Application Requirements Specification
For
<Library Management System>

Prepared By:

<Haoyu Hu, Li Fu, Taixu Jin>

Purpose and Scope Statement

The purpose of this project is to provide a centralized platform for library administrators to manage books and students to use library services such as search, borrow and return basic services. The system will allow administrators to add, delete, and update book records, and the system will also allow students use basic library services such as search, borrow and return.
The scope of project is mainly three parts, one is overall UI interactive interface, including register, login and the other subfunction user interface, the other is overall interaction logic implementation of the entire back-end program, the last is database for implementing search, add, delete and update functions.

Requirements Narrative

In this project, this application has two user roles: librarian and student.

When a librarian account is created, they can log in through the librarian user interface. The librarian user interface includes the following functionalities:

1. Add Book Information: Librarians can add and book information, including book number, location, and borrowing information.
2. Search Book Information: Librarians can search for book information using various criteria, such as book title, author, or book number.
3. Delete book information: Librarians have the capability to delete book information by specifying the book number.
4. Update Book Information: Librarians can update book information by specifying the book number and updating relevant fields as listed in the table as item 1.

When a student user account is created, they can log in through the user interface designed for students. The student user interface includes the following functionalities:

1. Overview of Book Information: Students can view a summary of the available book information in the library.

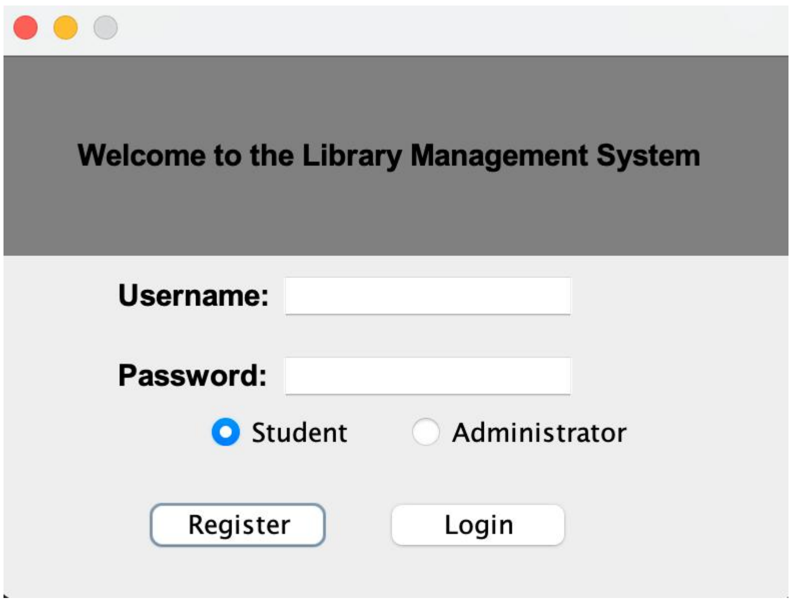
2. Book Retrieval Function: Students can search for books using criteria such as title, author, or book number.
3. Borrow book: Students can borrow books from library if books available.
4. Return book: Students can return books and then update the book borrow status.

Additionally, the application provides user registration and login interfaces:

1. Registration Page: Both librarians and students can register by selecting their user category (librarian or students) and providing information such as name, username, password, etc.
2. Login Page: Users can log in by selecting their role and entering their username and password.

Objectives

1. The Login screen



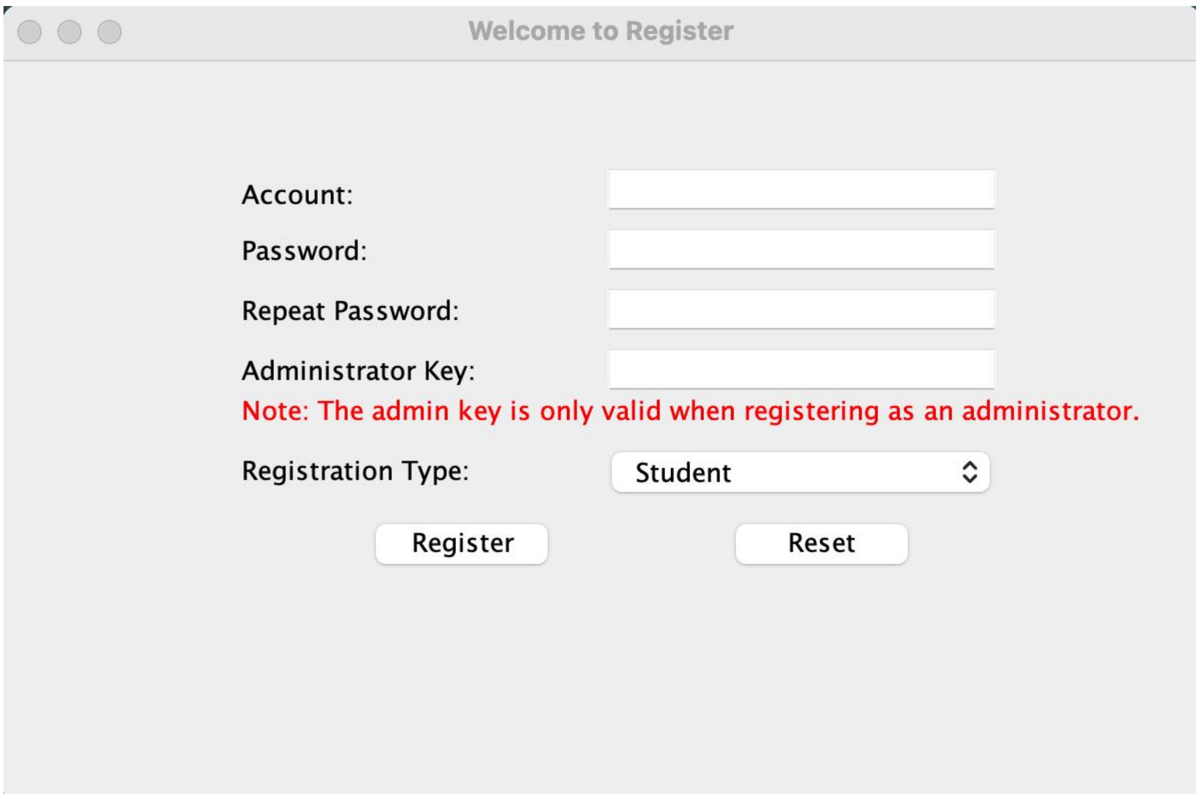
Welcome to the Library Management System

