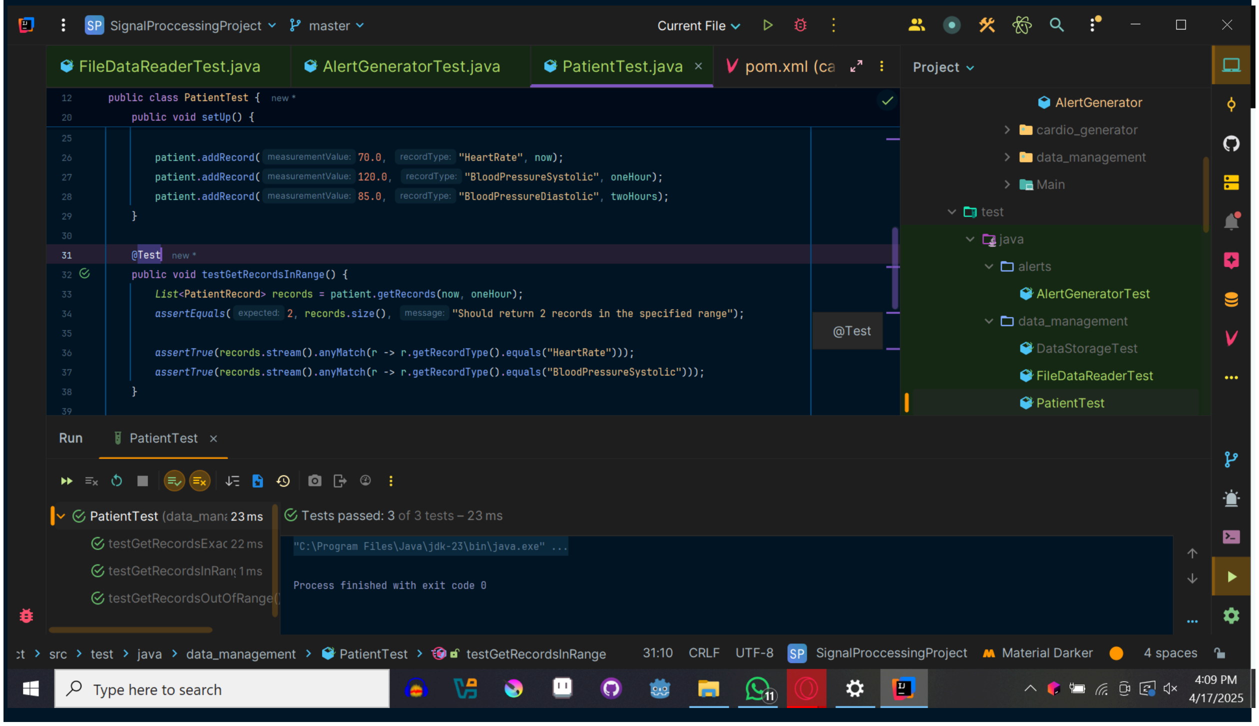
```
Code Coverage Report

-------------------------------------------------------------------------------
Test set: alerts.AlertGeneratorTest
-------------------------------------------------------------------------------
Tests run: 11, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 0.100 s -- in
alerts.AlertGeneratorTest
```

```
-------------------------------------------------------------------------------
Test set: data_management.DataStorageTest
-------------------------------------------------------------------------------
Tests run: 1, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 0.006 s -- in
data_management.DataStorageTest
```

