广西大学

**本科生课程设计**

|  |  |
| --- | --- |
| 课程名称 | 数据库系统原理课程设计 |
| 课程编号 | J1620036 |
| 学生姓名 | 龙凯 |
| 学号 | 1007300326 |
| 所在专业 | 计算机科学与技术 |
| 所在班级 | 计科102 |
| 指导教师 | 顾平 |
| 成绩 |  |
| 教师签字 | 2013年 月 日 |

课程设计时间：2012年 12 月 29 日

至2013年 1 月 13 日

**目 录**

[设计总说明 I](#_Toc345992562)

[1 需求分析 3](#_Toc345992563)

[2 概念结构设计 5](#_Toc345992564)

[2.1 实体关系图 5](#_Toc345992565)

[3 逻辑结构设计 8](#_Toc345992566)

[3.1 数据库表结构 8](#_Toc345992567)

[4 物理结构设计 11](#_Toc345992568)

[5 系统实现与测试 14](#_Toc345992569)

[5.1 类图的组织层次 15](#_Toc345992570)

[5.2 数据访问层接口（rose画出） 17](#_Toc345992571)

[5.3 对外服务接口 25](#_Toc345992572)

[5.4 以一个例子来具体描述功能的实现 30](#_Toc345992573)

[5.5 客户端试图展现 33](#_Toc345992574)

[5.6 后台管理 45](#_Toc345992575)

[6 优缺点自我评价 48](#_Toc345992576)

[7 参考（都来自互联网的文档） 49](#_Toc345992577)

# 设计总说明

1. **系统开发目的**

几年前的web2.0时代给我们带来的冲击还在扩大。紧接着的iPhone，Android

平板电脑的日益流行，使得我们可以用一种很轻松的手势，触摸，语言等方式与互联网交互。根据调查显示，移动设备的占有率会在几年后超过传统pc，Windows在2012年9月推出的最新Windows8系统也第一次把触控功能带到了Windows系统，这不可谓是一种趋势：互联网已经逐渐靠着移动终端发展。此外，HTML5的日益规范化，其在各种平台上都得到了较好的支持。这些现象，构成了本次系统的开发目的：打造一个移动客户端。

1. **设计内容**

设计一个**模仿广西大学图书馆**的**移动客户端**与一个使用MySQL数据库的服务端。

1. **设计要求**

要求本客户端实现的基本功能如下：

1：读者的登陆，注册。

2：读者的查询图书。

3：读者的预约图书。

4：读者的借阅图书。

5：读者的还书。

6：读者的借阅图书超期的罚款。

7：有合理的异常处理机制将出错友善的告诉读者出错在哪。

8：管理员对图书，用户的基本管理。

1. **设计平台**

**客户端**使用HTML5，JavaScript，css3加上jQuery Mobile搭建，除了可以在web浏览器上访问外，也可以转换为Android，ios，Windows phone等平台上的原生app应用。

**服务端**使用Java平台开发，使用MySQL作为底层数据库。除此之外，还应用了多种技术开发，所有的技术都是目前工业界成熟的，流行的，稳定的。

开发环境为Windows8，jdk1.7\_10，MySQL\_5.5.29,eclipse\_3.7

1. **技术路线**

本次系统服务端所用到的技术列在下面，由于使用maven作为项目管理，故直接把项目所用到的核心依赖及其版本号以maven定义的xml列在下面：

<java.version>1.7</java.version>

<spring.version>3.2.0.RELEASE</spring.version>

<freemarker.version>2.3.19</freemarker.version>

<hibernate.version>4.1.9.Final</hibernate.version>

<logback.version>1.0.9</logback.version>

<servlet-api.version>3.0.1</servlet-api.version>

<mysql.jdbc.version>5.1.22</mysql.jdbc.version>

<json.version>20090211</json.version>

<jetty.version>8.1.8.v20121106</jetty.version>

其中，java使用的是jdk7的最新版本，使用spring提供的依赖注入，控制反转以及数据库的声明式事务管理给应用程序提供更好的灵活性与解耦性与安全性。此外，web层也使用了spring mvc提供的mvc机制来处理用户的请求与返回相应的视图，对于视图层，使用了freemarker的模版引擎技术来动态的生成视图文件返回到客户端，除了单一的请求，应用也提供了能够Json数据的Ajax的请求方式，使得应用程序更简单，更高效。

另外，本次数据库课程设计的核心，数据层由于一些原因被弱化了（原因见后面），所以使用了hibernate的orm技术，对象关系映射来处理数据库的访问，在比较大的程度上简化了数据库的代码量（当然，功能是实现了的），spring的事务管理也得到非常好的应用。MySQL数据库无须多说，jdbc也是最新版本的。

