Software Requirements Specification

for

Library Management System

Prepared by

Group 11

MK Chaitanya (17100026),
Nitin Chandra (17100030)
C. Tarun Sai (17101014)
Vinayak Bhartia (17101054)

8th November 2019

1. Introduction **3**
1.1 Requirement Analysis 4
1.2 Software Design 4
1.3 Coding 4
1.4 Testing 5
1.5 Purpose 5
1.6 Scope: 5
1.7 Definition, Acronyms, Abbreviation: 6
1.8 Reference: 6
1.9 Overview 6

2. Overall Description **6**
2.1 Product Perspective: 6
2.2 Product function: 7

3. Specific Requirement: **8**
3.1.1 User Interface: 8
3.1.2 Hardware interface: 8
3.1.3 Software interface 9
3.2 Functional requirements: 9
3.3 Performance requirements: 9
3.4 Design constraints 9
3.5 System attributes 9

Analysis Models: **10**
Use-Case Diagram 10
Sequence Diagram 11
Collaboration Diagram 12
Activity Diagram 13
Class Diagram 14
Component Diagram 15

1. Introduction

A development process consists of various phases, each phase ending with a defined output. The main reason for having a phased process is that it breaks the problem of developing software into successfully performing a set of phases, each handling a different concern of software development. This ensures that the cost of development is lower than what it would have been if the whole problem was tackled together. Furthermore, a phased process allows proper checking for quality and progress at some defined points during the development (end of process).Without this one would have to wait until the end to see what software has been produced.

Any problem solving in software consist of these steps:-

1.1 Requirement Analysis

Requirement Analysis is done in order to understand the problem the software system is to solve.

There are two major activities in this phase:

1. Problem Understanding or Analysis
2. Requirement Specification.

In problem analysis, the aim is to understand the problem and its context, and the requirements of the new system that is to be developed. Once the problem is analyzed and the essentials understood, the requirements must be specified in the requirement specification document. The requirements document must specify all functional and performance requirements; the formats of inputs and outputs etc.,

1.2 Software Design

The purpose of design phase is to plan a solution of the problem Specified by the requirements document. This phase is the first step in moving from the problem domain to solution domain. The design activity often results in three separate outputs:

Architecture Design, High level Design and Detailed Design.

1.3 Coding

The main goal of the coding phase is to translate the design of the System into code in a given programming language. The coding phase affects both testing and maintenance profoundly. The goal of coding should be to reduce the testing and maintenance effort, hence during coding the focus should be on developing programs that are easy to read and understand.

1.4 Testing

The function of testing is to detect the defects in the software. The main goal of testing is to uncover requirement, design and coding errors in the programs. The main goal of the requirement phase is to produce the software requirement specification (SRS), which accurately captures the client's requirements. SRS is a document that describes what the software should do. The basic purpose of SRS is to bridge the communication gap between the clients, the end users and the Software developers. Another purpose in helping users to understand their own needs.

1.5 Purpose

The SRS typically contains the brief description of the project. The purpose of the requirement document is to specify all the information required to design, develop and test the software.

- The purpose of this project is to provide a friendly environment to maintain the details of books and library members.
- The main purpose of this project is to maintain an easy circulation system using computers and to provide different reports.

1.6 Scope:

The document only covers the requirements specifications for the Library Management System. This document does not provide any references to the other component of the Library Management System. All the external interfaces and the dependencies are also identified in this document.

Feasibility study: The overall scope of the feasibility study was to provide sufficient information to allow a decision to be made as to whether the Library Management System project should proceed and if so, its relative priority in the context of other existing Library Management Technology. The feasibility study phase of this project had undergone through various steps which as describe as under:

- Identify the origin the information at different level.
- Identify the expectation of the user from computerized system.
- Analyze the drawback of existing system(manual system)

1.7 Definition, Acronyms, Abbreviation:

- JAVA -> platform independence
- SQL -> Structured query Language
- DFD -> Data Flow Diagram
- CFD -> Context Flow Diagram
- ER -> Entity Relationship
- IDE -> Integrated Development Environment
- SRS -> Software Requirement Specification

1.8 Reference:

- An Integrated Approach Software Engineering Third Edition by Pankaj Jalote.
- Java :- Balaguru swamy
- SQL :- JosephL Jorden

1.9 Overview

The implementation of Library Management starts with entering and updating master records like book details, library information. Any further transaction like book issue, book return will automatically update the current books.

