

**Library Management System**

Bael, Kyle F

Gumbay, Jerome L.

Camba, Jhon Carlo S..

Clotario, Jim Latrill M.

Cañete, kent Andrei E.

Nuñez, Prinz Emmanuel B.

Amporingues, Vincent B.

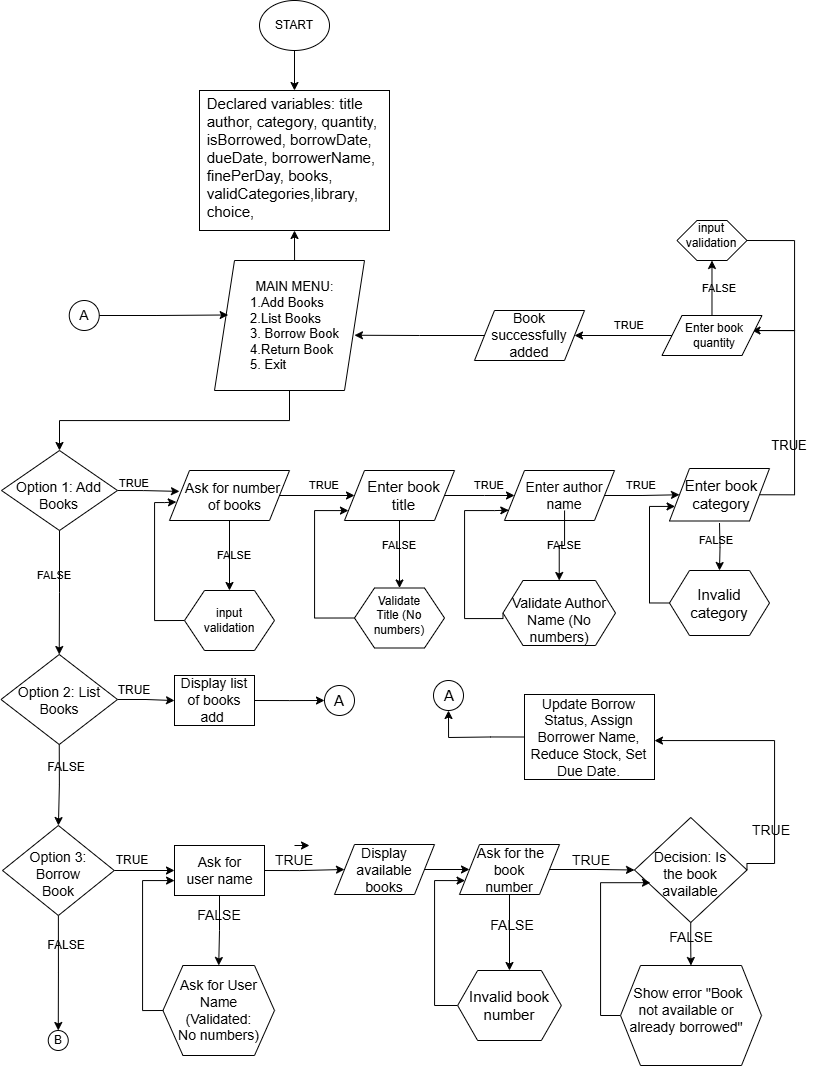
Rodrigo, Kenneth N.

