МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ

Национальная научно-образовательная корпорация ИТМО

Факультет программной инженерии и компьютерной техники

ОТЧЕТ ПО ЛАБОРАТОРНОЙ РАБОТЕ №1

по дисциплине

«Бизнес-логика программных систем»

Выполнил:

Студент группы P3309

Муратов Михаил Александрович

Преподаватель:

Кривоносов Егор Дмитриевич

Санкт-Петербург, 2025

Задание

Лабораторная работа #1

Введите вариант 789

Вариант №789: Деливери — быстрая доставка еды и продуктов — https://market-delivery.yandex.ru. Бизнес-процесс: работа с интерфейсом продавца — управление заказами, работа с курьерами.

Описать бизнес-процесс в соответствии с нотацией BPMN 2.0, после чего реализовать его в виде приложения на базе Spring Boot.

Порядок выполнения работы:

1. Выбрать один из бизнес-процессов, реализуемых сайтом из варианта задания.
2. Утвердить выбранный бизнес-процесс у преподавателя.
3. Специфицировать модель реализуемого бизнес-процесса в соответствии с требованиями BPMN 2.0.
4. Разработать приложение на базе Spring Boot, реализующее описанный на предыдущем шаге бизнес-процесс. Приложение должно использовать СУБД PostgreSQL для хранения данных, для всех публичных интерфейсов должны быть разработаны REST API.
5. Разработать набор curl-скриптов, либо набор запросов для REST клиента Insomnia для тестирования публичных интерфейсов разработанного программного модуля. Запросы Insomnia оформить в виде файла экспорта.
6. Развернуть разработанное приложение на сервере `helios`.

Содержание отчёта:

1. Текст задания.
2. Модель потока управления для автоматизируемого бизнес-процесса.
3. UML-диаграммы классов и пакетов разработанного приложения.
4. Спецификация REST API для всех публичных интерфейсов разработанного приложения.
5. Исходный код системы или ссылка на репозиторий с исходным кодом.
6. Выводы по работе.

Вопросы к защите лабораторной работы:

1. Понятие бизнес-логики в программных системах. Уровень бизнес-логики в многоуровневой архитектуре программных систем.
2. Основные концепции, используемые при разработке бизнес-логики. CDI, IoC, управление транзакциями, безопасность, распределённая обработка данных.
3. Моделирование бизнес-процессов. BPM и BPMN.
4. Спецификация BPMN 2.0. Принципы составления и основные элементы моделей бизнес-процессов.
5. Объекты потока управления, роли и артефакты в BPMN.
6. Использование Spring Framework для реализации бизнес-логики. Реализация CDI и IoC. Связь уровня бизнес-логики с другими уровнями архитектуры программных систем в Spring.
7. Spring Boot. Способы конфигурации бинов. Двухфазовый, трёхфазовый конструктор.
8. Профили запуска приложения в Spring Boot.

