**CONTENTS**

[I. OVERVIEW 3](#_Toc533680048)

[II. INTRODUCTION 4](#_Toc533680049)

[1. Purpose 4](#_Toc533680050)

[2. Scope 4](#_Toc533680051)

[3. Document description 4](#_Toc533680052)

[III. USE CASE MODEL 5](#_Toc533680053)

[1. Use Case diagram 5](#_Toc533680054)

[*1.1. General 5*](#_Toc533680055)

[*1.2. Manage system 5*](#_Toc533680056)

[*1.3. Manage information about books 6*](#_Toc533680057)

[*1.4. Return and borrow books 6*](#_Toc533680058)

[*1.5. Search 7*](#_Toc533680059)

[*1.6. Manage readers 7*](#_Toc533680060)

[2. Scenario 8](#_Toc533680061)

[3. Use case descriptions 8](#_Toc533680062)

[*3.1. Use case: “See information of books and readers” 8*](#_Toc533680063)

[*3.2. Use case: Manage system 9*](#_Toc533680064)

[*3.3. Use case Manage employees 10*](#_Toc533680065)

[*3.4. Use case Manage books 12*](#_Toc533680066)

[*3.5. Use case Manage readers 14*](#_Toc533680067)

[*3.6. Use case Manage book’s kind 15*](#_Toc533680068)

[*3.7. Use case Manage publishing company 17*](#_Toc533680069)

[*3.8. Use case Borrow and return books 19*](#_Toc533680070)

[*3.9. Use case Search 22*](#_Toc533680071)

[*3.10. Use case Report 23*](#_Toc533680072)

[IV. SYSTEM ARCHITECTURE 24](#_Toc533680073)

[1. Program architecture 24](#_Toc533680074)

[2. Detailed description 25](#_Toc533680075)

[3. Functional decomposition diagram 28](#_Toc533680076)

[V. DATABASE DESIGN 29](#_Toc533680077)

[1. Entity – Relationship Diagram (8 entities): 29](#_Toc533680078)

[2. Logical Database Design 30](#_Toc533680079)

[3. Physical Database Design 30](#_Toc533680080)

[VI. USER INTERFACE DESIGN 33](#_Toc533680081)

[1. Main Interface Form 33](#_Toc533680082)

[*1.1. Login Form 33*](#_Toc533680083)

[*1.2. Personal information Form 34*](#_Toc533680084)

[*1.3. Change the password Form 34*](#_Toc533680085)

[*1.4. Employee information Form 34*](#_Toc533680086)

[2. Management Form 35](#_Toc533680087)

[*2.1. Employee management Form 35*](#_Toc533680088)

[*2.2. Reader management Form 35*](#_Toc533680089)

[*2.3. Book management Form 35*](#_Toc533680090)

[*2.4. Author management Form 36*](#_Toc533680091)

[*2.5. Kind of book management Form 36*](#_Toc533680092)

[3. Borrow- Return Form 37](#_Toc533680093)

[*3.1. Borrow information Form 37*](#_Toc533680094)

[*3.2. Return information Form 37*](#_Toc533680095)

[4. Look up Form 37](#_Toc533680096)

[*4.1. Reader information Form 37*](#_Toc533680097)

[*4.2. Book information Form 37*](#_Toc533680098)

[*4.3. Author information Form 38*](#_Toc533680099)

[*4.4. Kind of book information Form 38*](#_Toc533680100)

[*4.5. Publishing company information Form 38*](#_Toc533680101)

[5. Search Form 38](#_Toc533680102)

[*5.1. Search for book Form 38*](#_Toc533680103)

[*5.2. Search for reader Form 38*](#_Toc533680104)

[6. Report Form 39](#_Toc533680105)

[*6.1. Report to return books Form 39*](#_Toc533680106)

[*6.2. Report readers Form 39*](#_Toc533680107)

[**VII. IMPLEMENTATION 40**](#_Toc533680108)

[VIII. TESTING DOCUMENT 41](#_Toc533680109)

[1. Test Cases 41](#_Toc533680110)

[2. Test Scenario 43](#_Toc533680111)

[3. Requirement Traceability Matrix 46](#_Toc533680112)

# **I. OVERVIEW**

Nowadays, the development of information technology has great effect on our life. Many applications of information technology has been developed to meet people’s requirements, included applications of information technology in management.

Library management system is a project which aims in developing a computerized system to maintain all the daily work of library. This report describes the project of ours which is being developed to help the students as well as staff of library to maintain the library in the best way possible and also reduce the human efforts.

This library management system is mainly used by librarian and library manager. Normal librarian is able to manage the member maintenance module, book mantenance module and also the most important module in a library which is book transaction module. Beside that, library management system also allows user to manage the publisher as well as report module. On the other hand, other type of user which is admin level employee is abe to handle the emloyee module.

Compare to the existing library system, this system has some strength and weaknesses. In the future, we can enhance the system to make it more perfect.

**II. INTRODUCTION**

**1. Purpose**

The purpose of the project is to help librarian to manage the library using a computerized system where he/she can record various transactions like issue of books, return of books, addition of new books, addition of new students etc. Books and student maintenance modules are also included in this system which would keep track of the students using the library and also a detailed description about the books a library contains. With this computerized system there will be no loss of book record or member record which generally happens when a non computerized system is used. In addition, report module is also included in Library Management System. If user’s position is admin, the user is able to generate different kinds of reports like lists of students registered, list of books, issue and return reports. All these modules are able to help librarian to manage the library with more convenience and in a more efficient way as compared to library systems which are not computerized.

**2. Scope**

Library Management System is an application which refers to library systems which are generally small or medium in size.

## **3. Document description**

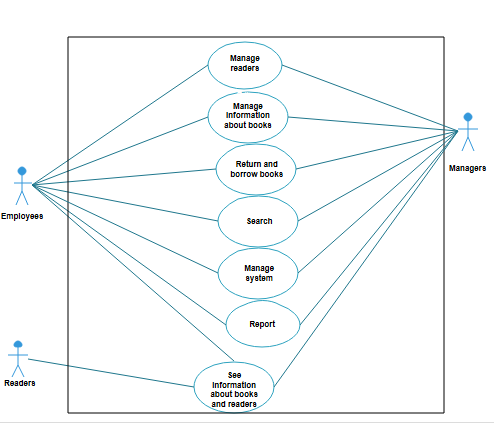
The structure of the document consists of 8 parts:

* Part 1 - Overview: General introduction of the document, help the reader visualize the content, purpose and general layout of the document.
* Part 2 - Introduce: Pose the most basic requirements that software must have.
* Part 3 - Use case model.: Building architectural models and describing system architecture.
* Part 4 - System architecture: Describe data dictionary and build data dictionary.
* Part 5 - Database design: Design subsystems of the system.
* Part 6 - User Interface Design: An overview of the interfaces, interface images, and accompanying operations.
* Part 7 - Implementation: An overview of the interfaces, interface images, and accompanying operations.
* Part 8 – Testing documents: An overview of the interfaces, interface images, and accompanying operations.

