

# 佛山大学

## 系统设计文档

基于SpringBoot + Vue的图书管理系统  
的设计与实现

学	院：	电子信息工程学院
专	业 名 称：	物联网工程2班
学	生 姓 名：	刘嘉明
学	号：	20220070204

完成日期： 2024 年 7 月 24 日

# 目录

1 引言 .....	3
1.1 编写目的 .....	3
2 总体设计 .....	3
2.1 需求概述 .....	3
2.2 软件结构 .....	4
3 模块设计 .....	4
3.1 模块基本信息 .....	5
3.2 功能概述 .....	6
3.3 算法 .....	6
3.4 模块处理逻辑 .....	7
4 数据库设计 .....	8
4.1 ER 图表 .....	8
4.2 表设计 .....	8
4.2.1 图书信息 .....	8
4.2.2 图书类型 .....	9
4.2.3 借阅信息 .....	9
4.2.4 用户信息 .....	9
5 接口设计 .....	9
5.1 外部接口 .....	9
5.1.1 登录界面 .....	9
5.1.2 首页 .....	10
5.1.3 用户管理界面 .....	11
5.1.5 图书类型列表界面 .....	12
5.1.6 图书借阅管理界面 .....	12
5.2 内部接口 .....	13
5.2.1 登录接口 .....	15
5.2.3 图书管理接口 .....	16
5.2.4 图书类型管理接口 .....	16
5.2.5 图书借阅管理接口 .....	17
5.2.6 用户管理接口 .....	17
6 性能 .....	18
6.1 精度 .....	18
6.2 时间特性 .....	18
6.3 灵活性 .....	18
7 测试 .....	18

# 1 引言

## 1.1 编写目的

本报告的目的是对图书管理系统进行详细设计说明,以便用户及项目开发人员了解产品详细的设计与实现,为开发人员提供开发参考书。以下叙述将结合文字描述、伪代码,图表等来描述图书管理系统的详细设计和相关的模块描述。本报告的预期读者有客户、项目经理、开发人员以及跟该项目相关的其他人员。

# 2 总体设计

## 2.1 需求概述

- (1) 图书信息的管理,包括图书信息的条件查询、录入、修改和删除。
- (2) 图书类型的管理,包括建立图书类型、查询图书类型、修改图书类型和删除图书类型。
- (3) 用户信息管理。包括对系统管理员和读者的信息进行维护管理,可以新增、修改、删除和条件分页查询用户信息。
- (4) 图书借阅与归还,记录图书借阅与归还信息,能判断该书是否可借,供读者选择图书并借阅,对借阅的图书执行归还操作。
- (5) 用户注册和登录,读者可以通过登录页面注册新账号,管理员和读者可以使用用户名和密码登录图书管理系统。

