A Project Report On

Library Management System

Submitted in partial fulfillment of the requirement for the award of the degree

Bachelor of Computer Application

BCA

Academic Year 2025 – 26

Het Padariya **92300527254**

Dharmik Jiyani **92300527173**

Prince Jakhotra **92300527149**

Internal Guide

(Jignesh sir kariya)

Faculty of Computer Applications (FCA)

Certificate

This is to certify that the project work entitled

Library Management System

submitted in partial fulfillment of the requirement for

the award of the degree of

Bachelor of Computer Application

BCA

of the

Marwadi University

is a result of the bonafide work carried out by

Het padariya (92300527254)

Dharmik jiyani (92300527173)

Prince jakhotra (92300527149)

during the academic year 2025-26

Jignesh kariya	*sunil bajeja*	*sridaran rajagopal*
Faculty Guide	**HOD**	**Dean**

DECLARATION

I/We hereby declare that this project work entitled **Library Management System** is a record done by me.

I also declare that the matter embodied in this project is genuine work done by me and has not been submitted whether to this University or to any other University / Institute for the fulfillment of the requirement of any course of study.

Place :marwadi university Rajkot

Date :

Het padariya (92300527254) Signature : _____

Dharmik jiyani (92300527173) Signature : _____

Prince jakhotra (92300527149) Signature :_____

CONTENTS

1.Table index

2.Figure Index

1. SYNOPSIS

Introduction to Project:

The **Library Management System** is a GUI-based mini project developed in **Python (Tkinter)**. It provides an easy-to-use interface for managing library operations such as adding books, registering members, and handling borrowing and returning transactions. Instead of using a heavy database like MySQL or Oracle, this project stores records in **CSV files** (books.csv, members.csv, and borrowings.csv), making it lightweight and portable. The system is designed to simplify routine tasks for librarians and ensure accurate record-keeping with minimal manual effort.

Objective of the System:

The main objectives of the Library Management System are:

1. To provide a **user-friendly interface** for managing books and members.
2. To maintain a **record of borrowed and returned books** efficiently.
3. To **reduce manual work** by automating routine library tasks.
4. To implement a **search and filter mechanism** for quick retrieval of information.
5. To create a **cost-effective solution** using Python and CSV files instead of large databases.

2. PREAMBLE

General Introduction

A library plays a crucial role in educational institutions by providing access to books and resources for students, faculty, and researchers. Traditionally, library tasks such as maintaining book records, registering members, and tracking borrowed or returned books were managed manually using registers. This manual system is time-consuming, error-prone, and inefficient.

Features of the System

1. **User-Friendly Interface:** Designed with Tkinter for simple navigation.
2. **Book Management:** Add, edit, and view book details.
3. **Member Management:** Register new members with their details.
4. **Borrowing & Returning:** Track issue and return of books.
5. **Search & Reporting:** Search for books/members and generate borrowing reports.
6. **Lightweight & Portable:** Uses CSV files instead of heavy databases.

Module Description

1. Book Management

- Allows librarians to add new books into the system with details such as title, author, and availability.
- Updates book status automatically when borrowed or returned.
- Provides a view of all available and issued books.

2. Member Management

- Stores details of library members such as name, ID, and contact information.
- Allows librarians to register new members and update member records.
- Ensures unique identification for each member.

3. Borrowing & Returning

- Facilitates the issuing of books to registered members.
- Tracks the due date and return status of issued books.
- Updates the availability of books once returned.

4. Search & Reporting

- Provides quick search functionality to find books or members.
- Generates reports of borrowed books and their status.
- Helps in maintaining transparency and accuracy in library operations.

3. TECHNICAL DESCRIPTION

3.1 Hardware Requirements:

To run the Library Management System smoothly, the following hardware configuration is recommended:

Processor	Intel Dual Core or higher (i3/i5 recommended)
RAM	Minimum 2 GB (4 GB recommended)
Storage	120 GB SSD
Hard Disk	At least 200 MB free space
Input Devices	Standard keyboard and mouse
Graphics	Integrated graphics are sufficient

3.2 Software Requirements

The project is lightweight and can run on most operating systems with Python installed.