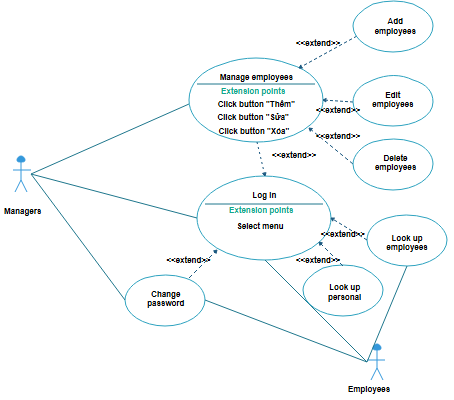
**III. USE CASE MODEL**

## **1. Use Case diagram**

### **1.1. General**



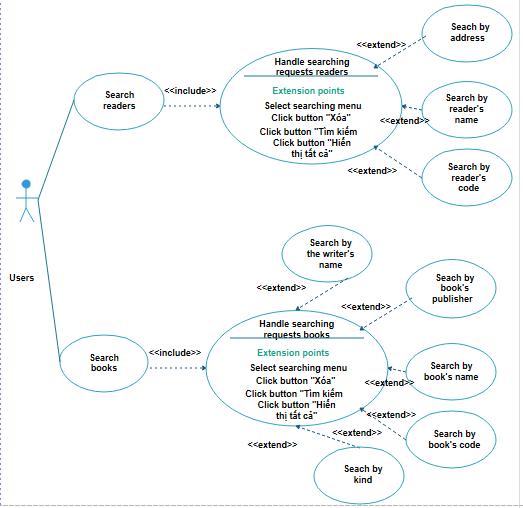
### **1.2. Manage system**



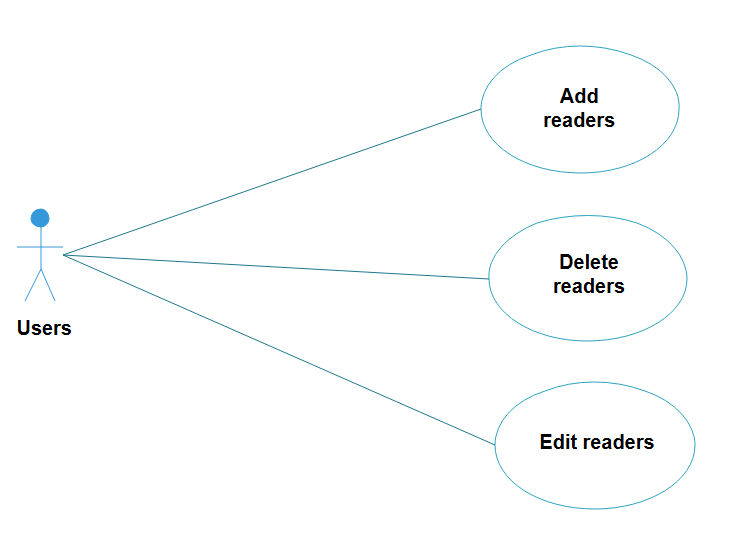
### **1.3. Manage information about books**

### **1.4. Return and borrow books**

**1.5. Search**



### **1.6. Manage readers**



## **2. Scenario**

*The scenario of borrowing the book*

Anna goes to the library, provides library card to Jasmine and selects a book.

After selecting a book, Anna brings it to the checkout desk.

Jasmine enters the book’s number into Anna's borrowed book list and enters Anna's information into the reader card.

*The scenario of returning the book*

Anna bring the book to the library.

Jasmine notifies the book return and issue a receipts of the library card and the book.

*The borrowing scenario is unsuccessful*

Bob provides to library cleck with library cards and information about the book which he wantes.

Jasmine search for information about books in stock. Because the book has been borrowed, Bob can not borrow this book.

Bob get back the library card.

## **3. Use case descriptions**

### **3.1. Use case:** “See information of books and readers”

- Actors: Readers, Managers and Employees

- Goal: See author, kind, publishing company,books and readers

- Overview: Readers and managers and employees can see author, kind, publishing company,books and readers when join the system

- Typical course of events

|  |  |
| --- | --- |
| **Actor action** | **System response** |
| 1. Readers and managers and employees select the search function by author, kind, publishing company,books and readers. | 2. The system display interface allows selecting buttons. |
| 3. Readers and managers and employees select to button. | 4. The inspection system displays complete information about the books and readers |
| 5. Readers and managers and employees see the necessary information. |  |

- Alternative courses: Steps 3, 4:Don’t select to button or the inspection system doen’t display.

### **3.2. Use case: Manage system**

#### **3.2.1. Use case:** Log in

- Actors: Managers and employees

- Goal: Log in into system

- Overview: Managers and employees join to mange system

- Typical course of events

|  |  |
| --- | --- |
| **Actor action** | **System response** |
| 1.The managers and employees chooses the function to log in. | 2. The system displays the interface for login |
| 3. The managers and employees enter information about the login: employeename, password. | 4. The system checks the validity of the information entered into the system. |
| 5. The managers and employees manage system. |  |

- Alternative courses: Steps 3,4: employeename or password information is incorrect. Can not manage system.

#### **3.2.2. Use case :** Change password

- Actors: Managers and employees

- Goal: Change password

- Overview: The managers and employees uses the function to Change password to change infomation of account in the library..

- Typical course of events

|  |  |
| --- | --- |
| **Actor action** | **System response** |
| 1.The managers and employees chooses the function to Change password . | 2. The system displays the interface for changing password |
| 3. The managers and employees enter information about the login: old password, new password and repeat new password. | 4. The system checks the validity of the information entered into the system. |
| 5. The managers and employees change successfully. |  |

- Alternative courses: Steps 3,4: old password information is incorrect or new password and repeat new password aren’t same . Can not change password.

### **3.3. Use case Manage employees**

The managers uses the function to Add, Delete, Edit employees to add, delete, edit employees to the library system..

#### **3.3.1. Use case :** Add employees

- Actors: Managers

- Goal: Add new employees to the library system

- Overview: The managers add new employees, update information to employees such as: login name, full name, title, gender, email, phone,

- Typical course of events

|  |  |
| --- | --- |
| **Actor action** | **System response** |
| 1. The Managers select the add employees function | 2. The system displays the interface for additional employees |
| 3. The Managers enter information related to the employees such as: login name, full name, job title, gender, email, phone. |  |
| 4. The Managers select save information | 5. System validation of information and add employees to the system |

- Alternative courses: Step 4: The managers remove the add employee. Can not add new employee to system. Step 5: The information for the login is invalid. The system displays an error message. Can not add new employee to system.

#### **3.3.2. Use case :** Edit employees

- Actors: Managers

- Goal: Edit infomation employees to the library system

- Overview: The managers changes information to employees such as: login name, full name, title, gender, email, phone,…

- Typical course of events

|  |  |
| --- | --- |
| **Actor action** | **System response** |
| 1. The managers select the edit employee function | 2. The system displays the interface for edit employee |
| 3. The managers select the record that needs to correct the employee information properly | 4. The system checks the validity of the information entered |
| 5. The managers choose save information | 6. The system validates and updates new employee information. |

- Alternative courses: Step 4: The managers remove the edit employee. Can not edit new employee to system. Step 5: The information is invalid. The system displays an error message. Can not edit new employee to system.

#### **3.3.3. Use case :** Delete employees

- Actors: Managers

- Goal: Delete infomation employee to the library system

- Overview: The managers delete information to employees

- Typical course of events

|  |  |
| --- | --- |
| **Actor action** | **System response** |
| 1. The managers select the delete employee function | 2. The system displays the interface for delete employee |
| 3. The managers select the record that needs to delete the employee information | 4. The system verification and deletion information employee. |
| 5. The managers confirms deletion of book information. |  |

- Alternative courses: Step 5: The managers remove the delete employee. Can not delete employee to system. Step 4: The system encountered an error processing the deletion of the employee information. The system displays an error message. Employee information is not deleted.

### **3.4. Use case Manage books**

#### **3.4.1. Use case: Add books**

- Actors: Managers and Employees

- Goal: Add new books to the library

- Overview: The Managers and Employees uses the function to add books to add new books to the library..

- Typical course of events

|  |  |
| --- | --- |
| **Actor action** | **System response** |
| 1.The Managers and Employees chooses the function to add books. | 2. The system displays the interface for additional books |
| 3. The Managers and Employees enter information about the book: book code, book title, author name, publisher ... | 4. The system checks the validity of the information entered into the system. |
| 5. The Managers and Employees keeps adding books and finishes. |  |

- Alternative courses: Steps 4: The Managers and Employees information entered is invalid, the system gives the error message. Books can not be added to the library system.

#### **3.4.2. Use case :** Edit books

- Actors: Managers and Employees

- Goal: Change book information

- Overview: The Managers and Employees uses the function to Edit books to change infomation of books in the library..

- Typical course of events

|  |  |
| --- | --- |
| **Actor action** | **System response** |
| 1.The Managers and Employees chooses the function to edit books. | 2. The system displays the interface for edit books |
| 3. The Managers and Employees enter to change information about the book:book title, author name, publisher ... | 4. The system checks the validity of the information entered into the system. |
| 5. The Managers and Employees keeps adding books and finishes. |  |

- Alternative courses: Steps 4: The system encountered an error in adding books. The system displays error messages

#### **3.4.3. Use case :** Delete books

- Actors: Managers and Employees

- Goal: Delete books to the library

- Overview: The Managers and Employees uses the function delete books to delete books to the library.

- Typical course of events

|  |  |
| --- | --- |
| **Actor action** | **System response** |
| 1.The Managers and Employees chooses the function to delete books. | 2. The system displays the interface for delete books |
| 3. The Managers and Employees enter the code to delete. | 4. The system verification and deletion information book. |
| 5. The Managers and Employees confirms deletion of book information. |  |

- Alternative courses: Steps 4: The system encountered an error processing the deletion of the book information. The system displays an error message. Book information is not deleted. Steps 5: The Managers and Employees remove book information deletion. Book information is not deleted.

### **3.5. Use case Manage readers**

#### **3.5.1. Use case :** Add readers

- Actors: Managers and Employees

- Goal: Add new readers to the library

- Overview: The Managers and Employees uses the function to add readers to add new readers to the library..

