```
package Library_management_system.newpackage;

import java.io.*;
import java.util.ArrayList;
import java.util.Scanner;

// Book class
class Book {
    private String name;
    private String author;
    private int quantity;

    // Constructor
    public Book(String name, String author, int quantity) {
        this.name = name;
        this.author = author;
        this.quantity = quantity;
    }

    // Getter methods
    public String getName() {
        return name;
    }

    public String getAuthor() {
        return author;
    }

    public int getQuantity() {
        return quantity;
    }

    // Setter method
    public void setQuantity(int quantity) {
        this.quantity = quantity;
    }
}

// Library class
class Library {
    private final String booksFile = "books.txt";
    private final String hiredBooksFile = "hired_books.txt";
    private ArrayList<Book> books;
```

```
// Constructor
public Library() {
    books = new ArrayList<>();
    loadBooks();
}

// Load books from file
private void loadBooks() {
    try (BufferedReader reader = new BufferedReader(new FileReader(booksFile))) {
        String line;
        while ((line = reader.readLine()) != null) {
            String[] parts = line.split(",");
            String name = parts[0];
            String author = parts[1];
            int quantity = Integer.parseInt(parts[2]);
            books.add(new Book(name, author, quantity));
        }
    } catch (IOException e) {
        e.printStackTrace();
    }
}

// Save books to file
private void saveBooks() {
    try (BufferedWriter writer = new BufferedWriter(new FileWriter(booksFile))) {
        for (Book book : books) {
            writer.write(book.getName() + "," + book.getAuthor() + "," + book.getQuantity() + "\n");
        }
    } catch (IOException e) {
        e.printStackTrace();
    }
}

// Add a book to the library
public void addBook(String name, String author, int quantity) {
    for (Book existingBook : books) {
        if (existingBook.getName().equals(name) && existingBook.getAuthor().equals(author)) {
            existingBook.setQuantity(existingBook.getQuantity() + quantity);
            saveBooks();
            return;
        }
    }
    books.add(new Book(name, author, quantity));
    saveBooks();
```

```
    }

    // Record a book lent to a student
    private void recordHiredBook(String name, String author, String studentName, String
studentEmail) {
        try (BufferedWriter writer = new BufferedWriter(new FileWriter(hiredBooksFile, true))) {
            writer.write(name + "," + author + "," + studentName + "," + studentEmail + "\n");
        } catch (IOException e) {
            e.printStackTrace();
        }
    }

    // Lend a book to a student
    public void lendBook(String name, String author, String studentName, String studentEmail) {
        for (Book book : books) {
            if (book.getName().equals(name) && book.getAuthor().equals(author)) {
                if (book.getQuantity() > 0) {
                    book.setQuantity(book.getQuantity() - 1);
                    saveBooks();
                    recordHiredBook(name, author, studentName, studentEmail);
                    System.out.println("Book '" + name + "' by " + author + " lent successfully to " +
studentName);
                    return;
                } else {
                    System.out.println("Book '" + name + "' by " + author + " is out of stock.");
                    return;
                }
            }
        }
        System.out.println("Book '" + name + "' by " + author + " not found in the library.");
    }

    // Display students who borrowed a specific book
    public void whoHiredBook(String name, String author) {
        try (BufferedReader reader = new BufferedReader(new FileReader(hiredBooksFile))) {
            String line;
            System.out.println("Students who borrowed '" + name + "' by " + author + ":");
            while ((line = reader.readLine()) != null) {
                String[] parts = line.split(",");
                if (parts[0].equals(name) && parts[1].equals(author)) {
                    System.out.println("Name: " + parts[2] + ", Email: " + parts[3]);
                }
            }
```

```
        } catch (IOException e) {
            e.printStackTrace();
        }
    }

    // Return a book to the library
    public void returnBook(String name, String author) {
        for (Book book : books) {
            if (book.getName().equals(name) && book.getAuthor().equals(author)) {
                book.setQuantity(book.getQuantity() + 1);
                saveBooks();
                System.out.println("Book '" + name + "' by " + author + " returned successfully.");
                return;
            }
        }
        System.out.println("Book '" + name + "' by " + author + " not found in the library.");
    }

    // Delete a book from the library
    public void deleteBook(String name, String author) {
        for (Book book : books) {
            if (book.getName().equals(name) && book.getAuthor().equals(author)) {
                books.remove(book);
                saveBooks();
                System.out.println("Book '" + name + "' by " + author + " deleted successfully.");
                return;
            }
        }
        System.out.println("Book '" + name + "' by " + author + " not found in the library.");
    }

    // Display all books in the library
    public void displayBooks() {
        System.out.println("Books in the library:");
        for (Book book : books) {
            System.out.println("Name: " + book.getName() + ", Author: " + book.getAuthor() + ",
Quantity: " + book.getQuantity());
        }
    }
}

// Main class
public class LibraryManagementSystem {
    public static void main(String[] args) {
```

