COMP214 AI Group Project

**Library Path Searching System (LPSS)**

Appendix A

Requirement Document

**Team: AI Group2**

**– CodeCooks**

Jiazhe Wang

Wenyang Cai

Xueli Jia

Yiming Li

ZhaoZhao

2012/5/13

**Content**

[1 Preface 1](#_Toc317177283)

[1.1 Revision history: 1](#_Toc317177284)

[1.2 Audience 1](#_Toc317177285)

[2 Introduction 2](#_Toc317177286)

[2.1 Purpose of This Document 2](#_Toc317177287)

[2.2 How to Use This Document 2](#_Toc317177288)

[2.2.1 Types of Reader 2](#_Toc317177289)

[2.2.2 Technical Background Required 3](#_Toc317177290)

[2.2.3 Overview Sections 3](#_Toc317177291)

[2.3 Scope of the Product 3](#_Toc317177292)

[2.4 Business Case for the Product 3](#_Toc317177293)

[3 Glossary 4](#_Toc317177294)

[4 User requirement definition 6](#_Toc317177295)

[4.1 User relationship diagram 6](#_Toc317177296)

[4.2 About Interface 6](#_Toc317177297)

[4.3 About Database 8](#_Toc317177298)

[4.3.1 Mission statement 8](#_Toc317177299)

[4.3.2 Mission objectives 8](#_Toc317177300)

[5 System architecture 9](#_Toc317177301)

[5.1 Layers: 9](#_Toc317177302)

[6 System requirement specification 10](#_Toc317177303)

[6.1 Specification Overview 10](#_Toc317177304)

[6.2 Library Path Search System Overview 10](#_Toc317177305)

[6.2.1 Definition 11](#_Toc317177306)

[6.2.2 Goal 11](#_Toc317177307)

[6.2.3 Context of the Library 11](#_Toc317177308)

[6.2.3.1 External Hardware 11](#_Toc317177309)

[6.2.3.2 External Roles 12](#_Toc317177310)

[6.2.3.3 External Software 12](#_Toc317177311)

[6.2.3.4 External Systems 12](#_Toc317177312)

[6.2.4 Library Path Search System Capabilities 13](#_Toc317177313)

[6.2.4.1 Summary of System Capabilities 13](#_Toc317177314)

[6.2.4.1.1 Employee Capabilities 13](#_Toc317177315)

[6.2.4.1.2 Users Capabilities 13](#_Toc317177316)

[6.2.4.1.3 Other Capabilities 14](#_Toc317177317)

[6.3 Functional Requirements 14](#_Toc317177318)

[6.3.1 Summary Use Case Diagrams 14](#_Toc317177319)

[6.3.2 External Roles 16](#_Toc317177320)

[6.3.2.1 General User 16](#_Toc317177321)

[6.3.2.1.1 Definition 16](#_Toc317177322)

[6.3.2.1.2 Requirement 16](#_Toc317177323)

[6.3.2.2 Junior Librarian 17](#_Toc317177324)

[6.3.2.2.1 Definition 17](#_Toc317177325)

[6.3.2.2.2 Use case diagram 17](#_Toc317177326)

[6.3.2.2.3 Requirement 18](#_Toc317177327)

[6.3.2.3 Senior Librarian 18](#_Toc317177328)

[6.3.2.3.1 Definition 18](#_Toc317177329)

[6.3.2.3.2 Use case diagram 19](#_Toc317177330)

[6.3.2.3.3 Requirement 19](#_Toc317177331)

[6.3.2.4 Database Administrator 19](#_Toc317177332)

[6.3.2.4.1 Definition 19](#_Toc317177333)

[6.3.2.4.2 Use case diagram 20](#_Toc317177334)

[6.3.2.4.3 Requirement 20](#_Toc317177335)

[6.3.2.5 Curator 20](#_Toc317177336)

[6.3.2.5.1 Definition 20](#_Toc317177337)

[6.3.2.5.2 Use case diagram 21](#_Toc317177338)

[6.3.2.5.3 Requirement 21](#_Toc317177339)

[6.4 Data Requirements[1] 22](#_Toc317177340)

[6.4.1 User view: 22](#_Toc317177341)

[6.4.2 Data requirements: 23](#_Toc317177342)

[6.4.3 Initial database size: 23](#_Toc317177343)

[6.4.4 Database rate of growth: 23](#_Toc317177344)

[6.4.5 Performance: 24](#_Toc317177345)

[6.4.6 Security: 24](#_Toc317177346)

[6.4.7 Backup and recovery: 24](#_Toc317177347)

[6.4.8 Legal issues: 24](#_Toc317177348)

[6.5 System Quality Requirements 24](#_Toc317177349)

[6.5.1 Correctness 25](#_Toc317177350)

[6.5.1.1 Accuracy 25](#_Toc317177351)

[6.5.1.2 Precision 25](#_Toc317177352)

[6.5.1.3 Timeliness 25](#_Toc317177353)

[6.5.2 Efficiency 25](#_Toc317177354)

[6.5.3 Extensibility 26](#_Toc317177355)

[6.5.4 Interoperability 26](#_Toc317177356)

[6.5.5 Maintainability 26](#_Toc317177357)

[6.5.6 Performance 26](#_Toc317177358)

[6.5.6.1 Capacity 26](#_Toc317177359)

