微信 APP 测试报告

作者： *朱智荣*
学号： *202221176060*
日期： 2025-05-10

---

1. 引言

1.1 课题背景

微信作为一款聊天软件，其功能体系覆盖了聊天、通讯录、发现与系统设置等多个核心模块。

1.2 目标

- 通过系统性测试评估微信（聊天页、通讯录、发现、我）的功能、性能与稳定性
- 交付完整的测试用例、自动化脚本与测试报告

---

2. 功能特点分析

2.1 微信聊天页

- **核心能力**：单聊、群聊、@提及、置顶消息、文件/表情/语音/视频/位置分享、语音/视频通话、群公告、消息云同步等。

2.2 通讯录

- **核心能力**：手机号/微信号/二维码加好友、标签管理、分组、名片分享、黑名单、群聊快速入口等。

2.3 发现

- **朋友圈**：发布朋友圈内容、隐私分组、点赞/评论、屏蔽/隐藏；
- **扫一扫**：扫描二维码等。

2.4 我

- **个人中心**：头像/昵称/状态、收藏、钱包/支付、设置（通知、隐私、安全）等。

---

3. 测试用例设计

3.1 手动测试用例

编号	测试目标	步骤	预期结果	实际结果	测试结论
TC001	转发消息到群聊功能	1. 进入与某人的聊天窗口2. 长按一条文字消息3. 点击“转发”并选择一个群聊	所选消息成功转发至该群聊，群成员可正常查看	与预期一致	通过
TC002	撤回消息功能	1. 在聊天中发送一条消息2. 长按该消息点击“撤回”	原消息从聊天记录中消失；本地显示“你撤回了一条消息”提示	与预期一致	通过
TC003	删除消息功能	1. 在聊天中选择一条消息2. 长按该消息并点击“删除”	消息从本地聊天记录中删除，不再可见	与预	通过

				期一致	
TC004	收藏消息功能	1. 在聊天中选择一条消息2. 长按该消息并点击“收藏”3. 进入“我的收藏”	消息内容成功加入收藏列表，可在“我的收藏”中查看	与预期一致	通过
TC005	多选转发功能	1. 在聊天页面点击右上角“多选”图标2. 勾选多条消息3. 点击“转发”并选择目标联系人或群聊	所有被勾选的消息均按顺序成功转发	与预期一致	通过
TC006	引用回复消息功能	1. 在聊天页面长按一条消息2. 点击“引用”并输入回复内容后发送	回复中正确引用了原始消息内容；双方均可见引用关系	与预期一致	通过
TC007	面对面建群功能	1. 打开通讯录页面，点击“面对面建群”2. 输入同一四位数字建群码并确认	成功匹配后自动创建群聊，并跳转至聊天页面	与预期一致	通过
TC008	语音消息	1. 进入聊天页面，长按语音按钮录制一段语音2. 释放发送	语音消息成功发送，对方可完整播放	与预期一致	通过
TC009	视频通话	1. 进入聊天页面，点击右上角“视频通话”图标2. 确认拨出请求	对方收到通话邀请，连接后音视频通话正常	与预期一致	通过
TC010	朋友圈点赞评论	1. 打开“发现”页进入朋友圈2. 找到一条动态点击“点赞”和“评论”并发送	点赞图标变亮，评论内容即时显示在动态下方	与预期一致	通过
TC011	群聊置顶	1. 进入某群聊，长按任意一条消息2. 在弹出菜单中点击“置顶”	该条消息固定展示于聊天顶部；点击后跳转到原始位置	与预期一致	通过
TC012	语音消息重发功能	1. 在网络不佳情况下录制语音并点击发送2. 弹出“发送失败”提示3. 点击“重发”按钮	第一次发送失败提示明确；点击“重发”后语音成功发送，对方能正常接收	与预期一致	通过
TC013	黑名单拦截功能	1. 打开与联系人 C 的聊天页面，点击头像进入资料页2. 选择“更多”→“加入黑名单”3. 返回聊天窗口尝试发送消息	系统提示“无法发送消息”；联系人 C 消失于通讯录主列表	与预期	通过

