

软件测试文档-系统测试

1. 系统需求

1.1 引言

1.1.1 目的

为了明确用户的需求并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，并作为整个软件开发的基础。

1.1.2 背景

近年来，游戏产业不断发展，许多精美的游戏上线，改变了人们的娱乐方式，随之而来的，对游戏零售平台的需求也不断上升。StarGames 是由我们团队开发的一个游戏信息管理系统，依据一定的隐私条例来供用户在 StarGames 平台上进行游戏的发布与购买，以此方便用户对自己的游戏进行管理。当今，随着游戏数量的不断增长，用户需要处理大量的游戏数据信息，从而快捷地管理自己所拥有的游戏并进行各种操作，用户同时也需要一个集中的平台来了解当前游戏如评价、价格和销量等各种信息，与好友进行交流；管理员也需要一个可以快速管理买家与商品的途径。

我们的游戏管理系统主要面向以下三种人群，为买家提供查看各种游戏信息、购买游戏、收藏游戏至购物车、与好友聊天等功能；为卖家提供发布游戏、发布更新公告、查看已发布游戏等功能。为管理员提供一系列便于管理的功能，如下架商品、封禁买家等等；此外，用户也可以通过主界面的游戏信息来了解当季热销的游戏商品。

1.1.3 参考资料

GB/T 15532-2008 计算机软件测试规范

GB/T 39788-2021 系统与软件工程 性能测试方法

GB/T 9385-2008 计算机软件需求规格说明规范

1.1.4 术语

1.2 项目概述

1.2.1 开发软件的一般描述

使用scrum过程进行管理的在线游戏销售平台，买家可以在商城当中注册，然后在商城当中挑选自己想要的游戏加入购物车、加入愿望单、购买游戏、查看已发布游戏、查看游戏详情，添加好友，删除好友，发布游戏评论；卖家可以查看自己发行的游戏、上架自己发行的游戏；管理员可以下架商城当中下架的游戏、封禁买家

1.2.2 开发软件的功能描述

该软件是一个24小时实时服务系统，可以划分为八个子系统：登录注册子系统、管理交易子系统、个人中心子系统、游戏库管理子系统、游戏发布子系统、信息管理子系统、游戏管理子系统、好友管理子系统。账号分为三类：一类是管理员，用于系统的最高权限，一类是商家，对本系统当中的游戏进行维护和上架，另一类是买家，可以在商城当中浏览和购买游戏，并且和好友进行互动。

1.2.3 实现语言

后端使用Java语言进行开发，前端使用Vue框架，其中主要用到了JavaScript语言。

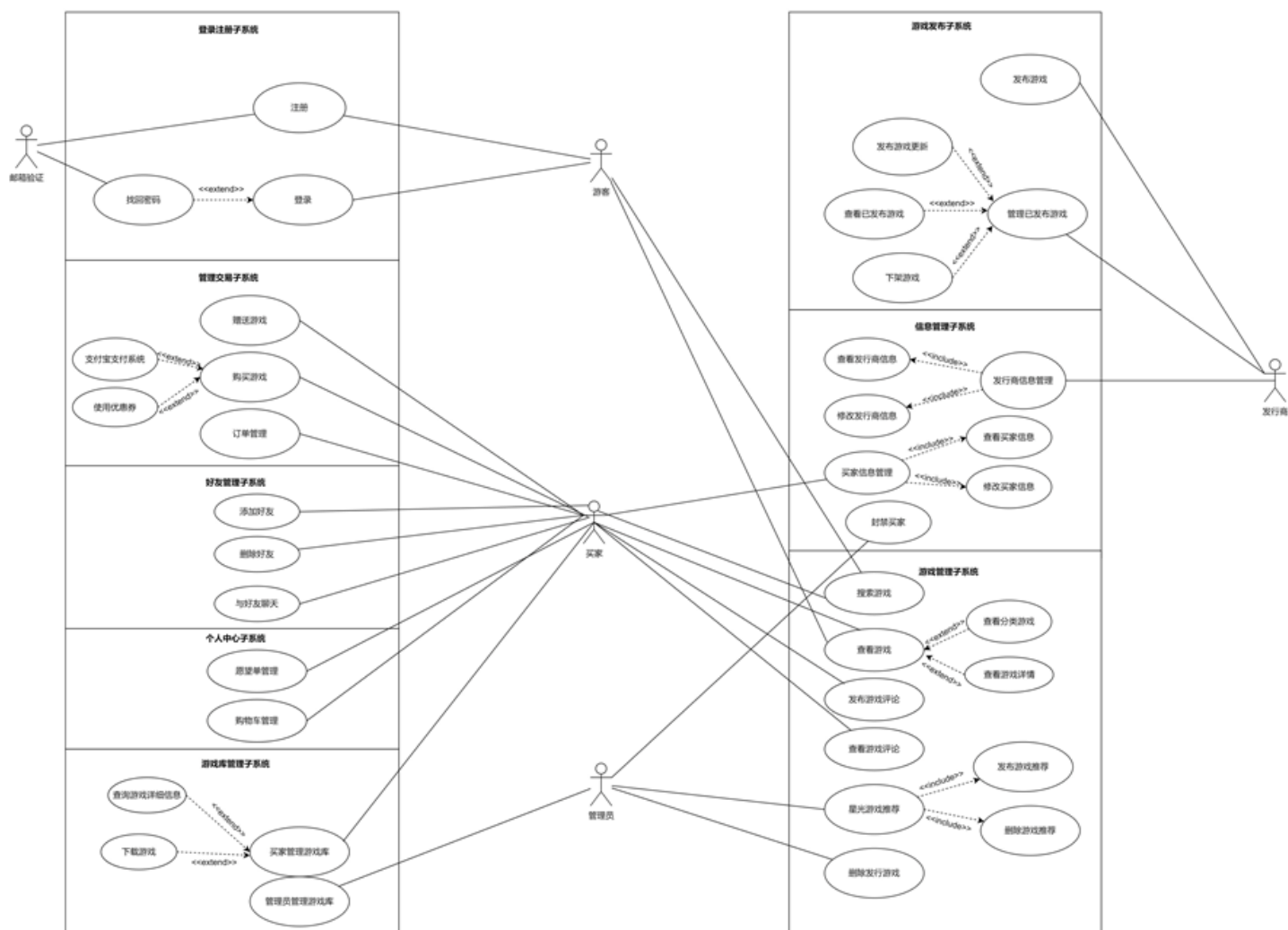
1.2.4 用户特点

本软件的主要用户为目前使用游戏平台较多的游戏发行商和用户，大多都有游戏购买，游戏维护的经验，另外，系统要实现的一个重要目标就是界面友好性和易操作性。即使是一个对游戏商城完全陌生的客户，也可以顺利完成游戏的选择和购买。另一类用户是管理员。

1.3 需求说明

基本描述

本系统为买家、游戏发行商、管理员三种角色提供服务，并对外提供注册功能。登陆后，买家可以购买游戏、赠送游戏、查看订单、管理愿望单、管理购物车、管理游戏库、搜索游戏、评论游戏、添加好友、删除好友、和好友聊天等功能；游戏发行商可以发布游戏、发布游戏更新、下架游戏；管理员可以封禁买家、封禁卖家、删除发行游戏。



1. 用户信息：用户 ID，用户名，密码，用户类型
2. 商品信息：商品 ID，商品名称，发行商，价格，发行时间，分类，商品描述，图片，下载链接，销量
3. 买家信息：联系方式、注册时间、生日
4. 游戏发行商信息：开店时间、卖家简介
5. 管理员信息：部门
6. 通知信息：通知 ID，标题，内容，时间
7. 联系客服信息：联系编号
8. 评论信息：评论内容和评论时间
9. 订单信息：订单编号，交易时间
10. 更新日志信息：更新时间、更新内容
11. 礼品码信息：激活码，生成时间，使用情况
12. 验证信息：备注，操作类型

1.3.1 功能需求

1. 描述：用户进入在线考试系统起始页。

输入：用户输入邮箱、用户名、密码。

处理：检测邮箱、电话号码是否已被注册过。

输出：注册过提示已经被注册，否则注册成功。

2. 描述：已注册用户进入登录页面。

输入：用户名和密码。

处理：系统收到输入的信息后，发送至后台进行验证，验证通过则登录成功，验证失败则发送提示信息，要求 用户重新输入。

输出：返回主页。

3. 描述：用户进入修改信息页。

输入：用户输入新信息。

处理：系统将新信息发送到数据库。

输出：返回用户的新信息页。

4. 描述：用户进入主页。

输入：用户点击主页。

处理：系统从后端加载数据。

输出：返回首页商品信息。

5. 描述：用户进入游戏详情页。

输入：用户点击游戏详情。

处理：系统从后端加载数据。

输出：返回游戏的详情信息。

6. 描述：用户将游戏加入购物车。

输入：用户点击游戏详情页当中的将游戏加入购物车。

处理：系统将该信息发送到数据库。

输出：提示用户加入购物车成功。

7. 描述：用户将游戏加入愿望单。

输入：用户点击游戏详情页当中的将游戏加入愿望单。

处理：系统将该信息发送到数据库。

输出：提示用户加入愿望单成功。

8. 描述：用户添加好友。

输入：用户输入添加的好友的ID。

处理：系统查询对应的用户ID，如果查询到，则返回对应的用户信息，如果查询不到该用户，则提示用户查询不到用户。

输出：错误信息或者查询到的好友。

9. 描述：用户删除好友。

输入：用户删除添加的好友的ID。

处理：系统在用户的好友表当中查询对应的用户ID，如果查询到，则返回对应的用户信息，如果查询不到该用户，则提示用户查询不到用户。

输出：错误信息或者查询到的好友。

10. 描述：用户找回密码。

输入：用户要找回的账号以及其对应的邮箱，以及用户对应的新密码。

处理：系统给用户对应的邮箱发送一封带有验证码的邮件，之后判断用户输入的验证码是否和发送的相同，如果相同的话，就提示用户输入新密码，然后将用户的新密码存储到数据库当中。

