EXPERIMENT 4

Learning Objective: **Modeling using Box and Line and UML diagrams.**

Tools: Draw.io, word, Canva

Theory: In software architecture, a box and line diagram is a simplified representation used to depict the structure and organization of a software system. This type of diagram is commonly employed to visualize the components of a system and the relationships between them. Unlike detailed diagrams like UML (Unified Modeling Language), box and line diagrams are often high-level and abstract, providing an overview rather than a deep dive into specific details.

Components of a Box and Line Diagram in Software Architecture

1. **Boxes**:
   - **Components**: Each box typically represents a component, module, or subsystem of the software. This could be anything from a microservice, database, application server, or any other functional part of the system.
   - **Layers**: In layered architectures, boxes might represent different layers such as presentation, business logic, and data access layers.
2. **Lines**:
   - **Connections**: Lines indicate the relationships or interactions between components. These can represent data flow, control flow, dependencies, or communication paths.
   - **Directionality**: Arrows on lines can show the direction of data flow or the direction of dependencies.
3. **Annotations**:
   - Descriptive text may be used to label components and connections, providing additional context.
   - Annotations can include protocols (e.g., HTTP, TCP), data formats (e.g., JSON, XML), or other relevant details.

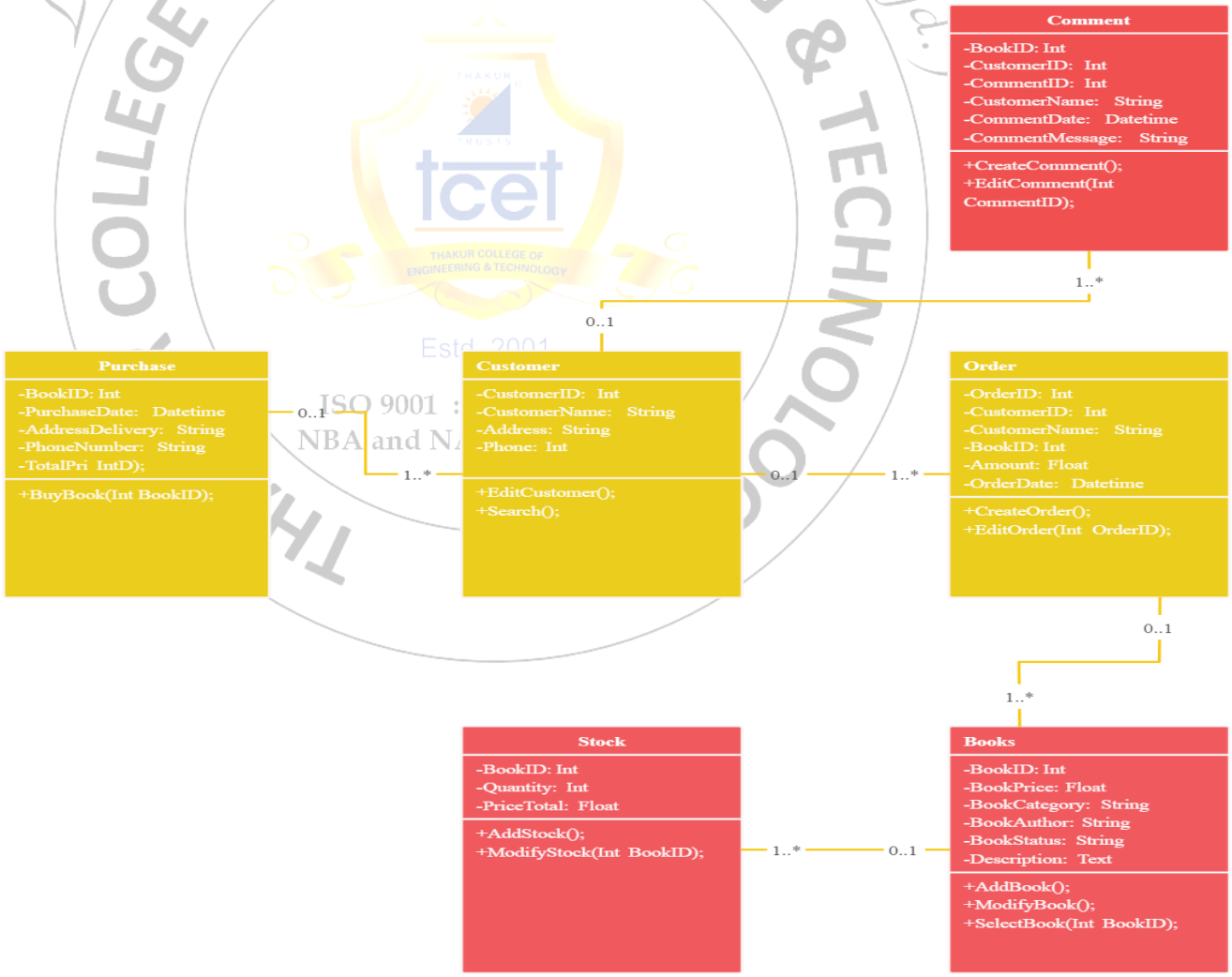
Purpose of a Box and Line Diagram

- **Communication**: Provides a clear and simple way to communicate the system architecture to both technical and non-technical stakeholders.
- **High-Level Overview**: Offers a bird's-eye view of the system, helping architects and developers understand the overall structure.
- **Planning and Design**: Assists in the initial phases of system design and architecture planning, allowing teams to identify key components and their interactions.

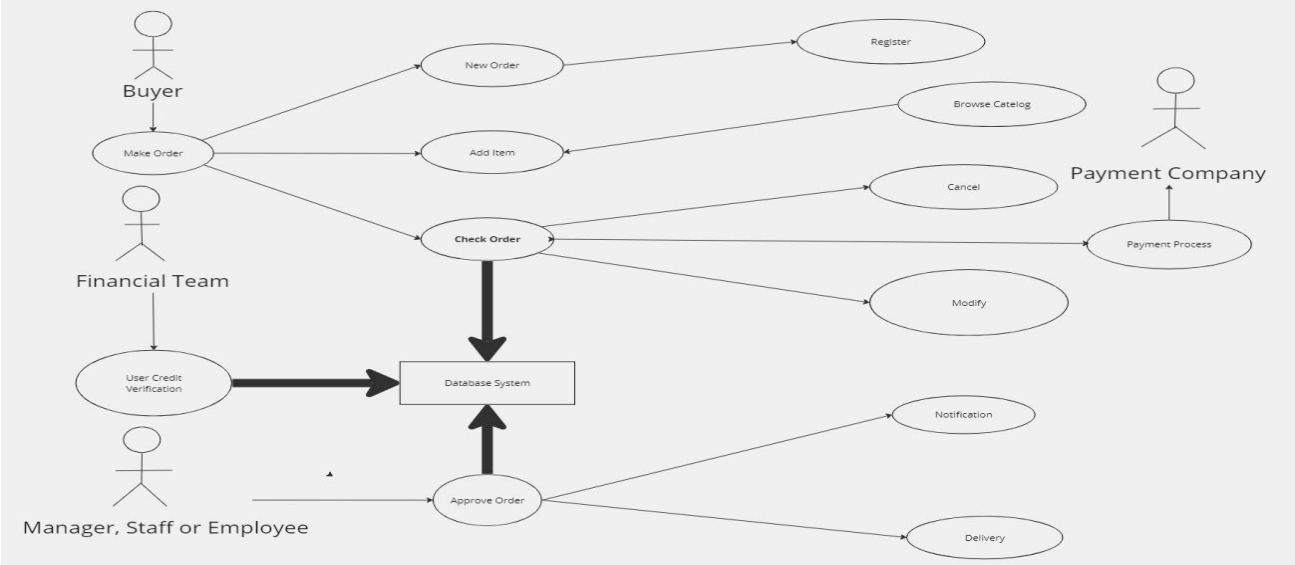
Explanation

- **API Gateway**: Acts as a single entry point for client requests, routing them to the appropriate microservices.
- **Authentication Service**: Handles user authentication and authorization.
- **User Service**: Manages user-related operations.
- **Order Service**: Handles order processing and management.
- **Payment Service**: Processes payments and manages financial transactions.
- **Notification Service**: Sends notifications to users based on certain events.
- **Inventory Service**: Manages inventory and product availability.

