DESIGN DOCUMENT
FOR
SMART LIBRARY
MANAGEMENT SYSTEM

GROUP – 06
Prepared by

MUKKALA KEERTHI SAI NANDAN REDDY

GUVVALA SHANMUKH SHASHANK REDDY

VATRAPU RAMI REDDY

YENDLURI JAHNAV

KAKARA KARTHIKEYA

KALINGIRI SAI MANIKANTA

SHREYAS S N

ABHINANDH K R

NIDHIN S

POTHULA MATTHEWS

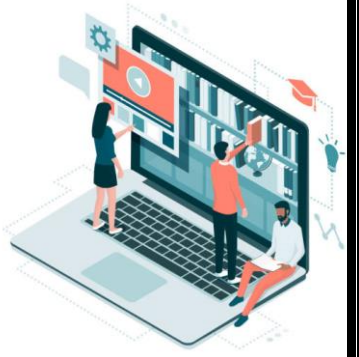


Table of Contents

1 Purpose 3

1.1 Document Objectives 3
1.2 Audience Intended..... 3
1.3 Acronyms and Abbreviations..... 4
1.4 Features..... 4

2 Assumptions, Constraints & Dependencies.... 5

2.1 Assumptions..... 5
2.2 Constraints & Dependencies..... 5

3 System Overview..... 5

3.1 Database Software Utilities..... 5
3.2 Support Software..... 5

4 Architecture..... 6

4.1 Hardware Architecture..... 6
4.2 Software Architecture 7
4.3 Datastores..... 10

5 Database-Wide Design Decisions..... 11

5.1 Interfaces..... 11
5.1.1 Register Interface..... 11
5.1.2 Login Page..... 12
5.1.3 Details Page..... 13
5.1.4 Change Password..... 14
5.1.5 Search Book..... 15
5.1.6 Recommendations..... 16
5.1.7 View Responses..... 17
5.1.8 Add Book..... 18
5.1.9 View Recommendations..... 19
5.1.10 View Requests..... 20
5.1.11 Currently Issued Books..... 21
5.2 Key Factors Influencing Design..... 22
5.2.1 Functional Requirements..... 22
5.2.2 Non-Functional Requirements..... 23

5.3 Behaviour…… 24
5.4 DBMS Platform…… 25
5.5 Security and Availability…… 26
5.6 Distribution…… 26
5.7 Backup and Restore Operations…… 27
5.8 Maintenance…… 27
5.9 Performance and Availability Decisions…… 28
6 Database Administrative Functions…… 29
6.1 Database Identification…… 29
6.2 Schema Information…… 29
6.3 Schema Description…… 30
6.4 Physical Structure…… 32
6.5 Entity Mapping…… 35
6.6 Mapping Rules…… 35
6.7 Operational Implications…… 36
6.8 Backup and Recovery…… 37
6.9 Applications/Systems Using the Database…… 38
6.10 Relation to Other Databases…… 38
7 Detailed Database Design…… 38
7.1 Data Software objects and resultant data structures... 38
8 Appendix…… 41

Revisions

Version	Date	Author	Description
Version <1.0>	10 November,2022	Group 6	This is 1st version of this model

1. Purpose

This design document helps us to map the logical data to the required Database Management System keeping an account system's performance requirements by converting logical or conceptual data constructs to physical storage constructs (e. g. tables, files) of the Database Management System.

1.1 Document Objectives

Main objectives of this design documentation are:

- To specify the design of a database. i.e. a collection of related data stored in one or more computerized files that can be accessed by users or computer developers via a DBMS.
- To serve as a basis for implementing the database and related software units. It provides the acquirer visibility into the design and provides information necessary for software development.

1.2 Audience Intended

This document focuses the following groups:

- Technical reviewers who need to evaluate the quality of this document.
- Architects whose overall architectural design must meet the requirements described in this document.
- Designers whose design must meet the requirements specified in this document
- Developers whose software needs to implement the requirements specified in this document.
- Quality Assurance Officer whose test case needs to verify the requirements specified in this document.

1.3 Acronyms and Abbreviations

Acronym/Abbreviation	Meaning
RDBMS	Relational Database Management System
DBA	Database Administrator
1NF	First Normal Form
2NF	Second Normal Form
3NF	Three Normal Form
BCNF	Boyce Codd Normal Form
OS	Operating System

1.4 Features

To solve inconveniences as mentioned in the existing system, **Online Library** is proposed. The proposed system contains the following features.

- Students will register themselves through online mode.
- Individually each member can access it through his own individual account.
- Book details like book, authors, number of copies totally maintained by library.
- Issue dates and returns of each member is maintained separately and fine charged if there is any delay in return the book.
- Administrator can add, update the books.
- Time consuming is low, gives accurate results, reliability can be improved with the help of security.

2. Assumptions, Constraints, & Dependencies

2.1 Assumptions

The product needs the following third-party applications for the development of the project.

- Xampp (To run our Database)
- Visual Studio Code
- Photoshop (for editing layouts, icons, etc.)

2.2 Constraints & Dependencies

Any update regarding the book from a library is to be recorded and updated to have correct values.

3. System Overview

This database was developed for users who use web applications to access the library. This database stores details of students/faculties using the library, as well as administrators who track student orders, updates, and book updates.

3.1 Database Software Utilities

The database management system used in MySQL. As the name says SQL, here SQL is used for creating and handling database and all required functions related to the database. We require to start Xampp servers before performing any functions or queries.

3.2 Support Software

The software directly related to the database we are using for storing our data is MySQL database. It uses SQL as its query language to perform all the database related queries whereas **MySQL** is written in C and C++.

