Университет ИТМО

Факультет программной инженерии и компьютерной техники

**Лабораторная работа №3**

по курсу «Бизнес-логика программных систем»

Вариант 453

Выполнили:

Батманов Д.Е.

Сафонова О.Д.

Преподаватели:

Цопа Е.А.

Кривоносов Е.Д.

Санкт-Петербург

2025

Оглавление

[Задание 3](#_Toc197779295)

[Требования к реализации асинхронной обработки 3](#_Toc197779296)

[Требования к реализации распределённой обработки 3](#_Toc197779297)

[Требования к реализации запуска периодических задач по расписанию 3](#_Toc197779298)

[Требования к интеграции с внешней Корпоративной Информационной Системой (EIS) 4](#_Toc197779299)

[Правила выполнения работы 4](#_Toc197779300)

[Выполнение 4](#_Toc197779301)

[BPMN модель 4](#_Toc197779302)

[Спецификация REST API для всех публичных интерфейсов разработанного приложения. 6](#_Toc197779303)

[Исходный код 6](#_Toc197779304)

[UML-диаграммы классов и пакетов разработанного приложения 6](#_Toc197779305)

[Диаграмма развёртывания (Deployment Diagram), визуализирующая интеграцию с EIS 10](#_Toc197779306)

[Схема взаимодействия для JCA: 10](#_Toc197779307)

[Вывод 10](#_Toc197779308)

# Задание

Доработать приложение из лабораторной работы #2, реализовав в нём асинхронное выполнение задач с распределением бизнес-логики между несколькими вычислительными узлами и выполнением периодических операций с использованием планировщика задач, а также интеграцию с внешней информационной системой.

## Требования к реализации асинхронной обработки

1. Перед выполнением работы неободимо согласовать с преподавателем набор прецедентов, в реализации которых целесообразно использование асинхронного распределённого выполнения задач. Если таких прецедентов использования в имеющейся бизнес-процесса нет, нужно согласовать реализацию новых прецедентов, доработав таким образом модель бизнес-процесса из лабораторной работы #1.
2. Асинхронное выполнение задач должно использовать модель доставки "подписка".
3. В качестве провайдера сервиса асинхронного обмена сообщениями необходимо использовать сервис подписки на базе Apache Kafka + ZooKeeper.
4. Для отправки сообщений необходимо использовать Kafka Producer API.
5. Для получения сообщений необходимо использовать API KafkaConsumer (org.apache.kafka.clients.consumer.KafkaConsumer).

## Требования к реализации распределённой обработки

1. Обработка сообщений должна осуществляться на двух независимых друг от друга узлах сервера приложений.
2. Если логика сценария распределённой обработки предполагает транзакционность выполняемых операций, они должны быть включены в состав распределённой транзакции.

## Требования к реализации запуска периодических задач по расписанию

1. Согласовать с преподавателем прецедент или прецеденты, в рамках которых выглядит целесообразным использовать планировщик задач. Если такие прецеденты отсутствуют -- согласовать с преподавателем новые и добавить их в модель автоматизируемого бизнес-процесса.
2. Реализовать утверждённые прецеденты с использованием планировщика задач Spring (@Scheduled).

