# Technical solution description

T-SYSTEMS, JAVASCHOOL, LOGIWEB PROJECT

KUZMENKOV NIKOLAY

## Table of contents

**GitHub**

https://github.com/nkuzm/lw-ee - Main application.

https://github.com/nkuzm/lw-client-ee – client for WS interface of main application

## Introduction

The task is to write an application that simulates the work of the information system of a company engaged in the carriage of freights. Below is a more detailed subject area and technical requirements. System must be able to manage Drivers, Trucks, Orders and Freights.

The following types of entities:

- Truck
    - Reg. number (2 letters +5 digits)
    - Size driver shift
    - Capacity
    - Status (work, faulty)
    - Current city

- Driver
    - First name
    - Last name
    - Personal number
    - Hours of operation this month
    - Status (free, rest in shift, driving)
    - Current truck
    - Current city

- Order
    - Number
    - Completed (yes / no)
    - List of waypoints (status, city from/to, name of freight)
    - Truck appointed to carry out the order
    - The list of drivers who carry out the order

- Freight
    - Number
    - Name
    - Weight
    - Status (waiting for pick up, picked up, delivered)

- City (location for drivers and trucks, destination for freight waypoints)

## Project goals and requirements

- Through UI for managers:
    - add, delete, edit and show Drivers and Trucks;
    - add and show Orders ensuring that: all freight have origin and destination points;
    - show statuses for Orders and Freights;
    - show list of Trucks that are suitable for if they are: in unbroken state, fit by freight capacity, free from other orders;
    - search for and assign Drivers to Trucks by driver shift 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 driver number and get info on your assignments: Personal driver number, co-drivers, truck number, order ID, waypoints list.

- For drivers through WS or RS interface:
    - driver started new shift: driver personal number, driver status (main driver or resting);
    - driver changed status: driver personal number, driver status;
    - driver finished the shift: driver personal number;
    - order status changed: freight ID, freight status (picked up, delivered).

As a result, required a multi-user application with the client for the company's employees, the remote server (network connection) and open (on the server) WS/RS systems, interfaces to third-party driver systems. All data stored on the server side.

Each client may download some data, after each change, the data must be synchronized with the server. The client must have a graphical user interface (console interface is allowed but not recommended). For WS or RS additional client interface is required.

