Software Requirements Specification

Version 1.0

September 25 , 2020

Library Management System

Tarun Kharkwal (2K20/SWE/24)
M.Tech. Software Engineering
1st Semester

LIBRARY MANAGEMENT SYSTEM

PROBLEM STATEMENT:

The Library Management System is one of the most efficient ways to keep tracks of books and journals in the library and having a large number of books and students also comes with so many challenges and requests that have to be handled. These challenges can include the following:

1. Delays in returning books.
2. Losing track of maintenance requests.
3. Delayed renewal of books.
4. None or delayed security deposits.
5. Improper records of a current student in the premise or any information of those that stayed there before.
6. No records of inspections reports.
7. Keeping records of books and students who issued that particular book.
8. Fine in book returning..

To overcome the above challenges with tenancies, we have decided to develop a computerized library management system (LMS) which will deal with the above situations by:

1. Issue/Return a book.
2. Attractive user interface.
3. Keep track of the book.
4. The report generation facility.
5. An authorized user can register himself.

Table of Contents

Revision History

Name	Date	Version
Library Management System	September 30, 2020	1.0.0

1. Introduction

1.1 Purpose

The purpose of this document is to present a detailed description of a Library Management System (LMS). It will explain the purpose and features of the software, the interfaces of the software, what the software will do and the constraints under which it must operate. This document is intended for users of the software and also potential developers.

1.2 Document Conventions

This Document was created based on the IEEE template for System Requirement Specification Documents.

1.3 Intended Audience and Reading Suggestions

Typical Users, such as vice-chancellor / librarian, who want to use the library management system for handling all library related matters (library information, fine payments, security deposits, book management, maintenance requests).
Advanced / Professional Users, such as engineers, researchers or programmers, who are interested in working on the project by further developing it, fixing bugs or enhancing its current features.

1.4 Product Scope

LMS is a system that library staff can use to manage their books information, handle maintenance requests from students and faculty, manage student and book information, send reminder messages to students such as renewal dates, delayed payments, and easily generate reports.
The system also aimed at allowing stakeholders to make online payments for the rent and security deposits.

Dos

1. Issue of login ID and password to system operators.
2. Maintain details of books available in the library.
3. Maintain details of the students in the university to provide student membership.
4. Maintain details of the faculty and employees in the university to provide faculty and employees membership.
5. Issue a book.
6. Bring a book back, and calculate fine if the book is being returned after the specified due date.

Don’ts

1. Periodicals section is not covered.
2. Book bank (if any) is not covered.
3. Records of digital books are not available.
4. Purchase of books facility is not available.

Benefits

The LMS provides the following benefits.

1. Easy searching of books.
2. Efficient issue and return of books.
3. Accurate and automated fine calculation.
4. Printing of reports.

1.5 Definitions, Acronyms and Abbreviations

SRS: Software requirement specification
LMS: Library management system
System operator: System administrator, library staff, data entry operator
RAM: Random access memory
Student: Any candidate admitted in a programme offered by a school.
System administrator/Administrator: User having all the privileges to operate the LMS.
Data entry operator (DEO): User having privileges to maintain book, student and faculty /employee details.
Library staff: User having privilege to issue, return, reserve and query books.
Faculty: Teaching staff of the university—Professor, Associate Professor, Assistant Professor

1.6 References

(a) IEEE Standard for Software Test Documentation—IEEE Std. 829-1998
(b) IEEE Recommended Practice for Software Requirements Specifications— IEEE Std 830-1998.
(c) slideshare.net
(d) Software Engineering by K.K. Aggarwal & Yogesh Singh, New Age Publishing House, 3rd Edition, 2008.

2. Overall Description

2.1 Product Perspective

The LMS shall be developed using client/server architecture and will be compatible with Microsoft Windows Operating System. The LMS is designed for university/college libraries to easily manage the book information, easily determine the available books and which books are taken by whom, and also allow library staff to easily keep track of books online for both rental and readers.

The system is also meant to easily deal with library maintenance requests such as renovations and arranging books and to purchase and avail new books for the library. This can easily help track the progress of these requests and easily assign staff to handle them. The system is also meant to help manage the library and send reminders to students on renewal dates, late payments fines and security deposit for books.

2.2 Product Function:

The LMS will allow access only to the authorized users with specific roles (system administrator, library staff, DEO and student). Depending upon the user's role, he/she will be able to access only specific modules of the system.

A summary of major functions that the LMS shall perform includes:

- A login facility for enabling only authorized access to the system.
- The system administrator/DEO will be able to add, modify, delete or view book, student, faculty/employee and login information.
- The system administrator/library staff will be able to issue, bring back, and reserve books.
- The system administrator/library staff will be able to query a book in order to check the availability of the book in the library.
- The system administrator/students/faculty/employee will be able to search a book from the library catalogue (author-wise, title-wise and publisher-wise).
- The system administrator/library staff will be able to generate various reports from the LMS.

2.3 User Classes and Characteristics:

- Typical users such as students who use the software for getting the functionality that they need for easily registering themselves like adding courses,viewing their fee details,viewing their schedule.
- Registrar,Administrator who has authoritative access over all the records and who can maintain the details of all the students and also cancel any invalid registration if made.
- Advanced / Professional Users, such as engineers, researchers or programmers, who are interested in working on the project by further developing it, fixing bugs or enhancing its current features.

2.4 Operating Environment

The Library Management System is designed to be accessed through most of the browsers and therefore is compatible with operating systems such as:

- Windows 7
- Windows 8
- Windows 10
- Mac OS X
- Linux

2.5 Design and Implementation Constraints

The Library Management System is developed in PHP (Yii2 framework) which handles every functionality in a different module and makes the best use of MVC (Model, View, and Controller). It has made use of bootstrap for its rich user friendly interface.

2.6 User Documentation

There is a quick start guide available on the github page.
https://github.com/tarunkharkwal98/Library-Management-System

2.7 Assumptions and Dependencies

- The acquisition section of the library will provide the list of the books purchased by the library.
- The academic section will provide the list of the admitted students (school-wise and programme-wise).
- The establishment section will provide the list of the faculty members in various schools of the university. It will also provide the list of employees working in various branches such as examination, academics, establishment and schools of the university.
- The login ID and password must be created by the system administrator and communi- cated to the concerned user confidentially to avoid unauthorized access to the system.

3. External Interface Requirements

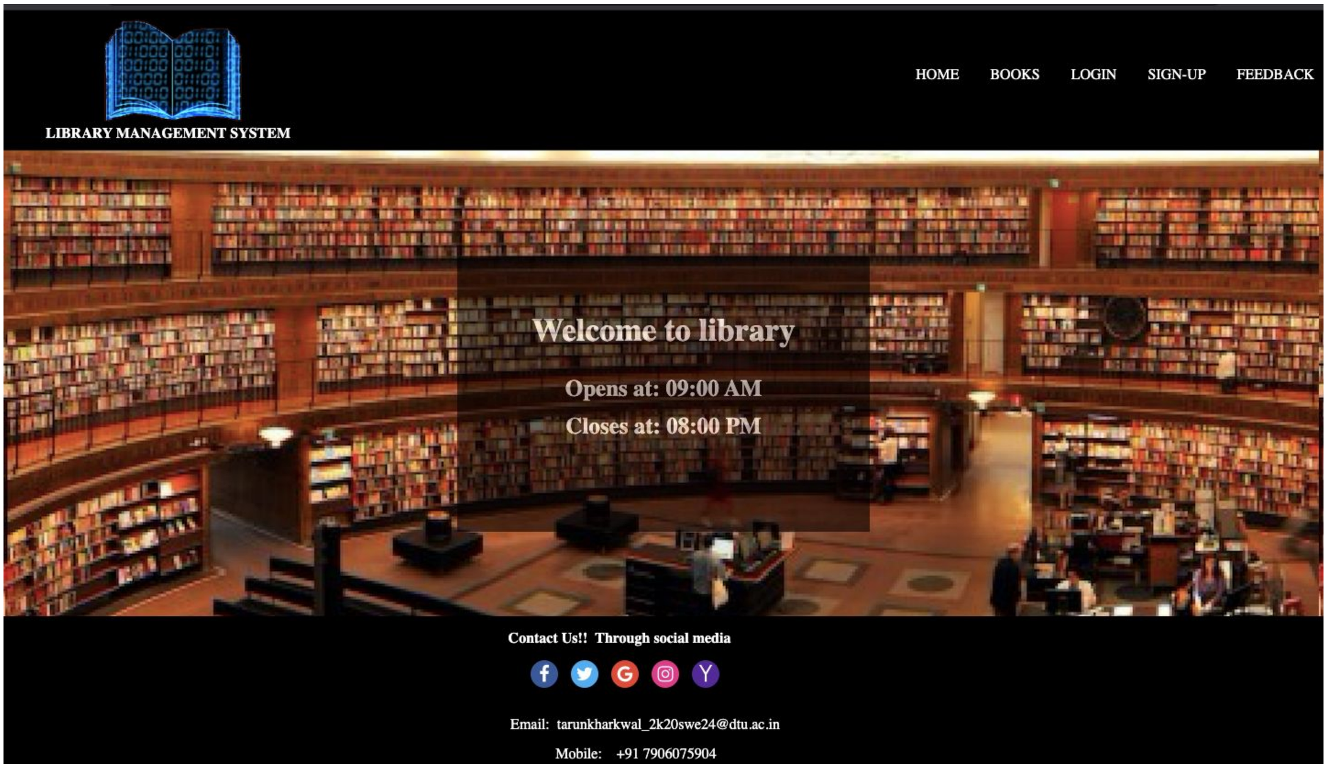
3.1 User Interfaces

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

- It allows user to view quick reports like Book Issued/Returned in between particular time.
- It provides stock verification and search facility based on different criteria.
- The user interface must be customizable by the administrator.
- All modules provided with the software must fit into a graphical user interface and accomplish to the standard defined.
- The design should be simple and all the different interfaces should follow a standard template.

The following user interfaces (or forms) will be provided by the system.

Home Page: This is the home page of LMS.



Welcome to library

Opens at: 09:00 AM

Closes at: 08:00 PM

Contact Us!! Through social media

Email: tarunkharkwal_2k20swe24@dtu.ac.in

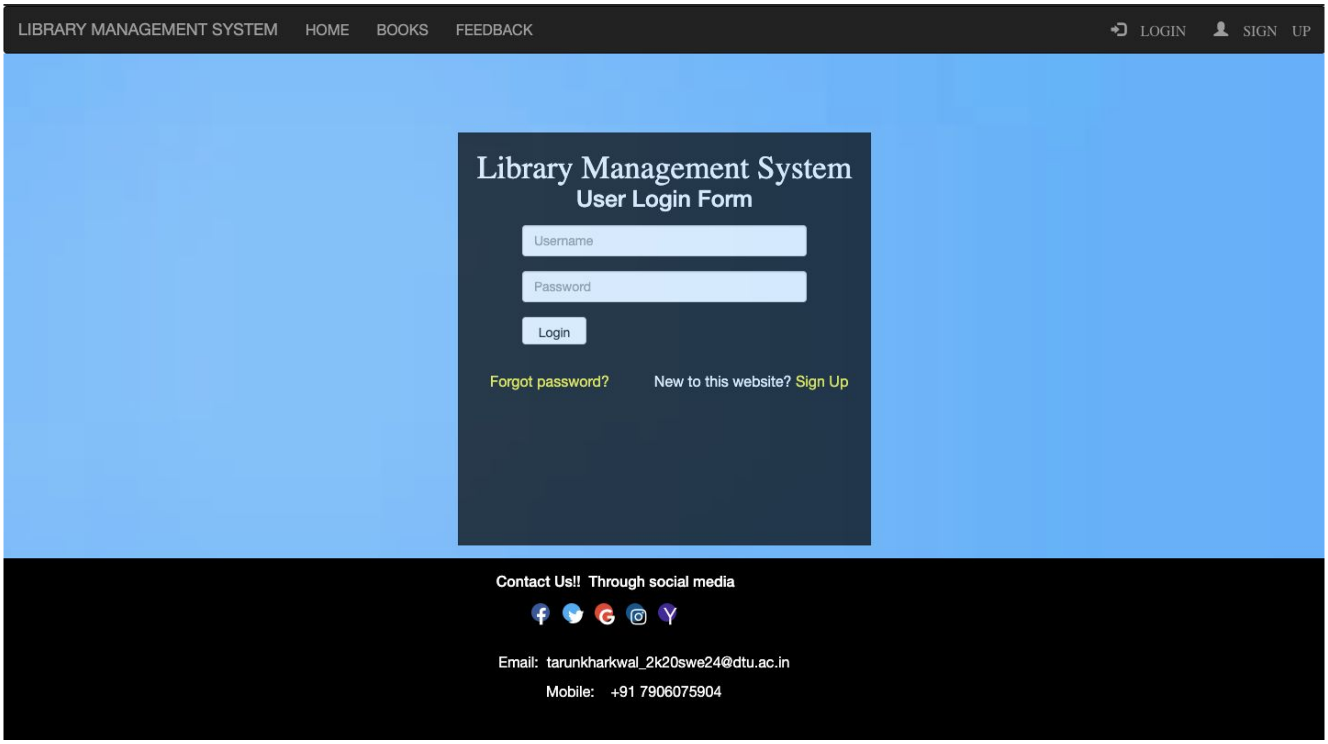
Mobile: +91 7906075904

Login Screen

This will be the first form, which will be displayed. It will allow the user to access the different forms based on his/her account.

Various fields available on this form will be:

- ***Login ID*****:** alphanumeric of length in the range of 4 to 15 characters. Special characters and blank spaces are not allowed.
- ***Password*****:** alphanumeric of length in the range of 4 to 15 characters. Blank spaces are not allowed. However, special characters are allowed.



Library Management System
User Login Form

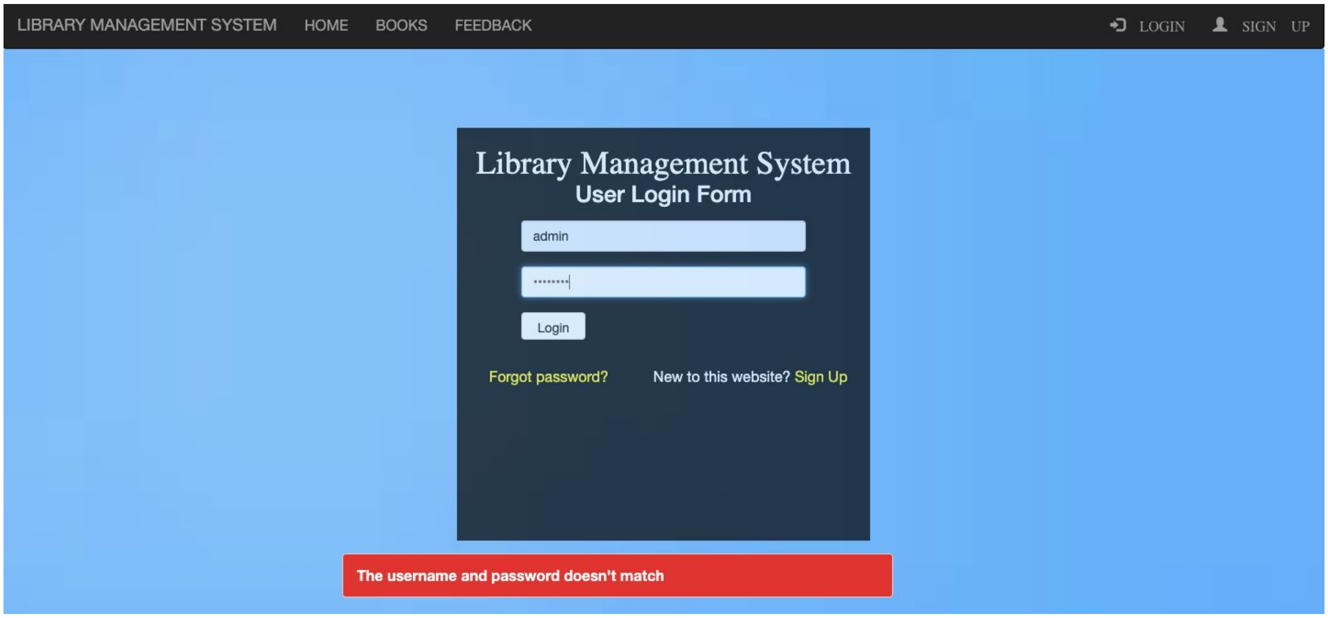
Forgot password? New to this website? Sign Up

Contact Us!! Through social media

Email: tarunkharkwal_2k20swe24@dtu.ac.in
Mobile: +91 7906075904

Output Screens:

Login Failed:



Library Management System
User Login Form

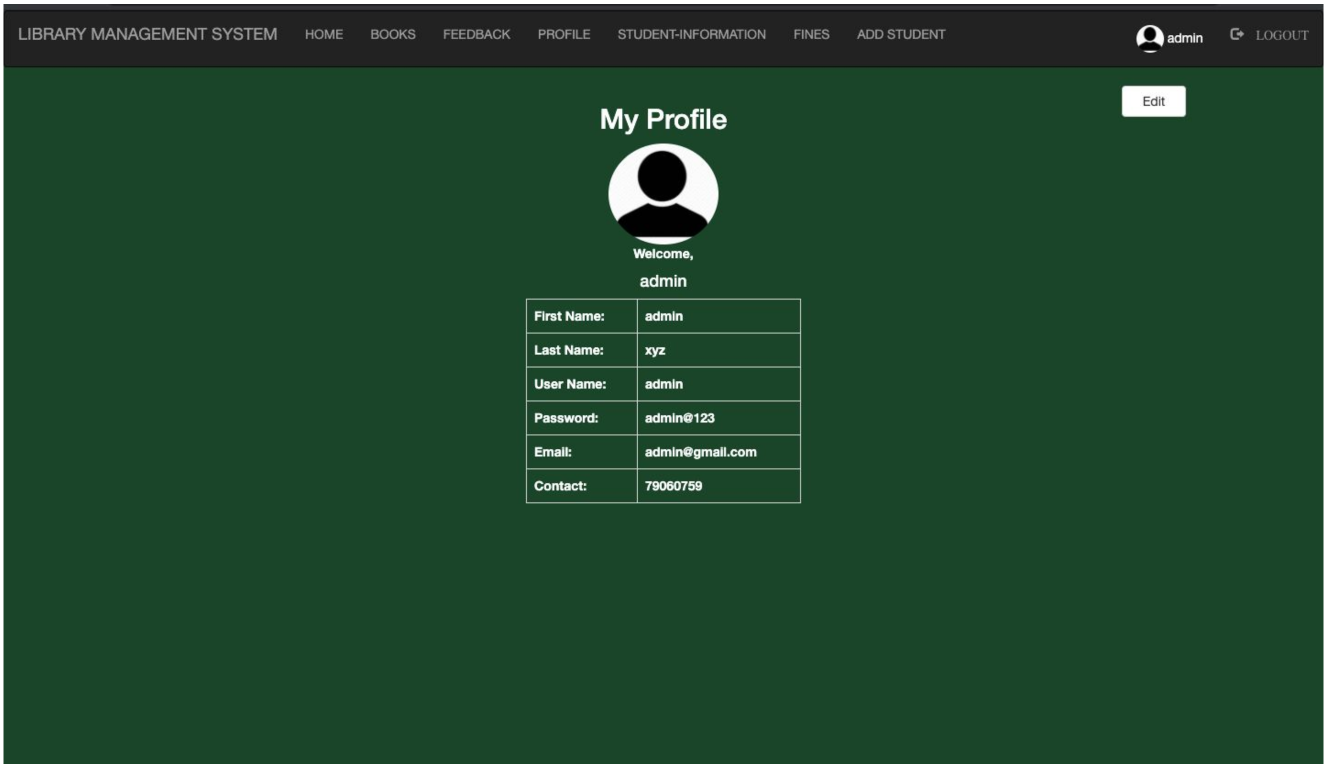
admin

••••••••

Forgot password? New to this website? Sign Up

The username and password doesn't match

Login: After successfully login this profile page will be opened.



