

Testing checklist: ride-hailing app

Source: <https://howtotestaproduct.com/how-to-test/ride-hailing-app/>

Generated: 2026-10-07. Licence: CC BY 4.0.

☐ 1. release smoke tests

Done when: On every build, a test rider requests a ride, a test driver accepts it, and both apps reach trip complete without a crash.

Risks: store-rejection. Tool class: mobile automation. Usually run by: QA engineer.

☐ 2. penetration testing

Done when: A tester cannot read another rider's trips, cannot reach driver documents from a support or contractor account, and cannot log a trip with a spoofed location.

Risks: data-breach, fraud-abuse. Tool class: proxy. Usually run by: security specialist.

☐ 3. exploratory testing

Done when: A session log lists attempts at fake trips, promo stacking and account takeover with the outcome of each, and every successful attempt has a ticket.

Risks: fraud-abuse. Tool class: test management. Usually run by: QA engineer.

☐ 4. compliance checks

Done when: A data map shows where each driver document type is stored, with the transfer basis for any copy outside its home region, and store privacy labels match it.

Risks: regulatory-penalty, store-rejection. Tool class: test management. Usually run by: product manager.

☐ 5. end-to-end functional tests

Done when: A scripted trip with cancellation, wait time and a tip produces a receipt where each line matches what the rider was shown before confirming, and a rider flagged as needing boarding time sees no wait-time line.

Risks: accessibility-liability. Tool class: mobile automation. Usually run by: QA engineer.

☐ 6. accessibility testing

Done when: A screen reader user books, tracks and pays for a ride on both platforms, and a rider flagged as needing boarding time is not charged wait-time fees.

Risks: accessibility-liability. Tool class: accessibility checker. Usually run by: accessibility specialist.

☐ 7. security scanning

Done when: The scan report for the rider and driver APIs has no open high findings before the build ships.

Risks: data-breach. Tool class: SAST scanner. Usually run by: security specialist.

☐ 8. performance and load testing

Done when: At the forecast New Year's Eve request rate, ride requests are matched, priced and billed with no rise in error responses on the status dashboard.

Risks: outage-under-load. Tool class: load generator. Usually run by: developer.

☐ 9. device and browser compatibility

Done when: Request, track and pay complete on the oldest supported phone and OS for riders, and accept and navigate complete on the oldest supported driver device.

Risks: store-rejection. Tool class: device cloud. Usually run by: QA engineer.

☐ 10. data integrity and migration testing

Done when: Every driver deactivation in a test week traces to a named human reviewer, and a sample of fraud flags rechecked by hand matches the recorded reason.

Risks: wrong-decisions. Tool class: data validation. Usually run by: QA engineer.