**Test Strategy**

**Table of Contents:**

Contents

[Abstract: 1](#_Toc16605708)

[Test Approach: 2](#_Toc16605709)

[Test Actors: 2](#_Toc16605710)

[Types of Testing: 3](#_Toc16605711)

[Unit Testing 3](#_Toc16605712)

[Feature Testing 3](#_Toc16605713)

[Functional Testing 3](#_Toc16605714)

[Integration Testing 3](#_Toc16605715)

[User Acceptance Testing 3](#_Toc16605716)

[Regression Testing 4](#_Toc16605717)

[Performance Testing 4](#_Toc16605718)

[Production Validation 4](#_Toc16605719)

[Training Validation 4](#_Toc16605720)

[Testing Environments: 5](#_Toc16605721)

[Test Automation and Testing Tools 5](#_Toc16605722)

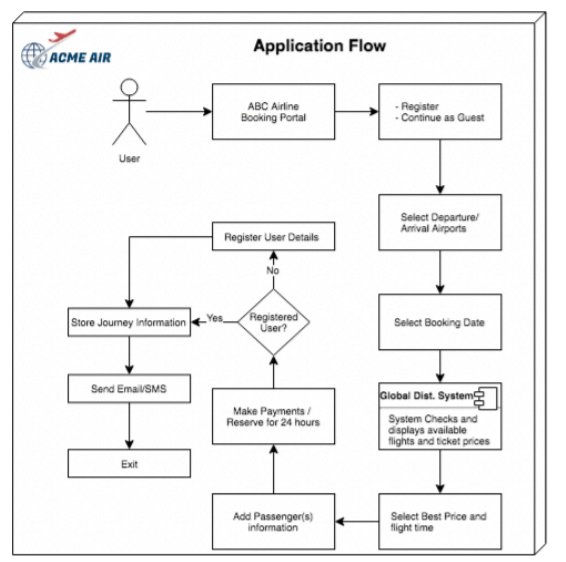
[Risk Analysis 5](#_Toc16605723)

[Test Planning and Execution 5](#_Toc16605724)

[Review and Approval 6](#_Toc16605725)

# Abstract:

Acme Airlines is planning to launch a new web portal for flight bookings. This portal will be a responsive web application. It will allow users to book domestic and international flights as a guest as well as allow users to register and store user information for ease of booking in future. This application will require using Galileo Global distribution System to manage bookings. Galileo doesn't provide a Sandbox environment and only have a production environment to work with. CBA is partnered to act as a payment gateway. Payment service will be integrated along with flight booking system.



# Test Approach:

Feature testing occurs throughout each sprint with the final user acceptance testing (UAT) conducted in the testing sprint before going live. During the release planning meeting, the team should capture acceptance criteria and immediately add them as logical test cases linked to the product backlog item. In each sprint planning meeting, a sprint test plan should be created and reviewed. As the testing is done in each sprint, the results should be tracked along with which features have been successfully tested and which ones have defects. Care should be taken to distinguish defects and changes in requirements. If the team requires a change in requirement, this should be added to the product backlog and prioritized

# Assumptions:

The following are the assumptions:

* The tests assume that user logins are created that would be used for testing.
* Valid test data is available to test.
* Systems that would be used for testing are available.
* Payment credentials are integrated with Payment Systems.
* Network between the components in 100% reliable.
* Payment gateway is always up and running.

# Test Actors:

To efficiently test the various rights assigned to the various security groups, the team should create a mini test organization by following these steps:

1. If a reporting organizational business structure (OBS) is used in the system, create the appropriate reporting relationship for the test actors.
2. Validate the access rights for each of the security groups by logging in as the test actor and execute each test case.
3. When the system goes live, the test actors should be de-activated. Whenever there is a need to conduct use case or system validation, test actors can be re-activated.

The following actors could be required:

• Project manager or scrum master (PM1, PM2)

• Idea manager

• Project team member (TM1, TM2)

• Resource or capacity manager

• Scrum product owner (presuming the organization uses this role)

• Project stakeholders/executives (persons interested in the project, but is not assigned any specific tasks)

• PMO/IT requestor

• IT Finance team member

• Governance board member

• Project accounting team member

# Types of Testing:

The following types of Testing would be performed:

## Unit Testing

Unit tests are required whenever a software component is installed into an environment. While testing of the installation in the development environment would be adequate to validate the installation procedure prescribed for the solution, any installation into any subsequent environment must be appropriately validated. This testing is conducted by the agile core team members, along with the help of selected technical support staff.