- Typical course of events

|  |  |
| --- | --- |
| **Actor action** | **System response** |
| 1.The Managers and Employees chooses the function to add readers. | 2. The system displays the interface for additional readers |
| 3. The Managers and Employees enter information about the reader: reader code, reader title, author name, publisher ... | 4. The system checks the validity of the information entered into the system. |
| 5. The Managers and Employees keeps adding readers and finishes. |  |

- Alternative courses: Steps 4: The Managers and Employees information entered is invalid, the system gives the error message. Readers can not be added to the library system.

#### **3.5.2. Use case :** Edit readers

- Actors: Managers and Employees

- Goal: Change reader information

- Overview: The Managers and Employees uses the function to Edit readers to change infomation of readers in the library..

- Typical course of events

|  |  |
| --- | --- |
| **Actor action** | **System response** |
| 1.The Managers and Employees chooses the function to edit readers. | 2. The system displays the interface for edit readers |
| 3. The Managers and Employees enter to change information about the reader:reader title, author name, publisher ... | 4. The system checks the validity of the information entered into the system. |
| 5. The Managers and Employees keeps adding readers and finishes. |  |

- Alternative courses: Steps 4: The system encountered an error in adding readers. The system displays error messages

#### **3.5.3. Use case :** Delete readers

- Actors: Managers and Employees

- Goal: Delete readers to the library

- Overview: The Managers and Employees uses the function delete readers to delete readers to the library.

- Typical course of events

|  |  |
| --- | --- |
| **Actor action** | **System response** |
| 1.The Managers and Employees chooses the function to delete readers. | 2. The system displays the interface for delete readers |
| 3. The Managers and Employees enter the code to delete. | 4. The system verification and deletion information reader. |
| 5. The Managers and Employees confirms deletion of reader information. |  |

- Alternative courses: Steps 4: The system encountered an error processing the deletion of the reader information. The system displays an error message. Reader information is not deleted. Steps 5: The Managers and Employees remove reader information deletion. Reader information is not deleted.

**3.6. Use case Manage book’s kind**

#### **3.6.1. Use case :** Add book’s kind

- Actors: Managers and Employees

- Goal: Add new book’s kinds to the library

- Overview: The Managers and Employees uses the function to add book’s kinds to add new book’s kinds to the library..

- Typical course of events

|  |  |
| --- | --- |
| **Actor action** | **System response** |
| 1.The Managers and Employees chooses the function to add book’s kinds. | 2. The system displays the interface for additional book’s kinds |
| 3. The Managers and Employees enter information about the book’s kind: book’s kind code, book’s kind title, author name, publisher ... | 4. The system checks the validity of the information entered into the system. |
| 5. The Managers and Employees keeps adding book’s kinds and finishes. |  |

- Alternative courses: Steps 4: The Managers and Employees information entered is invalid, the system gives the error message. Book’s kinds can not be added to the library system.

#### **3.6.2. Use case :** Edit book’s kinds

- Actors: Managers and Employees

- Goal: Change book’s kind information

- Overview: The Managers and Employees uses the function to Edit book’s kinds to change infomation of book’s kinds in the library..

- Typical course of events

|  |  |
| --- | --- |
| **Actor action** | **System response** |
| 1.The Managers and Employees chooses the function to edit book’s kinds. | 2. The system displays the interface for edit book’s kinds |
| 3. The Managers and Employees enter to change information about the book’s kind:book’s kind title, author name, publisher ... | 4. The system checks the validity of the information entered into the system. |
| 5. The Managers and Employees keeps adding book’s kinds and finishes. |  |

- Alternative courses: Steps 4: The system encountered an error in adding book’s kinds. The system displays error messages

#### **3.6.3. Use case :** Delete book’s kinds

- Actors: Managers and Employees

- Goal: Delete book’s kinds to the library

- Overview: The Managers and Employees uses the function delete book’s kinds to delete book’s kinds to the library.

- Typical course of events

|  |  |
| --- | --- |
| **Actor action** | **System response** |
| 1.The Managers and Employees chooses the function to delete book’s kinds. | 2. The system displays the interface for delete book’s kinds |
| 3. The Managers and Employees enter the code to delete. | 4. The system verification and deletion information book’s kind. |
| 5. The Managers and Employees confirms deletion of book’s kind information. |  |

- Alternative courses: Steps 4: The system encountered an error processing the deletion of the book’s kind information. The system displays an error message. Book’s kind information is not deleted. Steps 5: The Managers and Employees remove book’s kind information deletion. Book’s kind information is not deleted.

**3.7. Use case Manage publishing company**

#### **3.7.1. Use case :** Add publishing company

- Actors: Managers and Employees

- Goal: Add new publishing companys to the library

- Overview: The Managers and Employees uses the function to add publishing companys to add new publishing companys to the library..

- Typical course of events

|  |  |
| --- | --- |
| **Actor action** | **System response** |
| 1.The Managers and Employees chooses the function to add publishing companys. | 2. The system displays the interface for additional publishing companys |
| 3. The Managers and Employees enter information about the publishing company: publishing company code, publishing company title, author name, publisher ... | 4. The system checks the validity of the information entered into the system. |
| 5. The Managers and Employees keeps adding publishing companys and finishes. |  |

- Alternative courses: Steps 4: The Managers and Employees information entered is invalid, the system gives the error message. Publishing companys can not be added to the library system.

#### **3.7.2. Use case :** Edit publishing companys

- Actors: Managers and Employees

- Goal: Change publishing company information

- Overview: The Managers and Employees uses the function to Edit publishing companys to change infomation of publishing companys in the library.

- Typical course of events

|  |  |
| --- | --- |
| **Actor action** | **System response** |
| 1.The Managers and Employees chooses the function to edit publishing companys. | 2. The system displays the interface for edit publishing companys |
| 3. The Managers and Employees enter to change information about the publishing company:publishing company title, author name, publisher ... | 4. The system checks the validity of the information entered into the system. |
| 5. The Managers and Employees keeps adding publishing companys and finishes. |  |

- Alternative courses: Steps 4: The system encountered an error in adding publishing companys. The system displays error messages

#### **3.7.3. Use case :** Delete publishing companys

- Actors: Managers and Employees

- Goal: Delete publishing companys to the library

- Overview: The Managers and Employees uses the function delete publishing companys to delete publishing companys to the library.

- Typical course of events

|  |  |
| --- | --- |
| **Actor action** | **System response** |
| 1.The Managers and Employees chooses the function to delete publishing companys. | 2. The system displays the interface for delete publishing companys |
| 3. The Managers and Employees enter the code to delete. | 4. The system verification and deletion information publishing company. |
| 5. The Managers and Employees confirms deletion of publishing company information. |  |

- Alternative courses: Steps 4: The system encountered an error processing the deletion of the publishing company information. The system displays an error message. Publishing company information is not deleted. Steps 5: The Managers and Employees remove publishing company information deletion. Publishing company information is not deleted.

**3.8. Use case Borrow and return books**

#### **3.8.1. Use case:** Borrow books

- Actors: Managers and Employees

- Goal: Borrow books

- Overview: When the reader arrives, the Managers and Employees will select the loan book manager. The Managers and Employees will enter the reader information and book information, wait for the system to check the information and then make a loan. The Managers and Employees archives the reader loan information, the reader receives the borrowed books. And , the Managers and Employees can delete or edit borrowing informations

- Typical course of events

|  |  |
| --- | --- |
| **Actor action** | **System response** |
| 1. Readers choose books | 2. System display book information |
| 3. The Managers and Employees chooses book lending management | 4.Interface display system for borrowing books. |
| 5.Incoming mailer information and reader information | 6.System to check the validity of input information |
| 7. The Managers and Employees chooses to make a loan slip | 8. The system makes a voucher and stores information about readers and books in the system |
| 9. The Managers and Employees collects the loan slip |  |
| 10. If Managers and Employees select the delete,edit borrowing informations function | 11. The system displays the interface for delete, edit borrowing informations |
| 12. The Managers and Employees enter to change information about the borrowing informations or enter the code to delete. | 13. The system checks the validity of the information entered into the system. Or The system verification and deletion information book |
| 14. The Managers and Employees keeps adding books and finishes.or The Managers and Employees confirms deletion of book information. |  |

- Alternative courses: Steps 6-9: The Managers and Employees information entered is invalid, the system gives the error message. Readers can not borrow books.

