Техническое задание для онлайн-магазина товаров для дома и сада

1. Введение

1.1. Цель документа

Данный документ представляет собой техническое задание для разработки бэкенда на Java Spring **для онлайн-магазина товаров для дома и сада**. ТЗ предназначено для команды студентов бэкенд-разработчиков итогового проекта.

2. Общее описание

2.1. Обзор

Макет онлайн-магазина (функционал на макете может немного отличаться от функционала этого ТЗ).

Приложение позволяет клиентам выбирать товар из каталога, добавлять в корзину, оформлять заказ и отслеживать его статус в реальном времени. Для администраторов и сотрудников приложение предоставляет инструменты для управления каталогом товаров, заказами, акциями и аналитикой продаж.

2.2. Функции продукта

- Система авторизации и учета пользователей: регистрация, аутентификация и управление учетными записями пользователей.
- Управление каталогом продуктов: добавление, редактирование и удаление товаров в каталоге.
- Возможность добавления товара в корзину с последующей покупкой без реальной оплаты.
- История покупок.
- Управление скидками: возможность указывать скидочную цену на товар.

2.3. Классы пользователей и характеристики

- **Клиенты**: конечные пользователи приложения, желающие купить товар в онлайн-магазине.
- **Администраторы**: администраторы и менеджеры онлайн-магазина, использующие систему для управления каталогом товаров, заказами и акциями.

3. Требования к функциональности

3.1. Управление учетными записями пользователей

Система должна предоставлять возможности для регистрации, аутентификации и управления учетными записями пользователей. Это включает:

- Регистрацию новых пользователей с базовой информацией (имя, email, номер телефона, пароль). Данные из формы должны проходить валидацию.
- Аутентификацию пользователей с использованием email и пароля.
- Редактирование профиля пользователя (имя, контактная информация).
- Удаление учетной записи по запросу пользователя.

3.2. Управление каталогом товаров

Администраторы должны иметь возможность управлять каталогом товаров, включая:

- Добавление новых товаров в каталог с указанием названия, описания, цены, категории и изображения, редактирование существующих товаров и удаление товаров из каталога.
- Управление категориями товаров: добавление новых категорий, редактирование существующих, удаление.
- Категории товаров
    - Fertilizer (Удобрения),
    - Protective products and septic tanks (Защитные средства и септики),
    - Planting material (Посадочный материал),
    - Tools and equipment (Инструменты и оборудование),
    - Pots and planters (Горшки и кашпо).

3.3. Фильтрация и сортировка товаров на сайте

- Просмотр и фильтрация товаров: Пользователь должен иметь возможность фильтровать товары по:
    - цене (мин - макс)
    - наличию скидки
    - категории

- Сортировка товаров:
    - по-умолчанию - Название
    - по цене (увеличение, уменьшение)
    - по дате появления товара
    - по алфавиту в обе стороны

3.4. Система заказов

Система должна предоставлять функциональность для оформления и управления заказами:

- Возможность добавления товара из каталога в корзину и оформление заказа.
- Выбор способа доставки (доставка курьером или самовывоз).

- Ввод адреса доставки и контактных данных для доставки.
- Просмотр статуса заказа.
- Отмена заказа.
- Просмотр истории заказов.

В данном проекте система доставки должна быть реализована при помощи триггера, который каждые 30 секунд меняет статус в базе на следующий.
Список статусов для Доставки:

- **Ожидает оплаты** - заказ добавлен в корзину и ожидает оплаты
- **Оплачен** - заказ оплачен
- **В пути** - транспортировка товара до пользователя
- **Доставлено** - курьер доставил товар (Конечный статус)
- **Отменено** - пользователь отменил заказ

Статус **Отменено** может быть присвоен только если товар в одном из статусов:

- **Ожидает оплаты**
- **Создан**

3.5. Система оплаты

В данном проекте не нужно реализовывать систему оплаты. При нажатии на кнопку купить информация о покупке отправляется на склад, а из корзины удаляются купленные товары. История покупок пополняется данными о купленных товарах: название, описание, фото, количество, стоимость, категория.

