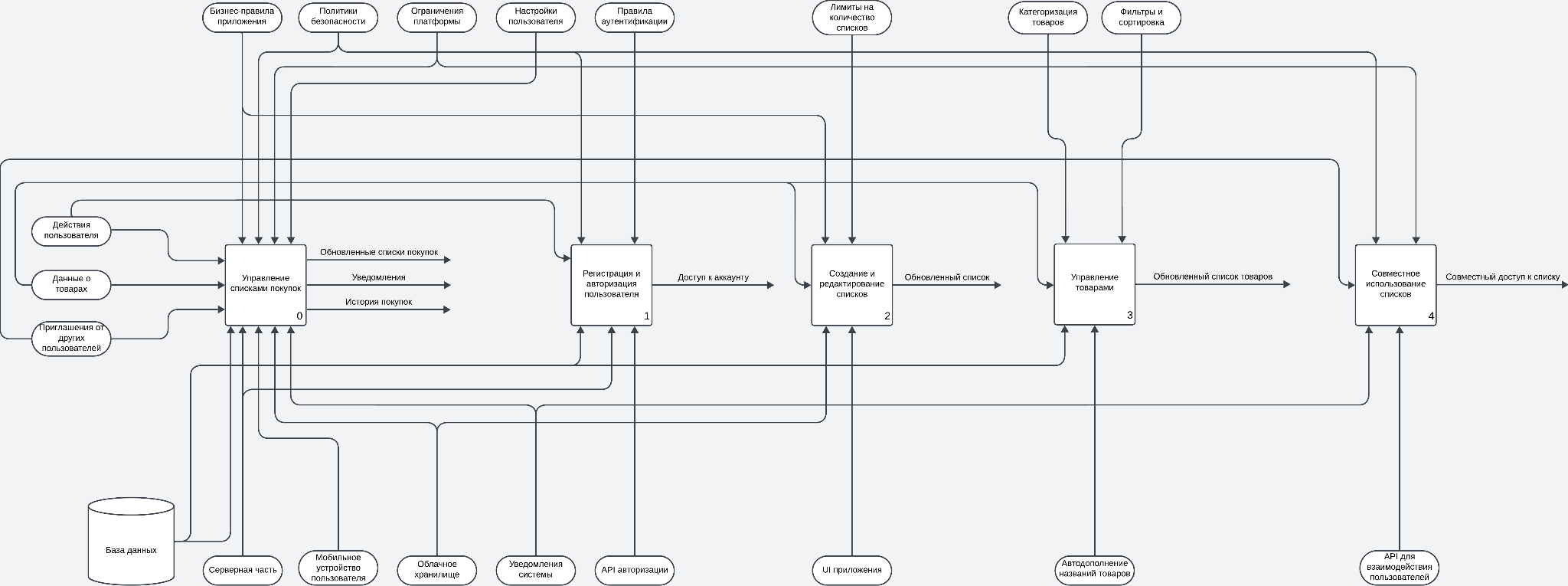
Задание 1:



Бизнес-логика мобильного приложения с нотацией IDEF0

Задание 2:

Основные функции приложения

1 Создание и редактирование списка покупок – добавление, удаление и изменение товаров.  
 2 Отмечание купленных товаров – возможность отмечать товары как купленные.  
 3 Сохранение и управление списками – создание нескольких списков, удаление, архивирование.  
 4 Совместный доступ к списку – приглашение других пользователей для совместного использования.  
 5 Автодополнение и категории товаров – быстрое добавление популярных товаров.  
 6 Работа в офлайн-режиме – доступ и редактирование списка без интернета.  
 7 Уведомления и напоминания – напоминания о необходимости совершить покупки.  
 8 Синхронизация между устройствами – доступ к списку с нескольких устройств.  
 9 История покупок – просмотр ранее купленных товаров.  
 10 Настройки пользовательских предпочтений – включение/выключение уведомлений, сортировка списков.

Задание 3:



