```java
import java.util.ArrayList;

class Flight {
    int flightId;
    String source, destination;
    int seats;

    Flight(int flightId, String source, String destination, int seats) {
        this.flightId = flightId;
        this.source = source;
        this.destination = destination;
        this.seats = seats;
    }

    @Override
    public String toString() {
        return "Flight ID: " + flightId + ", Source: " + source + ", Destination: " +
    }
}

class Booking {
    int bookingId, userId, flightId;

    Booking(int bookingId, int userId, int flightId) {
        this.bookingId = bookingId;
        this.userId = userId;
        this.flightId = flightId;
    }

    @Override
    public String toString() {
        return "Booking ID: " + bookingId + ", User ID: " + userId + ", Flight ID: " + flightId;
    }
}

public class AirlineSystem {
    private List<Flight> flights = new ArrayList<>();
    private List<Booking> bookings = new ArrayList<>();
    private int bookingIdCounter = 1;

    void addFlight(Flight flight) {
        flights.add(flight);
    }

    void searchFlights(String source, String destination) {
        for (Flight flight : flights) {
            if (flight.source.equalsIgnoreCase(source) &&
                System.out.println(flight);
            }
        }
    }

    void bookFlight(int userId, int flightId) {
        for (Flight flight : flights) {
            if (flight.flightId == flightId && flight.seats > 0) {
                bookings.add(new Booking(bookingIdCounter++, userId, flightId));
                flight.seats--;
                System.out.println("Booking successful! Booking ID: " + (bookingIdCounter - 1));
                return;
            }
        }
```

```java
64            System.out.println("Booking failed. No seats available or invalid flight ID.");
65        }
66
67    void viewBookings() {
68        for (Booking booking : bookings) {
69            System.out.println(booking);
70        }
71    }
72
73    public static void main(String[] args) {
74        AirlineSystem system = new AirlineSystem();
75        Scanner scanner = new Scanner(System.in);
76
77        // Add sample flights
78        system.addFlight(new Flight(1, "New York", "London", 100));
79        system.addFlight(new Flight(2, "London", "Paris", 50));
80
81        // Menu
82        while (true) {
83            System.out.println("\n1. Search Flights\n2. Book Flight\n3. View Bookings\n4.
84            System.out.print("Choose an option: ");
85
86            try {
87                int choice = scanner.nextInt();
88                scanner.nextLine(); // Consume the newline character
89
90                switch (choice) {
91                    case 1:
92                        System.out.print("Enter source: ");
93                        String source = scanner.nextLine();
94                        System.out.print("Enter destination: ");
95                        String destination = scanner.nextLine();
96                        system.searchFlights(source, destination);
97                        break;
98
99                    case 2:
100                       System.out.print("Enter User ID: ");
101                       int userId = scanner.nextInt();
102                       System.out.print("Enter Flight ID: ");
103                       int flightId = scanner.nextInt();
104                       system.bookFlight(userId, flightId);
105                       break;
106
107                   case 3:
108                       system.viewBookings();
109                       break;
110
111                   case 4:
112                       System.out.println("Exiting...");
113                       return;
114
115                   default:
116                       System.out.println("Invalid choice. Try again.");
117                }
118            } catch (java.util.InputMismatchException e) {
119                System.out.println("Invalid input. Please enter a number.");
120                scanner.nextLine(); // Clear the invalid input
121            }
122        }
123    }
124 }
```