## Требования к интеграции с внешней Корпоративной Информационной Системой (EIS)

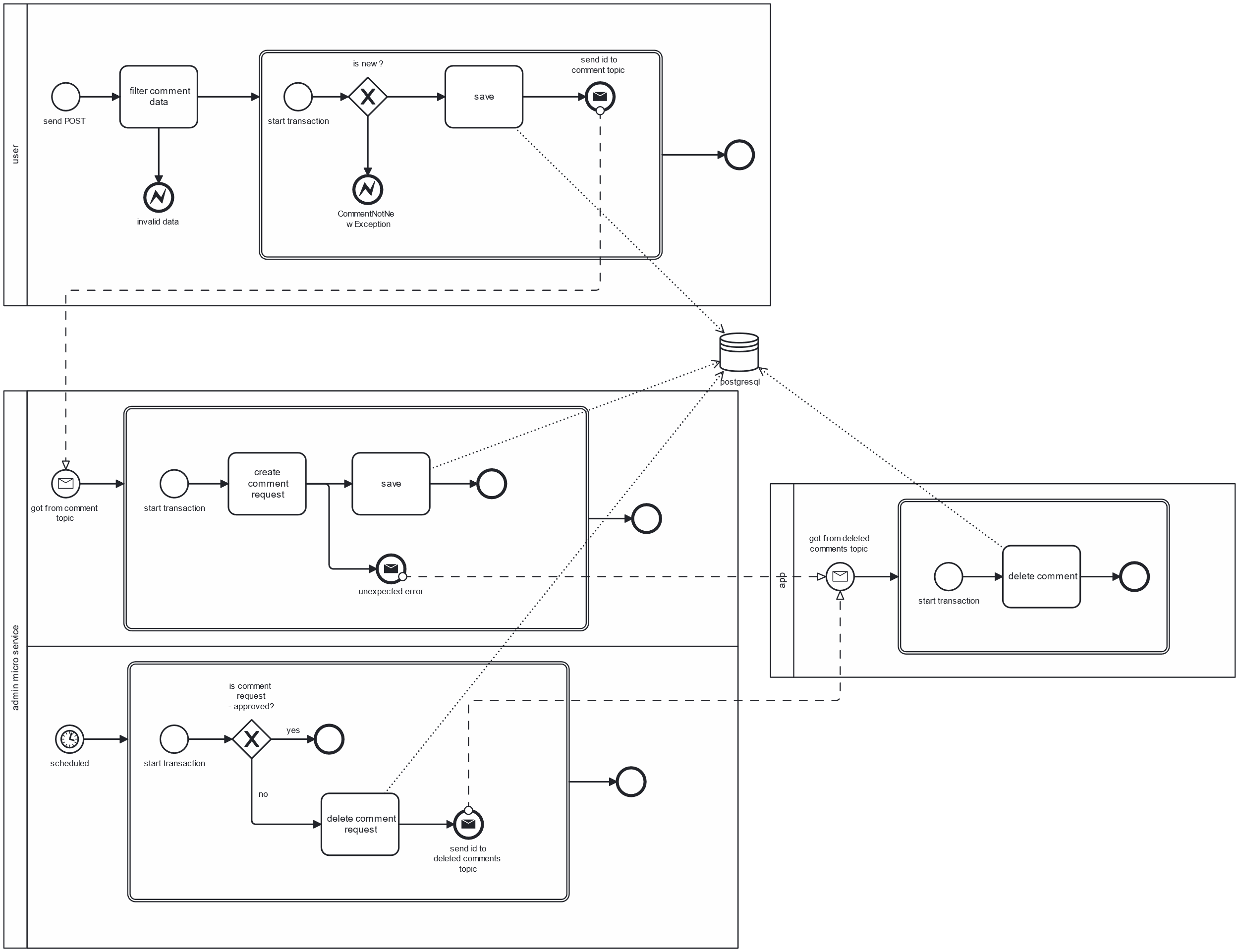
1. Корпоративная Информационная Cистема, с которой производится интеграция, а также её функциональные возможности выбираются на усмотрение преподавателя и согласуются с ним.
2. Взаимодействие с внешней Копроративной Информационной Системой должно быть реализовано с помощью технологии JCA (Jakarta Connectors).

