

김 용 빈

Portfolio



Contents

/ 01 프로젝트 소개

/ 02 개발 환경

/ 03 시스템 프로세스

/ 04 ERD

/ 05 화면 소개



DADOK



도서 구매 및 중고 거래 플랫폼

Git :

<https://github.com/kimyongbinn/DADOK>

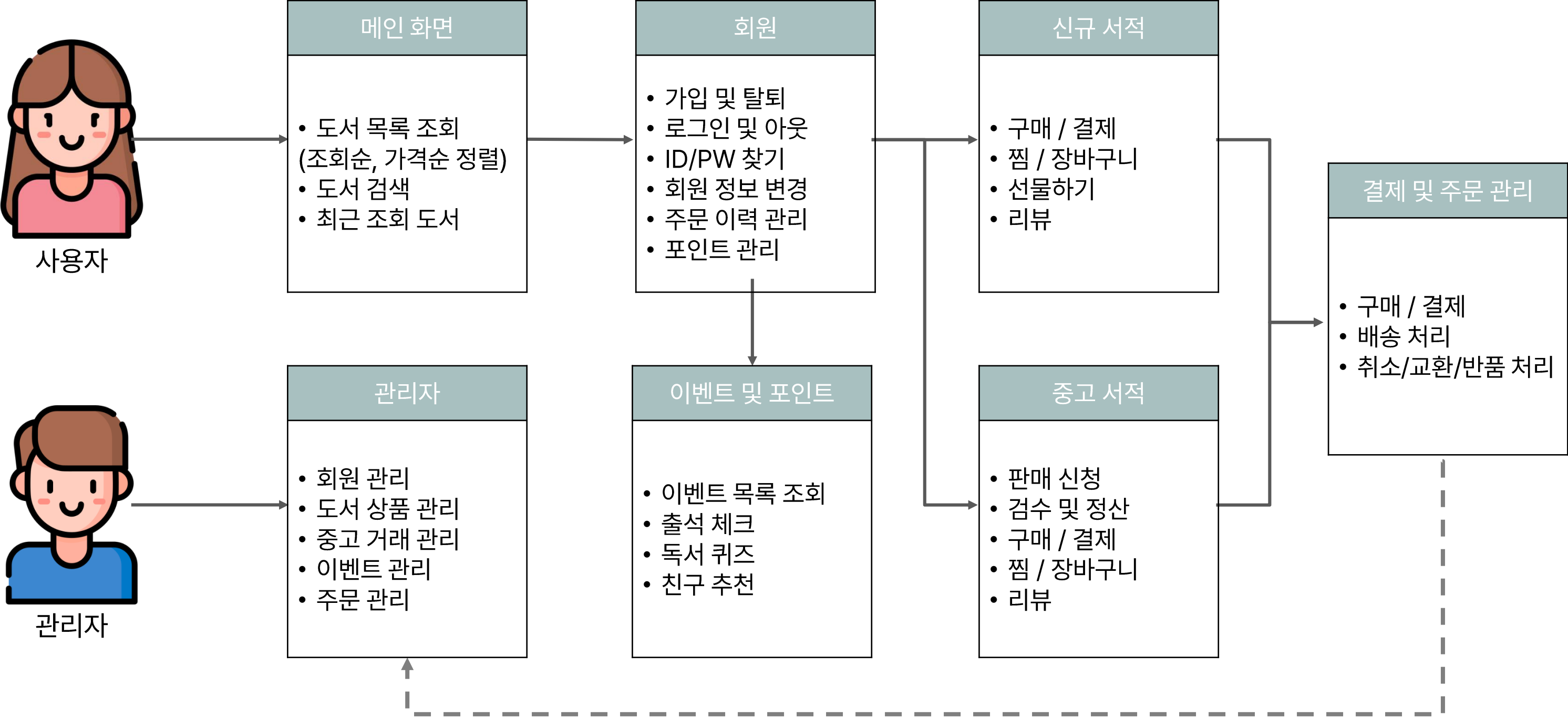
2023. 10. 27 ~

2023. 11. 29

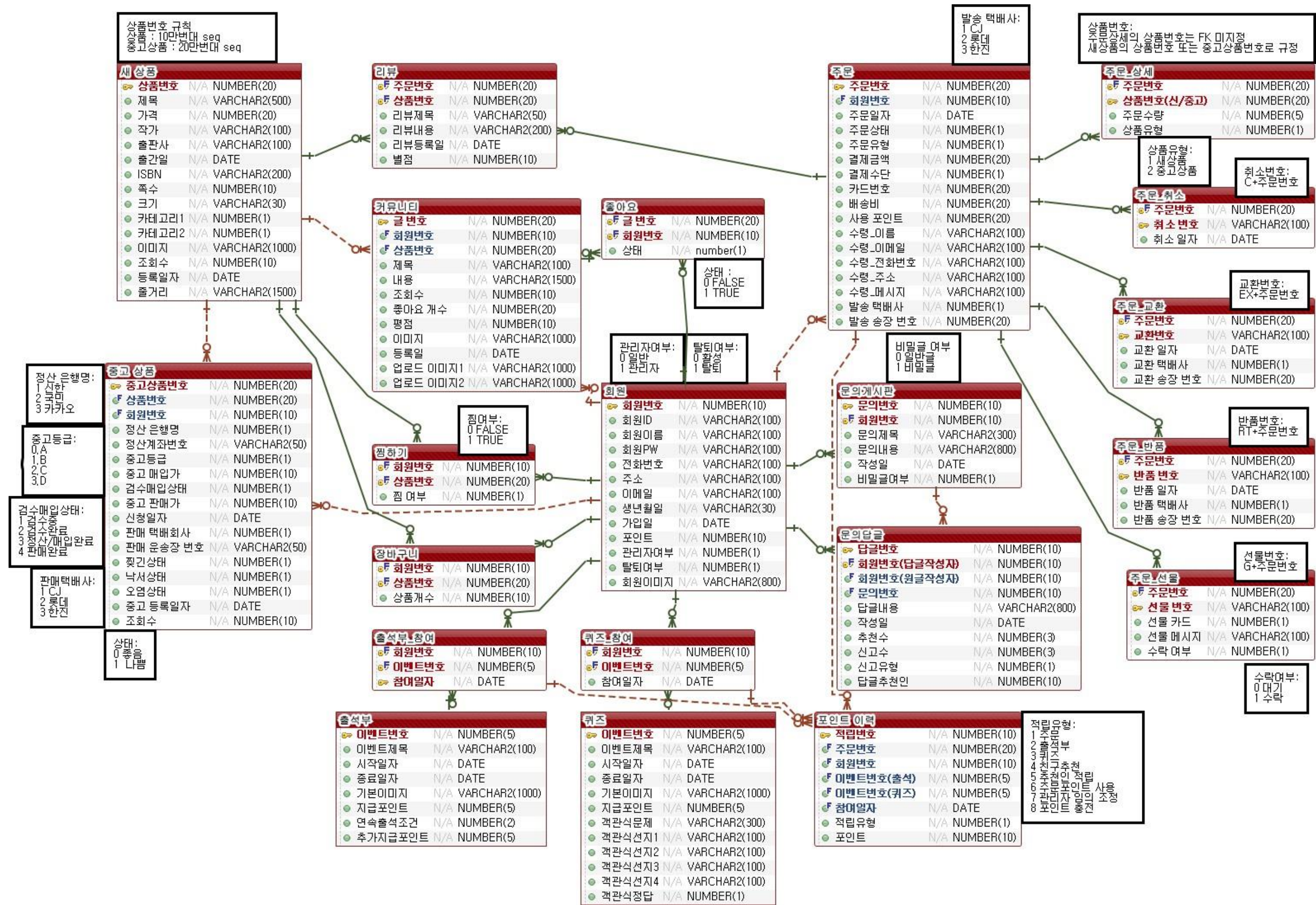


4

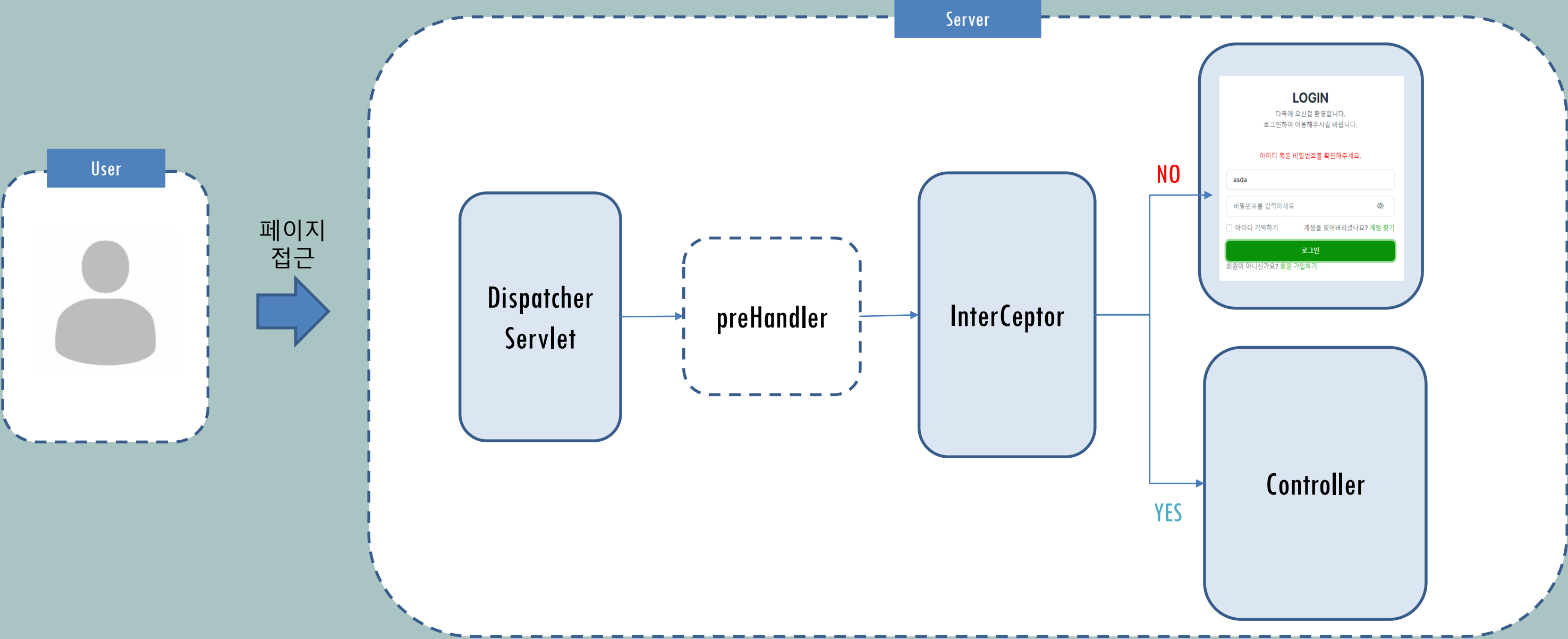




04 ERD



INTERCEPTOR



INTERCEPTOR

```
public class LoginInterCeptor implements HandlerInterceptor {

    private static final String LOGIN = "login";

    @Override
    public boolean preHandle(HttpServletRequest request, HttpServletResponse response, Object handler)
        throws Exception {
        System.out.println("LoginInterCeptor preHandle() Start...");
        // session 객체를 가져옴
        HttpSession session = request.getSession();
        // login처리를 담당하는 사용자 정보를 담고 있는 객체를 가져옴
        System.out.println("LoginInterCeptor preHandle() dest before remove => " + session.getAttribute("dest"));
        session.removeAttribute("dest");
        System.out.println("LoginInterCeptor preHandle() dest after remove => " + session.getAttribute("dest"));
        if ( session.getAttribute(LOGIN) == null ){
            // 로그인에 안되어 있는 상태임으로 로그인 폼으로 다시 돌려보냄(redirect)
            System.out.println("LoginInterCeptor Start... You do not have permission. Please log in first");
            session.removeAttribute(LOGIN);
            // 더이상 컨트롤러 요청으로 가지 않도록 false로 반환함
            return false;
        }
        return true;
    }
}
```

```
package com.choongang.s202350103.ybService;
import javax.servlet.http.HttpServletRequest;

public class ReturnPageInterCeptor implements HandlerInterceptor {

    // 로그인 페이지 이동 전, 현재 페이지를 Session에 저장
    private void saveDest(HttpServletRequest request) {
        System.out.println("saveDest Start...");
        String uri = request.getRequestURI();
        String query = request.getQueryString();
        String method = request.getMethod();

        if ("GET".equals(method)) {
            // Handle GET request
            if (query == null || query.equals("null")) {
                query = "";
            } else {
                query = "?" + query;
            }
            System.out.println("GET dest: " + (uri + query));