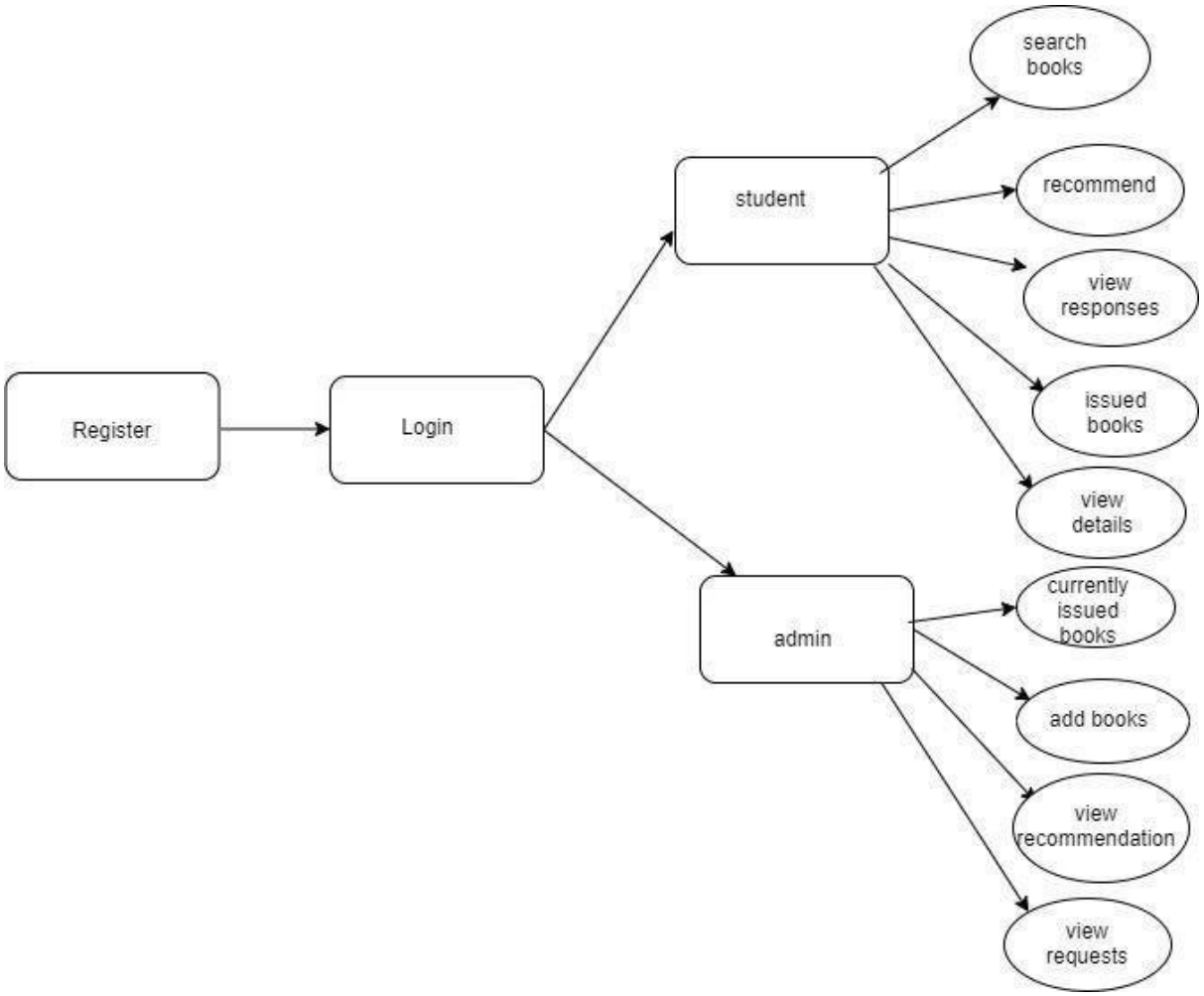
MySQL is a web server database and all the details regarding it we are going to handle through Apache server. It will handle all the files related to the database.

4. Architecture

4.1 Hardware Architecture

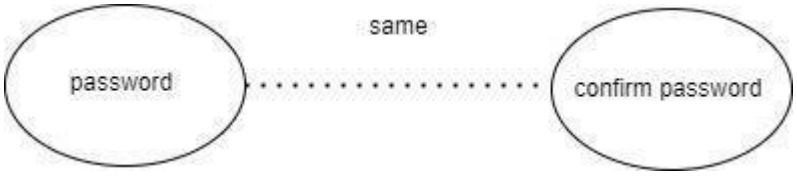
In hardware architecture we will mostly see how the data is stored in databases and how they are connected, it mostly comprises internal level. Therefore, it deals with how the information is stored.

Flow Chart:



Register Table:

Register table stores username, email, password, branch. Here we need to enter password in confirm password. If password and confirm password are same the user will be registered.



Login Table:

Login table stores email and password and once the user enters the email and password if they are present in the register table then it will be stored in the login table.

Add books table:

In this table book id, book names are stored when the admin adds the book.

Recommendations table:

Once the student enters a book as a recommendation it will be stored in this table.

Issued table:

In this table book id, student id will be stored.

View Requests table:

In this table the book id and book name which is requested, and student id is stored

View Responses table:

In this table the responses given by admin is stored.

4.2 Software Architecture:

Software architecture mostly deals with external level. Software architecture deals with interfaces. In Library Management System, we have many interfaces and now let us see how they work.

INTERFACES:

1.Register interface

2.Login interface

3.Admin interface

4.Student interface

1. ***Register interface:*** In this interface the user can register by entering details like email, password, username, branch etc. And when they completed their registration, they can login into the portal by clicking on the login button. Then it will redirect them to the login page.

2. ***Login interface:*** In the login interface the user can enter their email and password and they can login to the portal and after login they have many options like search books, request a book. In this Login interface we have many options like home, if we click on home, we will be redirected to register page.

3. ***Admin interface:*** In the login interface, the admin can enter his email and password and it will redirect him to the admin page where he can add books, see requests, issue books.

4.*Student interface*: In the login interface, students can enter their email and password and it will redirect them to a student page where they can search for books, see responses, and recommend books.

Components: Library Management System has many components. They are listed below:

Student: A student can search books, put forward a request for a book and view recommended books, view their details, issued books, return/renewal request, view responses sent by admin.

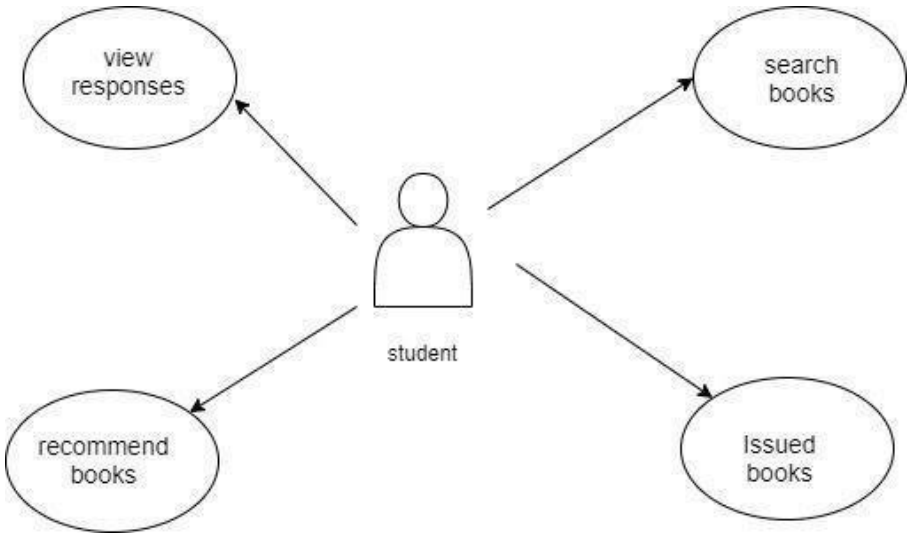
Search books: A student can search for books which are added by admin. If the book that is required for the student is present then the student can request for the book.

