Итоговое задание

В задании требуется написать приложение, моделирующее работу информационной системы некоторой компании, осуществляющей перевозку грузов. Ниже более подробно описаны предметная область и технические требования.

# Предметная область

Существуют следующие виды сущностей:

* Фура
  + Рег. номер (2 лат.буквы + 5 цифр)
  + Размер смены водителей
  + Вместимость (тонн)
  + Состояние (исправен, неисправен)
  + Текущий город
* Водитель
  + Имя
  + Фамилия
  + Личный номер
  + Отработано часов в этом месяце
  + Статус (отдых, в смене, за рулем)
  + Текущий город
  + Текущая фура
* Заказ
  + Уникальный номер
  + Выполнен (да/нет)
  + Список маршрутных точек
    - Город
    - Груз
    - Тип (погрузка/выгрузка)
  + Фура назначенная выполнять заказ
  + Список водителей, которые выполняют заказ
* Груз
  + Номер груза
  + Наименование
  + Масса (в кг)
  + Статус (подготовлен, отгружен, доставлен)
* Карта страны
  + Города
  + Расстояния

Приложение должно обеспечивать следующую функциональность:

* Для сотрудников компании (через UI-интерфейс):
  + просмотр списка, добавление, редактирование и удаление фур, водителей;
  + просмотр списка и добавление новых заказов с проверкой, что:
    - все загружаемые грузы должны быть где-то выгружены;
    - все выгружаемые грузы должны быть где-то загружены;
  + просмотр состояния заказов и грузов;
  + вывод списка фур, которые подходят для доставки заказа, если:
    - фура находится в исправном состоянии;
    - фура подходит по вместимости (с учетом погрузки/выгрузки грузов в городах по маршруту следования);
    - фура не выполняет в данный момент никаких заказов;
  + подбор и назначение водителей на основании размера смены используемой фуры и примерного времени в пути (рассчитывается по карте городов и путевым точкам):
    - лимит времени за месяц (176 часов) для каждого из водителей в смене не будет превышен в ходе выполнения этого заказа (учесть также смену месяцев в ходе заказа);
    - водителем не выполняет сейчас другие заказы;
    - при назначении водитель находится в том же городе, что и фура.
* Для водителей (через UI-интерфейс):
  + получить данных о своих назначениях, указав свой личный номер:
    - личный номер водителя
    - личные номера ко-водителей
    - рег. номер фуры
    - номер заказа
    - список маршрутных точек
* Для водителей (через ws/rs-интерфейс), фиксируется фактическое время работы:
  + водитель заступил на смену:
    - личный номер водителя;
    - статус (основной водитель, отдых);
  + водитель изменил статус:
    - личный номер водителя;
    - статус (основной водитель, отдых);
  + водитель окончил смену;
    - личный номер водителя;
  + статус заказа изменился:
    - номер груза;
    - статус (отгружен, доставлен).

# Технические требования

В итоге требуется получить многопользовательское приложение с клиентом для сотрудников компании, удаленным сервером (соединение по сети) и открытыми (на сервере) WS/RS интерфейсами для сторонних систем водителя.

Все данные хранятся на стороне сервера. Каждый клиент может загружать некоторые данные, после каждой операции изменения данные должны быть синхронизованы с сервером.

Клиент должен иметь графический интерфейс (консольный интерфейс допускается, но не рекомендуется).

Для WS/RS интерфейсов дополнительный клиент не требуется. Должно быть возможно вызвать сервисы стандартными средствами типа SoapUI или RestConsole.

Приложение должно обрабатывать аппаратные и программные ошибки.

Используемые технологии:

1 часть

* IDE - Any (Eclipse, IDEA)
* Tomcat 8.0.x
* Servlet/JSP
* DB – MySQL 5.6
* Maven 3.\*
* JPA 2.0

2 часть (включая предыдущие)

* AS – Glassfish 4.0/WildFly 8.x
* JAX-WS/RS
* EJB 3.x
* JSF 2.x

или

* Spring Framework

# Критерии успешного выполнения

1. Функциональность работает (обязательно наличие UI и сервисов)

2. Maven-based проект, разбитый на модули (билд одной командой, деплой одной командой)

3. Описаны интерфейсы предметной области

4. Подключена БД MySQL

5. Созданы сущности предметной области; маппинг на таблицы в БД

6. Работа с сущностями через DAO

7. Приложение развернуто на AS Glassfish/Wildfly\* (актуально для 2й части)

8. Реализована обработка исключений

9. Подключено логгирование

10. Наличие technical solution description

11. Наличие unit-тестов на бизнес логику

Подключение любого фреймворка/библиотеки принимаются в зачет только при условии выполнения пунктов, описанных выше.

При наличии дополнительного времени будут плюсом моковые клиенты для вызова сервисов со стороны водителя.

# Дополнительные требования

!Относится ко второй части обучения!

Реализовать клиент для WS/RS интерфейсов водителя.

1. Приложение должно связываться с основным приложением через веб-сервис.
2. Технологии приложения – EJB (или CDI), JSF.