            request.getSession().setAttribute("dest", uri + query);
        } else if ("POST".equals(method)) {
            // Handle POST request
            System.out.println("POST dest: " + (uri));
            System.out.println("POST request detected: " + uri);
        }
    }

    @Override
    public boolean preHandle(HttpServletRequest request, HttpServletResponse response, Object handler)
        throws Exception {
        System.out.println("AfterLoginReturnPaget Prehandle Start...");
        HttpSession session = request.getSession();

        if(session.getAttribute("member") != null) {
            return true;
        }
        System.out.println("Login First!");
        saveDest(request);

        // 로그인 안한 상태면 로그인 폼으로 이동
        response.sendRedirect("/loginForm");
        return false;
    }
}
```

```
@Override
public void postHandle(HttpServletRequest request, HttpServletResponse response, Object handler,
    ModelAndView modelAndView) throws Exception {
    System.out.println("LoginInterCeptor postHandle Start...");
    HttpSession session = request.getSession();
    ModelMap modelMap = modelAndView.getModelMap();
    Object member = modelMap.get("member");
    Object dest = null;
    if(member != null) {
        System.out.println("Login success!");
        session.setAttribute(LOGIN, member);
        // 이전 페이지 불러오기
        dest = session.getAttribute("dest");

        String redirectUrl = dest != null ? dest.toString() : "/";
        System.out.println("LoginInterCeptor postHandle redirectUrl => " + redirectUrl);
        session.setAttribute("redirectUrl", redirectUrl);
        session.removeAttribute("dest");
    }
}
```

INTERCEPTOR 이용한 페이지 권한 설정

1. SESSION NULL이면 로그인 페이지 전환
NULL 아니면 Controller 실행
2. 로그인 페이지 이동 전 ->
Dest (이전 페이지 URL) session에 저장
3. 로그인 시 session에 정보 담고
이전 페이지로 이동



도서제목



찾으실 도서를 검색해보세요.

검색

메인메뉴

도서 상품 ▾ 중고판매 · 이벤트 · 커뮤니티 ▾ 고객센터 ▾

LOGIN

다독에 오신걸 환영합니다.
로그인하여 이용해주시길 바랍니다.

아이디 혹은 비밀번호를 확인해주세요.

asda

비밀번호를 입력하세요

☐ 아이디 기억하기계정을 잊어버리셨나요? [계정 찾기](#)

로그인

[회원이 아니신가요? 회원 가입하기](#)

주요 기술

1) SESSION

- 로그인 & 로그아웃

2) COOKIE

- 아이디 기억하기

3) AJAX

- 회원 정보 비교 후 로그인 성공 여부 결정



도서제목



찾으실 도서를 검색해보세요.

검색

