

CSC 179

Summer 2022

Project Name:

Sponsor: Vendia

Team Name: Red Shift

Team Members: Santiago Bermudez,
Sabeeha Baqui, Steven Graham, Hardev
Singh, Ivan Gutierrez, Eric Truong, Alex
Tran

Date: 07/08/22

Table of Contents:

1. Introduction

2. Objectives

3. Testing Scope

 In-Scope

 Out of Scope

4. Test Strategies/Techniques

5. Test Design

6. Testing Environment

7. Testing Tools

8. Defects

1. Introduction

The objective of this test plan is to verify that the functionality of our website functions properly according to our specifications. The tests we have listed will be used to identify any bugs or defects and will be corrected when found.

Our final product will work toward the following:

- A working dashboard
- Must utilize vendia to store employee data

Revised on:	Version:	Description:	Revised By:
June 8th, 2022	1.0	Document Created	Santiago Bermudez
June 10th, 2022	1.1	Added In-Scope Testing Details	Santiago Bermudez
June 13th, 2022	1.2	Added Out-of-Scope Testing Details Added Testing Tools	Santiago Bermudez Sabeeha Baqui
June 14th, 2022	1.3	Added Test Design Details Added Introduction	Sabeeha Baqui Hardev Singh
June 17th, 2022	1.4	Added Testing Environment Details Added Defect Details	Sabeeha Baqui Ivan Gutierrez
June 18th, 2022	1.5	Added Testing Strategies and Techniques	Hardev Singh
June 20th, 2022	1.6	Added Objectives	Eric Truong

2. Objectives

This section describes the overall testing objectives

We would like to ensure that our website is fully functional with all intended features.

- Make sure our dashboard links work as intended
- Ensure we are pulling the correct data and updating them accordingly

- Verify users are inputting correct info at the right places (no Strings in age section)

3. Testing Scope

In-Scope

Address the testing scope and the features that will be tested.

Our testing scope involves looking at and testing for the following:

- Can all data be stored through Vendia?
- Can the data be created according to a certain format with all the correct parameters and constraints?
- Can data be shared or unshared with or without restriction to other users?
- Are aggregate data averages displayed and calculated correctly?

Features that will be tested are:

- The ability for some HR representative to add employee data on a dashboard.
- The ability for HR to be able to view aggregate data for all employees.
- The ability for HR to be able to view aggregate data for a certain gender or other category.
- The ability for HR to be able to share data with someone else (*in this case a representative from an insurance company).
- The ability for HR to revoke access to shared data.

Out of Scope

List any features or components that are outside the testing scope (if any).

Some features or components that are outside of our testing scope (*or likely will be) for this project are:

- User verification and authentication.
- Password and security-related features.
- Performance and optimization testing.
- Third-party system connection.

- Using a second web app to determine whether the first web app has been set up correctly

4. Test Strategies/Techniques

Test Strategies:

- Unit Testing - We used unit testing to test inputs in our front-end and see if the outputs match our intended outputs with Jest.
- Integration Testing - We used React and Jest to test our math calculations and other functions together as a group and looked for faults in their interaction.
- System Testing - We used system testing to look at our complete integrated system as a whole to evaluate whether the system complies with our requirements.
- Acceptance Testing - We used this testing to determine whether our system would meet consumer requirements.

5. Test Design

Considering the scope of the project and expectations, outline the testing that will be performed – examples

Functional Testing

- Regression testing
- Integration testing
- UI testing
- System testing
- Black-box testing
- Acceptance testing

Usability

- Moderated vs. unmoderated.
- Remote vs. in person.
- Explorative vs. comparative

GUI

- Manual testing: involves a human tester who manually checks each screen to validate functionality by creating and executing test cases
- Record and Replay testing: testers run an application and record user interaction with the app
- Model-based testing: a model is created to understand and evaluate the system's behavior
- Testing the size, position, height, width of the visual elements
- Verifying and testing the error messages are displayed or not
- Testing different sections of the display screen
- Checking the navigation elements at the top of the page, alignment of radio buttons, drop downs

6. Testing Environment

Address the testing environment, web browsers, and any existing limitations or constraints

- Test the web app in different browsers, such as Google Chrome and Firefox.
- Test the web app on different operating systems, such as Linux, Windows, and Mac.
- Use different testing environments:
 - Performance Testing Environment
 - System Integration Testing (SIT)
 - User Acceptance Testing (UAT)
 - Quality Assurance (QA)
 - Security Testing

7. Testing Tools

Describe any testing tools that might be utilized in testing the system

- Automated testing using Jest

8. Defects

Defect definition - Defect Severity, known as Bug Severity, is a classification of software defect or “bug” that indicates the degree of negative impact on the quality of software.

Document your team testing effort and defect report as follows:

Test case ID

Test case detail (procedure)

Expected output

Actual output

Defect:

Defect Severity- see defect categories below

Date found

Defect resolution - date fixed –closed date

Severity Classification -

Defect Severity Classification:

Severity 4 - Low defect -
Does not impact the functionality- Trivial.

Severity 3 - Medium defect-
issue with a specific item and a function - does not block functionality.

Severity 2- Major defect -
impact a certain module and the function is not fully working

Severity 1- Critical defect-
Impacts the most important features – not working and not functional as expected.