

《数据库系统原理》课程设计

系统实现报告

题目名称：图书馆管理系统

|  |  |
| --- | --- |
| 学 号： | 12061154 |
| 学生姓名： | 冯飘飘 |
| 学 号： | 12061153 |
| 学生姓名： | 刘丽萍 |

2014年 12 月

目录

[一、 实现环境 3](#_Toc407559741)

[二、 系统功能需求分析 3](#_Toc407559742)

[三、 系统功能结构设计 4](#_Toc407559743)

[四、 数据库基本表的定义 6](#_Toc407559744)

[五、 触发器的定义与实现 9](#_Toc407559745)

[六、 存储过程的定义与实现 11](#_Toc407559746)

[七、 系统实现结果 15](#_Toc407559747)

[八、 总结 33](#_Toc407559748)

# 实现环境

1. 数据库系统：MySql 5.5
2. 数据库前端管理工具：Navicat Premium
3. 实现语言：java
4. 实现环境：Elicpse Luna Service Release 1(4.4.1)

jre1.7.0

# 系统功能需求分析

经过对图书馆日常工作的调查研究及管理流程的分析，要求本系统具有以下功能：

1. 读者信息管理功能：读者对自己的相关信息进行管理，包括私人信息（电话邮箱等信息）和历史或现在借阅信息的查询，只能对私人信息进行更改。
2. 读者信息录入功能：读者进行注册，将自己的信息录入图书馆管理系统数据库，以便借阅图书等相关功能。
3. 图书信息管理功能：管理员对图书的相关信息进行管理，包括图书的名字、类型、所在阅览室等，可以增加图书信息、删除图书信息、修改图书信息、查询图书信息。
4. 图书借阅、归还功能：借阅图书时，管理员对图书索书号和读者ID建立相应的联系，添加进入借阅记录表中；归还图书时，管理员输入图书ID即可修改借阅记录表中的数据，添加该书的归还时间，表示该书已经归还。
5. 查询功能：方便读者和管理员查询各种信息。读者和管理员可以根据索书号、书名或者作者来查询需要的图书。管理员还可以查询所有图书和读者的信息列，以及所有借阅信息的记录。

建立一个图书馆管理系统，使图书馆的日常管理工作规范化、系统化、程序化，避免管理的随意性，提高信息处理的速度和准确性，能够及时、准确、有效地查询图书借还情况。

# 系统功能结构设计

图书馆管理系统

主界面

检索图书信息

登录

系统

注册读者信息

读者登录

管理员登录

个人信息查询与修改

历史借阅记录

查询

当前借阅记录

查询

检索图书信息

个人信息查询与修改

图书管理

系统

读者管理

系统

借阅管理

系统

图书管理系统

查看所有图书

增加图书

修改图书信息

删除图书

读者管理系统

查看所有读者

增加读者

修改读者信息

删除读者

借阅信息管理系统

查看读者借阅信息

借书操作

还书操作

查看图书借阅情况

# 数据库基本表的定义

* **tb\_book(bid,bookname,typeID,author,ISBN,bookcase,storage)**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 字段名 | 类型 | 长度 | 默认值 | 是否主键 | 描述 |
| bid | varchar | 5 | Null | yes | 索书号 |
| bookname | varchar | 100 | Null |  | 书名 |
| typeID | varchar | 30 | Null |  | 类型 |
| author | varchar | 30 | Null |  | 作者 |
| ISBN | varchar | 30 | Null |  | 出版社 |
| bookcase | varchar | 30 | Null |  | 所在书架 |
| storage | int | 1 | Null |  | 库存 |

* **tb\_reader(rid,rname,rpwd,gender,tel,email)**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 字段名 | 类型 | 长度 | 默认值 | 是否主键 | 描述 |
| rid | varchar | 10 | Null | Yes | 读者ID |
| rname | varchar | 30 | Null |  | 读者姓名 |
| rpwd | varchar | 10 | Null |  | 读者密码 |
| gender | varchar | 10 | Null |  | 读者性别 |
| tel | varchar | 11 | Null |  | 读者电话 |
| email | varchar | 30 | Null |  | 读者邮箱 |