Operating System	Windows 7/8/10/11 or Linux (Ubuntu/window)
Programming Language	Python 3.x
Libraries Used	or GUI design csv → for handling file storage
datetime	for recording issue/return dates
os	for file path handling

File Storage: CSV files act as the database

- **books.csv → stores book records (title, author, availability)**
- **members.csv → stores member details (name, ID, contact)**
- **borrowings.csv → stores borrowing/return transactions**

Development Tools Used:

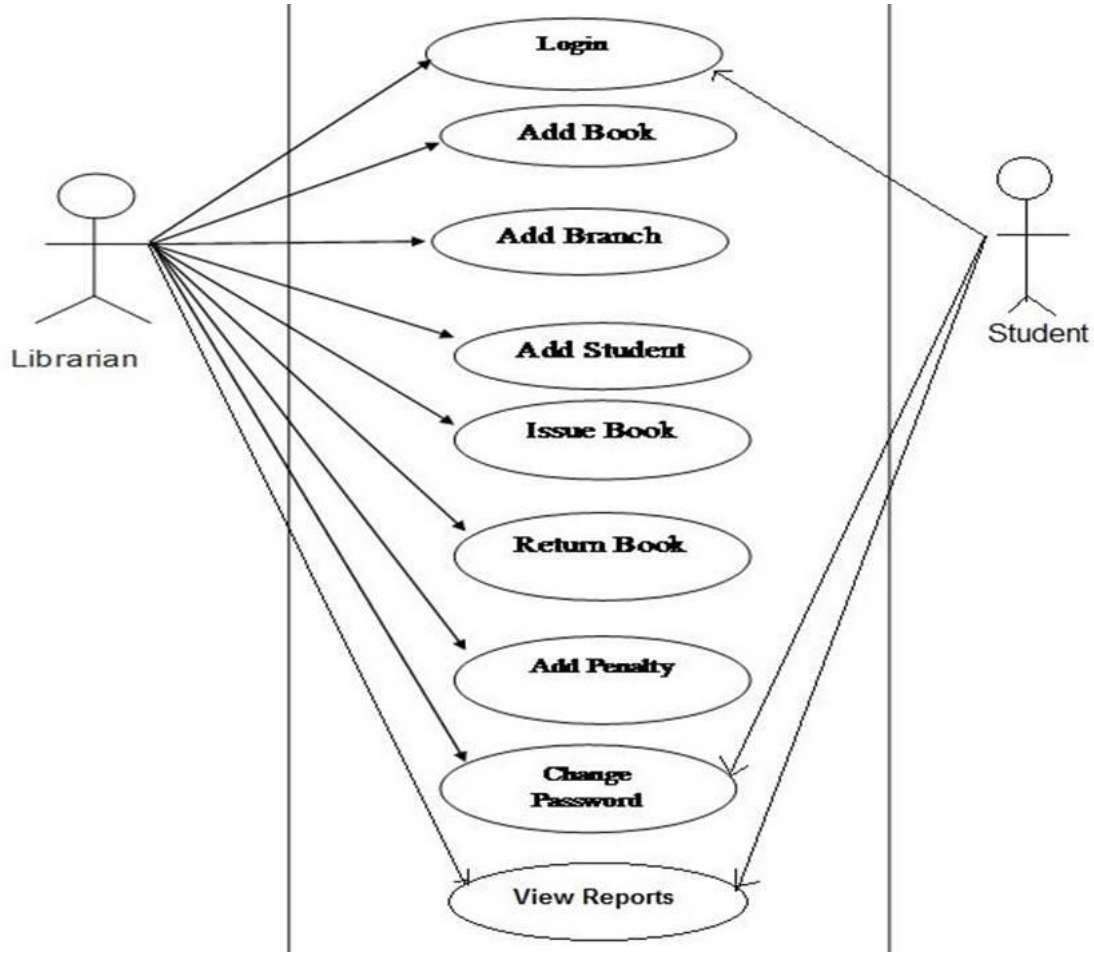
IDE (Optional):

- **IDLE (default Python editor)**
- **VS Code / PyCharm (for better coding experience)**