## 2.2 软件结构

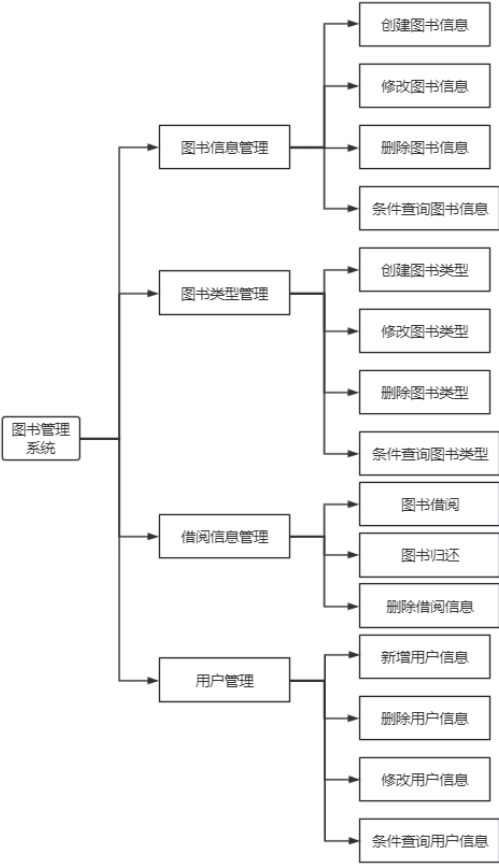
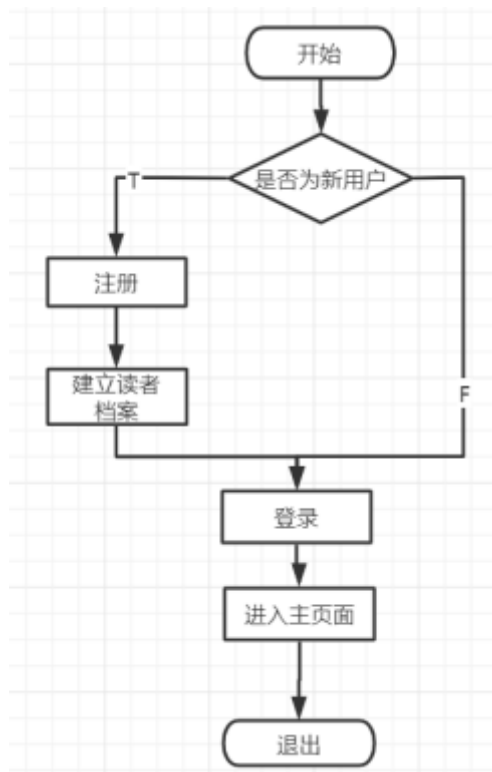


图 1 高校图书管理系统的软件结构图

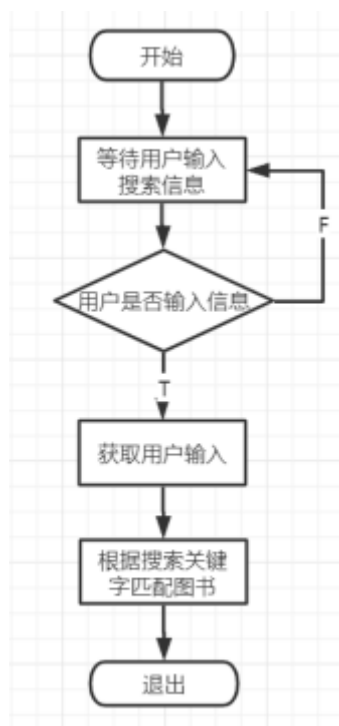
描述：图书管理系统向下划分为图书管理、图书类型管理、用户信息管理和借阅管理四个模块。图书管理主要负责对图书的信息进行管理，以及管理员进行图书的查询；图书类型管理主要关联图书信息进行分类，可以对类型进行管理，以及对类型进行条件查询；借阅管理主要负责读者进行借书、还书操作，以及查询读者历史借阅信息；用户信息管理可以针对图书管理员和读者的信息进行管理。

