Part 5 Interim Report

# 5. Testing and Evaluation

## 5.1. Introduction

This chapter will delve in the different steps and stages of what the plan is for the testing of the system combined with the evaluation mechanisms drafted up. This will incorporate tools from software as well as human interaction methods too.

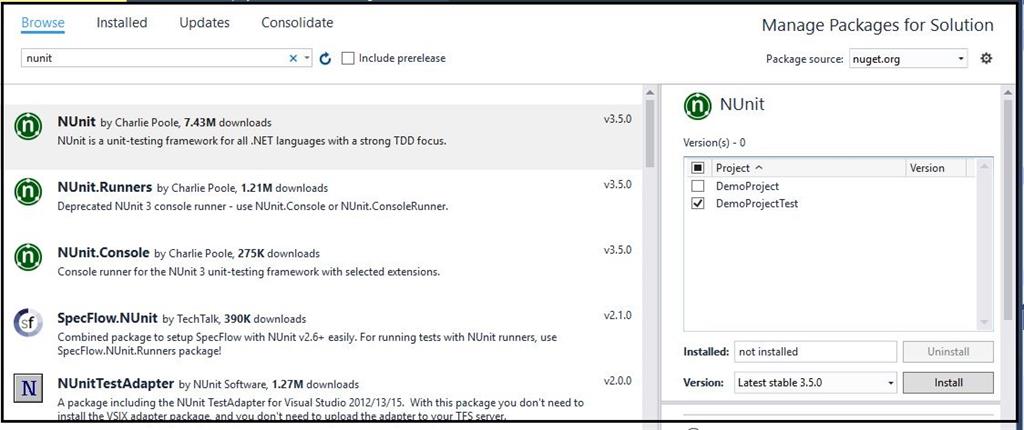
## 5.2. Plan for Testing

NDMA will be tested across all the different layers of the application. The plan for the testing aspect, in a high-level overview, is to combine the usage of using manual / human methods and integrated automated software. The system in its entirety will use Acceptance testing in the full stack development to ensure each layer interact with one another correctly and the state of NDMA is acceptable according to minimum operations standards. Tools, such as Benchmarks or Performance analysis or profiling tools, which are freely provided by Google and other big tech companies, will also be used in conjunction with the debugging process of the application.

For the front-end of the application, I will develop an auto testing application. The technologies used in this development will be the frameworks of Selenium and NUnit. The Selenium framework allows the tests developed to be automate tasks combined with ensuring the testing system itself is decoupled. This will allow the potential of testing certain components or functionalities without affecting other aspects of the NDMA or the testing system. Reusing certain tests for similar functionality is another reason for using Selenium framework. Selenium is a working example of module testing.

NUnit is an open source Unit testing framework released under the MIT license for the .Net framework. Unit testing involves taking certain components of the application, test their performance against certain cases and assert them against expected outcome. This could be testing certain operations of the application and can use self-managed inputs as part of each case. NUnit, when combined with Selenium, can test each component of NDMA in automated session and assert their outcome.





For the middleware, NUnit testing will be integrated. Since this area is the key part of NDMA, Black box, White box and Grey box testing will be used here to ensure the performance and efficiency are optimised simultaneously with correct information. This is combined with ensuring the application can withstand faults, whether it is a technical issue or an injection attack from the user from the front-end of the application. Ensuring NDMA is de-coupled as necessary, so Subsystem testing will be incorporated. This is to ensure each application area only communicates in the way they are designed to do, combined with handling items they need to handle with only.

For database testing, DbFit will be used. This framework uses test-driven development as the methodology and encapsulates both unit testing with integration testing as part of its features. This will allow the test cases to capture specific information of the relational database. As a result, we can test the local storage and the remote storage using the same test scenarios with different cases. With the remote storage, Azure offers the services of regression testing with performance analytics, which will be integrated into the database testing. This will allow test cases to be designed for the availability and consistency of the remote application to be completed in ongoing iterations.

Manual testing will also be captured. A set of instructions will be provided to each person whom tests the application in order to provide as much accuracy as possible to the test cases. The instructions will be specific enough for the user to understand what to do combined with the vagueness of allowing the user free roaming of what to test. This will provide an more overview of the application behaviour combined with another opinion on the application design

## 5.3. Plan for Evaluation

The plan for the evaluation is in three different stages, which is to be completed altogether and separated for a diverse set of detail. The three stages are using the 10 Neison's Heurstics as an evaluation method, comparison with industry technology as another evaluation method and finally getting feedback from the users who have teste the application.

When getting the feedback from the users who have tested the system, two methods will be used. One is gathering the opinion of the users from their experience in terms of the clarity of the application, the design, the user interface, the user experience and the performance of the application. The other method would be questions drafted using the Neison's Heurstics as the guide to gather the other experience of the user. A comparison from other technologies that the user has used is an extendable option.

The evaluation of the system will be completed by Self too. This will be done using the same format as the methods used for the users’ evaluation described above. From comparison with industry technology to using the Neison's Heurstics (can be simultaneously completed), the similarities and contrasts with be documented and outlined in each iteration. Once the evaluation is completed, using both the users’ report and the self-documentation, the system will be enhanced and modified to ensure it suits the needs of the users completely.

## 5.4. Conclusions

This chapter delved into the different test techniques and types planned for the system, as well as the evaluation mechanisms for the optimising of the application design. From the boxes testing (white, grey and black) to combining unit and module testing to combining different technologies to accomplish these goals when developing the system. The evaluation phase will be key to the development of the application through gathering the user assessment to the comparison to the industry technology. Using Neison's Heurstics is also integral to the evaluation phase too.