2. Overall Description

2.1 Product Perspective:

The proposed Library Management System will take care of the current book detail at any point of time. The book issue, book return will update the current book details automatically so that the user will get the updated current book details.

2.2 Product function:

- The main purpose of this project is to reduce the manual work.
- This software is capable of managing Book Issues, Returns, and Calculating/Managing Fine.
- Generating various Reports for Record Keeping according to end user requirements

2.3 User characteristics:

We have 2 different types of modules

- **User module**:
    - Students can register themselves and after registration they will get student_id.
    - After login, students can view their own dashboard.
    - Students can update their profile.
    - Students can view issued books and book return date-time.

    - Students can also change their own password.
    - Students can also recover their own password.

- **Administration module**: The following are the sub module in the administration module.
    - Unlimited User Accounts.
    - End User Access Permissions can be specified by the Admin.
    - Fetching Books
    - Alerts to email related to book return date, fines etc.,
    - Newsletters regarding newly added books are sent periodically.
    - Book Issue, Return, Fine Calculation.

2.4 General Constraints:

Any update regarding the book from the library is to be recorded to have update & correct values.

2.5 Assumption and dependencies:

All the data entered will be correct and up to date. This software package is developed using java as front end which is supported by the sun micro system. Microsoft SQL server 2005 as the back end which is supported by Window 7.

3. Specific Requirement:

3.1 External Interface Requirement:

The user should be simple and easy to understand and use. Also be an interactive interface .The system should prompt for the user and administrator to login to the application and for proper input criteria

3.1.1 User Interface:

The software provides a good graphical interface for the user any administrator can operate on the system, performing the required task such as create, update, viewing the details of the book.

- Allows users to view quick reports like Book Issues/Returns etc in between particular times.
- Stock verification and search facility based on different criteria.

3.1.2 Hardware interface:

- Operating system : Windows
- Hard disk : 40 GB
- RAM : 256 MB
- Processor : Pentium(R)Dual-core CPU

3.1.3 Software interface

- Java language
- Net beans IDE 7.0.1
- MS SQL server 2005

3.2 Functional requirements:

- Book entry: In this module we can store the details of the books.
- Register student: in this module we can keep the details of the new student.
- Book issue: This module is used to keep a track of book issue details.
- Book return: This module enables us to keep a track of returning the books.

3.3 Performance requirements:

The capability of the computer depends on the performance of the software. The software can take any number of inputs provided the database size is larger enough. This would depend on the available memory space.