ER-диаграмма

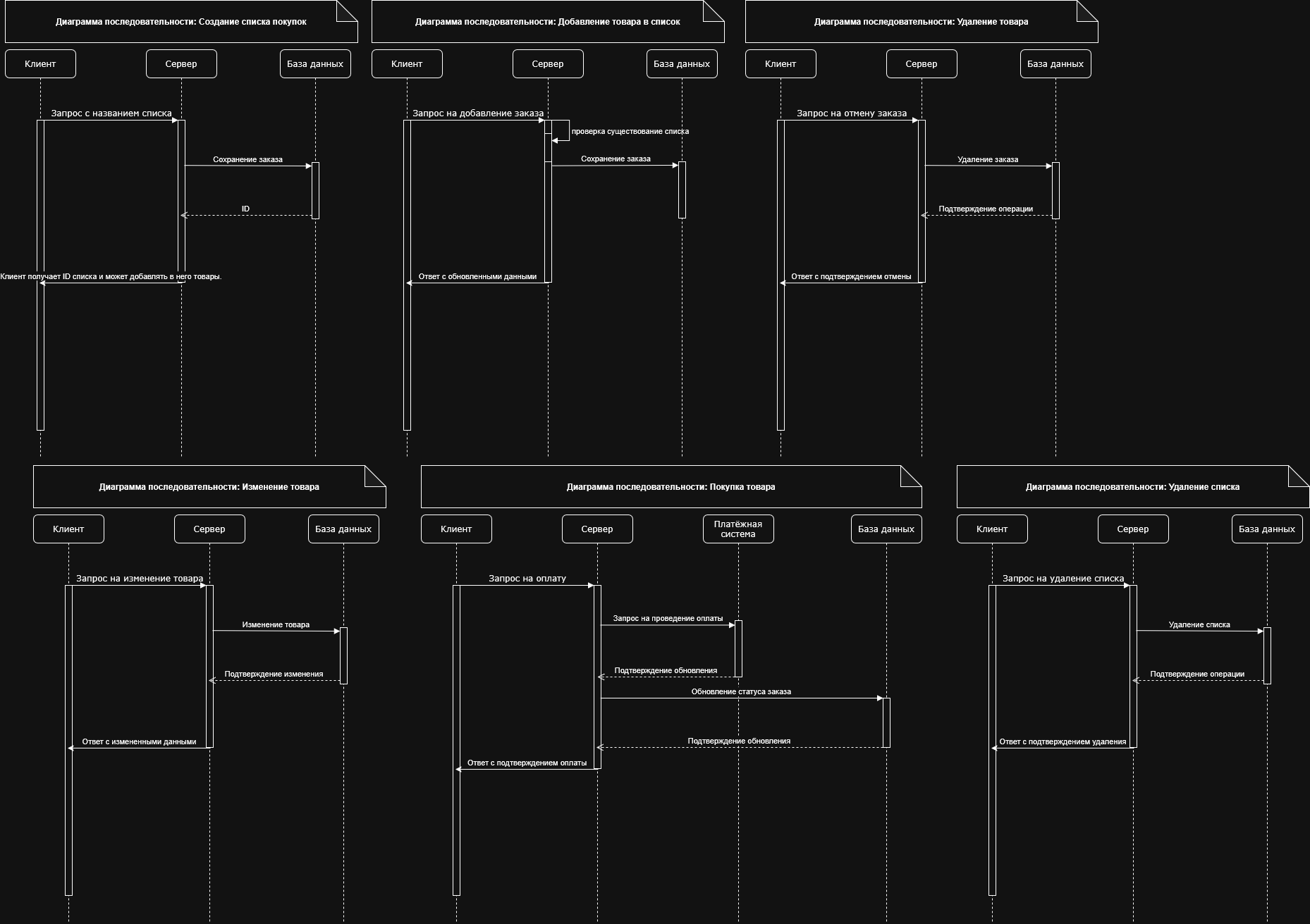


Диаграмма последовательностей

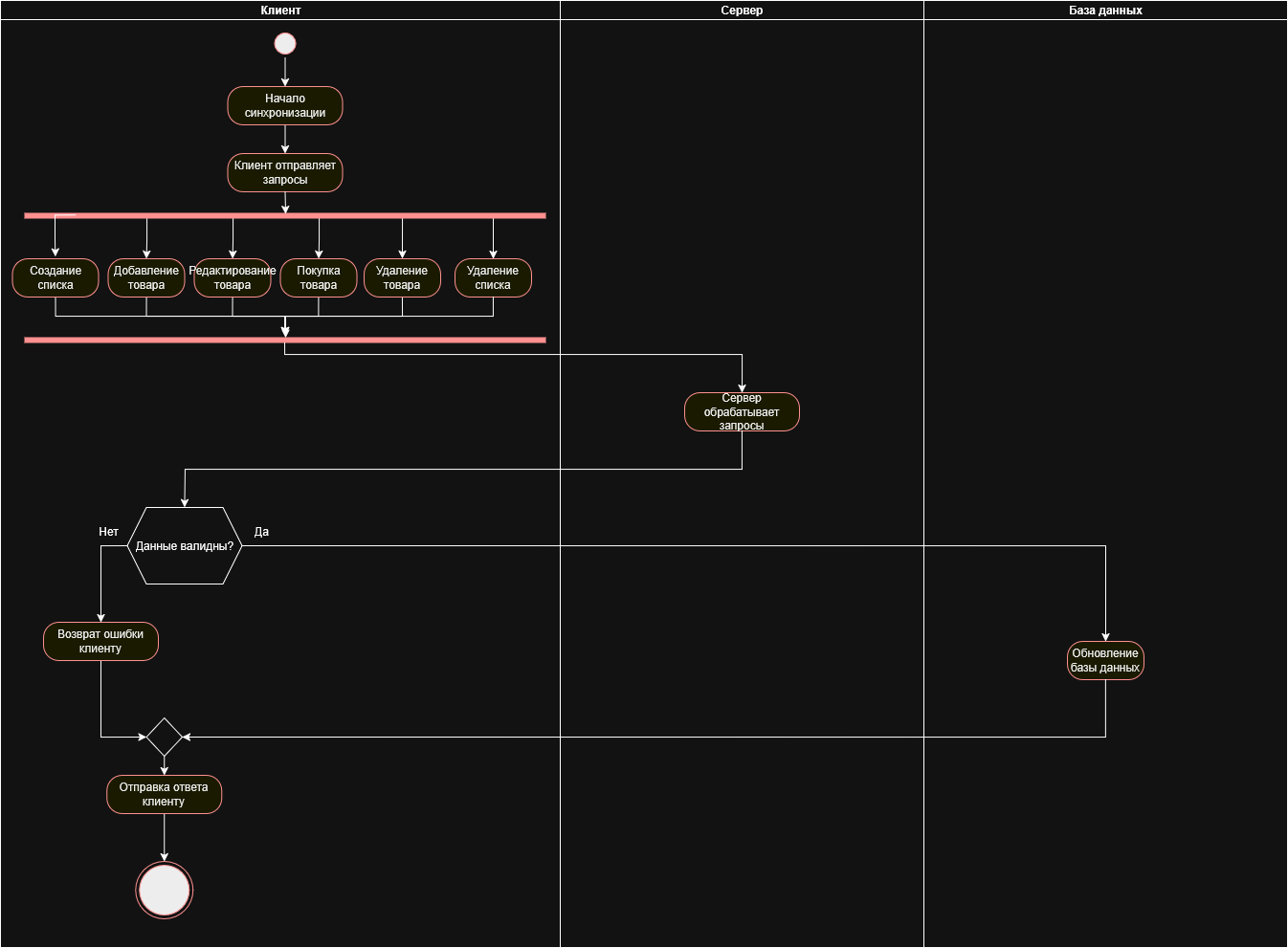


Диаграмма активностей

### API методы (REST API)

Пример API для взаимодействия клиента и сервера:

* POST /list – создать список
* GET /list/{id} – получить список
* PUT /list/{id} – обновить название списка
* DELETE /list/{id} – удалить список
* POST /list/{id}/items – добавить товар
* PUT /list/{id}/items/{item\_id} – обновить товар
* PATCH /list/{id}/items/{item\_id} – отметить товар купленным
* DELETE /list/{id}/items/{item\_id} – удалить товар

Задание 4:

<https://www.figma.com/design/mgVhHBqfjAyVooLEF8QvDT/Untitled?node-id=0-1&p=f&t=LhQ11pbmlsPH0hDt-0>

Задание 5:

#### 1. Получение списка покупок

Метод: GET /shopping-lists/{list\_id}/items  
Описание: Возвращает актуальный список товаров для определенного списка.  
Параметры запроса:

* list\_id (UUID) – идентификатор списка.  
  Ответ: JSON с массивом товаров.

#### 2. Отметка товара как купленного

Метод: PATCH /shopping-lists/{list\_id}/items/{item\_id}  
Описание: Обновляет статус товара на «куплено».  
Тело запроса:

{

"status": "purchased",

"purchase\_date": "2025-02-13T12:00:00Z"

}

Ответ: 200 OK + обновленные данные товара.

#### 3. Отмена покупки

Метод: PATCH /shopping-lists/{list\_id}/items/{item\_id}  
Описание: Сбрасывает статус товара.  
Тело запроса:

{

"status": "pending"

}

Ответ: 200 OK + обновленные данные.

#### 4. Получение истории покупок

Метод: GET /users/{user\_id}/purchase-history  
