一丶需求分析

1. 引言

本次课程软件课程设计的目的是通过独立地完成一个工作量适当的软件系统使学生较系统地实践从选题、需求分析、概要设计、算法设计、代码编写、调试测试、总结报告等基本过程，加深对数据结构、操作系统、程序设计等相关课程理论知识的分析和理解。通过实践引导学生由原理分析向工程设计过渡，掌握软件系统设计的基本方法和一般规则，提高综合应用所学理论知识的能力，同时培养学生的创新思维和实践能力，发挥自我能动性和创造力，为后续专业课程的学习打下坚实的基础。

这次，我们组做的任务是设计一个可以将课表上的课程信息录入到数据库，共享课程信息的学籍管理系统，我们选择了Html css js php和MySQL编程语言合作开发完成设计任务。

1. 项目描述

实现功能有注册、登录功能，系统总共有三种身份的人登录，分别是老师，学生，管理员；其中只有学生（S开头）有注册功能（包括学院、专业、年级、班级等信息），老师的账号（T开头）由系统管理员分配，管理员的账号（\_1000）由系统设计者指定。管理员具有最高权限，可以管理课程信息（包括录入和修改等），可以管理老师和学生的账号信息。老师可以修改自己的课程信息，管理自己账号，如修改密码，修改名称之类的基本功能。学生可以查看自己的课程信息，管理自己账号，如修改密码等基本功能。

1. 具体页面操作

(1). 学生账号为数据库初始添加，登录时填入学号（S开头）和密码（初始密码为123456）.登录后可查看课表。点击个人栏可查看个人信息，修改后点击修改即可修改信息（学号默认不可修改），点击退出退出学生界面返回登录界面.

(2). 老师账号为数据库初始添加，登录界面和学生一致。登录时填入老师工号（T开头）和密码（初始密码为123456）.登录后可查看课表，点击修改课表，选择时间后确认即可显示当前时间的课表，修改后点击确认，即可修改课表。点击个人栏可查看个人信息，修改后点击修改即可修改信息（工号默认不可修改），点击退出退出老师界面返回登录界面.

(3). 管理员账号为初始默认，不可修改，登录账号为（\_1000）和密码（123456）.登录后选择院系专业班级可查看课表和修改课表。点击老师栏选择院系可查看老师信息，点击修改修改信息后选择确定即可修改信息，点击学生选择院系专业班级后即可显示班级学生列表，修改学生信息如修改老师信息类似。点击退出界面返回登录界面.

二丶系统概要设计

1. 性能需求

（1）系统易操作性。所开发的系统应做到操作简单，尽量使系统操作不受用户对电脑知识水平的限制。

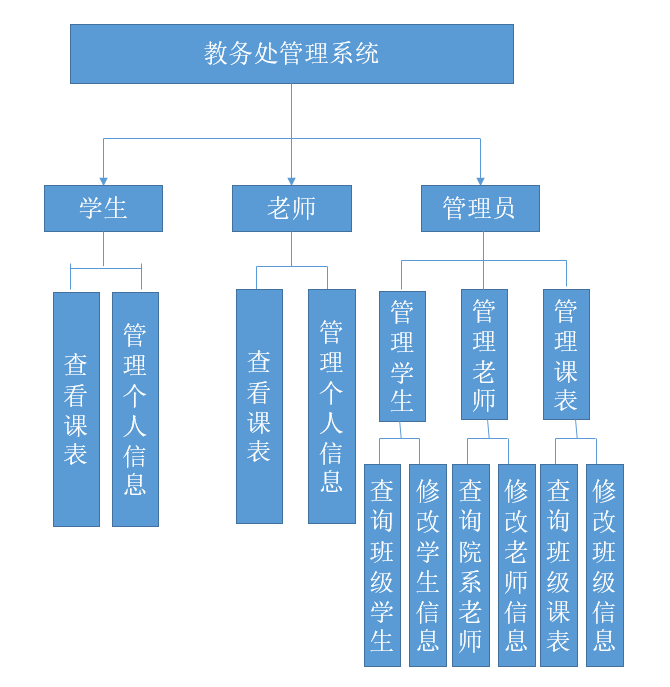
（2）系统具有可维护性。由于系统涉及的信息比较广，数据库中的数据需定期修改，系统可利用的空间及性能也随之下降，为了使系统更好地运转，学院可以对系统数据及一些简单的功能进行独立的维护及调整。