Webserver使用jetty嵌入式服务器，轻量型，但是非常强大。

至于本次系统的客户端，使用的事最新的HTML5，css3，与JavaScript，这些也都无需赘言。

整个项目使用非常流行和成熟的maven管理，开发，构建，测试，集成，部署一应俱全，非常的方便。

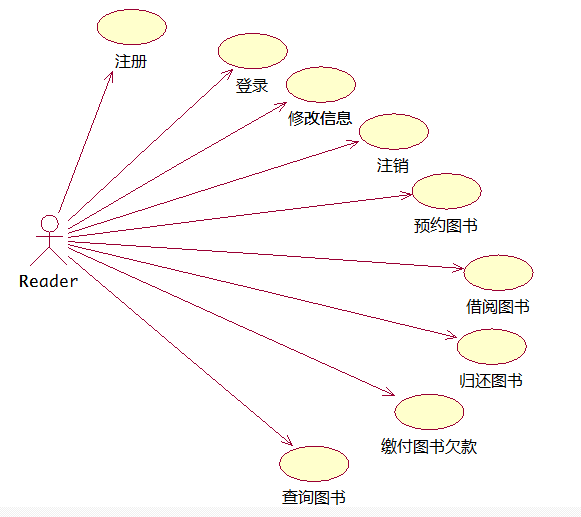
数据库系统原理教学系统设计报告

指导教师：顾平

# 需求分析

本次课程设计的设计内容是设计一个关于数据库系统原理的图书馆系统。通过自己在西大图书馆的经历，模拟出其应有的功能，用rose画出的用例图如下：

当读者想要借阅的书已经全部被借走时，用户可以预约这本书，当该书归还后，用户可以有优先权去借阅。预约有效时间为30天。过期不侯，并且当有多个读者共同预约同一本书时，依照先到先得的策略分配借阅优先权。用户有效的最大预约书本数量为10本。并且有欠款未缴付的不予预约。

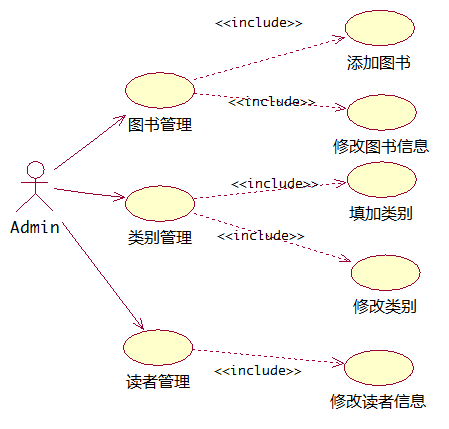


详细说明如下：

1. 注册：用户只有经过注册认证后才能享有图书馆提供的服务，注册必须提供正确的个人信息，否则不予通过注册。
2. 登陆：注册认证后的用户唯有经过登陆认证后才能在网上享受服务。
3. 修改信息：登陆后的读者可以更新自己个人信息。
4. 注销：当用户想要安全退出时，提供这个方法保护用户的个人信息不被他人窃取。
5. 预约图书：当读者想要借阅的书已经全部被借走时，用户可以预约这本书，当该书归还后，用户可以有优先权去借阅。预约有效时间为30天。过期不侯，并且当有多个读者共同预约同一本书时，依照先到先得的策略分配借阅优先权。用户有效的最大预约书本数量为10本。并且有欠款未缴付的不予预约。
6. 借阅图书：认证的读者可以借阅图书。最长的借阅时间为30天，每位读者最大借阅数量为10本。超过之后予以罚款，每本书每天0.1元。借阅的规则如下：1，若读者有欠款未缴付，不予；2，若读者达到了借书数量最大值，不予；3，若该书被预约并且该读者未预约，不予
7. 归还图书：读者借阅的书籍必须按时归还，否者将会按照每本书每日0.1元处以罚款。更严重的，将加以法律追究！书本归还之后，将对该书的预约信息进行更新。
8. 缴付图书欠款：当读者由于超出还书期限时，图书馆将冻结其一些操作。读者惟有缴付欠款才能得到相应的服务。
9. 读者可以对图书进行查询，按关键字，复杂，或者简单，然后图书馆返回相应的结果。

10）查询图书：按关键字，复杂，或者简单，然后图书馆返回相应的结果。

此外，还设置了一些基本的管理员的功能，由于重点没有放在管理员处，故这里的用例较少也较为简单。管理员用例如下：



详细说明如下：

1）：图书管理：简单地获取图书列表（含分页）,查看图书信息，修改图书信息，以及添加图书。

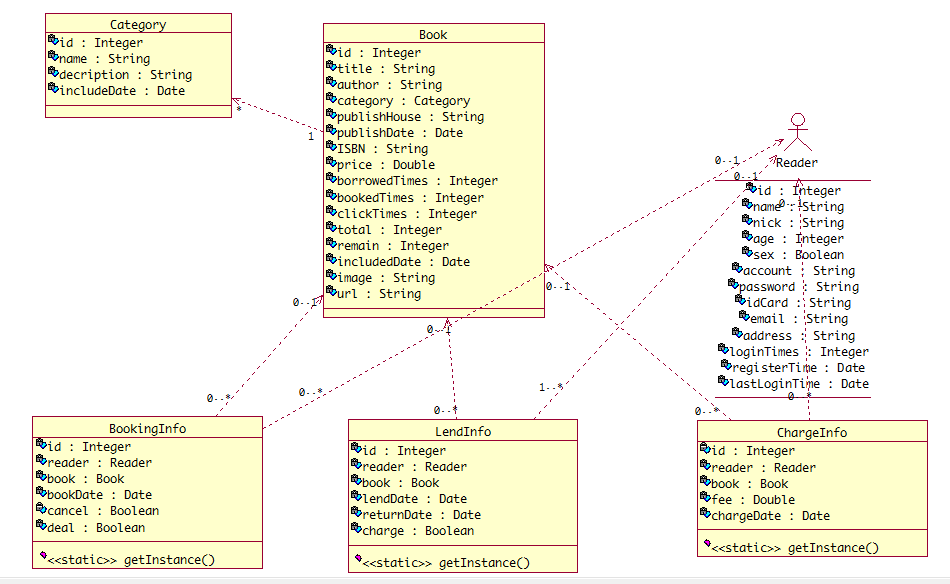
2）：类别管理：简单地获取类别列表（含分页），查看类别信息，修改类别信息，以及添加类别。

