MODEL DRIVEN SOFTWARE ENGINEERING (COEN 6312)

WINTER 2016

Deliverable 3: Class Diagram

Submitted to

Professor Abdelwahab Hamou-Lhadj

By

Team - High5

Jithin Nair(27093888)
NandKishore(27644396)
Renuka Milkoori(27188196)
Swamy Yogya Reddy(27170386)
Sai Teja(27757824)

Table of Contents

1. Introduction

The document describes the Class diagram with classes, objects, relationship among the artifacts. OCL constraints are applied on the level of classes and its semantics are formulated on the level of objects in the model of E-Library System. Each class comprises of attributes and operations which share common features and characteristics. OCL plays a crucial role in describing the constraints on the models of these classes.

2. Purpose

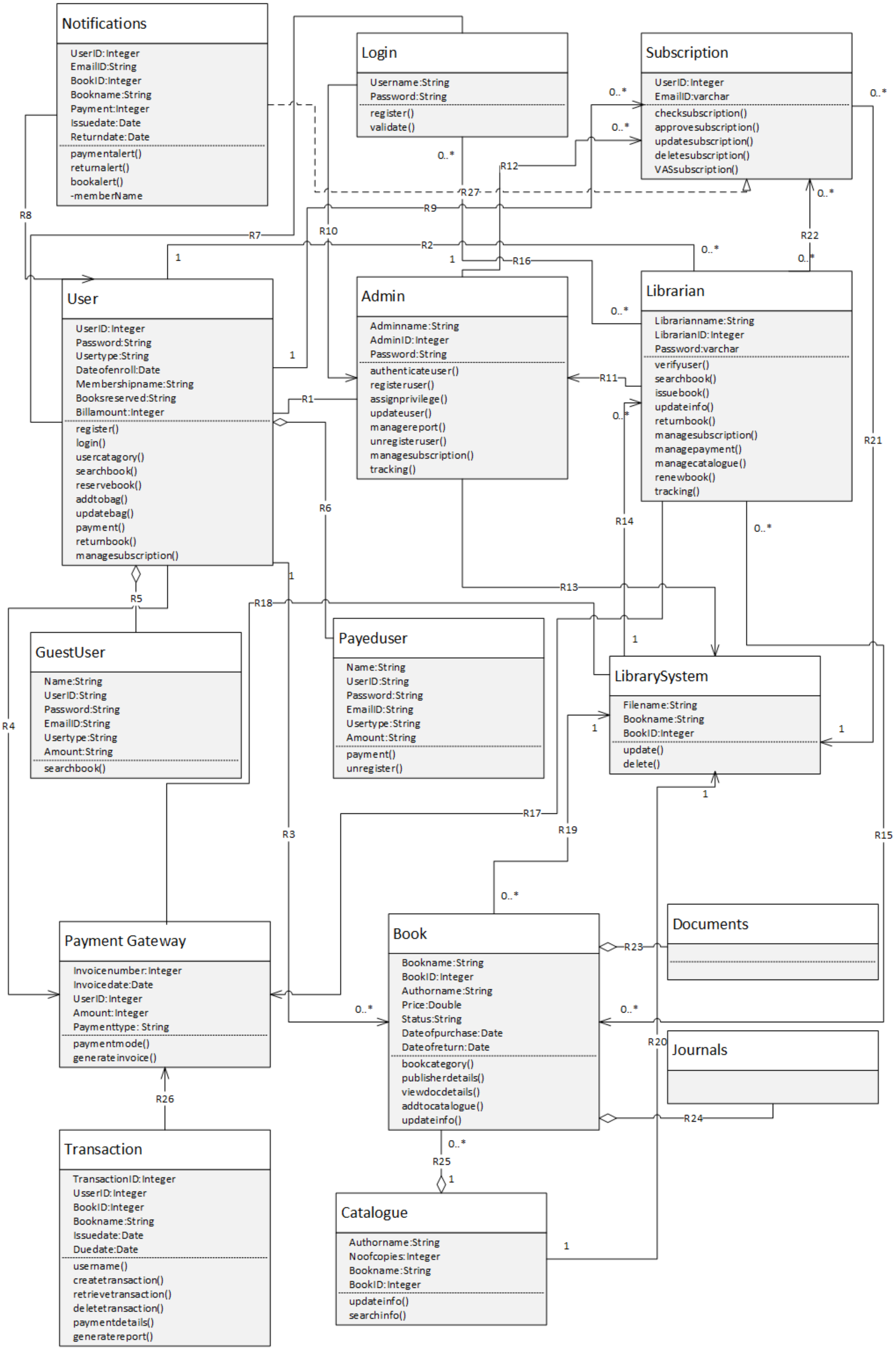
A Class diagram in Unified Modelling Language is a static diagram which represents the domain analysis and the detailed design of static view of the system. These diagrams helps in direct association with the object oriented languages consisting of attributes and operations and the relationship among them. Besides, OCL is a declarative language, which imposes certain constraints about the object oriented models and associated artifacts.

