

THE UNIVERSITY OF BUEA

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

FACULTY OF ENGINEERING AND

TECHNOLOGY (FET)

DEPARTMENT OF COMPUTER

ENGINEERING

COURSE TITLE; OBJECT ORIENTED

MODELING

COURSE CODE: CEF 331

**LIBRARY AUTOMATING SYSTEM**

PROJECT REPORT

Group 4

**Name**

**Matricule**

DJEUNOU DJEUNOU MARIEKE JETTIE

FE21A168

DJEUTIO QUO

IMON ANDERSON ROY

FE21A169

DJIALEU TANKOUA KEANE R

ANDY

FE21A170

DJITUE TOGUE BRINDA SPACHELLE

FE21A171

DJOUMESSI DONGMO RONSARD CARNEGIE

FE21A172

DJOUMESSI LEKANE WENDY FORTUNE

FE21A173

DJOUMESSI NGUEFACK IVAN

FE21A174

DUESENBERRY MBIKANG AGBORTAR AKO

FE21A175

EBAI ENOWNKU JANE

FE21A176

EBOMESUMBE

NGOLE DANDY BRADLEY

FE21A177

EDI EDISON FORNANG

FE21A178

EFUETANZOH ASONG RODERIC

FE21A179

EKPOMBANG TABENDANG

FE21A180

EKUNDIME GLEAN MAKOGE

FE21A433

ELOMBA KARESLY SAKWE

FE21A181

Course lecturer

err

:

Dr FOZIN & DR djouela ines

# INTRODUCTION

Library management is difficult because librarians have to keep the library organized and stocked. Poorly managed libraries can have many problems such as disorganized shelves, unfilled book orders, outdated reference and more. It is where **Library automating software** helps extensively. It eliminates the need for a human being to do this monotonous task of organizing and updating catalog.

