

Software Testing Portfolio

B241143

Outline of Software being Tested

The role of the project is purely a fictional one made up for the ILP course, which this year posed that we were building a medicine delivery service that utilised drones.

These deliveries are from Service Points to arbitrary locations. Drone paths must avoid certain no-fly zones and move with certain other constraints. Each medicine dispatch has attributes (needs heating, needs cooling, is a certain size, etc) that a drone must fulfil to deliver that medicine (is capable of heating, is capable of cooling, has capacity, etc).

This project is a REST service that allows a user to query a variety of endpoints that can be broadly split into:

- Drone Navigation (delivering the medicine)
- Drone Attribute Querying (which drones can deliver which medicine)

For this portfolio, I focused on the drone navigation, as this was the more interesting part to not only implement during the coursework, but test as part of this portfolio.

Project link: https://github.com/Espian05/ilp_cw

Learning Outcomes

LO 1 - Analyse requirements to determine appropriate testing strategies (20%)

1. Range of requirements, functional requirements, measurable quality attributes, qualitative requirements - [LO1.1](#)

I identified a wide range of requirements from all three mentioned sections, briefly explaining what the requirement is, and splitting those requirements into the sections mentioned.

2. Level of requirements, system, integration, unit - [LO1.2](#)

I split the range of requirements from 1.1 into system, integration, and unit testing. I split larger requirements into smaller requirements where necessary, or place the same requirement in multiple sections.

3. Identifying test approach for chosen attributes - [LO1.3](#)

Here, I pick out 7 core requirements to focus on. This is perhaps a larger number than usual, however many are closely linked and are able to reuse the same scaffolding. When I deemed it necessary, rather than talking about the strategy for each individual requirement, I grouped them and wrote about how I would address that group, as I felt otherwise I would be needlessly repeating myself.

4. **Assess the appropriateness of your chosen testing approach - [LO1.4](#)**

In this document I briefly reflect on the nature of the strategy I have chosen, providing justification as to my decisions thus far.

LO 2 - Design and implement comprehensive test plans with instrumented code (20%)

1. **Construction of the test plan**

Please refer to the [Test Planning Document](#)

2. **Evaluation of the quality of the test plan**

Please refer to the "Limitations of the test plan" section of the [Test Planning Document](#)

3. **Instrumentation of the code**

Please refer to the "Instrumentation and Scaffolding" section of the [Test Planning Document](#)

4. **Evaluation of the instrumentation**

Please refer to the "Evaluation of Instrumentation" section of the [Test Planning Document](#)

LO 3 - Apply a wide variety of testing techniques and compute test coverage and yield according to a variety of criteria (15%)

1. **Range of techniques - [LO3.1](#)**

I identify which techniques I use, and justify why I chose those and not others.

2. **Evaluation criteria for the adequacy of the testing - [LO3.2](#)**

I identify the criteria for which I'm going to evaluate the adequacy of my testing, and justify these criteria against each level of testing I've performed.

3. **Results of testing - [LO3.3](#)**

I briefly go through the results of all of my testing, showing that all tests pass by providing screenshots of the results. I also provide evidence of all tests passing at once, and show my file structure/naming convention.

4. **Evaluation of the results - [LO3.4](#)**

I evaluate the results of the testing vs my identified evaluation criteria for each level of testing I do, justifying why they do or don't meet these criteria, and how they may go over and above the criteria stated.

LO 4 - Evaluate the limitations of a given testing process, using statistical methods where appropriate, and summarise outcomes (25%)

1. **Identifying gaps and omissions in the testing process - [LO4.1](#)**

I identify multiple areas in which I could improve, acknowledging that whilst I largely completed what I set out to do in my Test Planning Document, there are still many areas to improve upon were this a production system.

2. Identifying target coverage/performance levels for the different testing procedures

- [LO4.2](#)

Four main areas for crucial coverage are identified, and a one sentence criteria is laid out for each one. Along with the sentence, there is also a brief explanation of *why* this area/criteria was selected.

3. Discussing how the testing carried out compares with the target levels - [LO4.3](#)

Criteria by criteria, I break down how my testing either falls short of the criteria identified in 4.2, or surpasses it. I also mention how this criteria would likely be changed in a professional setting.

4. Discussion of what would be necessary to achieve the target level - [LO4.4](#)

I discuss what changes would have to be made to both the testing suite and the code-base to achieve the targeted level of testing identified in 4.2. This mainly consists of considering theoretical changes that could be made to increase coverage and performance.

LO 5 - Conduct reviews, inspections, and design and implement automated testing processes (20%)

1. Identify and apply review criteria to selected parts of the code and identify issues in the code - [LO5.1](#)

Here, I provide a surface level code review of a crucial part of my implementation, going over weaknesses in my code quality, and identifying areas I would improve the code given the time.

2. Construct an appropriate CI pipeline for the software - [LO5.2](#)

I discuss the inspiration for my CI pipeline design, and adapt it to suit this project and the tools I have available. I also discuss where I would next take the pipeline given time.

3. Automate some aspects of the testing - [LO5.3](#)

Here, I discuss the range of errors that my pipeline covers, and how I would develop it further given a larger test suite and more requirements. I also evaluate the quality of my current pipeline against my theoretical plan.

4. Demonstrate the CI pipeline functions as expected - [LO5.4](#)

I back up the claims I made in LO 5.3 as to the errors that my pipeline covers, pushing intentionally pipeline-failing code, and providing screenshots as evidence where necessary.