

COPENHAGEN BUSINESS ACADEMY



Semester Project “Momondo”

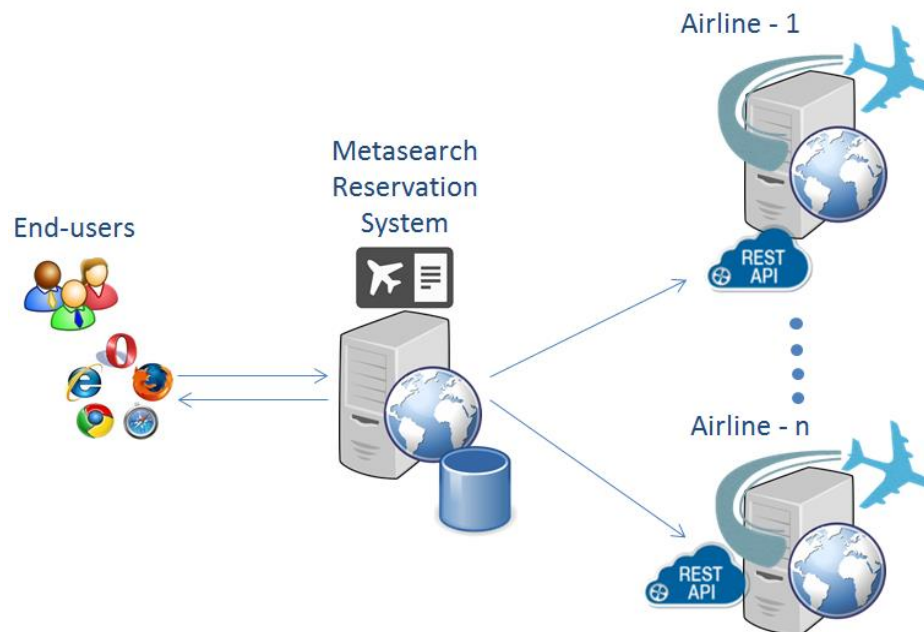


Momondo Projectc

The task for this project is to create a Proof of Concept solution for an Online Reservation System, similar to Web sites like Momondo.com.

The solution must include the following subsystems:

- A Prototype for an actual airline that exposes its reservation system via a REST API (Airline 1-n, below)
- The [Metasearch](#) Reservation System



