Your Company Name

System Test Plan

Date

Revision History

Date	Version	Author	Change

Table of Contents

Note: Text displayed in blue italics is included to provide guidance to the author and should be deleted before publishing the document. In any table, select and delete any blue line text; then click Home→Styles and select "Table Text" to restore the cells to the default value

1 Purpose

The purpose of the System Test Plan is to obtain all required information outlined in the requirements, specifications, and design documentation to plan and implement tests that ensure a high level of assurance. The plan and tests will provide information and guidance to management and staff that the application works as expected. It supports the following goals and objectives:

- Functions, features, and items to be tested.
- The testing approach.
- Resources to be used and estimated testing time.
- The system performs correctly as planned without error.

1.1 Reference Documents

The System Test Plan contains information based on the following documents.

Document	Date or Version Number
Requirements Document	
Design Document	
Specifications Document	

2 System Test Description

This section provides the goals and objectives of system testing, entrance and exit criteria, and what will be tested.

2.1 Test Goals and Objectives

System testing provides confidence that the application functions and features specified in the referenced documentation performed as expected and meet the following criteria:

- Functions, features, and items tested perform without error.
- Data integrity is maintained.
- Data access and security levels are enforced.
- System functions and features respond correctly to error situations.
- Compliance to legal and regulatory requirements is maintained.

2.2 Test Entrance and Exit Criteria

2.2.1 Entrance Criteria

Define the test situations or criteria that must be in place before a specific test is performed, e.g.,

- ***Requirements, design, and specifications have been approved.***
- ***System Test Plan has been approved.***
- ***Unit testing is complete.***
- ***System test scripts have been written and reviewed before execution.***

2.2.2 Exit Criteria

Define what needs to be performed (i.e., criteria) to consider the testing finished, e.g.,

- ***All test scripts have been executed, reviewed, and approved.***
- ***All incidents identified during system testing were logged and resolved.***
- ***Regression testing was performed for all resolved incidents.***

2.3 Test Deliverables

System test deliverables include the following documents:

- ***The System Test Plan.***
- ***Test scripts with supporting documentation (e.g., tester, test dates, results, incidents, logs.)***
- ***System Test Traceability Matrix.***
- ***The System Test Summary Report.***

3 System Test Approach

This section provides information about the system testing environment that will be used, which should replicate the production environment. System testing includes test scripts based on the requirements, design, and specifications documentation.

3.1 Scope of System Testing

Provide information about what is and what is not part of system testing.

3.1.1 Test Categories

List the tests that will be performed as part of system testing.
Note: Add categories not listed and remove categories that will not be tested.

Category	Description
Functionality	*Functions as described in documentation.*
Security and Access Control	*Provides the proper application and user level security.* • *Application-level security, including access to the Data or Business Functions* • *System-level security, including logging into or remote access to the system.*
Data, Database, and Data Integration	*Ensure data accessed, used, and applied is valid and correct. Test databases and the database processes as a subsystem.*
Boundaries	*Fields perform in accordance with the constraints placed on the fields.*
Audit Trail	*Can track user and system activity (adds, changes, and deletes).*
Error Conditions	*Provides specific confirmation and error messages.*
Performance	*Provides or meets required performance guidelines as described in documentation.*
External Interfaces	*Functions timely and correctly with other external systems.*
User Interface	*User interfaces functions as described in documentation.*
Reporting	*Prints or displays report data as described in documentation.*

3.2 Risk Assessment

Provide any risks associated with the test plan, including any work-around alternatives for the risks indicated.

4 Test Environment

4.1 Hardware

Provide a description of the hardware that will be used in the system testing.

4.2 Software

Provide a description of software and applications that will be used in the system testing.

4.3 Tools

Provide a description of the testing tools that will be used in the system testing.

5 Test Plan Schedule

Task Description	Duration	Start Date	End Date

Task Description	Duration	Start Date	End Date

6 Testing Matrix

Provide a separate section for each type of testing that is performed that includes the following information:

- ***Assumptions, Pre-conditions, and Risks***
- ***Test instructions with Entrance and Exit Criteria***
- ***Defect Metrics.***

6.1 Assumptions, Pre-Conditions, Risks

Assumptions:	
Pre-conditions	
Risks:	

6.2 Test Instructions

Entrance Criteria:	
Exit Criteria:	

Step	Test Instructions	Expected Result	Pass / Fail	Comments
1.				
2.				
3.				
4.				

