**Proposal**

**For**

**MyLibrary**

**Prepared by**

|  |  |
| --- | --- |
| Iman Bin Baharuddin | 192658 |
| Fatin Nabilla binti Muhamad Zaini | 193576 |
| Imran Bin Zakaria | 192659 |
| Muhammad Izwan Bin Suhaime | 193907 |

**TABLE OF CONTENT**

[**1.0 Introduction**](#_s5v8zqfv8p3h) **3**

[1.1 Purpose](#_c3dh0fs3gg5u) 3

[1.2 Scope](#_csoceudeup78) 3

[1.3 Definitions, Acronyms and Abbreviations](#_aqnmyrasha9k) 4

[1.4 References](#_osubijiz381j) 4

[**2.0 Overall Description**](#_oqf8a4nvdixy) **5**

[2.1 User Classes and Characteristics](#_3wjsrka1ub4m) 5

[2.2 Operating Environment](#_kg6e79tp3cnv) 6

[2.3 Constraints](#_w81el6nmbkee) 6

[2.4 Assumptions and Dependencies](#_yf21bzykjsr9) 6

[**3.0 External Interface Requirements**](#_sy0hcxmuv64q) **7**

[3.1 User Interfaces](#_rhegpy1repvz) 7

[3.2 Hardware Interfaces](#_fgo9u3gu4yqd) 12

[3.3 Software Interfaces](#_ol05q5xny26f) 12

[3.4 Communication Interfaces](#_3h5irirg8sw) 12

[**4.0 Functional Requirements and System Use Case**](#_x0t3v0d6w3sy) **13**

[4.1 Functional Requirements](#_vf36q8317xv7) 13

[4.2 Use Case Diagram](#_e6y0aq6adxd2) 14

[4.3 Use Case Description](#_ptx1gsqcup7l) 15

[**5.0 Non Functional Requirements**](#_oflgbwccmm80) **23**

[**6.0 Entity Relationship Diagram**](#_z2wsop5eaj5d) **24**

# 

# 1.0 Introduction

## 1.1 Purpose

The purpose of MyLibrary system is to provide a systematic way to manage and store book information. The system will help both students and library staff to keep a constant track of all books in the library. With this system, all work will become much easier as managing books can be done online. Students will be able to view all the list of books available in the library. The traditional way of managing library is much harder to keep track of things and are prone to human errors. These errors can be avoided by using the system to keep track of information systematically and thus it will help in avoiding making mistakes.

## 1.2 Scope

MyLibrary will be used mainly in institutions such as universities. This system will enable both librarian and students to have direct access to information about books in the library online. This system will help students to check the availability of the books via online. Besides, the librarian can also add new books and update the information about the books. This is seen to be practical compared to manual way. In this documentation students will be referred as borrower.

## 1.3 Definitions, Acronyms and Abbreviations

Table 1: Definition of Terms

|  |  |
| --- | --- |
| Term | Definition |
| Borrower | Registered user for borrowing books |
| System | MyLibrary which helps librarians and students to manage books easily. |
| Librarian | User who works in the library and has additional privilege. |

## 1.4 References

1. Software Requirements Specification IEEE Std 830-1998. <https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=720574&tag=1>

### 

# 2.0 Overall Description

## 2.1 User Classes and Characteristics

Borrower:

* Search for books by using titles following the categories.
* View available books in the library.
* Place a request to borrow a book.

Librarian:

* Add new books into the system.
* Update description about books in the system.
* Record the borrower details.

## 2.2 Operating Environment

MyLibrary is a web-based system. The appropriate web server and browser are required for the development. All the data will be stored in MySQL database management system. Users should be able to access the system using any type of browser such as Google Chrome, Mozilla Firefox, Internet Explorer, Safari etc. The connections to the servers will be based on the criteria of attributes of the user like his/her location and server will be working whole 24/7.

## 2.3 Constraints

The constraints in design and implementation phase are as follows:

* Language requirement: This system only caters in English.
* Browser support limitations: This system is a web-based application. There is an issue of an unsupported browser problem.
* Security consideration: All users will have their own username and password.

## 2.4 Assumptions and Dependencies

* The system does require database server MySQL for storing the username and password for different types of user of the system.
* The system also assumes that details regarding each customer would be made correctly.
* The end users of this software are assumed to have a basic level of computer knowledge i.e. point and click.
* Users must have basic knowledge of English.

