

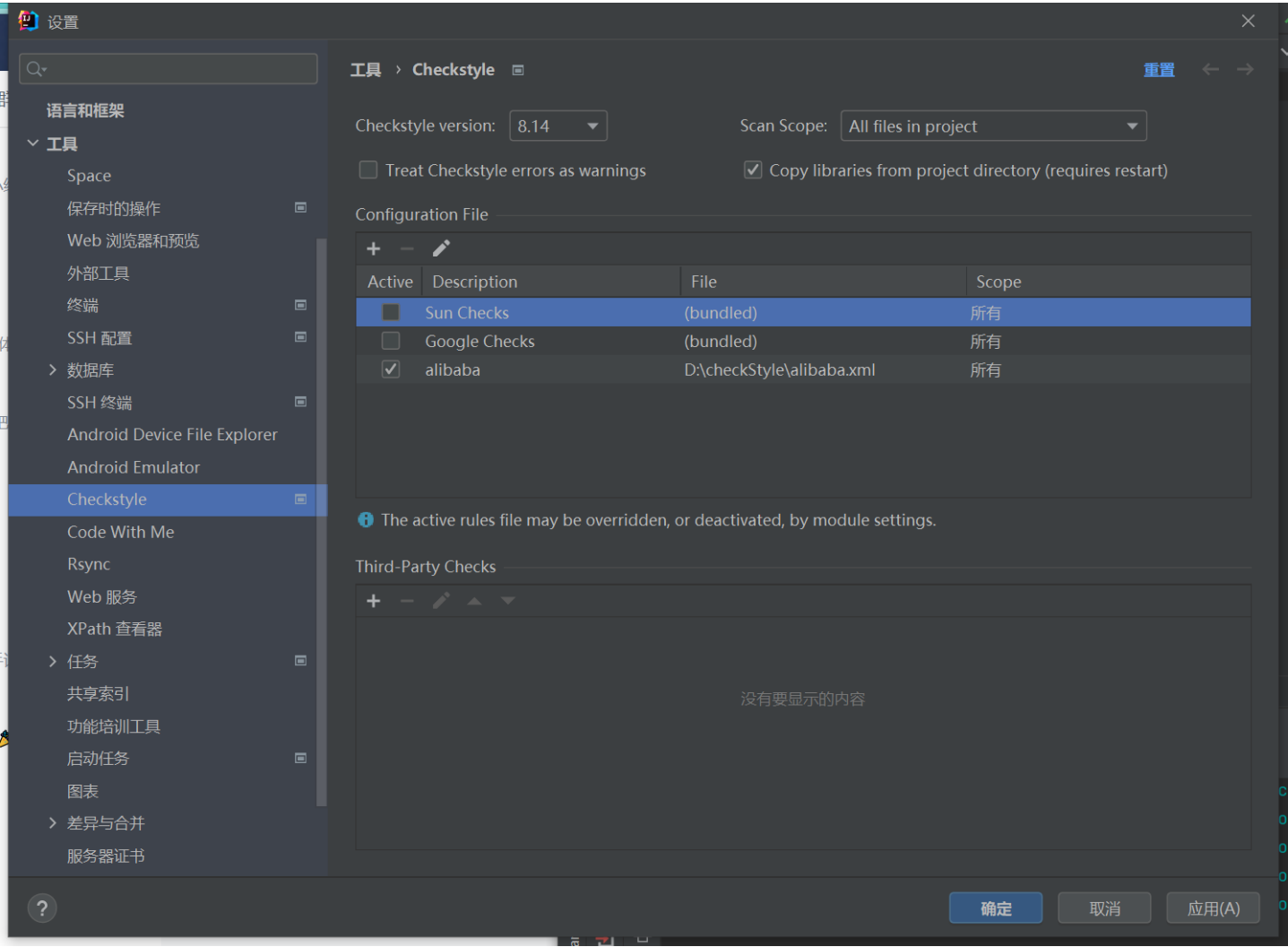
# 软件测试文档

## 单元测试

### 1.静态代码分析：

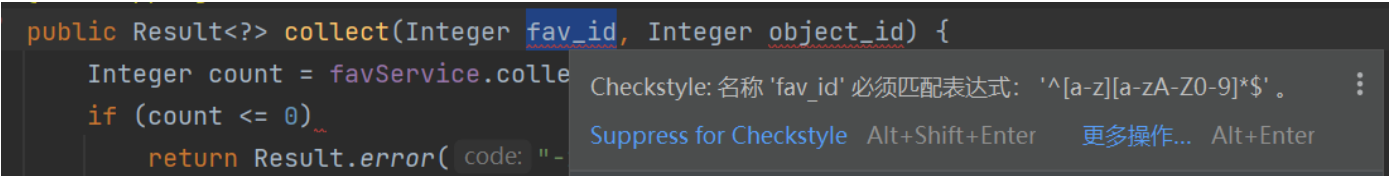
#### 后端部分：

在我们的项目当中，采用了checkstyle工具来进行代码规范的检查，我们导入了阿里巴巴的代码规范



下面将展示我们项目当中出现的一些代码问题以及改善结果。

### 1. 命名规范问题



- 解决办法：将命名改为小驼峰命名法
- 处理结果：

```
public Result<?> collect(Integer favId, Integer objectId) {
```

## 2. 存在部分if语句未加括号的情况

```
public Result<?> collect(Integer favId, Integer objectId) {  
    Integer count = favService.collect(favId, objectId);  
    if (count <= 0)  
        return Result.error( code: "-1", msg: "收藏失败");  
    else return Result.success();  
}
```

- 解决办法：添加括号
- 处理结果：

```
    if (count <= 0) {  
        return Result.error( code: "-1", msg: "收藏失败");  
    } else {  
        return Result.success();  
    }  
}
```

## 3. 存在无用导入

```
import com.example.demo.service.impl.ObjectServiceImpl;  
import io.swagger.models.auth.In;  
import o Checkstyle: 无用导入 - io.swagger.models.auth.In 。  
import o Suppress for Checkstyle Alt+Shift+Enter 更多操作... Alt+Enter  
import j 未找到任何文档。
```

- 解决办法：删除无效导入

## 4. Return 次数超过了3次

```
@GetMapping("cancelOrder")
```

```
pub
```

Checkstyle: Return 次数 4 次 (最大允许非空虚方法/ 拉姆达: 3 次)。

Suppress for Checkstyle Alt+Shift+Enter 更多操作... Alt+Enter

```
@Target({ElementType.METHOD})
```

```
@Retention(RetentionPolicy.RUNTIME)
```

```
@Documented
```

```
@RequestMapping(method = {RequestMethod.GET})
```

```
public @interface GetMapping
```

```
extends annotation.Annotation
```

Maven: org.springframework:spring-web:5.3.13 (spring-web-5.3.13.jar)

- 解决办法：减少方法中的返回次数

## 5. 单行字数超过了140个

```
int count = orderService.pay(orderId);
```

```
String payUrl = "http://101.35.194.132:9090/alipay/pay?subject=" + objectService.getById((orderService.getById(orderId).getObjectId()));
```

- 解决办法：将单行代码分成两行

## 6. 在项目当中使用了System.out.print语句

```
// 支付宝验签
```

```
if (Factory.Payment.Common().verifyNotify(params)) {
```

```
    // 验签通过
```

```
    System.out.println("交易名称: " + params.get("subject"));
```

```
    System.out.println("交易状态: " + params.get("trade_status"));
```

```
    System.out.println("支付宝交易凭证号: " + params.get("trade_no"));
```

```
    System.out.println("商户订单号: " + params.get("out_trade_no"));
```

```
    System.out.println("交易金额: " + params.get("total_amount"));
```

```
    System.out.println("买家在支付宝唯一id: " + params.get("buyer_id"));
```

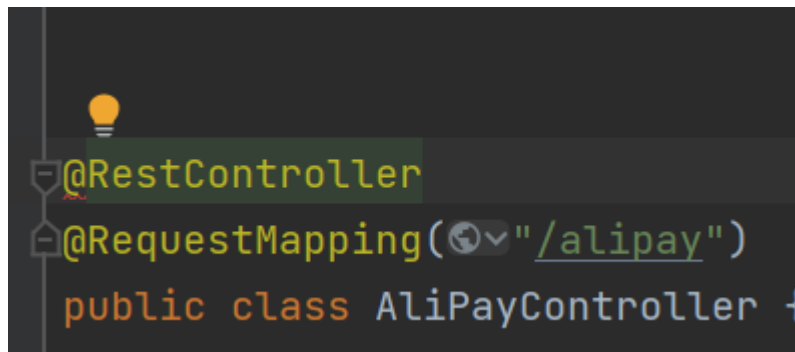
```
    System.out.println("买家付款时间: " + params.get("gmt_payment"));
```

```
    System.out.println("买家付款金额: " + params.get("buyer_pay_amount"));
```

使用system.out.print会大幅度降低后台的性能，所以在项目中尽量不要使用system.out.print

- 解决办法，将语句输出到日志当中，并且删除这些语句。

## 7. 缺乏javadoc注释



```

@RestController
@RequestMapping("/alipay")
public class AliPayController {

```

- 解决办法：添加javadoc



```

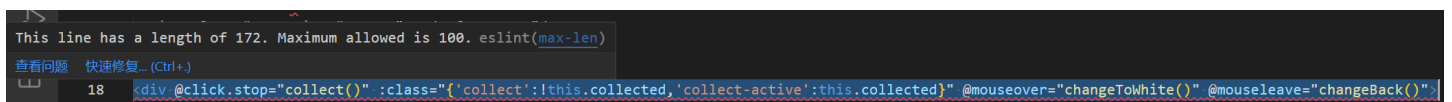
/**
 *
 * @author cj
 * @date 2022/6/11 20:41
 * @param
 */
@RestController
@RequestMapping("/alipay")

```

## 前端部分：

我们使用了ESLint 作为语法规则和代码风格的检查工具，采用了最为严格的airbnb规范，检测出错误后部分错误可以使用ESLint自动修复错误

### 1.单行字数超过100个



```

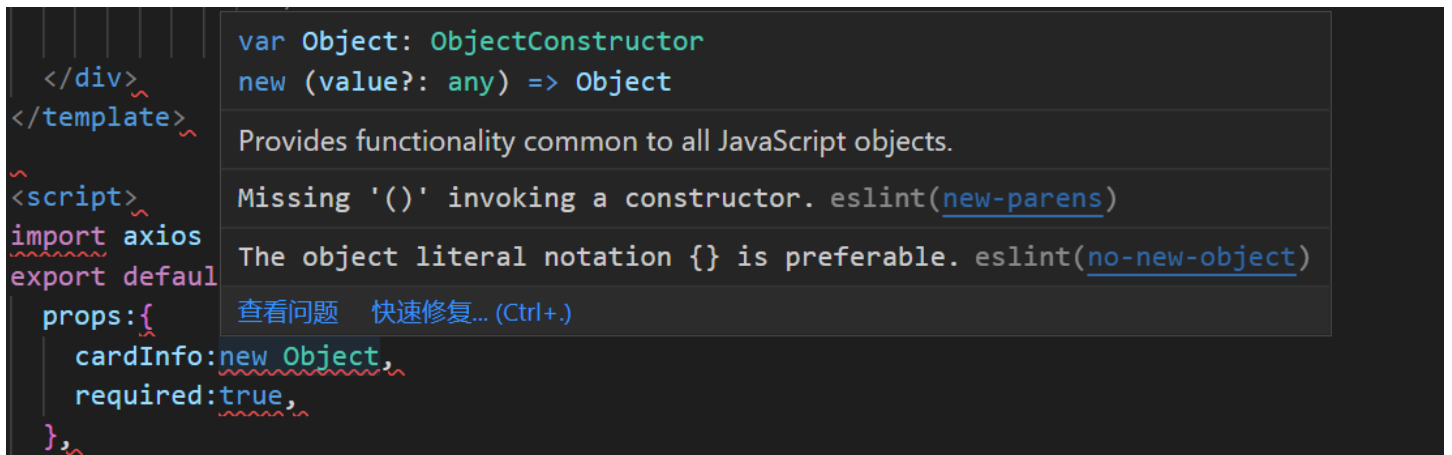
This line has a length of 172. Maximum allowed is 100. eslint(max-len)


>


```

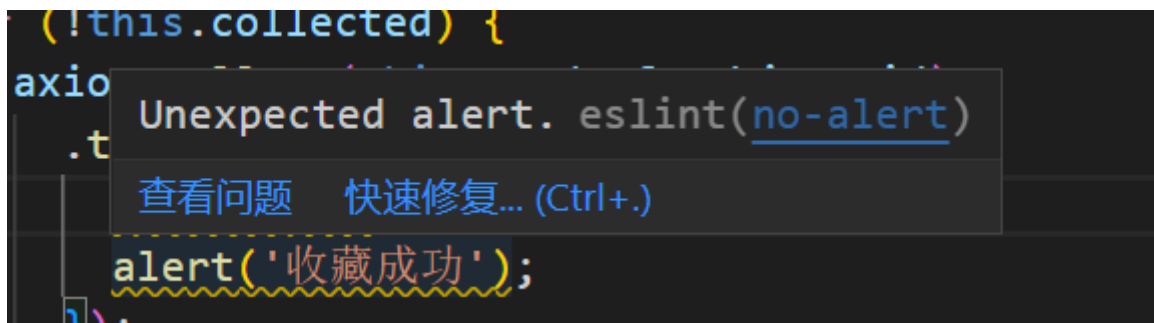
解决办法：将代码依据内容分成4行

### 2.构造函数后缺少括号



解决办法：添加括号

### 3.不合适的alert

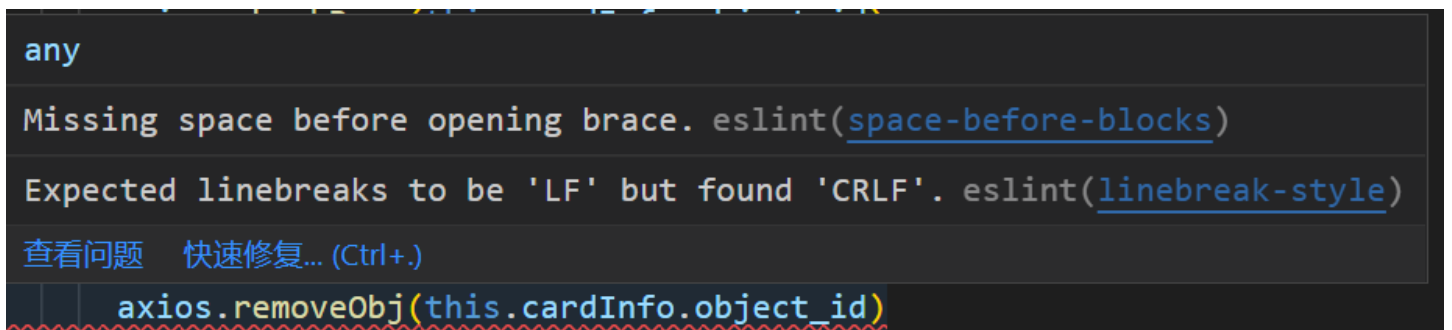


解决办法：改用message提示

处理结果：



### 4.缺少空格、句末缺少分号



解决办法：使用ESLint自动修复

## 5.存在无用导入

```
(alias) const router: VueRouter
import router

'router' is declared but its value is never read. Vetur(6133)
'router' is defined but never used. eslint(no-unused-vars)
```

[查看问题](#) [快速修复...](#) (Ctrl+.)

```
import router from '../router';
```

解决办法：删除该导入

## 6.缺少文件后缀名

```

component: () => import('../views/ItemList.vue'), // lazy-load
},
{
  path: 'ItemDetail/:id',
  name: 'ItemDetail',
  component: () => import('../views/ItemDetail'), // lazy-load
}

```

### 解决办法：添加文件本身扩展名

## 7.多余的文件后缀名

```
// 引入
import Unexpected use of file extension "js" for "./assets/sakura.js" eslint(import/extensions)
import 查看问题 快速修复... (Ctrl+.)
import './assets/sakura.js';
```

解决办法：删除多余的后缀名

## 8.使用 '+' 号拼接字符串

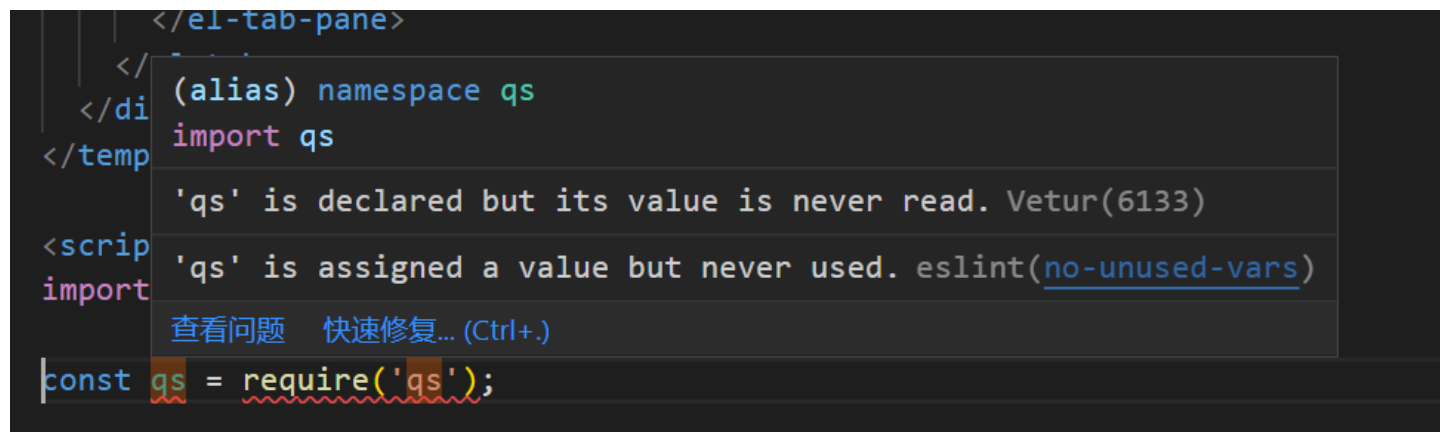
```
getRecommendList() {
  return instance.get(`${url}object/recommend`);
},
getListByType(data) {
  return instance.get(`${url}object/getObjectListByType?` + `type=${data.type}&page=${data.page}`);
}
```

解决办法：用ES6方法来拼接字符串

处理结果:

```
getListByType(data) {
  return instance.get(`${url}object/getObjectListByType?type=${data.type}&page=${data.page}`);
},
searchListByName(data) {
```

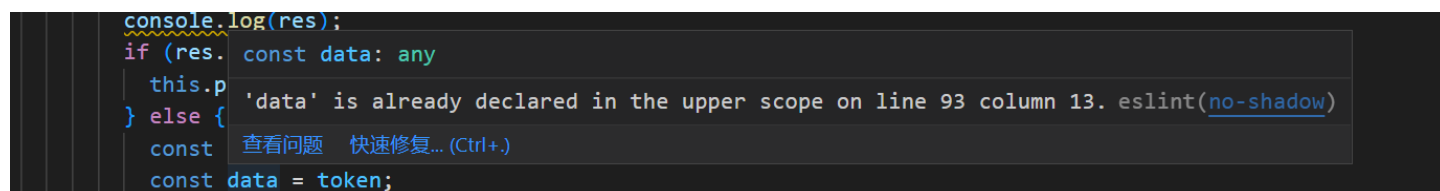
## 9.定义了为未使用的变量



```
</el-tab-pane>
</...
</di...
</temp...
import qs
'qs' is declared but its value is never read. Vetur(6133)
'qs' is assigned a value but never used. eslint(no-unused-vars)
查看问题 快速修复... (Ctrl+.)
const qs = require('qs');
```

解决办法：删除该变量

## 10.有同名变量



```
console.log(res);
if (res. const data: any
  this.p
} else {
  'data' is already declared in the upper scope on line 93 column 13. eslint(no-shadow)
  const
  查看问题 快速修复... (Ctrl+.)
  const data = token;
```

解决办法：修改变量名

# 2.单元测试计划

## 1引言

### 1.1 目的

本文档的读者主要是开发（测试）经理、测试人员和开发人员

### 1.2测试策略

以类为单元，采用独立的单元测试策略，通过设计相应的驱动和桩的方法来测试类中的方法。在选择类的被测方法时，根据方法的规模和复杂度进行判定。非空非注释代码行数LOC>10,或者复杂度VG>3的方法进行单元测试。

对于子类的测试来用分层增量测试(Hierarchical Incremental Testing)策略，对子类的变化部分设计新的测试用例,与父类相同的部分则重用父类的测试用例。

执行单元测试的次序是根据《易租网软件详细设计说明书》中的用例实现交互图，从图中最小依赖关系的类开始测试，再逐步扩大到依赖关系较强的类，直至所有类测试完毕。

### 1.3 范围

单元测试主要包含了计划阶段、设计阶段、实现阶段和执行阶段4个阶段。本单元测试计划是整个软件开发项目中的一部分，起始于详细设计阶段，直到单元测试阶段结束后终止。该计划主要处理与本系统单元测试有关的任务安排、资源需求、人力需求、风险管理、进度安排等内容。

## 1.4参考范围

《易租网软件需求规格说明书》

《易租网软件详细设计说明书》

《易租网用户界面规格说明书》

## 1.5系统大概需求

### 1. 项目概述：

本系统旨在帮助校园内学生转化闲置物品，减少物品闲置率，顺应共享经济的潮流，减少学生的经济负担。本校园租赁系统面向全校用户，其中包括学生，教师以及其他的职工人员。本系统的项目范围包括提供了物品管理平台，用户可以将闲置的物品上架到我们的平台当中，之后可以通过我们的平台进行出租物品的操作。不管是学生还是其他的用户，都可能有某些价值比较昂贵，但是使用频率并不是很高的物品，在购买之后大部分时间里都是处于闲置状态。而对于选择租赁物品的用户，可能是因为对某件产品只有单次的需求，或者是短期的需求，也可以通过我们的平台寻找合适的产品。这样就可以极大地减少用户在这类产品上的经济开销。

### 2. 项目范围

账户管理功能包括登录，注册，个人信息维护功能      物品管理功能包括搜索物品，上传物品。

收藏夹管理功能包括收藏物品，取消收藏等。

需求管理功能包括发布需求等。

### 3. 功能简述：

用户可以进行注册的操作，之后可以修改个人信息，浏览商品信息，进行下单，查看订单信息，上传用户需求等。

## 2.测试项目

根据《软件详细设计说明书》中的详细设计内容，单元测试的测试项目如以下2.1~2.8小节所示

### 2.1 模块

我们以我们系统当中的

- Package UserAdminister（用户信息管理模块）
- Package ObjectAdminister (物品管理模块) 来进行我们的单元测试:

在Package UserAdminister 当中包含了 **UserController.java UserServiceImpl.java**



在Package ObjectAdminister当中包含了 **ObjectController.java ,ObjectServiceImpl.java FavController.java , FavServiceImpl.**

