

Санкт-Петербургский национально исследовательский университет
информационных технологий, механики и оптики
Факультет программной инженерии и компьютерной техники



Лабораторная работа № 1 по дисциплине
«Бизнес-логика программных систем»

Вариант 331315.

Выполнили:

Студенты группы Р33131

Осипов Василий
Ибадуллаев Алибаба

Преподаватель: Кулинич Я. В.

Санкт-Петербург

2023

Задание.

Введите вариант

Вариант №331315: Letyshops - <http://letyshops.ru/>

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**.

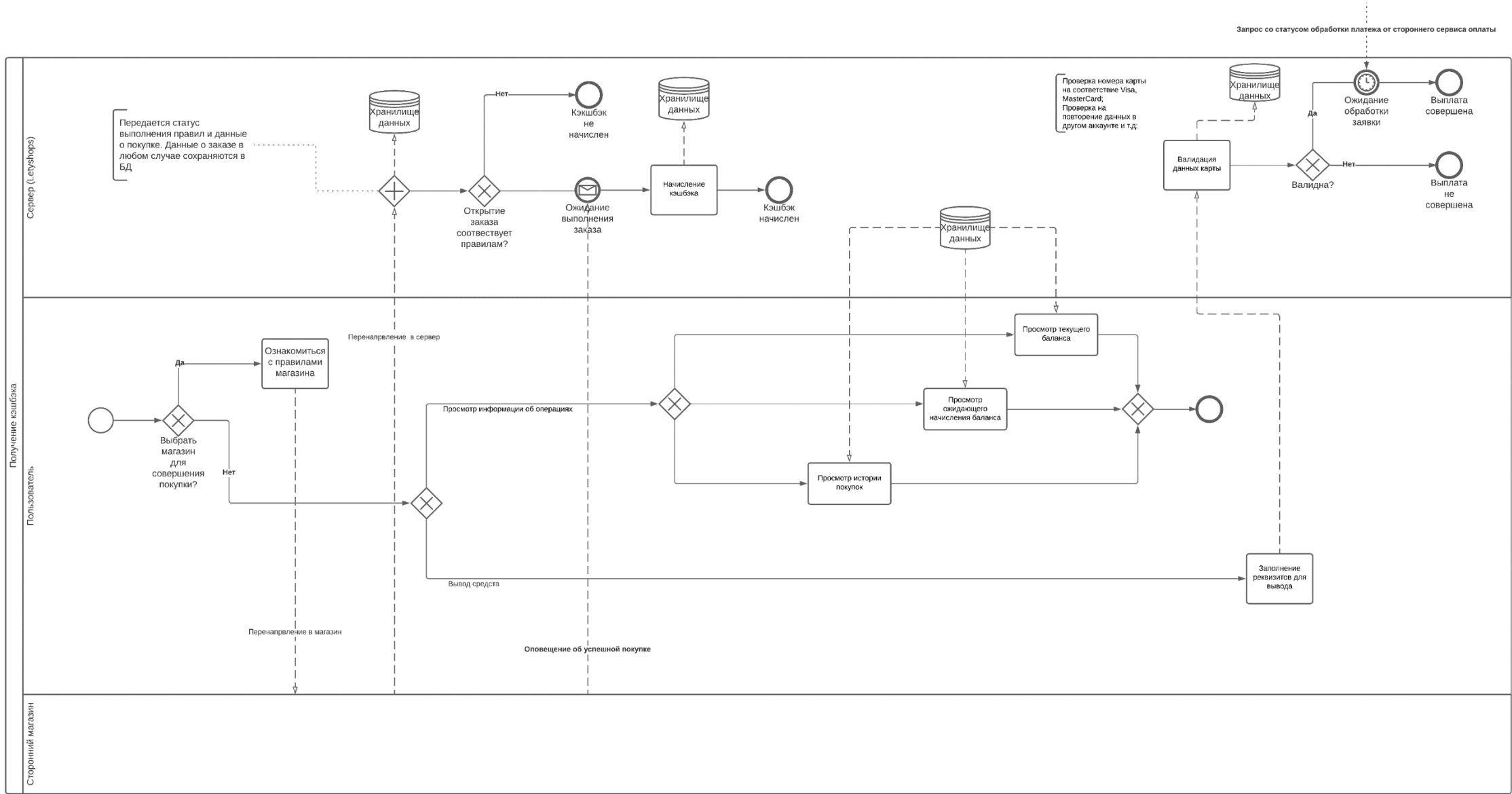
Содержание отчёта:

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.

