《专业综合实训》
报告

题目： 图书管理系统

班级：软件 173

姓名：王旭坤

学号：201707231

指导教师：李壮

时间：2020 年 12 月

任务书

1.设计目的

计目的

书籍作为人类的精神食粮，自古以来就在人类社会占有重要的位置，现在人们对书籍的需求也越来越大。但随着书籍种类的增加，人们对书籍的管理难度越来越大。通常查找一本书籍需要花费大量的时间，这不仅效率低，工作量也很大。如今，计算机技术高速发展，促进了人类社会的发展，也解决了许多问题，图书管理系统也就随之出现。图书管理系统可以帮助人们进行快速的信息处理，提供充足的信息，把大量的人力物力从繁重，枯燥的工作中解脱出来。同时，帮助人们实现对信息的规范管理、科学统计和快速查询，从而提高图书馆的工作效率，有力保障图书馆日常事务的高效运作。

本次专业综合实训中，我设计了一个简单的图书管理系统，实现了对图书信息管理，读者信息管理，图书种类管理，以及管理员登录以及图书借阅、归还、续借等功能。本系统可以减少图书管理员进行图书信息管理、读者信息管理的工作量。本次专业综合实训，我采用了 ssm 框架进行界面以及后台代码设计，采用 sql server 数据库进行数据的存储，采用模块化的原理进行系统设计。

2.设计内容及要求

使用 ssm 框架相关的知识实现一个图书管理系统后端代码设计，并实现与前端交互，编程实现以下的功能：

（1）在 MYSQL 数据库中建立一个 library 数据库，并创建相关的表 book，reader，borrow，manager，type 实现对各种不同数据的存储。

（2）创建各种资源文件，完成资源配置。

（3）创建 mapper 文件以及实体类实现数据库映射。

（4）创建 dao 层、service 层、controller 层，实现前后端数据交互，数据库数据管理，页面跳转等功能。

3.项目分工

模块标号	模块名称	实施人员
1	管理员登陆	王旭坤
2	图书信息管理	王旭坤
3	种类管理	王旭坤
4	读者信息管理	王旭坤
5	图书借阅、归还、续借管理	王旭坤

4. 时间安排及考核方式

设计时间为 5 周。

采用答辩和报告相结合的考核方法进行考核。其中报告(不少于 5000 字)（占总成绩 50%），答辩（占总成绩 50%）。考核共分五个等级，优秀、良好、中等、及格、不及格。

1.可行性研究

（1）技术可行性

本次图书管理系统的开发是对 ssm 框架知识的一次综合性考察。我们所学的 ssm 框架知识加适当的扩展完全可以实现图书管理系统的开发。

SSM（Spring+SpringMVC+MyBatis）框架集由 Spring、MyBatis 两个开源框架整合而成（SpringMVC 是 Spring 中的部分内容），适合本次图书管理系统的开发。

本次图书管理系统的开发使用我自己的笔记本进行，该计算机安装有数据库工具（MYSQL）、应用程序开发工具（IDEA）、wps、浏览器以及相应的配置环境，满足本次图书管理系统开发的需求。

本次系统开发过程中可能会遇到一些困难，但我可以通过搜索引擎查询相关的知识以及与同学交流等途径完全可以解决遇到的问题。因此从技术方面此系统的开发可行。

（2）操作可行性

本次图书管理系统的开发采用 ssm 框架完成后台的代码设计，使用已有的前端界面完成前后端交互，使用 MYSQL 进行数据的存储和管理。开发完成后，用户可通过前台界面选择要进行的操作，在相应的文本框中输入数据，点击相应的按钮后，系统会自动执行相应的后台代码，系统完成对数据的操作以及页面的跳转等。

以上这些操作完全可以使用 ssm 框架技术完成，具有操作上的可行性。

（3）经济可行性

本次系统的开发的成本很低，主要使用自己的电脑以及相应的软件完成，经济成本很低，具有经济上的可行性。

综上，本次图书管理系统的开发可行。

2.需求分析

本系统的开发是为了解决图书管理效率低下需求，提高图书管理的效率，使用计算机技术给图书管理员带来便利，提高管理的效率。

本系统分为管理员登录，图书信息管理，读者信息管理，图书借阅、归还、续借管理，图书种类管理五个模块。其中，图书信息管理中，管理员可以登记、注销、修改、查询图书信息，简化图书管理的难度；读者信息管理中，管理员可以登记新的读者信息、注销、修改、删除读者信息操作，完成对读者信息的管理；图书借阅、归还、续借管理中，管理员帮助读者完成图书借阅、归还、续借操作，简化相应的流程，提高效率；登录功能实现对图书管理员身份的验证；图书种类管理中，管理员可以查看所有图书种类，以及增加新的图书种类。

