Итоговое задание

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

# Предметная область

Существуют следующие виды сущностей:

* Заказ
  + Клиент
  + Адрес клиента
  + Способ оплаты (Наличный\По карте(дополнительно, сделать тривиальную эмуляцию платёжной системы))
  + Способ доставки
  + Товары
  + Статус оплаты (ожидает оплаты\оплачен)
  + Статус заказа (ожидает оплаты\ожидает отгрузки\отгружен\доставлен)
* Товар
  + Название
  + Цена
  + Категория
  + Параметры (брэнд\цвет\мощность\вес ...)
  + Вес
  + Объём
  + Количество на складе
* Клиент
  + Имя
  + Фамилия
  + Дата рождения
  + Адрес электронной почты
  + Пароль для личного кабинета
* Адрес клиента
  + Страна
  + Город
  + Почтовый код
  + Улица
  + Дом
  + Квартира

Приложение должно обеспечивать следующую функциональность:

* Для клиентов компании
  + Просмотр каталога с возможностью фильтрации по параметрам
  + Просмотр и редактирование профиля
    - Информации
    - Адресов
    - Пароля
  + Заказы
    - Оформление заказа
    - Просмотр истории заказов
    - (Дополнительно) Повторить заказ
* Для сотрудников компании
  + Заказы
    - Просмотр
    - Изменение статуса заказа (отгружен\доставлен\оплачен)
  + Статистика продаж (топ 10 товаров, клиентов, выручка за месяц\неделю)
  + Товары
    - Добавление
    - Создание и управление категориями каталога
    - (Дополнительно) импорт из файла

При совершении покупки на каждой странице до оформления заказа должна отображаться корзина, в которой отображаются выбранные позиции пользователя. Корзина отображается как для гостя, так и для авторизированного пользователя. Причём после авторизации содержимое корзины не теряется. После закрытия страницы корзина не очищается.

# Технические требования

В итоге 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.

# II часть

Реализовать отдельное клиент-приложение для рекламного стенда. Приложение должно отображать топ товаров и их стоимости (или товары, на которые действует скидка). Данные должны подгружаться при старте и храниться на стороне клиента. Перезагрузка данных осуществляется в случае получения уведомления от сервера об изменении товаров и их стоимости.

# 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 выше.

Рекомендуем использовать последнюю стабильную версию технологий, обязательных к использованию.

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».