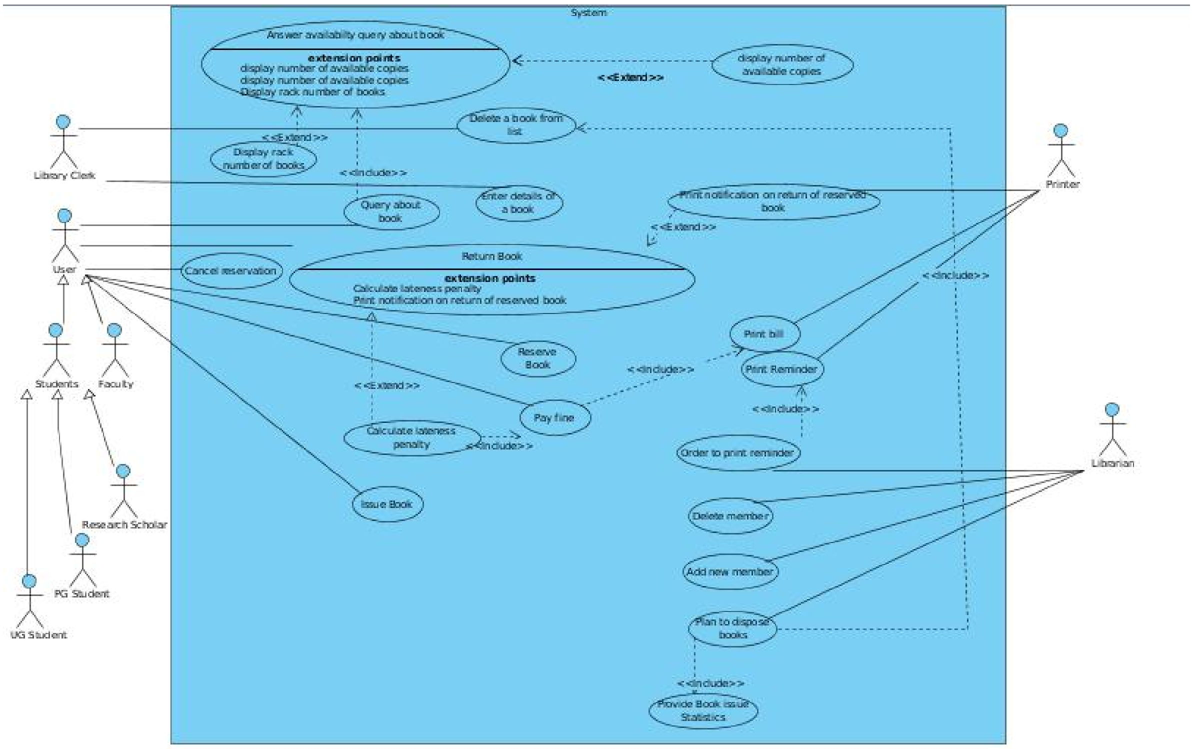
UML Diagrams

LIBRERIA
LIBRARY INFORMATION SYSTEM
Group Number : 52
Group members:

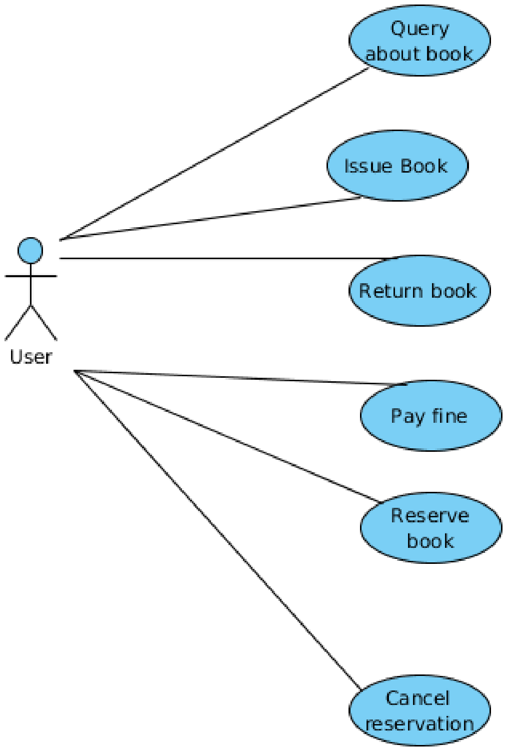
- Ashrujit Ghoshal (14CS10060)
- Sayan Ghosh (14CS10061)

Contents

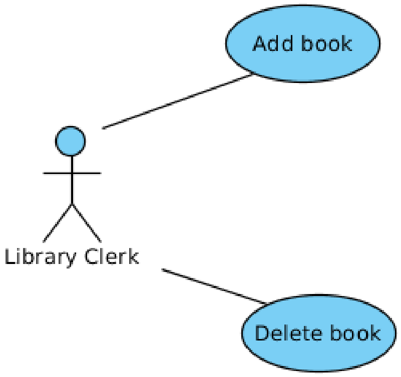
1 Use Case Diagram



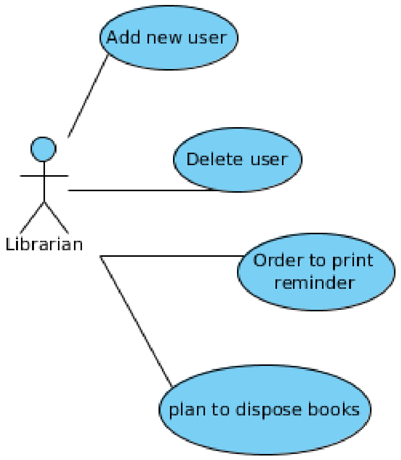
Use case of User :



Use case of Library Clerk :



Use case of Librarian :



Description :

1. User use cases:

   - Query about Book

     – Preconditions:
     1. The user must be logged in .
     2.The book must exist in the library

     – Postcondition:
     If the book exists in the library,the availablity status of the book is returned

     – Failure Situations:
     The library does not have the book

     – Postcondition in case of failure:
     A message to user about the same

     – Actors:
     User communicates with the system

     – Trigger:
     User chooses the option to search books

     – Main Success Scenario:
     The library has a copy of the book and it is available for issue.

