Software Requirements Specification

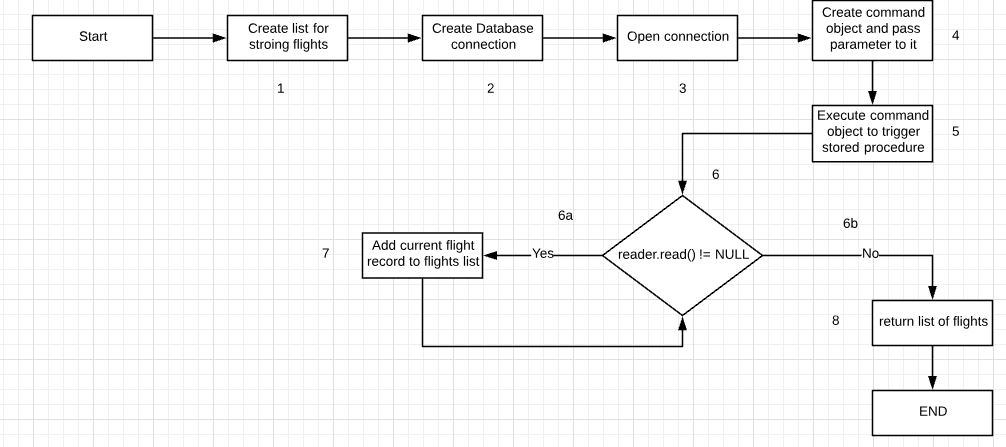
Version 3.0

AirBook

Team # 1

|  |  |  |
| --- | --- | --- |
| Member Name | Member Roll # | Primary Responsibility |
| Omer Iftikhar | 15-4055 | Testing one method |
| Sharjeel Tariq | 15-4008 | Testing one method |
| Habibah Shahid | 14-4150 | Testing one method |
| Uzair Ahmad | 14-4062 | Testing one method |
| Saif Ullah | 15-4091 | Testing one method |

|  |  |
| --- | --- |
| **Identifier** | User - SearchFlights |
| **Priority** | High |
| **Short description** | It returns list of flight according to the input given by the user. |
| **Pre-condition(s)** | User must be logged in. |
| **Input data** | Origin, Destination, Date, Departure Time, Economy class seats, Business class seats, First class seats. |
| **Detailed steps** | User fills the form on the screen to search for flights that match his/her needs. |
| **Expected result(s)** | Flight appears in the list of available flights for the user. |
| **Post-condition(s)** | None |



**Test case:**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **S#** | **Test Data** | **Expected Result** | **Actual Result** | **Branch path covered** | **Status (Pass/Fail)** |
| 1 | Origin=”Lahore”  Destination=”Multan”  Date=”21/03/2019”  Departure Time =”08:00”  Economy class seats=3  Business class seats=1  First class seats=0 | List of flights matching the input criteria | Flights matching the input criteria | 1 -> 2 -> 3 -> 4 -> 5 -> 6-> 6a -> 7 -> 6b -> 8 | Pass |

**Branch Coverage**:   
Branches covered/Total Branches= 2/2=100%  
  
**Path Coverage:**   
Paths covered/Total Paths=2/2=100%  
  
**Statement Coverage**:   
Statements covered/Total Statements=8/8=100%  
  
**Loop Boundary Testing**:   
No such loop exists whose boundary is to be tested. If flight data from database is empty then simply program will go to return flight list with list size of 0 else it will loop through the data and keep on adding it to flight list until entire data list is exhausted.

|  |  |
| --- | --- |
| **Identifier** | TC-1 SearchFlightsById |
| **Priority** | High |
| **Short description** | This function is primarily a helper function for the User, Admin and Manager. When searching up flight from the database by the admin or manager, this function is used to get the flight details from the database by using it ID as an identifier. Thus, once the flight object is returned from the database, it can then be used to present it inform of understandable information. |
| **Pre-condition(s)** | 1. Know the ID of the flight to be searched. 2. Know the data on which the flight was created. |
| **Input data** | 1. FlightID 2. FlightDate |
| **Detailed steps** | The function follows the following steps in getting the data from the database.   * Establish connection with the database. * Connection is not Established with database. |
| **Expected result(s)** | Null is returned |
| **Post-condition(s)** | Flight Not Found |

