EECE 2560 Final Presentation Flight Reservation System

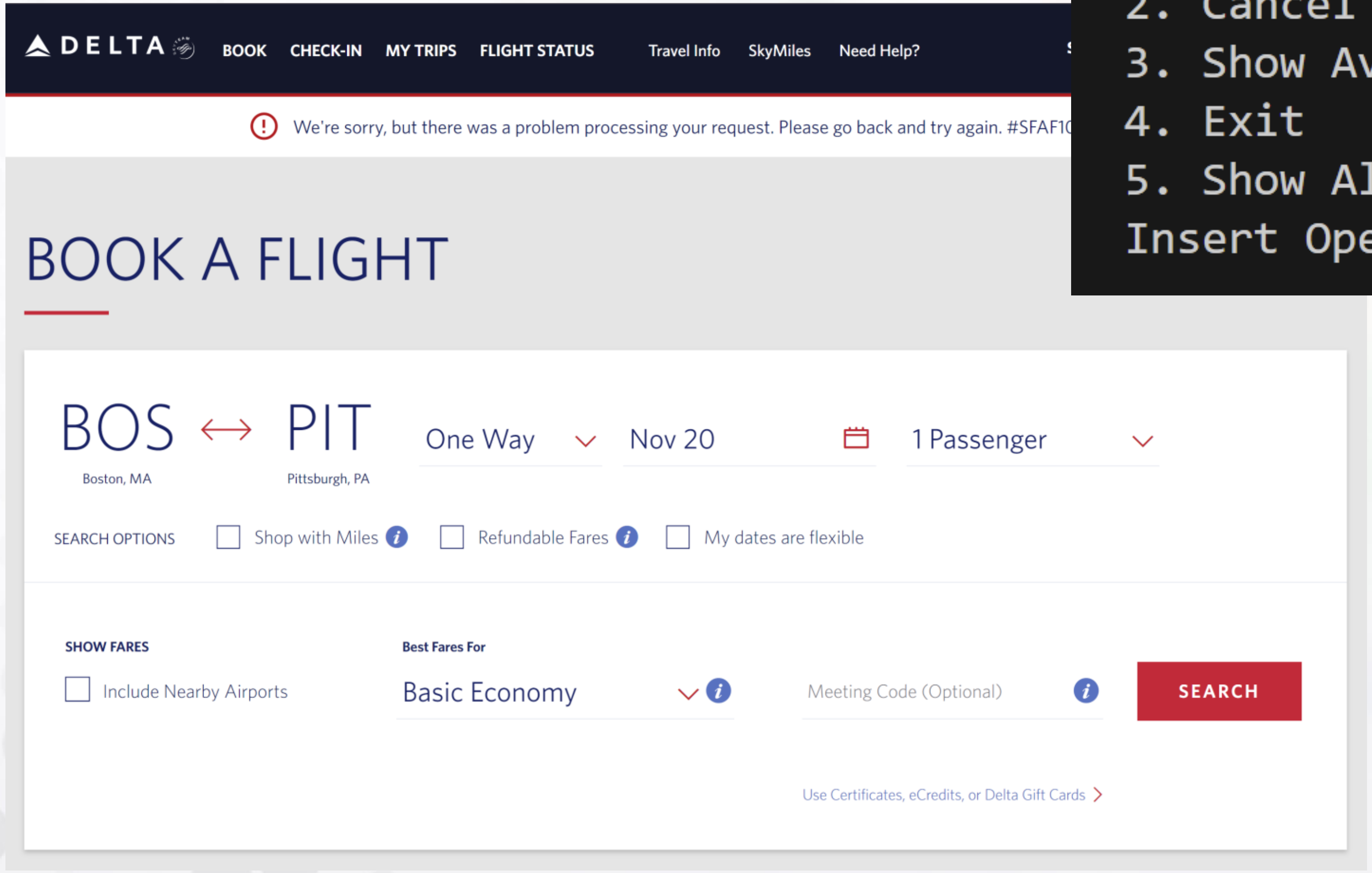
By Chris Lam, Dante LoPriore, Jasmine Sajna

Project Scope & Primary Objectives

- **Objective:**
  - Build a flight booking interface!
- **Project Scope:**
  - Linked List structures for Seats
    - Sort by Cost, Status, ID
  - Arrays for Flights
    - Filter by source & dest
    - Sort by date, duration
  - Backend & UI to mock existing websites

Related Work

Existing Flight Booking Sites



We're sorry, but there was a problem processing your request. Please go back and try again. #SFAF1(

BOOK A FLIGHT

BOS ⟷ PIT
Boston, MA Pittsburgh, PA
One Way
Nov 20
1 Passenger

SEARCH OPTIONS ☐ Shop with Miles ☐ Refundable Fares ☐ My dates are flexible

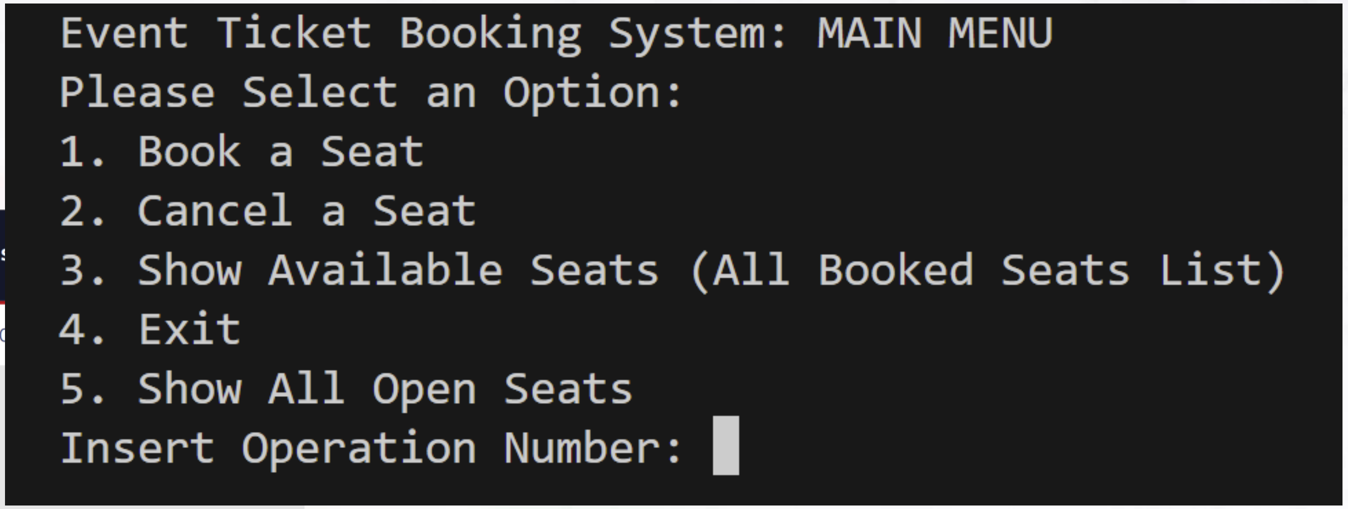
SHOW FARES
☐ Include Nearby Airports

Best Fares For
Basic Economy

Meeting Code (Optional)

