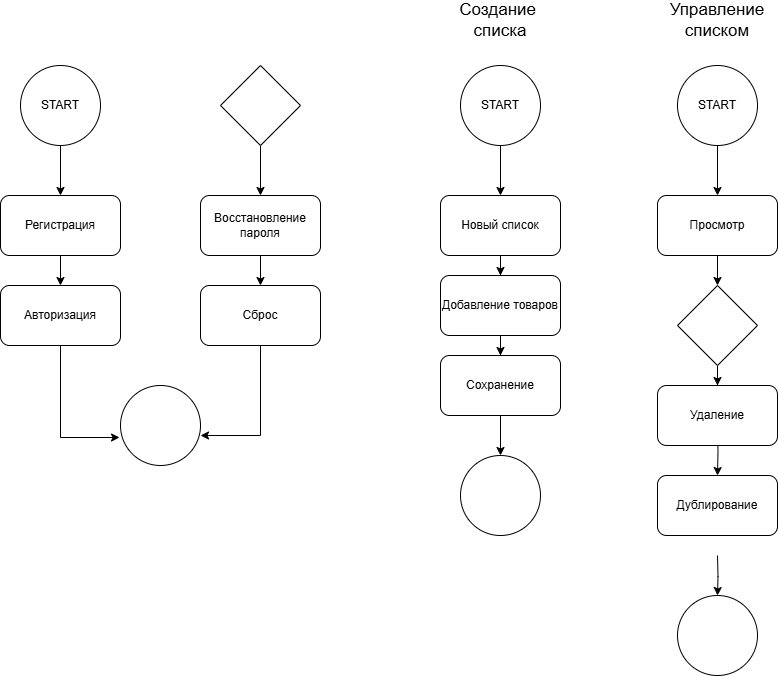
Кулагиной Валерии

***Задание 1***



***Задание 2***

Основные функции приложения «Список покупок» для магазинов включают:

1. Добавление, удаление и редактирование товаров (название, количество, категория).
2. Маркировка товаров как купленных или не купленных.
3. Синхронизация данных между устройствами.
4. Автоматическое сохранение списков в облаке.
5. Доступ к спискам с любого устройства после авторизации.
6. Категоризация товаров.
7. Группировка товаров по категориям (например, «Молочные продукты», «Овощи», «Бытовая химия»).
8. Возможность создания пользовательских категорий.
9. Напоминания и уведомления.
10. Установка времени и даты для напоминаний о походе в магазин.
11. Push-уведомления с актуальным списком покупок.
12. Совместное использование списков.
13. Возможность делиться списками с другими пользователями (семья, друзья).
14. Редактирование списка несколькими участниками в режиме реального времени.
15. Резервное копирование и восстановление данных.
16. Автоматическое резервное копирование в облако.
17. Голосовой ввод товаров.
18. Добавление товаров через голосовые команды.
19. Интуитивный интерфейс.
20. Простой дизайн с возможностью настройки тем (светлая/тёмная тема).
21. Быстрый поиск по спискам и товарам.

