**Информационная система интернет-магазина. Заказы, клиенты, доставка, партнеры.**

**Содержание**

[**1. Виденье 3**](#_Toc185554863)

[**2. Use Case диаграмма 6**](#_Toc185554867)

[**3. Диаграмма программных классов 8**](#_Toc185554870)

[**4. Диаграмма последовательностей 13**](#_Toc185554887)

[**5. Диаграмма коопераций 15**](#_Toc185554890)

[**6. Диаграмма состояний 17**](#_Toc185554893)

[**7. Диаграмма активности 20**](#_Toc185554897)

[**8. Диаграмма компонентов 23**](#_Toc185554899)

[**9. Диаграмма развёртывания 25**](#_Toc185554901)

# **Виденье**

**Краткая информация о компании (профиль клиента):**

* Название компании: “FoxyGames”.
* Описание: Компания занимается продажей товаров для животных через интернет, включая доставку.
* Рынок: В основном работая с частными покупателями.
* Партнеры: Логистические компании, поставщики товаров.

**Цели проекта:**

* Автоматизация процессов обработки заказов.
* Оптимизация взаимодействия с клиентами.
* Улучшение управления доставкой товаров.
* Подключение и управление партнерами (поставщиками и логистикой).

**Задачи проекта:**

* Разработка и внедрение системы управления заказами
* Интеграция с логистическими партнерами
* Разработка модуля управления партнерами
* Обновление системы управления товарами
* Внедрение системы обработки платежей
* Обеспечение взаимодействия между отделами
* Тестирование и отладка системы
* Подготовка документации
* Запуск и поддержка системы

**Подразделения и пользователи системы:**

* + Отдел продаж (работа с клиентами и заказами).
  + Логистический отдел (организация доставки).
  + Отдел партнёрских отношений (работа с поставщиками и партнёрами по доставке).
  + IT-отдел (поддержка и настройка системы).
  + Финансовый отдел (обработка платежей и финансовый контроль).

**Подразделения:**

* + Интернет-магазин внедряет ИС для отдела продаж, логистики и финансов.

**Границы проекта:**

* + Система будет включать в себя управление заказами клиентов, работу с базой клиентов, интеграцию с доставкой и партнерами.
  + В проекте не рассматриваются сложные аналитические модули или CRM-системы, только базовые функции интернет-магазина.

**Список объектов системы:**

* **Заказы:** данные о заказах (товары, количество, цена, статус).
* **Клиенты:** информация о клиентах (имя, контактные данные, история заказов).
* **Доставка:** информация о доставках (статус, дата доставки, партнер по доставке).
* **Партнёры:** данные о партнерах (контакты, условия сотрудничества, логистика).
* **Товары:** ассортимент товаров (название, категория, цена, наличие на складе).
* **Платежи:** информация о проведенных оплатах, возвратах.

**Список функций системы:**

* **Управление заказами:** функции для оформления, изменения и отслеживания заказов. *Используемые объекты*: *заказы, товары, клиенты, платежи.*
* **Управление клиентами:** хранение и обновление информации о клиентах. *Используемые объекты: клиенты, заказы.*
* **Управление доставкой:** отслеживание статусов доставки, взаимодействие с партнерами. *Используемые объекты: доставка, партнеры, заказы.*
* **Управление партнерами:** подключение и взаимодействие с партнерами. *Используемые объекты: партнеры, доставка.*
* **Управление товарами:** обновление данных о наличии товаров и ценах. *Используемые объекты: товары, заказы.*

**Супертипы объектов:**

#### 1. ****Пользователи****

Пользователи— это супертип, который объединяет объекты "Клиенты" и "Партнеры". Эти объекты имеют много общих атрибутов и операций, таких как контактные данные, условия взаимодействия, платежные данные и историю взаимодействий.

* **Клиенты** — это физические лица или компании, которые оформляют заказы в интернет-магазине. Для них сохраняется информация, такая как:
  + Имя/название организации.
  + Контактная информация (телефон, электронная почта, адрес).
  + История заказов.
* **Партнеры** — это поставщики товаров и логистические компании. Для них также сохраняются контактные данные, но с дополнительной информацией о коммерческих условиях и логистических маршрутах:
  + Название компании.
  + Контактная информация (телефон, электронная почта, юридический адрес).
  + Условия сотрудничества (цены на услуги доставки, сроки поставок).

##### Общие атрибуты (свойства):

* Идентификационные данные (ID).
* Контактные данные (адрес, телефон, email).
* История взаимодействий (история заказов для клиентов и история поставок/услуг для партнеров).

##### Общие функции для работы с пользователями:

* Добавление новых контрагентов.
* Обновление контактной информации.
* Анализ истории взаимодействий.
* Оформление и обработка платежей (как со стороны клиентов, так и партнёров).

#### 2. Транзакции

Транзакции — это супертип, который объединяет объекты "Заказы" и "Платежи". Оба объекта связаны с денежными операциями и обработкой товаров, и оба требуют отслеживания и управления статусов.