SEARCH

Use Certificates, eCredits, or Delta Gift Cards



```
Event Ticket Booking System: MAIN MENU
Please Select an Option:
1. Book a Seat
2. Cancel a Seat
3. Show Available Seats (All Booked Seats List)
4. Exit
5. Show All Open Seats
Insert Operation Number:
```

Mini-Project

Data Entities Relationship (UML Diagram)

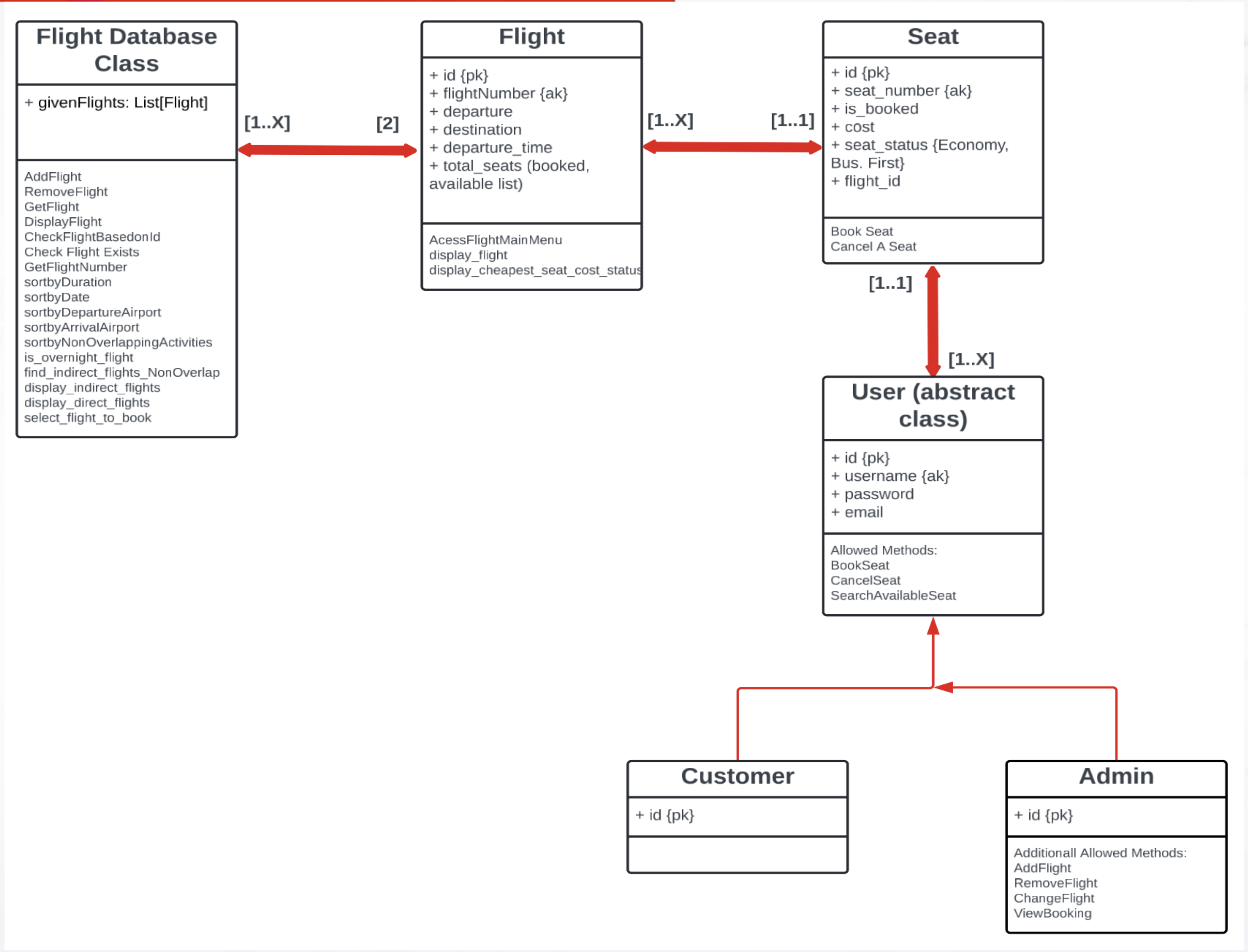
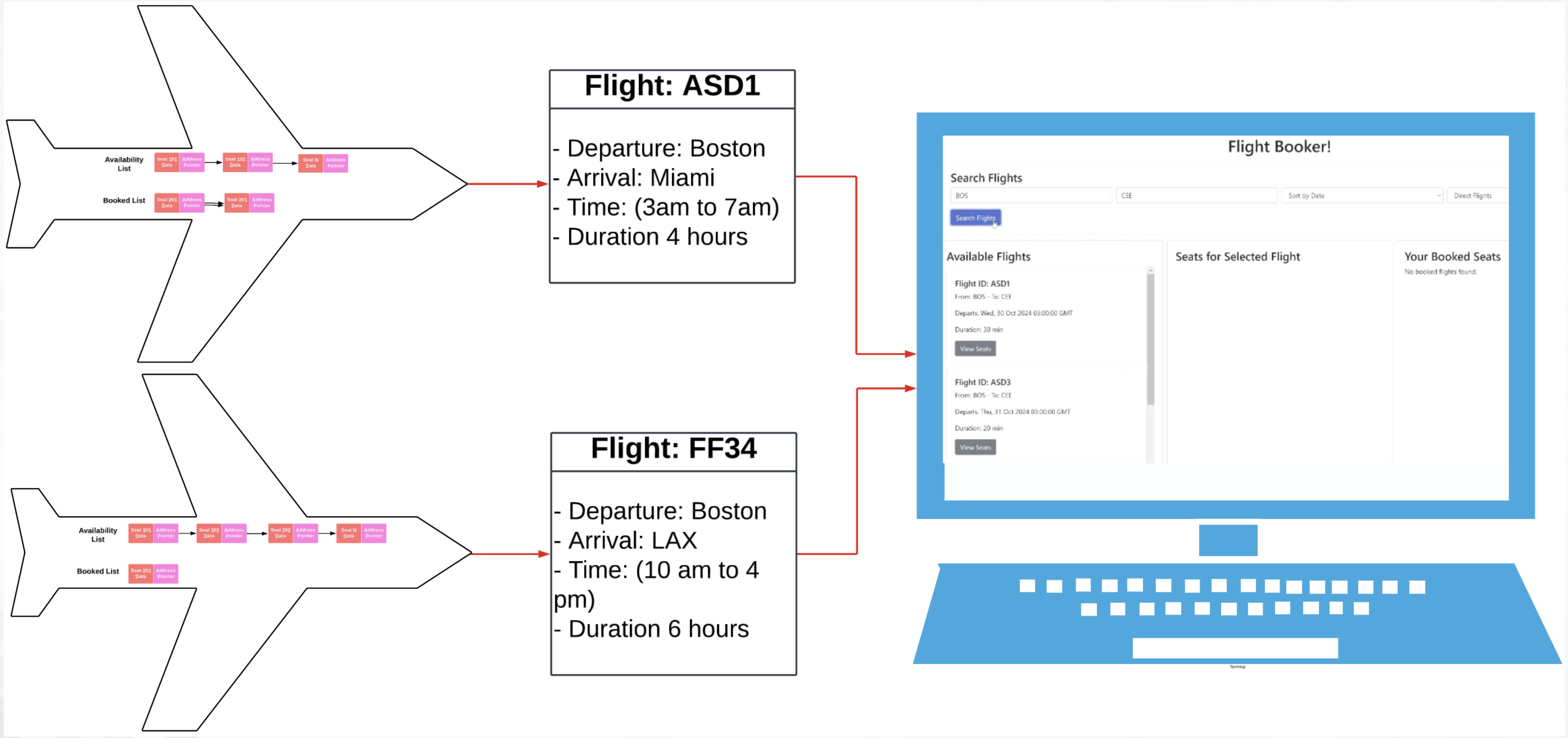


Figure 1: UML Diagram to Representation

Emphasize of Linked Lists To Create Seats



Flight: ASD1

- Departure: Boston
- Arrival: Miami
- Time: (3am to 7am)
- Duration 4 hours

Flight: FF34

- Departure: Boston
- Arrival: LAX
- Time: (10 am to 4 pm)
- Duration 6 hours

Flight Booker!

