**Тестовое задание**

Предметная область:

Сущность *договор:*

* название
* тип договора (одно из 3 значений: «Закупка», «Поставка», «Работы»)
* плановые сроки начала и окончания
* фактические сроки начала и окончания
* сумма договора
* может включать один или более этапов
* может включать один или более договоров с контрагентами

Сущность *этап договора:*

* название
* плановые сроки начала и окончания
* фактические сроки начала и окончания
* сумма этапа
* расходы, которые входят в этап (два типа: план и факт) – расходы на материалы, расходы на зарплату

Сущность *договор с контрагентами:*

* название
* тип договора (одно из 3 значений: «Закупка», «Поставка», «Работы»)
* организация-контрагент
* сумма договора
* плановые сроки начала и окончания
* фактические сроки начала и окончания

Сущность *организация-контрагент:*

* название
* адрес
* ИНН

Сущность *пользователь:*

* ФИО
* логин
* пароль
* дата прекращения действия

Технологии и ограничения:

* Клиент-серверное приложение
* ОС сервера: Ubuntu
* Клиент: Vue.js (версии 2), Vuetify, TypeScript
* Сервер: Java 1.8, Spring, Hibernate, Apache POI, Spring Data JPA
* База данных: MySql Community Edition
* Браузер Chrome

Постановка задачи:

Необходимо реализовать учетную систему для указанной предметной области, с возможностью формирования отчетных форм в MS Excel

Основные требования:

1. Спроектировать схему БД на основе описания предметной области
2. Реализовать программу
   1. Меню пунктами:

* Договоры (таблица договоров с возможностью фильтрации по полям сущности)
* Организации-контрагенты (таблица организация с возможностью фильтрации по полям сущности)
* Отчеты
* Администрирование
  1. Карточка договора. Открывается на редактирование при выборе договора или добавлении нового из таблицы договоров. При этом:
* должны быть отражены все поля сущности *договор*, доступны на редактирование должна быть вложенная таблица с перечнем этапов (доступных на редактирование аналогично, через карточку этапа)
* должна быть вложенная таблица с перечнем договоров с контрагентами (доступных на редактирование аналогично, через карточку договора с контрагентами)
  1. Справочник Организаций-контрагентов (таблица-перечень, редактирование через карточку).
  2. Отчеты. Должно быть реализовано 2 отчета в Ms Excel:
* вывод всех договоров за задаваемый плановый период – должны фильтроваться по плановым срокам. Вывод в табличном виде, выводить только поля самого договора. Напротив каждого договора указывать тип: основной/ или это договор с контрагентами. Для договоров с контрагентами предусмотреть вывод того основного договора, в рамках которого он заключен.
* Вывод всех этапов для выбранного пользователем договора. Выводить в табличном виде (отобразить все поля сущности *этап договора*)
  1. Администрирование. Форма отображения, редактирования пользователей.

Технические требования:

1. Авторизация по логину/паролю (хранение данных в сущности пользователь) и далее работа на основе SessionID
2. Взаимодействие с БД с помощью Hibernate
3. Поля в карточках должны валидироваться в соответствии с форматами данных, установленными в БД.
4. Все добавленные записи должны иметь возможность удаления. При этом не должна нарушаться целостность зависимых данных. При удалении зависимых данных – выдавать запрос подтверждения их удаления (или сообщение о невозможности удаления из-за связанных записей)
5. Отчеты должны выгружаться в формат MS Excel с помощью Apache POI
6. Исходные коды должны быть в Git. Желательно делать частые информативные коммиты, активно использовать ветки.
7. Должен быть описан порядок развертывания и запуска проекта в среде разработки (включая настройку БД, дамп схемы БД, настройку сервера приложения для запуска и т.д.)
8. Сервисы (серверная часть) должны быть максимально покрыты модульными тестами

Дополнительные требования (желательно сделать, будет хорошим плюсом к демонстрации знаний):

1. Ведение истории входов пользователей в отдельной NoSQL БД (например, MongoDB)
2. Авторизация через JWT токены
3. Реализовать разграничение доступа: к администрированию только у супер-пользователя. Также на каждый договор назначается ответственный пользователь, который после авторизации видит только свои договоры
4. Реализация навигации в клиенте через Breadcrumbs
5. Покрытие тестами клиентской части (интерфейс)
6. Результат работ разворачивается в Docker
7. Использовать шаблоны \*.xlt для выходных форм для того, чтобы менять названия столбцов в отчетах, не изменяя код программы
8. Сделать UML диаграммы: классов, последовательности и т.д. (любые диаграммы, которые могут продемонстрировать знание UML)