3：）读者管理：简单地获取读者列表（含分页）,查看读者信息。

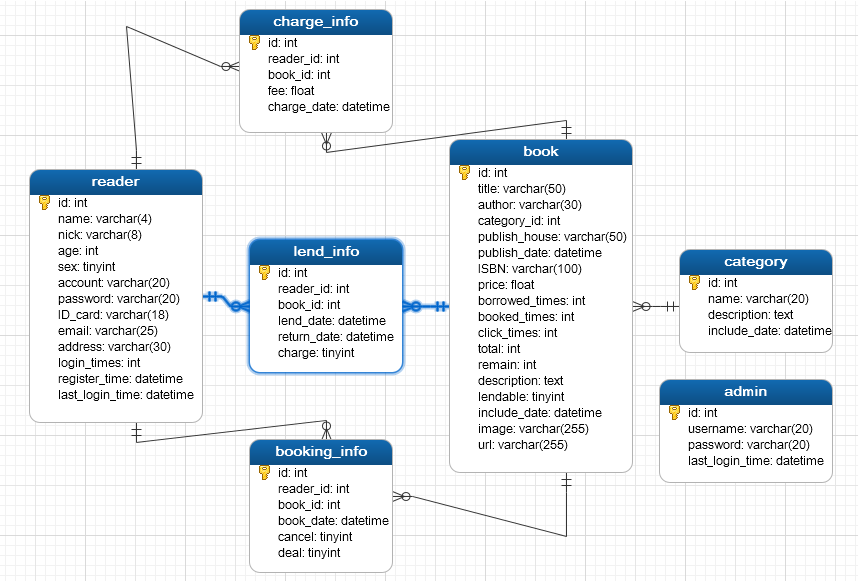
# 概念结构设计

## 实体关系图

1. 实体关系类图（用rose画出）



1. 数据库表结构关系图（用navicat画出）



1. 关系模型，各个表（实体类）的说明

Category(类别)：id（主键，自增），name，description，includeDate（录入日期）

Book（图书）：id（主键，自增），title，author，category\_id（外键，关联Category 的id字段，多对一关系），publishHouse，publishDate，ISBN，price，borrowedTimes，bookedTimes，clickTimes，clickTimes，remain（剩余图书数量），description，lendable（是否可借，默认true），includedDate（录入日期），image，url（链接信息）

Reader（读者）：id（主键，自增），name，nick，age，sex，account，password，idCard，email，address，loginTimes（登陆次数），registerTime（注册日期），lastLoginTime（上次登录时间）。

BookingInfo（预约信息）：id（主键，自增），reader\_id（预约的读者，外键，关联读者的主键id），book\_id(预约的图书，外键，关联图书的主键id)，bookDate,

cancel(是否取消), deal（预约是否成功）

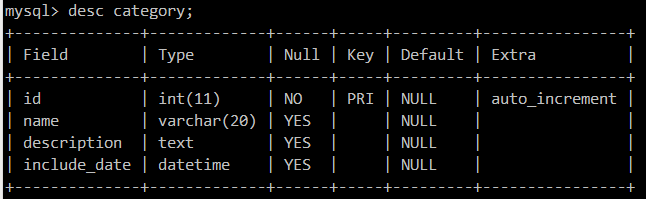
LendInfo(借书信息)：id（主键，自增），reader\_id（借书读者，外键，关键读者的主键id），book\_id（借阅的图书,外键，关联图书的主键id），lend\_date，returnDate（还书日期，默认为空），charge（是否有罚款，默认为false）