메인메뉴

도서 상품

중고판매

이벤트

커뮤니티

고객센터

카테고리

국내도서



해외도서



중고국내도서



중고해외도서



최근 본 상품

최근 조회된 도서가 없습니다.

비밀번호 찾기

본인 명의 이메일로 인증하여 주세요

96dydqls@naver.com

가입하신 이메일을 입력해주세요.

인증번호 전송 성공! 인증번호를 입력해주세요

인증 메일발송

인증번호를 입력해주세요

인증번호 확인

요? 1544-1125 고객센터 문의

< 전체메일 1655 / 3418 안읽음 삭제

답장 전체답장 전달 삭제 스팸차단 안읽음 이동 더보기

☆ DADOK 인증번호 입니다. 📧

보낸사람 dadok202350103@gmail.com

받는사람 96dydqls@naver.com

2024년 1월 2일 (화) 오후 4:56

인증번호는 [673622]입니다.

계정 찾기 (이메일)

```
// 이메일 전송 내장 클래스
@Data
@AllArgsConstructor
class sendMailData {
    private String certNum;
    private String m_email;
}

// 이메일 전송
@SuppressWarnings({"unused"})
@ResponseBody
@RequestMapping(value = "/mailTransport")
public sendMailData mailTransport(Model model, String memberMail) throws MessagingException {
    System.out.println("YbController mailTransport start...");
    System.out.println("YbController mail memberMail -> " + memberMail);
    Member findEmail = ms.findEmail(memberMail);
    String certNum = null;

    if (findEmail != null) {
        String tomail = memberMail; // 받는 사람 email
        System.out.println("YbController mailTransport memberMail -> " + tomail);
        String setfrom = "96dydqls@gmail.com";
        String title = "DADOK 인증번호 입니다."; // 제목
        // Mime 전자우편 Internet 표준 Format
        MimeMessage message = mailSender.createMimeMessage();
        MimeMessageHelper messageHelper = new MimeMessageHelper(message, true, "UTF-8");
        messageHelper.setFrom(setfrom); // 보내는 사람 생각하거나 하면 정상작동 안함
        messageHelper.setTo(tomail); // 받는 사람 이메일
        messageHelper.setSubject(title); // 메일제목은 생략 가능
        certNum = certNum();
        messageHelper.setText("인증번호는 [ " + certNum + " ]입니다."); // 메일 내용
        System.out.println("인증번호입니다." + certNum);
        mailSender.send(message);

        model.addAttribute("memberMail", memberMail);

        return new sendMailData(certNum, memberMail);
    } else {
        return new sendMailData(certNum, memberMail);
    }
}
```

주요 기술

1) AJAX

- 페이지 이동 없이 TEXT (내장클래스 이용하여 값 전달)

2) JQuery

- 인증번호 input, button display 상태 변경

3) JavaMailSender

- JavaMail API의 클래스를 이용한 이메일 전송



카테고리

국내도서 >

해외도서 >

중고국내도서 >

중고해외도서 >

최근 본 상품

최근 조회된 도서가 없습니다.

