

Фреймворк методик системного проектирования

Константин Валеев, май 2021



Привет!

Руководю Центром компетенций по системному анализу в «Ростелеком Информационных Технологиях».

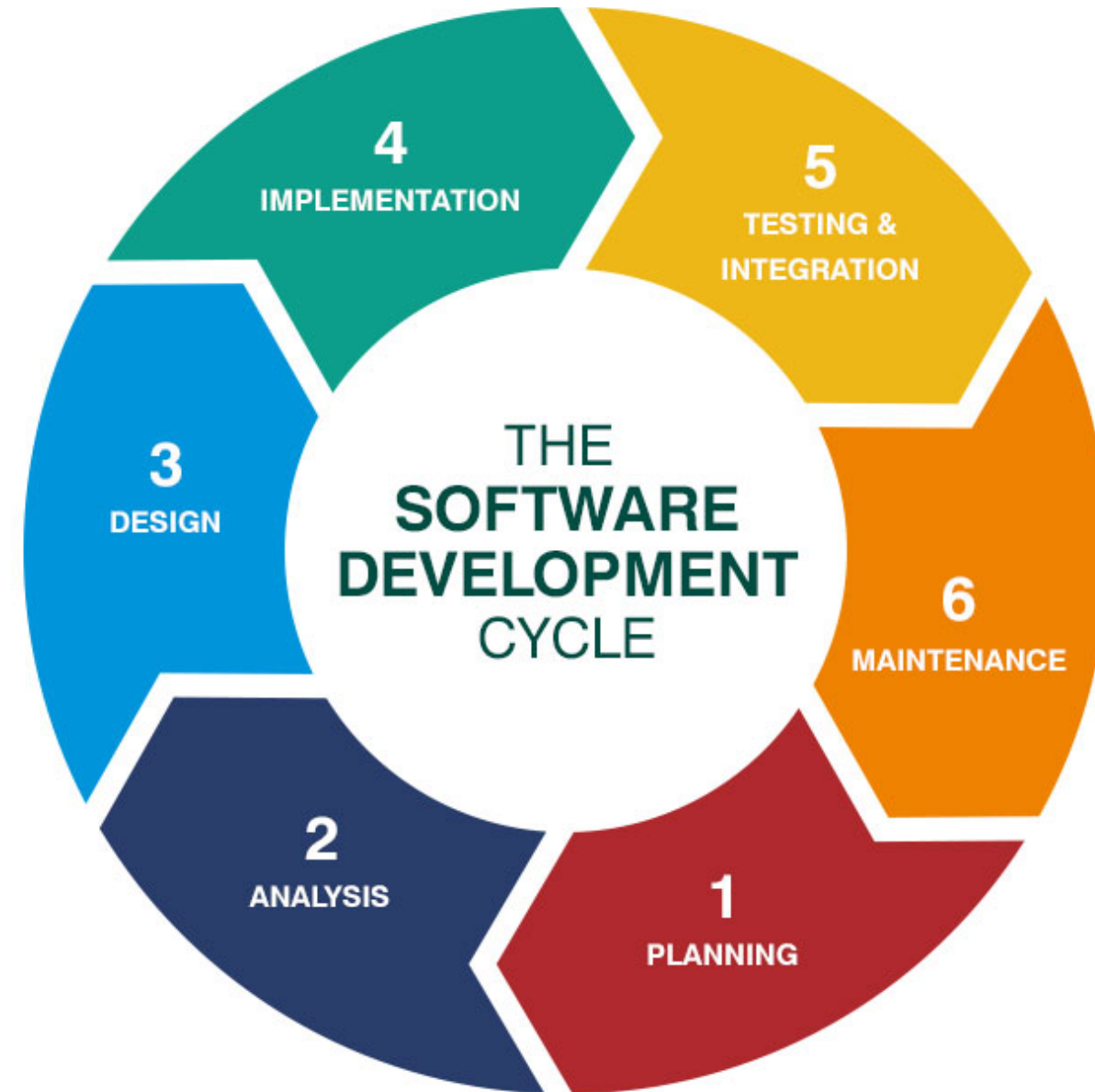
Управляю проектом по разработке интеграционного микросервисного слоя для цифровых продуктов.

Что обсудим

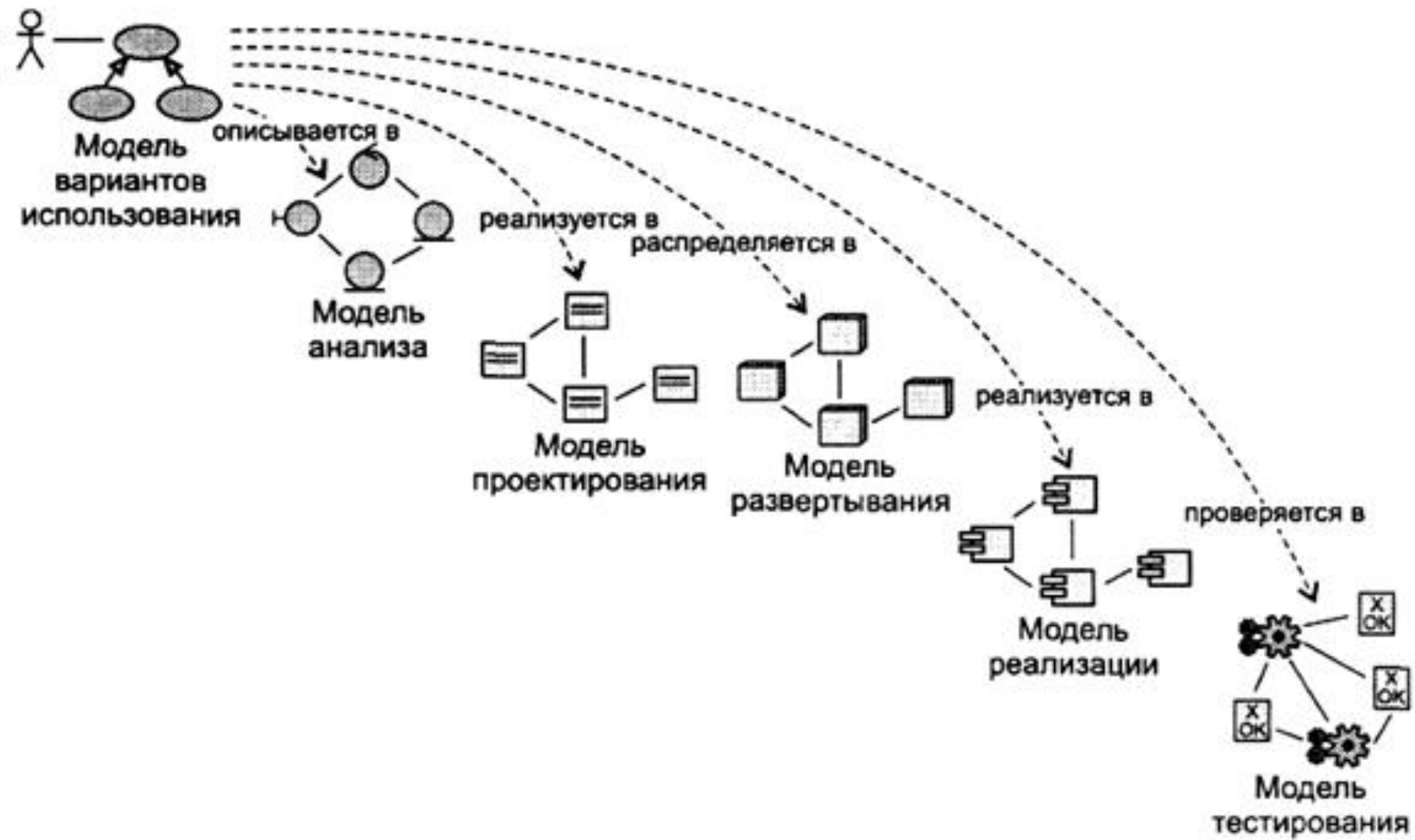
Поговорим про фреймворк системного аналитика: основные инструменты и как их совместно использовать при создании продукта.

1. Пользовательские требования.
2. Сценарии использования.
3. Моделирование данных.
4. Проектирование API.
5. Прототипирование UX.
6. Отдельные системные требования.

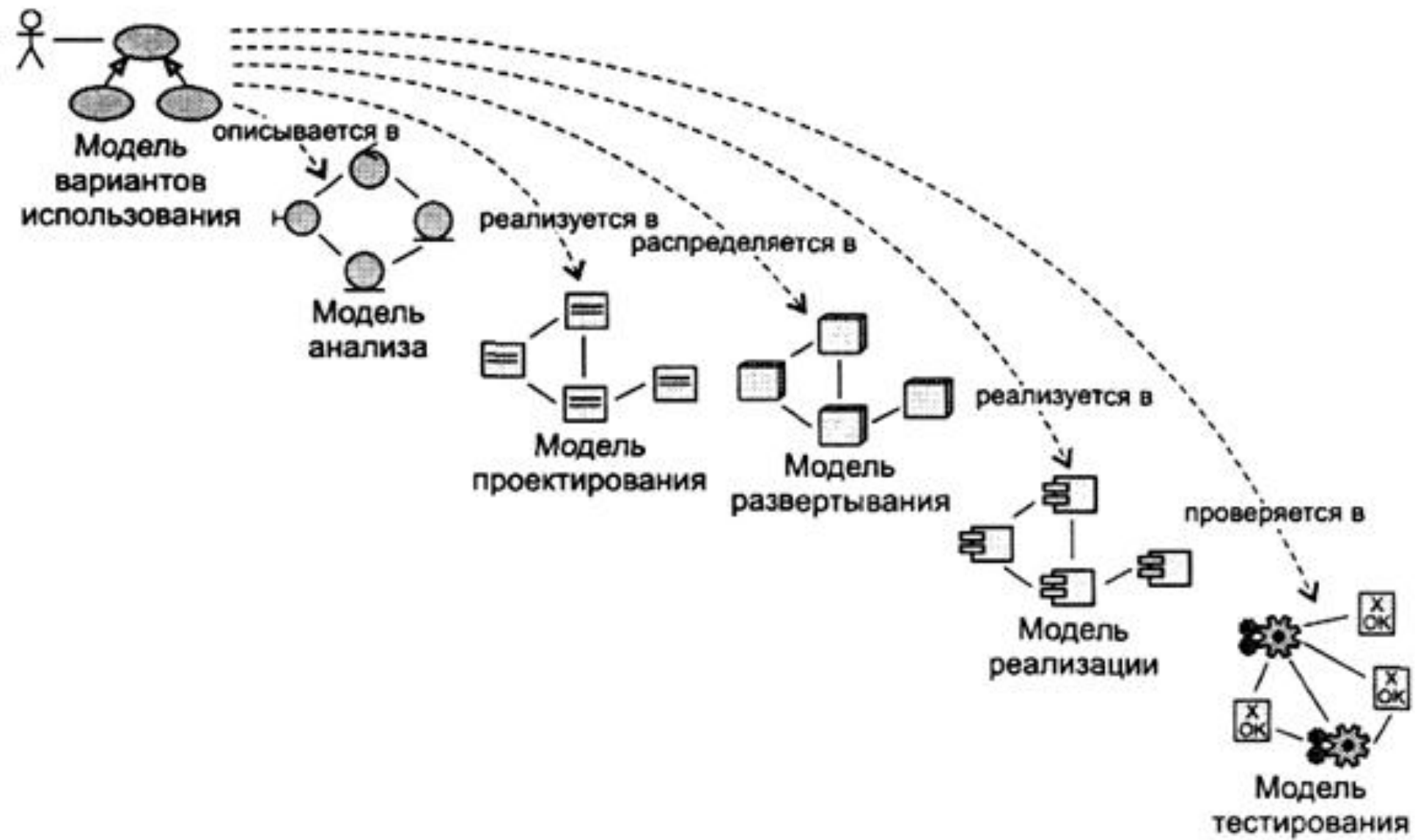
Пару слов про процесс разработки



Как проводить проектирование?