**Path:** 1A-11C-3

**Statement Coverage:** 27%

**Branch Coverage:** 50%

**Loop Testing:** No Loops in this Path

|  |  |
| --- | --- |
| **Identifier** | TC-2 SearchFlightsById |
| **Priority** | High |
| **Short description** | This function is primarily a helper function for the User, Admin and Manager. When searching up flight from the database by the admin or manager, this function is used to get the flight details from the database by using it ID as an identifier. Thus, once the flight object is returned from the database, it can then be used to present it inform of understandable information. |
| **Pre-condition(s)** | 1. Know the ID of the flight to be searched. 2. Know the data on which the flight was created. |
| **Input data** | 1. FlightID 2. FlightDate |
| **Detailed steps** | The function follows the following steps in getting the data from the database.   * Establish connection with the database. * Connection is Established with database. * Database does not contain the data |
| **Expected result(s)** | Null is returned |
| **Post-condition(s)** | Flight Not Found |

**Path:** 1A-11B-2D-4E-5F-6G-7H

**Statement Coverage:** 72%

**Branch Coverage:** 100%

**Loop Testing:** No Loops in this Path

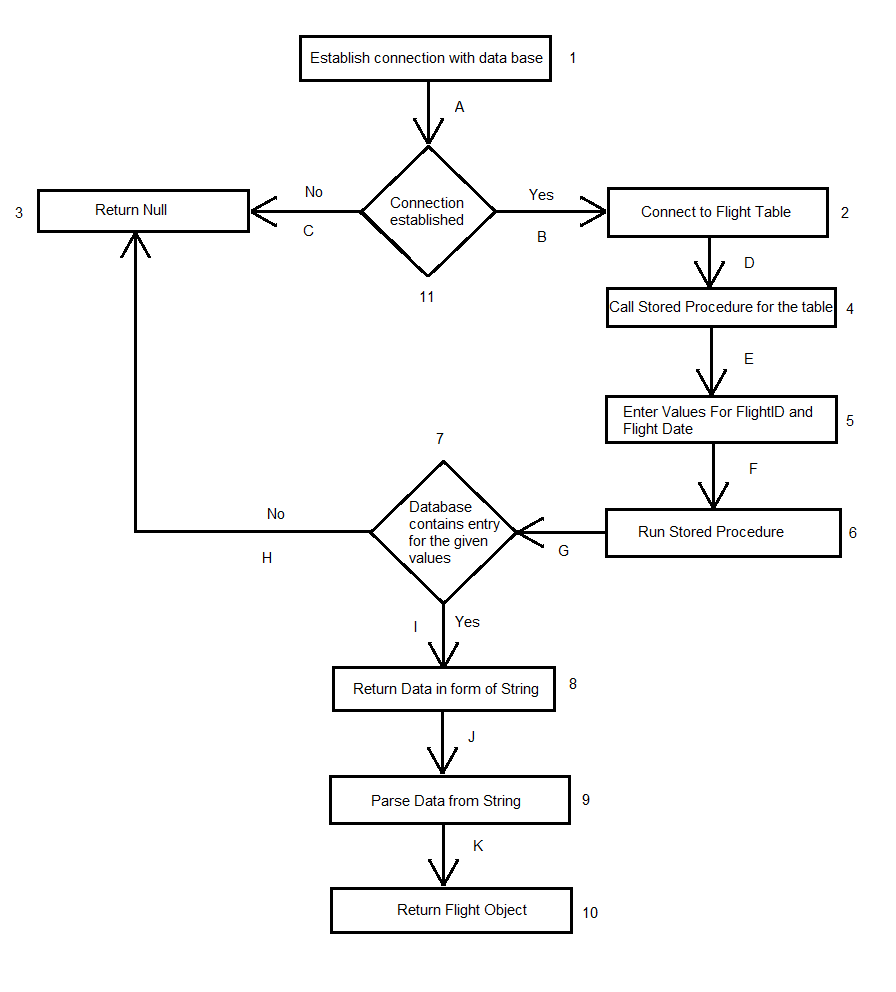
|  |  |
| --- | --- |
| **Identifier** | TC-3 SearchFlightsById |
| **Priority** | High |
| **Short description** | This function is primarily a helper function for the User, Admin and Manager. When searching up flight from the database by the admin or manager, this function is used to get the flight details from the database by using it ID as an identifier. Thus, once the flight object is returned from the database, it can then be used to present it inform of understandable information. |
| **Pre-condition(s)** | 1. Know the ID of the flight to be searched. 2. Know the data on which the flight was created. |
| **Input data** | 1. FlightID 2. FlightDate |
| **Detailed steps** | The function follows the following steps in getting the data from the database.   * Establish connection with the database. * Connection is Established with database. * Database contains the data * The database returns the data in form of a String. * The string is parsed * The data from the string is then used to create a FLIGHT object * That object is then returned. |
| **Expected result(s)** | Flight is returned |
| **Post-condition(s)** | Flight is Found and Returned for further functions. |