Search Flights

Available Flights

Flight ID: ASD1
From: BOS - To: CEE
Departs: Wed, 30 Oct 2024 03:00:00 GMT
Duration: 30 min

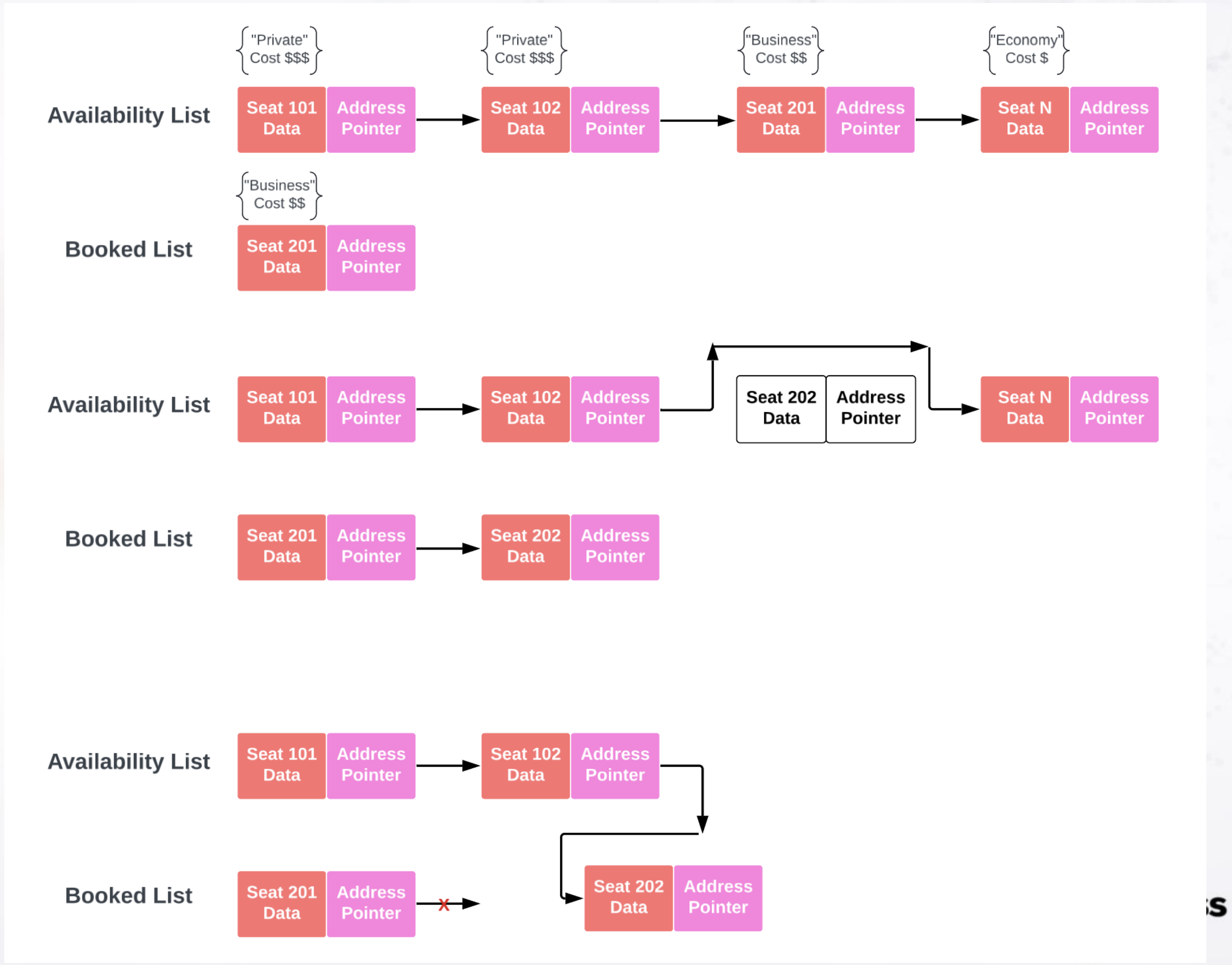
Flight ID: ASD3
From: BOS - To: CEE
Departs: Thu, 31 Oct 2024 03:00:00 GMT
Duration: 20 min

Seats for Selected Flight

Your Booked Seats
No booked flights found.

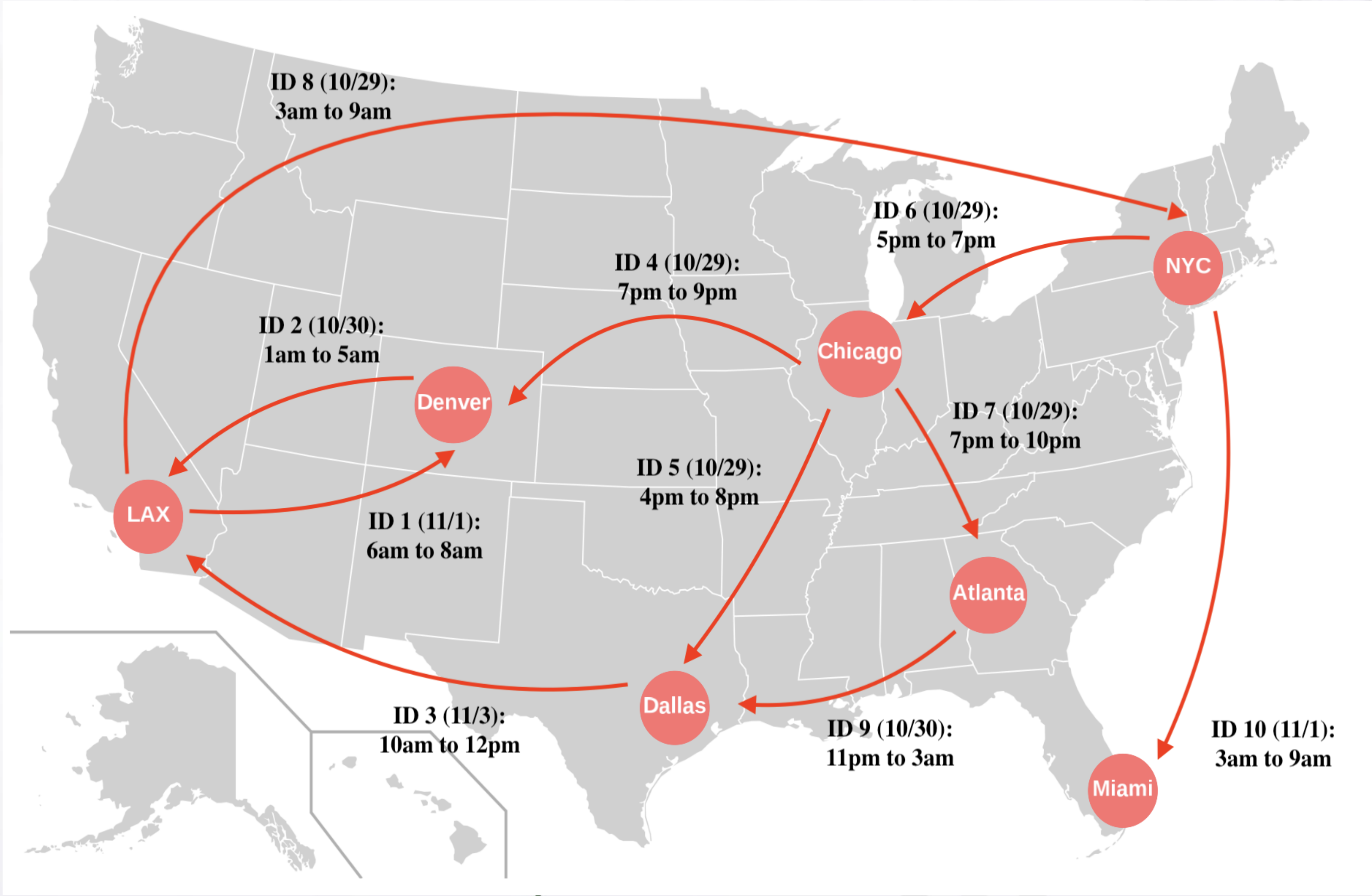
Emphasize of Linked Lists To Create Seats

