

# 《社区图书管理系统》检测报告

PaperOK

总相似率：53.87%

## 基本信息

文档名称	社区图书管理系统
报告编号	ba5d41c0-e336-4c18-a33e-384d9bc1ad52
文档字数	10043
提交人姓名	张无忌
提交方式	粘贴文本检测
检测范围	学位论文库（含硕博）、学术期刊库、会议论文库、法律法规库、互联网资源库、自建比对库
提交时间	2018-04-21 19:28:06

## 检测报告指标详情

原创率	抄袭率	引用率	字数统计	参考文献字数
46.13%	53.87%	0%	10043	-

## 相似片段位置图



注：红色部分为重度相似，橙色部分为中度相似，蓝色部分为引用部分

## 相似片段详情（仅显示前10条）

序号	篇名	来源	命中率
1	“呼应”服务平台服务器端的设计与实现	学位论文库	3.41%
2	我们为什么要尝试前后端分离 - 萝卜 - 博客园	互联网资源库	2.27%
3	基于Vue.js的前后端分离Demo-程序员	互联网资源库	1.8%
4	云课堂家校互动系统的设计与实现	学位论文库	1.77%
5	maven编写主代码与测试代码 - qileilove - BlogJava	互联网资源库	1.03%
6	基于MVC的学生管理信息系统的设计与实现	学位论文库	0.86%
7	图书管理系统需求分析-CSDN下载	互联网资源库	0.85%
8	基于SpringBoot + Mybatis实现SpringMVC Web项目 - 追着..._博客园	互联网资源库	0.82%
9	基于MOOC模式的课程教学系统的设计与实现	学位论文库	0.8%
10	企业信息管理系统的研究与实现	学位论文库	0.8%

## 文档原文标注

### 1 绪论

社区图书管理系统是一套为便利社区图书管理员管理社区图书资源，读者资源，借书还书信息等

而编写的中小型图书管理系统。

## 1.1 课题背景及目的

社区图书管是社区精神文明建设的必要公共设施。由于社区图书馆在规模和重视程度上远远不能比拟市属图书馆等大型图书馆。所以目前仍有很多社区图书馆管理图书工作基本上是人工操作,面对大量书籍、读者信息以及两者间相互联系产生的借书信息、还书信息等大量任务,出现了负荷严重,缺乏方便性,效率很低,出错率高的弊端。加之,社区图书馆较普通图书馆存在着社区化,便民化,鼓励私人贡献图书等等特性。针对社区定制化的社区的图书管理系统的产生迫在眉睫。

## 1.2 论文的目的

本系统可实现社区读者查询图书信息,借阅图书,归还图书,管理个人账户等读者常用功能;图书管理员可通过本系统进行图书管理,图书借阅管理与归还管理,图书偿还管理,读者管理等;超级管理员可进行系统数据的管理和备份等。通过本系统可满足不同角色的需求,从而实现便民化,社区化的图书管理体系,为社区精神文明建设添砖加瓦。

## 1.3 论文的主要内容

论文内容主要包括:绪论,系统相关的技术,系统分析,系统详细设计和实现,系统测试以及总结和展望

## 2 系统可行性分析

本章节从技术可行性、经济性、可实施性、可操作性等几个方面论证了系统研发的可践行。

### 2.1 技术可行性

2.1.1本系统开发语言选用了Java。Java是一种面向对象设计语言,支持封装、抽象、继承和多态。这些特性使得系统的设计更便于基于现实业务进行模拟和抽象,更加便于逻辑的组织 and 处理。Java语言支持跨平台,可以比较便利的移植到各种操作系统上,增强了系统的可移植性和可维护性。Java语言的健壮性还能保障服务的稳定和安全。Java做Web开发更是有其广大的技术成熟做支持。

2.1.3 本系统使用了SpringBoot框架,便于快速构建项目,简捷的安全策略集成,自动管理依赖,自带应用监控,支持运行期内嵌容器,其内嵌的容器的设计更是便于部署和安装。

2.1.4 本系统使用了Spring MVC框架。Spring MVC框架提供了一个DispatcherServlet作用前端控制器来分派请求,同时提供灵活的配置处理程序映射、视图解析、语言环境和主题解析,并支持文件上传。Spring MVC还包含了多种视图技术,例如JSP、Velocity、Tiles、iText和POI等。Spring MVC分离了控制器、模型对象、分派器以及处理程序对象的角色,这种分离让它们更容易进行定制。

