1. **需求分析**
2. 用户可以注册、登录。

普通用户的功能：

(1)普通用户不登录也可以浏览新闻中心页面、浏览图书页面。

(2)用户可以书名、书号、作者等条件查询图书的信息。可以根据书籍的类别查询相关的书籍。

(3)用户登录后可以下载到某本书籍相应的电子资源。

(4)用户可以浏览、查询数字书籍信息。

管理员的功能：

(1)管理员负责撰写、管理新闻，能够对新闻进行增删改、查询。

(2)管理书籍信息，能够对书籍信息进行增删改、查询。

(3)管理电子资源，某些书籍配备了相应的PPT、PDF等文件，管理员负责上传这些文件。

首页

图书中心

下载中心

新闻中心

数字图书中心

1. **总体设计**

运用Asp.net+Div+Css+ js（jquery）技术对系统网页的前台和后台进行设计，

用户体验层(前台)：

1.导航栏等共有页面元素作为整个前台系统网页布局的母模版，包括（新闻中心、下载中心、图书中心、数字图书中心、搜索、登陆\注册）

2.首页为3个主要板块的内容简介和热点。新闻板块根据时间更新业内业外的动态新闻。图书板块查看各类书籍的详情简要介绍。下载板块进入前需要登陆，提供书籍电子资源的下载。数字图书板块提供书籍内容的在线阅览。

管理员登陆管理(后台)：

1.管理员对图书和新闻进行图书信息和新闻的增、删、查、改操作。

2.管理员可以新增、删除普通用户。

3.管理员可以管理电子资源的删除和上传。

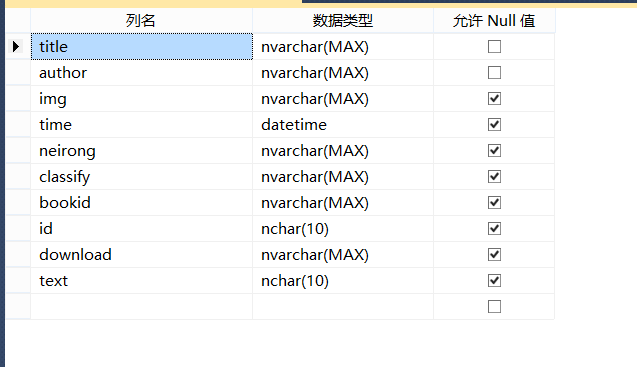
系统设计流程思想：

数据库设计：使用SQL Server 2014 Management Studio进行数据的存放。

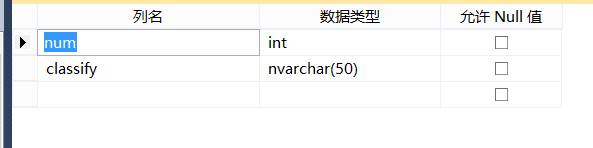
数据库：chubanshe(系统书籍、新闻信息的保存)+wgd(用户、管理员数据的保存)