输出：错误信息或者查询到的好友。

11. 描述：用户购买游戏。

输入：用户要购买的游戏以及用户ID。

处理：系统和支付宝系统进行交互，通过拉起支付宝界面来实现支付，如果用户支付完成之后，将游戏添加到用户的游戏库当中。

输出：购买成功或者失败的信息。

12. 描述：用户赠送游戏。

输入：用户要赠送的游戏以及要赠送的用户ID。

处理：系统和支付宝系统进行交互，通过拉起支付宝界面来实现支付，如果用户支付完成之后，将游戏添加到用户要赠送的用户的游戏库当中。

输出：购买成功或者失败的信息。

13. 描述：用户查看订单。

输入：用户点击查看订单按钮，进入订单查看界面。

处理：系统与后端进行交互，获取数据。

输出：用户的所有订单信息。

14. 描述：用户查看游戏库。

输入：用户点击查看游戏库按钮，进入游戏库查看界面。

处理：系统与后端进行交互，获取数据。

输出：用户所拥有的所有游戏信息。

15. 描述：发行商查看已发行游戏。

输入：发行商点击查看已发行游戏按钮，进入已发行游戏查看界面

处理：系统与后端进行交互，获取数据。

输出：发行商所发行的所有游戏信息。

16. 描述：发行商发布游戏。

输入：游戏的ID，描述，定价，下载连接等一系列数据

处理：系统将这些数据存储在数据库当中。

输出：发布游戏成功或者失败。

17. 描述：发行商发布游戏更新。

输入：游戏更新的数据

处理：系统将这些数据存储在数据库当中。

输出：发布游戏更新成功或者失败。

18. 描述：发行商下架游戏。

输入：要下架的游戏ID

处理：系统将要下架的游戏的标记位置为0，并存入数据库当中。

输出：下架游戏成功或者失败。

19. 描述：发行商管理信息。

输入：发行商要更改的个人信息

处理：系统将这些数据存储在数据库当中。

输出：管理个人信息成功或者失败。

20. 描述：封禁买家。

输入：要封禁的买家ID

处理：系统将要买家ID是否有效的标记位置为0，并存入数据库当中。

输出：封禁买家成功或者失败。

21. 描述：搜索游戏

输入：要搜索的游戏ID或者名称

处理：系统和后端交互从后端获取信息。

输出：搜索到的游戏结果。

22. 描述：和好友聊天。

输入：用户进入和对应好友的聊天界面

处理：系统通过后端将用户发送的聊天内容发送到好友当中。

输出：用户发送的聊天。

1.3.2 非功能需求

1. 安全性需求

- 为保护系统的可靠性，通过严格的权限访问控制，使得用户只能进行其权限范围内的操作，不同用户选择不同模式，以保护数据不被非法访问以及恶意篡改。系统能够提供最基本的输入数据检查，能够处理一些异常情况，如人为操作的错误和硬件失败等。
- 为保护系统的保密性，所有用户的个人基本信息，信息公开，授权人员可以更新；所有用户的私密性信息，信息保密，数据库人员维护更新。
- 为保证用户核心安全性，采用Token身份验证机制。前端每次在登录的过程中将会由后端生成一段Token信息，并存储到Redis数据库中。而当用户的操作涉及到与个人信息有关的时候，在发送API请求的过程中，将会自动将当前Token信息放在Request请求的头部。后端在接收到请求后，会先检验该Token信息是否过期。如果过期或者该用户无权限进行该操作，则返回值的errorCode码将会取400。反之，返回该请求的结果。

2. 性能与兼容性需求

- 要保证较快的响应时间，一般响应时间要控制在1秒以内；
- 支持多并发高并发，配置简便，启动较快，支持众多主流操作系统。
- 支持大量用户同时使用，对系统资源的占用极小。
- 系统对环境兼容性好，大部分代码都不针对特定的操作系统与运行环境，支持打包应用，环境需求较少。

3. 可测试性需求

- 可测试性良好，复杂度不高，系统必须通过单元测试，并完全测试覆盖，开发运用回归测试的方式，允许在一段时间内重新进行完整的测试。
- 系统应在保证当前业务正常运作的情况下实现各模块的独立测试，保证测试的独立性和方便性。

4. 可维护性需求

- 系统上线后会持续运行，当接收到修改请求的时候，能够快速完成，不大的请求能在一周内完成，并保证系统的运行和数据库的内容保持不变。
- StarGames系统的前后端代码保证了开放-封闭原则，在进行业务的扩充和业务的修改时尽量保证现有模块无需进行大幅度修改。

5. 其他需求

除功能性需求和非功能性需求外，本系统还需满足相关的法律和政策需求。

- 系统应进行“防沉迷”设定，为18岁以下的未成年用户进行实名认证，防止未成年人沉迷游戏；
- 系统应该对用户的重要隐私信息，如邮箱、手机号，进行保护和加密，防止数据泄露和黑客攻击；
- 系统应对游戏的内容、评论的内容以及买家和发行商的个人信息进行审核，防止出现违反国家法律法规的内容。

2. 采用的测试方法

系统从如下方面进行测试分析并结合具体的用例设计技术设计测试用例：

- 系统级功能
- 系统业务流
- 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
- 系统非功能

由于运用这些方法分析并设计的测试用例有交叉重复的可能，因此，测试用例集需要经过严格的评审，除去不合理的和重复的测试用例。另外，可能由于没有达到某种覆盖要求或覆盖不足还需要补充一些测试用例。

3. 测试环境

- 硬件需求
 - 测试主机一台
- 软件需求
 - 应用软件测试版
 - Windows 11 操作系统
 - Ubuntu 20.04 操作系统
 - MySQL 数据库
 - JMeter
- 网络需求
 - 100MB以上的带宽

4. 测试对象

StarGames星游游戏零售平台系统。应用软件版本：sys_test_tag（系统测试版本标签）

5. 覆盖业务功能

| 子系统名称 | 功能点 | 测试类型 |
|-----------|------|------|
| 用户登录注册子系统 | 注册账号 | 功能测试 |

| | | |
|----------|----------|------|
| | 登录账号 | 功能测试 |
| | 找回密码 | 功能测试 |
| 管理交易子系统 | 购买游戏 | 功能测试 |
| | 赠送游戏 | 功能测试 |
| | 订单管理 | 功能测试 |
| 个人中心子系统 | 愿望单管理 | 功能测试 |
| | 购物车管理 | 功能测试 |
| 游戏库管理子系统 | 买家管理游戏库 | 功能测试 |
| | 管理员管理游戏库 | 功能测试 |
| 游戏发布子系统 | 发布游戏 | 功能测试 |
| | 发布游戏更新 | 功能测试 |
| | 下架游戏 | 功能测试 |
| 信息管理子系统 | 发行商信息管理 | 功能测试 |
| | 买家信息管理 | 功能测试 |
| | 封禁买家 | 功能测试 |
| 游戏管理子系统 | 搜索游戏 | 功能测试 |
| | 查看游戏 | 功能测试 |
| | 发布游戏评论 | 功能测试 |
| | 查看邮箱评论 | 功能测试 |
| | 星光游戏推荐 | 功能测试 |
| | 删除发行游戏 | 功能测试 |
| 好友管理子系统 | 添加好友 | 功能测试 |
| | 删除好友 | 功能测试 |
| | 与好友聊天 | 功能测试 |

6. 测试分析与用例设计

| | A | B | C | D | E | F | G | H | I |
|---|-----------|--|----------------------------------|----------------------------|----------------------------------|---------------------------------------|---------------------------------|-----------------------------|---|
| 1 | 用例编号 | SG_Sys_Test_case_fun_001 | | | | | | | |
| 2 | 测试覆盖的系统功能 | 1.注册 2.以买家身份登录 3.以卖家身份登录 4.以管理员身份登录 5.登出 | | | | | | | |
| 3 | 用例设计方法 | 场景法 | | | | | | | |
| 4 | 前置条件 | 无 | | | | | | | |
| 5 | 输入 | | | | | | | | |
| 6 | 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 | 预期交互输入、输出6 | 预期交互输入、输出7 | 预期交互输入、输出8 |
| 7 | | 输入：用户进入注册界面 输出：成功进入注册界面 | 输入：用户输入手机号，密码，邮箱等基本信息 输出：注册成功 | 输入：用户输入正确的账号和密码 输出：登录成功 | 输入：用户输入不存在的账号 输出：提示账号不存在，登录失败 | 输入：用户在注册时输入已经被注册过的手机号 输出：提示该手机号已注册 | 输入：用户登陆时输入错误的密码 输出：密码错误，登录失败 | 输入：用户点击登出按钮 输出：清除登录信息，登出 | 输入：用户在注册的时候输入的密码前后不相同 输出：提示用户两次输入的密码不相同，注册失败 |
| 8 | 最后预期输出 | | | | | | | | |
| 9 | | | | | | | | | |

| | A | B | C | D | E | F | G | H |
|---|-----------|--|---|---|---------------------------------|------------|------------|------------|
| 1 | 用例编号 | SG_Sys_Test_case_fun_002 | | | | | | |
| 2 | 测试覆盖的系统功能 | 1.购买游戏 2.赠送游戏 3.订单管理 | | | | | | |
| 3 | 用例设计方法 | 场景法 | | | | | | |
| 4 | 前置条件 | 用户以买家身份已经登录 | | | | | | |
| 5 | 输入 | | | | | | | |
| 6 | 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 | 预期交互输入、输出6 | 预期交互输入、输出7 |
| 7 | | 输入：用户选择购物车当中的商品，点击购买游戏按钮，并且成功支付 输出：支付成功，将游戏添加到游戏库 | 输入：用户选择购物车当中的商品，点击购买游戏按钮，但支付失败 输出：支付失败 | 输入：用户选择购物车当中的要赠送的游戏和好友，点击赠送游戏按钮，并成功支付 输出：支付成功，游戏赠送成功 | 输入：用户点击订单管理按钮 输出：系统展示用户的所有订单 | 无 | 无 | 无 |
| 8 | 最后预期输出 | | | | | | | |
| 9 | | | | | | | | |

| | A | B | C | D | E | F | G | H |
|---|-----------|---------------------------------------|---------------------------------------|---|---|----------------|----------------|----------------|
| 1 | 用例编号 | SG_Sys_Test_case_fun_003 | | | | | | |
| 2 | 测试覆盖的系统功能 | 1.愿望单管理 2.购物车管理 | | | | | | |
| 3 | 用例设计方法 | 场景法 | | | | | | |
| 4 | 前置条件 | 用户以买家身份登录 | | | | | | |
| 5 | 输入 | | | | | | | |
| 6 | 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 | 预期交互 输入、输出6 | 预期交互 输入、输出7 |
| 7 | | 输入：用户在商品详情页点击加入购物车按钮 输出：成功将游戏加入购物车 | 输入：用户在商品详情页点击加入购物车按钮 输出：成功将游戏加入愿望单 | 输入：用户在愿望单界面点击删除出愿望单按钮 输出：成功将游戏删除出愿望单 | 输入：用户在愿望单界面点击删除出购物车按钮 输出：成功将游戏删除出购物车 | 无 | 无 | 无 |
| 8 | 最后预期输出 | | | | | | | |
| 9 | | | | | | | | |