2.1.5 本系统选用MySQL作为数据库。MySQL是一个小型关系型数据库管理系统。支持FreeBSD、Linux、MAC、Windows等多种操作系统。可移植行高,安装简单小巧。支持常见的SQL语句规范。调试、管理,优化简单。其InnoDB引擎支持索引创建,能够高效的进行查询,还支持事务的创建,能够保障数据的事务安全性。

### 2.2 经济性

本系统所选用的软件和技术都是开源和免费技术,无需额外为此支付费用。且服务所需的硬件设备没有硬性要求,普通的机器,甚至廉价硬件都可以胜任。

## 2.3 可实施性

本系统部署架构简单，只需Web服务器和容器，数据库服务器，即可满足基本需求。如若为增强数据安全和可靠性，可增加主从配置。选用的SpringBoot框架可内嵌Tomcat容器，支持以Jar包的方式通过简单命令即可部署。整体而言，本系统部署简单，可实施性极强。

## 2.4 可操作性

本系统为B/S架构，用户打开浏览器即可使用系统。系统图书、读者、借阅等模块清晰，操作界面简洁，易于上手熟悉。

## 3 需求分析

本章节主要包括需求概述、功能需求分析、数据库需求分析等。

### 3.1 需求概述

系统主要包括图书管理、借阅管理、图书检索、读者管理等功能。按管理业务可将整个管理系统分为三大类：用户、图书信息、读者数据管理。用户管理又分为用户的增删改查。其中，用户又细分为超级管理员、管理员、读者等。图书管理又包括图书信息的增删改查等。

图书借阅又包括图书的借阅和图书的归还。借阅管理还包括图书借阅信息的增删改查。其中，图书借阅还涉及图书借阅的本数、欠费等逻辑，算是比较复杂的流程。

### 3.2 功能需求分析

系统用户主要包括读者、图书管理员、会员管理员、系统管理员。

#### 3.2.1 读者功能需求分析

- (1) 图书信息查询：根据图书名称、编号、作者等进行图书检索。
- (2) 图书借阅查询：根据时间段等查询借阅历史。
- (3) 会员信息查询以及欠费信息查询

#### 3.2.2 图书管理员功能需求分析

- (1) 图书信息管理：包括图书信息的增删改查
- (2) 图书借阅信息管理：包括借书还书信息的管理

#### 3.2.3 会员管理员功能需求分析

- (1) 读者信息管理：包括读者信息增删改查
- (2) 黑名单信息管理
- (3) 欠费信息管理
- (4) 偿还欠费管理

### 3.2.4 系统管理员功能需求分析

- (1) 权限管理
- (2) 用户数据管理
- (3) 图书数据管理
- (4) 系统配置及维护
- (5) 公告管理

## 3.3 数据库需求分析

### 3.2.1 数据库性能需求预估

社区图书管同时读者流量并不大。因此，初步定位数据库的最大并发连接数为100，可满足社区读者产生的数据的读写需求。

### 3.2.2 数据库能力需求

- (1) 读写分离

图书管理系统中图书数据的查询要远远高于数据的写入。因此数据库要满足快速高效查询的需求，具有读写分离的能力。

- (2) 分库分表

图书管理系统会产生大量的数据，常年累月的数据累计可能会导致数据查询效率的下降，根据时间等做分库分表可解决查询效率问题，所以可分库分表也是数据的必备能力。

- (3) 事务支持

图书管理系统流程复杂，有的流程可能包括多次对数据库的读写。为保证数据的安全性，事务支持的能力也是必须的。

- (4) 索引支持

大量的查询操作，大量的数据，索引查询能提高查询效率，提升用户体验。

### 3.2.3 硬件服务器需求

社区资金和资源有限，在有限的开支下保证业务和功能需求，需要性价比高的数据库软件和硬件选材标准。

综上所述，MySQL数据库软件是为最佳选择。

## 4 系统设计

本章节主要包括系统部署架构设计、系统技术架构设计、系统功能设计、数据库设计等。

### 4.1 系统部署架构设计

系统采用简单且实用的部署架构。Web服务器采用Nginx，web容器采用Tomcat，数据库采用MySQL。Nginx服务器对请求进行均衡和转发，同时部署有系统的静态资源。两个Tomcat服务器向外提供系统服务，部署有各模块服务。两个MySQL服务器读写分离，保障读写的高效性。

