CSCI 5448 - Object-Oriented Analysis & Design
Homework 6 Final Report (Library System Web Application)

Group members:

1. Dwight Browne
2. Sepideh Goodarzy
3. Maziyar Nazari
4. Maram Kurdi

Documented Final System State

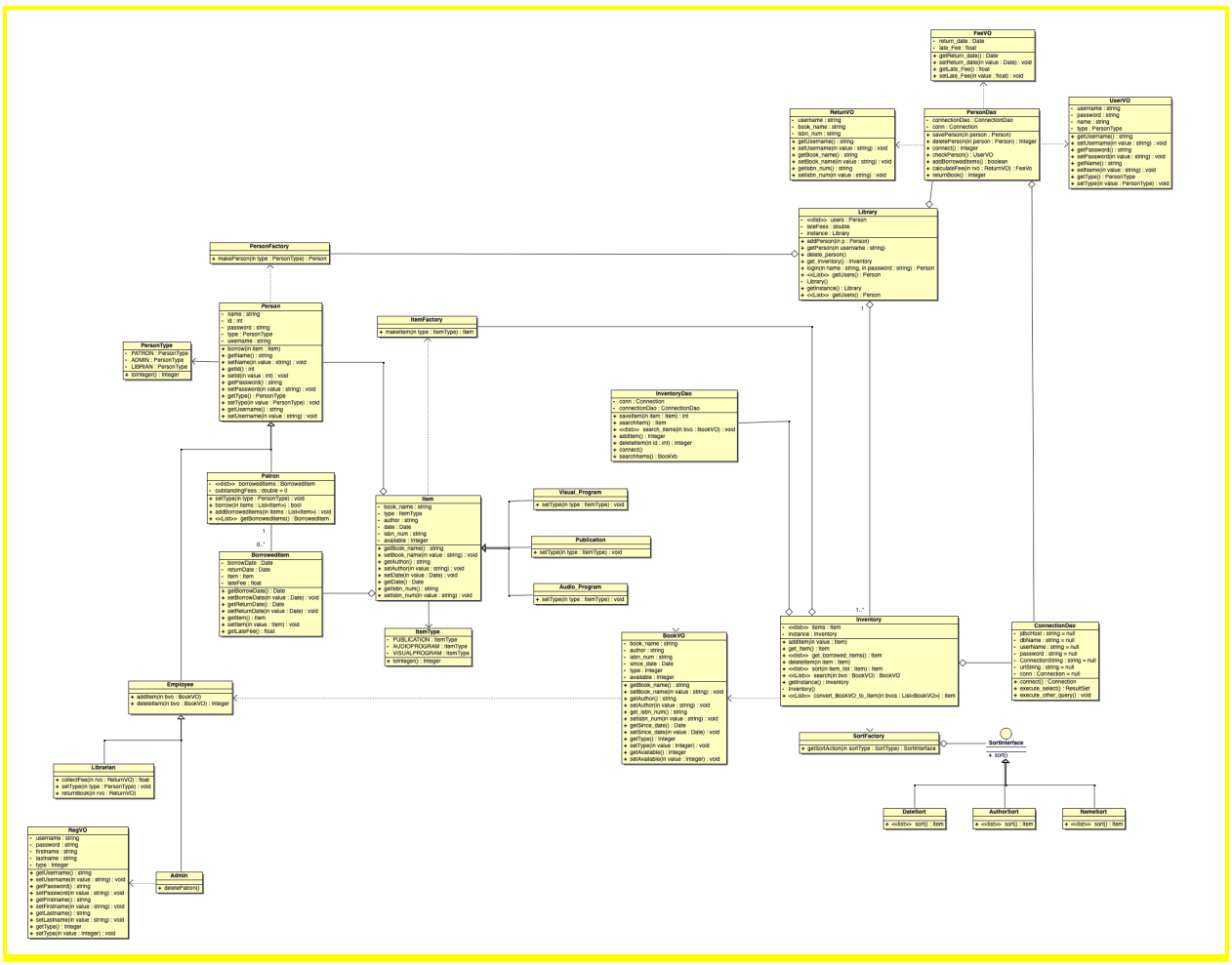
1. Discuss state of system, what features are implemented?