비밀번호 찾기

본인 명의 휴대전화로 인증하여 주세요

01091304771

- 생략하고 번호만 입력해주세요

[인증번호 전송 성공! 인증번호를 입력해주세요.]

인증번호 전송

인증번호를 입력해주세요

인증번호 확인

[Web발신]
[(주)DADOK] [323807] 인증번호를
정확히 입력해주세요.

요? 1544-1125 고객센터 문의

계정 찾기 (문자)

```
// 이메일 전송 내장 클래스
@Data
@AllArgsConstructor
class sendMailData {
    private String certiNum;
    private String m_email;
}

// 이메일 전송
@SuppressWarnings("unused")
@ResponseBody
@RequestMapping(value = "/mailTransport")
public sendMailData mailTransport(Model model, String memberMail) throws MessagingException {
    System.out.println("YbController mailTransport start...");
    System.out.println("YbController mail memberMail -> " + memberMail);
    Member findEmail = ms.findEmail(memberMail);
    String certiNum = null;

    if (findEmail != null) {
        String tomail = memberMail; // 받는 사람 email
        System.out.println("YbController mailTransport memberMail -> " + tomail);
        String setfrom = "96dydqls@gmail.com";
        String title = "DADOK 인증번호입니다."; // 제목
        // Mime 전자우편 Internet 표준 Format
        MimeMessage message = mailSender.createMimeMessage();
        MimeMessageHelper messageHelper = new MimeMessageHelper(message, true, "UTF-8");
        messageHelper.setFrom(setfrom); // 보내는 사람 생략하거나 하면 정상작동 안함
        messageHelper.setTo(tomail); // 받는 사람 이메일
        messageHelper.setSubject(title); // 메일제목은 생략 가능
        certiNum = certiNum();
        messageHelper.setText("인증번호는 [ " + certiNum + " ]입니다."); // 메일 내용
        System.out.println("인증번호입니다." + certiNum);
        mailSender.send(message);

        model.addAttribute("memberMail", memberMail);

        return new sendMailData(certiNum, memberMail);
    } else {
        return new sendMailData(certiNum, memberMail);
    }
}
```

주요 기술

1) AJAX

- 페이지 이동 없이 TEXT
(내장클래스 이용하여 값 전달)

2) JQuery

- 인증번호 input, button display 상태 변경

3) 문자 API 사용

- 인증번호 문자로 전송



비밀번호 변경

비밀번호가 일치합니다.

새 비밀번호

●●●●●●●●

비밀번호 확인

●●●●●●●●



비밀번호 확인

비밀번호 저장

비밀번호 변경

```
// 이메일 전송 내장 클래스
@Data
@AllArgsConstructor
class sendMailData {
    private String certiNum;
    private String m_email;
}
// 이메일 전송
@SuppressWarnings({"unused"})
@ResponseBody
@RequestMapping(value = "/mailTransport")
public sendMailData mailTransport(Model model, String memberMail) throws MessagingException {
    System.out.println("YbController mailTransport start...");
    System.out.println("YbController mail memberMail -> " + memberMail);
    Member findEmail = ms.findEmail(memberMail);
    String certiNum = null;

    if (findEmail != null) {
        String tomail = memberMail; // 받는 사람 email
        System.out.println("YbController mailTransport memberMail -> " + tomail);
        String setfrom = "96dydqls@gmail.com";
        String title = "DADOK 인증번호 입니다."; // 제목
        // Mime 전자우편 Internet 표준 Format
        MimeMessage message = mailSender.createMimeMessage();
        MimeMessageHelper messageHelper = new MimeMessageHelper(message, true, "UTF-8");
        messageHelper.setFrom(setfrom); // 보내는 사람 생각하거나 하면 정상작동 안함
        messageHelper.setTo(tomail); // 받는 사람 이메일
        messageHelper.setSubject(title); // 메일제목은 생략 가능
        certiNum = certiNum();
        messageHelper.setText("인증번호는 [ " + certiNum + " ]입니다."); // 메일 내용
        System.out.println("인증번호입니다." + certiNum);
        mailSender.send(message);

        model.addAttribute("memberMail", memberMail);

        return new sendMailData(certiNum, memberMail);
    } else {
        return new sendMailData(certiNum, memberMail);
    }
}
```