# FUNCTIONAL REQUIREMENTS

We have three types of functional requirements which are;

* System requirements
* User requirements
* System Analyst

#  SYSTEM REQUIREMENTS

* Keep track of books available in the library.
* Record name, identification number and password of each customer.
* Add and acquire new books digitally.
* Allows members to login, search, select, issue and return books by themselves.
* Determine when materials are due to be return.
* Keep track of financial obligations.
* Send notifications to customers.
* Provide certain criteria which makes the customer illegible to borrow a book.
* Store information about the types books highly demanded by customer at a given time.

#  USER REQUIREMENTS

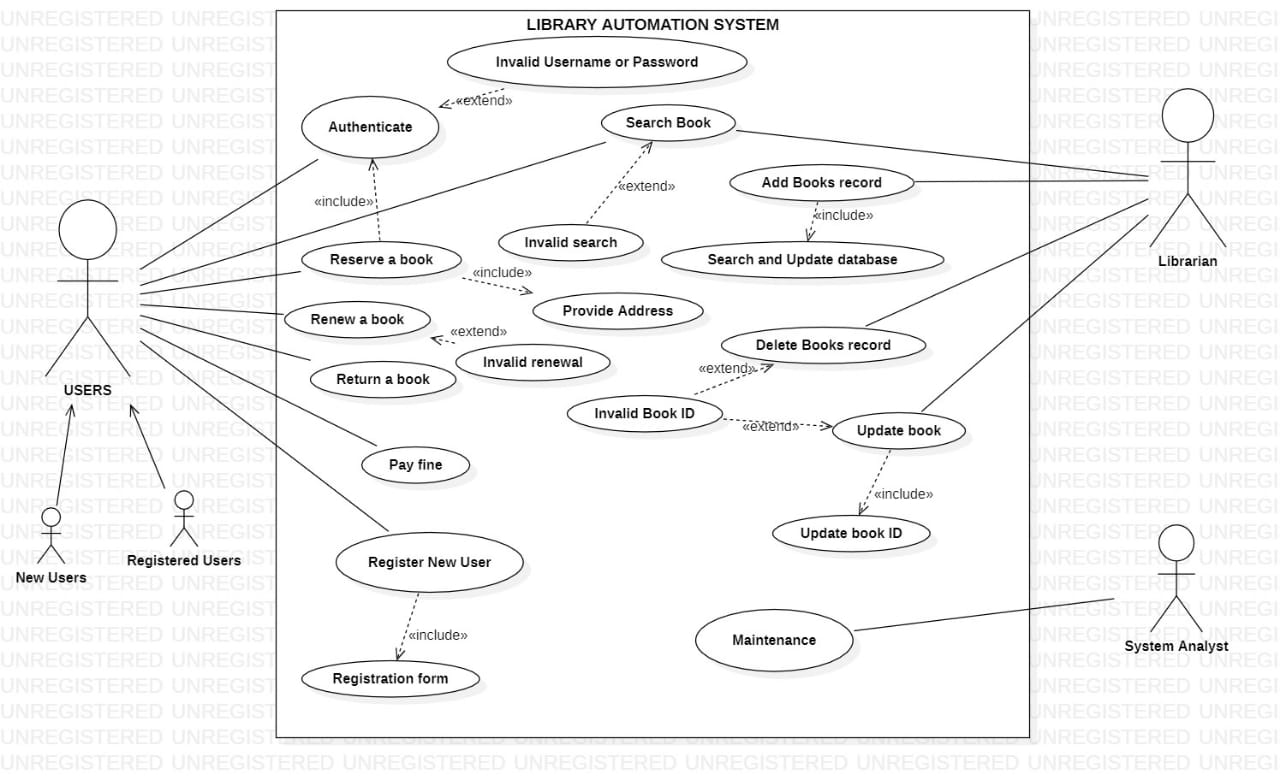
* Customers should be able to borrow a book.
* Customer should be able to research a book by title or author.
* Customer should be able to pay fines.
* Customer should be able to authenticate.
* Admin or Liberian should be able to check books availability.
* Customer and admin should be able to search or locate a book by the unique ID code, title, and author or publication date.