View details: A student can view his/her details like username id, branch, email, password.

Issued books: A student can see the books issued to him by clicking on this. By this he can have an idea of what books he has taken.

Recommend book: A student can recommend a book which he/she feels best for that particular subject.

View Responses: A student can view the responses he received from the admin, if the book he requested is issued or not and his renewal request is accepted or not.



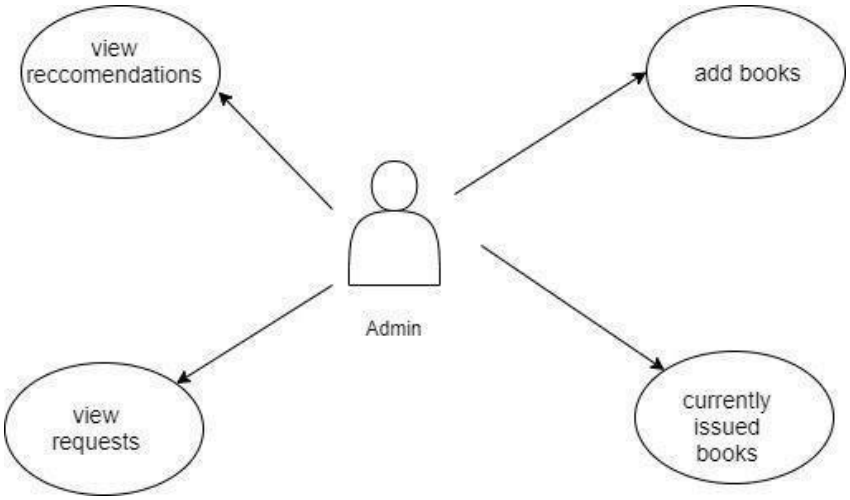
Admin: An admin can add books, view requests, can view currently issued books, view recommended books.

Add books: Admin can add books to the library subject wise so that students can get the books required for that subject.

View requests: Admin can view requests received from students and can issue books accordingly. If a student did not return the book, then admin can or cannot issue the requested book according to admin's wish.

Currently issued books: Admin can view the books issued to the students. He can or cannot issue the book based on whether the student had returned the book or not.

View recommended books: Admin can view books recommended by students so he can add more books which are recommended so that all the students can have access to those books.



4.3 Datastores

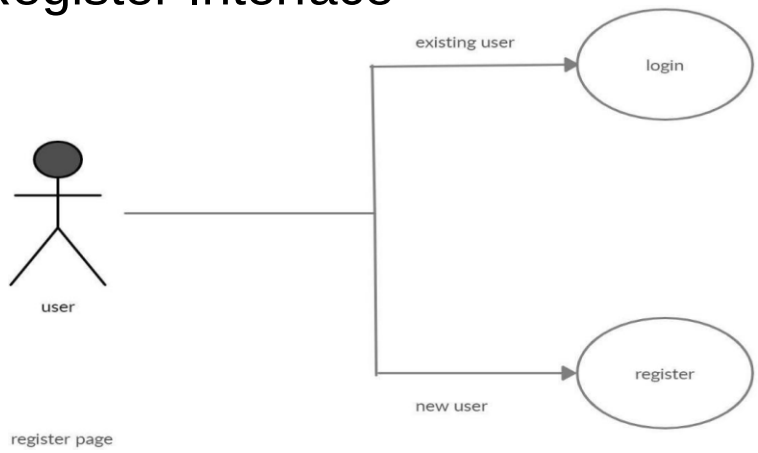
Data is stored in the library management database and in the tables mentioned above. For file management, we are storing

the contents in htdocs in the Xampp folder. PHP files are used for backend and HTML and CSS files are used for the front end.

5. Database- Wide Design Decisions

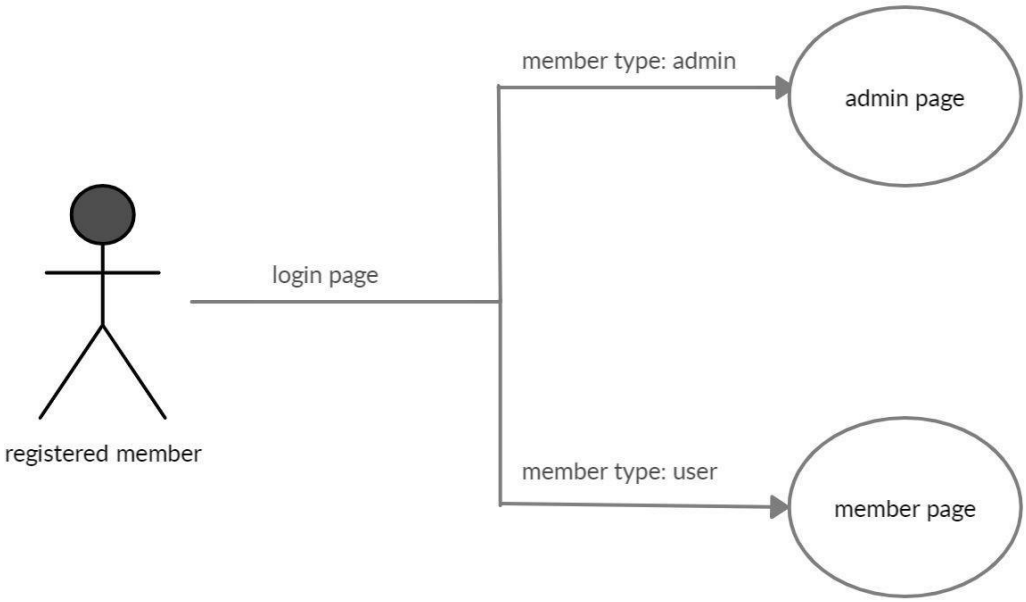
5.1 Interfaces

5.1.1 Register Interface



Interface	Register Interface
Summary	User have to register in the Database
Actor	User
Trigger	Admin, Student
Primary Scenario	User have to fill in the details to register
Alternative Scenario	If user exists, then user can login directly
Exceptional Scenario	None
Pre-Conditions	None
Post-Conditions	User will be redirected to login page
Assumptions	User has an internet connection

