

프로젝트 소개

개발 환경

DB 설계

컨트롤러 설계

프론트 구현

주요 코드

후기

나만의 클래스를 만들어보자



나만의 특색이 담긴 공방을 개설하고,
다른 사람이 만든 클래스를 들을 수 있다.

github

<https://github.com/valorjj/gongbang-jj.git>

notion

<https://cedar-echinacea-afc.notion.site/GONGBANG-6af052ece8454a40b679e37e59586796>

프로젝트 소개

개발 환경

DB 설계

컨트롤러 설계

프론트 구현

주요 코드

후기

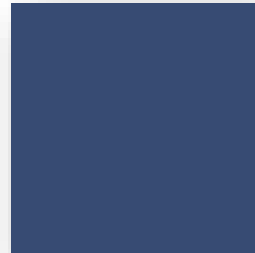
나만의 클래스를 만들어보자

타겟 : 모든 일반인

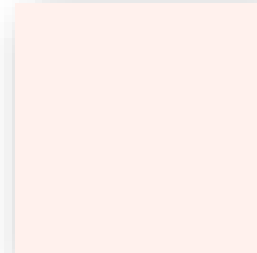
프로젝트 메인 컬러



#ff9a9a



#374b73



#fff1ed

프로젝트 소개

개발 환경

DB 설계

컨트롤러 설계

프론트 구현

주요 코드

후기

나만의 클래스를 만들어보자

개요

프로젝트 목적

카테고리, 지역을 선택해서 나만의 원데이 클래스를 개설한다

다른 회원은 결제 후 포인트 차감 형식으로 개설된 강좌를 신청할 수 있다.

프로젝트 소개

개발 환경

DB 설계

컨트롤러 설계

프론트 구현

주요 코드

후기

Backend



frontend



db



server



프로젝트 소개

개발 환경

DB 설계

컨트롤러 설계

프론트 구현

주요 코드

후기

나만의 클래스를 만들어보자

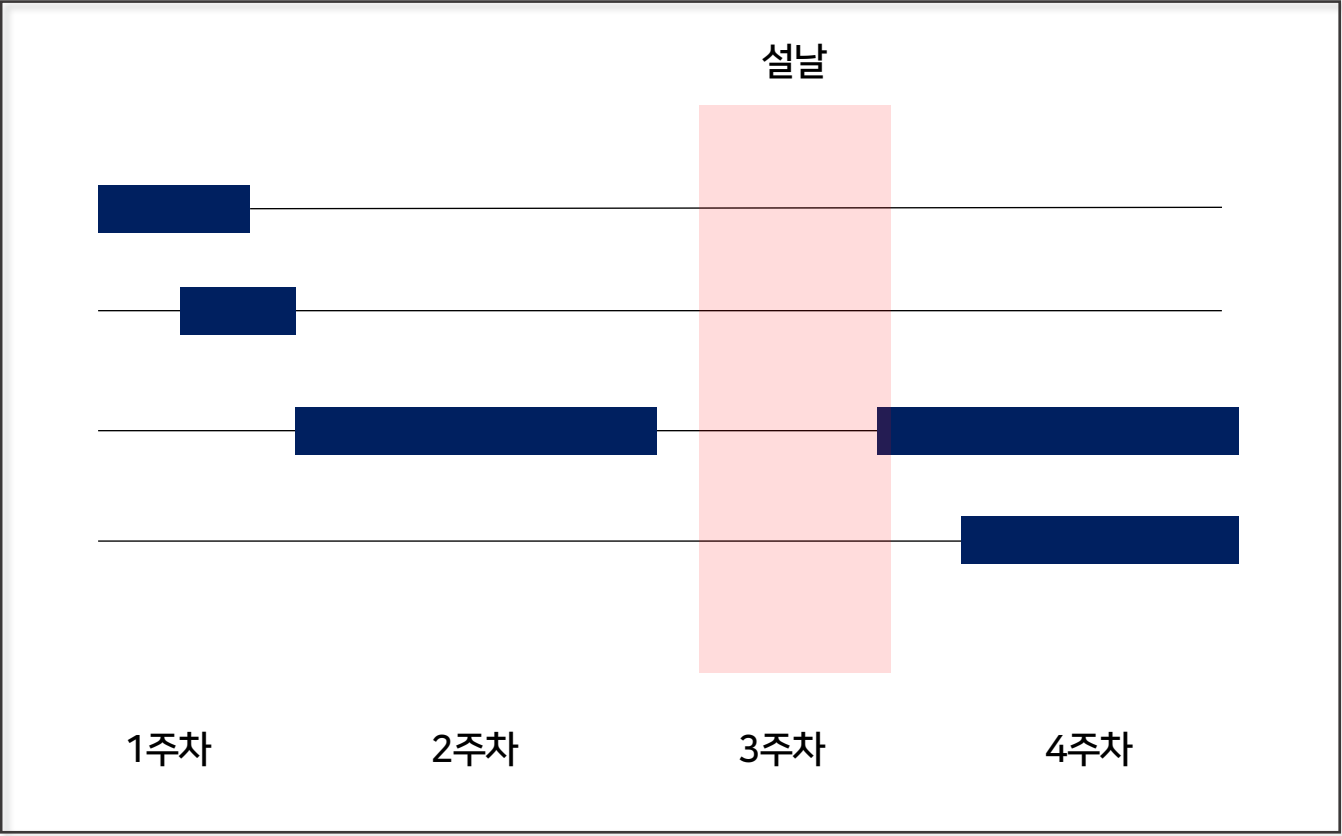
프로젝트 기간

분석 및 기획 단계

UI 및 DB 설계

UI 및 기능 구현

테스트 및 디버깅



프로젝트 소개

개발 환경

DB 설계

컨트롤러 설계

프론트 구현

주요 코드

후기

나만의 클래스를 만들어보자

역할분담

김정진	김태호	신우현	조지훈
DB 설계	프론트 전반	쪽지 기능	클래스 메인 페이지
검색 관련 모든 Query 문 작성	회원 가입 및 로그인	결제 API	초기 레이아웃
카카오맵 API DB 연동		TMAP API	개인 채널 페이지
캘린더 작성 및 DB 연동		별점 리뷰	
클래스 등록 및 수정, 삭제		클래스 상태 변경	
관리자 페이지		좋아요 기능	
개인 통계 페이지			

프로젝트 소개

개발 환경

DB 설계

컨트롤러 설계

프론트 구현

주요 코드

후기

나만의 클래스를 만들어보자

주요 기능

클래스 등록, 수정 삭제 (CRUD)

