

Монолит

- 1) Позволит быстрее разработать и запустить первую версию
- 2) Проще разработка и сопровождение, тк не сразу настраивать сложную инфраструктуру для микросервисов
- 3) Меньше затрат на старте
- 4) Легче организовать тестирование всех функций
- 5) Быстрая работа, тк все модули вызываются напрямую внутри приложения, без HTTP-запросов между сервисами

Use cases

1)

Описание	Этот Use Case описывает процесс добавления товара из списка акционных товаров в корзину
Актор	Пользователь
Предуслови я	PRE-1. Пользователь находится на сайте интернет-магазина PRE-2. Пользователь открыл страницу с акционными товарами <i>PRE-3. Пользователь имеет текущий статус (например, "Новый", "Постоянный", "VIP")</i>
Постуслови я	POST-1. Товар из списка акционных товаров успешно добавлен в корзину <i>POST-2. Статус клиента может измениться на основании правил</i> <i>POST-3. Скидка применена в зависимости от условий акции или статуса клиента</i> <i>POST-4. Данные об успешном добавлении товара отправлены в систему отчётности</i>
Основной сценарий	

Шаг	Действие
1	Пользователь открывает раздел «Акции» на сайте
2	Система отображает перечень товаров, участвующих в акции на текущий период.
3	Пользователь просматривает список товаров
4	Пользователь выбирает товар
5	Система добавляет выбранный товар в корзину
6	Система определяет, применяется ли дополнительная скидка в зависимости от статуса клиента
7	Система пересчитывает стоимость заказа с учётом акции и скидки клиента
8	Система проверяет, соответствует ли клиент условиям для изменения статуса
9	Система обновляет статус клиента (если условия выполнены)
10	Система уведомляет пользователя о добавлении товара в корзину и изменении скидки
11	Система отправляет информацию в модуль отчётности для формирования отчёта менеджеру

2)

Описание	Этот Use Case описывает процесс оформления заказа пользователем после выбора товаров в корзине
Актор	Пользователь

Предусловия	<p>PRE-1. Пользователь добавил хотя бы один товар в корзину</p> <p>PRE-2. Пользователь авторизован</p> <p><i>PRE-3.</i> Пользователь имеет текущий статус (например: "Новый клиент", "Постоянный клиент", "VIP клиент")</p>
Постусловия	<p>POST-1. Заказ создан в системе</p> <p>POST-2. Пользователь получил подтверждение о заказе</p> <p>POST-3. Пользователь может отслеживать статус заказа («Оформлен», «В ожидании оплаты» и тд)</p> <p><i>POST-4.</i> Статус клиента обновлён, если выполнены условия повышения</p> <p><i>POST-5.</i> Применена персональная скидка в зависимости от статуса</p> <p><i>POST-6.</i> Данные о заказе переданы в систему отчетности для менеджера</p>
Основной сценарий	
Шаг	Действие
1	Пользователь открывает корзину
2	Система отображает список добавленных товаров и сумму заказа
3	Пользователь выбирает товары из корзины
4	Пользователь вводит/подтверждает свои контактные данные
5	Пользователь выбирает способ доставки (курьер, самовывоз)
6	Пользователь выбирает способ оплаты (картой, наличными, онлайн-оплата)
7	Система подсчитывает стоимость заказа

8	Система применяет скидку в зависимости от статуса клиента
9	Система проверяет условия изменения статуса клиента
10	Система обновляет статус клиента, если условия выполнены
11	Пользователь подтверждает оформление заказа
12	Система создает заказ и отображает сообщение о его успешной регистрации
13	Система отправляет данные о заказе в систему отчетности для менеджера

Альтернативный сценарий 1 — Ввод промокода

Шаг замены	для	Название шага	Действие
7		7a	Пользователь вводит промокод. Система проверяет его: если промокод действителен — пересчитывает стоимость заказа с учётом скидки; если недействителен — выводит сообщение "Промокод не действителен"

3)