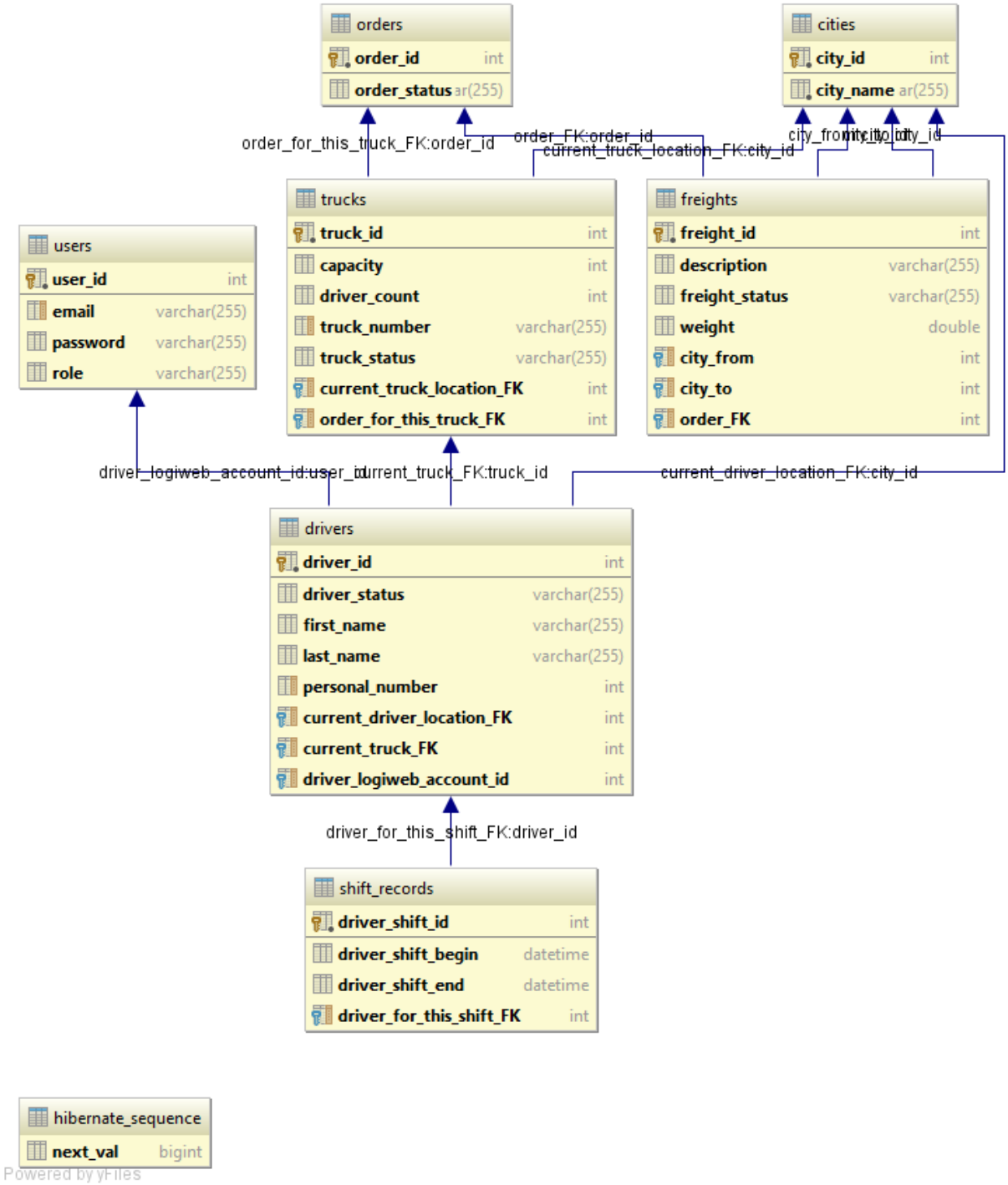
**Development tools**

- IntelliJ IDEA – IDE
- Maven
- Tomcat
- Wildfly
- Git
- Sonar
- Cobertura

**Technologies**

- Java 1.8
- MySQL
- JPA
- Spring Framework
- EJB
- JSF
- JAX-WS
- Bean Validation
- Apache CXF
- PicketLink
- JUnit
- Log4j
- Bootstrap framework
- JQuery

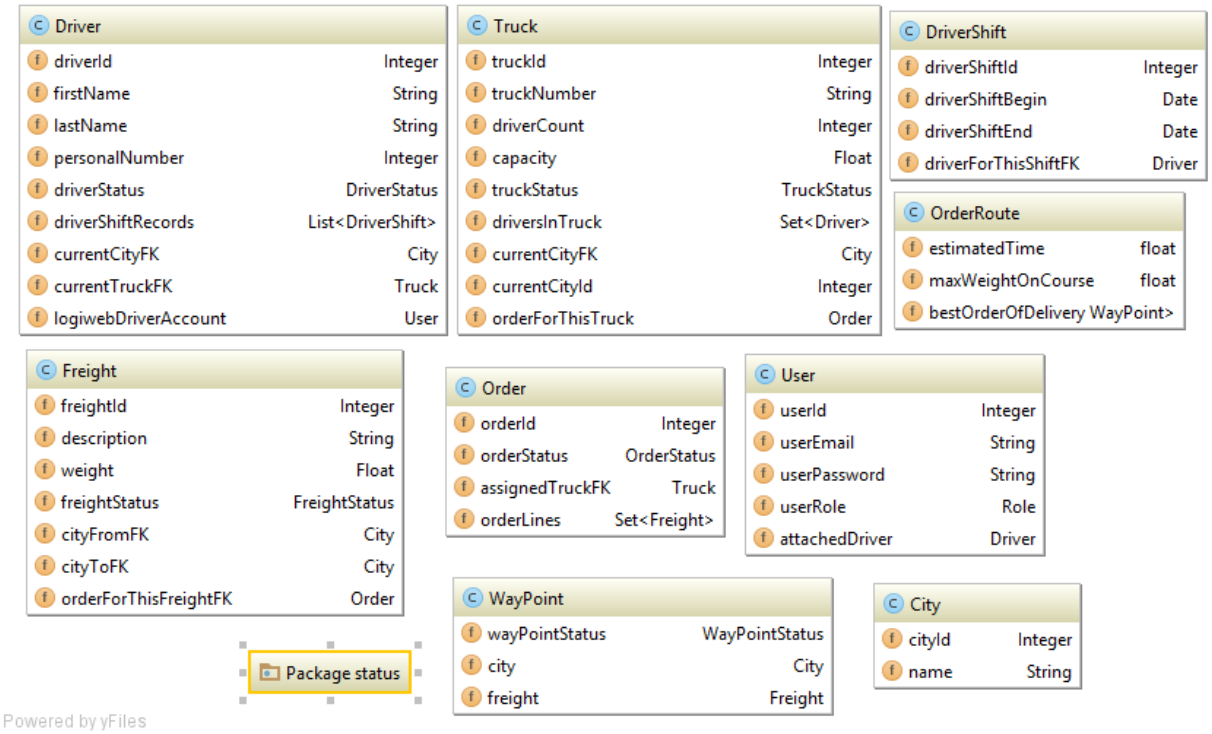
## Database model



## System architecture

Entities:

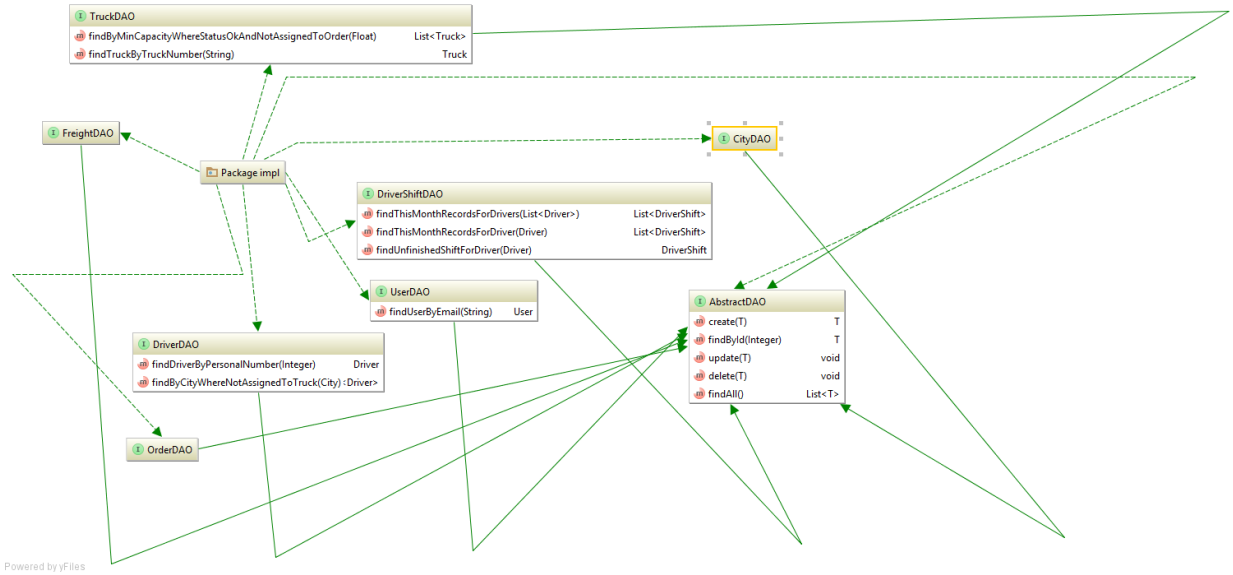
This Application consists of seven entities with the corresponding mapping to tables in the database and three supporting classes in package “entities”.



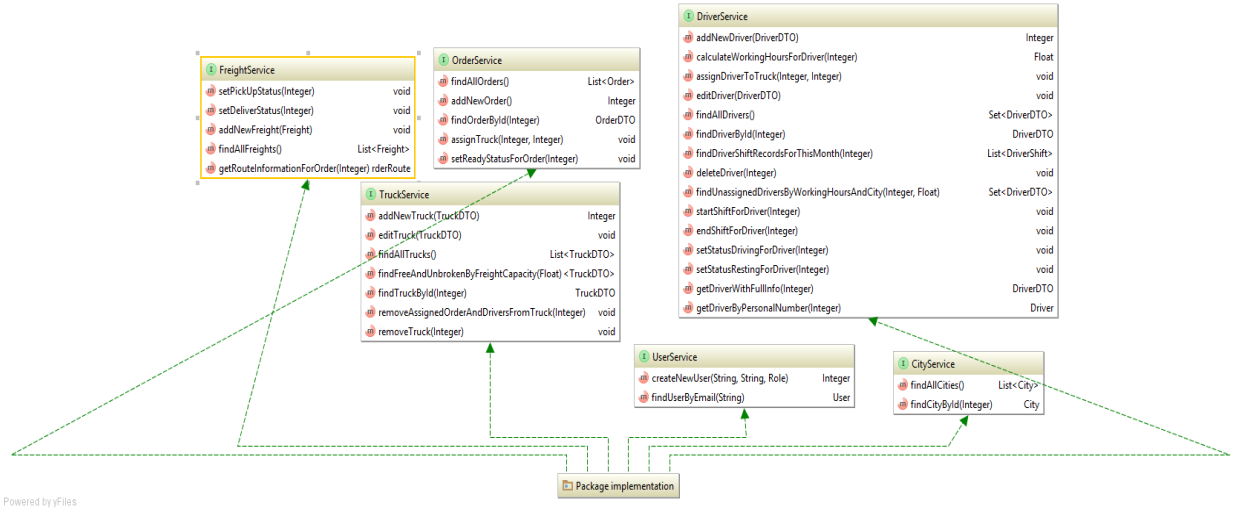

The application has 5 layers defining its architecture:

- Entities;
- DAO classes;
- Service classes including business logic of project;
- Controllers process requests and parameters from UI and dispatch queries;
- View layer consists of JSP-pages for the first application and JSF-pages for the second application, js-scripts, css styles, elements of Bootstrap framework.

# DAO:




# Services:



# Quality report