Change control

This page shows the change status of this document. With each change a new edition takes place.

|  |  |  |  |
| --- | --- | --- | --- |
| VERSION | REVISION | AUTHOR | DATE |
| V1.0 | Draft Version | Apelo Mikaella Anne (ERNI) | August 15, 2024 |
| V1.1 | Draft Revision | Apelo Mikaella Anne (ERNI) | August 20, 2024 |
| V2.0 | Final Document and Approval | San Juan Romar (ERNI) | August 21, 2024 |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

Property rights

This document is the intellectual property of ERNI Development Center Philippines Inc., and may not be copied, reproduced, transmitted or used without the permission of ERNI Development Center Philippines Inc.

Unauthorized exploitation is punishable according to Art. 23.

The work is also copyrighted. The rights justified thereby, in the translation, the reprinting, the removal of illustrations, the passing on photomechanical or similar way and the storage in data processing plants remain, even with only partial use, reserved.  
The functions, data and illustrations described in this documentation are subject to change at any time. They serve the better understanding of the matter, without claiming completeness and accuracy in detail.

COPYRIGHT © 2020 ERNI Development Center Philippines Inc.   
All rights reserved  
Content

1 Introduction 4

1.1 Purpose 4

1.2 In Scope 4

1.3 Out Scope 5

2 Test strategy 5

3 Test process 6

3.1 When To Create Test Case 7

3.2 When To Begin Testing 7

3.3 Defect Management Process 8

3.3.1 When To Log Bugs 8

3.3.2 When To Retest Bugs 8

3.4 Defect Reporting 8

3.5 Bug Severity and Priority Definition 9

4 Test Organization 10

5 Test Environment 10

6 Test Tasks 11

7 Test Report 12

7.1 Smoke Testing Report 12

7.2 Regression Testing Report 12

7.3 Test Execution Report 12

7.4 Bug Status Report 12

7.5 User Story Refinement Report 13

7.6 Sprint Planning Report 13

7.7 Test Plan Report 13

7.8 Test Case Creation Report 13

7.9 Test Case Validation Report 13

7.10 Integration Testing Report 13

7.11 Bug Report 13

7.12 Sprint Review and Demo Report 13

7.13 Pre-Production Release Test Report 13

7.14 Production Release Test Report 13

8 Acceptance Criteria 13

8.1 Test Entry Criteria 13

8.2 Suspension Criteria 14

8.3 Resumption Criteria 14

8.4 Test Exit Criteria 14

9 Risk Management 14

# Introduction

The **JPetStore** website caters to consumers who are specifically looking to buy a pet. It is a platform that features a wide variety of animals, allowing users to find the perfect pet to suit their needs and preferences. Whether you're searching for a fish, dogs, cats, reptiles, and birds, **JPetStore** offers information and options to help you come up with a decision on what pet to buy.

## Purpose

The purpose of this document is to outline the test plan for the **JPetStore** website, ensuring it meets quality standards and functions correctly. **JPetStore** Test Plan aims to develop an effective approach for detecting bugs through the test strategies outlined in this document and to recognize potential risks along with strategies for managing them. It helps and ensures consistency in testing and serves as a documentation.

