软件测试技术

实训报告

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 班级 | 学号 | 姓名 |  | 成绩 |
| 21软件4班 | 21137040221 | 许楷涛 | 登陆注册模块，超星自动化模块，性能测试 |  |
| 21软件4班 | 1613234113 | 阮子豪 | 登陆注册模块，搜索模块，性能测试 |  |
|  |  |  |  |  |

目录

[1.功能需求分析 2](#_Toc138786033)

[1.1 功能结构思维导图 2](#_Toc138786034)

[1.2 用户界面描述 2](#_Toc138786035)

[1.3 逻辑规则 3](#_Toc138786036)

[1.4 数据状态 3](#_Toc138786037)

[1.5 模块关联 4](#_Toc138786038)

[1.6 权限差异 4](#_Toc138786039)

[2.功能测试用例设计 4](#_Toc138786040)

[2.1 登录功能测试用例 4](#_Toc138786041)

[2.2 注册功能测试用例(手机/邮箱注册) 5](#_Toc138786042)

[2.3 搜索功能测试用例 7](#_Toc138786043)

[3.登陆注册功能自动化测试 8](#_Toc138786044)

[4.搜索功能手动测试 20](#_Toc138786045)

[5超星自动化测试 22](#_Toc138786046)

[7.性能测试设计与执行 26](#_Toc138786047)

[8.缺陷报告 32](#_Toc138786048)

[9.测试总结 35](#_Toc138786049)

[6.1测试环境与配置 35](#_Toc138786050)

[6.2 测试方法 35](#_Toc138786051)

[6.3 用例汇总 35](#_Toc138786052)

[6.4 缺陷分析 36](#_Toc138786053)

[6.5测试结论（或对系统的建议） 36](#_Toc138786054)

[8.个人实训周心得体会 37](#_Toc138786055)

测试对象：TP商城

前台用户系统

http://hmshop-test.itheima.net/

后台管理系统

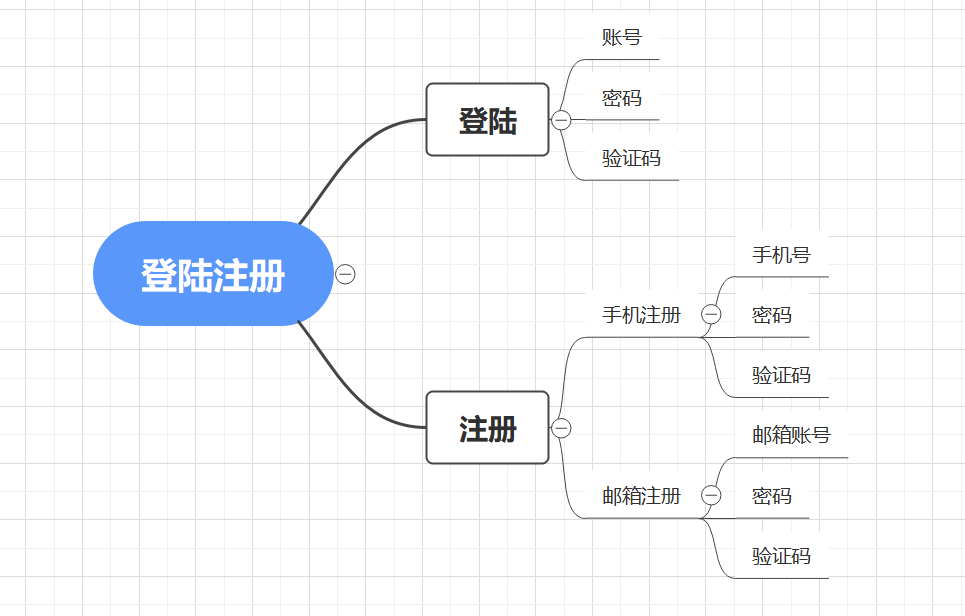
http://hmshop-test.itheima.net/Admin/Admin/login

账号/密码：admin/ HM\_2023\_test

# 1.功能需求分析

## 功能结构思维导图

登陆注册：



## 用户界面描述

(1)登陆页面:

1. 首次登入商城首页为游客状态，网站头部显示登录注册两个连接

2. 注册用户登录后，网站头部显示用户的用户名全部内容和安全退出。邮箱注册后用户名为邮箱地址。

3. 登录方式：有用户名登陆方式（手机，邮箱），和第三方登录（支付宝，QQ，微信 ）本次需求只关注用户名登录方式。

(2)注册页面：

1. 首次登入商城首页为游客状态，网站头部显示登录注册两个链接

2. 注册页面，默认为手机号注册

3．可以切换为邮箱注册

(3)搜索列表页面：

1.左边为推荐内容

2.右边为搜索商品展示

## 1.3 逻辑规则

1. 登陆页面：

手机号或者邮箱账号以及密码验证码都不可为空

1. 注册页面：
2. 手机号：必填项，第一位为1，第二位非2，的11位自然数，不能重复

邮箱：正确格式的邮箱格式 [xxxx@[126/sina/qq].com](mailto:xxxx@[126/sina/qq%5d.com)

注意：对于邮箱注册，仅验证邮箱格式是否正确

**xxxx-长度为4-16，类型为数字，字母，下划线组合（可以是纯字母或者数字，不能以下划线开头）**

1. 验证码：必填项，字母或者数字，与图片一致，点击刷新按钮可以更新验证码
2. 设置密码：必填项，要求同输入框提示，6-16位大小写英文字母，数字或符号的组合 (可以是纯字母，数字，符号）
3. 确认密码：必填项，同设置密码
4. 推荐人手机：（非必填，一旦填写必须为已注册账户手机号），如果填写的是未注册手机号则给出提示:”该推荐人手机号不系统不存在！”注册失败！
5. 我已阅读并同意：勾选之后才可以点击同意协议并注册

## 1.4 数据状态

1. 登陆页面：

1.首次登入商城首页为游客状态，网站头部显示登录注册两个连接

2. 注册用户登录后，网站头部显示用户的用户名全部内容和安全退出。邮箱注册后用户名为邮箱地址。

3. 登录方式：有用户名登陆方式（手机，邮箱），和第三方登录（支付宝，QQ，微信 ）本次需求只关注用户名登录方式。

(2)注册页面：

1. 首次登入商城首页为游客状态，网站头部显示登录注册两个链接

2. 注册页面，默认为手机号注册

3．可以切换为邮箱注册

## 1.5 模块关联

(1）登陆注册：

1.进行注册账号后可以用于该账号用于下次登陆

## 1.6 权限差异

(1)登陆注册：

1. 已注册，可登陆
2. 未注册不可登陆，需先注册

# 2.功能测试用例设计

## 登录功能测试用例

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 用例编号 | 模块 | 页面位置 | 功能点 | 测试标题 | 重要级别 | 预置条件 | 输入 | 执行步骤 | 预期输出 |
| A001 | 正确登陆验证 | 登陆页面 | 登陆功能 | 登陆验证 | 高 | 进入首页，点击登陆 | 输入正确手机号或邮箱账号，密码，验证码 | 点击登陆按钮 | 无提示语，跳转到我的商城界面 |
| A002 | 账号验证 | 登陆页面 | 登陆功能 | 登陆验证 | 高 | 进入首页，点击登陆 | 非正确的手机号或邮箱格式 | 点击登陆 | 出现弹窗提示账号格式不匹配 |
| A003 | 账号验证 | 登陆页面 | 登陆功能 | 登陆验证 | 高 | 进入首页，点击登陆 | 输入正确格式（手机号，邮箱）未注册的账号，密码，正确验证码 | 点击登录 | 出现弹出信息提示账号不存在 |
| A004 | 账号为空 | 登陆页面 | 登陆功能 | 登陆验证 | 高 | 进入首页，点击登陆 | 账号为空 | 点击登录 | 出现弹出信息提示账号不能为空 |
| A005 | 密码为空 | 登陆页面 | 登陆功能 | 登陆验证 | 高 | 进入首页，点击登陆 | 账号输入正确，未输入密码，验证码填写正确 | 点击登录 | 出现弹出信息提示密码不能为空 |
| A006 | 密码错误 | 登陆页面 | 登陆功能 | 登陆验证 | 高 | 进入首页，点击登陆 | 账号输入正确，输入错误密码，验证码填写正确 | 点击登陆 | 出现弹出信息提示密码错误 |

## 注册功能测试用例(手机/邮箱注册)

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 用例编号 | 模块 | 页面位置 | 功能点 | 测试标题 | 重要级别 | 预置条件 | 输入 | 执行步骤 | 预期输出 |
| B001 | 正确注册验证 | 注册页面 | 注册功能 | 注册验证 | 高 | 进入首页，点击注册 | 输入符合条件的手机号/邮箱，验证码，密码，勾选同意协议 | 点击注册 | 跳转到首页 |
| B002 | 重复注册 | 注册页面 | 注册功能 | 注册验证 | 高 | 进入首页，点击注册 | 输入符合条件的手机号/邮箱（已存在），验证码，密码，勾选同意协议 | 点击注册 | 出现弹出信息提示账号已存在 |
| B003 | 手机号/邮箱验证 | 注册页面 | 注册功能 | 注册验证 | 高 | 进入首页，点击注册 | 不输入手机号/邮箱或者输入错误的手机号/邮箱，点击注册 | 点击注册 | 出现弹出信息提示  请用手机号或邮箱注册 |
| B004 | 验证码错误 | 注册页面 | 注册功能 | 注册验证 | 高 | 进入首页，点击注册 | 输入错误的验证码，点击注册按钮 | 点击注册 | 出现弹出信息提示图形验证码错误 |
| B005 | 验证码为空 | 注册页面 | 注册功能 | 注册验证 | 高 | 进入首页，点击注册 | 不输入验证码，点击注册 | 点击注册 | 出现弹出信息提示图形验证码不能为空 |
| B006 | 密码为空 | 注册页面 | 注册功能 | 注册验证 | 高 | 进入首页，点击注册 | 未输入密码，点击注册按钮 | 点击注册 | 出现弹出信息提示  请输入用户名或密码 |
| B007 | 密码验证 | 注册页面 | 注册功能 | 注册验证 | 高 | 进入首页，点击注册 | 两次密码输入不一致，点击注册按钮 | 点击注册 | 出现弹出信息提示  俩次密码不一致 |
| B008 | 密码验证 | 注册页面 | 注册功能 | 注册验证 | 高 | 进入首页，点击注册 | 密码输入不符合要求，点击注册 | 点击注册 | 出现弹出信息提示  密码为6-16位大小写英文字符、数字或符号 |
| B009 | 手机号验证/邮箱验证 | 注册页面 | 注册功能 | 注册验证 | 高 | 进入首页，点击注册 | 不输入手机号/邮箱，或者输入错误的手机号/邮箱，验证码，密码正确输入，点击注册 | 点击注册 | 出现弹出信息提示  请输入正确手机号/请输入正确邮箱 |

