

集成测试文档

第一部分 集成测试计划

1. 引言

1.1 目的

本文是“邻临七企划--十五分钟生活圈临期食品智慧社区”系统的集成测试的大纲，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能正常工作，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) 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。

1.3 术语

本计划涉及的主要术语如下：

- 集成测试（Unit Testing）：将多个单独的模块组合在一起进行测试，以确保它们能协同工作。
- 驱动（Driver）：用于调用和测试被测模块的特殊代码或程序，模拟上层模块对被测模块的调用。
- 接口（Interface）：系统或模块之间的交互边界，定义了数据交换的方式和协议。
- 测试用例（Test Case）：用于验证系统或模块功能是否符合预期的具体场景描述，包括输入、预期输出和执行步骤。
- 测试环境（Test Environment）：执行测试所需的软件、硬件和网络配置。
- 测试计划（Test Plan）：描述测试目标、策略、范围、资源、时间表和测试活动的文档。

- 测试策略（Test Strategy）：高层次的测试方法和测试设计的指导方针，决定了测试的总体方向和优先级。
- 覆盖率（Coverage）：衡量测试用例覆盖代码、功能或需求的程度。

1.4 测试环境

序号	描述	配置
1#	浏览器	Google Chrome
2#	输入习惯	中文
3#	操作系统环境	Windows 11
4#	测试工具	Apifox

1.5 参考文件及工作产品

集成测试涉及以下文档：

- 《需求分析规约》
- 《概要设计说明书》
- 《详细设计说明书》
- 《单元测试报告》

执行测试前涉及的任务：

- 用例设计完成并通过评审；
- 测试脚本开发完成；
- 测试环境搭建完成；
- 测试过程及缺陷管理流程确定。

集成测试交付物：

- 集成测试计划
- 集成测试计划用例规格
- 集成测试用例脚本
- 集成测试驱动和桩代码
- 集成测试执行日志

- 集成测试报告

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

如把公共服务子系统、顾客子系统及商家子系统进行集成。

2.3 集成策略

系统的集成测试采用自底向上的集成(Bottom-Up Integration)方式。自底向上集成方式从程序模块结构中最底层的模块开始组装和测试。因为模块是自底向上进行组装的，对于一个给定层次的模块，它的子模块(包括子模块的所有下属模块)事前已经完成组装并经过测试，所以不再需要编制桩模块。选择这种集成方法，管理方便，测试人员能较好地锁定软件故障所在位置。

集成测试中的主要步骤：

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

活动	输入	输出	职责
制订并审核集成测试计划	概要设计说明书等	集成测试计划	制定测试计划
测试用例分析、设计及评审	集成测试计划概要设计说明书	设计测试用例书	设计测试用例并评审
测试的实施	集成测试用例 测试过程	测试脚本 测试环境	开发测试脚本、搭建测试环境
		测试驱动	开发驱动
测试的执行	测试脚本	测试结果	记录结果、跟踪缺陷
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	会同开发人员评估测试结果，得出测试报告

3. 测试过程描述

3.1 软件集成测试

在“邻临七企划--十五分钟生活圈临期食品智慧社区”项目中，集成测试的主要过程如下：

(1) 设计集成测试用例。

自底向上集成测试的步骤：

步骤1：按照概要设计规格说明，明确哪些是被测模块。在熟悉被测模块性质的基础上对被测模块进行分层，在同一层次上的测试可以并行进行，然后排出测试活动的先后关系，制订测试进度计划。

步骤2：在步骤1的基础上，按时间线序关系将软件单元集成为模块，并测试在集成过程中出现的问题。这里可能需要测试人员开发一些驱动模块来驱动集成活动中形成的被测模块。对于比较大的模块，可以先将其中的某几个软件单元集成为子模块，然后再集成为一个较大的模块。

步骤3：将各软件模块集成为子系统(或分系统)。检测各自子系统是否能正常工作。同样，可能需要测试人员开发少量的驱动模块来驱动被测子系统。

步骤4：将各子系统集成为最终用户系统，测试各子系统能否在最终用户系统中正常工作。

(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。

(3) 问题反馈：

反馈Bug给开发人员。

(4) 回归测试：

重新测试修复Bug后的系统。重复（3），直到（4）中回归测试结果到达系统验收标准。

(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。

3.2 子系统集成测试

项目采用Spring Boot经典的分层架构，自底向上分为Repository层、Service层和Controller层。其中，**Repository层** 负责与数据库的交互、**Service层** 负责主要业务逻辑的实现、**Controller层** 负责前后端的接口交互。每个用例的实现流程可概括为，Controller层调用Service层，Service层调用Repository层。各层之间的实现类不相互调用，子系统间高内聚，且无需相互调用服务实现。**故项目中无子系统间集成。**

3.3 外部系统集成测试

外部系统接口集成的主要目的是实现不同系统之间的数据交换和功能协同，以提升整体业务流程的自动化和效率。本系统为实现更全面的功能，涉及与支付宝沙盒系统、百度地图系统等外部系统的交互。故在集成测试阶段，还需要对上述接口进行覆盖性的测试，以验证接口交互的正确性，保障系统功能正常运行。

4. 集成测试验收标准

4.1 模块验收标准

- 接口：接口提供的功能或者数据正确。
- 功能点：验证程序与产品描述、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应，一致性。

4.2 集成测试验收标准

集成测试用例中所设计的功能测试用例必须全部通过。

5. 测试工具

- **PingCode**：测试进度管理与缺陷跟踪工具
- **Github**：代码版本管理与协作工具
- **ApiFox**：集成测试工具
- **飞书**：项目文档协作工具

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：傅佳恒

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：韩嘉睿、傅佳恒

设计集成测试用例。

测试人员：傅佳恒、韩嘉睿、郑皓予

按照测试用例进行测试活动。

开发人员：吴冠亚

程序Bug修改，程序员间协调。

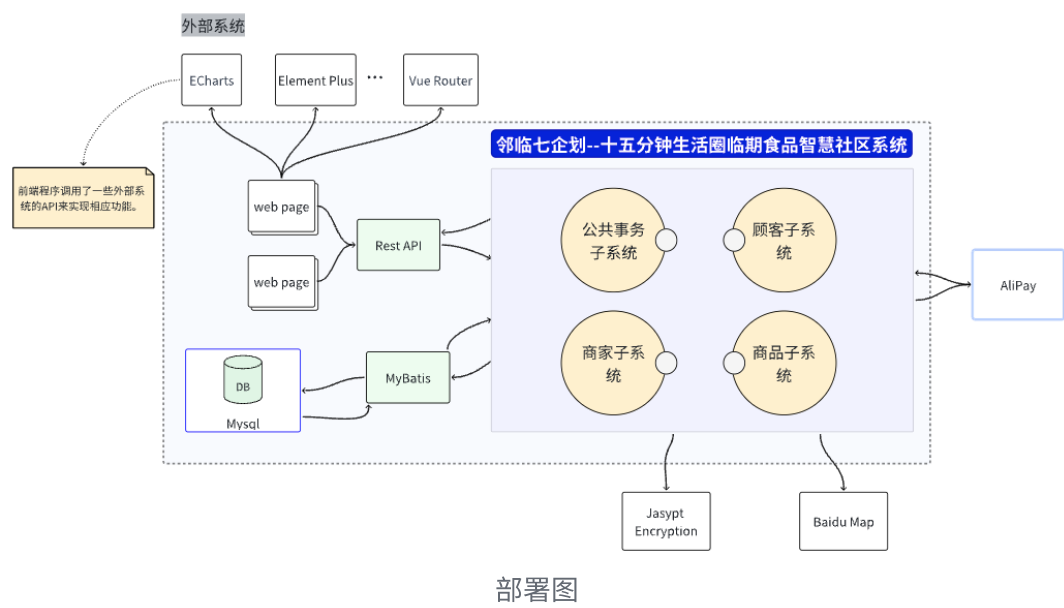
第二部分 集成测试分析与设计

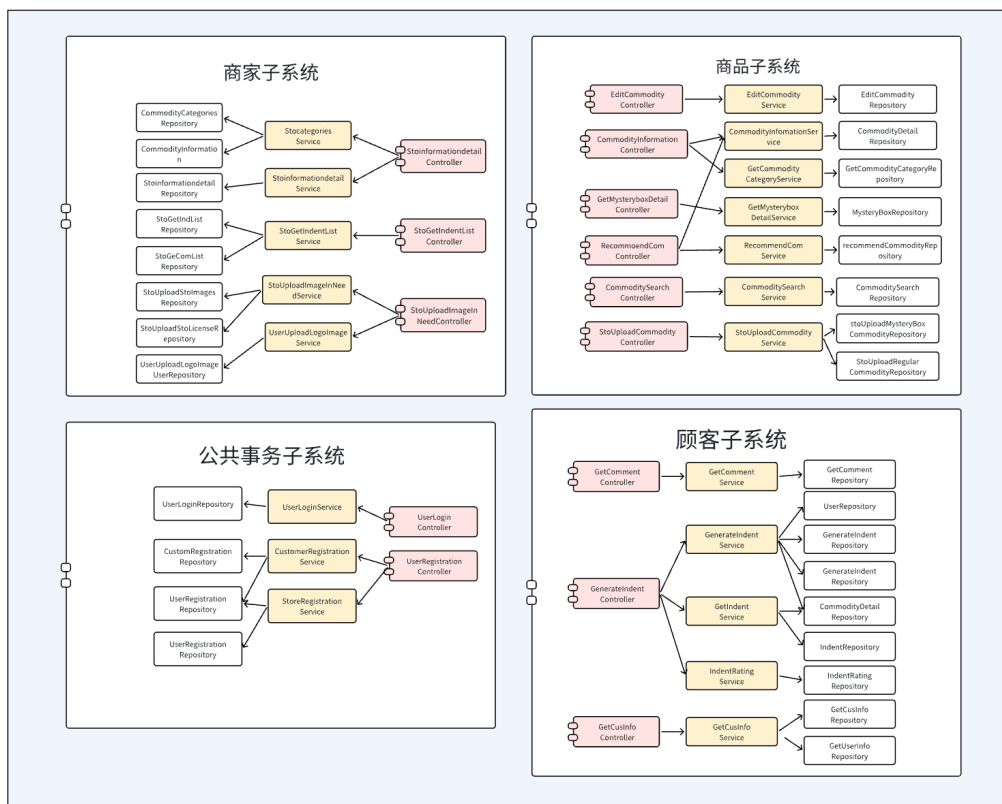
1. 集成测试依据

集成测试的主要参考文档包括：

- 《概要设计说明书》
- 《详细设计说明书》

特别要注意的而是，接口识别和集成测试用例设计直接依赖于概要设计说明书当中的部署图及架构图。架构图中展示了类间、子系统间接口，部署图中展示了系统级别间的接口。





架构图

2. 测试内容

2.1 系统内集成

根据架构图，本项目包含四个子系统：商家子系统、商品子系统、公共事务子系统和顾客子系统。系统内部的集成测试将针对商家子系统和公共事务子系统进行。

项目采用Spring Boot经典的分层架构，自底向上分为Repository层、Service层和Controller层。其中，**Repository层** 负责与数据库的交互、**Service层** 负责主要业务逻辑的实现、**Controller层** 负责前后端的接口交互。每个用例的实现流程可概括为，Controller层调用Service层，Service层调用Repository层。各层之间的实现类不相互调用，子系统间高内聚，且无需相互调用服务实现。

