Franklin Camacho
Chicks Group Inc.

Table of Contents

Franklin Camacho
Chicks Group Inc.

Test Plan Design

This document outlines the plan for the creation, development, and execution of test cases for the Chicks Gold project as part of the QA Challenge. The main purpose of this document is to provide detailed guidance for the design, development and execution of test cases. The plan ensures that all components of the system meet the defined requirements, ensuring quality, minimizing risks and documenting a structured approach for the comprehensive evaluation of the final product.

This plan is designed to be a key resource in coordinating test activities, ensuring that all aspects of the system are properly validated and that defects are identified in a timely manner.

Scope

The scope of this test plan covers both Frontend and Backend components, ensuring that critical functionalities are thoroughly evaluated.

Frontend (ReactJS)

The frontend scope includes:

- **Pages:** Login, Dashboard, Products, and Orders.
- **Key validations:**
  - Smooth and responsive navigation.
  - Compatibility with modern browsers.
  - Proper integration and API response validation.
- **Usability tests:** Accessibility and user experience evaluation.

Backend (C# APIs)

The backend focuses on:

- **Endpoints:** Validation of responses and proper error handling.
- **CRUD Operations:** Evaluation of create, read, update, and delete operations for products and orders.
- **Security:** Testing user authentication and token management.

- **Performance:** Measuring scalability and performance under various loads.

Objectives

The main objectives of this plan include:

- **Ensure quality:** Verify that system functionalities meet the defined requirements.
- **Early defect detection:** Identify issues in early development stages to avoid additional costs in production.
- **Comprehensive testing:** Ensure full coverage, including edge cases and unforeseen scenarios.
- **Data security:** Validate that the system protects sensitive information and handles errors correctly.
- **Professional documentation:** Provide clear and detailed reports to facilitate analysis and decision-making.

Resources

Tools

- **Postman:** For manual and automated API testing.
- **Cypress:** Frontend automation testing.
- **Git:** Version control and code storage.
- **Visual Studio Code:** IDE for development and testing.

Test environments

- **Local:** Local development environment for backend testing.
- **Browsers:** Google Chrome, Microsoft Edge, and Opera.
- **Operating system:** Windows 10 Pro 22H2.

Test data

- Valid and invalid user credentials.
- Products with various configurations and attributes.
- Orders in different states and configurations.

Risks and Mitigation Strategies

Potential risks

- **Technical incompatibilities:** Version mismatches between tools and development environments.
- **Misconfigurations:** Poorly configured test environments impacting execution.
- **Insufficient coverage:** Critical scenarios potentially being overlooked.

Mitigation strategies

- Setting up replicable and well-documented test environments.
- Running tests on multiple devices and browsers.
- Prioritizing test cases for high-impact functionalities.

Deliverables

The testing process will conclude with the following deliverables:

- **Test plan design:** A detailed document outlining scope, objectives, resources, risks, and deliverables.
- **Test cases:** A set of cases with descriptions, expected results, preconditions, and detailed steps.
- **Execution reports:** Summaries of successes, failures, and detected defects.
- **Automated scripts:** Configuration files and code for automated tests.
- **Test repository:** Access to code and documentation generated during the process.

Defect Classification

Defects will be categorized based on their impact:

- **Low:** Minor issues that do not affect core functionality.
- **Medium:** Defects impacting secondary functionalities but not blocking the system.
- **High:** Failures limiting main functionalities requiring high-priority attention.
- **Critical:** Defects blocking core functionalities needing immediate resolution.

Conclusion

This plan provides a clear and structured guide for test execution, ensuring a comprehensive approach to system quality evaluation. The combination of tools, environments, and methodologies ensures accurate results that contribute to project success. Detailed documentation and continuous analysis will allow iterative improvement of the QA process, strengthening the quality and reliability of the final product.