TRANSPORT-INFO

Technical Solution Description

Annotation

The aim of this study project is to create an application simulating transportation company activity: keeping information of trucks, drivers, cargoes, depots and making deliveries to clients.

Additional features: interactive map, usage of route optimization API and driver's page adjusted for usage on smartphones.

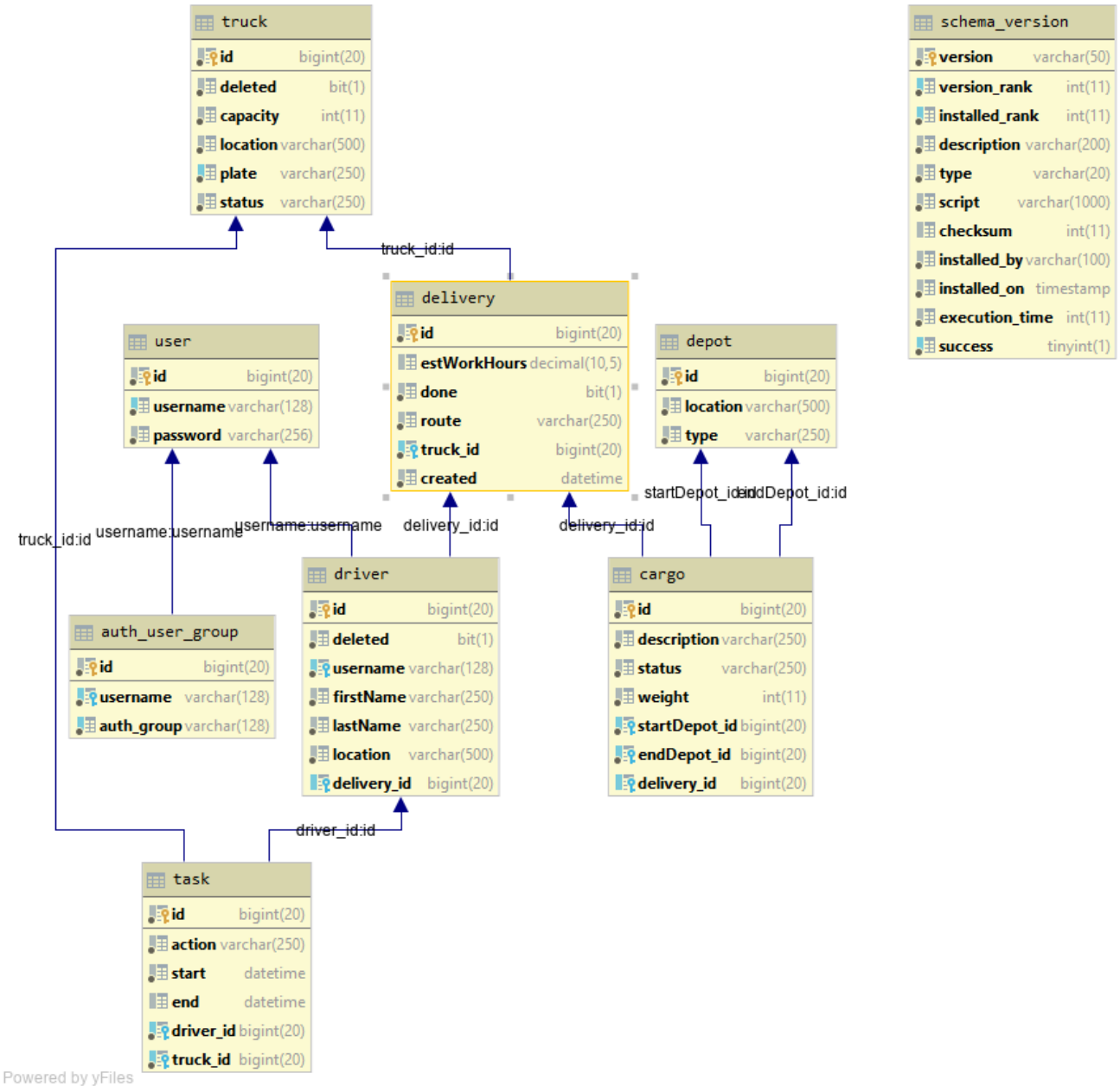
Nikolaev Vasily
vsnikalex@gmail.com

Contents

Used Technologies and Frameworks 3
Database Scheme 4
Models Implementation 5
GraphHopper library Geocoding Entry 5
Depot 5
Cargo 6
Truck 6
Driver 6
Delivery 7
Application Modules and Deployment 7
Maven modules 7
Docker containers 8
Mobile Device Screenshots 9
Sonar Report and Unit Tests 10
Logging 11
Possible Improvements 12

