



인터넷 서점



INDEX

- 01 시스템 개요
- 02 팀원별 역할
- 03 개발 시 사용한 언어 및 기술
- 04 기능 정의서 내용
- 05 Table 설계도
- 06 앱 주요 화면
- 07 핵심 기능 source code 설명

▷ 프로젝트 명

온라인 서점(도서판매,관리 시스템)

▷ 프로젝트 설명

인터넷에서 사용자가 책을 주문 할 수있고 관리자가 책을 등록,수정,삭제 등 관리하는 웹을 구현한다.



02

팀원별 역할

김광수
프로젝트 팀장

김태연
개발자

박병국
개발자

김이현
디자이너

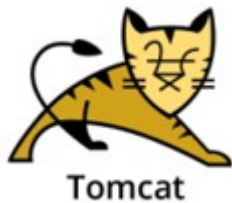
하태창
문서작성자



03

사용한 언어 및 기술

개발도구



개발언어



개발기술



04 기능 정의서

업무명	기능명	요구사항	담당자
회원	회원 가입	회원가입을 한다.	김광수
	회원 수정	회원가입할때 입력한 정보를 수정한다.	김태연
	로그인	로그인 한다.	김광수
	로그인 저장	로그인 체크를 하면 로그인창에 아이디를 저장시킨다.	김태연
	로그아웃	로그아웃을 한다.	김광수
	회원 탈퇴	회원 탈퇴를 하면, 로그인할수없다.	김태연
	회원 목록	관리자가 회원 목록을 확인한다.	김태연
	회원정보 조회	회원이나, 관리자가 회원정보를 조회한다.	김태연