Сложно, долго, дорого



Agile!

User Story

User Story

User Story

User Story

User Story



Как проводить проектирование?

User Story

User Story

User Story

User Story

User Story



Как проводить проектирование?

User Story

User Story

User Story

User Story

User Story



Как проводить проектирование?

Acceptance Test

Business Rule (Template)

Change Case (Template)

Class Responsibility Collaborator (CRC) model

Constraint

Contract model (Template)

Data Flow Diagram (DFD)

Domain Model

Essential/Abstract Use Case (Template)

Essential/Abstract User Interface Prototype

Feature

Free-Form Diagrams

Flow Chart

Glossary

System Use Case (Template)

Technical Requirement

UML Activity Diagram

UML Class Diagram

UML Communication/Collaboration Diagram

UML Component Diagram

UML Composite Structure Diagram

UML Deployment Diagram

UML Interaction Overview Diagram

UML Object Diagram

UML Package Diagram

UML Sequence Diagram

UML State Machine Diagram

Logical Data Model (LDM)

Mind Map

Network Diagram

Object Role Model (ORM) Diagram

Personas

Physical Data Model (PDM)

Robustness Diagram

UML Timing Diagram

UML Use Case Diagram

Usage Scenario

User Interface Flow Diagram (Storyboard)

User Interface Prototype

User Story

Value Stream Map

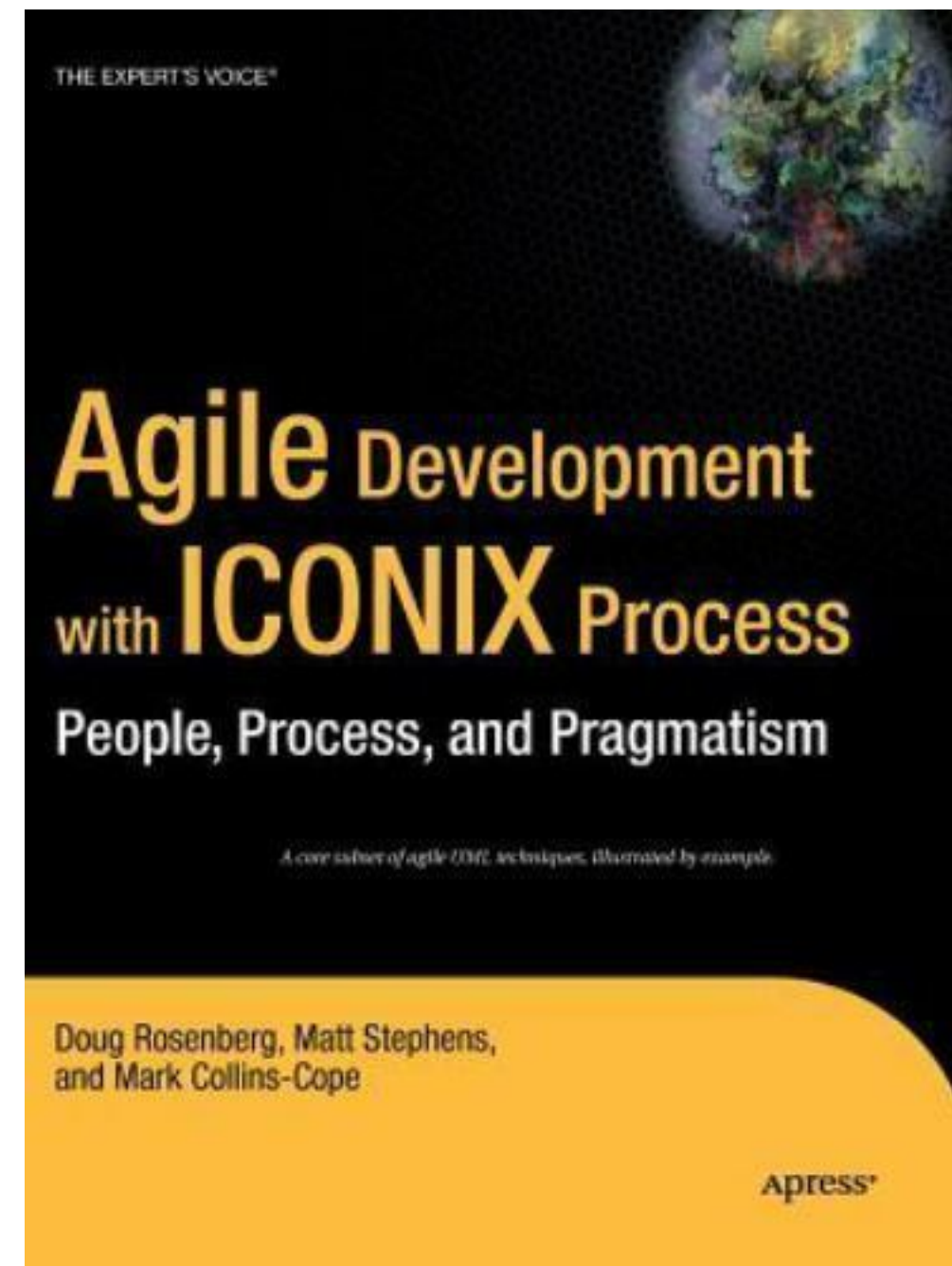
Security Threat Model

Хочется фреймворк

- Проще чем UR, но всё-таки дающий рекомендации.
- Не просто справочник инструментов, но собранный набор.
- Правила совместного использования инструментов.
- Минимально необходимый набор нотаций.
- Фокус на верхнеуровневом проектировании
- Детали — разработке.

ICONIX Process

- Более легковесный, чем UP.
- Сквозная упрощенная модель.
- Но, как и UP, всё ещё сильно привязан к коду.



Область действия

- Разграничиваем ответственность разработчика и системного аналитика.
- Проектирование останавливается на компонентах, до классов и кода.
- Концептуальная модель данных, не нормализуем до логической.
- Используем UML, но только основные диаграммы (можно дополнить).
- Проектируем API (REST от модели данных, RPC от функций).
- Проектируем структуру пользовательского интерфейса (UX, не UI).

Два столпа

1. Модель данных — описание предметной области.
2. Сценарии использования — описание взаимодействия пользователя и системы. Связующий артефакт всей модели системы.

Процесс работы

1. Первичная модель данных предметной области.
2. Пользовательские цели.
3. Раскрытие целей сценариями.
4. Уточнение модели данных.
5. Проектирование интерфейсов.
6. Вычленение системных требований (если нужно).

Модель данных

- Начинаем проект с погружения в предметную область.
- Схема основных сущностей предметной области.
- ER-модель: сущности, атрибуты, связи.
- Концептуальная, не логическая! Структуру БД определяет разработка.

Модель данных

Нотации:

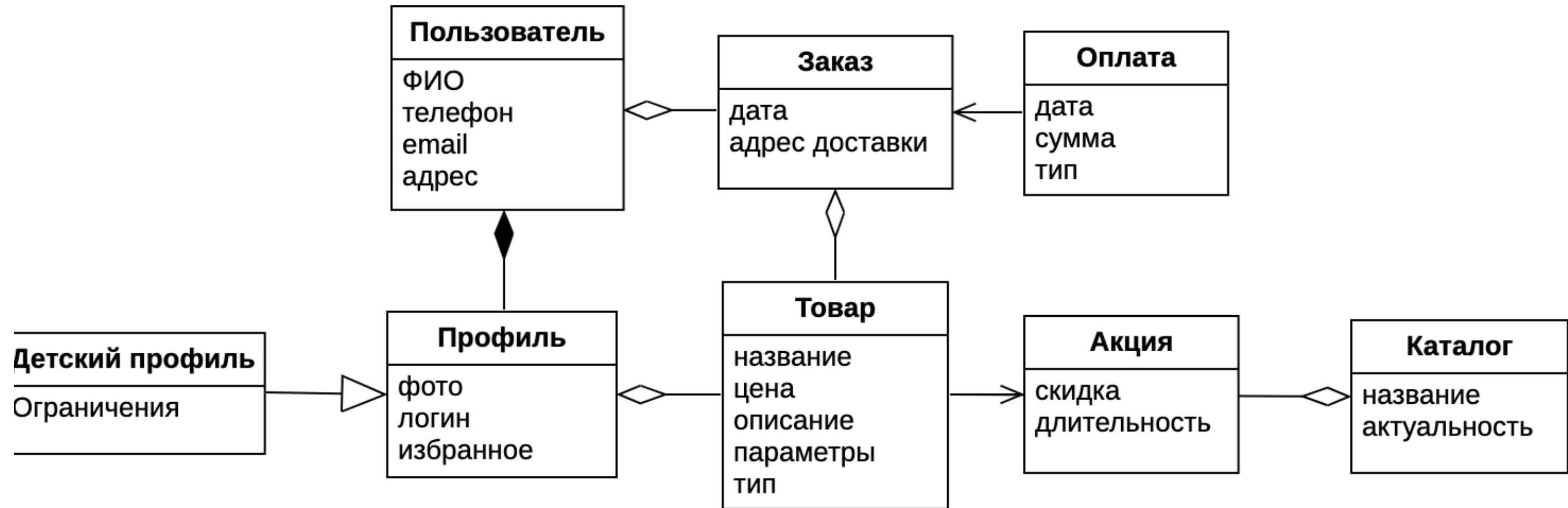
- Crow's Foot ER
- EERD
- Archimate
- UML Class Diagram

Модель данных

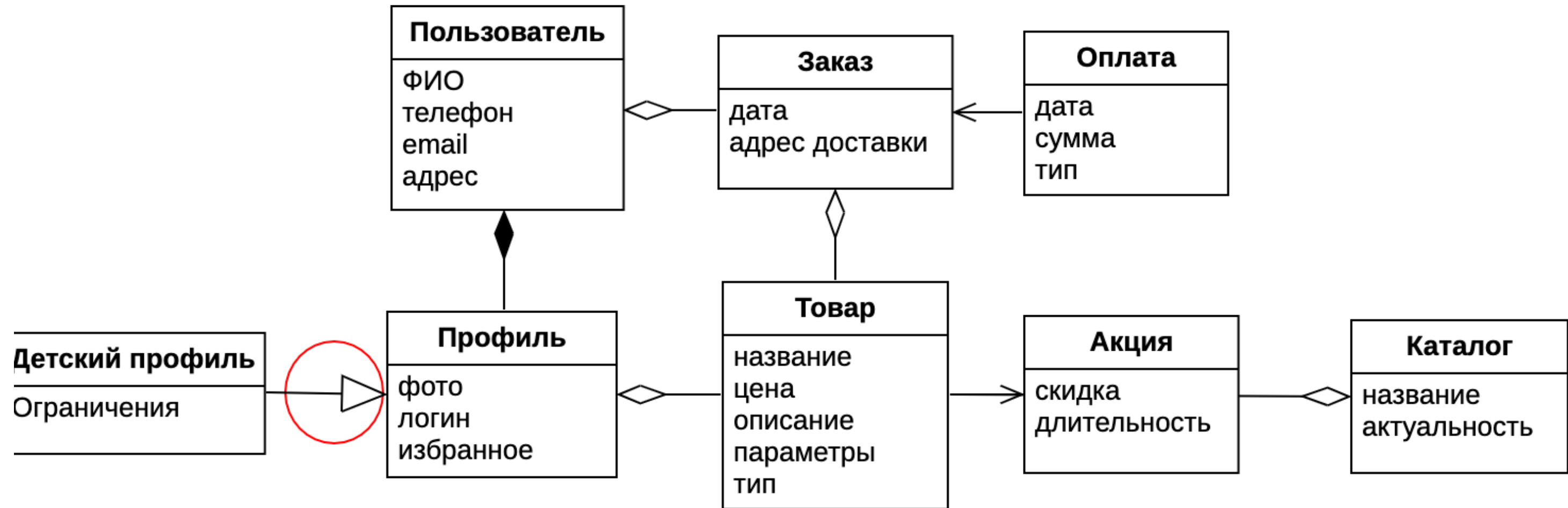
Нотации:

- Crow's Foot ER
- EERD
- Archimate
- **UML Class Diagram**

Модель предметной области



Модель предметной области



Пользовательские цели

- Считаем, что бизнес-требования есть.
- Определяем потребности пользователя.
- Job Story — первичная потребность.
- User Story — пользовательское требование.