3.4 Design constraints

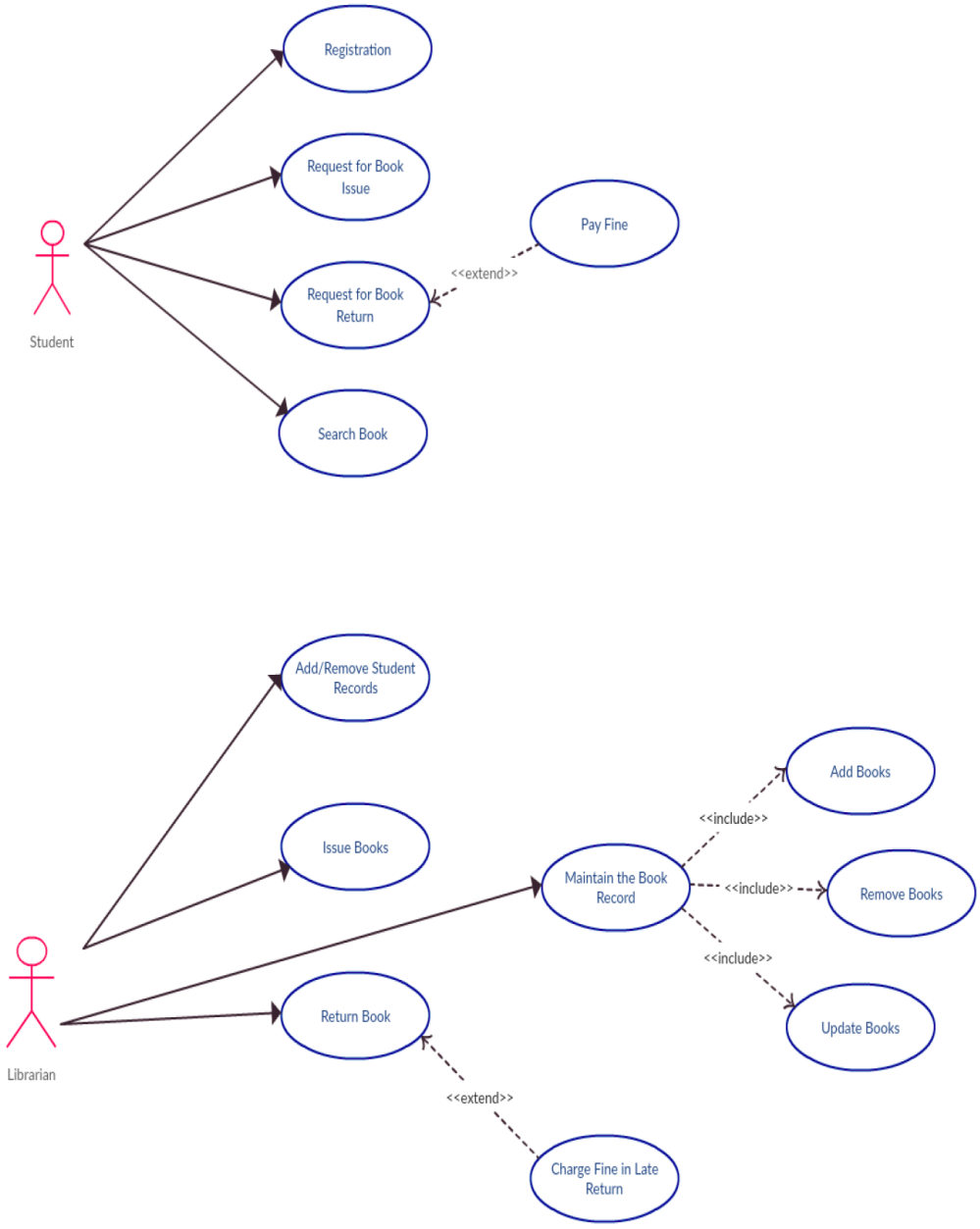
Each member will have an identity card which can be used for the library book issue, fine payment etc. whenever library members wish to take a book, the book issued by the library authority will check both the book details as well as the student details and store it in the library database. In case of retrieval of books much of human intervention can be eliminated.