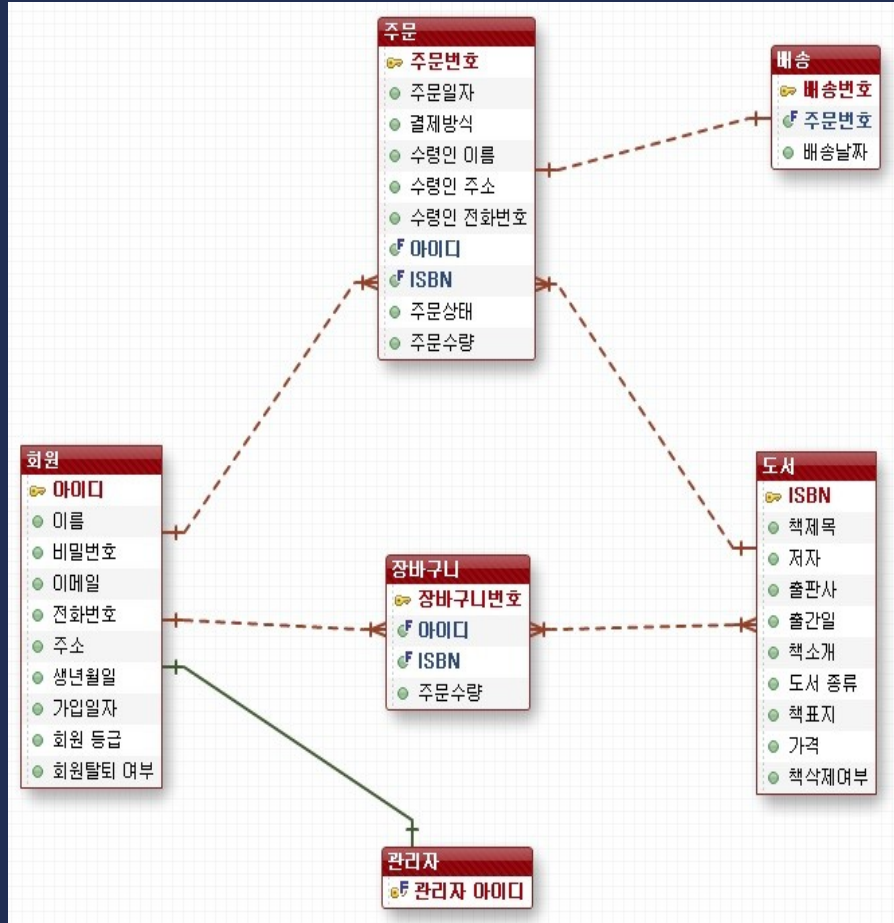
04 기능 정의서

도서	도서 목록	관리자가 도서 목록을 확인한다.	박병국
	도서정보 조회	회원이나 관리자가 도서정보를 조회한다.	박병국
	도서 추가	도서 정보를 입력하여, 도서를 추가한다.	박병국
	도서 수정	도서 정보를 수정한다.	하태창
	도서 삭제	도서 정보를 삭제한다.	박병국
	도서 검색	회원이 도서를 검색한다.	김이현
장바구니	장바구니 목록	장바구니의 목록을 표시한다.	김광수
	장바구니 추가	사용자가 구매할 책을 장바구니에 추가한다.	김광수
	장바구니 수정	장바구니의 내용의 수량을 변경한다.	김광수
	장바구니 삭제	장바구니를 삭제한다.	김이현
주문	주문상품 조회	주문할 상품을 확인한다.	김태연
	주문정보 입력	주문에 필요한 정보들을 입력한다.	김태연
	결제수단 선택	구매시 결제 방법을 선택한다.	하태창

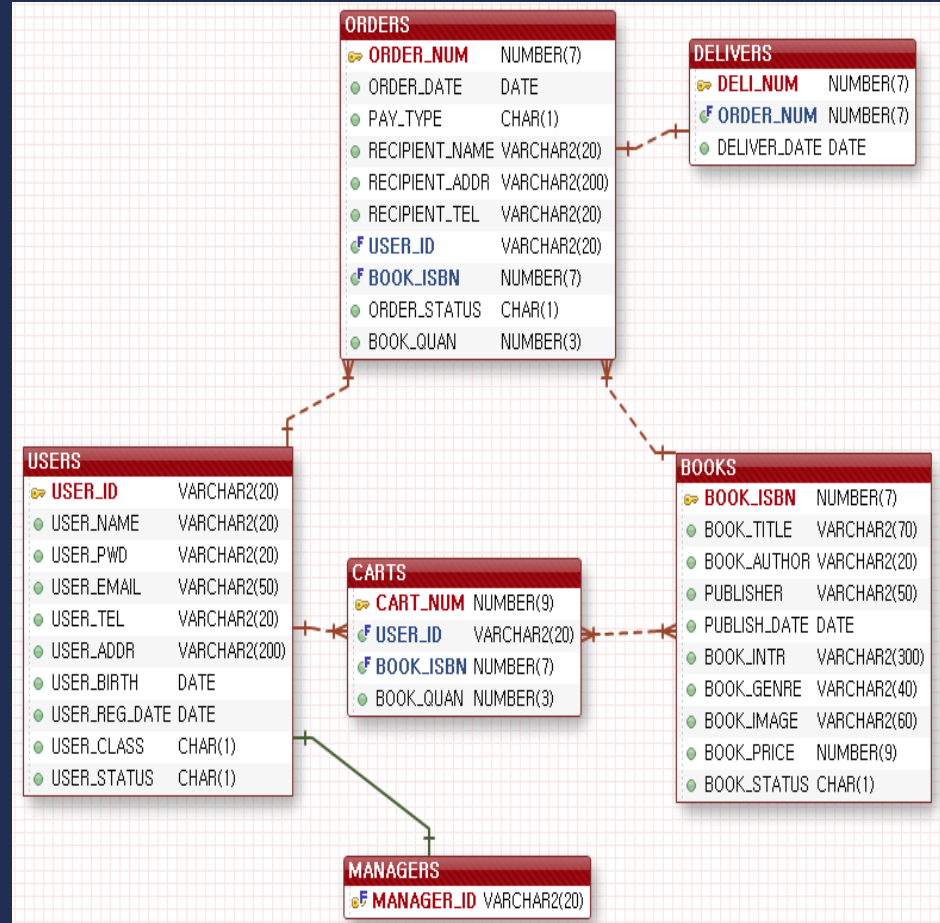


05 Table 설계도

Logical ERD



Physical ERD



06 시스템 주요 화면-사용자

회원가입

회원가입

회원가입을 위해 아래의 양식을 작성해주세요.