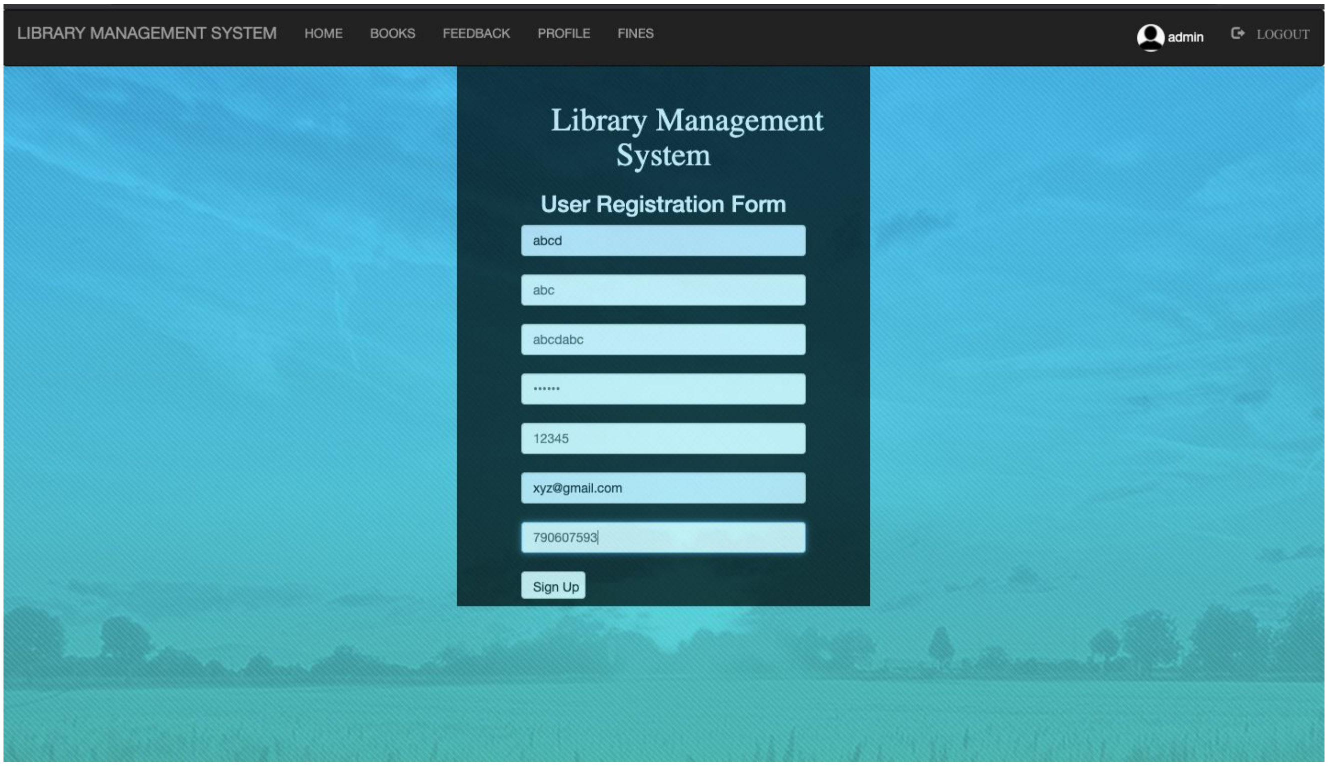
My Profile

Welcome,

admin

First Name:	admin
Last Name:	xyz
User Name:	admin
Password:	admin@123
Email:	admin@gmail.com
Contact:	79060759

Add Student: After login admin can add student to the library database using ADD STUDENT button. and the following page will open after clicking on the button.



Library Management System

User Registration Form

abcd

abc

abcdabc

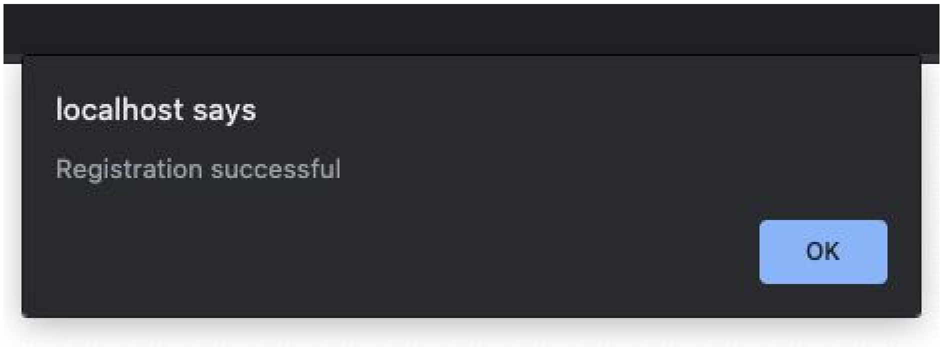
••••••

12345

xyz@gmail.com

790607593

After clicking the Sign Up button a pop window will be opened for successful registration of a student.

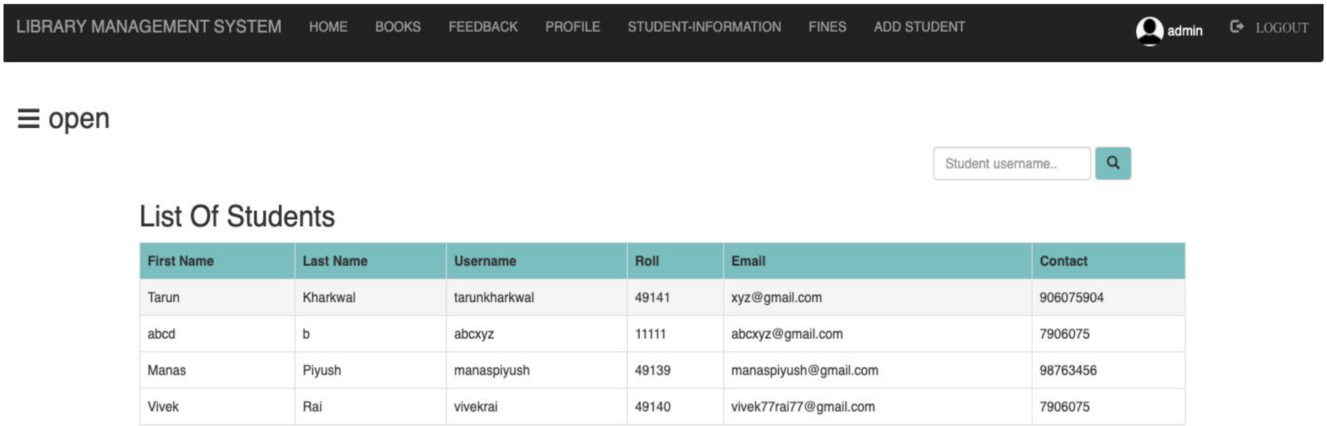


localhost says

Registration successful

Student-Information:

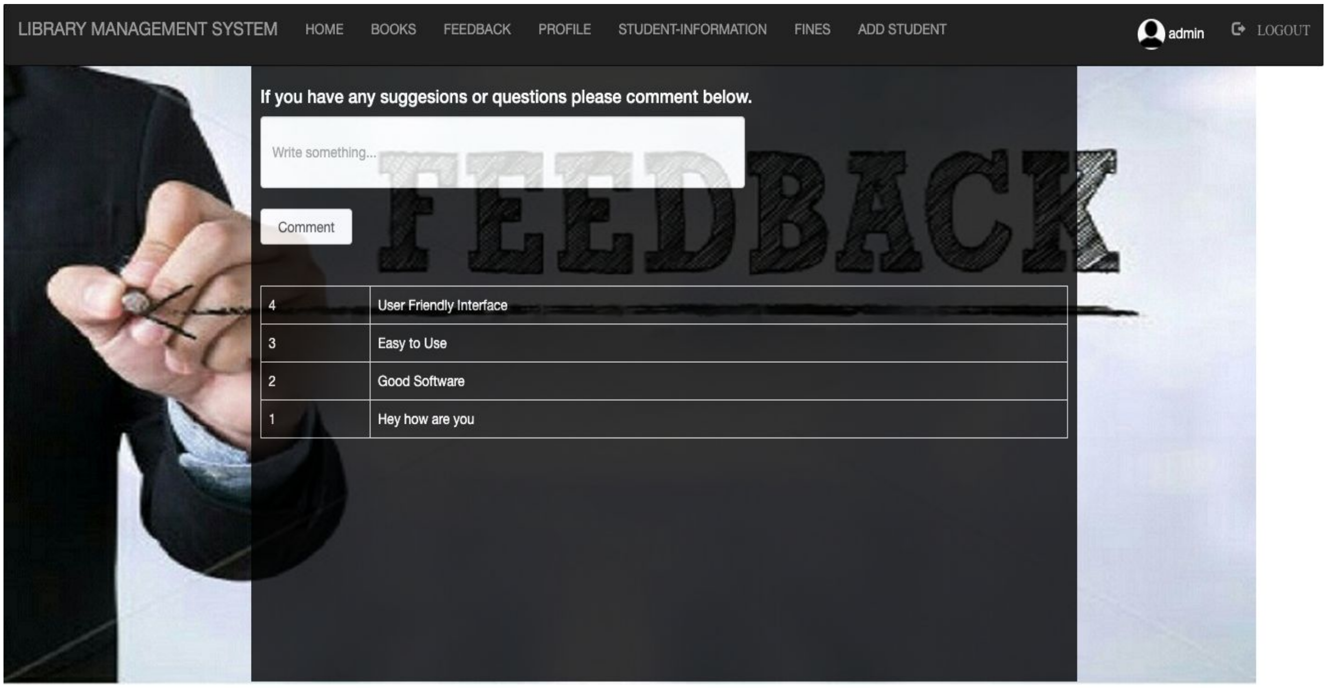
Then we can see students' details in the STUDENT-INFORMATION button.



List Of Students

First Name	Last Name	Username	Roll	Email	Contact
Tarun	Kharkwal	tarunkharkwal	49141	xyz@gmail.com	906075904
abcd	b	abcxyz	11111	abcxyz@gmail.com	7906075
Manas	Piyush	manaspiyush	49139	manaspiyush@gmail.com	98763456
Vivek	Rai	vivekrai	49140	vivek77rai77@gmail.com	7906075

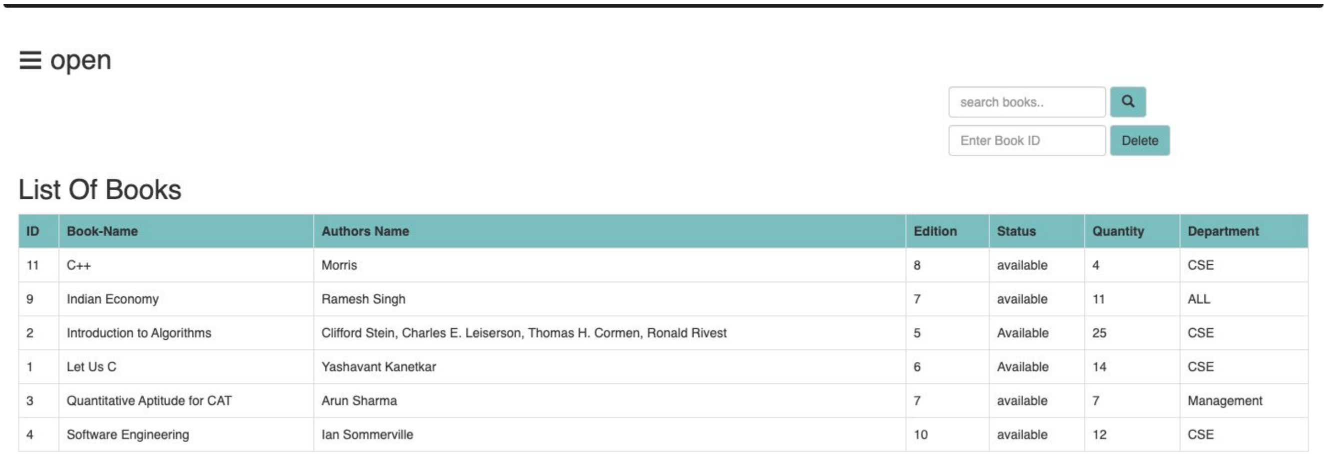
Feedback: Using a feedback button users can send feedback to the user and the user and the admin can see the feedback.



If you have any suggesions or questions please comment below.

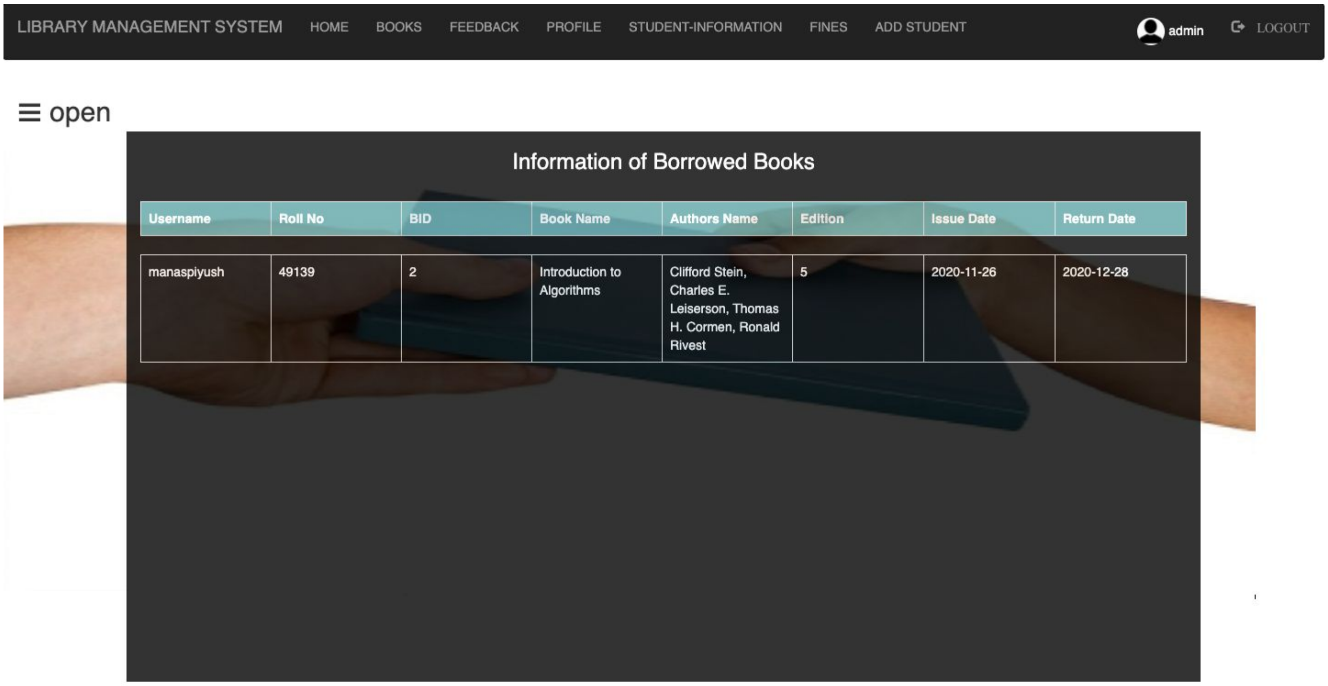
4	User Friendly Interface
3	Easy to Use
2	Good Software
1	Hey how are you

Book: Using a book button we can view, update, delete, add and query book details. And here are all the screenshots of the book page and functionalities of other buttons and other pages are given.



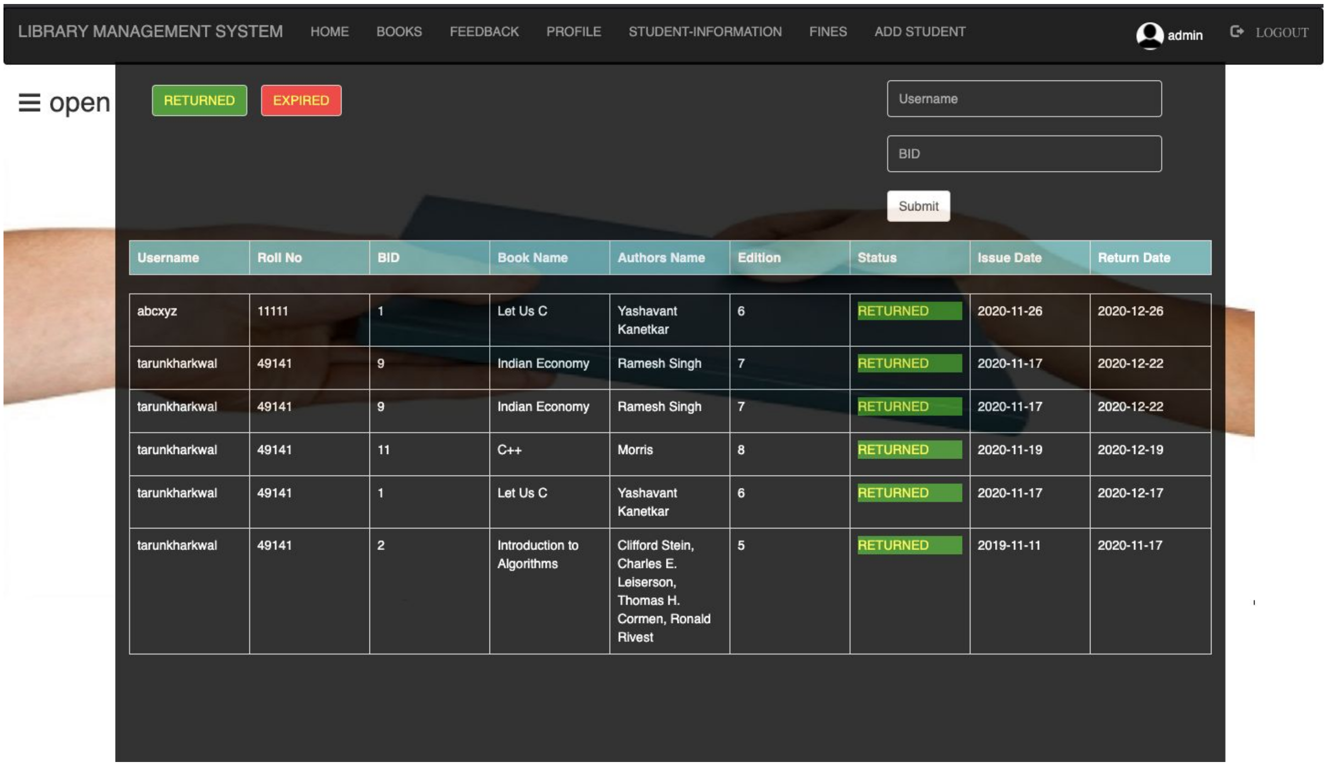
List Of Books

ID	Book-Name	Authors Name	Edition	Status	Quantity	Department
11	C++	Morris	8	available	4	CSE
9	Indian Economy	Ramesh Singh	7	available	11	ALL
2	Introduction to Algorithms	Clifford Stein, Charles E. Leiserson, Thomas H. Cormen, Ronald Rivest	5	Available	25	CSE
1	Let Us C	Yashavant Kanetkar	6	Available	14	CSE
3	Quantitative Aptitude for CAT	Arun Sharma	7	available	7	Management
4	Software Engineering	Ian Sommerville	10	available	12	CSE

