SA SD Document

LIBRERIA
LIBRARY INFORMATION SYSTEM
Group Number : 52
Name of group members:

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

Contents

Part I
System Analysis

1 Feasability Study

1.1 Understanding the problem

In this section, we make an effort to understand the purpose of the software.
The LIS(Library Information Software) is intended to ensure a systematic manner of maintaining the database of a library and easing the job of a librarian.The LIS will help maintain a record of books in the library and members registered with the library.The software aims at hassle-free handling of basic library functions like issue/reserve/return books as well as giving the freedom to the librarian to add and delete books and members to the database. Extending further the software can be used by the librarian to decide on disposing of books which have not been issued for a long time. This software also helps to keep track of statistics of issue of a particular book ver a period of time.Since everything is done on the computer, it is easy to record all data and there are minimal chances of inconsistencies or ambiguities. Also use of a secure database support enhances the security aspects of the software thus enabling to create a more robust and secure software

1.2 Scope of the Problem

In this section,we list the various functions performed by the software:

- Add books to the database
- Modify existing records of books
- Delete unused books from database
- Add users to the database
- Modify existing details of users
- Remove user from the database
- Query regarding books on ISBN number or name
- Query regarding user on user id or name
- Record issue,return and reservation of books
- Calculation of penalty on overdue books
- Maintainence of statistics of all books

1.3 Analysing stakeholders

The various stakeholders are:

- Librarian
- Library Clerk

- Member / Library user
  - Faculty
  - Research Scholar
  - PG Student
  - UG Student

These stakeholders have been clearly described with their attributes and methods in the SRS.

1.4 Defining Alternatives

1.4.1 Connection between software and librarian

The librarian is treated as the administrator of the software who has the sole capability to create and delete the members like the faculty or the students (including research scholars,UG and PG students). The librarian can send a print notification if required if a penalty is required to be paid in case of overdue of a book
Also the librarian can study the book statistics and can send a notification to the clerk to dispose off old and unused books from the library database

1.4.2 Connection between software and member

The members can perform the following operartions:

- Can issue a book if available in the library
- Can request for reserving a book if already issued to another user (subjected to the condition that he does not exceed his/her maximum book limit)
- Can return a book to the library
- pay fine/penalty if required in case of a overdue of book

1.4.3 Hardware Infrastructure

1. Processor Pentium II processor or higher
2. Hard Disk space 500MB
3. RAM 512 MB
4. Network/Internet access

1.4.4 Software Infrastructure

1. Operating system
   - Windows 7 or later
   - Linux distributions like Ubuntu 14.04.03 or other
2. MySQL
3. Java JDK platform 1.7 or higher

1.4.5 Technology Used

The technilogies used are as follows :

- Use of the platform independent JVM using JDK 1.8
- MySQL support to provide DBMS features to the software

1.4.6 Security

The security of the software has been taken care of in a very cautious and judicious manner. The protocol followed to ensure data security and software safety are as follows :

- User can acces the books only after a successful login with a valid username and password.
- Check has been done during creation of new user to ensure a unique user id.
- Also check has been done to set a miminum length of 8 characters and use of alphanumeric characters to make a password without storing any value (to prevent publicity of password) so that users can give a safe and efficient password to safeguard their account.
- Unsuccessful attempts may occur if the user gives wrong username or wrong pasword or even wron user type.

1.5 Defining Criteria to evaluate

The software is evaluated on the basis of the follwoing criterias:

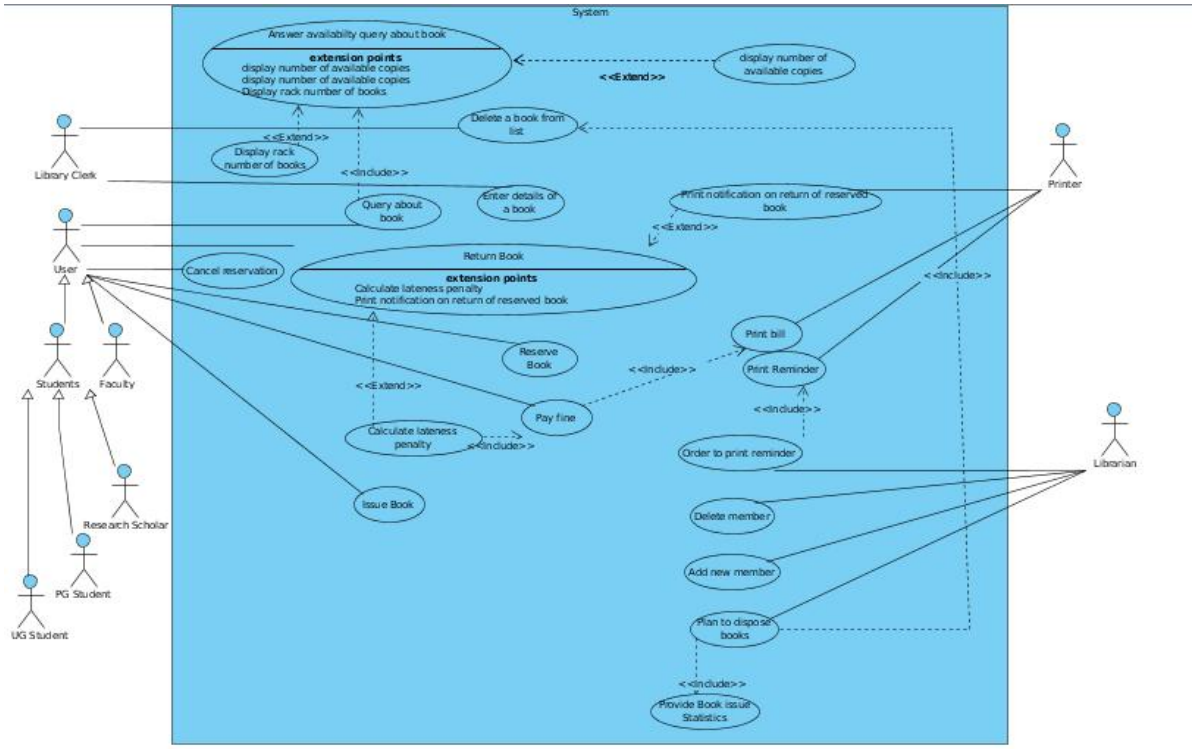
- Only the librarian can be created without any preprocessing. But only one librarian must be present.
- Addition and deletion of books can be performed by library clerk in this software
- Creation of users by the librarian can be performed effectively
- Users should be able to successfully login into the software
- Creation of effective users who can perform all the operations like issue, reserve and returning of a book

1.6 Report

The software designed is able to perform all the above operations mentioned above. One can create a single librarian at first whon hold the admin acces of the software.The librarian is able to create the users or the library members who can perform all the required operations as mentioned in the SRS.
The library clerk is able to add books into the library and can remove the unrequired books from the database.
The login feature has also been created,tested and is working perfectly.Only the authenticate users can log into the software to perform the designated operations.
A background cron job takes care of all the background processes and helps to maintain all the records as required like in the case of issuing and returning.It also takes care of the issued books and takes care if a penalty is to be issued against any user account.
All the process as mentioned in the SRS document provided have been implemented and tested successfully.

2 UML Diagrams

2.1 Refinement of Use Case Diagram



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 communiates 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 communiates 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 communiates 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 communiates 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 communiates 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 communiates 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 communiates 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 communiates 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 communiates 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 communiates 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 communiates 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 communiates with the system

  - Trigger:
    order to print reminder triggers this

- Print notification on return of reserved book

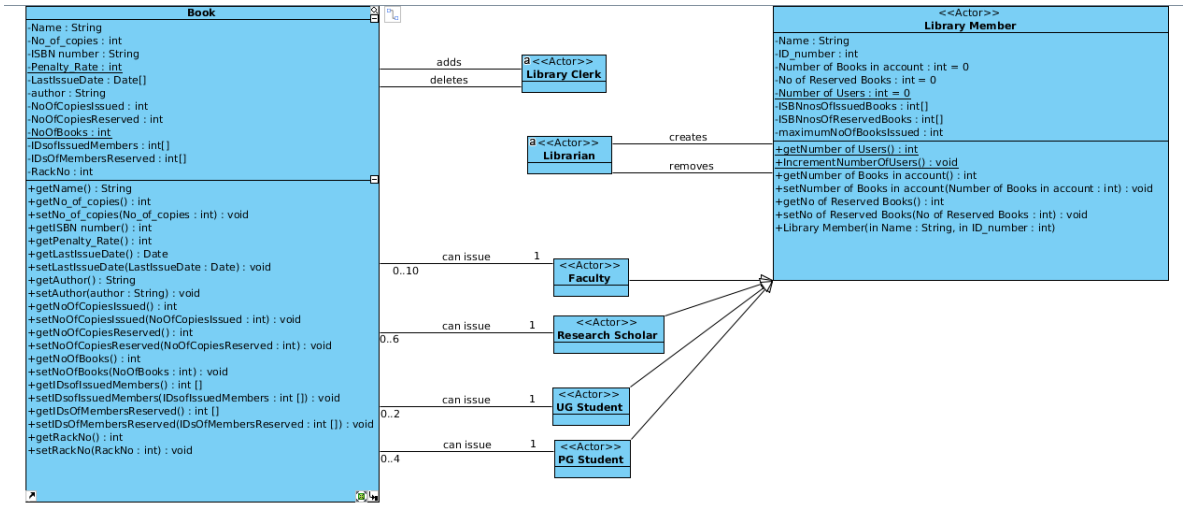
– Preconditions:
Some user must have returned a reserved book

