

Technical Solution

Logiweb project @ JavaSchool#14

Author: Andrey Baliushin

<https://github.com/AndrewBaliushin/Logiweb>

Table of Contents

Project goals.....	3
Requirements.....	3
Development tools.....	5
Technologies.....	6
Architecture.....	7
Class diagram.....	8
Entity–relationship model.....	9
Sequence diagrams.....	10
Quality report.....	12

Project goals

Develop software that represents information system for some fictional transportation company.

System must be able to manage Trucks, Drivers, Orders and Cargoes.

Requirements

- Through UI for managers:
 - Add, edit and show Trucks and Drivers
 - Add and show Orders ensuring that:
 - All cargoes have destination and origin points
 - Show statuses for cargoes and orders
 - Show list of Trucks that are suitable for order if they are:
 - in unbroken state
 - fit by cargo capacity
 - free from other orders
 - Search for and assign Drivers to Trucks by crew size limit and time, that is required to complete Order (calculated by map with the cities)
 - Drivers monthly working hours limit (176 hours) should not be exceeded.
 - Drivers are not busy by other Orders.
 - Drivers are in same city as Truck.
- Through UI for Drivers:
 - Provide your personal employee id and get info on your assignments:
 - Personal employee id
 - Co-drivers personal ids
 - Truck license plate number
 - Order id
 - Way-points list
- For Drivers through WS/RS interface:
 - Driver started new shift
 - Driver personal employee id

- Driver status (main driver or resting)
- Driver changed status
 - Driver personal employee id
 - Driver status
- Driver finished the shift
 - Driver personal id
- Order status have changed
 - Cargo id
 - Status (Picked up, Delivered)

Development tools

- Eclipse IDE
- Maven
- Tomcat
- WildFly
- Git
- MySQL Workbench
- Astah*
- Cobertura
- Sonar

Technologies

- Java 1.7
- JPA 2.0
- MySQL
- JAX-WS (CXF)
- Spring Framework

Architecture

Project relies on 3-tire architecture (presentation, services, persistence). Project is divided to modules accordingly. There is addition module for web-services.

Project tree:

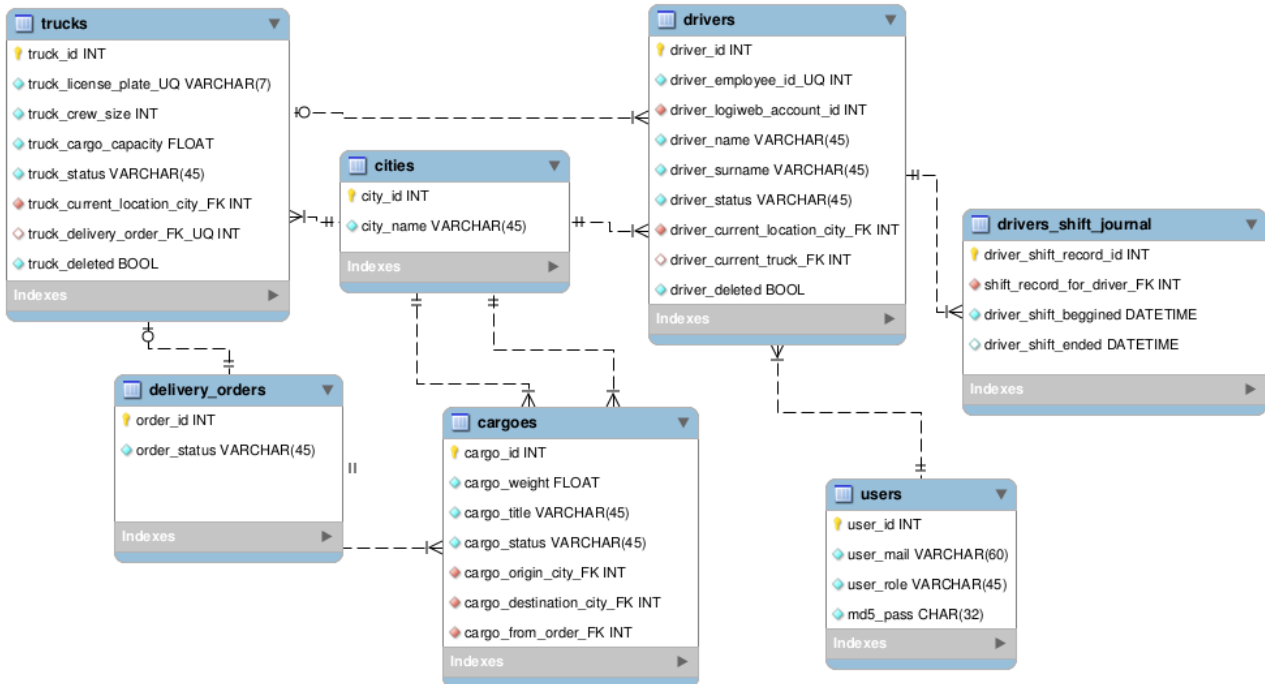
```
|— pom.xml
|— db-entity-relationship-model.png
|— DEPLOYMENT-GUIDE.md
|— README.md
|— common
|   |— pom.xml
|   |— src
|— persistence
|   |— pom.xml
|   |— src
|— presentation
|   |— pom.xml
|   |— src
|— service
|   |— pom.xml
|   |— src
|— webservice
|   |— pom.xml
|   |— src
```