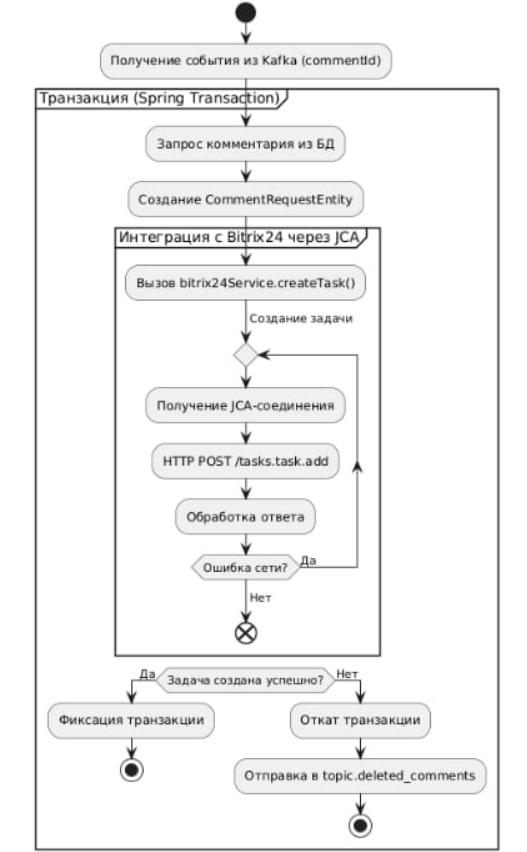
## Правила выполнения работы

1. Все изменения, внесённые в реализуемый бизнес-процесс, должны быть учтены в описывающей его модели, REST API и наборе скриптов для тестирования публичных интерфейсов модуля.
2. Доработанное приложение необходимо либо развернуть на сервере helios, либо продемонстрировать его работоспособность на собственной инфраструктуре обучающегося.

# Выполнение

## BPMN модель





# 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>

## Исходный код

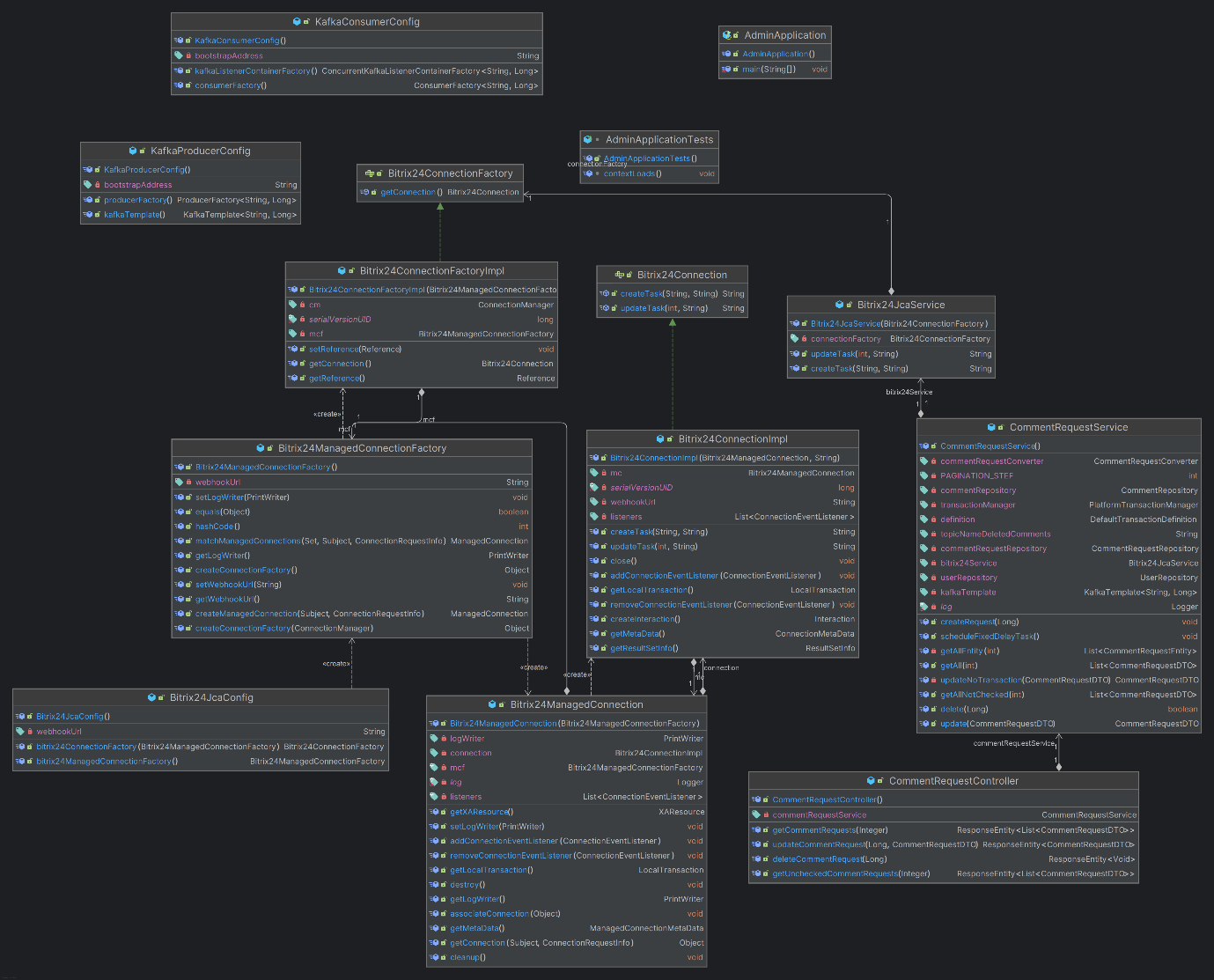
Исходный код приведён по ссылке: <https://github.com/snOlga/web3.0>

