Flight Management System

Version 1.0

Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| **Date** | **Version** | **Description** | **Author** |
| 11/Apr/2015 | 1.0 | Created the use case specification and description document | Michael Kong |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

Table of Contents

1. Introduction 4

1.1 Brief Description 4

1.2 Glossary of Terms 4

2. Use Case Specification and Description 4

2.1 Use Case: Login 4

2.2 Use Case: Enter Wrong Credentials 5

2.3 Use Case: Signup 5

2.4 Use Case: Enter Invalid Details 5

2.5 Use Case: Cancel Booking 6

2.6 Use Case: Edit Account Details 6

2.7 Use Case: Make Booking 6

2.8 Use Case: Generate Flight Reports 7

2.9 Use Case: Add Persons 7

2.10 Use Case: Add Customers 8

2.11 Use Case: Enter Invalid Username 8

2.12 Use Case: Book Services 8

2.13 Use Case: Cancel Customer Booking 9

2.14 Use Case: Edit Services 9

2.15 Use Case: Add Services 10

2.16 Use Case: Remove Services 10

2.17 Use Case: Edit Service Price 11

2.18 Use Case: Edit Service Availability 11

2.19 Use Case: Edit Routes 11

2.20 Use Case: Edit Fleet 12

2.21 Use Case: Edit Flight Schedule 13

2.22 Use Case: Edit Airports 14

2.23 Use Case: Move Passengers Between Flights 15

2.24 Use Case: Set Ticket Prices 16

2.25 Use Case: Change Passenger Seating 16

2.26 Use Case: Edit Watch and No Fly List 16

2.27 Use Case: Enter Invalid Flight 17

2.28 Use Case: Enter Invalid Price 17

2.29 Use Case: Enter Invalid Seat Number 18

2.30 Use Case: Enter Flight and Customer Details 18

2.31 Use Case: Edit Travel Agency and Customer Account 18

2.32 Use Case: Close Account 19

2.33 Use Case: Create Staff Profiles 19

2.34 Use Case: Edit Staff Profiles 20

2.35 Use Case: Change Password 20

# Introduction

## Brief Description

This document covers all the use cases involved in the Flight Management System. Each use case description will cover exactly one use case as presented in the use case diagram. A single use case description contains the use case name, the use case ID, the stakeholders and goals, the description of the use case in question, the participating actors, the trigger that activates this use case, and the flow of events including normal flows, sub-flows and exceptional flows.

## Glossary of Terms

**Airport details** – Refers to the details of an airport that are: airport ID, airport name, city, country, IATA/FAA code, latitude, longitude, altitude, time zone, DST and TZ database time zone.

**Flight details** – Refers to the details of a flight that are: flight ID, plane type, route number, departure time and arrival time.

**Person details –** Refers to the details of the extra person that the Customer actor or Travel Agency actor might add in a booking that are: title, first name, last name, gender, date of birth, phone number, email, street address, state, city, country, credit card type, credit card number, and whether they hold a passport.

**Plane details** – Refers to the details of a plane that are: the aircraft model, number of first class seats, number of business class seats, number of premium economy class seats and number of ecnomy seats.

**User** – When in used in context of a use case description, it refers to the actors of that use case in question.

**User** **credentials** – This refers to the username and password of the system’s user that will be used as the primary means of identifying each user of the system.

**User details** – Refers to the details of the user that are: title, first name, last name, e gender, date of birth, phone number, email, street address, state, city, country, credit card type, credit card number, and whether they hold a passport.

# Use Case Specification and Description

## Use Case: Login

|  |  |
| --- | --- |
| **Name:** Login | **ID:** ACC\_05 |
| **Stakeholders and goals:**  Airline’s Human Resource Manager – Wants to save cost on hiring extra staff to serve customers.  Customers, Travel Agencies and Normal Staff – Wants a fast and convenient method to access the system. | |
| **Description:** A user must enter their user credentials into the system through the interface to access the functionality of the system. | |
| **Actors:** Travel agency, Customer, Normal Staff | |
| **Trigger:** User accesses the login interface of the system. | |
| **Normal flow:**   1. System prompts user for user credentials. 2. User enters user credentials into the system. 3. System checks the user credentials with the database. 4. System logs user into the system and displays the appropriate user main menu for that particular user. 5. End | |
| **Sub-flows:** None | |
| **Alternative/Exceptional flows:**  2a Incorrect user credentials entered: The *Enter Wrong Credentials* use case is performed. Go to step 1 after execution. | |

## Use Case: Enter Wrong Credentials

|  |  |
| --- | --- |
| **Name:** Enter Wrong Credentials | **ID:** EX\_02 |
| **Stakeholders and goals:**  Customers, Travel Agencies and Normal Staff – Wants a way to know whether the credentials entered are valid or not. | |
| **Description:** A user has entered incorrect user credentials into the system. | |
| **Actors:** Travel agency, Customer, Normal Staff | |
| **Trigger:** User enters the wrong credentials into the system. | |
| **Normal flow:**   1. System finds a mismatch in the user credentials and cannot find the corresponding user credentials. 2. System displays an error message on the user interface. 3. User acknowledges the error. 4. End | |
| **Sub-flows:** None | |
| **Alternative/Exceptional flows:** None | |

