

**《软件测试》**

**期末课程设计**

**怡运动——运动爱好者的运动与社交平台**

**系统测试文档**

小组成员

2253407 柳阳

2252721 韩坤甫

2254302 商一诚

2254305 夏济舟

目录

[1. 系统需求 1](#_Toc75989746)

[1.1 引言 1](#_Toc1957401666)

[1.1.1 目的 1](#_Toc1243105339)

[1.1.2 背景 1](#_Toc1130973966)

[1.1.3 参考资料 1](#_Toc1899516885)

[1.1.4 术语 1](#_Toc425747348)

[1.2 项目概述 2](#_Toc1415673246)

[1.2.1 开发软件的一般描述 2](#_Toc653852169)

[1.2.2 开发软件的功能描述 2](#_Toc230665077)

[1.2.3 实现语言 3](#_Toc217762337)

[1.2.4 用户特点 3](#_Toc60878855)

[1.2.5 一般约束 4](#_Toc815341967)

[1.3 需求说明 4](#_Toc71264789)

[1.3.1 基本描述 4](#_Toc939252202)

[1.3.2 功能需求 5](#_Toc2047589130)

[1.3.3 性能与其他需求 6](#_Toc1929165118)

[1.3.4 对输入输出的规定 7](#_Toc1440226306)

[2. 采用的测试方法 8](#_Toc1448003079)

[3. 测试环境 9](#_Toc959671501)

[硬件需求： 9](#_Toc444298884)

[4. 测试对象 9](#_Toc979757651)

[5. 测试分析与用例设计 9](#_Toc928069574)

[5.1 测试数据准备 9](#_Toc2088235604)

[5.2 基于系统级功能的测试分析与用例设计 10](#_Toc43694617)

[5.3 基于系统业务流测试分析与用例设计 26](#_Toc215899483)

[6. 测试结果 30](#_Toc125757830)

[6.1系统级功能结果 30](#_Toc261940132)

[6.2 系统业务流测试 32](#_Toc1930024732)

[6.3系统测试缺陷跟踪管理 33](#_Toc220343456)

[6.3.1. 系统测试用例与缺陷标注表 33](#_Toc96045053)

[6.3.2. 缺陷跟踪表 33](#_Toc1961335597)

[6.3.3. 缺陷跟踪过程说明 34](#_Toc296333202)

[7. 性能测试 35](#_Toc2053446719)

[7.1 测试目的 35](#_Toc1056957289)

[7.2 测试工具 35](#_Toc1427307169)

[7.3 测试需求分析 35](#_Toc1805479956)

[7.4 测试方案设计 36](#_Toc1482704637)

[7.5 测试过程 36](#_Toc695496767)

[7.6 测试结果 39](#_Toc311848478)

[7.6.1 登录性能 39](#_Toc1713369714)

[7.7 测试结论 39](#_Toc381228033)

## 1. 系统需求

### 1.1 引言

#### 1.1.1 目的

本系统测试文档旨在明确“怡运动”平台的功能、性能、接口等方面的测试要求、策略、方法和结果，确保测试活动有据可依，系统质量达到预期目标。它是开发人员、测试人员和质量管理人员共同遵循的测试基准，也为后续系统维护提供测试依据。

#### 1.1.2 背景

“怡运动”是一个面向体育运动爱好者的综合性平台，旨在解决运动爱好者面临的组团不便和场地预约困难等痛点。平台整合了线上社交、运动组团和场地预约功能，为个人用户和运动俱乐部/社团提供便捷服务。随着平台开发接近尾声，需要进行全面、系统的测试以验证其是否符合需求规格说明书中定义的功能和非功能需求，确保上线后的稳定性和用户体验。

#### 1.1.3 参考资料

《怡运动需求分析规约》

《怡运动概要设计规约》

Roger S.Pressman&Bruce R.Maxim.软件工程---实践者的研究方法（第八版）.郑仁杰译.北京:机械工业出版社.2016.12

GB/T 25000.51-2016 系统与软件工程 系统与软件质量要求和评价(SQuaRE) 第51部分：就绪可用软件产品(RUSP)的质量要求和测试细则

GB/T 15532-2008 计算机软件测试规范

#### 1.1.4 术语

|  |  |
| --- | --- |
| **术语** | **定义** |
| 怡运动平台 | 本项目所涉及的网站/应用程序，用于运动社交、组团活动、场地预约等。 |
| 普通用户 | 注册并使用平台进行运动社交、预约场地、参与或发起活动的个人用户。 |
| 运动俱乐部/社团 | 在平台上注册的组织用户，用于管理成员、组织团体活动、预约场地等。 |
| 运动活动 | 用户（个人或团体）在平台上发起的，邀请他人参与的特定时间、地点的运动项目（如篮球局、羽毛球局）。 |
| 场地预约 | 用户通过平台预订特定运动场馆在特定时间段的场地使用权。 |
| 拼场预约 | 用户预约场地时，选择公开招募其他用户共同使用该场地并分担费用的预约方式。 |

### 1.2 项目概述

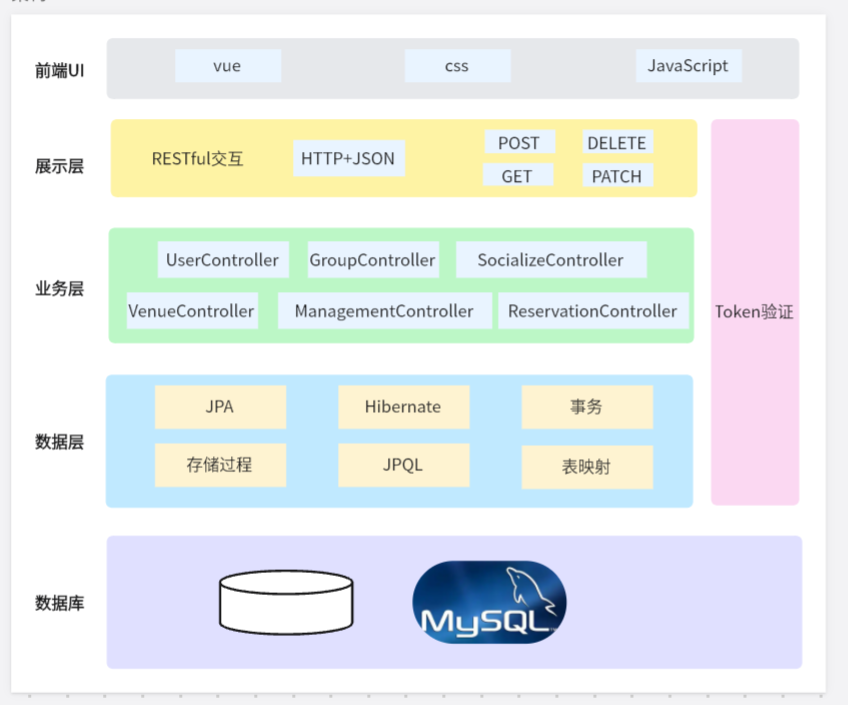
#### 1.2.1 开发软件的一般描述

“怡运动”是一个基于Web和移动端的平台，核心功能包括：用户管理（注册/登录/信息维护）、运动活动管理（创建/浏览/参与/管理）、场地信息浏览与预约（个人/拼场/团体）、社交功能（添加好友/群聊）、团体管理（创建/加入/管理成员）。平台旨在为运动爱好者提供一个集社交、组局、场地预约于一体的便捷环境。

#### 1.2.2 开发软件的功能描述

怡运动系统主要提供的功能有：用户注册与登录，个人信息管理，创建/浏览/搜索/参与运动活动，浏览/搜索运动场地，进行场地预约（个人/拼场/团体），管理个人预约，场地评价，添加/管理好友，发起/参与群聊，创建/加入/管理运动团体。

分为如下子系统：用户管理子系统，运动活动管理子系统，场地预约管理子系统，社交管理子系统，团体管理子系统。



#### 1.2.3 实现语言

后端：Java (Spring Boot框架)

前端：JavaScript (Vue3框架)

数据库：MySQL

#### 1.2.4 用户特点

普通用户：年龄多在18-40岁，具备智能手机操作能力。关注健康生活，乐于社交和寻找运动伙伴。期望操作简单、预约流程顺畅、信息准确。使用频度：中高频（每周多次访问，用于组团、预约、社交）。

运动俱乐部/社团：有定期组织活动和场地预约的需求。需要高效的团体管理和预约工具。具备基本的互联网和办公软件使用能力。使用频度：高频（频繁访问，用于组织活动和管理预订）。

维护人员：具备软件工程背景，熟悉系统架构。负责系统监控、故障排查、性能优化和版本更新。系统界面设计需简洁友好，操作逻辑清晰，易于学习使用。

#### 1.2.5 一般约束

本软件的主要约束是时间约束和法律约束。

### 1.3 需求说明

#### 1.3.1 基本描述

“怡运动”平台的核心业务是连接运动爱好者，提供便捷的运动组局和场地预约服务。用户流程主要包括：注册/登录 -> 浏览/创建活动或场地 -> 预约/参与 -> 社交互动 -> 活动结束/场地使用 -> 评价。运动俱乐部/社团可创建团体，管理成员，进行团体预约。

