**Project on**

**Library Management System**

**Spring 2023**

**Course Code:** CSE 224, **Section:** 3

**Course Title:**  **Object Oriented Programming Laboratory**

**Submitted By**

**Name 1: MD Kawchar Hossen**

**ID : 221400055**

**Name 2: Noor Alam Islam Manik**

**ID : 221400044**

**Name 3: Lamia Alif Rouza**

**ID : 221400040**

****

**Department of Computer Science and Engineering**

**Faculty of Engineering and Technology**

**Eastern University**

TABLE OF CONTENTS

[Chapter 1 3](#_Toc137986756)

[**Introduction** 3](#_Toc137986757)

[1.1 Introduction: 3](#_Toc137986758)

[1.2 Objective: 3](#_Toc137986759)

[Chapter 2 4](#_Toc137986760)

[**Methodology and Tools** 4](#_Toc137986761)

[2.1 Methodology: 4](#_Toc137986762)

[2.2 Tools: 4](#_Toc137986763)

[Chapter 3 5](#_Toc137986764)

[**Design and Implementation** 5](#_Toc137986765)

[3.1 Use Case/Block Diagram of the System: 5](#_Toc137986766)

[3.2 Implementation(Code): 5](#_Toc137986767)

[Chapter 4 11](#_Toc137986768)

[**Result and Conclusion** 11](#_Toc137986769)

[4.1 Result( Screen shoot): 11](#_Toc137986770)

[4.2 Conclusion: 14](#_Toc137986771)

# Chapter 1

# **Introduction**

## 1.1 Introduction:

The library management system is a software application designed to help library administrators and students manage the library operations. It provides a user-friendly interface for the library users to view available books, borrow a book, and return a book. The system helps to improve the efficiency and accuracy of the library operations by keeping track of the library resources and their borrowing history. The objective of the system is to provide a user-friendly interface for the library users and administrators to manage the library operations more efficiently and effectively.

### 1.2 Objective:

The objective of the library management system is to provide a user-friendly interface for library administrators and students to manage the library operations. The system should allow them to view available books, borrow a book, return a book, and exit the program. The system is designed to improve the efficiency and accuracy of the library operations by providing a user-friendly interface for the library users. It allows the library administrators and students to easily view available books, borrow a book, and return a book. This helps them to keep track of their library resources and their borrowing history, which leads to increased borrowing and returning rates. The system should be easy to use and navigate, with clear instructions and prompts for the library users. It should also be able to handle a large number of library resources and users, and be scalable to fit the specific needs of the library. The objective of the system is to provide a user-friendly interface for the library users, who can easily manage their borrowing and returning processes. The system should help to improve the efficiency and accuracy of the library operations, and provide a more user-friendly interface for the library users and administrators.

#### 1.3 Expected Outcome**:**

The expected outcome of the library management system is to improve the efficiency and accuracy of the library operations. It will help the library administrators and students to keep track of their library resources and their borrowing history, which will lead to increased borrowing and returning rates, and a better management of the library resources. The system will also provide a more user-friendly interface for the library users, who will be able to manage their borrowing and returning processes more easily and efficiently. The expected outcome of the system is to provide a more efficient and effective way of managing the library operations, reduce the number of errors and inconsistencies that can occur during the library operations, and lead to a more accurate and reliable management of the library resources. The system will also improve the overall management of the library resources, increase the efficiency and effectiveness of the library operations, and lead to a more sustainable and long-term use of the library resources.

# Chapter 2

# **Methodology and Tools**

## 2.1 Methodology:

The library management system was developed using the Waterfall model as a methodology. The Waterfall model is a linear sequential approach to software development, where each phase of the development process must be completed before moving on to the next phase. The Waterfall model is widely used in software development, and is considered to be the most effective methodology for managing complex projects.

### 2.2 Tools:

The code is written in Java and can be run within an IDE such as Eclipse or NetBeans. It also requires the Java Development Kit (JDK) to be installed on the computer. The program can be compiled and run from the command line using a text editor or an integrated development environment (IDE). The program uses arrays to store data, and it includes error handling to ensure that the user inputs valid data. The program is easy to use and understand, with clear instructions and error handling. Overall, the tools required for this project are a Java IDE, a JDK, and a text editor or IDE.