## 3 模块处理逻辑



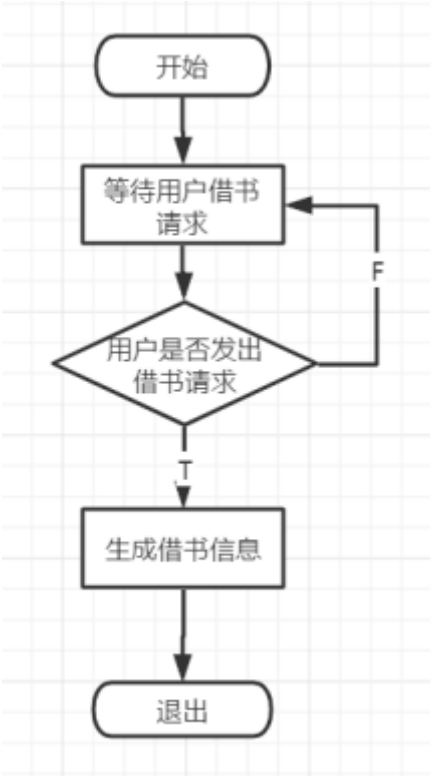
读者管理程序流程图

图 2 为读者管理的程序流程图，新读者要先注册，然后建立读者档案，之后再选择登录，已经注册过的读者可以直接进行登录，登录之后进入主页面。



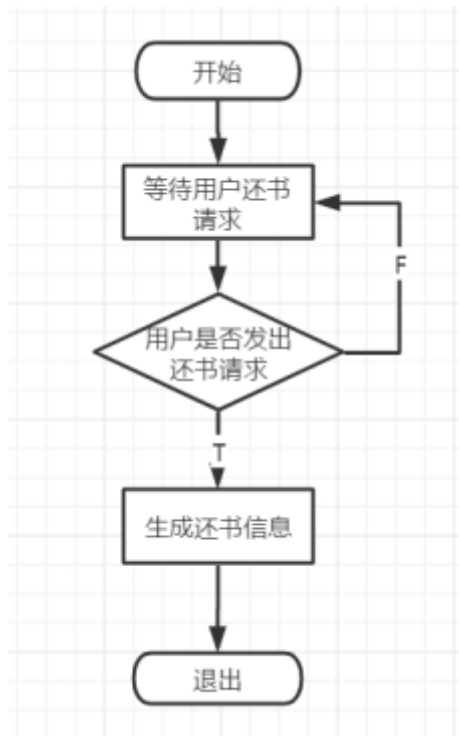
查询图书信息程序流程图

图 3 为查询图书信息的查询流程图，该模块首先进入等待用户输入搜索信息状态。如果用户已经输入则获取用户输入并根据输入的搜索关键字在图书库中匹配图书最后退出。如果用户未输入则模块继续处于等待用户输入搜索信息状态。



读者借书程序流程图

图 4 是读者借书的程序流程图，该模块首先进入等待用户借书请求状态。如果用户发出请求则生成借书信息并写入图书借阅库，如果用户未发出请求则模块继续处于等待用户借书请求状态。