##### 1.3.1.1 用户信息

普通用户信息： 用户ID、昵称、头像URL、密码（加密存储）、手机号、性别、运动偏好标签。

团体信息： 团体ID、名称、Logo URL、简介、创建者ID、创建时间、状态（正常/解散）。

团体成员信息： 成员ID（关联用户ID）、团体ID、角色（创建者/管理员/普通成员）、加入时间、状态（正常/已退出）。

管理员信息： 管理员ID、用户名、密码（加密存储）、联系方式、权限级别。

##### 1.3.1.2 运动活动信息

活动信息： 活动ID、活动标题、活动描述、运动类型、活动时间、活动地点（场地ID或自定义地址）、发起人ID、最大参与人数、当前参与人数、创建时间、状态（招募中/进行中/已结束/已取消）。

活动参与者信息： 参与记录ID、活动ID、参与者ID（用户ID）、参与时间、状态（已报名/已参加/已取消）。

##### 1.3.1.3 场地预约信息

场地信息： 场地ID（由场地管理方系统提供）、名称、类型（篮球场、羽毛球场等）、地址、图片URL、开放时间、价格、联系方式、简介、平均评分。

预约信息： 预约ID、场地ID、预约用户ID（个人）或团体ID、预约类型（个人/拼场/团体）、预约时间段（开始时间-结束时间）、预约人数、费用、支付状态（待支付/已支付/已取消）、预约状态（待确认/已确认/已使用/已过期/已取消）、创建时间。 （拼场预约有额外字段：拼场状态（招募中/已成团/失败）、拼场成员列表）。

场地评价信息： 评价ID、场地ID、评价用户ID、评分（1-5星）、评价内容、评价时间。

#### 1.3.2 功能需求

1. 登录注册子系统：

* 用户注册（手机号/验证码、邮箱/密码）。
* 用户登录（手机号/验证码、邮箱/密码）。
* 找回/重置密码。
* 查看/编辑个人资料（昵称、头像、性别、运动偏好等）。

1. 运动活动管理子系统：

* 浏览/搜索运动活动（按类型、时间、地点、关键词等）。
* 查看活动详情。
* 创建新的运动活动（设置标题、类型、时间、地点、人数等）。
* 报名参加活动。
* 取消活动报名。
* 管理自己发起的活动（查看报名者、修改活动信息、取消活动）。

1. 场地预约管理子系统：

* 浏览/搜索运动场地（按类型、地点、价格、评分等）。
* 查看场地详情（包含图片、价格、开放时间、评价）。
* 进行场地预约：
* 个人预约：选择场地、时间段、人数 -> 预约成功。
* 拼场预约：选择场地、时间段、招募人数 -> 等待成团/成团失败处理。
* 团体预约（需有团体权限）：选择场地、时间段、人数（代表团体） -> 预约成功。
* 查看/管理个人预约记录（包括个人、拼场参与、团体预约）。
* 取消预约（需符合取消规则）。
* 对已使用的场地进行评价和打分。
* 查看场地评价。

1. 社交管理子系统：

* 查找用户（按昵称、ID）。
* 发送/接收/同意/拒绝好友申请。
* 查看好友列表。
* 删除好友。
* 发起群聊（选择好友）。
* 参与群聊（发送/接收消息）。
* 退出群聊。

1. 团体管理子系统：

* 创建新的运动团体（设置名称、Logo、简介）。
* 浏览/搜索运动团体。
* 申请加入团体。
* 审批团体加入申请（团体管理员）。
* 查看/管理团体成员（角色管理、移除成员）。
* 查看/管理团体信息（名称、Logo、简介 - 仅管理员/创建者）。
* 解散团体（仅创建者）。
* 进行团体场地预约（需有权限）。

#### 1.3.3 性能与其他需求

**精度**

* 时间：精确到分钟。
* 金额：保留小数点后两位（人民币元）。
* 评分：整数（1-5）。

**时间特性要求**

* 主要操作响应时间（如浏览列表、查看详情、预约提交）：在正常网络下，95%的请求响应时间 ≤ 2秒，高峰时段 ≤ 3秒。
* 关键事务操作（如支付、创建活动、团体管理）：在正常网络下，响应时间 ≤ 3秒。
* 数据更新与查询应保证实时性，避免用户看到过期信息。

**可靠性需求**

* 容错性： 对用户输入进行有效性校验，提供清晰错误提示，防止无效数据导致系统崩溃。
* 健壮性： 能处理网络中断、服务暂时不可用等异常情况，在网络恢复后能继续未完成操作（如支付重试）。
* 成熟性： 故障率低，因故障停止运行的时间可控。系统需记录详细日志，便于故障定位。
* 可用性： 提供24/7服务，计划内维护时间需提前公告。

**易用性需求**

* 易学习性： 界面直观，新用户能快速上手。提供必要的操作引导。
* 易操作性： 操作流程顺畅，步骤清晰，减少冗余操作。符合用户习惯。
* 用户界面美观： 设计简洁大方，视觉舒适，布局合理（适配PC主流分辨率）。
* 用户错误防御： 对可能造成损失的操作（如删除、取消预约）提供二次确认。对输入错误提供即时提示。

**可维护性需求**

* 模块性： 系统划分为清晰的子系统，模块间耦合度低，便于独立测试和修改。
* 易分析性： 提供详细的系统运行日志和错误日志，便于诊断问题原因。
* 可移植性： 基于Web，确保在不同操作系统（Windows, macOS）的主流浏览器上兼容性良好。
* 文档： 提供完善的开发文档、API文档和部署维护文档。

#### 1.3.4 对输入输出的规定

| **数据类型** | **格式** | **数值范围/约束** | **精度/备注** |
| --- | --- | --- | --- |
| 整数 | int32 | -2147483647 ~ 2147483647 | 精确到1 (如人数, ID) |
| 字符串 | String (UTF-8) | 长度 1 ~ 255 (用户名、昵称、标题等) |  |
|  |  | 长度 1 ~ 1000 (描述、评价内容等) |  |
| 长文本 | Text | 长度不限 | 用于详细描述 |
| 时间 | Timestamp / DateTime | 标准日期时间格式 | 精确到分钟 |
| 日期 | Date | 标准日期格式 |  |
| 时间间隔 | Duration | 起始时间 + 结束时间 或 起始时间 + 时长(分钟) | 精确到分钟 |
| 金额 | Decimal | >= 0.00 | 精确到分 (小数点后两位) |
| 评分 | Integer | 1 ~ 5 |  |
| 布尔值 | Boolean | true / false |  |
| 图片 | URL (存储路径) | 支持格式: JPG, PNG, WebP; 大小 < 2MB | 前端上传时压缩 |
| 手机号 | String | 符合中国大陆手机号格式 (1[3-9][0-9]{9}) |  |

## 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（主要是黑盒测试技术）来设计测试用例：

• 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具有交叉重复的可能，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：1台 (Windows 10/11 或 macOS)

• 应用服务器：1台 (阿里云ECS, 配置参考需求规约4.1节: >= 8GB RAM, >= 256GB SSD, Intel Core i5/Ryzen 5+)

• 数据库服务器：1台 (阿里云RDS for MySQL 或与App Server同机部署)

软件需求：

• 操作系统 (测试机)：Windows 10/11 或 macOS

• 浏览器 (测试)：最新版 Chrome, Firefox, Edge (Safari for macOS)

• 被测应用：怡运动 v1.0.0 (Web前端 + 后端API)

• 数据库：MySQL 8.0+

• 测试工具：Cypress (功能/业务流测试), JMeter (性能测试), Postman (接口测试), Git (版本控制)

网络需求：

• 测试机：稳定宽带连接，推荐带宽 >= 10Mbps。

• 服务器：部署在阿里云，公网带宽按需配置。

## 4. 测试对象

怡运动系统 Web 应用。应用软件版本：v1.0.0。

## 5. 测试分析与用例设计

### 5.1 测试数据准备

| **账号** | **身份** | **密码** | **所属团体 (如适用)** | **备注** |
| --- | --- | --- | --- | --- |
| 1234 | 普通用户 | 1234 | - | 用于个人操作测试 |
| 13900139000 | 普通用户 | 456 | - | 用于社交、参与活动/拼场测试 |
| 15000150000 | 普通用户 (团体创建者) | 789 | 飞驰篮球队 (ID:1) | 用于团体相关操作测试 |
| 15100151000 | 普通用户 (团体成员) | 000 | 飞驰篮球队 (ID:1) | 用于团体成员操作测试 |
| admin | 系统管理员 | admin | - | 用于后台管理功能测试 |