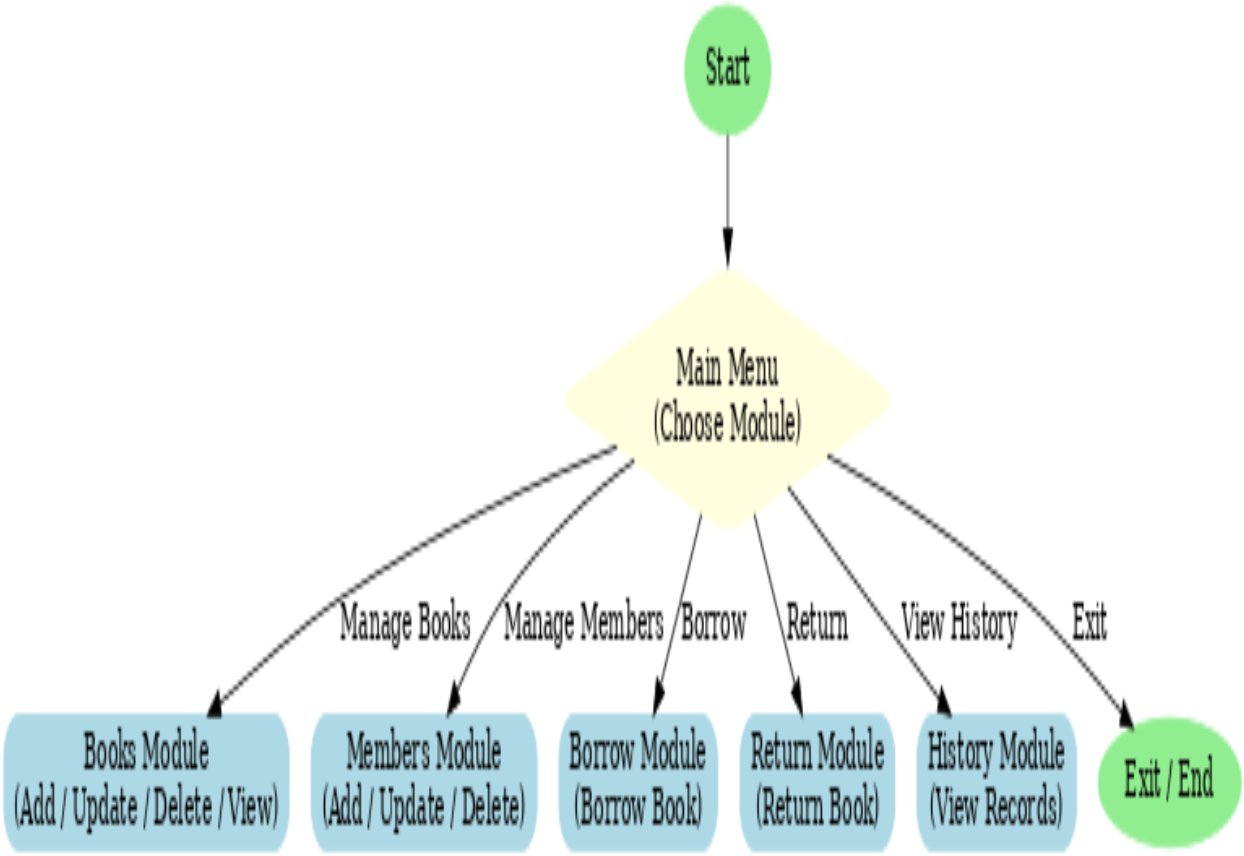
- **Version Control (Optional): Git / GitHub (for maintaining project version)**
- **File Storage: CSV format, ensuring portability without requiring external database installation.**
- **Execution Environment:**
    - **Run directly using python library_gui.py in terminal or command prompt.**

