老师好，下面开始我们组的项目介绍。我们的项目是图书管理系统，有三个用户对象，分别是读者、图书管理员和后台管理员，所以我们的项目更像一个图书馆系统。我们项目的名称是LocaLibrary，是我们的专业——Localization和Library的结合。

**需求分析**

**概念结构设计：**

分析图书管理系统的系统需求，将系统中涉及的人、物进行抽象，得到的系统实体如下：

读者实体集。其属性有：借书卡号，姓名，性别，学院，班级，可借书数目。

管理员实体集。其属性有：管理员账号，姓名，密码。

图书信息实体集。其属性有：图书编码，书名，类别，作者，出版社，状态，总数量。

**关系模型设计：**

我们的模型有四个表，borrow\_list:主键 俩

Book主键是bookcode，它是下面borrowlist bookcode列的外键。

Reader主键是card，它是borrowlist card列的外键

Librarian主键是account

**物理结构设计：**

读者表

（#借书卡号，姓名，性别，学院…）

管理员表

（#管理员账号，姓名，密码）

图书信息表

（#图书编码，书名，类别，作者…）

图书借阅表

（#图书编码，借阅者卡号）

**分工明细：**

左侧是我们小组初期页面设计的商量结果 右侧是我们的具体分工明细

**代码分析：**

首先介绍一下我们组的项目文件包，admin里面存放的是图书管理员相关代码，includes里面是我们的config文件，主要是数据库的链接代码，因为我们组员之间数据库的密码会有不同所以我们统一提取出来放在了一个文件夹里。reader里面放的是读者相关代码，resources是css和js文件。

登陆页面-index.php

(1)首先在登陆页面我们学习借鉴了一些css和js的文件，结合起来做了一个可以滑动的动态登陆界面。(2)首先展示css和js文件。Css文件里面就是一些关于样式和排版的定义，主要作用于html标签。这一块因为与课程内容关系不大所以就不再赘述。

然后是我们的index.php页面，首先是一些引用和设置，(3)我们在head标签里添加了一段js代码，是关于消除session登录信息的，(4)并在body标签中设置加载页面时自动运行这段代码，达到的效果就是当我们跳转到登录页面时就会自动要求用户重新登录(5)接下来是一段html表单，关于判断读者和图书管理员输入信息是否正确的代码我们组放在了其他页面，这样index就看起来比较简洁易懂。

后台管理系统页面-system\_admin.php

(6)然后是后台管理系统界面，因为后台管理系统的操作对象是我们小组成员，所以我们把它单独出来，没有与其他页面建立链接，这个系统的主要作用就是添加图书管理员，我们把这段的html代码与php代码分在两个文件中呈现，在form这里用action链接到add\_system\_admin.php页面，并且实现数据传输。(7)这个页面就是添加用户的PHP代码。使用内置数组$\_POST获取相关变量，然后再对数据库进行添加操作，添加成功后弹出提示框，点击确认后通过js的window对象的属性跳转回图书管理页面。

图书管理员页面-admin\admin\_home.php

（1）我们的导航栏用<nav>标签来实现，左侧的导航条分别为系统的名字“Localibrary”、“管理员主页”、“图书管理”、“用户管理”链接到各操作页面，目前所处页面的导航条用 class=‘active’标注加深，右侧的导航

条则为管理员名字和“退出”，管理员的名字通过session从登录页面接收

数据，点击退出则重新返回登录页面。（2）图书管理页面-

admin\_book.php 这个页面拥有添加图书以及删除图书两个功能，网页间

的数据传递使用的是表单提交数据的方法，通过‘添加按钮’把填入的ISBN

、图书名、作者等信息传到页面add\_book.php,（3）使用内置数组$\_POST

获取相关变量，然后再对数据库进行添加操作，添加成功后弹出提示框，

点击确认后通过js的window对象的属性跳转回图书管理页面。（4）删除

功能则通过点击‘删除’按钮实现，点击后触发myfunction函数,（5）以ajax方法把同一行的图书信息传入后端页面delete\_book.php，（6）后端页面接

收数据后首先在数据库判断图书状态是否为‘在库’，若已借出则提示无法

完成删除操作，待数据库中对应的图书信息删除后，（7）把更新后的列表重新

打印出来。

我们将与读者相关网页放入reader文件夹存储管理，这张图片显示的就是我们reader文件夹的目录

读者个人信息页面-reader\readerpage.php+getsite\_mysql1.php

（2）我们把判断读者卡号输入是否正确的代码放到了这一块，在用户输入卡号以后，我们用sql语句到数据库里进行查找，若未能查找到记录，则用alert弹出框提示读者输入不正确，然后跳转回Index页面重新登录。（3）若能够查找到记录，就用select语句取出改读者的个人信息显示在页面上（4）页面包含的第三部分就是读者借书记录部分的代码，这里除了从数据库取出数据显示到页面上外，还包含最后一列的还书按钮，点击后更改数据库信息，局部刷新页面，所以这里我们使用了AJAX，将按钮对应那一行的图书编码使用AJAX传入后端，更改图书状态和删除borrow\_list表记录，再将局部刷新后的页面传回前端显示。（5）测试网页时，当刷新页面或者从图书信息检索页面切换回来时，会出现错误，debug之后发现query从上一个网页的表单中获取信息，刷新后数据丢失，所以我们先将POST传输过来的数据存储在session里，再次刷新时将session里的数据赋给query，就可以解决问题。

