**Project Log Book**

**Group Members:**

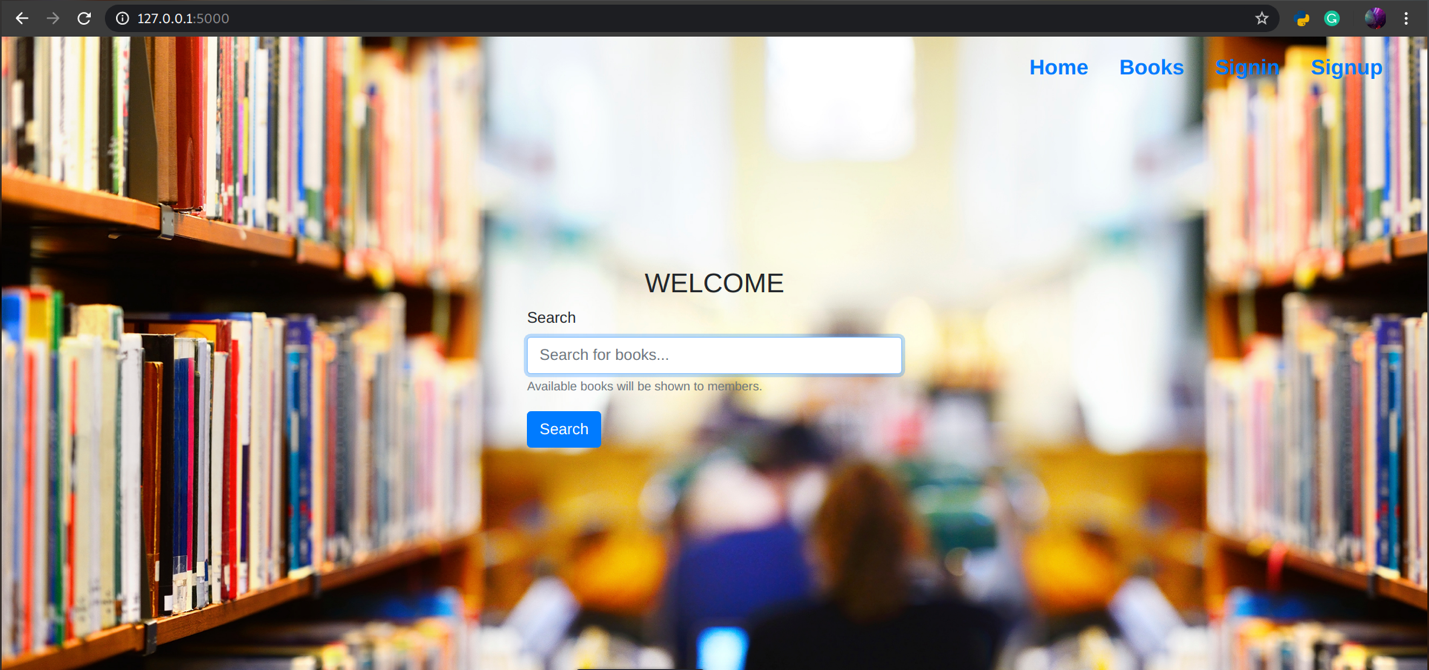
|  |  |
| --- | --- |
| **Entry Date** | **Work Done** |
| September 7th, 2000 | Discussed the basic plan to build the prototype for LMS in class, noting down all constraints to be taken care of. Furthermore, we decided our next group meeting would be on September 15th, 2000 (Friday) at 5:30, meeting place: Zaida Morales' House. |
| September 15th, 2000 | Meeting at Zaida's Place: We discussed about the project objective. Using the Software Management Plan template printed from the web site, we stepped through each section and discussed what was required and what resources were available to us. We also discussed how this prototype should be flexible for other countries. There was constant reference to the "Chinese Railway Passenger Reservation System" and other related articles. |
| September 16th, 2000 | Finished a rough draft prototype and set it up on the online account. |
| September 19th, 2000 | Zaida M. Morales checked the document of the Software Project Management Plan, and she made some correction marking the corrections in red. |
| September 20th, 2000 | The mistakes were corrected on the web site, and email was sent to Zaida M. morales to check the document for any more mistakes |
| September 20th, 2000 | The document was checked by Zaida M. Morales and few more mistakes were found. These mistakes were corrected and put on the web. |
| September 22th, 2000 | Meeting at Zaida's Place: We discussed the Reservation System in more detail and added more information to the SPMP document. |
| September 25th, 2000 | Zaida M. Morales checked the document of the Software Project Management Plan, and she made some corrections. |
| September 27th, 2000 | The mistakes were corrected on the web site, and email was sent to Zaida M. morales to check the document for any more mistakes. |
| September 29th, 2000 | Meeting at Zaida's Place: We discussed parts 4 and 5 of the Software Project Management Plan in more detail and decided to update some information in the SPMP document. The different parts of the document were divided between the team for updates. |
| October 3th, 2000 | Finished updating the rough draft prototype and set it up on the online account. Sent all team members email with link to latest copy of the document. |
| October 4th, 2000 | Zaida M. Morales checked the document of the Software Project Management Plan. The mistakes were corrected on the web site. The latest version of the document is available online. |

***Last Updated on Novemeber 7th, 2000***

**Software Requirements Specification**

*for*

***Library Management System***



***Saad Ahmed (62499)***

***Osama Awan (62500)***

***Huzaifa Tariq (9771)***

***Moiz Sheikh (9805)***

|  |  |  |
| --- | --- | --- |
| Version | Changes Made | Date |
| 1.0 | [First Pass for Review](http://www.geocities.com/cs5391/SRS1.htm) | 10/24/2020 |
| 1.2 | [Second Pass for Review](http://www.geocities.com/cs5391/SRS2.htm) | 11/07/2020 |
| 1.3 | [Third Pass for Review](http://www.geocities.com/cs5391/SRS3.htm) | 11/28/2020 |
| 1.4 | LMS Review Version | 12/04/2020 |
|  | | |

**Table of Contents**

1. Introduction

2. The General Description

3. Specific Requirements

4. Supporting Information

**1. Introduction**

**1.1 Purpose**

This document describes the software requirements for the Automated Railroad Reservation System built for the Library Management System (LMS).

**1.2  Scope In**

The LMS is requesting proposals to build a prototype of a Library Management System for their current system.

Libraries are essential in a process of giving citizens access to knowledge. In digital times they are needed more than ever before. In our country, library systems are written that's why we make a great use of paper.

The objectives of this development effort are:

1. Managing Books
2. Managing Users
3. Inserting Books
4. The software is for the automation of library work. It provides the facilities to Librarian that he/she can enter details related to a particular book and can provide membership to members and update as required

**1.3  Scope Out**

The following features will not be the part of this Project:

**1.3 Definitions, Acronyms, and Abbreviations.**

APPM – AsiaPac Marketing Manager

CASE – Computer Aided Software Engineering

PP - Project Plan

SDD - Software Design Description

SRS - Software Requirement Specification

SDS – Software Design Specification

SPMP - Software Project Management Plan

GUI – Graphical User Interface

QAM – Quality Assurance Manager

PDM – Project Development Manager

PMP – Project Management Professional

TBD – To be determined

UML – Unified Modeling Language

**1.4 References**

       <https://www.sciencedirect.com/topics/computer-science/library-management-system>

       <https://drive.google.com/open?id=1mMYQ8Bfr7Xg523meMQgwPnxDxib0S74o>

       <https://getbootstrap.com/>

**1.5 Overview**

Chapter 2 of the SRS is a brief description of the characteristics of the software to be built, its functions, its users, its constraints and its dependencies.

Chapter 3 is about specific requirements, such as functional requirements, external interface requirements, performance requirements, and also design constraints and quality characteristics.

Finally, chapter 4 includes all the supporting information, such as the Table of Contents, the Appendices, and the Index.

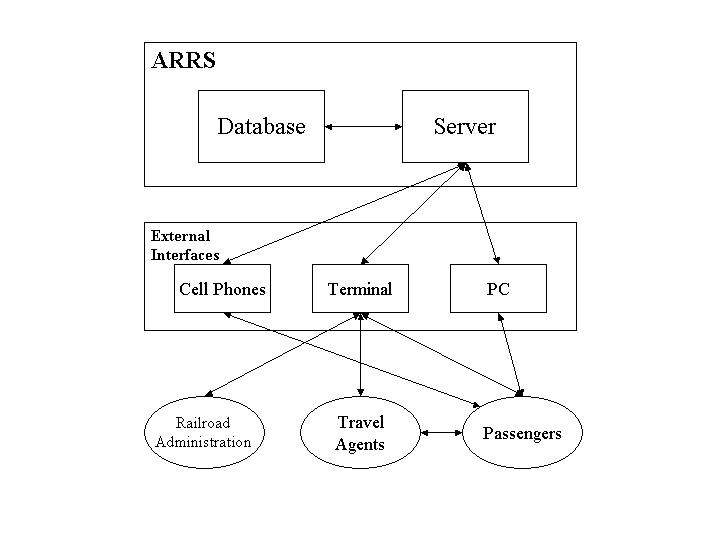
**2. The General Description**

This section describes the general factors that affect the product and its requirements. This section consists of five subsections that follow. This section does not state specific requirements. Each of the subsections makes those requirements easier to understand, it does not specify design or express specific requirements. Such detail is provided in section 3.