Пользовательские цели






Когда _____, я хочу _____, чтобы _____

Когда я нашёл товар, я хочу его отложить на потом, чтобы найти его, когда захочу купить.

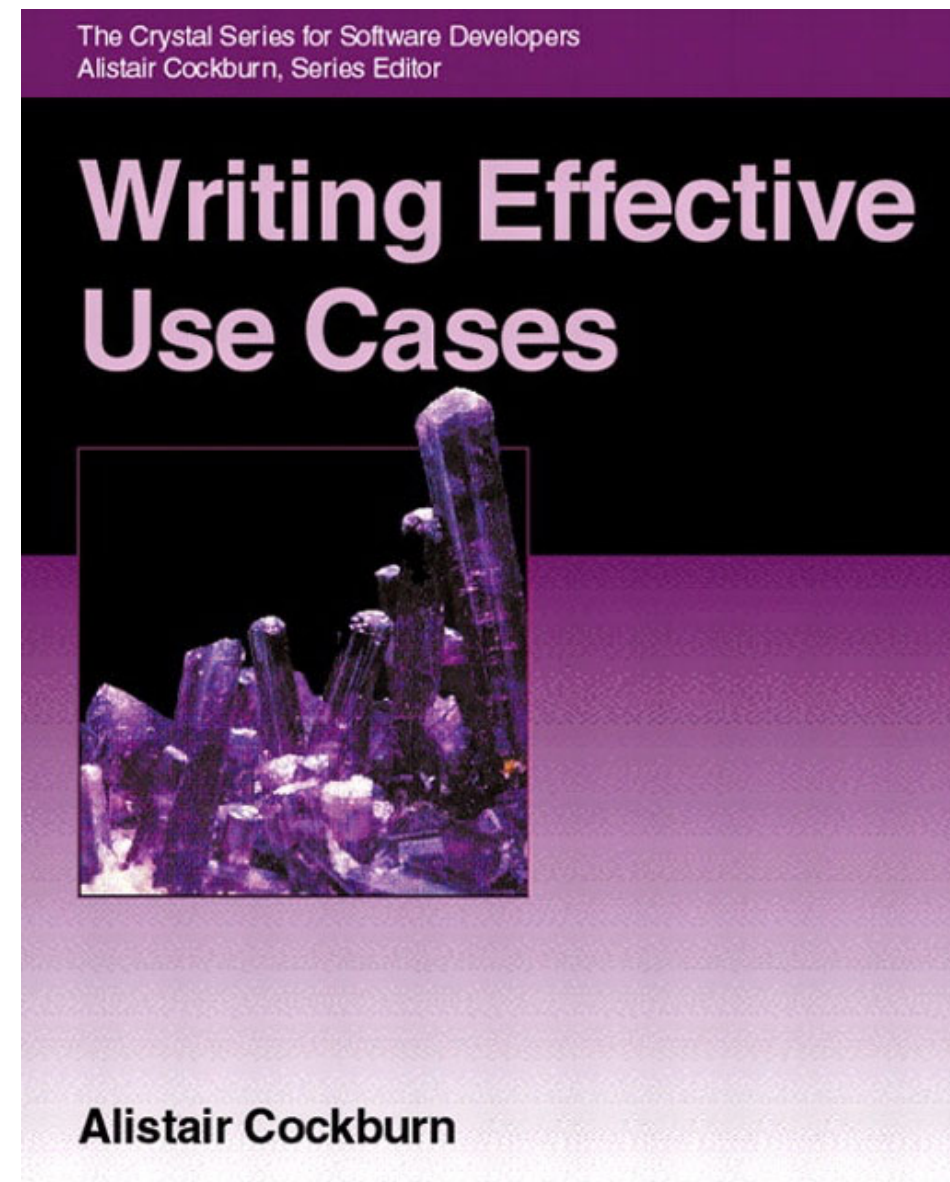
Я как _____, хочу _____, чтобы _____

Я как пользователь хочу добавить товар в избранное, чтобы найти его, когда захочу его купить.

Уровни целей

Cloud level		High summary
Kite level		Summary
Sea level		User goal
Fish level		Sub-function
Claim level		Too low

Уровни целей



Уровни целей

Я как пользователь хочу оплачивать услуги через PayPal, чтобы не вбивать данные карты.

Я как пользователь хочу видеть текущий баланс при оплате услуг, чтобы знать на сколько нужно его пополнить.

Cloud level



High summary

Kite level



Summary

Sea level



User goal

Fish level



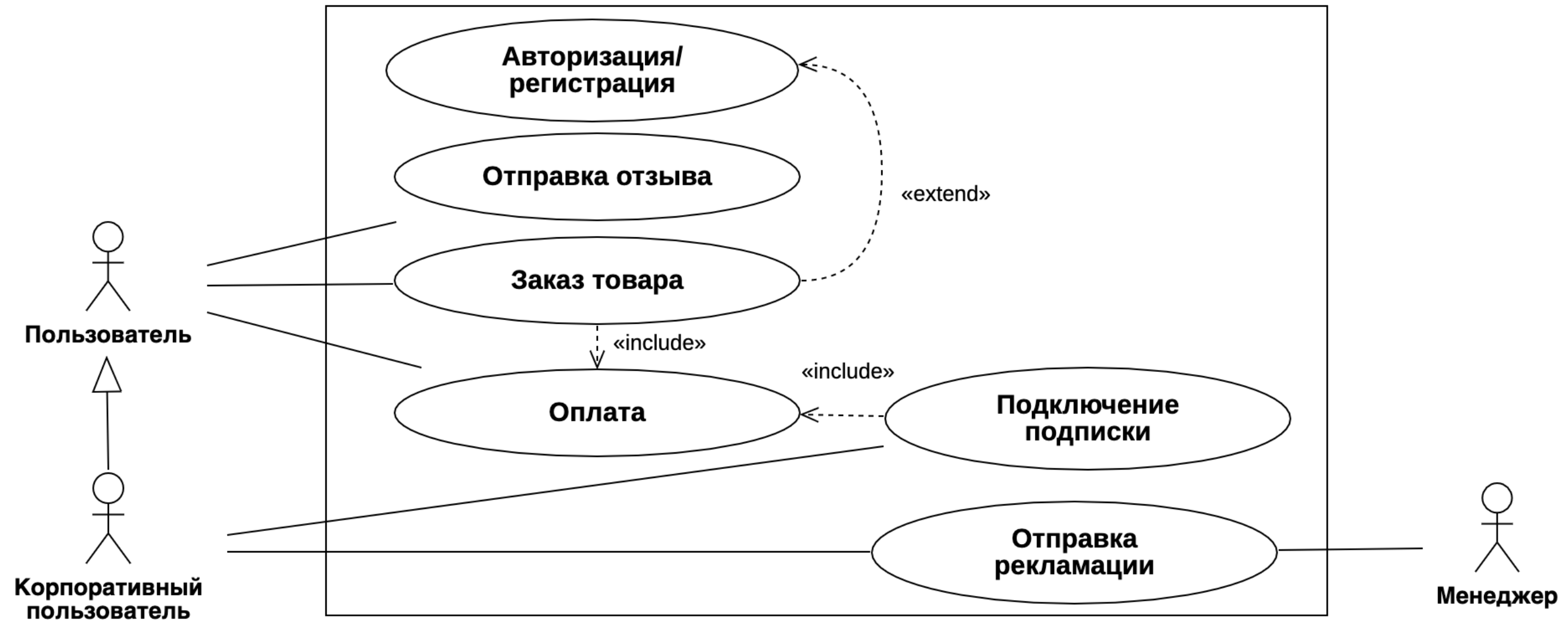
Sub-function

Claim level



Too low

Уровни целей



Сценарий использования

1. Реализует цель пользователя.
2. Описывает взаимодействие пользователя и системы.
3. Состоит из шагов стандартного формата.
4. Кроме основной ветки — альтернативные и исключения.
5. Те же названия сущностей, что в модели данных.
6. Не описывает детали интерфейса, но содержит названия экранов и их основных частей (модальных окон, блоков).

Сценарий использования

1. Реализует цель пользователя.
2. Описывает взаимодействие пользователя и системы.
3. Состоит из шагов стандартного формата.
4. Кроме основной ветки — альтернативные и исключения.
5. **Те же названия сущностей**, что в модели данных.
6. Не описывает детали интерфейса, но **содержит названия экранов и их основных частей** (модальных окон, блоков).

Сценарии использования

Цель сценария:

я как пользователь,

хочу оставить отзыв к товару,

чтобы поделиться своим мнением о нём.

Сценарии использования

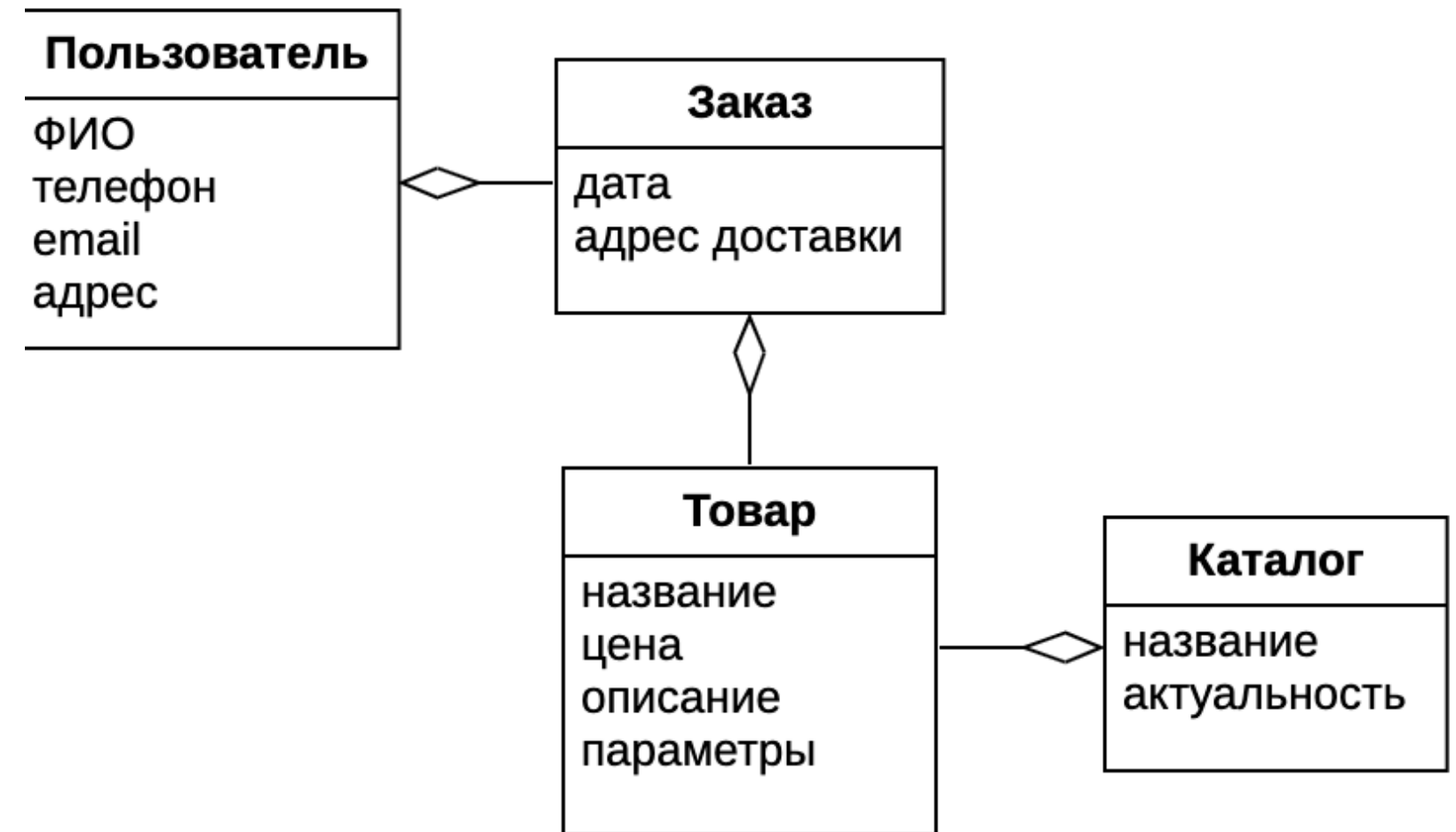
1. Пользователь запрашивает список товаров из каталога.
2. Приложение отображает страницу каталога.
3. Пользователь выбирает товар.
4. Приложение отображает страницу товара.
5. Пользователь выбирает просмотр отзывов.
6. Приложение отображает страницу отзывов.
7. Пользователь оставляет новый отзыв.
8. Приложение добавляет отзыв к товару.