  - Extensions/Variations:
    The library has a copy of the book but currently none of the copies are available.The book maybe reserved by the user.

- Issue Book

  - Preconditions:
    1. The user must be logged in .
    2.The book must exist in the library
    3.It must be available for issue.
    4.The user must not have exhausted his quota of number of books

  - Postcondition:
    After successful issue the user account is updated

  - Failure Situations:
    1. The library does not have the book
    2.The library has the book and it is not avaiable for issue.
    3.The user has exhausted his quota of maximum number of books

  - Postcondition in case of failure:
    In failure case 2. the user may choose to reserve the book if he has not exhausted his quota

  - Actors:
    User communicates with the system

  - Trigger:
    User chooses the option to issue books

  - Main Success Scenario:
    The library has a copy of the book and it is available for issue.

- Return Book

  - Precondition:
    1.User must be logged in.
    2.User must have previously issued the book.

  - Postcondition:
    1.If the book was overdue the penalty is calculated and a bill is printed
    2.In case the book was reserved by some other user, a notification is sent out to the other user.
    3.The user account is updated

  - Failure Situations:
    The user has not issued any book

  - Postcondition in case of failure:
    A message is give to the user about the same

- Actors:
  User communicates with the system

- Trigger:
  User chooses the option to return issued books

- Main Success Scenario:
  The user had previously issued the book

- Reserve Book

  - Preconditions:
    1. The user must be logged in .
    2.The book must exist in the library
    3.It must not be available for issue.
    4.The user must not have exhausted his quota of number of books

  - Postcondition:
    1.After successful issue the user account is updated
    2.When the book is returned a notification is sent to the user.

  - Failure Situations:
    1. The library does not have the book
    2.The library has the book and it is avaiable for issue.
    3.The user has exhausted his quota of maximm number of books

  - Postcondition in case of failure:
    In failure case 2. the user may choose to issue the book if he has not exhausted his quota

  - Actors:
    User communicates with the system

  - Trigger:
    User chooses the option to reserve book

  - Main Success Scenario:
    The library has a copy of the book and it is not available for issue.

- Cancel Reservation

  - Preconditions:
    1. The user must be logged in .
    2.The user must have reserved the book

  - Postcondition:
    1.After successful issue the user account is updated
    Failure Situations:

The user has not issued any book

  - Actors:
    User communicates with the system

  - Trigger:
    1.User chooses the option to cancel reservation of a book
    2.User does not issue the reserved book within 7 days of return

  - Main Success Scenario:
    The library has a copy of the book and the user must have reserved it previously

- Pay Fine

  - Precondition:

    (a) User must be logged in.

    (b) User must have previously issued the book.
    The book must be overdue

  - Postcondition:
    The book was overdue the penalty is calculated and a bill is printed
  - Failure Situations:
    (a) The user has not issued any book
    (b) No returned books are overdue
  - Actors: User communicates with the system
  - Trigger: User chooses the option to return issued books
  - Main Success Scenario: The user had previously issued the book and the book is overdue

2. Library Clerk Use Cases

- Enter details of a book
  - Preconditions:1. The clerk must be logged in.
    2.The book must not be previously entered in the system
  - Failure Situations:
    The book is already in the system
  - Postcondition in case of failure:
    A message to clerk about the same
  - Actors:
    library clerk communicates with the system
  - Trigger:
    Clerk chooses the option to enter new books
  - Main Success Scenario:
    The library does not have the book and the book is newly entered in the system
  - Extensions/Variations:
    The library hasthe book and the umber of copies is increased

- Delete a book
  - Preconditions:
    1. The clerk must be logged in.
    2.The book must be previously entered in the system
    3.The librarian has decided to dispose the book
  - Failure Situations:
    The book is not in the system
  - Postcondition in case of failure:
    A message to clerk about the same
  - Actors:
    library clerk communicates with the system
  - Trigger:
    Clerk chooses the option to delete books
  - Main Success Scenario:
    The library has the book and it is removed from the system.
  - Extensions/Variations:
    The library has the book and the number of copies is reduced.

3. Librarian Use Cases:

- Add new member
  - Preconditions:1.Librarian must be logged in
    2. A person must apply for membership
  - Postcondition:
    A new member account is created
  - Failure Situations:
    The user is already registered
  - Postcondition in case of failure:
    A message to librarian about the same
  - Actors:
    Librarian communicates with the system
  - Trigger:
    Librarian chooses the option to add member

  - Main Success Scenario:
    The user is not previously registered

- Delete member

  - Preconditions:
    1.Librarian must be logged in
    2. A person must apply for cacellation membership

  - Postcondition:
    The member account is deleted

  - Failure Situations:
    The user has no account

  - Postcondition in case of failure:
    A message to librarian about the same

  - Actors:
    Librarian communicates with the system

  - Trigger:
    Librarian chooses the option to delete member

  - Main Success Scenario:
    The user previously has an account

- Order to print reminder

  - Preconditions:
    1.Librarian must be logged in
    2. A book issued by a member must be overdue

  - Postcondition:
    A message is sent to the user.

  - Failure Situations:
    There are no overdue books

  - Postcondition in case of failure:
    A message to librarian about the same

  - Actors:
    Librarian communicates with the system

  - Trigger:
    Librarian chooses the option to print reminder

  - Main Success Scenario :
    There are some overdue books

- Plan to dispose books

  - Preconditions:
    1.Librarian must be logged in
    2. The book must not have been issued even once for 5 years

  - Postcondition:
    The book is disposed with a message to the library clerk to delete it.