### 3.被测函数

根据测试策略当中制定的被测方法选取标准，被测函数如表所示：

方法标识符	方法名	代码行
LLD_001_FUN_001	public Result<?> login(User user)	15
LLD_001_FUN_002	public Result<?> register(@RequestBody User user)	14
LLD_001_FUN_003	public Result<?> getObjectDetail(int objectId)	10
LLD_001_FUN_004	public Result<?> searchObject(String name, Integer page)	13
LLD_001_FUN_005	public Result<?> collect(Integer favId, Integer objectId)	10
LLD_001_FUN_006	public Result<?> deleteCollection(Integer favId, Integer objectId)	12

### 4.不被测函数（方法）

对不满足测试策略当中的选取被测方法标准的方法将不进行单元测试，但这些方法必须经过严格代码检视，以保证不会出现一些低级性的错误，并且在集成测试阶段同意验证其接口功能的正确性。

不被测函数如表3所示：

方法标识符	方法名	代码行	
LLD_001_UNT_001	public Result<?> modifyUserInfo ( Integer userId, String nickName,String password,String avatar,String gender,Integer telephone,String email,String zone)	9	
LLD_001_UNT_002	public User getUserByToken(@RequestBody String token)	8	
LLD_001_UNT_003	Public Result<?> modifyObject(Integer objectId, String name,String description, Integer deposit, String tag, Integer rentDaily, String newLevel)	6	
LLD_001_UNT_004	public Result<?> viewOwnObjectList(Integer userId)	5	

## 5.测试方法

- 保证所有的语句、分支被覆盖
- 参考等价类划分方法
- 参考边界值分析方法
- 参考使用错误猜测法
- 使用junit来实现单元测试

## 6.测试通过/失败准则

测试通过的标准表述如下：

- (1) 所有单元测试的用例都被执行并通过；
- (2) 所有发现的缺陷都被修正并通过回归测试；
- (3) 所有被测对象的前置条件和后置条件组合覆盖率达到100%，或能明确给出不需要达到的理由；
- (4) 单元测试报告被授权人批准。

测试失败标准表述如下：

- (1) 严重缺陷密度大于15个/KLOC；
- (2) 发现软件结构有重大设计问题，其修改会导致20%以上的接口、功能、数量的变化,进一步测试相关特性已经无意义；
- (3) 发现关键功能未被设计，该功能的设计会导致20%以上的接口、功能、数量的变化,进一步测试相关特性已经无意义。

测试结果审批过程：开发人员提交单元测试报告—开发或测试经理签字并提交 SQA  
SQA 对报告进行评审并签字(测试经理参与)—产品经理签字。

## 7.测试挂起/恢复的条件测试挂起的条件有：

- (1) 当某个类在单元测试执行过程中发现有阻塞用例的时候，该类的单元测试被挂起。
- (2) 当有20%以上的被测类都遇到有阻塞用例时,所有类的单元测试都被挂起。
- (3) 当出现有新增需求的时候，与该需求相关的所有类的单元测试都被挂起。
- (4) 当开发人员提出要进行设计变更的时候，相关类的单元测试将被挂起。

测试恢复的条件有：

- (1) 测试被挂起的条件已经被解决。
- (2) 需要恢复测试的对象达到单元测试入口条件，在这里要求这些被测对象已经通过代码走读(要提交走读报告)和语法检查(要提交检查结果)。

## 8 测试对象

Package ObjectAdminister\_01版本

## 9 用例分析与设计

### 1. login测试与分析

- a. 标识符定义：LLD\_001\_FUN\_001
- b. 被测特性：
  - 输入普通用户账户不存在的时候，登录失败
  - 输入普通用户账户与密码不匹配的时候，登录失败
  - 输入管理员账户不存在的时候，登录失败
  - 输入管理员账户与密码不匹配的时候，登录失败
  - 输入账户为空的时候，登录失败
  - 输入密码为空的时候，登录失败
  - 输入账户存在且与密码匹配时，登录成功

### c. 测试方法：

管理员账户参数的等价类划分考虑为空和非空的情况对于非空情况，可以划分为存在和不存在的  
情况。

普通用户账户参数的等价类划分考虑为空和非空的情况对于非空情况，可以划分为存在和不存  
在的情况。

密钥参数的等价类划分考虑空和非空情况，对于非空情况，可以划分为正确和不正确两种情况

### d. 测试项标识：

测试项标识符	测试项描述	优先级
UT_TC_001_001_001	输入参数任意为空情况	低
UT_TC_001_001_002	输入管理员账号密码不匹配	高
UT_TC_001_001_003	输入管理员账号不存在	中
UT_TC_001_001_004	输入参数合法	高
UT_TC_001_001_005	输入普通用户账号密码不匹配	高

UT_TC_001_001_006	输入普通用户账号不存在	高
-------------------	-------------	---

- a. 测试通过/失败标准：
- 所有的用例都必须被执行，且没有发现错误
- b. 对应用例：

测试用例1

测试编号	UT_TC_001_001_001	
优先级	低	
测试项描述	输入参数任意为空情况	
前置条件	进入登录页面	
用例	输入	期望结果
001	userName="zqf" password="" role="0"	返回false
002	userName="" password="12345" role="0"	返回false
003	userName="zqf" password="12345" role=""	返回false

测试用例2

测试编号	UT_TC_001_001_002	
优先级	高	
测试项描述	输入管理员账号密码不匹配	
前置条件	进入登录页面	
用例	输入	期望结果

001	userName="mjy" password="1234" role="1"	返回false
-----	---	---------

测试用例3

测试编号	UT_TC_001_001_003	
优先级	中	
测试项描述	输入管理员账号不存在	
前置条件	进入登录页面	
用例	输入	期望结果
001	userName="lwy" password="1234" role="1"	返回false

测试用例4

测试编号	UT_TC_001_001_004	
优先级	高	
测试项描述	输入参数合法	
前置条件	进入登录页面	
用例	输入	期望结果
001	userName="mjy" password="12345" role="1"	登录成功
002	userName="zqf" password="123456" role="0"	登录成功

测试用例5

测试编号	UT_TC_001_001_005	
优先级	高	
测试项描述	输入普通用户账号密码不匹配	
前置条件	进入登录页面	
用例	输入	期望结果
001	userName="zqf" password="12345" role="0"	返回false

测试用例6

测试编号	UT_TC_001_001_006	
优先级	高	
测试项描述	输入普通用户账号不存在	
前置条件	进入登录页面	
用例	输入	期望结果
001	userName="lwy" password="12345" role="0"	返回false

2. register测试与分析

- a. 标识符定义：LLD\_001\_FUN\_002
- b. 被测特性：
  - 输入特定参数为空时，注册失败
  - 输入已存在的用户名时，注册失败
  - 输入参数合法且用户名不重复的时候，注册成功

c. 测试方法：

用户的userName参数的等价类划分为空和非空的情况

用户的userName参数不为空的时候又可以划分为已存在和不存在两种情况

用户的password参数划分为空和非空两种情况。

d. 测试项标识：

测试项标识符	测试项描述	优先级
UT_TC_001_002_001	输入特定参数为空的情况	低
UT_TC_001_002_002	输入用户名已存在的情况	高
UT_TC_001_002_003	输入参数合法且用户名不存在的情况	高

e. 测试通过/失败标准：

所有的用例都必须被执行，且没有发现错误

f. 对应用例：

测试用例1

测试编号	UT_TC_001_002_001	
优先级	低	
测试项描述	输入特定参数为空的情况	
前置条件	进入注册页面	
用例	输入	期望结果
001	userName="" password="12345"	返回false
002	userName="lwy" password=""	返回true

测试用例2

测试编号	UT_TC_001_002_002

优先级	高	
测试项描述	输入用户名已存在的情况	
前置条件	进入注册页面	
用例	输入	期望结果
001	userName="mjy" password="12345"	返回false

测试用例3

测试编号	UT_TC_001_002_003	
优先级	高	
测试项描述	输入参数合法且用户名不存在的情况	
前置条件	进入注册页面	
用例	输入	期望结果
001	userName="lwy" password="12345"	返回true

3. getObjectDetail测试与分析

- a. 标识符定义：LLD\_001\_FUN\_003
- b. 被测特性：

输入参数不能为空，输入参数必须存在
- c. 测试方法：

使用等价类测试方法，将输入参数分为非空和空两种情况

如果参数非空，分为输入参数在数据库中存在还是不存在两种情况
- d. 测试项标识：

测试项标识符	测试项描述	优先级
UT_TC_001_003_001	输入参数为空情况	低
UT_TC_001_003_002	输入参数在数据库中不存在	高



UT_TC_001_003_003	输入参数在数据库中存在	高
-------------------	-------------	---

- e. 测试通过/失败标准：  
所有的用例都必须被执行，且没有发现错误

f. 对应用例：

测试用例1

测试编号	UT_TC_001_003_001	
优先级	低	
测试项描述	输入参数为空情况	
前置条件	进入搜索页面	
用例	输入	期望结果
001	objectId=""	返回false

测试用例2

测试编号	UT_TC_001_003_002	
优先级	高	
测试项描述	输入参数在数据库中不存在	
前置条件	进入搜索页面	
用例	输入	期望结果
001	objectId="-1"	返回false

测试用例3

测试编号	UT_TC_001_003_003	
优先级	高	
测试项描述	输入参数在数据库中存在	
前置条件	进入搜索页面	

用例	输入	期望结果
001	objectId="4"	返回物品详情

4. searchObject测试与分析

- a. 标识符定义：LLD\_001\_FUN\_004
- b. 被测特性：

输入参数不能为空，输入参数必须存在
- c. 测试方法：

使用等价类测试方法，将输入参数分为非空和空两种情况

如果参数非空，分为输入参数在数据库中存在还是不存在两种情况
- d. 测试项标识：

测试项标识符	测试项描述	优先级
UT_TC_001_004_001	输入参数为空情况	低
UT_TC_001_004_002	输入参数在数据库中不存在	高
UT_TC_001_004_003	输入参数在数据库中存在	高

- e. 测试通过/失败标准：

所有的用例都必须被执行，且没有发现错误
- f. 对应用例：

测试用例1

测试编号	UT_TC_001_004_001	
优先级	低	
测试项描述	输入参数为空情况	
前置条件	进入搜索页面	
用例	输入	期望结果
001	objectId="", page="1"	返回false

--	--	--

测试用例2

测试编号	UT_TC_001_004_002	
优先级	高	
测试项描述	输入参数在数据库中不存在	
前置条件	进入搜索页面	
用例	输入	期望结果
001	objectId="adad", page="1"	返回false

测试用例3

测试编号	UT_TC_001_004_003	
优先级	高	
测试项描述	输入参数在数据库中存在	
前置条件	进入搜索页面	
用例	输入	期望结果
001	objectId="手表", page="1"	返回物品详情

5. collect测试与分析

- a. 标识符定义：LLD\_001\_FUN\_005
- b. 被测特性：
  - 输入的参数两者都不能为空，
  - 输入的参数两者在数据库当中都存在
  - 收藏的物品不能已经存在于收藏夹当中
- c. 测试方法：

输入的objectId的等价类划分为空和非空情况，输入的userId的等价类划分为空和非空情况  
若均非空，分为uerId存在和不存在，以及objectId存在和不存在两种情况。  
若均非空且，均存在考虑objectId和userId的关系

d. 测试项标识：

测试项标识符	测试项描述	优先级
UT_TC_001_005_001	输入参数任意为空情况	高
UT_TC_001_005_002	输入任意参数不存在	高
UT_TC_001_005_003	输入参数关系已存在	高
UT_TC_001_005_004	输入参数合法	高

e. 测试通过/失败标准：

所有的用例都必须被执行，且没有发现错误

f. 对应用例：

测试用例1

测试编号	UT_TC_001_005_001	
优先级	高	
测试项描述	输入参数任意为空情况	
前置条件	进入物品详情页面	
用例	输入	期望结果
001	favId="4" objectId=""	返回false
002	favId="" objectId="4"	返回false
003	favId="" objectId=""	返回false

测试用例2

测试编号	UT_TC_001_005_002	
优先级	高	
测试项描述	输入任意参数不存在	
前置条件	进入物品详情页面	
用例	输入	期望结果
001	favId="-1" objectId="4"	返回false
002	favId="4" objectId="-1"	返回false
003	favId="-1" objectId="-1"	返回false

测试用例3

测试编号	UT_TC_001_005_03	
优先级	高	
测试项描述	输入参数关系已存在	
前置条件	进入物品详情页面	
用例	输入	期望结果
001	favId="1" objectId="1"	返回false

测试用例4

测试编号	UT_TC_001_005_004	
优先级	高	
测试项描述	输入参数合法	
前置条件	进入物品详情页面	

用例	输入	期望结果
001	fav="1"  objectId="23"	收藏成功

6. deleteCollection测试与分析

a. 标识符定义：LLD\_001\_FUN\_006

b. 被测特性：

输入的参数两者都不能为空，

输入的参数两者在数据库当中都存在

c. 测试方法：

    输入的favId的等价类划分为空和非空情况

    输入的objectId的等价类划分为空和非空情况

    如果两者均非空，考虑两者是否在数据库当中存在

    如果两者均存在，考虑object是否和fav有关系

d. 测试项标识：

测试项标识符	测试项描述	优先级
UT_TC_001_006_001	输入任意参数为空情况	高
UT_TC_001_006_002	输入任意参数不存在情况	高
UT_TC_001_006_003	输入参数均存在，但是object和fav无关	高
UT_TC_001_006_004	输入参数合法	高

e. 测试通过/失败标准：

    所有的用例都必须被执行，且没有发现错误

f. 对应用例：

测试用例1	
测试编号	UT_TC_001_006_001
优先级	高
测试项描述	输入参数任意为空情况

前置条件	进入收藏夹页面	
用例	输入	期望结果
001	favId="1" objectId=""	返回false
002	favId="" objectId="5"	返回false
003	favId="" objectId=""	返回false

测试用例2

测试编号	UT_TC_001_006_002	
优先级	高	
测试项描述	输入任意参数不存在情况	
前置条件	进入收藏夹页面	
用例	输入	期望结果
001	favId="1" objectId="-1"	返回false
002	favId="-1" objectId="5"	返回false
003	favId="-1" objectId="-2"	返回false

测试用例3

测试编号	UT_TC_001_006_003	
优先级	高	

测试项描述	输入参数均存在，但是object和fav无关	
前置条件	进入收藏夹页面	
用例	输入	期望结果
001	favId="2" objectId="1"	返回false

#### 测试用例4

测试编号	UT_TC_001_006_004	
优先级	高	
测试项描述	输入参数合法	
前置条件	进入收藏夹页面	
用例	输入	期望结果
001	favId="1" objectId="7"	删除成功

## 10环境需求

### 10.1硬件需求

win10系统电脑

### 10.2软件需求

Intellij IDEA 2021.3.2

Github, Junit

### 10.3测试工具

JUnit 5.7.2

mockMvc

## 11 角色和职责



单元测试角色和职责如表5所示

角色	职责
产品经理	解决资源（包括人，工具等)需求，对单元测试结果进行监督
开发经理	协助制定单元测试计划，安排单元测试计划
测试经理	制订单元测试计划，安排单元测试任务，参与单元测试结果验收
SQA	对单元测试过程（包括代码走读、正规检视活动）进行监控
开发或（和)测试人员	完成单元测试需要的输入，并完成单元测试设计规格、单元测试用例规格、单元测试规程的制定，执行单元测试，记录发现的问题，修改问题，并负责问题的回归测试。与此同时

## 12 测试脚本

我们使用脚本测试了

- Package UserAdminister（用户信息管理模块）
- Package ObjectAdminister（物品管理模块）

Package UserAdminister，被测源代码为UserController.java ,测试用例脚本为UserController.java，使用mockmvc模拟网络请求。

驱动器UserController.java将自动化地调用测试用例，并得到测试用例的正确执行与否，驱动器使得测试不必与用户界面真正交互，下面为UserController.java的代码：

### 12.1 Package UserAdminister

#### 12.1.1测试脚本

```
@SpringBootTest(classes = {DemoApplication.class}, webEnvironment =
SpringBootTest.WebEnvironment.RANDOM_PORT)
@AutoConfigureMockMvc
class UserControllerTest {
    @Resource
    private MockMvc mockMvc;
```

```

@Test
void UT_TC_001_001_001_1() throws Exception {
    MvcResult result = mockMvc.perform(MockMvcRequestBuilders.post("/user/login")
        .param("nickName", "zqf")
        .param("password", "")
        .param("role", "0"))
        .andExpect(MockMvcResultMatchers.status().isOk())
        .andDo(MockMvcResultHandlers.print())
        .andExpect(MockMvcResultMatchers.jsonPath("code").value(-1))
        .andReturn();
}

```

```

@Test
void UT_TC_001_001_001_2() throws Exception {
    MvcResult result = mockMvc.perform(MockMvcRequestBuilders.post("/user/login")
        .param("nickName", "")
        .param("password", "12345")
        .param("role", "0"))
        .andExpect(MockMvcResultMatchers.status().isOk())
        .andExpect(MockMvcResultMatchers.jsonPath("code").value(-1))
        .andDo(MockMvcResultHandlers.print())
        .andReturn();
}

```

```

@Test
void UT_TC_001_001_001_3() throws Exception {
    MvcResult result = mockMvc.perform(MockMvcRequestBuilders.post("/user/login")
        .param("nickName", "zqf")
        .param("password", "12345")
        .param("role", ""))
        .andExpect(MockMvcResultMatchers.status().isOk())
        .andExpect(MockMvcResultMatchers.jsonPath("code").value(-1))
        .andDo(MockMvcResultHandlers.print())
        .andReturn();
}

```

```

@Test
void UT_TC_001_001_002() throws Exception {
    MvcResult result = mockMvc.perform(MockMvcRequestBuilders.post("/user/login")

```

```

        .param("nickName", "mjoy")
        .param("password", "1234")
        .param("role", "1"))
    .andExpect(MockMvcResultMatchers.status().isOk())
    .andExpect(MockMvcResultMatchers.jsonPath("code").value(-1))
    .andDo(MockMvcResultHandlers.print())
    .andReturn();
}

```

@Test

```

void UT_TC_001_001_003() throws Exception {
    MvcResult result = mockMvc.perform(MockMvcRequestBuilders.post("/user/login")
        .param("nickName", "lwy")
        .param("password", "1234")
        .param("role", "1"))
    .andExpect(MockMvcResultMatchers.status().isOk())
    .andExpect(MockMvcResultMatchers.jsonPath("code").value(-1))
    .andDo(MockMvcResultHandlers.print())
    .andReturn();
}

```

@Test

```

void UT_TC_001_001_004_1() throws Exception {
    MvcResult result = mockMvc.perform(MockMvcRequestBuilders.post("/user/login")
        .param("nickName", "mjoy")
        .param("password", "12345")
        .param("role", "1"))
    .andExpect(MockMvcResultMatchers.status().isOk())
    .andExpect(MockMvcResultMatchers.jsonPath("code").value(0))
    .andDo(MockMvcResultHandlers.print())
    .andReturn();
}

```

@Test

```

void UT_TC_001_001_004_2() throws Exception {
    MvcResult result = mockMvc.perform(MockMvcRequestBuilders.post("/user/login")
        .param("nickName", "zqf")
        .param("password", "123456")
        .param("role", "0"))
    .andExpect(MockMvcResultMatchers.status().isOk())
}

```

```
        .andExpect(MockMvcResultMatchers.jsonPath("code").value(0))
        .andDo(MockMvcResultHandlers.print())
        .andReturn();
    }
}
```

@Test

```
void UT_TC_001_001_005() throws Exception {
    MvcResult result = mockMvc.perform(MockMvcRequestBuilders.post("/user/login")
        .param("nickName", "zqf")
        .param("password", "12345")
        .param("role", "0"))
        .andExpect(MockMvcResultMatchers.status().isOk())
        .andExpect(MockMvcResultMatchers.jsonPath("code").value(-1))
        .andDo(MockMvcResultHandlers.print())
        .andReturn();
}
}
```

@Test

```
void UT_TC_001_001_006() throws Exception {
    MvcResult result = mockMvc.perform(MockMvcRequestBuilders.post("/user/login")
        .param("nickName", "lwy")
        .param("password", "12345")
        .param("role", "0"))
        .andExpect(MockMvcResultMatchers.status().isOk())
        .andExpect(MockMvcResultMatchers.jsonPath("code").value(-1))
        .andDo(MockMvcResultHandlers.print())
        .andReturn();
}
}
```

@Test

```
void UT_TC_001_002_001_1() throws Exception {
    MvcResult result = mockMvc.perform(MockMvcRequestBuilders.post("/user/register")
        .param("nickName", "")
        .param("password", "12345"))
        .andExpect(MockMvcResultMatchers.status().isOk())
        .andExpect(MockMvcResultMatchers.jsonPath("code").value(-1))
        .andDo(MockMvcResultHandlers.print())
        .andReturn();
}
}
```

@Test

void UT\_TC\_001\_002\_001\_2() throws Exception {

```
    MvcResult result = mockMvc.perform(MockMvcRequestBuilders.post("/user/register")
        .param("nickName", "lwy")
        .param("password", ""))
        .andExpect(MockMvcResultMatchers.status().isOk())
        .andExpect(MockMvcResultMatchers.jsonPath("code").value(-1))
        .andDo(MockMvcResultHandlers.print())
        .andReturn();
}
```

@Test

void UT\_TC\_001\_002() throws Exception {

```
    MvcResult result = mockMvc.perform(MockMvcRequestBuilders.post("/user/register")
        .param("nickName", "mjy")
        .param("password", "12345"))
        .andExpect(MockMvcResultMatchers.status().isOk())
        .andExpect(MockMvcResultMatchers.jsonPath("code").value(-1))
        .andDo(MockMvcResultHandlers.print())
        .andReturn();
}
```

@Test

void UT\_TC\_001\_002\_003() throws Exception {

```
    MvcResult result = mockMvc.perform(MockMvcRequestBuilders.post("/user/register")
        .param("nickName", "lwy")
        .param("password", "12345"))
        .andExpect(MockMvcResultMatchers.status().isOk())
        .andExpect(MockMvcResultMatchers.jsonPath("code").value(0))
        .andDo(MockMvcResultHandlers.print())
        .andReturn();
}
}
```

12.1.2测试报告如图所示：

[Collapse](#) | [Expand](#)

UserControllerTest		2.52 s
UT_TC_001_002_001_1()	passed	829 ms
UT_TC_001_002_001_2()	passed	11 ms
UT_TC_001_002()	passed	1.10 s
UT_TC_001_001_001_1()	passed	50 ms
UT_TC_001_001_001_2()	passed	44 ms
UT_TC_001_001_001_3()	passed	48 ms
UT_TC_001_001_004_1()	passed	87 ms
UT_TC_001_001_004_2()	passed	43 ms
UT_TC_001_001_002()	passed	44 ms
UT_TC_001_001_003()	passed	45 ms
UT_TC_001_001_005()	passed	46 ms
UT_TC_001_001_006()	passed	44 ms
UT_TC_001_002_003()	passed	129 ms

