**Software Development CA2**

Create a Java-based library catalogue system that manages information about resources, authors and users.

The objective of this assignment is to design and implement a file-based data management system using Java. Students will work in groups to create a system that stores, retrieves and manipulates data objects, making use of Java interfaces, abstract classes, regular classes, stacks, queues and/or arraylists (one or more), external files (CSV) for storage and Maven dependencies for enhanced functionality.

Requirements

1. Define the following classes:
   1. **Book/Audio Book**: Represent with attributes like title, author, ISBN, availability status
   2. **Theses/Dissertation**: Represent with attributes like title, author, topic, abstract, date published, availability status
   3. **CD/DVD**: Represent with attributes like title, producer, director, playtime, availability status
   4. **Author**: Represent attributes like name, list of authored books
   5. **Library User**: Represent with attributes like name, ID, list of borrowed assets

***Note***: The above list of classes and attributes is the minimum requirement and is not definitive. You can add any others you deem necessary for your design.

1. Implement appropriate constructors and getters/setters
2. Use inheritance and interfaces to design an efficient class hierarchy, eg, you could create an abstract class to represent library item attributes and behaviours shared by both books and media or an interface that defines methods for adding, editing, deleting and retrieving content from a data file. These would then by used by classes which would implement the required methods accordingly.

*The design should also consider use of built in interfaces such as Comparable*

1. The system will implement stacks and/or queues and/or linked lists.
2. The system will implement file handling.
3. The system will implement error handling.
4. The system will implement sorting and searching algorithms.
5. The system will implement the following functionality:
   1. Add an asset to the catalogue
   2. Add an author to the catalogue
   3. Add a library user to the system
   4. Borrow an asset (update asset availability and library user's borrowed assets)
   5. Return an asset (update asset availability and library user's borrowed assets)
   6. List available books
   7. List assets borrowed by a user
   8. List assets authored by an author
   9. List of overdue assets
   10. Any other options you desire
6. The system will implement Maven Dependencies.
7. User Interface
   1. Create a simple command-line user interface to interact with the library system
   2. Allow users to execute library operations through text-based menus

Consider what interfaces, classes, data structures and searching/sorting algorithms you intend to use for the implementation. Justification for your decisions, which should be backed up by theory (eg, big O notation, running time, complexity, stability, etc), needs to be provided in the design documentation along with class diagrams.