**2.1 Product Perspective**

The Automated Library Management System diagram showing the overview of the system’s modules and the relationship of the system to external interfaces is presented in Figure 2.1.

## Figure 2.1 Overview/Architecture Diagram of the LMS



**Functions of System Components:**

Database:

       Stores data

       Creates reports

       Provides access to data

       Updates information

Server:

       Provides access to the database

       Authenticates users

       Processes reservations

       Performs backups

       Produces reports

**External Interfaces:**

User Interface

       Users signup to the website to access the LMS.

       Users can search for books and can borrow the book from LMS.

Admin Interface

       Admin can manage the books in the LMS.

       Admin can manage manage the users in the LMS.

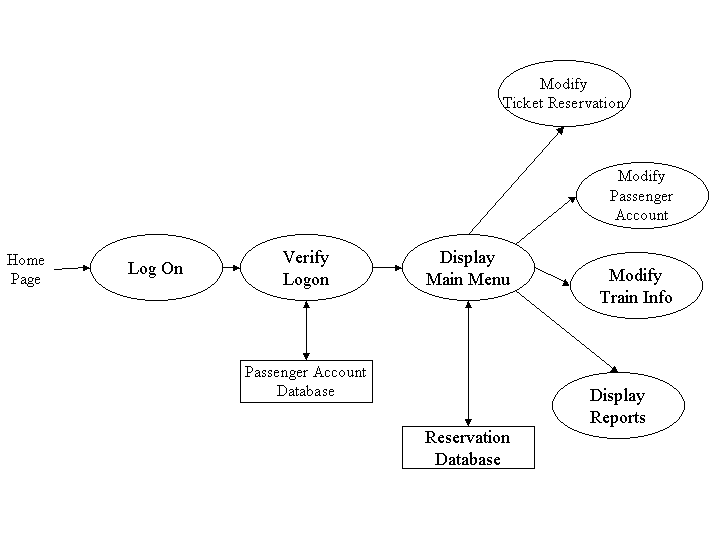
Computer Hardware and Peripheral Equipment to be used:

       Laptop

**2.2 Product Functions**

This section provides a summary of the functions that the software will perform.

**2.2.1 Function Relationships**

Figure 2.2 to 2.6 depict the relationships among the functions to be implemented by the system.  
  
Figure 2.2 LMS General Function Relationship/**Higher Level Usecase** Diagram

Module 3

Module 1

Module 2

Module 4

**2.2.2 Function Descriptions (Functional Requirement Listings)**

**2.2.2.1 Log in Function**

***Description:*** This function ensures that only authorized users gain access to the Reservation databases. An authorized user is a user who has an account on the system. Users include passengers, train officials, and LMS ministry officials. The user must type a valid username and password to gain access.

**2.2.2 Module 1:**

***Description:*** This function allows the user to Register and authenticate the user in sign in process that user enter the valid login credential or not.

**2.2.3 Module 2:**

***Description:*** This function allows users to search the books from the LMS and shows only available books to user which he/she can borrow from the LMS.

**2.2.4 Module 3:**

***Description:*** This function allows librarian to manage the registered users in the LMS. He can get all the information of any member who borrow or return the book from the LMS.

**2.2.5 Module 4:**

***Description:*** This function allows Librarian to add remove and update the books in the LMS. Librarian can add/edit the authors and publisher information.

**2.2.6 Module 5:**

***Description:*** This function

**2.3 User Characteristics**

The main users of the system will be the users borrows the books from the LMS, the Librarian who can manage the books in the LMS. He can edit the authors and publisher information in the books. Librarian also can manage the users who registered in the LMS he can get all the information of any users who borrow or return the book from the LMS.

**2.4 General Constraints**

The constraints for the project are:

* The functional prototype should be available after 30 days upon the arrival of the management team to Pakistan. This may prove to be a serious time constraint on the development of a successful prototype.
* Communication with the team members may prove to be difficult since some developers do not speak English and the management team does not speak Chinese. Even with the presence of a translator, communication may be difficult. Absence of the translator may severely affect project development.
* Team members are restricted from bringing their own equipment, and insufficient equipment supply may hinder project development.
* Team members are restricted to bringing only the analysts of the team to China. This might affect the project development if more people are needed or the required skills are not available.

**2.5 Assumptions and Dependencies** **or Business Logic**

The assumptions for the project are:

* The software is for the automation of library work
* It provides the facilities to Librarian that he/she can enter details related to a particular book and can provide membership to members and update as required.
* On the other hand; Member Can request for issue, return book and check for any existing fine by asking the librarian and can pay his fine and the details are updated accordingly by the librarian.

## <ADD OOAD REPORT DIAGRAMS HERE>

**3. Specific Requirements**

This section of the SRS contains design requirements for the Automated Railway Reservation System.

**3.1 Functional Requirements**

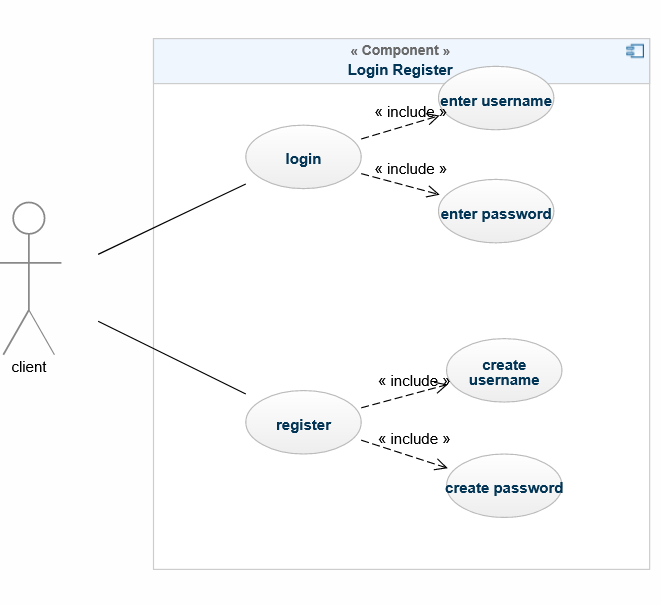
**Module 1/2 complete CRUD Make a Register/Login Function**

1. ***Description:*** This function ensures that only authorized users gain access to the Reservation databases. An authorized user is a user who has an account on the system. The user must type a valid username and password to gain access.

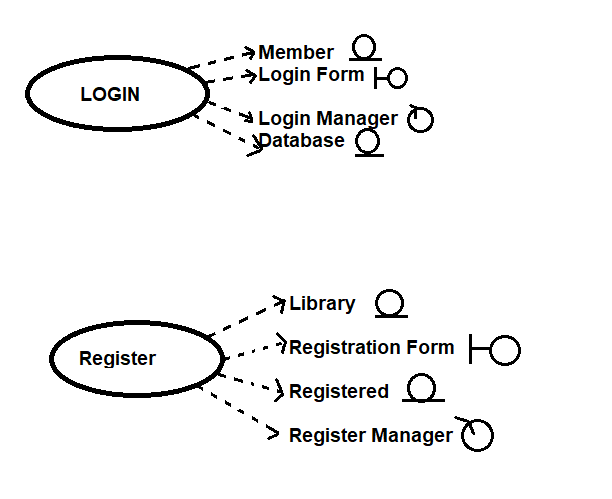
1. ***Usage Scenario/ Use case Description/******Specification:***

|  |  |
| --- | --- |
| Description | Allows access to LMS |
| Inputs | Username, password |
| Source | 1. User inputs username and password 2. Press Login Button |
| Alternate case |  |
| Outputs | Successful login; unsuccessful login |
| Destination | None |
| Precondition | Authorized User |
| Post Condition | No change to Passenger Accounts Database |
| Side Effects | Failures and successful logins are sent to Reservation Database |