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

– Actors:
Printer communiates with the system which communicates with user

– Trigger:
return book may trigger this use case

2.2 Refinement of Class Diagrams



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

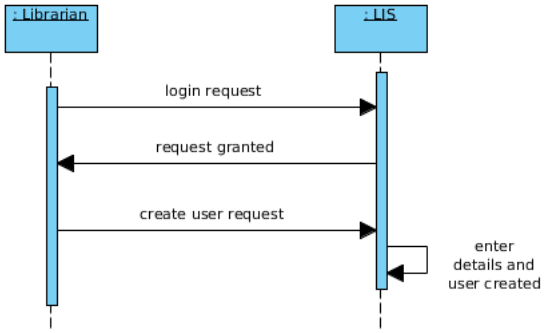
- Library Clerk

  - 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

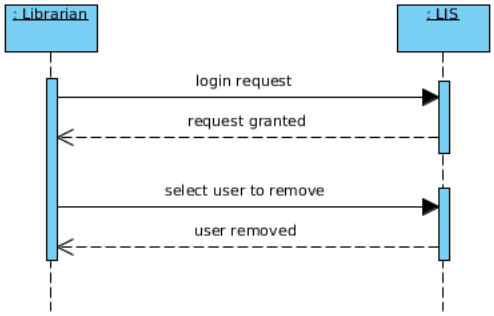
2.3 Sequence Diagram

Add User Sequence Diagram



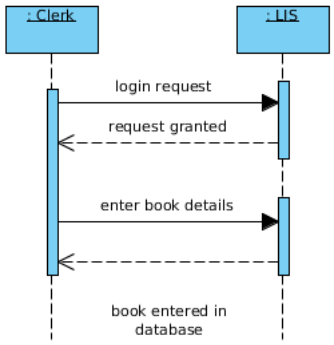
Description : The user needs to login succesfully into the account as the librarian as he/she only has the priviledge to add a user .He then provides the suitable details required to create a user in the library and creates it.

Remove User Sequence Diagram



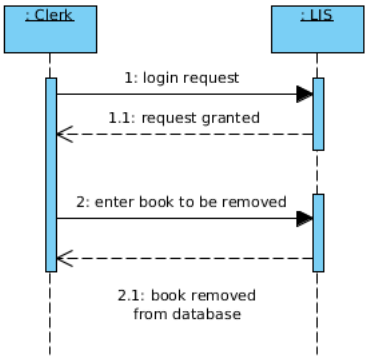
Description : The user needs to login succesfully into the account as the librarian as he/she only has the priviledge to remove a user .He then provides the suitable details required to remove a user in the library and the user along with all the user history is removed from the library database.

Add book Sequence Diagram



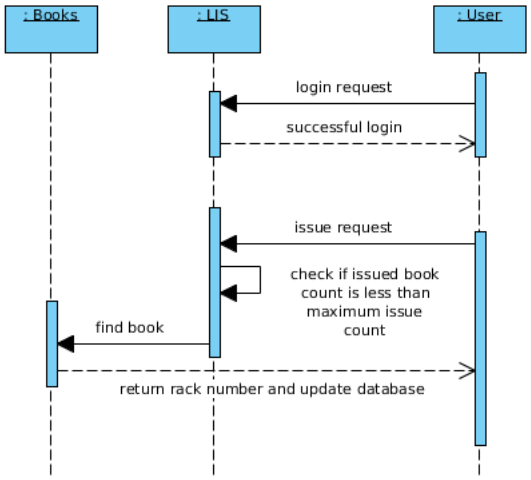
Description : The user needs to login succesfully into the account as the library clerk as he/she only has the priviledge to add a book into the database of the library.The clerk then provides the suitable details of the book into the system and it automatically updates this into the database at the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

Remove book Sequence Diagram



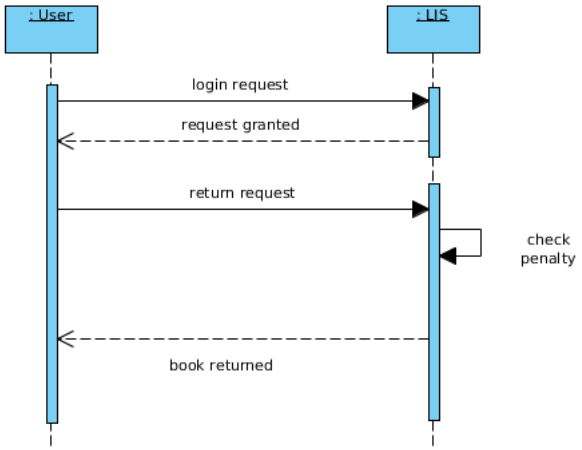
The user needs to succesfully login as the library clerk as he/she only has the priviledge to remove a book from the sytem if he/she feels that the book has been unused for a long period of time and is not required anymore.The user feeds the details of the book to be removed and the system automatically 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



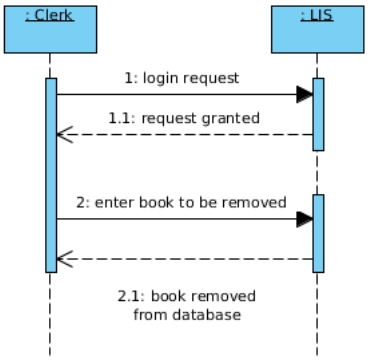
The user needs to succesfully login as a valid member or user of the library to issue a book. Also the book must be available in the library at that instant and also he/she must not exceed the maximum book count against his/her account in the library.The book is then added into the user's account and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

Return book Sequence Diagram



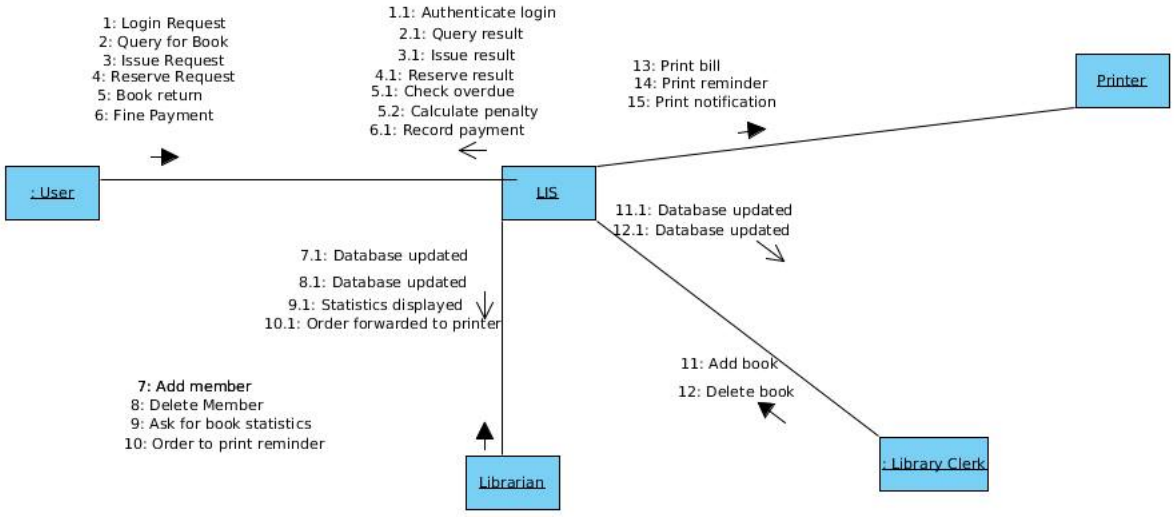
The user needs to successfully login as a valid member of the library to avail the option of returning the book.The book should not be overdue else the user has to pay a fine as per the rate predecided by the library authority. The book in any case is returned and all changes are updated in the user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

