UNIVERSITI TEKNOLOGI MALAYSIA

SECJ2154 OBJECT ORIENTED PROGRAMMING
SECTION 02

REPORT (LIBRARY MANAGEMENT SYSTEM)

LECTURER: ASSOC. PROF. DR. MOHD. YAZID BIN IDRIS

GROUP NAME: LogiCode

NAME	MATRIC NO.
KOH SU XUAN	A22EC0060
ONG YI YAN	A22EC0101
TANG YAN QING	A22EC0109

1.0 Project Summary

The LogiCode Library Management System is an application designed to streamline the operations of the LogiCode Library by managing the physical books as well as e-books within the library environment. It will provide functionalities for users such as librarians, borrowers (including lecturers and students) to perform tasks of managing library resources such as borrowing and returning books, keeping track of user borrowing records, searching for books and managing fines of the borrowers who are late on returning the books. The system will be developed using the Java programming language, aiming for simplicity, efficiency and ease of use by the library. Overall, the system enhances the efficiency of library operations and improves the user experience by providing functionalities such as user authentication, book management, borrowing and returning books, fine tracking and searching for books based on various criteria such as the book title and book id.

2.0 Project Background

The LogiCode Library is an academic library that offers a large collection of physical books and e-books to students and lecturers from different faculties. However, students and lecturers always face challenges in the manual search process for locating desired physical books and e-books within the library's vast collection. Moreover, librarians rely on manual record-keeping and fine calculation methods, which are time-consuming, error-prone and inefficient.

As a result, our project addresses this need by developing a modern and user-friendly LogiCode Library Management System that uses technology to manage both physical and electronic resources within the library. Using the Java programming language, we aim to create a stable and scalable system capable of handling the complexities of library management with ease. We intend to deliver a solution that meets the needs of both librarians and borrowers by including features like user authentication, book management, borrowing history tracking and fine management.

Overall, the LogiCode Library Management System project demonstrates dedication to using technology to improve the efficiency and usability of library services, ultimately advancing the academic community.

3.0 Scope of Works

To effectively develop the Library Management System within the given timeframe, the project will be divided into the following scopes, with each group member responsible for specific tasks:

Group Member	Scope of Works
Ong Yi Yan	1. User class 2. BookSearching class 3. BookFineTransaction class
Tang Yan Qing	1. Librarian class 2. BorrowingRecord class 3. Data class
Koh Su Xuan	1. Borrower class 2. Book class 3. eBook class
All	1. Library class

4.0 List of Classes Exist in Each Scope

In LogiCode Library Management System, there are 10 classes with each group member owning 3 classes of the Library Management System and sharing a main class, which are User, Librarian, Borrower, Book, eBook, Data, BorrowingRecord, BookSearching, BookFineTransaction and Library. Each class is listed out along with their descriptions.

No.	Class	Attribute	Description
1.	User	- username - password - email	User class is the superclass of Librarian and Borrower classes with the concept of inheritance. This class holds the common attributes and methods that are shared by librarians and borrowers. It also contains the method for the users to login into the system.
2.	Librarian	- librarianId	Librarian class is the subclass of User class with the concept of inheritance. It manages the information specific to librarians.
3.	Borrower	- borrowerId - role - paidFine - borrowerBorrowRecord	Borrower class is the subclass of User class with the concept of inheritance. It is also associated with the BorrowingRecord and BookFineTransaction class through the concept of composition. The class is responsible for handling the information related to borrowers such as the borrowing records and fine transactions. It also provides methods to calculate fines for overdue books, display borrower details and add new borrowing records and transactions.
4.	Book	- bookId - title - author	Book class is the superclass of eBook class with the concept of inheritance. It is also associated with the

		- genre - availability - bookBorrowingRecord	BorrowingRecord class through the concept of aggregation. The class maintains a vector of BorrowingRecord objects to track the borrowing history of each book. This allows the library system to keep detailed records of when a book was borrowed and its current status.
5.	eBook	- link	eBook class is the subclass of Book class with the concept of inheritance It inherits all attributes of the Book class and adds an additional attribute, “link” which stores the download link for the eBook.
6.	Data	-	Data class is used to handle file operations related to the reading and filtering of users and books. It provides methods to read users data and books data from files and load into a vector of User objects and Book objects respectively. The class also includes functionality to filter out Borrower objects from a list of User objects.
7.	BorrowingRecord	- recordId - borrowDate - returnDate - book	BorrowingRecord class is used to manage and track the borrowing details of books by storing information about each borrowing transaction. It also provides methods to calculate fines for overdue books.
8.	BookSearching	- choice - searchKey - bookList - searchResults	BookSearching class is used to search for certain books using certain criterias such as the book title and the book id in the library. It is linked to the Book class through a one-directional association relationship from BookSearching to Book class.
9.	BookFineTransaction	- transactionID - amount - payDate	BookFineTransaction class is used by the librarians to handle the records of fines imposed on borrowers for overdue books.

10.	Library	-	Library class acts as a main class for the whole library system. It serves as a guide for the librarians and borrowers to perform various operations in the system through the menus. For the librarian, they can register the borrowers in the system, adding physical books and eBooks to the library inventory, recording borrowings and returns of books and managing fine transactions. While for the borrowers, they can search for desired books, view borrowed books and review their fine transactions history and view their personal information.