## Use Case: Signup

|  |  |
| --- | --- |
| **Name:** Signup | **ID:** ACC\_01 |
| **Stakeholders and goals:**  Airline’s Human Resource Manager – Wants to save cost on hiring extra staff to serve customers.  Customers, Travel Agencies and Normal Staff – Wants a fast and convenient method create an account to access the system. | |
| **Description:** A customer or travel agency wants to create a new account. | |
| **Actors:** Travel agency, Customer | |
| **Trigger:** User accesses the signup interface of the system. | |
| **Normal flow:**   1. System prompts user for user details. 2. User enters user details into the system. 3. System displays a “Signup Successful” message 4. End | |
| **Sub-flows:** None | |
| **Alternative/Exceptional flows:**  2a Invalid details are entered: The *Enter Invalid Details* use case is performed. Go to step 1 after execution. | |

## Use Case: Enter Invalid Details

|  |  |
| --- | --- |
| **Name:** Enter Invalid Details | **ID:** EX\_03 |
| **Stakeholders and goals:**  Customers and Travel Agencies – Wants a way to know whether the details entered are valid or not. | |
| **Description:** A user has entered invalid user details into the system | |
| **Actors:** Travel agency, Customer | |
| **Trigger:** User enters invalid details into the system. | |
| **Normal flow:**   1. System finds that the details do not match the expected format for a particular field. 2. System displays an error message on the user interface. 3. End | |
| **Sub-flows:** None | |
| **Alternative/Exceptional flows:** None | |

## Use Case: Cancel Booking

|  |  |
| --- | --- |
| **Name:** Cancel Booking | **ID:** BOOK\_06 |
| **Stakeholders and goals:**  Airline’s Human Resource Manager – Wants to save cost on hiring extra staff to cancel flight booking for customers.  Customers and Travel Agencies – Wants a fast and convenient method create an account to cancel a flight booking. | |
| **Description:** User wants to cancel a flight booking, but if the user is a Travel Agency or Customer, they will be charged a cancellation fee. | |
| **Actors:** Travel agency, Customer | |
| **Trigger:** User chooses the Cancel Booking option from the main menu. | |
| **Normal flow:**   1. User selects *Cancel Booking* option from the main menu. 2. System displays all bookings in a numbered list and prompts the user for the number associated with the booking. 3. User selects the number associated with the booking to cancel. 4. System acknowledges the cancellation and updates the database. 5. System charges a cancellation fee to the user’s account. 6. System returns to the main menu. 7. End | |
| **Sub-flows:** None | |
| **Alternative/Exceptional flows:** None | |

## Use Case: Edit Account Details

|  |  |
| --- | --- |
| **Name:** Edit Account Details | **ID:** ACC\_02 |
| **Stakeholders and goals:**  Airline’s Human Resource Manager – Wants to save cost on hiring extra staff to manually edit customer details.  Customers and Travel Agencies – Wants to be able to update and edit their personal details in the account. | |
| **Description:** User wants to edit their user details and have it updated in the system. | |
| **Actors:** Travel agency, Customer | |
| **Trigger:** User chooses the Modify Account Details option from the main menu. | |
| **Normal flow:**   1. System displays the modifiable user details in a numbered list and prompts user for the number associated with the detail. 2. User selects the user detail that they want to modify by the number. 3. System displays a prompt to retrieve input. 4. User enters new detail. 5. System saves modified user detail to database. 6. System returns to the main menu. 7. End | |
| **Sub-flows:** None | |
| **Alternative/Exceptional flows:** None | |

## Use Case: Make Booking

|  |  |
| --- | --- |
| **Name:** Make Booking | **ID:** BOOK\_01 |
| **Stakeholders and goals:**  Airline’s Human Resource Manager – Wants to save cost on hiring extra staff to manually make bookings for customers.  Customers and Travel Agencies – Wants to be able to make flight bookings quickly and conveniently. | |
| **Description:** User wants to make a booking for a flight and perform payment for their booking. | |
| **Actors:** Travel agency, Customer | |
| **Trigger:** User chooses the Book Flight option from the main menu. | |
| **Normal flow:**   1. System prompts user for flight destination and source. 2. User enters desired flight destination and source. 3. System displays the list of flights available corresponding to the given flight destination and source in a numbered list. 4. User selects flight to book by the corresponding number. 5. System performs the appropriate flow according to the type of user:    1. If the user is a Customer Actor, the *Add Persons* use case is performed.    2. If the user is a Travel Agency Actor, the *Add Customers* use case is performed. 6. System displays available seats in a numbered list and prompts user for choice. 7. User enters the seat numbers. 8. System prompts user if they want to book services. 9. User enters choice whether they want to book services. 10. System displays booking summary and prompts user to make payment. 11. User pays by credit card and can choose whether to use frequent flier points to get a discount. 12. System records payment details and updates database. 13. System sends flight information to user via email. 14. System returns to main menu. 15. End | |
| **Sub-flows:** None | |
| **Alternative/Exceptional flows:**  7a User enters invalid seats: The *Enter Invalid Seats* use case is performed. Go to step 6 after execution.  9a User chooses to book services: The *Book Services* use case is performed. Go to step 8 after execution. | |

