**Wearable Epilepsy Device & MC Team**

**T4: Patterns Matching, Association and Prediction**

# Test Harness

* Created a test harness in the main to perform every test
  + Default Test Values Used for Various Tests
    - Issue 1
      * Pattern
        + True, True, True, True, True
      * Message
        + Seizure Detected: EMERGENCY (Complete bodily shutdown)
    - Issue 2
      * Pattern
        + True, False, False, False, True
      * Message
        + Seizure Detected: Minor (Sudden, repeated fear or anger)
    - Issue 3
      * Pattern
        + True, False, True, False, True
      * Message
        + Seizure Detected: Minor (Repeated, unusual movements such as head nodding or rapid blinking)
    - Patient Stats
      * Signal 1: 100 to 130
      * Signal 2: 50 to 100
      * Signal 3: 80 to 150
      * Signal 4: 20 to 40
      * Signal 5: 130 to 180

# Unit Testing

* Confirm that each class and its methods perform as expected
  + Test the modifying and returning of attributes
    - Test setters to modify attributes
      * Signal class must be able to set the value for the frequency
      * Issue class must be able to set values for the pattern and message
      * Patient class must be able to set values for the stats
        + isAbnormal is class exclusive so there is no setter
        + Stats must have a valid range. Tested using White-Box
    - Test getters to verify proper attribute values
      * Signal class must be able to get the value for the frequency
      * Issue class must be able to get values for the pattern and message
      * Patient class must be able to get values for the stats
        + isAbnormal is class exclusive so there is no getter
  + All the setters and getters behaved properly

# White-Box Testing

Test all known paths of the code for valid and invalid inputs. We may assume that all values will work if the boundaries behave as expected. So, test every boundary for correctness.

## Test for Patient stats range

* Equivalence Classes
  + Invalid Ranges
  + Valid Ranges
* Range must go from a smaller number to a larger number. (Ex: 80-100)
  + Test for when numbers are the same. (Valid)
    - Test Valid Signal 3 Stats: 103 to 103
      * Expected: Signal 3 Stats Set
      * Result: Signal 3 Stats Set
      * Conclusion: Test value behaves properly
  + Test for when smaller number is one greater than larger number (Invalid)
    - Test Invalid Signal 3 Stats: 103 to 2
      * Expected: “Signal 3 has an invalid range: 103 to 2”
      * Result: “Signal 3 has an invalid range: 103 to 2”
      * Conclusion: Test value behaves properly
  + Test for when smaller number is one less than larger number (Valid)
    - Test Valid Signal 3 Stats: 102 to 103
      * Expected: Signal 3 Stats Set
      * Result: Signal 3 Stats Set
      * Conclusion: Test value behaves properly