  - Actors:
    Librarian communicates with the system

  - Trigger:
    Librarian chooses the option to dispose book

  - Main Success Scenario:
    The book has not been issue for 5 years

4. System Use Cases:

- Answer availibilty Query about Book

  - Preconditions:
    1.An user makes a query

  - Postcondition:
    If the book is available, use cases display rack number and number of copies are called

  - Actors:
    System communicates with the user

  - Trigger:
    User chooses the option to search books

- Display rack number of book

  - Preconditions:
    1.If the book is available the use case answer availibility query invokes this

- Postcondition:
  Rack numbers are displayed

- Actors:
  System communicates with the user

- Trigger:
  answer availibity query triggers this

- Display number of copies of book

  - Preconditions:
    1.If the book is available the use case answer availibility query invokes this

  - Postcondition:
    the number of copies of a book are displayed

  - Actors:
    System communicates with the user

  - Trigger:
    answer availibity query triggers this

- Calculate lateness penalty

  - Preconditions:
    1.If the book is overdue, return book invokes this

  - Postcondition:
    Penalty is calculated and print bill is invoked

  - Actors:
    System communicates with the user

  - Trigger:
    return book query triggers this

- Provide book issue statistics
  - Preconditions:
    1.The use case is invoked by plan to dispose books

  - Postcondition:
    Statistics of books is displayed

  - Actors:
    system communicates with librarian

    - Trigger:
      Plan dispose book is invoked

5. Printer use cases:

    - Print bill of penalty

        - Preconditions:
          Some user must have returned the issued book later than his designated return date.

        - Postcondition:
          Bill of penalty is printed

        - Actors:
          Printer communicates with the system

        - Trigger:
          Calculate lateness penalty triggers print bill

    - Print reminder

        - Preconditions:
          Some user must have exceeded the due date

        - Postcondition:
          Reminder to user is printed

        - Actors:
          Printer communicates with the system

        - Trigger:
          order to print reminder triggers this

    - Print notification on return of reserved book

        - Preconditions:
          Some user must have returned a reserved book

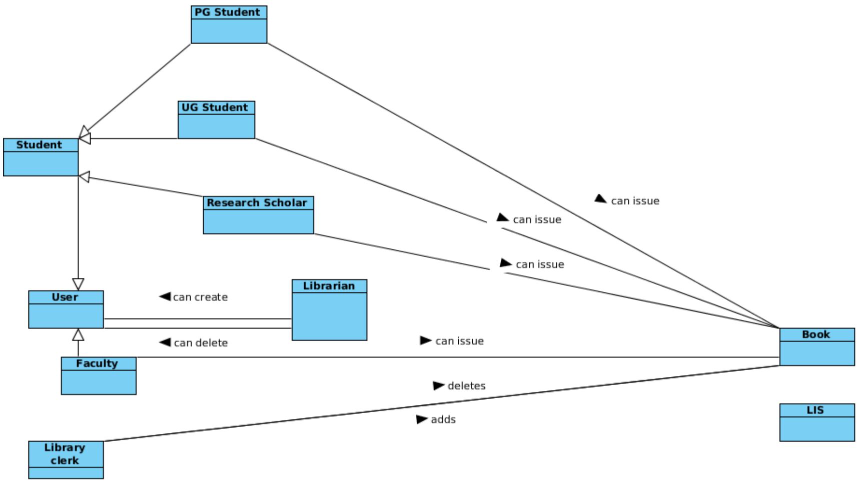
        - Postcondition:
          A notification is printed to the user who reserved the book

        - Actors:
          Printer communicates with the system which communicates with user

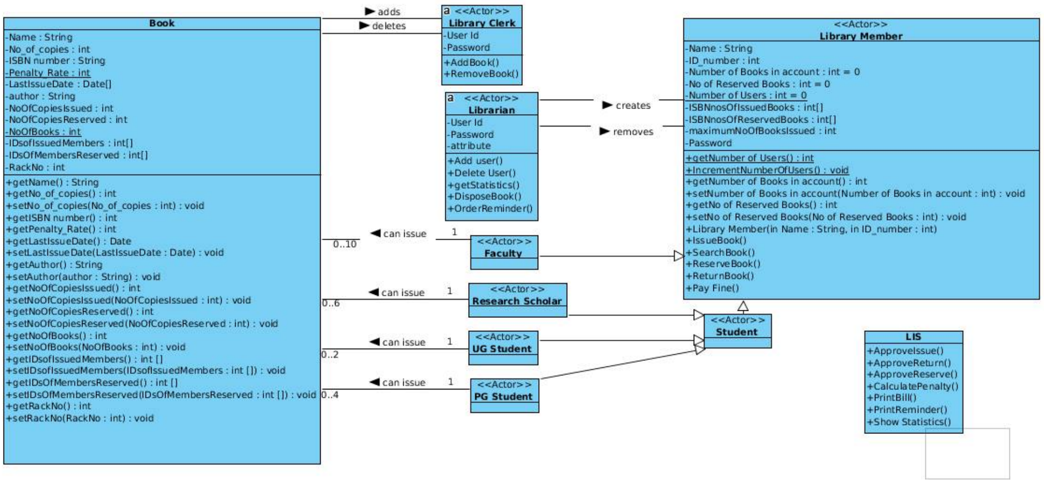
- Trigger:
  return book may trigger this use case