주요 기술

1) AJAX

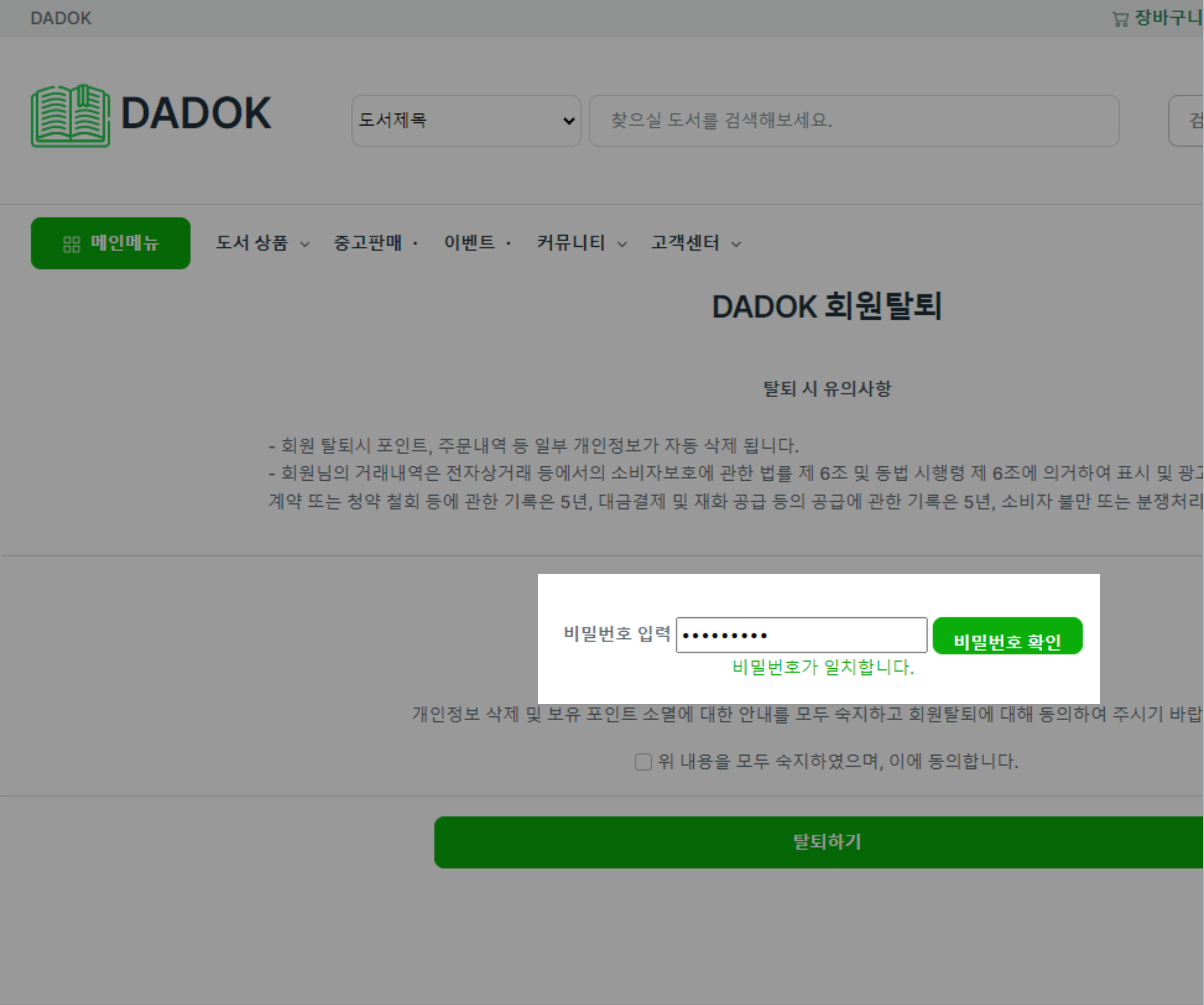
- 페이지 이동 없이 비밀번호 비교 후 result 값에 따른 TEXT (내장클래스 이용하여 값 전달)

2) JQuery

- 인증번호 input, button display block / none 상태 변경

3) MyBatis + 동적 쿼리

- 해당 목록 동적 쿼리 수행 (DB INSERT) DADOK



회원 탈퇴

```
// 이메일 전송 내장 클래스
@Data
@AllArgsConstructor
class sendMailData {
    private String certiNum;
    private String m_email;
}
// 이메일 전송
@SuppressWarnings("unused")
@ResponseBody
@RequestMapping(value = "/mailTransport")
public sendMailData mailTransport(Model model, String memberMail) throws MessagingException {
    System.out.println("YbController mailTransport start...");
    System.out.println("YbController mail memberMail -> " + memberMail);
    Member findEmail = ms.findEmail(memberMail);
    String certiNum = null;

    if (findEmail != null) {
        String tomail = memberMail; // 받는 사람 email
        System.out.println("YbController mailTransport memberMail -> " + tomail);
        String setfrom = "96dydqls@gmail.com";
        String title = "DADOK 인증번호 입니다."; // 제목
        // Mime 전자우편 Internet 표준 Format
        MimeMessage message = mailSender.createMimeMessage();
        MimeMessageHelper messageHelper = new MimeMessageHelper(message, true, "UTF-8");
        messageHelper.setFrom(setfrom); // 보내는 사람 생략하거나 하면 정상작동 안함
        messageHelper.setTo(tomail); // 받는 사람 이메일
        messageHelper.setSubject(title); // 메일제목은 생략 가능
        certiNum = certiNum();
        messageHelper.setText("인증번호는 [ " + certiNum + " ]입니다."); // 메일 내용
        System.out.println("인증번호입니다." + certiNum);
        mailSender.send(message);

        model.addAttribute("memberMail", memberMail);

        return new sendMailData(certiNum, memberMail);
    } else {
        return new sendMailData(certiNum, memberMail);
    }
}
```

주요 기술

1) AJAX

- 페이지 이동 없이 TEXT (내장클래스 이용하여 값 전달)

2) JQuery

- 인증번호 input, button disabled true / false 상태 변경

3) MyBatis + 동적 쿼리

- 해당 목록 동적 쿼리 수행 (DB INSERT)

찜 목록

honggd 님의 찜 목록입니다.

총 상품 개수 : 3

제목	출판사
 마지막으로 할 만한 멋진 일 제임스 톱트리 주니어	아작
 이제 이혼합니다 가키야 미우	문예춘추사
 보각국사 일연의 삼국유사(하) 일연	넥센미디어

포인트 목록

honggd 님의 포인트 목록입니다.

총 포인트 : 67473 P

적립 유형	이벤트명(주문번호)
출석 적립	24년 새해 출석 이벤트
퀴즈 적립	24년 새해 퀴즈 이벤트
출석 적립	24년 새해 출석 이벤트
출석 적립	24년 새해 출석 이벤트
포인트 사용	202312220001
주문 적립	202312220001

장바구니

honggd 님의 장바구니 목록입니다.

총 상품 개수 : 1

제목	수량	가격
 Probabilistic Machine Learning MIT Press 삭제하기	- 1 +	150,000원
1		

상품금액 150,000 원
배송비 0 원

결제예정금액 150,000 원

주문하기(1)

포인트 충전

중고 판매 현황

honggd 님의 중고 판매 목록입니다.

