ProNef - Test Plan

Test plan revision history

Date	Version	Name	Details
2/28/2024	1.0	Stanislav Kovalchuk	Creation

Test plan identifier

- Project Name: ProNef NFT Marketplace
- Master Test Plan Identifier: PMTP-2024-PRONEF-001
- Revision: 1.0

References

1. IEEE Standard for Software Test Documentation - IEEE Std 829-2008 - this standard provides guidelines for the format and content of documentation for software testing processes, including test plans.
2. ProNef NFT Marketplace Project Description Document - this document outlines the key features, functionalities, and technical details of the ProNef NFT Marketplace project.
3. React Native Documentation - official documentation for React Native, providing insights into the framework used for developing the ProNef mobile application.
4. Expo Documentation - documentation for Expo, the technology used in building the ProNef application, offering cross-platform development capabilities.
5. GitHub Repository for ProNef - the official repository containing the source code for the ProNef NFT Marketplace native app.
6. Expo Store - the Expo Store is referenced for users to view the ProNef app. The link is available on the landing page.
7. GitHub - ProNef Landing Page Source Code - the GitHub repository containing the source code for the ProNef landing page.

These references are essential for understanding the testing standards, project details, and technologies employed in the ProNef NFT Marketplace. They align with the IEEE 829 standard and provide comprehensive information for effective test planning.

Introduction

This Test Plan serves as a comprehensive guide for the testing activities conducted for the ProNef NFT Marketplace project. The purpose of this plan is to outline the approach, methodologies, and strategies employed to ensure the reliability, functionality, and performance of the ProNef application.

Purpose of the Plan

The primary purpose of this test plan is to establish a structured framework for testing activities, encompassing both the ProNef NFT Marketplace native app and the associated landing page. It aims to ensure that the delivered software meets specified requirements, functions as intended, and provides a seamless experience for users engaging in activities such as buying, storing, collecting NFTs, exchanging, and participating in auctions.

Level of the Plan

This test plan represents the Master Test Plan for the ProNef NFT Marketplace project. It provides an overarching view of the testing efforts that will be carried out at different levels, including unit testing, integration testing, system testing, and acceptance testing.

Scope of the Project

The ProNef project involves the development of an NFT marketplace with a React Native mobile application and a dedicated landing page. The application includes features such as NFT listing, detailed NFT views, search functionality, likes, auctions with bidding capabilities, and bid history.

Testing Effort

The testing effort encompasses a comprehensive evaluation of both the React Native mobile application and the landing page. It includes functional testing, performance testing, security testing, and user acceptance testing to ensure the robustness and effectiveness of the ProNef NFT Marketplace.

Relationship to Other Evaluation Activities

Testing activities are closely aligned with other evaluation activities such as code reviews, continuous integration, and user feedback. Regular collaboration with development teams and stakeholders ensures a holistic approach to software quality.

Change Control and Communication

The testing process will adhere to established change control processes. Any modifications to the test plan will be documented, and communication channels will be maintained to ensure all relevant stakeholders are informed of changes and updates.

Coordination of Key Activities

Key testing activities will be coordinated with development, quality assurance, and project management teams. Regular meetings and status updates will facilitate effective communication and collaboration throughout the project lifecycle.

This Test Plan provides a roadmap for testing activities, ensuring the ProNef NFT Marketplace meets high-quality standards and delivers a secure and user-friendly experience to its vast user base.

Test Items (Functions)

The testing scope within this Test Plan encompasses the ProNef NFT Marketplace, including both the React Native application and the landing page. The identified test items and functions to be tested are outlined below:

1. React Native Application:

- **User Interface (UI):** Ensure consistency and responsiveness across various devices.
- **Navigation:** Verify seamless navigation between the list of NFTs and the detailed view of a particular NFT.
- **Search Functionality:** Validate the accuracy and efficiency of the search feature.
- **NFT Likes:** Confirm the proper functioning of the NFT liking system.
- **Auctions with Bids:** Test the functionality of auctions, including bid placement and tracking.
- **Bid History:** Verify the accuracy of bid history display.

2. Landing Page:

- **UI Design:** Confirm that the landing page adheres to the smooth, constant color scheme.
- **Buttons Functionality:** Ensure the functionality of the 'View it on Expo Store' and 'Source Code' buttons.
- **External Links:** Validate the correctness of links to Expo.dev and GitHub.

3. Cross-Platform Compatibility:

- **Device Compatibility:** Test the application on various devices to ensure compatibility and responsiveness.
- **Cross-Browser Compatibility:** Confirm proper rendering and functionality across different web browsers.

4. Integration Testing:

Integration with Expo: Validate that the Expo integration works seamlessly across user devices.

Cryptocurrency Integration: Test the integration with cryptocurrency features, including buying and earning.

5. Performance Testing:

- **Response Time:** Measure and verify the application's response time under different load conditions.

- **Scalability:** Assess the system's scalability in handling a large user base.

6. Security Testing:

- **Data Encryption:** Verify the encryption of sensitive data such as user information and transaction details.
- **Authentication and Authorization:** Test the effectiveness of user authentication and authorization processes.

7. Documentation Validation:

- **Expo Documentation:** Ensure that the documentation provided by Expo aligns with the application's behavior.
- **GitHub Repository:** Validate the correctness of the source code and documentation available on GitHub.

The testing of these items will contribute to the overall assurance of quality for the ProNef NFT Marketplace, addressing the specified requirements and ensuring a reliable and user-friendly experience for the 25 million users.

