Software Requirements Specification (SRS)

version 1.0

Prepared by

Ahmed Abdalla El-Tahan

Abdelrahman Gawad

OCT 2024 – NOV 2024

1. Introduction

1.1 Purpose

The purpose of this document is to define the software requirements for the Library Management System (LMS). This system aims to simplify the management of books, users, and transactions for a bookstore or library. It provides an intuitive GUI for admins and users to manage and interact with the inventory.

1.2 Scope

The Library Management System allows:

- Admins to manage book inventory, including adding and removing books.
- Users to search for books, view inventory, manage a personal cart, and checkout.
- Both admin and user operations through a GUI-based system built using Python and Tkinter.

Data for users is stored in a JSON file, while the book inventory is stored in a CSV file.

1.3 Definitions, Acronyms, and Abbreviations

- **LMS**: Library Management System
- **GUI**: Graphical User Interface
- **JSON**: JavaScript Object Notation
- **CSV**: Comma-Separated Values

1.4 References

- Python Documentation: https://docs.python.org/
- Pandas Library: https://pandas.pydata.org/
- Tkinter GUI Toolkit: https://docs.python.org/3/library/tkinter.html

---

2. Overall Description

2.1 Product Perspective

The LMS is a standalone application for local use. It does not require integration with external systems. The system leverages the Singleton design pattern to ensure only one instance of the backend logic is created.

2.2 Product Features

- **Admin Functionalities:**
    - Add new books to inventory.
    - Remove books by title.
    - View the complete inventory.
- **User Functionalities:**
    - Search for books by title, author, or genre.
    - Add books to a cart.
    - Remove books from the cart.
    - Checkout to view a summary of selected books and their total price.
- **Common Features:**
    - Login and signup functionality.
    - Logout functionality.

2.3 User Classes and Characteristics

- **Admin:** Responsible for managing the book inventory.
- **User:** End-users who interact with the system to search for and purchase books.

2.4 Operating Environment

- Operating System: Windows, macOS, or Linux
- Python 3.8 or later
- Required Libraries: Pandas, Tkinter, Base64

2.5 Constraints

- Data storage is limited to local JSON and CSV files.
- The system must ensure a smooth user experience via Tkinter GUI.

2.6 Assumptions and Dependencies

- Users have basic computer skills.
- Python and required libraries are pre-installed on the system.

---

3. Specific Requirements

3.1 Functional Requirements

3.1.1 Admin Functionalities

- The system must allow the admin to add books by entering their details (Book ID, Title, Author, Price, Quantity, Genre).
- The system must allow the admin to remove books by title.
- The system must display the complete inventory in a tabular format.

3.1.2 User Functionalities

- The system must allow users to search for books using a search bar with optional filters (Title, Author, Genre).
- The system must allow users to add books to a cart.
- The system must allow users to remove books from the cart.
- The system must provide a checkout feature to summarize the cart and calculate the total price.

3.1.3 Common Features

- The system must allow users to log in with a valid username and password.
- The system must allow new users to sign up.
- The system must allow users to log out.

3.2 Non-Functional Requirements

3.2.1 Usability

- The GUI should be intuitive and easy to navigate.
- Errors and success messages should be displayed using pop-ups.

3.2.2 Performance

- Inventory and user data must be loaded within 2 seconds.
- Search operations should return results within 1 second.

3.2.3 Security

- Passwords must be stored in an encrypted format using Base64 encoding.
- User roles must be validated before accessing admin functionalities.

3.2.4 Reliability

- The system must handle unexpected inputs gracefully without crashing.
- Data integrity must be maintained during file read/write operations.

3.2.5 Maintainability

- Code must follow a modular structure with classes for User, Admin, LibrarySystem, and GUI components.
- Functions and classes must be documented.

3.2.6 Portability

- The system must run on any machine with Python 3.8 or later.

---

4. Data Requirements

- **User Data:** Stored in loginDetails.json, containing username, encrypted password, and role.
- **Books Data:** Stored in books.csv, containing Book ID, Title, Author, Price, Quantity, and Genre.

5. System Models

5.1 Class Diagram

Classes:

- LibrarySystem
- Cipher
- Person
- Admin
- User
- Login_Gui
- Admin_Gui
- User_Gui

Relationships:

- Admin and User inherit from Person.
- LibrarySystem is used by all GUI components and manages core logic.
- Cipher is utilized by LibrarySystem for encryption.