Test Plan For

E-Commerce Sprint 3

Version	Change Date	By	Description
version number 0.1	3/31/2025	Caleb Stockton (cls1268)	Created document and completed sect. 1.1 (Scope)
version number 0.1.1	3/31/2025	Caleb Stockton (cls1268)	Began on sect. 1.3 (Roles and Responsibilities) and added to sect. 5 (Terms/Acronyms)
version number 0.2	4/12/2025	Caleb Stockton (cls1268)	Completed sect. 1.2 and all of section 2
version number 1.0	4/13/2025	Caleb Stockton (cls1268)	Completed remainder of Test Plan Document

1 Introduction

Brief introduction of the test strategies, process, workflow and methodologies used for the project

1.1 Scope

1.1.1 In-Scope

In-Scope features are the requirements of the software that will be tested. Here is all of the requirements that will be tested:

-- **Account Management**
- Registering for an account
- Logging in

-- **Buyer**
- Searching for items
- Viewing items
- Adding items to cart
- Removing items from cart
- Viewing items in cart
- Checking out
- Viewing previous purchases

-- **Seller**
- Listing items
- Removing items
- View their “for sale” items
- Edit their “for sale” items

-- **Admin Role**
- View new users
- Approve new users
- Deny new users
- Remove items for sale

-- **Cart Class**
- Item must be added to cart table when buyer adds it to cart
- Item must be removed from cart table when buyer removes it from cart
- All items must be removed from cart (table) when purchase goes through

-- **Order Class**
- All transactions information is properly stored in order table when purchase goes through

-- **Performance**
- Users must be able to login/logout within 2-3 seconds
- Users must be able to search through our catalogue (up to 50k items) within 2-3 seconds
- Website must load product details within 2-3 seconds after user clicks on item
- User’s checkout should be processed within 2-3 seconds
- Website should be able to handle up to 50 concurrent users

- Website should be able to handle up to 10 transactions per minute

-- **Safety**
- Website must give confirmation prompts to users for any critical actions
- Website must create data backup within 1-2 weeks

-- **Security**
- Website must enforce role-based access controls (for buyers, sellers, and admin)
- Website must expire sessions after approx. 30 minutes
- Website must follow basic security guidelines

-- **Software Quality**
- Website must be maintainable (document components and use a clear folder structure)
- Website must be reliable (handle errors gracefully and maintain tests for key features)
- Website must be usable (provide an intuitive interface)
- Website must be testable (ensure all functions are covered by these tests and easily replicated)

1.1.2 Out-of-Scope

Out-Of-Scope defines the features, functional or non-functional requirements of the software that will NOT be tested. Here are the requirements that will not be tested:

-- **Older Web Browsers**
- Website will not be tested on antiquated web browsers (such as Mosaic and Netscape)

-- **Performance Beyond Our Limits**
- Website will not be tested for more than 50 concurrent users
- Website will not be tested for more than 10 transactions per minute

-- **Low Priority Items**
- Website will not be tested for aesthetic items (such as fonts, text sizes, and formatting)
- Website will not be tested for any other low priority features (features that are luxuries rather than necessities)

1.2 Quality Objective

The overall objective of this testing effort is to ensure that the Application Under Test (AUT) meets all functional and non-functional requirements as specified in the project scope. The testing will verify that the system operates correctly across all user roles, those rolls being buyers, sellers, and admin. The testing will also verify that the system supports all key features such as account management, transactions, and refund handling. This process aims to confirm

that the application adheres to the quality standards defined by the client, including usability, performance, security, and reliability. Additionally, the goal is to identify and resolve any bugs or issues before the system goes live, ensuring a stable and seamless user experience at launch.

1.3 Roles and Responsibilities

Name	Net ID	GitHub username	Role
Noah Scott	njs283	Njscott03	Frontend, VC Manager
Terrance Moncure	tdm594	TJMonc	Backend, Test Manager
Nicholas (Chandler) Bryant	ncb211	steameam	Frontend, Configuration Manager
Caleb Stockton	cls1268	Cable07	Backend, Document Manager

2 Test Methodology

2.1 Overview

The Agile test methodology has been adopted for this project to support a flexible, collaborative, and incremental development process. Testing is integrated throughout each sprint, allowing us to validate functionality as it's developed and provide immediate feedback to the development team. This continuous testing approach ensures that issues are identified and resolved early, reducing the risk of major defects closer to deployment. Agile testing also allows us to adapt to changing requirements and client feedback in real time, helping us deliver a more refined, user-focused product. By aligning testing with Agile principles, we aim to maintain high software quality, improve team collaboration, and ensure rapid, reliable delivery of features.