## In Scope

| Feature | Description |
| --- | --- |
| User Registration and Login | The user registration section on the **JPetStore** website is used to collect essential information for creating an account, including a user ID and password for login and personal account information for the billing address. It also allows users to set profile preferences such as language, favorite category, and to enable features like MyList and MyBanner, with a button to save all account information. The login function requires users to enter their username and password to access their account, providing an indication if the entered credentials are invalid. Additionally, it offers a "Register Now" option for new users to create an account. The login page includes a login button, and upon successful login, it redirects users to the website’s homepage. |
| Browse Product and Search | The browse product section on the **JPetStore** website offers separate categories for each classification of animals, including the name and product ID of each specific animal. Upon clicking the product ID, more detailed classifications of that particular animal appear, along with the price, an "Add to Cart" button, and a description. Navigation options are available to return to the type of animal category, ensuring a clear division of animal classes for consumers and easier navigation for each classification. The search feature allows users to input specific keywords and be redirected to the relevant page, making it particularly helpful for finding a certain breed of animal without scanning through the entire catalog. |
| Cart, Checkout and Order History | The **JPetStore** website's shopping cart page provides a comprehensive summary of all orders, including navigation options to return to animal categories, the ability to remove specific orders, and an option to update the cart. Each order's quantity is displayed, and a checkout button is available for finalizing purchases, allowing users to review and correct any mistakes, such as accidentally adding an animal. During checkout, users can view payment details and billing addresses, with navigation options to return to animal categories and better visualization for payment methods and delivery addresses. The order history section provides a summary of each order, including the order date and time, billing and shipping addresses, payment details, and the total cost of the purchase. |
| Profile Management | The profile management section on the **JPetStore** website is used to manage and update user information, account information, and profile information. It allows users to maintain their user ID and password under user information, while account information contains all personal details needed for the billing address. Profile information includes settings for language preference, favorite category, and options to enable MyList and MyBanner. This section also provides the functionality to save any changes made to the specific informations. Additionally, the profile management section features a "My Orders" option, which is used for tracking pending deliveries or orders that are yet to be delivered, ensuring users can monitor their purchase status. |

## Out Scope

| Feature |
| --- |
| - |
| - |

# Test strategy

The test strategy will be created by the ERNI PH team by the local test manager in cooperation with the technical team. The test strategy will be based on the needs of the project, the expected timeline, availability of resources, sprint plans, and development design documents. The test strategy and coverage will be adjusted accordingly depending on the state of the project, the resources available, and any changes made to the design documents. Test requirements, activities, roles and responsibilities are included in the test plan.

Functional testing will be done to ensure that **JPetStore** Website is working correctly in terms of functionality and design.

**Smoke Testing** is performed to ensure that the core functionalities of the **JPetStore** website are working properly before proceeding with more detailed and thorough testing. By identifying major issues, smoke testing helps in saving time and resources, allowing testers to focus on more complex scenarios if the core functionalities are working properly.

**Integration Testing** is conducted to ensure that different modules of the **JPetStore** website work together as a whole. This testing tests that data is correctly passed between various pages, ensuring proper interaction among them. By identifying issues that may occurs when different modules are combined, integration testing helps in determining and resolving bugs early in the development cycle, ensuring a functional application.

**User Acceptance Testing** is essential for validating that the **JPetStore** website meets the business requirements and user needs. This testing ensures that the application aligns with the end-user requirements and expectations. It provides a final check before the application is released to production, it is an assurance that the website will deliver the intended value to its users.

**Usability Testing** evaluates how easy and user-friendly the **JPetStore** website is. This testing ensures that the application is straightforward to use, identifying areas for improvement in the user interface and overall user experience. Additionally, usability testing ensures that the application is accessible to all users, thereby enhancing the overall user satisfaction and inclusivity of the website.

**System Testing** involves testing the complete integrated system of the **JPetStore** website to evaluate its compliance with the specified requirements. This testing validates the entire application, ensuring that all components work together seamlessly. System testing ensures that the application meets all functional and non-functional requirements, which can include performance testing, security testing, and other non-functional aspects, thereby providing a comprehensive approach of the system's readiness for deployment.

**Black Box Testing** focuses on testing the functionality of the **JPetStore** website without considering the internal code structure. This testing validates the application from an end-user perspective, ensuring that it behaves as expected based on the requirements. Black box testing can be performed by testers who are not involved in the development process, providing an honest evaluation of the application's functionality and helping to identify any inconsistencies from the user's point of view.

# Test process

The following diagram shows the test process. The process includes test planning and control, test analysis and design, test implementation and execution, evaluating on exit criteria and reporting, and test closure activities.



Following descriptions give us a short overview of the activities that must be done in each phase.

**Planning -** During the planning phase, define the objectives and approach for testing, identify the resources needed, and create a test plan. The test plan includes identifying test strategy, test tools, and test environments.

