学生选课管理系统

程序设计及操作说明书

设计题目 学生选课管理系统

学生姓名 曹煜政 柴星宇 马智博 宁泊荣

学 号 201593136 201593104

201593033 201593013

专 业 软件工程

班 级 软日1501 软日1502

指导教师 周宽久

完成日期 ： 2018 年 7月19日

目录

[**第一章 绪论** 1](#_Toc519707083)

[**1.1开发背景** 1](#_Toc519707084)

[**1.2系统概述** 1](#_Toc519707085)

[**第二章 系统设计分析** 1](#_Toc519707086)

[**2.1系统需求** 1](#_Toc519707087)

[**2.1.1 总体需求** 1](#_Toc519707088)

[**2.1.2 功能需求** 1](#_Toc519707089)

[**2.2 功能描述** 2](#_Toc519707090)

[**2.3 系统设计** 2](#_Toc519707091)

[**2.3.1 总体设计** 2](#_Toc519707092)

[**2.3.2 模块设计** 3](#_Toc519707093)

[**第三章 系统运行环境** 5](#_Toc519707094)

[**3.1 系统运行环境** 5](#_Toc519707095)

[**第四章 系统功能操作说明** 5](#_Toc519707096)

[**4.1 学生选课系统程序设计** 5](#_Toc519707097)

[**4.2 模块功能设计说明** 6](#_Toc519707098)

[**4.2.1 登录页面** 6](#_Toc519707099)

[**4.2.2 课程表界面** 7](#_Toc519707100)

[**4.2.3 蹭课界面** 8](#_Toc519707101)

[**4.2.4 聊天机器人界面** 8](#_Toc519707102)

[**4.2.5 选课界面** 9](#_Toc519707103)

[**4.2.6 个人信息界面** 10](#_Toc519707104)

[**第五章 系统数据库设计** 16](#_Toc519707105)

[**5.1关系模式设计** 16](#_Toc519707106)

[**5.2 数据库表设计** 16](#_Toc519707107)

[**第六章 系统核心代码及说明** 17](#_Toc519707108)

[**第七章 总结与展望** 21](#_Toc519707109)

**第一章 绪论**

**1.1开发背景**

未来将是信息化社会，应用领域会越来越广泛，而且社会在不断的进步，信息化以更快捷更方便为目标来融入各行各业。学校也不能例外，尤其是高等院校。当前，我国教育行业正在加紧实施信息化建设。对于学生课程的管理，是大学教育管理者们相当重要的一部分。使用计算机管理信息系统，可以减少人力的投入，加大信息的处理效率并且可以降低管理的难度。结合大学生和教务人员的实际条件和需要，我们设计了这套学生选课管理系统。

**1.2系统概述**

学生选课管理系统包括两个部分：第一部分前台，学生通过手机可以完成登录、选课、查询可选课程、查询已选课程的操作、管理个人信息；第二部分后台，管理员可以通过网页注册登录、课程信息维护、学生信息维护、选课信息查询及统计等。

**第二章 系统设计分析**

**2.1系统需求**

**2.1.1 总体需求**

学生通过手机可以完成登录、课程管理（选课、查询可选课程、查询已选课程）、个人信息管理、课表等功能；管理员可以通过网页注册登录、课程信息维护、学生信息维护、选课信息查询及统计等。

**2.1.2 功能需求**

前台学生端基于Andriod开发，功能如下：

1. 课表
2. 课程管理
3. 个人信息管理
4. 聊天机器人
5. 其他（扫描二维码、定位等）

后台管理员端基于web开发，功能如下：

1. 网页注册登录
2. 课程信息维护
3. 学生信息维护
4. 选课信息查询
5. 选课信息统计

**2.2 功能描述**

基于android和web的学生选课系统

Android端包括：

课表模块、选课模块、个人信息模块和聊天机器人等模块。

课表模块包括课表的展示，蹭课功能，删除蹭课功能。

选课模块包括显示课程信息，选课，已选课程删除功能。

个人信息模块包括个人信息展示，编辑信息，物流信息查询功能。

聊天机器人模块包括聊天功能。

其他包括扫描二维码，位置查询，版本信息等功能。

Web端包括：

登陆模块，课程信息模块，学生信息模块，选课信息模块。

登陆模块包括登陆检查功能。

课程信息模块包括搜索和显示课程信息、添加修改和删除课程信息功能。

学生信息模块包括搜索和显示学生信息、添加修改和删除学生信息功能。

选课信息模块包括选课结果的显示与查询功能。

**2.3 系统设计**

**2.3.1 总体设计**

基于android和web的学生选课系统总体设计

设计说明：本阶段在需求分析的基础上，对系统进行总体设计。主要解决实现该系统需求的程序模块设计问题。包括如何把该系统划分成若干个模块、决定各个模块之间的接口、模块之间传递的信息，以及数据结构、模块结构的设计等。在以下的报告中将进行详细的说明。

