Test Summary Report



**Contents**

[1. Purpose (Damin) 1](#_heading=h.gjdgxs)

[2. Application Overview (Damin) 1](#_heading=h.30j0zll)

[3. Testing Scope (Jill) 1](#_heading=h.1fob9te)

[4. Metrics (Daulet) 2](#_heading=h.3znysh7)

[5. Types of testing performed (Laxman) 4](#_heading=h.2et92p0)

[6. Test Environment & Tools (Shubham) 4](#_heading=h.tyjcwt)

[7. Lessons Learnt (Zaid) 5](#_heading=h.3dy6vkm)

[8. Recommendations (Rishitha) 5](#_heading=h.1t3h5sf)

[9. Best Practices (Rishitha) 5](#_heading=h.4d34og8)

[10. Exit Criteria (Zhuowen) 6](#_heading=h.2s8eyo1)

[11. Conclusion/Sign Off (Sushanth) 6](#_heading=h.17dp8vu)

[12. Definitions, Acronyms, and Abbreviations (Rahul) 6](#_heading=h.3rdcrjn)

# 1. Purpose (Damin)

This document serves as a comprehensive guide outlining the structured activities undertaken during the rigorous testing phase of the Invoicing and Accounts Receivable Modules within the Xero Cloud-Based Accounting System. By meticulously documenting the testing procedures, criteria, and outcomes, it aims to ensure the robustness, functionality, and security of these crucial modules, thereby contributing significantly to the overall quality and usability of the Xero platform. Through a systematic approach to testing, this document strives to identify and rectify any potential issues or discrepancies, ultimately enhancing the user experience and instilling confidence in the reliability of the Xero system.

# 2. Application Overview (Damin)

The Xero accounting system stands as a pinnacle in cloud-based financial management solutions, meticulously crafted to cater to the diverse and evolving needs of modern businesses. With its comprehensive suite of features including invoicing, expense management, payroll, reporting, and purchase orders, Xero offers a holistic solution to streamline and optimize financial operations. Its innovative design and robust architecture facilitate seamless integration with a multitude of third-party applications, empowering users to automate workflows and gain real-time insights into their financial data. Furthermore, the user-friendly interface, accessible from any device with an internet connection, ensures unparalleled accessibility and convenience, enabling businesses of all sizes to effectively manage their finances with ease and efficiency. In essence, Xero redefines the paradigm of financial management, providing businesses with the tools and capabilities they need to thrive in today's dynamic marketplace.

# 3. Testing Scope (Jill)

### a). In Scope

#### Invoicing

* Invoice Management:
  + Create, edit, submit, approve, and delete invoices.
  + Set up recurring invoices for periodic billing.
  + Customize invoice templates to align with branding or preferences.
  + Generate credit notes for adjustments or refunds.
* Customer Management:
  + Manage customer records, including contact details and billing information.
  + Track invoice status and payment activities for each customer.
* Invoice Viewing and Tracking:
  + View draft, pending approval, and paid invoices.
  + Access recurring invoices for ongoing billing arrangements.
* Invoice Automation and Reminders:
  + Configure invoice reminders to prompt customers for payment on overdue invoices.
* Integration and Payments:
  + Integrate with online payment gateways for seamless payment processing directly from invoices.
* Tax Handling:
  + Apply appropriate tax rates and handle tax calculations on invoices.
* Data Import/Export:
  + Import and export invoice data for reporting or record-keeping purposes.

#### Account Receivables:

* Payment and Quotation Management:
  + Set up payment methods and manage customer quotations.
  + Accept and view accepted or invoiced quotations.
  + Convert quotations to invoices and add quotation notes.
* Statement and Aging Reports:
  + Create and filter statements for customer accounts.
  + Generate aging reports to track outstanding invoices and payments.
* Bank Account Reconciliation:
  + Complete bank account reconciliations to ensure accuracy in financial records.
* Customer Management:
  + Edit customer contact and financial details.
  + Set customer credit limits and manage payment terms.

### b) Out of Scope

* Non-Invoicing Related Functionalities:
  + Inventory management features.
  + Payroll processing functionalities.
  + Expense management features not directly related to invoicing.
* Non-Functional Requirements:
  + Testing of performance beyond scalability and responsiveness.
  + Comprehensive security testing beyond basic checks.
  + Usability testing for features not directly related to invoicing or accounts receivables.

