

# **Software Implementation and Testing Document**

**For**

**Group 7**

Version 1.0

**Authors:**

Meghan C  
Madelyn Y  
Jacqueline V

## **1. Programming Languages (5 points)**

HTML and CSS were needed to make the front end of the webpage. C# is something that two out of three of us have used before to make a full-stack application, and we figured it was pretty easy to pick up. SQL was used for its querying and database qualities because it is something that (usually) has pretty full integration and is easy to pick up as well.

## **2. Platforms, APIs, Databases, and other technologies used (5 points)**

We used Razor framework in our webpage, so we could use HTML and CSS to eventually get to a cross-platform application. We used .NET because we had experience with it. We used MySQL for our database to query and run through Docker. We used MudBlazor to get components and styling for our application, it just makes things look a bit nicer. We used Ngork to forward the port to a remote server.

## **3. Execution-based Functional Testing (10 points)**

We performed functional testing for our project by manually interacting with it. We had many users created but focused on testing with one testuser. We added, edited, and deleted habits for this user. We entered and saved notes, then deleted notes and logged in to make sure everything was working correctly. We also verified that the motivational quotes worked correctly and would refresh every session or manually. We also used the API on a different port to test different calls.

## **4. Execution-based Non-Functional Testing (10 points)**

We performed non-functional testing by checking our load times and adding a loading screen so you don't see old data. We also verified that everything was displayed correctly and promptly when grabbed from the API. We tested security by making sure that users would get 401 errors when accessing unavailable areas. We also made sure all inputs were clear and provided clear error messaging.

## **5. Non-Execution-based Testing (10 points)**

We did code reviews on our personal Increment branches before going into main, or if we had any specific questions or iffiness on something. We also met up to go over goals, what was expected of each member, and blockers. If we learned something, we were sure to walk through it with the group. We also looked through all our compiler warnings and made sure to clear them for things like static variables and null values.