## 2.3 搜索功能测试用例

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 用例编号 | 模块 | 页面位置 | 功能点 | 测试标题 | 重要级别 | 预置条件 | 输入 | 执行步骤 | 预期输出 |
| C001 | 搜索功能 | 首页页面 | 输入 | 支持汉字拼音单词输入, | 中 | 无 | Shouji,手ji,shou机 | 在搜索框进行输入 | 联想框出现手机字样 |
| C002 | 搜索功能 | 首页页面 | 输入 | 错别字纠正输入 | 中 | 无 | 首机 | 在搜索框进行输入 | 根据后台词库匹配，联想框出现手机字样 |
| C003 | 搜索功能 | 首页页面 | 输入 | 支持同音字搜索 | 中 | 无 | binxiang | 在搜索框进行输入 | 联想框出现冰箱字样 |
| C004 | 搜索功能 | 首页页面 | 输入 | 支持数字同音 | 中 | 无 | 1，2 | 在搜索框进行输入 | 联想框出现1或者一的字样 |
| C005 | 搜索功能 | 首页页面 | 输入 | 支持标点符号释义 | 中 | 无 | 手机】 | 在搜索框进行输入 | 联想框出现手机的字样 |
| C006 | 搜索功能 | 首页页面 | 输入 | 自动补全 | 中 | 无 | 手机 | 在搜索框进行输入 | 联想框有品牌匹配，常用匹配，出现手机，手机壳等 |
| C007 | 搜索功能 | 首页页面 | 匹配 | 匹配分类、标题、商品名称、品牌、厂商 | 中 | 无 | 华为 | 在搜索框进行输入 | 商品中出现华为手机，电脑，数据线等 |
| C008 | 搜索功能 | 首页页面 | 匹配 | 分词匹配 | 中 | 无 |  | 在搜索框进行输入 |  |
| C009 | 搜索功能 | 首页页面 | 匹配 | 匹配为空 | 中 | 无 | 无 | 无 | 推荐内容 |
| C010 | 搜索功能 | 搜索列表页面 | 排序 | 销量、人气、收藏、价格、最新 | 中 | 无 | 手机 | 无 | 默认销量排序 |

# 3.登陆注册功能自动化测试

封装打开网站驱动和获取消息

from selenium import webdriver  
  
  
class DriverUtil:  
 \_driver = None  
  
 # 打开浏览器，进入网站  
 @classmethod  
 def get\_driver(cls):  
 if cls.\_driver is None:  
 cls.\_driver = webdriver.Chrome()  
 cls.\_driver.get('http://hmshop-test.itheima.net/')  
 cls.\_driver.implicitly\_wait(3)  
 cls.\_driver.maximize\_window()  
  
 return cls.\_driver  
  
 # 退出网站  
 @classmethod  
 def quit\_driver(cls):  
 cls.\_driver.quit()  
 cls.\_driver = None  
  
  
# 获取弹出框的提示消息  
# :return: 消息文本内容  
def get\_msg():  
  
 msg = DriverUtil.get\_driver().find\_element\_by\_class\_name('layui-layer-content').text  
 return msg

登陆：对象层>控制层>业务层

class LoginPage:  
 # 对象层  
 def \_\_init\_\_(self):  
 self.driver = DriverUtil.get\_driver()  
  
 # 用户名  
 self.username = (By.ID, 'username')  
 # 密码  
 self.password = (By.ID, "password")  
 # 验证码  
 self.verify\_code = (By.ID, "verify\_code")  
 # 登陆按钮  
 self.login\_btn = (By.NAME, "sbtbutton")  
  
 # 定义操作方法,分别是用户名，密码，验证码，登陆按钮  
 def find\_username(self):  
 return self.driver.find\_element(\*self.username)  
  
 def find\_password(self):  
 return self.driver.find\_element(\*self.password)  
  
 def find\_verify\_code(self):  
 return self.driver.find\_element(\*self.verify\_code)  
  
 def find\_login\_btn(self):  
 return self.driver.find\_element(\*self.login\_btn)

class LoginHandle:  
 # 操作层  
 def \_\_init\_\_(self):  
 self.login\_page = LoginPage()  
  
 # 实现操作方法,分别是用户名，密码，验证码，登陆按钮  
 def input\_username(self, username):  
 self.login\_page.find\_username().send\_keys(username)  
  
 def input\_password(self, pwd):  
 self.login\_page.find\_password().send\_keys(pwd)  
  
 def input\_verify\_code(self, code):  
 self.login\_page.find\_verify\_code().send\_keys(code)  
  
 def click\_login\_btn(self):  
 self.login\_page.find\_login\_btn().click()

class LoginProxy:  
 # 业务层  
  
 def \_\_init\_\_(self):  
 self.login\_handle = LoginHandle()  
   
 # 实现真正业务  
 def login(self, username, password, code):  
 self.login\_handle.input\_username(username)  
 self.login\_handle.input\_password(password)  
 self.login\_handle.input\_verify\_code(code)  
 self.login\_handle.click\_login\_btn()

登陆测试数据和测试层：

{  
 "login\_username\_error": {  
 "username": "3133825137@qq",  
 "password": "123456",  
 "code": "8888",  
 "is\_success": false,  
 "expect": "账号格式不匹配!"  
 },  
 "login\_username\_no": {  
 "username": "1771069481@qq.com",  
 "password": "123456",  
 "code": "8888",  
 "is\_success": false,  
 "expect": "账号不存在!"  
 },  
 "login\_username\_empty": {  
 "username": "",  
 "password": "123456",  
 "code": "8888",  
 "is\_success": false,  
 "expect": "账号不能为空!"  
 },  
 "login\_password\_empty": {  
 "username": "3133825137@qq.com",  
 "password": "",  
 "code": "8888",  
 "is\_success": false,  
 "expect": "密码不能为空!"  
 },  
 "login\_password\_error": {  
 "username": "3133825137@qq.com",  
 "password": "123",  
 "code": "8888",  
 "is\_success": false,  
 "expect": "密码错误!"  
 },  
 "login\_success": {  
 "username": "3133825137@qq.com",  
 "password": "123456",  
 "code": "8888",  
 "is\_success": true,  
 "expect": "3133825137@"  
 }  
}

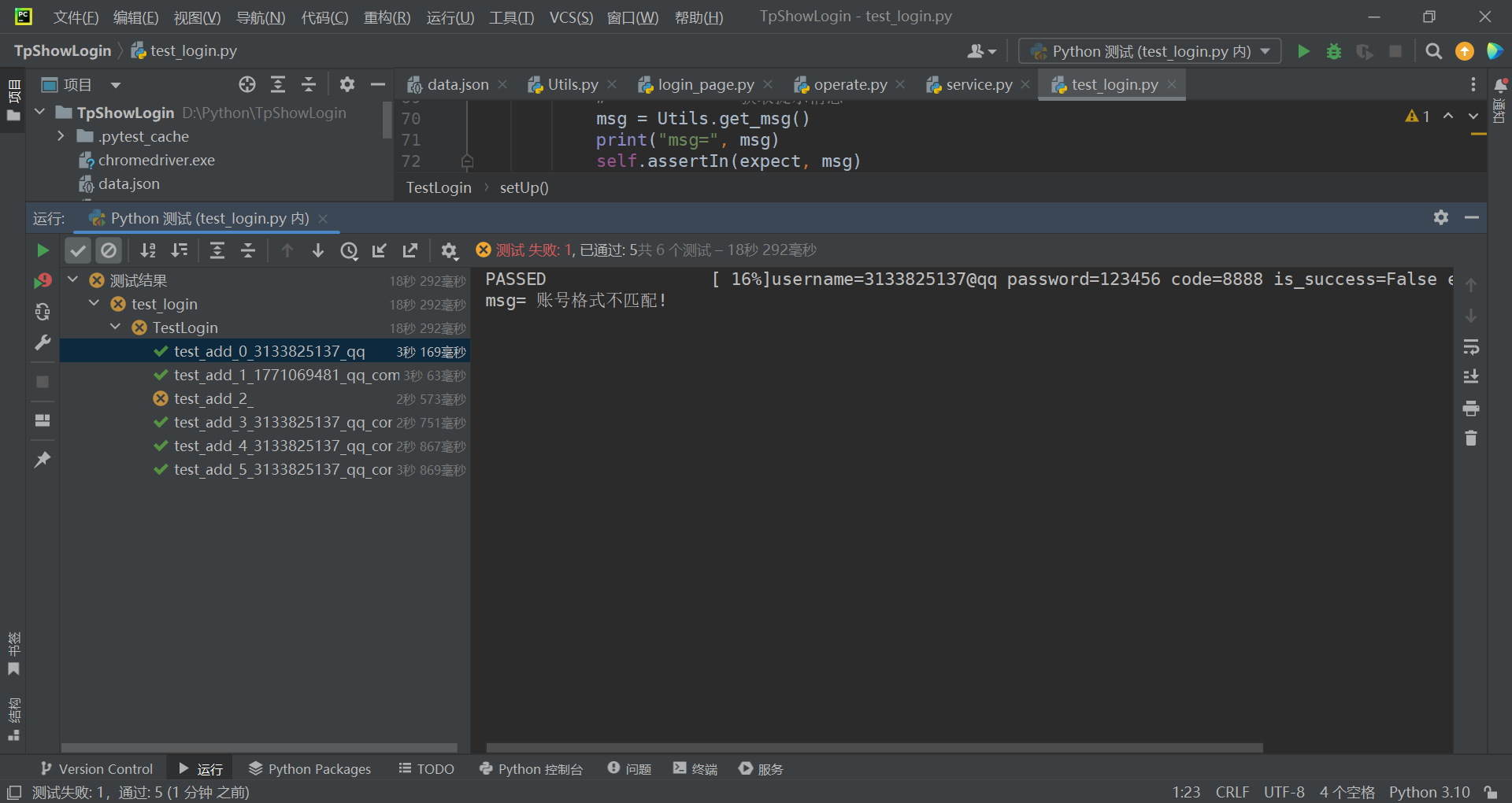
测试层

# 获取数据并存储  
def build\_data():  
 test\_data = []  
 with open('./data.json', encoding='UTF-8') as f:  
 json\_data = json.load(f)  
 for login\_data in json\_data.values():  
 test\_data.append((  
 login\_data.get("username"),  
 login\_data.get("password"),  
 login\_data.get("code"),  
 login\_data.get("is\_success"),  
 login\_data.get('expect')  
 ))  
 print("test\_data", test\_data)  
 return test\_data  
  
  
class TestLogin(unittest.TestCase):  
 @classmethod  
 def setUpClass(cls):  
 # 获取驱动  
 cls.driver = DriverUtil.get\_driver()  
 # 获取业务  
 cls.login\_proxy = LoginProxy()  
  
 @classmethod  
 def tearDownClass(cls):  
 # 关闭驱动  
 DriverUtil.quit\_driver()  
 # cls.driver.quit()  
   
 # 因为只是打开一次浏览器，setUp作为一个重复操作  
 def setUp(self):  
 # 进入首页，点击登陆连接  
 self.driver.get("http://hmshop-test.itheima.net/")  
  
 # self.driver.get("http://hmshop-test.itheima.net/Home/user/login.html")  
 self.driver.find\_element\_by\_link\_text("登录").click()  
  
