Notes before we start

* Documents inventory entity will be a database
* The print class will be implemented as a singleton pattern
* Staff list is also a database

Brief class descriptions

User

The abstract super class of which the other users extend from

Holds: user ID, user password, user name, login object

Functions: login, logout

Login

A class User uses for logging in

Holds: none

Functions: validate

Manager

A subclass of User for managers

Holds: no more than the superclass

Functions: check inventory

SystemAdmin

A subclass of User for the system admin

Holds: no more than the superclass

Functions: add staff info, remove staff info, update staff info

Author

A subclass of User for authors

Holds: list of contributed documents, a publish object

Functions: submit documents to be published

Operator

A subclass of User for operators

Holds: no more than the superclass

Functions: add document, remove document, update document

OrdinaryBuyer

A subclass of User for non-registered buyers

Holds: a list of transactions

Functions: search documents, order documents, pay documents, register

RegisteredBuyer

A subclass of OrdinaryBuyer for registered buyers

Holds: no more than the superclass

Functions: access promotion list, unsubscribe

TransactionHistory

A class which holds information on a buyer’s past transactions

Holds: list of receipts

Functions: calculate total sum of payments, find total amount of purchases

Receipt

A class which records a purchase details

Holds: price, date, list of items

Functions: none

Staff

A class which represents a staff member’s info

Holds info: name, staff ID, and position

Functions: none

Document

A class which represents a document’s info

Holds: name, author, document ID, publication date, document type

Functions: none

PromotionList

A class which represents the documents that are tagged as promoted in the database

Holds: list of documents, list of observers

Functions: none

Publish

A class which holds the queue of documents to publish, implements singleton design pattern

Holds: documents queue, a publish object

Functions: publish next, add document, get instance of publish

SystemAdminForm

A class which is the interface between the SystemAdmin and the StaffInfoDatabaseController

Holds: system admin name, system admin ID

Functions: add, remove, update

AuthorForm

A class which is the interface between the Author and the DocumentDatabaseController

Holds: manager name, manager ID

Functions: browse documents

BuyerForm

A class which is the interface between the Buyer and the DocumentDatabaseController

Holds: buyer name, buyer ID

Functions: search, order, pay, register, unregister, access promotion list

OperatorForm

A class which is the interface between the Operator and the DocumentDatabaseController

Holds: operator name, operator ID

Functions: add, remove, update

UserDatabaseController

A class which controls the reads from a user database

Holds: database name, database connection

Functions: read from database

StaffInfoDatabaseController

A class which controls the reads and writes to a staff info database

Holds: database name, database connection

Functions: add to database, remove from database, update database

DocumentDatabaseController

A class which controls the reads and writes to a document database

Holds: database name, database connection

Functions: add to database, remove from database, update database, get list of documents

GeneralBookFormat, GenericJournalFormat, and GenericMagazineFormat are strategies to set at runtime which will have algorithms to publish different types of documents.

DocumentDecorator is an abstract super class which implements and owns a component used to decorate. CoverLayerOpaquity and BackgroundImage are subclasses of DocumentDecorator that own a component themselves which allows for them to decorate. CoverBoarder implements Component but does not own a component itself so it is the last Component to add.