图4-1 系统部署架构图

## 4.2 系统技术架构设计

### 4.2.1 前后端分离

图4-2 前后端分离

前后端分离的设计，存在以下优势

#### (1)彻底解放前端

前端不再需要向后台提供模板或是后台在前端html中嵌入后台代码

#### (2)提高工作效率，分工更加明确

前后端分离的工作流程可以使前端只关注前端的事，后台只关心后台的活，两者开发可以同时进行，在后台还没有时间提供接口的时候，前端可以先将数据写死或者调用本地的json文件即可，页面的增加和路由的修改也不必再去麻烦后台，开发更加灵活。

#### (3)局部性能提升

通过前端路由的配置，我们可以实现页面的按需加载，无需一开始加载首页便加载网站的所有的资源，服务器也不再需要解析前端页面，在页面交互及用户体验上有所提升。

#### (4)降低维护成本

通过目前主流的前端MVC框架，我们可以非常快速的定位及发现问题的所在，客户端的问题不再需要后台人员参与及调试，代码重构及可维护性增强。

### 4.2.2 MVC设计

系统选用SpringMVC框架。MVC设计模式可以方便开发人员分工协作,提高开发效率,增强程序的可维护性和拓展性而且还利用Controller将Model与View分离,降低它们之间的耦合度。下图是本系统采用的SpringMVC 的原理说明。

图4-3 SpringMVC 原理

(1)客户端发出一个http请求给web服务器，web服务器对http请求进行解析，如果匹配DispatcherServlet的请求映射路径（在web.xml中指定），web容器将请求转交给DispatcherServlet。

(2)DispatcherServlet接收到这个请求之后将根据请求的信息（包括URL、Http方法、请求报文头和请求参数Cookie等）以及HandlerMapping的配置找到处理请求的处理器（Handler）。

(3-4)DispatcherServlet根据HandlerMapping找到对应的Handler,将处理权交给Handler（Handler将具体的处理进行封装），再由具体的HandlerAdapter对Handler进行具体的调用。

(5)Handler对数据处理完成以后将返回一个ModelAndView()对象给DispatcherServlet。

(6)Handler返回的ModelAndView()只是一个逻辑视图并不是一个正式的视图，DispatcherServlet通过ViewResolver将逻辑视图转化为真正的视图View。

(7)Dispatcher通过model解析出ModelAndView()中的参数进行解析最终展现出完整的view并返回给客户端。

## 4.3 系统模块设计

本系统包含的核心功能模块如下：系统登录模块、图书管理模块、图书借阅管理模块、读者管理模块、系统管理模块。

### 图4-4 系统模块

#### 4.3.1 系统登录模块

(1) 登入。游客无需登录，但只能查询图书信息。会员用户需要登录才能进入系统，登录后，后台会自动判断登录者的类型，授予用户不同权限的操作。权限包括游客权限、普通读者权限、管理员权限、系统管理员权限。系统用户类别如下图所示。

### 图4-5 系统用户

### 图4-6 登录流程

(2) 登出。登出后，用户身份权限等同游客，仅有查询图书权限。

#### 4.3.2 图书管理模块

图书管理员进入图书管理界面后，拥有对图书进行增删改查等一系列的操作权限。

(1) 添加图书：添加新的图书，在数据库中记录相应信息。

### 图4-7 添加图书流程

(2) 修改图书信息。

### 图4-8 修改图书流程

(3) 移除图书：图书馆中的一些图书会因为过时或损坏等一些列原因不在具有收藏价值，需要从图书库移除。

### 图4-9 删除图书流程

#### 4.3.3 图书借阅管理模块

图书借阅管理主要包括：图书借阅、图书归还、查看借阅信息。

(1) 图书借阅：根据读者编号和图书编号，进行图书的借阅过程，记录借阅信息入库。

### 图4-10 图书借阅流程

(2) 图书归还：根据读者编号和图书编号，查看借阅信息，并更新借阅记录为已还书。

#### 图4-11 图书归还流程

(3) 查看借阅记录：查看读者的借阅记录或者图书的借阅记录，需要读者编号或图书编号

#### 4.3.4 会员管理模块

会员管理主要包括用户信息的增删改查。

(1) 添加用户信息：管理员可对用户信息进行录入，使之成为系统用户，并为其分配角色，赋予其相应的权限。

(2) 删除用户信息：当用户办理注销业务时，将用户信息状态标记为已注销，进行逻辑删除。删除后的用户不再拥有原有的系统权限。

(3) 更改用户信息：用户注册信息有变化时，如手机、邮箱联系方式信息需要更改时或者用户角色发生改变时，进行用户信息的更新。

(4) 用户有权限查看个人的用户信息；管理员也有权限进行用户信息查询和进行统计。其中，个人信息包括但不限于用户的系统信息，用户的借阅信息，用户的欠费信息等。

(5) 系统进行违规操作时，会将用户加入黑名单。黑名单内的用户不再享有相应的系统权限。

(6) 用户欠费信息管理及欠费偿还。

#### 4.3.5 系统管理模块

系统管理主要包括以下模块：初始化系统、数据维护、权限管理、统计数据、公告管理等。

(1) 初始化系统：作为图书管理系统，初始化系统数据如图书类别信息、用户类型信息等字典数据，以便系统在其基础上运行。初始化配置如配置常用角色以及图书借阅规则等，以便系统流程按照预定设计进行。