5.0 Class Diagram

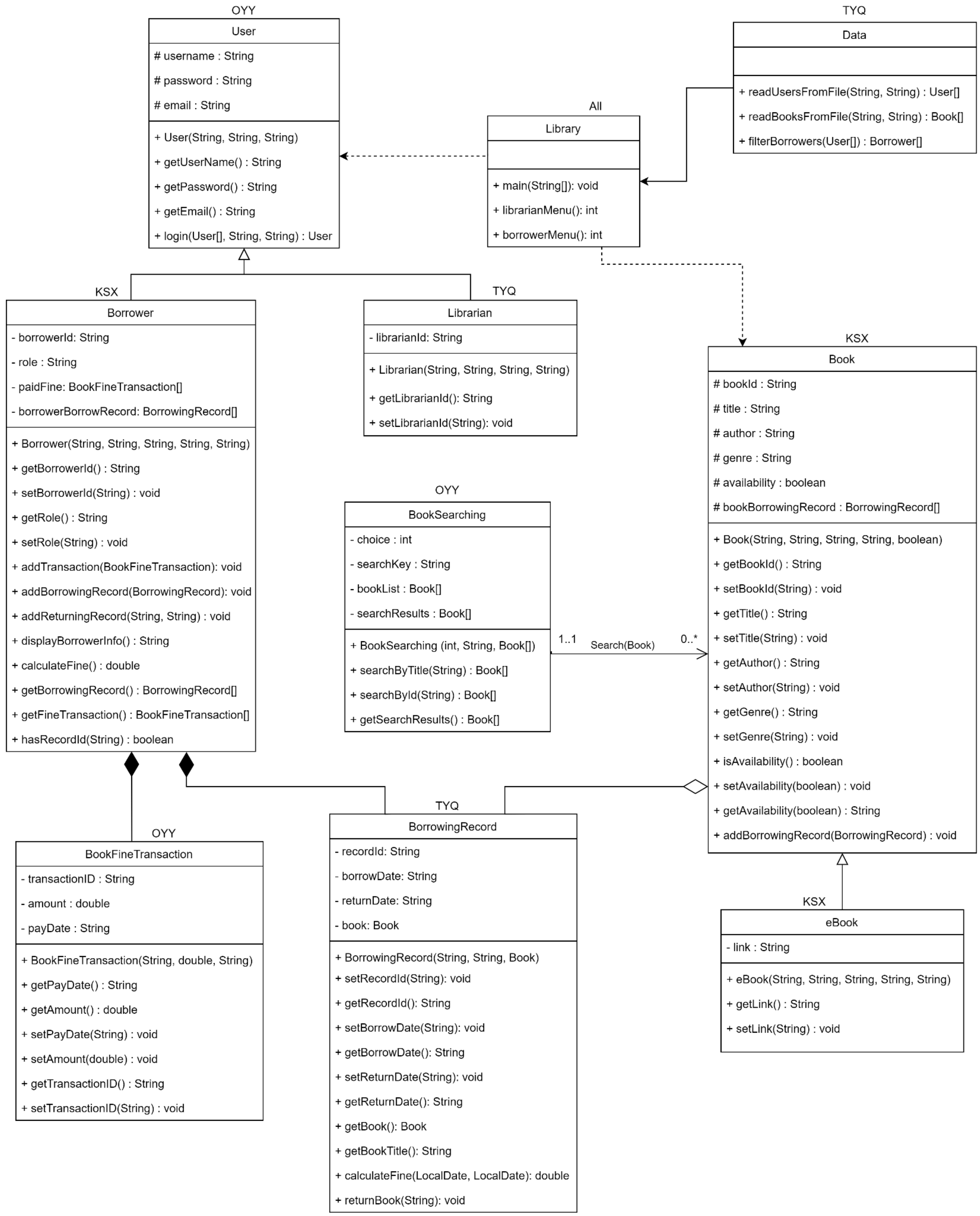

OYY
User
# username : String
# password : String
# email : String
+ User(String, String, String)
+ getUserName() : String
+ getPassword() : String
+ getEmail() : String
+ login(User[], String, String) : User
TYQ
Data
+ readUsersFromFile(String, String) : User[]
+ readBooksFromFile(String, String) : Book[]
+ filterBorrowers(User[]) : Borrower[]
All
Library
+ main(String[]): void
+ librarianMenu(): int
+ borrowerMenu(): int
KSX
Borrower
- borrowerId: String
- role : String
- paidFine: BookFineTransaction[]
- borrowerBorrowRecord: BorrowingRecord[]
+ Borrower(String, String, String, String, String)
+ getBorrowerId() : String
+ setBorrowerId(String) : void
+ getRole() : String
+ setRole(String) : void
+ addTransaction(BookFineTransaction): void
+ addBorrowingRecord(BorrowingRecord): void
+ addReturningRecord(String, String) : void
+ displayBorrowerInfo() : String
+ calculateFine() : double
+ getBorrowingRecord() : BorrowingRecord[]
+ getFineTransaction() : BookFineTransaction[]
+ hasRecordId(String) : boolean
TYQ
Librarian
- librarianId: String
+ Librarian(String, String, String, String)
+ getLibrarianId(): String
+ setLibrarianId(String): void
KSX
Book
# bookId : String
# title : String
# author : String
# genre : String
# availability : boolean
# bookBorrowingRecord : BorrowingRecord[]
+ Book(String, String, String, String, boolean)
+ getBookId() : String
+ setBookId(String) : void
+ getTitle() : String
+ setTitle(String) : void
+ getAuthor() : String
+ setAuthor(String) : void
+ getGenre() : String
+ setGenre(String) : void
+ isAvailability() : boolean
+ setAvailability(boolean) : void
+ getAvailability(boolean) : String
+ addBorrowingRecord(BorrowingRecord) : void
OYY
BookSearching
- choice : int
- searchKey : String
- bookList : Book[]
- searchResults : Book[]
+ BookSearching (int, String, Book[])
+ searchByTitle(String) : Book[]
+ searchById(String) : Book[]
+ getSearchResults() : Book[]
1..1
Search(Book)
0..*
OYY
BookFineTransaction
- transactionID : String
- amount : double
- payDate : String
+ BookFineTransaction(String, double, String)
+ getPayDate() : String
+ getAmount() : double
+ setPayDate(String) : void
+ setAmount(double) : void
+ getTransactionID() : String
+ setTransactionID(String) : void
TYQ
BorrowingRecord
- recordId: String
- borrowDate: String
- returnDate: String
- book: Book
+ BorrowingRecord(String, String, Book)
+ setRecordId(String): void
+ getRecordId(): String
+ setBorrowDate(String): void
+ getBorrowDate(): String
+ setReturnDate(String): void
+ getReturnDate(): String
+ getBook(): Book
+ getBookTitle(): String
+ calculateFine(LocalDate, LocalDate): double
+ returnBook(String): void
KSX
eBook
- link : String
+ eBook(String, String, String, String, String)
+ getLink() : String
+ setLink(String) : void

