Evaluation Plan

Our aim is to find the right balance between writing code and testing. It is really important to give testing enough time and importance especially that this is a group project and each individual will write their own piece of code towards the end product. Leaving obvious bugs in the code is a waste of time and a waste of other people’s time.

What we agreed on is this mix: 40% designing and writing code, 5% spent on code review and static code analysis, 25% spent on unit testing and integration testing and 30% on basic functionality testing and single performance user testing.

1. **Basic functionality testing:**

Making sure that every button on the screen works and also making sure that every field that can have text as an input will work and not crash the software.

The goal here is to not let other people touch our work if it's going to crash as soon as they enter their own name in the username field. We will have an API feature so it will be tested that the basic API functionality works well before submitting it for more intensive testing.

1. **Code review:**

Before code review the basic functionality testing will be done. Everyone in the group will have to look at the code, even if it's not the code they wrote. Another pair of eyes looking at the source code can uncover a lot of problems.

1. **Static code analysis:**

We are going to use tools that will perform analysis on the source code without executing it. These static code analysis will look for weaknesses such as security vulnerabilities and potential concurrency issues.

1. **Unit testing:**

Each of us will write unit tests to make sure that the unit (be it method, class, or component) is working as expected and tested across a range of valid and invalid inputs.

We can also work with mock objects and virtualized services.

1. **Single-user performance testing:**

Single-user performance on the front-end and making sure the software is responsive when only they are using the system. If it's taking more than a few seconds to display the web page or pull the products from the database then the client-side code is slowing things down and has to be fixed.