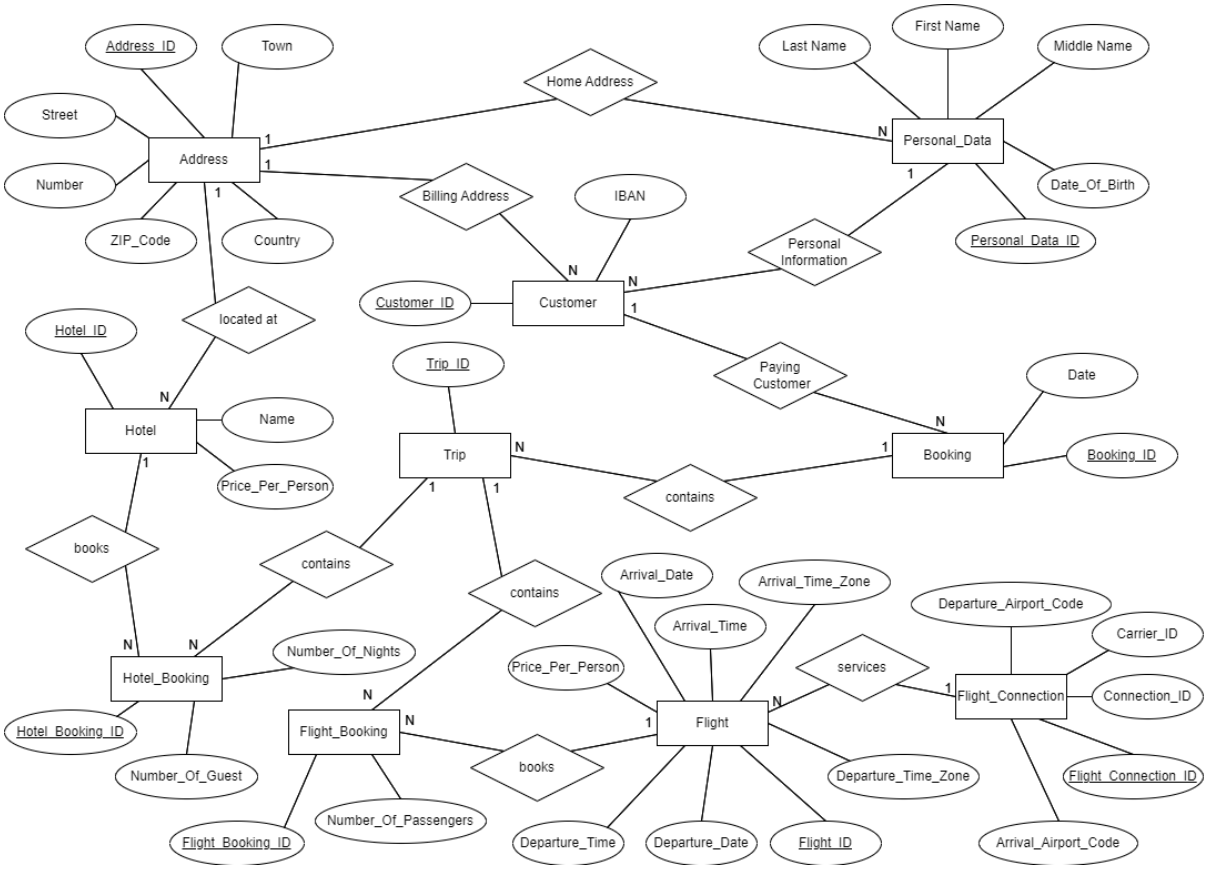
Table of Contents

0 ENTITIES – ENTITY RELATIONSHIP DIAGRAM....1

1 ENTITIES – CLASS DIAGRAM....2

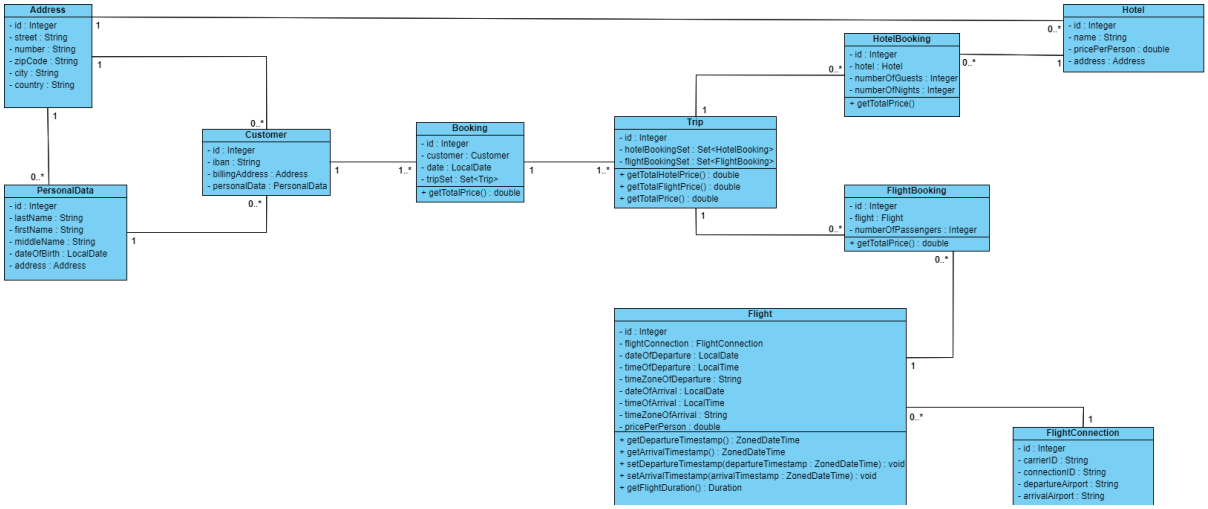
0 Entities – Entity Relationship Diagram

This UML-Entity-Relationship-Diagram models the underlying database scheme for this service.



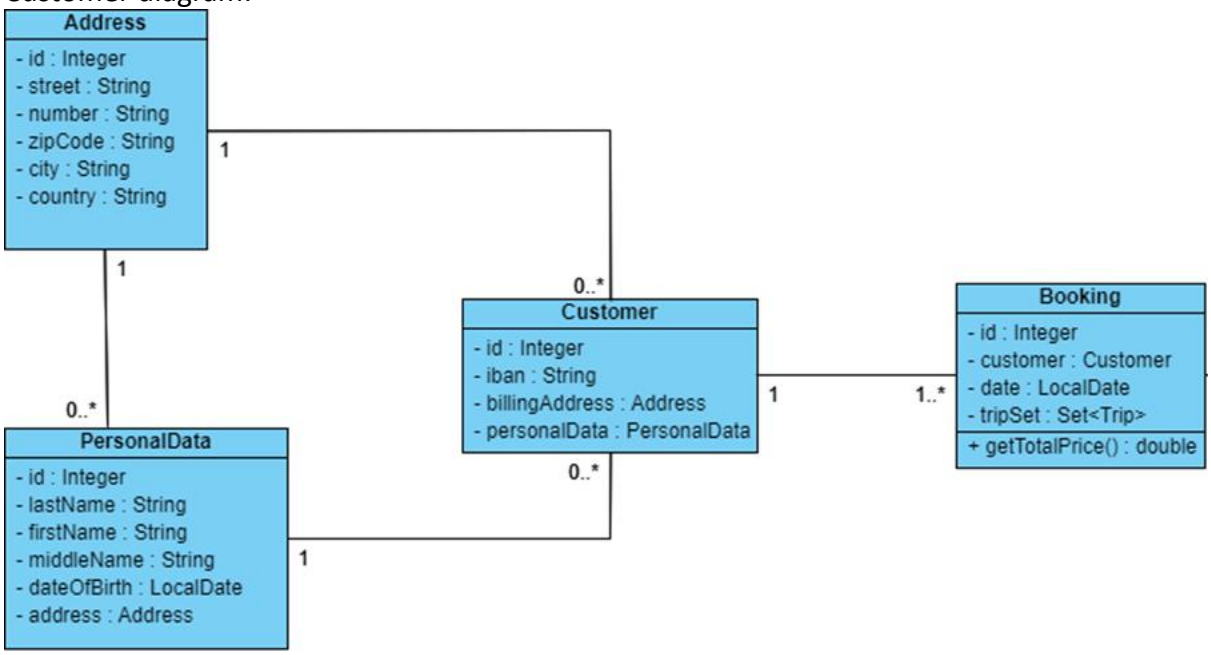
1 Entities – Class Diagram

This UML-Class-Diagram shows the underlying database scheme for this service.
For each private attribute there is a *getter* and a *setter* method which is not modelled for simplicity reasons. In case, only additional methods will are modelled in this diagram.
Please notice that there is not *setter* for the field *id das* it will be generated by the Persistence automatically.
Also the *getter* and *setter* for the attributes dateOfDeparture, *timeOfDeparture*, *timeZoneOfDeparture*, *dateOfArrival*, *timeOfArrival*, and *timeZoneOfArrival* will be left out and replaced with the *setter* and *getter* methods for *departureTimestamp* and *arrivalTimestamp*. The attributes mentioned will be derived from/used to create their corresponding timestamp.

