Итоговое задание

В задании требуется написать приложение, моделирующее работу информационной системы оператора сотовой связи. Ниже более подробно описаны предметная область и технические требования.

# Предметная область

Существуют следующие виды сущностей:

* Тариф
  + Название
  + Цена
  + Список возможных опций
* Опция
  + Название
  + Цена
  + ~~Стоимость подключения~~
* Клиент
  + Имя
  + Фамилия
  + Дата рождения
  + ~~Паспортные данные~~
  + Адрес
  + ~~Номера контрактов(телефонов у данного оператора)~~
  + Адрес электронной почты
  + Пароль для личного кабинета
* Контракт
  + Номер
  + Тариф
  + Выбранные опции для тарифа
  + Новое: Ссылка на клиента

Приложение должно обеспечивать следующую функциональность:

* Для клиентов компании
  + Просмотр контракта в личном кабинете;
  + Просмотр всех возможных тарифов для перехода
  + Возможность инициировать изменение тарифа

Es gibt die Tarife S, M, L.

Ein Upgrade kann man in jeden „höheren“ Tarif machen. (S->M; S->L; M->L).

Ein Downgrade ist nur in einen „nächst-niedrigeren“ Tarif möglich (L->M; M->S), L->S ist nicht erlaubt.

* + Просмотр возможных опций для тарифа, подключение новых опций, отключение существующих;

Варианты бронирования можно свободно продумать (например, MultiSIM, Hotspot Flat, Spotify и т. д.).

* + При редактировании договоров на странице, т.е. перед сохранением изменений, должна отображаться корзина с выбранными покупателем товарами.
  + Товары в бэкенде передаются только тогда, когда корзина «выписана», то есть «заказ платный».

# Технические требования

Требуется получить многопользовательское приложение типа клиент-сервер с соединением по сети.

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.

Приложение должно обрабатывать аппаратные и программные ошибки.

Используемые технологии:

* IDE: IntelliJ, Eclipse/STS, VSCode
* Server mit RESTful Web Services
  + Build Tool Maven
  + Spring Framework (Spring Boot, Spring Data JPA)
  + Database Backend – SQL
* Client
  + Browser based (Chrome, Firefox, Edge)
  + Build Tool npm
  + Single Page Application auf Basis von React oder Angular

1. Запрошенный функционал реализован, для него есть пользовательский интерфейс

2. Проект на основе Maven разделен на модули (отдельная команда для сборки и развертывания)

3. Есть техническое описание интерфейса.

4. База данных SQL подключена.

5. Реализованы технические объекты, есть сопоставление с соответствующими таблицами базы данных.

6. Приложение развертывается как приложение Spring Boot в образе Docker.

7. Реализовано логирование.

8. Имеется техническое описание решения.

9. Доступны модульные тесты для бизнес-логики.

# II часть

* Kunde
  + Name
  + Vorname
  + Adresse
  + E-Mail-Adresse
  + Tarifname
* Campaign
  + Id
  + Schedule Date
  + Kunden
  + Inhalt

После входа в систему менеджер по продукту должен иметь возможность создать маркетинговую кампанию для выбранных клиентов.

Клиенты должны иметь возможность быть отобранными для кампании на основе тарифа или места жительства.

В дополнение к списку выбранных клиентов каждая кампания имеет дату начала и текстовое поле, описывающее содержание кампании.

Объекты должны храниться в отдельной базе данных.

Маркетинговое приложение должно работать независимо от первого приложения.

Изменения данных о клиентах или тарифных данных в первом приложении должны быть синхронизированы с этим маркетинговым приложением.