## Use Case: Generate Flight Reports

|  |  |
| --- | --- |
| **Name:** Generate Flight Reports | **ID:** REP\_01 |
| **Stakeholders and goals:**  Airline’s Human Resource Manager – Wants to save cost on hiring extra staff to manually create reports from the raw data in the system.  Customers and Travel Agencies – Wants to be able to generate and view reports of their flight history.  Normal Staff – Wants to be able to generate and view reports of flight popularity, daily revenue and other informative reports. | |
| **Description:** User wants a copy of their flight details in a report format. | |
| **Actors:** Travel Agency, Customer, Normal Staff | |
| **Trigger:** User has chosen the Generate Flight Reports option in the main menu. | |
| **Normal flow:**   1. System lists the types of reports it can generate in a numbered list and prompts user to enter their selection. 2. User enters their choice into the system. 3. System retrieves the appropriate data from the database depending on the type of report chosen. 4. System processes retrieved data, creates the report and saves the report. 5. System sends the report to the user’s email. 6. System returns to main menu. 7. End | |
| **Sub-flows:** None | |
| **Alternative/Exceptional flows:** None | |

## Use Case: Add Persons

|  |  |
| --- | --- |
| **Name:** Add Persons | **ID:** BOOK\_04 |
| **Stakeholders and goals:**  Airline’s Human Resource Manager – Wants to save cost on hiring extra staff to manually add persons to a booking.  Customers and Travel Agencies – Wants to be able to conveniently add extra persons to their flight booking. | |
| **Description:** User wants to make a booking for additional people on their behalf. | |
| **Actors:** Travel Agency, Customer | |
| **Trigger:** Customer has selected a flight in the Make Booking use case OR Travel Agency chooses to make booking on behalf of other persons. | |
| **Normal flow:**   1. System prompts user if they want to add persons to the booking. 2. If user does not want to add persons to booking, go to step 7. 3. System prompts user for the person details. 4. User fills in the person details. 5. System prompts user whether there is another person. 6. If user enters that there is another person, perform step 3-5 again. 7. End | |
| **Sub-flows:** None | |
| **Alternative/Exceptional flows:**  4a Invalid details are entered: The *Enter Invalid Details* use case is performed. Go to step 3 after execution. | |

## Use Case: Add Customers

|  |  |
| --- | --- |
| **Name:** Add Customers | **ID:** BOOK\_05 |
| **Stakeholders and goals:**  Airline’s Human Resource Manager – Wants to save cost on hiring extra staff to manually add customers to a booking.  Travel Agencies – Wants to be able to add customers who already exist in the system to the booking. | |
| **Description:** User wants to make a booking for customers. | |
| **Actors:** Travel Agency | |
| **Trigger:** User has selected a flight in the Make Booking use case. | |
| **Normal flow:**   1. System prompts for the usernames of the customers to book for. 2. User enters usernames of the customers. 3. *Add Persons* use case is performed. 4. End | |
| **Sub-flows:** None | |
| **Alternative/Exceptional flows:**  2a Invalid usernames are entered: The *Enter Invalid Username* use case is performed. Go to step 1 after execution. | |

## Use Case: Enter Invalid Username

|  |  |
| --- | --- |
| **Name:** Enter Invalid Username | **ID:** EX\_04 |
| **Stakeholders and goals:**  Travel Agencies and Normal Staff – Wants a way to know whether the usernames entered are valid or not. | |
| **Description:** User has entered invalid username into the system and the system will display an error. | |
| **Actors:** Travel Agency, Normal Staff | |
| **Trigger:** User has entered a username/usernames that does not exist in the system. | |
| **Normal flow:**   1. System cannot find the username/usernames provided. 2. System displays an error message on the user interface. 3. User acknowledges the error message. 4. End | |
| **Sub-flows:** None | |
| **Alternative/Exceptional flows:** None | |

## Use Case: Book Services

|  |  |
| --- | --- |
| **Name:** Book Services | **ID:** BOOK\_02 |
| **Stakeholders and goals:**  Airline’s Human Resource Manager – Wants to save cost on hiring extra staff to manually book services on the flight for a customer.  Customer and Travel Agencies – Wants to be able to add services to their booking. | |
| **Description:** User wants to make booking for services that they can have on the flight. | |
| **Actors:** Travel agency, Customer, Normal Staff | |
| **Trigger:** User chooses to book services when prompted in the Make Booking use case. | |
| **Normal flow:**   1. System displays all services in a numbered list. 2. User selects the numbers of the services that they want. 3. System records the services for this user. 4. End | |
| **Sub-flows:** None | |
| **Alternative/Exceptional flows:** None | |