| | A | B | C | D | E | F | G | H |
|---|-----------|-------------------------------|---|------------|------------|------------|------------|------------|
| 1 | 用例编号 | SG_Sys_Test_case_fun_004 | | | | | | |
| 2 | 测试覆盖的系统功能 | 1.买家管理游戏库 2.管理员管理游戏库 | | | | | | |
| 3 | 用例设计方法 | 场景法 | | | | | | |
| 4 | 前置条件 | 用户以管理员或者买家身份登录 | | | | | | |
| 5 | 输入 | | | | | | | |
| 6 | 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 | 预期交互输入、输出6 | 预期交互输入、输出7 |
| 7 | | 输入：买家点击游戏库按钮 输出：买家所拥有的所有游戏 | 输入：管理员点击管理游戏库按钮，并且删除某个用户游戏库当中的违规游戏 输出：删除成功 | 无 | 无 | 无 | 无 | 无 |
| 8 | 最后预期输出 | | | | | | | |
| 9 | | | | | | | | |

| | A | B | C | D | E | F | G | H |
|---|-----------|---|---|--------------------------------------|------------|------------|------------|------------|
| 1 | 用例编号 | SG_Sys_Test_case_fun_005 | | | | | | |
| 2 | 测试覆盖的系统功能 | 1.发布游戏 2.发布游戏更新 3.下架游戏 | | | | | | |
| 3 | 用例设计方法 | 场景法 | | | | | | |
| 4 | 前置条件 | 用户以游戏发行商身份登录 | | | | | | |
| 5 | 输入 | | | | | | | |
| 6 | 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 | 预期交互输入、输出6 | 预期交互输入、输出7 |
| 7 | | 输入：用户输入游戏所需要的名称，简介，图片，定价等信息之后，点击发布游戏按钮 输出：发布游戏成功 | 输入：用户上传游戏需要更新的文件和更新日志，点击发布游戏更新按钮 输出：发布游戏更新成功 | 输入：用户选择所要下架的游戏，点击下架游戏按钮 输出：下架游戏成功 | 无 | 无 | 无 | 无 |
| 8 | | | | | | | | |
| 9 | | | | | | | | |

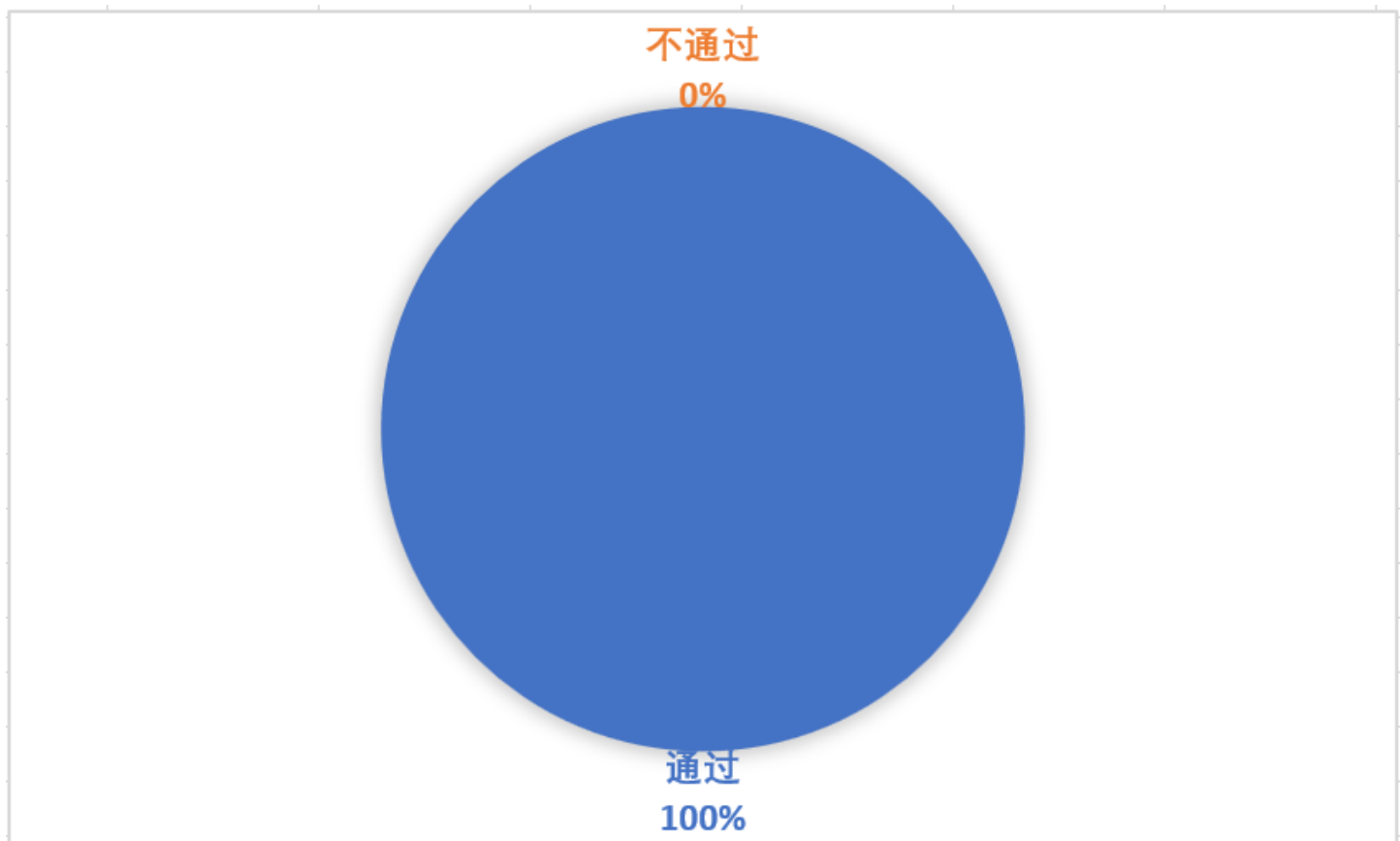
| | A | B | C | D | E | F | G | H |
|---|-----------|--|---|--|------------|------------|------------|------------|
| 1 | 用例编号 | SG_Sys_Test_case_fun_006 | | | | | | |
| 2 | 测试覆盖的系统功能 | 1.发行商信息管理 2.买家信息管理 3.封禁买家 | | | | | | |
| 3 | 用例设计方法 | 场景法 | | | | | | |
| 4 | 前置条件 | 用户已经登录 | | | | | | |
| 5 | 输入 | | | | | | | |
| 6 | 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 | 预期交互输入、输出6 | 预期交互输入、输出7 |
| 7 | | 输入：发行商输入自己要更改的信息，点击修改个人信息按钮 输出：修改个人信息成功 | 输入：买家输入自己要更改的信息，点击修改个人信息按钮 输出：修改个人信息成功 | 输入：管理员在买家面板当中选择某个用户，或者直接输入用户ID，点击封禁买家按钮 输出：封禁买家成功 | 无 | 无 | 无 | 无 |
| 8 | 最后预期输出 | | | | | | | |
| 9 | | | | | | | | |

| | A | B | C | D | E | F | G | H |
|---|-----------|--|--------------------------------------|--|---------------------------------|--------------------------------------|------------|------------|
| 1 | 用例编号 | SG_Sys_Test_case_fun_007 | | | | | | |
| 2 | 测试覆盖的系统功能 | 1.搜索游戏 2.查看游戏 3.发布游戏评论 4.星光游戏推荐 5.删除发行游戏 | | | | | | |
| 3 | 用例设计方法 | 场景法 | | | | | | |
| 4 | 前置条件 | 用户已经以买家或者管理员身份登录 | | | | | | |
| 5 | 输入 | | | | | | | |
| 6 | 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 | 预期交互输入、输出6 | 预期交互输入、输出7 |
| 7 | | 输入：用户输入游戏ID或者游戏名字进行搜索 输出：搜索到的游戏 | 输入：用户点击游戏的图片，进入游戏详情页 输出：游戏对应的详细信息 | 输入：用户输入对该游戏的评论，点击发布游戏评论按钮 输出：发布游戏评论成功 | 输入：管理员输入要推荐的星光游戏 输出：首页展示星光游戏 | 输入：管理员选择某个指定游戏，点击下架游戏按钮 输出：游戏下架成功 | 无 | 无 |
| 8 | 最后预期输出 | | | | | | | |
| 9 | | | | | | | | |

| | A | B | C | D | E | F | G | H |
|---|-----------|---|---|---|----------------|----------------|----------------|----------------|
| 1 | 用例编号 | SG_Sys_Test_case_fun_008 | | | | | | |
| 2 | 测试覆盖的系统功能 | 1.添加好友 2.删除好友 3.与好友聊天 | | | | | | |
| 3 | 用例设计方法 | 场景法 | | | | | | |
| 4 | 前置条件 | 用户以买家方式登录 | | | | | | |
| 5 | 输入 | | | | | | | |
| 6 | 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 | 预期交互 输入、输出6 | 预期交互 输入、输出7 |
| 7 | | 输入：用户输入好友ID，搜索出来对应的好友之后，点击添加好友按钮 输出：添加好友成功 | 输入：用户输入好友ID，搜索出来对应的好友之后，点击删除好友按钮 输出：删除好友成功 | 输入：点击好友聊天按钮，输入和好友聊天的信息 输出：展示和好友聊天的信息 | 无 | 无 | 无 | 无 |
| 8 | 最后预期输出 | | | | | | | |
| 9 | | | | | | | | |

7. 用例执行及报告分析

| 用例编号 | 执行情况 |
|--------------------------|------|
| SG_Sys_Test_case_fun_001 | 通过 |
| SG_Sys_Test_case_fun_002 | 通过 |
| SG_Sys_Test_case_fun_003 | 通过 |
| SG_Sys_Test_case_fun_004 | 通过 |
| SG_Sys_Test_case_fun_005 | 通过 |
| SG_Sys_Test_case_fun_006 | 通过 |
| SG_Sys_Test_case_fun_007 | 通过 |
| SG_Sys_Test_case_fun_008 | 通过 |



8. 利用测试脚本进行测试

我们采用了Selenium包来对我们的系统进行测试，我们测试了11个对应的系统功能，写出了如下所示的测试脚本：

8.1 买家购买游戏测试

Project: TJgames

Tests +

Search tests...

