**VIETNAM NATIONAL UNIVERSITY HO CHI MINH CITY**

**UNIVERSITY OF INFORMATION TECHNOLOGY**

**FACULTY OF SOFTWARE ENGINEERING**

**------------------------------**

****

<PROJECT NAME>

Test Plan Document

Project Code:

Document Code:

**TP. HỒ CHÍ MINH - 2023**

TABLE OF CONTENTS

1 Introduction 3

2 Business Background 3

3 Test Objectives 3

4 Scope 3

5 Test types Identified 3

6 Problems Perceived 3

7 Architecture 3

8 Environment 3

9 Assumptions 3

10 Functionality 3

11 Security 4

12 Performance 4

13 Usability 5

14 Test Team Organization 6

15 Schedule 6

16 Defects Classification Mechanism 6

17 Configuration Management 6

18 Release Criteria 6

Record of change

\*A - Added M - Modified D - Deleted

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Effective Date | Changed Items | A\* M, D | Change Description | New Version |
| 16-Apr-2023 |  | A | New document | 1.0 |
| 27-May-2023 | 2.4 Brush Cutter Details Page | M | Modify procedure | 1.1 |
|  | 2.6 Tiller Detail Page | M | Modify procedure | 1.1 |
|  | 2.8 Mower Detail page | M | Modify procedure | 1.1 |
|  | 2.9 Mower wizard | M | Modify procedure | 1.1 |
|  | 2.10 Garden wizard | M | Modify procedure | 1.1 |
| 18-Jun-2023 | 2.23 Buy Transaction | A | Add a new test case | 1.2 |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

SIGNATURE PAGE

**ORIGINATOR:** Trinh Dong Nguyen 18-Jun-2023

Tester

**REVIEWERS:** Trinh Dong Nguyen 19-Jun-2023

Test Leader

**APPROVAL:** Trinh Dong Nguyen 19-Jun-2023

Project Leader

# Introduction

1.1 Purpose

This document encompasses a comprehensive description of all test scenarios executed within the context of the project. It provides in-depth scenarios for all elements slated for testing, as outlined in the test plan (please consult the document labeled ... for additional details). All user interfaces will undergo testing in accordance with the User Interface Test Case Specification, while other components will be evaluated using the respective test case specifications provided below

NOTE: Due to the incompletion of the detailed design, some parts of pages are not clarified such as static textual contents, links, and images. Thus most of the test cases specified in this document are not completed. Only functional test cases are completed.

1.2 General information

<Provide a concise overview of the document's intent and structure, including the number of sections. It comprises a brief description of the content and focus of each section>

# Business Background

The business background provides the reader with information regarding:

1. The type of business (e.g., wholesale, retail, manufacturing, service, etc.).
2. The type of legal entity (e.g., corporation, LLC, partnership, sole proprietorship, etc.).
3. When the business was established.
4. Where the business is located.

# Test Objectives

The purpose of this part describes what are the objectives of this document including some contents as follows:

1. Prevent defects: Efecient testing helps preventing defects and that helps in providing an error-free application.
2. Evaluate the work products: The work products such as the Requirement document, Design, and User Stories should be verified before the developer picks them up for development. Identifying any ambiguity or contradicting requirements at this stage saves considerable development and test time. The static analysis of the code (reviews, walk-thru, inspection, etc.) happens before the code integrates/is ready for testing. This idea of testing is known as Verification. It is the process of evaluating the product during the development phase of each work product.
3. Verify the fulfillment of all specified requirements: This objective reveals the fact that the essential elements of testing should be to fulfill the customer’s needs. Testers test the product and ensure the implementation of all the specified requirements. Developing all the test cases, regardless of the testing technique ensures verification of the functionality for every executed test case. The Tester should also create a requirement traceability matrix (RTM), which will ensure the mapping of all the test cases to requirements. RTM is an effective way to ensure that test cases have got the right requirement coverage.
4. Validate if the test object is complete and works as per the expectation of the users and the stakeholders: Testing ensures the implementation of requirements along with the assurance that they work as per the expectation of users. This idea of testing is called Validation. It is the process of checking the product after development. Validation can be manual or automation. It usually employs various types of testing techniques, i.e., Black Box, White Box, etc. Generally, testers perform validation, whereas customers can also validate the product as part of User acceptance testing. Every business considers the customer as the king. Thus customer satisfaction is a predominant need for any business.
5. To prevent defects in the software product: One of the objectives of software testing is to avoid mistakes in the early stage of development. Early detection of errors significantly reduces the cost and effort. The prevention of defects involves doing a root cause analysis of the defects found previously and after that, taking specific measures to prevent the occurrence of those types of errors in the future. Efficient testing helps in providing an error-free application. If you prevent defects, it will result in reducing the overall defect count in the product, which further ensures a high-quality product to the customer.
6. To find defects in the software product: Another essential objective of software testing is to identify all defects in a product. The main motto of testing is to find maximum defects in a software product while validating whether the program is working as per the user's requirements or not. Defects should be identified as early in the test cycle as possible.
7. To provide sufficient information to stakeholders: The purpose of testing is to provide complete information to the stakeholders about technical or other restrictions, risk factors, ambiguous requirements, etc. It can be in the form of test coverage, testing reports covering details like what is missing, and what went wrong. The aim is to be transparent and make stakeholders fully understand the issues affecting quality.
8. To reduce the level of risk of insufficient software quality: The possibility of loss is also known as risk. The objective of software testing is to reduce the occurrence of the risk. Each software project is unique and contains a significant number of uncertainties from different perspectives, such as market launch time, budget, the technology chosen, implementation, or product maintenance. If we do not control these uncertainties, it will impose potential risks not only during the development phases but also during the whole life cycle of the product. So, the primary objective of software testing is to integrate the Risk management process to identify any risk as soon as possible in the development process.