Software Risk Issues

Identifying and addressing software risks is crucial for the successful testing and deployment of the ProNef NFT Marketplace. The following are potential risk areas that require careful consideration:

1. Third-Party Product Integration:

- **Risk:** Dependency on third-party products for specific functionalities may introduce compatibility issues or delays in updates.
- **Mitigation:** Regular communication and coordination with third-party providers, early testing of updates, and having contingency plans for potential issues.

2. Interfacing Software Version:

- **Risk:** The introduction of a new version of interfacing software may lead to compatibility challenges and disruptions in data flow.
- **Mitigation:** Prior testing and validation of the new interfacing software version in a controlled environment before integration into the ProNef NFT Marketplace.

3. Complex Cryptocurrency Functions:

- **Risk:** The complexity of cryptocurrency-related functions, such as buying and earning, may lead to transaction errors and security vulnerabilities.
- **Mitigation:** Rigorous testing of cryptocurrency functionalities, including security testing, to identify and address potential issues.

4. Modification of Historical Failure Components:

- **Risk:** Making modifications to components with a past history of failure may reintroduce defects or create new issues.
- **Mitigation:** In-depth testing of modified components, utilizing insights from past failures to guide testing efforts, and implementing comprehensive regression testing.

5. Poorly Documented Modules and Change Requests:

- **Risk:** Inadequate documentation and unclear change requests may lead to misunderstandings, resulting in defects or incomplete testing.

- **Mitigation:** Regular communication with developers, obtaining detailed documentation, and clarification of ambiguous requirements through close collaboration with stakeholders.

6. Safety and Compliance:

- **Risk:** Failure to comply with safety standards or government regulations may lead to legal and reputational consequences.
- **Mitigation:** Incorporating compliance checks and testing safety-critical functionalities to ensure adherence to regulations and standards.

7. Misunderstanding of Original Requirements:

- **Risk:** Misinterpretation of requirements at management, user, or developer levels may result in incomplete or inaccurate testing.
- **Mitigation:** Conducting thorough requirement analysis, continuous communication with stakeholders, and regular validation of testing against the original specifications.

8. Defect-Prone Areas from Unit Testing History:

- **Risk:** Past history of defects discovered during unit testing may indicate potential ongoing issues in specific areas of the software.
- **Mitigation:** Focused testing in historically defect-prone areas, additional scrutiny, and applying lessons learned from previous defects.

By proactively identifying and mitigating these software risk issues, the testing team aims to enhance the reliability, security, and overall quality of the ProNef NFT Marketplace.

Features To Be Tested

1. User Interface (UI) Consistency:

- **Description:** Ensure that the UI design is consistent across all screens and devices.
- **User Viewpoint:** Users should experience a visually cohesive and intuitive interface.
- **Risk Level:** Low (L) - Visual inconsistencies may impact user experience but are unlikely to pose severe risks to functionality.

2. Navigation:

- **Description:** Validate the ease of navigation between the list of NFTs and the detailed view of a specific NFT.
- **User Viewpoint:** Users should be able to effortlessly explore NFTs and access detailed information.
- **Risk Level:** Medium (M) - Navigation issues could impact user engagement, making it a moderate-risk area.

3. Search Functionality:

- **Description:** Confirm the accuracy and efficiency of the search feature for finding specific NFTs.
- **User Viewpoint:** Users should be able to quickly locate desired NFTs through the search functionality.

- **Risk Level:** High (H) - Inaccuracies in search results may frustrate users and hinder the primary purpose of the marketplace.

4. NFT Likes:

- **Description:** Test the functionality of liking NFTs and ensuring proper tracking.
- **User Viewpoint:** Users should be able to express their preferences for NFTs, and likes should be recorded accurately.
- **Risk Level:** Medium (M) - Issues with liking functionality may affect user engagement but are not critical to core functionality.

5. Auctions with Bids:

- **Description:** Validate the proper functioning of auctions, including bid placement and tracking.
- **User Viewpoint:** Users should be able to participate in auctions seamlessly.
- **Risk Level:** High (H) - Auctions are a critical feature, and any issues may lead to financial implications and dissatisfaction.

6. Bid History:

- **Description:** Verify the accuracy of bid history display for transparency.
- **User Viewpoint:** Users should have access to reliable information about past bids.
- **Risk Level:** Medium (M) - Inaccuracies in bid history may impact user trust, but the overall system functionality is not compromised.

7. UI Design (Landing Page):

- **Description:** Confirm that the landing page adheres to the smooth, constant color scheme.
- **User Viewpoint:** Users should experience a visually appealing and consistent landing page.
- **Risk Level:** Low (L) - Aesthetic issues may impact user perception but are unlikely to affect core functionality.

8. External Links (Expo Store, GitHub):

- **Description:** Validate the correctness of links to Expo.dev and GitHub from the landing page.
- **User Viewpoint:** Users should be able to access external resources seamlessly.
- **Risk Level:** Medium (M) - Incorrect links may lead to user frustration but are not critical to the core functionality.

9. Device Compatibility:

- **Description:** Test the application on various devices to ensure compatibility and responsiveness.
- **User Viewpoint:** Users should have a consistent experience across different devices.
- **Risk Level:** High (H) - Compatibility issues may severely impact user satisfaction and engagement.

10. Cryptocurrency Functions:

- **Description:** Test buying, storing, collecting, exchanging NFTs, and earning cryptocurrency.

- **User Viewpoint:** Users should be able to perform cryptocurrency-related actions seamlessly and securely.
- **Risk Level:** High (H) - Any issues with cryptocurrency functionalities may have financial implications and impact user trust.

This listing of features to be tested considers the user's perspective and assigns risk levels based on potential impact on user satisfaction, engagement, and the criticality of the functionality to the overall system.

