摘 要

当数据处理数量愈加庞大，超过了人工能够处理的范畴，这个时候就需要一个管理系统进行辅助管理。最方便的管理系统便是基于Web端的管理系统，不管是在寝室还是在办公室，通过计算机去访问Web端的管理系统，即使不在那个地方，也能够及时地快速地获取自己想要的信息。

然而，随着Web技术的不断发展，一个Web应用程序就必须提供给用户一个良好的人机交互界面。目前，流行的页面布局方法是DIV+CSS的布局模式，为了使得页面变得更加酷炫，使用bootstrap进行页面样式的渲染。

JavaScript用于创建动态页面。为了使得JavaScript使用户能更方便地处理HTML（标准通用标记语言），开发者们将JavaScript打包成JavaScript库，比如jQuery库和prototype库，这些JavaScript库能够简约js代码的编写，而且，降低JavaScript使用的难度，提高开发效率。

AngularJS是在2009年的时候，由Misko Hevery等人创建的，后为Google公司收购。AngularJS是一款优秀的前端JS框架，已经实现了Web应用，开发者只需要对它填充具体的业务逻辑就好，AngularJS框架在构建一个CRUD应用的时候特别有效，所以，在搭建类似的图书借阅管理系统的Web端的管理系统的时候，使用AngularJS框架能够使得事半功倍。

关键词：DIV+CSS，bootstrap，JavaScript，AngularJS，人机交互，层叠样式表

Abstract

We should have a manage system to help us to deal with the bigger amount of data which beyond that human beings can deal with today.And it may be easily for one to query and manage information he or she want by surfing in the Internet.No matter where are you,in your bedroom or other place,there just have Internet and you have a PC with you,then you can get the information you want in time by browsing the browser on your PC which is more quickly.

With the developing of the Web technology,A standard Web Application has to provides users with a friendly human-computer interface.Currently,it is popular to use DIV+CSS to build a Web page layout.Besides,to make a cooler page,we always use bootstrap to decorate the page.

JavaScript is used to build a deployed page.Developer always pack JavaScript as a package to make it easier to build a web application.Like Jquery and prototype,which are easy to use and are efficient.

In 2009，AngularJS is made by Misko Hevery and his friends,which is purchased by Google later.AngularJs is an excellent JS frame,which has realized that developers only need to fill it with specific business logic.AngularJS is excellent building a CRUD application.So,it is suitable to use AngularJS to development a manage system like this.

Keywords:DIV+CSS,bootstrap,JavaScript,AngularJs,human-computer interface,CSS

目 录

1. 引言
2. 概述

引 言

为了提高对图书借阅记录数据的管理，需要借助一款管理系统对书籍的每次借阅做登记，管理员则能够通过管理系统快速，准确的定位了某一条数据，这就是开发图书借阅管理系统的主要目的，就是为了减轻人工记录的压力，同时，加快对数据的查找速度与准确度。

传统的网站应用程序，都是前后端即浏览器端和服务器端分开的，整个应用程序的业务逻辑一般是使用Java或者C#等后台语言进行编写，部署在服务器上，当前端有请求产生，服务器接收请求，对数据进行解析，之后返回到前端，前端接收到数据后，对页面进行渲染。这样的方式，大大增加了服务端的负载。使用AngularJS框架进行开发，能够提升服务器性能。因为他只要求后台服务器端提供JSON数据，对数据的解析和计算都放在客户端来完成。使用JSON数据和前端缓存的方式，客户端的流量大大降低，服务器端只需要响应API的调用，并用JSON格式的数据返回。同时，Angular只需知道后台的API，在接收到后台穿回来的数据后，对数据进行解析，之后马上对页面进行渲染，当页面发生改变时，又能立刻将改变的数据通过API以json的形式发送回服务器端，这样，避免了传统的方式产生的整个页面刷新的弊端，提高了人机交互的友好度。

本课题主要的研究内容是AngularJS框架如何在构建一个网站式的管理系统应用程序。显然，一个管理系统的实现最基本的功能便是增加数据，删除数据，修改数据和查找数据，在这样的应用中，AngularJS会显得的特别的死板，他自带的数据双向绑定的功能，表单验证功能，路由功能和依赖注入功能等都是一个CRUD（增加Create、查询Retrieve、更新Update、删除Delete）应用可能用到的全部内容。换句话来说，AngularJS可能就是专门为管理系统网页应用程序而生的。具体来说，我所研究的内容是AngularJS如何是如何使用其特性，在构建一个管理系统的时候实现增删改查这四个基本功能。

图书借阅管理系统便是一个典型的需要CRUD操作的管理系统，如何将AngularJS框架用于这样一个实际的问题中便是主要解决的问题。首先，我们知道，AngularJS对视图模板进行渲染的前提条件就是服务器端必须有数据模型给angular。这就是第一个问题，我们将如何从服务端获取数据，同时，要将数据以json的形式传给angular。第二个问题，为了更好的用户体验，我们要实现的是网站做到尽量少的刷新，最好是局部刷新。第三个问题，在对网站页面进行渲染时，除了使用bootstrap进行样式渲染外，我们要构建一个动态页面离不开使用JavaScript对页面的DOM（）元素进行操作。为了减少对DOM()元素的操作难度，开发者们将JavaScript打包成一个一个Java的类库，目前，我采用的是jQuery类库，然而，在使用jQuery类库的时候，必须考虑到AngularJS对DOM()元素的渲染问题，解决两者之间存在的冲突是一个重要的问题。我将使用实例对这些问题一一进行解决。