（3）系统具有开放性。该系统能够在开放的硬件体系结构中运行，并且能与其他系统顺利连接，不会因外部系统的不同面要做在量的修改工作。

2. 系统的模块划分

课程管理系统分为三个模块，分别为学生模块、老师模块和管理员模块。根据系统中所包含的内容，该系统可以分为两个子系统，分别为查询系统和管理子系统。查询分为查询课表、查询老师、查询学生。管理分为管理课表、管理学生、管理老师、管理个人信息。

3.系统功能模块设计图



三丶数据与接口规范  
用户登陆：用户标示采用'session'这个存储API，用户发送账户密码后，后台用"session"在用户本地创建一个用户标识，每次传输数据时后台读取用户标识来决定发送的数据<br>注销用户：当用户点击 "退出" 按钮的话，前端控制消除'session',以完成用户退出。name的值为userId.  
后台设置  
sessionStorage.setItem(name,value);  
用户点击退出出，前端执行删除  
sessionStorage.removeItem(name);  
课表说明  
数据库只存在当前学期课表  
数据说明  
所有的原始数据信息都由数据库文件导入以初始化。管理员只拥有管理现有数据的能力。  
数据格式  
json  
接口规范  
**说明事项：**  
- 所有请求都由 'post' 完成。(不采用get,以避免混乱.post 更为安全，且数据传输无上限)    
- 管理员和老师不需要注册界面。**\*\*老师学生管理员三者账户默认不可修改\*\***。  
- 学生S开头。学生信息包括，用户名userName，账号userCode，密码password，学院academy，专业major，年级grade，班级class，邮箱email(未来密码靠找回靠邮箱验证，现在留待扩展，遗失密码，找管理找客服)  
- 老手T开头.  老师信息包括。用户名userName，账号userCode，密码password，学院academy，邮箱email(未来密码找回靠邮箱验证，现在留待扩展，遗失密码，找管理找客服)  
- 管理员默认登录号码为\_1000，密码为123456。  
**文件管理规范：**  
前端的url既可以确定为 诸如学生 ../php/student.php  
```  
--loginIn.html   //登陆界面,唯一的入口。内嵌注册的弹窗  
--php        
  --student.php  
  --teacher.php  
  --admin.php         
--template  
  --student.html  
  --teacher.html  
  --admin.html      
--script  
  --student.js  
  --teacher.js  
  --admin.js         
--css    
  --student.css  
  --teacher.css  
  --admin.css         
--images  //图片  
**后台传输格式**<br> 后台返回所有数据包括2个。一个是statusCode为布尔值**\*\*必传，**0为出错，1为正确。第二个data，存储所有的业务相关数据。如果出错的话只需要发 statusCode:0  
  
```  
 {  
 statusCode:1/0,  
 data:{}  
 }  
```  
  
**前端传输格式**前端发送数据，包括2个，一个是请求的事件。一个是传输的数据  
```  
 {  
  eventType:  {  //如果看不懂，可以直接理解下面每次的数据交互过程即可  
   //loginIn成功后默认跳转课表页面，学生和老师返回当前学期课表数据'getClass'事件，    
   //管理员返回全校院系专业班级信息'getSelectData'。  
   // 学生与老师事件说明 注册事件'register',独有。getInfo，modifyInfo。老师学生共有。老师增加一个文件上传的功能  
  
    register/getClass/getInfo/modifyInfo                          //学生  
    getClass/modifyClass/getInfo/modifyInfo           //老师  
    getSelectData/getclass/modifyClass/getStudentList/getTeacherList/modifyStudentInfo/modifyTeacherInfo    //管理员    
    }  
 data: {     //存储传输的数据  
  
  }  
  
}  
```  
学生部分

学生注册，请求数据  
post请求 返回所有年纪，专业及其对应的班级数。      
  
  前端请求格式  
  
```  
  {  
    eventType:register;  
  }  
```  
  
  后台返回格式  
  
```  
data:[  
  {academy:"电信学院"，  
   majors:[{major:通信工程，  
            grades:[ {grade:12,classNum:6},{grade:13,classNum:6},{grade:13,classNum:6}]}  
           {major:电子信息，  
            grades:[ {grade:12,classNum:6},{grade:13,classNum:6},{grade:13,classNum:6}]}  
          ]  
  }  
  {academy:"电信学院"，  
   majors:[{major:通信工程，  
            grades:[ {grade:12,classNum:6},{grade:13,classNum:6},{grade:13,classNum:6}]}  
           {major:电子信息，  
            grades:[ {grade:12,classNum:6},{grade:13,classNum:6},{grade:13,classNum:6}]}  
          ]  
  }  
     ]  
```  
  
学生注册发送注册数据  
学生填完表格后发送Post请求  
  
  传入json  
  