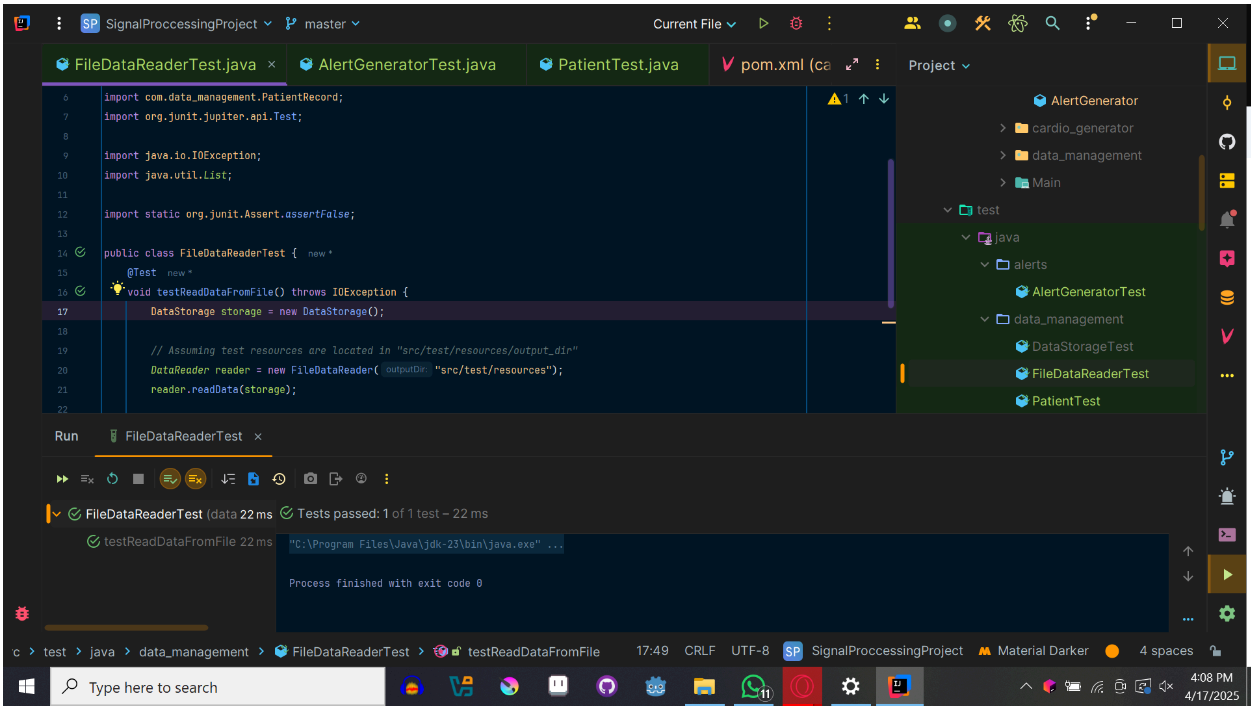
```
-------------------------------------------------------------------------------
Test set: data_management.FileDataReaderTest
-------------------------------------------------------------------------------
Tests run: 1, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 0.011 s -- in
data_management.FileDataReaderTest
```

```
-------------------------------------------------------------------------------
Test set: data_management.PatientTest
-------------------------------------------------------------------------------
Tests run: 3, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 0.010 s -- in
data_management.PatientTest


Test Cases Screenshots:
```



```java
public class PatientTest {
    public void setUp() {

        patient.addRecord(70.0, "HeartRate", now);
        patient.addRecord(120.0, "BloodPressureSystolic", oneHour);
        patient.addRecord(85.0, "BloodPressureDiastolic", twoHours);
    }

    @Test
    public void testGetRecordsInRange() {
        List<PatientRecord> records = patient.getRecords(now, oneHour);
        assertEquals(2, records.size(), "Should return 2 records in the specified range");

        assertTrue(records.stream().anyMatch(r -> r.getRecordType().equals("HeartRate")));
        assertTrue(records.stream().anyMatch(r -> r.getRecordType().equals("BloodPressureSystolic")));
    }
```

```
PatientTest (data_man: 23 ms
    testGetRecordsExac 22 ms
    testGetRecordsInRan( 1 ms
    testGetRecordsOutOfRange(

Tests passed: 3 of 3 tests – 23 ms

"C:\Program Files\Java\jdk-23\bin\java.exe" ...

Process finished with exit code 0
```



