|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 测试用例/内容/前置条件 | | | 测试输入 | 期望结果 | 测试结果 |
| 用例名称：注册表单填写  测试内容：输入多种可能的情况查看表单的错误提示是否符合预期 | | | 用户名：  密码：  重复密码：  邮箱：  手机号： | 只有当符合以下条件时可以提交注册请求：  密码： 6个字符及以上，同时包括字母及数字  重复密码：与密码完全相同  邮箱：符合邮件格式  手机号：11位数字 | 符合期望 |
| 用例名称：注册表单提交  测试内容：测试表单提交后查看结果，相同的用户名与手机号不能够用来重复注册  前置条件：数据库中user表中数据为空 | | | 用户名：testuser2  密码：testuser2  重复密码：testuser2  邮箱：testuser2@233.com  手机号：  12345678902 | 注册成功，返回id给用户，  数据库中密码应为testuser2的加密版本，即 58dd024d49e1d1b83a5d307f09f32734 | 符合期望 |
| 用户名：testuser2  密码：testuser2  重复密码：testuser2  邮箱：testuser2@233.com  手机号：  12345678900 | 注册失败，用户名重复 | 符合期望 |
| 用户名：testuser3  密码：testuser2  重复密码：testuser2  邮箱：testuser2@233.com  手机号：  12345678902 | 注册失败，手机号重复 | 符合期望 |
| 用例名称：用户登录  测试内容： 包括用户被封禁，用户名不存在，密码错误，登录成功  前置条件：数据库中user表中有两个用户，第一个用户username为testuser1，password为testuser1的加密，valid为0，第二个用户username 为testuser2，password为testuser2的加密，valid为1（其余信息省略）  （包含单元测试） | | | 用户名：testuser1  密码：testuser1 | 该用户被封禁 | 符合期望 |
| 用户名：testuser2  密码：testuser1 | 密码错误 | 符合期望 |
| 用户名：testuser3  密码：testuser2 | 用户名不存在 | 符合期望 |
| 用户名：testuser2  密码：testuser2 | 登录成功 | 符合期望 |
| 用例名称：用户登录成功  测试内容：测试用户登录成功后导航栏的转变、session的保持 | | | 点击欢迎弹窗中的OK按钮 | 页面重新加载（不跳转），导航栏中的登录、注册按钮变为登出、用户中心，左侧目录选项也从登录、注册变为登出、用户中心 | 符合期望 |
| 关闭页面，重新输入网址进入 | 用户应仍在登录状态 | 符合期望 |
| 用例名称：用户登出  测试内容：测试用户登出后导航栏的转变、session的失效 | | | 点击logout按钮并确认登出 | 页面重新加载（不跳转），导航栏中的登出、用户中心按钮变为登录、注册，左侧目录选项也从登出、用户中心变为登录、注册  若仍在用户中心，用户中心将不再显示，并有提醒用户登录的提示 | 符合期望 |
| 用例名称：管理员登录  测试内容：包括登录成功，登录失败  前置条件：数据库中admin表中有一个管理员，其id为111111，密码为123  （包含单元测试） | | | id：111110  密码：123 | 登录失败 | 符合期望 |
| id：111111  密码：1234 | 登录失败 | 符合期望 |
| Id：111111  密码：123 | 登录成功 | 符合期望 |
| 用例名称：管理员登录成功  测试内容：页面跳转，session保持 | | | 点击欢迎弹窗中的OK按钮 | 页面跳转至管理员页面，路径为/admin，导航栏转变与用户同理 | 符合期望 |
| 用例名称：管理员登出  测试内容：测试管理员登出后导航栏的转变、session的失效 | | | 点击logout按钮并确认登出 | 页面重新加载（不跳转），导航栏中的登出、管理员中心按钮变为登录、注册，左侧目录选项也从登出、管理员中心变为登录、注册  若仍在管理员页面，管理员页面将不再显示，并有提醒用户登录的提示 | 符合期望 |
| 用例名称：餐厅工作人员登录  测试内容：包括登录成功，登录失败  前置条件：数据库中worker表中有一条数据，其中windowId为1，密码为123  （包含单元测试） | | | id：abc  密码：123 | 登录失败 | 符合期望 |