## Use Case: Cancel Customer Booking

|  |  |
| --- | --- |
| **Name:** Cancel Customer Booking | **ID:** BOOK\_07 |
| **Stakeholders and goals:**  Airline’s Human Resource Manager – Wants to save cost on hiring extra staff to manually cancel bookings of customers.  Normal Staff – Wants to be able to cancel the bookings of customers conveniently and efficiently. | |
| **Description:** User wants to cancel the booking of a customer | |
| **Actors:** Normal Staff | |
| **Trigger:** User chooses the Cancel Customer Booking option from the main menu. | |
| **Normal flow:**   1. System prompts user for a customer or travel agency’s username. 2. User enters a username into the system. 3. System displays all bookings for the user associated with that username in a numbered list and prompts the user for the number associated with the booking. 4. User selects the number associated with the booking to cancel. 5. System acknowledges the cancellation and updates the database. 6. System returns to the main menu. 7. End | |
| **Sub-flows:** None | |
| **Alternative/Exceptional flows:**  3a Invalid username is entered: The *Enter Invalid Username* use case is performed. Go to step 2 after execution. | |

## Use Case: Edit Services

|  |  |
| --- | --- |
| **Name:** Edit Services | **ID:** BOOK\_03 |
| **Stakeholders and goals:**  Airline’s Human Resource Manager – Wants to save cost on hiring extra staff to edit the services booked by customers.  Customer and Travel Agencies – Wants to be able to edit the services that are booked for a flight.  Normal Staff – Wants to be able to edit the services of a customer. | |
| **Description:** User wants to edit services in a booking. | |
| **Actors:** Normal Staff, Customer, Travel Agency | |
| **Trigger:** User chooses the Modify Services option from the main menu. | |
| **Normal flow:**   1. System prompts user for customer’s username. 2. User enters the username of the customer. 3. System displays the current bookings of the customer with the corresponding username in a numbered list. 4. User selects the list number of the booking. 5. System displays the list of services that the customer booked for that booking in a numbered list. 6. User enters a whether they want to add, change or delete a service.    1. Add service – Sub-flow S1    2. Change service – Sub-flow S2    3. Delete service – Sub-flow S3 7. System returns to the main menu. 8. End | |
| **Sub-flows:**  S1: Add service   1. System displays a list of services currently available in a numbered list. 2. User enters numbers of the corresponding services. 3. System accepts input and saves the addition of service to database.   S2: Change service   1. User selects the number of the corresponding service that is currently booked. 2. System displays a list of services currently available in a numbered list. 3. User enters numbers of the corresponding services. 4. System accepts input and saves the change of service to database.   S3: Delete service   1. User selects the number of the corresponding service that is currently booked. 2. System accepts input and saves the deletion of service to database. | |
| **Alternative/Exceptional flows:** None | |

## Use Case: Add Services

|  |  |
| --- | --- |
| **Name:** Add Services | **ID:** SERV\_01 |
| **Stakeholders and goals:**  Service System Manager – Want to add services that customer can book for a flight. | |
| **Description:** User wants to add services to the current list of services so that customers can have a larger variety of services to choose from. | |
| **Actors:** Service System Manager | |
| **Trigger:** User chooses the Add Services option from the main menu. | |
| **Normal flow:**   1. System prompts user for the service name, service cost and availability. 2. User enters the required details. 3. System prompts user if there is another service to add. 4. If user wants to add another service, repeat steps 1-3; else go to step 5. 5. System saves all added services to database. 6. System returns to the main menu. 7. End | |
| **Sub-flows:** None | |
| **Alternative/Exceptional flows:** None | |

## Use Case: Remove Services

|  |  |
| --- | --- |
| **Name:** Remove Services | **ID:** SERV\_02 |
| **Stakeholders and goals:**  Service System Manager – Want to remove services that customer can book for a flight. | |
| **Description:** User wants to remove services from the current list of services. | |
| **Actors:** Service System Manager | |
| **Trigger:** User chooses the Remove Services option from the main menu. | |
| **Normal flow:**   1. System displays all services in a numbered list. 2. User selects the list numbers of the services to remove. 3. System removes all selected services in the database. 4. System returns to the main menu. 5. End | |
| **Sub-flows:** None | |
| **Alternative/Exceptional flows:** None | |

## Use Case: Edit Service Price

|  |  |
| --- | --- |
| **Name:** Edit Service Price | **ID:** SERV\_03 |
| **Stakeholders and goals:**  Service System Manager – Want to edit prices of the services that are saved in the system. | |
| **Description:** User wants edit the prices of existing services. | |
| **Actors:** Service System Manager | |
| **Trigger:** User chooses the Edit Service Price option from the main menu. | |
| **Normal flow:**   1. System displays all services in a numbered list. 2. User selects the list number of the service to adjust its price. 3. System prompts the user for the new price for the selected service. 4. User enters the new price for the selected service. 5. System saves the new price of the service to the database. 6. System prompts user whether they want to adjust the price of another service. 7. If the user wants to adjust the price of another service, then repeat steps 1-6; else go to step 8. 8. System returns to the main menu. 9. End | |
| **Sub-flows:** None | |
| **Alternative/Exceptional flows:** None | |