 # def setDown(self):  
 # self.driver.find\_element\_by\_link\_text('安全退出').click()  
   
 # 加载数据驱动  
 @parameterized.expand(build\_data)  
 def test\_add(self, username, password, code, is\_success, expect):  
 print("username={} password={} code={} is\_success={} expect={}".format(  
 username, password, code, is\_success, expect  
 ))  
  
 # 登陆  
 self.login\_proxy.login(username, password, code)  
 # 验证arg1是arg2的子串，不是则fail  
 if is\_success:

# 验证登陆后是否为该用户名，判断真正登陆成功了嘛  
 self.assertIn(expect, self.driver.find\_element\_by\_link\_text('3133825137@').text)  
 else:  
 # 获取提示消息  
 msg = Utils.get\_msg()  
 print("msg=", msg)  
 self.assertIn(expect, msg)  
  
  
if \_\_name\_\_ == '\_\_main\_\_':  
 unittest.main()

登陆测试结果：

|  |  |
| --- | --- |
| 测试编号 | 测试结果 |
| A001 | 通过 |
| A002 | 通过 |
| A003 | 通过 |
| A004 | 不通过 |
| A005 | 通过 |
| A006 | 通过 |



注册：对象层>控制层>业务层

class PostPage:  
 # 对象层  
 def \_\_init\_\_(self):  
 self.driver = DriverUtil.get\_driver()  
  
 # 用户名  
 self.username = (By.ID, 'username')  
 # 验证码  
 self.verify\_code2 = (By.ID, "verify\_code2")  
 # 密码  
 self.password = (By.ID, "password")  
 # 再次密码  
 self.password\_again = (By.CLASS\_NAME, "J\_password2")  
 # 推荐人手机  
 self.phone = (By.NAME, "invite")  
 # 注册按钮  
 self.post\_btn = (By.CLASS\_NAME, "regbtn")  
  
 # 定义操作方法，分别是用户名，验证码，密码，再次密码，推荐人手机，注册按钮  
 def find\_username(self):  
 return self.driver.find\_element(\*self.username)  
  
 def find\_verify\_code(self):  
 return self.driver.find\_element(\*self.verify\_code2)  
  
 def find\_password(self):  
 return self.driver.find\_element(\*self.password)  
  
 def find\_password\_again(self):  
 return self.driver.find\_element(\*self.password\_again)  
  
 def find\_phone(self):  
 return self.driver.find\_element(\*self.phone)  
  
 def find\_post\_btn(self):  
 return self.driver.find\_element(\*self.post\_btn)

class PostHandle:  
 # 操作层  
 def \_\_init\_\_(self):  
 self.post\_page = PostPage()  
  
 # 实现操作方法，分别是用户名，验证码，密码，再次密码，推荐人手机，注册按钮  
 def input\_username(self, username):  
 self.post\_page.find\_username().send\_keys(username)  
  
 def input\_verify\_code(self, code):  
 self.post\_page.find\_verify\_code().send\_keys(code)  
  
 def input\_password(self, pwd):  
 self.post\_page.find\_password().send\_keys(pwd)  
  
 def input\_password\_again(self, pwd2):  
 self.post\_page.find\_password\_again().send\_keys(pwd2)  
  
 def input\_phone(self, phone):  
 self.post\_page.find\_phone().send\_keys(phone)  
  
 def click\_post\_btn(self):  
 self.post\_page.find\_post\_btn().click()

class PostProxy:  
 # 业务层  
  
 def \_\_init\_\_(self):  
 self.post\_handle = PostHandle()  
  
 # 实现完整业务  
 def login(self, username, code, password, password2, phone):  
 self.post\_handle.input\_username(username)  
 self.post\_handle.input\_verify\_code(code)  
 self.post\_handle.input\_password(password)  
 self.post\_handle.input\_password\_again(password2)  
 self.post\_handle.input\_phone(phone)  
 self.post\_handle.click\_post\_btn()

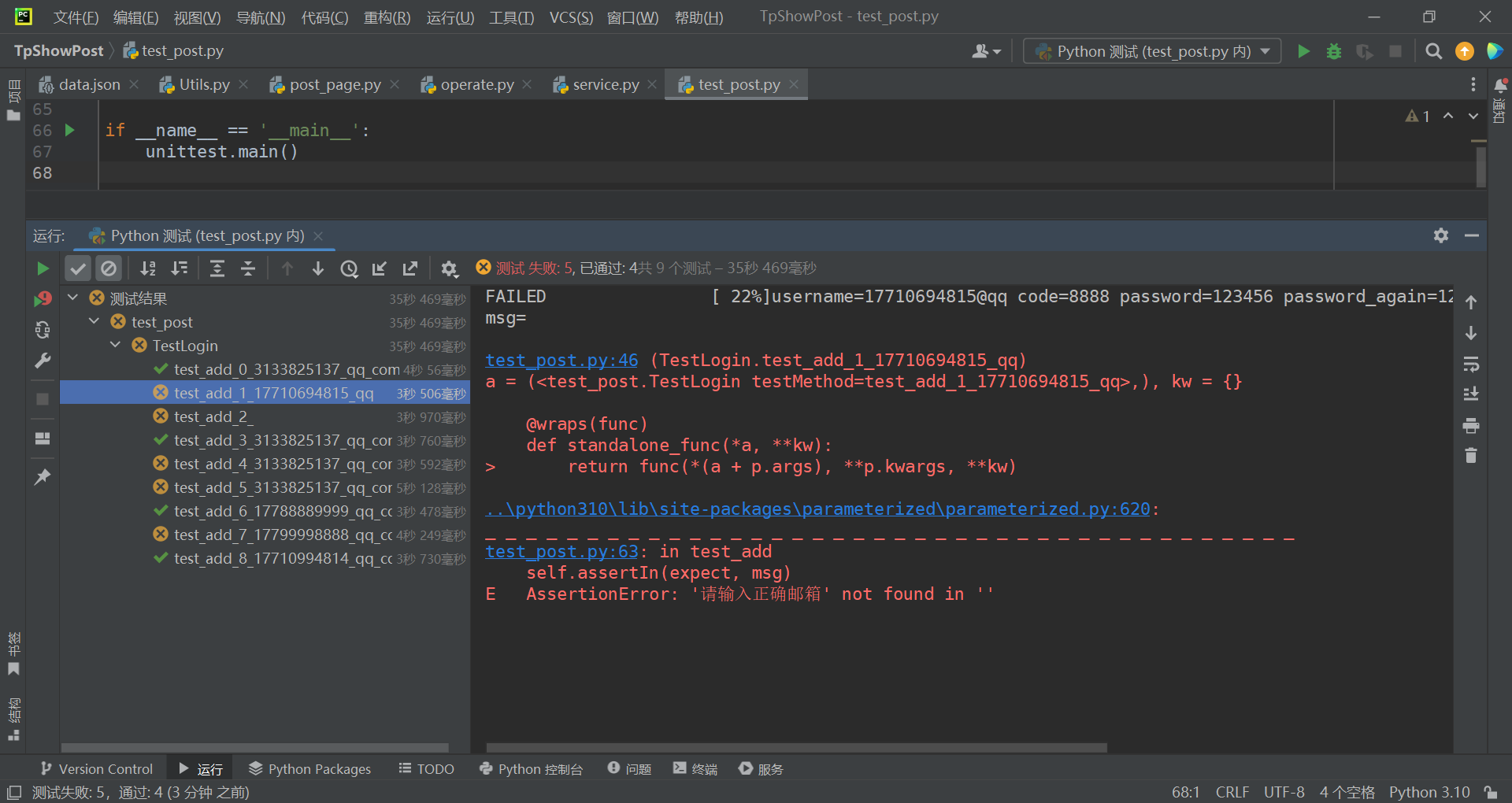
注册测试数据和测试层：

{  
 "post\_again": {  
 "username": "3133825137@qq.com",  
 "code": "8888",  
 "password": "123456",  
 "password\_again": "123456",  
 "phone": "",  
 "is\_success": false,  
 "expect": "账号已存在"  
 },  
 "post\_username\_error": {  
 "username": "17710694815@qq",  
 "code": "8888",  
 "password": "123456",  
 "password\_again": "123456",  
 "phone": "",  
 "is\_success": false,  
 "expect": "请输入正确邮箱"  
 },  
 "post\_username\_empty": {  
 "username": "",  
 "code": "",  
 "password": "",  
 "password\_again": "",  
 "phone": "",  
 "is\_success": false,  
 "expect": "请先输入用户名或密码"  
 },  
 "post\_code\_error": {  
 "username": "3133825137@qq.com",  
 "code": "888",  
 "password": "123456",  
 "password\_again": "123456",  
 "phone": "",  
 "is\_success": false,  
 "expect": "图像验证码错误"  
 },  
 "post\_code\_empty": {  
 "username": "3133825137@qq.com",  
 "code": "",  
 "password": "123456",  
 "password\_again": "123456",  
 "phone": "",  
 "is\_success": false,  
 "expect": "图像验证码不能为空"  
 },  
 "post\_password\_empty": {  
 "username": "3133825137@qq.com",  
 "code": "8888",  
 "password": "",  
 "password\_again": "",  
 "phone": "",  
 "is\_success": false,  
 "expect": "请输入用户名或密码"  
 },  
 "post\_password\_again\_error": {  
 "username": "17788889999@qq.com",  
 "code": "8888",  
 "password": "123456",  
 "password\_again": "12345678",  
 "phone": "",  
 "is\_success": false,  
 "expect": "两次密码不一致"  
 },  
 "post\_password\_error": {  
 "username": "17799998888@qq.com",  
 "code": "8888",  
 "password": "123",  
 "password\_again": "",  
 "phone": "",  
 "is\_success": false,  
 "expect": "密码有效长度为6-16位,并且不得有空格"  
 },  
  
 "post\_success\_second": {  
 "username": "17710994814@qq.com",  
 "code": "8888",  
 "password": "123456",  
 "password\_again": "123456",  
 "phone": "",  
 "is\_success": true,  
 "expect": "注册成功"  
 }  
}

测试层

# 获取数据并存储  
def build\_data():  
 test\_data = []  
 with open('./data.json', encoding='UTF-8') as f:  
 json\_data = json.load(f)  
 for post\_data in json\_data.values():  
 test\_data.append((  
 post\_data.get("username"),  
 post\_data.get("code"),  
 post\_data.get("password"),  
 post\_data.get("password\_again"),  
 post\_data.get("is\_success"),  
 post\_data.get("phone"),  
 post\_data.get('expect')  
 ))  
 print("test\_data", test\_data)  
 return test\_data  
  
  
class TestLogin(unittest.TestCase):  
 @classmethod  
 def setUpClass(cls):  
 cls.driver = DriverUtil.get\_driver()  
 cls.post\_proxy = PostProxy()  
  
 @classmethod  
 def tearDownClass(cls):  
 DriverUtil.quit\_driver()  
  
 def setUp(self):  
 # 进入首页，点击登陆连接  
 time.sleep(1)  
 self.driver.get("http://hmshop-test.itheima.net/")  
 self.driver.find\_element\_by\_link\_text("注册").click()  
 self.driver.find\_element\_by\_link\_text('邮箱注册').click()  
  