Сценарии использования

1. Пользователь запрашивает список **товаров** из **каталога**.
2. Приложение отображает страницу каталога.
3. Пользователь выбирает **товар**.
4. Приложение отображает страницу товара.
5. Пользователь выбирает просмотр отзывов.
6. Приложение отображает страницу отзывов.
7. Пользователь оставляет **новый отзыв**.
8. Приложение добавляет отзыв к **товару**.

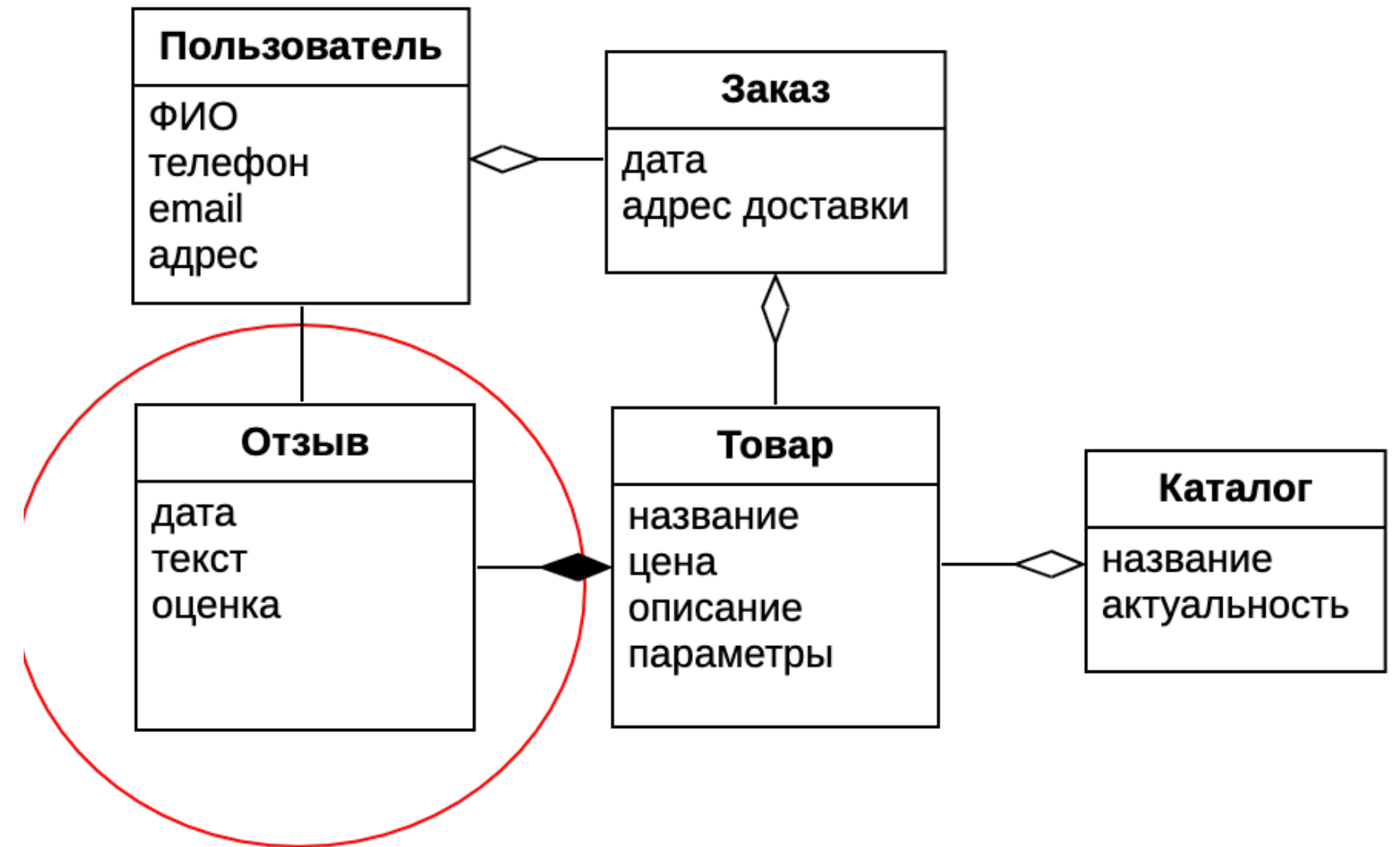
Сценарии использования и сущности

1. Пользователь запрашивает список **товаров** из **каталога**.
2. Приложение отображает страницу каталога.
3. Пользователь выбирает **товар**.
4. Приложение отображает страницу товара.
5. Пользователь выбирает просмотр отзывов.
6. Приложение отображает страницу отзывов.
7. Пользователь оставляет новый отзыв.
8. Приложение добавляет отзыв к **товару**.

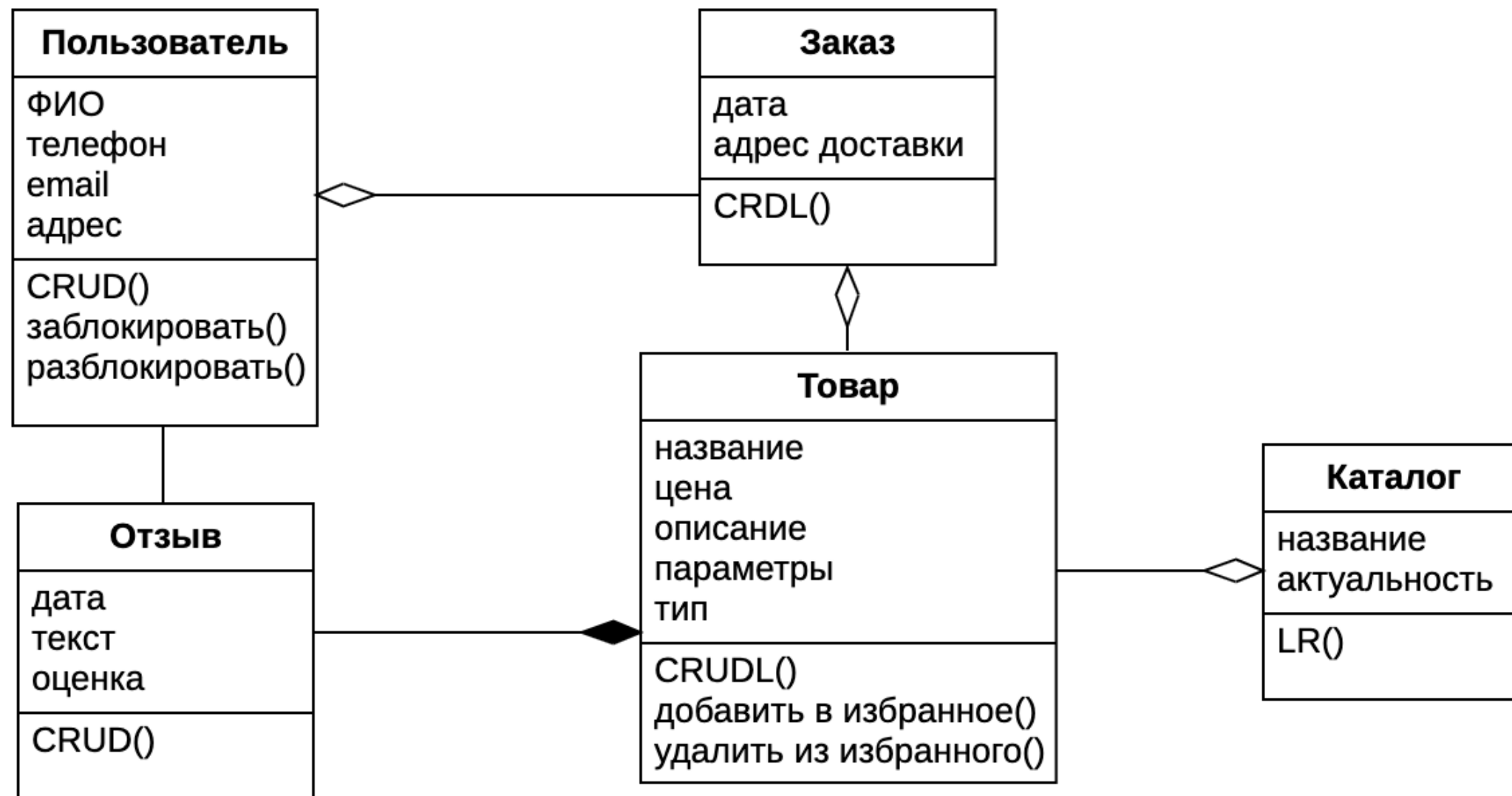


Сценарии использования и сущности

1. Пользователь запрашивает список товаров из каталога.
2. Приложение отображает страницу каталога.
3. Пользователь выбирает товар.
4. Приложение отображает страницу товара.
5. Пользователь выбирает просмотр **ОТЗЫВОВ**.
6. Приложение отображает страницу отзывов.
7. Пользователь оставляет новый **ОТЗЫВ**.
8. Приложение добавляет **ОТЗЫВ** к товару.