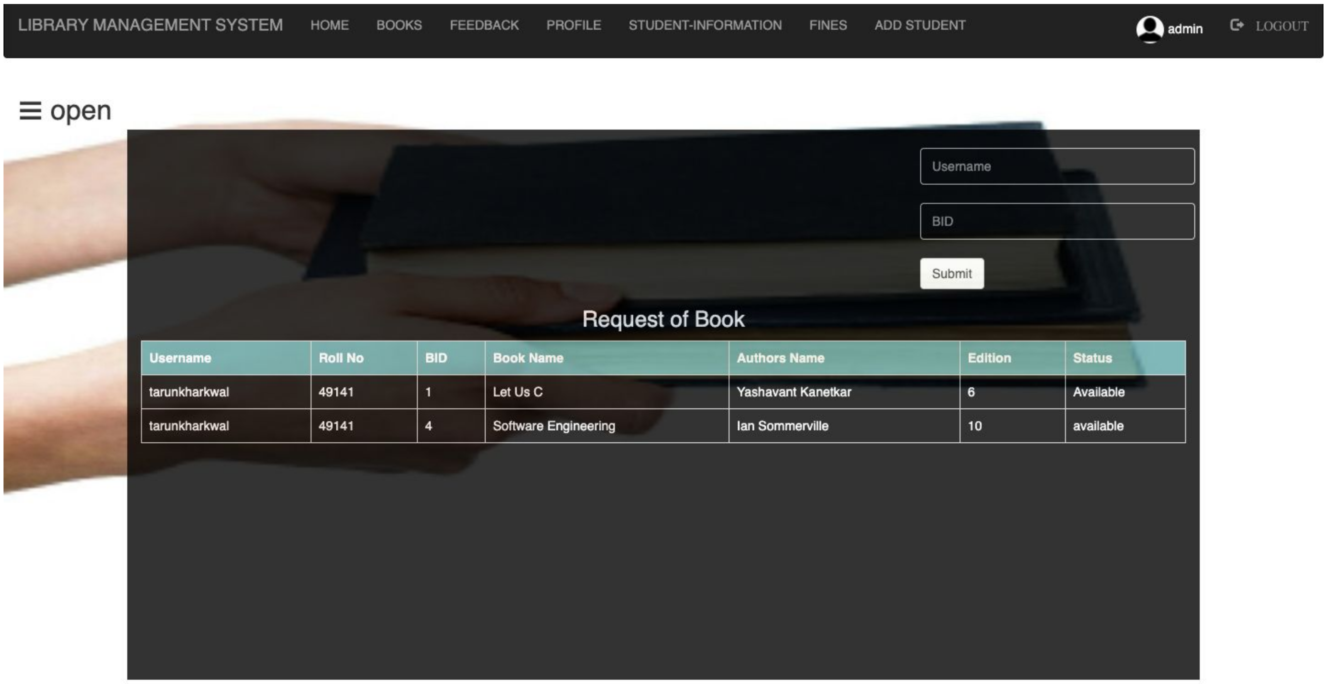


Information of Borrowed Books

Username	Roll No	BID	Book Name	Authors Name	Edition	Issue Date	Return Date
manaspiyush	49139	2	Introduction to Algorithms	Clifford Stein, Charles E. Leiserson, Thomas H. Cormen, Ronald Rivest	5	2020-11-26	2020-12-28



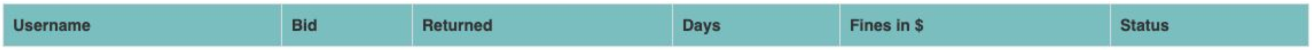
Username	Roll No	BID	Book Name	Authors Name	Edition	Status	Issue Date	Return Date
abcxyz	11111	1	Let Us C	Yashavant Kanetkar	6	RETURNED	2020-11-26	2020-12-26
tarunkharkwal	49141	9	Indian Economy	Ramesh Singh	7	RETURNED	2020-11-17	2020-12-22
tarunkharkwal	49141	9	Indian Economy	Ramesh Singh	7	RETURNED	2020-11-17	2020-12-22
tarunkharkwal	49141	11	C++	Morris	8	RETURNED	2020-11-19	2020-12-19
tarunkharkwal	49141	1	Let Us C	Yashavant Kanetkar	6	RETURNED	2020-11-17	2020-12-17
tarunkharkwal	49141	2	Introduction to Algorithms	Clifford Stein, Charles E. Leiserson, Thomas H. Cormen, Ronald Rivest	5	RETURNED	2019-11-11	2020-11-17



Request of Book

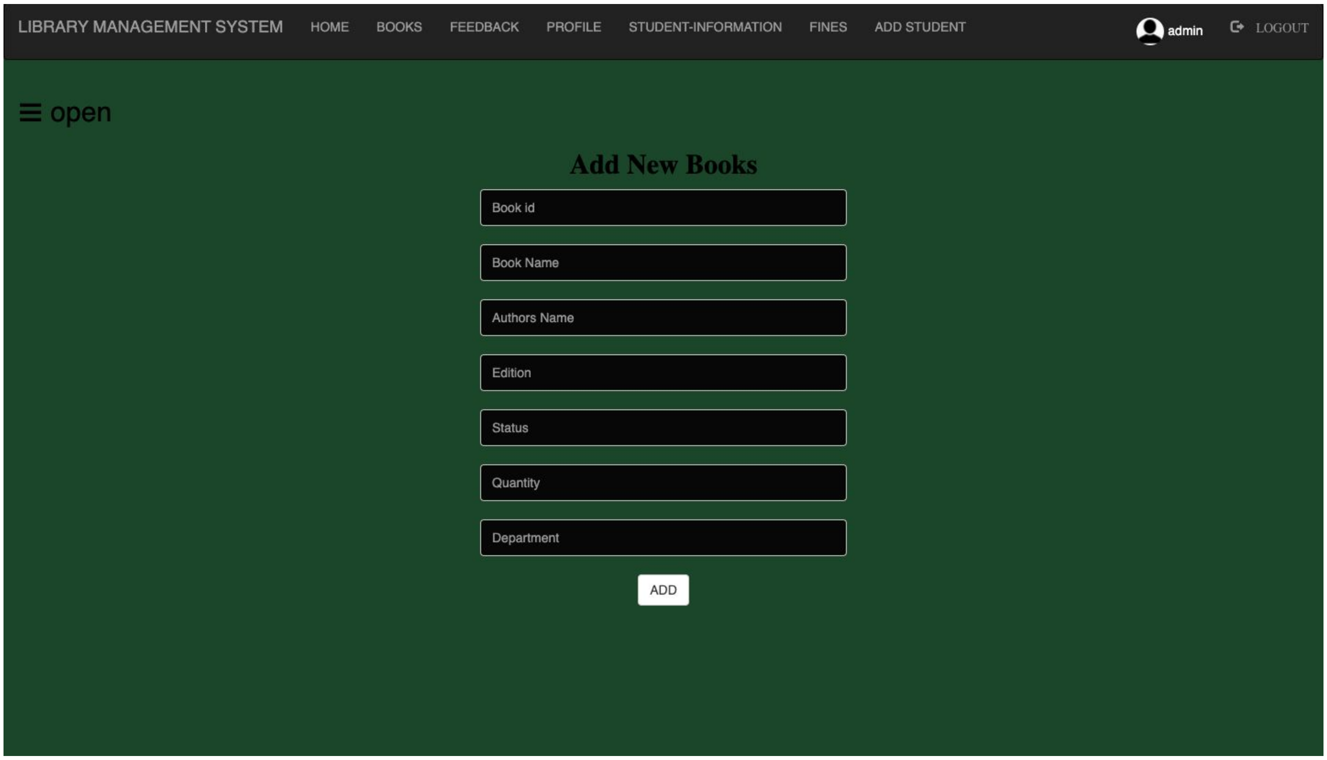
Username	Roll No	BID	Book Name	Authors Name	Edition	Status
tarunkharkwal	49141	1	Let Us C	Yashavant Kanetkar	6	Available
tarunkharkwal	49141	4	Software Engineering	Ian Sommerville	10	available

Fine: This is the interface for fine. If someone is eligible for fine we can see it there.

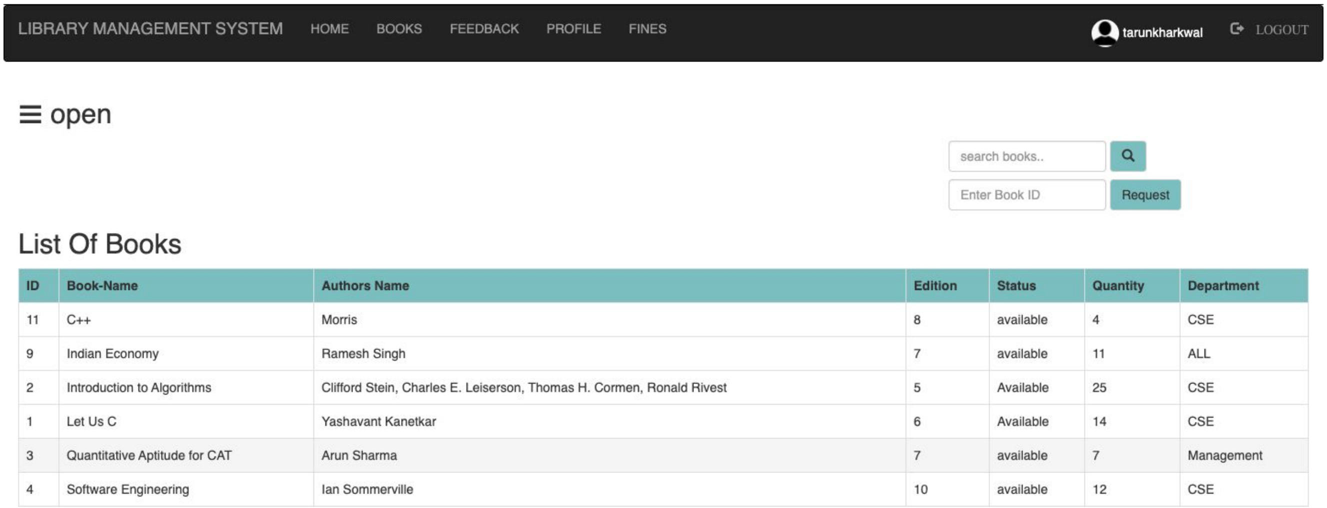


Username	Bid	Returned	Days	Fines in $	Status

Add Book: This is the add book interface for adding books in the system.



Request Book: Students can request for a book from here using the request button and he can also search for books and view the details of available books.



List Of Books

ID	Book-Name	Authors Name	Edition	Status	Quantity	Department
11	C++	Morris	8	available	4	CSE
9	Indian Economy	Ramesh Singh	7	available	11	ALL
2	Introduction to Algorithms	Clifford Stein, Charles E. Leiserson, Thomas H. Cormen, Ronald Rivest	5	Available	25	CSE
1	Let Us C	Yashavant Kanetkar	6	Available	14	CSE
3	Quantitative Aptitude for CAT	Arun Sharma	7	available	7	Management
4	Software Engineering	Ian Sommerville	10	available	12	CSE

3.2 Hardware Interfaces

Processor: i-3 processor
Hard Disk: 56GB or more
RAM : 256 MB or more

3.3 Software Interfaces

This package is developed using java as front end which is supported by the sun micro system. SQL server as backend to store the database.
Operating System: Windows XP, Windows 7, Windows 10, Macintosh
Language: HTML, CSS, JavaScript, PHP
Database: SQL Server (back end)
Software: Xampp, Google Chrome

3.4 Communication Interfaces

Describe the requirement associated with any communications functions required by LMS, including mail, web browser, network server communications protocols, electronic forms and so on. Define any pertinent message formatting. Identify any communication standards that will be used, such as FTP or HTTP. Specify any communication security or encryption issues, data transfer rates, any synchronization mechanism.

3.5 Software System Attributes

Usability

The application will be user-friendly and easy to operate, and the functions will be easily understandable.

Reliability

The applications will be available to the students throughout the registration period and have a high degree of fault tolerance.

Security

The application will be password protected. Users will have to enter the correct login ID and password to access the application.

Maintainability

The application will be designed in a maintainable manner. It will be easy to incorporate new requirements in the individual modules.

Portability

The application will be easily portable on any windows-based system that has SQL Server installed.

4. System Features

4.1 Use Case Diagram

This section demonstrates LMS's most prominent features and explains how they can be used and the results they will give back to the user.

4.1.1 Login

The system will control user login based on the category / rights that users have and whether he / she has been added to the system.

LOGIN:
Actors: Administrator, Library staff, Librarian, Data Entry Operator
Introduction: This use case describes all the steps followed in order to login into the Library Management System.
Pre-conditions: Administrator/Librarian/Data Entry Operator/Library staff must have valid username and password.
Post conditions: If username and password is valid, the actor is successfully logged in to the system; else an error message will be displayed.
Event Flow **Basic flow:** The system will request the actor to enter their login details for the system The actor fills in the following details: ● Username ● Password If login details are valid after clicking the login button, the actor is successfully logged into the system.
Alternate Flows: **Alternative Flow 1: Invalid Entry** If the username/password entered by the actor is invalid or if any field username/password are empty then the system will display an error message and the actor returns to the basic flow. **Alternative Flow 2: Forgot Password**

User clicks on the forgot password and the forgot password use case is called.

Alternative Flow 3:**User Exits**

Users can exit at any time and usecase ends.

Special requirements: None

Associated use cases: None

4.1.2 Issue Book

ISSUE BOOK:

Introduction:

This use case documents the steps that must be followed in order to get a book issued.

Actors: Administrator, Library staff

Precondition:

The administrator/library staff/librarian must be logged onto the system before the use case begins.

Post conditions:

If the use case is successful, a book is issued to the student/faculty/employee and the database is updated, else the system state remains unchanged.

Event Flow

Basic Flow

1. The student/faculty/employee membership number is read through the bar code reader
2. The system displays information about the student/faculty/employee.
3. Book information is read through the bar code reader.
4. The book is issued for the specified number of days and the return date of the book is calculated and
   stamped on the book.
5. The book and student/faculty/employee information is saved into the database.

Alternate Flows:

Alternative Flow 1: Unauthorized student/faculty/employee

If the system does not validate the student/faculty/employee membership number (due to membership expiry or any other reason), then an error message is flagged and the use case returns to the beginning of the basic flow.

Alternative Flow 2: Account is full

If the student/faculty/employee has requested a book and the account is full, i.e. he/she has already a maximum number of books issued on his/her name, then the request for issue is denied and the use case ends.

Alternative Flow 3: User exit

This allows the user to exit at any time during the use case. The use case ends.

Special requirements: None

Associated use cases: Login

4.1.3 Return Book

RETURN BOOK:

Introduction:

This use case documents the steps that must be followed in order to get a book issued.

Actors: Administrator, Library staff

Precondition:

The administrator/library staff must be logged onto the system before the use case begins.

Post conditions:

If the use case is successful, the book is returned back to the library and if needed, the "fine calculation" use case is called, otherwise the system state is unchanged.

Event Flow

Basic Flow

1. book information is read from the bar code of the book through the bar code reader.
2. The student/faculty/employee detail on whose name the books were issued is displayed on the system. The date of issue and return is also displayed.
3. The administrator/library staff checks the stamp on the book to check the duration of issue of the book.
4. The database is updated and the book status is updated.
5. The date stamp on the book is cancelled.

Alternate Flows:

Alternative Flow 1: Late return of book

If the duration for which the book has been kept by the student is more than 15 days, then a fine calculation use case is called. After the execution of the fine calculation use case, the original use case resumes the basic flow.

Alternative Flow 2: User exits

This allows the user to exit at any time during the use case. The use case ends.

Special requirements: None

Associated use cases: Login

4.1.4 Maintain Book Details

MAINTAIN BOOK DETAILS:

Introduction:

This use case documents the steps that the administrator/DEO must follow in order to maintain book details and add, update, delete and view book information.

Actors: Administrator, Librarian, DEO

Precondition:

The administrator/DEO/librarian must be logged onto the system before the use case begins.

Post conditions:

If the use case is successful, then the book information is added, updated, deleted or viewed. Otherwise, the system state is unchanged.

Event Flow

Basic Flow

1. Actor logs in to the system using his credentials.
2. Actors can access the database of books after log in into the system.
3. Actors can modify the database after log in into the system.
4. The database is updated and the book status is updated.

Alternate Flows

Alternative Flow 1: Invalid Entry

If in the **Add a Book** or **Update a Book** flow, the actor enters invalid accession number/ISBN/title/author/publisher/language/subject descriptor or leaves the accession number/ISBN/title/author/publisher/language/subject descriptor empty, the system displays an appropriate error message. The actor returns to the basic flow and may reenter the invalid entry.

Alternative Flow 2: Book Already Exists

If in the **Add a Book** flow, a book with a specified accession number already exists, the system displays an error message. The administrator returns to the basic flow and may reenter the book.

Alternative Flow 3: Book Not Found

If in the **Update a Book** or **Delete a Book** or **View a Book** flow, the book information with the specified accession number does not exist, the system displays an error message. The administrator returns to the basic flow and may reenter the accession number.

Alternative Flow 4: Update Cancelled

If in the **Update a Book** flow, the administrator/DEO decides not to update the book, the update is cancelled and the **Basic Flow** is restarted at the beginning.

Alternative Flow 5: Delete Cancelled

If in the **Delete a Book** flow, the administrator/DEO decides not to delete the book, the delete is cancelled and the **Basic Flow** is restarted at the beginning.

Alternative Flow 6: Deletion Not Allowed

If in the **Delete a Book** flow, issue/return/reserve details of the book selected exist, then the system displays an error message. The administrator returns to the basic flow and may reenter the student membership number.

Alternative Flow 7: User Exits

This allows the user to exit at any time during the use case. The use case ends.

Special requirements: None

Associated use cases: Login

Validity Checks

(1) Only the administrator/DEO will be authorized to access the Book Details module.
(2) Every book will have a unique accession number.
(3) Accession number cannot be blank.
(4) Accession number can only have values from 10 to 99999 digits.
(5) Subject descriptor cannot be blank.
(6) ISBN number cannot be blank.
(7) Length of ISBN number for any user can only be equal to 11 digits.
(8) ISBN numbers will not accept alphabets, special characters and blank spaces.

4.1.5 Maintain Students Details

MAINTAIN STUDENTS DETAILS:

Introduction:

This use case documents the steps that must be followed in order to maintain students details.

Actors: Administrator, Librarian, DEO

Precondition:

The administrator/DEO/librarian must be logged onto the system before the use case begins.

Post conditions:

If the use case is successful, the student details are maintained in the system and state remains the same.

Event Flow

Basic Flow

1. This use case starts when the administrator/DEO wishes to add, update, delete or view student information from the system.
2. The system requests that the administrator/DEO specify the function he/she would like to perform (either add a student, update a student record, delete a student record). Once the administrator/DEO provides the requested information, one of the subflows is executed.
   - If the administrator/DEO selects "Add a Student", the **Add a Student** subflow is executed.
   - If the administrator/DEO selects "Update a Student", the **Update a Student** subflow is executed.
   - If the administrator/DEO selects "Delete a Student", the **Delete a Student** subflow is executed.
   - If the administrator/DEO selects "View a Student", the **View a Student** subflow is executed.

Alternate Flows:

Alternative Flow 1: Invalid Entry

If in the **Add a Student** or **Update a Student** flow, the actor enters invalid photograph/ Roll No./Name/School/Programme/Father's name/Date of birth/Address/Telephone/Email/ Membership date/Valid up to/Password or leaves the photograph/Roll No./Name/School/ Programme/Father's name/Date of birth/Address/Telephone/Email/Membership date/Valid up to/Password empty, the system displays an appropriate error message. The actor returns to the basic flow and may reenter the invalid entry.

Alternative Flow 2: Student Already Exists

If in the **Add a Student** flow, a student with a specified membership number already exists, the system displays an error message. The administrator returns to the basic flow and may reenter the student information.

Alternative Flow 3: Student Not Found

If in the **Update a Student** or **Delete a Student** or **View a Student** flow, the student information with the specified membership number does not exist, the system displays an error message. The administrator returns to the basic flow and may reenter the membership number.

Special requirements: None

Associated use cases: Login

Validity Checks

(i) Only the administrator/DEO will be authorized to access the Student Membership Details module.
(ii) Every student will have a unique membership number.
(iii) Roll no. cannot be blank.
(iv) Length of Roll no. for any user can only be equal to 10 digits.
(v) Roll no. cannot contain alphabets, special characters and blank spaces.
(vi) Student name cannot be blank.
(vii) Length of student name can be of 3 to 50 characters.
(vii) Student name will only accept alphabetic characters and blank spaces.
(ix) Date of birth cannot be blank.
(x) Phone cannot be blank.

4.1.6 Generate Report

GENERATE REPORT

Introduction:

This use case documents the steps that must be followed in order to generate reports.

Actors: Administrator, Librarian, Library staff

Precondition:

The administrator/librarian/library staff must be logged onto the system before the use case begins.

