Test Plan

<Mini Project/Alta Shop>

Contents

[1. Introduction 3](#_Toc134726023)

[1.1 Objective 3](#_Toc134726024)

[1.2 Purpose 3](#_Toc134726025)

[1.3 Background 3](#_Toc134726026)

[2. Requirement 5](#_Toc134726027)

[2.1 Tools 5](#_Toc134726028)

[2.2 System 5](#_Toc134726029)

[2.3 Testing Method 5](#_Toc134726030)

[1. Api 5](#_Toc134726031)

[2. Web UI 6](#_Toc134726032)

[3. Mobile 6](#_Toc134726033)

# Introduction

## Objective

This Test Plan document explains how the software created can run according to a predetermined plan. Testing is not only carried out on source code, but testing is also carried out on databases, components, interfaces, security, business models, and performance of the software being built.

The test plan is the basis used to test the suitability of the software design results with the objectives set during project planning. With a well-detailed test plan, all aspects of the software in general can be covered and the design results are in accordance with the development plans that have been made before.

Alta Shop is an e-commerce platform developed by Altera Academy which includes API, website and mobile. Testing eCommerce applications is important to ensure that the application can function properly and can provide a satisfying shopping experience for users.

In the process of making the Alta Shop mini project test plan, it is hoped that it can fulfill the Altera Academy mini project task assessment, to measure my understanding of making mini projects, and to be able to carry out tests to ensure product quality is working properly and as desired.

## Purpose

The purpose of testing the Alta Shop application is as follows:

1. Ensuring that the Alta Shop application in the mini project runs according to standards.
2. Ensure feasibility of the Alta Shop application.
3. Find bugs or errors

## Background

The limitations on the tests to be carried out are as follows:

1. Testing was carried out on the Alta Shop api with Rest Assured, the Alta Shop Website with Web UI and the Alta Shop Mobile application with Appium.
2. The testing performed is unit testing.
3. Testing is done manually and automation testing.
4. Fire and Website testing is done using BDD serenity.
5. Mobile testing is only done manually

# Requirement

## Tools

Tools used:

1. Intellij IDEA (Serenity for Web UI, API and Mobile Testing Automation)
2. Android Studio for Android Virtual Devices in mobile testing
3. Postman for API testing automation
4. Google Chrome for help run web testing
5. Github for testing documentation and test repots
6. Google Spreadsheet and TestRail for test documentation

## System

|  |  |  |  |
| --- | --- | --- | --- |
|  | Tools | Vendor | Version |
| Test Case Document | Spreadsheet | Microsoft | 2021 |
| Test Plan and Report | Google Document | Microsoft | 2021 |
| Web UI and API Testing | Serenity Web UI and Serenity Rest Assured | Intellij IDEA | 2022.3.2 |
| Mobile Test | Appium | Appium | 1.22.3.4 |
| Manual Api Testing | Postman | Postman | 8.6.1 |

## Testing Method

### Api

1. Manual testing

|  |  |
| --- | --- |
| Test method | Unit testing: register, login, product, categories and orders |
| Objective | Ensuring each endpoint can send requests and produce the expected response |
| Test criteria | Response data verification  Ensure the result is the same as expected |

1. Automation testing

|  |  |
| --- | --- |
| Test Method | Unit testing: register, login, product, categories and orders |
| Objective | Ensuring each endpoint can send requests and produce the expected response |
| technique | 1. Create test cases 2. Set endpoints 3. Set method 4. Create a request body 5. Run tests 6. Make Report |
| Test criteria | 1. Endpoints can function properly 2. Ensuring the expected results match the actual ones |

### Web UI

1. Manual testing

|  |  |
| --- | --- |
| Test Method | Unit testing: register, login, product, categories and transactions, |
| Objective | Identify website functionality manually by checking whether the website is working properly or there are still bugs. |
| technique | 1. Create test scenarios 2. Create test cases 3. Check the functionality of each button 4. Check the functionality of each feature |
| Test criteria | 1. Ensuring every feature can run properly. 2. Ensuring the expected results match the actual results. |

1. Automation testing

|  |  |
| --- | --- |
| Test Method | Unit testing: register, login, product, categories and transactions, |
| Objective | Identifying website functionality automatically |
| Technique | 1. Create test cases 2. Calling the elements needed for the testing process 3. Running test cases 4. Analyze test results |
| Test criteria | 1. Ensuring every feature can run properly. 2. Ensuring the expected results match the actual results. |

### Mobile

1. Manual testing

|  |  |
| --- | --- |
| Test Method | Unit testing: register, login |
| Objective | Manually identify mobile functionality and ensure product is free from bugs |
| Technique | 1. Create test scenarios 2. Create test cases 3. Checking the functionality of each button 4. Check the functionality of each feature |
| Test criteria | 1. Ensuring every feature can run properly 2. Ensuring the expected results of each feature match the actual results. |

1. Project Milestones

|  |  |  |  |
| --- | --- | --- | --- |
| No | Milestone task | Start date | End date |
| 1. | Plan test | 18/04/2023 | 10/05/2023 |
| 2. | Design test | 22/04/2023 | 09/05/2023 |
| 3. | Implement test | 28/04/2023 | 11/05/2023 |
|  |  |  |  |