

# 도서관 관리용 웹 서비스 개발

개발자 안 승진 (2023-12-11)

# 목차

1. 주요 기능 및 기술(개발 환경)
2. 개발 일정
3. 데이터베이스 구조 (ERD)
4. 주요 기능 별 구현 사항
5. 주요 소스 트리

# 소스코드 : [https://github.com/sjahnsj/library\\_websvc](https://github.com/sjahnsj/library_websvc)

# I. 주요 기능 및 기술(개발 환경)

## ❖ 주요 기능

- 회원 등록
  - 관리자 또는 일반 회원을 선택하여 등록
  - 비밀번호는 Bcrypt로 암호화하여 저장
- 회원 관리
  - 회원정보 열람 및 회원 삭제
- 도서 관리
  - 도서 목록/ 등록/ 수정 /삭제
- 도서 대출(반납)
  - 도서 목록/대출/반납/대출기한 표시

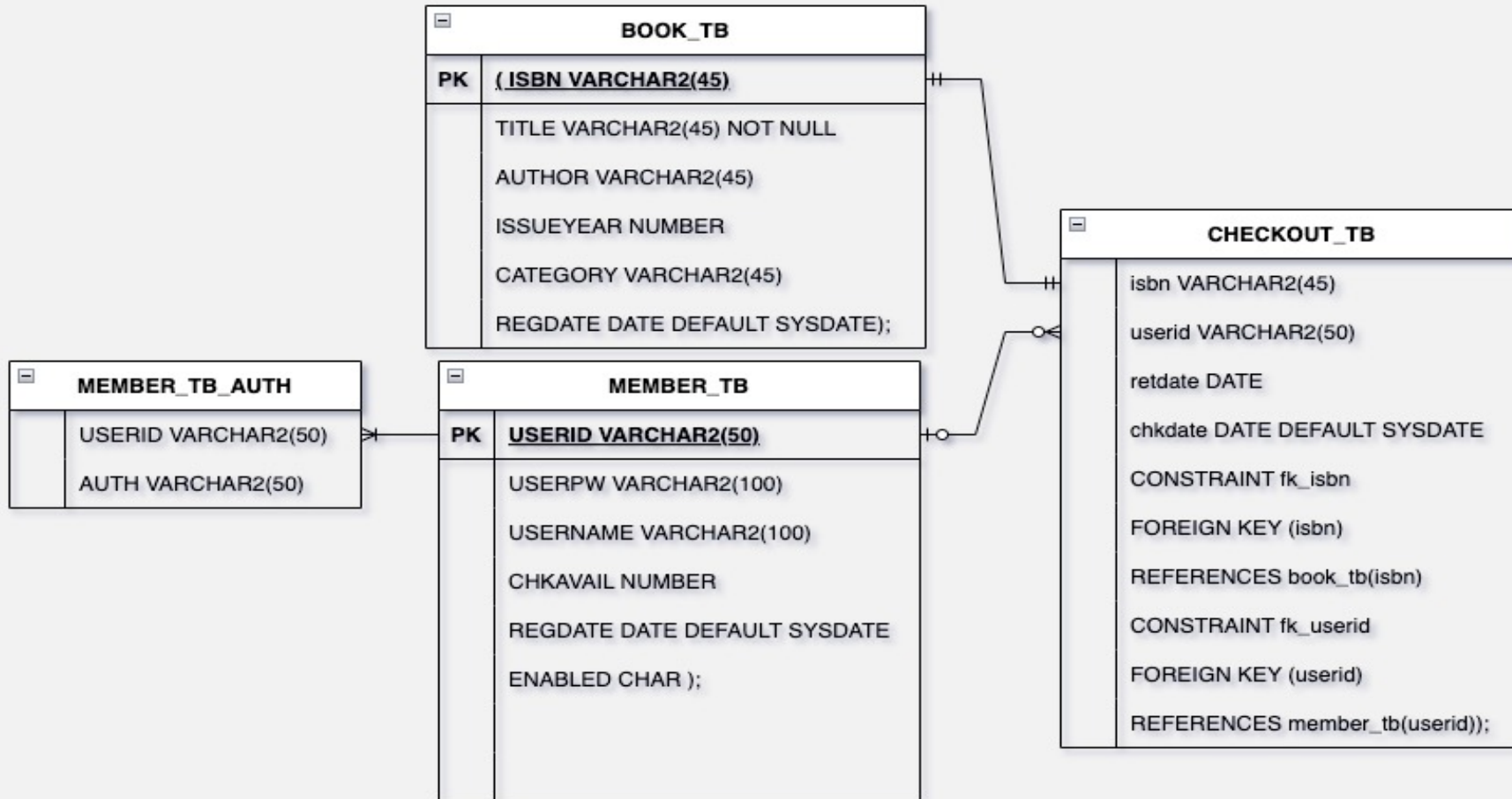
## ❖ 주요 기술 및 개발 환경

- STS3
  - Spring MVC framework, Eclipse, Tomcat, Maven
- Java
  - JDK11
- JSP
  - HTML, CSS, Javascript, JSTL
- Database
  - Oracle, SQL, Mybatis
- Docker

