online convention hall booking Test Plan
for the
Architectural Prototype

Version 1.0

Revision History

Date	Version	Description	Author
7/March/1999	1.0	Initial Release – Prototype Test Plan	Mohammed dawood

Table of Contents

Test Plan

for the

Architectural Prototype

1. Objectives

1.1 Purpose

This document describes the plan for testing the architectural prototype of the **online convention hall booking**. This Test Plan document supports the following objectives:

- Identify existing project information and the software that should be tested.
- List the recommended test requirements (high level).
- Recommend and describe the testing strategies to be employed.
- Identify the required resources and provide an estimate of the test efforts.

- List the deliverable elements of the test activities.

1.2 Scope

This Test Plan describes the integration and system tests that will be conducted on the architectural prototype following integration of the subsystems

It is assumed that unit testing already provided thorough black box testing, extensive coverage of source code, and testing of all module interfaces.

The interfaces between the following subsystems will be tested:

1. Login page
2. Search option
3. Book hall.

The external interfaces to the following devices will be tested:

1. Local PCs
2. Remote PCs.

The most critical performance measures to test are:

1. Response time for login to online convention hall booking.
2. Response time to search the halls.
3. Response time to book the hall.
4. response time when system loaded with 200 logged users.

2. Requirements for Test

2.1 Data and Database Integrity Testing

Verify access to booking hall Database.

Verify simultaneous record read accesses.

Verify lockout during booking hall details updates.

Verify correct retrieval of update of database data.

2.2 User Interface Testing

Verify ease of navigation through a sample set of screens.

Verify sample screens conform to GUI standards.

2.3 Performance Testing

Verify response time to access online booking hall application.

Verify response time to access booking hall subsystem.

Verify response time for login.

.

2.4 Load Testing

Verify system response when loaded with 200 logged on users.

Verify system response when 50 simultaneous users accesses to the online booking hall application.

2.5 Security and Access Control Testing

Verify Logon from a local PC.

Verify Logon from a remote PC.

Verify Logon security through user name and password mechanisms.

3. Test Strategy

The main considerations for the test strategy are the techniques to be used and the criterion for knowing when the testing is completed.

In addition to the considerations provided for each test below, testing should only be executed using known, controlled databases, in secured environments.

.

3.1 Testing Types

3.1.1 Function Testing

Testing of the application should focus on any target requirements that can be traced directly to use cases (or business functions), and business rules. The goals of these tests are to verify proper data acceptance, processing, and retrieval, and the appropriate implementation of the business rules. This type of testing is based upon black box techniques, that is, verifying the application (and its internal processes) by interacting with the application via the GUI and analysing the output (results). Identified below is an outline of the testing recommended for each application:

Test Objective:	Ensure proper application navigation, data entry, processing, and retrieval.
Technique:	• Execute each use case, use case flow, or function, using valid and invalid data, to verify the following: • The expected results occur when valid data is used. • The appropriate error / warning messages are displayed when invalid data is used. • Each business rule is properly applied.
Completion Criteria:	• All planned tests have been executed.

- All identified defects have been addressed.

3.1.2 User Interface Testing

User Interface testing verifies a user’s interaction with the software. The goal of UI Testing is to ensure that the User Interface provides the user with the appropriate access and navigation through the functions of the applications. In addition, UI Testing ensures that the objects within the UI function as expected and conform to corporate or industry standards.

Test Objective: Verify the following:

- Navigation through the application properly reflects business functions and requirements, including window to window, field to field, and use of access methods (tab keys, mouse movements, accelerator keys)
- Window objects and characteristics, such as menus, size, position, state, and focus conform to standards.

Technique:

- Create / modify tests for each window to verify proper navigation and object states for each application window and objects.

Completion Criteria: Each window successfully verified to remain consistent with benchmark version or within acceptable standard

Special Considerations:

- Not all properties for custom and third party objects can be accessed.

3.2 Tools

The following tools will be employed for testing of the architectural prototype:

	Tool	**Version**
Test Management	selenium	TBD
Test Design	selenium	TBD
Defect Tracking	selenium	TBD
Functional Testing	selenium	TBD
Performance Testing	selenium	TBD

4. Resources

This section presents the recommended resources for testing the online hall booking application architectural prototype, their main responsibilities, and their knowledge or skill set.