场地数据： 预先在数据库插入若干条不同运动类型、不同区域的场地数据。

团体数据： 预先创建“飞驰篮球队”团体，包含创建者和若干成员。

### 5.2 基于系统级功能的测试分析与用例设计

|  |  |
| --- | --- |
| 用例编号 | Sys\_Test\_case\_User\_001 |
| 测试覆盖的系统业务 | 用户注册 (覆盖正常流程) |
| 用例设计方法 | 场景法，需求规约导出法综合 |
| 前置条件 | 无 |
| 输入 | |
| 初始输入 | 用户访问注册页面 |
| 预期交互输入、输出1 | 输入：输入注册类型、用户名和密码  输出：确认密码 |
| 预期交互输入、输出2 | 输入：确认密码、真实姓名和手机号  输出：提示注册成功，自动跳转至登录页面 |
| 最后预期输出 | |
| 注册成功，可使用新账号登录 | |

|  |  |  |
| --- | --- | --- |
| 用例编号 | Sys\_Test\_case\_User\_002 | |
| 测试覆盖的系统业务 | 用户注册（覆盖注册信息异常） | |
| 用例设计方法 | 边界值分析，错误推测法的综合 | |
| 前置条件 | 无 | |
| 输入 | | |
| 初始输入 | 用户访问登录页，点击注册账号按钮进入注册界面 |
| 预期交互输入、输出1 | 输入：正确输入用户名，密码，确认密码，真实姓名；再输入超过11位的账号。  输出：提示手机号格式不对，注册失败 |
| 最后预期输出 | | |
| 注册失败 | | |

|  |  |
| --- | --- |
| 用例编号 | Sys\_Test\_case\_User\_003 |
| 测试覆盖的系统业务 | 用户登录（覆盖密码错误） |
| 用例设计方法 | 边界值分析，错误推测法 |
| 前置条件 | 用户已注册。 |
| 输入 | |
| 初始输入 | 用户访问登录页 | |
| 预期交互输入、输出1 | 输入：用户名"1234"，错误密码"12345"，点击登录  输出：提示"密码错误，请重新输入" | |
| 最后预期输出 | |
| 登录失败，停留在登录页面 | |

|  |  |
| --- | --- |
| 用例编号 | Sys\_Test\_case\_User\_004 |
| 测试覆盖的系统业务 | 用户登录（覆盖正常流程） |
| 用例设计方法 | 场景法，需求规约导出法的综合 |
| 前置条件 | 用户已注册。 |
| 输入 | |
| 初始输入 | 用户访问登录页 | |
| 预期交互输入、输出1 | 输入：用户名"1234"，密码"1234"，点击登录  输出：提示"登录成功"，进入首页 | |
| 最后预期输出 | |
| 登录成功，进入首页 | |

| 用例编号 | Sys\_Test\_case\_User\_005 |
| --- | --- |
| 测试覆盖的系统业务 | 用户注册（用户名已被注册） |
| 用例设计方法 | 错误推测法 |
| 前置条件 | 用户名“1234”已注册 |
| 输入 | |
| 初始输入 | 用户访问注册页面 |
| 预期交互输入、输出1 | 输入：用户名“1234”，设置密码，确认密码<br>输出：提示“用户名已注册，注册失败” |
| 最后预期输出 | |
| 注册失败，停留在注册页面 | |

|  |  |
| --- | --- |
| 用例编号 | Sys\_Test\_case\_User\_006 |
| 测试覆盖的系统业务 | 用户退出登录 |
| 用例设计方法 | 场景法 |
| 前置条件 | 用户已登录 |
| 输入 | |
| 初始输入 | 用户点击“退出登录”按钮 |
| 预期交互输入、输出1 | 输出：提示“已退出登录”，跳转到登录页面 |
| 最后预期输出 | |
| 用户退出系统，需重新登录才能访问受限页面 | |

|  |  |
| --- | --- |
| 用例编号 | Sys\_Test\_case\_Reserve\_001 |
| 测试覆盖的系统业务 | 个人场地预约 (覆盖正常流程) |
| 用例设计方法 | 场景法，需求规约导出法综合 |
| 前置条件 | 用户已登录；目标场地在所选时间段可用。 |
| 输入 | |
| 初始输入 | 用户登录系统 |
| 预期交互输入、输出1 | 输入：搜索/浏览找到场地，点击立即预约；  输出：进入场地预约详情/表单页； |
| 预期交互输入、输出2 | 输入：选择预约类型个人，时间段2024-10-01 19:00-21:00，人数5，点击提交预约；  输出：预约成功提； |
| 预期交互输入、输出3 | 无 |
| 预期交互输入、输出4 | 无 |
| 最后预期输出 | |
| 个人预约订单创建成功，可在我的预约中查看。 | |

|  |  |
| --- | --- |
| 用例编号 | Sys\_Test\_case\_Reserve\_002 |
| 测试覆盖的系统业务 | 发起拼场预约 (覆盖正常流程) |
| 用例设计方法 | 场景法，需求规约导出法的综合 |
| 前置条件 | 用户已登录；目标场地在所选时间段可用。 |
| 输入 | |
| 初始输入 | 用户登录系统 |
| 预期交互输入、输出1 | 输入：搜索/浏览找到场地，点击立即预约；  输出：进入场地预约详情/表单页； |
| 预期交互输入、输出2 | 输入：选择预约类型拼场，时间段2024-10-02 14:00-16:00，总人数8，需招募7人，点击提交预约；  输出：生成预约订单； |
| 预期交互输入、输出3 | 无 |
| 最后预期输出 | |
| 拼场预约发起成功，状态为招募中，可在我的拼场/我的预约中查看。 | |

|  |  |  |
| --- | --- | --- |
| 用例编号 | Sys\_Test\_case\_Reserve\_003 | |
| 测试覆盖的系统业务 | 团体预约场地 (覆盖正常流程) | |
| 用例设计方法 | 场景法，需求规约导出法的综合 | |
| 前置条件 | 用户（团队创建者）已登录；目标场地在所选时间段可用。 | |
| 输入 | | |
| 初始输入 | 用户登录系统 |
| 预期交互输入、输出1 | 输入：进入我的团体，选择飞驰篮球队，点击团体预约；  输出：进入团体预约场地选择页； |
| 预期交互输入、输出2 | 输入：搜索/浏览找到篮球场C，选择时间段2024-10-03 20:00-22:00，人数12，点击提交预约；  输出：生成团体预约订单； |
| 预期交互输入、输出3 | 无 |
| 最后预期输出 | | |
| 团体预约订单创建成功，状态为已支付，可在团体预约记录和我的预约（创建者）中查看。 | | |

|  |  |
| --- | --- |
| 用例编号 | Sys\_Test\_case\_Reserve\_004 |
| 测试覆盖的系统业务 | 取消个人预约（在允许时间内） |
| 用例设计方法 | 场景法，需求规约导出法的综合 |
| 前置条件 | 用户有未开始的个人预约订单 |
| 输入 | |
| 初始输入 | 用户登录系统 |
| 预期交互输入、输出1 | 输入：进入"我的预约"，选择预约记录12，点击进入详情页；  输出：进入详情页 |
| 预期交互输入、输出2 | 输入：在详情页点击"取消预约"按钮  输出：取消成功提示，状态变"已取消" |
| 最后预期输出 | |
| 预约取消成功，释放场地时段 | |

|  |  |
| --- | --- |
| 用例编号 | Sys\_Test\_case\_venue\_001 |
| 测试覆盖的系统业务 | 创建运动场馆 (覆盖正常流程) |
| 用例设计方法 | 场景法，需求规约导出法的综合 |
| 前置条件 | 无 |
| 输入 | |
| 初始输入 | 进入登录界面，点击创建账号 | |
| 预期交互输入、输出1 | 输入：注册类型选择"场地管理员"，输入用户名"羽毛球场1"，密码“1234”，确认密码，真实姓名“羽毛球”，电话“12237785564”  输出：注册； | |
| 预期交互输入、输出2 | 输入：使用注册后的账号登录  输出：进入场地管理界面； | |
| 预期交互输入、输出3 | 输入：设置场馆名称“羽毛球场”，介绍“羽毛球”，地点“五角场”,联系电话“12345678”，上传图片后点击“保存更改”  输出：提示“编辑场馆信息成功”； | |
| 最后预期输出 | |
| 场馆创建成功，用户可以在场馆列表中看见该场馆。 | |

|  |  |
| --- | --- |
| 用例编号 | Sys\_Test\_case\_venue\_002 |
| 测试覆盖的系统业务 | 创建场地并设置开放时间 (覆盖正常流程) |
| 用例设计方法 | 场景法，需求规约导出法的综合 |
| 前置条件 | 已经注册场馆并登录场地管理员账号 |
| 输入 | |
| 初始输入 | 登录账号并进入场地管理界面 | |
| 预期交互输入、输出1 | 输入：点击“添加场地”按钮创建新场地，输入场地名称“羽毛球场1号”、容量2、类型“羽毛球场”、状态“开放”、位置“二楼东”，点击保存  输出：新场地创建成功 | |
| 预期交互输入、输出2 | 输入：进入开放时间段管理界面，选择“羽毛球场1号”，点击添加配置  输出：进入配置页面； | |
| 预期交互输入、输出3 | 输入：输入配置名称“时间组1号”，点击“添加时间段”，输入时间“10.00”-“11.00”，价格 10.00元，提前预约天数1天，开放预约时间4时，重复 周一至周日，点击“保存”  输出：提示“已创建配置项”； | |
| 最后预期输出 | |
| 场地创建成功，用户可以在规定时间预约该场地。 | |

