## DEPARTMENT OF COMPUTER SCIENCE

Session: Aug– Dec 2017

# LAB MANUAL

| | |
|---|---|
| Semester: | VII |
| Sub Code: | UE14CS404 |
| Sub: | OBJECT ORIENTED MODELING AND DESIGN LAB |

ISA Evaluation

1. Observation→10 Marks
2. Conduction→20  Marks
3. Viva→10 Marks
4. Mini-Project→20 Marks

Total→60 Marks

ESA Evaluation

1. Lab Quiz→20 Marks
2. Lab Conduction→20 Marks

Total→40 Marks

## LIST OF ASSIGNMENTS/PROGRAMS

| *Week No* | *Program No* | *Title of the program/ Problem Statement* |
|---|---|---|
| 1 | Instruction / Introduction class | Introduction to Lab |
| 2 | Program 1 | Identify the Actors and the corresponding Use Cases; Describe each Actor and Use Case; Show relationships between Actors and Use Cases |
| 3 | Program 2 | Class Diagram-Identify the Classes, Responsibilities, Collaborators, Attributes, and Operations. Prepare the data dictionary for LIMS classes.-Identify the relationships (Associations, Generalization, Aggregation) |
| 4 | Program 3 | Identify major Activity Flows and prepare corresponding Activity Diagrams |
| 5 | Program 4 | Prepare Sequence Diagrams for the significant Use Cases |
| 6 | Program 5 | Prepare the State Diagram for the significant Classes |
| 7 | Program 6 | Prepare the Collaboration Diagram for the significant Use Cases |
| 8 | Program 7 | Prepare the Component and Deployment Diagram |
| 9 | Program 8 | Application of SOLID and GRASP principles. |
| 10 | Mini Project on Design Patterns | |
| 11 | | |
| 12 | | |

These experiments will be conducted using Star UML on Linux

**Case Study for all Lab exercises:**

# Library Information System
# Software Requirements Specification

## Introduction

City College of Engineering (CCE) would like to computerize their library operations with the help of a Library Information and Management System (LIMS). Currently library operations are manual, and CCE would like to use a state-of-the-art LIMS. This document provides the requirements for LIMS.

## Background

CCE was established in 1990 and has since then its reputation has grown steadily. It now has about 5,000 students and about 200 staff. Its library has over 50,000 items of various types such as books, journals, bound journals, magazines, educational CDs, and educational DVDs. Many items have multiple copies.

The users of LIMS will include students, faculty and staff of CCE. In general, users of the system can be classified into two categories: End Users and Administrative Users (typically library staff).

## Functional Requirements

### General

All users will have to logon to the system before they are provided access to any functionality. They will also be able to change their password at any time.

### End User Requirements

End users include the following: Students, Staff, and Faculty members. Every end user will be provided a library card with a barcode that is necessary to check out any item from the library.

End users should be able to:

- Search the catalog for library items. Search options include type of item and appropriate meta data
- View the details of a particular item, including whether it has been issued, and if so, when it is due
- Make a reservation for an item that has been issued. The system will accept at most three reservations for an item that is due, which will be serviced on a first come first served basis
- View the library items issued to them as well as their due dates. Overdue items should be highlighted so that attention can be drawn to them quickly
- View the fine, if any, due on each item
- Get general information about library policies
- Only faculty members will be able to submit requests for new items which will be reviewed by the librarian

**Administrative User Requirements**

There are different types of administrative users: Library Clerk, Library Staff, and Librarian.

The Library Clerk should be able to:

- Issue an item to an end user on the production of a library card. All library items will have a barcode. The issue process will use a barcode reader to read the code on the library card as well as that on the item(s).
- If the item has reservations on it, it can be issued only to the person who made the reservation, according to priority
- Process the return of an item. If there is a fine, the fine may be paid, in which case a receipt is issued. Fines may be accumulated up to a maximum determined by the librarian, who can also waive fines

Library Staff should be able to:

- Add new users to the system
- Remove users from the system. A student should be removed after they graduate provided they have paid their dues. CCE Employees will be removed after they cease to be employees
- Deactivate/reactivate a user when authorized by the librarian. Deactivated users will not have access to any functionality
- Add library items to the system when they come in
- Modify information about library items
- Remove library items from the system when authorized by the librarian

The Librarian will have access to all the functionality, and will, in addition, by able to

- Make queries pertaining to user and items
- Modify details of any user of item
- Generate reports on users and items

- Implement policy decisions such as the number of items that can be borrowed by a particular category of user, checkout duration, fines, and so on.
- Waive fines

## Non-Functional Requirements

### User Interface Requirements

- LIMS should be accessible to the general user through a standard browser interface.
- The Administrative users will use a thick client.
- The UI should be self-explanatory, and a help feature should also be available.