由 IntelliJ IDEA 在 2022/6/19 下午7:59 生成

### 12.1.3 以UT\_TC\_001\_001\_001\_1()为例：

UT_TC_001_001_001_1()	passed	50 ms
<div>Creating a new SqlSession SqlSession [org.apache.ibatis.session.defaults.DefaultSqlSession@305e95a4] was not registered for synchronization because synchronization is not active JDBC Connection [HikariProxyConnection@1646118461 wrapping com.mysql.cj.jdbc.ConnectionImpl@4de91056] will not be managed by Spring ==&gt; Preparing: SELECT user_id,nick_name,password,avatar,gender,telephone,email,zone,register_time,role,reputation FROM user WHERE (nick_name = ?) ==&gt; Parameters: zqf(String) &lt;== Columns: user_id, nick_name, password, avatar, gender, telephone, email, zone, register_time, role, reputation &lt;== Row: 1, zqf, 123456, https://tse2-mm.cn.bing.net/th/id/OIP-C.IN0ucr2k2k849LbVOXW-jwHaHa?pid=ImgDet, 男, 114, zqf@qq.com, 嘉定, null, 0, 100 &lt;== Total: 1 Closing non transactional SqlSession [org.apache.ibatis.session.defaults.DefaultSqlSession@305e95a4] MockHttpServletRequest: HTTP Method = POST Request URI = /user/login Parameters = {nickName=[zqf], password=[], role=[0]} Headers = [] Body = null Session Attrs = {} Handler: Type = com.example.demo.controller.UserController Method = com.example.demo.controller.UserController#login(String, String, String) Async: Async started = false Async result = null Resolved Exception: Type = null ModelAndView: View name = null View = null Model = null FlashMap: Attributes = null MockHttpServletResponse: Status = 200 Error message = null Headers = [Vary:"Origin", "Access-Control-Request-Method", "Access-Control-Request-Headers", Content-Type:"application/json"] Content type = application/json Body = {"code":-1,"msg":"ç ¨æ ¨â ¨ç ¨è¼ ¨â ¨¥è ¨", "data":null} Forwarded URL = null Redirected URL = null Cookies = []</div>		

### 12.1.4 覆盖率结果

Current scope: all classes | com.example.demo.controller

Coverage Summary for Class: UserController (com.example.demo.controller)

Class	Class, %	Method, %	Line, %
-------	----------	-----------	---------

```
1 package com.example.demo.controller;
2
3 import cn.hutool.core.util.StrUtil;
4 import cn.hutool.json.JSONObject;
5 import com.auth0.jwt.JWT;
6 import com.auth0.jwt.exceptions.JWTDecodeException;
7 import com.auth0.jwt.interfaces.DecodedJWT;
8 import com.baomidou.mybatisplus.core.conditions.query.LambdaQueryWrapper;
9 import com.baomidou.mybatisplus.core.toolkit.Wrappers;
10 import com.baomidou.mybatisplus.extension.plugins.pagination.Page;
11 import com.example.demo.annotation.PassToken;
12 import com.example.demo.annotation.UserLoginToken;
13 import com.example.demo.common.Result;
14 //import com.example.demo.entity.Object;
15 import com.example.demo.entity.User;
16 import com.example.demo.mapper.UserMapper;
17 //import com.example.demo.service.UserService;
18 import com.example.demo.service.impl.UserServiceImpl;
19 import com.example.demo.util.JwtUtil;
20 import org.springframework.web.bind.annotation.*;
21
22 import javax.annotation.Resource;
23
24 /**
25  * <p>
26  * 用户表 前端控制器
27  * </p>
28  *
29  * @author cJ
30  * @since 2021-12-23
31  */
32 @RestController
33 @RequestMapping("/user")
34 public class UserController {
35
36     @Resource
37     UserMapper userMapper;
38
39     @UserLoginToken
40     @PostMapping("/getList")
41     public Result<?> findPage(@RequestParam(defaultValue = "1") Integer pageNum,
42                               @RequestParam(defaultValue = "10") Integer pageSize,
43                               @RequestParam(defaultValue = "") String search) {
44         LambdaQueryWrapper<User> wrapper = Wrappers.<User>lambdaQuery();
45         if (StrUtil.isNotBlank(search)) {
46             wrapper.like(User::getNickName, search);
47         }
48         Page<User> userPage = userMapper.selectPage(new Page<>(pageNum, pageSize), wrapper);
49         return Result.success(userPage);
50     }
51
52     //新增
53     @UserLoginToken
54     @PostMapping("/insert")
55     @ResponseBody
56     public Result<?> save(User user) {
57         if (user.getPassword() == null) {
58             user.setPassword("123456");
59         }
60         userMapper.insert(user);
61         return Result.success();
62     }
63
64     //登录
65     @PassToken
66     @PostMapping("/login")
67     public Result<?> login(@RequestParam(value = "nickName", required = true) String nickName,
68                           @RequestParam(value = "password", required = true) String password,
69                           @RequestParam(value = "role", required = true) String role) {
70         JSONObject jsonObject = new JSONObject();
71         User res = userMapper.selectOne(Wrappers.<User>lambdaQuery().eq(User::getNickName, nickName));
72
73         if (res == null) {
74             || !res.getPassword().equals(password)
75             || !res.getRole().equals(role)) {
76             return Result.error("-1", "用户名或密码输入错误");
77         } else {
78             String token = JwtUtil.getToken(nickName, password);
79             jsonObject.put("token", token);
80             return Result.success(jsonObject);
81         }
82     }
83
84     @UserLoginToken
85     @GetMapping("/getMessage")
86     public String getMessage() {
87         return "你已通过验证";
88     }
89
90     //注册
91     @PassToken
92     @PostMapping("/register")
93     public Result<?> register(@RequestParam(value = "nickName", required = true) String nickName, @RequestParam(value = "password", required = true) String password) {
94         //判断用户名是否唯一
95         if (nickName.equals("")) {
96             return Result.error("-1", "用户名不能为空");
97         } else if (password.equals("")) {
98             return Result.error("-1", "密码不能为空");
99         }
100         User user = new User();
101         user.setNickName(nickName);
102         user.setPassword(password);
103         User res = userMapper.selectOne(Wrappers.<User>lambdaQuery().eq(User::getNickName, nickName));
104         if (res != null) {
105             return Result.error("-1", "用户名重复");
106         }
107         userMapper.insert(user);
108         return Result.success();
109     }
110
111     @Resource
112     private UserServiceImpl userService;
113
114     /**
115      * 多表联查，一对多，分页
116      *
117      * @param page 当前页
118      * @param size 每页条数
119      */
120     @GetMapping("/pageTestA/{page}/{size}")
121     public Page<User> pageTestA(@PathVariable Integer page, @PathVariable Integer size) {
122         Page<User> iPage = new Page<User>(page, size);
123         return userService.getAll(iPage);
124     }
125
126     // @UserLoginToken
127     @PostMapping("/getUserByToken")
128     public User getUserByToken(@RequestBody String token) {
129         String userNickName;
130         try {
131             DecodedJWT jwt = JWT.decode(token);
132             jwt.getClaim("username").asString();
133             userNickName = jwt.getClaim("username").asString();
134         } catch (JWTDecodeException j) {
135             throw new RuntimeException("401");
136         }
137         User user = userMapper.selectOne(Wrappers.<User>lambdaQuery().eq(User::getNickName, userNickName));
138     }
```



```

138         if (user == null) {
139             throw new RuntimeException("用户不存在");
140         }
141         return user;
142     }
143
144     @PostMapping("modifyUserInfo")
145     public Result<> modifyUserInfo(@RequestParam(required = true) Integer userId,
146                                   @RequestParam(required = false) String nickName,
147                                   @RequestParam(required = false) String password,
148                                   @RequestParam(required = false) String avatar,
149                                   @RequestParam(required = false) String gender,
150                                   @RequestParam(required = false) Integer telephone,
151                                   @RequestParam(required = false) String email,
152                                   @RequestParam(required = false) String zone) {
153         int count = userService.modifyUserInfo(userId, nickName, password, avatar, gender, telephone, email, zone);
154         if (count <= 0) {
155             return Result.error("-1", "修改失败");
156         } else {
157             return Result.success();
158         }
159     }
160
161 }
162
163

```

generated on 2022-06-19 21:50

## 12.2 Package ObjectAdminister

### 12.2.1 测试脚本

@SpringBootTest(classes = {DemoApplication.class}, webEnvironment =  
SpringBootTest.WebEnvironment.RANDOM\_PORT)

@AutoConfigureMockMvc

class ObjectControllerTest {

    @Resource

    private MockMvc mockMvc;

    @Test

    void UT\_TC\_001\_003\_001() throws Exception {

        MvcResult result =

mockMvc.perform(MockMvcRequestBuilders.get("/object/getObjectDetail"  
)

        .param("objectId", ""))

        .andExpect(MockMvcResultMatchers.status().is4xxClientError())

        .andDo(MockMvcResultHandlers.print())

        .andReturn();

    }

    @Test

    void UT\_TC\_001\_003\_002() throws Exception {

        MvcResult result =

mockMvc.perform(MockMvcRequestBuilders.get("/object/getObjectDetail"  
)

        .param("objectId", "-1"))

        .andExpect(MockMvcResultMatchers.status().isOk())



```
        .andDo(MockMvcResultHandlers.print())
        .andReturn();
    }
}
```

```
@Test
void UT_TC_001_003_003() throws Exception {
    MvcResult result =
mockMvc.perform(MockMvcRequestBuilders.get("/object/getObjectDetail"
)

        .param("objectId", "4"))
        .andExpect(MockMvcResultMatchers.status().isOk())
        .andDo(MockMvcResultHandlers.print())
        .andExpect(MockMvcResultMatchers.jsonPath("code").value(0))
        .andReturn();
}
}
```

```
@Test
void UT_TC_001_004_001() throws Exception {
    MvcResult result =
mockMvc.perform(MockMvcRequestBuilders.get("/object/searchObject")
        .param("name", "")
        .param("page", "1"))
        .andExpect(MockMvcResultMatchers.status().isOk())
        .andDo(MockMvcResultHandlers.print())
        .andExpect(MockMvcResultMatchers.jsonPath("code").value(-1))
        .andReturn();
}
}
```

```
@Test
void UT_TC_001_004_002() throws Exception {
    MvcResult result =
mockMvc.perform(MockMvcRequestBuilders.get("/object/searchObject")
        .param("name", "adad")
        .param("page", "1"))
        .andExpect(MockMvcResultMatchers.status().isOk())
        .andDo(MockMvcResultHandlers.print())
        .andExpect(MockMvcResultMatchers.jsonPath("code").value(-1))
}
```

```
        .andReturn();
    }
}
```

```
@Test
void UT_TC_001_004_003() throws Exception {
    MvcResult result =
mockMvc.perform(MockMvcRequestBuilders.get("/object/searchObject")
        .param("name", "手表")
        .param("page", "1"))
        .andExpect(MockMvcResultMatchers.status().isOk())
        .andDo(MockMvcResultHandlers.print())
        .andExpect(MockMvcResultMatchers.jsonPath("code").value(0))
        .andReturn();
}
}
```

```
@Test
void UT_TC_001_005_001_1() throws Exception {
    MvcResult result =
mockMvc.perform(MockMvcRequestBuilders.post("/fav/collect")
        .param("favId", "4")
        .param("objectId", ""))
        .andExpect(MockMvcResultMatchers.status().is4xxClientError())
        .andDo(MockMvcResultHandlers.print())
        .andReturn();
}
}
```

```
@Test
void UT_TC_001_005_001_2() throws Exception {
    MvcResult result =
mockMvc.perform(MockMvcRequestBuilders.post("/fav/collect")
        .param("favId", "4")
        .param("objectId", ""))
        .andExpect(MockMvcResultMatchers.status().is4xxClientError())
        .andDo(MockMvcResultHandlers.print())
        .andReturn();
}
}
```

```
@Test
void UT_TC_001_005_001_3() throws Exception {
    MvcResult result =
mockMvc.perform(MockMvcRequestBuilders.post("/fav/collect")
        .param("favId", "")
        .param("objectId", "4"))
    .andExpect(MockMvcResultMatchers.status().is4xxClientError())
    .andDo(MockMvcResultHandlers.print())
    .andReturn();
}
```

```
@Test
void UT_TC_001_005_002_1() throws Exception {
    MvcResult result =
mockMvc.perform(MockMvcRequestBuilders.post("/fav/collect")
        .param("favId", "-1")
        .param("objectId", "4"))
    .andExpect(MockMvcResultMatchers.status().isOk())
    .andDo(MockMvcResultHandlers.print())
    .andExpect(MockMvcResultMatchers.jsonPath("code").value(-1))
    .andReturn();
}
```

```
@Test
void UT_TC_001_005_002_2() throws Exception {
    MvcResult result =
mockMvc.perform(MockMvcRequestBuilders.post("/fav/collect")
        .param("favId", "4")
        .param("objectId", "-1"))
    .andExpect(MockMvcResultMatchers.status().isOk())
    .andDo(MockMvcResultHandlers.print())
    .andExpect(MockMvcResultMatchers.jsonPath("code").value(-1))
    .andReturn();
}
```

```
@Test
void UT_TC_001_005_002_3() throws Exception {
```

```
    MvcResult result =
mockMvc.perform(MockMvcRequestBuilders.post("/fav/collect")
    .param("favId", "-1")
    .param("objectId", "-1"))
    .andExpect(MockMvcResultMatchers.status().isOk())
    .andDo(MockMvcResultHandlers.print())
    .andExpect(MockMvcResultMatchers.jsonPath("code").value(-1))
    .andReturn();
}
```

```
@Test
void UT_TC_001_005_003() throws Exception {
    MvcResult result =
mockMvc.perform(MockMvcRequestBuilders.post("/fav/collect")
    .param("favId", "1")
    .param("objectId","1"))
    .andExpect(MockMvcResultMatchers.status().isOk())
    .andDo(MockMvcResultHandlers.print())
    .andExpect(MockMvcResultMatchers.jsonPath("code").value(-1))
    .andReturn();
}
```

```
@Test
void UT_TC_001_005_004 () throws Exception {
    MvcResult result =
mockMvc.perform(MockMvcRequestBuilders.post("/fav/collect")
    .param("favId", "1")
    .param("objectId","23"))
    .andExpect(MockMvcResultMatchers.status().isOk())
    .andDo(MockMvcResultHandlers.print())
    .andExpect(MockMvcResultMatchers.jsonPath("code").value(0))
    .andReturn();
}
```

```
@Test
void UT_TC_001_006_001_1 () throws Exception {
```

```
    MvcResult result =  
mockMvc.perform(MockMvcRequestBuilders.get("/fav/deleteCollection")  
    .param("favId", "1")  
    .param("objectId",""))  
    .andExpect(MockMvcResultMatchers.status().is4xxClientError())  
    .andDo(MockMvcResultHandlers.print())  
    .andReturn();  
}
```

@Test

```
void UT_TC_001_006_001_2 () throws Exception {
```

```
    MvcResult result =  
mockMvc.perform(MockMvcRequestBuilders.get("/fav/deleteCollection")  
    .param("favId", "")  
    .param("objectId","5"))  
    .andExpect(MockMvcResultMatchers.status().is4xxClientError())  
    .andDo(MockMvcResultHandlers.print())  
    .andReturn();  
}
```

@Test

```
void UT_TC_001_006_001_3 () throws Exception {
```

```
    MvcResult result =  
mockMvc.perform(MockMvcRequestBuilders.get("/fav/deleteCollection")  
    .param("favId", "")  
    .param("objectId",""))  
    .andExpect(MockMvcResultMatchers.status().is4xxClientError())  
    .andDo(MockMvcResultHandlers.print())  
    .andReturn();  
}
```

@Test

```
void UT_TC_001_006_002_1 () throws Exception {
```

```
    MvcResult result =  
mockMvc.perform(MockMvcRequestBuilders.get("/fav/deleteCollection")  
    .param("favId", "1")  
    .param("objectId","-1"))  
    .andExpect(MockMvcResultMatchers.status().isOk())  
    .andDo(MockMvcResultHandlers.print())
```

```
        .andExpect(MockMvcResultMatchers.jsonPath("code").value(-1))
        .andReturn();
    }
}
```

```
@Test
void UT_TC_001_006_002_2 () throws Exception {
    MvcResult result =
mockMvc.perform(MockMvcRequestBuilders.get("/fav/deleteCollection")
        .param("favId", "-1")
        .param("objectId","5"))
    .andExpect(MockMvcResultMatchers.status().isOk())
    .andDo(MockMvcResultHandlers.print())
    .andExpect(MockMvcResultMatchers.jsonPath("code").value(-1))
    .andReturn();
}
}
```

```
@Test
void UT_TC_001_006_003 () throws Exception {
    MvcResult result =
mockMvc.perform(MockMvcRequestBuilders.get("/fav/deleteCollection")
        .param("favId", "2")
        .param("objectId","1"))
    .andExpect(MockMvcResultMatchers.status().isOk())
    .andDo(MockMvcResultHandlers.print())
    .andExpect(MockMvcResultMatchers.jsonPath("code").value(-1))
    .andReturn();
}
}
```

```
@Test
void UT_TC_001_006_004 () throws Exception {
    MvcResult result =
mockMvc.perform(MockMvcRequestBuilders.get("/fav/deleteCollection")
        .param("favId", "1")
        .param("objectId","7"))
    .andExpect(MockMvcResultMatchers.status().isOk())
    .andDo(MockMvcResultHandlers.print())
    .andExpect(MockMvcResultMatchers.jsonPath("code").value(0))
}
```

```
        .andReturn();
    }
}
```

12.2.2测试结果如果所示：

ObjectControllerTest: 21 total, 21 passed

3.12 s

Collapse | Expand

ObjectControllerTest	3.12 s
UT_TC_001_005_001_1()	passed 626 ms
UT_TC_001_005_001_2()	passed 13 ms
UT_TC_001_005_001_3()	passed 9 ms
UT_TC_001_005_002_1()	passed 204 ms
UT_TC_001_005_002_2()	passed 11 ms
UT_TC_001_005_002_3()	passed 10 ms
UT_TC_001_003_001()	passed 12 ms
UT_TC_001_003_002()	passed 981 ms
UT_TC_001_003_003()	passed 400 ms
UT_TC_001_004_001()	passed 18 ms
UT_TC_001_004_002()	passed 216 ms
UT_TC_001_004_003()	passed 130 ms
UT_TC_001_005_003()	passed 67 ms
UT_TC_001_005_004()	passed 176 ms
UT_TC_001_006_003()	passed 37 ms
UT_TC_001_006_004()	passed 155 ms
UT_TC_001_006_001_1()	passed 11 ms
UT_TC_001_006_001_2()	passed 10 ms
UT_TC_001_006_001_3()	passed 11 ms
UT_TC_001_006_002_1()	passed 11 ms
UT_TC_001_006_002_2()	passed 10 ms

由 IntelliJ IDEA 在 2022/6/19 下午9:39 生成

12.2.3以UT\_TC\_001\_005\_001\_1()为例



ObjectControllerTest3.12 s

UT\_TC\_001\_005\_001\_1()passed626 ms

```
2022-06-19 21:39:17.586 WARN 17748 --- [ main] .w.s.m.s.DefaultHandlerExceptionResolver : Resolved [org.springframework.web.bind.MissingServletRequestParameterException: Required request parameter 'objectId' for method parameter type Integer is present but converted to null]
MockHttpServletRequest:
  HTTP Method = POST
  Request URI = /fav/collect
  Parameters = {favId=[4], objectId=[]}
  Headers = {}
  Body = null
  Session Attrs = {}
Handler:
  Type = com.example.demo.controller.FavController
  Method = com.example.demo.controller.FavController#collect(Integer, Integer)
Async:
  Async started = false
  Async result = null
Resolved Exception:
  Type = org.springframework.web.bind.MissingServletRequestParameterException
ModelAndView:
  View name = null
  View = null
  Model = null
FlashMap:
  Attributes = null
MockHttpServletRequest:
  Status = 400
  Error message = Required request parameter 'objectId' for method parameter type Integer is present but converted to null
  Headers = [Vary:"Origin", "Access-Control-Request-Method", "Access-Control-Request-Headers"]
  Content type = null
  Body =
  Forwarded URL = null
  Redirected URL = null
  Cookies = []
```

12.2.4覆盖率结果



Coverage Summary for Class: FavController (com.example.demo.controller)

Class	Class, %	Method, %	Line, %
FavController	100% (1/1)	100% (5/5)	41% (16/39)

```
1 package com.example.demo.controller;
2
3
4 import cn.hutool.json.JSONObject;
5 //import com.baomidou.mybatisplus.core.conditions.query.QueryWrapper;
6 import com.example.demo.common.Result;
7 //import com.example.demo.entity.Fav;
8 import com.example.demo.mapper.FavMapper;
9 import com.example.demo.mapper.UserMapper;
10 import com.example.demo.service.impl.FavServiceImpl;
11 import org.springframework.web.bind.annotation.*;
12
13 //import org.springframework.stereotype.Controller;
14
15 import javax.annotation.Resource;
16 //import java.util.ArrayList;
17 import java.util.List;
18
19 /**
20  * <p>
21  * 关系表 前端控制器
22  * </p>
23  *
24  * @author cj
25  * @since 2021-12-17
26  */
27 @RestController
28 @RequestMapping("/fav")
29 public class FavController {
30     @Resource
31     FavServiceImpl favService;
32
33     @Resource
34     FavMapper favMapper;
35
36     @Resource
37     ObjectController objectController;
38
39     @PostMapping("collect")
40     public Result<?> collect(@RequestParam("favId") Integer favId, @RequestParam("objectId") Integer objectId) {
41         if (favId != 1 || objectId < 0) {
42             return Result.error("-1", "收藏失败");
43         }
44
45         if (favService.isExist(favId, objectId) != 0) {
46             return Result.error("-1", "已收藏, 收藏失败");
47         }
48         Integer count = favService.collect(favId, objectId);
49         if (count <= 0) {
50             return Result.error("-1", "收藏失败");
51         } else {
52             return Result.success();
53         }
54     }
55
56     @GetMapping("deleteCollection")
57     public Result<?> deleteCollection(@RequestParam(value = "favId", required = true) Integer favId, @RequestParam(value = "objectId", required = true) Integer objectId) {
58         if (favId < 0 || objectId < 0) {
59             return Result.error("-1", "数据有误, 删除失败");
60         }
61
62         if (favService.isExist(favId, objectId) == 0) {
63             return Result.error("-1", "未收藏, 删除失败");
64         }
65         Integer count = favService.deleteCollection(favId, objectId);
66         if (count <= 0) {
67             return Result.error("-1", "收藏删除失败");
68         } else {
69             return Result.success();
70         }
71     }
72
73     @GetMapping("queryObjectInFav")//判断用户是否收藏了该商品
74     public Result<?> queryObjectInFav(Integer objectId, Integer userId) {
75         Integer favId = favService.queryObjectInFav(objectId, userId);
76         if (favId == null) {
77             return Result.error("-1", "用户未收藏该物品");
78         } else {
79             return Result.success(favId);
80         }
81     }
82
83     @GetMapping("viewFavouriteCart")
84     public Result<?> viewFavouriteCart(Integer favId) {
85         List<JSONObject> objectDetails = favService.viewFavouriteCart(favId);
86         if (objectDetails == null) {
87             return Result.error("-1", "收藏夹不存在");
88         }
89         if (objectDetails.isEmpty()) {
90             return Result.error("-2", "该收藏夹为空");
91         } else {
92             return Result.success(objectDetails);
93         }
94     }
95 }
96
97
```