| 用例编号 | Sys\_Test\_case\_venue\_003 |
| --- | --- |
| 测试覆盖的系统业务 | 设置场地开放状态 |
| 用例设计方法 | 场景法，需求规约导出法 |
| 前置条件 | 管理员已登录 |
| 输入 | |
| 初始输入 | 管理员在场地管理界面切换场地开放/关闭状态 |
| 预期交互输入、输出1 | 输出：提示“设置成功”，场地状态更新 |
| 最后预期输出 | |
| 场地状态变更，前台预约页面同步显示最新状态 | |

|  |  |
| --- | --- |
| 用例编号 | Sys\_Test\_case\_venue\_004 |
| 测试覆盖的系统业务 | 场地评价 |
| 用例设计方法 | 场景法，需求规约导出法 |
| 前置条件 | 用户有已完成的预约 |
| 输入 | |
| 初始输入 | 用户在预约详情页点击“评价”并输入内容 |
| 预期交互输入、输出1 | 输入：评分、评价内容，点击提交<br>输出：提示“评价成功”，评价显示在场地详情页 |
| 最后预期输出 | |
| 评价被保存并展示 | |

|  |  |
| --- | --- |
| 用例编号 | Sys\_Test\_case\_social\_001 |
| 测试覆盖的系统业务 | 发送好友请求 (覆盖正常流程) |
| 用例设计方法 | 场景法，需求规约导出法的综合 |
| 前置条件 | 已经登录 |
| 输入 | |
| 初始输入 | 登录账号并进入好友列表界面 | |
| 预期交互输入、输出1 | 输入：点击“添加好友”  输出：显示用户列表 | |
| 预期交互输入、输出2 | 输入：点击待添加用户条目右侧的添加按钮  输出：弹出详情页； | |
| 预期交互输入、输出3 | 输入：输入“申请说明”，点击“确定”  输出：提示“已发送好友申请”； | |
| 最后预期输出 | |
| 好友申请已发送到目标用户。 | |

|  |  |
| --- | --- |
| 用例编号 | Sys\_Test\_case\_Social\_002 |
| 测试覆盖的系统业务 | 拒绝好友申请 |
| 用例设计方法 | 场景法，需求规约导出法综合 |
| 前置条件 | 用户收到好友申请 |
| 输入 | |
| 初始输入 | 登录账号并进入好友申请界面 |
| 预期交互输入、输出1 | 输入：点击"拒绝"  输出：申请状态变"已拒绝" |
| 最后预期输出 | |
| 申请被拒绝，双方无好友关系 | |

|  |  |
| --- | --- |
| 用例编号 | Sys\_Test\_case\_Social\_003 |
| 测试覆盖的系统业务 | 接受好友申请 |
| 用例设计方法 | 场景法，需求规约导出法综合 |
| 前置条件 | 用户收到好友申请 |
| 输入 | |
| 初始输入 | 登录账号并进入好友申请界面 |
| 预期交互输入、输出1 | 输入：点击"同意"  输出：申请状态变"已同意" |
| 最后预期输出 | |
| 申请通过，双方建立好友关系 | |

|  |  |
| --- | --- |
| 用例编号 | Sys\_Test\_case\_Social\_004 |
| 测试覆盖的系统业务 | 添加好友时输入不存在的用户ID |
| 用例设计方法 | 错误推测法 |
| 前置条件 | 无 |
| 输入 | |
| 初始输入 | 用户在添加好友界面输入用户ID“99999999” |
| 预期交互输入、输出1 | 输出：提示“用户不存在” |
| 最后预期输出 | 添加好友失败 |

| 用例编号 | Sys\_Test\_case\_Social\_005 |
| --- | --- |
| 测试覆盖的系统业务 | 在群聊中发送信息 |
| 用例设计方法 | 场景法，需求规约导出法 |
| 前置条件 | 用户已加入群聊 |
| 输入 | |
| 初始输入 | 用户在群聊界面输入消息内容并发送 |
| 预期交互输入、输出1 | 输出：消息显示在群聊窗口，其他成员可见 |
| 最后预期输出 | |
| 消息实时同步，群聊正常互动 | |

|  |  |
| --- | --- |
| 用例编号 | Sys\_Test\_case\_Social\_006 |
| 测试覆盖的系统业务 | 创建群聊 |
| 用例设计方法 | 场景法，需求规约导出法 |
| 前置条件 | 创建者和被拉入群聊的用户是好友关系 |
| 输入 | |
| 初始输入 | 用户进入群聊界面 |
| 预期交互输入、输出1 | 输入：用户点击创建群聊，选择要拉入群的好友，输入群聊名称和信息，点击创建  输出：相关人员被拉入群聊中 |
| 最后预期输出 | |
| 所有选中用户都加入该群聊 | |

|  |  |
| --- | --- |
| 用例编号 | Sys\_Test\_case\_Group\_001 |
| 测试覆盖的系统业务 | 创建运动团体 |
| 用例设计方法 | 场景法 |
| 前置条件 | 无 |
| 输入 | |
| 初始输入 | 登录账号并进入创建团体界面 |
| 预期交互输入、输出1 | 输入：输入团体名称“羽毛球团”，团体描述“羽毛球爱好者”，点击“邀请好友”  输出：弹出“邀请用户”选择页 |
| 预期交互输入、输出2 | 输入：选择要邀请的好友，点击确定  输出：待邀请团体成员增加 |
| 预期交互输入、输出3 | 输入：点击“创建团体”  输出：团体创建成功； |
| 最后预期输出 | |
| 团体创建成功，已经向成员发送邀请 | |

|  |  |
| --- | --- |
| 用例编号 | Sys\_Test\_case\_Group\_002 |
| 测试覆盖的系统业务 | 团体邀请成员 |
| 用例设计方法 | 场景法 |
| 前置条件 | 用户是管理员 |
| 输入 | |
| 初始输入 | 登录账号并进入团体列表/我的团体界面 |
| 预期交互输入、输出1 | 输入：选择团体“羽毛球团”，点击“查看详情”  输出：进入团体详情页 |
| 预期交互输入、输出2 | 输入：点击“管理成员”  输出：进入管理团体成员页面 |
| 预期交互输入、输出3 | 输入：选择好友进行邀请  输出：已向好友发送邀请； |
| 最后预期输出 | |
| 邀请已发送，等待审核 | |

| 用例编号 | Sys\_Test\_case\_Group\_003 |
| --- | --- |
| 测试覆盖的系统业务 | 把团体中成员设置为团体管理员 |
| 用例设计方法 | 场景法，需求规约导出法 |
| 输入 | |
| 前置条件 | 团体已存在，当前用户为团体管理员 |
| 初始输入 | 管理员在团体成员列表中选择成员并设置为管理员 |
| 预期交互输入、输出1 | 输出：提示“设置成功”，成员权限变更 |
| 最后预期输出 | |
| 该成员拥有管理员权限，可进行管理操作 | |

|  |  |
| --- | --- |
| 用例编号 | Sys\_Test\_case\_Group\_004 |
| 测试覆盖的系统业务 | 用户接收团体邀请 |
| 用例设计方法 | 场景法，需求规约导出法 |
| 前置条件 | 用户收到团体邀请通知 |
| 输入 | |
| 初始输入 | 用户在通知中心点击“同意” |
| 预期交互输入、输出1 | 输出：提示“加入成功”，用户成为团体成员 |
| 最后预期输出 | |
| 用户加入团体，成员列表更新 | |

|  |  |
| --- | --- |
| 用例编号 | Sys\_Test\_case\_Group\_005 |
| 测试覆盖的系统业务 | 用户申请加入团体 |
| 用例设计方法 | 场景法，需求规约导出法 |
| 前置条件 | 用户未加入该团体 |
| 输入 | |
| 初始输入 | 用户登录 |
| 预期交互输入、输出1 | 输入：用户进入团体列表界面，搜素要加入的团体  输出：页面展示搜索到的团体 |
| 预期交互输入、输出2 | 输入：用户点击加入该团体，填写申请理由，确认发送申请  输出：申请被送给团体管理员 |
| 最后预期输出 | |
| 申请等待团体管理员审核 | |

