SECJ 2154 Object-Oriented Programming

SEM II 2024/2025

Mini Project Title:

Book Recommendation System

LECTURER : DR. ZURAINI BINTI ALI SHAH

SECTION : 04

GROUP NO. : 04

GROUP MEMBERS :

NAME	MATRIC NUMBER
MERVYN BEE ZHENG CHENG	A23CS0245
AMOS KEAGAN HOSEA	A21EC0161
NIK MUHAMMAD DANIEL HAKIM BIN KAMALRUZAMAN	B24CS0034
NIK BALQIS BINTI NIK JEFRY	A23CS0256

TABLE OF CONTENTS

1.0 Introduction

In today's digital age, the abundance of books available online and offline has made it increasingly difficult for readers to choose what to read next. Recommendation systems have emerged as a solution to this problem by helping users discover books that align with their interests and preferences. Our project aims to design and implement a **Book Recommendation System** using Java that provides personalized suggestions to users based on various factors such as genre preferences, user ratings, and reading history.

The system will serve as a simple yet effective tool that mimics real-world recommendation engines used by online platforms like Amazon, Goodreads, and Google Books. Through this project, we aim to apply core object-oriented programming concepts, data handling, and algorithmic thinking to solve a real-world problem.

2.0 Problem Statement

With thousands of books published each year and countless options across multiple genres, readers often face difficulty in selecting books that match their interests. This issue is especially challenging for new readers or those seeking to explore new genres. Traditional methods of book discovery such as browsing by category or relying on top charts can be inefficient and impersonal.

There is a need for an intelligent system that can recommend books tailored to individual user preferences. Without such a system, users may spend significant time searching for books, potentially leading to frustration and reduced interest in reading. Therefore, our goal is to develop a Java-based Book Recommendation System that provides efficient, personalized book suggestions, enhancing the overall user reading experience.

3.0 Objectives

The aim of our mini project are to:

- Create a functional book recommendation system using Java.
- Apply object-oriented principles such as inheritance, polymorphism, interfaces, and file handling.
- Personalize book suggestions for users based on input genre or field.
- Simulate real-world systems in a simplified form with reusable and scalable code.
- Handle user registration, login, and display of book recommendations effectively.

4.0 Project Design

4.1 Overview

The system consists of multiple interrelated Java classes, including abstract classes, inheritance hierarchies, interface implementations, and external data file interactions.

Concept	Description
Abstraction	Implemented via the *Book* abstract class.
Inheritance	*Fiction* and *NonFiction* classes inherit from *Book.*
Polymorphism	Used in book list handling and display methods.
Interface	*Displayable* interface ensures consistent method implementation.
Exception Handling	Applied in file I/O operations.
Association	*User* class stores a list of *Book* objects as recommendations.