[6.5.6.2 Latency 27](#_Toc317177360)

[6.5.6.3 Response Time 27](#_Toc317177361)

[6.5.7 Portability 27](#_Toc317177362)

[6.5.8 Reusability 27](#_Toc317177363)

[6.5.9 Robustness 28](#_Toc317177364)

[6.5.10 Safety 28](#_Toc317177365)

[6.5.11 Usability 28](#_Toc317177366)

[7 System models 29](#_Toc317177367)

[7.1 Boundary Diagram 29](#_Toc317177368)

[7.2 Component Diagram 30](#_Toc317177369)

[7.3 Activity Diagram 30](#_Toc317177370)

[7.4 Future consideration 31](#_Toc317177371)

[8 System evaluation 31](#_Toc317177372)

[8.1 Software Evolution: 31](#_Toc317177373)

[8.1.1 Fundamental Assumptions 32](#_Toc317177374)

[8.1.2 Program Evolution Dynamic 32](#_Toc317177375)

[8.1.2.1 Continuing change 32](#_Toc317177376)

[8.1.2.2 Increasing complexity 32](#_Toc317177377)

[8.1.2.3 Large program evolution 33](#_Toc317177378)

[8.1.2.4 Organizational stability 33](#_Toc317177379)

[8.1.2.5 Conservation of familiarity 33](#_Toc317177380)

[8.1.2.6 Continuing growth 33](#_Toc317177381)

[8.1.2.7 Declining quality 33](#_Toc317177382)

[8.2 Software Maintenance 34](#_Toc317177383)

[8.2.1 Corrective maintenance 34](#_Toc317177384)

[8.2.2 Adaptive maintenance 34](#_Toc317177385)

[8.2.3 Perfective maintenance 34](#_Toc317177386)

[8.2.4 Preventive maintenance 34](#_Toc317177387)

[8.3 Evolution Process 34](#_Toc317177388)

[8.3.1 Analysis 35](#_Toc317177389)

[8.3.2 Description 35](#_Toc317177390)

[8.3.3 Further Considerations 35](#_Toc317177391)

[9 Reference 36](#_Toc317177392)

[10 Appendices 36](#_Toc317177393)

# Preface

## Revision history:

|  |  |  |
| --- | --- | --- |
| LPSS Requirement Release | Update information | Update date |
| 1.0 | First draft | 12/02/2012 |
| 1.1 | Update Data Requirement and System Quality Requirement | 13/02/2012 |
| 1.2 | Update Introduction | 13/02/2012 |
| 1.3 | Update Functional Requirement | 14/02/2012 |
| 1.4 | Add several diagrams | 15/02/2012 |
| 1.5 | Update Evolution | 15/02/2012 |
| 2.0 | Update Introduction | 15/02/2012 |
| 2.1 | Type Setting | 16/02/2012 |
| 2.2 | Final version | 16/02/2012 |

## Audience

* Library staff (i.e. Curator, Senor Librarians, Junior Librarians, Database Administrators)
* Staff of Computing Services Department (i.e. Manager, Administrators of network)
* Project tutors

# Introduction

This document contains the requirements for Library Path Search System (LPSS).

## Purpose of This Document

This document is intended to describe the system to users and guide development of this project. It will go through several stages during the process of the project:

* **Draft**: The first version is documented after requirements have been discovered, recorded and classified.
* **Proposed**: The draft document is then proposed as a temporary requirements specification for the project. The proposed document should be reviewed by several audiences, who may comment on any requirements, either to agree, to disagree, or to identify missing requirements. Audiences include end-users (e.g. library staff), developers, project managers (module monitor), and any other stakeholders. The document may be amended several times before validation.
* **Validated**: Once stakeholders have agreed to the requirements in the document, the requirement document is considered validated.
* **Approved**: The validated document is accepted by representatives of each party of stakeholders as an appropriate statement of requirements for the project. The developers then use the requirements document as a guide to implementation and to check the progress of the project as it develops.

## How to Use This Document

This document will be used by people with different skill sets. This section explains which parts of this document should be reviewed by various types of readers.

### Types of Reader

* **End-users**: End-users may need to read the Glossary to learn some definitions of this document. User Requirement Definition is also useful for them. Sections about the system may be safely skipped.
* **Programmers**: Programmers shall read the System Architecture, System Requirement Specification and System Models to guide the development. GUI designer may also need to read the User Requirement Definition to know the user requirement.
* **Project managers**: Project managers may need to read all document. Glossary and System Architecture and User Requirement Definition can help them to understand what the system is. System Requirement Specification and System Model describe the system in more details. System Evolution looks into the potential of the system.

### Technical Background Required

Knowledge about Software Engineering, Web programming and Database may be required to read this document.

### Overview Sections

The Introduction, User Requirement Definition and System Architecture should be read by someone who only wishes to gain an overall understanding of the project. Glossary should be read first before technical requirements are reviewed.

## Scope of the Product

The product intended to be realized is a system that can find the shortest path for returning or collecting particular books in a specific library. This system is expected to be embedded into a webpage of the library’s website. It shall have three basic functions: guiding borrowers to find books, guiding librarians to return and collect books efficiently, controlling a library robot (if there is one) to return and collect books automatically. However, the system in this project is bounded in web software. The embedded software in a robot or other external equipment is not a content of this project.

