**ITWorks – Test Plan**

**Project name:** **SAAirline Testing**

**Author : Andre Alexandrov**

Version: 1.0

Date: 6/12/2022

Status: Draft

**For more information**

|  |  |
| --- | --- |
| **Project Contact:** Jake Brown |  |
| Name: xxxxxxxx  Email: xxxxxxxxx |  |

**Revision History:**

|  |  |  |  |
| --- | --- | --- | --- |
| Version | **Date** | **Author(s)** | **Change Description** |
| 1.0 | 13/11/2022 | Andre Alexandrov | *First draft.* |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

TABLE of Contents

[1. Introduction 4](#_Toc521390588)

[1.1. Test Plan Objectives 4](#_Toc521390589)

[2. Scope 4](#_Toc521390590)

[3. Test Strategy 4](#_Toc521390591)

[3.1 System Test 4](#_Toc521390592)

[3.2. Unit Test 4](#_Toc521390593)

[3.3. Security Test 4](#_Toc521390594)

[3.4. Automated Test 4](#_Toc521390595)

[3.5. Stress, Volume Test, Performance Test 4](#_Toc521390596)

[3.6. Recovery Test 5](#_Toc521390597)

[3.7. Documentation Test 5](#_Toc521390598)

[3.8. Beta Test 5](#_Toc521390599)

[3.9. User Acceptance Test 5](#_Toc521390600)

[4. Environment Requirements 5](#_Toc521390601)

[5. Test Schedule 5](#_Toc521390602)

[6. Control Procedures 5](#_Toc521390603)

[6.1 Reviews 5](#_Toc521390604)

[6.2 Bug Review meetings 6](#_Toc521390605)

[6.3 Change Request 6](#_Toc521390606)

[6.4 Defect Reporting 6](#_Toc521390607)

[7. Functions To Be Tested 6](#_Toc521390608)

[8. Resources and Responsibilities 6](#_Toc521390609)

[8.1. Resources 6](#_Toc521390610)

[8.2. Responsibilities 6](#_Toc521390611)

[9. Deliverables 7](#_Toc521390612)

[10. Suspension / Exit Criteria 7](#_Toc521390613)

[11. Resumption Criteria 8](#_Toc521390614)

[12. Dependencies 8](#_Toc521390615)

[12.1 Personnel Dependencies 8](#_Toc521390616)

[12.2 Software Dependencies 8](#_Toc521390617)

[12.3 Hardware Dependencies 8](#_Toc521390618)

[12.3 Test Data & Database 8](#_Toc521390619)

[13. Risks 8](#_Toc521390620)

[13.1. Schedule 8](#_Toc521390621)

[13.2. Technical 8](#_Toc521390622)

[13.3. Management 8](#_Toc521390623)

[13.4. Personnel 8](#_Toc521390624)

[13.5 Requirements 9](#_Toc521390625)

[14. Tools 9](#_Toc521390626)

[15. Documentation 9](#_Toc521390627)

[16. Approvals 9](#_Toc521390628)

# Introduction

The application that will be tested is the SAAirline ticketing application which was developed in C#, the test will be conducted using Nunit in visual studio 2019 along with two packages, Unit3TestAdapter, and Microsoft.NET.Test.Sdk

## Test Plan Objectives

This Test Plan objectives can be:

* Define the activities required to prepare for and conduct different type of tests.
* Communicate to all responsible parties related to different tests.
* Define deliverables and responsible parties.
* Communicate to all responsible parties the various dependencies of the system

# Scope

The scope of the application meets all the technical, functional and business requirements. This document will describe the overall test plan

# Test Strategy

The purpose of this section consists of a series of different tests that will fully exercise the system. The primary purpose of these tests is to uncover the systems limitations and measure its full capabilities. A list of the various planned tests and a brief explanation follows below.

## System Test

N/A

## 3.2. Unit Test

Unit test will be conducted to ensure that each method in each class should be performed or return the expected result with the designed input.

