

Университет ИТМО

Факультет программной инженерии и компьютерной техники

**Лабораторная работа №2**

по курсу «Бизнес-логика программных систем»

Вариант 54

Выполнили:

Батманов Д.Е.

Сафонова О.Д.

Преподаватели:

Цопа Е.А.

Кривоносов Е.Д.

Санкт-Петербург

2025

## Оглавление

Задание .....	3
Требования к реализации.....	<b>Error! Bookmark not defined.</b>
Выполнение .....	4
BPMN.....	<b>Error! Bookmark not defined.</b>
Программная реализация .....	<b>Error! Bookmark not defined.</b>
Реализация скриптов для тестирования .....	<b>Error! Bookmark not defined.</b>
Деплой на сервере helios .....	<b>Error! Bookmark not defined.</b>
Исходный код.....	5
Вывод.....	5

## Задание

Доработать приложение из лабораторной работы #1, реализовав в нём управление транзакциями и разграничение доступа к операциям бизнес-логики в соответствии с заданной политикой доступа.

### Управление транзакциями необходимо реализовать следующим образом

Переработать согласованные с преподавателем прецеденты (или по согласованию с ним разработать новые), объединив взаимозависимые операции в рамках транзакций.

Управление транзакциями необходимо реализовать с помощью Spring JTA.

В реализованных (или модифицированных) прецедентах необходимо использовать программное управление транзакциями.

В качестве менеджера транзакций необходимо использовать Jakarta EE JTA, предварительно преобразовав приложение в war, развёртываемый на сервере приложений WildFly.

### Разграничение доступа к операциям необходимо реализовать следующим образом

Разработать, специфицировать и согласовать с преподавателем набор привилегий, в соответствии с которыми будет разграничиваться доступ к операциям.

Специфицировать и согласовать с преподавателем набор ролей, осуществляющих доступ к операциям бизнес-логики приложения.

Реализовать разработанную модель разграничений доступа к операциям бизнес-логики на базе Spring Security. Информацию об учётных записях пользователей необходимо сохранять в реляционную базу данных, для аутентификации использовать JWT.

### Правила выполнения работы:

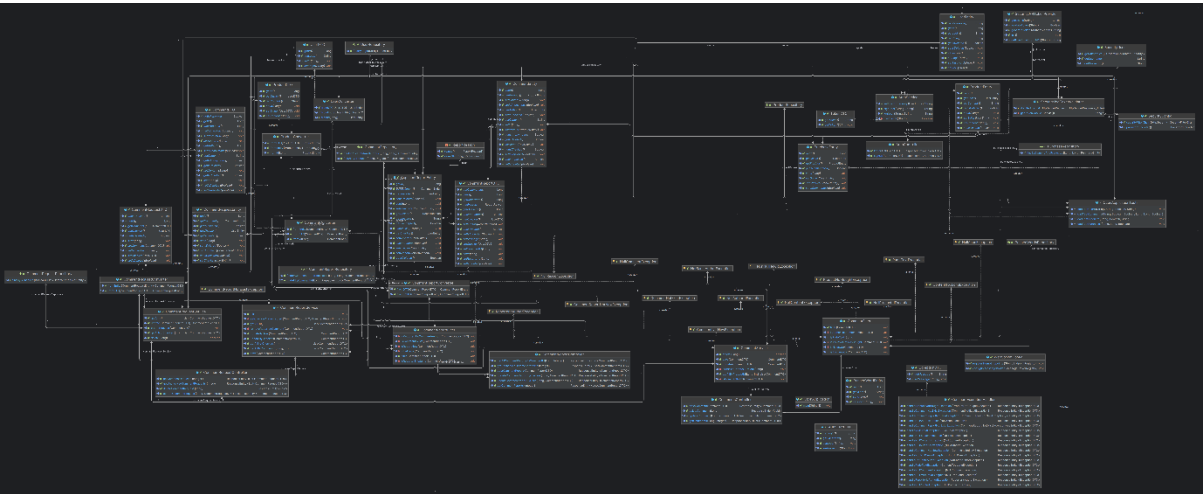
Все изменения, внесённые в реализуемый бизнес-процесс, должны быть учтены в описывающей его модели, REST API и наборе скриптов для тестирования публичных интерфейсов модуля.

Доработанное приложение необходимо развернуть на сервере helios.

# Выполнение

## UML

UML-диаграммы классов и пакетов разработанного приложения:



## Спецификация пользовательских привилегий и ролей, реализованных в приложении

Пакет	Действие	USER	MODER	ADMIN
Auth	Создавать аккаунта	Да	Нет	
	Входить в аккаунт	Да		
Comments	Просматривать комментарии на товар	Да		
	Добавлять комментарий на товар	Да, только если куплен	Нет	
	Изменять текст комментария	Только своего	Нет	Да
	Удалять комментарий	Только свой	Нет	Да
Comment requests	Получать все заявки на комментарий	Нет		Да
	Получать все непроверенные заявки на комментарий	Нет		Да
	Изменить заявку на комментарий	Нет		Да
	Удалять заявку на комментарий	Нет		Да

Comment reports	Получать все заявки на жалобу на комментарий	Нет	Да
	Получать все непроверенные заявки на комментарий	Нет	Да
	Получать все непроверенные заявки на комментарий по ID товара	Нет	Да
	Оставлять жалобу на комментарий	Да, но не на свой	Нет
	Изменять заявку на жалобу на комментарий	Да, но на свой	Нет

## Спецификация REST API для всех публичных интерфейсов разработанного приложения.

Спецификация REST API приведена по ссылке:

[https://github.com/snOlga/web3.0/blob/main/ozon/OZON%20API.postman\\_collection.json](https://github.com/snOlga/web3.0/blob/main/ozon/OZON%20API.postman_collection.json)

## Исходный код

Исходный код приведён по ссылке: <https://github.com/snOlga/web3.0>

Где

CodeAxeAttacks – Батманов Д.Е.

snOlga – Сафонова О.Д.

## Вывод

В ходе выполнения лабораторной работы была успешно реализована система управления транзакциями и разграничения доступа для приложения маркетплейса. Основное внимание уделялось двум ключевым аспектам: работе с транзакциями через Spring JTA и организации системы безопасности на базе Spring Security.

Для управления транзакциями были выделены взаимозависимые операции бизнес-логики, такие как создание комментария с одновременным обновлением рейтинга товара. Эти операции были объединены в транзакции с использованием Spring JTA. Приложение было преобразовано в WAR-архив для совместимости с Jakarta EE JTA и развернуто на сервере WildFly, что позволило обеспечить надежное выполнение распределенных транзакций.

В части безопасности разработана и реализована модель ролей, включающая USER, MODER и ADMIN, с четким распределением прав доступа. Аутентификация пользователей организована через JWT-токены, а информация об учетных записях хранится в реляционной базе данных PostgreSQL. Механизмы авторизации, построенные на аннотациях Spring Security, обеспечивают контроль доступа к REST-эндпоинтам в соответствии с назначенными ролями.

Проделанная работа позволила получить практический опыт в организации распределенных транзакций и реализации сложных систем авторизации в enterprise-приложениях. Результаты лабораторной работы могут служить основой для дальнейшего расширения функциональности системы.