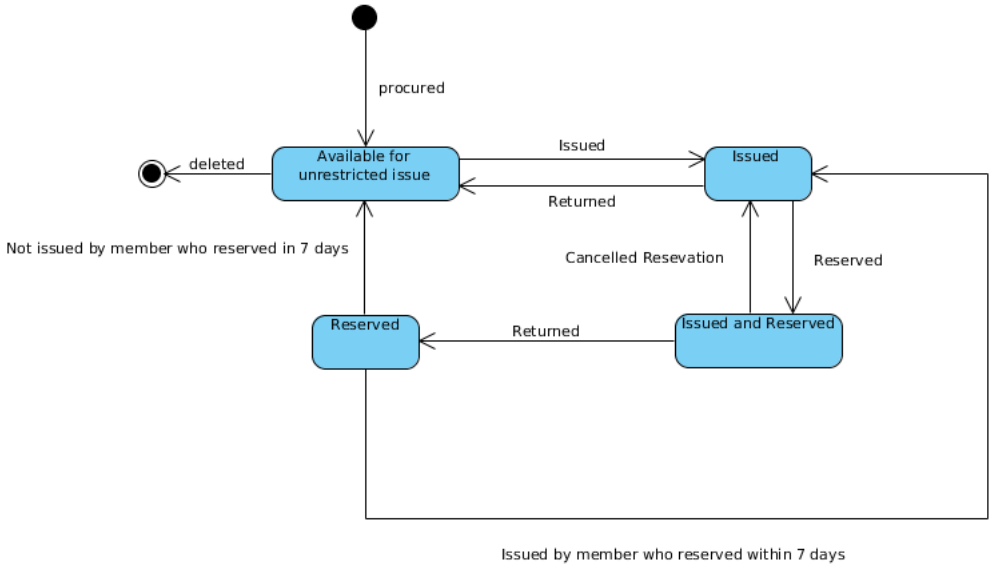
2.4 Collaboration Diagram



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

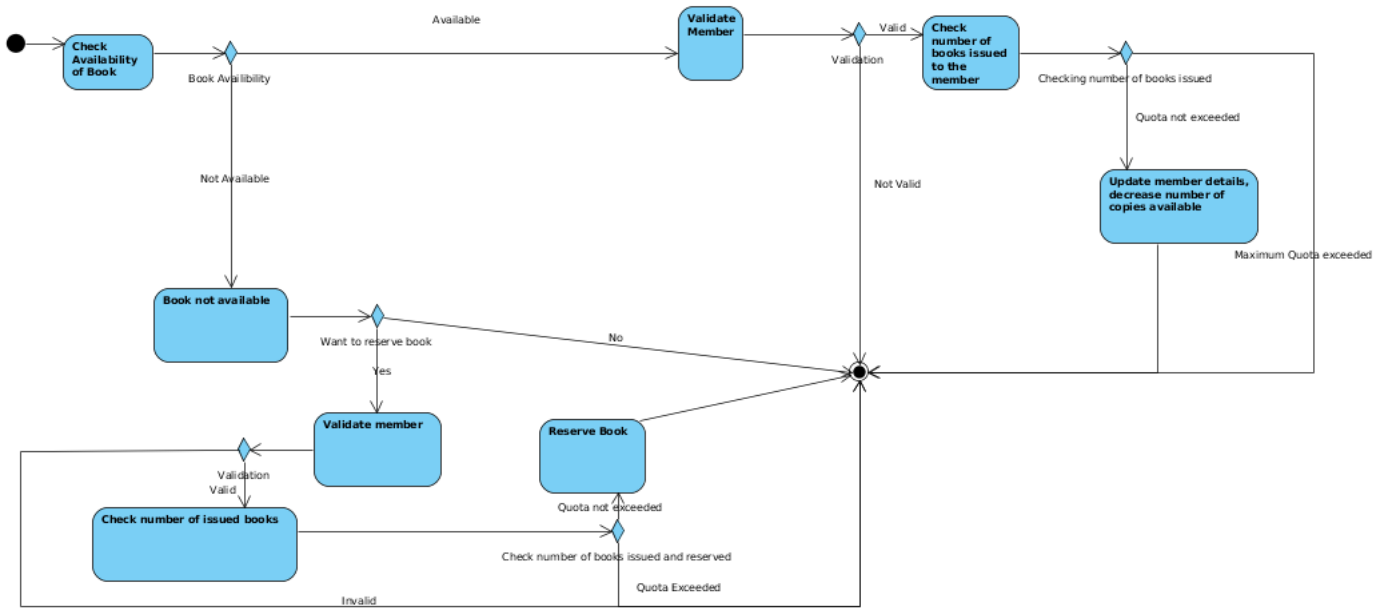
2.5 Statechart Diagram



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

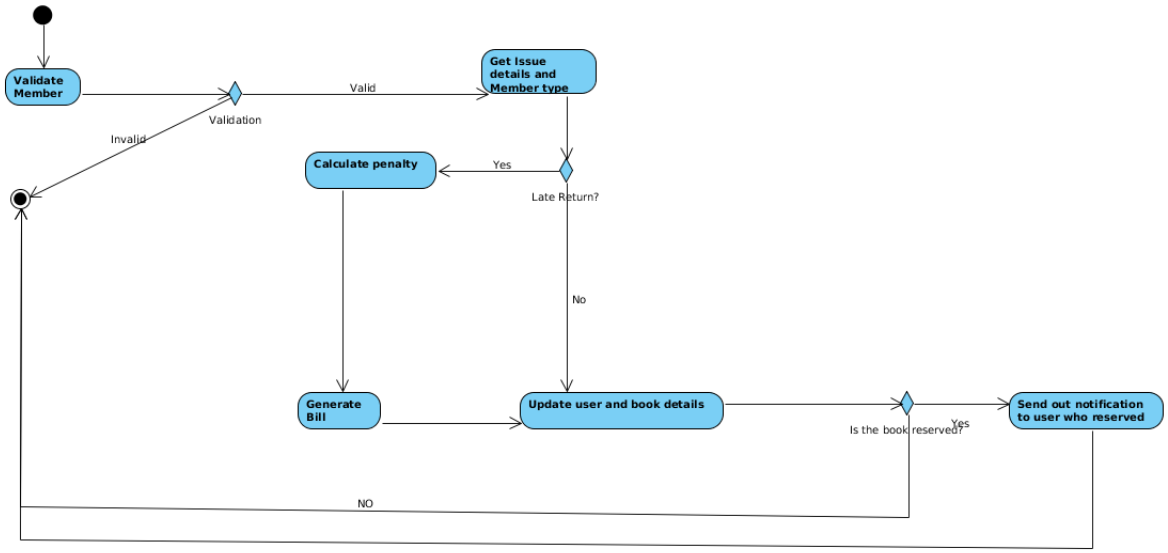
2.6 Activity Diagram

Issue Book Activity Diagram



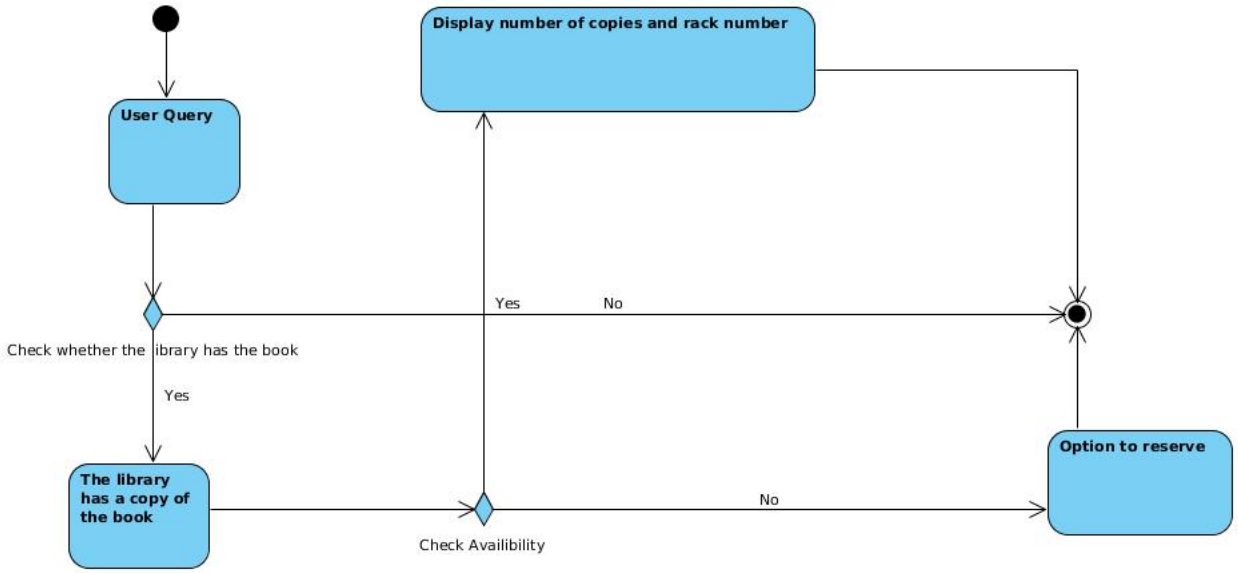
Description : First after succesful login the member has to check the availability of the book.We then check if the user is still allowed to issue books or if he has exceeded the maximum book count against his/her account.In case the book is avilable then the book is iisued and the required changes are 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.

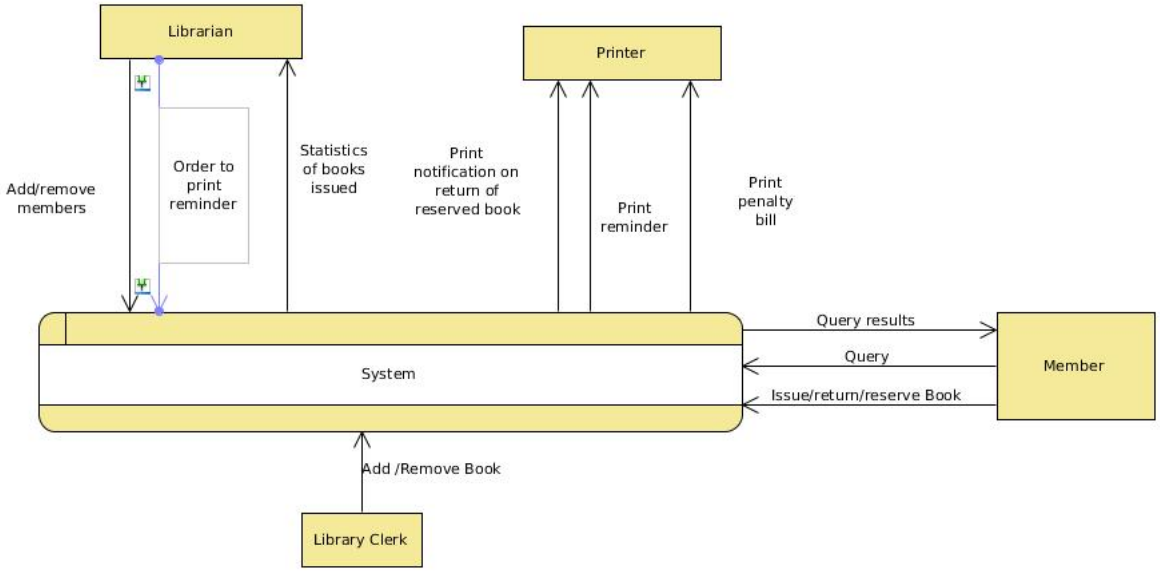
Search Book Activity Diagram



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.

