

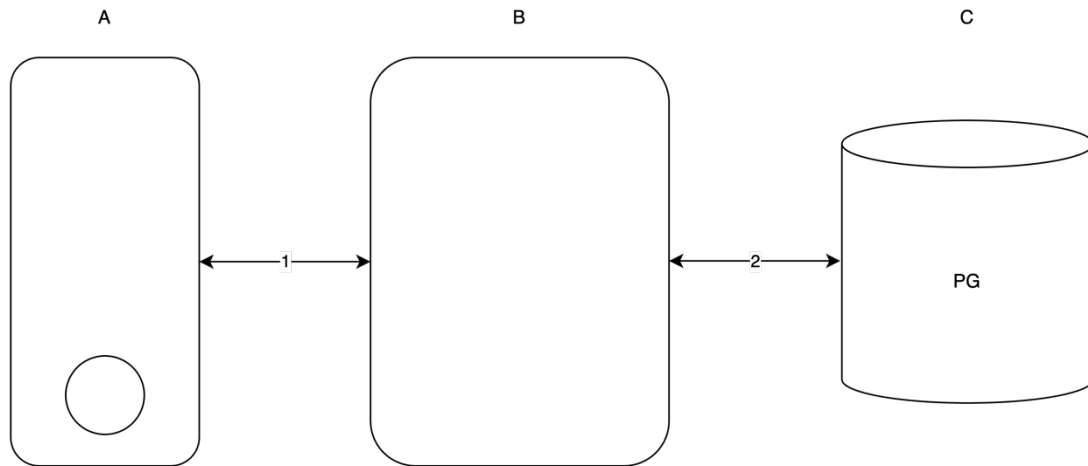
1. Архитектура

Фронтенд — это схематичный веб или мобильное приложение системы. Также фронтенд называют клиентом

Бэкенд — это внутренний сервер системы проекта. Сервер обращается к базе данных с полученными данными от клиента (фронтенда)

База данных — это хранилище данных, с которыми будет работать сервер

Пример



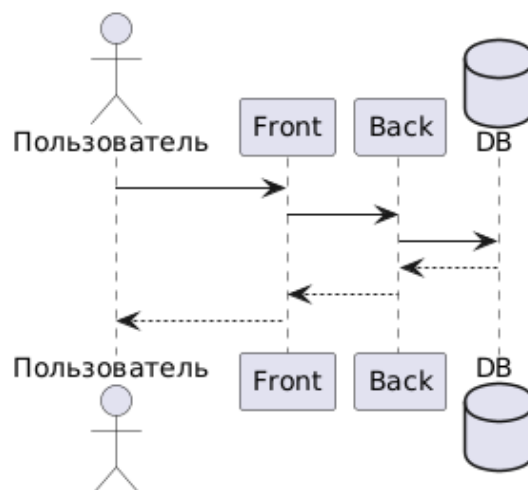
A - фронтенд, мобильное приложение
B - бэкенд/сервер
C - реляционная база данных, PG

