**app功能测试**

## 1.运行部分

1）APP安装完成后的试运行，可正常打开软件。  
2）APP打开测试，是否有加载状态进度提示。  
3）APP打开速度测试，速度是否可观。  
4）APP页面间的切换是否流畅，逻辑是否正确  
5）注册  
--同表单编辑页面  
--用户名密码长度  
--注册后的提示页面  
--前台注册页面和后台的管理页面数据是否一致  
--注册后，在后台管理中页面提示  
用户注册功能的测试:检验未注册和已注册用户是否能成功注册,以及注册信息是否完整。  
安装及卸载：验证App是否能正确安装、运行、卸载以及操作过程和操作前后对系统资源的使用情况。

## 2.登录部分

--使用合法的用户登录系统。  
--系统是否允许多次非法的登陆，是否有次数限制。  
--使用已经登陆的账号登陆系统是否正确处理。  
--使用禁用的账号登陆系统是否正确处理。  
--用户名、口令（密码）错误或漏填时能否登陆。  
--删除或修改后的用户，原用户登陆。  
--不输入用户口令和用户、重复点（确定或取消按钮）是否允许登陆。  
--登陆后，页面中登陆信息。  
--页面中有注销按钮。  
--登陆超时的处理。

## 3.注销部分

--注销原模块，新的模块系统能否正确处理。  
--终止注销能否返回原模块，原用户。  
--注销原用户，新用户系统能否正确处理。  
--使用错误的账号、口令、无权限的被禁用的账号进行注销  
退出功能要注意，安卓手机要注意双击返回键是否能退出，IOS系统的app一般有退出按钮。

## 4.切换部分

1. APP切换到后台，再回到APP，检查是否停留在上一次操作界面。
2. APP切换到后台，再回到APP，检查功能及应用状态是否正常，IOS4和IOS5的版本的处理机制有的不一样。
3. APP切换到后台，再回到前台时，注意程序是否崩溃，功能状态是否正常，尤其是对于从后台切换回前台数据有自动更新的时候。
4. 手机锁屏解屏后进入APP注意是否会崩溃，功能状态是否正常，尤其是对于从后台切换回前台数据有自动更新的时候。
5. 当APP使用过程中有电话进来中断后再切换到APP，功能状态是否正常
6. 当杀掉APP进程后，再开启APP，APP能否正常启动。
7. 出现必须处理的提示框后，切换到后台，再切换回来，检查提示框是否还存在，有时候会出现应用自动跳过提示框的缺陷。
8. 对于有数据交换的页面，每个页面都必需要进行前后台切换、锁屏的测试，这种页面最容易出现崩溃。

### 状态保存检测

1）是否确认某些操作需要保存上次操作状态(例如点击弹出图片提示)  
2）关闭页面再打开,是否需要保存上次状态  
3）切换页面再切回来,是否需要保存上次状态  
4）在当前页面来交叉时，交叉结束后是否保  
5）在当前界面 页面设置保存操作，通过页面  
6）切换再切回，是否保存了设置（例如：修改昵称，资料，电话）

## 5.免登陆部分

很多应用提供免登录功能，当应用开启时自动以上一次登录的用户身份来使用APP.

1. APP有免登录功能时，需要考虑IOS版本差异。
2. 考虑无网络情况时能否正常进入免登录状态。
3. 切换用户登录后，要校验用户登录信息及数据内容是否相应更新，确保原用户退出。
4. 根据MTOP的现有规则，一个帐户只允许登录一台机器。所以，需要检查一个帐户登录多台手机的情况。原手机里的用户需要被踢出，给出友好提示。
5. APP切换到后台，再切回前台的校验
6. 切换到后台，再切换回前台的测试
7. 密码更换后，检查有数据交换时是否进行了有效身份的校验
8. 支持自动登录的应用在进行数据交换时，检查系统是否能自动登录成功并且数据操作无误。
9. 检查用户主动退出登录后，下次启动APP，应停留在登录界面。

## 6.数据更新部分

根据应用的业务规则，以及数据更新量的情况，来确定最优的数据更新方案。

1. 需要确定哪些地方需要提供手动刷新，哪些地方需要自动刷新，哪些地方需要手动+自动刷新。
2. 确定哪些地方从后台切换回前台时需要进行数据更新。
3. 根据业务、速度及流量的合理分配，确定哪些内容需要实时更新，哪些需要定时更新。
4. 确定数据展示部分的处理逻辑，是每次从服务端请求，还是有缓存到本地，这样才能有针对性的进行相应测试。
5. 检查有数据交换的地方，均有相应的异常处理。

