**Practical 9**

**Draw the Deployment Diagram for the project definition.**

A Deployment Diagram is a structural diagram in UML that represents the physical deployment of software components onto hardware nodes. It helps visualize the system architecture, including servers, databases, and external APIs, and how they interact.

**Importance of Deployment Diagrams**

* **Understanding System Architecture:** Provides a clear view of how software components are distributed across different hardware nodes.
* **Optimizing Performance:** Helps in identifying the best placement of components to ensure efficiency and scalability.
* **Enhancing Security:** Highlights how different parts of the system interact, aiding in securing communication channels.
* **Simplifying Troubleshooting:** Facilitates debugging and system maintenance by providing a structured deployment view.
* **Supporting Cloud and On-Premise Decisions:** Assists in planning infrastructure for cloud-based or on-premise deployment models.

**Components of a Deployment Diagram**

* **Nodes:** Represent hardware devices such as servers, databases, and user devices.
* **Artifacts:** Software components deployed on the nodes.
* **Connections:** Define the communication between nodes, such as API calls and database queries.
* **Execution Environment:** Defines the runtime environment where components operate, such as containers or virtual machines.
* **Deployment Specification:** Describes configuration settings for each node.

**Deployment Diagram for Movie Booking Management System**

A Deployment Diagram showcases the different hardware and software components in an online movie ticket booking system and their interactions.

**Deployment Breakdown:**

1. **User Device (Mobile App / Web Browser) (Client Node)**
   * Users access the system via a web or mobile interface.
   * Allows users to log in, search for movies, select seats, and book tickets.
   * Displays booking history, offers, and available payment options.
   * Sends requests to the Web Server over HTTPS for processing.
2. **Web Server (Application Server - Hosted on Cloud/On-Premise Server)**
   * Manages core business logic for user and admin functionalities.
   * Handles authentication, movie search, seat selection, booking, and payments.
   * Hosts key modules:
     + **Login System:** Manages user authentication and session handling.
     + **Movie Search System:** Provides filters by genre, language, and showtime.
     + **Seat Selection System:** Displays available seats and facilitates selection.
     + **Booking System:** Processes ticket booking and cancellation requests.
     + **Payment System:** Integrates with external payment gateways for transactions.
     + **Admin Management System:** Handles movie management, user control, and pricing.
   * Connects to the Database Server and Payment Gateway for data retrieval and transactions.
3. **Database Server (SQL) (Storage Node)**
   * Stores user credentials, booking records, and movie details.
   * Maintains real-time seat availability updates.
   * Logs transaction details and revenue reports.
   * Ensures data consistency for all user interactions.
4. **Payment Gateway Server (External API - Razorpay/Stripe/PayPal/etc.) (External Service Node)**
   * Facilitates secure payment processing.
   * Supports multiple payment modes: UPI, card, and cash.
   * Generates invoices and transaction receipts.
   * Communicates with the Web Server via API calls.
5. **Admin Device (Web Panel / Desktop App) (Admin Node)**
   * Provides admin access to manage system functionalities.
   * Allows adding, updating, or removing movies and shows.
   * Manages user accounts and ticket pricing.
   * Monitors revenue statistics and booking trends.

**Deployment Relationships**

* **User Device → Web Server:** Secure HTTP communication for all requests.
* **Web Server → Database Server:** Handles queries for data retrieval and storage.
* **Web Server → Payment Gateway:** Secure API transactions for payments.
* **Admin Device → Web Server:** Admin functionalities via a secure web interface.

**Technology Stack (Example)**

* **Frontend:** React.js / Flutter (Mobile App)
* **Backend:** Node.js / Django / Spring Boot
* **Database:** MySQL / PostgreSQL / MongoDB
* **Hosting:** AWS / Azure / Google Cloud
* **Payment Integration:** Stripe / Razorpay / PayPal API

This architecture ensures an efficient, secure, and scalable movie booking system. Modify as needed for your project requirements.

A diagram of a computer system

AI-generated content may be incorrect.

Fig: Deployment Diagram for Movie Booking Management System