系统功能需求分析描述如下：

（1）图书信息管理功能

前端增加图书信息页面中，用户输入图书的各种信息，点击“保存”按钮，后端将用户输入的数据存储的数据库相应的表中，并完成页面跳转；删除图书信息页面中，用户点击要删除图书对应的“删除”链接，后端代码将数据库中相应的数据删除，跳转回所有图书页面；图书信息查询页面中，用户输入图书名字的关键字，点击“查询”按钮，系统模糊查询数据库中的数据，并显示在页面中；修改图书信息页面，需用户输入所有的图书信息，点击“保存”按钮，后端代码完成对数据库中数据的修改以及页面跳转。

（2）用户登录、注册功能

用户登录功能中，用户输入用户名、密码，点击“登录”按钮后，系统会访问数据库，验证数据库中是否存在与用户输入的账号、密码相对应的信息，如果存在这样的数据，系统会转跳到主界面，等待用户操作；否则，系统重新跳转至登陆页面，让用户重新登陆。

（3）读者信息管理功能

前端增加读者信息页面中，管理员输入读者的各种信息，点击“保存”按钮，后端将用户输入的数据存储的数据库相应的表中，并完成页面跳转；删除读者信息页面中，用户点击要删除读者对应的链接，后端代码将数据库中相应的数据删除，跳转回所有读者信息页面；读者信息查询页面中，用户输入读者姓名的关键字，点击“查询”按钮，系统模糊查询数据库中的数据，并显示在页面中；修改读者信息页面，需用户输入所有需要修改的读者信息，点击“保存”按钮，后端代码完成对数据库中数据的修改以及页面跳转。

（4）图书借阅、归还、续借功能

图书借阅功能中，管理员点击图书借阅模块，进入所有图书信息页面，查询并点击要借阅图书对应的“借阅”按钮，输入读者编号，系统完成借阅信息的添加，跳转至借阅信息页面；图书归还、续借模块中，管理员查询出要修改的数据后，可点击相应的链接完成图书归还、续借，以及页面的跳转。

（5）图书种类管理功能

图书种类管理模块中，管理员可查看所有的种类信息，以及增加新的图书种类。

3.系统设计

（1）总体设计思想

本次图书管理系统的开发采用模块化的思想，将该系统划分为 5 个模块。每个模块中，都使用已有前台界面；再使用 ssm 框架技术完成系统的后端设计；最后进行一定的测试。

（2）系统模块结构图

本系统模块结构图如图 3.1 所示，本系统共分管理员登录，图书信息管理，读者信息管理，图书借阅、归还、续借、图书种类管理五个模块。

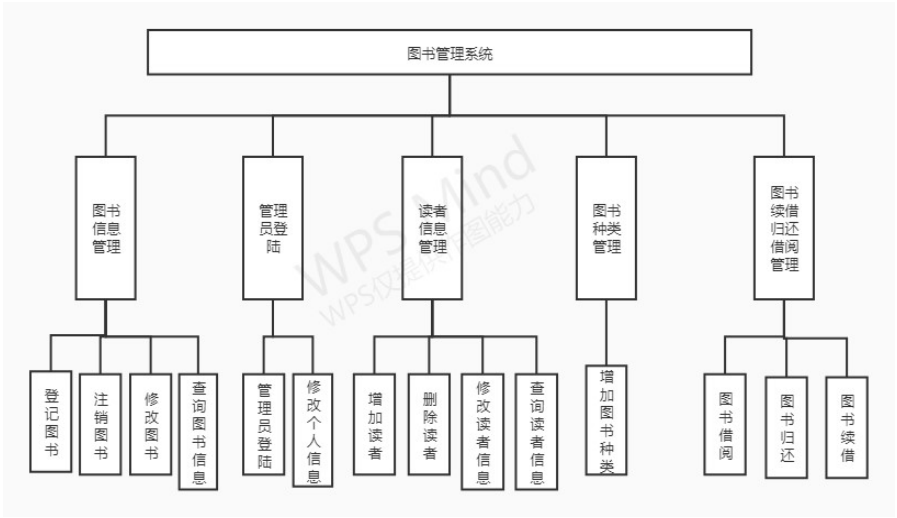

图 3.1 图书管理系统模块图

（3）各模块程序流程图

（a）系统登录模块

登录功能程序流程图如图 3.2 所示，用户填写账号、密码后，点击“登录”按钮后，在数据库中查询账号、密码是否有对应的信息，如果有则页面转跳到主页面，否则跳转回登陆页面。

