CS691 – Computer Science, Fall 2021

Pace University



SYSTEM TEST PLAN

TROX - Ecommerce Web Application

**Table of Contents**

[INTRODUCTION 3](#_heading=h.30j0zll)

[TESTING SCOPE 3](#_heading=h.3znysh7)

[TESTING OBJECTIVES 3](#_heading=h.tyjcwt)

[Features to be Tested 3](#_heading=h.1t3h5sf)

[Features not to be Tested 4](#_heading=h.2s8eyo1)

[TEST PROCESS DEFINITION 4](#_heading=h.17dp8vu)

[Test Process Phases and Tasks 4](#_heading=h.26in1rg)

[Deliverables 5](#_heading=h.35nkun2)

[APPROACH TO SYSTEM TESTING 5](#_heading=h.1ksv4uv)

[Approach to Functional Testing 5](#_heading=h.2jxsxqh)

[ENTRY/EXIT CRITERIA 6](#_heading=h.3j2qqm3)

[Entry Criteria 6](#_heading=h.4i7ojhp)

[Exit Criteria 6](#_heading=h.2xcytpi)

[ENVIRONMENTAL NEEDS 6](#_heading=h.3whwml4)

[ROLES AND RESPONSIBILITIES 7](#_heading=h.2bn6wsx)

[TEST CYCLES AND SCHEDULE 8](#_heading=h.3as4poj)

[RISKS AND CONTINGENCIES 8](#_heading=h.49x2ik5)

# INTRODUCTION

The primary purpose of this System Test Plan document is to provide a generalized overview and understanding of the “TROX” project to key stakeholders. The scope, objectives, and approach to performing the system testing will be laid out in the document, as well as, explanations of the features to be tested, testing criteria, testing entry/exit criteria, and testing schedule. Any risks or contingencies will be noted at the end of the document.

# TESTING SCOPE

There are two perspectives to be outlined in the testing scope – the functional scope and technical scope.

The functional scope will include the following modules of the “Trox” system:

* Account Authentication
* Account Authorization
* Account Summary
* Browse Products

The technical scope includes the following architectural components:

* Web browser
* Application server
* Database server

# TESTING OBJECTIVES

The purpose of the System Test Plan is to focus on the functional testing of the system and evaluate its implementation stability, specifically with the account features. The technical scope requires additional quality assurance testing, which can ensure it is of the highest quality for anyone using “Trox”.

The following sources will allow for successful evaluation during testing phases, as well as, ensuring the system has the proper and necessary functionality. These sources have been laid out to build a strong system, and will help to make sure the testing is done properly.

* Business Requirements Document (BRD)
* User Stories (functional requirements)
* Requirements Composition Table (supplementary requirements)

## Features to be Tested

This section below outlines the core features of “Trox” that will be tested. They are grouped by the application modules.

User’s Account - Account Authentication

* I Have Potential User Data
  + To test whether the user is successfully able to create a new account
* Type of User
  + To test whether a user can successfully have either a buyer account, seller account, or both
* Display User Account
  + Test if input data shows up correctly

User’s Account - Account Authorization

* I have Existing User Data
  + To test whether an existing user is able to access their account with their existing credentials
* I have a non-existent account
  + To test if the system rejects an inactive/non-existent account

User’s Account - Account Summary

* I Have An Existing Account
  + To test whether all of my information is accurate and visible on my account
* Change Account
  + To test if the system accurately updates and holds my updated account information

Browse Products

* I Have Active Products
  + Test whether a user can search for a specific product
* I Have Active Categories
  + Test whether a user can search through product categories
* I Have Active Seller Account
  + Test whether a user can buy product(s) from a specific seller

## Features not to be Tested

There are various tools required for different pieces of the testing of Trox. For that reason, we will not be testing customer service (which is important for the core features), but cannot be done without the proper testing currently. We will also not be testing payment functionalities.