### 5.3 基于系统业务流测试分析与用例设计

|  |  |  |
| --- | --- | --- |
| 用例编号 | | Sys\_Test\_case\_business\_001 |
| 测试覆盖的系统业务 | | 用户登录 -> 选择场地 -> 发起个人预约 -> 预约成功 ->取消预约->再次预约-> 用户进行场地评价->退出登录 |
| -用例设计方法 | | 场景法，需求规约导出法的综合 |
| 前置条件 | | 场地在对应时间段内可用；用户已注册。 |
| 输入 | | |
| 初始输入 | 用户进入网站 | | |
| 预期交互输入、输出1 | 输入：用户名123456, 密码123456;  输出：登录成功; | | |
| 预期交互输入、输出2 | 输入：点击“篮球场”  输出：进入“篮球场”详情页 | | |
| 预期交互输入、输出3 | 输入：选择当前时间后两天的11：00-14：00时间段的场地进行个人预约  输出：预约成功； | | |
| 预期交互输入、输出4 | 输入：进入预约记录，点击刚才的预约记录，取消预约  输出：预约被取消，场地状态被重置 | | |
| 预期交互输入、输出5 | 输入：再次进入相同的场馆，找到刚才预约过的场地，再次预约  输出：预约成功 | | |
| 预期交互输入、输出6 | 输入：进入场地详情页，发表评论“场地很好，值得推荐”，推荐指数4颗星，点击发表  输出：评论发布成功 | | |
| 预期交互输入、输出7 | 输入：进入个人信息页面，点击“退出登录”  输出：登录退出成功 | | |
| 最后预期输出 | | |
| 整个个人预约流程（预约->取消预约->预约->->评价）顺利完成，所有相关状态更新正确。 | | |

|  |  |
| --- | --- |
| 用例编号 | Sys\_Test\_case\_business\_002 |
| 测试覆盖的系统业务 | 用户登录->添加好友->同意好友请求->发消息->回复消息->用户退出 |
| 用例设计方法 | 场景法，需求规约导出法的综合 |
| 前置条件 | 两人不是好友关系 |
| 输入 | |
| 初始输入 | 用户登录系统 |
| 预期交互输入、输出1 | 输入：用户进入好友列表，点击添加好友，输入用户名称“testuser14”，点击搜索  输出：找到该用户，并且可以添加好友 |
| 预期交互输入、输出2 | 输入：点击添加，填写原因“我是testuser01”,点击确认  输出：好友申请已发送给testuser14 |
| 预期交互输入、输出3 | 输入：testuser14进入好友申请界面，同意testuser01的好友申请  输出：testuser01和testuser14成为好友 |
| 预期交互输入、输出4 | 输入：testuser14进入好友列表，找到testuser01，发送消息“你好你好”，“我是testuser14”  输出：消息成功发送 |
| 预期交互输入、输出5 | 输入：testuser01接收到消息，回复消息“收到”，“我是testuser01”  输出：消息成功发送 |
| 最后预期输出 | |
| 整个好友社交流程（发送申请->同意申请->发送消息->收到消息）顺利完成，所有相关状态更新正确。 | |

|  |  |
| --- | --- |
| 用例编号 | Sys\_Test\_case\_business\_003 |
| 测试覆盖的系统业务 | 用户创建团体 -> 邀请成员加入 ->成员确认加入-> 在群聊中发送信息 -> 进行团体预约 -> 团体成员查看预约 |
| 用例设计方法 | 场景法，需求规约导出法的综合 |
| 前置条件 | 用户已和备选团员成为好友；场地F在时间段内可用。 |
| 输入 | |
| 初始输入 | 用户登录系统 |
| 预期交互输入、输出1 | 输入：用户点击创建团体，团体名“测试团体”，邀请团员“testuser11”“testuser12”“testuser13”，点击确认  输出：团体邀请已发送给受邀用户 |
| 预期交互输入、输出2 | 输入：3名用户分别接受，拒绝，接受团体邀请  输出：团体成员已更新 |
| 预期交互输入、输出3 | 输入：加入团体的成员在团体专属群聊中发消息  输出：团员都能收到消息 |
| 预期交互输入、输出4 | 输入：团体管理员testuser01进行团体预约，并邀请所有团员  输出：所有团员都收到了团体预约 |
| 最后预期输出 | |
| 团体创建、成员邀请、团体预约流程顺利完成，团体成员能查看预约信息。 | |

|  |  |
| --- | --- |
| 用例编号 | Sys\_Test\_case\_business\_004 |
| 测试覆盖的系统业务 | 注册场地管理员->完善场馆信息->添加新场地->为场地添加时间段->开放场馆和场地->用户在场馆列表中找到场馆 |
| 用例设计方法 | 场景法，需求规约导出法的综合 |
| 前置条件 | 无 |
| 输入 | |
| 初始输入 | 无 |
| 预期交互输入、输出1 | 输入：点击注册用户，选择类型“场地管理员”，输入用户名，密码，确认密码，真实姓名，电话号码  输出：成功注册场地管理员和对应的场地 |
| 预期交互输入、输出2 | 输入：使用注册后的账户进入，修改场地信息，设置场馆状态为开放，点击保存  输出：场馆信息和状态已更新 |
| 预期交互输入、输出3 | 输入：为场馆添加新场地，输入场地信息，点击保存  输出：场馆增加了场地 |
| 预期交互输入、输出4 | 输入：在开放时间段配置中为新场地增加时间段配置，设置开放时间等内容，点击保存  输出：场地的时间段已配置 |
| 预期交互输入、输出5 | 输入：使用用户账户登录，搜索刚才创建的场地，点击进入该场地  输出：场馆和场地的详细信息已经显示 |
| 最后预期输出 | |
| 场地管理员注册，设置场馆信息，添加场地，配置开放时间段流程顺利完成，用户能看到新添加的场馆和场地 | |

## 测试结果

### 6.1系统级功能结果

|  |  |  |
| --- | --- | --- |
| 用例编号 | 用例描述 | 测试结果 |
| Sys\_Test\_case\_User\_001 | 用户注册 (覆盖正常流程) | 通过 |
| Sys\_Test\_case\_User\_002 | 用户注册（覆盖注册信息异常） | 异常（缺陷#1，缺陷#4） |
| Sys\_Test\_case\_User\_003 | 用户登录（覆盖密码错误） | 通过 |
| Sys\_Test\_case\_User\_004 | 用户登录（覆盖正常流程） | 通过 |
| Sys\_Test\_case\_User\_005 | 用户注册（用户名已被注册） | 通过 |
| Sys\_Test\_case\_User\_006 | 用户退出登录 | 通过 |
| Sys\_Test\_case\_Reserve\_001 | 个人场地预约 (覆盖正常流程) | 通过 |
| Sys\_Test\_case\_Reserve\_002 | 发起拼场预约 (覆盖正常流程) | 通过 |
| Sys\_Test\_case\_Reserve\_003 | 团体预约场地 (覆盖正常流程) | 通过 |
| Sys\_Test\_case\_Reserve\_004 | 取消个人预约（在允许时间内） | 通过 |
| Sys\_Test\_case\_venue\_001 | 创建运动场馆 (覆盖正常流程) | 通过 |
| Sys\_Test\_case\_venue\_002 | 创建场地并设置开放时间 (覆盖正常流程) | 通过 |
| Sys\_Test\_case\_venue\_003 | 设置场地开放状态 | 通过 |
| Sys\_Test\_case\_venue\_004 | 场地评价 | 通过 |
| Sys\_Test\_case\_social\_001 | 发送好友请求 (覆盖正常流程) | 通过 |
| Sys\_Test\_case\_Social\_002 | 拒绝好友申请 | 通过 |
| Sys\_Test\_case\_Social\_003 | 接受好友申请 | 通过 |
| Sys\_Test\_case\_Social\_004 | 添加好友时输入不存在的用户ID | 通过 |
| Sys\_Test\_case\_Social\_005 | 在群聊中发送信息 | 通过 |
| Sys\_Test\_case\_Social\_006 | 创建群聊 | 异常（缺陷#3） |
| Sys\_Test\_case\_Group\_001 | 创建运动团体 | 通过 |
| Sys\_Test\_case\_Group\_002 | 团体邀请成员 | 通过 |
| Sys\_Test\_case\_Group\_003 | 把团体中成员设置为团体管理员 | 通过 |
| Sys\_Test\_case\_Group\_004 | 用户接收团体邀请 | 通过 |
| Sys\_Test\_case\_Group\_005 | 用户申请加入团体 | 异常（缺陷#2） |

测试用例总计 25个 ， 通过 22 个 ，出现异常的有 3 个

### 6.2 系统业务流测试

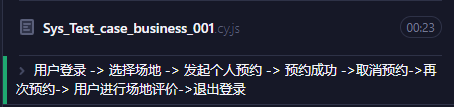
我们使用 Cypress工具帮助我们来执行系统业务流测试，并通过捕获页面上的成功信息和比较界面上的显示信息来确定测试用例是通过还是失败。测试方法为黑盒测试。

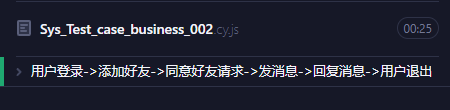
代码如下：

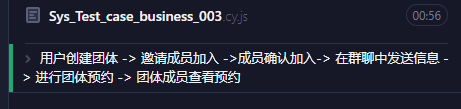
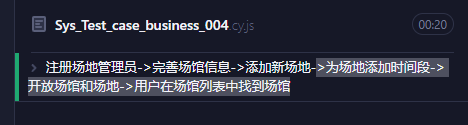