* **tb\_manager(sid,sname,spwd)**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 字段名 | 类型 | 长度 | 默认值 | 是否主键 | 描述 |
| sid | varchar | 10 | Null | yes | 管理员ID |
| sname | varchar | 30 | Null |  | 管理员姓名 |
| spwd | varchar | 6 | Null |  | 管理员密码 |

* **tb\_publishing(ISBN,pubname)**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 字段名 | 类型 | 长度 | 默认值 | 是否主键 | 描述 |
| ISBN | varchar | 10 | Null | Yes | 出版社号 |
| pubname | varchar | 30 | Null |  | 出版社名称 |

* **tb\_booktype(typeID,typename)**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 字段名 | 类型 | 长度 | 默认值 | 是否主键 | 描述 |
| typeID | varchar | 30 | Null | Yes | 类型ID |
| typename | varchar | 50 | Null |  | 类型名 |

* **tb\_bookcase(ID,name)**

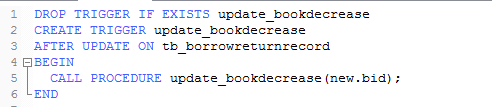
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 字段名 | 类型 | 长度 | 默认值 | 是否主键 | 描述 |
| ID | varchar | 10 | Null | Yes | 书架ID |
| name | varchar |  | Null |  | 书架名 |

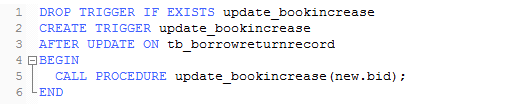
* **tb\_borrowreturnrecord(bookid,readerid,borrowdate,bookname,returndate)**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 字段名 | 类型 | 长度 | 默认值 | 是否主键 | 描述 |
| bookid | varchar | 5 | Null | bookid和borrowdate组合成主键 | 索书号 |
| bookname | varchar | 100 | Null | 图书名 |
| readerid | varchar | 5 | Null | 读者ID |
| borrowdate | varchar | 30 | Null | 借阅时间 |
| returndate | varchar | 30 | Null | 归还时间 |

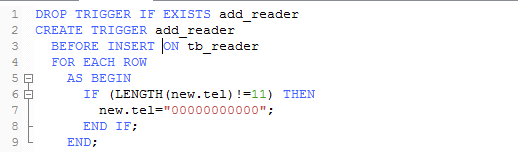
# 触发器的定义与实现

**触发器：**

1. **update-bookdecrease :** 
2. **update\_bookincrease:**



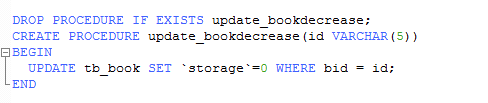
1. **add\_reader:**



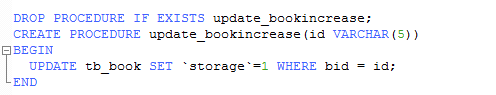
|  |  |
| --- | --- |
| 触发器 | 触发器说明 |
| update-bookdecrease | 每当有借阅的信息添加后，都要把相应图书的库存量修改一下，变成0 |
| update\_bookincrease | 每当还书来时，都把相应图书的库存量修改为1 |
| add\_reader | 每当用户的电话不足11位时，默认读者输入错误的电话，修改为十一位0 |

