



東南大學

软件架构课程实验报告

题 目 图书管理系统

软件学院 院（系） 软件工程专业

组长： 朱云天

组员：赵君亮、项桂巳雨、蔡林希、安典坤、

董睿、陈嘉博

指导教师 王璐璐

起止日期 2020.3.16-2020.4.12

设计地点 遍布中国

目录

第一章 课题概述.....	3
1.1 选题背景及意义.....	3
1.2 实验的主要工作.....	3
1.3 实验报告的组织结构.....	3
第二章 需求分析.....	4
2.1 用户故事.....	4
2.2 备选流.....	5
2.3.1 非功能需求分析.....	5
2.3.2 质量场景分析.....	6
2.4 用例.....	6
2.4.1 用例图.....	6
2.4.2 用例描述.....	8
2.5 活动图.....	8
2.6 本章小结.....	10
第三章 系统设计.....	11
3.1 总体系统功能设计.....	11
3.2 体系结构设计.....	13
3.3 接口设计.....	16
3.4 本章小结.....	17
第四章 系统实现.....	18
4.1 类设计.....	18
4.2 数据库表设计.....	18
第五章 系统测试.....	20
5.1 注册及登录.....	20
5.2 用户模块测试.....	21
5.3 管理员模块测试.....	25
5.4 非功能需求满足测试.....	30
第六章 实验小结.....	31

第一章 课题概述

1.1 选题背景及意义

当今，图书日益剧增，图书管理就愈显必要。特别在省、市、高校的公共图书馆里，如何将图书管理、用户操作、管理员处理等一系列操作有机地结合起来形成系统越来越必要。

1.2 实验的主要工作

本实验目的为打造一个校园图书管理系统，实现新书登记，旧书作废，图书在阅览室的分配，图书的借阅和归还，期限限制和违约惩罚等功能，出具各类统计报表等一系列基本功能。本实验主要包括需求分析、架构设计、代码实现、测试和自我总结六个部分，旨在完备地进行实验。

1.3 实验报告的组织结构

本文主要分为以下几个部分：

第一章是绪论，主要介绍图书管理系统地研究意义，研究背景，主要工作，章节安排等。

第二章是图书管理系统的需求分析，按总分结构介绍系统的需求。

第三章是图书管理系统的设计，主要是对各个模块设计的分析。

第四章是图书管理系统的实现，主要的实现见文档附带的.cpp, .h 文件。

第五章为系统测试，主要是测试各个功能是否正常运行。

第六章为实验总结，主要总结实验遇到的苦难和反思。

第二章 需求分析

2.1 用户故事

1) 作为一个用户，在登陆系统后：

- 可以用手机号注册一个账户，需要信息：账户名、密码；
- 可以查看自己的罚款记录以及交罚款情况
- 可以交罚款
- 可以在系统主页随意浏览图书馆藏书信息。
- 可以根据书本信息来查询书本。
- 可以查看自己的个人信息。
- 可以提出修改自己的个人信息申请。
- 可以预约图书。
- 可以提出荐购图书申请。
- 可以借阅图书。
- 可以续借图书。
- 可以归还已借图书。

2) 作为一个管理员，在登陆系统后：

- 如果用户预约书，到期没来取，则取消预约，1 天内不得再次预约，并降低用户信用度。
- 如果用户借书，到期未归还，则降低用户信用度，并且按逾期进行罚款，逾期一天按照书价格的 1%进行处罚。
- 处理用户丢失书本，要求其按照市场价进行赔偿，并降低用户信用度。
- 如果发现用户的归还的书有损坏，则按照程度不同进行 10%-100%进行赔偿。
- 对于一般的借阅请求，设置借阅的时间为 30 天。
- 处理用户的续借请求时，先考虑其是否超期。如果超期先提示用户交罚款。
- 对于珍贵文献一类书籍，需单独借阅，限制用户的范围为教师 and 大学生，并且限制借阅期限为一周。
- 如果发现文献有损坏，在赔偿的同时取消其借阅的资格。
- 当图书分类方法更新时可以对分类目录进行增删改查。
- 可以设定某类图书在图书馆的位置（某房间某书架）。

- 当新入库一本图书时，选择图书类别（多级），如果该分类下已经有相同图书，修改它的数量；如果以前未收录该图书，增加这本书并设定它的数量、分类号、作者、馆藏地、条码号等相关信息。

2.2 备选流

- 普通用户修改个人信息时，单日提交申请若超过三次，则本日内禁止再次提交申请。
- 普通用户提出荐购图书申请时，若荐购图书已在图书馆藏书中，则提示图书馆已有该书，退回申请。
- 普通用户预约图书时，若图书馆中已有状态为“可借”的图书，则提示该图书可借，不接受预约。
- 普通用户借阅图书时，若该图书已被其他人抢先借阅，则提示已被申请，不接受借阅。
- 普通用户续借图书时，若该图书已被其他人预约，则提示已被预约，不接受续借。
- 普通用户归还图书时，若图书损坏，则用户信息中应交罚款增加一定数额。
- 普通用户归还图书时，若归还逾期，则用户信息中应交罚款增加一定数额。
- 普通用户借阅图书后，若借阅到期及逾期，则到期之后每天提醒应归还图书，直到用户归还图书。
- 用户预约书到期没有来领取，则书可以再次被其他的用户借阅，而该用户24小时内不可以再次预约其他书籍。
- 用户信用度低于80，用户将暂时不能借阅书籍，信用度每天提升1。
- 书籍可以在网上续借，最多2次，但是逾期书籍不可以续借。迟归还一本书降低5的信用度。
- 丢失一本书降低10信用度。
- 管理员对于借阅即将到期的书籍（期限前三天），每日发送一次邮件进行提醒。
- 用户如未在规定时间内归还，则自动开始罚款和积分的计算，同时继续每日发送邮件提醒。
- 对于文献的借阅，如果用户为大学生老师以外身份，则拒绝借阅请求。

2.3.1 非功能需求分析

- 性能需求：用户访问的系统应该能在规定的时间内做出响应，如果系统由于网络或者数据库原因不能在规定时间内做出反应，那么系统应该提出警告，不能出现用户无故长时间等待的情况。

- 安全性需求：用户端与数据库之间应该有访问限制，防止部分非法的数据请求。
- 易用性需求：不同的用户应该能够以不同形式无需教学的访问不同的内容。
- 可用性需求：系统提供 24 小时的服务，且很少停机。
- 可测试性需求：系统是各部分易于单独测试，并能方便地进行整体测试。

2.3.2 质量场景分析

- 性能场景：用户访问的系统应该能快速做出相应。若系统本身或数据库发生故障，应当给予用户明显的提示，最好的是将原因展示给用户看，避免用户无故地长时等待，引起使用者不满。系统支持的并发访问量应不少于 1000。数据库应实现有组织地，动态地存储大量关联数据，并支持快速数据快速的增、删、改、查。
- 安全性场景：若是普通用户则不能对数据库进行任何直接修改（“借书使得库存减少”等由于用户合法产生的间接修改除外），任何用户在查询前必须进行身份认证，避免校外人员查看校内的库存情况。
- 易用性场景：系统应当提供简洁、直接、友好、易操作的 UI 界面供用户使用，无需额外的教学说明。
- 可用性场景：系统可以随时进行访问，数据库 24 小时在线，出现故障时系统应该能够采用预设好的对应的处理方法。
- 可测试性场景：系统的各个功能、各个部分可以容易获得单独测试的机会，并且可以提供给测试人员整体测试的方便性。