 SYSTEM ANALYST REQUIREMENTS

* Constantly makes sure that various parts of the system are are functioning properly.
* Repair any malfunctioning part of the system

* **Use Case Diagram For The LA(Library Automation) SOFTWARE :**
* The use case diagram is the diagram that summarizes the relationship between actors (, librarian, and the various users generalized as ) and the various use cases.

* The various actors are : **System Analyst**, **Librarian** and the **Users**(Registered and unregistered users ).



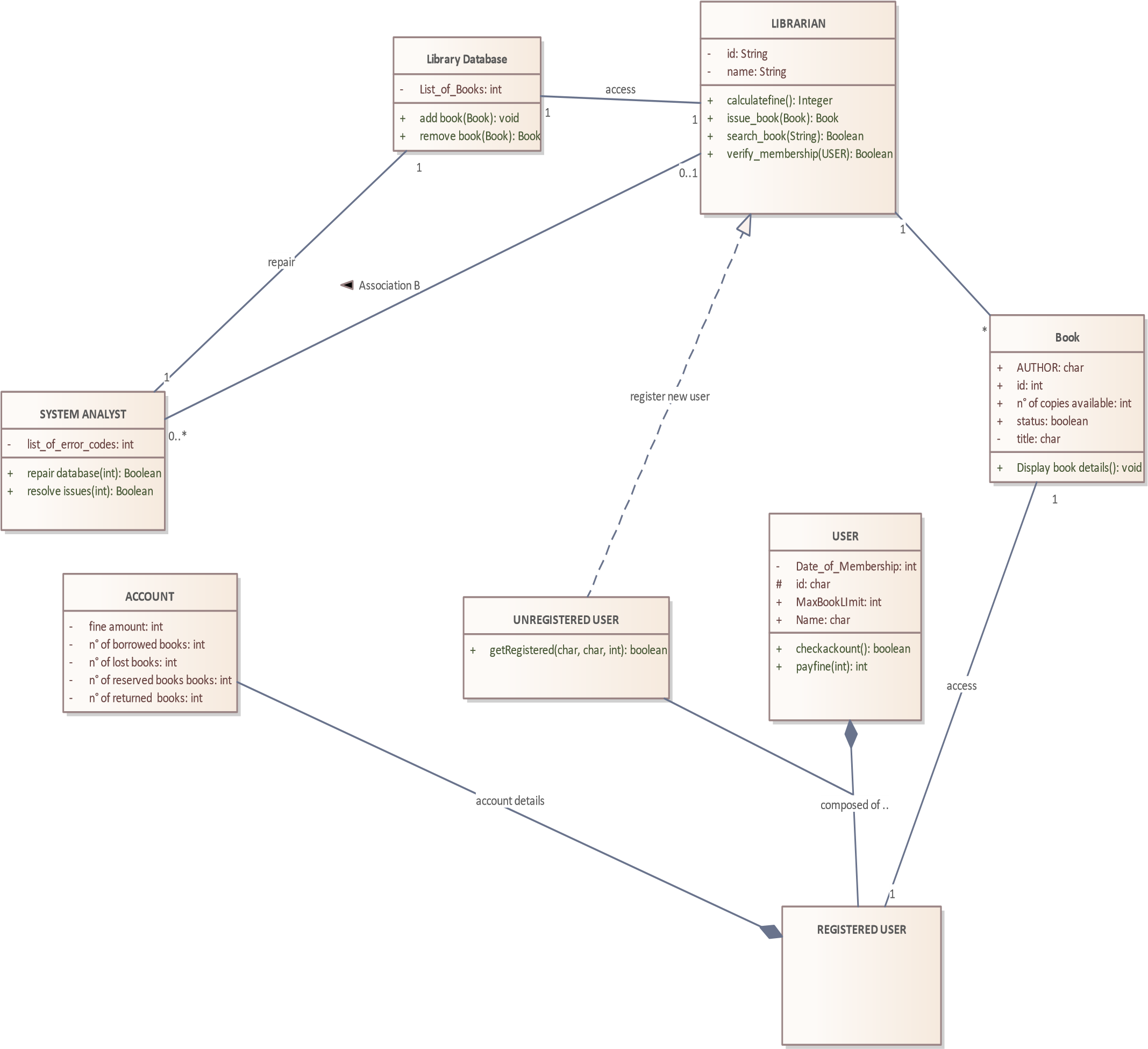
* **Class Diagram For The LA(Library Automation) SOFTWARE :**
* Describes the different types of objects and the static relationship that exists between them.
* User Class: It manages all the operations that a user can perform with respect to our Library automation system.
* Book Class: The book class manages all the operations a user as well as a Liberian can perform on a book in the library
* Liberian Class: The Liberian class manages all the operations performed by a Liberian in the system.
* Transaction Class: This class enables a Liberian to initiate a payment in the system
* Bill Class: This class enables a Liberian to create a document containing the amount of a fine paid by a specific user

**Attributes of the Library automation**

* User Class: Name, telnumber, password, Address, DateofMembership, MaxBookLimit, UserId.
* Book Class: Title, Author, Bookid, Status, publication\_date.
* Liberian Class: Name, password
* Transaction Class: Transid, Name, Userid, date\_of\_trans, Book\_id, Book\_title
* Bill Class: BillNumber, Userid, amount.