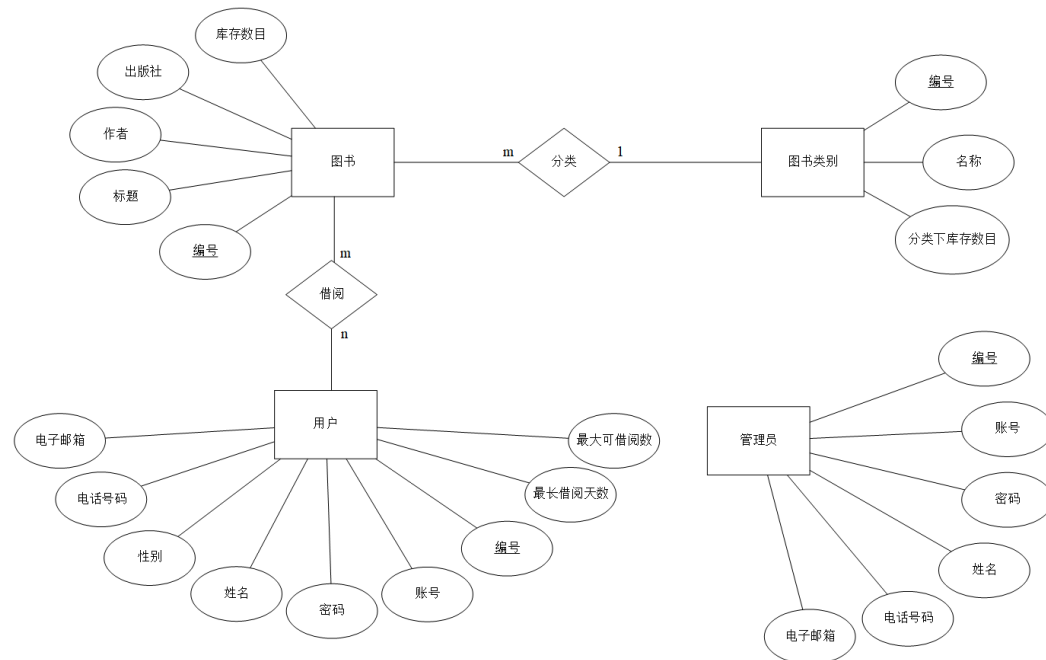


读者还书程序流程图

图 5 是读者还书的程序流程图，该模块首先进入等待用户还书请求状态。如果用户发出请求则生成还书信息并写入图书借阅库，如果用户未发出请求则模块继续处于等待用户还书请求状态。

## 4 数据库设计

### 4.1 ER 图表



### 4.2 表设计

#### 4.2.1 图书信息

数据名称	数据类型	数据描述
bookId	Int	图书 id
bookName	Varchar	图书名称
bookAuthor	Varchar	图书作者
bookPrice	Decimal	图书价格
bookTypeId	Int	图书类型 id
bookDesc	Varchar	图书描述
isBorrowed	Tinyint	借阅标记。1 表示借出，0 表示归还
bookImg	Varchar	图书封面图片链接



## 4.2.2 图书类型

数据名称	数据类型	数据描述
bookTypeId	Int	类型 id
bookTypeName	Varchar	图书类型名称
bookTypeDesc	Varchar	图书类型描述

## 4.2.3 借阅信息

数据名称	数据类型	数据描述
borrowId	Varchar	借阅信息 id
userId	Int	借阅读者的 id
bookId	Int	借阅图书信息 id
borrowTime	Datetime	借阅时间
returnTime	Datetime	归还时间

## 4.2.4 用户信息

数据名称	数据类型	数据描述
userId	Int	用户 id
username	Varchar	用户名
userPassword	Varchar	用户密码
isAdmin	Tinyint	是否管理员标记。1 是管理员，0 是读者

# 5 接口设计

## 5.1 外部接口

### (1) “用户注册” 模块

## 图书管理系统-登录界面

 admin

 ..... 

 管理员 

登录

注册

管理员 username: admin    password: admin

读者 username: liujiaming    password: 123456

用户需要输入用户名、密码。系统会对输入的信息进行验证，确保其符合要求。

注册成功后，用户的基本信息将被存储在用户表中。

### (2) 用户登录模块

## 图书管理系统-登录界面

 admin

 .....

 管理员

登录

注册

管理员 username: admin password: admin

读者 username: liujiaming password: 123456

系统会验证用户的凭证，如果凭证正确，用户将被重定向到相应的管理界面。

### (3) 首页

首页

图书管理

其他管理

首页

admin

欢迎使用图书管理系统

开发人员: 佛山大学 物联网工程2 刘嘉明

联系方式-Email: 3257715328@qq.com

联系方式-QQ: 3257715328

项目开发环境介绍

操作系统: Windows 11 专业版

开发工具: IntelliJ IDEA 2023.1.3

Java版本: 1.8.0

服务器: Tomcat 8.5

数据库: MySQL 5.7

前端技术: HTML+CSS+JavaScript+Vue+Axios+ElementUI

后端技术: Java+SpringBoot+MyBatis+MySQL



(4)

### 图书分类管理模块

图书分类管理模块允许管理员对图书的分类进行管理。管理员可以查看现有的分类列表，添加新的分类，编辑或删除现有分类。这个模块支持搜索功能，方便管理员快速找到特定的分类。

### (5) 图书管理模块



图书管理模块使管理员能够管理图书的详细信息，包括标题、作者、分类、库存数量和借阅所需的积分。管理员可以添加新书、编辑现有书籍信息或删除书籍。此外，还可以查看每本书的借阅情况。

