uaBETA SPRINT · DAY 4 · 测试岗交付

不鸽组队平台

Beta 冲刺 Day4 测试报告

边界矩阵专项：数值、时间、信用、权限四条线逐条执行

项目名称　不鸽组队 —— 校园搭子组队平台

团队名称　不鸽工程队

报告类型　Beta 冲刺日交付测试报告

被测版本　master 分支 b879eef，2026-11-16 13:10 校验

编写人　王嘉萍（测试岗）

报告日期　2026 年 11 月 16 日

目 录

报告信息 1
一、执行摘要 1
二、测试方法与判定口径 2
2.1 环境与数据量 2
2.2 四条边界线的切法 3
2.3 两类分页语义必须分开断言 4
三、边界矩阵总览 4
四、数值边界 5
五、时间边界 5
六、信用边界 6
七、权限边界 7
八、评价负向与两处缺陷 8
8.1 重复评价被拦（通过） 8
8.2 BUG-4：越权评价可刷分（P1） 9
8.3 BUG-5：被评价人不存在返回 5000（P2） 10
九、消息边界 10
十、未覆盖项与原因 11
十一、全量回归与结论 12
十二、次日计划 13
十三、提交记录与 AI 使用记录 13
附录：复现方式 15

报告信息

项目	内容
被测系统	不鸽组队 —— 校园搭子组队平台（Spring Boot 3.2.5 + MySQL 8）
被测地址	http://127.0.0.1:8081（真实后端，非 mock）
数据库	bugedui（852 活动 / 6630 报名 / 2068 用户，含历史造数）
代码版本	master b879eef（李佳种子修复 + 施乐山评价页已合入）
测试框架	pytest + requests，断言到具体错误码
新增用例	43 条（边界矩阵），全量 65 条
执行结果	61 通过 / 0 失败 / 3 预期失败（xfail）/ 1 意外通过（xpass）
新增缺陷	2（BUG-4 P1、BUG-5 P2）
编写人 / 日期	王嘉萍（测试岗） / 2026-11-16

一、执行摘要

Day4 测试岗三项任务全部完成：边界矩阵按数值、时间、信用、权限四条线铺开并逐条执行，信用与消息用例完成自动化，越权评价与重复评价负向用例补齐。**34 个功能点、43 条用例**，其中 40 条通过、3 条为今日新发现缺陷的预期失败。

> **核心结论一　边界拦截全部生效，没有"越界 5000"**
>
> 分页 pageSize 0/51、pageNum 0、推荐条数 0/51、评分 0/6 全部返回 **4001**，不是 5000 也不是静默全量返回；合法边界 1/50 正常放行。说明 @Min/@Max 在 @RequestParam 上确实生效，并被全局异常统一转成业务码。

核心结论二　时间边界精确到秒，被拒不留脏数据

截止前 2 秒可报名、截止后 1 秒返回 **4013「报名已截止」**，已结束活动 4013；信用 59 被拦后 activity 的 current_people 仍为 1，名额没有被白占。

核心结论三　评价模块有两处必须修

BUG-4（P1）：互评未校验"参加过该活动 + 活动已结束"，**任意登录用户可给任意人评价并 +2 信用分**，换 activityId 即可重复刷分；BUG-5（P2）：被评价人不存在时返回 5000 而非 4040。两条均已写成 xfail 用例挂账，修复后自动转通过。

今日任务（05 清单）	交付物	状态
边界测试专项（数值/时间/信用/权限）	docs/Day4_边界矩阵_王嘉萍.md + tests/test_04_boundary.py	完成，43 条
信用与消息用例自动化	tests/test_05_credit.py、tests/test_06_message.py	完成，22 条
越权评价、重复评价负向	test_bd_e01 – e05 + reports/duplicate_eval_evidence.png	完成，3 条挂账
全量回归与当日报告	reports/Day4_测试报告_20261116_1321.md	完成，61 通过

二、测试方法与判定口径

2.1 环境与数据量

被测后端为本地 8081 端口启动的真实实例，连接 bugedui 库。报告编写时的库内快照：852 个活动（招募中 128 / 已结束 712 / 已取消 12）、6630 条报名、2068 个用户。绝大部分活动来自 Day1 造数，早已过了结束时间、状态沉到 3，**列表默认视图只返回状态 1/2，因此可见的只剩 128 条**。这一点直接影响用例设计：