Username:

Password:

[✓ Student] Administrator

2. The registration screen



Welcome to Register

Account:

Password:

Repeat Password:

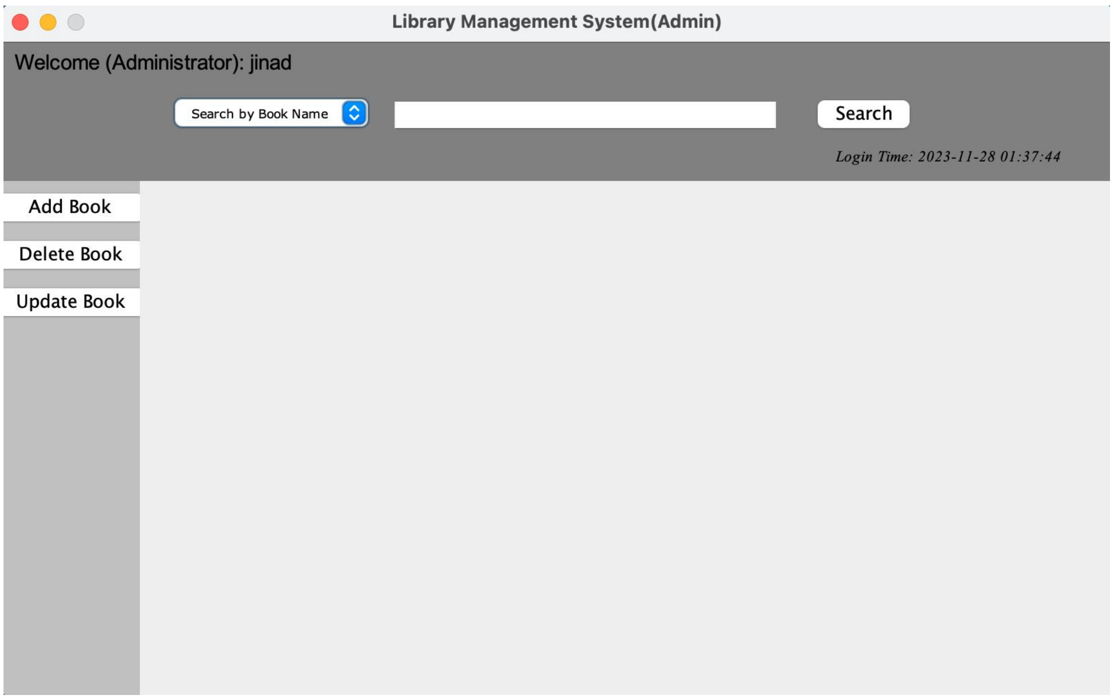
Administrator Key:

Note: The admin key is only valid when registering as an administrator.

Registration Type: Student

Register Reset

3. The screen of administration user



Library Management System(Admin)

