****

**员工年度总结表**

**（review流程：入职每满一年当月填写总结表，次月发薪前出review结果）**

**填表时间： 年 月 日**

|  |  |  |  |
| --- | --- | --- | --- |
| **基本信息** | | | |
| 姓 名 | 兰杨能 | 入职时间 | 2020.04.22 |
| 项 目 | RemarkableFoods QA Automation | 岗 位 | JAVA Engineer |
| **员工年度总结** | | | |
| **2021年工作中的自己用一句话总结，即：**  独立完成工作，主动、多方位思考，不断挑战新工作，掌握新技能。  **2021 年度关键词：**  坚持、独立、主动。  **2021年我完成的工作：**   1. 创建测试报告管理平台1.0和测试报告管理平台2.0，完成测试报告的采集、上传、存储、实时访问的工作。 2. 自动化准备、清理Wonder APP测试数据，设计“准备数据-测试-清理数据”的脚本模式，准备数据包括zone、truck、cor、user、credit，根据计算分配未被使用的测试地址和用户给测试脚本测试，完成测试脚本数据隔离的目的，并为以后达到脚本并行测试奠定基础。   3. 设计API代理BO、APP功能。模拟fmt、ccp、driver app、onfleet app等站点和app的功能，供测试脚本调用，包括但不限于push oder、modify order、refund order、create truck session、modify truck session status、picking list、 push onfleet task等。  4. 打包服务器维护、开发。维护Wonder APP 的打包（ipa、app、apk）、测试、回归测试等功能；新增Restaurant Portal APP的打包功能；升级和维护打包环境。  5. Appium、JUnit测试框架调研、应用。  6. 协助其他组成员工作，批量、定时准备truck session数据。  7. 维护、开发自动化测试脚本1.0和app自动化测试框架1.0，按迭代完成Wonder APP本地测试、回归测试、AppCenter测试，以及测试结果整合、汇报工作。  **2021年我的工作历程：**   1. 在测试组人手不够的时候，从初次接手自动化测试工作，到独立完成自动化测试的所有工作，我完成了以下几个主要任务：   1. 从0到1 实现精通APP自动化测试框架1.0。维护和新增测试脚本，熟练掌握已有的测试脚本，修复测试脚本。完成QA（ios、android）本地测试，UAT（ios、android）本地测试，UAT（ios）旧版本回归测试，UAT（ios、android）新版本APPCenter测试，完成人工采集、解析、汇报测试结果工作。  2. 熟悉react native代码结构。由于测试脚本需要在wonder APP代码中进行元素提取，在前端同事的帮助下，掌握阅读wonder app 代码，在wonder app代码中快速定位元素、埋点。  3. 解决测试报告滞后汇报问题，优化测试报告生成流程。测试一台设备平均花费1.5个小时，在有设备进行并行测试时，一批测试也需要花费很多时间，导致报告汇报滞后。经过和jake的分析和讨论，给设备设置优先级，采取分批汇报测试结；同时争取时间，优化人工采集测试报告的流程，在测试过程中，将测试报告压缩以ftp协议上传到mac mini服务器，同时在mac mini上部署Allure测试报告管理服务，实时解析和在线访问测试报告，提高测试工作效率，即测试报告管理平台1.0。  4. 管理打包服务器。测试最重要的是基础是wonder app的测试包，由部署在mac mini 上的打包服务器提供，我的工作任务是掌握mac mini上面ios、app、apk三种类型包的打包流程，同时配合前端进行打包环境的升级，处理打包失败的各种问题。  二、在新的测试人员加入测试组后，我的主要工作是：  1. 校验Wonder APP上MParticle event事件触发次数。因为在测试过程中无法从Wonder APP上获取到事件到唯一标识，MParticle系统只存放实时事件无法进行批量查询，且从MParticle系统同步数据到snowflake DB存在事件丢失的情况；经过和前端讨论，根据测试时间、设备唯一标识、测试会话去区分每次测试的事件，在测试结束1小时后的时候去snowflake DB获取wonder APP上报的次数，和测试脚上报的次数进行比较，并将结果存放在测试报告管理系统2.0，解决MParticle event事件次数校验问题、校验结果查看问题。   1. 设计API代理BO、APP功能。因为测试过程需要操作bo后台、dirver app、   Onfleet app, 且需要绕过登陆权限验证，成立“test-agent-site”模块用来代理人工行为，为测试顺利进行提供有效的帮助。遇到比较难的问题是需要模拟的功能在可视化界面上交互复杂，需要校验各种状态，需要调度其他项目组的API，需要了解其他项目模块的业务，再结合测试脚本的需求，设计代理API满足测试需要。其中复杂的包括push order status、modify order、refund order、push onfleet task、create truck session、picking list等。  3. 为web UI 测试人员和 压测人员提供测试数据的创建和清理API，包括create truck session、offDuty truck session、cancel order 等。  4. 帮助同组人员code review。  三、除了本组的工作，还为其他组提供了一些帮助：  1. 帮助ccp组完成定时上线truck session。  2. 帮助recipe组完成批量创建定制化truck session。  3. 帮助marketplace组完成Restaurant Portal app的可视化打包功能。  **2021年我取得了哪些进步：**  1. 设计能力提升。因为要实现的功能需要跨多个项目组，需要了解其他组的业务、分析其他项目的代码，设计并实现测试脚本需要的功能，对需求truck session的创建，库存的加载，订单的相关处理、onfleet task的相关处理等需求进行拆解分析，再考虑编码本身的健壮性，安全，可复用性，性能等非功能性需求，逐步提升设计能力。  2. 业务需求和业务流程的分析能力提升。因为要创建测试报告管理平台，从原型的设计、数据库的设计、前端页面、后端代码，再到测试脚本上报测试报告的流程、测试服务器环境的配置，通过对业务的分析能够抽象关键的业务模型和业务用例，业务建模完成后进行系统建模。  3. 思维能力的提升。  **2021年我有哪些需要改善的地方：**   1. 向上沟通能力不足。   **2022年工作思路：**  大方向跟着leader走，分步走、逐渐实现各个阶段的工作任务。  **2022年工作规划：**  1. app自动化测试框架2.0  2. 测试期间日志采集系统  3. 测试用例管理平台搭建  4. 测试报告管理平台3.0  5. web自动化测试框架2.0  **2022自身能力提升计划：**  1. 熟练应用Core-NG框架  2. 分析流行测试框架  3. 每周算法  4. bec中级  **对团队建议：**  1. 工作需求的生成，需要整组人员的讨论，再决定是否有必要落地，建议进行需求价值评估。  2. 组内成员工作互相不透明，工作进度不一致，工作推进困难，沟通不方便，建议开迭代会或每日站会。  3. 建工作issue描述需求的时候，建议应用场景、需求、目的要描述清楚，不要模棱两可，不要贴聊天记录。  4. 建议协调制定出规范化的工作模式。  **My self at work in 2021 can be summed up in one sentence, namely:**  Complete work independently, take the initiative and think in multiple directions, constantly challenge new work, and master new skills.  **2021 annual keywords:**  Persistence, independence, initiative.  **My work done in 2021:**  1. Create test report management platform 1.0 and test report management platform 2.0 to complete the collection, upload, storage and real-time access of test reports.  2. Automatically prepare and clean Wonder APP test data, design a script mode of "prepare data-test-cleanup data", prepare data including zone, truck, cor, user, credit, and allocate unused test addresses and users according to calculation. Test script testing, complete the purpose of test script data isolation, and lay the foundation for parallel script testing in the future.  