검색과 연동해서 지도 출력

Javascript 이용한 캘린더 출력 및 DB 연동

관리자 페이지 : 등록 현황 및 차트

개인 통계 페이지 : 등록 현황 및 차트

프로젝트 소개

개발 환경

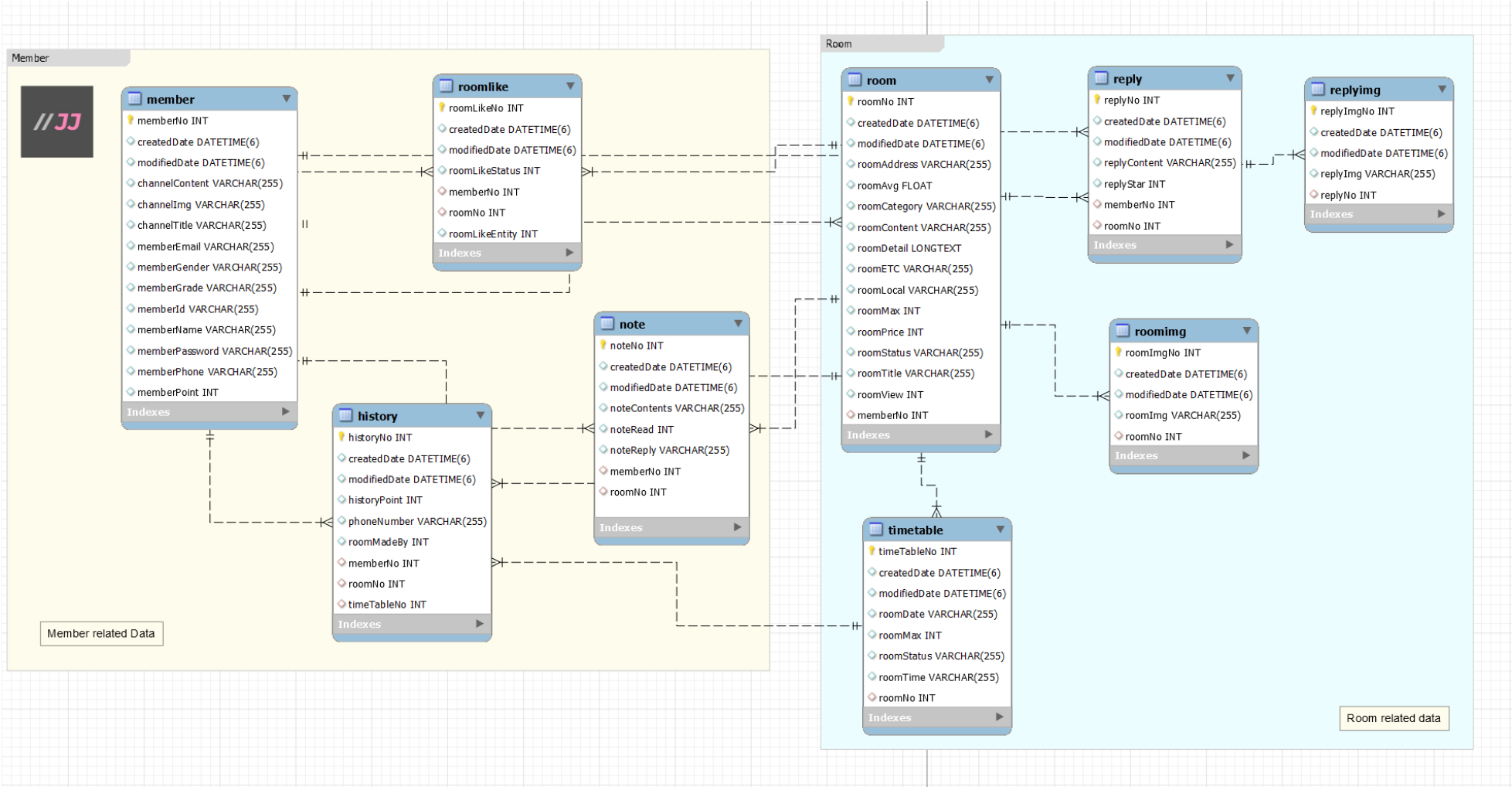
DB 설계

컨트roller 설계

프론트 구현

주요 코드

후기



프로젝트 소개

개발 환경

요구사항 정의

컨트롤러 설계

프론트 구현

주요 코드

후기

인덱스			
ID	요구사항명	요구사항설명	관련자
index	메인페이지	- header, main, footer 구성	모든 회원
헤더			
ID	요구사항명	요구사항설명	관련자
header	헤더	<div>- 사이트 로고, 검색창, 드롭다운 메뉴로 회원정보, 예약내역, 내가 등록한 클래스, 포인트 충전, 쪽지, 개인 채널, 로그아웃 메뉴가 있다. 드롭 다운에는 채널에서 등록한 사진이 표시된다. 등록된 사진이 없다면 관리자가 등록한 디폴트 사진이 출력된다.</div> <div>- 지역, 카테고리, 키워드 검색</div> <div>- 검색 후 해당 리스트, 등록된 주소를 카카오 맵에 표시</div> <div>- 로그인 모달 창에서 OAuth 2.0 사용한 SNS 로그인 구현</div>	모든 회원

공방 개설			
ID	요구사항명	요구사항설명	관련자
room_register_detail	공방 개설 페이지	<div><div>- 클래스 이름, 클래스 짧은 설명, 클래스 상세 설명, 사진, 지역, 카테고리, 주소, 가격, 총 인원 등을 입력한다.</div><div>- 개설된 클래스는 내가 만든 클래스 화면에서 확인할 수 있다.</div><div>- 관리자가 상태를 '승인완료'로 바꾸기 전에는 메인 화면이나 검색 대상에서 제외된다.</div></div>	모든 회원

개설된 공방 상세페이지			
ID	요구사항명	요구사항설명	관련자
room_view	상세페이지	<div><div>- 등록된 사진(캐러셀), 정보 출력</div><div>- 좋아요 기능</div><div>- 별점 리뷰</div><div>- 캘린더에 개설되어있는 강의가 있는 날짜만 선택 가능</div><div>- 날짜 선택 후 현재 예약 가능한 인원 수 출력</div><div>- 해당 날짜에 여러 개 강의를 있을 때 하나 선택 후 결제 진행</div></div>	모든 회원

프로젝트 소개

개발 환경

요구사항 정의

컨트롤러 설계

프론트 구현

주요 코드

후기

결제 페이지			
ID	요구사항명	요구사항설명	관련자
member_payment	결제페이지	<div>- 결제하는 공방 정보, 회원 정보 출력</div> <div>- 각 예약건에 따라 무조건 연락처를 입력해야함</div> <div>- 포인트 차감하며 결제 진행</div>	모든 회원
마이채널			
ID	요구사항명	요구사항설명	관련자
channel	개인채널	<div>- 회원가입 시 내용, 사진을 등록하지 않으면 클래스를 개설할 수 없다.</div> <div>- 본인이 개설한 강의, 개설된 강의에 작성된 리뷰가 출력된다.</div>	모든 회원

개설된 공방 리스트			
ID	요구사항명	요구사항설명	관련자
room_list	헤더 검색 후 출력 페이지	<div><div>- 왼쪽에는 개설된 리스트, 오른쪽에는 해당 위치를 지도에 표시</div><div>- 아무런 검색값도 없을때는 모든 데이터 출력</div><div>- 검색있을 때는 한 페이지에 출력되는 리스트만 지도에 출력 (한 페이지에 2개씩 결과가 출력되도록 설정)</div></div>	모든 회원

메인 페이지			
ID	요구사항명	요구사항설명	관련자
main	메인페이지	<div><div>- 최상단에 광고 형태의 동영상, 사진 캐러셀</div><div>- 지역, 카테고리 분류 캐러셀(이미지 파일 캐러셀)</div><div>- '승인완료' 처리된 개설된 공방 캐러셀</div><div>- 작성된 리뷰 캐러셀</div></div>	모든 회원

개설된 공방 리스트			
ID	요구사항명	요구사항설명	관련자
room_list	헤더 검색 후 출력 페이지	<div><div>- 왼쪽에는 개설된 리스트, 오른쪽에는 해당 위치를 지도에 표시</div><div>- 아무런 검색값도 없을때는 모든 데이터 출력</div><div>- 검색있을 때는 한 페이지에 출력되는 리스트만 지도에 출력 (한 페이지에 2개씩 결과가 출력되도록 설정)</div></div>	모든 회원

예약 내역			
ID	요구사항명	요구사항설명	관련자
history_list	예약 내역	<div><div>- 첫 진입 시 회원이 결제했던 모든 리스트 출력</div><div>- 오른쪽 상단 캘린더에 날짜를 클릭하면 해당 날짜에 예약한 내역만 출력</div></div>	모든 회원

개설된 공방 리스트			
ID	요구사항명	요구사항설명	관련자
room_list	헤더 검색 후 출력 페이지	<div><div>- 왼쪽에는 개설된 리스트, 오른쪽에는 해당 위치를 지도에 표시</div><div>- 아무런 검색값도 없을때는 모든 데이터 출력</div><div>- 검색있을 때는 한 페이지에 출력되는 리스트만 지도에 출력 (한 페이지에 2개씩 결과가 출력되도록 설정)</div></div>	모든 회원

예약 내역			
ID	요구사항명	요구사항설명	관련자
history_list	예약 내역	<div><div>- 첫 진입 시 회원이 결제했던 모든 리스트 출력</div><div>- 오른쪽 상단 캘린더에 날짜를 클릭하면 해당 날짜에 예약한 내역만 출력</div></div>	모든 회원

프로젝트 소개

개발 환경

요구사항 정의

컨트롤러 설계

프론트 구현

주요 코드

후기

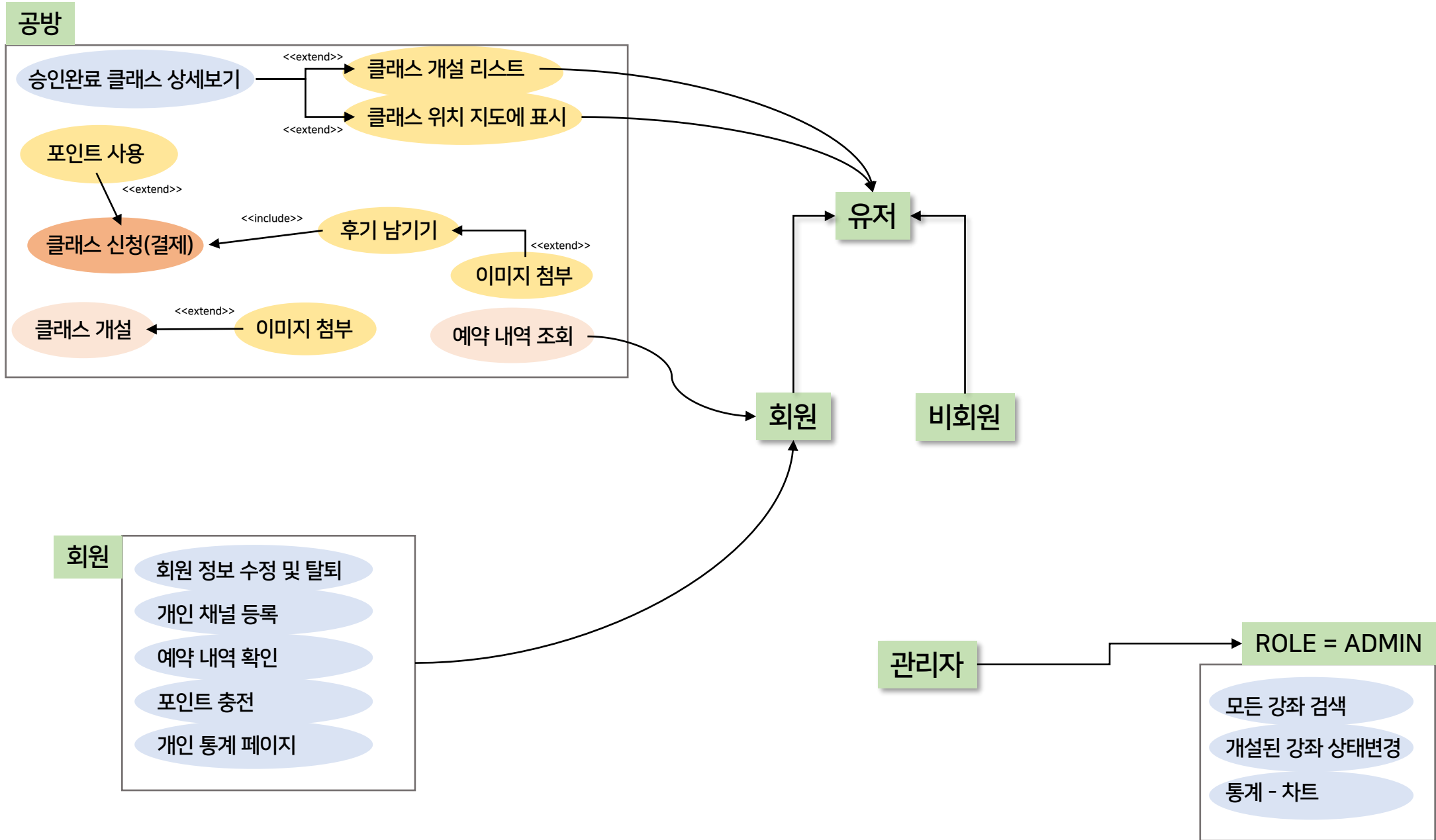
관리자 페이지

ID	요구사항명	요구사항설명	관련자
adminList	관리자 페이지	<ul style="list-style-type: none">- 카테고리, 지역, 키워드 검색 가능한 테이블 출력- 아무 검색값도 없으면 모든 개설된 강좌 출력- 개설된 강좌의 상태를 변경하거나 삭제 가능- 통계 탭에서 차트 출력- 연도별, 달, 일, 카테고리, 지역 별 차트 출력- amchart api 사용	관리자