# Scope

In this part, students should describe what is being tested, which is new to all the functions of a specific product, its existing interfaces, integration of all functions, etc.

**4.1 Inclusions:** Features to be tested: This section defines a list of features/modules, functional or non-functional requirements of the Application Under Test, that need to be tested.

**Example**: The following features are to be tested:

Login page

Products

Product details

Shopping cart

**4.2 Exclusions:** Features NOT to be tested: This section defines a list of features/modules, which are not to be tested. It also mentions the reason for exclusion.

Example: Payment using UPI need not be tested, as the feature is going to be disabled.

# Test types Identified

This section describes the overall approach to testing. Various techniques which will be used for testing are mentioned. It also details the methodology and lists the different testing levels and testing types that will be carried out. Many rounds of testing to be carried out are also mentioned. It also mentions the defect-tracking process and the list of tools that will be used. Students chose a tuple of kinds of tests below. Suggested that in this circumstance, all types of tests should be chosen.

**Test Levels:**

1. Unit testing: In Unit/Component testing, the objective is to find as many defects as possible. Fixing defects at the unit level saves cost, effort, and time.
2. Integration testing: The objective is to find defects in the integration touchpoints of 2 or more systems. Individual system testing is not a focus here.
3. User acceptance testing: The objective of this type of testing is to confirm the system works as expected by the end user. It is the final stage of the testing before its deployment.
4. Regression testing: The objective of Regression testing is to ensure no new defects have been introduced mainly in case of enhancements or defect fixes.
5. System Testing: This testing checks the reliability and performance of the software. The software should be tested to find out if it works as expected, even under the maximum workload.

**Test Types:**

1. API Testing: Test all the developed APIs even before they are integrated with the UI.
2. UX Testing (Usability/User Experience Testing): Test the user-friendliness of the application.
3. GUI Functional Testing: Test the features for functionality against requirements.
4. Automation Testing: Run available automation scripts.
5. Smoke Testing: Run smoke tests on each new build.

In addition to the type of testing, the objectives also vary with the SDLC model. For instance, in a waterfall model, the aim is to be as detailed as possible in the artifact creation (e.g., test cases). Whereas in the agile model, the artifacts are created just in time and with the minimum required details. created just in time and with the minimum required details.

# Problems Perceived

# Architecture

# Environment

This section defines the list of hardware and software requirements for testing. Make sure to detail specific devices and software versions. It also mentions the tools that will be used.

**Example:**

Windows Server: Staging QA

Operating System on Test Machines: Windows 10

Browser (must have): Chrome – Last available version

Network – Wi-Fi: So that laptops can be used for testing

Jira: For bug tracking

# Assumptions

# Functionality

***Constraints and Resolutions***

|  |  |  |
| --- | --- | --- |
| **Parameter** | **Customer Constraints** | **Infosys Limitations** |
| Constraint 1 |  |  |
| Constraint 2 |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

***Risk Identified & Mitigation Planned***

***Test Strategy***

***Automation Plans***

***Deliverables***

# Security

***Constraints and Resolutions***

|  |  |  |
| --- | --- | --- |
| **Parameter** | **Customer Constraints** | **Infosys Limitations** |
| Constraint 1 |  |  |
| Constraint 2 |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