3.5 System attributes

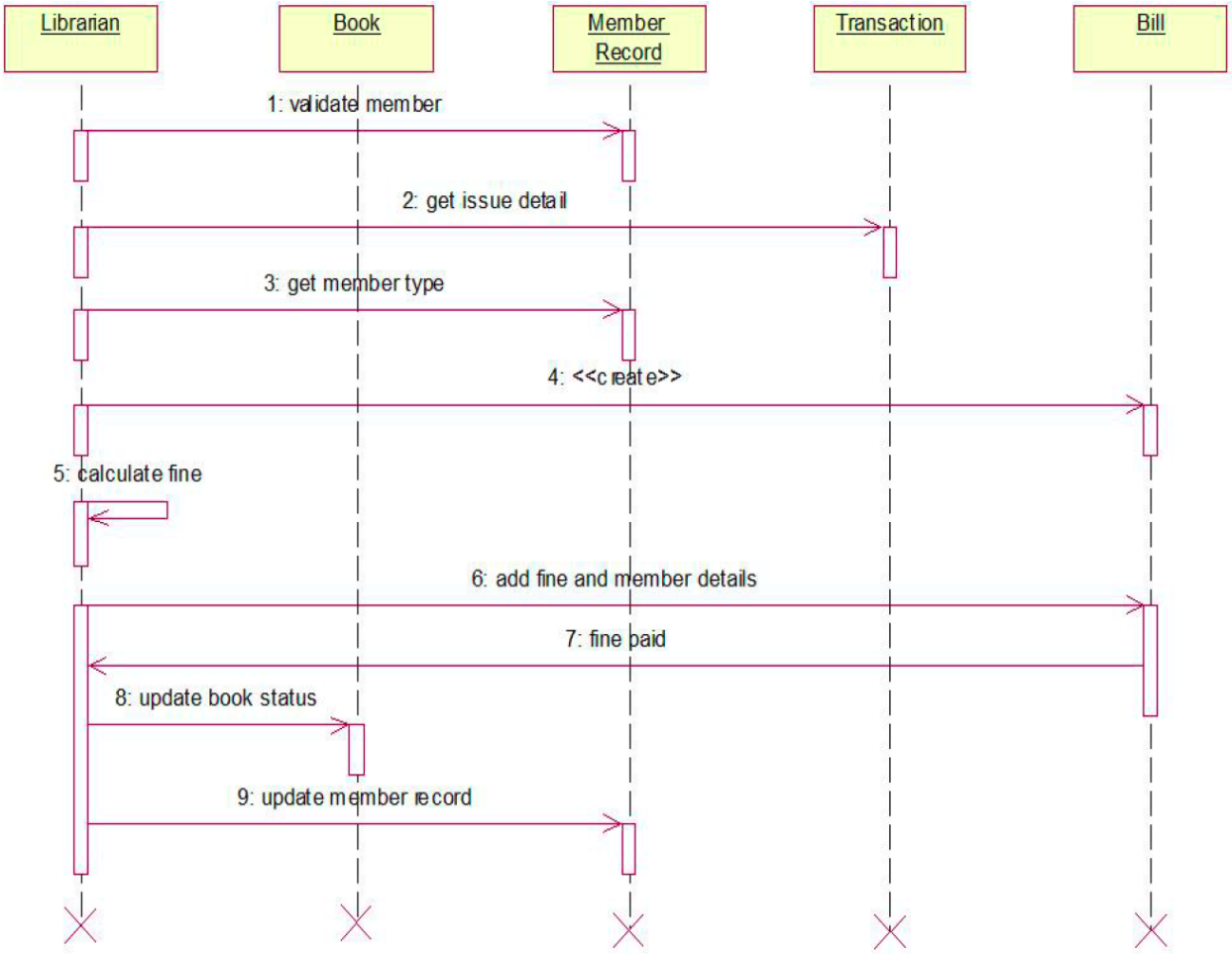
- **Maintainability**: There will be no maintained requirement for the software. The database is provided by the end user and therefore is maintained by this user.
- **Portability**: The system is developed for secured purpose, so it is can't be portable.
- **Availability**: This system will available only until the system on which it is install, is running.
- **Scalability:** Applicable.