5.1.2 Login Page



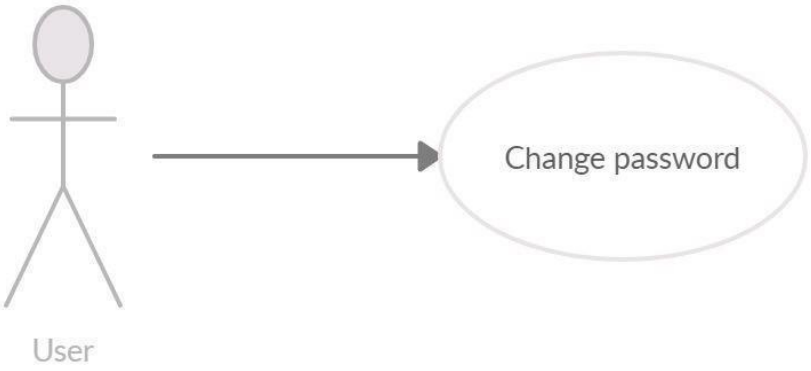
Interface	Login Page
Summary	Registered Members Login
Actor	Registered Member
Trigger	Admin/User
Primary Scenario	Member gives Login ID & Password
Alternative Scenario	If password is wrong, ERROR message is shown
Exceptional Scenario	None
Pre-Conditions	Member should be registered
Post-Conditions	If details are right, will be redirected to respected member page
Assumptions	User has an internet connection

5.1.3 Details Page

When a user logs in, he/she will be directed to the details page.

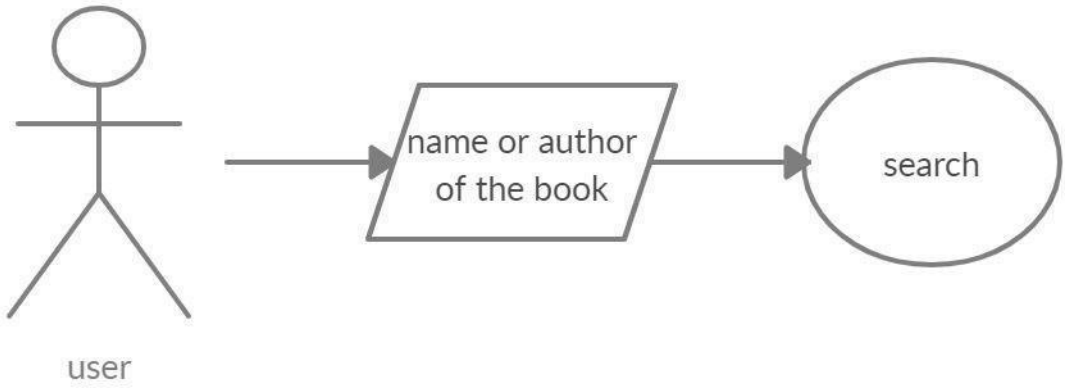
Interface	Details of user and links to other interfaces
Summary	Can view their details
Actor	User
Trigger	Logging in
Primary Scenario	User can see his/her details after logging in
Alternative Scenario	None
Exceptional Scenario	None
Pre-Conditions	User should successfully log in to the portal
Post-Conditions	User can view their details
Assumptions	User has an internet connection

5.1.4 Change Password



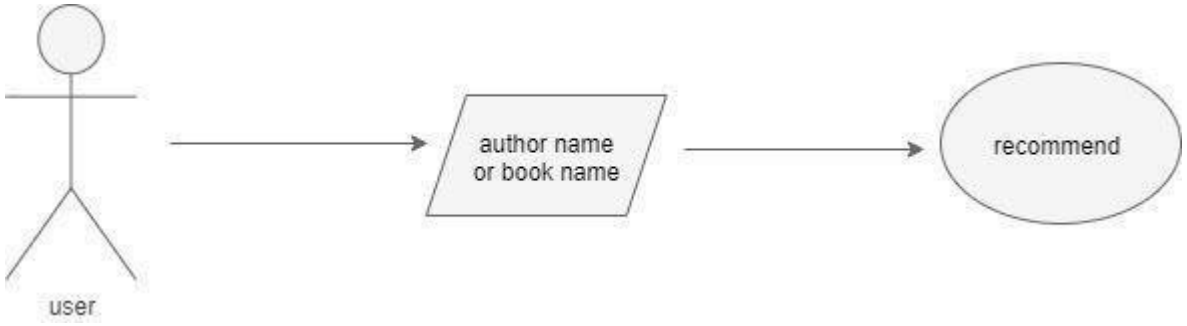
Interface	Details of user and links to other interfaces
Summary	Can change their password
Actor	User
Trigger	Logging in
Primary Scenario	User has a new password
Alternative Scenario	None
Exceptional Scenario	None
Pre-Conditions	User should successfully log in
Post-Conditions	User has successfully changed the password and view their details
Assumptions	User has an internet connection

5.1.5 Search Book



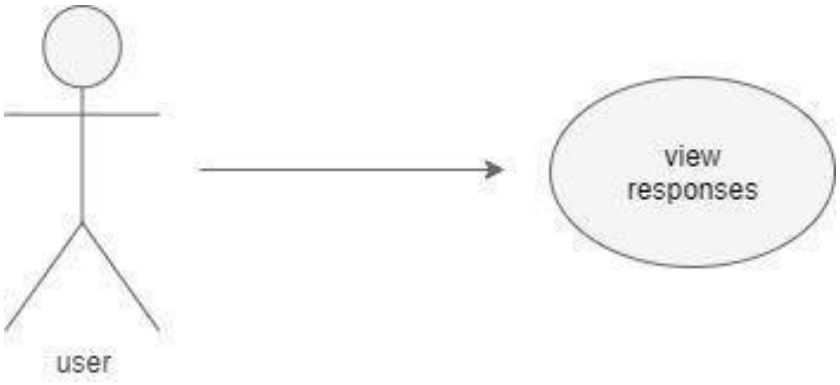
Interface	Search for a Book
Summary	User search for a required book
Actor	User
Trigger	Search Button
Primary Scenario	User can search for a required book through search button
Alternative Scenario	None
Exceptional Scenario	None
Pre-Conditions	User should successfully log in
Post-Conditions	User can request the issue of the searched book
Assumptions	User has an internet connection