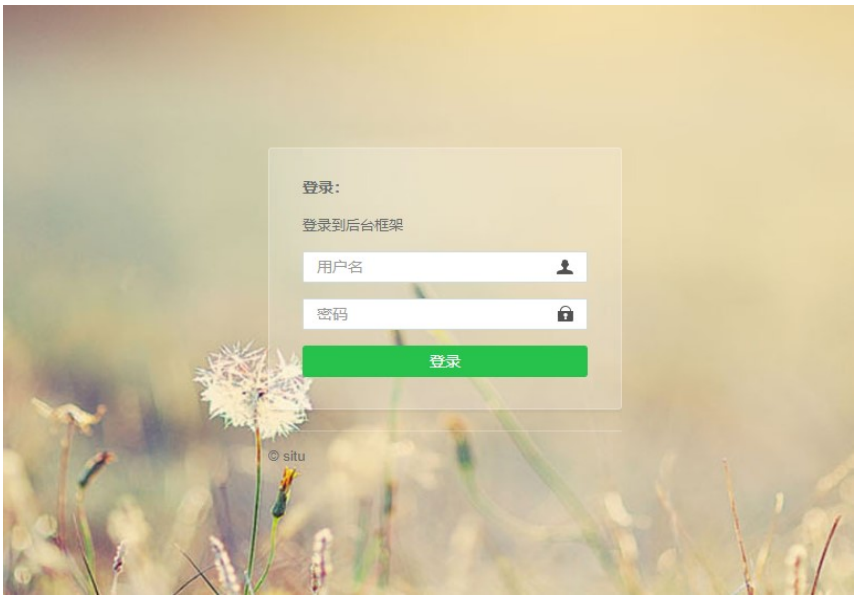

图 3.2 注册功能实现界面

（b）读者信息管理模块

管理员点击读者信息管理，系统进入全部读者信息界面，分页显示所有读者信息。

管理员点击”增加“按钮，系统进入注册读者信息界面，管理员输入读者信息，点击”保存“按钮，系统在数据库中添加对应数据，并返回全部读者信息界面。

管理员点击要修改的读者信息的”修改“链接，系统跳转进入修改读者信息界面，管理员修改读者信息以后，点击”保存“按钮，系统修改数据库中对应数据，跳转的全部读者信息界面。

