# 项目介绍

* 1. 项目描述简介

通过完成一个仿京东的电子商城项目，让大家掌握企业开发特点，以及解决问题的能力。

该项目会涉及以后工作中的一些非常有代表性的功能。

我们使用git作为项目的版本控制软件。

在项目中会使用很多其他的知识，比如系统架构、维护、优化等等。

* 1. 主要功能模块
  2. 开发环境和技术

|  |  |
| --- | --- |
| 开发环境 | Window |
| 开发工具 | Django |
| 相关技术 |  |

* 1. 项目人员组成周期成本
     1. 人员组成

|  |  |  |
| --- | --- | --- |
| 职位 | 人数 | 备注 |
| 项目经理和组长 | 1 | 一般小公司由项目经理负责管理，中大型公司项目由项目经理或组长负责管理 |
| 开发人员 | 2~3 |  |
| UI设计人员 | 1 |  |
| 前端开发人员 | 1 | 专业前端不是必须的，所以前端开发和UI设计人员可以同一个人 |
| 测试人员 | 1~3 | 有些公司并未有专门的测试人员，测试人员可能由开发人员完成测试。  公司有测试部，测试部负责所有项目的测试。  项目测试由产品经理进行业务测试。 |

* + 1. 项目周期成本

|  |  |  |
| --- | --- | --- |
| 人数 | 周期 | 备注 |
| 3 | 四个月  第一个月需求设计  第二三个月编码实现  第四个月编码实现、测试 |  |

1. 用户登录注册功能模块
   1. 功能需求

1.登录:

1.用户名为手机号,验证手机号格式是否正确和注册,格式不正确提示不是正确手机号,格式正确验证是否注册,如果未注册提示请注册,如果已注册验证密码是否正确,

2.如果正确跳转到个人中心,如果错误提示账号或密码错误

2.注册:

验证字段:手机号,密码,重复密码,验证码

手机号: 1必填,不填提示错误;

2.验证格式是否正确,不正确提示不是有效手机号

密码: 1必填,不填提示错误;

2.验证格式,格式不正确提示错误

重复密码: 1.必填,不填提示错误;

2.验证格式,格式不正确提示错误;

3.验证是否和密码填写一致,不一致提示错误

验证码:

* 1. 流程

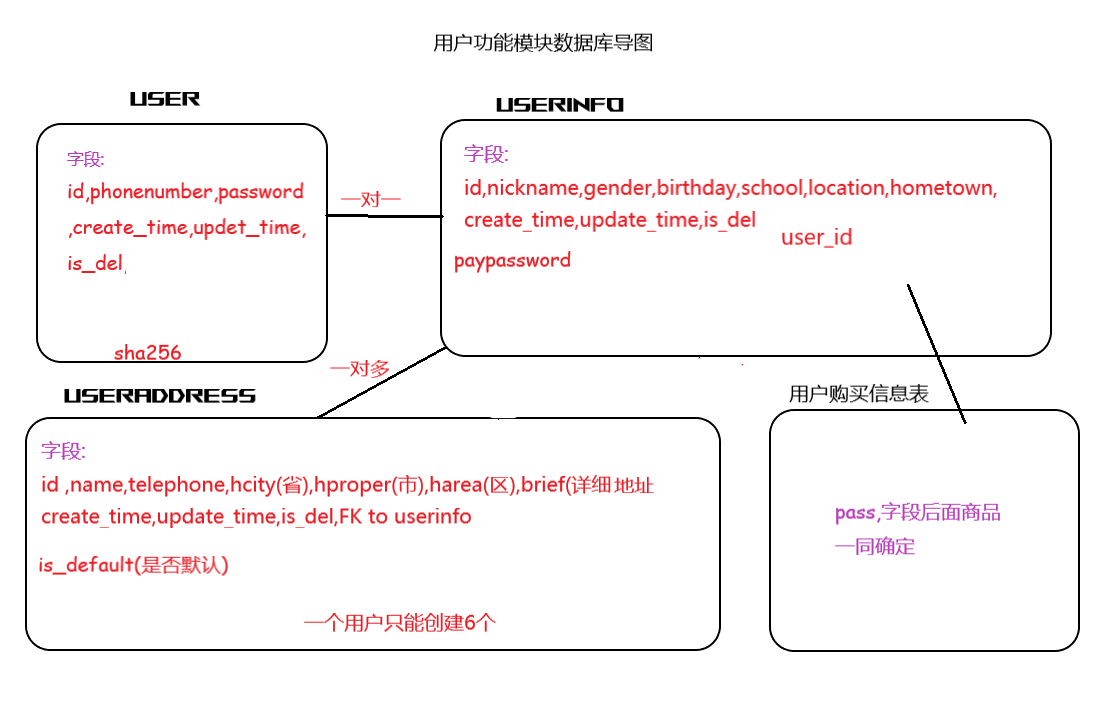
个人中心---->登录页面:

登录成功---发送session---个人中心

登录失败:

注册----注册成功----个人中心

* 1. 设计要点（数据库和页面交互）



Sha256:字段长度设为64位,加密单独建立函数,需要用时在引用

一对一关系表,先添加基本的,所以把关键id设在userinfo表上

Userinfo 字段和表单允许为空

设计model.form,密码,重复密码,用户协议勾选需单独设立字段验证

用户地址表需要添加是否默认选择字段,默认否

优化:

1. 每个表都有相同字段,可以设置模型基类用于继承
2. 登录,修改信息都会设置session,所以可以设置一个函数减少代码
3. 很多页面都会在登录后才能显示,说以可以定义一个装饰器:

装饰器返回调用原函数时要加(),返回新函数时不加()

1. 有些功能视图会以视图类形式创建(1个页面,会有多个请求方式访问时),但原装饰器只能在函数上使用(函数第一参数为request,类第一个函数为self),所有可以通过设置一个视图基类来增强原父视图类的dispatch方法,通过继承,来减少代码量:

用@method\_decorator(check\_session)来装饰dispatch,不能直接用原函数装饰器直接装饰

* 1. 要点难点及解决方案

要点:

确定数据库关系,和字段属性(能否为空),

多个模型(视图)使用相同字段(方法)时可以建立父模型(父视图)继承,

出现多次使用的代码可以尽量进行函数封装,

注册密码要sha256(长度64位)加密,登录时登录密码先加密后再验证

注册成功跳转到登录界面

登录成功设置session

设置session过期时间,不填为两周,0为关闭浏览器过期,固定过期时间,None永不过期  
 request.session.set\_expiry()

session保存到redis数据库减少服务器压力(安装pip install django-redis库,使用时要开始redis-server)

CACHES = {  
 "default": {  
 "BACKEND": "django\_redis.cache.RedisCache",  
 "LOCATION": "redis://127.0.0.1:6379/1",

#域名/端口号/redis数据库(0-15)  
 #有密码格式: redis://[:password]@localhost:6379/0  
 "OPTIONS": {  
 "CLIENT\_CLASS": "django\_redis.client.DefaultClient",  
 }  
 }  
 }  
session存取方式:db(数据库),cache(内存),cache\_db(不能内存就数据库)   
SESSION\_ENGINE = "django.contrib.sessions.backends.cache"  
SESSION\_CACHE\_ALIAS = "default"

难点:

验证码验证:

1. 用户修改密码,手机号,忘记密码功能模块
   1. 需求

忘记密码:用户通过手机号接收验证码,设置新密码.如果手机号存在则替换原密码,手机号不存在则创建新用户

修改密码:用户登录后通过个人中心进入安全设置进入修改密码,旧密码输入一次,新密码输入两次,两次必须一致

更换手机号:用户登录后通过个人中心进入安全设置进入更换手机号,通过新手机号获取验证码替换原手机号

* 1. 流程

忘记密码:

登录---->忘记密码---->验证手机号是否注册,

未注册-->提示手机号未注册

已注册-->发送验证码验证

合法--->更改密码

不合法-->返回提示错误

登录---->忘记密码---->失败返回当前页并进行错误提示;

修改密码:

登录成功--->个人中心--->安全设置--->修改密码

-->修改成功,返回个人中心

-->修改失败,返回当前页并进行错误提示

更换手机号:

