Flight Management System

Test plan and Test case

Version 1.4

Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| **Date** | **Version** | **Description** | **Author** |
| 18/5/2015 | 1.0 | Created the test plan and test case document | Siyuan Chen |
| 19/5/2015 | 1.1 | Added introduction and requirement | Siyuan Chen |
| 20/5/2015 | 1.2 | Added test case | Siyuan Chen |
| 31/5/2015 | 1.3 | Test and record | Siyuan Chen |
| 1/6/2015 | 1.4 | Finalization and spelling check | Siyuan Chen |

Table of Contents

Test Plan

# Introduction

## Purpose

This Test Plan document for the Flight Management System supports the following objectives:

• We will test the Flight Management System whether or not can successful work and those functional/nonfunctional requirements are covered by the software.

• The major test requirements are follow the main software components: Reservation System, Profile Subsystem, Service Subsystem, Reporting Subsystem

• The test strategies will based on the different user side: Administrator Side, Customers and travel agencies, Reservation Manager and Staff, Flight Manager, Services Manager. We will test the results of each functionality respectively. If any part won’t work successfully, the error information will send to the coding member and modify the program.

• Based on the above requirements, we hope to achieve all the functions perfectly.

## Background

There are four main components to the system which will be described below:

* Reservation System that manages all flight reservations, seat selection, ticketing, flight availability, flight details, rates and conditions.
* Profile Subsystem that manages individual passengers and travel agency profiles.
* Service Subsystem that manages in-flight services such as food and drinks.
* Reporting Subsystem to generate various summary reports such as Passenger Report, Cash Report and Booking Summaries.

Each user will use their unique account name and password to login the system. There will also demonstrate different interface to different type of user. And the database that has been created and connected should be successful used when call or input new data.

## Scope

The programmer should have the unit testing in the coding step to ensure the internal workings of each module can successful compile and work. When the whole software finished, we will use system testing for the function and performance.

In the whole testing, it includes all functional features based on three use case: System administrator, Project manager, Project member. And also few nonfunctional requirement such as speed and security part.

Assumptions: 1. If find any changing of interface could let user to feel more comfortable or convenient during testing, although the function can work well, the idea should be recorded and develop the software.

2. If the reaction speed of the software is too low and make user to feel impatient, although the function can work well, it should also be optimized.

Risk and contingencies : 1. The software could not finished when need to test.

2. Software might crash because of memory reasons when testing.

3. The functionality can not work successfully

## Project Identification

The table below identifies the documentation and availability used for developing the *test plan*:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Document  (and version / date)** | **Created or Available** | **Received or Reviewed** | **Author or Resource** | **Notes** |
| Requirements Specification | Yes | Yes |  |  |
| Functional Specification | Yes | Yes |  |  |
| Use-Case Reports | Yes | Yes |  |  |
| Project Plan | Yes | Yes |  |  |
| Design Specifications | Yes | Yes |  |  |
| Prototype | Yes | Yes |  |  |
| User’s Manuals | Yes | Yes |  |  |
| Business Model or Flow | Yes | Yes |  |  |
| Data Model or Flow | Yes | Yes |  |  |
| Business Functions and Rules | Yes | Yes |  |  |
| Project or Business Risk Assessment | Yes | Yes |  |  |

# Requirements for Test

The listing below identifies those items⎯use cases, functional requirements, and non-functional requirements⎯that have been identified as targets for testing. This list represents what will be tested.