(2) 数据维护：为预防不可预测因素导致的数据丢失，系统提供数据备份和数据恢复功能。系统管理员通过手动方式和定期任务方式将系统数据以指定格式导出。一旦出现数据丢失，及时进行数据导入进行恢复，可减少和挽回一定的经济损失。

(3) 权限管理：系统管理员可维护所有的权限，包括给图书管理员和会员管理员指定权限。维护用户密码等。

(4) 统计数据：数据的收集和统计可以帮系统运营着更好的了解系统的运营状况；分析图书管理的动态走势，用户的活跃度；分析各类图书的热度等等。这些都有利于系统功能的改进和系统服务的优化，使系统更好的利于系统用户。本系统统计数据的数据类型主要包括图书信息统计、读者信息统计、借阅信息统计，维度为图书种类、读者类别、图书数量、借阅时间等等。

(5) 公告管理：为了让用户有更好的系统使用体验。本系统提供系统公告管理功能。系统管理员可通过公告告知用户将最新图书到馆，系统维护时间，图书馆活动信息等。公告将直接发布到系统首页，这样，用户可以在访问系统时方便的阅读公告信息。此外已注册用户还可以通过手机短信、邮箱收件的方式接受到系统公告，达到第一时间知晓公告的良好体验。

#### 4.4 系统数据库设计

系统数据库设计包括：实体设计、E-R设计、表设计等。

( 1 ) 实体设计： 主要实体如下

图4-12 图书信息实体

图4-13 读者信息实体

图4-14 图书借阅信息

( 2 ) E-R设计： 主要E-R图如下

图4-15 读者图书ER图

( 3 ) 表设计

图书信息表

属性名称 含义 数据类型 长度 必须 约束 描述

id 图书ID bigint Y 主键

code 图书编号 varchar 50

name 图书名称 varchar 50 Y

type 图书类型 tinyint 4 Y

author 作者 varchar 50 Y

translator 译者 varchar 50 Y

isbn ISBN varchar 30 Y

publisher 出版社 varchar 50 Y

edition 版次 varchar 20

price 图书价格 bigint Y 单位分

quantity 副本数量 int Y

brief\_intro 简介 varchar 500

pic 封面图片 blob

读者信息表

属性名称 含义 数据类型 长度 是否必须 约束 描述

id 读者ID bigint Y 主键



name 读者姓名 varchar 50 Y

reader\_type 读者类型 tinyint 4 Y 1 游客 , 2 普通读者 , 3 VIP

card\_no 卡号 varchar 20

card\_date 发卡日期 datetime

card\_status 卡状态 tinyint 4 1 可用 , 2 欠费 , 3 禁用 , 4 注销

phone 电话 varchar 15

借阅信息表

属性名称 含义 数据类型 长度 是否必须 约束 描述

id 借阅编号 bigint Y 主键

book\_id 图书ID bigint

book\_code 图书编码 varchar 50

reader\_id 读者ID bigint Y

out\_date 借书日期 datetime Y

return\_date 还书日期 datetime Y

lost\_date 挂失日期 datetime Y

borrow\_times 借书次数 int 10 Y

operator 操作员 bigint Y

图书类别表

属性名称 含义 数据类型 长度 是否必须 约束 描述

id 图书类别编号 bigint Y 主键

name 图书类别名称 bigint Y

description 描述 varchar 100 Y

读者类型表

属性名称 含义 数据类型 长度 是否必须 约束 描述

id 读者类别ID bigint Y 主键

name 读者类别名称 varchar 50 Y

reader\_type 读者类型 tinyint 4 Y 1 游客， 2 普通读者， 3 VIP

quantity 限借数量 int Y

days 限借期限 int

penalty 逾期罚金 bigint 按逾期每天

## 5 系统实现

本章节主要包括登录模块的实现、图书管理模块的实现、会员管理模块的实现、图书借阅模块的实现、系统管理模块的实现等。

### 5.1 登录模块的实现

本系统可在非登录状态下使用，但是只能进行图书查询。若果想要其他操作如借阅等必须在登录状态下。点击登录进入登录页面，用户需输入正确的用户名和与之相匹配的密码才能进入系统。若用户名和密码不匹配或者用户不存在，不能进入系统，系统会进行错误信息提示。登录成功后系统会根据用户角色授予用户相应的权限。用户在自己的权限集内使用本系统。

