Final Task

# Information system of the logistics company (transportation of goods).

There are the following types of entities:

* Truck
* Number (4 digits (from 0000 to 9999) + 3 letters)
* Capacity (tons)
* Status (OK, NOK)
* Current city
* Driver
  + Name
  + Surname
  + Personal number
  + Working hours in the current month
  + Status (rest, driving)
  + Current city
  + Current truck
* Order
  + Unique number
  + Completed (yes/no)
  + Waypoint list
    - City
    - Cargo
    - Type (loading/unloading)
  + Truck to complete order
  + List of the drivers to complete the order
* Cargo
  + Unique number
  + Name
  + Weight (Kg)
  + Status (ready, shipped, delivered)
* Country map
  + Cities
  + Distances

The application must provide the following functionality:

* For company employees (via UI):
  + viewing the list, adding, editing and deleting trucks, drivers;
  + reviewing the list and adding new orders, checking that:
    - all loaded cargos must be unloaded somewhere;
    - all unloaded goods must be loaded somewhere;
  + view the status of orders and cargos;
  + displaying a list of trucks that are suitable for order delivery if:
    - the truck is in good condition;
    - the truck is suitable in terms of capacity (taking into account the loading / unloading of goods in the cities along the route);
    - the truck is not fulfilling any orders at the moment;
  + selection and assignment of drivers based on the size of the work shift used by the truck and the estimated travel time (calculated from the map of cities and waypoints):
    - the time limit per month (176 hours) for each of the drivers in the shift will not be exceeded during the execution of this order (also take into account the change of months during the order);
    - the driver is not currently fulfilling other orders;
    - at the time of appointment, the driver is located in the same city as the truck.
* For drivers (via UI):
  + display the following information:
    - driver's personal number
    - personal numbers of co-driver(s)
    - truck number
    - order number
    - list of waypoints
  + change the actual working time and order status:
    - the driver entered/finished the shift
    - driver changed status:
      * Behind the wheel
      * Second driver
      * Loading and unloading work
      * Rest
    - the driver received/unloaded the cargo (change order status)
      * Uploaded
      * Unloaded

# Technical requirements

As a result, it is required to obtain a multi-user client-server application with a network connection.

All data is stored on the server side. Each client can upload some data, after each change operation the data must be synchronized with the server.

The client must have a graphical interface.

The application must handle hardware and software errors.

# Criteria for successful completion of the task

1. Functionality works (UI required)

2. Maven-based project divided into modules (build with one command, deploy with one command)

3. Domain interfaces are described

4. Connected MySQL database

5. Entities of the subject area are created; mapping to tables in the database

6. Working with entities through DAO

7. Application deployed on AS

8. Implemented exception handling

9. Logging enabled

10. Availability of technical solution description

11. Availability of unit tests for business logic

# Technologies for project implementation

Within the framework of the school, students are not limited in the choice of technologies for project implementation within the technology stack used in the company in various projects (lectures cover a small part of them). The list of technologies to choose from is given below. However, the project must be written primarily in java (minor parts, separate modules, or additional functionality can be written in other languages, such as Kotlin or Python). After discussion with the mentor and in agreement with him, the student can choose any combination of technologies and frameworks, but you should know that the mentor may be unfamiliar with some of them.

List of technologies used in some of the company's projects:

Spring Core,Spring Data,AOP,Spring Boot,Spring Security,MVC,Spring Cloud, Microservices, Data processing(Spark, Apache Flume), CDI/EJB, Apache CXF, RedHat Fuse, Oracle eCommerce (ATG, Endeca...), JBoss, Apache Tomcat, Embedded Tomcat H2 (in-memory), Websphere Application Server 9.0, React, Angular/NPM/Webpack, typescript, JSF/ExtJs, JSP/Servlets, JavaScript / TypeScript / HTML / Sass / CSS/LESS/jQuery/GraphQL, SOAP/REST, JDBC, XML/XSLT/XSD,Bash scripting, Oracle Database, Gradle, Git, Maven, Subversion, Jenkins, Grafana, GitlabCI, Sonar, UNIX shell, Bootstrap, Formbased authorisation/JWT, Drools, ARS, DOM, PL SQL, Oracle AQ, IBM MQ, JMS, OracleDB ,Apache Camel, Docker, Microsoft T-Sql, WebDriver (Java), Oracle ATG, python, mapstruct, Lombok, Groovy, openshift, EC2, S3, Mongo, Casandra, hazelcast ,Junit, testing, Selenium, Mockito, Selenid, Prometheus, Kubernetes , Helm, Kibana, AWS, Apache Camel, Go, K8s, ETCD, PostgreSql, Wicket, blockchain, Web Flux, JPA/Hibernate

It will be a plus to use the following technologies: Selenium, Sonar, Angular / React, Docker, Microservices, using an available cloud to deploy an application and/or the presence of "killer features".