이름

ID

암호는 영문,특수문자,숫자를 포함해 8자 이상을 적어주세요.

암호확인

연도-월-일

E-mail

전화번호

주소

주소찾기

상세주소를 입력해주세요.

회원가입

로그인

오늘의 책

ID

암호

☐ ID 저장

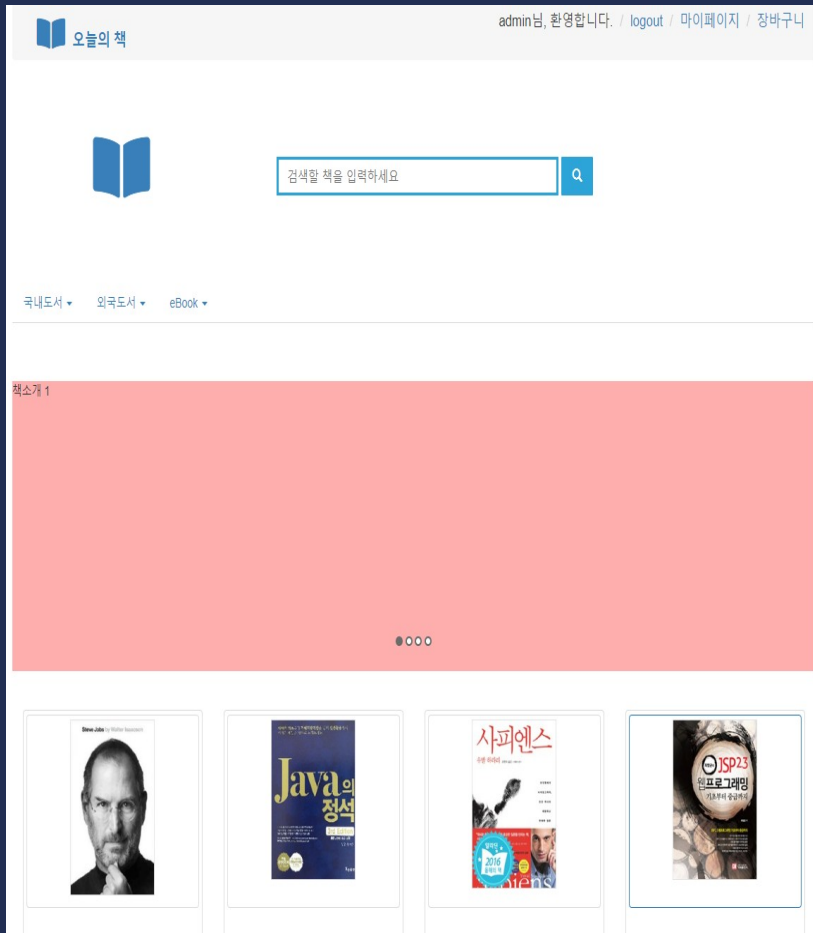
login

회원가입

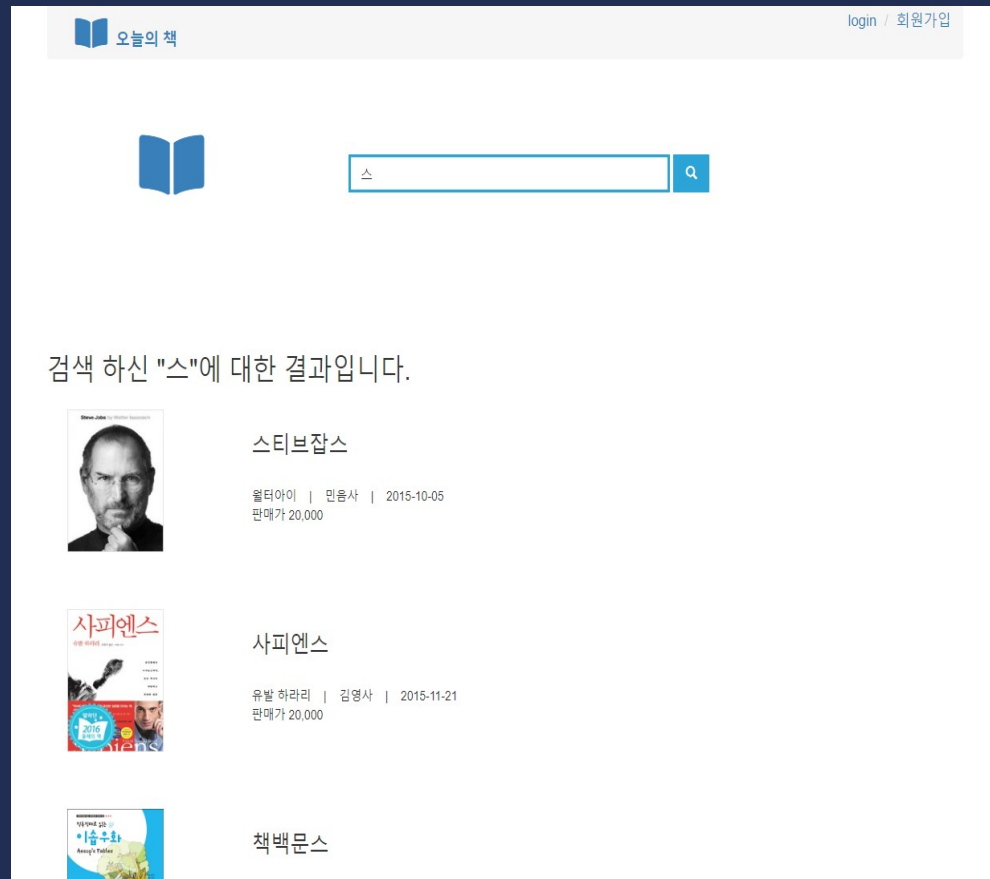


06 시스템 주요 화면-사용자

메인



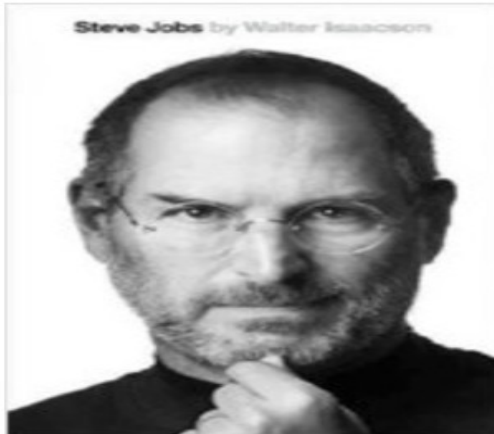
도서 검색



06

시스템 주요 화면-사용자

도서 정보



스티브잡스

월터아이 저 | 민음사 | 2015-10-05

정가 20,000원

판매중

수량 개

배송비: 무료

카트에 넣기

바로구매

품목정보

출간일	2012년 9월 21일
페이지	880쪽
ISBN13	9788960773417
ISBN10	8960773417

책소개

잡스의 사망 4주기를 맞아 그의 공식 전기 '스티브 잡스'가 가독성과 휴대성을 높인 보급판을 선보인다.



06 시스템 주요 화면-사용자

장바구니 및 주문,결제

장바구니

상품명	정가	수량	선택
<div>  <div>스티브잡스</div> </div>	20,000원	1 권	<input type="checkbox"/>

선택 주문

선택 상품 제거

주문/결제

책주문 정보

책번호	30029
책이름	스티브잡스
주문 권수	1 권
책가격	20,000원

주문자 정보

이름

연락처

배송주소

주소 찾기

결제방법

- ☒ 신용카드
- ☐ 무통장 입금
- ☐ 휴대폰 결제

결제 금액

20,000원

취소하기

결제하기



06

시스템 주요 화면-사용자

마이페이지

주문내역

주문번호	주문도서	수령인	배송상태
50019	스티브잡스	박병국	배송 중

1

주문 상세내용

주문 상세내용

현재 회원님이 주문하신 상세내용입니다.

주문번호: 50033

책제목: 트래블로그 폴란드

주문 권수: 2권

총가격: 32,000원

결제방식: 신용카드

수령인: 박병국

배송지: 미국 2층

연락처: 010-8486-1633

주문일자: 2020-05-11

배송(예상)일자: 2020-05-13

배송상태: 배송 중



06

시스템 주요 화면-사용자

회원정보

회원정보

현재 회원님의 정보입니다.

