

Q1 - Please identify your testing strategy and approach to testing this application in an Agile development environment. Please list down any assumptions and questions you have about the application brief.

Testing can be done :

1. Component Tests (Payment Gateway, distribution system)
2. Integration Tests
3. E2E business automated tests (Positive flows)
4. FrontEnd Tests (Negative / Field level UI validations)
5. Performance Tests

In Agile, if the product is developed in iterative fashion (sprint by sprint) say

1. Sprint-1 – Goal : Simple Guest user flow checking Distribution system and getting desired results
2. Sprint-2 – Goal : Simple Guest user flow checking Distribution system and completing payment with notification
3. Sprint-3- Goal : Login with existing user and complete Booking
4. Sprint-4- Goal : E2E Full product completion

Testing also will be progressing in Sprint fashion, by focusing testing on the smaller chunk developed in a particular sprint.

In this way both development and testing continuously progress throughout and help to deliver the product with outstanding quality.

Questions

- Any ways to retrieve booking and change passenger details / dates ?
- Card / Payment details to be used for testing ?
- Browsers to be tested ?
- Any API tests can be integrated for Galileo Global distribution system or Business Level E2E tests preferred?

Q.2 - Please list down test scenarios you think will cover the majority of application features. Mention any negative, positive and edge cases based on the application details.

UAT Positive Scenarios

1. Complete End-to-End booking flow as GUEST user
2. Register as NEW user & login with new user, complete End-to-End Booking flow
3. Block ticket for 24 hours and check its availability after 24 hours (Guest / Registered User)
4. Retrieve booking history and check for its details
5. Check notifications received in a booking flow (Email / SMS)
6. Session Logout / Browser relaunch checking

Few Negative Scenarios to mention are :

1. Invalid credit card / payment details
2. From / To fields selected of same venues
3. Choosing Return date less than departure date (round trip)
4. System behaviour when Galileo Distribution system is down
5. Login multiple attempts with wrong credentials
6. Passenger details related scenarios (child not accompanied with adult ..)
7. Notification latency tests

Q-3 :Considering this application has a number of third-party API integrations, how would you go about testing the system?

I would suggest use Service Virtualisation Testing techniques, for instance, WireMock is a very good candidate for mimicking Third party API testing with controlled request and response behaviours

Q4- What will be your automating strategy, approach and choice of tools?

Automation can be done extensive in component level (Distribution system, Payments) and UI business flows for above mentioned scenarios can be automated in different browsers.

Tools for API – Java-RestAssured / WireMock

Tools for UI – Selenium Java