设计框图，如图所示：

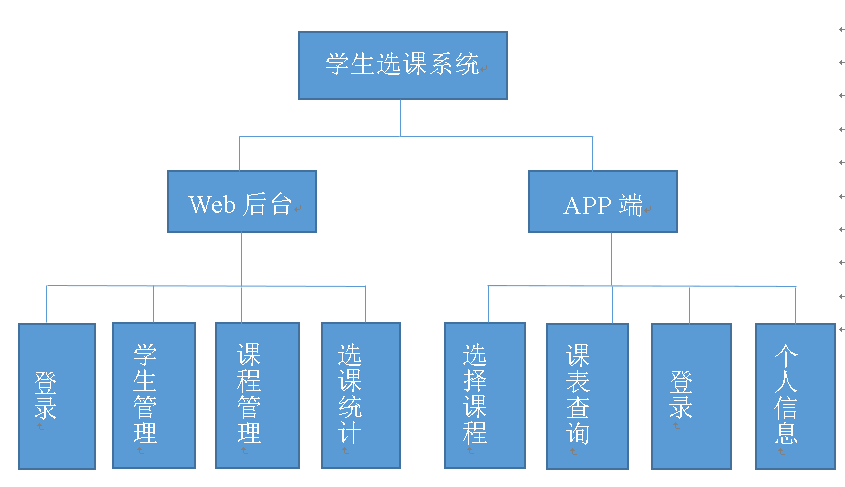


图2-1

**2.3.2 Android模块设计**

登录模块如图2-2所示：

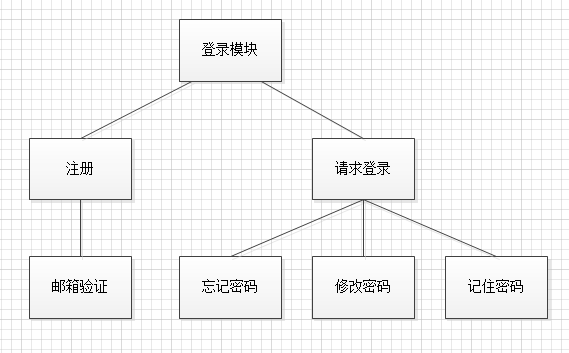


图 2-2

选课模块如图2-3所示：

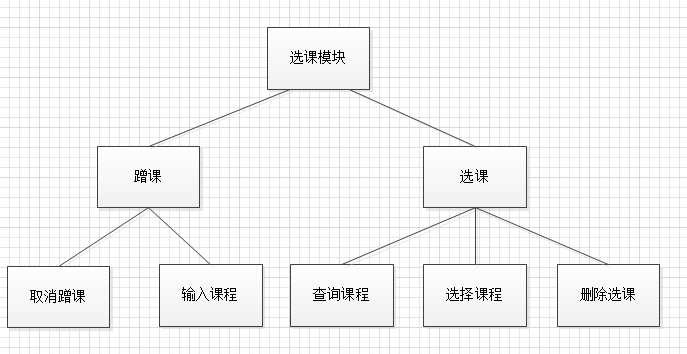


图2-3

**2.3.3 Web后台模块设计**

登陆模块如图所示



图2-4

课程信息模块：



图2-5

学生信息模块：



图2-6

选课信息模块：



图2-7

**第三章 系统运行环境**

**3.1 系统运行环境**

在windows环境下，使用JS与Java开发，运行环境需要有支持JavaScript的浏览器（Chrome浏览器）中访问，手机端支持Android系统。

**第四章 系统功能操作说明**

**4.1 学生选课管理系统程序设计**

**4.1.1学生选课系统设计**

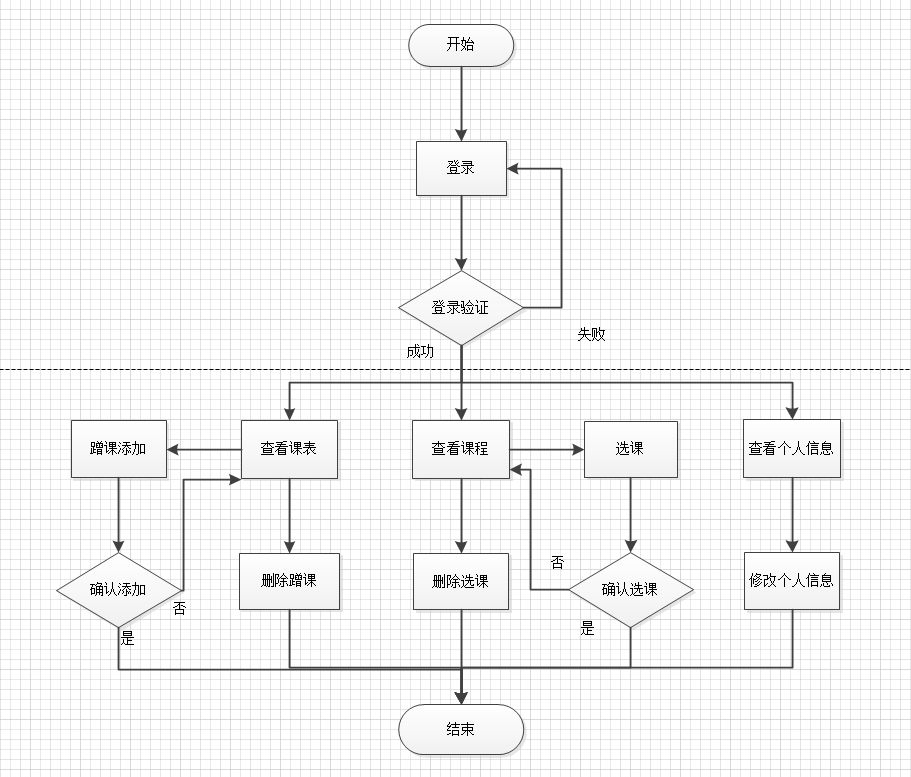
****

图4-1 Android端选课系统程序流程图

**4.1.2 后台管理员系统程序设计**

图4-2 web端管理员系统程序流程图

**4.2 Android模块功能设计说明**

**4.2.1 登录页面**

界面设计：

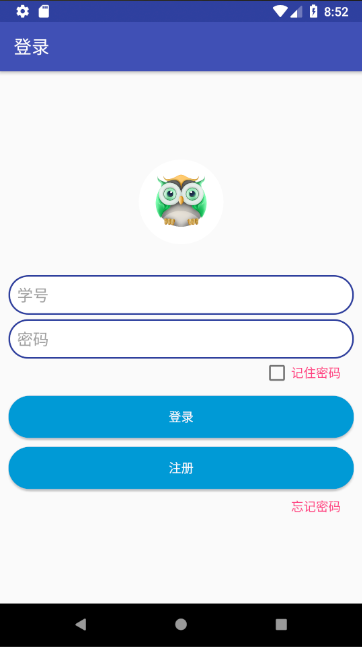
**** 



图4-2 登录界面

页面介绍：

启动应用程序后，进入全屏的闪屏页，首次下载软件有3个起始页面，可选择跳过，之后进入登录页面，可选择登录与注册，登录可以选择记住密码与忘记密码。

**4.2.2 课程表界面**

界面设计

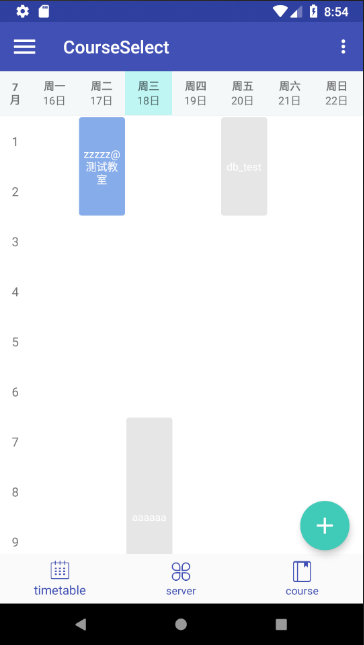
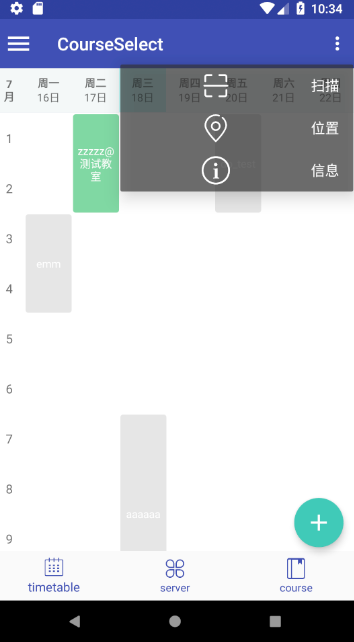
  

