**SOFTWARE ENGINEERING LAB**

**PRACTICAL-7**

**NAME: VEDANT BHUTADA NAME: YASH PATNI**

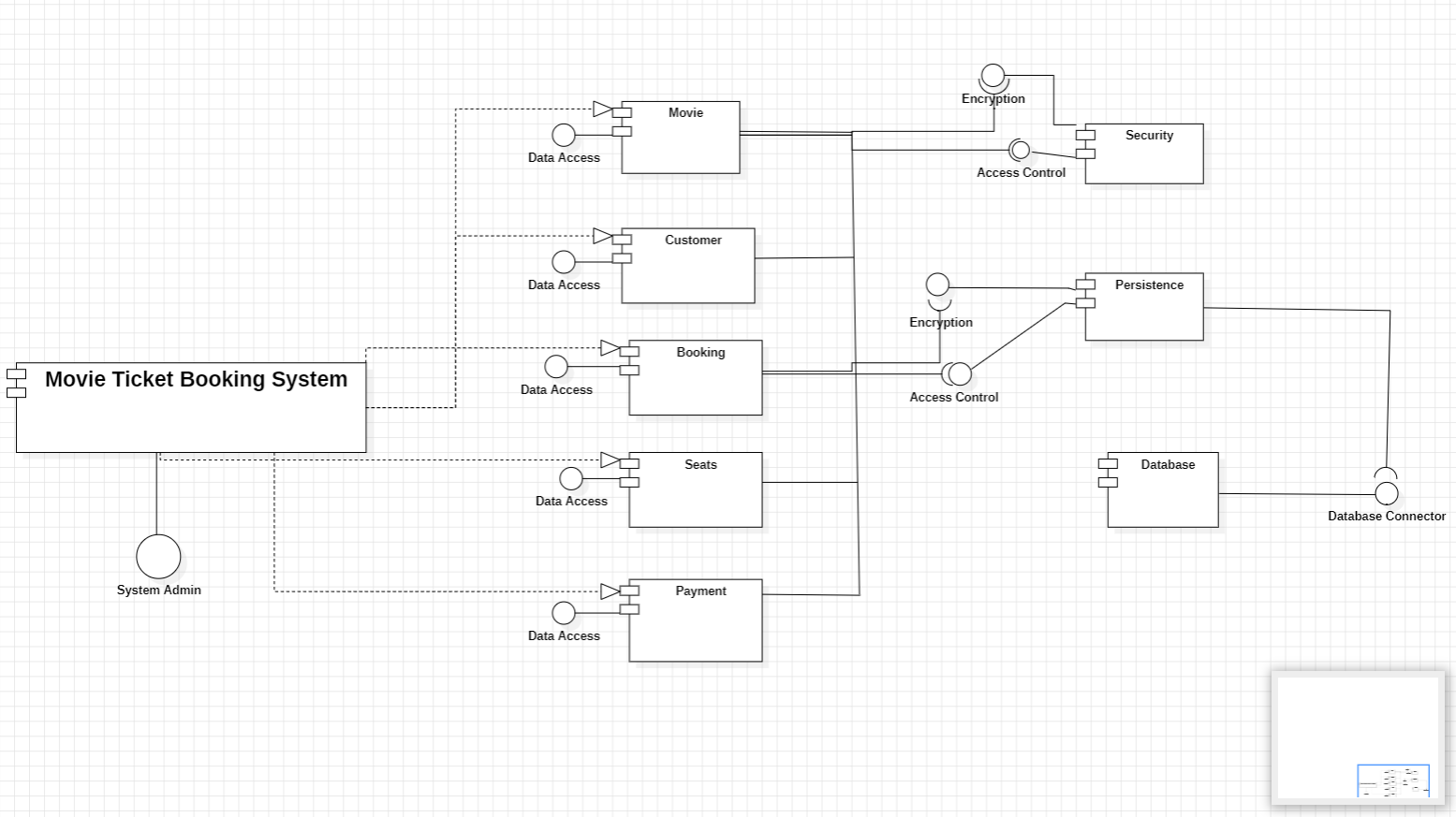
**ROLL NO: 69 ROLL NO: 71**

**BATCH: A4**

**CASE STUDY: MOVIE TICKET BOOKING SYSTEM**

**Aim:** To construct a Component Diagram and Deployment Diagram to depict the structural view of the system.

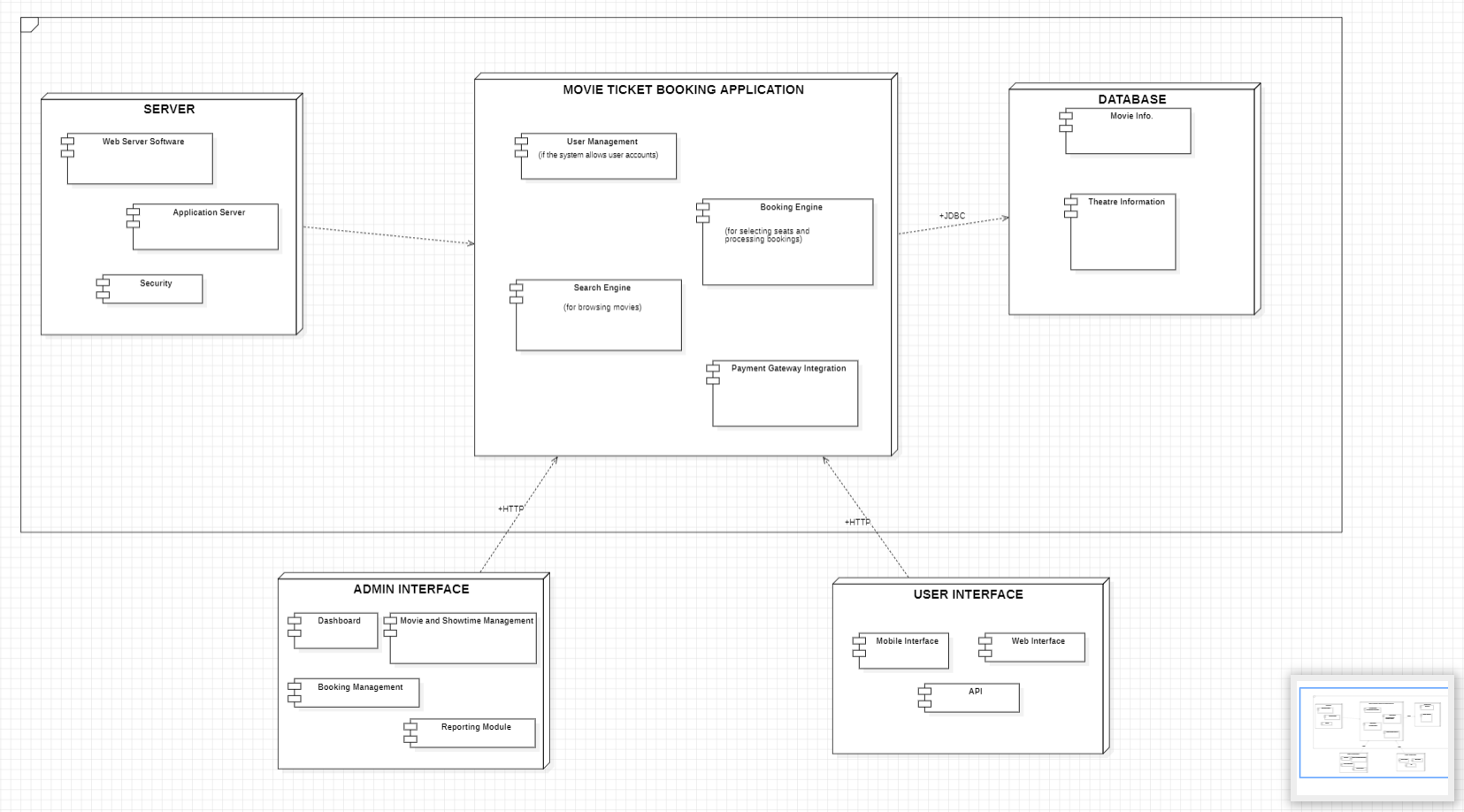
**Component Diagram:**

****

The **component diagram** for a **movie ticket booking system** outlines the system’s functionalities based on the various components:

* **Customer:** Interacts with the **Booking** component to browse movies, select seats, and make purchases.
* **Booking:** Facilitates communication between the Customer, Seats, and Payment components. It likely interacts with other components (not shown) for functionalities like movie listings and user accounts.
* **Seats:** Manages the theater's seating arrangements and provides seat availability information to the **Booking** component.
* **Payment:** Handles financial transactions with the **Customer** component, likely receiving payment information and processing it.
* **Data Access:** Provides a centralized access point for retrieving and updating data from various sources:
  + Relates to the **Seats** component to access seat availability data.
  + Connects to the **Database** component through the **Database Connector** for data storage and retrieval.
* **Database:** Stores the system's data:
  + Accessed by the **Data Access** component through the **Database Connector**.
* **Database Connector:** Mediates communication between the **Data Access** component and the **Database** component.
* **Access Control:** Governs user access to the system's functionalities. It might also connect with the **Security** component for user authentication.
* **Security:** Handles encryption of sensitive data:
  + Might interact with the **Encryption** component (shown as external) if the system uses an external service for encryption.

**Deployment Diagram:**

****

The deployment diagram for a movie ticket booking system depicts the system's functionalities based on the various components:

* **Server** runs the **Movie Ticket Booking Application**.
* **Movie Ticket Booking Application** facilitates communication between various components including User Interface, Admin Interface, and Database. It likely handles functionalities like user management, search engine, booking engine, and payment gateway integration (not shown in the diagram).
* **Database** stores the system’s data, such as movie listings, seat availability, and customer information.
* **User Interface** allows customers to interact with the system. They can browse movies, select showtimes and seats, and potentially view booking history (if the system allows user accounts).
* **Admin Interface** provides a way for administrators to manage the system. They can add new movies and showtimes, manage theater information, create and manage user accounts (if applicable), and monitor bookings and generate reports.

**Conclusion:**

We understood demonstration of component-based design and deployment architectures for a real-world application like an online movie ticket booking system. By constructing these diagrams, we've visualized the system's internal functionalities and how different components interact with each other.