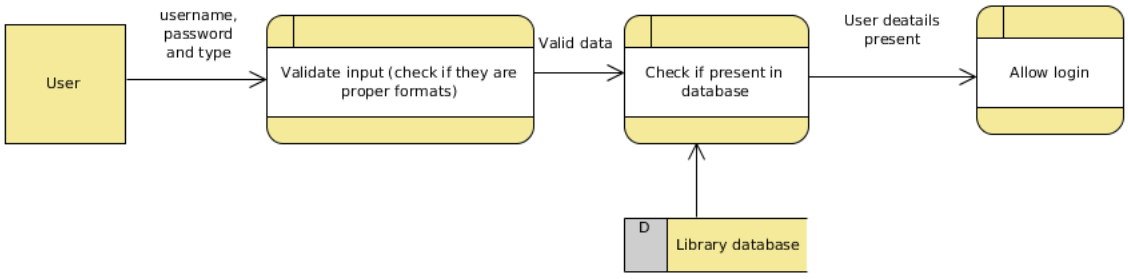
2.7 Data Flow Diagram

- Context Diagram

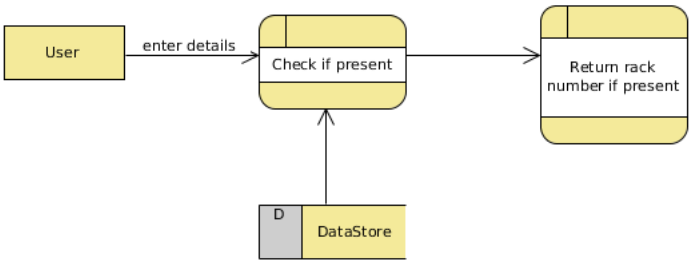


- Level 1 DFD

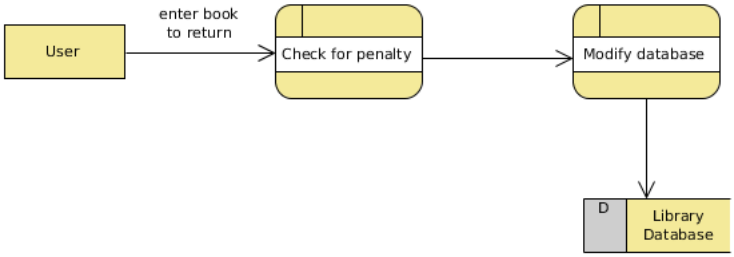
  – Diagram for login :



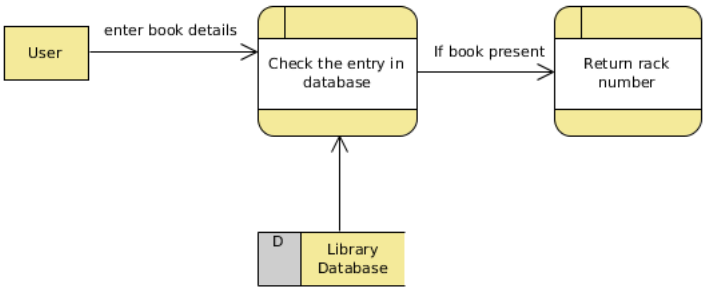
  – Diagram for issuing books :



– Diagram for returning books :



– Diagram for searching books :



3 System Parameters

3.1 Platform

The software requires the following platforms to run : Minimum system requirements :

- Hardware Requirements :
  1. Processor Pentium II processor or higher
  2. Hard Disk space 500MB
  3. RAM 512 MB
- Software Requirements :
  1. Operating system
     - Windows 7 or later
     - Linux distributions like Ubuntu 14.04.03 or other
  2. MySQL
  3. Java JDK platform 1.7 or higher

3.2 Language

The entire software is written using the language Java. The object oriented programming paradigm of Java has been followed throughout.

3.3 Build System

The code is compiled and built using Java Standard library.Java converts the code into bytecode which is platform independent and can be run onnany system with JRE installed.

3.4 Libraries

The libraries used for the software are as follows :

1. Java Swing (java.awt.*)
2. Java AWT (javax.swing.*)
3. Java Exception (java.lang.Exception.*,java.io.*)
4. Java mySQL libraries:(java.sql.*)

3.5 Sizing

This is an extremely lightweight software consuming negligibly small amount of main memory during runtime.
At a later point of time it can be modified to run as a web applet or servelet.

3.6 Performance

Performance of the Library Information of System is enhanced by the following listed technique:

1. There is practically no lag in the main GUI.This has been ensured by ensuring that all cron jobs are run in the background, not hampering the running time of the main GUI.

2. The implementation of database through mySQL considerably speeding up the insertion,deletion and answering queries. mySQL through the use of balanced search trees does all of these in worst case O(lg n) time.This reduces the processing time considerably and speeds up the execution of the software.

4 Limitations and Exceptions

The number of books which can be incorporated into the system has been kept to be constant in this case to ensure a simpler design. In case of a dynamically ncreasing book set the algorithms and the data structures used to mange have to be optimized further to keep the running time of the software as fast as possible.
Exceptions which are present or handled in this software are as follows :

1. Username cannot contain any special characters except underscore.Entering any username not following this rule will throw an exception and prompt the user to re-enter the username.

2. The password of the user must be atleast 8 charactes long,must containt atleast one alphabet,atleast one digit and atleat one special character. Violation of these rules will throw an exeption and prompt the user to re-enter password.This is done to enhance the security level by making the password extremely difficult to crack by brute force algorithms.

3. The user type by default is empty in the initial loginscreen.The user must specify the user type while logging in.Keeping it default shall throw an exception and prompt the user to try and login again.

4. The user who has fulfilled his issue/reserve quota may click on issue book.This shall throw an exception and a dialogue box shall appear informing the user to return issued books or cancel reserved books(if any) and then try to issue books.

5. The user who has fulfilled his issue/reserve quota may click on reserve book.This shall throw an exception and a dialogue box shall appear informing the user to return issued books or cancel reserved books(if any) and then try to reserve books.

5 Other Information about analysis

The other analysis performed are mainly on focusing how to achieve more abstraction and also at the same time maintain the safety and the security of the software

Part II
System Design

6 Refined System Parameters

6.1 Global System Architecture

The overall system architecture is a 2-tier architecture which includes client at one end and the database at the other. There is no server based middle tier in the software being designed.

6.2 Platform

6.2.1 Hardware

The hardware platform required in this software is :

1. Processor Pentium II processor or higher
2. Hard Disk space 500MB
3. RAM 512 MB

6.2.2 Software

The software platform of the software is fully developed in Java.
For suitable execution Java JDK 1.7 or later is required.
The database management has been done using a DBMS software like MySQL.

6.2.3 Networking

The GUI support and the database support are interlinked via a tcp network which facilitates the flow of information between the two ends.It provides scope for dividing the space requirements by distributing the load. The use of tcp network ensures secure data flow between the two interfaces.
The tcp network identifies the two ends of the network and extablishes a secured connection between the two ends.Keeping the data at a separate location can improve the safety of the software as it prevents data loss to a huge extents in case of a glitch in the GUI end.
Thus we can ensure a safe, secure, reliable and fast software design.

6.3 Software Architecture

Object-oriented architecture forms the basis of the LIS. In this style data representations and their associated primitive operations are encapsulated in an abstract data type or object. The components of this style are the objects—or instances of the abstract data types. Objects interact through function and procedure invocations. Two important aspects of this style are A. that an object is responsible for preserving the integrity of its representation (usually by maintaining some invariant over it), and B. that the representation is hidden from other objects. Thus the aspects of OOA mentioned justify our choice.

7 Database Design

There is need to maintain a central database to keep all the details of the users and the books in the library to maintain the data more easily.
Even though this thing could have been done by using a simole list or even a balanced tree (to increase efficiency) but the use of DBMS is preferred to perform the operation more easily without worrying how the operation is being impolemented.
Also the DBMS managaement softwares or the technologies provide very less time complexity to perform all the required operations like adding a record, deleting a record or even searching and modifying a data record which is perfectly applicable for our software.
The database used will have the following table :

1. Table of users :
   The list or table of all the users/members in the library along with their usernames and passwords which is used for validation of the user.

2. Table of clerks :
   The list or table of all the clerks in the library along with their usernames and password and some other relevant information which may be required in the system.

3. Table of books :
   The table of all the books along with the specification of the copies of the same book present in the library.
   The fields which needed to be maintained in this table are :

   - the ISBN id or number of the book
   - the name of the book
   - the author of the book
   - number of copies of the same book present in the library
   - the list stating whether any copy of the book is issued
   - the list specifying which user issued which copy

   Also a backup copy of this entire databse needs to maintained to keep the data safety to the maximum level possible.This is to ensure that the data is not lost even if a glitch occurs in the software at any point of time.
   This make the software more user friendly and realistic. The updation needs to be carried out in both the databases at the same time so that no discrepancy occurs in the tables in both the databases.