Сценарии использования и сущности: операции



Сценарии использования и UI

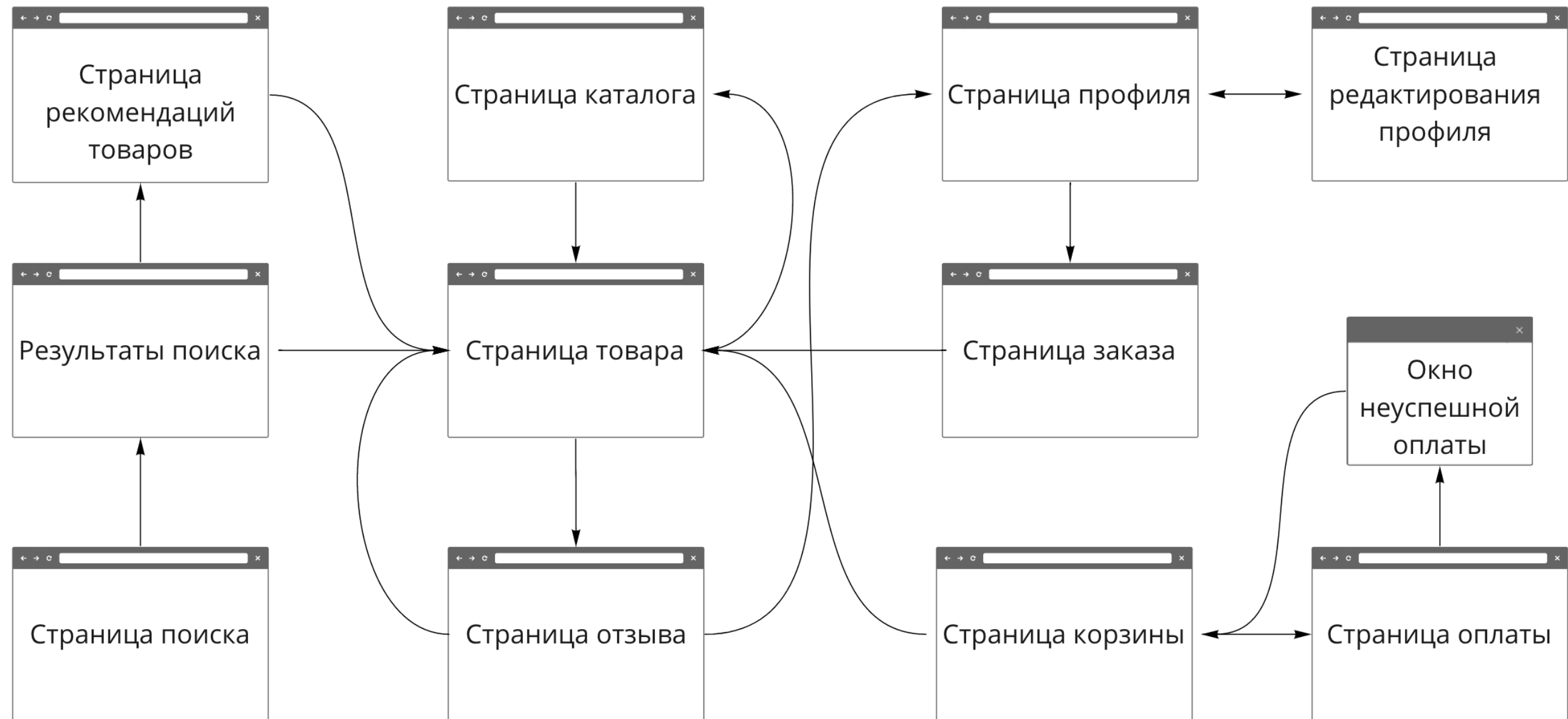
1. Пользователь запрашивает список товаров из каталога.
2. Приложение отображает **страницу каталога**.
3. Пользователь выбирает товар.
4. Приложение отображает **страницу товара**.
5. Пользователь выбирает просмотр отзывов.
6. Приложение отображает **страницу отзывов**.
7. Пользователь оставляет новый отзыв.
8. Приложение добавляет отзыв к товару.

Карта экранов

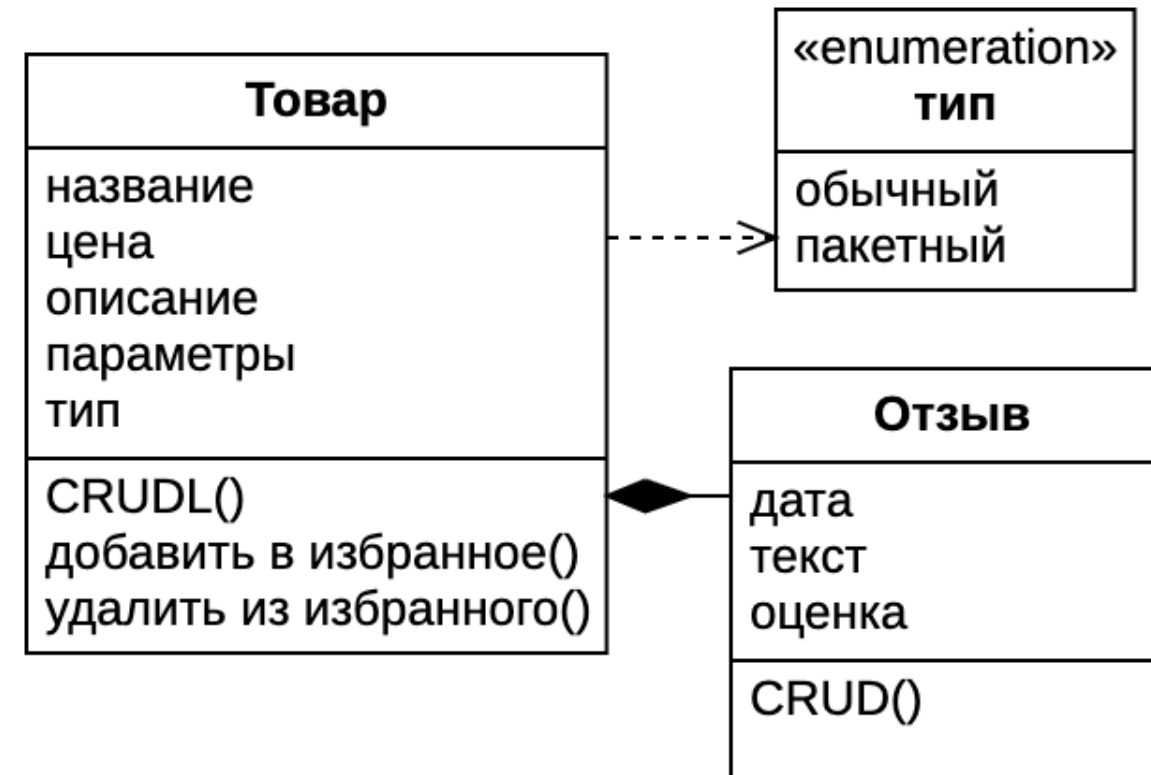
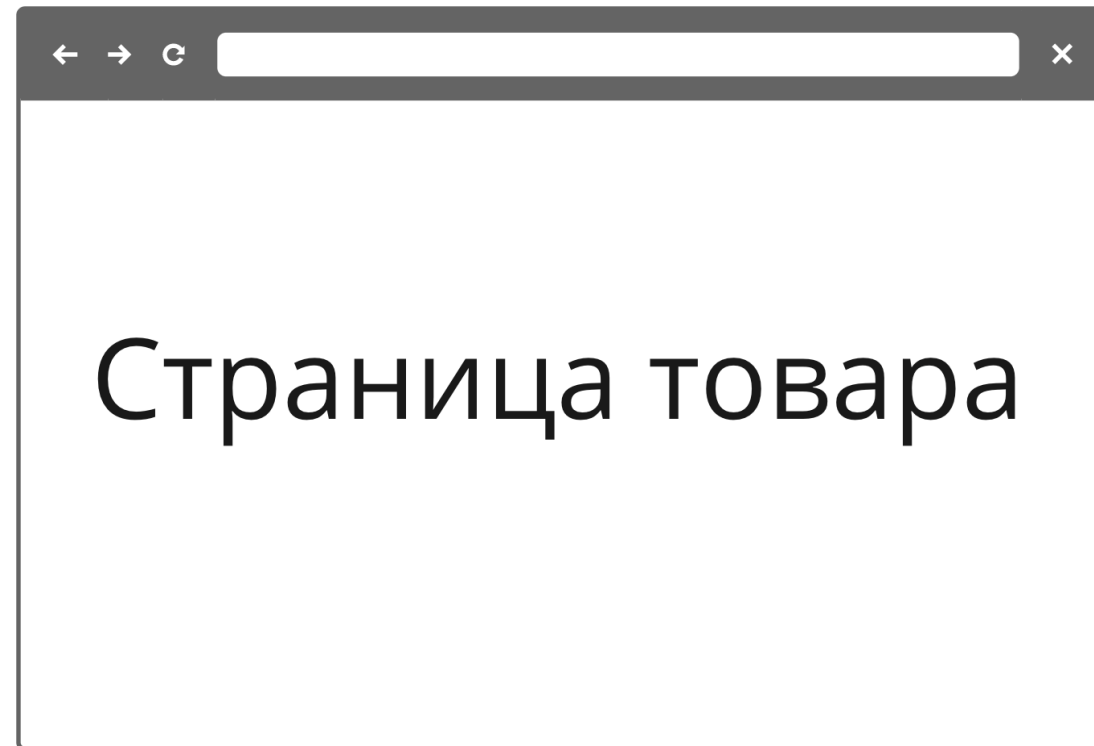
1. Пользователь запрашивает список товаров из каталога.
2. Приложение отображает **страницу каталога**.
3. Пользователь выбирает товар.
4. Приложение отображает **страницу товара**.
5. Пользователь выбирает просмотр отзывов.
6. Приложение отображает **страницу ОТЗЫВОВ**.
7. Пользователь оставляет новый отзыв.
8. Приложение добавляет отзыв к товару.