图书信息查询借阅页面-search.php+explore.php+getsite\_mysql.php

Search 页面带有检索框，并且分页展示未搜索时所有图书的信息和借阅按钮，explore页面分页展示搜索结果，getsite\_mysql页面是上述两个页面运行AJAX时的后端，（这一段不需要图片就放第六张可以）（6）分页代码我们主要靠get方法实现， pagesize是我们自行设置的每页显示记录数，用get获取当前页码，并计算page，即当前页面显示的记录在数据库中的起始位置（7）这段代码展示的是我们在search和explore两个页面分别计算页面总数pagenum和显示出来的最小页码的begin，最大页码end，通过这段计算公式，我们实现的效果就是当页码超过3,并且pagenum超过5时，begin和end发生变化，并始终保证显示5个页码数和前进后退按钮,prev next变量就是这两个按钮要使用到的，他们会根据当前页面的页码，加一减一计算出上一页下一页的页码数。（8）为了在网页刷新或者换页时，（因为我们的换页也是全部刷新）所以想要保证搜索数据不丢失，依然显示的是该搜索条件下的结果，在search和explore在两个页面上稍有不同的就是这段代码所显示的，explore页面需要采用session记住搜索内容，（9）这段代码显示向后端请求借阅时两个页面的传入数据相同，需要进行的数据库操作也相同，都是修改状态并且向borrowlist中添加数据，（10）但最终局部刷新的显示结果有差别，所以在共用一个后端的情况下我们决定传入标记flag，就是这段代码中的temp3加以区分，如果temp3=1，相应select操作需要使结果符合记录在session里的搜索条件，而如果是未检索传过来的temp3=0，直接select book表里的所有数据即可（11）在这三个网页中还需要注意的问题就是如果图书已借出或是当前用户已借数量已经达到规定的3本时，需要限制读者的借阅操作，所以在三个网页中都添加了相应的判断代码，如果触碰限制条件就不显示操作按钮，在三个网页中都写判断代码是防止切换网页时出现条件不符合但又可以继续借阅的情况。

**项目特色：**

1. 登陆页面的亮点

前面的代码分析中我们已经讲解过关于登陆界面的创新，呈现了一种动态的网页特效。

1. 针对不同用户角色进行操作权限、管理范围和功能操作设置

我们这个项目的三个用户实体对于数据库内容都有不同的操作权限，也可以实现不同的功能，综合起来涵盖了对于数据库数据的增删改查，又在功能上富于变化。

1. 页面整体风格一致

我们所有代码运用了相同的css文件，保证每个网页之间切换时页面风格不变，整体比较美观简洁。

1. 数据库内容分类

在数据库中我们为不同的图书添加了类别，这样既便于管理员管理，也便于读者的检索和借阅。

**所遇困难及解决方法：**

1. **出现在图书查询页面已达到借书上限但刷新之后又可以借阅的情况；**

经调试后发现在3个网页同时添加判断代码就可以避免这个情况

1. **添加管理员时表单和添加代码放在一个页面时会向数据库中添加一个空白值**

我们尝试把表单和添加代码放在不同文件，发现这样就不会向数据库中添加空白值。

1. **实现搜索功能后，在点击借阅按钮局部刷新后只显示改变状态的记录无法显示此搜索条件下其他未改变状态的记录；**

这个问题我们的解决方法是用session记录下检索条件，在后端操作数据库时添加where判断，select出所有符合条件的记录传回前端

1. **实现分页功能后，检索状态下改变页码或者点击借阅都会显示未检索状态的记录；**

这个问题的核心就是如何区分检索和未检索两个状态，虽然使用了session记录检索条件，但是从检索状态回到未检索状态后session依然存在，所以在一个网页里我们没有想到很好的区分方式，于是我们将网页拆分，两个状态两个网页，分离之后就不会出现问题，实现了我们的预期效果。

1. **退出登录功能的实现。**

我们将session清空销毁函数放在登录页面，并设置加载网页时自动执行，再在其他任何页面设置点击退出跳转到登录页面，这样在跳转回登录页面的同时也清楚了记录在session里的登录信息。

**未实现功能：**

在数据库物理设计的几个步骤中，我们完善了图书管理系统数据库，通过设置三个用户完成了安全性设计，但是忽略了**数据库维护**的问题而没有编写定时备份等功能，这就使得我们难以防范对数据的威胁，可能导致对访问数据的系统的毁坏，比如管理系统的设计漏洞和人为的操作失误。

第二点未实现的是**罚金功能**，还有与其相对应的对借还日期的设定。我们在需求分析中有考虑到这个比较有代表性的功能，因为这个功能能够维持图书管理系统的稳定秩序，对不遵守规则的用户可以起到限制作用，但迫于能力和精力，我们没能更好地完善该系统，取而代之的是限制读者的最大借书数目，小程度地限定读者权限。

第三点的**代码封装性**问题，是我们在分工完成代码的编写后，整合在一起时发现需要花费大量的精力来统一代码结构，同时与其他页面建立连接；并且我们在大部分页面中都需要连接数据库，每次更改数据库地址和密码的时间成本较大，因此决定整合成一个config.php来直接引用，但这并没有改变该系统代码和数据冗余的现状，因此我们需要增加代码的可重用性和可维护性，提高系统的管理效率。

以上就是我们小组汇报的全部内容，感谢老师本学期的教学，期待老师的批评和指正。