5.1.6 Recommendations



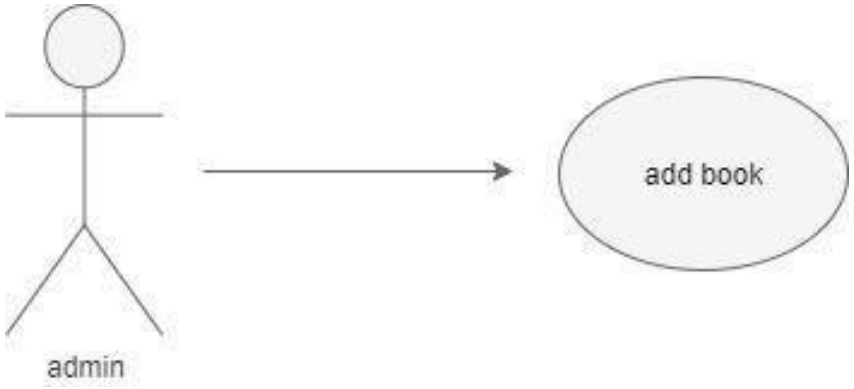
Interface	Recommend a book
Summary	User can recommend a book
Actor	User (student)
Trigger	Recommend button
Primary Scenario	User(student) can recommend a book that he/she finds useful
Alternative Scenario	None
Exceptional Scenario	None
Pre-Conditions	User should successfully login
Post-Conditions	Student can recommend a book when he/she found a book which is helpful
Assumptions	User has an internet connection

5.1.7 View Responses



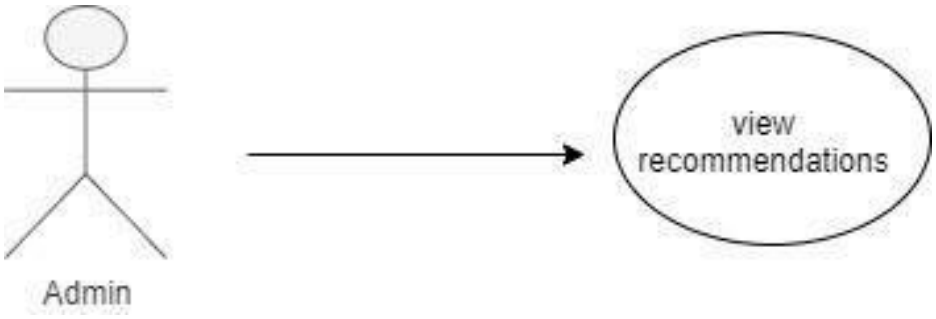
Interface	View responses interface
Summary	User(students) can view responses sent to him/her by admin
Actor	User(student)
Trigger	View responses button
Primary Scenario	Student should have put forward an issue request to the admin
Alternative Scenario	None
Exceptional Scenario	None
Pre-Conditions	User should successfully login
Post-Conditions	User can view whether his request is accepted or not
Assumptions	User has an internet connection

5.1.8 Add Book



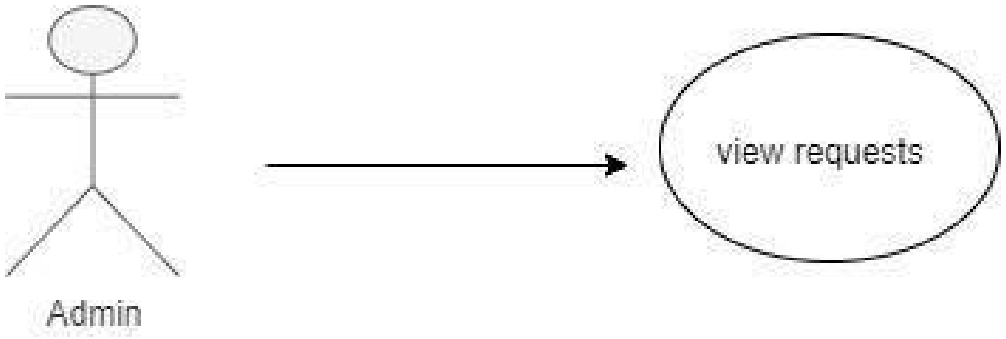
Interface	Add book interface
Summary	Admin can add a book
Actor	Admin
Trigger	Add book button
Primary Scenario	Admin can add books according to the requirement
Alternative Scenario	None
Exceptional Scenario	None
Pre-Conditions	Admin should successfully login and should have a list of books to add
Post-Conditions	Admin can successfully add books, so that every student can have access to them
Assumptions	User has an internet connection

5.1.9 View Recommendations



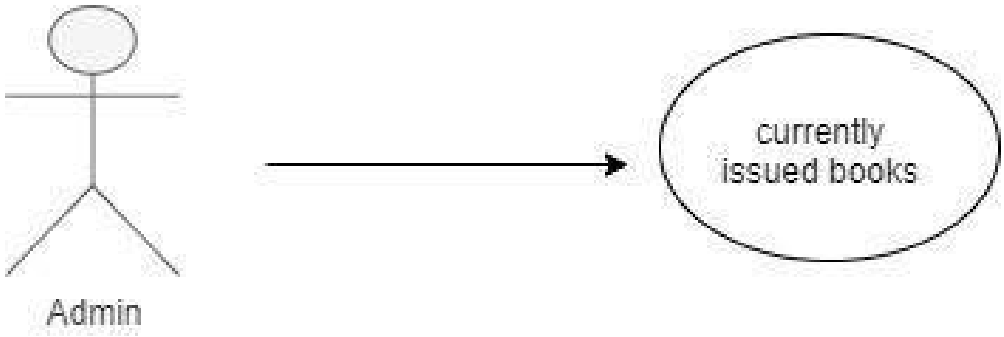
Interface	View Recommendations interface
Summary	Admin can add books by viewing recommendations
Actor	Admin
Trigger	View recommendations button
Primary Scenario	Admin can view books that are recommended by students
Alternative Scenario	None
Exceptional Scenario	None
Pre-Conditions	Admin should successfully login
Post-Conditions	Admin can view books recommended and add books accordingly
Assumptions	User has an internet connection

5.1.10 View Requests



Interface	View requests interface
Summary	Admin can view requests received from students
Actor	Admin
Trigger	View requests button
Primary Scenario	Admin can view the requests sent to him by students and issue books accordingly
Alternative Scenario	None
Exceptional Scenario	None
Pre-Conditions	Admin should successfully login
Post-Conditions	Admin can issue requested books to students
Assumptions	User has an internet connection

5.1.11 Currently issued books



Interface	Currently issued books interface
Summary	Admin can view the list of books issued recently
Actor	Admin
Trigger	Currently issued books button
Primary Scenario	Admin can keep track of books by knowing what and how many books were issued
Alternative Scenario	None
Exceptional Scenario	None
Pre-Conditions	Admin should successfully login
Post-Conditions	Admin can keep track of books easily
Assumptions	User has an internet connection

5.2 Key Factors Influencing Design

Key factors influencing the design are the functional and non-functional requirements.