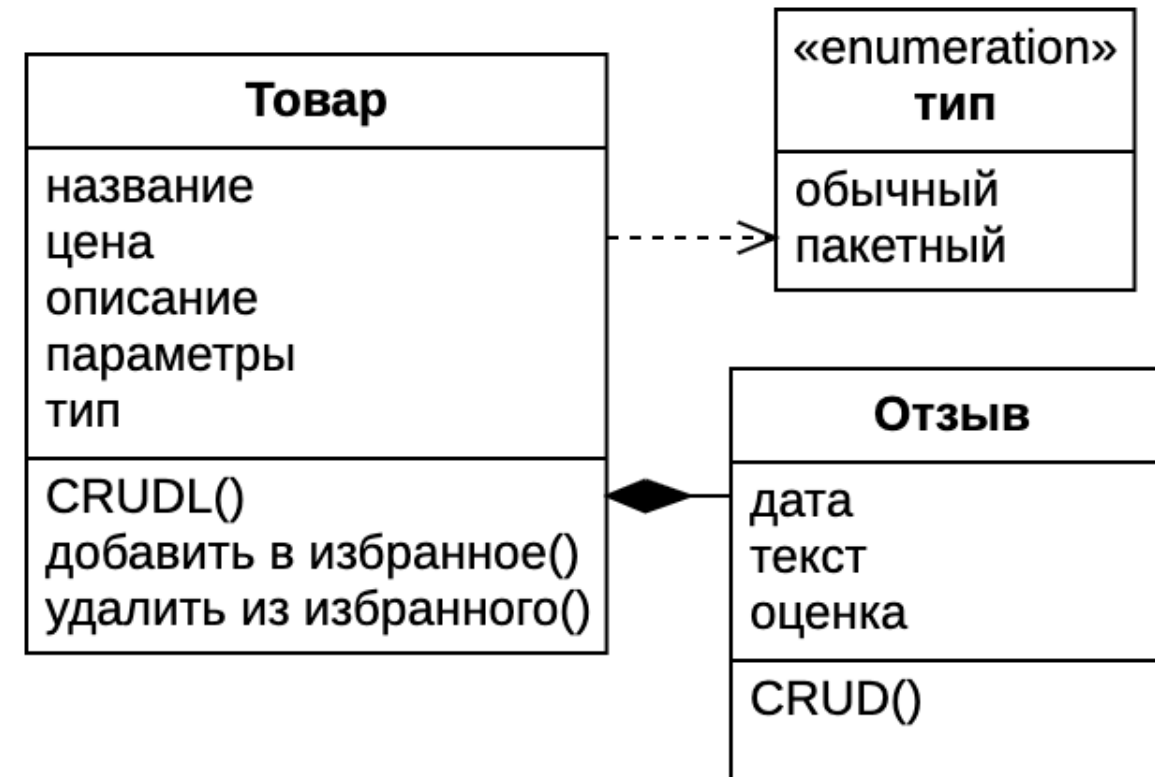
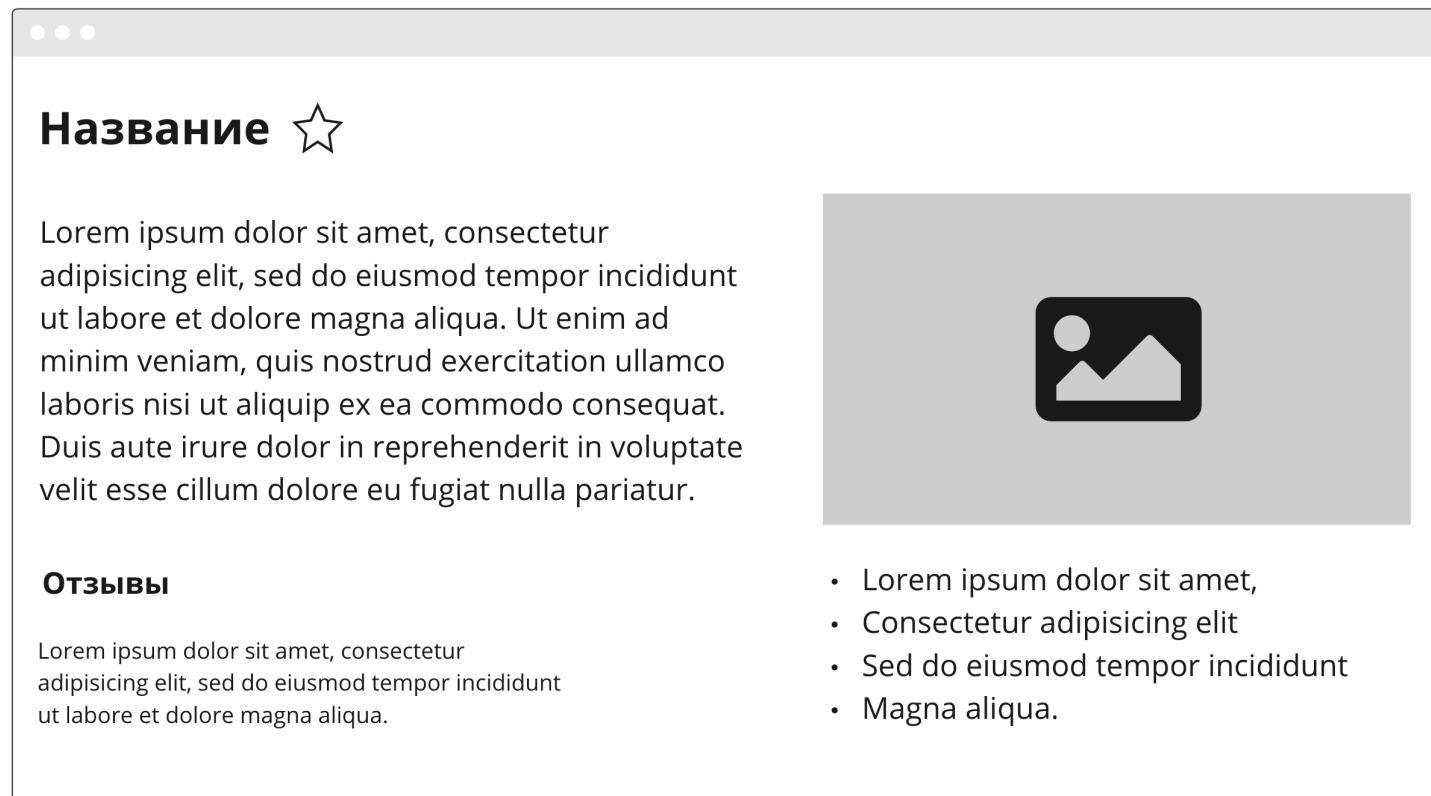
Карта экранов



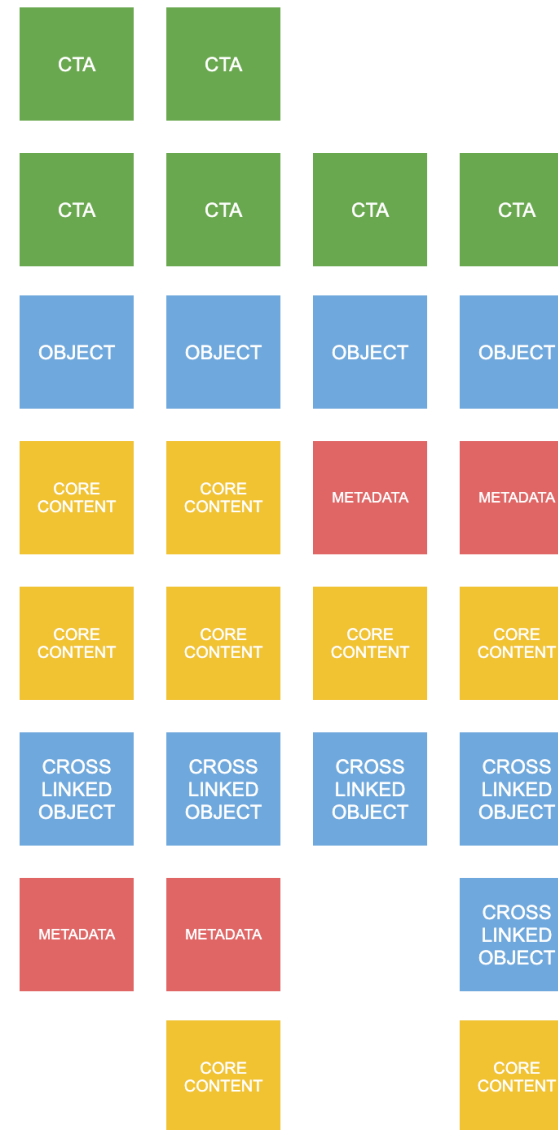
Содержимое экранов



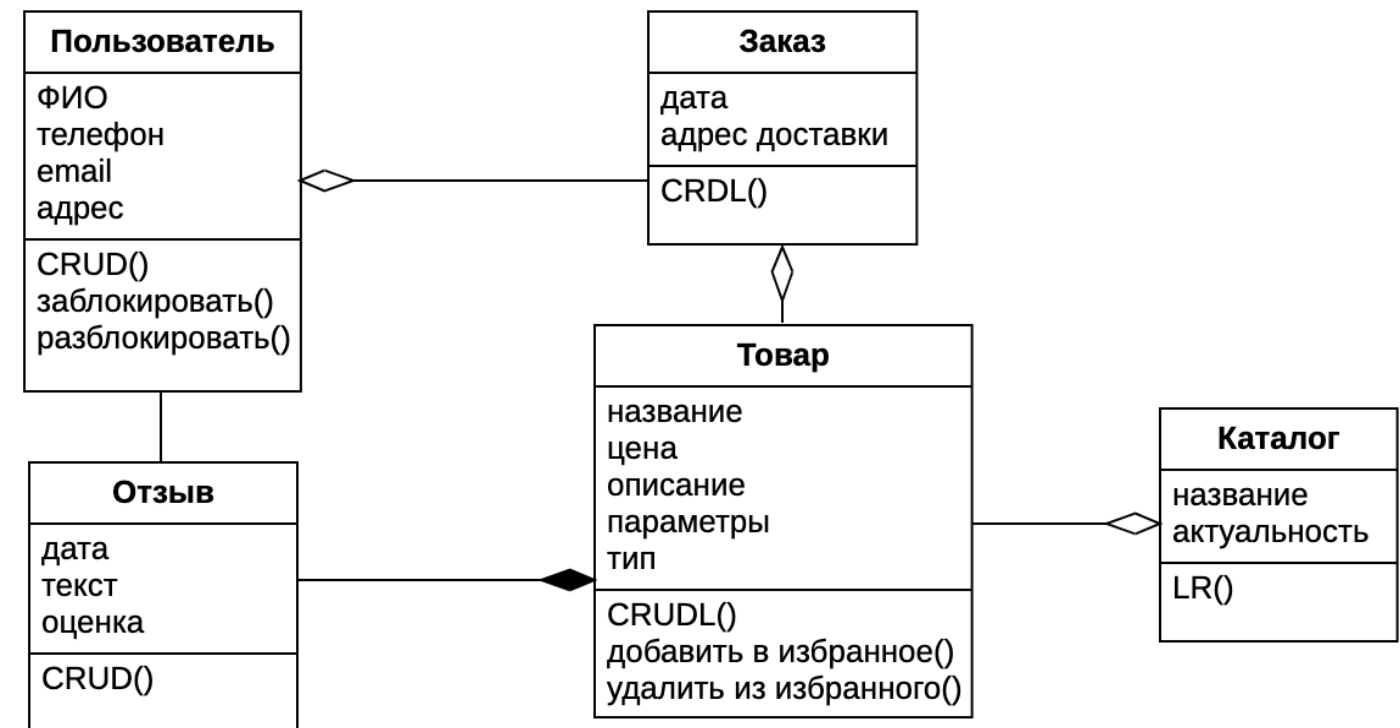
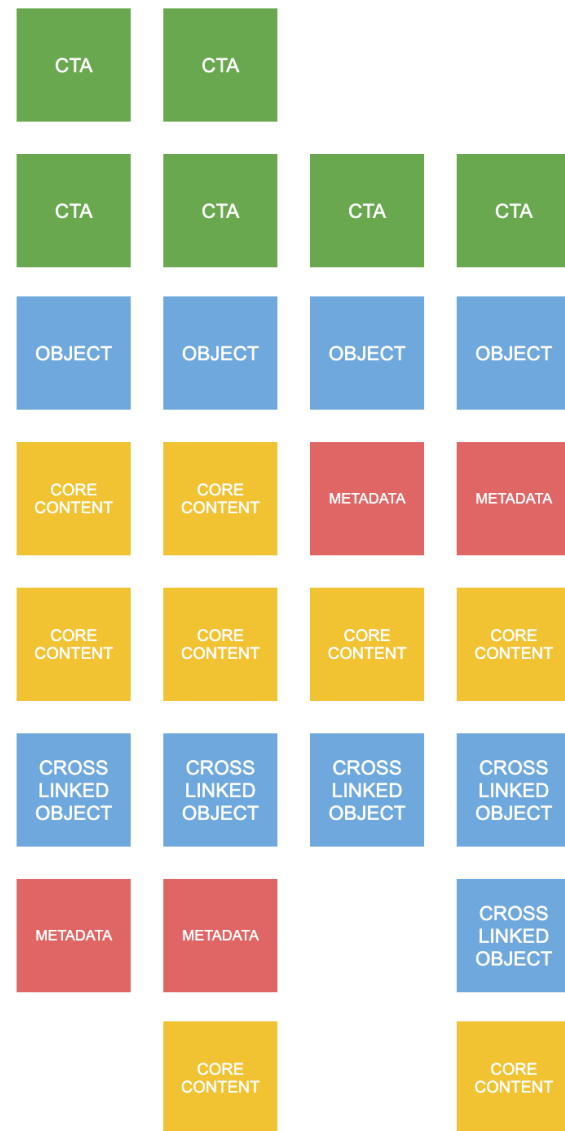
Содержимое экранов