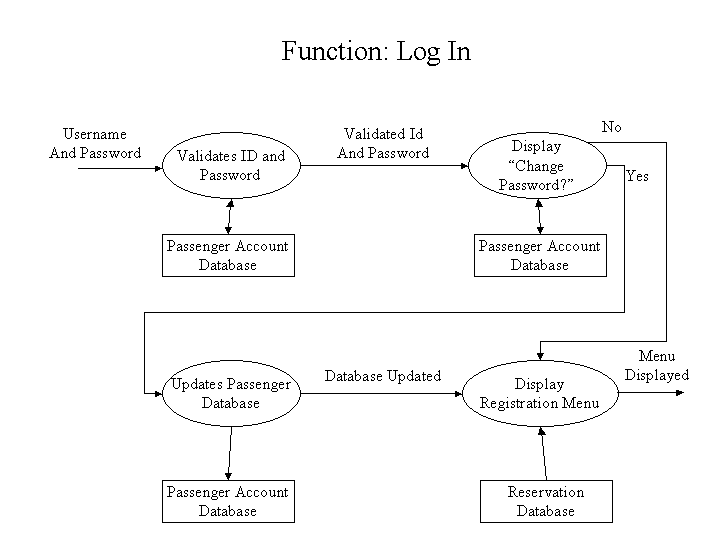
1. ***Detailed Use case Diagram for Register/Login: optional***



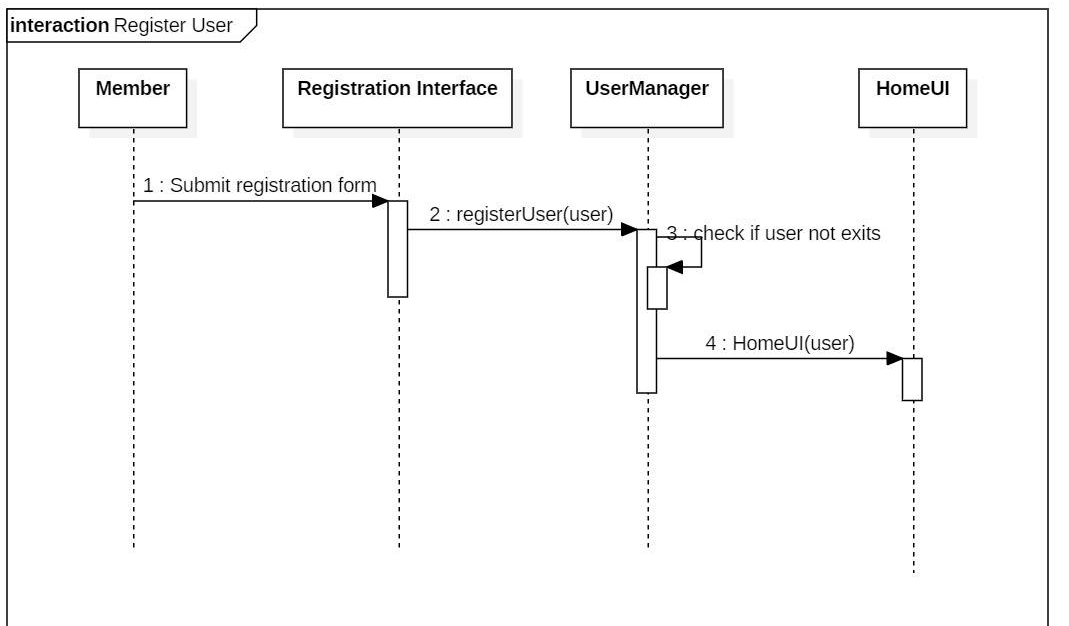
1. ***Use case Realization for Register/Login: optional***



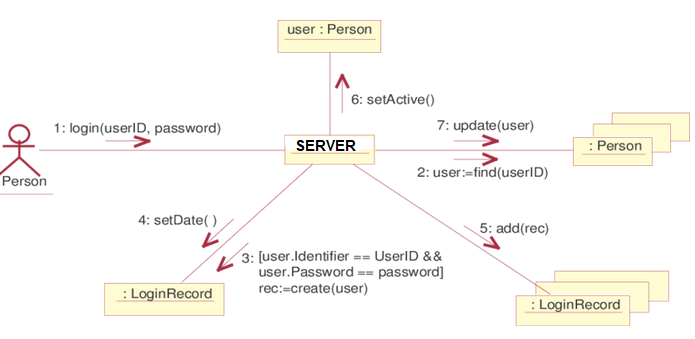
1. ***Flow of Event or Data Flow Diagram for Register/Login: optional***



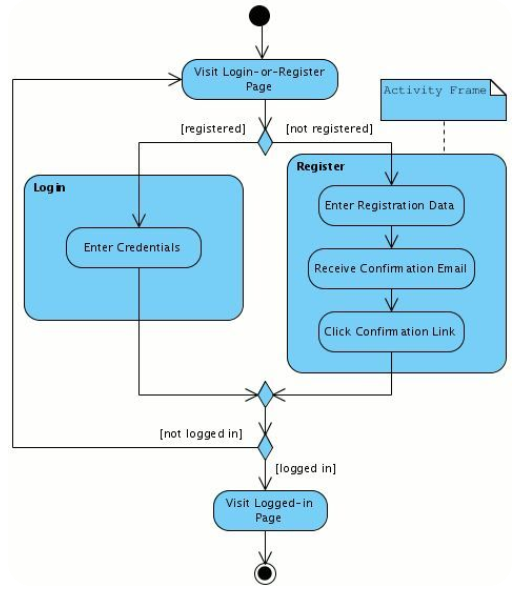
1. ***Sequence Diagram for Register/Login: optional***



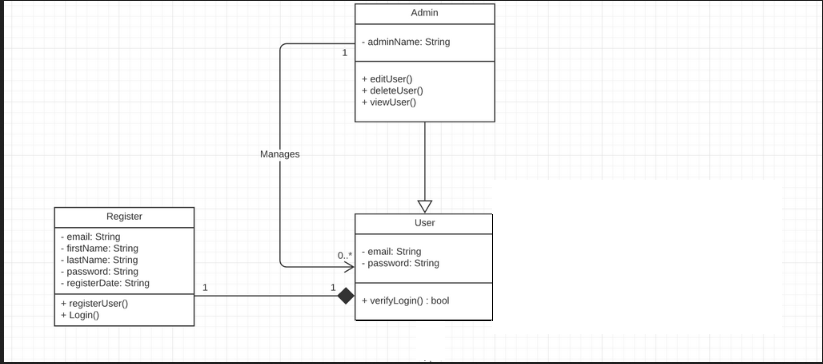
1. ***Collaboration Diagram for Register/Login: optional***



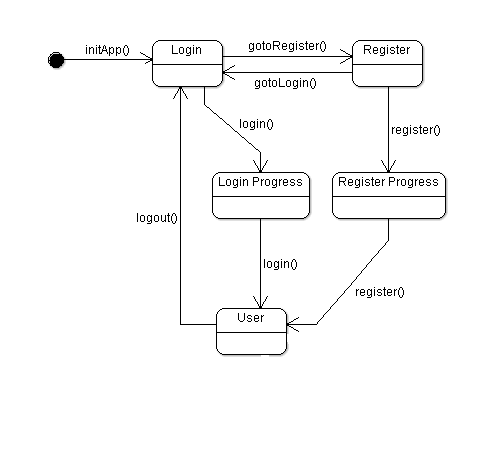
1. ***Activity Diagram for Register/Login: optional***



1. ***Class Diagram for Register/Login: optional***



1. ***State Chart Diagram for Register/Login: optional***

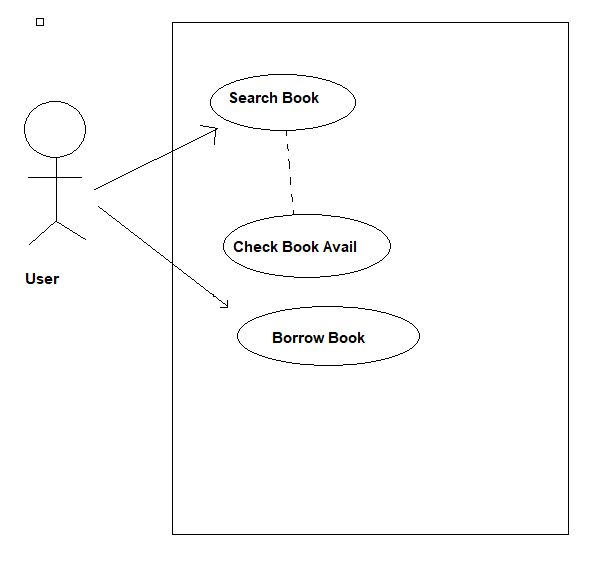


**3.1.3 Module 3 complete CRUD Make a Search/Borrow Function**

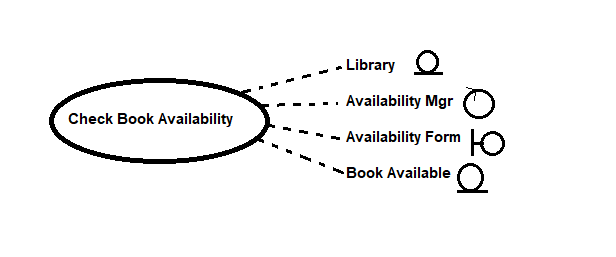
1. ***Description:*** This function allows users to search the books from the LMS and shows only available books to user which he/she can borrow from the LMS.
2. ***Usage Scenario/ Use case Description/******Specification***

|  |  |
| --- | --- |
| Description | Allows access to LMS |
| Inputs | Search book |
| Source | 1. User inputs book name 2. Press search button |
| Alternate case |  |
| Outputs | Book found; Book not found |
| Destination | None |
| Précondition | Authorized User |
| Post Condition | Book should not have been borrowed |
| Side Effects | None |

