

Technological Institute of Costa Rica

Computer Engineering School

Test Model Specification

Software Quality Assurance

I-2024

Professor: Erick Hernández Bonilla

Members:

María Paula Bolaños Apú

Marco Herrera González

Jeffrey Leiva Cascante

Content

Overview	4
Introduction.....	4
Scope	4
References	4
Internal	4
Test Model Specification – Test Model 1	5
Test Model Unique Identifier.....	5
Objective	5
Priority.....	5
Type of Testing Required (from Test Strategy).....	5
Test Model	5
Traceability	5
Test Model Specification – Test Model 2	6
Test Model Unique Identifier.....	6
Objective	6
Priority.....	6
Type of testing required (from Test Strategy).....	6
Test Model	6
Traceability	7
Test Model Specification – Test Model 3	8
Test Model Unique Identifier.....	8
Objective	8
Priority.....	8
Type of testing required (from Test Strategy).....	8
Test Model	8
Traceability	8
Test Model Specification – Test Model 4	9
Test Model Unique Identifier.....	9
Objective	9
Priority.....	9
Type of testing required (from Test Strategy).....	9
Test Model	9

Traceability	9
--------------------	---

Revision Log

Date	Version	Changes
April 9, 2024.	1.0.	Initial draft released for review.
April 12, 2024.	2.0.	Includes corrections and additions specified by the Quality Assurance Team members after the review of the document.

Overview

Introduction

This document details the test model specification for project #1 of the Software Design course, whose objective is the development of a web system for image publications and sales of beauty and skincare products. The objective of the test model specification is to provide a representation of the aspects of the test items, focused on the attributes required in the test coverage.

Scope

This document covers the test model specification for the static testing of the project. This document is based on the project test plan document. The static testing will be applied to all source code of the project.

References

Internal

1. Project Test Plan V 1.0.
2. Source Code of Project.

Test Model Specification – Test Model 1

Test Model Unique Identifier

Static Test Model #1

Objective

Verify that all functions in the code are mapped to a functional requirement, so that the source code contains only the necessary code to implement the requirements, and there are no additional features implemented.

Priority

1

Type of Testing Required (from Test Strategy)

According to the project test plan document, static testing is required for this test. This test will be manually executed, as there is no automatic tool to map the source code to the requirements or that can detect additional features in the code. For the previous reason, the source code will be manually reviewed in via an inspection of the source code of the project.

Test Model

This test will be carried out through a review of the source code of the project, to determine jointly if the objective was achieved. The test will be carried out following the next steps:

1. A document with all the functional requirements of the project and the functions of the code will be sent to the Development Team before the meeting review.
2. The Development Team will check the source code of the project to determine if there is code that is not mapped to a requirement of the project. This will be done before the meeting.
3. During the meeting, the team will discuss their findings and define if the objective was meet. The results of the meeting and final decision will be documented.
4. The report document will be made after the meeting by a member of the Development Team, and it will be sent to the Quality Assurance Team.
5. The report must specify all parts of the code (indicating the source files, function names, and code line numbers) where additional code or features were found.
6. To consider this test as passed, all functional requirements specified in the software requirements document have to be implemented in the source code. If there are additional features, the test is still partially passed until these requirements are removed. If there are requirements missing, then the test is no passed.

Traceability

The file with the requirements and the source code of the project can be found in the online repository of the project.

Test Model Specification – Test Model 2

Test Model Unique Identifier

Static Test Model #2

Objective

Evaluate the time complexity of the source code to identify potential areas that may affect performance.

Priority

1

Type of Testing Required (from Test Strategy)

According to the project test plan document, static testing is required for this test. This test will be manually executed, as there is no automatic tool to analyze the time complexity of the code without actually executing the code, and static testing does not include code execution. For the previous reason, the source code will be manually reviewed in via an inspection of the source code of the project.

Test Model

This test will be carried out through a review of the source code of the project, to determine jointly if the objective was achieved. The test will be carried out following the next steps:

1. The Development Team will check the source code of the project and analyze the time complexity of critical functions in the code. Such functions include database queries and functions with complex logic and calculations. Functions used to redirect calls can be ignored. The Development Team is required to identify parts of the code that can be refactored for better time performance, identify inefficient database queries, and identify potential bottlenecks in the code.
2. During the review meeting, the team will discuss their findings and define if the objective was meet. The results of the meeting and final decision will be documented.
3. The report document will be made after the meeting by a member of the Development Team, and it will be sent to the Quality Assurance Team.
4. The report must specify all parts of the code (indicating the source files, functions, queries, and code line numbers) that comprise performance or that can be refactored for better time complexity.
5. To pass this test, no critical or catastrophic anomalies have to be presented. If the are marginal or negligible anomalies, then the test is partially approved until such anomalies are corrected. If there are critical or catastrophic anomalies, then the test is no passed.

Traceability

This test applies to all source code of the project, which can be found in the online repository of the project.

Test Model Specification – Test Model 3

Test Model Unique Identifier

Static Test Model #3

Objective

Verify that the code complies with established coding standards to ensure maintainability.

Priority

2

Type of Testing Required (from Test Strategy)

According to the project test plan document, static testing is required for this test. This test will be automatically executed, using the ESLint framework for such purpose.

Test Model

A member of the Development Team is required to use the ESLint framework to ensure that the source code follows coding standards. The automatic inspections will be executed in the environment defined in the test plan. The coding standards will be provided by the Quality Assurance Team, specifically coding standards for Typescript and React. After the source code is automatically reviewed, a report document will be prepared by the Development Team member specifying the results of the review. The report must include the places (indicating the source file and code line numbers) in the code that did not follow the provided coding standards. After the report is completed, it must be sent to the Quality Assurance Team.

To consider this test as passed, there should be no more than 50 non-compliances to coding standards in the entire source code. Between 51 and 100 non-compliances means the test is partially approved. More than 100 non-compliances mean the test is not approved.

Traceability

This test applies to all source code of the project, which can be found in the online repository of the project.

Test Model Specification – Test Model 4

Test Model Unique Identifier

Static Test Model #4

Objective

Identify and eliminate duplicate code to improve maintainability, reduce the risk of errors, and prevent the use of obsolete code.

Priority

3

Type of Testing Required (from Test Strategy)

According to the project test plan document, static testing is required for this test. This test will be automatically executed, using the JSCPD framework for such purpose.

Test Model

A member of the Development Team is required to use the JSCPD framework to detect all duplicated code in the source code of the project. The automatic inspections will be executed in the environment defined in the test plan. After the source code is automatically reviewed, a report document will be prepared by the Development Team member specifying the results of the review. The report must include the places (indicating the source file and code line numbers) in the code where duplicated code was found. After the report is completed, it must be sent to the Quality Assurance Team.

To consider this test as passed, there should be no more than 15.00% of duplicated code in the frontend and backend of the system. More than 15.00% means the test is not approved.

Traceability

This test applies to all source code of the project, which can be found in the online repository of the project.