2016 －2017 学年第 二 学期



校园新闻APP开发

程序清单



实验课程名称 软件工程导论

专 业 班 级 软件工程1501

学 号 31501310 31501391 31501404

学 生 姓 名 奕吉 陈妍蓝 靳泽旭

实验指导老师 杨枨

|  |  |  |
| --- | --- | --- |
| 文件状态： | 文件标识： | SE-2017-G17 |
| [ √ ]:草稿 | 当前版本： | 1.0 |
| [ ]:正式发布 | 作 者： | 陈妍蓝、奕吉、靳泽旭 |
| [ ]:正在修改 | 完成日期： | 2017年6月16日 |

# 1 引言

## 1.1 标题

1.1.1 管理员子系统

1.1.2 用户子系统

界面

# 2 管理员子系统

## 2.1 主界面代码

## 2.2 登录

### 2.2.1 功能说明

本模块的功能为管理员子系的登录，需要输入管理员账号密码

### 2.2.2 设计说明

1、 读取用户输入信息后验证信息格式

2、 验证用户输入账号是否存在

3、 验证用户输入密码是否正确

4、将系统当前用户置为成功登陆用户

### 2.2.3代码

## 2.3 管理员新建文章

### 2.3.1 功能说明

本模块主要功能是新建文章，输入新闻的标题和链接。

### 2.3.2 设计说明

1、读取用户输入的标题和链接

2、将新闻标题的链接读入数据库

### 2.3.3 代码

## 2.4管理员修改文章

### 2.4.1 功能说明

本模块主要功能是修改文章，修改新闻的标题和链接。

### 2.4.2 设计说明

1、读取用户输入的标题和链接

2、将新新闻标题的链接读入数据库代替原有的内容

### 2.4.3 代码

## 2.5 管理员用户设置

### 2.5.1 功能说明

本模块主要功能是管理员可以修改自己的密码

### 2.5.2 设计说明

1、读取管理员输入的用户名，原密码和新密码

2、将原密码和数据库中相对应的内容进行验证，验证成功即可将新密码代替原密码

### 2.5.3 源代码

# 3 用户子系统

本组的用户子系统遇到了技术上的难题，失败了，但也做了相应的界面，只是功能上无法实现预期的目标。以下是我们用户子系统界面的部分代码和截图。

<?xml version="1.0" encoding="utf-8"?>

<FrameLayout xmlns:android="http://schemas.android.com/apk/res/android"

xmlns:fab="http://schemas.android.com/apk/res-auto"

android:layout\_width="match\_parent"

android:layout\_height="match\_parent">

<ListView

android:id="@android:id/list"

android:layout\_width="match\_parent"

android:layout\_height="match\_parent"

android:cacheColorHint="@android:color/transparent"

android:headerDividersEnabled="false"/>

<com.melnykov.fab.FloatingActionButton

android:id="@+id/fab"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_gravity="bottom|right"

android:layout\_margin="16dp"

android:src="@drawable/ic\_add\_white\_24dp"

fab:fab\_colorNormal="@color/accent"

fab:fab\_colorPressed="@color/accent\_pressed"

fab:fab\_colorRipple="@color/ripple"/>

</FrameLayout> xmlns:fab="http://schemas.android.com/apk/res-auto"

android:layout\_width="match\_parent"

android:layout\_height="match\_parent">

<ListView

android:id="@android:id/list"

android:layout\_width="match\_parent"

android:layout\_height="match\_parent"

android:cacheColorHint="@android:color/transparent"

android:headerDividersEnabled="false"/>

<com.melnykov.fab.FloatingActionButton

android:id="@+id/fab"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_gravity="bottom|right"

android:layout\_margin="16dp"

