**Software Implementation and Testing Document**

**For**

**Group 8**

Version 3.0

**Authors**:

Amelia Sieg

Madalina Bulat

Julia Sveen

Reagan Bourne

# 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).*

For the backend aspect, we used Python. For the frontend, we used React, which is a specific JavaScript library (with CSS). We decided to use Python for the backend because we felt as if it would be the more efficient language to use to effectively develop a website. Python also works well with APIs, which we are using for this project. We used React for the frontend portion because it runs on a browser, which makes it very efficient and easy to comprehend.

# 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).*

We all used Visual Studio Code to code, debug, and test the code, as well as the online browser for React. We also used the Yelp Fusion API for the restaurant names, locations, phone numbers, ratings, and more. We also followed a youtube video on how code the robot. We also attempted to utilize Heroku, a cloud platform service supporting several programming languages, to host and deploy our website, but we were unsuccessful. For this project, it is also necessary to have downloaded Flask, Node.js, and everything needed for React.

# 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).*

For the execution-based functional testing we ran the program and entered false input for the city name and food category to ensure that the program would give us an error message. We also made sure that when we entered correct input for the city name and food category, that the program outputted the correct information (the restaurant name, the address etc.). The program also allows the user to click on the website for the restuant that they would like to visit.

# 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).*

For the execution-based non-functional testing, we ran the program to ensure that our graphics (the waving robot) and the output printed and looked as expected. To make it easier to read we printed out the results one by one and allow the user to click yes/no depending on whether or not they want to eat at that specific restaurant. Once the user clicked yes, confetti explodes indicating an happy robot.

# Non-Execution-based Testing (10 points)

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

For non-execution-based testing we all collectively stepped through each source code file line by line to completely understand where it came from and how it all works. We also went through each file to ensure that we are following proper programming practices and formatting. We also reviewed the code to make sure all requirments were met and did error checking along the way.