#### **3.8.2. Use case :** Return books

- Actors: Managers and Employees

- Goal: Give books back

- Overview: When the reader arrives, the Managers and Employees will select the return book management manager. The Managers and Employees will scan the book code, wait for the system to check the information and display the information of the loan slip. The Managers and Employees update reader loan information. And the Managers and Employees can delete or edit returning informations.

- Typical course of events

|  |  |
| --- | --- |
| **Actor action** | **System response** |
| 1. Readers come to return book. |  |
| 2. The Managers and Employees selects the payroll management function. | 3. System display interface return books. |
| 4. The Managers and Employees performs a scan of the book code. | 5. System to receive the book code, display the information of the loan. |
| 6. The Managers and Employees updates the information of the loan. | 7. The system performs updating and saving of loan information into the system. |
| 8. End the book returns |  |
| 9. If Managers and Employees select the delete,edit returning informations function | 10. The system displays the interface for delete, edit returning informations |
| 11. The Managers and Employees enter to change information about the returning informations or enter the code to delete. | 12. The system checks the validity of the information entered into the system. Or The system verification and deletion information returning informations |
| 13. The Managers and Employees keeps adding returning informations and finishes.or The Managers and Employees confirms deletion of returning informations information. |  |

- Alternative courses: Steps 5-8: Incorrect book code, the Managers and Employees can not return the book.

### **3.9. Use case Search**

#### **3.9.1 Use case :** Search books

- Actors: Managers and Employeess

- Goal: Search book information

- Overview: Managers and Employeess search for book information to perform other operations.

- Typical course of events

|  |  |
| --- | --- |
| **Actor action** | **System response** |
| 1. Readers and Managers and Employeess select the search function by writer, publishing company, name, kind or code of book | 2. System display interface allows to enter writer, publishing company, name, kind or code of book to search. |
| 3. Managers and Employeess enter the writer, publishing company, name, kind or code of book to search. | 4. The inspection system exists and displays complete information about the book |
| 5. Managers and Employeess see the necessary information. |  |

- Alternative courses: Steps 3,4: writer, publishing company, name, kind or code of book is incorrect. Can not find book information.

#### **3.9.2. Use case :** Search readers

- Actors: Managers and Employeess

- Goal: Search readers information

- Overview: Managers and Employeess search for readers information to perform other operations.

- Typical course of events

|  |  |
| --- | --- |
| **Actor action** | **System response** |
| 1. Readers and Managers and Employeess select the search function by address, name or code of readers | 2. System display interface allows to enter address, name or code of readers to search. |
| 3. Managers and Employeess enter the address, name or code of readers to search. | 4. The inspection system exists and displays complete information about the readers |
| 5. Managers and Employeess see the necessary information. |  |

**-** Alternative courses: Steps 3,4: address, name or code of readers is incorrect. Can not find readers information.

### **3.10. Use case Report**

- Actors: Managers and employees

- Goal: Report

- Overview: The managers and employees report returning books and readers

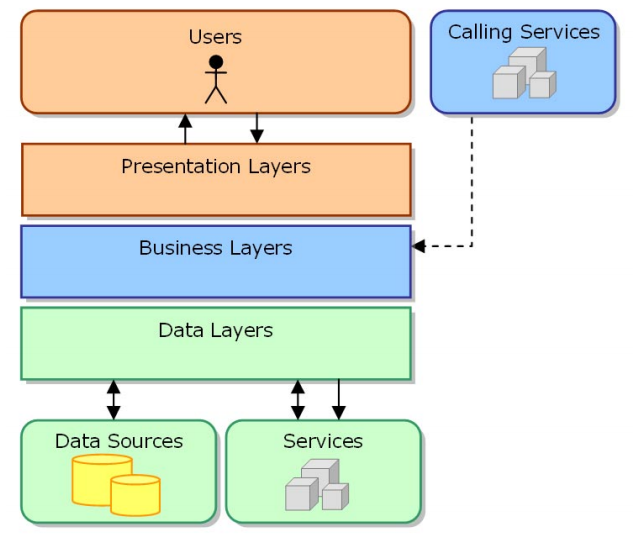
- Typical course of events

|  |  |
| --- | --- |
| **Actor action** | **System response** |
| 1. The managers and employees select the report function | 2. The system displays the interface for report readers or returning books |
| 3. The managers and employees select the button | 4. The system verification and report information readers or returning books. |
| 5. The managers and employees see information about report readers or returning books. |  |

# **IV. SYSTEM ARCHITECTURE**

## **1. Program architecture**

- The program is built according to MVC model (3-layer architecture).

- General model of 3-layer architecture

- 3-layer model consists of 3 main parts:

+ Presentation Layer (GUI): This class has the main task of communicating with users. It includes interface elements (win form, web form, ...) and performs tasks such as input, data display, data validation before calling Business Logic Layer (BLL).

+ Business Logic Layer (BLL): This layer is divided into 2 tasks:

This is where the data manipulation requirements of the GUI layer are met, processing the data source from Presentation Layer first before transferring it to Data Access Layer and saving it to the database administration system.

This is also a place to check constraints, integrity and validity of data, perform calculations and handle business requirements, before returning results to Presentation Layer.

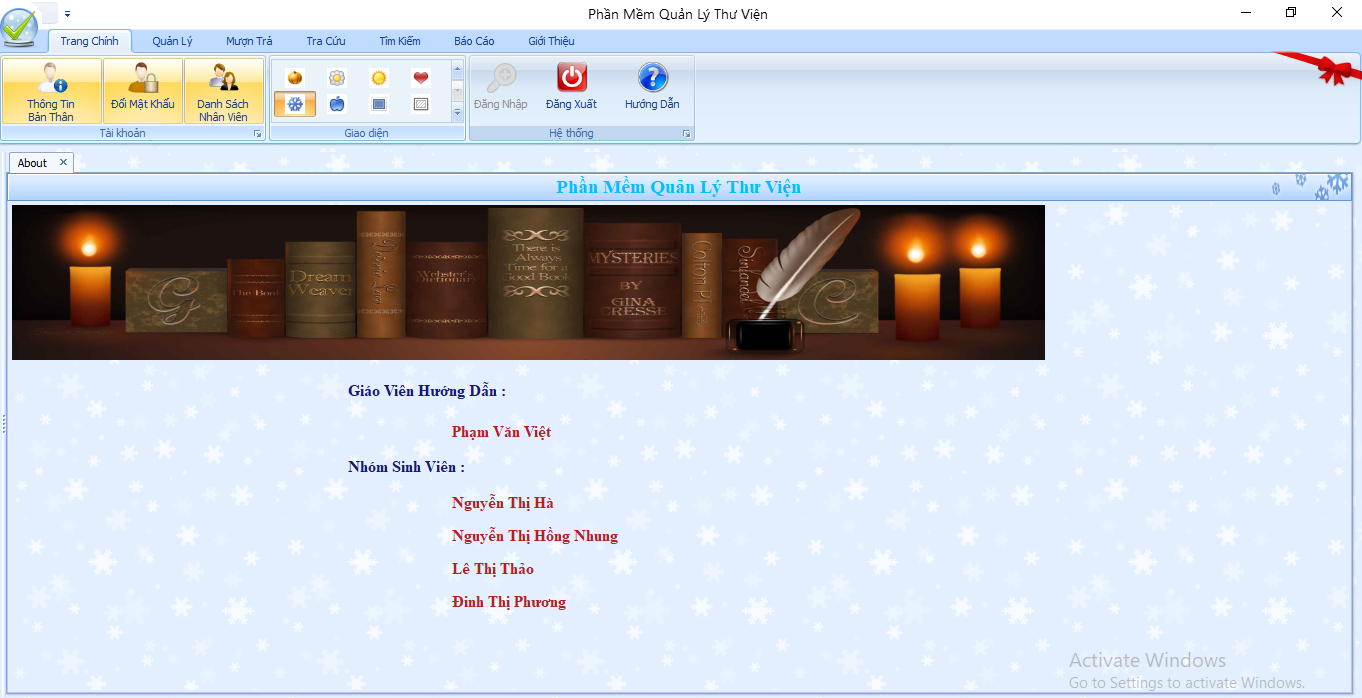
+ Data Access Layer (DAL): This class has the function of communicating with database management system such as performing tasks related to data storage and querying (search, add, delete, edit, ...)..