# TEST PROCESS DEFINITION

## Test Process Phases and Tasks

Test process lifecycle includes five conventional phases and each phase has their own deliverables and tasks.

1. Test Planning: Test planning takes the insights found during requirements or product analysis and turns them into a documented QA strategy. The test plan spells out several details of the QA work to be done, including the scope, objectives, types of functional and non-functional tests and the details for the test environments.
2. Test Design: With the test plan in place, testers can begin to write and create detailed test cases. Test cases should be simple and well understood for any member of the team but also unique from other test cases. Test cases should aim to achieve full coverage of the requirements in the specification document. Its important that test cases be identifiable and repeatable, as developers will add new functionality to the product over time.
3. Test preparation: The test environment provides the setting where the actual testing occurs. When ready, testers establish the parameters for the test environment, which include the hardware, software, test data, frameworks and configurations and network. Testers adjust these environment parameters depending of what the test case requires.
4. Test Execution: Testers executes all of the test cases or as many as is possible within allotted time. Testers will identify and report detailed bugs that arise from test case execution and log the system’s performance compared to the requirements. As developers makes fixes, testers often retest the product to make sure new defects don’t materialize.
5. Test Reporting: Testing team provides a test closure report, which summarizes and communicates its findings with the rest of the team. The summary includes the testing work and results, an assessment of the testing and the manager’s approval. During the test cycle closure, the testing team checks its deliverables, which include details relevant to the testing work, like the test strategy, test case documents, test scripts and test results. The team will complete and close incidents reports, which details unusual or unexpected behavior that test team observe during testing.

## Deliverables

On this project, the test process deliverables include:

* Test Planning
  + Test strategy document
  + Test plan document
* Test Design
  + Test design document
  + Test case specifications
* Test Preparation
  + Test environment readiness
  + Loaded test data
  + Refined test schedule and confirmed testing resources
* Test Execution
  + Test execution status
  + Defect reports
  + Test execution progress report
* Test Reporting
  + Test summary report
  + Testing metrics

# APPROACH TO SYSTEM TESTING

## Approach to Functional Testing

The overall approach to functional testing will be based on the Black-box method:

* Test cases will be designed using some formal black-box techniques such as boundary-value analysis, equivalent-class partitioning, cause-effect graphing, decision tables, and state-transition testing, where applicable.
* Test execution will be conducted manually, from the user perspective and based on formal test case specifications.

The test execution results will be captured and reported in test execution logs.

# ENTRY/EXIT CRITERIA

This section defines both Entry and Exit Criteria for test execution and is intended to establish a common understanding about the conditions when the test execution can start and when it can stop.

## Entry Criteria

The test Entry Criteria include the following items:

* The application build is produced and deployed to the test environment
* The system test plan is produced and approved
* The test environment is ready for testing
* Test Designs and test case specifications are completed