登录成功--->个人中心--->安全设置--->更换手机号

--->验证手机号是否存在,

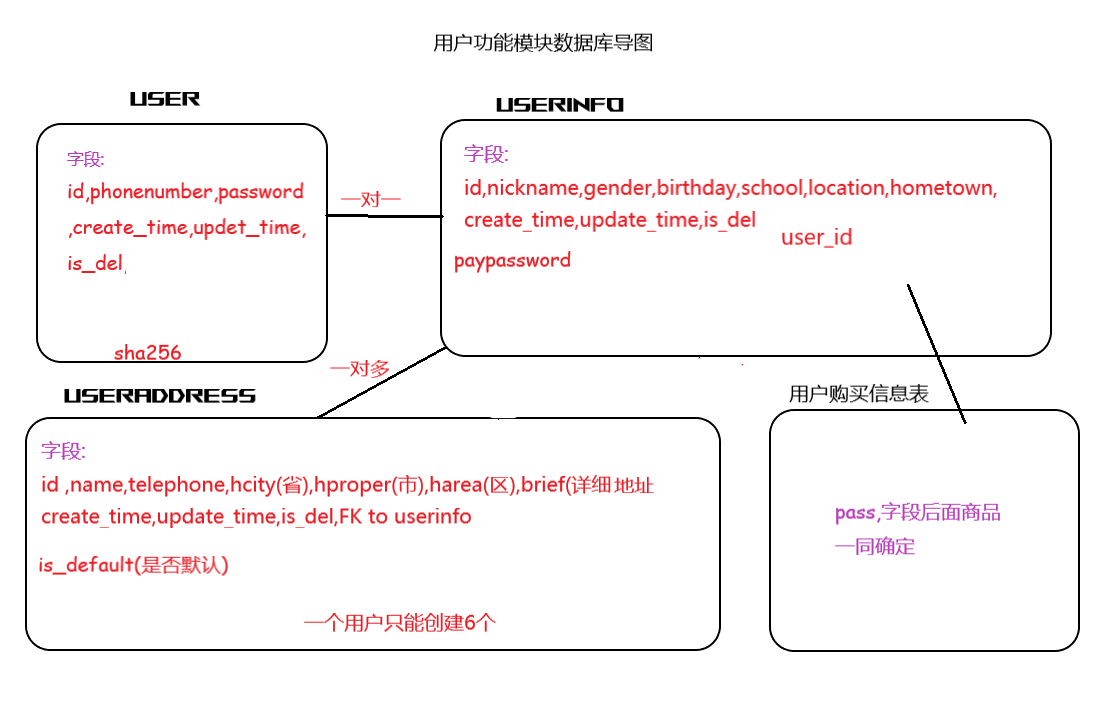
---存在,返回并提示错误信息

不存在,发送验证码验证

验证通过-->修改成功,返回个人中心

验证失败-->修改失败,返回当前页并进行错误提示

* 1. 设计要点（数据库和页面交互）



忘记密码:

根据post请求传入的手机号进行判断,存在修改该手机号的密码,不存在添加手机号和密码

Get\_or\_create

修改密码:

根据Session传入user的主键ID查询原密码

验证原密码是否正确,正确验证新密码是否合法,不正确返回进行错误提示

合法,修改原密码,不合法返回进行错误提示

更换手机号:

根据Session传入user的主键ID查询手机号

验证新手机号是否合法

不合法返回进行错误提示

合法,验证验证码,

正确:修改手机

不正确:返回进行提示

* 1. 要点难点及解决方案

要点:

忘记密码的form可以继承注册的form,去掉用户协议验证属性和重新修改手机号验证方法即可

修改密码时要验证输入密码是否输入正确,在form中自定义一个方法,使用session得到id验证,

使用add\_error添加错误信息

难点:

短信验证码

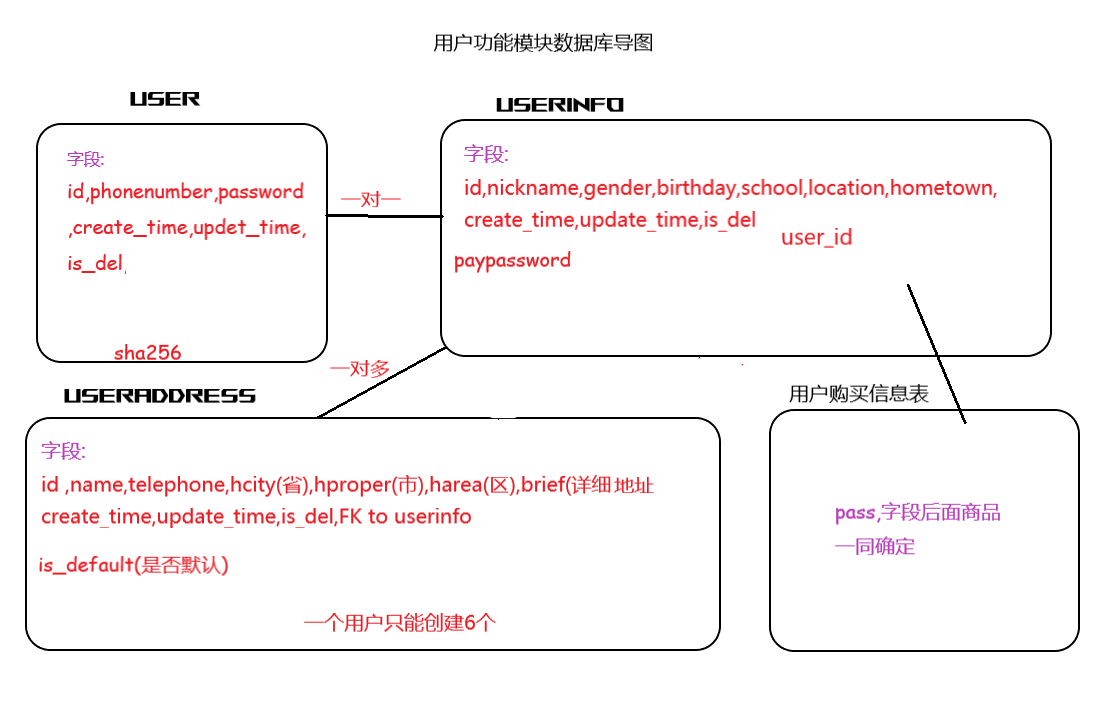
1. 用户个人资料功能模块
   1. 需求

展示个人资料,头像切换

* 1. 流程

个人中心--->个人资料---保存成功返回个人中心页面

* 1. 设计要点（数据库和页面交互）



根据关联字段添加用户信息

字段属性null=True的时候blank也要设置True

* 1. 要点难点及解决方案

要点:

!!!在注册时就要在userinfo表中添加关联ID,添加字段名要和数据库记录名一致

可以增加JS遮蔽电话号码中间4位数:

<script type="application/javascript">

var phone = $('#phone').val();

var newphone = "";

if (phone.length > 0) {

for (var i = 0; i < phone.length; i++) {

if (i < 3 || i >= phone.length - 4) {

newphone += phone[i];

}

else {

newphone += '\*';

}

$('#phone').val(newphone)

}

}

</script>

难点:

更换头像:

1. 验证码验证功能模块
   1. 需求

在注册,忘记密码,和更换手机号时要使用短信验证码登录

* 1. 流程

点击获取验证码---->判断手机号码填写格式

错误--->提示格式错误

正确--->发送验证码--->验证验证码是否正确

* 1. 设计要点（数据库和页面交互）

在父模板中的预留位添加ajax请求:

需在加载完页面再执行,先利用id或class绑定点击事件,

和使用.val()方式得到电话号码进行正则验证格式是否正确,

错误alter弹出显示错误,正确进行ajax请求:

<script type="application/javascript">

{#页面加载完再执行#}

$(function () {

$('.yzm-hq').on('click', function () {

{#绑定点击事件#}

var phone\_num = $('#phone\_num').val();

var re = /^1[3-9]\d{9}$/;

{#获取手机号,然后正则验证#}

if (!re.test(phone\_num)) {

alert('手机号码格式不正确!');

return;

}

var self = this;

{#绑定获取验证码按钮#}

//发送ajax请求

$.ajax({

type: 'post',

url: '{% url "user:发送验证码" %}',

data: {'phone\_num': phone\_num, 'csrfmiddlewaretoken': '{{ csrf\_token }}'},

dataType: 'json',

success: function (data) {

//响应数据,判断是否成功

if (data.error == 0) {

//发送成功按钮变灰色

$('.yzm-hq').css("background-color",'gray');

//禁止点击

$(self).prop('disabled', true);

//显示倒计时

var time = 120;#2分钟后过期

var msg = time + '秒后重新发送';