## **2. Detailed description**

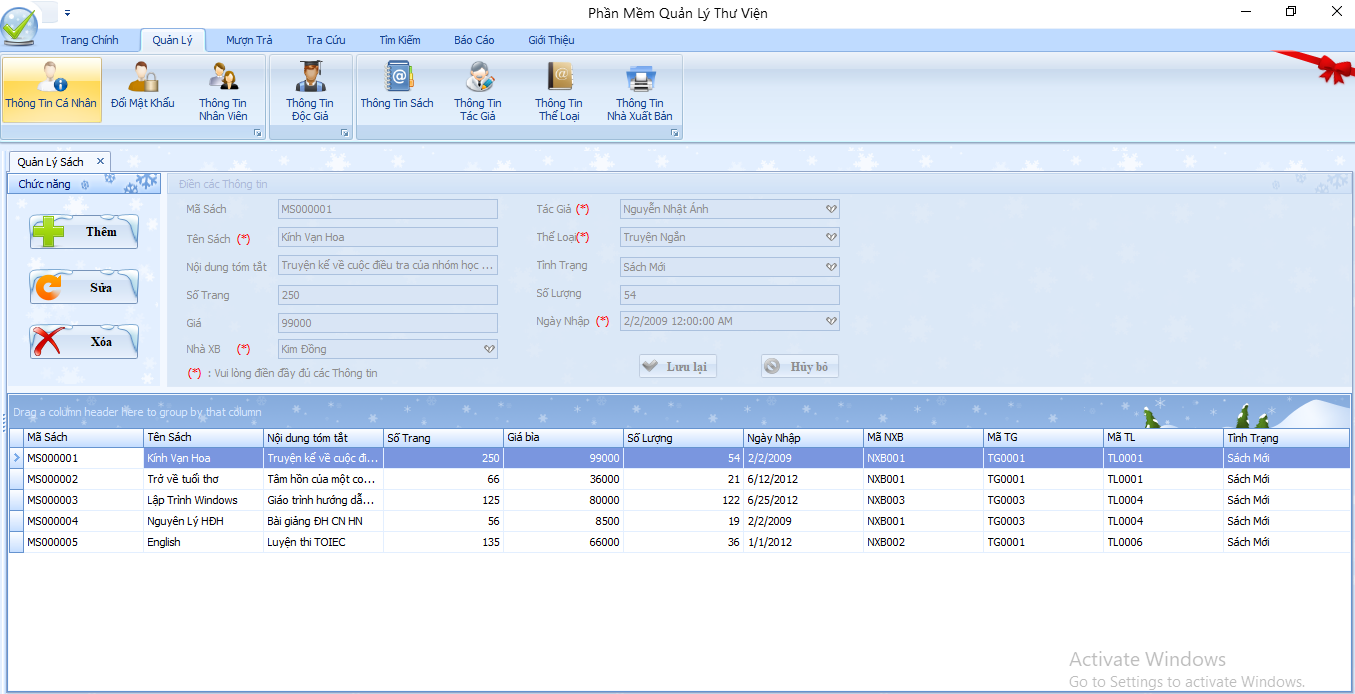
1. *Presentation Layers*

Some main interfaces

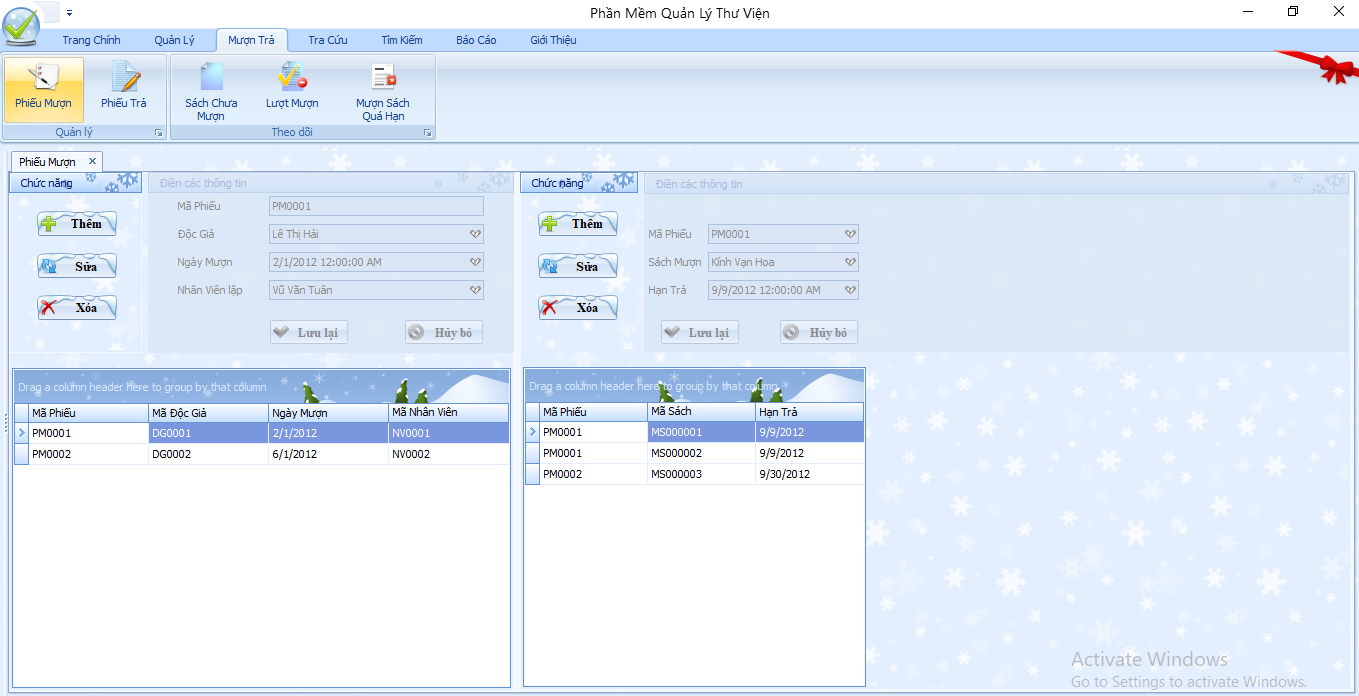
* Homepage Form: (TrangChu.cs)



* Management Form of Book Information (cnSach.cs)



* Borrow Books Form (MuonSach.cs)



Other forms can be viewed at the program.

*b. Business Layers*

This class includes functions related to GUI layer operations (data interfaces) with data taken from that database converted by DTO.

A class of BLL usually includes functions: The function takes all the data list to display, the function checks before saving, Ham added, Ham edit, Ham xua. There are also other functions corresponding to the functions: search, display, output information ... for each function related to that data

**For example**: Class DocGia\_BLL.cs: (corresponding to operations related to DocGia table data)

- BLL includes 9 classes corresponding to operations on 9 data tables: DocGia\_BLL.cs, TacGia\_BLL.cs, NhanVien\_BLL.cs, Sach\_BLL.cs, TheLoai\_BLL.cs, NXB\_BLL.cs, PhieuMuon\_BLL.cs, CTPhieuMuon\_BLL.cs, PhieuTra\_BLL.cs

*c. Data Layers*

DAL class includes functions:

- Function to create connection string to SQL Server database: This function creates connection string to data table ‘QLTV’ in SQL Server, using Try {} - Catch {} function to catch data connection error

- The function takes data from the database according to the sql command passed.

- The function manipulates adding, editing, deleting data according to the sql statement entered.

- The function checks for existence.

- The function takes the final code.