## Feature Testing

Feature tests confirm the desired functionality of the delivered solution, and are specifically designed to validate that the solution addresses requirements as closely as possible. A key purpose of testing is to provide timely information about the quality of the system being built, and this testing is conducted by the agile core team members. Acceptance criteria should be clearly defined for each feature

## Functional Testing

Functional testing confirms the desired functionality of the delivered solution and is commonly based on the use cases defined for the solution. Specifically, functional tests are designed to simulate project management processes used by the customer, as closely as possible. Test cases step through the project management process through a variety of scenarios

## Integration Testing

Integration testing should be performed when all the required components have been configured or built, and the required interfaces have been installed. Each interface can be tested individually and test results signed off. This testing is conducted by the agile core team members, and with the functional administrators of interfacing systems.

## User Acceptance Testing

This testing is performed by end users before the system can be certified as ready to go live. It is conducted in the last sprint before go live, and by the end users who would use the system after it is live. Users must be given brief training in the requirements, how to use ACME Web Portal to book flights, where to find the various features, and how the system has been configured. Acceptance criteria must be clearly defined for the system functions required to go live.

## Regression Testing

This testing should be carried out on any systems which are part of the solution, and have been upgraded or patched during the lifetime of the solution. An appropriate regression test would involve appropriate unit, feature and integration testing related to the affected software product, and a specific test or set of tests related to any feature that is impacted by the upgraded or patched software

## Performance Testing

Performance testing evaluates system performance for speed and stability under load and stress. Tests should be conducted through emulations in a test environment before the full go live to all the end users. Since testing requires use of a separate automated testing tool, organizations should enlist help from their technical staff.

## Production Validation

Validation testing should be carried out after the build has been released to the production environment to ensure the functionality has been deployed correctly. Typically, production validation is executed during off hours using a sample data set in production. Proper and complete production verification requires knowledge of the solution and experience with common build issues. If issues are found, the agile team should determine if the functionality was deployed correctly or whether roll-back procedures must be implemented. Once completed, the test data sets should be removed or deactivated within the environment. Production validation is conducted by the agile core team members, administrator and end users.

## Training Validation

The purpose of training validation is to help ensure trainees will experience the same behavior/functionality that will be deployed in the production environment. This testing is conducted by the trainer or training administrator. In addition to validating functionality, trainers should ensure the performance of the application is sufficient for the number of users being trained. This testing is conducted by the trainer or training administrator.

# Testing Environments:

|  |  |  |  |
| --- | --- | --- | --- |
|  | Development Environment | Testing and Training Environments | Production Environment |
| Unit Testing | Yes |  |  |
| Feature Testing | Yes |  |  |
| Functional Testing | Yes |  |  |
| Integration Testing | Yes |  |  |
| Regression Testing |  | Yes |  |
| User Acceptance Testing |  | Yes |  |
| Production Validation. |  |  | Yes |
| Training Validation |  | Yes |  |

# Test Automation and Testing Tools

Automated testing tools can help to create and link test cases to backlog items. This reduces the possibility for human error during regression testing and iterative tests. As noted above, these types of tools are likely required to conduct performance testing. The project team should define test management and automation tools required for test execution. If no automated testing tools are available on-site, test cases and results should be tracked separately, such as in a spreadsheet

# Risk Analysis

• Resource Availability: Systems downtime, Team members going on leave

• Technical: Lacking Skilled Staff, Trainings

• Scope: Lack of Wide range of Devices and their Form Factors in Web, Mobile, Platforms

• Business Process:

• Training/User Adoption:

• Communication: Not being able to reach customers thru Chat, Phone and Email

# Test Planning and Execution

The steps involved in planning and executing testing are:

• Use cases/user stories have been clearly defined

• Build test cases and test scripts based on use cases

• Define acceptance criteria for each feature and each process

• Conduct testing

• Review and approve test results

• Migrate changes to production or for the next level of testing (such as user acceptance testing)

# Review and Approval

The agile core team and the project leader should be responsible for validating and approving all test results.

• The functional administrators of the interfacing systems should also sign off on integration test results.

• The end users and their authorized representative(s) should validate and sign off on the User Acceptance Test results and production validation.

• The trainer should validate and sign-off on the training content and training validation.