SOLUTIONS TO CHALLENGE 1.

Qn1. How many test cases will you need to adequately ensure the service acts as expected.
Please provide details about the organization of these test cases and what you will be testing.

Following the analogy of too much testing is a sin, too little testing is a crime.

There is no specific number, and we also can't write infinite number of test cases or any number of non-important ones, that's why there's a Review, by a Peer, or Business Analyst, or anyone else depending on the QA team structure, to make sure that the most important scenarios are covered, the ones that if they failed , will affect the main path and will cause issues of highest priorities to the customer. Our goal is to cover as much as we can (from the highest priorities to the lowest ones) in the specified time,
So what should we do, to make sure that we covered the most important ones at the specified time?
1- Write all high level test cases that may come into your mind
2- Specify the number of the most important test cases that could be written and executed in the specified time for testing
3- Ask the project owner, whether the other ones are important or not to be written and executed, with mentioning the extra time that will be needed to write and execute them
And in case he chose some of them, their time should be added to the needed time for testing.
4- Start writing test cases for all your scenarios, and if you thought about any others you may have forgot, you can return back to the project owner at anytime and ask him.

Organization of test cases

TestCase N01.Description	Test Data	Expected Result	Actual Result	Status (Pass or Fail)	Pre-Conditions	Assumptions

- Some structures include the scenario id, others don’t depending on the way test cases are organised.
- TestCaseId identifies a test case.
- Then the description gives detail about a given test case.
- test Data is data used during execution of that test case.
- Then the results we should expect after execution.
- Actual result is what we see after execution.
- Then status I.e: did the test fail or pass.
- Pre-condition describes what should be done for test execution and Assumptions are what we assume are already set up

QN2. Assuming this is a standalone NodeJs microservice. What technologies will you utilize
to set up automated tests for this. Please describe your setup in as much detail as possible.

I would choose cypress to be my testing framework using the page object model design pattern as explained below;

I would choose the Cypress framework which is a JavaScript-based end-to-end testing framework built on top of Mocha making asynchronous testing simple and convenient. It also uses a BDD/TDD assertion library and a browser to pair with any JavaScript testing framework.

The two main merits of using cypress are;

Automatic waiting – Cypress waits for the elements to become visible, the animation to complete,DOM to load, the XHR and AJAX calls to be finished.

Real-Time Reloads – Cypress is intelligent enough to know that after saving a test file, the tester is going to run it again. So, it automatically triggers the run next to the browser as soon as the tester saves their file.

Page Object Design Pattern

A page object is a class that represents a page in the web application. Under this model, the overall web application breaks down into logical pages. Each page of the web application generally corresponds to one class in the page object, but can even map to multiple classes also, depending on the classification of the pages. This Page class will contain all the **locators** of the Web Elements of that web page and will also contain **methods** that can perform operations on those Web Elements.

Code re-usability – The same page class can be used in different tests and all the locators and their methods can be re-used across various test cases.

Code maintainability – There is a clean separation between test code which can be our functional scenarios and page-specific code such as locators and methods. So, if some structural change happens on the web page, it will just impact the page object and will not have any impacts on the test scripts.

QN3

Explain the steps you would need to take to test the user behavior for this service in a live production environment. What manual testing do you conduct before a deployment to feel confident that the new code doesn't introduce any bugs.

Smoke Testing

The first manual test I would look out for is smoke testing also known as confidence testing, so basically smoke testing is a software testing process that determines whether the deployed software build is stable or not. Smoke testing is a confirmation for QA team to proceed with further software testing. It consists of a minimal set of tests run on each build to test software functionality.

Sanity testing which is also part of regression testing is a stoppage to check whether testing for the build can proceed or not. The focus of the team during sanity testing process is to validate the functionality of the application and not detailed testing. Sanity testing is generally performed on build where the production deployment is required immediately like a critical bug fix.

Integration Testing

This is mostly done to test data communication between modules, for example after signing up can I use that same data to login. So basically Integration Testing is defined as a type of testing where software modules are integrated logically and tested as a group. A typical software project consists of multiple software modules, coded by different programmers. The purpose of this level of testing is to expose defects in the interaction between these software modules when they are integrated

Acceptance Testing

Once the System Testing process is completed by the testing team and is signed-off, the entire Product/application is handed over to the customer/few users of customers/both, to test for its acceptability i.e., Product/application should be flawless in meeting both the critical and major Business requirements. Also, end-to-end business flows are verified similar to in a real-time scenarios.

