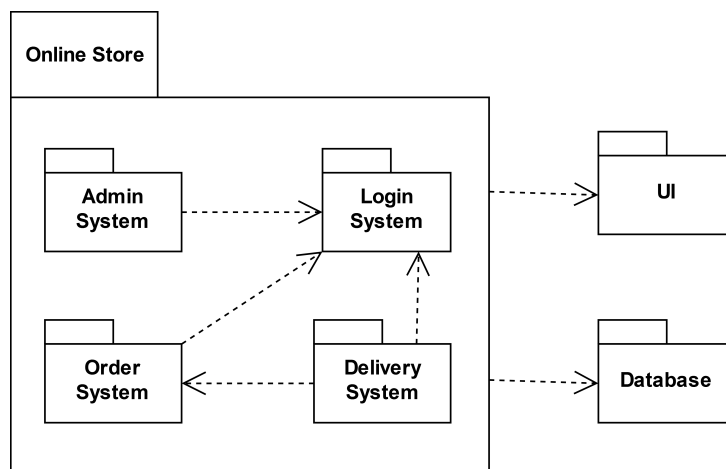


## 1. Диаграмма пакетов

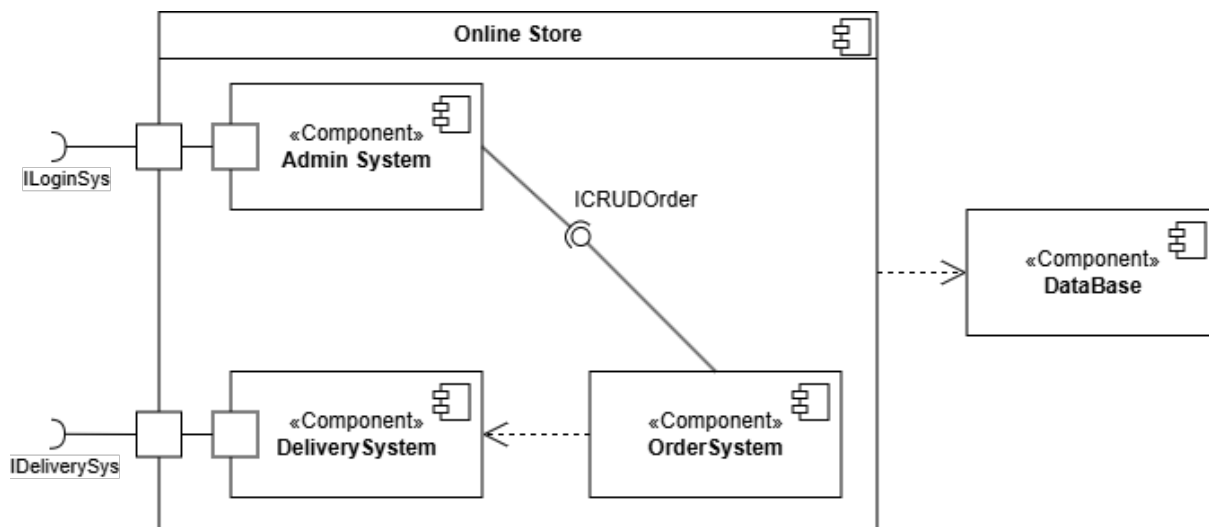


Описание:

1. **Admin System:** Управляет административными функциями;
2. **Order System:** Отвечает за обработку заказов, включая их создание, обновление и хранение;
3. **Database:** Центральное хранилище данных, используемое всеми системами;
4. **Online Store:** Компонент, предоставляющий функционал интернет-магазина, включая витрину товаров и работу с заказами;
5. **Delivery System:** Система, обеспечивающая управление доставкой товаров;
6. **Login System:** Отвечает за аутентификацию и авторизацию персонала;
7. **UI (User Interface):** Пользовательский интерфейс для взаимодействия с системой;
8. **Web:** Связывает пользовательский интерфейс с системой через интернет, используя соответствующую инфраструктуру.

На диаграмме показаны составляющие элементы пакета **Online Store**: пакеты **Admin System**, **Login System**, **Order System** и **Delivery System**. **Admin System** взаимодействует с **Login System** для *CRUD* пользователей, **Order System** и **Delivery System** взаимодействуют с **Login System** для получения информации о пользователе, **Delivery System** взаимодействует с **Order System** для получения информации о заказе. Главная по функциональности часть **Online Store** взаимодействует с пакетом графического интерфейса **UI** для *веб*- и *мобильной* вёрстки, а также с пакетом **Database**, для хранения данных.

