

图书馆管理系统技术报告

单 位：中国海洋大学信息科学与技术学院

项目名称：图书管理系统

小组名称：系统分析第七小组

小组组长：左云浩

小组成员：张根、黄可人、李思琦、张慧婷、屈耘野、王泓元

指导老师：张春海

**目录**

1. 引言 ·······························································3

1.1 编写目的 ·······························································3

1.2 项目背景 ·······························································3

1.3 术语与缩写解释 ·······························································3

1.4 开发工具 ·······························································3

1.5 开发架构 ·······························································3

1. 系统设计 ·······························································4

1.1需求分析 ·······························································4

2.1.1功能需求分类——前台

2.1.2功能需求分类——后台

2.1.3系统用例图

2.2数据结构设计 ·······························································7

2.2.1用户User

2.2.2书籍Book

2.2.3借阅记录BorrowInfo

2.2.4 留言板Board

2.2.5 公告notice

2.2.6用户反馈Feedback

1. 技术说明 ·······························································10

3.1数据生成 ·······························································10

3.2数据储存说明 ·······························································10

3.3 书籍检索底层技术说明 ·······························································10

3.3.1按照书名查询

3.3.2按照ISBN查询

3.3.3按照编号查询

3.4 借阅与归还 ·······························································12

3.4.1借阅流程图

3.4.2归还流程图

四、技术难题与解决方法 ·······························································13

五、收获与感想 ·······························································14

1. **引言**

1.1编写目的

对项目开发中设计原则、技术创新点、技术难点进行总结。

1.2项目背景

1.2.1项目名称

图书馆管理系统

1.2.2项目提出者

《系统分析》课程教师——张春海教授

1.2.3项目开发人员

设计人员：黄可人

开发人员：左云浩、张根、王泓元、屈耘野

测试人员：张慧婷、李思琦

1.2.4项目使用者

普通用户、图书管理员、游客

1.3 术语与缩写解释

|  |  |
| --- | --- |
| **缩写、术语及符号** | **解 释** |
| 用户 | 系统的使用者，可以注册或登录账号 |
| 管理员 | 管理网站后台的用户 |
| 公告 | 系统（管理员）发布公告 |
| 检索 | 根据不同类别进行搜索，获得相关信息 |
| 借阅/归还 | 将图书带离/带回图书馆 |
| 编号 | 给每本书一个特定编码 |
| 添加/删除图书 | 管理员在后台对图书信息进行管理 |
| 反馈 | 用户将相关信息反映给管理员 |