# 3.0 External Interface Requirements

### 3.1 User Interfaces

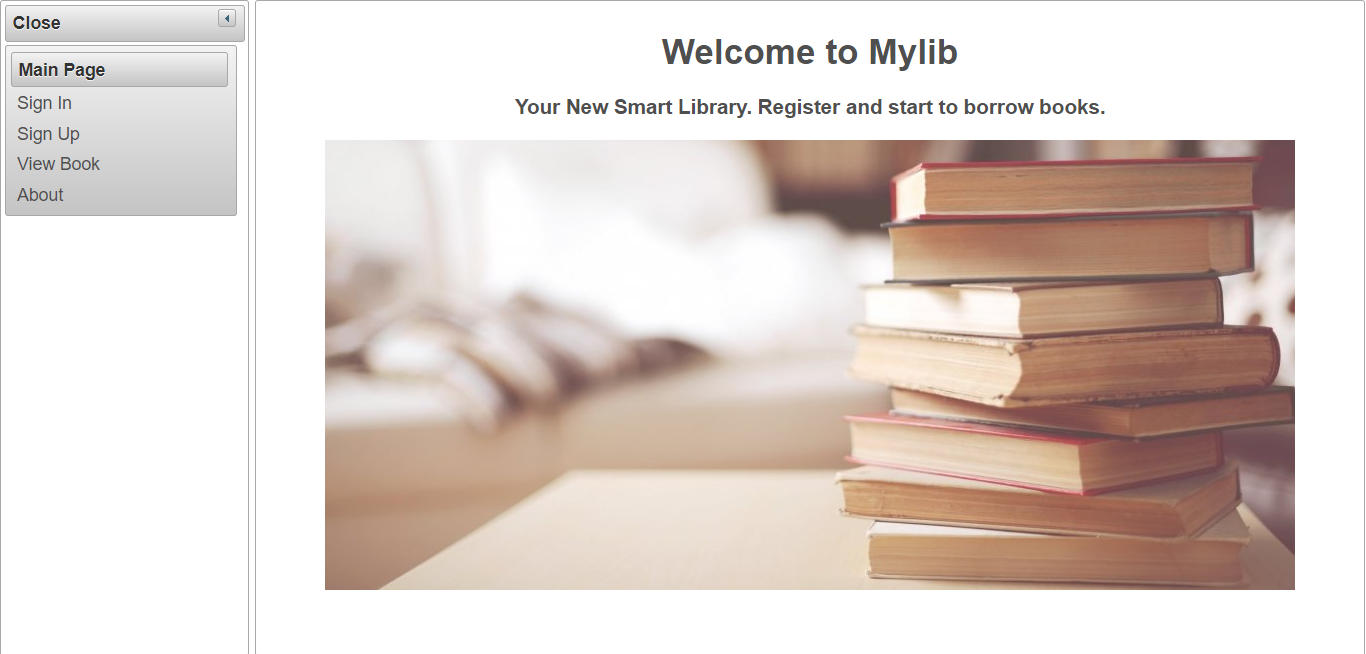


Figure 3.1.1 Main Page (Borrower)

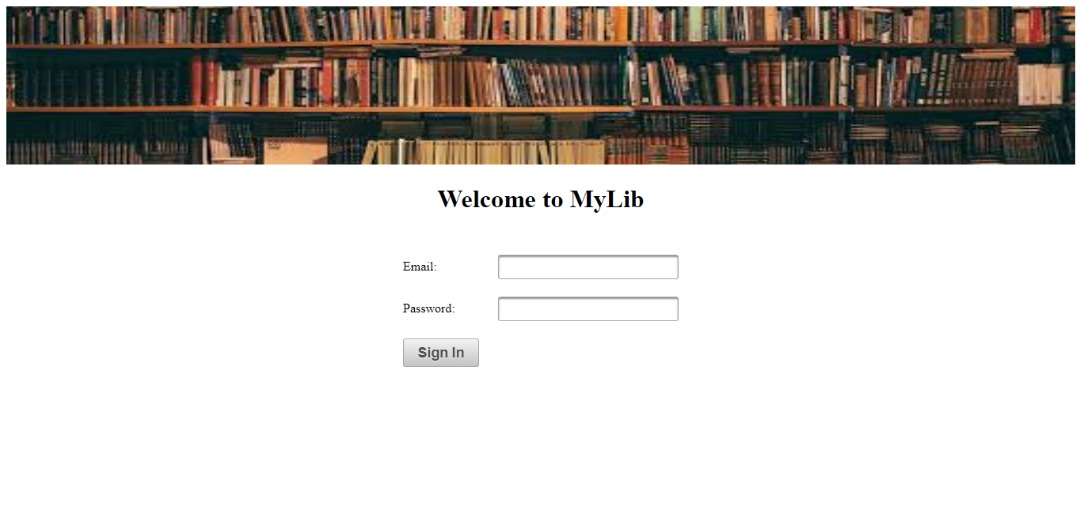


Figure 3.1.2 Log In (Librarian)