| id：1  密码：1234 | 登录失败 | 符合期望 |
| id：2  密码：123 | 登录失败 | 符合期望 |
| id：1  密码：123 | 登录成功，提示工作人员其所服务的窗口 | 符合期望 |
| 用例名称：餐厅工作人员登录成功  测试内容：页面跳转，session保持 | | | 点击欢迎弹窗中的OK按钮 | 跳转至工作人员界面，路径为/worker，导航栏转变与用户同理 | 符合期望 |
| 用例名称：餐厅工作人员登出  测试内容：测试餐厅工作人员登出后导航栏的转变、session的失效 | | | 点击logout按钮并确认登出 | 页面重新加载（不跳转），导航栏中的登出、工作人员中心按钮变为登录、注册，左侧目录选项也从登出、工作人员中心变为登录、注册  若仍在工作人员页面工作人员页面将不再显示，并有提醒用户登录的提示 | 符合期望 |
| 用例名称：新用户引导  测试内容：测试对新用户或还没有设置标签的用户推送设置标签的提醒 | | | 用户登录 | 页面上方出现系统推送，提醒用户设置标签，若不点任何按钮，10秒后消失 | 符合期望 |
| 用例名称：推送菜品推荐  测试内容：对设置过标签的用户推送 | | | 用户登录 | 页面上方出现系统推送，提醒用户查看今日菜品推荐，若不点任何按钮，10分钟后消失 | 符合期望 |
| 用例名称：主页广告  测试内容：测试主页的广告的定时切换与手动切换 | | | 进入首页  点击next和back按钮 | 点击next按钮查看后一张广告，  点击back按钮查看前一张广告，每过10秒广告会向后一张循环 | 符合期望 |
| 用例名称：菜单目录切换  测试内容：测试菜单按钮，主页按钮的路由跳转 | | | 分别点击主页的各个餐厅入口按钮，分别点击左侧菜单目录内的各个餐厅入口按钮 | 按钮被点击后应向对应窗口跳转 | 符合期望 |
| 用例名称：总体人流量可视化展示  测试内容：按人数、时间关系转化为折线图 | | | 点击菜单栏chart按钮进入总数据界面 | 图标展示该窗口现在时间前后半小时的人数状况，实线表示实时人数，仅提供过去半小时的，虚线表示历史/预期人数，默认展示的昨天该时间点前后半小时的人数情况。  图表每2分钟自动刷新一次。  图表下方文字将提供实时新来的人数信息，最近一次刷新时间信息，估计等待时间信息 | 符合期望 |
| 用例名称：切换数据展示模式  测试内容：切换数据展示模式 | | | 点击切换chart mode->total | 图表将从时间点人数，转化为累计人数 | 符合期望 |
| 点击切换history mode->predict | 图表的虚线将从展示昨日数据变为展示预期数据（运用算法实现） | 符合期望 |
| 点击切换 select restaurant | Select window的待选项将根据选择的窗口改变，默认选择第一个窗口，并显示改窗口的可视化数据 | 符合期望 |
| 点击切换Select window | 根据选择的窗口改变可视化显示的数据 | 符合期望 |
| 用例名称：比较人流量可视化展示  测试内容：比较两个窗口的可视化数据 | | | 选择需要比较的餐厅、窗口，点击compare按钮查看结果 | 根据选择的窗口显示可视化数据 | 符合期望 |
| 用例名称：窗口内可视化数据展示 | | | 进入餐厅 | 窗口默认为All，不展示可视化数据 | 符合期望 |
| 进入餐厅选择窗口 | 根据窗口展示可视化数据，切换数据展示模式同上 | 符合期望 |
| 用例名称：导航栏按关键字搜索菜品  测试内容：输入不同关键字查看菜品搜索结果  前置条件：  food   |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | | foodId | foodName | tips | Window\_id | Price | like | | 1 | 烤鸭 | meat | 1 | 10 | 1 | | 2 | 排骨 | meat | 2 | 10 | 2 | | 3 | 土豆 | vegetable | 3 | 10 | 3 |   todayfood   |  |  | | --- | --- | | foodID | time | | 1 | 1 | | 2 | 1 | | 3 | 2 |   tag   |  |  | | --- | --- | | tagID | tagName | | 1 | 酸 | | 2 | 甜 | | 3 | 辣 |   foodtag   |  |  | | --- | --- | | tagID | foodID | | 1 | 1 | | 1 | 2 | | 2 | 3 | | 3 | 3 |   window   |  |  |  | | --- | --- | --- | | window\_id | restaurant | windowName | | 1 | one | 川 | | 2 | one | 湘 | | 3 | two | 鲁 | | | | 在搜索框中输入烤鸭并点击查询，（假设当前时间在time为1的范围内） | 烤鸭  > restaurant: one  > window: 川  name: 烤鸭- like: 1 - price: 10￥- tips: meat | 符合期望 |
| 在搜索框中输入土豆并点击查询，（假设当前时间在time为1的范围内） | No result | 符合期望 |
| 在搜索框中输入meat并点击查询，（假设当前时间在time为1的范围内） | 烤鸭  > restaurant: one  > window: 川  name: 烤鸭- like: 1 - price: 10￥- tips: meat  -----------------  排骨  > restaurant: one  > window: 湘  name: 排骨- like: 2 - price: 10￥- tips: meat | 符合期望 |
| 在搜索框中输入酸并点击查询，（假设当前时间在time为1的范围内） | 烤鸭  > restaurant: one  > window: 川  name: 烤鸭- like: 1 - price: 10￥- tips: meat  -----------------  排骨  > restaurant: one  > window: 湘  name: 排骨- like: 2 - price: 10￥- tips: meat | 符合期望 |
| 在搜索框中输入辣并点击查询，（假设当前时间在time为1的范围内） | No result | 符合期望 |
| 用例名称：窗口跳转  测试内容：点击搜索结果中的窗口名直接跳转至窗口  前置条件：搜索结果为  烤鸭  > restaurant: one  > window: 川  name: 烤鸭- like: 1 - price: 10￥- tips: meat  -----------------  排骨  > restaurant: one  > window: 湘  name: 排骨- like: 2 - price: 10￥- tips: meat | | | 点击搜索结果中的窗口名 | 跳转至餐厅one的川窗口 | 符合期望 |
| 点击搜索结果中的窗口名 | 跳转至餐厅one的湘窗口 | 符合期望 |
|  | | |  |  |  |
| 用例名称：用户查看并修改个人信息（密码需通过其他方式单独修改）  测试内容：   1. 输入多种可能的情况查看表单的错误提示是否符合预期 2. 密码不能直接修改，需点击单独的修改框 3. 密码显示加密版本后的密码格式 4. 只有当符合以下条件时可以提交修改请求：   邮箱：符合邮件格式  手机号：11位数字  头像：小于2MB且为JPG格式  前置条件：用户登录成功 | | | 用户根据需要，修改头像、用户名、邮箱、电话 | 1. 个人信息正确显示 2. 密码框不可直接修改 | 符合期望 |
| 用例名称：用户修改个人信息表单提交  测试内容：测试表单提交后查看结果，相同的用户名与手机号不能够用来重复注册  前置条件：   1. 用户登录成功 2. 数据库中user表中有另外两条数据，一条的用户名为testuser1，手机号为12345678900；另一条的用户名为testuser3   （包含单元测试） | | | 用户名：testuser2  邮箱：testuser2@233.com  手机号：  12345678902 | 修改成功 | 符合期望 |
| 手机号：  12345678900 | 修改失败，手机号重复 | 符合期望 |
| 用户名：testuser3 | 修改失败，用户名重复 | 符合期望 |
| 用例名称：用户修改密码  测试内容：  测试表单能否正确检测并判断用户密码修改操作  成功修改密码的条件：   1. 旧密码需正确，否则修改失败   2.新密码： 6个字符及以上，同时包括字母及数字  3.重复密码：与新密码完全相同  前置条件：   1. 用户登录成功 2. 数据库中user表中有一条数据，密码为zbn456的加密版   （包含单元测试） | | | 点击修改密码，弹出弹框，输入旧密码、新密码、重复新密码 |  | 符合期望 |