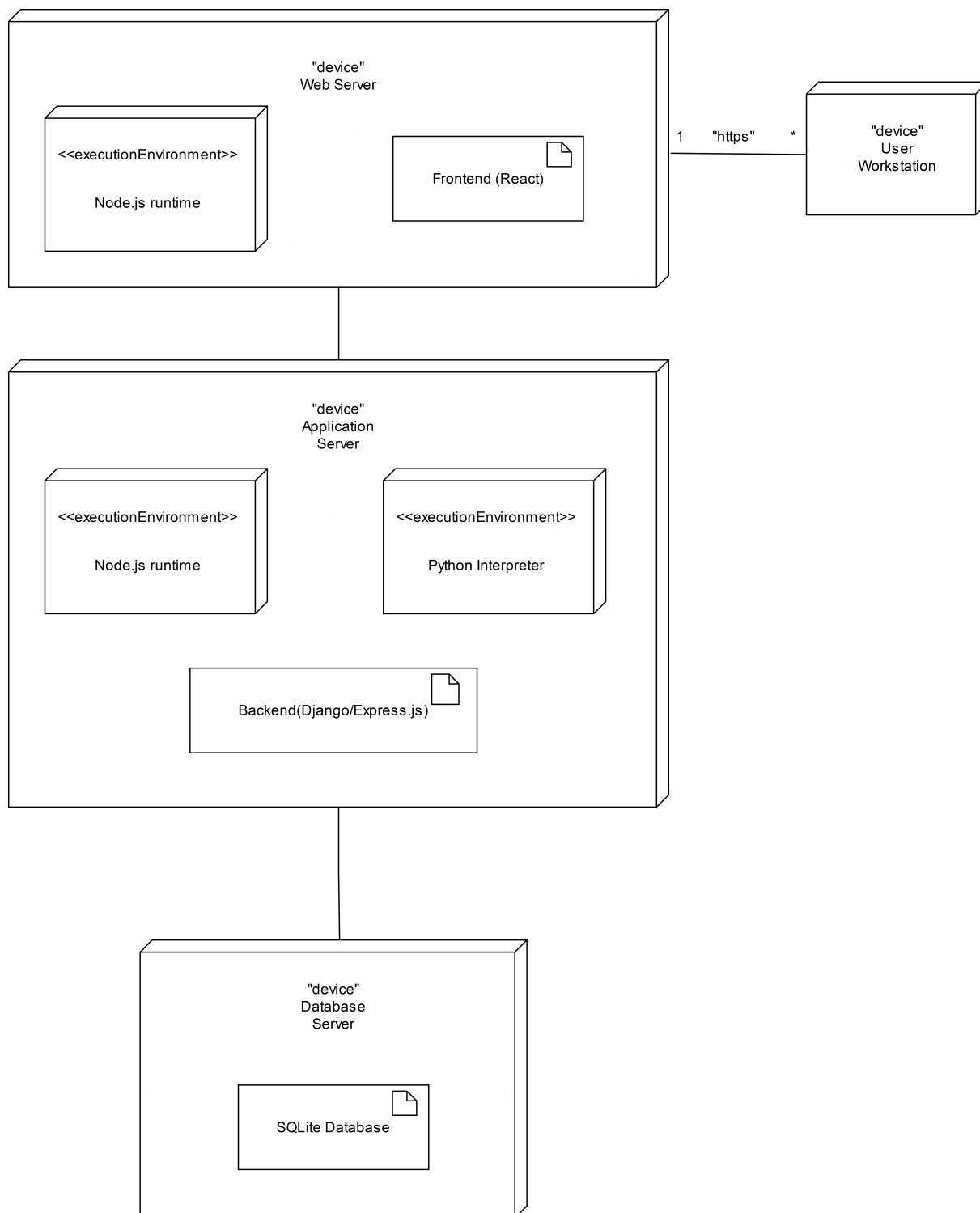
## 2. Диаграмма компонентов



Описание:

1. **Online Store:** основной контейнер, в который входят все компоненты системы;
2. **Admin System:** компонент, отвечающий за административные функции.
  - (a) Он требует интерфейс *ILoginSys*, для аутентификации пользователей;
  - (b) Он требует интерфейс *ICRUOrder*, взаимодействуя с системой управления заказами **OrderSystem**.
3. **Order System:** компонент, управляющий заказами.
  - (a) Он предоставляет интерфейс *ICRUOrder*, предоставляет функционал для управления заказами;
  - (b) Он использует компонент **Delivery System**, для получения информации о заказах для их доставки..
4. **Delivery System:** Отвечает за обработку и управление доставкой заказов.
  - (a) Он требует интерфейс *IDeliverySys*, для обмена данными о заказе;
  - (b) Он используется компонентом **Order System**, для получения информации о заказах для их доставки.
5. **DataBase:** Компонент базы данных, отвечающий за хранение данных системы, для записи информации о новых заказах, хранении информации о пользователе, и о самих товарах (Все компоненты **Online Store** его используют).

### 3. Диаграмма развёртывания



Описание:

#### 1. Рабочая станция пользователя (User Workstation)

На стороне клиента используется:

- **Web Browser** – среда выполнения, предназначенная для работы с интерфейсом приложения. Веб-браузер выполняет React-приложение, которое загружается с веб-сервера.

#### 2. Веб-сервер (Web Server)

Веб-сервер отвечает за предоставление фронтенда клиенту. Его компоненты:

- **Frontend (React)** – артефакт, представляющий собой React-приложение, которое используется для пользовательского интерфейса.
- **Node.js** – среда выполнения для сборки и (при необходимости) серверного рендеринга React.

### 3. Сервер приложений (Application Server)

Сервер приложений обрабатывает бизнес-логику и взаимодействует с базой данных. Компоненты сервера приложений:

- **Backend (Django/Express.js)** – артефакт, представляющий серверную часть веб-приложения.
- **Python Interpreter** – среда выполнения, если сервер написан на Django.
- **Node.js Runtime** – среда выполнения, если серверная часть реализована на Express.js.

### 4. Сервер базы данных (Database Server)

Сервер базы данных хранит всю информацию о продуктах, пользователях и заказах. Используется:

- **SQLite Database** – встроенная база данных, не требующая отдельной среды выполнения.

#### \* Взаимодействие между компонентами

- Пользователь через Web Browser отправляет HTTP-запрос на Web Server.
- Web Server передаёт запрос на Application Server для обработки бизнес-логики.
- Application Server отправляет SQL-запросы к SQLite Database для получения или обновления данных.
- Ответ от базы данных возвращается на Application Server, а затем обратно на Web Server.
- Web Server формирует и отправляет HTTP-ответ пользователю.