1. 概述

2.1基于AngularJS的图书借阅管理系统概述

与传统的图书借阅管理系统不同的是，我采取了前端使用AngularJS框架，将业务逻辑的编写从后台迁移到了前端JS部分。这就降低了服务器端的负载，我们要求后端只需以json形式将前端请求的数据提供过去就好；除此之外，angular可以实时监听页面模板发生的改变，一旦页面模板发生改变后，angular通过接口将数据传送给后台。虽然，这听起来和传统的网站运作模式差不多，但是，使用angular能够让你的页面局部刷新，而不是每次请求都让整个页面刷新，这就是AJAX（）原理，angular通过封装的方式，大大简化了原生JavaScript对AJAX（）的使用，提高了代码利用率的同时，降低了Ajax的使用门槛，开发者们只需会用Angular提供的AJAX接口，将数据填入即可，不用再自己对浏览器是否为IE进行判断，对请求的状态status进行分类讨论。

一个网页应用程序的开发，开发者们往往会使用MVC（）框架进行。因为这样，我们能够将数据模型（model）和视图（view）分开，用逻辑控制器（controller）来连接两者，使得整个网站应用程序能够正常的运行起来，同时，通过这样模块化的布局，也方便开发者进行端到端的调试，以及日后对代码的维护。

Angular的迷人之处就是，它将MVC模型从后台搬到了前端来，我们在前端部分构建了一个MVC框架体系，与传统的模式相同，它也具有数据模型、视图模板、逻辑控制器三个部分，但是，angular的模型是负责管理应用程序的数据的，当视图发生改变时，数据模型能够响应视图模型发来的请求，同时，也能够响应从控制器发来的请求，进行数据模型的自我更新；至于视图，是呈现给用户的数据模型的展示，它是由控制器来控制所展示给用户的数据；逻辑控制器则负责与用户进行交互，当用户有输入时，控制器能够执行交互数据模型对象。

基于AngularJS的图书借阅管理系统离不开AngularJS本身在管理系统方面的优势。一个管理系统离不开对数据库插入数据，删除数据，修改数据，查找数据，还有频繁的表单验证功能。庆幸的是，AngularJS框架能够帮我解决大部分表单验证的功能以及涉及到增删改查的操作。例如，在表单验证的时候，我们只需在必填的字段添加一个required命令，在表单提交按钮上ng-disabled命令赋值为!Formname.$vaild，这样一来，当用户必填字段出现空值的时候，就无法正常提交表单，同时，会在空字段出现提示，要求必须填上值才能提交表单。这样的做法，比起使用传统的JavaScript进行DOM元素的验证效率更加高，而且，节约了代码的书写，使得代码变得更加的简洁，可读性更加的高。这就符合AngularJS的创始人的初衷，即要求开发者要从构造声明式界面入手，使得一个网页应用程序的层次更加分明，可读性更加高。

该图书管理系统核心的观念就是页面的局部刷新原理，我将使用AngularJS的路由包angular-route.js实现对页面视图的局部刷新，一些公共部分，例如网站的标题栏和网站的导航栏不需要刷新就不去刷新他，要做的是定位到ng-view指令的位置，对该位置的视图模板进行刷新，这样做的好处显然易见，在用户看来，页面能够立刻回应自己的请求，仿佛是一个静态页面一样，而不是像以往那种服务器端接受到前端的请求后，立刻刷新整个页面。而且，AngularJS允许我们只在一个页面上进行所有功能的操作，这就是近年特别流行的OnePage的理念。一个网站的所有页面肯定会有重复的部分，可能是导航栏部分，可能是页脚部分，如果，每做一个页面都重新编写这些部分的话，那么会增加这个应用程序的代码冗余度，之前，<iframe>标签非常受欢迎，因为，使用它能够使得所有的页面展现在同一个页面上，然而，一切只是看起来比较顺利而已，时间一久，开发者陆续发现，在一个页面使用<iframe>标签来实现页面嵌套页面，看起来已经是在同一个页面中了，可是，iframe标签使得整个页面框架变得不稳定，并没有得到认可是OnePage的最好实现形式。相反的，我们采用angular-route实现的效果类似于iframe，但是，ng-view下的视图模型并不会破坏页面的整体性。

但AngularJS毕竟只是前端的框架，我们都知道，在客户端上是无法对服务器端进行数据的直接修改和编辑的，要实现对用户修改的数据进行保存，我们必须借助一种服务器端语言来完成。首先，我们必须明确的是，基于AngularJS的图书借阅管理系统要求后台实现的功能是什么，上文也有说道，AngularJS的视图展示和控制器的逻辑操作都离不开数据模型，而数据模型必定是要从后台服务器获取到的。后台所要进行的操作就是从数据库中取出数据，并将它与数据模型进行绑定；同时，当数据模型发生改变时，后台又要将新的数据传送回数据库，更新数据库。除此之外，貌似，后台并没有干其他的事情，而所传送的数据必须是Json数据格式。目前主流的服务器端脚本语言有Java，C#，PHP等，而这三种脚本语言都能够实现，但是，考虑到实现的工作量大小，PHP这样的轻语言貌似更能胜任这项工作，因为PHP不需要太多的配置，对平台要求不高，所需要的IDE（集成开发环境）也不用太复杂，相对于其他的服务器端的脚本语言来说，在做简单的数据交互这项工作时效率更高。

