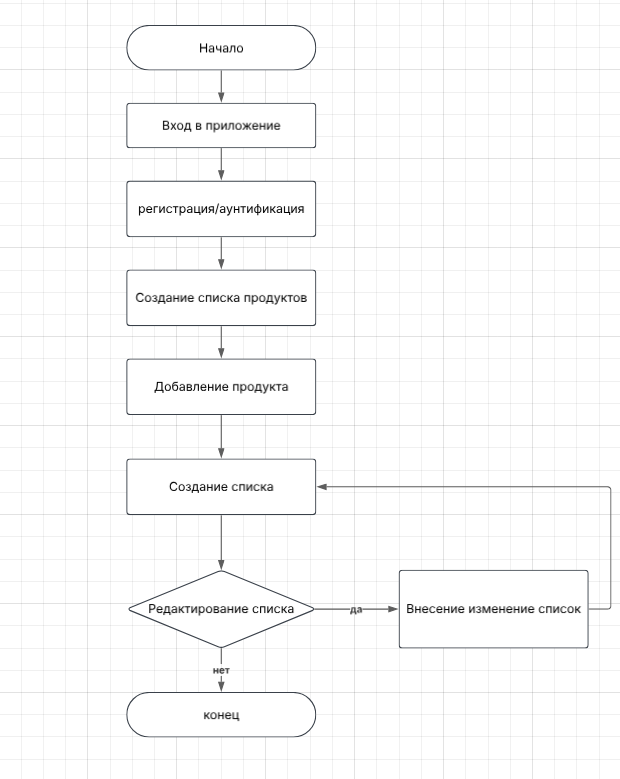
Тестовое задание по аналитике

Дано: Мобильное клиент-серверное приложение «Список покупок в магазине» (Приложение без интеграции с магазинами).

**1. Бизнес-логика мобильного приложения**

**2. Основные функции данного приложения**

Бизнес-процесс: Управление списком покупок

1. Начало процесса

Пользователь запускает приложение.

2. Создание нового списка покупок

- Пользователь выбирает опцию "Новый список".

- Вводит название для списка.

- Сохраняет список.

3. Добавление товаров в список

- Пользователь выбирает список покупок.

- Нажимает на опцию "Добавить товар".

- Вводит название, количество и цену товара.

- Сохраняет товар.

4. Редактирование товара

- Пользователь выбирает товар из списка.

- Нажимает на опцию "Редактировать".

- Изменяет информацию о товаре.

- Сохраняет изменения.

5. Удаление товара из списка

- Пользователь выбирает товар.

- Нажимает на опцию "Удалить".

- Подтверждает удаление товара.

6. Просмотр списка покупок

- Пользователь открывает список и просматривает все добавленные товары.

- Видит общую стоимость.