Class diagram

Class diagram is not attached to this document due to it's size, but it can be found here:

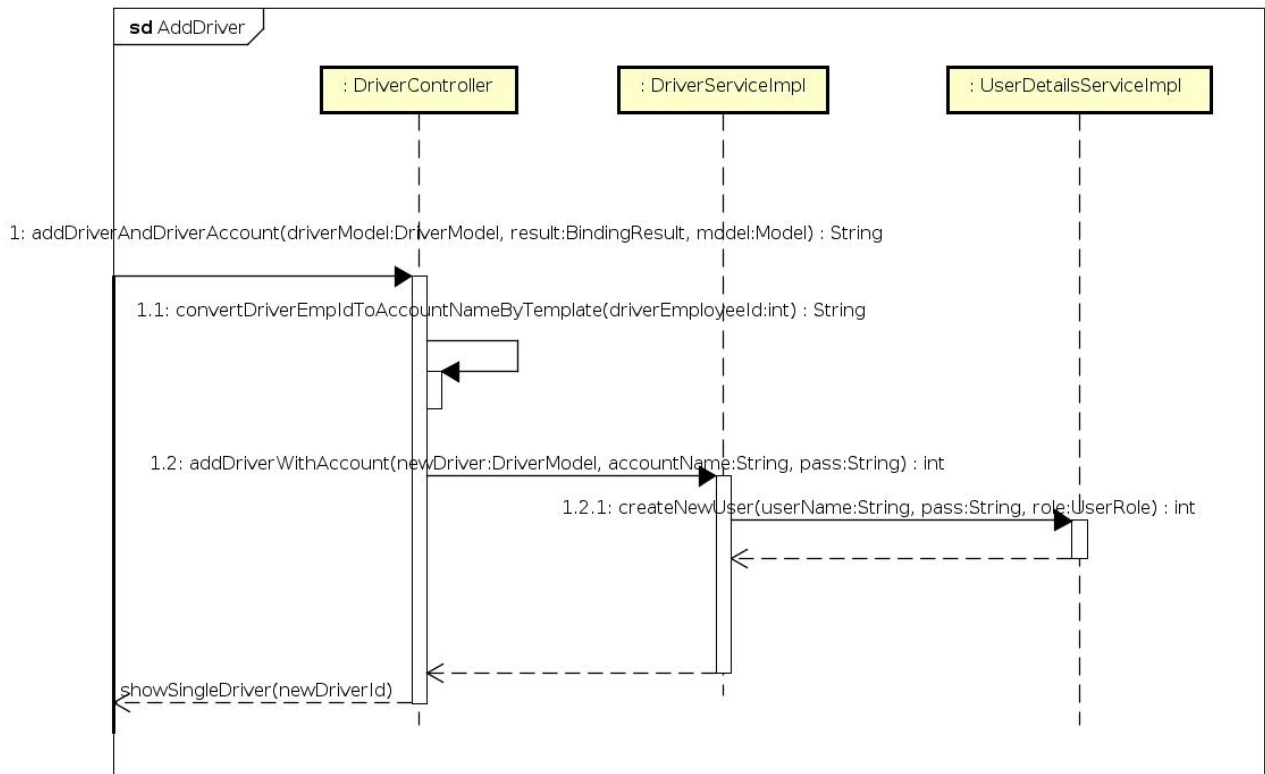
<https://github.com/AndrewBaliushin/Logiweb/blob/master/class-diagram.jpg> (2.2MB)

