Your Company Name

User Acceptance Test Plan (UAT)

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 User Acceptance Test (UAT) Plan is to provide management an overview of the system, applications, functions, and features that are to be tested in the UAT process. Detailed information is outlined in the requirements, specifications, and design documentation. The plan and tests will provide information and guidance to management, staff, and the user community that the application works as expected and ensures a high level of confidence to implement.

It supports the following goals and objectives, which will help to verify the following:

- o   Functions, features, and items to be tested.
- o   The testing approach.
- o   Resources to be used and estimated testing time.
- o   The system can be used to perform the required business functions and processes under conditions that closely mirror the production environment.
- o   The system performs correctly as planned without error.
- o   System performance is acceptable.
- o   All requirements have been met through traceability from the documented requirements to the UAT scripts.

1.1   Background

Testers have been using test cases since the inception of computer programming over 50 years ago. The difficult part of creating test cases is determining which processing events should be made into test cases. Experience has shown that it is uneconomical to test all cases in an application system. The Return On Investment (ROI) is not worth the effort. Experience further shows that most testing exercises less than one-half of the total of all computer instructions. Therefore, it is mandatory that selecting the most important processing events is the key ingredient in building test cases.

1.1.1   Building Test Cases

The recommended process for the creation and use of test cases should follow the guidelines below:

Identify Test Resources	Testing use cases can be as extensive or limited a process normally dictated by time and budget constraints. Unfortunately, many testers approach the creation of test cases under duress and attempt to "catch" the most critical processing steps. Where their allotted time has expired, testing somehow is complete. One must bear in mind the actual time that has been allocated to conduct testing and then a process developed that optimizes that time.
Identify Conditions	Development of a testing matrix is recommended as the basis for identifying

to be Tested	conditions to test, whereby all possible test conditions are identified.

Rank Test Conditions	If resources are limited (the normal state of the IT environment), the best use of resources will be obtained by logically testing the most important test conditions. The objective of ranking is to identify high-priority test conditions that must be tested first. However, note that ranking does not mean that low-ranked test conditions WILL NOT BE TESTED.
	
	Ranking can be used for two purposes:
	  1. To determine which conditions should be tested first.
	  2. To determine the amount of resources allocated to each of the test conditions.
	
	For example, in testing a payroll application, withholding for a minor tax may only be tested once, while federal tax deductions may be tested 5 or 6 times.

Select Conditions for Testing	Based on the above ranking, the conditions to be tested should be selected. Each test situation should be documented in a testing matrix that was hopefully started during the Requirements Definition Phase.
Determine Correct Results of Processing	Accurate processing results for each situation should be carefully determined. A unique identifier will be assigned to each test case. The correct time to determine the correct processing results are before the test transactions have been created. This step helps determine the reasonableness and usefulness of test transactions. This process can also show if there are ways to extend the effectiveness of test transactions, and whether the same condition has been tested by another transaction.
Create Test Cases	Each test situation needs to be converted into a format suitable for testing, depending on whether you will be using key entry, a test data generator or the preparation of an input form which will be given to personnel to conduct testing.
Document Test Conditions	The test case and the results are documented within this document.
Conduct Tests	The application should be run using the test conditions. Depending on the extent of the test, it can be run under a test condition or in a pseudo production environment.
Verify and Correct	The results of testing should be verified and any necessary bugs indentified and documented; then corrections to the programs accomplished and the test case re-executed. Problems detected as a result of testing can be attributable to not only application defects, but to the possibility of test data defects as well. The individual conducting the tests should be aware of both possibilities

1.2 Reference Documents

The UAT Plan contains information based on the following documents.

Document	Date or Version Number
Requirements Document	
Design Document	
Specifications Document	

2   User Acceptance Test Description

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

2.1   Test Goals and Objectives

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

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

2.2   Test Entrance and Exit Criteria

2.3   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.*
- *UAT Plan has been approved.*
- *Unit testing and system testing have been completed.*
- *UAT test scripts have been written and reviewed before execution.*

2.4   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 UAT testing were logged and resolved.*
- *Regression testing was performed for all resolved incidents.*

2.5   Test Deliverables

UAT test deliverables include the following documents:
- *The User Acceptance Test Plan.*
- *Test scripts with supporting documentation (e.g., tester, test dates, results, incidents, logs.)*
- *UAT Traceability Matrix (if applicable).*
- *The UAT Summary Report.*

3 UAT Test Approach

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

3.1 Scope of UAT Testing

Summarize and list the functions, features, and applications to be tested. The items listed here should directly relate to the main features / major business functions discussed in the requirements and design (if applicable) documentation.

