**TEST PLAN DOCUMENT**

**App – Lyft Rider Rentals App**

**Scope**

All Testable entities defined within the scope and functionality of lyft app is considered. Critical functions like ride booking, Booking history check, Maps and location specific features, payment etc are considered in testable scope.

**In-Scope**

1. Ride Search
2. Ride filters as per location
3. Maps and Geolocation
4. Filtering of results
5. Booking
6. Payment and Confirmation
7. Booking History
8. User Accounts and authentication

**Out of Scope**

1. Non-functional aspects like UI refresh rate
2. Page load times

Non-Functional Tests are Excluded from current testable scope.

**Objective:**

Objective of the Test is to determine that all In-Scope features of the application is working as expected and could be concluded with positive and negative test evidences as required by the business function.

**Roles and Responsibilities**

Test user profiles with simulated locations and categories are to be created to replicate close to real production scenario. Few of the roles to be dictated within the application as per test scope are.

1. Test User – 20 Different Profiles filtered based on Location
2. Same Test User accounts with simulated Payment options
3. Admin User to checkout admin functions (if any)

**Test Methodology:**

**Overview:**

Test methodology to be followed includes Agile – Scrum approach with distinct User Stories Structured as per unique features to each Test Team member. All Testable items are to be recorded within Organization strategic Dashboard or reporting platform i.e. JIRA in this case.

Team to follow Spring planning where backlog grooming sessions are going to include individual features as decided and agreed within planning poker sessions with the wider team.

**Test Levels:**

Test Levels to include below phases planned as per the JIRA Swim Lanes.

1. Unit Testing – During Feature Dev cycle
2. Integration Testing – To be performed at individual feature level
3. User Acceptance Testing – to be performed by Business POC
4. Pre-Production Test – Sanity by QA and Business
5. Production Checkout – Solely performed and agreed by business during non-business hours.

**Bug Triage:**

Bug Triage and regulation calls to be scheduled by QA POC with wider team on weekly basis to demonstrate bugs and resolve any blockers.

**Test Completeness:**

Test conclusion will be observed once all testable entities with its associated features are marked complete and all associated bugs are closed.

**Test Deliverables:**

1. All QA Test Cases Passed Report
2. Defect Density and phase report
3. QA Sign-off on a feature basis

**Test Resources:**

Test resources would include any direct or indirect assets associated with the Test conclusion process. In this case Testable assets would include application Test Environment, Test Management Tools etc.

Test Resources include:

1. QA App Instance
2. Test Tracking and Bug Management Tool (JIRA in this case)
3. Test Entitlements and Technical Accounts
4. Any other associated assets from Stakeholders.

**Test Timelines**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Phase** | **Start Date** | **End Date** | **Sign-Off** | **Remarks** |
| Unit Testing | 13/08/2023 | 20/08/2023 | Mohammad | NA |
| Integration Testing | 21/08/2023 | 4/09/2023 | Mohammad | NA |
| User Acceptance Testing | 05/09/2023 | 28/09/2023 | Mohammad | NA |
| Production Checkout | 30/09/2023 | 30/09/2023 | Mohammad | NA |