**CPS Faster Payment- Software Upgrade of Uipath**

**Alpha Mackie**

**Software Developer Level 4**

Introduction

The AIG was doing a Software upgrade of Uipath from version 17.1 to 18.4 and I was tasked with building test harnesses for 20 components to check and recorded the outputs of the two versions in a spreadsheet by the lead developer in my team. I have only included some of the test harnesses and the outputs.

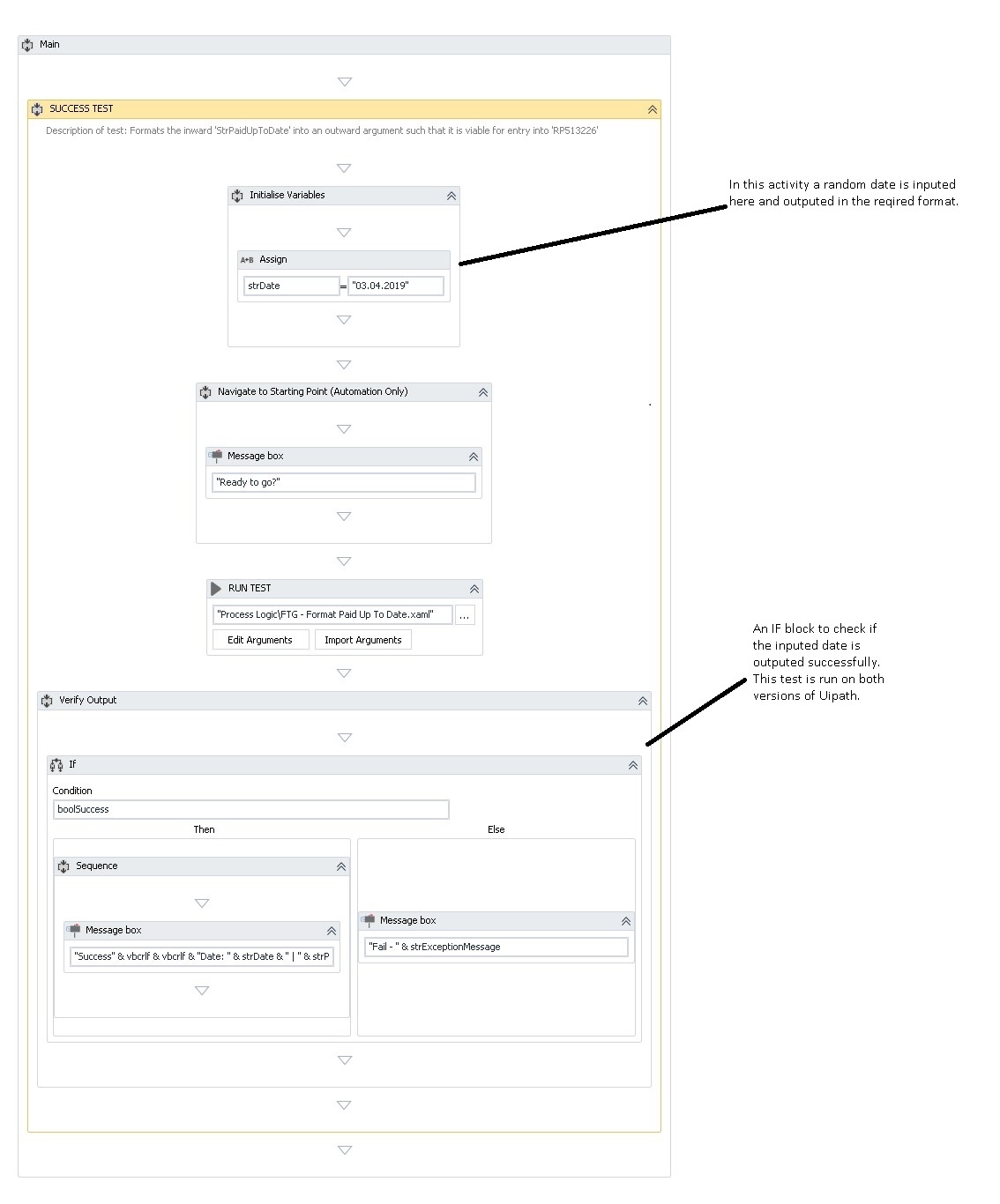
First step

I went through each of the components to understand what processes they were built to automate. This was a challenging process for me as this was not a training exercise, but real-world automation processes and this unit testing was part of the migration process. I consulted with the lead developer to get a better understanding of what was required of me and also asked him to talk me through building one test harness.

