**Тестовое задание при приеме на работу**

**Должность**: backend инженер-программист.

**Цель тестового задания**: подтверждение квалификации кандидата, в том числе на удаленную работу.

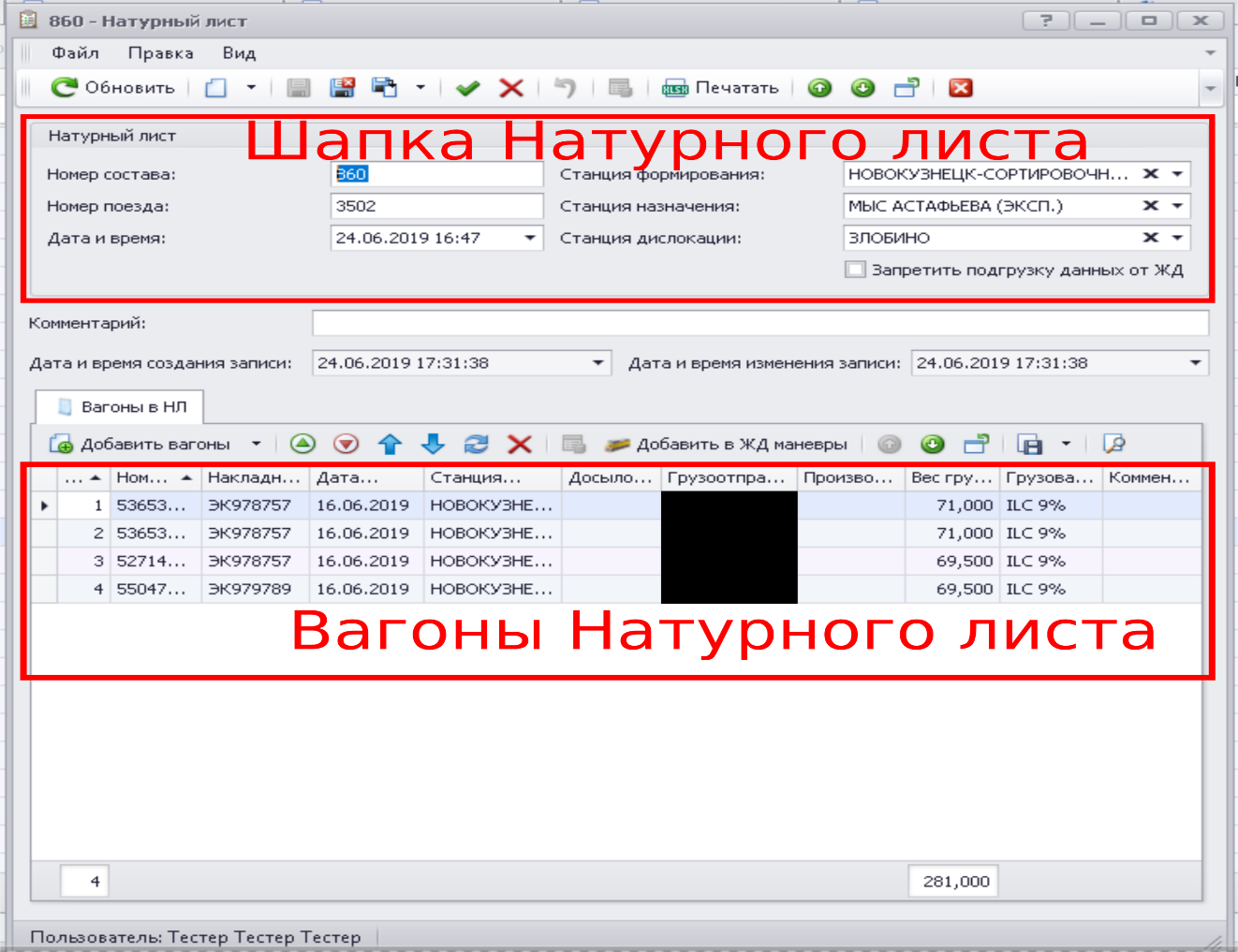
**Средства реализации:**

* **язык программирования:** С#;
* **инструменты реализации:**
  + **.NET (на выбор кандидата):**
    - **.NET Framework 4.8:**
      * EPPlus (<https://github.com/JanKallman/EPPlus>);
      * Console Application (<https://docs.microsoft.com/en-us/dotnet/standard/building-console-apps>)
    - **.NET Core 3.1-6.0:**
      * EPPlus (<https://github.com/JanKallman/EPPlus>);
      * Console Application (<https://docs.microsoft.com/ru-ru/dotnet/core/tutorials/using-with-xplat-cli>);
  + **СУБД (на выбор кандидата):**
    - MS SQL Server Express (не ниже версии 2015);
    - PostgreSQL (не ниже версии 10);
  + **ORM (на выбор кандидата)**:
    - EntityFramework (<https://docs.microsoft.com/ru-ru/ef/>);
    - NHibernate (<https://github.com/nhibernate/nhibernate-core> );
    - XPO (<https://www.devexpress.com/products/net/orm/>);
    - если с ORM совсем тяжко, допускается использовать Dapper (<https://github.com/DapperLib/Dapper>);
  + **WebApi** (<https://docs.microsoft.com/ru-ru/aspnet/core/tutorials/first-web-api?view=aspnetcore-3.1&tabs=visual-studio>);
  + **IOC DI (на выбор кандидата, необязательно использовать)**:
    - MS (<https://docs.microsoft.com/en-us/aspnet/core/fundamentals/dependency-injection?view=aspnetcore-5.0> );
    - Autofac (<https://autofac.org/>) .