chubanshe数据库：

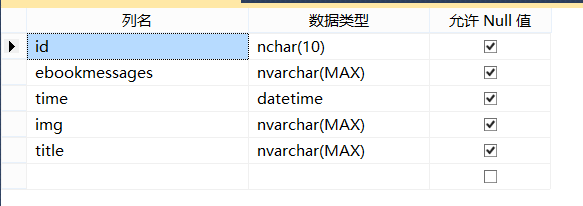
书籍信息存放表：



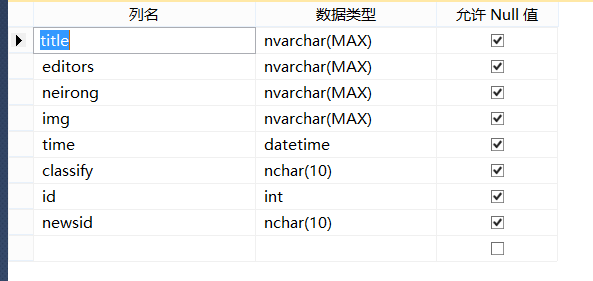
书籍类别表：



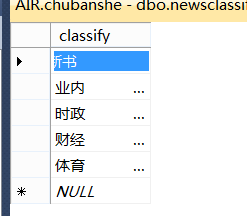
电子书籍信息存放表：



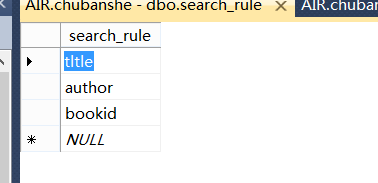
新闻信息存放表：



新闻类别存放表：

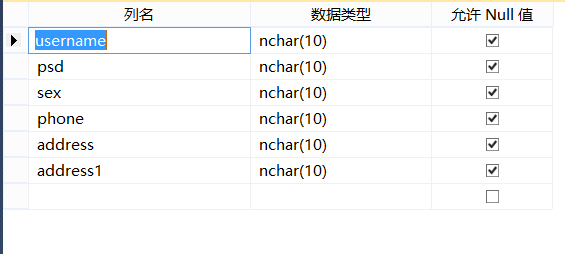


搜索条件表：

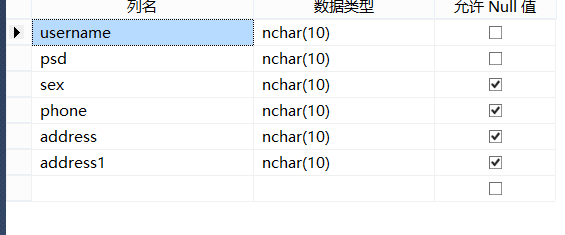


Wgd()

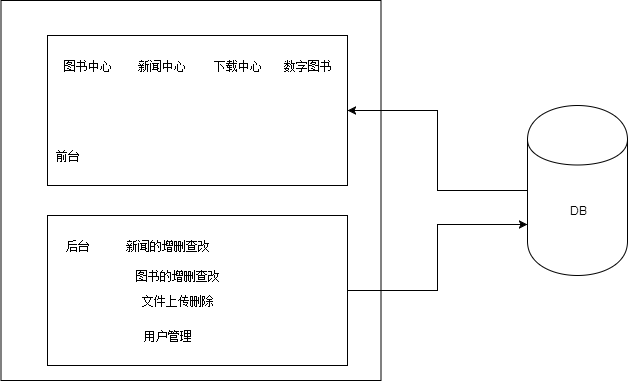
普通用户表



管理员



流程图



1. **详细设计**

登陆：从登陆页面获取用户的登陆信息，进而进行数据库的密钥匹配，通过后保存session进行网页的重定向

string username = TextBox1.Text;

string psd = TextBox2.Text;//

string name = "";

string mima = "";

string sql = "select \* from yonghu where username='" + username + "' and psd='" + psd + "'";

sqlconn.Open();

SqlCommand cmd = new SqlCommand(sql, sqlconn);

SqlDataReader datar = cmd.ExecuteReader();

//if (datar.HasRows)

//{

// Server.Transfer("shangpinggridview.aspx");

//}

while (datar.Read())

{

name = datar["username"].ToString();

mima = datar["psd"].ToString();

}

sqlconn.Close();

if (name != "" && mima != "")

{

Session["user"] = name;

Session["password"] = mima;

Server.Transfer("index.aspx");

}

else

{

Response.Write("<script>alert('用户名或者密码错误')</script>");

}

注册：将用户填写的合法信息保存在数据库中，填写不合法返回注册失败。

int i = -1;

string password = txtpassword.Text; //密码

string sex = RadioButtonList1.SelectedValue; //性别

//string leixing = RadioButtonList2.SelectedValue;

//string birth = txtBirth.Text; //出生日期

//string eduation = dropEducation.SelectedValue; //学历

string phone = txtdianhua.Text; //电话

string address = txtdizhi.Text; //地址

//string zip = txtZip.Text; //邮政编码

//string email = txtEmail.Text; //电子邮件

//string regdate = DateTime.Now.ToShortDateString(); //注册时间

string address1 = "会员";