1.5 开发工具

Visual Studio 2017

QT

Github共享文档

1.6 开发架构

数据访问层——业务逻辑层——界面层

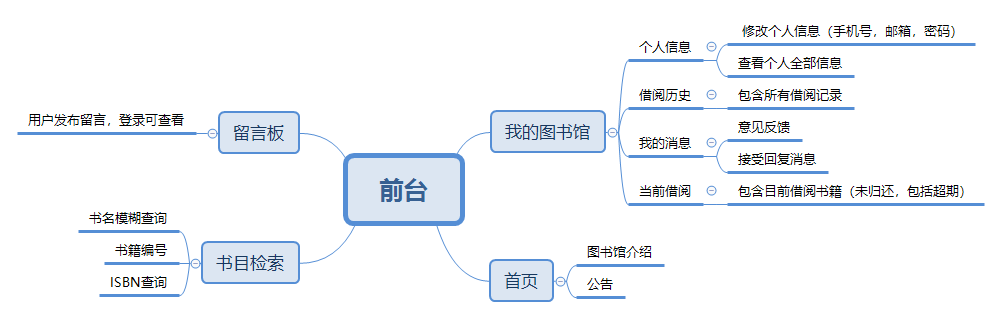
1. **系统设计**

**2.1需求分析**

2.1.1产品中的角色

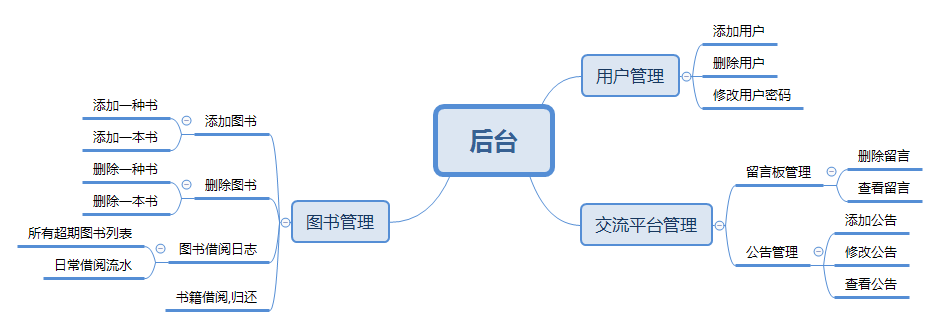
|  |  |
| --- | --- |
| **角色名称** | **职责描述** |
| 后台管理员 | 进行图书管理，可负责图书借阅、归还，管理图书（添加、修改图书信息，查询借阅信息），添加公告，管理留言板，反馈用户信息。用户管理，查询该用户个人基本信息，目前借阅情况（当前借阅目录，历史借阅目录），初始化用户登录密码 |
| 前端用户 | 可进入我的图书馆查看或修改个人信息，查看借阅历史及当前借阅，书籍检索（按类别、书名、作者），查看图书推荐，分类浏览图书（按专业排序），进行留言或向管理员反馈信息 |
| 游客 | 可以查看图书馆简介及规章制度，进行图书检索，可以登陆 |

2.1.2功能需求分类——前台



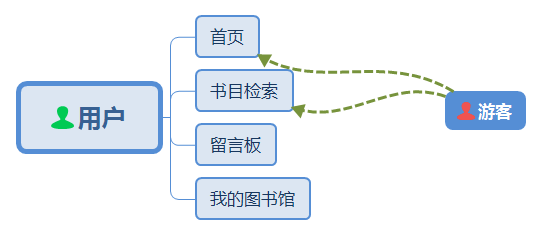
| 功能类别 | 功能名称、标识符 | 描述 |
| --- | --- | --- |
|  | 用户登录 | 如果未登录，点击我的图书馆进入登陆系统，获得前用户权限，可以进入我的图书馆 |
| 书目检索 | 检索 | 可分别通过按书名、书籍编号、书籍ISBN编码，获得馆内藏书相关信息 |
| 我的图书馆 | 个人信息 | 管理（查看、修改）个人信息，包括基本信息（姓名、昵称、学号、手机号，邮箱，专业，年级等），修改用户的手机号，邮箱，登录密码 |
| 借阅历史 | 查看用户所有书籍借阅历史 |
| 当前借阅 | 展示当前已借阅书籍及借阅、应归还日期，特别标注超期图书 |
| 我的消息 | 可以向管理员提出反馈意见，接受管理员的回复消息 |
| 首页 | 图书馆介绍 | 简单介绍图书馆信息 |
| 公告 | 可查看管理员发布的公告信息 |
| 留言板 | 留言板 | 用户登录后，可发布、查看留言 |

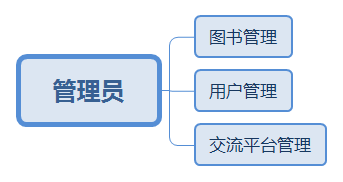
2.1.3功能需求分类——后台



| 功能类别 | 功能名称、标识符 | 描述 |
| --- | --- | --- |
| 图书管理 | 添加/删除图书 | 增添/删除图书信息，可添加删除一种书籍，也可根据ISBN查询添加删除相关书籍的副本 |
| 管理图书借阅日志 | 管理读者借阅归还书籍的相关信息，生成修改相关借阅归还记录。输出所有超期的图书借阅信息。 |
| 书籍借阅、归还 | 在管理员权限内，可以让用户对书籍进行借阅和归还，录入借书人和图书相关信息 |
| 用户管理 | 添加用户 | 可通过输入用户的详细信息添加用户 |
| 删除用户 | 通过查询用户学号，删除用户信息 |
| 修改用户密码 | 帮助忘记密码的用户，修改用户密码 |
| 交流平台管理 | 公告管理 | 管理员可以随时发布新的公告，删除公告 |
| 留言板，用户反馈 | 管理员可以删除留言板不恰当言论，接受用户反馈信息，并进行回复 |