## 2. 개발 일정

- 개발환경 구축 및 DB 설계 : 2023-12-4
- VO/DAO/Mapper 구현 : 2023-12-5
- 회원관리 및 도서관리 구현 : 2023-12-6
- 스프링 시큐리티 기반 인증 및 인가 구현(회원등록) : 2023-12-7
- 도서대출/반납 구현 : 2023-12-8

### 3. 데이터베이스 구조 (ERD)



## 4. 주요 기능별 구현 사항

스마트 도서관에 오신걸 환영합니다

회원 등록

회원 관리

도서 대출

도서 관리



## 4.1 인증 및 인가 기능

### 로그인

취소

ID

이메일 입력

비밀번호

비밀번호 입력

로그인

```
<security:http>
  <security:intercept-url pattern="/"
    access="permitAll" />

  <security:intercept-url pattern="/book/list"
    access="hasRole('ROLE_ADMIN')" />
  <security:intercept-url pattern="/member/list"
    access="hasRole('ROLE_ADMIN')" />
  <security:intercept-url
    pattern="/book/bookform" access="hasRole('ROLE_ADMIN')" />
  <security:intercept-url pattern="/book/get"
    access="hasRole('ROLE_ADMIN')" />
  <security:intercept-url pattern="/book/add"
    access="hasRole('ROLE_ADMIN')" />
  <security:intercept-url pattern="/member/get"
    access="hasRole('ROLE_ADMIN')" />
  <security:intercept-url pattern="/member/checkout"
    access="hasRole('ROLE_MEMBER')" />

  <security:form-login login-page="/mlogin" />
  <security:logout logout-url="/mlogout"
    invalidate-session="true" />

  <security:access-denied-handler
    error-page="/custom403" />
</security:http>
```

- 스프링 시큐리티 기반으로 Filter를 통해 인증 요청할 페이지 설정
- 각 인증 페이지 별로 접근 권한 (관리자/일반회원) 별도 설정

## 4.2 회원 등록 및 사용자 인증 정보 관리

### 회원 등록

취소

ID

이메일 입력

비밀번호

비밀번호 입력

이름

이름 입력

관리자 ☐ 일반회원 ☐

등록하기

```
member.setUserpw(pwencoder.encode(member.getUserpw()));
memberDAO.regiMember(member);

if(role.equals("ROLE_ADMIN")) {
    System.out.println("admin match");
    auth.setUserid(member.getUserid());
    auth.setAuth("ROLE_ADMIN");
    authDAO.addAuth(auth);
    auth.setUserid(member.getUserid());
    auth.setAuth("ROLE_MEMBER");
    authDAO.addAuth(auth);
} else {
    System.out.println("member match");
    auth.setUserid(member.getUserid());
    auth.setAuth("ROLE_MEMBER");
    authDAO.addAuth(auth);
}
```

- 사용자 Password는 Bcrypt를 이용하여 암호화하여 저장
- 인증 후 로그인한 사용자 정보는 userdetails를 구현하여 관리한다.

```
@Override
public UserDetails loadUserByUsername(String username) throws UsernameNotFoundException {

    MemberVO vo = memberMapper.read(username);
    return vo == null ? null : new CustomUser(vo);
}
```



## 4.3 회원 관리

### 회원 목록

admin3님 로그아웃

ID(이메일)	이름	대출가능권수	등록일
member1@gmail.com	member1	2	2023-12-03
admin3@gmail.com	admin3	2	2023-12-03
admin2@gmail.com	admin2	3	2023-12-03

```
@GetMapping("/list")
public void list(Model model) {

    model.addAttribute("list", memberDAO.getList() );

}
```

- 회원목록과 회원 삭제를 위한 해당 DAO를 controller에서 호출

### 회원 정보

회원 목록

회원 삭제

ID(이메일)

admin3@gmail.com

이름

admin3

대출가능권수

2

등록일

Sun Dec 03 11:37:38 KST 2023

```
@GetMapping("/delete")
public String delete(String userid, Model model, RedirectAttributes rttr) {
    System.out.println("/delete-----");
    System.out.println(userid);
    authDAO.deleteAuth(userid);
    if (memberDAO.delete(userid)) {
        rttr.addFlashAttribute("result", "삭제");
    }
    return "redirect:/member/list";
}
```