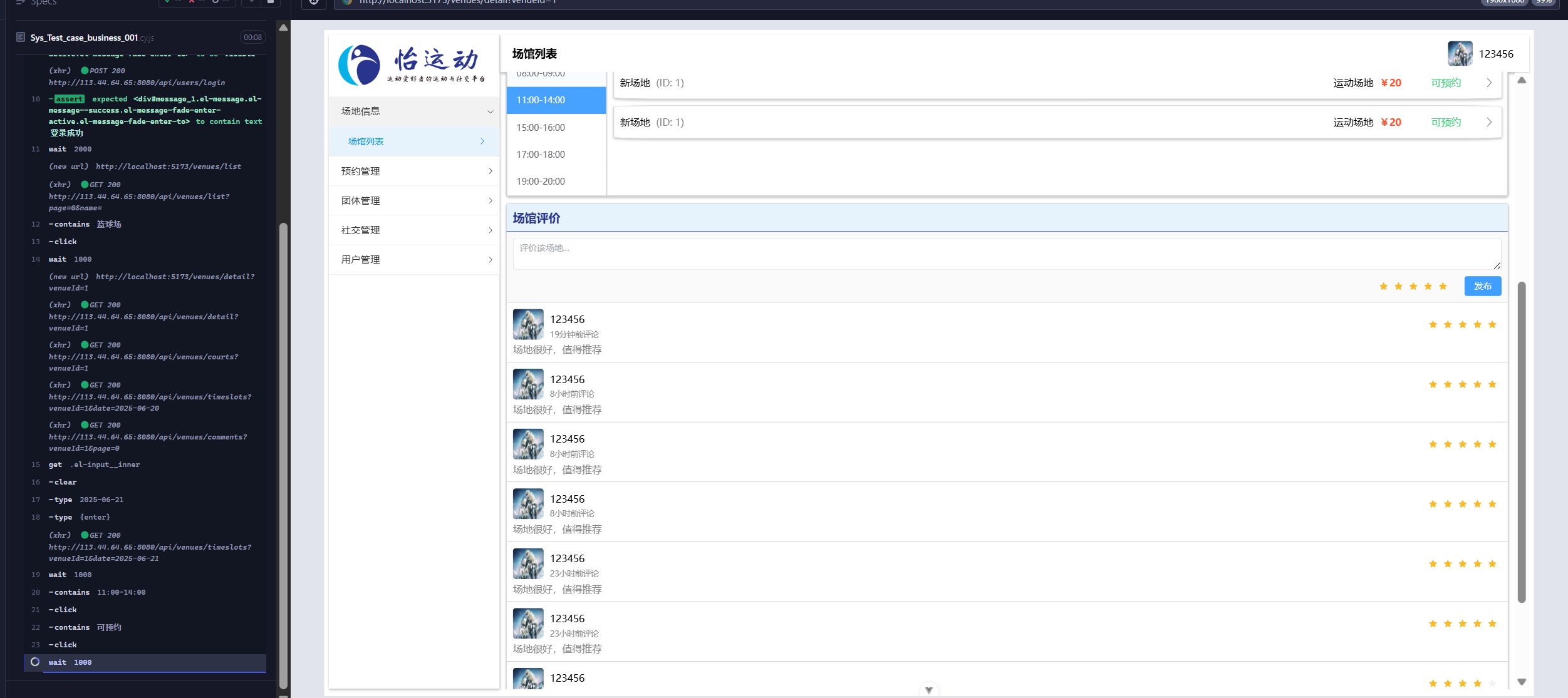


测试结果（部分）：









### 6.3系统测试缺陷跟踪管理

#### 6.3.1. 系统测试用例与缺陷标注表

| **缺陷编号** | **用例编号** | **用例名称** | **预期结果** | **实际结果** | **是否通过** | **缺陷简述** |
| --- | --- | --- | --- | --- | --- | --- |
| #1 | Sys\_Test\_case\_User\_002 | 注册手机号输入 | 输入超11位应提示错误 | 未提示，注册失败 | 否 | 未做输入容错性检查 |
| #2 | Sys\_Test\_case\_Social\_006 | 团体创建人身份校验 | 创建人不应申请加入团体 | 创建人也可申请加入团体 | 否 | 创建人身份校验逻辑缺失 |
| #3 | Sys\_Test\_case\_Group\_001 | 群聊创建 | 至少2人才能建群聊 | 1人也可建群聊 | 否 | 群聊人数下限未校验 |
| #4 | Sys\_Test\_case\_User\_002 | 用户真实姓名校验 | 单字名应正常处理 | 单字名导致错误 | 否 | 单字名处理逻辑有误 |

#### 6.3.2. 缺陷跟踪表

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **缺陷编号** | **标题** | **提出人** | **负责人** | **严重程度** | **当前状态** | **处理过程说明** |
| #1 | 注册时手机号输入 | to-dreamboat | 123-123-1 | bug | Open | 发现注册手机号超11位未校验，已指派开发，待修复 |
| #2 | 团体创建人身份校验 | to-dreamboat | 123-123-1 | critical bug | Open | 发现团体创建人也可申请入团体，已指派开发，待修复 |
| #3 | 单人可创建群聊 | to-dreamboat | 123-123-1 | minor bug | Open | 发现1人也可建群聊，已指派开发，待修复 |
| #4 | 用户真实姓名单字问题 | to-dreamboat | 123-123-1 | minor bug | Open | 发现单字名导致预约时姓名处理出错，已指派开发，待修复 |

#### 6.3.3. 缺陷跟踪过程说明

以“注册时手机号输入”Bug为例：

1. 缺陷发现：在系统测试用例TC01执行时，发现注册手机号输入超11位未做容错性校验，导致注册失败。
2. 缺陷登记：测试人员在GitHub Issues新建Issue，详细描述缺陷现象、重现步骤，并添加“bug”标签。
3. 缺陷分配：将该Issue指派给相关开发人员（123-123-1）。
4. 缺陷处理：开发人员分析并修复代码，提交PR并在PR描述中引用，合并后Issue自动关闭。
5. 回归验证：测试人员验证修复效果，若通过则Issue保持关闭，否则重新打开并补充说明。
6. 缺陷关闭：修复验证通过后，Issue状态为Closed，缺陷生命周期结束。

## 7. 性能测试

### 7.1 测试目的

压力测试在性能测试中是一个重要的环节，其主要目的是模拟系统在正常或峰值负载下的表现，以评估系统的稳定性、可靠性和性能指标。通过在系统承受高负载的情况下进行压力测试，可以识别系统的性能瓶颈，即系统在特定负载下的性能瓶颈点。这有助于开发团队针对性地优化和改进系统，提升性能。

### 7.2 测试工具

JMeter 是一款开源的性能测试工具，用于模拟多种协议的负载，并收集性能指标和测试结果。它能够模拟大量用户并发送请求到目标服务器，以评估系统的性能表现。本次测试将使用 JMeter 进行压力负载测试，并使用华为云服务器任务管理器观察服务器的系统资源与性能指标。

### 7.3 测试需求分析

一般来说，需要进行压力测试的通常满足以下条件：核心业务功能、用户量以及与外部接口的对接。登录是系统的关键功能之一，如果登录功能在高负载下出现性能问题，会导致用户无法正常访问系统的其他功能，因此需要对其进行压力测试。

性能指标如下：

* 非硬件指标：
* 50%的请求响应时间小于1秒
* 90%的请求响应时间小于1秒
* TPS（每秒事务数）
* 事务成功率达到100%（响应时间在几十毫秒到几百毫秒之间）
* 硬件指标：
* CPU 内存占用率不超过70%

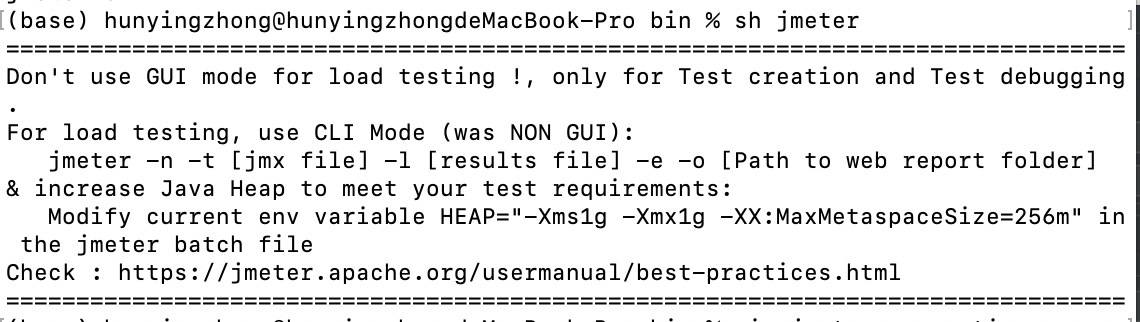
### 7.4 测试方案设计

性能测试场景设计如下：

1. 在1秒内启动所有线程（共100个），并进行为期5分钟的压力测试，观察性能指标并记录结果。
2. 在1秒内启动所有线程（共3000个），并进行为期5分钟的压力测试，再次观察性能指标并记录结果。

