 수 있었고 이를 통해 팀원 간의 원활한 소통과 협업이 프로젝트의 성공에 얼마나 중요한지 다시 한번 깨달았습니다.

- 충분히 실습했다고 생각했는데도 인프라 구축은 어려웠다 덕분에 새로 깔고 지우며 다양한 경우의 수를 접하며 익숙해질 수 있었다 주차 관리 서비스를 위한 DB를 설계하면서 게시판과 비슷한 흐름을 가졌기에 정리해보는 느낌으로 진행할 수 있겠다 자신했었는데 연관 관계를 참조하면서 많아지는 DTO, Entity를 다루다보니 객체를 생성하는데도 신중해지고 영속성에 대해 참 많이 배우게 되었다
- 무엇보다 팀원들과 끊임없이 소통하며 문제를 하나하나씩 해결해 나가는 과정이 뿌듯했다 각자 맡는 기능의 관련된 부분을 나누며 개선한 문제도 있었고, 개인이 해결하기에 고민이 많아 서로 공유하며 다루었던 문제들도 있었다 협업하였기에 짧은 시간 안에 이렇게 멋진 결과물을 만들 수 있었다

회고-이홍비

결과 및 성과

- 계획했던 기능 대부분 구현 완료 : 시간이 촉박한 관계로 기능 한두 개 구현 x ⇒ 추후 개발 예정
- 프로젝트 완성이라는 목표를 달성하기 위해 **팀원과 힘을 모아 으쌔으쌔 협력함**
⇒ 처음 보는 타인과 소통하고 협력하는 능력이 향상됨

향후 계획

- 앞으로도 열심히 수업 참여 예정
- 이번 프로젝트의 주제 선정 때 생각했던 '영화' 관련 웹 애플리케이션 개발하고 싶음
- 팀원과 함께 설계하고 구현한 이 프로젝트를 확장할 기회가 있다면 확장하고픈

후기 및 배울 점

- 주식 관련 처리를 제안했는데 **다들 흔쾌히 그러자고 해 주시고 잘해 주셔서 좋았음**
- 조별 과제의 장점을 느낀 프로젝트
⇒ 모르는 것, 오류 등을 묻고 함께 고민할 동지가 있어 든든한 느낌
⇒ 손발이 잘 맞았던 것 같음
⇒ 프로젝트를 진행하며 즐거웠음
- 프로젝트 시간 : 나홀
⇒ 시간이 촉박한 상황에서도 서로 다독이며 잘 프로젝트를 수행한 것 같음
⇒ 촉박했던 만큼 해냈을 때 쾌감이란 짜릿함
- 앞에서 끌어 주고 뒤에서 밀어 주며 함께 마라톤을 완주한 느낌
⇒ 누군가에게 도움을 줄 수 있는 관참은 조원이 되고 싶음
- **예쁘게 완성된 프로젝트를 보니 뿌듯함** : "이 맛에 프로그래밍 하지.."

마무리

- 조 발표 후 초반에는 어색하고 서먹했으나 지금은 많이 나아진 것 같음
- 앞으로 남은 두 번의 프로젝트에서 또 만날 수 있을지 모르겠지만, 이번 프로젝트를 계기로 관참은 인연으로 남길 바람

```
1/* ParkingAreaService.java
2 * Parking Lot Management Service
3 * 주차 구역 서비스
4 * 작성자 : semi_lion2 (고민정, 박창조, 이홍비, 허선호)
5 * 최종 수정 날짜 : 2024.12.13
6 *
7 * =====
8 * 프로그램 수정 / 보완 이력
9 * =====
10 * 작업자      날짜      수정 / 보완 내용
11 * =====
12 * 고민정      2024.12.12    ParkingArea 서비스 생성
13 * 고민정      2024.12.12    id, location으로 ParkingArea 조회 메서드 작성
14 * 박창조      2024.12.13    사용 가능한 주차공간 및 전체 주차 공간 카운트 메서드 추가
15 * =====
16 */
17
18 package team2.parking.service;
19
20 import org.springframework.stereotype.Service;
21
```


좋았던 점

- git의 사용 방법 및 Docker를 통한 인프라 구축+실행 방법을 자세히 알 수 있었음
- 내 의견만 내세우기보다 다른 입장도 공감하며 함께 조율하려는 모습이 보여서 좋았음
- 처음하는 개발 프로젝트라 다소 난해했지만 시간이 갈수록 완성되어가는 결과를 보니 뿌듯했음

아쉬운 점

- 기능을 설계할 때는 욕심이 많았으나 시간과 스킬 부족으로 실제로 구현하지 못한 부분이 있음
- 기능을 만드는 것 보다 예상하지 못했거나 처음 보는 오류들의 발생 원인을 파악하고 수정하느라 시간을 많이 소모해, 다른 팀원들보다 코드 기여가 적었던 것