### 3.集成测试

#### 1 引言

##### 1.1目的

本文是大学生租赁平台的集成测试的大纲，主要描述如何进行集成测试活动，如何控制集成测试活动、集成测试活动的流程以及集成测试活动的工作安排等，以保证程序集成起来正常工作，保证程序的完整运行。

## 1.2范围

本测试计划主要是针对软件的集成测试：不含硬件、系统测试以及单元测试（完成单元测试是前提）。

主要的任务：

- (1)测试在把各个模块连接起来的时候，穿越模块接口的数据是否会丢失；
- (2)测试各个子功能组合起来，能否达到预期要求的父功能；
- (3)一个模块的功能是否会对另一个模块的功能产生不利的影响；
- (4)全局数据结构是否有问题；
- (5)单个模块的误差积累起来，是否会放大，从而达到不可接受的程度。

主要测试方法是黑盒测试方法。必要的集成测试是回归测试。

本文主要的读者对象是项目负责人、集成部门负责人、集成测试设计师。

## 1.3 术语

和业务及技术相关的术语。

## 1.4 测试环境

测试环境

序号	描述	配置
1	客户端浏览器	Chrome, Microsoft Edge, Firefox
2	输入习惯	中文，英文
3	服务端操作系统	Windows10
4	测试工具	JUnit, Selenium, PingCode, Jmeter
5	数据库	MySQL

## 1.5 参考文件及工作产品

《校园租赁系统需求规格说明书》

《校园租赁系统概要设计设计说明书》

## 2 集成策略

### 2.1 进入标准

编码完成，单元测试完成。集成测试计划完成，时间表、工具以及人员安排到位。

### 2.2集成内容

1.函数集成

如函数间接口、函数是否调用正常。

2.功能集成

如不同函数间实现的业务功能。

3.数据集成

如数据传递是否正确，对于传入值的控制范围是否一致等。

4.子系统集成

把不同账户管理子系统、收藏夹子系统进行集成。

### 2.3集成策略

采用自顶向下的集成方式。自顶向下的集成测试就是按照系统层次结构图，以主程序模块为中心，采用自上而下的对各个模块一边组装一边进行测试。选择这种集成方法，在测试的过程中，可以较早地验证主要的控制和判断模块。在一个功能划分合理的程序模块结构中，控制和判断模块常常出现在较高的层次中，因而可以提前做测试，以便发现问题。

集成测试的主要步骤：

- 制订并审核集成测试计划。
- 测试用例分析、设计及评审。
- 测试的实施。
- 测试的执行。
- 测试的分析和评估。

如下所示

集成测试的主要步骤

活动	输入	输出	职责
----	----	----	----

制定并审核集成测试计划	概要设计说明书等	集成测试计划	制定测试计
测试用例分析和设计及评审	集成测试计划概要设计说明书	设计测试用例	设计测试月
测试的实施	集成测试用例测试过程	测试脚本、测试环境、测试驱动或桩	开发测试朋 驱动或桩
测试的执行	测试脚本	测试结果	记录结果、
测试的分析和评估	集成测试计划测试结果	测试分析和评估报告	会同开发人 试报告

2.4集成顺序

1.软件集成顺序

自顶向下。

2.软件/硬件集成顺序

无。

3.子系统集成顺序

略。

3 测试过程描述

在单元测试中以Package UserAdminister（用户信息管理模块）和Package ObjectAdminister（物品管理模块） 来进行单元测试。在集成测试中实现对Package UserAdminister（用户信息管理模块）和Package ObjectAdminister（物品管理模块） 以及Package UserAdminister模块和Package ObjectAdminister模块与界面模块的集成。

3.1 软件集成测试

在本项目中，集成测试的主要过程如下：

(1)设计集成测试用例：

自顶向下集成测试的步骤：

- (1) 以主模块为所测模块兼驱动模块，所有直属主模块的下属模块全部用桩模块对主模块进行测试。
- (2) 采用深度遍历或广度遍历的策略，用实际源代码模块替换相应桩模块，再用桩代替它们的下属模块，与已经测试的模块或子系统集成为新的子系统，每次只替换一个桩为源代码。

(3) 进行回归测试（即重新执行以前的全部测试或部分测试用例），排除集成过程中引起错误的可能。

(4) 判断是否所有的模块都已经集成到系统中，是则结束测试，否则转到（2）去继续执行。

(2)集成测试：组织人员按照(1)中的集成测试用例测试系统集成度

- 测试人员按照测试用例逐项进行测试，并且将测试结果填写在测试报告上(测试报告必须覆盖所有测试用例);
- 测试过程中发现Bug,将Bug填写在PingCode上记录在云平台上（Bug状态为New）。
- 对应负责人接到PingCode通过Email发过来的Bug信息；
- 对于明显的并且可以立刻解决的Bug，将Bug发给开发人员（Bug状态为ASSIGNED，对于不是Bug的提交，集成部经理通知测试设计人员和测试人员，对相应文档进行修改（Bug状态为RESOLVED,决定设置为INVALID）；对于目前无法修改的，将这个Bug放到下一轮次进行修改（Bug状态为REOPENED，决定设置为REMIND）。

(3)问题反馈：反馈Bug给开发人员。

- 开发人员接到发过来的Bug立刻修改(Bug状态为RESOLVED.决定设置为FIXED);
- 测试人员接到 Jira 通过email发过来的Bug更改信息，应该逐项复测，填写新的测试报告(测试报告必须覆盖上一次中所有REOPENED的测试用例)。

(4)回归测试：重新测试修复Bug后的系统。重复(3)，直到(4)中回归测试结果达到系统验收标准。

- 如果复测有问题返回第(2)步（Bug状态为REOPENED）,否则关闭这项Bug（Bug状态为CLOSED）。
- 本轮测试中测试用例中有90%一次性通过测试，结束测试任务；本轮测试中发现的Bug有95%经过修改并且通过再次测试（即Bug状态为CLOSED），返回进行新一轮测试。

(5)集成测试测试总结报告：完成以上4步后，综合相关资料生成报告。

(6)进入系统测试：α测试、β测试。

## 3.2 软件/硬件测试

主要涉及硬件和软件间集成，硬件和硬件间集成这里一般不涉及，集成关注:

(1) 功能点：根据用户文档列出所有功能点，检验其正确性。

(2)接口：根据用户文档列出所有接口，检验其正确性。

(3)流程处理：根据用户文档列出所有流程，检验其正确性。

(4)外部接口：根据用户文档列出所有外部接口，检验其正确性。

## 3.3 子系统集成测试

完成子系统间集成。

## 4 集成测试验收标准

### 4.1 模块（指单元集成后的模块）验收标准

- 接口：接口提供的功能或者数据正确。
- 功能点：验收程序和产品描述、用户文档中的全部说明相对应，一致性。
- 流程处理：验证程序与产品描述、用户文档中的全部说明相对应，一致性。
- 外部接口：验证程序与产品描述、用户文档中的全部说明相对于，一致性。

### 4.2 集成测试验收标准

首先，集成测试用例中所设计的功能测试用例必须全部通过，性能及其他类型测试用例95%以上通过。在未通过的测试用例中，不能含有“系统崩溃”和“严重错误”这两种错误，“一般错误”小于1%。测试结果与测试用例中期望的结果一致，测试通过，否则标明测试未通过。

## 5 测试工具

### 5.1 测试工具

- 测试中心平台：JUnit
- 性能测试工具：JMeter
- 集成测试工具：Selenium

### 5.2 其他工具

电子表格软件：Excel

缺陷跟踪工具：Pingcode

## 6 挂起恢复和退出条件

### 6.1 挂起

- 进入第一轮集成测试，测试人员大体了解一下产品情况，如果发现在单元内存在三个以上错误或缺陷以及操作性的错误，退回单元测试组测试；

- 遇到有项目优先级更高的集成测试任务；
- 在复测过程中发现产品无法运行下去；
- 人员、设备不足；
- 重大突发紧急情况。

## 6.2 恢复

- 符合进入集成测试条件；
- 项目优先级更高的集成测试任务暂告完成；
- 复测过程中产品可以运行下去；
- 人员、设备到位；
- 突发事件处理完成。

## 6.3 退出

- 项目因故终止；
- 不可抗力：合同专用条款中约定等级以上的自然灾害也属不可抗力；
- 其他原因的测试工作频频被挂起或者挂起后迟迟恢复不了，并超过了客户要求的期限。

## 7 责任人

- 测试负责人：程敬
  - 控制并完成测试任务和测试过程，决定测试人员提交上来的Bug是否需要修改。
- 测试设计人员：赵兰德龙，罗检，刘招兵。
  - 设计集成测试用例。
- 测试人员：程敬，赵兰德龙，罗检，刘招兵。
  - 按照测试用例进行测试活动。
- 开发人员：程敬，赵兰德龙，罗检，刘招兵。
  - 程序Bug修改，程序员间协调。
- 用户代表：无。

## 8 记录和解决问题

- 记录：利用PingCode平台记录Bug，并指定相关责任人。
- 解决问题：小组会议以及开发人员协调负责人，协调测试开发之间的工作。

## 9 集成测试接口分析

Package UserAdminister模块接口如表9-1所示。

表9-1 Package UserAdminister模块接口

方法标识符	方法名	调用层次
HLD_001_INT_001	public Result<?> login(User user)	2
HLD_001_INT_002	public Result<?> register(@RequestBody User user)	1
HLD_001_INT_003	public Result<?> getObjectDetail(int objectId)	1
HLD_001_INT_004	public Result<?> searchObject(String name, Integer page)	1
HLD_001_INT_005	public Result<?> collect(Integer favId, Integer objectId)	1
HLD_001_INT_006	public Result<?> deleteCollection(Integer favId, Integer objectId)	1

Package UserAdminister模块和Package ObjectAdminister模块集成接口如表9-2所示。

表9-2 Package UserAdminister模块和Package ObjectAdminister模块集成接口

标识符	名称	调用层次数	调用函数次数
HLD_003_INT_001	commit()	1	1

Package UserAdminister模块、Package ObjectAdminister模块和界面模块集成接口如表9-3所示。

表9-3 Package UserAdminister模块、Package ObjectAdminister模块和界面模块集成接口

接口序号	接口描述
GUN_INT_001	主页面用户登录界面设置接口
GUN_INT_002	主页面注册新用户界面设置接口
GUN_INT_003	商品详细信息界面设置接口
GUN_INT_004	用户搜索商品搜索结果显示接口
GUN_INT_005	添加商品进入收藏夹界面设置接口



通过分析Package UserAdminister模块和Package ObjectAdminister模块的每个对外的接口函数，其中的许多函数都只有调用一层的函数。因此只测login, register和searchObject三个接口。对于GUI接口，鉴于其测试与系统测试有重复性，这里只进行最基本的接口功能验证。

被测接口及标识如表9-4所示。

表9-4 被测接口及标识

标识符	名称
HLD_001_INT_001	public Result<?> login
HLD_001_INT_002	public Result<?> register(@Rec
HLD_001_INT_004	public Result<?> searchObject(Str
GUN_INT_001	主页面用户登录界面
GUN_INT_002	主页面注册新用户界i
GUN_INT_003	商品详细信息界面i
GUN_INT_004	用户搜索商品搜索结!
GUN_INT_005	添加商品进入收藏夹界
GUN_INT_006	删除收藏夹商品界面

## 10 采用的测试方法和技术

(1)对UserAdminister模块内集成，例子仅对login和register进行测试。

(2)在完成UserAdminister模块集成测试后，对UserAdminister模块和ObjectAdminister模块进行集成测试。对于login, register和searchObject这两个接口的测试，可以从接口的输入域和输出域覆盖上进行测试用例的设计，同时考虑接口功能的完整性：虽然仅仅是两个模块的集成，但实际采用的是自顶向下的集成方法。

(3)对于界面对接口，利用场景法设计测试用例，根据用例进行手动验证其功能，同时，关注其是否调用正确的接口函数，并且是否能够正确地传递输入参数，正确地获取返回值和输出参数。

(4)参考等价类划分方法。

(5)参考边界值分析方法。

(6)参考使用错误猜测方法。

(7)覆盖分析。

## 11 测试通过/失败标准

测试通过的标准表述如下：

- 所有的接口用例都被执行过并通过：
- 所有发现的真实缺陷都被修正并回归测试：
- login接口、register接口和searchObject接口的输入域被10%覆盖；
- login接口、register接口和searchObject接口的函数调用路径被100%覆盖。

测试失败的标准表述如下：

- 缺陷密度大于2个/KLOC；
- 发现有重大结构设计问题，其修改会导致20%以上的函数接口、功能、模块数量的变化，进一步测试相关接口已经无意义；
- 发现关键功能未被设计，该功能的设计会导致20%以上的函数接口、功能、模块数量的变化，进一步测试相关接口已经无意义。

## 12 测试对象

(1)Package UserAdminister\_01版本

(2)Package ObjectAdminister\_01版本

(3)界面模块tag\_01版本

## 13 用例分析与设计

### 1. HLD\_001\_INT\_001测试分析与设计

- a. 设计标识符：IT\_TD\_001。
- b. 被测特性。
  - 输入普通用户账户不存在的时候，登录失败
  - 输入普通用户账户与密码不匹配的时候，登录失败
  - 输入管理员账户不存在的时候，登录失败
  - 输入管理员账户与密码不匹配的时候，登录失败
  - 输入账户为空的时候，登录失败
  - 输入密码为空的时候，登录失败
  - 输入账户存在且与密码匹配时，登录成功

c. 测试方法。

分析login的函数调用关系图和流程图，从login函数测试知道，用桩对JwtUtil.getToken,UserMapper.getNickname,UserMapper.getPassword进行替代。因此对于login的集成来说，关键是验证login与JwtUtil.getToken, UserMapper.getNickname, UserMapper.getPassword的接口是否与预计的效果相同。

d. 测试项标识如表9-5所示。

表9-5 测试项标识1

测试项标识符	测试项描述	优先级
IT_TD_001_001	输入参数任意为空情况	低
IT_TD_001_002	输入管理员账号密码不匹配	高
IT_TD_001_003	输入管理员账号不存在	中
IT_TD_001_004	输入参数合法	高
IT_TD_001_005	输入普通用户账号密码不匹配	高
IT_TD_001_006	输入普通用户账号不存在	高

e. 测试通过/失败标准：

所有的用例都必须被执行，且没有发现缺陷

f. 对应用例：

测试用例1

测试编号	IT_TD_001_001	
优先级	低	
测试项描述	测试参数任意为空情况	
前置条件	用户单击“登录”按钮，进入登录界面	
用例	输入	期望结果
001	userName="zldl" password="" role="0"	返回false 反馈密码不能为空的错误信息

002	userName="" password="12345" role="0"	返回false  反馈userName不能为空的错误信息
003	userName="zldl" password="12345" role=""	返回false

测试用例2

测试编号	IT_TD_001_002	
优先级	高	
测试项描述	输入管理员账号密码不匹配	
前置条件	管理员单击“登录”按钮，进入登录界面	
用例	输入	期望结果
001	userName="mgy" password="1234" role="1"	返回false  反馈用户帐号或密码错误的错误信息

测试用例3

测试编号	IT_TD_001_003	
优先级	中	
测试项描述	输入管理员账号不存在	
前置条件	管理员单击“登录”按钮，进入登录界面	
用例	输入	期望结果
001	userName="lwy" password="1234" role="1"	返回false  反馈用户帐号或密码错误的错误信息

测试用例4

测试编号	IT_TD_001_004	
优先级	高	
测试项描述	输入参数合法	
前置条件	用户单击“登录”按钮，进入登录界面	
用例	输入	期望结果
001	userName="mjy" password="12345" role="1"	登录成功
002	userName="zdl" password="12345" role="0"	登录成功

测试用例5

测试编号	IT_TD_001_005	
优先级	高	
测试项描述	输入普通用户账号密码不匹配	
前置条件	用户单击“登录”按钮，进入登录界面	
用例	输入	期望结果
001	userName="zqf" password="12345" role="0"	返回false  反馈用户帐号或密码错误的错误信息

测试用例6

测试编号	IT_TD_001_006	
优先级	高	
测试项描述	输入普通用户账号不存在	
前置条件	用户单击“登录”按钮，进入登录界面	

用例	输入	期望结果
001	userName="lwy"  password="12345"  role="0"	返回false  反馈用户帐号或密码错误的错误信息

2. HLD\_001\_INT\_002测试分析与设计

- a. 设计标识符：IT\_TD\_002。
- b. 被测特性：
  - 输入特定参数为空时，注册失败
  - 输入已存在的用户名时，注册失败
  - 输入参数合法且用户名不重复的时候，注册成功

c. 测试方法：

分析register的函数调用关系图和流程图，从register函数测试知道，用桩对UserMapper.getNickname, UserMapper.getPassword进行了替代。因此对于register的集成来说，关键是验证register与UserMapper.getNickname, UserMapper.getPassword的接口是否与预计的效果相同。

d. 测试项标识：

测试项标识符	测试项描述	优先级
IT_TD_002_001	输入特定参数为空的情况	低
IT_TD_002_002	输入用户名已存在的情况	高
IT_TD_002_003	输入参数合法且用户名不存在的情况	高

e. 测试通过/失败标准：

所有的用例都必须被执行，且没有发现缺陷

f. 对应用例：

测试用例1

测试编号	IT_TD_002_001
------	---------------

优先级	低	
测试项描述	输入特定参数为空的情况	
前置条件	进入注册页面	
用例	输入	期望结果
001	userName="" password="12345"	返回false  反馈userName不能为空的错误信息
002	userName="testIT1" password=""	返回true

测试用例2

测试编号	IT_TD_002_002	
优先级	高	
测试项描述	输入用户名已存在的情况	
前置条件	进入注册页面	
用例	输入	期望结果
001	userName="zldl" password="12345"	返回false  反馈用户名重复的错误信息

测试用例3

测试编号	IT_TD_002_003	
优先级	高	
测试项描述	输入参数合法且用户名不存在的情况	
前置条件	进入注册页面	
用例	输入	期望结果
001	userName="testIT2" password="12345"	返回true

3. HLD\_001\_INT\_004测试与分析

a. 标识符定义：IT\_TD\_003。

b. 被测特性：

输入参数不能为空，输入参数必须存在

c. 测试方法：

分析searchObject的函数调用关系图和流程图，从searchObject函数测试知道，用桩对objectService.searchObject进行了替代。因此对于searchObject的集成来说，关键是验证searchObject与objectService.searchObject的接口是否与预计的效果相同。

d. 测试项标识：

测试项标识符	测试项描述	优先级
IT_TD_003_001	输入参数为空情况	低
IT_TD_003_002	输入参数在数据库中不存在	高
IT_TD_003_003	输入参数在数据库中存在	高

e. 测试通过/失败标准：

所有的用例都必须被执行，且没有发现错误

f. 对应用例：

测试用例1

测试编号	IT_TD_003_001	
优先级	低	
测试项描述	输入参数为空情况	
前置条件	进入搜索页面	
用例	输入	期望结果
001	objectId="", page="1"	返回false  展示搜索结果为空信息



测试用例2

测试编号	IT_TD_003_002	
优先级	高	
测试项描述	输入参数在数据库中不存在	
前置条件	进入搜索页面	
用例	输入	期望结果
001	objectId="adad", page="1"	页面没有发生改变

测试用例3

测试编号	IT_TD_003_003	
优先级	高	
测试项描述	输入参数在数据库中不存在	
前置条件	进入搜索页面	
用例	输入	期望结果
001	objectId="手表", page="1"	返回物品详情

4. GUN\_INT\_001测试分析与设计

- a. 设计标识符：IT\_TD\_004。
- b. 被测特性。  
主页面用户登录界面设置接口

- c. 测试方法。  
检测输入的合法用户帐号、密码是否能够正确的传递给login接口函数。检验接口输出的正确性分为两方面：一方面，验证能否正确获取login的返回值；另一方面，能够正确获取到错误信息并显示给用户。
- d. 测试项标识如表9-5所示。

表9-5 测试项标识1

测试项标识符	测试项描述	优先级
IT_TD_004_001	验证接口的输入正确性	高
IT_TD_004_002	验证返回值处理的正确性	高
IT_TD_004_003	验证错误信息捕获的正确性	高

- e. 测试通过/失败标准：
- 所有的用例都必须被执行，且没有发现缺陷。
- f. 对应用例：

测试用例1

测试编号	IT_TD_004_001	
优先级	低	
测试项描述	验证接口的输入正确性	
前置条件	用户单击“登录”按钮，进入登录界面	
用例	输入	期望结果
001	userName="zqf" password="" role="0"	返回false 反馈密码不能为空的错误信息
002	userName="" password="12345" role="0"	返回false 反馈userName不能为空的错误信息
003	userName="zqf" password="12345" role=""	返回false 反馈角色不能为空的错误信息

测试用例2

测试编号	IT_TD_004_002
优先级	高

测试项描述	验证返回值处理的正确性	
前置条件	用户单击“登录”按钮，进入登录界面	
用例	输入	期望结果
001	userName="mjy" password="12345" role="1"	登录成功
002	userName="zdl" password="12345" role="0"	登录成功

测试用例3

测试编号	IT_TD_004_003	
优先级	高	
测试项描述	验证错误信息捕获的正确性	
前置条件	用户单击“登录”按钮，进入登录界面	
用例	输入	期望结果
001	userName="zqf" password="12345" role="0"	返回false 反馈用户帐号或密码错误的错误信息
001	userName="lwy" password="12345" role="0"	返回false 反馈用户帐号或密码错误的错误信息

5. GUN\_INT\_002测试分析与设计

- a. 设计标识符：IT\_TD\_005。
- b. 被测特性：  
主页面注册新用户界面设置接口
- c. 测试方法：

检测输入的合法注册信息是否能够正确的传递给register接口函数。检测输入的必填项为空时，是否能正确提示出错信息。检验接口输出的正确性分为两方面：一方面，验证能否正确获取register的返回值；另一方面，能够正确获取到错误信息并显示给用户。

d. 测试项标识：

测试项标识符	测试项描述	优先级
IT_TD_005_001	验证接口的输入正确性	高
IT_TD_005_002	验证返回值处理的正确性	高
IT_TD_005_003	验证错误信息捕获的正确性	高

e. 测试通过/失败标准：

所有的用例都必须被执行，且没有发现错误

f. 对应用例：

测试用例1

测试编号	IT_TD_005_001	
优先级	高	
测试项描述	验证接口的输入正确性	
前置条件	进入注册页面	
用例	输入	期望结果
001	userName="" password="12345"	返回false 反馈用户名不能为空的错误信息

测试用例2

测试编号	IT_TD_005_002	
优先级	高	
测试项描述	验证返回值处理的正确性	
前置条件	进入注册页面	

用例	输入	期望结果
001	userName="testIT3" password="12345"	返回true

测试用例3

测试编号	IT_TD_005_003	
优先级	高	
测试项描述	验证错误信息捕获的正确性	
前置条件	进入注册页面	
用例	输入	期望结果
001	userName="zldl" password="12345"	返回false 反馈用户名重复的错误信息