Admin（管理员）：id（主键，自增），username，password，last\_login\_time。

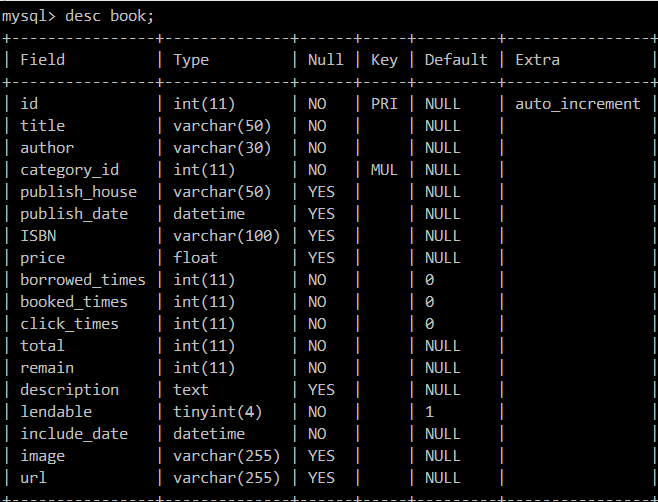
# 逻辑结构设计

## 数据库表结构

（1）Category



（2）Book

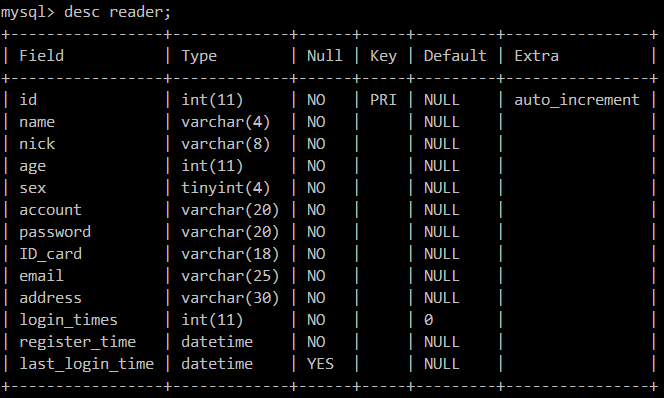


（3）Reader

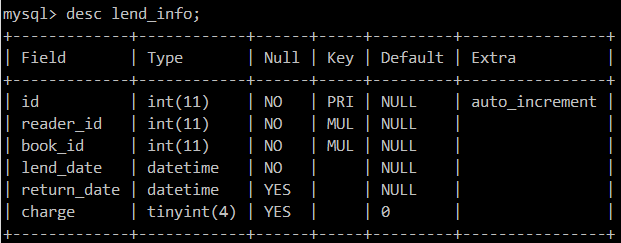
主键：ppt\_Id

外键：ppt\_ItemId 参照ppt \_Item的ppt \_ItemId；

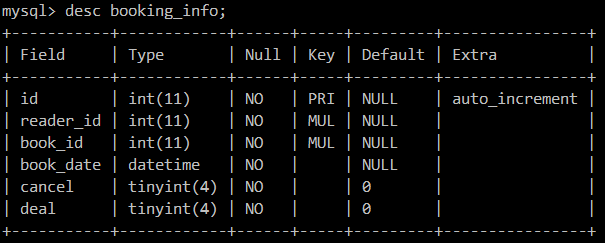
Variety\_ID 参照 Variety 的 Variety\_ID



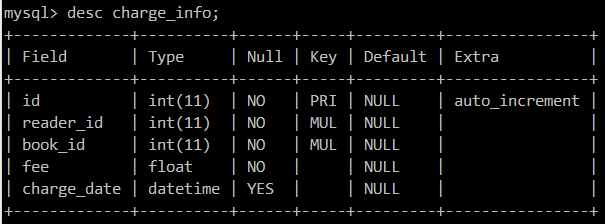
（4）lend\_info 预约信息



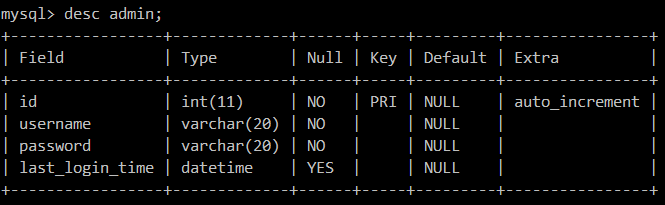
（5）booking\_info 预约信息



1. charge\_info 罚款信息表

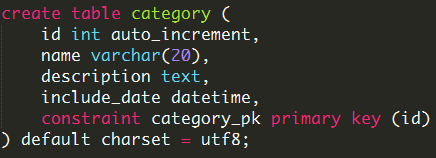


（7）admin

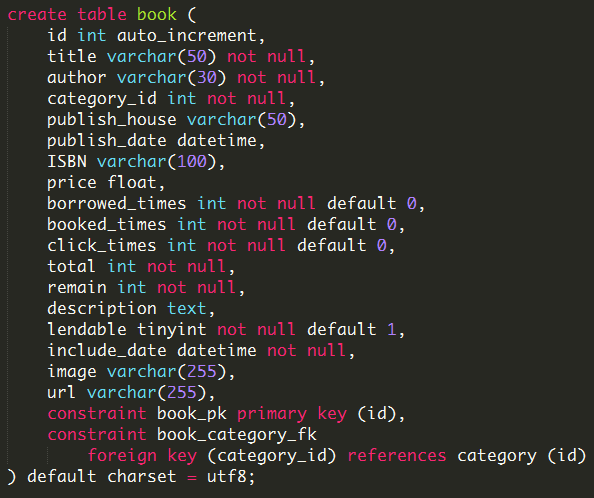


# 物理结构设计

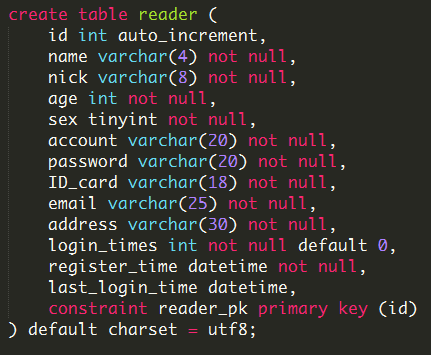
（1）Category（类别表）



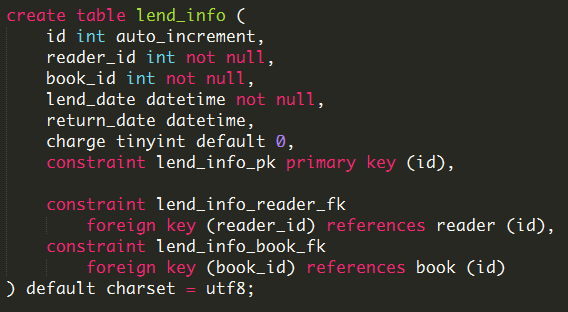
（2）book （图书表）



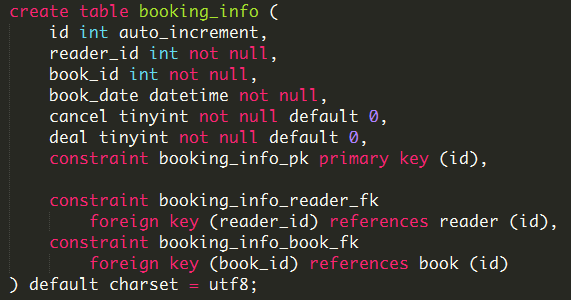
（3）reader（读者表）



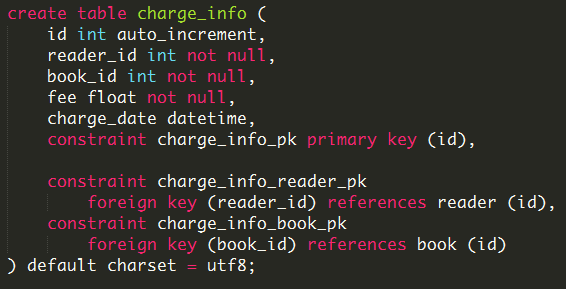
（4）lend\_info（借阅信息表）



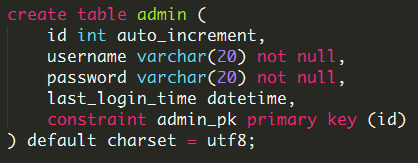