## **3. Functional decomposition diagram**

**LIBRARY MANAGEMENT**

MANAGE LIST

MANAGE BORROW AND RETURN

SEARCH

REPORT

MANAGE BOOKS

MANAGE EMLOYEES

MANAGE

AUTHOR

MANAGE PUBLISHING

MANAGE TYPES

MANAGE

READERS

BORROW BOOKS

RETURN BOOKS

READERS

IMPORT BOOKS

BORROWED BOOKS

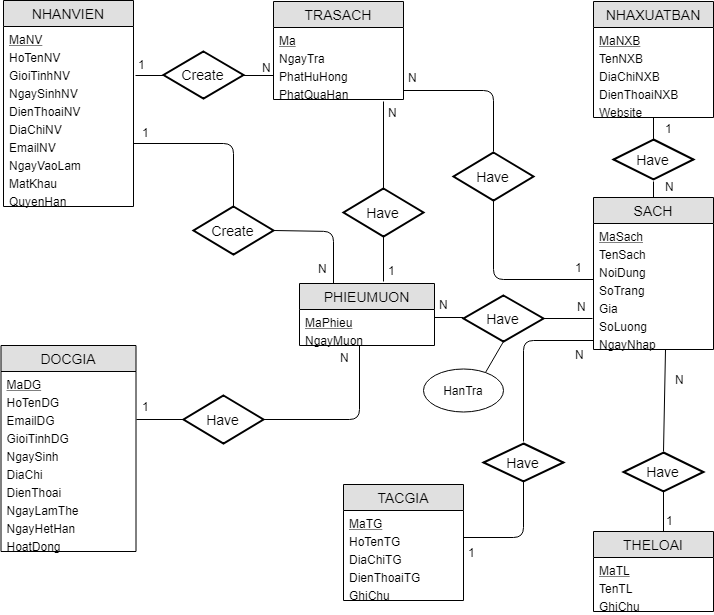
READERS

IMPORTED BOOKS

BORROWED BOOKS

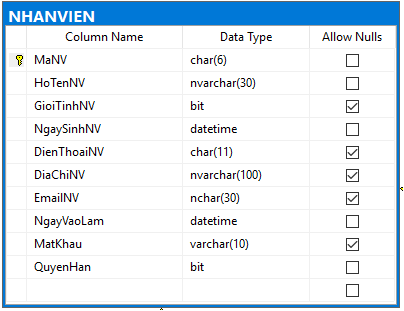
**V. DATABASE DESIGN**

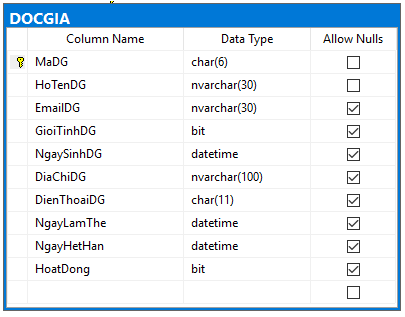
## **1. Entity – Relationship Diagram (8 entities):**