## 7.APP更新部分（升级update）

1. 当客户端有新版本时，有更新提示。
2. 当版本为非强制升级版时，用户可以取消更新，老版本能正常使用。用户在下次启动APP时，仍能出现更新提示。
3. 当版本为强制升级版时，当给出强制更新后用户没有做更新时，退出客户端。下次启动APP时，仍出现强制升级提示。
4. 当客户端有新版本时，在本地不删除客户端的情况下，直接更新检查是否能正常更新。
5. 当客户端有新版本时，在本地不删除客户端的情况下，检查更新后的客户端功能是否是新版本。
6. 当客户端有新版本时，在本地不删除客户端的情况下，检查资源同名文件如图片是否能正常更新成最新版本。如果以上无法更新成功的，也都属于缺陷。  
   7)新版本可以离线安装覆盖当前版本。  
   8）升级不能覆盖用户的数据。

## 8.按钮

软件按钮：软件中相关按钮的功能实是否实现，如返回按钮之类的。  
物理按钮：指手机的功能键按钮的功能实现的测试。

## 9.忘记密码

## 10.个人信息管理

主要是对已注册用户，登录后的相关信息的管理，信息的内容与需求有关。

## 11.分享

1）分享功能是否能正常分享  
2）APP上分享成功后，需检测实际微博上是否分享成功 ，分享鉴权是否能正常通过微博用户名和密码的验证（验证成功和失败需提示）  
3）取消分享是否能正常取消，是否会出现卡住情况 分享鉴权成功后，下次分享是否还需再次输入微博用户名和密码  
4）邮件分享时，当机器没有绑定邮箱时，页面提示是否正确或者能否成功跳转到邮件绑定页面。

## 12. 支付功能检测

1）订单金额与支付金额验证（单件，多件总额）数额一直（可使用特殊金额如：0、99、99.999、44.44445等）  
2）支付优惠劵或者选择扣款验证，能正常使用  
3）未提交支付的时候能正常取消支付　　  
4）在未提交支付页面长时间停留再次操作需要给用户提示  
5）点击支付响应后，在单个交易周期里只响应一次  
6）订单支付失败可以重新进入支付流程　　  
7）用户在支付订单的时候需要再次提示用户金额以及收款方信息用户确认  
8） 如果用户发现数据不对可以返回重新提交支付  
9）用户确认付款后跳正常转到响应的支付页面（APP本身的支付页面，其他APP或者网页的交互平台。根据业务的不同进行测试）　　  
10）测试账户下余额不足的情况下支付失败，给出提示，但不扣款  
11）测试账户下余额足够的情况下支付成功，能给出提示和跳转，扣款成功  
12）测试账户下余额不足的情况下，取消支付。正常取消。  
13）测试账户下余额充足的情况下，取消支付。正常取消。  
14）支付成功后订单状态正常转换  
15）支付失败或者取消支付是订单状态不发生变化  
16）网络异常情况下测试即服务器正常响应了请求包，但 APP收到请求报超时。APP订单应该重新刷新状态  
17) 支付账户余额不足补足后可重新进行正常支付

## 13.app关联事务

因为不同公司的APP关联的事业有所不同，这里主要简单讲一下，主要有以下几点：  
1、与其它应用有业务关联  
2、公司的用户体系  
3、其他

## 14.app外部环境功能及测试要点

### 1、网络

（2G、3G、4G、wifi、离线）app的运行情况，主要有以下几点:  
1）离线模式下,功能是否正常 在线模式下加载过的页面会缓存,在离线模式下是否能显示缓存的页面  
2）在线模式未加载过的页面,在离线模式下是否有加载图标或者提示  
3）离线模式下操作需要连接网络的功能,是否会有网络未连接提示  
4）离线模式下操作不需要连接网络的功能，是否操作正常  
5）离线模式下操作拨打电话的功能，提示是否正常

### 2、系统（OS）

IOS:(系统版本的适配、屏幕尺寸的适配）  
Android:(厂商（如三星、小米、华为）、系统版本、屏幕尺寸的适配）

### 3、分辨率

主要是测试app的适配情况。

### 4、任务管理器进入

### 5、桌面点击再次进入

### 6、手机交叉事件测试

1)低电量、来电话、来短信、锁屏、  
2)push提 醒、usb数据线插拔提醒  
3)充电提醒是否能正常保用

## 15.app其他测试

### 1、 UI检测

1）UI风格是否符合UI原型图标准  
2）字体类型，字体大小是否与UI原型图保持一致  
3）间距，边距是否和UI原型图保持一致(包括图片,文本,图标等)  
4）UI按钮，图标，图片的阴影效果  
5）UI按钮，图标，图片的高效果  
6）图片的清晰度，是否失真  
7）是否支持横竖屏，旋转  
8）图片显示是否完整，清晰（旋转过后或者45度角度倾斜旋转）

### 2、ios与android机器交互检测

APP有ios版本和android版本才需要这些测试  
1）ios与ios机器相互发送消息、图片、语音、能正常接收  
2）android与android机器相互发送消息、图片、语音，能正常接收  
3）ios与android机器相互发送消息，图片，语音

以上是APP功能测试流程的基础部分，具体的各个功能点的测试就要看，被测试的APP的具体功能才能设计满足需求的测试用例流程。在APP测试中功能方面一向是最复杂的工作，几千条用例不在少数。