（5）booking\_info（预约信息表）



(6)charge\_info（罚款信息表）



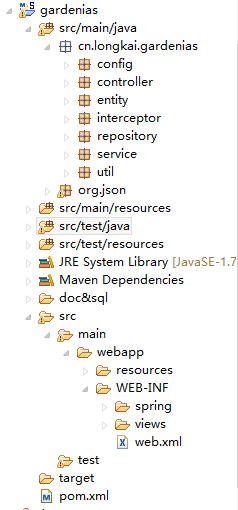
（7）admin (管理员表)



# 系统实现与测试

本系统使用经典的mvc（model，view，controller）思想，配以先进流行的技术来实现，以面向接口来驱动开发。测试使用了JUnit单元测试，所有的核心接口都通过了JUnit单元测试。

# 5.1 类图的组织层次



这是一个典型了maven管理的项目，简单地描述一下包图的构造：

src/main/java java主程序放置目录

config: 这里放置了一些配置文件的java类

controller：这里放置的是控制器类，用来处理客户端的请求并返回给对应的视图。

entity：实体类，与数据库里的表一一对应。

interceptor：拦截器类，拦截没有权限的用户非法进入某些页面

repository：数据访问接口类，这里定义了各个实体类相关处理数据访问的接口，子impl包有对应的实现类。

service：对外服务接口，这里定义了系统对于客户端提供的服务，子包impl有其实现

util：工具类，提供一些工具方法，比如异常，时间处理等

org.json：这个是使用json.org官方提供的处理json数据的一组类

src/main/resources 应用程序资源文件 放置了一些配置文件，比如logback的日记记录配置文件等

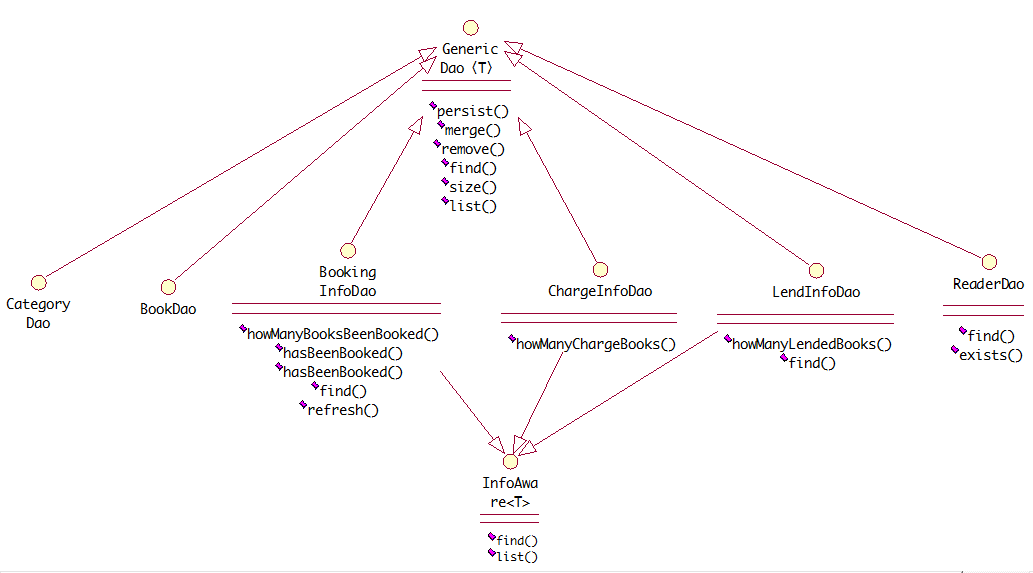
src/test/java java JUint测试类放置目录，这里对数据访问接口和对外接口进行了测试

src/test/resources 测试时提供的资源文件放置目录

src/main/webapp web应用程序文件放置目录，包括web配置文件，视图文件，spring配置文件还有html，css，JavaScript，图片资源等文件。