受影响的写法	后果	本报告的处理
依赖"库里已有活动"断言列表非空	历史数据到期后列表为空，用例假失败	每条用例自建活动，断言自己的数据
依赖历史数据验证分页唯一性	数据不足两页，翻页跨不出去，缺陷触发不了	数据不足时 skip，不给假 PASS（详见 11 节）
用造数账号做报名用例	1/3 账号信用分被扣到 50，稳定 4012	用一次性账号，或按 0/59/60/100 定向造分

2.2 四条边界线的切法

边界测试最容易退化成"随便挑几个数字试一下"。本次每条线都按三层切：**合法边界内**（必须放过）、**恰好越界**（必须拦住且错误码正确）、**被拒之后系统状态是否被污染**（名额、计数、记录有没有残留）。第三层最容易被忽略，但它是"这个拦截是否真安全"的唯一证据。

维度	切入点	为什么选这些点	状态污染检查
数值	页码 0、页长 0/1/50/51、推荐 0/1/50/51、评分 0/1/5/6	两端都取，才能挡住"只校验上界"或"只校验下界"	越界后不产生新记录
时间	截止前 2 秒、截止后 1 秒、已结束	秒级差才能判断是哪一层在拦（deadline / status）	被拒后 current_people 不变
信用	0 / 59 / 60 / 100 四档	闸门在 60，必须验证 59 与 60 之间的切换，以及 0 和 100 的极端值	被拒后不占名额
权限	未登录、伪造 token、篡改签名、看他人名单、退他人报名、越权取消	纵向（未认证）与横向（他人资源）两类越权各覆盖	被拒后活动状态与人数不变

2.3 两类分页语义必须分开断言

这是本次最容易被写错的地方：**活动列表越界直接报 4001，消息接口越界静默收敛到 1~50**。两者都对，但断言方向相反。如果照抄活动列表的写法去测消息，就会把"正常收敛"报成缺陷；反过来则会把"该拦的没拦"放过去。

接口	pageSize=0	pageSize=51	实现依据
GET /api/activities	4001「每页最小 1 条」	4001「每页最多 50 条」	@Min(1) @Max(50) + 全局异常转 4001
GET /api/notifications	收敛为 1，正常返回	收敛为 50，正常返回	Service 内 Math.min(Math.max(size,1),50)
GET /api/users/{id}/evaluations	4001「每页最小 1 条」	放行（无上限约束）	只有 @Min(1)，缺上限 —— 见第 11 节

三、边界矩阵总览

维度	功能点	用例数	通过	挂账	自动化文件
数值	9	12	12	0	test_04_boundary.py
时间	3	3	3	0	test_04_boundary.py
信用	5	8	8	0	test_05_credit.py
权限	6	6	6	0	test_04_boundary.py
评价负向	5	5	2	3	test_05_credit.py
消息	6	9	9	0	test_06_message.py
合计	**34**	**43**	**40**	**3**	3 个文件

功能点数小于用例数，是因为参数化把一个功能点展开成多条：信用闸门 0/59/60/100 是 1 个功能点 4 条用例，推荐条数 0/1/50/51 同理。报告里按用例计通过数，避免"1 个功能点算 1 条"导致覆盖率虚高。

四、数值边界

编号	覆盖点	输入	期望	实测
N01	列表 pageSize 下界	0	4001	4001「每页最小 1 条」
N02	pageSize 最小合法值	1	200，size=1	size=1，records=1
N03	pageSize 上界合法值	50	200，size=50	size=50
N04	pageSize 越上界	51	4001	4001「每页最多 50 条」
N05	pageNum 下界	0	4001	4001「页码最小为 1」
N06	页码远超总页数	99999	空页，total 如实	records=[]，total 不变
N07	推荐条数双向越界	1/50/0/51	前两个合法，后两个 4001	与期望一致（4 条）
N08	评分越出 1–5	0/6/-1	4001	4001「准时评分范围为 1-5」
N09	评分恰好取两端	1 与 5	均能入库并返回 evaluationId	两条均成功