图4-3 注册界面

页面介绍：

主界面是课程表界面，可以查看课程，点击课程后进入课程详情页面，点击右上角按钮弹出菜单，点击下方导航栏按钮可进入相应页面，点击后进入相应页面，点击右下角圆形按钮后进入添加蹭课课程界面……。

**4.2.3 蹭课界面**

界面设计

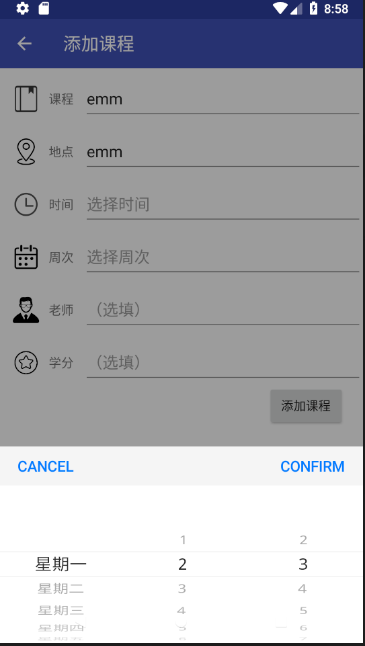
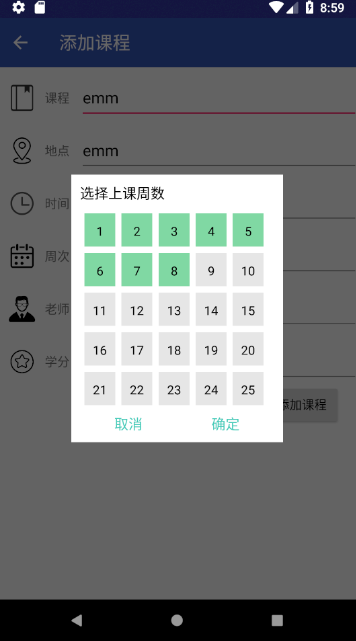
  

图4-4 添加蹭课课程界面

页面介绍：

添加蹭课课程如上所示，添加后可在主界面点击相应课程进行删除蹭课课程。

**4.2.4 聊天机器人界面**

界面设计



图4-5 聊天机器人界面

页面介绍：

点击server按钮进入聊天机器人界面，可以和机器人聊天。

**4.2.5 选课界面**

界面设计

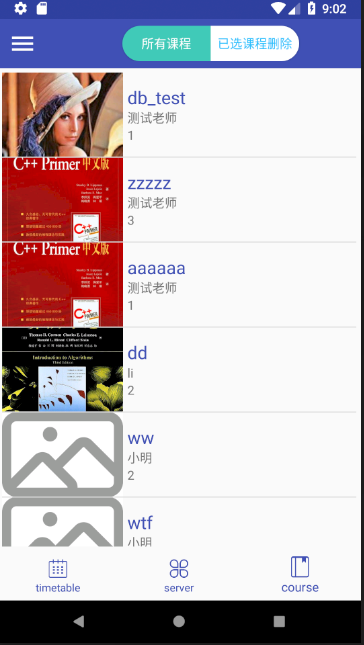
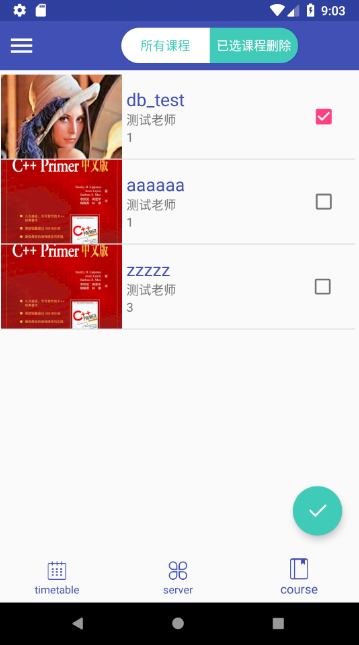
  

图4-6 选课界面

页面介绍：

选课界面标题栏有两个按钮，可以切换所有课程与已选课程删除。在所有课程中点击课程能进入到选课界面进行选课。删除课程点击选项框后再点击下方的确认按钮可以删除已选课程。

**4.2.6 个人信息界面**

界面设计

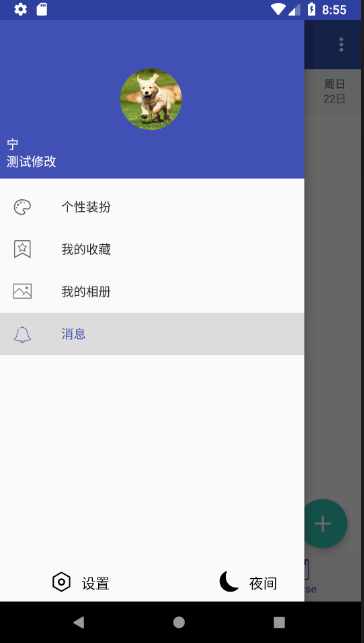
 

图4-6个人信息界面

页面介绍：

点击左上角的按钮可以弹出侧滑菜单，上面有自己的头像，姓名和简介，下方的按钮暂时没有设计，点击头像可以进入个人信息编辑页面，点击下方物流查询可以输入物流单号查询物流信息。