```  
   {  
   eventType:register;  
   data:{  
    userName:””,  
    userCode:””,  
    password:””,      
    academy:"",  
    major:"",  
    grade:””,  
    class:””,  
    email:"",  
    }  
   }  
```  
  
  返回    
  
  statusCode  
  
  0 表示 参数错误    
  
  1 表示 正确无误    
  
```  
   {  
   statusCode:"0",//创建失败，重新注册  
       }  
```  
  
  或者  
  
```  
  
   {  
   statusCode:”1”;//注册成功，返回1，前端判断后转入到登陆的页面  
   }  
```  
  
学生登录  
填完表格 判断为学生后,发送post请求  
  
```  
    {  
    eventType:loginIn;  
    data:{  
      uesrName:"",  
      password:""}  
    }  
```  
  
失败:  
  
```  
{  
  statusCode:0;  
}  
```  
  
成功：  
  
```  
  {  
    statusCode:1;  
  }  
```  
登陆成功后请求课表  
前端：  
```  
{  
  eventType:getClass;  
}  
```  
后台：发送课表过来  
```  
{  
  statusCode:0/1,  
  data:[{  
    class:"模电",  
    teacherName:"wislon",  
    time:"23(星期三第三节课(星期一为0))",  
    duration:"3-8(第三周到第八周)",  
    place:"东九B302"}，  
    {  
      class:"模电",  
      teacherName:"wislon",  
      time:"23(星期三第三节课(星期一为0))",  
      duration:"3-8(第三周到第八周)",  
      place:"东九B302"}，  
    {以此类推....}]  
  }  
```  
  
点击'个人'按钮后  
前端向后台请求学生信息数据.  
  
```  
{  
  eventType:'getInfo',  
}  
```  
  
  后台返回  
  
```  
{  
    statusCode:1,  
    data:{  
     userName:””,  
     userCode:””,  
     password:””,      
     academy:"",  
     major:"",  
     grade:””,  
     class:””,  
     email:"",  
     }  
  
  }  
```  
  
学生修改信息  
前端上传所有信息。  
  
```  
{  
    eventType:modifyInfo;  
    data:{  
     userName:””,  
     userCode:””,  
     password:””,      
     academy:"",  
     major:"",  
     grade:””,  
     class:””,  
     email:"",  
     }  
  
  }  
```  
  
后台返回  
  
```  
{  
  statusCode:'1/0'  
}  
```  
  
--------------------------------------------------------------------------------  
  
老师部分

老师登录  
传入  
  
```  
    {  
    event:'loginIn';  
    data:{  
      userCode:"",  
      password:""}  
    }  
```  
返回  
  
```  
{  
  statusCode:0/1;  
  
}  
```  
  
  
登陆成功后即跳转页面  
前端：  
```  
{  
  eventType:getClass;  
}  
```  
后台：发送课表过来  
```  
{  
  statusCode:0/1,  
  
  }  
```  
  
请求课表  
  
```  
{  
  eventType:getClass;  
}  
  ```  
```  
{  
 statusCode:1,  
 data: [{  
   class:"模电",  
   time:"23",  
   duration:"3-8",  
   academy:"电信",  
   major:"通信工程",  
   grade:"2013",  
   classNo:"3-4-5",  
   place:"东九B302"},  
   {  
     class:"模电",  
     time:"23",  
     duration:"3-8",  
     academy:"电信",  
     major:"通信工程",  
     grade:"2013",  
     classNo:"3-4-5",  
     place:"东九B302"}  
   ]  
  
 };  
```  
  
老师查看个人信息  
点击个人信息，前端发送事件请求  
  
```  
{  
  eventType:getInfo,  
  }  
```  
  
后台传回数据  
  
```  
{  
  statusCode:1,  
  data:{  
    userName:，  
    userCode:，  
    password:，  
    academy:,  
    email:  
  
  }  
}  
```  
  
修改个人信息，  
前端发送  
  
```  
{  
  eventType:modifyInfo;  
  data:{  
    userName:，  
    userCode:，  
    password:，  
    academy:,  
    email:  
  }  
}  
```  
  
管理员部分

管理员登录  
填完表格 判断为管理员后,发送post请求<br>json如下<br>传入  
  
前端发送  
  
```  
    {  
    eventType:'loginIn';  
    data:{  
      uesrName:"".  
      password:""  
  }  
    }  
```  
  
后台返回  
  
```  
{  
  statusCode:1,  
}  
```  
请求院系专业数据  
前端格式  
```  
{  
  eventType:getSelectData,  
}  
  ```  
后台返回格式  
  