***Risk Identified & Mitigation Planned***

***Test Strategy***

***Automation Plans***

***Deliverables***

# Performance

***Constraints and Resolutions***

|  |  |  |
| --- | --- | --- |
| **Parameter** | **Customer Constraints** | **Infosys Limitations** |
| Constraint 1 |  |  |
| Constraint 2 |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

***Risk Identified & Mitigation Planned***

***Test Strategy***

***Automation Plans***

***Deliverables***

# Usability

***Constraints and Resolutions***

|  |  |  |
| --- | --- | --- |
| **Parameter** | **Customer Constraints** | **Infosys Limitations** |
| Constraint 1 |  |  |
| Constraint 2 |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

***Risk Identified & Mitigation Planned***

***Test Strategy***

***Automation Plans***

***Deliverables***

***Compatibility Constraints and Resolutions***

|  |  |  |
| --- | --- | --- |
| **Parameter** | **Customer Constraints** | **Infosys Limitations** |
| Constraint 1 |  |  |
| Constraint 2 |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

***Risk Identified & Mitigation Planned***

***Test Strategy***

***Automation Plans***

***Deliverables***

# Test Team Organization

# Schedule

Test Schedule and Estimation

It includes details about the target start and end date of test execution, and a detailed schedule of all testing milestones, activities, and deadlines.

Example:

Test case design:

Start Date – <Test Case design start date>

End Date – <Test Case design end date>

Sign-in/Sign-up: 6 hours

Shopping cart: 8 hours

# Defects Classification Mechanism

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Type of Defects | Functionality | Performance | Security | Usability | Compatibility |
| Critical |  |  |  |  |  |
| Major |  |  |  |  |  |
| Minor |  |  |  |  |  |
| Cosmetics |  |  |  |  |  |

***Defects Logging and Status Changing Mechanism***

***Turn Around Time for defect fixes***

# Configuration Management

# Release Criteria

# APPENDIX 01: TEST CASE SPECIFICATION TEMPLATE

| ID | Test Case Field | Description |
| --- | --- | --- |
| 1 | Test case ID | Test case ID Each test case should have a unique ID |
| 2 | Test Priority | Test Priority is very useful while executing tests. There are types of priorities:  - Low  - Medium  - High |
| 3 | Test Designed by | Name of test case writer (tester) |
| 4 | Date of test designed | The date on which the tests were created |
| 5 | Test Executed by | Tester’s Name |
| 6 | Date of the Test Execution | Date of test execution |
| 7 | Name or Test Title | The title should provide a brief description of the test case, such as "Reset password". The title is quite important because it is often the first or only thing you see when glancing at a list of test cases. Clear titles are the key to helping testers quickly find the right test cases. |
| 8 | Description/Summary of Test | Detailed description for the test case (test case). In this section, you can also set up categories to organize test cases into logical groups. |
| 9 | Pre-condition | Any requirements that need to be completed before test case execution |
| 10 | Test Steps | Test Steps Test steps, give the tester a numbered list of steps to be performed in the system, making the test case easier to understand.  There should be 3-8 testing steps per test case. Too many steps will make it difficult for programmers and testers to reproduce the steps when a bug report is issued based on the test case. |
| 11 | Test Data | Test Data You can enter test data directly into the test data fields, or specify a separate file containing test data for 1 or more test cases. By using such a test data file, you avoid hard-coding test data in test cases, so a single test case can be used to test a set of test cases. |
| 12 | Expected Results | Expected Results Refers to the expected results including errors or messages appearing on the screen. The tester needs to know the expected result to evaluate whether this test case is successful or not. Details about the optimal level of this field vary depending on the situation. |
| 13 | Post-Condition | Post-Condition What is the state of the system after running the test case? |
| 14 | Status (Fail/Pass) | Status (Fail/Pass) Mark this field as failed if the actual result is not the same as the expected result.  Mark this field as Pass if the actual result is the same as the expected result. |
| 15 | Notes/Comments/Questions | Notes/Comments/Questions If some special conditions need to be noted related to the fields above. |
| 16 | Requirements | Requirements List of requirements for a specific testing cycle. |
| 17 | Attachments/References | Attachments/References Files and documents attached to the test case, such as screenshots and other supporting documents |
| 18 | Automation? (Yes/No) | Automation? (Yes/No) Fill in "YES" when the test cases using test automation. |