鉴于项目使用清晰的分层架构，项目采用自底向上的集成策略，对Repository层与Service层间、Service层与Controller层分别进行集成测试。针对选定的商家子系统和公共事务子系统，识别其各层间的接口，设计测试用例，实现子系统内测试接口的全覆盖。

- 针对公共事务子系统，对Service层调用Repository层接口服务的接口识别如下：

序号	接口标识符	接口名称	描述	重要性
1	INT_001_001	UserLoginRepository.findByUserPhone(String user_phone)	UserLoginService调用该服务查询数据的对应电话号码信息的	重要
2	INT_002_001	UserRegistrationRepository.existsByUserPhone(String	CustomerRegistrationService调用该服务判断对应电话号码是否	重要

		user_phone)	存在	
3	INT_002_002	UserRegistrationRepository.save(newUser)	CustomerRegistrationService调用该服务向数据库中插入新用户	重要
4	INT_002_003	CustomRegistrationRepository.save(newCus)	CustomerRegistrationService调用该服务向数据库中插入新顾客	重要
5	INT_003_001	UserRegistrationRepository.existsByUserPhone(String user_phone)	StoreRegistrationService调用该服务判断对应电话号码是否存在	重要
6	INT_003_002	UserRegistrationRepository.save(newUser)	StoreRegistrationService调用该服务向数据库中插入新用户	重要
7	INT_003_003	StoreRegistrationRepository.save(newSto)	StoreRegistrationService调用该服务向数据库中插入新商家	重要

- 针对公共事务子系统，对Controller层调用Service层接口服务的接口识别如下：

序号	接口标识符	接口名称	描述	重要性
1	INT_004_001	UserLoginService.UserLogin(UserLoginRequestDTO)	UserLoginController调用该服务实现用户的登录功能	重要
2	INT_005_001	CustomerRegistrationService.UserRegister(CustomRegistrationRequestDTO)	UserRegistrationController调用该服务实现顾客注册的功能	重要
3	INT_005_002	storeRegistrationService.StoreRegister(StoreRegistrationRequestDTO)	UserRegistrationController调用该服务实现商家注册的功能	重要

- 针对商家子系统，由于Service层与Controller层以及Repository层与Service层之间的调用关系较为简单，我们可以直接进行两层一并的集成测试，简化测试流程，提高测试效率，对Controller层调用Service层的接口服务识别如下：

序号	接口标识符	接口名称	描述	重要性
1	INT_006_001	StoGetIndentListService.StoGetIndent(sto_id)	StoGetIndentListController调用该服务实现商家订单信息列表的获取	重要

2	INT_007_001	StoinformationdetailService.stoinformationdetail(sto_id)	StoinformationdetailController调用该服务实现商家信息的获取	重要
3	INT_007_002	StocategoriesService.getStoCategoriesOnsale(sto_ID)	StoinformationdetailController调用该服务实现商家在售商品类别的获取	重要
4	INT_008_001	StoUploadImageInNeedService.UploadStoImage()	StoUploadImageInNeedController调用该服务实现商家照片的上传	重要
5	INT_008_002	StoUploadImageInNeedService.UploadStoLicenseImage()	StoUploadImageInNeedController调用该服务实现商家资质的上传	重要
6	INT_008_003	StoUploadImageInNeedService.userUploadLogo	StoUploadImageInNeedController调用该服务实现商家标志的上传	重要

2.2 外部系统集成

根据部署图，本系统涉及与支付宝支付沙箱、百度地图等外部系统的集成。在本报告中，以支付宝的部分接口为例，进行本系统与外部系统的集成。考虑到此类外部系统设计通常较为稳定与健壮，在用例设计的时候仅考虑遵循每个接口都覆盖到的基本标准。识别相关接口如下：

序号	接口标识符	接口名称	描述	重要性
1	ALI_INT_001	/api/payment	调用支付宝的付款接口，返回支付宝支付界面相关信息	一般

3. 测试方法

- a. 采用自下而上的集成策略。
- b. 作为一个例子对UserLoginService进行模块内集成，对findByUserPhone进行测试。
- c. 作为例子对公共事务子系统的UserLoginController、UserRegistrationController以及商家子系统的StoinformationdetailController、StoGetIndentListController、StoUploadImageInNeedController模块进行集成测试。
- d. 覆盖分析：要求做到识别出的接口设计的用例全部通过。
- e. 对于重要性为“重要”的接口，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。考虑使用等价类划分及边界值分析方法。

4. 测试用例设计

4.1 INT_001_001测试用例设计

4.1.1 设计标识符

INT_001_001

4.1.2 被测特性

- 输入手机号存在时，返回对应用户信息。
- 输入手机号不存在（账号未注册）时，返回null。

4.1.3 测试方法

使用强一般等价类进行测试用例的分析和设计，可以划分为两种测试情况，分别进行测试用例的设计。

4.1.4 测试项标识

测试项标识符	测试项目描述	优先级
INT_001_001_001	输入已注册手机号	中
INT_001_001_002	输入未注册手机号	中

4.1.5 测试通过/失败标准

- 测试用例集中，函数调用的接口及调用接口的路径至少覆盖一次。
- 所有的测试用例都必须被执行，且没有发现缺陷。

4.1.6 测试用例表

测试项编号	INT_001_001_001	
优先级	中	
测试项描述	输入已注册手机号	
前置条件	用户进入网站登录页面，前端向后端“/api/pub/login”接口发送登录请求，Controller的调用UserLoginService类的UserLogin方法	
用例序号	输入	期望结果
001	user_phone="15821295304" user_password="123456"	UserEntity{ user_ID=,

		<pre>user_phone=, user_password=, user_address=, user_state=, user_balance=, user_regTime=, user_type=, user_gender= }</pre>
--	--	--

测试项编号	INT_001_001_002	
优先级	中	
测试项描述	输入未注册手机号	
前置条件	用户进入网站登录页面，前端向后端“/api/pub/login”接口发送登录请求，Controller的调用UserLoginService类的UserLogin方法	
用例序号	输入	期望结果
001	<pre>user_phone="15899583310" user_password="123456"</pre>	null

4.2 INT_004_001测试用例设计

4.2.1 设计标识符

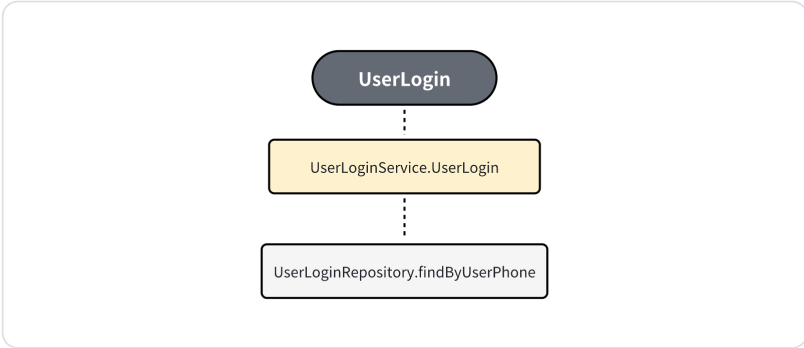
INT_004_001

4.2.2 被测特性

- 输入手机号不合法时，登录失败，返回错误信息
- 输入手机号不存在（账号未注册）时，登录失败，返回错误信息
- 输入手机号与密码不匹配时，登录失败，返回错误信息
- 输入手机号为商家账号时，商家登录成功，返回正确的商家登录信息
- 输入手机号为顾客账号时，顾客登录成功，返回正确的顾客登录信息

4.2.3 测试方法

分析UserLoginController的函数调用关系图，Controller层函数UserLogin接受前端传递的信号并调用UserLoginService类的UserLogin函数进行交互完成登录的功能，Service函数调用UserLoginRepository类的findByUserPhone方法查询手机号对应的用户。



当调用该接口时，执行输入手机号合法且存在的用例时，即可完成调用接口的全覆盖。使用强一般等价类进行测试用例的分析和设计，结合业务需求，可以划分为五种测试情况，分别进行测试用例的设计。

4.2.4 测试项标识

测试项标识符	测试项目描述	优先级
INT_004_001_001	输入已注册商家手机号，密码正确	高
INT_004_001_002	输入已注册顾客手机号，密码正确	高
INT_004_001_003	输入已注册手机号，密码错误	高
INT_004_001_004	输入未注册手机号	高
INT_004_001_004	输入不合法手机号	低

4.2.5 测试通过/失败标准

- 测试用例集中，函数调用的接口及调用接口的路径至少覆盖一次。
- 所有的测试用例都必须被执行，且没有发现缺陷。

4.2.6 测试用例表

测试项编号	INT_004_001_001
优先级	高
测试项描述	输入已注册商家手机号，密码正确
前置条件	用户进入网站登录页面，前端向后端 “/api/pub/login” 接口发送登录请求

用例序号	输入	期望结果
001	user_phone="15821295304" user_password="123456"	msg="登录成功" user_type=1 user_id=43

测试项编号	INT_004_001_002	
优先级	高	
测试项描述	输入已注册顾客手机号，密码正确	
前置条件	用户进入网站登录页面，前端向后端“/api/pub/login”接口发送登录请求	
用例序号	输入	期望结果
001	user_phone="13837829689" user_password="123456"	msg="登录成功" user_type=0 user_id=49

测试项编号	INT_004_001_003	
优先级	高	
测试项描述	输入已注册手机号，密码错误	
前置条件	用户进入网站登录页面，前端向后端“/api/pub/login”接口发送登录请求	
用例序号	输入	期望结果
001	user_phone="13837829689" user_password="111111"	msg="密码错误" user_type=-1 user_id=-1
002	user_phone="15821295304" user_password="654321"	msg="密码错误" user_type=-1 user_id=-1

测试项编号	INT_004_001_004	
优先级	高	

测试项描述	输入未注册手机号	
前置条件	用户进入网站登录页面，前端向后端“/api/pub/login”接口发送登录请求	
用例序号	输入	期望结果
001	user_phone="15899583310" user_password="123456"	msg="账号不存在" user_type=-1 user_id=-1

测试项编号	INT_004_001_005	
优先级	低	
测试项描述	输入不合法手机号	
前置条件	用户进入网站登录页面，前端向后端“/api/pub/login”接口发送登录请求	
用例序号	输入	期望结果
001	user_phone="10086" user_password="111111"	msg="电话格式错误" user_type=-1 user_id=-1

4.3 INT_005_001测试用例设计

4.3.1 设计标识符

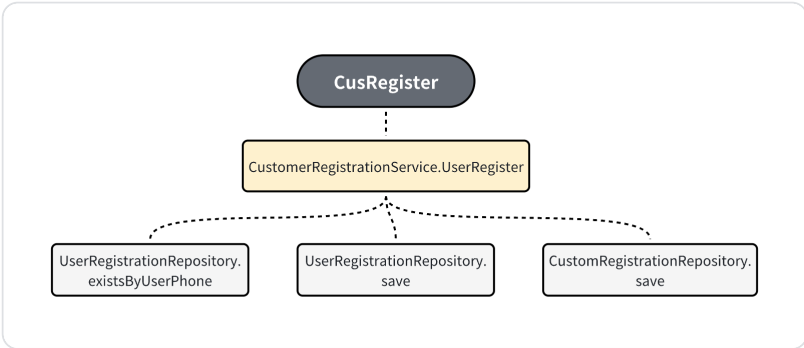
INT_005_001

4.3.2 被测特性

- 输入电话号码格式不合法（非11位数字）时，返回默认错误信息。
- 输入电话号码已被其他用户使用时，返回默认错误信息。
- 输入密码格式不合法（合法密码长度不低于6位，不超过12位，且包含字母、数字和特殊符号）时，返回默认错误信息。
- 输入性别值不合法（非0或1）时，返回默认错误信息。
- 输入昵称长度不合法（合法长度大于0位，小于10个字符）时，返回默认错误信息。
- 输入支付密码不合法（非6-10位数字时），返回默认错误信息。