Features Not To Be Tested

1. Obsolete Features:

- **Reason:** Features that have been deprecated or declared obsolete and will not be included in this release.
- **Explanation:** These features are no longer part of the active development roadmap, and their exclusion is a strategic decision to focus testing efforts on current functionalities.

2. Legacy UI Elements:

- **Reason:** UI elements that have been replaced or significantly updated in the current release.
- **Explanation:** Testing these elements is unnecessary as they have been superseded, and the focus is on ensuring the functionality and stability of the updated UI.

3. Out-of-Scope Platforms:

- **Reason:** Platforms or devices not specified in the project scope.
- **Explanation:** Testing on platforms beyond the defined scope would extend the testing effort beyond the project's current objectives, introducing unnecessary complexity.

4. Third-Party Product Features Beyond Integration:

- **Reason:** Features of third-party products that extend beyond integration points.
- **Explanation:** The scope of testing for third-party products is limited to integration points, as comprehensive testing of all features of external products is impractical.

5. Non-Cryptocurrency Related Financial Transactions:

- **Reason:** Features involving financial transactions not directly related to cryptocurrency actions.
- **Explanation:** Focusing on cryptocurrency-related transactions aligns with the project's priorities, and other financial transactions are considered out of scope for this release.

6. Non-Critical UI Aesthetic Adjustments:

- **Reason:** Minor UI adjustments that do not impact functionality or user experience significantly.
- **Explanation:** While UI aesthetics are important, minor adjustments that do not pose a risk to functionality or user satisfaction are not a priority for testing in this release.

7. Documentation Verification:

- **Reason:** Verification of external documentation beyond basic validation of links.
- **Explanation:** In-depth verification of extensive documentation is beyond the scope of this testing plan. Basic validation ensures users can access the provided resources.

8. Non-Functional Elements of Landing Page:

- **Reason:** Non-functional elements of the landing page that do not impact user interaction.
- **Explanation:** Elements such as decorative images or static text that do not affect user functionality are excluded from testing to streamline efforts.

The exclusion of these features from the testing scope is based on factors such as obsolescence, strategic priorities, and the need to focus on critical functionalities within the defined project scope. This allows for a more efficient and targeted testing process.

Approach (Strategy)

1. Overview:

- This test plan serves as a Master Test Plan for the ProNef NFT Marketplace project, providing an overarching strategy for testing at various levels. It aligns with both higher and lower-level plans.

2. Testing Levels:

- The testing levels include Unit Testing, Integration Testing, System Testing, and Acceptance Testing.
- Each level will have a specific focus on validating different aspects, from individual components to the overall user experience.

3. Test Process:

- Testing will follow the IEEE 829 standard and will encompass various testing activities such as test design, execution, and defect tracking.
- Test activities will be integrated into the development life cycle, promoting continuous feedback and collaboration.

4. Special Tools:

- Testing tools like Jira for defect tracking, Selenium for automated UI testing, and performance testing tools will be employed.
- Training sessions will be conducted to ensure the team is proficient in using these tools effectively.

5. Metrics Collection:

- Metrics such as test coverage, defect density, and test execution progress will be collected.
- Test metrics will be collected at each testing level to assess the effectiveness of the testing process.

6. Configuration Management:

- Configuration Management will involve version control for the application code, ensuring that the correct version is tested at each level.
- Different configurations, including hardware, software, and combinations, will be tested to cover diverse user environments.

7. Regression Testing:

- Regression testing will be conducted at each testing level to ensure that new changes do not adversely affect existing functionalities.
- The level of regression testing will be proportional to the impact of changes and severity of defects detected.

8. Handling Untestable Requirements:

- Requirements and design elements that are unclear or untestable will be flagged for clarification through close collaboration with stakeholders.
- Ambiguities will be resolved through regular communication and iteration with the development and business teams.

9. Project Testing Approach:

- The project testing approach is comprehensive, covering critical functionalities such as cryptocurrency transactions, user interface, and third-party integrations.
- Special requirements include thorough testing of the full component, including both the React Native app and the landing page.

10. Reliability Measures:

- MTBF (Mean Time Between Failures) will be monitored, where applicable, to measure the reliability of critical components.
- SRE (Software Reliability Engineering) principles will be applied, focusing on improving reliability through proactive identification and mitigation of potential failures.

11. Meetings and Organizational Processes:

- Regular meetings will be held to discuss progress, challenges, and priorities.
- Agile methodologies will be employed to adapt to changing requirements and ensure continuous improvement.

This testing approach aims to provide a structured and comprehensive strategy for testing the ProNef NFT Marketplace, aligning with industry standards and project-specific requirements. It emphasizes collaboration, efficiency, and the delivery of a reliable and user-friendly product.

Item Pass/Fail Criteria

1. Unit Testing:

- **Criteria:**
  - All unit test cases executed.
  - Code coverage tool indicates a minimum of 90% code coverage.
  - Zero critical defects. Minor defects are accepted but should not exceed 5% of executed test cases.
- **Pass:** If all criteria are met, indicating the stability and completeness of individual components.

2. Integration Testing:

- **Criteria:**
  - Successful integration of all components.
  - Zero critical defects. Minor defects should be less than 5% of executed test cases.
- **Pass:** If the integration testing process is smooth with minimal defects, ensuring that components work seamlessly together.

3. System Testing:

- **Criteria:**
  - All system test cases executed.

  - Functional requirements validation without critical defects.
  - Performance metrics within acceptable limits.
- **Pass:** If the system meets functional and performance requirements without critical defects.