Welcome (Administrator): jinad

Search by Book Name Search

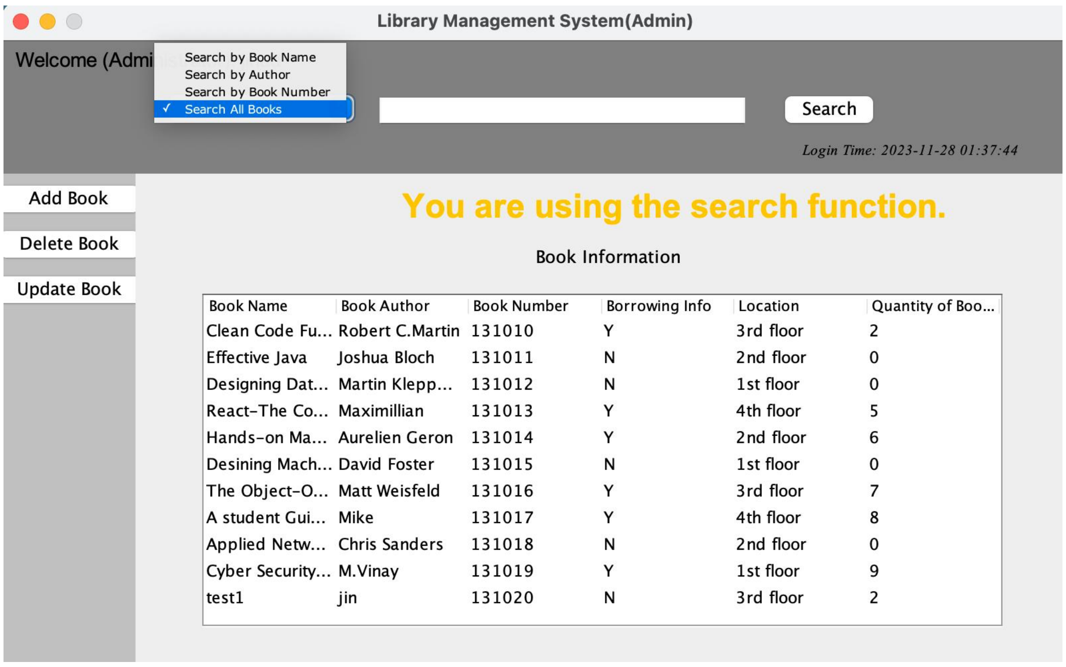
Login Time: 2023-11-28 01:37:44

Add Book

Delete Book

Update Book

2. The screen of administration search function



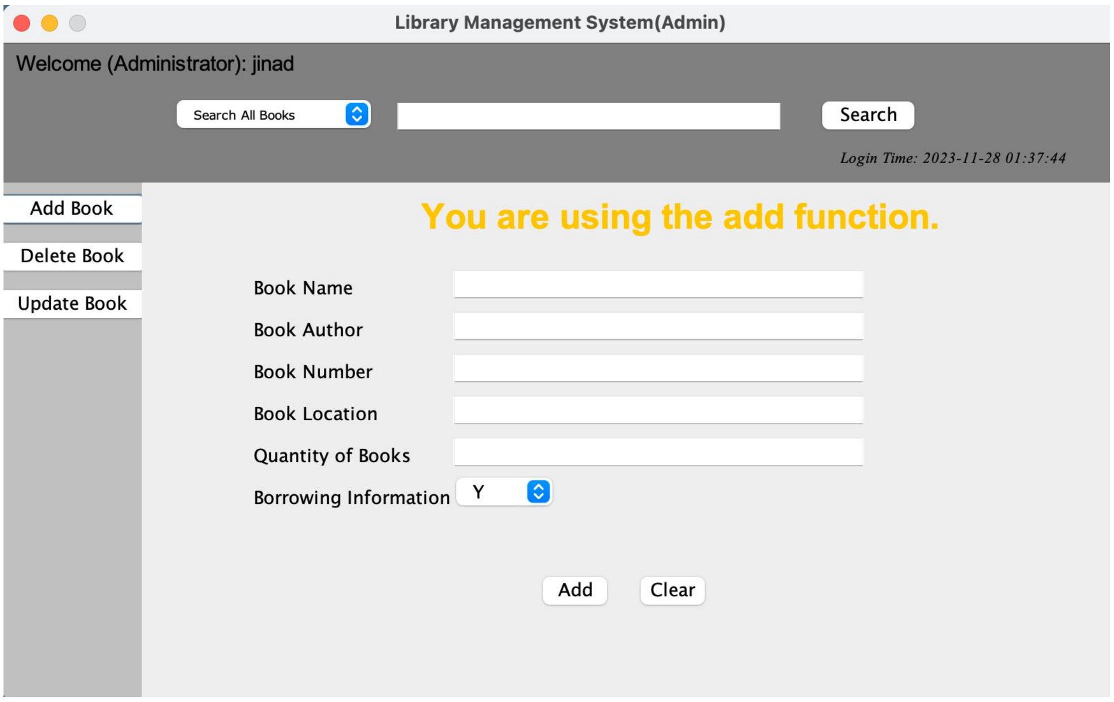
Login Time: 2023-11-28 01:37:44

You are using the search function.

Book Information

Book Name	Book Author	Book Number	Borrowing Info	Location	Quantity of Boo...
Clean Code Fu...	Robert C.Martin	131010	Y	3rd floor	2
Effective Java	Joshua Bloch	131011	N	2nd floor	0
Designing Dat...	Martin Klepp...	131012	N	1st floor	0
React-The Co...	Maximillian	131013	Y	4th floor	5
Hands-on Ma...	Aurelien Geron	131014	Y	2nd floor	6
Desining Mach...	David Foster	131015	N	1st floor	0
The Object-O...	Matt Weisfeld	131016	Y	3rd floor	7
A student Gui...	Mike	131017	Y	4th floor	8
Applied Netw...	Chris Sanders	131018	N	2nd floor	0
Cyber Security...	M.Vinay	131019	Y	1st floor	9
test1	jin	131020	N	3rd floor	2

3.The screen of administration add function



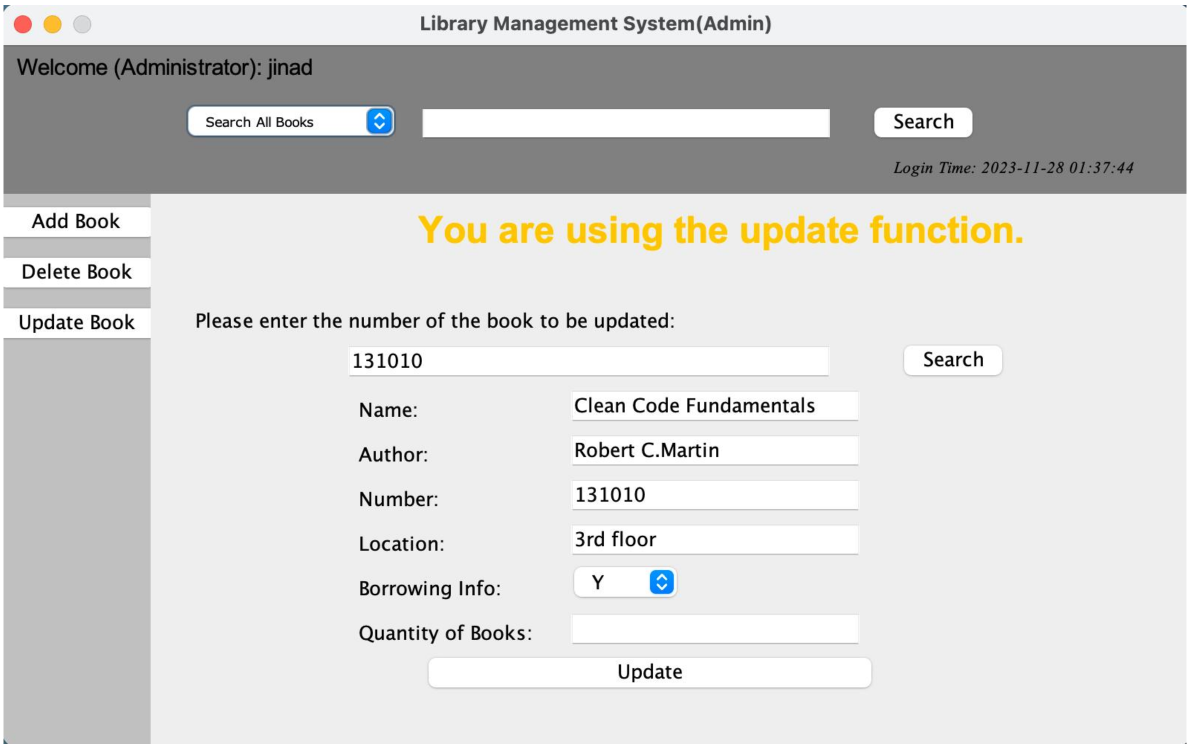
Welcome (Administrator): jinad

Login Time: 2023-11-28 01:37:44

You are using the add function.