删除读者信息功能中，管理员点击要删除读者信息的”删除“链接，系统删除数据库中对应的读者信息，并返回全部读者信息界面。

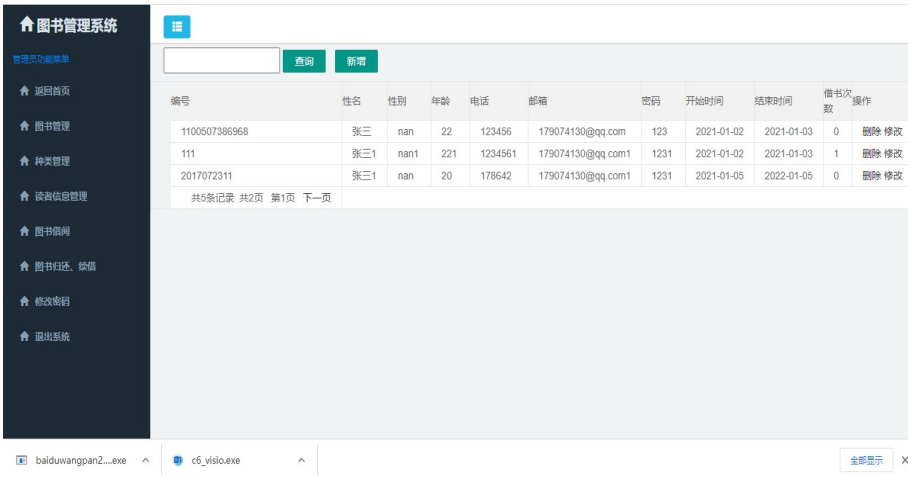

图 3.3 全部读者信息界面

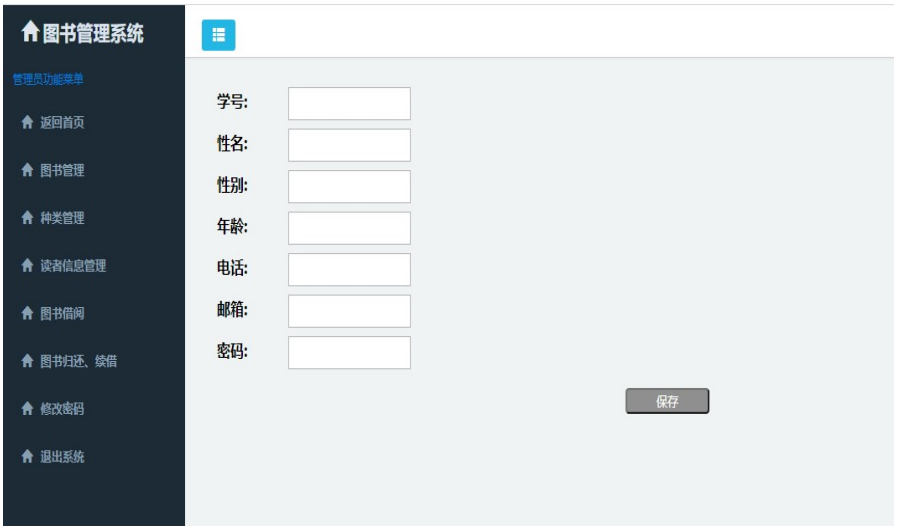

图 3.4 读者信息修改、增加界面

（c）图书信息管理模块

管理员点击图书信息管理，系统进入全部图书信息界面，分页显示所有图书信息。

管理员点击”增加“按钮，系统进入登记新图书信息界面，管理员输入图书信息，点击”保存“按钮，系统在数据库中添加对应数据，并返回全部图书信息界面。

管理员点击要修改的图书信息的”修改“链接，系统跳转进入修改图书信息界面，管理员修改图书信息以后，点击”保存“按钮，系统修改数据库中对应数据，跳转的全部图书信息界面。

删除图书信息功能中，管理员点击要删除图书信息的”删除“链接，系统删除数据库中对应的图书信息，并返回全部图书信息界面。

图 3.5 全部图书信息界面

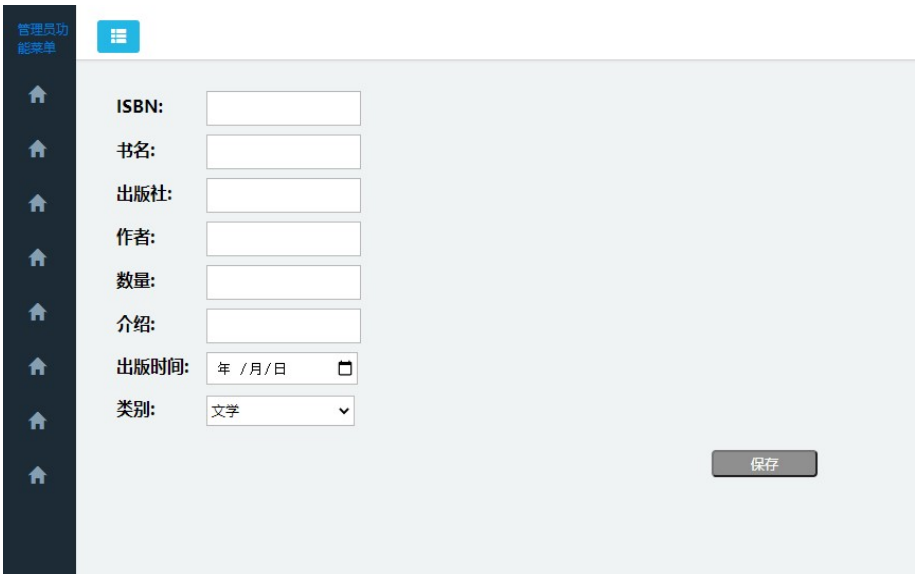

图 3.6 图书信息修改、增加界面

（d）图书借阅、归还、续借模块

图书借阅模块中，管理员查询到要借阅的图书以后，点击对应的”借阅“链接，在弹出的窗口中输入要借阅该图书的读者编号，系统在数据库的借阅表中添加相应的信息，转跳到全部借阅信息页面。

图书归还、续借模块中，管理员在文本框中输入读者编号，查询读者所有的借阅信息，找到要进行操作的信息。管理员点击”续借“链接，系统增加读者本次借阅图书的持有时间；管理员点击”归还“链接，系统设置对应图书借阅的状态为”已归还“，并修改数据库中相应的信息。