***Задание 3***

**API методы**

**Добавление списка:**

POST /api/lists

{

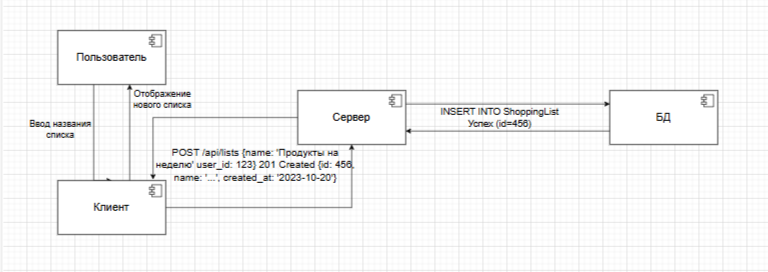
"name": "Продукты на неделю",

"user\_id": 123

}

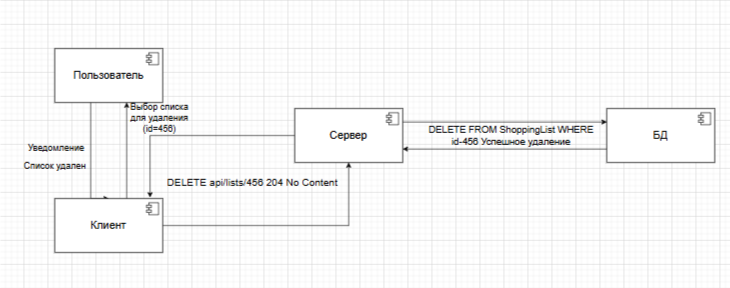
**Ответ:**

{ "id": 456, "name": "Продукты на неделю", "created\_at": "2023-10-20" }



**Удаление списка:**

DELETE /api/lists/{listId}



**Добавление товара:**

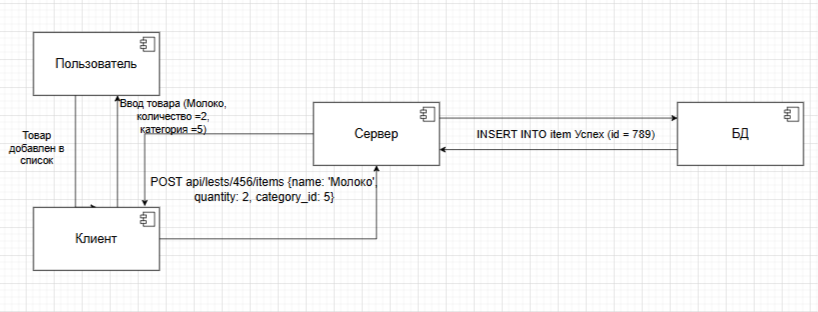
POST /api/lists/{listId}/items

{

"name": "Молоко",

"quantity": 2,

"category\_id": 5

} 

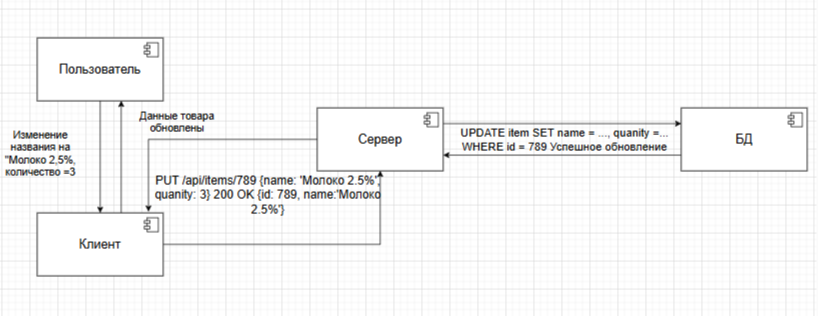
**Редактирование товара:**

PUT /api/items/{itemId}

{

"name": "Молоко 2,5%",

"quantity": 3

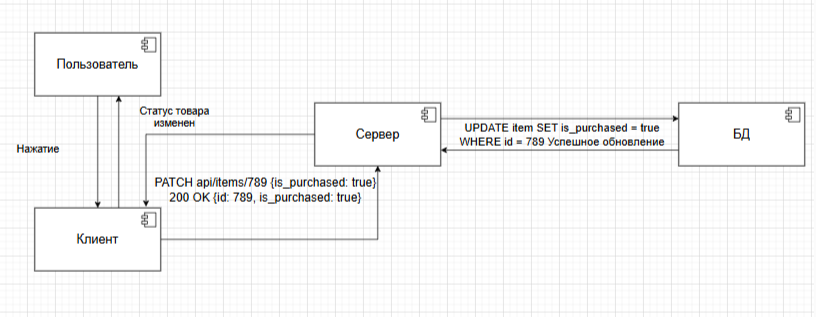
} 

**Покупка/отмена покупки:**

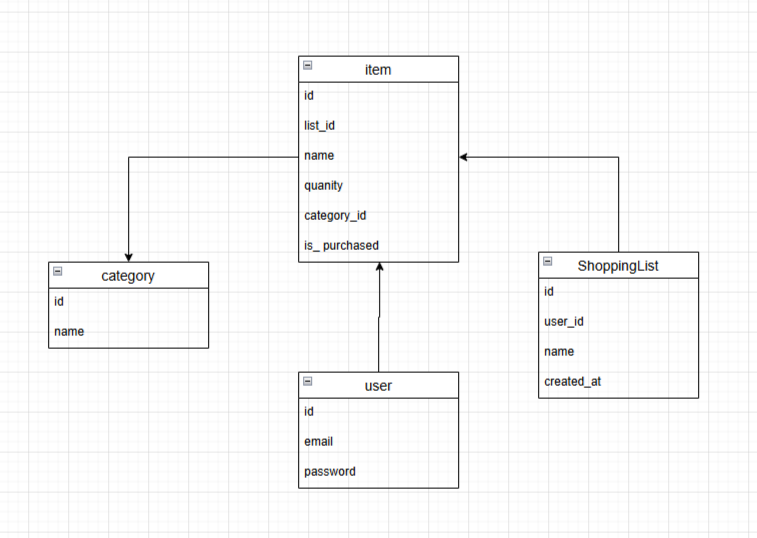