Post conditions:

If the use case is successful, reports are generated by the system and state remains the same.

Event Flow

Basic Flow

1. Actor logs in to the system using his credentials.
2. Actors can access the database of the library system after log in into the system.
3. The actor can generate the report of the system.

Alternate Flows:

Alternative Flow 1: User exits

This allows the user to exit at any time during the use case. The use case ends.

Special requirements: None.

Associated use cases: Login

4.1.7 Search Catalog

SEARCH CATALOG

Introduction:

This use case documents the steps that must be followed in order to search the book in catalog.

Actors: Administrator, Librarian, Student

Precondition:

None

Post conditions:

If the use case is successful, the actor can search his/her book in the library.

Event Flow

Basic Flow

Actors can search for the book in the database of books.

Alternate Flows:

Alternative Flow 1: User exits

This allows the user to exit at any time during the use case. The use case ends.

Special requirements: None.

Associated use cases: None.

4.1.8 Log Out

The system will securely log the user out of the system once he/she clicks on the logout button.

LOG OUT

Introduction:

This use case describes all steps taken in order to exit from the system.

Actors: Admin, DEO, Student, librarian, library staff

Pre-conditions:

The Admin/DEO/Student/librarian/library staff must be logged in to the system.

Post conditions:

If this use case is successful then Admin/DEO/Student/librarian/library staff successfully exits from the system, else the system remains unchanged.

Basic flow:

1. Admin/DEO/Student/librarian/library staff makes a request to logout from the system.
2. The system sends a confirmation request.
3. Admin/DEO/Student/librarian/library staff confirms the request and the system returns to the home page.

Alternative flow:

Alternative Flow 1:

When the system sends a confirmation request if the Admin/DEO/Student/librarian/library staff cancels that request then system state remains unchanged.

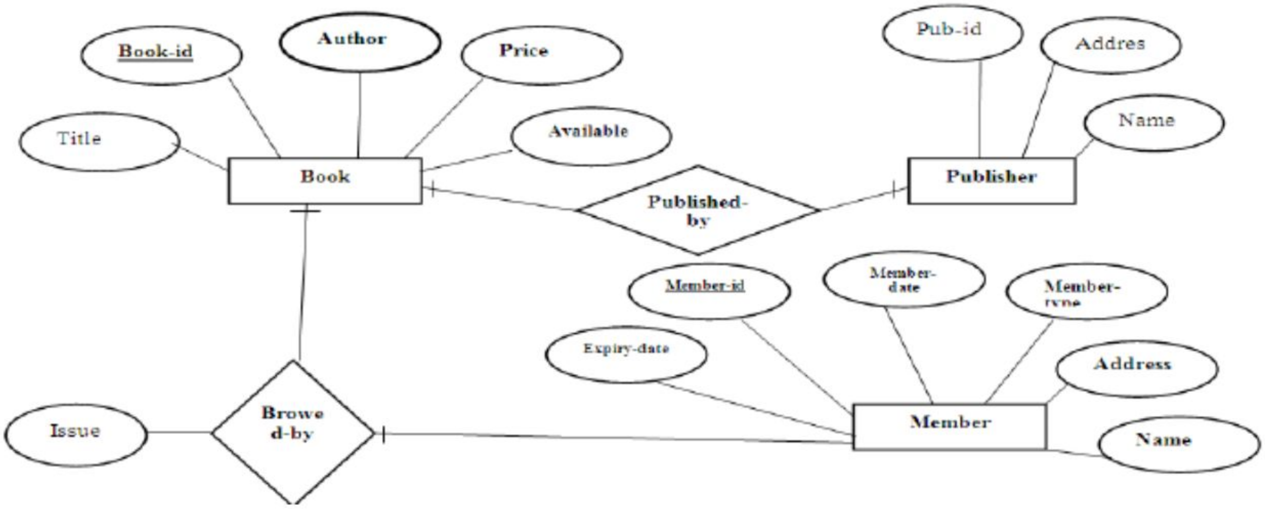
Alternative Flow 2:

The User can exit at any time and use case ends.

Special Requirements: None

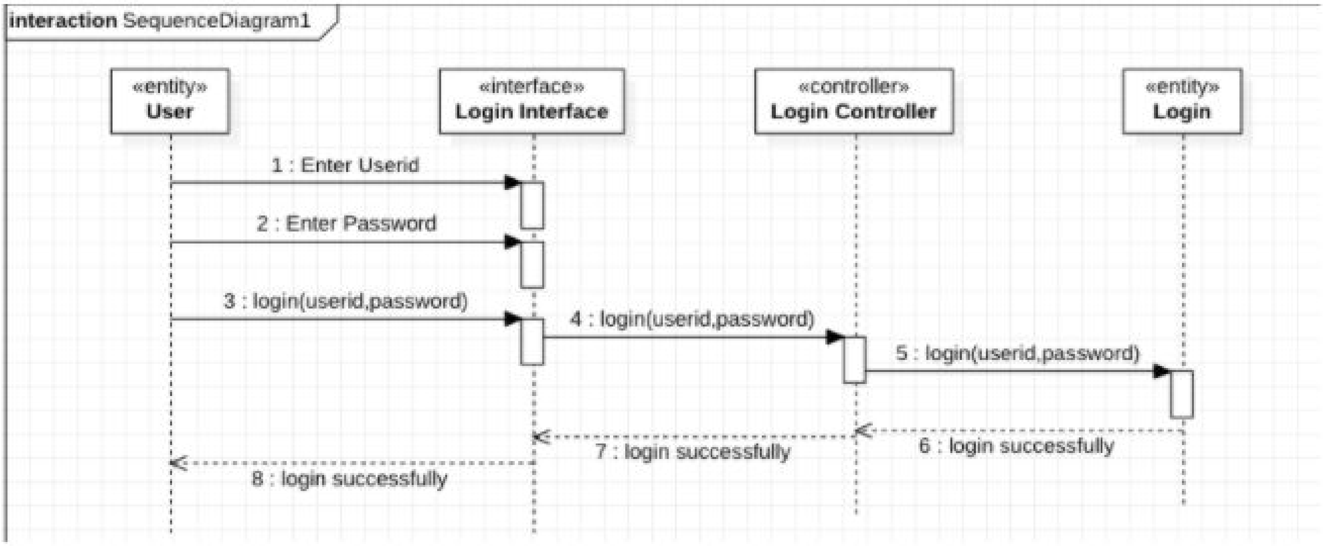
Associated Use Case: Login

4.2 ER Diagram For LMS:

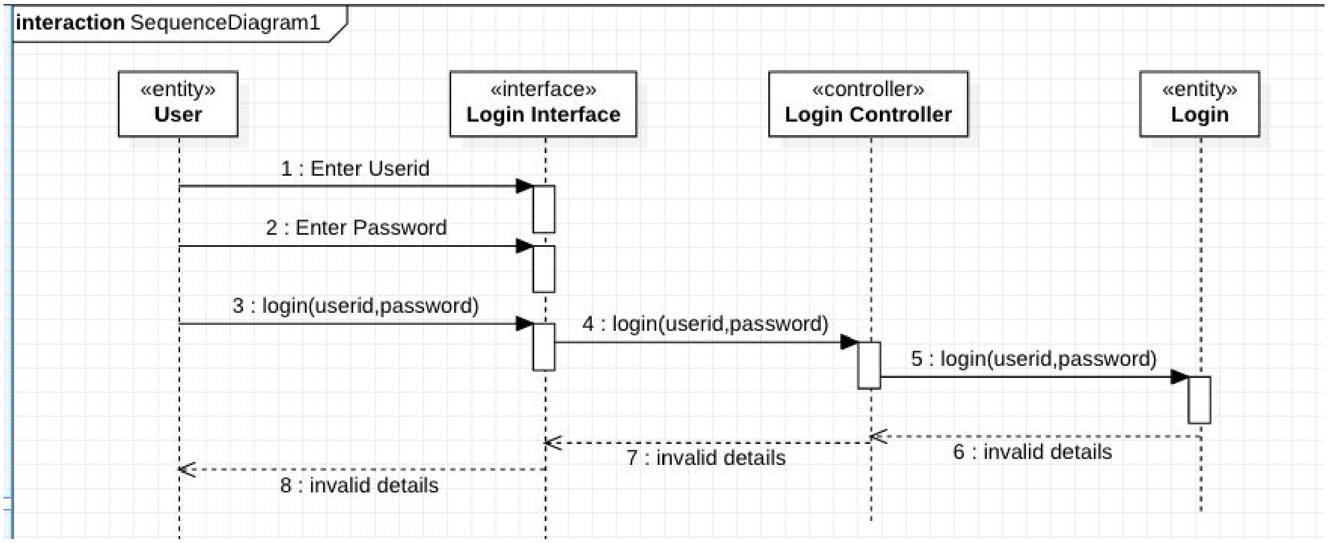


4.3 Sequence Diagram:

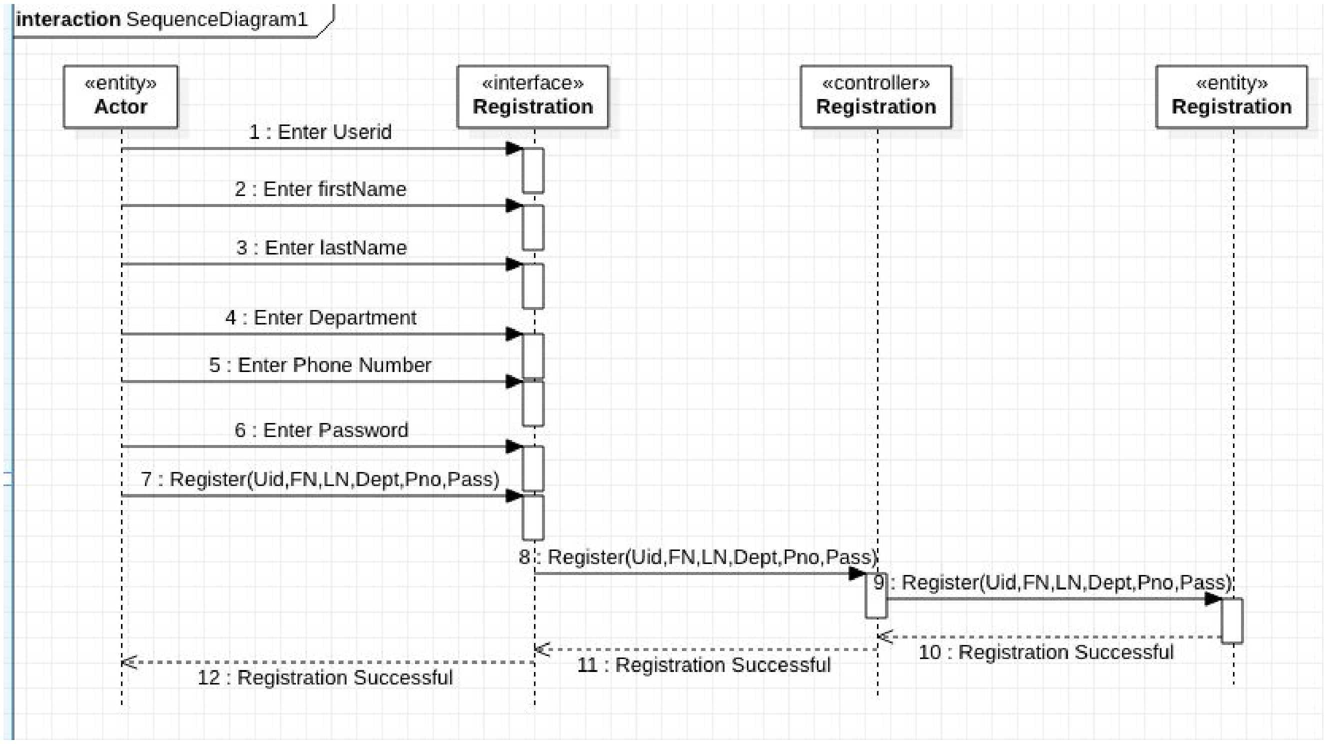
LOGIN BASIC SEQUENCE DIAGRAM:



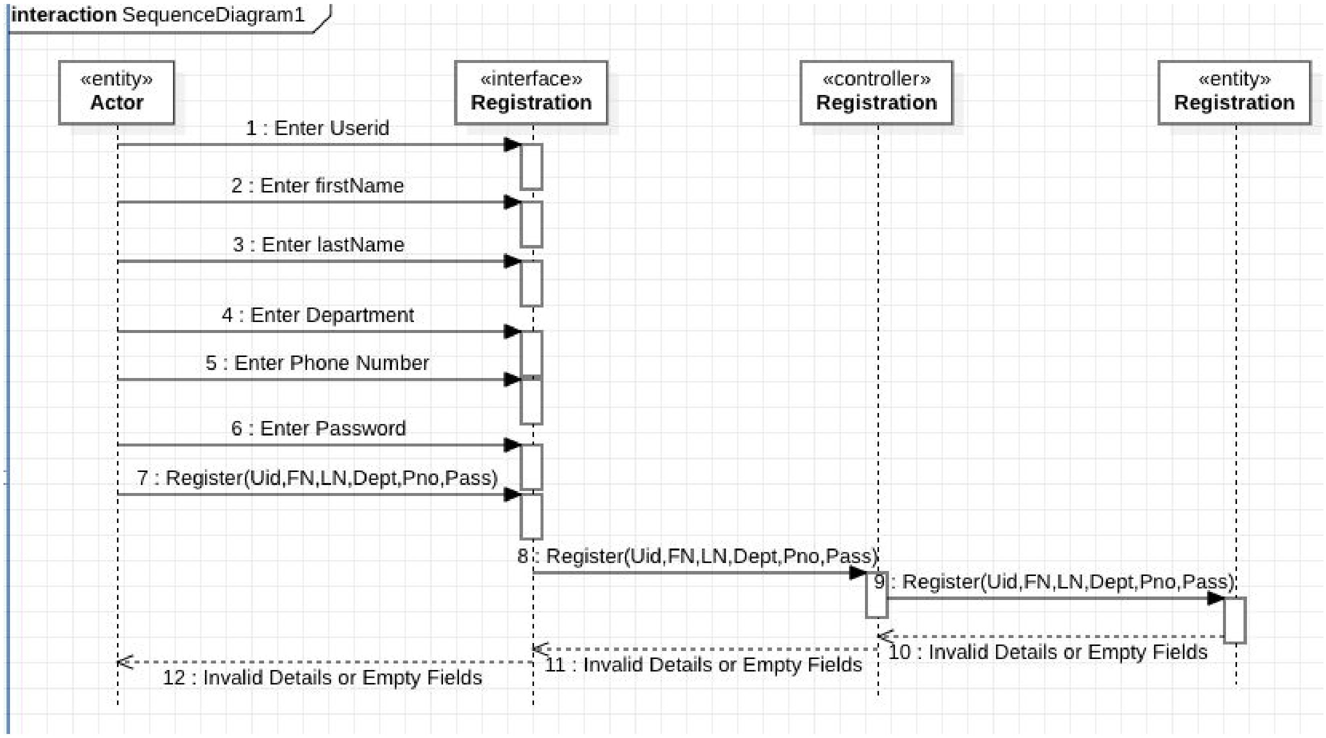
LOGIN ALTERNATIVE SEQUENCE DIAGRAM:



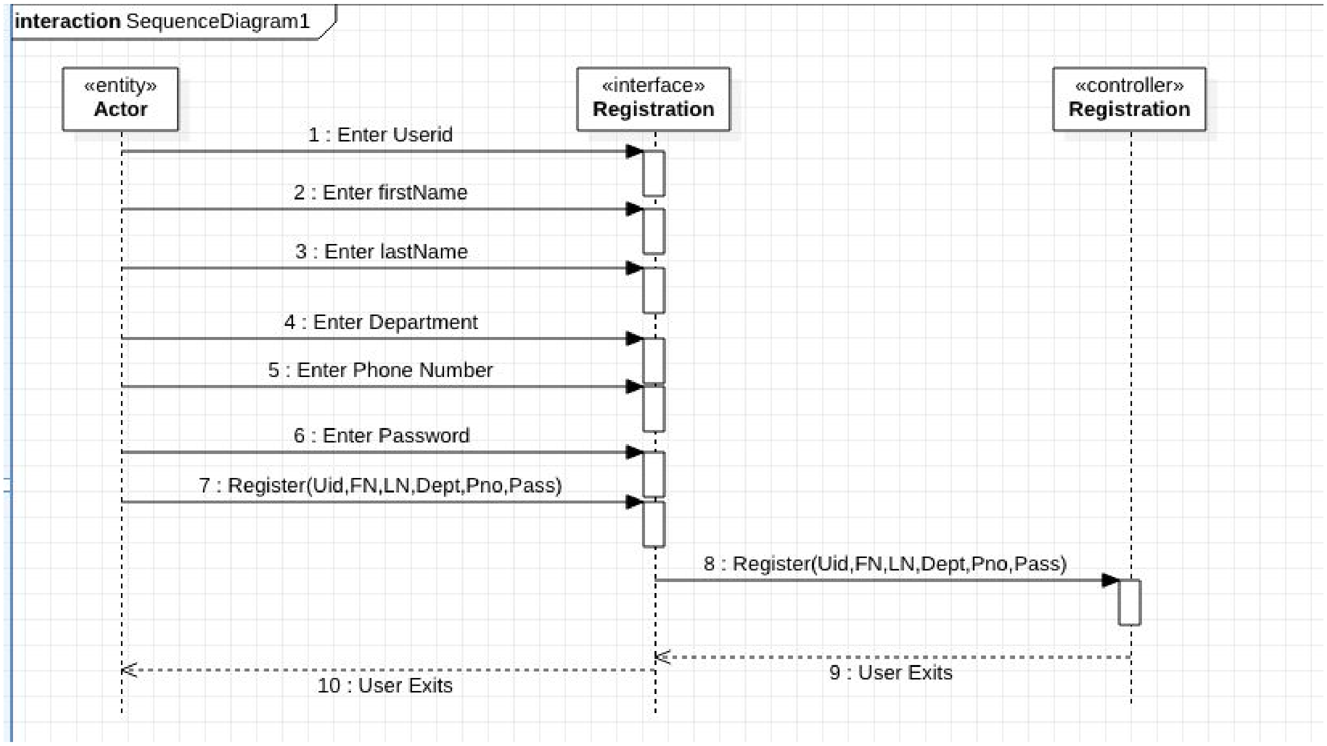
ENTER STUDENT DETAILS BASIC SEQUENCE DIAGRAM:


interaction SequenceDiagram1
«entity»
Actor
«interface»
Registration
«controller»
Registration
«entity»
Registration
1 : Enter Userid
2 : Enter firstName
3 : Enter lastName
4 : Enter Department
5 : Enter Phone Number
6 : Enter Password
7 : Register(Uid,FN,LN,Dept,Pno,Pass)
8 : Register(Uid,FN,LN,Dept,Pno,Pass)
9 : Register(Uid,FN,LN,Dept,Pno,Pass)
10 : Registration Successful
11 : Registration Successful
12 : Registration Successful

