# Chapter 1: Introduction

A library management system is a web-based application designed to maintain, organize, and handle countless books systematically. It can track the numbers of books in the library, issued books, and returned books. If the due date is past, it has the function of charging the fine. The users can find the books in an instant and even reserve them through this system. This will help eliminate repetitive manual work and minimize the chance of errors and the immense use of paperwork. Earlier, the librarian used to manage the whole work in manual mode in the form of files and record books. Also, the process of adding new books, new users, issuing, and returning books had to be managed in a manual manner, which is very slow and inefficient. This application resolves this problem and provides a better solution. Users can access the system from any location with an internet connection to search for the required books. This web application can track the books taken by the user and their due date for return. And the users will also get notifications about the new arrival of books and about returning books.

* 1. **Background**

A library management system is software that is designed to manage the functions of a library. It helps librarian to maintain the database of new books and the books that are borrowed by members along with their due dates. This system completely automates all your library's activities. This system is about managing the books in systematic and efficient manner so that borrowing and issuing of books becomes easier. So, we proposed this project in order to help the people of specific schools and colleges of Kathmandu valley.

* 1. **Problem statement**

In case of manual search of books, it experiences high cost of time loss for both librarian and user. Keeping of manual record may cause loss of record in case of any physical or natural harm. Managing records of incoming and outgoing of books manually is very tedious. Users get bad experience visiting library in search of same book repeatedly. Users have to loss their precious time just to know that the library doesn’t have a book that the user is seeking for.

* 1. **Objectives**

Due to the problem in the manual library management, we propose this system with the following objectives:

* To reduce the time required to maintain library functions manually by making it computerized.
* To search the books easily with easy update and adding in database.
* To save the user data so that they do not have to verify each time to access library.
  1. **Scope and Limitations**

# The "Library Management System" project aims to revolutionize the way user’s access and navigate their learning environment. It encompasses the following key areas:

1. Provide the list of books the users can borrow: Users do not have to wait for a long time to find the book they want to borrow the system inbuild search function get the job done within no time.
2. Facility to reserve books that are available: Users do not have to visit library manually they can easily reserve book that are available in library via. Online.
3. A status page for all users to view books borrowed by them, their individual due dates and their individual penalties if any.
4. Providing interface to add or delete books: The system allow admin to add or delete the books according to the requirement of user.

**Limitations:**

# Coverage of Educational Institution: While the project will aim to include a wide range of colleges and universities, it may not cover every educational institution outside Kathmandu valley.

1. Real-Time Updates: Users may not be updated by librarian or admin regarding the arrival of new books in library.

**Chapter 2: Background study and literature review**

* 1. **Background Study**

Throughout history, library has evolved from manually managing to the integration of technology. The project acknowledges the impact of this evolution on how librarian or admin can manage book and user’s access the books.

In the present situation, librarian as well as users have to face different difficulties in managing library and accessing the books. The project recognizes the need for simplifying managing and accessing of books.

* 1. **Literature review**

Before we build our project, we first analyze other existing systems.

* + 1. **Destiny Library Manager**

Library management system that you can access 24\*7 from anywhere anytime. It can also build a various print and digital resources by keeping in mind every kind of students. not only for students the destiny library system also provides customizable tools, extensive collection of books for teachers also. It can be locally installed as well as Follet-hosted.

* + 1. **Blackboard Library Management:**

It is one of the library management systems used in Nepal. It has the most of the exiting features. dashboard that displays a wide range of information about the operations of the school library. The system organizes books by the level of reading, determines the reading level of each book, determines the genre for each book. The library management system allows students to borrow books from the library when they need them. System has a history with books in the library starting with writing development and other inventions like paper and book printing.

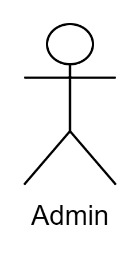
**Chapter 3: System analysis and design**

**3.1 System Analysis**

**3.1.1 Requirement Analysis**

**I. Functional Requirements**

**Library management system**

****

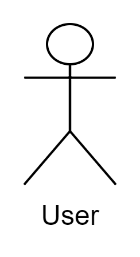
****

Fig: Use case diagram for library management system

List of functional Requirement

|  |  |  |
| --- | --- | --- |
| Requirement Id | Functional Requirement | Description |
| FR1 | Add Book | Allow admin to add books, |
| FR2 | Book Search | Allow admin as well as user to search book, |
| FR3 | Book Booking | Allow user to book books, |
| FR4 | Book Category | Allow admin to add category of books, |
| FR5 | Book Issue | Allow admin to issue book to users, |
| FR6 | Login | Allow users as well as admin to enter into system, |
| FR7 | Sign Up | Allow user to register into system |
| FR8 | Book List | Allow admin and users to view all the books in library. |

**II. Non-Functional Requirement**

Following are the non-functional requirement of library management system:

* **Performance**: This project is for the specific college or school which does not contain large amount of data. So, our system is fast.
* **Security:** In order to get access into the system, Users should first login and then only they are allowed to operate.
* **User-friendly**: Our design is optimal and optimized so, end user can easily understand and use the interface.
* **Scalability**: We can later on expand the project and make it online. We use Laravel framework so that we can add module and expand further**.**

**3.1.2 Feasibility test**

**I. Technical:**

HTML/CSS is used for frontend and Laravel is user for backend, MySQL for database.