4. Acceptance Testing:

- **Criteria:**
  - Successful completion of all acceptance test cases.
  - User acceptance of the application based on predefined criteria.
- **Pass:** If the application is accepted by users without critical issues.

5. Master Test Plan Level:

- **Criteria:**
  - Completion of all lower-level plans.
  - Percentage of lower-level plans without errors should be at least 90%.
  - Defect density across all levels should be less than 5%.
- **Pass:** If all lower-level plans are completed successfully with minimal defects.

6. Defect Metrics:

- **Criteria:**
  - Total number of defects located and resolved.
  - Severity distribution: Critical (0%), Major (10% or less), Minor (90% or less).
- **Pass:** If the number and severity of defects align with the predefined criteria.

7. Comparison to Total Defects:

- **Criteria:**
  - Comparison of the total number of detected defects to the total estimated defects.
- **Pass:** If a reasonable percentage of total defects are detected, indicating effective testing coverage.

8. Defect vs. Failure Analysis:

- **Criteria:**
  - Analysis of defects that led to failures.
  - Identification of defects acceptable to leave in the application without causing failures.
- **Pass:** If the defects are managed effectively, minimizing failures and addressing critical issues.

These criteria are designed to assess the completion and effectiveness of testing at different levels, providing a comprehensive evaluation of the ProNef NFT Marketplace before release.

Suspension Criteria And Resumption Requirements

Testing activities may be temporarily suspended under the following conditions, with specific criteria and requirements for resumption:

1. Critical Defects:

- **Suspension Criteria:** If a critical defect is identified, which significantly impacts the core functionality or security of the ProNef NFT Marketplace.
- **Resumption Requirements:** Testing will resume only after the critical defect is addressed, retested, and verified as resolved.

2. Showstopper Defects:

- **Suspension Criteria:** Identification of a showstopper defect that halts the testing process, preventing further progress.
- **Resumption Requirements:** Testing will resume once the showstopper defect is resolved, and the affected areas are retested to ensure stability.

3. High Severity Defect Cluster:

- **Suspension Criteria:** If a cluster of high-severity defects is detected in a specific component or feature, hindering effective testing.
- **Resumption Requirements:** Testing will resume after addressing and resolving the high-severity defects. The affected areas will undergo thorough retesting.

4. Resource Constraints:

- **Suspension Criteria:** Resource constraints, such as unavailability of necessary testing environments or personnel.
- **Resumption Requirements:** Testing will resume upon the availability of required resources, ensuring a consistent testing environment.

5. Inadequate Test Coverage:

- **Suspension Criteria:** Test coverage falls below the predefined threshold, limiting the effectiveness of testing.
- **Resumption Requirements:** Testing will resume after adjusting test plans to enhance coverage and ensure a more comprehensive evaluation.

6. Unstable Test Environment:

- **Suspension Criteria:** The test environment becomes unstable, leading to unreliable test results.
- **Resumption Requirements:** Testing will resume after stabilizing the test environment, and affected test cases will be rerun for validation.

7. Ambiguous Requirements:

- **Suspension Criteria:** Identification of ambiguous or unclear requirements that hinder effective testing.
- **Resumption Requirements:** Testing will resume after clarification and resolution of ambiguous requirements. Adjustments to test cases may be necessary.

8. External Dependencies:

- **Suspension Criteria:** Unavailability or issues with external dependencies, such as third-party APIs or services.
- **Resumption Requirements:** Testing will resume once external dependencies are restored, and affected test cases will be rerun for validation.

The decision to suspend testing will be made in collaboration with the project stakeholders, and resumption will occur only when the identified issues are addressed and the testing environment is conducive to effective evaluation. This approach ensures that testing resources are utilized efficiently and that the testing process contributes meaningful insights to the quality of the ProNef NFT Marketplace.

Test Deliverables

The following test deliverables will be produced as part of the testing process for the ProNef NFT Marketplace:

1. Test Plan Document:

- A comprehensive document outlining the testing strategy, approach, scope, and details of the testing activities.

2. Test Cases:

- Detailed test cases designed to validate the functionality, performance, and security aspects of the ProNef NFT Marketplace.

3. Test Design Specifications:

- Documentation providing insights into the design of test cases, including test data, expected results, and testing conditions.

4. Tools and Outputs:

- Documentation on the utilization of testing tools, such as Selenium and Jest, and the corresponding outputs generated during automated testing.

5. Simulators:

- Simulators used for emulating specific scenarios or conditions during testing, ensuring a comprehensive evaluation of the application.

6. Static and Dynamic Generators:

- Tools or scripts used for generating static and dynamic test data to simulate different user scenarios.

7. Error Logs and Execution Logs:

- Logs capturing errors, defects, and the execution history of test cases. These logs will provide a detailed record of testing activities.

8. Problem Reports and Corrective Actions:

- Reports documenting identified problems, defects, and corresponding corrective actions taken to address these issues.

9. Traceability Matrix:

- A matrix mapping test cases back to requirements, ensuring comprehensive coverage and traceability throughout the testing process.

10. Test Summary Report:

- A summary document outlining the overall results of the testing activities, including key metrics, defect statistics, and recommendations for release.

11. Test Environment Configuration:

- Documentation detailing the configuration of the test environment, including hardware specifications, software versions, and network configurations.

12. Training Materials:

- If applicable, materials related to the training of testing team members on the use of testing tools and processes.

These test deliverables are essential for providing transparency, accountability, and documentation throughout the testing lifecycle. They serve as valuable artifacts for stakeholders, development teams, and quality assurance professionals, ensuring a thorough understanding of the testing process and its outcomes.