3. Class Diagram

Aforementioned is the list of all concepts for the E-Library System along with a description and the relationships they may exist between them

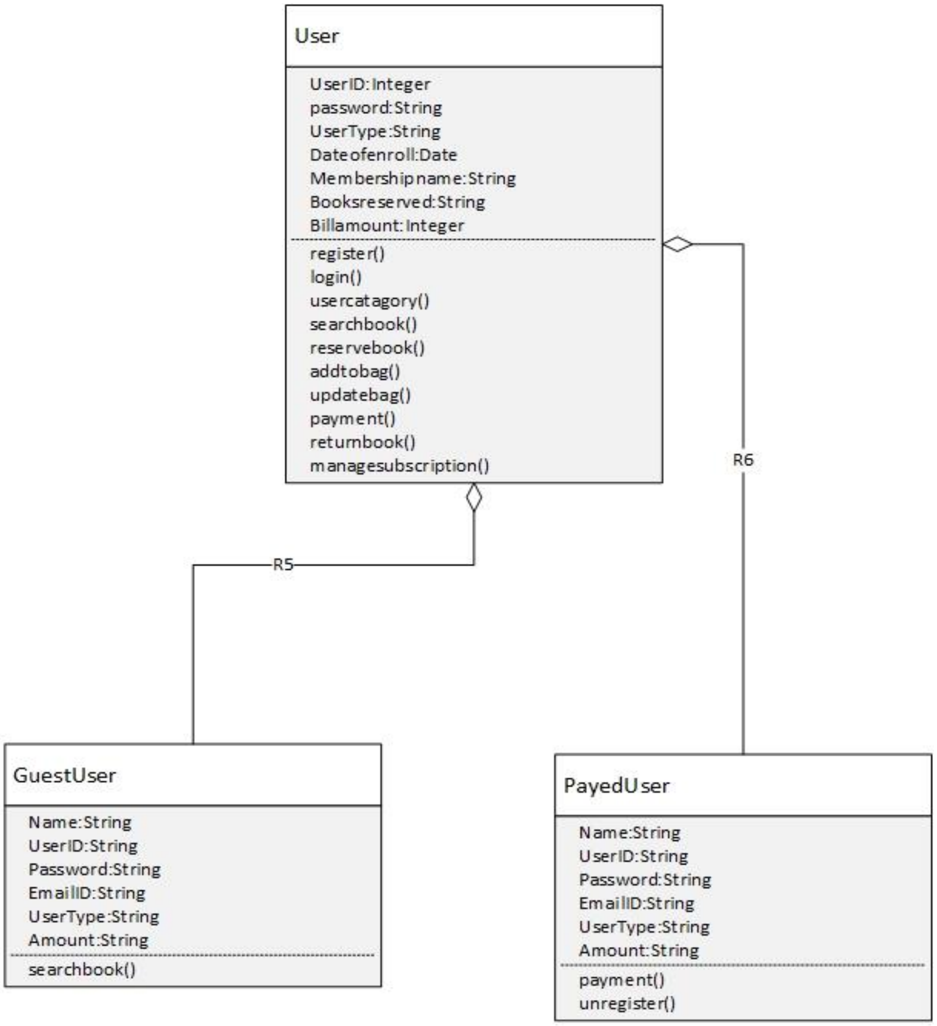
Note:

- *We have refrained from showing UML Note and Dependency to avoid a congested diagram.*
- *Also, only the necessary associations are represented. Example, if A is associated with B, B is associated with C and A is associated with C, we have not shown association between A and C, and we assume the flow or retrieval of information is thus derived.*


Notifications
UserID:Integer
EmailID:String
BookID:Integer
Bookname:String
Payment:Integer
Issuedate:Date
Returndate:Date
paymentalert()
returnalert()
bookalert()
-memberName
Login
Username:String
Password:String
register()
validate()
Subscription
UserID:Integer
EmailID:varchar
checksubscription()
approvesubscription()
updatesubscription()
deletesubscription()
VASsubscription()
0..*
0..*
0..*
0..*
R12
R27
R9
R8
R7
R10
R2
R16
R22
0..*
0..*
1
1
User
UserID:Integer
Password:String
Usertype:String
Dateofenroll:Date
Membershipname:String
Booksreserved:String
Billamount:Integer
register()
login()
usercatagory()
searchbook()
reservebook()
addtobag()
updatebag()
payment()
returnbook()
managesubscription()
1
R1
Admin
Adminname:String
AdminID:Integer
Password:String
authenticateuser()
registeruser()
assignprivilege()
updateuser()
managereport()
unregisteruser()
managesubscription()
tracking()
R11
Librarian
0..*
Librarianname:String
LibrarianID:Integer
Password:varchar
verifyuser()
searchbook()
issuebook()
updateinfo()
returnbook()
managesubscription()
managepayment()
managecatalogue()
renewbook()
tracking()
0..*
R21
R6
R14
0..*
R13
1
R5
R18
GuestUser
Name:String
UserID:String
Password:String
EmailID:String
Usertype:String
Amount:String
searchbook()
R4
Payeduser
Name:String
UserID:String
Password:String
EmailID:String
Usertype:String
Amount:String
payment()
unregister()
1
LibrarySystem
Filename:String
Bookname:String
BookID:Integer
update()
delete()
1
1
1
R17
R3
R19
R15
0..*
Payment Gateway
Invoicenumber:Integer
Invoicedate:Date
UserID:Integer
Amount:Integer
Paymenttype: String
paymentmode()
generateinvoice()
0..*
Book
Bookname:String
BookID:Integer
Authorname:String
Price:Double
Status:String
Dateofpurchase:Date
Dateofreturn:Date
bookcategory()
publisherdetails()
viewdocdetails()
addtocatalogue()
updateinfo()
R23
Documents
0..*
R20
Journals
R24
R26
Transaction
TransactionID:Integer
UsserID:Integer
BookID:Integer
Bookname:String
Issuedate:Date
Duedate:Date
username()
createtransaction()
retrievetransaction()
deletetransaction()
paymentdetails()
generatereport()
0..*
R25
1
Catalogue
Authorname:String
Noofcopies:Integer
Bookname:String
BookID:Integer
updateinfo()
searchinfo()
1

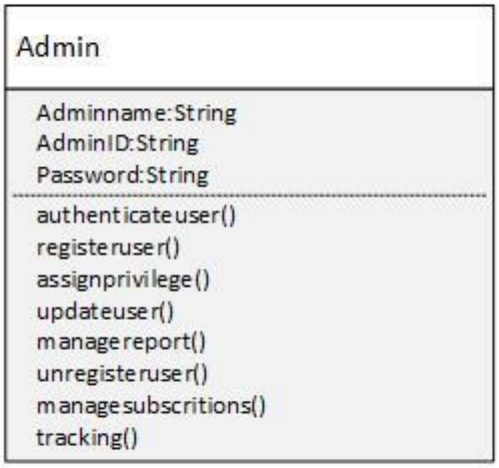
4. Description of Classes

1. User



The user has to register to the access the E-Library System. Admin grants privileges to the user and then the user can login to the system. There are two types of users namely, Payed users and Guest users. Once the user logs into the system, the user can to search for the books of their interest, reserve or renew the book, then add to bag and update the bag. Then the user can go through payment options to complete the book reservation. The user can cancel the reservation on the book, if the user is not interested or reserved a wrong edition of the book.