Описание: Возвращает список всех купленных товаров с датами.  
Ответ: JSON с историей покупок.

Задание 6:

#### 1. Сложности синхронизации данных

* Обеспечение корректной синхронизации списка покупок между клиентом и сервером при потере соединения.
* Конфликты данных (например, если несколько пользователей редактируют один и тот же список).
* Обновление данных в реальном времени без перегрузки сети.

### Вопросы заказчику:

* Должна ли быть возможность совместного редактирования списка несколькими пользователями?
* Нужно ли добавлять категории товаров или достаточно обычного списка?
* Требуется ли поддержка нескольких языков?
* Как часто должна происходить синхронизация данных? В реальном времени или по запросу?
* Будут ли интеграции с другими сервисами (например, уведомления о скидках, прогнозирование покупок)?
* Нужно ли поддерживать голосовое управление или сканирование штрих-кодов?
* Нужна ли реклама или подписочная модель?

Задание 7:

1. Общая стоимость книг для каждого автора и сортировка по убыванию:

SELECT a.AuthorName, SUM(b.Price) AS TotalPrice

FROM Authors a

JOIN Books b ON a.Id = b.AuthorId

GROUP BY a.AuthorName

ORDER BY TotalPrice DESC;

2. Стоимость книг автора превышает 1500:

SELECT a.AuthorName, SUM(b.Price) AS TotalPrice

FROM Authors a

JOIN Books b ON a.Id = b.AuthorId

GROUP BY a.AuthorName

HAVING SUM(b.Price) > 1500;

3. Вывести авторов с количеством их 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;

4. Получить авторов без книг:

SELECT a.AuthorName

FROM Authors a

LEFT JOIN Books b ON a.Id = b.AuthorId

WHERE b.Id IS NULL;