Book Name	
Book Author	
Book Number	
Book Location	
Quantity of Books	
Borrowing Information	Y

4.The screen of administration update function


Library Management System(Admin)

Welcome (Administrator): jinad

Search All Books | Search

Login Time: 2023-11-28 01:37:44

Add Book
Delete Book
Update Book

You are using the update function.

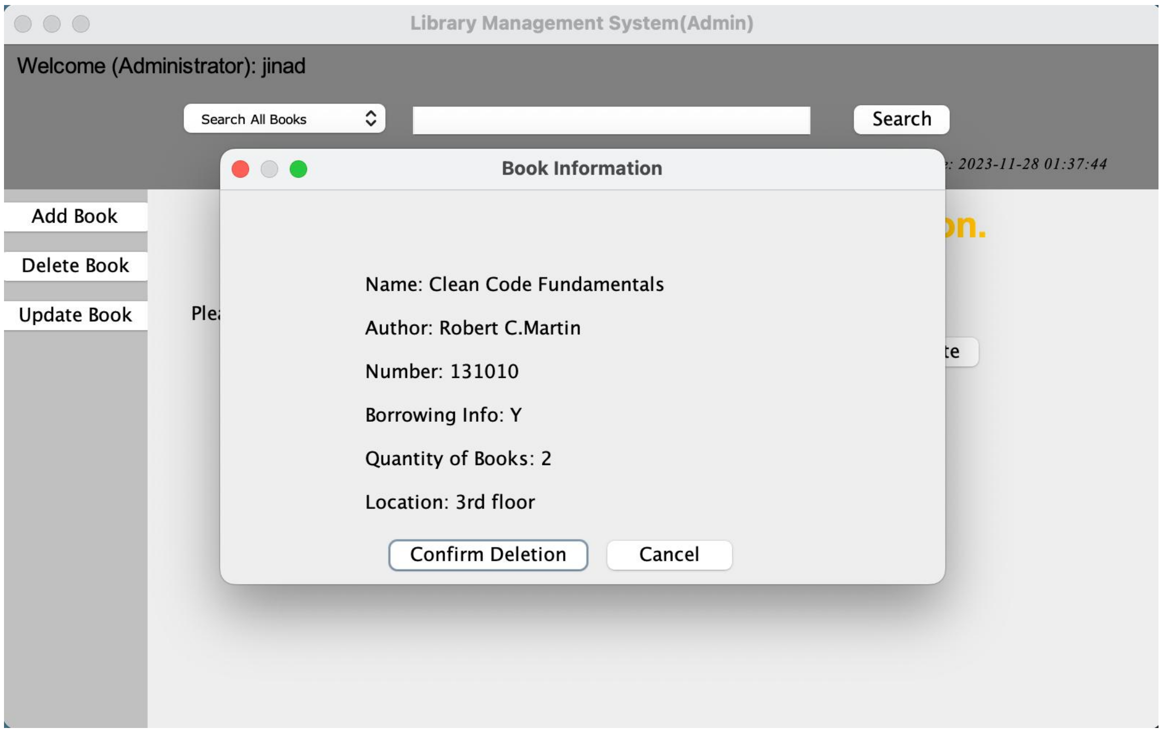
Please enter the number of the book to be updated:

131010 | Search

Name:	Clean Code Fundamentals
Author:	Robert C.Martin
Number:	131010
Location:	3rd floor
Borrowing Info:	Y
Quantity of Books:	

Update

5. The screen of administration delete function


Library Management System(Admin)