## Test Entire Program with Default Test Values.

### Test Patient Stat Range Boundaries

* Signal Abnormality Equivalence Classes
  + Signals below valid Patient Stat Ranges. (Abnormal)
  + Signals within valid Patient Stat Ranges. (Normal)
  + Signals above valid Patient Stat Ranges. (Abnormal)
* Test the boundaries of Patient Stat Ranges with the Signals.
  + Test All Signals for Lower Boundary (Within and Normal)
    - Signals are normal if they are within the boundaries.
      * LowerBoundary1(100) <= Signal1(100) <= UpperBoundary1(130) is True
        + Signal is normal so isAbnormal[0] = False
      * LowerBoundary2(50) <= Signal2(50) <= UpperBoundary2(100) is True
        + Signal is normal so isAbnormal[1] = False
      * LowerBoundary3(80) <= Signal3(80) <= UpperBoundary3(150) is True
        + Signal is normal so isAbnormal[2] = False
      * LowerBoundary4(20) <= Signal4(20) <= UpperBoundary4(40) is True
        + Signal is normal so isAbnormal[3] = False
      * LowerBoundary5(130) <= Signal5(130) <= UpperBoundary5(180) is True
        + Signal is normal so isAbnormal[4] = False
    - Results
      * Expected: isAbnormal is All False
      * Result: isAbnormal is All False
      * Conclusion: Test value behaves properly
  + Test All Signals for Upper Boundary (Within and Normal)
    - Signals are normal if they are within the boundaries.
      * LowerBoundary1(100) <= Signal1(130) <= UpperBoundary1(130) is True
        + Signal is normal so isAbnormal[0] = False
      * LowerBoundary2(50) <= Signal2(100) <= UpperBoundary2(100) is True
        + Signal is normal so isAbnormal[1] = False
      * LowerBoundary3(80) <= Signal3(150) <= UpperBoundary3(150) is True
        + Signal is normal so isAbnormal[2] = False
      * LowerBoundary4(20) <= Signal4(40) <= UpperBoundary4(40) is True
        + Signal is normal so isAbnormal[3] = False
      * LowerBoundary5(130) <= Signal5(180) <= UpperBoundary5(180) is True
        + Signal is normal so isAbnormal[4] = False
    - Results
      * Expected: isAbnormal is All False
      * Result: isAbnormal is All False
      * Conclusion: Test value behaves properly
  + Test All Signals for One Below the Lower Boundary (Below and Abnormal)
    - Signals are abnormal if they are not within the boundaries.
      * LowerBoundary1(100) <= Signal1(99) <= UpperBoundary1(130) is False
        + Signal is abnormal so isAbnormal[0] = True
      * LowerBoundary2(50) <= Signal2(49) <= UpperBoundary2(100) is False
        + Signal is abnormal so isAbnormal[1] = True
      * LowerBoundary3(80) <= Signal3(79) <= UpperBoundary3(150) is False
        + Signal is abnormal so isAbnormal[2] = True
      * LowerBoundary4(20) <= Signal4(19) <= UpperBoundary4(40) is False
        + Signal is abnormal so isAbnormal[3] = True
      * LowerBoundary5(130) <= Signal5(129) <= UpperBoundary5(180) is False
        + Signal is abnormal so isAbnormal[4] = True
    - Results
      * Expected: isAbnormal is All True
      * Result: isAbnormal is All True
      * Conclusion: Test value behaves properly.
  + Test All Signals for One Above the Upper Boundary (Above and Abnormal)
    - Signals are are abnormal if they are not within the boundaries.
      * LowerBoundary1(100) <= Signal1(131) <= UpperBoundary1(130) is False
        + Signal is abnormal so isAbnormal[0] = True
      * LowerBoundary2(50) <= Signal2(101) <= UpperBoundary2(100) is False
        + Signal is abnormal so isAbnormal[1] = True
      * LowerBoundary3(80) <= Signal3(151) <= UpperBoundary3(150) is False
        + Signal is abnormal so isAbnormal[2] = True
      * LowerBoundary4(20) <= Signal4(41) <= UpperBoundary4(40) is False
        + Signal is abnormal so isAbnormal[3] = True
      * LowerBoundary5(130) <= Signal5(181) <= UpperBoundary5(180) is False
        + Signal is abnormal so isAbnormal[4] = True
    - Results
      * Expected: isAbnormal is All True
      * Result: isAbnormal is All True
      * Conclusion: Test value behaves properly

