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?

By itself, this documentation does not provide the business use of this technology. Without this perspective, we would be hindered in prioritizing work or evaluating severity of defects based on user impact. This is a lot of functionality, so prioritization would be critical.

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

I would work with product owners to document the needs that we are addressing. Based on the functionality planned for the next release (or Go Live), the work would have to be broken into user stories. At this point we could go through the process of fleshing out and understanding each of these pieces to determine testing flows/scenarios. As we make progress through these pieces, we would document the overall testing plan while considering all commitments/guidelines (coverage, entry/exit criteria, special requests, etc.) Once we are ready for execution, we can move through the standard phases (system test, integration, regression, user acceptance, performance.)

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

At a minimum, tests would need to be included to cover scenarios through all pieces of functionality in the next release (users, repositories, issues, etc.) Any tests that will be repeated (regression) or include many data permutations through the same process should be automated, but only if the design is finalized and the functionality is stable. If the functionality is not stable, then manual testing should continue until all major/blocker issues have been resolved.

• What tools would you use?

As this documentation is for a REST API, SOAP UI is a standard tool that would provide testing capability. Any limitations of SOAP UI could be addressed by building testing frameworks in Java or Python, but these should be limited to avoid the upfront investment and ongoing maintenance required.

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.

I would work with the product owner to document specific demo scenarios so that expectations can be fixed. I would ensure that the level of detail in these scenarios provide a clear process that the user is expected to follow and the expected outcomes.

• What are your areas of concern?

My main concern is that these stories do not provide any details of what the customer interactions would look like (mobile/SMS based). This may also require integration with an existing Point of Sale system, and these interactions would likely have to be mocked in a test environment.

• How would you address them?

The type of system being built should have already been addressed at this point, so we would move forward with this understanding (with mobile testing tools or whatever tools are required for the application.) Setting up these testing environments, mocks, tools, and other resources would have to be built into our plan (schedule/resourcing/cost) or planned/coordinated with the team that would provide this support (scheduling).