Где

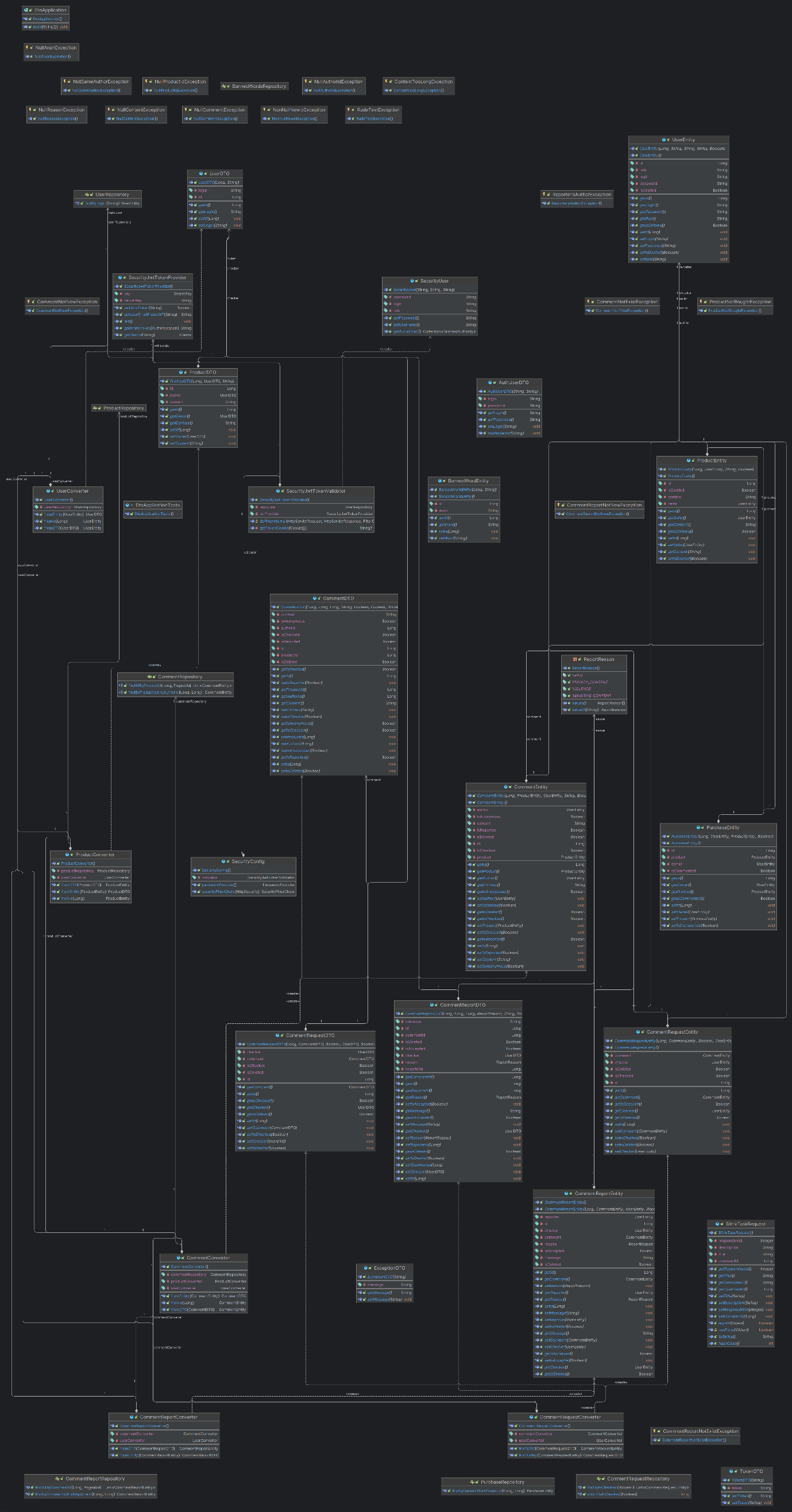
CodeAxeAttacks – Батманов Д.Е.