人数边界（maxPeople 1/2/20/21）在 Day3 的 TC008、TC009 已覆盖并通过，本次只在矩阵里登记、不重复执行，避免同一断言两处维护。N06 值得单独说一句：页码超出总页数时返回的是"空列表 + 真实 total"，而不是报错也不是把全部数据吐出来——这是分页接口的正确姿势，说明 MyBatis-Plus 的物理分页配置正常。

五、时间边界

报名接口的时间判定有三层：deadline（报名截止）、status（1 招募中 / 2 满员 / 3 已结束 / 4 已取消）、乐观锁扣名额。三层都可能拦住同一请求，所以

必须用"只差一秒"的场景验证是哪一层在起作用。

编号	覆盖点	造数方式	期望	实测
T01	截止前一秒	库侧 deadline 设为当前时间 +2 秒，立即报名	报名成功	成功，返回 registrationId
T02	截止后一秒	deadline 设为当前时间 - 1 秒	4013 且提示"报名已截止"	4013「报名已截止」
T03	已结束活动	status 置 3，三个时间字段一并挪到过去	4013	4013「活动状态异常或已取消」

两个坑值得记下来。第一，**不能只改 end_time**：数据库约束 chk_activity_time 要求"结束时间晚于开始时间、截止时间不晚于开始时间"，只改一个字段直接被约束挡下（errno 3819），造"已结束"必须三个时间字段一起挪。第二，**截止与状态共用错误码 4013**，只有 msg 不同（"报名已截止"和"活动状态异常或已取消"）。前端若只看 code 无法区分"再等等还能报"还是"活动没了"，建议后续拆码，本轮按现有实现断言 msg。

六、信用边界

编号	覆盖点	输入	期望	实测
C01	报名信用闸门四档	0/59/60/100	0、59 拒；60、100 放行	与期望一致（4 条）
C02	被拦截不占名额	信用 59 报名	4012，人数仍 1	4012，current_people=1
C03	加分封顶	99 分 + 一次评价	停在 100，不溢出	100，等级仍"优秀"
C04	等级阈值联动	88 分（良好）+2	90 分并升级"优秀"	90 / 优秀
C05	信用流水可追溯	60 分 + 一次评价	type=2 流水，score_after 与档案一致	一致，reason="完成活动互评"

C01 四档一起跑，是为了让结论能说"闸门在 60 这条线上切得准"——只测 59 与 60 也能过，但 0 和 100 能挡出"负数被当合法""满分被误判为异常"这类实现错误。C05 顺带发现一个 P3 观察项：99 分被评价时实际只加 1 分（封顶），但 credit_log 的 change_value 仍记 +2（理论奖励值）。靠 score_after 对账不会错，但"本次变动多少"这一列会让人以为加了 2 分，建议记实际差值。

七、权限边界

编号	覆盖点	场景	期望	实测
P01	未登录	不带 Authorization 头	HTTP 401 + code 401，无 data	一致
P02	伪造 token	三段式乱码	401	一致
P03	篡改签名	真实 token 改末段	401	一致
P04	看他人活动名单	非发起人查名单	拒绝	4040（不暴露活动存在性）
P05	退他人报名	第三者退他人 registrationId	拒绝，名额不变	4040「无权操作他人的报名记录」，人数仍 2
P06	越权取消活动	非发起人取消	4030，状态不变	一致（与 Day3 TC012 同口径）

P01 的实现值得说明：鉴权拦截器直接写 HTTP 401 并输出 {"code":401,"msg":"未登录或登录已过期"}，没有走全局异常。这意味着**前端不能只判断 HTTP 200**，必须同时校验 HTTP 状态码。P05 是横向越权里最容易被漏的一条——"退出报名"用的是 registrationId 而不是 activityId，如果服务端只按 registrationId 更新，任何人拿到别人的 ID 就能把别人踢出活动。

八、评价负向与两处缺陷

8.1 重复评价被拦（通过）

防重复有两道：应用层先查（给友好提示），数据库唯一索引 uk_eval_once(activity_id, from_user_id, to_user_id) 兜底并发。为了不只给一行"通过"，取证脚本把过程完整打出来：第一次成功加分，第二次被 4015 拒绝，并复查库表确认没有重复加分、没有重复流水。