**4.3 模块功能设计说明**

**4.3.1 登陆页面**

界面设计：

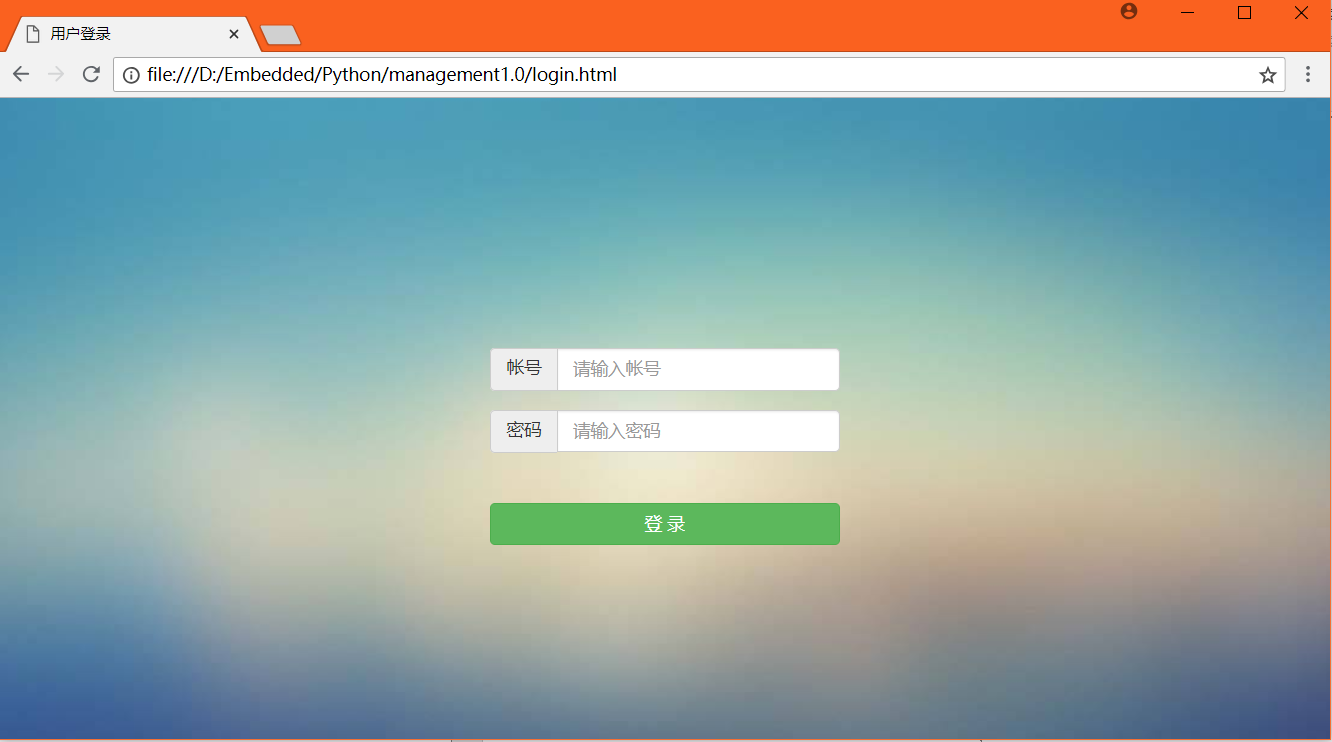


图4-2 登陆界面

页面介绍：

启动应用程序后,进入登陆界面，输入正确的管理员账号和密码，登陆后进入管理界面。

**4.3.2 管理员界面**

界面设计

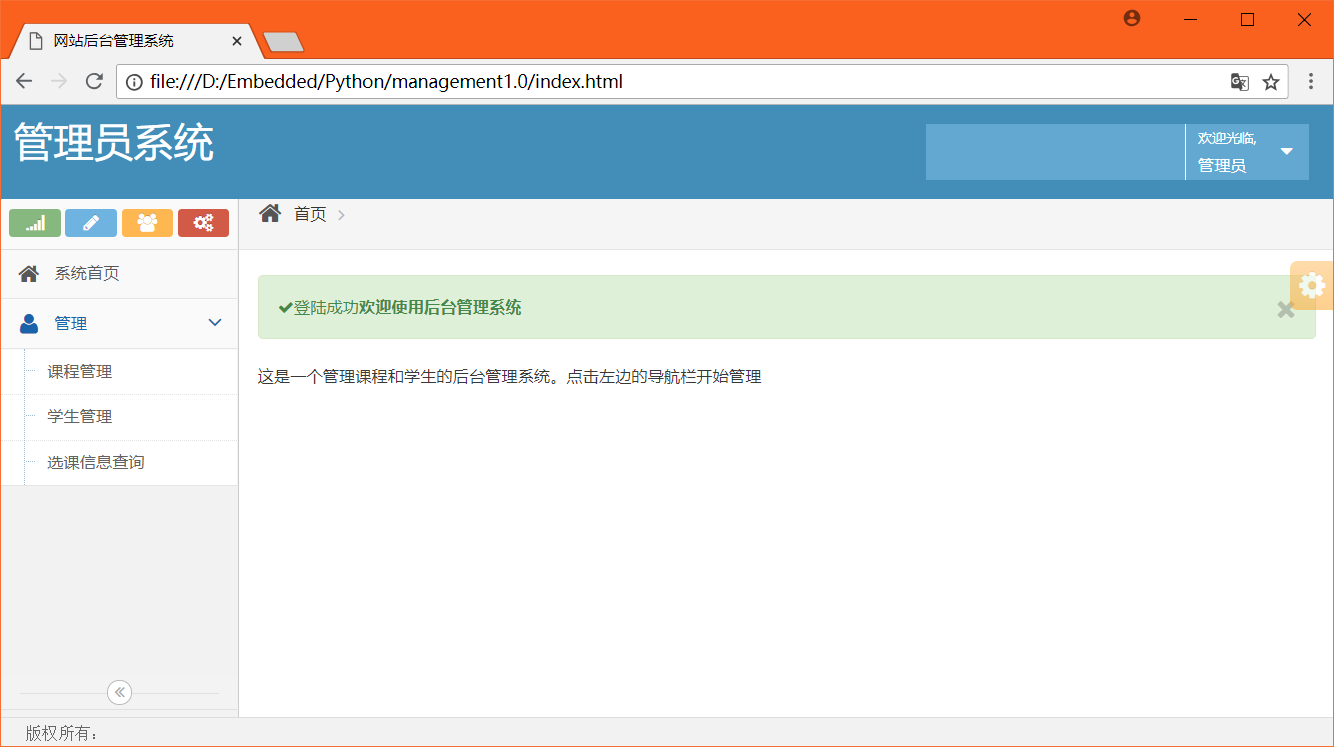


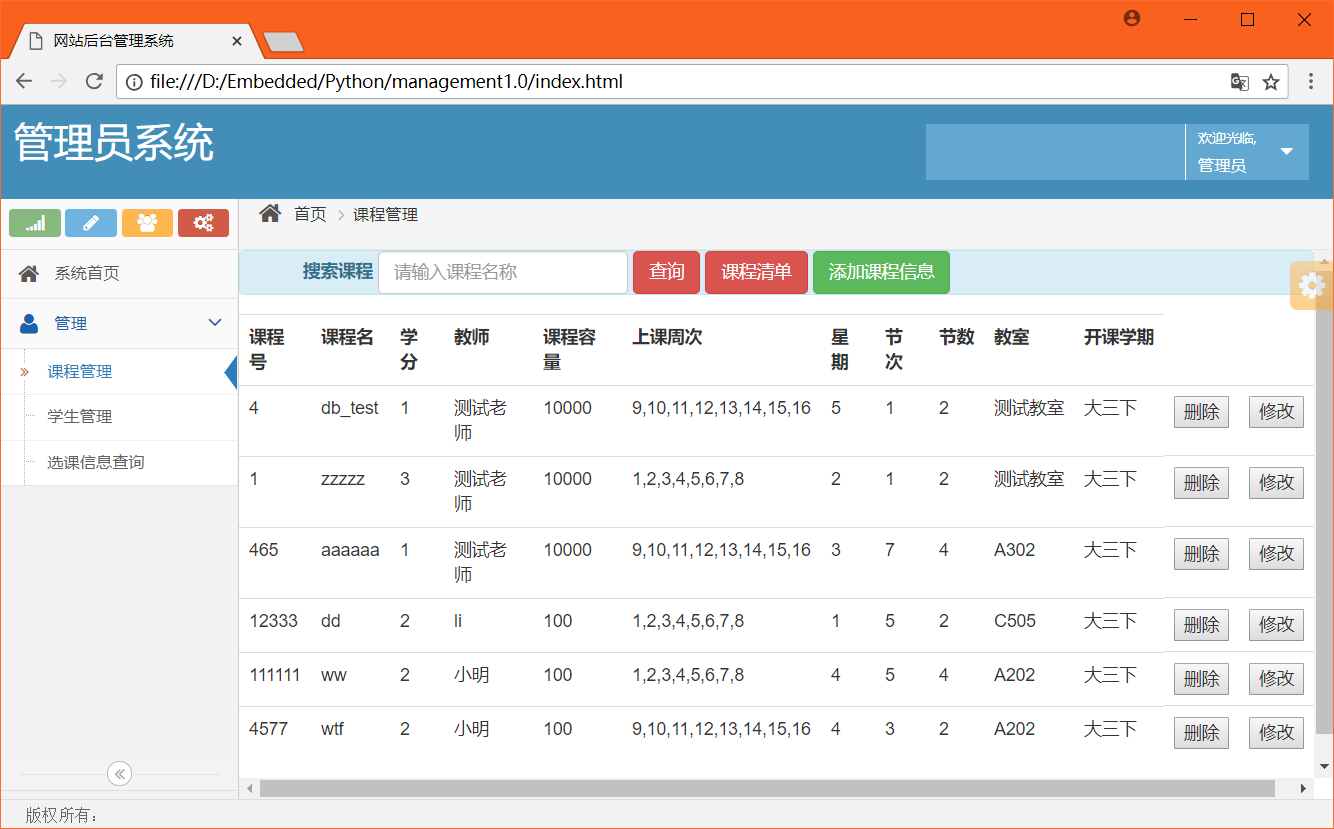