 @parameterized.expand(build\_data)  
 def test\_add(self, username, code, password, password\_again, phone, is\_success, expect):  
 print("username={} code={} password={} password\_again={} phone={} is\_success={} expect={}".format(  
 username, code, password, password\_again, phone, is\_success, expect  
 ))  
  
 # 注册  
 self.post\_proxy.login(username, code, password, password\_again, phone)  
 # 验证arg1是arg2的子串，不是则fail  
 if is\_success:  
 self.assertIn(expect, self.driver.find\_element\_by\_class\_name('userinfo').text)  
 self.driver.find\_element\_by\_link\_text("安全退出").click()  
 else:  
 # 获取提示消息  
 msg = Utils.get\_msg()  
 print("msg=", msg)  
 self.assertIn(expect, msg)  
  
  
if \_\_name\_\_ == '\_\_main\_\_':  
 unittest.main()

|  |  |
| --- | --- |
| 测试编号 | 测试结果 |
| B001 | 通过 |
| B002 | 通过 |
| B003 | 未通过 |
| B004 | 通过 |
| B005 | 未通过 |
| B006 | 未通过 |
| B007 | 通过 |
| B008 | 未通过 |
| B009 | 通过 |



# 4.搜索功能手动测试

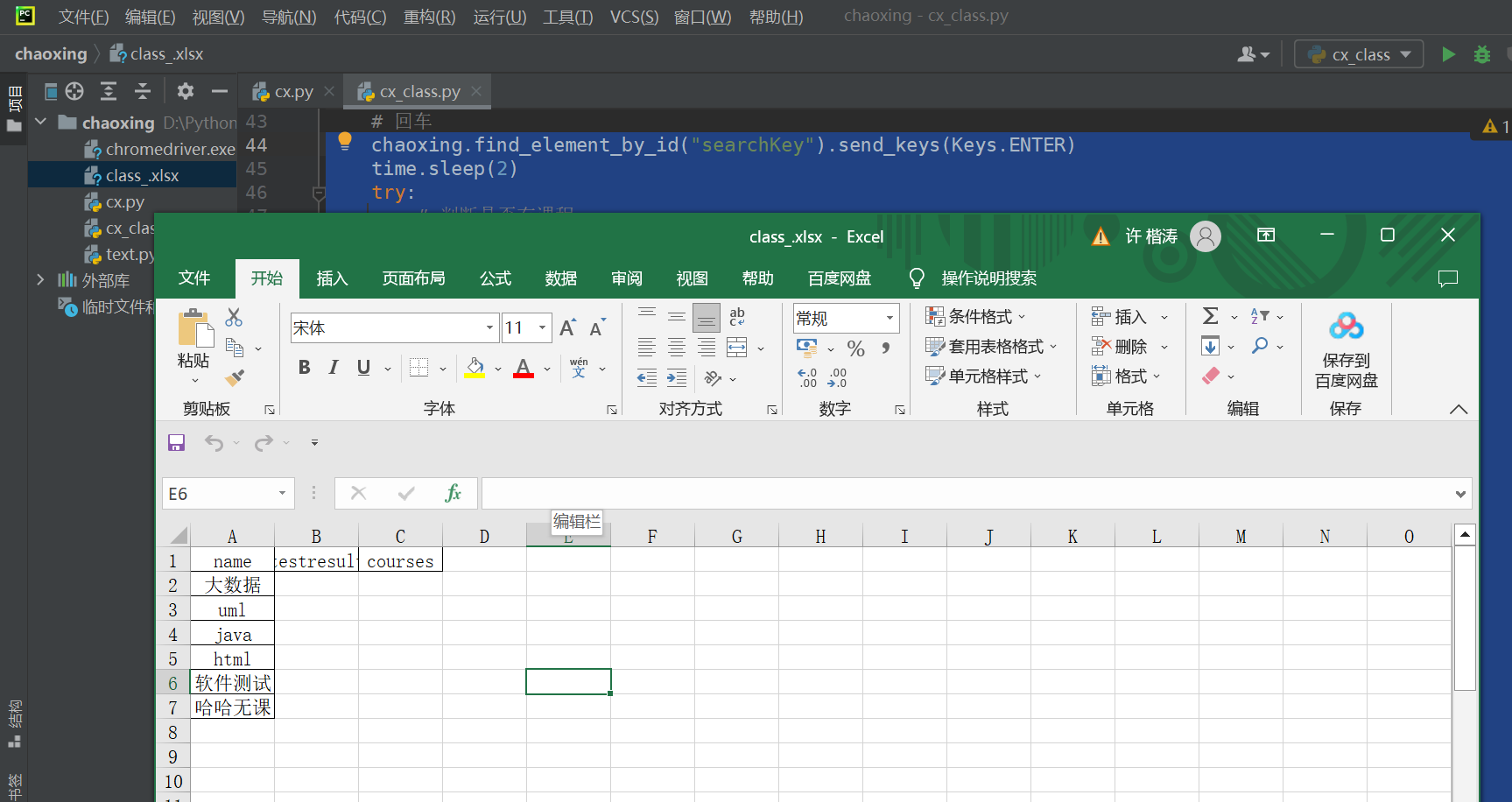
|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 用例编号 | 模块 | 测试目的 | 预置条件 | 输入 | 执行步骤 | 预期输出 | 实际输出 | 测试结果 |
| C001 | 搜索功能 | 支持汉字拼音单词输入, | 无 | Shouji,手ji,shou机 | 在搜索框进行输入 | 联想框出现手机字样 |  | 通过 |
| C002 | 搜索功能 | 错别字纠正输入 | 无 | 首机 | 在搜索框进行输入 | 根据后台词库匹配，联想框出现手机字样 |  | 通过 |
| C003 | 搜索功能 | 支持同音字搜索 | 无 | binxiang | 在搜索框进行输入 | 联想框出现冰箱字样 |  | 通过 |
| C004 | 搜索功能 | 支持数字同音 | 无 | 1，2 | 在搜索框进行输入 | 联想框出现1或者一的字样 |  | 未通过，输入一时未有联想匹配 |
| C005 | 搜索功能 | 支持标点符号释义 | 无 | 手机】 | 在搜索框进行输入 | 联想框出现手机的字样 |  | 未通过，未有联想匹配 |
| C006 | 搜索功能 | 自动补全 | 无 | 手机 | 在搜索框进行输入 | 联想框有品牌匹配，常用匹配，出现手机，iP手机壳等 |  | 未通过，  未能产生手机，手机壳联想匹配 |
| C007 | 搜索功能 | 匹配分类、标题、商品名称、品牌、厂商 | 无 | 华为 | 在搜索框进行输入 | 商品中出现华为手机，电脑，数据线等 |  | 通过 |
| C008 | 搜索功能 | 分词匹配 | 无 | 华为冰箱 | 在搜索框进行输入 | 商品中出现华为品牌和冰箱 |  | 未通过，无法进行分词搜索 |
| C009 | 搜索功能 | 匹配为空 | 无 | 无 | 无 | 推荐内容 |  | 未通过，未能出现推荐内容，直接跳转首页 |
| C010 | 搜索功能 | 销量、人气、收藏、价格、最新 | 无 | 手机 | 无 | 默认销量排序 |  | 未通过，默认为综合排序 |