snOlga – Сафонова О.Д.

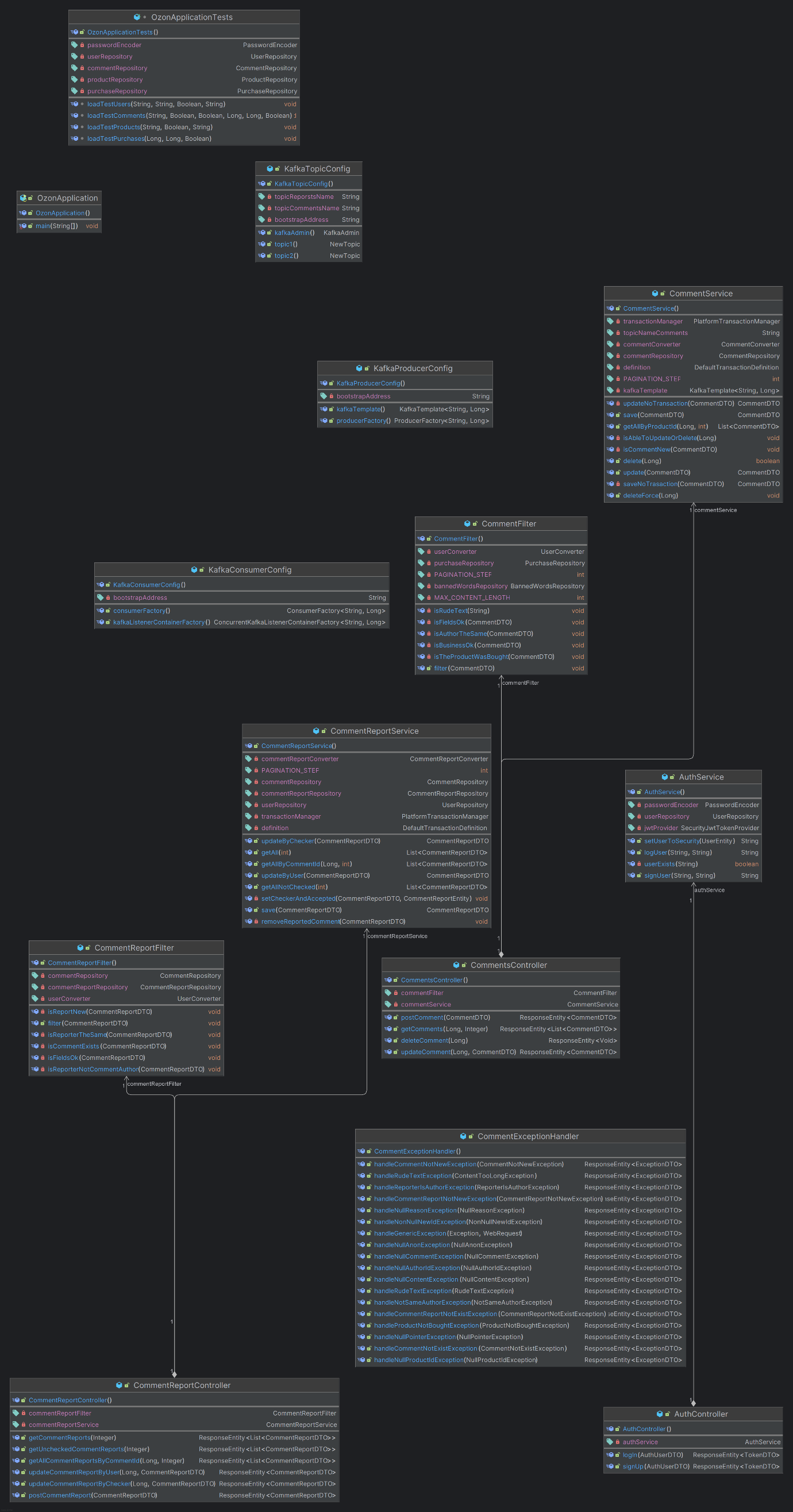
## UML-диаграммы классов и пакетов разработанного приложения

