Contract

Møjungerne & TeamPer

Logical data model

Native	Alien

Car	Truck	Heavy machinery

Passenger

Vehicle
Weight
RegNo
Lenght

Admin
UserName
Password

TravelingEntity

Reservation
ReservationNumber
HasArrived

Booker — 1

Person
FirstName
LastName
Email

TravelingEntityType

1

Departure
DepartureTime
DepartureDate

0..1

FerryConfig
PeopleCapacity
VehicleCapacity
WeightCapacity

1..* — 1

Ferry
Name

Price
Amount

* — 1

Route
TravelTime

Destination — 1

Harbour
Name

Origin — 1

Use case model - Use case diagram
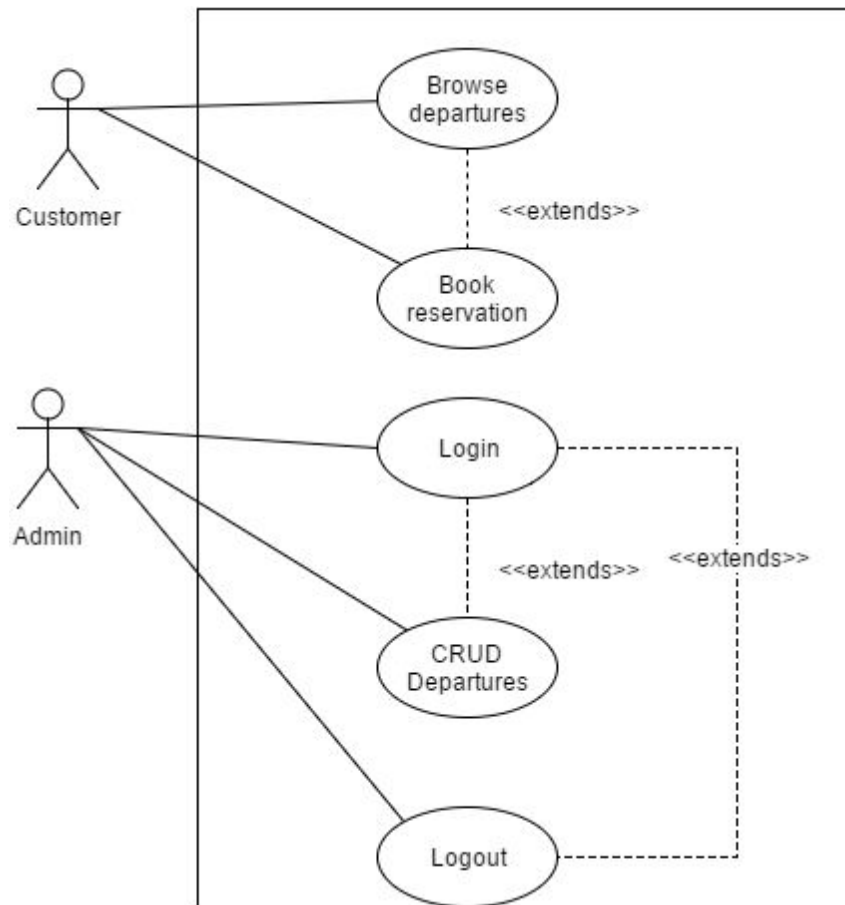
Use case descriptions

Use case #1 Browse departures

- **Goal in context:** Customer filters departures.

- **Scope:** Customer

- **Level:** Primary task

- **Precondition:** A customer (no login required), Departures setup by an admin.

- **Success and condition:** The system displays departures for the Customer

- **Failed condition:** The system fails to display departures for the Customer.

- **Primary Actor:** A customer

- **Trigger:**

Success scenario

1. Customer selects "Browse departures"
2. System(Front end) shows a list of all current routes
3. Customer adds filters to meet his/her wishes
4. System allows for filtering to meet Customer wishes
5. Customer select a given departure
6. System displays additional information about the given departure

Use case #2 Make Reservation

Scope: Customer
Level: Goal: New reservation are created
Primary Actor: User/Customer
Precondition: The customer has selected a valid departure.