- System administrator: 1. Login

2. Create staff profile

3. Edit staff profile

- Customers and travel agencies: 1. Login

2. Sign up

3. Make booking

4. Edit account details

5. Cancel booking

6. Edit service

7. Generate flight reports

8. Close account

- Reservation Manager: 1. Set ticket price

2. Move passengers between flights

3. Change passengers seating

4. Edit watch and no fly list

- Flight Manager: 1. Edit routes

2. Edit fleet

3. Edit flight schedule

4. Edit airports

- Services Manager: 1. Add service

2. Remove service

3. Edit service price

4. Edit service availability

- Profile Manager: 1. Edit watch and no fly list

2. Edit travel agency and customer profile

3. Close account

# Test Strategy

## Testing Types

### Function Testing

3.1.2.1 Profile Subsystem Testing

3.1.2.1.1 Administrator Testing

|  |  |
| --- | --- |
| Test Objective: | Ensure functionality of System administrator |
| Technique: | 1. Login  • use correct user name and password, Successfully login  • use invalid user name and password, can’t login and displays an error message on the user interface  2. Logout  • Successfully logout  3. Create a staff profile  • Successfully create a staff profile  • Point out the error and doesn’t produce a new staff profile, when the user name is same with an existing one  4. Modify a staff profile  • Successfully modify a staff profile  • Point out the error and doesn’t change the profile, when the user name is same with an existing one  5. Remove a staff profile  • Successfully remove a staff profile  6. On occurrence of any exception during the action  • Cancel the action and doesn’t change any thing |
| Completion Criteria: | All planned tests have been executed.  All identified defects have been addressed. |
| Special Considerations: |  |

3.1.2.1.2 User Testing

|  |  |
| --- | --- |
| Test Objective: | Ensure functionality of user |
| Technique: | 1. Login  • use correct user name and password, Successfully login  • use invalid user name and password, can’t login and displays an error message on the user interface  2. Logout  • Successfully logout  3. Create a user profile  • The user profile has been created successfully  • Point out the error and doesn’t produce a new staff profile, when the user name is same with an existing one  4. When login to the user side  • Includes the domain specific options: Make a Booking, Modify Services for Existing Booking, Cancel a Booking, Change Password, Modify Profile, Remove Profile, Request Statistical Report  5. Modify the details of profile  • Successfully modify a user profile  6. Change the password  • Successfully change the user password  • Point out the error and let user input a new password, when the new password is same with old one  7. Remove the profile  • Successfully remove the user profile  8. On occurrence of any exception during the action  • Cancel the action and doesn’t change any thing |
| Completion Criteria: | All planned tests have been executed.  All identified defects have been addressed. |
| Special Considerations: |  |

3.1.2.1.3 Profile manager Testing

|  |  |
| --- | --- |
| Test Objective: | Ensure functionality of Profile manager |
| Technique: | 1. Login  • use correct user name and password, Successfully login  • use invalid user name and password, can’t login and displays an error message on the user interface  2. Logout  • Successfully logout  3. Remove the profile  • Successfully remove the user profile  4. Change the profile  • Successfully change the user profile  5. On occurrence of any exception during the action  • Cancel the action and doesn’t change any thing |
| Completion Criteria: | All planned tests have been executed.  All identified defects have been addressed. |
| Special Considerations: |  |

3.1.2.2 Reservation Subsystem

3.1.2.2.1 Reservation Manager and Staff Testing

|  |  |
| --- | --- |
| Test Objective: | Ensure functionality of Reservation Manager and Staff |
| Technique: | 1. Login  • use correct user name and password, Successfully login  • use invalid user name and password, can’t login and displays an error message on the user interface  2. Logout  • Successfully logout  3. When login to the reservation manager and staff side  • Includes the domain specific options: Flight Transfer, Modify Customer Seat Selection  4.Transfer a customer from one flight to another  • Successfully transfer the flight  • Point out the error and the flight doesn’t be changed when the flight seats are not enough  5. Modify customer seat selection  • Successfully modify the seat selection  • Point out the error and the seat selection doesn’t be changed when the flight seats are not enough  6. Set ticket pricing of a flight  • Successfully set the price  7. On occurrence of any exception during the action  • Cancel the action and doesn’t change any thing |
| Completion Criteria: | All planned tests have been executed.  All identified defects have been addressed. |
| Special Considerations: |  |

3.1.2.2.2 Booking Process Testing

|  |  |
| --- | --- |
| Test Objective: | Ensure functionality of Booking Process |
| Technique: | 1. The user selects the flight and book it  • Successfully book the flight  • Point out the error and the details don’t be recorded when the flight seats are not enough  • Point out the error and the details don’t be recorded when the Customer no fly status is set to “No Fly”  • Point out the error and the details don’t be recorded when the Customer passport holder status is set to false  2. The user views the booking summary  • Successfully display a booking summary to the user  3. The user selects in-flight services for a flight  • Successfully select services  4. Modify in-flight services for existing booking  • Successfully modify services  5. The user selects the flight and cancel it  • Display warning and cancellation fee, successfully cancel the flight  6. On occurrence of any exception during the action  • Cancel the action and doesn’t change any thing |
| Completion Criteria: | All planned tests have been executed.  All identified defects have been addressed. |
| Special Considerations: |  |