Remaining Test Tasks

While this test plan addresses specific functionalities and features of the ProNef NFT Marketplace, it's important to acknowledge that certain areas of the application may not be within the scope of this plan. This section highlights the remaining test tasks that are not covered in the current phase:

1. Future Phases and Incremental Releases:

- The ProNef NFT Marketplace may undergo further development and enhancement in future phases or incremental releases. Test tasks related to these upcoming phases are not covered in this plan.

2. Incomplete Functions and Features:

- Some functions or features of the application may still be under development or planning, and testing for these incomplete areas is not addressed in the current plan.

3. Multi-Party Development:

- If the project involves multiple parties, such as third-party developers, external testing teams, or collaborative efforts, this plan may only cover a subset of the total functions/features. Other areas may have separate plans developed by different parties.

4. Interconnected Systems:

- If the ProNef NFT Marketplace interacts with external systems, services, or components that are not fully integrated or tested within the scope of this plan, those interactions may be considered for future testing efforts.

5. External Dependencies:

- Dependencies on external components, services, or APIs that are not thoroughly tested in the current plan may require separate testing efforts in coordination with external teams or stakeholders.

6. Cross-Platform Compatibility:

- While this plan addresses cross-platform compatibility within the React Native application, compatibility testing with other platforms or devices may be subject to future testing.

7. User Experience Enhancements:

- User experience enhancements, design changes, or additional features that are not part of the current plan may require dedicated testing efforts in subsequent phases.

8. Performance Tuning:

- In-depth performance tuning and optimization efforts may be necessary in the future, and specific testing tasks related to these activities are not included in this plan.

To avoid confusion and resource wastage, it is recommended to develop separate test plans for these remaining test tasks. This ensures that testing efforts align with the ongoing development and release cycle of the ProNef NFT Marketplace, allowing for comprehensive coverage and effective defect tracking. Coordination with development teams, external parties, and stakeholders will be crucial for successful testing in these remaining areas.

Environmental Needs

The ProNef NFT Marketplace Test Plan outlines specific environmental needs to ensure a controlled and efficient testing process. These requirements encompass various aspects, including hardware, test data, testing on multi-part features, power considerations, software versions, and system usage during testing.

Hardware Requirements:

1. Mobile Device Simulators:

- **Purpose:** Simulate different mobile devices to test the React Native mobile application on various screen sizes and resolutions.
- **Specification:** Use emulators or simulators supporting popular devices and operating systems.

2. Static and Dynamic Data Generators:

- **Purpose:** Generate static and dynamic test data to cover a range of scenarios and conditions.
- **Specification:** Implement tools for generating realistic and diverse data for testing.

Test Data Provision:

1. Test Data Sources:

- **Source:** Utilize real-world data sources to create authentic test scenarios.
- **Specification:** Access databases or repositories that mirror the expected data in a production environment.

2. Data Ranges:

- **Specification:** Define specific ranges of data to be used in testing to cover a spectrum of conditions.

Multi-Part Feature Testing:

1. Component-wise Testing:

- **Approach:** Test each component of a multi-part feature individually to ensure functionality and integration.
- **Specification:** Develop a testing strategy that addresses each component's unique functionality.

Power Requirements:

1. Battery Consumption Testing:

- **Purpose:** Evaluate the application's impact on device battery life during prolonged usage.

- **Specification:** Test on devices with varying battery capacities to assess power consumption.

Software Versions:

1. Platform Version Compatibility:

- **Requirement:** Ensure compatibility with specific versions of the React Native platform and Expo.
- **Specification:** Specify the supported versions in the test environment configuration.

2. Browser Compatibility:

- **Requirement:** Validate the compatibility of the landing page with specified web browsers.
- **Specification:** Test on browsers such as Chrome, Firefox, and Safari.

System Usage during Testing:

1. Restricted User Access:

- **Requirement:** Limit access to the system during critical testing phases to prevent interference.
- **Specification:** Communicate testing schedules to stakeholders and restrict user access as needed.

2. Load Testing Considerations:

- **Requirement:** Plan load testing during off-peak hours to minimize the impact on live users.
- **Specification:** Schedule load testing activities during times of lower user activity.

Special Configurations:

1. Wallet Integration Environment:

- **Requirement:** Set up a dedicated environment for testing cryptocurrency transactions and wallet integrations.
- **Specification:** Isolate the environment to avoid real transactions during testing.

2. Expo Store and GitHub Links:

- **Requirement:** Validate the functionality of external links to Expo Store and GitHub.
- **Specification:** Use specific test environments to avoid redirection to live Expo Store or GitHub pages.

These environmental needs are crucial to creating a controlled and representative testing environment for the ProNef NFT Marketplace. The specifications ensure that testing is conducted under conditions that mimic real-world scenarios, leading to more accurate assessments of the application's performance, functionality, and compatibility.

Staffing And Training Needs

To ensure the successful execution of the ProNef NFT Marketplace Test Plan, attention must be given to staffing and training needs. This section outlines the training requirements for the application/system, test tools to be used, and the responsibilities of individuals involved in testing and training.

Application/System Training:

1. Purpose:

- Ensure that testing personnel are well-acquainted with the ProNef NFT Marketplace, understanding its functionalities, user interfaces, and key features.

2. Training Content:

- Cover the entire application workflow, emphasizing critical areas such as cryptocurrency transactions, NFT likes, auctions, and bid history.

3. Training Methods:

- Utilize training sessions, documentation, and walkthroughs to familiarize testing personnel with the application.

4. Target Audience:

- Testing personnel involved in the execution of test cases, test automation, and other testing-related activities.