| 旧密码：zbn456  新密码：llh123  重复输入新密码：llh123 | 修改失败，旧密码不正确 |
| 旧密码：zbn456  新密码：llh123  重复输入新密码：llh12 | 修改失败，重复输入不一致 |
| 用例名称：用户设置个性化标签  测试内容：   1. 用户进入界面即可显示曾选择过的标签 2. 用户可一次自定义一个标签 3. 用户可一次选择不同种类的多个标签 4. 用户自定义的标签提交后，新标签即可马上出现在标签选择界面   前置条件：用户登录成功  （包含单元测试） | | | 1. 点击用户个人中心-个性化设置，进行个性化标签选择 2. 标签选择完成后，点击提交 | 1. 界面成功显示曾经选择的标签 2. 自定义标签提交后，新标签显示在标签选择界面 3. 数据库正确记录用户选择的标签和自定义标签 | 符合期望 |
| 用例名称：用户收藏  测试内容：   1. 用户进入界面即可显示所有收藏的菜品 2. 菜品显示地理位置（食堂、楼层）、菜品图片、菜品名、价格等信息 3. 若菜品今日有售，则会出现售卖信息提示 4. 用户可取消收藏 5. 分页设置，一页显示三个菜品   前置条件：用户登录成功  （包含单元测试） | | | 1. 点击用户个人中心-我的收藏 2. 点击红❤取消收藏，❤变为灰色；若再次点击灰❤，则变为红❤，代表重新收藏成功 3. 点击下一页，查看第二页的收藏菜品 | 1. 页面显示所有数据库中用户收藏的菜品 2. 取消收藏后，数据库相应删除相关数据 3. 分页显示正确，菜品无重复显示 | 符合期望 |
| 用例名称：用户历史记录浏览  测试内容：   1. 浏览量由高至低显示 2. 用户可选择删除某条历史浏览记录或者删除全部历史浏览记录   前置条件：用户登录成功 | | | 1. 点击用户个人中心-我的收藏 2. 点击选择框，选择删除 | 1. 浏览量按预期排序显示 2. 删除成功 | 符合期望 |
| 用例名称：用户个性化推荐  测试内容：   1. 根据设计算法进行个性化推荐 2. 只推荐今日此时的菜品 3. 将推荐菜品分组，根据组别分批显示 4. 若不喜欢推荐菜品，可删除菜品，为后续的推荐提供信息   前置条件：用户登录成功 | | | 1. 点击用户个人中心-个性化推荐 2. （可选）删除不喜欢的菜品 3. 点击换一批，显示下一批推荐菜品 | 1. 算法执行正确，无报错 2. 后台数据库成功记录用户不喜欢的菜品的标签 | 符合期望 |
| 用例名称：首次登陆的用户提示标签选择  测试内容:   1. 一次显示一个类别的标签 2. 若用户选择某个标签，则出现成功选择的提示 3. 点击下一页/上一页，可随意重复修改标签 4. 完成标签选择后系统根据用户的偏好为用户推荐当日当时菜品 5. 将推荐菜品分组，根据组别分批显示   前置条件：用户登录成功 | | | 1. 点击设置标签 2. 选择“口味”标签喜好 3. 点击下一步 4. 跳转至“国家”标签喜好 5. 点击下一步 6. 跳转至“忌口”标签设置 7. 点击下一步 8. 跳转至“食材”标签喜好 9. 点击上一步，可修改“忌口”标签 10. 完成标签选择 11. 查看系统推荐的菜品 12. 点击“换一批”，查看下一批推荐菜品 | 1. 后台数据库记录用户标签无误 2. 个性化推荐算法无误，无报错 | 符合期望 |
|  | | |  |  |  |
|  | | |  |  |  |
|  | | |  |  |  |
| 用例名称：进入餐厅页面  测试内容：从Home页跳转到对应的餐厅页面 | | | 点击Home页的餐厅按钮  （六个都测） | 跳转到对应的餐厅页面 | 符合期望 |
| 用例名称：进入餐厅页面  测试内容：通过侧边栏跳转到餐厅页面 | | | 在任意界面点击侧边栏的餐厅按钮  （六个都测） | 跳转到对应的餐厅页面 | 符合期望 |
| 用例名称：餐厅楼层显示  测试内容：餐厅页面显示该餐厅所有的楼层  测试环境：数据库中有四个窗口：   |  |  |  | | --- | --- | --- | | restarurant | windowName | floor | | one | Test1 | 1 | | one | Test2 | 2 | | two | Test3 | 3 | | two | Test4 | 4 |   （包含单元测试） | | | 进入one餐厅页面； | 显示楼层：  全部，1楼，2楼 | 符合期望 |