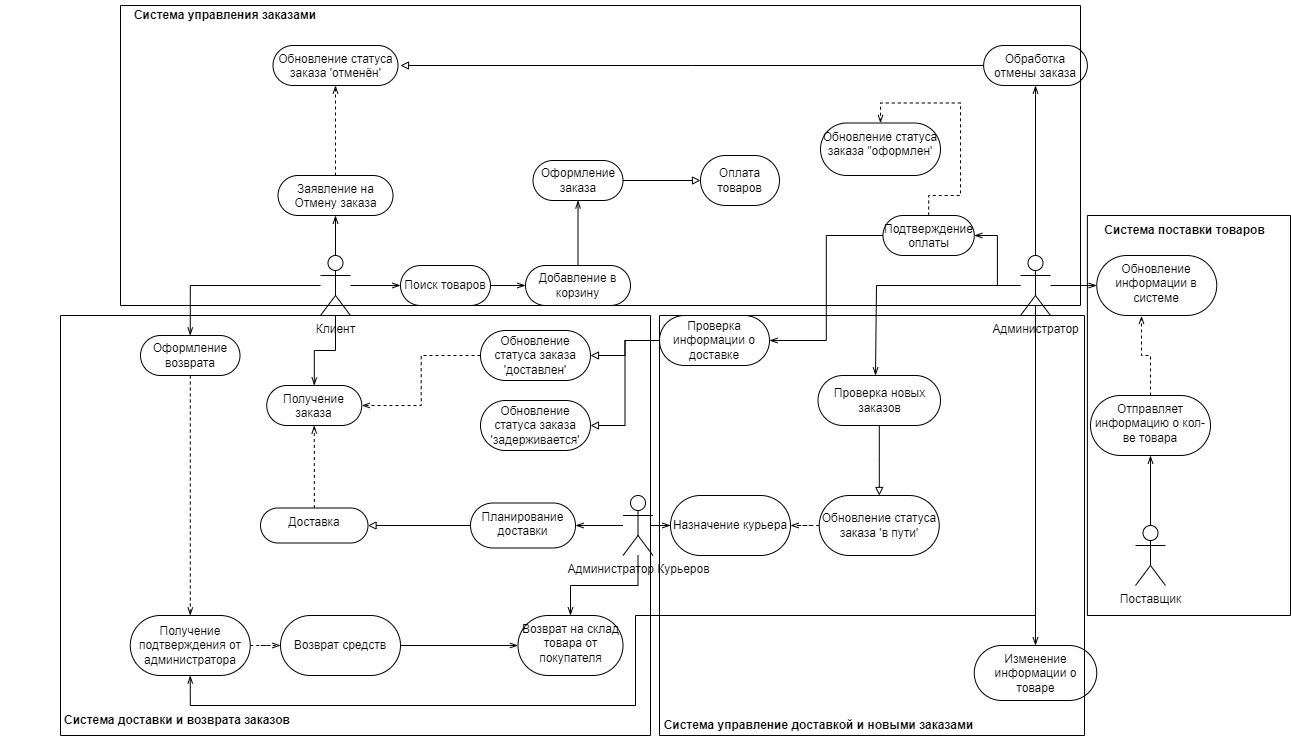
* **Заказы** — это информация о том, что клиент заказал (какие товары, количество, цена), а также статус выполнения заказа (обрабатывается, отправлен, доставлен).
* **Платежи** — это информация о финансовых операциях, связанных с заказами, включая подтверждение оплаты, возвраты, тип оплаты (онлайн-платеж, наложенный платеж и т.д.).

##### Общие атрибуты (свойства транзакций):

* Идентификатор транзакции.
* Дата создания/проведения транзакции.
* Статус (новая, оплачена, отменена, возвращена).
* Сумма операции.

##### Общие функции для работы с транзакциями:

* Создание транзакции (создание заказа или нового платежа).
* Изменение статуса транзакции (оплата, отмена, возврат).
* Анализ и мониторинг транзакций (для выявления проблемных заказов, возвратов, долгов).

1. **Use Case диаграмма**

### Акторы (участники системы):

* **Клиент**: покупатели, оформляющие заказы в интернет-магазине.
* **Администратор**: сотрудники магазина, которые управляют заказами, товарами и платежами.
* **Администратор Курьеров**: человек, отвечающие за доставку заказов клиентам.
* **Поставщик**: компании, предоставляющие товары интернет-магазину.

### Прецеденты:

#### 1. Заказ товаров

* **Акторы**: Клиент
* **Описание**: Клиент выбирает товары, добавляет их в корзину, оформляет заказ и оплачивает его

#### 2. Управление товарами

* **Акторы**: Администратор
* **Описание**: Администратор проверяет новые товары, редактирует существующие и управляет их наличием.

#### 3. Обработка заказов

* **Акторы**: Администратор
* **Описание**: Администратор проверяет поступившие заказы, обновляет их статус и связывается с Администратором Курьеров для доставки.

#### 4. Управление доставкой

* **Акторы**: Администратор Курьеров, Администратор
* **Описание**: Администратор Курьер получает информацию о заказах для доставки, планирует маршрут и доставляет.

#### 6. Управление платежами

* **Акторы**: Администратор, Клиент.
* **Описание**: Клиенты совершают платежи за заказы, а администратор отслеживает и подтверждает оплату, после чего меняет статус заказа.

#### 7. Управление поставками

* **Акторы**: Поставщик, Администратор.
* **Описание**: Поставщик предоставляет интернет-магазину товары, а администратор управляет процессом получения и обновления информации о товарах.