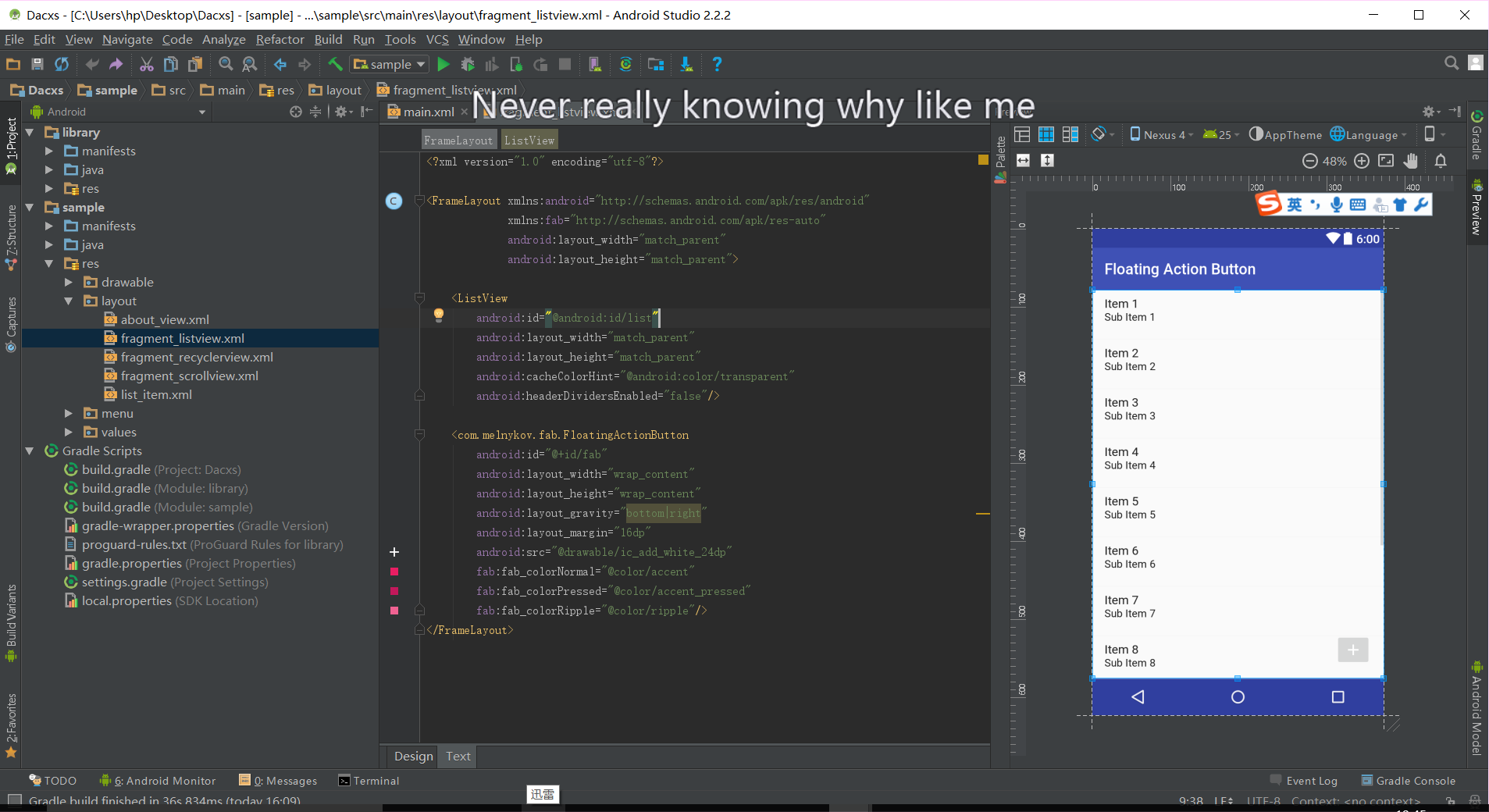
android:src="@drawable/ic\_add\_white\_24dp"

fab:fab\_colorNormal="@color/accent"

fab:fab\_colorPressed="@color/accent\_pressed"

fab:fab\_colorRipple="@color/ripple"/>

</FrameLayout>  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:cacheColorHint="@android:color/transparent"  
 android:headerDividersEnabled="false"/>  
  
 <com.melnykov.fab.FloatingActionButton  
 android:id="@+id/fab"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_gravity="bottom|right"  
 android:layout\_margin="16dp"  
 android:src="@drawable/ic\_add\_white\_24dp"  
 fab:fab\_colorNormal="@color/accent"  
 fab:fab\_colorPressed="@color/accent\_pressed"  
 fab:fab\_colorRipple="@color/ripple"/>  
</FrameLayout>