* Input: Project requirements and specifications, Scope of the website functionalities, Availability of Resources, such as teams and tools, Risk assessment
* Output artifact: Test Plan Document, Test Strategy, Allocation of Resources, Risk Management Plan

**Analysis** **-** During the analysis phase, requirements and specifications of the website will be reviewed to identify the testable aspects of the system. This includes understanding the functionalities, user interactions, and the data flow within the website.

* Input: Requirements and specifications document, Functional and non-functional requirements, Use cases and user stories
* Output artifact: Requirement Traceability Matrix (RTM), Test Scenarios, Test Data Requirements

**Design** **-** During the design phase, create detailed test cases and test scripts based on the requirements and identified test scenarios in the previous phase. This includes defining the test data, expected results, and any necessary test environment configurations.

* Input: Requirement Traceability Matrix (RTM), Test Scenarios, Test Data Requirements
* Output artifact: Test Cases, Test Scripts, Test Data Sets, Test Environment Setup Plan

**Implementation** **-** During the implementation phase, set up the test environment, configure the necessary tools, and prepare the test data. This phase ensures that everything is ready for the execution of test cases.

* Input: Test Cases, Test Scripts, Test Data Sets, Test Environment Specifications
* Output artifact: Configured Test Environment, Loaded Test Data, Test Execution Schedule

**Execution** **-** During the execution phase, run the test cases and record the results. This involves executing both manual and automated tests, logging defects, and retesting any fixed issues.

* Input: Test Cases and Scripts, Configured Test Environment, Test Data
* Output artifact: Test Execution Logs, Defect Reports, Test Logs

**Evaluation and Reporting** **-** During the evaluation and reporting phase, analyse the test results to determine if the criteria have been met. This includes compiling test metrics, assessing the quality of the product, and preparing test summary reports.

* Input: Test Execution Logs, Defect Reports, Test Logs
* Output artifact: Test Summary Report, Defect Analysis Report, Test Metrics

**Closure -** During the test closure phase, involves finalizing all test activities, ensuring that all test cases have been executed, and all defects have been addressed. This phase also includes knowledge transfer documents, and documenting lessons learned.

* Input: Test Summary Report, Defect Analysis Report, Test Metrics
* Output: Test Closure Report, Knowledge Transfer Documents, Finalized Test Artifacts, such as Test Cases, Scripts, etc., Lessons Learned Document

**Monitoring and Control**

Monitoring and Control tracks the progress of testing activities to ensure they are on schedule and within scope, helping to identify discrepancies from the plan and take corrective measures. This involves activities such as regular status meetings and updates, progress tracking against the test plan, risk management and mitigation, issue tracking and resolution, and metrics collection and analysis.

## When To Create Test Case

Test Cases are created after the analysis phase when the requirements and test scenarios are well comprehended, once the Requirement Traceability Matrix (RTM) is finalized, and when the test environment setup plan is done. This ensures that the test case is aligned with the project requirements.

## When To Begin Testing

Test Cases are created after the analysis phase when the requirements and test scenarios are well comprehended, once the Requirement Traceability Matrix (RTM) is finalized, and when the test environment setup plan is done. This ensures that the test case is aligned with the project requirements.

## Defect Management Process

Microsoft Devops will also be used to track the defect/bug and its resolution. Defects will be reported by the QA team during Test Execution. Reported Bugs will be reviewed and analyzed, make additional changes, and inform the Product Owner regarding the report bugs. The reported bugs will be reviewed and accepted by the Product Owner and make necessary changes to its Severity and Priority.

### When To Log Bugs

Bugs should be logged as soon as they are identified during test execution to ensure that issues are tracked and addressed immediately. This should be done during the test execution, when a test case fails, or when unexpected behavior occurs.

### When To Retest Bugs

Retesting Bugs should begin after the development team has fixed the reported bugs to ensure that the fixes are verified, and the website is functioning as expected. This should be done after the defect is marked as resolved by the development team and a new build containing the fixed bugs is deployed to the test environment.