Test Tool Training:

1. Tools: Selenium, Appium, JMeter, Gatling:

- **Purpose:** Equip testing personnel with the necessary skills to effectively use test automation and performance testing tools.
- **Training Content:** Cover tool functionalities, scripting techniques, and best practices for test automation and performance testing.

2. Tool-Specific Training Sessions:

- Conduct hands-on training sessions led by experienced practitioners to ensure a practical understanding of tool usage.

3. Continuous Learning:

- Encourage continuous learning and exploration of new features or updates to the testing tools to maximize efficiency.

Testing and Training Responsibilities:

1. Test Case Execution:

- **Responsibility:** Testing personnel responsible for executing test cases as per the test plan.
- **Training Needs:** Thorough understanding of test case design, execution procedures, and defect reporting.

2. Test Automation:

- **Responsibility:** Test automation engineers responsible for developing and maintaining automated test scripts.
- **Training Needs:** Proficiency in scripting languages (e.g., Java, JavaScript) and automation tool usage (Selenium, Appium).

3. Performance Testing:

- **Responsibility:** Performance testing specialists responsible for designing and executing performance tests.
- **Training Needs:** In-depth knowledge of performance testing concepts, tools (JMeter, Gatling), and analysis of test results.

4. Defect Reporting and Tracking:

- **Responsibility:** Testing personnel responsible for reporting defects and tracking their status.
- **Training Needs:** Familiarity with defect reporting tools and clear understanding of defect life cycle.

5. Stakeholder Communication:

- **Responsibility:** Test leads or coordinators responsible for communicating testing progress to stakeholders.
- **Training Needs:** Effective communication skills and regular status reporting.

6. Training Coordinator:

- **Responsibility:** Designated individual responsible for coordinating training sessions and ensuring that all testing personnel receive necessary training.
- **Training Needs:** Organizational and coordination skills, understanding of training requirements.

Continuous Improvement:

1. Regular Feedback Mechanism:

- Establish a feedback mechanism for continuous improvement, allowing testing personnel to provide insights on training effectiveness and suggest improvements.

2. Skill Enhancement Programs:

- Implement skill enhancement programs to address emerging testing challenges and new technologies.

By addressing staffing and training needs, the testing team can enhance their capabilities, ensuring a proficient and well-prepared approach to testing the ProNef NFT Marketplace. Continuous learning and improvement will contribute to the overall success of the testing process.

Responsibilities

The successful execution of the ProNef NFT Marketplace Test Plan relies on clear delineation of responsibilities across various areas of the testing process. Each aspect of the plan involves specific individuals or roles to ensure a well-coordinated and effective testing effort.

Test Planning and Strategy:

1. Test Strategy Development:

Responsible Party: Test Manager or Test Lead.

Responsibilities:

- Develop the overall test strategy for the ProNef NFT Marketplace, aligning with project goals and objectives.
- Identify risks, select features to be tested, and define the overall testing approach for this level of the plan.

2. Risk Identification and Mitigation:

Responsible Party: Test Manager or Risk Manager.

Responsibilities:

- Identify potential risks associated with the testing process and the application.
- Develop strategies for mitigating identified risks and ensuring risk awareness among the testing team.

Test Execution and Coordination:

1. Test Case Execution:

Responsible Party: Testing Team (Testers).

Responsibilities:

- Execute test cases according to the defined test plan and strategy.
- Report defects, track their status, and provide feedback on test case execution.

2. Test Automation:

Responsible Party: Test Automation Engineers.

Responsibilities:

- Develop and maintain automated test scripts using Selenium and Appium.
- Ensure the reliability and effectiveness of automated tests.

3. Performance Testing:

Responsible Party: Performance Testing Specialists.

Responsibilities:

- Design and execute performance tests using JMeter and Gatling.
- Analyze performance test results and provide recommendations for improvements.

Training and Skill Development:

1. Application/System Training:

Responsible Party: Training Coordinator.

Responsibilities:

- Organize training sessions for testing personnel to ensure familiarity with the ProNef NFT Marketplace.
- Cover application workflow, key features, and critical functionalities.

2. Test Tool Training:

Responsible Party: Training Coordinator.

Responsibilities:

- Coordinate training sessions for test automation and performance testing tools (Selenium, Appium, JMeter, Gatling).
- Facilitate hands-on sessions for tool proficiency.

Decision-Making and Conflict Resolution:

1. Go/No-Go Decisions:

Responsible Party: Project Manager, Test Manager, or Test Lead.

Responsibilities:

- Make critical go/no-go decisions for items not explicitly covered in the test plans.
- Assess overall readiness for deployment based on test results and other relevant factors.

2. Scheduling Conflicts Resolution:

Responsible Party: Project Manager or Test Manager.

Responsibilities:

- Resolve scheduling conflicts, especially if testing is conducted on the production system.
- Ensure that testing activities align with project timelines and priorities.

Continuous Improvement:

1. Feedback and Improvement:

Responsible Party: Test Manager or Continuous Improvement Coordinator.

Responsibilities:

- Establish a feedback mechanism for continuous improvement in testing processes.
- Implement suggestions for enhancing testing efficiency and effectiveness.

2. Skill Enhancement Programs:

Responsible Party: Training Coordinator.

Responsibilities:

Develop and implement skill enhancement programs to address emerging testing challenges and new technologies.

By clearly defining responsibilities in each area, the testing team can operate cohesively, ensuring that each aspect of the ProNef NFT Marketplace Test Plan is executed efficiently and in alignment with project objectives.