**Описание предметной области:**

В качестве объекта предметной области возьмем документ «Натурный лист поезда» в морском торговом порту.

Натурный лист поезда рис 1:



«Натурный лист поезда» состоит из «шапки» и списка вагонов:

* + в «шапке» располагаются данные:
    - номер поезда;
    - номер состава;
    - дата и время формирования состава либо операции с составом;
    - станция назначения (конечной станции маршрута);
    - станция отправления;
    - станция дислокации (текущего местонахождения вагона);
  + в списке вагонов натурного листа содержаться информация:
    - позиция вагона в составе;
    - номер вагона;
    - номер накладной;
    - дата отправления;
    - наименование груза;
    - вес по документам (см. рис. 1).

Информация из натурного листа помогает бизнесу определиться с дислокацией(местоположением) вагонов с грузом на сети дорог РЖД, порядке следования вагонов в составе и принять соответствующие действия:

* + подготовить склады к приему груза;
  + удостовериться что груз идет в полном заявленном объеме.

**Задание:**

1. Проанализировать данные из Приложение 1(Data.xml), исходя из описания предметной области;
2. создать таблицы\бизнес-объекты в 3-й нормальной форме;
3. загрузить данные из Приложение 1(Data.xml) в таблицы;
4. реализовать службу WebApi со следующими функционалом:
   * 1. приём файлов с аналогичной структурой в Data.xml, с последующей обработкой и записью данных в БД;
     2. выдача файлов отчета с фильтром по полю «**Номер поезда**», сформировать отчет по данным одного натурного листа в формате Excel, базируясь на шаблоне Приложение 2(NL\_Template.xlsx);
     3. выдача структуры натурного листа в формате **Json** с фильтром по полю «**Номер поезда**»;
5. дополнительный функционал, необязательный к исполнению, но являющийся конкурентным преимуществом:
   1. использовать DI (Dependency Injection) c IoC-контейнером (Inversion of Control) на выбор кандидата;
   2. на службе WebApi реализовать авторизацию, на выбор кандидата Basic, JWT и т.д.

При выполнении задания, необходимо учитывать следующие условия:

* **с примером можно ознакомится в Приложении 3(NL\_Пример.xlsx);**
* **номер состава закодирован в индексе поезда (86560-925-98570) — 925 является номером состава;**
* **данные списка вагонов натурного листа выводятся в порядке возрастания «Позиции вагона в составе»;**
* **в конце вывода списка вагонов натурного листа производятся расчеты:**
  + **кол-во «вагонов» и «веса по документам» по грузам;**
  + **общее кол-во «вагонов», «груза» (типов груза) и «веса по документам».**

**Приложение 1. Описание полей Data.xml:**

<?xml version="1.0" encoding="utf-8"?>

<Root>

<row>

**<--Номер поезда -->**

<TrainNumber></TrainNumber>

**<--Индекс поезда -->**

<TrainIndexCombined></TrainIndexCombined>

**<--Наименование станции отправления -->**

<FromStationName></FromStationName>

**<--Наименование станции назначения -->**

<ToStationName></ToStationName>

**<--Наименование станции дислокации (текущего местонахождения) -->**

<LastStationName></LastStationName>

**<--Дата и время операции над составом -->**

<WhenLastOperation></WhenLastOperation>

**<--Наименование операции -->**

<LastOperationName></LastOperationName>

**<-- Номер накладной -->**

<InvoiceNum></InvoiceNum>

**<-- Позиция вагона в составе -->**

<PositionInTrain></PositionInTrain>

**<-- Номер вагона -->**

<CarNumber></CarNumber>

**<-- Наименование груза -->**

<FreightEtsngName></FreightEtsngName>

**<-- Общий вес вагона с грузом -->**

<FreightTotalWeightKg></FreightTotalWeightKg>

</row>

{0…..N}

</Root>

**Примечание:**

- Неописанные явным образом условия остаются на усмотрение разработчика;

**Демонстрация:**

Выполненную работу необходимо продемонстрировать в электронном виде и предоставить исходный код проекта (можно загрузить на GitHub).

Результатом выполнения задания является:

* функционирующее приложение, соответствующее пункту «**Задание**» и всем его подпунктам;
* пояснительная записка с кратким обоснование выбранного инструментария.

В определенных случаях тестовое задание может рассматриваться как, предмет для обсуждения на собеседовании, а также возвращено на доработку, если:

* кандидат не разобрался с предметной областью;
* выполнено менее 90% подпунктов пункта «Задание»;
* имеют неточности относительно предметной области.