最后，我们需要的是一个能够存储数据的容器。图书借阅管理系统离不开大量的书籍信息的保存，用户信息的保存，用户借阅书籍记录的保存，这时候就需要使用数据库来对数据进行保存。最终，我采用的是MySql数据库，第一，因为它是一个开源的免费的数据库，可以避免不必要的开发耗费。网上有许多Mysql的可视化管理工具，在使用mysql数据库的时候也比较方便，遇到问题能够及时快速的找到解决方案；第二，由于服务器脚本语言采用的是PHP，PHP中有支持mysql数据库的PDO对象，这样的组合将会给后续的开发带来事半功倍的好处，我们将节约大量的时间成本不去配置数据库驱动，PHP语言的PDO对象将会为我们搞定一切繁琐的配置行为，我们只需传入数据库的名称和数据库管理员的账号密码就能够通过PHP对数据库的数据进行操作。

以AngularJS为核心，结合PHP脚本语言，使用Mysql数据库，采用单页面来实现图书借阅管理系统的基本功能，同时，会增加用户对书籍的评论功能，使得这个管理系统变得像一个交流社区一样，增加了管理系统的人性化设置。

2.2使用AngularJS进行开发的方案设计概述

AngularJS框架2009年创造出来，后被谷歌公司收购后，在谷歌公司这样大的公司的维护下，广大开源社区的开发者们陆陆续续使用angular进行前端的开发，90%的网页应用都是CRUD应用，所以，angularJS的舞台非常的广阔，图书借阅管理系统类似的各种管理系统通常都是首选AngularJS框架进行开发的。

所以，我将会以AngularJS为骨架，给它添加其他的血肉，使得一个网站真正的活起来。首先，要对视图模型进行渲染，我使用的是bootstrap样式库实现的。由于Bootstrap拥有海量的样式形式，我们只需在HTML标签上给class属性赋值，便能获得相应的样式。通过这种方法，在减少我们自己写样式的同时也能提高样式的质量，增强视图模型的渲染效果。

除了AngularJS这个JS框架之外，我将“冒险”地采取jQuery库来实现简单的DOM（Document Object Model）事件绑定，比如，对元素的样式class进行改变。虽然，从AngularJS的开发初衷来说，这是有违发明者的初衷的，AngularJS的发明者所希望看到的是前端开发者在进行HTML（超文本标记语言）元素布局时候，就能够将逻辑指令注入进去，这体现了AngularJS对HTML元素的补充与完善，是进步的体现。可是，从另外一方面来说，一些简单的事件响应，用jQuery能够花几行代码就实现，AngularJS的话可能需要编写一个命令，插入到html标签中，这么一来，工作量明显会增加许多，所以，我认为，在AngularJS中使用jQuery也是必要的。为什么要考虑到AngularJS与jQuery的并存问题呢？因为上文说道，两者的开发理念并不相同。jQuery开发者们往往优先考虑写好DOM架构之后，使用jQuery对已有的DOM进行操作；相反的，AngularJS就要求开发者要有全局统一的思想。在进行DOM架构的时候，也要考虑到响应逻辑实现，在html元素中插入命令属性，其实，真相是AngularJS使用内置的jqlite库实现jQuery的功能。所以，两者一起使用的话，必须解决他们之间的冲突问题。

当我能够正确的合理的协调好jQuery和AngularJS之间的矛盾之后，接下来要考虑的就是如何处理好AngularJS和后台的交互问题。上文我曾经说道，选择了PHP脚本语言作为后台逻辑脚本，其实，这边并不涉及到什么逻辑问题，因为，我只需要使用它从服务器端获取数据，并且将数据以json的形式传送给前端，其他的逻辑问题就全全交给AngularJS来完成。使用PHP语言的好处之一就是，能够与Mysql数据库进行比较良好的连接。不必像使用Java语言开发一样，使用时候还需要安装mysql的驱动包，再继续配置数据库的链接，PHP 5.0版本以上可以使用PDO对象，直接传入数据库的名称，mysql的账号，登录密码，便可以建立连接。这样，减少了不必要的代码耗费，减轻了配置服务器和数据库的负担，方便开发者进行开发。

1. 系统分析

3.1、系统目标分析

使用图书借阅管理系统的目标是双方面的。第一个方面，为了提高图书管理员对书籍借阅数据的管理效率；另一个方面，为了方便借阅者对书籍信息的查询与自己借阅记录的综合统计。考虑到双方的需求，制定的具体目标如下几点：

1. 实现对管理员和普通用户的划分，管理员拥有普通用户不具有的管理权限。
2. 管理员能够要能够实现书籍的上架与下架功能，能够查询到每本书籍的借阅信息，当借阅者超期时，能够发送通知到指定账号，通知借阅者及时归还书籍的功能。
3. 普通的用户要能够实现对书籍根据书籍名称、书籍作者、书籍出版社进行查询功能；以及，用户能对一本书籍进行评论以及星级判定。用户还能够续借自己借阅的书籍，如果所借书籍超期，则系统会提醒用户及时归还，并不会允许用户续借。
4. 当架上书籍的数量为0 的时候，用户能够预定书籍，当书籍数量不为0 时，系统将根据预定者的时间进行排序，管理员依据排序来通知预定者来借阅自己预定的书籍。
5. 系统能够依据用户的好评与差评数对书库中的书籍进行热度排行，用户可以根据书籍的种类来查询相对分类中的热门书籍。