## Business Case for the Product

Libraries have numerous books in a number of bookshelves on different floors. It is quite hard and will cost much time to collect a number of books in a library. This product can help a library to construct a user-friendly system for borrowers and improve the efficient of its staff.

# Glossary

* **Library Path Search System (LPSS):**

LPSS is the system which is intended to be developed in this project. It can find the shortest path for returning or collecting particular books in a specific library. This system is expected to be embedded into a webpage of the library’s website. It shall have three basic functions: guiding borrowers to find books, guiding librarians to return and collect books efficiently, controlling a library robot (if there is one) to return and collect books automatically.

* **Shortest path**:

The shortest path is a path which has the minimum sum of distances that visit several locations of books.

* **General** **user**:

General users are any LPSS users except employees, may be students or staffs of university.

* **Junior** **Librarian**:

Junior Librarians are who works for the library, maybe temporary workers. They return books to the right locations and collect books for general user every day. They can use LPSS to improve efficiency or control a library robot to do these jobs.

* **Senior** **librarian**:

Senior librarians are who manages Junior Librarians and important work of the library. Senior librarian can manipulate the database of books in LPSS. (E.g. change the locations of books on bookshelves)

* **Database** **Administrator**:

Database Administrators are who updates, manipulates and maintains all databases of LPSS. (E.g. update the maps of library)

* **Curator**:

Curators are who has the highest authority. Curator can manage other employees.

* **User**:

Users of this system contain all general users, junior librarians, senior librarians, database administrators and the curator.

* **Employee**:

Employees are users except general users.

* **Book** **list**:

A book list is a list of user’s favorite books. It should be generated and stored in the library system and LPSS imports it.

* **Operation** **list**:

An operation list is a list of books which the user wants to return/collect at a time. Books in an operation list can be selected from book list or search results.

* **Path** **list**:

A path list is a list of sorted books with their location information. It is generated from the operation list by LPSS. Users can use a Path List Printing System to print the path list so that they can take it with them to find these books.

* **Path** **List** **Printing** **System (PLPS)**:

A PLPS is a printing system which connects to LPSS. It can print a path list after LPSS generated it.

* **To-do list**:

A To-do list lists the tasks of a librarian, including books which should be returned to original location and books which should be collected from bookshelves.

* **Library** **robot**:

A library robot is a machine that can return and collect books automatically. If there is a library robot in a library, it is controlled by librarians and can help them to do return and collect tasks. It will access the LPSS to obtain the map information and the shortest path.

* **Map**:

In this system, a “map” is environment data of an individual floor of the library. It specifies the bound of the library and the locations of bookshelves. Maps are stored in the map database. Database Administrators can update maps.

# User requirement definition

## User Relationship Diagram

|  |
| --- |
|  |

## About Interface

|  |  |
| --- | --- |
| User | Functional requirement |
| General user | User shall be able to log in the system by entering username and password.  User shall be able to change password.  User shall be able to view personal information after logging in.  User shall be able to view the borrowed-book list and relevant information.  User shall be able to search for a book and view book information.  User shall be able to choose the database of a particular library among many libraries.  User shall be able to search for books with relevant attributes.  User shall be able to search for the shortest path to get the books. |
| Junior Librarian | Junior Librarian shall be able to log in the system by entering username and password.  Junior Librarian shall be able to change password.  Junior Librarian shall be able to view personal information after logging in.  Junior Librarian shall be able to view the borrowed-book list and relevant information.  Junior Librarian shall be able to search for a book and view book information.  Junior Librarian shall be able to choose the database of a particular library among many libraries.  Junior Librarian shall be able to search for books with relevant attributes.  Junior Librarian shall be able to search for the shortest path to get the books.  Junior Librarian shall be able to view particular general user information.  Junior Librarian shall be able to reset general user password to a system generated password after general user’s permission.  Junior Librarian can control library robot to assist working if there is one. |
| Senior librarian | Senior librarian shall be able to log in the system by entering username and password.  Senior librarian shall be able to change password.  Senior librarian shall be able to view personal information after logging in.  Senior librarian shall be able to view the borrowed-book list and relevant information.  Senior librarian shall be able to search for a book and view book information.  Senior librarian shall be able to choose the database of a particular library among many libraries.  Senior librarian shall be able to search for books with relevant attributes.  Senior librarian shall be able to search for the shortest path to get the books.  Senior librarian shall be able to search for information about Junior Librarians from different viewpoints.  Senior librarian shall be able to allocate jobs and working times of Junior Librarians. |
| Database administrator | Database administrator shall be able to view and update the configuration information of any particular floor of a library.  Database administrator shall be able to update, manipulate, and modify the databases of books and maps. |
| Curator | Curator shall be able to view all the information of library facilities from different viewpoints.  Curator shall be able to view/add/delete/modify accounts and authorities of all employees. |

## Database Section

### Mission statement

The purpose of library database system is to collect, store, manage, and control access to the data that supports general user or librarian to derive information or control on it.