4.3.3 测试方法

分析UserRegisterController的函数调用关系图，Controller层函数接受前端传递的信号并调用CustomerRegistrationService类的UserRegister函数进行交互完成顾客注册的功能，Service函数调用UserRegistrationRepository类的existsByUserPhone方法查询手机号是否已被注册，调用UserRegistrationRepository类的save方法保存新注册用户信息，调用CustomRegistrationRepository类的save方法保存顾客个人信息。



当调用该接口时，执行输入手机号合法且未注册且所有注册信息输入合法时，即可完成调用接口的全覆盖。使用强一般等价类进行测试用例的分析和设计，可以划分为7种情况，分别进行测试用例的设计。

4.3.4 测试项标识

测试项标识符	测试项目描述	优先级
INT_005_001_001	输入手机号合法且未注册、密码合法、用户性别为0、昵称长度合法、支付密码合法，成功注册。	高
INT_005_001_002	输入手机号合法但已注册、密码合法、用户性别为1、昵称长度合法、支付密码合法，注册失败。	高
INT_005_001_003	输入手机号不合法	高
INT_005_001_004	输入密码不合法	高
INT_005_001_005	输入用户性别不合法	低
INT_005_001_006	输入用户昵称不合法	中
INT_005_001_007	输入支付密码不合法	高

4.3.5 测试通过/失败标准

- 测试用例集中，函数调用的接口及调用接口的路径至少覆盖一次。
- 所有的测试用例都必须被执行，且没有发现缺陷。

4.3.6 测试用例表

测试项编号	INT_005_001_001	
优先级	高	
测试项描述	输入手机号合法且未注册、密码合法、用户性别为0、昵称长度合法、支付密码合法，成功注册。	
前置条件	用户进入网站注册页面，前端向后端 “/api/pub/register/customer” 接口发送登录请求	
用例序号	输入	期望结果
001	user_phone="15815673200" user_password="123456" user_gender=0 cus_nickname="Donald" cus_Paypassword="111111"	msg="顾客成功注册" user_type=0 user_id=3

测试项编号	INT_005_001_002	
优先级	高	
测试项描述	输入手机号合法但已注册、密码合法、用户性别为1、昵称长度合法、支付密码合法，注册失败。	
前置条件	用户进入网站注册页面，前端向后端 “/api/pub/register/customer” 接口发送登录请求	
用例序号	输入	期望结果
001	user_phone="13837829689" user_password="123456" user_gender=1 cus_nickname="Trump" cus_Paypassword="111111"	msg="该号码已存在" user_type=0 user_id=-1

测试项编号	INT_005_001_003	
优先级	高	
测试项描述	输入手机号不合法	
前置条件	用户进入网站注册页面，前端向后端 “/api/pub/register/customer” 接口发送登录请求	
用例序号	输入	期望结果

001	user_phone="phonenumber" user_password="123456" user_gender=0 cus_nickname="Donald" cus_Paypassword="111111"	msg="电话号码格式不正确，必须为11位数字" user_type=0 user_id=-1
-----	--	---

测试项编号	INT_005_001_004	
优先级	高	
测试项描述	输入密码不合法	
前置条件	用户进入网站注册页面，前端向后端 “/api/pub/register/customer” 接口发送登录请求	
用例序号	输入	期望结果
001	user_phone="13837829689" user_password="Ab_1" user_gender=0 cus_nickname="Donald" cus_Paypassword="111111"	msg="密码长度必须不低于6位，并且包含字母、数字和特殊符号" user_type=0 user_id=-1

测试项编号	INT_005_001_005	
优先级	低	
测试项描述	输入用户性别不合法	
前置条件	用户进入网站注册页面，前端向后端 “/api/pub/register/customer” 接口发送登录请求	
用例序号	输入	期望结果
001	user_phone="15899583310" user_password="123456" user_gender=2 cus_nickname="Donald" cus_Paypassword="111111"	msg="性别必须为0或1" user_type=0 user_id=-1

测试项编号	INT_005_001_006	

优先级	中	
测试项描述	用户昵称不合法	
前置条件	用户进入网站注册页面，前端向后端 “/api/pub/register/customer” 接口发送登录请求	
用例序号	输入	期望结果
001	user_phone="15899583310" user_password="123456" user_gender=0 cus_nickname="" cus_Paypassword="111111"	msg="昵称长度必须小于10" user_type=0 user_id=-1

测试项编号	INT_005_001_007	
优先级	高	
测试项描述	输入支付密码不合法	
前置条件	用户进入网站注册页面	
用例序号	输入	期望结果
001	user_phone="15899583310" user_password="123456" user_gender=0 cus_nickname="Donald" cus_Paypassword=""	msg="支付密码长度必须为6到10位数字" user_type=0 user_id=-1

4.4 INT_005_002测试用例设计

4.4.1 设计标识符

INT_005_002

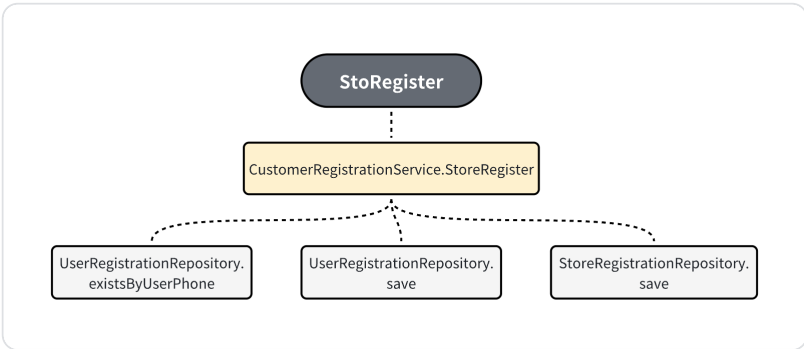
4.4.2 被测特性

- 输入电话号码格式不合法（非11位数字）时，返回默认错误信息。
- 输入电话号码已被其他用户使用时，返回默认错误信息。
- 输入密码格式不合法（合法密码长度不低于6位，不超过12位，且包含字母、数字和特殊符号）时，返回默认错误信息。

- 输入商家名称不合法（为空或大于20个汉字），返回默认错误信息。
- 输入性别值不合法（非0或1）时，返回默认错误信息。
- 输入介绍信息不合法（合法长度不超过50个字符）时，返回默认错误信息。
- 输入营业时间不合法（关门事件早于开门时间）时，返回默认错误信息。

4.4.3 测试方法

分析UserRegisterController的函数调用关系图，Controller层函数接受前端传递的信号并调用CustomerRegistrationService类的StoreRegister函数进行交互完成顾客注册的功能，Service函数调用UserRegistrationRepository类的existsByUserPhone方法查询手机号是否已被注册，调用UserRegistrationRepository类的save方法保存新注册用户信息，调用StoreRegistrationRepository类的save方法保存顾客个人信息。



当调用该接口时，执行输入手机号合法且未注册且所有注册信息输入合法时，即可完成调用接口的全覆盖。使用强一般等价类进行测试用例的分析和设计，可以划分为7种情况，分别进行测试用例的设计。

4.4.4 测试项标识

测试项标识符	测试项目描述	优先级
INT_005_001_001	输入手机号合法且未注册、密密码、地址、性别、商店名称、商店介绍、营业时间、GPS坐标合法，成功注册。	高
INT_005_001_002	输入手机号合法但已注册	高
INT_005_001_003	输入手机号不合法	高
INT_005_001_004	输入密码不合法	高
INT_005_001_005	输入地址不合法	中
INT_005_001_006	输入用户性别不合法	低
INT_005_001_007	输入商店名称不合法	中

INT_005_001_008	输入商店介绍不合法	低
INT_005_001_009	输入营业时间不合法	中
INT_005_001_010	输入经纬度信息不合法	中

4.4.5 测试通过/失败标准

- 测试用例集中，函数调用的接口及调用接口的路径至少覆盖一次。
- 所有的测试用例都必须被执行，且没有发现缺陷。

4.4.6 测试用例表

测试项编号	INT_005_002_001	
优先级	高	
测试项描述	输入手机号合法且未注册、密密码、地址、性别、商店名称、商店介绍、营业时间、GPS坐标合法，成功注册。	
前置条件	用户进入网站注册页面，前端向后端 “/api/pub/register/store” 接口发送登录请求	
用例序号	输入	期望结果
001	user_phone="15815673200" user_password="123456" user_address="上海市嘉定区嘉三路" user_gender=0 sto_name="爱国水果店" sto_introduction="本店经营各类新鲜水果。" sto_openingTime=08:30 sto_closingTime=17:30 sto_latitude="31.294428760329854" sto_longitude="121.21897535333184"	msg="Store成功注册" user_type=1 user_id=3

测试项编号	INT_005_002_002

优先级	高	
测试项描述	输入手机号合法但已注册	
前置条件	用户进入网站注册页面，前端向后端 “/api/pub/register/store” 接口发送登录请求	
用例序号	输入	期望结果
001	user_phone="15821295304" user_password="123456" user_address="上海市嘉定区嘉三路" user_gender=0 sto_name="爱国水果店" sto_introduction="本店经营各类新鲜水果。" sto_openingTime=08:30 sto_closingTime=17:30 sto_latitude="31.294428760329854" sto_longitude="121.21897535333184"	msg="该号码已存在" user_type=1 user_id=-1

测试项编号	INT_005_002_003	
优先级	高	
测试项描述	输入手机号不合法	
前置条件	用户进入网站注册页面，前端向后端 “/api/pub/register/store” 接口发送登录请求	
用例序号	输入	期望结果
001	user_phone="phonenumner" user_password="123456" user_address="上海市嘉定区嘉三路" user_gender=0 sto_name="爱国水果店" sto_introduction="本店经营各类新鲜水果。"	msg="电话号码格式不正确，必须为11位数字" user_type=1 user_id=-1

	sto_openingTime=08:30 sto_closingTime=17:30 sto_latitude="31.294428760329854" sto_longitude="121.21897535333184"	
--	---	--

测试项编号	INT_005_002_004	
优先级	高	
测试项描述	输入密码不合法	
前置条件	用户进入网站注册页面，前端向后端“/api/pub/register/store”接口发送登录请求	
用例序号	输入	期望结果
001	user_phone="13837829689" user_password="Ab_1" user_address="上海市嘉定区嘉三路" user_gender=0 sto_name="爱国水果店" sto_introduction="本店经营各类新鲜水果。" sto_openingTime=08:30 sto_closingTime=17:30 sto_latitude="31.294428760329854" sto_longitude="121.21897535333184"	msg="密码长度必须不低于6位，并且包含字母、数字和特殊符号" user_type=1 user_id=-1

测试项编号	INT_005_002_005	
优先级	中	
测试项描述	输入地址不合法	
前置条件	用户进入网站注册页面，前端向后端“/api/pub/register/store”接口发送登录请求	
用例序号	输入	期望结果

001	user_phone="15815673200" user_password="123456" user_address="" user_gender=0 sto_name="爱国水果店" sto_introduction="本店经营各类新鲜水果。" sto_openingTime=08:30 sto_closingTime=17:30 sto_latitude="31.294428760329854" sto_longitude="121.21897535333184"	msg="商家地址不为空" user_type=1 user_id=-1
-----	---	--

