SOFTWARE ENGINEERING PRACTICAL DOCUMENT

PRACTICAL NO: 1

AIM: Study and implementation of class diagrams.

SOLUTION

In Object-Oriented modeling, the main building block generally represents different objects in a system, their attributes, their different functions, and relationships among objects. These building blocks are known as Class Diagram.

Class diagrams are generally used for conceptual modeling of static view of a software application, and for modeling translating models into programming code in a detailed manner. At time of developing or construction software systems, a class diagram is widely used. They are also used for data modeling. It is used to show classes, relationships among them, interface, association, etc. Class in a class diagram simply is a blueprint of an object. It simply describes and explains different type of objects in system, and different types of relationships that exist between them

Class Diagram for Library Management System:

Aggregation and Multiplicity are two important points that need to take into consideration while designing a Class Diagram. Let us understand in detail.

1. **Aggregation –**
   Aggregation simply shows a relationship where one thing can exist independently of other thing. It means to create or compose different abstractions together in defining a class. Aggregation is represented as a part of relationship in class diagram. In diagram given below, we can see that aggregation is represented by an edge with a diamond end pointing towards superclass. The "Library Management System" is superclass that consists of various classes. These classes are User, Book, and Librarian as shown in diagram. Further, for "Account" class, "User" is a superclass. All of this share a relationship and these relationships are known as aggregate relationships.

2. **Multiplicity –**
   Multiplicity means that number of elements of a class is associated with another class. These relations can be one-to-one, many-to-many, and many-to-one or one-to-many. For denoting one element we use **1**, for zero elements we use **0**, and for many elements we use *****. We can see in diagram; many users are associated with many books denoted by ***** and this represents a **many-to-many** type of relationship. One user has only one account that is denoted by 1 and this represents a **one-to-one** type of relationship.
   Many books are associated with one librarian and this represents **many-to-one** or **one-to-many** type of relationship. All these relationships are shown in diagram.

Class Diagram for Library Management System simply describes structure of Library Management System class, attributes, methods or operations, relationship among objects.

Classes of Library Management System:

- **Library Management System class –**
  It manages all operations of Library Management System. It is central part of organization for which software is being designed.
- **User Class –**
  It manages all operations of user.
- **Librarian Class –** It manages all operations of Librarian.
- **Book Class –**
  It manages all operations of books. It is basic building block of system.
- **Account Class –**
  It manages all operations of account.
- **Library database Class –**
  It manages all operations of library database.
- **Staff Class –**
  It manages all operations of staff.
- **Student Class –**
  It manages all operations of student.

Attributes of Library Management System:

- **Library Management System Attributes –**
  User Type, Username, Password
- **User Attributes –**
  Name, Id
- **Librarian Attributes –**
  Name, Id, Password, Search String
- **Book Attributes –**
  Title, Author, ISBN, Publication
- **Account Attributes –**
  no_borrowed_books, no_reserved_books, no_returned_books, no_lost_books fine_amount
- **Library database Attributes –**
  List_of_books
- **Staff Class Attributes –**
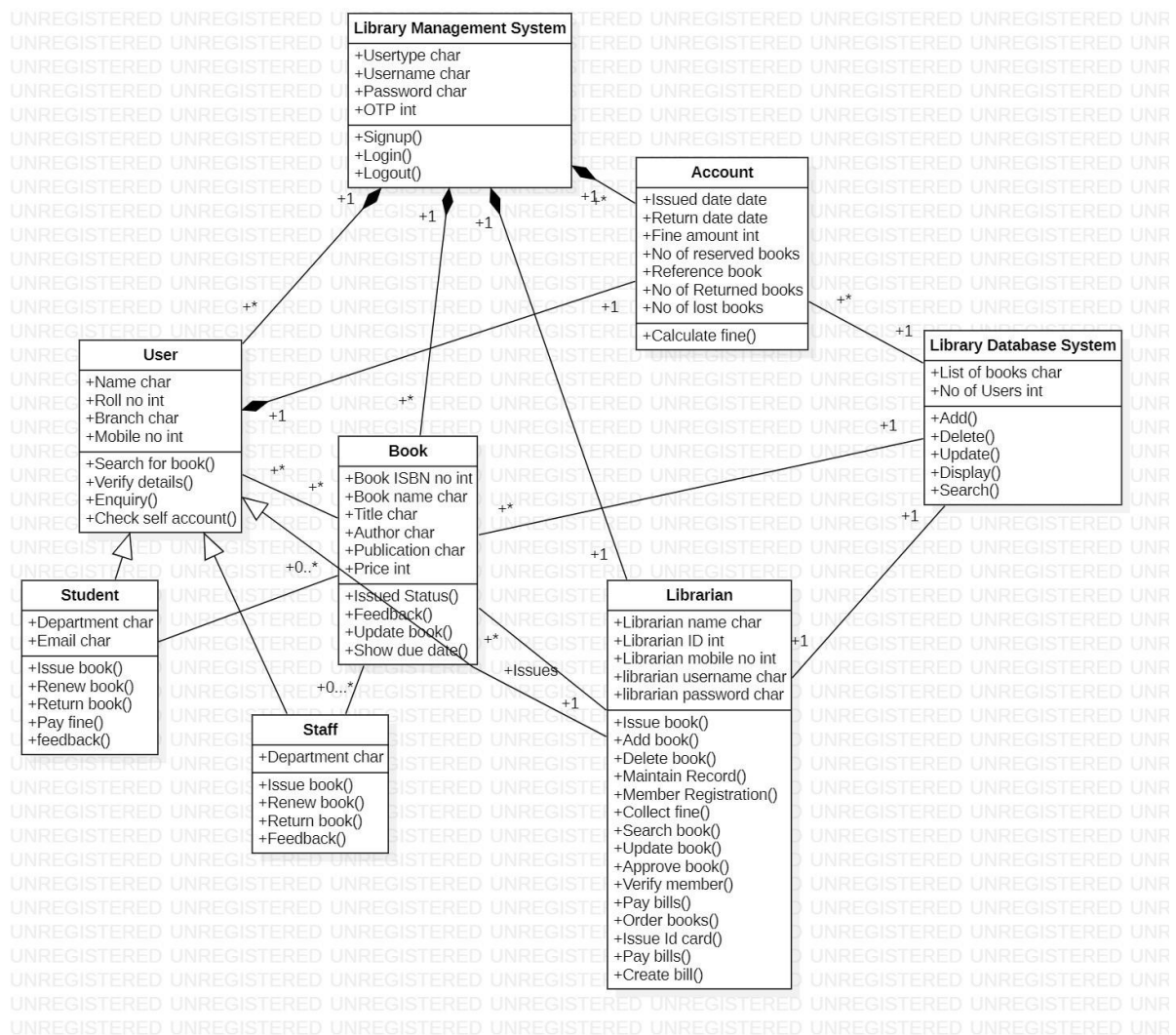 Dept
- **Student Class Attributes –**
  Class

Methods of Library Management System:

- **Library Management System Methods –**
  Login(), Register(), Logout()
- **User Methods –**
  Verify(), CheckAccount(), get_book_info()
- **Librarian Methods –**
  Verify_librarian(), Search()
- **Book Methods –**
  Show_duedt(), Reservation_status(), Feedback(), Book_request(), Renew_info()
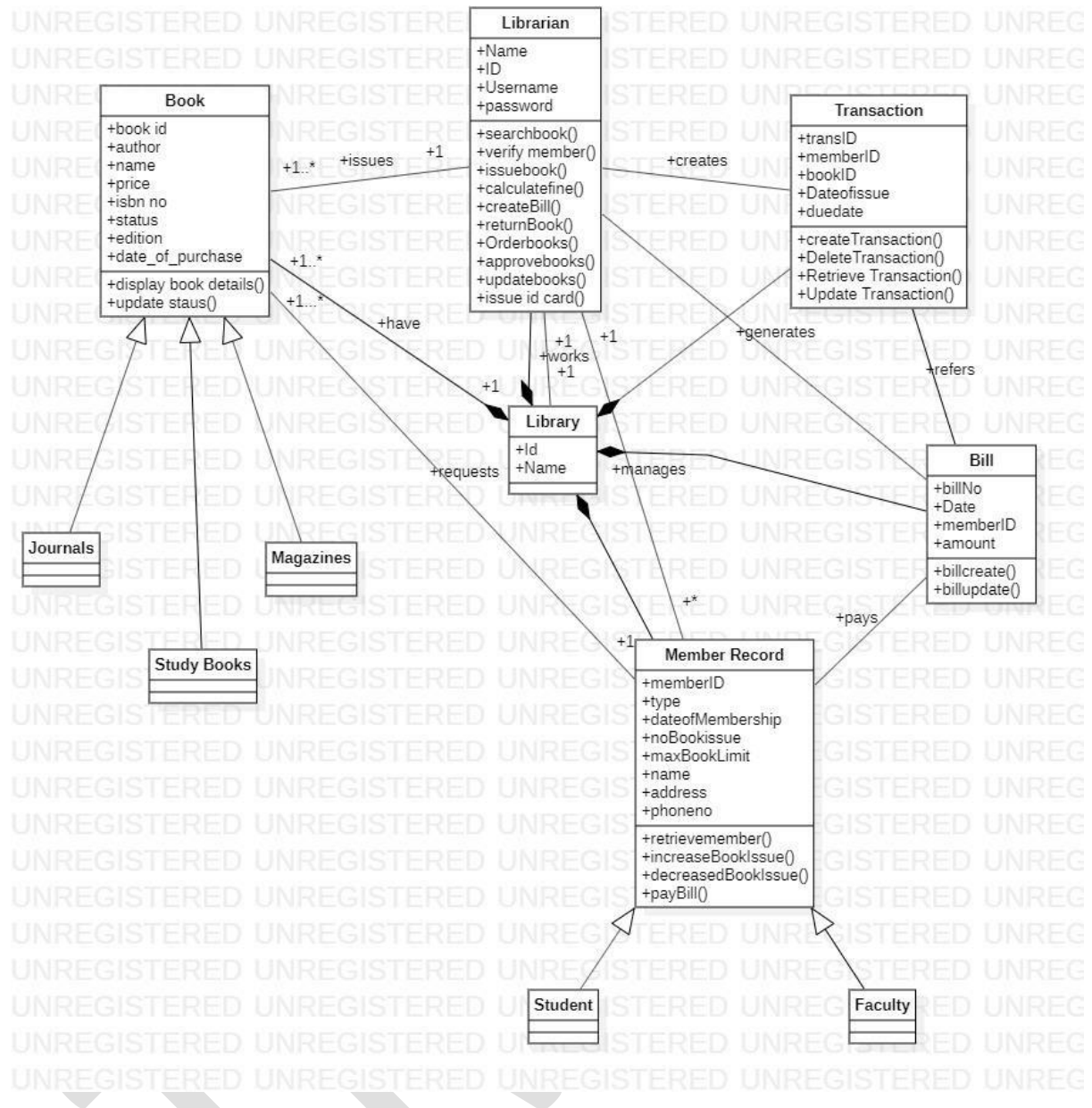- **Account Methods –**
  Calculate_fine()

- **Library database Methods –**
  Add(), Delete(), Update(), Display(), Search()

