**Задание 1. Дано: Необходимо описать: Мобильное клиент-серверное приложение «Список покупок в магазинах» (Примечание: приложение без интеграции с магазинами). 1. Бизнес-логику мобильного приложения (использовать любую удобную нотацию моделирования бизнес-процессов).**

Вот основные бизнес-процессы, которые можно включить в модель:

1. Регистрация и авторизация пользователя
   * Пользователь регистрируется, указывая email и пароль.
   * Авторизуется в системе, используя учетные данные.
   * При успешной авторизации получает доступ к функционалу.
2. Создание и управление списками покупок
   * Пользователь создает новый список покупок.
   * Добавляет товары в список, указывая название, количество, категорию.
   * Редактирует или удаляет товары из списка.
3. Отметка товаров как купленных
   * Пользователь отмечает купленные товары, они перемещаются в категорию «Куплено».
4. Совместное использование списка
   * Пользователь может пригласить другого пользователя для совместного использования списка.
   * Другой пользователь получает доступ к списку и может его редактировать.
5. Синхронизация с сервером
   * Все данные (списки, товары, статусы) синхронизируются с сервером.
   * Доступ к спискам возможен с разных устройств.
6. Выход из системы
   * Пользователь может выйти из аккаунта, данные остаются сохраненными.

**Задание 2 Основные функции данного приложения (список должен быть составлен в порядке убывания важности функций для пользователя)**

**Создание и управление списками покупок** – добавление, редактирование и удаление списков.

**Добавление товаров в список** – возможность указывать название, количество, категорию.

**Отметка товаров как купленных** – простая галочка или перемещение в «Купленные».

**Сохранение и синхронизация данных** – чтобы списки не терялись и были доступны на разных устройствах.

**Совместное использование списка** – возможность делиться списком с другим пользователем.

**Регистрация и авторизация** – вход в аккаунт для сохранения данных.

**Автоматические предложения товаров** – подсказки на основе прошлых покупок.

**Категоризация товаров** – возможность сортировать товары по категориям (овощи, напитки и т. д.).

**История покупок** – просмотр предыдущих списков и купленных товаров.

**Офлайн-режим** – доступ к спискам без подключения к интернету.

**Напоминания и уведомления** – оповещения о незавершенных покупках или предложениях.

**Импорт/экспорт списка** – возможность сохранить список в текстовом формате или поделиться им в мессенджерах.

**Темная/светлая тема интерфейса** – настройка внешнего вида приложения.

**задание 3 Процесс синхронизации данных между клиентом и сервером (добавление и удаление списка, наполнение и редактирование списка, покупка/«откупка продукта» и т.д.). Представить все в диаграммах UML, API методах и других представлениях, также составить ER-диаграмму сущностей**

{

"POST /lists": {

"description": "Добавление списка",

"request": {

"name": "Grocery Shopping"

},

"response": {

"list\_id": "12345",

"name": "Grocery Shopping",

"created\_at": "2025-02-13T10:00:00Z"

}

},

"DELETE /lists/{list\_id}": {

"description": "Удаление списка",

"request": {},

"response": {

"message": "List deleted successfully"

}

},

"POST /lists/{list\_id}/items": {

"description": "Добавление товара в список",

"request": {

"name": "Milk",

"quantity": 2

},

"response": {

"item\_id": "67890",

"name": "Milk",

"quantity": 2,

"purchased": false

}

},

"PUT /lists/{list\_id}/items/{item\_id}": {

"description": "Редактирование товара",

"request": {

"name": "Almond Milk",

"quantity": 1,

"purchased": true

},

"response": {

"item\_id": "67890",

"name": "Almond Milk",

"quantity": 1,

"purchased": true

}

}

}

**Задание 4 Подготовить прототип одного из экранов мобильного приложения и описать пользовательский интерфейс для данного экрана (например, покупка товара)**

<https://www.figma.com/design/IhRPs92GKAP3qeVBXyd7BW/Untitled?node-id=0-1&p=f&t=04NWkut2o53nWGWp-0>

**Задание 5 Подготовить подробное описание функции покупки товара, которую можно было бы использовать в качестве постановки задачи для разработки (помимо текстового описания использовать UML диаграммы, указать используемые API методы, передаваемые и получаемые параметры, описать процесс хранения информации о покупках пользователя).**

{

"GET /orders/{order\_id}": {

"description": "Получение информации о заказе",

"request": {},

"response": {

"order\_id": "12345",

"items": [

{

"product\_id": "67890",

"name": "Продукт 1",

"quantity": 2,

"price": 500

}

],

"total\_price": 1000

}

},

"PUT /orders/{order\_id}": {

"description": "Обновление информации о заказе",

"request": {

"items": [

{

"product\_id": "67890",

"quantity": 3

}

]

},

"response": {

"message": "Order updated successfully"

}

}

}

**Задание 6 Основные, на ваш взгляд, сложности разработки такого приложения. Вопросы, возникшие при выполнении тестового задания, которые вы бы задали заказчику.**

