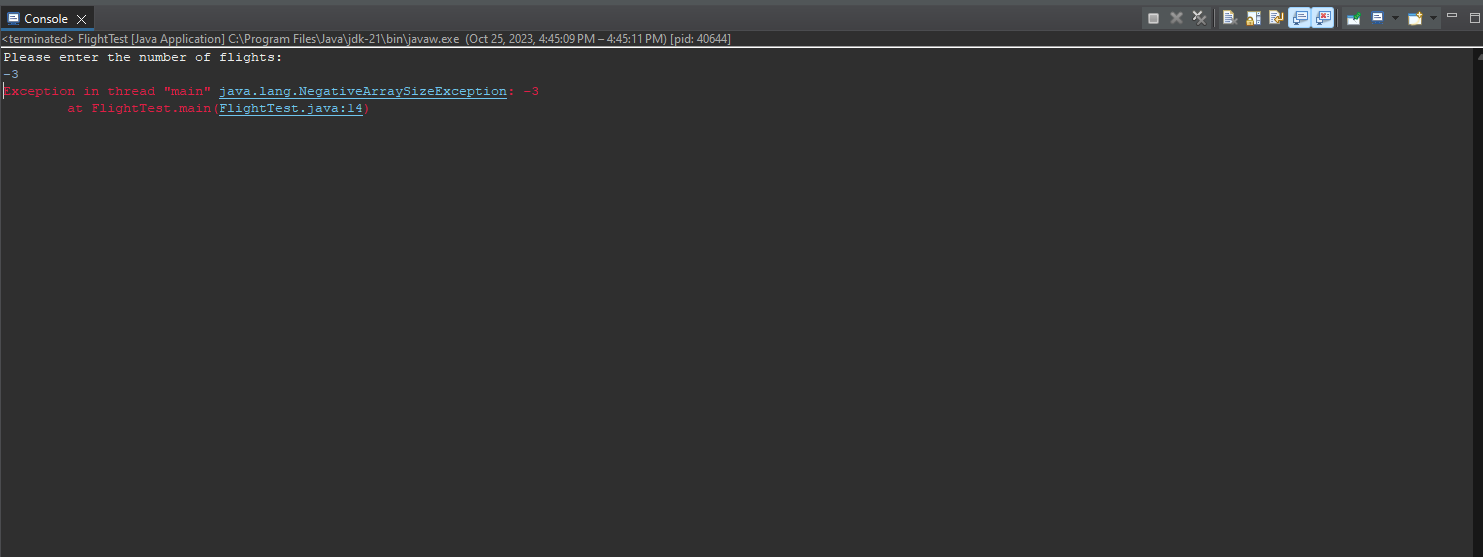
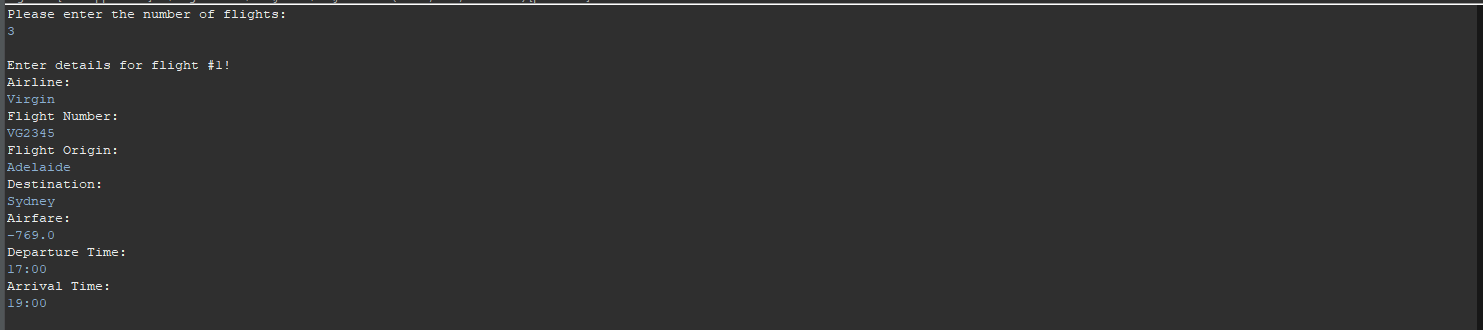
**Test Cases and Results**