Main success scenario
1. System shows an overview of the departure.
2. Customer presses on Book on a selected departure
3. System shows details about the booking, the customer is trying to make and ask the customer to confirm.
4. Customer confirms the booking
5. System add the customer to the created booking

Success guarantees: The customer have a valid booking.
Extensions: None

Use case #3 Login

Name: Login.

Scope: Admin.

Level: Login to admin panel.

Primary Actor: Administrator of the ferry company(employee).

Precondition: The admin has a username and a password.

Success and condition: Admin is logged in.

Failed condition: Admin is not logged in.

Success scenario:
1. Admin navigate to login site
2. Admin writes username and password
3. Username and password is found in the admin system and correct.
4. Admin is logged into the admin panel.

Use case #4 Logout

Name: Logout

Scope: Admin

Level: Logout from the admin panel.

Primary Actor: Administrator of the ferry company(employee).

Precondition: The admin is logged in.

Success and condition: Admin is logged out from the admin panel.

Failed condition: Admin is not logout.

Success scenario:

1. Admin press on logout button.
2. Admin is logged out and the system redirects to login site.

Use case #5 CRUD departures

Name: Create departures

Scope: Admin

Level: Goal: New departure created

Primary Actor: Administrator of the ferry company(employee).

Precondition: The admin is logged in.

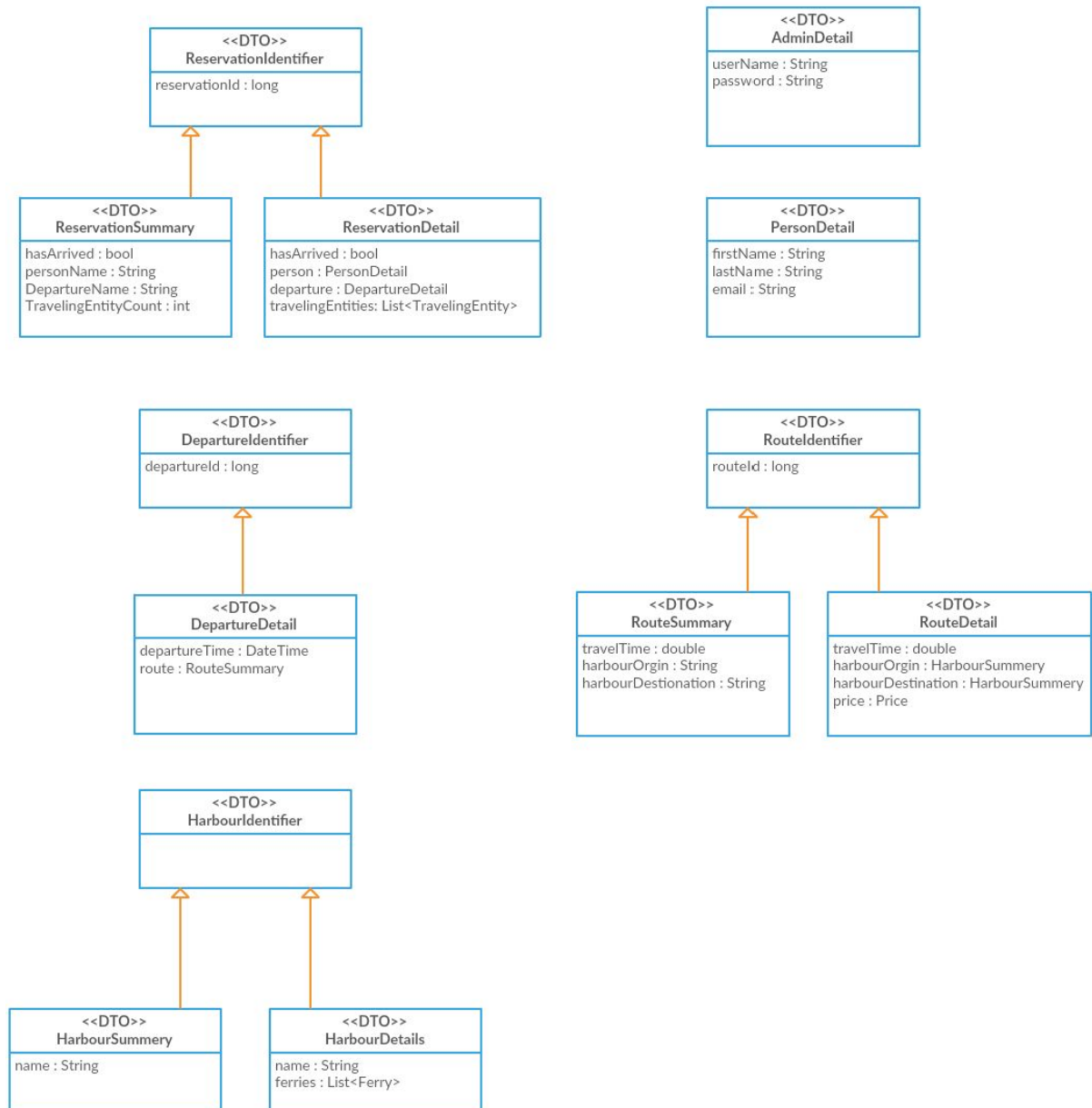**Success and condition:** A new departure is created.