5.2.1 Functional requirements

User Login

- User Id is provided when they register
- The system allows user with valid id and password only
- The system can perform an authorization process which what a user can access
- The User must be able to log out after they finish using the system

Register

- System must be able to verify information
- System must be able to delete information if it is wrong

Search

- System must be able to search the database based on select search type
- System must be able to filter book based on keyword entered
- System must be able to show the filtered book in table view

Add Book

- System must be able to verify information
- System must be able to enter the number of copies into a table

- System must be able to not allow two books having same book id

Recommend Books

- System must be able to enter recommended book information into the database

Issue and Return

- System must be able to enter issue information in the database
- System must be able to update number of books
- System must be able to search if book is available or not before issuing the book
- System should be able to enter issue and return date information

Change Password

- System must be able to verify the information
- System must be able to update the user password when changed

5.2.2 Non-Functional Requirements

1. Flexibility: The operation may be flexible and reports can be presented in many ways.
2. Maintainability: After the deployment of the project if any error occurs then it can be easily maintained by the software developer.
3. Portability: The software can be deployed at any machine.
4. Reusability: The data and record that are saved in the database can be reused if needed.

5. Robustness: If there is any error in any window or module then it does not affect any other part of the software.
6. Timelines: The time limit is very important. It will save much time and provide fast accessing.
7. Security: Data is backed up regularly, the database is protected and users cannot edit the database. For security reasons, only administrators are allowed to edit the database, and only admins can access the database.
8. Performance: Performance should be accurate and response time should be kept minimum so that there are no issues with processing the request. Data should be backed up regularly as a log file to prevent data loss if the data is lost completely due to server crash or file corruption. To be more effective, the DBMS needs to be able to process large amounts of data and tale action in less time.

5.3 Behaviour

Decisions on behaviour in response to each input or query in each page/interface:

1) In register interface, users enter all required details and click submit then the user is redirected to the login page and details are stored in the database.
2) If the same user tries to register again it will show that the user is already registered and there will be no redirection.
3) Email and ID are unique so when matching details are entered it will show data exists.
4) On the login page if details are entered correctly the user is redirected to the user page.

5) If login credentials entered are wrong, it will show that the data entered is wrong.
6) Special credentials are provided for admin. So, when he/she enters them, they will be redirected to the admin page.
7) In admin page, he/she can issue a boo for students when the requested page will reload and show the issued books for respected students.
8) The admin can add a book in the add book section with certain details and the page gets reloaded and displays the container with no details.
9) Admin can add books of his/her choice or add books requested by the students.
10) Students can request for the issue, renew and these respective pages will be redirected after the admin's response.
11) Students can recommend books in the recommend section by entering the details.
12) Students can search books in the search book section by entering details and after hitting search a page will appear with book details if the book is present.
13) Both student and admin can logout from their respective pages by hitting the logout button and they will be redirected to the registered page.

5.4 DBMS Platform

This software can be implemented with ease on Linux based operating software systems, Windows OS and Mac OS.

An open-source cross platform web-server should be installed to create a database (ex: Xampp)

Which can have the following system requirements:

- Windows 2008 or later
- Mac OS X 10.6 and above
- CentOS, Ubuntu, Fedora, Gentoo, Arch, SUSE
- Hard disk space: 50GB, RAM: 512MB
- Processor: Pentium® Dual-core CPU and above

5.5 Security and Availability

The software will,

- Authenticate each user based on type of user (which are student and admin).
- When a user performs an unauthorized action, it will display an error message saying it is an unauthorized access.

5.6 Distribution

The master database is the SQL server primary configuration database. It contains information about all the databases on the server, including physical database files and their location. The master database file also contains SQL server configuration settings and login account information.

Components in Master Database:

- Registrations and Remote Logins

- Local Database and Database files
- Login Accounts
- Server Configuration Settings
- Processes and Locks

A current backup will be kept updating it from time to time as it is critical to any server recovery.

Integrity standards are very high and the privacy is kept for the user, no authorized information will be disclosed.

5.7 Backup and Restore Operations

The user information will be kept private for safety and security issues and will not be disclosed to any other third-party organizations so that user privacy is intact and information is safe.

The data is backed up on a regular basis so the data will not be lost if database crashes or any other harm which leads to loss of data. Also, as a safety measure, the data is stored on a private storage so it can't be accessed from outside. For detailed information on actions in backup please refer to the 6.16 module below.

5.8 Maintenance

Maintenance includes modifications in the software product after it is delivered. Automatic logging and error reporting techniques, automatic error message generation.

5.9 Performance and Availability Decisions

The developed DBMS software needs to be able to output information effectively when needed and save the data without waiting time to avoid problems. Several factors that affect performance are the need for adequate system resources and the basic requirements for the software to run successfully.

Software availability is sufficient for users and some features are limited to users due to the risk of data loss and data protection issues. The database can only be edited by an administrator to change the data as needed.

6. DATBASE ADMINISTRATIVE FUNCTIONS :

6.1 Database Identification

Element	Element Name	Description
db_name	library	Library is the name of the database when it is originally created.
db_path	C:\xampp\mysql\data\library	The full path to where the database is stored on the system.
db_location	C:\xampp\mysql\data\library	Location where the database is stored

6.2 Schema Information:

In Entity Relation Schema,

Entities:

1. Students
2. Books

Relations:

1. Issue
2. Return request
3. Renew request
4. Issue request

Attributes:

1. Student
   - Id
   - Fname ❑
   - Lname ❑
   - Branch ❑
   - Email ❑
   - password
2. Books:
   - Bookid
   - Book name
   - Book author
   - Copies

6.3 Schema Description:

Here in student relation, user id is the primary key and in books relation book id is the primary key. In issue relation, user id and book id are foreign keys acting as primary keys at the same time. For issue_request relation, user id and book id are foreign keys acting as primary keys at the same time. For return_request relation, user id and book id are foreign keys acting as primary keys at the same time. For renew_request

relation, user id and book id are foreign keys acting as primary keys at the same time.