### Mission objectives

|  |
| --- |
| To maintain (enter, update, and delete) data on personal information.  To maintain (enter, update, and delete) data on books of own booklist.  To maintain (enter, update, and delete) data on all of librarian.  To maintain (enter, update, and delete) all the books on the shelves.  To maintain (enter, update, and delete) the locations of shelves.  To perform searches on books.  To perform searches on all the general users.  To track the status of books.  To report on all the librarians.  To report on staff and general users.  To report on all books of own booklist.  To report on the locations of books and shelves. |

# System architecture

## Layers:

|  |  |
| --- | --- |
| User interface layer | Web browser interface |

|  |  |
| --- | --- |
| User communications layer | LPSSlogin  Application management  Query management  Connecting functions |

|  |  |
| --- | --- |
| Information retrieval and modification layer | Security  Search |

|  |  |
| --- | --- |
| Database management layer | DBMS |

|  |  |
| --- | --- |
| Database | DB |

* Web browser interface: There are 5 different kinds of end-user, so there are 5 different kinds of user interface.
* LPSSlogin component identifies users. Different user has different authorities to access different data.
* Application management including three layers of actions which are presenting data, application processing, data management.
* Query management component optimize and control the query form the database.
* Connecting function allow this system to connect to other outside systems and applications.
* Print management manages printing maps and book list.
* Security system control the authority checking and protecting database access and system.
* Monitoring record all the operations of user.
* Search component search the shortest path of the map.
* DBMS including backup and recovery, security management, concurrency control, Change management.

# System requirement specification

## Specification Overview

This section introduces the system requirements specification (SRS) for the Library Path Search System (LPSS).

This specification is organized into the following sections:

* System Overview, which provides a high level description of the LPSS including its definition, goal, context, and capabilities.
* Functional Requirements, which specifies the functional system requirements of each external’s, use cases.
* Data Requirements, which specifies data requirements of this system.
* Quality Requirements, which specifies the required system quality factors.

## Library Path Search System Overview

This section provides a high level description of the Library Path Search System (LPSS) system including its definition, goal, context, and capabilities.

### Definition

The Library Path Search System (LPSS) will be a system which always finds a shortest path to return or get appointed books from the library and show this path to users or use this path to control a library robot if there is one.

### Goal

The goals of the Library Path Search System (LPSS) are:

* To simplify the process of searching the locations of books and guide the borrowers.
* To save time of librarians for managing books.
* To control a library robot (if there is one) to return or collect books.

### Context of the Library

This subsection documents the context of the LPSS in terms of the significant externals with which it interacts. The Library Path Search System (LPSS) interfaces with some external actors and systems:

* Students or staffs of a university act as general users. General users can only search books, obtain the shortest path and print a path list with the LPSS.
* Junior Librarians are officers who work for the library. They may be temporary workers. Junior Librarians perform their duties using the LPSS. They return many books to the original locations and collect many books for general user every day. They can use LPSS to improve efficiency or control a library robot to do these jobs.

#### External Hardware

The LPSS interacts with the following significant external hardware:

**Client** **Hardware**:

* **Employee** **Computers**, which are the personal computers used by employees.
* **Library** **Guide** **Computers**, which are the personal computers in the library, users use it to access LPSS.

**Networks**:

* **Internet**, which is the global network used for communication among employees, users, and the LPSS.
* **Local** **Area** Network, which is the internal local area network used for communication among employees and the LPSS.

#### External Roles

The LPSS interacts with the following significant client roles:

**General** **users**, who are any LPSS users except employees, may be students or staffs of university.

**Employees**, who are any actors who works for the library:

* **Junior** **Librarian**, who works for the library (maybe temporary workers). They return books to the right locations and collect books for general user every day. They can use LPSS to improve efficiency or control a library robot to do these jobs.
* **Senior** **librarian**, who manages Junior Librarians and important work of the library. Senior librarian can manipulate the database of books in LPSS. (e.g. change the locations of books on bookshelves)
* **Database** **Administrator**, who updates, manipulates and maintains all databases of LPSS. (e.g. update the maps of library)
* **Curator**, who has the highest authority. Curator can manage other employees.

#### External Software

The LPSS interacts with the following significant client software:

* **Browser**, which is the software tool that runs on computers in the library that allow users to communicate over the networks with the LPSS.

#### External Systems

The LPSS interacts with the following significant external systems:

* **Library** **robot** **control** **system**, which is the system in a robot to control its actions.
* **Path** **List** **printing** **system**, which communicates LPSS and a printer to print out path list in paper.

### Library Path Search System Capabilities

This subsection provides a high-level overview of major capabilities of the LPSS.

#### Summary of System Capabilities

##### Employee Capabilities

The LPSS will provide the following capabilities to employees:

* **Junior** **Librarian**. The LPSS Junior Librarian to:
* Search books
* Print path list.
* Check shortest path.
* Generate operation list.
* Maintain To-do list.
* Control library robot.
* Search users.
* **Senior** **Librarian**. The Senior Librarian has all authorities of Junior Librarian, and:
* Manage Junior Librarian.
* Maintain books (add, modify, delete).
* **Database** **Administrator**. The LPSS will enable database administrator to:
* Search books.
* Maintain books (add, modify, delete).
* Maintain Maps (add, modify, delete).
* **Curator**. The LPSS will enable the curator to:
* Maintain librarians (query, add, modify, delete).

##### Users Capabilities

The LPSS will provide the following capabilities to users:

* General Users. The LPSS will enable all general users to:
* Maintain book list.
* Search books.
* Print path list.
* Check shortest path.
* Generate operation list.

##### Other Capabilities

* **Path** **List** **Print** **System**. The LPSS will use a printer to print path list:
* **Library** **Robot** **Control** **System**. The LPSS shall connect to a library robot if there is one.