```  
{  
  statusCode:1/0,  
  data:[  
  {academy:"电信学院"，  
   majors:[{major:通信工程，  
            grades:[ {grade:12,classNum:6},{grade:13,classNum:6},{grade:13,classNum:6}]}  
           {major:电子信息，  
            grades:[ {grade:12,classNum:6},{grade:13,classNum:6},{grade:13,classNum:6}]}  
          ]  
  }  
  {academy:"电信学院"，  
   majors:[{major:通信工程，  
            grades:[ {grade:12,classNum:6},{grade:13,classNum:6},{grade:13,classNum:6}]}  
           {major:电子信息，  
            grades:[ {grade:12,classNum:6},{grade:13,classNum:6},{grade:13,classNum:6}]}  
          ]  
  }  
     ]  
}  
```  
管理员查看课表  
 前端发送  
  
```  
 {  
   eventType:'getClass';  
   data:{  
     academy:"",  
     major:"",  
     grade:””,  
     classNum:””,  
   }  
 }  
```  
  
后台返回  
  
```  
{  
  statusCode:'1',  
  data:{  
    //这个班当前的课表  
  }  
}  
```  
  
管理员修改课表（添加请到数据库导入文件添加）  
post传入  
  
```  
    {  
    eventType:'modifyClass',  
    data:[{  
      classId:"0(课程编号,修改课表时使用)"  
      teacherName:"郭教授",  
      className:"模电"(防止老师上不同的课),  
      time:"23(星期三第三节课(星期一为0))",  
      duration:"3-8(第三周到第八周)",  
      academy:"",  
      major:"",  
      grade:””,  
      class:"3-4-5(防止有10班的存在)"  
      place:"东九B302"},  
    {以此类推....}]  
        }  
    }  
```  
  
返回StatusCode  
  
```  
{  
  statusCode:1,  //修改成功  
}  
```  
  
```{  
    statusCode:0,  //修改失败  
  }  
```  
  
管理员查看学生列表  
前端发送  
  
```  
{  
  eventType:'getStudentList',  
  data:{  
    academy:"",  
    major:"",  
    grade:””,  
    class:””,  
  }  
}  
```  
  
后台返回  
  
```  
{  
  statusCode:'1',  
  data:[{      
      userName:””,  
      userCode:””,  
      password:””,      
      academy:"",  
      major:"",  
      grade:””,  
      class:””,  
      email:"",},  
      {  
        userName:””,  
        userCode:””,  
        password:””,      
        academy:"",  
        major:"",  
        grade:””,  
        class:””,  
        email:"",  
        }]  
       //学生信息，包括所有的学生数据  
  
}  
```  
  
管理员修改学生信息  
前端发送  
  
```  
{  
  eventType:'modifyStudentInfo',  
  data:{  
    userName:””,  
    userCode:””,  
    password:””,      
    academy:"",  
    major:"",  
    grade:””,  
    class:””,  
    email:"",  
  }  
  
}  
```  
  
后台返回  
  
```  
{  
  eventType:'modifyTeacherInfo',  
  data:{  
    userName:，  
    userCode:，  
    password:，  
    academy:,  
    email:  
  }  
  
}  
```  
  
管理员查看老师列表  
前端传输  
  
```  
{  
  eventType:'getTeacherList',  
  data:{  
      academy:"",  
  }  
}  
```  
  
后台返回  
  
```  
{  
  statusCode:'1',  
  data:[  
     { userName:，  
    userCode:，  
    password:，  
    academy:,  
    email:},  
    { userName:，  
   userCode:，  
   password:，  
   academy:,  
   email:}  
  ]  
  }  
```  
  
管理员修改老师信息  
前端传输  
  
```  
{  
  eventType:'modifyTeacherInfo',  
  data:{  
    userName:，  
    userCode:，  
    password:，  
    academy:,  
    email:  
  }  
}

四丶我的职责

我所负责的是网站前端的设计，采用html,css,javascript。这次的制作抛弃了所有的框架与类库。采用前后端分离，ajax异步请求数据。所有任务也并不轻。

从商量网站的架构，到设计交互文档。整个过程还是十分艰辛的。同时，这也是我第一次，完整的走过整个流程。

前端的页面采用统一入口，一个登陆页面，根据用户账号进行判断，导向不同的页面。不同的页面之中，大量复用，减轻了不少工作量。同时，三套页面，老师，学生，系统管理员。由难到易，学生界面最为简单，管理员则稍显复杂。