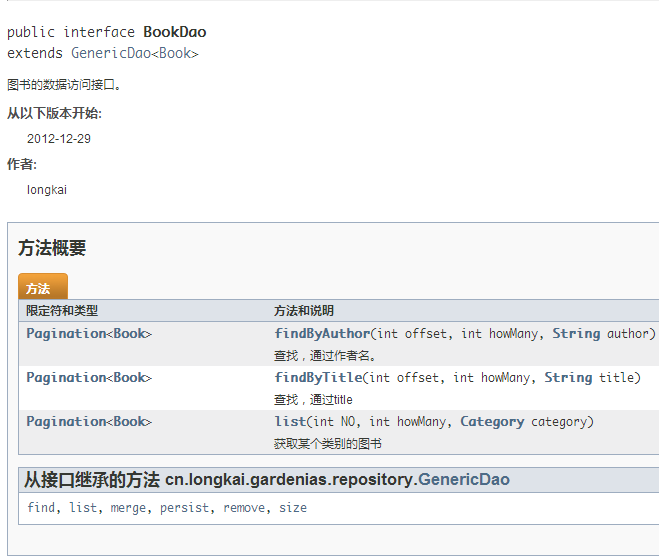
# 5.2 数据访问层接口（rose画出）



由于rose在这里不显示参数，所以看不到方法所需要的参数了，那么下面简单地说明一下接口的方法，简单，简洁，起见，将自己写的代码用javadoc生成了说明文档，这里就截图啦。之前写代码时写的注释文档现在就起到了非常好的作用！







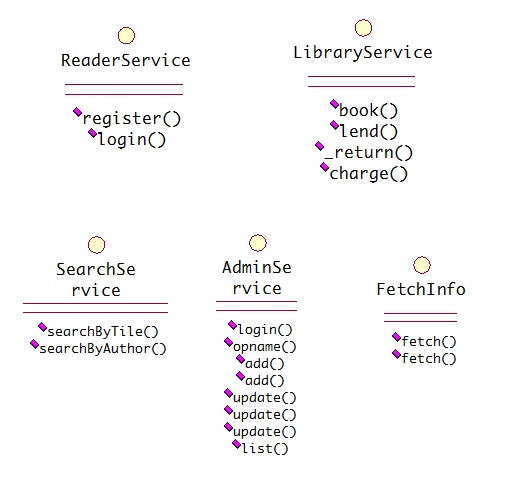








# 5.3 对外服务接口



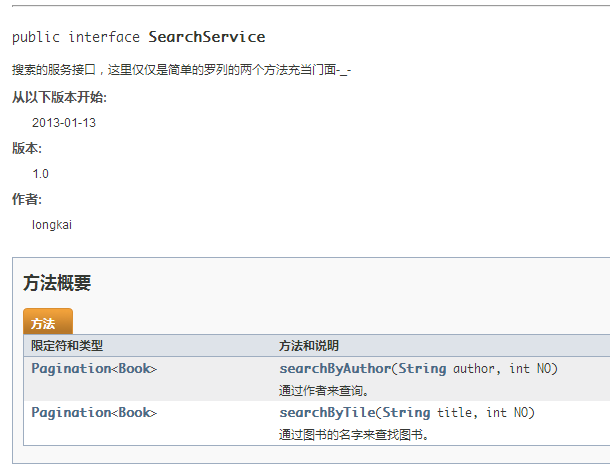
其相应的方法签名如下：











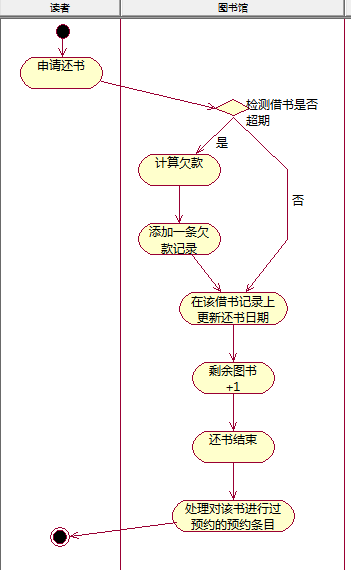
# 5.4 以一个例子来具体描述功能的实现

通过如前所述定义的数据访问层和对外服务层的接口，根据需求分析中的说明分别写好接口的实现。在对外服务层中调用数据访问层的接口，再在控制器通过调用对外服务层来对客户端的读者提供服务，返回相应的视图，这就是这个系统的设计模式。接口与实现相分离，即方便了代码的编写与测试，又增加了代码的优雅性，利于日后的维护与扩展。

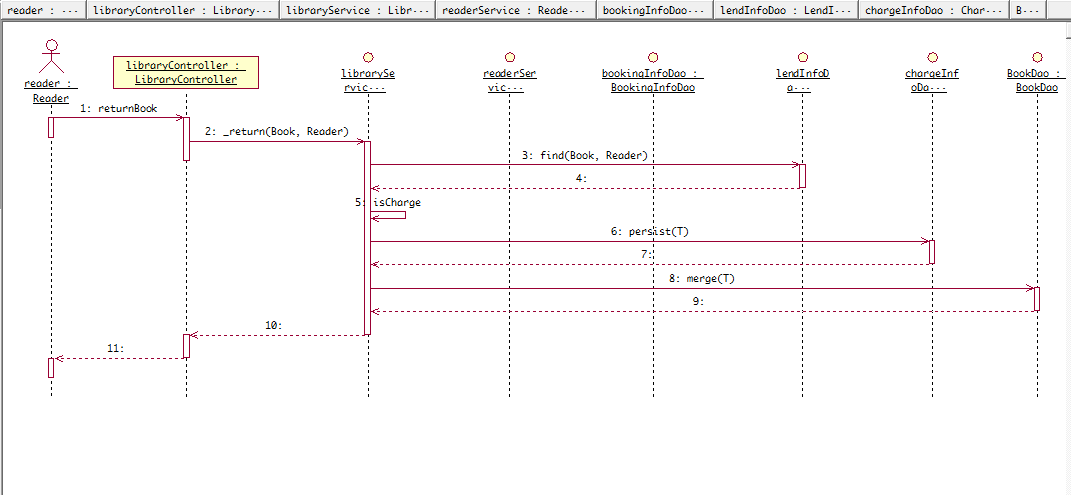
在这里我们通过一个用例来看看整个系统是如何交互的，使用rose画的图，以**还书**为例。