3.1.2.2.3 Flight Manager Testing

|  |  |
| --- | --- |
| Test Objective: | Ensure functionality of Flight Manager |
| Technique: | 1. Login  • use correct user name and password, Successfully login  • use invalid user name and password, can’t login and displays an error message on the user interface  2. Logout  • Successfully logout  3. Modify details of routes  • Successfully modify details of routes: origin, destination, codeshare, number of stops  4. Modify records  • Successfully modify records: Airport Id Number,  Airport Name, City, Country, IATA Code, Latitude, Longitude, Altitude, Time Zone, DST, TZ Database Time Zone  5. Modify the airlines fleet of aircraft  • Successfully modify the airlines fleet of aircraft: Aircraft Name, Total in Service, First Class Seats Available, Business Class Seats Available, Premium Economy Seats Available, Economy Class Seats Available, Total Seats Available  6. Modify the flight schedule  • Successfully modify the airlines fleet of aircraft: Aircraft Name, Total in Service, First Class Seats Available, Business Class Seats Available, Premium Economy Seats Available, Economy Class Seats Available, Total Seats Available  7. On occurrence of any exception during the action  • Cancel the action and doesn’t change any thing |
| Completion Criteria: | All planned tests have been executed.  All identified defects have been addressed. |
| Special Considerations: |  |

3.1.2.3 Services Subsystem Testing

3.1.2.3.1 Services Manager Testing

|  |  |
| --- | --- |
| Test Objective: | Ensure functionality of Services Manager |
| Technique: | 1. Login  • use correct user name and password, Successfully login  • use invalid user name and password, can’t login and displays an error message on the user interface  2. Logout  • Successfully logout  3. When login to the services manager side  • Includes the domain specific options: Modify Services  4. Modify the in-flight services available for purchase from the airline  • Successfully modify the in-flight services  5. On occurrence of any exception during the action  • Cancel the action and doesn’t change any thing |
| Completion Criteria: | All planned tests have been executed.  All identified defects have been addressed. |
| Special Considerations: |  |

3.1.2.4 Reporting Subsystem Testing

|  |  |
| --- | --- |
| Test Objective: | Ensure functionality of user to request statistical report |
| Technique: | 1. The user requests the statistical report  • Successfully request statistical report  2. On occurrence of any exception during the action  • Cancel the action and doesn’t change any thing |
| Completion Criteria: | All planned tests have been executed.  All identified defects have been addressed. |
| Special Considerations: |  |

|  |  |
| --- | --- |
| Test Objective: | Ensure functionality of reporting system |
| Technique: | 1. The reporting manager login to the system  • Includes the domain specific options: Request Revenue Report, Request Popular Services Report, Request Popular Route Report  2. Display reports relating to (daily/monthly/annual) airline revenue  • Successfully request report  3. Display reports relating to (daily/monthly/annual) service popularity  • Successfully display the service popularity  4. Display reports relating to user (customer/travel agent) statistics and historical data  • Successfully display the statistics and historical data  5. Display reports relating to (daily/monthly/annual) route popularity  • Successfully display the route popularity  6. On occurrence of any exception during the action  • Cancel the action and doesn’t change any thing |
| Completion Criteria: | All planned tests have been executed.  All identified defects have been addressed. |
| Special Considerations: |  |

### Performance Profiling

|  |  |
| --- | --- |
| Test Objective: | Verify performance behaviors for designated transactions |
| Technique: | 1. The login process should not take longer than 10 seconds  • have been tested 5 times and all results are below 10 seconds.  2. The system must not take longer than 10 seconds to generate each of the analysis  • When did the function testing above, each part all below 10 seconds  3. The system should automatically log out after 5 minutes of being idle and ask the user to re-log in  • Successfully |
| Completion Criteria: | Single Transaction or single user: Successful completion of the test scripts without any failures and within the expected or required time allocation per transaction.  Multiple transactions or multiple users: Successful completion of the test scripts without any failures and within acceptable time allocation. |
| Special Considerations: |  |

### Configuration Testing

|  |  |
| --- | --- |
| Test Objective: | Verify that the target-of-test functions properly on the required hardware and software configurations. |
| Technique: | 1. Open the Microsoft Word with the software  • Successfully  2. Open the Adobe PDF reader with the software  • Successfully |
| Completion Criteria: | For each combination of the target-of-test and non-target-of-test software, all transactions are successfully completed without failure. |
| Special Considerations: | • A 40- page document in Word  • A 38- page document in PDF |

# Resources

## Workers

This table shows the staffing assumptions for the project.

