Design Rationale – Mini Library Management System

This rationale explains the design decisions behind the Mini Library Management System project, including the data structures, functions, file separation, and rules for borrowing and returning books. The goal of this system is to model a simple console-based library environment while fulfilling assignment requirements.

Programming Approach

A procedural programming style was chosen instead of classes/objects because the system is small and the assignment focuses on data structures (lists, tuples, dictionaries). Storing the logic in functions ensures code readability, modularity, easier testing, and meets assessment requirements.

Data Structures and Their Purpose

1. Dictionary – Books

Books are stored using a dictionary of dictionaries with ISBN as the key. This ensures fast and direct access for CRUD operations.

2. List – Members

Members are stored in a list to support dynamic resizing as new members are added. Searching is acceptable due to small expected data size.

3. Tuple – Genres

Genres are constant values. A tuple ensures immutability and data validation.

Borrowing and Returning Rules

- A member can borrow a maximum of 3 books
- Book copies decrease when borrowed and increase when returned
- Both member and book must exist to complete the action

File Structure

operations.py – Contains logic and data

demo.py – Shows working example for demonstration

test.py – Automated functional testing with assert statements

This improves clarity, marking efficiency, and maintainability.

UML Diagram

The UML diagram visually represents relationships between data (Books, Members) and functions in operations, helping reviewers quickly understand structure.

Testing

Planned asserts in test.py verify core functionality and ensure consistency of future changes.

Conclusion

This design focuses on simplicity, reliability, and clarity. It fulfills all assignment requirements while modeling a realistic library borrowing system suitable for academic submission.