Cardinality:

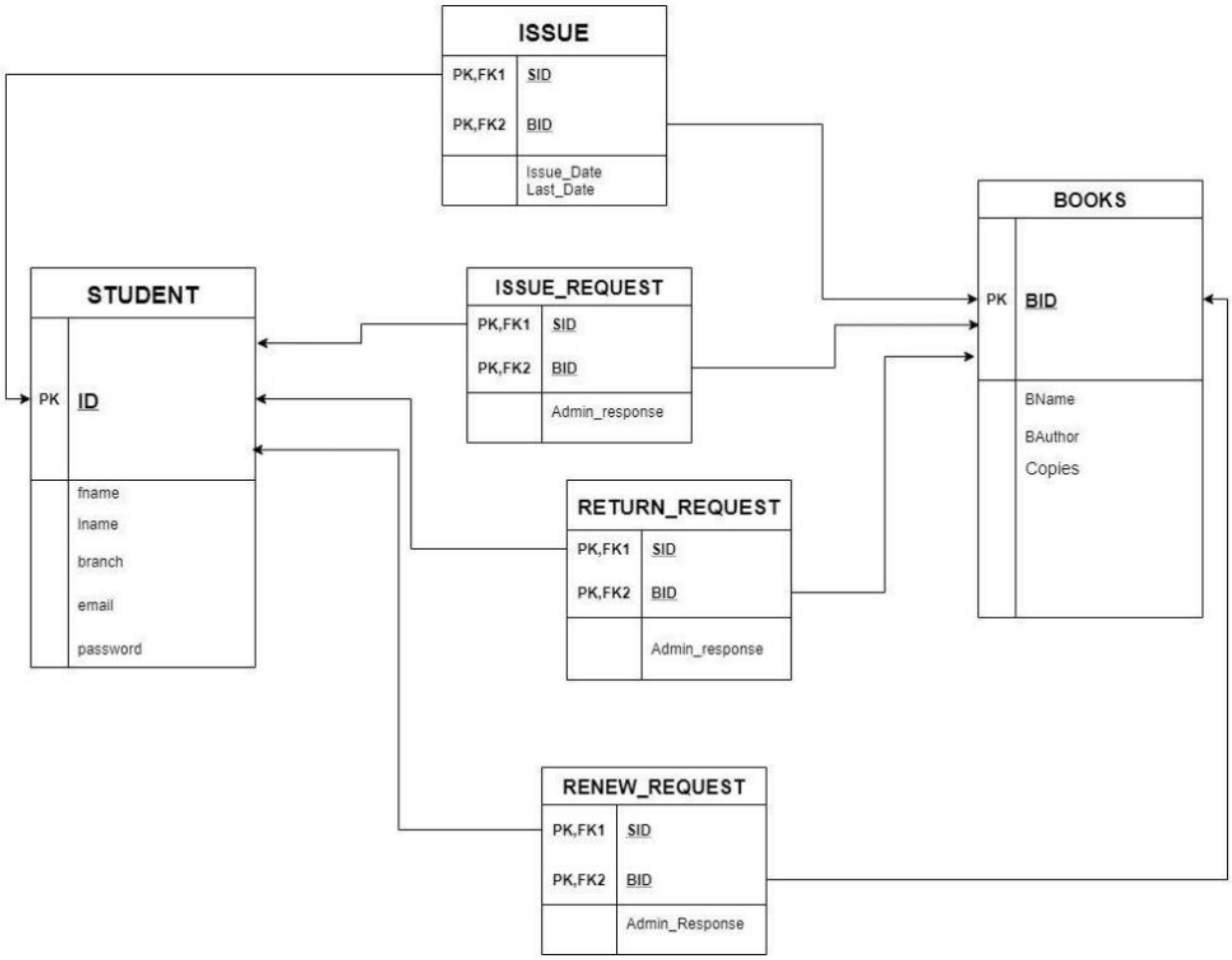
Student-issue-books is m : n

Student-issue-request-books is m : n

Student-return-request-books is m : n

Student-renew-request-books is m : n

Relational Mapping



6.4 Physical structure:

Normalization of database

1. It is a technique used to remove redundancy in the database and to maintain consistency and integrity.
2. It is used to remove certain anomalies in the database to maintain consistency.

The anomalies are:

1. Insertion Anomalies
2. Deletion Anomalies
3. Update Anomalies

All the tables in our database:

Register:

SID	Fname	Iname	Branch	Email	Password

Books:

BID	BName	BAuthor	Copies

Issue:

SID	BID	First_date	Last_date

Issue_request:

SID	**BID**	**Admin_response**

Return_request:

SID	**BID**	**Admin_reponse**

Renew_request:

SID	**BID**	**Admin_response**

All the tables above are in normal form already and the tables are

Normalized up the Boyce Codd normal form

Checking 1st normal form:

As we can see all our tables are already in normal form as in each table there are no redundancies, so the table is in **1NF**

Checking for 2nd normal form:

And in register and book table there is no composite key so no transitive

Dependencies and in other tables date completely depend on the two key attributes so no partial dependencies.so table is in **2NF**