2.4 用例

2.4.1 用例图

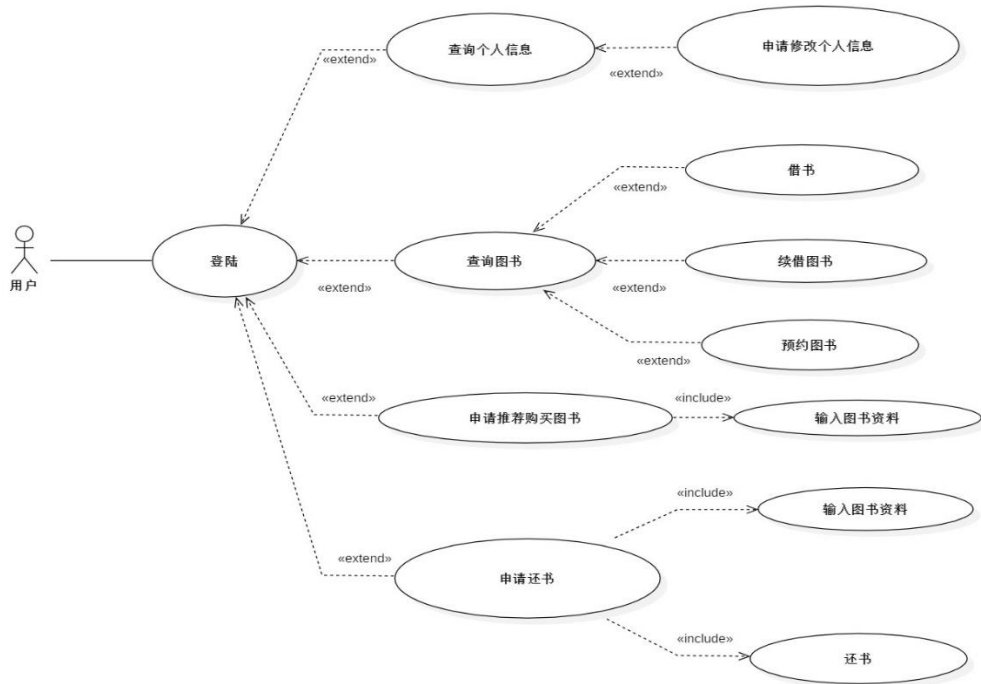


图 2.4.1.1 用户角度的用例图

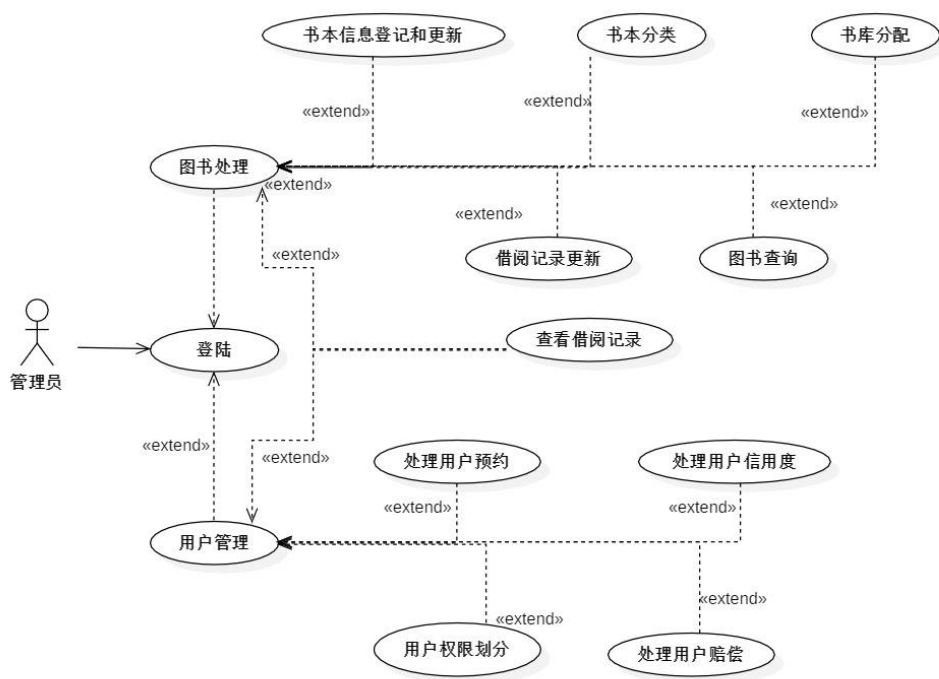


图 2.4.1.2 管理员角度的用例图

2.4.2 用例描述

表 1. 用户预约图书的用例描述

项	描述	注明
用例名称	预约图书	
简介	在库存为空时，用户可以预约图书，以便图书被他人归还时能第一时间接到书本	
基本流	1. 用户在弹出页面的搜索框中搜索书本名/作者名/出版社。 2. 在搜索结果页面中选中想要预约的书，点击预约图书按钮。 3. 等待通知来取书。	
可选流	1. 如果书本有库存且没人预约，提示不能被预约。 2. 如果搜索结果为空，提示用户，并返回上层菜单。 3. 如果用户信用过低，不能预约图书。 4. 如果用户所借图书已达上限，不能预约图书。	
先决条件	1. 用户登陆系统，选择了查询图书。	
后继条件	1. 在用户及书本数据中增加这一条预约记录。	

表 2. 管理员处理用户信用度的用例描述

项	描述	注明
用例名称	处理用户信用度	
简介	管理员可以根据用户行为，改变用户信用度	
基本流	1. 管理员可以在数据库中通过用户名字/一卡通号搜索到用户。 2. 管理员可以在选择用户后查看用户的信用度。 3. 管理员可以根据用户的行为，修改用户的信用度。	
可选流	1. 如果用户信用度减为一定值，通过预留的手机号码、邮箱地址提醒用户。 2. 在不同信用度区间内，对应地修改用户的借书上限、一次性预约数、借书时间等等。	
先决条件	1. 登陆系统，选择了用户管理。	
后继条件	1. 在用户数据库中修改更新对应用户的信用度、借书上限、一次性预约数、借书时间等数值。	

