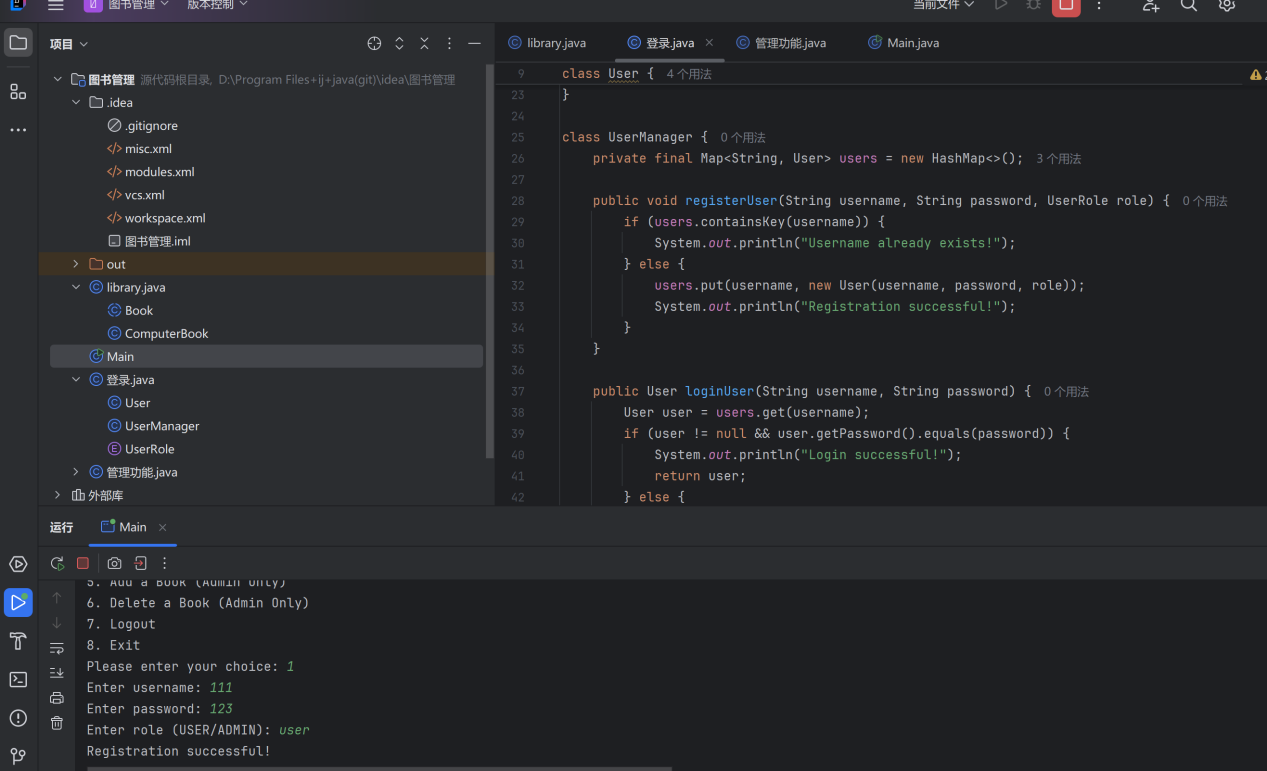
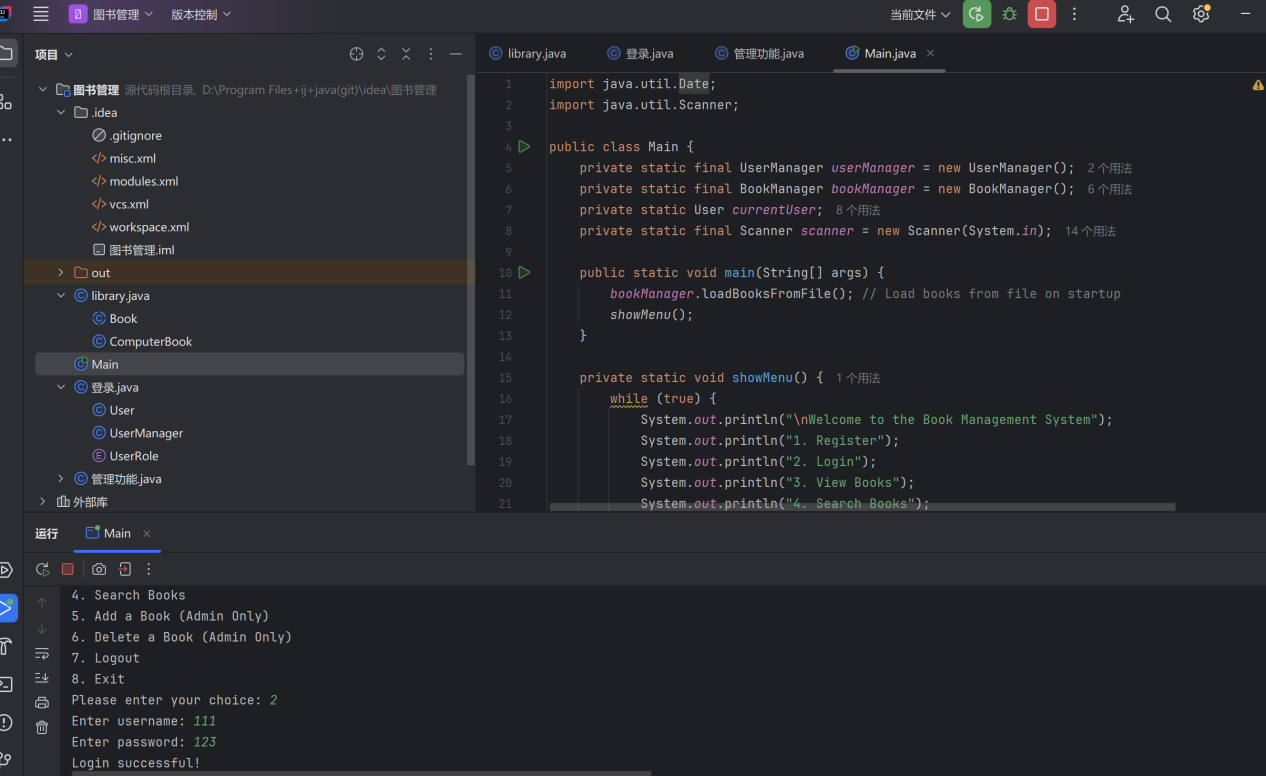