admin:  


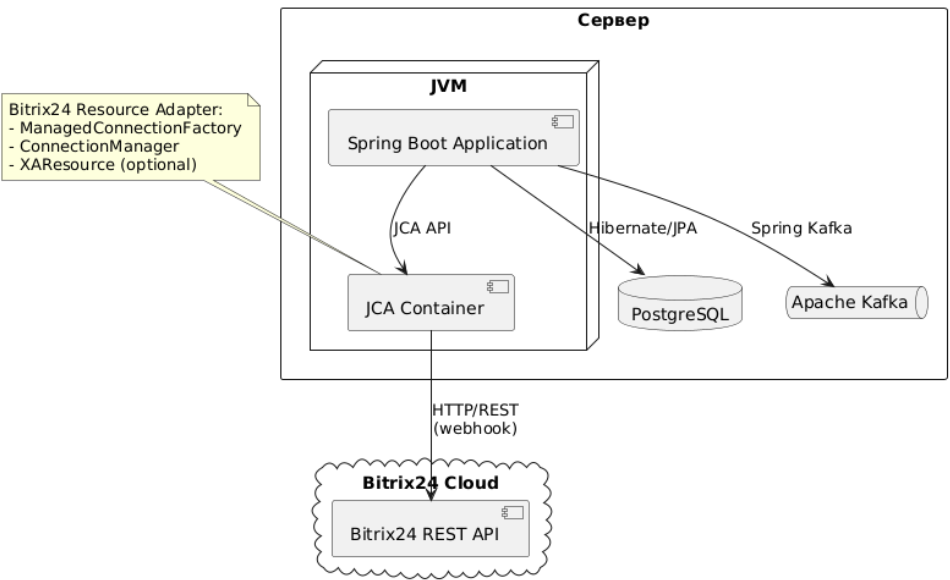
dto:



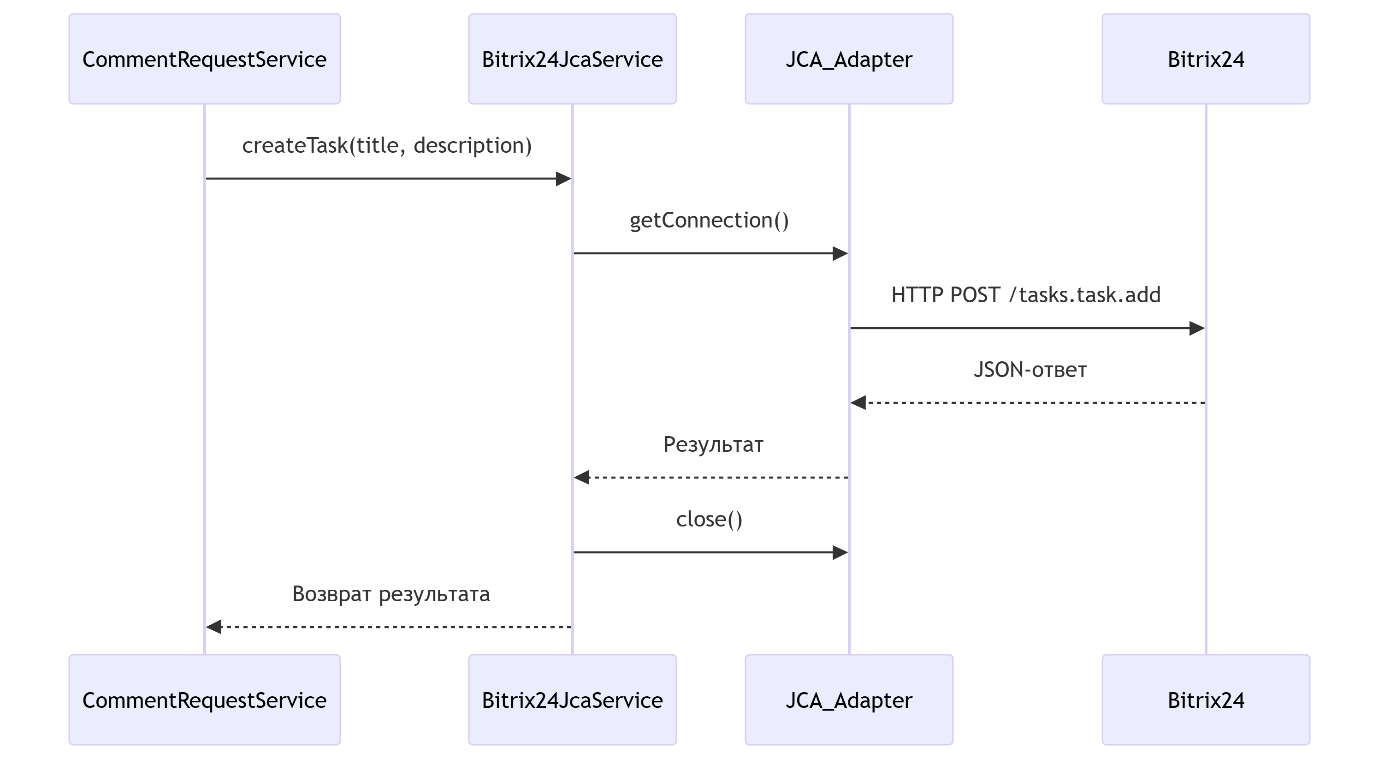
ozon:



## Диаграмма развёртывания (Deployment Diagram), визуализирующая интеграцию с EIS



## Схема взаимодействия для JCA:



# Вывод

В ходе выполнения лабораторной работы монолитное приложение было разбито на микросервисы с асинхронным сообщением с помощью брокера сообщений Kafka.

Было внедрено задание с выполнением по расписанию. Взаимодействие с внешней Корпоративной Информационной Системой (Битрикс24 )реализовано с помощью технологии JCA (Jakarta Connectors).