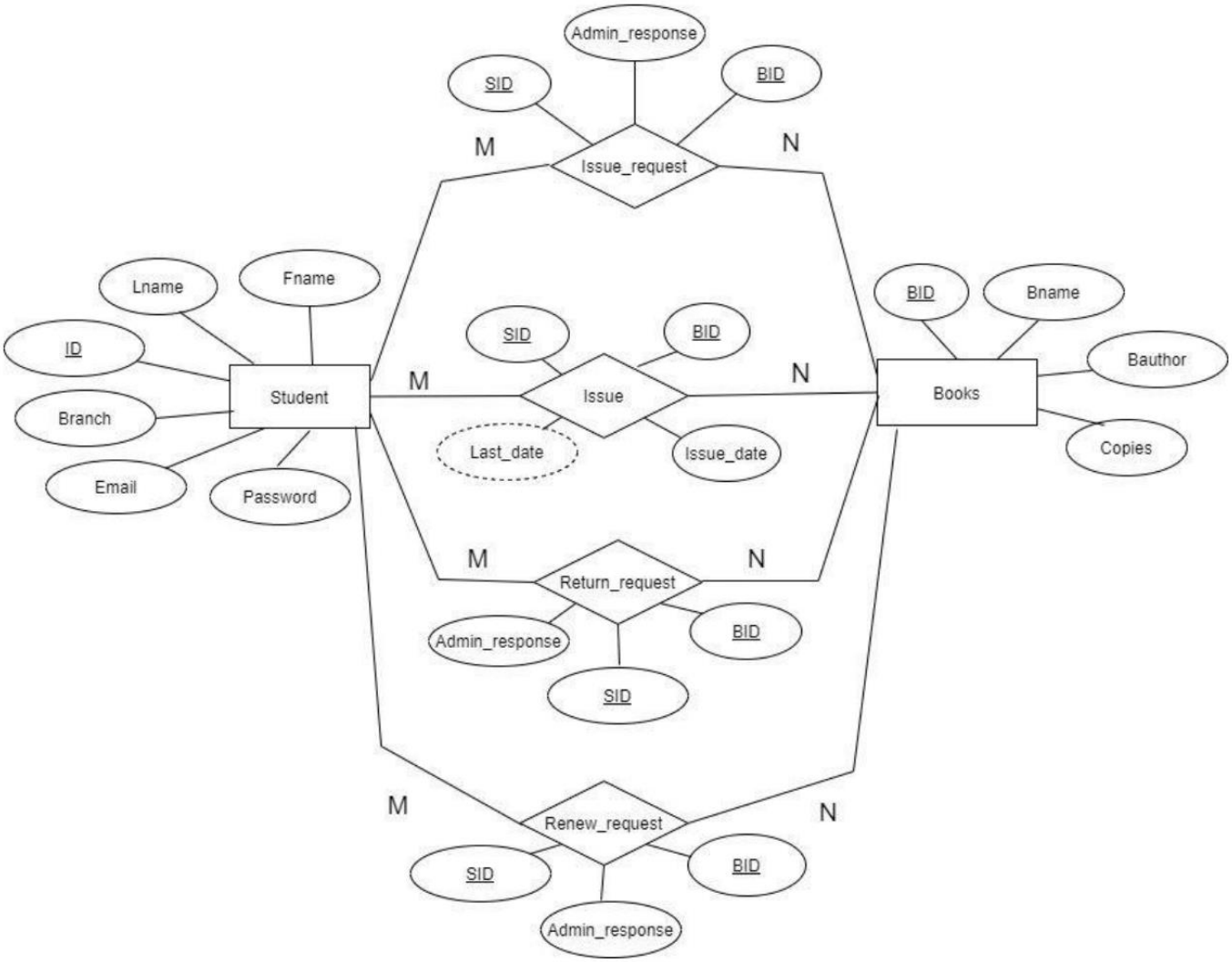
Checking for 3rd normal form:

And in tables register and Books and issue all non-prime attributes depend on and prime attributes and in other three tables there is only one non-prime attribute, so tables are in **3NF**

Checking for Boyce Codd normal form:

Since all the super keys in each of the tables determine tuples uniquely all tables are in **BCNF**.

6.5 Entity Mapping:



6.6 Mapping Rules:

Rules for mapping entries into tables:

Data type of the entry should be the same as the data type of the column

For example: a column named id(int) will accept only integer data types

1. The length of any accepted value will be the same as the length defined for it; no additional characters or numbers will be inserted into the table.

Example: in table we defined name (varchar) of length four and suppose entry contains a name “xyzru” only xyzr will be mapped into table and “u” is ignored.

2.on the backend in php or any other language the values to which the names on the front end are inserted should be the names of tables and should not be different.

Example: Name is inserted into username in php then all the values entered for Name will be inserted into username

3.All values should be entered correctly in their provided space.

6.7 Operational Implications

Refresh:

When the page is refreshed, it will redirect to the same web page.

Update:

1. As there is no separate link provided for updating user’s data there is no option for updating from the student side.

2. An admin can update the details of books the changes will be reflected in the database

3. An admin adds new books, updating the data.

4. After updating by admin page returns to page with no entries of book.

Data transfer:

1. Student registers himself by entering appropriate details.

2. After entering details when he clicks register his details will be stored in the database in the table allotted for storing students' details.

3. Students will be redirected to the login page.

4. After logging in, the student will be redirected to the student page.

5. Where he can request for book, request for issue of book return book by selecting respective options and he will be redirected.

6.8 Backup and Recovery

For backup in case server fails or pc is damaged a backup shall be created by following steps:

1. Go to the root directory where the xampp folder is located.

2. Go to the mysql folder in the xampp folder.

3. Go to bin folder

4. Go to sql.

5. Copy the path.

6. Open the terminal and go to the directory where xampp is located.

7. Paste the path.

8. Type the command "mysqldump -u root -p portal>backup.sql"

9. Execute the command

10. A backup will be created.

11. Save the backup file

12. For recovery start xampp server open mysql/php my admin

13. Go to export select the backup .sql file and click export

14. Data will be recovered

6.9 Applications/Systems Using the Database

The database library created and accessed through phpMyAdmin is used only by web application “library management system” there is no specific version for it.

6.10 Relationship to Other databases:

Our application has only required only one database so there can’t be any relationship with other databases.

7. Detailed Database Design

7.1 Data Software objects and resultant data structures

In detailed database design, the data type of the attributes, their lengths and primary keys and foreign keys are described

For register page:

We need to store the user’s Id, first name, last name, branch, email, password.

User id:

- Primary key
- data type: Varchar
- length:25

Fname:

- Data type: varchar
- length:25

Lname:

- Data type: varchar
- length:25

Branch:

- Data Type: varchar
- length:25

Email:

- Data type: varchar
- length:250

Password:

- Data type: varchar
- length:25

For Books page:

For books, we need to store book names, book id, book authors, no. of copies.

Book id:

- Primary key
- Data type: varchar
- length:25

Book name:

- Data type: varchar
- length:250

Book author:

- Data type: varchar
- length:250

Copies:

- Data type: int
- length:25

For issue/renew/return request:

For issue/return/renew request, we have user id and book id as primary keys and foreign keys.

PK, FK1: User id

PK, FK2: book id

For recommendation:

For recommendations, we will have user id and book id as primary keys and foreign keys.

PK, FK1: User id

PK, FK2: book id

8. APPENDIX

Work done by

MUKKALA KEERTHI SAI NANDAN REDDY

GUVVALA SHANMUKH SHASHANK REDDY

YENDLURI JAHNAV

VATRAPU RAMI REDDY

KAKARA KARTHIKEYA

KALINGIRI SAI MANIKANTA

SHREYAS S N

ABHINANDH K R

NIDHIN S

POTHULA MATTHEWS