Код

https://github.com/foliageh/itmo-blps-lab1

BPMN-диаграмма

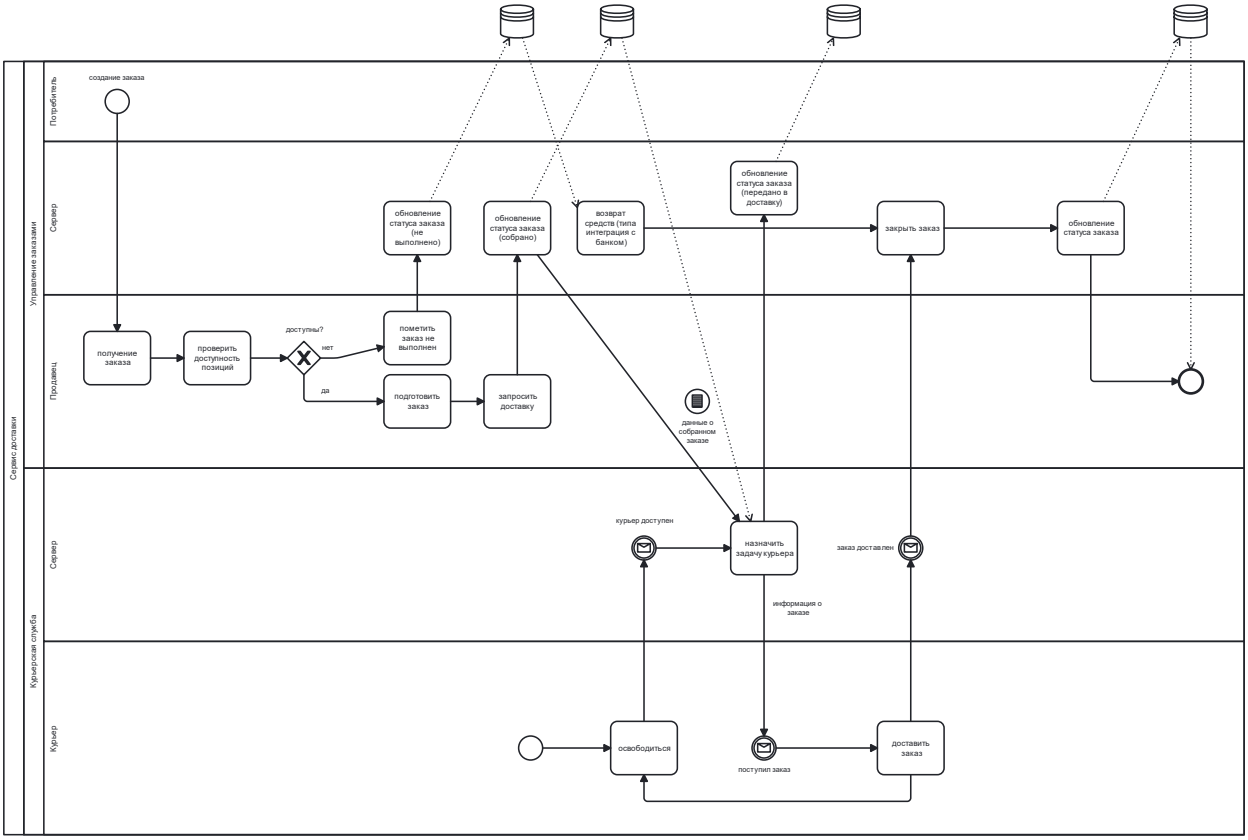
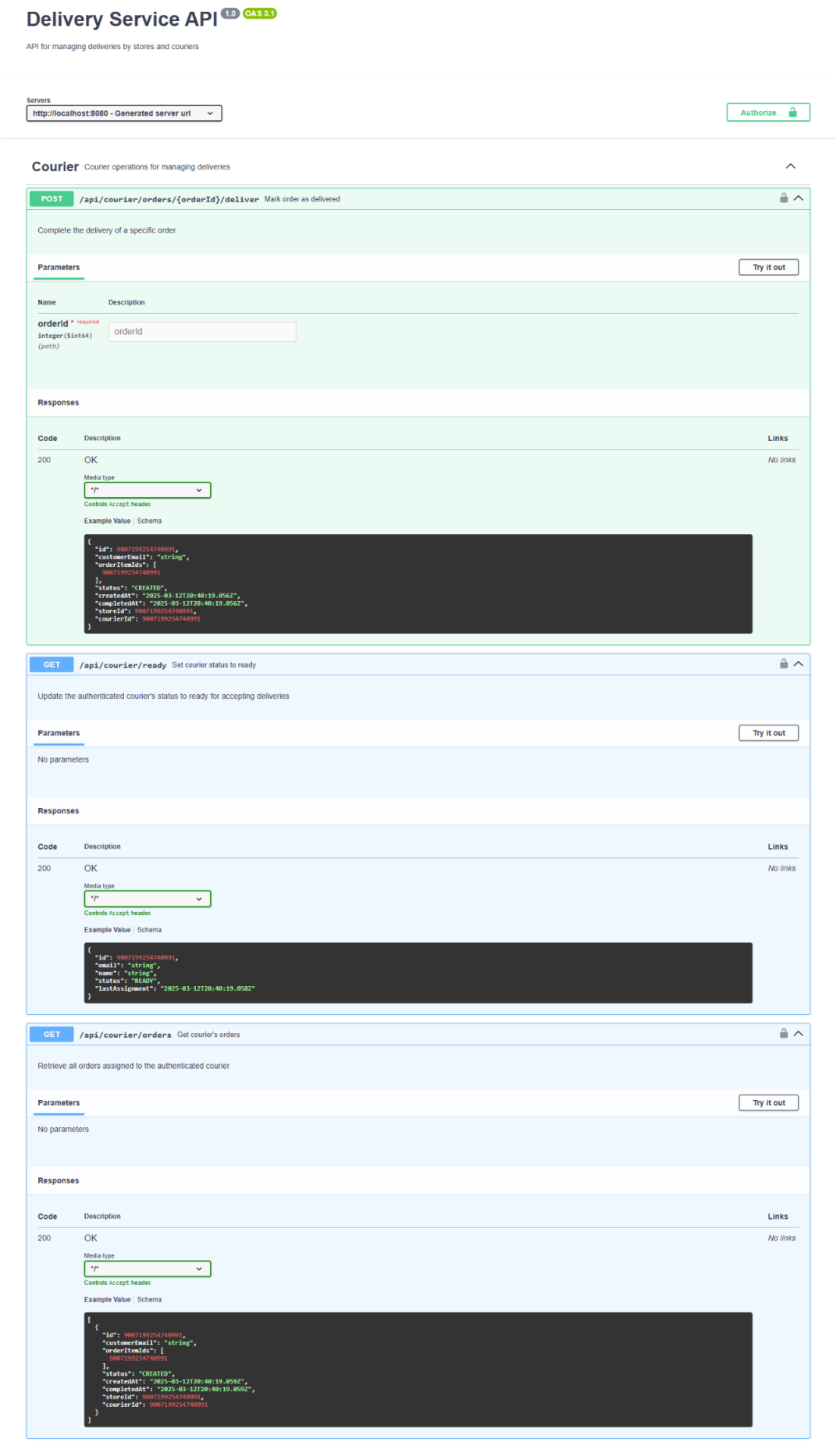