## Functional Requirements

The section of the SRS specifies the functional requirements of the LPSS in terms of use cases and their associated use case paths. The use case model is primarily organized in terms of the externals that benefit from the use cases.

### Summary Use Case Diagrams

The following use case diagrams summarize the functional requirements for the LPSS:

|  |
| --- |
| C:\Users\Neo\Desktop\all user(new).png |

### External Roles

This subsection describes and specifies external roles, the associated responsibilities, and all use cases primarily driven by these externals.

#### General User

##### Definition

General users are any LPSS users except employees, may be students or staffs of university.

Use case diagram

|  |
| --- |
| **general user** |

##### Requirement

* **Search books.**

The LPSS shall enable general users to search information of books and provide further operations.

* **Maintain book list.**

The LPSS shall enable general users to maintain book list, including import personal book list from library system and tick books from the book list to add them into an operation list.

* **Generate operation list**

The LPSS shall enable general users to generate operation lists. The LPSS search the shortest path to collect books on one operation list.

* **Check shortest path**

The LPSS shall display the shortest path on the map (screen) and enable the general users to check each location step by step.

* **Print path list**

The LPSS shall enable general users to print a path list. The path list lists each book with its location and return/collect order of an operation list.

#### Junior Librarian

##### Definition

Junior Librarians is employee who works at help desk in the library. They use LPSS to find the shortest path to return and collect books. They can also use LPSS to control a library robot if there is one.

##### Use case diagram

|  |
| --- |
| C:\Users\Neo\Desktop\Junior librarian.png |

##### Requirement

* **Search books**

The LPSS shall enable Junior Librarian to search information of books and provide further operations.

* **Maintain To-do list**

The LPSS shall enable Junior Librarian to maintain his/her To-do list. A librarian To-do list lists the books the librarian shall return to the locations of bookshelves and the books the librarian shall collect from the bookshelves.

* **Generate operation list**

The LPSS shall enable Junior Librarian to generate operation lists. A librarian can divide the To-do list into several operation lists. The LPSS search the shortest path to return and collect books on one operation list.

* **Check shortest path**

The LPSS shall display the shortest path on the map (screen) and enable the Junior Librarian to check each location one by one.

* **Print path list**

The LPSS shall enable Junior Librarian to print a path list. The path list lists each book with its location and return/collect order of an operation list.

* **Search users**

The LPSS shall enable Junior Librarian to search information of general users in order to help users and manage user borrowing states.

* **Control library robot**

The LPSS shall enable Junior Librarian to manage and control a library robot (if there is one) to return and collect books along with the calculated shortest path,

#### Senior Librarian

##### Definition

Senior librarian is employee who works in the office of the library. They manage Junior Librarians and important work of the library. Senior librarian can manipulate the database of books in LPSS. (e.g. change the locations of books on bookshelves)

##### Use case diagram

|  |
| --- |
| C:\Users\Neo\Desktop\Senior librarian(new).png |

##### Requirement

* **Manage Junior Librarian**

The LPSS shall enable senior librarian to manage Junior Librarians. They can help to query the status of the account.

* **Maintain books (add, modify, delete).**

The LPSS shall enable senior librarian to maintain database of books, for example, add, modify or delete data of books in database.

#### Database Administrator

##### Definition

Database administrator is responsible for updating, manipulating and modifying the databases of the books and the library maps.

##### Use case diagram

|  |
| --- |
| database admin |

##### Requirement

* **Search books**

The system shall enable database administrator to search information of books and provide further operations.

* **Maintain books**

The LPSS shall enable database administrator to maintain book database of the library including addition, modification and deletion.

* **Maintain Maps**

The LPSS shall enable database administrator to maintain map database of the library including addition, modification and deletion.

#### Curator

##### Definition

Curator is the manager with the highest authority of the library. A curator manages the accounts of all employees, for example, addition, deletion and querying of employees’ accounts.

Use case diagram

##### Use case diagram

|  |
| --- |
| curator |

##### Requirement

* + **Maintain librarian (query, add, modify, delete).**
* Query

The LPSS shall enable the curator to query the employee list as well as their information.

* Add accounts

The LPSS shall enable the curator to add new accounts.

* Modify accounts

The LPSS shall enable the curator to modify the data of employees such as changing authorization and information.

* Delete accounts

The LPSS shall enable the curator to delete accounts.

## Data Requirements[1]

### User view:

|  |  |
| --- | --- |
| User | Requirements |
| General user | To maintain (enter, update, and delete) data on personal information.  To maintain (enter, update, and delete) data on books of own booklist.  To perform searches on books.  To report on all books of own booklist. |
| Junior Librarian | To maintain (enter, update, and delete) data on personal information.  To perform searches on books.  To perform searches on all the general users.  To track the status of books.  To report on general users. |
| Senior Librarian | To maintain (enter, update, and delete) data on personal information.  To maintain (enter, update, and delete) all the books on the shelves.  To perform searches on books.  To perform searches on all the general users.  To track the status of books.  To report on general users.  To report on the locations of books. |
| Database Administrator | To maintain (enter, update, and delete) data on personal information.  To maintain (enter, update, and delete) all the locations of books.  To maintain (enter, update, and delete) the locations of bookshelves.  To perform searches on books.  To track the status of books.  To report on the locations of books and shelves. |
| Curator | To maintain (enter, update, and delete) data on personal information.  To maintain (enter, update, and delete) data on all librarian.  To report on all the librarians. |