# Test Case 1 [User Input Validation]





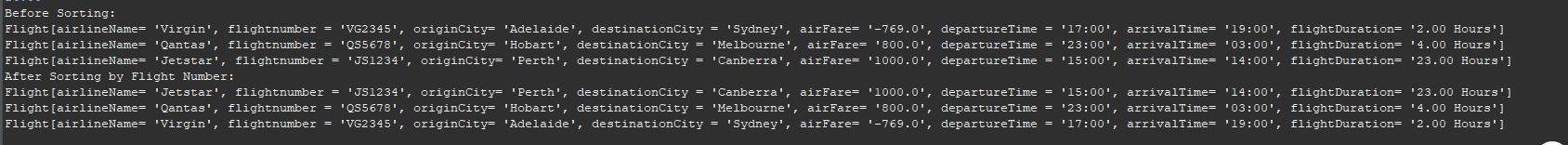
In this test case 1, I was trying to test Input Validation for the number of flights and airfare. As you can see, the program throws an Exception error when I try to put a negative value to the number of flights. However, it didn’t throw any error messages when I input a negative value to the airfare. This indicates that my program only has proper handling for the input of the number of flights but not with the airfare. I’ve also mentioned in my documentation that one of the limitations of my program is proper error handling.

# Test Case 2 [User Input for departure time and arrival time format]



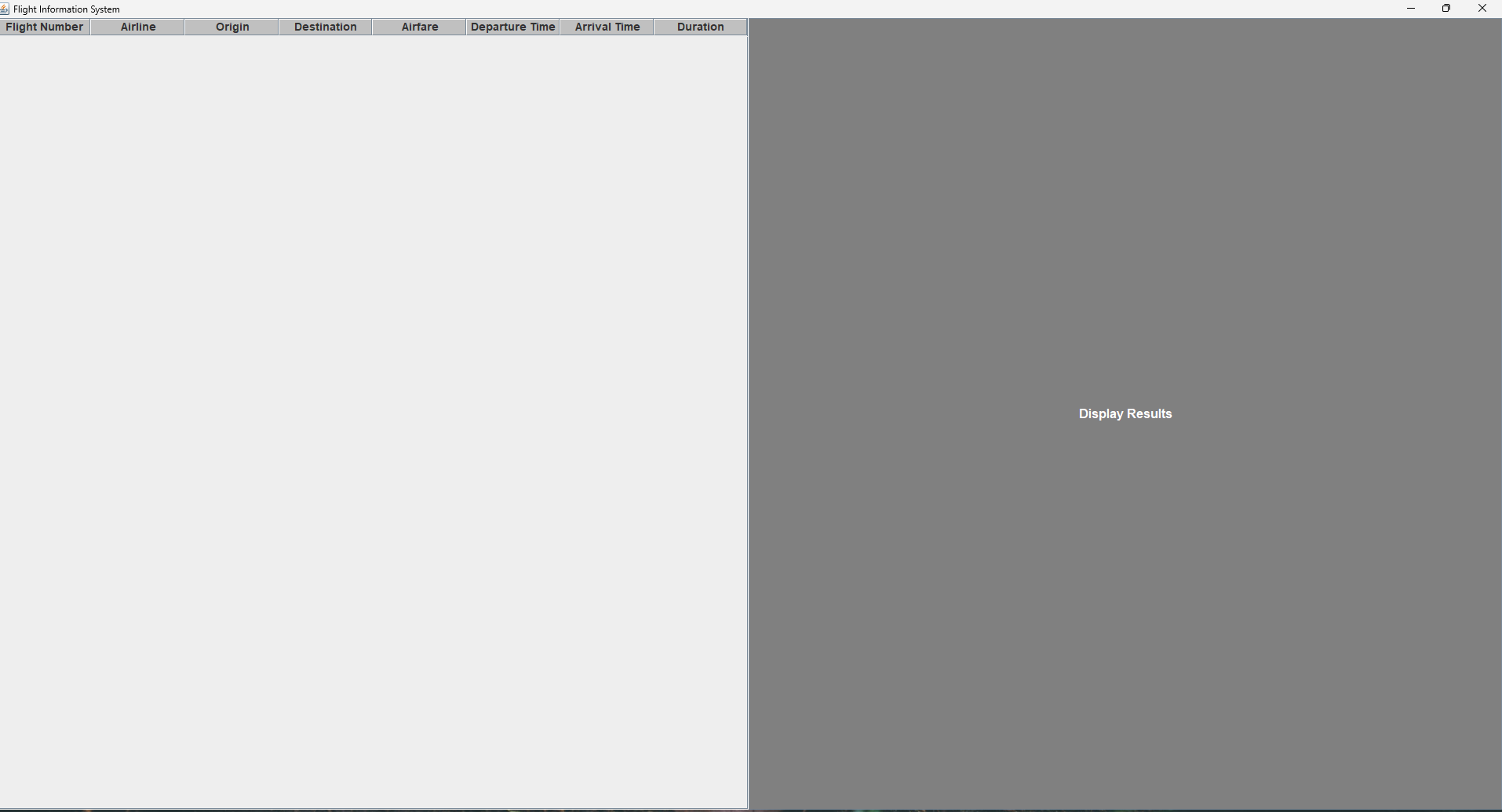
In Test Case 2, I was trying to test if the user’s input for departure time and arrival time varies. You can see clearly that when I try to use the format “HH.mm”, it didn’t work and it throws me an error. The reason of this happening is that because I use this line of code DateTimeFormatter formatter = DateTimeFormatter.ofPattern("HH:mm"); This specific line of code creates a dateTimeFormatter object with a specific pattern. Therefore, it can only accept time input with the format of “HH:mm” to ensure consistent and readable formatting.