买家购买游戏

卖家上架新游戏

卖家查看已发布游戏

搜索自己的好友

注册模块-买家

添加商品进入愿望单

添加商品进入购物车

登录模块-买家

登录模块-卖家

登录模块-管理员

管理员封禁买家

124.70.163.146:8080/

| Command | Target | Value |
|-------------------|--|---------|
| 1 open | http://124.70.163.146:8080/#/Login | |
| 2 set window size | 1051x798 | |
| 3 click | css= el-form-item:nth-child(1) .el-input__inner | |
| 4 type | css= el-form-item:nth-child(1) .el-input__inner | 10001 |
| 5 click | css= el-input-suffix > .el-input__inner | |
| 6 type | css= el-input-suffix > .el-input__inner | 1234567 |
| 7 click | css= el-button > span | |
| 8 mouse over | css= el-button > span | |
| 9 click | css= el-card__body .el-menu-item:nth-child(4) | |
| 10 click | css= el-table__row:nth-child(2) > .el-table_1_column_1 > .cell | |
| 11 click | css= el-table__row:nth-child(2) .el-checkbox__inner | |

Command

open

#

Target

http://124.70.163.146:8080/#/Login

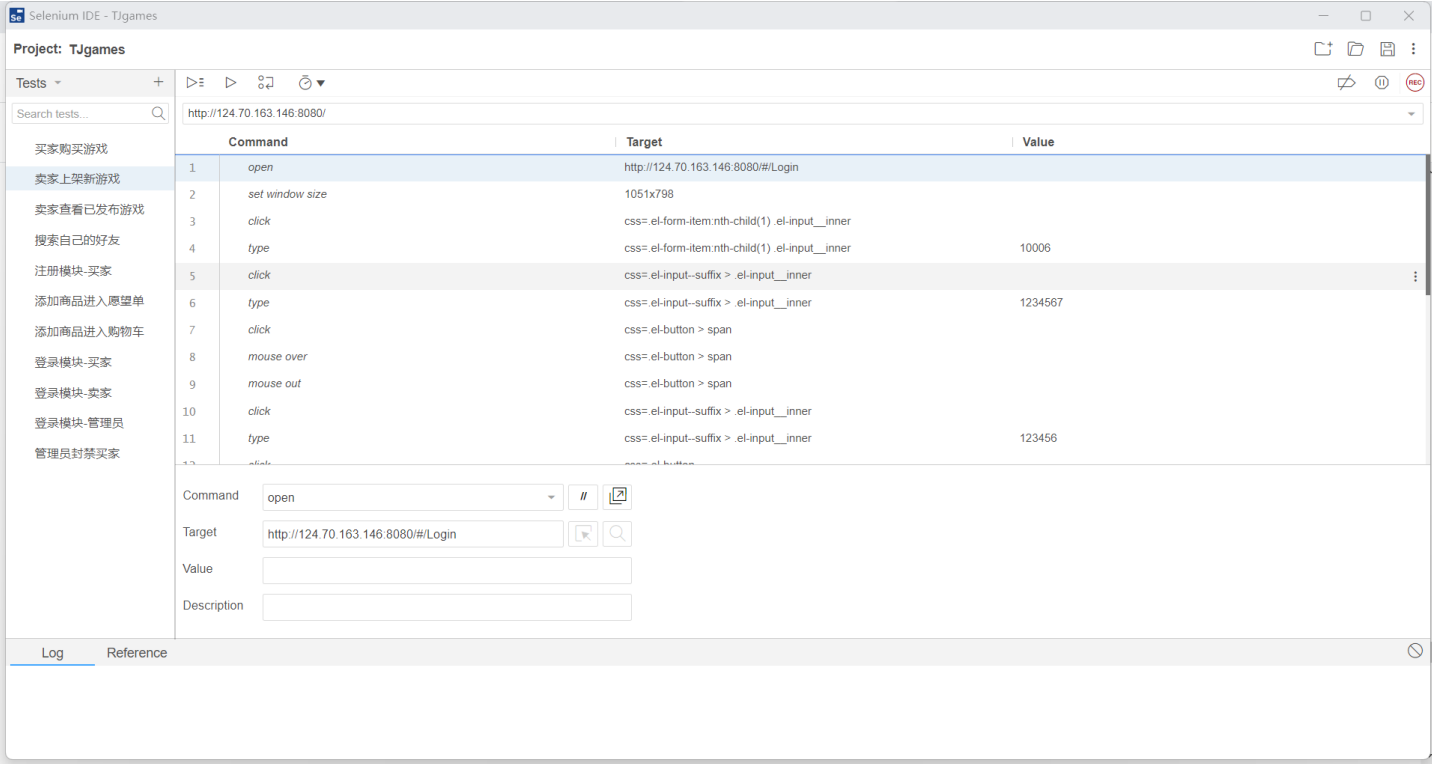
Value

Description

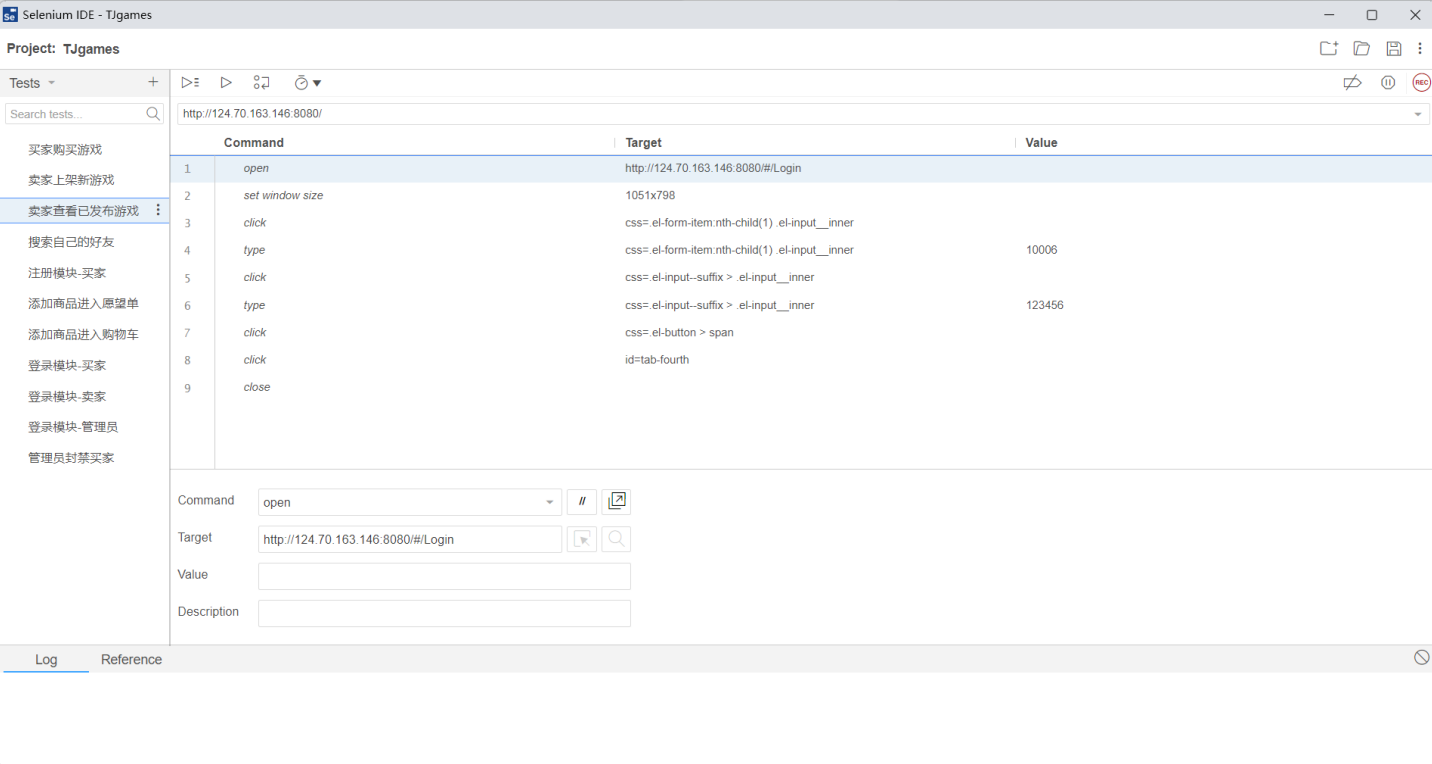
Log

Reference

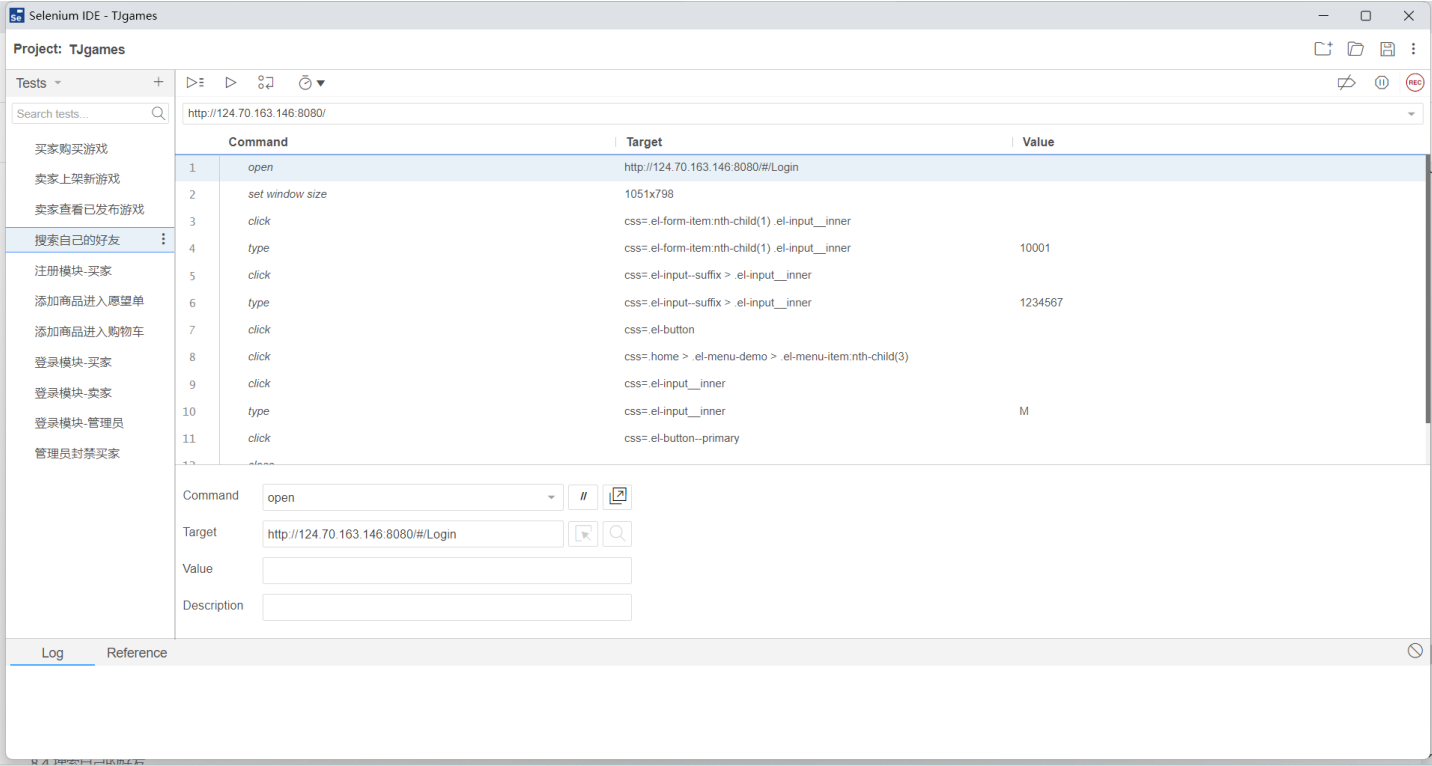
8.2 卖家上架新游戏



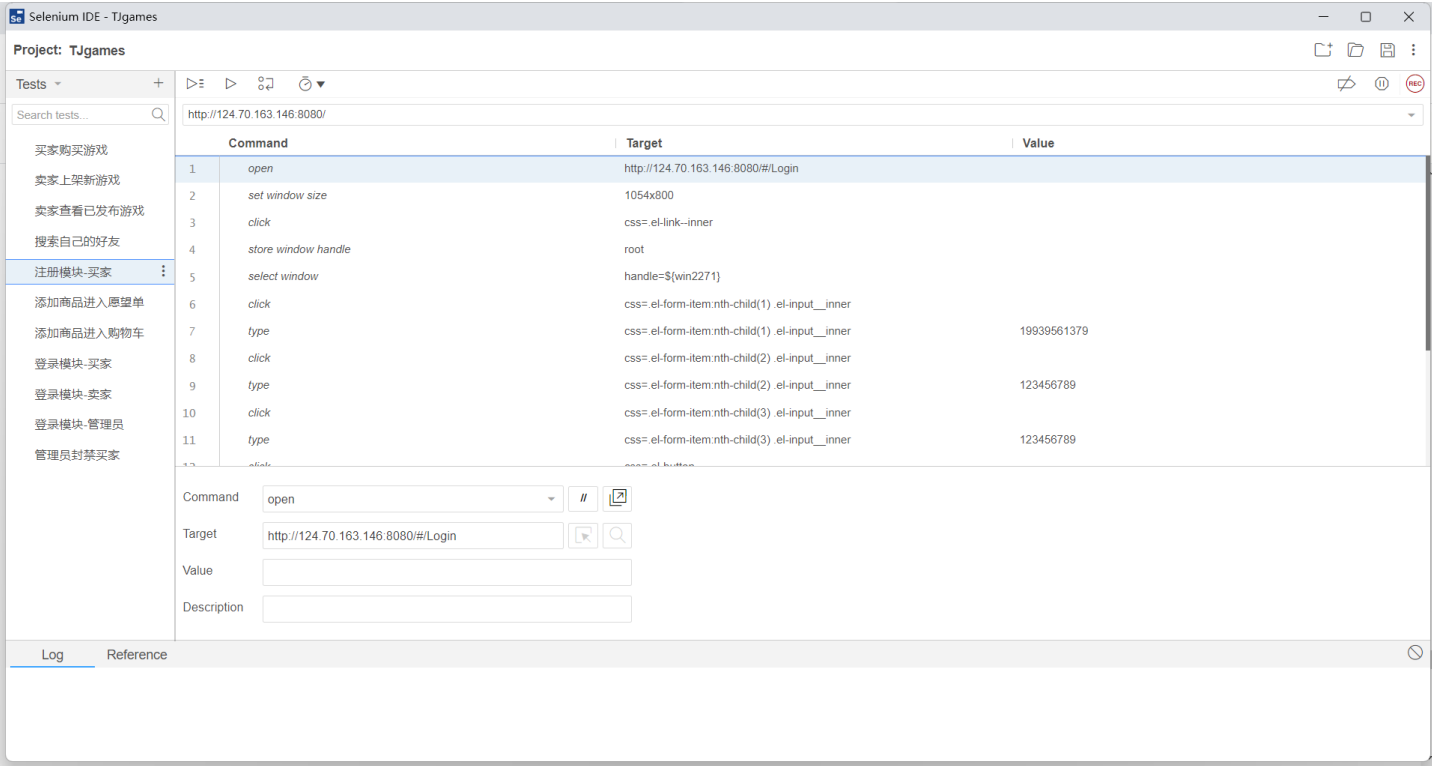
8.3 卖家查看已发布游戏



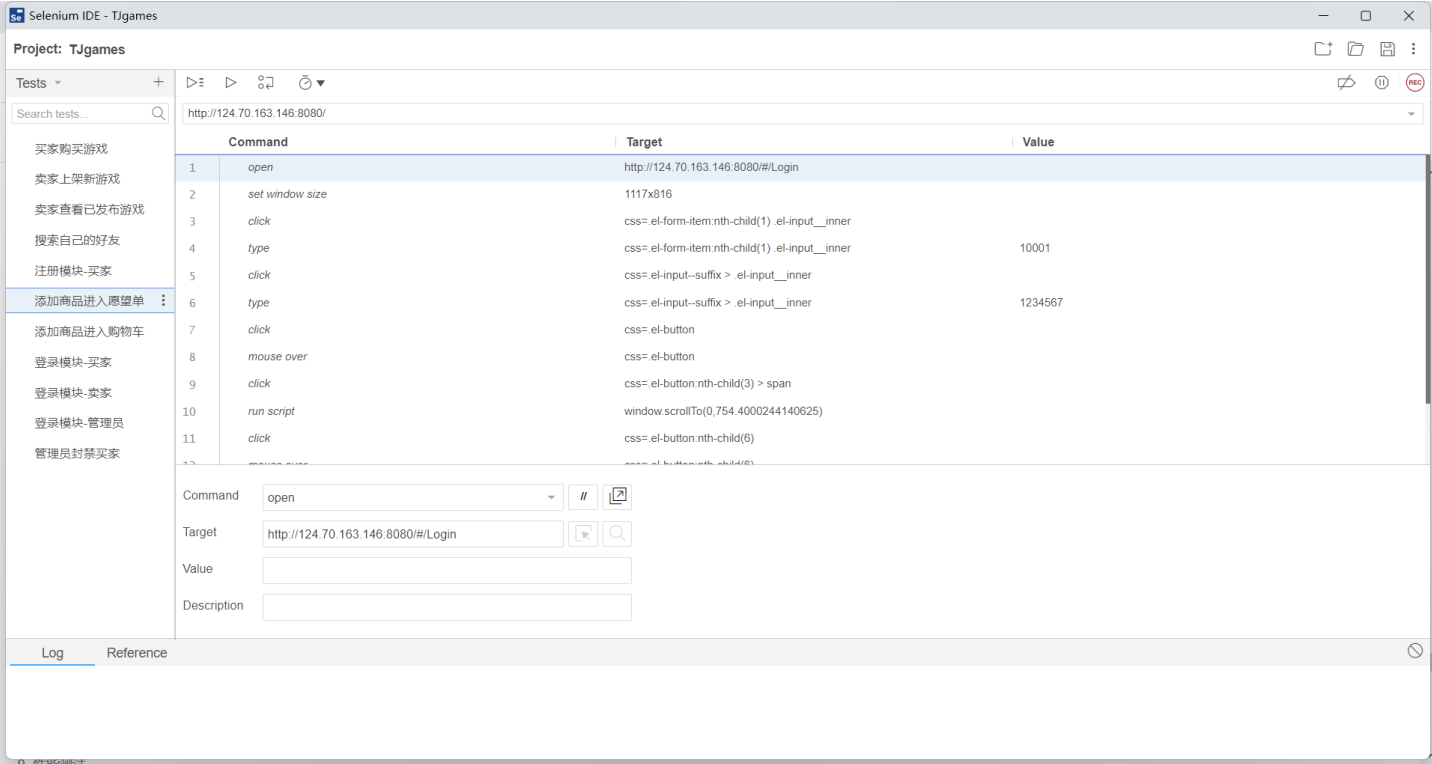
8.4 搜索自己的好友



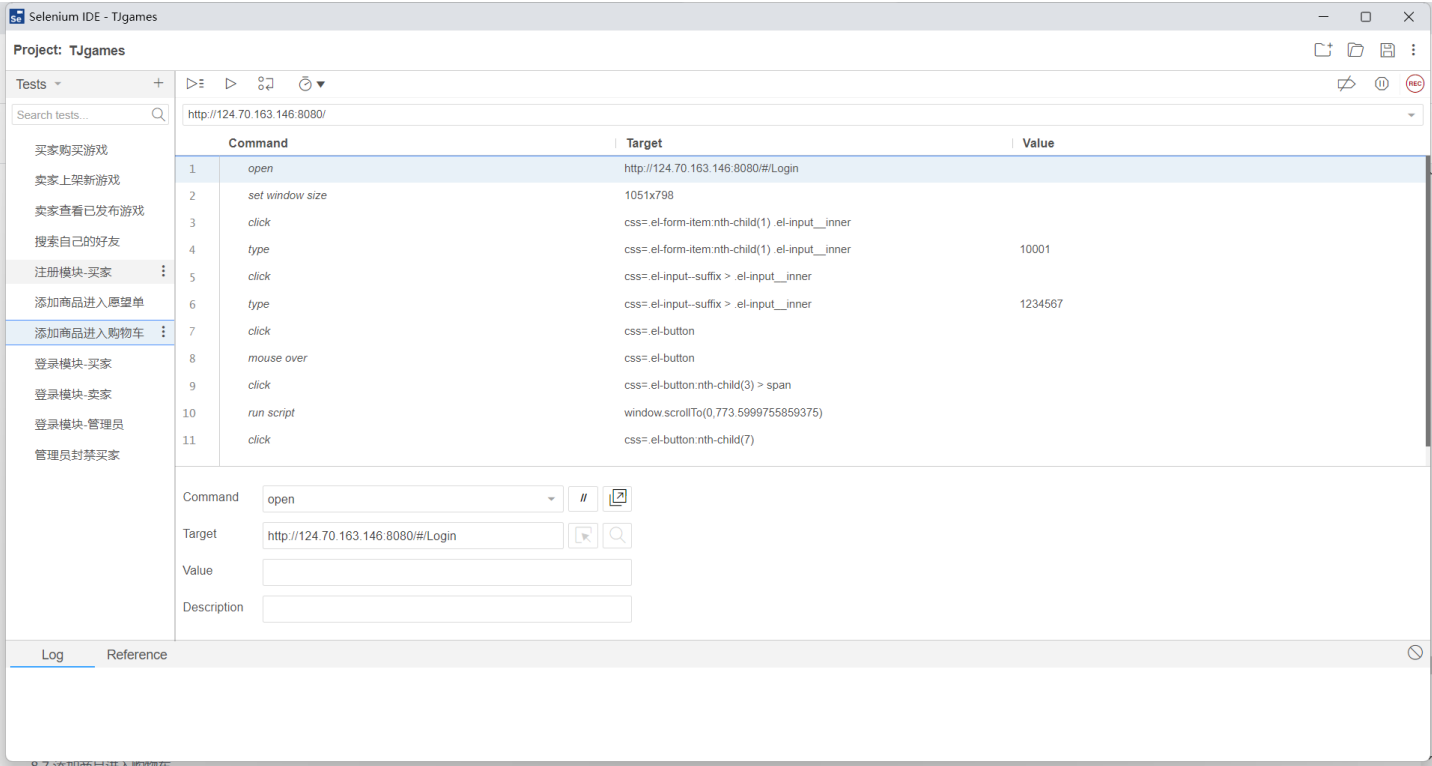