2.5 活动图

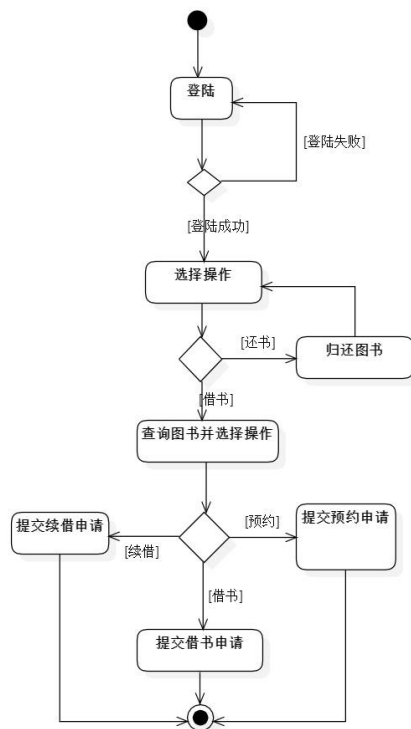


图 2.5.1 用户借书、还书、续借的活动图

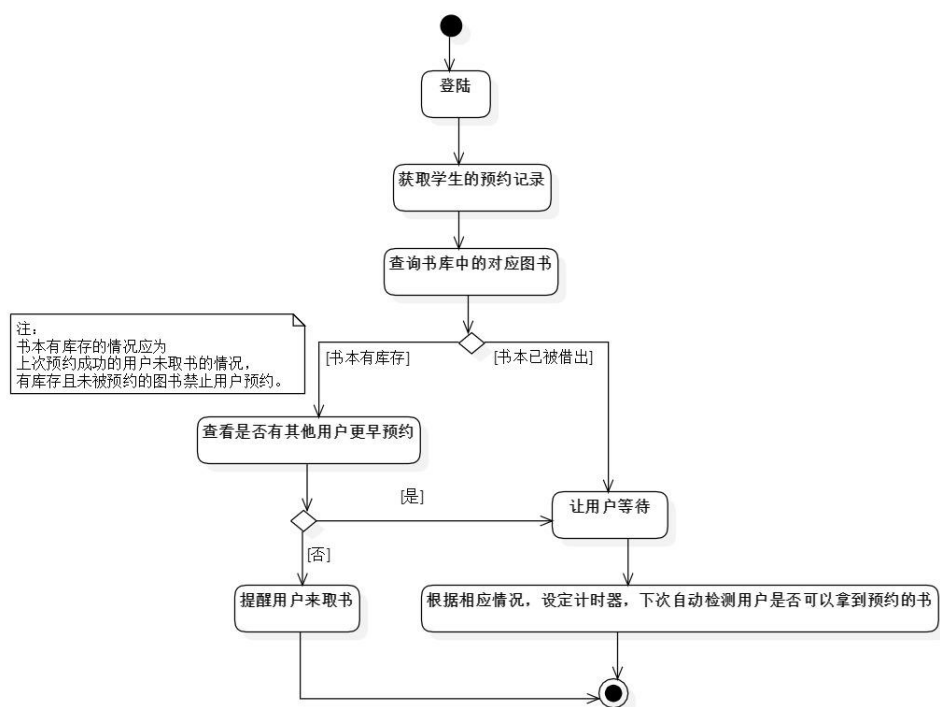


图 2.5.2 管理员处理预约的活动图

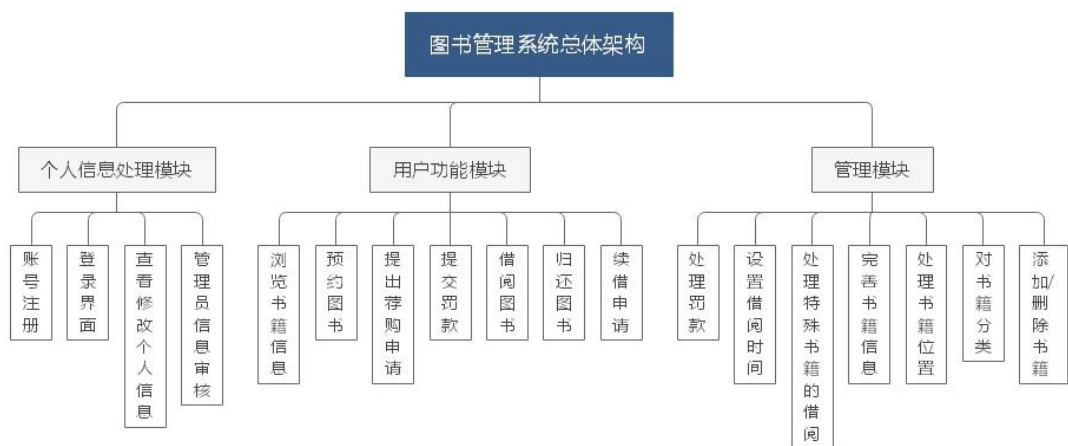
2.6 本章小结

本章主要介绍了系统的需求分析，包括系统功能性需求,并通过使用用例图介绍了三个模块的具体功能，通过表格详细分析了用例图，并在最后详细说明通常情况下的进校流程。

第三章 系统设计

3.1 总体系统功能设计

本系统为图书管理系统，新用户通过填写个人信息登陆界面之后给管理员审核（老用户直接验证身份后可以进入）。完成认证后可以进行一系列操作，包括预约借书，申请续借，查看书籍信息，提交罚款等。管理员则对于用户的借阅情况对用户的信息、操作以及图书的信息进行管理。

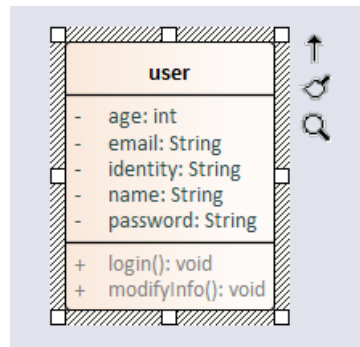


如图，大致可将系统分为三大块：个人信息处理模块包括用户的注册与登录，个人信息的完善等等以及管理员对用户的信息处理；功能模块包括用户的借阅功能（预约，续借，提交罚款等）以及管理员的处理罚款等操作；书籍管理模块主要包括管理员对于书籍信息的维护。

信息处理模块

（1）这里为用户提供一个进入系统的界面用户完成信息的验证后进入。如果用户第一次登录时，需要完成基本的信息填写方可进入。验证失败的用户不得进入。

（2）用户进入系统后，可以修改个人信息。修改后的信息经由管理员审核后存入数据库。



用户功能模块

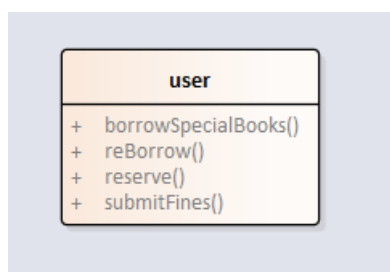
用户登录成功后会进入功能界面，这是系统的主要界面。

用户功能模块是图书管理系统的主要模块，主要内容为用户的功能：申请预约、借阅、续借、归还、提交罚款等。

(a) 用户预约书籍：用户首先检查自己有没有尚未提交的罚款，如果有先提交罚款；之后选择时间预约书籍，没有与当前数据冲突(已经存在的预约时间)则提示预约成功，否则预约失败。

(b) 用户申请珍贵书籍的借阅：首先会发送请求进行身份的核实，核实标准(identity)为是否为大学生或是老师，管理员处理信息后返回结果；其次检查信用情况，同时如果有罚款未提交应先提交罚款；身份满足要求后，选择预约的时间，时间如果没有冲突的话借阅成功。

(c) 用户续借图书：用户发送续借的请求；管理员首先检查是否超期，超期则提醒先提交罚款；设置当前时间为新的借阅时间，重新计算应归还日期。



书籍管理模块

该模块主要是由管理员用来实现功能：对用户请求的操作响应与对书籍进行的管理。对于用户的身份进行了限制（管理员）。

处理用户借阅珍贵图书的请求：核实标准(identity)为是否为大学生或是老

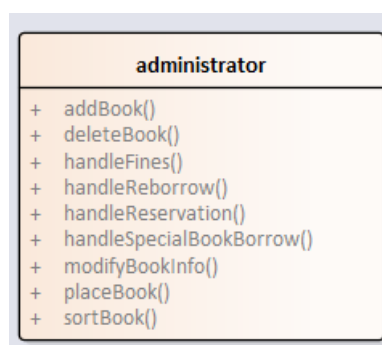
师，管理员处理信息后返回结果；其次检查信用情况，同时如果有罚款未提交提示先交罚款；身份满足要求后，处理申请预约的时间，时间如果没有冲突的话借阅成功。