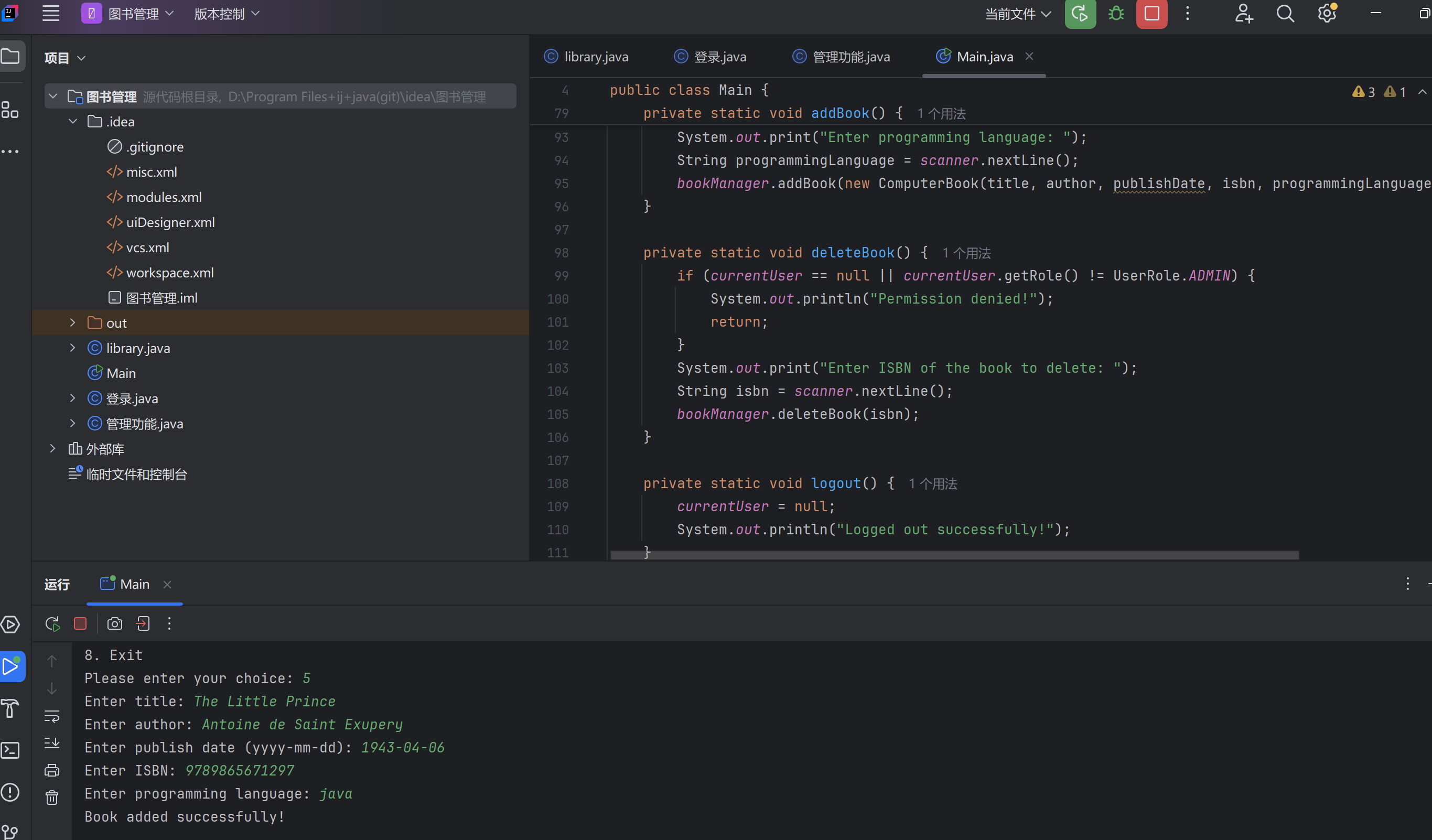
该过程注册用户user的用户名和密码



该过程登录用户user的用户名和密码



该过程admin add book 的具体信息



### 创建基类和子类

import java.io.Serializable;  
import java.util.Date;  
  
// 基类 Book  
abstract class Book implements Serializable {  
 private String title;  
 private String author;  
 private Date publishDate;  
 private String isbn;  
  
 public Book(String title, String author, Date publishDate, String isbn) {  
 this.title = title;  
 this.author = author;  
 this.publishDate = publishDate;  
 this.isbn = isbn;  
 }  
  
 public abstract void displayInfo();  
  
 // Getters and Setters  
 public String getTitle() { return title; }  
 public void setTitle(String title) { this.title = title; }  
 public String getAuthor() { return author; }  
 public void setAuthor(String author) { this.author = author; }  
 public Date getPublishDate() { return publishDate; }  
 public void setPublishDate(Date publishDate) { this.publishDate = publishDate; }  
 public String getIsbn() { return isbn; }  
 public void setIsbn(String isbn) { this.isbn = isbn; }  
}  
  
// 子类 ComputerBook  
class ComputerBook extends Book {  
 private String programmingLanguage;  
  
 public ComputerBook(String title, String author, Date publishDate, String isbn, String programmingLanguage) {  
 super(title, author, publishDate, isbn);  
 this.programmingLanguage = programmingLanguage;  
 }  
  
 @Override  
 public void displayInfo() {  
 System.*out*.println("Computer Book: " + getTitle());  
 System.*out*.println("Author: " + getAuthor());  
 System.*out*.println("Publish Date: " + getPublishDate());  
 System.*out*.println("ISBN: " + getIsbn());  
 System.*out*.println("Programming Language: " + programmingLanguage);  
 }  
}

### 用户注册和登录

import java.util.HashMap;  
import java.util.Map;  
import java.util.Scanner;  
  
enum UserRole {  
 *USER*, *ADMIN*}  
  
class User {  
 private final String username;  
 private final String password;  
 private final UserRole role;  
  
 public User(String username, String password, UserRole role) {  
 this.username = username;  
 this.password = password;  
 this.role = role;  
 }  
  
 public String getUsername() { return username; }  