2 Class Diagrams

Initial Draft



Final Draft


Book
-Name : String
-No_of_copies : int
-ISBN number : String
-Penalty_Rate : int
-LastIssueDate : Date[]
-author : String
-NoOfCopiesIssued : int
-NoOfCopiesReserved : int
-NoOfBooks : int
-IDsofIssuedMembers : int[]
-IDsOfMembersReserved : int[]
-RackNo : int
+getName() : String
+getNo_of_copies() : int
+setNo_of_copies(No_of_copies : int) : void
+getISBN number() : int
+getPenalty_Rate() : int
+getLastIssueDate() : Date
+setLastIssueDate(LastIssueDate : Date) : void
+getAuthor() : String
+setAuthor(author : String) : void
+getNoOfCopiesIssued() : int
+setNoOfCopiesIssued(NoOfCopiesIssued : int) : void
+getNoOfCopiesReserved() : int
+setNoOfCopiesReserved(NoOfCopiesReserved : int) : void
+getNoOfBooks() : int
+setNoOfBooks(NoOfBooks : int) : void
+getIDsofIssuedMembers() : int []
+setIDsofIssuedMembers(IDsofIssuedMembers : int []) : void
+getIDsOfMembersReserved() : int []
+setIDsOfMembersReserved(IDsOfMembersReserved : int []) : void
+getRackNo() : int
+setRackNo(RackNo : int) : void
adds
deletes
a <<Actor>>
Library Clerk
-User Id
-Password
+AddBook()
+RemoveBook()
a <<Actor>>
Librarian
-User Id
-Password
-attribute
+Add user()
+Delete User()
+getStatistics()
+DisposeBook()
+OrderReminder()
creates
removes
<<Actor>>
Library Member
-Name : String
-ID_number : int
-Number of Books in account : int = 0
-No of Reserved Books : int = 0
-Number of Users : int = 0
-ISBNnosOfIssuedBooks : int[]
-ISBNnosOfReservedBooks : int[]
-maximumNoOfBooksIssued : int
-Password
+getNumber of Users() : int
+IncrementNumberOfUsers() : void
+getNumber of Books in account() : int
+setNumber of Books in account(Number of Books in account : int) : void
+getNo of Reserved Books() : int
+setNo of Reserved Books(No of Reserved Books : int) : void
+Library Member(in Name : String, in ID_number : int)
+IssueBook()
+SearchBook()
+ReserveBook()
+ReturnBook()
+Pay Fine()
0..10
can issue
1
<<Actor>>
Faculty
0..6
can issue
1
<<Actor>>
Research Scholar
0..2
can issue
1
<<Actor>>
UG Student
0..4
can issue
1
<<Actor>>
PG Student
<<Actor>>
Student
LIS
+ApproveIssue()
+ApproveReturn()
+ApproveReserve()
+CalculatePenalty()
+PrintBill()
+PrintReminder()
+Show Statistics()

Description :

- Book

  - Class Attributes:

    Name:Store the name of book
    No_of_copies:Stores the number of copies of a book
    Penalty_Rate:the rate of penalty for book.A static member as it is common for all the books
    Author:stores name author of book
    LastIssueDateDate:Stores the last issued date of book
    NoOfCopiesIssued:Stores the number of copies of the book that have been issued out
    NoOfCopiesReserved:Stores the number of issued books that have been reserved
    NoOfBooks:A static variable storing the total number of books
    IDsofIssuedMembers[]:An array of integers storing the ids of all the members who have issued copies of the book
    IDsOfMembersReserved[]:An array of integers storing the ids of all the members who have issued copies of the book
    RackNo:stores the rack number where the book is kept

  - Operations:
    Getter functions for name,No_of_copies,ISBN number,Penalty_rate,LastIssueDate,author,NoOfCopiesIssued,NoOfC

    We have used getters for these as we need to view these attributes from outside
    Setters for No_of_copies,LastIssueDate,NoOfCopiesIssued,NoOfCopiesReserved,IDsOfIssuedMembers, IDSOfMembersReserved,RackNo.

    We have used setters for these as these are changeable with time and need to be changesd at a later point of time.These being private members setters ar only way to modify them

- Library Member

  - Attributes:
    - Name:Name of the user
    - ID_number:Login id of the user
    - Number of books in account:Total number of issued books
    - Number of reserved books:Total number of books issued by the user
    - NoOfUsers:A static variable storuing total number of users
    - ISBNnosOfIssuedBooks[]:an array storing the ISBN number of all books issued by the user
    - ISBNnosOfReservedBooks[]:an array storing the ISBN number of all books reserved by the user
    - maximumNoOfBooksIssued:the maximum number of books that the user can issue
    - Pasword:The login password of the user which is necessary for login authentication
  - Operations:
    - Constructor to initialize member

    - Getters for NumberOfBooksinAccount,No ofReservedBooks,NoOfUsers. We have used getters for these as we need to view these attributes from outside

    - Setters for NumberOfUsers,Number of books in account,No Of reserved Books.
      We have used setters for these as these are changeable with time and need to be changesd at a later point of time.These being private members setters ar only way to modify them

    - IssueBook:called when member tries to issue a book
    - SearchBook:Called when member tries to search fo a book
    - Reserve Book:Called when member tries to reserve a book
    - ReturnBook:called when member tries to return a book
    - PayFine:Called when member returns an overdue book

- Library Clerk
  - Attributes:
    - UserId:User Id of the library Clerk
    - Password:Password of the library Clerk
  - Operations:
    - AddBook:Adds a new procured book to database
    - Removebook:Deletes a disposed book from database

- Librarian
  - Attributes:
    - UserId:User Id of the librarian
    - Password:Password of the librarian
  - Operations:
    - AddUser:Adds a new user account
    - DeleteUser:Deletes an existing user account
    - getStatistics: asks for statistics from LIS
    - DisposeBooks:Disposes a book not issued in 5 years
    - OrderReminder:Orders LIS to print reminder on overdue books

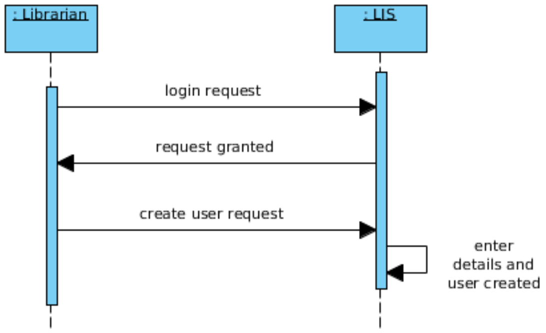
- LIS
  - Operations:
    - ApproveIssue:Called when user tries to issue a book
    - ApproveReturn:Called when user tries to return a book
    - ApproveReserve:Called when user tries to reserve a book
    - CalculatePenalty:Calculates penalty on overdue book
    - PrintBill:prints penalty Bill
    - PrintReminder:prints Reminder on overdue books

- ∗ Show Statistics:Display statistics of books issued

- Faculty
  It is a generalization of Library User
- Student
  It is a generalization of Library User
- Research Scholar
  It is a generalization of Student
- UG Student
  It is a generalization of Student
- PG Student
  It is a generalization of Student

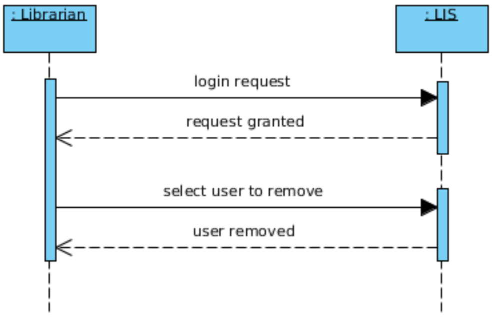
3 Sequence Diagram

Add User Sequence Diagram



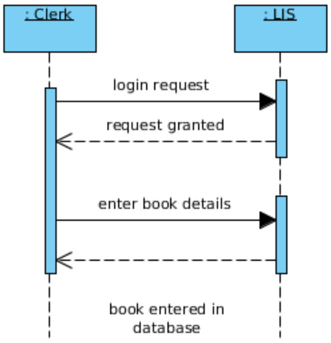
Description : The user needs to login succesfully into the account as the librarian as he/she only has the priviledge to add a user .He then provides the suitable details required to create a user in the library and creates it.