(b) 处理续借的请求：管理员首先检查是否超期，超期则提醒先提交罚款；设置当前时间为新的借阅时间，重新计算应归还日期。

(c) 修改书籍的信息：管理员对书籍的信息（分类、位置、编号）进行修改。

(d) 处理书籍的位置：改变书籍的位置（房间号+书架号）。

(e) 对于书库中的书籍，进行增加/删除等工作。



3.2 体系结构设计

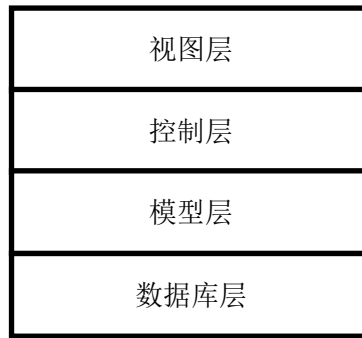
3.2.1 基本体系结构设计

该系统的适用场景为一个校园的图书馆，对象为该校园的师生及其他工作人员。所需要的功能包括借还书、荐购、预约等一般的图书馆所具有功能，方便图书馆管理员的管理以及师生对图书馆的利用。

本系统主要采用分层次结构进行架构。该结构有以下特点：升级和维护方式简单，对功能的修改便捷，减少维护成本。支持扩展与重用。

因为系统同时使用人数较少，信息库中存储的信息都为纯文本形式。考虑成本问题以及前期对需求的分析，决定基于 Qt 架构与 SQLite 来构建该分层次图书管理系统。

从上到下依次为视图层、控制器层、模型层和数据库层。如下图所示：



视图层：提供图书馆系统的基本视图显示，与用户的交互界面，将用户对应用程序的操作请求下传给其他层处理。为用户显示模型信息。

控制层：按预定的业务逻辑处理视图层提交的请求。

模型层：将模型的状态转交视图层或者控制层，以更新视图或控制信息，提供简化的 SQL 语句，及时更新修改数据库数据，提供数据持久化。

数据库层：用户信息数据库的建立与管理。

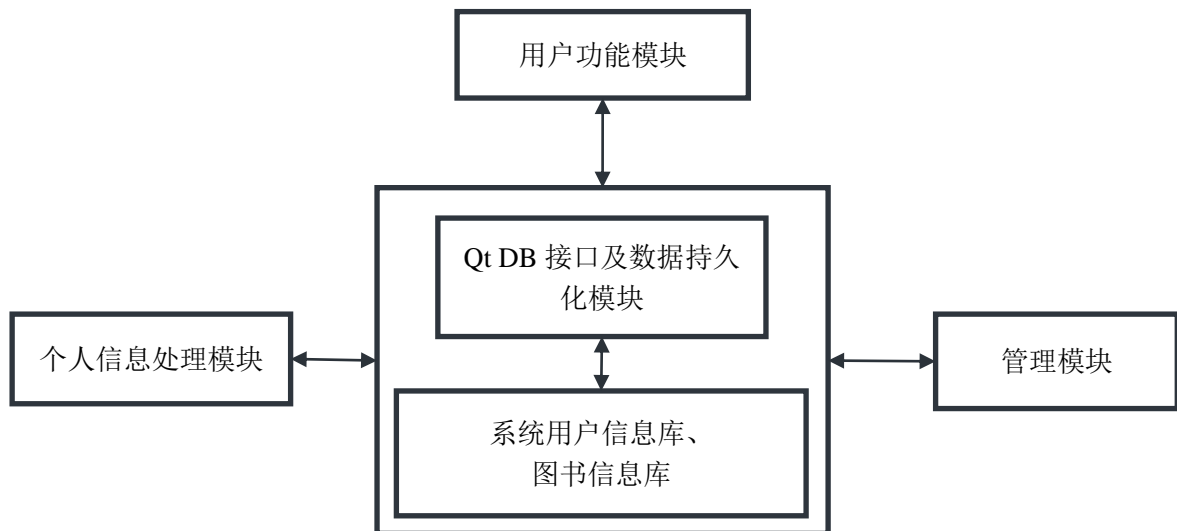
3.2.2 质量属性的满足

目标	实现方式	所采用战术
安全性	系统必须限制用户对数据的访问范围，对于没有权限的用户防止其对非自己可见的信息的访问，系统还应提供必须的身份检测与授权手段。	身份验证 身份授权 数据机密性
可用性	对于用户的非法访问与操作，需要及时制止防止对系统的损害，系统还应具有一定的故障恢复能力。	异常检测及程序终止 故障恢复性
可维护性	系统实现时分层实现，控制不同层次之间的交流，将错误一定程度限制在本层中。	分层次
可移植性	采用跨平台框架 Qt 对系统进行架构，减少了系统移植的成本。	跨平台框架

3.2.3 模型层架构分析

因为本系统所采用的功能都与中心数据库相关，所以在模型层中通过 Qt DB 接口将所有功能模块与数据库之间采用仓库模式连接。以此方便模块的添加修改以及部分模块之间的数据共享。