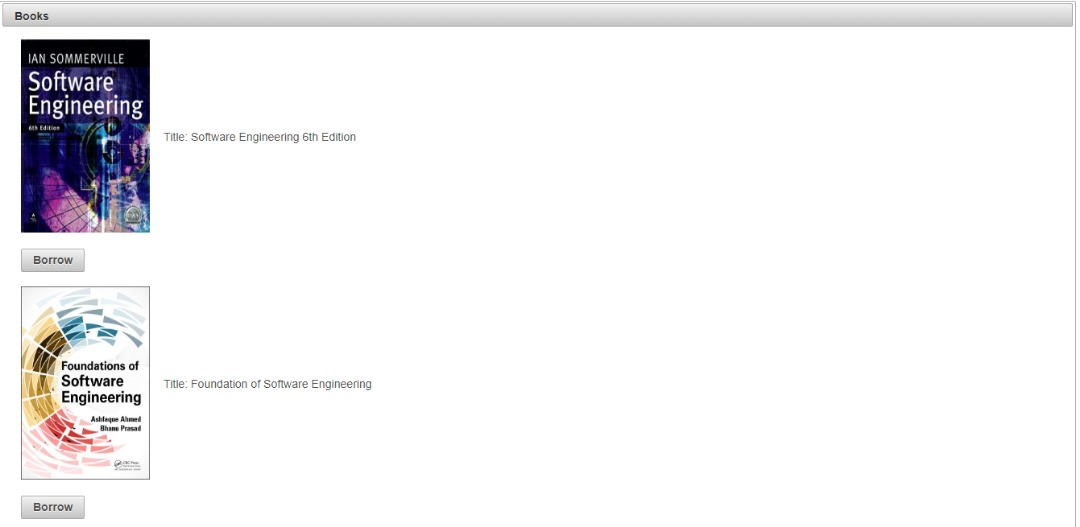


Figure 3.1.3 View List of Book (Borrower)

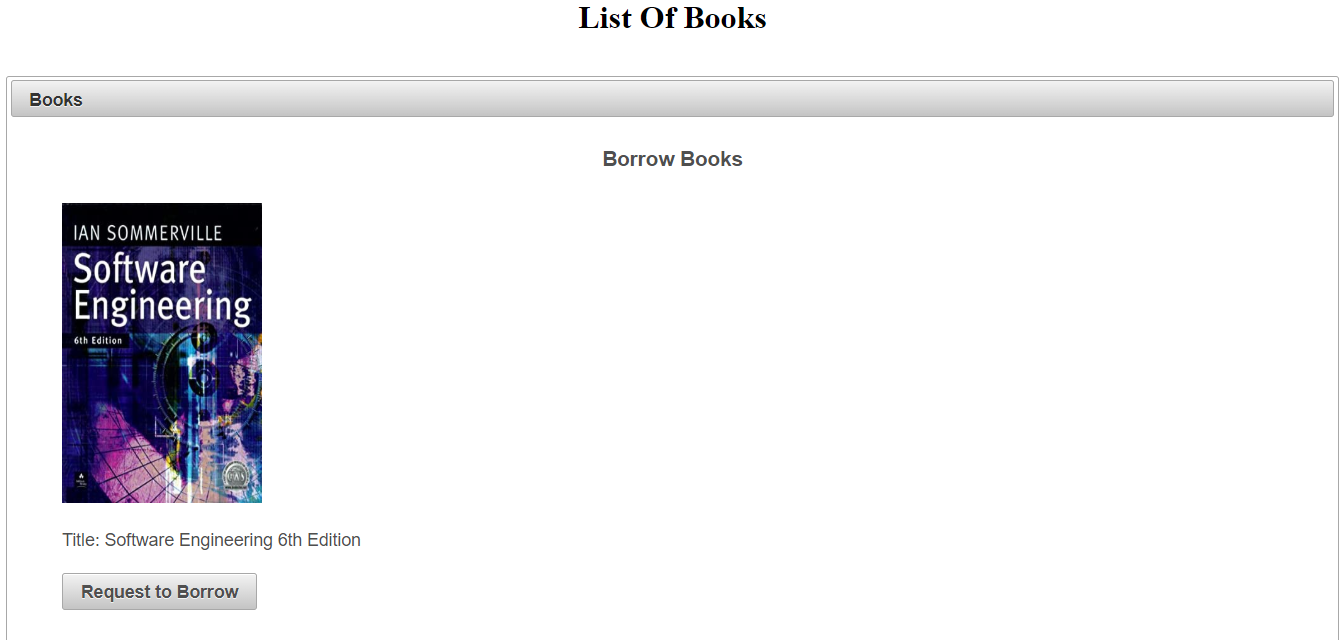


Figure 3.1.4 Borrow Book (Borrower)