Failed condition: The departure couldn't be created.
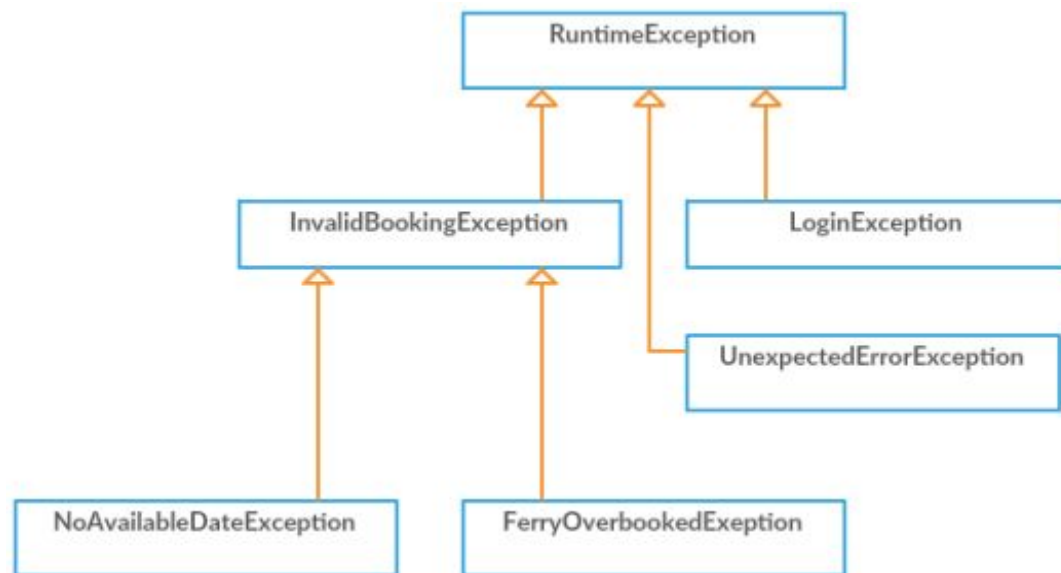
Success scenario:

- System shows a form for the user to fill
- Admin fills the form and presses submit
- System confirms the new route is created.
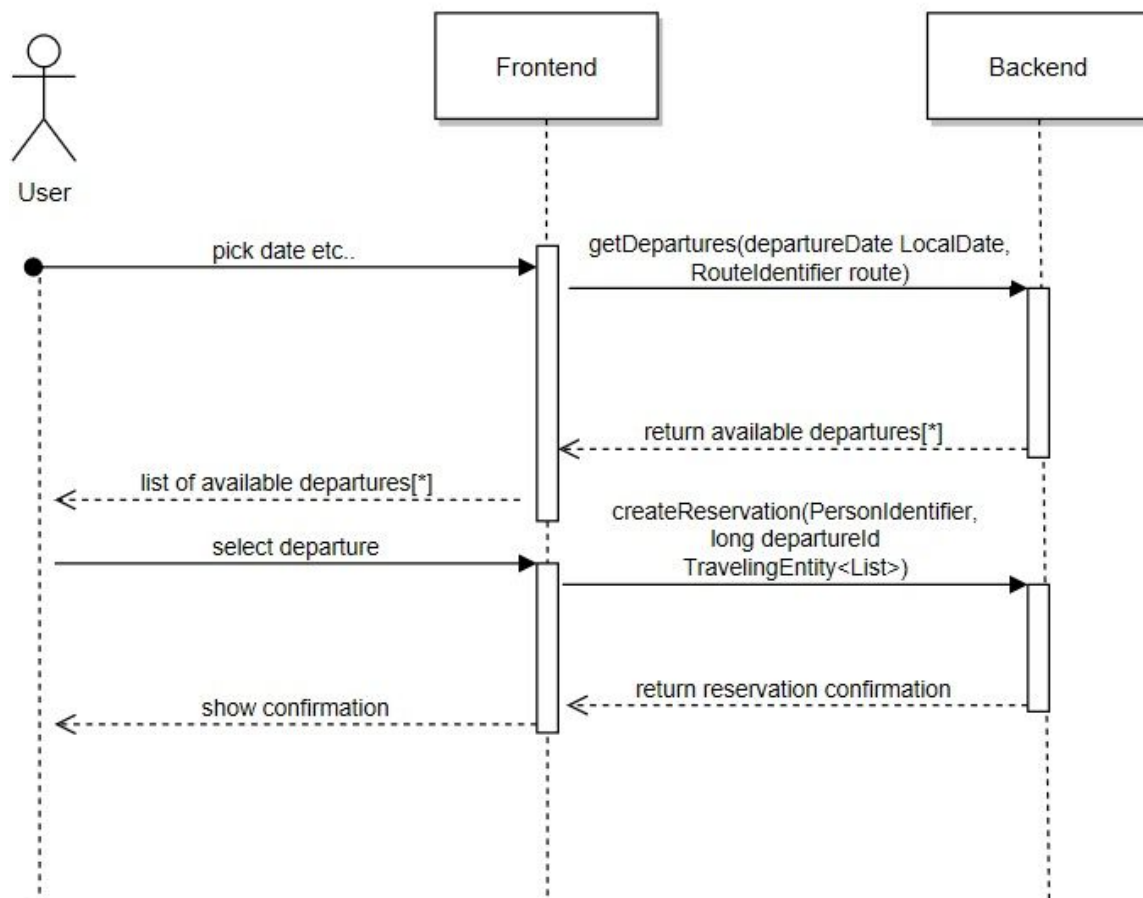- Admin sees and overview over the created route

Transfer objects

Data Transfer Objects (DTOs)

<<DTO>> ReservationIdentifier
reservationId : long

<<DTO>> AdminDetail
userName : String
password : String

<<DTO>> ReservationSummary
hasArrived : bool
personName : String
DepartureName : String
TravelingEntityCount : int

<<DTO>> ReservationDetail
hasArrived : bool
person : PersonDetail
departure : DepartureDetail
travelingEntities: List<TravelingEntity>

<<DTO>> PersonDetail
firstName : String
lastName : String
email : String

<<DTO>> DepartureIdentifier
departureId : long

<<DTO>> RouteIdentifier
routeId : long

<<DTO>> DepartureDetail
departureTime : DateTime
route : RouteSummary

<<DTO>> RouteSummary
travelTime : double
harbourOrgin : String
harbourDestionation : String

<<DTO>> RouteDetail
travelTime : double
harbourOrgin : HarbourSummery
harbourDestination : HarbourSummery
price : Price

<<DTO>> HarbourIdentifier

<<DTO>> HarbourSummery
name : String

<<DTO>> HarbourDetails
name : String
ferries : List<Ferry>

Exception Transfer Objects (ETOs)

Sequence diagram

Standard units

Date	Year-month-day-hour-minute-second
Amount	In dkk eg 1457 dkk
Travel time	in minutes eg 90 min = 1:30 hours
Weight	In kg eg 1750 kg

Git link

GIt code