**Software Implementation and Testing Document**

**For**

**Group <X>**

Version 1.0

**Authors**:

Jayna Spikes

Donald Walton

Brunas Joseph

Mihye Lim

Kesnel Mezinnord

# 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).*

# 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).*

Backend:

* **MySQL**:
  + Used to create and manage the relational database.
  + Contains tables such as restaurants, categories, restaurantCategories, and account.
  + Stores user account information, restaurant details, and category relationships for efficient data retrieval.
* **Yelp API**:
  + Used to fetch restaurant data, including names, locations, ratings, and categories.
  + Integrated into the backend to populate the database with relevant restaurant information.
  + API key security was ensured through environment variables.

# 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).*

Backend: we tested out the if the Api’s were communicating in our MySQL database. We wrote simple queries in python to see how the MySQL database is responding with the information.

*Backend:*

 **API Communication with MySQL Database:**

* We tested whether the API correctly inserted, retrieved, updated, and deleted data from the MySQL database.
* Sample API calls were made to verify that the expected responses matched the actual stored data.

 **Query Validation in Python:**

* We wrote and executed SQL queries using Python to check the database’s response to different inputs.
* Queries were tested for retrieving restaurant data, filtering by categories, and checking account authentication.

# 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).*

Backend:

During the development of the database and API integration, we implemented various non-functional testing strategies to ensure security, performance, and reliability.

* **Security Testing**: In the process of integrating the Yelp API, we needed to protect the API key to prevent unauthorized access and potential abuse. To achieve this:
  + We stored the API key in environment variables instead of hardcoding it in the codebase.
  + The API key was never exposed in public repositories, ensuring it remained confidential.
* **Performance Testing**:
  + We tested the response times of database queries to ensure efficient retrieval of data.
* **Reliability and Fault Tolerance**:
  + Backup and recovery procedures were tested to ensure data persistence in case of failures

Frontend:

* **Usability Testing**:
  + We gathered user feedback on the UI/UX design to enhance ease of use and navigation.
  + Accessibility testing was performed to ensure compatibility with screen readers and different devices.
* **Compatibility Testing**:
  + The application was tested across different browsers (Chrome, Firefox, Safari) and devices to verify proper rendering and functionality.

# Non-Execution-based Testing (10 points)

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

Code reviews:

* Reviews focused on identifying syntax errors, logical inconsistencies, redundant code, and potential security vulnerabilities.
* Git version control was used to track changes, and review commits before