Random Test Data Generation

Specification Document



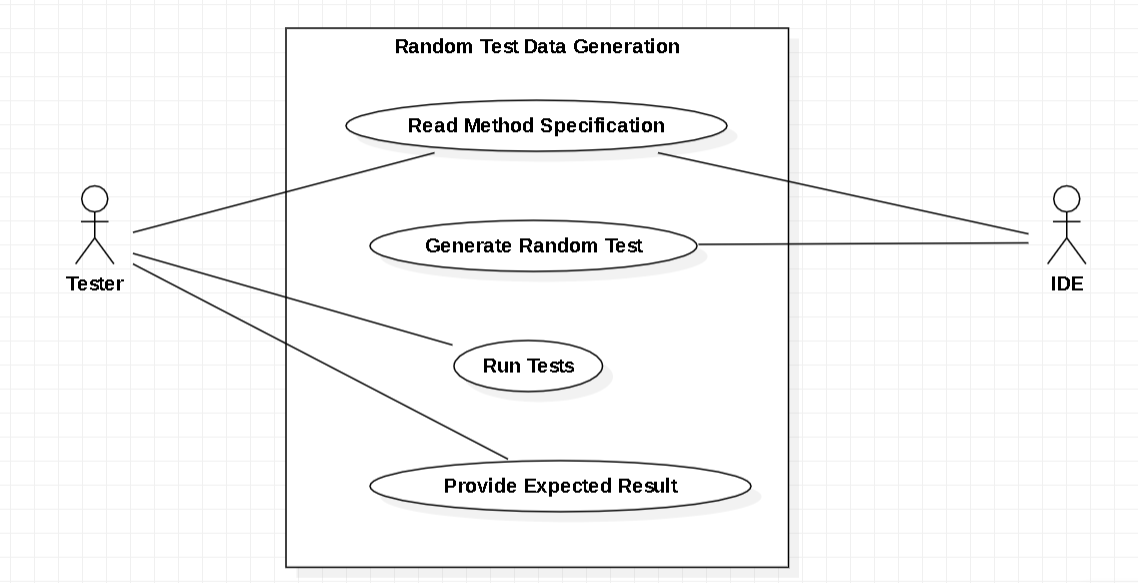
Course: Bachelor of Science (Honours) in Software Development

Name: Kyle Kinsella

Date when Specification Document Started: 22/01/2024

Supervisor: Chris Meudec

Use Case Diagram



Brief use cases

**Name** - Read Method Specification

**Actor** - Tester

**Description** - The tester will have to select a test in order to test it, then the tool will retrieve the information about whatever test that has been selected and specified and the tool will tell the tester what information it has gathered, such as the name of the method, the return type and how many parameters the method has.

**Name** - Run Tests

**Actor** - Tester

**Description** - When the tester wishes to run the tests that have been generated. The tester will select the desired test suite and indicate that they want to run that test. The system will launch JUnit using an IDE, then the test will run in the normal manner.

**Name** - Provide Expected Result

**Actor** - Tester

**Description** - When the tester is running a test they are expecting for the tool to automatically show the tester what tests are generated.

**Name** - Read Method Specification

**Actor** - IDE

**Description** - The IDE is connected to my tool via a desktop extension. The IDE must be able to read the method specification to get some information that will be used later on.

**Name** - Generate Random Test

**Actor** - IDE

**Description** - When the IDE has received the information that it needs to function, the IDE will be able to generate a random test for the specific information that it has.

Non-Functional Requirements

1. Functionality - The tool must have the capability to generate a minimum of twenty-five unique and random test’s within five minutes.
2. Usability - When the tester is testing a specific task, the average time should be less than two minutes.
3. Reliability - My tool should have less than a 1% failure rate when generating tests.
4. Performance - My tool should have a very fast response time of about less than 2.5 seconds per test, when generating a test.
5. Supportability - My tool should be able to maintain and support a minimum of twenty-five concurrent testers, ensure optimal performance and responsiveness.

Gui Mock-up