# 存储过程的定义与实现

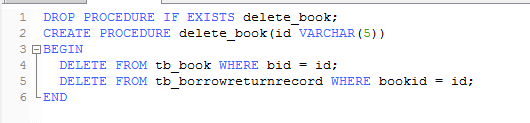
1. **update-bookdecrease：**



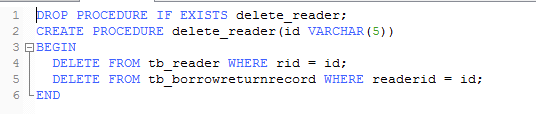
1. **update\_bookincrease：**



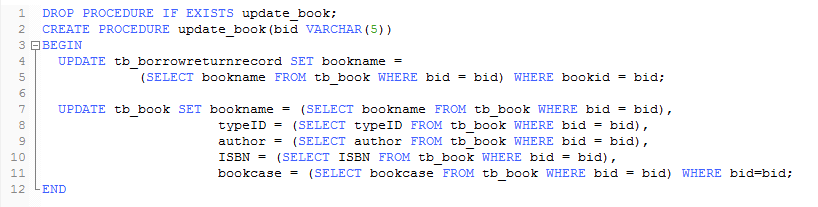
1. **delete\_book：**



1. **delete\_reader：**

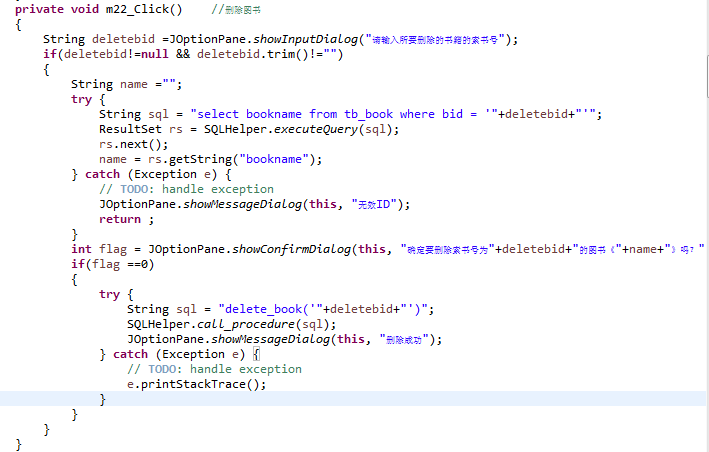


1. **update\_book：**



|  |  |
| --- | --- |
| 存储过程 | 存储过程说明 |
| update-bookdecrease | 每当有借阅的信息添加后，都要把相应图书的库存量修改一下，变成0 |
| update\_bookincrease | 每当还书来时，都把相应图书的库存量修改为1 |
| delete\_book | 若有图书要被删除，相应的图书表和借阅记录中的图书信息也被删除 |
| delete\_reader | 若有读者要被删除，相应的读者表和借阅记录中的读者信息也被删除 |
| update\_book | 若有图书名字被修改后，相应借阅记录中图书的名字也被修改 |

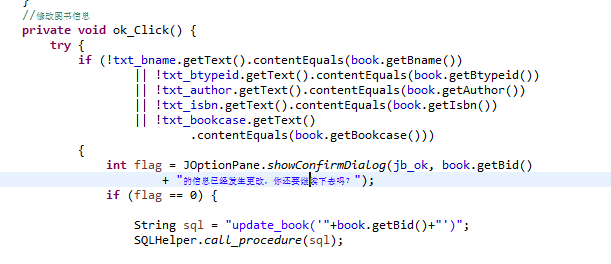
1. **delete\_book存储过程的实现：**



1. **delete\_reader存储过程的实现**



1. **update\_book存储过程的实现:**



# 系统实现结果

程序运行弹出初始界面：



选择搜索，进入搜索页面：

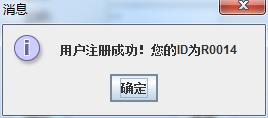
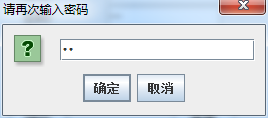