### Data requirements:

* + The data held on the general users or staffs is made up of personal information (name, ID, major, address, telephone numbers, password) and borrowed books.
  + The data held on the librarians is made up of personal information (name, ID, major, address, telephone numbers, password).
  + The data held on a library is made of information of books(Title, barcode, status, call number and location, copies)

### Initial database size:

* There are approximately 1,000,000 books.
* There are approximately 20 bookshelves on each floor and there are five floors. There is an average of 1,000 and a maximum of 1500 books for each shelf.
* There are approximately 10000 general users registered across all the university.
* There are approximately 20 librarians (Junior Librarian -13, Senior Librarian –4, Database Administrator -2, and Curator 1).

### Database rate of growth:

* Approximately 100 new books and 10 copies of each book are added to the database each month.
* Approximately 2 members of librarians join and leave the library each month. The records of librarians who have left the library are deleted after one year.
* Approximately 1000 general users join and leave the library each year. The records of them who have left the library are deleted (optional).

### Performance:

* During opening hours but not during peak periods expect less than 1 second response for all singe record searches. During peak periods (9AM-5PM) expect less than 5 second response for all singe record searches.
* During opening hours but not during peak periods expect less than 5 second response for all multiple record searches. During peak periods (9AM-5PM) expect less than 10 second response for all multiple record searches.
* During opening hours but not during peak periods expect less than 1 second response for all updates/saves. During peak periods (9AM-5PM) expect less than 5 second response for all updates/saves.

### Security:

* The information of login with invalid password exceed three times will be recorded.
* The password should encrypt.
* Each member should be assigned database access privileges appropriate to a particular user view, namely General User, Junior Librarian, Senior Librarian, Database Administrator, Curator.

### Backup and recovery:

The database should be backed up each day at 12 midnight.

### Legal issues:

Each library has laws that govern the way that the computerized storage of personal data is handled. As the library database holds data on members, any legal issues that must be complied with should be investigated and implemented.

## System Quality Requirements

This section specifies the required system quality factors that are not related to the specific functional requirements documented in the use case model.

### Correctness

This subsection specifies the following requirements concerning the degree to which the system can contain defects and still be acceptable to the customer.

#### Accuracy

The degree of correctness of the system’s outputs:

* COR1 - Founded paths shall be correct to the shortest.
* COR2 - Locations of books shall be correct to the nearest position on the bookshelf.

#### Precision

The resolution of the system’s numerical outputs:

* COR3 – Founded paths shall be close to the shortest one.
* COR04 – Location of books shall be close to the nearest position on the bookshelf.

#### Timeliness

The degree to which the system must ensure that its persistent information is current:

* COR5 - Once one user updates some data, the system shall ensure that other users shall automatically see the updated data within 1 seconds.
* COR6 - The system shall automatically transfer old version information from on-line database to off-line archives after 24 hours.
* COR7 - The system shall permanently delete “obsolete” information from all databases after 7 days.

### Efficiency

This subsection specifies the following requirements associated with the degree to which the system effectively uses its resources:

* EFF1 - The LPSS shall not require users to download software to their computers.
* EFF2 - The LPSS shall not require users to input much information to search a book.
* EFF3 - The LPSS shall display the shortest path in 5 seconds after users’ queries.

### Extensibility

This subsection specifies the following requirements associated with the degree to which the system can be modified to meet changing requirements or goals.

* EXT1 - The LPSS shall be easily extendable to connect to a library robot.
* EXT2 - The LPSS shall be easily extendable to add database from other libraries.

### Interoperability

This subsection specifies the following requirements associated with the ease with which the system can be integrated with other system.

The LPSS shall interoperate with the following browsers:

* IOP1 - Internet Explorer
* IOP2 - Firefox
* IOP3 - Google Chrome

### Maintainability

This subsection specifies the following requirements associated with the ease with which the system can be maintained:

* MAI1 - The LPSS shall permit the upgrade of database without down time.
* MAI2 - The LPSS shall permit the upgrade of software without down time.
* MAI3 - The Mean Time To Fix (MTTF) shall not exceed one day.

### Performance

This subsection specifies the following requirements associated with the speed with which the system shall function.

#### Capacity

The minimum number of objects that the system can support:

* PER1 - The system shall support a minimum of 2,000 simultaneous interactions.
* PER2 - The system shall support a minimum of 20,000 accesses every day.
* PER3 - The system shall support a minimum of 10 employees.
* PER4 - The system shall support a minimum of 50,000 users.

#### Latency

The maximum time that is permitted for the system to execute specific tasks:

* PER6 - The typical user shall be able to find a book within 1 minutes.
* PER7 - The typical user shall be able to import a book list within 3 minutes.
* PER8 - The typical user shall be able to find a shortest path within 5 minutes.

#### Response Time

The maximum time that is permitted for the system to respond to requests:

* PER9 - All system responses shall occur within 30 seconds.

### Portability

This subsection specifies the following requirements associated with the ease with which the system can be moved from one environment to another.

