**Software Implementation and Testing Document**

**For**

**Group 8**

Version 1.0

**Authors**:

Marc Boustani

Bryan Dean

Carlos Saucedo

Carson Walker

Tomas Munoz-Moxey

# Programming Languages (5 points)

*List the programming languages use in your project, where you use them (what components of your project) and your reason for choosing them (whatever that may be).*

C#, MSSQL, and HTML/CSS. Pretty much everything is written in C# thus far, frontend and backup. MSSQL is used for the database management. HTML/CSS is used for the front end and styling. We chose to use C# because of our choice to use Blazor, which requires C# to make the frontend easier to work with/manage. We chose to use MSSQL because of it’s integration with Windows and ASP.NET Identity. We chose to HTML/CSS because Bootstrap comes with it, which is implemented in Blazor.

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

*List all the platforms, APIs, Databases, and any other technologies you use in your project and where you use them (in what components of your project).*

Google Maps API, BrowserInterop, Visual Studio, Blazor, MSSQL, ASP.NET Identity. The Google Maps API is used to open the coordinate locations supplied by the users. BrowserInterop is used to get the current user’s location. Visual Studio is the IDE where we have been writing the source code. Blazor is being used to create the actual web application. MSSQL is being used for database management. ASP.NET Identity is being used to handle user accounts, login, registration, password encryption, and authorization.

# Execution-based Functional Testing (10 points)

*Describe how/if you performed functional testing for your project (i.e., tested for the* ***functional requirements*** *listed in your RD).*

We have performed some functional testing of our functional requirements. As of currently, functional requirements 1 – 4 and 6 – 7 are working as they should be, with plans to update/upgrade their functionality in the coming weeks. Functional dependency #5, user social stories, is not currently functional, but that is one of our next goals for the project.

# Execution-based Non-Functional Testing (10 points)

*Describe how/if you performed non-functional testing for your project (i.e., tested for the* ***non-functional requirements*** *listed in your RD).*

We have performed some non-functional testing of our non-functional requirements. We have not met some of the requirements, such as requirement #1, encrypting user passwords for security/safety reasons, but we are working towards that goal. Also, requirement #6, email verification, is not functional as well. Additionally, as of now, the rest of the non-functional requirements seem to be met; limiting user interactivity, load time reliability, perform hindering, and maintainability.

# Non-Execution-based Testing (10 points)

*Describe how/if you performed non-execution-based testing (such as code reviews/inspections/walkthroughs).*

We have done walkthroughs of the source code as a group to ensure we all have a good understanding of what is going on and what each team member has been working on. We also have been inspecting and reviewing the source code for any potential bugs as we work through the code.