图5-1 登录界面

### 5.2 图书管理模块的实现

系统的图书管理模块功能主要有浏览查询图书信息、新书入库、移除图书、统计等。

#### 5.2.1 新书入库

新书入库业务如下图所示，填写图书的基本信息，完成后点击添加。图书录入信息校验通过后会插入到图书信息表，并会自动为图书分配系统唯一标识图书编号。

图5-2 添加图书界面

#### 5.2.2 图书管理

图书管理主要是修改图书信息、删除图书等。根据图书查询条件进行图书查询，然后对结果列表中的某条图书信息进行编辑和删除操作。图示如下：

图5-3 图书管理界面

图5-4 修改图书信息界面

### 5.3 会员管理模块的实现

#### 5.3.1 添加用户

添加用户信息，填写完用户信息后，点击保存完成。校验通过的信息会写入用户表，同时按照编号生成规则生成用户编号。

图5-5 添加用户信息界面

#### 5.3.2 用户信息管理

此界面主要是管理用户信息，会员管理员拥有删除和修改用户信息的权限。会员管理员按照查询条件查询到结果后，即可对目标记录进行管理操作。

图5-6 用户信息管理界面

## 5.4 图书借阅模块的实现

图书借阅管理功能包括办理借书业务、图书归还、借阅信息浏览。浏览借阅信息可以根据图书编号或读者编号来查看。

## 5.5 系统管理模块的实现

系统管理主要包括以下模块：初始化系统、数据维护、权限管理、统计数据、公告管理等。

（1）初始化系统：系统管理员登录系统后，通过系统初始化界面，选择初始化数据文件，导入初始化数据即可。初始化系统数据如图书类别信息、用户类型信息等字典数据等。

（2）数据维护：数据维护：系统管理员可在数据维护界面指定导出数据文件的名将数据文件导出到指定位置。另外系统会自动根据任务规则进行数据备份。

（3）权限管理：系统管理员可在权限管理界面选定角色进行权限维护。也可在用户信息对用户指定特定的角色。

（4）统计数据：系统管理员可在系统统计模块进行数据统计。统计类型初步提供读者借阅信息统计、读者统计、图书统计。在借阅数据统计界面，指定时间段等查询条件即可进行相应的统计。读者统计和图书统计操作类似。

（5）公告管理：系统管理员在公告页面填写公告信息。公告信息包括标题、内容、公告开始时间、公告结束时间、公告类型（系统展示还是推送用户）。公告创建后，后台任务会在定时扫描公告表，以便发现公告并触发公告展示或发送。

## 6 系统测试

本章节主要包括测试目的、测试环境、测试用例覆盖、功能模块测试、可靠性测试、兼容性测试、易用性测试和安全性测试。

### 6.1 测试目的

软件测试是为了发现错误而执行程序的过程。其目的是为了发现尽可能多的缺陷，但不能保证软件无任何缺陷。所以，本系统测试目标是保证图书管理系统输出结果和功能流程能按照预期正常执行。且能保证系统的可靠性、安全性和易用性。

### 6.2 测试环境

本系统测试环境如下，系统为Ubuntu Server 16.04LTS, SDK为JDK1.8，容器为Tomcat9，浏览器为Chrome、火狐和IE8,9,10。上述测试环境，基本覆盖了常用的服务端和客户端环境，能够较好的逼近现实实施场景。

### 6.3 测试用例覆盖

#### 6.3.1 查询或统计

单独遍历各查询条件，测试按各查询条件是否都能够查询出相应的值；随机对各查询条件进行组合查询，测试是否能够实现组合查询。如查询借阅信息，可分别根据读者编号和图书编号进行查询，也可根据二者结合条件进行查询。