개인 통계 페이지

ID	요구사항명	요구사항설명	관련자
calculate	개인 통계 페이지	<ul style="list-style-type: none">- 개설한 강좌에 등록한 모든 회원 목록 출력- 개별 강좌에 등록한 회원 목록 출력- 통계 탭에서 차트 출력- '일' 별로 등록한 인원수 line chart 출력- amchart api 사용	모든 회원

유스케이스



프로젝트 소개

개발 환경

DB 설계

컨트롤러 설계

프론트 구현

주요 코드

후기

사용자

클래스 등록

관리자 승인 대기

날짜, 시간 선택

클래스 상세 페이지

캘린더에 등록된 클래스 출력

좋아요, 문의 쪽지 남기기

별점 후기 남기기

클래스 신청

예약 가능한 날짜에 예약 신청

결제

lampoort 이용한 결제

예약 내역

캘린더 DB 연동

정산 페이지

내가 개설한 강좌에 등록된 회원 목록

Amchart 이용한 linechart

프로젝트 소개

개발 환경

DB 설계

컨트롤러 설계

프론트 구현

주요 코드

후기

관리자

등록 현황

날짜, 카테고리 기준 검색

통계

amchart api 이용

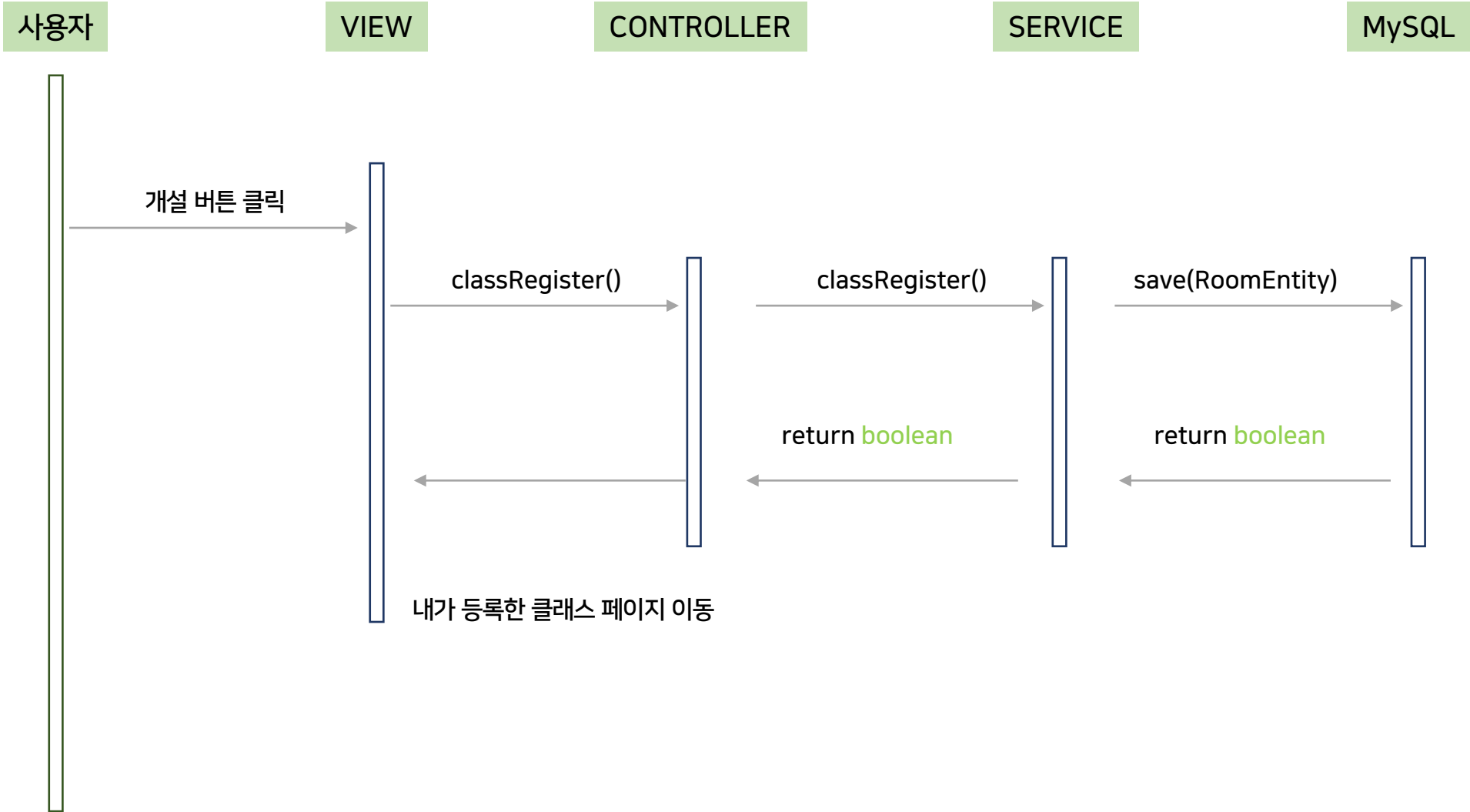
연도 - 일일 등록 회원 수 line chart

월 별로 개설된 클래스 개수

일 기준 카테고리 별 수강 신청 인원 수

- 프로젝트 소개
- 개발 환경
- DB 설계
- 컨트롤러 설계
- 프론트 구현
- 주요 코드
- 후기

클래스 등록



프로젝트 소개

개발 환경

DB 설계

컨트롤러 설계

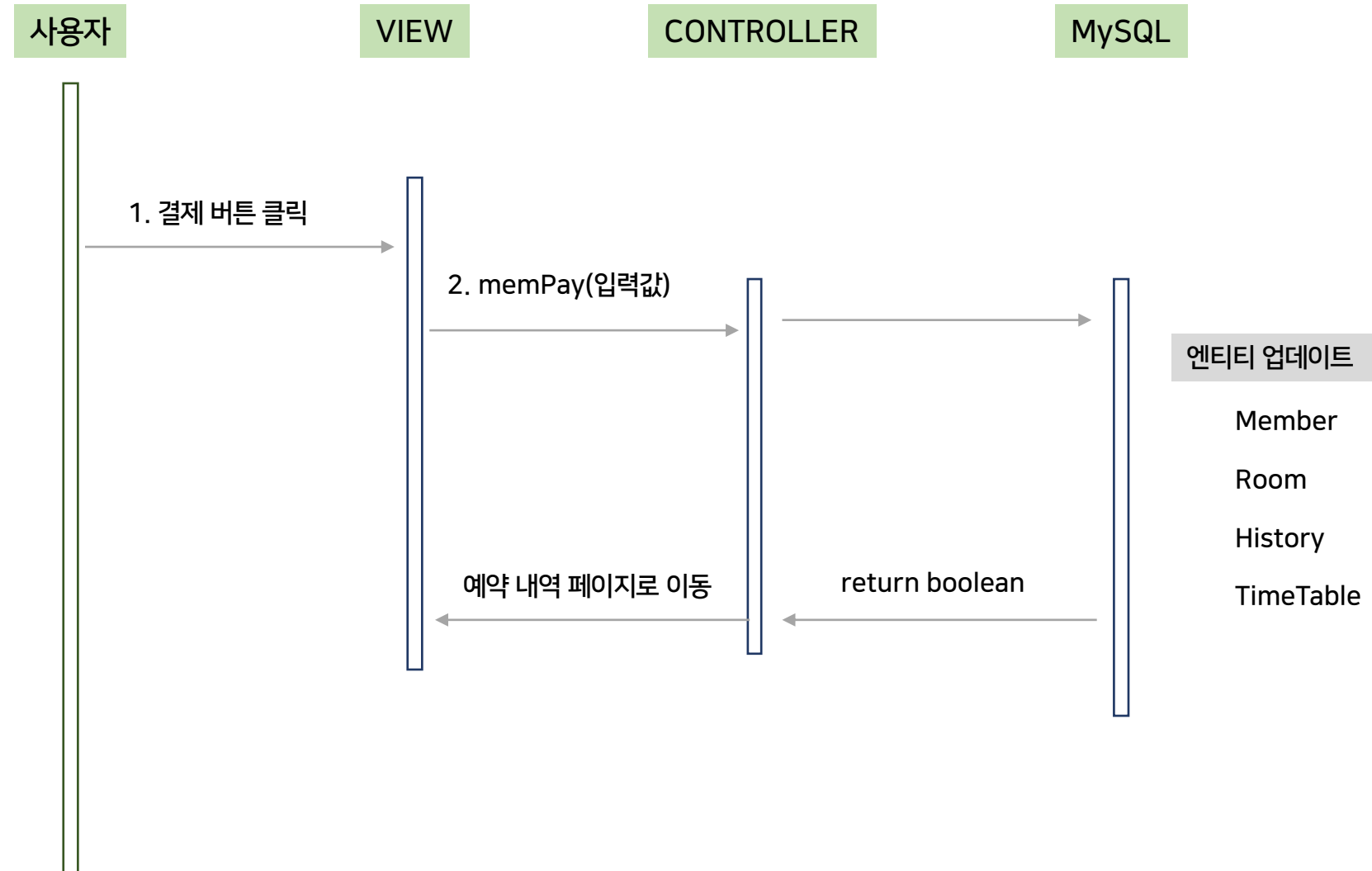
프론트 구현

주요 코드

후기

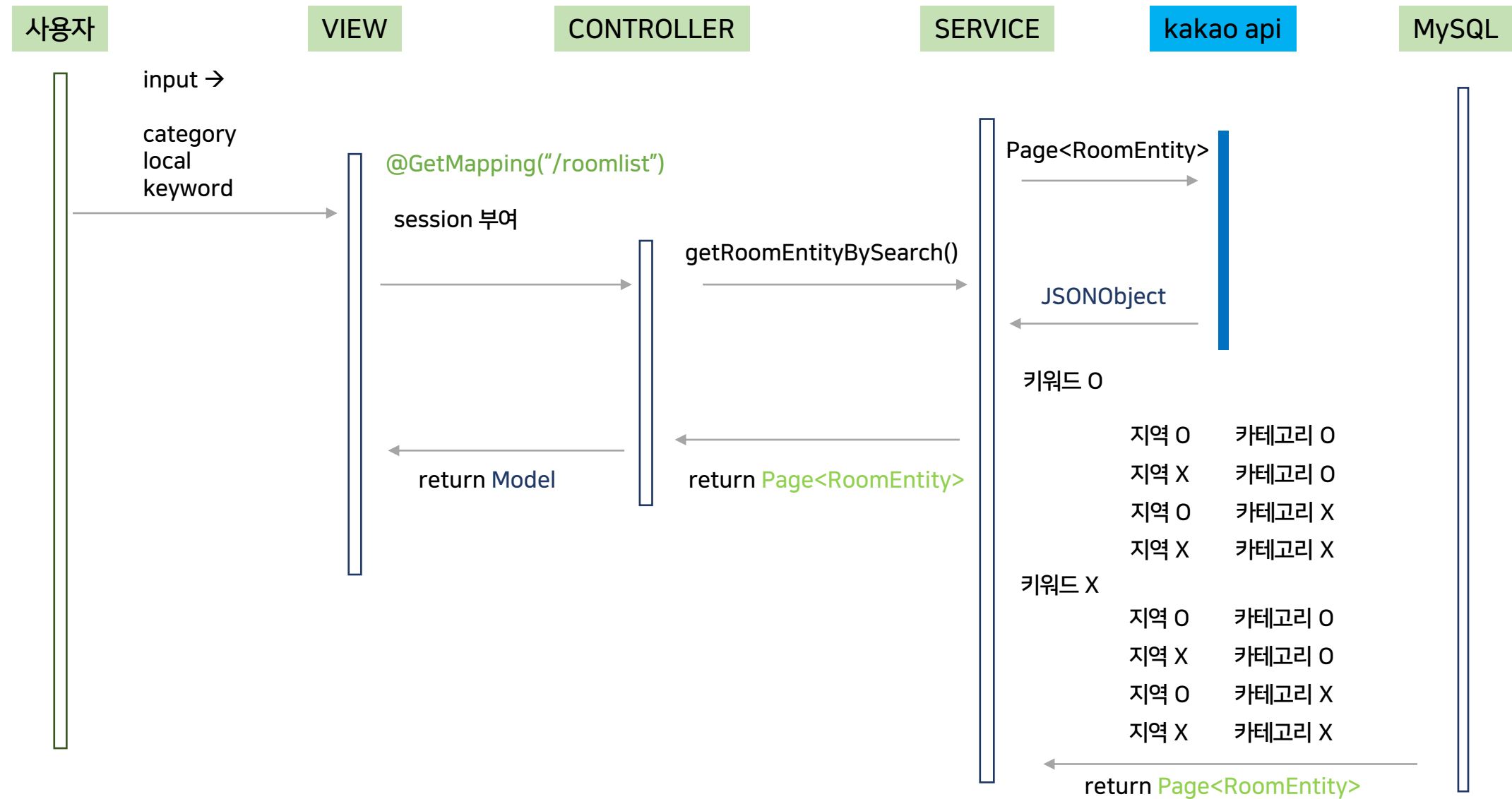
클래스 신청(결제)