3.2、功能需求分析

该图书借阅管理系统要实现的功能有以下几个：

1. 登录功能：系统能够对用户名和密码进行校验，非注册用户或者用户名和密码不匹配者无法进入系统，并提示错误；当用户成功登录之后，系统保存用户登录信息。如果检测到保存的登录信息为空的话，则强制返回登录界面。
2. 注册功能：若用户没有用户名则可以通过注册功能进行注册，注册功能实现的是对用户名进行校验，保证用户名的唯一性，不得重名。
3. 权限功能：根据登录用户的权限等级来赋予不同的操作等级。例如，管理员账号能够使用管理员版块的功能，而普通用户无法使用管理员版块以及进行管理员相关的操作功能。
4. 书籍的搜索功能：根据用户输入的书籍的关键字，可以是书籍的作者名，可以是书名也可以是书籍的出版社来对相关的书籍进行搜索。
5. 书籍的详细信息查看：用户能够点击书名进入书籍详细页面，该页面详细的显示包括书籍的书名，作者，出版社，版本号，书籍的内容简介等信息。
6. 对书籍进行评价：用户能够根据自己对于书刊的喜好进行好评或者差评，这有助于其他用户发现好书而避免借阅不满意的书的情况发生。除了能够好评或者差评之外，用户还能够发表自己关于这本书的文字评论，显得更加详细，读者们也能够通过评论进行交流。
7. 热门书籍的排行：系统能够根据每本书的好评数对书进行热度排行，让读者们能够快速了解到哪些书是有趣的，同时，也方便管理员进行管理，对于长期处于垫底的书籍，可以进行下架处理，换上更好的书。
8. 用户信息修改：用户能够修改自己的昵称和登录密码，当然，系统会校验用户的输入两次密码是否一致等功能。
9. 图书预约功能：当有些书因为太受欢迎，可能会出现供不应求的情况，这个时候，用户可以对这些书进行预约；当前一批借阅者将书归还后，有库存后，管理员能够在预约名单上通知那些较早进行预约的同学来领书。
10. 借阅记录查询：用户能够通过该系统来查询自己的借阅记录。借阅记录分为两类。第一类，是借阅后未归还的。在这边，用户能够看到自己借阅的书籍信息，如果希望延长借阅时间，可以在到期前进行延长借阅；如果借阅已经逾期，那么将无法进行延长借阅 的操作，相对的，系统会提醒你尽快归还书籍；管理员的话，能够发送信息给逾期不还的用户，提醒他/她及时归还书籍。第二类，借阅后归还的记录。用户可以看到自己借阅的书籍名称，借阅时间，归还时间等信息。
11. 消息中心功能：当用户登录进入系统后，如果有消息未读的话，则系统会循环滚动提醒尽快阅读消息；或者，用户可以直接通过导航栏进入消息中心，在这里，用户能够看到哪些消息是未读哪些消息是已读的，对于未读消息，能够标记为已读，或者可以直接删除消息。
12. 管理员新书上架：当用户权限为管理员时，才能够使用此功能。录入要上架的书籍信息，包括书籍的名称，作者，出版社，版次，书籍简介，书籍分类等。当书籍成功上架后，系统会自动记录上架时间和书籍编号。
13. 管理员下架功能：当用户权限为管理员时，才能够使用该功能。对于书籍列表中的书，可以选中要下架的书，将其下架。下架后的书籍，用户无法查询到。
14. 站内通告功能：管理员进行发布，将一些通知贴到首页，告知读者们信息。
15. 站外链接功能：定时更新一些优秀的读书网站，供读者们访问。
16. 退出功能：注销在线用户，退出管理系统，要想进入系统必须重新登录系统。

3.3、技术可行性分析

从开发技术来开，管理系统主要分为前端，后端和服务器端。

前端：采用的是HTML（超文本标记语言）进行页面的开发。采用CSS（层叠样式表）对HTML（超文本标记语言）进行渲染，使得页面变的更加友好，有利于方便人机交互。使用JavaScript进行前后端的传值，将静态页面变成一个动态页面。

为了方便开发，我将使用CSS(层叠样式表)的一个库即Bootstrap库。使用Bootstrap可以帮助我们更好解决页面的渲染问题，以及不同浏览器之间的兼容性问题，从而达到更优秀的渲染效果，同时，又能减少我们的开发量。

对于一些简单的元素事件，我将采用JavaScript的一个轻量级的库叫做jQuery。顾名思义，我们将搜索到要绑定事件的DOM（Document Object Model,简称DOM）元素，将事件绑定到这个元素上，如果只是简单的动作，使用jQuery的效果会比使用Angularjs要好。

前端开发的重点就是AngularJS架构的使用。上文也说过，AngularJS架构特别适合用在像图书管理系统这样的管理系统，所以，在开发图书借阅管理系统的时候，使用AngularJS是再合适不过了，因为，不需要大量的页面动画的逻辑渲染，我们只需要简单的增删改查的操作即可。

由于用户要不停地对信息进行更新，为了良好的用户体验，我们要实现的是防止页面的整体不停的刷新，打个比方来说，我们要对数据表进行分页操作的时候，应该要采取分页的操作，要实现这样，我们可以使用AngularJS的$http对象，这个对象就像AJAX一样，实现只有表格部分的数据刷新，而其他界面并不需要刷新。