2.1.4系统用例图





**2.2数据结构设计**

2.2.1用户User

|  |  |  |  |
| --- | --- | --- | --- |
| 表名 | 用户User | | |
| 属性名 | 别名 | 类型 | 长度 |
| 学工号 | id | Char\* | 30 |
| 姓名 | realName | Char\* | 30 |
| 专业 | major | Char\* | 30 |
| 年级 | grade | Char\* | 30 |
| 密码 | pwd | Char\* | 30 |
| 手机 | phone | Char\* | 30 |
| 邮箱 | email | Char\* | 30 |
| 性别 | sex | Char\* | 2 |
| 年龄 | age | int | sizeof(int) |

|  |  |  |  |
| --- | --- | --- | --- |
| 表名 | userIdIndex | | |
| 属性名 | 别名 | 类型 | 长度 |
| 用户id | id | Char\* | 30 |
| 位置地址 | index | long | sizeof(long) |

2.2.2书籍Book

|  |  |  |  |
| --- | --- | --- | --- |
| 表名 | 书籍Book | | |
| 属性名 | 别名 | 类型 | 长度 |
| 书籍编号 | ISBN | String | 30 |
| 书名 | name | String | 30 |
| 作者 | author | String | 30 |
| 出版社 | press | String | 30 |
| 专业类别 | category | String | 30 |
| 出版时间 | pubdate | long | sizeof(long) |
| 已借阅数量 | borrowNum | int | sizeof(int) |
| 总数量 | num | int | sizeof(int) |
| 价格 | price | float | sizeof(float) |

|  |  |  |  |
| --- | --- | --- | --- |
| 表名 | BookNameIndex | | |
| 属性名 | 别名 | 类型 | 长度 |
| 书名 | name | string | 30 |
| 地址位置 | index | long | sizeof(long) |

|  |  |  |  |
| --- | --- | --- | --- |
| 表名 | BookIdIndex | | |
| 属性名 | 别名 | 类型 | 长度 |
| 书籍编号 | id | string | 30 |
| 地址位置 | index | long | sizeof(long) |
| 是否被借阅 | isBorrowed | int | sizeof(int) |

2.2.3 借阅记录BorrowInfo

|  |  |  |  |
| --- | --- | --- | --- |
| 表名 | 借阅记录BorrowInfo | | |
| 属性名 | 别名 | 类型 | 长度 |
| 借阅人id | userid | Char\* | 30 |
| 借阅人姓名 | Username | Char\* | 30 |
| 书籍Id | Bookid | Char\* | 30 |
| 书籍名 | Bookname | Char\* | 30 |
| 借阅日期 | Borrowdate | Char\* | 20 |
| 应还日期 | Backdate | Char\* | 20 |
| 是否归还 | Isback | Int | sizeof(int) |

2.2.4 留言板board

|  |  |  |  |
| --- | --- | --- | --- |
| 表名 | 留言板board | | |
| 属性名 | 别名 | 类型 | 长度 |
| 留言人姓名 | Name | Char\* | 30 |
| 留言人id | Id | Char\* | 30 |
| 内容 | Content | Char\* | 100 |
| 留言时间 | Time | Char\* | 30 |
| 是否保留 | Delflag | Int | sizeof(int) |

2.2.5公告notice

|  |  |  |  |
| --- | --- | --- | --- |
| 表名 | 公告notice | | |
| 属性名 | 别名 | 类型 | 长度 |
| 公告编号 | Id | Int | Sizeof（int） |
| 公告标题 | Title | Char\* | 30 |
| 内容 | Content | Char\* | 100 |
| 时间 | Times | Char\* | 30 |

2.2.6用户反馈Feedback

|  |  |  |  |
| --- | --- | --- | --- |
| 表名 | 用户反馈Feedback | | |
| 属性名 | 别名 | 类型 | 长度 |
| 反馈编号 | Id | Int | Sizeof(int) |
| 用户编号 | UserId | Char\* | 30 |
| 用户姓名 | RealName | Char\* | 30 |
| 反馈内容 | Content | Char\* | 100 |
| 回复内容 | ReplyContent | Char\* | 100 |
| 反馈时间 | Times | Char\* | 20 |

1. **技术说明**

**3.1数据生成**

第一步：创建一个bookInfo.txt文件（利用函数随机生成或者爬取数据），图书信息按书本结构体顺序用英文逗号“,”分割。

第二步：利用txtToVector\_book()函数，将.txt数据读取存入vector中，返回vector<Book>参数。

第三步：调用 GenerateData::CreateBook(vector<Book> book);其中有三次写入，一次是对书本信息的写入，输出给 Book.dat 文件，一次是生成编号索引，输出给 BookIdIndex.dat ，最后是生成书名索引，输出为 BookNameIndex.dat。