Remove User Sequence Diagram



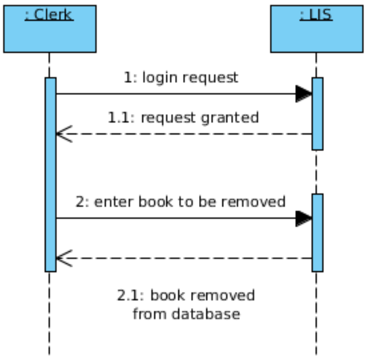
Description : The user needs to login succesfully into the account as the librarian as he/she only has the priviledge to remove a user .He then provides the suitable details required to remove a user in the library and the user along with all the user history is removed from the library database.

Add book Sequence Diagram



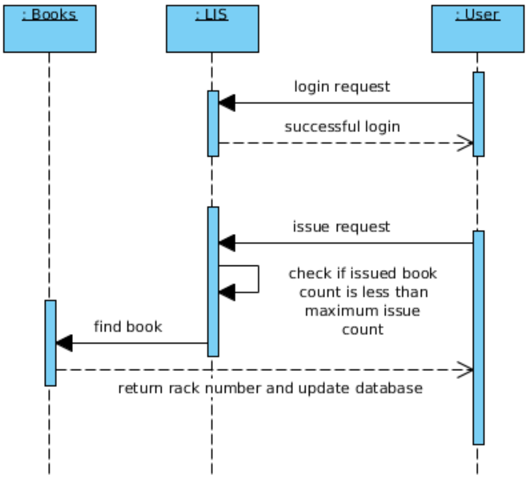
Description : The user needs to login succesfully into the account as the library clerk as he/she only has the priviledge to add a book into the database of the library.The clerk then provides the suitable details of the book into the system and it automatically updates this into the database at the same time. The book will then be available for issuing ad reservation by the members.

Remove book Sequence Diagram



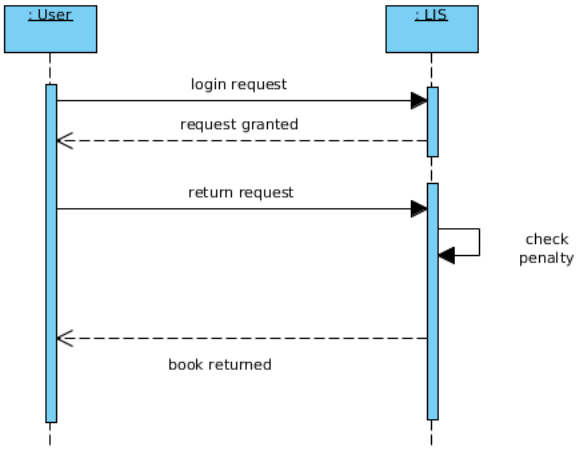
The user needs to succesfully login as the library clerk as he/she only has the priviledge to remove a book from the sytem if he/she feels that the book has been unused for a long period of time and is not required anymore.The user feeds the details of the book to be removed and the system automatically removes all the records related to this book from the system and will not be thereafter available for issuing and reservation by the other library members/users.

Issue book Sequence Diagram


: Books
: LIS
: User
login request
successful login
issue request
check if issued book count is less than maximum issue count
find book
return rack number and update database

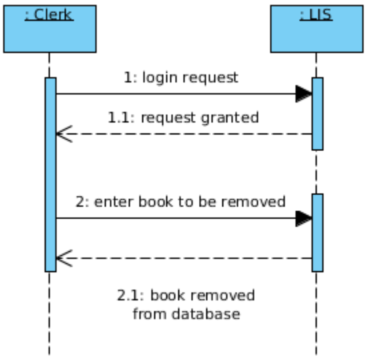
The user needs to succesfully login as a valid member or user of the library to issue a book. Also the book must be available in the library at that instant and also he/she must not exceed the maximum book count against his/her account in the library.The book is then added into the user's account and is thus issued.

Return book Sequence Diagram



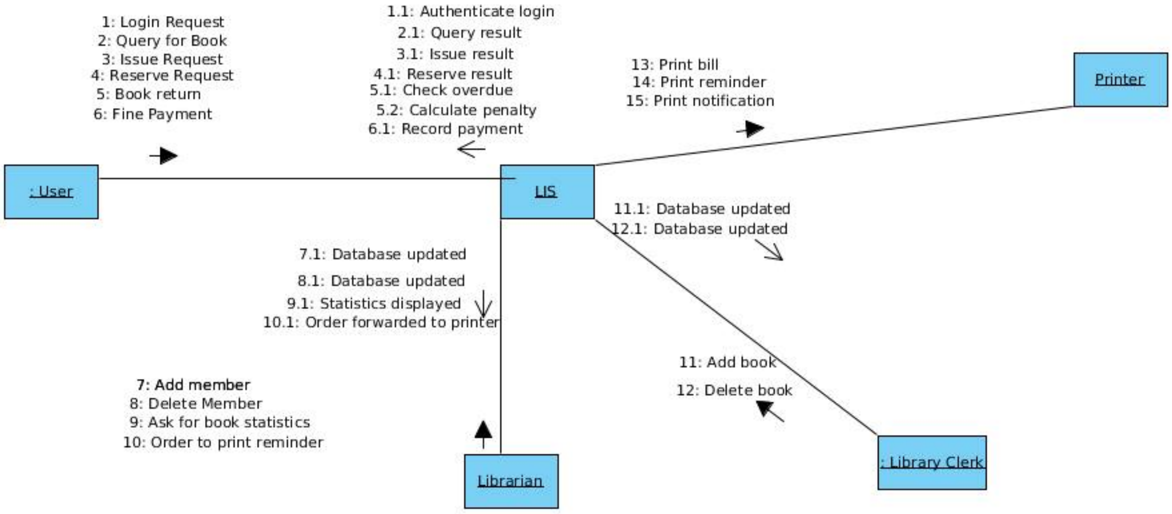
The user needs to successfully login as a valid member of the library to avail the option of returning the book.The book should not be overdue else the user has to pay a fine as per the rate predecided by the library authority. The book in any case is returned and all changes are updated in the user details of the database.