Welcome (Administrator): jinad

Search All Books | Search

Add Book
Delete Book
Update Book

Book Information

Name: Clean Code Fundamentals

Author: Robert C.Martin

Number: 131010

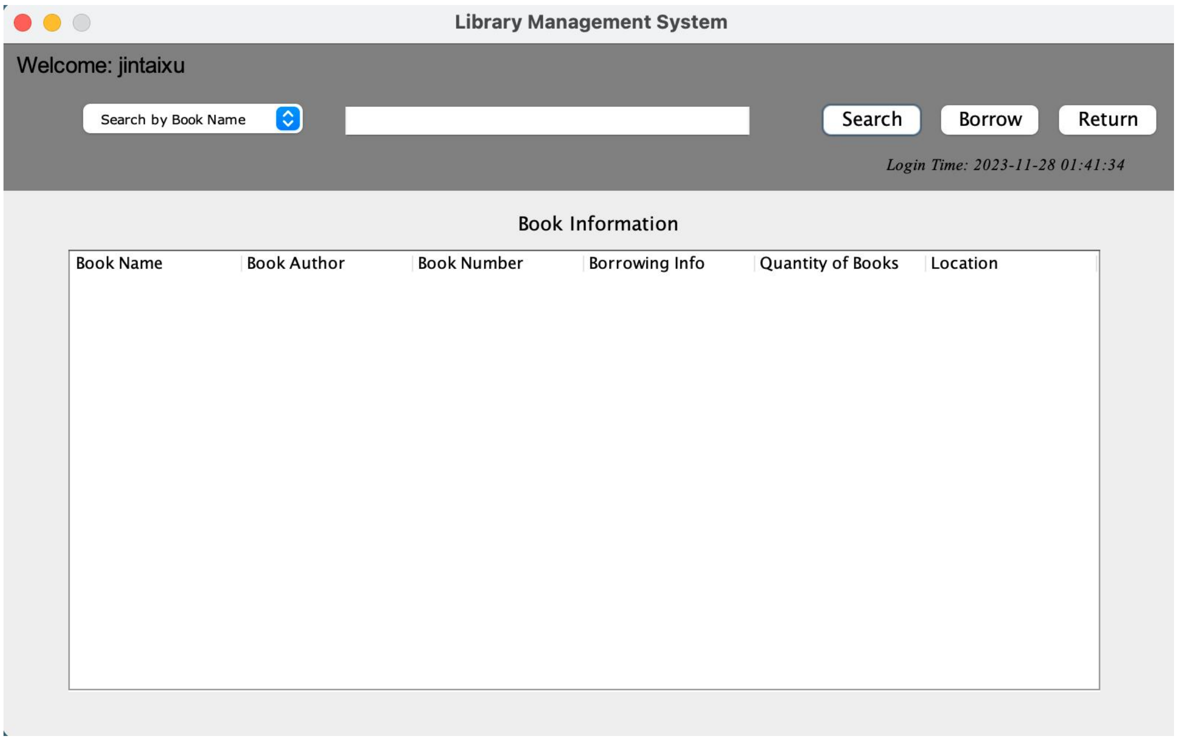
Borrowing Info: Y

Quantity of Books: 2

Location: 3rd floor

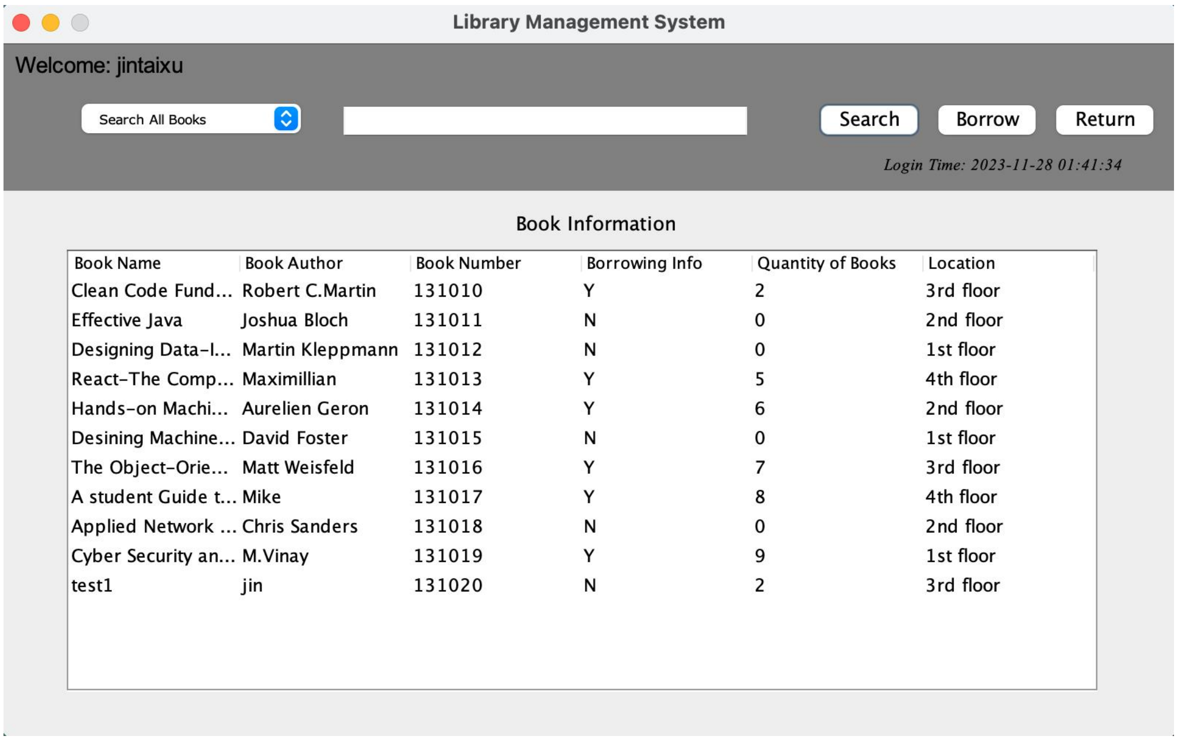
Confirm Deletion | Cancel

6. The screen of student user



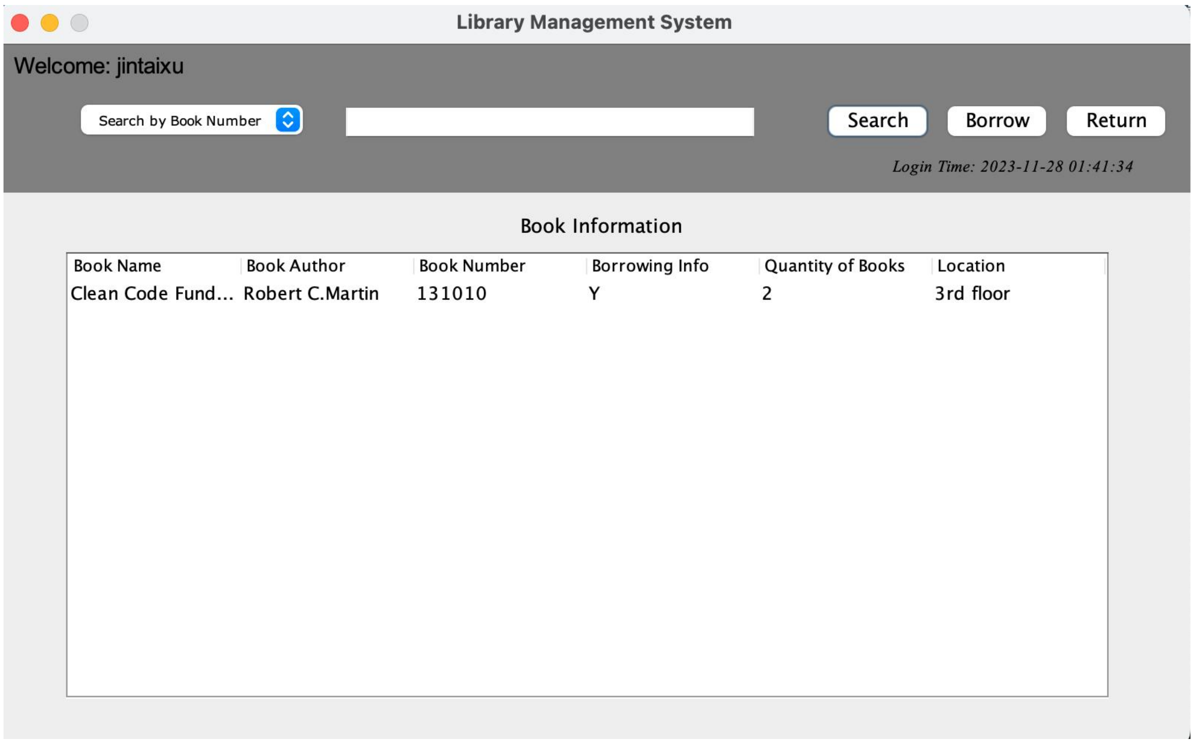
Book Name	Book Author	Book Number	Borrowing Info	Quantity of Books	Location