Описание	Этот Use Case описывает процесс написания пользователем отзыва на ранее приобретённый товар
Актор	Пользователь
Предусловия	PRE-1. Пользователь авторизован PRE-2. У пользователя есть хотя бы один завершённый заказ с доставленным товаром
Постусловия	POST-1. Новый отзыв сохранён в системе и отправлен на модерацию

	POST-2. Отзыв отображается на странице товара (если сразу публикуется)	
Основной сценарий		
Шаг	Действие	
1	Пользователь заходит в свой личный кабинет или историю заказов	
2	Пользователь выбирает заказ и переходит к карточке товара	
3	Пользователь нажимает кнопку "Оставить отзыв"	
4	Система отображает форму для написания отзыва	
5	Пользователь заполняет текст отзыва, указывает рейтинг	
6	Пользователь отправляет отзыв	
7	Система сохраняет отзыв и уведомляет пользователя об успешной отправке	
Альтернативный сценарий 1 — Отзыв отправлен на модерацию		
Шаг для замены	Название шага	Действие
7	7а	Система отправляет уведомление пользователю "Ваш отзыв находится на проверке"
Исключительный сценарий — Пользователь отправил пустой отзыв		
Шаг для замены	Название шага	Действие
6	6а	Система показывает сообщение об ошибке "Пожалуйста, введите текст отзыва".

Описание	Этот Use Case описывает процесс добавления администратором нового товара в каталог интернет-магазина.		
Актор	Администратор магазина		
Предусловия	PRE-1. Администратор авторизован PRE-2. У администратора есть права на добавление и редактирование товаров		
Постусловия	POST-1. Новый товар успешно добавлен в каталог товаров POST-2. При ошибке система информирует администратора		
Основной сценарий			
Шаг	Действие		
1	Администратор заходит в административную панель		
2	Администратор выбирает раздел "Товары"		
3	Администратор нажимает "Добавить новый товар"		
4	Система отображает форму для ввода информации о товаре		
5	Администратор заполняет основные поля: название, описание, цена, наличие, категория и фото		
6	Администратор сохраняет товар		
7	Система добавляет товар в каталог и отображает сообщение об успешном добавлении		
Альтернативный сценарий 1 — Администратор заполнил не все обязательные поля			
Шаг для замены	Название шага	Действие	

5	5a	Система показывает сообщение об ошибке "Пожалуйста, заполните обязательные поля"
Исключительный сценарий — Ошибка сервера при добавлении товара		
Шаг для замены	Название шага	Действие
6	6a	Система выводит сообщение "Ошибка сохранения товара. Попробуйте позже"

5)

Описание	Этот Use Case описывает процесс просмотра администратором пользовательских отзывов и принятия решения об их публикации или отклонении.
Актор	Администратор магазина
Предусловия	PRE-1. Администратор авторизован PRE-2. У администратора есть доступ к модулю управления отзывами
Постусловия	POST-1. Отзыв одобрен и опубликован или отклонен POST-2. Отзыв отображается на странице товара (если сразу публикуется)
Основной сценарий	
Шаг	Действие
1	Администратор заходит в административную панель
2	Администратор открывает раздел "Отзывы"
3	Система отображает список новых отзывов, ожидающих проверки
4	Администратор выбирает отзыв для просмотра

5	Администратор просматривает текст отзыва и другие параметры
6	Администратор принимает решение: "Одобрить" или "Отклонить" отзыв
7	Система применяет решение: публикует отзыв или удаляет его из публичного доступа

Альтернативный сценарий 1 — Отзыв нарушает правила

Шаг для замены	Название шага	Действие
6	6a	Администратор выбирает "Отклонить" и добавляет причину отказа (если требуется)

Исключительный сценарий — Ошибка сервера при попытке опубликовать или ОТКЛОНИТЬ ОТЗЫВ

Шаг для замены	Название шага	Действие
6	6b	Система показывает сообщение "Не удалось применить действие. Повторите попытку позже"