测试项编号	INT_005_002_006	
优先级	低	
测试项描述	输入用户性别不合法	
前置条件	用户进入网站注册页面，前端向后端“/api/pub/register/store”接口发送登录请求	
用例序号	输入	期望结果
001	user_phone="15815673200" user_password="123456" user_address="上海市嘉定区嘉三路" user_gender=2 sto_name="爱国水果店" sto_introduction="本店经营各类新鲜水果。" sto_openingTime=08:30 sto_closingTime=17:30 sto_latitude="31.294428760329854" sto_longitude="121.21897535333184"	msg="性别必须为0或1" user_type=1 user_id=-1

测试项编号	INT_005_002_007	
优先级	中	
测试项描述	输入商店名称不合法	
前置条件	用户进入网站注册页面，前端向后端 “/api/pub/register/store” 接口发送登录请求	
用例序号	输入	期望结果
001	user_phone="15815673200" user_password="123456" user_address="上海市嘉定区嘉三路" user_gender=0 sto_name="" sto_introduction="本店经营各类新鲜水果。" sto_openingTime=08:30 sto_closingTime=17:30 sto_latitude="31.294428760329854" sto_longitude="121.21897535333184"	msg="商家名称长度不正确" user_type=1 user_id=-1

测试项编号	INT_005_002_008	
优先级	低	
测试项描述	输入商店介绍不合法	
前置条件	用户进入网站注册页面，前端向后端 “/api/pub/register/store” 接口发送登录请求	
用例序号	输入	期望结果
001	user_phone="15815673200" user_password="123456" user_address="上海市嘉定区嘉三路" user_gender=0 sto_name="爱国水果店"	msg="介绍长度必须不超过50个字符" user_type=1 user_id=-1

	<p>sto_introduction="本店位于上海市嘉定区嘉三路，坐落于美丽的同济大学图书馆楼下，同舟河畔，专业经营各类新鲜水果十八年，欢迎各位同学进店选购"</p> <p>sto_openingTime=08:30</p> <p>sto_closingTime=17:30</p> <p>sto_latitude="31.294428760329854"</p> <p>sto_longitude="121.21897535333184"</p>	
--	--	--

测试项编号	INT_005_002_009	
优先级	中	
测试项描述	输入的营业时间不合法	
前置条件	用户进入网站注册页面，前端向后端“/api/pub/register/store”接口发送登录请求	
用例序号	输入	期望结果
001	<p>user_phone="15815673200"</p> <p>user_password="123456"</p> <p>user_address="上海市嘉定区嘉三路"</p> <p>user_gender=0</p> <p>sto_name="爱国水果店"</p> <p>sto_introduction="本店经营各类新鲜水果。"</p> <p>sto_openingTime=12:30</p> <p>sto_closingTime=08:30</p> <p>sto_latitude="31.294428760329854"</p> <p>sto_longitude="121.21897535333184"</p>	<p>msg="关门时间必须晚于开门时间"</p> <p>user_type=1</p> <p>user_id=-1</p>

测试项编号	INT_005_002_010	
优先级	中	

测试项描述	输入经纬度信息不合法	
前置条件	用户进入网站注册页面，前端向后端 “/api/pub/register/store” 接口发送登录请求吗，m'n	
用例序号	输入	期望结果
001	user_phone="15815673200" user_password="123456" user_address="上海市嘉定区嘉三路" user_gender=0 sto_name="爱国水果店" sto_introduction="本店经营各类新鲜水果。" sto_openingTime=08:30 sto_closingTime=17:30 sto_latitude="aaaa" sto_longitude="bbbb"	msg="错误的经纬度格式" user_type=1 user_id=-1

4.5 INT_006_001测试用例设计

4.5.1 设计标识符

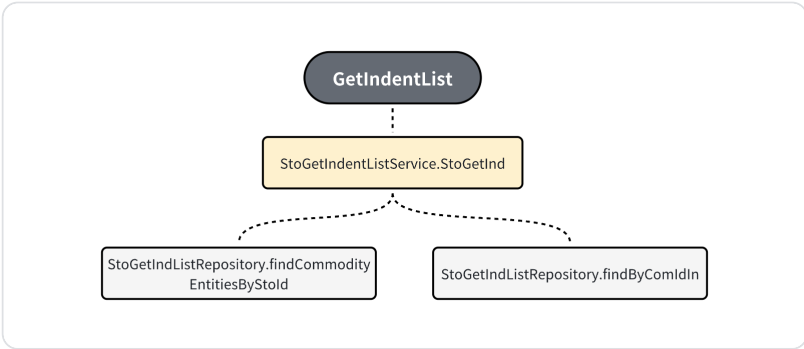
INT_006_001

4.5.2 被测特性

- 输入参数为空时，方法调用失败
- 输入参数不合法时，方法调用失败，返回错误信息
- 输入商家不存在时，返回空列表
- 输入商家合法且存在时，返回正确的订单列表

4.5.3 测试方法

分析StoGetIndentListController的函数调用关系图，Controller层函数GetIndentList接受前端传递的信号并，调用StoGetIndentListService.StoGetInd函数进行交互完成获取订单商品的功能，Service函数调用StoGetIndListRepository.findCommodityEntitiesByStoId(sto_id)查询商家对应商品信息列表，进而调用stoGetIndListRepository.findByComIdIn(com_ids)实现查询商品对应的订单列表，最终实现对商家订单信息的查询。



当调用该接口时，执行输入商家合法且存在的用例时，即可完成调用接口的全覆盖，使用强一般等价类进行测试用例的分析和设计，可以划分为上述被测特性的四种等价类情况，分别进行测试用例的设计。

4.5.4 测试项标识

测试项标识符	测试项目描述	优先级
INT_006_001_001	输入参数为空情况	低
INT_006_001_002	输入参数不合法情况	低
INT_006_001_003	输入商家不存在情况	低
INT_006_001_004	输入商家合法且存在情况	高

4.5.5 测试通过/失败标准

- 测试用例集中，函数调用的接口及调用接口的路径至少覆盖一次。
- 所有的测试用例都必须被执行，且没有发现缺陷。

4.5.6 测试用例表

测试项编号	INT_006_001_001	
优先级	低	
测试项描述	输入参数为空情况	
前置条件	商家进入订单界面，前端向后端“/api/sto/getIndentList”接口发送查询订单列表请求	
用例序号	输入	期望结果
001	sto_id=""	输入错误，返回报错信息

测试项编号	INT_006_001_002	
优先级	低	
测试项描述	输入参数不合法情况	
前置条件	商家进入订单界面，前端向后端“/api/sto/getIndentList”接口发送查询订单列表请求	
用例序号	输入	期望结果
001	sto_id="sssss"	输入错误，返回报错信息

测试项编号	INT_006_001_003	
优先级	低	
测试项描述	输入商家不存在情况	
前置条件	商家进入订单界面，前端向后端“/api/sto/getIndentList”接口发送查询订单列表请求	
用例序号	输入	期望结果
001	sto_id="10"	商家ID合法但不存在，返回空列表

测试项编号	INT_006_001_004	
优先级	高	
测试项描述	输入商家合法且存在情况	
前置条件	商家进入订单界面，前端向后端“/api/sto/getIndentList”接口发送查询订单列表请求	
用例序号	输入	期望结果
001	sto_id="41"	返回商家的订单列表，但商家无订单故列表为空
002	sto_id="43"	返回商家的订单列表

4.6 INT_007_001测试用例设计

4.6.1 设计标识符

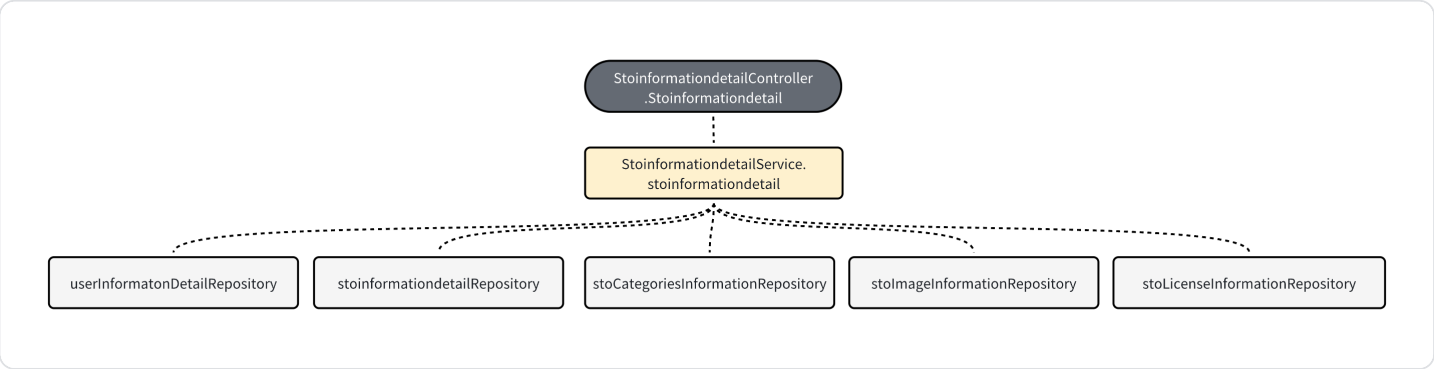
INT_007_001

4.6.2 被测特性

- 输入参数为空时，返回空数据
- 输入参数不合法时，方法调用失败，返回错误信息
- 输入商家不存在时，返回默认商家错误信息
- 输入商家合法且存在时，返回正确商家信息

4.6.3 测试方法

分析StoinformationdetailController函数的调用图，函数Stoinformationdetail接受前端的请求并调用StoinformationdetailService.stoinformationdetail函数获取商家所有信息并返回前端，Service层函数调用下面调用图中五个Repository层中函数查询商家不同的新信息并进行整合并返回，最终实现对商家信息的返回。



当调用该接口时，执行输入商家合法且存在的用例时，即可完成调用接口的全覆盖，使用强一般等价类进行测试用例的分析和设计，可以划分为上述被测特性的四种等价类情况，分别进行测试用例的设计。

4.6.4 测试项标识

测试项标识符	测试项目描述	优先级
INT_007_001_001	输入参数为空情况	低
INT_007_001_002	输入参数不合法情况	低
INT_007_001_003	输入商家不存在情况	低
INT_007_001_004	输入商家合法且存在情况	高

4.6.5 测试通过/失败标准

- 测试用例集中，函数调用的接口及调用接口的路径至少覆盖一次。
- 所有的测试用例都必须被执行，且没有发现缺陷。

4.6.6 测试用例表

测试项编号	INT_007_001_001	
优先级	低	
测试项描述	输入参数为空情况	
前置条件	商家个人信息页面，或顾客进入商家主页，前端向后端 “/api/sto/informationdetail” 接口发送查询商家信息请求	
用例序号	输入	期望结果
001	sto_id=""	输入错误，返回为空

测试项编号	INT_007_001_002	
优先级	低	
测试项描述	输入参数不合法情况	
前置条件	商家个人信息页面，或顾客进入商家主页，前端向后端 “/api/sto/informationdetail” 接口发送查询商家信息请求	
用例序号	输入	期望结果
001	sto_id="sssss"	输入错误，接口调用失败，返回报错信息

测试项编号	INT_007_001_003	
优先级	低	
测试项描述	输入商家不存在情况	
前置条件	商家个人信息页面，或顾客进入商家主页，前端向后端 “/api/sto/informationdetail” 接口发送查询商家信息请求	
用例序号	输入	期望结果
001	sto_id="10"	商家ID合法但不存在，返回空