| 进入two餐厅页面 | 显示楼层：  全部，3楼，4楼 | 符合期望 |
| 用例名称：窗口显示  测试内容：餐厅页面显示某楼层的对应窗口  测试环境：数据库中有六个窗口   |  |  |  | | --- | --- | --- | | restarurant | windowName | floor | | one | Test1 | 1 | | one | Test2 | 1 | | one | Test1 | 2 | | one | Test3 | 2 | | two | Test1 | 1 | | two | Test4 | 2 |   （包含单元测试） | | | 进入餐厅one，楼层：全部  点击窗口菜单 | 显示窗口：  所有窗口，Test1窗口，Test2窗口，Test1窗口，Test3窗口 | 符合期望 |
| 进入餐厅one，楼层：1楼  点击窗口菜单 | 显示窗口：  所有窗口，Test1窗口，Test2窗口 | 符合期望 |
| 进入餐厅one，楼层：2楼  点击窗口菜单 | 显示窗口：  所有窗口，Test1窗口，Test3窗口 | 符合期望 |
| 进入餐厅two，楼层：全部  点击窗口菜单 | 显示窗口：  所有窗口，Test1窗口，Test4窗口 | 符合期望 |
| 进入餐厅two，楼层：1楼  点击窗口菜单 | 显示窗口：  所有窗口，Test1窗口 | 符合期望 |
| 用例名称：菜品展示  测试内容：只展示todayfood中的food  测试环境：  food：   |  |  |  | | --- | --- | --- | | foodId | foodName | windowId | | 1 | test1 | 1 | | 2 | test2 | 1 |   其他不重要的属性省略  Todayfood:   |  |  | | --- | --- | | foodid | windowId | | 1 | 1 |   其他不重要的属性省略  windowId为1 的window位于one餐厅一楼 | | | 进入one餐厅，楼层：全部，窗口：所有窗口 | 显示菜品：  只显示foodName为test1的菜品  不含test2 | 符合期望 |
| 用例名称：菜品展示  测试内容：展示对应的窗口的todayfood  测试环境：  food：   |  |  |  | | --- | --- | --- | | foodId | foodName | windowId | | 1 | food1 | 1 | | 2 | food1-1 | 1 | | 3 | food2 | 2 | | 4 | food3 | 3 | | 5 | food4 | 4 |   以上food在todayfood中都有  window：   |  |  |  |  | | --- | --- | --- | --- | | windowId | restarurant | windowName | floor | | 1 | one | win1 | 1 | | 2 | one | win2 | 1 | | 3 | one | win3 | 2 | | 4 | two | win4 | 1 |   一餐一楼有两个窗口win1,win2，二楼一个win3，二餐一楼一个win4，win1有两个菜品food1,food1-1;win2有food2,win3有food3,win4有food4  （包含单元测试） | | | 进入餐厅：one  楼层：全部  窗口：所有 | 显示：  food1,food1-1,food2,food3 | 符合期望 |
| 进入餐厅：one  楼层：一楼  窗口：全部 | 显示：  food1,food1-1,food2 | 符合期望 |
| 进入餐厅：one  楼层：一楼  窗口：win1 | 显示  food1,food1-1 | 符合期望 |
| 进入餐厅：one  楼层：全部  窗口：win3 | 显示：  food3 | 符合期望 |
| 进入餐厅：two  楼层：全部  窗口：所有 | 显示：  food4 | 符合期望 |