用户数据生成同理。

**3.2 数据存储说明**

3.2.1图书文件

图书文件分为三个：

1.Book.dat 用于存放不同种类书籍的信息

2.BookNameIndex.dat 用于存放书名索引

3.BookIdIndex.dat 用于存放不同书籍编号索引

其中，Book.dat中的数据与BookNameIndex.dat一一对应，两者存放的信息以一类书为基本单位，而BookIdIndex.dat存放的是某类书的所有编号对应的书籍。

3.2.2 用户文件

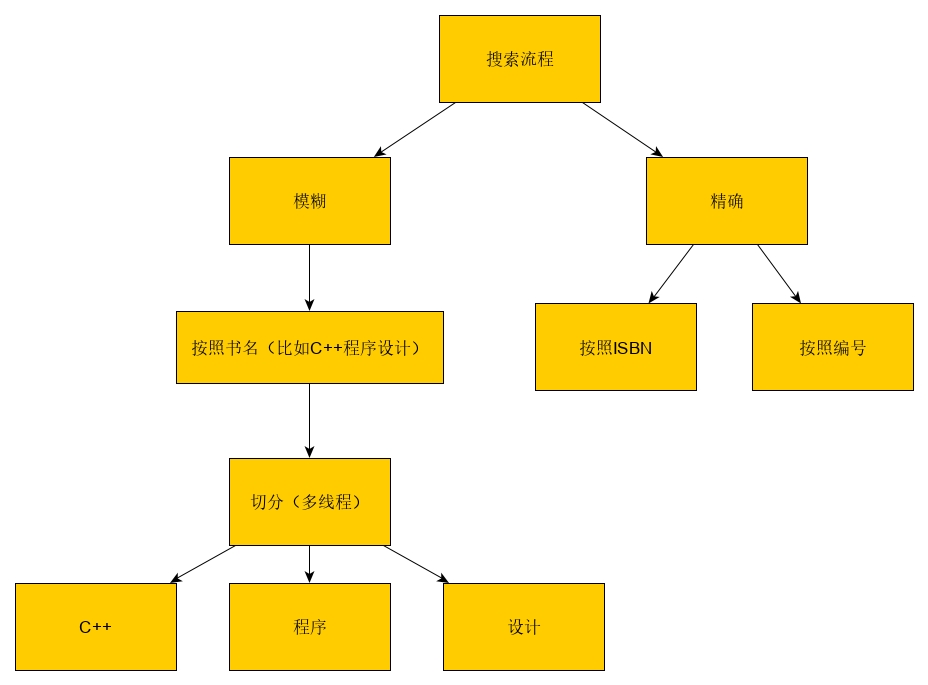
用户文件分为二个：

1. User.dat 用于存放每一个用户的具体信息（包括管理员的信息）
2. UserIdIndex.dat 用户存放每一个用户的编号索引

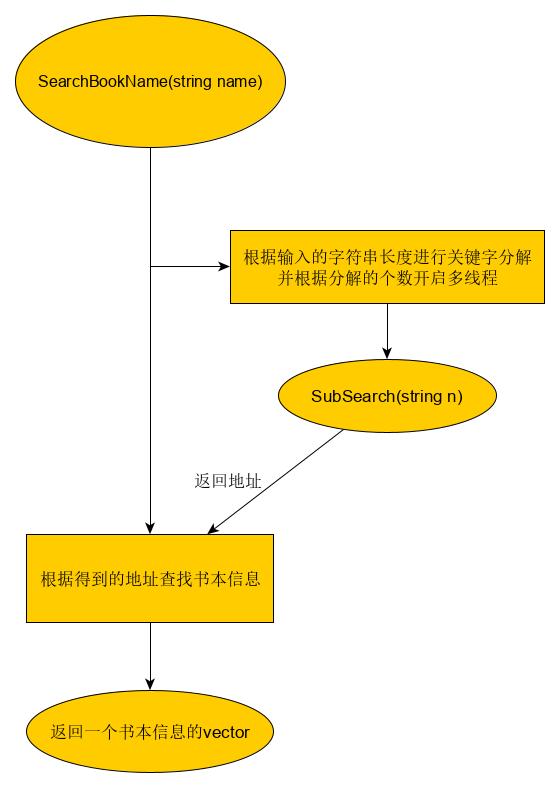
User.dat中的数据与UserIdIndex.dat一一对应,l两者存放的信息以用户id为一个基本单位。

