# Rationale for the Mini Library Management System

## 1. Purpose of the System

The Mini Library Management System was developed to simplify the process of managing books, members, and borrowing records in a small-scale library. It allows administrators, staff, and students to perform basic operations such as adding, updating, borrowing, and returning books in a structured and reliable way. The main goal is to simulate how a real library system operates—by keeping track of resources, managing users, and maintaining accurate borrowing history—while staying lightweight and easy to understand for demonstration or educational purposes.

## 2. System Design Overview

The system follows a modular design split into three main components: operations.py, demo.py, and tests.py. Each part plays a specific role in maintaining the structure and logic of the program.

## a. operations.py (Core Logic Layer)

This file serves as the backend of the system. It contains all the main operations and data structures required for the library’s functionality. It manages books, members, borrowing, returning, and user authentication. Each function is written to perform a specific task for better organization and reusability.

## b. demo.py (User Interface Layer)

This file acts as the user-facing interface through the command line. It provides different menus based on user roles such as Admin, Staff, or Student. It handles user input and uses the operations module to perform tasks securely.

## c. tests.py (Testing Layer)

This file ensures that the main functions perform as expected. It uses assert-based testing to confirm that operations like adding, borrowing, and returning books behave correctly. It helps maintain stability when making updates to the system.

## 3. Data Handling

The system uses in-memory Python data structures such as dictionaries and lists. Books are stored in a dictionary for quick lookup, members are stored as a list, and borrow history is recorded as a list of transactions. This keeps the program simple and efficient without requiring a database.

## 4. Borrowing Logic

Each member can borrow up to three books for seven days. The system ensures books are available before borrowing, prevents deletion of borrowed items, and updates both member and borrow records for accuracy.

## 5. Role-Based Access

Three user roles—Admin, Staff, and Student—reflect typical library responsibilities. Admins manage all data, staff can borrow and return books, and students can only handle their own records. The verify\_login function enforces these permissions for security.

## 6. Benefits of the Design

The system is simple, modular, testable, and easy to extend. Each function is independent, which allows future improvements such as connecting to a database or adding a web interface without rewriting the logic.

## 7. Conclusion

This Mini Library Management System demonstrates clear programming structure, role-based control, and data integrity. It is an excellent educational project for understanding how real-world systems manage users, records, and operations in an organized way.