4. Analysis Models:

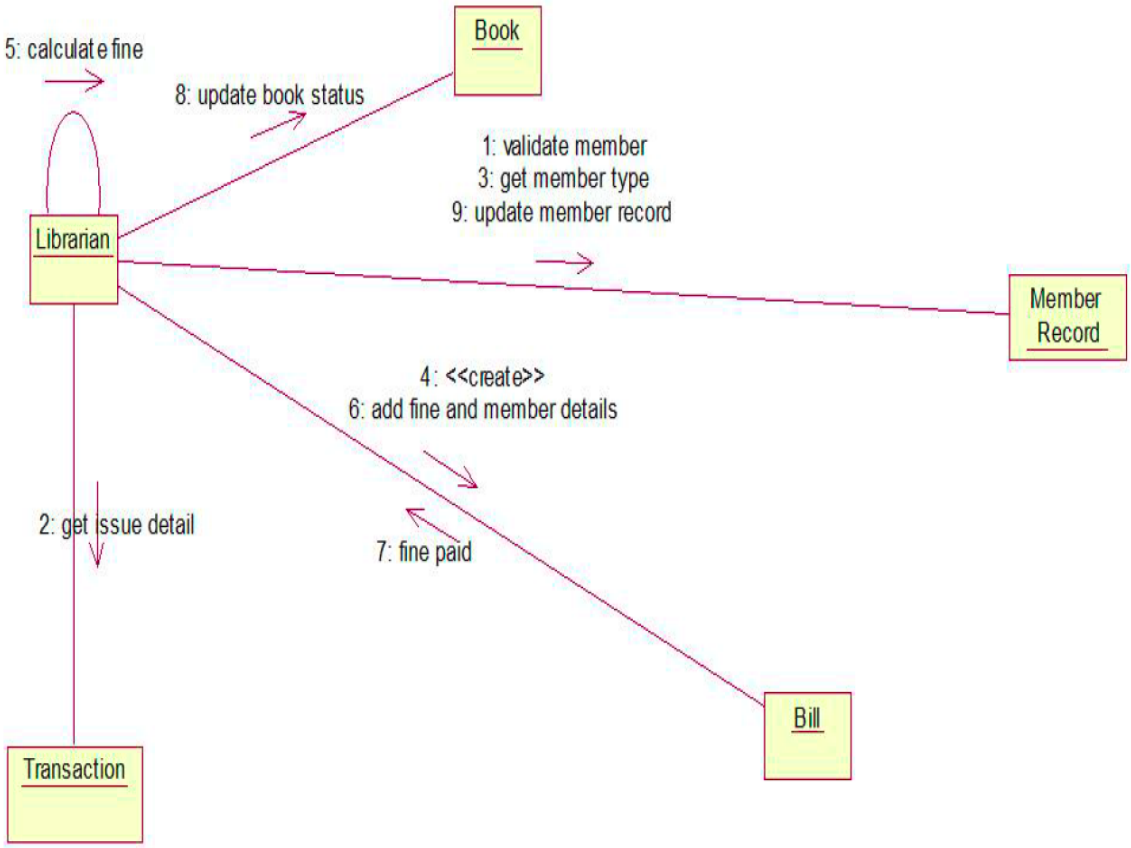
4.1 Use-Case Diagram



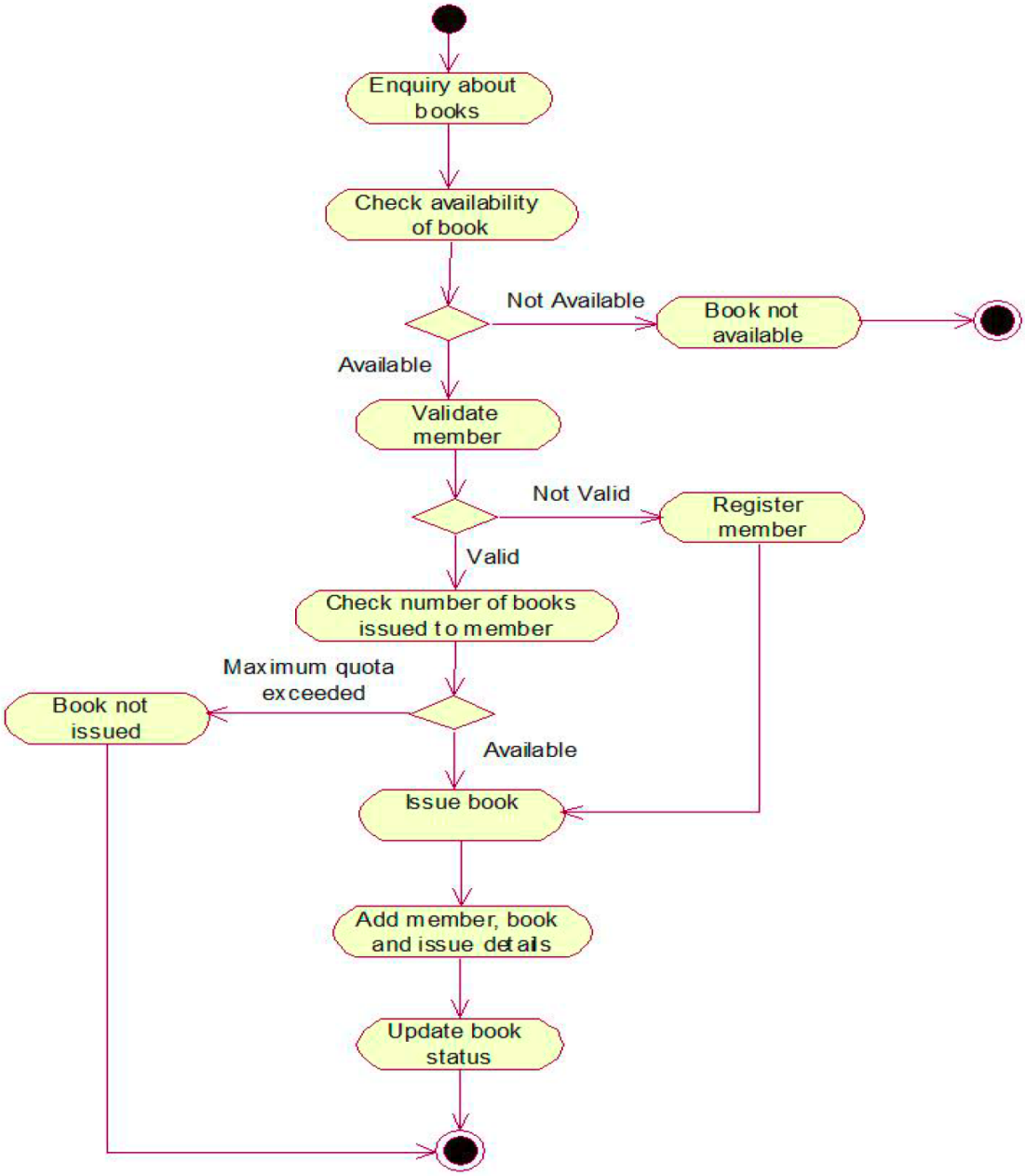
4.2 Sequence Diagram