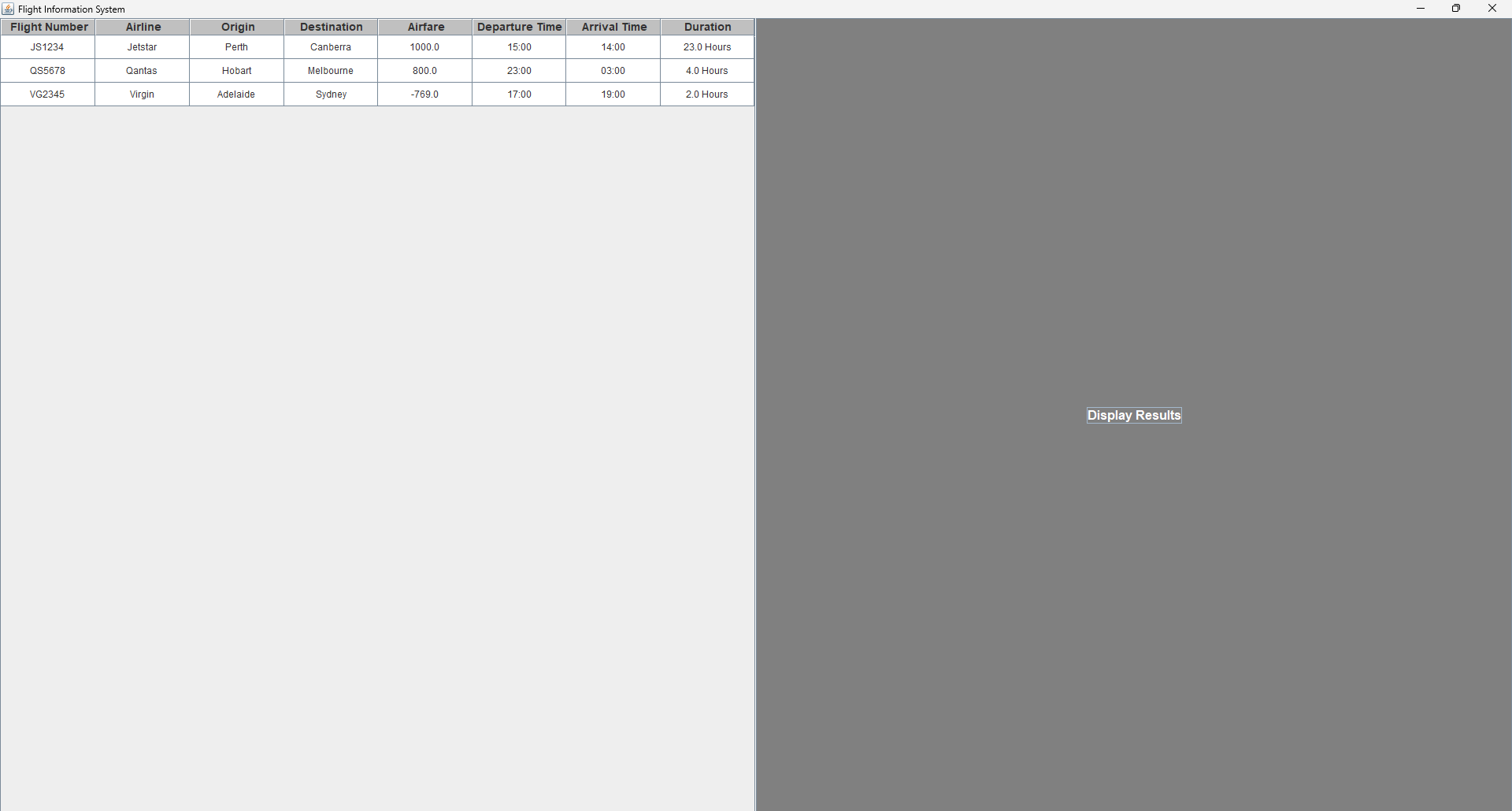
# Test Case 3 [Sorting functionality based on flight numbers]