총 상품 개수 : 6

제목	상태	등급	판매 예상가	정산 금액	신청일자	정산여부 / 상세보기
 나우: 시간의 물리학 리처드 물러 바다출판사	판매완료	B 등급	11,900 원	11,900 원	2023-12-01	상세보기
 일란성 남세오 오려	매입완료	B 등급	8,400 원	8,400 원	2023-12-01	상세보기
 Probabilistic Machine Learning Murphy, Kevin P. MIT Press	매입완료	C 등급	60,000 원	60,000 원	2023-12-01	상세보기

마이페이지 목록

- 마이 페이지 >
 - 찜 목록
 - 장바구니
 - 포인트 내역
 - 중고 판매 현황

주요 기술

- MyBatis + 동적 쿼리
 - 해당 목록 동적 쿼리 수행 (DB SELECT)

DADOK



도서 구매 및 중고 거래 플랫폼

Git :

<https://github.com/kimyongbinn/DADOK>

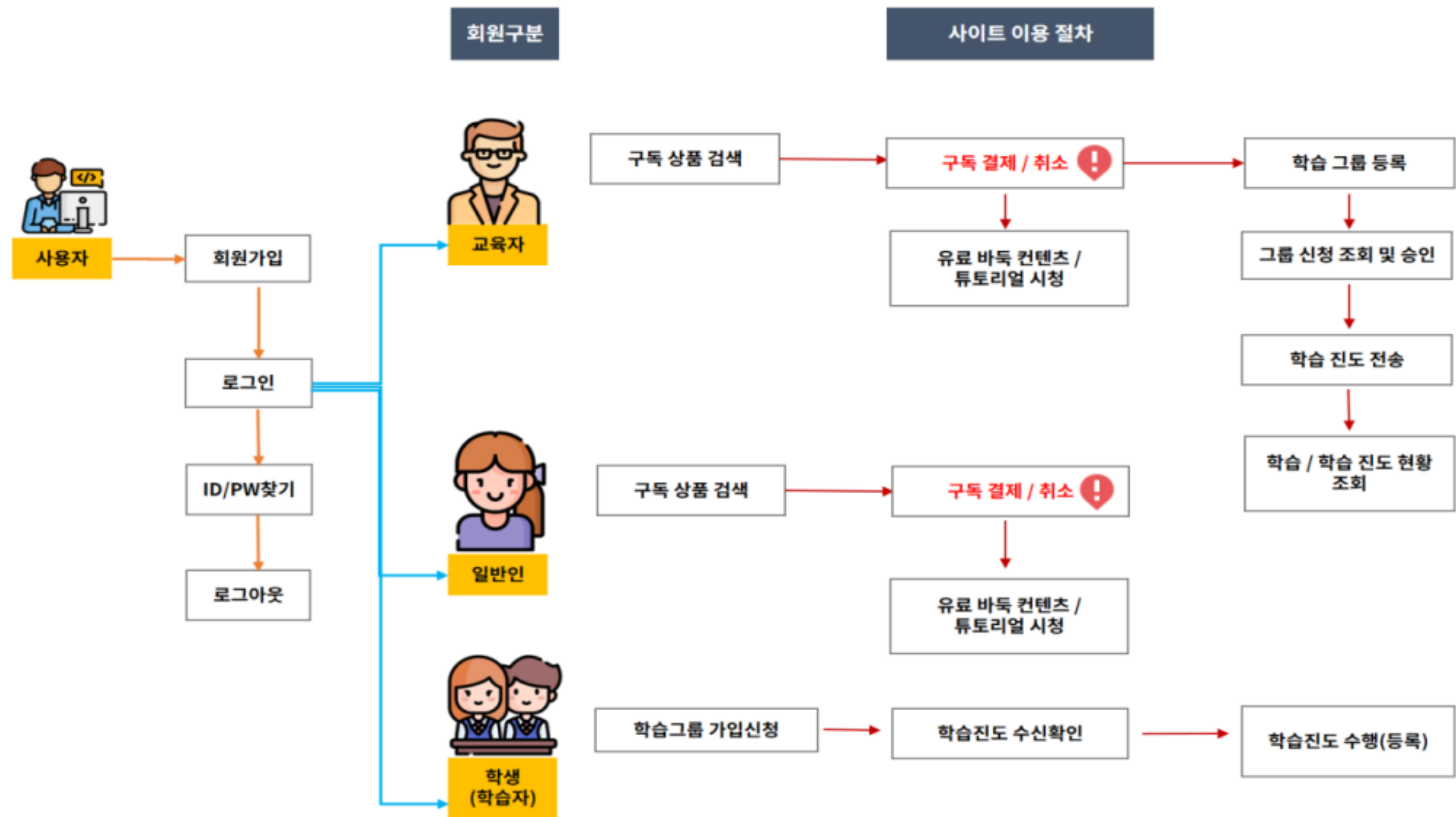
2023. 10. 27 ~

2023. 11. 29

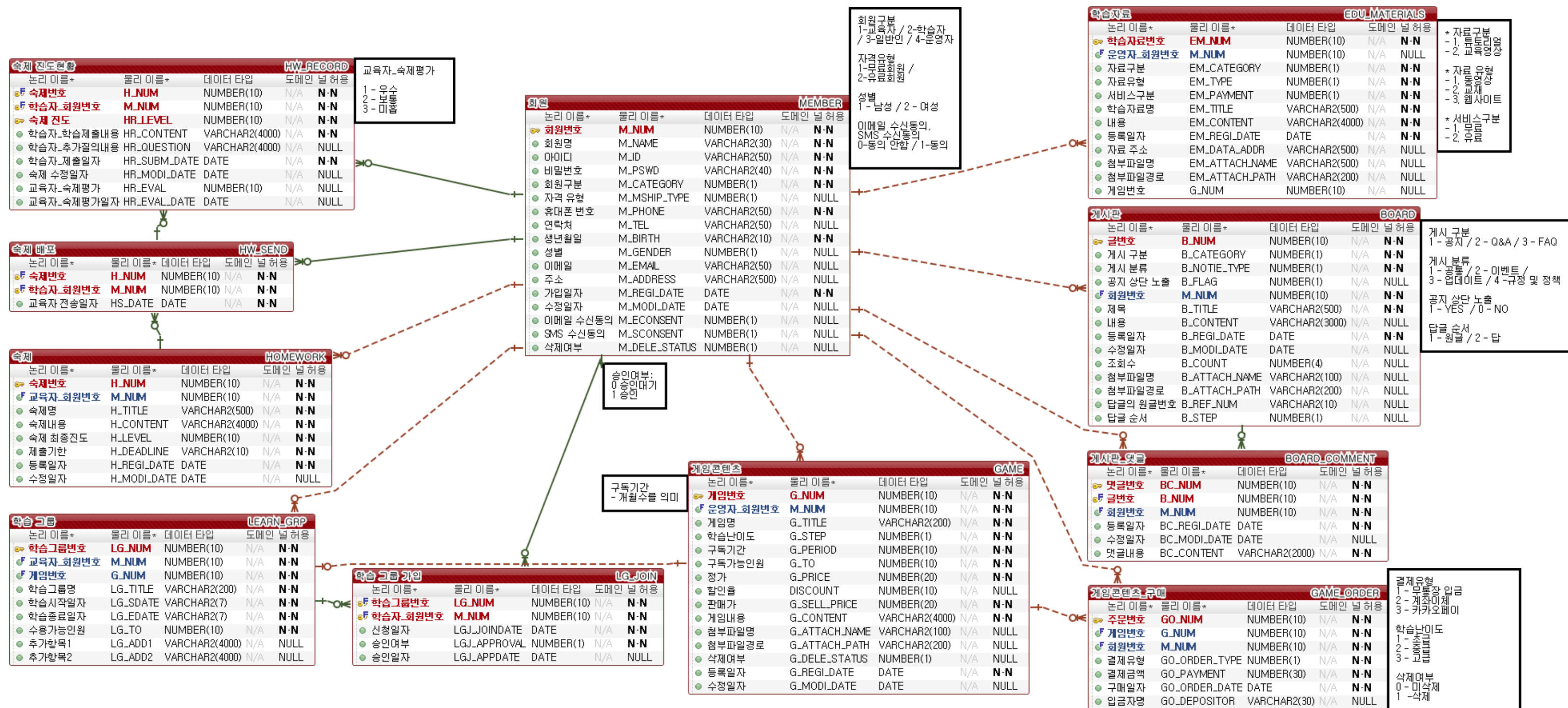


17





04 ERD



정의

- 교육자가 개설한 한 학습 그룹을 학습자가 선택하여 신청하는 화면

[Home](#) [이용안내](#) [구독서비스](#) [학습서비스](#) [고객센터](#)

학습그룹 가입신청

총 2 건

교육자명 ▾

전체보기 ▾

조회

신청

선택	콘텐츠 이미지	그룹명 / 교육자명	학습 기간	잔여 인원
<input type="radio"/>		ooo / 박현우 (전술의 바둑)	2023-12-19 ~ 2024-01-20	10 / 10
<input type="radio"/>		test1 / 이지영 (명장의 수법)	2023-12-20 ~ 2024-03-20	14 / 15

학습그룹 가입 신청

주요 기술

1) 학습그룹 조회

- 교육자가 개설한 학습 그룹 조회
- 그룹명, 교육자명 선택 시 해당 그룹 조회

2) 학습그룹 선택 신청

- 학습그룹 목록 중 1건 선택 후 신청

주요 기술

1) AJAX

- 페이지 이동 없이 TEXT
(내장클래스 이용하여 값 전달)

2) JQuery

- 인증번호 input, button disabled true / false 상태 변경

3) MyBatis + 동적 쿼리

- 해당 목록 동적 쿼리 수행 (DB INSERT)

정의

- 운영자가 등록된 학습자료 목록 조회하는 화면

한국기원

게임으로 배우는 바둑 교실