8 Design Details

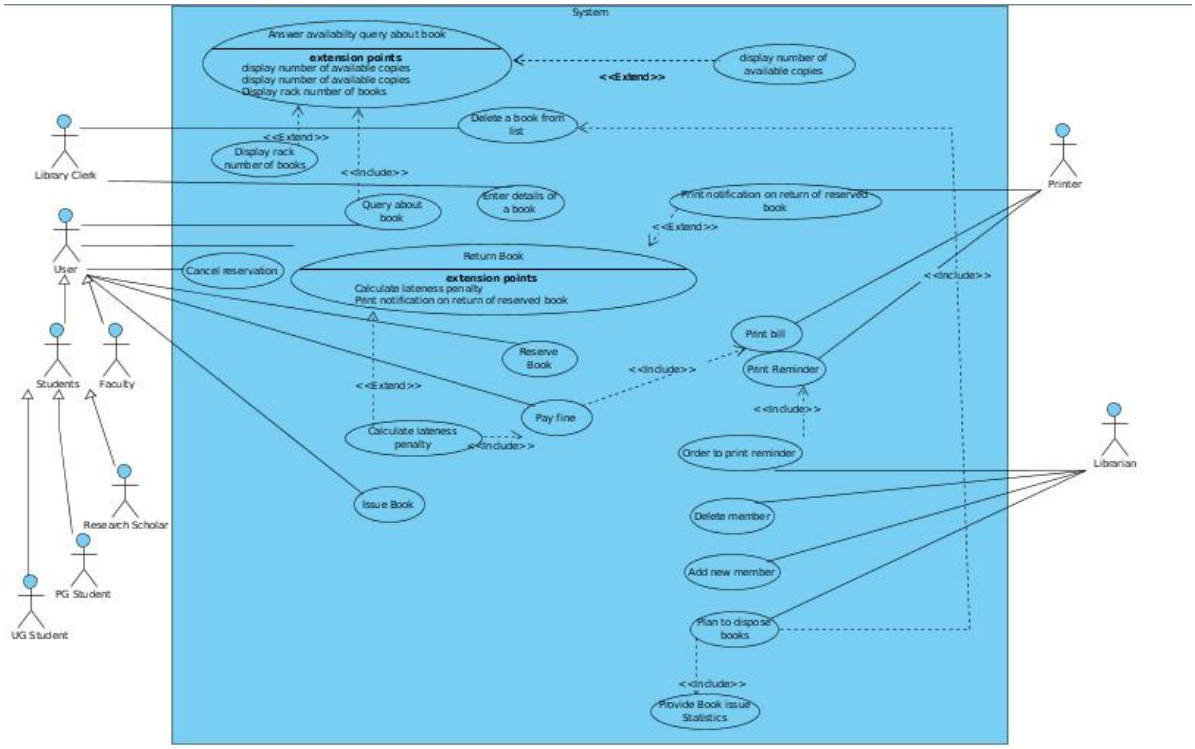
The GUI is an interactive environment designed solely in JAVA.The details of the software involves the following steps :

- User login :
  The person log into the library system as a regular user or a clerk or as a librarian.Upon successful logging in the user is then directed to his/her respected home page to perform actions of his/her choice.

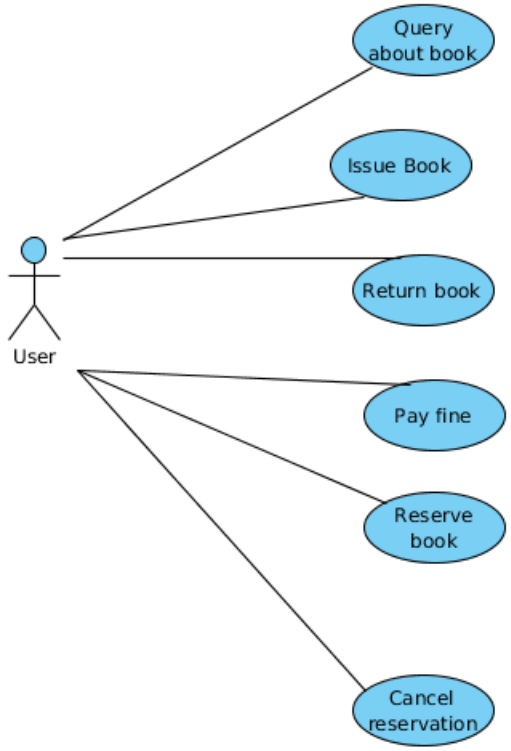
- Add a user :
  The librarian has the opportunity to add a user into the system by entering relevant details.
- Remove a user : The librarian also has the opportunity to remove a user from the library database if he/she finds it necessary. Upon doing this all then details of the user is removed from the system.
- Issue book:
  The user can search and issue a book if it is available in the library at that point of time.
- Reserve book :
  The user can also reserve a book if he/she needs it and the book is not available at that point of time at the library.
  The library system will issue a notification whenever the book is available.
- Return book :
  The user upon successful login will be able to return a book issued earlier.It also checks for the penalty in this stage.

8.1 Refinement of UML diagrams

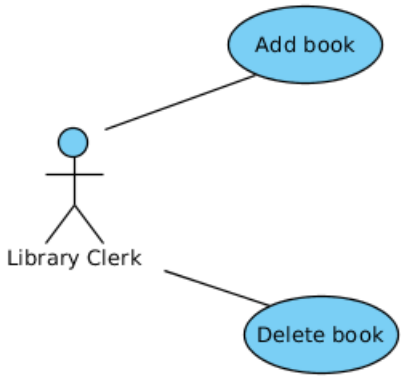
8.2 Refinement of Use Case Diagram



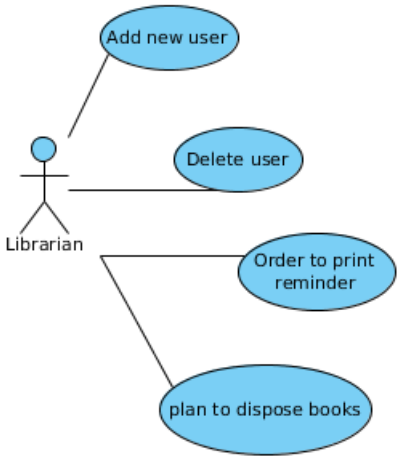
Use case of User :



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 communiates 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 communiates 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 communiates 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 communiates 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 communiates 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 communiates 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 communiates 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 communiates 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 communiates 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 communiates 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 communiates 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 communiates 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

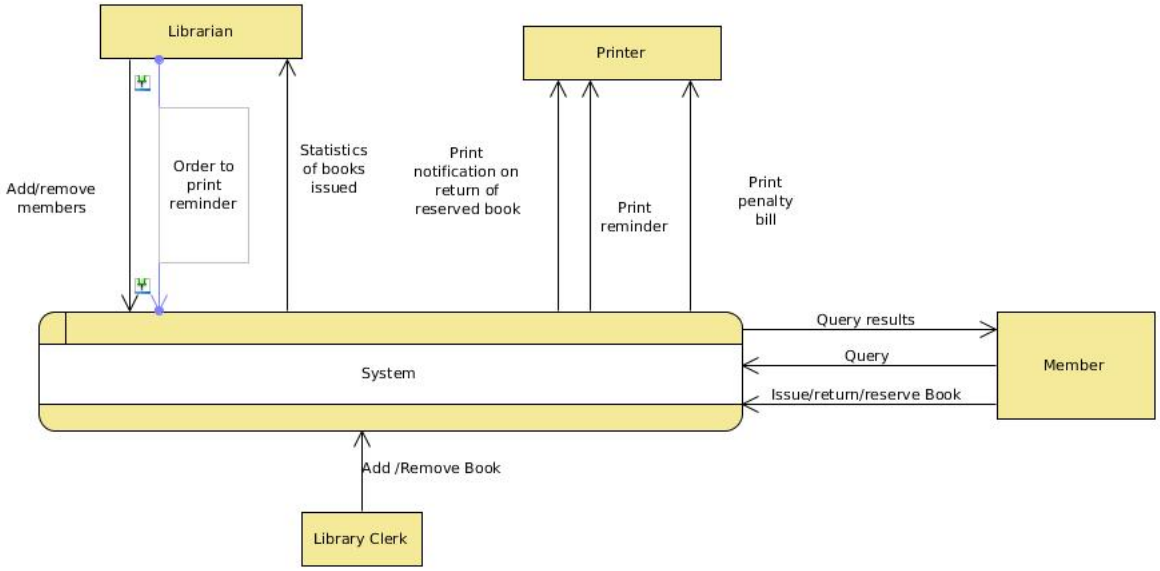
     - Postcondition:
       A notification is printed to the user who reserved the book

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

– Trigger:
return book may trigger this use case

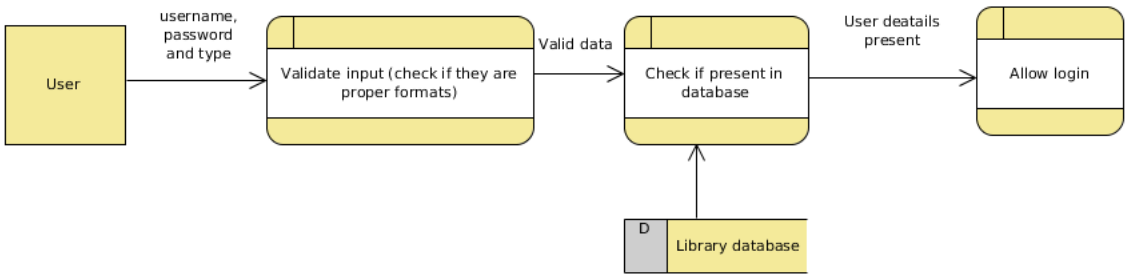
8.3 Data Flow Diagram

- Context Diagram

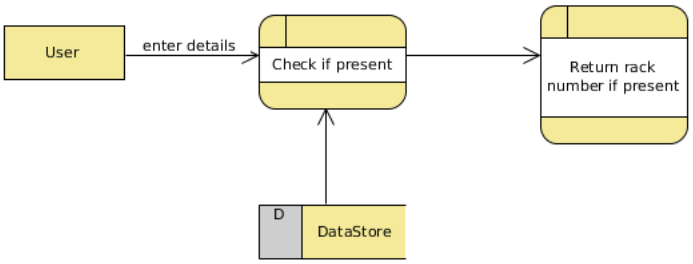


- Level 1 DFD

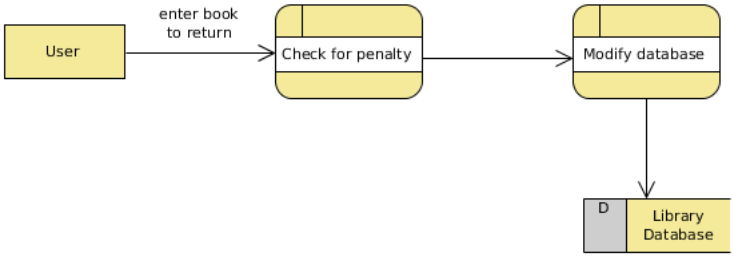
  – Diagram for login :