### 7.5 测试过程

1. 配置测试环境：确保 JMeter 已正确安装，并配置好测试环境，包括服务器端和客户端。

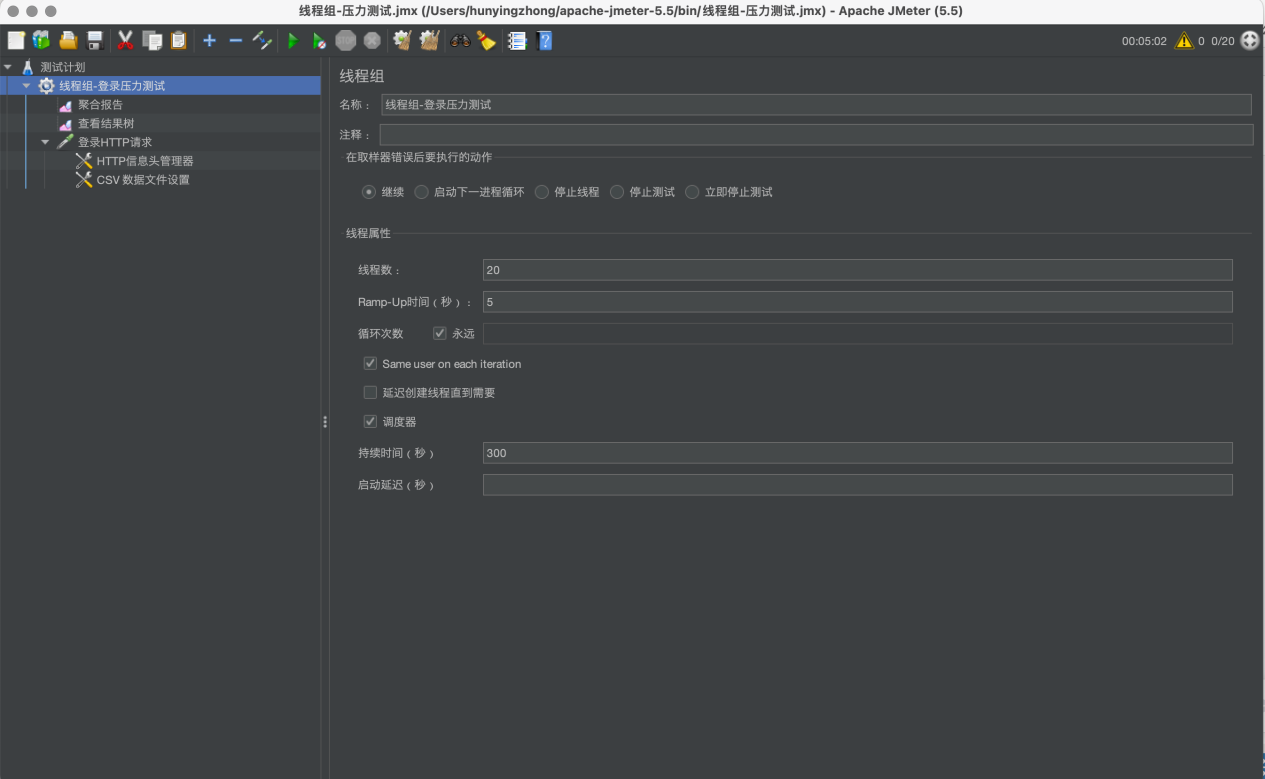


1. 创建测试计划和配置线程组：在 JMeter 中创建一个新的测试计划。设置线程组的属性，包括线程数、循环次数和启动延迟等。根据性能需求分析中确定的性能场景，配置线程组启动所有线程的时间间隔为5s。

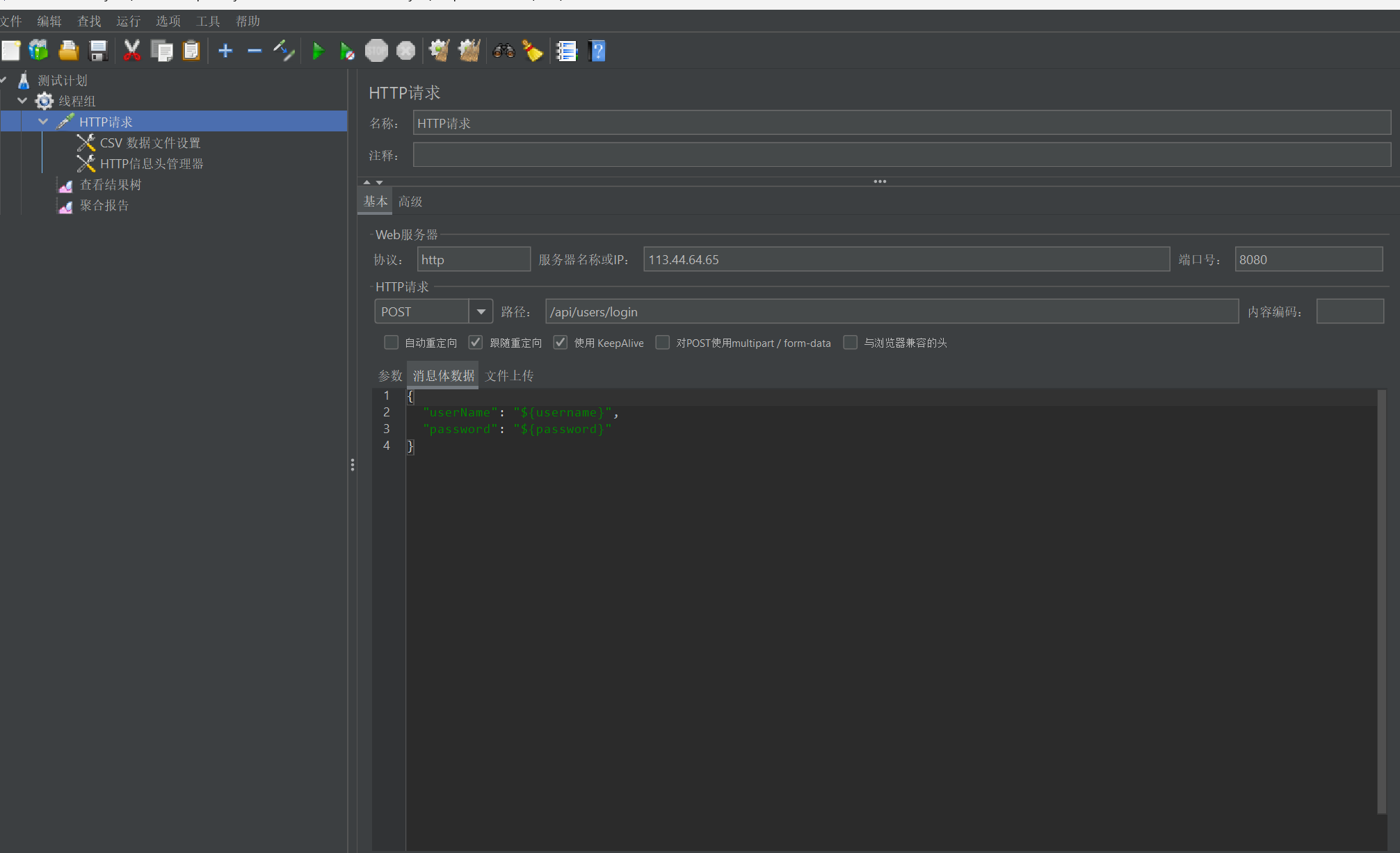
一个线程组就是一组虚拟用户(**virtual users**)，这些虚拟用户用来模拟访问被测系统。

**1**)线程数:这里就是指虚拟用户数，默认的输入是“1”，则表明模拟一个虚拟用户访 问被测系统，如果想模拟 100 个用户，则此处输入 100。  
 **2**)**Ramp-Up Period (in seconds):** 虚拟用户增长时长。

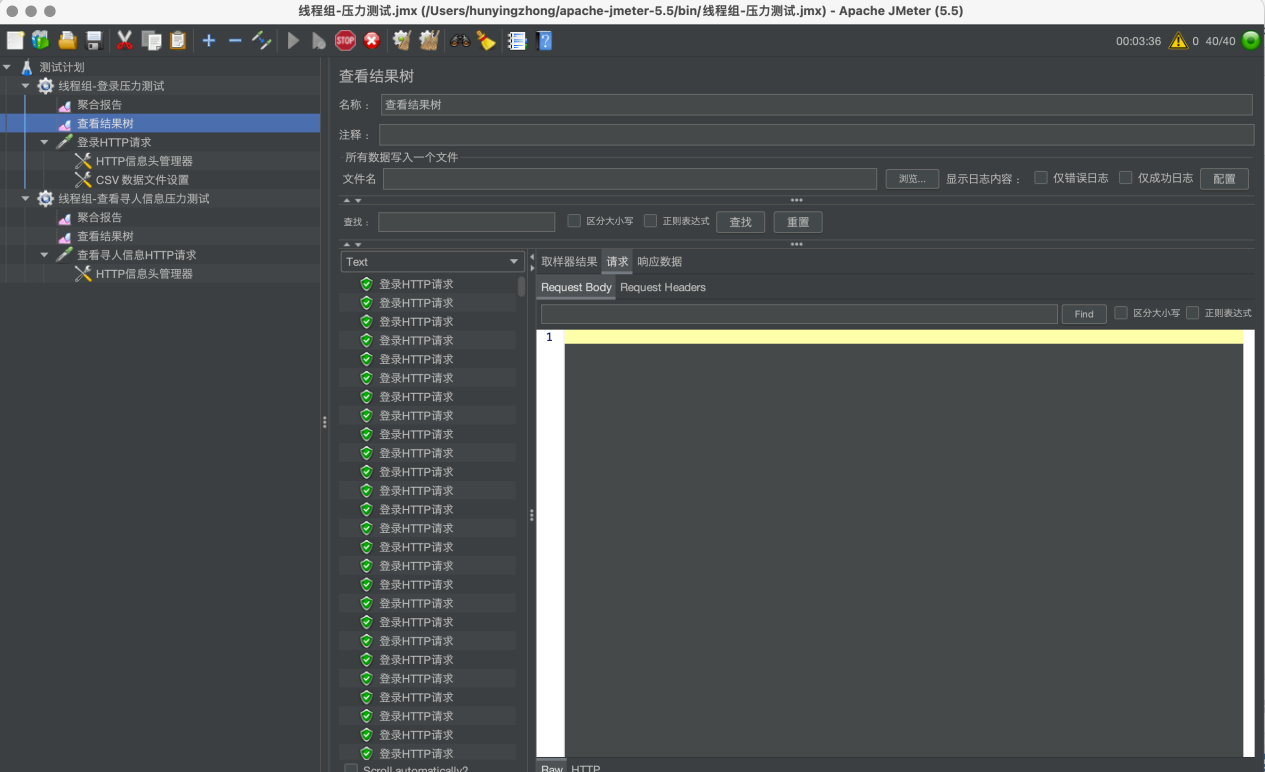
**3**)循环次数:该处设置一个虚拟用户做多少次的测试。默认为 1，意味着一个虚拟用 做完一遍事情之后，该虚拟用户停止运行。