图4-3 管理员界面

页面介绍：

点击管理标签，展开三个子功能的连接标签。点击课程管理 ，进入课程管理页面。点击学上管理，进入学生管理页面。点击选课信息查询，进入选课信息查询页面。

**4.3.3 课程管理界面**

界面设计



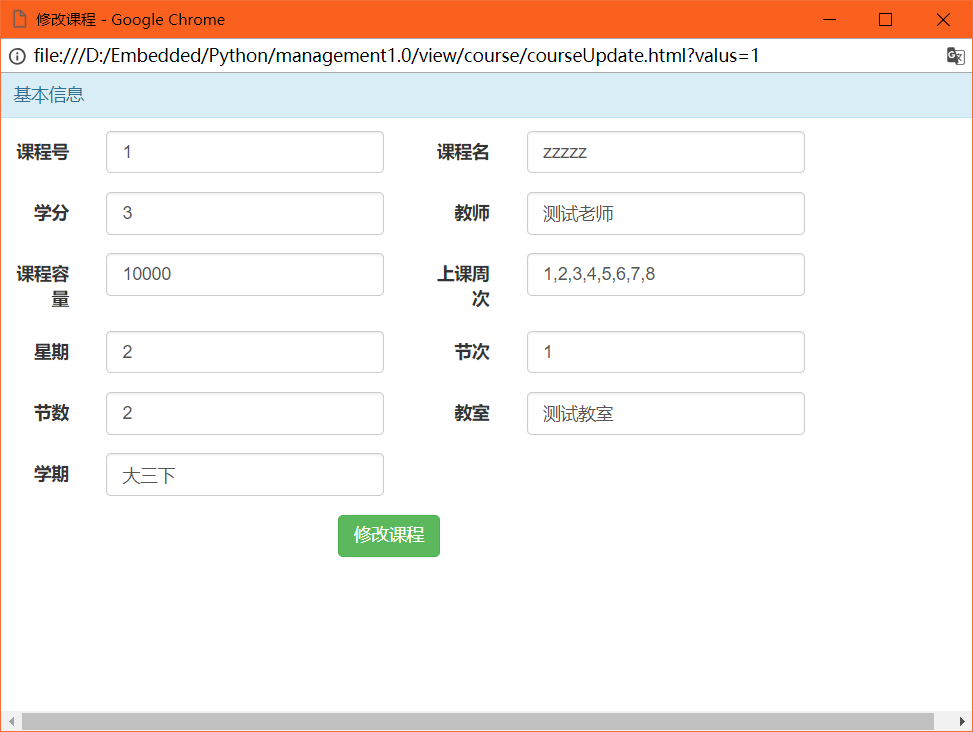
页面介绍：

点击课程清单显示所有的课程信息。在搜索课程输入框输入课程名，显示对应课程信息。点击删除，删除对应行的课程。点击添加课程信息，进入添加课程页面。点击修改按钮，弹出新的修改课程页面。

