Software Testing and Debugging

**Milestone 2 - Lessons**

horizontal line

# Introduction

In this report I will detail the things i gained from developing tests and structure for the Android application MinimalTODO. For an android application that did not have any unit tests there were many challenges surrounding the infrastructure and using several tools for the first time.

### Using Mocks, Fakes, JUnit and Simulating Behavior:

One of the learnings in this whole project was to understand and utilize the use of mock objects, fake objects and then controlling the behaviour of the mocks to test for all possible cases for failure and testing developer behavior. For example if there is a part of the code where I am handling exceptions, I could ensure that the exception is triggered and handled appropriately.

### Using the Android Framework and Gradle:

Another positive from this project was understanding how to trigger and build test cases for the entire project. Also, understanding scripting and automation made me realize how powerful test automation can become.

### Coverage alone is not enough:

Another lesson that was reinforced in my head is that code/test coverage alone does not mean anything and cannot uncover any bugs.

### We cannot test badly written Code:

I learnt that tests are not a cover for badly written code. In this project I had to change several classes to get them to be test compatible. For example, using dependency injection and construction rather than creating new objects inside methods for example, which can become hard to examine and test.