***Test plan***

**Tim Verhees**

*04-11-2022   
V0.1*

# **Versions**

|  |  |  |
| --- | --- | --- |
| **Date** | **Version** | **Description** |
| **04-11-2022** | 0.1 | Initial version |

# **Distribution list**

|  |  |  |  |
| --- | --- | --- | --- |
| **Date** | **Version** | **Name** | **Function** |
| **04-11-2022** | 0.1 | Tim Verhees | Project Handler |
|  |  | Maja | Technical Teacher |
|  |  | Jacco | Technical Teacher |

# **Test report**

Testing your application is important from several aspects. Namely Quality Assurance, Iterative Development and especially Agile Development. Testing your application helps you throughout the process to make sure everything works after making changes, so that the final version is a fully functional program. The are many ways to test an application and in this document I aim to outline which I use for my individual project.

Method

The way I will test my application is through Continuous Integration and human testing. Continuous Integration will be done and achieved through unit testing. This is a method of testing that is built into the program and is a set of testing functions that are executed to check whether the functions in the application work. The advantage of this is that it can be automated through a runner. This runner will then run all the unit tests once I push all my changes to the repository. Meaning everything I commit a change to my application, all already accounted for functions will be tested automatically.

Now this is of course only half the work, bugs can still occur under certain circumstances that the unit tests do not account for. This is where the human testing aspect comes in. Here, I (or a fellow student for example) run through the application and test all of it’s features in various ways. This includes, but is not limited to;

* Wrongful input
* Attempted overloading
* SQL Injection
* Zooming in on the page

These are just some of the examples that can and will be used when testing the application through human testing. It will help stop the end user from breaking the application.

User Experience

Another important thing to test is the actual user experience. Does it feel good to use the application? Is it self-explanatory, or is it too complicated at face value? Testing this through letting other users use your application will help you gain perspective from and end-user point of view and helps you in forming a complex yet useable application.

This doesn’t only include the overall design but also the look and feel. Because they can take away from the usage of the application when not done right. The way these things will be tested is once again with human testing, where I will have some of my classmates use the application and give their feedback on its ease-of-use and look.