Schedule

The schedule for the ProNef NFT Marketplace testing process is meticulously designed to align with the development timeline, ensuring thorough testing without compromising on the delivery timeline. Realistic and validated estimates form the basis of this schedule, and any potential slippage in the development process is addressed transparently. The schedule is structured to emphasize the importance of comprehensive testing in delivering a robust and reliable application.

Milestones and Dependencies:

1. Development Milestones:

- Identify key development milestones and activities crucial to the delivery of the ProNef NFT Marketplace.
- Establish a clear relationship between development activities and corresponding testing phases.

2. Dependent or Relative Dating:

- Implement dependent dating to tie all test dates directly to their related development activity dates.
- Ensure that testing activities commence immediately after the completion or delivery of the associated development deliverables.

Testing Phases:

1. Unit Testing:

- Commence unit testing activities in parallel with module development.
- Ensure that each module undergoes unit testing promptly upon completion.

2. Integration Testing:

- Initiate integration testing as modules are integrated to form the complete application.
- Leverage continuous integration practices to streamline the integration testing process.

3. System Testing:

- System testing starts the day after the delivery of the final build.
- Adopt a continuous testing approach to identify and address issues as the system evolves.

4. User Acceptance Testing (UAT):

- UAT is scheduled to begin upon successful completion of system testing.
- Engage users early in the testing process to gather valuable feedback and ensure alignment with user expectations.

Handling Slippage:

1. Slippage Identification:

- Monitor the development milestones and identify any potential slippage in the schedule.
- Implement a proactive approach to detect slippage early in the process.

2. Impact Assessment:

- Clearly outline the effects of schedule slippage on the testing process and overall system delivery.
- Discuss potential scenarios and their impact on the quality of the application.

3. User Agreement on Defects:

- Engage users in discussions about potential slippage and its impact.
- Seek user agreement on potential trade-offs, such as tolerance for a specified number of defects in case of schedule slippage.

4. Transparent Communication:

- Maintain transparent communication with stakeholders regarding any identified slippage and its implications.
- Collaborate on solutions and adjustments to mitigate the impact on the overall project timeline.

Continuous Monitoring and Adaptation:

1. Regular Progress Meetings:

- Conduct regular progress meetings to assess the status of development and testing activities.
- Identify any deviations from the schedule and implement corrective actions promptly.

2. Agile Adaptation:

- Embrace an agile mindset to adapt the testing schedule based on evolving development requirements.
- Prioritize collaboration and flexibility to accommodate changes while maintaining the testing rigor.

The ProNef NFT Marketplace testing schedule is designed to foster collaboration between development and testing teams, ensuring a synchronized and quality-driven delivery of the application. The approach of tying test dates directly to development activity dates minimizes the perception of the test team as a cause of delays and promotes a cohesive project timeline.

Planning Risks And Contingencies

The success of the ProNef NFT Marketplace testing process relies on a comprehensive identification of potential risks and the formulation of effective contingencies. The following outlines potential risks to the project, particularly within the testing process, and the corresponding contingency plans:

Overall Risks to the Project:

1. Personnel Resource Constraints:

Risk: Insufficient testing personnel resources when testing is scheduled to begin.

Contingency:

- If resource shortages are identified, additional personnel will be recruited or existing team members may be reassigned from other areas with lower priority.

2. Availability of Resources:

Risk: Lack of availability of required hardware, software, data, or tools for testing.

Contingency:

- Establish a proactive procurement plan to ensure timely availability of all required resources.
- Identify alternative resources or tools to mitigate potential delays.

3. Late Delivery of Software, Hardware, or Tools:

Risk: Delays in the delivery of the software, hardware, or testing tools.

Contingency:

- Adjust the testing schedule to accommodate the updated delivery timeline.
- Prioritize critical testing activities to minimize the impact on overall project timelines.

4. Training Delays:

Risk: Delays in training on the application and/or testing tools.

Contingency:

- Conduct early training sessions to ensure testing personnel are well-prepared.
- Explore alternative training methods, such as online resources, to mitigate delays.

5. Changes to Requirements or Designs:

Risk: Changes to the original requirements or designs after the planning phase.

Contingency:

- Establish a change management process to document and assess the impact of changes.
- Communicate changes promptly to the testing team and adjust the testing schedule accordingly.

Contingency Actions for Various Events:

1. Requirements Changes:

Contingency Actions:

- If requirements change after a specified date, adjust the test and development schedules accordingly.
- Evaluate the impact on testing resources, and, if necessary, allocate additional resources.

2. Reduced Testing Scope:

Contingency Actions:

- In the event of reduced testing scope, prioritize critical test cases and functionalities.
- Clearly communicate the impact on overall quality and assess the acceptability of increased defects.

3. Resource Optimization:

Contingency Actions:

- Optimize testing resources by redistributing tasks and responsibilities within the team.
- Identify opportunities for resource sharing across different phases of testing.

4. Scope Changes:
Contingency Actions:

- If there are changes to the scope of the plan, assess the impact on testing objectives and coverage.
- Collaborate with stakeholders to avoid unnecessary scope changes and maintain testing integrity.

5. Work Overtime:
Contingency Actions:

- If necessary, the test team may work overtime to meet critical deadlines.
- Monitor team morale and well-being, and address concerns promptly.

Avoiding Extreme Scenarios:

1. Scenario Optimization:
Avoidance Approach:

- Strive to avoid extreme scenarios, such as quitting or major scope changes, by proactive risk management.
- Continuously assess and adapt to evolving project dynamics to minimize the need for drastic measures.

2. Communication and Collaboration:
Avoidance Approach:

- Maintain open and transparent communication with all stakeholders to address challenges collaboratively.
- Foster a culture of proactive risk identification and resolution to avoid extreme scenarios.