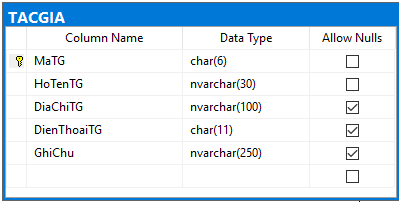


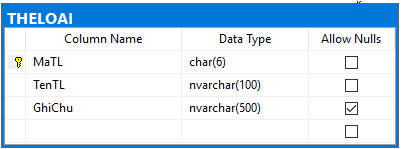
## **2. Logical Database Design**

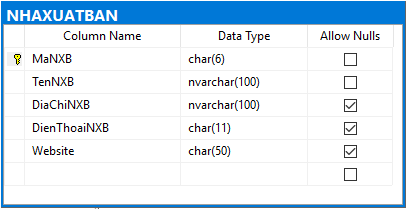
## **3. Physical Database Design**

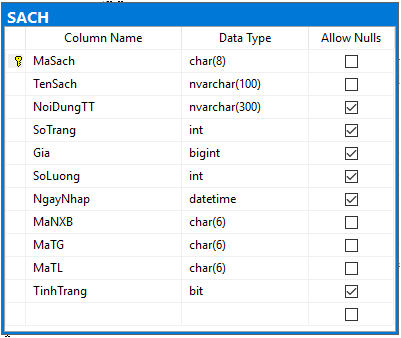


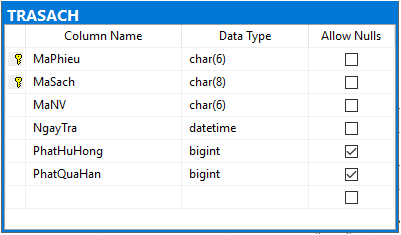


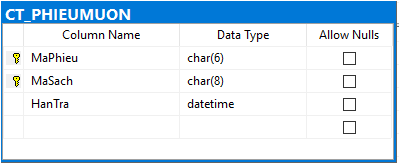


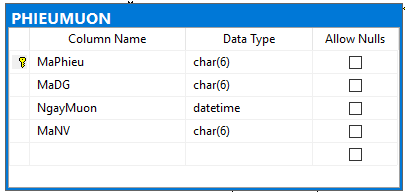












# **VI. USER INTERFACE DESIGN**

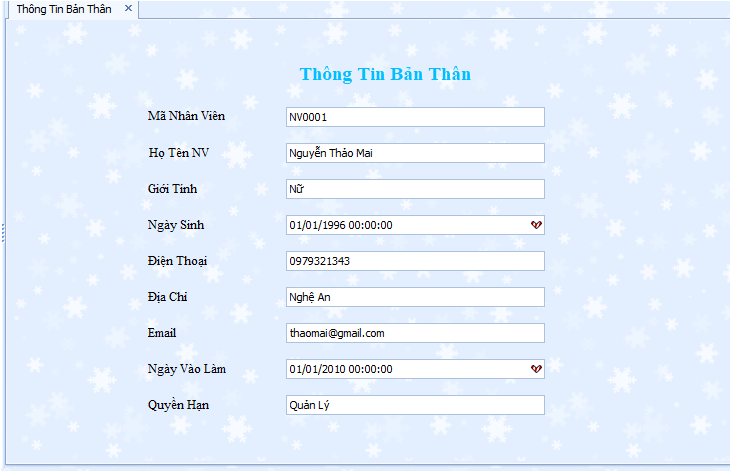
**1. Main Interface Form**



**1.1. Login Form**



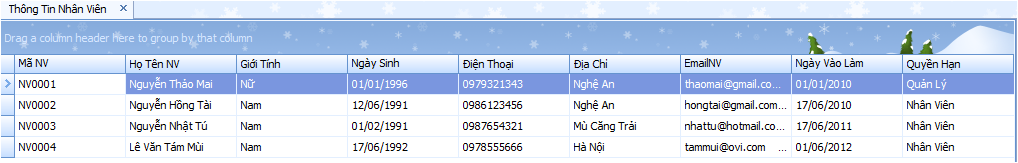
**1.2. Personal information Form**



**1.3. Change the password Form**



**1.4. Employee information Form**



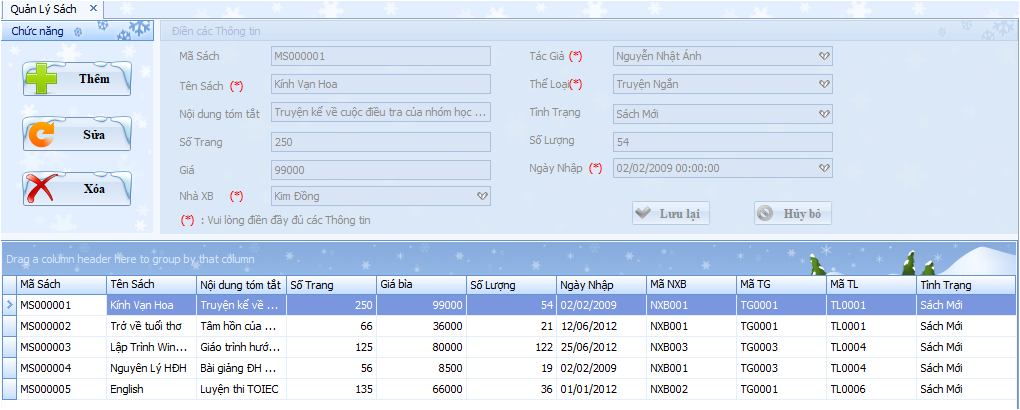
## **2. Management Form**

### **2.1. Employee management Form**

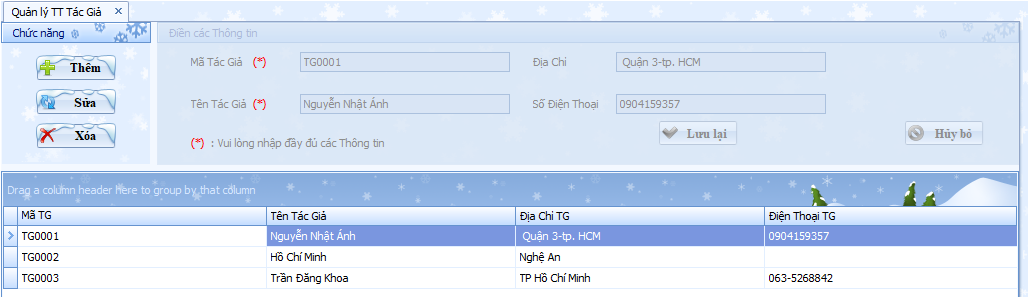
**2.2. Reader management Form**



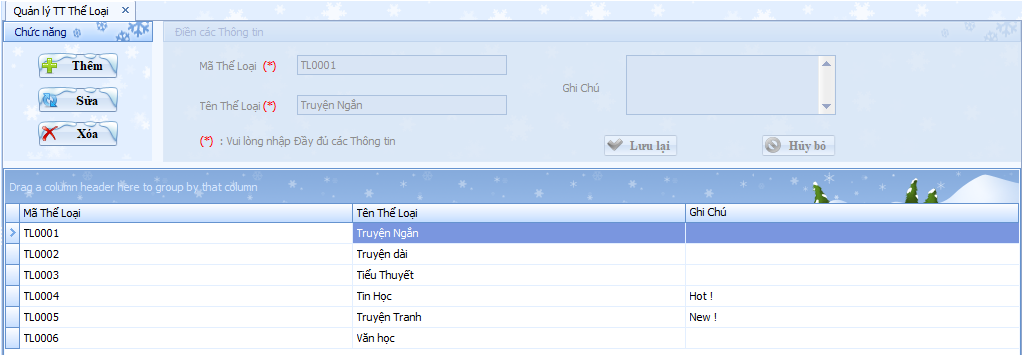
**2.3. Book management Form**



**2.4. Author management Form**



**2.5. Kind of book management Form**

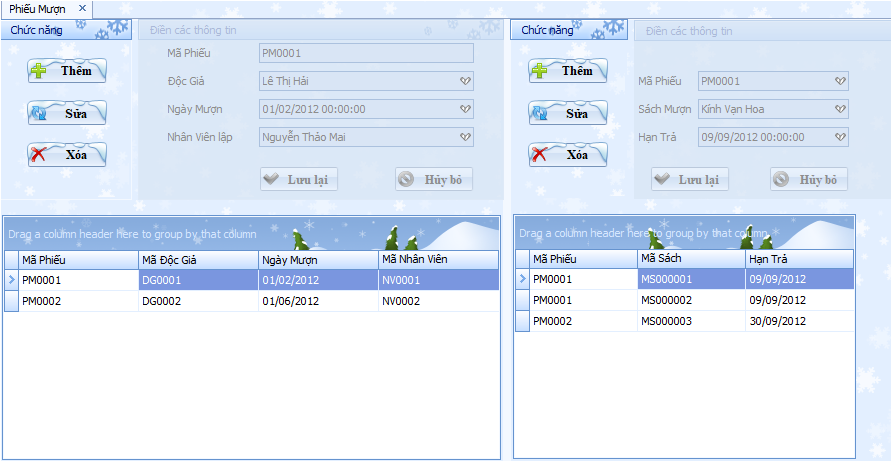


**2.6. Publishing company management Form**



**3. Borrow- Return Form**

**3.1. Borrow information Form**

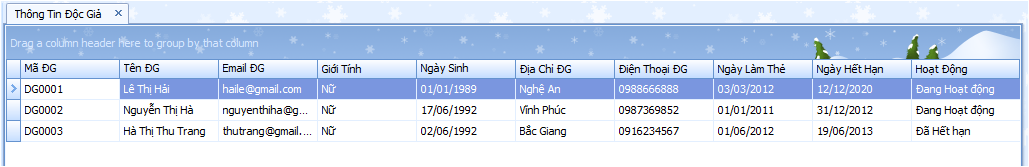


**3.2. Return information Form**

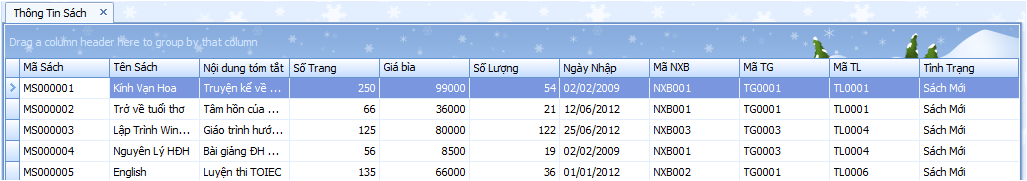


**4. Look up Form**

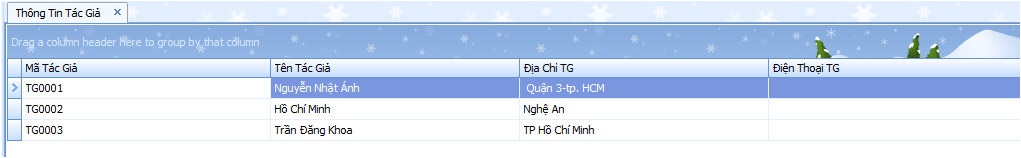
**4.1. Reader information Form**



**4.2. Book information Form**



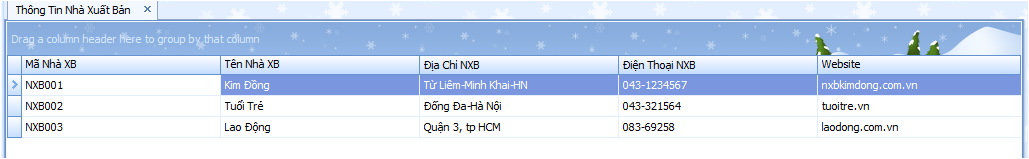
**4.3. Author information Form**



**4.4. Kind of book information Form**

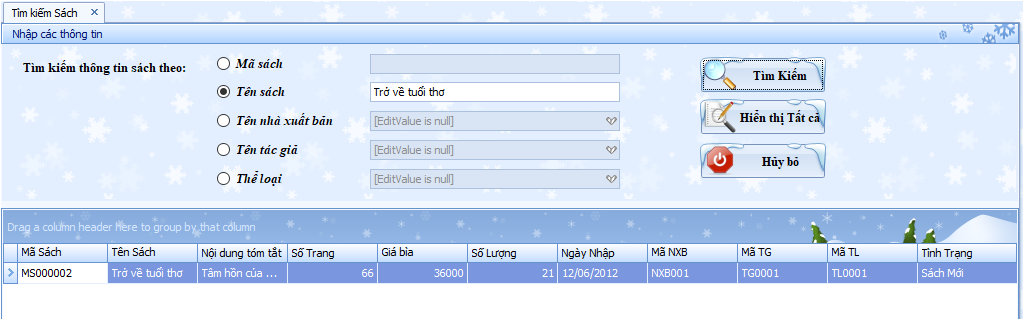


**4.5. Publishing company information Form**

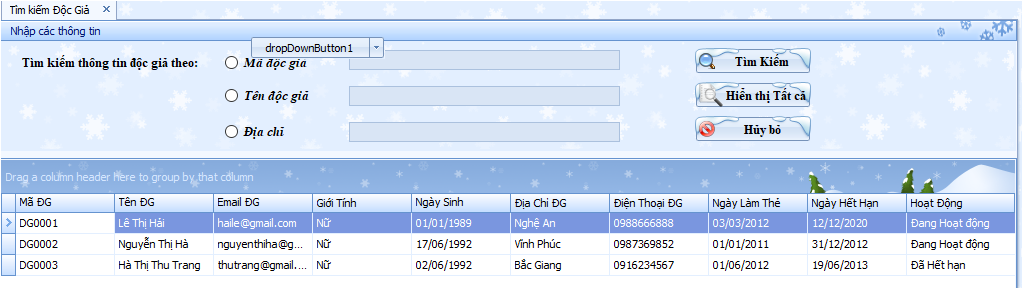


**5. Search Form**

**5.1. Search for book Form**



**5.2. Search for reader Form**



**6. Report Form**



**6.1. Report to return books Form**



**6.2. Report readers Form**



**VII. IMPLEMENTATION**

This part discusses about the programming coding of the library management system. Different logic thinking and coding is required at development of the system. This project is programmed in C# with three layers model – which is popular in software development.

Details can be viewed in the source code.