The LPSS shall enable users to use the following environments to interact with the LPSS:

* Operating Systems:
* POR1 - Windows XP
* POR2 - Windows 7
* POR3 - Mac OS X

### Reusability

This subsection specifies the following requirements associated with the degree to which the system can be used for purposes other than originally intended.

* REU1 - The LPSS shall incorporate a database continuous availability layer.
* REU2 - The LPSS shall reuse common classes.
* REU3 - The LPSS shall reuse existent web functions of the library.

### Robustness

This subsection specifies the following requirements associated with the degree to which the system continues to properly function under abnormal circumstances.

* ROB1 - The LPSS should gracefully handle invalid input.
* ROB2 - The LPSS should gracefully handle database failures.

### Safety

This subsection specifies the following requirements associated with the degree to which the system does not directly or indirectly cause accidental harm to life or property.

* SAF1 - The LPSS shall not accidentally lose user account information.
* SAF2 - The LPSS shall not permit access with wrong account information.
* SAF3 - The LPSS shall not provide functions to user who has not relevant permissions.

### Usability

This subsection specifies the following requirements associated with the ease with which the system can be used.

* USE1 - The application shall enable at least 90% of a statistically valid sample of representative novice users to:
* Login within 3 minutes.
* Generate an operation list with 5 books within 15 minutes.
* Check the shortest path and print a path list within 10 minutes.
* USE2 - The application shall enable at least 90% of a statistically valid sample of representative experienced users to:
* Login within 1 minute.
* Generate an operation list with 5 books within 5 minutes.
* Check the shortest path and print a path list within 5 minutes.
* USE3 - The average user shall be able to learn to use LPSS to search path within 10 minutes.
* USE4 - The average Junior Librarian shall be able to learn to use LPSS to manage the library within 30 minutes.
* USE5 - The average senior librarian shall be able to learn to use LPSS to manage the library within 45 minutes.
* USE6 - The average database administrator shall be able to learn to use LPSS to manage the database of the library within 120 minutes.
* USE7 - The average curator shall be able to learn to use LPSS to manage the account information of staffs within 30 minutes.
* USE8 - The typical user shall be able to easily and quickly navigate between relevant web pages.

# System models

In this section, system model diagrams are built for the purpose of showing the relationships between the system components and the system and its environment. These diagrams give a clear and brief description on the boundary of the system, the structure of the system and the functionalities provided by the system.

*[See Appendices to check the diagrams.]*

## Boundary Diagram

As the carrier of the core system is a library management system, the system boundary is concerned as a set of all the systems that interact with the core system. To model the system boundary, a boundary diagram is designed.

The boundary systems are:

* Web site server system: interact with the client directly
* Maintenance system: it has the responsibility of maintain the stability, consistency and the recoverability of the whole system
* Security system: this system ensures the security of data transmission between client and server or application system and database.
* Application tools system: this is the system that provides some mathematical aid, map generation environment and decision advices.
* Database management system and usage database system: this is the library database system which provide by this library management system or it also could be some extern connected database system
* External connection system: a coordinator between the whole system and some external connected system

## Component Diagram

The system can be decomposed into three main components; each component consists of several smaller components.

* Web site server system: interact with the client directly
* Application tools system:
* User management system: it has the responsibility of management any activities of client it also act as a coordinator for client to access to the other two system components
* Shortest path finding system: the core system
* Statistical aid system: provide some statistical information base on some real-time data in database
* Database system
* Security system: this system ensures the security of data transmission between client and server or application system and database
* Database management system and usage database system: this is the library database system which provide by this library management system or it also could be some extern connected database system

## Activity Diagram

To illustrate the functionalities provided y by the system, an activity diagram is designed. As the complicity of the system, the activity diagram is decomposed designed from the point of view of different kinds of users

* general user
* Junior Librarian
* senior librarian
* Database administrator
* Curator

The available functionalities for each kind of user has been illustrated in user views section

To avoid description duplicate and for the purpose to simplify the activity diagram, some “shared functions” (functionality is accessible by different kinds of users e.g. log in, book item search…..) are omitted or abstracted in a higher description in some activity diagram. For example, Junior Librarian can search and view general user’s account information, so is senior librarian. To simplify the diagram, this function is only described in detail in Junior Librarian activity diagram and abstracted as single activity in the senior librarian activity diagram.

## Future consideration

As time constrains in the requirement engineering, the relationships between system components is only illustrate as a high level abstract component diagram. A more detail sequence diagram with corresponds component diagram and class diagram design should be put on the time table for more precise and effective software design.

# System evaluation

## Software Evolution:

Software evolution is used to develop the initially software, and updating it repeatedly for various reasons. Software evolution is to change the system to correspond to business user and customer needs. The system’s environment may change as new hardware is introduced. Errors, undiscovered during system validation may emerge, requiring repair.

The discussion of software evolution includes analyzing fundamental assumptions, recognizing program evolution dynamic, implementing evolution process, and discussing software maintenance and cost.

The architecture of the system is designed as aggregation of highly independent components. This design pattern corresponds to the high cohesion and low coupling attributes of each component. Consequently, each component of the system is easy to change, convenient to maintenance and easy to transfer to other systems.

### Fundamental Assumptions

This system involves in a list of fundamental assumptions on which the system is based to support the implementing of the system.

The carrier of the system is a library management system which contains all the boundary system (discussed in system model section) that interacts with the core system.