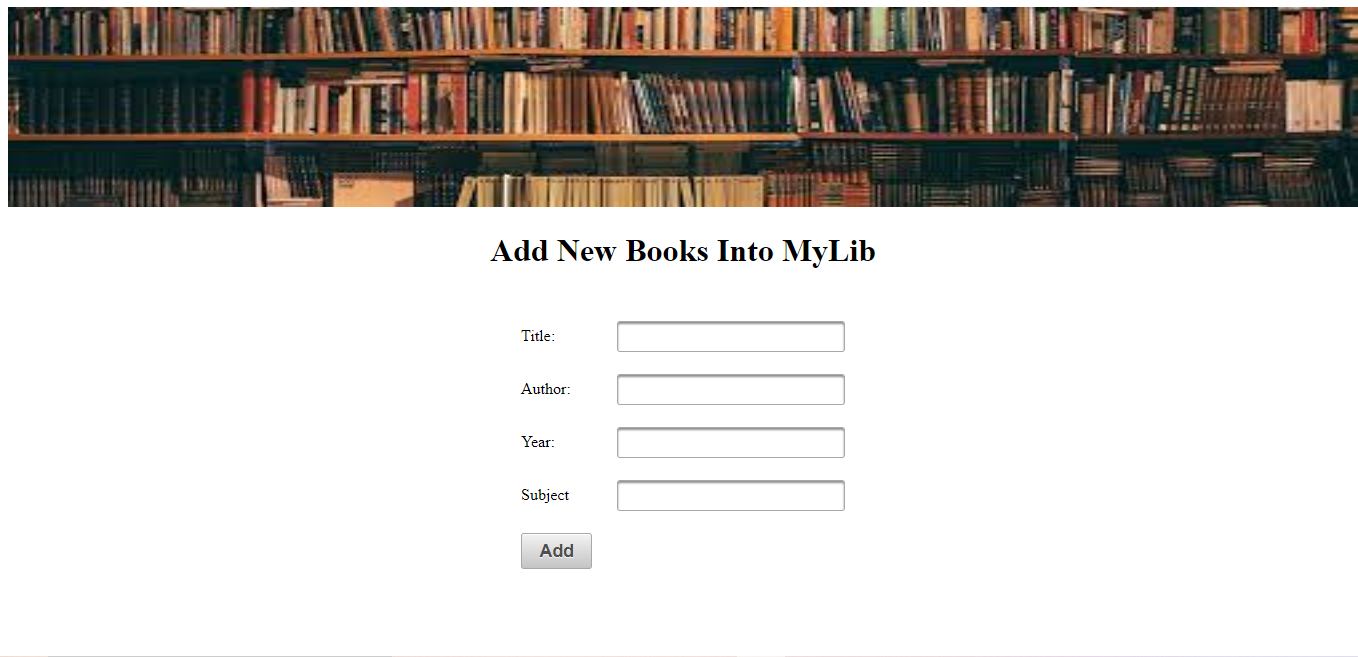


Figure 3.1.5 Add New Books (Librarian)