## Defect Reporting

The purpose of bug reporting is to state the problem as clearly as possible so that developers can replicate and fix it easily. Azure Devops will be used for Defect/Issue tracking. The reported bug should follow the stated guideline and format:

The Bug title should be suggestive enough that the reader can understand it. It must be specific and must provide context to the defect itself.

**“Bug ID – Page/Feature Covered - Bug Short Description”**

The following information is indicated in each bug report:

| Item | Description |
| --- | --- |
| ID | Azure Devops generated ID |
| Description | Short Description of the bug with format: «Page/Feature Covered - Bug Short Description» |
| Environment | Environment where the bug is found. |
| Build Version | Build version where the bug is found. |
| Device Details | Test device used in replicating the bug. |
| User Account | User Account used in replicating the bug. |
| Steps To Reproduce | Step by step description of the way to reproduce the bug. Number the steps. Describe the test environment in detail. Do not assume or skip any reproducing step. |
| Actual Result | The actual result received when the replication steps are followed. |
| Expected Result | The expected results-based User Story acceptance criteria. |
| Attachments | Attach any additional information that will provide help on replicating and determining the bug cause: Screen recording, Screenshots and Log File. After fixing and retesting the bug Attachments is also necessary to support the testing result. |
| Reported By | The name of the person who reported the bug. |
| Assigned To | The name of the person that is assigned to analyze and fix the bug. |
| Status | Bug current status. |
| Severity/Priority | Severity and Priority of the bug. |

## Bug Severity and Priority Definition

The following table explain the severity list:

| Severity | Description |
| --- | --- |
| Critical (Blocker) | A bug is considered Critical in severity if it completely blocks the functionality of the website, making it unusable. This includes system crashes, data loss, or severe security vulnerabilities that must be fixed immediately. |
| High | A bug is considered High in severity if it hinders the website’s functionality without entirely blocking it. This includes major feature failures or significant performance issues that affect user experience. |
| Medium | A bug is considered Medium in severity if it affects some functions but has an alternative solution. This includes small feature failures or moderate performance issues that don’t seriously impact the user experience. The bug may cause some unwanted behavior, but most system features still work well. Medium severity bugs don't cause the system to completely fail but may lead to incorrect, incomplete, or inconsistent results. |
| Low | A bug is considered Low in severity if it has a minimal impact on the functionality or user experience. This includes cosmetic issues, minor usability problems, or trivial performance issues that do not affect the main functionality. |

The following table explain the priority list:

| Priority | Priority Level | Description |
| --- | --- | --- |
| 1 | Critical (Blocker) | A bug should be labeled with Critical Priority when it completely blocks the **JPetStore** website's functionality or causes a system crash, making it unusable. This is the highest priority level, taking priority over all other bugs. Immediate attention and resolution are required. Such bugs include the website not loading, payment processing failures, or login issues. These issues are urgent because they directly impact the user's ability to use the website, leading to poor user experience. The application cannot be released with an open bug of this priority, and it should be fixed within the same day it is reported. |
| 2 | High | A bug should be labeled with High Priority if it hinders major functionality or severely impacts the user experience, even if the **JPetStore** website can still operate. These issues need to be fixed as soon as possible, ideally within the current sprint, to prevent significant delays. If not addressed promptly, such bugs can lead to decreased user trust, and overall dissatisfaction. Issues like preventing users from completing purchases or causing frequent crashes must be resolved immediately to maintain user trust and satisfaction. |
| 3 | Medium | A bug should be labeled with Medium Priority if it affects some functionality but has an alternative solution. These include small feature problems or moderate performance issues that do not critically impact the user experience. For instance, a minor glitch in a non-essential feature or a slight delay in loading times can be noted and fixed in the next sprint if they do not seriously affect how the system works. While these issues should be addressed promptly to ensure smooth operation and prevent them from escalating into more significant problems, they do not require immediate attention. |
| 4 | Low | A bug should be labeled with Low Priority if it has a minimal impact on functionality or user experience. These are usually cosmetic issues or minor usability problems that don't need immediate attention. For instance, a spelling error in the user interface or a minor text alignment issue can be recorded and addressed later if they become more severe or likely, or after all higher-priority bugs have been resolved. Fixing these issues can make the JPetStore look better but does not affect its functionality. |