SERVICE 생략, `@Transactional` 사용해 repository 접근



- 프로젝트 소개
- 개발 환경
- DB 설계
- 컨트롤러 설계
- 프론트 구현
- 주요 코드
- 후기

검색 및 지도 출력



프로젝트 소개

개발 환경

DB 설계

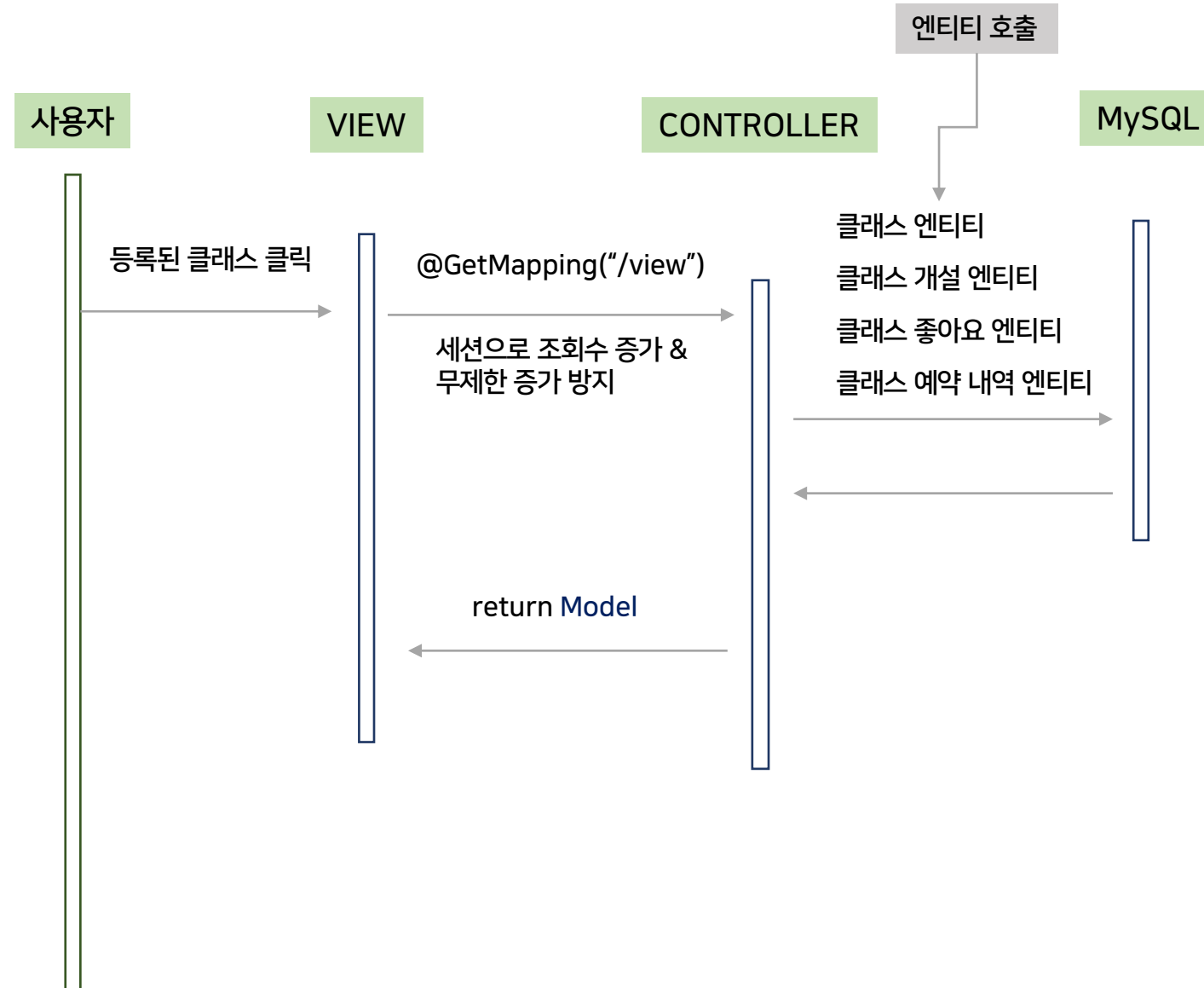
컨트롤러 설계

프론트 구현

주요 코드

후기

클래스 상세보기



- 프로젝트 소개
- 개발 환경
- DB 설계
- 컨트롤러 설계
- 프론트 구현
- 주요 코드
- 후기

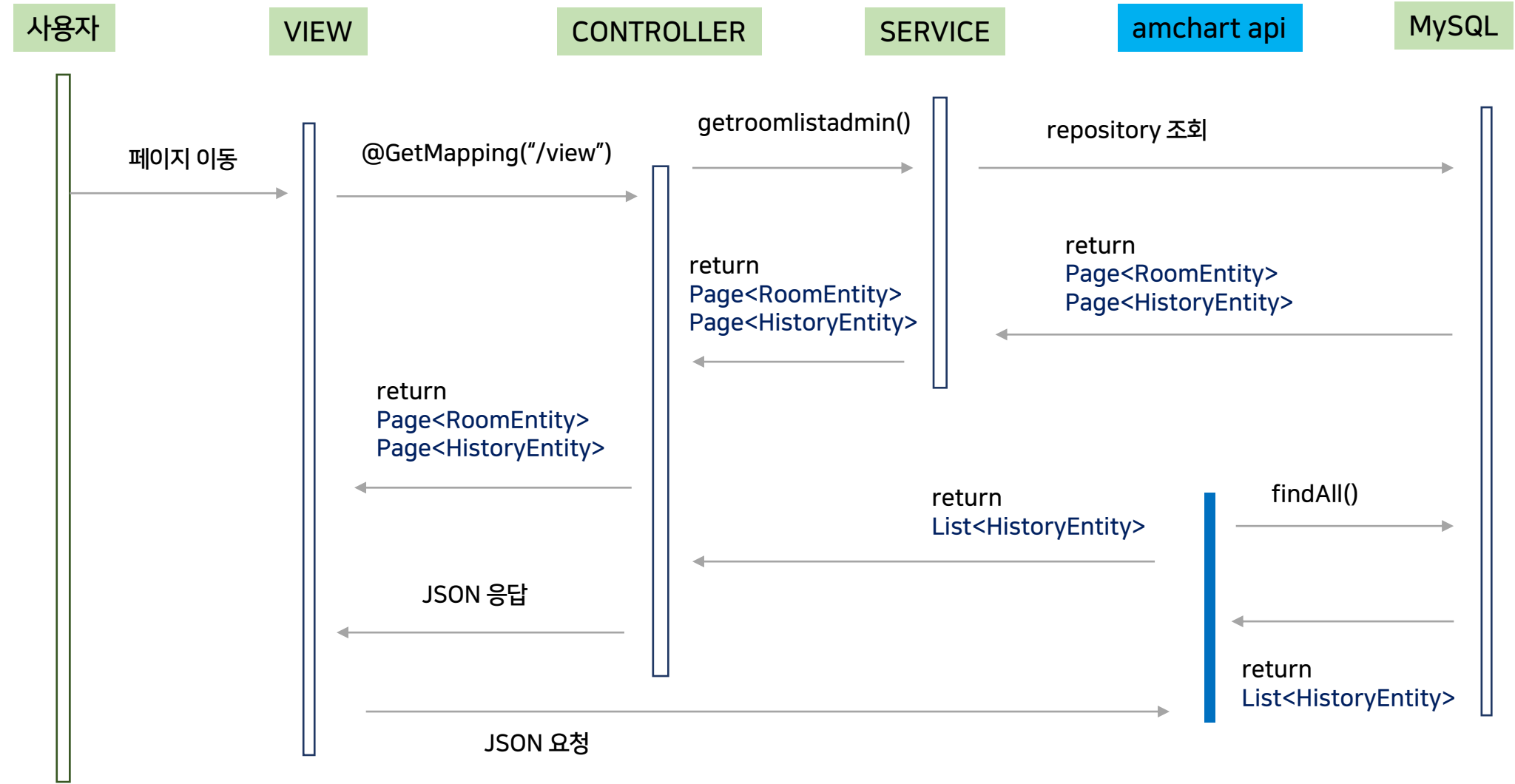
개설현황 및 차트 출력

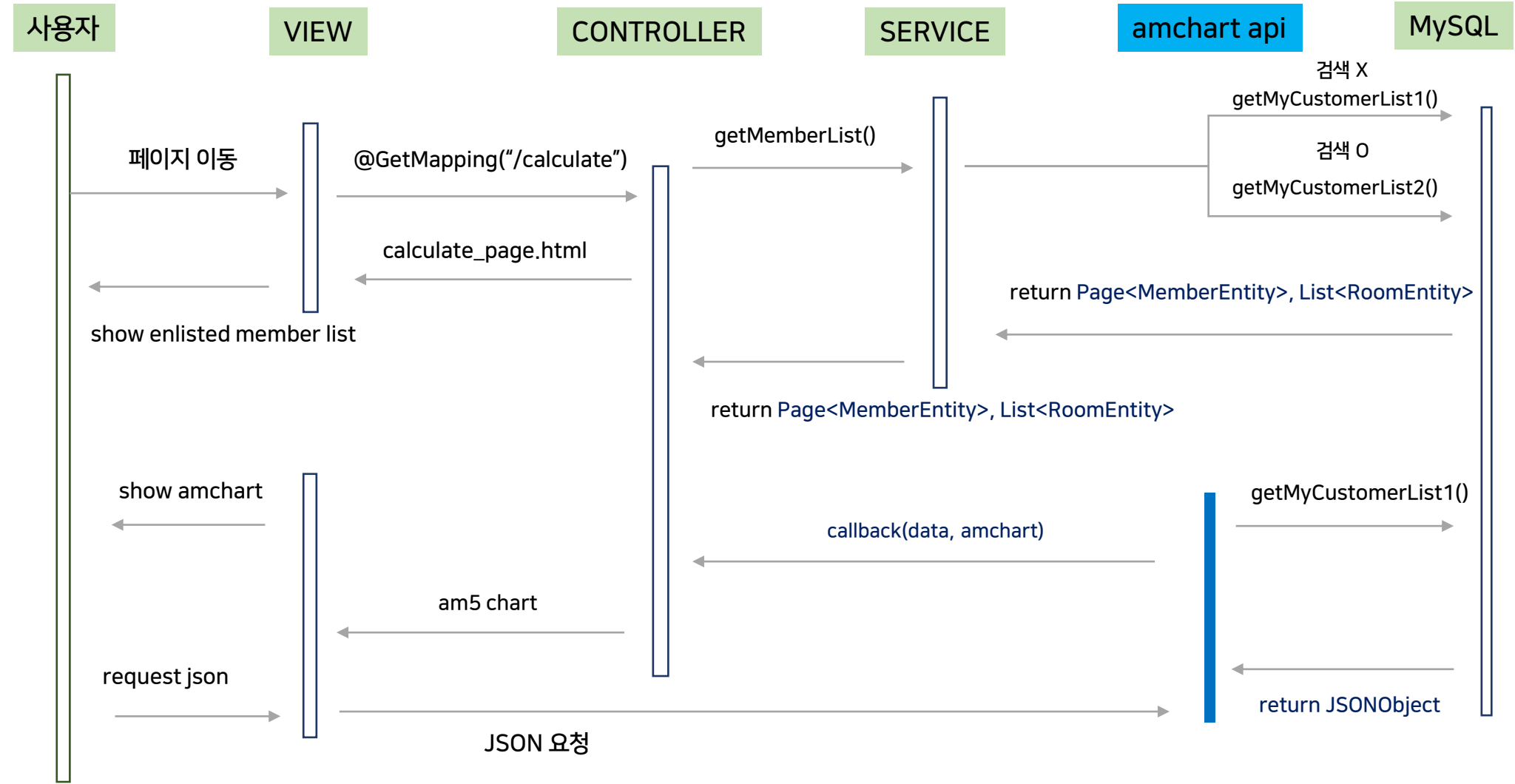