Hardware to operate like laptop and mobile with web browser installed.

**II. Operational:**

Our project can solve basic operation performed in library like adding book, issuing books, listing books, check availability of books. So, it is operationally feasible.

**III. Economic:**

We use our own resources like (net and laptop) to propose the project. So, it is economically feasible.

**IV. Schedule:**

Our system was proposed within the time frame.

**3.1.3 Data Modeling**

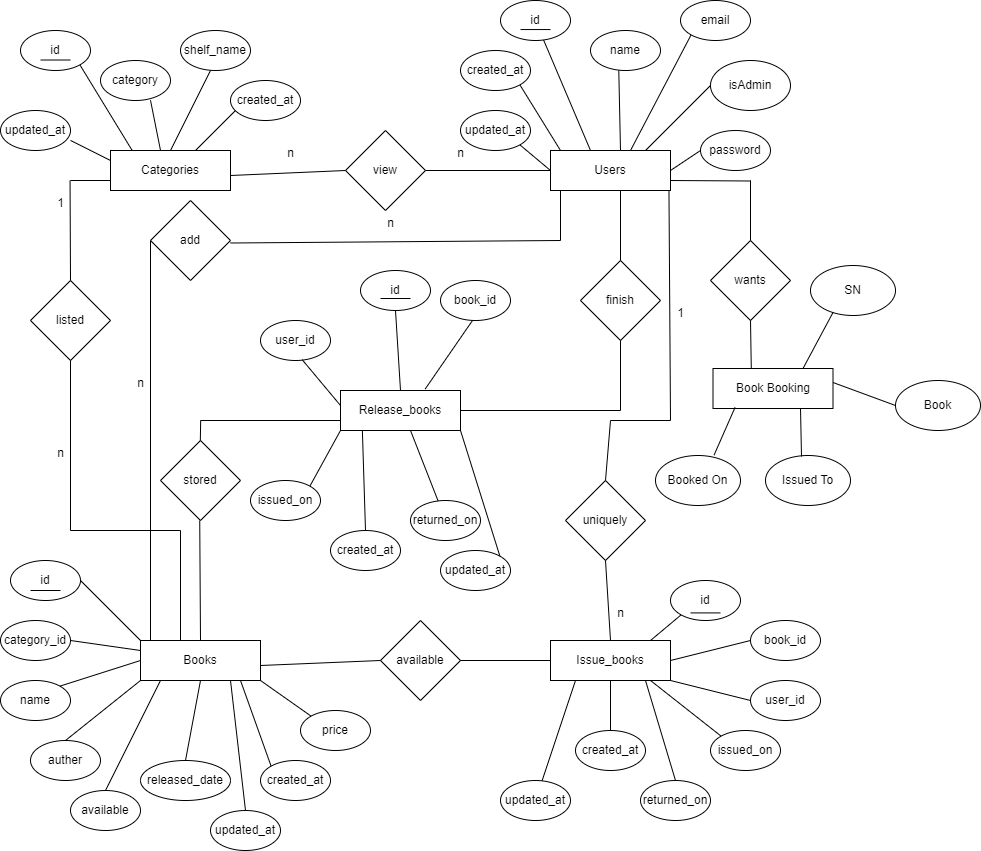
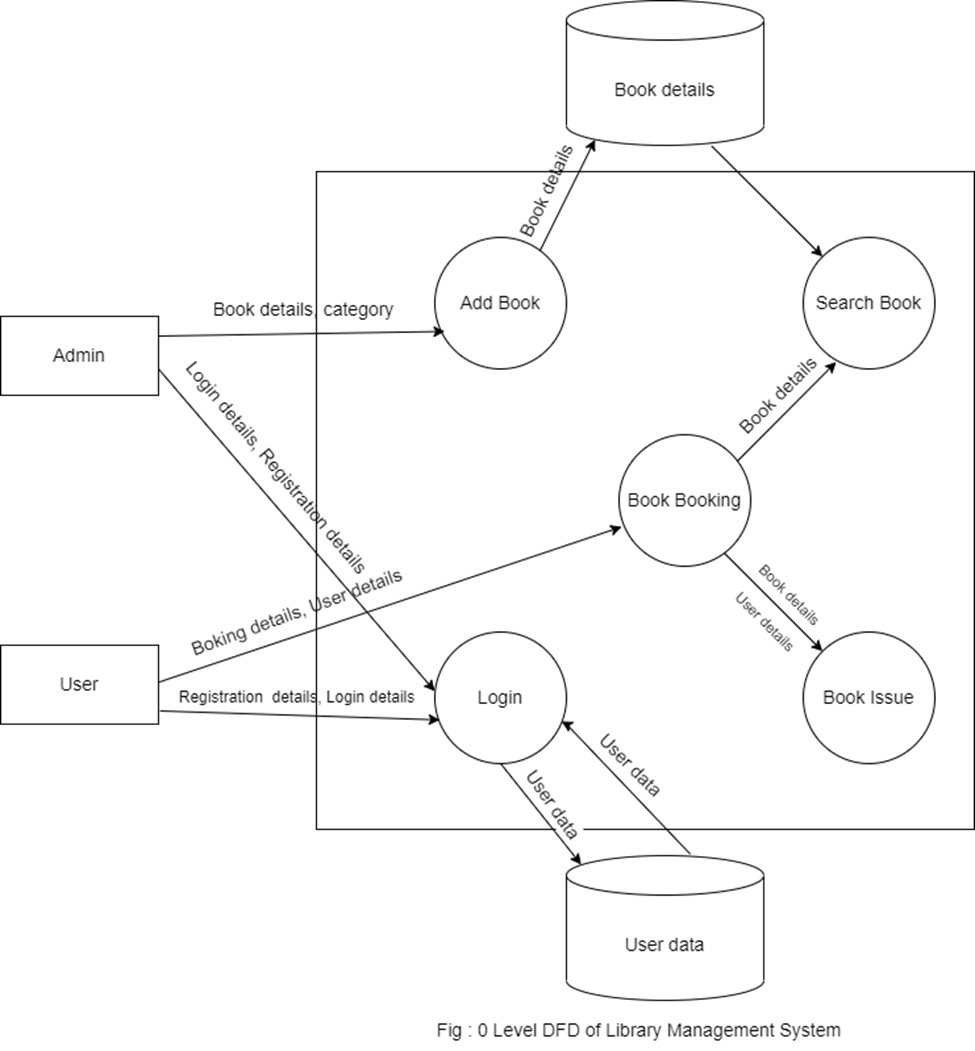