其他文件同理，不做详细说明。

**3.3 书目检索底层技术说明**

****

3.3.1 按照书名查询



vector<pair<Book, long>> SearchTool::SearchBookName(string name)

此函数为模糊搜索，传入参数为书名，将传入的书名根据长度切分，最多三等分，然后调用多线程进行搜索。

具体搜索行为在void SearchTool::SubSearch(std::string name)中。如果 name 是书名中的字符串，如 name = "C", 书名：“C++程序设计”。则将对应书本的地址存入全局变量 vector<long> index中。

搜索结束后返回SearchBookName(string name)，将index中的重复地址去除，然后根据得到的地址在书本文件中直接获取书本信息。将得到的信息存入vector<Book> book中返回。

3.3.2 按照ISBN查询

pair<Book, long> SearchBookISBN(char id[avglen])

传入对应的ISBN编号就可以搜索，如果没有，返回的地址将是 -1。

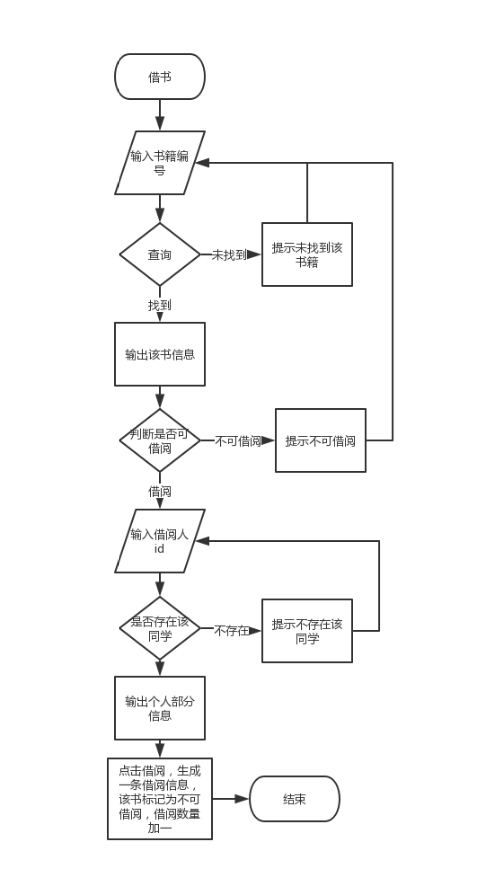
3.3.3按照编号查询

pair<Book, long> SearchTool::SearchBookId(char id[avglen])

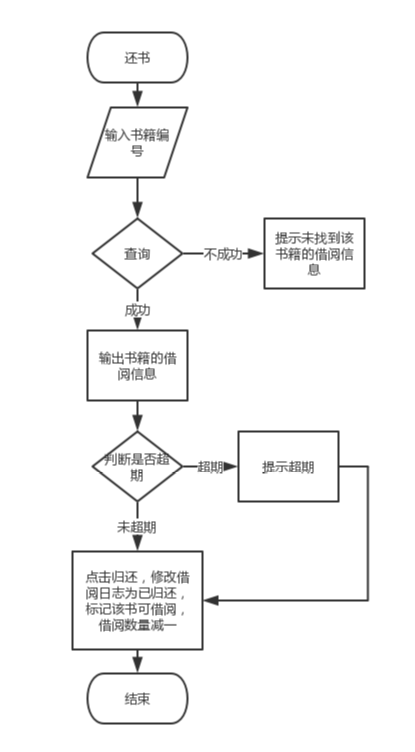
传入一个书本编号进行搜索，主要用于管理员界面。同样返回一个包含书本信息的vector<Book>，不同的是，返回值只包含最多一本书。因为本项目默认书本编号不重复，此函数在查到符合的编号后便返回。

**3.4借阅归还**

3.4.1借阅流程图



3.4.2归还流程图



1. **技术难题与解决方法**

① 用户反馈（管理员界面）：管理员回复某条消息时，需要修改feedback.dat文件该条消息所在的位置存放的feedback.content。最初想要通过文档的重新读写来解决这个问题，但操作繁琐，浪费时间。