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

System.out.println("Welcome to the Library Management System");
System.out.println("Choose an option:");
System.out.println("1. As Author");
System.out.println("2. As Student");
System.out.print("Enter your choice: ");
int option = scanner.nextInt();
scanner.nextLine(); // Consume newline

switch (option) {
    case 1:
        // Author options
        Author author = new Author();
        System.out.println("Choose an option:");
        System.out.println("1. Create Author Account");
        System.out.println("2. Login as Author");
        int authorOption = scanner.nextInt();
        scanner.nextLine(); // Consume newline
        switch (authorOption) {
            case 1:
                author.createAccount();
                break;
            case 2:
                System.out.print("Enter your name: ");
                String authorName = scanner.nextLine();
                System.out.print("Enter your email: ");
                String authorEmail = scanner.nextLine();
                if (author.login(authorName, authorEmail)) {
                    Library library = new Library();
                    boolean running = true;
                    while (running) {
                        // Author menu
                        System.out.println("1. Add Book");
                        System.out.println("2. Delete Book");
                        System.out.println("3. Display Books");
                        System.out.println("4. Students who borrowed a book");
                        System.out.println("5. Exit");
                        System.out.print("Enter your choice: ");
                        int choice = scanner.nextInt();
                        scanner.nextLine(); // Consume newline
                        switch (choice) {
                            case 1:
                                // Add a book
```

```
                    System.out.print("Enter Book Name: ");
                    String bookName = scanner.nextLine();
                    System.out.print("Enter Author Name: ");
                    String bookAuthor = scanner.nextLine();
                    System.out.print("Enter Quantity: ");
                    int quantity = scanner.nextInt();
                    library.addBook(bookName, bookAuthor, quantity);
                    break;
                case 2:
                    // Delete a book
                    System.out.print("Enter Book Name to delete: ");
                    String bookToDelete = scanner.nextLine();
                    System.out.print("Enter Author Name to delete: ");
                    String authorToDelete = scanner.nextLine();
                    library.deleteBook(bookToDelete, authorToDelete);
                    break;
                case 3:
                    // Display all books
                    library.displayBooks();
                    break;
                case 4:
                    // Display students who borrowed a book
                    System.out.print("Enter Book Name: ");
                    String hiredBookName = scanner.nextLine();
                    System.out.print("Enter Author Name: ");
                    String hiredAuthorName = scanner.nextLine();
                    library.whoHiredBook(hiredBookName, hiredAuthorName);
                    break;
                case 5:
                    running = false;
                    System.out.println("Exiting Library Management System. Thank you!");
                    break;
                default:
                    System.out.println("Invalid choice. Please try again.");
                }
            }
        }
        break;
    default:
        System.out.println("Invalid option.");
    }
    break;
case 2:
    // Student options
```