 public String getPassword() { return password; }  
 public UserRole getRole() { return role; }  
}  
  
class UserManager {  
 private final Map<String, User> users = new HashMap<>();  
  
 public void registerUser(String username, String password, UserRole role) {  
 if (users.containsKey(username)) {  
 System.*out*.println("Username already exists!");  
 } else {  
 users.put(username, new User(username, password, role));  
 System.*out*.println("Registration successful!");  
 }  
 }  
  
 public User loginUser(String username, String password) {  
 User user = users.get(username);  
 if (user != null && user.getPassword().equals(password)) {  
 System.*out*.println("Login successful!");  
 return user;  
 } else {  
 System.*out*.println("Invalid username or password!");  
 return null;  
 }  
 }  
}

### 图书管理功能

import java.util.\*;  
import java.io.\*;  
  
class BookManager {  
 private List<Book> books = new ArrayList<>();  
 private static final String *BOOKS\_FILE* = "books.dat";  
  
 public void addBook(Book book) {  
 books.add(book);  
 saveBooksToFile();  
 System.*out*.println("Book added successfully!");  
 }  
  
 public void deleteBook(String isbn) {  
 books.removeIf(book -> book.getIsbn().equals(isbn));  
 saveBooksToFile();  
 System.*out*.println("Book deleted successfully!");  
 }  
  
 public void viewBooks() {  
 for (Book book : books) {  
 book.displayInfo();  
 }  
 }  
  
 public void searchBooks(String keyword) {  
 for (Book book : books) {  
 if (book.getTitle().contains(keyword) || book.getIsbn().contains(keyword)) {  
 book.displayInfo();  
 }  
 }  
 }  
  
 public void loadBooksFromFile() {  
 try (ObjectInputStream ois = new ObjectInputStream(new FileInputStream(*BOOKS\_FILE*))) {  
 books = (List<Book>) ois.readObject();  
 } catch (IOException | ClassNotFoundException e) {  
 System.*out*.println("No existing books found or error loading books.");  
 }  
 }  
  
 public void saveBooksToFile() {  
 try (ObjectOutputStream oos = new ObjectOutputStream(new FileOutputStream(*BOOKS\_FILE*))) {  
 oos.writeObject(books);  
 } catch (IOException e) {  
 System.*out*.println("Error saving books to file.");  
 }  
 }  
}

