Use case #1 - Create Reservation

- Goal: customer wants to create a reservation
- Scope: System under design
- Level: primary task (summary, primary task, sub-function)
- Pre condition: NONE
- Post condition:
    - Customer completes a reservation and is sent to view reservation page
    - Customer is presented with reservation not complete and reservation is not in the system
- Primary Actor: Customer

Success scenario

1. Customer initiates reservation
2. Customer chooses ferry line and date
3. System displays ferry line and its associated departure times
4. Customer selects preferred time
5. System asks for customer information
6. Customer fills necessary information
7. Customer submits information
8. System asks for confirmation
9. Customer confirms reservation

Related information

- Priority: High

Open issues

What happens when a departure gets cancelled, for an already existing reservation

Use case #2 - Delete Reservation

- Goal: customer deletes a reservation
- Scope: System
- Level: primary task (summary, primary task, sub-function)
- Pre condition: customer has a reservation
- Post condition:
    - Customer receives delete confirmation.
    - Customer receives no delete confirmation.
- Primary Actor: Any customer with a valid reservation
- Trigger: customer wants to delete reservation

Success scenario

1. customer fills in reservation ID
2. customer fills in email
3. customer clicks delete
4. System deletes reservation
5. customer gets confirmation

Related information

- Priority: High

Open issues

Use case #3 - View Reservation

- Goal: customer wants to view a reservation
- Scope: System under design
- Pre condition: NONE
- Post condition:
  - Customer can see reservation.
  - System cannot find the reservation and shows the customer an error
- Primary Actor: Customer

Success scenario
Customer selects view reservation
System asks for reservation id
System asks for email
Customer fills in reservation id
Customer fills in email
System finds the reservation and shows it to the customer

Related information

- Priority: Medium

Use case #4 ViewAllDepartures

- Goal: Customer can see all departures for a specific route
- Scope: customer
- Level: primary task (summary, primary task, sub-function)
- Pre condition: customer has navigated to the webpage
- Success end condition: customer receives a list of all departures for a specific route (empty/full)
- Failed end Condition: customer gets an error
- Primary Actor: Any customer / customer
- Trigger: customer navigates to view all departures

Success scenario

1. Customer navigates to webpage
2. A route is selected
3. Customer submits the selected route
4. Customer receives a list of all departures

Related information

- Priority: High

Open issues

Use case #5 - Create Schedule

- Goal: Admin wants to create a schedule
- Scope: System under design
- Pre condition: Admin is logged in
- Post condition:
    - Schedule has been created.
    - Schedule for day already exists
- Primary Actor: Customer

Success scenario

Admin selects create schedule
System asks for FerryLine info
Customer fills in FerryLine info
System asks for date of schedule
Customer fills in date of schedule
System asks for departure times
Customer fills in departure times
System asks for confirmation
Customer confirms schedule
System confirms creation of schedule

Related information

- Priority: High

Use case #6 - View Schedule

- Goal: Admin wants to view schedule
- Scope: System
- Level: sub-function (summary, primary task, sub-function)
- Pre condition: Admin is logged in. Schedule exists. Admin is navigated to search/choose FerryLine page.
- Success end condition: Admin picks a FerryLine and is sent to view schedules page
- Failed end Condition: Schedules not retrieved
- Primary Actor: Admin
- Trigger: Admin need to see schedules for a given FerryLine

Success scenario

1. Admin chooses FerryLine
2. UI navigates to page with schedules for the given FerryLine.

Related information

- Priority: Medium

Open issues

Use case #7 EditSchedule

- Goal: Admin can edit a schedule for a specific route
- Scope: Admin
- Level: Sub function (summary, primary task, sub-function)
- Pre condition: Admin is signed into the system
- Success end condition: The schedule is changed for the specific route
- Failed end Condition: The schedule is not changed
- Primary Actor: A registered administrator
- Trigger: Administrator wants to edit the schedule

Success scenario

1. Administrator navigates to page
2. A route is selected
3. The schedule is displayed
4. Administrator can add or remove departure times
5. Administrator save the changes
6. The schedule is updated

