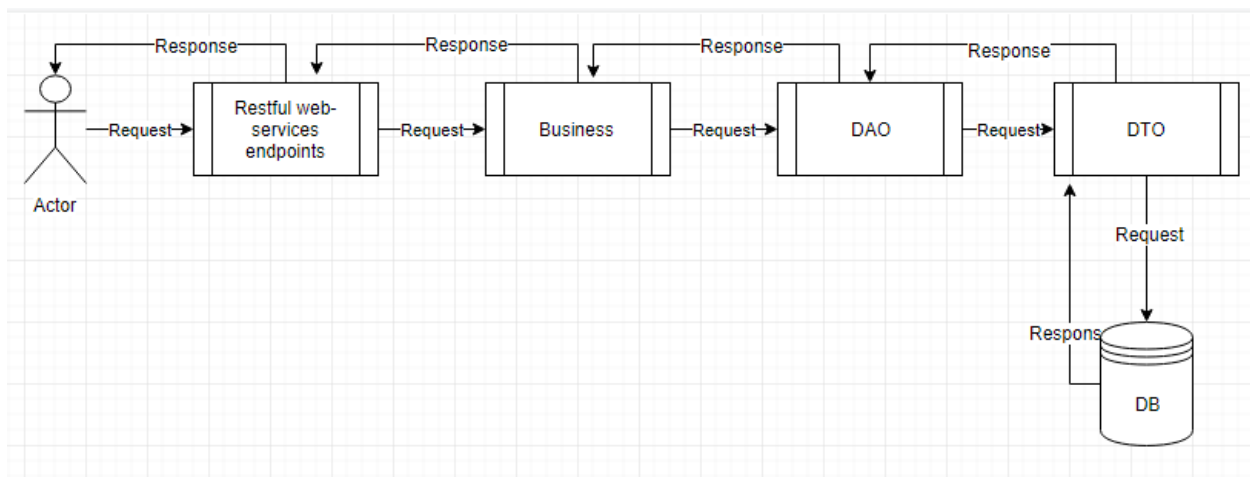


Service Description:

Unidirectional voice notes service which implemented in **JAVA** which allows:

- As a customer
 - Subscribe to a “journey” in order to receive voice-notes from the captain running this journey.
 - Receive multiple voice notes from the same captain as long as the rider is still waiting for the bus
 - Listen to voice note
- As a captains
 - Send voice notes to all customers who have booked this journey but currently waiting in the next bus stops
 - The ability to know how many customers received the voice note and how many of them have listened to the voice note

Architecture Description:



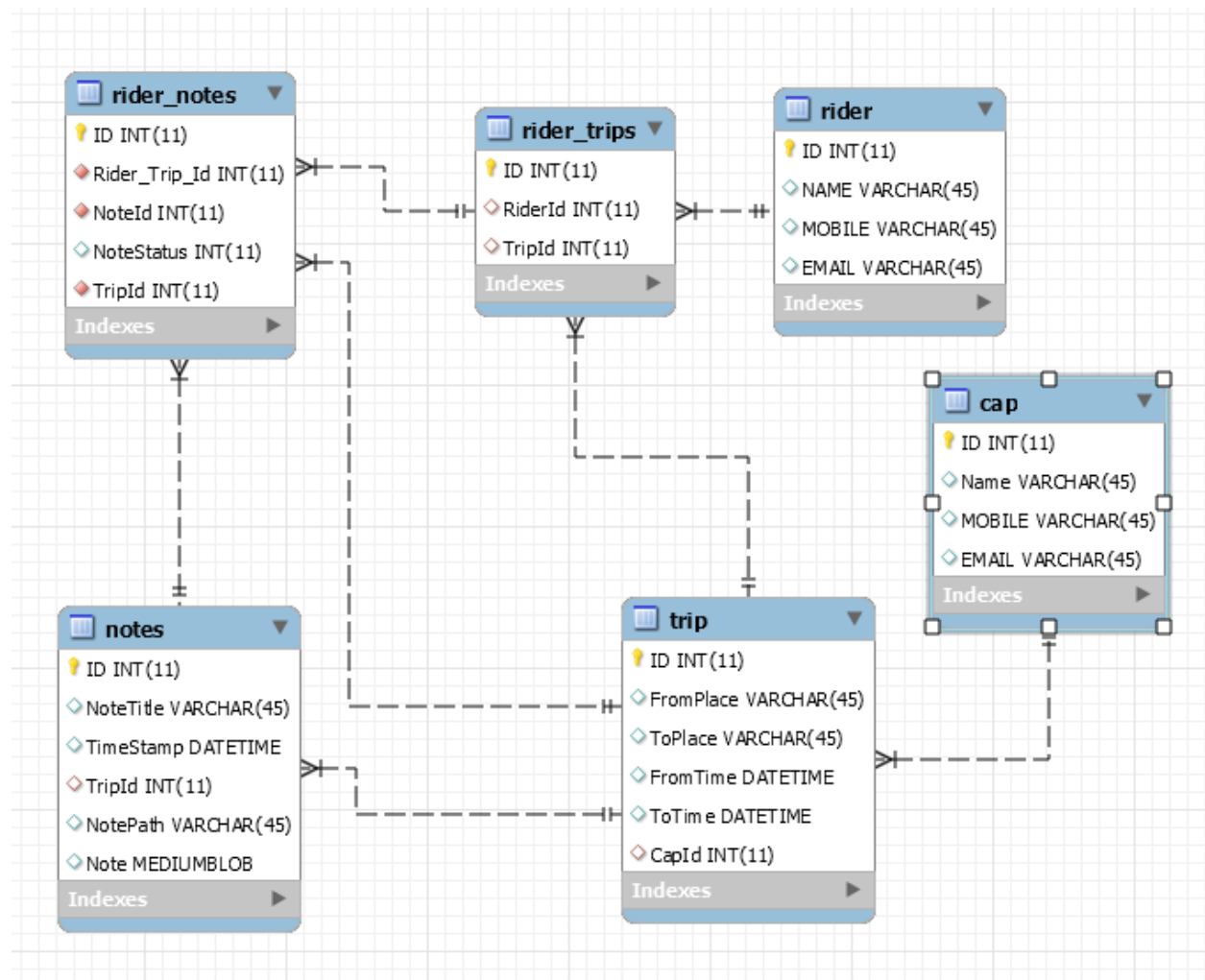
System actors will interact with Restful service which act as service interface and each service will be responsible for specific task like one for sending notes to riders , listen captain notes and etc...

Each service will call specific façade which responsible for gather all business related to each service. And I prefer to be implemented using EJBs but for simplicity of deployment I prefer to use regular java class

DAO act as intermediate liar btw business and DB which act as maintain transaction (persist , merge , etc...) over DB

DTO implemented using hibernate and act as DB tables representation which used by java class to interact with DB easily.

DB Diagram:



Above diagram represent DB tables used for this service and relation btw each on as forging key

So in my code I take care of transaction lifecycle because as you see each table has forging key for another table so we should make sure all entities inserted correctly so **I configured server data source to be read committed transactions**

Services Description:

| Service name | | Service Details |
|-----------------------------------------|--------------|-------------------------------------------------------------------------------------------------------|
| careem/captain/get-notes | Type: | Get |
| | Produce: | List<NoteDto> (JSON) |
| | Consume: | No thing |
| | Query prams: | tripId (int) |
| | Description: | used to return all notes related to specific trip with status (how many received , readied or sent) |
| careem/captain/add-note | Type: | Post |
| | Produce: | int (JSON) - note id |
| | Consume: | NoteDto (JSON) |
| | Query prams: | No thing |
| | Description: | used by captain to send note for rider trip |
| careem/rider/get-available-trips | Type: | Get |
| | Produce: | List<TripDto> (JSON) |
| | Consume: | No thing |
| | Query prams: | riderId (int) |
| | Description: | used to return all available trips which rider can join it |
| careem/rider/join-trip | Type: | Get |
| | Produce: | Confirmation Msg (String - JSON) |
| | Consume: | No thing |
| | Query prams: | tripId (int) && riderId (int) |
| | Description: | used by rider to join specfic trip (depends on previous service) |
| careem/rider/get-notes | Type: | Get |
| | Produce: | List<NoteDto> (JSON) |
| | Consume: | No thing |
| | Query prams: | riderId (int) |
| | Description: | used by rider to return all unread notes |