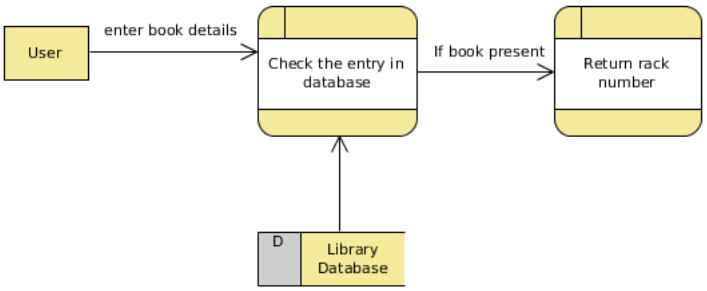
  – Diagram for issuing books :



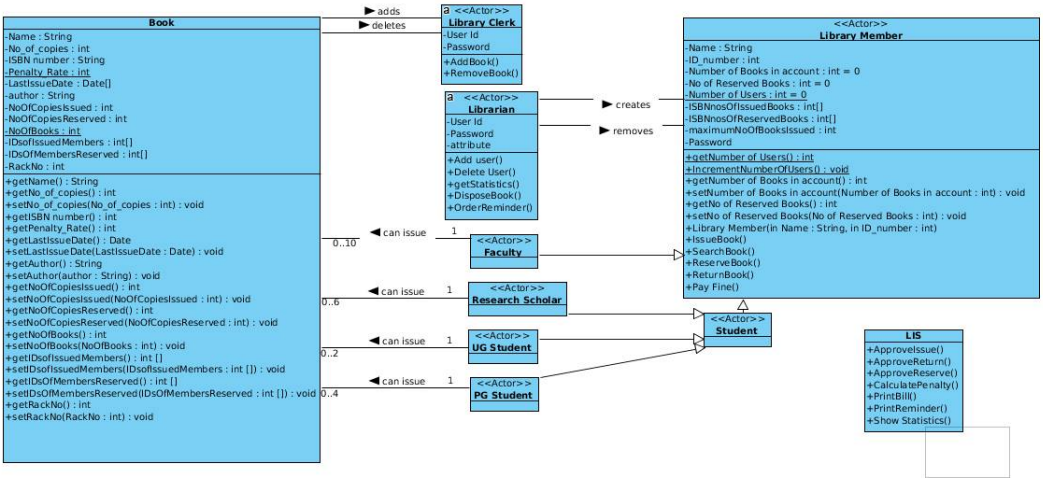
– Diagram for returning books :



– Diagram for searching books :



8.4 Refinement of Class Diagrams



Description of classes:

- 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:

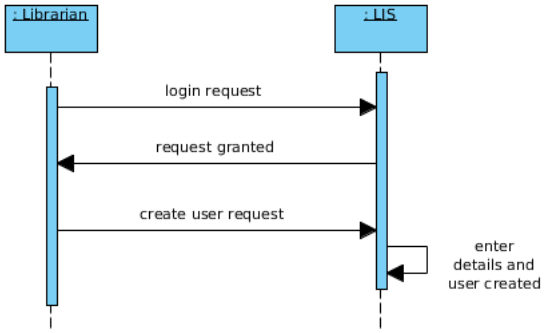
        * UserId:User Id of the librarian
        * Password:Password of the librarian
    - Operations:
        * AddUser:Adds a new user account
        * DeleteUser:Deletes an existing user account
        * getStatistics: asks for statistics from LIS
        * DisposeBooks:Disposes a book not issued in 5 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

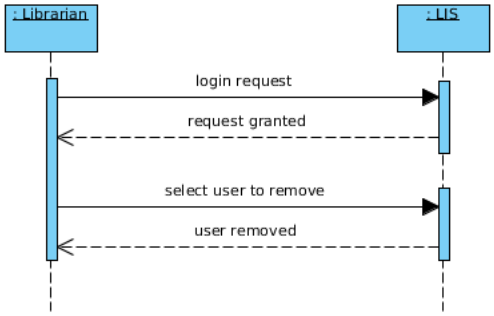
8.5 Sequence Diagram

Add User Sequence Diagram



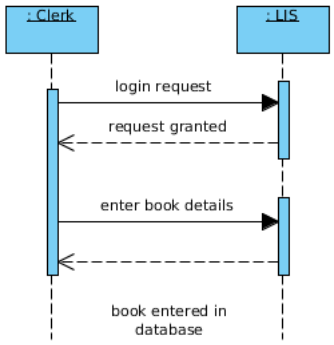
Description : The user needs to login succesfully into the account as the librarian as he/she only has the priviledge to add a user .He then provides the suitable details required to create a user in the library and creates it.

Remove User Sequence Diagram



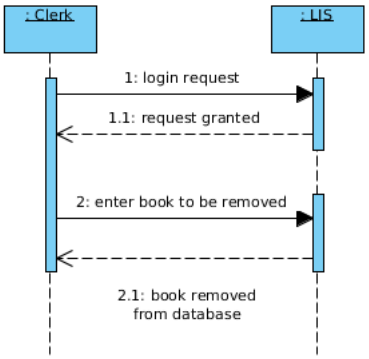
Description : The user needs to login succesfully into the account as the librarian as he/she only has the priviledge to remove a user .He then provides the suitable details required to remove a user in the library and the user along with all the user history is removed from the library database.

Add book Sequence Diagram



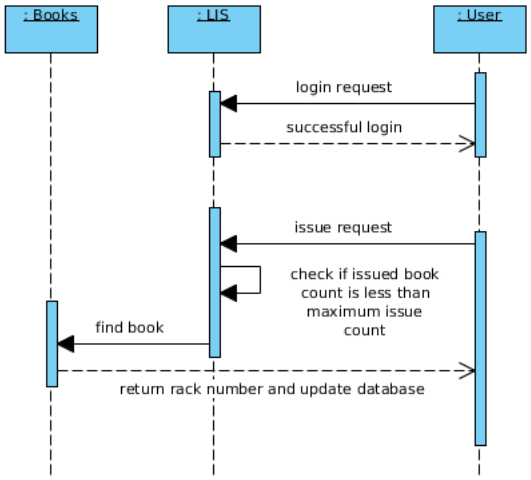
Description : The user needs to login succesfully into the account as the library clerk as he/she only has the priviledge to add a book into the database of the library.The clerk then provides the suitable details of the book into the system and it automatically updates this into the database at the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

Remove book Sequence Diagram



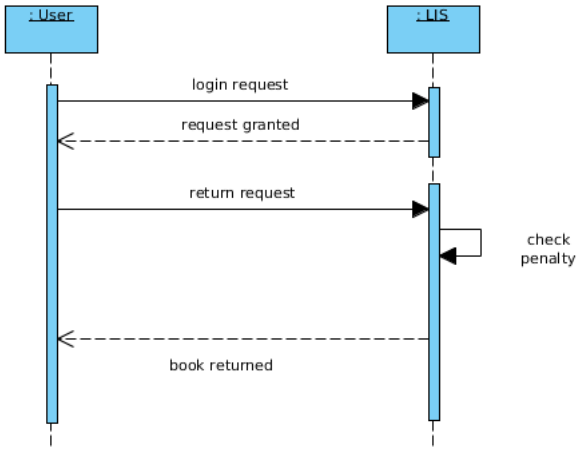
The user needs to succesfully login as the library clerk as he/she only has the priviledge to remove a book from the sytem if he/she feels that the book has been unused for a long period of time and is not required anymore.The user feeds the details of the book to be removed and the system automatically 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



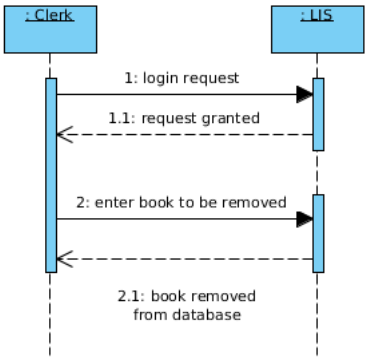
The user needs to succesfully login as a valid member or user of the library to issue a book. Also the book must be available in the library at that instant and also he/she must not exceed the maximum book count against his/her account in the library.The book is then added into the user's account and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