Related information

- Priority: High

Open issues

- A departure time is removed with registrations
- A departure time is postponed
- Can there be conflicting departures (harbour size, ferries)

Use case #8 - Delete Schedule

- Goal: Admin wants to delete a schedule
- Scope: System
- Level: sub-function (summary, primary task, sub-function)
- Pre condition: Admin is logged in. Schedule exists. Admin is navigated to search/choose FerryLine page.
- Success end condition: A given schedule, the Admin has chosen, has been deleted.
- Failed end Condition: Schedule not deleted
- Primary Actor: Admin
- Trigger: Admin need to delete a schedule for a given FerryLine

Success scenario

1. Admin chooses FerryLine.
2. UI navigates to page with schedules for the given FerryLine.
3. Admin picks a schedule and clicks on delete button.
4. UI prompts the customer with a confirmation-dialog.
5. customer agrees, and the schedule has been deleted.

Related information

- Priority: Medium

Open issues

Use case #9 - Create Ferry

- Goal: Admin wants to add a new ferry to the system
- Scope: System under design
- Pre condition: Admin is logged in
- Post condition:
    - Schedule has been created.
    - Schedule for day already exists
- Primary Actor: Customer

Success scenario

1. Admin selects create ferry
2. System asks for Ferry serial number
3. System asks for Ferry name
4. Customer fills in Serial number
5. Customer fills in Ferry name
6. Systems asks for confirmation
7. Customer confirms info

Related information

- Priority: High

Use case #10 - View Ferry

- Goal: Admin wants to view a Ferry
- Scope: System
- Level: sub-function (summary, primary task, sub-function)
- Pre condition: Admin is logged in. Ferry exists. Admin is navigated to search/choose Ferry page.
- Success end condition: Admin picks a Ferry and is sent to view Ferry page
- Failed end Condition: Ferry not retrieved
- Primary Actor: Admin
- Trigger: Admin need to view a given Ferry

Success scenario

1. Admin chooses Ferry.
2. UI navigates to page with Ferry

Related information

- Priority: Medium

Open issues

Use case #11 EditFerry

- Goal: Admin can edit a ferry
- Scope: Admin
- Level: Primary task (summary, primary task, sub-function)
- Pre condition: Admin is signed into the system
- Success end condition: The ferry docked state or lent state can are edited
- Failed end Condition: The states are not changed
- Primary Actor: A registered administrator
- Trigger: Administrator wants to edit a ferry

Success scenario

1. Administrator navigates to page
2. Ferry is selected
3. Administrator changed the fields
4. Administrator save the changes
5. The Docked state or Lent state is updated

Related information

- Priority: High

Open issues

Use case #12 - Delete Ferry

- Goal: Admin deletes a ferry
- Scope: System
- Level: primary task (summary, primary task, sub-function)
- Pre condition: Admin logged in and system has a ferry.
- Success end condition: Admin receives delete confirmation.
- Failed end Condition: Admin receives no delete confirmation.
- Primary Actor: Admin
- Trigger: Admin needs to delete a ferry

Success scenario

1. Admin fills in ferry serial number
2. Admin clicks delete
3. System deletes ferry
4. Admin gets confirmation

Related information

- Priority: High

Open issues

Use case #13 - Remove Ferry

- Goal: Admin removes ferry from line
- Scope: System
- Level: primary task (summary, primary task, sub-function)
- Pre condition: Admin logged in and system has a route and a ferry.
- Success end condition: Admin receives remove confirmation.
- Failed end Condition: Admin receives no remove confirmation.
- Primary Actor: Admin
- Trigger: Admin needs to delete a ferry

Success scenario

1. Admin fills in ferry info
2. Admin fill in route info
3. Admin clicks remove
4. System removes ferry from line
5. Admin gets confirmation

Related information

- Priority: High

Open issues

Use case #14 - Add Ferry

- Goal: Admin adds a ferry to line
- Scope: System
- Level: primary task (summary, primary task, sub-function)
- Pre condition: Admin logged in and system has a route and a ferry.
- Success end condition: Admin receives confirmation.
- Failed end Condition: Admin receives no add confirmation.
- Primary Actor: Admin
- Trigger: Admin needs to add a ferry to line