Fig: Data modeling for Library management System

**3.1.4 Process modeling**



Fig: Context level Diagram for Library management system



* 1. **System Design**
     1. **Architectural design**

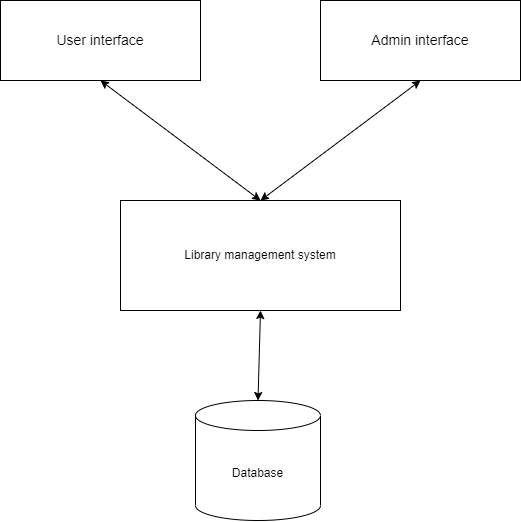
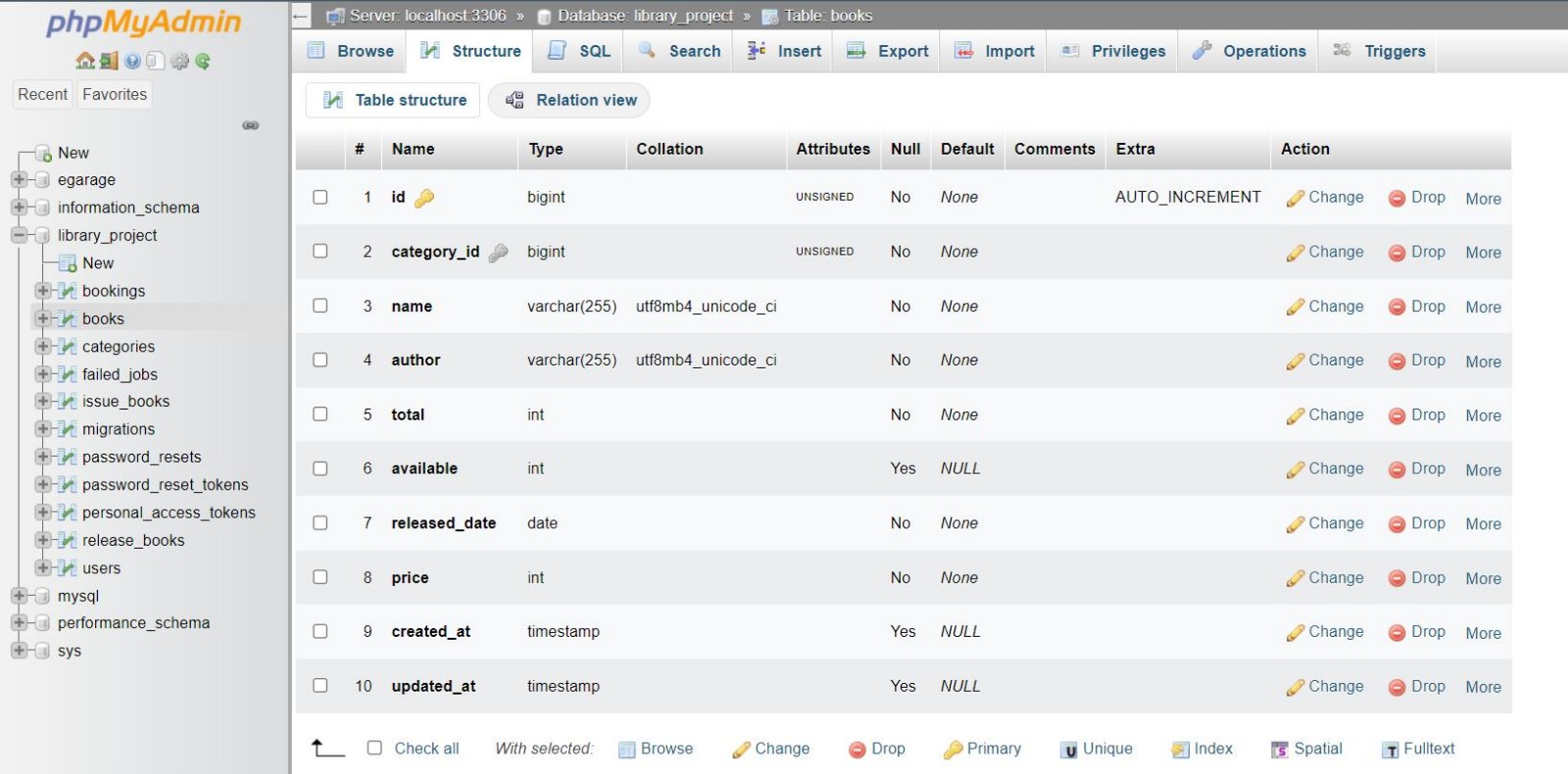