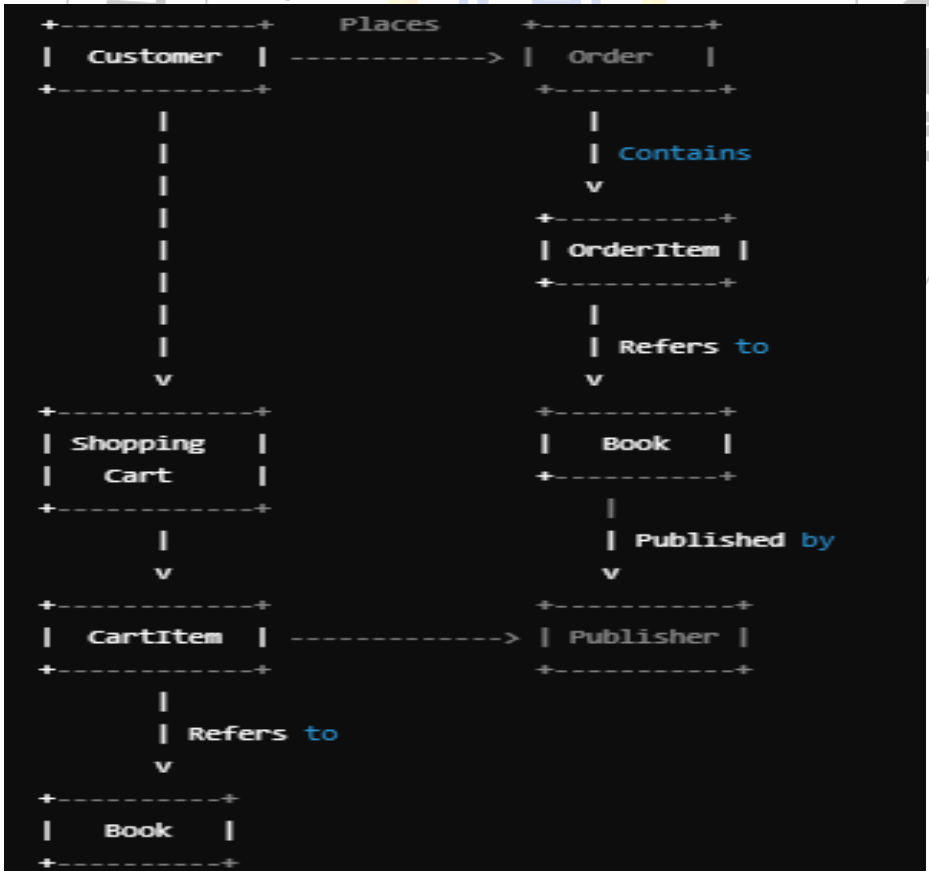
Design: UML Diagram(Class diagram)



UML Diagram(Use Case Diagram)



Box and Line Diagram



Result and Discussion: The box and line diagram for the online bookstore management system effectively represents the key components and their interactions, such as user interface, database, payment gateway, and inventory management. The diagram shows clear data flow and connectivity among components, aiding in understanding system structure and design.

Learning Outcomes: The student should have the ability to

LO1: Gained the ability to visualize system architecture through component relationships.

LO2: Enhanced understanding of data flow and connectivity within the system.

Conclusion: The box and line diagram serves as a useful tool for visualizing the overall architecture of the online bookstore management system, helping in planning and communication, but it should be supplemented with more detailed diagrams for comprehensive development guidance.

For Faculty Use:

Correction Parameters	**Formative Assessment [40%]**	**Timely completion of Practical [40%]**	**Attendance / Learning Attitude [20%]**	
Marks Obtained				