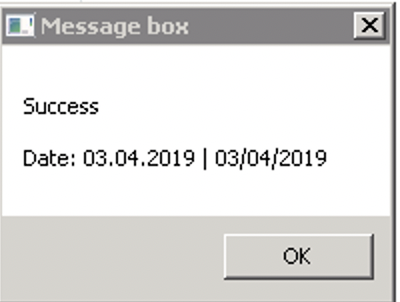
Development process

I have not included the spreadsheet that shows all of the components that I built test harnesses for and unit tested. In this submission, I have only included four test harnesses and the resulting outputs. I built the test harnesses in the two versions of UiPath and ran the tests from them. Each of the test harnesses started with an activity that initialises the variable inputs that will provide the outputs in the required format. An Invoke activity that gets the component to be tested and an if block to check whether the test was successful or not.

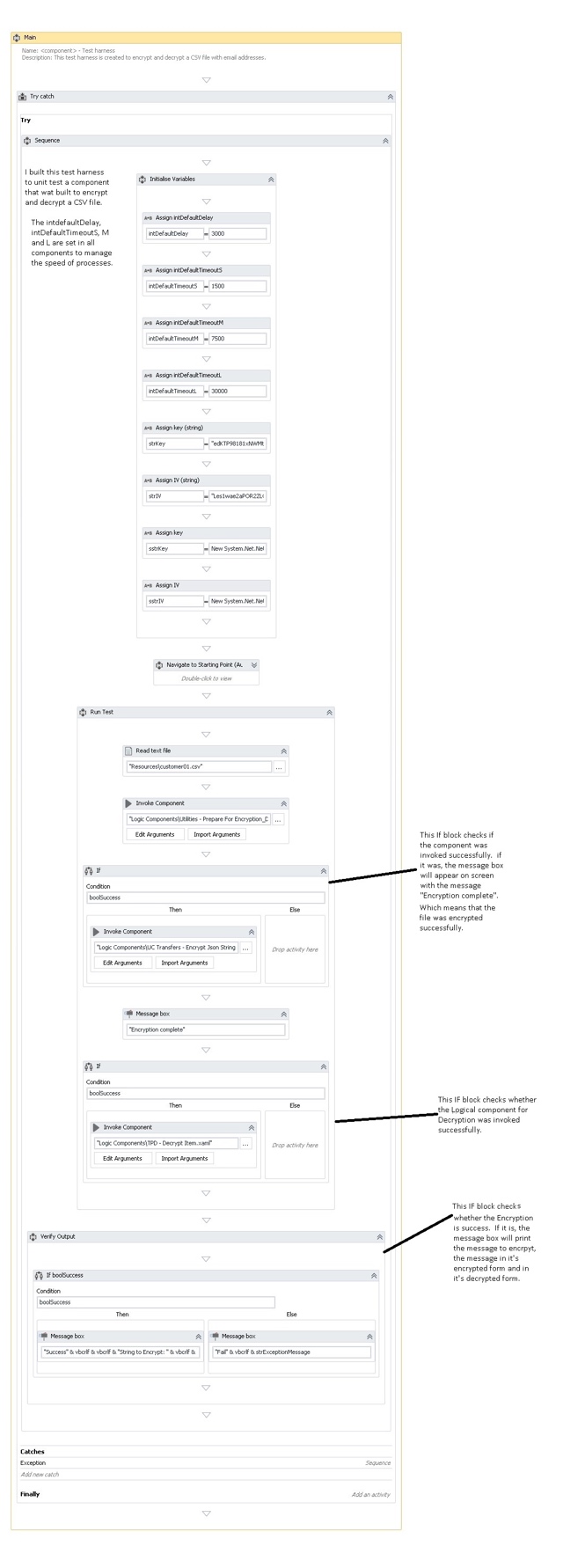
Please see annotations on each of the submissions below:



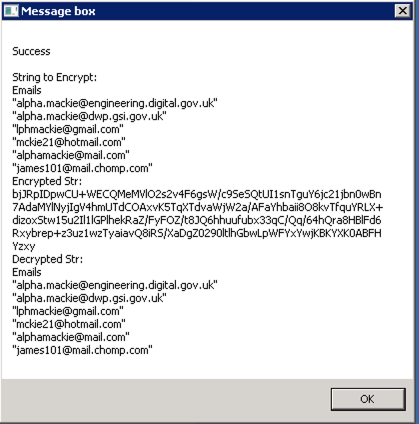
*Figure 1: Formatted Dates*



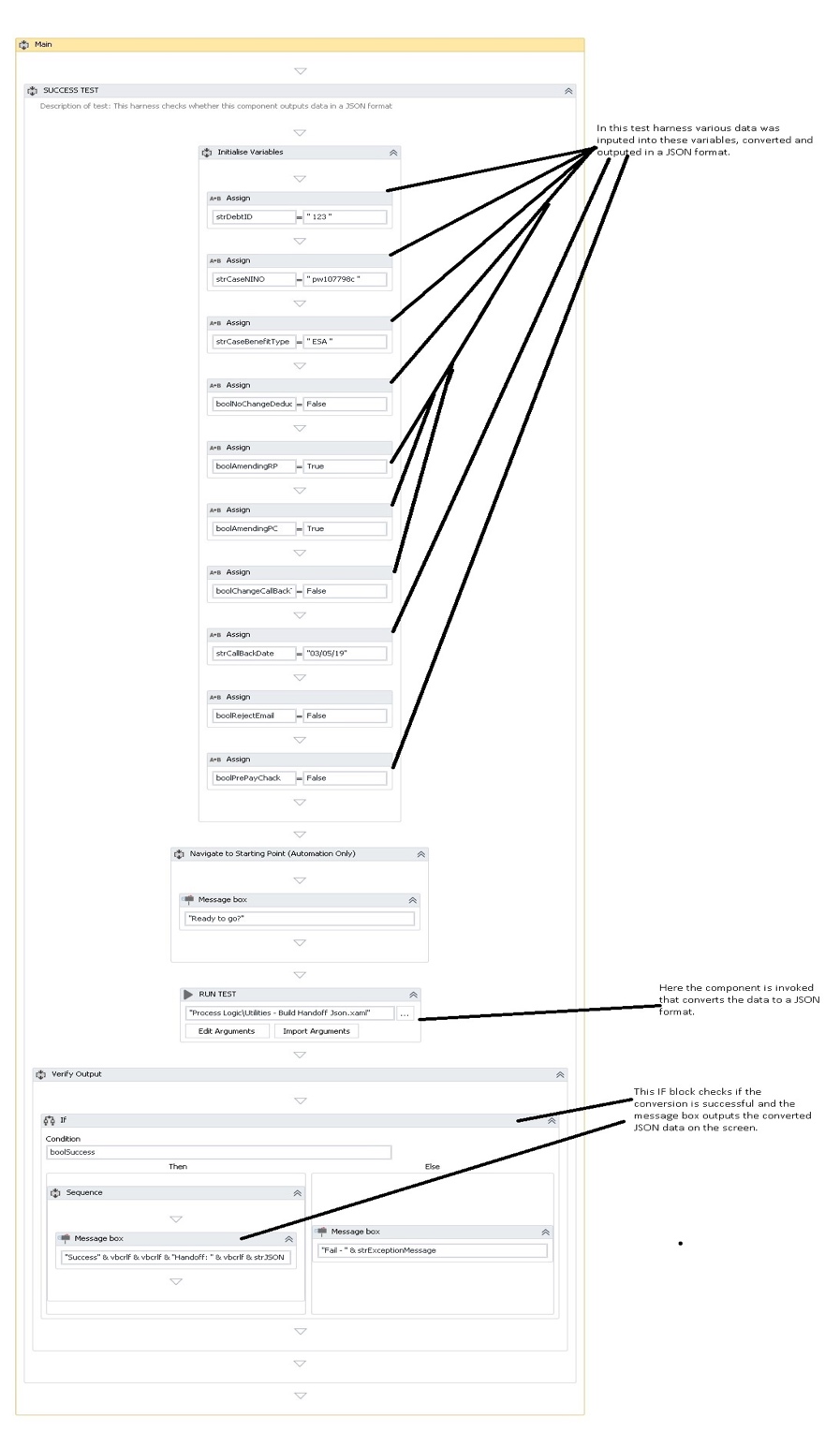
*Figure 2: Test output.*



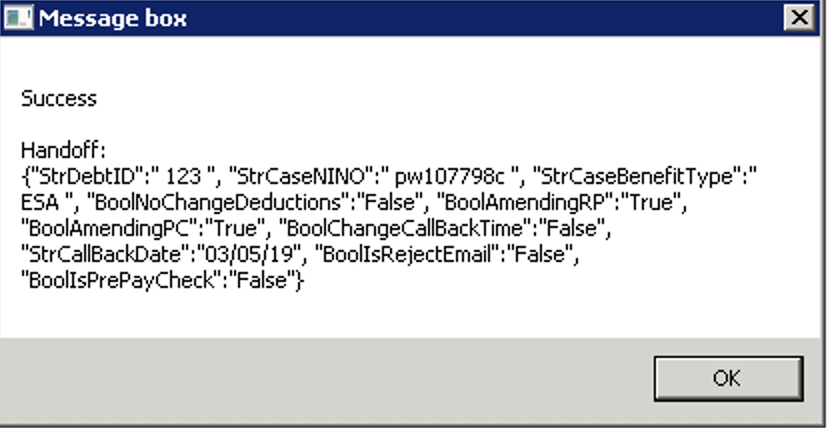