Выбранный бизнес-процесс:

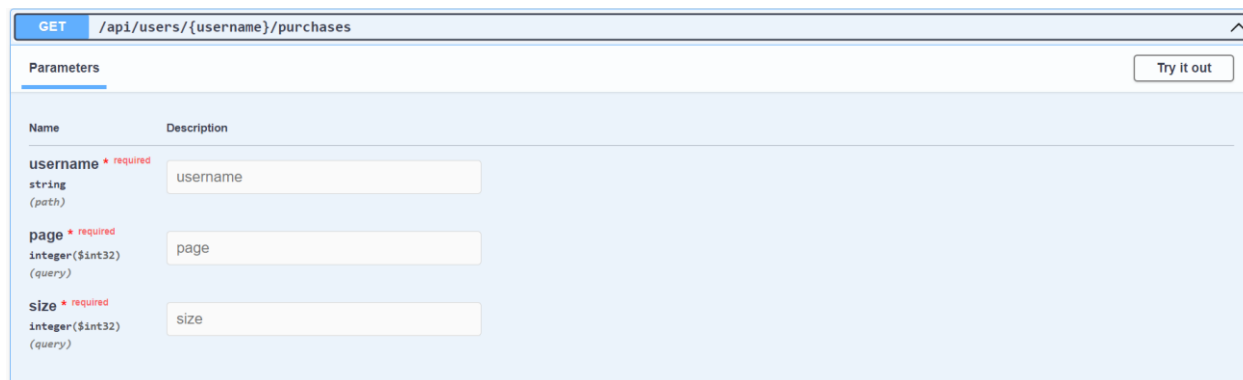
Процесс начисления кэшбека с возможностью вывода средств и просмотра связанной информации о балансе и истории начисления.

BPMN:



Спецификация REST API разработанного приложения

Все покупки пользователя:



GET /api/users/{username}/purchases

Parameters

Name	Description
username * required string (path)	username
page * required integer(\$int32) (query)	page
size * required integer(\$int32) (query)	size

Ответы:

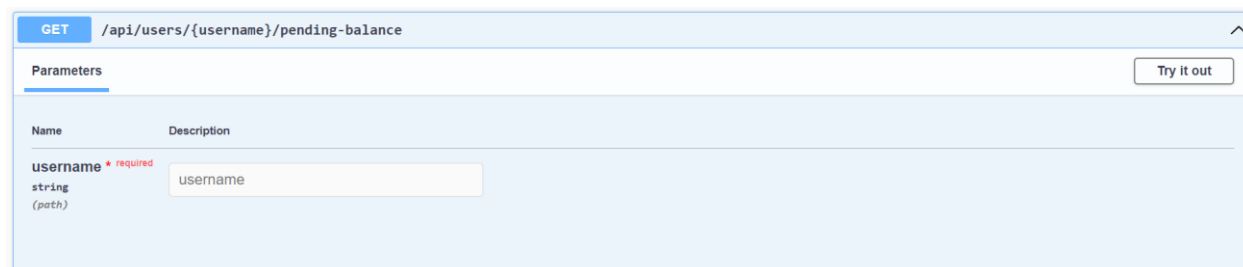
200 – OK

```
1  {
2    {
3      "id": 27,
4      "marketplaceName": "fghfgh",
5      "cashbackPaymentStatus": "REJECTED",
6      "rulesRespected": true,
7      "timestamp": "2023-02-24T20:05:00.517+00:00",
8      "cashbackPercent": 3.0,
9      "totalPrice": 29215.0
10   },
11   {
12     "id": 26,
13     "marketplaceName": "gdsfg",
14     "cashbackPaymentStatus": "REJECTED",
15     "rulesRespected": true,
16     "timestamp": "2023-02-24T20:04:58.493+00:00",
17     "cashbackPercent": 7.0,
18     "totalPrice": 20700.0
19   }
20 }
```

400 – Bad Request – некорректные параметры страницы

404 – Not Found – не найден указанный пользователь

Замороженный баланс пользователя:



GET /api/users/{username}/pending-balance

Parameters

Name	Description
username * required string (path)	username

Ответы:

200 – OK

1 228.0

404 – Not Found – не найден указанный пользователь

Доступный баланс пользователя:

GET /api/users/{username}/available-balance

Parameters

Try it out

Name	Description
username * required	
string (path)	username

Ответы:

200 – ОК

1 12676.97

404 – Not Found – не найден указанный пользователь

Переход на сайт магазина для совершения покупки:

POST /api/users/redirect

Parameters

Try it out

No parameters

Request body required

application/json

Example Value | Schema

```
{  "userId": "string",  "marketplaceId": 0}
```

Ответы:

200 – ОК

404 – Not Found – не найден пользователь с таким именем или маркетплейс

Запрос на вывод средств:

POST /api/users/withdraw

Parameters

Try it out

No parameters

Request body required

application/json

Example Value | Schema

```
{  "amount": 0,  "paymentMethodId": 0,  "username": "string",  "credential": "string"}
```

Ответы:

200 – OK

400 – Bad Request – некорректные данные о выводе

404 – Not Found – не найден пользователь с таким именем

Ответ на запрос о выводе средств от платежной системы:

The screenshot shows a REST client interface for a POST request to the endpoint `/api/acquire/approve-withdraw`. The interface includes a 'Parameters' section with 'No parameters', a 'Request body' section marked as 'required' with a dropdown menu set to 'application/json', and an 'Example Value' section showing a JSON object: `{ "withdrawId": 0, "isApproved": true }`. A 'Try it out' button is visible in the top right corner.

Ответы:

200 – OK

400 – Bad Request – некорректный id заявки на вывод

Совершение покупки в магазине (открытие заказа):

The screenshot shows a REST client interface for a POST request to the endpoint `/api/marketplaces/purchase`. The interface includes a 'Parameters' section with 'No parameters', a 'Request body' section marked as 'required' with a dropdown menu set to 'application/json', and an 'Example Value' section showing a JSON object: `{ "cashbackPercent": 0, "totalPrice": 0, "marketplaceId": 0, "username": "string" }`. A 'Try it out' button is visible in the top right corner.

Ответы:

200 – OK

400 – Bad Request – указанную покупку нельзя подтвердить

404 – Not Found – не найден переход пользователя в указанный магазин

Подтверждение от магазина о закрытии заказа:

POST /api/marketplaces/approve-purchase

Parameters Try it out

No parameters

Request body *required* application/json

Example Value | Schema

```
{  "purchaseId": 0,  "isApproved": true,  "marketplaceId": 0}
```

Ответы:

200 – OK

400 – Bad Request – указанную покупку нельзя подтвердить

Получение списка всех магазинов:

GET /api/marketplaces

Parameters Try it out

Name	Description
page * <i>required</i> integer(\$int32) (query)	page
size * <i>required</i> integer(\$int32) (query)	size

Ответы:

200 – OK

```
1  {
2      {
3          "id": 1,
4          "rules": {
5              "id": 1,
6              "minPrice": 10000.0,
7              "certificates": true,
8              "promocodes": false
9          },
10         "description": "Top 1 in SPB flower shop",
11         "name": "Kazflor"
12     },
13     {
14         "id": 2,
15         "rules": {
16             "id": 1,
17             "minPrice": 10000.0,
18             "certificates": true,
19             "promocodes": false
20         },
21         "description": "sdfg",
22         "name": "gdsfg"
23     }
24 }
```

400 – Bad Request – некорректные параметры страницы

Добавление нового пользователя:

Пример работы программы:

Make purchase witch cashback - Run results

Run Again

Automate Run

+ New Run

Run on Yesterday, 17:27:38

View all runs

Source	Environment	Iterations	Duration	All tests	Avg. Resp. Time
Runner	BLPS	1	2s 961ms	8	84 ms

All Tests

Passed (8)

Failed (0)

Skipped (0)

Iteration 1

GET

Get available balance Copy

http://localhost:8779/api/users/twbeamer/available-balance

/ Get available balance Copy

200 OK

245

Pass

Status code is 200

GET

Get pending balance Copy

http://localhost:8779/api/users/twbeamer/pending-balance

/ Get pending balance Copy

200 OK

12

Pass

Status code is 200

GET

Get marketplaces Copy

http://localhost:8779/api/marketplaces?page=1&size=5

/ Get marketplaces Copy

200 OK

115

Pass

Status code is 200

POST

Add redirect Copy

http://localhost:8779/api/users/redirect

/ Add redirect Copy

200 OK

130

Pass

Status code is 200

POST

Purchase from marketplace Copy

http://localhost:8779/api/marketplaces/purchase

/ Purchase from marketplace Copy

200 OK

107

Pass

Status code is 200

GET

Get pending balance Copy

http://localhost:8779/api/users/twbeamer/pending-balance

/ Get pending balance Copy

200 OK

15

Pass

Status code is 200

Cookies

Capture requests

Runner

Вывод.

В ходе выполнения данной лабораторной работы мы познакомились с нотацией для описания бизнес-процессов BPMN = 2.0 и применили её на практике. Так же вспомнили концепции разработки REST приложения на Spring Boot и реализовали бизнес-процесс. Попрактиковались в составлении EntityGraph для запросов к базе данных.