Test Plan

VENDOR Online MARKETPLACE

Robert Enuta

2021

Contents

[1.Context 2](#_Toc89439048)

[1. Introduction 2](#_Toc89439049)

[2. Objectives 2](#_Toc89439050)

[3. Scope 2](#_Toc89439051)

[2.Testing strategy 2](#_Toc89439052)

[1. Overview 2](#_Toc89439053)

[2. Testing 3](#_Toc89439054)

[1. Unit Tests 3](#_Toc89439055)

[2. Integration Tests 3](#_Toc89439056)

[3. User Acceptance Tests 3](#_Toc89439057)

[3. Schedule 3](#_Toc89439058)

[3. Hardware Requirements 4](#_Toc89439059)

[4. User Acceptance 4](#_Toc89439060)

# 1.Context

## 1. Introduction

Vendor is an online marketplace where users can sell and buy items. The main purpose of the app is making buying and selling as easy and as safe as possible. This is achieved by allowing the user to rate the experiences they had with others and making it obvious when something shady is happening. Users will be dissuaded to interact with malicious sellers and if an account keeps getting reported it will be banned from buying or selling anymore.

After selecting an item to buy, the user will enter a conversation with the seller and discuss the details. And after settling on a price and the delivery, the seller is supposed to take care of the rest themselves.

## 2. Objectives

Creating a well-rounded and rich application, while using agile, means continuous changes and updates are made. Changes that can lead to oversights in code or even bugs, for this reason extensively testing the product is important.

Testing will be done in multiple ways to ensure coverage of all possible problems.

## 3. Scope

While the project is in development, both back-end and front-end need to be tested before deployment, but also the connectivity between the front-end and back-end and between back-end and database.

# 2.Testing strategy

## 1. Overview

For testing of the methods and objects found inside the java application, unit tests will be implemented alongside new code, but this will be also tested in GitLab, by the CI/CD pipelines that have been set up, and a visualisation of the quality of the code will be available in SonarQube.

The React functionality will be tested using Cypress.

The test will mostly focus on testing the algorithms in the backend and the logic used to perform tasks. Unit test will cover all the code that focuses on changing the data received.   
The testing done in the front-end will focus mostly on making sure the html elements work and display properly.

Testing will be done on a high level, focused on each individual element rather than only on the totality of the system. Having a large number of detailed tests will make discovering and fixing issues more efficient.

## 2. Testing

### 1. Unit Tests

The unit tests will be written in parallel with the java code. Every class will have its own unit tests, because of this finding errors and fixing them will be much faster.

The testing will happen in IntelliJ Idea, every time the updates are pushed to the git repository.

### 2. Integration Tests

Integration tests are ran automatically bit GitLab every time a new version is uploaded to the repository. These tests will build the project, run the unit tests and , using SonarQube, will reveal what flaws the code contains.

### 3. User Acceptance Tests

User acceptance tests confirm that the project is ready to be rolled out and used. During acceptance tests the users compare the final product with the initial requirements.

Since this set of tests requires users to use the system, a random group of people will be selected for this test. A test script will be created to guide users through the tasks they need to perform. The tests will be recorded, and the data will be used for further improvements.

## 3. Schedule

Unit tests and CI/CD tests take place whenever a new version of the app is pushed to the GitLab repository, these tests don’t take much time to conduct and are vital to keeping the code in check, for these reasons they will be conducted as often as possible.

User acceptance tests will take place after a version stable enough for users becomes available.

# 3. Hardware Requirements

Different tests will require different equipment for through completion.

Continuous integration tests are temporary ran through the development computer, but they will later be set up to run through a docker.

User acceptance tests will be done on as many devices as possible to ensure the quality of the app, these devices should include: a smartphone utilizing Android, a computer running on Window 10 and any other versions released after Windows 7, an IOS device.

# 4. User Acceptance

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ID | Scenario | Acceptance criteria | Requirements | Testing plan |
| 1 | As a buyer, I want to be able to browse all items so that I can find exactly what I’m looking for. | I can browse a list of items. | User is connected to the website. | -Tester opens the main page and a list of items shows up. |
| 2 | As a user, I want to create an account so that I can easily sell and buy items. | I can create an account. | User has an email that can be used for registration. | -Tester opens the registration page.  -Input valid information. |
| 3 | As a buyer, I want to have a search function so that I can easily search for the items I want. | I can search for items. | There are items listed that match the search. | -Tester opens the website and goes to the main page.  -Tester clicks the search bar and types in the product they are searching for. |
| 5 | As a seller, I want to list my item so that I can sell it to other users. | I can create a listing for my item. | User has a valid account. | -Tester opens the product creation page.  -Tester inputs valid information about the product and confirms the listing. |
| 7 | As a buyer, I want to respond to item listings so that I can buy them. | I can buy an item that is being sold. | User has a valid account.  There are items to be bought. | -Tester opens a product page.  -Tester saves product to their basket. |
| 10 | As a buyer, I want to check the account of other users, so that I can verify their legitimacy. | I can check other user accounts and see their activity. | User is logged in. | -Tester logs in.  -Tester clicks on a product.  -Tester clicks on a user account. |
| 11 | As a seller, I want to have a sell history so I can easily check the items I’ve sold in the past. | I can check my sell history. | User has sold or listed items before.  User is logged in. | -Tester visits their profile.  -Tester checks their item history. |
| 12 | As a buyer, I want to have a buy history so that I can easily verify purchases I’ve made in the past. | I can check my item purchase history. | User has purchased items.  User is logged in. | -Tester visits their profile.  -Tester checks their item history. |
| 13 | As a user, I want to have a homepage where I can receive item suggestions. | I can view items suggested for me on the home page. | User is logged in.  User has searched or looked at items before. | -Tester opens the main page and a list of items shows up. |