4 . SYSTEM DESIGN AND DEVELOPMENT

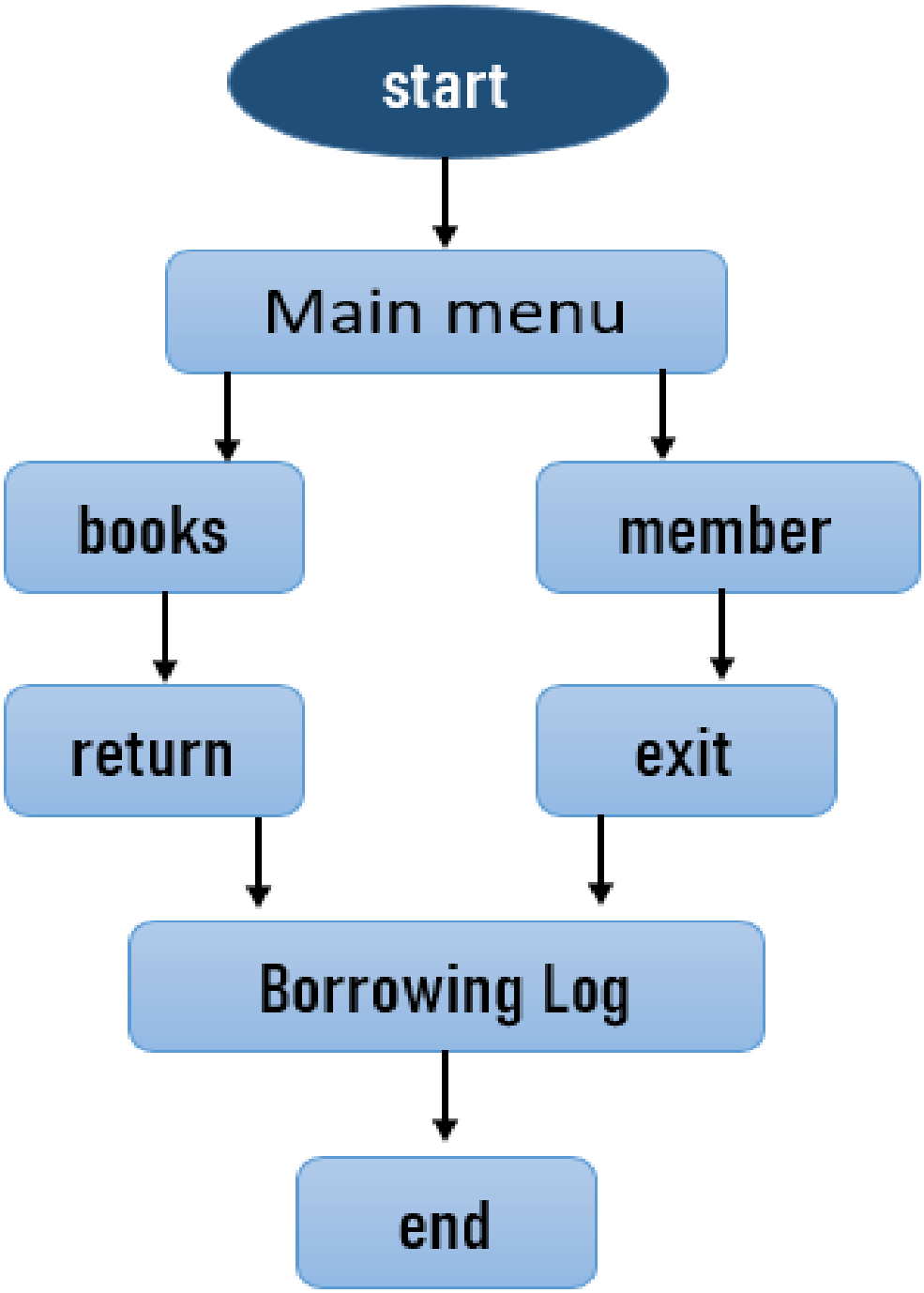
4.1 Use case Diagram

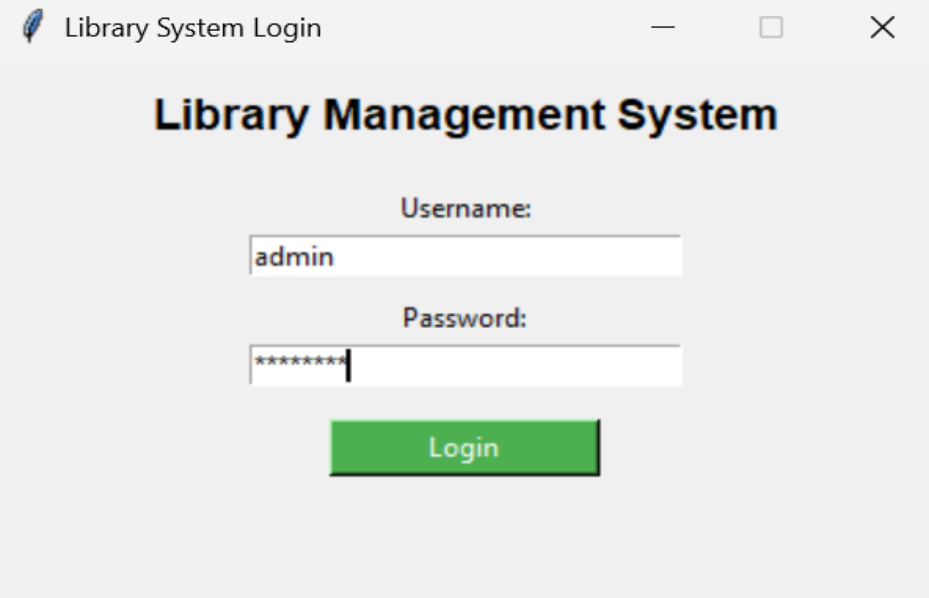


4.2 Data flow Diagram