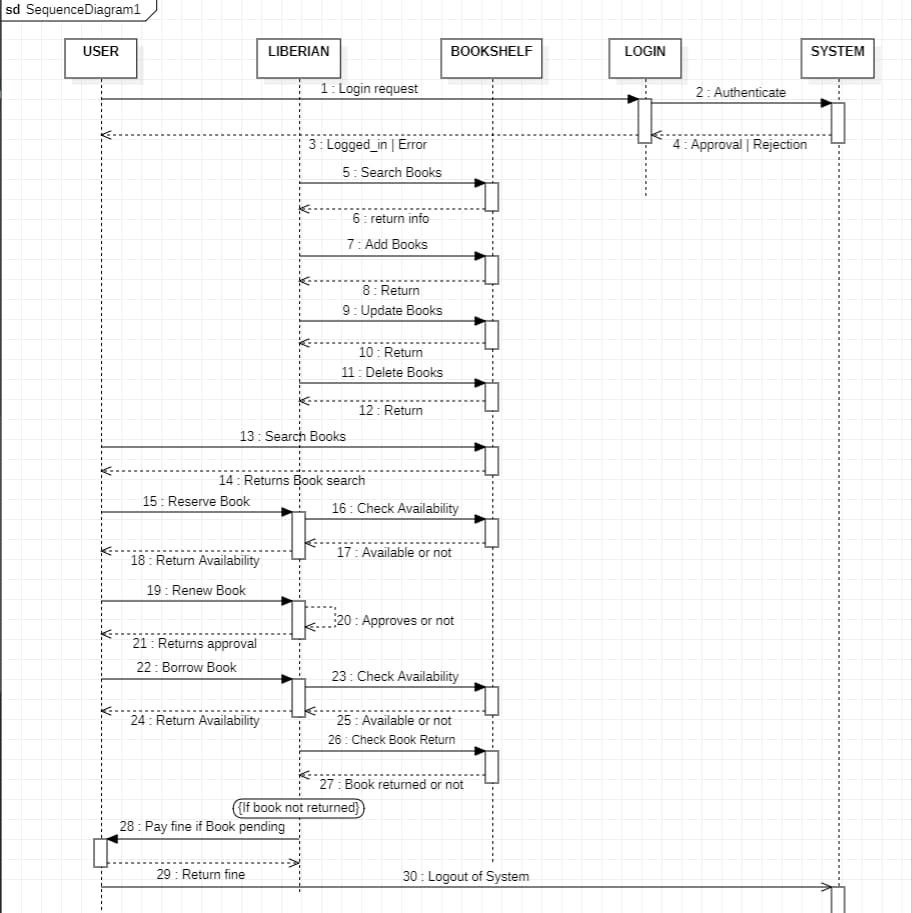
**Methods of the Library Automation**

* User Class: Verify(), search\_Book(), payBill().
* Book Class: DisplayBookDetails(), UpdateStatus()
* Liberian Class: Search\_Book(), VerifyMembership(), issueBook(), calculateFine(), createBill().
* Transaction Class: createtransaction(), DeleteTransaction()
* Bill Class: BillCreate(), BillUpdate()



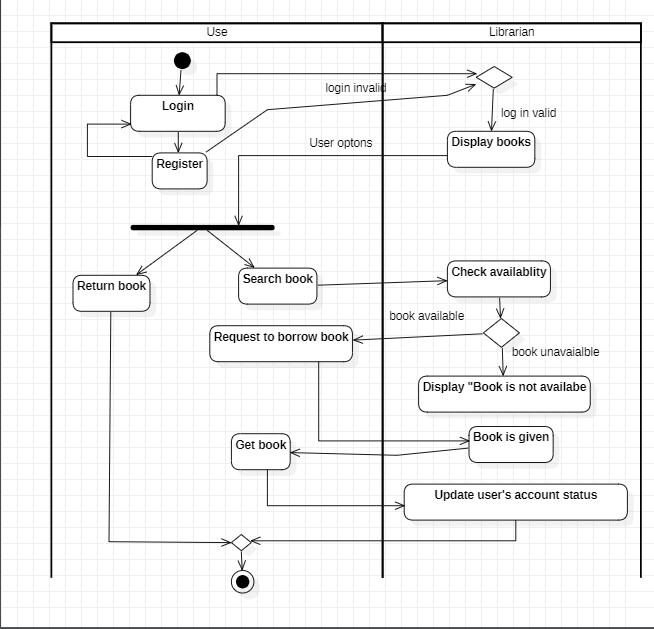
**Sequence Diagram For The LA(Library Automation) SOFTWARE :**

* Sequence diagrams detail how operations are performed.
* Capture interaction between objects in the context of collaboration.