APP测试，除了传统的那些测试方法外，需要额外关注以下方面：  
兼容性测试、流量测试、电量测试、弱网络测试、稳定性测试、安全测试和环境相关测试。

### 第一：兼容性测试

针对App通常会考虑这些方面：  
1）操作系统版本  
包括Android版本，iOS版本  
2）屏幕分辨率  
android 800480, 960640,1280720(720p),19201080(1080p),2560\*1440(2k).  
对于iOS,考虑最近几代机型对应的分辨率即可.  
3）不同厂家的ROM  
不同厂家的ROM,大多厂家都对android 系统进行了定制、实际中会遇到例如调用相机和底层服务出现的不兼容问题以及摇一摇遇到的不同手机对于方向和重力传感器灵敏度设置不同的问题.  
4) 网络类型  
网络类型通常考虑wifi,2g,3g4g下的功能情况。另外针对m版网站考虑不同浏览器类型和屏幕分辨率.

### 第二：流量测试

在移动产品的测试中，很有必要对App使用的流量进行度量，大致来说，流量可以从用户使用的相关性角度分为：一类是用户的操作直接导致的流量消耗；另一类是后台，即在用户没有直接使用情况下的流量消耗。  
流量的测试方法：  
基于系统自带功能.  
eg android proc/uid\_stat/{uid} /tcp\_send  
android proc/uid\_stat/{uid} /tcp\_rcv  
通过API或者系统埋点来获取数据。  
通用的流量测试方法：手机抓包，或者wifi代理（Fiddler， Charles）。  
常见的流量节省方法：  
数据压缩。  
压缩包含接口文本数据的压缩，js文件的压缩及图片的压缩。  
不同数据格式的采用  
例如采用JSON格式作为接口数据返回格式通常比XML格式要小。  
控制访问的频次  
这个主要针对后台数据上报，PUSH消息检查等定时机制的。  
只获取必要的数据  
有时候APP一页的内容非常多，而用户可能只会看一部分，过多的从后台拉去数据就是浪费，所以可以采用分屏加载或者懒加载的方式来减少流量消耗。  
缓存  
可将图片，js等数据暂存起来，但由于手机存储空间有限，也需要控制整个缓存大小，并给用户提供清理缓存的选项。  
针对不同网络类型设计不同的访问策略  
有些APP不同的网络类型返回的内容不一样。

### 第三：电量测试

在电器电池技术没有取得巨大突破前提下，这方面始终会存在一些瓶颈，如果一些App架构设计的不好，或者代码偶缺陷，就可能导致电量消耗比较高，所以电量测试也是很重要的。  
工具 GSam Battery Monitor Pro.

### 第四：弱网络测试

移动互联网产品相比PC互联网产品，有一个特点是前者使用的网络比较多样，除了Wifi之外，很多时候是在移动网络下使用的，移动网络遇到的情况又比较复杂，比如地铁、隧道、体育场等。所以网络不稳定的情况是比较容易发生的，很多情况下App的一些问题是在复杂的网络情况下才会暴露，与其让用户发现和投诉这些问题，不如我们在测试阶段尽量模拟这样的网络情况，及早发现和修复这些问题。  
工具：  
Windows下的Network Delay Simulator  
Mac下的Network Link Conditioner

### 第五：稳定性测试

在保证基本功能正确基础之上，App的稳定性就显得非常重要，如果一个App经常出现闪退或者卡死，那么用户体验就会受到很大伤害，在有其他竞争产品的情况下很容易造成用户的流失。

### 第六：安全测试

包括安装包的安全测试（能否反编译代码、安装包是否签名，完整性校验，权限设置检查等）。  
敏感信息测试（数据库，日志，配置文件）。  
软键盘劫持（金融类APP登录页面的用户名密码输入框）、  
账户安全（密码是否明文，密码传输是否加密，账户输入错误次数过多锁定，同时会话提醒， 注销机制）  
数据通信安全（关键数据是否散列或加密，关键连接是否使用安全通信，是否对数字证书合法性进行验证，是否校验数据合法性。  
组件安全测试。  
服务器端接口测试（SQL注入测试、XSS跨站脚本攻击， CSRF跨站请求伪造，越权访问等）。

### 第七：环境相关的测试

在实际项目中，有一些缺陷我发现是和App所处的运行环境相关的，所以设计测试的时候，要多考虑这些场景，比如：  
1）干扰测试  
收到电话、收到短信、收到通知栏消息、无电提示框弹出、第三方安全软件告警弹出。  
2）权限测试  
一些用户在实际使用App的时候会有意识阻止某些功能。例如有的用户感觉让某个App访问电话本或者相册可能泄漏隐私，就在手机中设置了禁止了该App访问相册的权限。  
3）边界测试  
手机环境本身也有其边界情况需要在测试中覆盖。常见的场景有：  
可用存储空间过少、没有SD卡/双SD卡、飞行模式、系统时间有误（晚于和早于标准时间）、第三方依赖（比如我们的App依赖第三方App，但是现在第三方App没有安装或者版本过低的测试情况）。  
4）Android定位测试  
用白盒方式模拟