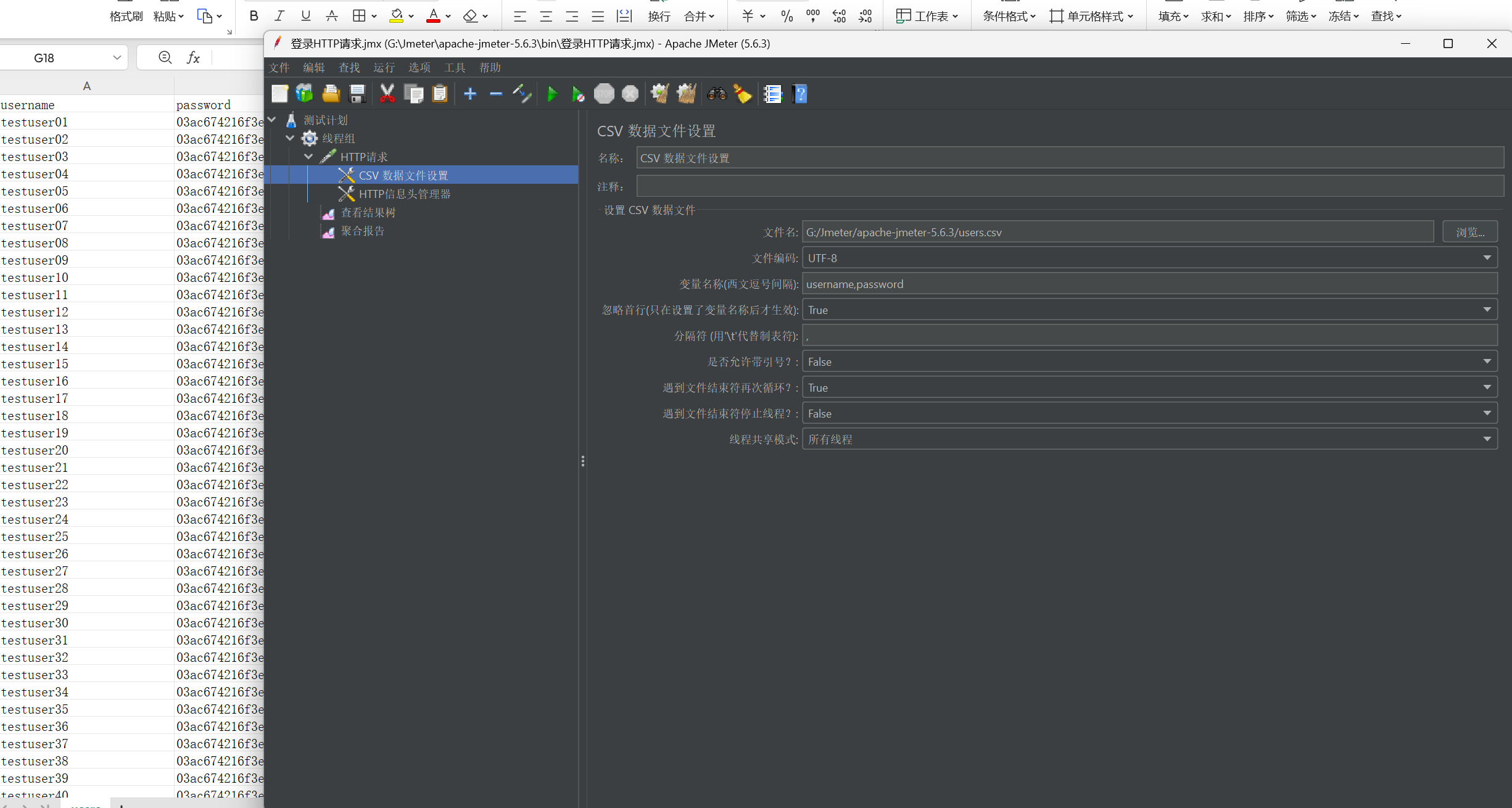


1. 添加请求：根据需要测试的业务功能，在测试计划中添加相应的登录请求。配置请求的参数和目标服务器的地址。



1. 添加监听器：选择聚合报告和查看结果树来监控性能指标和收集测试结果，显示响应时间、吞吐量和错误率等重要指标。



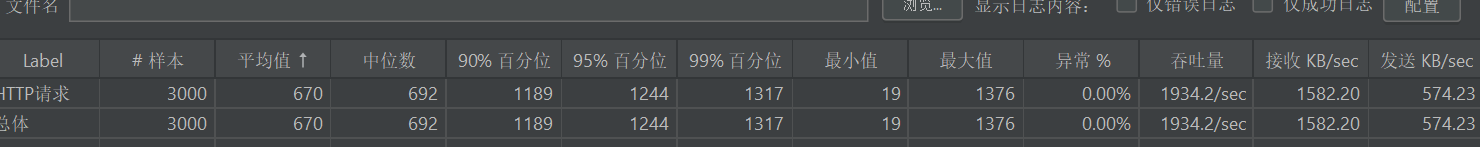
1. ****添加CSV数据文件设置：从数据库中将账号和密码导出存入数据库中。在测试计划中添加CSV数据文件配置元件。配置CSV文件的路径和字段分隔符，并定义存储账号密码数据的列名。这样就可以根据csv文件中的账号和密码创建不同的用户。
2. 监控服务器资源：使用华为云服务器任务管理器观察服务器的系统资源和性能指标，包括 CPU 使用率、内存占用等。确保服务器在测试期间正常运行。
3. 收集和分析结果：测试运行完成后，使用 JMeter 的监听器和报告功能收集测试结果。分析响应时间、吞吐量和错误率等指标，以评估系统的性能表现。
4. 性能优化和调整：根据测试结果进行性能优化和调整。根据发现的性能缺陷，定位问题所属的系统业务模块，并采取相应的优化措施，如代码优化、资源调整或系统配置修改等。

### 7.6 测试结果

#### 7.6.1 登录性能

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 线程数 | 启动时长(秒) | 运行时长(秒) | 样本 | CPU使用率 | 50%line响应时间(毫秒) | 90%line响应时间(毫秒) | 成功率 |
| 100 | 1 | 300 | 100 | 21% | 498 | 920 | 100% |
| 3000 | 1 | 300 | 3000 | 21% | 692 | 1189 | 100% |

****

****

### 7.7 测试结论

服务器瓶颈：3000个线程

1. 响应时间分析

在100线程压力下，50%的请求响应时间为498ms，90%的请求响应时间为920ms，均低于1秒，系统能够快速响应用户请求。在3000线程高并发压力下，50%的请求响应时间为692ms，90%的请求响应时间为1189ms。虽然90%分位响应时间略高于1秒，但整体仍处于可接受范围，说明系统具备较强的并发处理能力。

1. 吞吐量与成功率

两组测试的事务成功率均为100%，无失败请求，系统在高并发场景下依然能够稳定处理所有登录请求，未出现超时或错误。TPS（每秒事务数）可通过样本数和运行时长计算，表现良好，满足业务需求。

1. 硬件资源消耗

测试期间CPU使用率均为21%，远低于70%的性能阈值，说明服务器资源充足，系统运行稳定，无资源瓶颈。内存占用未出现异常增长，系统未发生OOM等资源溢出问题。

1. 系统稳定性与可扩展性

在高并发压力下，系统未出现崩溃、卡顿等不稳定现象，表现出良好的稳定性和可扩展性。说明系统架构设计合理，能够支撑大规模用户同时在线登录的业务场景。

1. 性能瓶颈与优化建议

90%分位响应时间在极高并发下略有上升，建议后续可针对数据库连接池、缓存、接口优化等方面进一步提升性能。持续关注高并发下的资源利用率，预防极端场景下的性能退化。