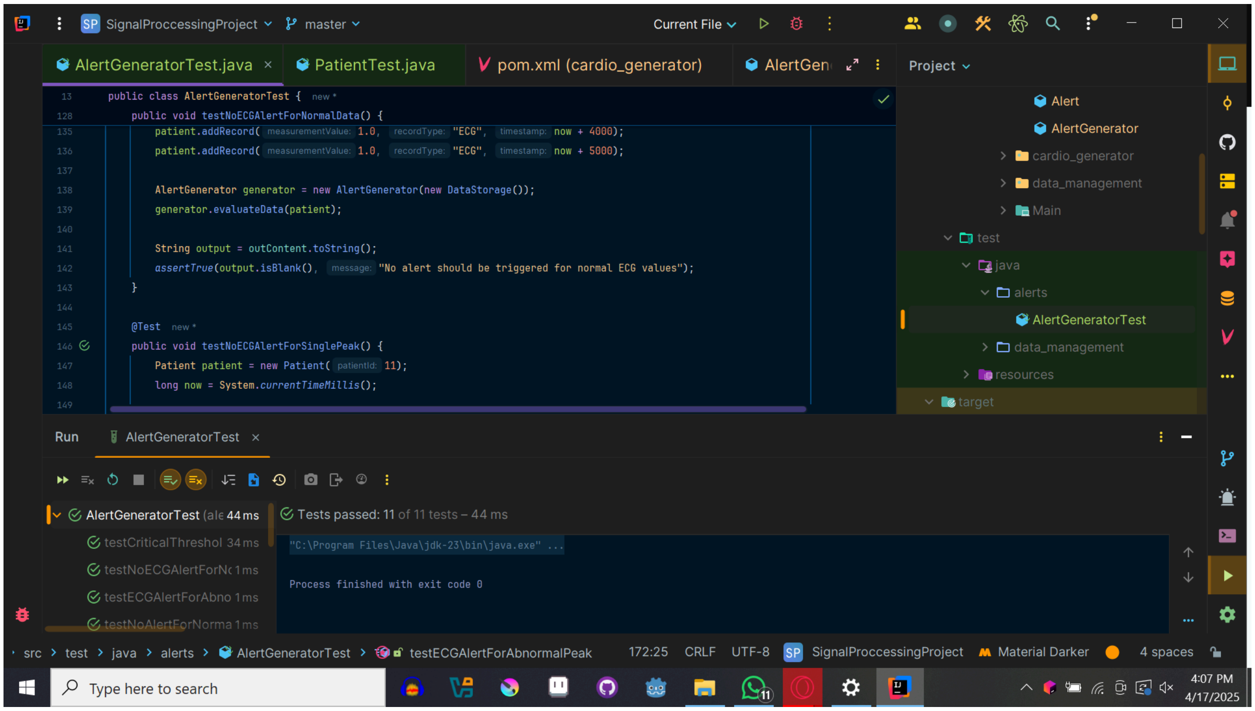
```java
import com.data_management.PatientRecord;
import org.junit.jupiter.api.Test;

import java.io.IOException;
import java.util.List;

import static org.junit.Assert.assertFalse;

public class FileDataReaderTest {  new *
    @Test  new *
    void testReadDataFromFile() throws IOException {
        DataStorage storage = new DataStorage();

        // Assuming test resources are located in "src/test/resources/output_dir"
        DataReader reader = new FileDataReader( outputDir: "src/test/resources");
        reader.readData(storage);
```

FileDataReaderTest (data 22 ms

Tests passed: 1 of 1 test – 22 ms

testReadDataFromFile 22 ms

```
"C:\Program Files\Java\jdk-23\bin\java.exe" ...

Process finished with exit code 0
```



```java
public class AlertGeneratorTest {  new *
    public void testNoECGAlertForNormalData() {
        patient.addRecord( measurementValue: 1.0,  recordType: "ECG",  timestamp: now + 4000);
        patient.addRecord( measurementValue: 1.0,  recordType: "ECG",  timestamp: now + 5000);

        AlertGenerator generator = new AlertGenerator(new DataStorage());
        generator.evaluateData(patient);

        String output = outContent.toString();
        assertTrue(output.isBlank(),  message: "No alert should be triggered for normal ECG values");
    }

    @Test  new *
    public void testNoECGAlertForSinglePeak() {
        Patient patient = new Patient( patientId: 11);
        long now = System.currentTimeMillis();
```

AlertGeneratorTest (ale 44 ms

Tests passed: 11 of 11 tests – 44 ms

testCriticalThreshol 34 ms

testNoECGAlertForNc 1 ms

testECGAlertForAbno 1 ms

testNoAlertForNorma 1 ms

```
"C:\Program Files\Java\jdk-23\bin\java.exe" ...

Process finished with exit code 0
```