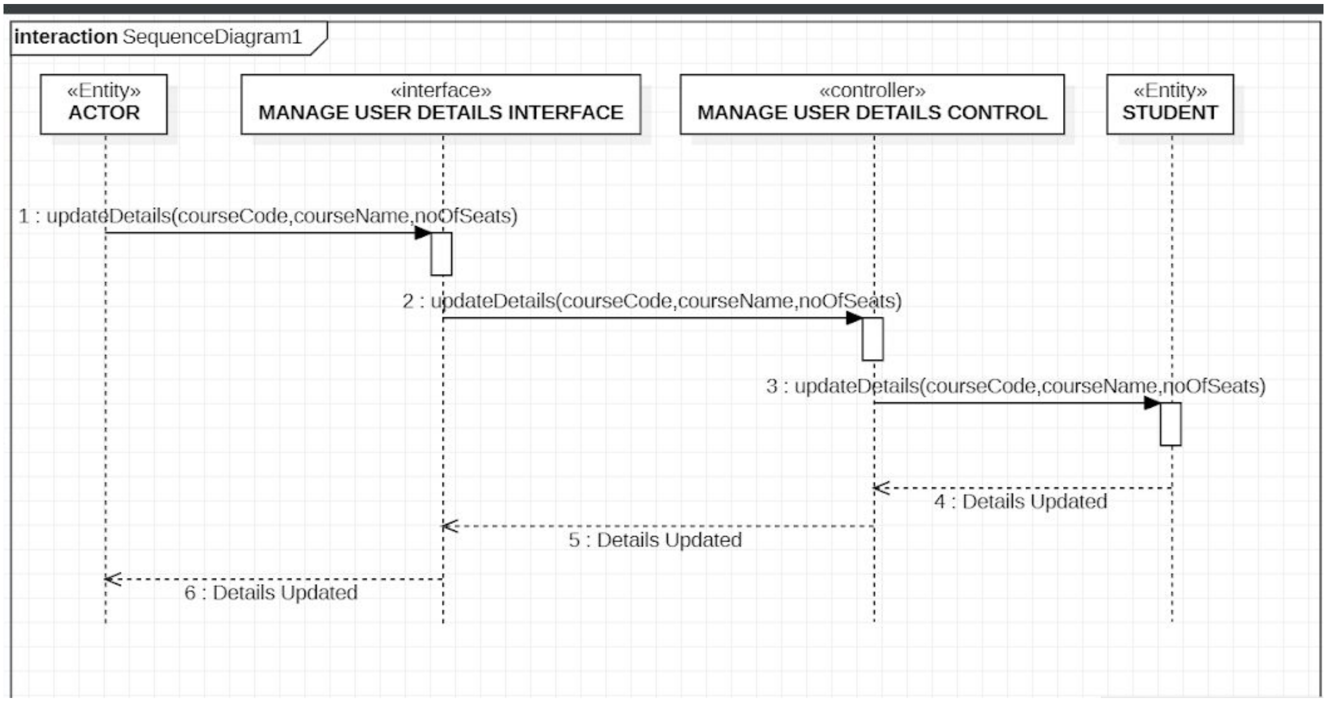
ENTER STUDENT DETAILS ALTERNATIVE SEQUENCE DIAGRAM:


interaction SequenceDiagram1
«entity»
Actor
«interface»
Registration
«controller»
Registration
«entity»
Registration
1 : Enter Userid
2 : Enter firstName
3 : Enter lastName
4 : Enter Department
5 : Enter Phone Number
6 : Enter Password
7 : Register(Uid,FN,LN,Dept,Pno,Pass)
8 : Register(Uid,FN,LN,Dept,Pno,Pass)
9 : Register(Uid,FN,LN,Dept,Pno,Pass)
10 : Invalid Details or Empty Fields
11 : Invalid Details or Empty Fields
12 : Invalid Details or Empty Fields

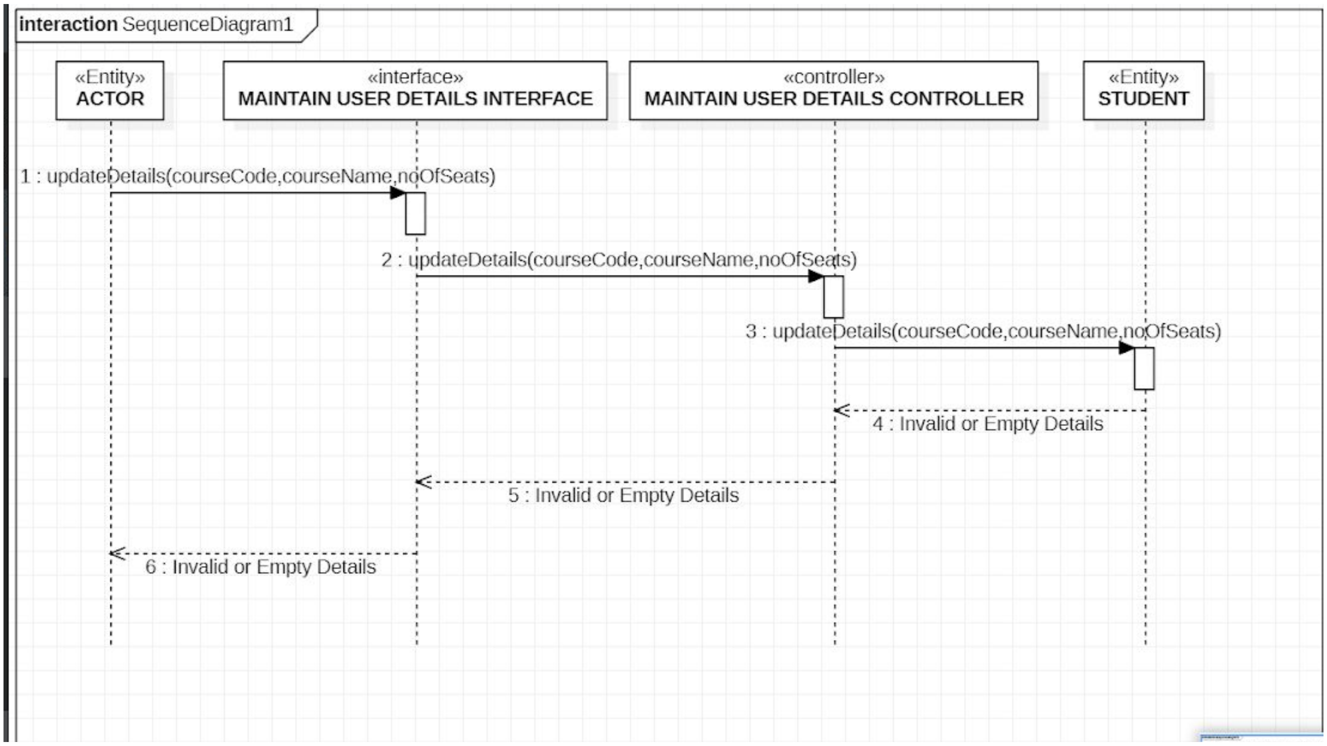
ENTER STUDENT DETAILS SEQUENCE DIAGRAM (ALTERNATIVE FLOW 2: USER EXITS):



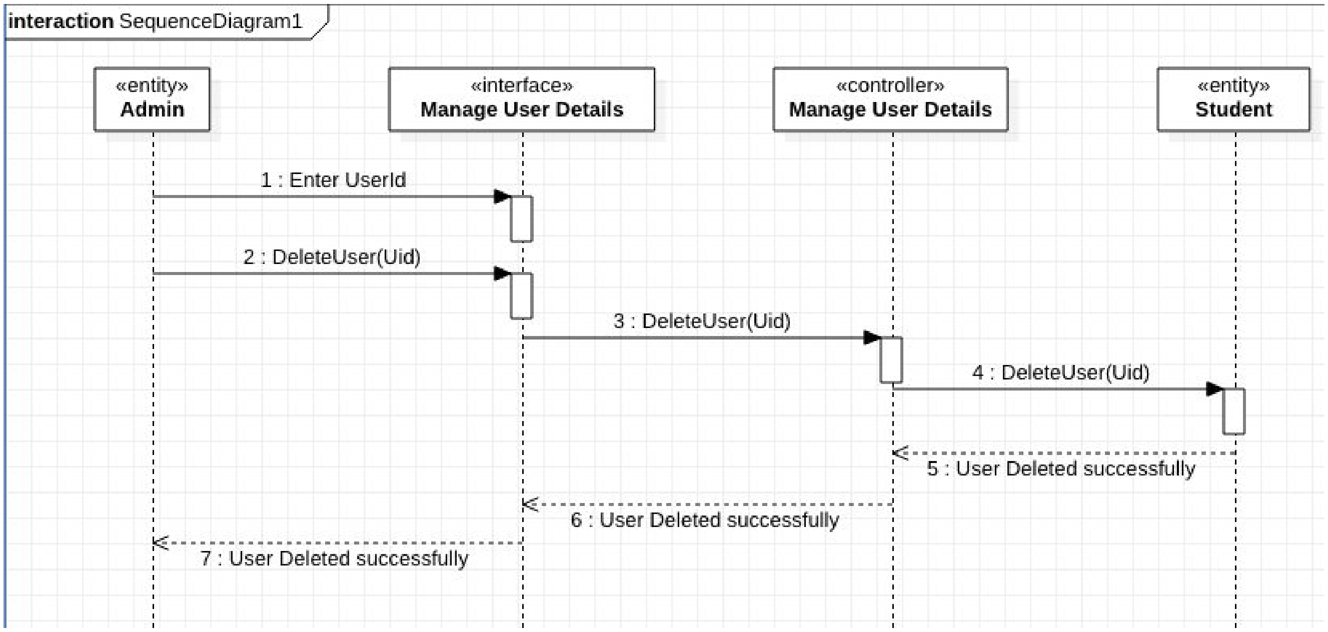
UPDATE USER DETAILS BASIC SEQUENCE DIAGRAM:


interaction SequenceDiagram1
«Entity»
ACTOR
«interface»
MANAGE USER DETAILS INTERFACE
«controller»
MANAGE USER DETAILS CONTROL
«Entity»
STUDENT
1 : updateDetails(courseCode,courseName,noOfSeats)
2 : updateDetails(courseCode,courseName,noOfSeats)
3 : updateDetails(courseCode,courseName,noOfSeats)
4 : Details Updated
5 : Details Updated
6 : Details Updated

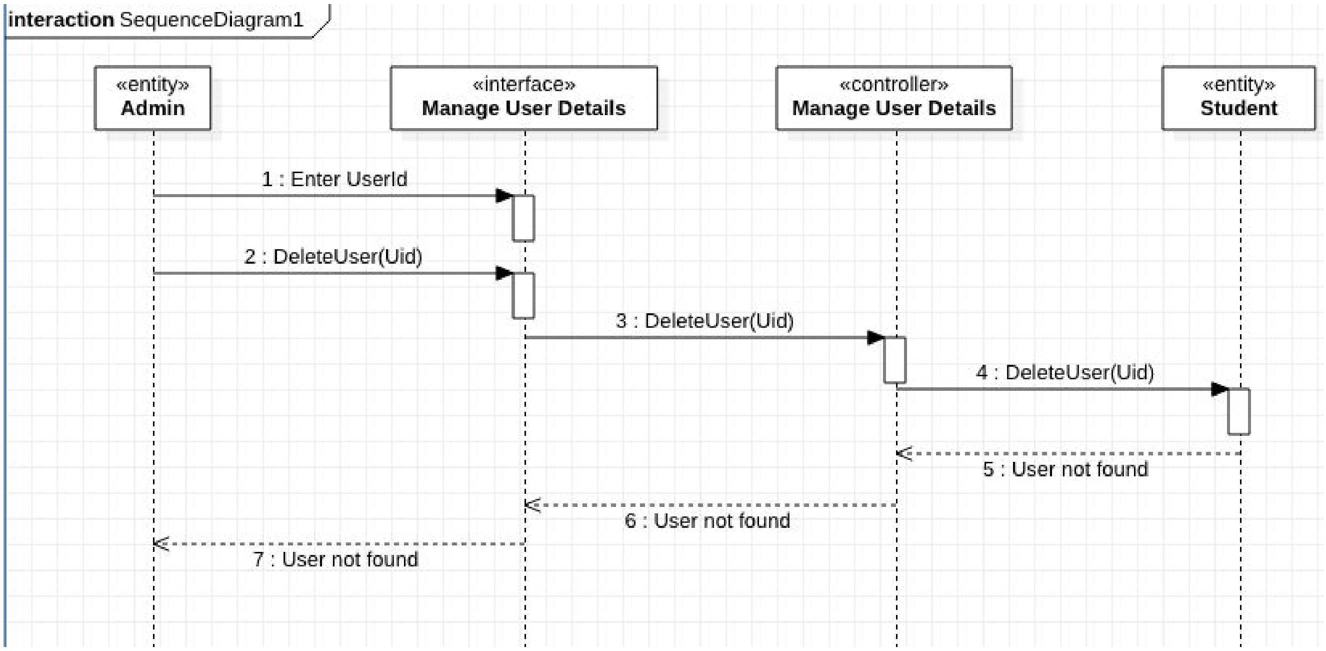
UPDATE USER DETAILS ALTERNATIVE SEQUENCE DIAGRAM:


interaction SequenceDiagram1
«Entity»
ACTOR
«interface»
MAINTAIN USER DETAILS INTERFACE
«controller»
MAINTAIN USER DETAILS CONTROLLER
«Entity»
STUDENT
1 : updateDetails(courseCode,courseName,noOfSeats)
2 : updateDetails(courseCode,courseName,noOfSeats)
3 : updateDetails(courseCode,courseName,noOfSeats)
4 : Invalid or Empty Details
5 : Invalid or Empty Details
6 : Invalid or Empty Details

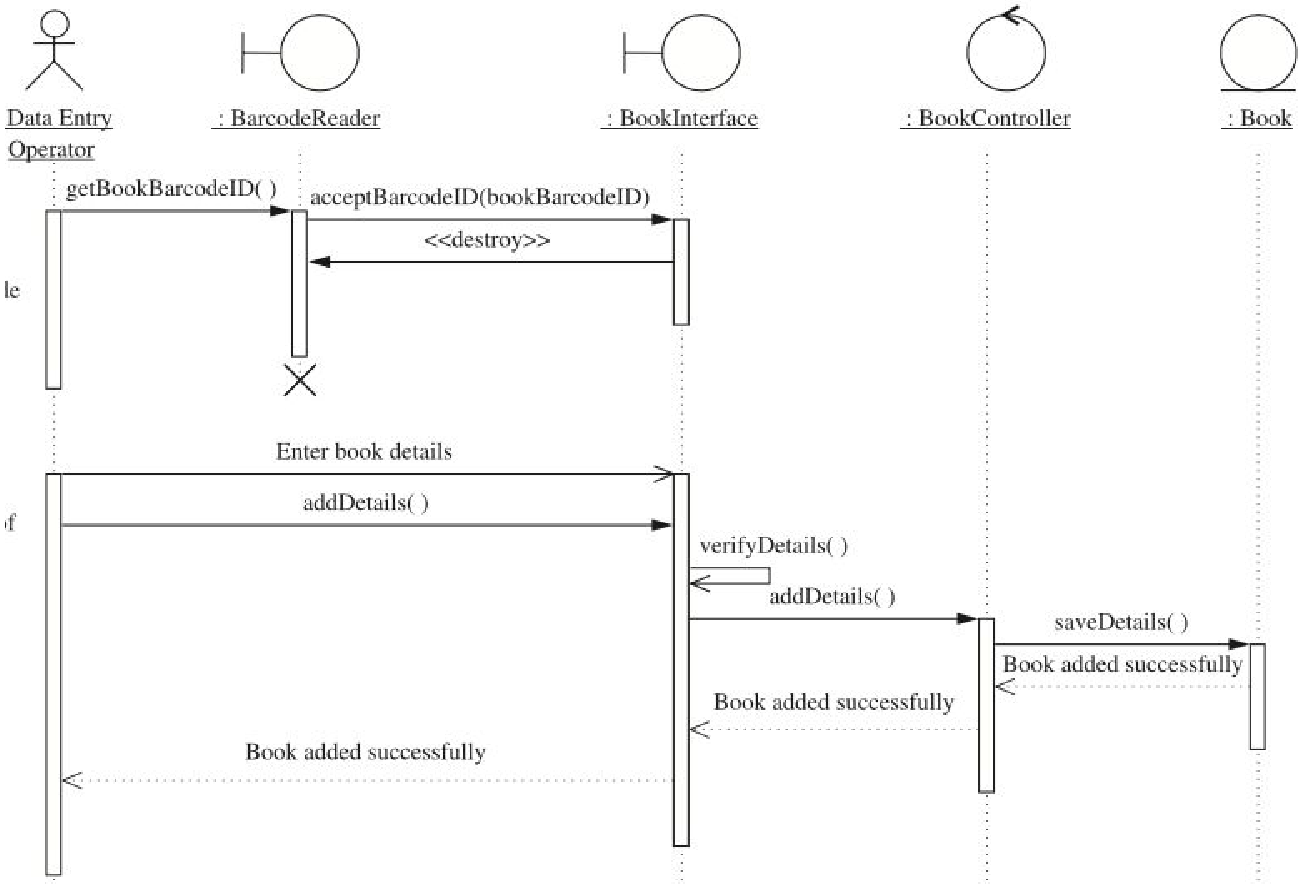
DELETE USER DETAILS BASIC SEQUENCE DIAGRAM:



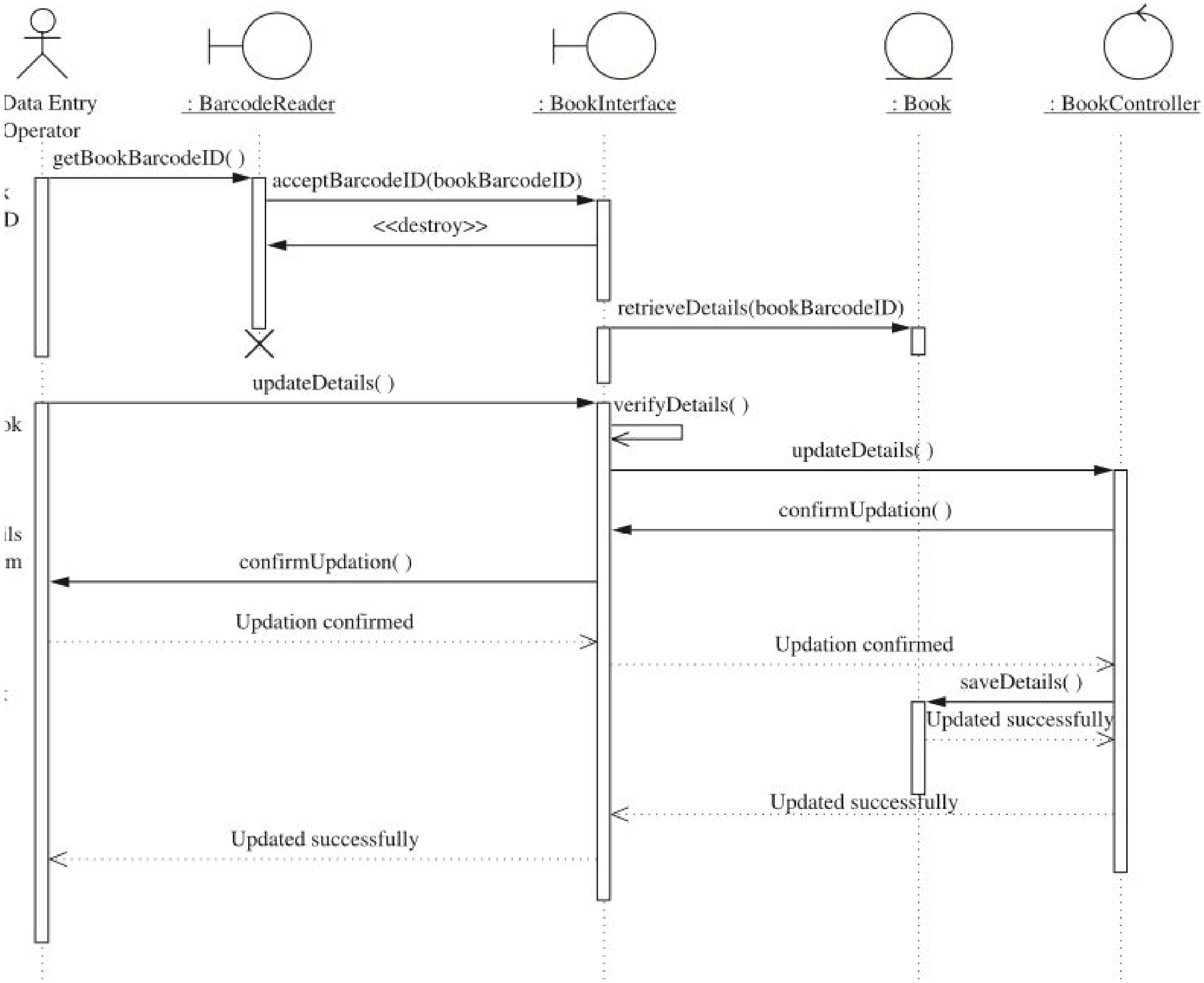
DELETE USER DETAILS ALTERNATIVE SEQUENCE DIAGRAM:



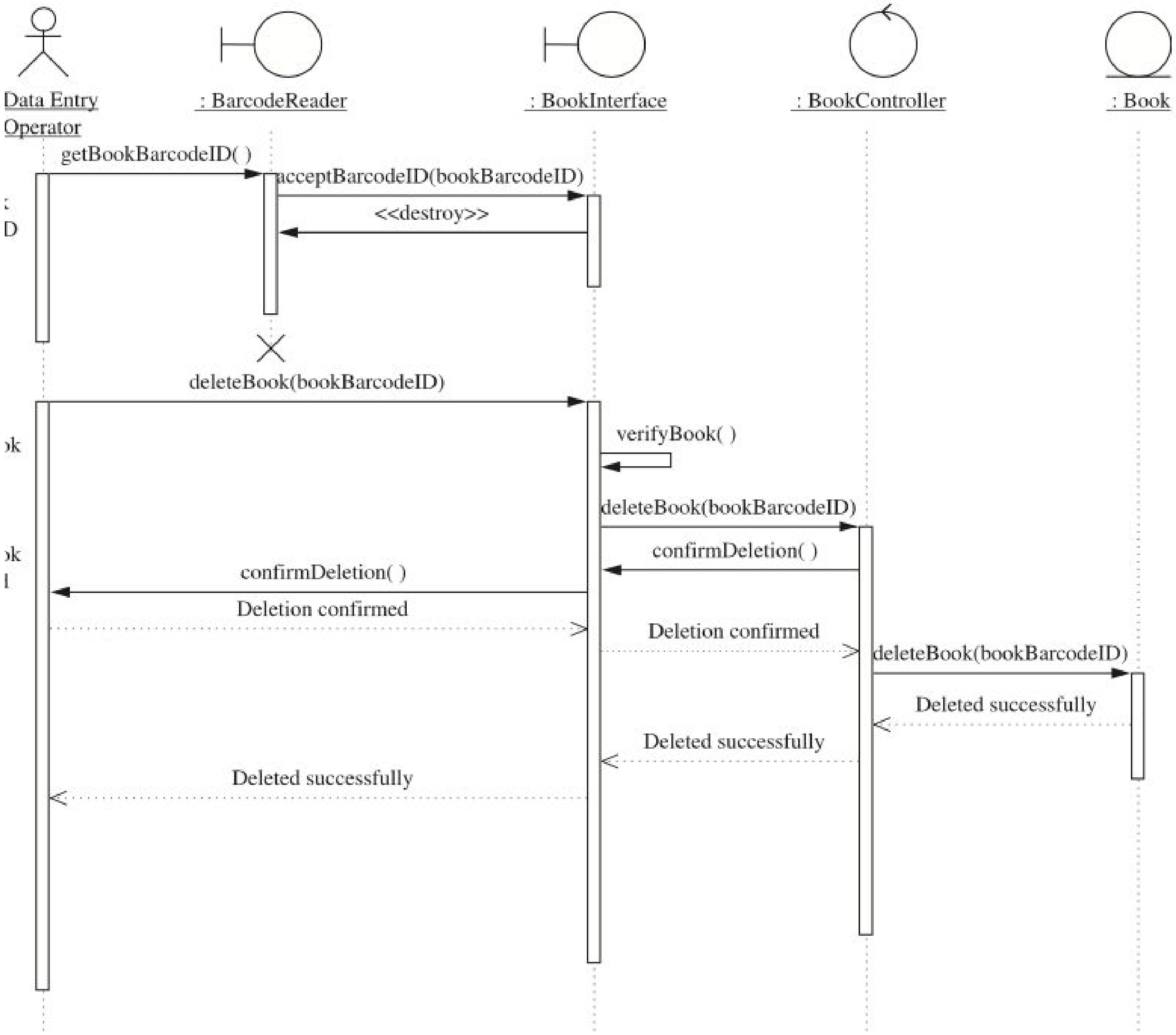
ADD BOOK BASIC SEQUENCE DIAGRAM:



UPDATE BOOK BASIC SEQUENCE DIAGRAM:



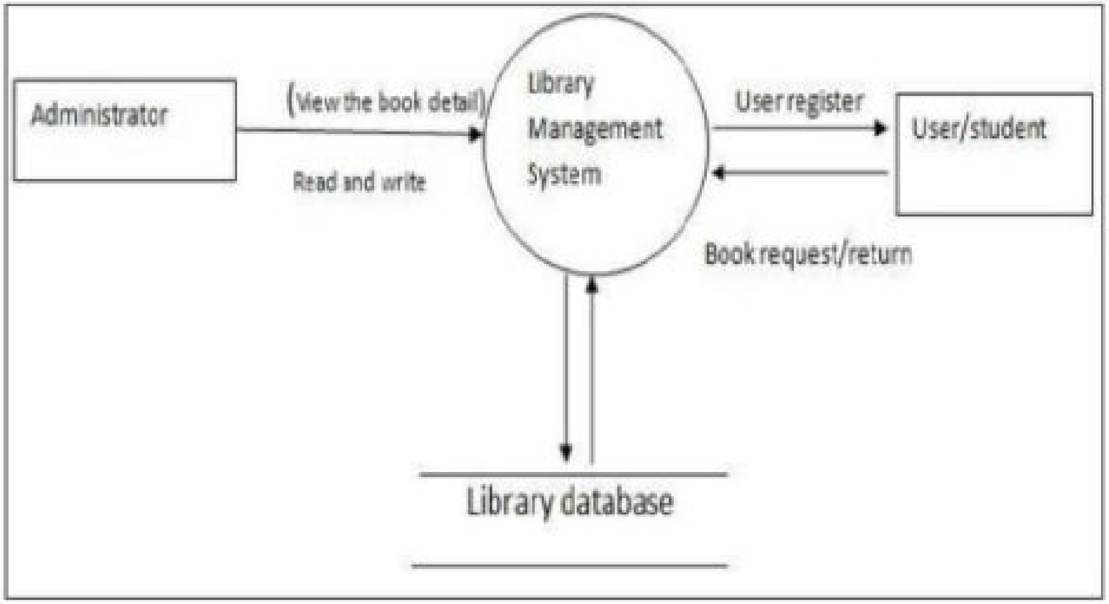
DELETE BOOK BASIC SEQUENCE DIAGRAM:



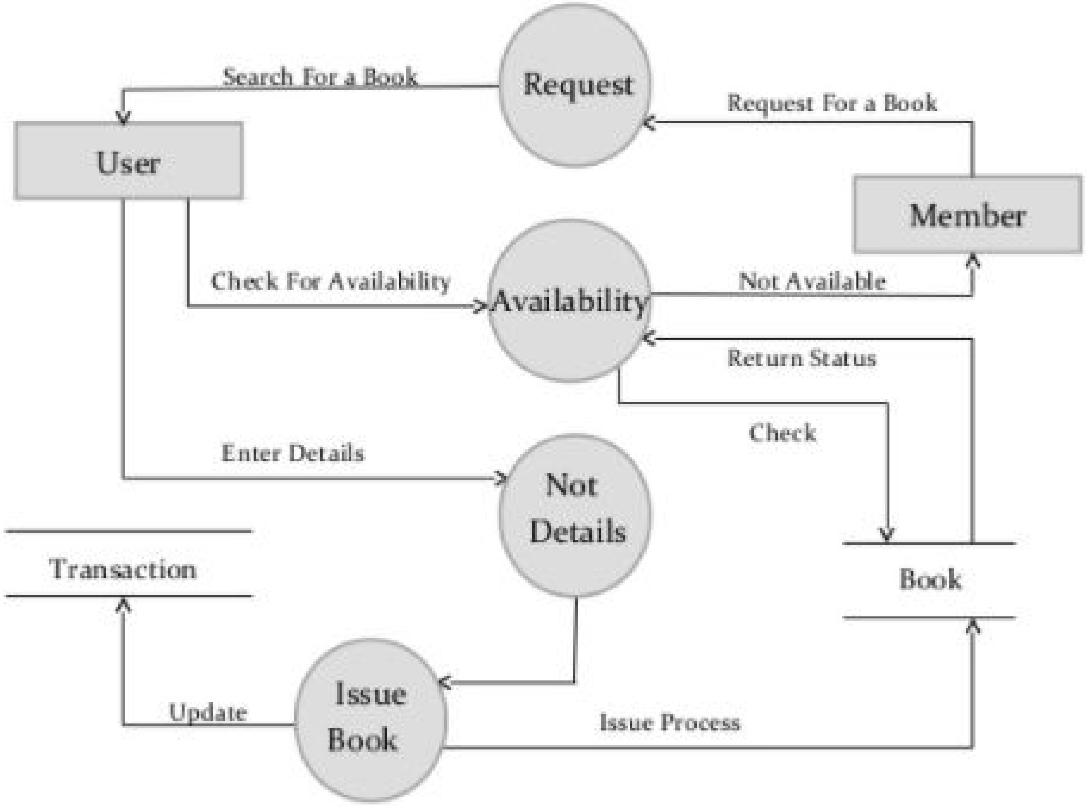
4.4 Data Flow Diagram:

A data-flow diagram is a way of representing a flow of data through a process or a system. The DFD also provides information about the outputs and inputs of each entity and the process itself. A data-flow diagram has no control flow, there are no decision rules and no loops. There are three levels of Data flow diagram. Which are shown below for the LMS.

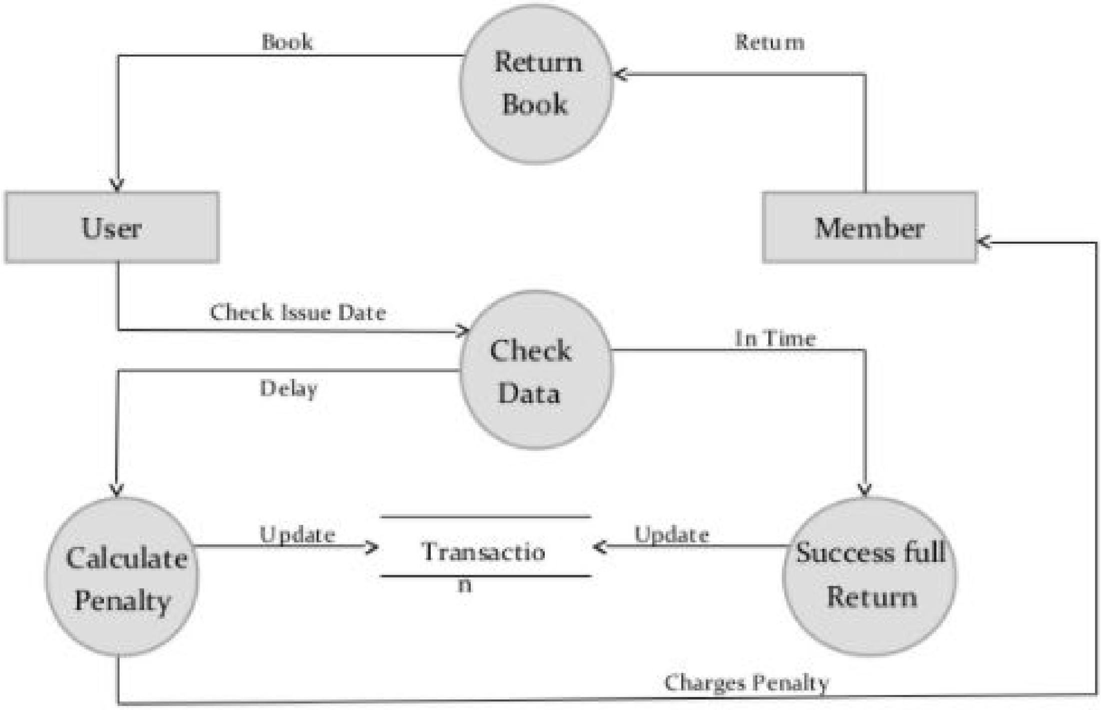
Zero Level DFD:



Level 1 DFD:

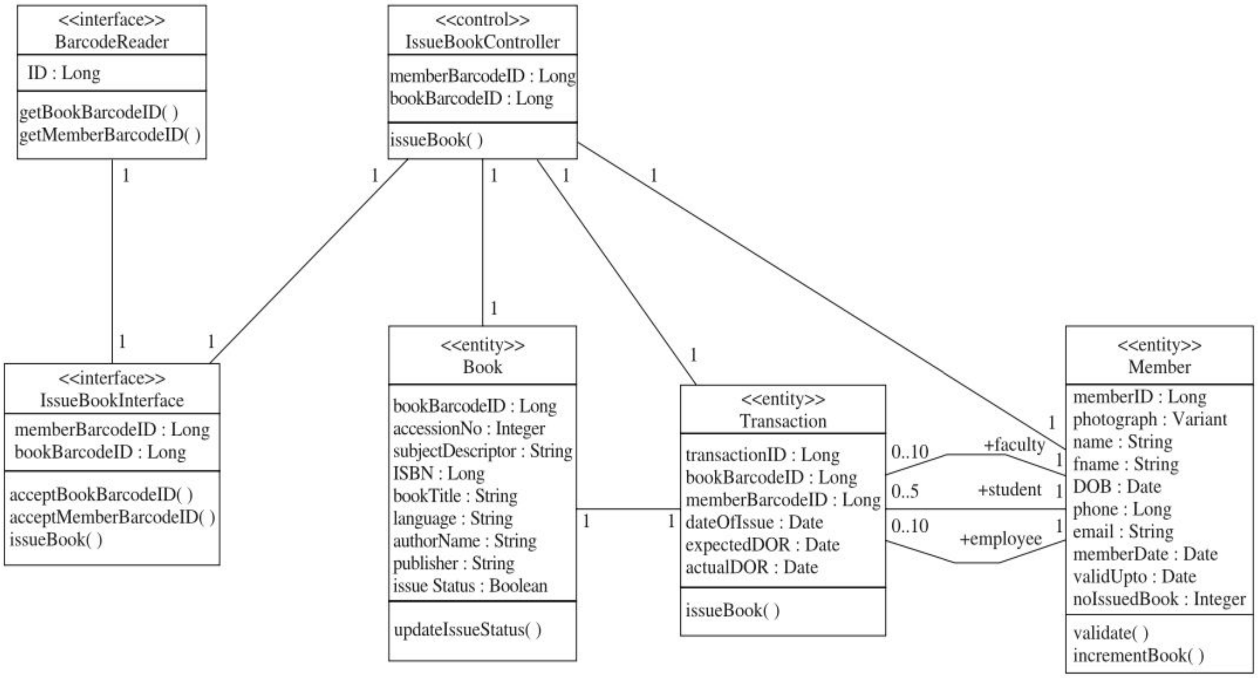


Level 2 DFD:

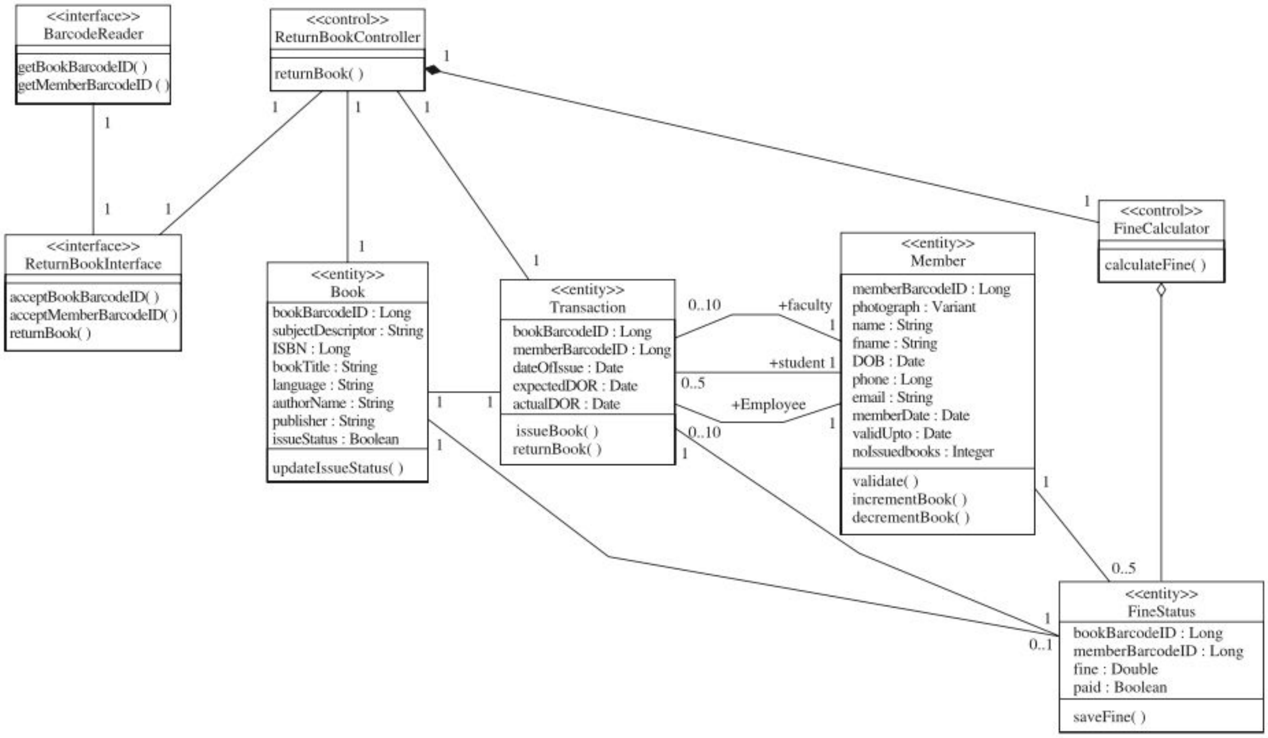


4.5 Class Diagram:

Issue Book Class Diagram:



Return Book Class Diagram:



5. Other Non-Functional Requirements

5.1 Performance Requirements

(a) Internet connectivity must be required in case the system is hosted remote in a WAN.
(b) Should support at least 7 users simultaneously.
(c) Should run on a 500 MHz, 512 MB RAM machine.
(d) Responses should be within 2 seconds.

5.2 Safety Requirements

To ensure that no one of LMS's users loses any data while using LMS (due to a crash or a bug of some kind) the developer team updates LMS regularly. There is a bug tracker available where users can report any bugs they have encountered so that the developers can fix it in the next release. This can be found under
https://github.com/tarunkharkwal98/LMS

5.3 Security Requirements

- LMS has three levels of users each with different privileges i.e.
  Administrator who is the super user and can perform most of the functions in the system..

- Library-Staff who are the employees in the library. They can basically view the task that has been assigned to them and update it once complete.