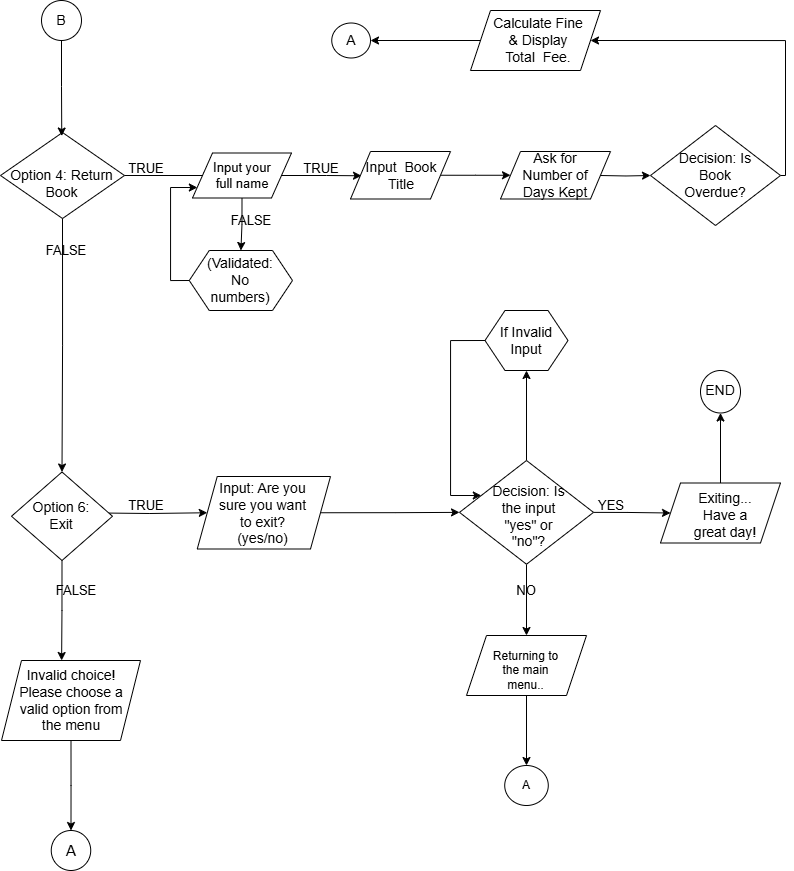
**INTRODUCTION**

Library Management System is a software application designed to facilitate the borrowing and returning of books in an organized manner. In traditional library systems, keeping track of borrowed books, due dates, and available stock can be challenging. This system automates these processes by allowing users to borrow books, track due dates, and calculate fines for late returns. By implementing such a system, libraries can efficiently manage their book inventory while providing a seamless experience for users.

Our system enables users to borrow books by entering their name, selecting a book from the available collection, and specifying the number of days they wish to keep it. Each book's availability is updated in real-time, ensuring accurate tracking of borrowed and returned books. Additionally, when a book is returned, the system checks for overdue days and applies a fine accordingly. This helps maintain discipline among borrowers and ensures books are returned on time for others to use.

**Flowchart**





**System Output and Code**

Code:

import java.time.LocalDate;

import java.time.temporal.ChronoUnit;

import java.util.ArrayList;

import java.util.Arrays;

import java.util.List;

import java.util.Scanner;

class Book {

    String title;

    String author;

    String category;

    int quantity;

    boolean isBorrowed;

    LocalDate borrowDate;

    LocalDate dueDate;

    String borrowerName;

    public Book(String title, String author, String category, int quantity) {

        this.title = title;

        this.author = author;

        this.category = category;

        this.quantity = quantity;

        this.isBorrowed = false;

        this.borrowerName = "";

    }

}

class Library {

    ArrayList<Book> books = new ArrayList<>();

    double finePerDay = 5.0;

    List<String> validCategories = Arrays.asList("Fiction", "Non-Fiction", "Novel", "Romance", "Children's Books");

    public void borrowBook(Scanner scanner) {

        System.out.print("Enter your full name: ");

        String userName = scanner.nextLine();

        System.out.println("\nAvailable Books:");

        listBooks();

        System.out.print("Enter the title of the book you want to borrow: ");

        String title = scanner.nextLine();

        System.out.print("Enter the number of days you want to borrow the book for: ");

        int borrowDays = scanner.nextInt();

        scanner.nextLine();

        for (Book book : books) {

            if (book.title.equalsIgnoreCase(title) && book.quantity > 0) {

                book.isBorrowed = true;

                book.borrowerName = userName;

                book.quantity--;

                book.borrowDate = LocalDate.now();

                book.dueDate = book.borrowDate.plusDays(borrowDays);

                System.out.println("Book borrowed successfully by " + userName + ". Please return it by " + book.dueDate + ".");

                return;

            }

        }

        System.out.println("Book not available or already fully borrowed.");

    }

    public void returnBook(Scanner scanner) {

        String userName;

        while (true) {

            System.out.print("Enter your full name: ");

            userName = scanner.nextLine();

            if (!userName.matches(".\*\\d.\*")) {

                break;

            }

            System.out.println("Invalid name! Names cannot contain numbers. Please enter a valid name.");

        }

        System.out.print("Enter book title to return: ");

        String title = scanner.nextLine();

        System.out.print("Enter the number of days you kept the book: ");

        int keptDays = scanner.nextInt();

        scanner.nextLine();

        for (Book book : books) {

            if (book.title.equalsIgnoreCase(title) && book.isBorrowed && book.borrowerName.equalsIgnoreCase(userName)) {

                LocalDate returnDate = book.borrowDate.plusDays(keptDays);

                long overdueDays = ChronoUnit.DAYS.between(book.dueDate, returnDate);

                double fine = (overdueDays > 0) ? overdueDays \* finePerDay : 0;

                book.isBorrowed = false;

                book.quantity++;

                book.borrowerName = "";

                book.borrowDate = null;

                book.dueDate = null;