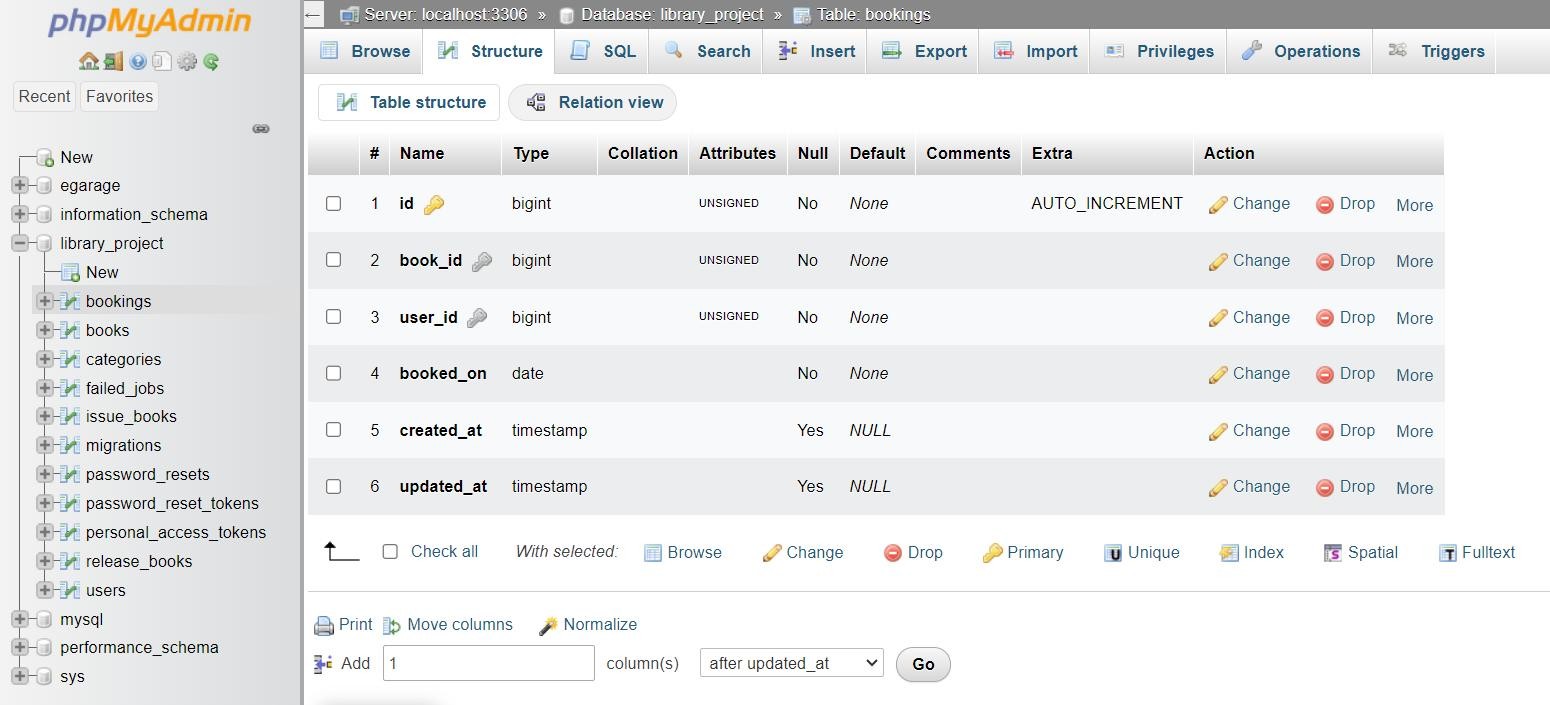
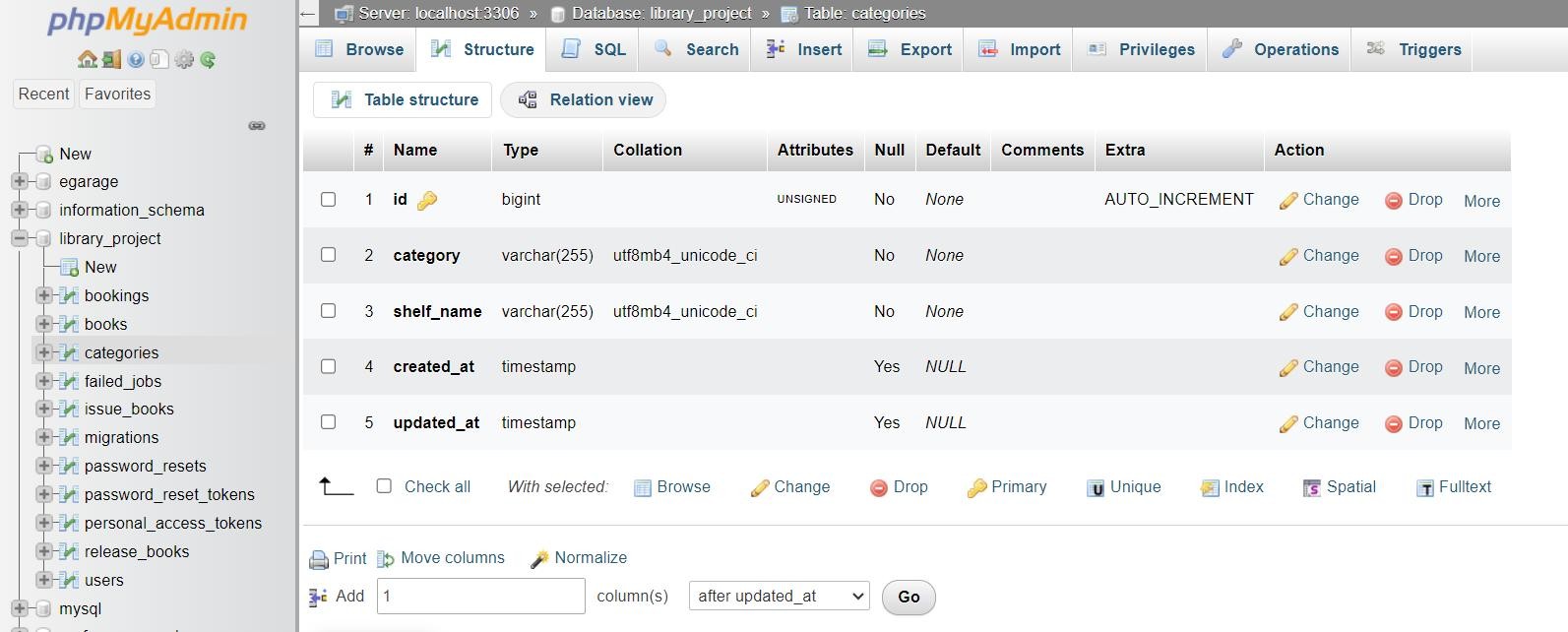
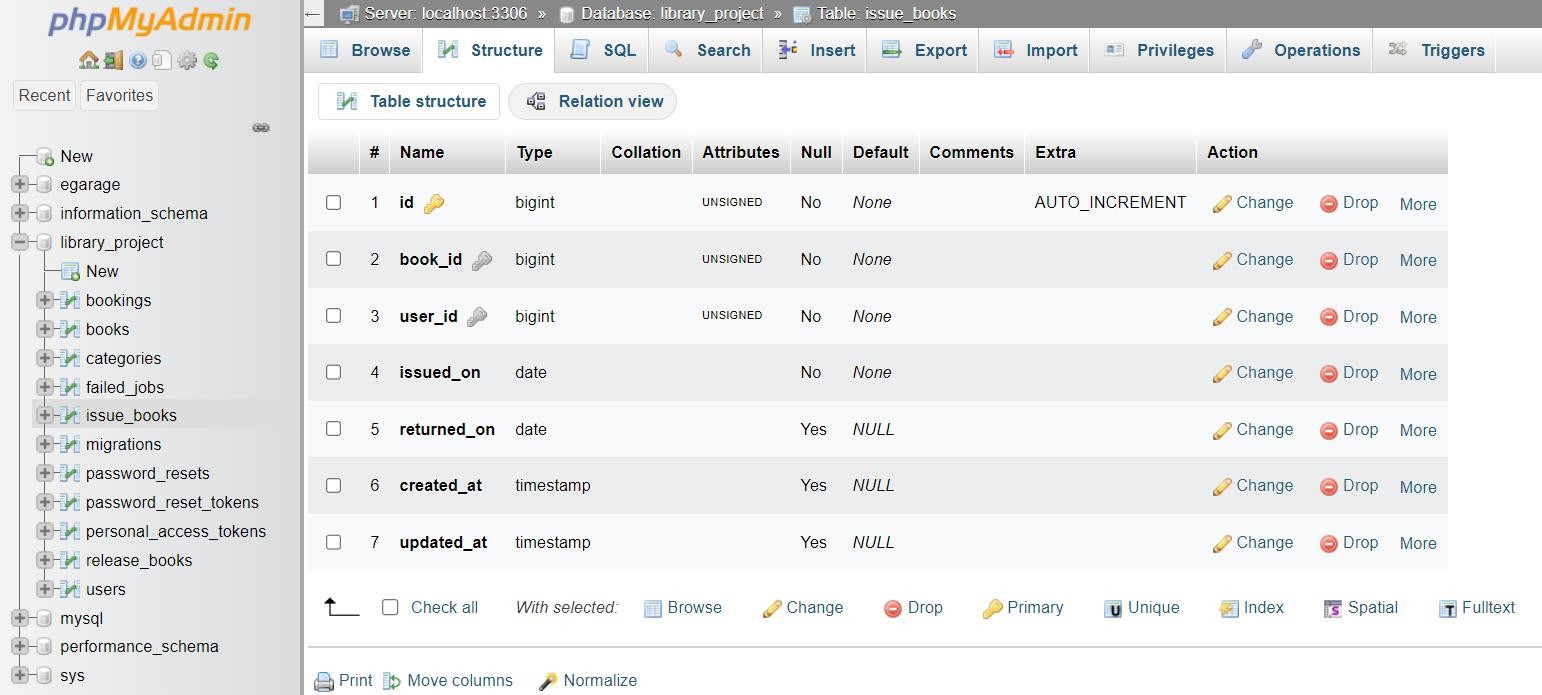
****