3. Design API proxy BO and APP functions. Simulate the functions of sites and apps such as fmt, ccp, driver app, onfleet app, etc. for test scripts to call, including but not limited to push oder, modify order, refund order, create truck session, modify truck session status, picking list, push onfleet task Wait.  4. Package server maintenance and development. Maintain the packaging (ipa, app, apk), testing, regression testing and other functions of Wonder APP; add the packaging function of Restaurant Portal APP; upgrade and maintain the packaging environment.  5. Appium, JUnit test framework research and application.  6. Assist other team members to prepare truck session data in batches and regularly.  7. Maintain and develop automated test scripts 1.0 and app automated test framework 1.0, and complete Wonder APP local testing, regression testing, AppCenter testing, and test results integration and reporting work iteratively.  **My work history in 2021:**  First. When the test team was short of manpower, I completed the following main tasks from taking over the automated testing work for the first time to completing all the work of automated testing independently:  1. From 0 to 1 to achieve proficient in APP automation testing framework 1.0. Maintain and add new test scripts, master existing test scripts, and repair test scripts. Completed QA (ios, android) local test, UAT (ios, android) local test, UAT (ios) old version regression test, UAT (ios, android) new version APPCenter test, completed manual collection, analysis, and reporting of test results.  2. Familiar with react native code structure. Since the test script needs to extract elements from the wonder app code, with the help of front-end colleagues, master reading the wonder app code, and quickly locate elements and bury points in the wonder app code.  3. Solve the problem of delayed reporting of test reports and optimize the process of generating test reports. It takes an average of 1.5 hours to test one device, and when there are devices for parallel testing, a batch of tests also takes a lot of time, resulting in a lag in reporting. After analysis and discussion with jake, set priorities for the devices and report the test results in batches; at the same time, gain time to optimize the process of manually collecting test reports. During the test process, the test reports are compressed and uploaded to the mac mini server by ftp protocol. , at the same time deploy Allure test report management service on mac mini, real-time analysis and online access to test reports, improve test work efficiency, namely test report management platform 1.0.  4. Manage packaging servers. The most important test is that the foundation is the test package of the wonder app, which is provided by the packaging server deployed on the mac mini. My task is to master the packaging process of the three types of packages: ios, app, and apk on the mac mini, and cooperate with the front end. Upgrade the packaging environment to deal with various problems of packaging failure.  Second. After new testers join the test group, my main tasks are:  1. Verify the number of times the MParticle event event is triggered on the Wonder APP. Because the event to the unique identifier cannot be obtained from the Wonder APP during the test process, the MParticle system only stores real-time events and cannot perform batch query, and the synchronization data from the MParticle system to the snowflake DB has event loss; after discussions with the front-end, according to the test Time, device unique identifier, and test session to distinguish the events of each test, go to snowflake DB 1 hour after the end of the test to obtain the number of times reported by the wonder APP, compare it with the number of times reported by the test feet, and store the results in the test report The management system 2.0 solves the problem of checking the number of MParticle events and checking the verification results.  