                System.out.println("Thank you, " + userName + ", for returning the book: " + title);

                System.out.println("Due date was: " + book.dueDate);

                System.out.println("Returned on: " + returnDate);

                if (fine > 0) {

                    System.out.println("Overdue fine: " + fine);

                }

                System.out.println("Book successfully returned. Current stock: " + book.quantity);

                return;

            }

        }

        System.out.println("Book not found or wasn't borrowed by you.");

    }

    public void listBooks() {

        System.out.println("\n--------------------------------------------------------------------------------");

        System.out.printf("%-30s %-25s %-15s %-10s\n", "Title", "Author", "Category", "Quantity");

        System.out.println("--------------------------------------------------------------------------------");

        for (Book book : books) {

            System.out.printf("%-30s %-25s %-15s %-10d\n", book.title, book.author, book.category, book.quantity);

        }

        System.out.println("--------------------------------------------------------------------------------");

    }

    public void addBooks(Scanner scanner) {

        int numBooks;

        while (true) {

            System.out.print("Enter the number of books to add: ");

            if (scanner.hasNextInt()) {

                numBooks = scanner.nextInt();

                scanner.nextLine();

                break;

            } else {

                System.out.println("Invalid input! Please enter a valid number.");

                scanner.next();

            }

        }

        for (int i = 0; i < numBooks; i++) {

            System.out.println("Adding Book " + (i + 1) + ":");

            String title;

            while (true) {

                System.out.print("Enter book title: ");

                title = scanner.nextLine();

                if (!title.matches(".\*\\d.\*")) {

                    break;

                }

                System.out.println("Invalid title! Please enter a valid title.");

            }

            String author;

            while (true) {

                System.out.print("Enter author: ");

                author = scanner.nextLine();

                if (!author.matches(".\*\\d.\*")) {

                    break;

                }

                System.out.println("Invalid author name!  Please enter a valid name.");

            }

            String category;

            while (true) {

                System.out.println("Choose a category: " + validCategories);

                System.out.print("Enter book category: ");

                category = scanner.nextLine();

                if (validCategories.contains(category)) {

                    break;

                }

                System.out.println("Invalid category! Please choose from: " + validCategories);

            }

            System.out.print("Enter quantity: ");

            int quantity = scanner.nextInt();

            scanner.nextLine();

            books.add(new Book(title, author, category, quantity));

            System.out.println("Book " + (i + 1) + " added successfully: " + title + " by " + author + " (Category: " + category + ", Quantity: " + quantity + ")\n");

        }

        System.out.println("\nUpdated list of books:");

        listBooks();

    }

}

public class OnlineLibrary {

    public static void main(String[] args) {

        Scanner scanner = new Scanner(System.in);

        Library library = new Library();

        while (true) {

            System.out.println("\nWelcome to PHINMA COC Online Library:");

            System.out.println("1. Add Books");

            System.out.println("2. List Books");

            System.out.println("3. Borrow Book");

            System.out.println("4. Return Book");

            System.out.println("5. Exit");

            System.out.print("Choose an option: ");

            int choice = scanner.nextInt();

            scanner.nextLine();

            switch (choice) {

                case 1:

                    library.addBooks(scanner);

                    break;

                case 2:

                    library.listBooks();

                    break;

                case 3:

                    library.borrowBook(scanner);

                    break;

                case 4:

                    library.returnBook(scanner);

                    break;

                case 5:

                    while (true) {

                        System.out.print("Are you sure you want to exit? (yes/no): ");

                        String confirmExit = scanner.nextLine().trim().toLowerCase();

                        if (confirmExit.equals("yes")) {

                            System.out.println("Exiting... Have a great day!");

                            scanner.close();

                            return;

                        } else if (confirmExit.equals("no")) {

                            System.out.println("Returning to the main menu...");

                            break;

                        } else {

                            System.out.println("Invalid input! Please enter yes and 'no only.");

                        }

                    }

                    break;

                default:

                    System.out.println("Invalid choice! Please choose a valid option from the menu.");

                    break;