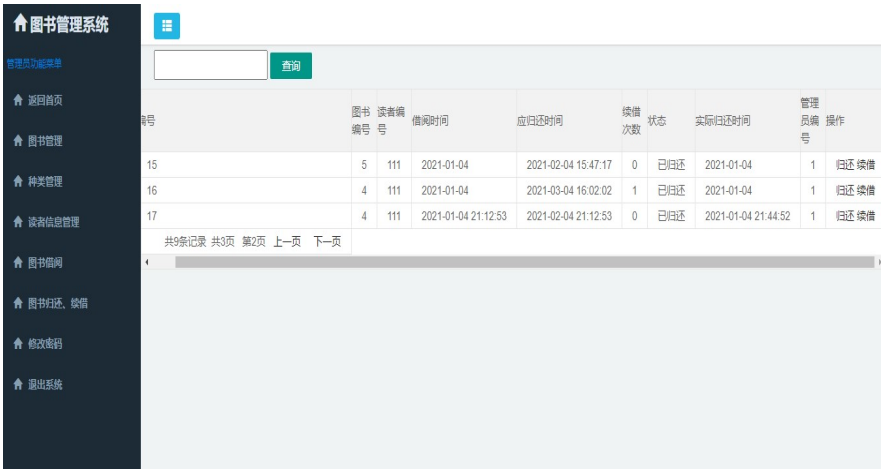

图 3.7 图书归还页面

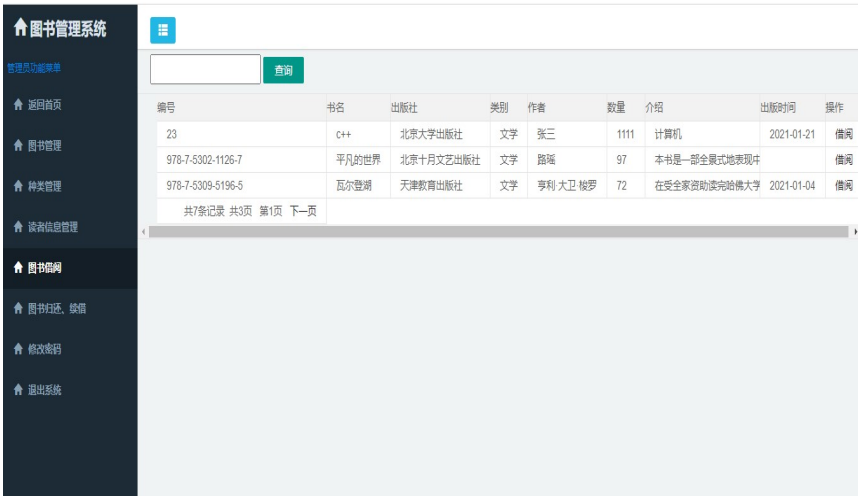

图 3.8 图书借阅、续借页面

（e）图书种类管理模块

在此模块中，管理员点击种类管理可查看所有的图书种类。管理员在文本框中输入新图书种类，点击”增加“按钮，即可完成新图书种类的增加。

图 3.9 图书种类管理页面

（f）数据库设计

本次图书管理系统需要设计图书信息表、读者信息表、种类信息表、管理员信息表、图书借阅表存储相应数据。

4. 实现与测试

（1）管理员登录模块

后台关键代码：

```
@Controller
public class ManagerController {

    @Autowired
    ManagerService managerService;

    @RequestMapping("/manager")
    public String toLogin(@ModelAttribute("manager")Manager manager){
        return "/manager/login";
    }
```

```
    @RequestMapping("/manager/login")
    public String login(@ModelAttribute("manager") Manager manager,
HttpSession session){
        Boolean rst=managerService.isLogin(manager);
        if(rst){
            Manager manager1=managerService.findManager(manager);
            session.setAttribute("manager",manager1);
            return "/manager/index";
        }else {
            return "/manager/login";
        }
    }

    @RequestMapping("/manager/toUpdatePass")
    public String toUpdatePass(){
        return "/manager/pass";
    }

    @RequestMapping("/manager/updatePass")
    public String updatePass(String pass, String repass, Manager manager,
HttpSession session, HttpServletRequest request){
        if(pass.equals(repass)){
            Manager m=(Manager)session.getAttribute("manager") ;
            manager.setId(m.getId());
            manager.setPassword(pass);

            managerService.updatePass(manager);
            request.setAttribute("state","sucess");
            return "forward:/manager/toUpdatePass";
        }else {
            request.setAttribute("state","f");
            return "forward:/manager/toUpdatePass";
        }
    }

    @RequestMapping("/manager/outlogin")
```

```
    public String outlogin(HttpSession session){
        session.invalidate();
        return "/manager/login";
    }

    @RequestMapping("/manager/toUpdateManager")
    public String toUpdateManager(Model model,HttpSession session){
        Manager manager=(Manager) session.getAttribute("manager");
        Manager
newManager=managerService.findManagerById(manager.getId());

        model.addAttribute("manager",newManager);
        session.setAttribute("manager",newManager);
        return "/manager/manageredit";
    }

    @RequestMapping("/manager/updateManager")
    public       String       updateManager(@ModelAttribute       Manager
manager,HttpServletRequest request){
        managerService.updateManager(manager);

        request.setAttribute("state","s");
        return "forward:/manager/toUpdateManager";
    }
}
```