*Figure 3: Test harness for encrypting and decrypting data*



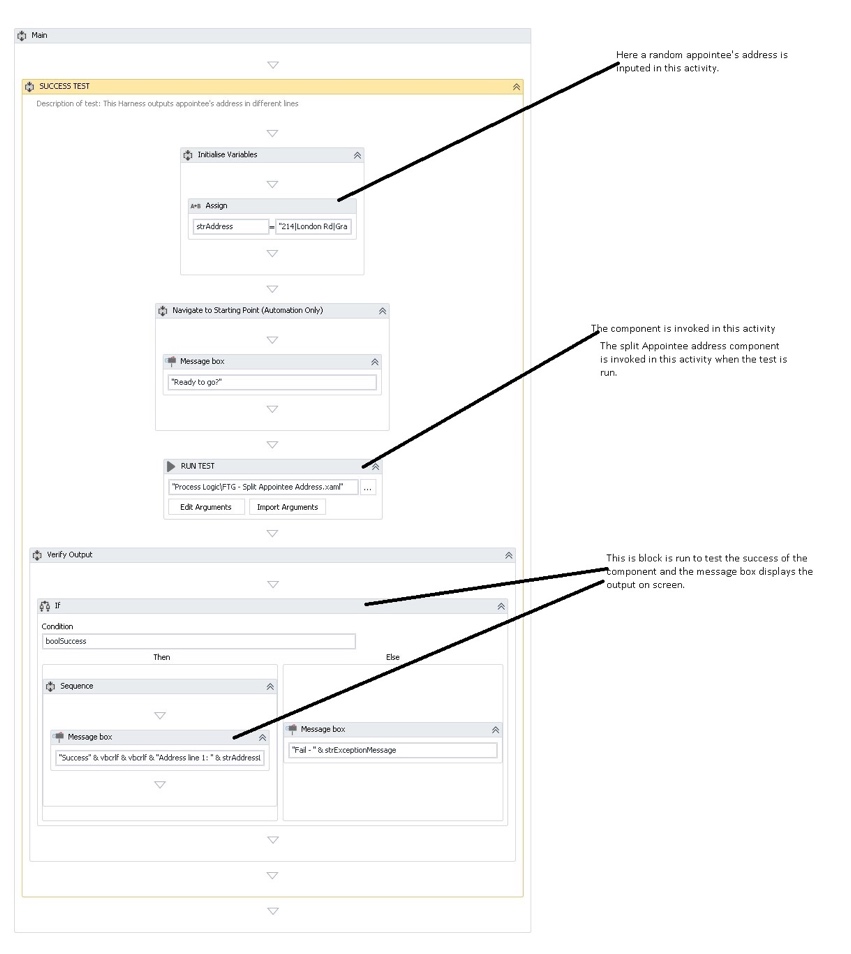
*Figure 4: The above screenshot shows the unencrypted data, the encrypted data and the decrypted data output.*