其需求如下：读者借阅的书籍必须按时归还，否者将会按照每本书每日0.1元处以罚款。更严重的，将加以法律追究！书本归还之后，将对该书的预约信息进行更新。

对应的活动图：



对应的序列图如下：



这个序列图是调用接口的，具体实现没有来得及画，可能会有点问题，但是无碍我们来看还书的过程各个对象见是如何调用的。由于序列图可以转换成交互图，所以这里就不画了。

另外，通过这个用例的实现，也可以大致看出整个系统的实现概况。所以，整个系统的实现就介绍到这里，**完整的uml图和源代码见附录**。

# 5.5 客户端试图展现

客户端各个页面的实现原理很简单，就是通过移动设备采用触摸手势来与服务端进行交互。读者(管理员)通过触控来传递请求，服务端通过对外服务接口再调用数据访问接口处理请求后返回相应的视图文件，若出了错，就会跳转到错误页面，给予用户相应的提示。

（1）首页界面（登陆页面）



（2）注册页面



（3）登陆成功后的主页



（4）搜索结果页面（就在刚才那个页面，我们选择**通过书名**）

下面是输入关键字“红”的结果视图：



（5）查看修改个人信息



（6）通过类别获取相应类别的图书

触摸历史按钮的视图，貌似把红楼梦放错的类别-\_-）

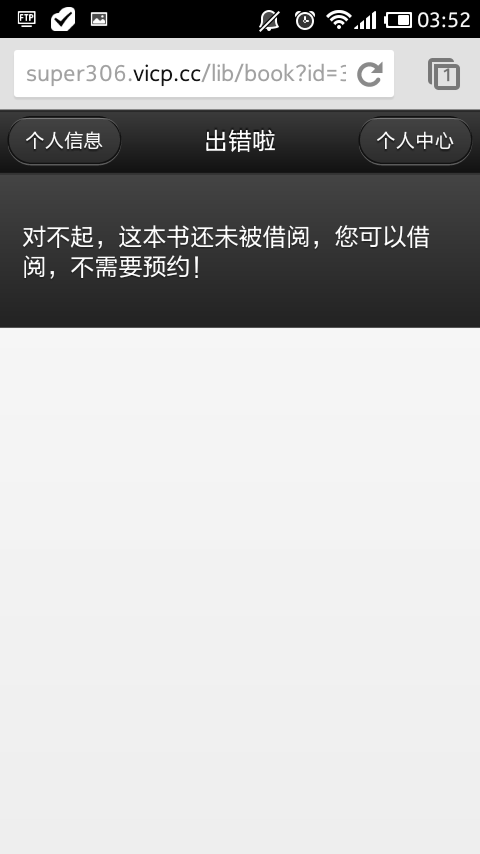


(7)点击查看图书的信息



（8）预约图书

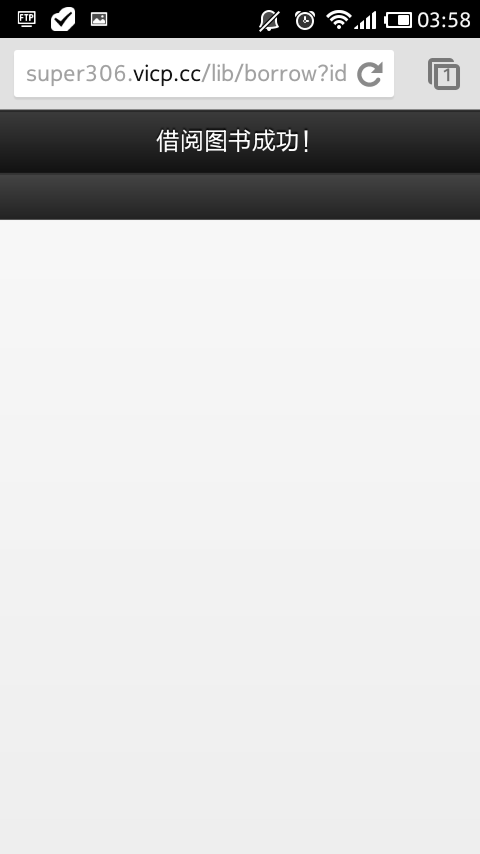
在上一个试图，我们点击预约图书，会出现如下的情况：



功能及原理：当读者触摸预约按钮时，请求就会发送到服务段，控制器收到请求交给服务接口，但是实现原理里面对预约有这样的要求，如果书还有未借的，那么预约就会失败。所以将错误信息试图返回到了客户端。（类似的，错误处理在系统中大量出现，保障了系统的稳定性），碍于篇幅以及其他，其他的错误提示就不截图了。

（9）借阅图书

在上上一个页面触摸借阅按钮，则会出现如下试图：



功能及原理：由于我没有欠款，也没有借满10本书，之前也没有借阅过这本书，而且这本书也没有被人预约，所以我能借阅这本书。（这个是成功的试图，返回了一条成功的信息，其他的成功信息也类似，不再附图）。