| 用例名称：记录浏览历史  测试内容：点击某个窗口，数据库能够记载  前置条件：用户已登陆  测试环境：  userId: 1  userId: 2  window：   |  |  |  |  |  | | --- | --- | --- | --- | --- | | windowId | restarurant | windowName | floor | | | 1 | one | win1 | 1 | | 2 | one | win2 | 1 | | 3 | one | win3 | 2 | | 4 | two | win4 | 1 |   viewHistory为空  （包含单元测试） | | | 未登录点击窗口 | 数据库不做改变 | 符合期望 |
| 1. userId为1的用户登陆后点击进入win1 2. 该用户点击win2 3. userId为1的用户登陆后再次点击win1 4. 用户退出登录，点击win2 5. userId为2的用户登陆，点击win1 | 1.viewhistory变为：   |  |  |  | | --- | --- | --- | | userId | windowId | count | | 1 | 1 | 1 |   2.viewhistory变为   |  |  |  | | --- | --- | --- | | userId | windowId | count | | 1 | 1 | 1 | | 1 | 2 | 1 |   3.viewhistory变为   |  |  |  | | --- | --- | --- | | userId | windowId | count | | 1 | 1 | 2 | | 1 | 2 | 1 |   4.数据库不做改变  5. 数据库中viewhistory中变为   |  |  |  | | --- | --- | --- | | userId | windowId | count | | 1 | 1 | 2 | | 1 | 2 | 1 | | 2 | 1 | 1 | | 符合期望 |
| 用例名称：未登录收藏  测试内容：  1. 未登录用户的所有菜品收藏状态为未收藏  2.用户未登录点击某个菜品的收藏按钮，提示用户登录  前置条件：用户未登录 | | | 1．用户未登录，浏览菜品  2. 点击收藏按钮 | 1. 菜品的收藏按钮全是灰色 2. 点击后弹窗提示用户登陆，❤还是灰色 | 符合期望 |
| 用例名称：已登录收藏  测试内容：   1. 菜品的收藏量显示 2. 菜品的收藏状态 3. 收藏菜品收藏量和收藏状态变化 4. 取消收藏收藏量和收藏状态变化   前置条件：用户已登录  测试环境：  user:两个user     |  |  | | --- | --- | | userId | userName | | 1 | user1 | | 2 | user2 |   food:   |  |  |  | | --- | --- | --- | | foodId | foodName | likes | | 1 | food1 | 0 | | 2 | food2 | 1 |   userlikefood:   |  |  | | --- | --- | | userId | foodId | | 2 | 2 | | | | 1. 登陆user1,浏览菜品。 2. 点击food1的❤ 3. logout user1，login user2，浏览菜品。 4. 点击food1的红色的❤ 5. 点击food2的红色的❤ | 1. food1的❤为灰色，收藏量为0，food2的❤也为灰色，收藏量为1 2. food1的❤变为红色，收藏量变为1，数据库中userlikefood表  |  |  | | --- | --- | | userId | foodId | | 2 | 2 | | 1 | 1 |   food中food1的likes变为1   1. food1的收藏量为1，❤为灰色，food2的收藏量为1，❤为红色 2. food1的❤变红，收藏量变为2，数据库中userlikefood表增加一条userId=2,foodId=1 ，food中food1的likes变为2 3. food2的红❤变灰，收藏量变为0，数据库中userlikefood表userId=2,foodId=2被删除，food中food2的likes变为0 |  |
| 用例名称：评论展示  测试内容：   1. 评论内容显示在对应的某一窗口下 2. valid为0的封禁评论不显示   前置条件：切换到该窗口  测试环境：  win1窗口下有两条评论，Id为1和2，1合法，2封禁  win2窗口有一条Id为3的合法评论  win3窗口没有评论  comment:   |  |  |  |  | | --- | --- | --- | --- | | commentId | windowId | commentContent | valid | | 1 | 1 | com1 | 1 | | 2 | 1 | com2 | 0 | | 3 | 2 | com3 | 1 | | | | 1. 进入win1窗口，进入评论区 2. 进入win2窗口，进入评论区 3. 进入win3窗口，进入评论区 | 1. 只能看到内容为com1的评论 2. 只能看到内容为com3的评论 3. 没有看到任何评论 | 符合期望 |