选择登录进入登陆界面：



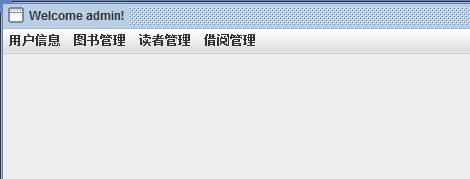
选择注册，进入注册界面输入信息：



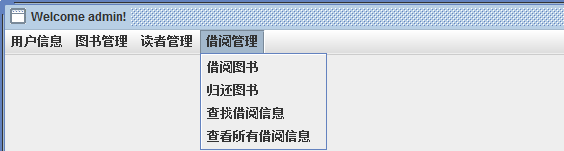
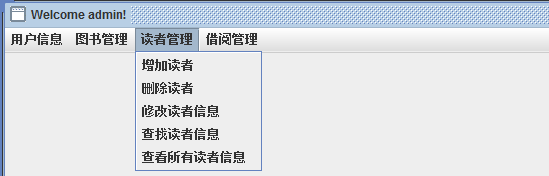
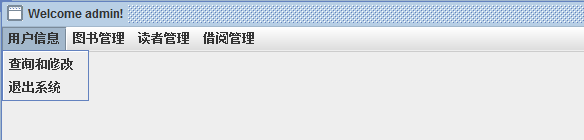


选择管理员进行登陆，进入如下界面：

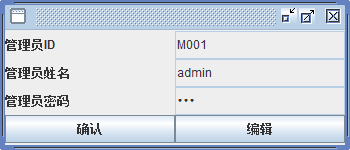


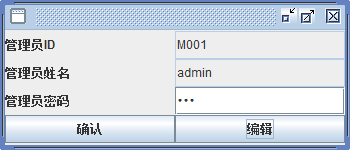


管理员可以选择自己想要管理的项目：

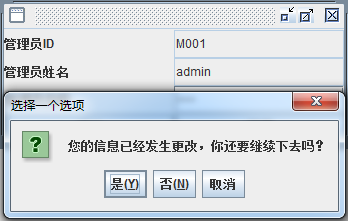


用户信息查询与修改：

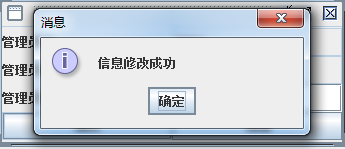
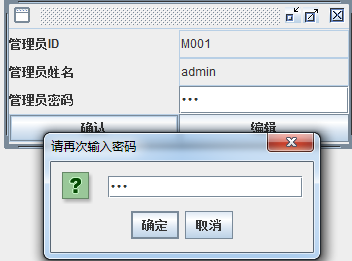