Search book Sequence Diagram



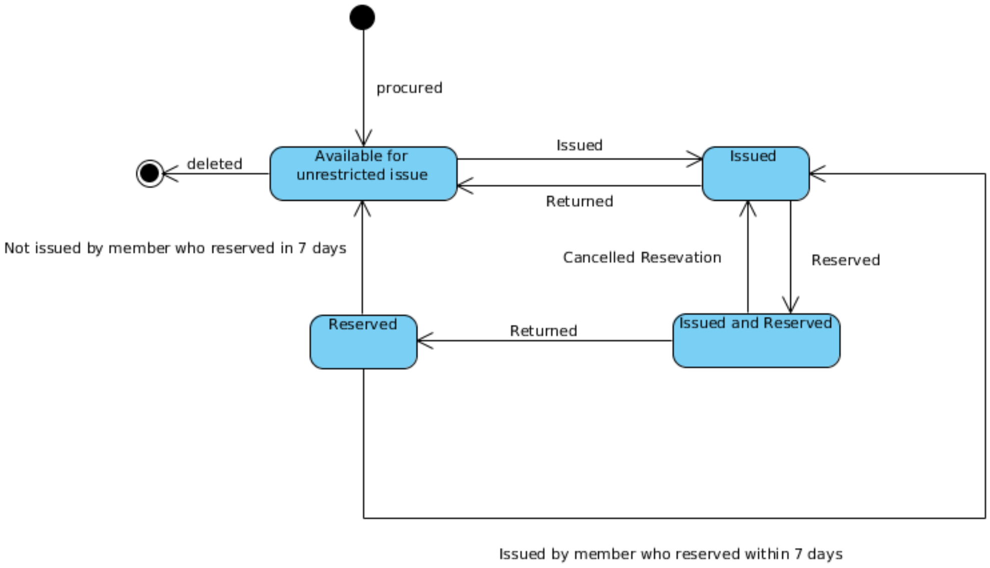
The user needs to successfully login to search if a book is present in the library.The user needs to give the details of the book in the system and it will notify about the prescence of the book in the library.

4 Collaboration Diagram



Description : The user send a specific set of requests to the LIS system like the login request , issue request , return request , serach request and penalty/fine payment.The LIS is bound to send back suitable return messages in case of each of the request messages sent ot the system.
The library clerk has the priviledge of adding and removing books from the system and the LIS system will automatically update its database inside by a cron job.
The librarian has the sole administrator access to the software and the priviledge of adding a member as well as removing his record from the library.The librian can acces the user details or the history of any user.He can also take a alook into the statistics of any book that is can see the book history and can send a notification to remove a book to the library clerk in case the book has been unused and unissued for a long period of time.
The printer is also linked with the sytem and is used to print the notification and the penalty bills and reminder notifications

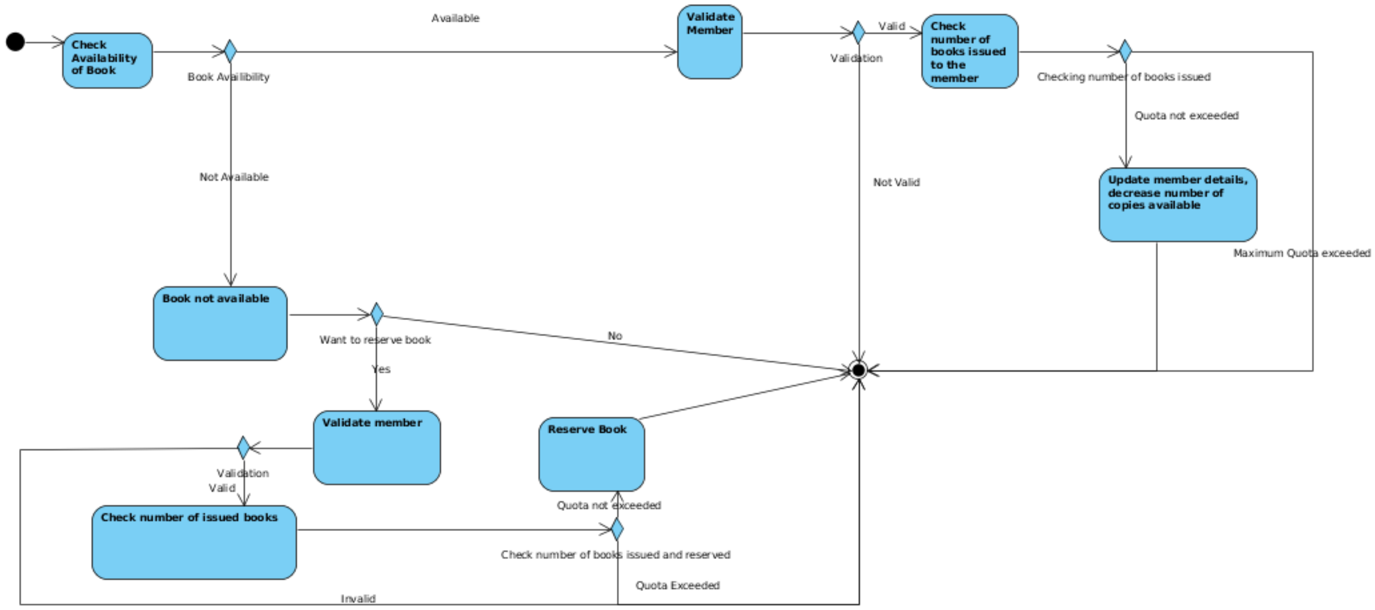
5 Statechart Diagram


procured
Available for unrestricted issue
deleted
Issued
Issued
Returned
Not issued by member who reserved in 7 days
Cancelled Resevation
Reserved
Reserved
Returned
Issued and Reserved
Issued by member who reserved within 7 days