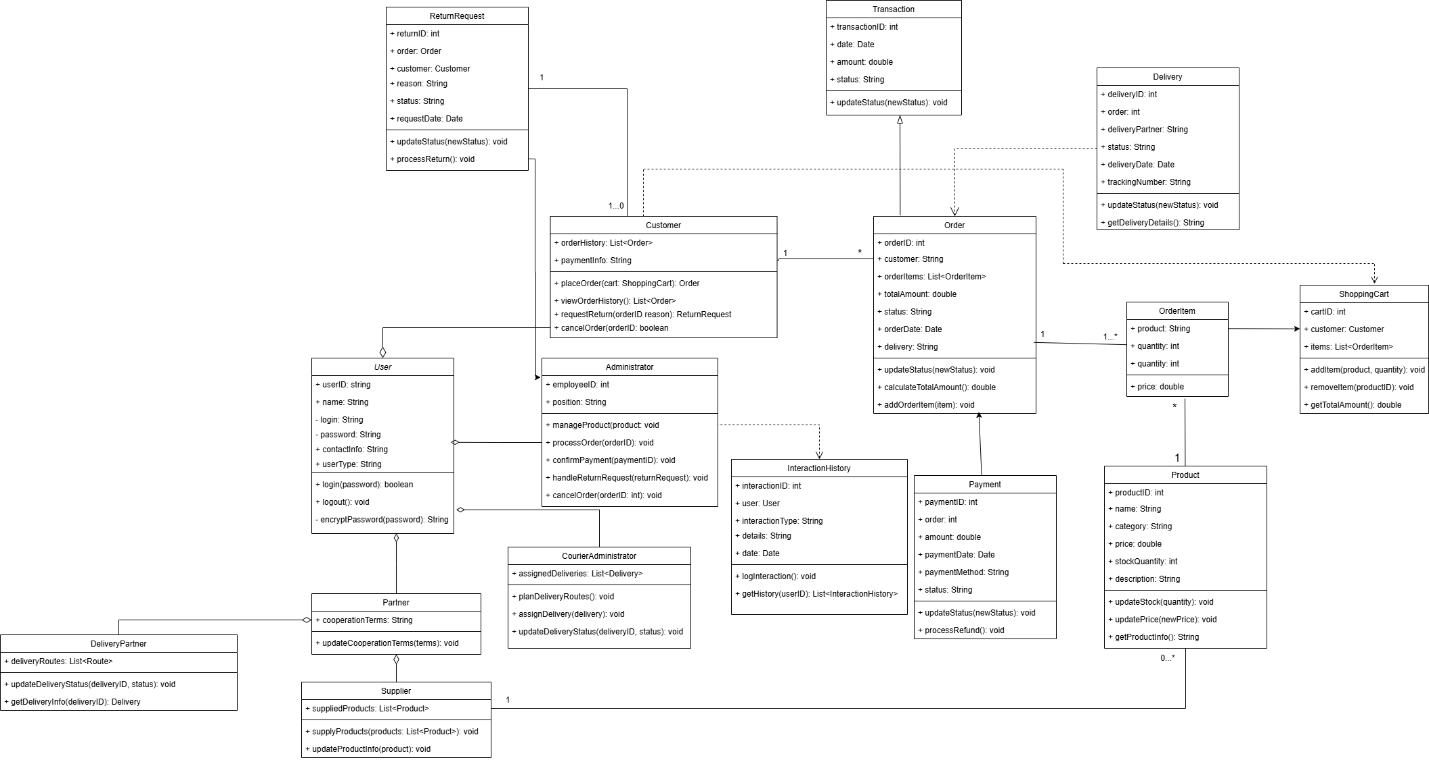
**8.** **Оформление возврата**

* **Акторы:** Администратор, Администратор Курьеров, Клиент
* **Описание:** Клиент оформляет возврат, получает подтверждение возврата от Администратора, после чего Курьер забирает заказ и возвращает его на склад.

**9. Оформление отмены заказа**

* **Акторы:** Клиент, Администратор
* **Описание:** Клиент может отменить заказ, если ошибся или передумал. Администратор отменяет его и меняет статус заказа.

# Диаграмма программных классов



### 1. ****User (Пользователь)****

**Описание:**

* Базовый класс для всех типов пользователей системы.
* Определяет общие атрибуты и методы, которые наследуются специализированными пользователями.

**Роль:**

* Обеспечивает общую структуру для различных типов пользователей, таких как клиенты, партнёры и администраторы.
* Содержит информацию о пользователе и методы для авторизации.

### 2. ****Customer (Клиент)****

**Описание:**

* Представляет покупателей, которые совершают покупки в интернет-магазине.

**Роль:**

* Позволяет клиентам:
  + Просматривать и выбирать товары.
  + Добавлять товары в корзину (ShoppingCart).
  + Оформлять заказы (Order).
  + Просматривать историю заказов.
  + Отменять заказы и запрашивать возвраты (ReturnRequest).

### 3. ****Partner (Партнёр)****

**Описание:**

* Базовый класс для деловых партнёров интернет-магазина, таких как поставщики и логистические партнёры.

**Роль:**

* Определяет общие атрибуты и методы для взаимодействия с партнёрами.
* Управляет условиями сотрудничества.

### 4. ****Supplier (Поставщик)****

**Описание:**

* Компании, которые поставляют товары для интернет-магазина.

**Роль:**

* Управляют списком поставляемых товаров.
* Поставляют товары в магазин.
* Обновляют информацию о товарах (например, цены, описание, наличие).

### 5. ****DeliveryPartner (Логистический партнёр)****

**Описание:**

* Логистические компании, отвечающие за доставку заказов клиентам.

**Роль:**

* Управляют маршрутами доставки.
* Обновляют статус доставок.
* Предоставляют информацию о доставке заказов.

### 6. ****Administrator (Администратор)****

**Описание:**

* Сотрудники интернет-магазина, ответственные за управление товарами, заказами и платежами.

**Роль:**

* Управляют ассортиментом товаров.
* Обрабатывают заказы клиентов.
* Подтверждают и контролируют платежи.
* Обрабатывают запросы на возврат товаров (ReturnRequest).
* Могут отменять заказы по запросу клиентов.

### 7. ****CourierAdministrator (Администратор Курьеров)****

**Описание:**

* Сотрудники, ответственные за организацию и контроль процесса доставки заказов.

**Роль:**

* Планируют маршруты доставки.
* Назначают доставки курьерам.
* Обновляют статусы доставок.
* Координируют работу с логистическими партнёрами.

### 8. ****Product (Товар)****

**Описание:**

* Представляет товары, доступные для продажи в интернет-магазине.

**Роль:**

* Хранит информацию о товаре, такую как название, категория, цена, количество на складе и описание.
* Обеспечивает методы для обновления информации о товаре, включая количество на складе и цену.
* Предоставляет информацию о товаре клиентам и сотрудникам магазина.

### 9. ****Order (Заказ)****

**Описание:**

* Содержит информацию о заказах, оформленных клиентами.

**Роль:**

* Хранит детали заказа, включая список товаров (OrderItem), общую сумму, статус и дату оформления.
* Связан с клиентом, который сделал заказ, и с информацией о доставке.
* Обеспечивает методы для обновления статуса заказа и расчёта общей суммы.

### 10. ****OrderItem (Позиция заказа)****

**Описание:**

* Представляет отдельный товар в составе заказа.

**Роль:**

* Хранит информацию о товаре, количестве и цене за единицу.
* Используется для расчёта суммы по позиции и общей суммы заказа.