Диаграмма классов

Слишком большая, см. файл UML.pdf в репозитории.

Спецификация API



Store Store operations for managing orders

POST /api/store/orders/{orderId}/collect Mark order as collected

Parameters | Try it out

Name	Description
orderId * required integer($int64) *(path)*	orderId

Responses

Code	Description	Links
200	OK	*No links*

Media type: */*

Controls Accept header.

Example Value | Schema


```
{
  "id": 9007199254740991,
  "customerEmail": "string",
  "orderItemIds": [
    9007199254740991
  ],
  "status": "CREATED",
  "createdAt": "2025-03-12T20:40:19.061Z",
  "completedAt": "2025-03-12T20:40:19.061Z",
  "storeId": 9007199254740991,
  "courierId": 9007199254740991
}
```


POST /api/store/orders/{orderId}/cancel Cancel an order

Cancel a specific order that hasn't been collected yet

Parameters | Try it out

Name	Description
orderId * required integer($int64) *(path)*	orderId

Responses

Code	Description	Links
200	OK	*No links*

Media type: */*

Controls Accept header.

Example Value | Schema


```
{
  "id": 9007199254740991,
  "customerEmail": "string",
  "orderItemIds": [
    9007199254740991
  ],
  "status": "CREATED",
  "createdAt": "2025-03-12T20:40:19.064Z",
  "completedAt": "2025-03-12T20:40:19.064Z",
  "storeId": 9007199254740991,
  "courierId": 9007199254740991
}
```


GET /api/store/orders Get store's orders

Retrieve all orders associated with the authenticated store

Parameters | Try it out

No parameters

Responses

Code	Description	Links
200	OK	*No links*

Media type: */*

Controls Accept header.