# 5超星自动化测试

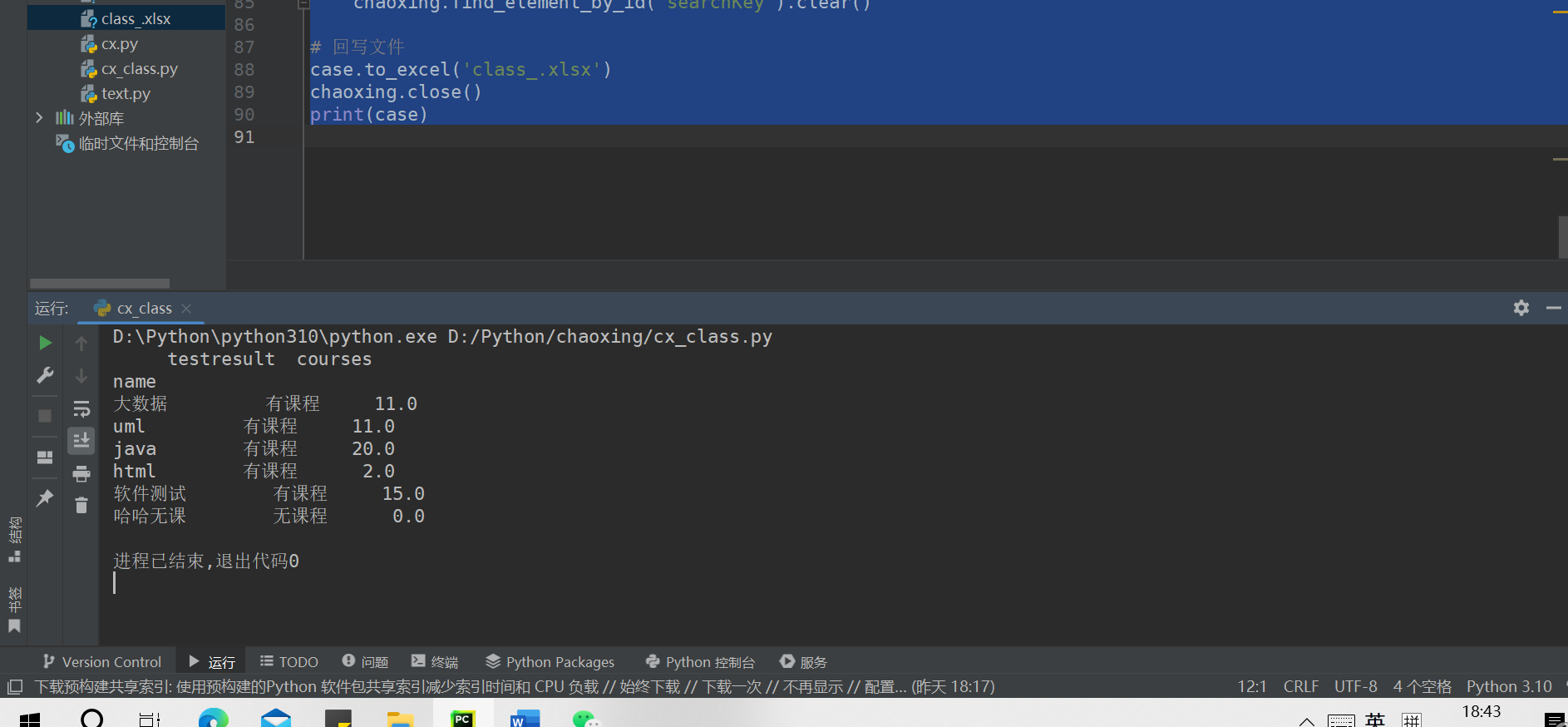
(1)循环选择下拉框的所有选项，其中全国课程关闭弹窗

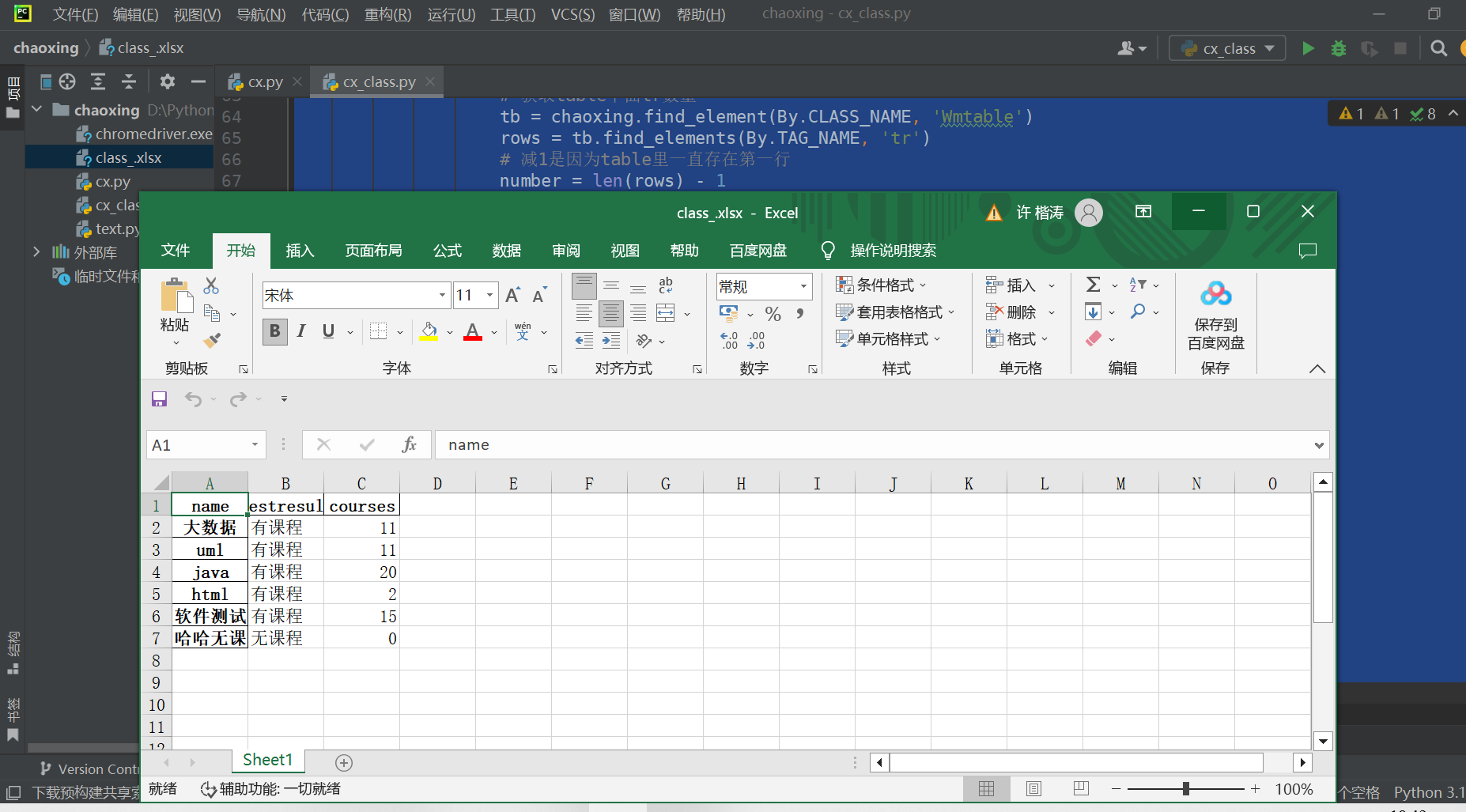
# 引入selenium包  
import time  
  
import selenium  
# 引入操作excel表的包  
# import xlrd  
import os  
# 引入pandas  
import pandas  
from selenium import webdriver  
from time import sleep  
from selenium.webdriver.support.select import Select  
  
# 打开谷歌浏览器，打开超星，  
chaoxing = webdriver.Chrome()  
chaoxing.get("http://gdcp.fanya.chaoxing.com/portal")  
sleep(2)  
chaoxing.maximize\_window()  
for i in range(1, 4):  
 time.sleep(2)  
 chaoxing.find\_element\_by\_class\_name('alreadychoice').click()  
 # 依次点击下拉框元素  
 chaoxing.find\_element\_by\_css\_selector('h4:nth-child(' + str(i) + ')').click()  
 chaoxing.find\_element\_by\_id("searchKey").send\_keys("软件测试技术")  
 chaoxing.find\_element\_by\_xpath('/html/body/div[5]/div/div[1]/input[3]').click()  
 time.sleep(2)  
 text = chaoxing.find\_element\_by\_css\_selector('h4:nth-child(' + str(i) + ')').text  
 if text == '全国课程':  
 alert = chaoxing.switch\_to.alert # 处理alert  
 alert.accept() # 确定弹窗  
 else:  
 # 回退页面  
 chaoxing.back()  
 time.sleep(1)  
 chaoxing.find\_element\_by\_id("searchKey").clear()

1. 测试5条不同关键字，并查找是否有课程，有多少条课程



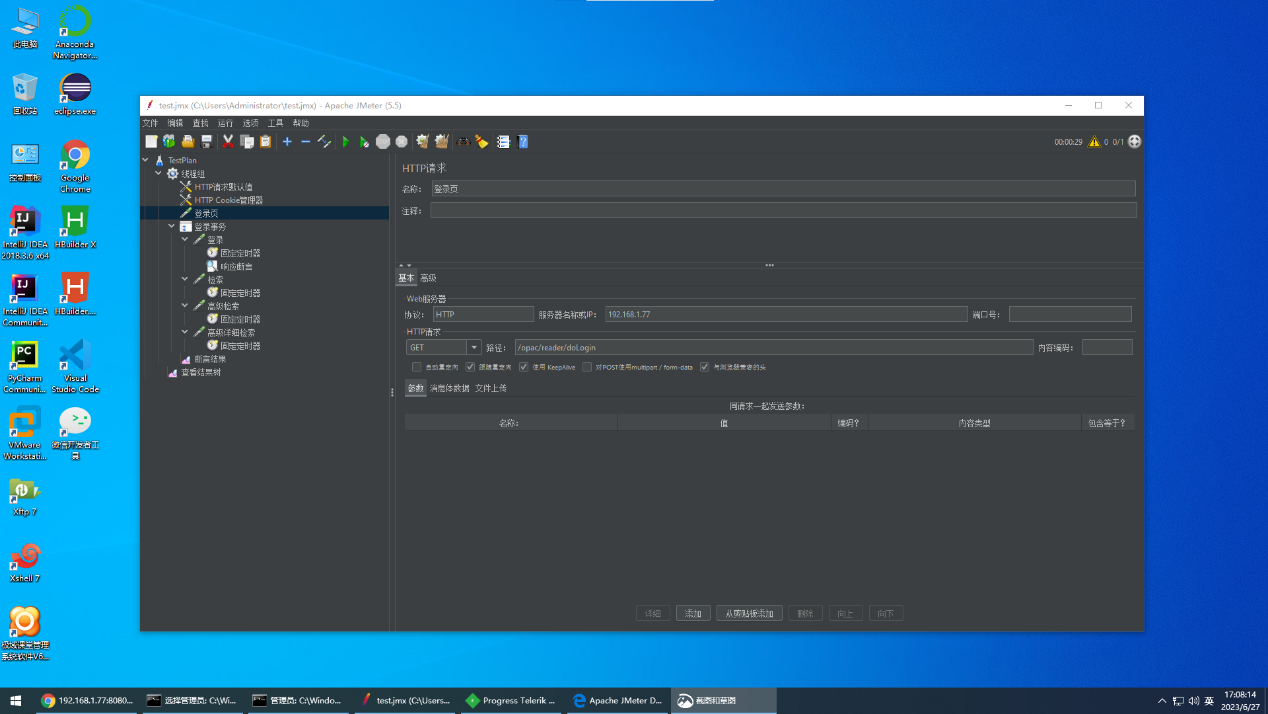
# 引入selenium包  
import time  
# 引入pandas  
import pandas as pd  
from selenium import webdriver  
from time import sleep  
from selenium.common.exceptions import NoSuchElementException  
from selenium.webdriver.common.by import By  
from selenium.webdriver.support.select import Select  
from selenium.webdriver.common.keys import Keys # 引用keys包  
  
chaoxing = webdriver.Chrome()  
chaoxing.get("http://gdcp.fanya.chaoxing.com/portal")  
sleep(2)  
chaoxing.maximize\_window()  
# 读取数据  
case = pd.read\_excel('class\_.xlsx', index\_col='name')  
data = case.index.tolist()  
# n表示case数据行索引  
n = 0  
havedata = "有课程"  
nodata = "无课程"  
  
  
# 用于判断是否超链接下一页元素是否存在，存在返回True，否则为False  
def get\_element(borwser, xpaths):  
 try:  
 borwser.find\_element\_by\_link\_text(xpaths)  
 return True  
 except NoSuchElementException:  
 return False  
  
  
# 循环搜索每个数据  
for i in data:  
 time.sleep(2)  
 # 点击下拉框  
 chaoxing.find\_element\_by\_class\_name('alreadychoice').click()  
 # 选择本校课程  
 chaoxing.find\_element\_by\_css\_selector('h4:nth-child(1)').click()  
 # 输入name搜索  
 chaoxing.find\_element\_by\_id("searchKey").send\_keys(i)  
 # 回车  
 chaoxing.find\_element\_by\_id("searchKey").send\_keys(Keys.ENTER)  
 time.sleep(2)  
 try:  
 # 判断是否有课程  
 if chaoxing.find\_element\_by\_xpath("/html/body/div[3]/div[3]/table/tbody/tr[2]/td[5]") != "":  
 time.sleep(2)  
 # 判断是否存在下一页  
 if get\_element(chaoxing, "下一页"):  
 sums = 0  
 # 先获取第一页  
 # 获取table下面tr数量  
 tb = chaoxing.find\_element(By.CLASS\_NAME, 'Wmtable')  
 rows = tb.find\_elements(By.TAG\_NAME, 'tr')  
 # 减1是因为table里一直存在第一行  
 number = len(rows) - 1  
 sums = sums + number  
 # 因为当下一页无法按时，则变成span元素不是超链接a元素，所以进行循环，是超链接则继续，否则退出  
 while get\_element(chaoxing,'下一页'):  
 chaoxing.find\_element\_by\_link\_text('下一页').click()  
 # 获取table下面tr数量  
 tb = chaoxing.find\_element(By.CLASS\_NAME, 'Wmtable')  
 rows = tb.find\_elements(By.TAG\_NAME, 'tr')  
 # 减1是因为table里一直存在第一行  
 number = len(rows) - 1  
 sums = sums + number  
 time.sleep(1)  
  
 else:  
 tb = chaoxing.find\_element(By.CLASS\_NAME, 'Wmtable')  
 rows = tb.find\_elements(By.TAG\_NAME, 'tr')  
 sums = len(rows) - 1  
 # 对case的DataFrame进行赋值  
 case.iloc[n, 0] = havedata  
 case.iloc[n, 1] = sums  
 n = n + 1  
 # 找不到课程时，  
 except NoSuchElementException:  
 case.iloc[n, 0] = nodata  
 case.iloc[n, 1] = 0  
 n = n + 1  
 # 清空输入框  
 chaoxing.find\_element\_by\_id("searchKey").clear()  
  