# Chapter 3

# **Design and Implementation**

## 3.1 Use Case/Block Diagram of the System:

General Use Case

User

Librarian

### 3.2 Implementation(Code):

**package project;**

**import java.util.Scanner;**

**import java.util.Scanner;**

**public class LibraryManagementSystem {**

**private static String[] usernames = {"kawchar", "manik", "lamia"}; // sample usernames**

**private static String[] passwords = {"123", "456", "789"}; // sample passwords**

**private static String[] books = {"java", "data structure", "python", "how to become a good student"}; // sample books**

**private static String librarianUsername = "librarian"; // sample librarian username**

**private static String librarianPassword = "12345"; // sample librarian password**

**public static void main(String[] args) {**

**Scanner scanner = new Scanner(System.in);**

**boolean loggedIn = false;**

**String username = "";**

**String password = "";**

**boolean isLibrarian = false;**

**while (!loggedIn && !isLibrarian) {**

**boolean en = false;**

**System.out.println("Enter username:");**

**username = scanner.nextLine();**

**System.out.println("Enter password:");**

**password = scanner.nextLine();**

**if (isValidUser(username, password)) {**

**loggedIn = true;**

**System.out.println("Login successful!");**

**} else if(isLibrarian = (username.equals(librarianUsername) && password.equals(librarianPassword))) {**

**isLibrarian = true;**

**System.out.println("Login successful!");**

**}else{**

**System.out.println("Invalid username or password. Please try again.");**

**}**

**}**

**System.out.println("Welcome " + username + "!");**

**while (true) {**

**if (isLibrarian) {**

**System.out.println("You have access to all methods.");**

**} else {**

**System.out.println("You have access to viewBooks() only.");**

**}**

**System.out.println("Enter 1 to view available books.");**

**System.out.println("Enter 2 to borrow a book.");**

**System.out.println("Enter 3 to return a book.");**

**System.out.println("Enter 4 to exit.");**

**int choice = scanner.nextInt();**

**switch (choice) {**

**case 1:**

**if (isLibrarian) {**

**viewBooks();**

**} else {**

**viewBooks();**

**}**

**break;**

**case 2:**

**if (isLibrarian) {**

**borrowBook();**

**} else {**

**System.out.println("You do not have access to borrowBook().");**

**}**

**break;**

**case 3:**

**if (isLibrarian) {**

**returnBook();**

**} else {**

**System.out.println("You do not have access to returnBook().");**

**}**

**break;**

**case 4:**

**System.out.println("Exiting program...");**

**System.exit(0);**

**default:**

**System.out.println("Invalid choice.");**

**}**

**}**

**}**

**private static boolean isValidUser(String username, String password) {**

**for (int i = 0; i < usernames.length; i++) {**

**if (usernames[i].equals(username) && passwords[i].equals(password)) {**

**return true;**

**}**

**}**

**return false;**

**}**

**private static void viewBooks() {**

**System.out.println("Available books:");**

**for (String book : books) {**

**System.out.println(book);**

**}**

**}**

**private static void borrowBook() {**

**Scanner scanner = new Scanner(System.in);**

**System.out.println("Enter the name of the book you want to borrow:");**

**String bookName = scanner.nextLine();**

**boolean bookFound = false;**

**for (int i = 0; i < books.length; i++) {**

**if (books[i].equals(bookName)) {**

**bookFound = true;**

**books[i] = "not available";**

**System.out.println("Book borrowed successfully!");**

**break;**

**}**

**}**

**if (!bookFound) {**

**System.out.println("Book not found.");**

**}**

**}**

**private static void returnBook() {**

**Scanner scanner = new Scanner(System.in);**

**System.out.println("Enter the name of the book you want to return:");**

**String bookName = scanner.nextLine();**

**boolean bookFound = false;**

**for (int i = 0; i < books.length; i++) {**

**if (books[i]== "not available") {**

**bookFound = true;**

**books[i] = bookName;**

**System.out.println("Book returned successfully!");**

**break;**

**}**

**}**

**if (!bookFound) {**

**System.out.println("Book not found.");**

**}**

**}**

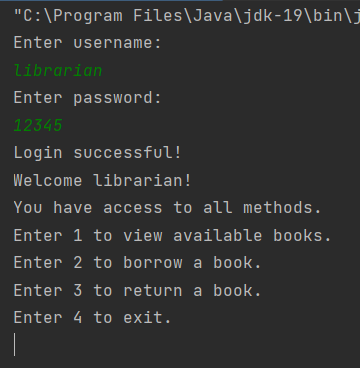
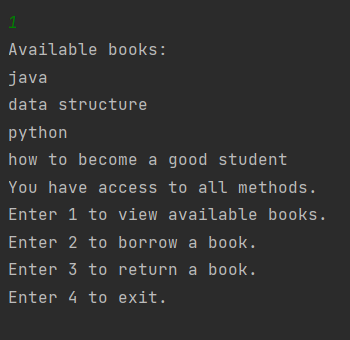
**}**