（2）读者信息管理模块

后台关键代码：

```
@Controller
@RequestMapping("/manager")
public class ReaderController {

    @Autowired
    ReaderService readerService;

    DataFormat dataFormat=new DataFormat();
```

```
    @RequestMapping("/allReaders")
    public String findAllReaders(Model model,Integer pageCur,HttpSession
session){
        if (pageCur == null) {
            pageCur = 1;
        }

        int pageRow = 3;
        List<Reader>     readerList=readerService.findAllReaders(pageCur,
pageRow);
        Integer totalCount = readerService.findReadersTotalCount();
        Integer totalPage = (totalCount+pageRow-1)/pageRow;
        model.addAttribute("totalCount", totalCount);
        model.addAttribute("totalPage", totalPage);
        model.addAttribute("pageCur", pageCur);
        model.addAttribute("readers",readerList);

        session.setAttribute("select","all");
        return "/manager/readermain";
    }

    @RequestMapping("/toAddReader")
    public String toAddReader(HttpSession session){
        session.setAttribute("info","insert");
        return "/manager/readeredit";
    }

    @InitBinder("reader")
    public void init(WebDataBinder binder) {
        binder.setFieldDefaultPrefix("reader.");
    }
    @RequestMapping("/addReader")
    public String addReader(@ModelAttribute Reader reader,String cmd){
        if(cmd.equals("insert")){
            reader.setStart(dataFormat.currentTime());
            reader.setEnd(dataFormat.lastTime());
            reader.setPunishcount(0);//未实现惩罚功能
```

```
            reader.setLendcount(0);
            readerService.addReader(reader);
        }else if(cmd.equals("update")){
            readerService.updateReader(reader);
        }
        return "forward:/manager/allReaders";
    }

    @RequestMapping("/toUpdateReader")
    public String toUpdateReader(Model model, HttpSession session,String
id){
        Reader reader=readerService.findReaderById(id);
        model.addAttribute("reader",reader);
        session.setAttribute("info","update");
        return "/manager/readeredit";
    }

    @RequestMapping("/deleteReader")
    public String deleteReader(String id){
        readerService.deleteReaderById(id);
        return "forward:/manager/allReaders";
    }

    @RequestMapping("/findReadersByTag")
    public String findReadersByTag(String name,Model model,HttpSession
session){
        List<Reader> readerList=readerService.findReadersByTag(name);
        model.addAttribute("readers",readerList);

        session.setAttribute("select","tag");
        return "/manager/readermain";
    }
}
```

（3）图书信息管理模块

后台关键代码：

```
@Controller
@RequestMapping("/manager")
```

```
public class BookController {

    @Autowired
    BookService bookService;

    @Autowired
    TypeService typeService;

    @RequestMapping("/allBooks")
    public String findAllBooks(Model model,Integer pageCur,HttpSession
session){
        if (pageCur == null) {
            pageCur = 1;
        }

        int pageRow = 3;
        List<Book> bookList=bookService.findAllBooks(pageCur, pageRow);
        Integer totalCount = bookService.findBooksTotalCount();
        Integer totalPage=(totalCount+pageRow-1)/pageRow; //= totalCount /
pageRow + 1;
        model.addAttribute("totalCount", totalCount);
        model.addAttribute("totalPage", totalPage);
        model.addAttribute("pageCur", pageCur);
        model.addAttribute("allBooks",bookList);

        session.setAttribute("selectAct","all");
        return "/manager/bookmain";
    }

    @RequestMapping("/toAddBook")
    public  String toAddBook(Model model,HttpSession session){
        List<Type> typeList=typeService.findAllGoodsType();
        model.addAttribute("allTypes",typeList);
        session.setAttribute("info","insert");
        return "/manager/bookedit";
    }
```

```
    @InitBinder("book")
    public void init(WebDataBinder binder) {
        binder.setFieldDefaultPrefix("book.");
    }
    @RequestMapping("/addBook")
    public String addBook(@ModelAttribute("book") Book book,Integer
tid,String cmd){
        Type type=typeService.findTypeById(tid);
        book.setType(type);
        if(cmd.equals("insert")) {
            bookService.addBook(book);
        }else if(cmd.equals("update")){
            bookService.updateBook(book);
        }
        return "forward:/manager/allBooks";
    }

    @RequestMapping("/toUpdateBook")
    public String updateBook(Model model, Integer id, HttpSession session){
        Book book=bookService.findBookByISBN(id);
        List<Type> typeList=typeService.findAllGoodsType();
        model.addAttribute("allTypes",typeList);
        model.addAttribute("oneBook",book);
        session.setAttribute("info","update");
        return "/manager/bookedit";
    }

    @RequestMapping("/deleteBook")
    public String deleteBook(Integer id){
        bookService.deleteBookById(id);
        return "forward:/manager/allBooks";
    }

    @RequestMapping("/findBookByTag")
    public String findBookByTag(String name,Model model,Integer
pageCur,HttpSession session){//取消分页功能
        if (pageCur == null) {
```

```
            pageCur = 1;
        }

        int pageRow = 3;
        Integer totalCount = bookService.findGoodsByTagCount(name);
        List<Book>      bookList=bookService.findGoodsByTag(name,pageCur,
pageRow);
        Integer totalPage=(totalCount+pageRow-1)/pageRow; //= totalCount /
pageRow + 1;
        model.addAttribute("totalCount", totalCount);
        model.addAttribute("totalPage", totalPage);
        model.addAttribute("pageCur", pageCur);
        model.addAttribute("allBooks",bookList);

        session.setAttribute("selectAct","tag");
        return "/manager/bookmain";
    }
}
```