- **Test**
  - A



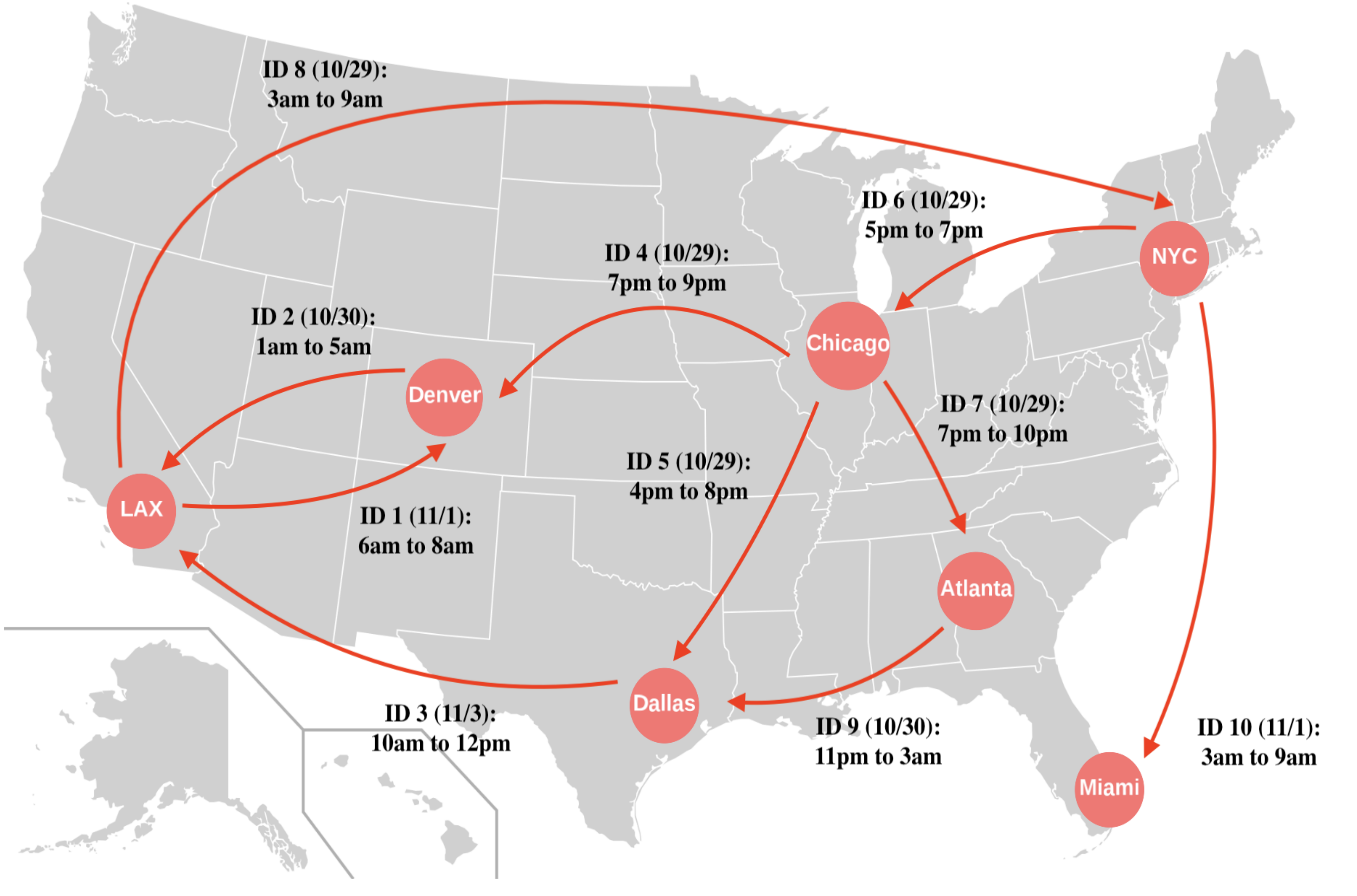
Step #1: Create Dictionary Of All Departing Flights Each

- **Key: NYC**
  - Items: Chicago, Miami
- **Key: Chicago**
  - Items: Denver, Atlanta, Dallas
- **Key: Atlanta**
  - Items: Dallas
- **Key: Dallas**
  - Items: LAX
- **Key: Denver**
  - Items: LAX
- **Key: LAX**
  - Items: Denver, NYC



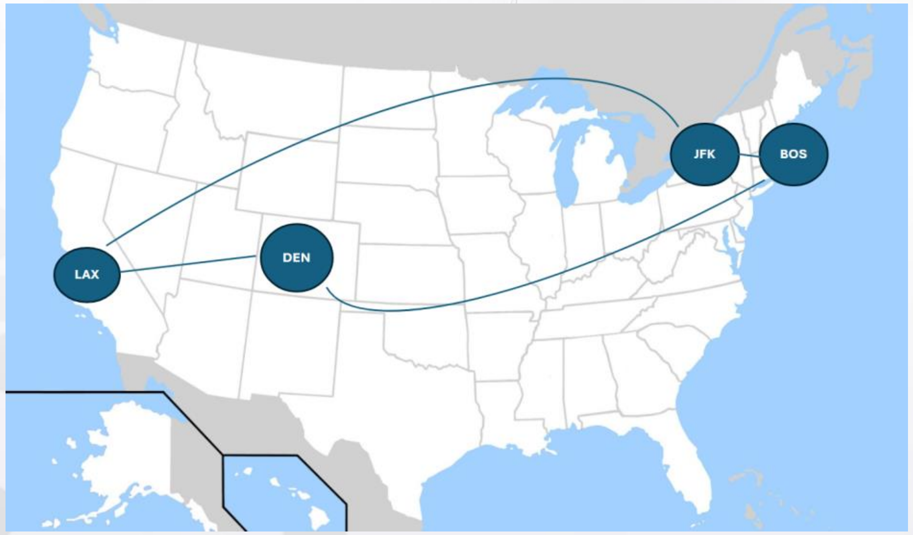
Step #2: Start BFS Sorting Algorithm / Check Known Pa