이름: 박병국

ID: user94

생년월일: 2020-05-07

E-mail: bbc926@naver.com

전화번호: 010-0101-0101

주소: 부산 해운대구 APEC로 30 대국빌딩 306호

회원정보 수정,탈퇴

회원정보 수정

회원 정보를 수정합니다 .

박병국

user94

기존패스워드.

영문/특수문자/숫자를 포함해 8자 이상을 적어주세요.

비밀번호를 한번더 입력해주세요

bbc926@naver.com

010-0101-0101

부산 해운대구 APEC로 30 대국빌딩 306호

상세주소를 입력해주세요.

수정

탈퇴



06 시스템 주요 화면-관리자

관리자 로그인

오늘의 책

ID

암호

☐ ID 저장

login

회원가입

회원목록

회원목록		
이름	회원 ID	가입일
KKS	kks01	2020-02-02
곽두팔	cqws03	2020-02-09
박병국	user94	2020-05-11
사용자	user	2020-04-27

1

2

3

4

5



회원정보

회원정보

이름: KKS

ID: kks01

생년월일: 1990-11-24

E-mail: kks01@gle.com

전화번호: 000000000001

주소: 부산시 해운대구

가입일: 2020-02-02



06 시스템 주요 화면-관리자

도서목록,삭제

도서목록

도서번호	도서명	작가명
<input type="radio"/> 10001	스티브잡스	월터아이
<input type="radio"/> 10002	자바의정석	남궁성
<input type="radio"/> 10003	사피엔스	유발 하라리
<input type="radio"/> 10004	JSP 웹	송영미

1 2 3 4 5 »



도서 정보

도서정보

도서번호: 10001

책제목: 스티브잡스

저자: 월터아이작슨

출판사: 민음사

출판일: 2015-10-05

책 소개: 스티브 잡스가 직접 참여한 유일한 공식 전기

장르: 경영

가격: 25,000원

06

시스템 주요 화면-관리자

도서 등록

도서등록

책 제목:

저자:

출판사:

출간일:

연도-월-일

책소개:

장르:

가격:

도서 이미지:

파일 선택

선택된 파일 없음

등록

취소



06 시스템 주요 화면-관리자

도서 수정

도서수정

수정할 항목을 작성해주세요.

10003

사피엔스

유발 하라리

김영사

2015-11-21

유인원에서 사이보그까지, 인간 역사의 대담하고 위대한 질문

인문학

파일 선택 선택된 파일 없음

20000

수정

취소



07

핵심 기능 source code 설명 - 장바구니 등록

```

CartService cartService = new CartServiceImpl();
List<Cart> existCarts = cartService.getCarts(userId);
boolean isExistCart = false;

if(existCarts.size()>=5){%>
<script>alert("카트에 5개까지만 담을수 있습니다.");
    history.back();</script>
<% }else{%>

<%
    for(Cart cart:existCarts) {
        if(cart.getBookIsbn()==bookIsbn) {
            totBookQuan = cart.getBookQuan() + bookQuan;
            cart.setBookQuan(totBookQuan);

            cartService.updateCart(cart);
            isExistCart = true;
            break;
        }
    }

    if(!isExistCart) {
        Cart cart = new Cart();
        cart.setBookIsbn(bookIsbn);
        cart.setUserId(userId);
        cart.setBookQuan(bookQuan);
        cartService.addCart(cart);
        bookService.updateBookStatus(bookIsbn);
    }%>

```

회원이 선택한 물품을 장바구니에 담는다. 회원이 기존에 가지고 있던 장바구니의 목록들을 카트 서비스로부터 불러와서, 해당 회원이 현재 선택한 물품과 비교한다. 장바구니에 현재 선택한 물품이 이미 존재하면 장바구니에서 해당물품의 수량을 추가하고, 새로운 물품일 경우에는, 장바구니에 물품의 항목을 새로 추가한다.