（10）个人中心

当读者触摸个人中心按钮后则会出现如下视图：



这里使用了json数据请求，是一个亮点，可以在一个试图上以异步请求的方式在同一个试图（不需要切换试图）动态加载数据，比如，读者触摸我的借阅信息时：



点击查看更多，则会再次通过异步请求获取数据在动态的加载到当前页面，不再赘述。

同时，我们在这里也可以看到相关的信息，比如说书未归还，有欠款，取消预约信息等，取取消预约图书为例：



触摸取消预约按钮后，则会出现相应的成功试图，与前类似，不再赘述。

# 5.6 后台管理

管理员这层不是我此次的重点，原理也类似，代码也差不多，就连拦截器也是几乎一样，所以，这里就简单的展现几个试图吧。

（1）管理员管理首页



<2>让我们来增加一本书吧

要不要考研呢，这是我们想要添加的书名-\_-



整个系统都有对字段是否合法进行验证，so，得老实填满哦

<3>查看图书列表

来看看，我们刚才添加的那本书有木有。在管理员首页，触摸查看书籍列表按钮。



Nice.

(4) 其他

管理员的其他操作大致一样，其实现原理也如前，这里就不截图了。细节太多，事无巨细，最好的办法就是亲自去体验，截图太累，排版太麻烦。。。源代码及其所有文件件附录。

# 优缺点自我评价

本次数据库项目其实做了两次，这是第二次了，第一次将近做好了，但是在年底被舍友给把整个硬盘删除了，可以没找回来。很郁闷，于是就有了第二次（这次吸取教训，使用git来做为版本控制工具）。第一次和第二次的差别在于第一次是使用原声的jdbc操作数据库语句，视图使用传统的web页面。第二次，也就是这一次，出于快刀斩乱麻的原因，使用了hibernate这个开源的orm框架来与数据库层打交道，减轻了一部分代买量。另外，第二次也在视图层使用了适合于移动终端的视图展现，通过触摸来实现交互，还挺漂亮的挺方便的。

本次数据库设计，我个人淡化了数据库在整个系统中的**出镜率。**虽然可能有些不符合数据库课程设计这个名字。但是，个人觉得，数据库固然重要，但是它应该也无可争议作为应用程序的最底层，底层的东西，是高层东西的基石，但是底层的东西就是传统的sql语句之类，在应用程序中代码大多都类似，过度强调还不如直接在控制台写数据库语句来得直接，所以，我这篇报告写得可能有点四不像吧，既不像数据库，也不像软件工程，像个大杂烩，不过这其中体现的思想和方法确实也是我学习和实践的收获。

本次课程设计，采用了很多当前工业界流行的，最新的技术，完全可以作为一个showcase来展现，一开始写就打算把它共享，或者说开源吧，所以风格代码写的非常规范，也有点刻意的取使用某些面向对象的设计模式，封装等思想，目的是想从这个项目中传达一种思想之类的吧。毕竟，感觉周围的人还欠火候。大量的注释带来的好处就是生产的javadoc文档，省去了很多敲击。最后，这个项目使用uml类图类建模，现学现卖，感觉帮助挺大的。

整个项目使用到的平台技术，框架，第三方类库都是免费和开源的，没有一分钱的花费，这个比起sqlserver要好一些吧，感谢开源的东西，这个小项目，我也已经提交到了github上去了。

至于优缺点，我觉得优点就是在于一个“新”字，这个项目可以跨平台的运行在各种移动设备上，甚至，还可以转换为相应平台（Android，ios）上的原声app。对于性能和安全都有考虑（具体见附录）不过这个是后话了。

说一下自己感觉的缺点

<1>由于使用了太多的技术，编写的代码量比较多，对于这个简单的系统，甚至有些过了 ，造成了一定的复杂。

<2>控制器没有完善，写得不是很合理，get，post方法没有连接好。

<3>没有使用原生的jdbc和sql来操作数据库，弱化了sql ，性能上也有减弱。

<4>题材不够新，很多人是写图书馆系统，这个就显得不够新了。但是这个是完全模仿西大图书馆的用例，作为学生，作为读者，也希望能有图书馆的移动客户端出现。

<5>图书馆的续借功能忘记了，没有实现。

<6>管理员模块较为简单。

<7>没有对外提供删除的功能，虽然程序有设置。（ps：数据库一般应该是少量的删除与大量的查询，我就懒得没有提供删除的接口了）。但是从数据库设计的角度来说是不对的。

<8>出于时间问题，代码没有经过大量的测试，某些地方人存在问题。

<9>在写报告时，太多的截图了。

# 参考（都来自互联网的文档）

主要是这个网站

jquery mobile http://jquerymobile.com/

附 录

最初是想把源代码附在这里的，但是写到这里还是觉得没有必要了，这个毕竟是数据库课设，不是语言或者软件工程课程，贴上来的话太长了。

源代码我已经共享到github.com上了，至于git，是现在一个非常非常流行的版本控制工具，Linux的内核，Android的源码都是用这个进行管理的。Github是一个最流行的代码托管网站，对于开源的代码提供免费的托管，任何人都可以获取和修改该代码。Git会追踪每一行代码的变更，回滚至原来的版本等。（这里的介绍稍显多余）

<https://github.com/longkai/gardenias>

这个连接可以自由地得到所有的源代码和相关的文件，包括这个报告的文档，sql文件，rose uml图，reademe等。运行试试比截图效果更佳-\_-

gardenias是本次课设的开发代号。无实际意义。

super306.vicp.cc是我用自己的机器和宿舍的路由器加上一个免费的二级域名来构造的一个真实的服务器，假如我的电脑开启了这个系统服务，那么还真可以对外提供服务。

有问题的话可以通过邮件方式联系im.longkai@gmail.com我。