后端：

4.2系统界面设计

登录界面设计:图书借阅管理系统的的登录界面采取简洁的设计风格，用户登录输入框，用户密码输入框，登录按钮和注册按钮。



图4-2-1 用户登录界面示意图

用户注册界面设计:包括页面的标题，新用户账号输入框，用户登录密码输入框，用户对密码进行确认框，验证码刷新框和验证码输入框，注册信息提交按钮。

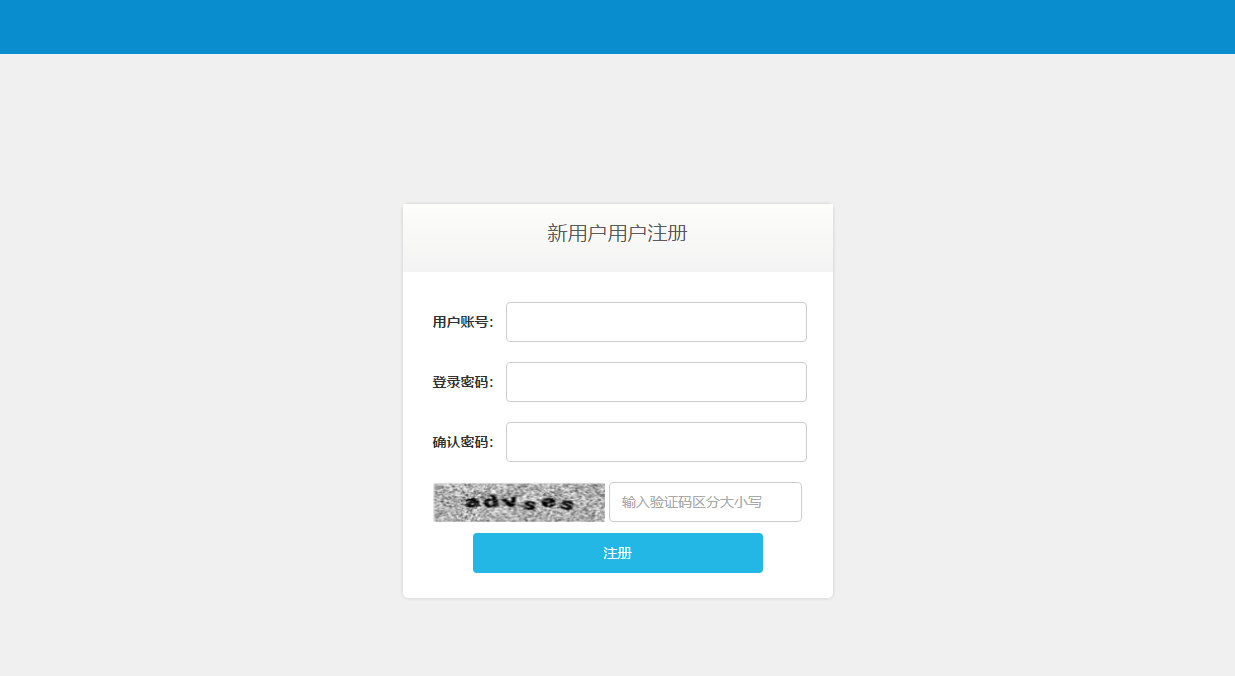


图4-2-2 用户注册页面示意图

系统界面设计：页面的页头设计包括登录者的登录信息和退出链接，导航菜单，导航菜单主要分为五个功能模块，图书查询功能块、用户版块、管理员版块、站外链接版块、站内通知版块。采用可收缩的方式进行菜单导航栏的布局，增加导航栏的灵活性和趣味性。



