**Test plan for**

**Ecommerce Store**

*ChangeLog*

|  |  |  |  |
| --- | --- | --- | --- |
| **Version** | **Change Date** | **By** | **Description** |
| version number | Date of Change | Name of person who made changes | Description of the changes made |
| 1.0.0 | 03/24/2023 | Wen Chen, Aaron Le, Scott Hoerchler, Nathan Lechner | Initial set up |
| 1.1.0 | 03/29/2023 | Wen Chen, Aaron Le, Scott Hoerchler, Nathan Lechner | Began working on back end features and requirements |
| 1.2.0 | 03/31/2023 | Wen Chen, Aaron Le, Scott Hoerchler, Nathan Lechner | Create Admin Functions,  Created Cart Functions,  Created the Database table for variables |
| 1.3.0 | 04/03/2023 | Wen Chen, Aaron Le, Scott Hoerchler, Nathan Lechner | Finalized all back end features and began to unit test each feature. |
| 1.4.0 | 04/04/2023 | Wen Chen, Aaron Le, Scott Hoerchler, Nathan Lechner | Created all files needed to create a front end page and buttons required. |
| 1.5.0 | 04/05/2023 | Wen Chen, Aaron Le, Scott Hoerchler, Nathan Lechner | Updated the front end page with functions and began working on the css files to create a more appealing page. |
| 1.6.0 | 04/07/2023 | Wen Chen, Aaron Le, Scott Hoerchler, Nathan Lechner | Began focusing on unit testing and create 5 unit test per feature to make sure no errors occurred. |
| 1.7.0 | 04/08/2023 | Wen Chen, Aaron Le, Scott Hoerchler, Nathan Lechner | Began linking all back end functions to the front end. |
| 1.8.0 | 04/10/2023 | Wen Chen, Aaron Le, Scott Hoerchler, Nathan Lechner | Troubleshooting functions by adding variables and cases to reference the database. |
| 1.9.0 | 04/11/2023 | Wen Chen, Aaron Le, Scott Hoerchler, Nathan Lechner | Added more css to make all pages match. |
| 2.0.0 | 04/13/2023 | Wen Chen, Aaron Le, Scott Hoerchler, Nathan Lechner | Ran the final product and made sure everything was up to date and that we did not need to implement any more functions. |

1 Introduction 2

1.1 Scope 2

1.1.1 In-Scope 2

1.1.2 Out-of-Scope 2

1.2 Quality Objective 2

1.3 Roles and Responsibilities 2

2 Test Methodology 3

2.1 Overview 3

2.2 Test Levels 3

2.3 Bug Triage 3

2.4 Suspension Criteria and Resumption Requirements 4

2.5 Test Completeness 4

3 Test Deliverables 4

4 Resource & Environment Needs 4

4.1 Testing Tools 4

4.2 Test Environment 5

5 Terms/Acronyms 5

# Introduction

The testing strategies that we will be using is unit testing and real time user testing

## Scope

### In-Scope

We will test all the functions related to the buyer, seller, admin, and owner as well as the shopping cart. We will also be testing things such as how fast it takes to load the pages and if there are any possible security errors that occur along the way.

Buyer – Test the functionality of things such as adding an item to a cart, comparing items, and viewing items on a main page, we will also test if the buyer’s page is functioning properly and everything is going as expected.

Seller – Testing the functionality and error handling of things such as listing an item, removing a listing, setting a price for an item, and changing the quantities of each item, also making sure the front-end thing such as the buttons are linked correctly and functioning as needed.

Admin – Testing functions such as blocking users, setting sellers, and removing items from a seller’s listing. The admin should be able to access a page limited to only them with these functions and should only be allowed to be accessed by Admins.

Owner – Owner will be testing the same functionalities as the admin with another function that can set Admins. Only the owner should be able to access this page and we want to make sure that all of these functions work properly and go with the correct command and button.

We will also have functions that go along with all of the users such as deleting an account and logging out properly so that there is safety.

All of these functions will be tested by using built-in test features from our framework, Node.js.

We will also implement real-time testing by running the program ourselves and trying things that could possibly cause errors and fixing any that may occur during that time.

### Out-of-Scope

Features that will not be tested are logout because it is not really a function that cannot be tested as it simply deletes the data and sets global variable to 0.

## Quality Objective

* Make sure the ecommerce store functions work correctly and has all the requirements.
* Make sure all bugs are fixed before final launch.
* Make sure that the ecommerce store looks moderate to okay and is easy navigate.

## Roles and Responsibilities

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Net ID** | **GitHub username** | **Role** |
| Aaron Le | Adl409 | Adl409 | Project Team Leader/Test Manager |
| Wen Chen | Wxc3 | Wchen2654 | Front-End Development/ Configuration Manager |
| Nathan Lechner | Nml162 | Na7hanL | Database Creation/Back-End Development |
| Scott Hoerchler | Slh838 | Rien98 | Back-End Development |

# Test Methodology

## Overview

The testing methodology that we used was Agile. The reason that we chose this methodology is because we thought of our ecommerce store to be user interacted as well as constantly being updated in the backend. We believe that if there are any errors during final production, we should be able to fix the situation on the spot. This also makes it a more customized environment based off of what the customer wants and how it should be established.

## Test Levels

The types of test level we had were Unit Test using Node.js Jest , and Integration testing within Github

## Bug Triage

The goal of the triage is to

* Make sure queries are correctly spelled.
* Small syntax errors are fixed.
* Any function not created correctly is optimized.
* Make all interface objects such as buttons work correctly.

## Suspension Criteria and Resumption Requirements

We use Github continuous integration and the criteria we need to suspend a function would be when we need edit a test case or if a test is constantly failing and we need to figure out where this error is occurring. We resume the test when we believe the issue is resolved and run it through the continuous testing process to see if it works.

## Test Completeness

* 100% test coverage
* All Manual & Automated Test cases executed
* All open bugs are fixed or will be fixed in next release
* User is able to do all required things on the website.
* All test cases run correctly.

# Test Deliverables

Here mention all the Test Artifacts that will be delivered during different phases of the testing lifecycle.

Here are the sample deliverables

|  |
| --- |
| * Test Cases * Test Strategies * Test Results |
|  |

# Resource & Environment Needs

## Testing Tools

* Jest(A Node.js testing environment)
* Github CI testing

## Test Environment

It mentions the minimum **hardware** requirements that will be used to test the Application.

Following **software’s** are required in addition to client-specific software.

1. Windows 8 and above
2. A coding environment with the framework Node.js and its testing features
3. Databasing software MySQL as well as XAAMP to run the website.

# Terms/Acronyms

Make a mention of any terms or acronyms used in the project

| TERM/ACRONYM | DEFINITION |
| --- | --- |
| CI | Continuous Integration |
|  |  |