## 3.3. Security Test

N/A

## 3.4. Automated Test

N/A

## 3.5. Stress, Volume Test, Performance Test

N/A

## 3.6. Recovery Test

N/A

## 3.7. Documentation Test

N/A

## 3.8. Beta Test

N/A

## 3.9. User Acceptance Test

N/A

# 4. Environment Requirements

* Visual studio
* Nunit 3.13.3
* NUnit3TestAdapter 4.3.0
* Microsoft.NET.Test.Sdk 17.4

# 5. Test Schedule

This section covers the test schedule on different test type. Only include the type of tests apply to your situation. Put N/A if it is not applicable e.g.

* Ramp up / System familiarization N/A - N/A
* Unit Tests 13/11/2022 - 09/12/2022
* Integration Tests (if in scope) N/A - N/A
* User Acceptance Tesst (if in scope) N/A - N/A

# 6. Control Procedures

## 6.1 Reviews

A meeting notice, with related documents, will be emailed to each participant.

## 6.2 Bug Review meetings

Regular weekly meeting will be held to discuss reported defects. The development team will provide status/updates on all defects reported and the test department will provide addition defect information if needed. All member of the project team will participate.

## 6.3 Change Request

If functional changes are required, these proposed changes will be discussed with the Change Control Board (CCB). The CCB will determine the impact of the change and if/when it should be implemented.

## 6.4 Defect Reporting

When defects are found, the testers will complete a defect report on the defect tracking system. The defect tracking Systems is accessible by testers, developers & all members of the project team. When a defect has been fixed or more information is needed, the developer will change the status of the defect to indicate the current state. Once a defect is verified as FIXED by the testers, the testers will close the defect report.

# 7. Functions To Be Tested

The following is a list of functions that will be tested:

* Invoice.getDiscount(int memberCode)
* Invoice.setNumberOfSeats(int numberOfSeats)
* Seat.bookSeats(int num)
* customer.getFirstName()
* Customer.Customer(int memberType, String firstName, String lastName, String creditNumber, String creditType, String expiry)
* Sales.Flight(int economyRows, int economySeats, int firstClassRows, int firstClassSeats)
* Sales.Flight(int economyRows, int economySeats, int firstClassRows, int firstClassSeats)
* Sales. Invoice(int priceCode, Customer theCust, int rowNum, int startSeatNum, int seatsBooked)
* Flight(int economyRows, int economySeats, int firstClassRows, int firstClassSeats)
* Flight(int economyRows, int economySeats, int firstClassRows, int firstClassSeats)