<RelativeLayout

xmlns:android="http://schemas.android.com/apk/res/android"

android:layout\_width="match\_parent"

android:layout\_height="match\_parent">

<TextView

android:id="@+id/lyla\_text\_view"

android:text="最新资讯"

android:textSize="30sp"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_marginTop="16dp"

android:background="@android:color/holo\_blue\_light"

android:textColor="@android:color/holo\_red\_light"/>

<TextView

android:id="@+id/me\_text\_view"

android:text="二课活动"

android:textSize="30sp"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_alignParentTop="true"

android:layout\_centerHorizontal="true"

android:layout\_marginTop="16dp"

android:background="@android:color/holo\_blue\_light"

android:textColor="@android:color/holo\_red\_light"/>

<TextView

android:id="@+id/me\_text\_view"

android:text="竞赛考试"

android:textSize="30sp"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_alignParentTop="true"

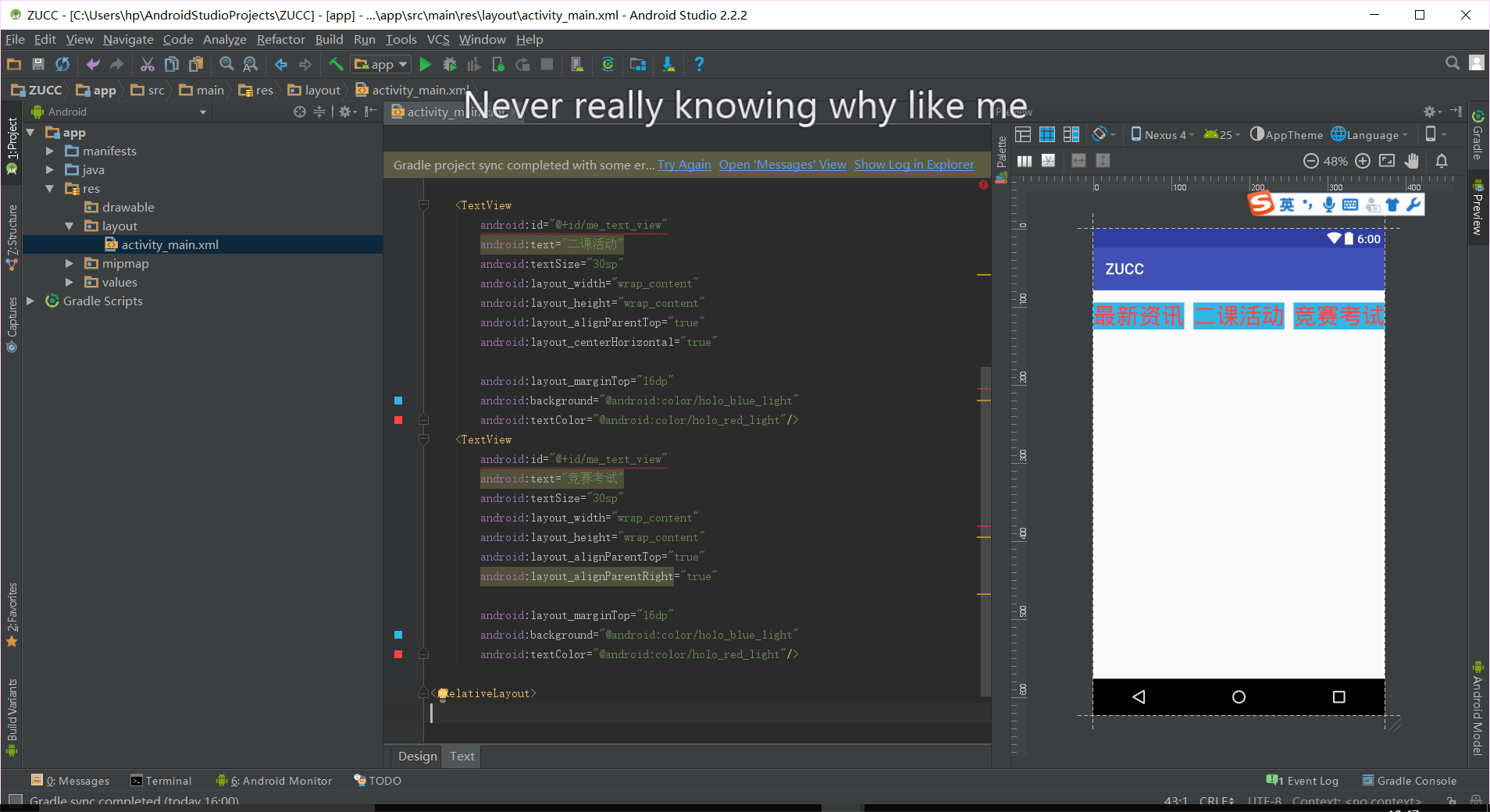
android:layout\_alignParentRight="true"

android:layout\_marginTop="16dp"

android:background="@android:color/holo\_blue\_light"