## 4.4 도서관리(I)

### 도서 목록

[도서 등록](#)[admin3님 로그아웃](#)

제목	저자	분류	발행연도	ISBN	등록일
오리엔트 특급 살인 사건	아가사 크리스티	문학	1969	67767373	2023-11-30
셜록홈즈	코난도일	문학	1959	3242349879	2023-11-30
판다스와 시각화	백정진	자기계발	2017	67891111	2023-11-30
머신러닝	김영수	자기계발	2022	213333ANB	2023-11-30
혼자 공부하는 파이썬	윤인성	자기계발	2018	34090AA	2023-11-29
코드로 배우는 스프링	구멍가게 코딩단	자기계발	2019	A345BN234	2023-11-28
자바 웹을 다루는 기술	이건영	자기계발	2019	33332989	2023-11-28
교양있는 삶 3집	박진영	교양	2007	59365679	2023-11-27
교양있는 삶 2집	김현영	교양	1997	51365679	2023-11-27
교양있는 삶 1집	글쓰이1	교양	1981	50365679	2023-11-27
문학의 밤 3집	윤선도	문학	2021	32365679	2023-11-27
문학의 밤 2집	윤선도	문학	1990	32365678	2023-11-27
문학의 밤 1집	윤선도	문학	1989	31365678	2023-11-27
자기계발 인생2막	박명수	자기계발	2023	12365678	2023-11-27
영어독해	김성문	자기계발	1989	12355678	2023-11-27
흐르는 강물처럼	최철민	문학	2005	12345678	2023-11-27

### 도서 등록

[도서 목록](#)

제목

제목 입력

저자

저자 입력

이 입력란을 작성하세요.

발행연도

발행연도 입력

분류

자기계발

ISBN

ISBN 입력

등록하기

### 도서 정보

[도서 목록](#)[도서 수정](#)[도서 삭제](#)

제목

오리엔트 특급 살인 사건

저자

아가사 크리스티

발행연도

1969

분류

문학

ISBN

67767373

## 4.4 도서관리(2)

```
@GetMapping("/list")
public void list(Model model) {

    model.addAttribute("list", bookDAO.getList() );

}
```

- 도서목록/수정/삭제에 해당하는 DAO를 controller에서 호출

```
@PostMapping("/add")
public String add(BookVO book, RedirectAttributes rttr) {
    System.out.println("add-----");
    System.out.println(book);
    if (bookDAO.addBook(book)) {
        rttr.addFlashAttribute("result", "등록");
    }
    return "redirect:/book/list";
}

@PostMapping("/modify")
public String modify(BookVO book, RedirectAttributes rttr) {
    System.out.println("modify-----");
    System.out.println(book);
    if (bookDAO.modifyBook(book)) {
        rttr.addFlashAttribute("result", "수정");
    }
    return "redirect:/book/list";
}
```

```
@GetMapping("/delete")
public String delete(String isbn, Model model, RedirectAttributes rttr) {
    System.out.println("/delete-----");
    System.out.println(isbn);
    if (bookDAO.delete(isbn)) {
        rttr.addFlashAttribute("result", "삭제");
    }
    return "redirect:/book/list";
}
```

## 4.5 도서 대출/반납(I)

### 도서 정보

도서 목록

제목

영어독해

저자

김성문

발행연도

1989

분류

자기계발

ISBN

12355678

대출

### 도서 반납

도서 목록

ISBN	제목	대출일	반납예정일
67891111	판다스와 시각화	2023-12-11	2024-01-01
67767373	오리엔트 특급 살인 사건	2023-12-11	2024-01-01

### 도서 목록

도서 반납

member1님 로그아웃

제목	저자	분류	발행연도	ISBN	등록일	대출여부
오리엔트 특급 살인 사건	아가사 크리스티	문학	1969	67767373	2023-11-30	대출중
셜록홈즈	코난도일	문학	1959	3242349879	2023-11-30	대출가능
판다스와 시각화	백정진	자기계발	2017	67891111	2023-11-30	대출중
머신러닝	김영수	자기계발	2022	213333ANB	2023-11-30	대출가능
혼자 공부하는 파이썬	윤인성	자기계발	2018	34090AA	2023-11-29	대출가능
코드로 배우는 스프링	구명가게 코딩단	자기계발	2019	A345BN234	2023-11-28	대출가능
자바 웹을 다루는 기술	이건영	자기계발	2019	33332989	2023-11-28	대출가능
교양있는 삶 3집	박진영	교양	2007	59365679	2023-11-27	대출가능
교양있는 삶 2집	김현영	교양	1997	51365679	2023-11-27	대출가능
교양있는 삶 1집	글쓴이1	교양	1981	50365679	2023-11-27	대출가능
문학의 밤 3집	윤선도	문학	2021	32365679	2023-11-27	대출가능
문학의 밤 2집	윤선도	문학	1990	32365678	2023-11-27	대출가능
문학의 밤 1집	윤선도	문학	1989	31365678	2023-11-27	대출가능
자기계발 인생2막	박명수	자기계발	2023	12365678	2023-11-27	대출가능
영어독해	김성문	자기계발	1989	12355678	2023-11-27	대출가능
흐르는 강물처럼	최철민	문학	2005	12345678	2023-11-27	대출가능