Return book Sequence Diagram



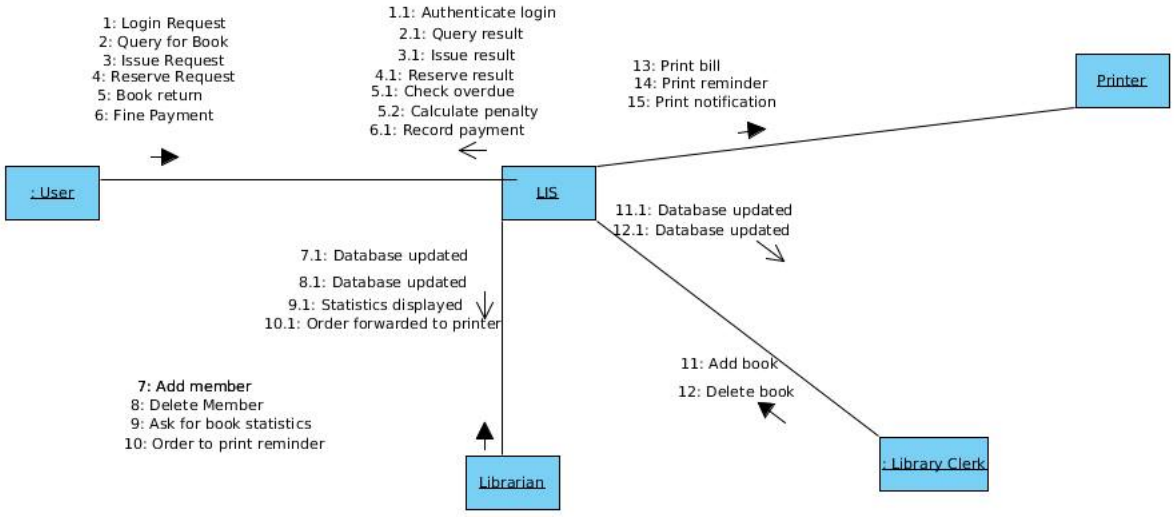
The user needs to successfully login as a valid member of the library to avail the option of returning the book.The book should not be overdue else the user has to pay a fine as per the rate predecided by the library authority. The book in any case is returned and all changes are updated in the user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