### Program Evolution Dynamic

The study of system change is known as program evolution dynamics. Lehman’s Laws [2] have been proposed based on software engineering observations. Each section of them is discussed as below.

#### Continuing change

A program that is used in the real-world environment necessarily must change or become progressively less useful in that environment. [2]

It is not surprise that the system will be changed rapidly when it come into service in different real-world environments. The software prototype cannot be used in real- world directly as different environment may have completely different requirement and constrains. Consequently, modifications are needed to satisfy these constrain to make the system suits well in particular environment. For example, different libraries may have different user hierarchy and as consequence different user management system is needed.

#### Increasing complexity

During the changing of the evolving system component, the structure of the system tends to become more complex because it must devoted extra resources and simplifying the structure. [2] For future consideration, the core system can also be used in other complete different environment like warehouses. In this environment, constrains may more rigorous which lead to the complicity of the system increased dramatically. The complexity of the system and software programing is even more complex when the core system is transferred and embedded in an automatic forklift.

#### Large program evolution

Program evolution is a self-regulating process. System attribute such as size, time between releases and the number of reported errors is approximately invariant for each system release. [2] When this process is come into real practice, besides the basic algorithm concept, the evolution may overturning system as different system release has different constrains (e.g. programing language, user hierarchy, system architecture) or requirement.

#### Organizational stability

Over a program’s lifetime, its rate of development is approximately constant and independent of the resources devoted to the system development. [2]

The development model of this library management system is considered as a combination of spiral model, stepwise refinement model and evolutionary model. The resources devoted to the system development are all awarded.

#### Conservation of familiarity

Over the lifetime of a system, the incremental change in each release is approximately constant. [2]For this library management system, it can be predicted that the only development process is database update and user management system reconfiguration. It is clear that the development rate is approximately a small constant

#### Continuing growth

The functionality offered by systems has to continually increase to maintain user satisfaction. [2] As the software architecture is designed as an aggregation of highly independent components, it is simplify the function increment process. However, for a completely different environment, the change of system architecture may still overturning and the growth rate may also various

#### Declining quality

The quality of system will appear to be declining unless they are adapted to changes in their operational environment. [2] For several of reasons, such as technique problem or budget problem, it is impossible to make the system self-adaptive. Consequently, to prevent quality declining, make the system adapted to changes in their operational environment is required to perform manually.

## Software Maintenance

Lienz and Swanson[3] catalogued four categories of software maintenance, which are corrective maintenance, adaptive maintenance, performance or maintainability and preventive maintenance. During the process of maintenance, the particular form of the system brings benefit to detect and correct problems. At the same time, it can save time.

### Corrective maintenance

When discovering problems, the system needs to correct the problems firstly. Then, it is necessary to reactive the program to check the performance of the modification system.

### Adaptive maintenance

To make the system usable in changing environment, the performance of the delivery system need to be modified. For example, if it changes the library floor from two to three, then the map should switch to the third floor with the corresponding information that should change at the same time.

### Perfective maintenance

After delivering the system, in order to improve the performance and maintainability, it may be necessary to modify the system product. For instance, the storage shall need to enlarge, if the size of the modified system have magnified.

### Preventive maintenance

The preventive maintenance has the ability to detect and correct the potential faults which are implemented by modify the software product. For example, when delivering the system of library or warehouse, analyzing the latent problem it may arise and then change the product appropriately to collect the faults and correct them.

## Evolution Process

The evolution process is an iteration of the development process which mainly involving in the fundamental activities of change analysis, release planning, system implementation and releasing a system to customers. The components of LPSS are highly independent which makes the system easily to implement evolution process.

### Analysis

The analysis of evolution is considered before implementing the system which is a process of value evaluation. During releasing the program, all proposal changes including fault repair, adaptation, functionality and costs to make sure the evolution is worthy. In planning stage, the developer must make sure that major faults can be repaired and the new system with new functions will be accepted. In addition, it is necessary to evaluate the average time and money required for impact analysis and verify a change request list.

### Description

First step of evolution is changing requests elicitation and analysis which come from the discussions between the system users and developers.

The second step is analyzing the feasibility of these requests. A very difficult part for developer is adding delivery function to the existing system without affecting other functions in the system.

Evolution process is an iterative process and there are variable ways to achieve evolution. While adding functions to the system, it is iterative operation of designing, implementation, testing.

### Further Considerations

Because each modular of the system is relatively independent, many useful components can be devoted to other systems. Such as, the core of this system, which applies special heuristic algorithm to help users find a best solution. Besides using components in this system, it can be derived to other systems involved in path searching. What’s more, the other parts of programs also can be used to other systems referring to corresponding functions. This can improve reuse of applications. The library management system is only a carrier of the core system. That means this system can be transferred or even embedded in various kinds of carrier system. For example, this path searching system can also be embedded in an automatic forklift in factory warehouse. It even can be applied in mobile aid software to help the post man find the shortest path.

# Reference

[1] Thomas M. Connolly, Carolyn E. Begg, *Database Solutions*, Addison Wesley, 2004.

[2] Ian Sommerville, *Software Engineering*, Assison-Wesley, 2006, p491.

[3] Lientz, B.P. and Swanson, E.B., *Software Maintenance Management*, A Study Of The Maintenance OfComputer Application Software In 487 Data Processing Organizations. Addison-Wesley, Reading MA, 1980. ISBN 0201042053