ouser1 로그아웃

Home 이용약관 구독서비스 ▾ 학습서비스 ▾ 교육자마당 ▾ 운영마당 ▾ 고객센터 ▾

학습 자료 목록

총 20 건

자료명 ▾ 🔍

등록

No.	썸네일	학습자료명	자료구분 ▾	자료유형 ▾	서비스구분 ▾	
1		바둑사활 1000제(상)11123	교육영상	웹사이트	유료	<div>상세</div>
2		Title10	교육영상	웹사이트	유료	<div>상세</div>
3		10분만에 바둑기초(바둑규칙) 마스터하기!	튜토리얼	웹사이트	무료	<div>상세</div>
4		바둑입문강의!	교육영상	동영상	유료	<div>상세</div>

↑



학습자료 등록 / 상세



주요 기능

- 1) 학습자료 조회
 - 등록 된 학습자료 목록 조회
 - 검색 및 유형에 따른 필터 가능



주요 기술

- **JPA** : 학습 자료 조회 시, 검색 및 정렬 JPQL 이용해 createQuery 실행
- **JavaScript** : 정렬 조건 value 값 가지고 페이지 이동 처리

정의

- 운영자가 등록된 학습자료 목록 등록하는 삭제하는 화면

학습 자료 등록

게임 콘텐츠	<div>미선택</div>
자료구분	<div><input type="radio"/> 튜토리얼 <input type="radio"/> 교육영상</div>
자료유형	<div><input type="radio"/> 동영상 <input type="radio"/> 교재 <input type="radio"/> 웹사이트</div>
자료구분	<div><input type="radio"/> 무료 <input type="radio"/> 유료</div>
자료명	<div></div>
내용	<div></div>
자료주소	<div></div>
썸네일	<div><div>파일 선택</div><div>선택된 파일 없음</div></div>
<div><div>등록</div><div>목록</div></div>	



학습자료 등록 / 상세

주요 기능

1) 게임콘텐츠 선택

- 게임콘텐츠 목록 중 1건 선택

2) 파일 업로드

- 해당 학습자료 관련 이미지 1개 업로드

3) 학습자료 등록

- 게임 콘텐츠 선택 & 상세 정보 입력 후 DB INSERT

주요 기술

- **MyBatis + 동적 쿼리** : 게임콘텐츠 조회 시, 검색 및 정렬 조건에 따라 동적 쿼리 수행
- **JPA** : JpaRepository 활용 -> 등록(save)
- **MultipartFile** : 스프링의 MultipartFile 인터페이스 사용하여 파일첨부 기능 구현