```
                Student student = new Student();
                System.out.println("Choose an option:");
                System.out.println("1. Create Student Account");
                System.out.println("2. Login as Student");
                int studentOption = scanner.nextInt();
                scanner.nextLine(); // Consume newline
                switch (studentOption) {
                    case 1:
                        student.createAccount();
                        break;
                    case 2:
                        System.out.print("Enter your name: ");
                        String studentName = scanner.nextLine();
                        System.out.print("Enter your email: ");
                        String studentEmail = scanner.nextLine();
                        if (student.login(studentName, studentEmail)) {
                            Library library = new Library();
                            boolean running = true;
                            while (running) {
                                // Student menu
                                System.out.println("1. Display Books");
                                System.out.println("2. Lend Book");
                                System.out.println("3. Return Book");
                                System.out.println("4. Exit");
                                System.out.print("Enter your choice: ");
                                int choice = scanner.nextInt();
                                scanner.nextLine(); // Consume newline
                                switch (choice) {
                                    case 1:
                                        // Display all books
                                        library.displayBooks();
                                        break;
                                    case 2:
                                        // Lend a book
                                        System.out.print("Enter Book Name to lend: ");
                                        String bookToLend = scanner.nextLine();
                                        System.out.print("Enter Author Name to lend: ");
                                        String authorToLend = scanner.nextLine();
                                        library.lendBook(bookToLend, authorToLend, studentName,
studentEmail);
                                        break;
                                    case 3:
                                        // Return a book
                                        System.out.print("Enter Book Name to return: ");
```

```
                        String bookToReturn = scanner.nextLine();
                        System.out.print("Enter Author Name to return: ");
                        String authorToReturn = scanner.nextLine();
                        library.returnBook(bookToReturn, authorToReturn);
                        break;
                    case 4:
                        running = false;
                        System.out.println("Exiting Library Management System. Thank you!");
                        break;
                    default:
                        System.out.println("Invalid choice. Please try again.");
                }
            }
        }
        break;
    default:
        System.out.println("Invalid option.");
}
break;
default:
    System.out.println("Invalid option.");
}

scanner.close();
}
}

// Author class
class Author {
    private final String authorDetailsFile = "authorDetails.txt";
    private String name;
    private String email;

    // Constructor
    public Author() {
        // Constructor
    }

    // Create author account
    public void createAccount() {
        try {
            BufferedWriter writer = new BufferedWriter(new FileWriter(authorDetailsFile));
            Scanner scanner = new Scanner(System.in);
            System.out.print("Enter your name: ");
```

```
            name = scanner.nextLine();
            System.out.print("Enter your email: ");
            email = scanner.nextLine();
            writer.write(name + "," + email);
            writer.close();
            System.out.println("Author account created successfully.");
        } catch (IOException e) {
            e.printStackTrace();
        }
    }

    // Author login
    public boolean login(String name, String email) {
        try {
            BufferedReader reader = new BufferedReader(new FileReader(authorDetailsFile));
            String line;
            while ((line = reader.readLine()) != null) {
                String[] parts = line.split(",");
                if (parts[0].equals(name) && parts[1].equals(email)) {
                    System.out.println("Login successful as Author.");
                    return true;
                }
            }
            reader.close();
        } catch (IOException e) {
            e.printStackTrace();
        }
        System.out.println("Invalid credentials. Please try again.");
        return false;
    }
}

// Student class
class Student {
    private final String studentDetailsFile = "studentDetails.txt";
    private String name;
    private String email;

    // Constructor
    public Student() {
        // Constructor
    }

    // Create student account
```

```
    public void createAccount() {
        try {
            BufferedWriter writer = new BufferedWriter(new FileWriter(studentDetailsFile));
            Scanner scanner = new Scanner(System.in);
            System.out.print("Enter your name: ");
            name = scanner.nextLine();
            System.out.print("Enter your email: ");
            email = scanner.nextLine();
            writer.write(name + "," + email);
            writer.close();
            System.out.println("Student account created successfully.");
        } catch (IOException e) {
            e.printStackTrace();
        }
    }

    // Student login
    public boolean login(String name, String email) {
        try {
            BufferedReader reader = new BufferedReader(new FileReader(studentDetailsFile));
            String line;
            while ((line = reader.readLine()) != null) {
                String[] parts = line.split(",");
                if (parts[0].equals(name) && parts[1].equals(email)) {
                    System.out.println("Login successful as Student.");
                    return true;
                }
            }
            reader.close();
        } catch (IOException e) {
            e.printStackTrace();
        }
        System.out.println("Invalid credentials. Please try again.");
        return false;
    }
}
```