In Test Case 3, I am trying to ensure that my program can successfully sort the flight details based on their flight numbers using array.sort method and lambda expressions. It is clear that the program works perfectly fine displaying before sort details and after sort details. It is lexicographically correct that “JS1234” comes first before “QS5678” and “VG2345”.

# Test Case 4 [GUI Initialization and Button Click Event]

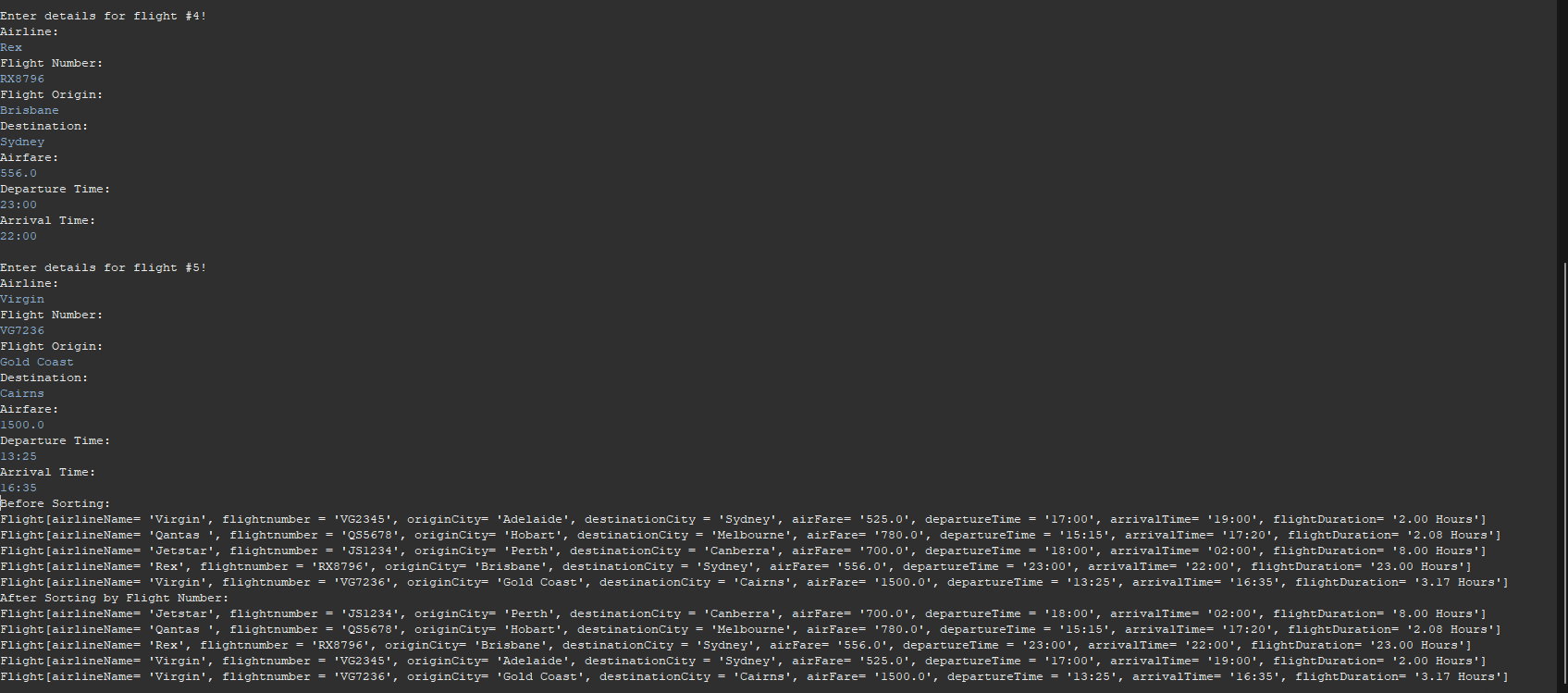


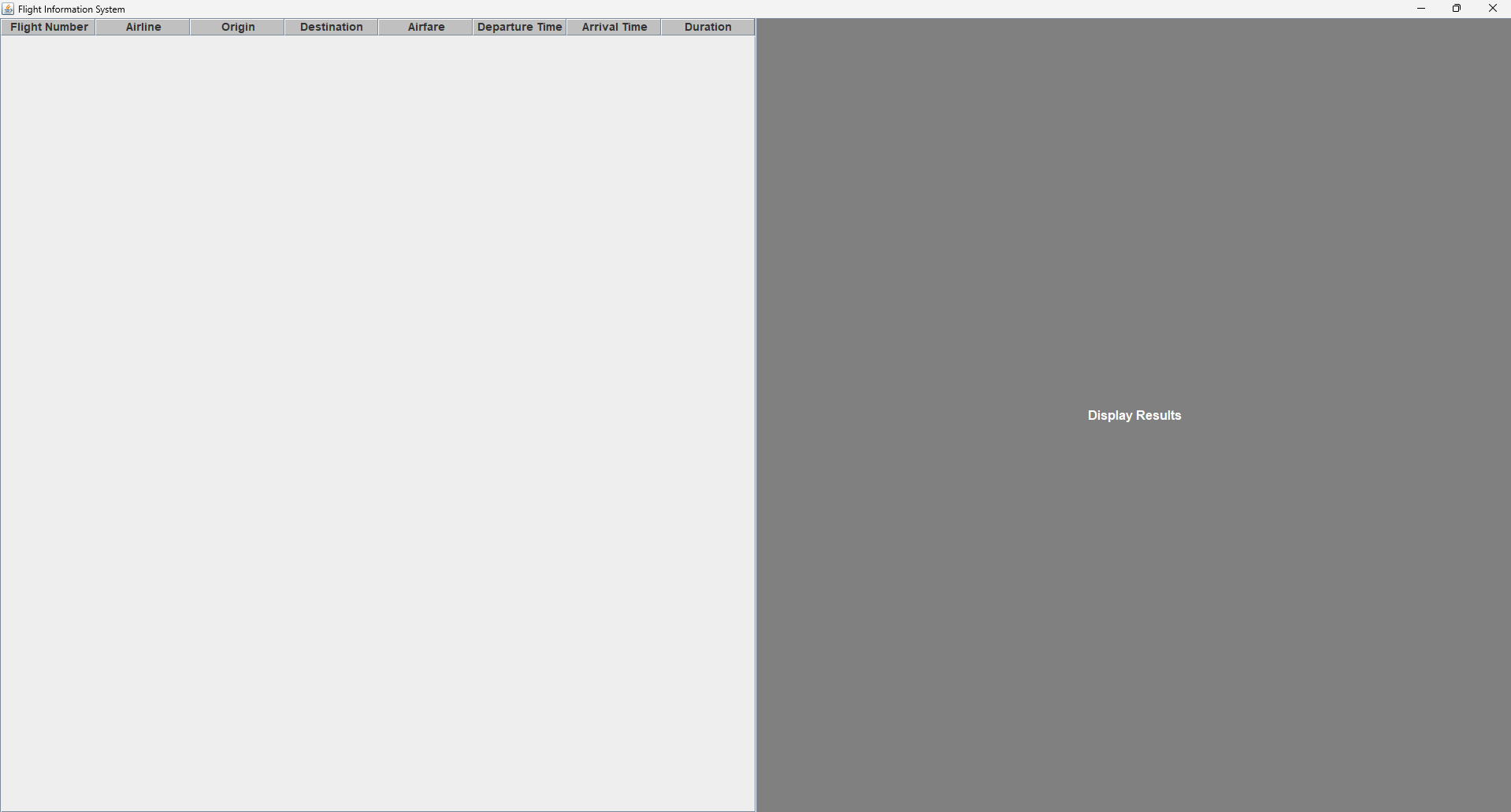


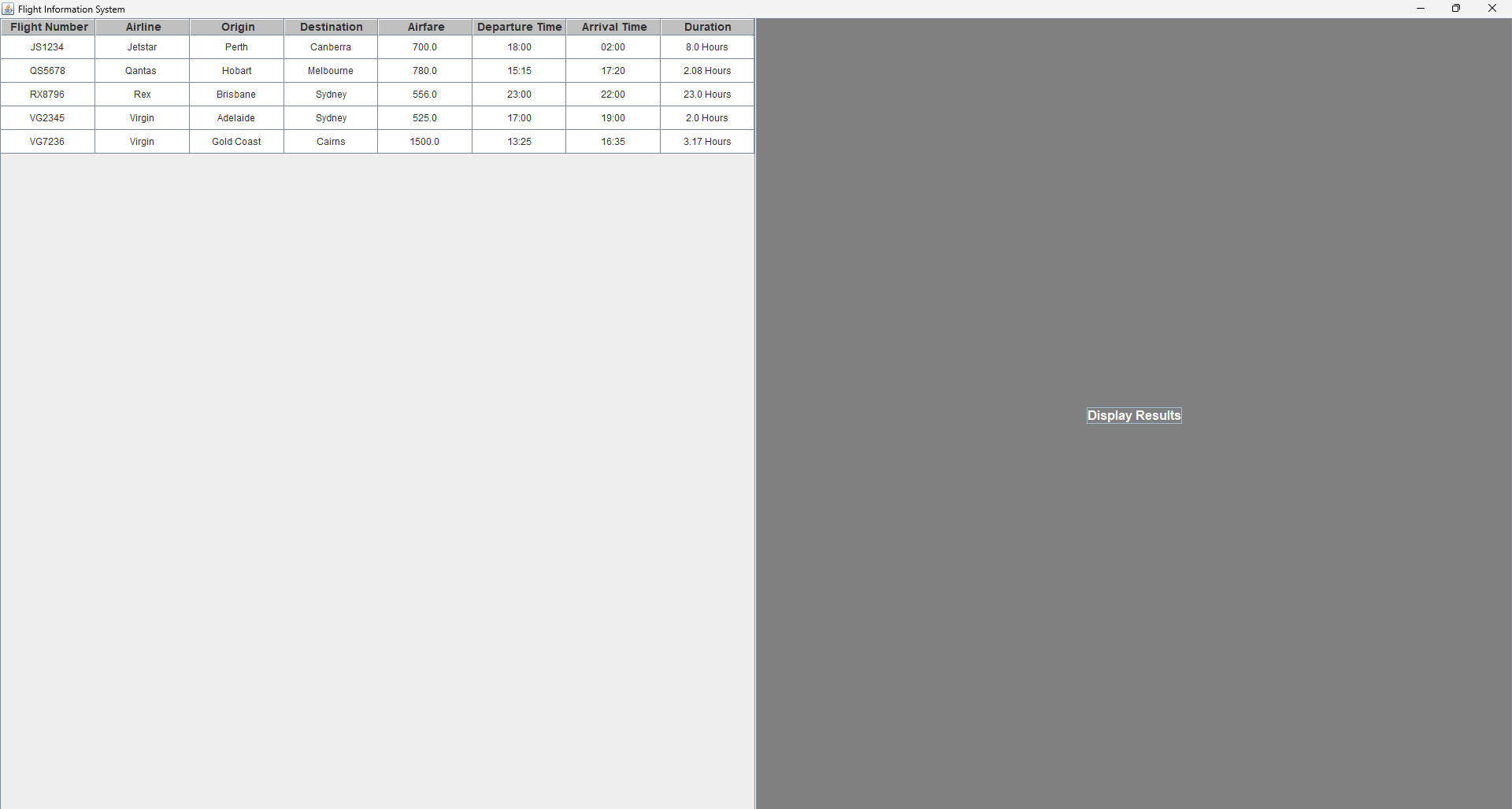