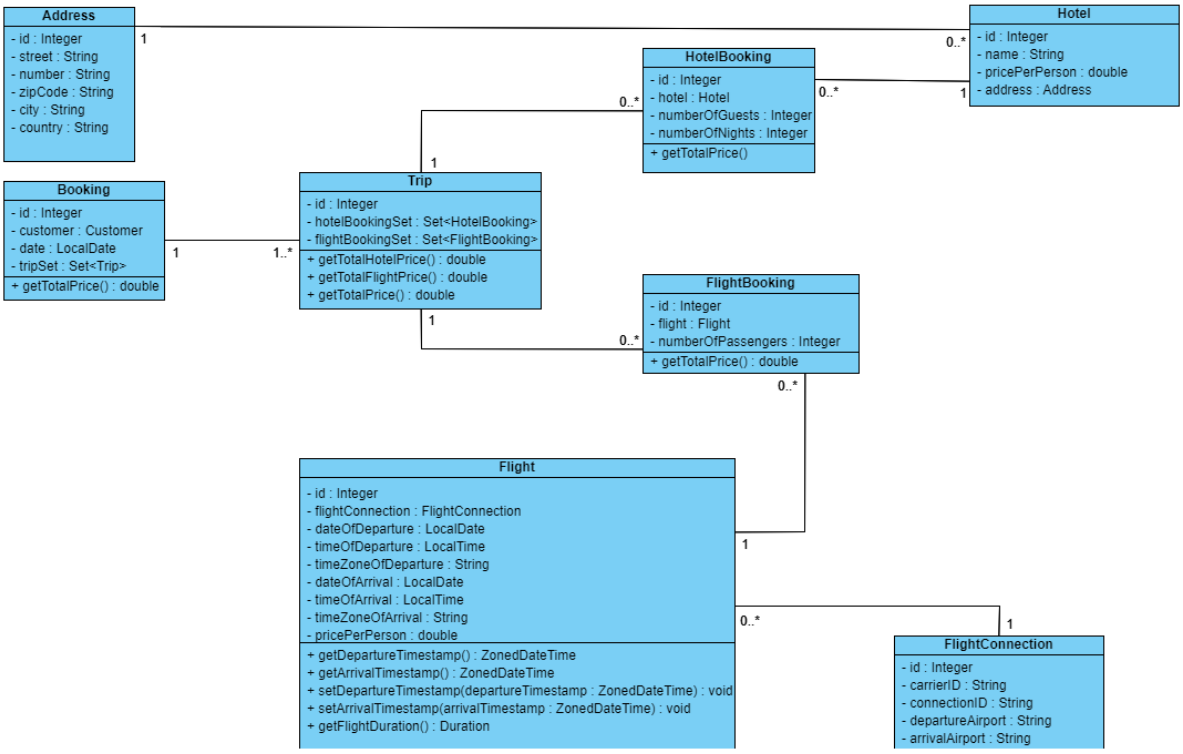


For better understanding, the diagram can also be divided into the customer side and the trip side.

Customer diagram:



Trip diagram:


Address
- id : Integer
- street : String
- number : String
- zipCode : String
- city : String
- country : String
Hotel
- id : Integer
- name : String
- pricePerPerson : double
- address : Address
HotelBooking
- id : Integer
- hotel : Hotel
- numberOfGuests : Integer
- numberOfNights : Integer
+ getTotalPrice()
Booking
- id : Integer
- customer : Customer
- date : LocalDate
- tripSet : Set<Trip>
+ getTotalPrice() : double
Trip
- id : Integer
- hotelBookingSet : Set<HotelBooking>
- flightBookingSet : Set<FlightBooking>
+ getTotalHotelPrice() : double
+ getTotalFlightPrice() : double
+ getTotalPrice() : double
FlightBooking
- id : Integer
- flight : Flight
- numberOfPassengers : Integer
+ getTotalPrice() : double
Flight
- id : Integer
- flightConnection : FlightConnection
- dateOfDeparture : LocalDate
- timeOfDeparture : LocalTime
- timeZoneOfDeparture : String
- dateOfArrival : LocalDate
- timeOfArrival : LocalTime
- timeZoneOfArrival : String
- pricePerPerson : double
+ getDepartureTimestamp() : ZonedDateTime
+ getArrivalTimestamp() : ZonedDateTime
+ setDepartureTimestamp(departureTimestamp : ZonedDateTime) : void
+ setArrivalTimestamp(arrivalTimestamp : ZonedDateTime) : void
+ getFlightDuration() : Duration
FlightConnection
- id : Integer
- carrierID : String
- connectionID : String
- departureAirport : String
- arrivalAirport : String
1
0..*
1
0..*
0..*
1
1
1..*
1
0..*
0..*
1
0..*
1