System Design Specification Document

Automated Book Reservation System

Table of Contents:

1. Introduction:

1.1 Purpose:

The purpose of this document is to provide a detailed description of the design specifications for the Automated Book Reservation System. It outlines the system's architecture, components, functionalities, and design considerations to guide the development process.

1.2 Scope:

The Automated Book Reservation System aims to automate various processes involved in library management, including user registration, book reservation, borrowing, returning, and administrative tasks. The system caters to users, administrators, and librarians, providing each with specific functionalities tailored to their roles.

1.3 Audience:

This document is intended for developers, designers, testers, and stakeholders involved in the development, testing, and implementation of the Automated Book Reservation System. It serves as a comprehensive guide to understanding the system's design and functionality.

2. System Overview:

The Automated Book Reservation System comprises three main components: the User Interface, Back end Logic, and Database. These components work together to provide a seamless experience for users, administrators, and librarians.

2.1 Component Overview:

User Interface: Provides an interactive interface for users, administrators, and librarians to access system features.
Back end Logic: Implements logic for user authentication, book management, reservation, borrowing, and returning.
Database: Stores user information, book details, reservation records, and system configurations.

2.2 Technologies:

Programming Language: C++
Libraries: Standard Template Library (STL)
Database: No Separate Database Currently (Database Class Only)
User Interface: Command-line interface (CLI)

2.3 Minimum Requirements:

Processor: Intel Core i3
Clock Speed: 1 GHz or higher
RAM: 1 GB DDR3 RAM
Storage: 1 GB Free Memory

3. Module Details:

3.1 User Management Module:

Responsible for user registration, authentication, and profile management. It also includes functionalities for creating new user accounts, verifying user credentials, and updating user profiles.

3.2 Book Management Module:

Manages book inventory and related operations. It allows users to search for books, reserve them, borrow, and return them. Also provides functionalities for librarians to add new books, update book details, and manage inventory.

3.3 Admin Module:

Handles administrative tasks such as admin registration, login, and reporting. And on the other hand, it allows administrators to register new admin accounts, authenticate, and access reporting functionalities.

3.4 Librarian Module:

Enables librarians to verify user accounts and manage book inventory. Librarian Module also provides functionalities for librarians to verify user credentials, check book availability, and update book status.

3.5 Database Module:
Manages database operations including CRUD operations for user information, book details, reservations, etc. It ensures data integrity, security, and scalability of the system's database.

3.6 User Interface Module:
Provides an intuitive and user-friendly interface for users, administrators, and librarians. It presents functionalities in a clear and organized manner, facilitating easy navigation and interaction.

4. Integration:

4.1 Integration Strategy:
Modules are integrated to ensure seamless communication and functionality across the system. It is also made sure that upon interaction of modules, system behaves normally and any integration issues are then identified and resolved.

5. Design Considerations:

5.1 Goals and Guidelines:
The primary goal of the Automated Book Reservation System is to streamline library management processes by automating tasks such as user registration, book reservation, borrowing, returning, and administrative functions. The design guidelines emphasize scalability, maintainability, usability, and security as key principles guiding system development.

5.2 Development Methods & Contingencies:
The development approach for the system follows an Agile methodology, allowing for iterative development and continuous feedback from

stakeholders.Contingencies include different plans and different ways to implement code if one approach doesn't work.

5.3 Architectural Strategies:

The Automated Book Reservation System is designed like building blocks. Each part works independently, making it easy to add or change things without affecting the whole system. Modules communicate seamlessly, like different departments in a company. Overall, it's flexible, scalable, and easy to manage.

6. Interfaces:

6.1 Interface Architecture:

The Automated Book Reservation System, coded in C++, orchestrates database interactions and system functionalities. The back end collaborates with the front end interface to process user requests, manage book inventory, authenticate users, and execute administrative tasks. This ensures seamless coordination between the user interface and the underlying database, facilitating effective data management and system operation.

6.2 User Interface Design:

The user interface is on the command line and the user interface design will prioritize simplicity, accessibility, and intuitiveness, with clear navigation paths, descriptive labels, and consistent layout across different device types.

7. Conclusion:

In conclusion, the Software Design Specification (SDS) document provides a comprehensive overview of the design and architecture of the Automated Book Reservation System. By detailing the system's modules, functionalities, and interactions, along with software requirements and design considerations, this document serves as a roadmap for developers,

designers, testers, and stakeholders involved in the system's development, testing, and implementation. With a clear understanding of the system's design and functionality, the development team can effectively implement and deploy the Automated Book Reservation System, meeting the needs of users, administrators, and librarians alike.