7. The overview function screen of student user



Book Name	Book Author	Book Number	Borrowing Info	Quantity of Books	Location
Clean Code Fund...	Robert C.Martin	131010	Y	2	3rd floor
Effective Java	Joshua Bloch	131011	N	0	2nd floor
Designing Data-I...	Martin Kleppmann	131012	N	0	1st floor
React-The Comp...	Maximillian	131013	Y	5	4th floor
Hands-on Machi...	Aurelien Geron	131014	Y	6	2nd floor
Desining Machine...	David Foster	131015	N	0	1st floor
The Object-Orie...	Matt Weisfeld	131016	Y	7	3rd floor
A student Guide t...	Mike	131017	Y	8	4th floor
Applied Network ...	Chris Sanders	131018	N	0	2nd floor
Cyber Security an...	M.Vinay	131019	Y	9	1st floor
test1	jin	131020	N	2	3rd floor

8. The book retrieval function of student user


Library Management System

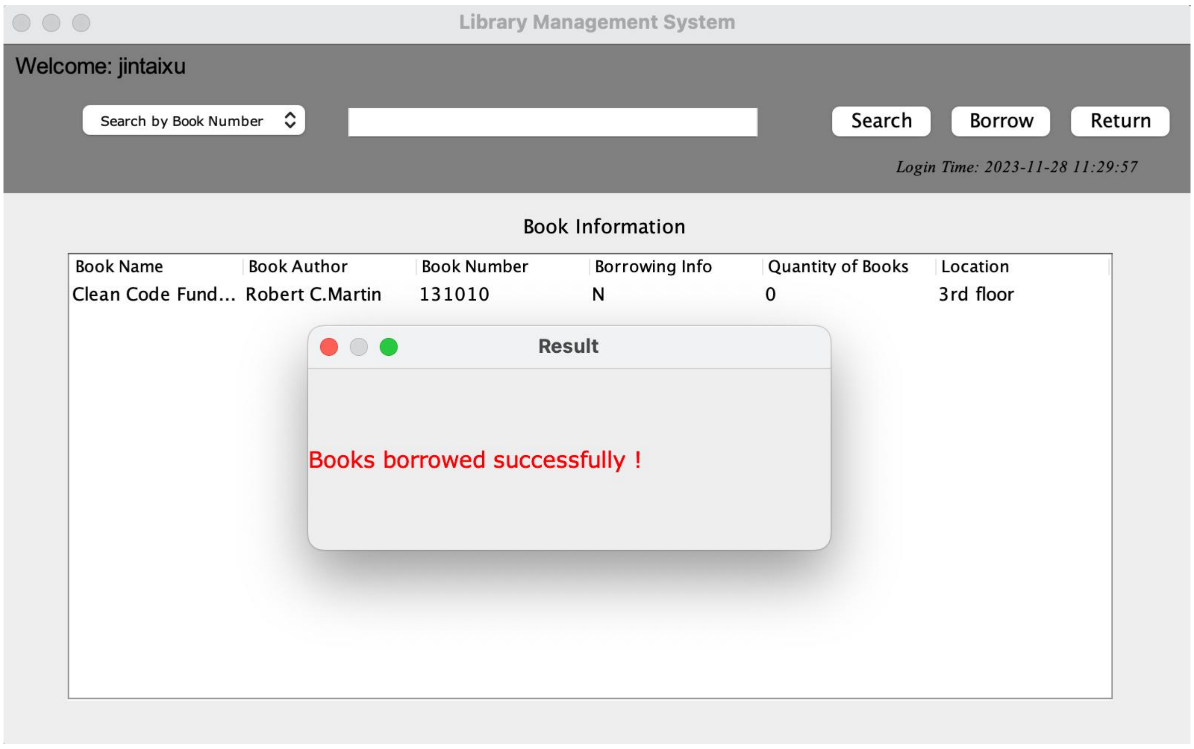
Welcome: jintaixu

Login Time: 2023-11-28 01:41:34

Book Information

Book Name	Book Author	Book Number	Borrowing Info	Quantity of Books	Location
Clean Code Fund...	Robert C.Martin	131010	Y	2	3rd floor