Модули системы

1. Каталог товаров

- ProductController — принимает HTTP-запросы, связанные с товарами
- ProductService — содержит бизнес-логику работы с товарами
- ProductRepository — взаимодействует с базой данных для операций над товарами

2. Корзина

- CartController — принимает запросы для работы с корзиной
- CartService — реализует бизнес-логику управления корзиной

- CartRepository — управляет сохранением состояния корзины в базе данных

3. Оформление заказа

- OrderController — принимает запросы на создание заказов и отображение их статуса
- OrderService — обрабатывает оформление заказа
- OrderRepository — сохраняет заказы в базе данных, обновляет их статус

4. Отзывы

- ReviewController — обрабатывает запросы на отправку и просмотр отзывов
- ReviewService — реализует проверку отзывов, работу с модерацией и отображением отзывов
- ReviewRepository — сохраняет отзывы в базе данных, извлекает их по запросу

5. Авторизация и пользователи

- AuthController — принимает запросы на вход в систему, регистрацию и выход
- AuthService — обрабатывает проверку паролей, генерацию токенов или работу с сессиями
- UserRepository — сохраняет и извлекает информацию о пользователях

6. Административная панель

- AdminController — обрабатывает запросы администратора
- AdminService — реализует бизнес-логику, связанную с административными действиями

Уровень 1. Диаграмма контекста

```
workspace {

  model {
    user = person "Пользователь" {
      description "Покупает товары, оформляет заказы, оставляет отзывы."
      tags "User"
    }

    admin = person "Администратор" {
      description "Управляет товарами, модерирует отзывы через административную панель."
    }
  }
}
```

```

tags "Admin"
}

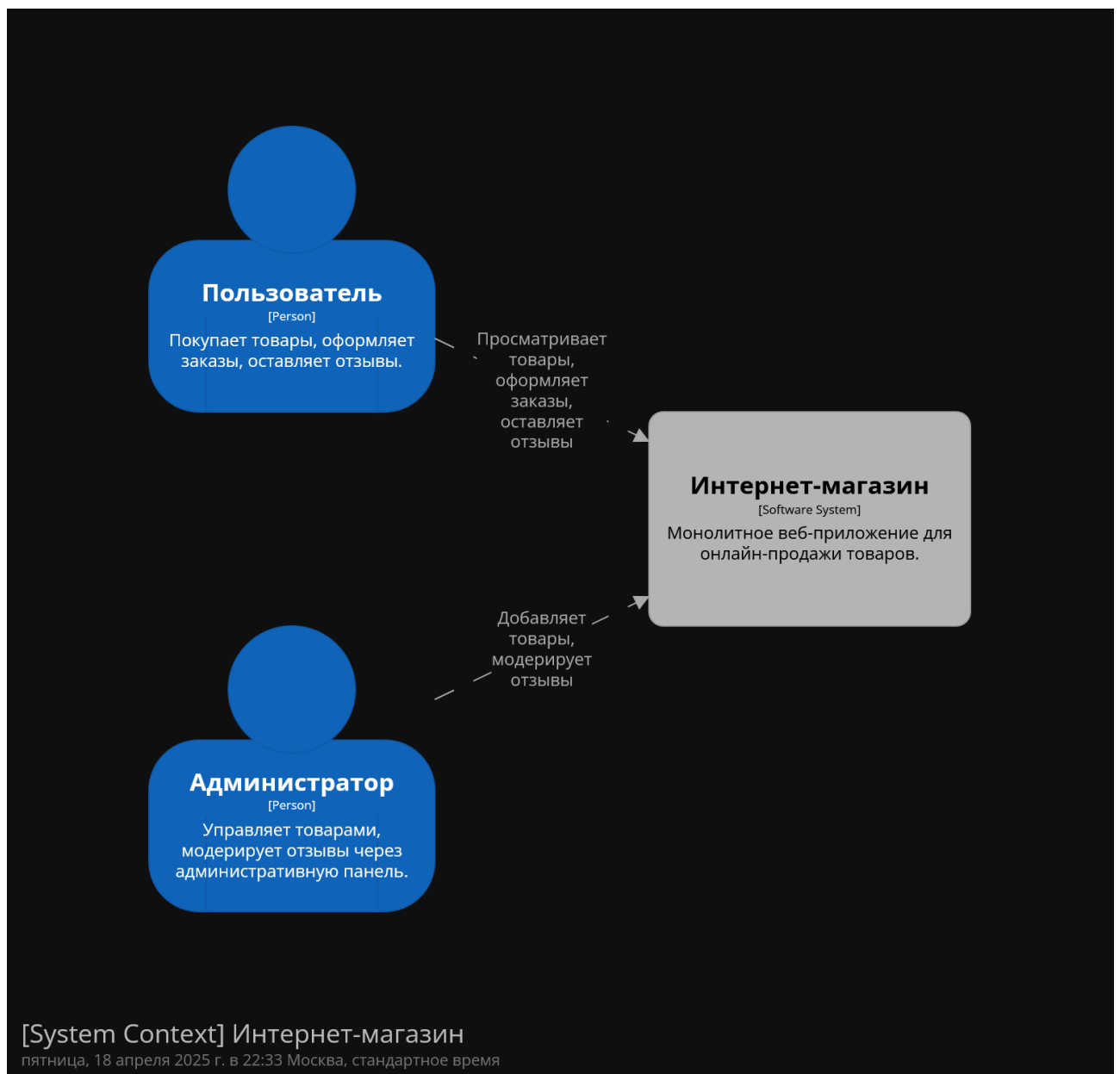
internetShop = softwareSystem "Интернет-магазин" {
description "Монолитное веб-приложение для онлайн-продажи товаров."
tags "System"
}

user -> internetShop "Просматривает товары, оформляет заказы,
оставляет отзывы"
admin -> internetShop "Добавляет товары, модерирует отзывы"
}

views {
  systemContext internetShop {
    include *
    autolayout lr
    animation user, internetShop, admin
  }
}

styles {
  element "User" {
    shape person
    background "#1168bd"
    color "#ffffff"
  }
  element "Admin" {
    shape person
    background "#1168bd"
    color "#ffffff"
  }
  element "System" {
    shape roundedbox
    background "#b4b4b4"
    color "#000000"
  }
}
}
}
}

