1. Please read the documentation of the GitHub API at <https://developer.github.com/v3/>. Suppose that you worked with a team that was tasked with implementing this specification.

• What concerns would you have from a testing perspective?

Answer:

- Setting up the infrastructure that will be used to perform API testing may be a challenge. Deciding on the tool set and figuring out if they serve the purpose and convincing the stake holders may be some concerns initially.
- Any schema changes that may happen needs to be maintained throughout the testing process.
- Assigning data to parameters and making use of these parameters within requests and handling the data types and formatting can be a challenge especially when working on a large project.
- Figuring out what calls to the API must be done sequentially.
- Figuring out the data combinations to be used with the requests made to the API.
- Figuring out what validations must be done on the response.

• How would you go about tackling the QA for this work?

Answer:

- Analyze requirements and clarify any questions from the testing side.
- Figure out what needs to be tested.
- Have test cases in place.
- Figure out what tools are best for performing testing activities.
- Figure out what tests need to be automated.
- Put a testing framework or a foundation in place with the selected tool set.
- Figure out how to manage test data and how to use it in testing activities.
- Do a dry run to demonstrate the tool set serves the testing needs.
- Help other testing team members with getting up to date on the technicalities.
- Communicate the testing results to the stakeholders.

• What sort of tests would be worth describing or worth automating?

Answer:

- Repetitive tests.
- Tests that may involve operations that may be result in error when performed by a human.
- Tests that needs to be run with multiple sets of test data.
- Tests that deal with functionality that will be used frequently.
- Tests that may take the manual testing team a lot of time.
- Tests that may need to be run against multiple configurations like software, hardware etc.

• What tools would you use?

Answer:

- There are wide variety of tools out there. But, ReadyAPI from Smartbear will be a good fit as the tool can be used for functional as well as load testing APIs all within one tool.
- Opensource tools like REST Assured can be with Java and Eclipse to perform API testing.
- Visual Studio and C# can be used with a NuGet package called RestSharp to perform API testing.

Please select an endpoint to test and implement a test suite for that endpoint. You may choose a tech stack of your choice for the tests. Provide the URL of a public Git repo that contains the tests. Include documentation in your repository that contains your written answers to the questions above.

Answer: The endpoint that I choose to implement the test suite is: "repository_url": https://api.github.com/repos/{owner}/{repo}

2. Assume you are part of an engineering team that is building a loyalty app for a large retailer. You are in a meeting in which the following stories are being discussed by the product owner and engineering team:
a) As a customer, I want to enroll in the loyalty program.
b) As a program participant, I want to check my balance of reward points.
c) As a program participant, I want to redeem some of my points for a reward.
Describe how you might participate in this meeting to ensure that the development work for these stories can be demonstrated to the product owner. What are your areas of concern? How would you address them? Provide your answers as a PDF.
Answer: Actively take notes throughout the meeting and be prepared with any questions regarding the stories. Clarify the functionality so everyone is on the same page. Work closely with developers and business analysts throughout the development phase to ensure the product that they are building is as per the specs. If not, gather any concerns and get them corrected before the demo to the product owner.

An area of concern is when the decisions are made on the fly through email or IMs. Making sure there is a proper requirement specification document clearing stating the requirements for a functionality can help everyone stay on the same page. Sometimes the development of a story may not be what a product owner expected. In this case, the team should be able to explain why the product is developed differently – if there was any technical blockage, misinterpretation of requirements, making assumptions etc. To avoid this the entire team should be involved from the beginning. This way everyone can come together and have a chance to discuss and point out any ideas, concerns or clarify any questions.