### Основные сложности разработки приложения "Список покупок"

1. Синхронизация данных между клиентом и сервером
   * Обновление списка покупок в реальном времени.
   * Разрешение конфликтов, если несколько пользователей редактируют список одновременно.
2. Работа с оффлайн-режимом
   * Как хранить данные локально и синхронизировать их при подключении к сети.
   * Что делать, если данные в оффлайне изменились, а на сервере уже есть новые версии?
3. Аутентификация и авторизация пользователей
   * Нужно ли регистрация через email/пароль или OAuth (Google, VK и т. д.)?
   * Как разделять списки между пользователями (индивидуальные или совместные)?
4. Оптимизация работы с БД
   * Как обрабатывать большие объемы данных (например, тысячи продуктов и списков)?
   * Какие индексы и связи в базе использовать для быстрого поиска?
5. UI/UX удобство для пользователей
   * Какие действия должны быть доступны в одно касание?
   * Как сделать интуитивно понятный интерфейс для работы со списками?
6. Уведомления
   * Нужно ли пользователям напоминать о покупках?
   * Какие события должны вызывать пуш-уведомления?

### Вопросы к заказчику

1. Будет ли приложение поддерживать совместное использование списка покупок?
2. Какие роли пользователей предусмотрены (например, администратор списка, участник)?
3. Должны ли товары автоматически добавляться в "избранное" или предлагаться на основе истории покупок?
4. Нужна ли интеграция с внешними сервисами (например, поиск цен в магазинах, рекомендации)?
5. Какие платформы приоритетные: только мобильное приложение или ещё и веб-версия?
6. Какой уровень защиты данных требуется (например, шифрование списков, защита от утечки данных)?
7. Нужно ли хранить историю покупок для дальнейшего анализа и отчётов?
8. Какие требования по скорости отклика приложения?
9. Нужна ли поддержка голосового ввода для добавления товаров в список?

**Задание 7!**

### Общая стоимость книг для каждого автора (отсортировано 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;

### Авторы, у которых стоимость книг превышает 150

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;

### 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;

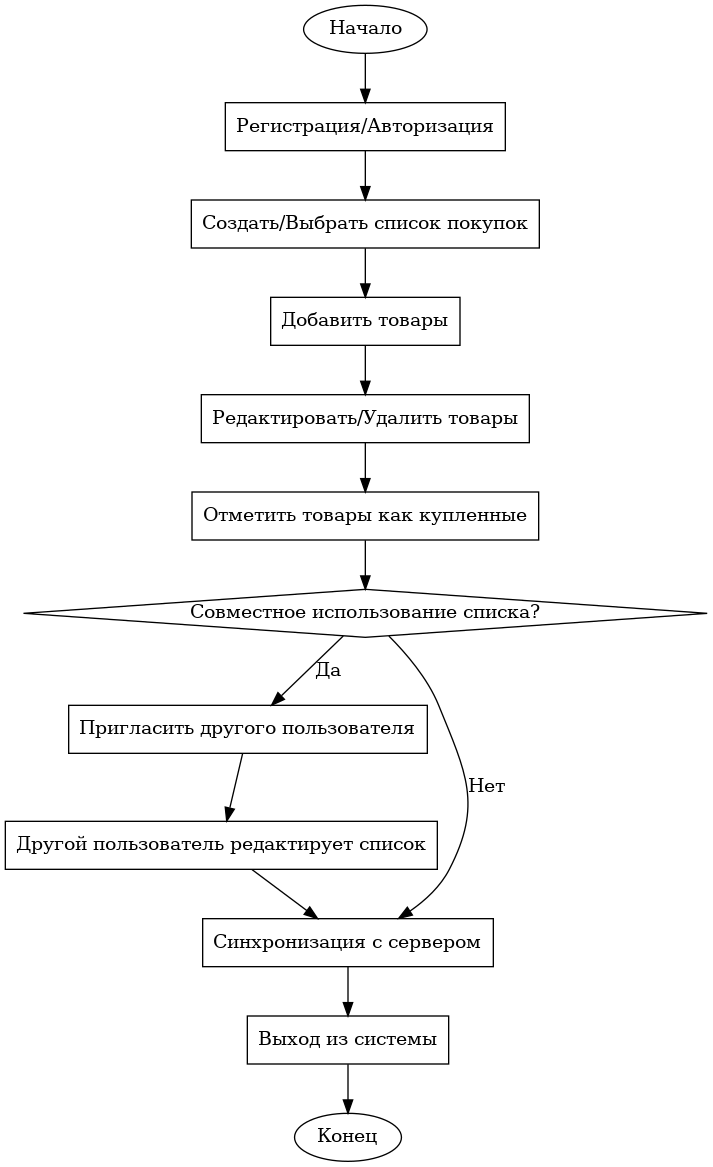
### Авторы без книг

SELECT a.AuthorName

FROM Authors a

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

WHERE b.Id IS NULL;

****