

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 авторизованного 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.

Используемые технологии:

- IDE - Any (Eclipse, IDEA)
- Tomcat 8/9
- DB – MySQL 5.7
- Maven 3.*
- JPA 2.0

- Spring Framework
- JSP

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.

Используемые технологии:

- Maven 3.x
- AS – WildFly 10.x
- EJB 3.x
- JSF 2.x
- MQ (для уведомлений от сервера)
- WebServices (для обмена данными между клиентом и сервером)

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

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

Плюсом будет использование следующих технологий: Selenium, Sonar, Angular (например, админка для рекламного стенда), Docker, Microservices и/или наличие «killerfeatures».