**Introduction to C# Exercise**

This exercise is to allow you to put some of the techniques you have learnt into practice from scratch. You will put together your own application in this exercise based on the instructions given.

You will have one session 4 hours to work on this exercise and you can continue at home. This work will not be assessed and one possible solution will be presented at the next session.

Scenario

A company named Echo Holdings runs a small in-house library for its staff. Up till now the manager responsible for the library has used a spreadsheet to store the catalogue of books in the library and to keep track of loans. They would like you to come up with a C# application which enables them to automate the process and allow for expansion of the library.

Description of Current Solution

The spreadsheet contains a sheet containing details on the books held in the library, the catalogue. Another sheet is used to keep track of the books loaned to borrowers. The following information is stored in two sheets of the spreadsheet.

[Book Tab]

Title

Author

ISBN – (International Subscription Book Number)

Catalogue Number – Unique for each book as there may be more than one copy of each title

Cost - Replacement cost of the book

Format – Paperback, Hardback, Ebook

Dewey Decimal code – Catagorisation system used for non-fiction books

All the books the library owns are on this tab. There may be more than one entry for each book if more than one copy is owned

[User Tab]

First Name

Last Names

Employee ID

Catalogue Numbers – cells containing catalogue numbers of the books the user has borrowed along with the date each one must be returned

All library users are on this sheet. More detailed information for each user can be referenced from the HR database using their employee ID. For example if the Library manager wants to follow up and overdue book, this information is on the HR database. For this reason only the user’s names are stored on the spreadsheet to make them identifiable when using the spreadsheet. The loan period for the book is three weeks from the loan date. This date may occur on weekends and public holidays.

Requirements for new solution

Echo Holdings (referred to as the client) have created files containing data output from the spreadsheet. These are as follows.

**Book.xxx**

This file contains the book data with each copy of a book as a separate record in the file. This includes multiple copies of the same book. Each book record has the following fields.

Catalogue Number – Unique ID for each individual book , starts at 1000

Title

Author

ISBN – (13 digit International Subscription Book Number, also hold 9 digit ISBN)

Dewey Decimal code – Used only for non-fiction books, 9 or 13 digit

Cost - Replacement cost of the book

Format – Type represented by Integers for each type - Paperback, Hardback, Ebook

**User.xxxx**

First Name

Last Names

Employee Id

**Loans.xxx**

Employee ID

Catalogue Number

Date Due

The client would like develop the application iteratively with an initial set of features and more features being added over time. In discussion with the client you have determined that they would like to implement the following requirements at this stage.

1. They would like users, either the librarian or the borrower to be able to login to the application and see options and information appropriate for their role.
2. The Borrower should have the following options when they log in:

* They are able to view a list of books available to borrow
* They are able to view a list of books they have on loan along with due dates

1. When the borrower first logs in a dialog box is displayed showing any books which are overdue. If the borrower has no overdue books this is not displayed
2. The librarian has the following options when they log in:

* They can view a list of books in the library, each book shows the details of the book including replacement cost and indicates if the book is out on loan and if so the name of the borrower (perhaps allow drill down here with borrower details on the detail child)
* They can view a list of overdue books including catalogue number, title and the borrower details
* They can select and check books out to borrowers
* They can select and return books to the library
* They can add new borrowers manually using details provided from HR

1. When a librarian checks out a book if the borrower has an overdue book a message is displayed and they may not borrow more books until the overdue one is returned.
2. When the application is first loaded all records should be loaded from the files and stored in memory

1. Updates to records are to be written to the appropriate files as soon as the transaction occurs.
2. The solution is to implement error handling; users should be notified of any errors which arise.
3. You may need to add one or more additional files to implement this solution.

Optional Activity

* Create unit tests for one of the classes you have created.

General

You solution is up to you. You can C# features such as serialization, classes, enumerations, class inheritance it is up to you. Keep it as simple as possible. You may complete your solution at home if required and the solution with be discussed in the following class session.

Feel free to adapt the WPF code from the examples we have looked at so far (perhaps read the section on this to help). Alternatively use a text based console interface. Consider how you will provide a list of options to the user and allow them to select them. The number of items is kept low to allow a text based solution if required.

Consider how to adapt files such as the user file to indicate different types of users and the fact they will need to login. You may change the structure of the files and ad fields if required once the. You may need to read them in, modify them and save them again. This is a normal part of a project such as this where a manual system is being replaced.