***Theme:***

* Travel/Booking Application
  + You’re at the airport ordering your ticket or tickets. It will check the airplane status, and assign your ticket number

***Classes:***

* TicketBookingSystem (main)
* Plane
* SchedulePlanes
* Abstract Passenger
  + Adult extends Passenger
  + Senior extends Passenger (or Adult)
  + Child extends Passenger (or Adult)

***Class Building:***

**TicketBookingSystem: <Everyone>**

* Variables:
  + Static List<List<Plane>> Planes (month (12), list of days (7))
  + Static Scanner in
* Methods:
  + Static getNextSeat
    - Takes a seat from AvailableSeats and puts it into TakenSeats
  + Static listFlightDetails
    - lists total number of passengers and the breakdown of seniors, adults, and children for a particular flight
  + Static getFlightDetails
    - get month and weekday: return plane object.toString
  + Static getString
  + Static getInt

**Plane: <Melissa>**

* Variables:
  + Static char[] ROWS;
  + Static int[] SEATS\_IN\_ROW;
  + Static double[][][] PRICEs (Month, weekday, Adult / Child / SeniorPrice)
  + Static ArrayList<String> SEATS
  + String Month
  + String Weekday
  + Double[] prices
  + ArrayList<Adult> adults
  + AttayList<Children> children
  + ArrayList<Seniors> seniors
  + ArrayList<String> availableSeats
  + ArrayList<String> takenSeats
* Methods
  + Plane
    - Give a seating chart for seatInfo
    - Set the three lists to empty
  + Plane
    - Initialise lists from saved object
  + ToString
    - List the prices, number of passengers, how many passengers, and the passenger sub types
  + Private List<double> getPrice
    - Returns the pricelist based on month and weekday of plane
  + GetChildPrice
    - Return the second price in GetPrice getPrice()[1]
  + GetAdultPrice
    - Return the first price in GetPrice getPrice()[0]
  + GetSeniorPrice
    - Return the third price in GetPrice getPrice()[2]

**SchedulePlanes: <Melissa>**

* Variables:
  + Hashtable<String, ArrayList<Plane>> schedule
* Methods:
  + Static void generateMonth
    - Generates a month’s worth of planes and puts them in order by date. Then it adds it to the hashtable under the provided key
  + Static void updateSchedule
    - Updates a schedule by adding the current month if it isn’t there, add the next month if it isn’t there and removing the previous month (as you can’t board a plane in the past)
  + Static void addMonthToSchedule
    - Adds a month to the schedule by calling generateMonth
  + Static void removeMonthFromSchule
    - Removes a month using the provided key

**Passenger: <Ciaran>**

* Variables:
  + String seat
  + Static Int numberOfPassengers = 0
  + String firstName
  + String lastName
  + Int ticketNumber
  + String departureDate
* Methods:
  + Getters and Setters
  + toString
    - Lists full name, seat number, ticketNumber, and price they paid

**Adult: <Ciaran>**

* Variables:
  + Static Int numberOfAdults
* Methods:
  + Constructor
    - Make a constructor like the one in Passenger using this() for the default constructors and using super() to fill out the Passenger details from the super class.
  + Getters and setters for all variables
    - Make getters public, setters protected.

**Child: <Michelle>**

* Variables:
  + Static Int numberOfChildren
  + Boolean travelingAlone
* Methods:
  + Constructor
    - Make a constructor like the one in Passenger using this() for the default constructors and using super() to fill out the Passenger details from the super class.
  + Getters and setters for all variables
    - Make getters public, setters protected.

**Senior: <Michelle>**

* Variables:
  + Static Int numberOfSeniors
* Methods:
  + Constructor
    - Make a constructor like the one in Passenger using this() for the default constructors and using super() to fill out the Passenger details from the super class.
  + Getters and setters for all variables
    - Make getters public, setters protected.

***Things to consider:***

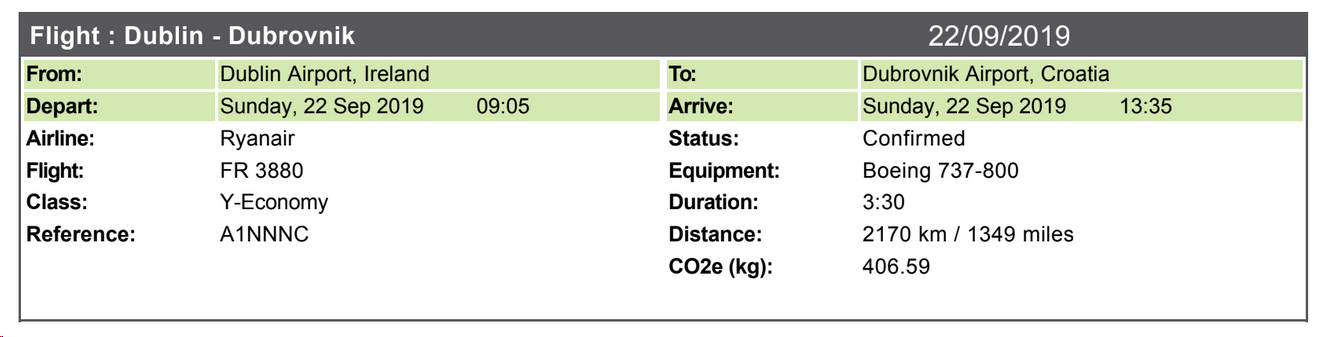


Figure : An example plane ticket confirmation for Melissa's trip to Dubrovnik back in 2019