### 主程序和菜单交互

import java.util.Date;  
import java.util.Scanner;  
  
public class Main {  
 private static final UserManager *userManager* = new UserManager();  
 private static final BookManager *bookManager* = new BookManager();  
 private static User *currentUser*;  
 private static final Scanner *scanner* = new Scanner(System.*in*);  
  
 public static void main(String[] args) {  
 *bookManager*.loadBooksFromFile(); // Load books from file on startup  
 *showMenu*();  
 }  
  
 private static void showMenu() {  
 while (true) {  
 System.*out*.println("\nWelcome to the Book Management System");  
 System.*out*.println("1. Register");  
 System.*out*.println("2. Login");  
 System.*out*.println("3. View Books");  
 System.*out*.println("4. Search Books");  
 System.*out*.println("5. Add a Book (Admin Only)");  
 System.*out*.println("6. Delete a Book (Admin Only)");  
 System.*out*.println("7. Logout");  
 System.*out*.println("8. Exit");  
 System.*out*.print("Please enter your choice: ");  
 int choice = *scanner*.nextInt();  
 *scanner*.nextLine(); // Consume newline character  
 switch (choice) {  
 case 1: *register*(); break;  
 case 2: *login*(); break;  
 case 3: *viewBooks*(); break;  
 case 4: *searchBooks*(); break;  
 case 5: *addBook*(); break;  
 case 6: *deleteBook*(); break;  
 case 7: *logout*(); break;  
 case 8: *exit*(); break;  
 default: System.*out*.println("Invalid choice! Please try again."); break;  
 }  
 }  
 }  
  
 private static void register() {  
 System.*out*.print("Enter username: ");  
 String username = *scanner*.nextLine();  
 System.*out*.print("Enter password: ");  
 String password = *scanner*.nextLine();  
 System.*out*.print("Enter role (USER/ADMIN): ");  
 UserRole role = UserRole.*valueOf*(*scanner*.nextLine().toUpperCase());  
 *userManager*.registerUser(username, password, role);  
 }  
  
 private static void login() {  
 System.*out*.print("Enter username: ");  
 String username = *scanner*.nextLine();  
 System.*out*.print("Enter password: ");  
 String password = *scanner*.nextLine();  
 *currentUser* = *userManager*.loginUser(username, password);  
 }  
  
 private static void viewBooks() {  
 if (*currentUser* == null) {  
 System.*out*.println("Please login first!");  
 return;  
 }  
 *bookManager*.viewBooks();  
 }  
  
 private static void searchBooks() {  
 if (*currentUser* == null) {  
 System.*out*.println("Please login first!");  
 return;  
 }  
 System.*out*.print("Enter keyword to search: ");  
 String keyword = *scanner*.nextLine();  
 *bookManager*.searchBooks(keyword);  
 }  
  
 private static void addBook() {  
 if (*currentUser* == null || *currentUser*.getRole() != UserRole.*ADMIN*) {  
 System.*out*.println("Permission denied!");  
 return;  
 }  
 System.*out*.print("Enter title: ");  
 String title = *scanner*.nextLine();  
 System.*out*.print("Enter author: ");  
 String author = *scanner*.nextLine();  
 System.*out*.print("Enter publish date (yyyy-mm-dd): ");  
 String publishDateStr = *scanner*.nextLine();  
 Date publishDate = null; // You need to parse the date string here using SimpleDateFormat or similar method.  
 System.*out*.print("Enter ISBN: ");  
 String isbn = *scanner*.nextLine();  
 System.*out*.print("Enter programming language: ");  
 String programmingLanguage = *scanner*.nextLine();  
 *bookManager*.addBook(new ComputerBook(title, author, publishDate, isbn, programmingLanguage));  
 }  
  
 private static void deleteBook() {  
 if (*currentUser* == null || *currentUser*.getRole() != UserRole.*ADMIN*) {  
 System.*out*.println("Permission denied!");  
 return;  
 }  
 System.*out*.print("Enter ISBN of the book to delete: ");  
 String isbn = *scanner*.nextLine();  
 *bookManager*.deleteBook(isbn);  
 }  
  
 private static void logout() {  
 *currentUser* = null;  
 System.*out*.println("Logged out successfully!");  
 }  
  
 private static void exit() {  
 *bookManager*.saveBooksToFile(); // Save books to file before exiting  
 System.*exit*(0);  
 }  
}