Success scenario

1. Admin fills in ferry serial number
2. Admin fills in route info
3. System adds ferry to line
4. Admin gets confirmation

Related information

- Priority: High

Open issues

customer

Create reservation
Collection<DepartureTime> GetAllDepartures(Date, FerryLine)
Info on full departures as well.
DepartureTimes gives dates and VehicleTypes for ferry.
InvalidDateException
InvalidRouteException

Reservation createReservation(PersonInfo, Route, DepartureTime, Vehicle, VehiclePassengers, WalkOns)
ReservationInfo returns all the above along with price, reservationDate and id if succesfull.
NoVacanciesException
InvalidDataException

Boolean ConfirmReservation(ReservationId)
InvalidReservationIdException

Delete Reservation
Id DeleteReservation(ReservationId, Email)
Return id if succesful
InvalidReservationIdException
InvalidEmailException
InvalidReservationException

View Reservation
Reservation ViewReservation(ReservationId, email)
InvalidReservationIdException
InvalidEmailException
ReservationNotFoundException

ViewAllDepartures
GetAllDepartures method from CreateReservation

Admin

Create schedule
Schedule FindSchedule(FerryLineInfo, Date)
Finds schedule if existing, creates it if it does not.
Schedule includes id's and timestamps.
FerryLineNotFoundException
InvalidDateException

UpdateSchedule(Coll(DepartureTimes), ScheduleId)
<u>ScheduleIdNotFoundException</u>
<u>InvalidDepartureTimeException</u>

View Schedule
Uses findSchedule in create schedule

Edit Schedule
Uses UpdateSchedule in create schedule

Delete Schedule
ScheduleId DeleteSchedule(Id)
ScheduleNotFoundException.

Create Ferry
Ferry CreateFerry(Ferry, FerryConfig[])
<u>DuplicateFerryException</u>
<u>InvalidDataException</u>

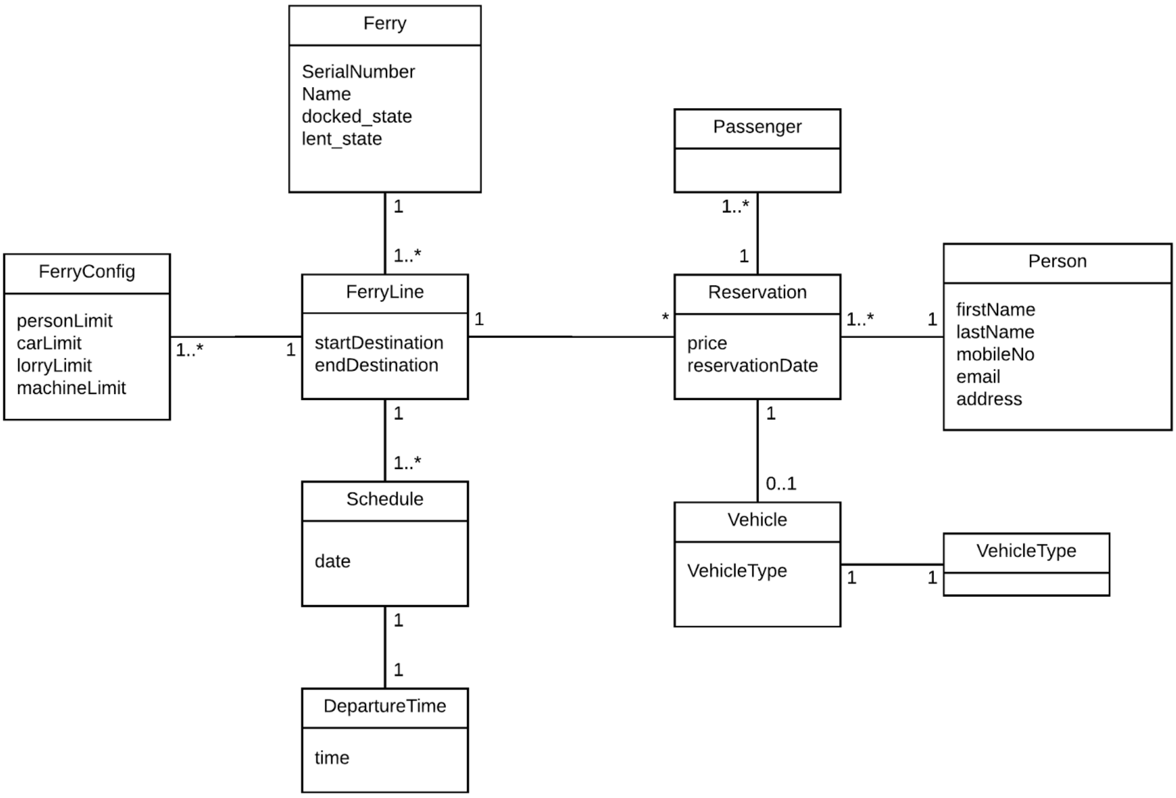
View Ferry
Ferry ViewFerry(SerialNumber)
<u>FerryNotFoundException</u>

