## Experiment No. 4: Class Diagram for Online Bookstore Management System

**Aim:** To Sketch Class Diagram for Online Bookstore Management System.

**Theory:** A class diagram in software engineering is a visual representation of the structure and relationships among classes in a system. It is one of the most widely used diagrams in object-oriented modeling and design. Here's a brief overview of the theory behind class diagrams:

Classes:

- A class is a blueprint for creating objects in object-oriented programming.
- It encapsulates data for the object (attributes or properties) and behaviors (methods or functions).
- In a class diagram, a class is represented as a rectangle with three compartments: the top compartment contains the class name, the middle compartment contains the attributes, and the bottom compartment contains the methods.

Attributes:

- Attributes represent the data or properties associated with a class.
- They describe the state of an object.
- Attributes are typically represented in the middle compartment of the class rectangle in a class diagram.

Methods:

- Methods represent the behaviors or operations that objects of a class can perform.
- They define how objects interact with each other and with their own data.
- Methods are typically listed in the bottom compartment of the class rectangle in a class diagram.

Relationships:

- Relationships define how classes are related to each other.
- There are several types of relationships:
  - Association: Represents a relationship where objects of one class are connected to objects of another class.
  - Aggregation: Represents a "whole-part" relationship where one class (the whole) contains or is composed of other classes (the parts).
  - Composition: Similar to aggregation, but with stricter ownership semantics, where the parts are entirely contained within the whole.
  - Inheritance: Represents an "is-a" relationship between classes, where one class (subclass or derived class) inherits attributes and methods from another class (superclass or base class).

Multiplicity:

- Multiplicity specifies the number of instances of one class that are related to instances of another class.
- It is represented using numbers or symbols near the association lines between

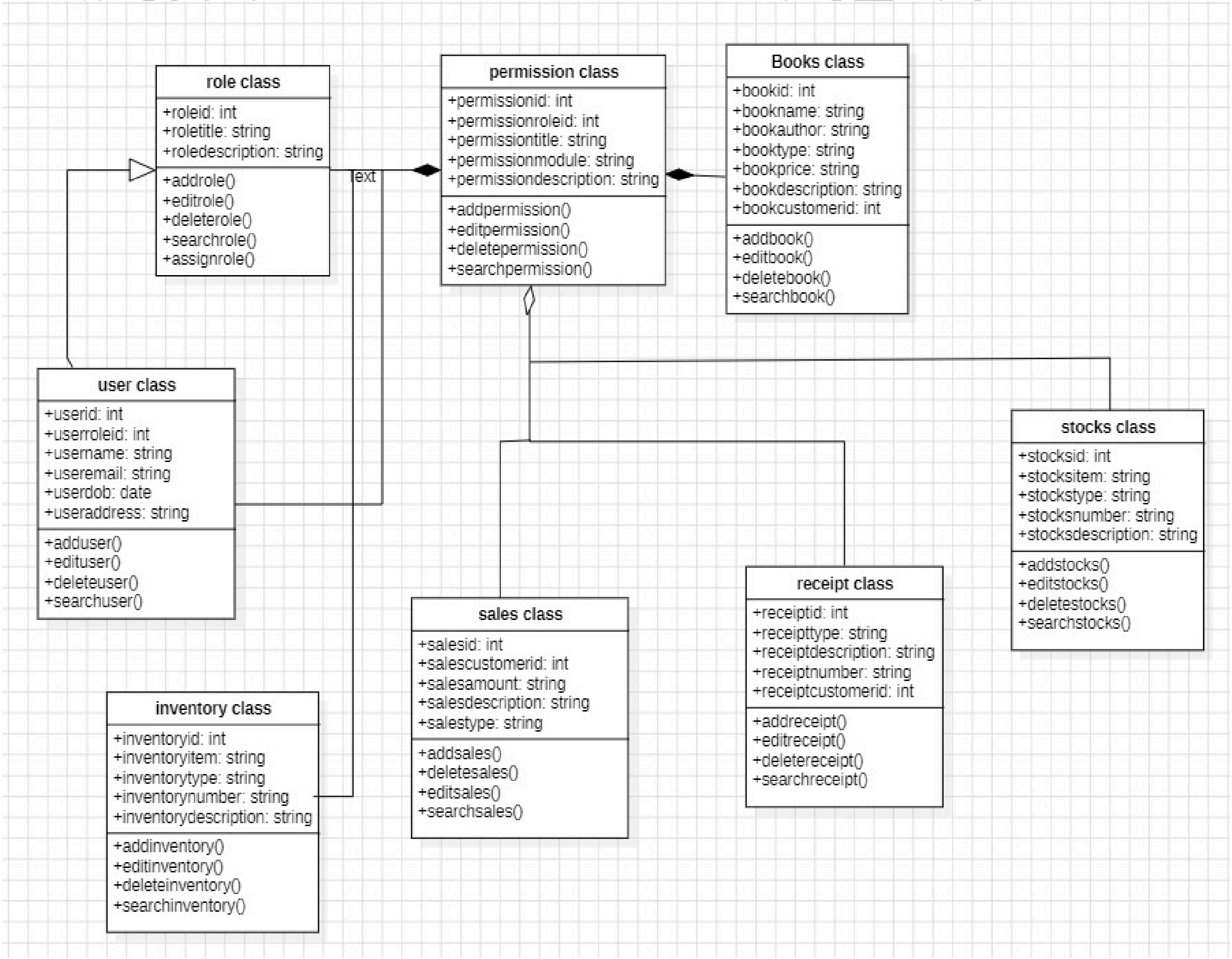
classes in a class diagram.

Generalization and Specialization:

- Generalization is the process of extracting common characteristics from two or more classes and combining them into a generalized superclass.
- Specialization is the opposite process, where a specialized subclass inherits attributes and methods from a more general superclass.

Class diagrams are essential tools for designing and visualizing the structure of object-oriented systems, aiding in communication among stakeholders and guiding the implementation process. They provide a high-level overview of the system's architecture and help developers understand how different parts of the system interact with each other.

**Output:**



Class Diagram for Online Bookstore Management System

**Conclusion:** We have successfully implemented the Class Diagram of the Online Bookstore Management System.

**Learning Outcome:**

- Understand the concept of Class Diagram.
- Learn how to model the behavior of a system by representing a Class Diagram.

**For Faculty Use:**

| Correction Parameters | Formative Assessment [40%] | Timely completion of Practical [ 40%] | Attendance / Learning Attitude [20%] | |
|---|---|---|---|---|
| Marks Obatined | | | | |