测试项编号	INT_007_001_004	

优先级	高	
测试项描述	输入商家合法且存在情况	
前置条件	商家个人信息页面，或顾客进入商家主页，前端向后端 “/api/sto/informationdetail” 接口发送查询商家信息请求	
用例序号	输入	期望结果
001	sto_id="39"	返回正确的商家信息

4.7 INT_007_002测试用例设计

4.7.1 设计标识符

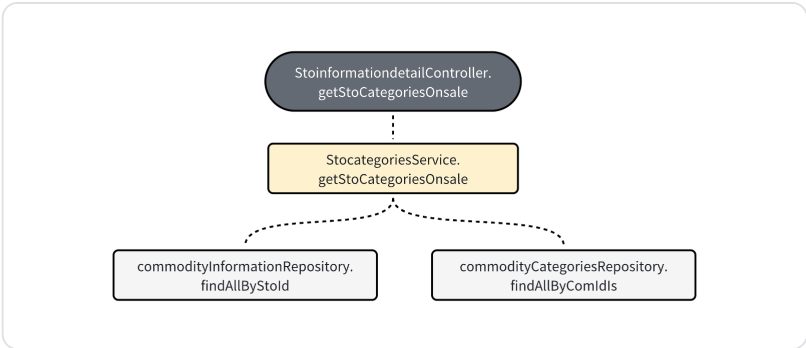
INT_007_002

4.7.2 被测特性

- 输入参数为空时，方法调用失败，返回错误信息
- 输入参数不合法时，方法调用失败，返回错误信息
- 输入商家不存在时，返回默认空信息
- 输入商家合法且存在时，商家无在售商品，返回默认空信息
- 输入商家合法且存在时，商家有在售商品，返回正确商家信息

4.7.3 测试方法

分析StoinformationdetailController函数的调用图，函数Stoinformationdetail接受前端的请求并调用StoinformationdetailService.getStoCategoriesOnsale函数获取所有在售商品的类别并返回前端，Service层调用commodityinformationRepository.findAllByStold函数查询所有的商品信息，查询到商品信息后通过commodityCategoriesRepository.findAllByComIds查询商品的类别信息，最后Service层函数对信息进行整合并返回，最终实现对商家在售商品类别的返回。



当调用该接口时，执行输入商家合法且存在的用例时，即可完成调用接口的全覆盖，使用强一般等价类进行测试用例的分析和设计，可以划分为输入参数为空、输入参数不合法、输入商家不存在、

商家合法且存在四种情况，输入输入商家合法且存在时，又可以划分为商家有无在售商品两种情况，分别进行测试用例的设计。

4.7.4 测试项标识

测试项标识符	测试项目描述	优先级
INT_007_002_001	输入参数为空情况	低
INT_007_002_002	输入参数不合法情况	低
INT_007_002_003	输入商家不存在情况	低
INT_007_002_004	输入商家合法且存在时，商家无在售商品情况	高
INT_007_002_005	输入商家合法且存在时，商家有在售商品情况	高

4.7.5 测试通过/失败标准

- 测试用例集中，函数调用的接口及调用接口的路径至少覆盖一次。
- 所有的测试用例都必须被执行，且没有发现缺陷。

4.7.6 测试用例表

测试项编号	INT_007_002_001	
优先级	低	
测试项描述	输入参数为空情况	
前置条件	顾客进入商家主页，前端向后端“/api/sto/stocategories”接口发送查询商家信息请求	
用例序号	输入	期望结果
001	sto_id=""	输入参数为空，方法调用失败，返回错误信息

测试项编号	INT_007_002_002	
优先级	低	
测试项描述	输入参数不合法情况	
前置条件	顾客进入商家主页，前端向后端“/api/sto/stocategories”接口发送查询商家信息请求	
用例序号	输入	期望结果

001	sto_id="sssss"	输入参数有误，方法调用失败，返回错误信息
-----	----------------	----------------------

测试项编号	INT_007_002_003	
优先级	低	
测试项描述	输入商家不存在情况	
前置条件	顾客进入商家主页，前端向后端“/api/sto/stocategories”接口发送查询商家信息请求	
用例序号	输入	期望结果
001	sto_id="10"	商家ID合法但不存在，默认空信息

测试项编号	INT_007_002_004	
优先级	高	
测试项描述	输入商家合法且存在情况	
前置条件	顾客进入商家主页，前端向后端“/api/sto/stocategories”接口发送查询商家信息请求	
用例序号	输入	期望结果
001	sto_id="41"	输入商家合法且存在，但商家无在售商品，返回默认空信息

测试项编号	INT_007_002_005	
优先级	高	
测试项描述	输入商家合法且存在情况	
前置条件	顾客进入商家主页，前端向后端“/api/sto/stocategories”接口发送查询商家信息请求	
用例序号	输入	期望结果
001	sto_id="39"	输入商家合法且存在，且商家有在售商品，返回正确在售商品信息列表

4.8 INT_008_001测试用例设计

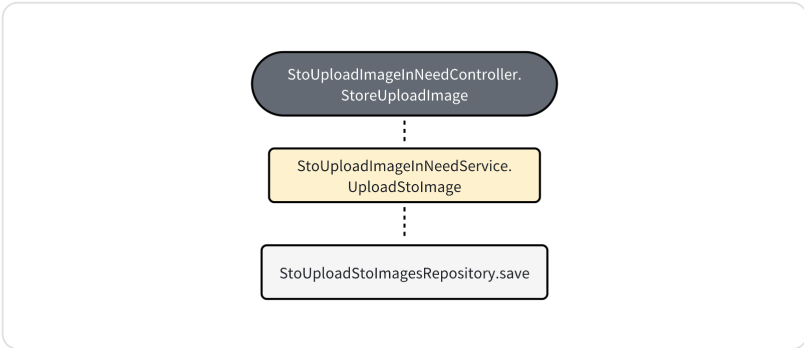
4.8.1 设计标识符

4.8.2 被测特性

- 输入参数为空时，方法调用失败，返回错误信息
- 输入参数商家ID非法时，返回错误信息
- 输入参数商家ID合法但不存在时，返回错误信息
- 输入商家合法且存在时，返回商家照片上传成功信息

4.8.3 测试方法

分析StoUploadImageInNeedController函数的调用图，函数StoreUploadImage接受前端传递图片和信息调用StoUploadImageInNeedService.UploadImage函数实现上传商家宣传图操函数调用，Service层StoUploadImageRepository.save函数将数据存储存储在数据库中，最终实现上传商品功能。



函数调用单向调用，当调用该接口时，执行输入商家合法且存在的用例时，即可完成调用接口的全覆盖，使用强一般等价类进行测试用例的分析和设计，可以划分为上述被测特性的四种等价类情况，分别进行测试用例的设计。

4.8.4 测试项标识

测试项标识符	测试项目描述	优先级
INT_008_001_001	输入参数为空时，方法调用失败情况	低
INT_008_001_002	输入参数商家ID非法情况	低
INT_008_001_003	输入参数商家ID合法但不存在情况	低
INT_008_001_004	输入商家合法且存在时情况	高

4.8.5 测试通过/失败标准

- 测试用例集中，函数调用的接口及调用接口的路径至少覆盖一次。
- 所有的测试用例都必须被执行，且没有发现缺陷。

4.8.6 测试用例表

测试项编号	INT_008_001_001	
优先级	低	
测试项描述	输入参数为空情况	
前置条件	商家上传商店图片，前端向后端“/api/sto/uploadStolmage”接口上传请求	
用例序号	输入	期望结果
001	sto_id=""	输入错误，方法调用失败，返回报错信息

测试项编号	INT_008_001_002	
优先级	低	
测试项描述	输入参数不合法情况	
前置条件	商家上传商店图片，前端向后端“/api/sto/uploadStolmage”接口上传请求	
用例序号	输入	期望结果
001	sto_id="sssss"	输入错误，方法调用失败，返回报错信息

测试项编号	INT_008_001_003	
优先级	低	
测试项描述	输入商家不存在情况	
前置条件	商家上传商店图片，前端向后端“/api/sto/uploadStolmage”接口上传请求	
用例序号	输入	期望结果
001	sto_id="10"	商家不存在，返回服务端错误信息

测试项编号	INT_008_001_004	

优先级	高	
测试项描述	输入商家合法且存在情况	
前置条件	商家上传商店图片，前端向后端“/api/sto/uploadStoImage”接口上传请求	
用例序号	输入	期望结果
001	sto_id="41"	商家存在，返回上传成功提示

4.9 INT_008_002测试用例设计

4.9.1 设计标识符

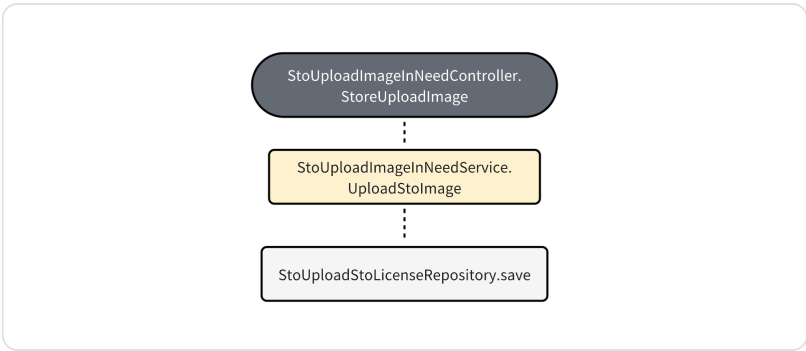
INT_008_002

4.9.2 被测特性

- 输入参数为空时，方法调用失败，返回错误信息
- 输入参数商家ID非法时，返回错误信息
- 输入参数商家ID合法但不存在时，返回错误信息
- 输入商家合法且存在时，返回商家营业执照上传成功信息

4.9.3 测试方法

分析StoUploadImageInNeedController函数的调用图，函数StoreUploadImage接受前端传递图片和信息调用StoUploadImageInNeedService.UploadImage函数实现上传商家宣传图操函数调用，Service层StoUploadStoLicenseRepository.save函数将数据存储在数据库中，最终实现上传商家资质图片功能。



函数调用单向调用，当调用该接口时，执行输入商家合法且存在的用例时，即可完成调用接口的全覆盖，使用强一般等价类进行测试用例的分析和设计，可以划分为上述被测特性的四种等价类情况，分别进行测试用例的设计。

4.9.4 测试项标识

测试项标识符	测试项目描述	优先级
INT_008_002_001	输入参数为空时，方法调用失败情况	低
INT_008_002_002	输入参数商家ID非法情况	低
INT_008_002_003	输入参数商家ID合法但不存在情况	低
INT_008_002_004	输入商家合法且存在时情况	高

4.9.5 测试通过/失败标准

- 测试用例集中，函数调用的接口及调用接口的路径至少覆盖一次。
- 所有的测试用例都必须被执行，且没有发现缺陷。

4.9.6 测试用例表

测试项编号	INT_008_002_001	
优先级	低	
测试项描述	输入参数为空情况	
前置条件	商家上传商家营业执照，前端向后端“/api/sto/uploadStoLicense”接口上传请求	
用例序号	输入	期望结果
001	sto_id=""	输入错误，方法调用失败，返回报错信息

测试项编号	INT_008_002_002	
优先级	低	
测试项描述	输入参数不合法情况	
前置条件	商家上传商家营业执照，前端向后端“/api/sto/uploadStoLicense”接口上传请求	
用例序号	输入	期望结果
001	sto_id="sssss"	输入错误，方法调用失败，返回报错信息

测试项编号	INT_008_002_003	
-------	-----------------	--

优先级	低	
测试项描述	输入商家不存在情况	
前置条件	商家上传商家营业执照，前端向后端“/api/sto/uploadStoLicense”接口上传请求	
用例序号	输入	期望结果
001	sto_id="10"	商家不存在，返回服务端错误信息