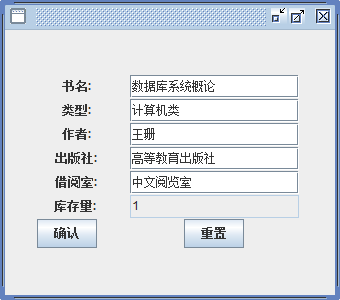
修改密码后：



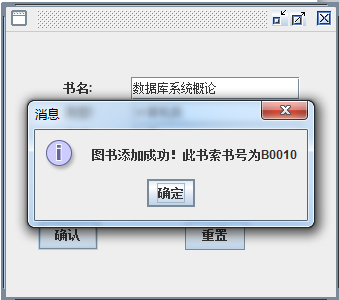
选择”是”：



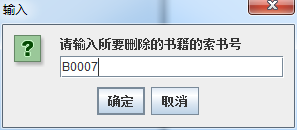
添加书籍相关信息：（不可为空）



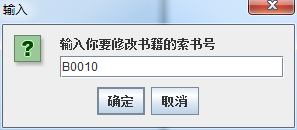
添加成功，系统自动分配索书号B0010



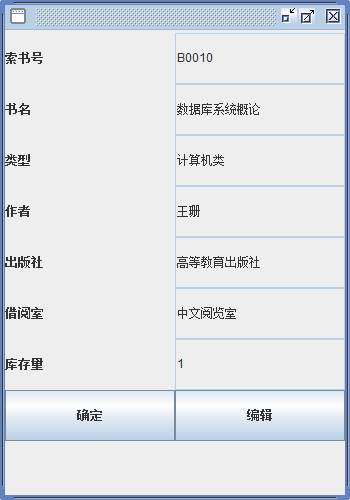
删除指定图书：



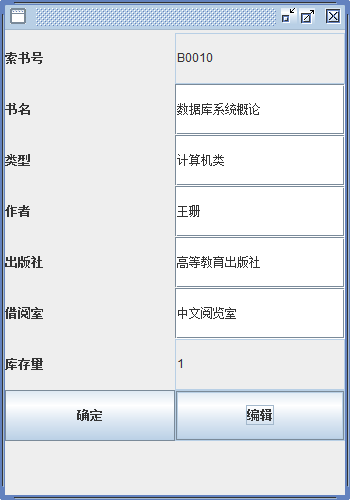
修改指定图书：



点击”确定”：



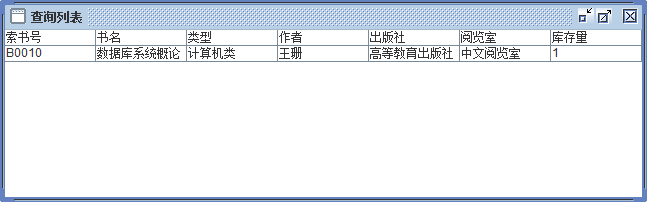
点击”编辑”：



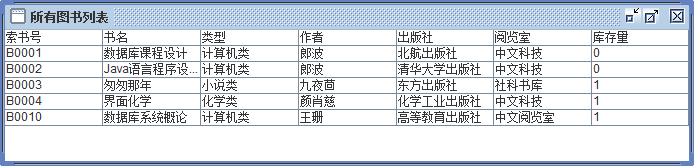
查询所需图书：



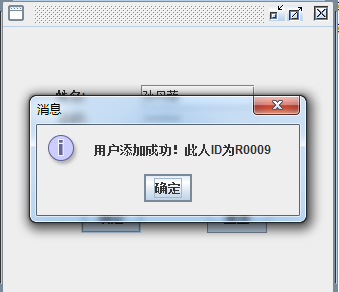
库存量为1表示可借，为0表示已经被借走尚未归还



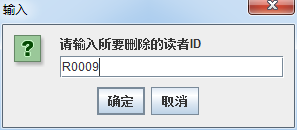
查看所有图书信息：



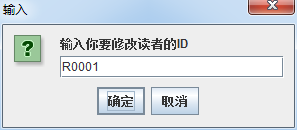
添加读者：（默认密码为00000）



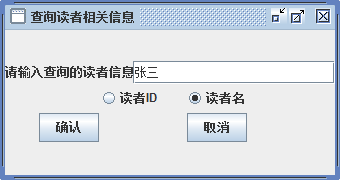
删除读者信息：

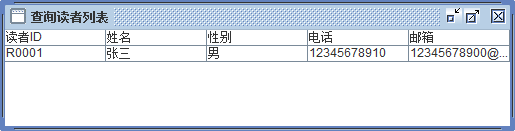


修改读者信息：

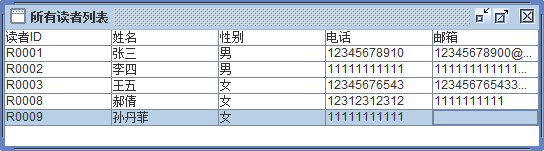


查询读者信息：

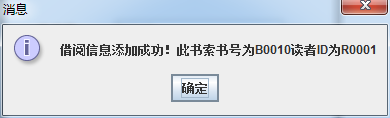
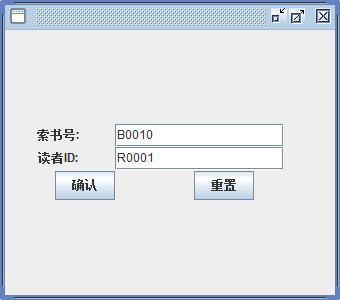




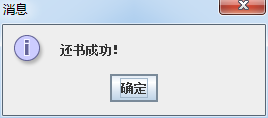
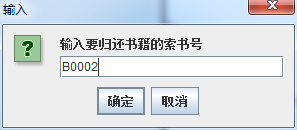
查询所有读者信息：