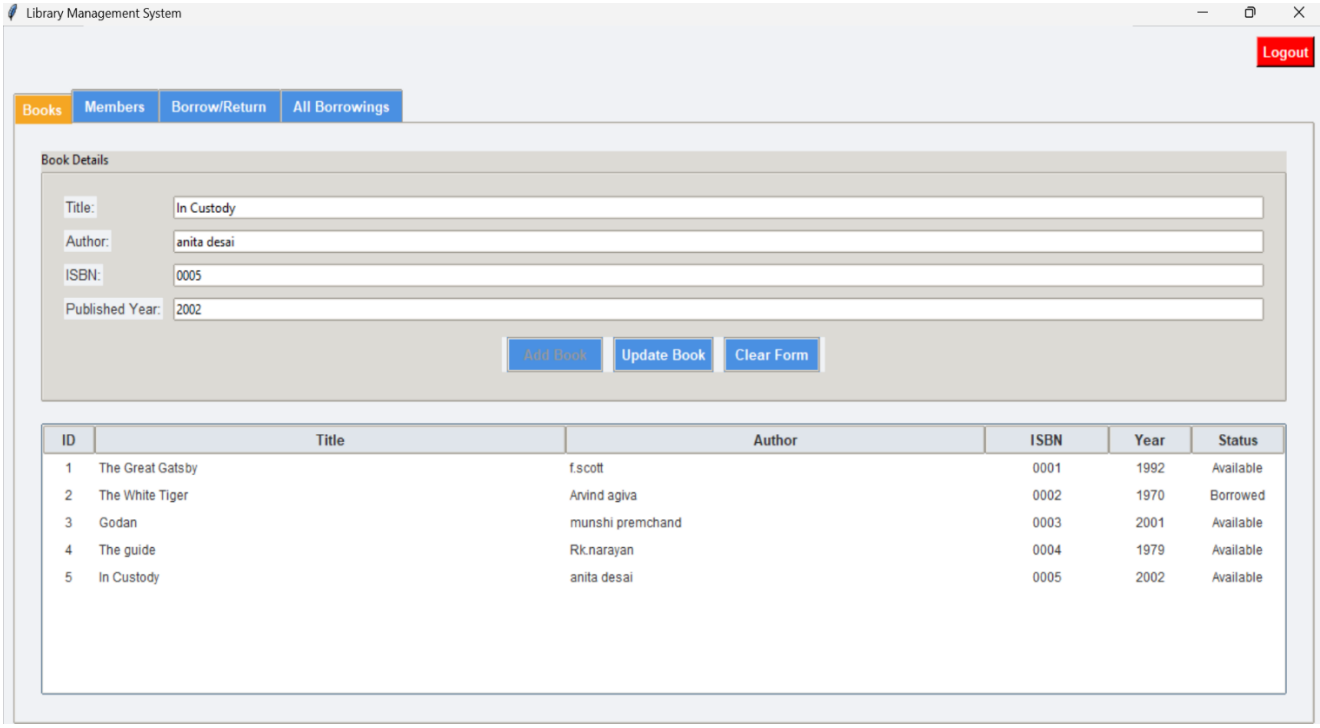


4.3 Flowchart.





4.4 Login Screen



4.5 Home Screen

5. CONCLUSION

Achievements of the Project:

- Successfully developed a GUI-based Library Management System using Python Tkinter.
- Automated essential library functions such as adding books, registering members, issuing, and returning books.
- Implemented a file-based database system (CSV files) to maintain books, members, and borrowings records.
- Designed a user-friendly interface that is easy to navigate for non-technical users (librarians, staff).
- Reduced manual work by replacing paper-based registers with an automated system.
- Implemented a search and reporting feature for quick access to records.

Limitations

- The system currently supports only single-user access (no multi-user or admin authentication).
- Data storage is limited to CSV files, which may not be suitable for large-scale libraries compared to SQL/NoSQL databases.
- No support for fine calculation or overdue tracking.
- Does not include barcode/QR code integration for book scanning.
- Lacks online accessibility (system runs only locally on the installed computer).

Future Enhancements

- Upgrade data storage from CSV files to a database like MySQL or SQLite for better scalability.
- Implement multi-user authentication with separate roles (Admin, Librarian, Student).
- Add fine management system for late book returns.
- Enable cloud integration so the library can be accessed online.

- Provide a mobile-friendly interface or web version for remote access.
- Add report generation (PDF/Excel export) for borrowed/returned books.

6. LEARNING DURING PROJECT

Skills Learned

1. Tkinter GUI Development – Learned how to design interactive user interfaces in Python using Tkinter widgets like buttons, labels, entry fields, and tables.
2. File Handling with CSV – Gained practical knowledge of reading, writing, and updating CSV files (books.csv, members.csv, borrowings.csv) to act as a database.
3. Python Programming Concepts – Applied concepts such as classes, functions, loops, and error handling in a real-world application.
4. System Design – Understood how to create flowcharts, data flow diagrams, and database design for better project planning.
5. GUI-based Data Management – Learned how to integrate backend logic with frontend GUI components.

Problem Solving Skills

Designed solutions for managing book borrowing and returning while ensuring records remain accurate in CSV files.

- Solved challenges related to data consistency, such as updating a book's status when borrowed or returned.

- Implemented search and filter features to quickly access information.
- Improved logical thinking by converting manual library processes into automated functions.

Teamwork / Individual Contribution

- If completed individually: Took responsibility for system analysis, coding, testing, and documentation.
- If done in a team:
    - One member focused on GUI design and Tkinter coding.
    - Another member worked on CSV file handling and backend logic.
    - Another handled documentation and diagrams (Flowcharts, DFDs, Use Cases).
- Collaborated effectively to integrate all modules into one functional system.

7. BIBLIOGRAPHY

7.1 Online References

1. **Python Official Documentation** – https://docs.python.org/3/
   (For Python basics, file handling, and object-oriented programming concepts)
2. **Tkinter Documentation** – https://docs.python.org/3/library/tkinter.html
   (For GUI design elements like buttons, labels, entry fields, and message boxes)
3. **GeeksforGeeks – Python CSV Handling** – https://www.geeksforgeeks.org/working-csv-files-python/
   (For reading, writing, and updating CSV files in Python)
4. **TutorialsPoint – Python Tkinter** – https://www.tutorialspoint.com/python/python_gui_programming.htm
   (For understanding GUI layouts and event handling)
5. **Stack Overflow & GitHub Discussions**
   (For troubleshooting errors and enhancing project functionality)

7.2 Offline References

1. **Beginning Python: From Novice to Professional** by Magnus Lie Hetland – (For advanced concepts in Python programming)
2. **Object-Oriented Programming with Python** – University lecture notes

3. **Classroom notes and faculty guidance** from the BCA program (for system design, flowcharts, and documentation format).