9. The book borrow function of student user


Library Management System

Welcome: jintaixu

Login Time: 2023-11-28 11:29:57

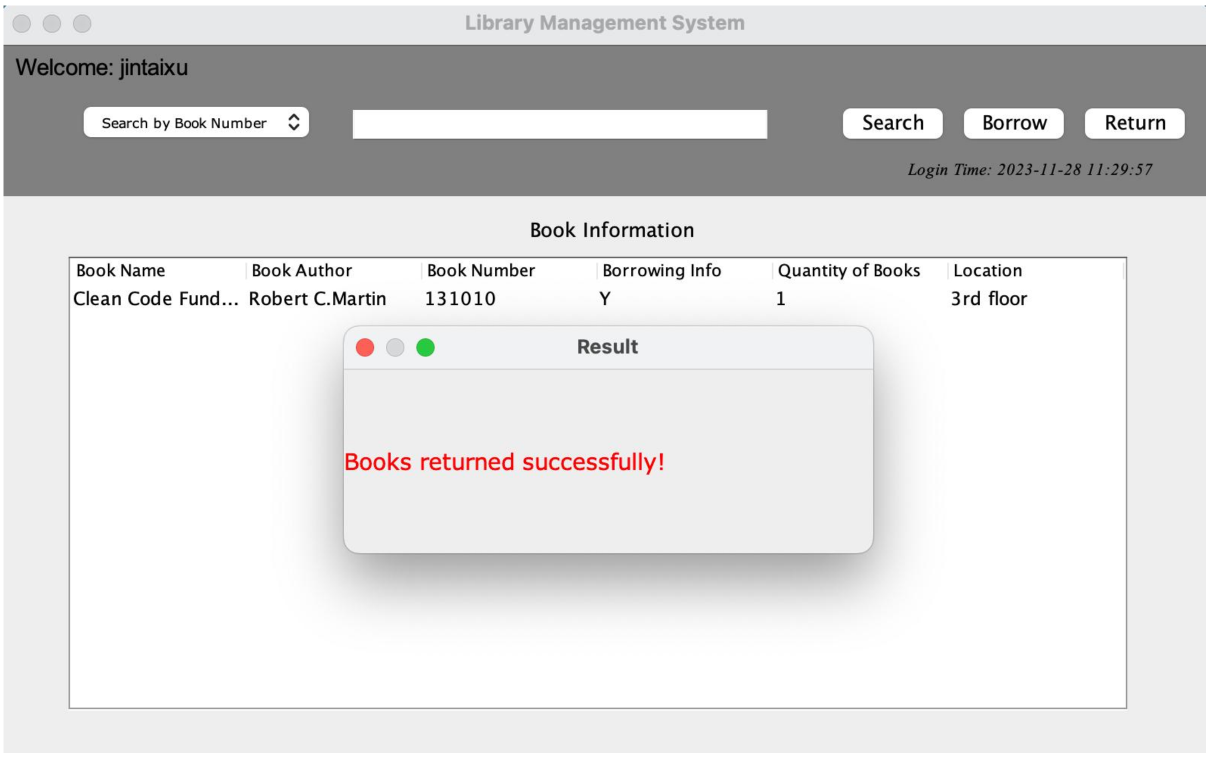
Book Information

Book Name	Book Author	Book Number	Borrowing Info	Quantity of Books	Location
Clean Code Fund...	Robert C.Martin	131010	N	0	3rd floor

Result

Books borrowed successfully !

10. The book return function of student user


Library Management System

Welcome: jintaixu

Login Time: 2023-11-28 11:29:57

Book Information

Book Name	Book Author	Book Number	Borrowing Info	Quantity of Books	Location
Clean Code Fund...	Robert C.Martin	131010	Y	1	3rd floor

Result

Books returned successfully!

Functional Specification

The overall programming follows MVC design pattern, so in this project we divide into five parts, and we will descript detail as below.

1. Manipulating data Function

This file represents the data and the business logic of the application. It is responsible for accessing the data layer, retrieving data, processing it, and sending it back to the view or updating the database as needed. (Because in this part, how to define methods are important, so methods will be depicted below)

Including classes:

(1) AdminDao.java

- findAllByAdminAndPasswd: Finds an admin by username and password.
- addBook: Adds a book to the database.
- deleteByNum: Deletes a book from the database based on the book number.
- updateBook: Updates book details in the database.

(2) RegistDao.java

- checkAdminCode: Checks if an admin code exists in the database.
- updateAdminCode: Updates the usage count of an admin code.
- userRegist: Registers a new user.
- adminRegist: Registers a new admin.
- findUserByCode: Finds a user by username.
- findAdminByCode: Finds an admin by username.

(3) UserDao.java

- findAllByUserAndPasswd: Finds a user by username and password.
- findBookByBookname: Retrieves a list of books by book name.
- findBookByAuthor: Retrieves a list of books by author name.
- findBookByNum: Retrieves a list of books by book number.
- findAllBook: Retrieves a list of all books.
- borrowBookByBookname: Borrows a book by book name.
- borrowBookByAuthor: Borrows a book by author name.
- borrowBookByNum: Borrows a book by book number.
- returnBookByBookname: Returns a book by book name.
- returnBookByAuthor: Returns a book by author name.
- returnBookByNum: Returns a book by book number.

- updateAfterBorrow: Updates book details after a borrow or return operation.

2. Objects define Function(Model file)

This package serves as a blueprint for the entire code. Each class in this package includes private attributes and setter methods for accessing and modifying these attributes. Each class in the package has a parameterized constructor to set these attributes during object creation. (Because in this part, defining attributes is more important, so attributes will be depicted below)

Including classes:

(1) Admin.java

- adminID (Type: Integer) - Represents the ID or identifier of the admin.
- username (Type: String) - Represents the username of the admin.
- password (Type: String) - Represents the password of the admin.

(2) AdminCode.java

- id (Type: int) - Represents the identifier or ID of the admin code.

- code (Type: String) - Represents the actual code or key.
- count (Type: int) - Represents the count or frequency associated with the code.

(3) Book.java

- bookid (Type: Integer) - Represents the identifier or ID of the book.
- bookname (Type: String) - Represents the name of the book.
- author (Type: String) - Represents the author of the book.
- num (Type: Long) - Represents a numeric value associated with the book.
- borrow (Type: String) - Represents whether the book can be borrowed or not.
- quantity (Type: Integer) - Represents the quantity of books available.
- location (Type: String) - Represents the location where the book is stored or kept.