模型层架构如下：

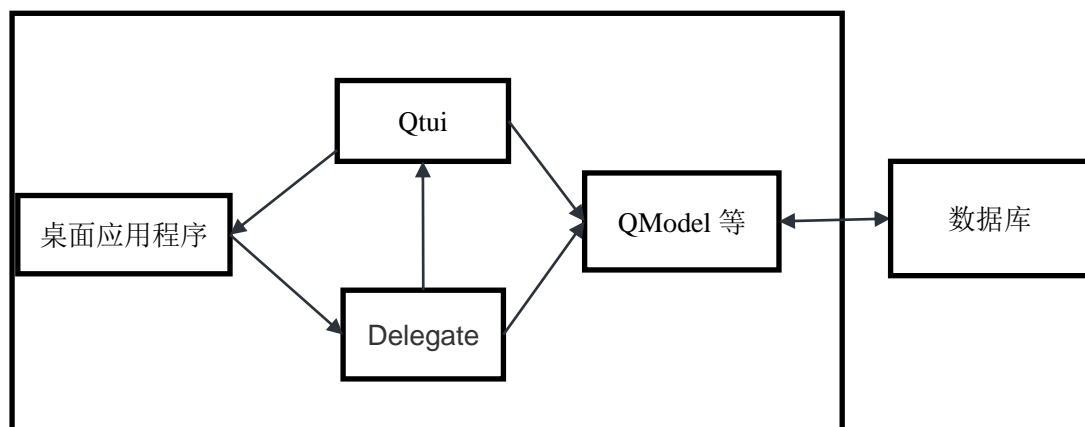


3.2.4 业务逻辑层架构分析

业务逻辑层决定了系统的实用性，根据 Qt 框架的特点，我们决定对业务逻辑层采用 MVC 风格进行架构。

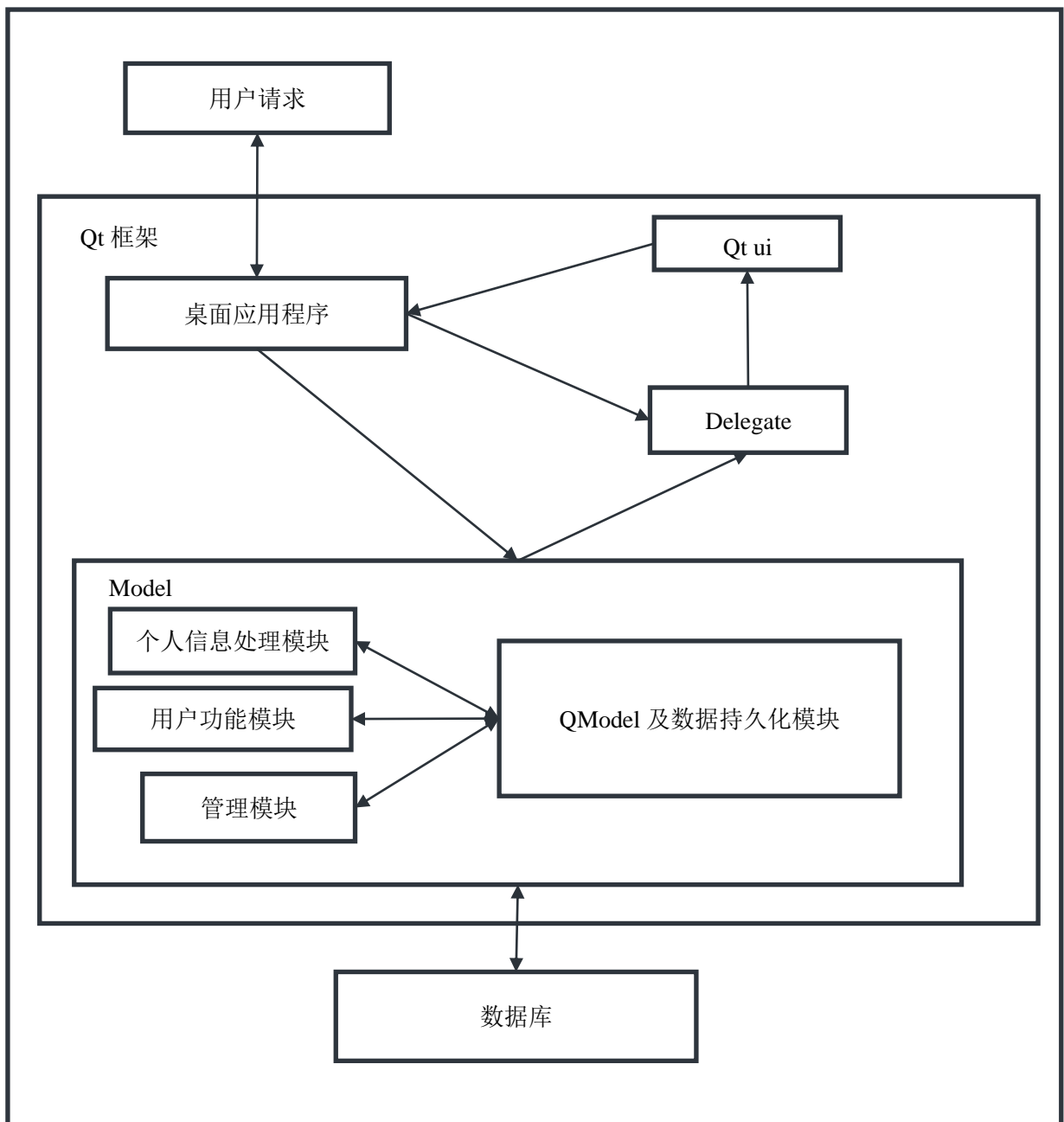
及当用户使用该系统的桌面应用程序时，在用户在 GUI 上发出的相关信号时，采用 Qt 默认的 Delegate 作控制器(为 Qt 预定义组件，实现时不用关注)，QModel 等为主的功能类作模型层，封装核心数据与部分功能。

业务逻辑层架构如下：



3.2.5 整体架构分析

由上述分析，整体架构如下：



3.3 接口设计

3.3.1 数据库接口

接口说明

服务器端通过调用函数来完成数据库的增删改查工作。服务端也能通过通信接口向服务器发出请求，并接受服务器的请求响应，从而实现借书还书等功能。

调用方式

服务器端与 SQLite 数据库通信接口:利用 Qt DB 与 SQLite 数据库进行通信,同时构建 SQLite 类用于与数据库交互。SQLite 类中根据不同的功能提供数据库信息的增加、删除、更改、查询功能。实际调用时,使用 SQLite 的单例化实例,传入相关数据,调用相应的 API 函数即可。

3.3.2 读者功能模块接口

接口说明

客户端通过调用固定函数来完成与服务器的信息交互,简化通信内容,提高通信效率。根据软件需求,有以下 9 个函数:注册、登录、查看罚款、缴纳罚款、查询藏书信息、借阅图书、续借图书、归还图书、提交推荐购书申请。

调用方式

客户端与服务器端通信接口:遵循网络协议,满足局域网的通信需求。同时通过约定请求的消息类型,用于区分不同请求的解析方式与调用不同的方法。

3.3.3 管理员模块接口

接口说明

客户端通过调用固定函数来完成与服务器的信息交互,简化通信内容,提高通信效率。图书处理接口有以下 5 个函数:书本分类、书库分配、图书查询、借阅记录更新、书本信息登记更新。用户管理接口有以下 4 个函数:处理用户预约、处理用户信用度、用户权限划分、处理用户赔偿。

调用方式

客户端与服务器端通信接口:遵循网络协议,满足局域网的通信需求。同时通过约定请求的消息类型,用于区分不同请求的解析方式与调用不同的方法。

3.4 本章小结

本章主要介绍了该系统的设计。首先从总体出发,描述的系统的总体模块,然后从三大模块详细分析各个模块的内容。在最后,并介绍了整体的数据库结构,以及一些比较重要的接口函数的介绍。

第四章 系统实现

4.1 类设计

(1) user 类：负责（借书、还书、预约、续借，推荐书籍），修改个人信息，交罚款。

(2) manager 类：查看 user 信息、同意或拒绝 user 的各种申请、进行图书的入库管理以及出库管理。

(3) book 类：存放书籍的所有信息包括（书名、作者、出版社、编码）。

(4) 流水（waterflow）表，又称事件类：存放 user 对 manager 的各种申请以及处理结果。

4.2 数据库表设计

以下是数据库表的详细设计表格：

表 4.1 Users 信息表

字段名	含义	数据类型	备注
RecNo	编号	Int	代理主键，自动增长
ID	学号	Int	代理主键，自动增长
Name	姓名	Varchar	
Username	用户名	Varchar	不可重复
Password	密码	Varchar	
Address	家庭地址	Varchar	
Sex	性别	Varchar	
Credit	信用分	Int	
Users	判断是否为管理员	BOOL	
Fine	累计罚款	Double	

表 4.2 Book 字段信息表

字段名	含义	数据类型	备注
RecNo	编号	Int	代理主键，自动增长
Barcode	书籍编码	QString	不能为空

Writer	作者	Varchar	不能为空
Price	价格	Double	不能为空
Num	库存	Int	
Press	出版社	Varchar	不能为空
Category	范畴	Varchar	

表 4.3 WaterFlow 字段信息表

字段名	含义	数据类型	备注
Operation	操作类型	Int	表明该事件类型
Booid	对哪一本书操作	Int	不能为空
Userid	哪一个用户申请操作	Int	不能为空
bDate	借书日期	Date	
rDate	还书日期	Date	
Day	借书时长	Int	
Ifhandle	是否管理员已处理	Bool	
Result	处理结果	Int	表明不同的处理结果
Recbook	书名	Varchar	