// string insertstr = "insert into 用户信息$(用户名,密码,性别,联系电话,地址,身份 ) values('" + txtusername.Text + "','" + txtpassword.Text + "','" + RadioButtonList1.SelectedItem.Text + "','" + txtdianhua.Text + "','" + txtdizhi.Text + "','" + RadioButtonList2.SelectedItem.Text + "')";

string insertstr = "insert into yonghu(username,psd,sex,phone,address,address1 ) values('" + txtusername.Text + "','" + txtpassword.Text + "','" + RadioButtonList1.SelectedItem.Text + "','" + txtdianhua.Text + "','" + txtdizhi.Text + "','" + address1 + "')";

SqlCommand mycmd = new SqlCommand(insertstr, sqlconn);

sqlconn.Open();

try

{

i = mycmd.ExecuteNonQuery();

string selectstr = "select username,psd,sex,phone,address,assress1 from yonghu";

SqlCommand mycmd2 = new SqlCommand(selectstr, sqlconn);

SqlDataReader mydr = mycmd2.ExecuteReader();

}

图书的加载和显示

利用listview空间绑定图书信息数据源，对需要的图书信息进行显示

<asp:ListView ID="ListView1" runat="server" ItemPlaceholderID="itemhorder"

DataKeyNames="bookid" >

<LayoutTemplate>

<div id="itemhorder" runat="server"></div>

</LayoutTemplate>

<ItemTemplate>

<li class="left">

<a href="renwen.aspx?id=<%#Eval("id")%>">

<div class="book\_size left">

<div class="bookimg"> <img alt="" src='<%#Eval("img","img/{0}") %>' /></div>

<div class="left">

<p><%#Eval("title") %></p>

</div>

</div>

</a>

</li>

</ItemTemplate>

</asp:ListView>

图书详情的加载显示

<a href="renwen.aspx?id=<%#Eval("id")%>">

利用a标签对页面数据流进行值的匹配和显示

<asp:ListView ID="ListView1" runat="server" ItemPlaceholderID="itemhorder"

DataSourceID="SqlDataSource1">

<LayoutTemplate>

<div id="itemhorder" runat="server"></div>

</LayoutTemplate>

<ItemTemplate>

<div class="box">

<div class="a">

<p class="b"><%#Eval("title") %><span>&nbsp;(<%#Eval("author") %>)</span></p>

<small><%#Eval("time") %></small>

</div>

<hr />

<div class="c" style="">

<img style="width:300px;" alt="" src='<%#Eval("img","img/{0}") %>' />

<p><%#Eval("neirong") %></p>

</div>

</div>

</ItemTemplate>

</asp:ListView>

图书搜索：

根据搜索条件进行搜索（标题、作者、书籍号）

else if (!string.IsNullOrEmpty(Request.QueryString["search\_rule"]))

{

//实例化SqlConnection对象

SqlConnection sqlCon = new SqlConnection();

//实例化SqlConnection对象连接数据库的字符串

sqlCon.ConnectionString = "server=AIR;uid=wtaure;pwd=123456;database=chubanshe";

//定义SQL语句

string SqlStr;

switch (Request.QueryString["search\_rule"])

{

case "bookid":

SqlStr = "select \* from book\_messages where bookid = '" + Request.QueryString["textbox"] + "'";

break;

case "title":

SqlStr = "select \* from book\_messages where title like '%" + Request.QueryString["textbox"] + "%'";

break;

case "author":

SqlStr = "select \* from book\_messages where author like '%" + Request.QueryString["textbox"] + "%'";

break;

default:

SqlStr = "select \* from book\_messages ";

break;

}

将查询到的数据绑定到listview

//实例化SqlDataAdapter对象

SqlDataAdapter da = new SqlDataAdapter(SqlStr, sqlCon);

//实例化数据集DataSet

DataSet ds = new DataSet();

da.Fill(ds, "book\_messages");

