Airline Reservation System Test Plan

Han Xiao - hxia0019@student.monash.edu
Yuhang Huang - yhua0226@student.monash.edu
Jinhui Yuan - jyua0030@student.monash.edu

Testing Objectives

The system integration test of the Airline Reservation System should validate from the requirements that:

1. Query flight information function works correctly.
2. Display the correct value to the passengers when they query flight information.
3. Buying direct flight tickets function works correctly.
4. Buying tickets with the transfer flights function works correctly.
5. Display the correct value to the passengers when they buy a ticket.
6. All calculations are correct.
7. Display prompts when the user enters the error message。
8. The system is easy to use by the end-users.

Scope of Testing

The system integration testing for the Airline Reservation System will cover the airplane, flight, flight collection, passenger, ticket, ticket collection, and ticket system components.

Approach

Assumptions/Constraints

- Assume that the test environment has been built, including the operating system, Java development environment and necessary library files.
- Assume that all dependencies (such as databases, network connections, etc.) are running normally without any faults or interruptions.
- Utilize the TDD approach to test and expand the code base's components, including Java classes and methods.
- Before being moved to the system integration testing environment, every build of the Airline Reservation System has successfully passed unit testing.

Software requirements

- All the software modules in the Airline Reservation System will be tested.
- All requirements outlined in the Specification Document will be tested.

Testing tools

- JUnit
- Git
- Maven

Test Plan

Test team

Name	ID	Role	Responsibility
Han Xiao	33414548	Unit Test	Responsible for unit testing of airplane, flight, flight collection, person and passenger.
Yuhang Huang	33414513	Unit Test & IntegrationTest	Responsible for unit testing of ticket, ticketCollection and Integration of ticketSystem
Jinhui Yuan	33414521	Unit Test & Integration Test	Responsible for unit testing of flightCollection, ticketCollection, Ticket and Integration of ticketSystem

Test environment

All test cases will be executed on the IntelliJ IDEA. The SDK version of this project is Oracle OpenJDK version17.

Hardware

- Macbook Air

  Apple M2 chip

  8GB unified memory

  256GB SSD memory

- ASUS TUF GAMING A15 FA507RM_FA507RM

AMD Ryzen 7 6800H with Radeon Graphics 3.20 GHz

16.0 GB Memory

512 SSD memory

- ASUS TUF GAMING A15 FA507RM_FA507RM

  AMD Ryzen 7 6800H with Radeon Graphics 3.20 GHz

  16.0 GB Memory

  512 SSD memory

Network

- LAN

Software

- Airline Reservation System

Server

- MacOS Monterey 12.5
- MS Windows 11 operating system
- MS Windows 11 operating system

Testing procedure

Test Cases

ID	Purpose for Test	Test Steps	Expected Result
001	When choosing a ticket, a valid city is used	Input ticket with invalid city name.	Error message
002	Appropriate checks have been implemented to validate flight information	Input a flight with two cities which do not really exist in the ticket system.	Error message
003	Test for buying an already booked ticket	Set a ticket which status is true then buy this	Error message

ID	Purpose for Test	Test Steps	Expected Result
		ticket.	
004	Test validate passenger information	Buy a ticket and then input different format information.	Invalid email & code & Age format
			Passport number should not be more than 9 characters long
			cardNumber & phoneNumber cannot be empty
			Error choice for buying a ticket
005	Test valid Ticket Information	Buy a ticket with valid input and then check the returned value.	Test Pass
006	Test correct value displayed to passenger	But a ticket with valid input and then check the returned price value.	Test Pass

Order of Testing

Testing will be conducted in a systematic manner, with the sequence of tests largely determined by the order of builds. In each test we will also extend the original code following the requirements in assignment document and continue to do the test on our extend code:

Unit test:

1. Airplane test
2. Flight test
3. Person & Passenger test
4. Ticket test
5. Ticket collection test
6. Flight collection test:

Integration test:

1. Check ticket system (containing choose ticket & buy ticket)