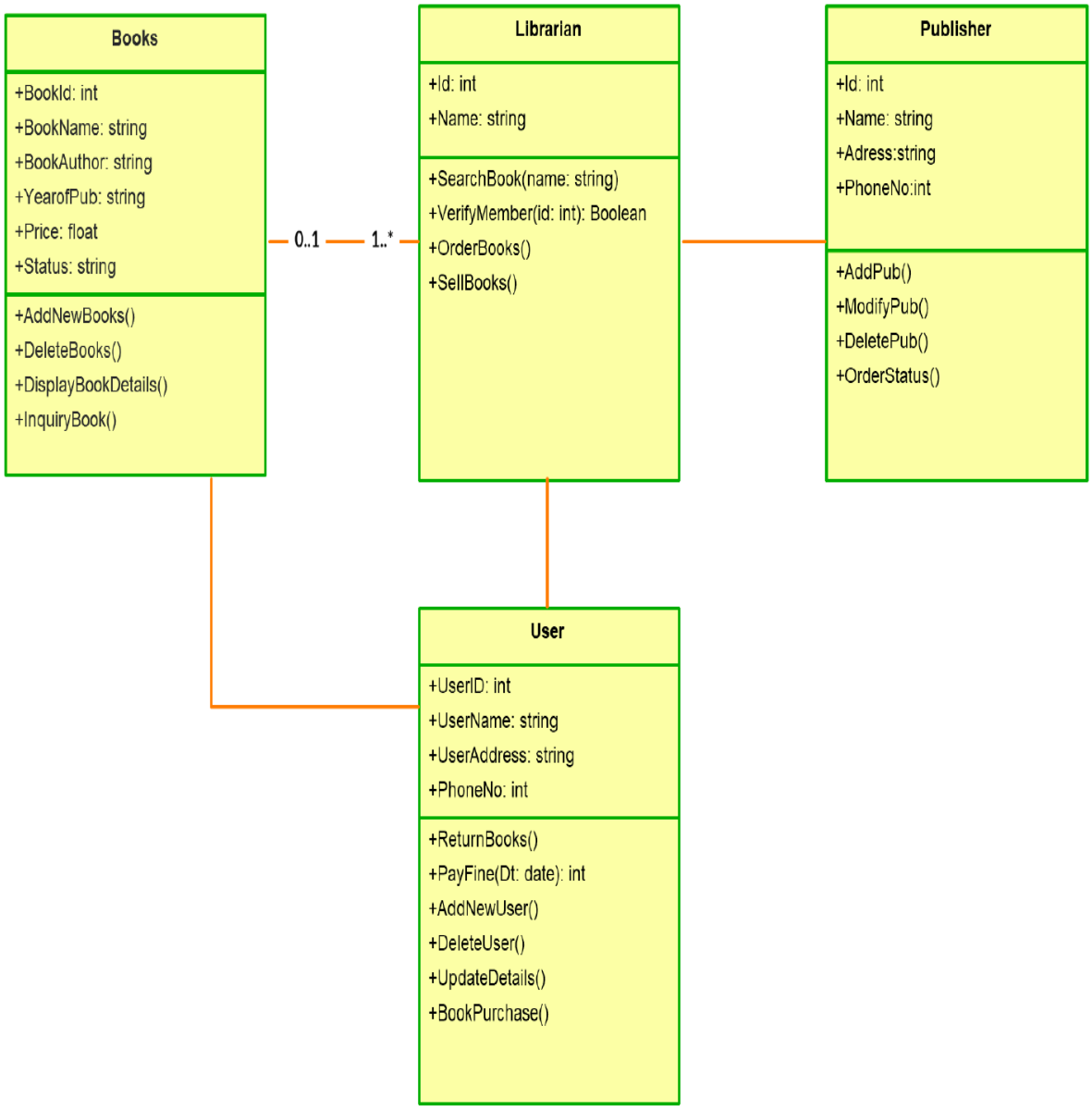
4.3 Collaboration Diagram



4.4 Activity Diagram



4.5 Class Diagram



4.6 Component Diagram