Vel Tech
Rangarajan Dr. Sagunthala
R&D Institute of Science and Technology
(Deemed to be University Estd. u/s 3 of UGC Act, 1956)

SCHOOL OF COMPUTING

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

MINI PROJECT

Programme	**:**	B. Tech Computer Science & Engineering
Course Code / Course Name	**:**	10211CS207 / Database Management Systems
Year / Semester	**:**	2025-26 / Summer
Faculty Name	**:**	Dr. Krishnaveni.N
Slot	**:**	S7L1
Title	**:**	**LIBRARY AUTOMATION AND BORROWING SYSTEM**
Name of the students	**:**	

1)	B. VENKATANAVEEN	- VTU29571
2)	P. TARUN	- VTU29624
3)	SK. MASTHAN VALI	-VTU29717
4)	SHAIK RASOOL	-VTU29782
5)	P. CHAITRA NAGA SRI	-VTU29807
6)	V. BHARATH KUMAR REDDY	-VTU29817

LIBRARY AUTOMATION AND BORROWING SYSTEM

Problem Statement:

The traditional library management process involves manual record keeping of books, members, and loans, which often leads to inefficiency, human errors, and difficulty in maintaining accurate records. A Library Automation and Borrowing System aims to digitalize these operations—automating book issue, return, and catalog management. The system will support multiple users accessing and updating records concurrently while ensuring data integrity and fast retrieval of information.

Abstract:

The Library Automation and Borrowing System is designed to streamline the process of book management and borrowing in an academic or institutional library. It enables librarians to manage the catalog efficiently, members to search and borrow books, and administrators to track transactions and usage. The system consists of modules for book inventory, member registration, book lending/return tracking, fine calculation, and reporting. It employs a relational database for data consistency, and user interfaces for staff and members to perform CRUD operations easily. The project also focuses on concurrency handling, ensuring multiple users can issue or return books simultaneously without conflict. The backend uses SQL for transaction control, normalization for data efficiency, and supports SQLite for easy integration and portability.

Objectives:

- Automate book issue and return operations.
- Maintain accurate records of books, members, and loans.
- Provide quick search and filter options for books and authors.
- Handle concurrent access in a multi-user environment securely.
- Support CRUD operations and data normalization for efficient storage.

Actors:

- Librarian
- Library Member
- Administrator
- Database System

Primary Use Cases (brief):

- Add, update, or delete books and members.
- Search for books by title, author, or ISBN. Borrow a book and record loan details.
- Return a book and update inventory count.
- Calculate fines for overdue returns.
- Generate reports on library usage and book availability**.**

Typical Success Scenario (Borrow and Return Process):

1. A member searches for a book using title or author.
2. The librarian checks the availability of the book.
3. If available, a new loan record is created and the book is issued.
4. The system decreases the available copies count.
5. When the member returns the book, the loan record is updated with the return date.
6. Any overdue fines are calculated automatically.
7. The system updates the available copies count accordingly.

Functional Requirements:

- Member registration and authentication.
- Book catalog management (Add, Update, Delete).
- Issue and return book tracking.
- Fine calculation for overdue books.
- Search functionality by title, author, or category.
- Report generation for admin users.

Non-Functional Requirements:

- System should support multiple concurrent users.
- Data integrity and transaction consistency (ACID properties).
- Fast response for search and update queries.

- Secure authentication and access control.
- Data backup and recovery mechanisms.

Data Model (Entities):

- Book(Book_ID, Title, ISBN, Publisher, Year_Published, Copies_Available, Author_ID)
- Author(Author_ID, Name, Nationality, Birth_Year)
- Member(Member_ID, Name, Address, Phone_No, Email, Membership_Date) □ Loan(Loan_ID, Book_ID, Member_ID, Issue_Date, Due_Date, Return_Date, Fine)

Architecture (high-level):

The system follows a client-server architecture with a front-end application (Python/Tkinter or Web UI) connected to a SQLite database backend. The server handles requests for borrowing, returning, and searching books. SQL transactions ensure atomicity and consistency. The database schema is normalized to 3NF to eliminate redundancy.

Sequence / Flow (Borrowing Process):

Member requests a book → System checks availability → Loan record created → Book count updated → Book borrowed → Member returns → Loan record updated → Fine calculated if overdue → Availability restored.

Technology Stack (suggested):

- Frontend: HTML, CSS, Python Tkinter (or Flask Web Interface).
- Backend: Python with SQLite / MySQL.
- Database: SQLite3 (for local), PostgreSQL (for production).
- Language: SQL for queries and transactions.

Testing & Evaluation:

- Unit testing for CRUD operations.
- Integration testing for issue/return process.
- Concurrency testing with multiple transactions.

- User acceptance testing with sample library data.

Implementation Plan (Milestones):

- Requirement analysis and database design (Week 1).
- Interface and table creation (Week 2).
- Book and member CRUD operations (Week 3).
- Loan and fine management (Week 4).
- Testing and debugging (Week 5).

- Final documentation and presentation (Week 6).

Extensions & Future Work:

- Integration with barcode/RFID scanners for automation.
- Online portal for members to reserve books remotely.
- Integration with e-book lending platforms.
- Advanced analytics and reporting dashboard for administrators.

Conclusion:

The Library Automation and Borrowing System automates traditional library operations such as book issue, return, and member management. It improves efficiency, accuracy, and user experience while maintaining data consistency and supporting concurrent operations. The modular design allows easy extension to online and mobile applications in the future.