在如何能够以最少代码实现最多功能这个点上，也思虑颇多。最后，终于完成了这次课设。

五、系统详细设计

根据设计要求设计出各个表结构

表4-1 各个表结构如下所示

学生表：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 列名 | 数据类型 | 长度 | 说明 | 备注 |
| id | char | 8 | 学生帐号 | Primary key |
| userName | char | 50 | 学生姓名 | Not null |
| Password | char | 50 | 密码 | Not null |
| major | Char | 50 | 学院 | Not null |
| class | Char | 50 | 班级 | Not null |
| academy | Char | 50 | 院系 | Not null |
| grade | Char | 50 | 年纪 | Not null |
| Email | Char | 50 | 邮箱 | Not null |

老师表：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 列名 | 数据类型 | 长度 | 说明 | 备注 |
| id | char | 8 | 学生帐号 | Primary key |
| teacherName | char | 50 | 学生姓名 | Not null |
| Password | char | 50 | 密码 | Not null |
| academy | Char | 50 | 院系 | Not null |
| Email | Char | 50 | 邮箱 | Not null |

院系表：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 列名 | 数据类型 | 长度 | 说明 | 备注 |
| major | Char | 50 | 学院 | Not null |
| classNum | Char | 50 | 班级 | Not null |
| academy | Char | 50 | 院系 | Not null |
| grade | Char | 50 | 年纪 | Not null |

课程表：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 列名 | 数据类型 | 长度 | 说明 | 备注 |
| lessonName | Char | 50 | 学院 | Not null |
| place | Char | 50 | 班级 | Not null |
| time | Char | 50 | 院系 | Not null |
| Duration | Char | 50 | 年纪 | Not null |
| teacherName | Char | 50 | 老师姓名 | Not null |
| Major | Char | 50 | 专业 | Not null |
| Grade | Char | 50 | 年纪 | Not null |
| Class | Char | 50 | 上课的班级 | Not null |
| teacherId | Char | 50 | 上课老师id | Not null |

六丶使用结果

测试结果，系统完成了要求的各项功能，满足以下约束：要求各个测试内容必须按照严格的字符类型，输入其他格式将报错；字符长度按照下表给出的严格测试，超出范围报错；用户账号非空，当输入这项为空时拒绝插入或修改；用户账号唯一，插入已存在账号时提示错误。

七丶总结

通过这次课程设计，我了解到了课程设计主要的目标是利用课程中学到的软件工程知识和技术较好的开发设计出应用系统，去解决各行各业信息化处理的要求。完成这个设计可以巩固我们对软件工程理论的理解，进一步提高我们综合运用所学知识的能力。  
　　我们这组要做的是课程信息管理系统，开始时，我们并没有着手写程序。而是大家一起商量这个系统需求分析、概要设计、详细设计。当这些都准备好了之后，我们进行模块的分工。每个人都有自己的模块设计，而且写出来的代码要求可以实现相应模块的功能，得到理想的效果。当每个人都把自己的分工做好了，最后会由一个人把这些全部组合搭建在一起。这样就组成了一个完整的系统。我所负责的模块是个人信息管理系统中个人日记管理模块的设计。自己刚知道自己题目的时间还是对要做的系统不是很了解，自己不知道具体的功能是什么，我就先从需求分析开始，一步一步地进行分析，最后明白了自己的工作内容。但是在这个过程中也遇到了一些问题，起先要确定系统要划分为几个模块，E-R图的设计。这些问题自己都是和组员一起商量得出的最后结果。最后，通过老师和同组组员的帮助，我顺利完成了这次课程设计我所负责的模块的设计。从这次课程设计中，我深刻的体会到了团队合作的重要性，大家需要多交流和沟通才能更好的去完成老师所布置的任务。同时，在这次设计中，郭老师也对我们进行了很多耐心和细心的讲解，使我们的课程设计能够顺利的完成。再次特别感谢郭老师的细心指导和同组组员的帮助。

参考文献

1. 王珊, 萨师煊. [数据库系统概论](http://www.wl.cn/Product/BookInTag.aspx?tagName=数据库系统概论)(第四版)[M]. 北京: 高等教育出版社, 2006.
2. 张海藩.软件工程导论(第五版).清华大学出版社, 2008年2.
3. [1] Head First HTML与CSS、XHTML（中文版）[Elisabeth Freeman](http://book.douban.com/search/Elisabeth Freeman) / [Eric Freeman](http://book.douban.com/search/Eric Freeman)  中国电力出版社
4. [2] http://www.w3school.com.cn/
5. [3] PHP+MySQL+Dreamweaver动态网站建设从入门到精通 陈益材 机械工业出版社