//绑定GridView控件

// GridView1.DataSource = ds;//设置数据源，用于填充控件中的项的值列表

//GridView1.DataBind();//将控件及其所有子控件绑定到指定的数据源

int i = ds.Tables[0].Rows.Count;

ListView1.DataSourceID = null;

ListView1.DataSource = ds;

ListView1.DataBind();

书籍信息添加：

将书籍填写的信息从后台插入到数据库中

string shujihao = TextBox1.Text;

string biaoti = TextBox2.Text;

string neirong = TextBox3.Text;

//int nianling = Convert.ToInt16(TextBox3.Text);

string author = TextBox4.Text;

string classify = DropDownList1.Text;

string id = TextBox5.Text;

string img = TextBox6.Text;

DateTime d = DateTime.Now;

string sqltext = "insert into book\_messages(bookid,title,neirong,classify,id,author,time,img) values ('" + shujihao + "','" + biaoti + "','" + neirong + "','" + classify + "','" + id + "','" + author + "','" + d+ "','" + img + "')";

SqlCommand cmd = new SqlCommand(sqltext, sqlconn);

sqlconn.Open();

int countrecord = cmd.ExecuteNonQuery();

string sqltext2 = "select \* from book\_messages";

SqlCommand cmd2 = new SqlCommand(sqltext2, sqlconn);

SqlDataReader sdr = cmd2.ExecuteReader();

GridView2.DataSource = sdr;

GridView2.DataBind();

sqlconn.Close();

if (FileUpload1.HasFile)//文件上传

{

string tar\_file = MapPath("~/img/") + FileUpload1.FileName;

FileUpload1.SaveAs(tar\_file);

Response.Redirect(Request.RawUrl);

}

书籍删除

根据设置的主键查询图书信息

SqlConnection myConn = GetCon();

//定义SQL语句

string SqlStr = "select \* from book\_messages";

//实例化SqlDataAdapter对象

SqlDataAdapter da = new SqlDataAdapter(SqlStr, myConn);

//实例化数据集DataSet

DataSet ds = new DataSet();

da.Fill(ds, "book\_messages");

//绑定DataList控件

GridView1.DataSource = ds;//设置数据源，用于填充控件中的项的值列表

GridView1.DataKeyNames = new string[] { "bookid" };

GridView1.DataBind();//将控件及其所有子控件绑定到指定的数据源

删除信息

//获取被删除行的键值

string cname=(GridView1.DataKeys[e.RowIndex].Value).ToString();

//定义SQL语句

string SqlStr = " delete from book\_messages where bookid='" + cname+ "'";

SqlConnection myConn = GetCon();

SqlCommand sqlcom = new SqlCommand(SqlStr, myConn);

myConn.Open();

sqlcom.ExecuteNonQuery();

ClientScript.RegisterClientScriptBlock(this.GetType(), "info", "alert('记录已删除');", true);

//重新绑定数据

myConn.Close();

GridViewBind();

书籍更新：

//取得编辑行的关键字段的值

string stuID = GridView1.DataKeys[e.RowIndex].Value.ToString();

//取得文本框中输入的内容

string stuName = ((TextBox)(GridView1.Rows[e.RowIndex].Cells[1].Controls[0])).Text.ToString();

string stuSex = ((TextBox)(GridView1.Rows[e.RowIndex].Cells[2].Controls[0])).Text.ToString();

string stringstuAge = ((TextBox)(GridView1.Rows[e.RowIndex].Cells[3].Controls[0])).Text.ToString();

//int stuAge = Convert.ToInt16(stringstuAge);

string studept = ((TextBox)(GridView1.Rows[e.RowIndex].Cells[4].Controls[0])).Text.ToString();

string sqlStr = "update book\_messages set title='" + stuName + "',author='" + stuSex + "',classify='" + stringstuAge + "',neirong='" + studept + "' where bookid='" + stuID + "'";

SqlConnection myConn = GetCon();

myConn.Open();

SqlCommand myCmd = new SqlCommand(sqlStr, myConn);

myCmd.ExecuteNonQuery();

myCmd.Dispose();

myConn.Close();

GridView1.EditIndex = -1;