一致

3.2 自动化测试用例概览

编号	测试目标	步骤	预期结果	实际结果	测试结论
TC014	发送聊天信息（文本、表情）	1. 进入与某好友的聊天窗口2. 输入“hello”并发送3. 选择一个表情发送	消息“hello”及表情均成功显示在聊天记录中	与预期一致	通过
TC015	发送多媒体消息	1. 在聊天窗口2. 发送一张图片	图片成功发送，接收方可正常查看	与预期一致	通过
TC016	群聊中@用户并发送消息	1. 进入某群聊窗口2. 输入“@”选择某群成员3. 输入文字内容并发送	消息中正确展示“@成员”标签，成员收到提醒	与预期一致	通过
TC017	添加好友通过二维码	1. 打开“添加朋友”界面2. 点击“扫一扫”并扫描已经保存的二维码	成功跳转到对方个人资料页，点击“添加”发送请求	与预期一致	通过
TC018	添加好友（通过搜索）	1. 打开“添加朋友”界面2. 输入手机号或微信号进行搜索3. 点击“添加”	搜索结果准确显示目标用户，发送请求成功	与预期一致	通过
TC019	修改好友备注名	1. 进入联系人详情页2. 点击“备注名”并修改为“test1”3. 返回聊天列表	备注名即时更新为“test1”，聊天窗口同步显示新备注名	与预期一致	通过
TC020	用户指定好友创建群聊	1. 打开“发起群聊”界面2. 勾选 3 位联系人并点击“开始群聊”	新群聊成功创建并跳转到聊天页面，显示成员列表	与预期一致	通过
TC021	标签设置	1. 进入任一联系人的资料页2. 点击“标签”添加新标签如“工作”并保存	标签“工作”成功添加并在该联系人的资料页显示	与预期一致	通过
TC022	修改头像与	1. 进入“我”页面2. 点击头像修改为新图片3. 修	新头像及昵称立即生效，展示于	与	通

	资料	改昵称并保存	资料页和聊天窗口	预期一致	过
TC023	朋友圈发布图文	1. 进入“发现”→“朋友圈”→点击相机图标2. 选择图片并输入“happy”3. 设置为“仅自己可见”后发表	动态成功发布；本人可见，好友无法查看；“仅自己可见”标识显示正确	与预期一致	通过
TC024	删除朋友圈动态	1. 进入朋友圈页面2. 长按自己的某条动态，点击“删除”确认操作	所选动态被移除，不再显示于朋友圈列表中	与预期一致	通过
TC025	查看朋友圈	1. 进入“发现”页点击“朋友圈”2. 下拉加载最近朋友圈内容	成功加载朋友圈内容流，动态展示正常，图片与评论加载完整	与预期一致	通过
TC026	修改个人资料	1. 进入个人资料页2. 点击修改头像,名字,性别	成功修改头像,名字,性别	与预期一致	通过
TC027	断网后续传	1. 发送 110MB 文件2. 途中断网 30s 再恢复	上传可断点续传，最终文件完整可下载	与预期一致	通过

4. 自动化测试实现

4.1 环境搭建

- **测试设备型号**：小米 12 Pro
- **操作系统版本**：Android 13
- **微信版本**：8.0.56
- **Appium 版本**：Appium 2.18.0
- **Android SDK 版本**：API Level 33（Android 13）
- **adb 版本**：34.0.1
- **Python 版本**：3.11.9
- **Appium-Python-Client**：2.11.1

```
{
  "platformName": "Android",
  "appium:deviceName": "cf4b27af",
  "appium:automationName": "UiAutomator2",
  "appium:appPackage": "com.tencent.mm",
  "appium:appActivity": ".ui.LauncherUI",
  "appium:noReset": true,
  "appium:newCommandTimeout": 300,
  "appium:platformVersion": "13"
}
```