1. ***Detailed Use case Diagram for Check Book: optional***



1. ***Use case Realization for Check Book: optional***



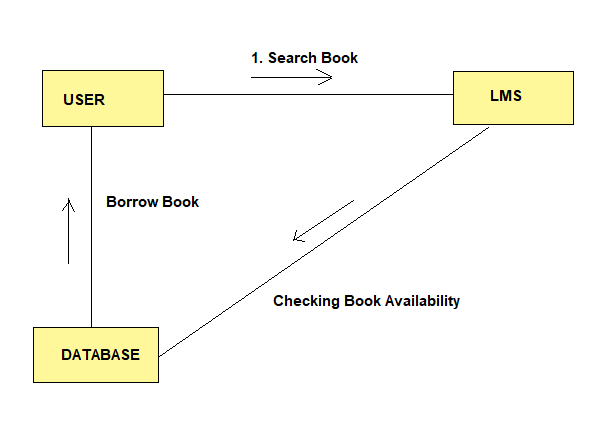
1. ***Flow of Event or Data Flow Diagram for Check Book: optional***



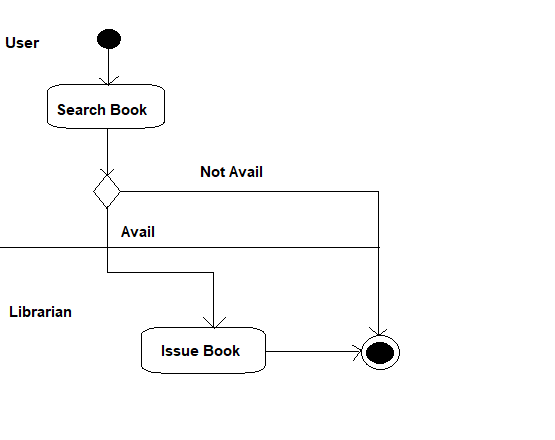
1. ***Sequence Diagram for Check Book: optional***



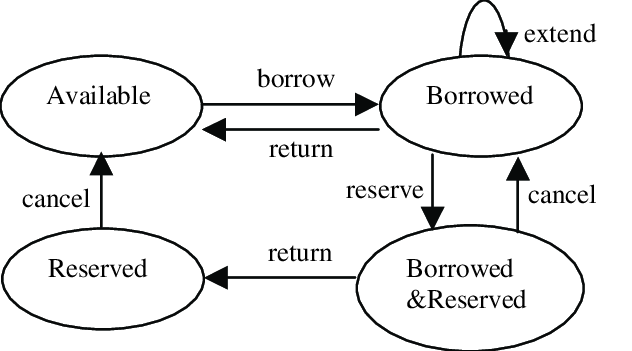
1. ***Collaboration Diagram for Check Book: optional***



1. ***Activity Diagram for Check Book: optional***



1. ***State Chart Diagram for Check Book: optional***

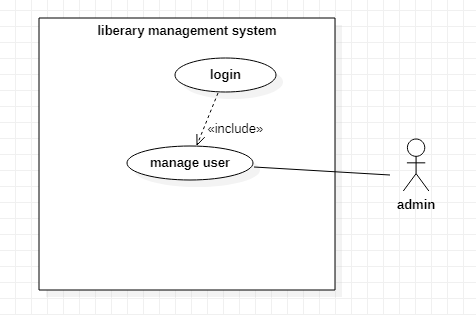


**3.1.4 Module 4 complete CRUD Make a Manage User Function**

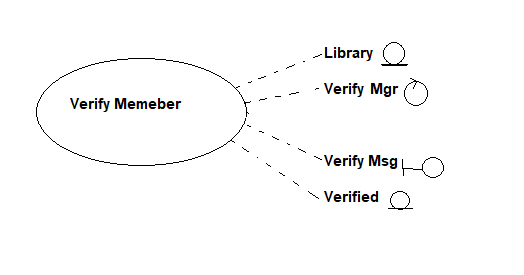
1. ***Description:*** This function allows librarian to manage the registered users in the LMS. He can get all the information of any member who borrow or return the book from the LMS.
2. ***Usage Scenario/ Use case Description/******Specification***

|  |  |
| --- | --- |
| Description | Allows access to LMS |
| Inputs | None |
| Source | 1. Admin click manage users button |
| Alternate case |  |
| Outputs | Registered users |
| Destination | None |
| Précondition | Authorized Admin |
| Post Condition | There must be some users |
| Side Effects | None |

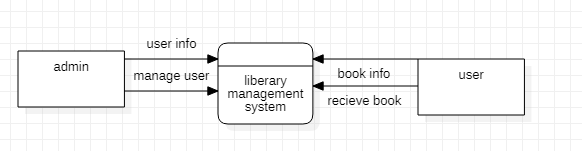
1. ***Detailed Use case Diagram for Manage User: optional***

******

1. ***Use case Realization for Manage User: optional***



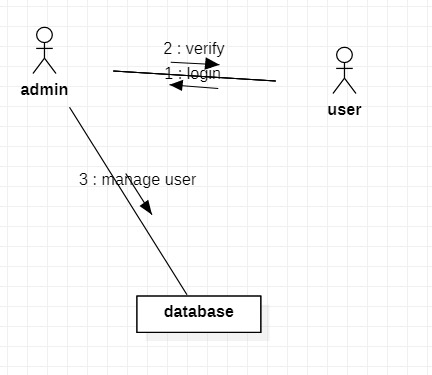
1. ***Flow of Event or Data Flow Diagram for Manage User: optional***

******

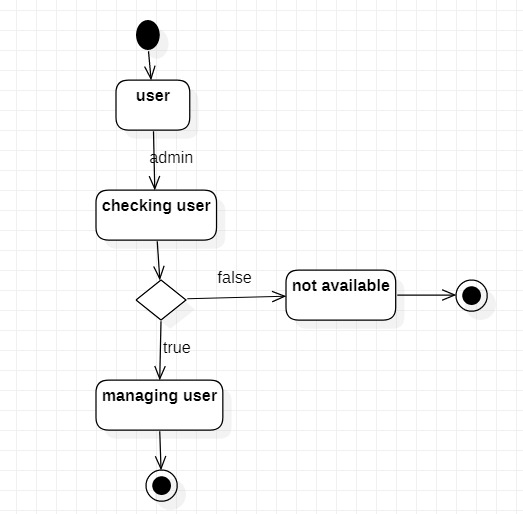
1. ***Sequence Diagram for Manage User: optional***

******

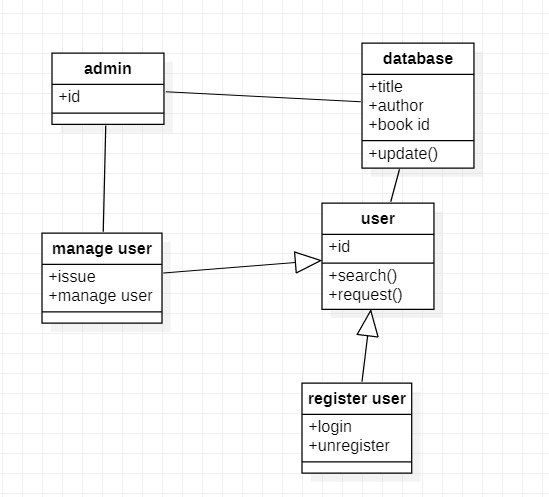
1. ***Collaboration Diagram for Manage User: optional***

