**Software Implementation and Testing Document**

**For**

**Business Survey Analytics**

Version 1.3

**Authors**:

Samuel Anderson

Michael Fogarty

Joseph Fields

Trey Hamilton

Luis Corps

# Programming Languages (5 points)

UI:

CSS3

* Is used to style our web pages, and provide a seamless and clear experience for our users.

JavaScript

* Is used to add interactivity to the web pages for our users, establish routes with a server, query the database, add logic to server or client side pages, and to make templates.
* Also used in client side scripts to make requests to the database.

HTML5 -> EJS

* Is used to provide the structure for our web pages, prompt users for input, add emphasis or importance to that structure, and organize the web page in a way that promotes readability and meaning.
* Put html code into ejs files so we can load dynamic javascript and give our project “views”.

Backend:

MySQL

* Is used to query and structure the database.

Shell

* Is used to conduct testing as well as set up environment variables and make using the database easier.

*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)

# Database: MySQL Server

# Server: Node.js (using packages expressjs and mysqljs, managed by npm)

Libraries: jQuery - to make manipulating the DOM quicker and easier, and to enhance the readability of code

Portion of code borrowed from Jon Kantner for animation on ‘waiting.ejs’ and ‘waiting.css’ through: https://freefrontend.com/css-loaders/

*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).*

# 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).*

The majority of testing on the front end will be black box testing. Testing will be based on the navigation of the site and making sure the site is functional. An example would be making sure every page that has navigational buttons correctly navigates the user to the designated page. So far, all cosmetic aspects of the UI have been tested. The process of creating a survey as a Business Owner would, and answering surveys based off surveyID as a customer would has been thoroughly tested to ensure our product works for our clients as they expected . The testing on the backend is conversely glass box. The group has created a number of SQL procedures that return different necessary clusters of data. A javascript file has queries that display the results of testing these procedures on all the tables and checks are made confirming the results of the query and the table are correct. Additionally, a shell script has been created to test the procedures independently of the server. Also, tables “questions,” “answers,” and “surveys\_created” are tested with the server, using get requests.

# 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).*

Time-based non-functional requirements are tested when navigating the site. Additionally, to test permission requirements we will log in as each role (admin, business owner, survey taker) and make sure the data they can access is limited, for example a survey taker should not be able to access the database. We also created a number of accounts to test the functionality of the login system, and used the site as each different role to simulate what the user experience would be like.

# Non-Execution-based Testing (10 points)

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

We have been having weekly meetings to review code and demo to one another our individual contributions to the code. We also had a group chat we used for more irregular communication which we mainly used for communicating what procedures to retrieve data were needed for the front end to work properly.

Towards the end of the project, we were having issues with database queries and redirecting to the analytics page. We connected on a Zoom call and shared screens to troubleshoot the issue as a group.