（4）图书借阅、归还、续借模块

后台关键代码：

```
@Controller
@RequestMapping("/manager")
public class BorrowController {

    @Autowired
    ReaderService readerService;

    @Autowired
    BorrowService borrowService;

    @Autowired
    BookService bookService;

    DataFormat dataFormat=new DataFormat();

    @RequestMapping("/getAllBooks")
```

```
    public String toLendBook(Model model,Integer pageCur){
        if (pageCur == null) {
            pageCur = 1;
        }

        int pageRow = 3;
        List<Book> bookList=bookService.findAllBooks(pageCur, pageRow);
        Integer totalCount = bookService.findBooksTotalCount();
        Integer totalPage=(totalCount+pageRow-1)/pageRow; //= totalCount /
pageRow + 1;
        model.addAttribute("totalCount", totalCount);
        model.addAttribute("totalPage", totalPage);
        model.addAttribute("pageCur", pageCur);
        model.addAttribute("allBooks",bookList);
        return "/manager/borrowmain";
    }

    @RequestMapping("/lendBook")
    public String lendBook(Borrow borrow, String bid, String rid,
HttpSession session){
        Manager manager=(Manager)session.getAttribute("manager") ;
        Integer mid=manager.getId();
        borrow.setBid(bid);
        borrow.setRid(rid);
        borrow.setBorrowtime(dataFormat.currentTimeS());
        borrow.setReturntime(dataFormat.returnTime());
        borrow.setState("未归还");
        borrow.setRenewtimes(0);
        borrow.setMid(mid);

        readerService.updateLendCount(rid);//读者借阅次数加一
        bookService.updateBookNum(Integer.valueOf(bid));//图书数量减一
        borrowService.addBorrow(borrow);//借阅表信息增加
        return "forward:/manager/toReturnBook";
    }

    @RequestMapping("/toReturnBook")
```

```
    public String toReturnBook(Model model,Integer pageCur,HttpSession
session){
        if (pageCur == null) {
            pageCur = 1;
        }

        int pageRow = 3;
        List<Borrow>      borrowList=borrowService.findAllBorrow(pageCur,
pageRow);
        Integer totalCount = borrowService.findBorrowsTotalCount();
        Integer totalPage=(totalCount+pageRow-1)/pageRow; //= totalCount /
pageRow + 1;
        model.addAttribute("totalCount", totalCount);
        model.addAttribute("totalPage", totalPage);
        model.addAttribute("pageCur", pageCur);
        model.addAttribute("allBorrow",borrowList);

        session.setAttribute("s","all");
        return "/manager/returnmain";
    }

    @RequestMapping("/returnBook")
    public String returnBook(Borrow borrow,Integer id){
        borrow.setState("已归还");
        borrow.setPracticaltime(dataFormat.currentTimeS());
        borrow.setId(id);

        borrowService.returnBorrow(borrow);
        Integer bid=borrowService.findBookId(id);
        if(bid>0) {
            bookService.addBookNum(bid);
        }
        return "forward:/manager/toReturnBook";
    }

    @RequestMapping("/renewBook")
    public String renewBook(Borrow borrow,Integer id){
```

```
        String time=borrowService.findReturnTime(id);
        borrow.setReturntime(dataFormat.addNewMonth(time));
        borrow.setId(id);

        borrowService.renewBorrow(borrow);
        return "forward:/manager/toReturnBook";
    }

    @RequestMapping("/findBorrowByTag")
    public  String  findBorrowByTag(String  rid,Model  model,HttpSession
session){
        List<Borrow> borrowList=borrowService.findBorrowByTag(rid);
        model.addAttribute("allBorrow",borrowList);

        session.setAttribute("s","tag");
        return "/manager/returnmain";
    }
}
```