6.3 Test Completion Summary

Test #	Tester	Date	Description	Pass/ Fail
1.				
2.				
3.				
All Tests				

6.4 Associated Defects

Total number of defects opened during testing.	
Number of defects fixed during testing.	
Number of defects to be fixed after system implementation.	
Number of other defects that will not be fixed or will be dropped.	

Note:
Management can track incidents and defects using information in the following tables:

Priority

Priority	Definition
Critical	***Required function is not working and there is no workaround. Unable to continue testing with the current functionality.***
High	***Required function is not working, but there is a workaround. Testing can continue in other areas.***
Medium	***Functionality achieves the intent, but not to the letter of the requirement and/or design.***
Low	***Functionality achieves the intent but includes inconveniences or annoyances.***

Status

Status	Definition
New	***Incident has just been entered.***
Open	***Incident is being reviewed.***
In-Progress	***Incident has been assigned for correction.***
Migrated	***Incident has been coded for correction and is ready for re-test.***
Deferred	***Incident will be addressed in another software version.***
Cancelled	***Incident has been found to be a non-issue.***
Duplicate	***Incident is a duplicate of another incident.***
Closed	***Incident has been re-tested and correction has been confirmed.***

7 Glossary

Glossary Topic	Description
Ad-hoc Testing	This type of testing is done without any formal Test Plan or Test Case creation. Ad-hoc testing helps in deciding the scope and duration of other testing and it also helps testers in learning the application prior to starting with any other testing.
Branch Coverage	No software application can be written in a continuous mode of coding, at some point we need to branch out the code to perform a particular functionality. Branch coverage testing helps in validating all the branches in the code and making sure that no branching leads to abnormal behavior of the application.
Enhancement	The addition of new functionality and/or the change / removal of existing functionality to improve the application.
Exploratory Testing	This testing is similar to the ad-hoc testing and is done to learn / explore the application.
Functional Testing	In this type of testing, the software is tested for functional requirements. The tests are written to check if the application behaves as expected.
Incident	A problem with the application, test script, tester performance and/or documentation, which prevents verification of a particular requirement or design element.
Incident Tracking Tool	The tool is used to document and track all incidents discovered during system test.
Load Testing	The application is tested against heavy loads or inputs such as testing of web sites to find out at what point the web-site / application fails or at what point its performance degrades.
Mutation Testing	A kind of testing in which, the application is tested for code that was modified after fixing a particular bug / defect. It also helps in finding out which code and which strategy of coding can help in developing the functionality effectively.
Recovery Testing	Recovery testing is basically done to check how fast and to ensure the application can recover against any type of crash or hardware failure, etc. Type or extent of recovery is specified in the requirement specifications.
Regression Testing	The process of re-executing one or more test scripts to verify that errors have been properly corrected and that no new errors have been introduced.

Glossary Topic	Description
Security Testing	Security Testing is carried out to find out how well the system can protect itself from unauthorized access, hacking – cracking, any code damage, etc. It deals with the application code. This type of testing needs sophisticated testing techniques.
Smoke Testing	This type of testing is also called sanity testing and is done to check if the application is ready for further major testing and is working properly without failing up to least expected level.
Statement Coverage	In this type of testing the code is executed in such a manner that every statement of the application is executed at least once. It helps in assuring that all the statements execute without any side effect.
Static and dynamic Analysis	Static analysis involves going through the code to find out any possible defect in the code. Dynamic analysis involves executing the code and analyzing the output.
Stress Testing	The application is tested against a heavy load, such as complex numerical values, large number of inputs, large number of queries, etc. It checks for the stress / loads the applications can withstand.
Test Case	A planned sequence of events designed to verify one or more requirements or design elements.
Test Script	A document containing a series of step by step instructions that are followed by a tester to verify functionality associated with one or more requirements or design elements. Verification is performed by comparing the documented expected results to the actual system results.
Unit Testing	The developer carries out unit testing to check if the particular module or unit of code is working fine. Unit Testing comes at the very basic level as it is carried out when the unit of the code is developed or a particular functionality is built.
Usability Testing	This testing is also called 'Testing for User-Friendliness'. This testing is done if User Interface of the application stands an important consideration and needs to be specific for a unique type of user.
Volume Testing	Volume testing is done against the efficiency of the application. A high volume of data is processed through the application (which is being tested) to check the extreme limitations of the system.

8 APPENDIX