Qantas QA Test

***Task 1 - Case Study***

Acme Airlines is planning to launch a new web portal for flight bookings. This portal will be a responsive web application. It will allow users to book domestic and international flights as a guest as well as allow users to register and store user information for ease of booking in future.

This application will require using Galileo Global distribution System to manage bookings. Galileo doesn't provide a Sandbox environment and only have a production environment to work with.

CBA is partnered to act as a payment gateway. Payment service will be integrated along with flight booking system.

***Questions:***

**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.

**A.1** - By the description , I believe that , we have three systems of interaction.

1. UI – Web Portal for flight booking
2. Payment gateway- CBA
3. Database- Galileo Global distribution System

**Assumptions:**

1. For UI and Payment Gateway, we have a sandbox environment.
2. Only Galileo System is of production as mentioned in the case study.
3. And they are interacting like the below diagram.

If Payment Gateway API gives success response

Web Portal

Galileo Database Distribution System

(3)

Request (1) Response(2)

Payment Gateway  
API

As mentioned in the case study we are storing booking and payment information of the user in Galileo System. The Galileo API will only hit when payment is successful else the end user will see an error message that “Payment Is Not Successful”.

1. I assume that I have APIs for both Payment Gateway and Galileo Database.

**Working:**

When user successfully enters data for flight booking and make payment then it will hit Galileo API. But we can’t test that as it is a prod database. In that case (as per testing point of view) we can’t check for positive scenarios in Galileo because it’s the production system, but we can check for negative scenarios. Testing Approach -API testing only.

However, we can check interaction between Web Portal and Payment Gateway as we have Sandbox environment for both, so we can check positive and negative scenarios. We can only test positive scenario if the Galileo system is not integrated in the lower environments. In this we case testing approach we follow- API and UI testing.

**Questions:**

* Could we use Test Data in the Production?
* Is there any stubbed API available for Galileo system?

**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.

**A.2** - **UI Testing : only possible when Galileo is not integrated**

Positive Scenarios

* Registration Success.
* Successful Login for registered user.
* Booking confirmation for registered and unregistered users.
* Payment Success (Message display-payment successful) in lower environment
* Edge cases- Assuming for single login user can do bookings for 1(min)-6(max) people

Try to book for 5 (accepted)

Try to book for 1(accepted)

Negative Scenario

* Registration unsuccessful (due to some invalid entry of data like: wrong email, contact no. etc)
* Payment Failed (error message display)
* Edge Cases: Try to book flight for 0 (for single login)

Try to book flight for 7 people (for single login)

**API Testing:**

Positive scenarios

* Payment gateway Sandbox API response success (status-200)

Negative Scenario

* Payment gateway Sandbox API response status 400(bad request) and 500(server error) or 504(Timeout).
* Galileo Database API responses 400(bad request) and 500(server error)

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

**A.3** - As per the case study, there are two third party integrations- Payment Gateway API and Galileo System API. As already mentioned, we don’t have the sandbox environment for Galileo System so we can’t check for positive scenarios i.e. we can’t do the UI testing for it. However, we can definitely check the negative scenarios by giving incorrect payment info using postman. For payment gateway I assume we will have the sandbox environment so I can test sandbox payment gateway API using postman and check for both positive and negative scenarios. Also, if in our web portal’s lower environment Galileo system is not connected then I can do the UI testing as well in qa or staging environments.

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

**A.4** - API testing and UI testing, Tools Used – POSTMAN, Selenium, Java, NFT tools (Non-functional testing tools like Jmeter but can be possible when Galileo is not integrated)

API Testing to check interaction between three systems UI-Payment Gateway- Galileo System for only Negative test cases.

API and UI testing for Web portal- Payment Gateway, we can check both positive and negative scenarios (Positive scenarios only when Galileo is not integrated).

***Task 2 - Tech Exercise Guidelines***

***UI Automation***

1. Navigate to https://www.qantas.com/au/en/travel-info/check-in.html
2. Assert the title of the page
3. Type in Booking or voucher reference number as “QWE456” and LastName as “LastName”
4. Click on continue
5. Assert the error message “It looks like something went wrong there”

*Folder attached with name* ***Qantas\_TestingTask***

***API Automation***

Write API tests using **Postman tool, export and send the script** for flightsearch api

- This api will return all flights between two destinations (Sydney to Melbourne), POST request will create a sessionid and this sessionid should be passed in GET request to get all the flights and price between two destinations(Sydney to Melbourne),

 More details in , <https://rapidapi.com/skyscanner/api/skyscanner-flight-search?endpoint=5a9b572de4b06ec3937b1296>

1. POST <https://skyscanner-skyscanner-flight-search-v1.p.rapidapi.com/apiservices/pricing/v1.0> - This endpoint will return sessionid in response header(bold in example) with response code 201.

**Example**: http://partners.api.skyscanner.net/apiservices/pricing/hk1/v1.0/**699401ed-8917-4027-a236-a5d6af6cfe7e**

**Header:**

x-rapidapi-host:skyscanner-skyscanner-flight-search-v1.p.rapidapi.com

x-rapidapi-key: [Sign up in <https://rapidapi.com/skyscanner/api/skyscanner-flight-search?endpoint=5a9b572de4b06ec3937b1296> to get the key]

Content-Type:application/x-www-form-urlencoded

**RequestBody:(form-urlencoded)**

inboundDate:2019-12-20

cabinClass:business

children:0

infants:0

country:AU

currency:AUD

locale:en-AU

originPlace:MEL-sky

destinationPlace:SYD-sky

outboundDate:2019-12-01

adults:1

1. GET: [https://skyscanner-skyscanner-flight-search-v1.p.rapidapi.com/apiservices/pricing/uk2/v1.0/**699401ed-8917-4027-a236-a5d6af6cfe7e**](https://skyscanner-skyscanner-flight-search-v1.p.rapidapi.com/apiservices/pricing/uk2/v1.0/699401ed-8917-4027-a236-a5d6af6cfe7e)**-**This is the sessionid that is from POST request, - Response will be in JSON,

**Header:**

x-rapidapi-host:skyscanner-skyscanner-flight-search-v1.p.rapidapi.com

x-rapidapi-key:9 [Sign up in <https://rapidapi.com/skyscanner/api/skyscanner-flight-search?endpoint=5a9b572de4b06ec3937b1296> to get the key]

Expected Result: Get all carriers between Sydney and Melbourne that is returned in the response

**Sample Response:**

"Carriers": [

{

"Id": 1825,

"Code": "VA",

"Name": "Virgin Australia",

"ImageUrl": "<https://s1.apideeplink.com/images/airlines/V1.png>",

"DisplayCode": "VA"

},

{

"Id": 1606,

"Code": "QF",

"Name": "Qantas",

"ImageUrl": "<https://s1.apideeplink.com/images/airlines/QF.png>",

"DisplayCode": "QF"

}

]

*File Attached having with* ***Qantas\_Test.postman\_collection***