### 11. ****Payment (Платёж)****

**Описание:**

* Содержит информацию о платежах, связанных с заказами.

**Роль:**

* Хранит детали платежа, включая сумму, дату, способ оплаты и статус.
* Связан с конкретным заказом.
* Обеспечивает методы для обновления статуса платежа и обработки возвратов средств.

### 12. ****Delivery (Доставка)****

**Описание:**

* Хранит данные о доставке конкретного заказа.

**Роль:**

* Содержит информацию о статусе доставки, дате доставки, номере отслеживания и логистическом партнёре.
* Обеспечивает методы для обновления статуса доставки и получения деталей доставки.

### 13. ****ShoppingCart (Корзина покупок)****

**Описание:**

* Позволяет клиентам собирать товары перед оформлением заказа.

**Роль:**

* Хранит список выбранных товаров (OrderItem) для будущего заказа.
* Обеспечивает методы для добавления и удаления товаров из корзины.
* Рассчитывает общую сумму товаров в корзине.

### 14. ****Transaction (Транзакция)****

**Описание:**

* Базовый класс для финансовых операций в системе, таких как заказы и платежи.

**Роль:**

* Определяет общие атрибуты и методы для транзакций, включая идентификатор, дату, сумму и статус.
* Обеспечивает метод для обновления статуса транзакции.

### 15. ****ReturnRequest (Запрос на возврат)****

**Описание:**

* Обрабатывает запросы клиентов на возврат товаров.

**Роль:**

* Хранит информацию о запросе на возврат, включая причину, статус и дату запроса.
* Связан с заказом и клиентом, инициировавшим возврат.
* Обеспечивает методы для обновления статуса запроса и обработки возврата.

### 16. ****InteractionHistory (История взаимодействий)****

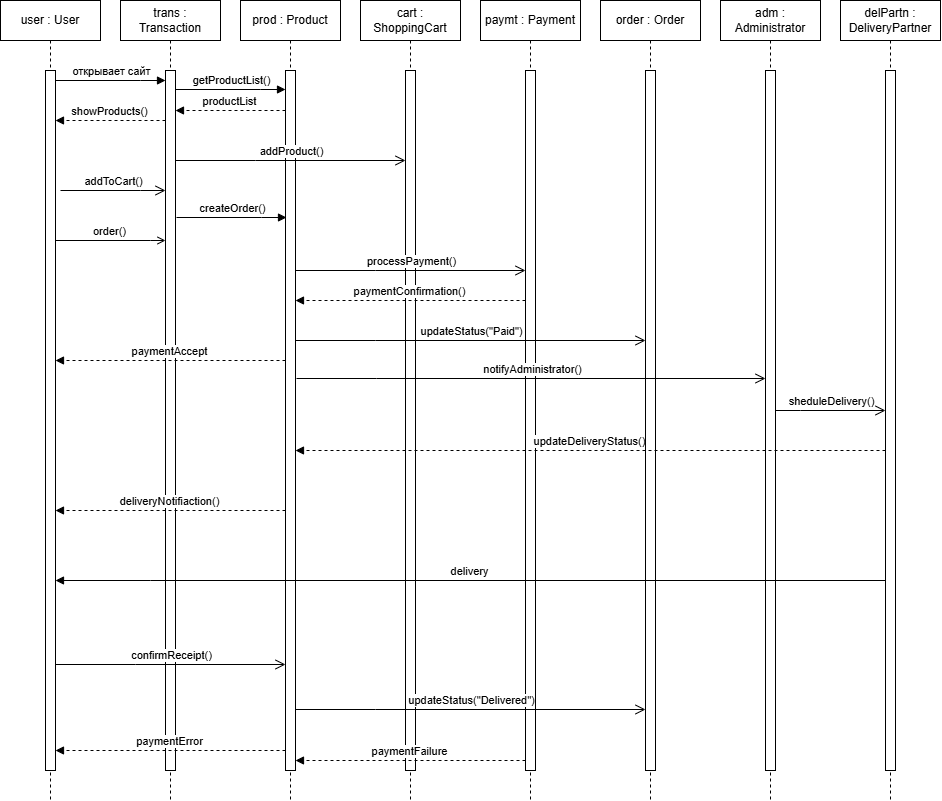
**Описание:**

* Хранит историю взаимодействий с клиентами и партнёрами.

**Роль:**

* Записывает различные события и действия в системе, такие как заказы, платежи, поставки и т.д.
* Используется администраторами для анализа поведения пользователей и улучшения сервисов.
* Обеспечивает методы для записи взаимодействий и получения истории по пользователю.

# Диаграмма последовательностей



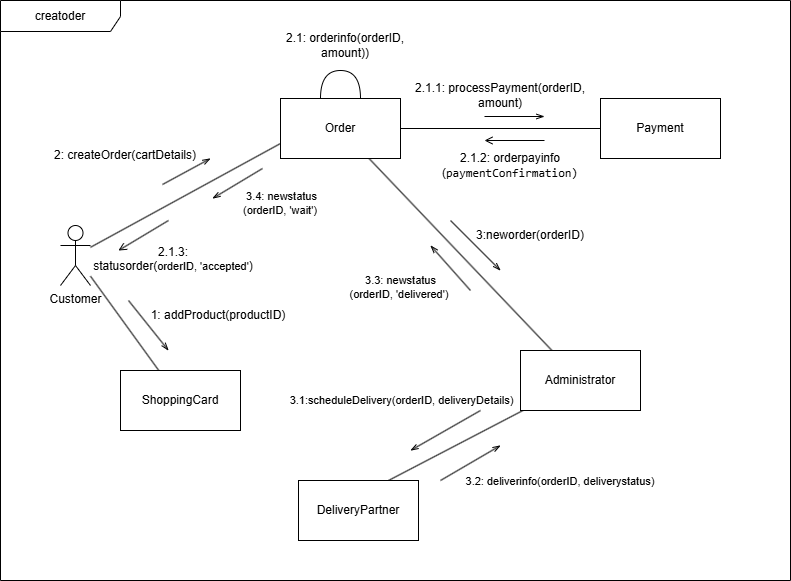
**Участники (Линии жизни):**

* **Клиент** (User)
* **Транзакция**(Transaction)
* **Корзина покупок** (ShoppingCart)
* **Администратор** (Administrator)
* **Товар** (Product)
* **Заказ** (Order)
* **Платёж** (Payment)
* **Логистический партнёр** (DeliveryPartner)