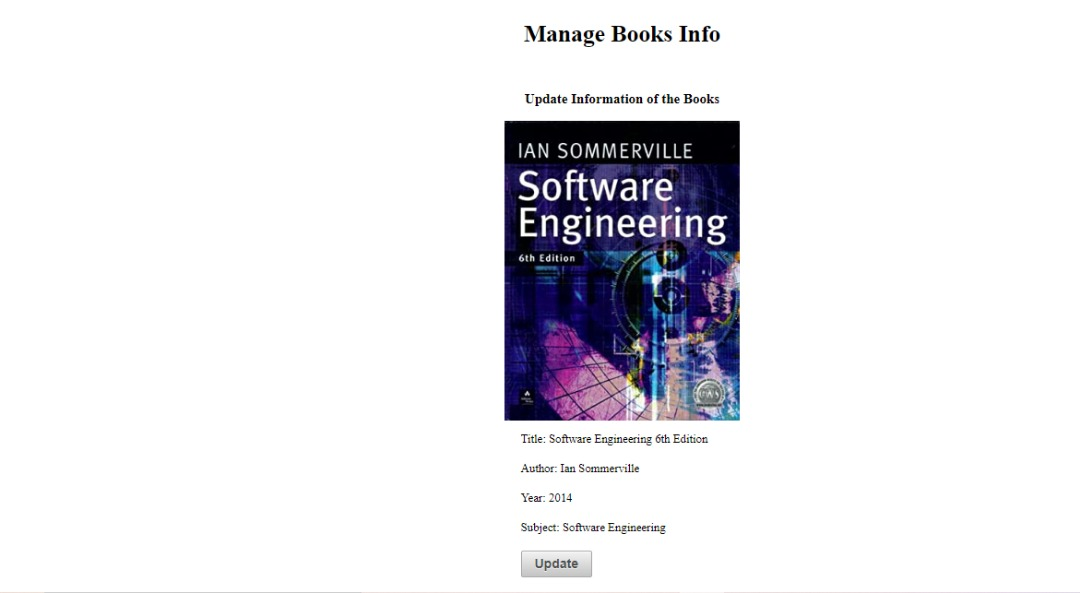


Figure 3.1.6 Manage Book Info (Librarian)

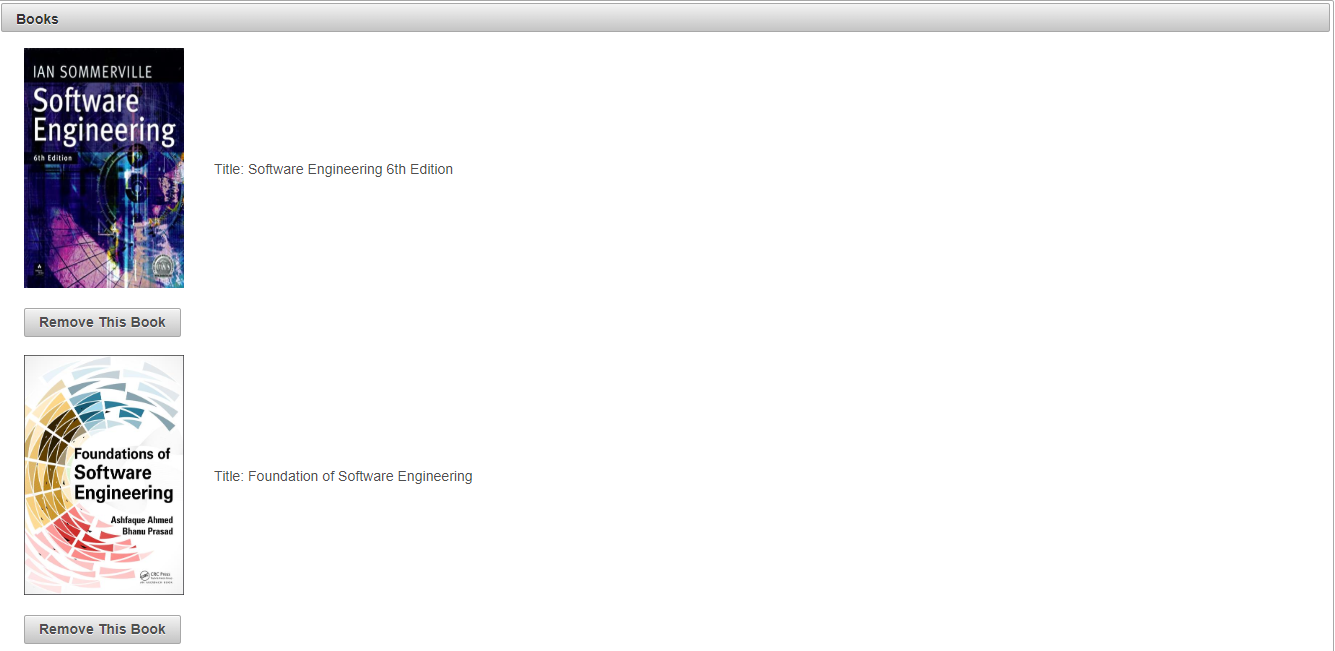


Figure 3.1.7 Remove Book (Librarian)