Used Technologies and Frameworks

Database Schema



Models Implementation

GraphHopper library Geocoding Entry

GHGeocodingEntry	
street	String
point	Point
houseNumber	String
state	String
extent	Double[]
osmType	String
osmId	Long
name	String
osmKey	String
extendBBox	BBox
country	String
osmValue	String
postcode	String
city	String

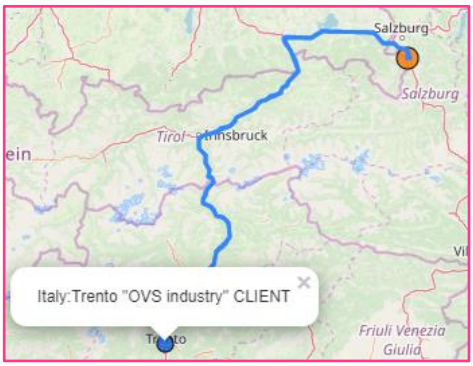
OpenStreetMap crowdsourced data can be stored in this object. It is used for geocoding of address and place names, as well as for route planning.

Depot

Depot	
id	Long
location	GHGeocodingEntry
type	DepotType

Cargo

Cargo	
description	String
endDepot	Depot
id	Long
startDepot	Depot
delivery	Delivery
weight	int
status	CargoStatus



Truck

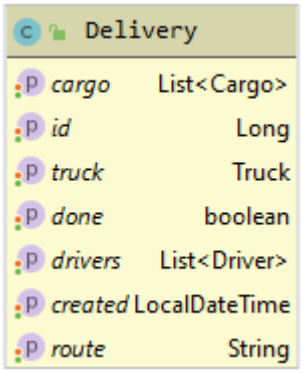
Truck	
capacity	int
plate	String
id	Long
location	GHGeocodingEntry
status	TruckStatus
delivery	Delivery

Driver

Driver	
lastName	String
tasks	List<Task>
id	Long
location	GHGeocodingEntry
username	String
firstName	String
delivery	Delivery

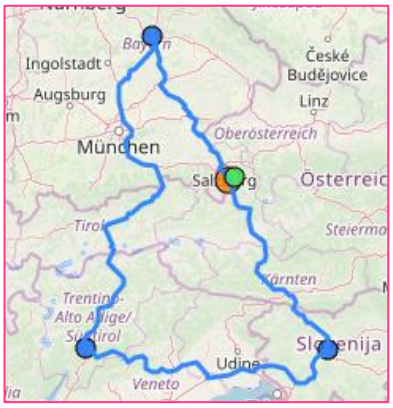
Task	
id	Long
truck	Truck
start	LocalDateTime
driver	Driver
action	DriverAction
end	LocalDateTime