GridViewBind();

取消编辑按钮

//设置GridView控件的编辑项的索引为－1，即取消编辑

GridView1.EditIndex = -1;

//数据绑定

GridViewBind();

更新编辑按钮

//设置GridView控件的编辑项的索引为选择的当前索引

GridView1.EditIndex = e.NewEditIndex;

//数据绑定

GridViewBind();

书籍的查询：

//实例化SqlConnection对象

SqlConnection sqlCon = new SqlConnection();

//实例化SqlConnection对象连接数据库的字符串

sqlCon.ConnectionString = "server=AIR;uid=wtaure;pwd=123456;database=chubanshe";

//定义SQL语句

string SqlStr = "select \* from book\_messages where bookid like '%"+TextBox1.Text+"%'";

//实例化SqlDataAdapter对象

SqlDataAdapter da = new SqlDataAdapter(SqlStr, sqlCon);

//实例化数据集DataSet

DataSet ds = new DataSet();

da.Fill(ds, "book\_messages");

//绑定GridView控件

GridView1.DataSource = ds;//设置数据源，用于填充控件中的项的值列表

GridView1.DataBind();//将控件及其所有子控件绑定到指定的数据源

int i= GridView1.Rows.Count;

if (i == 0) { ClientScript.RegisterStartupScript(typeof(string), "print", "<script>alert('没有满足条件的记录，请确定书籍号是否正确！')</script>"); }

// Response.Write(i.ToString ());

TextBox1.Text = "";

TextBox1.Focus();