# 回写文件  
case.to\_excel('class\_.xlsx')  
chaoxing.close()  
print(case)

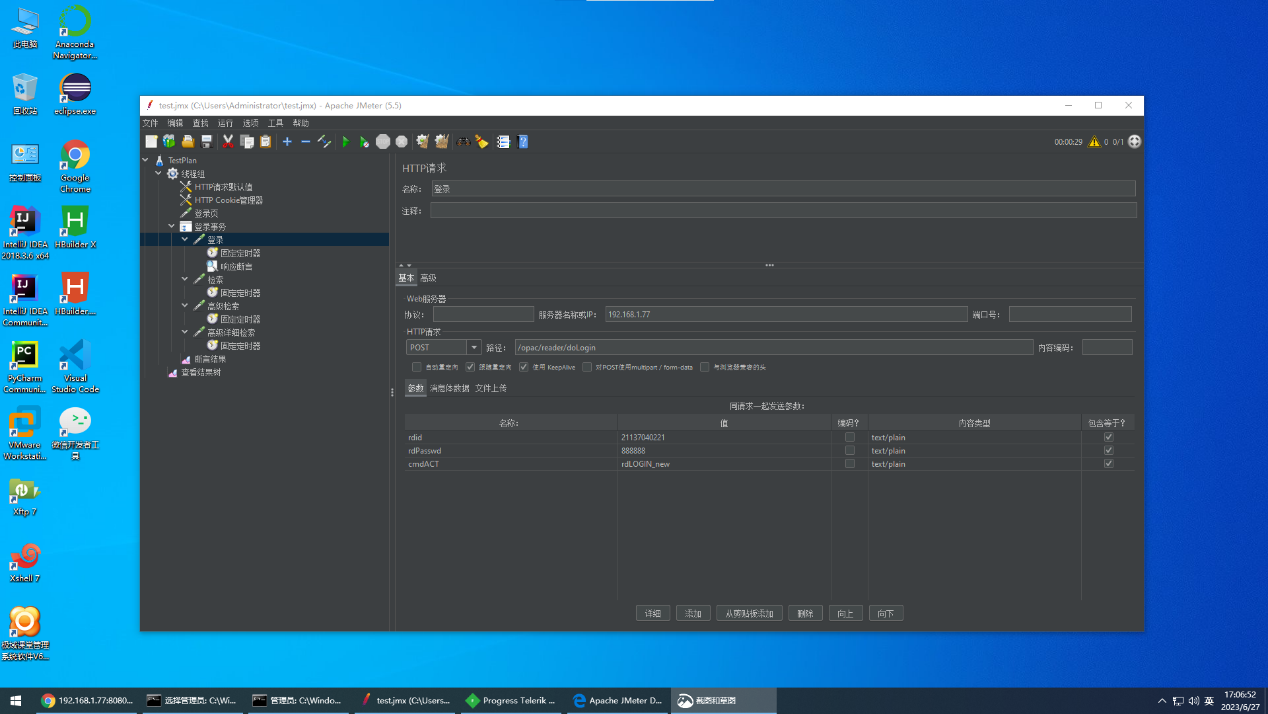




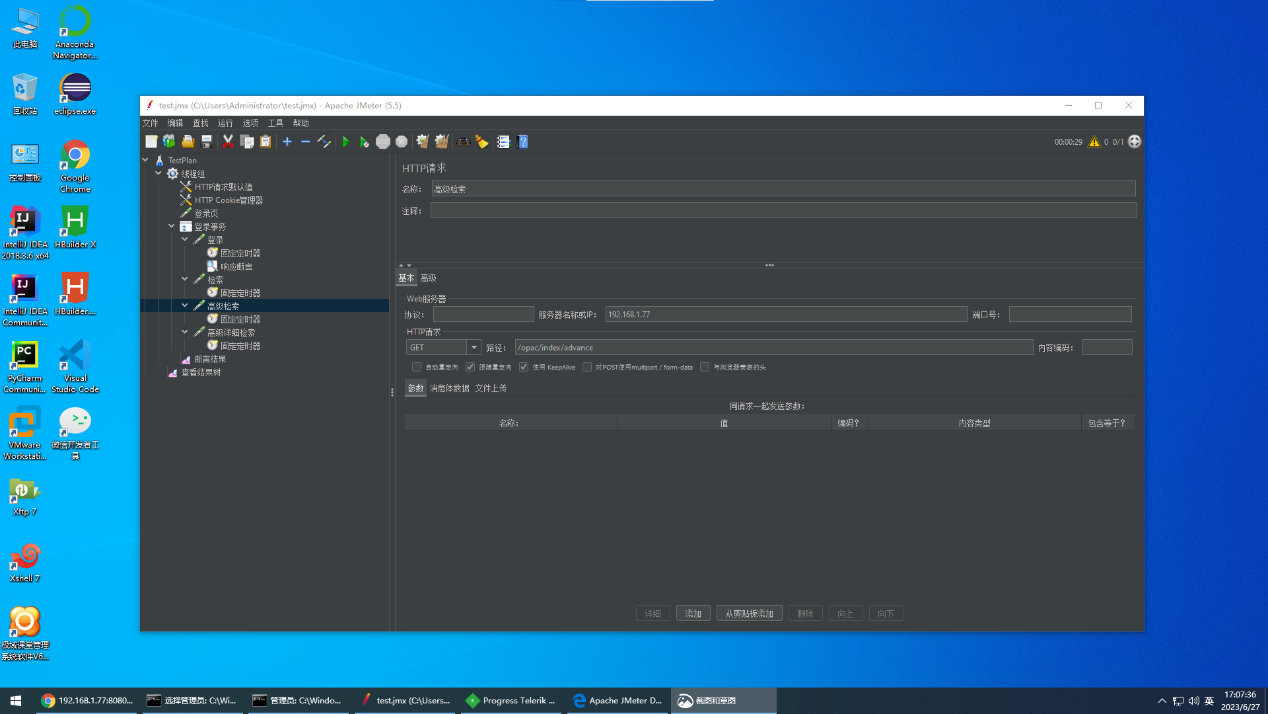
# 7.性能测试设计与执行

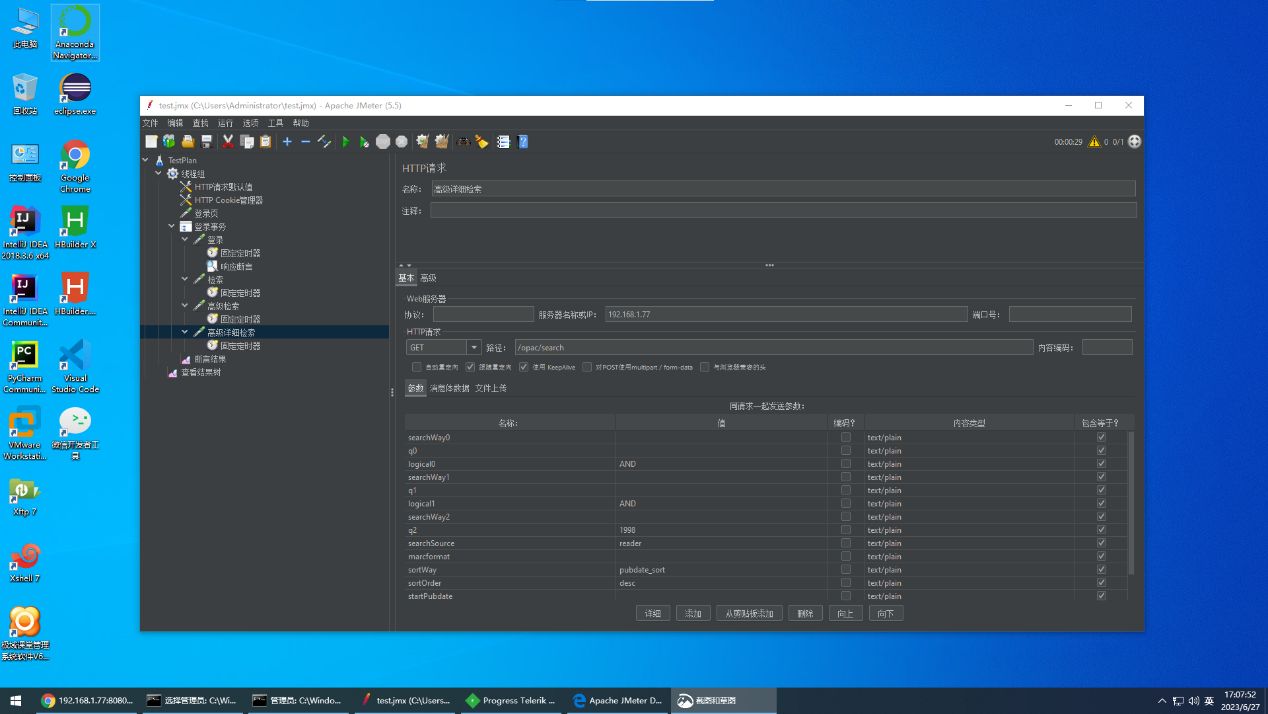
步骤：只选择了单用户，进入登录页>登陆>馆藏查询的检索/高级检索

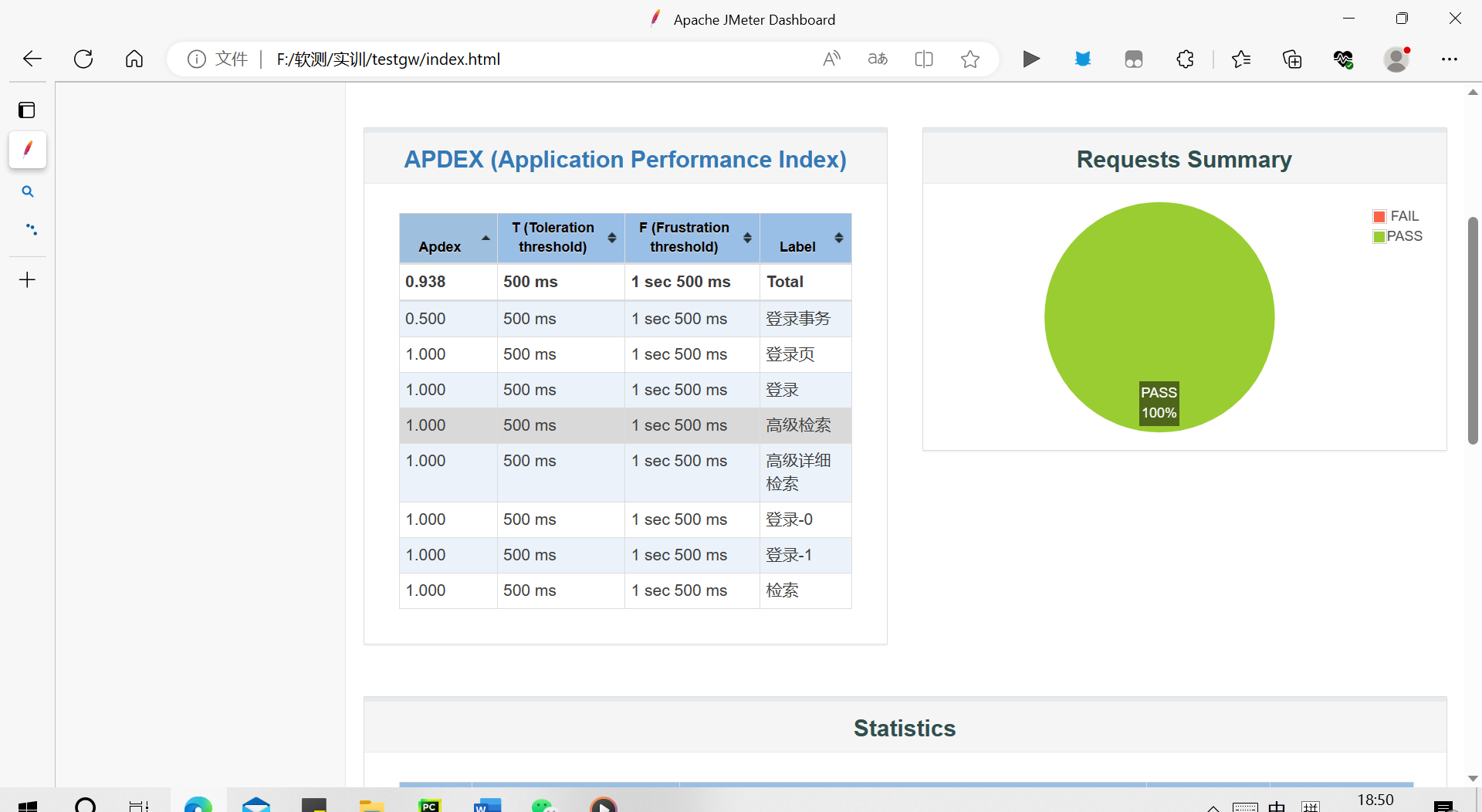


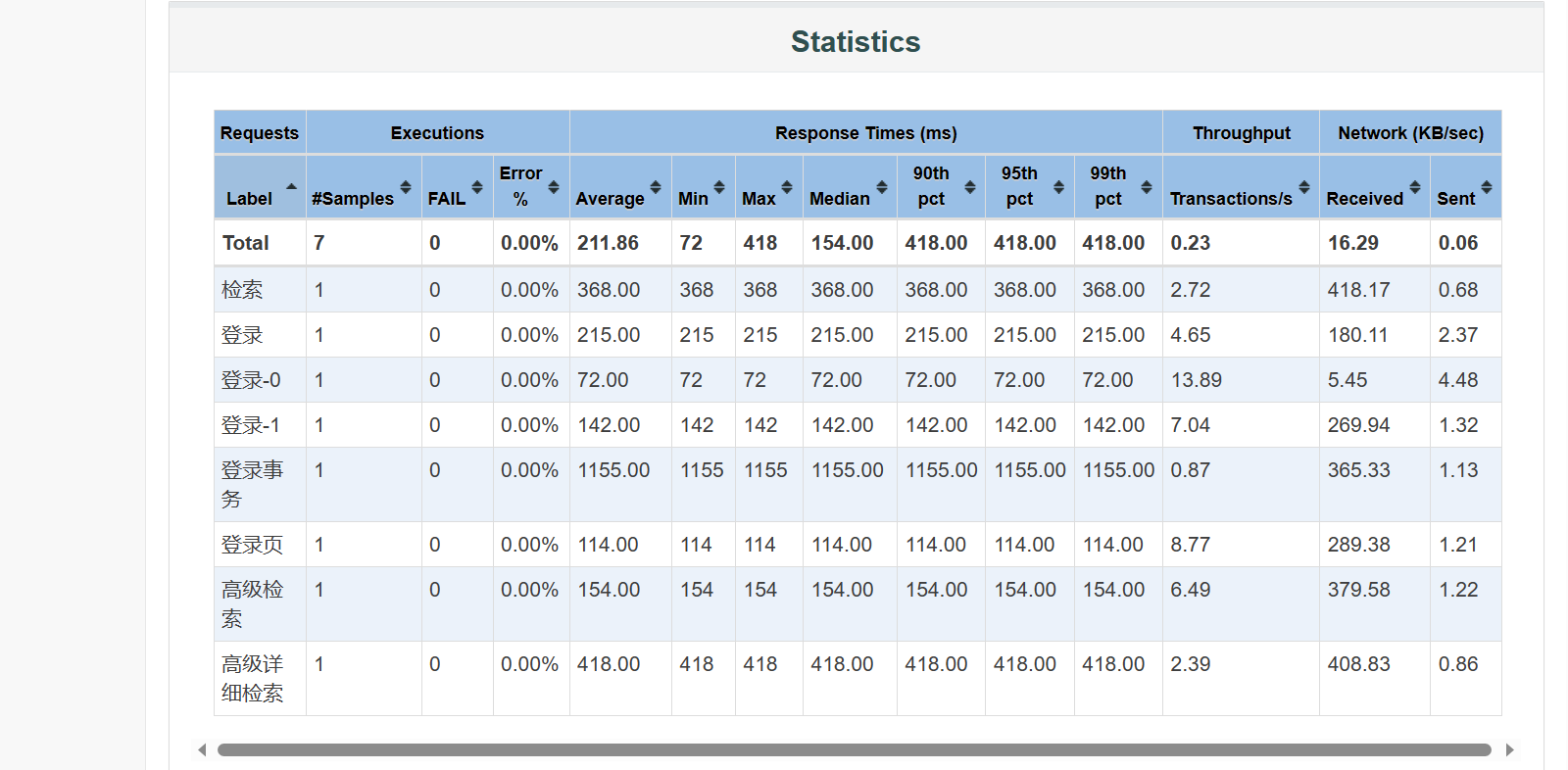








概要图：

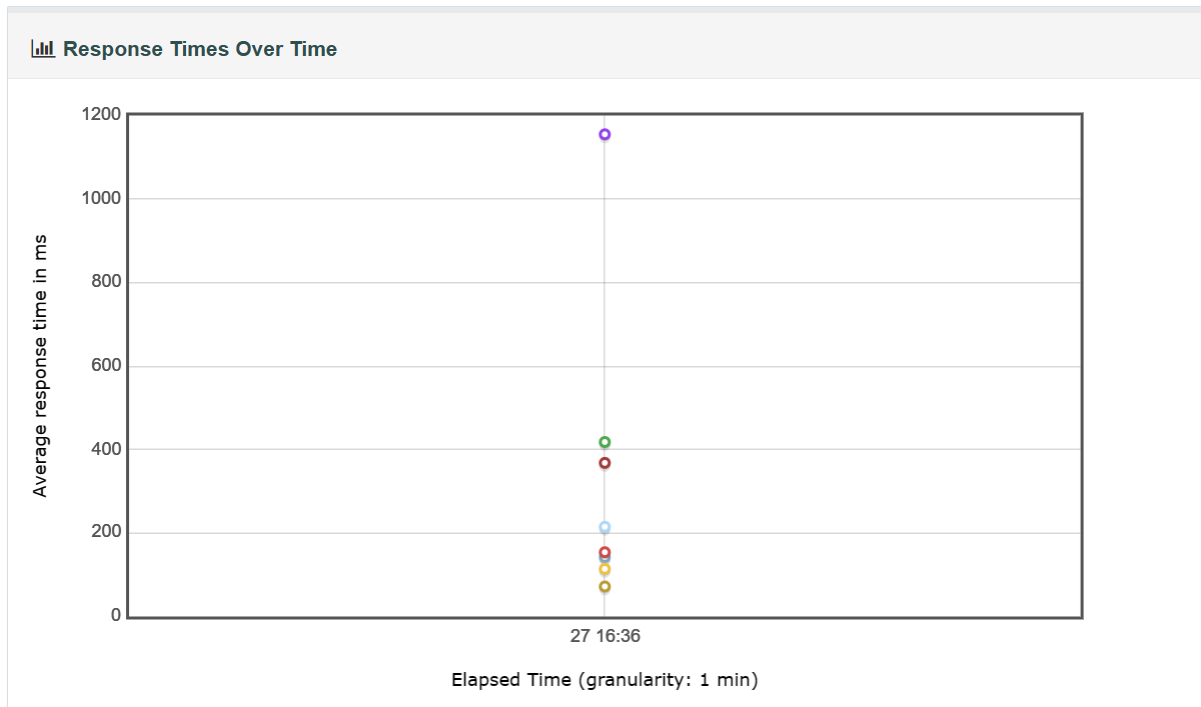
事务图：

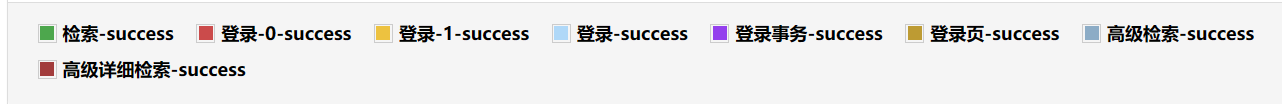
Response Times的Average为平均响应时间,

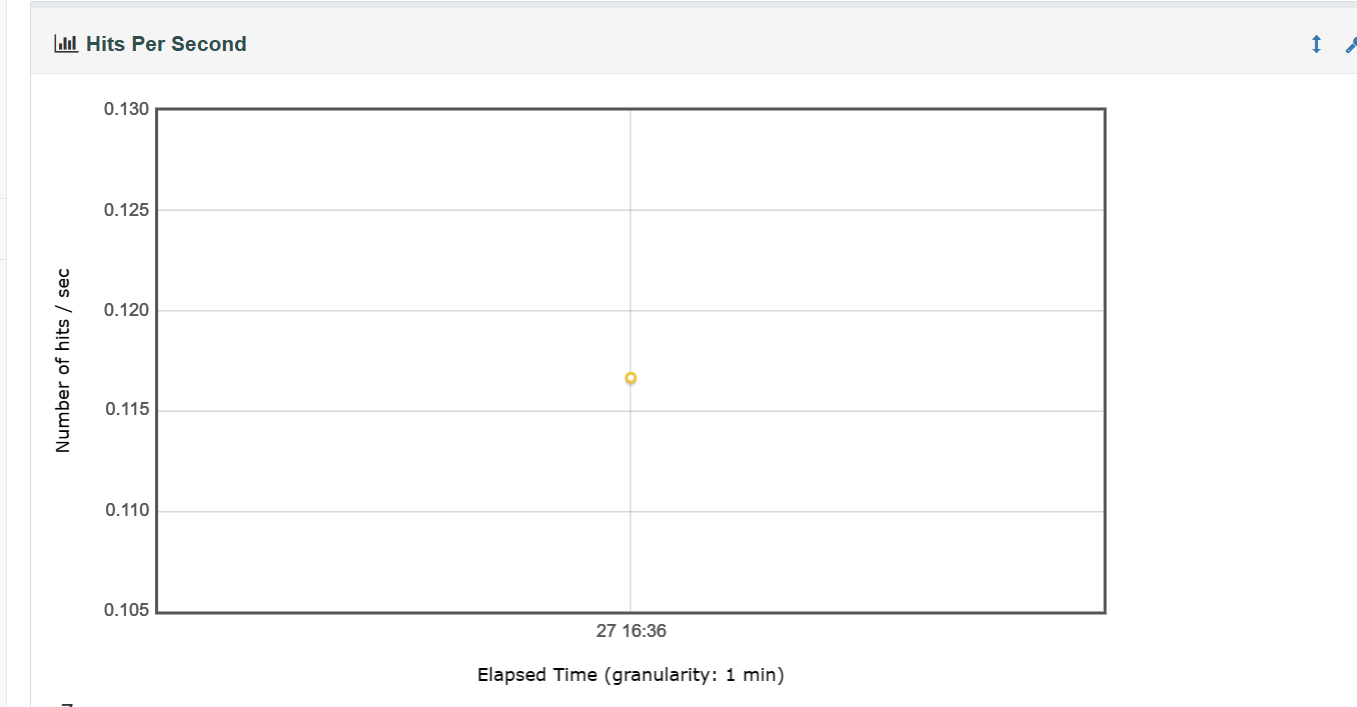
Throughput为吞吐量，每秒吞吐数量

Nekwork下的分别是发送数量和接受数量

主要是每个样本的平均响应时间，单位为ms；





资源图：

每秒的一个点击率

# 8.缺陷报告

5.1 测试缺陷等级定义

（1）软件功能未实现且影响安全使用，缺陷等级定义为：高；

（2）软件功能未实现且不影响安全使用，缺陷等级定义为：中；

（3）软件功能实现后有错别字但是不影响用户理解，缺陷等级定义为：低。

5.2 缺陷报告

采用以下模板报告资产管理系统中发现的错误。

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **模块名称** | **页面/窗口** | **摘要** | **描述** | **缺陷严重程度** |
| 登录 | 登陆页面 | 进入登录界面，账号为空，点击登录 | 浏览器：谷歌 版本号：86.0.4240.111 1、进入登陆界面 2、点击登录 预期结果：出现信息提示“账号不能为空” 出现结果：未提示 | 高 |
| 注册 | 注册页面 | 进入注册页面，不输入手机号/邮箱，或者输入错误的手机号/邮箱，点击注册 | 浏览器：谷歌 版本号：86.0.4240.111 1、进入注册页面 2、点击注册，观察界面 预期结果：出现弹出信息提示“请用手机号或邮箱注册” 出现结果：未提示 | 高 |
| 注册 | 注册页面 | 进入注册页面，不输入验证码，点击注册 | 浏览器：谷歌 版本号：86.0.4240.111 1、进入注册页面 2、点击注册，观察界面 预期结果：出现弹出信息提示图形验证码不能为空 出现结果：未提示 | 高 |
| 注册 | 注册页面 | 进入注册页面，未输入密码，点击注册按钮 | 浏览器：谷歌 版本号：86.0.4240.111 1、进入注册页面 2、点击注册，观察界面 预期结果：出现弹出信息提示“请输入用户名或密码”  出现结果：未提示 | 高 |
