Mini Project One - Library System

Brief Design

Zhipeng Zhou, Sen Li, Li Huang, Run Yan

2014/8/12

# Requirements:

Mini Project - Books Library

-----------------------------------------

Develop a Java GUI application using Java Swing and Collections API to maintain a books library.

As this application does not use a permanent storage database, all your data would be stored in the form of collection objects at the run time and will be destroyed when you exit the application.

You can as well think of forming the basic data objects (java objects) for books, customers and transactions into permanent storage like files, so that every time the application loads, you can read these objects from files and load the respective object to populate the data in the application.

# System Features:

The system must have the following basic features:

(1) \*- Adding, Updating and Deleting of Books

(2) \*- Adding, Updating and Deleting of customers

(3) \*- Customer to view the list of books available and take it on rental basis. There is no rental fee charged but the book must be returned by due date. If not, a penalty amount per day must be paid. This renting of a book is called as transaction.

(4) \* - Customer should be able to view his current books taken from library and mark for returning

(5) \* - Admin to notify all the customers when a new book arrives to library

(6) \*- Different categories of books can be created so that each category can have many books

(7) \*- Customer can opt for notifications about new books arrivals category-wise

(8) \*- An image of book can as well be displayed when the customer is browsing the list of books available.

(9)\*- Admin to view the following reports:

- List of customers

- List of books/category

- List of books rented out

- List of books which are yet to be rented out

- List of books overdue from customers

# User Interface:

(1-1)Main Frame



## Administrator:

(2-1)Login



(2-2) Admin Interface



(2-3)Manage Books



If admin clicks “Add” button, the “Add Book” Frame will show.

(2-3-1)



(2-4)Manage Users



(2-5)View Categories



(2-6)View Rent Books

It is triggered by “Manage Books” frame.

- List of books rented out

- List of books which are yet to be rented out

- List of books overdue from customers



## Customer:

(3-1)Login



(3-2)Customer Interface



(3-3)View/Return rented books



Customer select a book to return from the “Rented Books” list. In the right part of the form, this book’s information will be shown.

Then customer clicks “Return” button if he or she wants to return the selected book. The “Return Book Finish” Dialog box will show.

**(3-3-1) Return Book Finish (Dialog)**



(3-4)Borrow books/View new books notifications



If this book is not “available”, then the “Borrow” button will be disable(Gray).

If Customer click “Borrow” button, then the “Book Borrow Message” Dialog will show.

**(3-4-1) Borrow Book Finish (Dialog)**



If book cannot be borrowed(error). This dialog below shows.



## Common Controls：

**(C-1) Book Info** (GroupBox):



We can make a common”Book Info” GroupBox or Panel, for commonly used in other frames。

Methods needed:

Load(Book book) - Load the book information from object, and fill the attributes in this panel.

Save(Book book) - Write back the information to the book object from the panel

“Status” attribute item could be invisible depends on needs.

# Use Cases:

integration tests must pass all these use cases below:

## Admin Use Cases

**(1-1)Book Management**

View list of books

View book information

View list of book categories

Adding, Updating and Deleting of Books

**(1-2)Customer Management**

View List of customers

Adding, Updating and Deleting of customers

**(1-3)View Books Rent Report**

View List of books rented out

View List of books which are yet to be rented out

View List of books overdue from customers

## Customers Use Cases

**(2-1)Borrow Book:**

View new arrival books.

View the list of books available.

View book information

Rent a book from the available book list

**(2-2)Return Book:**

View borrowed book list.

Return a book

# Files:

users.dat - Store all the users

books.dat - Store all the books

Book’s image files:

All the book image files put into the folder “bookimg/”

bookimg/12345.jpg

stands for the image of the book which has ISBN=12345

# Class Diagram:

Brief Design

**User (Customer)**

id

name

password

….

rentBook()

returnBook()

**Book**

isbn

name

category

…

isRented

lastRentDate

addedDate

**Library (Admin)**

users

books

rentedBooks

unrentedBooks

…

…

String getBookImageFileName(isbn)

LoadUsers()

SaveUsers()

LoadBooks()

SaveBooks()

Load()

Save()