5.4 Software Quality Attributes

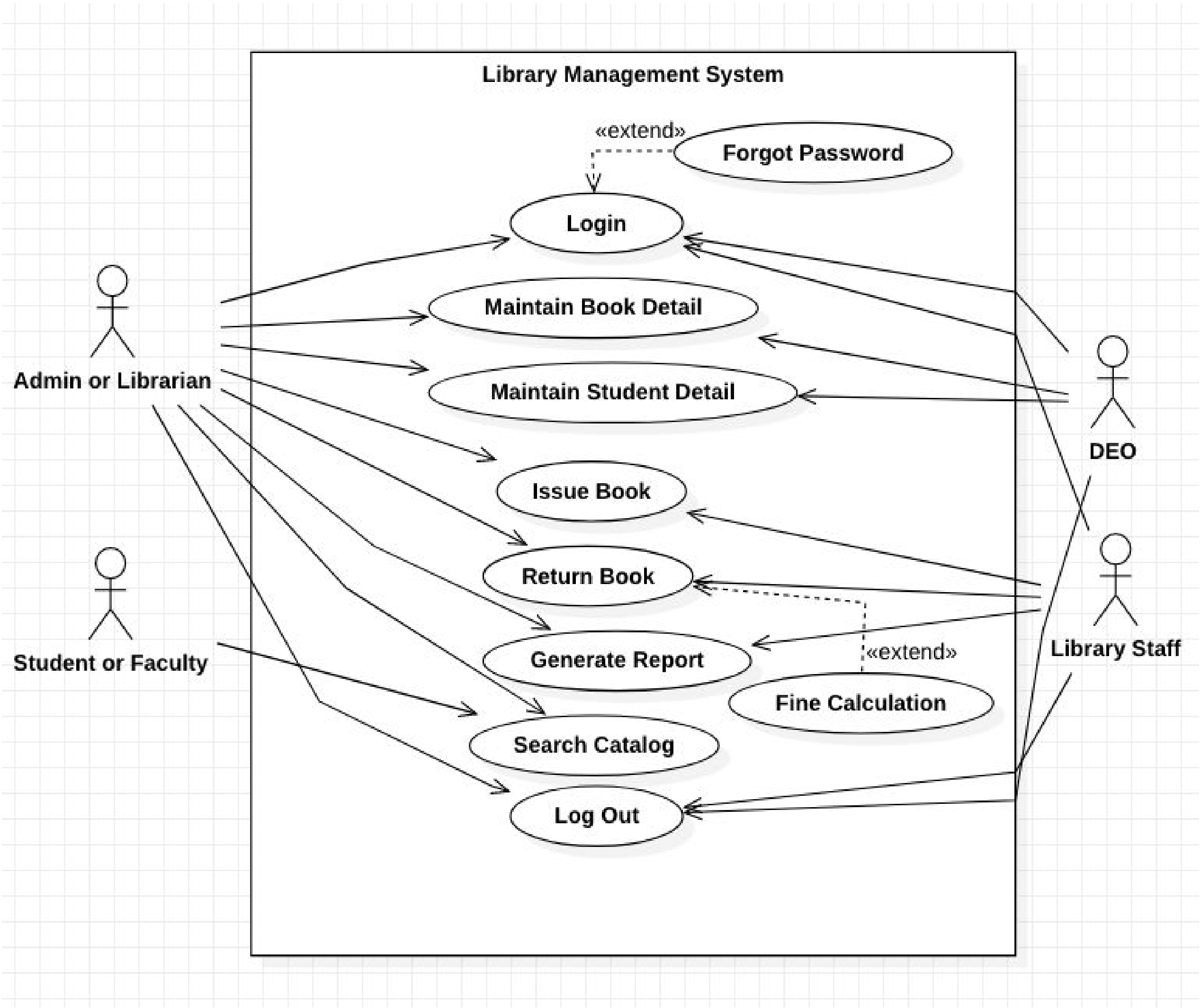
LMS provides the users with both simple and advanced features. Due to its well designed and easy to use interface it can be used by both Advanced and typical users. However, users must already have a basic knowledge of using a computer.

Appendix A: Glossary

SRS- Software Requirements Specification
LMS- Library Management System
LAN- Local Area Network
WAN- Wide Area Network
IEEE- Institute of Electrical and Electronics Engineers
CPU- Central Processing Unit
RAM- Random Access Memory

Appendix B: Analysis Models

Use Case Diagram



Appendix C: Source Code:

In source code there are several files for this project

addCourse.php:

```
<?php
 include "connection.php";
 include "navbar.php";
?>
<!DOCTYPE html>
<html>
<head>
```

```
 <title>Student Registration</title>
 <link rel="stylesheet" type="text/css" href="style.css">
 <meta charset="utf-8">
 <meta name="viewport" content="width=device-width, initial-scale=1">

 <link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/css/bootstrap.min.css">
   <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.3.1/jquery.min.js"></script>
   <script src="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/js/bootstrap.min.js"></script>

 <style type="text/css">
   section
   {
     margin-top: -20px;
     height: 900px;
   }
 </style>
</head>
<body>

<section>
 <div class="reg_img">

   <div class="box2">
         <h1 style="text-align: center; font-size: 35px;font-family: Lucida Console;"> &nbsp &nbsp
&nbsp
       University Registration System</h1>
       <h1 style="text-align: center; font-size: 25px;">ADD COURSE</h1>

     <form name="Registration" action=" " method="post">

       <div class="login">
         <input class="form-control" type="text" name="cid" placeholder="Course Id" required=""><br>
```

```
          <input class="form-control" type="text" name="cname" placeholder="Course Name" required=""> <br>
        <input class="form-control" type="text" name="instructor" placeholder="Instructor Name" required=""> <br>
       <input class="form-control" type="text" name="credits" placeholder="Credits" required=""> <br>

     <input class="btn btn-default" type="submit" name="submit" value="Add Course" style="color: black; width: 100px;
     height: 30px"> </div>
   </form>

  </div>
 </div>
</section>

  <?php

   if(isset($_POST['submit']))
   {
    $count=0;



    $res=mysqli_query($db,"SELECT cid from `courses`");

    while($row=mysqli_fetch_assoc($res))
    {
     if($row['cid']==$_POST['cid'])
     {
      $count=$count+1;
     }
    }
    if($count==0)
    {
            mysqli_query($db,"INSERT INTO `courses` VALUES ('$_POST[cid]','$_POST[cname]',
'$_POST[instructor]',
     '$_POST[credits]')");
    ?>
     <script type="text/javascript">
      alert("Course Added successfully");
     </script>
    <?php
    }
    else
    {

     ?>
       <script type="text/javascript">
        alert("The course is already exist.");
       </script>
     <?php

    }

   }

  ?>

</body>
</html>
```

connection.php:

```
<?php
```

```
  $db=mysqli_connect("localhost","root","","URS");
              /* server name, username, password, database name */
?>
edit.php:
<?php
  include "navbar.php";
  include "connection.php";
?>
<!DOCTYPE html>
<html>
<head>
  <title>edit profile</title>
  <style type="text/css">
     .form-control
     {
        width:250px;
        height: 38px;
     }
     .form1
     {
        margin:0 540px;
     }
     label
     {
        color: white;
     }

  </style>
</head>
<body style="background-color: #004528;">

  <h2 style="text-align: center;color: #fff;">Edit Information</h2>
```

```
<?php

   $sql = "SELECT * FROM admin WHERE username='$_SESSION[login_user]'";
   $result = mysqli_query($db,$sql) or die (mysql_error());

   while ($row = mysqli_fetch_assoc($result))
   {
      $first=$row['first'];
      $last=$row['last'];
      $username=$row['username'];
      $password=$row['password'];
      $email=$row['email'];
      $phone=$row['phone'];
   }

?>

<div class="profile_info" style="text-align: center;">
   <span style="color: white;">Welcome,</span>
   <h4 style="color: white;"><?php echo $_SESSION['login_user']; ?></h4>
</div><br><br>

<div class="form1">
   <form action="" method="post" enctype="multipart/form-data">

   <input class="form-control" type="file" name="file">

   <label><h4><b>First Name: </b></h4></label>
   <input class="form-control" type="text" name="first" value="<?php echo $first; ?>">

   <label><h4><b>Last Name</b></h4></label>
   <input class="form-control" type="text" name="last" value="<?php echo $last; ?>">
```

```
    <label><h4><b>Username</b></h4></label>
    <input class="form-control" type="text" name="username" value="<?php echo $username; ?>">

    <label><h4><b>Password</b></h4></label>
    <input class="form-control" type="text" name="password" value="<?php echo $password; ?>">

    <label><h4><b>Email</b></h4></label>
    <input class="form-control" type="text" name="email" value="<?php echo $email; ?>">

    <label><h4><b>Contact No</b></h4></label>
    <input class="form-control" type="text" name="phone" value="<?php echo $phone; ?>">

    <br>
    <div style="padding-left: 100px;">
      <button class="btn btn-default" type="submit" name="submit">save</button></div>
  </form>
</div>
  <?php

    if(isset($_POST['submit']))
    {
      move_uploaded_file($_FILES['file']['tmp_name'],"images/".$_FILES['file']['name']);

      $first=$_POST['first'];
      $last=$_POST['last'];
      $username=$_POST['username'];
      $password=$_POST['password'];
      $email=$_POST['email'];
      $contact=$_POST['contact'];
      $pic=$_FILES['file']['name'];
```

```
            $sql1= "UPDATE admin SET pic='$pic', first='$first', last='$last', username='$username',
password='$password', email='$email', contact='$contact' WHERE
username='".$_SESSION['login_user']."';";

        if(mysqli_query($db,$sql1))
        {
          ?>
            <script type="text/javascript">
              alert("Saved Successfully.");
              window.location="profile.php";
            </script>
          <?php
        }
      }
  ?>
</body>
</html>
footer.php:
<!DOCTYPE html>
<html>
<head>
  <title></title>

  <meta charset="utf-8">
  <meta name="viewport" content="width=device-width, initial-scale=1">
                                                      <link rel="stylesheet"
href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/4.7.0/css/font-awesome.min.css">

  <style type="text/css">
  footer
  {
    height: 200px;
    width: 1361px;
    background-color: black;
}
    .fa
    {
        margin: 0px 5px;
        padding: 5px;
        font-size: 20px;
        width: 20px;
        height: 20px;
        text-align: center;
        text-decoration: none;
        border-radius: 50%;
    }
    .fa:hover
    {
        opacity: .7;
    }
    .fa-facebook
    {
        background: #3B5998;
        color: white;
    }
    .fa-twitter
    {
        background: #55ACEE;
        color: white;
    }
    .fa-google
    {
        background: #dd4b39;
        color: white;
```

```
      }
      .fa-instagram
      {
         background: #dd3b87;
         color: white;
      }
      .fa-yahoo
      {
         background: #500297;
         color: white;
      }
   </style>

</head>
<body>
<footer style="background-color: black; width: 1440px; ">
   <br>
            <h3 style="color:white;text-align: center;">Contact Us!! &nbsp;Through social media&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;</h3><br>

   <div style="margin:0px 570px;">

      <a href="#" class="fa fa-facebook"></a>
      <a href="#" class="fa fa-twitter"></a>
      <a href="#" class="fa fa-google"></a>
      <a href="#" class="fa fa-instagram"></a>
      <a href="#" class="fa fa-yahoo"></a>
   </div>

   <br>
   <p style="color:white;text-align: center;">
```

```
    <br>
          Email:&nbsp  admin@dtu.ac.in  &nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;
<br><br>
                                   Mobile:&nbsp     &nbsp     +91     7906075904
&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;
  </p>
</footer>
</body>
</html>
```

index.php:

```
<?php
  session_start();
?>
<!DOCTYPE html>
<html>
<head>
  <title>
  University Registration System
  </title>
  <link rel="stylesheet" type="text/css" href="style.css">
  <meta charset="utf-8">
  <meta name="viewport" content="width=device-width, initial-scale=1">

<style type="text/css">
  nav
  {
     float: right;
     word-spacing: 30px;
     padding: 20px;
  }
  nav li
  {
    display: inline-block;
    line-height: 80px;
  }
</style>
</head>


<body>
  <div class="wrapper">
    <header>
    <div class="logo">
      <img src="images/9.png" alt="Image" style="width:120px;height:120px;">
          <h1 style="color: white;">&nbsp;&nbsp;&nbsp;&nbsp;UNIVERSITY REGISTRATION SYSTEM</h1>
    </div>

    <?php
    if(isset($_SESSION['login_user']))
    {
      ?>
        <nav>
          <ul>
            <li><a href="index.php">HOME</a></li>
            <li><a href="profile.php">PROFILE</a></li>
            <li><a href="logout.php">LOGOUT</a></li>
          </ul>
        </nav>
      <?php
    }
    else
    {
```

```
   ?>
            <nav>
               <ul>
                  <li><a href="index.php">HOME</a></li>
                  <li><a href="login.php">LOGIN</a></li>
                  <li><a href="registration.php">SIGN-UP</a></li>
               </ul>
            </nav>
 <?php
 }

 ?>



 </header>
 <section>
 <div class="sec_img">
    <br><br><br>
    <div class="box">
       <br><br><br><br>
       <h1 style="text-align: center; font-size: 35px;">University Website</h1><br><br>
       <h1 style="text-align: center;font-size: 25px;">Open for Registration</h1><br>
                    <h1 style="text-align: center;font-size: 25px;">Last Date of Registration:
30-11-2020</h1><br>
    </div>
 </div>
 </section>



</div>
<?php
 include "footer.php";
```

```
  ?>
</body>
</html>
```

login.php:

```
<?php
 include "connection.php";
 include "navbar.php";
?>

<!DOCTYPE html>
<html>
<head>

 <title>Student Login</title>
 <link rel="stylesheet" type="text/css" href="style.css">
 <meta charset="utf-8">
 <meta name="viewport" content="width=device-width, initial-scale=1">
                                                            <link                    rel="stylesheet"
href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/css/bootstrap.min.css">
  <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.3.1/jquery.min.js"></script>
  <script src="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/js/bootstrap.min.js"></script>
 <style type="text/css">
  section
  {
    margin-top: -20px;
  }
 </style>
</head>
<body>

<section>
```

```
 <div class="log_img">
  <br>
   <div class="box1">
              <h1 style="text-align: center; font-size: 35px;font-family: Lucida Console;">University
Registration System</h1>
       <h1 style="text-align: center; font-size: 25px;">User Login Form</h1><br>
     <form  name="login" action="" method="post">

       <div class="login">
                   <input class="form-control" type="text" name="username" placeholder="Username"
required=""> <br>
                <input class="form-control" type="password" name="password" placeholder="Password"
required=""> <br>
               <input class="btn btn-default" type="submit" name="submit" value="Login" style="color:
black; width: 70px; height: 30px">
       </div>

     <p style="color: white; padding-left: 15px;">
       <br><br>
                  <a style="color:yellow; text-decoration: none;" href="update_password.php">Forgot
password?</a> &nbsp &nbsp &nbsp &nbsp &nbsp
                         New to this website?<a style="color: yellow; text-decoration: none;"
href="registration.php">&nbspSign Up</a>
     </p>
   </form>
   </div>
 </div>
</section>

 <?php

   if(isset($_POST['submit']))
    {
     $count=0;
      $res=mysqli_query($db,"SELECT * FROM `admin` WHERE username='$_POST[username]' &&
password='$_POST[password]';");

     $row= mysqli_fetch_assoc($res);
     $count=mysqli_num_rows($res);

     if($count==0)
     {
       ?>

            <div class="alert alert-danger" style="width: 600px; margin-left: 370px; background-color:
#de1313; color: white">
        <strong>The username and password doesn't match</strong>
       </div>
       <?php
     }
     else
     {
       $_SESSION['login_user'] = $_POST['username'];
       $_SESSION['pic']= $row['pic'];

       ?>
       <script type="text/javascript">
         window.location="profile.php"
       </script>
       <?php
     }
   }

 ?>
```

```
</body>
</html>
```

logout.php:

```
<?php
  session_start();
  if(isset($_SESSION['login_user']))
  {
     unset($_SESSION['login_user']);
  }
  header("location:index.php");
?>
```

Navbar.php:

```
<?php
 session_start();
?>
<!DOCTYPE html>
<html>
<head>
 <title>
 </title>

  <link rel="stylesheet" type="text/css" href="style.css">
  <meta charset="utf-8">
  <meta name="viewport" content="width=device-width, initial-scale=1">
 <link rel="stylesheet" href="bootstrap/css/bootstrap.min.css" >



</head>
<body>


   <nav class="navbar navbar-inverse">
  <div class="container-fluid">
    <div class="navbar-header">
     <a class="navbar-brand active">UNIVERSITY REGISTRATION SYSTEM</a>
    </div>
    <ul class="nav navbar-nav">
     <li><a href="index.php">HOME</a></li>
    </ul>
    <?php
     if(isset($_SESSION['login_user']))
     {

    ?>
       <ul class="nav navbar-nav">
        <li><a href="profile.php">PROFILE</a></li>
        <li><a href="addcourse.php">ADD-COURSE</a></li>
        <li><a href="viewcourse.php">COURSE-LIST</a></li>
        <li><a href="registration.php">REGISTER-STUDENT</a></li>

       </ul>
       <ul class="nav navbar-nav navbar-right">
        <li><a href="">
         <div style="color: white">
          <?php
                          echo "<img class='img-circle profile_img' height=30 width=30
src='images/".$_SESSION['pic']."'>";
           echo " ".$_SESSION['login_user'];
          ?>
         </div>
        </a></li>
                    <li><a href="logout.php"><span class="glyphicon glyphicon-log-out">
LOGOUT</span></a></li>
       </ul>
```

```
          <?php
        }
        else
        {   ?>
          <ul class="nav navbar-nav navbar-right">

                                <li><a href="login.php"><span class="glyphicon glyphicon-log-in"> LOGIN</span></a></li>

                        <li><a href="registration.php"><span class="glyphicon glyphicon-user"> SIGN UP</span></a></li>
          </ul>
            <?php
        }
      ?>

    </div>
  </nav>
  <?php

  ?>
</body>
</html>
```

Profile.php:

```
<?php
  include "connection.php";
  include "navbar.php";
?>
<!DOCTYPE html>
<html>
<head>
  <title>Profile</title>
  <style type="text/css">
    .wrapper
    {
      width: 300px;
      margin: 0 auto;
      color: white;
    }
  </style>
</head>
<body style="background-color: #004528; ">
  <div class="container">
    <form action="" method="post">
      <button class="btn btn-default" style="float: right; width: 70px;"
name="submit1">Edit</button>
    </form>
    <div class="wrapper">
      <?php

        if(isset($_POST['submit1']))
        {
          ?>
            <script type="text/javascript">
              window.location="edit.php"
            </script>
          <?php
        }
        $q=mysqli_query($db,"SELECT * FROM admin where username='$_SESSION[login_user]'
;");
      ?>
      <h2 style="text-align: center;">My Profile</h2>

      <?php
```

```
            $row=mysqli_fetch_assoc($q);

            echo "<div style='text-align: center'>
                <img class='img-circle profile-img' height=110 width=120
src='images/".$_SESSION['pic']."'>
            </div>";
        ?>
        <div style="text-align: center;"> <b>Welcome, </b>
            <h4>
                <?php echo $_SESSION['login_user']; ?>
            </h4>
        </div>
        <?php
            echo "<b>";
            echo "<table class='table table-bordered'>";
                echo "<tr>";
                    echo "<td>";
                        echo "<b> First Name: </b>";
                    echo "</td>";

                    echo "<td>";
                        echo $row['first'];
                    echo "</td>";
                echo "</tr>";

                echo "<tr>";
                    echo "<td>";
                        echo "<b> Last Name: </b>";
                    echo "</td>";
                    echo "<td>";
                        echo $row['last'];
                    echo "</td>";
                echo "</tr>";

                echo "<tr>";
                    echo "<td>";
                        echo "<b> User Name: </b>";
                    echo "</td>";
                    echo "<td>";
                        echo $row['username'];
                    echo "</td>";
                echo "</tr>";

                echo "<tr>";
                    echo "<td>";
                        echo "<b> Password: </b>";
                    echo "</td>";
                    echo "<td>";
                        echo $row['password'];
                    echo "</td>";
                echo "</tr>";

                echo "<tr>";
                    echo "<td>";
                        echo "<b> Email: </b>";
                    echo "</td>";
                    echo "<td>";
                        echo $row['email'];
                    echo "</td>";
                echo "</tr>";

                echo "<tr>";
                    echo "<td>";
                        echo "<b> Contact: </b>";
```

```
            echo "</td>";
            echo "<td>";
              echo $row['phone'];
            echo "</td>";
          echo "</tr>";



        echo "</table>";
        echo "</b>";
      ?>
    </div>
  </div>
</body>
</html>
```

