Project Library Collection

Functionalities

In this project we create an online library that allows customers to buy or rent Books and Journals
This project also includes a management interface for employees and managers.

Customers can:

- LogIn
- Sign up
- Select books or Journals.
- View information on these.
- Establish a basket.
- Buy books or Journals.
- Rent books.
- Pay for their purchase.
- Pay if they did not return the rented books.

Employees can:

- Select books or Journals.
- View information on these.
- Add and remove books or Journals.
- Manager stock of these.
- See books rentals
- indicate that a book have been returned or lost

Managers can:

- do the same things that Employee
- Add "Employee" users and "Manager"
- View report on sales, rental, users, and turnover 'case.

Model composition

The model defined an abstract class:

- AbstractItem

Two objects inherit this class

- Booklib
- JournalLib

An interface that defines the function of ItemCollection:

- IItemCollection

Three Collections:

- ItemCollection
- RentedItemCollection

- UserCollection
-

And Class that defined exceptions:

- RunOutException

AbstractItem

Defines an abstract object which takes as parameters:

- ISBN
- Price
- Title
- EditionDate
- stock
- Subject
- discount
- Image

2 functions to give a more or less detailed descriptions of the object.
a class that inherits IComparer order to sort the items.
An enumerator that contains the different categories of Books and Journals.

BookLib and JournalLib

are two classes that inherit from AbstractItem.
They override the ToString () function to add the nature of the item.

The three collections

ItemCollection

Gathers from a list of different items for sale and for rent. It has functions to get an instance:

- constructor obeying the singleton design pattern

to add Item:

- Add (AbstractItem absItem) That allow to add an Item to the collection and check if another Item have the same ISBN

To get an object:

- Get (int index) allow you to Get an item in the collection
- this [Guid ISBN]
- this [string title]

to get an object list based on criteria:

- Get (Category category)
- Get (decimal discount)
- PartialTitleResearch (string searchedWord)

- PriceResearch (double min double max)
- DateResearch (DateTime min, max datetime)

These functions are usefull during research by users

To remove an object from the collection features following criteria:

- the ISBN - remove (AbstractItem absItem)
- remove (int index)

To obtain information about the company:

- TotalStock
- Bookstock
- JournalStock
- bookcount
- JournalCount

These functions are usefull to make the report to managers.

UserCollection

Includes a userCollection.
An enumerator AutorisationLevel
and class User

User take as a parameter:

- Level that can be Used Client or Manager
- Name
- password
- Total That Gives the total amount of the current shopping cart, before purchase.
- Turnover That Gives the total amount of the user's purchased this item.
- NonReturnedBook That Gives the total amount due of the user for non-returned item.

The User Collection:

Includes a list of different users. It has functions to get an instance:

- constructor obeying the singleton design pattern

Obtain a user to display a matching interface to its AutorisationLevel and store his purchases and rentals:

- GetUser (string name, string password)

to check whether a user or username already exists in the collection

- IsUser (string name, string password)
- IsUser (string name)

to add a user: AddUser (user user)

to obtain certain data for the report to the manager:

- the per user AutorisationLevel: UserPerLevel (AutorisationLevel AUTOLEVEL)

- The total turnover: turnOver

RentedItemCollection

Contains a class RentedItem that combines a Book (AbstractItem), a user, and a rental date

Contains RentedItem Collection which includes a list of different RentedItem. It has functions to get an instance:

- constructor obeying the singleton design pattern

Add rental:

- Add (AbstractItem absItem, User user)

Remove rental:

- Remove (RentedItem item) when a client return or loses a book

Get a list of RentedItem for people who have exceeded the time:

- DateResearch (int dayNumber)
- DateResearch (DateTime date)

RunOutException

Contains three classes of exceptions:

- RunOutException when an item is out of stock.
- DoubleISBNException when trying to introduce an item with an existing ISBN in the collection.
- DoubleUserException when trying to introduce a user with an existing username in the collection.

Composition of the User Interface

The UI consists of three pages

- A login page
- The MainPage
- The ManagerPage

And a Class Manager that gathers the logic function.

Login page

This allows users to:

- Log in to the system.
- To create a user with a AutorisationLevel "Customer".

The page verify that the user exists and that passwords are valid.

The MainPage

is composed of two parts.

The first part

you can view the different books and Journals.

the user can choose to select them:

- by ISBN
- by Name
- By Price
- Per Date
- By category

and display the best promotions.

For each book selected by a client, he will be able to buy or rent. For each Journal only to buy it.

Employees and managers will be able to select each element:

- eliminate it.
- add or adjust its stock

The second part

- is for customers, it display the shopping cart where they'll be able to add and remove elements and complete their purchase.

- for managers and employees, there it shows the inventory management window to add new book and Journal.

The Page Manager

access only the manager and employees.
This page is also separated into two parts.

Part One

It provides access to the leased items.
Users can select multiple items rented and report that they have been returned by customers, lost by them, or give more time to return it.
The user can also filter the items by customer, date of return. to show only late client.

The second part

reserved for managers, it can display two different windows:

a report on the system involving the following information, it will collected in the various collections:

- Turnover
- The number of book
- The number of Journal
- The stock book
- The Journal stock
- The number of rented book
- The number of managers
- The number of employees

- The number of customers
- The average turnover by customer

It can also display a window for adding a user with a AutorisationLevel Manager or employee.

InterfaceManager
Brings together the logical part of the UI and the UI statics useful functions:

The statics functions
- Plurial (int data)

add an "s" when the data is superior to 1

- ValidPassword (string password)

testpassword

- ShowValidationOnTextBox(string message , color, color, TextBlock tBDestination)

show a Mail to selectionned txtblock

- ShowElementInListView (List <AbstractItem> absList, ListView ListV, TextBlock tBDestination)

Show on Item selected listview

- ShowElementInListView (List <RentedItem> absList, ListView ListV, TextBlock tBDestination)

Show Rented is Selected listview

logical fontions:
- ShowDependingLevel ()

Show element DEPENDING on user authorization level

- RequestISBN ()

Show the itemthat-have a specific ISBN inlistview
publicvoid requestName ()

- RequestPrice ()

Show the items That-have a price entre two price in listview

- RequestDate ()

Show the items That entre-have a date two dates in listview

- ActionAfterResearchTypeSelected ()

select qui Action will therefore DEPENDING on user choice

- Initi aliseListeView (IComparer <AbstractItem> alternativeComparer = null)

Sort item and show it in listview

- LviewAction (AbstractItem absItem)

Show Item selected in the textbox and in picture