Example Value | Schema


```
[
  {
    "id": 9007199254740991,
    "customerEmail": "string",
    "orderItemIds": [
      9007199254740991
    ],
    "status": "CREATED",
    "createdAt": "2025-03-12T20:40:19.065Z",
    "completedAt": "2025-03-12T20:40:19.065Z",
    "storeId": 9007199254740991,
    "courierId": 9007199254740991
  }
]
```

Authentication

Authentication endpoints for stores and couriers

POST /api/auth/store/register

Register a new store

Parameters

Try it out

No parameters

Request body required

application/json

Example Value | Schema

```
{
  "name": "string",
  "email": "string",
  "password": "string"
}
```

Responses

Code	Description	Links

200	OK
	Media type: */*
	Controls Accept header.
	Example Value | Schema
	{
	"additionalProp1": "string",
	"additionalProp2": "string",
	"additionalProp3": "string"
	}
		No links

POST /api/auth/store/login

Login as a store

Parameters

Try it out

No parameters

Request body required

application/json

Example Value | Schema

```
{
  "email": "string",
  "password": "string"
}
```

Responses

Code	Description	Links

200	OK
	Media type: */*
	Controls Accept header.
	Example Value | Schema
	{
	"token": "string"
	}
		No links

POST /api/auth/courier/register

Register a new courier

Parameters

Try it out

No parameters

Request body required

application/json

Example Value | Schema

```
{
  "name": "string",
  "email": "string",
  "password": "string"
}
```

Responses

Code	Description	Links

200	OK
	Media type: */*
	Controls Accept header.
	Example Value | Schema
	{
	"additionalProp1": "string",
	"additionalProp2": "string",
	"additionalProp3": "string"
	}
		No links

POST /api/auth/courier/login

Login as a courier

Parameters

Try it out

No parameters

Request body required

application/json

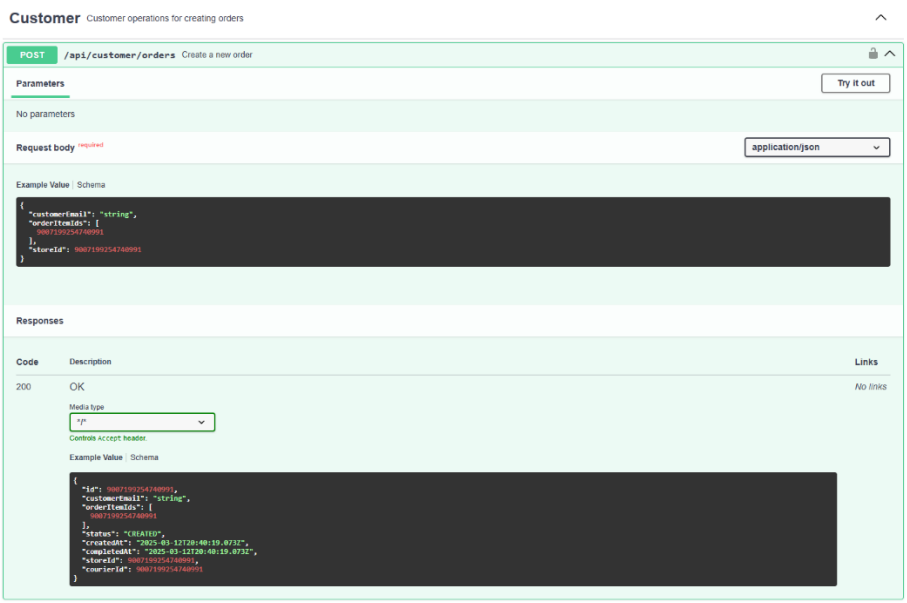
Example Value | Schema

```
{
  "email": "string",
  "password": "string"
}
```

Responses

Code	Description	Links

200	OK
	Media type: */*
	Controls Accept header.
	Example Value | Schema
	{
	"token": "string"
	}
		No links



Вывод

В рамках лабораторной работы я описал бизнес-процесс управления заказами и взаимодействия с курьерами в нотации BPMN 2.0, а затем реализовал его в виде приложения на Spring Boot с использованием PostgreSQL. В процессе разработки я научился проектировать бизнес-процессы с учетом требований BPMN, создавать REST API на их основе и тестировать функционал с помощью Insomnia.