registration.php:

```
<?php
 include "connection.php";
 include "navbar.php";
?>

<!DOCTYPE html>
<html>
<head>

 <title>Student Registration</title>
 <link rel="stylesheet" type="text/css" href="style.css">
 <meta charset="utf-8">
 <meta name="viewport" content="width=device-width, initial-scale=1">

 <link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/css/bootstrap.min.css">
   <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.3.1/jquery.min.js"></script>
   <script src="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/js/bootstrap.min.js"></script>
 <style type="text/css">
   section
   {
     margin-top: -20px;
     height: 900px;
   }
 </style>
</head>
<body>

<section>
 <div class="reg_img">

   <div class="box2">
          <h1 style="text-align: center; font-size: 35px;font-family: Lucida Console;"> &nbsp &nbsp
&nbsp
       University Registration System</h1>
       <h1 style="text-align: center; font-size: 25px;">User Registration Form</h1>

     <form name="Registration" action=" " method="post">

       <div class="login">
         <input class="form-control" type="text" name="id" placeholder="Id Number" required=""><br>
            <input class="form-control" type="text" name="first" placeholder="First Name" required="">
<br>
             <input class="form-control" type="text" name="last" placeholder="Last Name" required="">
<br>
                     <input class="form-control" type="text" name="username" placeholder="Username"
required=""> <br>
                <input class="form-control" type="password" name="password" placeholder="Password"
required=""> <br>
```

```
        <input class="form-control" type="text" name="dob" placeholder="DOB" required=""> <br>
        <input class="form-control" type="text" name="category" placeholder="Category" required="">
<br>
        <input class="form-control" type="text" name="email" placeholder="Email" required=""><br>
                <input class="form-control" type="text" name="phone" placeholder="Phone No"
required=""><br>

          <input class="btn btn-default" type="submit" name="submit" value="Sign Up" style="color:
black; width: 70px;
         height: 30px">
       </div>
     </form>

   </div>
 </div>
</section>

   <?php

     if(isset($_POST['submit']))
     {
      $count=0;
      $res=mysqli_query($db,"SELECT username from `student`");

      while($row=mysqli_fetch_assoc($res))
      {
       if($row['username']==$_POST['username'])
       {
        $count=$count+1;
       }
      }
      if($count==0)
```

```
       {
         mysqli_query($db,"INSERT INTO `student` VALUES ('$_POST[id]','$_POST[first]',
'$_POST[last]',
         '$_POST[username]', '$_POST[password]', '$_POST[dob]','$_POST[category]',
'$_POST[email]',
         '$_POST[phone]', 'p.jpg' )");
       ?>
         <script type="text/javascript">
          alert("Registration successful");
         </script>
       <?php
       }
       else
       {
         ?>
           <script type="text/javascript">
             alert("The username already exist.");
           </script>
         <?php
       }
     }
   ?>
</body>
</html>
```

request.php:

```
<?php
include "connection.php";
 include "navbar.php";
?>
<!DOCTYPE html>
<html>
<head>
```

```
    <title>Add Course</title>
    <meta name="viewport" content="width=device-width, initial-scale=1">
    <style type="text/css">
        .srch
        {
            padding-left: 1000px;
        }
        body {
  font-family: "Lato", sans-serif;
  transition: background-color .5s;
}

.sidenav {
  height: 100%;
  margin-top: 50px;
  width: 0;
  position: fixed;
  z-index: 1;
  top: 0;
  left: 0;
  background-color: #222;
  overflow-x: hidden;
  transition: 0.5s;
  padding-top: 60px;
}

.sidenav a {
  padding: 8px 8px 8px 32px;
  text-decoration: none;
  font-size: 25px;
  color: #818181;
  display: block;
```

```
 transition: 0.3s;
}

.sidenav a:hover {
 color: white;
}

.sidenav .closebtn {
 position: absolute;
 top: 0;
 right: 25px;
 font-size: 36px;
 margin-left: 50px;
}

#main {
 transition: margin-left .5s;
 padding: 16px;
}

@media screen and (max-height: 450px) {
 .sidenav {padding-top: 15px;}
 .sidenav a {font-size: 18px;}
}
.img-circle
{
   margin-left: 20px;
}
.h:hover
{
   color:white;
   width: 300px;
```

```
  height: 50px;
  background-color: #00544c;
}

  </style>

</head>
<body>
<!--_________________sidenav_______________-->

  <div id="mySidenav" class="sidenav">
     <a href="javascript:void(0)" class="closebtn" onclick="closeNav()">&times;</a>

        <div style="color: white; margin-left: 60px; font-size: 20px;">

           <?php
           if(isset($_SESSION['login_user']))

                            {      echo "<img class='img-circle profile_img' height=120 width=120
src='images/".$_SESSION['pic']."'>";
              echo "</br></br>";

              echo "Welcome ".$_SESSION['login_user'];
           }
           ?>
        </div><br><br>

        <div class="h"> <a href="addcourse.php">Add Course</a></div>
        <div class="h"> <a href="selectcourse.php">Select Course</a></div>
        <div class="h"> <a href="viewcourse.php">View Course</a></div>
  </div>
<div id="main">
```

```
  <span style="font-size:30px;cursor:pointer" onclick="openNav()">&#9776; open</span>
<div class="container">

   <script>
   function openNav() {
     document.getElementById("mySidenav").style.width = "300px";
     document.getElementById("main").style.marginLeft = "300px";
     document.body.style.backgroundColor = "rgba(0,0,0,0.4)";
   }

   function closeNav() {
     document.getElementById("mySidenav").style.width = "0";
     document.getElementById("main").style.marginLeft= "0";
     document.body.style.backgroundColor = "white";
   }
   </script>
   <br><br>

   <?php
   if(isset($_SESSION['login_user']))
       {
                                  $q=mysqli_query($db,"SELECT * from selectedcourse where
username='$_SESSION[login_user]' ;");

           if(mysqli_num_rows($q)==0)
           {
               echo "No courses Added";
           }
           else
           {
       echo "<table class='table table-bordered table-hover' >";
           echo "<tr style='background-color: #6db6b9e6;'>";
```

```
        //Table header

        echo "<th>"; echo "Course-Id";  echo "</th>";
        echo "<th>"; echo "Course Name";  echo "</th>";
        echo "<th>"; echo "Instructor Name";  echo "</th>";
        echo "<th>"; echo "Credit Hrs.";  echo "</th>";

      echo "</tr>";

      while($row=mysqli_fetch_assoc($q))
      {
        echo "<tr>";
        echo "<td>"; echo $row['cid']; echo "</td>";
        echo "<td>"; echo $row['cname']; echo "</td>";
        echo "<td>"; echo $row['instructor']; echo "</td>";
        echo "<td>"; echo $row['credits']; echo "</td>";

        echo "</tr>";
      }
    echo "</table>";
      }
    }
    else
    {
      echo "</br></br></br>";
      echo "<h2><b>";
      echo " Please login first to see the request information.";
      echo "</b></h2>";
    }
    ?>
  </div>
</div>
```

```
</body>
</html>
```

selectcourse.php:

```
<?php
 include "connection.php";
 include "navbar.php";
?>
<!DOCTYPE html>
<html>
<head>
	<title>Books</title>
	<meta name="viewport" content="width=device-width, initial-scale=1">

	<style type="text/css">
		.srch
		{
			padding-left: 1000px;
		}
		body {
 font-family: "Lato", sans-serif;
 transition: background-color .5s;
}
.sidenav {
 height: 100%;
 margin-top: 50px;
 width: 0;
 position: fixed;
 z-index: 1;
 top: 0;
 left: 0;
 background-color: #222;
 overflow-x: hidden;
```

```
  transition: 0.5s;
  padding-top: 60px;
}
.sidenav a {
  padding: 8px 8px 8px 32px;
  text-decoration: none;
  font-size: 25px;
  color: #818181;
  display: block;
  transition: 0.3s;
}
.sidenav a:hover {
  color: white;
}
.sidenav .closebtn {
  position: absolute;
  top: 0;
  right: 25px;
  font-size: 36px;
  margin-left: 50px;
}
#main {
  transition: margin-left .5s;
  padding: 16px;
}
@media screen and (max-height: 450px) {
  .sidenav {padding-top: 15px;}
  .sidenav a {font-size: 18px;}
}
.img-circle
{
	margin-left: 20px;
```

```
}
.h:hover
{
	color:white;
	width: 300px;
	height: 50px;
	background-color: #00544c;
}
	</style>
</head>
<body>
<!--__________________________________sidenav______________________________________-->
	<div id="mySidenav" class="sidenav">
  <a href="javascript:void(0)" class="closebtn" onclick="closeNav()">&times;</a>

			<div style="color: white; margin-left: 60px; font-size: 20px;">

		<?php
		if(isset($_SESSION['login_user']))

		{ 	echo 	"<img 	class='img-circle 	profile_img' 	height=120 	width=120
src='images/".$_SESSION['pic']."'>";
		   echo "</br></br>";

		   echo "Welcome ".$_SESSION['login_user'];
		}
		?>
	</div><br><br>


  <div class="h"> <a href="addcourse.php">Add-Course</a></div>
  <div class="h"> <a href="viewcourse.php">View-Course</a></div>
```

```
</div>

<div id="main">

  <span style="font-size:30px;cursor:pointer" onclick="openNav()">&#9776; open</span>



<script>
function openNav() {
  document.getElementById("mySidenav").style.width = "300px";
  document.getElementById("main").style.marginLeft = "300px";
  document.body.style.backgroundColor = "rgba(0,0,0,0.4)";
}

function closeNav() {
  document.getElementById("mySidenav").style.width = "0";
  document.getElementById("main").style.marginLeft= "0";
  document.body.style.backgroundColor = "white";
}
</script>
        <!--____________________search bar_________________________-->

        <div class="srch">
                <form class="navbar-form" method="post" name="form1">

                                <input class="form-control" type="text" name="search"
placeholder="search courses.." required="">
                                <button style="background-color: #6db6b9e6;" type="submit"
name="submit" class="btn btn-default">
                                        <span class="glyphicon glyphicon-search"></span>
                                </button>
                </form>
        </div>
        <!--___________________request book__________________-->
        <div class="srch">
                <form class="navbar-form" method="post" name="form1">

                                <input class="form-control" type="text" name="bid"
placeholder="Enter Course ID" required="">
                                <button style="background-color: #6db6b9e6;" type="submit"
name="submit1" class="btn btn-default">
           Add Course</button>
                </form>
        </div>
        <form name="Registration" action=" " method="post">

      <div class="login" style="width= 40px;">
       <input class="form-control" type="text" name="id" placeholder="Id Number"
required=""><br>
       <input class="form-control" type="text" name="cid" placeholder="Course Id" required="">
<br>
               <input class="form-control" type="text" name="cname" placeholder="Course Name"
required=""> <br>
                <input class="form-control" type="text" name="instructor" placeholder="Instructor"
required=""> <br>
                    <input class="form-control" type="text" name="credits" placeholder="Credits"
required=""><br>

               <input class="btn btn-default" type="submit" name="submit1" value="Add Course"
style="color: black; width: 100px;
       height: 30px">
      </div>
    </form>
    <h2>List Of Courses</h2>
    <?php


        if(isset($_POST['submit']))
        {
            $q=mysqli_query($db,"SELECT * from courses where cname like
'%$_POST[search]%' ");

    if(mysqli_num_rows($q)==0)
            {
                echo "Sorry! No Course found. Try searching again.";
            }
    else
    {
        echo "<table class='table table-bordered table-hover' >";
            echo "<tr style='background-color: #6db6b9e6;'>";
                //Table header
                echo "<th>"; echo "Course-Id";      echo "</th>";
                echo "<th>"; echo "Course-Name";  echo "</th>";
                echo "<th>"; echo "Instructor Name";  echo "</th>";
                echo "<th>"; echo "Credit Hrs.";  echo "</th>";

            echo "</tr>";

            while($row=mysqli_fetch_assoc($q))
            {
                echo "<tr>";
                echo "<td>"; echo $row['cid']; echo "</td>";
                echo "<td>"; echo $row['cname']; echo "</td>";
                echo "<td>"; echo $row['instructor']; echo "</td>";
                echo "<td>"; echo $row['credits']; echo "</td>";
                echo "</tr>";
            }
        echo "</table>";
            }
        }
            /*if button is not pressed.*/
        else
        {
            $res=mysqli_query($db,"SELECT * FROM `courses` ORDER BY
`courses`.`cname` ASC;");

        echo "<table class='table table-bordered table-hover' >";
            echo "<tr style='background-color: #6db6b9e6;'>";
                //Table header
                echo "<th>"; echo "Course-Id";      echo "</th>";
                echo "<th>"; echo "Course-Name";  echo "</th>";
                echo "<th>"; echo "Instructor Name";  echo "</th>";
     echo "<th>"; echo "Credit Hrs.";  echo "</th>";
    echo "</tr>";

            while($row=mysqli_fetch_assoc($res))
            {
                echo "<tr>";
                echo "<td>"; echo $row['cid']; echo "</td>";
                echo "<td>"; echo $row['cname']; echo "</td>";
                echo "<td>"; echo $row['instructor']; echo "</td>";
                echo "<td>"; echo $row['credits']; echo "</td>";
     echo "</tr>";
            }
        echo "</table>";
        }
        if(isset($_POST['submit1']))
        {
            if(isset($_SESSION['login_user']))
            {
            mysqli_query($db,"INSERT INTO selectedcourse
Values('$_SESSION[login_user]' , '$_POST[id]' , '$_POST[cid]',
            '$_POST[cname]' , '$_POST[instructor]' , '$_POST[credits]' );");
                ?>
                    <script type="text/javascript">
                        window.location="request.php"
                    </script>
                <?php
            }
            else
            {
                ?>
                    <script type="text/javascript">
                        alert("You must login to Add a course");
                    </script>
                <?php
            }
           }
        ?>
</div>
</body>
</html>
```

style.css:

```
html, body, div, span, object, iframe,
h1, h2, h3, h4, h5, h6, p, blockquote, pre,
abbr, address, cite, code,
del, dfn, em, img, ins, kbd, q, samp,
small, strong, sub, sup, var,
b, i,
dl, dt, dd, ol, ul, li,
fieldset, form, label, legend,
table, caption, tbody, tfoot, thead, tr, th, td,
article, aside, canvas, details, figcaption, figure,
footer, header, hgroup, menu, nav, section, summary,
time, mark, audio, video {
margin:0;
padding:0;
border:0;
outline:0;
font-size:100%;
vertical-align:baseline;
background:transparent;
}

body {
line-height:1;
}

article,aside,details,figcaption,figure,
footer,header,hgroup,menu,nav,section {
display:block;
}

nav ul {
list-style:none;
}

blockquote, q {
quotes:none;
}
```

```
blockquote:before, blockquote:after,
q:before, q:after {
content:";
content:none;
}

a {
margin:0;
padding:0;
font-size:100%;
vertical-align:baseline;
background:transparent;
}

/* change colours to suit your needs */
ins {
background-color:#ff9;
color:#000;
text-decoration:none;
}

/* change colours to suit your needs */
mark {
background-color:#ff9;
color:#000;
font-style:italic;
font-weight:bold;
}

del {
text-decoration: line-through;
}
```

```
abbr[title], dfn[title] {
border-bottom:1px dotted;
cursor:help;
}

table {
border-collapse:collapse;
border-spacing:0;
}

/* change border colour to suit your needs */
hr {
display:block;
height:1px;
border:0;
border-top:1px solid #cccccc;
margin:1em 0;
padding:0;
}

input, select {
vertical-align:middle;
}

/*------------------------------Main code   INDEX ------------------------------------------------------*/

.wrapper
{
  height: 660px;
  width: 1440px;
  /*background-color: red;*/
```

```
}
header
{
  height: 130px;

  background-color: black;
  padding: 10px;
}
section
{
  height: 550px;

  background-color: grey;
}
footer
{

  background-color: black;
}
.logo
{
  float :left;
  padding-left: 40px;

}
.logo img
{
  padding-left: 120px;
}

li a
{
  color: white;
  text-decoration: none;
}
/*nav
{
  float: right;
  word-spacing: 30px;
  padding: 20px;
}
nav li
{
  display: inline-block;
  line-height: 80px;
}*/

section .sec_img
{
  height: 550px;
  margin-top: 0px;
  background-image: url("images/1.jpg");
}
.box
{
  height: 300px;
  width: 450px;
  background-color: #030002;
  margin: 70px auto;
  opacity: .6;
  color: white;
}
/*------------------------------st_login----------------------------*/
.log_img
```

```
{
  height: 850px;
  margin-top: 0px;
  background-image: url("images/3.jpg");
}
.box1
{
  height: 450px;
  width: 450px;
  background-color: black;
  margin: 70px auto;
  opacity: .7;
  color: white;
  padding: 20px;
}
form .login
{
  margin: auto 50px;
}
input
{
  height: 25px;
  width: 300px;
}
/*-----------------------------registration--------------------*/
.reg_img
{
  height: 1050px;
  margin-top: 0px;
  background-image: url("images/4.png");
}
.box2
```

```
{
  height: 710px;
  width: 450px;
  background-color: black;
  margin: 0px auto;
  opacity: .7;
  color: white;
  padding: 20px;
}
```

viewcourse.php:

```
<?php
 include "connection.php";
 include "navbar.php";
?>
<!DOCTYPE html>
<html>
<head>
  <title>Books</title>
  <meta name="viewport" content="width=device-width, initial-scale=1">
  <style type="text/css">
     .srch
     {
        padding-left: 1000px;
     }
     body {
 font-family: "Lato", sans-serif;
 transition: background-color .5s;
}
#main {
 transition: margin-left .5s;
 padding: 16px;
}
```

```
  </style>
</head>
<body>
  <!--_________________sidenav_______________-->

  <div id="mySidenav" class="sidenav">
      <div style="color: white; margin-left: 60px; font-size: 20px;">
        <?php
        if(isset($_SESSION['login_user']))

        {
          echo "Welcome ".$_SESSION['login_user'];
        }
        ?>
      </div><br><br>
</div>

<div id="main">
  <h2>List Of Courses</h2>
  <?php

    if(isset($_POST['submit']))
    {
      $q=mysqli_query($db,"SELECT * from courses where name like '%$_POST[search]%' ");

      if(mysqli_num_rows($q)==0)
      {
        echo "Sorry! No Course found. Try searching again.";
      }
      else
      {
    echo "<table class='table table-bordered table-hover' >";
    echo "<tr style='background-color: #6db6b9e6;'>";
        //Table header
        echo "<th>"; echo "Course-Id";  echo "</th>";
        echo "<th>"; echo "Course-Name";  echo "</th>";
        echo "<th>"; echo "Instructor Name";  echo "</th>";
        echo "<th>"; echo "Credit Hrs.";  echo "</th>"
    echo "</tr>";

    while($row=mysqli_fetch_assoc($q))
    {
        echo "<tr>";
        echo "<td>"; echo $row['cid']; echo "</td>";
        echo "<td>"; echo $row['cname']; echo "</td>";
        echo "<td>"; echo $row['instructor']; echo "</td>";
        echo "<td>"; echo $row['credits']; echo "</td>";
        echo "</tr>";
    }
echo "</table>";
    }
}
    /*if button is not pressed.*/
else
{
    $res=mysqli_query($db,"SELECT * FROM `courses` ORDER BY `courses`.`cname` ASC;");
echo "<table class='table table-bordered table-hover' >";
    echo "<tr style='background-color: #6db6b9e6;'>";
        //Table header
        echo "<th>"; echo "Course-Id";  echo "</th>";
        echo "<th>"; echo "Course-Name";  echo "</th>";
        echo "<th>"; echo "Instructor Name";  echo "</th>";
        echo "<th>"; echo "Credit Hrs.";  echo "</th>";
    echo "</tr>";
```

```
      while($row=mysqli_fetch_assoc($res))
      {
        echo "<tr>";
        echo "<td>"; echo $row['cid']; echo "</td>";
        echo "<td>"; echo $row['cname']; echo "</td>";
        echo "<td>"; echo $row['instructor']; echo "</td>";
        echo "<td>"; echo $row['credits']; echo "</td>";
        echo "</tr>";
      }
    echo "</table>";
    }
    if(isset($_POST['submit1']))
    {
      if(isset($_SESSION['login_user']))
      {
  mysqli_query($db,"INSERT INTO selectedcourse Values( '$_POST[id]' , '$_SESSION[login_user]'
, '$_POST[cid]' ,
        '$_POST[cname]' , '$_POST[instructor]' , '$_POST[credits]' );");
        ?>
          <script type="text/javascript">
            window.location="request.php"
          </script>
        <?php
      }
      else
      {
        ?>
          <script type="text/javascript">
            alert("You must login to Add a course");
          </script>
        <?php
      }
```

```
        }
    ?>
</div>
</body>
</html>
```