**4.3.4 添加和修改课程界面**

界面设计



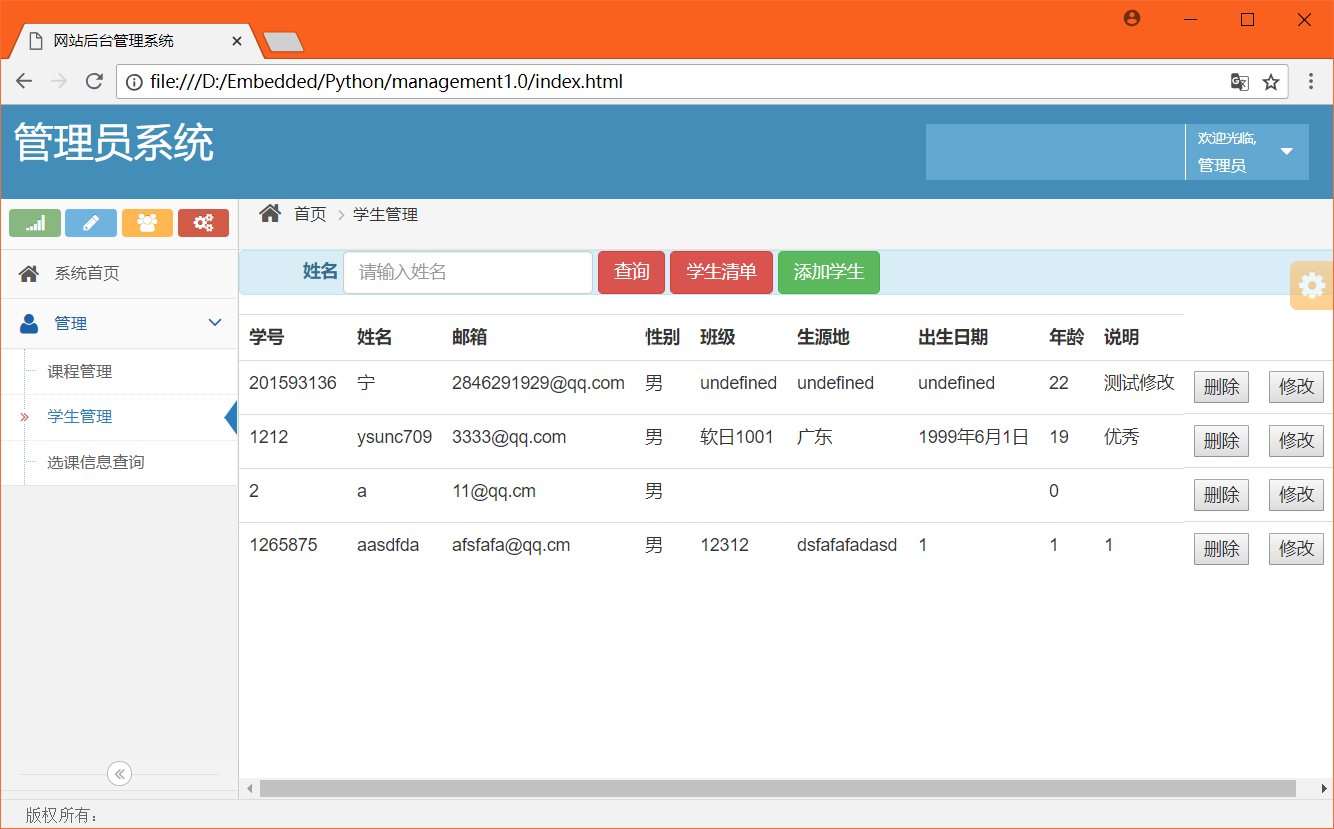


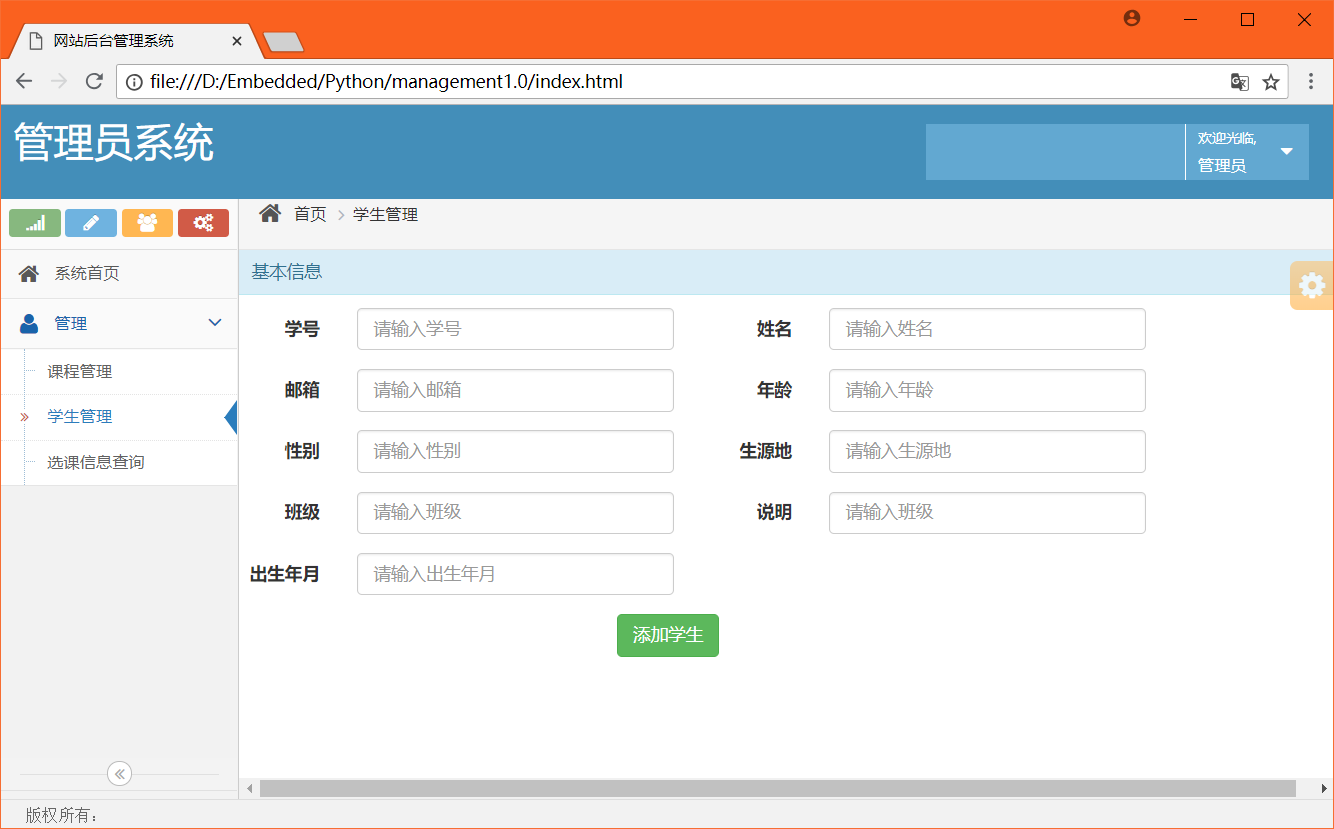
页面介绍：

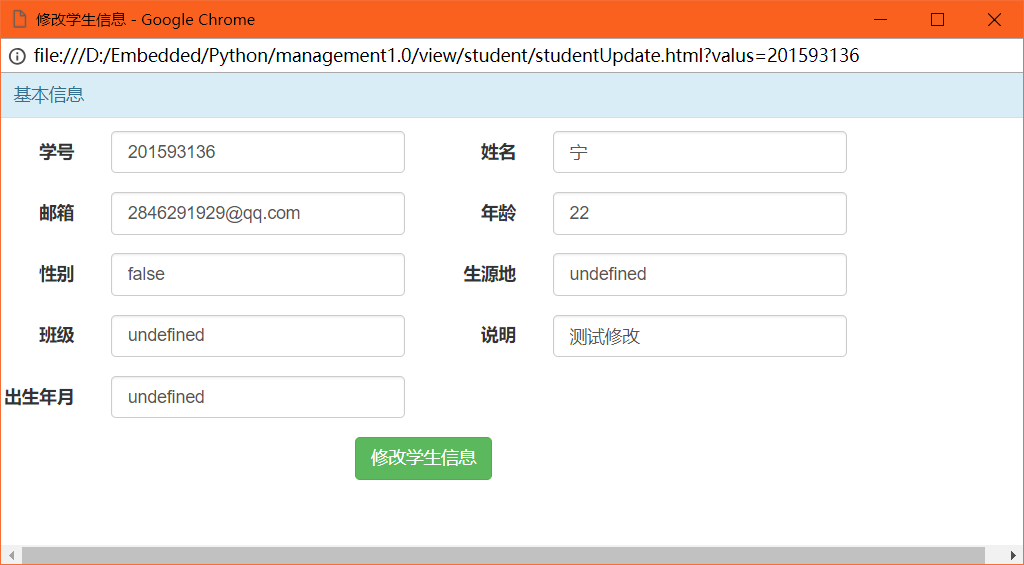
按提示在对应输入框中输入信息，点击添加/修改按钮提交数据。修改界面会显示要修改课程的原始信息。