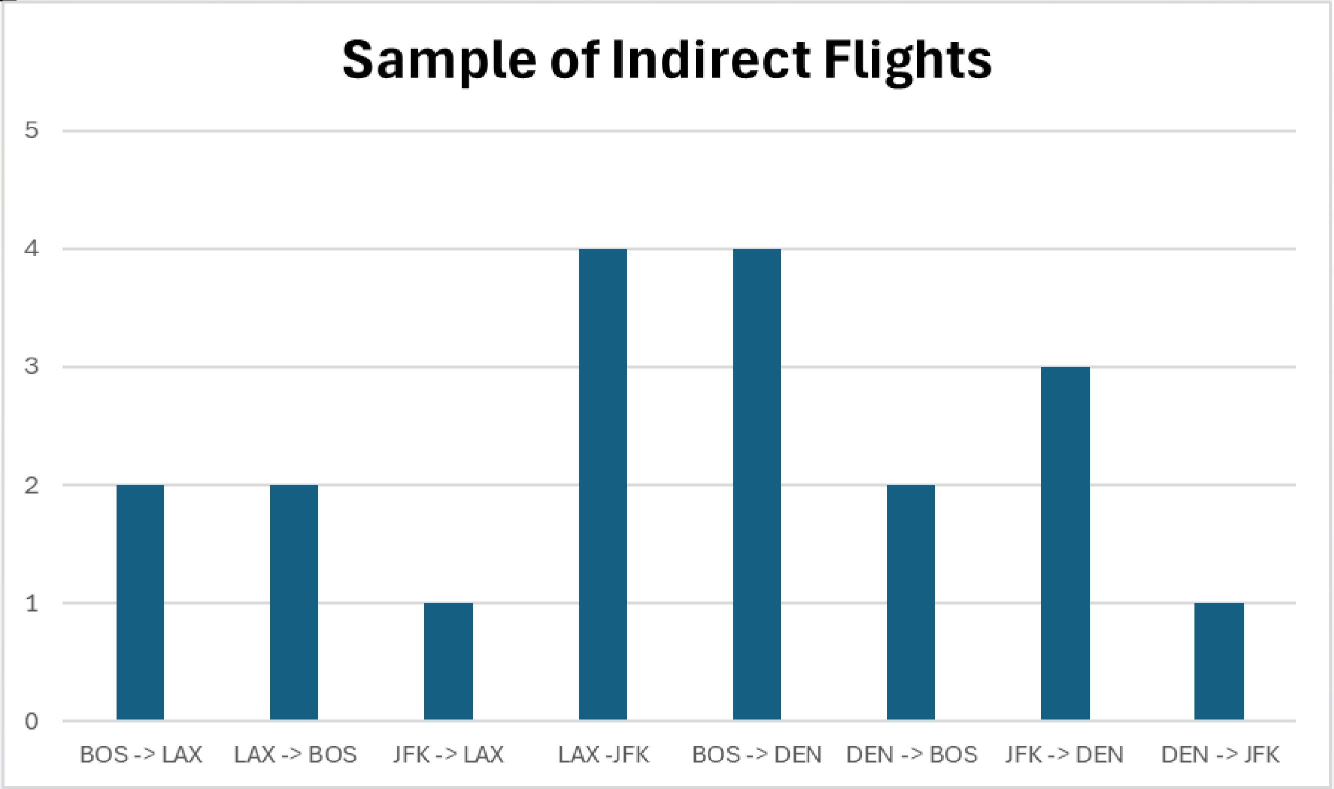
- **Constraints**
  - Non-overlapping Time Intervals
  - Indirect Flight Path duration has limit of 2 days from original source
  - Overnight Flights Are Considered
  - Can't Visit Same Airport More Than Once



Filtering Indirect Flights (BFS Algorithm)

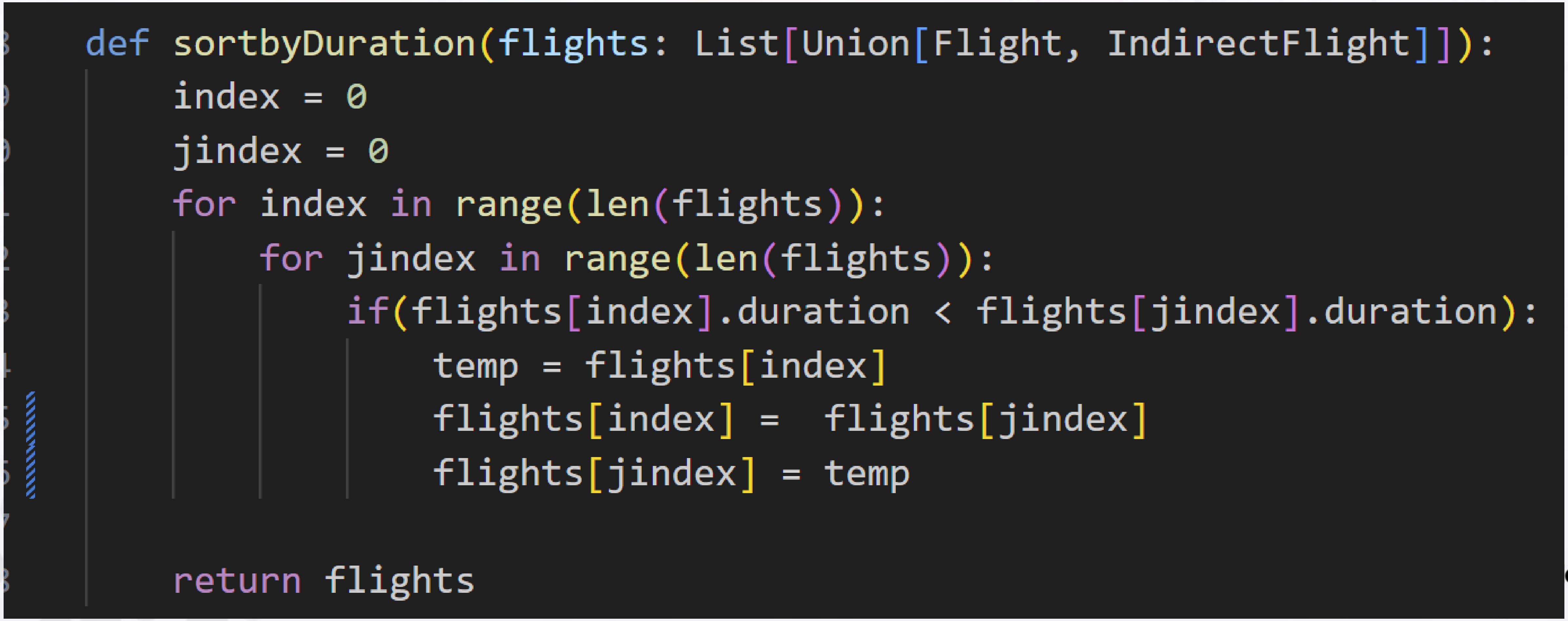
- Test: BOS to LAX (2 results)
  - Option1: Regular indirect flight
    - AA11(BOS) on 11/18
    - DD11(DEN) on 11/18
  - Option 2: Overnight/Different day flight
    - AA12(BOS) on 11/19
    - CC14(JFK) on 11/21