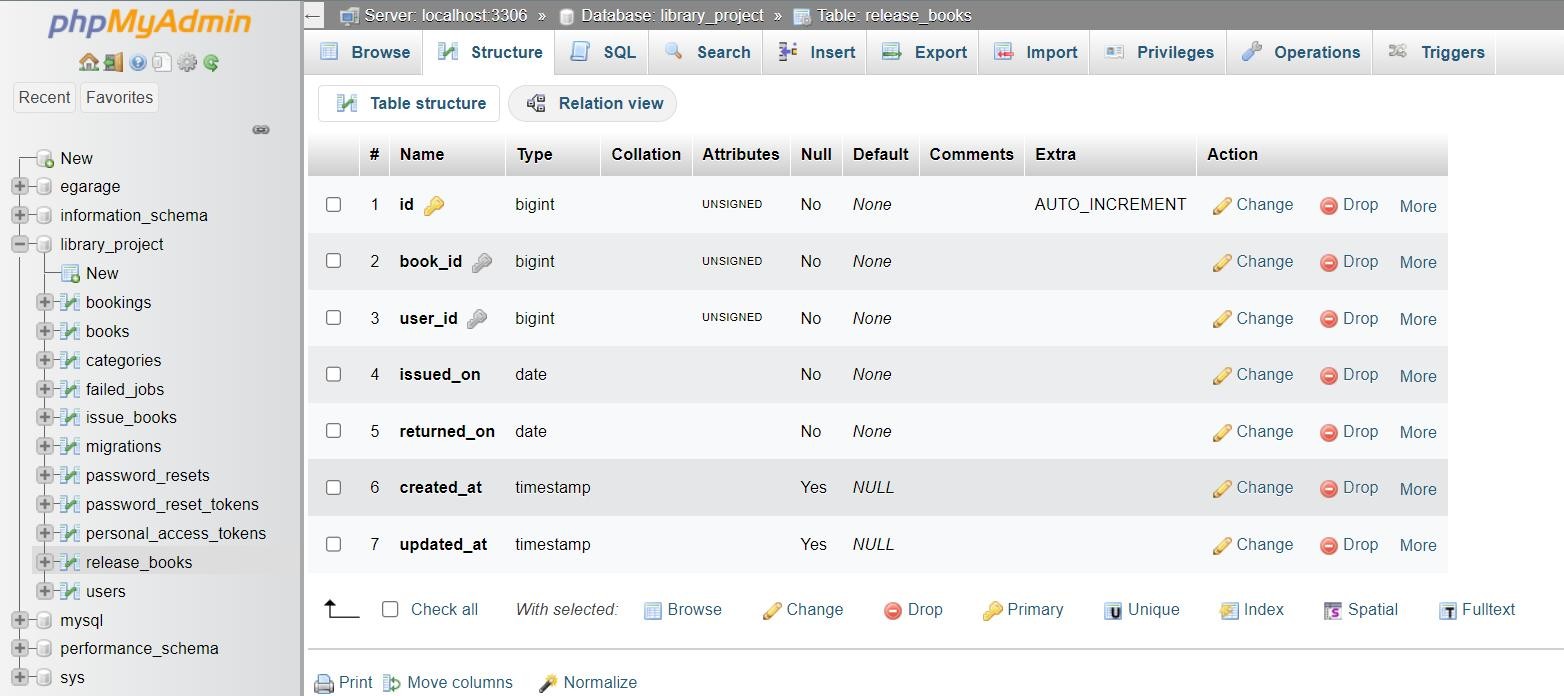
Fig: Architectural design**3.2.2** **Database Schema Design**

