**Rent a Car Test Plan**

Prepared By: Shaveta Dhingra

1. **INTRODUCTION**

The purpose of this document is to provide a high-level test strategy for "Rent a Car" functionality in a Lyft-like Car Rental App. It will cover various aspects to ensure the feature's reliability, usability, performance, security, and compatibility among other factors.

1. **TEST PLAN OBJECTIVE**

"Rent a Car" feature should allow users to easily rent cars, providing a flawless, efficient, and secure experience.

1. **SCOPE**

In this test plan we consider project flow, test cases and breakage points. Testing will cover the end-to-end process of renting a car, from vehicle selection to rental confirmation, including user interface, backend services, payment processing, and notifications.

1. **TESTING STRATEGY**

**Functional Testing:**  
  - All UI elements on the "Rent a Car" pages  
  - Integration with the backend for fetching available cars  
  - Correct calculation of rental prices, including handling of discounts  
  - Payment gateway integration  
  - Email and SMS notifications upon successful rental  
**Usability Testing:**  
  - Ease of navigation through the car rental process  
  - Clarity of instructions and information provided  
  - Responsiveness of customer support features  
**Performance Testing:**  
  - Load time of the "Rent a Car" feature under different network conditions  
  - Backend response times when querying available vehicles  
**Security Testing**  - Data encryption of personal and payment information  
  - Authentication and authorization mechanisms  
**Compatibility Testing:**  
  - Cross-browser testing on Chrome, Firefox, Safari, and Edge  
  - Mobile responsiveness testing across iOS and Android devices

1. **ENVIRONMENT REQUIREMENTS**

What environments are being used?

* It will be tested in staging environment first.
* After passing all the scenarios, it will be tested in pre prod.

1. **TEST SCHEDULE**

Testing is an iterative process. Testing needs to happen at every code check-in if possible.

What are some test milestones?

* A Rent Car test suite needs to exist for each story, and executed during the QA step on the Sprint board
* Three milestone blitz sessions to be organized through the release

What is going to be tested during each milestone?

* Happy path
* No regression introduced in all relevant code paths changed
* Negative test cases
* Edge cases with support team

When are features going to be delivered and ready for testing?

* Feature availability throughout release as work becomes available

1. **CONTROL PROCEDURES**

Problem Reporting: What happens when we run into problems while testing?

* We create bug tickets after discussing the steps and symptoms with the team.
* Bugs get prioritized by the team Product Manager

*Change Requests: What is the process of requesting modifications?*

* A ticket is created against our team.
  + The Change request gets prioritized by our Product Manager

**Bug Reviews**

* During Sprint Grooming bugs are reviewed
* Once in production, for major bugs a Root Cause Analysis is prepared
* Action items based on the review will be added to the Sprint Retrospective event

1. **FEATURES TO BE TESTED**

* Vehicle Selection interface
* User navigates to the "Rent a Car" section.
* User selects desired pick-up location, date, and time.
* User selects drop-off location, date, and time.
* The app displays available vehicles for the selected time and location.
* User selects a preferred vehicle based on size, price, or other criteria.

1. **APPROACH**

* Automated tests for repetitive tasks like UI interactions and backend requests
* Manual testing for usability aspects and complex scenarios not covered by automation
* Security penetration tests to identify potential vulnerabilities

1. **RESOURCES, ROLES & RESPONSIBILITIES**

Who is responsible for managing, designing, preparing, executing, and resolving the test activities as well as related issues?

* Dev Engineers
* QA Engineers

Who is responsible for providing the test environment?

* + QA for integrated staging environments
  + Scrum Teams are responsible for spinning test orgs for User Stories testing

What resources are needed for testing?

* + Staging environments up & running for testing and blitzing
  + off core services are available for testing
  + on-core services integrated in org for testing

1. **DEPENDENCIES**

What are the dependencies (significant constraints on testing, test-item availability, testing-resource availability, deadlines, etc.)

* + off-core services
  + on-core services connectivity and availability
  + Personnel, Software, Hardware, Test Data, etc.

1. **RISKS AND CONTINGENCIES**

* Risks will include third-party service downtimes (e.g., payment gateways) and plan for alternative testing strategies.

1. **TOOLS**

**Automation Tools**

* End-to-End Selenium/Appium Automation Framework
* Jenkins with slack alerts

**Bug Tracking**

* JIRA

1. **APPROVALS**

Who will review and approve this Test Plan?

* + QA automation workgroup
  + Async project lead