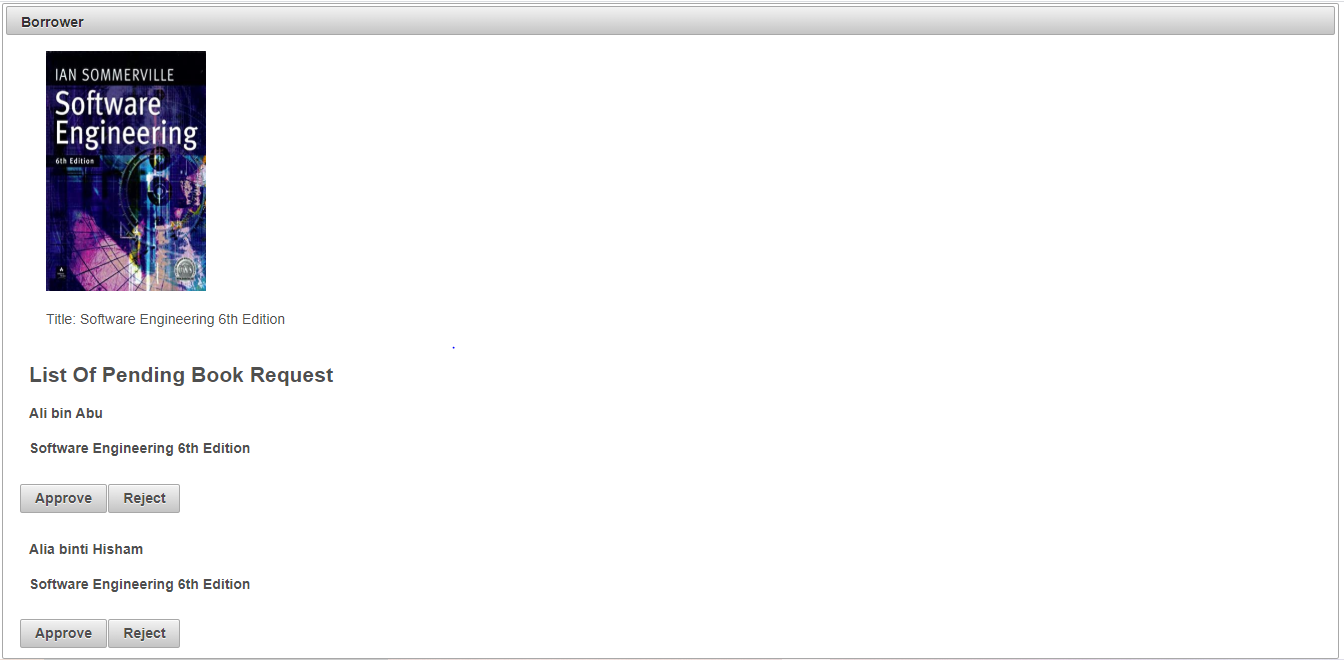


Figure 3.1.8 Manage Borrower (Librarian)

### 3.2 Hardware Interfaces

Hardware requirements for MyLibrary on internet will be same for both the parties which are follows:

* Processor: Pentium 1 or above.
* RAM: 128MB or above.
* HD: 20GB or above.

### 3.3 Software Interfaces

* Operating System: Windows.
* Development Tool: NetBeans, Primefaces, JavaServer Faces
* Database: MySQL (PhpMyAdmin).

### 3.4 Communication Interfaces

* MyLibrary system shall notify the borrower to pick up the book that they had requested to borrow.
* A list of books available will be displayed to the students when they request to view it.

# 4.0 Functional Requirements and System Use Case

## 4.1 Functional Requirements

Table 2: Functional Requirement

|  |  |  |  |
| --- | --- | --- | --- |
| Use Case ID | Use Case | Description | Actors Involved |
| SRS\_REQ\_001 | Add Book | Librarians add new books into the MyLibrary system. | Librarian |
| SRS\_REQ\_002 | Search Book | Search the available books in the system’s database. | Librarian, Borrower |
| SRS\_REQ\_003 | Log in | Log in as authorize user | Librarian |
| SRS\_REQ\_004 | View Book | View book descriptions | Librarian, Borrower |
| SRS\_REQ\_005 | Borrow Book | Borrower can borrow books | Librarian |
| SRS\_REQ\_006 | Manage Book Info | Manage the book’s information | Librarian |
| SRS\_REQ\_007 | Manage Borrower Info | Manage the borrower details and status. | Librarian |
| SRS\_REQ\_008 | Return Book | Borrower returning book to the library | Librarian |