## Use Case: Edit Service Availability

|  |  |
| --- | --- |
| **Name:** Edit Service Availability | **ID:** SERV\_04 |
| **Stakeholders and goals:**  Service System Manager – Want to edit the availability of the services for different types of flights. | |
| **Description:** User wants to edit the availability of services to “international” or “all” flights. | |
| **Actors:** Service System Manager | |
| **Trigger:** User chooses the Adjust Service Availability option from the main menu. | |
| **Normal flow:**   1. System displays all services in a numbered list. 2. User selects the list number of the service to adjust availability. 3. System prompts user to input the availability of the selected service. 4. User enteres the new availability for the selected service. 5. System saves the new availability of the service to the database. 6. System prompts the user whether they want to adjust the availability of another service. 7. If the user wants to adjust the availability of another service, then repeat steps 1-6; else go to step 8. 8. System returns to the main menu. 9. End | |
| **Sub-flows:** None | |
| **Alternative/Exceptional flows:** None | |

## Use Case: Edit Routes

|  |  |
| --- | --- |
| **Name:** Edit Routes | **ID:** FLI\_01 |
| **Stakeholders and goals:**  Flight Manager – Want to edit the routes that a flight can take. | |
| **Description:** User wants to add, modify or delete the routes that a flight can take which include the origin airport and the destination airport of the route. | |
| **Actors:** Flight Manager | |
| **Trigger:** User chooses the Edit Routes option from the main menu. | |
| **Normal flow:**   1. System prompts user if they want to add, modify or delete route. 2. System displays the appropriate flow according to the user’s choice.    1. Add route – Sub-flow S1    2. Edit route – Sub-flow S2    3. Delete route – Sub-flow S3 3. System returns to main menu. 4. End | |
| **Sub-flows:**  S1: Add route   1. System prompts user for the origin airport and destination airport. 2. User enters the origin and destination airport. 3. If route already exists, system displays an error message and go to step 1; else go to step 4. 4. System prompts user for codeshare and number of stops. 5. User enters codeshare and number of stops for the route. 6. System saves the new route to database.   S2: Edit route   1. System prompts user for the origin airport and destination airport. 2. User enters the origin and destination airport. 3. If route is not found, display an error message and go to step 1; else go to step 4. 4. System displays the origin airport, destination airport, codeshare and number of stops in a numbered list and prompts user to choose which detail to edit. 5. User enters which detail they want to edit. 6. System prompts user for new detail. 7. User enters new detail. 8. If the new detail is an airport code and it does not exist in the database, system displays an error message and go to step 4; else go to step 9. 9. If the route already exists, system displays an error message and go to step 4; else go to step 10. 10. System updates the database with the new route information.   S3: Delete route   1. System prompts user for the origin airport and destination airport. 2. User enters the origin and destination airport. 3. If route is not found, system displays an error message and go to step 1; else go to step 4. 4. System deletes route from the database. | |
| **Alternative/Exceptional flows:** None | |

## Use Case: Edit Fleet

|  |  |
| --- | --- |
| **Name:** Edit Fleet | **ID:** FLI\_03 |
| **Stakeholders and goals:**  Flight Manager – Want to edit the planes that are curently in the system. | |
| **Description:** User wants to add, modify or delete the planes. | |
| **Actors:** Flight Manager | |
| **Trigger:** User chooses the Edit Fleet option from the main menu. | |
| **Normal flow:**   1. System prompts user if they want to add, modify or delete plane. 2. System displays the appropriate flow according to the user’s choice.    1. Add plane – Sub-flow S1    2. Edit plane – Sub-flow S2    3. Delete plane – Sub-flow S3 3. System returns to main menu. 4. End | |
| **Sub-flows:**  S1: Add plane   1. System prompts user for the plane details. 2. User enters the plane details into the system. 3. If the system detects a mismatch in the input with the expected input, system displays error message and go to step 1; else go to step 4. 4. System saves the new plane to database.   S2: Edit plane   1. System prompts user for the plane details. 2. User enters the aircraft type into the system. 3. If the system is not able to find the aircraft type entered, system displays error message and go to step 1; else go to step 4. 4. System displays seat types and the number of seats for that type in a numbered list, and prompts the user for 5. User selects the seat type by the number. 6. System prompts user for seat addition or remove. (Positive number for adding seats and negative number for removing seats) 7. If the number of seats to remove is less than the actual number, system displays an error message and go to step 4; else go to step 8. 8. System saves the modified seats to database.   S3: Delete plane   1. System prompts user for the plane details. 2. User enters the plane details into the system. 3. If the system detects a mismatch in the input with the expected input, system displays error message and go to step 1; else go to step 4. 4. If the system detects that the number of seats entered by the user is greater than that number stored in the database, system displays an error message and go to step 1; else go to step 5. 5. System deletes the plane from the database. | |
| **Alternative/Exceptional flows:** None | |