2.2 Test Levels

Testing for this project will be conducted across the following levels to ensure there are no issues with the AUT. We will perform unit testing, system testing, integration testing, and acceptance testing. Unit testing is performed by developers during development. It’s goal is to verify if individual components or functions (i.e. login logic, adding an item to the cart) are correct. System testing is conducted by the QA team on the completed application. These tests are meant to validate that the system meets the specified requirements in the scope (i.e. handling 50 concurrent users or loading aspects of the website in 3 seconds or less). System testing will include both functional and non-functional tests. Integration testing focuses on the interaction between different components and functions. These tests are comprised of integrating databases, API communitcation, and proper data flow between each of the features. Finally, acceptance testing is meant to validate that the software is fully ready for release from a user’s perspective. The includes covering real-world usage of buyers, sellers, and admins.

2.3 Bug Triage

The bug triage process plays a critical role in managing and resolving defects while testing. The goal of triage is to categorize bugs, determine their severity and priority, and decide on the appropriate resolution strategy. During triage, all reported bugs are first reviewed by the QA team, who document the steps in which they produced the bug. Each bug is then classified based on its priority (i.e. high, medium, or low). High priority bugs are those that prevent essential features from working and must be addressed immediately. This contrasts to the low priority bugs which may do something as trivial as a create cosmetic quirk and will be scheduled to be dealt with in future sprints (or saved for any time in which we can deal with it). After classification, bugs are assigned to developers to be fixed and tracked. Regular triage reviews are held during our weekly meetings to review bug status, ensure accountability, and keep the resolution schedule aligned with sprint goals. Once a bug is fixed, the QA team re-tests the fix before closing the issue.

2.4 Suspension Criteria and Resumption Requirements

There are certain conditions under which testing may need to be temporarily halted. These suspension criteria are established to avoid wasting resources when essential components are unavailable or do not work properly. Testing will be suspended if any critical components are broken or unavailable, or if the test environment becomes unstable due to server downtime. Testing will resume once the underlying issues have been addressed. These issues could be things such as the environment being stabilized, the broken modules being fixed, or any unclear requirements being clarified and implemented. Resuming any tests will also require the deployment of a new, testable build that restores minimum viable functionality for the paused test cases.

2.5 Test Completeness

The testing process will be considered complete when several key criteria are met. Firstly, 100% of planned test cases, both manual and automated, must be executed. This ensures that all in-scope features have been thoroughly validated. This includes anything that the user role may want to do, such as registration, browsing, purchasing, listing items, and handling refunds. Secondly, all high and critical priority bugs must be fixed, and medium-to-low priority bugs must either be fixed, documented, or agreed upon for resolution in a future release. Additionally, all functional and non-functional requirements defined in the scope such as role based access control, performance limits, and usability standards, must have been covered by corresponding test cases. Regression testing will also be conducted to verify that recent changes have not introduced any new issues. Finally, a formal sign-off will be obtained from the QA team and stakeholders, indicating confidence in the stability and readiness of the system for production deployment.

3 Test Deliverables

Here, we will mention all test artifacts that will be delivered during different phases of the testing lifecycle.

- **Test Plan**
  - Outlines the overall testing approach, including scope, objectives, timeline, resources, and testing methodology.

- **Bug Triage**
  - Documents that capture defects found during testing, including severity, reproduction steps, and fix status.

- **Test Completeness**
  - All test cases must be executed, all requirements must be covered, regression testing must be implemented, and formal approval must be acquired from the QA team that confirms that the system meets release criteria.

4 Resource & Environment Needs

4.1 Testing Tools

- All requirements tracking is done through Microsoft Word
- All bug tracking is done through Microsoft Word
- All the testing itself is done manually, without the use of any automated testing tools

4.2 Test Environment

The following software is required (at minimum) in addition to client-specific software. While it is possible that lower versions may also function correctly, we did not test on them and cannot guarantee compatibility.

* Windows 10 and above
* Chrome Version 135 and above (or any equivalent modern browser version and above)

5 Terms/Acronyms

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

TERM/ACRONYM	DEFINITION
API	Application Program Interface
AUT	Application Under Test
NA	Not Applicable
2FA	Two-Factor Authentication
FAQ	Frequently Asked Question
QA	Quality Assurance
VC	Version Control