관리자

사용자

전체

개인 정산 페이지





프로젝트 소개

개발 환경

DB 설계

컨트roller 설계

프론트 구현

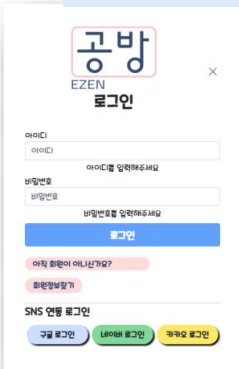
주요 코드

후기

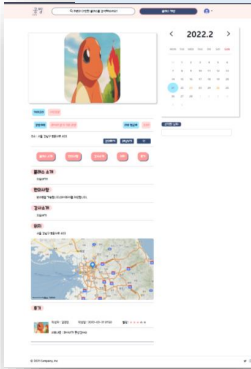
메인



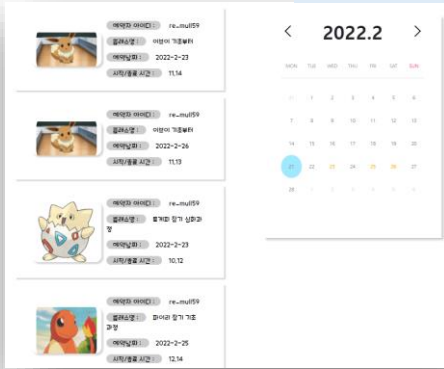
로그인모달



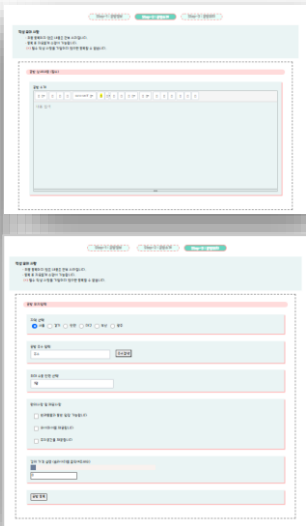
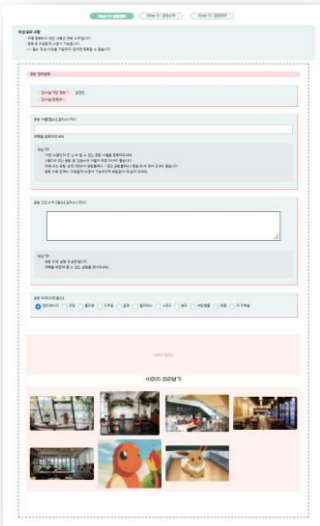
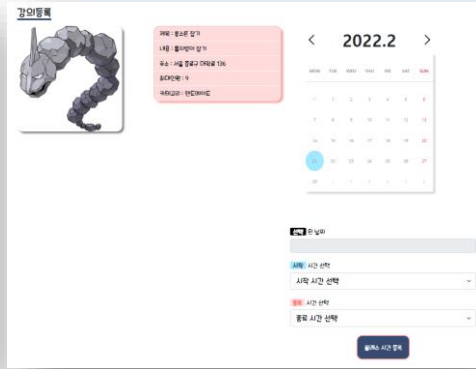
상세페이지