| 用例名称：未登录评论  测试内容：  未登录用户发表评论提示登录  前置条件：用户未登录 | | | 未登录状态点击发表评论按钮 | 1. 不弹出编辑框，弹出提示登录框 | 符合期望 |
| 用例名称：发表评论  测试内容：   1. 评论编辑框的弹出与退出 2. 评论区增加刚才发表的评论   前置条件：用户已登录  测试环境：user1登陆 | | | 1. 点击发表评论按钮 2. 点击cancel 3. 点击发表评论按钮 4. 输入“1”，点击submit 5. 切换user2，前往评论区 | 1. 弹出编辑框 2. 编辑框退出，评论区不变 3. 弹出编辑框 4. 编辑框退出，评论区增加一条发表人为user1，内容为1 ，时间为提交时的时刻的评论 5. 可以看到评论区增加了user1刚发的那条评论。 | 符合期望 |
| 用例名称：删除评论  测试内容：   1. 删除按钮只显示在自己的评论上 2. 在评论区删除自己的评论   前置条件：用户已登录  测试环境：  user1,win1  win1下有三条评论，com1来自user1，com2来自user2，com3来自user3 | | | 1. 点击进入win1的评论区 2. 点击com1的删除按钮 3. 切换用户user3，进入win1的评论区 | 1. 只有自己的com1上有删除按钮，com2和com3都没有 2. 评论区com1那条评论被删除 3. 评论区中没有com1的评论了，com2没有删除按钮，com3有删除按钮 | 符合期望 |
|  | | |  |  |  |
|  | | |  |  |  |
|  | | |  |  |  |
|  | | |  |  |  |
|  | | |  |  |  |
|  | | |  |  |  |
|  | | |  |  |  |
|  | | |  |  |  |
|  | | |  |  |  |
|  | | |  |  |  |
|  | | |  |  |  |
|  | | |  |  |  |
|  | | |  |  |  |
|  | | |  |  |  |
|  | | |  |  |  |
|  | | |  |  |  |
|  | | |  |  |  |
|  | | |  |  |  |
|  | | |  |  |  |
|  | | |  |  |  |
|  | | |  |  |  |
|  | | |  |  |  |
|  | | |  |  |  |
|  | | |  |  |  |
|  | | |  |  |  |
|  | | |  |  |  |
|  | | |  |  |  |
|  | |  | |  |  |
|  |  | | |  |  |
|  |  | | |  |  |
|  |  | | |  |  |
|  |  | | |  |  |
|  |  | | |  |  |
|  |  | | |  |  |
|  |  | | |  |  |
|  |  | | |  |  |
|  |  | | |  |  |
|  |  | | |  |  |
|  |  | | |  |  |
|  |  | | |  |  |
|  |  | | |  |  |
|  |  | | |  |  |
|  |  | | |  |  |
|  |  | | |  |  |
|  |  | | |  |  |
|  |  | | |  |  |
|  |  | | |  |  |
|  |  | | |  |  |
|  |  | | |  |  |
|  |  | | |  |  |
|  |  | | |  |  |
|  |  | | |  |  |
|  |  | | |  |  |
|  |  | | |  |  |
|  |  | | |  |  |
|  |  | | |  |  |
|  |  | | |  |  |
|  |  | | |  |  |
|  |  | | |  |  |
|  |  | | |  |  |
|  |  | | |  |  |
|  |  | | |  |  |
|  |  | | |  |  |