Testing your project after project/iteration completion

1. Type of testing

Besides unit testing and UX (User experience) testing, there are a lot of testing needed to perform to ensure integrity of the project.

We will list the most common testing that needs to be performed after project/ iteration completion and provide further explanation:

(*Regarding iteration: in agile software development, an iteration is a single development cycle, usually measured as one week or two weeks. An iteration may also be defined as the elapsed time between iteration planning sessions.)

1.1 Stress testing

1.2 Load testing

1.3 System testing

1.4 Backup & Recovery testing

2. Testing in detail

2.1 Stress Testing

Stress testing is to test the robustness of your web application through testing beyond the limits of normal operation. It is mostly using scripts and denial of service tools to observe how the performance of website is impacted.

However, it would be best to perform such kind of testing using standalone server in the development environment (The local server specification shall be the same with your production server) that you have full control of it, performing such kind of testing under production environment might causes your website to crash thus out of reach by audience. And it has less control if you are under a shared hosting site.

Most importantly, **do not ever perform stress testing on other websites that is not belong to yours** unless they accept and acknowledge the impact it will have to their website. It might cause business to lose its revenue and there are chances that you are getting sued because of that.

The deny of service tools are most commonly used instead of people writing script here, you can find the common tools from this website, most of them are user friendly and you can get hands on quite easily, but be really careful about what you were doing:

https://resources.infosecinstitute.com/dos-attacks-free-dos-attacking-tools/#gref

2.2 Load testing

Compared to stress test, which test the limits of the normal operation, load testing is performed under more controlled environment, it moves from a low load (Number of peoples who are accessing your website at the same time) to high load then monitor the impact of it.

You could use either virtual machine to test (Under development environment) or running it directly through online services such as: https://loadimpact.com/, since the testing is performed under a more controlled environment, most of the people prefer to test directly through online.

To perform the testing under virtual machine, you could follow the link given below to understand how it could be performed locally.

https://almvm.azurewebsites.net/labs/tfs/load/

To perform the testing online, you could go to the site here, enter the website URL you wish to perform testing on and get results in a few minutes. https://loadimpact.com/, the screenshot provided below is the result of the testing.

We will provide more explanations:

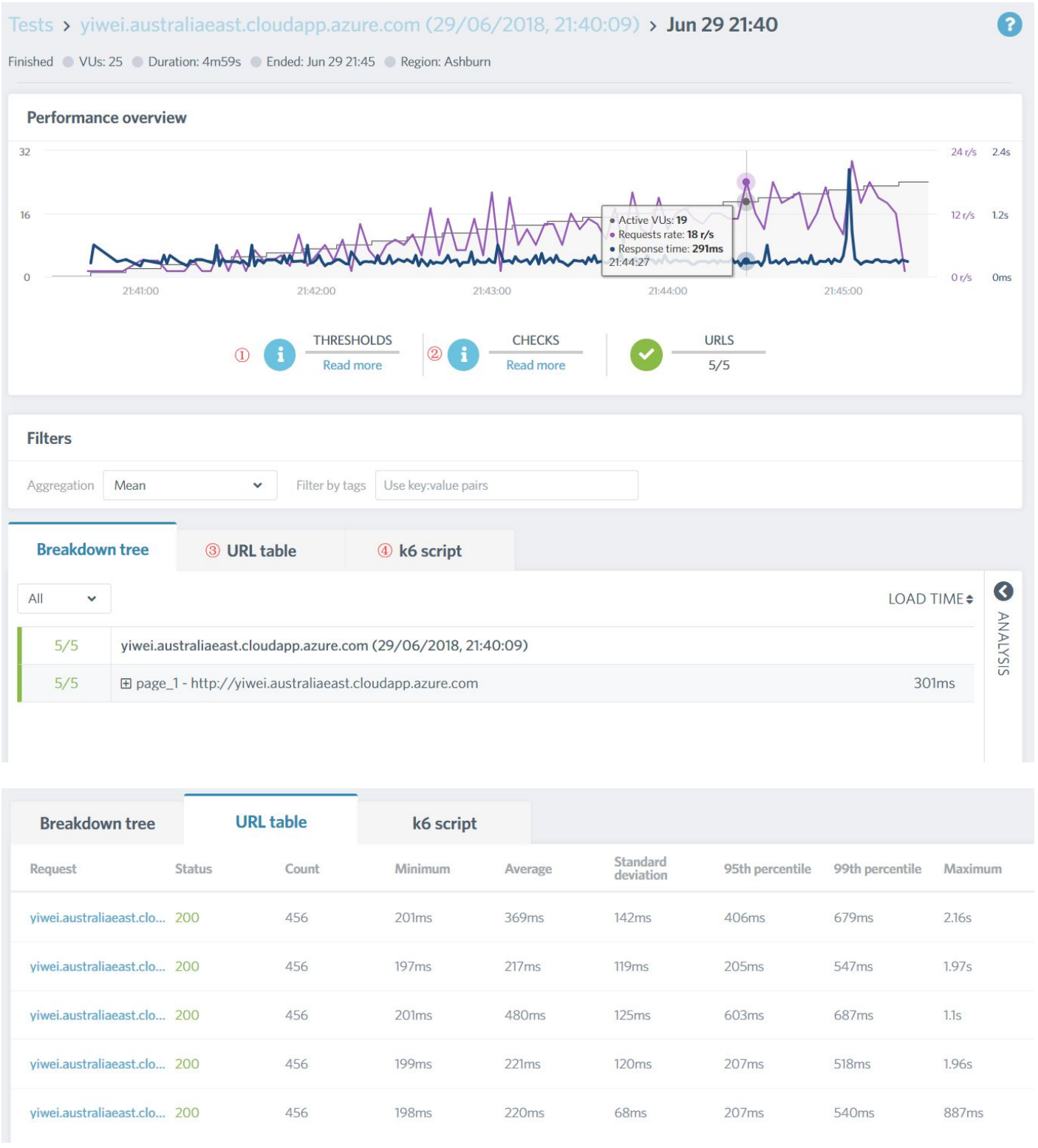
Both ① and ② are used to customized what you think as a "failure"

1 Thresholds: In simple term, it is the criteria that you wish to set to consider the result as a failure.
For example, you could set a threshold saying if there are 5% or more requests resulted in the server returning anything else besides 200-responses. Then it is considered as a failure.
*200-response: Standard response for successful HTTP requests.

