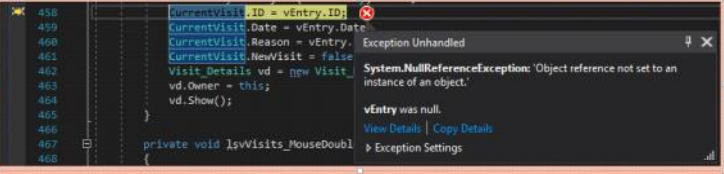


Destructive Testing Response - Modular and Unit Testing

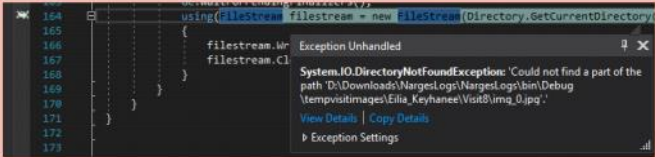
Saturday, 7 July 2018 9:21 PM

Test 1 - Major Runtime Crash

Runtime Crash	Double clicked the reason tab with 1 or 2 entries.		Not Yet Fixed	Major
---------------	--	--	---------------	-------

The crash cannot be recreated. It is highly likely that the solution to the previous error was the solution to this error, as both errors involved double clicking an invalid space in the Visits list. The error is marked as fixed for the time being, until it is recreated at another destructive test.

Test 1 - Minor Runtime Crash

Runtime Crash	Opened multiple windows of the same visit and closed them until crash.		Not Yet Fixed	Minor
---------------	--	---	---------------	-------

The error is caused by unconventional behaviour and can therefore be prevented by the limitation of this behaviour. The error can be prevented if the user is prevented from opening two visits at the same time.

Solution:

Opening a visit window now disables the Patient Details window, and closing a visit window enables the owner window.

Test 1 - Minor Bug

Solution:

The previous solution, of making the user unable to interact with the details window when the visits window is open, addressed this issue as well.

Test 2 - Critical Bug

The issue seems to be the calculation of each patient's age value. A Unit test will be conducted on the calculation algorithm in an attempt to locate the issue.

Unit Testing - Age Algorithm Unit:

Desk Checking - Upon conducting a desk check it was discovered that the binary decision statement in the algorithm that was responsible for incrementing the age if the birth date of the current year had already passed was faulty. The decision statement checked that both the month and the date had passed, while it is only meant to check if the month had passed or if the month was the current month and the day had passed.

Solution:

Edited the decision statement to correct the algorithm.

Test 2 - Major Bug

The server seems to be stuck in an infinite loop in which the text 'Command received' is repeatedly output. The function that contains this text will be subject to unit testing.

Unit Testing - Client Handling Function:

Desk Checking - Desk checking did not reveal any results. Breakpoint and line-by-line execution - Upon executing the server line by line once a breakpoint was reached, it became apparent that the pre-test for the handling loop was incorrect, as the check would always pass even if the client was disconnected. A work around was deployed where the loop would be always set to true, and there would be a final 'else' statement at the end of the loop which would only be executed if the received command was none of the recognised commands, which would disconnect the client and return to waiting for the next client with a command line available.