观测点	第 1 次	第 2 次	结论
接口返回	code=0，evaluationId=56	code=4015	第二次被拦
被评价人信用分	60 → 62	仍为 62	未重复加分
evaluation 记录数	1	1	未产生重复行
credit_log 流水数	1	1	未重复记账

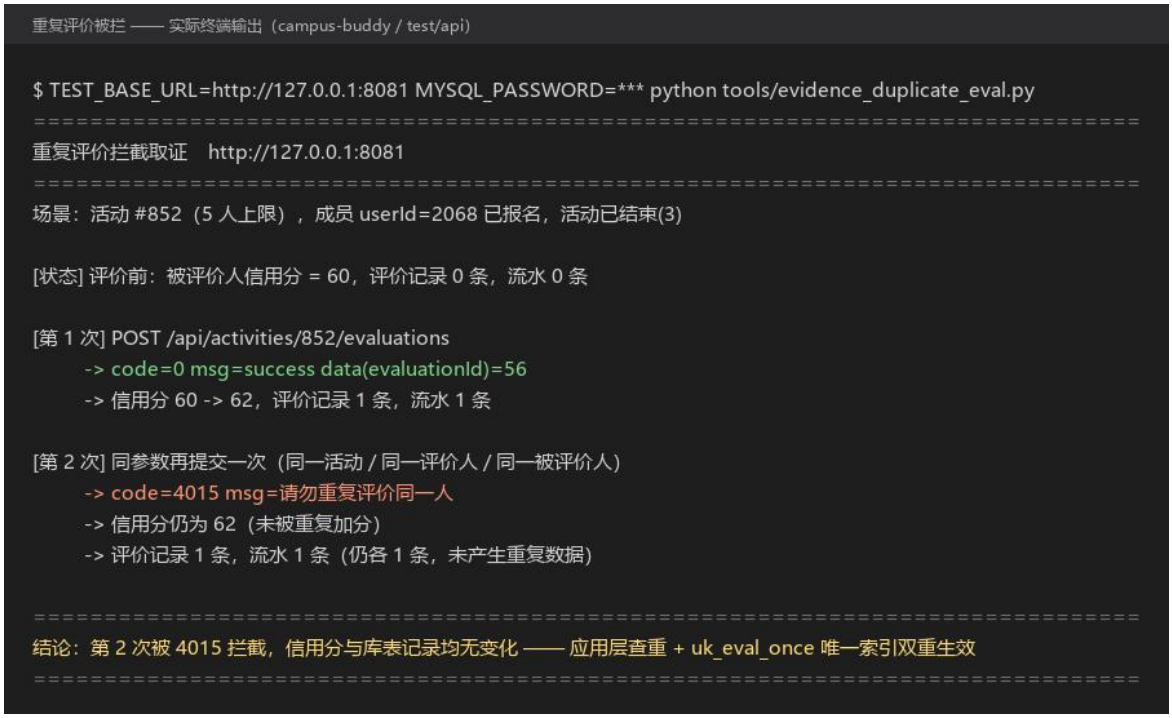


图 8-1　重复评价拦截取证（tools/evidence_duplicate_eval.py 的真实终端输出，命令与返回逐行可核对）

8.2 BUG-4：越权评价可刷分（P1）

现象：任意登录用户，对自己**没参加过的活动**、给**任意用户**提交评价，接口返回 code=0，被评价人信用分 +2 并落一条流水。活动还在招募中（status=1）时同样能评。

危害：这是可无限刷分的口子。唯一约束是三列组合 (activity_id, from_user_id, to_user_id)，换一个 activityId 就能对同一个人再评一次，每次 +2。注册即 100 分的账号看不出差别，但被打到 50 分的账号（Day3 压测里就有一批爽约扣分的）可以靠刷评价几分钟内回到 60 分以上，信用体系等于失效。

根因：EvaluationServiceImpl 的 validate() 只校验了参数完整性、自评、评分区间，缺少"评价人是否参加过该活动""活动是否已结束"两条业务前置。对应 05 清单 Day4 李佳任务①，尚未合入 master，故当前必现。

```
// 现状：validate() 之后直接查重 → insert
validate(evaluation);
// 1. 防重复评价（只有这一道业务校验）
Long existed = evaluationMapper.selectCount(...);
...
// 2. 评价入库
evaluationMapper.insert(evaluation);

// 建议补在 validate() 之后、查重之前：先验"参加过 + 已结束"
// ① registration(activity_id, from_user_id) 存在且 status in (1 已报名, 3 已签到)
// ② activity.status == 3（已结束）或 end_time 已过
// 任一不满足 → 4030 / 4013，不写库
```