**Path:** 1A-11B-2D-4E-5F-6G-7I-8J-9K-10

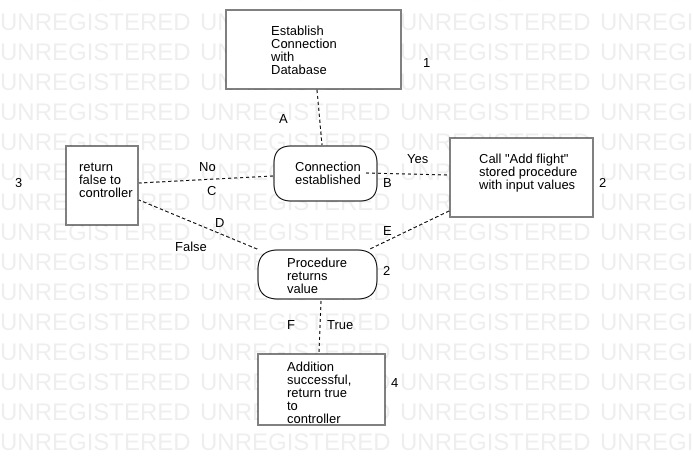
**Statement Coverage:** 90%

**Branch Coverage:** 100%

**Loop Testing:** No Loops in this Path



|  |  |
| --- | --- |
| **Identifier** | Admin – Add Flights |
| **Priority** | Medium |
| **Short description** | Admin adds flight data into the system which will be approved by the manager in order to be finalized |
| **Pre-condition(s)** | Admin must Log in |
| **Input data** | Flight name, Flight Departure time and location, Flight arrival time and location, Number of seats (Economy, business and first class) and seat prices (categorical) |
| **Detailed steps** | Admin fills the form on the screen, entering with caution the exact details of the flights, because once entered they cannot be edited, only deleted. The admin must verify the data three times and then post the form. Upon which the admin’s part of the process will be complete. |
| **Expected result(s)** | Flight appears in the list of unapproved flights, accessible by the manager. |
| **Post-condition(s)** | none |

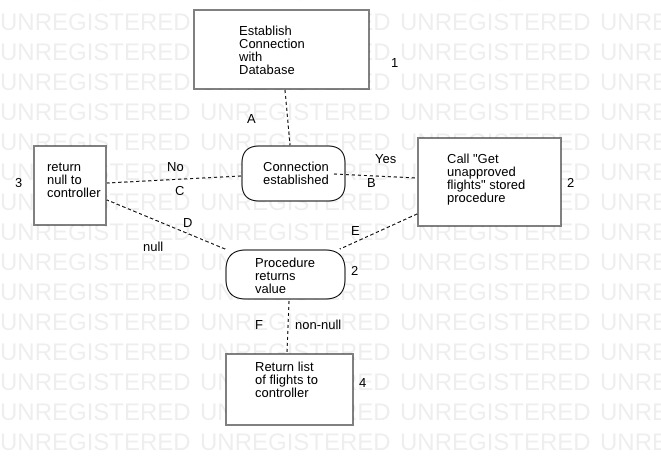


**Test case:**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **S#** | **Test Data** | **Expected Result** | **Actual Result** | **Branch path covered** | **Status (Pass/Fail)** |
| 1 | Origin=”Lahore”  Destination=”Multan”  Date=”21/03/2019”  Departure Time =”08:00”  Economy class seats=30  Business class seats=10  First class seats=20 | Return true | Return true | 1a -> 2e -> 3 -> 4 | Pass |