1 - протокол взаимодействия http

2. Диаграмма последовательности

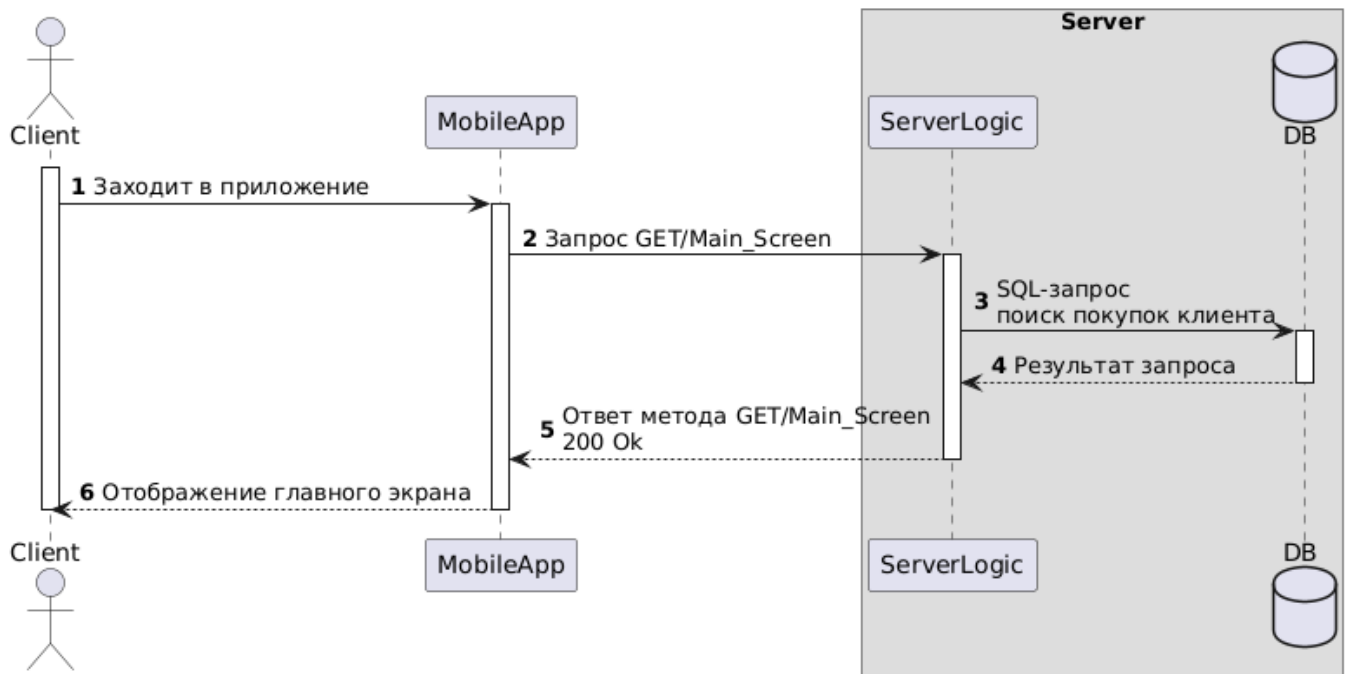
Диаграмма последовательности — это тип UML-диаграммы, который используется для моделирования взаимодействий между объектами в системе. Она отображает, в каком порядке и как взаимодействуют между собой различные объекты или компоненты во времени. Основное назначение — показать, как запросы и ответы передаются между элементами системы.

Шаблон



№	Описание
Номер шага	Описание того, что происходит на шаге

Пример



№	Описание
1	Вход пользователя на главный экран приложения или нажатие на кнопку "Подобрано для вас" из экрана "Хиты продаж"
2	Запрос на отображение главного экрана
3	SQL-запрос на получение покупок по данному клиенту
4	Возврат данных, содержащих все покупки клиента
5	Успешный ответ метода GET/Main_Screen – 200 OK. В ответе содержатся данные о покупках клиента
6	Отображение главного экрана приложения со списком покупок пользователя

3. Модель данных

Модель данных – это представление данных, атрибутивного состава сущностей, как сущности связаны друг с другом.

Шаблон

Родительская сущность	Атрибут	Описание

Пример

Объект User

Родительская сущность	Атрибут	Описание
User	----	Объект пользователя, который имеет атрибуты и ссылки на другие объекты
	FootSize	Размер ноги
	Gender	Пол
	WorkingAddress	Рабочий адрес. Ссылка на объект рабочий адрес.
	WorkingPhone	Рабочий телефон. Ссылка на объект рабочий телефон.