## 4.2 Use Case Diagram

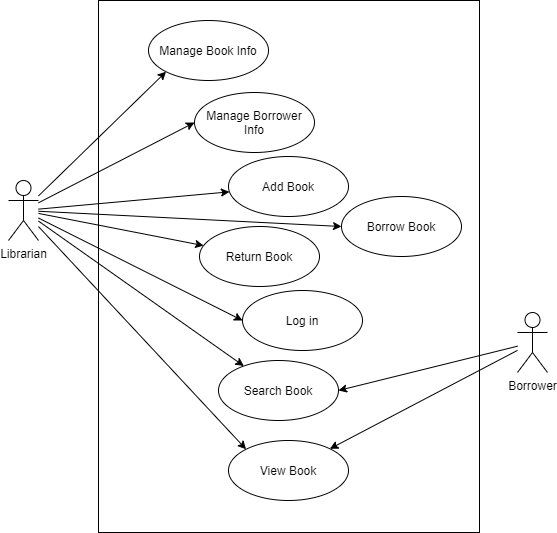


Figure 4.2.1 Use Case Diagram for MyLibrary system

## 4.3 Use Case Description

Table 3: Use Case 1: Add Book

|  |  |
| --- | --- |
| Identifier | SRS\_REQ\_001 |
| Name | Add Book |
| Objective | Librarians add new books into the MyLibrary system |
| Priority | High |
| Actor | Librarian |
| Pre-condition(s) | 1. Sign in as librarian 2. Librarian click register book button |
| Flow of Event | **Basic Flow**   1. Book id is given by continuing from the previous book id. 2. The librarian enters information about the book including the title, barcode, author etc. 3. Click register book button 4. The system displays successful message |
| Alternative Flow | 3. a) Invalid or incomplete information entered. Back to step 2. |
| Post-condition(s) | Book detail and information is registered into the system database |

### 

Table 4: Use Case 2: Search Book

|  |  |
| --- | --- |
| Identifier | SRS\_REQ\_002 |
| Name | Search Book |
| Objective | Search for book that exist in the system’s database |
| Priority | High |
| Actor | Librarian, Borrower |
| Pre-condition(s) | No pre-condition |
| Flow of Event | **Basic Flow**   1. The actors insert the title of the book in the search textbox. 2. Click the search button. 3. If the search input entered has similarity to the books in the database, the system will show to the user the list of books. 4. The information of the book will also be shown such as status, book id, bookshelf id, etc. |
| Alternative Flow | 3. a) If the search input entered has no similarity to the books in the  database, the system will display “The book is not available here” |
| Post-condition(s) | The librarian, borrower and clerk will be able to view the list of books successfully. |

Table 5: Use Case 3: Log in

|  |  |
| --- | --- |
| Identifier | SRS\_REQ\_003 |
| Name | Log in |
| Objective | Log in as authorize user |
| Priority | High |
| Actor | Librarian |
| Pre-condition(s) | The email and password is already in the system database |
| Flow of Event | **Basic Flow**   1. The users click the login button 2. The librarian enter their email and password 3. If the email and password is correct, the user will be redirect to another page |
| Alternative Flow | 3.a) if the email or password is wrong or invalid, they need to re-enter  their email or password. |
| Post-condition(s) | The user will be redirect to their respective page |

Table 6: Use Case 4: View Book

|  |  |
| --- | --- |
| Identifier | SRS\_REQ\_004 |
| Name | View Book |
| Objective | View the book descriptions. |
| Priority | Medium |
| Actor | Librarian, Borrower |
| Pre-condition(s) | 1. Use case 2 is used 2. List of books is displayed |
| Flow of Event | **Basic Event**   1. The user clicks on the book title 2. The system will display the information of the book from the database. 3. Information such as title, barcode, author, book’s location, status etc. |
| Post-condition(s) | Book description is displayed |

Table 7: Use Case 5: Borrow Book

|  |  |
| --- | --- |
| Identifier | SRS\_REQ\_005 |
| Name | Borrow Book |
| Objective | Borrow book from the list of available books in the database |
| Priority | High |
| Actor | Librarian |
| Pre-condition(s) | 1. Borrower information is already in the database 2. Logged in as librarian 3. Book to be borrow is given to the librarian |
| Flow of Event | **Basic Event**   1. Librarian enter the borrower id into the system 2. Display borrower information 3. Librarian enter the book id into the system 4. Confirmation for borrowing 5. System will display the return date of the book |
| Alternative Flow | 2. a) The borrower id is incorrect. Back to step 1.  b) The borrower id is invalid due to late return. Fine needed to be  paid to enable borrowing. Return to step 1  4. a) The book id is invalid or incorrect.  b) The book status is borrowed. |
| Post-condition(s) | Date for return book is created |