8.5 买家注册



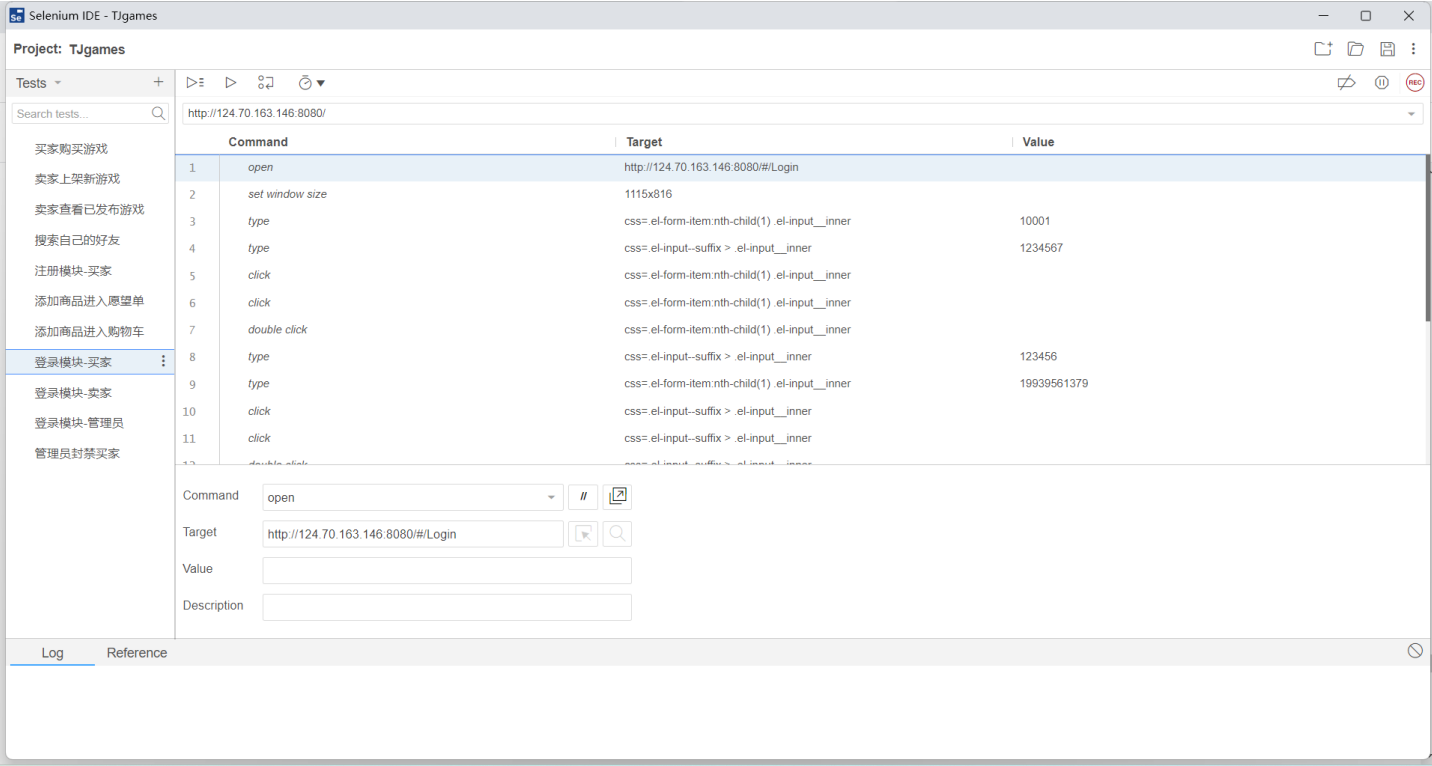
8.6 添加商品进入愿望单



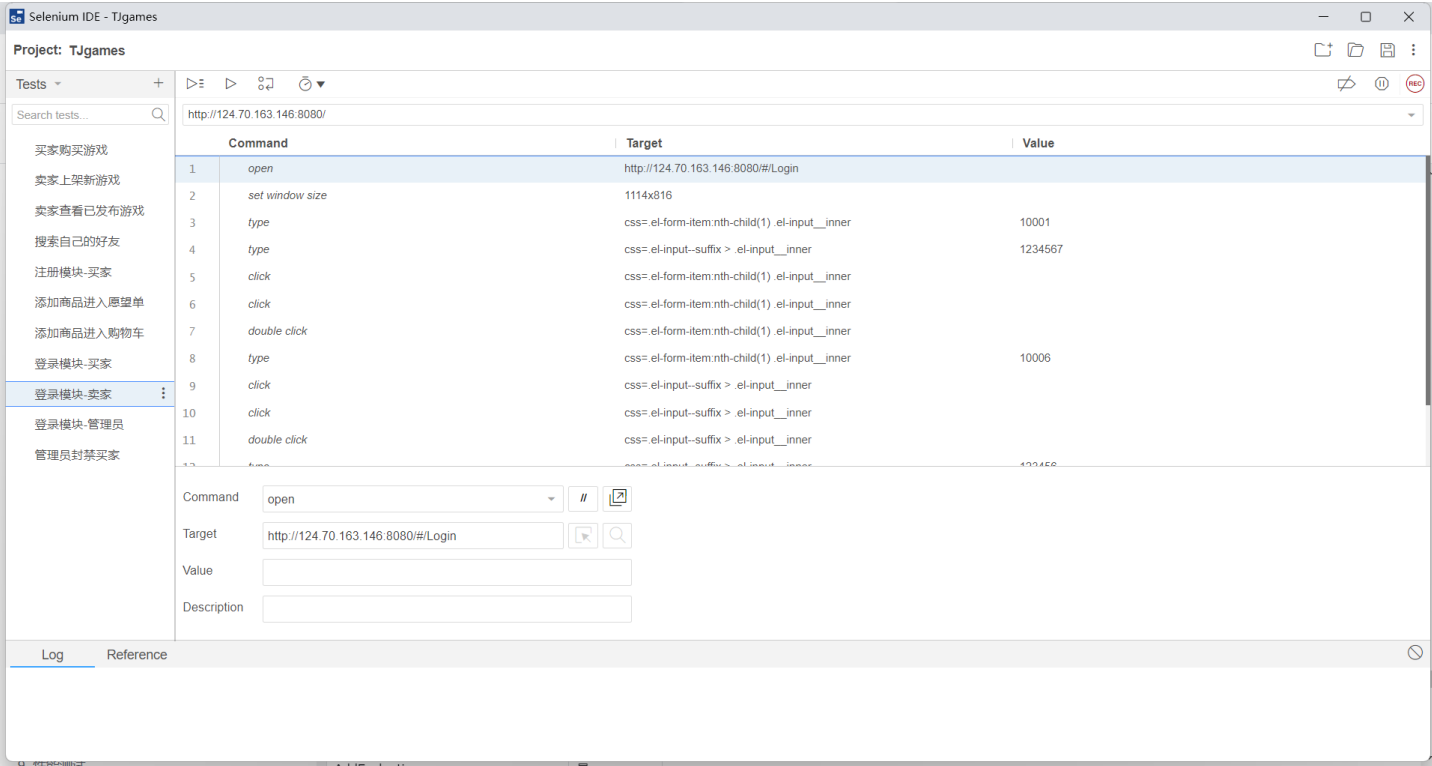
8.7 添加商品进入购物车



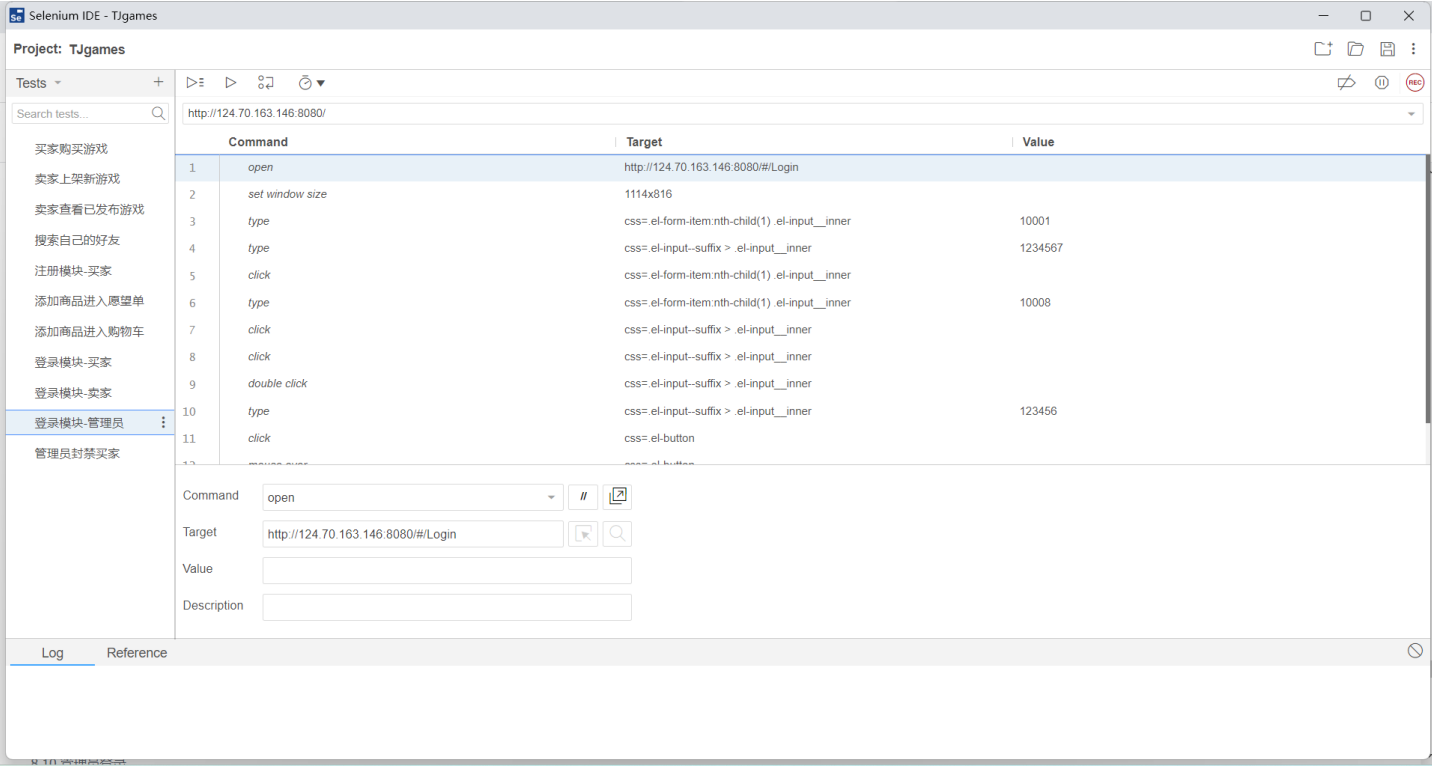
8.8 买家登录



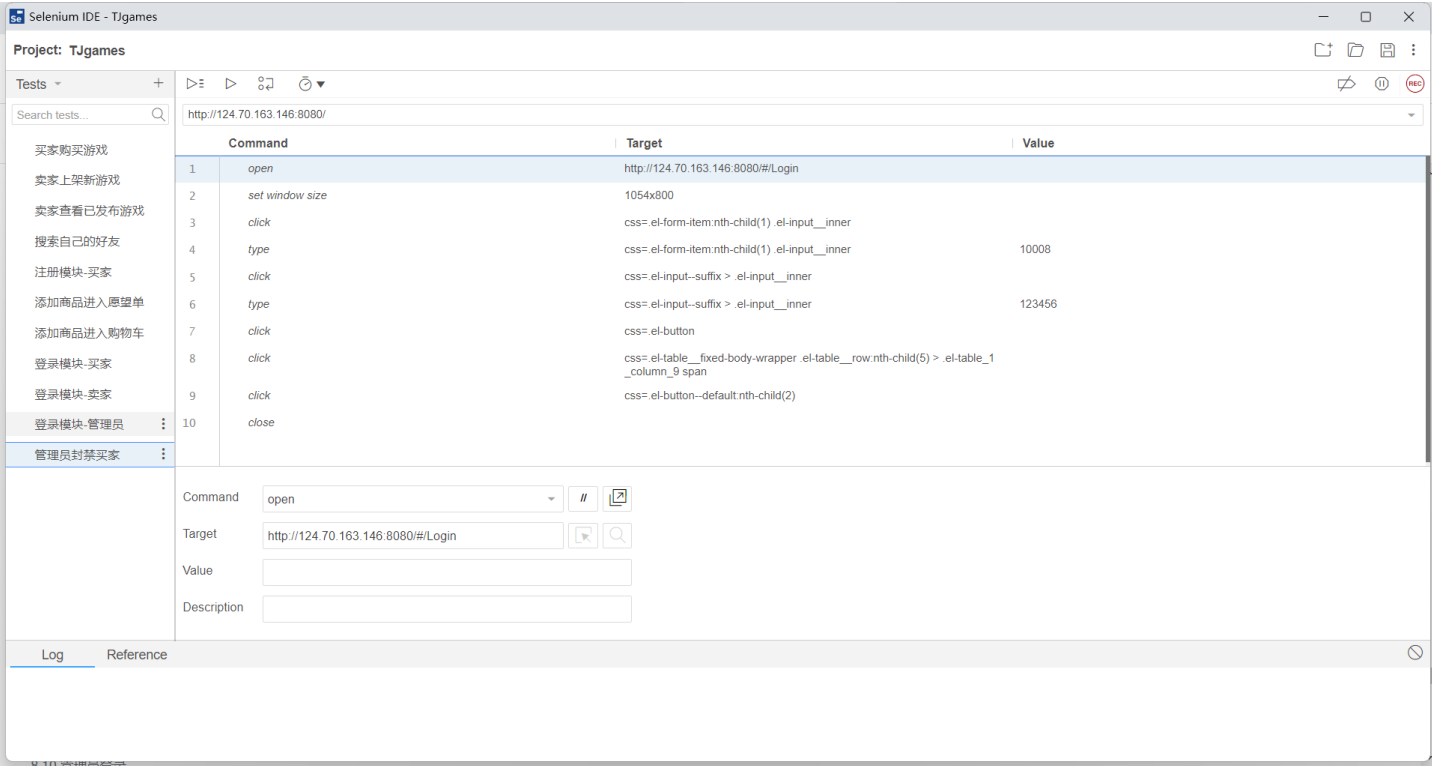
8.9 卖家登录



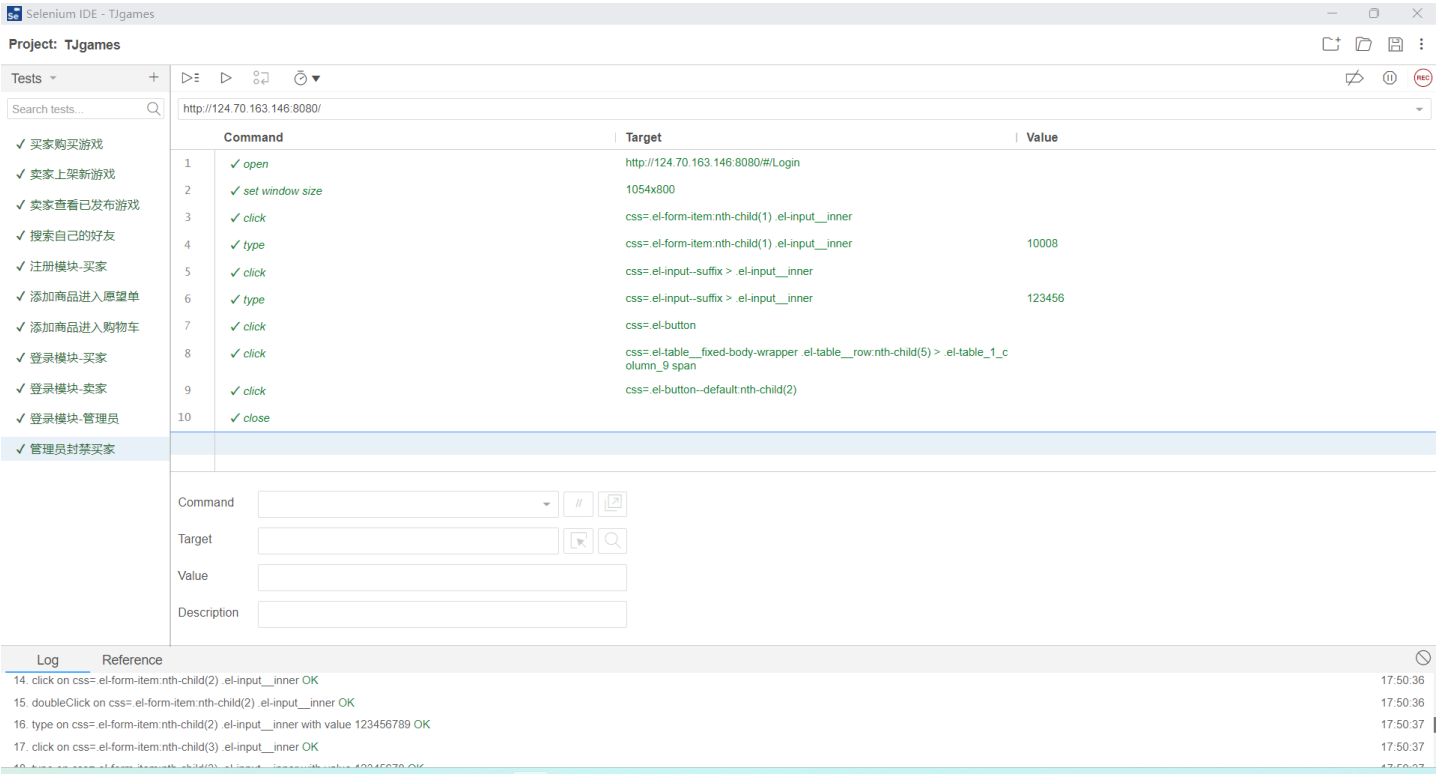
8.10 管理员登录



8.11 管理员封禁买家



最后经过我们的测试，我们系统所有的系统测试均已通过

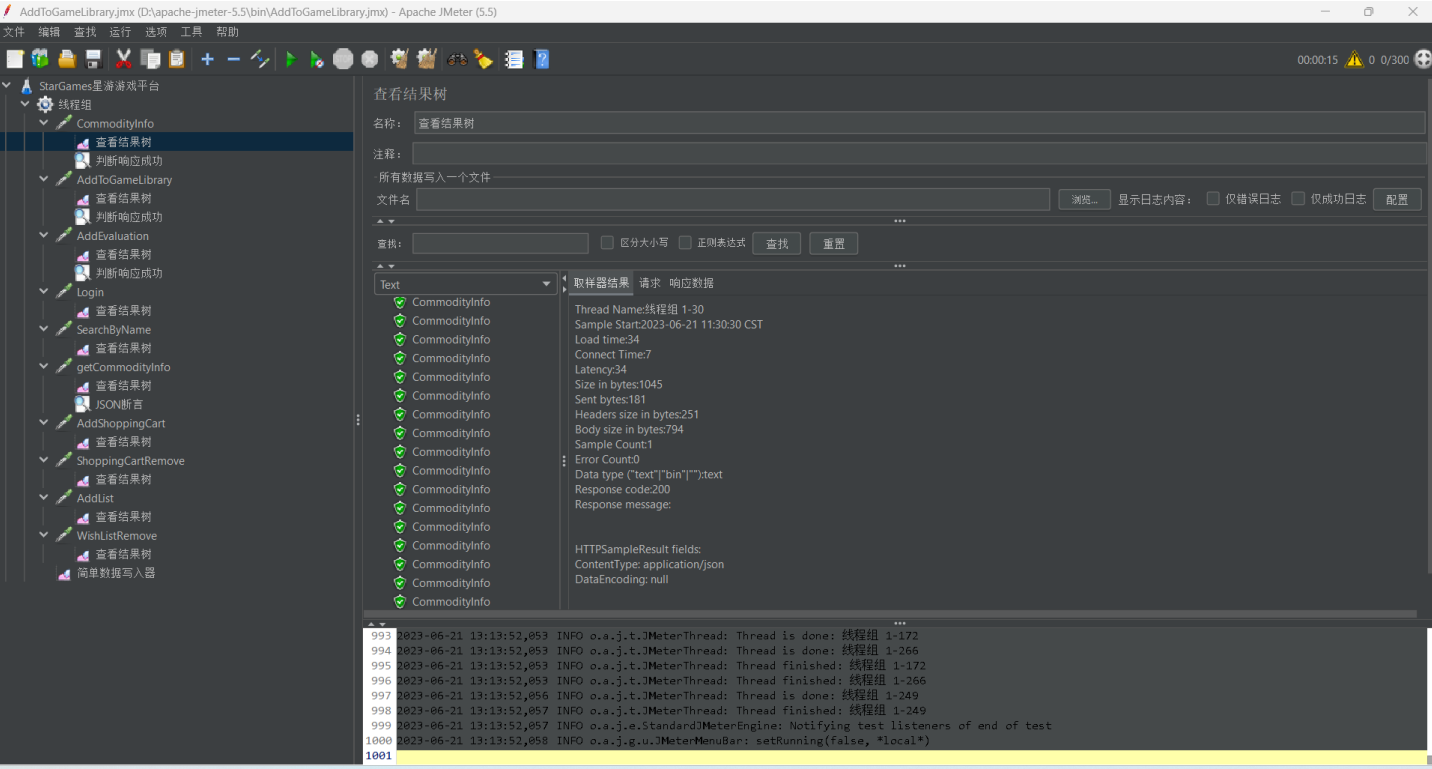


9. 性能测试

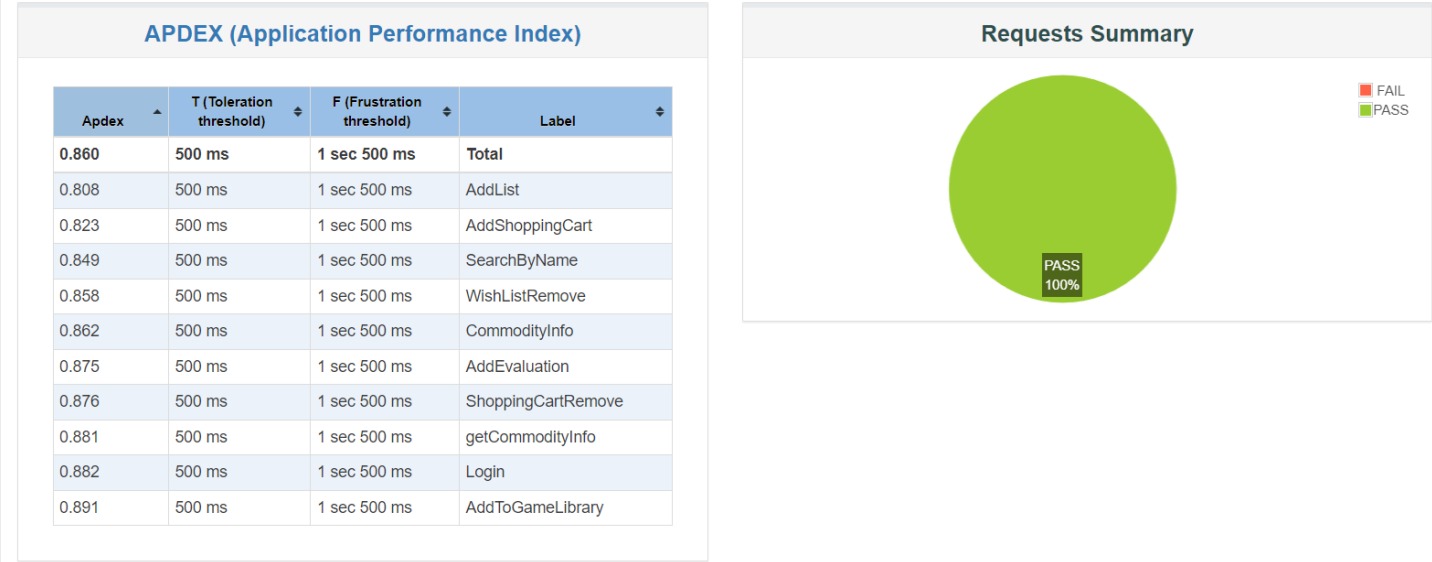
我们使用了JMeter软件来对我们的系统进行性能测试，我们设置了300个线程，每次测试间隔3s，循环3轮，从而来测试我们系统的性能表现，我们测试的接口如下：

| 接口名称 | 是否通过 |
|--------------------|------|
| CommodityInfo | 是 |
| AddToGameLibrary | 是 |
| AddEvaluation | 是 |
| Login | 是 |
| SearchByName | 是 |
| AddShoppingCart | 是 |
| ShoppingCartRemove | 是 |
| AddList | 是 |
| WishListRemove | 是 |

测试工具的截图如下：



经过测试，我们生成了如下图所示的图表：



Statistics

| Requests | Executions | | | Response Times (ms) | | | | | | | Throughput | Network (KB/sec) | |
|--------------------|------------|------|---------|---------------------|-----|------|--------|----------|----------|----------|----------------|------------------|--------|
| Label | #Samples | FAIL | Error % | Average | Min | Max | Median | 90th pct | 95th pct | 99th pct | Transactions/s | Received | Sent |