### (6) 用户管理模块



用户管理模块允许管理员查看和编辑用户信息，包括用户名、积分和用户身份（如普通用户或管理员）。管理员可以为用户充值积分或更改用户的身份和权限。

## (7) 借阅管理模块



借阅管理模块处理用户的借书和还书请求。用户可以在这个模块中搜索图书、发起借阅请求，并在借阅成功后扣除相应的积分。还书时，系统会更新图书的库存状态和用户的借阅记录，并相应地调整积分。

## 5.2 内部接口

### (1) 数据库连接模块

```
server.port=8092
server.servlet.context-path=/BookManager
```

```

spring.datasource.driver-class-
name=com.mysql.cj.jdbc.Driver
spring.datasource.url=jdbc:mysql://127.0.0.1:3306/book_ma
nager?useAffectedRows=true
spring.datasource.username=root
spring.datasource.password=123456
spring.redis.host=100.75.29.25
spring.redis.port=6379
spring.redis.password=123456
spring.servlet.multipart.max-file-size=10MB
spring.servlet.multipart.max-request-size=100MB
spring.servlet.multipart.enabled=false
logging.config.classpath=log4j2.xml

```

## (2) 跨域问题

### (3) @Configuration

```

public class CorsConfig implements WebMvcConfigurer {

    @Override
    public void addCorsMappings(CorsRegistry registry)
    {
        registry.addMapping("/**")

            //是否发送 Cookie

            .allowCredentials(true)

            //设置放行哪些原始域    SpringBoot2.4.4 下低版
            本使用.allowedOrigins("*")

            .allowedOriginPatterns("*")

            //放行哪些请求方式

            .allowedMethods("GET", "POST", "PUT", "DELET
E")

            //.allowedMethods("*") //或者放行全部

            //暴露哪些原始请求头部信息

            .allowedHeaders("*");
    }
}

```

### (3)安全校验

```
@Configuration //表明是配置类
public class InterceptorConfig implements WebMvcConfigurer {

    // 所有的页面和请求
    private static final String[] all = {
        "**/*.html",
        "/bookInfo/**",
        "/bookType/**",
        "/borrow/**",
        "/update/**",
        "/user/**"
    };

    // 与登录和注册相关的界面和请求
    public static final String[] aboutLogin = {
        "/",
        "/index.html",
        "/register.html",
        "/user/login",
        "/user/register"
    };

    // 与读者相关的页面和请求
    public static final String[] aboutReader = {
        "/reader/**/*.html", // 读者页面
        "**/reader/**",        // 读者请求
    };

    @Override
    public void addInterceptors(InterceptorRegistry registry) {
        // 注意，因为浏览器缓存问题，导致不一定能拦截页面，但是功能请求
        // 肯定做不了，所以也不影响安全性。

        /**
         * 用户拦截器思路：
         * 1. 拦截所有的页面和请求
         * 2. 排除与登录和注册相关的
         */
    }
}
```

```

        */
        /*registry.addInterceptor(new UserInterceptor())
            .addPathPatterns(all)
            .excludePathPatterns(aboutLogin);*/

        /**
         * 读者拦截器思路:
         * 1. 拦截所有页面和请求
         * 2. 排除与登录相关的页面和请求
         * 3. 排除与读者有关的页面和请求
         */
        /*registry.addInterceptor(new ReaderInterceptor())
            .addPathPatterns(all)
            .excludePathPatterns(aboutLogin)
            .excludePathPatterns(aboutReader);*/
    }
}

```

#### (4) 异常处理

```

public class NotEnoughException extends RuntimeException{
    public NotEnoughException() {
        super();
    }

    public NotEnoughException(String message) {
        super(message);
    }
}

public class OperationFailureException extends RuntimeException{
    public OperationFailureException() {
    }

    public OperationFailureException(String message) {
        super(message);
    }
}

```

#### (6) 拦截器

```

public class ReaderInterceptor implements HandlerInterceptor {
    @Override
    public boolean preHandle(HttpServletRequest request, HttpServletResponse
response, Object handler) throws Exception {