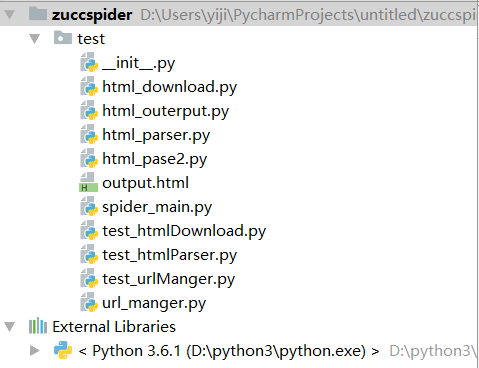
android:textColor="@android:color/holo\_red\_light"/>

</RelativeLayout>



# 4 管理员后台

**爬虫模块结构目录：**



**spider\_main.py**

**from** test **import** html\_download, html\_parser, html\_outerput, html\_pase2  
**from** test **import** url\_manger  
**class** SpiderMain(object):  
 **def** \_\_init\_\_(self):  
 self.urls = url\_manger.UrlManger()  
  
 self.downloader = html\_download.HtmlDownload()  
  
 self.parser = html\_parser.HtmlParser()  
  
 self.outputer = html\_outerput.HtmlOutputer()  
  
 self.paser\_data = html\_pase2.HtmlPaserData()  
  
 **def** craw(self, root\_url):  
 self.urls.add\_new\_url(root\_url)  
  
 new\_url = self.urls.get\_new\_url()  
  
 html\_cont = self.downloader.download(new\_url)  
  
 new\_urls = self.parser.parser(new\_url, html\_cont)  
  
 self.urls.add\_new\_urls(new\_urls)  
  
 **while** self.urls.has\_new\_url():  
 new\_url2 = self.urls.get\_new\_url()  
  
 html\_cont2 = self.downloader.download(new\_url2)  
  
 new\_data = self.paser\_data.newsdata(new\_url2, html\_cont2)  
  
 self.outputer.collect\_data(new\_data)  
 self.outputer.output\_html()  
  
  
**if** \_\_name\_\_ == **"\_\_main\_\_"**:  
 root\_url = **"http://jsxy.zucc.edu.cn/index.html"** obj\_spider = SpiderMain()  
 obj\_spider.craw(root\_url)

**url\_manger.py**

**class** UrlManger(object):  
 **def** \_\_init\_\_(self):  
 self.new\_urls = set()  
 self.old\_urls = set()  
  
 **def** add\_new\_url(self, url):  
 **if** url **is None**:  
 **return  
 if** url **not in** self.new\_urls **and** url **not in** self.old\_urls:  
 self.new\_urls.add(url)  
  
 **def** add\_new\_urls(self, urls):  
 **if** urls **is None or** len(urls) == 0:  
 **return  
 for** url **in** urls:  
 self.add\_new\_url(url)  
  
 **def** has\_new\_url(self):  
 **return** len(self.new\_urls) != 0  
  
 **def** get\_new\_url(self):  
 new\_url = self.new\_urls.pop()  
 self.old\_urls.add(new\_url)  
 **return** new\_url

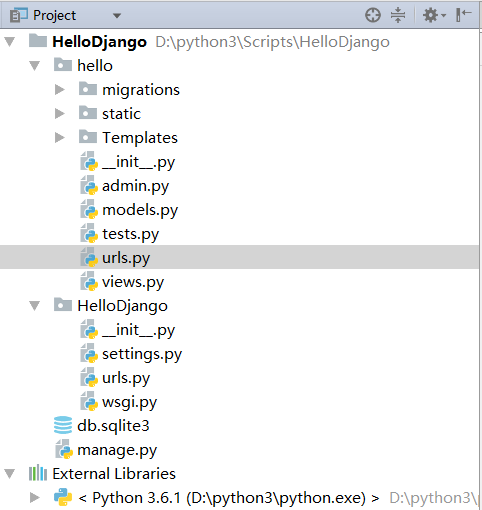
**html\_download.py**

**class** UrlManger(object):  
 **def** \_\_init\_\_(self):  
 self.new\_urls = set()  
 self.old\_urls = set()  
  
 **def** add\_new\_url(self, url):  
 **if** url **is None**:  
 **return  
 if** url **not in** self.new\_urls **and** url **not in** self.old\_urls:  
 self.new\_urls.add(url)  
  