**4.3.5 学生管理界面**

界面设计





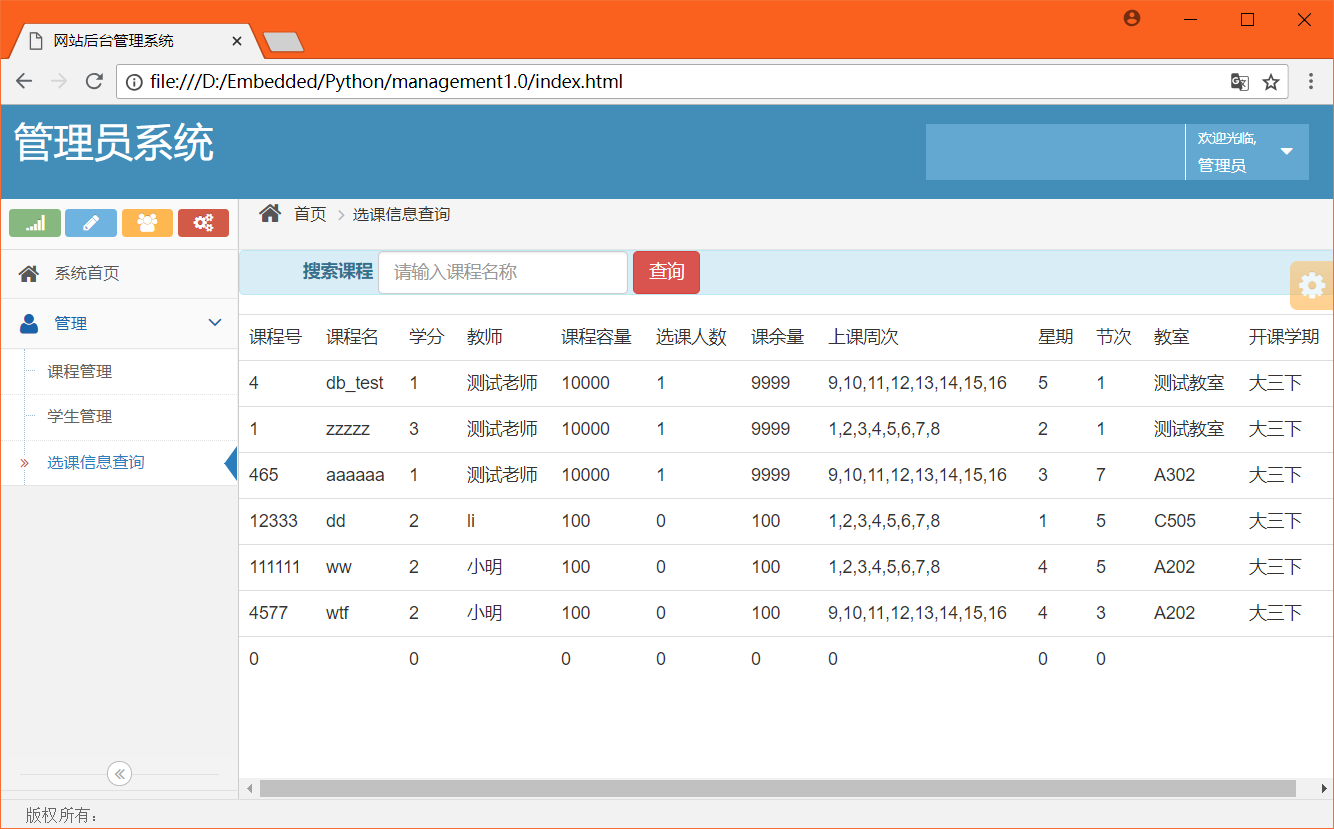


页面介绍：

详细操作和页面跳转同课程管理类似。

**4.3.6 选课信息查询界面**

界面设计



页面介绍：

进入页面会显示选课结果（选课人数，课余量等），在搜索课程的输入框输入想查找的课程名称，显示查询的课程结果。

**第五章 系统数据库设计**

**5.1关系模式设计**

关系模式

1. 学生：（用户ID，姓名，密码，生日，说明，学号，邮箱，性别，班级，地址，年龄，电话号码）为学生实体对应的关系模式，其中ID是主键。
2. 课程：（课程ID，课程名，课程显示颜色，当前时间，学期，老师，开始节次，经过节次，教室，课程图片，星期，学分，选课，周次，课程总量）为课程实体对应的关系模式，其中ID是主键，选课是外键，对应学生表。
3. 管理员：（ID，姓名，密码，邮箱）为管理员对应的关系模式，其中ID是主键。

**5.2 数据库表设计**

根据系统需求分析和数据库设计，我们使用bmob后端云存放数据库，并设计了如下数据库表结构：

（1）学生表（\_user）

用户表如表5-1所示。

表5-1 学生表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 序号 | 字段名 | 字段意义 | 数据类型 | 备注 |
| 1 | objectId | 用户ID | String | Primary key |
| 2 | username | 用户名 | String |  |
| 3 | password | 密码 | String |  |
| 4 | address | 地址 | String |  |
| 5 | age | 年龄 | Number |  |
| 6 | birthday | 生日 | String |  |
| 7 | cname | 课程名 | String |  |
| 8 | desc | 描述 | String |  |
| 9 | gender | 性别 | Boolean |  |
| 10 | phone | 手机 | String |  |
| 11 | sno | 学号 | Number |  |
| 12 | email | 邮箱 | String |  |

（2）管理员表（admin）

用户表如表5-2所示。

表5-2 admin表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 序号 | 字段名 | 字段意义 | 数据类型 | 备注 |
| 1 | objectId | 用户ID | String | Primary key |
| 2 | username | 用户名 | String |  |
| 3 | password | 密码 | String |  |
| 4 | mail | 邮箱 | String |  |

（3）课程表（Mysubject）

用户表如表5-3所示。

表5-3 课程表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 序号 | 字段名 | 字段意义 | 数据类型 | 备注 |
| 1 | objectId | 课程ID | String | Primary key |
| 2 | name | 用户名 | String |  |
| 3 | imgUrl | 图片URL | File |  |
| 4 | colorRandom | 课程颜色 | Number |  |
| 5 | time | 当前时间 | String |  |
| 6 | term | 学期 | String |  |
| 7 | teacher | 老师 | String |  |
| 8 | step | 几节课 | Number |  |
| 9 | start | 开始节次 | Number |  |
| 10 | room | 教室 | String |  |
| 11 | day | 星期 | Number |  |
| 12 | credit | 学分 | Number |  |
| 13 | courseSelect | 选课 | Relation | 学生表外键 |
| 14 | weekList | 周次 | Array |  |
| 15 | subjectNum | 课余量 | Number |  |