### 

Table 8: Use Case 6: Manage Book Info

|  |  |
| --- | --- |
| Identifier | SRS\_REQ\_006 |
| Name | Manage Book Info |
| Objective | Manage the book’s information. |
| Priority | Medium |
| Actor | Librarian |
| Pre-condition(s) | 1. Logged in as librarian 2. Book is in the database 3. Use case 4 is used |
| Flow of Event | **Basic Event**   1. The librarian clicks the update button 2. The librarian updates the information of the book 3. The librarian clicks the “Save” button. 4. The system shows an update message. |
| Alternative Flow | 3. a) The librarian clicks the cancel button. Update made to the book is  cancel |
| Post-condition(s) | Book information updated |

Table 9: Use Case 7: Manage Borrower Info

|  |  |
| --- | --- |
| Identifier | SRS\_REQ\_007 |
| Name | Manage Borrower Info |
| Objective | Manage the borrower details and status. |
| Priority | High |
| Actor | Librarian |
| Pre-condition(s) | Logged in as librarian |
| Flow of Event | **Basic Event**   1. Librarian click register a new borrower 2. System will create new borrower id 3. The librarian will enter the information of the borrower into the system 4. The librarian clicks the “Save” button. |
| Post-condition(s) | New borrower is created |

Table 10: Use Case 8: Return Book

|  |  |
| --- | --- |
| Identifier | SRS\_REQ\_008 |
| Name | Return Book |
| Objective | Borrower returning book to the library |
| Priority | High |
| Actor | Librarian |
| Pre-condition(s) | 1. Logged in as librarian 2. The book status is borrowed |
| Flow of Event | **Basic Flow**   1. The librarian clicks on return book button 2. Then, the librarian enters the book id 3. Display return book success |
| Alternative Flow | 3.a) The book id is invalid.  b) Book status may not be borrowed.  c) Return book is late. Fine is charged to the borrower for late.  Borrower unable to borrow book until paid |
| Post-condition(s) | The returned book status become available |

# 5.0 Non-Functional Requirements

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **QA ID** | **Quality Attributes** | **Attribute Refinement** | **ASR ID** | **Architecture Significant Requirement**  **(ASR)** |
| **1** | Performance | Response Time | 1.1 | Our Application will be interactive, and the delays involved will be less, so in every action-response of the application, there are no immediate delays and will be complete in less than 5 second. |
| **2** | Reliability | Connection | 2.1 | The system will be able to operate and be accessed by multiple users even when it is flooded with requests or if one of the servers is down. There will be a backup server to make sure the system is always connected. |
| **3** | Security | Confidentiality | 3.1 | The system will use define user type in database to authorize in order to grant them the privilege and access to the application |
| Integrity | 3.2 | The system will resist unauthorized access and report the attempts |
| **4** | Availability | No downtime | 4.1 | Our system is accessible and available from 7.30a.m. until 11.00p.m. since that is 30 minutes before and after closing hour |

# 6.0 Entity Relationship Diagram

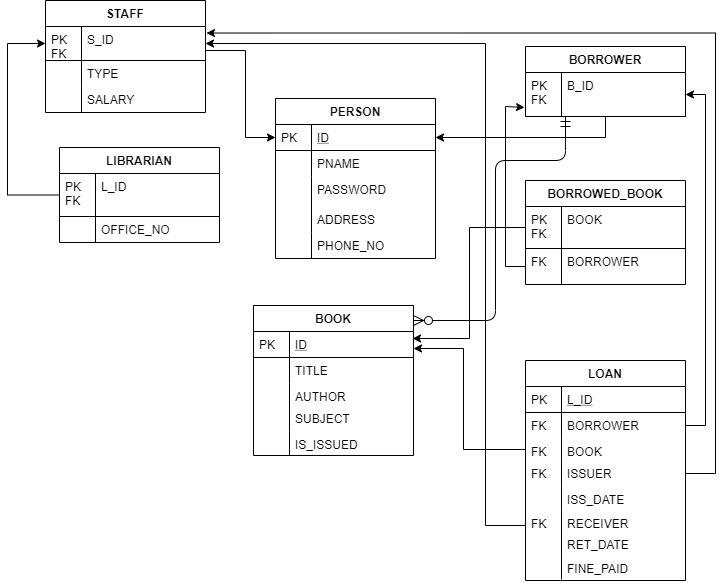


Figure 6: Entity Relationship Diagram