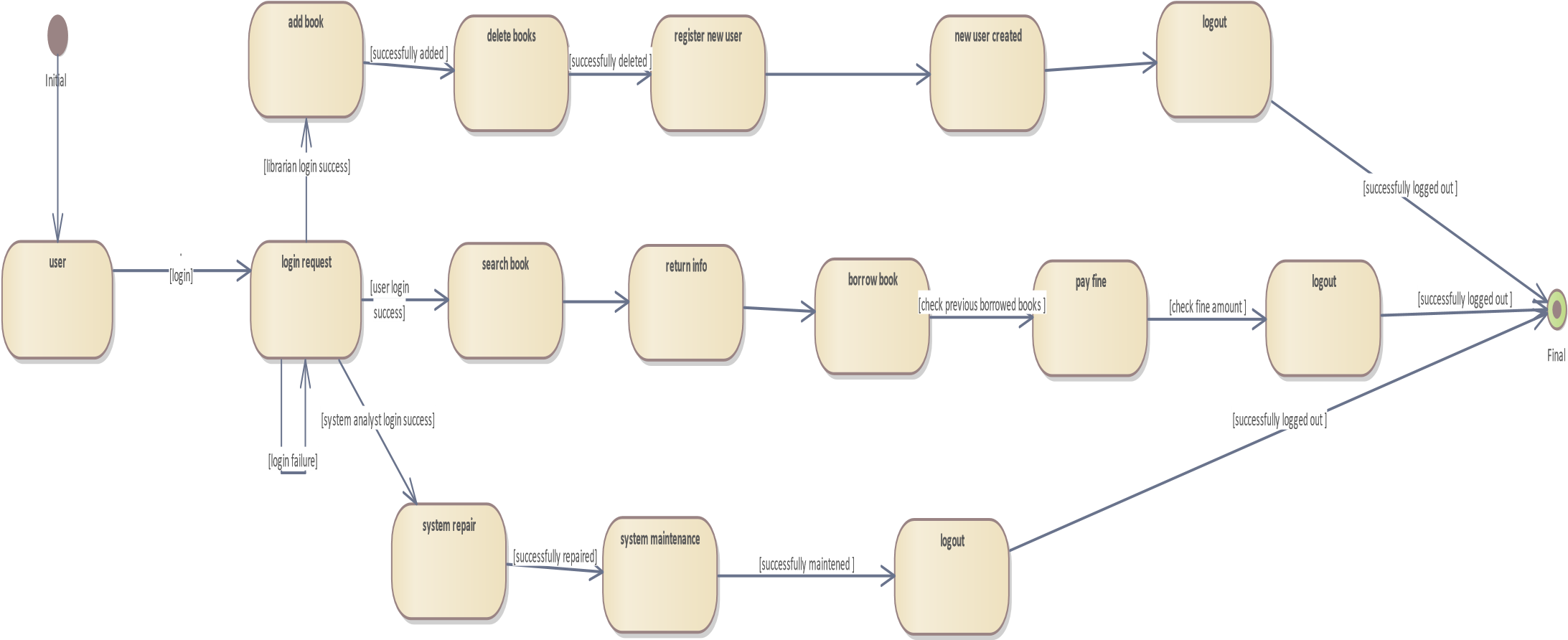
**Activity Diagram For The LA(Library Automation) SOFTWARE :**

* Actitivity Diagram models the transition from one actitiy to another.

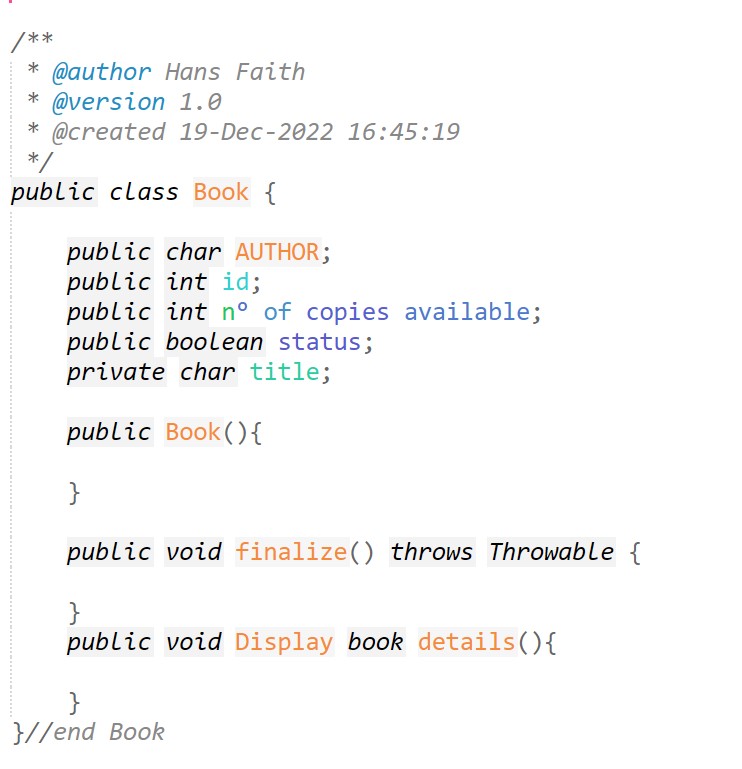


**State Diagram For The LA(Library Automation) SOFTWARE :**

* State machine diagram shows the various states of a system and how the states react to various events.
* State machine describes all events in relation with a single object.



**Java Code for Book Class**



**Java Code for Account Class**



**Java Code for User Class**