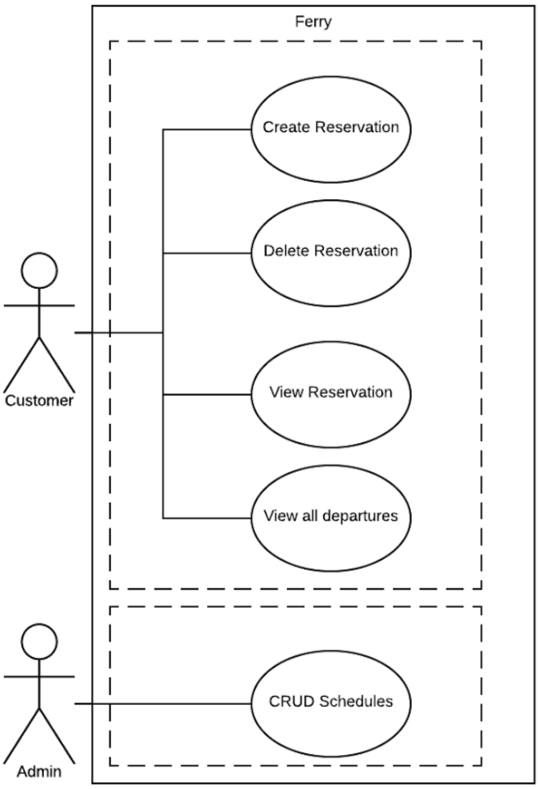
Update Ferry
UpdateFerry(SerialNumber, Dockstate, lendstate)
<u>FerryNotFoundException</u>
<u>InvalidStateException</u>