   In this project we implemented a web based library system that provides Patrons, Admin and Librarian with the necessary functionality to borrow and manage a non trivial universe of materials that would be borrowed in a contemporary library. The system allows admin to register patrons. Librarian can add, delete, and search for materials as well as submitting a return request on behalf of a Patron and knowing how much does patron have to pay because of the late returning(if there is any). On the other hand, patron can login to the system, search for materials and borrow material.
2. List of features implemented

Librarian features implemented	Admin Features implemented	Patron features implemented
Librarian can login	Register new patron	Patron can login
Librarian can get return fee	Search for material	Patron can search for material
Librarian can add/delete materials	See the result	Patron can borrow material(add to his/her cart)
Librarian can search for materials	Sort the search results based on name, author and date	Sort the search results based on name, author and date
Librarian can submit and confirm a return request on behalf of a patron		
Sort the search results based on name, author and date		

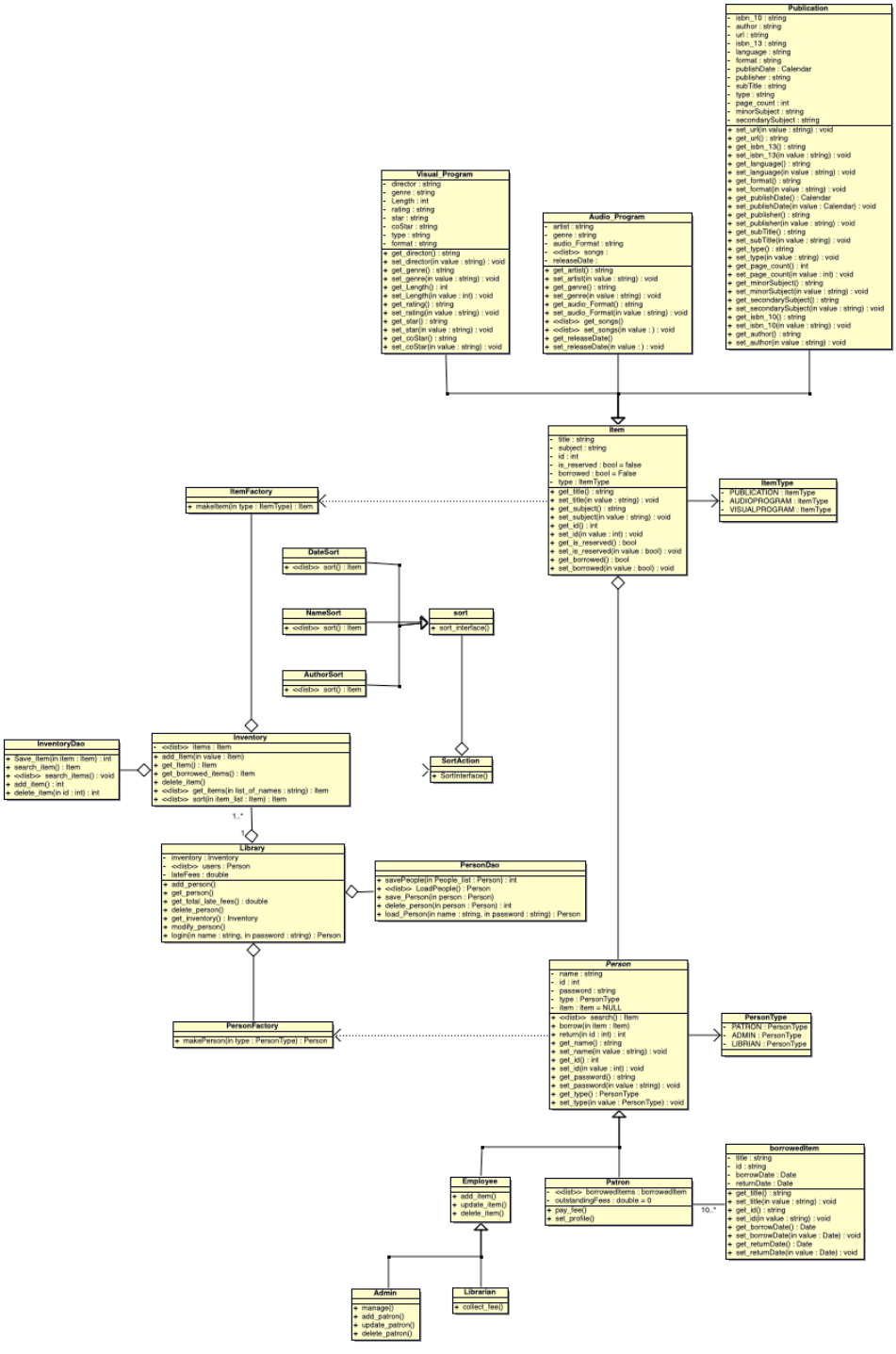
Class Diagram

1. UML showing final set of classes and relationships in the systems



Final Project vs. Initial Design

1. Include Class Diagram from Homework 4 to illustrate changes



2. Discussion of changes (If no changes from initial design, this should be discussed)
    - Added 5 new classes :
        i. connectionDao: missing from original design
        ii. BookVO: *VO classes were needed for the web interface
        iii. UserVO
        iv. FeeVO
        v. ReturnVO
    - Modified 13:
        i. PersonDao : structural changes added some book functionality
        ii. InventoryDao : Added *VO class support
        iii. Admin: Minor functional changes
        iv. BorrowedItem : functional changes
        v. Employee :Functional changes
        vi. Inventory: Now a Singleton and added conversion method for BookVO
        vii. Item : Major changes to accomodate simplification of project
        viii. ItemType : added toInteger method
        ix. Librarian : some functionality changes and *VO class handling
        x. Patron : changes to borrow functionality
        xi. Library: Changed to Singleton and added *VO class functionality
        xii. Person: Simplified person and removed some functionality
        xiii. PersonType :added toInteger method

Third-Party Code vs. Original Code

1. What external code was used, what code did you write?
   Code was written from scratch, no external code was used.
2. List of third party code elements (libraries, utilities) with cited sources
    A. Bootstrap framework: used to design the websites. It includes HTML and CSS based design templates for typography, forms, buttons, tables, navigation, modals, image carousels, etc. It also gives support for JavaScript plugins.