******

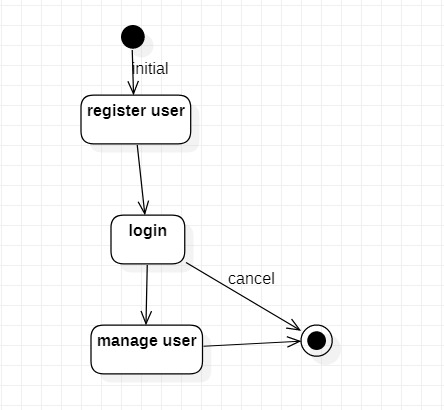
1. ***Activity Diagram for Manage User: optional***

******

1. ***Class Diagram for Manage User: optional***

******

1. ***State Chart Diagram for Manage User: optional***

******

**3.1.5 Module 5 complete CRUD Make a Manage Book Function**

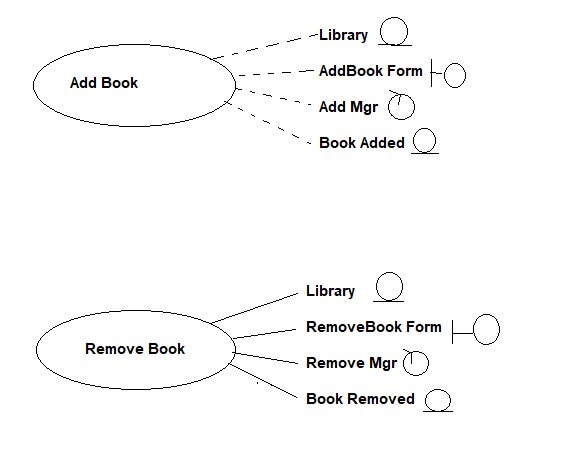
1. ***Description:*** This function allows librarian to manage the registered users in the LMS. He can get all the information of any member who borrow or return the book from the LMS.
2. ***Usage Scenario/ Use case Description/******Specification***

|  |  |
| --- | --- |
| Description | Allows access to LMS |
| Inputs | Book ID |
| Source | 1. User select Book 2. Press delete button |
| Alternate case |  |
| Outputs | Book deleted |
| Destination | None |
| Précondition | Authorized Admin |
| Post Condition | Book should be listed |
| Side Effects | Book will be removed from DB |

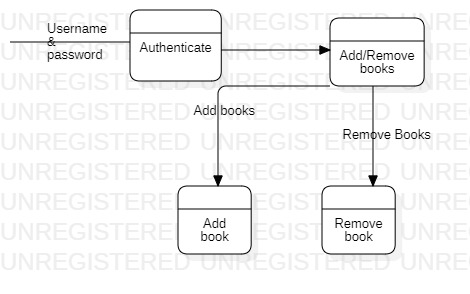
1. ***Detailed Use case Diagram for Manage Book: optional***

******

1. ***Use case Realization for Manage Book: optional***



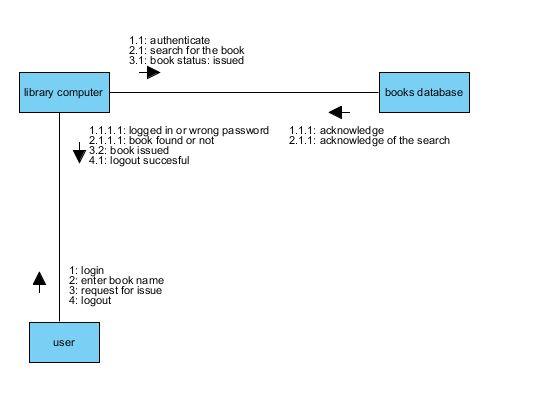
1. ***Flow of Event or Data Flow Diagram for Manage Book: optional***

******

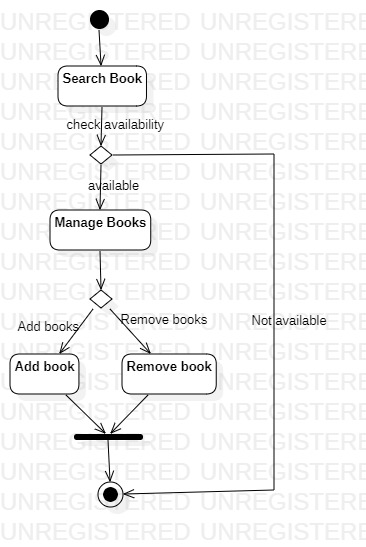
1. ***Sequence Diagram for Manage Book: optional***



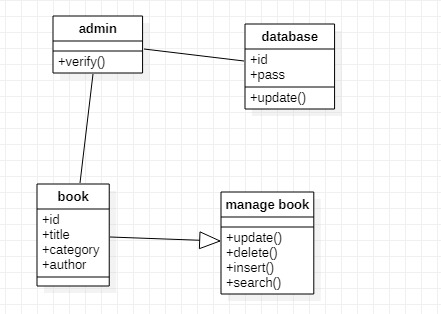
1. ***Collaboration Diagram for Manage Book: optional***

******

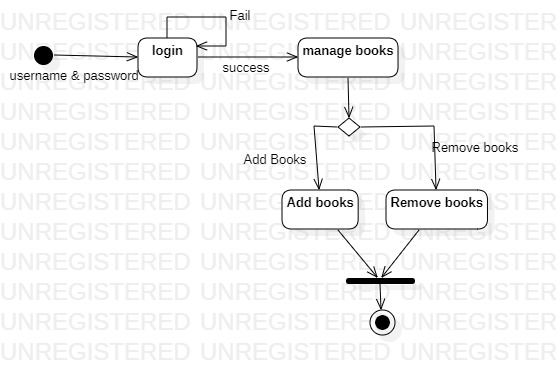
1. ***Activity Diagram for Manage Book: optional***