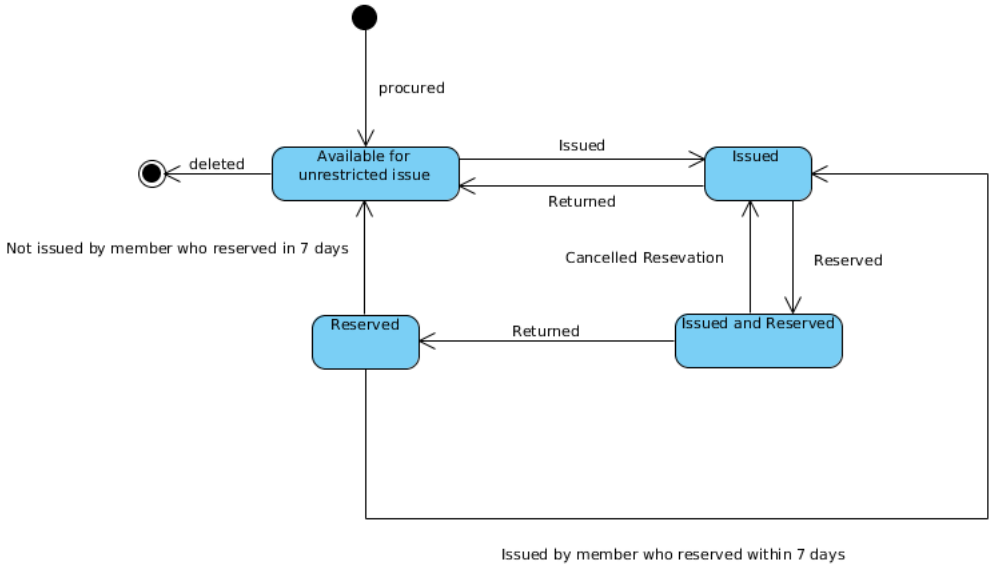
8.6 Collaboration Diagram



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

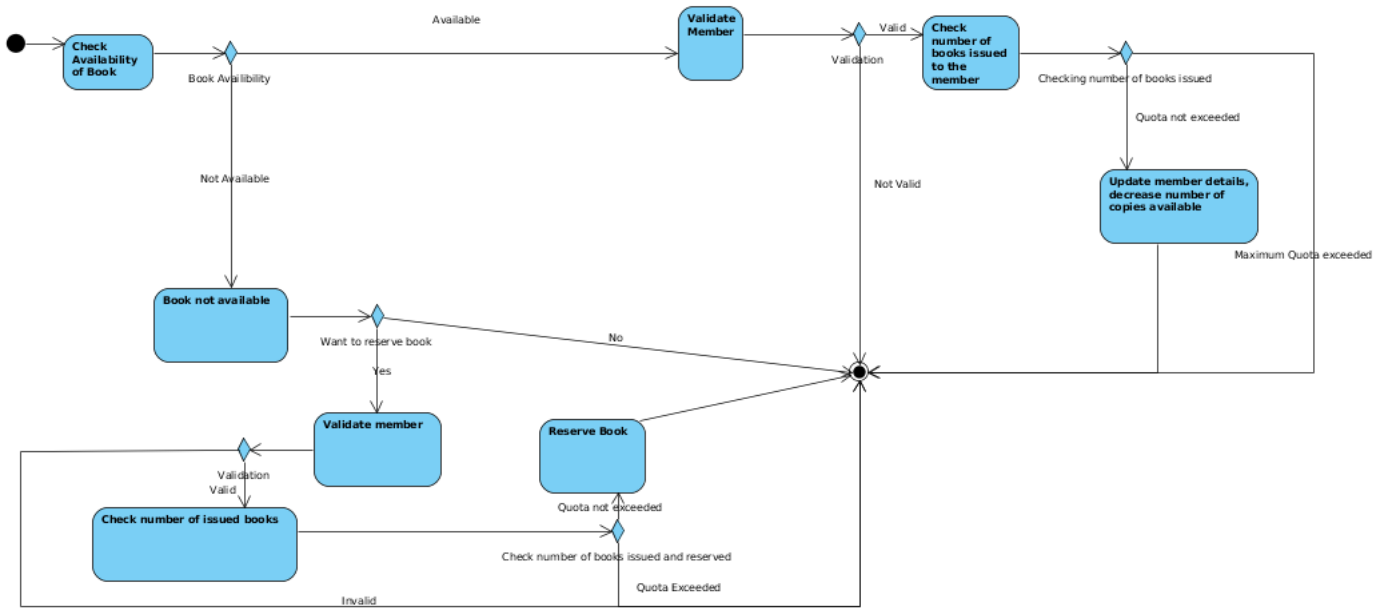
8.7 Statechart Diagram



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

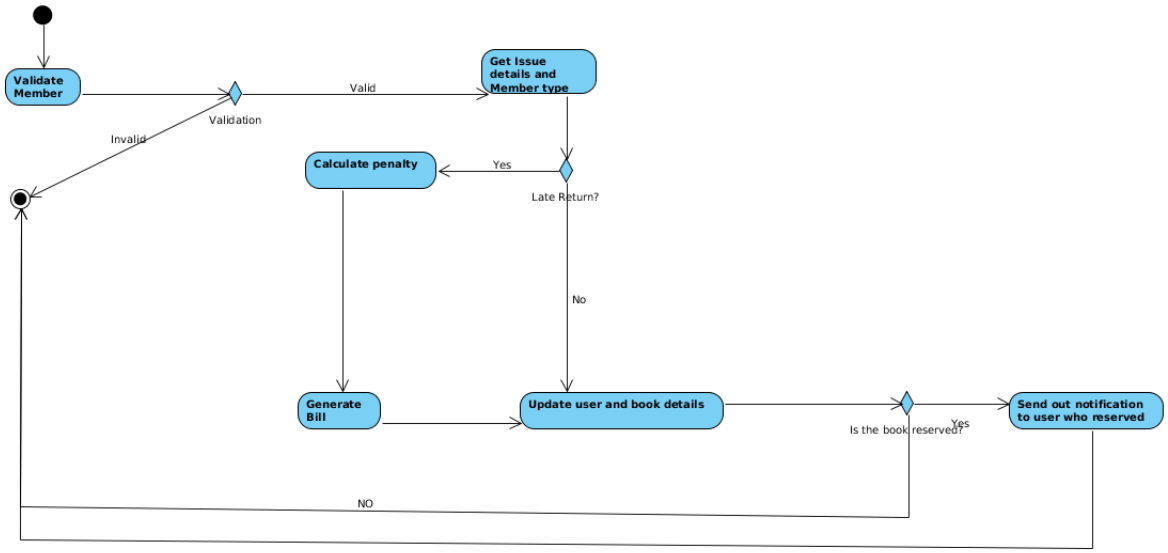
8.8 Activity Diagram

Issue Book Activity Diagram



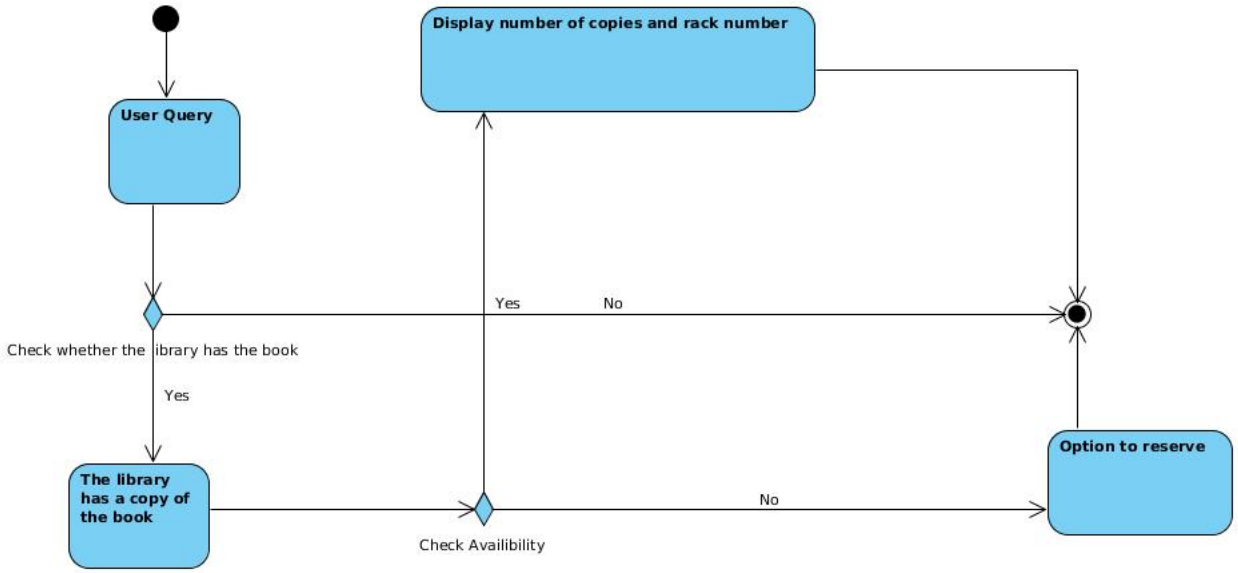
Description : First after succesful login the member has to check the availability of the book.We then check if the user is still allowed to issue books or if he has exceeded the maximum book count against his/her account.In case the book is avilable then the book is iisued and the required changes are 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



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.

8.9 Prototype Design

The prototype design involves creation of the follwing things :

User Login

Any user with a valid username and password will be able to login to the software to access the features. Initially just after the installation of the software only a librarian can be created in the software. Upon thd creation of the librarian the librarian can then create the other users in the library.

- For librarian :
  If a user logs in as librarian he is then shown a librarian home page screen with the functionalities exclusively of a libraraian.
- For clerk :
  If user logs in as clerkhe is then shown the clerk home page with the functions like adding or deleting book etc.
- For others(members) :
  We switch on to a new screen which is the home screen of the member. The member has the options of issuing, reserving and returning a book.

User creation

The librarian can be created in the beginning when the software is first installed. Once a librarian is creates an new librarian can not be created until and unless the old one is removed.
Only the librarian can create a new member of the library like the faculty or the students. During the creation of a member the librarian should also provide the type of the user as the maximim book count is different for different categories of the users.Once the type of a library member is given during creation one does not need to provide it anywhere else during issuing or returning of a book as it will be saved in the database and automatically fetched when required.
There are several rules while creating the user :

- The username can consist of alphanumeric characters along with an underscore (_)
- The password needs to consist atleast one of each type i.e. atleast one alphabet , atleast one number and atleast one special character.
  The password needs to atlest 8 characters long to ensure good security.
- The user type also needs to be provided.

Librarian Home Screen

The librarian hold the administrator access of the library.The feature available to him are as follows :

- Creation of user :
  For this purpose the librian will be directed to a new screen where he will have to provide the necessary details in order to create a new account and he has to provide the member type during account creation.
- Removing user :
  The librarian will provide with the details of the user to remove . If clicked to remove the details of the corresponding user will be removed from the database if and only if all his issued books are returned and there is no penalty due for him.
- Print notifications:
  The librarian can access any user history an can send a request or notification of printing the same
- Notification to print penalty bill :
  The librarian can send message to the printer to print the penalty bill.
- Notification to clerk to remove book :
  The librarian can also issue a request to the clerk to remove a book from the kibrary database is he finds it necessary.

User home screen

- Issue book:
  The user can issue book from this portal if the book is present in the library at that point of time.
- Return book :
  The user can return the book earlier issued via a portal.It also checks if the book is returned in time and if any fine is due.

- Reserve book :
  The user can reserve any book if it is not present at that point of time and the user will not exceed his maximum issue count.

- Search book :
  The user can search for any book in the library database by providing any details of the book.If it is present the software will display the rack number of the location of the book.

Clerk home screen

- Add book :
  The clerk can add a book into the library database with all the relevant details

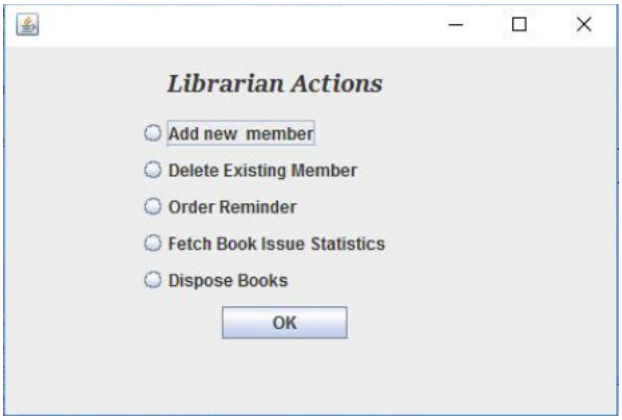
- Remove book :
  The clerk can remove any old and unused book from the library and remove all its details from the database.

8.10 Design I/O procedures and user interfaces

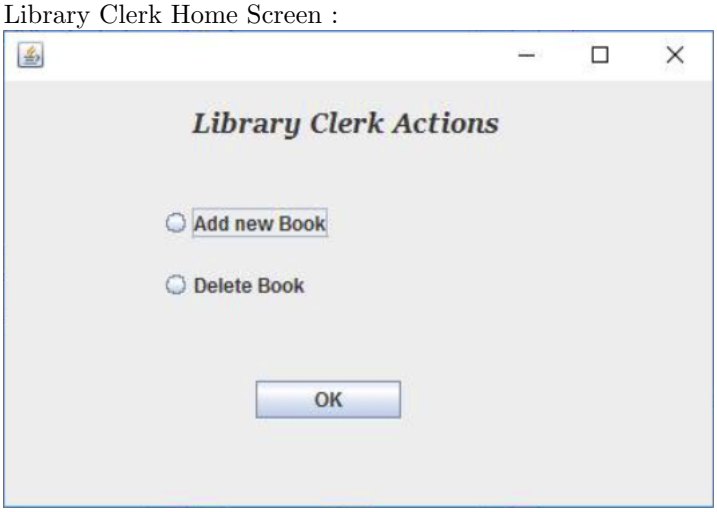
The software takes input through a interactive GUI application.
Few rudimentary GUI screen of the software have been provided in the section below for showing which are subject to changes as the coding stage progresses.

- Login Screen :

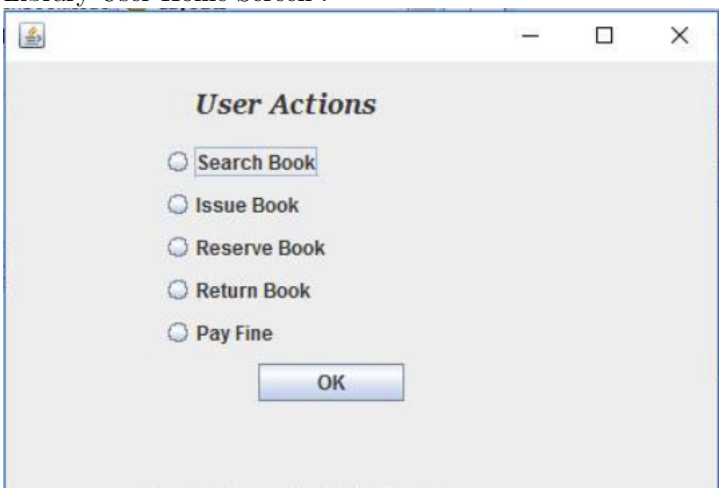
- Librarian Home Screen :



- Library Clerk Home Screen :



- Library User Home Screen :



8.11 Design of classes in target language

Java being an object oriented language implements classes as a blueprint or template of objects. Here the main classes shall be books and user. We shall be using multiple inheritance to classify users into UG students,PG students,Research Scholar and Faculty. All the data members will be made private to restrict them from being accessed and modified from outside.Some data members like user id which do need to be accessed outside will have getter functions and setter functions.

8.12 Exception Design

Exceptions which are present or handled in this software are as follows :

1. Username cannot contain any special characters except underscore.Entering any username not following this rule will throw an exception and prompt the user to re-enter the username.
2. The password of the user must be atleast 8 charactes long,must containt atleast one alphabet,atleast one digit and atleat one special character. Violation of these rules will throw an exeption and prompt the user to re-enter password.This is done to enhance the security level by making the password extremely difficult to crack by brute force algorithms.
3. The user type by default is empty in the initial loginscreen.The user must specify the user type while logging in.Keeping it default shall throw an exception and prompt the user to try and login again.
4. The user who has fulfilled his issue/reserve quota may click on issue book.This shall throw an exception and a dialogue box shall appear informing the user to return issued books or cancel reserved books(if any) and then try to issue books.
5. The user who has fulfilled his issue/reserve quota may click on reserve book.This shall throw an exception and a dialogue box shall appear informing the user to return issued books or cancel reserved books(if any) and then try to reserve books.

9 Adoptable Practices

Some adoptable practices for this software are:

- Reuse of code from one part to other wherever possible
- Division of code into small modules for better organization
- Using camel case and meaningful variable names