

QA Assignment - Bruna Gaino Lipovscek

The Task

We're excited to announce an opportunity for a QA analyst. We specialize in exploring innovative niches and projects and one of our most exciting ventures has been developing an electric vehicle platform. Your goal is to analyze the document outlining the test cases for a web platform designed to introduce and promote electric vehicles while providing a seamless onboarding experience for users.

Application overview and deliverables

• Test Case Development

1. **Vehicle Selection:** Users can select their current gasoline vehicle and compare it with various electric vehicle models.

Task: Develop two test cases for the Vehicle Selection feature (Example: Verify that the user can select a vehicle from the provided list.).

Scenario: Valid Vehicle Comparison	Given that I am on the vehicle comparison screen When I select a valid gasoline vehicle And I click the "compare" button Then the system displays the vehicle comparison result
Scenario: Invalid Vehicle Comparison	Given that I am on the vehicle comparison screen When I do not select a valid gasoline vehicle And I click the "compare" button Then the system shows an error message saying "you need to choose a vehicle to compare correctly"

2. **Driving Habits Input:** Users input their average daily driving distance and fuel efficiency.

Task: Develop two test cases for the Driving Habits Input feature (Example: Ensure that the system accepts valid numerical inputs for daily driving distance).

Scenario: Driving Habits Input Successful	Given that I am on the driving habits screen When I enter a valid average daily driving distance And I enter a valid fuel efficiency And I click the "calculate" button Then the system displays the result of my driving habits.
Scenario: Driving Habits Unsuccessful	Given that I am on the driving habits screen When I enter a valid average daily driving distance And I enter "eight miles" in the fuel efficiency field And I click the "calculate" button Then the system displays an error message saying "only numbers are valid in the 'fuel efficiency' field"

• Key Testing Points Identification

1. **Cost Calculation:** The platform calculates and compares the annual fuel costs of the current vehicle versus the selected electric vehicle.

Task: What are the main aspects you would validate when testing the Cost Calculation feature? (Hint: Consider the accuracy of calculations based on user inputs)

The first validation I perform is to ascertain the prices of gasoline (MPG) and energy (kWh) that the system uses for the cost calculation. This is crucial because it needs to align with the current prices the user pays for gasoline and energy in the country/state/city where the vehicle is/will be most frequently used.

The second validation I perform is to create a spreadsheet with data that the user might input (whole and decimal numbers) to calculate the annual fuel costs. I then verify if the system's resulting calculation matches the calculation derived from the spreadsheet data.

2. **Savings Estimation:** Provides an estimate of potential annual savings when switching to an electric vehicle.

Task: Identify two potential issues when testing the Savings Estimation feature (Hint: Think about data accuracy and result presentation).

As addressed in the previous question, regarding data accuracy, it's essential to know the prices of gasoline (MPG) and energy (kWh) that the system is using to calculate the potential annual savings. This is because it needs to align with the current prices the user pays for gasoline and energy in the country/state/city where the vehicle is/will be most frequently used.

The first issue would be the potential annual savings result appearing as a negative value; in other words, instead of being a saving, it would indicate a waste. In this scenario, how would this information be displayed to the user?

The second issue would be the potential annual savings result being a very large number, with many decimal places. It would be beneficial for the value to be rounded, making it a more "comfortable" number to display to the user and preventing it from breaking the layout.

• Implementation Planning

1. Assuming the development team plans to implement the platform in two phases, how would you suggest dividing the implementation to facilitate testing? (Hint: Consider starting with the vehicle selection and driving habits input features).

My suggested implementation:

1. **First phase - vehicle selection and driving habits.** Why? First, this is the initial screen the user will interact with in the system. Secondly, by implementing these two features first, I will have real data to test the next two features, without needing mocked data.
2. **Second phase - cost calculation and savings estimation.** Why? I will leave the two calculation features for later, ensuring a more mature registration step and enabling calculations with real data.

• Critical Analysis and Improvements

1. Based on the described features, would you add any tests or validations not mentioned? Explain your choice. (Hint: Think about user experience and data validation)

I believe it's possible to verify if there's a "save progress" button between steps, in case the user tries to exit without completing the information (so they can return later and finish). Similarly, it's good to check if "back" and "forward" buttons function correctly. Each feature would need to be easy for the user to understand, with clear and explanatory "success" and/or "error" messages.

In the "driving habits" feature, I would test other data that might be incorrectly entered, besides the "eight miles" example I described, such as special characters. I think it would also be interesting to validate the data the user inputs, in cases where a number is very large, equal to zero, or a negative value, among others.

When "selecting the vehicle", the list might be very long, and if it takes too long to load, it could create a broken expectation for the user, or even cause a system crash, loading failure, etc.

2. What problems could arise from using the data entered by users for calculations? How would you prevent these problems? (Hint: Think about whether the user might enter incorrect or incomplete information)

Well, the user could input text instead of numbers (I used "eight miles" as an example), special characters, too many decimal places, very large numbers, zero, negative numbers, or incorrect decimal formats (e.g., using a comma instead of a period). In these cases, we can perform tests to verify all possibilities and ensure the system provides clear error messages to the user, indicating that the entered data is inappropriate (and it's also necessary to show, in the layout, the correct and expected format for the user to input the requested data, preferably using practical and comparative examples). We need tests to verify if the user has filled in all fields correctly or if any fields were left blank.

We can check if it's possible to implement input masks or real-time validation in the system to prevent the user from typing invalid data. I also believe the system needs to verify the unit of measurement and currency type the user employs based on their location and where they use/will use the vehicle: km/l, MPG, kWh, US\$, R\$, etc. All of this influences the final calculation and makes it relevant to the user. Finally, we need to pay attention to the layout, ensuring fields are correctly named, sufficiently explanatory so the user understands how data should be entered correctly, where data needs to be entered, a user-friendly identification of steps, among other aspects

• Automation Process

1. Considering that the team is creating regression automated tests for the platform, describe how you would automate these tests for the key features mentioned above (Hint: Consider mentioning which features should be automated and why).

Okay. Let's perform automated regression tests on the system. I believe the main features to be tested, in order of criticality, would be:

1 - Cost calculation and savings estimation: because they involve calculations/financial calculations, which are important information the user wants to know, and this will influence their decision to replace their gasoline car with an electric car. If the user sees that the calculations were done correctly, the system will provide more credibility and confidence for them to make the best decision. Automating regression tests here would prevent many future bug issues.

2 - Vehicle selection: here I think the biggest problem is a performance issue (imagine the user clicking on the vehicle selection and, because it's a very long list, it takes ages to load?). It needs to be performant. In a way, this list is the gateway for the user to proceed with the simulation or not. An automated test here can facilitate checking vehicle selection, if the loading is too heavy, if it's working correctly, etc.

That said, regarding the automation itself, I've already done a small login automation project using Python+Selenium, and I believe it would be possible to adapt some things to automate this test. For me, it would be easier to start with the user interface. I know you can also use JavaScript+Cypress, Java+Selenium... it's necessary to see what the best architecture is for the team and the project.

I also understand that it's important to have different levels of testing. In my research on automated regression tests, I've seen that there is an opportunity to do integration tests, as they verify if the system works as a whole, ensuring that the system's features work together and that the system meets the end-user's needs.

Previously, regarding the cost calculation question, I suggested using a fixed data spreadsheet for initial calculation validation. With automation, it's possible to use generated test data or a set of data from a test database, which will help create more diverse and varied test scenarios.

Ultimately, an automated regression test brings many benefits, such as long-term time savings, speed, reliability, and bug detection before they cause problems or major problems.