$(self).val(msg); //更改注册按钮的值

var timer = window.setInterval(function () {

//设定定时执行方法

time -= 1;

if (time == 0) {

//倒计时结束变回原样

$('.yzm-hq').css("background-color", '#76bb2a');

$(self).prop('disabled', false);

$(self).val('获取验证码');

//清除定时方法

window.clearInterval(timer);

return;

}

msg = time + '秒后重新发送';

$(self).val(msg)

}, 1000); //毫秒

}else {

alert(data.errmsg)

}

}

})

})

});

</script>

在视图中添加发送验证码试图类,根据请求方式获取手机号,并用random.randint和列表推导式

获取随机验证码,数字需转换为字符串类型:

"".join([str(random.randint(0,9)) for \_ in range(6)])

* 1. 要点难点及解决方案

要点:

点击获取验证码后,点击按钮要进入倒计时并禁止点击触发事件

$(self).prop('disabled', true)或$(self).attr('disabled', 'disabled')

验证码根据键值对用redis保存,减少服务器压力

并且redis自带expire方法设置键值过期时间和incr方法限制获取验证码次数

注意,在redis中根据键get到的值为二进制,数字可以强转,字符串职能decode(utf-8)

进行转码

短信验证码:

设置一个视图函数,接收ajax请求

注册云服务获取密钥并下载安装服务商第三方包

获取签名和模板code

在setting中添加密钥值

添加链接服务商接口方法:

# 完成 定义一个方法 发送短消

# 注意：不要更改

REGION = "cn-hangzhou"

PRODUCT\_NAME = "Dysmsapi"

DOMAIN = "dysmsapi.aliyuncs.com"

acs\_client = AcsClient(ACCESS\_KEY\_ID, ACCESS\_KEY\_SECRET, REGION)

region\_provider.add\_endpoint(PRODUCT\_NAME, REGION, DOMAIN)

def send\_sms(business\_id, phone\_numbers, sign\_name, template\_code, template\_param=None):

smsRequest = SendSmsRequest.SendSmsRequest()

# 申请的短信模板编码,必填

smsRequest.set\_TemplateCode(template\_code)

# 短信模板变量参数

if template\_param is not None:

smsRequest.set\_TemplateParam(template\_param)

# 设置业务请求流水号，必填。

smsRequest.set\_OutId(business\_id)

# 短信签名

smsRequest.set\_SignName(sign\_name)

# 数据提交方式

# smsRequest.set\_method(MT.POST)

# 数据提交格式

# smsRequest.set\_accept\_format(FT.JSON)

# 短信发送的号码列表，必填。

smsRequest.set\_PhoneNumbers(phone\_numbers)

# 调用短信发送接口，返回json

smsResponse = acs\_client.do\_action\_with\_exception(smsRequest)

# TODO 业务处理

return smsResponse

在发送验证码视图中根据签名调用链接方法发送短信:

\_\_business\_id = uuid.uuid1()

params ="{\"code\":\"%s\",\"product\":\"Supermarket\"}" % random\_code

# print(params)

rs = send\_sms(\_\_business\_id, phone, "注册验证", "SMS\_2245271", params)

print(rs.decode('utf-8'))

验证码验证:

在form中进行综合验证:

1.添加captcha验证字段

2.要在验证密码一致后验证:

3.在redis中获取的值为二进制转码

4.要使用try,因为验证有过期时间!

1. 验证码验证功能模块
   1. 需求
   2. 流程
   3. 设计要点（数据库和页面交互）
   4. 要点难点及解决方案
2. 验证码验证功能模块
   1. 需求
   2. 流程
   3. 设计要点（数据库和页面交互）
   4. 要点难点及解决方案
3. 验证码验证功能模块
   1. 需求
   2. 流程
   3. 设计要点（数据库和页面交互）
   4. 要点难点及解决方案
4. 验证码验证功能模块
   1. 需求
   2. 流程
   3. 设计要点（数据库和页面交互）
   4. 要点难点及解决方案
5. 验证码验证功能模块
   1. 需求
   2. 流程
   3. 设计要点（数据库和页面交互）
   4. 要点难点及解决方案
6. 常见面试问题
7. 常见面试问题
8. 常见面试问题
9. 常见面试问题