Mars Test Analyst - Standard Sprint

Introduction:

We believe that you have been through a great processes and journey to complete the Onboarding and Competition Sprint. These previous 2 sprints was prepared you very well with both manual & automation testing skills from automating test cases using Specflow, Selenium DDD, Extent Report and Excel Data Reader.

The Mars Standard Sprint is the final Sprint to prepare you be in the part of the Production team. Task is designed to check the potential of each and every team member before they actually move into the project. Also to check if you can work with a team spirit and achieve the goals before the deadlines. While the instructions tell you what to do, they do not always necessarily tell you how to do it. This is deliberate as it is important for you to develop an independent drive to solve problems on your own. The performance of this Sprint will determine if you are eligible to be invited to work with Production Team.

Standard Sprint Scenario:

• Be able to analyse requirements that are not documented and create test conditions and cases
• Test design process and estimations.
• Using Docker to run the solution locally.
• Automation with Selenium Web Driver using C#, POM, Page Factory, Data driven approach, NUnit, Extent Reports, Excel Data Reader, AutoIt and Asserts.
• BDD Automation with Specflow
• Performance Testing
• API Testing

Instruction:

- You will be working in a team of 3 - 4 members to complete this Sprint. The final solution will be uploaded to Github (preferred option) or Google Drive.
- You will have appointed Team Leader in your team and assigned Mentor who can give you feedback and help you with direction
- Utilise Slack, QuestionHub to ask question - Research and seeking solution online is an important skill if you want a long term career in Software Industry.
- Create Google Sheet to track your team member progresses, tasks status by copy and paste the Task List.
- Sets of tasks cover various aspects from Manual, Automation (Selenium & Specflow), Performance Testing and API Testing.

- Completing this sets of tasks, you will be very confident with your technical testing skills before working closely with our Production QA Team

Task List:

Task List

Frequently Asked Questions:

1. Why should I start from step 1 for every single test case?
    a. Assuming that you're a executing a test case that was written by your team mate. If the test steps does not clearly describe in a sequential order on how it needs to be tested, you will have to be dependent on that tester to have your questions addressed. To avoid this, it is always a good practice to have test steps starting from Login step for every test case.

2. Why is it necessary to have expected results for every test step?
    a. If there is a test case with 15 test steps and one expected result, a test executor would not know which step of the test case failed. So, in order to make it very clear it is important to have expected result designed for every test step.

3. Why we need to choose the complexity for the Test case?
    a. Most of the companies, they will calculate the person's work based upon the test case complexity, A person can execute 8 complex test cases, 16 Medium test cases and 32 simple test cases per day. Apart from this we can also choose/select the complex test cases to do the automation.
        a.) Simple: The test cases that takes less time and have less steps to execute, we will mark them as "Simple"
        b.) Medium: Which takes a bit longer than the Simple test cases
        c.) Complex: Test Cases which have many steps to execute and takes a bit longer time, we will mark them as "Complex".

4. Why we need to attach the Defect ID to the test case sheet?
    a. It is always a good idea to map your test cases with the defects to avoid confusions. In Excel, for your test case sheet it is always recommended to have a column for your Defect ID, so that you will know if the defect is already logged or not. Also If the Defect reappears after it is closed, you can Reopen the same defect if you have a defect ID with the test case.

5. Things to remember when a task is allocated
    a.) Write Test conditions, have reviewed it with your team lead
    b.) Based on the feedback, modify your test conditions and start writing the test cases

c.) Have your test cases reviewed by the Team Lead again, If any further modifications need to be done, please do it

d.) Execute your test cases when your Team lead asks you to do it

e.) Log the Defects for the failed test cases in JIRA(Please take help from the Team lead if you have no idea about JIRA)

f.) Follow up your defects whenever there is a deployment

6. Whom should I contact if got a question?

You have been allocated to your mentors/Senior Test Analysts on the first day of your internship. And if you need any help you can reach out them. But in all the times we expect you to at least do some basic research to sort out your problems before you seek the help from mentors.

Commonly made mistakes!

1. Every test case should start from step 1 which is from Login
2. Please provide double quotes when using the words from the UI
3. Always write the "Expected Result" in future tense
4. Write the steps in an order so that you will not miss the flow
5. Provide Unique Test Conditions ID and Test case ID
6. Please follow the “Things to remember” that are mentioned above