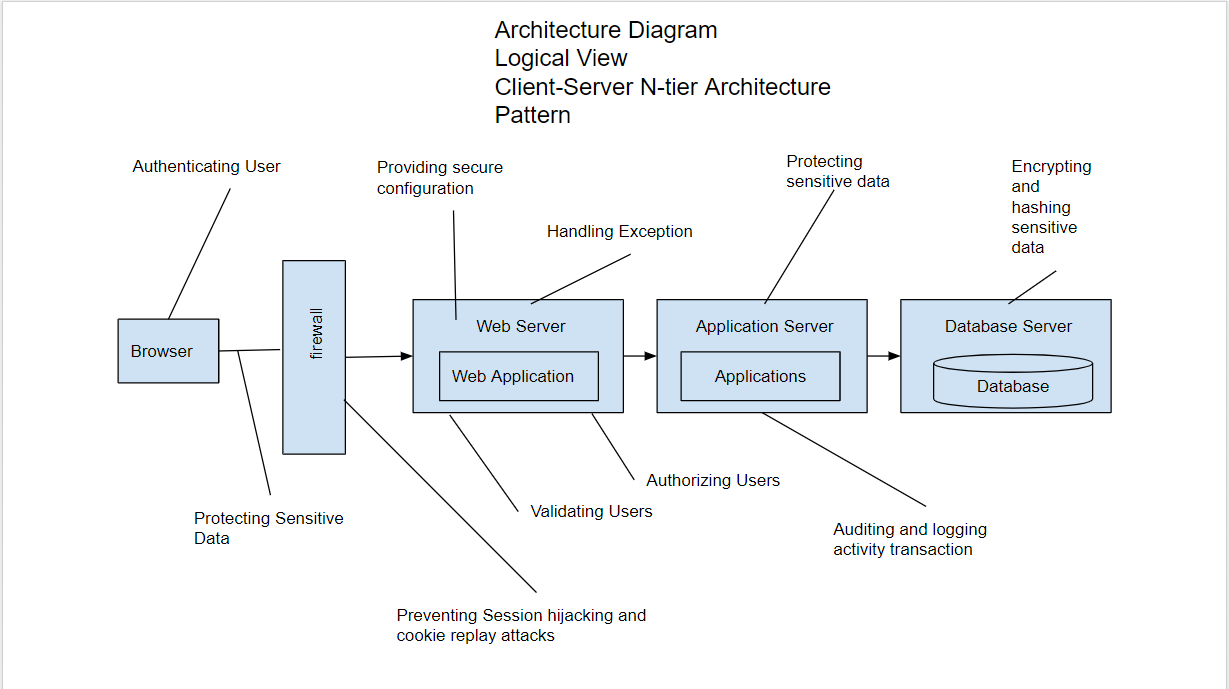
## Exit Criteria

The test Exit Criteria include the following items:

* All test cases have been executed
* Zero defects of Critical and High-severity remain open
* Open defects of Medium and Low severity have known work-arounds
* Test Summary report is produced and published

# ENVIRONMENTAL NEEDS

The Test Environment should be available to start test execution. It includes a laptop with virtual machine running the web server and database, and internet browsers (Chrome, Firefox, Internet Explorer and Safari) to access the application. The architecture of the test environment is shown below.



# ROLES AND RESPONSIBILITIES

The project team has seven members that are assigned various project roles including Project Manager, Product Owner, Lead Business Analyst, Lead Developer, DBA, Lead QA Analyst. Their responsibilities are defined in the table below.

| **Project Role** | **Role Responsibilities** |
| --- | --- |
| Project Manager | Responsible for the overall project timelines, review  and approval of the System Test Plan, escalation of  issues. |
| Lead QA Analyst | Designing a test plan, establishing a test repository, developing test case specifications, executing testing and reporting defects. |
| Product Owner | Contributing to the test plan and test case specifications. Reviewing test results. |
| Lead Business Analyst | Contributing to the test plan and test case specifications. Reviewing test results. |
| Lead Developer/Developer | Responsible for producing a working software build,  communicating release notes, investigating and  fixing software defects. |
| DBA | Assisting the Lead Developer in establishing and maintaining the test environment. Entering the required data in the databases. |

# TEST CYCLES AND SCHEDULE

The system test execution will be conducted as three test cycles that are aligned with three application modules as follows:

Cycle 1. User Experience I

* This cycle concentrates on testing the first part. This cycle concentrates on testing User Register/Login Module

Cycle 2. User Experience II

* This cycle concentrates on testing the second part (homepage) of the User Experience Module

Cycle 3. User Register/Login

* This cycle concentrates on testing the third part (clicking on products/images/logo) of the User Experience Module..

See the schedule of the test execution cycles in the project plan.

# RISKS AND CONTINGENCIES

This section highlights a few potential risks and contingencies that may have happened during the system testing.

* Limited testing resources may result in a delay.
* Any changes on the scope objectives can cause a delay or extra work.
* A large number of defects require a longer time to fix defects and complete testing.
* Lack of collaboration of the team members can have a negative impact on the testing progress.