1. **程序运行结果测试与分析**

管理员(后台)设计

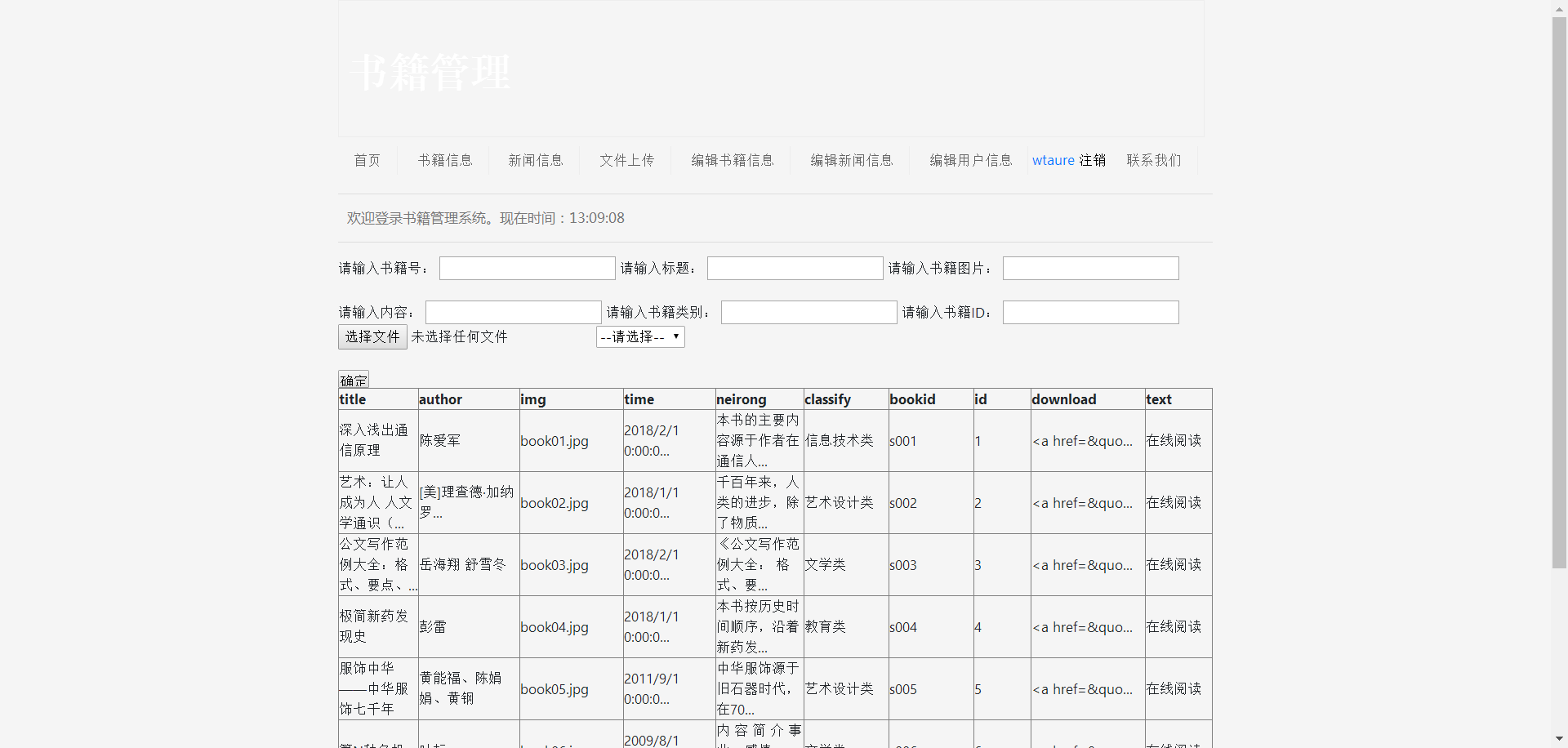
静态网络的设计

图书信息：

查询



新增



更新\修改啊啊啊啊啊啊啊啊啊啊啊啊啊啊啊啊啊啊啊啊啊啊啊啊啊啊啊啊啊啊啊啊啊啊啊



删除啊啊啊啊啊啊啊啊啊啊啊啊啊啊啊啊啊啊啊啊啊啊啊啊啊啊啊啊啊啊啊啊啊啊啊



新闻信息：

查询啊啊啊啊啊啊啊啊啊啊啊啊啊啊啊啊啊啊啊啊啊啊啊啊啊啊啊啊啊啊啊啊啊啊啊



新增啊啊啊啊啊啊啊啊啊啊啊啊啊啊啊啊啊啊啊啊啊啊啊啊啊啊啊啊啊啊啊啊啊啊啊