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Test Cases Design** | | | | | | | |
| **Test case id**  **TC xxx** | **Name of method being tested e.g. class.method().**  **No need to include the parameters in this column.** | **Reason to test in business context** | **Data input (Constructor used and the Method to be tested include the parameters)** | **Test Case Design Technique used** | **Date/time**  **&**  **Duration (hours)** | **Resources Required**  **(Name of tester, hardware & software needed)** | **Expected result** |
| TC 1 | Invoice.setNumberOfSeats(int numberOfSeats) | To make sure that invoice seat selection is working for customers for correct payment | Constructors:  customerMain = new Sales.Customer(1, "john", "Smith", "4560361423329093", "Visa", "03/2026");  invoiceMain = new Sales.Invoice(0, customerMain, 1, 4, 1);  invoiceMain.setNumberOfSeats(2);  Assert.AreEqual(2, invoiceMain.getNumberOfSeats()); | Equivalence | 15/11/2022 | Andre Alexandrov  PC  Visual Studio  Microsoft.NET.Test.Sdk  NUnit  NUnit3TestAdapter | 2 |
| TC 2 | Invoice.getDiscount(int memberCode) | Ensure that discounts customers have work | Constructors:  customerMain = new Sales.Customer(1, "john", "Smith", "4560361423329093", "Visa", "03/2026");  invoiceMain = new Sales.Invoice(0, customerMain, 1, 4, 1);  Assert.AreEqual(0.9, invoiceMain.getDiscount(1)); | Equivalence | 15/11/2022 | Andre Alexandrov  PC  Visual Studio  Microsoft.NET.Test.Sdk  NUnit  NUnit3TestAdapter | 0.9 |
| TC 3 | Seat.bookSeats(int num) | To ensure that seat booking deducts seats correctly | Constructors:  seatMain = new Sales.Seat(100, 0);  seatMain.bookSeats(5);  Assert.AreEqual(6, seatMain.getCurrentSeat());  Assert.AreEqual(1, seatMain.getLastBooked()); | Equivalence | 15/11/2022 | Andre Alexandrov  PC  Visual Studio  Microsoft.NET.Test.Sdk  NUnit  NUnit3TestAdapter | 6  1 |
| TC 4 | customer.getFirstName() | To test if a null name is passed into the method | Constructors:  Sales.Customer customerNull = new Sales.Customer(1, null, "Smith", "4560361423329093", "Visa", "03/2026");  Assert.IsNull(customerNull.getFirstName()); | Equivalence | 15/11/2022 | Andre Alexandrov  PC  Visual Studio  Microsoft.NET.Test.Sdk  NUnit  NUnit3TestAdapter | Is Null |
| TC 5 | Customer.Customer(int memberType, String firstName, String lastName, String creditNumber, String creditType, String expiry) | Tests if identical customers are the same and that the customer is linked to an invoice | Constructor:  invoiceMain = new Sales.Invoice(0, customerMain, 1, 4, 1);  customerMain = new Sales.Customer(1, "john", "Smith", "4560361423329093", "Visa", "03/2026");  Assert.AreSame(invoiceMain.getTheCust(), customerMain); | Equivalence | 15/11/2022 | Andre Alexandrov  PC  Visual Studio  Microsoft.NET.Test.Sdk  NUnit  NUnit3TestAdapter | invoiceMain.getTheCust() |
| TC 6 | Sales.Flight(int economyRows, int economySeats, int firstClassRows, int firstClassSeats) | Making sure that flights have an exception when economy seats are entered as 0 | Constructor:  Sales.Flight flightTest;  Assert.Throws<ArgumentException>(() => flightTest = new Sales.Flight(10, 0, 15, 10)); | Equivalence | 15/11/2022 | Andre Alexandrov  PC  Visual Studio  Microsoft.NET.Test.Sdk  NUnit  NUnit3TestAdapter | Exception |
| TC 7 | Sales.Flight(int economyRows, int economySeats, int firstClassRows, int firstClassSeats) | Making sure that flights have an exception when first class seats are entered as 0 | Constructors:  Sales.Flight flightTest;  Assert.Throws<ArgumentException>(() => flightTest = new Sales.Flight(10, 10, 15, 0)); | Equivalence | 15/11/2022 | Andre Alexandrov  PC  Visual Studio  Microsoft.NET.Test.Sdk  NUnit  NUnit3TestAdapter | Exception |
| TC 8 | Sales. Invoice(int priceCode, Customer theCust, int rowNum, int startSeatNum, int seatsBooked) | Making sure charges give the expected result | Constructors:  invoiceMain = new Sales.Invoice(0, customerMain, 1, 4, 1);  Assert.AreEqual(2000, invoiceMain.getCharge(1)); | Equivalence | 15/11/2022 | Andre Alexandrov  PC  Visual Studio  Microsoft.NET.Test.Sdk  NUnit  NUnit3TestAdapter | 2000 |
| TC 9 | Flight(int economyRows, int economySeats, int firstClassRows, int firstClassSeats) | Making sure there is an exception when economy rows has no rows as there can not be seats without rows | Constructors:    Sales.Flight flightTest;  Assert.Throws<ArgumentException>(() => flightTest = new Sales.Flight(0, 10, 10, 10)); | Equivalence | 15/11/2022 | Andre Alexandrov  PC  Visual Studio  Microsoft.NET.Test.Sdk  NUnit  NUnit3TestAdapter | exception |
| TC 10 | Flight(int economyRows, int economySeats, int firstClassRows, int firstClassSeats) | Making sure there is an exception when first class rows has no rows as there can not be seats without rows | Constructors:  Sales.Flight flightTest;  Assert.Throws<ArgumentException>(() => flightTest = new Sales.Flight(10, 10, 0, 10)); | Equivalence | 15/11/2022 | Andre Alexandrov  PC  Visual Studio  Microsoft.NET.Test.Sdk  NUnit  NUnit3TestAdapter | Exception |

# 8. Resources and Responsibilities

Test manager will also be responsible for coordinating schedules, equipment, & tools for the testers as well as writing/updating the Test Plan, Weekly Test Status reports and Final Test Summary report. The testers will be responsible for writing the test cases and executing the tests.

