Q.1 - 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.

Agile has proven its credentials, especially for modern day responsive applications such as Acme Flight booking. It allows you to response effectively & efficiently to the continuous changes not only from UI perspective but also from functionality perspective as well.

My strategy has always been around

* Categorise
* Prioritise
* Time box

For this reason I’ll break down the testing domain into the following categories which can be further subdivided

1. UI Testing
2. Functionality Testing
3. End-2-End Testing
4. Integration Testing
5. Non-Functional Testing

**Assumptions:**

* There are only two third parties involved in the booking system – namely CBA for payments & Galileo Global distribution System for flight details.
* Mobile browser view is optimised same as desktop.

**Questions:**

* Is there any local database linked to the application e.g. for User details or anything else?

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

I usually categorise the functionality and build the test suites upon the

**Domestic Flights**

* + Flight Search Trip Type – One way, Round, Multi city etc
  + Verify Bookings
  + Verify Add ONs e.g. Hire Car
  + Verify Payment Options

**International Flights**

* + Trip Type – One way, Round, Multi city
  + Verify Bookings
  + Verify Add ONs e.g. Hire Car
  + Verify Payment Options

**User Details Testing**

**Registered User**

* + Details are appearing correct
  + Details can be updated.
  + Verify user credentials.

**New User**

Verify a new user can be registered

**Cross browser Testing**

* + Verify AUT behaviour in different browsers.
  + Verify AUT behaviour in different View-ports.

**Specific Negative Testing e.g.**

* Verify when one of the third party APIs are not available.
* Verify session timeout.
* Verify Error Messages
* Payment is refused.

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

We’d need to mock the responses for third part API integrations

* Galileo Global distribution System
* CBA Payment Gateway

At integration testing, a set of scenarios must be executed with the real APIs.

Q.4 - What will be your automating strategy, approach and choice of tools?

I’ll be employing Agile Methodology, Considering for a functionality on the basis of

* Criticality
* Priority
* Delivery Time

|  |  |  |  |
| --- | --- | --- | --- |
| Testing Type | Testing tool | Project | Comments |
| UI Testing | Selenium, Cucumber | Java, Maven, Test NG | Cucumber - If BDD approach is needed |
| API Testing | Rest Assured | Java, Maven, Test NG |  |
| Contract Testing | Pact.io |  |  |
| Visual Regression Testing | Percy, Applitools, VRT | Java, Maven, Test NG | Depending on the budget & requirements any of the tools can be used. |
| Non-functional Testing |  |  | Exploratory Testing, System Availability Testing etc. |