Delivery



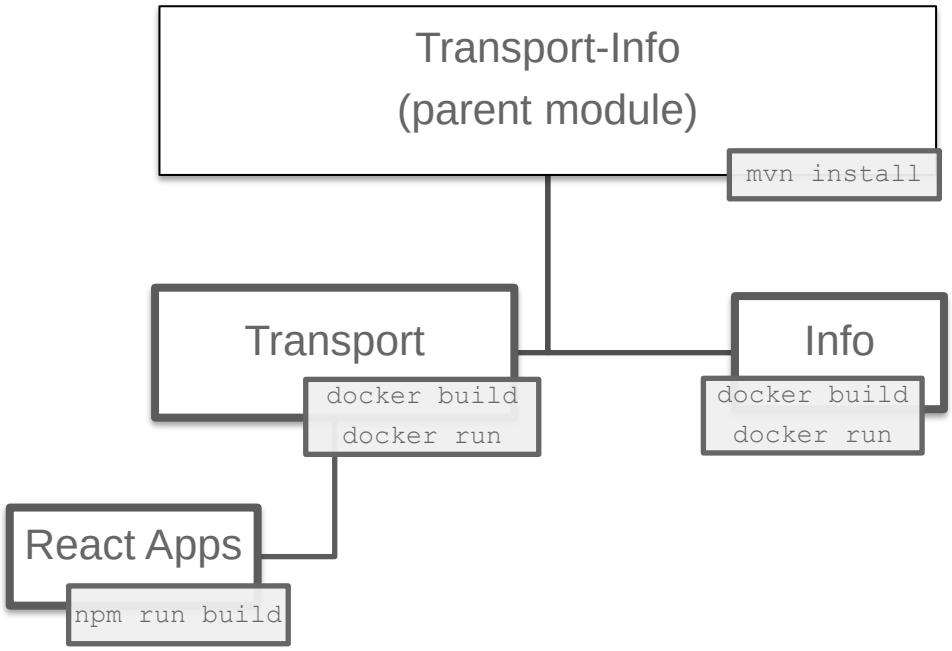
route

```
{
"0" : "47.686662,13.098481",
"1" : "46.068837,14.489973",
"2" : "46.094432,11.115757",
"3" : "49.01333,12.05184",
"4" : "47.686662,13.098481"
}
```