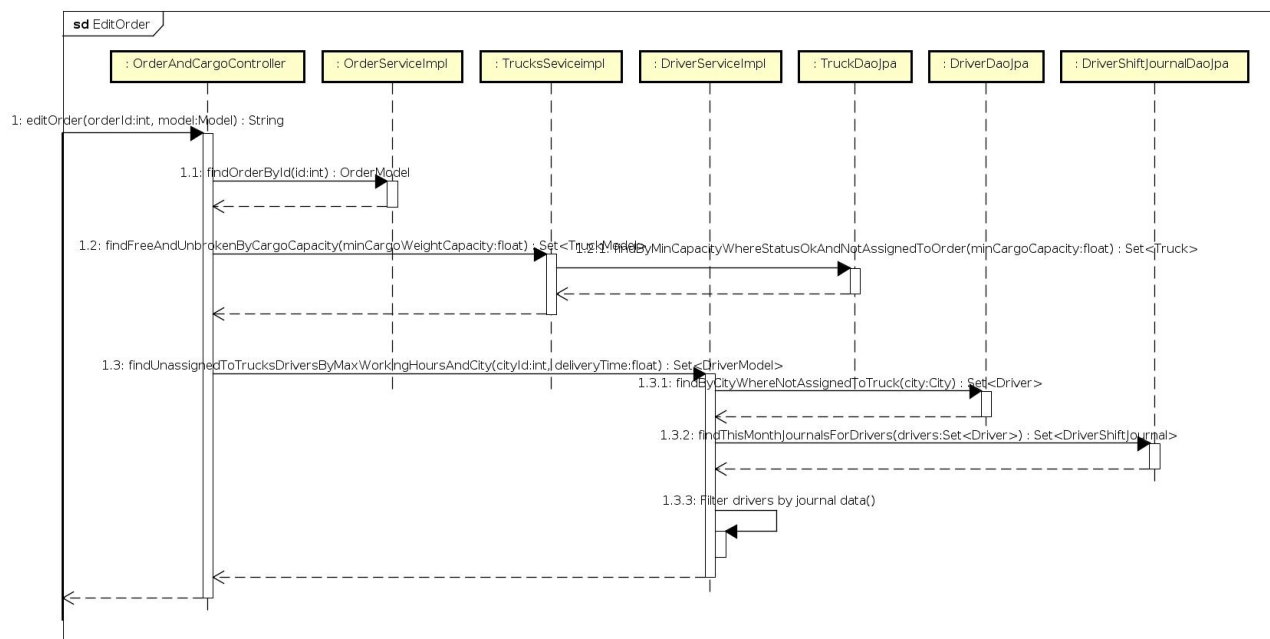
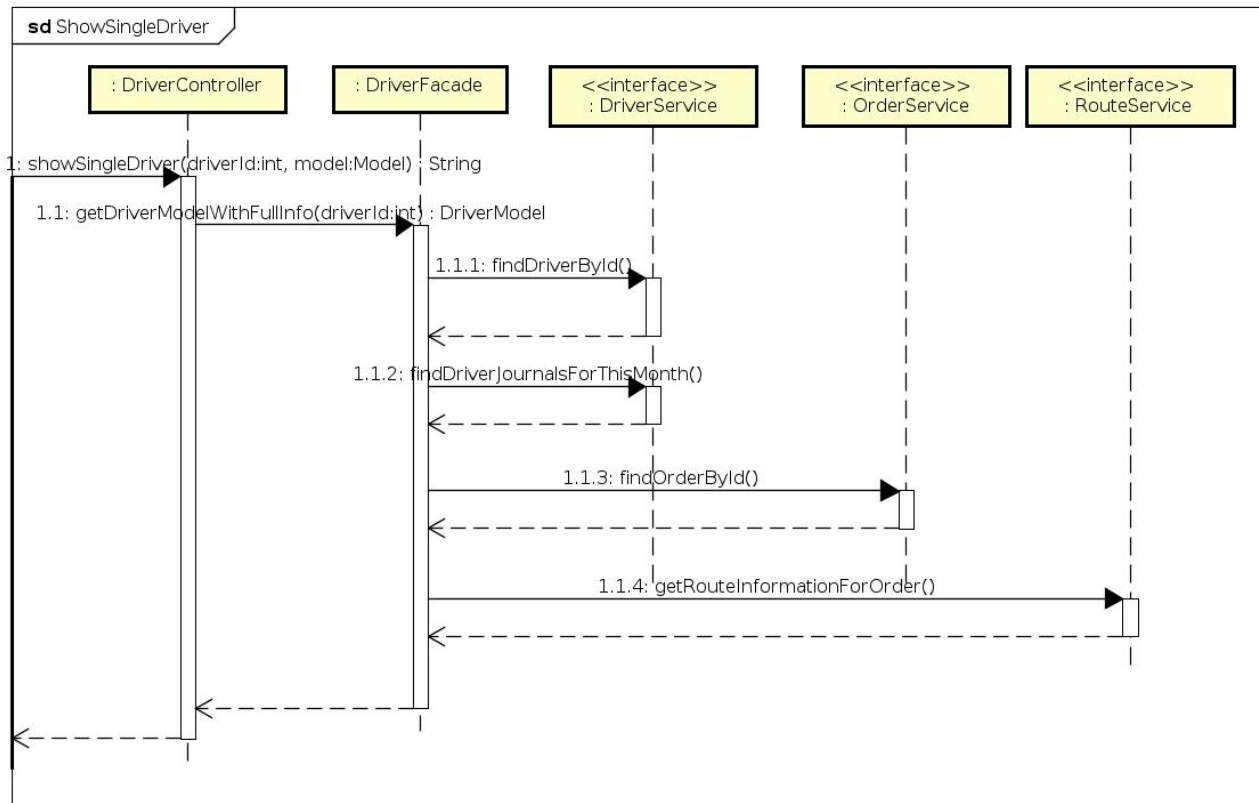
Entity-relationship model