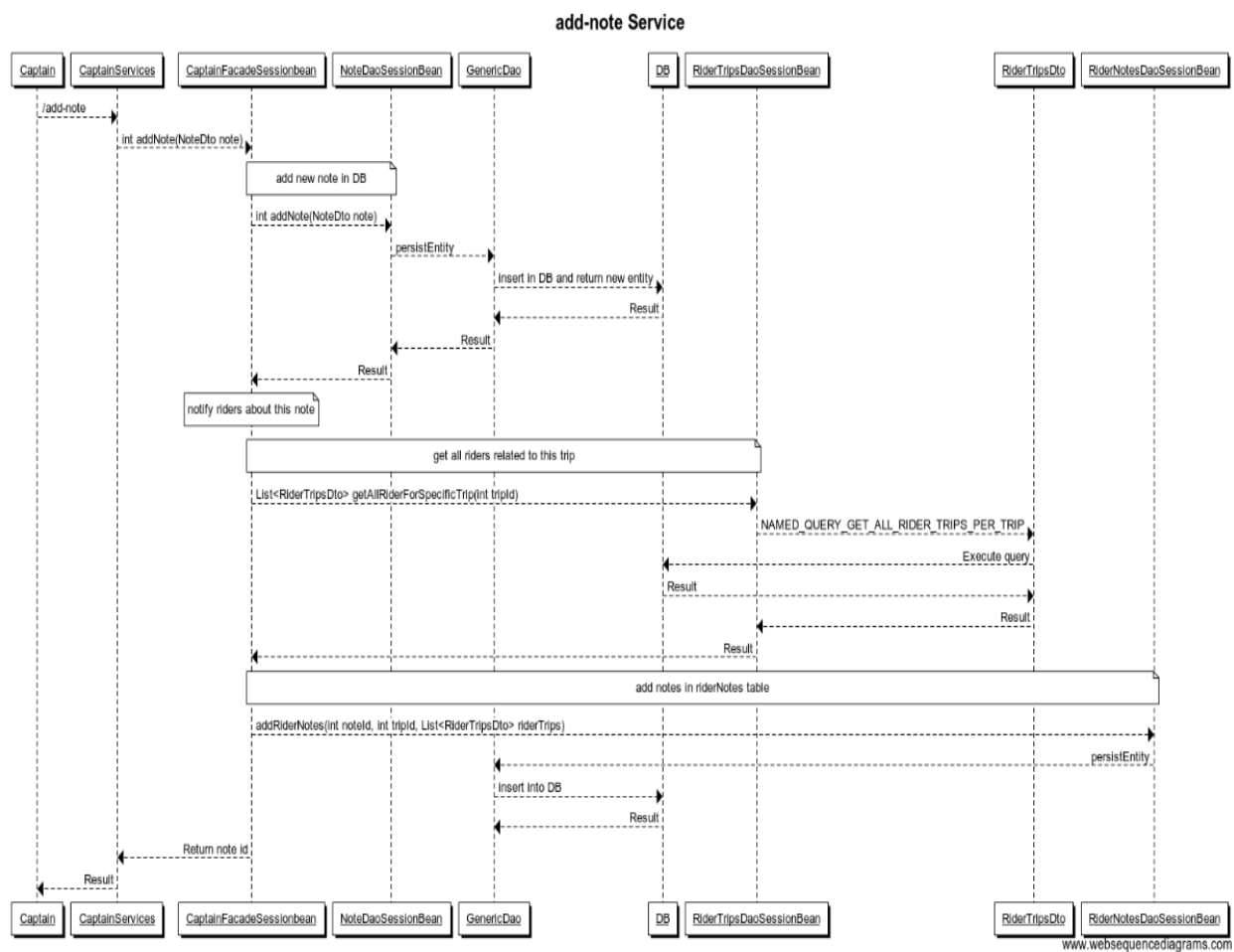
| | | |
|-------------------------------|--------------|-----------------------------------------------|
| careem/rider/read-note | Type: | Pose |
| | Produce: | RiderNotesDto (JSON) |
| | Consume: | No thing |
| | Query prams: | noteId (int) && tripId (int) && riderId (int) |
| | Description: | used by rider to read specific note |