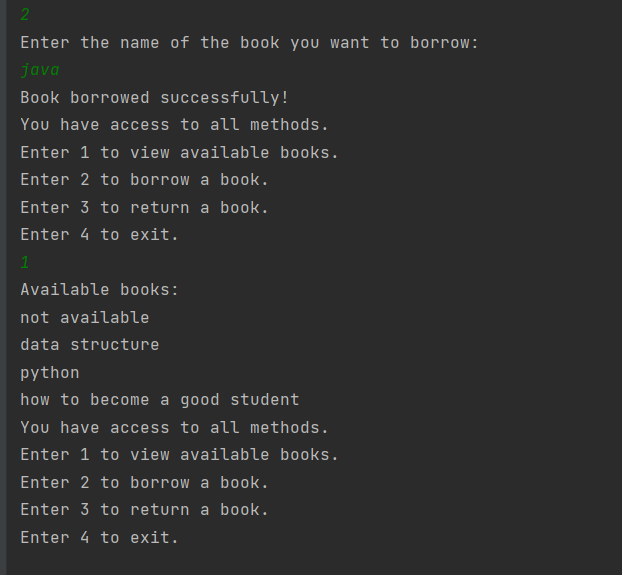
# Chapter 4

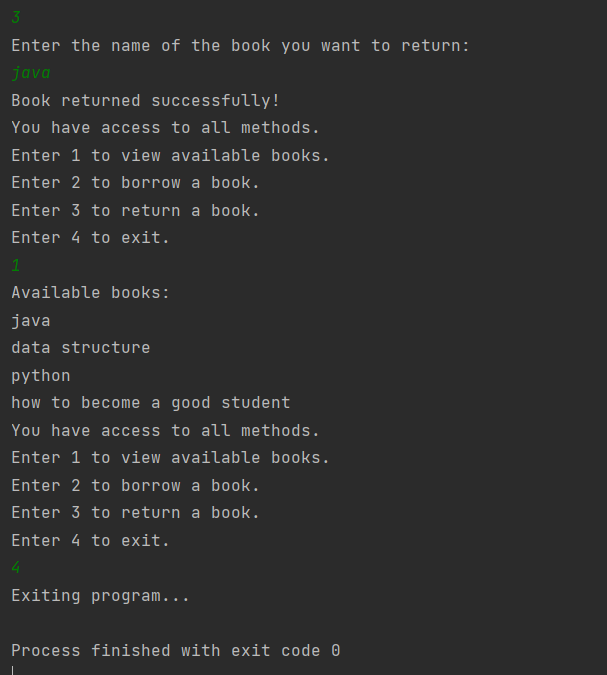
# **Result and Conclusion**

## 4.1 Result( Screen shoot):

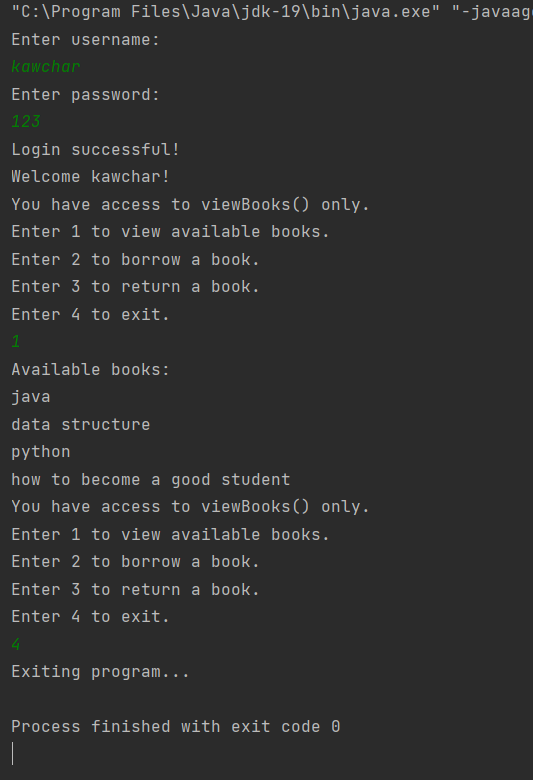
**For librarian:**

** **

****

****

**For user:**

****

### 4.2 Conclusion:

This is a Java program for a library management system. The program allows users to log in and access different methods depending on their role (librarian or regular user). The librarian can view all books, borrow books, and return books, while regular users can only view books. The program also includes sample data for usernames, passwords, and books. The program should be improved by adding more functionality to the librarian's access to the different methods. Overall, this is a basic implementation of a library management system and can be improved by adding more features and functionality.