UTCN
Computer Science Department
Software Design 2018

ASSIGNMENT A2

==========

1. Objective

The objective of this assignment is to allow students to become familiar with the Model View Controller architectural pattern and the Factory Method design pattern.

Design Patterns in wrong places lead to anti-patterns. Anti-patterns lead to stinky code. Stinky code leads to bad grades. Bad grades lead to the unhappy student. The unhappy student leads to the Dark Side.
Such is life - don't force design patterns.

Alex C.

2. Application Description

Use Java/C# API to design and implement an application for the employees of a book store. The application should have two types of users (a regular user represented by the book store employee and an administrator user) which have to provide a username and a password in order to use the application.

The regular user can perform the following operations:

- Search books by genre, title, author.
- Sell books.

The administrator can perform the following operations:

- CRUD on books (book information: title, author, genre, quantity, and price).
- CRUD on regular users' information.

- Generate two types of reports files, one in pdf format and one in csv format, with the books out of stock.

- **('lot of) BONUS POINTS:**
    - **Integrate a "Book Api" (e.g. Google Books API) in the app (4p)**
        - Suggestion for books and book data based on search title should be fetched from the API
        - The administrator searches for a book, selects something from the results and the fetched data is used to fill the local database (author, publishing date etc)
    - **Use PDFBox as a .pdf formatting library (2p)**
    - **Use Jasper as a .pdf formatting library (2p)**

3. Application Constraints

- The information about users, books and selling will be stored in ***a database***, **not in freaking XML files**. Use the Model View Controller in designing the application (this should come as a default for most frameworks). Use the Factory Method design pattern for generating the reports (***if you see fit, if not, use other patterns)***.
- All the inputs of the application will be validated against invalid data before submitting the data and saving it.

4. Requirements

- Implement and test the application
- Use a framework (Spring, .NET, Grails, Rails)
    - NOT a necessity but a **very, very strong** recommendation
- Everything through www.github.com

5. Deliverables

- Implementation source files
- Proper GitHub repository
- Integration **and** unit tests for services

6. References

https://github.com/UTCN-SoftwareDesignLab/SoftwareDesign2018

Oh, and Google.