예약내역



날짜, 시간 등록 후 개설



클래스 등록



포인트 충전



내가 등록한 클래스 내역



문의 쪽지



개인 채널

후기

찾아오는 길

찾아오는 길 ×

교통직적자도전
교통장별 보물 창고

광주 기

종 거리 : 68.7km, 중 시간 : 67분, 예상 택시 요금 : 64500원

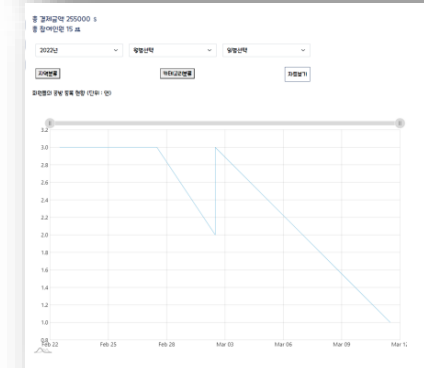
총 불하금액
295000
총 불하인원
13

연도별액 ▼ 연월별액 ▼ 일월별액 ▼

지역별액 지역구별액 지역별기

세종시 ▼ 북부 연월 (단위: 세원)

연월	금액 (세원)
1월	1.0
2월	0.0
3월	0.0
4월	0.0
5월	1.9
6월	0.0
7월	1.0
8월	0.0
9월	0.0
10월	0.0
11월	1.9
12월	0.0



동북를 상대변경

평복을 상대를 변경합니다. 변경하시겠습니까?

검토중

승인중

승인완료

승인거부

취소

프로젝트 소개

개발 환경

DB 설계

컨트롤러 설계

프론트 구현

주요 코드

후기

내가 개설한 강좌에 등록된 회원 목록

총 신청 인원 수
4

최신순

개설한 클래스

검색

검색

회원번호	회원아이디	회원이메일	연락처
8	test5	test5@gmail.com	010-1234-1234
7	test4	test4@gmail.com	010-1234-1234
1	re_mull59	re_mull59@naver.com	연락처가 등록되지 않았습니다.
4	test1	test1@gmail.com	010-1234-1234

처음

이전

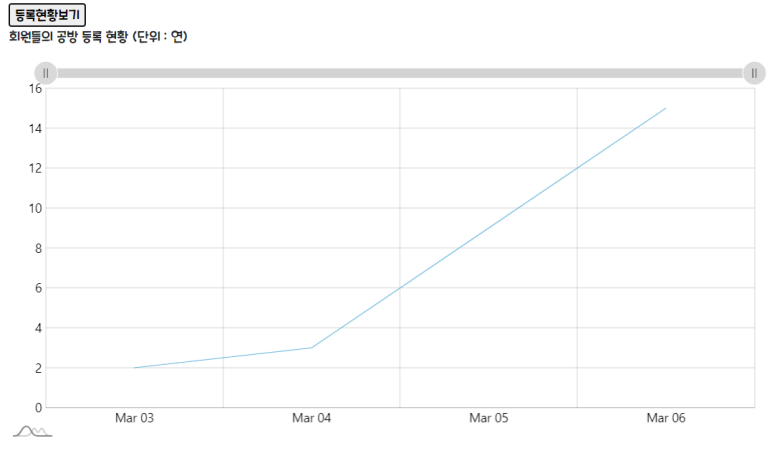
1

다음

마지막

- 개설한 클래스
- 개설한 클래스
- [서울] 쿠키만들기
- [서울] 기타배우기
- [서울] 원데이 축구 포인트 강의
- [서울] 안녕
- 테스트

내가 개설한 강좌에 등록된 '일' 기준 회원 수



프로젝트 소개

개발 환경

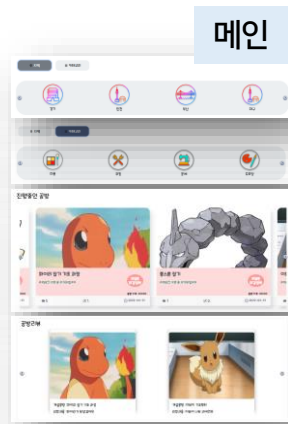
DB 설계

컨트roller 설계

프론트 구현

주요 코드

후기



메인 화면 출력

카테고리, 지역 아이콘 캐러셀 → 해당 개설된 강좌 검색 기능

가장 최근에 개설된 순으로 정렬된 강좌 → 무한 캐러셀

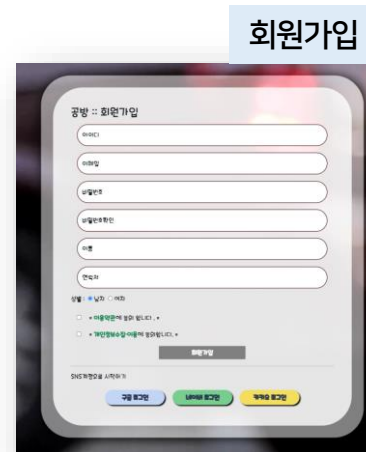
가장 최근에 작성된 순으로 정렬된 리뷰 → 무한 캐러셀



로그인 모달

아이디, 비밀번호를 입력해야만 로그인 버튼 활성화

구글, 네이버, 카카오 SNS 소셜 연동 로그인 지원



회원가입

아이디 : 영문, 숫자 5-15글자, 자동 중복체크

비밀번호 :

영문 대소문자, 숫자, 특수문자가 적어도 하나는 포함된 8-15글자

이메일 유효성 : 유효한 이메일 체크, 자동 중복체크

프로젝트 소개

개발 환경

DB 설계

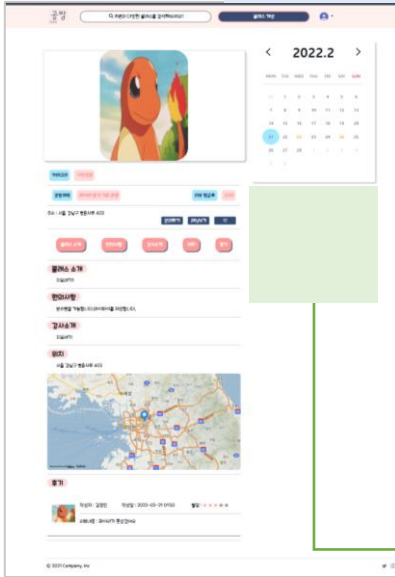
컨트roller 설계

프론트 구현

주요 코드

후기

상세페이지



상세 페이지

등록된 사진 및 정보 출력

좋아요 - 강의를 등록한 본인은 막혀있음

리뷰 작성 - 이 강의를 수강했던 다른 회원만 작성가능

캘린더에 등록된 강좌가 있다면 해당 날짜만 하이라이트

선택한 날짜

2022-2-23

클래스 이름 : 파이어라 잡기 기초 과정
시작시간 : 14 종료시간 : 16
지역 : 서울
신청 가능한 인원 : 4

선택한 날짜

2022-2-22

클래스 이름 : 뽕스톤 잡기
시작시간 : 11 종료시간 : 13
지역 : 서울
신청 가능한 인원 : 6

선택

본인이 개설한 강좌는 선택 버튼 출력 X

선택한 날짜

2022-2-22

날짜 : 2022-2-22
시작시간 : 11 종료시간 : 13
최대 수강 신청 인원 : 3명
2명
선택한 인원 수 : 2

가격 : 50000
신청

선택한 인원수에 따른 가격 출력

클래스 신청

특정 날짜 클릭 시 controller 접근 → json 형태로 javascript 에 전달

```
// 해당 날짜를 선택했을 때의 이벤트를 부여합니다.  
// DB 를 조회해서 선택한 날짜에 개설된 강좌 정보를 불러옵니다.  
$.ajax({  
  url: "/room/toJSON", // RoomEntity 를 JSON 형태로 받아온다.  
  data: {"activeId" : date, "roomNo" : roomNo},  
  method: "GET",  
  async: false,  
  contentType: "application/json",  
  success: function(data){
```

```
// 받아온 정보들을 html 로 넘겨준다.  
// 반복문을 돌아야하니, 다시 controller 로 정보를 넘겨준 뒤  
// hashmap 형태로 받아서 model 로 넘겨준다.  
var rooms = $(data.json).map(function(i, room) {  
  var roomNo = room.roomNo;  
  var roomCategory = room.category;  
  var roomTitle = room.title;  
  var roomBeginTime = room.beginTime;  
  var roomEndTime = room.endTime;  
  var roomLocal = room.local;  
  var roomMax = room.max;  
  var roomDate = room.date;  
  var realMax = room.timeMax;
```

```
@GetMapping("/toJSON")  
@ResponseBody  
public JSONObject getRoomEntityByTimeTableToJson(@RequestParam("activeId") String roomDate,  
                                                  @RequestParam("roomNo") int roomNo) {  
    HttpSession session = request.getSession();  
    MemberDto loginDto = (MemberDto) session.getAttribute( name "loginDto");  
    int memberNo = loginDto.getMemberNo();  
    JSONObject jsonObject = new JSONObject(); // json  
    JSONArray jsonArray = new JSONArray(); // json  
    // roomNo 에 해당하는 TimeTable 엔티티만 리스트에 담아서 호출한다.  
    List<TimeTableEntity> timeTableEntities = timeTableRepository.getTimeTableByRoomNo(roomNo);  
    // roomNo 에 해당하는 개설된 강좌 전체를 for 문으로 조회한다.  
    for (TimeTableEntity timeTableEntity : timeTableEntities) {  
        int roomNoFromTimeTable = timeTableEntity.getRoomNo();  
        // TimeTable 에 저장된 RoomNo 를 사용해서 Room Entity 를 호출하되, '승인완료' 인 클래스만 리턴한다.  
        RoomEntity roomEntity = roomRepository.findRoomByStatusAndNo(roomStatus: "승인완료", roomNoFromTimeTable);  
        // 선택한 date 에 해당하는 Room 정보를 json 에 저장시킨다.  
        if (timeTableEntity.getRoomDate().equals(roomDate)) {  
            JSONObject data = new JSONObject(); // json  
            try {  
                data.put("roomNo", roomEntity.getRoomNo());  
                data.put("category", roomEntity.getRoomCategory());  
                data.put("title", roomEntity.getRoomTitle());  
                data.put("date", timeTableEntity.getRoomDate());  
                data.put("beginTime", timeTableEntity.getRoomTime().split(" ")[0]);  
                data.put("endTime", timeTableEntity.getRoomTime().split(" ")[1]);  
                data.put("local", roomEntity.getRoomLocal());  
                data.put("max", roomEntity.getRoomMax());  
                data.put("timeMax", timeTableEntity.getRoomMax());  
                jsonArray.add(data);  
            } catch (Exception e) {  
                e.printStackTrace();  
            }  
        }  
    }  
    jsonObject.put("rooms", jsonArray);  
    return jsonObject;
```