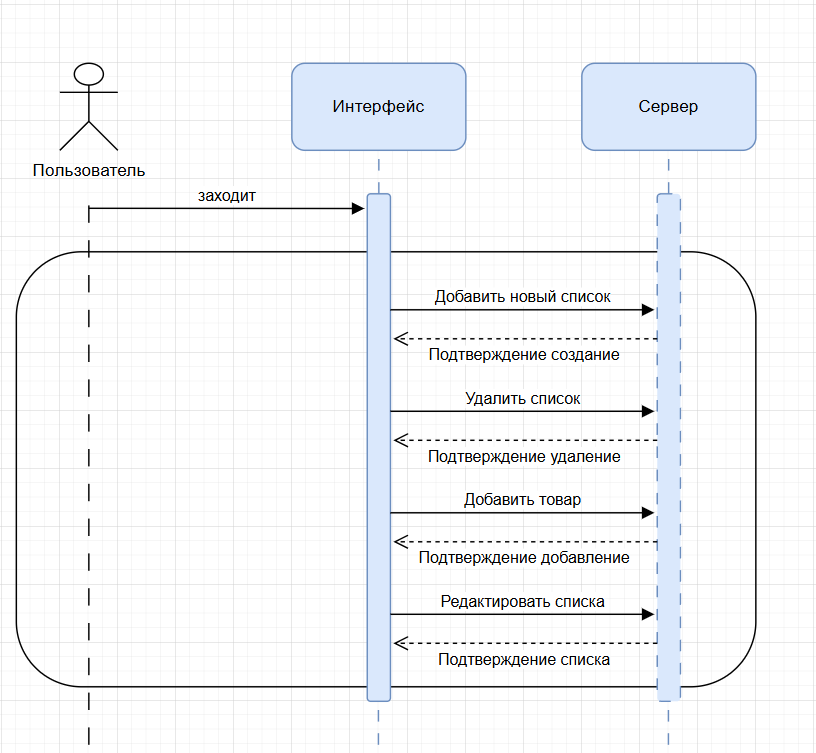
7. Синхронизация данных

- Приложение автоматически синхронизирует данные с сервером (по необходимости).

- Загружает/забивает данные на сервер и сохраняет изменения.

8. Завершение процесса

Пользователь закрывает приложение или выходит из списка покупок.

**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.**

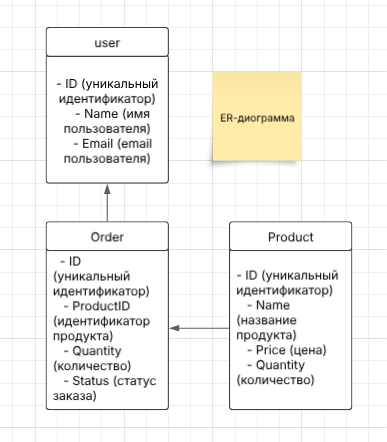
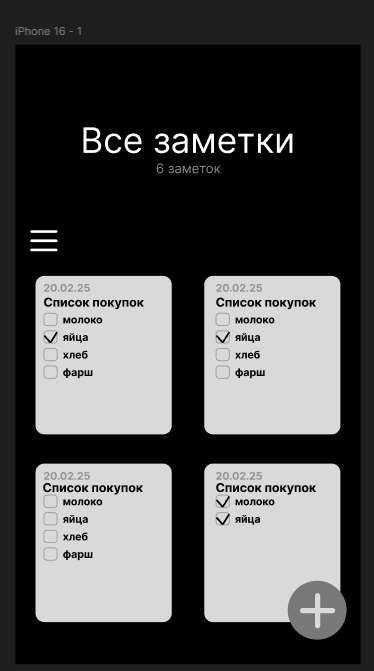
Рисунок 1 – ULM диаграмма

Рисунок 2 – ER-диаграмма

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

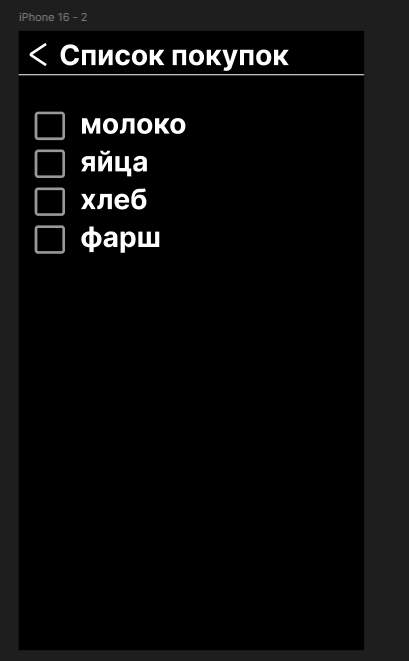
Рисунок 3 – пример главного экрана

Рисунок 4 – пример списка

Главная страница:

1. Навигационное меню:

* Сортировка списков
* Профиль
* Аналитика

2. Настройки

3. Добавление нового списка продуктов

4. Просмотр списка продуктов и переход на его страницу

Страница списка продуктов:

1. Кнопка возвращения на главную страницу

2. Название списка, возможность его изменения

3. Просмотр списка продуктов

4. Возможность отметки купленного товара с помощью чекбокса

5. Добавление новых продуктов в список

**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).**

Сценарий использования

1. Пользователь выбирает товар.