3.2 Test Categories

List the tests that will be performed as part of user acceptance testing.

Note: This table is an example, 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 interface functions as described in documentation.*
Reporting	*Prints or displays report data as described in documentation.*

3.3 Risks, Dependencies, Assumptions and Constraints

Describe any risks, dependencies, assumptions, and constraints that would affect user acceptance testing and implementation. Provide any work-around solutions that may apply.

4 Functional Testing

Specify major activities, techniques, and tools to be used to test the functions, features, and applications. Provide information about the major testing tasks and approximate time to run each one.

4.1 Functionality Included

Provide a high level outline of the major testing functions planned for the UAT testing.

4.2 Functionality Excluded

Provide a high level outline of the tests that have been specifically excluded from UAT testing.

5 Test Environment

5.1 Hardware

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

5.2 Software

Provide a description of software and applications that will be used in user acceptance testing.

5.3 Tools

Provide a description of the testing tools (if any) that will be used in user acceptance testing.

6 Test Plan Schedule

Complete the table with the appropriate test schedule information.

Task Description	# of Days	Start Date	End Date

6.1 Roles and Responsibilities

Complete the table with the appropriate organizational role and responsibility information.

Role	Responsibility Description	Assigned To
Project / Product Manager	*Plan and/or assist with user acceptance testing.* *Prepare Product Acceptance Plan.*	
Test Manager	*Responsible for System Testing.* *Identifies resources to assist in test case and test script selection and preparation.*	
Business / Systems Analyst	*Perform analysis and prepare requirements documentation.* *Provide an overview and/or demonstrate the application to business and support users.* *Assist staff and UAT Testers with resolutions to problems and implement required system related modifications.* *Prepare UAT Plan and requirements as appropriate.* *Finalize and publish test script(s).* *Train testers on the UAT execution, problem reporting, and resolution processes.*	

Role	Responsibility Description	Assigned To
	Coordinate and assist in execution of UAT scripts.	
User Acceptance Tester(s) (Business Users)	*Execute the UAT script(s) and assist in the preparation of discrepancy reports or problem logs.* *Log test results.* *Submits UAT defects discovered.* *Validate accuracy of User Guide & Business Procedures.* *Participate in resolution of problems.*	
Database Administrator	*Ensure test data is managed and maintained.* *Perform data restoration to refresh data before UAT.*	

7 Testing Matrix

Provide a separate section for each series of testing (from test plan schedule) that is performed for the functions, features, or applications that include the following information:

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

7.1 Assumptions, Pre-Conditions, Risks

Assumptions:	
Pre-conditions	
Risks:	

7.2 Test Instructions

Entrance Criteria:	
Exit Criteria:	

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

7.3 Test Completion Summary

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

7.4   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	Definitions
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.

8   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 the other testing and it also helps testers in learning the application prior to starting with any other testing.
Enhancement	New functionality and/or the change or removal of existing functionality to improve the application.
Functional Testing	The software is tested for the functional requirements. The tests are written to check if the application behaves as expected, i.e., verifies that the application fulfills the requirements as documented in requirements documentation.
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 used to document and track all incidents discovered during testing.
Recovery Testing	Recovery testing is performed to check how fast the application can recover against any type of crash or hardware failure.
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.
Security Testing	Security Testing is performed to find out how well the system can protect itself from unauthorized access, e.g., hacking / cracking any code damage, etc. which deals with the application code. This type of testing needs sophisticated testing techniques.
Stress Testing	The application is tested against a heavy load such as complex numerical values, large number of inputs, large number of queries, etc. that checks for the stress or load 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.

Topic	Description
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 as 'Testing for User-Friendliness'. This testing is done if the application User Interface is 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 huge amount of data is processed through the application (which is being tested) to check the extreme limitations of the system.

9 Appendix