图4-2-3 系统界面页头部分示意图

图4-2-4 系统菜单导航收缩形态 图4-2-5系统菜单导航栏展开

页面整体布局设计：将页面的页头，导航栏，主题内容部分和页脚合理的组合在一起，完成图书借阅管理系统的主要页面框架，示意图如下。

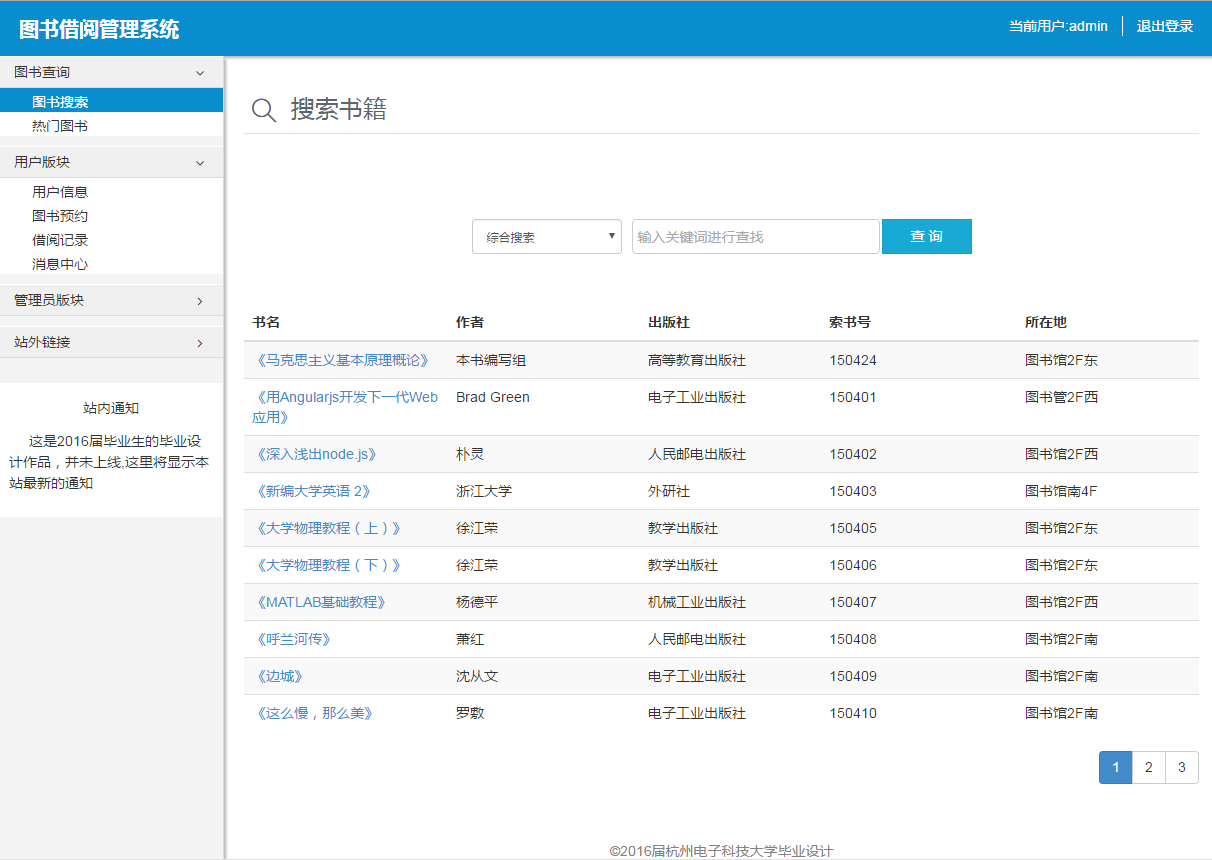


图4-2-6 页面整体布局设计示意图

图书搜索界面布局设计：主要由三个部分组成，页面的标题，搜索部分，搜索内容展现部分。标题采用的是传统的一个ICON图标配上功能模块的名称，由一条下划线分开与主体部分的内容；搜索部分，则是由一个下拉框，输入框和一个按钮组成；内容展现部分由一个表格展现，根据数据的多少进行分页操作。



图4-2-7 图书搜索页面示意图

热门图书搜索页面设计:由三个部分组成，页面标题，图书种类选择，内容展现部分。由于本系统的功能页面的标题部分采取统一的设计模式，在这边就不再一一赘述，都是有一个ICON图标和中文文字来组成页面的标题部分，以下划线划分；书籍种类选择部分，当选择书籍种类后，系统会高亮该种类书籍；内容展现依旧采取的是表格的形式，可根据内容的多少进行分页，每页预计最多展现10条记录。示意图如下图所示：



图4-2-8热门图书搜索页面示意图

书籍信息详情页面设计:由页面标题，书籍信息部分，读者评论部分，撰写提交评论部分。书籍信息部分对该书的版本号，简介，可借数量以及综合评价（以星级表示，总共5颗星）等进行充分的展现，当然，用户也能在这个页面上对书籍进行赞或者差评；读者评论部分，提供给用户一个浏览其他读者评论的平台，对于这本书有更加的全面的了解；评论发表部分，用户能够发表自己对于这本书的见解，给其他读者提供具有参考价值的信息。页面示意图如下



图4-2-9 书籍详情页面设计示意图

图书预约界面设计：主要是由图书种类，数据展现部分。根据不同的类别将会有不同的书籍展现，同样的，用表格形式展现，会根据数据的长短，进行分页，每页预设为10条数据，不同的是，每一行将会有一个预约按钮，用户可进行预约，当成功预约后，按钮变灰色，并且，无法再次点击，防止用户重复预约。