내가 개설한 강의

남이 개설한 강의

<div> 태그로 감싼 뒤 room_view.html 에 append()

```
var roomhtml = "<div class='col-md-8'>";  
roomhtml += "<div class='classContent' style='border: 3px solid #374b73; background-color: #ffffff; color: #374b73; padding: 0.5rem;'>";  
roomhtml += "<div> 클래스 이름 : " + roomTitle + "</div>";  
roomhtml += "<div> <span> 시작시간 : " + roomBeginTime + "</span> <span> 종료시간 : " + roomEndTime + "</span> </div>";  
roomhtml += "<div> 지역 : " + roomLocal + "</div>";  
roomhtml += "<div> 신청 가능한 인원 : " + realMax + "</div>";  
roomhtml += "</div>";  
roomhtml += "</div>";  
$("#time-select-inner").append(roomhtml);
```

클래스 신청

강좌 신청하는 버튼 클릭

Controller 에서 신청이 가능한지 여부 (유효성 검사)

```
$("#myClassRegisterBtn").on("click", function() {  
    var person = $("#input-person").val();  
    var roomNo = $("#register-room-no").val();  
    var roomDate = $("#register-room-date").val();  
    var roomTime = $("#register-room-time").val();  
    // person : 선택한 인원 수  
    // roomNo : 클래스 식별 번호  
    // roomDate : 날짜  
    // roomTime : 시작시간, 끝나는시간 (TimeTable 에 있는 것과 동일)  
    $.ajax({  
        url: "/member/registerClass",  
        data: {  
            "roomNo": roomNo,  
            "roomDate": roomDate,  
            "roomTime": roomTime,  
            "person": person,  
            "price": price  
        },  
        method: "GET",  
        success: function(data) {
```

```
@GetMapping("/registerClass")  
@ResponseBody  
@Transactional  
public String registerClass(@RequestParam("roomNo") int roomNo,  
                             @RequestParam("roomTime") String classTime,  
                             @RequestParam("roomDate") String roomDate,  
                             @RequestParam("person") int person,  
                             @RequestParam("price") int price) {
```

포인트 부족 : return 3
인원시 꽉 찬 경우 : return 2
신청이 가능 한 경우 : return 1

연락처를 무조건 입력해야 넘어갈 수 있다.

```
@GetMapping("/memberPaymentController")  
public String memberPaymentController(Model model,  
                                       @RequestParam("roomNo") int roomNo,  
                                       @RequestParam("roomDate") String roomDate,  
                                       @RequestParam("roomTime") String roomTime,  
                                       @RequestParam("person") int person,  
                                       @RequestParam("price") int price) {
```

```
// 인수가 많아서 dto 를 새로 만들어서 넘깁니다.  
RoomPaymentDto roomPaymentDto = RoomPaymentDto.builder()  
    .roomDate(roomDate)  
    .roomTime(roomTime)  
    .person(person)  
    .price(price)  
    .build();  
  
model.addAttribute( attributeName: "info", roomPaymentDto);  
model.addAttribute( attributeName: "member", memberEntity);  
model.addAttribute( attributeName: "room", roomEntity);
```

Member 엔티티
TimeTable 엔티티
Room 엔티티

roomPaymentDto 생성 (@Builder)

프로젝트 소개

개발 환경

DB 설계

컨트롤러 설계

프론트 구현

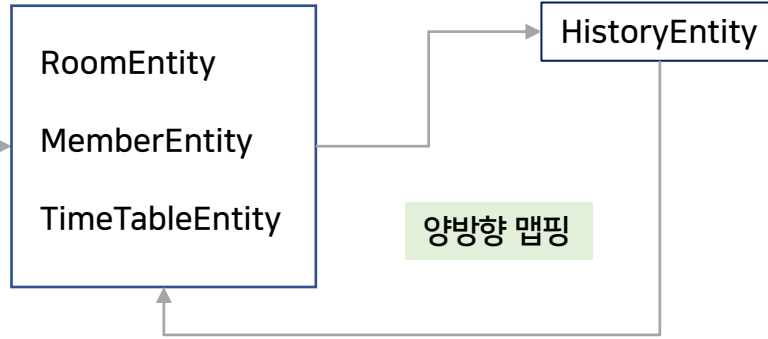
주요 코드

후기

클래스 신청

포인트 차감 및 DB 등록

```
@PostMapping("/memberPay")
@Transactional
public String memberPay(Model model,
    @RequestParam("roomNo") int roomNo,
    @RequestParam("roomTime") String classTime,
    @RequestParam("roomDate") String roomDate,
    @RequestParam("person") int person,
    @RequestParam("price") int price,
    @RequestParam("phoneNumber") String phone) {
```



개설된 강좌 인원 감소

클래스 등록하는 회원의 포인트 감소

꽉 찬다면 '모집완료' 로 상태 변경

```
// 5. 신청한 정원만큼 클래스 수용 인원을 감소시킵니다.
assert timeTableTmp != null;
timeTableTmp.setRoomMax(timeTableTmp.getRoomMax() - person);
// 수용 가능 인원이 '0' 명이 된다면, 클래스 상태를 '모집완료' 로 바꿉니다.
if (timeTableTmp.getRoomMax() <= 0) {
    timeTableTmp.setRoomStatus("모집완료");
}
```

```
// 8. 최종적으로 회원이 가진 포인트를 감소시킵니다.
memberEntity.setMemberPoint(memberEntity.getMemberPoint() - price);

int memberPoint;
if (historyEntity.getHistoryPoint() == 0) {
    memberPoint = 0;
} else {
    memberPoint = historyEntity.getHistoryPoint();
}
historyEntity.setHistoryPoint(memberPoint + price);
```


프로젝트 소개

개발 환경

DB 설계

컨트롤러 설계

프론트 구현

주요 코드

후기

개인 채널

내가 개설한 클래스에 등록된 회원 목록

총 신청 인원 수
4

최신순 ▼ 개설한 클래스 ▼ 검색

검색

회원번호	회원아이디	회원이메일	연락처
8	test5	test5@gmail.com	010-1234-1234
7	test4	test4@gmail.com	010-1234-1234
1	re_mull59	re_mull59@naver.com	연락처가 등록되지 않았습니다.
4	test1	test1@gmail.com	010-1234-1234

처음 이전 1 다음 마지막

개설한 클래스 ▼

- 개설한 클래스
- [서울] 쿠키만들기
- [서울] 기타배우기
- [서울] 원데이 축구 포인트 강의
- [서울] 안녕
- 테스트

검색이 없는 경우

모든 목록

특정 클래스 선택

```
// [키워드가 존재하지 않는 경우] 특정 멤버가 개설한 클래스에 등록된 회원 내역 불러오기
String myCustomerQuery1 = "select T.* from member T "
+ "where memberNo in "
+ "(select H.memberNo from member M "
+ "join history H on H.roomMadeBy = :memberNo group by H.memberNo)";

@Query(nativeQuery = true, value = myCustomerQuery1)
Page<MemberEntity> getMyCustomerList1(Pageable pageable, @Param("memberNo") int memberNo);
```

```
String getMyCustomerQuery3 = "select T.* from member T "
+ "where memberNo IN "
+ "(select H.memberNo from member M "
+ "join history H on H.roomMadeBy = :memberNo AND H.roomNo = :roomNo group by H.memberNo)";

@Query(nativeQuery = true, value = getMyCustomerQuery3)
Page<MemberEntity> getMyCustomerList3(Pageable pageable, @Param("memberNo") int memberNo, @Param("roomNo") int roomNo);
```

검색이 존재하는 경우

```
String myCustomerQuery2 = "SELECT T.* FROM member T JOIN (select H.memberNo from member M join history H on H.roomMadeBy = :memberNo group by H.memberNo) F "
+ "ON T.memberNo = F.memberNo "
+ "WHERE "
+ "T.memberId like %:keyword% OR "
+ "T.memberPhone like %:keyword% OR "
+ "T.memberGender like %:keyword% OR "
+ "T.memberName like %:keyword%";

@Query(nativeQuery = true, value = myCustomerQuery2)
Page<MemberEntity> getMyCustomerList2(Pageable pageable, @Param("memberNo") int memberNo, @Param("keyword") String keyword);
```