    B. jQuery: used to simplify JavaScript code. It takes common tasks that require many lines of JavaScript code to accomplish, and wraps them into methods that can be called with a single line of code.
    C. Tomcat web server: Tomcat acts or behaves as a development server on your desktop to use for testing when building applications that use JSF 2, servlets/JSP, or other Java-based dynamic Web technologies.
    D. Mysql database: is an open-source relational database management system. Used to store the data of our system.

E. AngularJS: is a structural framework for dynamic web apps. It lets you use HTML as your template language and lets you extend HTML's syntax to express your application's components clearly and succinctly
F. Three Tier Architecture: presentation (view), application processing (domain), and data management functions (data access) are physically separated from each other.

3. List of original code elements of design

Code was written from scratch, no external code was used.

Design Patterns

1. Were any used, if so how?

A. Factory design pattern:
- Used in the abstract Person class to create through inheritance three different types of person: patron, admin, and librarian.
- Used in the abstract Item class to create through inheritance three different types of item: publication, audio program, and visual program.

B. Singleton design pattern:
- Used for Library class and Inventory class to restricts the instantiation of a class to one single instance.

C. Strategy design pattern:
- Used to select the appropriate sorting items algorithm at runtime based on name, author, or date.

Learnings on OOAD

1. What has been learned about the process of OO analysis and design from this project?

After taking OOAD class this semester, we know that even though UML is easy to be ignored but extremely useful, which we cannot understand if we were not doing this project. UML not only helps us build the project quickly, but also makes someone else easily understand the whole architecture and even small components, which is surely helpful for others to reuse the system. In other words, paying more attention on system design, especially for large systems, is very necessary even though it takes some time before the implementation, it will definitely save much more time and avoid some issues later. Additionally, help of design patterns is also beyond our imagination. Since lots of pieces of code are almost same, we can achieve reusability and flexibility via utilization of design patterns. However, there are many design patterns we can use, which causes a problem that which is the most appropriate one to our system. In this case, we need to fully understand the features of different design patterns and application scenarios, and practice frequently.