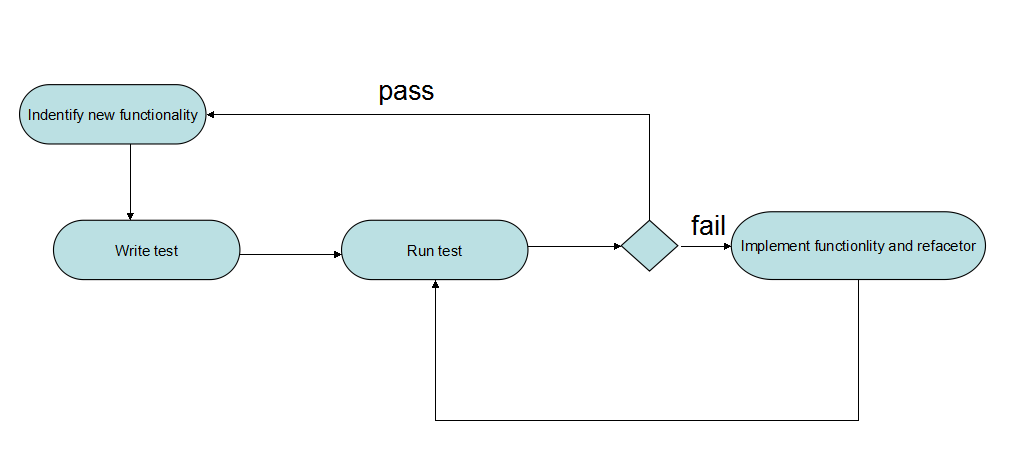
|  |  |  |
| --- | --- | --- |
| Human Resources | | |
| Worker | Minimum Resources Recommended  (number of full-time workers allocated) | Specific Responsibilities or Comments |
| Test Manager,  Test Project Manager | Matthew King  Siyuan Chen  Jicheng Li  Michael Kong | Provides management oversight.  Responsibilities:   * provide technical direction * acquire appropriate resources * provide management reporting |
| Test Designer | Matthew King  Siyuan Chen  Jicheng Li  Michael Kong | Identifies, prioritizes, and implements test cases.  Responsibilities:   * generate test plan * generate test model * evaluate effectiveness of test effort |
| Tester | Matthew King  Siyuan Chen  Jicheng Li  Michael Kong | Executes the tests.  Responsibilities:   * execute tests * log results * recover from errors * document change requests |
| Test System Administrator | Matthew King  Siyuan Chen  Jicheng Li  Michael Kong | Ensures test environment and assets are managed and maintained.  Responsibilities:   * administer test management system * install and manage worker access to test systems |
| Database Administratator, Database Manager | Jicheng Li  Michael Kong | Ensures test data (database) environment and assets are managed and maintained.  Responsibilities:   * administer test data (database) |
| Designer | Matthew King  Siyuan Chen  Jicheng Li  Michael Kong | 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 | Jicheng Li  Michael Kong | Implements and unit tests the test classes and test packages.  Responsibilities:   * creates the test classes and packages implemented in the test model |

# Project Milestones

|  |  |  |  |
| --- | --- | --- | --- |
| **Milestone Task** | **Effort** | **Start Date** | **End Date** |
| Plan Test |  | 4/5/2015 | 10/5/2015 |
| Design Test |  | 11/5/2015 | 22/5/2015 |
| Implement Test |  | 31/5/2015 | 31/5/2015 |
| Execute Test |  | 31/5/2015 | 31/5/2015 |
| Evaluate Test |  | 1/6/2015 | 1/6/2015 |

# Deliverables

## Test Model



## Test Logs

[Describe the method and tools used to record and report on the test results and testing status.]

## Defect Reports

[In this section identify the method and tools used to record, track, and report on test incidents and their status.]

# Appendix A: Project Tasks

Below are the test related tasks:

• Plan Test

* + identify requirements for test
  + assess risk
  + develop test strategy
  + identify test resources
  + create schedule
  + generate Test Plan

• Design Test

- prepare workload analysis

- identify and describe test cases

- identify and structure test procedures

- review and assess test coverage

• Implement Test

* + record or program test scripts
  + identify test-specific functionality in the Design and Implementation Model
  + establish external data sets

• Execute Test

- execute Test procedures

- 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