## Use Case: Edit Flight Schedule

|  |  |
| --- | --- |
| **Name:** Edit Flight Schedule | **ID:** FLI\_04 |
| **Stakeholders and goals:**  Flight Manager – Want to edit the flights in the system. | |
| **Description:** User wants to add, modify or delete flights. | |
| **Actors:** Flight Manager | |
| **Trigger:** User chooses the Edit Flight Schedule option from the main menu. | |
| **Normal flow:**   1. System prompts user if they want to add, modify or delete flight. 2. System displays the appropriate flow according to the user’s choice.    1. Add flight – Sub-flow S1    2. Edit flight – Sub-flow S2    3. Delete flight – Sub-flow S3 3. System returns to main menu. 4. End | |
| **Sub-flows:**  S1: Add flight   1. System prompts user for the flight details. 2. User enters the flight details into the system. 3. If the system detects a mismatch in the input with the expected input, system displays error message and go to step 1; else go to step 4. 4. System saves the new flight to database.   S2: Edit flight   1. System prompts user for the flight ID, source airport and destination airport. 2. User enters the flight ID, source airport and destination airport into the system. 3. If the system cannot find the flight with the corresponding flight ID, source airport and destination airport, system displays error message and returns to step 1; else go to step 4. 4. System displays all the flight details except for the flight ID in a numbered list and system prompts user to choose which detail they want to modify. 5. User enters the detail to modify by the associated number. 6. System prompts user for the updated detail. 7. User enters updated detail into the system. 8. If the system detects an input mismatch, system displays error message and go to step 4; else go to step 9. 9. System saves modifications to database.   S3: Delete flight   1. System prompts user for the flight ID, source airport and destination airport. 2. User enters the flight ID, source airport and destination airport into the system. 3. If the system cannot find the flight with the corresponding flight ID, source airport and destination airport, system displays error message and returns to step 1; else go to step 4. 4. System deletes the flight from the database. | |
| **Alternative/Exceptional flows:** None | |

## Use Case: Edit Airports

|  |  |
| --- | --- |
| **Name:** Edit Airports | **ID:** FLI\_02 |
| **Stakeholders and goals:**  !!!Flight Manager – Want to edit the flights in the system. | |
| **Description:** User wants to add, modify or delete airports. | |
| **Actors:** Flight Manager | |
| **Trigger:** User chooses the Edit Airport option from the main menu. | |
| **Normal flow:**   1. System prompts user if they want to add, modify or delete aiport. 2. System displays the appropriate flow according to the user’s choice.    1. Add airport – Sub-flow S1    2. Edit flight – Sub-flow S2    3. Delete flight – Sub-flow S3 3. System returns to main menu. 4. End | |
| **Sub-flows:**  S1: Add airport   1. System prompts user for the airport details. 2. User enters the airport details into the system. 3. If the system detects a mismatch in the input with the expected input, system displays error message and returns to step 1; else go to step 4. 4. If the system detects a duplicate airport ID or airport name in the database, system displays an error and returns to step 1; else go to step 5. 5. System saves the new airport to database.   S2: Edit airport   1. System prompts user for the airport ID. 2. User enters the airport ID into the system. 3. If the system cannot find the airport with the corresponding airport ID, system displays error message and returns to step 1; else go to step 4. 4. System displays all the airport details except for the airport ID in a numbered list and system prompts user to choose which detail they want to modify. 5. User enters the detail to modify by the associated number. 6. System prompts user for the updated detail. 7. User enters updated detail into the system. 8. If the system detects an input mismatch, system displays error message and returns to step 4; else go to step 9. 9. System saves modifications to database.   S3: Delete airport   1. System prompts user for the aiport ID. 2. User enters the airport ID into the system. 3. If the system cannot find the airport with the corresponding airport ID, system displays error message and returns to step 1; else go to step 4. 4. System deletes the airport from the database. | |
| **Alternative/Exceptional flows:** None | |

## Use Case: Move Passengers Between Flights

|  |  |
| --- | --- |
| **Name:** Move Passengers Between Flights | **ID:** RSV\_01 |
| **Stakeholders and goals:**  !!!Flight Manager – Want to edit the flights in the system. | |
| **Description:** User wants to move passengers between flights. | |
| **Actors:** Reservation System Manager | |
| **Trigger:** User chooses the Move Passengers Between Flights option from the main menu. | |
| **Normal flow:**   1. Perform the *Enter Flight and Customer Details* use case. 2. System prompts user for the flight ID, source airport and detination airport of the flight to transfer customers to. 3. User enters flight ID, source airport and destination airport into the system. 4. For each customer, system prompts user to enter the seat number for the customer. 5. System saves the changes to database. 6. System returns to main menu. 7. End | |
| **Sub-flows:** None | |
| **Alternative/Exceptional flows:**  3a User enters invalid flight: The *Enter Invalid Flight* use case is performed. Go to step 4 after execution.  4a User enters invalid seat number: The *Enter Invalid Seat Number* is performed. Redo step 4 after execution. | |

