测试模块：注册模块，条件组合覆盖

测试工具：IntelliJ IDEA

测试人员：冯一鸣

注册模块代码：

def register(request):  
 *"""  
 功能：注册  
 parm:(request)  
 注册表单类（email, username, pwd1, pwd2）  
 如果注册信息正确：  
 写入数据库  
 提示注册成功。  
 否则：  
 重新发送一张表单  
 @3y  
 """* curtime=time.strftime("%Y-%m-%d %H:%M:%S",time.localtime());  
 if request.method == 'POST':  
 if request.is\_ajax():  
 code\_code = ''  
 #if 'send\_check' in request.POST:  
 studyname = request.POST.get('studyname', '') # 学号  
 email = studyname + request.POST.get('domain', '') # 邮箱  
 code\_code = send\_send\_mail(email)  
 ret={"msg":"验证码发送成功!"}  
 return HttpResponse(json.dumps(ret))  
 elif 'registerbutton' in request.POST:  
 nickname = request.POST.get('nickname', '') # 用户名  
 password = request.POST.get('password', '')  
 repassword = request.POST.get('repassword', '')  
 studyname = request.POST.get('studyname', '') # 学号  
 email = studyname + request.POST.get('domain', '')  
 check\_get = request.POST.get('check\_get', '')  
 rs = User.objects.filter(UID=nickname)  
 if len(nickname) > 20:  
 return render(request, 'register-no.html', {'account\_msg': '用户名不能超过20个字符'})  
 elif nickname == "":  
 return render(request, 'register-no.html', {'account\_msg': '用户名不能为空'})  
 elif len(rs) > 0:  
 return render(request, 'register-no.html', {'account\_msg': '账号已存在'})  
 if password == "":  
 return render(request, 'register-no.html', {'password\_msg': '密码不能为空'})  
 elif len(password) < 6 or len(password) > 20:  
 return render(request, 'register-no.html', {'password\_msg': '密码必须在6-20位'})  
 if repassword == "":  
 return render(request, 'register-no.html', {'repassword\_msg': '密码不能为空'})  
 elif len(repassword) < 6 or len(repassword) > 20:  
 return render(request, 'register-no.html', {'repassword\_msg': '密码必须在6-20位'})  
 if password != repassword:  
 return render(request, 'register-no.html', {'repassword\_msg': '两次密码输入不一致'})  
 rs = User.objects.filter(username=studyname)  
 if len(rs) > 0:  
 return render(request, 'register-no.html', {'studyname\_msg': '学号已注册'})  
 if check\_get=="":  
 return render(request, 'register-no.html', {'emaicode\_msg': '验证码为空'})  
 try:  
 now\_info = Emailcode.objects.get(code=check\_get, email=email)  
 except Emailcode.DoesNotExist:  
 return render(request, 'register-no.html', {'emaicode\_msg': '验证码错误'})  
 # if now\_info == None:  
 # return render(request, 'register.html', {'studyname\_msg': '验证码错误'})  
 current\_time = timezone.now()  
 send\_time = now\_info.send\_time  
 past\_time = current\_time.timestamp() - send\_time.timestamp()  
 #15min  
 if (past\_time > 900):  
 return render(request, 'register-no.html', {'emaicode\_msg': '验证码过期'})  
 user = User.objects.create\_user(username=studyname, password=repassword, UID=nickname, email=email,  
 UCreateTime=curtime)  
 #注册成功  
 return render(request, 'successRegister.html')  
 #导航栏按钮  
 if 'register\_login' in request.POST:  
 # return render(request, '/account/register')  
 return HttpResponseRedirect('/account/login/') #####  
 elif 'search' in request.POST:  
 return HttpResponseRedirect(reverse('search'))  
 else:  
 return HttpResponse("ERROR")  
 elif request.method == 'GET':  
 return render(request, 'register-no.html')  
 else:  
 return HttpResponse("ERROR")

测试结果：

1. 用户名不输入，点击确定后运行如下：



2.用户名：zxcvbnmasdfghjklqwertyuio，点击确定后运行如下：



3.用户名：zdy，点击确定后运行如下：



4.登陆密码为空，点击确定后运行如下：



5. 用户名：fym登陆密码：123，点击确定后运行如下：



6. 用户名：fym登陆密码：1234567890zxcvbnmasdfghjkqwertyuio，点击确定后运行如下：



7. 用户名：fym，登陆密码：123456，确认登陆密码为空，点击确定后运行如下：



8. 用户名：fym，登陆密码：123456，确认登陆密码：123，点击确定后运行如下：



9. 用户名：fym，登陆密码：123456，确认登陆密码1234567890zxcvbnmasdfghjkqwertyuio，点击确定后运行如下：



10.用户名 fym登陆密码：12346确认登陆密码：1234567，点击确定后运行如下：



11.用户名 fym登陆密码：12346确认登陆密码：123456 ，邮箱：31601390，验证码为空，点击确定后运行如下：



12.用户名 fym登陆密码：12346确认登陆密码：123456 ，邮箱：31601390，验证码：1，点击确定后运行如下：



13.用户名 fym登陆密码：12346确认登陆密码：1234567，邮箱31601418，点击确定后运行如下：



14.用户名 fym登陆密码：12346确认登陆密码：1234567，邮箱31601418，验证码正确，点击确定后运行如下：



|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 用例名称 | 注册 | | | |
| 测试类型 | 单元测试 | | | |
| 测试目的 | 测试各种情况下用户能否能正常注册 | | | |
| 测试方法 | 在网站注册界面和数据库端进行条件组合覆盖测试 | | | |
| 测试用例设计原则 | 条件组合覆盖 | | | |
| 正常测试情况 | | | | |
| 编号 | 测试数据 | 数据描述 | 预期结果 | 实际测试结果 |
| 1 | 用户名为空 | 用户名不输入 | 提示错误 | 提示：用户名不能为空 |
| 2 | 用户名：zxcvbnmasdfghjklqwertyuio | 用户名超过20个 | 提示错误 | 用户名不能超过20个字符 |
| 3 | 用户名：zdy | 用户名重复 | 提示错误 | 账号已存在 |
| 4 | 登陆密码为空 | 密码不输入 | 提示错误 | 密码不能为空 |
| 5 | 用户名：fym登陆密码：123 | 密码小于6为 | 提示错误 | 密码必须在6-20位 |
| 6 | 用户名：fym登陆密码：1234567890zxcvbnmasdfghjkqwertyuio | 密码大于20为 | 提示错误 | 密码必须在6-20位 |
| 7 | 确认登陆密码为空 | 密码不输入 | 提示错误 | 密码不能为空 |
| 8 | 用户名：fym确认登陆密码：123 | 密码小于6为 | 提示错误 | 密码必须在6-20位 |
| 9 | 用户名：fym确认登陆密码：1234567890zxcvbnmasdfghjkqwertyuio | 密码大于20为 | 提示错误 | 密码必须在6-20位 |
| 10 | 用户名 fym登陆密码：12346  确认登陆密码：1234567 | 两次密码不一致 | 提示错误 | 两次密码输入不一致 |
| 11 | 验证码为空 | 验证码不输入 | 提示错误 | 验证码为空 |
| 12 | 验证码：0 | 验证码错误 | 提示错误 | 验证码错误 |
| 13 | 邮箱：31601418 | 邮箱重复 | 提示错误 | 学号已注册 |
| 14 | 用户名：fym 登陆密码：123456  确认登陆密码：123456 邮箱31601390 验证码正确 | 所有数据正确 | 提示注册成功 | 注册成功 |