Description : The book is avilable for unrestricted issue if it has not been reserved or enen it has been then the user who has reserved it has not issued it within the 7 days afer the notification that the book is available. The book is issued and returned in a separate state. There is also a facility to cancel a reservation .Even after issuing the book may be reserved by some other user. The transition between the states have been clearly shown and demarcated in the above diagram

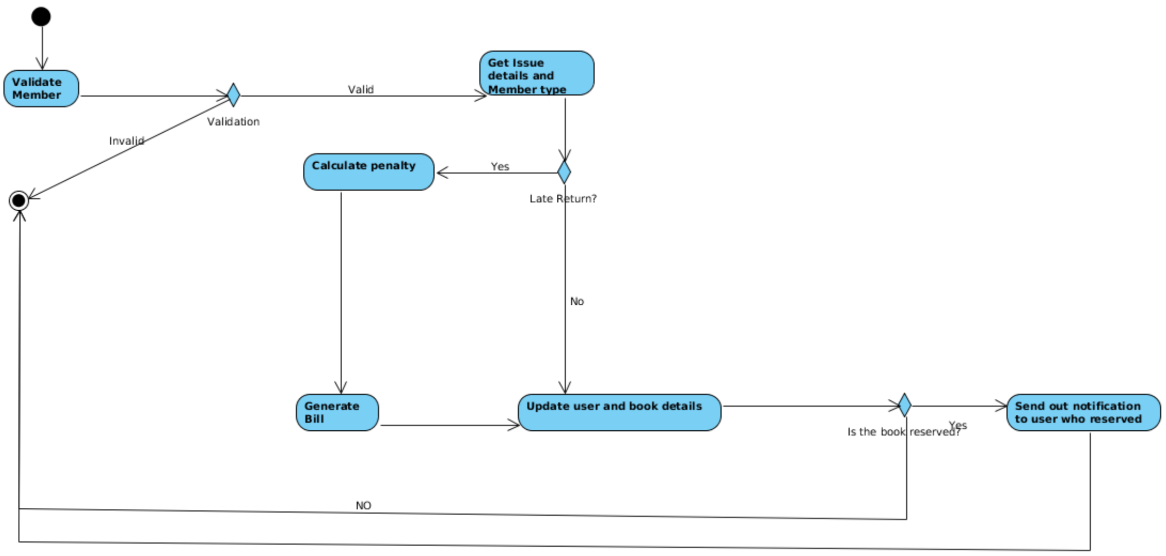
6 Activity Diagram

Issue Book Activity Diagram



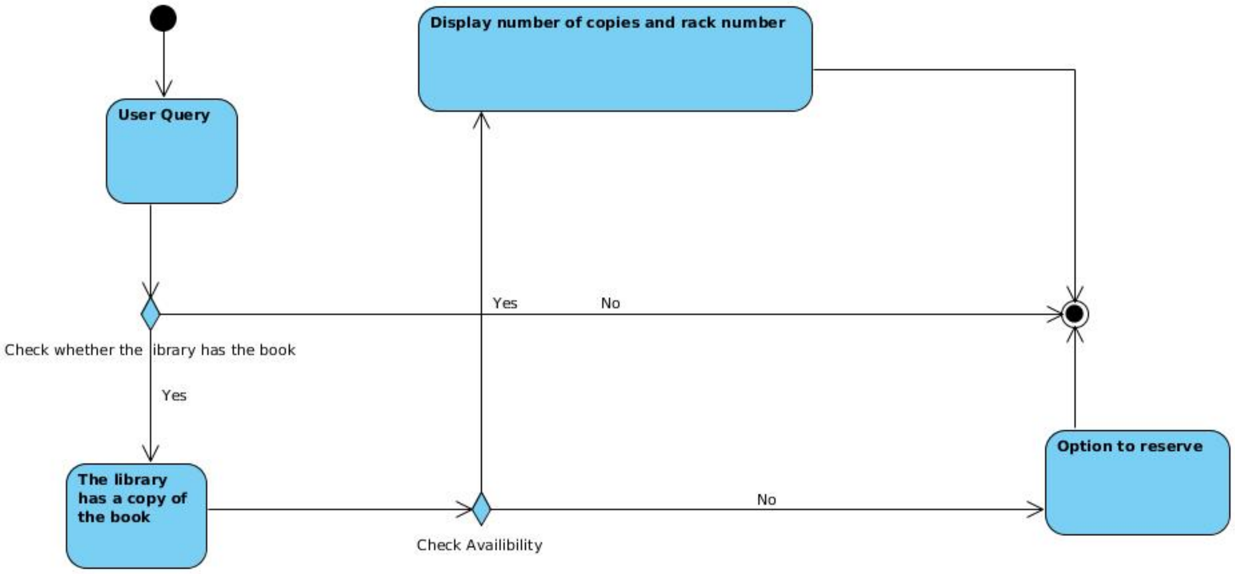
Description : First after succesful login the member has to check the availability of the book.We then check if the user is still allowed to issue books or if he has exceeded the maximum book count against his/her account.In case the book is avilable then the book is iisued and the required changes are made to his/her respective account. In case the book is not available then the user can decide to reserve the book or not.If he reserves the book then again we validate him and take a note of the reservation made.A notification will be made to him when the book will be available in the library

Return Book Activity Diagram



Description : We first check for the validation of the member. If the member succesfully logs into his/her account then we ask for the book to be returned. The time for the issue is then checked and using that the number of days he book has been kept is calculated. If trhe book has been kept for a longer period of time than it was meant to be the member or the user is liable to pay fines as per the rate predecided.The fine is calculated by a base on the time the book has been overdue.In any case the book is returned and required updation is donw on the account of the user.