Figure 5.1: Class Diagram of LogiCode Library Management System

6.0 Output Screen Capture

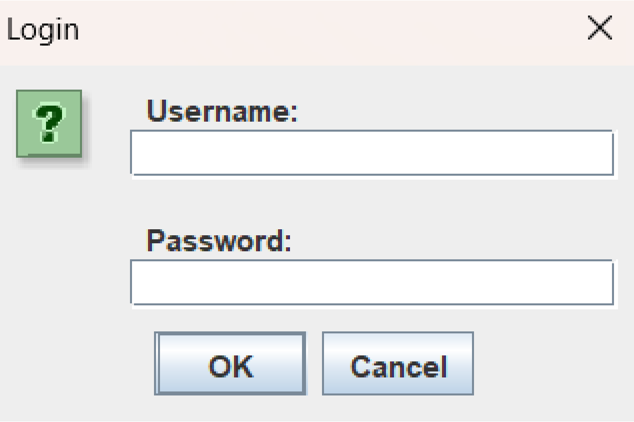
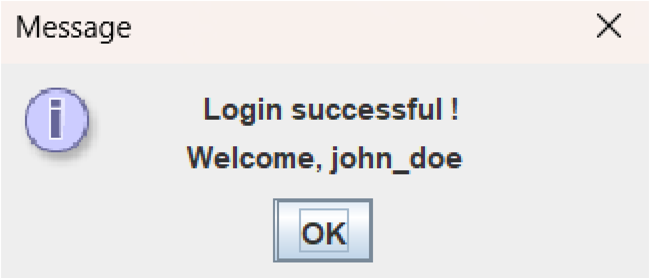
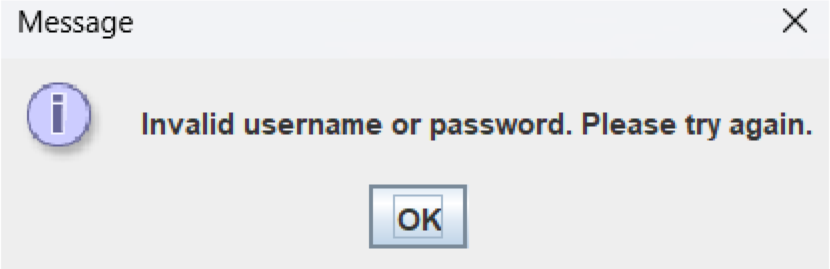


Figure 6.1: Login



Login successful !
Welcome, john_doe

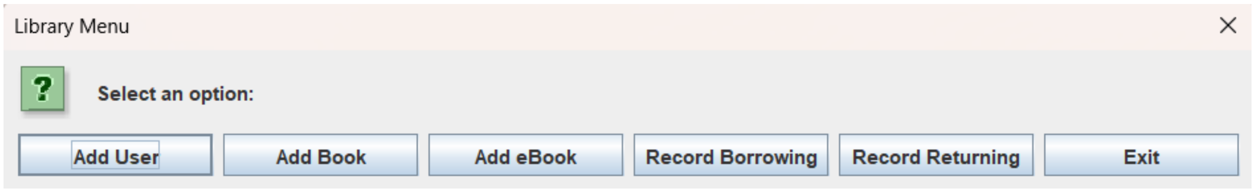
Figure 6.2: Login Successful



Invalid username or password. Please try again.

Figure 6.3: Login Unsuccessful

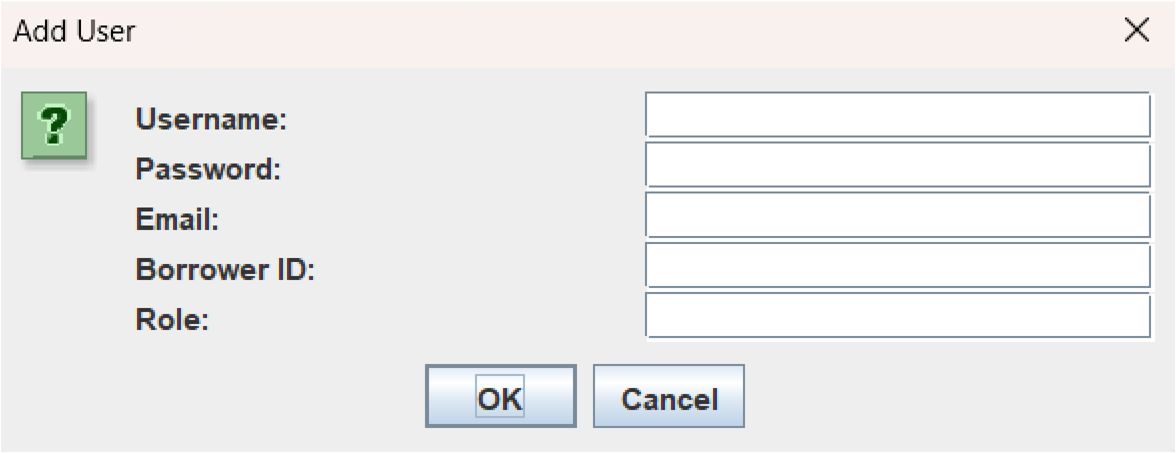
Librarian Menu



Library Menu

Select an option:

Figure 6.4: Menu for Librarian



Add User

Username:

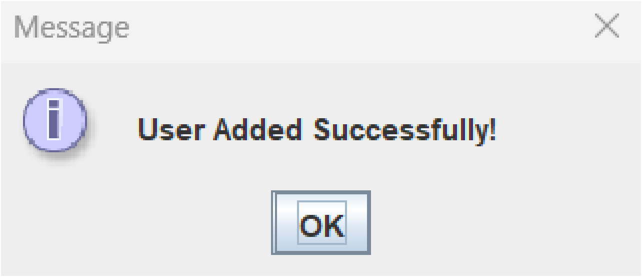
Password:

Email:

Borrower ID:

Role:

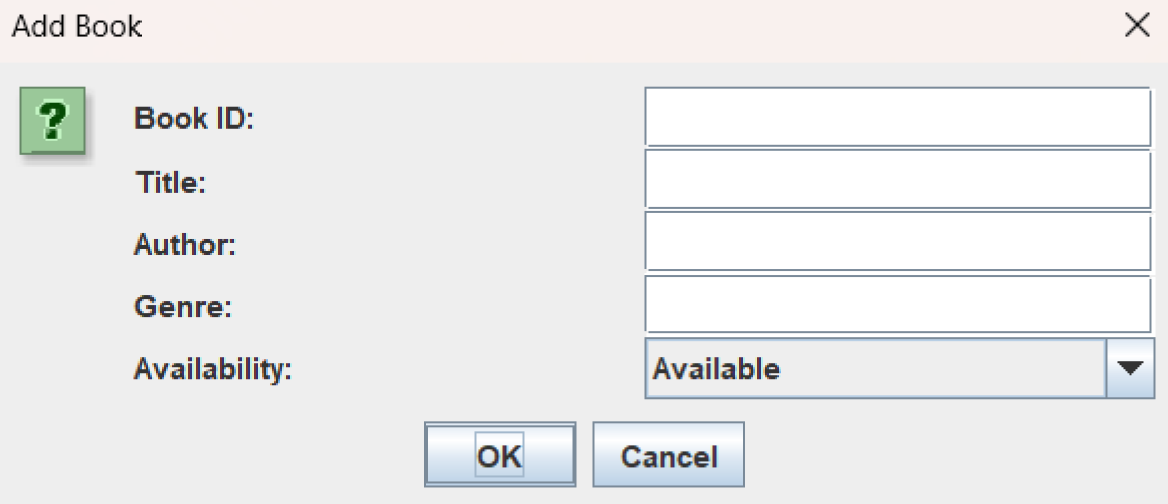
Figure 6.5: Function 1 Add User



Message

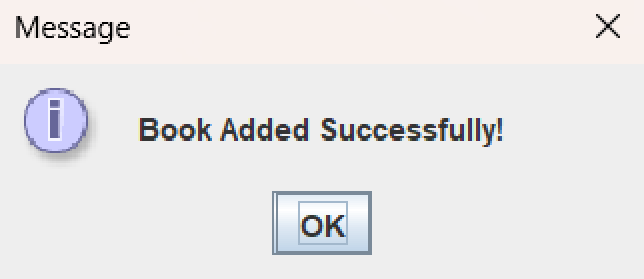
User Added Successfully!

Figure 6.6: User Addition Successful



Availability: Available

Figure 6.7: Function 2 Add Book



Book Added Successfully!

Figure 6.8: Book Addition Successful

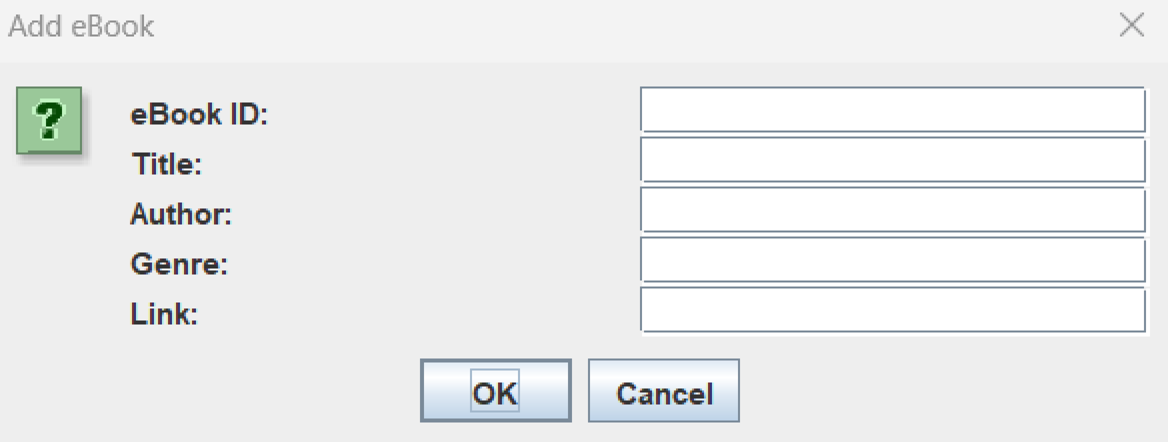
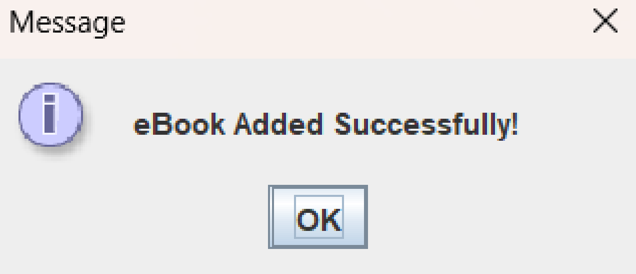
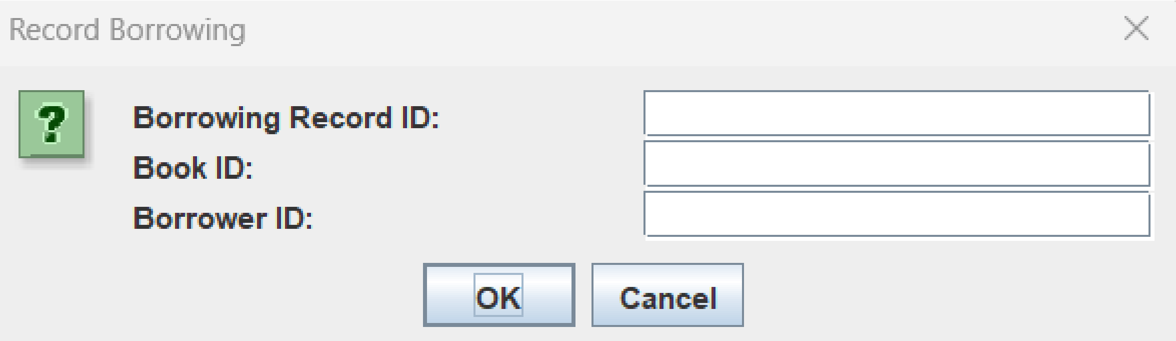


Figure 6.9: Function 3 Add eBook



eBook Added Successfully!