07

핵심 기능 source code 설명 - 장바구니 목록

```

<%
    session.setAttribute("cart", "notEmpty");
    request.setCharacterEncoding("utf-8");
    CartService cartService = new CartServiceImpl();
    String userId = (String)session.getAttribute("userId");
    List<Cart> carts = cartService.getCarts(userId);

    List<Book> books = new ArrayList<>();
    int totPrice = 0;

    DecimalFormat formatter = new DecimalFormat("###,###");

    for(Cart cart:carts) {
        books = cartService.getCart(cart.getCartNum()).getBooks();
        for(Book book:books) {

            <tr>
                <td>
                <td><%= formatter.format(book.getBookPrice()) %>원</td>
                <td><%= cart.getBookQuan() %> 권
                </td>
                <td><input type='checkbox' name = 'selectedBooks' value = '<%= cart.getCartNum() %>' />
            </tr>

            totPrice = totPrice + book.getBookPrice();
        }
    }
}
%>

```

회원 개개인의 장바구니를 확인할 수 있다. 회원의 아이디를 세션값으로 받아와서 해당 회원의 아이디에 대한 장바구니의 목록들만 카트서비스를 통해서 불러온다.



07

핵심 기능 source code 설명 - 장바구니 주문

```

CartService cartService = new CartServiceImpl();
List<Cart> carts = cartService.getCarts(userId);

int bookIsbn = Integer.parseInt(request.getParameter("bookIsbn"));
String recipientName = request.getParameter("recipientName");
String recipientAddr = request.getParameter("recipientAddr") + " "
    + request.getParameter("recipientAddr2");
String recipientTel = request.getParameter("recipientTel");
char payType = request.getParameter("payType").charAt(0);
int bookQuan = Integer.parseInt(request.getParameter("bookQuan"));
int booksPrice = Integer.parseInt(request.getParameter("booksPrice"));

if(cartNums.size()>0) {
    for(int cartNum:cartNums) {
        bookIsbn = cartService.getCart(cartNum).getBookIsbn();
        bookQuan = cartService.getCart(cartNum).getBookQuan();
        // book = bookService.getBook(bookIsbn);

        order.setBookIsbn(bookIsbn);
        order.setRecipientName(recipientName);
        order.setRecipientAddr(recipientAddr);
        order.setRecipientTel(recipientTel);
        order.setPayType(payType);
        order.setUserId(userId);
        order.setBookQuan(bookQuan);
        // book.setBookStatus('0');

        orderService.addOrder(order);

        cartService.deleteCart(cartNum);
        bookService.updateBookStatus(bookIsbn);
    }
}

```

회원이 장바구니에서 선택한 물품들을 주문한다. 장바구니에서 pk값을 통해 물품들을 불러올 수 있고, 그 물품들을 주문 서비스에 등록하여 주문을 진행한다. 장바구니에서는 해당 물품을 삭제하고, 도서에서는 해당물품을 주문 중인 상태로 바꾼다.



07

핵심 기능 source code 설명 - 페이징 도메인

```
package kimkwangsoo.book.page.domain;

public class Page {
    private int currentPage;
    private int rowCnt;
    private String userId;
    private String words;

    public Page() {
        this(1, 4, "default");
    }

    public Page(int currentPage) {
        this(currentPage, 4, "default");
    }

    public Page(int currentPage, int rowCnt) {
        this.currentPage = currentPage;
        this.rowCnt = rowCnt;
    }

    public Page(int currentPage, int rowCnt, String str) {
        this.currentPage = currentPage;
        this.rowCnt = rowCnt;
        this.userId = str;
        this.words = str;
    }
}
```

페이징 기능에 대한 도메인이다. 페이지 생성자에 현재 페이지 숫자, 한 페이지에 나타낼 목록의 갯수, 회원의 아이디나, 키워드를 넣어서 사용할 수 있다.