PATCH /api/items/{itemId}

{

"is\_purchased": true

} 

*ER-диаграмма сущностей:*



***Задание 4***



***Задание 5***

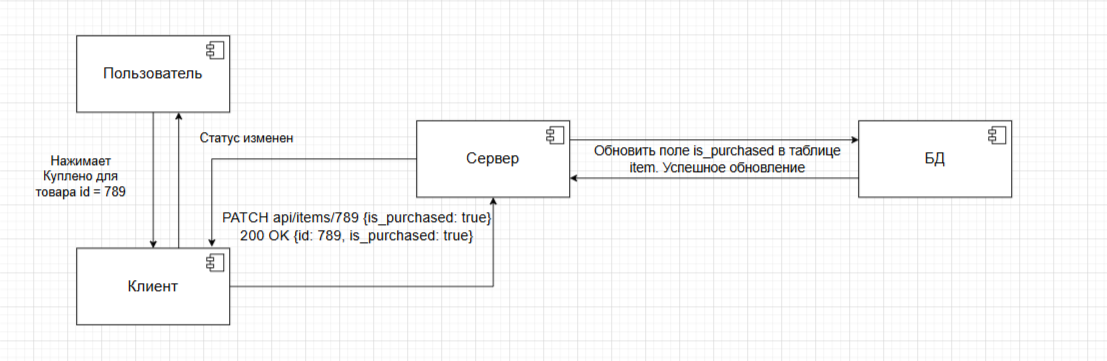
**1. Текстовое описание функции**

Функция позволяет пользователю отмечать товары в списке покупок как купленные или не купленные.

Сценарии использования:

* Пользователь отмечает товар как купленный после его покупки.
* Пользователь отменяет отметку, если товар не был куплен.

**2. UML-диаграмма последовательности**



**3. API-метод**

**Пример запроса**:

PATCH /api/items/789

{

"is\_purchased": true

}

**Ответы:**

**Успех (200 OK):**

{

"id": 789,

"is\_purchased": true,

"updated\_at": "2023-10-20T15:30:00Z"

}

**Ошибки**:

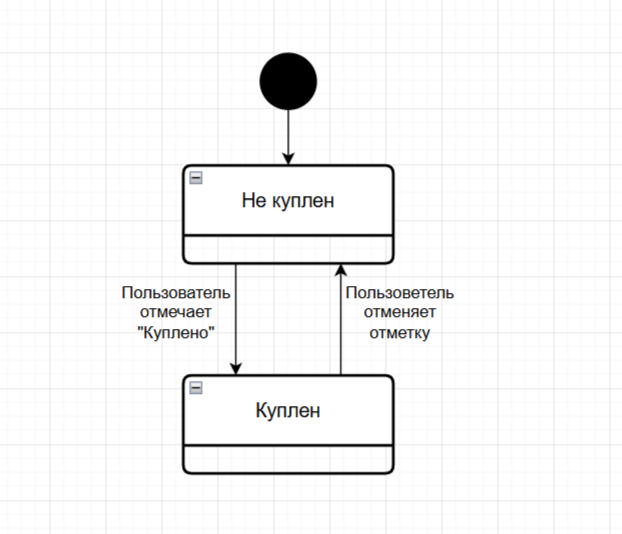
404 Not Found — товар не существует.