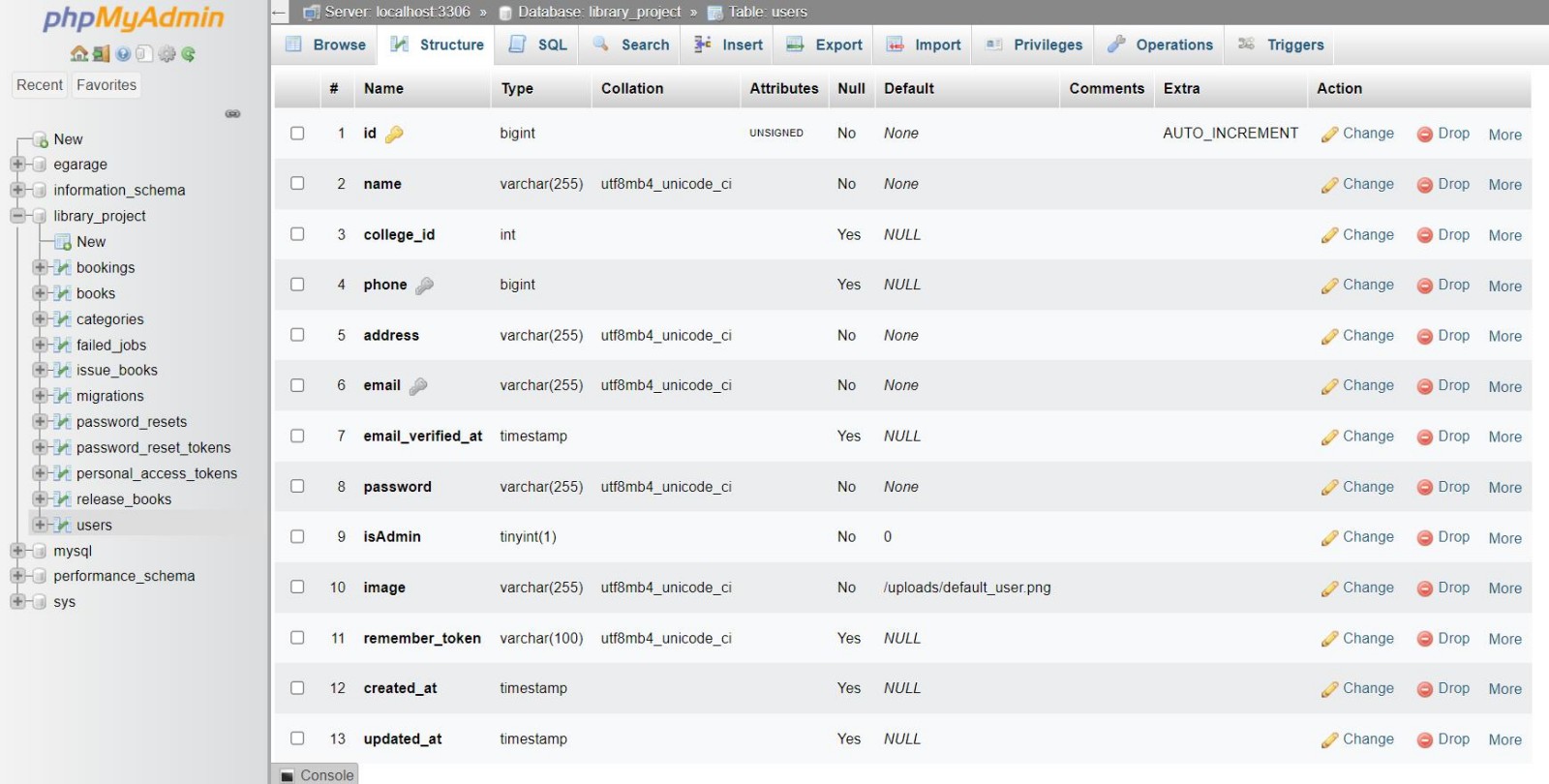












* + 1. **Interface design**

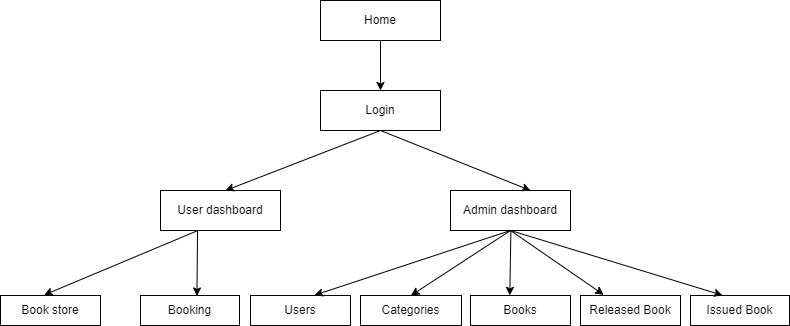
****

Fig: Interface Design of Library management system

* + 1. **Physical DFD**

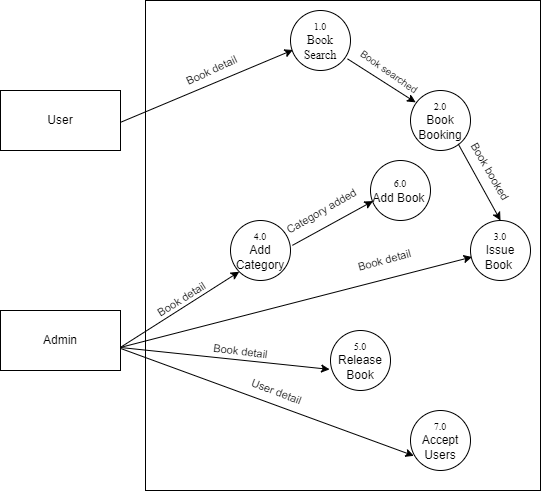
****

Fig: Physical DFD

**Chapter 4: Implementation and Testing**

**4.1 Implementation**

**4.1.1 Tools Used**

**I**. **draw.io:** we use draw.io in order to create the context diagram, ER diagram, DFD, system design, etc.

**II.** **VS code:** We prefer using visual studio to code the program because of its flexibility and ease.

**III.** **Programming language:** we use php language to perform backend operation and Laravel as a framework.

**Database platform:** MySQL is used as a database to store the information regarding books details, user details, etc.

* + 1. **Implementation details of module.**

**Login:** This is the landing page of our system. In this module both Admin as well as user should first fill-up their details to get into the system.

**Registration:** The users should register first to get access into the system.

**User dashboard:** This is the user dashboard, when they login as a user they are directed into this module. In this module, user can search books, view books and book books.

**Admin dashboard:** This is the admin dashboard, when they login as an admin they are directed into this module. In this module, Admin can add users, add books according to category, view books, issue books to users, release the issued books, view user details and report on issued and returned books.

**Book search:** This module provides facility to both users and admin to search the available books in library.

**Book booking:** Users can interact with this module, they can book the available books in the library.

**Category:** Only admin can manipulate this module. Admin have authority to add the categories along with the shelf no. So, books can be managed in a systematic fashion.

**Book:** In this module admin have the authority to add books in library with respect to the category available.

**Release book:** This module is controlled by the admin. Admin can see and update the books that are released by the users.

**Issue book:** This module is only for the admin. Admin have the authority to issue book to the users with the help of identification details.

* 1. **Testing**