And to get some services call samples u can follow bellow path:

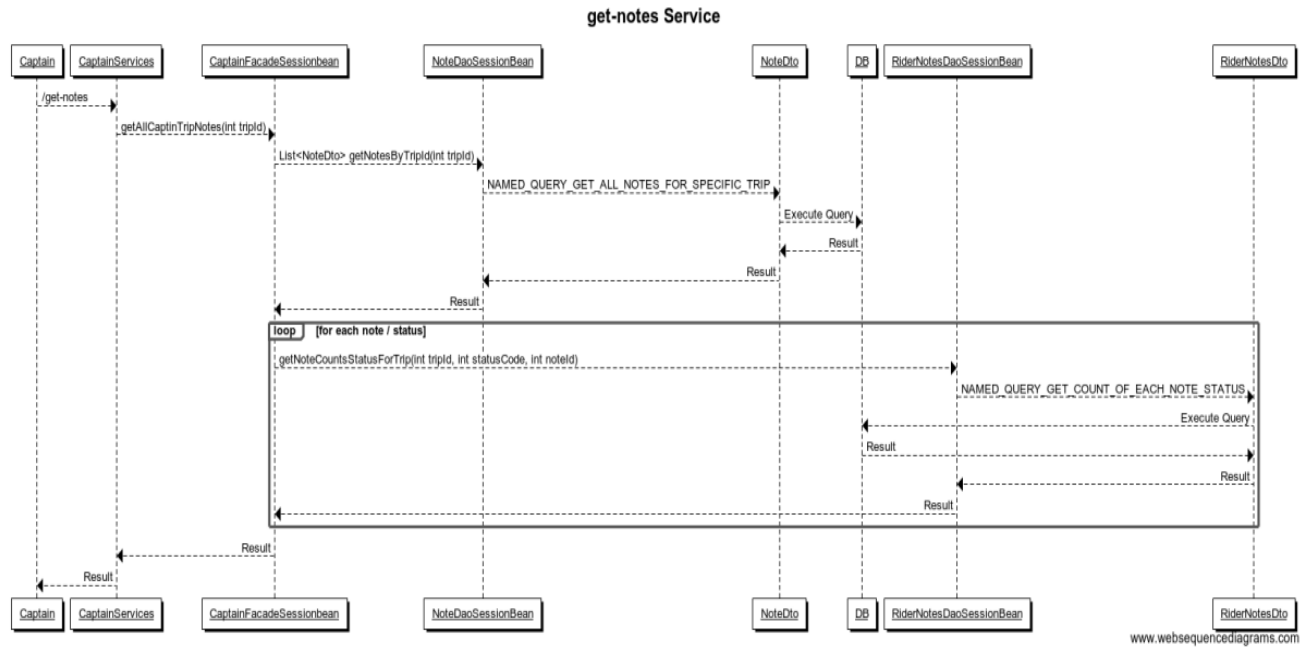
- [voice_note_service\service-documents\Restful\RequestSamples](#)

Service sequence diagram:

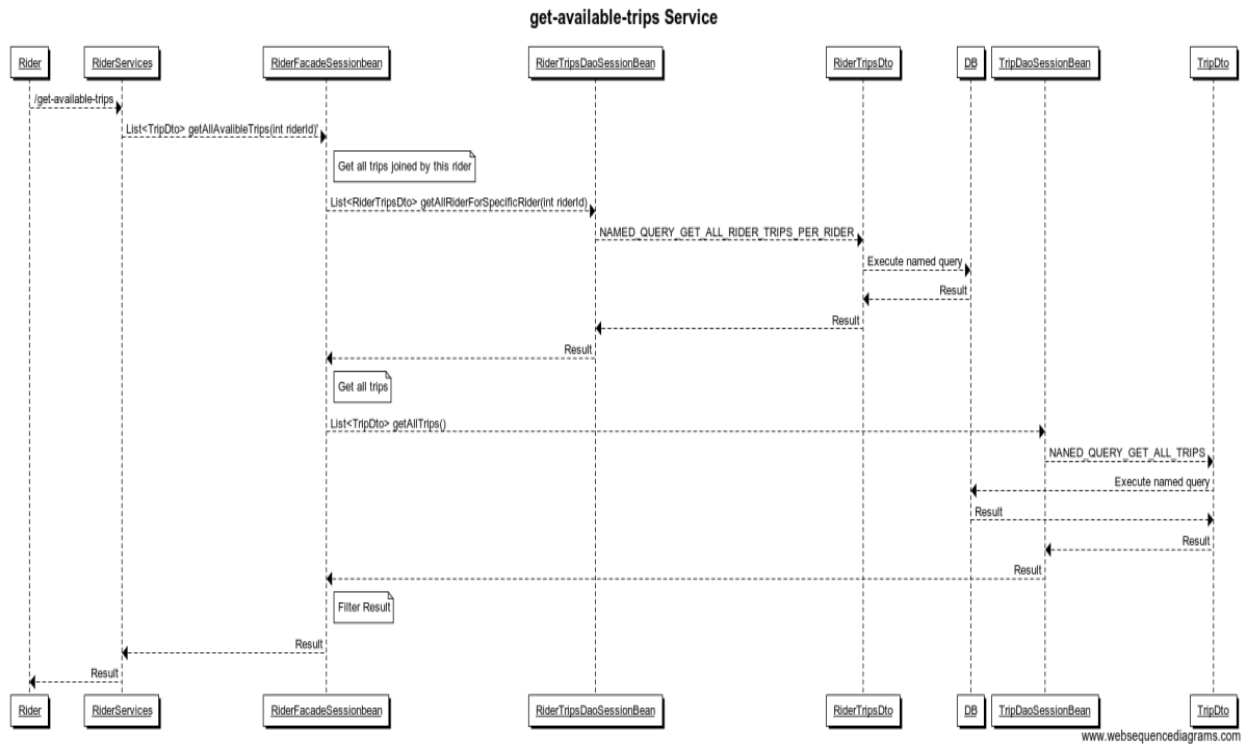
careem/captain/add-note



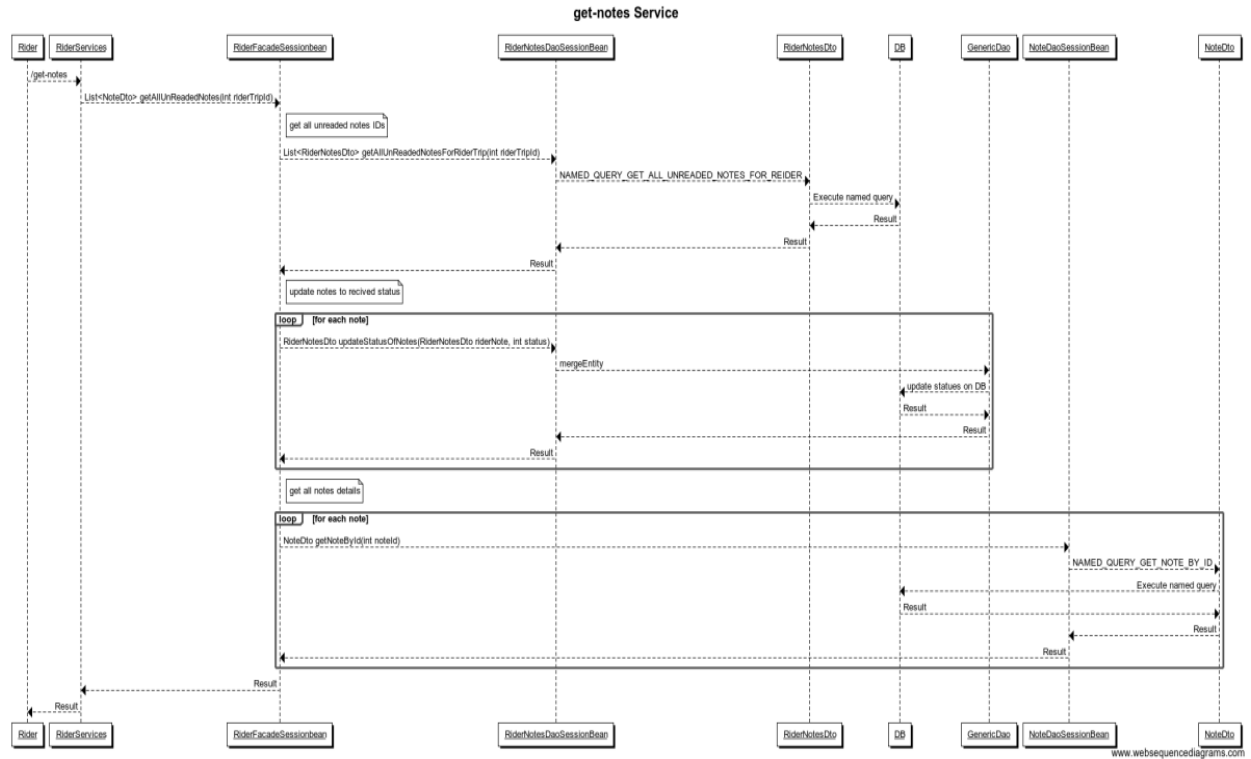
careem/captain/get-notes