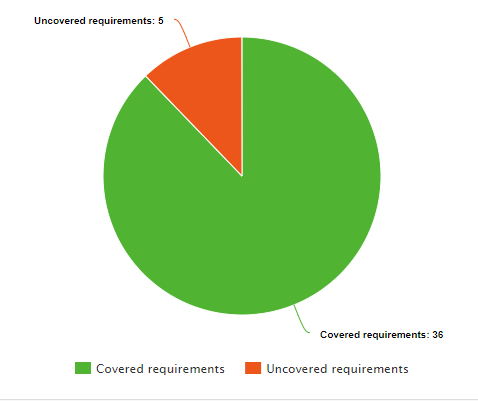
### c) Items not tested

* Third-Party Integrations:
  + Testing of third-party integrations not directly related to invoicing or accounts receivables.
  + Integration testing with non-standard or custom-built systems.
* Advanced Features:
  + Advanced customization options beyond standard templates.
  + Specific niche functionalities or features tailored to specific industries.
* Non-Critical or Rarely Used Features:
  + Features with limited usage or relevance to the majority of users.
  + Optional add-on features that are not commonly enabled or used by clients.

# 4. Metrics (Daulet)

### Test Coverage

| **Manual Test Cases** | **Manual Test Runs** |
| --- | --- |
| 36 | 31 |



### No. of test cases planned vs executed & No. of test cases passed/failed

**Group 1:**

| **Test cases planned** | **Test cases executed** | **TCs Pass** | **TCs Failed** |
| --- | --- | --- | --- |
| 67 | 67 | 62 | 5 |



**Group 2:**

### 

| **Test cases planned** | **Test cases executed** | **TCs Pass** | **TCs Failed** |
| --- | --- | --- | --- |
| 52 | 52 | 47 | 5 |



# 5. Types of testing performed (Laxman)

**1. Functional Testing:**

Functional testing focused on verifying all the features and operations within the Invoices and Account Receivables modules to ensure they function as expected. Tests include the creation, editing, updating, and deletion of invoices to check for correct data handling and outcome. Payment processing capabilities are thoroughly tested for accuracy, including scenarios involving partial and overpayments. Additionally, any automations or integrations with other systems like payment gateways or recurring invoice features are rigorously tested to ensure seamless operation.

**2. Usability Testing:**

Usability testing for the Invoices and Account Receivables modules in the Xero application focuses on ensuring that these financial tools are user-friendly and accessible. This testing evaluates the ease of navigation and operation within the modules, ensuring that features like invoice creation, payment processing, and account management are intuitive and straightforward. The tests also check for the system’s responsiveness on various devices, particularly mobiles, to ensure a consistent experience across platforms.

**3. Performance Testing:**

Performance testing for the Invoices and Account Receivables modules in the Xero application is designed to ensure that these financial systems can handle the required workload efficiently and reliably.This testing assesses how well the system performs under typical and peak load conditions. It includes load testing to observe how the system handles a large number of transactions, stress testing to see how it copes with loads beyond normal capacity, and concurrency testing to ensure it remains stable and responsive when multiple users are accessing the system simultaneously. Performance testing helps identify bottlenecks and optimize performance.

**4. Regression Testing:**

Regression testing was conducted whenever updates or changes are made to the application to ensure that new code additions have not adversely affected existing functionality. This involves testing that confirms recent patches or enhancements have not introduced new faults into the existing system and that previous bugs remain resolved.

# 6. Test Environment & Tools (Shubham)

This section describes the environment and tools employed for testing the Xero accounting software.

Environment: Production

The application was tested within a production-like environment, mimicking real-world user experiences.

**Test Management Tool:**

* JIRA: JIRA served as the central platform for managing the entire test lifecycle. It facilitated the creation, assignment, and tracking of test cases, as well as the analysis of test results, ensuring a well-organized and efficient testing process.

**Defect Logging Tools:**

* JIRA & qTest: Both JIRA and qTest were utilized for documenting and tracking all bugs identified during testing. This approach provided a comprehensive defect logging system for clear communication and efficient resolution of issues.

**Browsers Tested:**

* Chrome (latest version)
* Firefox (latest version)
* Safari (latest version)

Testing across these popular browsers ensured compatibility and functionality of the Xero application for a wide range of users.

# 7. Lessons Learnt (Zaid)

| **S. No** | **Issues faced** | **Solutions** |
| --- | --- | --- |
| 1 | Project milestones are not being met on schedule, impacting overall project timelines. | Set realistic deadlines and conduct regular reviews to track progress. |
| 2 | Few of the team members did not take full responsibility for their tasks, leading to delays and additional workload for others. | Establish clear deadlines. Hold regular check-in meetings where each member updates the team on their progress. Implement a peer accountability system. |
| 3 | RCT not being used effectively to support test case design and review. | Provide training sessions on how to utilize the RCT in daily testing activities. Integrate RCT analysis into regular test planning and review sessions to enhance its utility. |

# 8. Recommendations (Rishitha)

# Enhanced Communication Channels:

# We could have improved communication channels to ensure better coordination among team members, stakeholders, and developers. Utilizing dedicated communication platforms and scheduling regular sync-up meetings could facilitate smoother collaboration.

# More Comprehensive Test Case Coverage:

# In future cycles, we should aim for more comprehensive test case coverage. This includes identifying additional edge cases, unusual scenarios, and negative test cases to ensure robust testing of the application's functionalities.