Sorting Flights (Simple Sorting Algo)

Bubble Sort (By Duration & Depart Date)



```python
def sortbyDuration(flights: List[Union[Flight, IndirectFlight]]):
    index = 0
    jindex = 0
    for index in range(len(flights)):
        for jindex in range(len(flights)):
            if(flights[index].duration < flights[jindex].duration):
                temp = flights[index]
                flights[index] =  flights[jindex]
                flights[jindex] = temp

    return flights
```

ess

at Northeastern

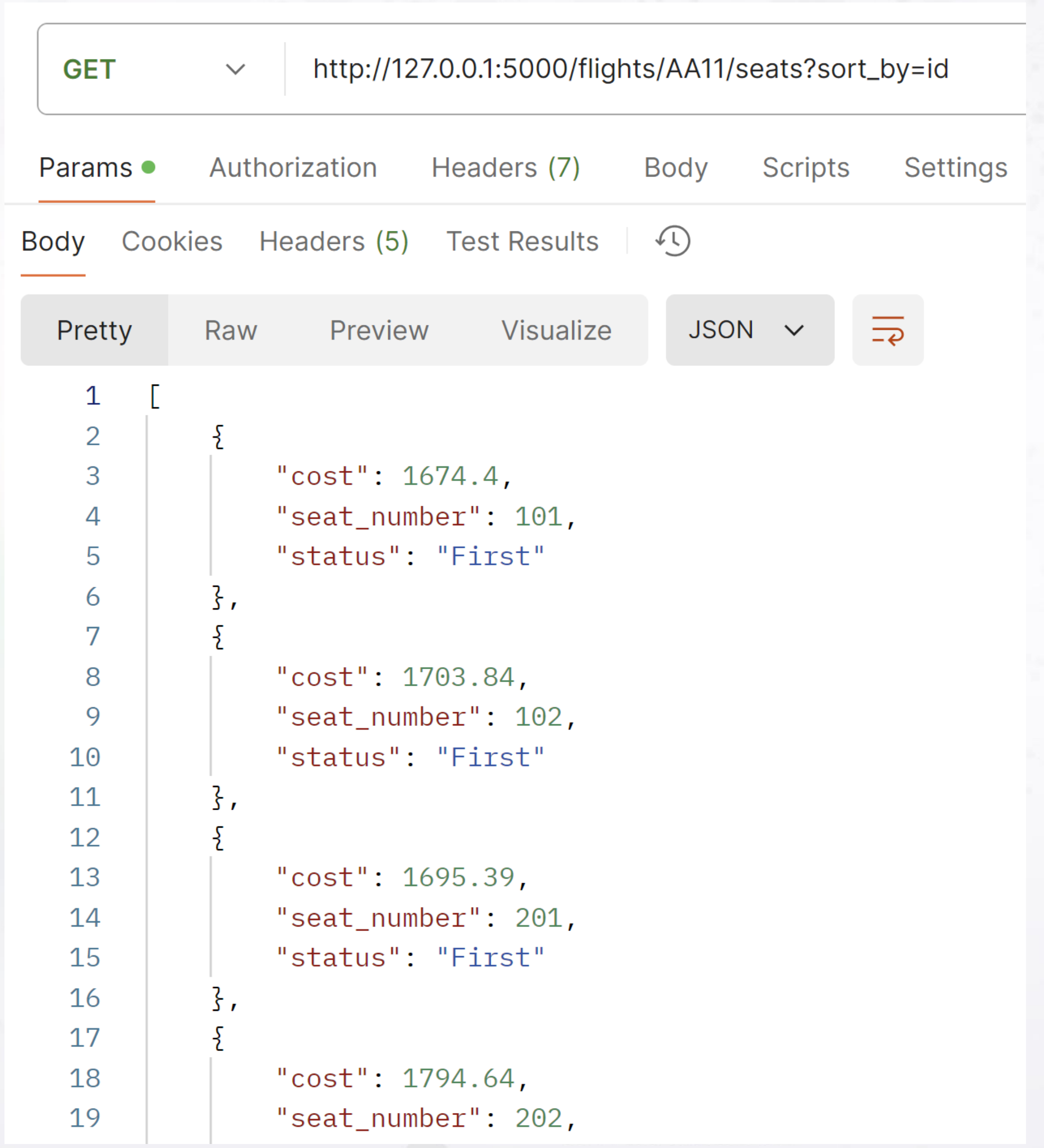
Backend Structures

Python Flask App

Convert Flight/Seat objects to JSON lists/objects:

- GET flight by ID
- GET direct and indirect flights for src & dest
- GET flights sorted by depart time, duration
- GET seats of a flight sorted by cost, status, id
- GET a single seat of a flight
- GET all booked seats