```



```

        User user = (User) request.getSession().getAttribute("userObj");

        // 根据链式拦截关系，到这个拦截器肯定是登录过的。
        if(user.getIsadmin() == 0) {    //如果是用户，则拦截
            System.out.println("读者不能进管理员界面!");
            // 重定向到登录界面
            response.sendRedirect(request.getContextPath() + "/index.html");
            return false;
        }
        return true;    //放行
    }
}

import cn.edu.fosu.liujiaming.model.User;
import org.springframework.web.servlet.HandlerInterceptor;

import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;

// 用户拦截器，拦截未登录不能访问的请求
public class UserInterceptor implements HandlerInterceptor {

    @Override
    public boolean preHandle(HttpServletRequest request, HttpServletResponse
response, Object handler) throws Exception {
        User user = (User) request.getSession().getAttribute("userObj");

        if(user == null) { //没有登录
            System.out.println("没有登录!不能访问!");
            // 重定向到登录界面
            response.sendRedirect(request.getContextPath() + "/index.html");
            return false;
        }

        return true;    //放行
    }
}
}

```

(7) 业务代码由于采用 MVM 模型，展示不方便，见压缩包

## 6 性能

### 6.1 精度

要按照严格的数据格式输入，不能输入非法字符，否则系统不给予响应进行处理，查询时要保证准确率为 100%，所有包含查询关键字的书籍都应能查到，不能有遗漏。

### 6.2 时间特性

- (1) 响应时间：用户任意操作后 5 秒内系统给予反馈信息。
- (2) 更新处理时间：由系统运行状态来决定。
- (3) 数据的转换和传送时间：能够在 20 秒内完成。

### 6.3 灵活性

当需求发生某些变化时，该软件的基本操作、数据结构、运行环境等等基本不会发生变化，只是对系统的数据库的文件和记录进行处理，就可以满足需求。

## 7 测试

功能点	测试用例	输出结果
读者登录	选择读者角色，输入读者的用户名和密码，点击登录	跳转到图书管理系统首页，并展示读者相关功能菜单
读者注册	点击注册，跳转注册页，填入用户名、密码、确认密码，点击注册	未填写完整表单，提示对应异常，填写完整表单，提示注册成功
管理员登录	选择管理员角色，输入管理员用户名和密码，点击登录	跳转到图书管理系统首页，并展示管理员相关功能菜单
图书信息新增	点击新增，输入图书信息，点击确认新增	未填写完整表单，提示对应异常，填写完整表单，提示新增成功
图书信息查询	输入对应查询条件，点击查询	图书信息列表
图书信息修改	选中图书信息，弹出图书信息修改框，修改对应信息，点击修改	未填写完整表单，提示对应异常，填写完整表单，提示修改成功

图书信息删除	选中删除的数据	提示删除成功
图书类型新增	点击新增，输入图书类型信息，点击确认新增	未填写完整表单，提示对应异常，填写完整表单，提示新增成功
图书类型查询	输入对应查询条件，点击查询	图书类型列表
图书类型修改	选中图书类型信息，弹出图书类型修改框，修改对应信息，点击修改	未填写完整表单，提示对应异常，填写完整表单，提示修改成功
图书类型删除	选中删除的数据	提示删除成功
图书借阅	读者选中图书，点击借阅图书	提示借阅成功，并生成借阅时间
图书归还	读者选中借阅中的图书	提示归还成功，并更新归还时间
用户新增	点击新增，输入用户信息，点击确认新增	未填写完整表单，提示对应异常，填写完整表单，提示新增成功
用户信息查询	输入对应查询条件，点击查询	用户信息列表
用户修改	选中用户信息，弹出用户信息修改框，修改对应信息，点击修改	未填写完整表单，提示对应异常，填写完整表单，提示修改成功
用户删除	选中删除的数据	提示删除成功