

QA Technical Challenge

Question 1

Given the following user story please write down the 6 main test cases you could think of:

“As a non registered user, I want to search for a product”

Note: We want you to focus on the search feature

Please use this site, <https://www.manomano.co.uk/>

User perform a search

Given 'Term' term 'result' results at Manomano website

When User type 'Term' at search field

And press enter

Then website goes to results screen and 'results' results

Examples:

| Term | Result |
|----------|-------------------|
| Table | has more than 500 |
| Siena | has 158 |
| asdfasdf | doesn't have |

User can see extra information when type a term

Given Table term has extra information related

When User type Table at search field

Then below search field web shows suggestions section with Table related options

And categories section with Table related options

And Ideas and Advices section with Table related options

User can interact with search suggestions

Given Table term has suggestions

When User type Table at search field

And click over the first suggestion

Then website go to search results screen

And results are related to the suggestion

User can erase previous search term

Given User already search for a term

When User type an already used term at search field

And press erase button from previous search

Then term is deleted and won't be shown again

User cannot perform a search

Given Manomano search response is 500 error

When User perform any search

Then Website shows an error

Search data is stored

When User perform a search

Then 'Search' data has been sent correctly

Question 2

As you noticed, the previous user story in exercise 1 doesn't have enough information, so please write an email to Joanna, our PM, asking for information that you think is missing in order to help build the quality strategy.

Good morning Joanna,

According to the regression test part that has been completed last week, I check, as a part of it, the Story MNMN-123 "As a non registered user, I want to search for a product". For this feature, I review the whole functionality for the frontend flow including and I got some doubts that I would like to discuss with you:

First of all, I miss information about the behavior of the API Endpoint that this feature calls. Apart of the happy path flow, I see interesting the addition of some edge cases as error codes that this EP allows and which data is mandatory in order to confirm that all the situations are fully covered and doesn't break the platform.

Also, I'm aware that data information is really important for our business so could be an improvement to add some testing just to be sure that what the website sends is correct and then confirm the veracity of it.

As always, Everything can be discussed, so please contact me if you have some doubts or more details to take into account and get into an agreement.

Best regards,
Javier.

Question 3

Please go to our site, try to find a defect and report it as if you were going to introduce it into a real defect-tracking system. In case you could not find any defect (hopefully) just create a defect that you can think of.

[FrontEnd] Search with empty search field shows an error

Browser: Mozilla firefox version 88.0 (64-bit)

Language: English

Country domain: [co.uk](https://www.manomano.co.uk)

Website version: 1.2.0 (I cannot find the real web version)

Environment: Prod

Prerequisites:

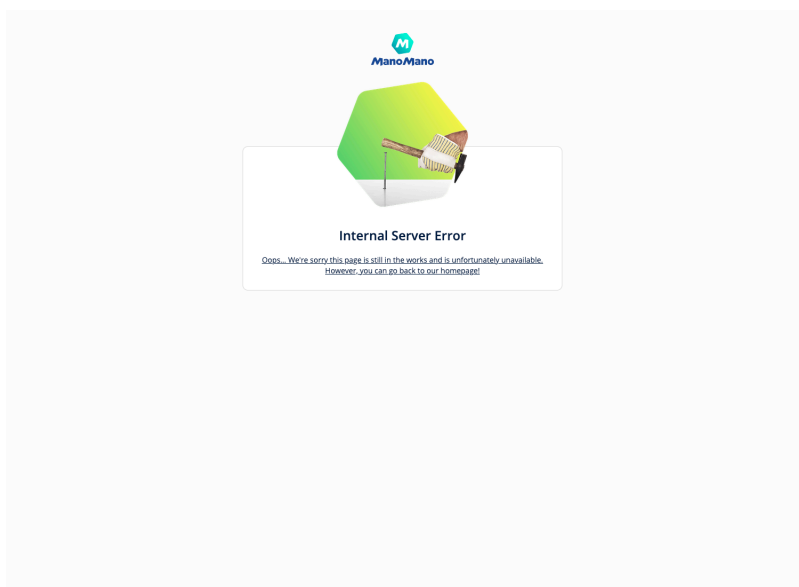
- User is not logged in

Procedure:

1. Acces to <https://www.manomano.co.uk/> website
2. Type any valid term
3. Search it
4. Erase all words inside search field
5. Press enter to search

Current Result:

App goes to 500 internal error website



Expected Result:

Website goes to home screen

Question 4

We would like you to automate the testing of an API, being as creative as you can with the test.