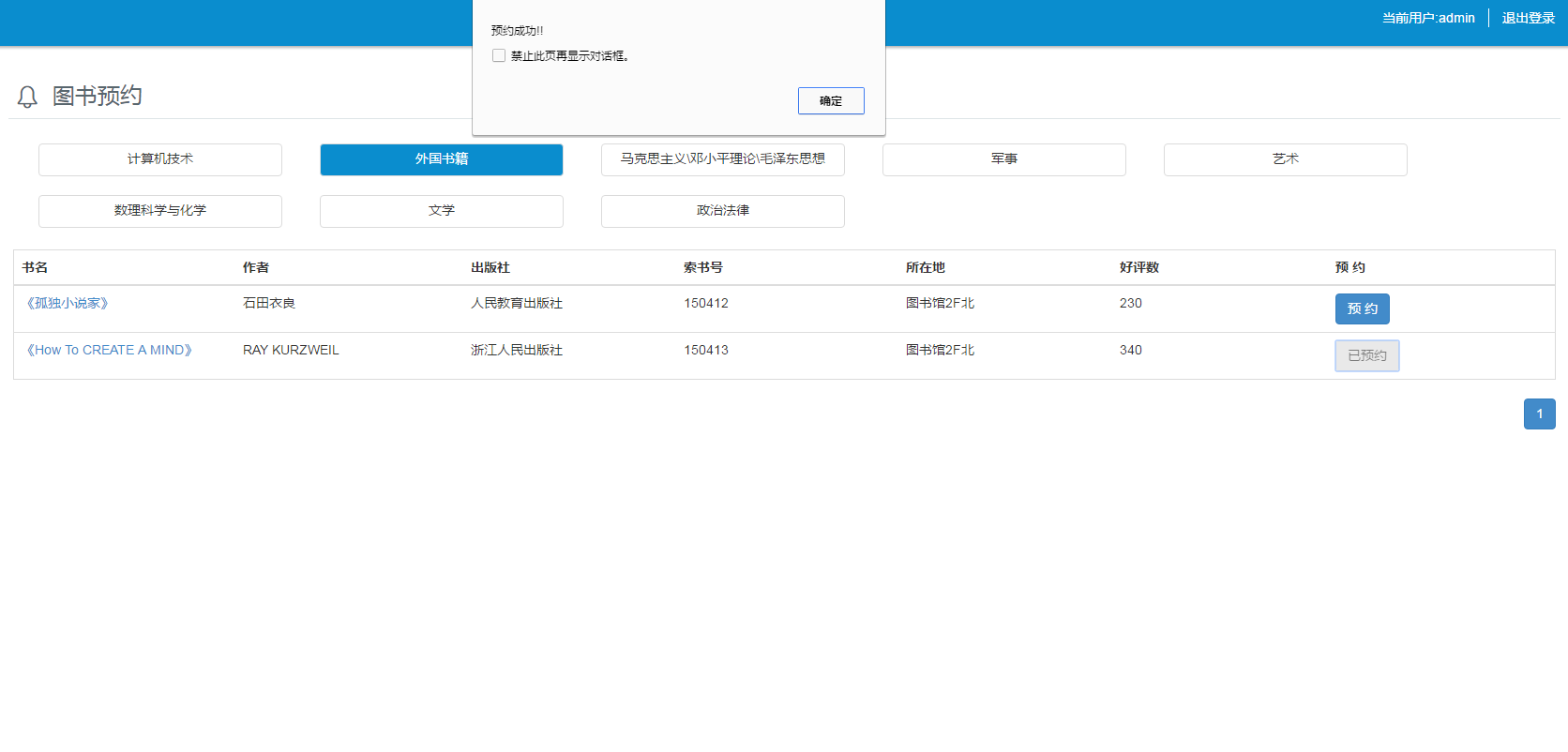
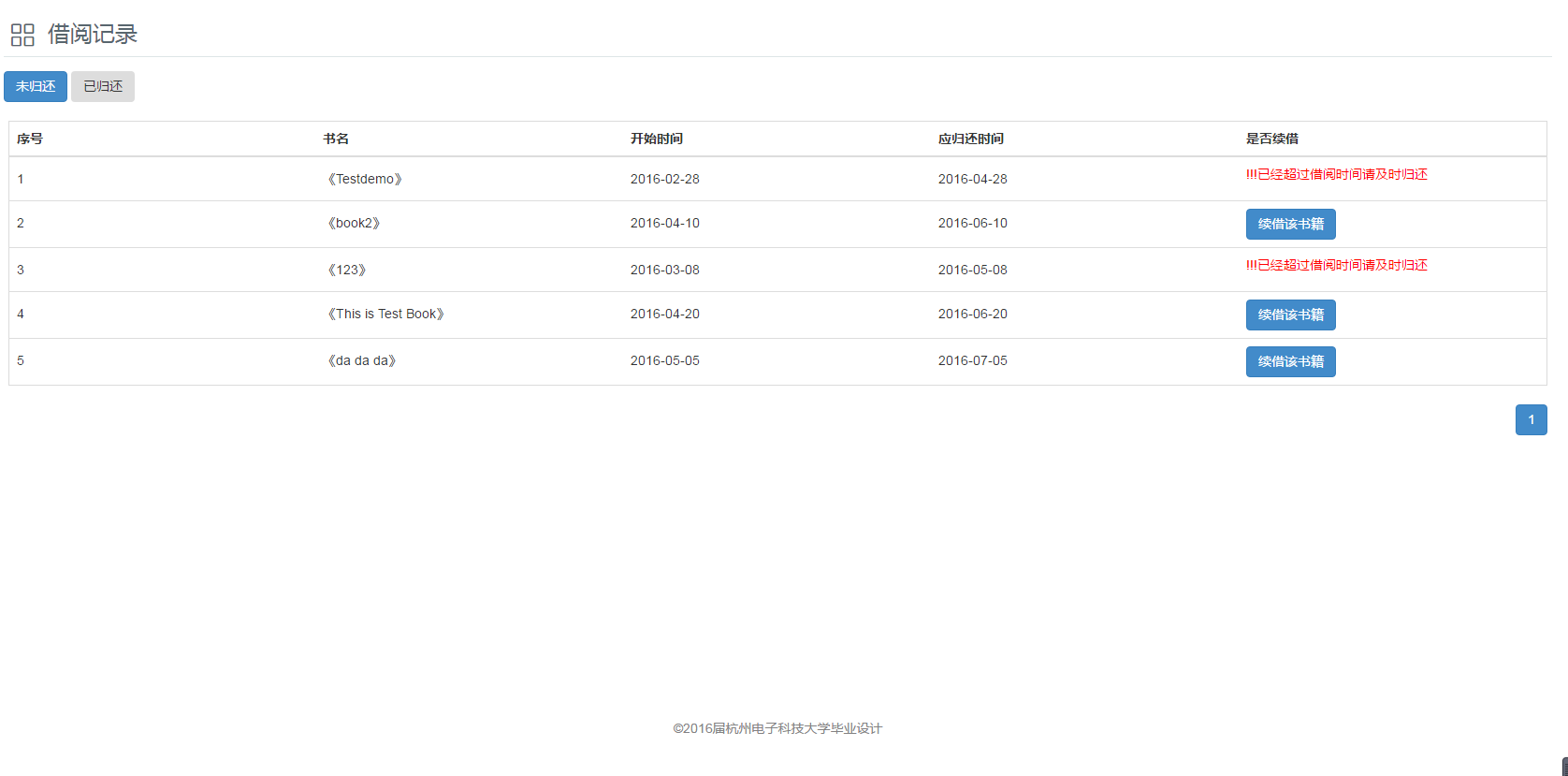


图4-2-10 图书预约界面设计示意图

借阅记录界面设计：主要由记录表格组成，表格分为两种，一种为为归还的借阅记录，另外一种是已归还的借阅记录列表，在为归还记录列表中又分为两种类型的数据，第一种是逾期未还会提示已经超期，如果未逾期的话，则能够续借该书籍。

表4-2-11 借阅记录界面设计示意图

消息中心页面设计:模块化的界面，每个模块一个短消息，所以，每个模块中都有对消息的删除功能，如果，消息是未读消息的话，将会有一个将此消息标记为已读消息的按钮，点击按钮后，消息变为已读，刷新列表，该条消息的已读按钮将会不见，这条消息变为了已读消息。

