

# Library Management System

## *Detailed Design Document*

### **1. Introduction**

Library management system is a computerized version of general library. It is a system to make easy to execute all daily works of a normal library into android phone or laptop. The software is designed to make secure, reliable, comfortable and less costly system to collect book, issue book, return book, adding late fee, ensure the notification for new book for student and teacher. The interface will show different categories of books and their details. If the user delay to return the book automatically generated a notification on his profile and also give an option to renew his return time. The librarian will monitor the whole system of library

This document was made on basis of Requirements Documentation so all requirements of customer was taken into account. However some changes concerning design of developed LMS can be brought after studying by client of Design Documentation (DD). Main goal of DD is description of design and structure of LMS system's interface. The document is intended first of all for reporting about implemented work to customer, and also for final coordination with him of questions connected to design of developed system. Last version of document will be given to developer for realization of chosen structure.

### **2. Proposed Conceptual class Diagram**

To develop a digital library Management System software we have some initial classes. Their structure and here we show the conceptual classes. In this class diagram classes only has some attribute or some variable with their type. In this

section no need class function or operation. It is discussed in detail class diagram section

Conceptual relation is given bellow to figure 01.

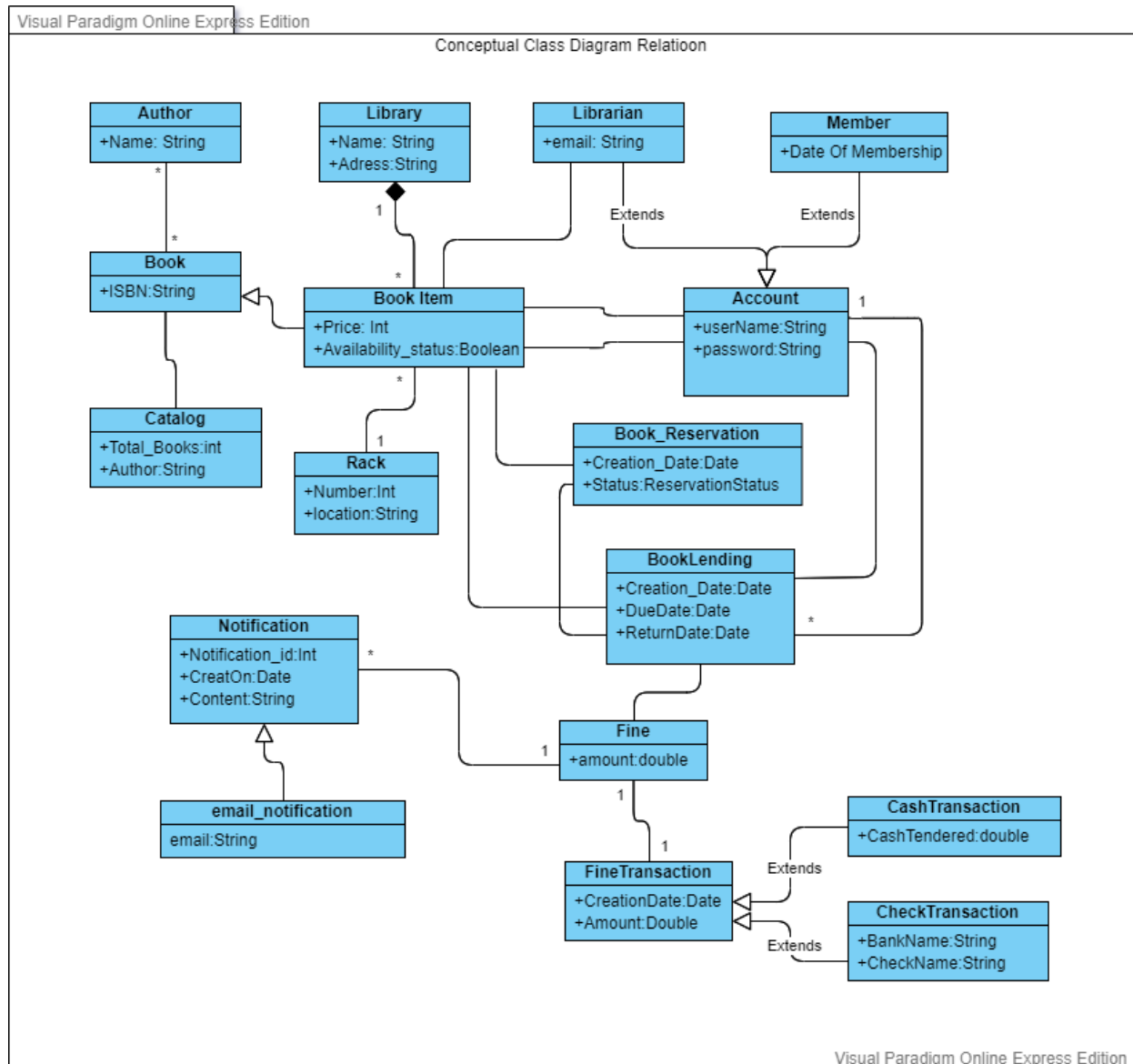


Figure 01: Conceptual Class Diagram Relation

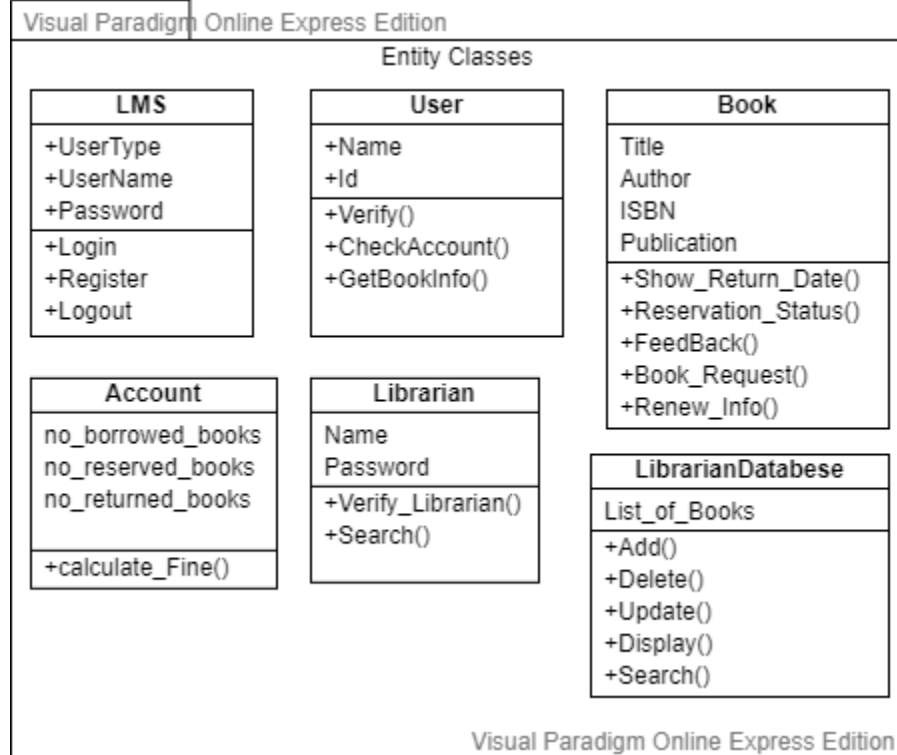


Figure 2: Entity Classes

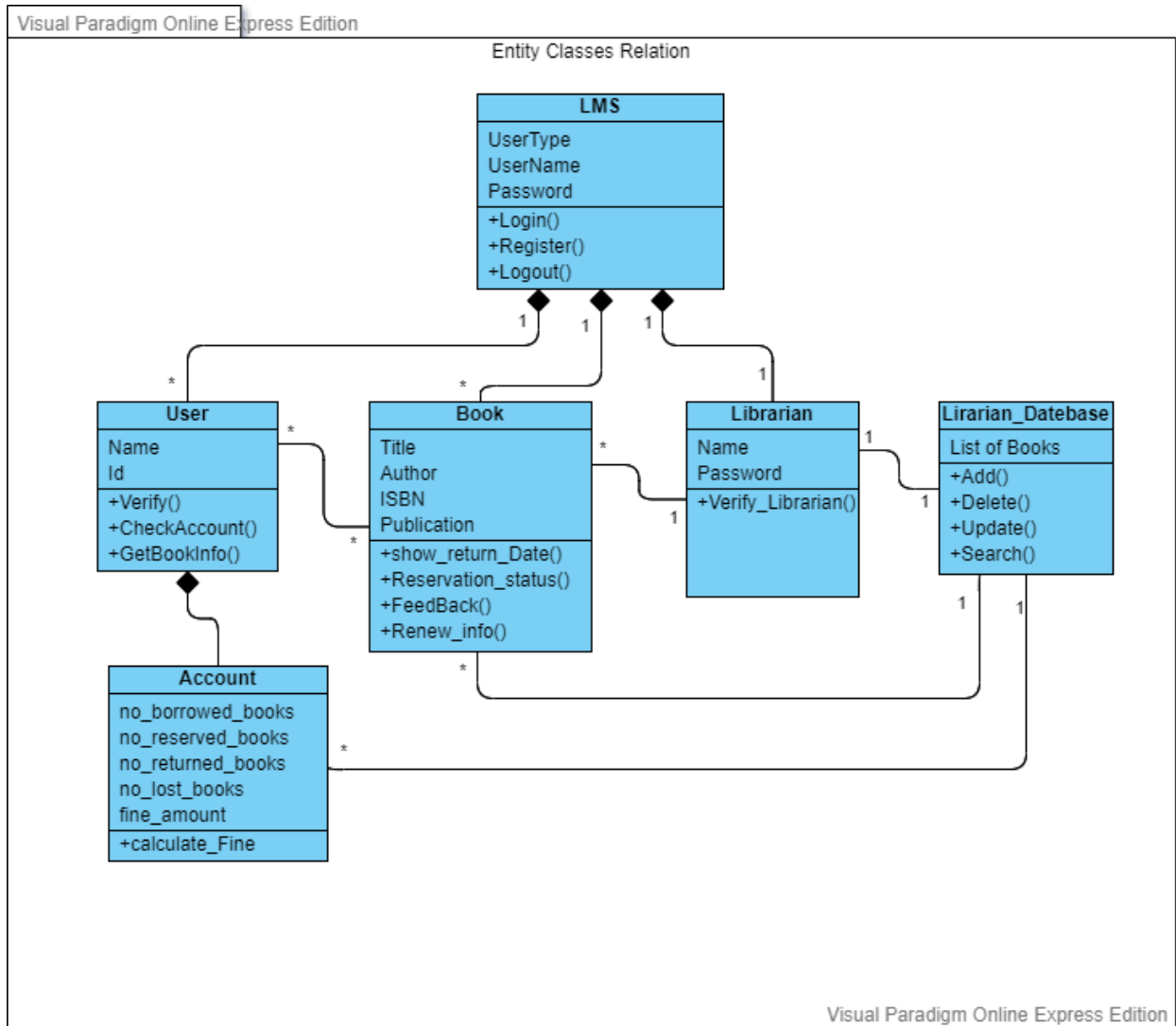


Figure3: Entity Classes Relation

### 3. Specify interfaces:

An important part of any design process is the specification of an interface between the components in the design we need to specify interfaces so that objects and subsystems can be designed in parallel. Once interface has been specified, the developers of other objects may assume that interface will be implemented.

Interface design is concerned with the specifying the detail of the interface to an object or to a group of objects. This means defining the signature and semantics of the services that are provided by the object or by a group of objects.

We should not include details of the data representation in an interface design, as attributes are not defined in an interface specification. However we should include operations to access and update data. As the data representation is hidden, it can be easily changed without affecting the objects that use that data. This lead to design inherently more maintainable. The interfaces of Library management system is given bellow.

