****

**员工年度总结表**

**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 | 岗 位 | Java Engineer |
| **员工年度总结** | | | |
| 感恩在公司工作满一年啦，在过去的一年里，我依序主要参与了三个项目：DW、Shift-Scheduler、自动化测试，当前所在的项目组是自动化测试平台组。前期能够认真完成DW、Shift-Scheduler、Slack Library、Fleet-Zone等相关项目的迭代工作，后期在自动化测试组能够独立负责app的回归测试、维护旧测试代码和为搭建自动化测试平台做准备；在应用、巩固、拓展自己已有的JAVA后端开发知识的同时，也不断的在学习测试技能应用、测试理论知识和提升测试工作经验。  其中我在DW项目中负责etl数据迁移，项目已完结。在Shift-Scheduler项目负责系统后端开发任务，第一期工作已完结。在自动化测试项目组负责wonder app的测试工作和测试平台的搭建，工作还在进行中。  根据项目分三部分汇报的我工作：  一、在DW项目中，我的工作内容是应用Kafka将其他项目组的数据从Mongo数据库迁移到MySQL数据库，初次正式使用Core-ng框架来做数据迁移，在导师的指导下顺利完成了工作。  二、在shift-scheduler项目中，我的工作内容是根据每次迭代分配的issue、严格按照导师制定的开发流程完成开发工作。在本项目工作前期，存在一系列问题（以下7点），后期有逐一解决:  1. 对Dayforce 模块没有交接过程，没有明确文档，写代码之前需要花费时间去了解原先的逻辑和细节，偶尔导致工作逾期完成。后期在工作中，采取绘制流程图和时序图对工作内容进行整理和记录。  2. 对新型数据库Cosmos DB使用不熟悉。后期在处理不熟悉的数据库语法时，首先快速查阅官方文档，还是不能很好解决问题会就主动向熟悉的人寻求帮助，业余抓紧时间去学习和使用Cosmos DB相关知识，使得自己能够在工作中熟练应用Cosmos DB。  3. 在每个迭代计划时间内，会有其他任务再分派给我紧急处理，后期从开发者和团队协作经验的角度出发，改变自己的工作方式，在处理此类问题时：先快速分析问题、定位问题的处理难度和紧急程度；接着把当前问题和已有工作排列优先级；然后向leader提交解决方案后再执行处理，保证个人和团队的工作互相协调。  4. 前期试应leader制定到开发流程和团队协作的工作流程，对于复杂的issue在处理上比较缓慢。后期在面对复杂的任务，将任务进行拆解成小任务，再评估每一个小任务的工作时长和紧急程度，划分优先级，向leader提供任务调整方案，提高工作效率、降低出错的风险。  5. Shift-Scheduler与Fleet Management Tool项目之间存在交互，前期排bug的时候需要去跨端排查，需要先去了解一下上游的逻辑，定位问题难。后期再遇到跨端交互相关问题，先排查本系统问题，接着寻找上游系统相关负责人排查问题，寻找机会了解业务，同时优化项目旧代码，做到代码显式实现，方便维护和排查问题。  6. 没有养成技术埋点的习惯，排bug花的时间比较长。后期思考埋点的重要性，学会区分技术埋点、业务埋点、功能埋点，养成埋点的好习惯，让自己根据日志即可快速定位问题。  7. 对每个issue完成时间预估不准确。后期在leader的帮助下，且掌握相关经验后，每个issue从需求分析，流程设计，API设计、编写代码等各方面进行详细的时间规划。  三、在自动化测试项目组，我的工作内容是为后续搭建自动化测试平台做准备和给wonder app做自动化测试，包括测试数据准备和清理，测试脚本编写、维护、优化，QA本地测试，UAT回归测试，App Center上测试，整理测试报告，还有wonder app打包项目的维护。由于目前只有我一个人在做wonder app的自动化测试，有时候会忙不过来，主要存在以下几点问题：  1. 第一次做自动化测试这一块，确实比较费劲，遇到问题需要花较多的时间去学习和消化。针对这类问题，整理了当前app回归测试所需要掌握的专业知识，在不影响日常工作进度的前提下，制定了相关的学习计划。  2. 跟其他系统的对接，测试数据和环境不稳定，导致测试进度滞后，针对这个问题没有想到可执行的解决方案。  3. 现在测试系统本身不支持并行测试，一个设备至少也要跑1.5个小时，导致测试进度滞后。时间不够用问题在第三方App Center上进行测试时体现得特别明显，寻找jake协调后，针对设备型号制定了优先级，对设备进行分批测试和汇报测试结果，目的在于让测试工作有条不紊的进行。  4. 准备自动化测试平台的搭建。时间花费分为3个部分：  第一部分：搭建数据初始化项目。了解所涉及系统的业务，采用Core-ng框架，引入公司内部其他组项目的依赖，实时与其他组项目的变更保持同步。  第二部分：搭建自动化测试项目。了解allure，将整理测试报告流程自动化，实现在线访问和管理。搭建规范的自动化测试项目，引用设计模式和测试框架。  第三部分：打包加回归测试项目。熟悉该项目，主要是了解\*.app,\*.ipa,\*.apk的命令行打包流程，测试和回归测试的流程。  总结在工作过程中我的收获，我在DW项目中负责etl数据迁移，学习和巩固Core-ng框架的使用。在Shift-Scheduler项目负责系统后端开发任务，团队协作的沟通能力，学会评估工作时间，注重提高工作效率；学会微服务分层架构理论入门、代码开发规范化且养成深度思考的好习惯，提高自己代码的可读性和可维护性。在自动化测试项目组独立负责wonder app的测试工作、搭建自动化测试平台，学会跨组沟通和处理问题的能力；为了按时出测试报告和维护旧测试代码，学会冷静处理和合理规划工作安排。  反思在工作过程中我的不足之处，对于初次遇见的问题，容易手忙脚乱，需要调整心态、变得更有耐心。在组内沟通和跨组沟通上，没有很强的危机意识，沟通不够及时，会改进。  在畅拓的一年，是收获满满的一年，不仅专业知识技能上得到拓展，最重要的是学习到了宝贵的工作经验和沟通技巧。  我在wonder app的开发过程中承担质量保证的角色，希望在未来可以结合自身JAVA后台开发经验和自动化测试知识，为了搭建自动化测试平台做更完整的准备，提高质量保证工作的效率。在未来一年我的工作目标是学习UI自动化测试框架，优化app测试项目，搭建自动化测试平台。我需要学习测试理论、测试设计模式、UI自动化测试框架、Allure、JUnit4、Appium。  I am grateful to have worked in the company for a full year. In the past year, I have participated in three projects in sequence: DW, Shift-Scheduler, and automated testing. The current project team is the automated testing platform team. In the early stage, I can earnestly complete the iterative work of DW, Shift-Scheduler, Slack Library, Fleet-Zone and other related projects. In the later stage, I can independently be responsible for the regression test of the app, maintain the old test code and prepare for the establishment of the automated test platform in the automated test group; While applying, consolidating, and expanding my existing JAVA back-end development knowledge, I am also constantly learning the application of testing skills, testing theoretical knowledge and improving testing work experience.  Among them, I am responsible for etl data migration in the DW project, and the project has been completed. Responsible for the system back-end development tasks in the Shift-Scheduler project, the first phase of work has been completed. In the automated testing project team, I am responsible for the testing of the wonder app and the construction of the testing platform. The work is still in progress.  