```



Уровень 2. Диаграмма контейнеров

```
workspace {  
  model {  
    user = person "Пользователь" {  
      description "Покупает товары, оформляет заказы, оставляет отзывы."  
      tags "User"  
    }  
  
    admin = person "Администратор" {  
      description "Управляет товарами и отзывами через административную  
панель."  
      tags "Admin"  
    }  
  }  
}
```

```

internetShop = softwareSystem "Интернет-магазин" {
  description "Монолитное веб-приложение для онлайн-продажи товаров."
  tags "System"

  webApp = container "Web Application" {
    description "Обрабатывает все пользовательские и административные
запросы."
    technology "Node.js / NestJS / Express.js"
    tags "WebApp"
  }

  database = container "Database" {
    description "Хранит данные о товарах, заказах, пользователях и
отзывах."
    technology "PostgreSQL / MySQL"
    tags "Database"
  }

  user -> webApp "Использует через браузер"
  admin -> webApp "Управляет через браузер"
  webApp -> database "Чтение и запись данных"
}

views {
  container internetShop {
    include *
    autolayout lr
    animation user, webApp, database, admin
  }
}

styles {
  element "User" {
    shape person
    background "#1168bd"
    color "#ffffff"
  }
  element "Admin" {
    shape person
    background "#1168bd"
    color "#ffffff"
  }
  element "System" {

```

```

    shape roundedbox
    background "#d4d4d4"
    color "#000000"
  }
  element "WebApp" {
    shape roundedbox
    background "#d4d4d4"
    color "#000000"
  }
  element "Database" {
    shape cylinder
    background "#d4d4d4"
    color "#000000"
  }
}
}
}

```