**Последовательность взаимодействий:**

1. **Клиент** открывает веб-сайт интернет-магазина через **Интерфейс пользователя**.
2. **Клиент** просматривает список **Товаров**:
   * **Интерфейс пользователя** запрашивает список товаров у **Системы интернет-магазина** (getProductList()).
   * **Система интернет-магазина** возвращает список товаров в **Интерфейс пользователя**.
3. **Клиент** выбирает товар и добавляет его в **Корзину покупок**:
   * **Клиент** вызывает addToCart(productID) в **Интерфейсе пользователя**.
   * **Интерфейс пользователя** обновляет **Корзину покупок** (ShoppingCart.addProduct(productID)).
4. **Клиент** просматривает **Корзину покупок** и инициирует оформление заказа:
   * **Клиент** нажимает кнопку "Оформить заказ".
   * **Интерфейс пользователя** отправляет данные **Корзины** в **Систему интернет-магазина** (createOrder(cartDetails)).
5. **Система интернет-магазина** создает новый **Заказ**:
   * Создается объект **Order** с уникальным orderID, привязанным к **Клиенту** и содержимому **Корзины**.
6. **Система интернет-магазина** инициализирует процесс оплаты:
   * Отправляет запрос в **Систему платежей** (PaymentSystem.processPayment(orderID, amount)).
7. **Система платежей** обрабатывает платеж:
   * Проверяет платежные данные **Клиента**.
   * Если платеж успешен, возвращает подтверждение (paymentConfirmation).
   * Создается объект **Payment** с деталями платежа.
8. **Система интернет-магазина** получает подтверждение платежа:
   * Обновляет статус **Заказа** на "Оплачен" (Order.updateStatus("Paid")).
9. **Система интернет-магазина** уведомляет **Администратора** о новом заказе:
   * Отправляет сообщение или создает задачу для **Администратора**.
10. **Администратор** проверяет **Заказ** и инициирует доставку:
    * Выбирает **Логистического партнёра** для доставки.
    * Передает информацию о заказе и доставке в **DeliveryPartner** (DeliveryPartner.scheduleDelivery(orderID, deliveryDetails)).
11. **Логистический партнёр** создает запись о **Доставке**:
    * Создается объект **Delivery** с deliveryID, связанным с **Заказом** и **Клиентом**.
    * Планируется маршрут доставки.
12. **Система интернет-магазина** обновляет статус **Заказа** на "В процессе доставки" (Order.updateStatus("Shipping")).
13. **Клиент** получает уведомление о статусе заказа:
    * **Интерфейс пользователя** отображает обновленный статус заказа и информацию о доставке.
14. **Логистический партнёр** осуществляет доставку **Заказа**:
    * **Курьер** доставляет товар по адресу **Клиента**.
15. **Клиент** получает **Заказ** и подтверждает получение:
    * Подписывает документы или подтверждает в **Интерфейсе пользователя** (confirmReceipt(orderID)).
16. **Логистический партнёр** обновляет статус **Доставки** на "Доставлено".
17. **Система интернет-магазина** обновляет статус **Заказа** на "Доставлено" (Order.updateStatus("Delivered")).
18. **Клиент** может оставить отзыв о товаре и обслуживании.

# 5. Диаграмма коопераций



**Участники (объекты):**

1. **Creator (createOrder)** — инициирует процесс создания заказа.
2. **ShoppingCart** — хранит выбранные товары перед оформлением заказа.
3. **Order** — управляет процессом оформления и обновления статуса заказа.
4. **Payment** — обрабатывает оплату заказа.
5. **Administrator** — координирует действия, связанные с доставкой.
6. **DeliveryPartner** — логистический партнер, отвечающий за доставку заказа.

**Взаимодействие между объектами:**

1. **Создание заказа:**
   * **Creator** вызывает метод createOrder для создания нового заказа.
   * **Order** получает заказ и обновляет его статус на **"accepted"** (принят).
2. **Планирование доставки:**
   * **Administrator** вызывает метод scheduleDelivery(orderID, deliveryDetails) у **DeliveryPartner** для планирования доставки.
   * **DeliveryPartner** подтверждает получение запроса на доставку и отправляет статус с помощью deliverInfo(orderID, deliveryStatus).
3. **Обновление статуса заказа:**
   * **Order** обновляет статус доставки на **"wait"** (в ожидании доставки) и затем на **"delivered"** (доставлено) при получении подтверждения от **DeliveryPartner**.
   * Это отражается через методы newStatus(orderID, 'wait') и newStatus(orderID, 'delivered').

**Итоговый процесс:**

1. Создание заказа.
2. Планирование доставки администратором с передачей данных логистическому партнеру.
3. Обновление статуса заказа на "в ожидании доставки" и далее на "доставлено".

Диаграмма показывает последовательность шагов и взаимодействие объектов, необходимых для выполнения заказа и доставки.

# 6. Диаграмма состояний



**Описание диаграммы состояний для процесса заказа:**

**Состояния и переходы:**

**1. Создание заказа**

* Это начальное состояние процесса обработки заказа.
* Заказ может быть создан пользователем через интерфейс или системой автоматически.
* **Переходы из этого состояния:**
  + Если заказ принят без изменений, он переходит в состояние **"Заказ принят"**.
  + Если необходимо внести изменения, заказ направляется в состояние **"Изменение заказа"**.

**2. Изменение заказа**

* На этом этапе в заказ вносятся правки, инициированные клиентом или компанией.
* Это состояние особенно важно для настройки заказа под индивидуальные запросы.
* **Переходы из этого состояния:**
  + Если возникают разногласия между сторонами (например, клиентом и системой), заказ переходит в состояние **"Уточнение по заказу"**.
  + Если разногласий нет, заказ возвращается в состояние **"Заказ принят"**.