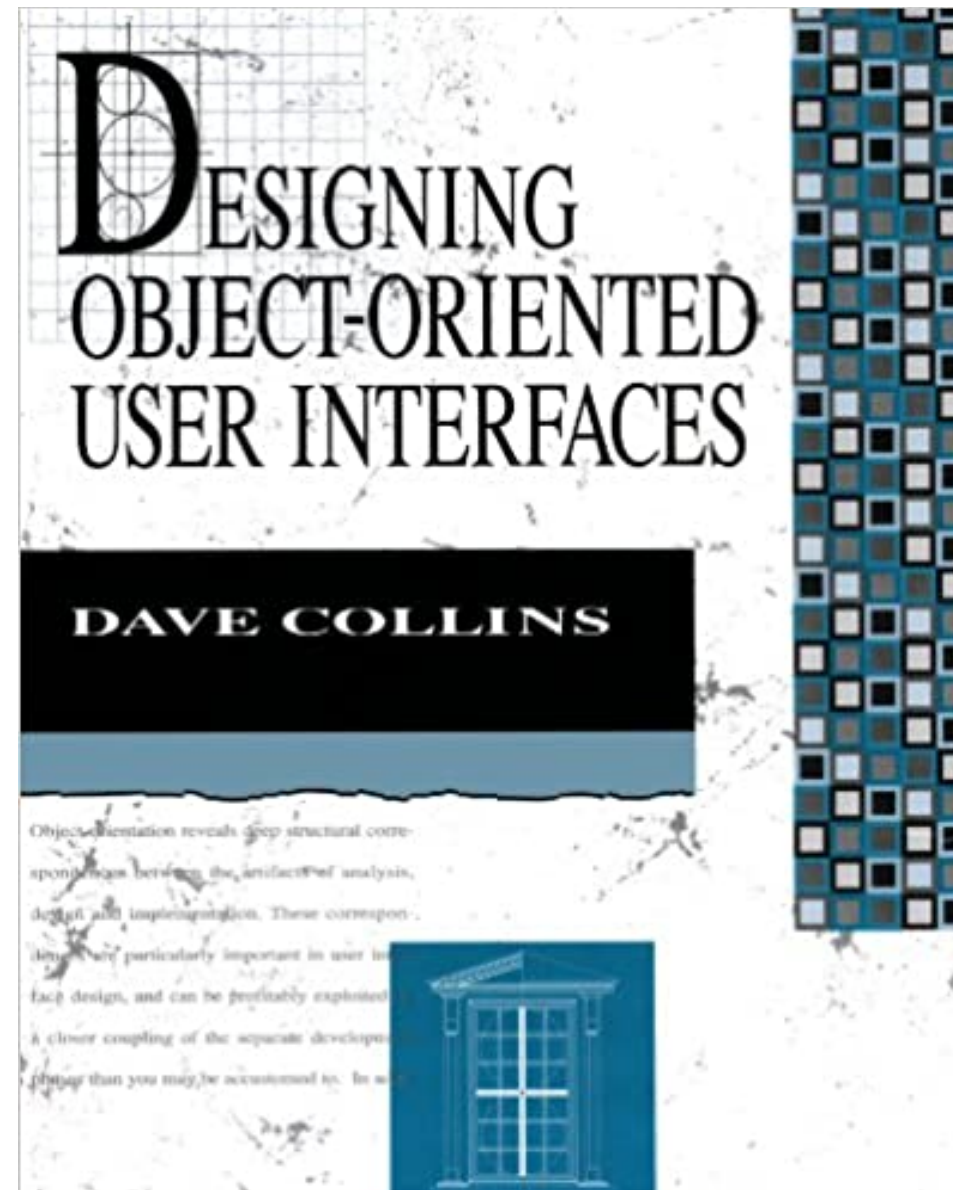
Objected Oriented UX



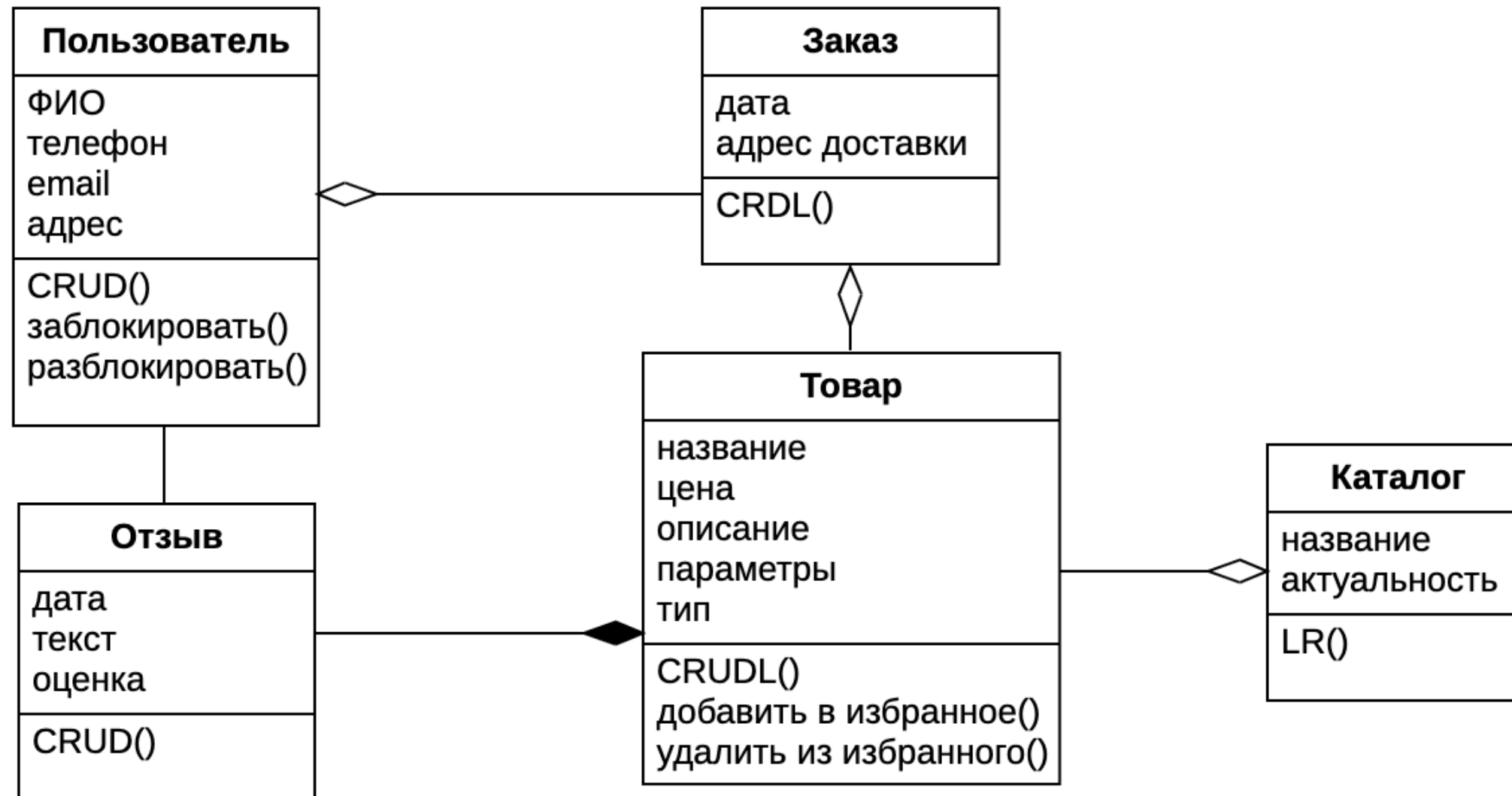
Objected Oriented UX



Objected Oriented UX

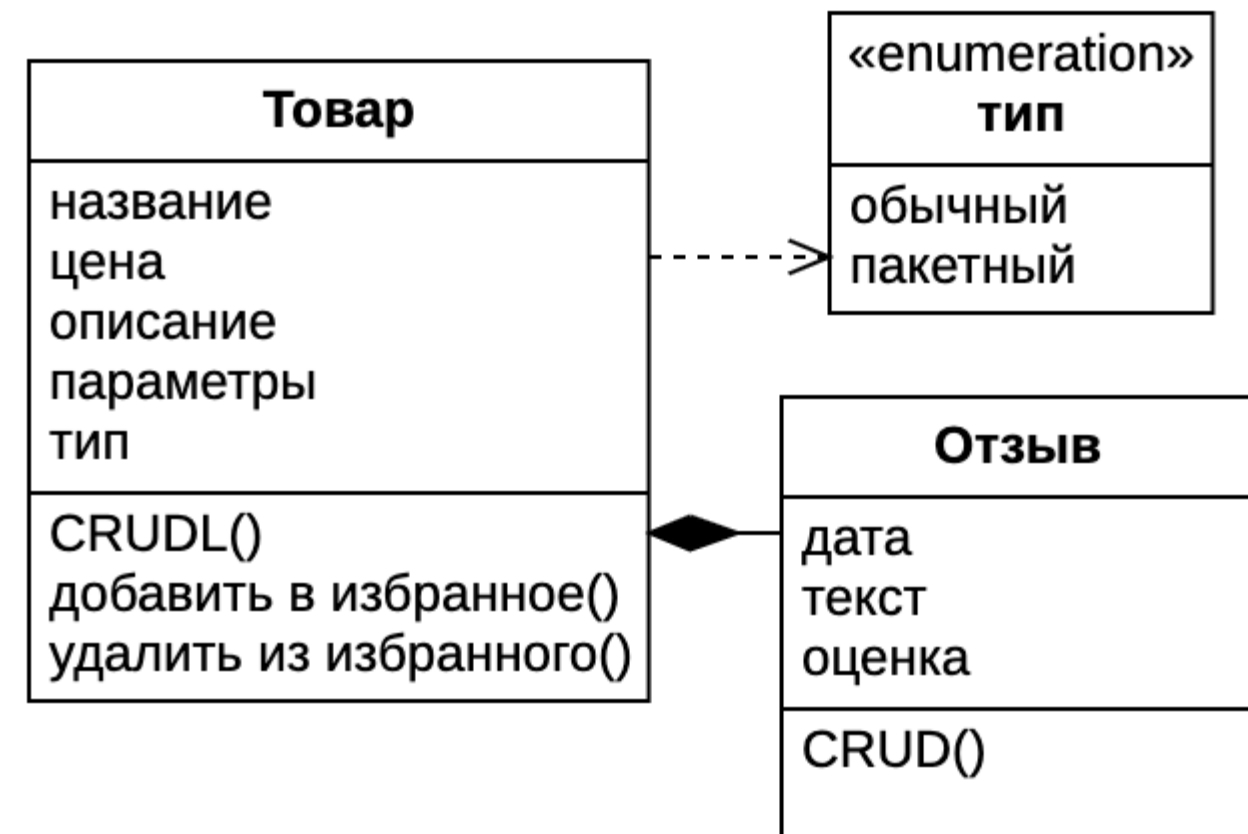


Программные интерфейсы



Программные интерфейсы

GET /users
POST /users
GET /users/{id}
PUT /users/{id}
DELETE /users/{id}
POST /users/{id}/block
POST /users/{id}/unblock
GET /products
GET /products/{id}
GET /products/{id}/reviews
POST /products/{id}/reviews
GET /products/{id}/reviews/{id}
PATCH /products/{id}/reviews/{id}
DELETE /products/{id}/reviews/{id}
...



Системные требования

1. Пользователь запрашивает список товаров из каталога.
2. Приложение отображает страницу каталога.
3. Пользователь выбирает товар.
4. Приложение отображает страницу товара.
5. Пользователь выбирает просмотр отзывов.
6. Приложение отображает страницу отзывов.
7. Пользователь оставляет новый отзыв.
8. Приложение добавляет отзыв к товару и пересчитывает итоговый отзыв.
9. Приложение отправляет СМС-нотификацию пользователю о новом отзыве.

Системные требования

1. Пользователь запрашивает список товаров из каталога.
2. Приложение **отображает страницу каталога**.
3. Пользователь выбирает товар.
4. Приложение **отображает страницу товара**.
5. Пользователь выбирает просмотр отзывов.
6. Приложение **отображает страницу отзывов**.
7. Пользователь оставляет новый отзыв.
8. Приложение **добавляет отзыв к товару и пересчитывает итоговый отзыв**.
9. Приложение **отправляет СМС-нотификацию** пользователю о новом отзыве.