프로젝트 소개

개발 환경

DB 설계

컨트롤러 설계

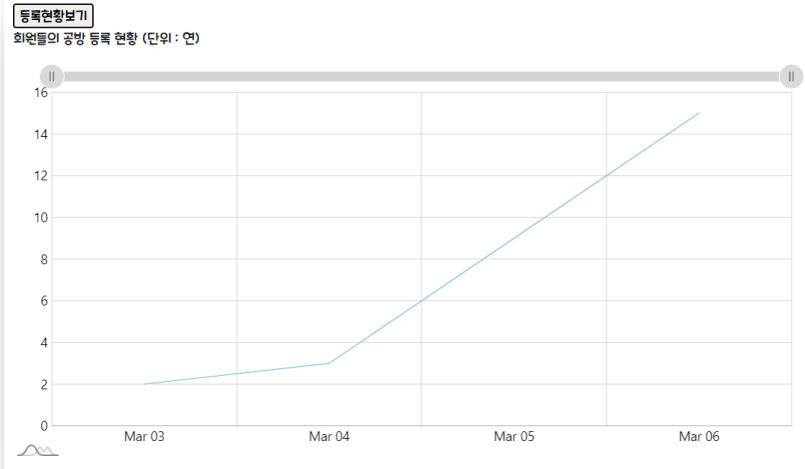
프론트 구현

주요 코드

후기

개인 채널

날짜별 예약 인원 수



amChart 를 이용하기 위해서는 날짜 순 정렬이 필수

DB 에는 예약건이 순서대로 정렬되어 있지 않음

Map<String, Integer> 이용해 중복되는 날짜에 인원 수 count++

```
// 동일한 날짜 값이 들어오면 더해서 업데이트 시켜야한다.
if (myMap.containsKey(date)) {
    myMap.put(date, myMap.get(date) + person);
} else {
    myMap.put(date, person);
}
```

json 으로 데이터 넘기기

```
for (String s : myMap.keySet()) {
    JSONObject data = new JSONObject();
    data.put("date", s);
    data.put("person", myMap.get(s));
    jsonArray.add(data);
}
```

```
var dataArray = [];

var bundleData = $(data.memberHistory).map(function(i, memberHistory){
    // 날짜 데이터 더하기
    var data = { date : 날짜, value : 등록한 인원 수 }
    dataArray.push(data);
})

am.ready(function(){
    // dataArray 이용한 차트 그리기
    series.data.setAll(dataArray);
})
```

프로젝트 소개

개발 환경

DB 설계

컨트roller 설계

프론트 구현

주요 코드

후기

관리자 페이지

입력이 발생할 수 있는 모든 케이스에 @Query 작성

등록된 모든 클래스

검색 버튼 클릭 or 지역, 카테고리 클릭하면 해당하는 클래스만 출력

클래스

종기

통계

지역선택

카테고리선택

검색

검색

등록번호	발생장소명	카테고리	강원	공방주소	승인상태	등록일자	비고	
5	통소문 잡기	핸드메이드	9명	서울 종로구 대학로 136	승인완료	02-21 12:44	상태변경	삭제
4	이발이 기초부터	음악	7명	서울 노원구 공릉로 95	승인완료	02-21 12:42	상태변경	삭제
3	토끼피 잡기 심화과정	뷰티	8명	서울 종로구 종로 20-2	승인완료	02-21 12:41	상태변경	삭제
2	피어라 잡기 기초 과정	애완동물	7명	서울 강남구 봉은사로 403	승인완료	02-21 12:41	상태변경	삭제

저음

이전

1

2

다음

마지막

검색	O	X
카테고리	O	X
지역	O	X

등록를 상태변경
등록을 상태를 변경합니다. 변경하시겠습니까?

검토중

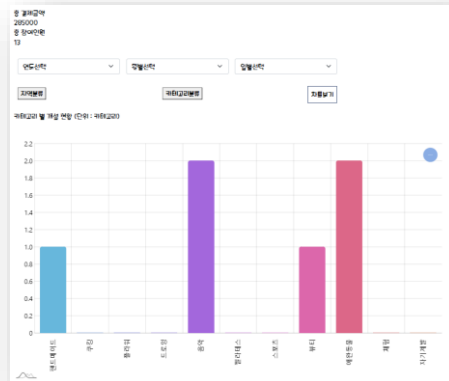
승인중

승인완료

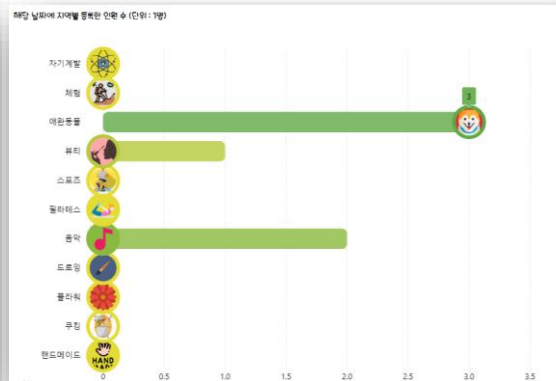
승인거부

취소

등록현황 - amchart 이용한 시각화



카테고리(& 지역) 별 개설 현황



'하루' 단위에 각 카테고리 별 등록 수



'일년' 단위로 매일 신청한 회원 수

프로젝트 소개

개발 환경

DB 설계

컨트roller 설계

프론트 구현

주요 코드

후기

검색 및 지도 출력

카테고리
지역
검색어

입력

미입력

입력된 값만 검색

'승인완료' 처리된 모든 강의

검색	O	X
카테고리	O	X
지역	O	X

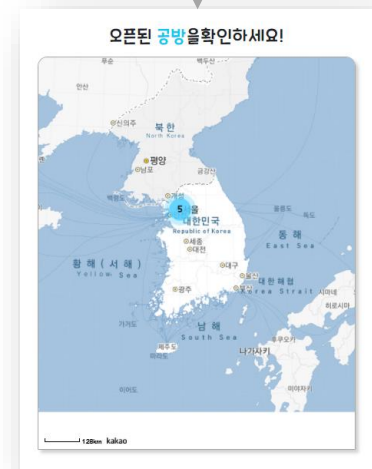
리스트 출력

입력된 값 (없으면 전체 리스트) 인수로 전달
Pageable 사용
입력된 값을 세션에 저장
front 에 model 전달



카카오 map 출력

저장된 세션값이 존재하는지 판별
없으면 전체 리스트를 출력하는 메소드
Pageable 적용 X
맵 클러스터에 전체 개설된 강의 개수 표시
있으면 특정 리스트만을 반환하는 메소드
Pageable 적용 O
한 페이지에 있는 공방에 대한 위치만 출력



시연 영상



<https://youtu.be/WyYObFmRypk>

- 프로젝트 소개
- 개발 환경
- DB 설계
- 컨트롤러 설계
- 프론트 구현
- 주요 코드
- 후기

CRUD	클래스 등록	이름, 내용, 상세 내용(summernote), 이미지 파일(Multipart), 주소(kakao map), 가격
	클래스 수정	이름, 내용, 상세 내용(summernote), 이미지 파일(Multipart), 주소(kakao map), 가격
	클래스 검색	지역, 카테고리, 검색어 존재 여부에 따른 Query 문 <div> 주소 정보 JSON 파싱 후 kakao map 에 전달 후 마커를 지도에 출력 </div>
	클래스 보기	<div> 특정 클래스와 맵핑된 Entity 호출 <div> <div>개설된 클래스 정보 Entity</div> <div>날짜, 시간 선택해서 개설한 클래스 Entity</div> <div>클래스를 개설한 회원 Entity</div> <div>예약 내역 Entity</div> </div> </div>
캘린더	캘린더 출력	javascript 이용한 캘린더 화면 출력
	DB 연동	<div> Entity → JSON parsing 후 javascript 로 전달 <div> <div>클래스</div> <div>사용자</div> </div> <div> 특정 클래스에 맵핑 되어있는 개설된 정보 전달 특정 회원에 맵핑 되어있는 정보 전달 </div> </div>

힘들었던 점

4명이 각자 다른 코드를 짜고, 매일 브랜치를 합치는 과정

javascript → java

java → javascript

네이밍 컨벤션 준수 X

중복 선언된 코드

초기 MVC 설계를 지키지 않은 코드

사이의 데이터 전달을 수월하게 사용하는 데까지 러닝 커브가 있었음 (주로 ajax)

MVC 설계가 부실해서 겪는 중간 단계의 뒤죽박죽 단계

웹 사이트 제작에 필요한 프론트 지식 → Element 에 이벤트 핸들러, 콜백 함수 부여 등

javascript, jquery 에 대해 많은 걸 알게됨

좋았던 점

막히는 부분을 팀원들과의 소통, 상호 피드백을 통해서 빠르게 해결

데이터를 프론트로 전달한 뒤 제어하는 다양한 방법을 익힘

여러가지 API 를 사용해 봤음

Notion, github 를 통한 일정 관리 및 자원 분배 과정 전반을 팀장 역할로 맡아서 진행했음

아쉬운 점

페이징 처리를 스프링부트가 제공하는 Pageable 인터페이스만 의존했다는 점 → 팀프로젝트 종료 후 개인적으로 구현 중, 페이징 처리 인터페이스 직접 구현 중

게층형 게시판 구조를 갖는 커뮤니티를 만들다가 그만뒀다는 점

AWS 배포를 http 만 해보고 https 는 실패함 → domain 을 구입해서 SSL 인증서를 등록해야 하는데 하지 못했음

→ http에서는 구글이 제공하는 geocode 가 작동하지 않음, 거기에 linux 폴더에는 이미지가 저장되지만 프론트에 출력이 안되는 문제 발생

→ 국비 교육 종료 후 배포 시도 중