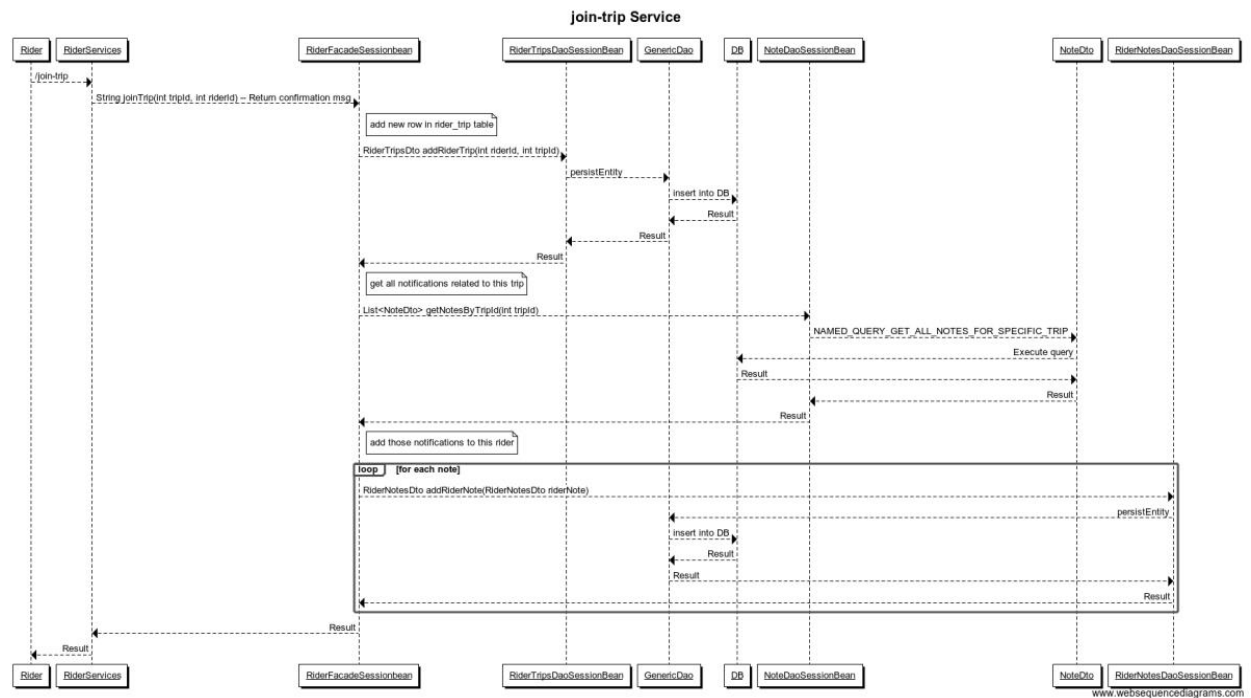
careem/rider/get-available-trips



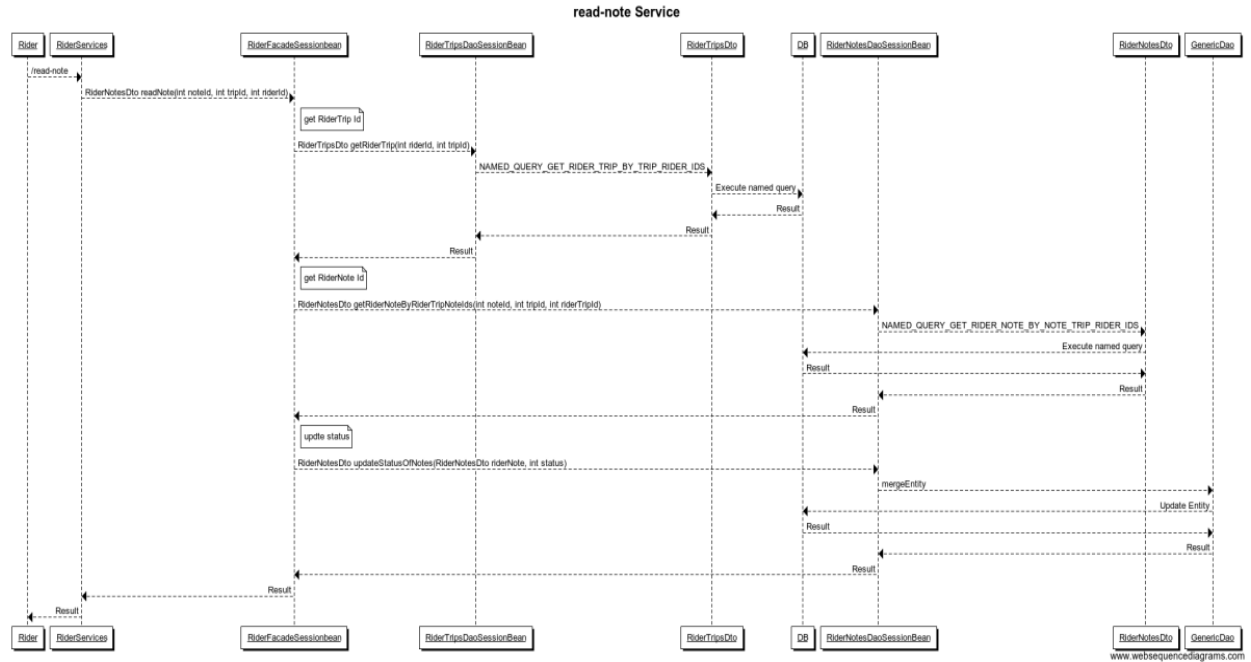
careem/rider/get-notes



careem/rider/join-trip



careem/rider/read-note



Technologies & Tools:

- MySQL DB
- Java EE 8
 - JPA (Hibernate)
 - RESTful web-services
- Maven
- Eclipse photon (IDE)
- Tomcat server