My work is reported in three parts according to the project:  <First>. In the DW project, my job is to use Kafka to migrate data from other project groups from Mongo database to MySQL database. For the first time, I formally used the Core-ng framework for data migration. I successfully completed the work under the guidance of the instructor.  <Second>. In the shift-scheduler project, my work content is to complete the development work according to the issue assigned in each iteration and strictly in accordance with the development process set by the instructor. In the early stage of this project, there were a series of problems (the following 7 points), which were solved one by one in the later stage:  1. There is no handover process for the Dayforce module and no clear documentation. It takes time to understand the original logic and details before writing the code, which occasionally causes the work to be completed overdue. In the later work, drawing flowcharts and sequence diagrams are used to organize and record the work content.  2. Not familiar with the use of the new database Cosmos DB. Later, when dealing with unfamiliar database syntax, first quickly consult the official documents. If the problem is still not solved well, I will take the initiative to seek help from familiar people. I will take the time to learn and use Cosmos DB related knowledge in my spare time so that I can work. Familiar with Cosmos DB.  3. During the planning time of each iteration, there will be other tasks assigned to me for urgent processing. Later, from the perspective of developer and team collaboration experience, I will change my way of working. When dealing with such problems: first analyze the problem quickly , Locate the difficulty and urgency of the problem; then prioritize the current problem and the existing work; then submit the solution to the leader before performing the processing to ensure that the work of the individual and the team is coordinated with each other.  4. In the early stage, the leader develops a workflow from the development process and team collaboration, and it is relatively slow to deal with complex issues. In the later stage, when faced with complex tasks, the tasks are disassembled into small tasks, and then the working time and urgency of each small task are evaluated, prioritized, and task adjustment plans are provided to the leader to improve work efficiency and reduce the risk of errors.  5. There is an interaction between the Shift-Scheduler and the Fleet Management Tool project. When troubleshooting the bug in the early stage, you need to cross-end the investigation. You need to understand the upstream logic first, and it is difficult to locate the problem. Later, when encountering problems related to cross-terminal interaction, first check the system problems, and then find the relevant person in charge of the upstream system to check the problems, look for opportunities to understand the business, and optimize the old code of the project to achieve the explicit implementation of the code, which is convenient for maintenance and troubleshooting.  6. There is no habit of burying technology, and it takes a long time to eliminate bugs. Later, think about the importance of burying points, learn to distinguish between technical burying points, business burying points, and functional burying points, and develop a good habit of burying points, so that you can quickly locate problems based on the log.  7. The estimation of the completion time of each issue is not accurate. Later, with the help of the leader and mastering relevant experience, each issue conducts detailed time planning from requirements analysis, process design, API design, and code writing.  <Third>. In the automated testing project team, my job content is to prepare for the subsequent building of an automated testing platform and do automated testing for the wonder app, including test data preparation and cleaning, test script writing, maintenance, optimization, QA local testing, UAT regression Testing, testing on the App Center, organizing test reports, and maintaining the wonder app packaging project. Since I am the only one doing automated testing of the wonder app, sometimes I can’t be too busy. The main problems are as follows:  1. It is really hard to do the automated test for the first time, and it takes more time to learn and digest when encountering problems. In response to this type of problem, the professional knowledge required for the current app regression test was sorted out, and the relevant learning plan was formulated without affecting the daily work progress.  2. Interfacing with other systems, the test data and the environment are unstable, resulting in a lag in the test progress, and there is no executable solution to this problem.  3. Now the test system itself does not support parallel testing, and a device has to run for at least 1.5 hours, causing the test progress to lag. The problem of insufficient time is particularly obvious when testing on the third-party App Center. After looking for jake coordination, priority is set for the device model, the device is tested in batches and the test results are reported, so that the test work is carried out in an orderly manner. .  4. Prepare to build an automated test platform. The time spent is divided into 3 parts:  The first part: build a data initialization project. Understand the business of the system involved, adopt the Core-ng framework, introduce the dependencies of other group projects within the company, and keep synchronized with the changes of other group projects in real time.  The second part: build an automated test project. Understand allure, automate the process of organizing test reports, and realize online access and management. Build standardized automated test projects, citing design patterns and test frameworks.  The third part: packaging plus regression test project. Familiar with the project, mainly to understand the command line packaging process of \*.app, \*.ipa, \*.apk, the process of testing and regression testing.  Summarizing my gains during the work process, I am responsible for etl data migration in the DW project, learning and consolidating the use of the Core-ng framework. In the Shift-Scheduler project, he is responsible for system back-end development tasks, teamwork communication skills, learning to evaluate working hours, focusing on improving work efficiency; learning to get started with microservice layered architecture theory, standardize code development, and develop a good habit of in-depth thinking to improve The readability and maintainability of your own code. In the automated testing project group, I am independently responsible for the testing of the wonder app, building an automated testing platform, and learning the ability to communicate and handle problems across groups; in order to issue test reports on time and maintain old test codes, learn to calmly handle and plan work arrangements reasonably.  Reflecting on my shortcomings in the process of work, it is easy to get confused about the problems encountered for the first time, and I need to adjust my mentality and become more patient. In intra-group communication and cross-group communication, there is no strong sense of crisis, and the communication is not timely enough and will improve.  The year in Changtuo is a year of full harvest. Not only the professional knowledge and skills have been expanded, but the most important thing is the valuable work experience and communication skills.  I assume the role of quality assurance in the development process of the wonder app. I hope that in the future, I can combine my own JAVA background development experience and automated testing knowledge to make more complete preparations to build an automated testing platform and improve the efficiency of quality assurance work. My goal in the next year is to learn the UI automation testing framework, optimize app testing projects, and build an automated testing platform. I need to learn test theory, test design patterns, UI automation test framework, Allure, JUnit4, Appium. | | | |