

ECS505U Software Engineering

Laboratory Exercise (**Assessed, 2.5% of final mark**)

Generating UML Diagrams from Requirements

Version 1.7, 07/11/2017

1. Prerequisites

- You should have access to Visual Paradigm. For campus students, version 14.1 is pre-installed on all machines. For non-campus students check the QMplus course page for download and licensing information.
- You should watch the video titled “UML extended class diagram video tutorial” available on QMplus course page.

2. Glossary

The library system (or the system): The software system designed to help people in the library.

System User: A person in the library who is using the system. Includes all two types of users described below.

User: Users are non-member system users.

Member: A system user who is registered using the library system.

Loan: Each borrowing of a library item by a user.

Library Inventory: Where the records of all items and loans in the library is kept.

Item: A physical entity that can be loaned from the library (such as a copy of a book or a movie DVD/Blu-ray).

3. System Requirements

3.1. Library Items

1. The system must be able to keep track of all library items: books, magazines and movies.
2. Each library item must be identified with a unique identification number which is assigned by the library.
3. The system must store the following details of a book: title, ISBN and author.
4. The system must store the following details of a magazine: title and ISSN.
5. The system must store the following details of a movie: title and ISSN.
6. The system must store the following details for a catalog: Catalog name and ID.

3.2. System Users

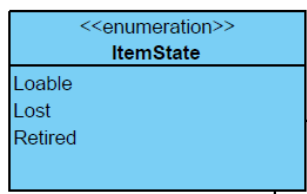
8. The system must have 2 types of users: user and member.
9. Users must be able to register using the library system to become members.
10. All members must be identified by a unique username.
11. System must store the following information for a member: first and last name.
12. A member must have at least one address in the system.
13. A member can have up to seven addresses.
14. An address must include the following details: address name, address Line, post code
15. All system users must be able to search items in the library by the following:
 - a. Search (for all items)
 - b. Books
 - c. Magazines
 - d. Movies
16. Only members should be able to loan items.

3.3. System functionality

17. The system must hold a list of all items in the library inventory.
18. The system must hold a list of all loans.
19. The system must hold a list of all catalogs.
20. The system must allow adding and removing items to/from library inventory.
21. Catalogs consist of books.
22. System must allow adding and removing books to catalogs.
23. An item can be loaned to a single member.
24. A member can loan up to 5 items.
25. The system must store the loan date and due date.
26. The system must allow having non-loaned items.
27. The system must be able track the state of items (use the ItemState enum for this).

HINTS:

- Create the following enumeration class for you item states.



- Item state can be a nested/inner class. You need to decide in which class it should be in.
- When designing the relation between ItemState and Item consider if there is a need for both to hold a link to each other.
- To fulfil RQ1 to RQ5 you need to think of a relation between item (a copy) and book/magazine/movie. One of the easiest ways of doing this is to use **abstraction-occurrence design pattern**. This design pattern is achieved using an abstraction class (book class representing a unique book) and occurrence class (item representing physical copies of the same book). The relation between these classes is 1 to many, to allow having multiple copies of a book.



Be careful when designing your system because you need to consider the fact that your design has to represent three different type of items.

- To fulfil RQ17 to RQ19 we represent the library inventory as a class. The system expected to keep lists of all loans, catalogs and items in the library inventory. (My design decision!)
- There are two types of system users with common behavior (such as search) but they also have their specific attributes and behavior. Think of a good design to represent this relation between two system users (use of an interface might be a good idea, my design decision again)
- To fulfil RQ21 consider the following scenario in which a catalog is removed from the system. In this scenario, should the system remove books in the catalog or should it keep the books.

You are expected to use the one of the names below for each class name in your class diagram. The methods described should be included one of the classes following the requirements.

Book, Magazine, Movie, Item, BookItem, MagazineItem, MovieItem, LibraryInventory, Catalog, Loan, UserInterface (this is an interface), User, Member, Address

Place the following attributes, methods and constructors in your classes to fulfil the requirements. You need to use some of the attributes more than once.

```
-isbn : String
-author : String
-title : String
-issn : String
+addItem(item : Item) : void
+removeItem(item : item) : void
-itemID : String
+Item(itemID : String)
+getItemID() : String
+getItemState() : ItemState
-catalogName : String
-catalogID : String
+Catalog(catalogName : String, catalogID : String)
+addBook(book : Book) : void
+removeBook(book : Book) : void
-loanDate : Calendar
-dueDate : Calendar
+Loan(item : Item, member :Member, dueDate : Calendar)
+search(text : String) : List< Item>
+searchBooks(text : String) : List< Book>
+searchMagazines(text : String) : List< Magazine>
+searchMovies(text : String) : List<Movie>
-userName : String
-firstName : String
-lastName : String
+Member(userName : String)
+loan(item : Item) : Boolean
-addressName : String
-addressLine : String
-postCode : String
+User()
+register(userName : string, firstName : string, lastName : string) : boolean
```