I didn't get this exercise working so let me explain how I would organize it and some examples.

This framework is divided by three parts:

Inside **Request** we will find different functions per each endpoint. The aim of it is to have all endpoints ready to be used, without information and waiting for it.

Inside **Test** we will find all the test that will cover the test plan for the API. Different flows per endpoint, giving them the information to make the api call and asserting the information that they return.

Inside **Resources** we will gather extra information needed for the assertions that test have to make. Sometimes we need to check that the json response itself is valid, for this reason is necessary to have all the schemas stored to be compared with the current json response.

The objective of this organization is to give scalability to the code, allow to make little change and don't affect the functionality of ever test that could be affected.

Question 5

The goal of this test is to check if you are able to automate a test of a given website, dealing with the locators, etc

The following test will be developed using Cypress framework with JS. But let me introduce here how I want to organize all the frontend Test Cases:

BDD or not to BDD?

First question we should make is to use Cucumber to create the whole test plan using an understandable language for people out of development team. This decision depends on the use we give to the test plan, for instance, a well defined test plan can be used as documentation. Also, a Product Manager can be familiar with this process and start creating tests as a part of the User Story agreement. These are some of the reasons to start using BDD, nevertheless is useless for QA testing purposes only.

PageObjects or App Actions

Is really common to use PageObjects pattern with python, Java or C# languages. However, is more suitable to use App Actions pattern instead.

SQL calls

Last but not least, SQL calls to enhance E2E Testing. Sometimes, a test will request a specific status of the website to review some functionality, for instance, a previous search or a specific price of a product. So some SQL calls that can initialize all information needed will save extra steps and time.

Question 6

With the following architecture please explain what kinds of tests would you do, including each level and type of tests you can think of).

In case you make any assumptions, please just justify them.

To answer this question let me divide my response in to parts. First, a review of every section of the architecture and which tests are useful to check properly that specific technology. Second, the process to test every features that enters the development process.

So what we have? Let me define the context:

- Computer with Firefox web browser
- iPhone mobile with Manomano app and safari web browser
- Android mobile with Manomano app and Firefox web browser
- Access to our Private API from BackEnd
- Access to Public API from PayPal
- Access to our Private Database

FrontEnd. Our Test Plan cover the whole website divided by priorities. It covers happy path and edge cases.

Mobile FrontEnd. An adaptation for FrontEnd Test Plan that will check the diferencies between both devices. It covers happy path and edge cases.

Mobile App. Our Test Plan cover the whole App for iOS and Android OS. It covers happy path and edge cases.

Manomano API. A test plan for every EndPoint. It covers happy path and edge cases.

Paypal API. A test plan for every EndPoint. It covers happy path and edge cases.

The first part of the process start with the definition of the new Feature. When the User Story fully defined and the designs attached, is our turn to check it: read it and think if all the flows are fully covered. After it, QA member may create all the Test Cases needed to test correctly this feature.

Secondly, the feature follow the agile process until is developed and ready to be tested. At this point, every workmate is able to test the new feature. It means to follow the test cases created previously and check that everything works as expected.

At the same time, it's a good moment to create the automated test of the ones defined and include them inside regression automated test plan.

Finally, once the sprint is ending and the new app version is built, is the turn of the automated test plan to check this app. It will cover the most important flows, all the happy path tests and it will be repeated for different devices and OS.

This process may vary for BackEnd or any other technology: in case we use continuous integration, the regression automated part can be launch automatically just before the deploy of the updates.