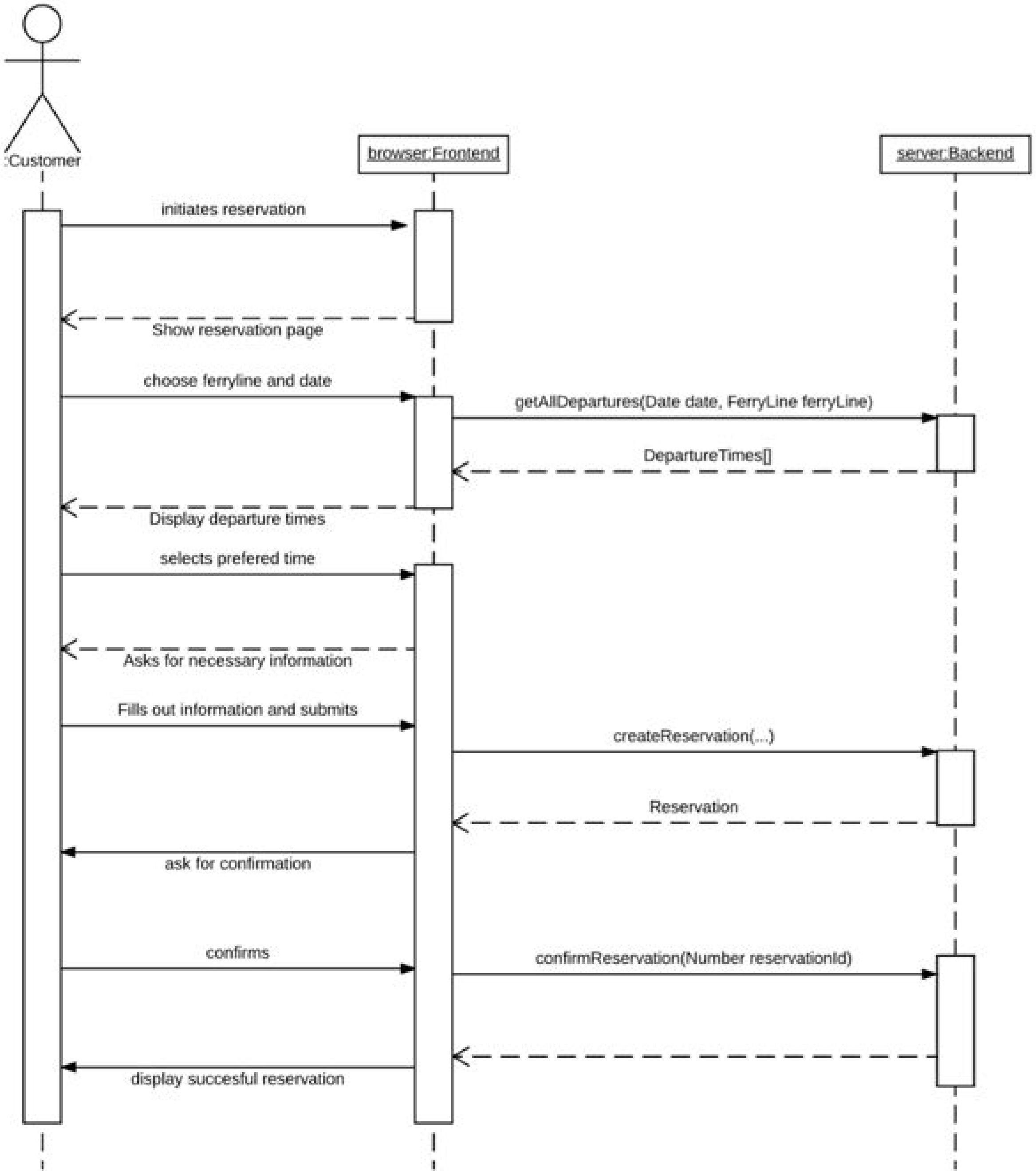
Delete Ferry
SerialNumber DeleteFerry(SerialNumber)
<u>FerryNotFoundException</u>

AddFerry
SerialNumber AddFerry(RouteId, serialNo)
<u>FerryNotFoundException</u>
<u>InvalidRouteException</u>
<u>FerryAlreadyOnRouteException</u>

Remove Ferry
SerialNo RemoveFerry(RouteId, SerialNo)
<u>FerryNotFoundException</u>
<u>InvalidRouteException</u>
<u>FerryNotOnRouteException</u>


Ferry
SerialNumber
Name
docked_state
lent_state
1
1..*
FerryConfig
personLimit
carLimit
lorryLimit
machineLimit
1..*
1
FerryLine
startDestination
endDestination
1
*
Passenger
1..*
1
Reservation
price
reservationDate
1..*
1
Person
firstName
lastName
mobileNo
email
address
1
1..*
Schedule
date
1
1
DepartureTime
time
1
0..1
Vehicle
VehicleType
1
1
VehicleType


Ferry
Create Reservation
Delete Reservation
View Reservation
View all departures
Customer
CRUD Schedules
Admin


:Customer
browser:Frontend
server:Backend
initiates reservation
Show reservation page
choose ferryline and date
getAllDepartures(Date date, FerryLine ferryLine)
DepartureTimes[]
Display departure times
selects prefered time
Asks for necessary information
Fills out information and submits
createReservation(...)
Reservation
ask for confirmation
confirms
confirmReservation(Number reservationId)
display succesful reservation