Hiring test for QA Engineer 1

Here are a few key functionalities we would like to consider in particular in this testing exercise:

1. **Ability for the user to authenticate with username and password, at https://demo.fundportal- staging.com/auth/login/ with the following valid credentials:**

Username: [redacted]

Password: [redacted]

2. **See a list of menu items such as CRM Contacts, CRM Companies, CRM Deals, Portal Home, etc.**
3. **An authenticated user can access the companies table and see a list of 20 companies per page**

An authenticated user can change the number of companies to 50 and 100 companies per page

4. **An authenticated user can search for the keyword 'securities' and have 6 companies returned**
5. **An authenticated user can filter by Company Subtype 'Bank' and Country 'Germany' and return 6 results**

An authenticated user can clear their filters. An authenticated user can add the 'Tags' filter to the table, and apply the [redacted]tag to filter results

6. **A user can select these results and export them (results in an Excel file being downloaded, which contains information about these company records)**

An authenticated user searches for [redacted]Select both company records. Merge those companies using the company record with the most contacts as the primary company record

7. **An authenticated user searches for [redacted]Select all results. Delete the selected company records**

Tasks:

1. Please describe, in text, an overall QA strategy for such a SaaS product, including your answers to the following questions:

   Below paragraphs describe QA process, test techniques and testing levels for EdgeFolio SaaS product FundPortal.

   QA process starts with test planning: For each new feature or change on the modules according to the scale of the changes or feature, a test plan may be prepared. Test plan may be prepared for each release cycle if there is any. In this test plan, we have what to be tested, testing schedule, entry criteria, exit criteria, risks, success measurement metrics, test templates, etc. At this stage, QA team should be involved as early as possible in the evolvement of the feature, or changes. This is paramount important as the principle of Shifting Left approach.

   Next stage is test analysis: In this stage, what is to be performed: general analysis of test artefacts, such as requirements in detail, specifications, user stories for the release, specifying testing artefacts for any new features -like an epic- or more of changes on the current features. Business requirements and specifications are split into functional and non-functional testing. In Agile, dev team and QA as well as PO/BA should discuss the

overall workload, feasibility, applicability of each requirement and specifications, and what level of functional testing and non-functional testing should be executed.

Third step is test design and implementation: At this stage, we define how to test; for this end, we prepare the test cases, test conditions, and identify test data and test environment for both functional and non-functional tests. Based on analytical and regression averse test strategies, the test cases are prioritised, test suits are prepared. If there is any regression suite, it must be updated. For E2E functionality testing, manual tests can be leveraged for the changes or for new feature testing, regression test is planned to run after changes in the product. At this stage, while dev team develop the code and perform test on component and integration level, QA team work on the testing artefacts as to how to test them. QA team also can perform static tests on the test artefacts (as well as new feature files are coded for automated regression.)

Final step is test execution and completion: At this stage, system level test is performed; automation tests are run, manual tests are executed, as well as analysing the test results, false positives etc. For functional tests, negative test scenarios should be executed as well. Alongside the functional tests, performance tests are performed in this stage. Raising issues for the defects, reporting test results are maintained at this stage, too. Retest of fixed issues, re-reporting the test status to the stakeholder are in test execution phase. For SaaS product tests, performance testing is utmost important, as such, QA/dev team should execute the security and stress and load tests for any vulnerabilities or robustness of the system when there is high volume of requests for scalability and availibility. As SaaS products work on role-based authentication, after any changes in the application, it should be tested per tenant profile. Compatibility testing is also highly important to test different systems and devices for various tenants. Within E2E tests (UI and API) front and backend functional test are executed by QA.

After completing the test execution, UAT tests should be performed by UAT team, that could be QA or stakeholders. After the UAT is singed-off, the build is released. At the maintenance level, exploratory testing can be implemented. This is semi-automated deployment, In full-automated deployment, in the prod stage pipeline if there are no issues/no regression, deployment is performed at the end of the pipeline.

Apart from above, QA activities should be monitored and controlled within validation and verification processes. This can be done timely reporting, updating tests documentation whenever needed, checking the quality of test artefacts based on the test results if there is need for more testing.

what aspects of QA would you consider? (e.g. end-to-end functionality, regressions, performance, load testing, security, etc.)

what QA process would you consider putting in place for each of these aspects, to practically test the impact of any changes to the product?

how would this QA process be integrated into the agile software development lifecycle?

2. Implement a few automated tests in the language/tools of your choice, targeting the web app available at https://demo.fundportal-staging.com, that checks that a user browsing to this web app can sign in and perform some of the functionalities described above. You are free to focus on some functionalities to test and skip others, based on your time spent on this exercise and your preference)

   The code is on Gitlab: **gitlab.com/gulhankaralibalci/edgefolio**

   Please also propose other things to test for or to consider in a QA process for this product, such as user errors, anti-patterns, etc.

Usability tests, compatibility tests (mobile phones, tablets), scalability, availability as well as exploratory tests can be performed alongside functional and non-functional tests.

Some examples:

E.g. on the first page, the Disclaimer is leaning on the right side of the page, it can be widened to both sides of the page.

When uploading file, if the user does not utilise the Excel template, it gives 500 error, later it gives right error message, yet, 500 error may not be shown to the users.

When saving search/filter on Find Prospects, special characters are being used, only certain special chars can be allowed.

In the email campaigns, if the emails go to spam folder or not.

On the contacts page, even if none of contacts are selected, export function exported all contacts without giving any warning.

Hiring test for QA Engineer 2

Have the output of the tests be reported in a standard format that can be interpreted by other systems and humans.

Bonus: Propose (in text) a way to embed this test process within the broader software development lifecycle of a small team of developers who use Github Pull Requests to merge new changes to the product

1. *Bonus: Propose a way to regularly test for performance of this web app and detect regressions in its performance when new code is merged before it is deployed to Production (you don't need to implement this test, you can just describe your approach/tooling)*

   CI/CD is integral part of Agile; as such, on different stages, a certain pipeline management system (eg Jenkins) can be utilized. The system can be configured with github, alongside email reporting set up within Jenkins for regression suits, smoke, and sanity tests, integration tests, and deployment. We can set timeout on the critical test cases, when they run successfully, if they cannot perform within given timeout, they fail, as such requested performance is validated.

2. *Bonus: Propose a testing plan for destructive user stories such as "User can delete Companies from their CRM database", and over-all for how to manage datasets and context that is reusable across tests.*

   There are a couple ways that can be done for how to manage datasets, and components during testing. At DAO level, certain annotations may be used to revert the transactions with leveraging Before and After Tests. Again, with BeforeAll and AfterAll annotations in JUnit, test data can be created before the tests in the background, and then can be deleted after the tests or during the tests. Redis like cache databases may be used to manage datasets for short term data life cycles. For example, the transaction life can be limited to a certain time, and it is killed before going to the database. Service virtualisation can be utilized to simulate downstream processes, too.

Deliverables:

During a next interview call, you should be able to share your screen and show the automated tests running and completing, outputting a report of the results. You should also be able to show us the source code or configuration of your solution, so that you can walk us through it and describe your approach and choices. You will also describe your answers to the first question, and we will discuss it together.

I realise this might look either trivial or too long as an exercise, but again feel free to skip some aspects of it or expand on some of them.

Let me know if you're up for that, and if you have any question about this or Edgefolio in general.