第五章 系统测试

5.1 注册及登录

程序的注册和登录界面如下：

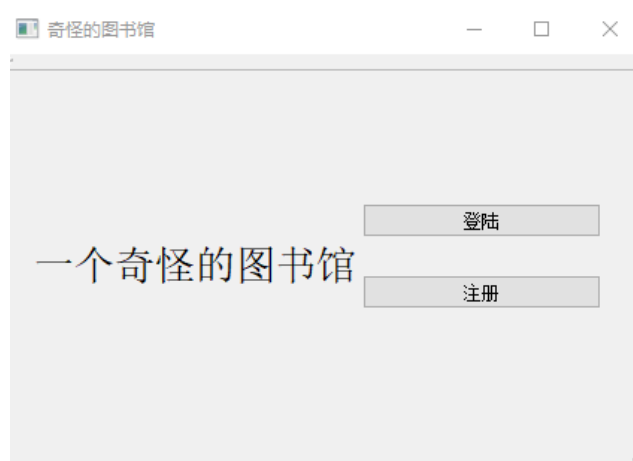


图 5.1.1 注册、登录界面

普通用户可以通过注册功能注册自己的账户，管理员不能注册，而由其他管理员直接在数据库中添加。测试用户注册功能，依次输入姓名、用户名、密码：



图 5.1.2 注册界面

点击注册按钮后，可见后台数据库成功添加了新的用户信息：

RecNo	id	name	username	password	address	sex	credit	users	fine
Click here to define a filter									
1	180000	管理人员1	manager	123123	(null)	(null)	(null)		(null)
2	180001	用户1	user	123	(null)	(null)	100		0

图 5. 1. 3 数据库在用于注册后的界面

用户通过登录，输入自己的用户名和密码，可以按预期成功登录系统。

5.2 用户模块测试

5.2.1 查看、修改个人信息

按预期，用户登录系统后，可以在个人中心查看自己的信息，包括 id、用户名、真实姓名、密码、地址、性别、信用值和当前的欠款，并能够编辑自己的真实姓名、密码、地址、性别。在图书馆主页面点击个人中心后，符合预期地可以满足以上操作，而且没有明显的延迟。

用户信息

id: 180001

用户名: user

真实姓名: 猪云天 [编辑]

密码: 123456 [编辑]

地址: 东南大学梅园XXX [编辑]

性别: 男 [编辑]

信誉: 100

身份: 1

欠款:

[保存] [取消]

图 5. 2. 1. 1 修改个人信息界面

5.2.2 浏览书本信息

测试该模块前，先在数据库中添加了三本书，分别为《百年孤独》、《皮囊》、《白夜行》。在用户登录图书馆后，按预期能够明显地看到图书馆新进的图书，

点击书名后能查看到书本的详细信息。经测试该模块符合预期。



图 5.2.2.1 查看新进书本信息界面（主界面）



图 5.2.2.2 查看书本详细信息界面

5.2.3 书本借阅、归还、续借

按预期，用户选择了书本后，若书本有库存，且未被他人预约，即可点击借阅按钮进行借阅，并能在个人中心中查看自己的借阅记录、应归还时间。用户在

个人中心中可以选择图书进行归还和续借申请，其中续借的申请需要管理员进行确认（在后续的管理员模块测试会进一步说明）。经测试上述预期可以得到满足。

测试借阅的书本为《百年孤独》、《皮囊》各一本，点击个人中心的图书管理界面可以看到自己所借书本的记录，并进行归还操作、申请续借。



图 5.2.3.1 查看已借阅书本详细信息界面

5.2.4 图书荐购

用户按预期能够向图书管理员推荐购买图书，经测试，用户提交了荐购建议后，数据库成功增加了流水（在后续的管理员模块测试会进一步说明）。

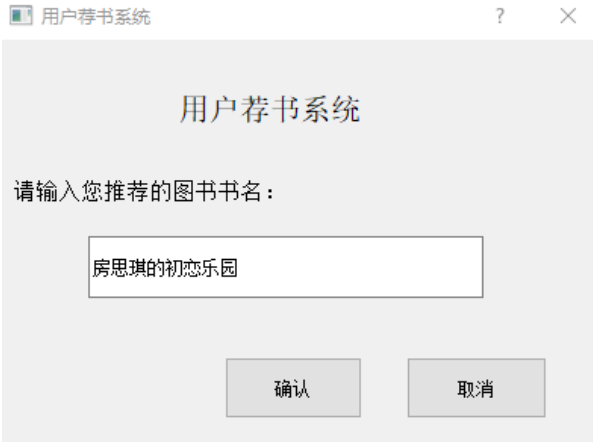


图 5.2.4.1 用户荐购界面

RecNo	operation	bookid	userid	bDate	rDate	dDate	day	ifhandle	result	recbook
Click here to define a filter										
1	1	177061	2	2020-04-12	(null)	2020-05-12	30	<input checked="" type="checkbox"/>	1	百年孤独
2	1	177061	2	2020-04-12	(null)	2020-05-12	30	<input checked="" type="checkbox"/>	1	百年孤独
3	1	177061	2	2020-04-12	(null)	2020-05-12	30	<input checked="" type="checkbox"/>	1	百年孤独
4	5	(null)	2	(null)	(null)	(null)	(null)	<input type="checkbox"/>	(null)	安徒生童话
5	4	177061	3	(null)	2020-04-12	(null)	(null)	<input checked="" type="checkbox"/>	1	百年孤独
6	4	177061	3	(null)	2020-04-12	(null)	(null)	<input checked="" type="checkbox"/>	1	百年孤独
7	4	177061	3	(null)	2020-04-12	(null)	(null)	<input checked="" type="checkbox"/>	1	百年孤独
8	4	177061	3	(null)	2020-04-12	(null)	(null)	<input checked="" type="checkbox"/>	1	百年孤独
9	4	177061	180001	(null)	2020-04-12	(null)	(null)	<input type="checkbox"/>	(null)	百年孤独
10	1	177062	180001	2020-04-12	(null)	2020-05-12	30	<input type="checkbox"/>	1	皮囊
11	3	177061	180001	(null)	(null)	(null)	30	<input type="checkbox"/>	(null)	百年孤独
12	3	177061	180001	(null)	(null)	(null)	30	<input type="checkbox"/>	(null)	百年孤独
13	3	177061	180001	(null)	(null)	(null)	30	<input type="checkbox"/>	(null)	百年孤独
14	4	177061	180001	(null)	2020-04-12	(null)	(null)	<input type="checkbox"/>	(null)	百年孤独
15	5	(null)	180001	(null)	(null)	(null)	(null)	<input type="checkbox"/>	(null)	房思琪的初恋乐园

图 5.2.4.2 数据库流水线表中新增了申请 (RecNo.15), operation=5 代表推荐操作

5.2.5 支付罚款

按预期,用户可以在登陆后支付罚款。经测试,在后台设置用户罚款为 10 元时,作为用户在点击支付罚款后,即弹出提示提交罚款成功,经查所欠罚款已减为 0。这个功能模块与实际生活的缴纳罚款不符,没有进一步的交付界面。

5.2.6 搜索功能

按预期,用户在搜索书本时,可以通过搜索书名、出版社、作者、id 号其中任意一条搜索到对应的书本,并且搜索系统支持模糊搜索。经测试,当想搜索《百年孤独》这本书时,通过搜索“孤独”、“百年孤独”、“南海出版社”、“加西亚马尔克斯”、“177061”,只有前两项搜索内容能正确所引到书本(但是在管理员搜索时却能满足上述所有预期,在管理员模块测试时会详细介绍)。故功能部分满足预期。