测试项编号	INT_008_002_004	
优先级	高	
测试项描述	输入商家合法且存在情况	
前置条件	商家上传商家营业执照，前端向后端“/api/sto/uploadStoLicense”接口上传请求	
用例序号	输入	期望结果
001	sto_id="41"	商家存在，返回上传成功提示

4.10 INT_008_003测试用例设计

4.10.1 设计标识符

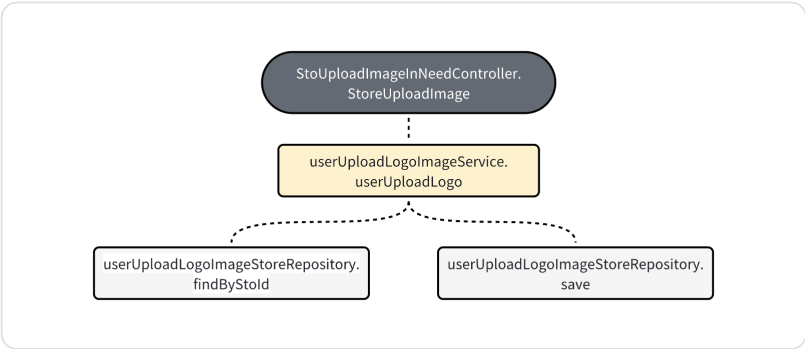
INT_008_003

4.10.2 被测特性

- 输入参数为空时，方法调用失败，返回错误信息
- 输入参数商家ID非法时，返回错误信息
- 输入参数商家ID合法但不存在时，返回错误信息
- 输入商家合法且存在时，返回商标上传成功信息

4.10.3 测试方法

分析StoUploadImageInNeedController函数的调用图，函数StoreUploadImage接受前端传递图片和信息调用StoUploadImageInNeedService.userUploadLogo函数实现上传商家宣传图操函数调用，Service层函数调用userUploadLogoImageStoreRepository.findByStold函数查找商家实体，根据查找到的商家实体Service调用StoUploadStoImageStoreRepository.save函数将数据存储在数据库中，最终实现上传商家资质图片功能。



当调用该接口时，执行输入商家合法且存在的用例时，即可完成调用接口的全覆盖，使用强一般等价类进行测试用例的分析和设计，可以划分为上述被测特性的四种等价类情况，分别进行测试用例的设计。

4.10.4 测试项标识

测试项标识符	测试项目描述	优先级
INT_008_003_001	输入参数为空时，方法调用失败情况	低
INT_008_003_002	输入参数商家ID非法情况	低
INT_008_003_003	输入参数商家ID合法但不存在情况	低
INT_008_003_004	输入商家合法且存在时情况	高

4.10.5 测试通过/失败标准

- 测试用例集中，函数调用的接口及调用接口的路径至少覆盖一次。
- 所有的测试用例都必须被执行，且没有发现缺陷。

4.10.6 测试用例表

测试项编号	INT_008_003_001	
优先级	低	
测试项描述	输入参数为空情况	
前置条件	商家上传商家商标，前端向后端“/api/sto/uploadLogolmage”接口上传请求	
用例序号	输入	期望结果
001	sto_id=""	输入错误，方法调用失败，返回报错信息

--	--

测试项编号	INT_008_003_002	
优先级	低	
测试项描述	输入参数不合法情况	
前置条件	商家上传商家商标，前端向后端“/api/sto/uploadLogoImage”接口上传请求	
用例序号	输入	期望结果
001	sto_id="sssss"	输入错误，方法调用失败，返回报错信息

测试项编号	INT_008_003_003	
优先级	低	
测试项描述	输入商家不存在情况	
前置条件	商家上传商家商标，前端向后端“/api/sto/uploadLogoImage”接口上传请求	
用例序号	输入	期望结果
001	sto_id="10"	商家不存在，返回服务端错误信息

测试项编号	INT_008_003_004	
优先级	高	
测试项描述	输入商家合法且存在情况	
前置条件	商家上传商家商标，前端向后端“/api/sto/uploadLogoImage”接口上传请求	
用例序号	输入	期望结果
001	sto_id="41"	商家存在，返回上传成功提示

4.11 ALI_INT_001测试用例设计

4.11.1 设计标识符

ALI_INT_001

4.11.2 被测特性

- 输入正确参数时，方法调用成功，返回付款页面信息

4.11.3 测试方法

目标为覆盖该接口，仅需测试执行输入正确参数的用例。

4.11.4 测试项标识

测试项标识符	测试项目描述	优先级
ALI_INT_001_001	输入正确参数的情况	低

4.11.5 测试通过/失败标准

- 所有的测试用例都必须被执行，且没有发现缺陷。

4.11.6 测试用例表

测试项编号	ALI_INT_001_001	
优先级	低	
测试项描述	输入正确参数的情况	
前置条件	顾客进入下单付款页面	
用例序号	输入	期望结果
001	userType=0 orderId="1234567890" price="100.00" name="Test Product" cus_id="43"	code=200 success=True paymentUrl=支付宝生成的支付页面Url

第三部分 集成测试执行报告

1. 测试前准备

1.1 测试用例托管

根据上述文档中测试测试用例的设计，我们利用国产测试软件PingCode，将测试用例导入并托管在平台PingCode平台当中。

TJ-software-test

工作台产品管理项目管理测试管理知识管理效能度量协作空间自动化工时管理自建应用

管理后台

临时食品项目测试计划

用例

集成测试模块...

+ 新建用例

全部用例 · 119

用例模块

单元测试模块用例 · 67

集成测试模块用例 · 52

系统测试模块用例 · 0

无模块用例 · 0

搜索 (Ctrl+G)

筛选

排序

52 个用例

	编号	标题	状态	版本	重要程度	用例类型
1	LQS-69	ALI_INT_001_001_001	就绪	v1	P0	接口测试
2	LQS-70	INT_008_003_001_001	就绪	v1	P0	接口测试
3	LQS-71	INT_008_003_002_001	就绪	v1	P0	接口测试
4	LQS-72	INT_008_003_003_001	就绪	v1	P0	功能测试
5	LQS-73	INT_008_003_004_001	就绪	v1	P0	功能测试
6	LQS-74	INT_008_002_001_001	就绪	v1	P0	接口测试
7	LQS-75	INT_008_002_002_001	就绪	v1	P0	接口测试
8	LQS-76	INT_008_002_003_001	就绪	v1	P0	功能测试
9	LQS-77	INT_008_002_004_001	就绪	v1	P0	功能测试
10	LQS-78	INT_008_001_001_001	就绪	v1	P0	接口测试
11	LQS-79	INT_008_001_002_001	就绪	v1	P0	接口测试

显示子模块用例

第 1-50 条 共 52 条

50 条/页

1 2

共 2 页 跳至

页

1.2 测试计划托管

利用PingCode平台进行测试计划的设计，将托管的用例导入集成测试计划中并准备开始进行测试。

TJ-software-test

工作台产品管理项目管理测试管理知识管理效能度量协作空间自动化工时管理自建应用

管理后台

临时食品项目测试计划

计划

普通测试

+ 新建计划

全部计划 · 5

普通测试 · 5

迭代测试 · 0

发布测试 · 0

搜索 (Ctrl+G)

状态

负责人

排序

5 个计划

计划名称	测试分类	状态	结果分布	类别
系统测试计划-非功能性测试	普通测试	未开始	0%	-
集成测试计划-功能性测试	普通测试	未开始	0%	-
集成测试计划	普通测试	进行中	0%	-
单元测试计划	普通测试	已完成	100%	-
静态测试计划	普通测试	已完成	0%	-

第 1-5 条 共 5 条

20 条/页

1

共 1 页 跳至

页

TJ-software-test

工作台

产品管理

项目管理

测试管理

知识管理

效能度量

协作空间

自动化

工时管理

自建应用

管理后台

临时食品项目测试计划

计划

用例

基线

更多

搜索 (Ctrl+K)

进行中

通过率 0%

已测 0 / 52

普通测试 / 集成测试计划

执行用例

缺陷

测试报告

执行用例

搜索 (Ctrl+G)

筛选

排序

导出

52 个用例

规划用例

	编号	标题	执行结果	关联缺陷	重要程度	
用例模块	1	LQS-69	ALL_INT_001_001_001	未测	0	PO
> 集成测试模块用例 · 52	2	LQS-70	INT_008_003_001_001	未测	0	PO
	3	LQS-71	INT_008_003_002_001	未测	0	PO
	4	LQS-72	INT_008_003_003_001	未测	0	PO
	5	LQS-73	INT_008_003_004_001	未测	0	PO
	6	LQS-74	INT_008_002_001_001	未测	0	PO
	7	LQS-75	INT_008_002_002_001	未测	0	PO
	8	LQS-76	INT_008_002_003_001	未测	0	PO
	9	LQS-77	INT_008_002_004_001	未测	0	PO
	10	LQS-78	INT_008_001_001_001	未测	0	PO
	11	LQS-79	INT_008_001_002_001	未测	0	PO

显示子模块用例

第 1-50 条 共 52 条

50 条/页

微信 1 2 >

共 2 页 跳至

页

1.3 Apifox测试工具配置

Apifox是一款强大的接口设计与管理工具，其中提供了对接口的自动化测试功能。首先在Apifox中对待测试接口完成用例的准备，并准备测试的前置操作（如数据库操作）和后置操作（设置断言）。

接口管理

项目概览

接口

根目录

pub (4)

POST INT_004_001 (6)

POST INT_005_001 (7)

INT_005_001_001_001

INT_005_001_002_001

INT_005_001_003_001

INT_005_001_004_001

INT_005_001_005_001

INT_005_001_006_001

INT_005_001_007_001

POST INT_005_002 (10)

POST INT_001_001 (2)

sto (6)

Alipay (1)

数据模型

组件库

快捷请求

回收站

项目概览

POST INT_005_001

INT_005_001 (INT_0...