## Use Case: Set Ticket Prices

|  |  |
| --- | --- |
| **Name:** Set Ticket Prices | **ID:** RSV\_03 |
| **Stakeholders and goals:**  !!!Flight Manager – Want to edit the flights in the system. | |
| **Description:** User wants to set the prices of tickets for flights. | |
| **Actors:** Reservation System Manager | |
| **Trigger:** User chooses the Set Ticket Prices option from the main menu. | |
| **Normal flow:**   1. System prompts user for the flight ID, source airport and detination airport. 2. User enters flight ID, source airport and detination airport into the system. 3. System displays current price and prompts user for new price. 4. User enters new price. 5. System saves updated price to the database. 6. System returns to main menu. 7. End | |
| **Sub-flows:** None | |
| **Alternative/Exceptional flows:**  2a User enters invalid flight: The *Enter Invalid Flight* use case is performed. Go to step 1 after execution.  4a User enters invalid price: The *Enter Invalid Price* use case is performed. Go to step 3 after execution. | |

## Use Case: Change Passenger Seating

|  |  |
| --- | --- |
| **Name:** Change Passenger Seating | **ID:** RSV\_02 |
| **Stakeholders and goals:**  !!!Flight Manager – Want to edit the flights in the system. | |
| **Description:** User wants to move passengers within a flight. | |
| **Actors:** Reservation System Manager | |
| **Trigger:** User chooses the Change Passenger Seat option from the main menu. | |
| **Normal flow:**   1. Perform the *Enter Flight and Customer Details* use case. 2. For each customer, system prompts user to enter the seat number for the customer. 3. System saves the changes to database. 4. System returns to main menu. 5. End | |
| **Sub-flows:** None | |
| **Alternative/Exceptional flows:**  2a User enters invalid seat number: The *Enter Invalid Seat Number* is performed. Redo step 4 after execution. | |

## Use Case: Edit Watch and No Fly List

|  |  |
| --- | --- |
| **Name:** Edit Watch and No Fly List | **ID:** ACC\_06 |
| **Stakeholders and goals:**  !!!Flight Manager – Want to edit the flights in the system. | |
| **Description:** The user wants to edit the watch and no fly list. | |
| **Actors:** Reservation System Manager, Profile System Manager | |
| **Trigger:** User chooses the Edit Watch and No Fly List option from the main menu. | |
| **Normal flow:**   1. System prompts user for the username of a customer. 2. User enters the username of a customer. 3. System displays the customer’s present flying status as recorded in the system and prompts the user for the new flying status. 4. User enters the new flying status for the customer. 5. System saves the changes to database. 6. System returns to main menu. 7. End | |
| **Sub-flows:** None | |
| **Alternative/Exceptional flows:**  2a User enters invalid username: The *Enter Invalid Username* use case is performed. Go to step 1 after execution. | |

## Use Case: Enter Invalid Flight

|  |  |
| --- | --- |
| **Name:** Enter Invalid Flight | **ID:** EX\_05 |
| **Stakeholders and goals:**  !!!Flight Manager – Want to edit the flights in the system. | |
| **Description:** User has entered an invalid flight and an error message is displayed. | |
| **Actors:** Reservation System Manager | |
| **Trigger:** User has entered details of a flight that are invalid or do not exist in the system. | |
| **Normal flow:**   1. System detects that the details of thebet entered flight are invalid or do not exist in the system. 2. System displays an error message. 3. User acknowledges the error message. 4. End | |
| **Sub-flows:** None | |
| **Alternative/Exceptional flows:** None | |

## Use Case: Enter Invalid Price

|  |  |
| --- | --- |
| **Name:** Enter Invalid Price | **ID:** EX\_06 |
| **Stakeholders and goals:**  !!!Flight Manager – Want to edit the flights in the system. | |
| **Description:** User has entered an invalid price and an error message is displayed. | |
| **Actors:** Reservation System Manager | |
| **Trigger:** User has entered a ticket price that is invalid. | |
| **Normal flow:**   1. System detects that the details of the entered ticket price is invalid. 2. System displays an error message. 3. User acknowledges the error message. 4. End | |
| **Sub-flows:** None | |
| **Alternative/Exceptional flows:** None | |

## Use Case: Enter Invalid Seat Number

|  |  |
| --- | --- |
| **Name:** Enter Invalid Seat Number | **ID:** EX\_01 |
| **Stakeholders and goals:**  !!!Flight Manager – Want to edit the flights in the system. | |
| **Description:** User has entered an invalid seat number and an error message is displayed. | |
| **Actors:** Reservation System Manager, Customer, Travel Agency | |
| **Trigger:** User has entered details of an airport that are invalid or do not exist in the system. | |
| **Normal flow:**   1. System detects that the details of the entered airport are invalid or do not exist in the system. 2. System displays an error message. 3. User acknowledges the error message. 4. End | |
| **Sub-flows:** None | |
| **Alternative/Exceptional flows:** None | |