# Test Organization

| Name | Role | | Responsibility |
| --- | --- | --- | --- |
| Your Name Here (ERNI) | | Test Manager | Defining the Testing Strategy, creating a comprehensive testing plan that outlines the scope, approach, resources, and schedule for testing activities. |
|  | |  | Risk Management and Mitigation: Identifying and mitigating risks in the testing process for **JPetStore**. This includes recognizing potential risks and developing strategies to address them. |
| Romar San Juan (ERNI) | | Quality Assurance Engineer | Test Case Design and Execution: Creating and executing test cases for **JPetStore**, ensuring all functional and non-functional requirements are covered. This includes writing detailed test scenarios and performing manual and automated tests. |
|  | |  | Defect Tracking and Reporting: Identifying, logging, and tracking defects found during testing. This involves reporting issues to the development team, verifying fixes, and ensuring all defects are resolved. |

# Test Environment

**Project Management Tool**

Azure DevOps

**Testing Environment**

The website on Dev Environment

**Test Device**

Standard Workstation or Laptop

**Web Browser Tool**

Microsoft Edge, latest stable version (Version 112.0.1722.34)

**Project Team Communication Tool**

MS Teams

# Test Tasks

| Activity | Task | Lead | Deliverable |
| --- | --- | --- | --- |
| Test Plan Creation | Defining the scope and objectives of testing, identifying the resources required, creating the testing schedule, determining the test environment setup, defining the test deliverables, identifying risks and possibilities, and outlining the test strategy and approach. | Test Manager | Comprehensive Test Plan Document |
| Pre Testing | User Story Refinement | QA Team | Refined User Stories with complete details and understandable acceptance criteria. Examine existing user stories and evaluate whether they are still relevant to the project. |
| Sprint Planning | QA Team | User Stories defined that can be delivered in the sprint and how that work will be achieved. |
| Test Case Creation | QA Team | Test Case Created for each User Stories covering its acceptance criteria. |
| Test Case Validation | QA Team | Test Case validated to ensure that the test case matches and adheres to the acceptance criteria and all expectations under different conditions are covered. |
| Smoke Testing | Smoke Testing | QA Team | Report confirming basic functionality, identifying critical issues or blockers, and verifying build stability for further testing. |
| Integration Testing | Integration Testing | QA Team | Report detailing the successful integration of components, identification of any integration issues, and confirmation that the system functions together seamlessly. |
| Regression Testing | Regression Testing | QA Team | Report confirming that existing functionality remains unaffected by recent changes, identification of any new defects introduced, and verification that the system continues to perform as expected. |
| Bug Reporting | Bug Reporting | QA Team | Bug reports documenting identified defects, including reproduction steps, severity, and relevant information for developers. |
| Sprint Review and Demo | Sprint Review and Demo with the client. | Test Manager | Client-approved presentation and demonstration of completed sprint work, including documented feedback and action items. |
| Pre Production Release | **JPetStore** Pre Production Release. | Test Manager | Approved pre-production release package for **JPetStore**, including final test results, release notes, and deployment instructions. |
| Production Release | **JPetStore** Production Release. | Test Manager | Approved production release package for **JPetStore**, including final test results, release notes, deployment instructions, and confirmation of successful deployment. |
| User Manual | Creation of User Manual. | Test Manager | Comprehensive user manual for **JPetStore**, including detailed instructions, screenshots, and troubleshooting information. |

# Test Report

## Smoke Testing Report

Smoke Testing Report provides a summary of the initial testing phase to ensure that the core functionalities of the website are working as expected. This report helps in identifying any critical issues early in the testing process that could block further testing.

## Regression Testing Report

The Regression Testing Report aims to ensure recent code changes have not negatively impacted existing functionalities and that new updates or fixes do not introduce new defects.

## Test Execution Report

Text Execution Report provides a detailed account of the execution of all test cases, including pass/fail status, execution time, and any deviations from expected results. This report helps in tracking the overall progress and effectiveness of the testing effort.