用例：test_bd_e03（未参加就评价）、test_bd_e04（活动未结束就评价），均按"应该被拒绝"断言，当前 xfail 挂账，李佳合入后自动转通过。

8.3 BUG-5：被评价人不存在返回 5000（P2）

现象：toUserId 传一个不存在的用户 ID，返回 code=5000「系统内部错误」。

根因：submitEvaluation() 的顺序是"先 insert 评价 → 再查被评价人资料"。evaluation.to_user_id 上有外键 fk_eval_to_user，插入时外键就先抛 DataIntegrityViolationException，被全局兜底成 5000，代码里那句 throw NOT_FOUND("被评价人资料不存在") 永远走不到。

建议：把资料查询提到 insert 之前。顺带省一次无效插入，也让"用户不存在"这类输入错误回 4040 而不是 5000——5000 意味着"服务端崩了"，会误导前端弹"稍后重试"，而实际上是调用方传错了参数。

用例：test_bd_e05，xfail 挂账。

九、消息边界

消息模块本轮覆盖分页收敛、未读过滤与角标一致性、已读的两种入口（单条/全部）以及空数组保护。

编号	覆盖点	输入	期望	实测
M01	size 四档收敛	0/1/50/51	收敛为 1/1/50/50	回显依次 1、1、50、50
M02	page 下界	0	收敛为第 1 页	page=1，列表非空
M03	hasNext 自洽	size=1 翻首页与末页	首页按公式，末页 false	一致
M04	未读过滤与角标同源	unread=1 + unread-count	两者数字相等	相等（3 = 3）

编号	覆盖点	输入	期望	实测
M05	全部已读	PUT 不传 ids	角标归零、未读清空、消息不删	一致（总数不变）
M06	空数组标记已读	{"ids":[]}	4001，未读数不变	4001「ids 不能是空数组」

M04 是这几条里最值得保留的：列表里未读条数和角标数字如果不同源，前端红点就会撒谎——用户点开看到 3 条，红点却显示 5。M05 则验证了"已读不等于删除"，消息总数在标记前后一致，避免实现时用 delete 代替 update 把历史抹掉。

造通知时踩了一个坑，记在这里：报名后**发起人收到的是 type=9（新的报名）**，type=1 是发给报名者自己的"报名成功"。第一版按 1 去等发起人的通知，等了 20 秒超时，差点当成"异步线程池吞消息"报缺陷，翻 notification 表才看清类型。

十、未覆盖项与原因

覆盖率高不代表全测了。以下 8 项本轮明确未覆盖，写清原因，避免答辩时被问到"这个测了吗"却说不出所以然。

#	未覆盖点	原因	后续处理
1	token 真实过期（等 TTL 到期）	需等数小时，且无法控制服务端时钟	用"篡改签名"等价覆盖（P03），请林严泽在文档写明有效期
2	手机号/学号格式边界	属注册参数校验，不在 Day4 四条边界线内	交林严泽接口文档核对，如需补 6 条参数化即可
3	超长字符串（标题/描述长度上限）	API 层无 @Size，只有 DB 字段长度兜底	建议确认是否要 API 层拦截，避免 5000

#	未覆盖点	原因	后续处理
4	跨日 00:00 时区边界	需改系统时间，会影响同机其他人跑测试	Day2 基线已用 +8h 核对时区口径，不做时钟穿越
5	消息 unread × page × size 全组合	组合数收益递减	抽样覆盖 4 档 size + 未读过滤 + 角标一致性
6	人数满员瞬间（第 20 人抢占）	需压测环境	Day3 已覆盖（100/500 并发，超卖 0）
7	已取消活动上的评价交互	取消场景 Day3 已覆盖，评价校验未上线	BUG-4 修复后补"已取消活动不可评价"
8	前端侧边界（输入长度、按钮禁用态）	本次为接口层测试	交施乐山页面走查