| Total | 9000 | 0 | 0.00% | 386.91 | 16 | 4701 | 221.00 | 957.00 | 1302.95 | 2140.99 | 573.58 | 262.49 | 115.05 |
| AddEvaluation | 900 | 0 | 0.00% | 358.56 | 23 | 3405 | 190.00 | 826.70 | 1258.85 | 2051.50 | 58.40 | 17.91 | 16.60 |
| AddList | 900 | 0 | 0.00% | 484.67 | 16 | 3426 | 287.00 | 1174.20 | 1595.45 | 2475.74 | 58.39 | 17.88 | 10.66 |
| AddShoppingCart | 900 | 0 | 0.00% | 451.28 | 17 | 3921 | 262.00 | 1112.50 | 1518.50 | 2366.89 | 58.24 | 17.78 | 11.09 |
| AddToGameLibrary | 900 | 0 | 0.00% | 326.43 | 18 | 4701 | 177.00 | 801.20 | 1111.30 | 1822.12 | 58.65 | 17.81 | 11.00 |
| CommodityInfo | 900 | 0 | 0.00% | 390.83 | 25 | 3074 | 211.00 | 1010.70 | 1359.45 | 2170.96 | 58.97 | 60.18 | 10.42 |
| getCommodityInfo | 900 | 0 | 0.00% | 363.25 | 25 | 3574 | 206.00 | 909.30 | 1178.85 | 2013.88 | 58.21 | 59.41 | 10.29 |
| Login | 900 | 0 | 0.00% | 342.33 | 22 | 2860 | 192.50 | 805.90 | 1145.60 | 2031.44 | 58.33 | 14.98 | 14.75 |
| SearchByName | 900 | 0 | 0.00% | 406.93 | 26 | 3351 | 244.50 | 960.20 | 1311.40 | 2348.77 | 58.23 | 31.33 | 10.12 |
| ShoppingCartRemove | 900 | 0 | 0.00% | 362.12 | 18 | 3473 | 216.50 | 820.80 | 1190.80 | 1769.97 | 58.34 | 15.13 | 11.34 |