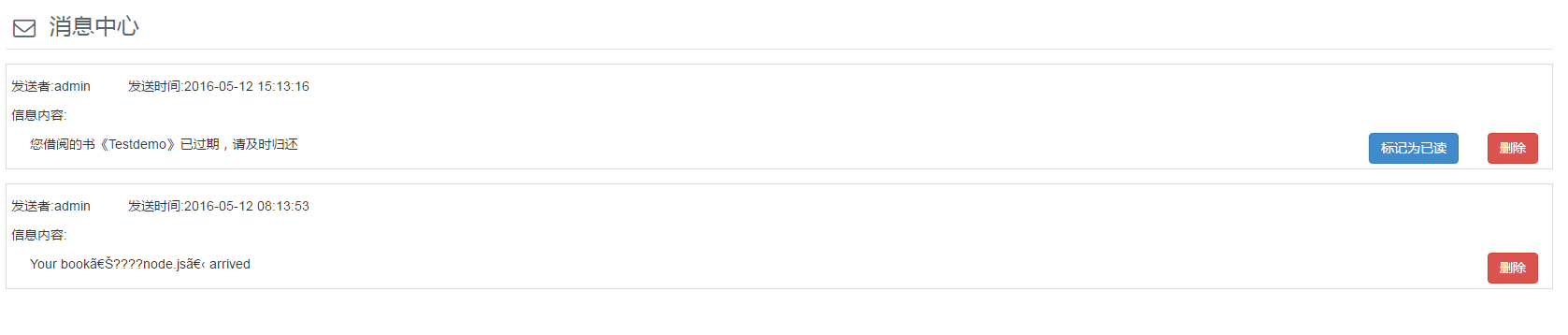


表4-2-12消息中心界面设计示意图

新书上架页面设计：采用多组输入框，对即将上架的新书进行上架操作。页面示意图如下图所示。

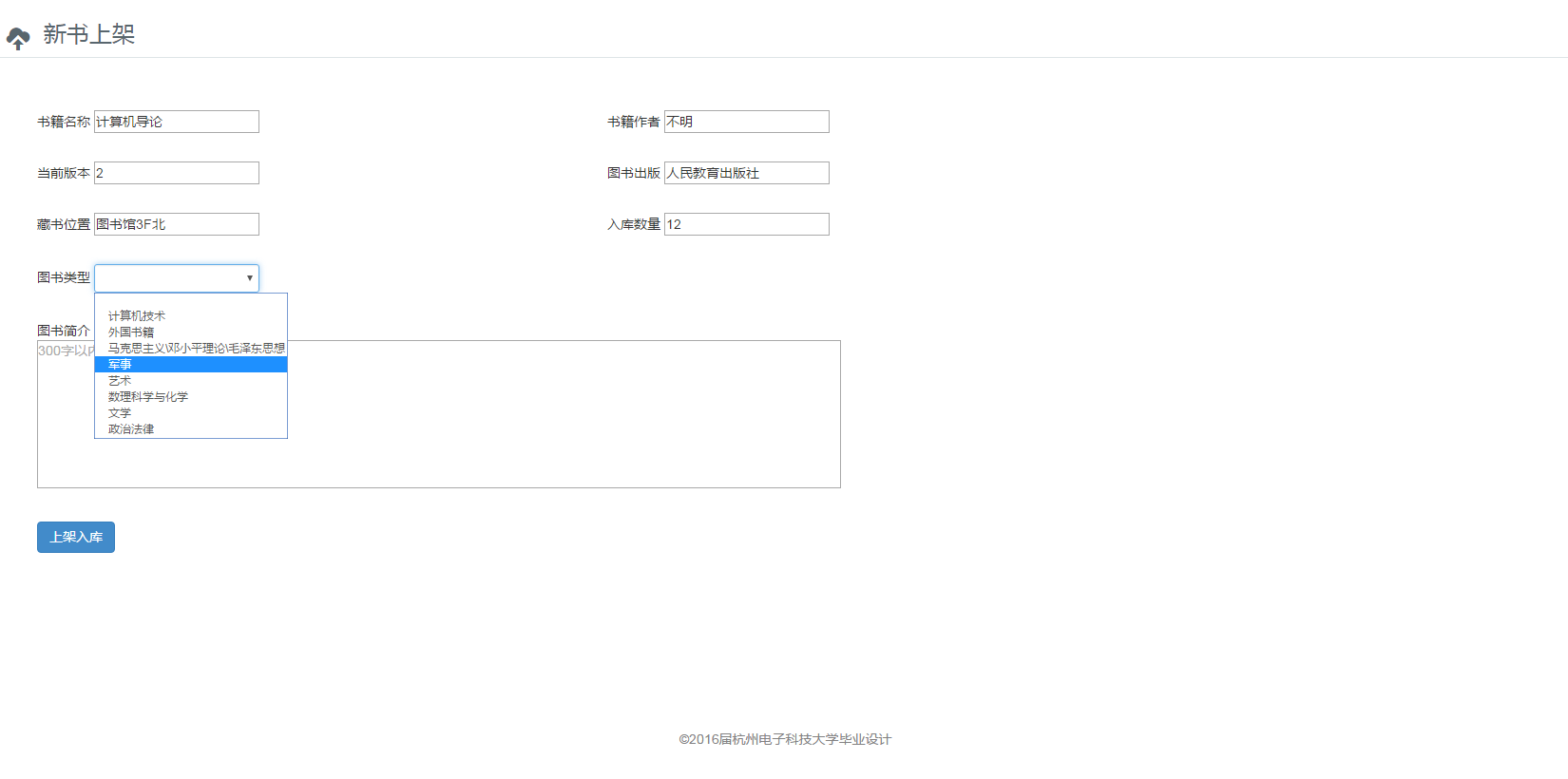


表4-2-13新书上架界面设计示意图