**Test plan for Accounting system web application**

Table of Contents

[1 Introduction](file:///C:\Users\dsingla\Downloads\Test%20plan%20&amp;%20strategy%20for%20Network%20Monitoring%20system%201.docx#_Toc515524393)

[1.1 Purpose](file:///C:\Users\dsingla\Downloads\Test%20plan%20&amp;%20strategy%20for%20Network%20Monitoring%20system%201.docx#_Toc515524394)

[1.2 Project Overview](file:///C:\Users\dsingla\Downloads\Test%20plan%20&amp;%20strategy%20for%20Network%20Monitoring%20system%201.docx#_Toc515524395)

[2 Scope](file:///C:\Users\dsingla\Downloads\Test%20plan%20&amp;%20strategy%20for%20Network%20Monitoring%20system%201.docx#_Toc515524396)

[2.1 In-Scope](file:///C:\Users\dsingla\Downloads\Test%20plan%20&amp;%20strategy%20for%20Network%20Monitoring%20system%201.docx#_Toc515524397)

[2.2 Out-of-Scope](file:///C:\Users\dsingla\Downloads\Test%20plan%20&amp;%20strategy%20for%20Network%20Monitoring%20system%201.docx#_Toc515524398)

[3 Testing Strategy](file:///C:\Users\dsingla\Downloads\Test%20plan%20&amp;%20strategy%20for%20Network%20Monitoring%20system%201.docx#_Toc515524399)

[3.1 Test Objectives](file:///C:\Users\dsingla\Downloads\Test%20plan%20&amp;%20strategy%20for%20Network%20Monitoring%20system%201.docx#_Toc515524400)

[3.2 Data Approach](file:///C:\Users\dsingla\Downloads\Test%20plan%20&amp;%20strategy%20for%20Network%20Monitoring%20system%201.docx#_Toc515524402)

[3.3 Level of Testing](file:///C:\Users\dsingla\Downloads\Test%20plan%20&amp;%20strategy%20for%20Network%20Monitoring%20system%201.docx#_Toc515524403)

[3.4 Functional Testing](file:///C:\Users\dsingla\Downloads\Test%20plan%20&amp;%20strategy%20for%20Network%20Monitoring%20system%201.docx#_Toc515524405)

[4 Execution Strategy](file:///C:\Users\dsingla\Downloads\Test%20plan%20&amp;%20strategy%20for%20Network%20Monitoring%20system%201.docx#_Toc515524408)

[4.1 Entry Criteria](file:///C:\Users\dsingla\Downloads\Test%20plan%20&amp;%20strategy%20for%20Network%20Monitoring%20system%201.docx#_Toc515524409)

[4.2 Exit criteria](file:///C:\Users\dsingla\Downloads\Test%20plan%20&amp;%20strategy%20for%20Network%20Monitoring%20system%201.docx#_Toc515524410)

[4.3 Validation and Defect Management](file:///C:\Users\dsingla\Downloads\Test%20plan%20&amp;%20strategy%20for%20Network%20Monitoring%20system%201.docx#_Toc515524411)

[5 Environment Requirements](file:///C:\Users\dsingla\Downloads\Test%20plan%20&amp;%20strategy%20for%20Network%20Monitoring%20system%201.docx#_Toc515524412)

[5.1 Test Environments](file:///C:\Users\dsingla\Downloads\Test%20plan%20&amp;%20strategy%20for%20Network%20Monitoring%20system%201.docx#_Toc515524413)

## 

1. **Introduction**
   1. **Purpose**

The purpose of a test plan for The accounting software is to test the application to make sure that every functionality is working properly and as expected.

**1.2 Project Overview**

The accounting software is a comprehensive solution designed to manage financial transactions, reporting, and analysis for businesses of varying sizes. It provides functionalities such as general ledger management, transaction processing, invoice and billing, finance reporting etc.

1. **Scope**

**2.1 In-scope**

Features to be tested.

* Login page and signup
* UI
* User interface
* Login functionality
* Dashboard analysis when logged in as Finance manager
* Report section
* Functional testing of reports
* Dashboard analysis when logged in as Accountant
* Chart of accounts
* Real time update when any entry is modified
* Functionality
* Dashboard analysis when logged in as Bookkeeper
* Transaction module
* Validation of transaction entry
* Functionality
* Dashboard analysis when logged in as Account payable/receivable clerk
* Invoices module
* Validation of invoice uploaded
* Payment batches generation validation
* Payment status tracking validation
* Dashboard analysis for Financial analyst
* Ad-hoc reporting module
* Functionality
* Real time data refresh validation

**2.2 Out-of-Scope**

**3.** **Testing Strategy**

**3.1 Test Objectives**

* Performance Monitoring: Validate whether all functions are working as expected
* Security Testing: Identify and mitigate security vulnerabilities to protect financial data from unauthorized access, manipulation, or breaches.
* Availability Monitoring: Ensure network components are up and running, minimizing downtime.
* Usability Testing: Evaluate the user-friendliness of the accounting system interface and workflows.

**3.2 Data Approach:**

* Valid test data
* Invalid test data
* New data entry
* Validation of data entry and accuracy
* Real time update of data over the application

**3.3 Levels of testing:**

* **Smoke Testing:** It is used to Determine if a new software build is ready for the next testing phase Smoke tests verify whether the most important features work as expected and that there are no showstopper issues in the build that can potentially lead to blocking the entire testing team. It helps in deciding if the build is flawed or not and hence, prevents the entire team from wasting time or resources.
* **Functional Testing**: In this testing tester tests whether application is working as expected or not as per functionalities defined.
* **Non-Functional Testing:** This is type of testing where software UI is tested to check whether UI is as per defined condition or not.
* **Regression Testing**: Type of testing to test whether application is working fine or not due to changes in code.

## 3.4 Functional Testing

Features to be tested.

* Login functionality
* Report section
* Chart of accounts
* Transaction module
* Invoices module
* Ad-hoc reporting module

**4 Execution strategy**

**4.1 Entry criteria**

* Availability of complete or partially testable code
* Appropriately defined and approved requirements
* Code has been merged successfully
* The readiness of test cases
* Testing is done, defects raised are fixed and retested

**4.2 Exit criteria**

* *100% Test Scripts executed*
* *No open Critical and High severity defects*
* *All remaining defects are either cancelled or documented as Change Requests for a future release*
* *All expected and actual results are captured and documented with the test script*
* *All defects logged in*
* Once defects are fixed, regression testing is done to make sure application is working properly after code change

**4.3 Validation and Defect Management**

Whenever defect is observed, ticket should be created on Jira

Defect severity levels: Critical, Major, Minor.

Defect priorities: High, Medium, Low.

* Critical defect are those due to which functionality is blocking and testing can not be proceeded further.
* Major defects are those due to which specific functionality is blocking but testing can be continued.
* Minor defects are defects like font size etc.

# **5. Environment Requirements:**

Test Environments:

Hardware, Software, and Network infrastructure required for testing.

Ensure that the test environment is isolated from production systems to avoid interference with live operations.