2. Admin



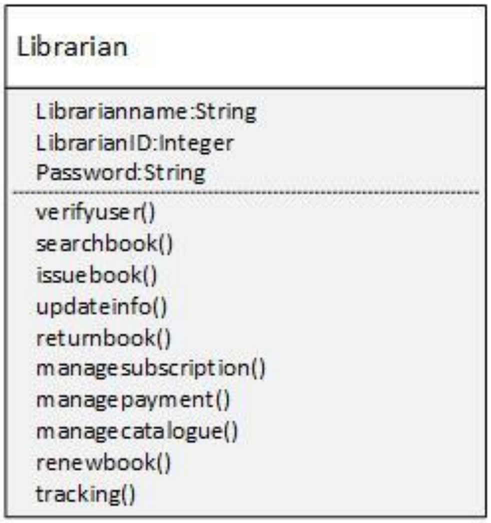
Admin

Adminname:String
AdminID:String
Password:String

authenticateuser()
registeruser()
assignprivilege()
updateuser()
managereport()
unregisteruser()
managesubscritions()
tracking()

Admin has the access towards the system and to validate users who are registered. Admin authenticates the user and assigns privileges in proportion to the user’s interest. Admin also has the permission to unregister the user. Admin tracks the data of the user and generates reports as well as manage subscription to keep them up to date.

3. Librarian



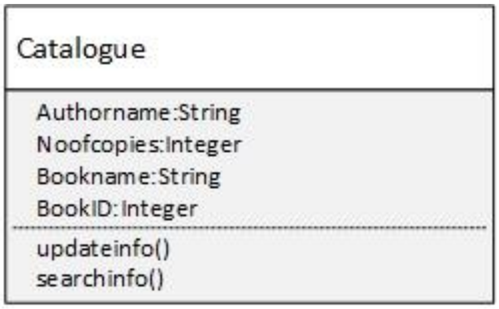
Librarian

Librarianname:String
LibrarianID:Integer
Password:String

verifyuser()
searchbook()
issuebook()
updateinfo()
returnbook()
managesubscription()
managepayment()
managecatalogue()
renewbook()
tracking()

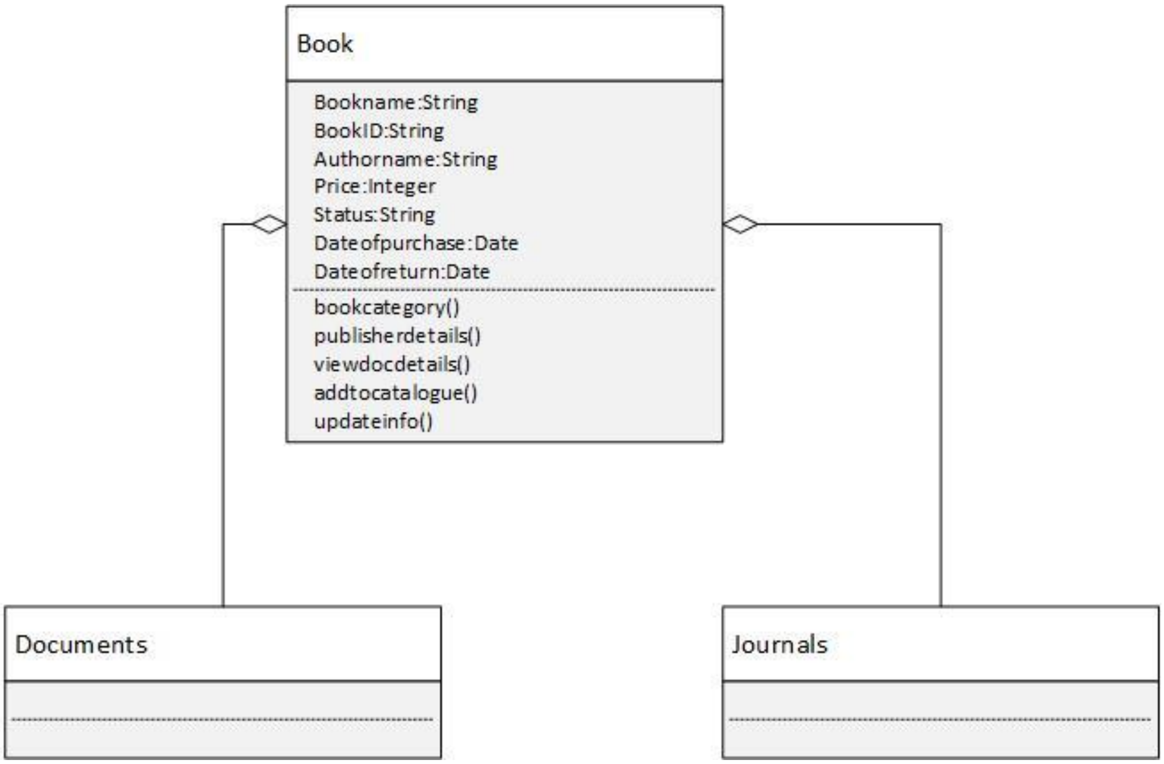
Librarian verifies the user's account and issues the books when the user reserves the book. Librarian has the permission to add and remove the books to update the catalogue. Librarian keeps the track of user data such as notifying deadlines and manages payment when the user has to return or renew the book.