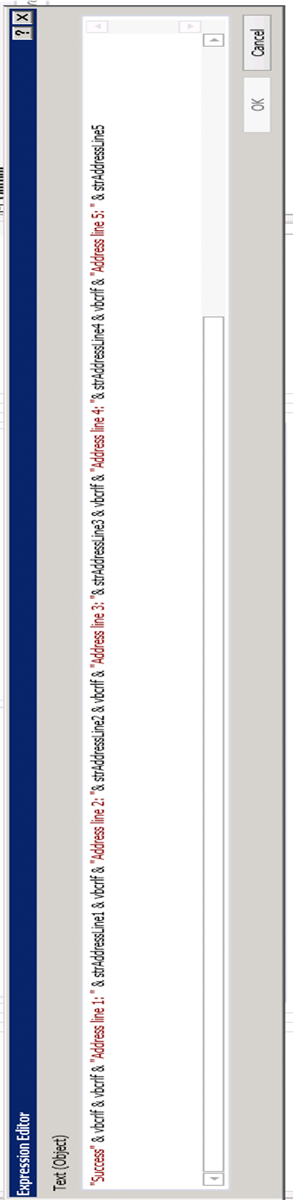
*Figure 5: Test harness to display data in JSON format*



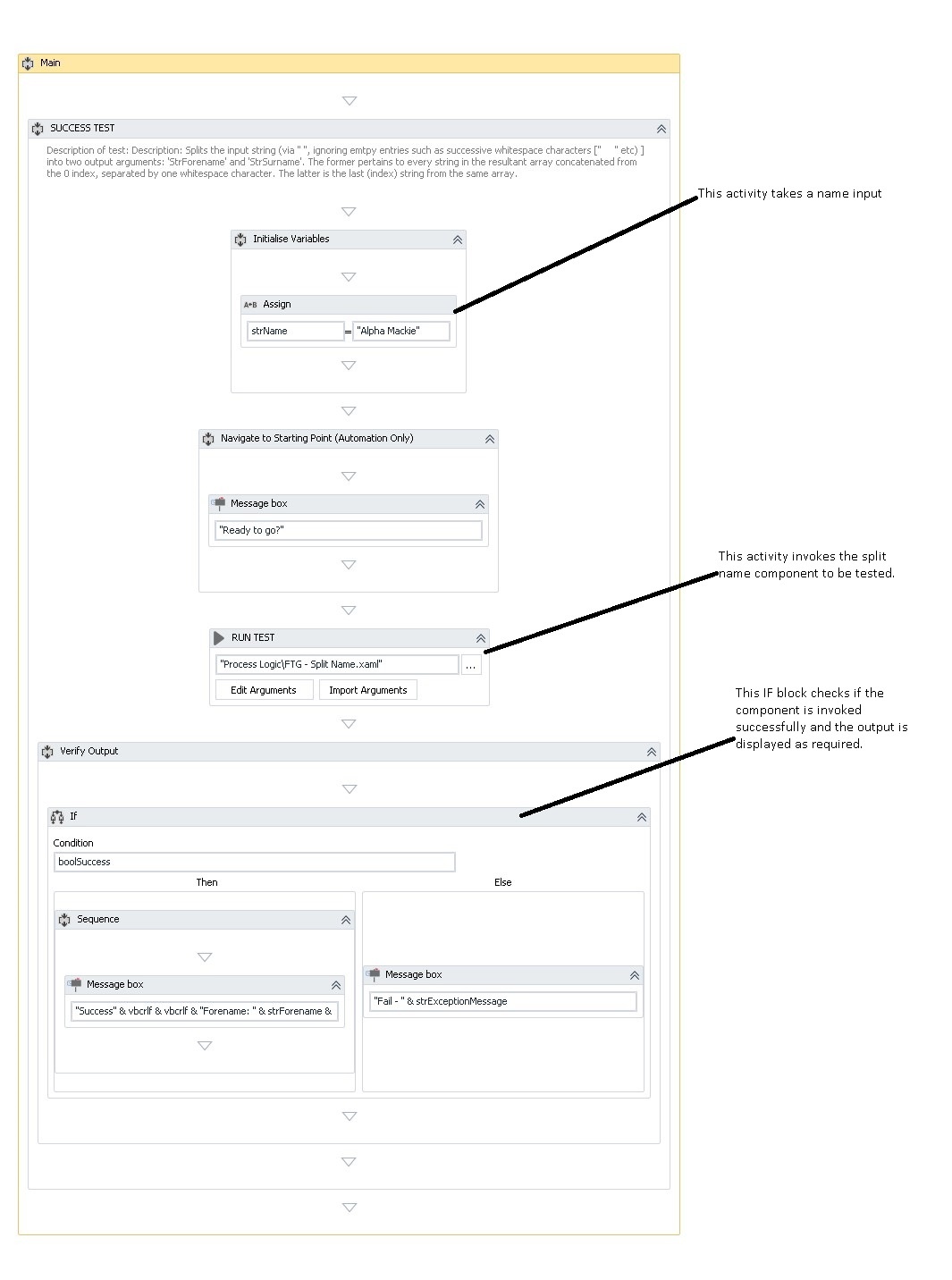
*Figure 6: This output shows the output displayed in a JSON format.*



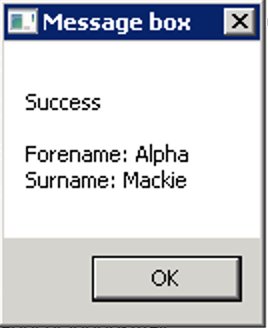
*Figure 8: Output address in multiple lines.*



*Figure 9: Output an address in multiple lines.*



*Figure 10: Split name into first name and surname.*



*Figure 11: Split name into first name and surname.*

Final steps

The senior developer was then provided with the opportunity to test and confirm that each of the unit tests worked and produced the correct outputs. If that was the case, it was then confirmed on JIRA as done (task completed), by the senior developer and the Project lead (Scrum master).

Conclusion

In this project I have included only four test harnesses out of the twenty. I examined the code in every component to understand what each component does and what output was expected. I then built the test harnesses and tested the components on both versions. The first one displays the output in the required format when the date is entered in any format. The second one tests a component that is supposed to encrypted and decrypted a strings of email addresses as shown in figure 4. The third test harness checks whether the component successfully converts data to a json object and the last one converts data and display the data in multiple lines as shown in figure 11.

From this project, I learned how not just to build test harnesses for unit testing, also how to configure them to ensure that I get the right outcome.

My main challenge was to understand what a component does and then build my test harness in order to get the required output.  
At every stage I would ask my colleagues the other experienced developers to look at my code and check that the code was right and it was written as per AIG standards.