6. GUN\_INT\_003测试与分析

a. 标识符定义：IT\_TD\_006

b. 被测特性：  
商品详细信息界面设置接口

c. 测试方法：  
检测输入的合法商品Id信息是否能够正确地传递给getObjectDetail接口函数。检测输入的必填项为空时，是否能正确提示出错信息。检验接口输出的正确性分为两方面：一方面，验证能否正确获取getObjectDetail的返回值；另一方面，能够正确获取到错误信息并显示给用户。

d. 测试项标识：

测试项标识符	测试项描述	优先级
IT_TD_006_001	验证接口调用的正确性	高

e. 测试通过/失败标准：  
所有的用例都必须被执行，且没有发现错误

f. 对应用例：

测试用例1

测试编号	IT_TD_006_001	
优先级	高	
测试项描述	验证接口调用的正确性	
前置条件	进入搜索页面	
用例	输入	期望结果
001	objectId="4"	返回物品详情并展示

7. GUN\_INT\_004测试与分析

- a. 标识符定义：IT\_TD\_007。
- b. 被测特性：  
    用户搜索商品搜索结果显示接口
- c. 测试方法：

检测输入的合法商品Id信息是否能够正确地传递给searchObject接口函数。检测输入的必填项为空时，是否能正确提示出错信息。检验接口输出的正确性分为两方面：一方面，验证能否正确searchObject的返回值；另一方面，能够正确获取到错误信息并显示给用户。

- d. 测试项标识：

测试项标识符	测试项描述	优先级
IT_TD_007_001	验证接口的输入正确性	高
IT_TD_007_002	验证返回值处理的正确性	高
IT_TD_007_003	验证错误信息捕获的正确性	高

- e. 测试通过/失败标准：  
    所有的用例都必须被执行，且没有发现错误
- f. 对应用例：

测试编号	IT_TD_007_001	
优先级	高	
测试项描述	验证接口的输入正确性	
前置条件	进入搜索页面	
用例	输入	期望结果
001	objectId="", page="1"	页面不发生改变

测试用例2

测试编号	IT_TD_007_002	
优先级	高	
测试项描述	验证返回值处理的正确性	
前置条件	进入搜索页面	
用例	输入	期望结果
001	objectId="手表", page="1"	返回物品详情

测试用例3

测试编号	IT_TD_007_003	
优先级	高	
测试项描述	验证错误信息捕获的正确性	
前置条件	进入搜索页面	
用例	输入	期望结果
001	objectId="adad", page="1"	返回false 展示搜索结果为空信息

8. GUN\_INT\_005测试与分析

a. 标识符定义：IT\_TD\_008

b. 被测特性：

添加商品进入收藏夹界面设置接口

c. 测试方法：

检测输入的合法商品信息是否能够正确地传递给collect接口函数。检测输入的必填项为空时，是否能正确提示出错信息。检验接口输出的正确性分为两方面：一方面，验证能否正确collect的返回值；另一方面，能够正确获取到错误信息并显示给用户。

d. 测试项标识：

测试项标识符	测试项描述	优先级
IT_TD_008_001	验证接口调用的正确性	高

e. 测试通过/失败标准：

所有的用例都必须被执行，且没有发现错误

f. 对应用例：

测试用例1

测试编号	IT_TD_008_001	
优先级	高	
测试项描述	验证接口调用的正确性	
前置条件	进入物品详情页面	
用例	输入	期望结果
001	fav="1" objectId="23"	收藏成功

9. GUN\_INT\_006测试与分析

a. 标识符定义：IT\_TD\_009

b. 被测特性：

删除收藏夹商品界面设置接口

c. 测试方法：



检测输入的合法商品信息是否能够正确地传递给deleteCollection接口函数。检测输入的必填项为空时，是否能正确提示出错信息。检验接口输出的正确性分为两方面：一方面，验证能否正确deleteCollection的返回值；另一方面，能够正确获取到错误信息并显示给用户。

d. 测试项标识：

测试项标识符	测试项描述	优先级
IT_TD_009_001	验证接口调用的正确性	高

e. 测试通过/失败标准：

所有的用例都必须被执行，且没有发现错误

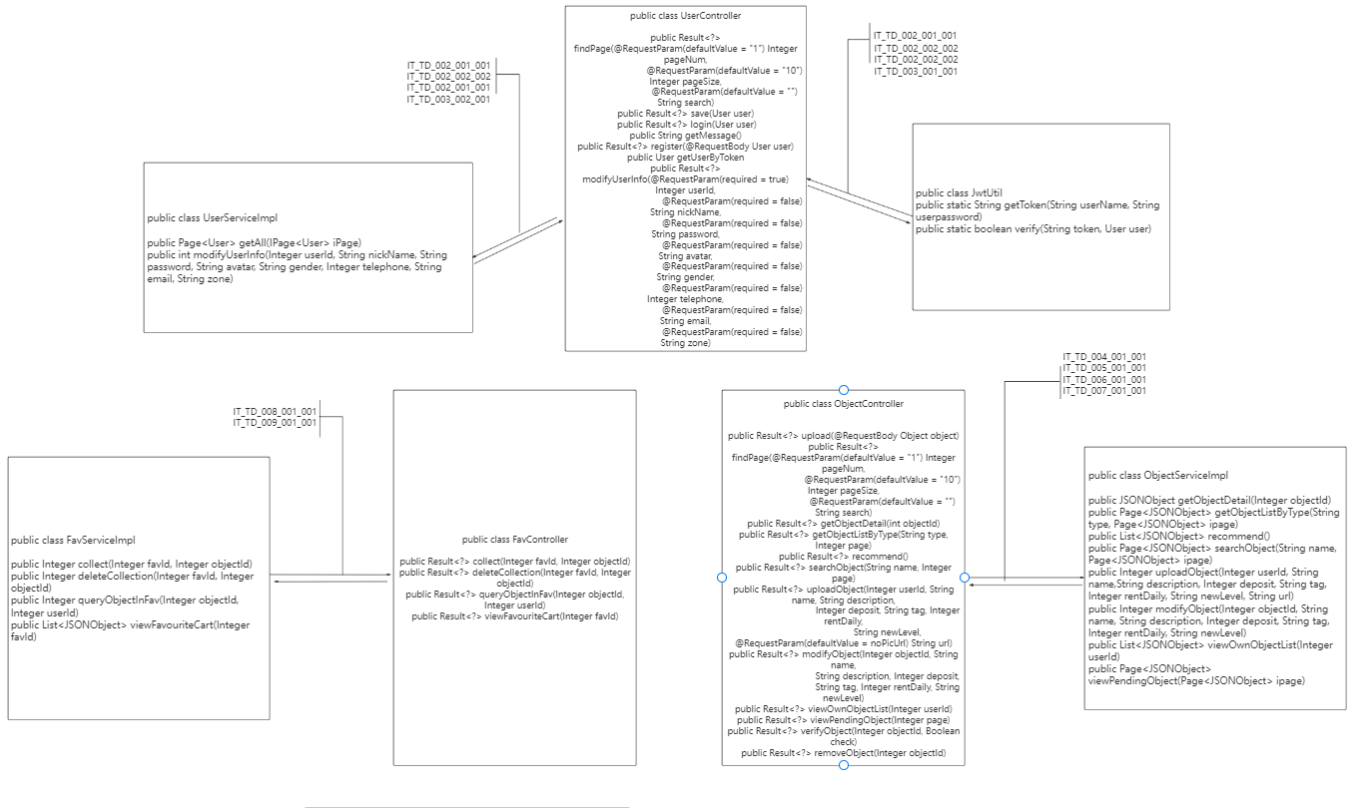
f. 对应用例：

测试用例1

测试编号	IT_TD_009_001	
优先级	高	
测试项描述	验证接口调用的正确性	
前置条件	进入收藏夹页面	
用例	输入	期望结果
001	favId="1" objectId="7"	删除成功

## 14 消息调用图及覆盖率

下图是Package UserAdminister模块和Package ObjectAdminister模块之间的消息调用。在Package UserAdminister 当中包含了 UserController.java UserServiceImpl.java；在Package ObjectAdminister当中包含了 ObjectController.java ,ObjectServiceImpl.java FavController.java , FavServiceImpl，在测试这两模块之间的接口需要调用 UserServiceImpl类方法，用桩代替。



模块之间消息调用图

覆盖率分析表如下

覆盖率分析表

标识符	名称	覆盖
HLD_001_INT_001	public Result<?> login(User user)	100%
HLD_001_INT_002	public Result<?> register(@RequestBody User user)	100%
HLD_001_INT_004	public Result<?> searchObject(String name, Integer page)	100%
GUN_INT_001	主页面用户登录界面设置接口	100%
GUN_INT_002	主页面注册新用户界面设置接口	100%
GUN_INT_003	商品详细信息界面设置接口	100%
GUN_INT_004	用户搜索商品搜索结果显示接口	100%
GUN_INT_005	添加商品进入收藏夹界面设置接口	100%
GUN_INT_006	删除收藏夹商品界面设置接口	100%

## 15 集成测试用例与报告

我们使用selenium进行了Package UserAdminister（用户信息管理模块）与Package ObjectAdminister（物品管理模块）的集成测试。

### 15.1 测试脚本

```
package integrationTest;// Generated by Selenium IDE
```

```
import org.junit.Test;
```

```
import org.junit.Before;
```

```
import org.junit.After;
```

```
import static org.junit.Assert.*;
```

```
import static org.hamcrest.CoreMatchers.is;
```

```
import static org.hamcrest.core.IsNot.not;
```

```
import org.openqa.selenium.By;
```

```
import org.openqa.selenium.WebDriver;
```

```
import org.openqa.selenium.firefox.FirefoxDriver;
```

```
import org.openqa.selenium.chrome.ChromeDriver;
```

```
import org.openqa.selenium.remote.RemoteWebDriver;
```

```
import org.openqa.selenium.remote.DesiredCapabilities;
```

```
import org.openqa.selenium.Dimension;
```

```
import org.openqa.selenium.WebElement;
```

```
import org.openqa.selenium.interactions.Actions;
```

```
import org.openqa.selenium.support.ui.ExpectedConditions;
```

```
import org.openqa.selenium.support.ui.WebDriverWait;
```

```
import org.openqa.selenium.JavascriptExecutor;
```

```
import org.openqa.selenium.Alert;
```

```
import org.openqa.selenium.Keys;
```

```
import java.util.*;
```

```
import java.net.MalformedURLException;
```

```
import java.net.URL;
```

```
import java.util.concurrent.TimeUnit;
```

```
public class IntegrationtestTest {
```

```

private WebDriver driver;

private Map<String, Object> vars;

JavascriptExecutor js;

@Before

public void setUp() {

    driver = new FirefoxDriver();

    js = (JavascriptExecutor) driver;

    vars = new HashMap<String, Object>();

}

@After

public void tearDown() {

    driver.quit();

}

@Test

public void iTTD001001001() {

    // Test name: IT_TD_001_001_001

    // Step # | name | target | value

    // 1 | open | /homepage/home |

    driver.get("http://localhost:9876/homepage/home");

    // 2 | setWindowSize | 1095x790 |

    driver.manage().window().setSize(new Dimension(1095, 790));

    // 3 | click | linkText=登录 |

    driver.findElement(By.linkText("登录")).click();

    // 4 | click | css=.login-input:nth-child(1) |

    driver.manage().timeouts().implicitlyWait(10, TimeUnit.SECONDS);

    driver.findElement(By.cssSelector(".login-input")).click();

    // 5 | type | css=.login-input:nth-child(1) | zldl

    driver.findElement(By.cssSelector(".login-input")).sendKeys("zldl");

    // 6 | click | css=.login-input:nth-child(3) |

    driver.findElement(By.cssSelector(".login-input:nth-child(3)")).click();

    // 7 | click | css=.login-btn |

```

```
driver.findElement(By.cssSelector(".login-btn")).click();
// 8 | assertText | css=.input-check:nth-child(4) | 请输入密码
assertThat(driver.findElement(By.cssSelector(".input-check:nth-child(4)")).getText(), is("请输入
密码"));
// 9 | close | |
driver.close();
}

@Test
public void iTTD001001002() {
    // Test name: IT_TD_001_001_002
    // Step # | name | target | value
    // 1 | open | /homepage/home |
    driver.get("http://localhost:9876/homepage/home");
    // 2 | setWindowSize | 1095x790 |
    driver.manage().window().setSize(new Dimension(1095, 790));
    // 3 | click | linkText=登录 |
    driver.findElement(By.linkText("登录")).click();
    // 4 | click | css=.login-input:nth-child(3) |
    driver.findElement(By.cssSelector(".login-input:nth-child(3)")).click();
    // 5 | type | css=.login-input:nth-child(3) | 12345
    driver.findElement(By.cssSelector(".login-input:nth-child(3)")).sendKeys("12345");
    // 6 | click | css=.login-btn |
    driver.findElement(By.cssSelector(".login-btn")).click();
    // 7 | assertText | css=.input-check:nth-child(2) | 用户名不能为空
    assertThat(driver.findElement(By.cssSelector(".input-check:nth-child(2)")).getText(), is("用户名
不能为空"));
    // 8 | close | |
    driver.close();
}

@Test
public void iTTD001001003() {
```

```
// Test name: IT_TD_001_001_003
// Step # | name | target | value
}

@Test
public void iTTD001002001() {
    // Test name: IT_TD_001_002_001
    // Step # | name | target | value
    // 1 | open | /homepage/home |
    driver.get("http://localhost:9876/homepage/home");
    // 2 | setWindowSize | 1095x790 |
    driver.manage().window().setSize(new Dimension(1095, 790));
    // 3 | click | linkText=登录 |
    driver.findElement(By.linkText("登录")).click();
    // 4 | click | css=.admin-login |
    driver.findElement(By.cssSelector(".admin-login")).click();
    // 5 | click | css=.login-input:nth-child(1) |
    driver.findElement(By.cssSelector(".login-input:nth-child(1)")).click();
    // 6 | type | css=.login-input:nth-child(1) | mgy
    driver.findElement(By.cssSelector(".login-input:nth-child(1)")).sendKeys("mgy");
    // 7 | click | css=.login-input:nth-child(3) |
    driver.findElement(By.cssSelector(".login-input:nth-child(3)")).click();
    // 8 | type | css=.login-input:nth-child(3) | 1234
    driver.findElement(By.cssSelector(".login-input:nth-child(3)")).sendKeys("1234");
    // 9 | click | css=.login-btn |
    driver.findElement(By.cssSelector(".login-btn")).click();
    // 10 | assertText | css=.input-check:nth-child(4) | 用户名或密码错误
    assertThat(driver.findElement(By.cssSelector(".input-check:nth-child(4)")).getText(), is("用户名或密码错误"));
    // 11 | close | |
    driver.close();
}
```

@Test

```
public void iTTD001003001() {  
    // Test name: IT_TD_001_003_001  
    // Step # | name | target | value  
    // 1 | open | /homepage/home |  
    driver.get("http://localhost:9876/homepage/home");  
    // 2 | setWindowSize | 1095x790 |  
    driver.manage().window().setSize(new Dimension(1095, 790));  
    // 3 | click | linkText=登录 |  
    driver.findElement(By.linkText("登录")).click();  
    // 4 | click | css=.admin-login |  
    driver.findElement(By.cssSelector(".admin-login")).click();  
    // 5 | click | css=.login-input:nth-child(1) |  
    driver.findElement(By.cssSelector(".login-input:nth-child(1)")).click();  
    // 6 | type | css=.login-input:nth-child(1) | lwy  
    driver.findElement(By.cssSelector(".login-input:nth-child(1)")).sendKeys("lwy");  
    // 7 | click | css=.login-input:nth-child(3) |  
    driver.findElement(By.cssSelector(".login-input:nth-child(3)")).click();  
    // 8 | type | css=.login-input:nth-child(3) | 1234  
    driver.findElement(By.cssSelector(".login-input:nth-child(3)")).sendKeys("1234");  
    // 9 | click | css=.login-btn |  
    driver.findElement(By.cssSelector(".login-btn")).click();  
    // 10 | assertText | css=.input-check:nth-child(4) | 用户名或密码错误  
    assertThat(driver.findElement(By.cssSelector(".input-check:nth-child(4)")).getText(), is("用户名  
或密码错误"));  
    // 11 | close | |  
    driver.close();  
}
```

@Test

```
public void iTTD001004001() {  
    // Test name: IT_TD_001_004_001
```

```

// Step # | name | target | value
// 1 | open | /homepage/home |
driver.get("http://localhost:9876/homepage/home");
// 2 | setWindowSize | 1095x790 |
driver.manage().window().setSize(new Dimension(1095, 790));
// 3 | click | linkText=登录 |
driver.findElement(By.linkText("登录")).click();
// 4 | click | css=.admin-login |
driver.findElement(By.cssSelector(".admin-login")).click();
// 5 | click | css=.login-input:nth-child(1) |
driver.findElement(By.cssSelector(".login-input:nth-child(1)")).click();
// 6 | type | css=.login-input:nth-child(1) | m jy
driver.findElement(By.cssSelector(".login-input:nth-child(1)")).sendKeys("m jy");
// 7 | click | css=.login-input:nth-child(3) |
driver.findElement(By.cssSelector(".login-input:nth-child(3)")).click();
// 8 | type | css=.login-input:nth-child(3) | 12345
driver.findElement(By.cssSelector(".login-input:nth-child(3)")).sendKeys("12345");
// 9 | click | css=.login-btn |
driver.findElement(By.cssSelector(".login-btn")).click();
// 10 | assertText | css=.username | m jy
assertThat(driver.findElement(By.cssSelector(".username")).getText(), is("m jy"));
// 11 | close | |
driver.close();
}

```

@Test

```

public void iTTD001004002() {
// Test name: IT_TD_001_004_002
// Step # | name | target | value
// 1 | open | /homepage/home |
driver.get("http://localhost:9876/homepage/home");
// 2 | setWindowSize | 1095x790 |

```



```

driver.manage().window().setSize(new Dimension(1095, 790));
// 3 | click | linkText=登录 |
driver.findElement(By.linkText("登录")).click();
// 4 | click | css=.login-input:nth-child(1) |
driver.findElement(By.cssSelector(".login-input:nth-child(1)")).click();
// 5 | type | css=.login-input:nth-child(1) | zldl
driver.findElement(By.cssSelector(".login-input:nth-child(1)")).sendKeys("zldl");
// 6 | click | css=.login-input:nth-child(3) |
driver.findElement(By.cssSelector(".login-input:nth-child(3)")).click();
// 7 | type | css=.login-input:nth-child(3) | 12345
driver.findElement(By.cssSelector(".login-input:nth-child(3)")).sendKeys("12345");
// 8 | click | css=.login-btn |
driver.findElement(By.cssSelector(".login-btn")).click();
// 9 | assertText | css=.section-banner-title | 租你想租的
assertThat(driver.findElement(By.cssSelector(".section-banner-title")).getText(), is("租你想租的"));
// 10 | close | |
driver.close();
}

@Test
public void iTTD001005001() {
    // Test name: IT_TD_001_005_001
    // Step # | name | target | value
    // 1 | open | /homepage/home |
    driver.get("http://localhost:9876/homepage/home");
    // 2 | setWindowSize | 1095x824 |
    driver.manage().window().setSize(new Dimension(1095, 824));
    // 3 | click | linkText=登录 |
    driver.findElement(By.linkText("登录")).click();
    // 4 | click | css=.login-input:nth-child(1) |
    driver.findElement(By.cssSelector(".login-input:nth-child(1)")).click();

```

```

// 5 | type | css=.login-input:nth-child(1) | zqf
driver.findElement(By.cssSelector(".login-input:nth-child(1)")).sendKeys("zqf");

// 6 | click | css=.login-input:nth-child(3) |
driver.findElement(By.cssSelector(".login-input:nth-child(3)")).click();

// 7 | type | css=.login-input:nth-child(3) | 12345
driver.findElement(By.cssSelector(".login-input:nth-child(3)")).sendKeys("12345");

// 8 | click | css=.login-btn |
driver.findElement(By.cssSelector(".login-btn")).click();

// 9 | assertText | css=.input-check:nth-child(4) | 用户名或密码错误
assertThat(driver.findElement(By.cssSelector(".input-check:nth-child(4)")).getText(), is("用户名或密码错误"));

// 10 | close | |
driver.close();
}

@Test
public void iTTD001006001() {
    // Test name: IT_TD_001_006_001
    // Step # | name | target | value
    // 1 | open | /homepage/home |
    driver.get("http://localhost:9876/homepage/home");
    // 2 | setWindowSize | 1095x824 |
    driver.manage().window().setSize(new Dimension(1095, 824));
    // 3 | click | linkText=登录 |
    driver.findElement(By.linkText("登录")).click();
    // 4 | click | css=.login-input:nth-child(1) |
    driver.findElement(By.cssSelector(".login-input:nth-child(1)")).click();
    // 5 | type | css=.login-input:nth-child(1) | lwy
    driver.findElement(By.cssSelector(".login-input:nth-child(1)")).sendKeys("lwy");
    // 6 | click | css=.login-input:nth-child(3) |
    driver.findElement(By.cssSelector(".login-input:nth-child(3)")).click();
    // 7 | type | css=.login-input:nth-child(3) | 12345

```

```
driver.findElement(By.cssSelector(".login-input:nth-child(3)")).sendKeys("12345");
// 8 | click | css=.login-btn |
driver.findElement(By.cssSelector(".login-btn")).click();
// 9 | assertText | css=.input-check:nth-child(4) | 用户名或密码错误
assertThat(driver.findElement(By.cssSelector(".input-check:nth-child(4)")).getText(), is("用户名
或密码错误"));
// 10 | close | |
driver.close();
}

@Test
public void iTTD002001001() {
    // Test name: IT_TD_002_001_001
    // Step # | name | target | value
    // 1 | open | /homepage/home |
    driver.get("http://localhost:9876/homepage/home");
    // 2 | setWindowSize | 1142x824 |
    driver.manage().window().setSize(new Dimension(1142, 824));
    // 3 | click | linkText=登录 |
    driver.findElement(By.linkText("登录")).click();
    // 4 | click | css=.register |
    driver.findElement(By.cssSelector(".register")).click();
    // 5 | click | css=.register-input:nth-child(3) |
    driver.findElement(By.cssSelector(".register-input:nth-child(3)")).click();
    // 6 | type | css=.register-input:nth-child(3) | 12345
    driver.findElement(By.cssSelector(".register-input:nth-child(3)")).sendKeys("12345");
    // 7 | click | css=.register-input:nth-child(5) |
    driver.findElement(By.cssSelector(".register-input:nth-child(5)")).click();
    // 8 | type | css=.register-input:nth-child(5) | 12345
    driver.findElement(By.cssSelector(".register-input:nth-child(5)")).sendKeys("12345");
    // 9 | click | css=.login-btn |
    driver.findElement(By.cssSelector(".login-btn")).click();
```

```

// 10 | assertText | css=.input-check:nth-child(2) | 用户名不能为空
assertThat(driver.findElement(By.cssSelector(".input-check:nth-child(2)")).getText(), is("用户名不能为空"));

// 11 | close | |
driver.close();
}

@Test
public void iTTD002001002() {
    // Test name: IT_TD_002_001_002
    // Step # | name | target | value
    // 1 | open | /homepage/home |
    driver.get("http://localhost:9876/homepage/home");
    // 2 | setWindowSize | 1142x824 |
    driver.manage().window().setSize(new Dimension(1142, 824));
    // 3 | click | linkText=登录 |
    driver.findElement(By.linkText("登录")).click();
    // 4 | click | css=.register |
    driver.findElement(By.cssSelector(".register")).click();
    // 5 | click | css=.register-input:nth-child(1) |
    driver.findElement(By.cssSelector(".register-input:nth-child(1)")).click();
    // 6 | type | css=.register-input:nth-child(1) | reportIT1
    driver.findElement(By.cssSelector(".register-input:nth-child(1)")).sendKeys("reportIT1");
    // 7 | click | css=.login-btn |
    driver.findElement(By.cssSelector(".login-btn")).click();
    // 8 | executeScript | return window.location.href | result
    vars.put("result", js.executeScript("return window.location.href"));
    // 9 | assert | result | http://localhost:9876/login
    assertEquals(vars.get("result").toString(), "http://localhost:9876/login");
    // 10 | close | |
    driver.close();
}