**3. Уточнение по заказу**

* Это состояние наступает, если заказ требует дополнительного согласования между клиентом и сотрудниками компании.
* Возможные причины уточнений: неточности в деталях заказа, необходимость изменения параметров доставки или оплаты.
* **Переходы из этого состояния:**
  + После успешного уточнения заказ переходит в состояние **"Заказ согласован с клиентом"**.
  + Если уточнение затягивается, заказ может быть временно приостановлен.

**4. Заказ согласован с клиентом**

* В этом состоянии заказ полностью утвержден клиентом и компанией.
* Клиент подтверждает детали заказа, а компания готовится к его выполнению.
* **Переходы из этого состояния:**
  + Если клиент оплачивает заказ, он переходит в состояние **"Оплачен заказ"**.

**5. Оплачен заказ**

* Заказ успешно оплачен клиентом. Это важный этап, так как только после оплаты заказ передается на исполнение.
* **Переходы из этого состояния:**
  + После подтверждения оплаты заказ передается в состояние **"Передача заказа отделу доставки"**.

**6. Передача заказа отделу доставки**

* На этом этапе заказ передается логистическому отделу для дальнейшей обработки.
* **Переходы из этого состояния:**
  + После получения заказа логистическим отделом он переходит в состояние **"Заказ у отдела доставки"**.

**7. Заказ у отдела доставки**

* В этом состоянии заказ находится в процессе транспортировки.
* Возможные действия: упаковка, передача курьеру, отправка уведомления клиенту.
* **Переходы из этого состояния:**
  + Если заказ успешно доставлен в пункт выдачи или на адрес клиента, он переходит в состояние **"Заказ доставлен"**.

**8. Заказ доставлен**

* Заказ доставлен в пункт выдачи или на адрес клиента.
* Клиент уведомляется о доставке.
* **Переходы из этого состояния:**
  + Если клиент получает заказ, он переходит в состояние **"Заказ выполнен"**.

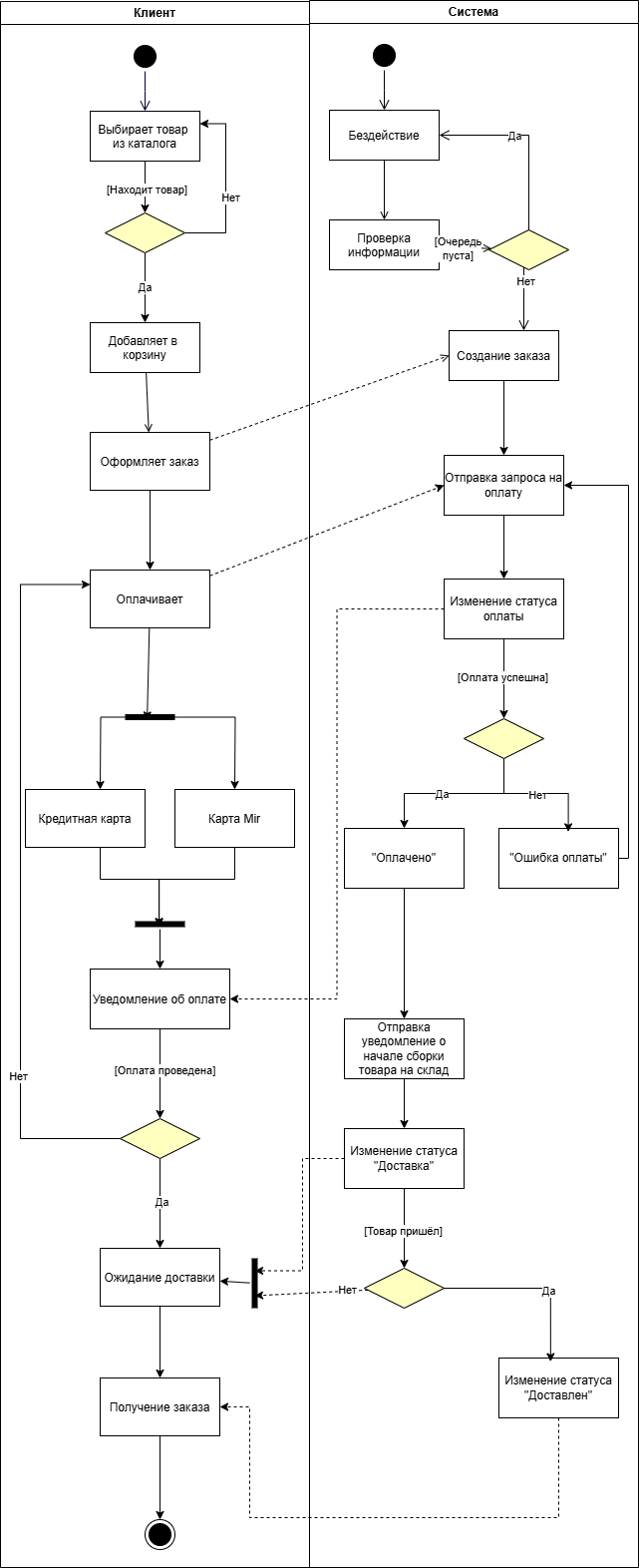
**9. Заказ выполнен**

* Это конечное состояние процесса заказа.
* Клиент успешно получил заказ, и все обязательства компании выполнены.
* На этом этапе процесс обработки заказа завершается.

**Ключевые особенности:**

1. **Разногласия и уточнения:**  
   Диаграмма учитывает возможные конфликтные ситуации, такие как разногласия по заказу, и предлагает пути их разрешения через уточнение.
2. **Последовательность:**  
   Процесс обработки заказа четко структурирован и охватывает все этапы: от создания заказа до его выполнения.
3. **Участие клиента:**  
   Клиент активно участвует на этапах создания, уточнения и подтверждения заказа, что повышает прозрачность процесса.
4. **Гибкость:**  
   Возможность возврата на предыдущие этапы (например, в случае необходимости изменений или уточнений) делает процесс адаптивным к различным ситуациям.