******

1. ***Class Diagram for Manage User: optional***

******

1. ***State Chart Diagram for Manage Book: optional***

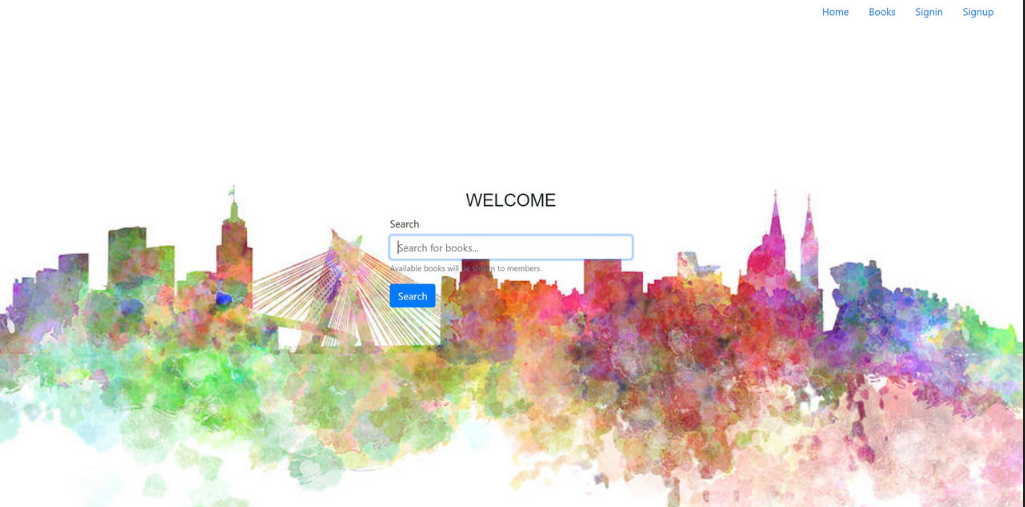
******

**3.2. External Interface Requirements**

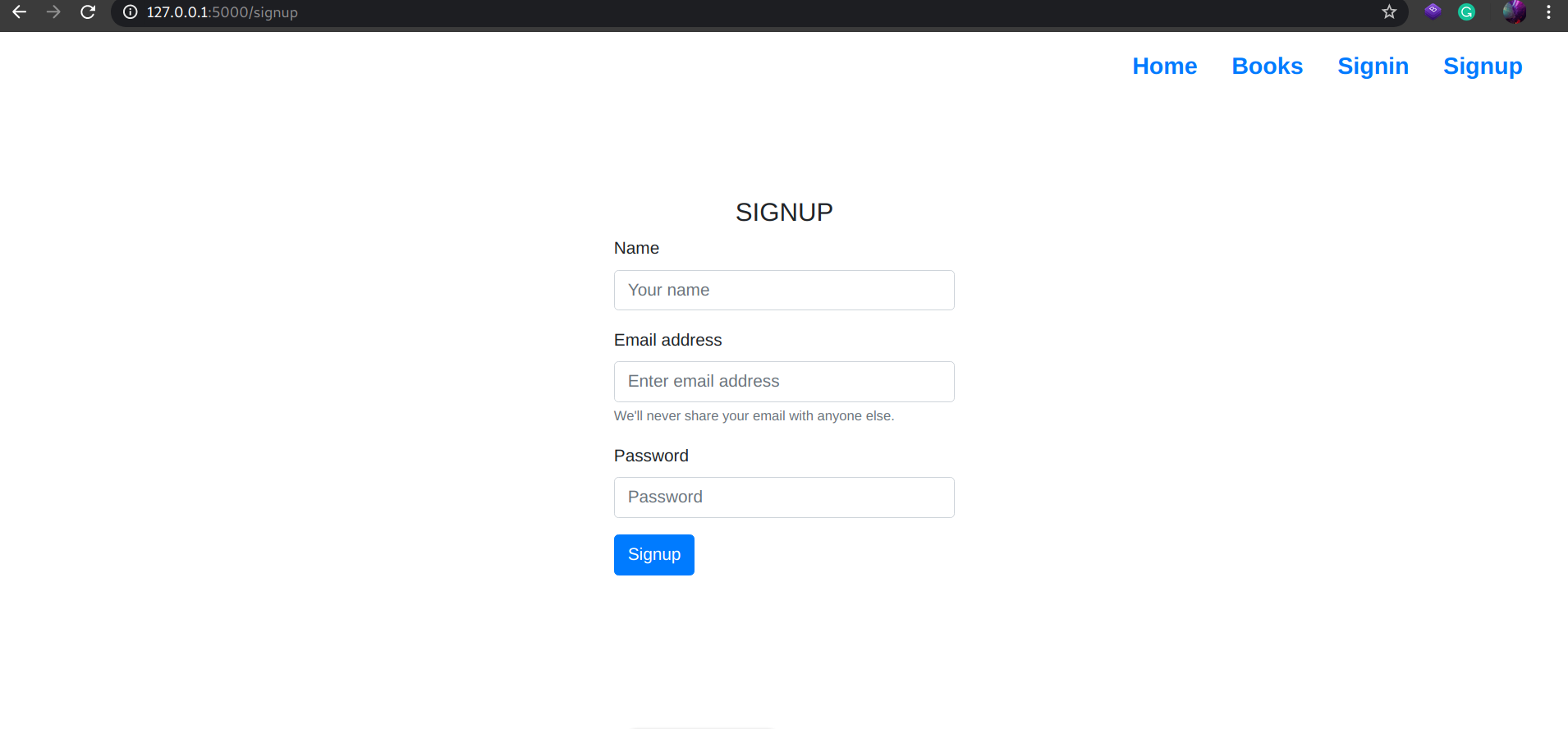
**3.2.1 User Interfaces**

The user interfaces are divided into two major components. One part includes the user accessing the system. The other portion involves accessing the system via admin panel

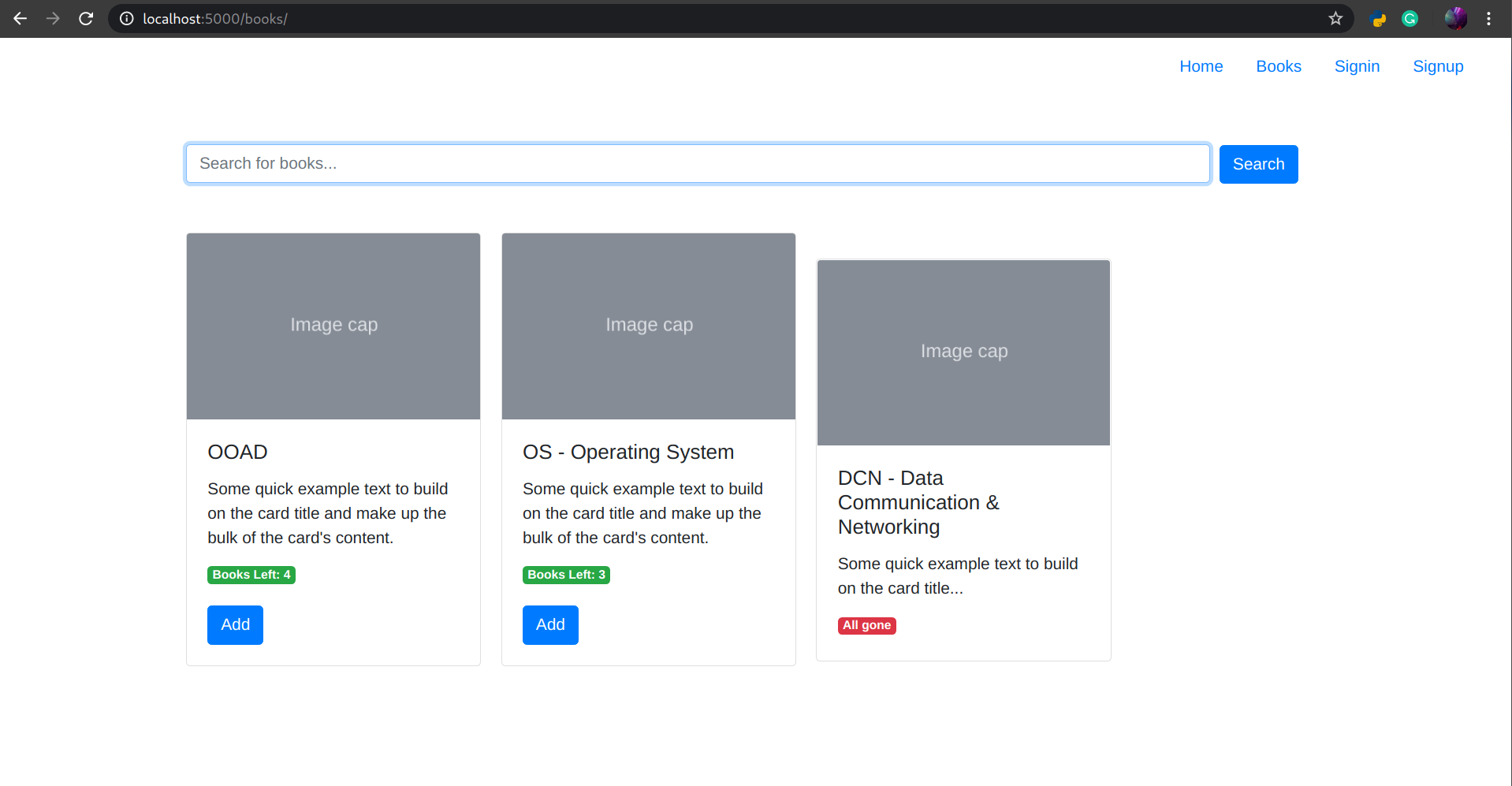
The diagram below illustrates the four **major functionalities or modules.** These functionalities will be displayed depending on the user. For instance, the LMS will see all four functionalities while the normal user and the admin will manage the user and the book on the LMS