As for Test Case 4, I’m trying to test whether my GUI can be Initialized and my button click event is working or not. In figure 1, you can see that my GUI initialized properly with my desired format indicating that my GUI initialization is successful. Figure 2 is a picture of what happens after I click the “Display Results” button. After I clicked the button, it represented sorted flight details that are center aligned with correct outputs matching the console outputs that you can see in my test case 3.

# Test Case 5 [Valid Inputs with expected outputs]









## Test Cases for Flight Class:

1. Test Case for calculateTotalTravelTime() Method:

Input: Departure Time, Arrival Time✓

Expected Output: Total Travel Time in Hours✓

2. Test Case for Input Validation in Flight Constructor:

Inputs: Invalid Airfare (Negative value), Invalid Departure/Arrival Time (Outside valid time range)✓

Expected Output: Proper error handling, exception, or default values set[Didn’t work for airfare]

3. Test Case for Sorting based on Flight Number:

Input: Array of Flight Objects (Unsorted)✓

Expected Output: Array of Flight Objects (Sorted by Flight Number)✓

## Test Cases for FlightTest Class:

1. Test Case for User Input Validation:

Inputs: Invalid Number of Flights (Negative value), Invalid Airfare (Negative value)✓

Expected Output: Proper error handling, exception, or default values set[Didn’t work for airfare]

2. Test Case for Sorting Functionality:

Input: Array of Flight Objects (Unsorted)✓

Expected Output: Array of Flight Objects (Sorted by Flight Number)✓

3. Test Case for GUI Display:

Input: Array of Flight Objects (Sorted)✓

Expected Output: GUI displaying Flight Information with proper formatting✓

## Test Cases for Form1 Class:

1. Test Case for GUI Initialization:

Input: None✓

Expected Output: GUI Form Initialized Properly✓

2. Test Case for Button Click Event:

Input: Button Click Event Triggered✓

Expected Output: Sorted Flight Information displayed in the GUI Table✓