2 Checks: Run checks on the value, its result will always return "true" or "false". For example, you can customize the condition for a 200-response, so that it could return "true" when the response is equals to 200.
This is often used to fast a result so that it could send the result to Threshold for further analysed for processing

3 The URL table is the table that contains links that is commonly accessed by the server, for example, css and JavaScript files.

4 k6 script is a developer-centric load testing tool script designed to developer, incorporate load testing into the workflow, and ultimately get it into automation flow. The threshold and Checks is using k6 script to adjust the final desired outcome you wish to see



2.3 System testing

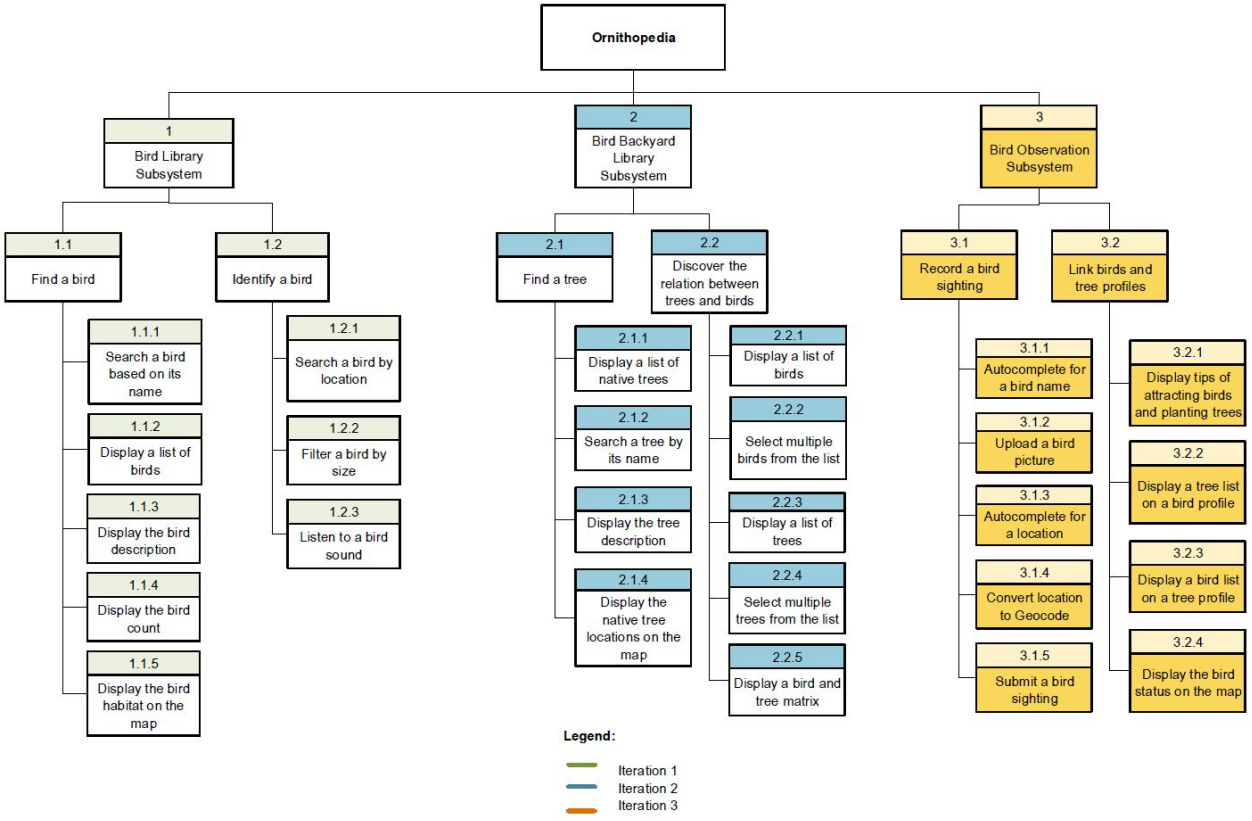
System testing which belongs to black-box testing is a testing to conduct a complete system to see whether the system matches the requirement or not.

To perform the test, you need to check all the possible functionality developed, and also its workflow as well as the user story of the whole business process to see whether it meets the expectation of customers.

To perform every aspect of testing, a proper functional decomposition diagram and proper user story is needed.

Example of diagram and user story is given below:

Functional decomposition diagram:



User story:

As a bird lover, I want to know what kind of birds are living in a specific area.
As a bird lover, I want to view a bird's profile.
As a bird lover, I want to identify a bird.
As a bird lover, I want to know the various locations of a specific bird around Australia.
As a bird lover, I want to know the bird count of specific bird in last 10 years.

After obtaining all these crucial information, system tester could test this functionality one by one to ensure everything is according to the requirement and design

2.4 Backup and recovery testing

This is to ensure that a proper backup is created and recovery is being done successfully so that once a failure over happens, the system could recover back in time.

For Azure hosting environment, you could follow the link provided below to perform backup and recovery testing:

https://docs.microsoft.com/en-us/azure/backup/backup-azure-vms-first-look-arm