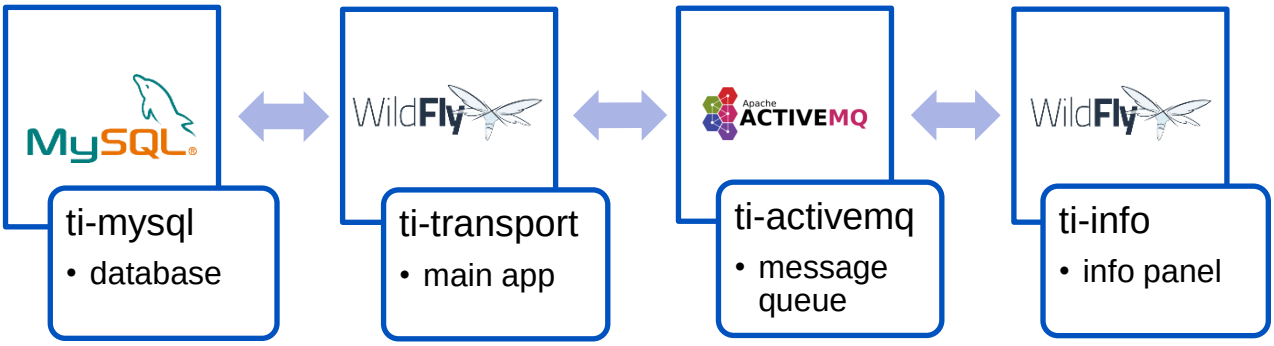


Application Modules and Deployment

Maven modules



Docker containers

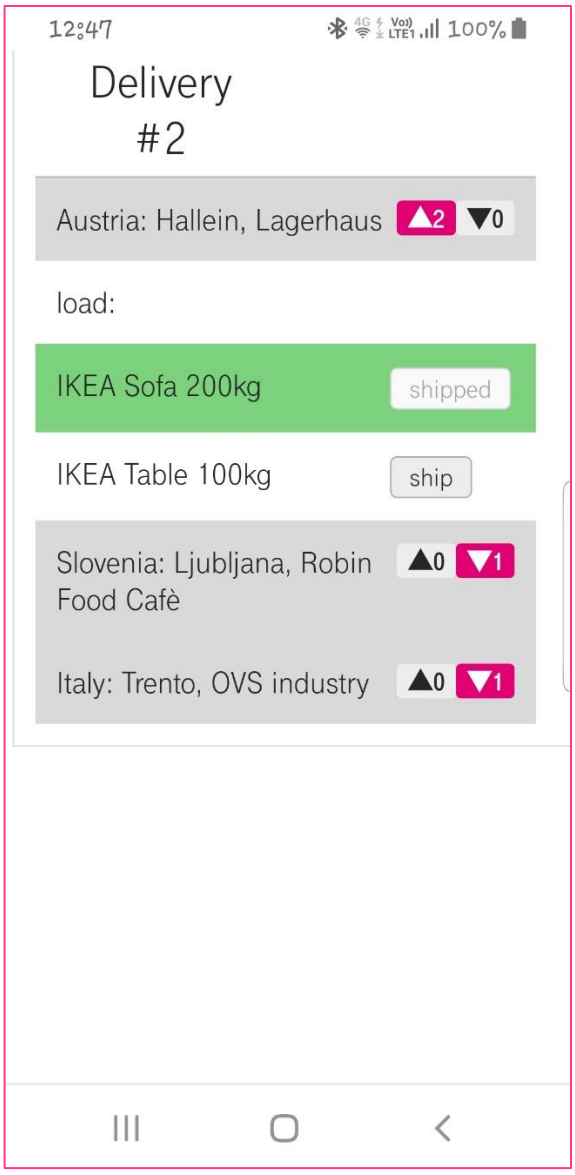


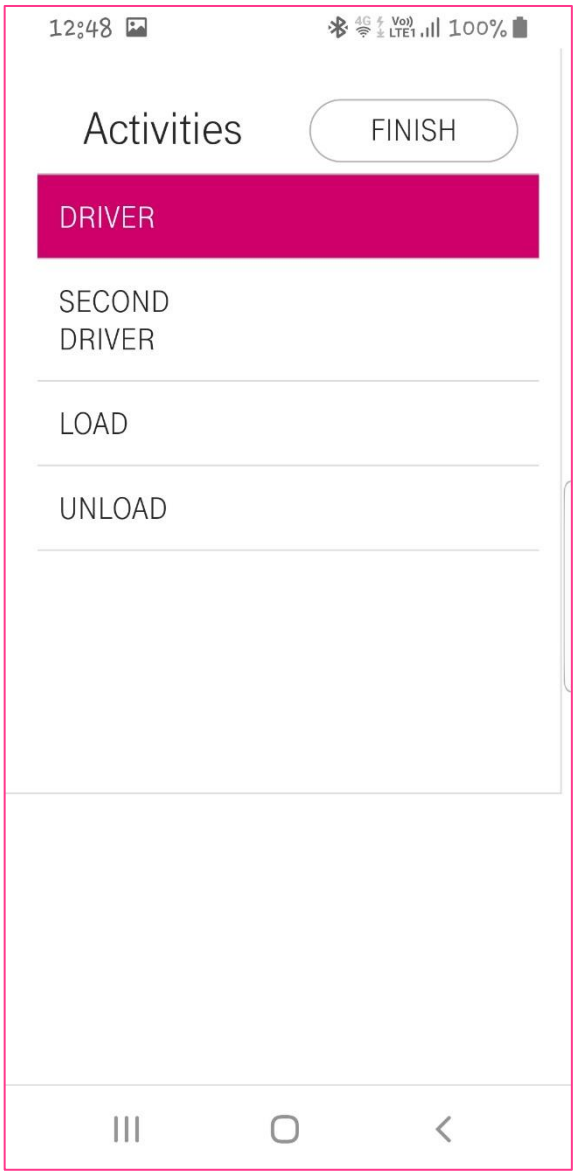
Deployed with Docker Compose:

```yaml
services:
  database:
    image: mysql
    hostname: ti-mysql
    container_name: ti-mysql
    environment:
      MYSQL_DATABASE: 'transportinfo'
      MYSQL_USER: 'admin'
      MYSQL_PASSWORD: 'admin'
      MYSQL_ROOT_PASSWORD: 'root'
    ports:
      - '3306:3306'
  message-queue:
    image: vromero/activemq-artemis
    hostname: ti-activemq
    container_name: ti-activemq
    environment:
      ARTEMIS_USERNAME: 'admin'
      ARTEMIS_PASSWORD: 'admin'
    ports:
      - '8161:8161'
      - '61616:61616'
```