2. Пользователь задает количество товара.

3. Пользователь добавляет товар в корзину.

4. Пользователь переходит в корзину.

5. Пользователь выбирает способ доставки и оплаты.

6. Пользователь подтверждает покупку.

7. Система обрабатывает заказ и отправляет уведомление.

Используемые API методы

1. Получение информации о товаре

- Метод: GET /api/products/{productId}

- Параметры:

- productId - идентификатор товара.

- Ответ:

- { "name": "Товар", "description": "Описание", "price": 100 }

2. Добавление товара в корзину

- Метод: POST /api/cart

- Параметры:

- productId - идентификатор товара.

- quantity - количество товара.

- Ответ:

- { "message": "Товар добавлен в корзину" }

3. Просмотр корзины

- Метод: GET /api/cart

- Ответ:

- { "items": [{ "productId": "1", "quantity": 2, "price": 200 }], "total": 200 }

4. Оформление заказа

- Метод: POST /api/orders

- Параметры:

- paymentMethod - метод оплаты.

- deliveryMethod - метод доставки.

- Ответ:

- { "orderId": "12345", "message": "Заказ успешно оформлен" }

5. Обработка платежа

- Метод: POST /api/payments

- Параметры:

- paymentDetails - детали платежа.

- Ответ:

- { "status": "success", "transactionId": "abc123**" }**

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

1. Каковы основные целевые платформы для приложения (iOS, Android, веб)?

2. Какие методы оплаты вы планируете использовать, и есть ли предпочтения по интеграции с платежными системами?

3. Есть ли необходимость в адаптации приложения для различных языков и регионов?

4. Как будет организована система управления товарами (например, добавление товаров, редактирование, удаление)?

5. Будет ли специальная функциональность для администраторов, чтобы управлять заказами и пользователями?

6. Каковы требования по безопасности данных и конфиденциальности (например, соответствие GDPR)?

7. Какова целевая аудитория приложения, и какие ключевые функции для них наиболее важны?

8. Какова предполагаемая нагрузка на систему и сколько пользователей вы планируете в первый год?

9. Есть ли необходимость в интеграции с внешними сервисами (например, для логистики или аналитики)?

10. Как будет организован процесс обратной связи от пользователей и их поддержки?

**7. Есть таблицы Books и Authors, где Authorld табл. Books равно Id табл. Authors.**

**Необходимо написать SQL-запрос, чтобы найти:**

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

sql

Copy

SELECT

a.Name AS AuthorName,

SUM(b.Price) AS TotalPrice

FROM Authors a

JOIN Books b ON a.Id = b.AuthorId

GROUP BY a.Id, a.Name

ORDER BY TotalPrice DESC;

2. Авторы с общей стоимостью книг > 1500

sql

Copy

SELECT

a.Name AS AuthorName,

SUM(b.Price) AS TotalPrice

FROM Authors a

JOIN Books b ON a.Id = b.AuthorId

GROUP BY a.Id, a.Name

HAVING SUM(b.Price) > 1500

ORDER BY TotalPrice DESC;

3. Авторы с количеством книг (предположим, что требовалось вывести количество книг)

sql

Copy

SELECT

a.Name AS AuthorName,

COUNT(b.Id) AS BookCount

FROM Authors a

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

GROUP BY a.Id, a.Name

ORDER BY BookCount DESC;**Ссылки:**

Диограммы: https://lucid.app/lucidchart/c833039b-2b0d-4f75-8948-36e94f4dd44f/edit?viewport\_loc=-766%2C-31%2C2474%2C1450%2C0\_0&invitationId=inv\_a36bdc02-40c0-4ed4-ba66-696766431885

Макет страницы: https://www.figma.com/design/ciyoFZRPhJ0BsIcEoIezkU/Untitled?m=auto&t=Y8uvzDxsoW7OH0Y3-6