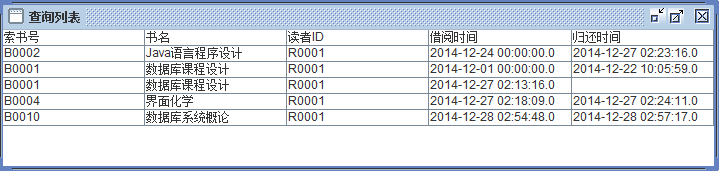
添加借阅信息（借阅时间自动添加为当前时间）：



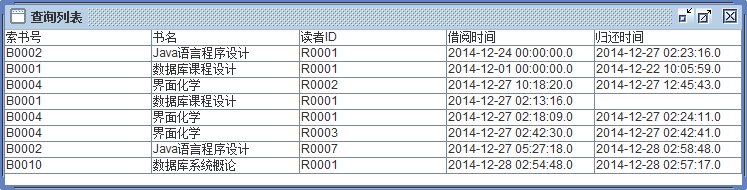
归还图书：



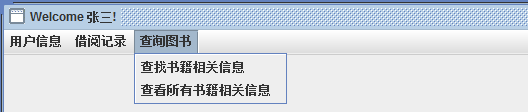
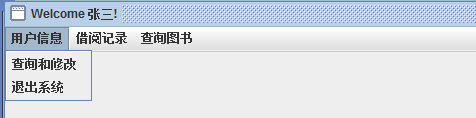
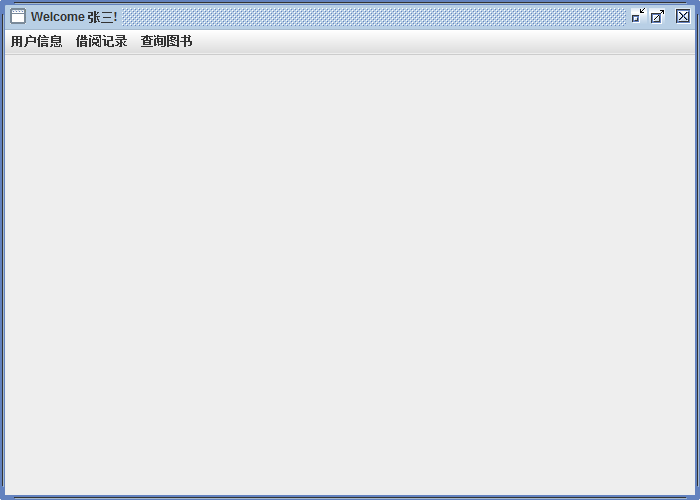
查询借阅信息：



查询所有借阅信息：



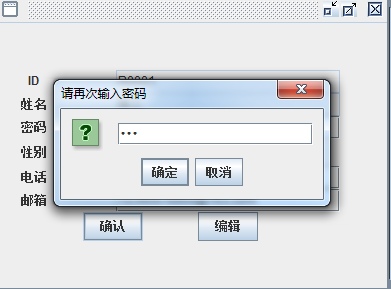
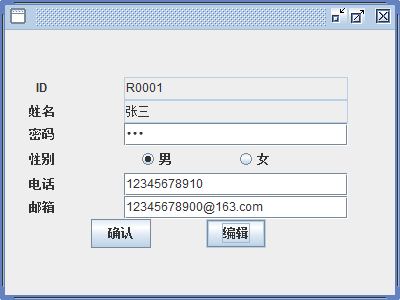
用户界面：



用户信息查询：



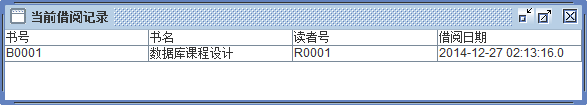
信息修改：



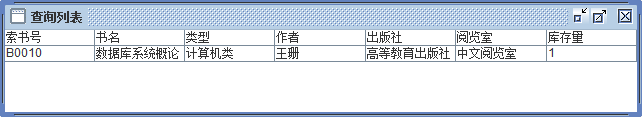
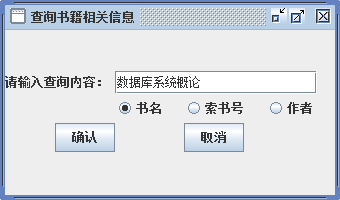
个人借阅历史记录查询：