Table 1: OOP Concepts Applied

4.2 Class Diagram

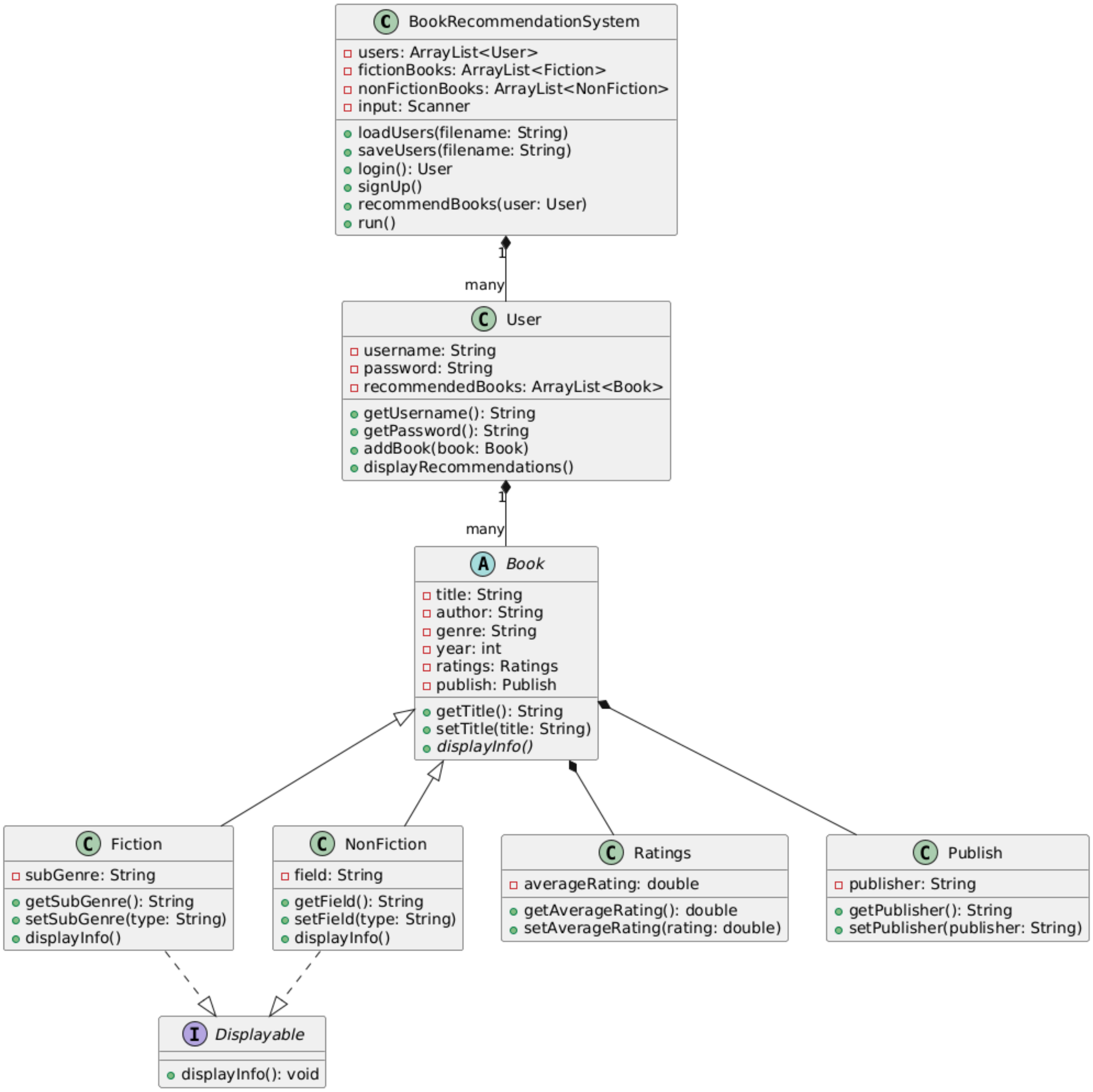


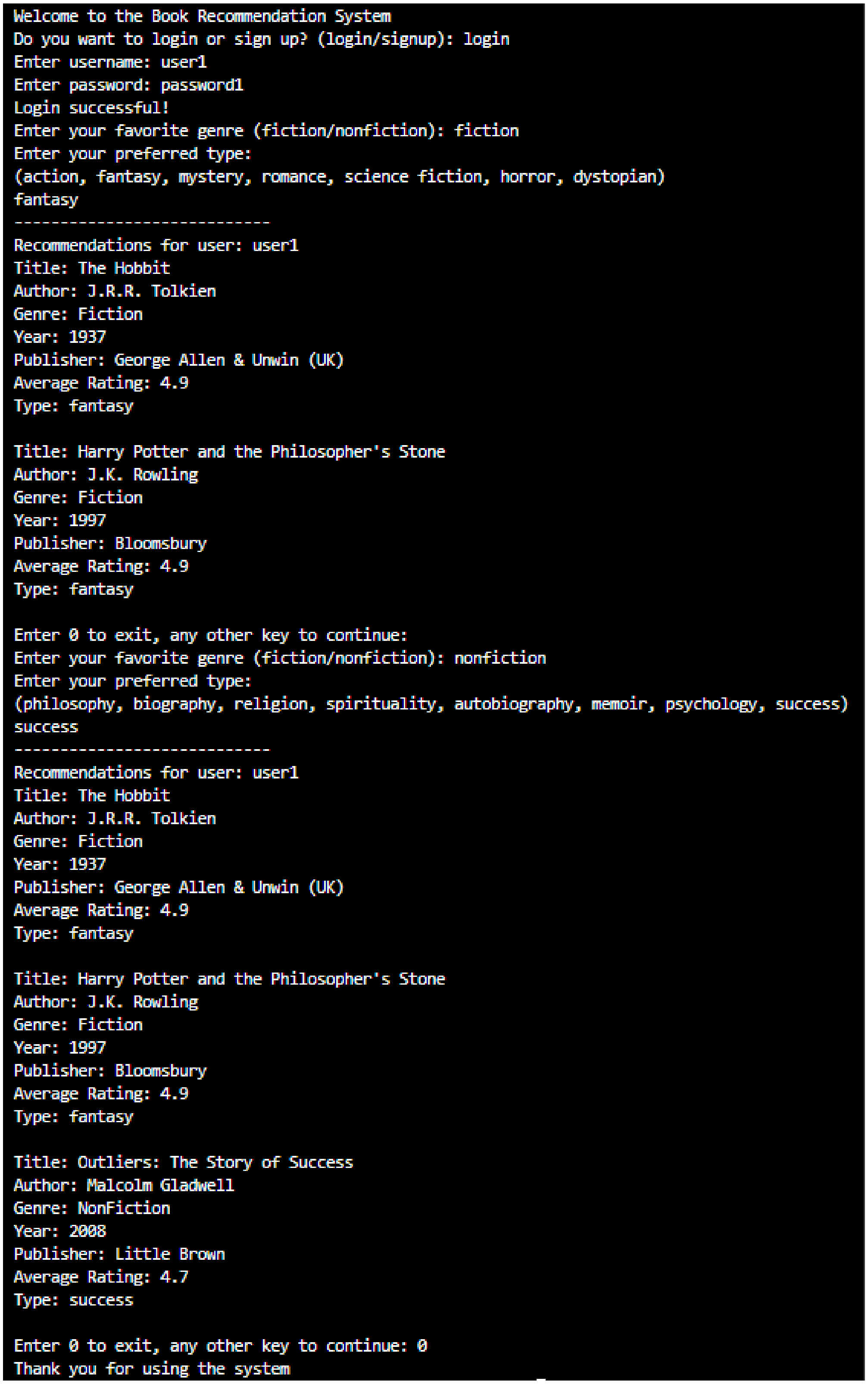
Diagram 1: Class Diagram

5.0 Project Output

5.1 How It Works

1. Users either sign up or log in.
2. System reads data from *users.txt*, *fiction.txt*, and *nonfiction.txt*.
3. Based on the user's selected genre and preference type, appropriate books are filtered and displayed.
4. Users may continue or exit the system.

5.2 Expected Output (Sample)



```
Welcome to the Book Recommendation System
Do you want to login or sign up? (login/signup): login
Enter username: user1
Enter password: password1
Login successful!
Enter your favorite genre (fiction/nonfiction): fiction
Enter your preferred type:
(action, fantasy, mystery, romance, science fiction, horror, dystopian)
fantasy
----------------------------
Recommendations for user: user1
Title: The Hobbit
Author: J.R.R. Tolkien
Genre: Fiction
Year: 1937
Publisher: George Allen & Unwin (UK)
Average Rating: 4.9
Type: fantasy

Title: Harry Potter and the Philosopher's Stone
Author: J.K. Rowling
Genre: Fiction
Year: 1997
Publisher: Bloomsbury
Average Rating: 4.9
Type: fantasy

Enter 0 to exit, any other key to continue:
Enter your favorite genre (fiction/nonfiction): nonfiction
Enter your preferred type:
(philosophy, biography, religion, spirituality, autobiography, memoir, psychology, success)
success
----------------------------
Recommendations for user: user1
Title: The Hobbit
Author: J.R.R. Tolkien
Genre: Fiction
Year: 1937
Publisher: George Allen & Unwin (UK)
Average Rating: 4.9
Type: fantasy

Title: Harry Potter and the Philosopher's Stone
Author: J.K. Rowling
Genre: Fiction
Year: 1997
Publisher: Bloomsbury
Average Rating: 4.9
Type: fantasy

Title: Outliers: The Story of Success
Author: Malcolm Gladwell
Genre: NonFiction
Year: 2008
Publisher: Little Brown
Average Rating: 4.7
Type: success

Enter 0 to exit, any other key to continue: 0
Thank you for using the system
```

Diagram 2: Sample Output

6.0 Conclusion