정의

- 운영자가 등록된 학습자료 목록 수정, 삭제하는 화면

학습 자료 상세

등록일 : 2023-12-08

게임 콘텐츠

자료구분

자료유형

자료구분

자료명

내용

자료주소

썸네일

미선택

미선택

바둑대결

돌전쟁

황제의 바둑판

승부의 국지

돌포로초

바둑의 고수

전략의 대결

바둑왕의 전쟁

바둑투표전

☒ 교육영상

☐ 교재

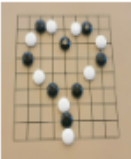
☒ 웹사이트

☒ 유료

바둑사활 1000제(상)11123

1000가지의 사활 문제를 상세하게 다룬 바둑 입문서로 초급자를 넘어 중급자로 가려는 바둑팬을 위한 정보12345611

https://www.yes24.com/Product/Goods/3956316



파일 선택

선택된 파일 없음

수정

삭제

목록

학습자료 등록 / 상세

주요 기능

1) 해당 학습자료 조회

- 선택한 학습자료 상세 데이터 조회

2) 게임콘텐츠 조회

- 해당 게임 콘텐츠 목록 조회 / 선택 or 미선택

3) 학습자료 내용 수정

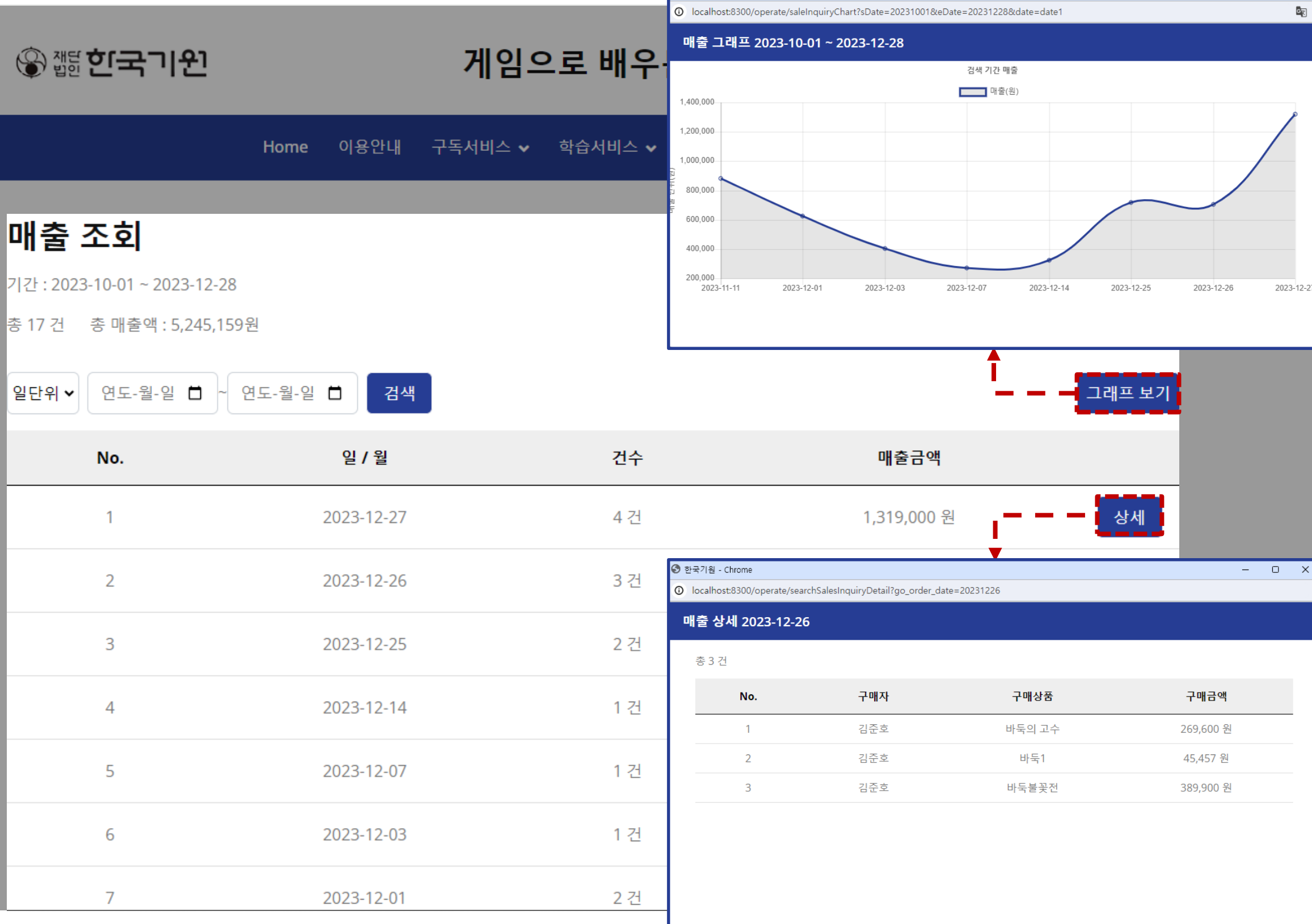
- 해당 학습자료 내용 DB UPDATE or DELETE

주요 기술

- **MyBatis + 동적 쿼리** : 게임콘텐츠 조회 시, 검색 및 정렬 조건에 따라 동적 쿼리 수행
- **JPA** : EntityManager -> 수정
JpaRepository 활용 -> 삭제
- **MultipartFile** : 스프링의 MultipartFile 인터페이스 사용하여 파일첨부 기능 구현

정의

- 일 / 월별 기간 내 해당 매출 리스트, 상세 결제 내역, 그래프 조회 화면



학습자료 등록 / 상세

주요 기능

1) 매출 조회

- 일/월별 해당 기간 내 매출 리스트 조회
- 해당 매출 상세 / 그래프 조회 기능

주요 기술

- JavaScript** : 일 / 월 별 선택 시, 해당 input 타입 display 상태 변경
- JPA** : 해당 기간 매출, 구매 상세, 그래프 Jpa 이용해 DB SELECT
- Chart.min.js** 그래프 생성 : 해당 기간 매출 리스트 중 날짜, 금액 데이터 형식 변환 후 Chart.js 이용해 그래프 생성