- POST a flight's seat as booked
- POST a flight's seat as cancelled

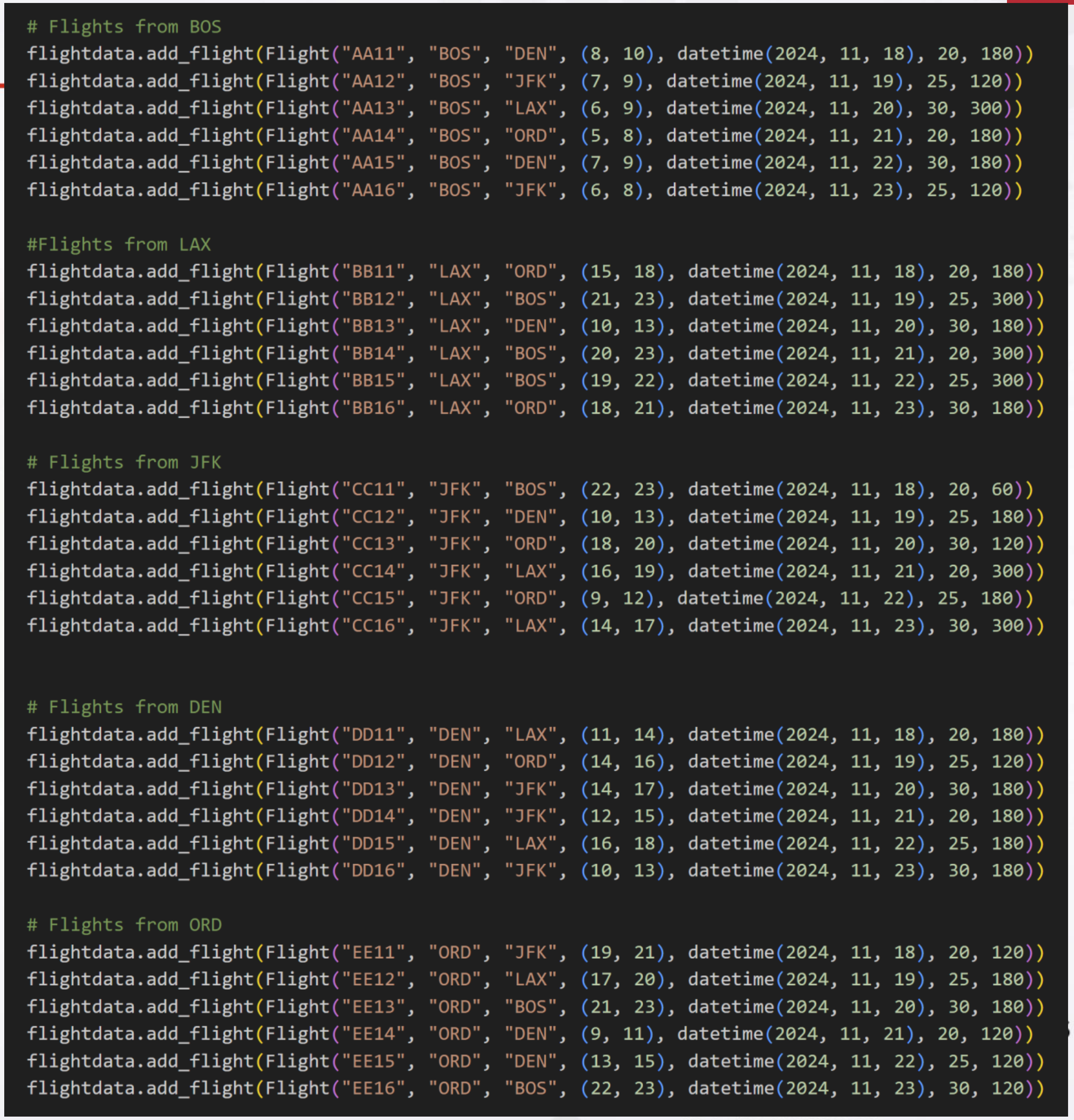


```
[
    {
        "cost": 1674.4,
        "seat_number": 101,
        "status": "First"
    },
    {
        "cost": 1703.84,
        "seat_number": 102,
        "status": "First"
    },
    {
        "cost": 1695.39,
        "seat_number": 201,
        "status": "First"
    },
    {
        "cost": 1794.64,
        "seat_number": 202,
```

Test Objects

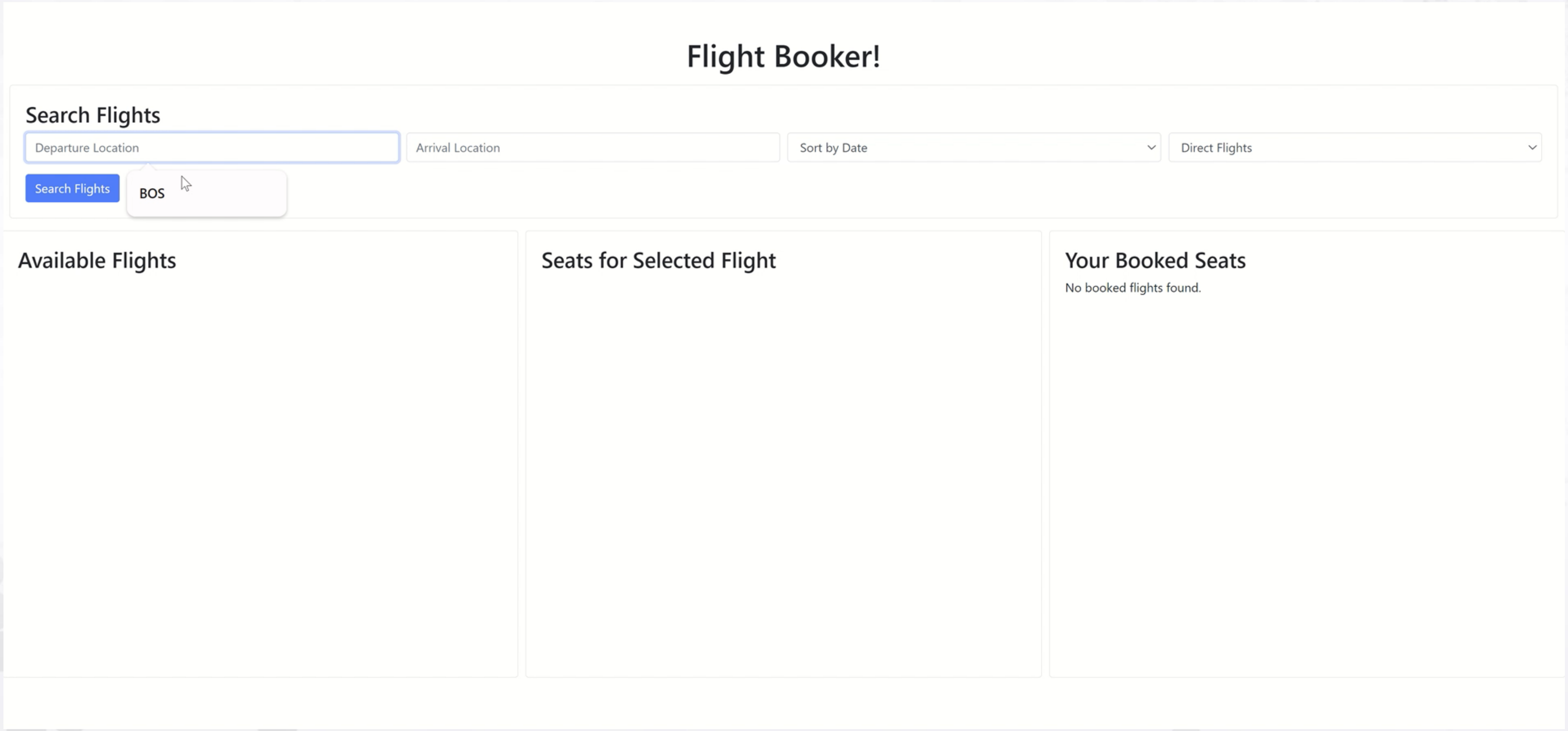
Populating the Flight Database:

- Overnight Flights
- Multiple Indirect Flight Routes
  - Shorter/Longer
  - Earlier/Later
- Multiple Direct Routes



```
# Flights from BOS
flightdata.add_flight(Flight("AA11", "BOS", "DEN", (8, 10), datetime(2024, 11, 18), 20, 180))
flightdata.add_flight(Flight("AA12", "BOS", "JFK", (7, 9), datetime(2024, 11, 19), 25, 120))
flightdata.add_flight(Flight("AA13", "BOS", "LAX", (6, 9), datetime(2024, 11, 20), 30, 300))
flightdata.add_flight(Flight("AA14", "BOS", "ORD", (5, 8), datetime(2024, 11, 21), 20, 180))
flightdata.add_flight(Flight("AA15", "BOS", "DEN", (7, 9), datetime(2024, 11, 22), 30, 180))
flightdata.add_flight(Flight("AA16", "BOS", "JFK", (6, 8), datetime(2024, 11, 23), 25, 120))

#Flights from LAX
flightdata.add_flight(Flight("BB11", "LAX", "ORD", (15, 18), datetime(2024, 11, 18), 20, 180))
flightdata.add_flight(Flight("BB12", "LAX", "BOS", (21, 23), datetime(2024, 11, 19), 25, 300))
flightdata.add_flight(Flight("BB13", "LAX", "DEN", (10, 13), datetime(2024, 11, 20), 30, 180))
flightdata.add_flight(Flight("BB14", "LAX", "BOS", (20, 23), datetime(2024, 11, 21), 20, 300))
flightdata.add_flight(Flight("BB15", "LAX", "BOS", (19, 22), datetime(2024, 11, 22), 25, 300))
flightdata.add_flight(Flight("BB16", "LAX", "ORD", (18, 21), datetime(2024, 11, 23), 30, 180))

# Flights from JFK
flightdata.add_flight(Flight("CC11", "JFK", "BOS", (22, 23), datetime(2024, 11, 18), 20, 60))
flightdata.add_flight(Flight("CC12", "JFK", "DEN", (10, 13), datetime(2024, 11, 19), 25, 180))
flightdata.add_flight(Flight("CC13", "JFK", "ORD", (18, 20), datetime(2024, 11, 20), 30, 120))
flightdata.add_flight(Flight("CC14", "JFK", "LAX", (16, 19), datetime(2024, 11, 21), 20, 300))
flightdata.add_flight(Flight("CC15", "JFK", "ORD", (9, 12), datetime(2024, 11, 22), 25, 180))
flightdata.add_flight(Flight("CC16", "JFK", "LAX", (14, 17), datetime(2024, 11, 23), 30, 300))


# Flights from DEN
flightdata.add_flight(Flight("DD11", "DEN", "LAX", (11, 14), datetime(2024, 11, 18), 20, 180))
flightdata.add_flight(Flight("DD12", "DEN", "ORD", (14, 16), datetime(2024, 11, 19), 25, 120))
flightdata.add_flight(Flight("DD13", "DEN", "JFK", (14, 17), datetime(2024, 11, 20), 30, 180))
flightdata.add_flight(Flight("DD14", "DEN", "JFK", (12, 15), datetime(2024, 11, 21), 20, 180))
flightdata.add_flight(Flight("DD15", "DEN", "LAX", (16, 18), datetime(2024, 11, 22), 25, 180))
flightdata.add_flight(Flight("DD16", "DEN", "JFK", (10, 13), datetime(2024, 11, 23), 30, 180))

# Flights from ORD
flightdata.add_flight(Flight("EE11", "ORD", "JFK", (19, 21), datetime(2024, 11, 18), 20, 120))
flightdata.add_flight(Flight("EE12", "ORD", "LAX", (17, 20), datetime(2024, 11, 19), 25, 180))
flightdata.add_flight(Flight("EE13", "ORD", "BOS", (21, 23), datetime(2024, 11, 20), 30, 180))
flightdata.add_flight(Flight("EE14", "ORD", "DEN", (9, 11), datetime(2024, 11, 21), 20, 120))
flightdata.add_flight(Flight("EE15", "ORD", "DEN", (13, 15), datetime(2024, 11, 22), 25, 120))
flightdata.add_flight(Flight("EE16", "ORD", "BOS", (22, 23), datetime(2024, 11, 23), 30, 120))
```

Results of Final System



Flight Booker!

Search Flights

Available Flights

Seats for Selected Flight

Your Booked Seats

No booked flights found.

Discussion & Conclusion

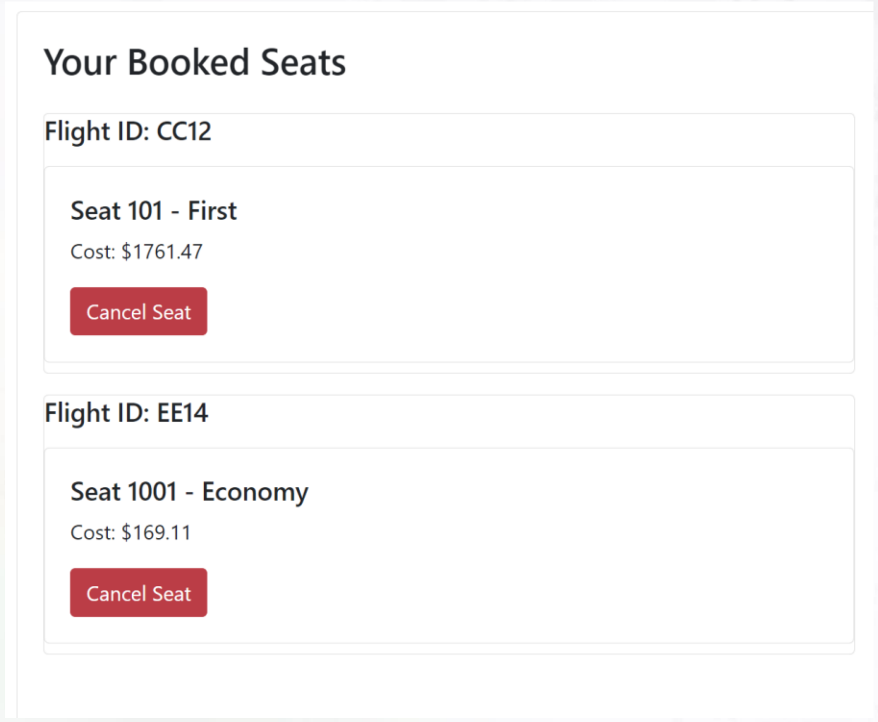
The booker works!

Drawbacks/Limitations:

- Assumes only ONE user
- Flight options are limited

Possible Extensions:

- Build a Customer/User class
- Use an API to grab real flights



Your Booked Seats

Flight ID: CC12

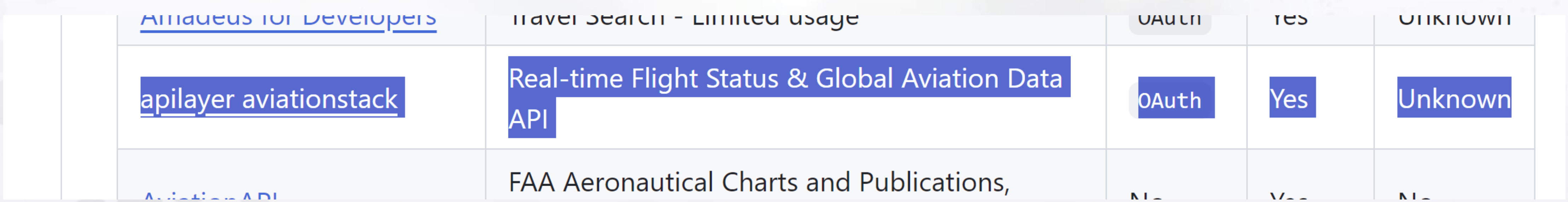
Seat 101 - First

Cost: $1761.47

Flight ID: EE14

Seat 1001 - Economy

Cost: $169.11



Amadeus for Developers	Travel Search - Limited usage	OAuth	Yes	Unknown
apilayer aviationstack	Real-time Flight Status & Global Aviation Data API	OAuth	Yes	Unknown
AviationAPI	FAA Aeronautical Charts and Publications,	No	Yes	No