Veenila Singarayani

[veenilasinga@gmail.com](mailto:veenilasinga@gmail.com)

# Assignment:

Article type: Project

STEP 1 : Manual QA tasks

This assessment consists on completing three tasks similar to the work we expect you to do on a manual QA testing assignment.

Requirements

1. Create high level Test Plan for "Rent a car" functionality of <https://www.lyft.com/rider/rentals>.
2. Describe in details at least 2 the most critical scenarios (1 positive and 1 negative) to test “Rent a car” form functionality of <https://www.lyft.com/rider/rentals>.

How to submit

Please upload the the document(s) you have created with your deliverables to any cloud storage service of your preference (GDrive, Dropbox, etc) Make sure to give open read access permissions. Paste the link on the box bellow.

Important: after finishing this assessment you need to click on the submit option.

Thank you for your time on this assessment and we look forward to speaking with you.

# Solution:

**Test Plan for "Rent a Car" Functionality - Lyft**

**Lyft website**: https://www.lyft.com/rider/rentals).

**Objective:** To ensure that users can successfully rent a car through this feature while also validating that the system handles negative scenarios gracefully.

**Scope**

The testing scope includes the "Rent a Car" feature accessible on the Lyft website.

**Positive Scenario**: Successful completion of the car rental process.

**Negative Scenario**: Handling of errors and issues during the car rental process.

**Test Environment**

Browser: Chrome, Firefox, Safari, Edge

Operating System: Windows 10, macOS, iOS, Android

**Test Scenarios**

Positive Scenario: Successful Car Rental

Precondition

* The user has a Lyft account.
* The user is logged in to their Lyft account.

Test Steps

* Navigate to the Lyft website (https://www.lyft.com/rider/rentals).
* Click on the "Rent a Car" option.
* Fill in the required information:
* Pickup location
* Pickup date and time
* Drop-off location
* Drop-off date and time
* Select a car type from the available options.
* Review the rental details (pickup, drop-off, car type, and estimated cost).
* Click the "Rent Now" button.
* Verify that the user is redirected to a confirmation page.
* Confirm that the rental details match the selected options.
* Confirm that the estimated cost is displayed correctly.
* Complete the rental process by providing any additional information required (e.g., payment information).
* Verify that the user receives a confirmation of the rental.
* Check for any email confirmation sent to the user's registered email address.

Expected Results

* The user can complete the car rental process without encountering errors.
* The rental details match the selected options.
* The estimated cost is accurate.
* The user receives a confirmation of the rental via the website and, if applicable, via email.

**Negative Scenario**: Handling of Invalid Input

Precondition

* The user has a Lyft account.
* The user is logged in to their Lyft account.

Test Steps

* Navigate to the Lyft website (https://www.lyft.com/rider/rentals).
* Click on the "Rent a Car" option.
* Fill in the required information:
* Enter an invalid pickup location (e.g., an address that does not exist).
* Enter a valid pickup date and time.
* Enter a valid drop-off location.
* Enter a valid drop-off date and time.
* Attempt to proceed with the rental by clicking the "Rent Now" button.
* Expected Results
* The system should detect the invalid pickup location and display an error message.
* The user should not be able to proceed with the rental until valid pickup location information is provided.
* Test Deliverables
* Test Plan document.
* Test cases based on the outlined scenarios.
* Test execution results, including screenshots and error logs if applicable.

Exit Criteria

The "Rent a Car" functionality will be considered tested and ready for release if the following EXIT CRITERIA are met:

* All test cases are executed and passed.
* Critical defects are identified, documented, and resolved.
* The user receives appropriate confirmation and notification upon successful rental.
* The system handles invalid input gracefully by displaying error messages.
* Risks and Assumptions

Assumptions:

The Lyft website and "Rent a Car" functionality are in a stable and testable state.

Risk: Network or server issues may affect the test execution and result in false positives/negatives.

Contingency and Mitigation plan: Report and work with relevant team to resolve the network issue

Sign-off

The Test Plan is approved and can be used as a basis for testing the "Rent a Car" functionality on the Lyft website.

**Defect Severity Definitions**

Critical (Severity 1)

Description: Critical defects render the application or feature unusable or severely impact core functionality.

Examples: Application crashes, data loss, security vulnerabilities, or complete feature failures.

Impact: High impact on users, business operations, or security.

Major (Severity 2)

Description: Major defects affect major functionality but do not render the application completely unusable.

Examples: Significant data inaccuracies, major UI issues, or functional features not working as intended.

Impact: Significant impact on user experience and business operations.

Moderate (Severity 3)

Description: Moderate defects have a noticeable impact on the application but do not disrupt major functionality.

Examples: Minor UI issues, inconsistencies in application behavior, or non-critical data inaccuracies.

Impact: Moderate impact on user experience or business operations.

Minor (Severity 4)

Description: Minor defects have minimal impact and typically include cosmetic issues.

Examples: Spelling errors, minor UI alignment issues, or non-impacting visual discrepancies.

Impact: Low impact on user experience or business operations.

**Defect Priority Definitions**

Immediate (Priority 1)

Description: Immediate priority defects require immediate attention and should be fixed and tested as soon as possible.

Examples: Critical defects that impact core functionality or security vulnerabilities.

Action: Start working on the defect immediately and prioritize it over other tasks.

High (Priority 2)

Description: High priority defects are important but do not require immediate attention. They should be addressed as soon as possible after critical defects.

Examples: Major defects that impact major functionality.

Action: Address the defect promptly, but it can wait briefly if critical defects are present.

Medium (Priority 3)

Description: Medium priority defects have a moderate impact and can be addressed in the normal development cycle.

Examples: Moderate defects that do not disrupt core functionality.

Action: Include the defect in the regular development and testing cycle.

Low (Priority 4)

Description: Low priority defects have minimal impact, and there is no urgency to address them.

Examples: Minor defects that do not significantly affect user experience.

Action: Address the defect when time allows, but it's not a priority.

-----------------

I would also include the below points in the test plan, for Automation perspective:

1. Test Automation Framework Selection

2. Test Environment Setup

3. Test Data Management

4. Automation Test Strategy

5. Test Script Development

6. Test Script Maintenance

7. Test Script Execution

8. Reporting and Logging

9. Parallel Execution

10. Test Data Cleanup

11.  Exception Handling

12.  Logging and reporting.

13.  Integration with Test Management Tools: For test case management, test execution tracking, and defect reporting.

14.  Test Environment Configuration Management:

15.  Test Data Security