4.1 Roles

This table shows the staffing assumptions for the test of the Prototype.

Human Resources		
Role	**Minimum Resources Recommended** (number of workers allocated full-time)	**Specific Responsibilities/Comments**
Test Manager	1 – Kerry Stone	Provides management oversight Responsibilities: • Provide technical direction • Acquire appropriate resources • Management reporting
Test Designer	Margaret Cox Carol Smith	Identifies, prioritizes, and implements test cases Responsibilities: • Generate test plan • Generate Test Suite • Evaluate effectiveness of test effort
System Tester	Carol Smith	Executes the tests Responsibilities: • Execute tests • Log results • Recover from errors • Document defects
Test System Administrator	Simon Jones	Ensures test environment and assets are managed and maintained. Responsibilities: • Administer test management system

		• Install / manage worker access to test systems
Database Administration / Database Manager	Margaret Cox	Ensures test data (database) environment and assets are managed and maintained. Responsibilities: • Administer test data (database)
Designer	Margaret Cox	Identifies and defines the operations, attributes, and associations of the test classes Responsibilities: • Identifies and defines the test class(es) • Identifies and defines the test packages
Implementer	Margaret Cox	Implements and unit tests the test classes and test packages Responsibilities: • Creates the test classes and packages implemented in the Test Suite.

5. Project Milestones

Testing of the C-Registration Architectural Prototype incorporates test activities for each of the test efforts identified in the previous sections. Separate project milestones are identified to communicate project status and accomplishments.

Refer to the Software Development Plan [13] and the E1 Iteration Plan [14] for the overall phase or master project schedule.

Milestone Task	**Effort (pd)**	**Start Date**	**End Date**
Prototype Test Planning	2	March 12	March 15
Prototype Test Design	3	March 15	March 18
Prototype Test Development	4	March 19	March 23
Prototype Test Execution	3	March 24	March 26
Prototype Test Evaluation	1	March 29	March 29

6. Deliverables

The deliverables of the test activities as defined in this Test Plan are outlined in the table below.

Deliverable	Owner	Review / Distribution	Due Date
Test Plan	K. Stone	Senior Project Mgmt Team	March 15
Test Environment	S. Jones	-	March 18
Test Suite	C. Smith and M. Cox	Internal Peer Review	March 23
Test Data Sets	M. Cox	Internal Peer Review	March 23
Test Scripts	M. Cox	Internal Peer Review	March 23
Test Scripts	M. Cox	-	March 23
Test Stubs, Drivers	M. Cox	-	March 23
Test Defect Reports	C. Smith	Senior Project Mgmt Team	March 26
Test Results	C. Smith	-	March 26
Test Evaluation Report	C. Smith	Senior Project Mgmt Team	March 29

6.1 Test Suite

The Test Suite will define all the test cases and the test scripts which are associated with each test case.

6.2 Test Logs

It is planned to use RequisitePro to identify the test cases and to track the status of each test case. The test results will be summarized in RequisitePro as untested, passed, conditional pass, or failed. In summary, RequisitePro will be setup to support the following attributes for each test case, as defined in the Requirements Attributes Guidelines [17]:

- Test status
- Build Number
- Tested By
- Date Tested
- Test Notes

It will be the responsibility of the System Tester to update the test status in RequisitePro.

Test results will be retained under Configuration Control.

6.3 Defect Reports

Rational ClearQuest will be used for logging and tracking individual defects.

7. Project Tasks

Below are the test related tasks for testing the C-Registration Architectural Prototype:

Plan Test

Identify Requirements for Test

Assess Risk

Develop Test Strategy

Identify Test Resources

Create Schedule

Generate Test Plan

Design Test

Workload Analysis (not applicable for Prototype)

Develop Test Suite

Identify and Describe Test Cases

Identify and Structure Test Scripts

Review and Access Test Coverage

Implement Test

Setup Test Environment

Record or Program Test Scripts

Develop Test Stubs and Drivers

Identify Test-Specific functionality in the design and implementation model

Establish External Data sets

Execute Test

Execute Test Scripts

Evaluate Execution of Test

Recover from Halted Test

Verify the results

Investigate Unexpected Results

Log Defects

Evaluate Test

Evaluate Test-Case Coverage

Evaluate Code Coverage

Analyze Defects

Determine if Test Completion Criteria and Success Criteria have been achieved

Create Test Evaluation Report