**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)

# 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 isAbnormal Boolean value
      * Issue class must be able to set values for the pattern and message
      * Seizure class does not have any setters.
    - Test getters to verify proper attribute values
      * Signal class must be able to get the value for the isAbnormal Boolean value
      * Issue class must be able to get values for the pattern and message
      * Patient does not have any getters.
  + All the setters and getters behaved properly

# White-Box Testing

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

## Test Entire Program with Default Test Values.

### 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}
      * Signal1(True)
        + Signal is abnormal so isAbnormal[0] = True
      * Signal2(True)
        + Signal is abnormal so isAbnormal[1] = True
      * Signal3(True)
        + Signal is abnormal so isAbnormal[2] = True
      * Signal4(True)
        + Signal is abnormal so isAbnormal[3] = True
      * Signal5(True)
        + 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}
      * Signal1(True)
        + Signal is abnormal so isAbnormal[0] = True
      * Signal2(False)
        + Signal is normal so isAbnormal[1] = False
      * Signal3(False)
        + Signal is normal so isAbnormal[2] = False
      * Signal4(False)
        + Signal is normal so isAbnormal[3] = False
      * Signal5(True)
        + 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}
      * Signal1(True)
        + Signal is abnormal so isAbnormal[0] = True
      * Signal2(False)
        + Signal is normal so isAbnormal[1] = False
      * Signal3(True)
        + Signal is abnormal so isAbnormal[2] = True
      * Signal4(False)
        + Signal is normal so isAbnormal[3] = False
      * Signal5(True)
        + 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)
      * Signal1(False)
        + Signal is normal so isAbnormal[0] = False
      * Signal2(False)
        + Signal is normal so isAbnormal[1] = False
      * Signal3(False)
        + Signal is normal so isAbnormal[2] = False
      * Signal4(False)
        + Signal is normal so isAbnormal[3] = False
      * Signal5(False)
        + 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 Boolean generator that sets each signal as true or false. These values 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.