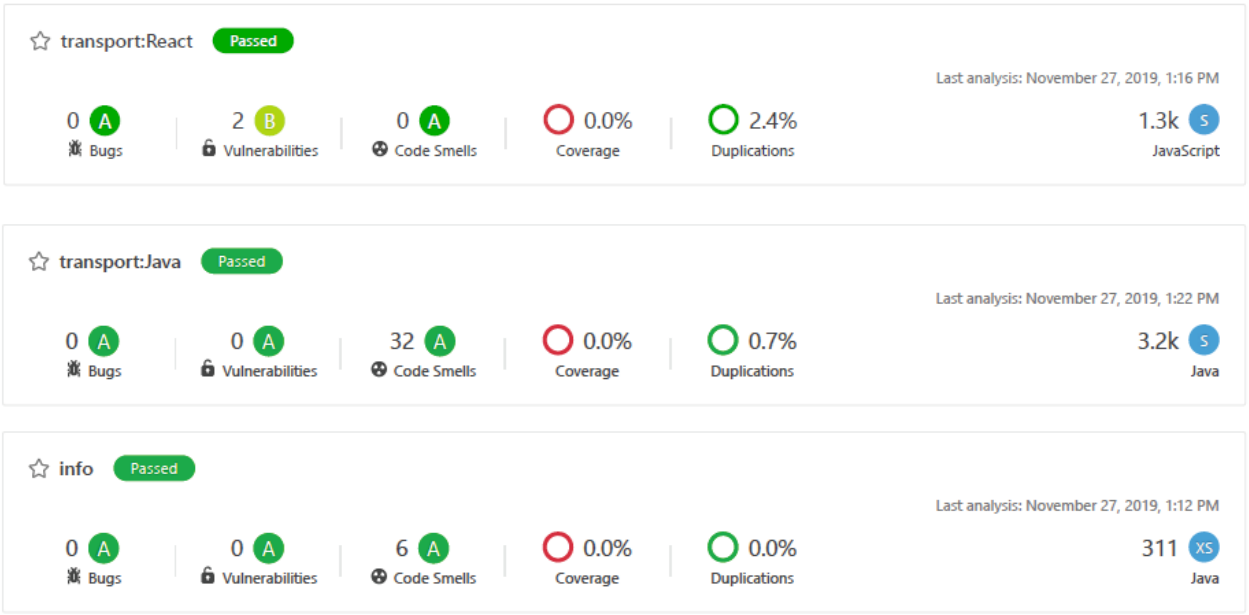
```yaml
info:
  build: info/
  hostname: ti-info
  container_name: ti-info
  depends_on:
    - message-queue
  ports:
    - '8080:8080'
transport:
  build: transport/
  hostname: ti-transport
  container_name: ti-transport
  depends_on:
    - database
    - message-queue
  environment:
    spring.profiles.active: 'docker'
    dbserver: 'ti-mysql'
    dbport: '3306'
    dbuser: 'admin'
    dbpassword: 'admin'
  ports:
    - '3333:8080'
```

Mobile Device Screenshots





Sonar Report and Unit Tests



30% classes, 19% lines covered in package 'com.tsystems.transportinfo.service'

Element	Class, %	Method, %	Line, %
CargoServiceImpl	100% (1/1)	72% (8/11)	65% (26/40)
DeliveryServiceImpl	0% (0/1)	0% (0/8)	0% (0/49)
DepotServiceImpl	0% (0/1)	0% (0/5)	0% (0/12)
DriverServiceImpl	0% (0/1)	0% (0/11)	0% (0/49)
GeoServiceImpl	100% (1/1)	75% (6/8)	58% (25/43)
JmsService	0% (0/1)	0% (0/2)	0% (0/8)
StatServiceImpl	0% (0/1)	0% (0/8)	0% (0/47)
TaskServiceImpl	0% (0/1)	0% (0/9)	0% (0/30)
TransportUserDetailsServiceImpl	0% (0/1)	0% (0/3)	0% (0/16)
TruckServiceImpl	100% (1/1)	40% (4/10)	34% (14/41)