POST http://localhost:8081/api/pub/register/customer

INT_005_001 (INT_005_001_002_001)

Params

Body 1

Headers

Cookies

Auth

前置操作

后置操作 3

设置

none

form-data

x-www-form-urlencoded

json

xml

raw

binary

GraphQL

msgpack

自动生成

动态值

1 {

2 "user_phone": "13837829689",

3 "user_password": "123456",

4 "user_address": "台湾省台北市中正区",

5 "user_gender": 1,

6 "cus_nickname": "Trump",

7 "cus_payPassword": "111111"

8 }

返回响应

校验响应

正常 (200)

Cookie 管理

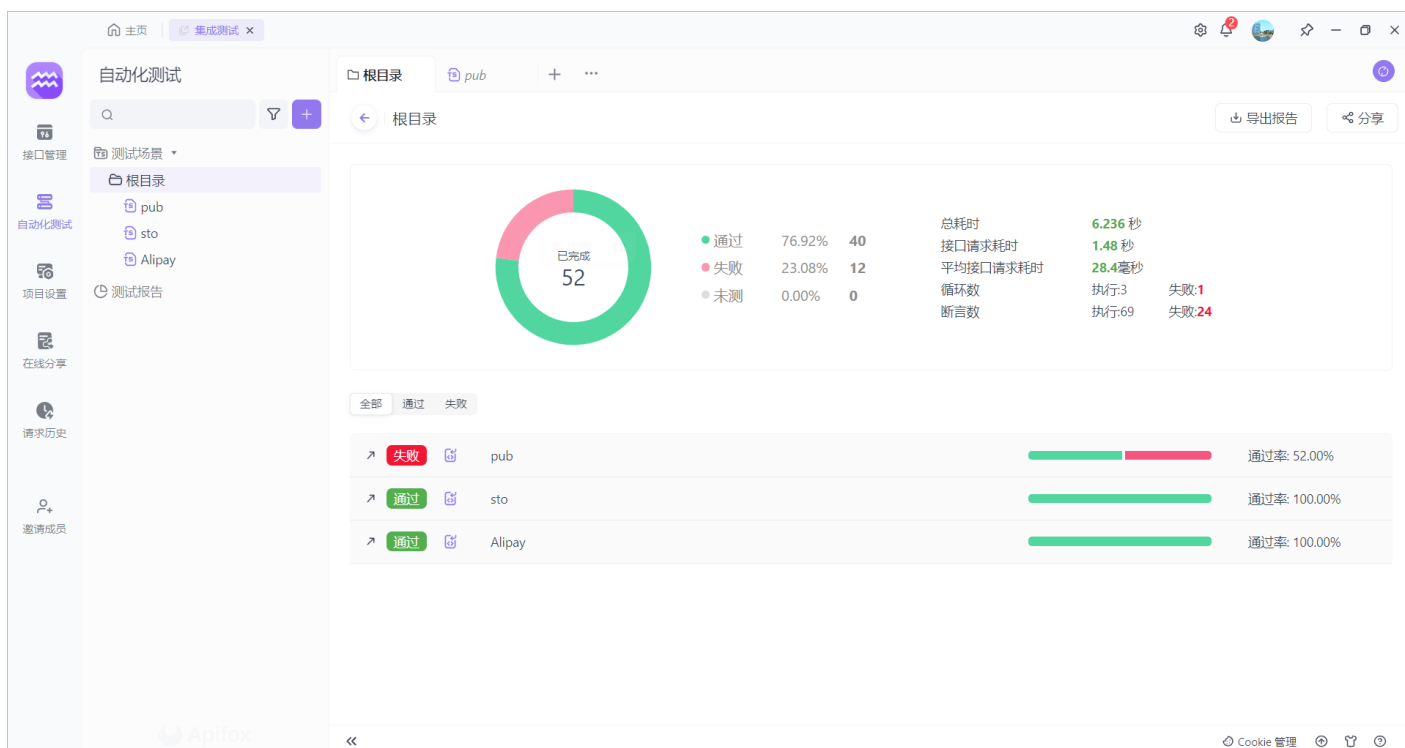
例如，在测试注册接口时，可以在前置操作中规定首先对数据库相关用户信息记录进行检査和删除，以保证每次测试都能正常执行：

数据库操作	删除storeLicense记录 本项目数据库	DELETE FROM storeLicense WHERE sto_ID = (SELECT user_ID FROM user WHERE user_phone = '15815673200');
数据库操作	删除storeImage记录 本项目数据库	DELETE FROM storeImage WHERE sto_ID = (SELECT user_ID FROM user WHERE user_phone = '15815673200');
数据库操作	删除store记录 本项目数据库	DELETE FROM store WHERE sto_ID = (SELECT user_ID FROM user WHERE user_phone = '15815673200');
数据库操作	删除user记录 本项目数据库	DELETE FROM user WHERE user_phone = '15815673200';

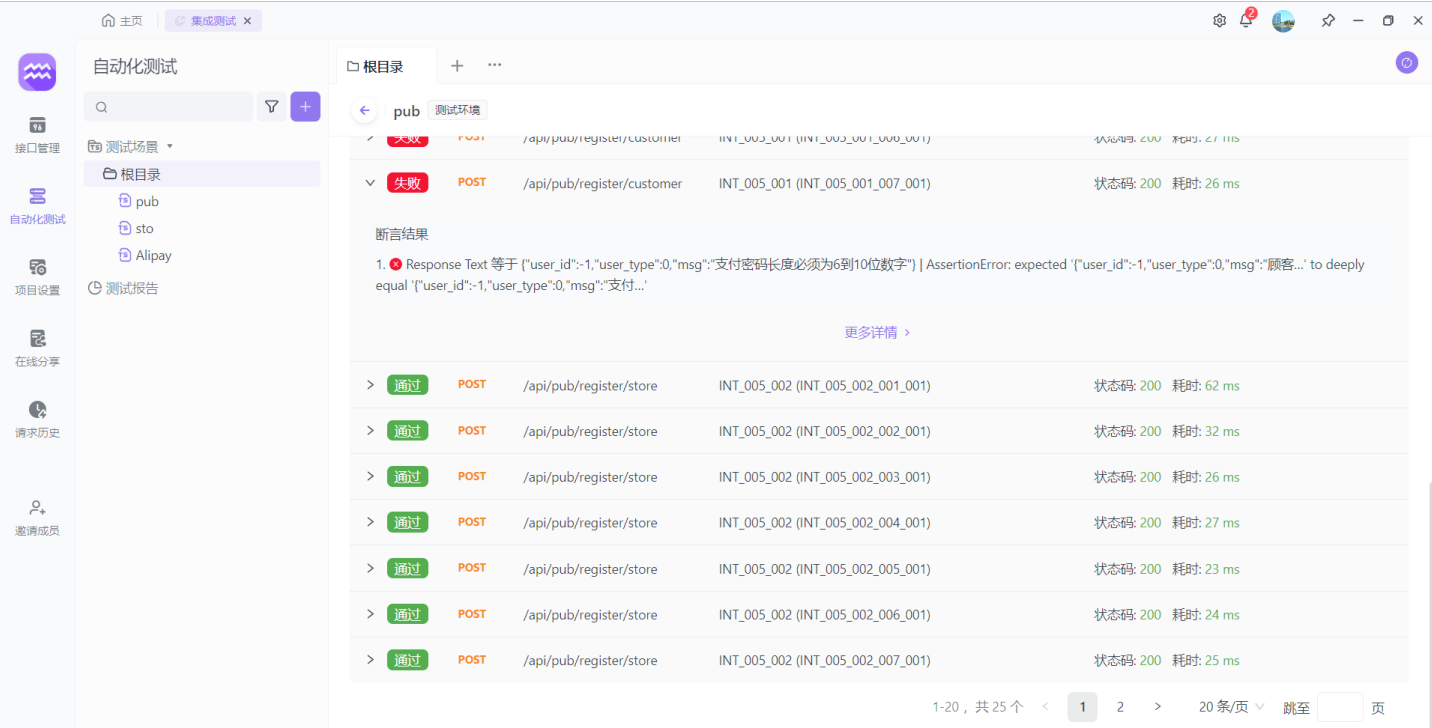
2. 集成测试结果

2.1 Apifox集成测试记录及统计

执行设计的全部52个测试用例，其中通过数为40，失败数为12，通过率为76.92%。



在Apifox工具中，可以进一步查看失败用例的断言结果，协助我们进行缺陷定位。



2.2 测试未通过情况及识别缺陷

测试用例未通过情况：

用例编号	通过情况	用例编号	通过情况
INT_004_001_002_001	未通过	INT_005_001_002_001	未通过
INT_004_001_002_002	未通过	INT_005_001_003_001	未通过
INT_004_001_003_001	未通过	INT_005_001_004_001	未通过
INT_004_001_003_002	未通过	INT_005_001_005_001	未通过
INT_004_001_004_001	未通过	INT_005_001_006_001	未通过
INT_004_001_005_001	未通过	INT_005_001_007_001	未通过

识别出的缺陷情况：

缺陷编号	缺陷情况描述	关联测试用例	缺陷修正负责人
INT_Flaw_001	测试用例覆盖了登录接口，根据 Apifox 生成的报告，返回的状态码为 404，判断缺陷来自代码中接口入口与设计文档不符（即 API 路径不正确）。	INT_004_001_002_001 INT_004_001_002_002 INT_004_001_003_001 INT_004_001_003_002 INT_004_001_004_001	韩嘉睿

		INT_004_001_005_001	
INT_Flaw_002	测试用例覆盖了登录接口，根据Apifox生成的报告，返回的字段断言错误，判断缺陷来自接口返回的消息字段与设计文档不符。	INT_005_001_002_001 INT_005_001_003_001 INT_005_001_004_001 INT_005_001_005_001 INT_005_001_006_001 INT_005_001_007_001	韩嘉睿

2.3 PingCode工具缺陷跟踪记录

- 使用PingCode工具记录测试结果，并设置关联用例

TJ-software-test

临时食品项目测试计划

计划

用例

基线

更多

搜索 (Ctrl+K)

+

?

🔔

佳恒

普通测试 / 集成测试计划

执行用例

缺陷

测试报告

进行中

通过率 77%

已测 52 / 52

⋮

执行用例

Q 搜索 (Ctrl+G)

筛选

排序

导出

52 个用例

规划用例

全部用例 · 52	□	编号	标题	执行结果	关联缺陷	重要程度		⌵
用例模块		36	LQS-104	INT_005_002_009_001	通过	0	P0	
集成测试模块用例 · 52		37	LQS-105	INT_005_002_010_001	通过	0	P0	
		38	LQS-106	INT_005_001_001_001	通过	0	P0	
		39	LQS-107	INT_005_001_002_001	失败	1	P0	
		40	LQS-108	INT_005_001_003_001	失败	1	P0	
		41	LQS-109	INT_005_001_004_001	失败	1	P0	
		42	LQS-110	INT_005_001_005_001	失败	1	P0	
		43	LQS-111	INT_005_001_006_001	失败	1	P1	
		44	LQS-112	INT_005_001_007_001	失败	1	P0	
		45	LQS-113	INT_001_001_001_001	通过	0	P0	
		46	LQS-114	INT_001_001_002_001	通过	0	P0	

显示子模块用例

第 1-50 条 共 52 条

50 条/页

<

1

2

>

共 2 页 跳至

页

- 使用PingCode工具进行缺陷记录与跟踪

TJ-software-test

> 临期食品项目测试计划

计划用例基线更多

搜索 (Ctrl+K)

+ ? 佳恒

普通测试 / 集成测试计划

执行用例缺陷测试报告

进行中 通过率 77% 已测 52 / 52

2个缺陷

+ 新建缺陷

搜索 (Ctrl+G)

筛选排序导出

#	编号	标题	状态	负责人	关联测试
1	LLQ-7	INT_Flaw_001	新提交		6
2	LLQ-8	INT_Flaw_002	新提交		6

第 1-2 条 共 2 条

50 条/页 < 1 > 共 1 页 跳至 页

- 使用PingCode工具导出当前实验报告



3. 回归测试

3.1 对单元测试的回归

根据上述测试结果，我们对识别出的缺陷进行修复，并进行回归测试。首先我们对单元测试进行回归，由于项目体量较小，执行所有单元测试的成本开销较小，我们回归测试的范围即包括所有测试计划中设计的单元测试用例。具体回归测试记录如下，单元测试全部通过，可见修改后代码不影响单元测试结果。