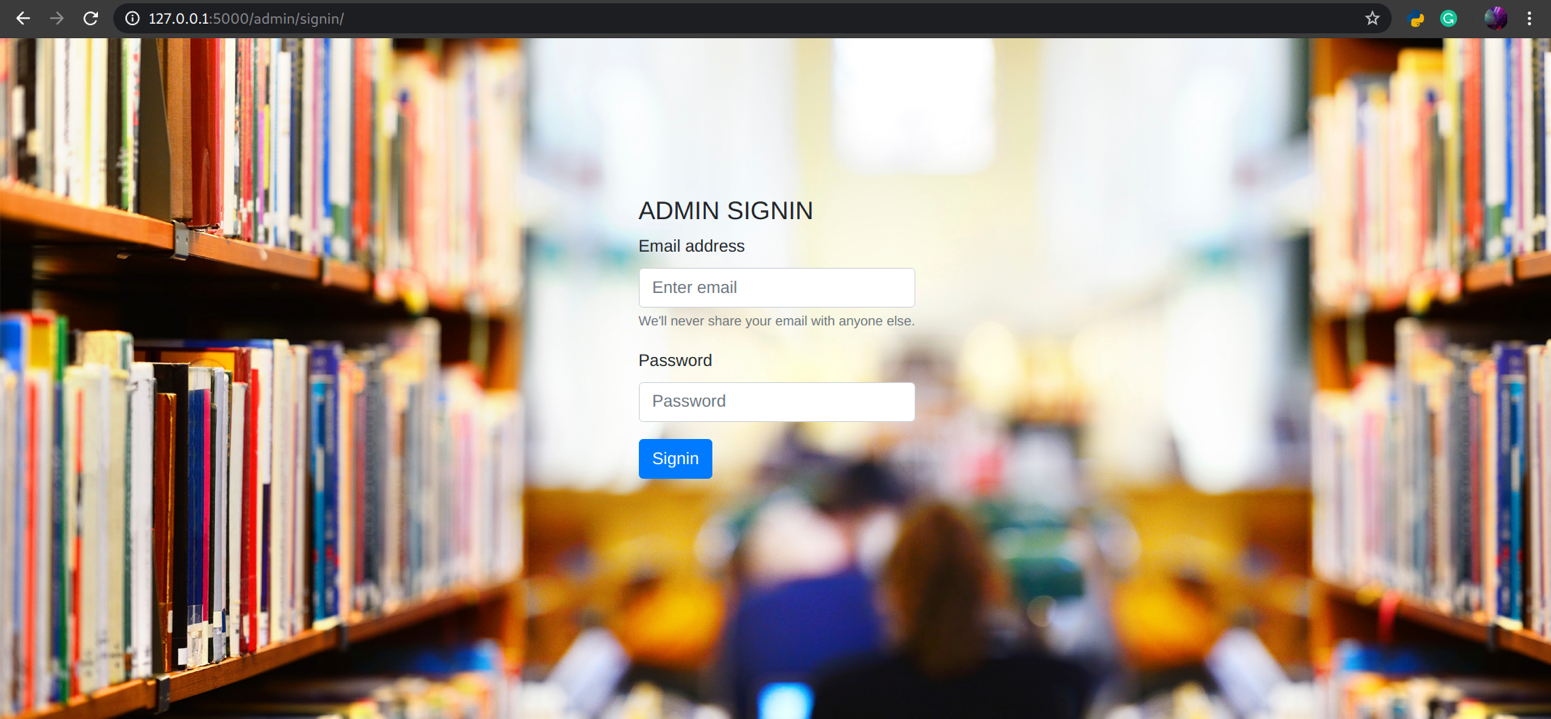
This is the main screen of the application. Where user can sign up or the application and after that user will login in to the account and use the application.



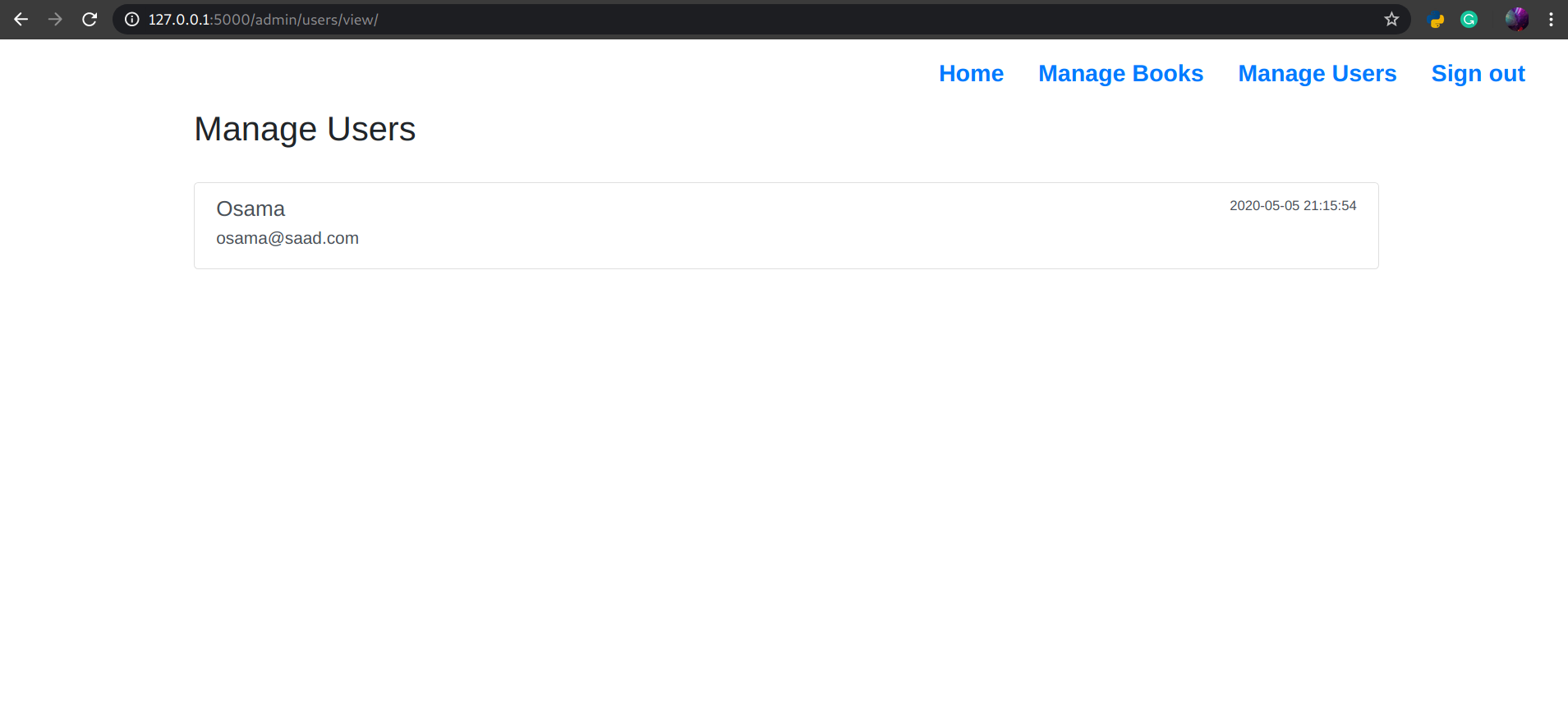
This is the sign up main of the application where user enter all the required fields. After registration user can login into his/her account.



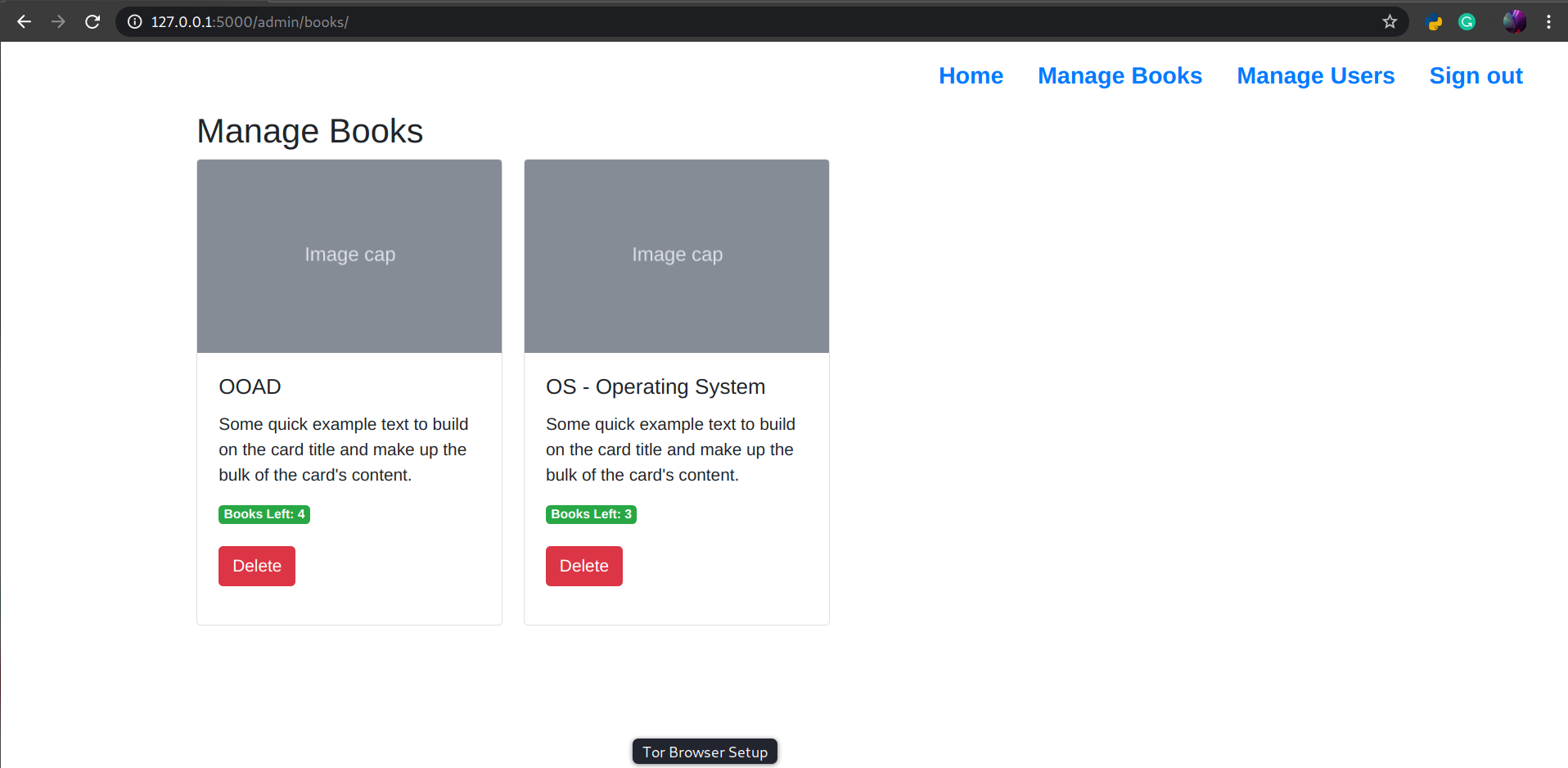
This is the display page of the LMS which user can see after login into his/her account. He can search for book and check the availability of the book. If the book is available user can borrow the book the LMS.