修改啊啊啊啊啊啊啊啊啊啊啊啊啊啊啊啊啊啊啊啊啊啊啊啊啊啊啊啊啊啊啊啊啊啊啊



删除啊啊啊啊啊啊啊啊啊啊啊啊啊啊啊啊啊啊啊啊啊啊啊啊啊啊啊啊啊啊啊啊啊啊啊



用户删除啊啊啊啊啊啊啊啊啊啊啊啊啊啊啊啊啊啊啊啊啊啊啊啊啊啊啊啊啊啊啊啊啊啊啊



用户体验层(前台)设计

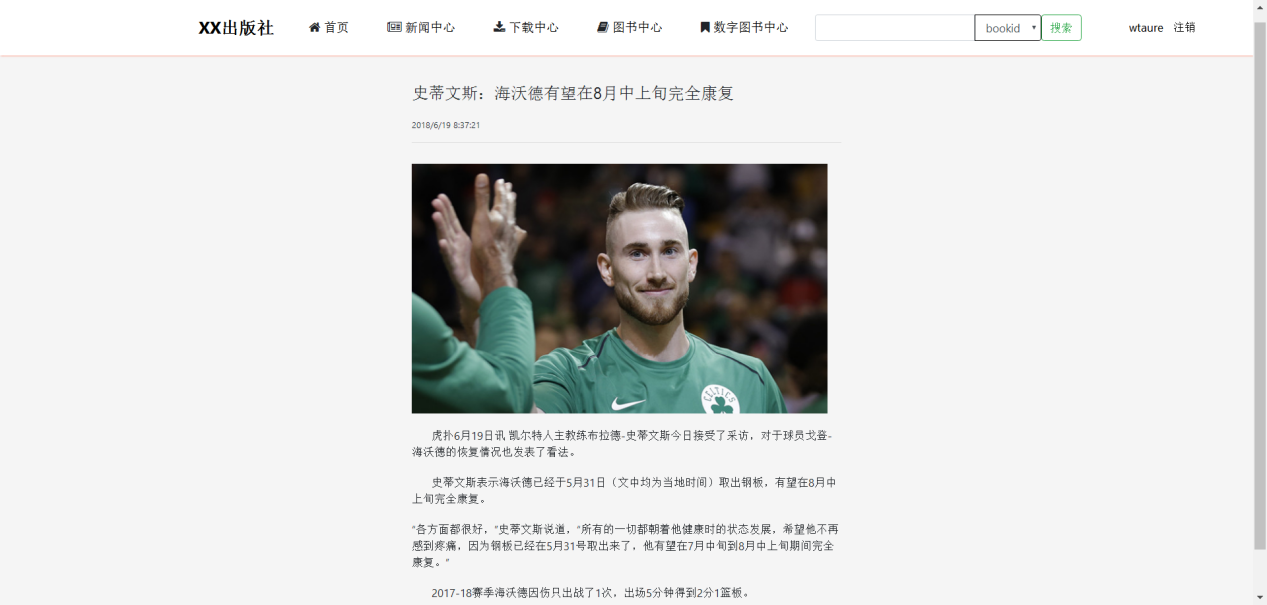
首页啊啊啊啊啊啊啊啊啊啊啊啊啊啊啊啊啊啊啊啊啊啊啊啊啊啊啊啊啊啊啊啊啊啊啊



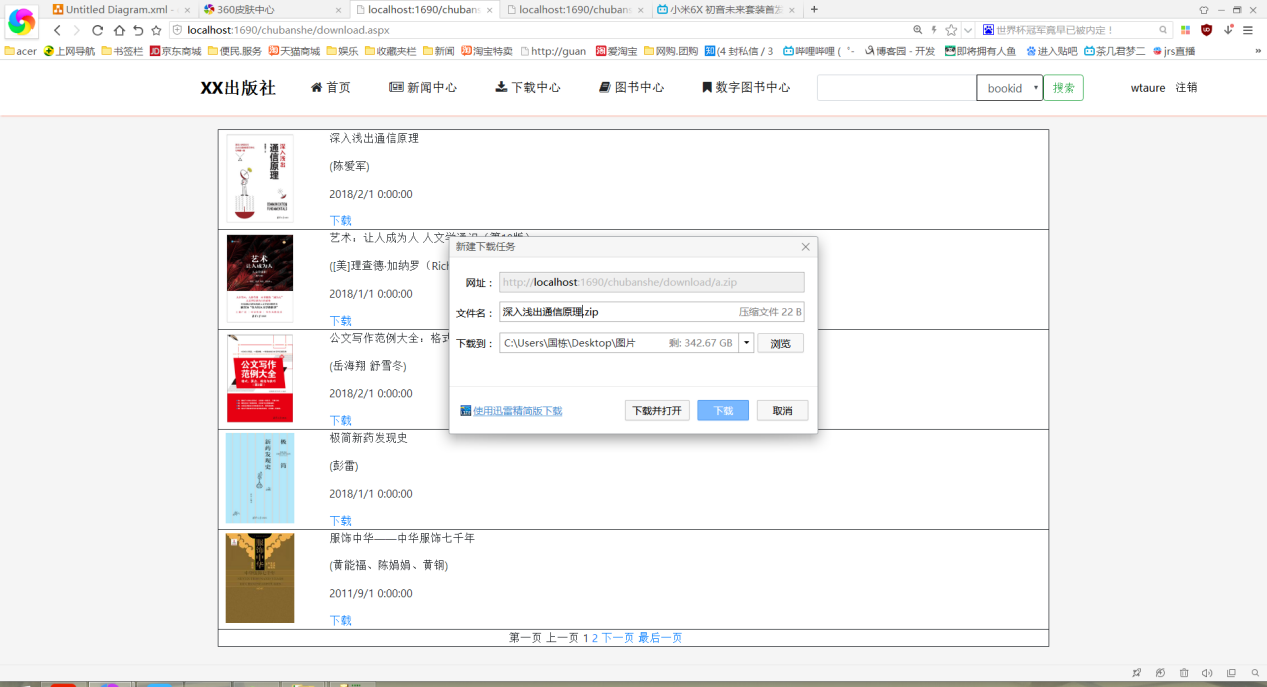
新闻中心啊啊啊啊啊啊啊啊啊啊啊啊啊啊啊啊啊啊啊啊啊啊啊啊啊啊啊啊啊啊啊啊啊啊啊