### Test Issue Detection

* Test Issues Equivalence Classes
  + Issue 1 Detected (True, True, True, True, True)
  + Issue 2 Detected (True, False, False, False, True)
  + Issue 3 Detected (True, False, True, False, True)
  + Normal (Any pattern besides the test Issues)
* Test Issue Pattern Detection with the Abnormality Patterns
  + Issue 1
    - Issue 1 is caused when isAbnormal = {True, True, True, True, True}
      * LowerBoundary1(100) <= Signal1(90) <= UpperBoundary1(130) is False
        + Signal is abnormal so isAbnormal[0] = True
      * LowerBoundary2(50) <= Signal2(110) <= UpperBoundary2(100) is False
        + Signal is abnormal so isAbnormal[1] = True
      * LowerBoundary3(80) <= Signal3(40) <= UpperBoundary3(150) is False
        + Signal is abnormal so isAbnormal[2] = True
      * LowerBoundary4(20) <= Signal4(80) <= UpperBoundary4(40) is False
        + Signal is abnormal so isAbnormal[3] = True
      * LowerBoundary5(130) <= Signal5(10) <= UpperBoundary5(180) is False
        + Signal is abnormal so isAbnormal[4] = True
    - Results
      * Expected: "Seizure Detected: EMERGENCY (Complete bodily shutdown)"
      * Result: "Seizure Detected: EMERGENCY (Complete bodily shutdown)"
      * Conclusion: Test value behaves properly
  + Issue 2
    - Issue 2 is caused when isAbnormal = {True, False, False, False, True}
      * LowerBoundary1(100) <= Signal1(90) <= UpperBoundary1(130) is False
        + Signal is abnormal so isAbnormal[0] = True
      * LowerBoundary2(50) <= Signal2(75) <= UpperBoundary2(100) is True
        + Signal is normal so isAbnormal[1] = False
      * LowerBoundary3(80) <= Signal3(110) <= UpperBoundary3(150) is True
        + Signal is normal so isAbnormal[2] = False
      * LowerBoundary4(20) <= Signal4(30) <= UpperBoundary4(40) is True
        + Signal is normal so isAbnormal[3] = False
      * LowerBoundary5(130) <= Signal5(10) <= UpperBoundary5(180) is False
        + Signal is abnormal so isAbnormal[4] = True
    - Results
      * Expected: "Seizure Detected: Minor (Sudden, repeated fear or anger)"
      * Result: "Seizure Detected: Minor (Sudden, repeated fear or anger)"
      * Conclusion: Test value behaves properly
  + Issue 3
    - Issue 3 is caused when isAbnormal = {True, False, True, False, True}
      * LowerBoundary1(100) <= Signal1(90) <= UpperBoundary1(130) is False
        + Signal is abnormal so isAbnormal[0] = True
      * LowerBoundary2(50) <= Signal2(75) <= UpperBoundary2(100) is True
        + Signal is normal so isAbnormal[1] = False
      * LowerBoundary3(80) <= Signal3(500) <= UpperBoundary3(150) is False
        + Signal is abnormal so isAbnormal[2] = True
      * LowerBoundary4(20) <= Signal4(30) <= UpperBoundary4(40) is True
        + Signal is normal so isAbnormal[3] = False
      * LowerBoundary5(130) <= Signal5(10) <= UpperBoundary5(180) is False
        + Signal is abnormal so isAbnormal[4] = True
    - Results
      * Expected: "Seizure Detected: Minor (Repeated, unusual movements such as head nodding or rapid blinking)"
      * Result: "Seizure Detected: Minor (Repeated, unusual movements such as head nodding or rapid blinking)"
      * Conclusion: Test value behaves properly
  + Normal
    - Normal occurs when isAbnormal does not match any Issue Pattern (i.e. All False)
      * LowerBoundary1(100) <= Signal1(110) <= UpperBoundary1(130) is True
        + Signal is normal so isAbnormal[0] = False
      * LowerBoundary2(50) <= Signal2(75) <= UpperBoundary2(100) is True
        + Signal is normal so isAbnormal[1] = False
      * LowerBoundary3(80) <= Signal3(110) <= UpperBoundary3(150) is True
        + Signal is normal so isAbnormal[2] = False
      * LowerBoundary4(20) <= Signal4(30) <= UpperBoundary4(40) is True
        + Signal is normal so isAbnormal[3] = False
      * LowerBoundary5(130) <= Signal5(155) <= UpperBoundary5(180) is True
        + Signal is normal so isAbnormal[4] = False
    - Results
      * Expected: "Normal”
      * Result: "Normal”
      * Conclusion: Test value behaves properly

### White-Box Testing Conclusions

All test performed as expected. Every path the program was tested and performed each operation as expected.

## Randomly Generated Signals Test

The Signals are determined using a random number generator between 0 and 200. These values are compared with the default Patient Stat Ranges. If the Signal’s random number is out of its corresponding range, then it is determined to be abnormal. The abnormalities are compared with the three default Issue Patterns. The randomly generated signals represent what the program would be receiving once implemented. Thus, it is important to run and make sure the program functions accordingly. All random tests we run produced the expected output.

## Testing Conclusions

1. All the Tests Passed
   1. The program operates as expected
2. Verification Confirmed
   1. All workflows were completed correctly
3. Validation Confirmed
   1. The product satisfies the requirements
4. Program is Ready for Integration
   1. Move to UML stage to determine compatibility with the previous team.