07

핵심 기능 source code 설명 - 페이징 서비스

```
public PageServiceImpl(int pageNumCnt, Page page, int pageType) {
    this.pageDao = new PageDaoImpl();
    this.usersTotRowCnt = getUsersTotRowCnt();
    this.booksTotRowCnt = getBooksTotRowCnt();
    this.pageNumCnt = pageNumCnt;
    this.page = page;

    endPage = (int)(Math.ceil(page.getCurrentPage()/(double)pageNumCnt)*pageNumCnt);
    startPage = (endPage-pageNumCnt) + 1;
    int lastEndPage = 0;

    switch(pageType) {
    case 1:
        lastEndPage = (int)(Math.ceil(usersTotRowCnt/(double)page.getRowCnt()));
        break;
    case 2:
        lastEndPage = (int)(Math.ceil(booksTotRowCnt/(double)page.getRowCnt()));
        break;
    }

    if(endPage > lastEndPage)
        endPage = lastEndPage;

    prev = startPage==1 ? false : true;
    switch(pageType) {
    case 1:
        next = endPage * page.getRowCnt() >= usersTotRowCnt ? false : true;
        break;
    case 2:
        next = endPage * page.getRowCnt() >= booksTotRowCnt ? false : true;
        break;
    }
}
```

페이지 서비스이다. 생성자에는 현재 페이지에 있는 목록의 숫자와, 페이지 객체와, 페이지타입의 번호가 들어간다. 페이지타입이 1이면 회원목록 페이지이고, 2이면 책목록 페이지이다.



07

핵심 기능 source code 설명 - 페이징_도서 갯수 선택

```
<select id="getBooksTotRowCnt" resultType="int">
    select count(book_isbn)
    from books
    where book_status!='D'
</select>
```

도서 테이블에서 도서상태가 폐기처분이 아닌 도서들의 갯수를 선택하여 불러온다.



07

핵심 기능 source code 설명 - 페이징_도서 목록 선택

```
<select id="getBooks" resultType="book" >
<![CDATA[
  select BOOK_ISBN bookIsbn, BOOK_TITLE bookTitle, BOOK_AUTHOR bookAuthor, PUBLISHER bookPublisher
  from

  (
    select rownum as rnum, x.* from (
      select * from books
      where book_status='A'
      or book_status='O'
      order by BOOK_ISBN
    )x where rownum <= #{rowCnt}*#{currentPage}
  ) where rnum > #{rowCnt}*({currentPage}-1)

]]>
</select>
```

도서 테이블에서 행의 갯수와, 현재페이지를 곱한 계산을 하여 이전 페이지와, 현재 페이지 사이에 있는 목록들만 선택하여 불러온다.



```

<%
    if (books != null && books.size() > 0) {
        // pageContext.setAttribute("books", books);
    }
%>

    <c:forEach var="book" items="${books}">
        <tbody>

            <tr
                style="overflow: hidden; white-space: nowrap; text-overflow: ellipsis;">
                <td><input type="radio" id='bookIsbn' name="bookIsbn"
                    value="${book.bookIsbn}" /> &nbsp; ${book.bookIsbn}</td>
                <td><a href="/bookInfo.jsp?bookIsbn=${book.bookIsbn}">${book.bookTitle}</a></td>
                <td>${book.bookAuthor}</td>
            </tr>
        </c:forEach>
    <% } %>
</tbody>
</table>

<div class="text-center">
    <ul class="pagination">
        <c:if test="${pageMaker.prev}">
            <li><a href="/bookList.jsp?currentPage=${pageMaker.startPage-1}">&laquo;</a></li>
        </c:if>

        <c:forEach begin="${pageMaker.startPage}" end="${pageMaker.endPage}" var="idx">
            <li <c:out value="${pageMaker.page.currentPage==idx ? 'class=active':''}" />>
                <a href="/bookList.jsp?currentPage=${idx}">${idx}</a>
            </li>
        </c:forEach>

        <c:if test="${pageMaker.next}">
            <li><a href="/bookList.jsp?currentPage=${pageMaker.endPage+1}">&raquo;</a></li>
        </c:if>
    </ul>
</div>

```

도서 목록을 확인할 수 있다. 폐기처분이 된 도서들을 제외한 나머지 모든 도서들을 출력한다. ul 클래스 pagination을 통해 페이지 버튼들을 생성한다. 한 화면에는 1부터 5까지 5개의 페이지 버튼들과, 이전 페이지 버튼, 다음 페이지 버튼이 나타난다. 이전 페이지 버튼과, 다음 페이지 버튼을 사용하여 5개씩 건너뛴 수 있다.





감사합니다

THANK YOU