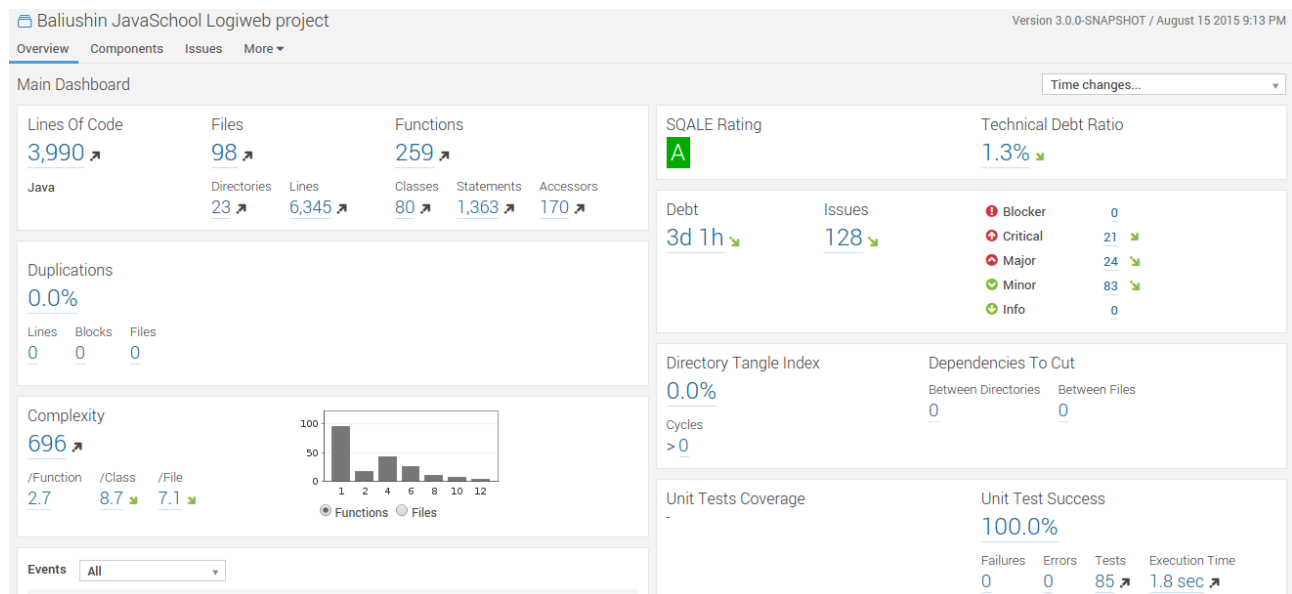
Sequence diagrams

Below are diagrams for non-trivial cases.





Quality report



Service layer test coverage:

Package ↗	# Classes	Line Coverage		Branch Coverage		Complexity
All Packages	33	48%	469/973	58%	151/260	2.474
com.tsystems.javaschool.logiweb.model	6	61%	64/104	N/A	N/A	1
com.tsystems.javaschool.logiweb.model.ext	1	27%	33/118	20%	7/34	2.417
com.tsystems.javaschool.logiweb.service	7	N/A	N/A	N/A	N/A	1
com.tsystems.javaschool.logiweb.service.aspects	1	0%	0/18	0%	0/6	3
com.tsystems.javaschool.logiweb.service.exceptions	4	17%	6/34	N/A	N/A	1
com.tsystems.javaschool.logiweb.service.ext	3	0%	0/33	N/A	N/A	1
com.tsystems.javaschool.logiweb.service.facades	1	0%	0/18	0%	0/4	3
com.tsystems.javaschool.logiweb.service.impl	8	56%	360/642	66%	140/212	5.721
com.tsystems.javaschool.logiweb.service.validators	2	100%	6/6	100%	4/4	3

Classes in this Package ↗	Line Coverage		Branch Coverage		Complexity
CargoServiceImpl	46%	40/86	44%	15/34	6.75
CityServiceImpl	0%	0/12	N/A	N/A	3
DriverServiceImpl	75%	165/218	87%	54/62	5.05
OrderServiceImpl	65%	64/97	85%	34/40	7.75
RouteServiceStub	0%	0/49	0%	0/16	3.5
RouteServiceStub\$1	0%	0/2	N/A	N/A	3.5
TrucksSeviceImpl	71%	91/128	77%	37/48	7.4
UserDetailsServiceImpl	0%	0/50	0%	0/12	4.667