(4) Register.java

- username (Type: String) - Represents the name of the user registering.
- password (Type: String) - Represents the password for the user account.
- adminCode (Type: String) - Represents the admin key used during registration.
- mold (Type: String) - Represents the type or category of registration.

(5) User.java

- userid (Type: Integer) - Represents the identifier or ID of the user.
- userName (Type: String) - Represents the username of the user.
- password (Type: String) - Represents the password associated with the user account.

3. Service Function

This file acts as an intermediary between the model and the UI. It listens to the input from the view, processes the user's data (with the help of the model), and returns the output display to the view. (Because in this part, how to define methods are important, so methods will be depicted below)

Including classes:

(1) AdminService.java

- addBook: Adds a book to the system after validating input and checking for its existence.
- findByNumber: Retrieves a book by its number if it exists.
- deleteByNum: Deletes a book based on its number.
- update: Modifies book details after validating the input.

(2) RegistService.java

- regist: Handles the registration process.
- checkAdminCode: Validates the provided admin code.
- isExist: Checks if a user or administrator is already registered.

(3) UserService.java

- checkCode: Validates the format of the username and password.
- login: Allows a regular user to log in.
- adminLogin: Allows an administrator to log in.

- findBooks: Searches for books based on specific criteria.
- borrowBooks: Searches for books to borrow based on specific criteria.
- returnBooks: Searches for books to return based on specific criteria.

4.UI design

This file is the user interface of the application. It displays the data (from the model) to the user and sends user commands (like button clicks, form submissions) to the controller.

Including classes:

(1) AdminView.java

(2) Index.java

(3) Login.java

(4) RegisterDialog.java

(5) UserView.java

5.utils

This file is to complete the database connection and some custom classes.

Including classes:

(1) JDBCUtil.java

(2) Tools.java

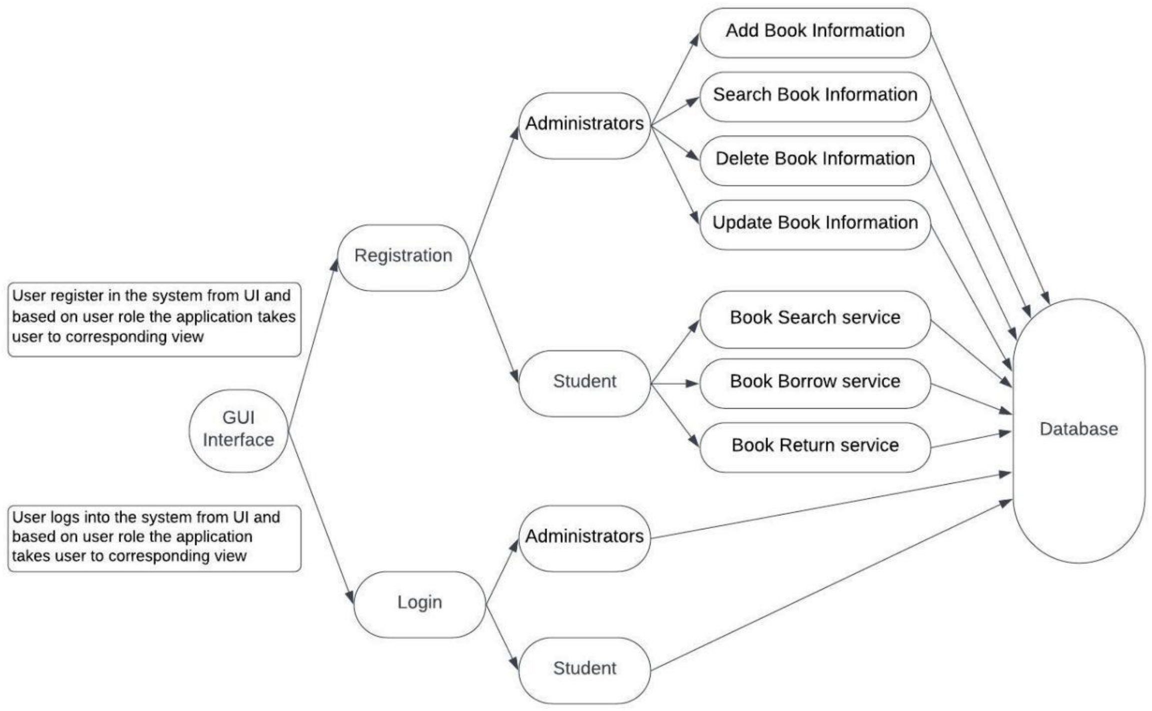
Technologies Needed:

Java

Java Swing gui for UI

Database – MySQL database

Logic Specification



The project combines Java Swing with MVC which likely provides a robust framework for creating a user-friendly interface while maintaining a clean separation of the application's data processing and user interaction logic. This approach is particularly beneficial for complex applications requiring a high degree of organization, maintainability, and scalability.

Java Swing:

1. Rich User Interface: Swing provides a comprehensive set of GUI components and allows for the creation of a more sophisticated and visually appealing user interface.
2. Platform Independence: Being part of Java, Swing applications are platform-independent, which means they can run on any operating system that supports Java.
3. Customizability: Swing components are highly customizable, allowing developers to create a unique look and feel for their applications.
4. Integration with MVC: Swing naturally supports the MVC pattern, making it easier to implement a clean separation of concerns, which is beneficial for the maintainability and scalability of the application.

MVC Pattern:

1. Organized Code Structure: MVC organizes the code in a way that separates the application's logic, UI, and data

management, which enhances code readability and maintainability.

2. Ease of Modification: Because the presentation and business logic are separated, changes in the user interface do not affect the data handling, and vice versa. This makes the application more adaptable and easier to update or modify.
3. Facilitates Team Development: Different teams can work on the Model, View, and Controller independently, improving development efficiency.
4. Reusability and Scalability: Models, being separated from Views and Controllers, can be reused across different parts of the application. Also, MVC supports scalability as the application grows in complexity.