Figure 6.10: eBook Addition Successful

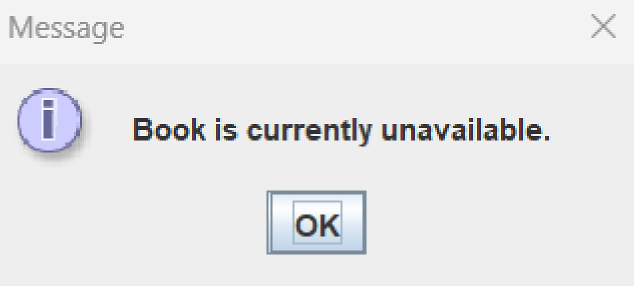


Borrowing Record ID:

Book ID:

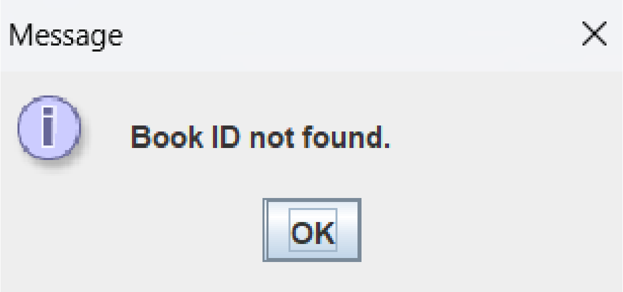
Borrower ID:

Figure 6.11: Function 4 Record Borrowing



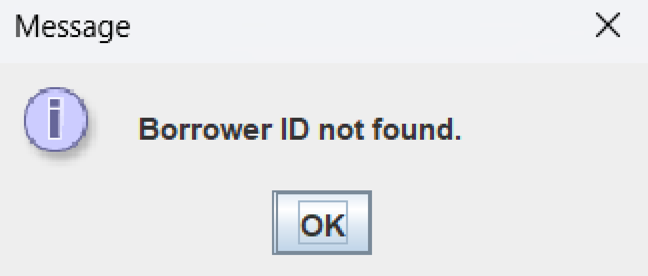
Book is currently unavailable.

Figure 6.12: Borrowing Record Unsuccessful due to Book Unavailability



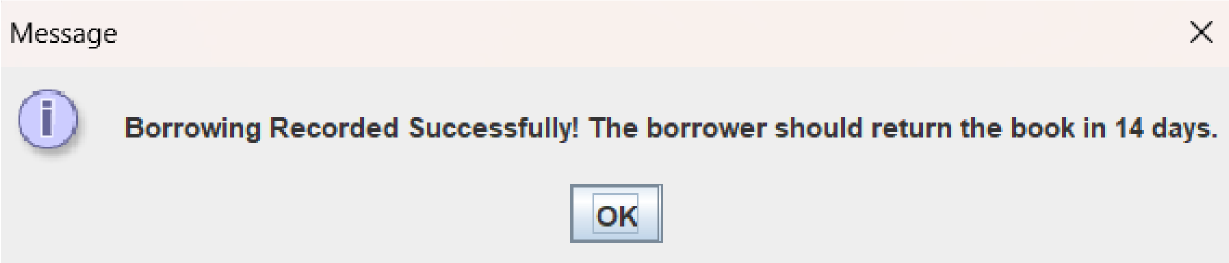
Book ID not found.

Figure 6.13: Borrowing Record Unsuccessful due to Invalid Book ID



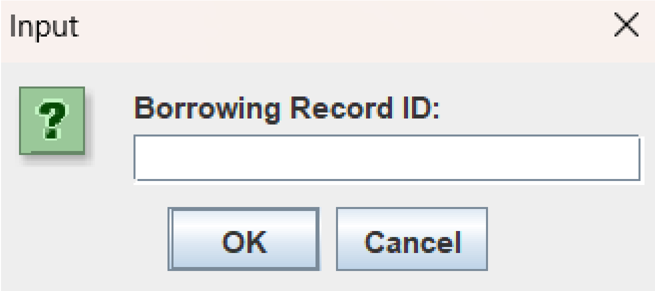
Borrower ID not found.

Figure 6.14: Borrowing Record Unsuccessful due to Invalid Borrower ID



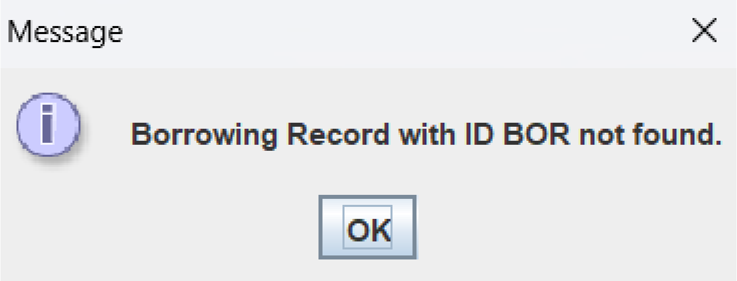
Borrowing Recorded Successfully! The borrower should return the book in 14 days.