# 7. Диаграмма активности



**Swimlane 1: Клиент**

1. **Выбор товара из каталога:**
   * Клиент начинает процесс поиска и выбора товара.
2. **Проверка наличия товара:**
   * Условие (Decision Node):
     + Если товар найден → добавляется в корзину.
     + Если товар **не найден**, процесс завершается.
3. **Добавление товара в корзину:**
   * Товар успешно добавляется в корзину.
4. **Оформление заказа:**
   * Клиент инициирует создание заказа.
5. **Оплата:**
   * Клиент производит оплату через выбранный способ:
     + **Кредитная карта**
     + **Карта Mir**
6. **Ожидание доставки:**
   * После успешной оплаты клиент ожидает доставки заказа.
7. **Получение заказа:**
   * Завершение процесса при доставке товара.

**Swimlane 2: Система интернет-магазина**

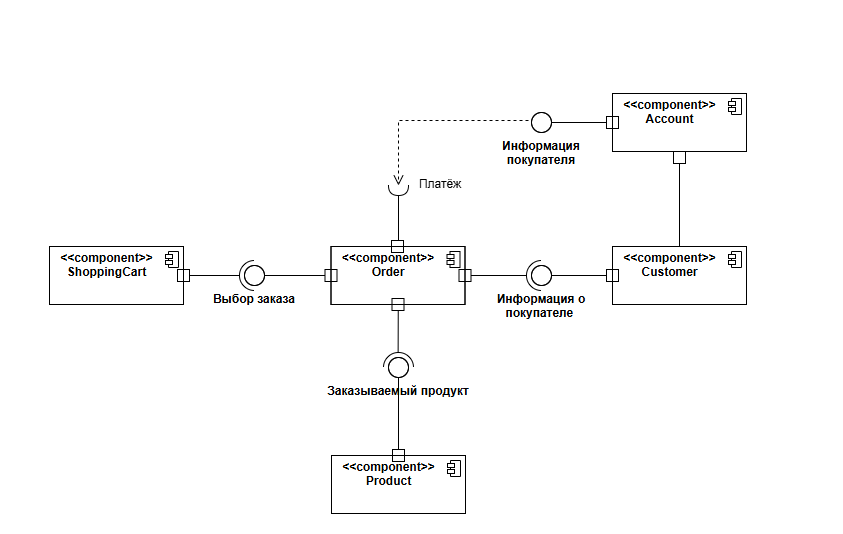
1. **Бездействие (ожидание запроса):**
   * Система ждет активности от клиента.
2. **Проверка информации:**
   * Проверка данных о товаре и состоянии очереди:
     + Если очередь пустая → завершение шага.
     + Иначе → создание заказа.
3. **Создание заказа:**
   * Система создает заказ на основе содержимого корзины клиента.
4. **Отправка запроса на оплату:**
   * Система отправляет запрос в платежную систему.
5. **Изменение статуса оплаты:**
   * Условие (Decision Node):
     + Если оплата успешна → статус заказа изменяется на **"Оплачен"**.
     + Если оплата неуспешна → ошибка оплаты, возврат клиенту.
6. **Уведомление об оплате:**
   * Клиент получает уведомление о проведении оплаты.
7. **Начало сборки товара:**
   * Система отправляет уведомление о начале сборки заказа на складе.
8. **Изменение статуса "Доставка":**
   * Статус заказа изменяется на **"Доставка"**.
9. **Проверка статуса доставки:**
   * Условие (Decision Node):
     + Если товар доставлен → статус изменяется на **"Доставлен"**.
     + Если товар **не доставлен** → продолжается мониторинг статуса.
10. **Завершение доставки:**
    * Статус заказа обновляется на **"Доставлен"**.

**Конец процесса**

* Клиент получает заказ, и система подтверждает успешное завершение доставки.

**Итог:** Диаграмма отражает последовательность шагов, начиная от выбора товара до его получения, с четким разделением на действия **клиента** и **системы интернет-магазина**. Включены проверки условий (оплата, статус доставки) и логические переходы.

# 8. Диаграмма компонентов



1. **ShoppingCart (Корзина покупок):**
   * Содержит информацию о товарах, выбранных клиентом для покупки.
   * Связана с компонентом **Order (Заказ)** через процесс **"Выбор заказа"**, где данные из корзины передаются для оформления заказа.
2. **Order (Заказ):**
   * Обрабатывает детали заказа, такие как список продуктов и информация о покупателе.
   * Связан с компонентом **ShoppingCart (Корзина покупок)** для получения списка товаров.
   * Связан с компонентом **Product (Продукт)** через процесс **"Заказываемый продукт"**.
   * Также взаимодействует с компонентом **Customer (Клиент)** для получения данных о покупателе.
3. **Product (Продукт):**
   * Представляет товары, которые можно добавить в корзину и заказать.
   * Связан с **Order (Заказ)** для передачи информации о заказываемых продуктах.
4. **Customer (Клиент):**
   * Хранит базовую информацию о клиенте, такую как имя, контакты и историю заказов.
   * Связан с компонентом **Account (Аккаунт)**, откуда получает дополнительную информацию о покупателе.
   * Также передает данные компоненту **Order (Заказ)** через процесс **"Информация о покупателе"**.
5. **Account (Аккаунт):**
   * Содержит расширенную информацию о клиенте, такую как платежные данные, история взаимодействия с системой и уровень лояльности.
   * Связан с **Customer (Клиент)** через процесс **"Информация покупателя"**.

**Основные процессы:**

1. **Выбор заказа:**
   * Информация из корзины покупок передается в заказ для оформления.
2. **Информация о покупателе:**
   * Сведения о клиенте, включая контактные данные и предпочтения, используются для обработки заказа.
3. **Заказываемый продукт:**
   * Список товаров, добавленных в корзину, передается в заказ для дальнейшей обработки.
4. **Информация покупателя:**
   * Расширенные данные о клиенте из аккаунта используются для дополнения профиля клиента в заказе.

