Deliverable #5

Team Members: Hannah Posch, Alex Thropp, Daniel Lee, Daniel Baczmaga

5 Deliverable 5

5.0 Description

For our fault injection, we chose to create a second folder called TeamCorrectFaultInjections that our testing script could loop through and run each faulted tested file. We wanted to create a second folder for ease of use and easy replication.

5.1 How to Run Faults

1. Open Terminal
2. git clone https://github.com/csci-362-fall-2018-01/TeamCorrect
3. cd TeamCorrect/TestAutomation/scripts (May need additional navigation, depending on where the repository was cloned to)
4. ./runAllTests.sh
5. Output will be stored in an html file which should open automatically in the default browser with the fault injection output on the bottom.

*If any of the above commands return permission denied, use 'sudo' in front of the command.

*If the ./runAllTests.sh returns command not found, use 'sudo chmod 755 runAllTests.sh', then './runAllTests.sh'

5.2 Faults Injects

1) Test Case 06 Fault Injection:
    a) File/Method: CholesterolReading.setLDLReading
    b) Fault: LDL Reading was changed to HDL Reading
        i) LDL Reading is not correctly set with input of 6.4
    c) Output: LDL reading set to 0.0
2) Test Case 10 Fault Injection:
    a) File/Method: ReadingTools.hourToSpinnerType
    b) Fault: changed hour>20 to hour<20
    c) Fault: changed hour>17 to hour<17
    d) Fault: changed hour>13 to hour<13

e) Fault: changed hour>7 to hour<7
f) Fault: changed hour>4 to hour<4
   i) Logic is now incorrect for the expression
g) Output:
   i) Hour: 24, SpinnerType: 8, Time of day: night
   ii) Hour: 8, SpinnerType: 5, Time of day: after dinner
   iii) Hour: 4, SpinnerType: 5, Time of day: after dinner
   iv) Hour: 19, SpinnerType: 5, Time of day: after dinner

3) Test Case 11 Fault Injection:
   a) File/Method: InputFilterMinMax.isInRange
   b) Fault: Changed operation signs in expression
      i) (Original: b > a ? c >= a && c <= b : c >= b && c <= a)
      ii) (Current: b < a ? c <= a && c >= b : c <= b && c >= a)
      iii) Logic is now incorrect.
   c) Output: Range Tester [12.0,1.5,8.2] = true
4) Test Case 14 Fault Injection:
   a) File/Method: PressureReading.setMaxReading
   b) Fault: Changed min to max and vice versa
      i) Incorrect value returned for set max
   c) Output: Max Reading returned = 0.0 Max Reading returned = 0.0
5) Test Case 20 Fault Injection:
   a) File/Method: ReadingTools.parseReading
   b) Fault: Set all returns to null.
      i) All values return as null.
   c) Output: str to number 6 = null

5.3 Evaluation and Experiences

Fault injection was a fun experience because it meant breaking the code that we have been thoroughly testing. For our application, modifying the logic and expressions for the Glucose application made the most sense to ensure that the application was not comparing or calculating numbers correctly. We are happy with our fault injection as they test a variety of the functionality of the system. All 5 fault injections broke their respective functions and did not break additional functionality of the code. This illustrates how well the Glucosio application is low coupled and ensures the maximum functionality of the system overall.