This is the admin sign in page for the admin. Once admin login into his account he can manage the books on the LMS. He can also manage the users registered on the LMS.



This is the manage users page only visible to admin after the admin authenticate to the application. He can manage the users and can view all the registered users on the application and can view all the details of the application.



This is the manage book page only visible to admin after the admin authenticate to the application. He can manage the book and can view all books. He can delete the book from the LMS

**3.2.2 Hardware Interfaces**

To develop or implement this project we need different types of hardware configuration for the server and client. The Client Machines: - The Server Machines: - Processor RAM Hard Disk Intel Pentium III or other

.

**3.2.3 Software Interfaces**

It is a web base application so it operates in all famous browsers for testing we take edge, Mozilla Firefox and Google Chrome. The main requirement to use this online product would be Internet Connection.

**3.3 Performance Requirements**

The following sections list the performance requirements for the system.

**3.3.1 User Requirements**

|  |  |
| --- | --- |
| User Requirements | **Description of Requirement for Design Environment** |
| Location(s) and Number(s) of Users | Pakistan |
| Expected Growth in Number of Users |  |
| After 1 Year | 50% |
| After 2 Years | TBD |
| After 3 Years | TBD |
| User Expectation |  |
| Interactivity | User expect that it provides a very easy to use graphical user interface |
| Reliability | For some applications, reliability must be 100% during the application session |
| Adaptability | Network must adapt to user additions, deletions and changes |
| Security | Encryption software would be used for Credit Card transactions |
| Cost / Funding | Less than $250K |

**3.3.2 Application Requirements**

Since no specified service is indicated, then we have listed the applications as best – efforts. This may change as we learn more about the application.

The communication package is determined to be bursty in nature, with small data sizes and frequent transmissions. We can consider this application to be interactive-burst, while the database transaction-processing application is described by the LMS as transferring large amounts of data (initial estimates are 1 MB/transaction), we have listed this application as interactive-bulk.

|  |  |  |
| --- | --- | --- |
| Categorizing  Applications | Best-Efforts | Application  Locations |
| Communication | 100 Kb/s | Pakistan |
| Database Access | 400 Kb/s | Pakistan |
| Database Transaction processing | 1.5 Mb/s | Pakistan |

**3.3.3 Host Requirements**

|  |  |  |
| --- | --- | --- |
|  | Type of Host or  Equipment | Numbers and  Locations |
| Host A | PC | Pakistan |
| Host B | Database Server | Pakistan |
| Host C | Application Server | Pakistan |

**3.4.1 Standards Compliance**

There are no design constraints that can be imposed by other standards limitations.

**3.4.2 Software Limitations**

        must be able to run Internet Explorer or Netscape Communicator web browsers to access the system.

        must have cell-phone web-based capability to access the system from a mobile phone.

**3.4.3 Hardware Limitations**

        Input/Output: One or two-button mouse, keyboard, cell-phone, or touch screen required.

        Network card required at thin-client terminals to make communication with server possible.

**3.5 Quality Characteristics**

There are a number of quality characteristics that apply to the ARRS software system.

**3.5.1 Portability**

The ARRS system will be developed using HTML and Java so that it can be accessed from any type of system using just a regular web browser. It will also be available to users that have web access on their cellular phones. The system will be tested on all types of hardware before being released to ensure that is it compliant with this requirement.

**3.5.2 Reliability**

The system should be capable of processing a given number of reservations within a give time frame with no errors and the system should be available and operational all the time. During the development of the prototype for the 3 cities, the system will be tested in its actual environment to ensure that it can handle the load of reservations that occur during a regular workday.

**3.5.3 Usability**

The LMS system will be developed so that it is an easy to use system that requires the least amount of user input possible. Every input will be validated. The user should only have general computer use knowledge. Error messages will be displayed if the user enters an invalid value or tries to access a function without the required permissions. An easy and well-structured user manual will be provided to the LMS and the system will include descriptive help for all operations allowed.

**3.5.4 Correctness**

The LMS system will be considered correct when the LMS approves the prototype presented and agrees that all the functions they require are implemented as stated in the Software Requirements Specification.

**3.5.5 Flexibility**

The ARRS system should be developed in such a way that it is easily customizable. If new functions are required by LMS, there will be little effort required to update the system to support new cities or new transactions.

**3.5.6 Security**

The LMS system should not compromise the customer information at any time. The user information will never be sold to other parties and will be kept secure at all times. Users will be authenticated to ensure that no unauthorized users gain access to private information.

**3.5.7 Maintainability**

The LMS source code will be kept well structure and documented so that it is easier to maintain and extend the system. All changes to the system shall be documented.

**3.6 Other Requirements**

Certain requirements may, due to the nature of the software, the user organization, etc., be placed in separate categories such as those below.

**3.6.1 Data Base**

The Automate LMS will have main databases. Which stores all the information about the users and book in the LMS. This database is MySql. The following are the requirements for these databases that are to be developed as part of the product. They include:

**Reservation Database**

|  |  |
| --- | --- |
| Types of information | Schedule information for the trains, including date, time, departure city, destination city, ticket cost and ticket availability for a particular train |
| Frequency of use | Depends on the passenger demand, which may reach 25,000 per day during peak periods |
| Accessing capabilities | The database should allow access to at least 1,000 people at once; the users will have a general access to the information about the train schedule, and a secure access to the reports (available only to LMS officials) using a username and a password |
| Data element and file descriptions | To be determined |
| Relationship of data elements, records and files | To be determined |
| Static and dynamic organization | To be determined |
| Retention requirements for data | user information will be available as long as the user have registered in the LMS |

**Passenger Account Database**

|  |  |
| --- | --- |
| Types of information | user account information including their name, address, phone numbers, last reservations, balance owed, credit card number (if they paid by a credit card) |
| Frequency of use | Depends on the passenger demand, which may reach 25,000 per day during peak periods |
| Accessing capabilities | The database should allow access to at least 500 people at once; the users will have a secure access to the database using a username and a password |
| Data element and file descriptions | To be determined |
| Relationship of data elements, records and files | To be determined |
| Static and dynamic organization | To be determined |
| Retention requirements for data | Passenger account will be available for as long as a passenger is using the account, and at least for 6 month since the passenger logged on last time. |

**3.6.2 Operations**

The normal operations required by the user can be viewed as the following:

User-initiated Operations:

These operations include the login operation, which is initiated by the users. Also, the process of becoming a new user is in this category. Building, changing, and viewing itineraries, as well as paying for the itinerary are all initiated by the users. The user initiates the report generation activity, as well as changing train schedules.

Interactive Operations and Unattended Operations:

The users initiate all the operations mentioned above, and almost all of them are somehow interactive. Displaying the train schedule is non-interactive. The report display is a non-interactive operation, although selecting the desired reports will require user input.

Data Processing Support Functions:

The user account data is used to create new accounts, as well as to validate user id's during login functions. For building itineraries, user input, user account data, and train schedule data are used, and processed. User data along with final results of user interaction (whether the user purchased a trip, number of tickets bought, etc.) are collected, and used for report generation purposes. Administrative users' inputs are collected in order to modify and present schedules.

Backup and Recovery Operations:

Both databases used (passenger account database and reservations database) are production databases. The main operation used for the backup and recovery is Oracle's built-in cold backup, which is also known as the "archive mode". Depending on the customer's needs and budget, additional redundancy can be added using systems like RAID 5 and tape backup.

**3.6.3 Site Adaptation Requirements**

There are no site adaptation requirements for this project.

**4. Supporting Information.**

There is no supporting information required for this project.