**Branch Coverage**:   
Branches covered/Total Branches= 1/2=50%  
  
**Path Coverage:**   
Paths covered/Total Paths=1/2=50%  
  
**Statement Coverage**:   
Statements covered/Total Statements=4/5=80%  
  
**Loop Boundary Testing**:   
No such loop exists whose boundary is to be tested.

|  |  |
| --- | --- |
| **Identifier** | Manager – Get Unapproved Flights |
| **Priority** | Medium |
| **Short description** | The manager is displayed a list of flights pending for final approval. |
| **Pre-condition(s)** | Admin must add flight data. |
| **Input data** | Requires no input, general search query. |
| **Detailed steps** | The Manager gets a list of flights on his screen awaiting approval. |
| **Expected result(s)** | Flight appears in manager’s list. |
| **Post-condition(s)** | None. |

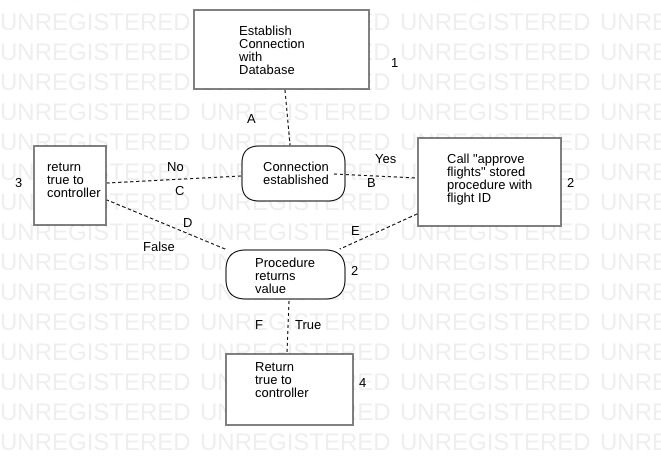


**Test case:**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **S#** | **Test Data** | **Expected Result** | **Actual Result** | **Branch path covered** | **Status (Pass/Fail)** |
| 1 | No input | Return list of unapproved flights | Return list of unapproved flights | 1a -> 2e -> 3 -> 4 | Pass |

**Branch Coverage**:   
Branches covered/Total Branches= 1/2=50%  
  
**Path Coverage:**   
Paths covered/Total Paths=1/2=50%  
  
**Statement Coverage**:   
Statements covered/Total Statements=4/5=80%  
  
**Loop Boundary Testing**:   
No such loop exists whose boundary is to be tested.

|  |  |
| --- | --- |
| **Identifier** | Manager – Approve Flights |
| **Priority** | Medium |
| **Short description** | The manager approves a flight from a list of flights pending for final approval. |
| **Pre-condition(s)** | Admin must add flight data. |
| **Input data** | Confirmation of approval from the manager (at the click of a button) |
| **Detailed steps** | The Manager gets a list of flights awaiting approval, rechecks them and gives them a final go ahead to be ready for booking. |
| **Expected result(s)** | Flight appears in user’s search. |
| **Post-condition(s)** | None. |



**Test case:**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **S#** | **Test Data** | **Expected Result** | **Actual Result** | **Branch path covered** | **Status (Pass/Fail)** |
| 1 | No input | Return true | Return true | 1a -> 2e -> 3 -> 4 | Pass |

**Branch Coverage**:   
Branches covered/Total Branches= 1/2=50%

**Path Coverage:**   
Paths covered/Total Paths=1/2=50%  
  
**Statement Coverage**:   
Statements covered/Total Statements=4/5=80%  
  
**Loop Boundary Testing**:   
No such loop exists whose boundary is to be tested.