4. Catalogue



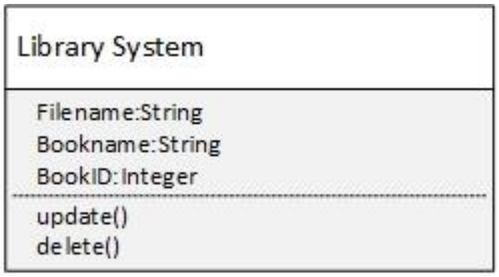
The catalogue includes the updated books by the librarian. The user has to search the catalogue to find the book of his/her interest so that the user can reserve it.

5. Book



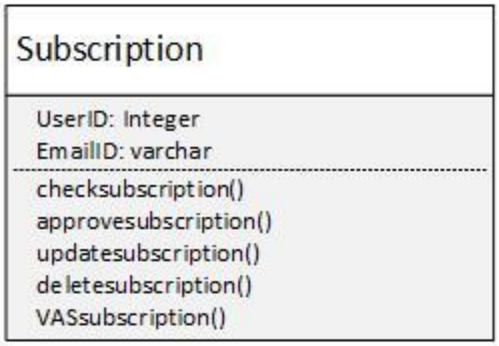
Books provided in the catalogue are classified so that the user can find the book easily. Classification contains publisher details as wells as the book details. When the user gets the book of his preference, a user adds it to catalogue and then librarian updates the records.

6. Library System



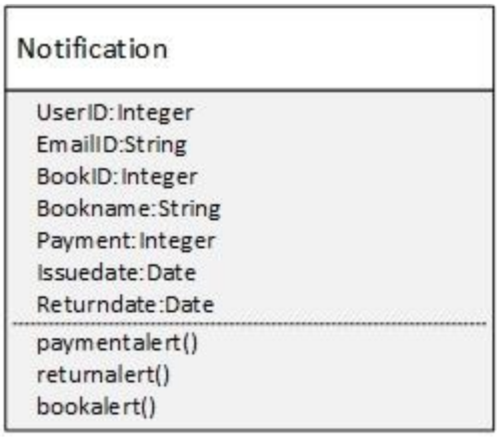
Library system updates and deletes the database, whenever user reserves, return or renew the books.

7. Subscription



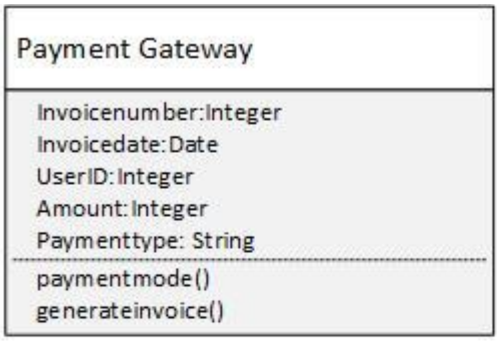
The user's subscription is validated and then approved by the admin as well as the librarian. They also manage updating and deletion of the user subscriptions.