# **VIII. TESTING DOCUMENT**

## **1. Test Cases**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Seq.No | UC ID | Condition to be tested | Test data | Expected result | Successful |
|  | TS\_login | Log in | True username and true password | Log in success | Successful |
|  | TS\_login | Log in | Wrong username or wrong password | Log in failed | Successful |
|  | TS\_addbook | Add a book | Information of the book is valid | Add success | Successful |
|  | TS\_addbook | Add a book | Information of the book is invalid | Add failed | Successful |
|  | TS\_editbook | Edit a book | Information of the book is valid | Edit success | Successful |
|  | TS\_editbook | Edit a book | Information of the book is invalid | Edit failed | Successful |
|  | TS\_deletebook | Delete a book | Information of the book is valid and the librarian confirm to delete the book | Delete success | Successful |
|  | TS\_deletebook | Delete a book | Information of the book is invalid or the librarian does not confirm to delete the book | Delete failed | Successful |
|  | TS\_searchbook | Search a book | Name or code of the book is incorrect | Search failed | Successful |
|  | TS\_searchbook | Search a book | Name or code of the book is correct | Search success | Successful |
|  | TS\_adduser | Add a user | Information of the user is valid | Add success | Successful |
|  | TS\_adduser | Add a user | Information of the user is invalid | Add failed | Successful |
|  | TS\_edituser | Edit a user | Information of the user is valid | Edit success | Successful |
|  | TS\_edituser | Edit a user | Information of the user is invalid | Edit failed | Successful |
|  | TS\_deleteuser | Delete a user | Information of the user is valid and the librarian confirm to delete | Delete success | Successful |
|  | TS\_deleteuser | Delete a user | Information of the user is invalid or the librarian does not confirm to delete | Delete failed | Successful |
|  | TS\_borrowbook | Borrow a book | Invalid student’s information or invalid book’s information | Update the borrowed book failed | Successful |
|  | TS\_borrowbook | Borrow a book | Valid student’s information or invalid book’s information | Update the borrowed book success | Successful |
|  | TS\_returnbook | Return a book | Correct book’s code | Update the returned book success | Successful |
|  | TS\_returnbook | Return a book | Incorrect book’s code | Update the returned book failed | Successful |

## **2. Test Scenario**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Seq.No | Requirement Name | Test scenario | | Test Case | |
| TS Name | TS ID | TC Name | TC ID |
| 1 | Login | OK | TS\_login\_01 | a. Verify OK with valid data | TS\_login\_01\_01 |
| b. Verify OK with invalid data | TS\_login\_01\_02 |
| Cancel | TS\_login\_02 | a. Verify Cancel with valid data | TS\_login\_02\_01 |
| b. Verify Cancel with invalid data | TS\_login\_02\_02 |
| 2 | Add a book | OK | TS\_addbook\_01 | a. Verify OK with valid data | TS\_addbook\_01\_01 |
| b. Verify OK with invalid data | TS\_addbook\_01\_02 |
| Cancel | TS\_addbook\_02 | a. Verify Cancel with valid data | TS\_addbook\_02\_01 |
| b. Verify Cancel with invalid data | TS\_addbook\_02\_02 |
| 3 | Edit book | OK | TS\_editbook\_01 | a. Verify OK with valid data | TS\_editbook\_01\_01 |
| b. Verify OK with invalid data | TS\_editbook\_01\_02 |
| Cancel | TS\_editbook\_02 | a. Verify Cancel with valid data | TS\_editbook\_02\_01 |
| b. Verify Cancel with invalid data | TS\_editbook\_02\_02 |
| 4 | Delete book | OK | TS\_deletebook\_01 | a. Verify OK with valid data | TS\_deletebook\_01\_01 |
| b. Verify OK with invalid data | TS\_deletebook\_01\_02 |
| Cancel | TS\_deletebook\_02 | a. Verify Cancel with valid data | TS\_deletebook\_02\_01 |
| b. Verify Cancel with invalid data | TS\_deletebook\_02\_02 |
| 5 | Add user | OK | TS\_adduser\_01 | a. Verify OK with valid data | TS\_adduser\_01\_01 |
| b. Verify OK with invalid data | TS\_adduser\_01\_02 |
| Cancel | TS\_adduser\_02 | a. Verify Cancel with valid data | TS\_adduser\_02\_01 |
| b. Verify Cancel with invalid data | TS\_adduser\_02\_02 |
| 6 | Edit user | OK | TS\_edituser\_01 | a. Verify OK with valid data | TS\_edituser\_01\_01 |
| b. Verify OK with invalid data | TS\_edituser\_01\_02 |
| Cancel | TS\_edituser\_02 | a. Verify Cancel with valid data | TS\_edituser\_02\_01 |
| b. Verify Cancel with invalid data | TS\_edituser\_02\_02 |
| 7 | Delete user | OK | TS\_deleteuser\_01 | a. Verify OK with valid data | TS\_deleteuser\_01\_01 |
| b. Verify OK with invalid data | TS\_deleteuser\_01\_02 |
| Cancel | TS\_deleteuser\_02 | a. Verify Cancel with valid data | TS\_deleteuser\_02\_01 |
| b. Verify Cancel with invalid data | TS\_deleteuser\_02\_02 |
| 8 | Search book | OK | TS\_searchbook\_01 | a. Verify OK with valid data | TS\_searchbook\_01\_01 |
| b. Verify OK with invalid data | TS\_searchbook\_01\_02 |
| Cancel | TS\_searchbook\_02 | a. Verify Cancel with valid data | TS\_searchbook\_02\_01 |
| b. Verify Cancel with invalid data | TS\_searchbook\_02\_02 |
| 9 | Borrow book | OK | TS\_borrowbook\_01 | a. Verify OK with valid data | TS\_borrowbook\_01\_01 |
| b. Verify OK with invalid data | TS\_borrowbook\_01\_02 |
| Cancel | TS\_borrowbook\_02 | a. Verify Cancel with valid data | TS\_borrowbook\_02\_01 |
| b. Verify Cancel with invalid data | TS\_borrowbook\_02\_02 |
| 10 | Return book | OK | TS\_returnbook\_01 | a. Verify OK with valid data | TS\_returnbook\_01\_01 |
| b. Verify OK with invalid data | TS\_returnbook\_01\_02 |
| Cancel | TS\_returnbook\_02 | a. Verify Cancel with valid data | TS\_returnbook\_02\_01 |
| b. Verify Cancel with invalid data | TS\_returnbook\_02\_02 |
| b. Verify OK with invalid data | TS\_setupcard\_01\_02 |

## **3. Requirement Traceability Matrix**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Seq.No | UC id | TS id | TC id | Status | Defect |
|  | TS\_login | TS\_login\_01 | TS\_login\_01\_01 | Passed | No |
| TS\_login\_01\_02 | Passed | No |
| TS\_login\_02 | TS\_login\_02\_01 | Passed | No |
| TS\_login\_02\_02 | Passed | No |
|  | TS\_addbook | TS\_addbook\_01 | TS\_addbook\_01\_01 | Passed | No |
| TS\_addbook\_01\_02 | Passed | No |
| TS\_addbook\_02 | TS\_addbook\_02\_01 | Passed | No |
| TS\_addbook\_02\_02 | Passed | No |
|  | TS\_editbook | TS\_editbook\_01 | TS\_editbook\_01\_01 | Passed | No |
| TS\_editbook\_01\_02 | Passed | No |
| TS\_editbook\_02 | TS\_editbook\_02\_01 | Passed | No |
| TS\_editbook\_02\_02 | Passed | No |
|  | TS\_deletebook | TS\_deletebook\_01 | TS\_deletebook\_01\_01 | Passed | No |
| TS\_deletebook\_01\_02 | Passed | No |
| TS\_deletebook\_02 | TS\_deletebook\_02\_01 | Passed | No |
| TS\_deletebook\_02\_02 | Passed | No |
|  | TS\_searchbook | TS\_searchbook\_01 | TS\_searchbook\_01\_01 | Passed | No |
| TS\_searchbook\_01\_02 | Passed | No |
| TS\_searchbook\_02 | TS\_searchbook\_02\_01 | Passed | No |
| TS\_searchbook\_02\_02 | Passed | No |
|  | TS\_adduser | TS\_adduser\_01 | TS\_adduser\_01\_01 | Passed | No |
| TS\_adduser\_01\_02 | Passed | No |
| TS\_adduser\_02 | TS\_adduser\_02\_01 | Passed | No |
| TS\_adduser\_02\_02 | Passed | No |
|  | TS\_edituser | TS\_edituser\_01 | TS\_edituser\_01\_01 | Passed | No |
| TS\_edituser\_01\_02 | Passed | No |
| TS\_edituser\_02 | TS\_edituser\_02\_01 | Passed | No |
| TS\_edituser\_02\_02 | Passed | No |
|  | TS\_deleteuser | TS\_deleteuser\_01 | TS\_deleteuser\_01\_01 | Passed | No |
| TS\_deleteuser\_01\_02 | Passed | No |
| TS\_deleteuser\_02 | TS\_deleteuser\_02\_01 | Passed | No |
| TS\_deleteuser\_02\_02 | Passed | No |
|  | TS\_borrowbook | TS\_borrowbook\_01 | TS\_borrowbook\_01\_01 | Passed | No |
| TS\_borrowbook\_01\_02 | Passed | No |
| TS\_borrowbook\_02 | TS\_borrowbook\_02\_01 | Passed | No |
| TS\_borrowbook\_02\_02 | Passed | No |
|  | TS\_returnbook | TS\_returnbook\_01 | TS\_returnbook\_01\_01 | Passed | No |
| TS\_returnbook\_01\_02 | Passed | No |
| TS\_returnbook\_02 | TS\_returnbook\_02\_01 | Passed | No |
| TS\_returnbook\_02\_02 | Passed | No |