图 5.2.6.1 搜索“孤独”成功搜索到图书



图 5.2.6.1 搜索“南海出版社”搜索不到图书

5.3 管理员模块测试

5.3.1 搜索功能

按预期，管理员在搜索书本时，可以通过搜索书名、出版社、作者、id 号其中任意一条搜索到对应的书本，并且搜索系统支持模糊搜索。经测试，当想搜索《百年孤独》这本书时，通过搜索“孤独”、“百年孤独”、“南海出版社”、“加西亚马尔克斯”、“177061”，都能正确所引到书本，故功能满足预期。

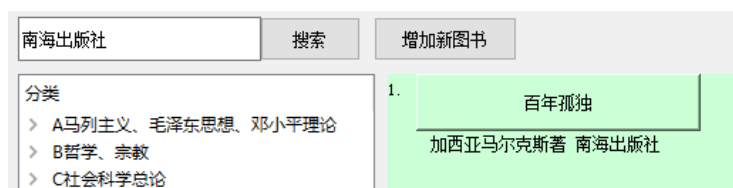


图 5.3.1.1 搜索“南海出版社”成功搜索到图书

5.3.2 图书添加、分类及图书信息修改

按预期，管理员添加新的图书入库，可以设置图书的书名、作者、价格、条形码、出版社、库存数量，及可以为书本进行分类。同时管理员也可以对已经添加的图书进行信息修改。经测试，上述预期可以成功得到满足。

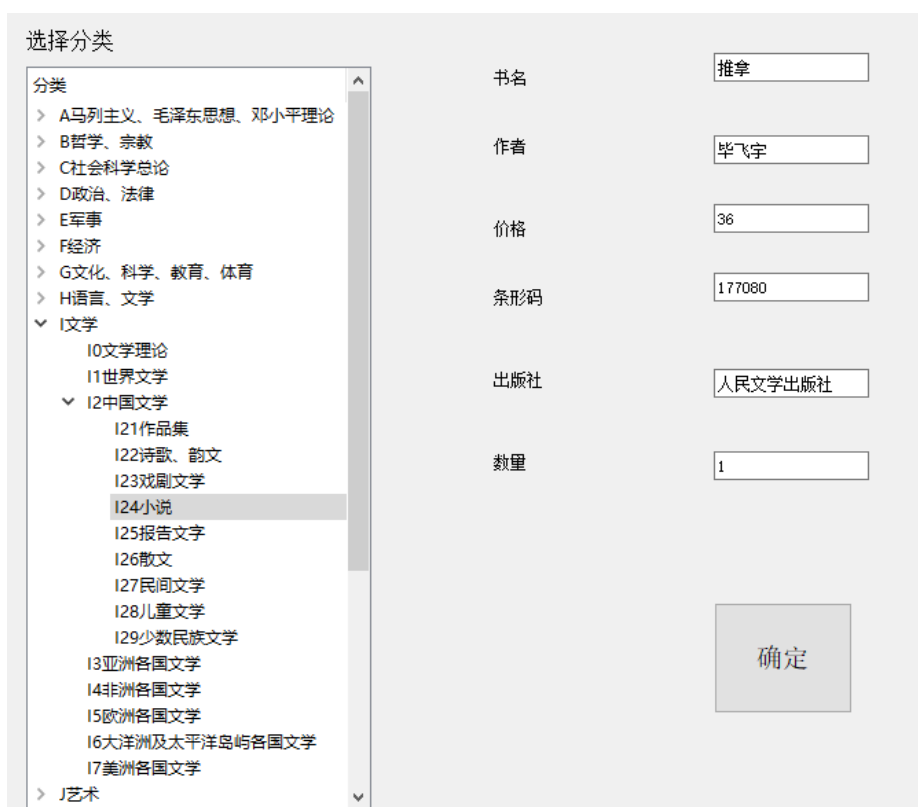


图 5.3.2.1 新添加图书《推拿》

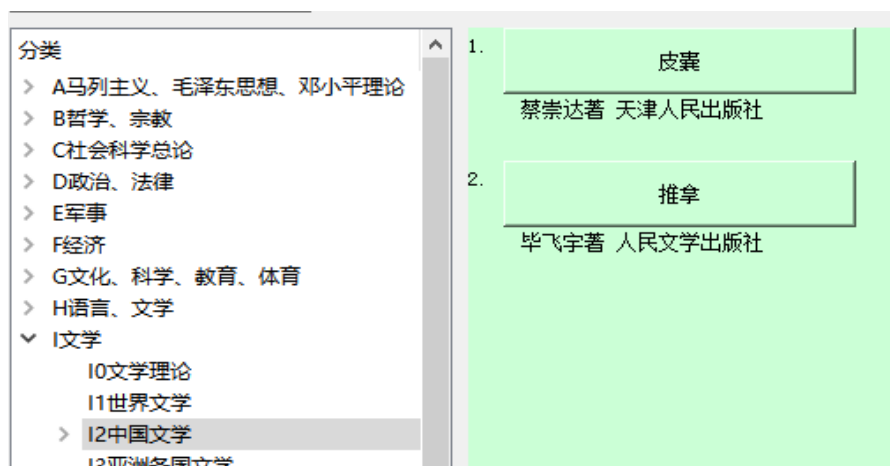


图 5.3.2.2 《推拿》成功添加

书名:	推拿
作者:	毕飞宇
条形码:	177080
价格:	40
出版社:	人民文学出版社
中图法分类:	I24小说
库存:	1
<input type="button" value="修改"/> <input type="button" value="确认"/>	

图 5.3.2.3 将《推拿》的价格修改为 40 元

RecNo	barcode	name	writer	price	num	press	category
Click here to define a filter							
1	177061	百年孤独	加西亚马尔克斯	49	4	南海出版社	I1世界文学
2	177062	皮囊	蔡崇达	47.7	2	天津人民出版社	I24小说
3	177063	白夜行	东野圭吾	59.6	4	南海出版公司	I3亚洲文学
4	177080	推拿	毕飞宇	40	1	人民文学出版社	I24小说

图 5.3.2.4 数据库中对应的条目同时进行了变动

5.3.3 用户信息管理

按预期，管理员能够看到当前注册的用户（包括所有管理员），并且可以查看他们的信息（包括用户名、性别、信用度、地址、罚款等），并且能够修改用户的罚款金额。经测试，上述功能可以得到实现。



图 5.3.3.1 查看注册的所有用户

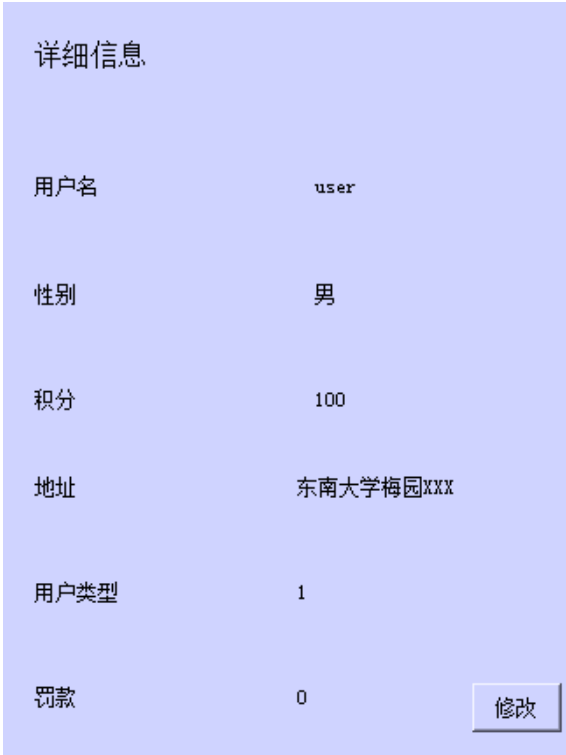


图 5.3.3.1 查看用户的详细信息

但是系统无法容易地获得确切用户的借阅记录。通过流水线形式，所有用户的借阅记录被放在同一个表中，难以统计单个用户的借阅记录：

	操作	书籍	用户	借阅日期	归还日期	截至归还日期	续借/续借时长	是否处理	处理结果	推荐书籍
1	借阅	百年孤独	2	2020-04-12		2020-05-12	30	已处理	同意	百年孤独
2	借阅	百年孤独	2	2020-04-12		2020-05-12	30	已处理	同意	百年孤独
3	借阅	百年孤独	2	2020-04-12		2020-05-12	30	已处理	同意	百年孤独
4	还书	百年孤独	3		2020-04-12			已处理	同意	百年孤独
5	还书	百年孤独	3		2020-04-12			已处理	同意	百年孤独
6	还书	百年孤独	3		2020-04-12			已处理	同意	百年孤独
7	还书	百年孤独	3		2020-04-12			已处理	同意	百年孤独

