# Data Structures

## User Data Structure

```
public class user {
        public long UserID {get;set;}
        public string Name {get;set;}
        public string Email {get;set;}
        public string PhoneNumber {get;set;}

}
```

## Alert Data Structure

```
public class FlightAlert {
        long AlertID {get;set;}
        long userID {get;set;}
        Flight flightData {get;set;}


}
public class Flight {
        int FlightNumber {get;set;}
        DateTime DepartureTime {get;set;}
        DateTime ArrivalTime {get;set;}
        double price {get;set;}
        public Currency PriceCurrency { get; set; }
        string DepartureAirPort {get;set;}
        string ArrivalAirPort {get;set;}

}
```

# Data Flow

1. **Price Data Reception:**
    - External sources send flight price data to the Processing flight prices service.
    - The flight prices service validates and normalizes data.
    - send event data to the User Alert managment service
2. **Processing flight  service:** Receives incoming price data and validates/normalizes the information.

2. **Alert Management (CRUD):**
    - Users interact with the **Alert Management Service** via mobile or web client.
    - They can create, update, delete, and view alerts.
    - All CRUD operations update the Database (storing user alerts and associated information).
    - Create the relevant event alert according the user's information and their alert data stored in the database and send it to queue.

3. **Event Generation & Queuing:**
    - receiving price data event as producer from Alert Management.
    - the event is placed on the **Event Queue**.

4. **Notification:**
    - The **Notification Service** listens to the Event Queue.
    - It fetches the corresponding user’s device token from the Database and forwards the push notification to the **Mobile Push Gateway**.
    - The Mobile Gateway forwards the push notifications to mobile devices.