另有一处"半覆盖"需要登记：评价列表 GET /api/users/{id}/evaluations 只有 @Min(1)，没有 @Max 上限，pageSize 传 1000 会被放行。目前评价数据量小，不构成风险，但属于"接口规范不一致"（活动列表有上限、评价列表没有），已列入明日跟进。

十一、全量回归与结论

项目	结果
执行条数	65（新增 43 + 原有 22）
通过 / 失败 / 跳过	61 / 0 / 0
预期失败（xfail）	3（BUG-4 ×2、BUG-5 ×1）
意外通过（xpass）	1（BUG-2，非修复，见下）
纯执行耗时	22.9s
新增缺陷	2（BUG-4 P1、BUG-5 P2）

关于那条"意外通过"：BUG-2（列表排序键并列导致翻页重复/漏项）今天全量回归显示通过，但它**不代表已修复**。该缺陷的触发条件是"同秒创建的多条活动

恰好跨越页边界"。实测执行该条时可招募活动 82 条、同秒并列组最多 4 条，都落在分页边界内部，翻页压根没有跨过并列组——缺陷只是没被触发。若直接写"已修复"，会掩盖 ORDER BY created_at 依然缺唯一兜底键的事实，等数据涨到几万条又会复发。

因此本轮做了两件事：一是给该用例加前置判断（可招募活动不足两页、或没有同秒并列时 skip，不给假 PASS）；二是修正测试工具的口径——原先 xfail 记成 skipped、xpass 记成 passed，日报会把"意外通过"算成真通过，现在单独统计并在报告第三节加提示。

十二、次日计划

任务（05 清单 Day5）	具体做法
高并发混合场景	三股流量打同一批活动：报名 ×N、退出 ×N、取消 ×1，统计错误率、死锁数（InnoDB 状态 + 日志 grep）、名额最终一致性
同用户并发重复报名幂等	单账号 20 线程同时报名，期望 1 成功 + 19 个 4011，库里只有 1 条报名行（唯一索引兜底）
多人同时评价的流水正确性	10 个评价人并发评价同一被评价人，核对流水条数、score_after 递增、最终封顶 100
安全用例	伪造 token（复用 P02/P03）、越权访问他人报名记录、SQL 注入尝试（title 传 ' OR 1=1 --）
跟进 BUG-4 / BUG-5	李佳合入后去掉 xfail 标记复跑，补"越权评价被拒"截图

十三、提交记录与 AI 使用记录

commit	message
2be874c	test(boundary): Day4 边界矩阵 43 条用例落地，含 xpass 口径修正 - 王嘉萍

commit	message
19b0d79	docs(test): Day4 边界矩阵清单与重复评价取证，README 排期更新 - 王嘉萍

分支 feature/wangjiaping-test，已 rebase 到 master b879eef。今日 master 前进了两条（李佳种子刷新、施乐山评价页），确认 src/main 无变化（改的是 frontend/sql/docs），测试结论不受影响。

AI 用途	具体做了什么
读源码确认边界口径	逐条核对 @Min/@Max 注解、消息分页的 clamp 实现、ErrorCode 全量码表，避免断言写在自己的想象上
先探针后断言	先跑一次性探针把"系统真实怎么反应"打出来，再据实写 43 条用例，挡掉三次断言方向写错
定位两个缺陷根因	BUG-4 定位到 validate() 缺业务前置；BUG-5 定位到"先 insert 后查资料"导致外键先炸
修正测试资产口径	conftest 区分 xfailed/xpassed、日报加 xpass 提示、分页唯一性用例加数据量前置
生成文档	边界矩阵清单、取证脚本、本报告

附录：复现方式

```
cd test/api
set TEST_BASE_URL=http://127.0.0.1:8081
set MYSQL_PASSWORD=******

:: 只跑边界矩阵（43 条）
python -m pytest tests/test_04_boundary.py tests/test_05_credit.py
tests/test_06_message.py -v

:: 全量回归并出当日报告
python scripts/run_smoke.py --day 4

:: 重复评价取证（输出可直接截图）
python tools/evidence_duplicate_eval.py
```

前置条件：后端已在 8081 运行、bugedui 库可连。信用类用例需要数据库写权限（按 0/59/60/100 定向造分）；没有 MYSQL_PASSWORD 时相关用例会 skip 而不是报错，保证接口用例集仍能在不碰库的环境里跑。