

# TEST PLAN FOR ANAZOM

## *ChangeLog*

| Version | Change Date | By   | Description   |
|---------|-------------|--|---|
| 1.0     | 4/12/2023   | Yuxin Xu, Ayden Kilpatrick, Jessica Oramous, Paul Scarmado | Produced a working e-commerce website that contains the functionalities of sellers, buyers, and login |

|   |          |
|---|----------|
| <b>1 Introduction</b>                               | <b>2</b> |
| 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        |
| <b>2 Test Methodology</b>                           | <b>3</b> |
| 2.1 Overview  | 3        |
| 2.2 Test Levels                                     | 3        |
| 2.3 Bug Triage                                      | 3        |
| 2.4 Suspension Criteria and Resumption Requirements | 5        |
| 2.5 Test Completeness                               | 5        |
| <b>3 Test Deliverables</b>                          | <b>5</b> |
| <b>4 Resource &amp; Environment Needs</b>           | <b>5</b> |
| 4.1 Testing Tools                                   | 5        |
| 4.2 Test Environment                                | 6        |
| <b>5 Terms/Acronyms</b>                             | <b>6</b> |

# 1 Introduction

For this software application testing, we will be performing unit tests on the individual functionalities of our website. We will do this by utilizing CircleCI, a platform that automatically performs the unit tests for us whenever a feature has been added or changed. The test methodology that we will be adopting for the project is agile.

## 1.1 Scope

---

### 1.1.1 In-Scope

The features that will be tested include login/logout, actions related to buyers, and actions related to sellers. Some examples of actions related to buyers are adding an item to cart, searching for an item, and returning a product. Some examples of actions related to sellers are adding a new product, editing an existing product, and deleting a product.

### 1.1.2 Out-of-Scope

The requirements that will not be tested include non-functional requirements such as portability, availability, and response time, as well as any features related to the admin user type due to its functional incompleteness.

## 1.2 Quality Objective

---

The overall objective that we are trying to achieve with our testing is to ensure a properly working software that includes all of the features that are desired by the customer.

Our other objectives include:

- Identify and fix all bugs within our software before release
- Verify the logics used in our program are correct
- Guarantee an easy-to-use GUI for the customer
- Ensure the current features will work in future updates through automated testing

## 1.3 Roles and Responsibilities

---

| Name           | Net ID | GitHub username | Role                  |
|----------------|--------|-----------------|-----------------------|
| Paul Scarmardo | pbn26  | PaulScarmardo   | Configuration Manager |

|                  |        |                  |              |
|------------------|--------|------------------|--------------|
| Yuxin Xu         | yx155  | yx155            | Developer    |
| Jessica Oramous  | jlo229 | jessoramous      | QA Analyst   |
| Ayden Kilpatrick | apk145 | Ayden-Kilpatrick | Test Manager |

## 2 Test Methodology

### 2.1 Overview

---

We are adopting the agile test methodology for this project for multiple reasons.

First, agile allows us to be flexible with our features. While we are given a minimum set of requirements to build our project out of in the beginning, our ideas of those features and any additional features will shift as expectations from the client changes.

Second, agile allows us to revisit previous steps in the software development process. Because expectations from the client may change, modifications of design diagrams may be needed. Agile is a test methodology that will allow us to make any modifications that we may need to our previously designed diagrams.

Third, because agile allows us to be flexible with the software development process, we are able to test our software as we are building it. Instead of following a strict pattern that requires us to finish the coding process before beginning the testing process, agile allows us to ensure the functionality of each individual feature as we create the application.

### 2.2 Test Levels

---

For this project, we will only be performing unit tests on our application due to time constraints.

### 2.3 Bug Triage

---

| Bug Name            | Bug Description   | Date Found | Date Fixed | Resolution Description   |
|---------------------|---|------------|------------|--|
| Excess Seller Perms | Sellers are able to gain buyers' abilities by clicking the 'store' link on their navbar | 4/4/2023   | 4/7/2023   | Removed the ability for sellers to access the link on their navbar |

|                                |   |           |           |   |
|--------------------------------|---|-----------|-----------|---|
| Image Error                    | Unable to load store when an image is not found for a product   | 4/5/2023  | 4/8/2023  | The seller is now required to upload an image for each product being registered                               |
| Payment Not Received           | Sellers do not properly receive payments when a product has been bought   | 4/6/2023  | 4/9/2023  | Added a column in the products database that indicates each product's seller                                  |
| Edit Product Redirect          | Instead of redirecting to the 'edit product' page when choosing to edit a product, it redirects to the 'add product' page | 4/7/2023  | 4/10/2023 | Removed an excess <code>&lt;/form&gt;</code> tag that caused the corresponding view to not be linked properly |
| Browsing Failure               | Unable to browse through categories (it would log the user out when clicking a category)                                  | 4/7/2023  | 4/10/2023 | Changed the redirect when clicking on a category from 'homepage' to 'store'                                   |
| Name Containing Numbers        | Users are able to sign up with numbers as their names   | 4/11/2023 | 4/12/2023 | Added a new error checking condition that makes sure names contain only characters                            |
| Back Arrow After Logout        | Users are able to click the back arrow to go back to the cached page after logging out                                    | 4/11/2023 | 4/12/2023 | Removed caching ability and directs the user to login page if back arrow is clicked after logging out         |
| Inventory Quantity Unavailable | Contents on sale appears to be infinite and no stock quantity can be seen   | 4/11/2023 | 4/12/2023 | Added a new field in the products database that accounts for the amount in stock                              |

## 2.4 Suspension Criteria and Resumption Requirements

---

Unit testing for other features will be suspended if an assumption can be made that the bugs are all highly related to the flaw of one specific feature. This decision prevents further complications that could arise from altering a fully functioning feature for the sake of a flawed correlated feature.

Resumption for unit testing for all features will occur once a conclusion can be drawn that either 1) the assumption that the bug was the fault of a certain feature is wrong or 2) the flaw related to the feature is believed to be fixed.

## 2.5 Test Completeness

---

Test completeness will be achieved once the following criterias have been met:

- All features are tested
- All Manual & Automated Test cases executed
- All open bugs are fixed or will be fixed in next release
- At least 5 test cases are written and passed for each feature
- Acceptance testing for each user story is successful

# 3 Test Deliverables

The Test Artifacts that will be delivered during different phases of the testing lifecycle are:

- 
- Test Plan document
  - Test case documentation on GitHub repo under wiki
  - Test case codes for each feature
  - Test case results generated by automated testing tool
  - Acceptance testing for every user story
- 

# 4 Resource & Environment Needs

## 4.1 Testing Tools

---

Tools required to test the project:

- CircleCI - automatic unit testing tool linked to GitHub
- GitHub - home of all source codes

## 4.2 Test Environment

---

The minimum hardware requirement needed to test the application is a computer of any kind. This can be a laptop or desktop.

Following software is required in addition to client-specific software.

- Windows or MacOS operating system
- Python 3.8
- A web browsing application such as Google Chrome

## 5 Terms/Acronyms

| TERM/ACRONYM | DEFINITION   |
|--------------|--|
| API          | Application Program Interface                              |
| AUT          | Application Under Test                                     |
| GUI          | Graphical User Interface                                   |
| CircleCI     | A continuous integration tool used to perform unit testing |
| GitHub       | An open source website used by software developers         |
| Python       | The coding language used to create this web application    |