2. Design API proxy BO and APP functions. Because the test process needs to operate the bo background, driver app,  Onfleet app, and needs to bypass the login permission verification, the "test-agent-site" module is established to act as a proxy for human behavior, providing effective help for the smooth running of the test. The difficult problem is that the functions that need to be simulated are complex to interact on the visual interface, various states need to be checked, the APIs of other project groups need to be scheduled, and the business of other project modules needs to be understood, and then combined with the requirements of the test script, design agents API meets testing needs. The complex ones include push order status, modify order, refund order, push onfleet task, create truck session, picking list, etc.  3. Provide APIs for creating and cleaning test data for web UI testers and stress testers, including create truck session, offDuty truck session, cancel order, etc.  4. Help the same group to review the code.  Third. In addition to the work of this group, it also provides some help for other groups:  1. Help the ccp group complete the scheduled online truck session.  2. Help the recipe group to create customized truck sessions in batches.  3. Help the marketplace team complete the visual packaging function of the Restaurant Portal app.  **What progress I have made in 2021:**  1. Improve design ability. Because the function to be implemented needs to span multiple project groups, it is necessary to understand the business of other groups, analyze the code of other projects, design and implement the functions required by the test script, create the demand truck session, load the inventory, order related processing, The related processing requirements of the onfleet task are disassembled and analyzed, and then the non-functional requirements such as the robustness, security, reusability, and performance of the coding itself are considered, and the design ability is gradually improved.  2. The ability to analyze business needs and business processes is improved. Because it is necessary to create a test report management platform, from the design of the prototype, the design of the database, the front-end page, the back-end code, to the process of reporting the test report by the test script, and the configuration of the test server environment, the key business can be abstracted through business analysis. Models and business use cases, system modeling after business modeling is completed.  3. Improve thinking ability.  What do I need to improve in 2021:  1. Insufficient upward communication skills.  2022 work ideas:  Follow the leader in the general direction, step by step, and gradually realize the tasks of each stage.  **2022 work plan:**  1. App automation testing framework 2.0  2. Log collection system during the test  3. Test case management platform construction  4. Test report management platform 3.0  5. Web automation testing framework 2.0  **2022 Self-ability improvement plan:**  1. Proficient in applying Core-NG framework  2. Analysis of popular testing frameworks  3. Weekly Algorithm  4. bec intermediate  **Advice to the team:**  1. The generation of job requirements requires the discussion of the entire group, and then decides whether it is necessary to implement. It is recommended to conduct a demand value assessment.  2. The work of members in the group is not transparent to each other, the work progress is inconsistent, the work progress is difficult, and the communication is inconvenient. It is recommended to hold an iteration meeting or a daily stand-up meeting.  3. When creating a work issue to describe the requirements, it is recommended that the application scenarios, requirements, and purposes should be clearly described, not ambiguous, and do not post chat records.  4. It is recommended to coordinate and formulate a standardized working model. | | | |