8. Notification



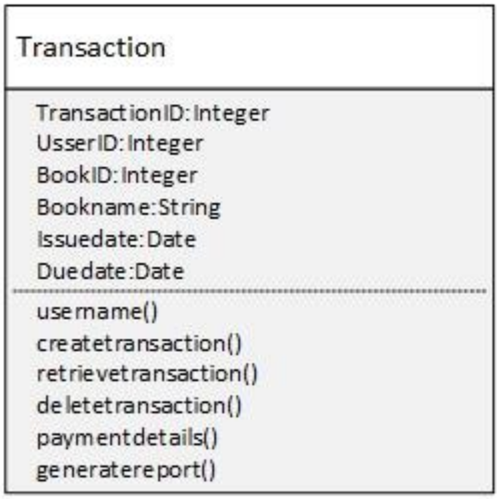
The user has to subscribe for the notifications and will have issue alerts, payment alerts, return alerts.

9. Payment Gateway



Payments are handled by the Payment Gateway, when the user wants to reserve, return or renew the book. And also, when the librarian wants to refund the payment to the user.

10. Transaction



Transactions are processed between the user and the Payment Gateway service application. Whenever a user performs a transaction the application generates the receipt of the payment details accordingly and also updates the report.

5. OCL Expressions

1) A User should have userID and password to access the system.

    Context User
    inv: self.userID → notEmpty() and self.password → notEmpty()

2) No two users should have same userID.

    Context User
    inv: self.forAll(u1,u2:users u1<>u2 implies u1.userID<>u2.userID)

3) A catalogue must contain atleast one book.

    Context Catalogue
    inv: self.R25 → notEmpty()

4) A library system must have one admin and can have one or more librarian.

Context Librarian
inv: self.R13 → size()=1 and self.R14 → size()>=1

5) Admin must have admin id, password and privileges authority to users.

Context Admin
inv: self.AdminID → notEmpty() and self.password → notEmpty() and
self.R1.allInstances → select (u : users | UserType : Payeduser)

6) A librarian should have a Librarian ID, username and password.

Context Librarian
inv: self.LibrarianID → notEmpty() and self.username → notEmpty() and
self.password → notEmpty()

7) A book should have unique bookid.

Context Book
inv: self.forAll(b1,b2:books b1<>b2 implies b1.bookID<>b2.bookID)

8) A Librarian and User must get notifications based on the subscription.

Context User
inv: self.R22.R9.R21 → includesAll(self.R22.R9.R27)

9) List of Users who are registered into the Library system

Context Admin
inv: self.R1.R5.R6 → forAll(Userid<>NULL)

10) Admin should check whether the user is Guest or Payed users.

Context Admin
inv: self.R1 → forAll(u1,u2: User u1<>u2 | u:usercategory)

11) List of books available in the LibrarySystem

Context Librarian
inv: self.R15 → forAll(bookID<>NULL)

12) A user can search and select book in the system based on the availability.

Context User
inv: self.R3 → notEmpty() and self.R3 → size()>=1

13) User can renew the book before the return date.

Context User
inv: self.renewbook (rb: returnbook, rn:renewbook | implies rn≤rb)

14) User should do the payment on the reserve book.

Context Admin
inv: self.R4 → forAll(u : User, p.Payment| p.UserID = u.reservebook)

15) User must do the payment on the pending transaction

Context User
inv: self.R4.R18 → includesAll(self.R4.R26)

16) Add user to the list of registered users in the system by Admin

Context Admin:: addUser(a: User)

pre: self.R1 → excludes(a)

post: self.R1 → includes(a) and self.R1 → size() = self.R1 → size() @ pre + 1

17) Remove User from the list of registered users in the system by Admin

Context Admin:: removeUser(a: User)

pre: self.R1 → includes(a)

post: self.R1→ excludes(a) and self.R1 → size() = self.R1 → size() @ pre – 1

References

1. Domain modeling by prof. Kamthan:
http://users.encs.concordia.ca/~kamthan/courses/soen6481/domain_modeling_introduction.pdf