End to End Testing

End To End Testing verifies complete system flow and increases confidence by detecting issues and increasing Test Coverage of subsystems. Modern software systems are complex and interconnected with multiple subsystems that may differ from current systems. The whole system can collapse by failure of any subsystem that is major risk which can be avoided by End-to-End testing.

QN 4.

You will also want to automate the testing of our mobile native applications for android and iOS in regards to how they use this service. Please share a plan about how you would go about automating these tests.

a. Which automation tools do you prefer to use in these situations?

I would use Appium for my mobile testing framework, I would also use the android debugging tool for the android application and also object page model design pattern.

Appium is an open source, and a cross platform Mobile Testing Tool for the hybrid and native iOS, it supports Android versions from 2.3 on-wards. Appium works like a server running in the background like selenium server. This mobile automation testing tool supports many programming languages, such as Java, Ruby, C# and other which are in the WebDriver library. Appium utilizes WebDriver interface for tests running.

Android Debug Bridge (adb) is a versatile command-line tool that lets you communicate with a device. The adb command facilitates a variety of device actions, such as installing and debugging apps, and it provides access to a Unix shell that you can use to run a variety of commands on a device. It is a client-server program that includes three components:

•A client, which sends commands. The client runs on your development machine. You can invoke a client from a command-line terminal by issuing an adb command.

•A daemon (adbd), which runs commands on a device. The daemon runs as a background process on each device.

•A server, which manages communication between the client and the daemon. The server runs as a background process on your development machine.

Page Object Design Pattern

A page object is a class that represents a page in the web application. Under this model, the overall web application breaks down into logical pages. Each page of the web application generally corresponds to one class in the page object, but can even map to multiple classes also, depending on the classification of the pages. This Page class will contain all the **locators** of the Web Elements of that web page and will also contain **methods** that can perform operations on those Web Elements.

Code re-usability – The same page class can be used in different tests and all the locators and their methods can be re-used across various test cases.

Code maintainability – There is a clean separation between test code which can be our functional scenarios and page-specific code such as locators and methods. So, if some structural

change happens on the web page, it will just impact the page object and will not have any impacts on the test scripts.

b. How do you ensure that we are testing on all devices relevant to our users.

I would look out for third party tools such as lambda tests and test project to test my application on multiple virtual devices.

QN5. Explain all of the cases you would need to test for to block fraudulent driver and customer behavior. Explain in detail how you would go about testing this behavior.

1 - Check for consistently identical rides at a specific time for over a long time, for example every day at 8:00 pm there these two riders with identical rides with same pickup and destination locations.

2- Check identical phone numbers between the passenger and the driver, some riders could pick themselves up for example after the day. They are paid for a trip to take them home.

3 – Check for when the rides are started, are they started at the picking up of the passenger or before the ride picks up the passenger.

QN6.

Assume that on the app during testing you see that a customer has a trip estimate of 20,000 UGX before they start the trip. However upon trip completion they receive a charge of 25,000 UGX for their ride. We call this a trip overcharge. Given the information about how we calculate trip estimates and costs come up with at least 5 reasons why you might see these discrepancies.

1- When the distance in kms covered is more than what the application had estimated which could come about by the rider using a different longer route diverting from the route shown by the application.

2- The application charges per minute, so in-case you find traffic jam on the way or when you are stopped for some time, this could cause discrepancies in the final charge being different from the estimate.

3- The customer may order the ride a few minutes before the peak hours and the ride is started when we are in peak hours so the charges here will be for peak hours.

4- For trips longer than 30km, the premium charged is higher. So if the destination picked is below 30km and then the customer lengthens the trip beyond 30km during the ride, the app will give an over charge.

5- There could also be an overcharge due to rounding off to the nearest 500 shillings.

QN7.

Create a tool that will allow you to take in a user's pick up and drop off GPS coordinates and return the trip estimated price for the user given the information you have been given. This can be done using any tool such as MS Excel, a software program such as Node JS or PHP, or any other calculation tool you feel comfortable using. We suggest using the Google Maps API to help you with this.