

Explain the purpose of the Test (what the original test exposed, and what your test expose)

The original tests contain two major test cases, one testing recognition of a single numberplate using an image and the second checking many images to a results file. The s

Explain about Parameterized Tests in JUnit and how you have used it in this exercise.

Parameterized tests allow several inputs to be put into a single method. This allows for quicker test creation and testing. You simply have to add parameters to a table/file and then a new test case is created.

Explain the topic Data Driven Testing, and why it often makes a lot of sense to read test-data from a file.

Data driven testing uses parameterized tests to test code using data that is read from a file rather than hard coded values. This allows for customers to participate in the testing phase as in order to add tests you simply have to add data to a table.

Your answers to the question; whether what you implemented was a Unit Test or a JUnit Test, the problems you might have discovered with the test and, your suggestions for ways this could have been fixed.

I implemented a JUnit test which is a unit test. JUnit is a framework used to create unit tests. The test seemed to work with no problems.

The steps you took to include Hamcrest matchers in the project, and the difference they made for the test

The way I included hamcrest matchers is by importing the hamcrest library. I also added hamcrest to the pom.xml file. The difference the matchers did are that the tests are more articulate. Reading the tests out actually makes sense as opposed to normal JUnit tests. Also the results of the tests were clearer. If a test failed it was easy to localize.

```
<dependency>
  <groupId>org.hamcrest</groupId>
  <artifactId>hamcrest-junit</artifactId>
  <version>2.0.0.0</version>
  <scope>test</scope>
</dependency>
```