Figure 6.15: Borrowing Record Successful



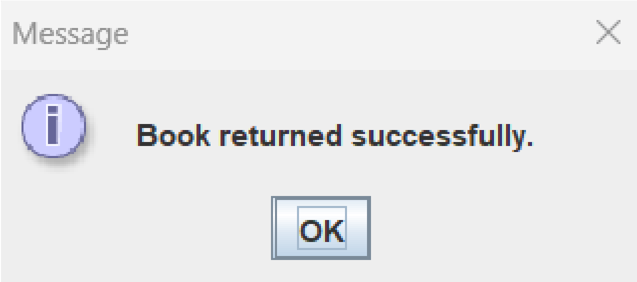
Borrowing Record ID:

Figure 6.16: Function 4 Return Borrowing



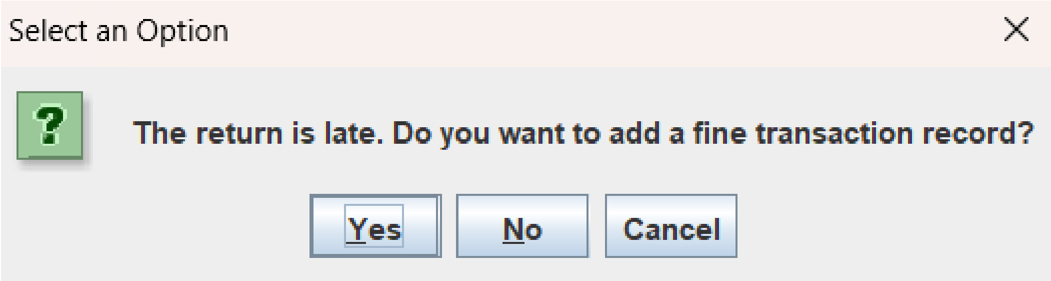
Borrowing Record with ID BOR not found.

Figure 6.17: Borrowing Return Unsuccessful due to Invalid Borrowing Record ID



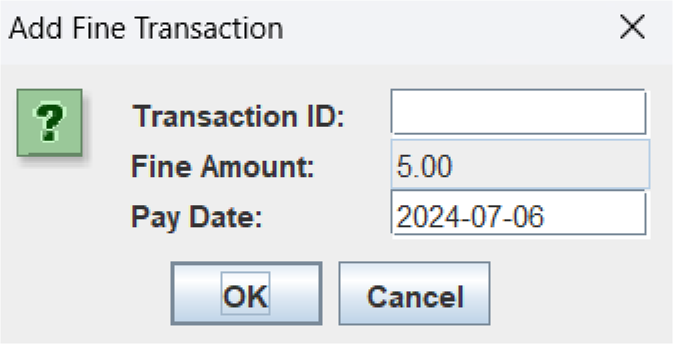
Book returned successfully.

Figure 6.18: Borrowing Return Successful



The return is late. Do you want to add a fine transaction record?

Figure 6.19: Prompt to Record Fine Transaction for Late Return

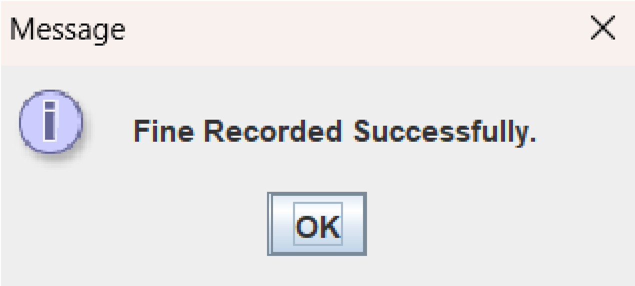


Transaction ID:

Fine Amount: 5.00

Pay Date: 2024-07-06

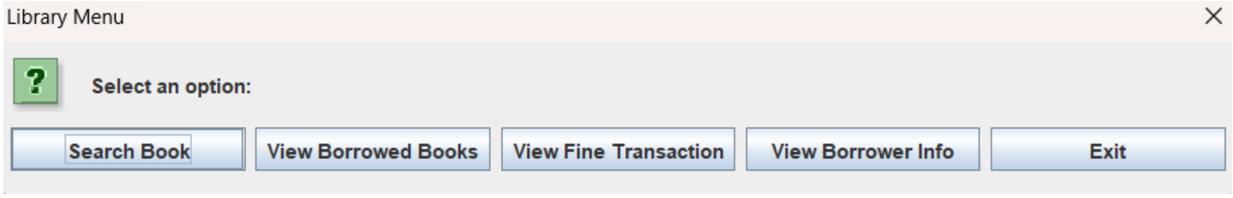
Figure 6.20: Add Fine Transaction



Fine Recorded Successfully.

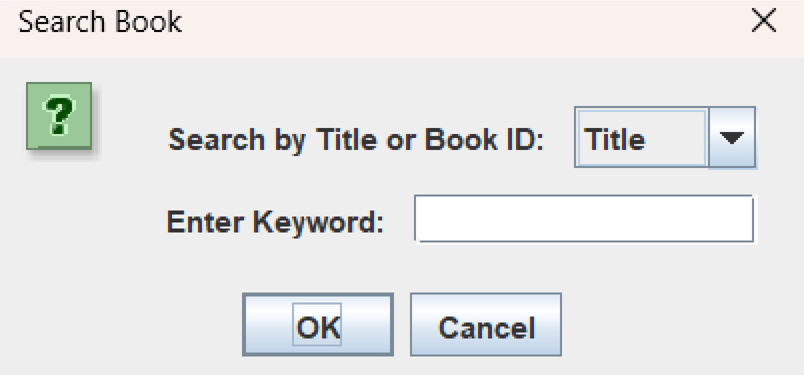
Figure 6.21: Fine Transaction Addition Successful

Borrower Menu


Library Menu

Select an option:

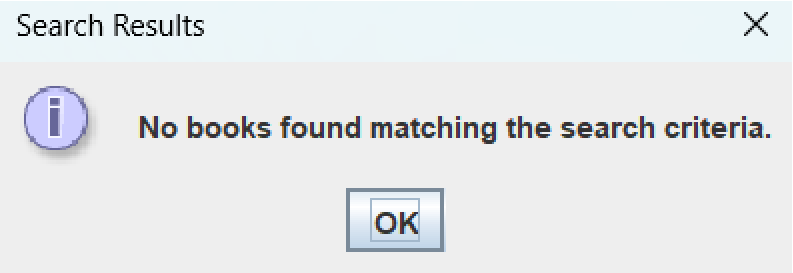
Figure 6.22: Menu for Borrower


Search Book

Search by Title or Book ID: Title

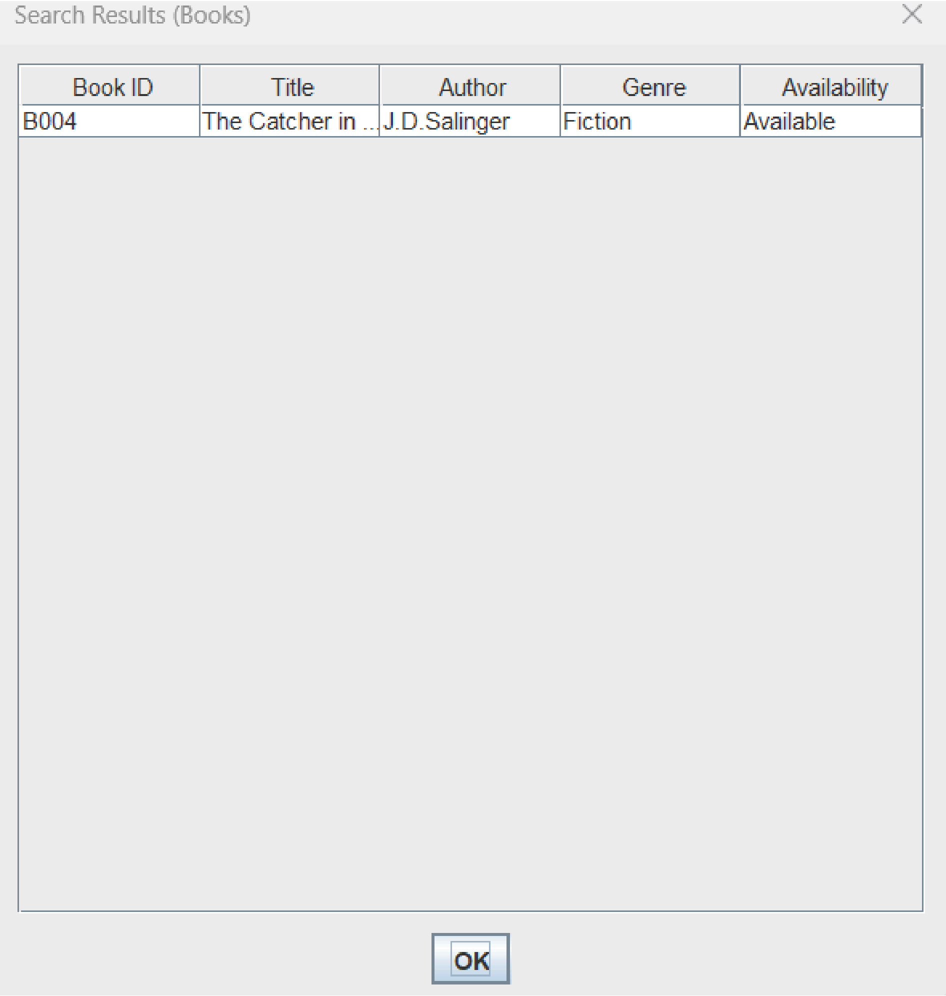
Enter Keyword:

Figure 6.23: Function 1 Search Book


Search Results

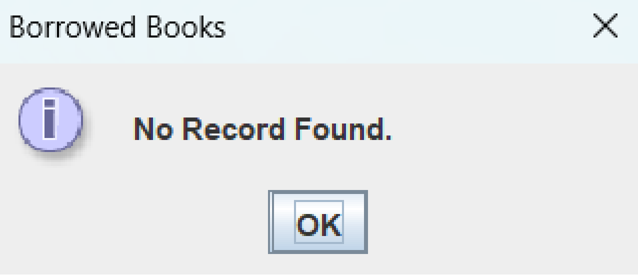
No books found matching the search criteria.

Figure 6.24: Book Search Results - No Matches Found



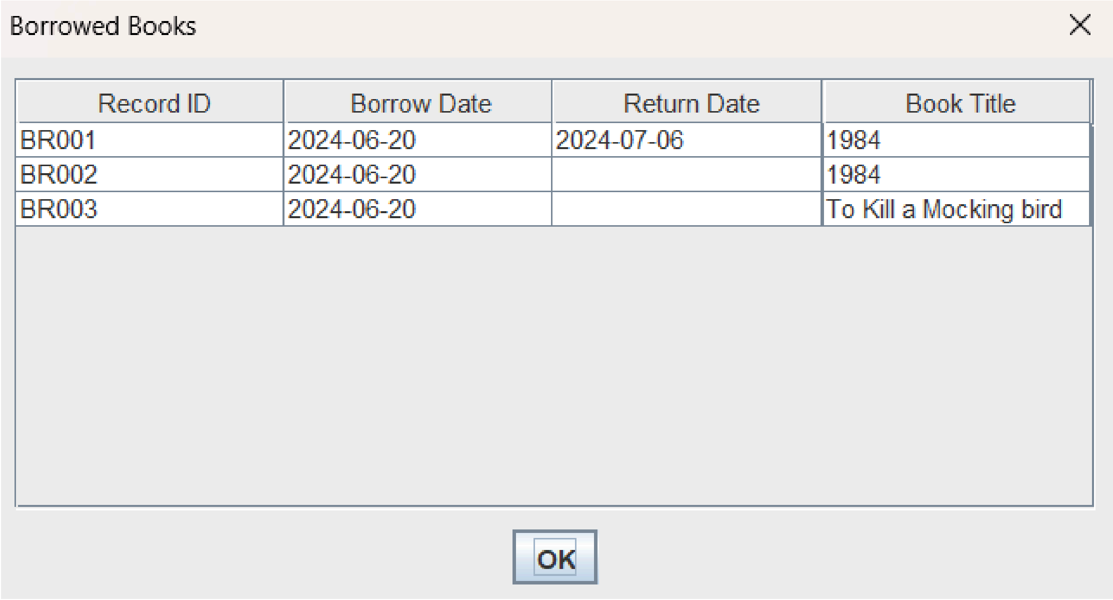
Book ID	Title	Author	Genre	Availability
B004	The Catcher in ...	J.D.Salinger	Fiction	Available