```

@Test

```
public void iTTD002002001() {
```

```
// Test name: IT_TD_002_002_001
```

```
// Step # | name | target | value
```

```
// 1 | open | /homepage/home |
```

```
driver.get("http://localhost:9876/homepage/home");
```

```
// 2 | setWindowSize | 1142x824 |
```

```
driver.manage().window().setSize(new Dimension(1142, 824));
```

```
// 3 | click | linkText=登录 |
```

```
driver.findElement(By.linkText("登录")).click();
```

```
// 4 | click | css=.register |
```

```
driver.findElement(By.cssSelector(".register")).click();
```

```
// 5 | click | css=.register-input:nth-child(1) |
```

```
driver.findElement(By.cssSelector(".register-input:nth-child(1)")).click();
```

```
// 6 | type | css=.register-input:nth-child(1) | zldl
```

```
driver.findElement(By.cssSelector(".register-input:nth-child(1)")).sendKeys("zldl");
```

```
// 7 | click | css=.register-input:nth-child(3) |
```

```
driver.findElement(By.cssSelector(".register-input:nth-child(3)")).click();
```

```
// 8 | type | css=.register-input:nth-child(3) | 12345
```

```
driver.findElement(By.cssSelector(".register-input:nth-child(3)")).sendKeys("12345");
```

```
// 9 | click | css=.register-input:nth-child(5) |
```

```
driver.findElement(By.cssSelector(".register-input:nth-child(5)")).click();
```

```
// 10 | type | css=.register-input:nth-child(5) | 12345
```

```
driver.findElement(By.cssSelector(".register-input:nth-child(5)")).sendKeys("12345");
```

```
// 11 | click | css=.login-btn |
```

```
driver.findElement(By.cssSelector(".login-btn")).click();
```

```
// 12 | assertText | css=.input-check:nth-child(2) | 用户名重复
```

```
assertThat(driver.findElement(By.cssSelector(".input-check:nth-child(2)")).getText(), is("用户名重复"));
```

```
// 13 | close | |
```

```
driver.close();
```

```
}
```

```
@Test
```

```
public void iTTD002003001() {
```

```
    // Test name: IT_TD_002_003_001
```

```
    // Step # | name | target | value
```

```
    // 1 | open | /homepage/home |
```

```
    driver.get("http://localhost:9876/homepage/home");
```

```
    // 2 | setWindowSize | 1142x824 |
```

```
    driver.manage().window().setSize(new Dimension(1142, 824));
```

```
    // 3 | click | linkText=登录 |
```

```
    driver.findElement(By.linkText("登录")).click();
```

```
    // 4 | click | css=.register |
```

```
    driver.findElement(By.cssSelector(".register")).click();
```

```
    // 5 | click | css=.register-input:nth-child(1) |
```

```
    driver.findElement(By.cssSelector(".register-input:nth-child(1)")).click();
```

```
    // 6 | type | css=.register-input:nth-child(1) | reportIT2
```

```
    driver.findElement(By.cssSelector(".register-input:nth-child(1)")).sendKeys("reportIT2");
```

```
    // 7 | click | css=.register-input:nth-child(3) |
```

```
    driver.findElement(By.cssSelector(".register-input:nth-child(3)")).click();
```

```
    // 8 | type | css=.register-input:nth-child(3) | 12345
```

```
    driver.findElement(By.cssSelector(".register-input:nth-child(3)")).sendKeys("12345");
```

```
    // 9 | click | css=.register-input:nth-child(5) |
```

```
    driver.findElement(By.cssSelector(".register-input:nth-child(5)")).click();
```

```
    // 10 | type | css=.register-input:nth-child(5) | 12345
```

```
    driver.findElement(By.cssSelector(".register-input:nth-child(5)")).sendKeys("12345");
```

```
    // 11 | click | css=.login-btn |
```

```
    driver.findElement(By.cssSelector(".login-btn")).click();
```

```
    // 12 | close | |
```

```
    driver.close();
```

```
}
```

```
@Test
```

```
public void iTTD003001001() {  
    // Test name: IT_TD_003_001_001  
    // Step # | name | target | value  
    // 1 | open | /homepage/home |  
    driver.get("http://localhost:9876/homepage/home");  
    // 2 | setWindowSize | 1095x755 |  
    driver.manage().window().setSize(new Dimension(1095, 755));  
    // 3 | click | css=.section-banner-button |  
    driver.findElement(By.cssSelector(".section-banner-button")).click();  
    // 4 | executeScript | return window.location.href | result  
    vars.put("result", js.executeScript("return window.location.href"));  
    // 5 | assert | result | http://localhost:9876/homepage/home  
    assertEquals(vars.get("result").toString(), "http://localhost:9876/homepage/home");  
    // 6 | close | |  
    driver.close();  
}
```

@Test

```
public void iTTD003002001() {  
    // Test name: IT_TD_003_002_001  
    // Step # | name | target | value  
    // 1 | open | /homepage/home |  
    driver.get("http://localhost:9876/homepage/home");  
    // 2 | setWindowSize | 1095x824 |  
    driver.manage().window().setSize(new Dimension(1095, 824));  
    // 3 | click | css=.section-banner-input |  
    driver.findElement(By.cssSelector(".section-banner-input")).click();  
    // 4 | type | css=.section-banner-input | adad  
    driver.findElement(By.cssSelector(".section-banner-input")).sendKeys("adad");  
    // 5 | click | css=.section-banner-button |  
    driver.findElement(By.cssSelector(".section-banner-button")).click();  
    // 6 | executeScript | return window.location.href | result
```

```

vars.put("result", js.executeScript("return window.location.href"));

// 7 | assert | result | http://localhost:9876/homepage/items?mode=2&q=adad
assertEquals(vars.get("result").toString(), "http://localhost:9876/homepage/items?mode=2&q=adad");

// 8 | close | |
driver.close();
}

@Test
public void iTTD003003001() {
    // Test name: IT_TD_003_003_001
    // Step # | name | target | value
    // 1 | open | /homepage/home |
    driver.get("http://localhost:9876/homepage/home");
    // 2 | setWindowSize | 1095x755 |
    driver.manage().window().setSize(new Dimension(1095, 755));
    // 3 | click | css=.section-banner-input |
    driver.findElement(By.cssSelector(".section-banner-input")).click();
    // 4 | type | css=.section-banner-input | 手表
    driver.findElement(By.cssSelector(".section-banner-input")).sendKeys("手表");
    // 5 | click | css=.section-banner-button |
    driver.findElement(By.cssSelector(".section-banner-button")).click();
    // 6 | executeScript | return window.location.href | result
    vars.put("result", js.executeScript("return window.location.href"));
    // 7 | assert | result | http://localhost:9876/homepage/items?mode=2&q=%E6%89%8B%E8%A1%A8
    assertEquals(vars.get("result").toString(), "http://localhost:9876/homepage/items?mode=2&q=%E6%89%8B%E8%A1%A8");
    // 8 | close | |
    driver.close();
}

@Test

```



```
public void iTTD004001001() {  
    // Test name: IT_TD_004_001_001  
    // Step # | name | target | value  
    // 1 | open | /homepage/home |  
    driver.get("http://localhost:9876/homepage/home");  
    // 2 | setWindowSize | 1095x790 |  
    driver.manage().window().setSize(new Dimension(1095, 790));  
    // 3 | click | linkText=登录 |  
    driver.findElement(By.linkText("登录")).click();  
    // 4 | click | css=.login-input:nth-child(1) |  
    driver.findElement(By.cssSelector(".login-input:nth-child(1)")).click();  
    // 5 | type | css=.login-input:nth-child(1) | zldl  
    driver.findElement(By.cssSelector(".login-input:nth-child(1)")).sendKeys("zldl");  
    // 6 | click | css=.login-input:nth-child(3) |  
    driver.findElement(By.cssSelector(".login-input:nth-child(3)")).click();  
    // 7 | click | css=.login-btn |  
    driver.findElement(By.cssSelector(".login-btn")).click();  
    // 8 | assertText | css=.input-check:nth-child(4) | 请输入密码  
    assertThat(driver.findElement(By.cssSelector(".input-check:nth-child(4)")).getText(), is("请输入  
密码"));  
    // 9 | close | |  
    driver.close();  
}
```

@Test

```
public void iTTD004001002() {  
    // Test name: IT_TD_004_001_002  
    // Step # | name | target | value  
    // 1 | open | /homepage/home |  
    driver.get("http://localhost:9876/homepage/home");  
    // 2 | setWindowSize | 1095x790 |  
    driver.manage().window().setSize(new Dimension(1095, 790));
```

```

// 3 | click | linkText=登录 |
driver.findElement(By.linkText("登录")).click();

// 4 | click | css=.login-input:nth-child(3) |
driver.findElement(By.cssSelector(".login-input:nth-child(3)")).click();

// 5 | type | css=.login-input:nth-child(3) | 12345
driver.findElement(By.cssSelector(".login-input:nth-child(3)")).sendKeys("12345");

// 6 | click | css=.login-btn |
driver.findElement(By.cssSelector(".login-btn")).click();

// 7 | assertText | css=.input-check:nth-child(2) | 用户名不能为空
assertThat(driver.findElement(By.cssSelector(".input-check:nth-child(2)")).getText(), is("用户名不能为空"));

// 8 | close | |
driver.close();
}

@Test
public void iTTD004001003() {
    // Test name: IT_TD_004_001_003
    // Step # | name | target | value
}

@Test
public void iTTD004002001() {
    // Test name: IT_TD_004_002_001
    // Step # | name | target | value
    // 1 | open | /homepage/home |
    driver.get("http://localhost:9876/homepage/home");
    // 2 | setWindowSize | 1095x790 |
    driver.manage().window().setSize(new Dimension(1095, 790));
    // 3 | click | linkText=登录 |
    driver.findElement(By.linkText("登录")).click();
    // 4 | click | css=.admin-login |
    driver.findElement(By.cssSelector(".admin-login")).click();

```

```
// 5 | click | css=.login-input:nth-child(1) |
driver.findElement(By.cssSelector(".login-input:nth-child(1)")).click();

// 6 | type | css=.login-input:nth-child(1) | mgy
driver.findElement(By.cssSelector(".login-input:nth-child(1)")).sendKeys("mgy");

// 7 | click | css=.login-input:nth-child(3) |
driver.findElement(By.cssSelector(".login-input:nth-child(3)")).click();

// 8 | type | css=.login-input:nth-child(3) | 12345
driver.findElement(By.cssSelector(".login-input:nth-child(3)")).sendKeys("12345");

// 9 | click | css=.login-btn |
driver.findElement(By.cssSelector(".login-btn")).click();

// 10 | assertText | css=.username | mgy
assertThat(driver.findElement(By.cssSelector(".username")).getText(), is("mgy"));

// 11 | close | |
driver.close();
}
```

@Test

```
public void iTTD004002002() {
    // Test name: IT_TD_004_002_002
    // Step # | name | target | value
    // 1 | open | /homepage/home |
    driver.get("http://localhost:9876/homepage/home");
    // 2 | setWindowSize | 1095x790 |
    driver.manage().window().setSize(new Dimension(1095, 790));
    // 3 | click | linkText=登录 |
    driver.findElement(By.linkText("登录")).click();
    // 4 | click | css=.login-input:nth-child(1) |
    driver.findElement(By.cssSelector(".login-input:nth-child(1)")).click();
    // 5 | type | css=.login-input:nth-child(1) | zldl
    driver.findElement(By.cssSelector(".login-input:nth-child(1)")).sendKeys("zldl");
    // 6 | click | css=.login-input:nth-child(3) |
    driver.findElement(By.cssSelector(".login-input:nth-child(3)")).click();
}
```

```
// 7 | type | css=.login-input:nth-child(3) | 12345
driver.findElement(By.cssSelector(".login-input:nth-child(3)")).sendKeys("12345");

// 8 | click | css=.login-btn |
driver.findElement(By.cssSelector(".login-btn")).click();

// 9 | assertText | css=.section-banner-title | 租你想租的
assertThat(driver.findElement(By.cssSelector(".section-banner-title")).getText(), is("租你想租的"));

// 10 | close | |
driver.close();
}

@Test
public void iTTD004003001() {
    // Test name: IT_TD_004_003_001
    // Step # | name | target | value
    // 1 | open | /homepage/home |
    driver.get("http://localhost:9876/homepage/home");

    // 2 | setWindowSize | 1095x824 |
    driver.manage().window().setSize(new Dimension(1095, 824));

    // 3 | click | linkText=登录 |
    driver.findElement(By.linkText("登录")).click();

    // 4 | click | css=.login-input:nth-child(1) |
    driver.findElement(By.cssSelector(".login-input:nth-child(1)")).click();

    // 5 | type | css=.login-input:nth-child(1) | zqf
    driver.findElement(By.cssSelector(".login-input:nth-child(1)")).sendKeys("zqf");

    // 6 | click | css=.login-input:nth-child(3) |
    driver.findElement(By.cssSelector(".login-input:nth-child(3)")).click();

    // 7 | type | css=.login-input:nth-child(3) | 12345
    driver.findElement(By.cssSelector(".login-input:nth-child(3)")).sendKeys("12345");

    // 8 | click | css=.login-btn |
    driver.findElement(By.cssSelector(".login-btn")).click();

    // 9 | assertText | css=.input-check:nth-child(4) | 用户名或密码错误
```

```
    assertThat(driver.findElement(By.cssSelector(".input-check:nth-child(4)")).getText(), is("用户名或密码错误"));

    // 10 | close | |
    driver.close();
}

@Test
public void iTTD004003002() {
    // Test name: IT_TD_004_003_002
    // Step # | name | target | value
    // 1 | open | /homepage/home |
    driver.get("http://localhost:9876/homepage/home");
    // 2 | setWindowSize | 1095x790 |
    driver.manage().window().setSize(new Dimension(1095, 790));
    // 3 | click | linkText=登录 |
    driver.findElement(By.linkText("登录")).click();
    // 4 | click | css=.admin-login |
    driver.findElement(By.cssSelector(".admin-login")).click();
    // 5 | click | css=.login-input:nth-child(1) |
    driver.findElement(By.cssSelector(".login-input:nth-child(1)")).click();
    // 6 | type | css=.login-input:nth-child(1) | lwy
    driver.findElement(By.cssSelector(".login-input:nth-child(1)")).sendKeys("lwy");
    // 7 | click | css=.login-input:nth-child(3) |
    driver.findElement(By.cssSelector(".login-input:nth-child(3)")).click();
    // 8 | type | css=.login-input:nth-child(3) | 1234
    driver.findElement(By.cssSelector(".login-input:nth-child(3)")).sendKeys("1234");
    // 9 | click | css=.login-btn |
    driver.findElement(By.cssSelector(".login-btn")).click();
    // 10 | assertText | css=.input-check:nth-child(4) | 用户名或密码错误
    assertThat(driver.findElement(By.cssSelector(".input-check:nth-child(4)")).getText(), is("用户名或密码错误"));
    // 11 | close | |
```

```
driver.close();
}

@Test
public void iTTD006001001() {
    // Test name: IT_TD_006_001_001
    // Step # | name | target | value
    // 1 | open | /homepage/home |
    driver.get("http://localhost:9876/homepage/home");
    // 2 | setWindowSize | 1095x783 |
    driver.manage().window().setSize(new Dimension(1095, 783));
    // 3 | click | linkText=登录 |
    driver.findElement(By.linkText("登录")).click();
    // 4 | click | css=.login-input:nth-child(1) |
    driver.findElement(By.cssSelector(".login-input:nth-child(1)")).click();
    // 5 | type | css=.login-input:nth-child(1) | zldl
    driver.findElement(By.cssSelector(".login-input:nth-child(1)")).sendKeys("zldl");
    // 6 | click | css=.login-input:nth-child(3) |
    driver.findElement(By.cssSelector(".login-input:nth-child(3)")).click();
    // 7 | type | css=.login-input:nth-child(3) | 12345
    driver.findElement(By.cssSelector(".login-input:nth-child(3)")).sendKeys("12345");
    // 8 | click | css=.login-btn |
    driver.findElement(By.cssSelector(".login-btn")).click();
    // 9 | click | css=.section-banner-input |
    driver.findElement(By.cssSelector(".section-banner-input")).click();
    // 10 | type | css=.section-banner-input | 手机
    driver.findElement(By.cssSelector(".section-banner-input")).sendKeys("手机");
    // 11 | click | css=.section-banner-button |
    driver.findElement(By.cssSelector(".section-banner-button")).click();
    // 12 | runScript | window.scrollTo(0,686.4000244140625) |
    js.executeScript("window.scrollTo(0,686.4000244140625)");
    // 13 | click | css=.section-featured-card-main |
```

```

driver.findElement(By.cssSelector(".section-featured-card-main")).click();
// 14 | runScript | window.scrollTo(0,0) |
js.executeScript("window.scrollTo(0,0)");
// 15 | executeScript | return window.location.href | result
vars.put("result", js.executeScript("return window.location.href"));
// 16 | assert | result | http://localhost:9876/homepage/ItemDetail/4
assertEquals(vars.get("result").toString(), "http://localhost:9876/homepage/ItemDetail/4");
// 17 | close | |
driver.close();
}

```

@Test

```

public void iTTD007001001() {
    // Test name: IT_TD_007_001_001
    // Step # | name | target | value
    // 1 | open | /homepage/home |
    driver.get("http://localhost:9876/homepage/home");
    // 2 | setWindowSize | 1095x755 |
    driver.manage().window().setSize(new Dimension(1095, 755));
    // 3 | click | css=.section-banner-button |
    driver.findElement(By.cssSelector(".section-banner-button")).click();
    // 4 | executeScript | return window.location.href | result
    vars.put("result", js.executeScript("return window.location.href"));
    // 5 | assert | result | http://localhost:9876/homepage/home
    assertEquals(vars.get("result").toString(), "http://localhost:9876/homepage/home");
    // 6 | close | |
    driver.close();
}

```

@Test

```

public void iTTD007002001() {
    // Test name: IT_TD_007_002_001
    // Step # | name | target | value

```

```

// 1 | open | /homepage/home |
driver.get("http://localhost:9876/homepage/home");

// 2 | setWindowSize | 1095x755 |
driver.manage().window().setSize(new Dimension(1095, 755));

// 3 | click | css=.section-banner-input |
driver.findElement(By.cssSelector(".section-banner-input")).click();

// 4 | type | css=.section-banner-input | 手表
driver.findElement(By.cssSelector(".section-banner-input")).sendKeys("手表");

// 5 | click | css=.section-banner-button |
driver.findElement(By.cssSelector(".section-banner-button")).click();

// 6 | executeScript | return window.location.href | result
vars.put("result", js.executeScript("return window.location.href"));

// 7 | assert | result | http://localhost:9876/homepage/items?
mode=2&q=%E6%89%8B%E8%A1%A8
assertEquals(vars.get("result").toString(), "http://localhost:9876/homepage/items?
mode=2&q=%E6%89%8B%E8%A1%A8");

// 8 | close | |
driver.close();
}

@Test
public void iTTD007003001() {
    // Test name: IT_TD_007_003_001
    // Step # | name | target | value
    // 1 | open | /homepage/home |
    driver.get("http://localhost:9876/homepage/home");

    // 2 | setWindowSize | 1095x755 |
    driver.manage().window().setSize(new Dimension(1095, 755));

    // 3 | click | css=.section-banner-input |
    driver.findElement(By.cssSelector(".section-banner-input")).click();

    // 4 | type | css=.section-banner-input | 手表
    driver.findElement(By.cssSelector(".section-banner-input")).sendKeys("手表");

```



```

// 5 | click | css=.section-banner-button |
driver.findElement(By.cssSelector(".section-banner-button")).click();

// 6 | executeScript | return window.location.href | result
vars.put("result", js.executeScript("return window.location.href"));

// 7 | assert | result | http://localhost:9876/homepage/items?mode=2&q=%E6%89%8B%E8%A1%A8
assertEquals(vars.get("result").toString(), "http://localhost:9876/homepage/items?mode=2&q=%E6%89%8B%E8%A1%A8");

// 8 | close | |
driver.close();
}

@Test
public void iTTD008001001() {
    // Test name: IT_TD_008_001_001
    // Step # | name | target | value
    // 1 | open | /homepage/home |
    driver.get("http://localhost:9876/homepage/home");
    // 2 | setWindowSize | 1095x783 |
    driver.manage().window().setSize(new Dimension(1095, 783));
    // 3 | click | linkText=登录 |
    driver.findElement(By.linkText("登录")).click();
    // 4 | click | css=.login-input:nth-child(1) |
    driver.findElement(By.cssSelector(".login-input:nth-child(1)")).click();
    // 5 | type | css=.login-input:nth-child(1) | zldl
    driver.findElement(By.cssSelector(".login-input:nth-child(1)")).sendKeys("zldl");
    // 6 | click | css=.login-input:nth-child(3) |
    driver.findElement(By.cssSelector(".login-input:nth-child(3)")).click();
    // 7 | type | css=.login-input:nth-child(3) | 12345
    driver.findElement(By.cssSelector(".login-input:nth-child(3)")).sendKeys("12345");
    // 8 | click | css=.login-btn |
    driver.findElement(By.cssSelector(".login-btn")).click();

```

```

// 9 | click | css=.section-banner-input |
driver.findElement(By.cssSelector(".section-banner-input")).click();

// 10 | type | css=.section-banner-input | 相机
driver.findElement(By.cssSelector(".section-banner-input")).sendKeys("相机");

// 11 | click | css=.section-banner-button |
driver.findElement(By.cssSelector(".section-banner-button")).click();

// 12 | runScript | window.scrollTo(0,686.4000244140625) |
js.executeScript("window.scrollTo(0,686.4000244140625)");

// 13 | click | css=.commodity-card:nth-child(1) path |
driver.findElement(By.cssSelector(".commodity-card:nth-child(1) path")).click();

// 14 | assertText | css=.el-message__content | 收藏成功
assertThat(driver.findElement(By.cssSelector(".el-message__content")).getText(), is("收藏成功"));

// 15 | close | |
driver.close();
}

@Test
public void iTTD009001001() {
    // Test name: IT_TD_009_001_001
    // Step # | name | target | value
    // 1 | open | /homepage/home |
    driver.get("http://localhost:9876/homepage/home");
    // 2 | setWindowSize | 1142x824 |
    driver.manage().window().setSize(new Dimension(1142, 824));
    // 3 | click | linkText=登录 |
    driver.findElement(By.linkText("登录")).click();
    // 4 | click | css=.login-input:nth-child(1) |
    driver.findElement(By.cssSelector(".login-input:nth-child(1)")).click();
    // 5 | type | css=.login-input:nth-child(1) | zldl
    driver.findElement(By.cssSelector(".login-input:nth-child(1)")).sendKeys("zldl");
    // 6 | click | css=.login-input:nth-child(3) |

