Library Database Management System
Database Requirements

Version 1.0

Table of Contents

Database Requirements

1. Introduction

1.1 Project Overview

The Library Database Management System project will deliver an efficient, user-friendly, and secure database system to support small library operations. The DBMS will enable seamless management of loanable items (books, digital media, magazines), enforce borrowing rules based on diverse membership categories, and generate insightful reports. By integrating modern database design principles, the system will streamline item tracking, membership management, and financial oversight.

1.2 Scope

The Library Database Management System (LMS) project will design and implement a relational database for a small library to manage loanable items, memberships, borrowing rules, and generate reports. It will model entities like books, digital media, magazines, and clients, and enforce constraints like borrowing limits and fees based on membership type. The system will include features for managing loans, returns, and client accounts, with user interfaces for both staff and clients. Advanced queries will generate financial and activity reports, while concurrency and transaction management will ensure seamless multi-user operations. The LMS will be developed through domain modeling, database design, and implementation phases, ensuring functionality and data integrity.

1.3 Glossary

LMS - Library Management System
DBMS - Database Management System

2. Stakeholders

This section outlines the requirements and expectations of the various stakeholders involved in the development and implementation of the library database management system (LDBMS). Each group of stakeholders has distinct needs and responsibilities, which must be addressed to ensure the system's success.

2.1 Library Administrators

- **System Maintenance and Support:** IT administrators need tools for managing the backend of the system, including system upgrades, database maintenance, and data backups. The system should allow easy configuration of software updates and patches.
- **Access Control:** Administrators should have control over user permissions, allowing them to restrict or grant access to different parts of the system depending on the user's role (e.g., patron, staff, administrator).
- **Data Security:** The system must comply with data protection regulations, ensuring that patron and staff data are encrypted, secured, and backed up regularly. It should include user authentication measures (e.g., two-factor authentication).
- **Performance Monitoring:** The system should offer real-time monitoring of system performance and usage metrics to help IT staff optimize server performance, ensure uptime, and handle system load effectively.

2.2 Library Staff

- **Book Cataloging and Management:** The system should allow staff to efficiently catalog books and other resources, including adding new materials, updating existing records, and deleting outdated or missing items. It must support a wide variety of resource types (books, journals, e-books, etc.).
- **User Account Management:** Staff should be able to create, modify, and manage patron accounts, including personal details, borrowing history, and active loans. There should be clear roles and permission associated with different user types (e.g., students, faculty, public members).
- **Circulation and Inventory Control:** The system should support check-ins, check-outs, renewals, and reservations. Inventory management features should notify staff of overdue items, misplaced items, and low stock levels of high-demand resources.

- **Reporting and Analytics:** The staff requires access to reporting tools to generate insights about circulation trends, overdue materials, fines, and popular titles. These reports should be customizable and exportable.

2.3 Library Patrons

- **Search and Discovery:** Patrons need to search for books, journals, and digital resources using various filters such as title, author, subject, and availability. The system should provide an intuitive and user-friendly interface for this purpose.
- **Account Access:** Patrons should have secure access to their accounts to view check-out materials, due dates, overdue fines, and borrowing history. They should also be able to update personal information and reset their passwords if necessary.
- **Self-Service Features:** Patrons should be able to place holds on items, renew borrowed materials, and pay fines online without requiring assistance from library staff.
- **Notifications and Alerts:** The system should send automatic notifications to patrons regarding due dates, overdue items, reserved item availability, and any outstanding fees.

3. Functional Requirements

3.1 Functional Requirements:

1) User Administration:
    a) Manage library staff and clients:
        i) Differentiate each role and the level of access they are granted.
        ii) Allow access to specific data upon request.
    b) Maintain user account information:
        i) Create new users.
        ii) Delete existing users.
        iii) Update information on existing users.
2) Data Entry, Updates, and Deletion:
    a) Allow library staff to add, update, and remove items including books, digital media, magazines.
    b) Store details for each item including title, author, ISBN, publication year, genre, availability status, etc..
    c) Track/change availability of items in real time.
3) Retrieval:
    a) List available books, digital media, and magazines.
    b) Generate reports on overdue items and associated fees.
    c) Track the number of items loaned by membership type.
    d) Identify the most popular items.
    e) Retrieve client borrowing histories and outstanding fees.
4) Report Generation:
    a) Monthly summary of items loaned, fees collected, and most popular items.
    b) Inventory report highlighting availability and inactive items.
    c) Overdue items report, showing late fees and responsible clients.
    d) Client activity reports showing borrowing history, fees, and reservations.