Figure 6.25: Book Search Results - Results Displayed



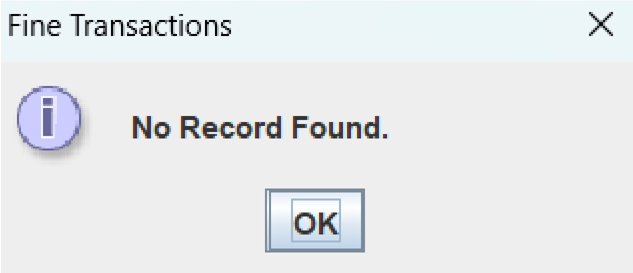
No Record Found.

Figure 6.26: Function 2 View Borrowed Books - No Record



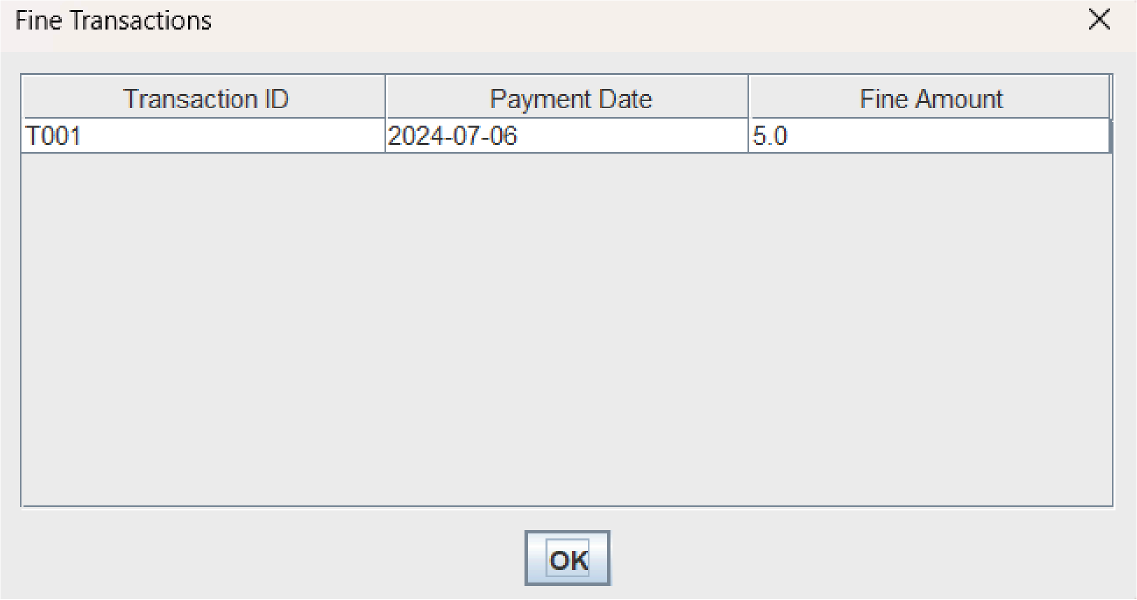
Record ID	Borrow Date	Return Date	Book Title
BR001	2024-06-20	2024-07-06	1984
BR002	2024-06-20		1984
BR003	2024-06-20		To Kill a Mocking bird

Figure 6.27: Function 2 View Borrowed Books - Records Displayed



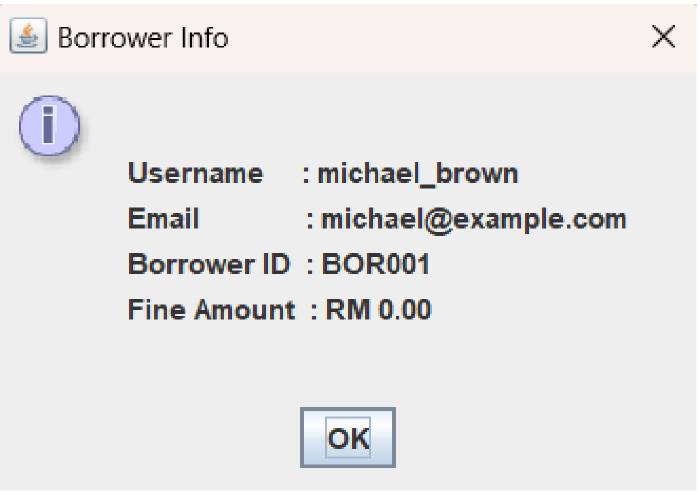
No Record Found.

Figure 6.28: Function 3 View Fine Transaction - No Record



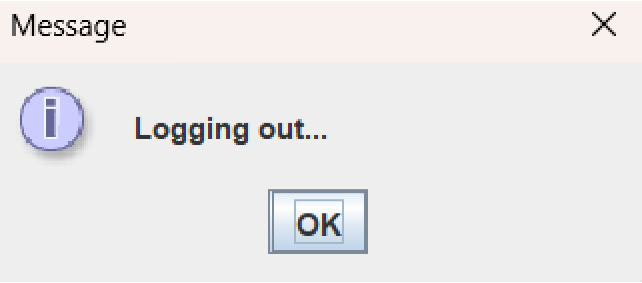
Transaction ID	Payment Date	Fine Amount
T001	2024-07-06	5.0

Figure 6.29: Function 3 View Fine Transaction - Records Displayed



Username : michael_brown
Email : michael@example.com
Borrower ID : BOR001
Fine Amount : RM 0.00

Figure 6.30: Function 4 View Borrower Info



Logging out...

Figure 6.31: Exit