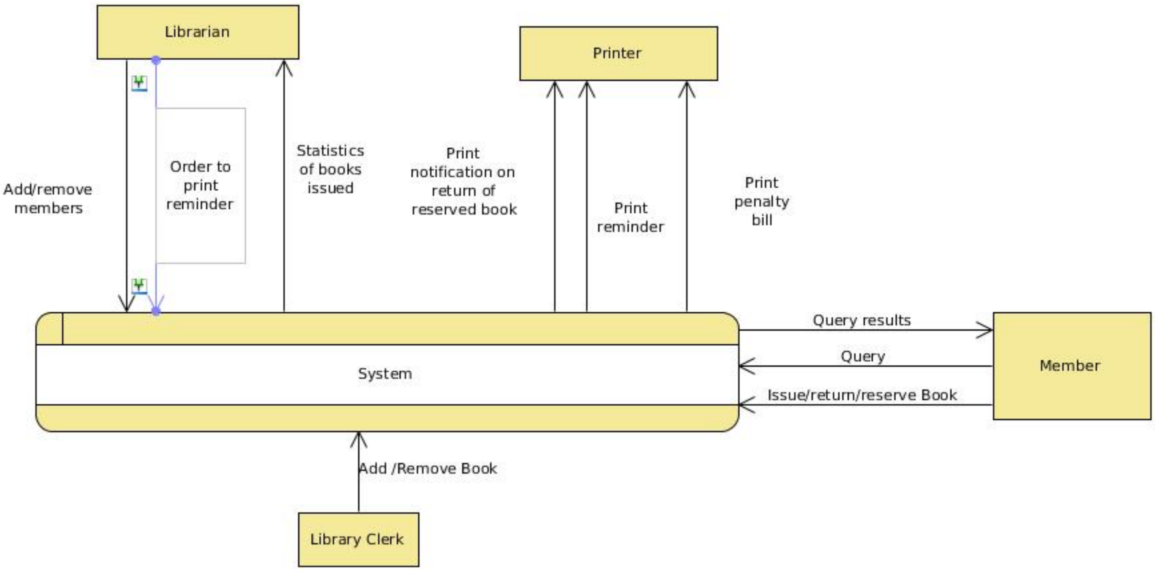
Search Book Activity Diagram


User Query
Display number of copies and rack number
Check whether the library has the book
Yes
No
Yes
The library has a copy of the book
Check Availibility
No
Option to reserve

Description : The user may not need to validate into the system that is the search feature is kept as a global access feature. The person needs to put in the details of the book required in the search field to search for its availability. If the book is present the software will diplay the number of the copies along with their respective rack number for easy access else it will diplay that the book is not currently available.

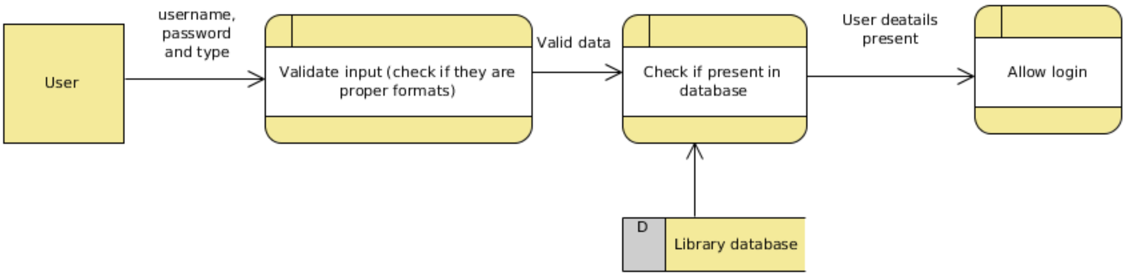
7 Data Flow Diagram

- Context Diagram

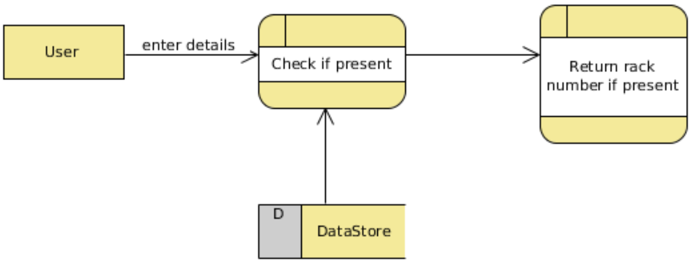


- Level 1 DFD

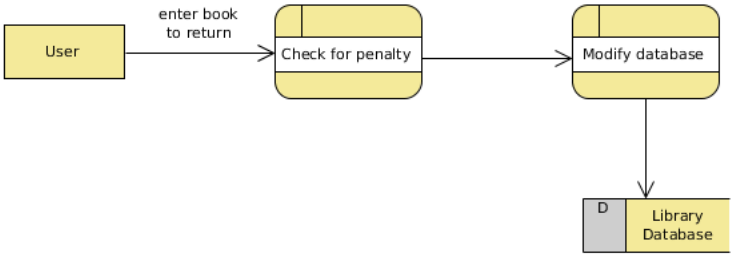
  – Diagram for login :


User
username, password and type
Validate input (check if they are proper formats)
Valid data
Check if present in database
User deatails present
Allow login
D
Library database

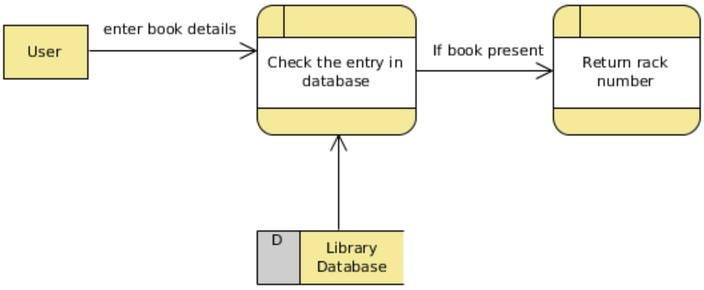
– Diagram for issuing books :



– Diagram for returning books :



– Diagram for searching books :


User
enter book details
Check the entry in database
If book present
Return rack number
D
Library Database