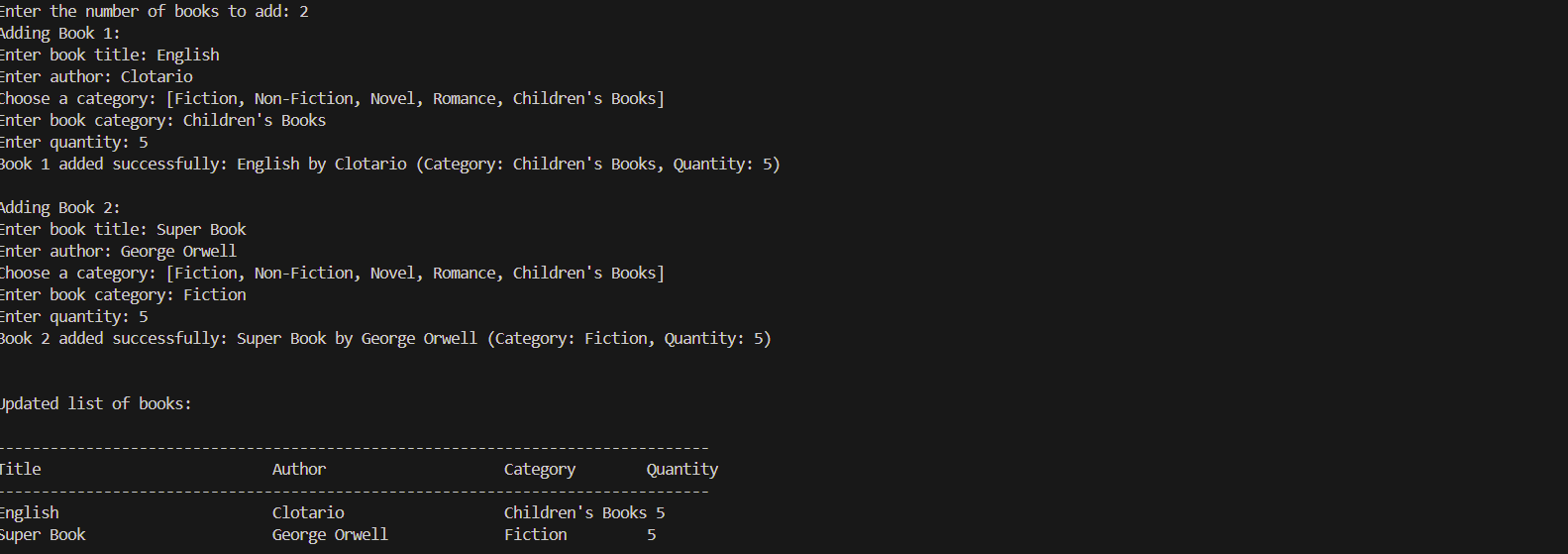
            }

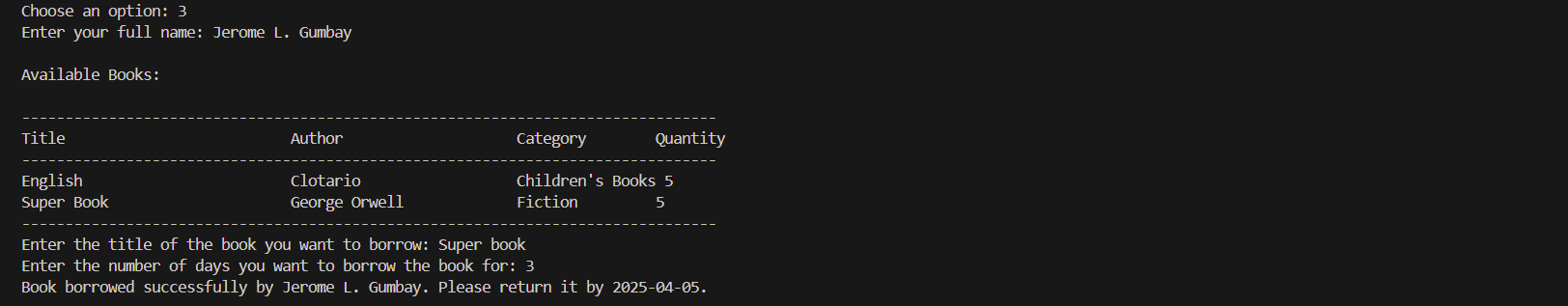
        }

}}

Output:









Screenshot 2025-04-02 095559

**DOCUMENTATION**



In this documentary, we share our journey as first-year IT students working on our second major programming project, creating a Java-based Library Management System. With little professional experience, we dove into applying everything we’ve learned in class to build a real-world system. The collage captures our collaboration, problem-solving, and learning as we faced the complexities of coding, from object-oriented design to integrating a simple console-type system. We slowly but surely brought our project to life through teamwork, debugging, and endless iterations.

As we worked together, we faced the typical struggles of beginners debugging endless errors, refining features, and learning how to think like developers. But each challenge became an opportunity to grow. We celebrated the small victories, whether it was fixing a bug or getting a feature to work. By the end of the project, we had created a fully functional Library Management-system, proving that with persistence, creativity, and teamwork, even first-year students can build something meaningful.