| WishListRemove | 900 | 0 | 0.00% | 382.68 | 17 | 2596 | 230.50 | 946.60 | 1216.95 | 1996.81 | 58.43 | 15.15 | 10.90 |

Errors

| Type of error | Number of errors | % in errors | % in all samples |
|---------------|------------------|-------------|------------------|
|---------------|------------------|-------------|------------------|

Top 5 Errors by sampler

| Sample | #Samples | #Errors | Error | #Errors | Error | #Errors | Error | #Errors | Error | #Errors | Error | #Errors |
|--------|----------|---------|-------|---------|-------|---------|-------|---------|-------|---------|-------|---------|
| Total | 9000 | 0 | | | | | | | | | | |

可以看出，我们小组编写的软件的性能较好，但是仍然有提升的空间，response time有时候会出现很大的情况，该种情况便是我们之后的优化重点

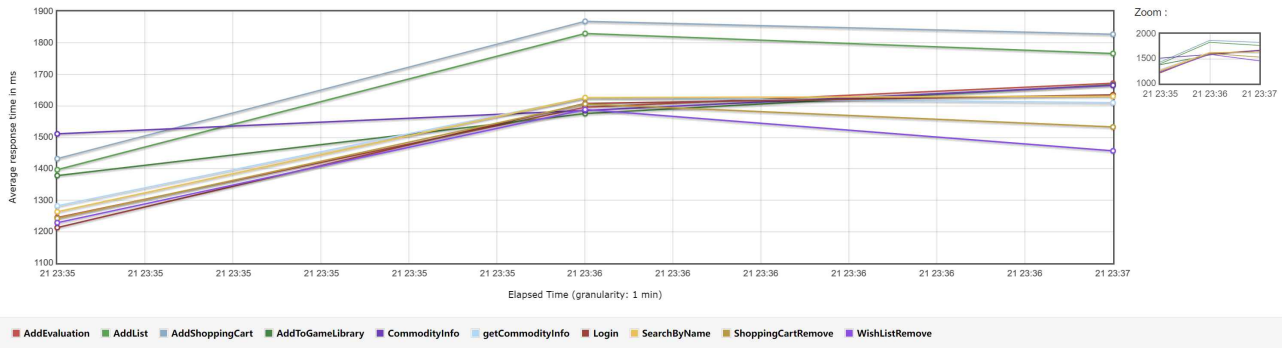
然后我们小组进行了压力测试，通过发送60000个请求，来测试我们系统的压力承受能力，压力测试结果如下图：

| Errors | | | | | | | | | | | | |
|---|------------------|-------------|------------------|--|--|--|--|--|--|--|--|--|
| Type of error | Number of errors | % in errors | % in all samples | | | | | | | | | |
| Non HTTP response code: org.apache.http.conn.HttpHostConnectException/Non HTTP response message: Connect to 124.70.163.146:8080 [/124.70.163.146] failed: Connection timed out: connect | 3 | 60.00% | 0.01% | | | | | | | | | |