（5）图书种类管理模块

后台关键代码：

```
@Controller
@RequestMapping("/manager")
public class TypeController {

    @Autowired
    TypeService typeService;

    @RequestMapping("/deleteType")
    public String toAddType(Integer tid){
        typeService.deleteTypeById(tid);
        return "forward:/manager/allTypes";
    }

    @RequestMapping("/addType")
    public  String addType(@ModelAttribute("type")Type type){
        typeService.addType(type);
```

```
        return "forward:/manager/allTypes";
    }

    @RequestMapping("/allTypes")
    public String findAllGoodsType(Model model){

        List<Type> typeList=typeService.findAllGoodsType();
        model.addAttribute("allTypes",typeList);
        return "/manager/typemain";
    }
}
```

5.结论与展望

经过两周的努力，我终于完成了本次图书管理系统的设计，在这个过程中学的了很多知识。虽然只是短短的两周，但在两周中我把老师在课堂上讲的知识运用到实践中来，巩固了所学的知识。

选定了专业综合实训的内容后，我到网上查找了一些资料，通过学习的理论知识与对查找的资料的分析，制定了图书管理系统的需求。在程序编写过程中使用已有的前台页面；完成管理员登录，图书信息管理，读者信息管理，图书种类管理，图书借阅、续借、归还管理，五个模块后台代码设计。同时在编程过程中解决了一系列遇到到问题，使程序可以正常使用。

这次专业综合实训可以看成一次理论与实践相结合的桥梁，通过这次专业综合实训我学习到了许多知识，也认识到了自己的一些不足，那就是缺乏相应的知识与经验。通过这段时间的课程设计，我认识的再编写程序时，应尽量是界面简洁大方，布局统一；一定要合理的定义与使用变量。在本次专业综合实训过程中难免存在一些的不足，希望老师能够批评和支出。

通过专业综合实训设计我进一步掌握了课本上的基础知识，掌握了专业综合实训的基本方法和原理，激发了学习的兴趣，加深了对理论知识的理解。同时。我也对 ssm 框架有了更深的认识，也认识的代码的编写不是一个简单过程，需要耐心和信心。只有多次编辑，多次编译，再多次运行才能写出好的程序，达到理想的效果。

6.主要参考文献

[1]陈恒，楼偶俊，张立杰 Java EE 框架整合开发入门到实践（微课版） 北京：清华大学出版社

[2]施伯乐，丁宝康，汪卫.数据库系统教程（第 3 版）.北京：高等教育出版社

[3]张海藩，牟永敏.软件工程导论（第 6 版）.北京：清华大学出版社

7.附录

列出源程序的文件名清单或需附加的文档（程序电子档）

（1）Controller

1）BookController.java

2）BorrowController.java

3）ManagerController.java

4）ReaderController.java

5）TypeController.java

（2）Service

1）BookService.java

2）BorrowService.java

3）ManagerService.java

4）ReaderService.java

5）TypeService.java

6）Impl

① BookServiceImpl.java

②BorrowServiceImpl.java

③ManagerServiceImpl.java

④ReaderServiceImpl.java

⑤TypeServiceImpl.java

（3）Dao

1）BookDao.java

2）BorrowDao.java

3）ManagerDao.java

4）ReaderDao.java

5）TypeDao.java

（4）Entity

1）Book.java

2）Borrow.java

3）Manager.java

4）Reader.java

5）Type.java

（5）Mapper

1）BookMapper.xml

2）BorrowMapper.xml

3）ManagerMapper.xml

4）ReaderMapper.xml

5）TypeMapper.xml

（6）utils

DataFormat.java

成绩评分表

序号	评价内容	评价等级				
1	设计思路阐述清晰，格式符合要求	优□	良□	中□	及格□	不及格□
2	结构严谨，逻辑性强，语言表达准确	优□	良□	中□	及格□	不及格□
3	文字通顺，技术用语准确，基本概念清楚	优□	良□	中□	及格□	不及格□
4	基础理论知识扎实，回答问题有理论根据	优□	良□	中□	及格□	不及格□
5	有较强的计算机应用能力	优□	良□	中□	及格□	不及格□
6	功能齐全完善，能正确处理实验数据	优□	良□	中□	及格□	不及格□
7	系统设计与实现方法有技巧性、新有创新意识	优□	良□	中□	及格□	不及格□
8	有一定的理论或应用价值，设计思路新颖，对问题有较深刻的认识	优□	良□	中□	及格□	不及格□
9	学习态度端正，主动参与性强	优□	良□	中□	及格□	不及格□
10	与指导教师及时沟通，学风严谨务实，按期圆满完成规定的任务	优□	良□	中□	及格□	不及格□
备注						
总分		教师签字				