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

Приложение должно обеспечивать следующую функциональность:

* Для клиентов компании
  + Просмотр контракта в личном кабинете;
  + Просмотр всех возможных тарифов для перехода, смена тарифа;
  + Просмотр возможных опций для тарифа, подключение новых опций, отключение существующих;
  + Блокировка/разблокировка номера(если номер заблокирован, то нельзя сменить тариф и опции; если номер заблокировал не пользователь, то разблокирповать сам он его не может);
* Для сотрудников компании
  + Заключение контракта с новым клиентом: выбор нового номера телефона с подключением тарифа и опций. Номер телефона должен быть уникальным.
  + Просмотр всех пользователей и контрактов;
  + Блокировка пользователя/разблокировка пользователя;
  + Поиск пользователя по номеру;
  + Смена тарифа, подключение и отключение опций пользователю;
  + Добавление новых тарифов, удаление старых;
  + Добавление/удаление опций для тарифа;
  + Управление опциями: некоторые опции могут быть несовместимы друг с другом или могут быть добавлены только с определенными опциями, сотрудник добавление и удаление этих правил.

При совершении действий с контрактами на каждой странице до сохранения изменений должна отображаться корзина, в которой отображаются выбранные позиции пользователя.

# 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 данных должны быть синхронизованы с сервером.

Клиент должен иметь графический интерфейс.

Приложение должно обрабатывать аппаратные и программные ошибки.

# II часть

Реализовать отдельное клиент-приложение для рекламного стенда. Приложение должно отображать список возможных тарифов и доступных опций с их стоимостями (или специальные акции - Hot tariff). Данные должны подгружаться при старте и храниться на стороне клиента. Перезагрузка данных осуществляется в случае получения уведомления от сервера об изменении тарифов.

# Критерии успешного выполнения

1. Функциональность работает (обязательно наличие UI)

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

8. Реализована обработка исключений

9. Подключено логгирование

10. Наличие technical solution description

11. Наличие unit-тестов на бизнес логику

# Используемые технологии

В рамках школы студенты не ограничены в выборе технологий для реализации проекта в рамках стека технологий, используемых в компании в различных проектах (лекции покрывают небольшую часть из них). Список технологий для выбора приведен ниже. Однако проект должен быть написан преимущественно на java (допускается написание незначительных частей, отдельных модулей или дополнительной функциональности на других языках, например, Kotlin или Python). После обсуждения с куратором и по согласованию с ним студент может выбрать любые комбинации технологий и фреймворков, но стоит учитывать, что куратор может быть незнаком с какими-то из них.

Список технологий, используемых в некоторых проектах компании:

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

Задание состоит из 2 частей, каждая часть должна быть реализована с использованием разных неповторяющихся технологий.

Предлагаемый вариант стека технологий для 1 реализации части на основании лекций:

* IDE - Any (Eclipse, IDEA)
* Tomcat
* DB – MySQL
* Maven
* JPA
* Spring Framework (кроме Boot, Data)
* JSP

Для 2 части:

* Maven
* AS – WildFly
* EJB
* JSF
* MQ (для уведомлений от сервера)
* WebServices (для обмена данных между клиентом и сервером)

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

Рекомендуем использовать последнюю стабильную версию технологий.

Плюсом будет использование следующих технологий: Selenium, Sonar, Angular/React (например, админка для табло), Docker, Microservices, использования доступного cloud для развертывания приложения (например, <https://cloud.yandex.ru/> - 1 месяц бесплатного использования) и/или наличие «killer features».