400 Bad Request — некорректный запрос

#### **4. Хранение информации о покупках**

**Структура таблицы Item в БД:**

| **Поле** | **Тип** | **Описание** |
| --- | --- | --- |
| id | INT (PK) | Уникальный ID товара. |
| list\_id | INT (FK) | ID списка покупок. |
| name | VARCHAR(255) | Название товара. |
| is\_purchased | BOOLEAN | Статус покупки (true/false). |
| updated\_at | TIMESTAMP | Время последнего изменения. |



***Задание 6***

Основные проблемы, с которыми сталкиваются разработчики при создании приложения:

1. Обеспечение синхронизации данных между устройствами.
2. Разрешение конфликтов при одновременном редактировании списков.
3. Обеспечение безопасности персональных данных (пароли, списки покупок).
4. Предотвращение утечек информации и защита от DDoS-атак.
5. Оптимизация запросов к базе данных при большом количестве пользователей.
6. Масштабирование архитектуры приложения.
7. Интеграция с API сторонних сервисов (если планируется интеграция с магазинами).
8. Обработка ошибок и ограничений сторонних систем.
9. Создание интуитивно понятного интерфейса.
10. Адаптация приложения для работы на мобильных устройствах и компьютерах.
11. Покрытие всех возможных сценариев использования, включая крайние случаи (например, потерю соединения).
12. Автоматизация тестирования.

* Разработка приложения требует ответов на ряд вопросов:

1. Кто будет использовать приложение? Физические лица, семьи или бизнес?
2. Нужна ли функция совместного редактирования списков?
3. Ожидается ли рост пользователей в ближайшем будущем?
4. Какие требования к производительности (время отклика, нагрузка)?
5. Планируется ли интеграция с магазинами или сервисами доставки в будущем?
6. Требуется ли соответствие GDPR или другим стандартам?
7. Нужна ли двухфакторная аутентификация?
8. Нужно ли разрабатывать нативные приложения для iOS и Android или достаточно веб-приложения?
9. Как приложение будет зарабатывать деньги (подписка, реклама, платные функции)?
10. Как часто нужно делать резервное копирование данных?
11. Сколько времени хранить данные удалённых списков?
12. Планируется ли поддержка нескольких языков и региональных настроек?
13. Есть ли предпочтения по языкам программирования, фреймворкам и базам данных?

***Задание 7***

**1. Общая стоимость книг для каждого автора (сортировка по убыванию):**

SELECT

a.AuthorName,

SUM(b.Price) AS TotalPrice

FROM

Authors a

LEFT JOIN

Books b ON a.id = b.AuthorId

GROUP BY

a.AuthorName

ORDER BY

TotalPrice DESC;

**Результат:**

AuthorName | TotalPrice

Дж. Роулинг | 2900

Л.Н.Толстой | 1600

М.А.Булгаков | 2000

А.С.Пушкин | 1500

П.Коэльо | 700

Д.Браун | NULL

**2. Авторы, у которых общая стоимость книг превышает 1500:**

SELECT

a.AuthorName,

SUM(b.Price) AS TotalPrice

FROM

Authors a

LEFT JOIN

Books b ON a.id = b.AuthorId

GROUP BY

a.AuthorName

HAVING

SUM(b.Price) > 1500;

**Результат:**

AuthorName | TotalPrice

Дж. Роулинг | 2900

Л.Н.Толстой | 1600

М.А.Булгаков | 2000

**3. Авторы с количеством книг:**

SELECT

a.AuthorName,

COUNT(b.id) AS BookCount

FROM

Authors a

LEFT JOIN

Books b ON a.id = b.AuthorId

GROUP BY

a.AuthorName;

**Результат:**

AuthorName | BookCount

А.С.Пушкин | 2

М.А.Булгаков | 1

Л.Н.Толстой | 2

П.Коэльо | 1

Дж. Роулинг | 1

Д.Браун | 0

**4. Авторы без книг:**

SELECT

a.AuthorName

FROM

Authors a

LEFT JOIN

Books b ON a.id = b.AuthorId

WHERE

b.id IS NULL;

**Результат:**

AuthorName

Д.Браун