# Prioritization of High-Risk Areas:

# It would have been beneficial to prioritize testing efforts based on the criticality and risk associated with different modules and functionalities. This would ensure that resources are allocated effectively and potential issues are identified and addressed promptly.

# 

# 9. Best Practices (Rishitha)

# Detailed Test Case Documentation:

# We followed best practices by documenting test cases and their expected outcomes meticulously. This documentation helped guide our testing efforts and served as a reference point for executing test cases systematically.

# Utilization of Testing Tools:

# We utilized Tricentis qTest for test case management and execution, which contributed to efficient testing processes. Leveraging testing tools enabled us to organize test cases effectively, track testing progress, and generate insightful reports.

# Effective Bug Reporting and Tracking:

# We used Jira Atlassian for bug reporting and tracking, ensuring that all identified issues were logged systematically. Providing detailed descriptions, steps to reproduce, and screenshots for each reported bug facilitated faster resolution and communication with the development team.

# 10. Exit Criteria (Zhuowen)

* All test cases have been executed. – Yes
* All test cases have passed or have acceptable justifications for failures. – Yes
* All defects have been reported and verified. – Yes
* All defects of Critical, Major, and Medium severity are either resolved or will make an acceptable mitigation plan. – Yes

# 11. Conclusion/Sign Off (Sushanth)

In light of the successful completion of the exit criteria detailed in Section 10 of this Test Summary Report, the Testing team affirms that the application is ready for deployment. All stipulated testing criteria have been satisfactorily fulfilled, affirming that the Invoicing and Account Receivables modules within the Xero application have undergone comprehensive testing and validation. Robust User Acceptance Testing and Business acceptance testing is also essential for mitigating potential risks and ensuring seamless alignment with stakeholder needs.

# 12. Definitions, Acronyms, and Abbreviations (Rahul)

* **qTest**: qTest is a comprehensive test management tool developed by QASymphony. It provides a centralized platform for creating, organizing, and executing test cases. Key features include test case repositories, test planning and scheduling, integration with defect tracking tools, customizable reporting and metrics, and support for various testing methodologies like agile and waterfall. qTest aims to streamline the testing process, improve traceability, and enhance collaboration between testers, developers, and stakeholders.
* **JIRA**: JIRA, developed by Atlassian, is a widely adopted issue and project tracking tool. It offers a flexible workflow system that can be customized to match an organization's development processes. JIRA supports agile methodologies like Scrum and Kanban, with features like backlogs, sprint planning, and burndown charts. It allows teams to create and manage issues (tasks, bugs, feature requests), assign them to team members, track progress, and generate reports. JIRA integrates with various development tools, facilitating seamless collaboration across teams.
* **UAT (User Acceptance Testing)**: UAT is a crucial phase in software testing where the final product is evaluated by actual end-users or client representatives. The primary goal is to validate that the application meets the specified business requirements, user expectations, and real-world scenarios. UAT typically occurs after functional and non-functional testing phases and involves testing the system with real-world data and use cases. Successful UAT is often a prerequisite for product release or deployment.
* **URL (Uniform Resource Locator)**: A URL is a unique address that specifies the location of a resource (website, web page, file, or other content) on the internet. It consists of several components, including the protocol (e.g., http, https), domain name (e.g., www.example.com), and path to the specific resource. URLs are used by web browsers and other internet applications to locate and retrieve the requested content from web servers.
* **Defect**: In software testing, a defect is an error, flaw, or deviation from the expected behavior or requirements of the software application. Defects can range from functional issues (incorrect calculations, missing features) to non-functional issues (performance problems, security vulnerabilities). Defects are typically reported, tracked, prioritized, and assigned for resolution during the testing and development process.
* **Action Plan**: An action plan is a detailed strategy that outlines the steps, tasks, and responsibilities required to address and resolve identified defects or issues. It typically includes information such as the root cause analysis, proposed solutions or fixes, timelines, assigned resources, and any dependencies or risks associated with the resolution process. Action plans help ensure that defects are systematically addressed and tracked until resolution, preventing them from being overlooked or forgotten.
* **Test Environment**: A test environment is a controlled setup that replicates the production environment (or a subset of it) where the software application will be deployed. It includes the necessary hardware, software, databases, network configurations, and any external systems or dependencies required for comprehensive testing. Test environments are isolated from production environments to prevent any unintended impacts during testing activities, ensuring the safety and stability of live systems.
* **Exit Criteria:** Exit criteria are predefined sets of conditions or requirements that must be met before a process, phase, or activity can be considered complete and officially concluded. Exit criteria serve as objective checkpoints or gates that must be passed before moving forward, ensuring that the necessary quality standards, requirements, and processes have been followed and met. They help maintain control over the project or process, mitigate risks, and ensure that stakeholders are aligned and satisfied with the outcomes before proceeding.