```

```
driver.findElement(By.cssSelector(".login-input:nth-child(3))).click();
// 7 | type | css=.login-input:nth-child(3) | 12345
driver.findElement(By.cssSelector(".login-input:nth-child(3))).sendKeys("12345");
// 8 | click | css=.login-btn |
driver.findElement(By.cssSelector(".login-btn")).click();
// 9 | click | linkText=个人主页 |
driver.findElement(By.linkText("个人主页")).click();
// 10 | click | linkText=收藏夹 |
driver.findElement(By.linkText("收藏夹")).click();
// 11 | click | css=.commodity-card:nth-child(2) path |
driver.findElement(By.cssSelector(".commodity-card:nth-child(2) path")).click();
// 12 | close | |
driver.close();
}

@Test
public void iTTD005001001() {
    // Test name: IT_TD_005_001_001
    // Step # | name | target | value
    // 1 | open | /homepage/home |
    driver.get("http://localhost:9876/homepage/home");
    // 2 | setWindowSize | 1142x824 |
    driver.manage().window().setSize(new Dimension(1142, 824));
    // 3 | click | linkText=登录 |
    driver.findElement(By.linkText("登录")).click();
    // 4 | click | css=.register |
    driver.findElement(By.cssSelector(".register")).click();
    // 5 | click | css=.register-input:nth-child(3) |
    driver.findElement(By.cssSelector(".register-input:nth-child(3))).click();
    // 6 | type | css=.register-input:nth-child(3) | 12345
    driver.findElement(By.cssSelector(".register-input:nth-child(3))).sendKeys("12345");
    // 7 | click | css=.register-input:nth-child(5) |
```

```

driver.findElement(By.cssSelector(".register-input:nth-child(5)")).click();
// 8 | type | css=.register-input:nth-child(5) | 12345
driver.findElement(By.cssSelector(".register-input:nth-child(5)")).sendKeys("12345");
// 9 | click | css=.login-btn |
driver.findElement(By.cssSelector(".login-btn")).click();
// 10 | assertText | css=.input-check:nth-child(2) | 用户名不能为空
assertThat(driver.findElement(By.cssSelector(".input-check:nth-child(2)")).getText(), is("用户名不能为空"));
// 11 | close | |
driver.close();
}

@Test
public void iTTD005002001() {
// Test name: IT_TD_005_002_001
// Step # | name | target | value
// 1 | open | /homepage/home |
driver.get("http://localhost:9876/homepage/home");
// 2 | setWindowSize | 1142x824 |
driver.manage().window().setSize(new Dimension(1142, 824));
// 3 | click | linkText=登录 |
driver.findElement(By.linkText("登录")).click();
// 4 | click | css=.register |
driver.findElement(By.cssSelector(".register")).click();
// 5 | click | css=.register-input:nth-child(1) |
driver.findElement(By.cssSelector(".register-input:nth-child(1)")).click();
// 6 | type | css=.register-input:nth-child(1) | reportIT3
driver.findElement(By.cssSelector(".register-input:nth-child(1)")).sendKeys("reportIT3");
// 7 | click | css=.register-input:nth-child(3) |
driver.findElement(By.cssSelector(".register-input:nth-child(3)")).click();
// 8 | type | css=.register-input:nth-child(3) | 12345
driver.findElement(By.cssSelector(".register-input:nth-child(3)")).sendKeys("12345");

```

```
// 9 | click | css=.register-input:nth-child(5) |  
driver.findElement(By.cssSelector(".register-input:nth-child(5)")).click();  
// 10 | type | css=.register-input:nth-child(5) | 12345  
driver.findElement(By.cssSelector(".register-input:nth-child(5)")).sendKeys("12345");  
// 11 | click | css=.login-btn |  
driver.findElement(By.cssSelector(".login-btn")).click();  
// 12 | close | |  
driver.close();  
}
```

@Test

```
public void iTTD005003001() {  
    // Test name: IT_TD_005_003_001  
    // Step # | name | target | value  
    // 1 | open | /homepage/home |  
    driver.get("http://localhost:9876/homepage/home");  
    // 2 | setWindowSize | 1142x824 |  
    driver.manage().window().setSize(new Dimension(1142, 824));  
    // 3 | click | linkText=登录 |  
    driver.findElement(By.linkText("登录")).click();  
    // 4 | click | css=.register |  
    driver.findElement(By.cssSelector(".register")).click();  
    // 5 | click | css=.register-input:nth-child(1) |  
    driver.findElement(By.cssSelector(".register-input:nth-child(1)")).click();  
    // 6 | type | css=.register-input:nth-child(1) | zldl  
    driver.findElement(By.cssSelector(".register-input:nth-child(1)")).sendKeys("zldl");  
    // 7 | click | css=.register-input:nth-child(3) |  
    driver.findElement(By.cssSelector(".register-input:nth-child(3)")).click();  
    // 8 | type | css=.register-input:nth-child(3) | 12345  
    driver.findElement(By.cssSelector(".register-input:nth-child(3)")).sendKeys("12345");  
    // 9 | click | css=.register-input:nth-child(5) |  
    driver.findElement(By.cssSelector(".register-input:nth-child(5)")).click();  
}
```

```
// 10 | type | css=.register-input:nth-child(5) | 12345
driver.findElement(By.cssSelector(".register-input:nth-child(5)")).sendKeys("12345");

// 11 | click | css=.login-btn |
driver.findElement(By.cssSelector(".login-btn")).click();

// 12 | assertText | css=.input-check:nth-child(2) | 用户名重复
assertThat(driver.findElement(By.cssSelector(".input-check:nth-child(2)")).getText(), is("用户名重复"));

// 13 | close | |
driver.close();
}
}
```

## 15.2 测试结果

### 15.2.1 集成测试第一版

▼ Integration-test

✓ IT\_TD\_001\_001\_001

✓ IT\_TD\_001\_001\_002

✓ IT\_TD\_001\_001\_003

✓ IT\_TD\_001\_002\_001

✓ IT\_TD\_001\_003\_001

✓ IT\_TD\_001\_004\_001

✓ IT\_TD\_001\_004\_002

✓ IT\_TD\_001\_005\_001

✓ IT\_TD\_001\_006\_001

✗ IT\_TD\_002\_001\_001

✗ IT\_TD\_002\_001\_002



✗ IT\_TD\_002\_002\_001

✗ IT\_TD\_002\_003\_001

✓ IT\_TD\_003\_001\_001

✓ IT\_TD\_003\_002\_001

✓ IT\_TD\_003\_003\_001

✓ IT\_TD\_004\_001\_001

✓ IT\_TD\_004\_001\_002

✓ IT\_TD\_004\_001\_003

✓ IT\_TD\_004\_002\_001

✓ IT\_TD\_004\_002\_002

✓ IT\_TD\_004\_003\_001

✗ IT\_TD\_004\_003\_002

✗ IT\_TD\_005\_001\_001

✗ IT\_TD\_005\_002\_001

✗ IT\_TD\_005\_003\_001

✓ IT\_TD\_006\_001\_001



✓ IT\_TD\_007\_001\_001

✓ IT\_TD\_007\_002\_001

✓ IT\_TD\_007\_003\_001

✓ IT\_TD\_008\_001\_001

✗ IT\_TD\_009\_001\_001

15.2.2 集成测试后回归测试

▼	Integration-test
✓	IT_TD_001_001_001
✓	IT_TD_001_001_002
✓	IT_TD_001_001_003
✓	IT_TD_001_002_001
✓	IT_TD_001_003_001
✓	IT_TD_001_004_001
✓	IT_TD_001_004_002
✓	IT_TD_001_005_001
✓	IT_TD_001_006_001
✓	IT_TD_002_001_001
✓	IT_TD_002_001_002
✓	IT_TD_002_002_001
✓	IT_TD_002_003_001
✓	IT_TD_003_001_001
✓	IT_TD_003_002_001
✓	IT_TD_003_003_001
✓	IT_TD_004_001_001
✓	IT_TD_004_001_002
✓	IT_TD_004_001_003
✓	IT_TD_004_002_001
✓	IT_TD_004_002_002
✓	IT_TD_004_003_001
✓	IT_TD_004_003_002
✓	IT_TD_006_001_001
✓	IT_TD_007_001_001
✓	IT_TD_007_002_001
✓	IT_TD_007_003_001
✓	IT_TD_008_001_001
✓	IT_TD_009_001_001

15.3 以IT\_TD\_003\_001\_001为例



Running 'IT\_TD\_003\_001\_001'

1. open on /homepage/home OK
2. setWindowSize on 1095x755 OK
3. click on css=.section-banner-button OK
4. executeScript on return window.location.href with value result OK
5. assert on result with value http://localhost:9876/homepage/home OK
6. close OK

'IT\_TD\_003\_001\_001' completed successfully

## 4.系统测试

### 1 引言

#### 1.1 目的

为了明确用户的需求并且较好地与开发人员进行沟通，使用户与开发人员双方对软件需求取得共同理解的基础上达成协议，特编写此文档，并且作为整个软件开发的基础。

#### 1.2 背景

本项目的开发是应软测小组的要求，开发的一套在线租物平台，以满足大学生闲置物品交流的需求。减少学生的经济负担，同时减少资源的浪费。

#### 1.3 参考资料

文档编写标准：GB99999\_1999

《计算机软件需求说明编制指南》（GB 9385-88）

《计算机软件开发文件指南》（GB 8567-88）

《校园租赁系统需求文档》

#### 1.4 术语

- 信誉度：

指用户在系统当中行为展现出来的信用程度，信誉度越高表示信用的水平越高，信誉度越低表示信用的水平越低。如果信誉度高的话，我们会提供相应程度的免押金额度。

- 租物：

指用户将自己闲置的物品上传到系统当中以供他人租用。

- 租金：

指物品固有的属性，我们的租金是以每日的租金乘以租用的时间来计算的。

- 押金：

指用户在上传物品的时候选择的押金，物品的价值越高，押金的金额也相应地越高。

- 账号

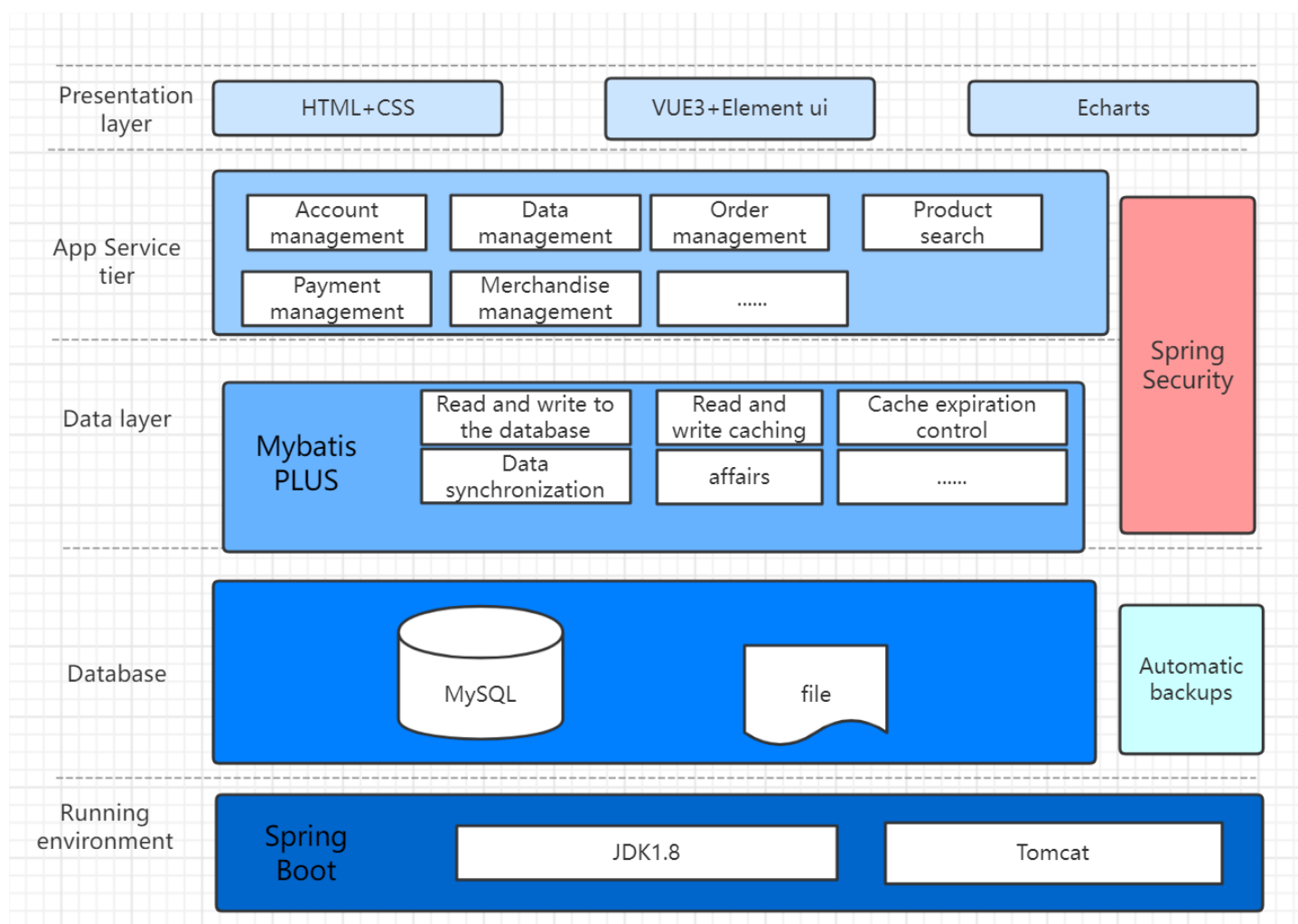
在我们系统当中，主要有两种用户角色，一种是管理员，负责物品上架之后的审核操作。另外一种 是普通用户，在系统当中进行物品上架，下单等操作。

## 2 项目概述

### 2.1 开发软件的一般描述

这个项目的开发是为了给大学生提供一个简单易用的在线交易平台，为学校内的学生上传闲置物品，租用他人的闲置物品等提供便利。可以划分为两个子系统，一个是普通用户的在线商城系统，可以使用本系统上架物品，下单物品，支付，发布需求，查看需求等。另外一个 是管理员的后台管理页面，可以查看用户新上架的物品，并且对物品进行一个审核的操作。

### 2.2 开发软件的功能描述



## 2.3实现语言

主要使用Java Springboot技术实现的后端，主要使用css,html，vue.js实现的前端页面的创建。

## 2.4 用户特点

本软件的用户主要是大学内的学生，大多数学生都具有购物平台的使用经验，所以我们采用了类似于现有的购物商城的购物页面，学生可以很自然地学会我们系统的使用。我们根据Fiit's法则，使用了隐喻的技巧，让用户的体验更加自然。另外一部分用户是我们的系统的管理员，管理员在界面里边应该要能够直接看到需要审核的物品，并且点击按钮即可实现物品的审核。

## 2.5一般约束

身份约束：用户只有当身份是管理员的时候才能登录并进入管理员的页面。

业务约束：考虑到业务的特殊性，用户上架商品的时候，只能同时上架一件物品。

时间约束：用户下单的时候，时间只能够选择当前时间之后的时间。

服务器约束：由于存储图片的服务器带宽和容量有限，所以用户上传的图片大小不能超过20m。

# 3 需求说明

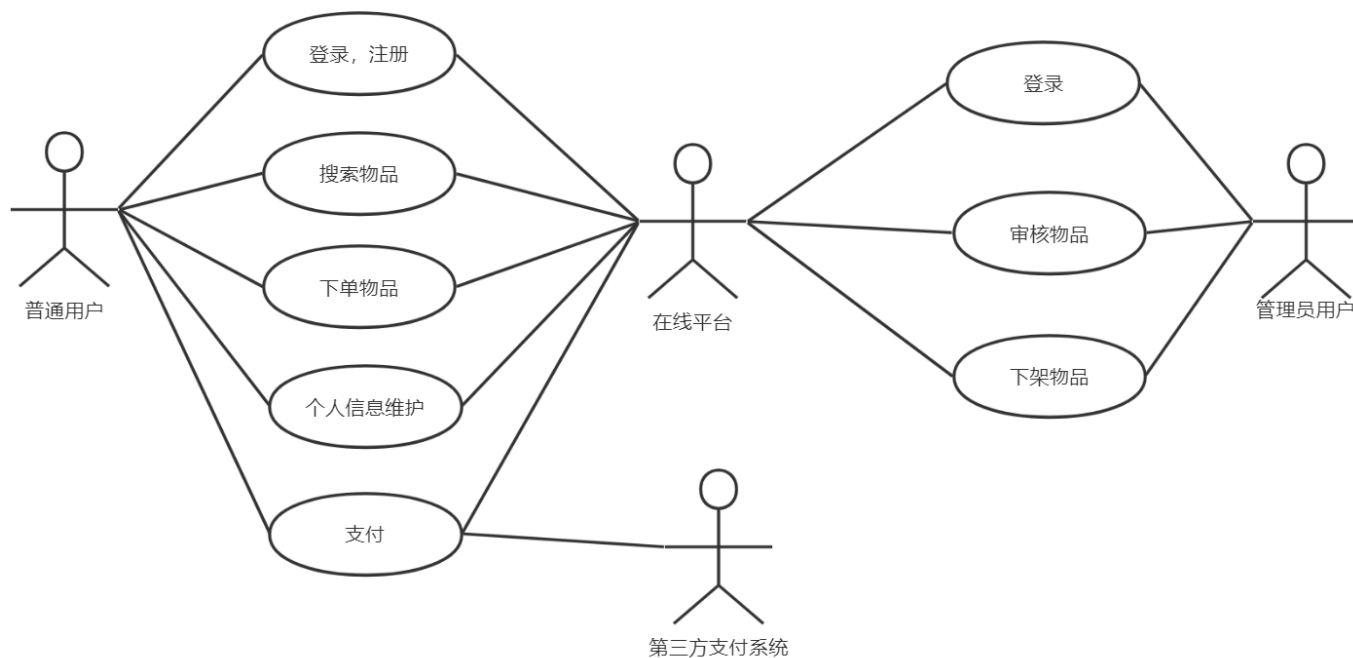
## 3.1 基本描述

用户可以以游客的身份在主页进行浏览的操作，这是购物网站的常用策略。防止用户没有账号就不能查看任何信息。当游客看到某件想要的物品时，可以点击卡片，如果没有登录的话，就会跳转到登录页面。若没有账号则可以选择一个注册的操作。

进入主页之后，可以对物品进行一个搜索的操作，点击物品，可以看到物品详情，包括物品的评分等信息。

之后可以进行下单的操作。

由于我们目前面向的用户是同济大学的学生，所以用户在下单的时候需要选择校区，以及需要的时间。之后点击支付按钮应该能正确跳转到第三方支付的面。



大学生在线租赁系统工作示意图

### 3.1.1 用户信息

用户信包括两个部分：普通用户信息和管理员信息

普通用户信息：用户id，用户名，用户头像，用户性别,邮箱，电话，地址，信誉度。

管理员用户信息: 用户id，用户名，用户头像，用户性别。

### 3.1.2 订单信息

物品信息：物品信息包括物品的归属，物品的名称，物品的状态，物品的押金，物品的标签，物品的新旧程度，物品的每日租金。

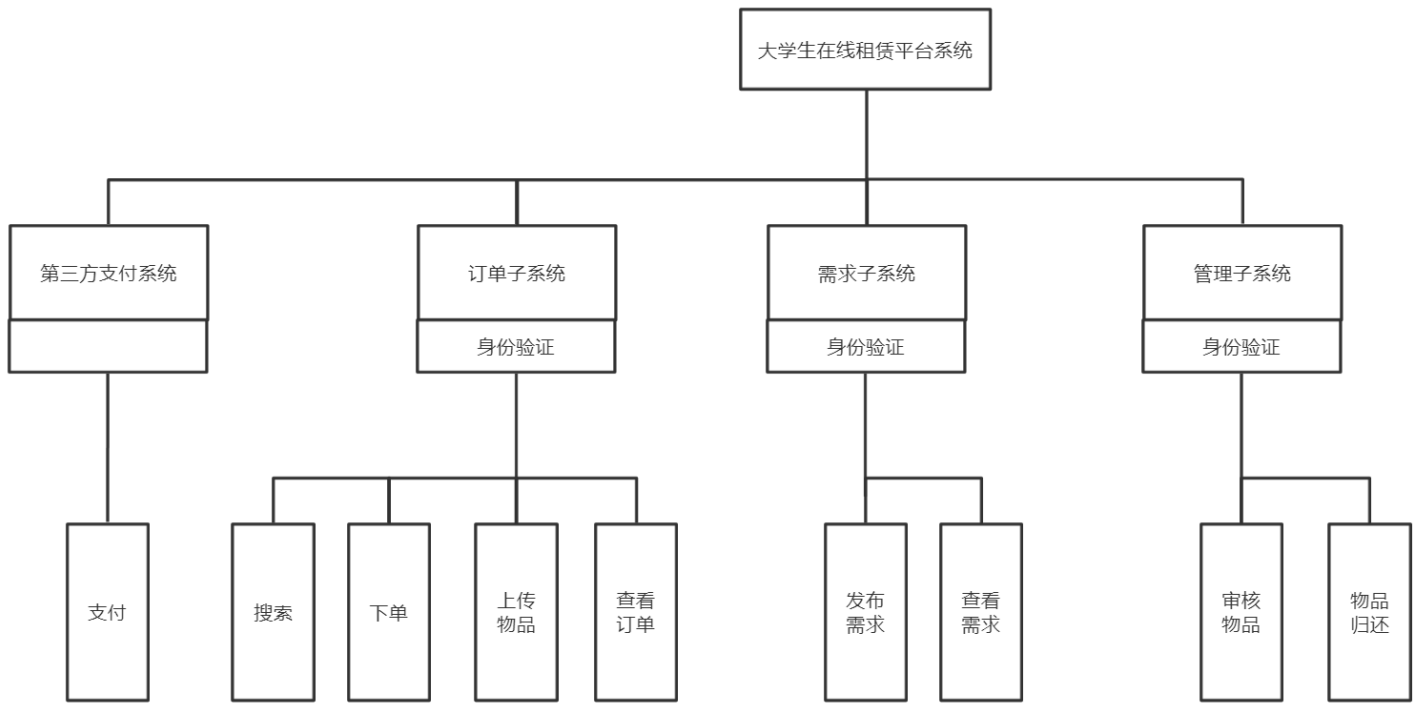
订单信息：物品的id，取货的地点，订单的持续时间，订单的总金额、订单的状态。

### 3.1.3 需求信息

需求信息：需求的发起人，需求的内容，期待的价格。

## 3.2 功能需求

根据本系统的需求，现在对本系统做了以下的功能设计，再给出了基本的框架之后，我们会逐一介绍功能。根据功能的不同点，我们将系统分为了以下三个子系统，每个子系统都包含了不同的功能。大学生在线租赁系统如下图所示