## 4.5 대출 및 반납(2)

```
@PostMapping("/checkout")
public String checkout(BookVO book, String userid, RedirectAttributes rttr) {
    CheckoutVO chkoutVO = new CheckoutVO();
    System.out.println("checkout-----");
    System.out.println(book);
    System.out.println(userid);
    System.out.println(book.getAvail());
    MemberVO member = memberDAO.get(userid);
    System.out.println(member.getChkavail());
    if (member.getChkavail() > 0) {
        book.setAvail(book.getAvail()-1);
        bookDAO.modifyBook(book);
        memberDAO.setavail(userid, member.getChkavail()-1);
        chkoutVO.setIsbn(book.getIsbn());
        chkoutVO.setUserId(userid);
        checkoutDAO.addCheckout(chkoutVO);
        rttr.addFlashAttribute("result", "대출 처리");

        return "redirect:/member/checkout";
    } else {
        rttr.addFlashAttribute("result", "대출한도 초과");
        return "redirect:/member/checkout";
    }
}
```

- 대출시 사용자의 대출 한도 체크 후 대출 정보 추가를 위한 DAO 호출
- 반납 시 대출 정보 삭제하는 DAO호출

```
@GetMapping("/delchk")
public String delchk(String isbn, String userid, RedirectAttributes rttr) {
    MemberVO member = memberDAO.get(userid);
    BookVO book = bookDAO.get(isbn);
    memberDAO.setavail(userid, member.getChkavail()+1);
    book.setAvail(book.getAvail()+1);
    bookDAO.modifyBook(book);
    checkoutDAO.delete(isbn);
    rttr.addFlashAttribute("result", "반납 처리");
    return "redirect:/member/checkout";
}
```



## 4.6 회원/인가/도서/대출관련 VO

```
public class MemberVO {  
    String userid;  
    String userpw;  
    String username;  
    Long chkavail;  
    Date regdate;  
    boolean enabled;  
  
    List<AuthVO> authList;  
}
```

- 회원용 데이터 : MemberVO (chkavail:회원당 가능 대출권수로서 기본값 3권)
- 인가 권한 데이터 : AuthVO
- 도서 데이터 : BookVO
- 대출 데이터 : CheckoutVO (retdate: 대출 기간으로 기본값 21일)

```
public class AuthVO {  
    String userid;  
    String auth;  
}
```

```
public class BookVO {  
    String isbn;  
    String title;  
    String author;  
    Long issueyear;  
    String category;  
    Date regdate;  
    Long avail;  
}
```

```
public class CheckoutVO {  
    String isbn;  
    String userid;  
    Date retdate;  
    Date chkdate;  
}
```

## 4.7 데이터 베이스 CRUD (DAO/MAPPER)

```
public interface MemberMapper {  
    public void regiMember(MemberVO member);  
    List<MemberVO> getList();  
    MemberVO get(String userid);  
    int delete(String userid);  
    MemberVO read(String userid);  
    void setavail(String userid, Long chkavail);  
}
```

- 회원 정보 DB 관리 : MemberMapper
- 인가 정보 DB관리 : AuthMapper
- 도서 정보 DB관리 : BookMapper
- 대출 정보 DB관리 : CheckoutMapper

```
public interface AuthMapper {  
    public void addAuth(AuthVO auth);  
    public void deleteAuth(String userid);  
}
```

```
public interface BookMapper {  
    List<BookVO> getList();  
    int addBook(BookVO book);  
    BookVO get(String isbn);  
    int delete(String isbn);  
    int modifyBook(BookVO book);  
}
```

```
public interface CheckoutMapper {  
  
    void addCheckout(CheckoutVO checkout);  
    void delete(String isbn);  
    CheckoutVO getbyisbn(String isbn);  
    List<CheckoutVO> getbyuserid(String userid);  
}
```

## 5 주요 소스 트리



- Controller: /, /book, /member 요청 처리
- Domain : VO 정의
- Exception : Exception 처리
- Mapper : DB CRUD 처리
- Security : Spring security
- Service : Mapper를 호출하는 svc
- Resources : css, image 화일
- Views : /, /book, /member 처리 후 사용자에게 보여지는 jsp 화일