Logging

```
<!--  Logging  -->
<dependency>
    <groupId>org.slf4j</groupId>
    <artifactId>slf4j-api</artifactId>
    <version>1.7.28</version>
</dependency>
<dependency>
    <groupId>org.slf4j</groupId>
    <artifactId>slf4j-jdk14</artifactId>
    <version>1.7.28</version>
</dependency>
<dependency>
    <groupId>log4j</groupId>
    <artifactId>log4j</artifactId>
    <version>1.2.17</version>
</dependency>
```

```
# Root logger option
log4j.rootLogger=INFO, file, stdout

# Direct log messages to a log file
log4j.appender.file=org.apache.log4j.RollingFileAppender
log4j.appender.file.File=logging.log
log4j.appender.file.MaxFileSize=10MB
log4j.appender.file.MaxBackupIndex=10
log4j.appender.file.layout=org.apache.log4j.PatternLayout
log4j.appender.file.layout.ConversionPattern=%d{yyyy-MM-dd HH:mm:ss} %-5p %c{1}:%L -
%m%n
```

```
2019-11-27 18:28:33 INFO  StatServiceImpl:67 - Received 2 deliveries
2019-11-27 18:28:34 INFO  DepotServiceImpl:48 - Request DepotDAO to find Depot with
coordinates = 47.686662,13.098481
2019-11-27 18:28:34 INFO  DepotServiceImpl:48 - Request DepotDAO to find Depot with
coordinates = 46.068837,14.489973
2019-11-27 18:28:34 INFO  DepotServiceImpl:48 - Request DepotDAO to find Depot with
coordinates = 46.094432,11.115757
2019-11-27 18:28:34 INFO  DepotServiceImpl:48 - Request DepotDAO to find Depot with
coordinates = 47.686662,13.098481
2019-11-27 18:28:34 INFO  DepotServiceImpl:48 - Request DepotDAO to find Depot with
coordinates = 48.752525,18.145055
2019-11-27 18:28:34 INFO  DepotServiceImpl:48 - Request DepotDAO to find Depot with
coordinates = 46.094432,11.115757
2019-11-27 18:28:34 INFO  DepotServiceImpl:48 - Request DepotDAO to find Depot with
coordinates = 48.752525,18.145055
2019-11-27 18:28:34 INFO  SoapAspect:48 - DeliveryList sent
2019-11-27 18:28:34 INFO  ReflectionServiceFactoryBean:390 - Creating Service
{http://soap.transportinfo.tsystems.com/}NotificationsServiceLocal from WSDL:
http://ti-info:8080/info/NotificationsServiceLocal?wsdl
2019-11-27 18:28:34 INFO  SoapAspect:26 - DriversStat sent
2019-11-27 18:28:34 INFO  ReflectionServiceFactoryBean:390 - Creating Service
{http://soap.transportinfo.tsystems.com/}NotificationsServiceLocal from WSDL:
http://ti-info:8080/info/NotificationsServiceLocal?wsdl
2019-11-27 18:28:34 INFO  SoapAspect:37 - TrucksStat sent
```

Possible Improvements

Frontend:
- pagination, jquery.dataTables
- error pages
- load bars instead of percentages at Delivery Editor
- corporative-style alerts
- map icons: depot, cargoes, truck
- bubble markers: admin_cargoes
- loading bar during routing optimization

Functionality:
- add new drivers, trucks, depots on map, address field
- Driver App: timer
- more information about drivers at admin_trucks
- usage of GPS data on maps to track drivers' location
- Delivery Editor: customize max travel duration for incoming trucks
- Delivery Editor: customize start datetime and mode for driver work
- truck transfer without cargo
- automobile maintenance schedule
- vehicle maintenance log
- spare part register

Validation:
- prohibit update if cargo is assigned
- backend delivery validation

Bugfix:

- React Apps:
  - selecting new truck and cargo often causes infinite loop
  - driver list doesn't refresh in Admin Driver App
    on up/down-arrow, but ok on click
  - wrong route points order
- Info:
  - no deliveries in database: webservice fails