## Use Case: Enter Flight and Customer Details

|  |  |
| --- | --- |
| **Name:** Enter Flight and Customer Details | **ID:** RSV\_01 |
| **Stakeholders and goals:**  !!!Flight Manager – Want to edit the flights in the system. | |
| **Description:** User wants to enter a flight and select the customers that are involved in that flight. | |
| **Actors:** Reservation System Manager | |
| **Trigger:** User has entered details of an airport that are invalid or do not exist in the system. | |
| **Normal flow:**   1. System prompts user for the flight ID, source airport and the destination airport of a flight. 2. User enters the flight ID, source airport and the destination airport of a flight. 3. System displays all customers that booked the entered flight with the customer’s username, first name and last name. System prompts user to enter the customer usernames that they want to select. 4. User enters the usernames of the customers that they want to select. 5. End | |
| **Sub-flows:** None | |
| **Alternative/Exceptional flows:**  2a User enters invalid flight details: The *Enter Invalid Flight* use case is performed. Go to step 1 after execution.  4a User enters invalid username: The *Enter Invalid Username* use case is performed. Go to step 3 after execution. | |

## Use Case: Edit Travel Agency and Customer Account

|  |  |
| --- | --- |
| **Name:** Edit Travel Agency and Customer Account | **ID:** RSV\_02 |
| **Stakeholders and goals:**  !!!Flight Manager – Want to edit the flights in the system. | |
| **Description:** User wants to edit the travel agency and customer account. | |
| **Actors:** Profile System Manager | |
| **Trigger:** User chooses the Edit Travel Agency and Customer Account option from the main menu. | |
| **Normal flow:**   1. System prompts the user to enter a username of a travel agency or a customer. 2. User enters a username into the system. 3. System displays details of the travel agency or customer that matches the corresponding username in a numbered list and prompts the user to select which detail that they want to edit. 4. User enters the number corresponding to the detail that they want to edit. 5. System prompts user for the new detail. 6. User enters the new detail into the system. 7. System updates the customer or travel agency’s details in the database. 8. System returns to main menu. 9. End | |
| **Sub-flows:** None | |
| **Alternative/Exceptional flows:**  2a User enters invalid username: The *Enter Invalid Username* use case is performed. Go to step 1 after execution. | |

## Use Case: Close Account

|  |  |
| --- | --- |
| **Name:** Close Account | **ID:** ACC\_04 |
| **Stakeholders and goals:**  !!!Flight Manager – Want to edit the flights in the system. | |
| **Description:** User wants to close a travel agency or customer account. | |
| **Actors:** Profile System Manager, Customer, Travel Agency | |
| **Trigger:** User chooses the Close Account option from the main menu. | |
| **Normal flow:**   1. If the user is a Profile System Manager, system prompts the user to enter a username of a travel agency or a customer; else go to step 4. 2. User enters a username into the system. 3. System displays details of the travel agency or customer that matches the corresponding username. 4. System prompts the user to confirm whether want to delete the account. 5. User enters their confirmation. 6. System deletes the selected travel agency or customer from the database. 7. System returns to main menu. 8. End | |
| **Sub-flows:** None | |
| **Alternative/Exceptional flows:**  2a User enters invalid username: The *Enter Invalid Username* use case is performed. Go to step 1 after execution. | |

## Use Case: Create Staff Profiles

|  |  |
| --- | --- |
| **Name:** Create Staff Profiles | **ID:** SA\_01 |
| **Stakeholders and goals:**  System administrator - | |
| **Description:** User wants to close a travel agency or customer account. | |
| **Actors:** System Administrator | |
| **Trigger:** User chooses the Close Account option from the main menu. | |
| **Normal flow:**   1. System prompts the user to enter a username of a travel agency or a customer. 2. User enters a username into the system. 3. System displays details of the travel agency or customer that matches the corresponding username and prompts the user to confirm whether want to delete the selected travel agency or customer. 4. User enters their confirmation. 5. System deletes the selected travel agency or customer from the database. 6. System returns to main menu. 7. End | |
| **Sub-flows:** None | |
| **Alternative/Exceptional flows:**  2a User enters invalid username: The *Enter Invalid Username* use case is performed. Go to step 1 after execution. | |

## Use Case: Edit Staff Profiles

|  |  |
| --- | --- |
| **Name:** Edit Staff Profiles | **ID:** SA\_02 |
| **Stakeholders and goals:** | |
| **Description:** | |
| **Actors:** System Administrator | |
| **Trigger:**. | |
| **Normal flow:**   1. End | |
| **Sub-flows:** None | |
| **Alternative/Exceptional flows:** | |

## Use Case: Change Password

|  |  |
| --- | --- |
| **Name:** Change Password | **ID:** ACC\_03 |
| **Stakeholders and goals:**  s | |
| **Description:** | |
| **Actors:** Normal Staff | |
| **Trigger:**. | |
| **Normal flow:**   1. End | |
| **Sub-flows:** None | |
| **Alternative/Exceptional flows:** | |