### 6.3.2 输入框

文本框输入场景，如图书信息、用户信息录入。对必填项，留空进行提交，看是否提示必填。对有长度限制项，输入超长字符，看有无超长提示。

### 6.3.3 数据导入导出

数据导入场景如初始化数据导入和恢复数据导入，可导入正常数据看能否完成导入，再次导入非正常数据看能否正确判断。

数据导出场景如备份系统数据。导出全部数据时，查看导出的数据正确性。在磁盘已满和无写盘权限时，看是否有友好提示。

### 6.3.4 分页

输入页数，点击go按钮，测试是否能执行翻页操作；点击首页、上一页、下一页、末页，测试各页面转换是否正常；测试在各边界面值再进行翻页是否正确。

### 6.3.5 保存

进行新增或修改操作，输入相关内容后，多次提交保存，测试是否限制了重复保存。

### 6.3.6 提交/确定按钮

对于提交/确定按钮，点击后，数据是否成功处理，页面跳转是否正确；连续点击提交/确定按钮页面如何处理，注意连续提交时操作要快，上个提交尚未处理完成，点击下一个提交，程序如何处理，是否报错。

## 6.4 单元测试

本系统借助SpringTest+JUnit+Mockito进行单元测试，对Dao层，Service层和Controller层的核心功能代码块进行了测试用例编写，并借助Maven的test插件批量执行测试用例。

## 6.5 功能模块测试

根据需求对应用的所有模块，所有功能的触发事件逐一进行验证。一方面顺从需求来对功能进行操作，看是否达到了需求规定的预期；另一方面与需求背道而驰看程序是否有相应的异常控制等。如图书借阅，需求中要校验读者是否已达借阅上限。可以先在低于借阅上限的场景借阅图书，若能成功借阅，则测试通过。然后在已达用户借阅上限的场景再次借阅图书，若能提示用户已达借阅上限，则测试通过。

## 6.6 可靠性测试

借阅图书场景做持续反复操作，借阅图书然后归还图书，看场景是否稳定可靠。

## 6.7 兼容性测试

浏览器兼容性，用Chrome，Firefox，IE等不同浏览器，不同版本分别执行各个场景。看展示效

果和流程结果是否一致。

## 6.8 易用性测试

6.8.1 不仅要验证操作流程是否符合大部分用户的操作习惯。

6.8.2 是否有多余的操作流程，对于多余的流程是没有任何好处的，不仅使操作繁琐还有可能出现不必要的漏洞。

6.8.3 操作流程是否顺畅，会不会产生让用户误解的步骤。

6.8.4 每个步骤操作完以后，需要给出提示的是否明确给出。

## 6.9 安全性测试

权限安全，如普通读者，发送删除图书信息请求，是否会无权限提示；不登录时，浏览器中直接输入登录后的地址，看是否可以直接进入。

## 6.10 小结

本系统测试过程由代码块的单元测试到模块的功能测试，再到整个系统的集成测试。层层递进的测试机制，提高了测试的覆盖率。

在测试过程中，发现了许多由于粗心导致的低级错误，也出现过影响流程和使用的缺陷。然而最终的结果是：系统的核心功能模块如系统登录、图书查询、图书借阅、会员管理、系统管理等都以正常运行。没有一个人能一次就写出完美无瑕的程序，系统测试是系统实施的前提保障。通过系统而又科学的测试，才得以使系统能够高效、稳定、安全的运行。

## 结 论

本系统是个人独立从零到有的完成的一个产品。从最初的可行性分析，到需求范围确定、需求分析，到系统设计，再到系统实现，再到系统测试，最终到部署到测试个人的测试服务器运行程序。每一个阶段都是一个调查、学习、运用知识、总结知识的过程。这每个过程都让我收获颇多，成就满满，进步多多。

需求分析过程中，我了解到了什么是迭代开发，先给系统圈定一个小的需求范围，比如，图书管理系统的核心就是图书管理，那就先确定图书管理的需求，然后去设计这各需求模块，再去实现；最核心模块实现后，再添加登录模块；然后是会员模块；再后来是借阅模块。就这样，一步一步，给系统添砖加瓦，最终集成一个完善的图书管理系统。

系统设计过程，是我投入精力最多，也是收获最多的一个过程。技术选型和架构方案，怎样才能更快速高效的将产品通过技术去实现，最终我选择了基于配置的可快速构建的SpringBoot+SpringMVC+MyBatis的框架组合；功能流程的设计我反复推敲，旨在让流程更简洁，让功能更简单专一。数据库和模型设计决定了程序设计的复杂度，将实体抽象化，单一化，用面向对象的思想去设计，会使问题变的更简单。

在不断的发现错误，修正错误；遇到困难，寻找解决方案。本次毕设，提高了自己解决问题的能力 and 实践能力，让我再次深刻体会到坚持和认真的重要性。