A Library Management System is software built to handle the primary housekeeping functions of a library. Libraries rely on library management systems to manage asset collections as well as relationships with their members. Library management systems help libraries keep track of the books and their checkouts, as well as members' subscriptions and profiles.
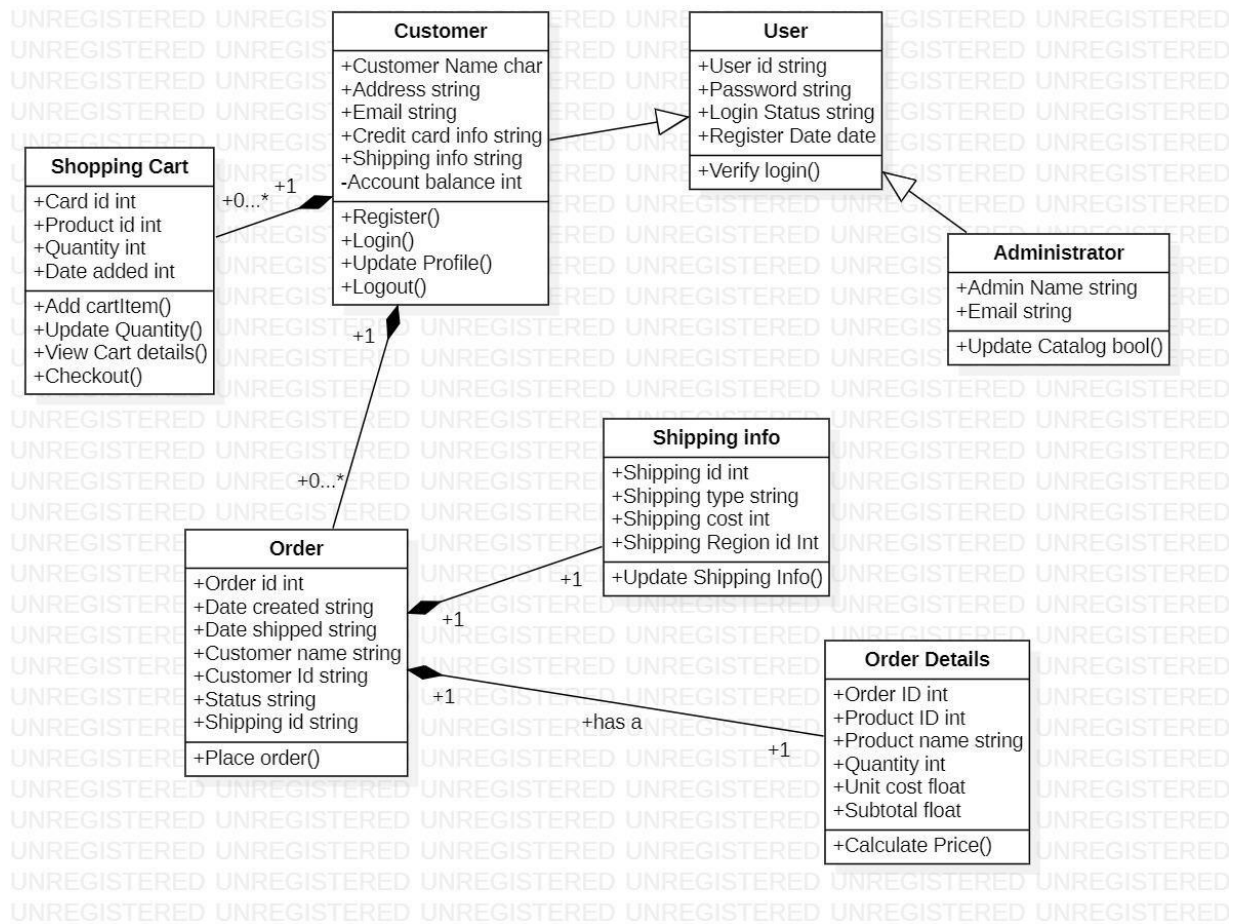
Library management systems also involve maintaining the database for entering new books and recording books that have been borrowed with their respective due dates.

ONLINE SHOPPING CART