**4.2.1 Test cases for unit testing:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Test case Id | Test cases | Input test data | Expected outcome | Result |
| **TC\_1** | **Test case for login with correct user name and wrong password** | **Username:** [**admin@test.com**](mailto:admin@test.com)  **Password:1234567** | **These credential do not match the record** | **fail** |
| **TC\_2** | **Test case for login with correct user name and correct password** | **Username:** [**admin@test.com**](mailto:admin@test.com)  **Password:password** | **Redirect to the admin dashboard** | **Pass** |
| **T**C\_3 | **Test case for login with correct user name and correct password** | **Username:** [**u**ser**@test.com**](mailto:user@test.com)  **Password:password** | **Redirect to the user dashboard** | **Pass** |
| **TC\_4** | **Test cases for adding category by the admin** | **Category: Novel**  **Shelf\_name:N1** | **Category added successfully** | **Pass** |
| **TC\_5** | **Test cases for adding book by the admin** | **Name: Harry potter**  **Author: Jk Rowling**  **Released\_date:** **2023-08-06**  **No of copies:3**  **Price:450** | **Book added successfully** | **Pass** |

* + 1. **Test cases for system testing**

Add book by the administrator.

|  |  |
| --- | --- |
| Steps | Description |
| Step 1 | Admin Login, Navigates to Admin dashboard |
| Step 2 | Select the correct category. Category is selected |
| Step 3 | Input the book title, author name, release date, availability, price. |
| Step 4 | Click the submit button. Book is added and saved in database |
| Step 5 | View the list of books clicking the book list button. |

Book booking by the user

|  |  |
| --- | --- |
| Steps | Description |
| Step 1 | User Login, navigates to user dashboard after the user is verified by admin. |
| Step 2 | Select the correct category. Category is selected |
| Step 3 | Input the book title, author name, release date, availability, price. |
| Step 4 | Click the submit button. Book is added and saved in database |
| Step 5 | View the list of books clicking the book list button. |