3.2.1 购物子系统

功能需求1：

描述:用户打开网站的时候，显示搜索主页

输入：无

处理：网站显示搜索主页。

输出：显示搜索主页

功能描述2：

描述：用户点击搜索栏，输入物品信息

输入：物品名称

处理：在数据库里边查找符合条件的物品

输出：搜索得到的物品

功能描述3：

描述：普通用户登录

输入：用户名、密码

处理：检验密码是否正确

输出：不正确要求用户重新输入

功能描述4：

描述：用户账号密码匹配之后，进入网站主页

输入：用户输入正确密码

处理：校验密码

输出：跳转到网页主页

功能描述5：

描述：用户点击物品卡片，跳转到物品详情页面

输入：用户点击卡片

处理：路由跳转

输出：跳转到物品详情页面

功能描述6：

描述：用户点击收藏按钮，显示物品已收藏

输入：点击收藏按钮

处理：将物品添加到收藏夹里边

输出：无

功能描述7：

描述：用户再次点击收藏按钮，可以取消物品的收藏

输入：点击收藏按钮

处理：将物品从收藏夹内删去

输出：无

功能描述8：

描述：用户在网站主页点击分类按钮

输入：点击分类按钮

处理：跳转到分类子页面

输出：跳转到分类子页面

#### 功能描述9：

描述：用户点击个人主页

输入：点击个人主页

处理：跳转到个人主页

输出：跳转到个人主页

#### 功能描述10：

描述：修改用户资料

输入：输入新的账户信息

处理：判断输入的电话号码，邮箱是否符合规范

输出：如果符合规范，则返回修改成功的信息。

#### 功能描述11：

描述：修改用户账户信息

输入：点击修改密码按钮

处理：判断当前密码是否正确，之后判断输入的两次新密码是否相等。

输出：如果符合要求，调用接口进行修改。

#### 功能描述12

描述：查看自己的历史订单

输入：点击历史订单按钮

处理：调用查找历史订单的接口

输出：跳转到历史订单页面

#### 功能描述13

描述：下单物品

输入：用户所在的校区，需要物品的时间

处理：根据用户的信誉度计算用户的押金，之后根据物品的每日租金和租用的总日期计算用户的总租金

输出：跳转到第三方支付系统页面

#### 功能描述14

描述：取消支付

输入：对物品下单之后，点击取消按钮

处理：将订单修改为已取消状态

输出：无

#### 功能描述15

描述：跳过支付

输入：对物品下单之后，点击跳过支付

处理：将订单状态改为待支付状态

输出：无

#### 功能描述16

描述：支付

输入：点击支付按钮，跳转到第三方支付页面

处理：调用第三方支付接口

输出：输出计算完的总金额

#### 功能描述17

描述：上架闲置物品

输入：输入物品的名称、选择物品的种类、上传物品的照片，选择物品的每日租金和押金，填写物品的新旧程度和详细描述

处理：将物品信息保存在数据库当中，把物品的状态设置为待审核状态

输出：无

### 3.2.2 需求子系统



#### 功能描述1

描述：用户上传自己的需求

输入：用户输入物品名称，需求详情和期望租金

处理：将需求存储到数据库当中

输出：无

#### 功能描述2

描述：查看需求广场

输入：用户选择需求广场界面

处理：跳转并显示需求广场内的其他人需求

输出：页面跳转到需求广场

#### 功能描述3

描述：查看自己发布的需求

输入：点击自己的需求

处理：跳转到自己需求页面

输出：跳转到自己需求的页面

#### 功能描述4：

描述：删除自己的历史需求

输入：点击删除按钮

处理：将历史需求在数据库当中删除

输出：返回删除操作的结果

### 3.2.2 管理子系统

#### 功能描述1：

描述：审核通过物品

输入：点击物品卡片下方的通过按钮

处理：将物品的状态由待审核修改为审核通过状态

输出：将物品成功上架

功能描述2：

描述：下架物品

输入：点击物品卡片下方的审核不通过按钮

处理：将物品从数据库当中删除

输出：删除的结果

### 3.3 性能及其他需求

在性能方面，由于本项目是在线电商类的平台，所以我们需要满足高并发的要求，因此我们使用了Jmeter工具来进行了压力测试。下面是我们得到的部分结果

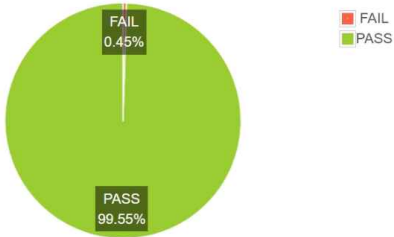
Test and Report information

Source file	"pressureTest.jtl"
Start Time	"6/19/22, 9:09 AM"
End Time	"6/19/22, 9:16 AM"
Filter for display	""

APDEX (Application Performance Index)

Apdex	T (Toleration threshold)	F (Frustration threshold)	Label
0.858	500 ms	1 sec 500 ms	Total
0.842	500 ms	1 sec 500 ms	查询测试
0.857	500 ms	1 sec 500 ms	发帖测试
0.874	500 ms	1 sec 500 ms	登录测试

Requests Summary



Statistics

Requests	Executions			Response Times (ms)							Throughput	Network (KB/sec)	
	#Samples	FAIL	Error %	Average	Min	Max	Median	90th pct	95th pct	99th pct	Transactions/s	Received	Sent
Total	4200	19	0.45%	1072.52	41	183656	164.00	1712.80	3886.60	17457.62	9.60	4.66	2.26
发帖测试	1400	3	0.21%	977.02	89	77728	179.00	1764.40	4105.55	15504.69	3.23	0.84	0.89
查询测试	1400	9	0.64%	1313.45	84	135960	179.00	2050.20	4310.70	29997.22	3.21	2.41	0.68
登录测试	1400	7	0.50%	927.08	41	183656	118.00	1484.90	2582.90	18820.67	3.20	1.43	0.71

Errors

Type of error	Number of errors	% in errors	% in all samples
500	19	100.00%	0.45%

Top 5 Errors by sampler

Sample	#Samples	#Errors	Error	#Errors	Error	#Errors	Error	#Errors	Error	#Errors	Error	#Errors
Total	4200	19	500	19								
发帖测试	1400	3	500	3								
查询测试	1400	9	500	9								
登录测试	1400	7	500	7								

我们对于主要的发布需求，查询物品以及登录等接口测试了并发，发现能够正常承载1400人左右的并发。

- 在下单的时候，选择的时间必须为当日及之后的日期，归还日期必须在期望日期之后

- 在支付的时候，点击取消订单状态转为待支付状态
- 在支付的时候，点击支付，订单状态变为待归还状态
- 归还之后，订单变为待评价状态。
- 订单支付之后信誉度加5
- 交易完成之后，用户可以点击退出按钮退出系统

### 3.4 对输入输出的规定

密码：由用户设置的一个6位整数

上传图片：只支持20m以下的图片

下单地址：通过单选框进行选择，目前仅提供嘉定和四平两个校区

期望日期：下单的时候选择的日期必须要是当日及当日之后的日期

响应时间：3s以内

## 4.采用的测试方法

系统从以下几个方面来进行测试分析并且结合具体的用例设计技术（主要是黑盒测试技术）设计测试用例

- 系统级功能；
- 系统级业务流；
- 系统级别的接口；
- 系统级别的输入和输出；
- 系统级别的状态转换；
- 系统级别数据；
- 系统级别非功能；

## 5.测试环境

(1) 硬件需求

- 笔记本一台
- 服务器一台

(2) 软件需求

- windows10操作系统
- edge浏览器

- mysql数据库
- centos7系统

(3) 网络需求

100mb以上的带框

6.测试对象

大学生在线租赁平台

软件版本：集成测试后回归测试第一版

7.测试分析与用例设计

(1) 测试数据准备，如下表所示

用户名	密码	角色
zqf	123456	普通用户
mjy	12345	管理员
xdx	12345	普通用户

(2) 基于系统级别功能的测试分析与用例设计

基于系统级别功能的测试分析需要在系统层面上考虑正常功能和非正常功能。覆盖率指标要求是正常功能和非正常功能全覆盖，正常功能和非正常功能主要是根据需求导出的，包括显式的和隐含的需求。

用例编号	EZR_Sys_Test_case_fun_001				
测试覆盖的系统功能	普通用户正常登陆（覆盖正常功能）				
用例设计方法	场景法				
前置条件	无				
输入					
初始输入	预期交互 输入、输出1	预期交互 输入、输出2	预期交互 输入、输出3	预期交互 输入、输出4	预期交互 输入、输出5

用户点击登录页面	用户输入用户名“zqf”,用户输入密码”123456“	用户正常登录，进入网站主页	用户点击管理登录，输入账户名“mjy”，输入密码”12345”，成功进入管理员页面	无
最后预期输出				
登录成功，进入主界面				

用例编号	EZR_Sys_Test_case_fun_002			
测试覆盖的系统功能	用户输入不匹配的账号密码（覆盖不正常功能）			
用例设计方法	错误推测法，场景法			
前置条件	无			
输入				
初始输入	预期交互 输入、输出1	预期交互 输入、输出2	预期交互 输入、输出3	预期交互 输入、输出4
用户点击登录页面	用户输入账户名"zqf",输入密码“54321”,系统提示账号密码不匹配	用户不输入账户名，输入密码“1234”，系统提示用户名不能为空	用户输入账户名为“zqf”，不输入密码，系统提示用户密码不能为空	用户点击管理员登录，输入账户名“zqf”，密码”123456“,提示账号密码不匹配
最后预期输出				
登陆失败，显示登录失败信息				

用例编号	EZR_Sys_Test_case_fun_003			
测试覆盖的系统功能	用户搜索物品（覆盖正常功能和不正常功能）			
用例设计方法	场景法，需求规约导出法综合			

前置条件	用户进入搜索页面			
输入				
初始输入	预期交互 输入、输出1	预期交互 输入、输出2	预期交互 输入、输出3	预期交互 输入、输出4
无	用户在搜索框输入”耳机”，系统正确返回所有和耳机相关的物品	用户未输入任何信息，点击搜索按钮，系统提示“请输入要搜索的物品！ ”	无	无
最后预期输出				
显示搜索结果				

用例编号	EZR_Sys_Test_case_fun_004			
测试覆盖的系统功能	用户点击卡片跳转到卡片标签页面（覆盖正常功能）			
用例设计方法	场景法，需求规约导出法综合			
前置条件	用户已成功登入			
输入				
初始输入	预期交互 输入、输出1	预期交互 输入、输出2	预期交互 输入、输出3	预期交互 输入、输出4
无	用户点击物品标签，之后系统正常跳转到标签页面	无	无	无
最后预期输出				
系统跳转到标签页面				

用例编号	EZR_Sys_Test_case_fun_005			
测试覆盖的系统功能	用户收藏和取消收藏物品（覆盖正常和非正常功能）			

用例设计方法	场景法，错误推测法			
前置条件	用户已登录			
输入				
初始输入	预期交互 输入、输出1	预期交互 输入、输出2	预期交互 输入、输出3	预期交互 输入、输出4
用户点击收藏按钮	用户对一个未收藏的物品点击收藏按钮，物品的收藏按钮变红，用户的收藏夹当中添加该物品	用户对一个已收藏的物品点击收藏按钮，系统提示该物品已被收藏。	用户打开收藏夹界面，对收藏夹当中的物品点击取消收藏，之后物品会从收藏夹里边删除	无

bug:RJC-3用户可以重复收藏同一个物品，RJC-13未登录时可以进行收藏操作，RJC-17已收藏的物品没有标红，RJC-31不能取消收藏，已解决

用例编号	EZR_Sys_Test_case_fun_006			
测试覆盖的系统功能	用户修改个人资料(覆盖正常和非正常功能)			
用例设计方法	场景法，错误推测法、需求规约导出法的综合			
前置条件	用户以普通用户的身份正常登录			
输入				
初始输入	预期交互 输入、输出1	预期交互 输入、输出2	预期交互 输入、输出3	预期交互 输入、输出4
用户使用”zqf”的账号登录	用户点击 编辑按钮，将自己的电话号码改为14413452563，邮箱地址改为zqf@163.com。系统正确修改了个人信息	用户点击编辑按钮，将自己的电话号码改为1234，系统提示”电话号码不符合规范"	用户点击编辑按钮，将自己的邮箱修改为1234@moc.361,系统提示邮箱格式错误	无



bug:RJC-18,RJC-21,用户性别、电话、邮箱可以随意输入，RJC-24电话无法修改为超过10位的号码，已解决

用例编号	EZR_Sys_Test_case_fun_007			
测试覆盖的系统功能	用户修改密码(覆盖正常和非正常功能)			
用例设计方法	场景法，错误推测法、需求规约导出法的综合			
前置条件	用户以普通用户的身份登录			
输入				
初始输入	预期交互 输入、输出1	预期交互 输入、输出2	预期交互 输入、输出3	预期交互 输入、输出4
用户使用“zqf”账号登录，并进入修改密码页面	用户点击修改密码，输入自己的历史密码为” 123456 “，连续输入两次新密码” 12345 “，系统提示修改成功，密码修改成为了123456	用户点击修改密码，输入自己的历史密码为” 123",系统系统历史密码错误。	用户点击修改密码，输入自己历史密码为” 123456 “，第一次输入新密码1234，第二次输入新密码123，系统提示两次输入的密码不匹配	无

用例编号	EZR_Sys_Test_case_fun_008			
测试覆盖的系统功能	用户下单物品(包含正常功能)			
用例设计方法	场景法			
前置条件	用户以普通用户正常登陆			
输入				
初始输入	预期交互	预期交互	预期交互	预期交互

	输入、输出1	输入、输出2	输入、输出3	输入、输出4
用户以“zqf”身份登录，选择押金为500，每日租金为30的物品，选择从7月1日租用到7月5日，总共4天，点击下载按钮	点击取消订单，物品状态变回可出租状态，在用户的历史订单当中添加一条取消了的订单的记录	点击跳过按钮，物品状态仍然保持已出租状态，在用户的历史订单页面添加一条待支付的记录。	用户点击支付按钮，跳转到第三方支付界面。支付之后变成待归还状态，	无
最后预期输出				
用户成功下单，历史订单页面新增对应状态的新订单记录				

用例编号	EZR_Sys_Test_case_fun_009			
测试覆盖的系统功能	用户下单物品（包含不正常功能）			
用例设计方法	错误推测法			
前置条件	用户以普通用户正常登陆			
输入				
初始输入	预期交互 输入、输出1	预期交互 输入、输出2	预期交互 输入、输出3	预期交互 输入、输出4
用户以“zqf”身份登录，选择押金为500，每日租金为30的物品	用户选择预期日期为5月2号，系统提示”不可用的时间 “	用户选择归还日期为5月2日，系统提示”不可用的时间 “	无	无
最后预期输出				
用户无法下单				

bug:RJC-23用户可以租用自己的物品，RJC-25可以选择今天之前的日期下单，已解决

用例编号	EZR_Sys_Test_case_fun_010			
测试覆盖的系统功能	用户上架物品功能（包含正常功能）			
用例设计方法	场景法			
前置条件	用户以普通用户身份正常登录			
输入				
初始输入	预期交互 输入、输出1	预期交互 输入、输出2	预期交互 输入、输出3	预期交互 输入、输出4
用户以“zqf”身份登录，进入上传物品的页面	用户上传一张大小为10m的图像，输入物品的标题为“电脑”，点击标签为电子产品，输入押金为1000元，输入每日租金100元。结果正常上传了物品。		无	无
最后预期输出				
用户成功上架物品				

用例编号	EZR_Sys_Test_case_fun_011			
测试覆盖的系统功能	用户上架物品（包含不正常功能）			
用例设计方法	错误推测法			
前置条件	用户以普通用户正常登陆			
输入				
初始输入	预期交互	预期交互	预期交互	预期交互

	输入、输出1	输入、输出2	输入、输出3	输入、输出4
用户以“zqf”身份登录，进入上传物品的页面	用户上传大小为100m的图片，系统表示上传失败	用户输入预期的价格为-5，系统自动将价格改为0	用户输入每日租金为-5，系统自动将每日租金改为0.	无

bug:RJC-26预期租金可以不输入，已解决

用例编号	EZR_Sys_Test_case_fun_012			
测试覆盖的系统功能	用户在网站主页点击分类按钮（覆盖正常功能）			
用例设计方法	场景法，需求规约导出法综合			
前置条件	用户已成功登录			
输入				
初始输入	预期交互 输入、输出1	预期交互 输入、输出2	预期交互 输入、输出3	预期交互 输入、输出4
用户以“zqf”身份登录，进入主界面	用户点击分类按钮，之后系统正常跳转到分类子页面	无	无	无
最后预期输出				
系统跳转至分类子界面				

用例编号	EZR_Sys_Test_case_fun_00
测试覆盖的系统功能	查看自己的历史订单（覆盖正常功能）

用例设计方法	场景法，需求规约导出法综合			
前置条件	用户已成功登录			
输入				
初始输入	预期交互 输入、输出1	预期交互 输入、输出2	预期交互 输入、输出3	预期交互 输入、输出4
用户以“zqf”身份登录，点击“个人主页-订单中心”	用户点击“订单中心”按钮，之后系统正常跳转到订单中心页面，展示用户历史订单	无	无	无
最后预期输出				
展示用户历史订单				

用例编号	EZR_Sys_Test_case_fun_010			
测试覆盖的系统功能	用户上传自己的需求（包含正常功能）			
用例设计方法	场景法			
前置条件	用户以普通用户身份正常登录			
输入				
初始输入	预期交互 输入、输出1	预期交互 输入、输出2	预期交互 输入、输出3	预期交互 输入、输出4
用户以“zqf”身份登录，进入需求中心页面，点击：发布需求“按钮	用户输入所需物品的标题为“电脑”，输入所需物品描述为“需要一台mac电脑”，输入预期租金价格为20元。结果正常上传了需求。		无	无

最后预期输出				
用户成功发布需求				

bug:RJC-1.RJC-4,无法正常上传需求，已解决

用例编号	EZR_Sys_Test_case_fun_011			
测试覆盖的系统功能	用户上传自己的需求（包含不正常功能）			
用例设计方法	错误推测法			
前置条件	用户以普通用户正常登陆			
输入				
初始输入	预期交互 输入、输出1	预期交互 输入、输出2	预期交互 输入、输出3	预期交互 输入、输出4
用户以“zqf”身份登录，进入需求中心页面，点击：发布需求“按钮	用户输入预期租金价格为-5，系统自动将价格改为1	用户输入所需物品的标题为“电脑”，不输入所需物品描述，系统提示请输入所需物品描述	用户所需物品描述为“需要一台mac电脑”，不输入所需物品的标题，系统提示请输入所需物品描述	无

用例编号	EZR_Sys_Test_case_fun_00
测试覆盖的系统功能	查看需求广场（覆盖正常功能）
用例设计方法	场景法，需求规约导出法综合
前置条件	用户已成功登录

输入				
初始输入	预期交互 输入、输出1	预期交互 输入、输出2	预期交互 输入、输出3	预期交互 输入、输出4
用户以“zqf”身份登录，进入需求中心页面，进入需求广场	用户点击“需求中心”按钮，之后系统正常跳转到需求广场页面，展示所有已发布需求	无	无	无
最后预期输出				
展示所有已发布需求				

bug:RJC-14排版问题，已解决

用例编号	EZR_Sys_Test_case_fun_00			
测试覆盖的系统功能	查看自己发布的需求（覆盖正常功能）			
用例设计方法	场景法，需求规约导出法综合			
前置条件	用户已成功登录			
输入				
初始输入	预期交互 输入、输出1	预期交互 输入、输出2	预期交互 输入、输出3	预期交互 输入、输出4
用户以“zqf”身份登录，进入需求中心页面，点击已发布的需求模块	用户点击“已发布的需求”按钮，之后系统正常跳转到已发布的需求页面，展示用户自己已发布的需求	无	无	无
最后预期输出				
展示用户自己已发布的需求				

bug:RJC-9无法获取用户发布的需求，RJC-23未发布需求时查看个人发布的需求时出错，已解决

用例编号	EZR_Sys_Test_case_fun_00			
测试覆盖的系统功能	删除自己的历史需求（覆盖正常功能）			
用例设计方法	场景法，需求规约导出法综合			
前置条件	用户已成功登录			
输入				
初始输入	预期交互 输入、输出1	预期交互 输入、输出2	预期交互 输入、输出3	预期交互 输入、输出4
用户以“zqf”身份登录，进入需求中心页面，点击已发布的需求模块	用户点击“已发布的需求”按钮，之后系统正常跳转到已发布的需求页面，在需要删除的历史需求上点击删除按钮，系统将历史需求从数据库当中删除，并提示删除结果	无	无	无
最后预期输出				
系统将历史需求从数据库当中删除，并提示删除结果				

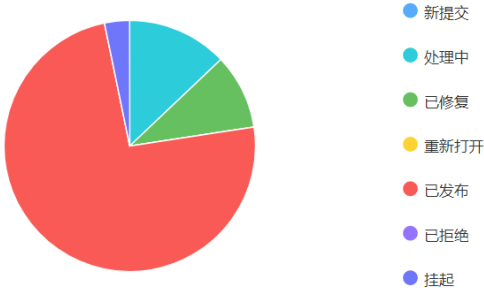
用例编号	EZR_Sys_Test_case_fun_005			
测试覆盖的系统功能	管理员审核通过物品（覆盖正常功能）			
用例设计方法	场景法，错误推测法			
前置条件	管理员已登录			
输入				
初始输入	预期交互	预期交互	预期交互	预期交互



	输入、输出1	输入、输出2	输入、输出3	输入、输出4
管理员点击审核通过按钮	管理员点击审核通过按钮，物品的状态由待审核修改为审核通过状态，物品成功上架	无	无	无
最终预期结果				
物品上架				

用例编号	EZR_Sys_Test_case_fun_005			
测试覆盖的系统功能	管理员审核不通过物品（覆盖非正常功能）			
用例设计方法	场景法，错误推测法			
前置条件	管理员已登录			
输入				
初始输入	预期交互 输入、输出1	预期交互 输入、输出2	预期交互 输入、输出3	预期交互 输入、输出4
管理员点击下架物品按钮	管理员点击下架物品按钮，物品从数据库当中删除	无	无	无
最终预期结果				
物品下架				

缺陷状态分析图

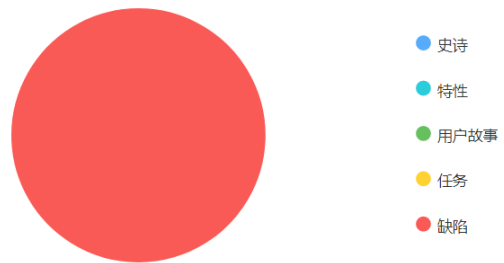


表格

状态	缺陷数	百分比
新提交	0	0.00%
处理中	4	12.90%
已修复	3	9.68%
重新打开	0	0.00%
已发布	23	74.19%
已拒绝	0	0.00%
挂起	1	3.23%

缺陷属性分析图

图表



表格

工作项类型	缺陷数	百分比
史诗	0	0.00%
特性	0	0.00%
用户故事	0	0.00%
任务	0	0.00%
缺陷	31	100.00%