The Book Recommendation System successfully fulfills its objective of providing personalized book suggestions based on user preferences. The system demonstrates how these principles can be used to solve real-world problems effectively by applying core object-oriented programming concepts such as inheritance, abstraction, polymorphism, and interface implementation. The use of file handling to read and write user and book data, along with exception handling to manage potential input/output errors, further contributes to the functionality of the system.

Throughout the development of this project, our team gained valuable experience in designing modular class structures, managing user interaction through a command-line interface, and implementing logic to filter and recommend books based on genre-specific input. The book recommendation system, although simple, reflects the foundational structure of more complex engines used in real-world applications.

Moving forward, several enhancements could be implemented to improve the system. These include developing a graphical user interface (GUI) to improve usability, adding search filters such as author, year, or rating to refine results, and integrating features like user reviews or comments. Additionally, we could make the recommendations even more personalized and dynamic by expanding the database to include a wider variety of book categories or formats and potentially incorporating machine learning algorithms.

7.0 Appendices

7.1 Java Program

Source Files	
Book.java	Abstract base class for all books
Fiction.java	Fiction subclass with subGenre
NonFiction.java	Nonfiction subclass with field
User.java	Stores user details and recommended books
Ratings.java	Encapsulates book ratings
Publish.java	Encapsulates book publisher info
Displayable.java	Interface for displaying book info
BookRecommendationSystem.java	Main logic controller
Data Files	
fiction.txt	Fiction book data (sample: The Hobbit, 1984, Dune, etc.)
nonfiction.txt	Nonfiction book data (sample: Sapiens, Educated, Outliers, etc.)
users.txt	Stores user credentials (sample: user1,password1, mervyn,mervyn123)

Table 2: JAVA Program Files

7.2 Sample Code Snippet

From BookRecommendationSystem.java:

Java

```
       public static void main(String[] args) {
               BookRecommendationSystem system = new BookRecommendationSystem();
               system.run();
           }
```

*Other java files are attached in zip folder.