| 注册 | 注册页面 | 进入注册页面，密码输入不符合要求，点击注册 | 浏览器：谷歌 版本号：86.0.4240.111 1、进入注册页面 2、id框输入小数，其他输入正确 预期结果：出现弹出信息提示“密码为6-16位大小写英文字符、数字或符号” 出现结果：未提示 | 高 |
| 搜索 | 首页页面 | 点击搜索框，输入数字同音 | 浏览器：谷歌 版本号：86.0.4240.111 1、点击搜索框 2、输入内容，观察界面 预期结果：联想框出现例如1或者一的同音字样  出现结果：未有联想匹配 | 中 |
| 搜索 | 首页页面 | 点击搜索框，输入标点符号释义 | 浏览器：谷歌 版本号：86.0.4240.111 1、点击搜索框 2、输入内容，观察界面 预期结果：联想框出现标点符号释义的字样  出现结果：未有联想匹配 | 中 |
| 搜索 | 首页页面 | 点击搜索框，输入内容自动补全 | 浏览器：谷歌 版本号：86.0.4240.111 1、点击搜索框 2、输入内容，观察界面 预期结果：联想框有品牌匹配，常用匹配等  出现结果：未能产生匹配 | 中 |
| 搜索 | 首页页面 | 点击搜索框，输入时有分词匹配 | 浏览器：谷歌 版本号：86.0.4240.111 1、点击搜索框 2、输入内容，观察界面 预期结果：商品中出现分词匹配  出现结果：未能实现，未进行分词搜索 | 中 |
| 搜索 | 首页页面 | 点击搜索框，输入匹配为空，点击搜索按钮 | 浏览器：谷歌 版本号：86.0.4240.111 1、点击搜索框 2、点击搜索，观察界面 预期结果：出现推荐内容  出现结果：未能出现推荐内容，直接跳转首页 | 中 |
| 搜索 | 搜索页面 | 点击搜索框，输入内容，点击搜索按钮 | 浏览器：谷歌 版本号：86.0.4240.111 1、点击搜索框 2、输入内容，点击搜索，观察界面 预期结果：默认销量排序  出现结果：默认为综合排序 | 中 |

# 9.测试总结

## 6.1测试环境与配置

|  |  |
| --- | --- |
| 资源名称/类型 | 配置 |
| 操作系统 | Window10 x64位 |
| 浏览器 | 谷歌浏览器 |
| 测试工具 | 性能测试(Jmeter) |
| 自动化测试(使用python，使用selenium，UnitTest框架，数据驱动，Pandas) |
|  | 手动测试 |

## 6.2 测试方法

【注意下面的\*\*要结合本组内容，替换成本组完成的手工测试模块和自动化测试内容的介绍】

* 软件审查法

软件审查法是根据测试过程中的各个阶段进行审查的测试方法，主要检查测试需求、概要信息、详细设计等方面产生的文档、代码、测试用例、测试方案，检查文档内容的一致性、完整性、准确性。UI界面测试

确保UI界面符合产品经理和设计师的需求，并且文案正确。

* 黑盒测试