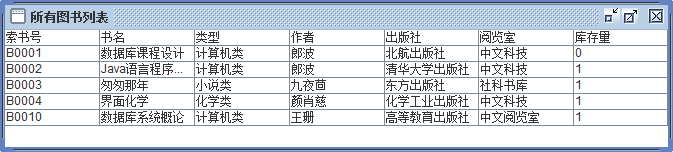
个人当前借阅记录查询：



书籍搜索功能：



查询所有图书：



# 总结

在完全投入于数据库课设的这前前后后一个星期里，有经验有教训。两人合作的形式，使得队友之间有赞赏有分歧，就整个过程来说，从最初的选题到系统实现到写文档，皆是两人合作完成，期间的一点一滴，对我们今后的编程之路都有着重要的启示作用。

首先，在最初的选题上，原以为图书馆管理系统算是较为简单的一个系统，实现过程并不会太复杂，我们可以将其做得更为精致，所以选题时并没有过多得考虑实现过程，只是单纯地想要凑实体，因此加了图书分馆、权限之类的实体，但在后期实现时才发现，实体跨度太大，管理过程过于复杂，若要按现实情况实现，则对我们来说还有些难度，因此只能将实体简化，管理过程略简单化，即删去图书馆分馆这个实体，即只针对一个图书馆，不可跨馆借阅，以及管理员不分级别、类型，统一管理图书和读者。

其次，在选题到实现的时间上，我们两个人都有一些拖延，老师十分良苦用心地在第八周就让我们确定了选题，以便能够早些完成，说来也羞愧，我们每周都商量着要抽时间做编译课设，但表面上因为其他事情耽搁了，实则心中没有紧迫感，总想着时间还早，但事实证明，越拖越忙。到最后的一周，只能昼夜马不停蹄地赶进度。

再次，在系统实现期间，更是受到了前期考虑不周的教训，关于思考实现语言就想了2天，不过也并不是单纯的想，我们查阅了各种资料，尝试了jsp、php、C#等多种语言，但现学现卖的效果并不太好，最终还是决定用我们更熟悉的java语言，但图形界面对我们来说又是一大难题，因为我们俩之前从未写过java图形界面，且java的图形界面并不像C#或其他部分语言一样可以自动生成布局代码，我们靠着强大的自学能力以及仅存的一点想象力，一步一步摸索着写图形界面，当第一次看到系统图形界面运行起来的时候，不亚于第一次看到“Hello World！”时的惊喜和成就感，一路波折下来，不仅仅是两个人互相鼓励着共同前行，也更是我们的程序给了我们信心，系统初步成型时还有很多bug，但每次bug修复成功都给我们带来很大的成就感，因此整个的实现过程也还算是苦中作乐的，虽然最后的结果并没有达到我们预想的那么尽善尽美，在图形界面上还需要做进一步的优化，但数据库的功能实现上，效果还是非常理想的。

总体来说，虽然有这么多的困难和挫折，一部分是客观因素，但大部分是由于我们主观上的考虑不周以及性格上的拖延症，这正是我们要吸取教训的地方，不畏艰难，现在正是我们发现问题，为今后的编程之路奠基的阶段。

最后的最后，要感谢老师给了我们一个这么好的锻炼机会，同时也要互相感谢我们的小伙伴，每天起早贪黑，天微亮便一同去图书馆，至晚上关门再一起回宿舍，数据库课设的完成之路并不顺利，但我们互相鼓励，一起讨论每一步的工作细节并无界限过于分明的分工，一个人累了，另一个人也能随时接手其工作；对于工作的细节有分歧时，互相迁就；遇到瓶颈心情烦躁时，也互相包容，最终目标一致、步伐一致地顺利完成了数据库课设。