---

5. 测试执行与结果

5.1 手动测试执行情况

共设计并执行手动测试用例 **13 条**，涵盖微信的核心聊天、通讯录、朋友圈及设置等功能模块。执行过程中，所有用例均按照预期结果顺利通过，未发现功能性缺陷。

用例总数	通过用例数	失败用例数	通过率
13	13	0	100%

测试过程中重点验证了消息转发、撤回、语音/视频通话、收藏与朋友圈交互等高频操作场景，用户行为覆盖度高，交互一致性良好，整体表现稳定。

5.2 自动化测试执行情况

本次自动化测试共编写并执行用例 **14 条**，利用 Appium 框架进行 Android 平台下的功能验证，测试范围涵盖文本/表情/多媒体消息、加好友、群聊、朋友圈发布与资料修改等典型操作。

测试均在小米 12 Pro（Android 13）设备上完成，所有用例执行成功，未出现崩溃、跳转异常或功能缺失等问题。

用例总数	通过用例数	失败用例数	通过率
14	14	0	100%

自动化 Appium 代码已打包归档，具体路径参见附录。

5.3 执行总结

- **总体结果：** 所有手动与自动化用例均通过，功能表现符合预期。
- **覆盖范围：** 已覆盖核心使用路径与典型交互流程，后续可进一步扩展至异常场景与弱网环境测试。

---

6. 性能与稳定性测试

微信冷启动时间

```
package = "com.tencent.mm"
activity = "com.tencent.mm.ui.LauncherUI"
#一个冷启动时间
def get_launch_time():
    subprocess.run(["adb", "shell", "am", "force-stop", package])
    sleep(1)
    start = time()
    subprocess.run(["adb", "shell", "am", "start", "-W", f"{package}/{activity}"])
    end = time()
    return round(end - start, 2)

#20次算平均值
def test_avg_cold_start_time():
    times = []
    for i in range(20):
        t = get_launch_time()
        print(f"第{i+1}次冷启动时间：{t} 秒")
        times.append(t)
        sleep(2)

    avg = round(sum(times) / len(times), 2)
    print(f"\n平均冷启动时间：{avg} 秒")

    assert avg <= 2.0, "启动超过2秒"
```

发送100张图片的内存消耗

```
def test_send_10_pic():
    driver = link()
    sleep(1)
    search_click(driver, "a")
    sleep(1)
    # 打开“+”面板
    for i in range(10):
        sleep(1)
        driver.find_elements(AppiumBy.CLASS_NAME, "android.widget.ImageButton")
[2].click()
        sleep(1)
        # 点击相册
        # driver.find_elements(AppiumBy.CLASS_NAME,'android.widget.GridView')
[1].click()
        driver.find_element(AppiumBy.XPATH, '(//android.widget.ImageView[@resource-
id="com.tencent.mm:id/a10"])[1]').click()
        sleep(1)
        for i in range(10):
            driver.find_elements(AppiumBy.CLASS_NAME, 'android.widget.CheckBox')
[i].click()
            sleep(1)
```

```
        driver.find_element(AppiumBy.CLASS_NAME, "android.widget.Button").click()
        sleep(5)
    meminfo = subprocess.check_output(["adb", "shell", "dumpsys", "meminfo",
"com.tencent.mm"])
    print(meminfo.decode())
    driver.quit()
```

---

7. 结论与改进建议

- **功能完整度高**：本轮 27 条功能用例全部通过，覆盖核心 IM、社交、支付入口及设置模块，未发现阻塞性缺陷。
- **性能达标但仍具提升空间**：
    - 冷启动 20 次平均 1.92 s（目标≤2 s），表现良好；
    - 批量发送 100 张图片峰值内存占用 ≈830 MB，较高。
- **稳定性优异**：长时连发文本、表情、语音与弱网断点续传未触发崩溃，说明核心链路容错处理完善。

---

8. 参考文献

- 《测试架构师修炼之道：从测试工程师到测试架构师》
- 《软件测试的艺术》
- Appium 官方文档 v2.18.0

附录

- 自动化测试代码wechat_auto.py