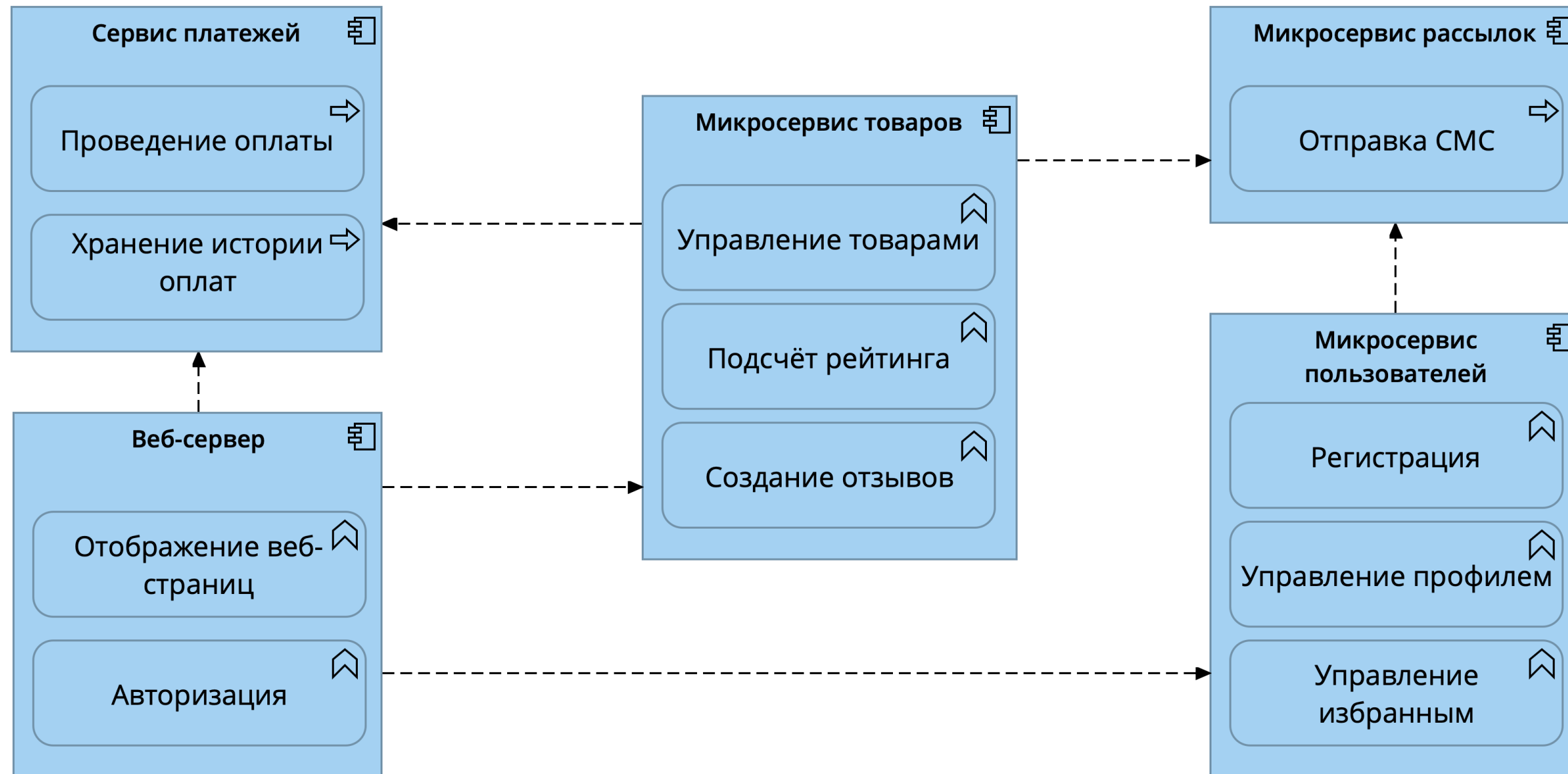
Системные требования: функциональность

Когда пользователь запрашивает информацию о товаре система должна отображать страницу товара.

Когда к товару добавлен новый отзыв, система должна отправлять СМС-нотификацию автору отзыва.

Когда к товару добавлен новый отзыв с рейтингом, система должна пересчитывать итоговый рейтинг товара.

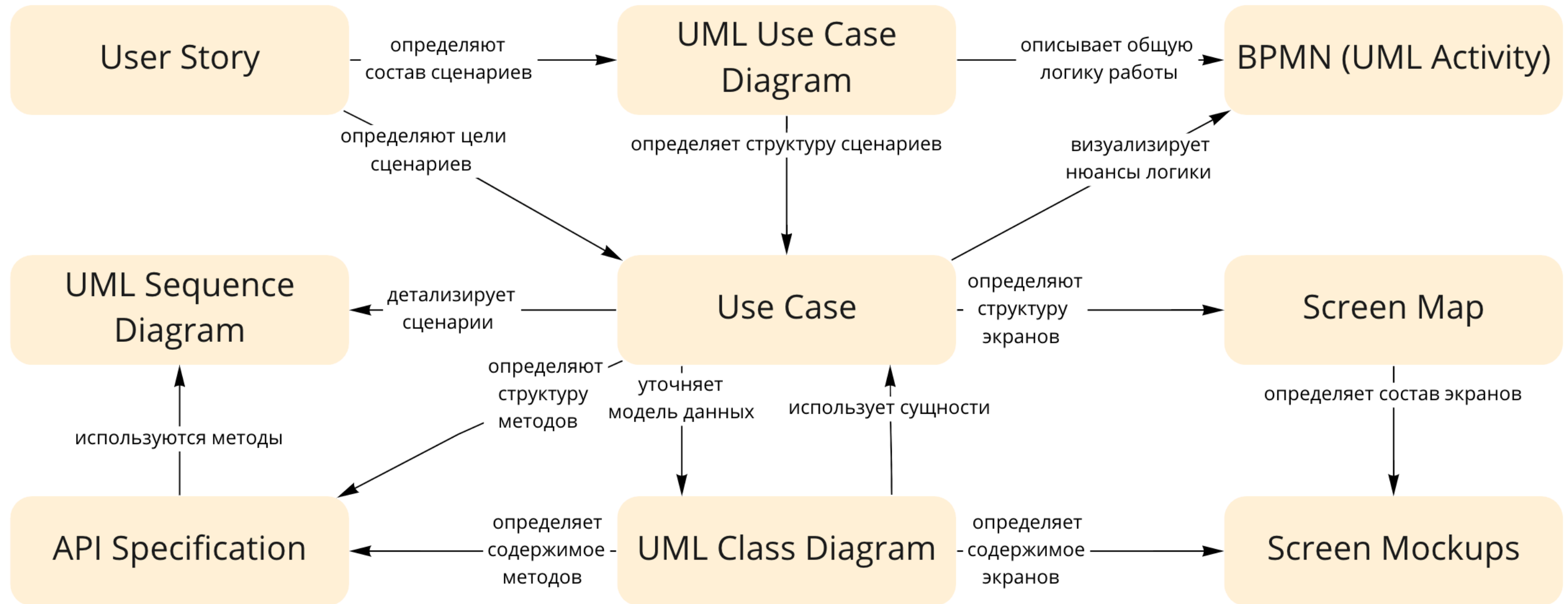
Системные требования: аллокация функций



Итого

1. Модель данных предметной области.
2. Пользовательские цели для сценариев.
3. Прорабатываем сценарии, уточняем модель данных.
4. Из сценариев — структура экранов и API.
5. Из модели данных — их содержимое.
6. Если нужно, вычленяем из сценариев функции.

Итого



Нотации

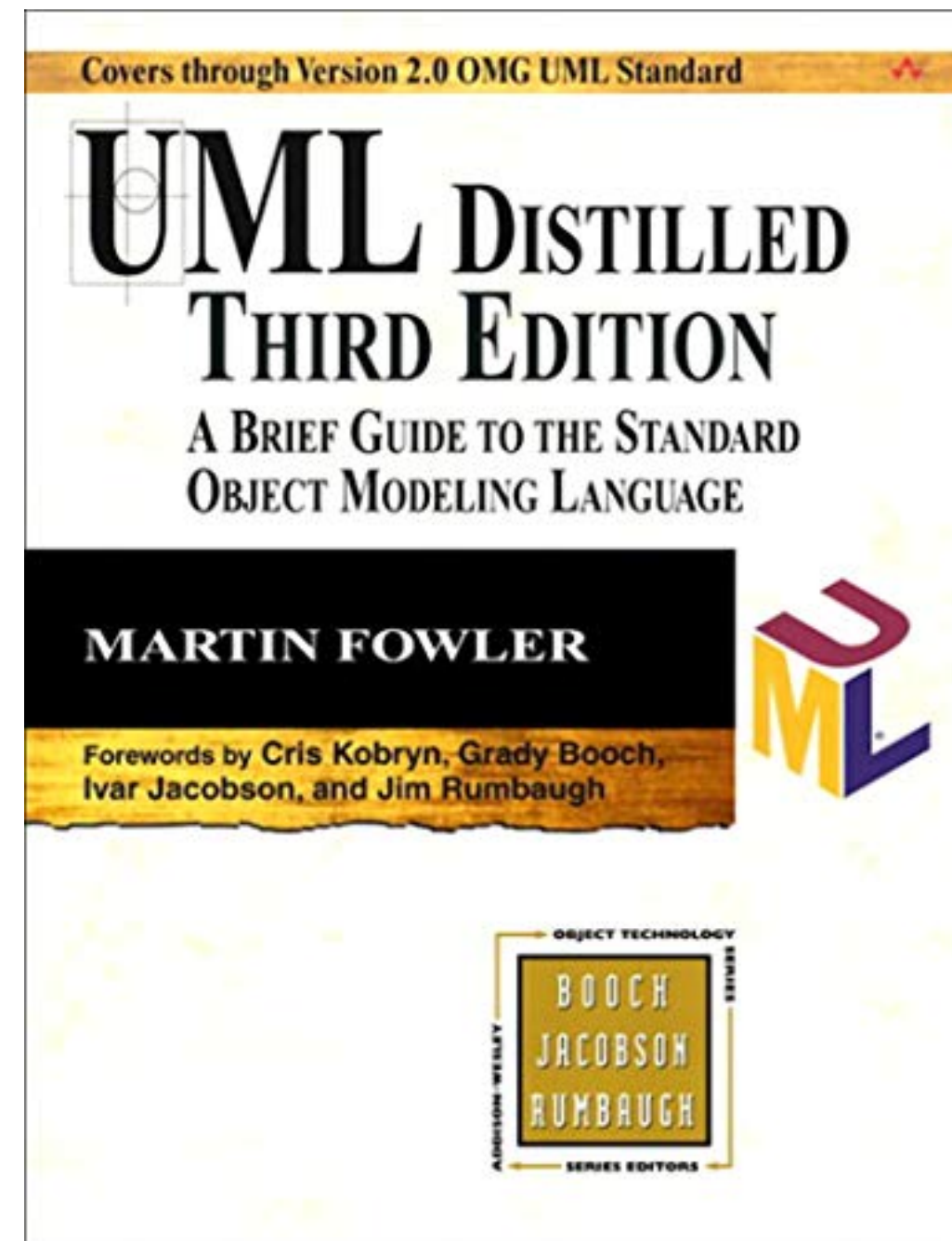
UML Use Case Diagram

UML Class Diagram

UML Sequence Diagram

UML Activity Diagram (или BPMN)

Бонусом — UML Information Flow Diagram (или DFD), Archimate



Спасибо!

hello@kvaleev.me

@kvaleev✉