 **def** add\_new\_urls(self, urls):  
 **if** urls **is None or** len(urls) == 0:  
 **return  
 for** url **in** urls:  
 self.add\_new\_url(url)  
  
 **def** has\_new\_url(self):  
 **return** len(self.new\_urls) != 0  
  
 **def** get\_new\_url(self):  
 new\_url = self.new\_urls.pop()  
 self.old\_urls.add(new\_url)  
 **return** new\_url

**html\_parser.py**

**from** bs4 **import** BeautifulSoup  
**import** re  
**import** urllib  
  
  
**class** HtmlParser(object):  
 **def** \_get\_new\_urls(self, page\_url, soup):  
 new\_urls = set() *#--------1  
 # href="/art/2017/5/10/art\_910\_139198.html"* links = soup.find\_all(**'a'**, href=re.compile(**r"/art/\d+\/\d+\/\d+\/art\_\d+\\_\d+\.html"**))*#----------2* **for** link **in** links:*#-----------------------3* new\_url = link[**'href'**]  
 new\_full\_url = urllib.parse.urljoin(page\_url, new\_url)  
 new\_urls.add(new\_full\_url)*#--------------------4  
 # print(new\_full\_url)* **return** new\_urls*#---------------------------5* **def** parser(self, page\_url, html\_cont):  
 **if** page\_url **is None or** html\_cont **is None**:  
 **return None** soup = BeautifulSoup(html\_cont, **'lxml'**)  
 new\_urls = self.\_get\_new\_urls(page\_url, soup)  
  
 **return** new\_urls

**web服务应用模块：总的结构目录**

****

**URLS.py**

**from** django.conf.urls **import** include, url  
**from** . **import** views  
  
urlpatterns = [  
  
 url(**r'^main/$'**,views.hello),  
 url(**r'^index/$'**, views.index),  
 url(**r'^home/$'**,views.home, name=**'home'**),  
 url(**r'^about/$'**,views.about,name=**'about'**),  
 url(**r'^setting/$'**,views.setting,name=**'setting'**),  
 url(**r'^manage/$'**,views.manage,name=**'manage'**),  
 url(**r'^message/$'**,views.message,name=**'message'**),  
 url(**r'^news/(?P<article\_id>[0-9]+)$'**,views.article\_page,name=**'article\_page'**),  
 url(**r'^article\_moife/(?P<article\_id>[0-9]+)$'**,views.article\_modife,name=**'article\_moife'**),  
 url(**r'^artical\_new/$'**,views.manage\_new,name=**'artical\_new'**),  
 url(**r'^article\_modife\_action/$'**, views.article\_modife\_action, name=**'artical\_modife\_action'**)  
]

**views.py**

**from** django.shortcuts **import** render  
**from** django.shortcuts **import** render,render\_to\_response  
**from** django.http **import** HttpResponse,HttpResponseRedirect  
**from** django.template **import** RequestContext  
**from** django **import** forms  
**from**.models **import** User  
**from**.models **import** Artical  
  
**class** UserForm(forms.Form):  
 username = forms.CharField(label=**'用户名'**,max\_length=100)  
 password = forms.CharField(label=**'密码'**,widget=forms.PasswordInput())  
**class** ChangeForm(forms.Form):  
 username = forms.CharField(label=**'用户名'**)  
 old\_password = forms.CharField(label=**'原密码'**,widget=forms.PasswordInput())  
 new\_password = forms.CharField(label=**'新密码'**,widget=forms.PasswordInput())  
  
  
  