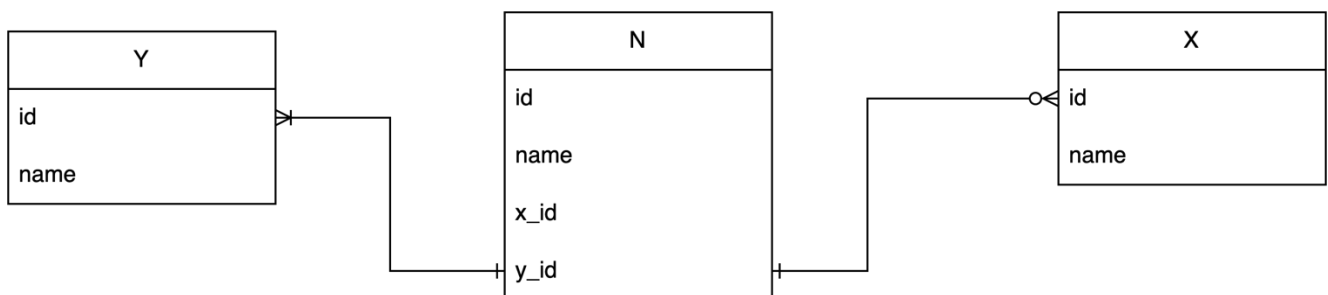
Объект WorkingAddress

Родительская сущность	Атрибут	Описание
WorkingAddress	----	Объект рабочего адреса.
	StreetName	Название улицы. Например, "Ленина".
	HomeIndex	Номер дома. Например, "14"
	PostIndex	Почтовый индекс. Например "644876"

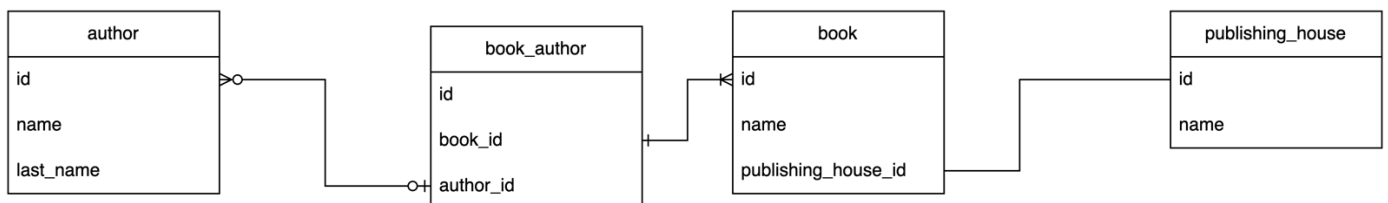
4. ERD-диаграмма

ERD-диаграмма — диаграмма где показано, как разные «сущности» (люди, объекты, концепции и так далее) связаны между собой внутри системы.

Шаблон



Пример



5. REST. Табличный вид

REST API подход использует HTTP-методы (GET, POST, PUT, DELETE и т.д.) для управления сущностями с уникальными URL.

Шаблон

Request

Название параметра	Тип данных	Находится в	Описание	Обязательность параметра

Response

Название параметра	Тип данных	Находится в	Описание	Обязательность параметра

Response code <200>

Пример

Get /v1/users/{userId}/purchases

Получение всех покупок определенного пользователя

Request

Название параметра	Тип данных	Находится в	Описание	Обязательность параметра
userId	int	path	Уникальный идентификатор пользователя	да

Response

Название параметра			Тип данных	Находится в	Описание	Обязательность параметра
userId			int	body	Уникальный идентификатор пользователя	да
purchases			array	body	Массив покупок	да
	purchase		object	body	Объект покупки	нет
		purchaseId	int	body	Уникальный идентификатор покупки	да
		date	string	body	Дата покупки	да
		shopId	int	body	Магазин, в котором совершили покупку	да

Response code <200>

6. Swagger

Swagger — это инструмент, который помогает аналитикам, разработчикам и тестировщикам создавать, документировать и проверять API.

Для чего нужен swagger системным аналитикам:

- Создание документации
- Тестирование API

Пример

pet Everything about your Pets [Find out more](#) ^

PUT **/pet** Update an existing pet

POST **/pet** Add a new pet to the store

GET **/pet/{petId}** Find pet by ID

POST **/pet/{petId}** Updates a pet in the store with form data

DELETE **/pet/{petId}** Deletes a pet

delete a pet

Parameters

Name	Description
api_key string (header)	<input type="text" value="api_key"/>
petId * required integer(\$int64) (path)	<input type="text" value="petId"/> Pet id to delete

Responses

Code	Description	Links
400	Invalid pet value	No links