**Основные связи и их значения:**

1. **Связь между ShoppingCart (Корзина покупок) и Order (Заказ) через "Выбор заказа":**
   * **Описание:**
     + Информация о товарах, добавленных клиентом в корзину, передается в модуль заказа.
   * **Роль:**
     + Позволяет системе сформировать заказ на основе содержимого корзины.
2. **Связь между Order (Заказ) и Customer (Клиент) через "Информация о покупателе":**
   * **Описание:**
     + Заказ получает данные о клиенте, такие как имя, адрес доставки и контактные данные.
   * **Роль:**
     + Необходима для персонализации заказа и его последующей обработки (например, доставки или выставления счета).
3. **Связь между Customer (Клиент) и Account (Аккаунт) через "Информация покупателя":**
   * **Описание:**
     + Модуль клиента получает дополнительные данные о покупателе из его аккаунта (например, платежные реквизиты, историю заказов).
   * **Роль:**
     + Обеспечивает полное представление о клиенте для обработки заказа и персонализации услуг.
4. **Связь между Order (Заказ) и Product (Продукт) через "Заказываемый продукт":**
   * **Описание:**
     + Заказ связан с продуктами, которые были выбраны клиентом.
   * **Роль:**
     + Позволяет системе зафиксировать, какие конкретно товары включены в заказ. Это важно для формирования итогового счета и последующей доставки.

# 9. Диаграмма развёртывания

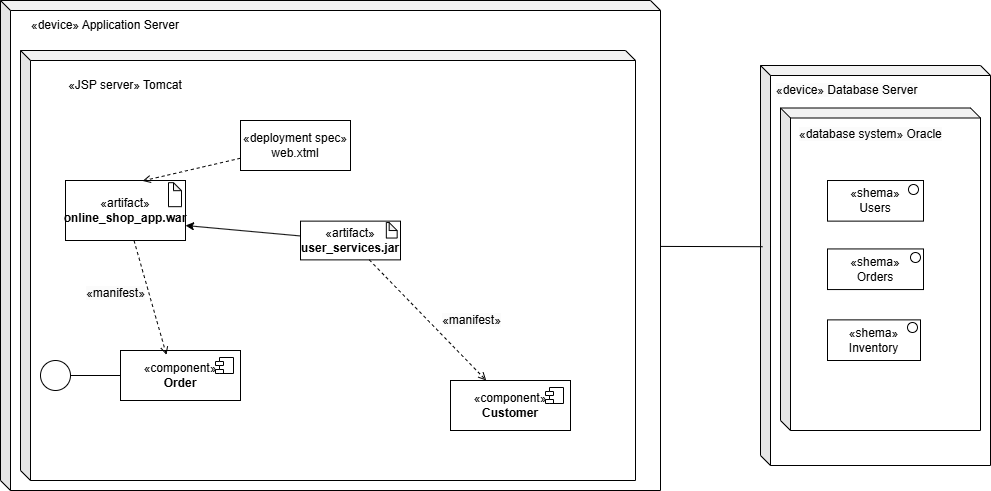


Диаграмма развёртывания показывает, как программные компоненты и артефакты развернуты в инфраструктуре. В данном случае представлены серверы, компоненты приложения и их взаимодействие с базой данных.

**Участники диаграммы:**

#### **1. Application Server (Сервер приложений)**

* **Описание:**  
  Это сервер, на котором размещены и выполняются компоненты веб-приложения.
* **Технология:**  
  Используется сервер JSP — **Tomcat**.
* **Артефакты:**
  1. **online\_shop\_app.war:**
     + Веб-архив приложения интернет-магазина.
     + Содержит основные компоненты и спецификации развертывания (например, web.xml).
  2. **user\_services.jar:**
     + Архив, содержащий логические модули для работы с пользователями (например, управление учетными записями, аутентификация).
* **Компоненты:**
  1. **Order (Заказы):**
     + Отвечает за обработку заказов, взаимодействуя с данными заказа.
  2. **Customer (Клиенты):**
     + Управляет данными о клиентах, такими как регистрация и обновление информации.
* **Связи:**
  1. **online\_shop\_app.war** взаимодействует с Order и Customer через спецификации развертывания (манифесты).
  2. **user\_services.jar** предоставляет функционал для обработки операций пользователей и связан с Customer.

#### **2. Database Server (Сервер базы данных)**

* **Описание:**  
  Содержит базу данных, которая используется для хранения данных приложения.
* **Технология:**  
  Используется система управления базами данных **Oracle**.
* **Схемы базы данных:**
  1. **Users:**
     + Хранит информацию о пользователях (логины, пароли, профили).
  2. **Orders:**
     + Содержит информацию о заказах, включая их статус, дату создания и сумму.
  3. **Inventory (Инвентаризация):**
     + Хранит данные о продуктах, их доступности и характеристиках.
* **Связи:**
  1. Компоненты Order и Customer из сервера приложений взаимодействуют с соответствующими схемами базы данных.

**Основные взаимодействия:**

1. **Сервер приложений (Tomcat):**
   * Размещает веб-архив (online\_shop\_app.war) и модуль (user\_services.jar).
   * Взаимодействует с компонентами Order и Customer для обработки заказов и работы с клиентами.
2. **База данных (Oracle):**
   * Обеспечивает хранение и доступ к данным для работы приложений.
   * Используется компонентами Order и Customer через схемы Users, Orders, Inventory.

**Элементы системы:**

 **Component** (Компонент): модуль или часть системы, реализующий определённую функциональность и взаимодействующий с другими частями приложения.

 **Artifact** (Артефакт): физический файл, содержащий реализацию кода или конфигурацию, необходимую для развертывания и работы системы.

 **Schema** (Схема): структура базы данных, определяющая организацию и хранение данных, таких как таблицы, связи и ограничения.

**Manifest** — это файл с настройками и инструкциями для запуска или развертывания приложения.