3.1.1 回归测试运行记录

重新执行已测试过的60个单元测试用例，全部通过测试。

✓ service (org.tju.food_007)	1秒 791毫秒	✓ 测试 已通过: 60共 60 个测试 - 1秒 791毫秒
✓ UserLoginServiceTest	1秒 84毫秒	timestamp = 2024-06-21T19:37:05.83
✓ 测试用户登录服务类	1秒 84毫秒	timestamp = 2024-06-21T19:37:05.83
✓ 测试用例 Unit_001_002_001_001_001	1秒 57毫秒	timestamp = 2024-06-21T19:37:05.83
✓ 测试用例 Unit_001_002_001_002_001	5毫秒	timestamp = 2024-06-21T19:37:05.98
✓ 测试用例 Unit_001_002_001_002_002	5毫秒	开始测试
✓ 测试用例 Unit_001_002_001_003_001	6毫秒	timestamp = 2024-06-21T19:37:05.98
✓ 测试用例 Unit_001_002_001_003_002	6毫秒	timestamp = 2024-06-21T19:37:05.98
✓ 测试用例 Unit_001_002_001_004_001	5毫秒	timestamp = 2024-06-21T19:37:05.98
✓ GenerateIndentServiceTest	392毫秒	timestamp = 2024-06-21T19:37:05.99
✓ Unit_004_002_001_生成订单测试用例	392毫秒	开始测试
✓ 测试用例 Unit_004_002_001_001_001	268毫秒	timestamp = 2024-06-21T19:37:05.99
✓ 测试用例 Unit_004_002_001_002_001	10毫秒	timestamp = 2024-06-21T19:37:05.99
✓ 测试用例 Unit_004_002_001_002_002	8毫秒	timestamp = 2024-06-21T19:37:05.99
✓ 测试用例 Unit_004_002_001_002_003	7毫秒	timestamp = 2024-06-21T19:37:05.99
✓ 测试用例 Unit_004_002_001_003_001	9毫秒	timestamp = 2024-06-21T19:37:06.00
✓ 测试用例 Unit_004_002_001_004_001	12毫秒	开始测试
✓ 测试用例 Unit_004_002_001_004_002	5毫秒	timestamp = 2024-06-21T19:37:06.00
✓ 测试用例 Unit_004_002_001_004_003	6毫秒	timestamp = 2024-06-21T19:37:06.00
✓ 测试用例 Unit_004_002_001_005_001	5毫秒	timestamp = 2024-06-21T19:37:06.00
✓ 测试用例 Unit_004_002_001_006_001	6毫秒	timestamp = 2024-06-21T19:37:06.00

3.1.2 Allure统计结果展示

使用Allure工具，对上述单元测试结果展示如下。



3.1.3 PingCode工具记录

PingCode工具记录结果图所示。

TJ-software-test

工作台

产品管理

项目管理

测试管理

知识管理

效能度量

协作空间

自动化

工时管理

自建应用

管理后台

> 临期食品项目测试计划

计划用例基线更多

搜索 (Ctrl+K)

+ ? 佳恒

普通测试 / 单元测试计划

执行用例缺陷测试报告

已完成通过率 100% 已测 60 / 60

执行用例

搜索 (Ctrl+G) 筛选排序导出60 个用例

全部用例 · 60

用例模块

单元测试模块用例 · 60

	编号	标题	执行结果	关联缺陷	重要程度	
1	LQS-9	Unit_002_002_001_001_001	通过	0	P0	
2	LQS-10	Unit_002_002_001_002_001	通过	0	P0	
3	LQS-11	Unit_002_002_001_003_001	通过	0	P0	
4	LQS-12	Unit_002_002_001_003_002	通过	0	P0	
5	LQS-13	Unit_002_002_001_003_003	通过	0	P0	
6	LQS-14	Unit_002_002_001_004_001	通过	0	P0	
7	LQS-15	Unit_002_002_001_004_002	通过	0	P0	
8	LQS-16	Unit_002_002_001_005_001	通过	0	P0	
9	LQS-17	Unit_002_002_001_006_001	通过	0	P0	
10	LQS-18	Unit_002_002_001_006_002	通过	0	P0	
11	LQS-19	Unit_002_002_001_007_001	通过	0	P0	

显示子模块用例

第 1-50 条 共 60 条

50 条/页

12 >

共 2 页 跳至 页

TJ-software-test

工作台

产品管理

项目管理

测试管理

知识管理

效能度量

协作空间

自动化

工时管理

自建应用

管理后台

> 临期食品项目测试计划

计划用例基线更多

搜索 (Ctrl+K)

+ ? 佳恒

普通测试 / 单元测试计划

执行用例缺陷测试报告

已完成通过率 100% 已测 60 / 60

测试报告

导出打印配置

报告概况

60用例数

42.55 %用例覆盖率

100.0 %用例通过率

4缺陷数

测试库: 临期食品项目测试计划

测试计划: 单元测试计划

计划开始: 2024年6月21日

计划结束: 2024年7月5日

实际开始: 2024年6月21日

实际结束: 2024年6月21日

状态: 已完成

负责人: 佳恒 傅佳恒

关联项目: 邻临七企划-基于15分钟生活圈的临期食品智慧社区

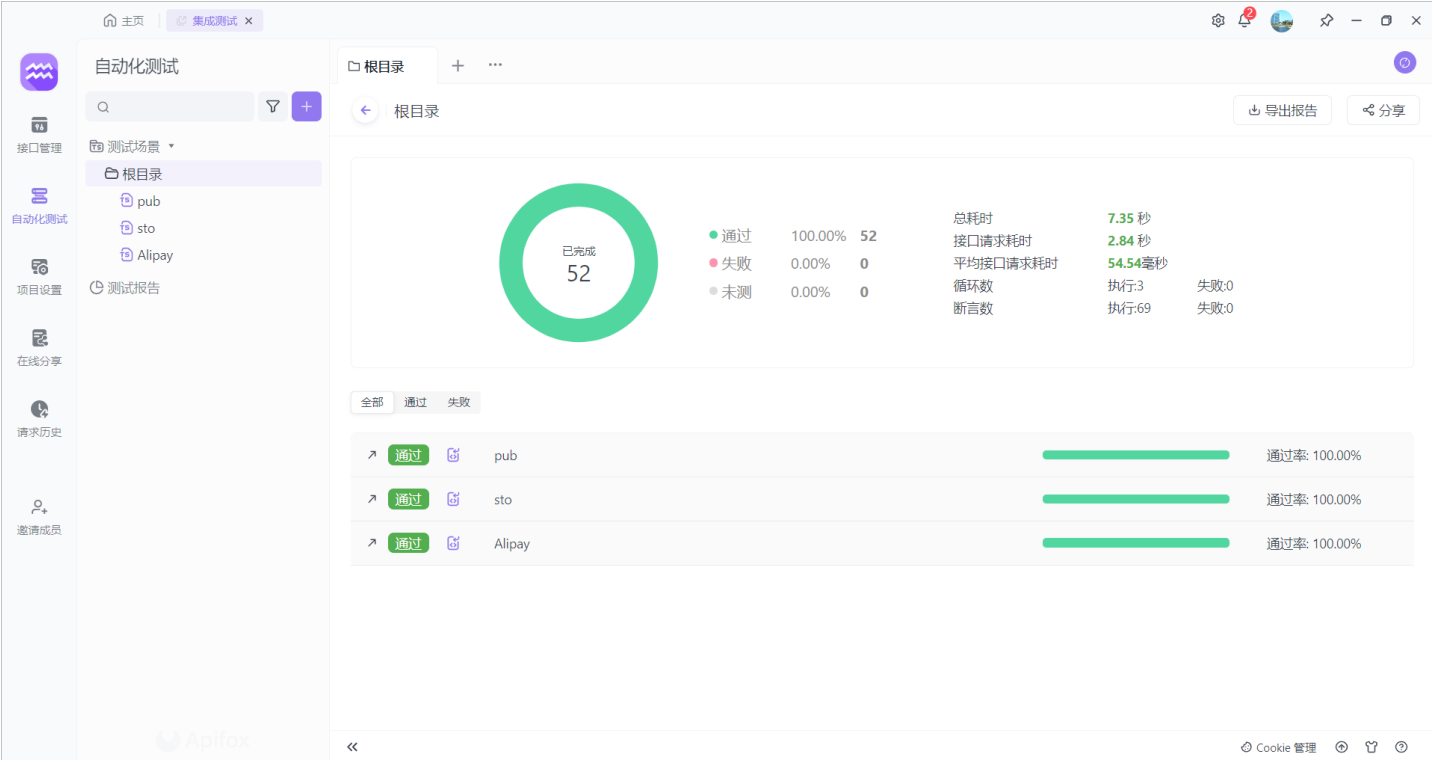
关联迭代: -

报告总结

3.2 对集成测试的回归

3.2.1 ApiFox统计结果测试

通过回归测试，ApiFox中测试用例全部通过测试。



TJ-software-test

工作台

产品管理

项目管理

测试管理

知识管理

效能度量

协作空间

自动化

工时管理

自建应用

管理后台

临时食品项目测试计划

计划

用例

基线

更多

搜索 (Ctrl+K)

佳恒

普通测试 / 集成测试计划

执行用例

缺陷

测试报告

进行中

通过率 100%

已测 52 / 52

执行用例

搜索 (Ctrl+G)

筛选

排序

导出

52 个用例

规划用例

	编号	标题	执行结果	关联缺陷	重要程度	
用例模块	1	LQS-69 ALL_INT_001_001_001	通过	0	P0	
> 集成测试模块用例 · 52	2	LQS-70 INT_008_003_001_001	通过	0	P0	
	3	LQS-71 INT_008_003_002_001	通过	0	P0	
	4	LQS-72 INT_008_003_003_001	通过	0	P0	
	5	LQS-73 INT_008_003_004_001	通过	0	P0	
	6	LQS-74 INT_008_002_001_001	通过	0	P0	
	7	LQS-75 INT_008_002_002_001	通过	0	P0	
	8	LQS-76 INT_008_002_003_001	通过	0	P0	
	9	LQS-77 INT_008_002_004_001	通过	0	P0	
	10	LQS-78 INT_008_001_001_001	通过	0	P0	
	11	LQS-79 INT_008_001_002_001	通过	0	P0	

显示子模块用例

第 1-50 条 共 52 条

50 条/页

1 2

共 2 页 跳至 页

记录缺陷修复记录

TJ-software-test

工作台

产品管理

项目管理

测试管理

知识管理

效能度量

协作空间

自动化

工时管理

自建应用

管理后台

临期食品项目测试计划

计划

用例

基线

更多

搜索 (Ctrl+K)

普通测试 / 集成测试计划

执行用例

缺陷

测试报告

进行中

通过率 100%

已测 52 / 52

搜索 (Ctrl+G)

筛选

排序

导出

2 个缺陷

新建缺陷

#	编号	标题	状态	负责人	关联测试		
1	LLQ-7	INT_Flaw_001	已修复		6		
2	LLQ-8	INT_Flaw_002	已修复		6		

第 1-2 条 共 2 条

50 条/页

1

共 1 页 跳至

页

● 设置记录缺陷版本

LLQ-8

INT_Flaw_002

负责人 未设置

状态 已修复

开始时间 -

截止时间 -

基本信息

产品需求

关联

工单

测试 6

版本记录 2

2 个版本

创建新版本

版本	备注	创建人	创建时间
v2	集成测试回归测试通过	傅佳恒	06月21日
v1	初始化创建	傅佳恒	06月21日

属性

优先级 最高

严重程度 致命

故事点 -

复现发布号 -

复现概率 -

缺陷类别 -

原因分析 -

解决方案 -

标签 + 标签

变更

关注 傅佳恒

已关注

● 测试报告更新



4. 集成测试退出

4.1 集成测试退出标准评估

根据对集成测试结果进行评审，测试满足测试覆盖标准，覆盖所有的调用路径，成功执行所有集成测试用例并纠正缺陷。满足单元测试退出标准。同时，项目已完成集成测试计划、集成测试计划用例规格、集成测试用例脚本、集成测试驱动和桩代码、集成测试执行日志及集成测试报告的整理与交付。

4.2 PingCode集成测试结束