图 5.3.3.2 借阅记录表

5.3.4 管理员处理用户预约、续借、荐购请求

按预期，管理员可以对用户的续借、荐购请求予以处理。若同意用户的续借请求，则延长用户的借阅时长一个月，若拒绝用户的续借请求则通知用户。若同意用户的荐购请求则通知采购部门进行统一采购，在书本入库时新增书本信息，若拒绝用户的荐购请求则通知用户。

经测试，该系统给管理员提供了简单清晰的 UI 界面，管理员的同意与否只用点击一个按钮即可完成，并可以选择查看待处理和已经处理的请求：

序号	书籍	推荐数量	
1	房思琪的初恋乐园	1	<div>同意</div> <div>拒绝</div>
			<div>同意</div> <div>拒绝</div>

图 5.3.4.1 处理用户的荐购请求

序号	用户	书籍	借阅日期	截止归还日期	续借时长	操作
1.	180001	177061			30	<div>同意</div> <div>拒绝</div>
2.	180001	177061			30	<div>同意</div> <div>拒绝</div>
3.	180001	177061			30	<div>同意</div> <div>拒绝</div>

图 5.3.4.2 处理用户的续借请求

但是仍存在一些显示问题。例如在处理荐购请求页面中，会出现多余的确认按钮；在处理用户续借请求页面中，会出现书籍、日期无法显示，对齐不当等问题；在查看已经处理的请求页面中，有时候无法显示（10 次中有 5 次）已经处

理的操作记录。这些显示问题对功能的影响不大,可以在软件新版本更新时处理,处理优先级不是很高。

5.4 非功能需求满足测试

目标	预期	测试
性能需求	用户访问的系统应该能快速做出相应。若系统本身或数据库发生故障,应当给予用户明显的提示,最好的是将原因展示给用户看,避免用户无故地长时间等待,引起使用者不满。系统支持的并发访问量应不少于 1000。使用 SQLite 进行数据库管理,实现有组织地,动态地存储大量关联数据,并支持快速数据快速的增、删、改、查。	在对数据库进行的测试中(包括增加书本,更新流水线,一共 30 次操作),平均的反应延迟均在 1ms 内;在测试时出现过管理员无法添加图书的情况,Qt 报错无法访问数据库,但切换管理员后又正常运行(出现频率为 $\frac{1}{20}$),经检测代码的实现没有问题,猜想可能是 Qt 本身的问题。由于系统未采用 C/S 架构,处于单机状态,故无法满足并发访问。系统正确使用 SQLite 进行数据管理,效率高。
安全性	对于用户的非法访问与操作,需要及时制止防止对系统的损害,系统还应具有一定的故障恢复能力。	在系统提供给用户的 UI 界面显示中,并没有出现用户可以试图操作的“非法访问”。对于偶尔发生的故障(例如管理员无法添加图书),只能通过重启系统或其他方式解决,系统自我修复能力低。
可用性	系统可以随时进行访问,数据库 24 小时在线,出现故障时系统应该能够采用预设好的对应的处理方法。	系统可以实现长期启动的模式,出现故障后可以采用预设的处理方法(例如重启)。
可测试性	系统的各个功能、各个部分可以容易获得单独测试的机会,并且可以提供给测试人员整体测试的方便性。	系统的模块分明,各个功能有明显的划分,便于测试和问题定位。
易用性	系统应当提供简洁、直接、友好、易操作的 UI 界面供用户使用,无需额外的教学说明。	系统的界面简洁明了,没有繁琐的功能以外的界面显示,便于用户、管理员的操作。

第六章 实验小结

本实验按照需求分析、架构设计、代码实现、测试的顺序进行，完整地开发了一个图书管理系统。虽然实验到这就结束了，但是在实验过程中有很多地方值得反思和总结。

首先是在需求分析阶段，某些需求的提出并没有很好地参考实际情况，考虑开发时候的困难。例如“预约通知”功能，按照需求的设想，在用户发现特定书本的库存为零时可以申请预约，当书本被归还时可以收到短信提醒，前来取书。但由于我们是通过 Qt 进行开发，系统平台实现实时感知、短信发放这些功能比较复杂，故最后的成品并没有成功地实现该功能。因此，在以后开发时的需求分析过程中，必须有机地将需求、开发成本（包括时间、人力）结合起来，避免出现想法好却无法实现的情况。

其次是架构的考虑与选择。刚开始我们一致认为应当采用仓库风格，即中心仓库是数据库，其他模块直接与其相连。但后来一位组员提出：由于我们是基于 Qt 开发，而依照 Qt 的标准框架开发必须采用分层风格，其中最下层是数据库层，我们应当在其上层的模型层中进行模块的开发，并与 QModel 及数据持久化相连。（详见第三章架构设计）。经过此次实验，我们明白架构风格的选择和软件开发的环境密切相关，不能单单从需求角度出发确定软件架构的风格。

在分工的时候我们也有些不恰当的安排。代码的实现过程中，出现了多人同时开发同一个模块的情况，即实现分工存在交集。这样安排导致了一些实现错误的产生，浪费了维护和修改的时间。故在下次代码实现的工作安排中，应当让不同的同学开发各自的功能模块，并提供自己模块的接口，让其他同学进行调用。这样就很大程度上将错误的产生缩小了影响范围。

在数据库表的实现时也有一些问题。我们的流水线表负责监管着借阅、预约、荐购等功能，这样的设计使得内聚过高，数据量的增加会使得性能的明显降低。故在数据库的设计时还有待优化。

再有，由于该次实验是网上交流合作，故对于一些特定问题的理解和处理不能得到很好的统一。实验刚开始时是采用 GitHub 进行合作的，但在进行了几十次 commit 后由于种种因素（学习时间成本、问题处理等）没能坚持到底，留下

了遗憾。我们希望在下一次线下实践中将节省的沟通成本用来学习 **GitHub** 协作，以提升自己的能力。

最后，在代码实现时，由于时间和网上协作的关系，各个部分在对接适合有些困难，虽然最后都大体解决，给下次对接提供了经验教训，但也花费了部分时间，导致某些模块存在一些非功能性的小问题（例如显示问题），**UI** 界面也不是很美观，没有进行进一步的美化，留下了遗憾。

总而言之，通过这次系统地进行软件开发，我们对需求分析、架构设计、实现和测试等方面有了更全新的认识和熟练，为下一次开发积攒了很多经验教训。