**第六章 系统核心代码及说明**

****

以上代码实现从云端数据库查询当前用户的已选课程，并用mTimetableView方法将课程展示在课程表中，学期目前只考虑了当前学期。



以上代码设置了一个对话框用于添加课程到数据库中，并设置课程与用户间的关联关系。

连接云端数据库和初始化数据库操作API。



获取登陆界面的输入信息，并在云端数据库查询，符合条件跳转到主页，不符合弹出提示信息。

**var *MySubject*** = ***Bmob***.Object.extend(**"MySubject"**);  
**var *query*** = **new *Bmob***.Query(***MySubject***);  
 *// 查询所有数据* ***console***.log(**"开始查询"**);  
***query***.find({  
 success: **function**(results) {  
 **var** table=***document***.getElementById(**"count\_tab"**);  
 **var** row0 = table.insertRow(0);*//表头信息* **var** cell0=row0.insertCell(0);cell0.innerHTML=**"课程号"**;

……

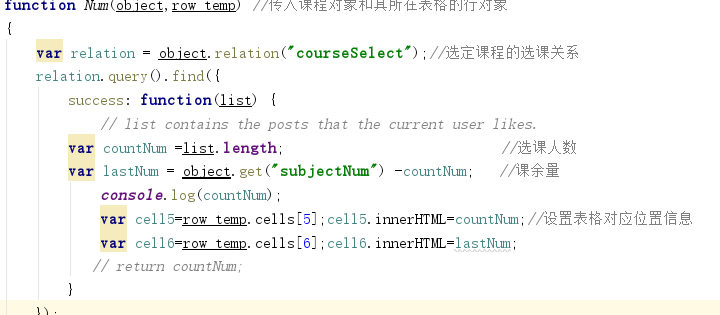
**var** cellB=row0.insertCell(11);cellB.innerHTML=**"开课学期"**;  
 *// 循环处理查询到的数据* **for** (**var** i = 0; i < results.**length**; i++)  
 {  
 **var** object = results[i];*//从所有课程对象的列表中每次取出一个课程* **var** row\_temp=table.insertRow(i+1);*//表格中新一行的对象* **var** countNum,lastNum;  
 cell0=row\_temp.insertCell(0);cell0.innerHTML=object.get(**'subjectId'**);*//设置对应位置信息* ***console***.log(**"查询到："**+i+**"信息，课程号："**+object.get(**'subjectId'**));  
 cell1=row\_temp.insertCell(1);cell1.innerHTML=object.get(**'name'**);

……

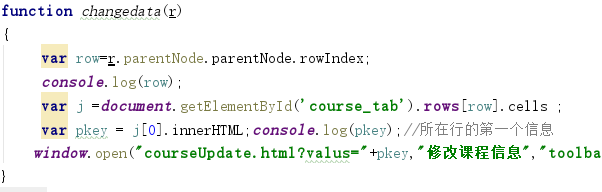
cellB=row\_temp.insertCell(11);cellB.innerHTML=object.get(**'term'**);  
 countNum=*Num*(object,row\_temp);

}  
 },  
 error: **function**(error) {  
 *alert*(**"查询失败: "** + error.**code** + **" "** + error.**message**);  
 }  
 });

从云端数据库获取课程信息并写入网页的对应表格中。调用Num函数查询关联关系获取选课人数和课余量。



查询云端数据库获取选课人数和课余量。



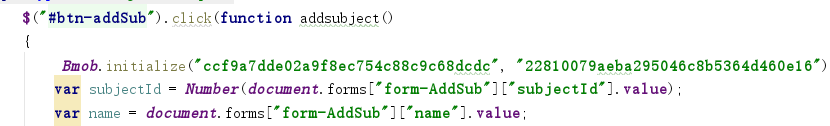
响应修改按钮的函数，获取修改数据所在行的信息，打开修改信息窗口，并向其传入主键值。

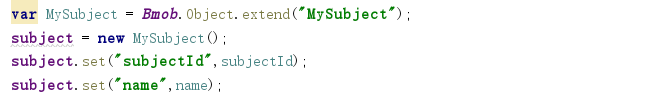


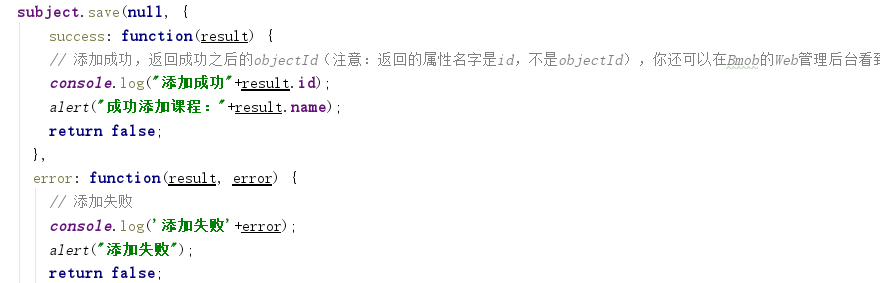
获取页面传来的主键值并返回。



获取搜索输入框的信息，并在所在页面的表格进行查询，只保留符合条件的信息行。







从添加/修改页面填写的表单中获取信息并写入云端数据库。

**第七章 总结与展望**

本次选课管理系统的开发，我们依据需求文档将系统分为了web端和Andriod移动端两部分进行设计，实现了一个较为完整的选课管理系统。这款系统为使用者从教务的管理员到学生，都提供了很大的便利，极大的提高对于大学课程管理的效率。

这是一次完整的开发过程，我们根据系统的应用需求文档划分系统的总体功能和功能模块，并运用系统部署实施环境和实现技术使各功能模块得以实现和运转。根据信息化管理系统的设计理论，进行总体设计、概要设计、详细设计，完成系统的详细和全面的设计，为系统的实现打下基础。开发过程中我们在对需求相关研究的基础上，进一步总结了系统的需求分析、概要设计、实现与完善等环节的工作，并对存在的问题进行了认真全面地分析，整理了相关的资料，一步一步解决问题，最终完成了我们的系统，实现了提供给用户一个高效选课管理系统的目标。

本次选课管理系统的开发我们完成了相关的需求，但是受限于时间，技术等因素此管理系统的功能有待添加，我们只是完成了需求所需的功能，而对于移动端的app用户体验，功能丰富上还有待提高。

在今后的系统发展阶段，我们还有许多工作要做。由于用户的需求不断变化，我们应当继续完善丰富我们的项目功能，以做出更加合理地一套系统框架，加强系统的健壮性和实用性。