1. User Login
2. User Registration
3. User Dashboard
4. Admin Login
5. Admin Dashboard

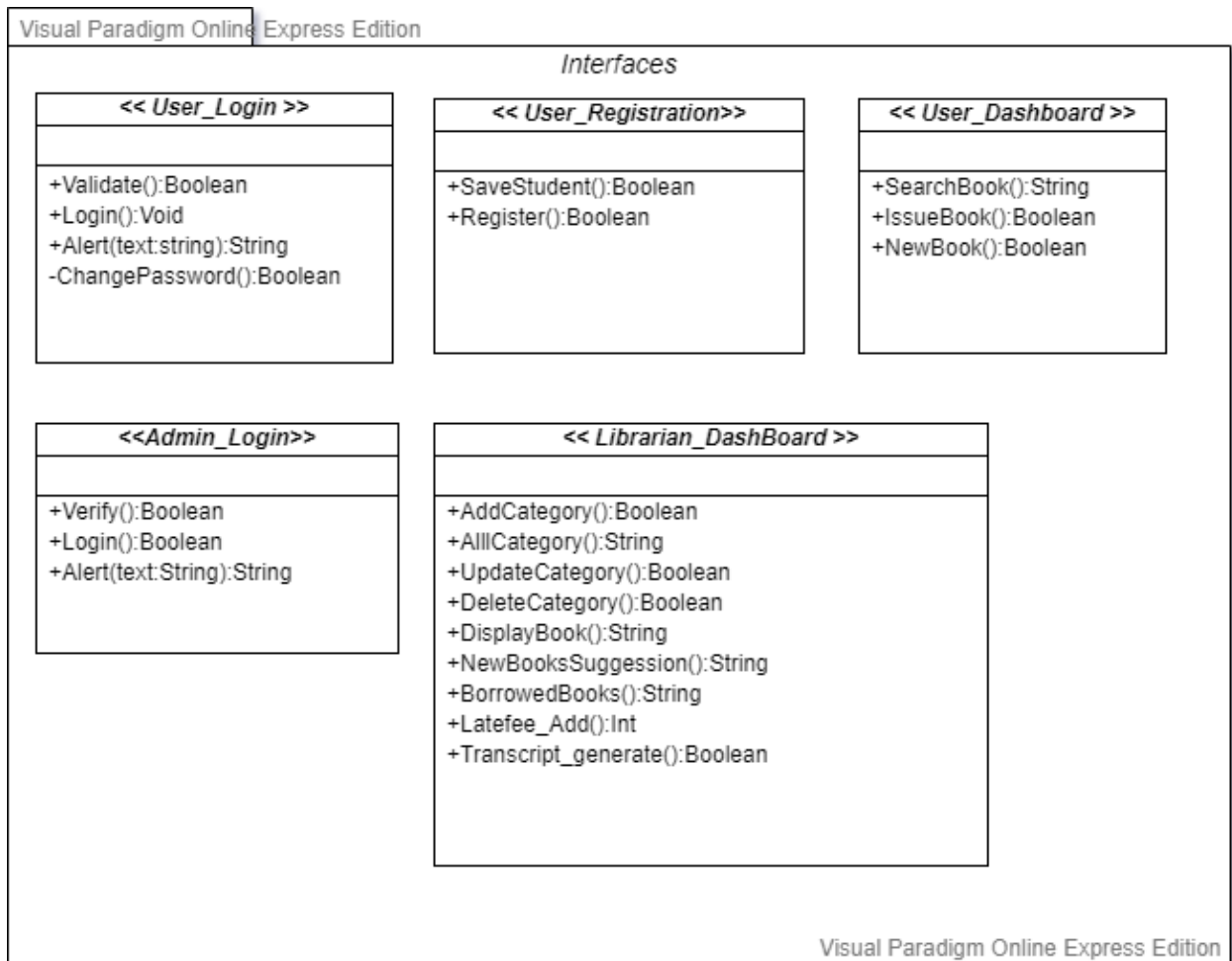


Figure 4: Interface Specified

## 4. Design Pattern

Software design pattern is a general and reusable solution to a commonly occurring problem within a given context in software design. It is description or template for how to solve a problem that can be used in many different situations. Design pattern are formalized best practice that the programmer can use to solve common problems when designing and application or the system.

Singleton is a creational design pattern which ensure that a class has only one instance and provide a global point access to it. It solves the problem of having multiple instance by hiding the constructor of the class as well as assigning an operation which return the sole instance of a class. The LMS system uses singleton design pattern to solve the problem of opening multiple connection to the database and authenticate users

Observer is a behavioral pattern which define a one to many dependency between object where state change in one object results in all its being notified and update automatically. It solves the problem by defining object name subject and observer. Subject maintains the list of dependents and notifies the observer automatically of any state change by calling one of their methods

## **5. Detailed Class Diagram**

A class defined a method and variable in an object which is a specify entity in a program or unit of a code representing the entity. Class diagram is an illustration of relationship and source code dependencies among classes. The detailed class diagram of LMS system is given bellow.





## **6. System Limitation**

There exist no system developed by man that is perfect and complete. LMS to has some limitations on its functions. The limitation of LMS system is given bellow

- Input of this system has to be given manually by keyboard
- There are no option for batch input to this system
- user has to be added manually to database
- The system will available for android device

## **7. Future work**

Technology is ever changing to cope with the change and its satisfaction of user, a system need to be kept constantly update. To overcome the limitation of the system as well as to satisfy the need of its user following changes are planned to make future versions of library management system

- ❖ Functionality to input the system using barcode scanner can be added
- ❖ Function for batch input using Excel work book format (.xlsx) will be added
- ❖ A registration system will be added to add new user and assigning privileges
- ❖ Similar system will be developed for pc platform

## **8. Conclusion**

Library Management system is a computerized version of general library system. It is developed to perform general task into virtual task. It will help student and

librarian to read and restore books. We hope according to the client requirements it will work properly. The library will also be digital by using the software.