Software Requirements Specification

for

Personal Library System

Version 1.0 approved

Prepared by Aditya Rastogi

Roll No.: 16CS30042

CSE, IIT Kharagpur

16th January, 2018

Table of Contents

Revision History

Name	Date	Reason for Changes	Version
Aditya Rastogi	16.01.18	First Draft.	1.0

1. Introduction

1.1 Purpose

- Management of huge collection of one's personal books adds unnecessary stress to one's daily life. Searching for a book in hundreds of books can be a very daunting task to do. This is the information age, the age of technology. Automated and intelligent systems are coming up every day.
- Issuing a book to a friend or a colleague not only adds extra load of remembering the details of the users but also creates the risk of losing the book due to the limitations of memory.
- Also, keeping track of the total count of books present in the library by manually counting the books every time after a week or two proves to be a very tedious task to do.

So, why not automate this process? This is the basic purpose behind this Personal Library System software.

1.2 Product Scope

Benefits:

- *To assist users, manage their personal libraries in a simpler and automated manner.*
- *Proper utilization of the available collection of books.*
- *To ensure safety of books.*
- *To keep account of the count of books in the library.*
- *Fast search of books in the library.*

End users:

1. *Students, academicians and other people with large collection of books requiring an automated library software.*

2. Overall Description

2.1. Product Perspective

This software will serve as a modified and better software than many softwares which have been developed for managing personal libraries. The software will also make use of web based query system for book search.

2.2. Product Features

The software will have the following four major features:
(a) Managing own books.
(b) Managing Friend details

(c) Managing Borrowed books
(d) Displaying Book count in the library

2.3. Design and Implementation constraints

Java is the primary programming language that will be used to build the software along with JavaScript/ Java based library for web development work required for this software, SQL or any other free of cost publicly available DBMS can be used for storing details of users and books.

3. Functional Requirements

The software needs to support four categories of functionalities as described below:

3.1 Manage own books

3.1.1. Register book

Description: To register a book in the personal library, the details of the book such as the name of the book, year of publication, date of purchase, price and publisher are entered. A unique serial number for the book has to be generated by the computer. These details are to be stored in the database.
Input: Book details
Output: Unique Serial number

3.1.2. Issue book

Description: A friend can be issued a book only if he or she is registered. The various books outstanding against him/her along with the dates on which the books were borrowed are first displayed.

3.1.2.1. Display Outstanding books
Description: The name of the friend and the serial number of the book to be issued are entered. Then the books outstanding against the friend should be displayed.
Input: Friend's name
Output: List of Outstanding books along with the dates on which the books were borrowed.

3.1.2.2. Confirm issue book
Description: If the owner of the book confirms, then the book should be issued and the relevant details should be updated.
Input: Owner confirmation for book issue
Output: Confirmation of book issue

3.1.3. Query outstanding books

Description: Upon query, the software displays the name, address, and telephone numbers of each friend against whom books are outstanding along with the titles of the outstanding books and the date on which those were issued.
Input: User selection of the query.
Output: The output display includes the name, address, and telephone numbers of each friend against whom books are outstanding along with titles and issue dates of books.

3.1.4. Query book

Description: Any user should be able to query about the availability of a particular book.
Input: Name of the book.
Output: Availability of the book, book available or issued out.

3.1.5. Return book

Description: When a friend returns a book, the date of return is stored and the book is removed from his/her borrowing list. A confirmation message must be displayed after completion of the process.
Input: Serial number of the book
Output: Confirmation message.

3.2 Manage friend details

3.2.1. Register friend

Description: Before a friend can be lent a book or a book can be borrowed from him/her, he/she must be registered. After the registration data is entered correctly, the data should be stored in the database and a confirmation message should be displayed.
Input: The registration data would include name of the friend, address, land line number, and mobile number.
Output: Confirmation of registration status

3.2.2. Update friend details

Description: When a friend's registration information changes, the same must be updated in the database of the computer.

3.2.2.1. Display current details

Description: The current details of the friend should be displayed given name as the input.
Input: Friend's name
Output: Current stored details

3.2.2.2. Update friend details
Input: Changes in the details of the friend.
Output: Updated details with confirmation of the changes.

3.2.3. Delete friend details

Description: Delete details of friends who have become inactive based on owner's choice.
Input: Friend's name
Output: Confirmation message of deletion of details of the friend.

3.3 Manage borrowed books

3.3.1 Register borrowed books

Description: The details of the books borrowed by the owner of the library software are registered.
Input: Title of the book, date on which book was borrowed and the name of the friend from whom the book was borrowed.
Output: Confirmation of the registration status.

3.3.2 Deregister borrowed books

Description: Deregistering of a borrowed book is done when the book is returned. The input data is to be stored in the database.
Input: Book title and date on which the book is returned.
Output: Confirmation of deregistration.

3.3.3 Display borrowed books

Description: The software should be able to display all the books borrowed from various friends.
Input: User selection.
Output: List of books borrowed from other friends.

3.4. Manage statistics

3.4.1. Display book count

Description: The owner should be able to query the total number of books in the personal library.
Input: User Selection.
Output: Count of books in the personal library.

4. Non-Functional Requirements

4.1 Platform/Portability

Both Windows and Linux based versions of the software need to be developed.

4.2 Web-Support

The software functionality of Query book availability by any user should be made possible to be invoked from any location by using a web browser.

4.3. Database

A database management system which is available free of cost publicly, should be used.

4.4. Security

Authentication and access rights to any user will be under the purview of the owner.

4.5. Safety

A backup of the software must be stored in an external memory disk/pen drive to handle any software, main hardware or power failure.

5. External Interface Requirements

5.1. User Interfaces

The user interface which needs to be developed for this software needs to be a GUI, for Windows, a GUI Windows application and similarly, for Linux, a Linux GUI application.

5.2. Hardware interfaces

The application requires a PC upon which it will work.
Minimum specifications the PC must possess: 2 GB RAM, Processor : 1.8 GHz, External Storage Availabilty : 1 GB

5.3. Software interfaces

The application uses the software knowledge of Java for building the application, JavaScript/Java library for building the web-based support for the application and DBMS software, here we will be using SQL for storing user and book details.

5.4. Communication Interfaces

The software requires web access for the query book functionality that it offers.

6. Tentative Development Schedule

The tentative development schedule for developing the application is approximately 10 weeks. The part wise schedule is given below:

(a) Learning design for Java software – 2 weeks
(b) Coding for the software – 4 weeks
(c) Web interfacing for query book functionality and assuring portability of software – 2 weeks
(d) Implementation and Testing – 2 weeks