黑盒测试又称为功能测试，黑盒测试只关注系统外部结构，不关注系统内部逻辑，针对系统功能需求和界面需求进行测试。

* 人工测试

本次测试中搜索功能采用人工测试。

* 自动化测试

本次测试中登陆以及注册采用自动化功能测试。

## 6.3 用例汇总

|  |  |  |  |
| --- | --- | --- | --- |
| **功能模块** | **测试用例总数** | **用例编写人** | **执行人（** |
| 登录 | 6 | 许楷涛，阮子豪 | 许楷涛，阮子豪 |
| 注册 | 9 | 许楷涛，阮子豪 | 阮子豪 |
| 搜索功能 | 10 | 阮子豪 | 阮子豪 |
| 用例合计（个） | 25 |  |  |

## 6.4 缺陷分析

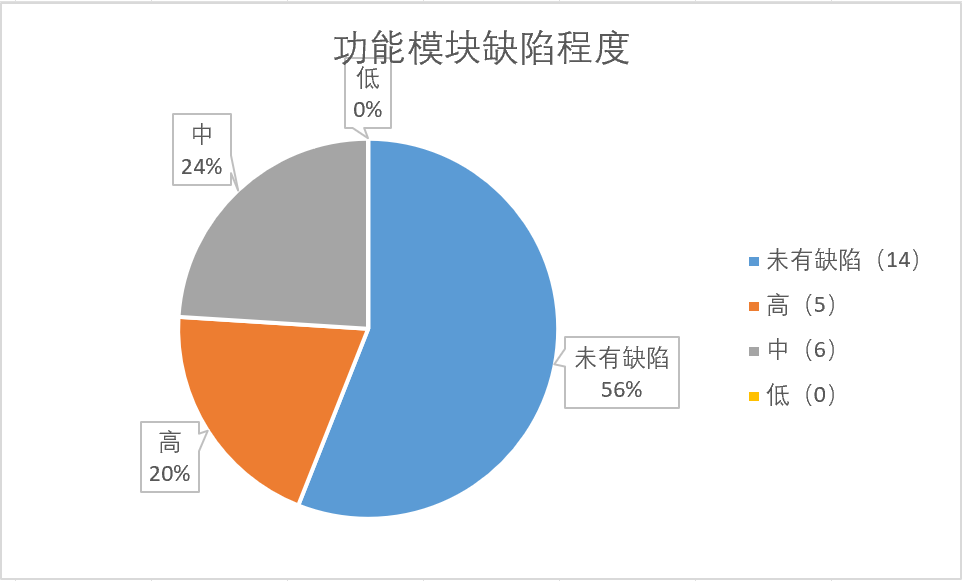
|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **功能模块** | **按BUG严重程度个数** | | | | **缺陷类型** | | | | |
| 高 | 中 | 低 | 合计 | 功能缺陷 | UI缺陷 | 兼容性缺陷 | 建议性缺陷 | 合计 |
| 登录 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 1 |
| 注册 | 4 | 0 | 0 | 4 | 4 | 0 | 0 | 0 | 4 |
| 搜索功能 | 0 | 6 | 0 | 6 | 0 | 0 | 0 | 6 | 6 |
| 合计（个） | 5 | 6 | 0 | 11 | 4 | 0 | 0 | 7 | 11 |

## 6.5测试结论（或对系统的建议）

【注意下面的\*\*要结合本组内容，替换成本组完成的真实内容和数据】

1. 整体测试总结：本测试任务由01测试小组完成，总耗时3天，编写了25条测试用例，缺陷有11条，其中缺陷程度为高的有5条，缺陷程度为中的有6条。本次测试的基本流程符合规范。

为直观表达，以上数据已转成饼图如下。



1. 分析：本系统在注册、登录功能上未能达到要求，在搜索功能上存在功能未实现和功能缺陷，，未达到交付用户使用的标准。
2. 建议：修复系统中存在的缺陷并由二次测试验证通过后才交付用户使用。

# 8.个人实训周心得体会

许楷涛：经过3天的软件测试实训，从对软件测试没有什么经验的我初步掌握了软件测试的方法和技能，收获颇多。在本次实训中，我回顾这学期所学的知识点，复习了各种测试方法。还在软件安装方面了解到各个安装版本的区别。在这次实训问题还是有的，但经过这些问题后解决了，我也学到了这些知识。

阮子豪：通过此次实训，我对整个软件测试过程有了大大的加深。在这次实训中，我最大的收获是对合作的理解，团队的重要性，发现在合作伙伴的共同努力下，我们解决了一个又一个的问题，而且在期间也学习到合作伙伴的许多知识，知道了自己的不足，懂得了学习的方法。我觉得在以后的学习之旅中，这是非常重要的。