Conclusion:

The Planning Risks And Contingencies section underscores the importance of proactive risk management and the development of effective contingency plans to address unforeseen challenges in the ProNef NFT Marketplace testing process. The goal is to maintain the quality and integrity of testing activities even in the face of potential disruptions, fostering a resilient and adaptive testing approach.

Approvals

The approval process for the ProNef NFT Marketplace testing process is designed to ensure comprehensive validation at each level, with considerations for the varying knowledge levels and perspectives of the involved parties. The approval authority depends on the specific level of the test plan, recognizing the different audiences and their expertise.

Unit Test Level Plan:

For Unit Test Level Plans, the approval authority includes:

1. Test Engineer or Tester:

- Approves the completion of the unit test plan, ensuring alignment with the detailed technical specifications and testing requirements.
- Validates that unit test cases adequately cover individual components and modules.

2. Development Team Lead:

- Reviews the unit test plan for coherence with development objectives and technical requirements.
- Ensures that the unit testing approach aligns with the overall development strategy.

Integration Test Level Plan:

For Integration Test Level Plans, the approval authority includes:

1. Integration Test Lead:

- Approves the completion of the integration test plan, verifying that it addresses the integration points and dependencies between modules.
- Validates that integration test cases effectively assess the interoperability of system components.

2. System Architect or Technical Lead:

- Reviews the integration test plan from a technical architecture perspective.
- Ensures that the plan aligns with the overall system design and architecture.

System Test Level Plan:

For System Test Level Plans, the approval authority includes:

1. System Test Manager or Test Lead:

- Approves the completion of the system test plan, ensuring comprehensive coverage of the entire application.
- Validates that system test cases address end-to-end scenarios and overall system functionalities.

2. Business Analyst or Product Owner:

- Reviews the system test plan with a focus on business requirements and user expectations.
- Ensures that the plan aligns with the intended business processes and objectives.

Master Test Plan:

For the Master Test Plan, the approval authority includes:

1. Project Manager:

- Approves the overall master test plan, ensuring alignment with project goals and timelines.
- Validates that the testing approach supports the successful delivery of the ProNef NFT Marketplace.

2. Quality Assurance Manager:

- Reviews the master test plan for adherence to quality assurance standards and best practices.
- Ensures that the plan incorporates comprehensive testing strategies.

3. Stakeholders and Project Sponsors:

- Validate that the master test plan aligns with overall project objectives and expectations.
- Approve the plan from a strategic and business perspective.

Approval Process:

1. Review and Feedback:

- All involved parties review the respective test plans, providing feedback and suggestions for improvement.

2. Approval Meeting:

- A formal meeting is held to discuss the test plan, address any concerns, and seek consensus on approval.

3. Documentation:

- The approved test plan is documented, and signatures or formal acknowledgment from the approving authorities are obtained.

4. Distribution:

- The approved test plan is distributed to relevant stakeholders, ensuring that all team members are aligned with the testing approach.

The approvals process is designed to foster collaboration, ensuring that each level of the ProNef NFT Marketplace testing process receives validation from the appropriate authorities with the necessary expertise and perspectives. This approach aims to enhance the overall quality and effectiveness of the testing activities.

Glossary

1. ProNef NFT Marketplace:

- **Definition:** The digital marketplace designed for buying, storing, collecting, exchanging, and earning cryptocurrency through Non-Fungible Tokens (NFTs).

2. NFT (Non-Fungible Token):

- **Definition:** A unique digital asset that represents ownership or proof of authenticity of a specific item or piece of content on the blockchain.

3. React Native:

- **Definition:** A cross-platform framework for building mobile applications using JavaScript and React.

4. Expo:

- **Definition:** A set of tools and services for building React Native applications more efficiently, with a focus on ease of development.

5. Selenium:

- **Definition:** An open-source framework for automating web browsers, used for testing web applications.

6. Appium:

- **Definition:** An open-source automation tool for testing mobile applications on Android and iOS platforms.

7. JMeter:

- **Definition:** An open-source tool for performance testing and load testing of applications.

8. Gatling:

- **Definition:** An open-source load testing framework based on Scala programming language.

9. User Acceptance Testing (UAT):

- **Definition:** The final phase of testing where end-users evaluate the application to ensure it meets their requirements and expectations.

10. Test Case:

- **Definition:** A set of conditions or variables under which a tester will determine if an application, system, or component is working as expected.

11. Regression Testing:

- **Definition:** The process of testing changes to an application to ensure existing functionalities are not affected.

12. Defect:

- **Definition:** A flaw or error in a software application that can lead to deviations from expected behavior.

13. Continuous Integration:

- **Definition:** A development practice where code changes are automatically integrated into a shared repository, allowing early detection of integration issues.

14. Change Management:

- Definition: The systematic approach to managing changes to the project scope, schedule, or resources to ensure successful project delivery.

15. Risk Management:

- **Definition:** The process of identifying, assessing, and mitigating potential risks that could impact the project.

16. Master Test Plan:

- **Definition:** A high-level document that outlines the overall testing strategy and approach for a project.

17. Unit Testing:

- **Definition:** The testing of individual units or components of a software application to ensure they function correctly in isolation.

18. Integration Testing:

- **Definition:** The testing of combined units or systems to verify their interactions and interfaces function as expected.

19. System Testing:

- **Definition:** The testing of a complete and integrated software system to evaluate its compliance with specified requirements.

20. Stakeholders:

- **Definition:** Individuals or groups with an interest or concern in the success of the project, including users, developers, and project managers.