*#登陆***def** hello(request):  
 **if** request.method == **'POST'**:  
 uf = UserForm(request.POST)  
 **if** uf.is\_valid():  
 *# 获取表单用户密码* username = uf.cleaned\_data[**'username'**]  
 password = uf.cleaned\_data[**'password'**]  
 *# 获取的表单数据与数据库进行比较* user = User.objects.filter(username=username)  
 **if** user:  
 *# 比较成功，跳转index* passwd = User.objects.filter(username=username, password=password)  
 **if** passwd:  
 *# 将username写入浏览器cookie,失效时间为3600* response = HttpResponseRedirect(**'/login/index/'**)  
 response.set\_cookie(**'username'**,username,3600)  
 **return** response  
 **else**:  
 info = **'请检查密码是否正确!'  
 return** HttpResponse(info)  
  
 **elif** len(user) == 0:  
 info = **'请检查用户名是否正确!'  
 return** HttpResponse(info)  
 **else**:  
 uf = UserForm()  
 **return** render\_to\_response(**'main.html'**,{**'uf'**:uf},context\_instance=RequestContext(request))  
  
  
  
  
  
*#登陆成功***def** index(request):  
 username = request.COOKIES.get(**'username'**,**''**)  
 **return** render\_to\_response(**'index.html'** ,{**'username'**:username})  
  
**def** home(request):  
 articles=Artical.objects.all()  
 **return** render(request,**'home.html'**,{**'articles'**:articles})  
  
**def** about(request):  
 **return** render(request,**'about.html'**)  
  
  
  
  
  
  
  
**def** setting(request):  
 **if** request.method == **'POST'**:  
 uf = ChangeForm(request.POST)  
 **if** uf.is\_valid():  
 username = uf.cleaned\_data[**'username'**]  
 old\_password = uf.cleaned\_data[**'old\_password'**]  
 new\_password = uf.cleaned\_data[**'new\_password'**]  
  
 *# 判断用户是否存在* user = User.objects.filter(username=username)  
 **if** user:  
 passwd = User.objects.filter(username=username, password=old\_password)  
 *# 判断用户原密码是否匹配* **if** passwd:  
  
 User.objects.filter(username=username, password=old\_password).update( password=new\_password)  
 *# 如果用户名、原密码匹配则更新密码* info = **'密码修改成功!'  
 return** HttpResponse(info)  
 **else**:  
 info = **'请检查原密码是否输入正确!'  
 return** HttpResponse(info)  
 **elif** len(user) == 0:  
 info = **'请检查用户名是否正确!'  
 return** HttpResponse(info)  
 **else**:  
 uf = ChangeForm()  
 **return** render\_to\_response(**'setting.html'**, {**'uf'**: uf},context\_instance=RequestContext(request))  
  
  
  
  
**def** manage(request):  
 **return** render(request, **'manage.html'**)  
  
  
**def** manage\_new(request):  
 title=request.POST.get(**'title'**,**'TITLE'**)  
 content=request.POST.get(**'content'**,**'CONTENT'**)  
 adminid=request.POST.get(**'adminid'**,**'ADMINID'**)  
 Artical.objects.create(title=title,contnt=content,adminid=adminid)  
 *#获取文章的标题，文章内容，文章管理者ID* articles = Artical.objects.all()  
 **return** render(request, **'home.html'**, {**'articles'**: articles})  
  
**def** message(request):  
 **return** render(request, **'message.html'**)  
  
  
  
**def** article\_page(request,article\_id):  
 *#获取文章的ID* article = Artical.objects.get(pk=article\_id)  
 **return** render(request,**'news.html'**,{**'article'**:article})  
  
  
  
  
**def** article\_modife(request,article\_id):  
 article=Artical.objects.get(pk=article\_id)  
 **return** render(request,**'article\_moife.html'**,{**'article'**:article})  
  
  
**def** article\_modife\_action(request):  
 title = request.POST.get(**'title'**, **'TITLE'**)  
 content = request.POST.get(**'content'**, **'CONTENT'**)  
 article\_id = request.POST.get(**'article\_id'**, **'0'**)  
 *#根据文章的ID来获得文章的标题，文章的内容，文章的ID* article=Artical.objects.get(pk=article\_id)  
 *#修改文章的标题* article.title=title  
 *#修改文章的内容* article.contnt=content  
 *#保存文章的信息。* article.save()  
 **return** render(request,**'news.html'**,{**'article'**:article})