## Bug Status Report

Bug Status Report tracks and document the status of identified defects, including details such as severity, priority, assigned developer, and current status. This report helps in managing and prioritizing bug fixes and provides visibility into the defect resolution process.

## User Story Refinement Report

User Story Refinement Report documents refined user stories with complete details and acceptance criteria, ensuring clarity and readiness for testing.

## Sprint Planning Report

Sprint Planning Report details the user stories for the sprint and how the work will be achieved, ensuring clear and achievable sprint goals from a testing perspective.

## Test Plan Report

Test Plan Report outlines the testing phase's scope, objectives, resources, schedule, environment setup, deliverables, risks, and strategy, ensuring a comprehensive approach to quality assurance and project success.

## Test Case Creation Report

Test Case Creation Report documents test cases for each user story, covering acceptance criteria to ensure all aspects are tested.

## Test Case Validation Report

Test Case Validation Report validates that test cases match acceptance criteria and cover all expectations under different conditions, ensuring accuracy and completeness.

## Integration Testing Report

Integration Testing Report details successful integration of components, identifies integration issues, and confirms seamless system functionality, providing critical insights for further development and deployment.

## Bug Report

Bug Report identifies defects with reproduction steps, severity, and relevant information for developers to fix issues, facilitating efficient resolution and improving overall software quality.

## Sprint Review and Demo Report

Pre-Production Release Test Report provides final test results, release notes, and deployment instructions for the pre-production release, ensuring readiness for production deployment.

## Pre-Production Release Test Report

Pre-Production Release Test Report provides final test results, release notes, and deployment instructions for the pre-production release, ensuring readiness for production deployment.

## Production Release Test Report

Production Release Test Report provides final test results, release notes, deployment instructions, and confirmation of successful deployment for the production release.

# Acceptance Criteria

## Test Entry Criteria

* Requirements and design documents are complete and approved.
* Test environment is set up and ready.
* Test data is prepared and available.
* Test cases are reviewed and approved.
* Necessary tools and resources are available.
* Development code is complete and has passed initial smoke testing.

## Suspension Criteria

* Critical defects or blockers are identified that prevent further testing.
* Test environment becomes unstable or unavailable.
* Major changes in requirements or design are introduced.
* Insufficient resources to continue testing.
* Significant deviations from the planned test schedule.

## Resumption Criteria

* Critical defects or blockers have been resolved.
* Test environment is restored and stable.
* Updated requirements or design changes are implemented.
* Necessary resources are available again.
* Test schedule is adjusted and approved.

## Test Exit Criteria

* All planned test cases have been executed.
* All critical and high-severity defects are resolved.
* Test coverage meets the defined requirements.
* Test results are reviewed and approved.
* Test deliverables are completed and signed off.
* Stakeholders have approved the test completion.

# Risk Management

| Risk | Prio | Impact | Action needed | Owner |
| --- | --- | --- | --- | --- |
| Test Environment Unavailability | High | Delays in testing activities and project timeline slippage. | Go to IT support for immediate resolution. Consider using a backup environment. | Test Manager |
| Critical Defects in Application | High | Testing is suspended until defects are fixed, causing delays. | Report to development team for urgent fixes and re-prioritize tasks. | QA Lead |
| Insufficient Test Data | Medium | Incomplete testing, potential undetected defects. | Request additional data and collaborate with data team. | Quality Assurance Engineer |
| Resource Unavailability | Medium | Reduced testing capacity possibly missed deadlines. | Reallocate tasks and potential request of additional resources. | Test Manager |
| Requirement Changes Mid-Testing | Medium | Rework needed, causing delays. | Assess impact; update test cases/plans; communicate changes. | Test Analyst |
| Insufficient Test Coverage | Low | Potential undetected defects, quality issues post-release. | Add more test cases and have team members review them. | Test Manager |
| Communication Breakdown | Low | Misunderstandings and delays in resolving issues. | Set up clear communication channels and schedule regular meetings | Test Manager |