解决方法：使用seekp函数跳转，根据sizeof(feedback)及类中的成员大小跳转到指定位置存放管理员回复的消息。

② 需求文档出来后，大家还是很迷茫不知道如何实现。就按照功能进行分工，国庆过后发现根本写不出来，只能写出一些伪代码，浪费了许多时间和精力。

解决方法：重新整理一下思路，决定将开发过程分为三层架构（数据访问层——业务逻辑层——界面层），两个方向前进，其一是解决命名规范化、解决各个文件的读写问题（数据访问层），然后才解决具体功能的实现，其二是设计出主要功能界面，等待接口函数完成，再将接口函数封装到qt中。

③在各部分功能完善后，我们简单测试一下，如果按照规定的正常输入不会出问题，但是比如在用户注册的时候，如果随便输入，用户名或者密码过长就会出现写入数据的错误，由此会出现很多意想不到的bug。

解决方法：我们通过学习，了解到qt中关于输入输出限制的函数，例如下面的代码：

QRegExp regx("[0-9]+$");

QValidator \*validator = new QRegExpValidator(regx, ui.lineEdit);

ui.lineEdit->setValidator(validator);

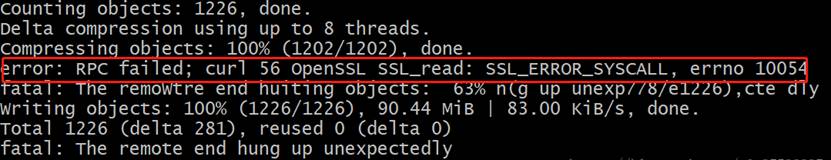
就可以限制那个输入框只能输入数字。

QRegExp regx2("^[A-Za-z0-9]+$");

这样就限制只能输入大小写字母和数字。

通过 **ui.lineEdit->setMaxLength(8);** 来限制输入的长度。同时我们在函数逻辑中加入更多的判断，来完善我们的输入输出的规范化，防止恶意输入，影响系统运行。

④在GitHub上传较大文件时，在最后的push步骤中，出现了中途上传失败的问题



解决方法：GitHub限制单个大文件上传最大为100M解决方案：输入命令git config http.postBuffer 1024\*1024\*500后再次上传 输入命令git push origin master或者安装git lfs实现Git对大文件的支持。

⑤图书借阅信息涉及日期的推移、计算，由于存在闰年和月份天数不一致的情况，若直接用字符串存储，后续还需较为复杂的字段分割解析和日期计算逻辑，实现较为复杂，容易产生bug。

解决方案：创建日期类，使用运算符重载，这样日期的推移计算只需实现一次，可直接使用减号取得结果。另外，对于超期日期计算，无需进行日期解析，直接使用strcmp()函数，自左向右逐个字符按ASCII值比较，可以更快返回结果。

⑥删除记录时使用懒惰删除实现，否则在中间位置删除需要O(n)时间复杂度。但会造成较大冗余空间，随着数据量上升，对读写速度造成影响。

解决方案：增加维护一个记录序号字段，通过记录序号实现数据位置(字节数)的计算，写入带有删除标志(为1)的数据。软件初始化时统计标记删除的留言数量和占比，当标记删除的数据过多(>=30%)时，进行实际的磁盘删除(重新遍历生成文件)，节省占用空间。

⑦从txt里面进行文件读取时读取中文不成功，会显示乱码。

解决方案：把txt文件的存储编码改成ANSI编码。

⑧测试第六小组的时候第六小组的程序突然崩溃，导致剩余功能无法测试完成。

解决方案：第六小组的技术人员临时来修改了代码，把登录部分的代码整个删除了，接着测试。

1. **收获与感想**

这次项目我们来说是一个挑战，在一共7周的时间内，从组队到设计系统，我们遇到了很多挫折，但一一克服，并取得了较好的成果。在需求分析到系统设计的过程中，中间任何一个步骤都不能马虎。我们作为程序设计人员，必须对用户的需要进行认真分析，并给予分类，进行模块化功能设计，画出流程图、结构图，这样一来，项目才可以有条不紊地进行。

后端代码的编写是程序编写的基础，我们规划了首页、我的图书馆、书籍检索（包含借阅、归还图书）、留言板四个模块，并且在代码中将这些功能分别实现。与以往不同的是，图书管理系统没有使用数据库，并且需要结合QT设计界面，拥有人性化的界面设计。在解决无数报错，并逐渐完善功能后，系统得到了较好的呈现。

系统设计的过程中有面对bug的无奈，也有豁然开朗的喜悦。虽然受时间限制，系统还有很多需要完善的地方，但通过这个项目，我们进一步学习了许多知识，相信对于将来的学习生活必定大有裨益。