3.6. Управление акциями и скидками

Необходимо предусмотреть возможность управления скидками для привлечения и удержания клиентов:

- Возможность добавить скидку на товар.
- Товар дня - товар с наибольшей скидкой (если товаров с одинаковой скидкой несколько, то случайный выбор товара из этого массива).

3.7 Отчетность

Необходимо реализовать функционал отчетности, который включает в себя

- Топ 10 купленных товаров
- Топ 10 часто отменяемых товаров
- Список товаров, которые находятся в статусе **Ожидает оплаты** более N дней
- Прибыль за N дней, месяцев, лет с группировкой по часам, дням, неделям, месяцам.

3.8 Избранные товары

Необходимо реализовать функционал добавления товаров в избранное и получение пользователем списка товаров в избранном.

4. Требования к технологическому стеку

4.1. Язык программирования и фреймворки

- **Язык программирования**: Java 11 или выше.
- **Основной фреймворк**: Spring Boot 2.x.
- **Безопасность**: Spring Security.
- **Обработка и маппинг данных**: Spring Data JPA/Hibernate.
- **Основная СУБД**: PostgreSQL.
- **Подключение к базе данных**: использование Spring Data для интеграции с базой данных с целью обеспечения упрощенной работы с данными и поддержки механизма миграций базы данных через Flyway или Liquibase для управления версиями схемы базы данных.
- **Docker**: должен использоваться для создания контейнеров для каждого компонента системы, включая приложение, базу данных и любые другие зависимости.

5. Требования к интерфейсу

5.1. Внешние интерфейсы (API для фронтенда)

- **RESTful API**: Должен быть разработан RESTful API для обеспечения взаимодействия между бэкендом и фронтендом. API должен поддерживать основные HTTP методы (GET, POST, PUT, DELETE) для обработки запросов.
- **Формат данных**: Обмен данными должен осуществляться в формате JSON.
- **Аутентификация и авторизация**: Для доступа к API должен использоваться JWT (JSON Web Tokens) для обеспечения безопасности данных пользователей.
- **Документация API**: Должна быть предоставлена подробная документация для API с описанием всех эндпоинтов, параметров запросов и ответов. Рекомендуется использовать Swagger или аналогичные инструменты для автоматизации создания документации.

Приложение

1. Описания REST API
2. Схема таблиц базы данных
3. Критерии оценивания

Разбивка выполнения ТЗ по спринтам

Спринт 1: Управление учетными записями пользователей

1. Создать структуру таблиц для базы данных (пример)
2. Создать удаленный репозиторий в гитхаб.
3. Разработка API и бизнес-логики для регистрации новых пользователей.
4. Имплементировать работу с JWT-токеном (секьюрити конфиг)
5. Имплементировать функционал добавления категорий

6. Имплементировать функционал просмотра категорий
7. Создать документацию по API в Swagger

Спринт 2: Работа с каталогом товаров и навигация по сайту

1. Имплементировать функционал удаления и редактирования категорий
2. Имплементировать функционал добавления товаров в каталог с необходимыми атрибутами (название, описание, цена, цена со скидкой, категория, изображение).
3. Имплементировать функционал редактирования и удаления товаров.
4. Создание API для отображения всех товаров, избранных товаров, товаров со скидкой и товаров по категориям и возможностью применить фильтры.

Спринт 3: Система заказов и оплаты

1. Имплементировать функционал добавления товаров в корзину и создание заказа.
2. Имплементировать функционал выбора способа доставки, ввода адреса и контактных данных для доставки.
3. Добавление возможности просмотра истории заказов.
4. Добавление триггера для автоматической смены статуса заказа для доставки

Спринт 4: Управление отчетностью

1. Имплементировать функционал отчетов по ТОПам
2. Имплементировать функционал агрегированных отчетов по прибыли
3. Имплементировать получение списка товаров, находящихся в статусе **Ожидает оплаты** более N дней
4. Связать фронтенд (предоставляется) с бэкендов в работающее приложение
5. Создание README к проекту. Пример структуры.