

WEEK 4: Unit Test Verification

```
package com.alerts;

import com.data.management.Patient;
import com.data.management.PatientRecord;
import org.junit.jupiter.api.BeforeEach;
import org.junit.jupiter.api.Test;
import org.junit.jupiter.api.DisplayName;
import org.junit.jupiter.api.Assertions;

import java.util.List;

import static org.junit.jupiter.api.Assertions.*;

public class BloodSaturationAlertStrategyTest {
    private BloodSaturationAlertStrategy strategy;
    private Patient testPatient;

    @BeforeEach
    void setUp() {
        strategy = new BloodSaturationAlertStrategy();
    }

    // ... other test methods ...
}
```

Run: BloodSaturationAlertStrategyTest

Tests passed: 11 of 11 tests - 10 ms

- Should trigger low sat
- Should filter out non
- Should not trigger repeat msg
- Should handle BloodSaturation
- Should trigger both
- Should not trigger repeat msg
- Should trigger repeat msg

```
package com.alerts;

import com.data.management.Patient;
import com.data.management.PatientRecord;
import org.junit.jupiter.api.BeforeEach;
import org.junit.jupiter.api.Test;
import org.junit.jupiter.api.DisplayName;
import org.junit.jupiter.api.Assertions;

import java.util.List;

import static org.junit.jupiter.api.Assertions.*;

public class ECGAlertStrategyTest {
    private ECGAlertStrategy strategy;
    private Patient testPatient;

    @BeforeEach
    void setUp() {
        strategy = new ECGAlertStrategy();
    }

    // ... other test methods ...
}
```

Run: ECGAlertStrategyTest

Tests passed: 8 of 8 tests - 13 ms

- Should return empty
- Should use isEcg when
- Should filter out non-ECG re
- Should not trigger alert for n
- Should limit to one alert per
- Should return strategy name
- Should handle zero values in

```
package com.alerts;

import ...

public class HypotensiveHypoxemiaAlertStrategyTest {
    private HypotensiveHypoxemiaAlertStrategy strategy;
    private Patient testPatient;

    @BeforeEach
    void setUp() {
        strategy = new HypotensiveHypoxemiaAlertStrategy();
        testPatient = new Patient("John Doe", 45);
    }

    @Test
    void test() {
        // ... test logic ...
    }

    @Test
    void testStrategyName() {
        assertEquals("HypotensiveHypoxemiaAlertStrategy", strategy.getStrategyName());
    }
}
```

Run: HypotensiveHypoxemiaAlertStrategyTest

Tests passed: 9 of 9 tests - 15 ms

- Should return empty
- Should handle Blood
- Should return strateg
- Should not trigger alert when
- Should not trigger alert when
- Should return empty alerts w