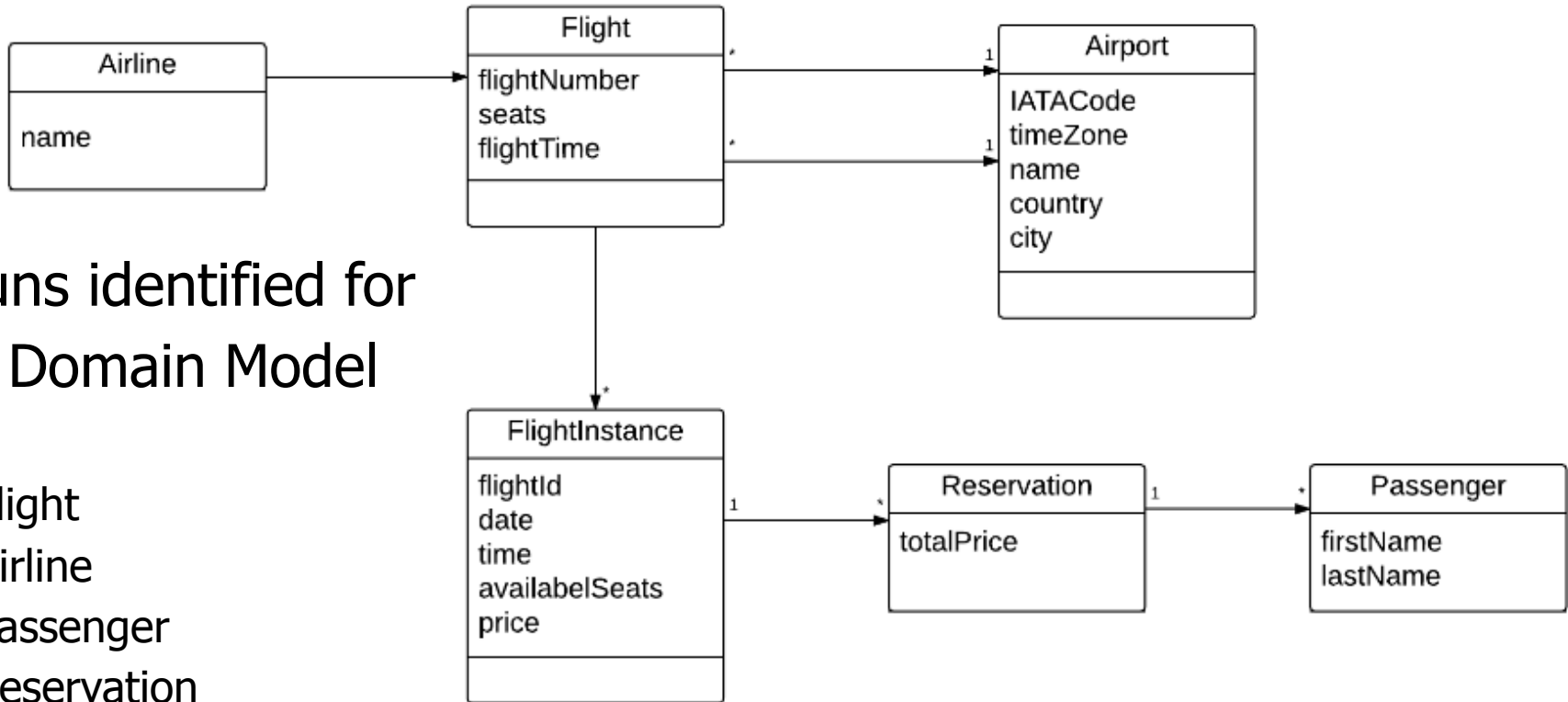
Semester Project – Assumptions

When planning/designing the semester project and the public contract between all "momondos" and all "airlines" you can plan/design under the following assumptions:

- Seat numbers in an airplane are given as 1 – numSeats in the plane (no row numbers)
- There is only one (monkey) class, and seats are sold un-numbered (first in, best seat)
- All airports must be given as a IATA-code + time zone (see for example: <https://www.world-airport-codes.com/>). Feel free to implement an extra REST-service to provide the real name, given an IATA code, in your client GUI
- We will not focus on money transactions, so you can only reserve seats, not buy
- TIMEs are all given as start times in local time, no end time is given but flight time must be given, so end time can be calculated (given the time zone included for all airports)
- Times are formatted as ISO 8601 (http://en.wikipedia.org/wiki/ISO_8601)
- When searching for available flights only a date is given, this should find all departures for that day
- We only handle Direct Flights
- Only one Way Flights, if you want to return, buy a ticket for the other way

Initial Planning

Assumptions made with previous classes




Nouns identified for the Domain Model

- Flight
- Airline
- Passenger
- Reservation
- Airport

What we (as a minimum) want

- Get all flights, with free tickets, from a given airport at a given date for the requested amount of tickets.
- Get all flights with free tickets from a given departure airport to a given destination airport, for a given date, and the requested amount of tickets.



A flight search interface with a teal background. It features two input fields for origin and destination cities, separated by a double-headed arrow. To the right are two date pickers for departure (25 Nov 2015) and return (30 Nov 2015), followed by a passenger count dropdown (1 Passenger) and a red circular search button.

The ability to reserve flight seats from a provided list of options

Implementing the Airline

Dummy-Airline using JPA and JAX-RS



Implement a "dummy airline" using JPA and implement the public REST API using JAX RS. Set up a number of dummy flights in the database and publish your available flights so that other teams knows what to search for

Real data using screen scraping or external APIs



Fetch real flight information, using screen scraping, or available APIs like QPX Express API² or similar. Implement an "adapter service" that transforms between our REST API and the external source you are using.

REST API that must be implemented by all  cphbusiness

"Semester-Airlines"

GET

Request all available flights that matches the provided search criteria's

api/flights/:from/:date/:tickets

Request all available flights that matches the provided search criteria

api/flights/:from/:to/date/:tickets

POST

Make a reservation for flight and persons provided with request

api/reservation/:flightId

flightId: id returned by one of the two GET requests

Test Dummy-Airline

<http://airline-plaul.rhcloud.com/#/links>



See and READ the project specification (or hand-out) for a full description of the API

Date/Time-assumptions in the project cphbusiness

To abstract away, most date/time-related problems, which a real-life Momondo system will face, we have made several assumptions:

- All airports must be given as a IATA-code + time zone (The only place we store time zone information)
- TIMEs are all given as start times in local time, no end time is given but flight time must be given, so end time can be calculated (given the time zone included for all airports)
- Times are formatted as ISO 8601 (http://en.wikipedia.org/wiki/ISO_8601) but we disregard the time-zone indicator of the String (assuming local time, and that we know the location)

See: The Semester Project and Date Handling.pdf