3. 2 Non-Functional Requirements:

1) Performance:
    a) Ensure queries are performed in a reasonable time.
    b) Ensure reports are generated on time.
    c) Ensure the system can handle an increase in data (extra storage).
2) Security:
    a) Only allow access to administrative functions to administrators.
    b) Protect private data from non administrative users.
    c) Implement an authentication system to determine if the user is an admin.
3) Usability:
    a) Create a user-friendly interface to allow easy access.
    b) Design a system that can work between all devices and platforms.
    c) Create a consistent layout to avoid confusion and mistakes.

4. Data Requirements

4.1 Items

a) Entity: Item
   i) Item is a total generalization of book, magazine, and digital_media. Items can be borrowed, renewed, returned, put on hold, or have a hold removed. There can be multiples of the same item. Administrators can add, update, and remove items. Item.ID can be an ISBN (book), ISSN (magazine), or a DOI (digital media).

b) Attribute:
   i) ID - int()
   ii) Price - numeric(5, 2)

4.2 Books

c) Entity: Book
   i) Book is a specialization of Item.

d) Attributes:
   i) ISBN - int()
      (1) Foreign key (ISBN) references Item
   ii) Author - varchar(20)
   iii) Publisher - varchar(20)
   iv) Subject - varchar(20)
   v) Publish date - char(10)

4.3 Magazines

e) Entity: Magazine
   i) Magazine is a specialization of Item.

f) Attributes:
   i) ISSN - int()
      (1) Foreign key (ISSN) references Item
   ii) Publisher - char(20)
   iii) Edition - int()
   iv) Publish_date - char(10)

4.5 Digital media

g) Entity: Digital_media
   i) Digital media is a specialization of Item.

h) Attributes:
    i) DOI - int()
        (1) Foreign key (DOI) references Item
    ii) Media_type - varchar(15)
    iii) Release_date - char(10)
    iv) Title - char(20)
    v) Creator - char(20)

4.6 Members

i) Entity: Member
    i) Members can borrow, renew, return, place a hold, or remove a hold. Members have a membership type, the domain is yet to be determined. Members will also have a record of their interactions including an overdue_flag attribute, and a history of their fees including a current account balance attribute.

j) Attributes:
    i) Card_number - int()
    ii) Name - varchar(20)
    iii) Address - varchar(30)
    iv) Email - varchar(20)
    v) Phone_number - char(12)
    vi) Membership_type - varchar(15)

4.7 Transactions

k) Entity: Transaction
    i) Transactions consist of a member and an item (book, digital media, or magazine). Transaction types include borrow, renew, return, place hold, remove hold, and pay fee. Transactions will update the status of items and member fees.

l) Attributes:
    i) ID - int()
    ii) Trans_type - varchar(10)
    iii) Date - char(10)
    iv) Card_number - int()
        (1) Foreign key (Card_number) references Member
    v) Item_ID - int()
        (1) Foreign key (Item_ID) references Item

5. User Requirements

5.1 Library Administrator

a) Access Level: Administrator Access; Full access to all system features and functionalities
b) Interface Requirements:
    i) Dashboard for viewing system status: checking overdue items, member activity, loans, etc.
    ii) Managing library inventory: adding/removing books, video CDs, magazines, technology
    iii) Managing Users: adding members, removing members, editing member accounts
    iv) Configuration Settings: being able to backup data, manage database, set user permissions

5.2 Library Staff

c) Access Level: Access is restricted to having the ability to perform daily employee operations.
d) Interface Requirements:
    i) Checking In/Out: tools to check in/out books and other materials for members
    ii) Monitoring Member Activity: being able to view and edit member profiles to update their status like check in/out history
    iii) Notification of Overdue Items: ability to know when a member has overdue materials
    iv) Managing Library Stock: ability to search and update what materials are available and loanable

5.3 Library Patron

a) Access Level: Limited to viewing and managing member profile
b) Interface Requirements:
    i) Dashboard to view due dates, overdue items, and loans/fines
    ii) Renewing: ability to renew materials checked out
    iii) Search Engine: ability to search if desired materials are available
    iv) History: ability to check history of past purchases.

6. Hardware and Software Requirements

The program will be a standalone system written in **[programming language]**. The program will interact with a local MySQL database via one of MySQL's supported connectors.

6.1 Hardware Requirements:

- Development Environment
    - Any computer meeting the following requirements:
        - Recommended 8GB of RAM, minimum 4GB
        - Available Storage of at least 50 GB recommended, minimum of enough to hold our program and database

6.2 Software Requirements:

- Operating System
    - Any (Windows / Mac / Linux)
- Database Management System
    - MySQL
- Development Tools
    - Any Code Editor
    - MySQL Connector
- Programming Environment
    - SQL Client Library for whichever programming language is selected