## 8.1. Resources

The test team will consist of:

* 1 x test manager
* 1 x testers

## Responsibilities

|  |  |
| --- | --- |
| Test Manager | Responsible for project schedules and the overall success of the project. Participate on Change Control Board (CCB). |
| Testers | Responsible for performing the execution of different kind of tests. |

# Deliverables

Examples only. Add or delete the deliverables to suit for your situation.

|  |  |  |
| --- | --- | --- |
| **Deliverable** | **Responsibility** | **Completion Date** |
| Develop Test cases | Testers | 15/11/2022 |
| Test Case Review | Test manager, Testers | 15/11/2022 |
| Develop test scripts / procedures | Testers | 06/12/2022 |
| Execute test scripts / procedures | Testers | 06/12/2022 |
| Complete defect report | Everyone testing the product | On-going |
| Final Test Summary Report | Test manager | 06/12/2022 |

# 10. Suspension / Exit Criteria

If any defects are found which seriously impact the test progress, the test manager may choose to suspend the testing process.

For example, criteria that will justify test suspension are:

* Hardware/software is not available at the times indicated in the project schedule.
* Source code contains one or more critical defects, which seriously prevents or limits testing progress.
* Assigned test resources are not available when needed by the test team.

# Resumption Criteria

eIf the testing is suspended, resumption will only occur when the problem(s) that caused the suspension has been resolved. When a critical defect is the cause of the suspension, the “FIX” must be verified by the test team before testing is resumed.

# Dependencies

## Personnel Dependencies

The test team requires experience testers to design, perform and validate test cases.

The test team will also need the following resources available: Application developers and product owner and their delegates.

## Software Dependencies

Need the testing framework available and the source code must be unit tested and provided within the scheduled time outlined in the Project Schedule.

## 12.3 Hardware Dependencies

Often the workstations in the testing environment is located in a local area network (LAN) environment which needs to be available during normal working hours. Any downtime will affect the test schedule.

## Test Data & Database

If data Driven Testing is required, test data & database should be made available to the testers for use during testing. However, if the test data come from the data object through the constructor, the constructor may be constructed with max. possible combination of possible values through the input parameters.

# Risks

## 13.1. Schedule

schedule for each phase is very aggressive and could affect testing. A slip in the schedule in one of the other phases could result in a subsequent slip in the test phase. Close project management is crucial to meeting the forecasted completion date.

## 13.2. Technical

Since this is a new system, in the event of a failure the old system can be used. We will run our test in parallel with the production system so that there is no downtime of the existing system.

## 13.3. Management

+ Management support is required so when the test project falls behind, the test schedule does not get squeezed to make up for the delay. Management can reduce the risk of delays by supporting the test team throughout the testing phase and assigning more testers to this project with the required skills set.

## 13.4. Personnel

Due to the aggressive schedule, it is very important to have experienced testers on this project. Unexpected turnovers can impact the schedule. If attrition does happen, all efforts must be made to replace the experienced individual

## 13.5 Requirements

test plan and test schedule are based on the current Requirements Document. Any changes to the requirements could affect the test schedule and will need to be approved by the CCB.

# 14. Tools

We have the licensed product onsite and installed. All of the testers have been trained on the use of this test tool.

# Documentation

This section lists the documents that will be available at the end of the test phase:

* Test Plan (if any)
* Test Cases (if any)
* Requirements Validation Matrix (if any)
* Defect reports (if any)
* Test Summary Report (if any)

# 16. Approvals

|  |  |  |
| --- | --- | --- |
| **Name** | **Role** | **Date** |
|  |  |  |
| 1. **Andre** | Tester | xxxxxxxx |
|  |  |  |
| 1. **Jake Brown** | Test manager | xxxxxxxx |
|  |  |  |