| 500 | 2 | 40.00% | 0.00% | | | | | | | | | |

| Top 5 Errors by sampler | | | | | | | | | | | | |
|-------------------------|----------|---------|---|---------|-------|---------|-------|---------|-------|---------|-------|---------|
| Sample | #Samples | #Errors | Error | #Errors | Error | #Errors | Error | #Errors | Error | #Errors | Error | #Errors |
| Total | 60000 | 5 | Non HTTP response code: org.apache.http.conn.HttpHostConnectException/Non HTTP response message: Connect to 124.70.163.146:8080 [/124.70.163.146] failed: Connection timed out: connect | 3 | 500 | 2 | | | | | | |
| AddEvaluation | 6000 | 2 | Non HTTP response code: org.apache.http.conn.HttpHostConnectException/Non HTTP response message: Connect to 124.70.163.146:8080 [/124.70.163.146] failed: Connection timed out: connect | 2 | | | | | | | | |
| Login | 6000 | 1 | Non HTTP response code: org.apache.http.conn.HttpHostConnectException/Non HTTP response message: Connect to 124.70.163.146:8080 [/124.70.163.146] failed: Connection timed out: connect | 1 | | | | | | | | |
| WishListRemove | 6000 | 2 | 500 | 2 | | | | | | | | |

我们接口的响应时间随着测试时间变化如下图：

Response Times Over Time



Response Time Percentiles Over Time (successful responses)

