Оглавление

[1. Глоссарий (термины и определения) 2](#_Toc138783850)

[2. Описание бизнес-процесса описание ролей процесса, алгоритм работы этих ролей, описание самого процесса 3](#_Toc138783851)

[3. Диаграммы BPMN, UML Use Case 4](#_Toc138783852)

[4. Бизнес-цели 5](#_Toc138783853)

[5. Функциональные требования, требования к разработке 6](#_Toc138783854)

# Глоссарий (термины и определения)

* 1. **Описание задачи**

«Я, как пользователь интернет магазина [toy.ru](https://toy.ru/), хочу видеть на витрине (сайт) продукты с контентом и ценами (только те продукты которые есть в наличии). С учетом его локации которая определяется на сайте. Вводная по архитектуре: определять остатки нужно по локации пользователя, есть система PIM которая по API может отдавать картинки и прочий продуктовый контент, также есть сиcтема SIS которая по локации может отдавать актуальные стоки и цены.»

# Описание бизнес-процесса описание ролей процесса, алгоритм работы этих ролей, описание самого процесса

Пользователь заходит на сайт, всплывает окно с просьбой указать город, если пользователь отказывается, данные определяются автоматически по его API, если же вводит город, то проверяется корректность введенных данных.

Далее идет обновление и поиск товаров с их ценой в зависимости от города, который выбрал пользователь.

Когда пользователь выбирает товар, он падает в корзину и либо товар остается там и при выходе со страницы кэш обновляется либо же пользователь решает купить его.

Если пользователь хочет, чтобы товар сохранился в корзине, то входит в свой аккаунт, либо регистрирует новый. Это касается и покупки товара.

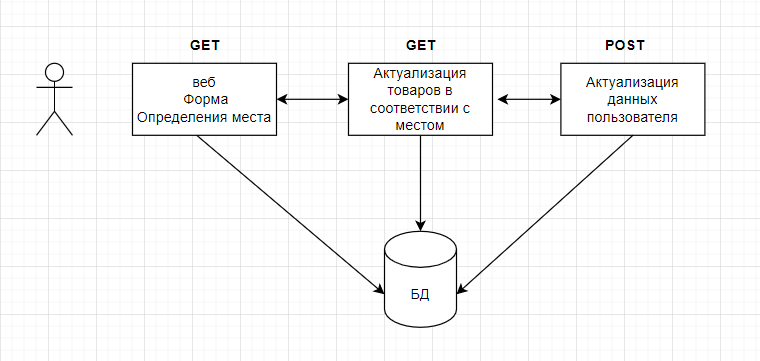
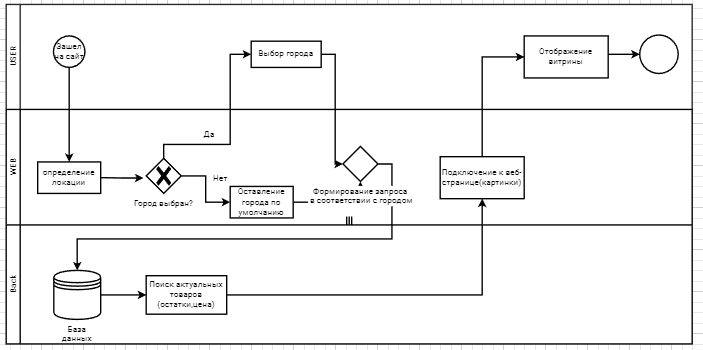


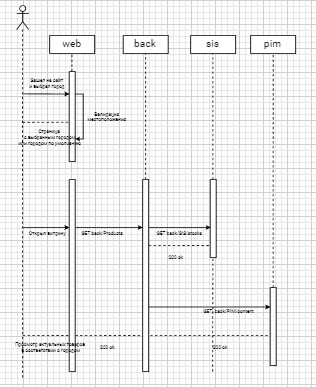
Рисунок 1 - Бизнес процесс сервера 1 и 2

# Диаграммы BPMN, UML Use Case

BPMN



UML



# Бизнес-цели

Перечень бизнес-целей, которые будут достигнуты после решения задачи:

|  |  |
| --- | --- |
| 1. | Авто определение города пользователя |
| 2. | Выбор города пользователем |
| 3. | Показ только актуальных товаров в соответствии с местом |
| 4. | Показ актуальных цен на товары |

# Функциональные требования, требования к разработке

***Сервис 1***

Разработать АПИ для актуализации города

**Алгоритм работы**

При вызове метода сервис обновляет информацию, запросив ее у устройства о местоположении, либо же пользователь введет ее сам, далее информация попадет в регистр.

Если пользователь не вводит город, то город остается по умолчанию Москва.

**Пример запроса**

SELECT city\_id, name

FROM margin

**Пример ответа**

City\_id| name

**Query Params**

[

{

City\_id: number

Name: string

}

]

|  |  |
| --- | --- |
| Cityd | City\_id |
| Name | name |

***Сервис 2***

Разработать АПИ для актуализации места с товарами

При вызове метода сервис обновляет информацию, запросив ее у устройства о местоположении, либо же пользователь введет ее сам, далее информация попадет в регистр и будет подключаться к данным о товарах выбранного или определенного города.

GET back/products

**Пример запроса**

SELECT city\_id, product\_id, count, price

FROM margin

WHERE city\_id= “<city\_id>” and count >0

**Пример ответа**

city\_id | product\_id | count | price

**Query Params**

[

{

City\_id: number

Product\_id: number

Count: string

Price: string

}

]

|  |  |
| --- | --- |
| Cityd | City\_id |
| Productid | Product\_id |
| Count | Count |
| Price | price |

***Сервис 3***

Разработать АПИ для отображения продуктового контента по товарам

GET back/PIM/content

Алгоритм работы

При вызове Query Params, передается айдишник места, продукта, далее формируется запрос на sql для поиска картинки, описания товара, цены

**Пример запроса**

{

SELECT city\_id, product\_id, count, price, image, description

FROM margin

WHERE city\_id= “<city\_id>” and count >0

}

**Пример ответа**

**Пример ответа**

city\_id | product\_id | count | price| image|description

**Query Params**

[

{

City\_id: number

Product\_id: number

Count: string

Price: string

Image: string

Description: string

}

]

|  |  |
| --- | --- |
| Cityd | City\_id |
| Productid | Product\_id |
| Count | Count |
| Price | price |
| Image | Image |
| Description | Description |