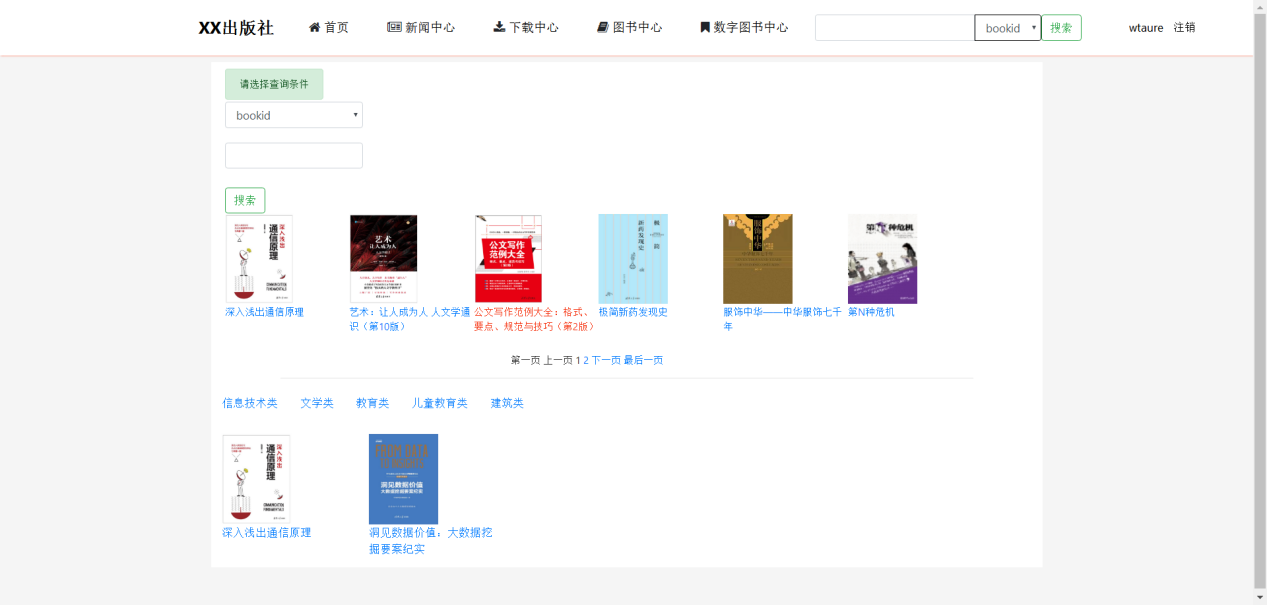
新闻详情啊啊啊啊啊啊啊啊啊啊啊啊啊啊啊啊啊啊啊啊啊啊啊啊啊啊啊啊啊啊啊啊啊啊啊



下载中心啊啊啊啊啊啊啊啊啊啊啊啊啊啊啊啊啊啊啊啊啊啊啊啊啊啊啊啊啊啊啊啊啊啊啊



图书中心



数字图书啊啊啊啊啊啊啊啊啊啊啊啊啊啊啊啊啊啊啊啊啊啊啊啊啊啊啊啊啊啊啊啊啊啊啊



图书详情啊啊啊啊啊啊啊啊啊啊啊啊啊啊啊啊啊啊啊啊啊啊啊啊啊啊啊啊啊啊啊啊啊啊啊



数字图书在线阅读啊啊啊啊啊啊啊啊啊啊啊啊啊啊啊啊啊啊啊啊啊啊啊啊啊啊啊啊啊啊啊

