SYNCFUSION LIBRARY

Application For Online Library Management

Need to prepare a console application for **Online Library Management and Book tracking** for the **“Syncfusion college”.**

The application should allow to do the below process.

Main Menu:

The Main menu must have the following options.

1. **UserRegistration**
2. **UserLogin**
3. **Exit**

If the user selects Option **3. Exit,** then the application should be closed.

1. UserRegistration:

On selection of the option “**1. User Registration**” from the main menu,

1. Need to get the below details from the user for the user registration.

   a. Username
   b. Gender (Enum – Select, Male, Female, Transgender)
   c. Department
   d. MobileNumber
   e. MailID
   f. WalletBalance

2. On completion of the user registration, we need to display the **UserID** (auto generated UserID).
3. Then need to show the **Main menu** again.

2. UserLogin

On selection of the Choice 2 in Main menu “**User Login**” ,

1. Ask for the “**UserID**” of the user. Check the “UserID” in the user list.
2. If the “UserID” does not exist , then show “**Invalid User ID. Please enter a valid one**”.

3. If “UserID” exists, then show the below menu.
    1. **Borrowbook.**
    2. **ShowBorrowedhistory.**
    3. **ReturnBooks**
    4. **WalletRecharge**
    5. **Exit**

- **BorrowBook**

    1. Show the list of Books available by printing **BookID**, **BookName**, **AuthorName**, **BookCount**.
    2. Then Ask the user to pick one book by Asking “**Enter Book ID to borrow**”.
    3. Check whether “**BookID**” is available or not.
    4. If not available display “ **Invalid Book ID, Please enter valid ID**”, else ask the user to “**Enter the count of the book**”. Then,
    5. Check the book count availability of the book selected.
        - If there is no book available, display as “**Books are not available for the selected count”**.
        - And print the next available date of book by getting that book’s “**BorrowedDate**” from the borrowed history information and adding with 15 days the borrowed date of that book.
        - Show **“The book will be available on {**borrowed date + 15 days**}”.**
    6. If the book is available to borrow,
        - need to check whether the user already have any borrowed book. Because user can have a maximum of only 3 borrowed books at a time.
        - If user having 3 borrowed books already then show “**You have borrowed 3 books already”.**
        - If the user’s already borrowed book count and requested book count exceeds 3 count, then show “**You can have maximum of 3 borrowed books. Your already borrowed books count is {BorrowBookCount} and requested count is {Current Requested Count}, which exceeds 3 ”.**
    7. Else allocate the book to the user and set status of the Booking Details as "**Borrowed**” and set the “BorrowedDate” as today’s date and “PaidFineAmount” as 0.
        - Create the Borrow Details object then store it and show **“Book Borrowed successfully**”.

- **ShowBorrowedHistory**

    Show the **Book borrowed details** of the current user.

- **ReturnBooks**

    1. Show the borrowed book details of current user whose status is “**borrowed**” also Print the return date of each book (**Return date will be 15 days after borrowing a book**).
    2. If the return date is elapsed more than 15 days then calculate and show the **fine amount** (**Rs. 1 / Day)** for each book.
    3. Ask him to select the **BorrowedID** to return book and validate that ID.
    4. If return date is elapsed,
        a. then check whether the user have sufficient balance for the fine amount,
        b. if he has sufficient balance then deduct the fine amount from his Wallet balance and change the Status in Booking History to “**Returned**” and update the fine amount to the “**PaidFineAmount**” calculated and show “**Book returned successfully”.** Also, update the “BookCount”.
        c. else show “**Insufficient balance. Please rechange and proceed**”.
    5. Else, change the Status in Booking History to “**Returned**” and show **Book returned successfully.** Also, update the “BookCount”.

- **WalletRecharge**
    1. Ask the customer whether he wish to recharge the wallet.
    2. If “Yes” then ask for the amount to be recharged and update the amount in the wallet and display the updated wallet balance.
    3. Else go the submenu.

- **Exit**

    Exit submenu and go to the main menu.

User Details Class:

User Details class should have below properties.

Properties:

a. UserID (Auto Increment – SF3000)
b. UserName
c. Gender
d. Department – (Enum – ECE, EEE, CSE)
e. MobileNumber
f. MailID

g. WalletBalance

Methods:

a. **WalletRecharge:** Get the recharge amount through parameters and update wallet balance.
b. **DeductBalance:** Get the deducted amount through parameters and update wallet balance.

BookDetails Class:

Book Details class should have below properties.

Properties:

1. BookID (Auto Increment - BID1000)
2. BookName
3. AuthorName
4. BookCount

BorrowDetails Class:

BorrowDetails class should have below properties.

Properties:

- BorrowID (Auto Increment – LB2000)
- BookID
- UserID
- BorrowedDate – (Current Date and Time)
- BorrowBookCount
- Status – (Enum - Default, Borrowed, Returned)
- PaidFineAmount

Default Data:

Objects for UserDetails

UserID	**UserName**	**Gender**	**Department**	**MobileNumber**	**MailID**	**WalletBalance**
SF3001	Ravichandran	Male	EEE	9938388333	ravi@gmail.com	100
SF3002	Priyadharshini	Female	CSE	9944444455	priya@gmail.com	150

Objects for BookDetails

BookID	BookName	AuthorName	BookCount
BID1001	C#	Author1	3
BID1002	HTML	Author2	5
BID1003	CSS	Author1	3
BID1004	JS	Author1	3
BID1005	TS	Author2	2

Objects for BorrowDetails

BorrowID	BookID	UserID	BorrowedDate	BorrowBookCount	Status	PaidFineAmount
LB2001	BID1001	SF3001	10/09/2023	2	Borrowed	0
LB2002	BID1003	SF3001	12/09/2023	1	Borrowed	0
LB2003	BID1004	SF3001	14/09/2023	1	Returned	16
LB2004	BID1002	SF3002	11/09/2023	2	Borrowed	0
LB2005	BID1005	SF3002	09/09/2023	1	Returned	20