### Performance Requirements

- Response to queries should not exceed 10 seconds
- The system must support up to 200 concurrent users

### Security

- Unauthorized access to the server must be prevented
- Unsuccessful login attempts must be logged and an alert generated

### Other Requirements

- LIMS is expected to run 365 X 24
- CCE would like to minimize the cost of developing LIMS; hence open source and freely available software must be used to develop LIMS

## Week #1: Instruction class - Introduction to

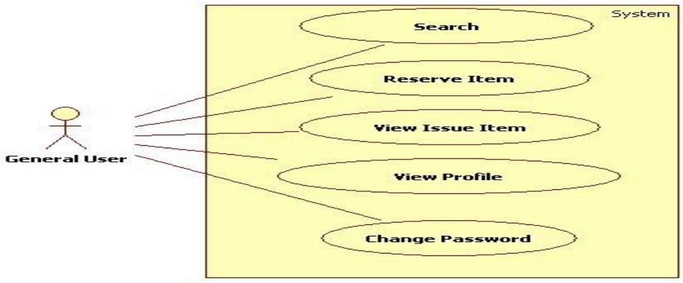
| | |
|---|---|
| Learning objectives | How to Use Star UML |

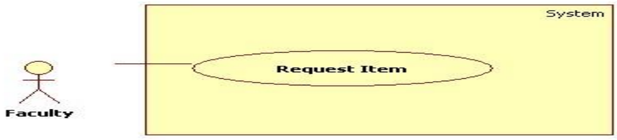
## Week 2 - Program # 1 - Use Case Diagram

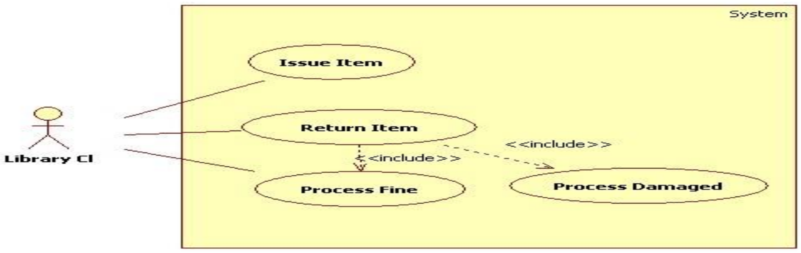
**Problem:** Identify the Actors and the corresponding Use Cases and come up with Use Case Diagram and also show the inheritance for the same

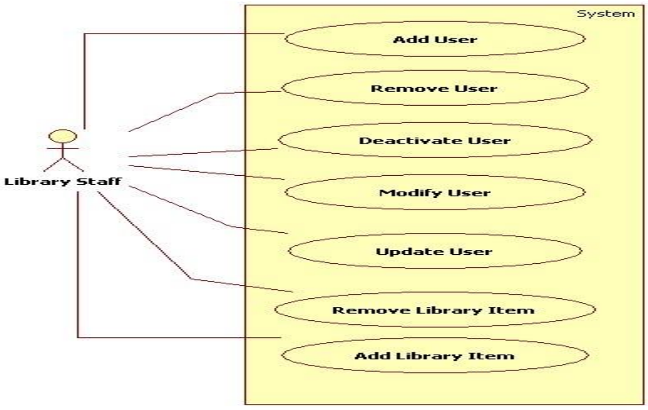
**Solution:**

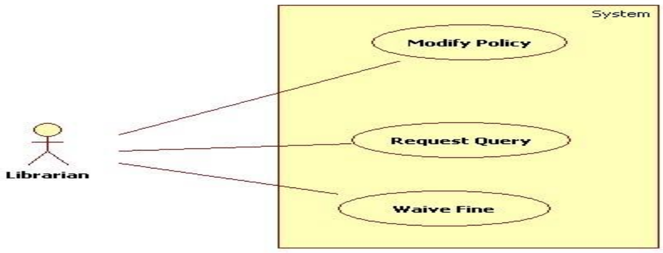
1. Identify the Nouns as the Actors and Verbs as the Use Cases.
2. Identify associations like <<includes>>, <<extends>> and generalizations.
3. Identify the actor hierarchy, implying what use cases an actor can invoke additionally.
4. Draw the corresponding use case/hierarchy

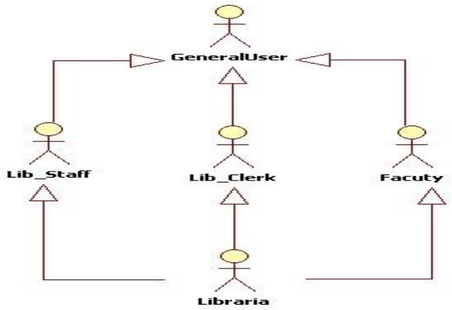

System
Search
Reserve Item
View Issue Item
View Profile
Change Password
General User


System
Request Item
Faculty


System
Issue Item
Return Item
<<include>>
<<include>>
Process Fine
Process Damaged
Library Cl


System
Add User
Remove User
Deactivate User
Modify User
Update User
Remove Library Item
Add Library Item
Library Staff


System
Modify Policy
Request Query
Waive Fine
Librarian


GeneralUser
Lib_Staff
Lib_Clerk
Facuty
Libraria

## Week – 3 , Program 2 : Class Diagram

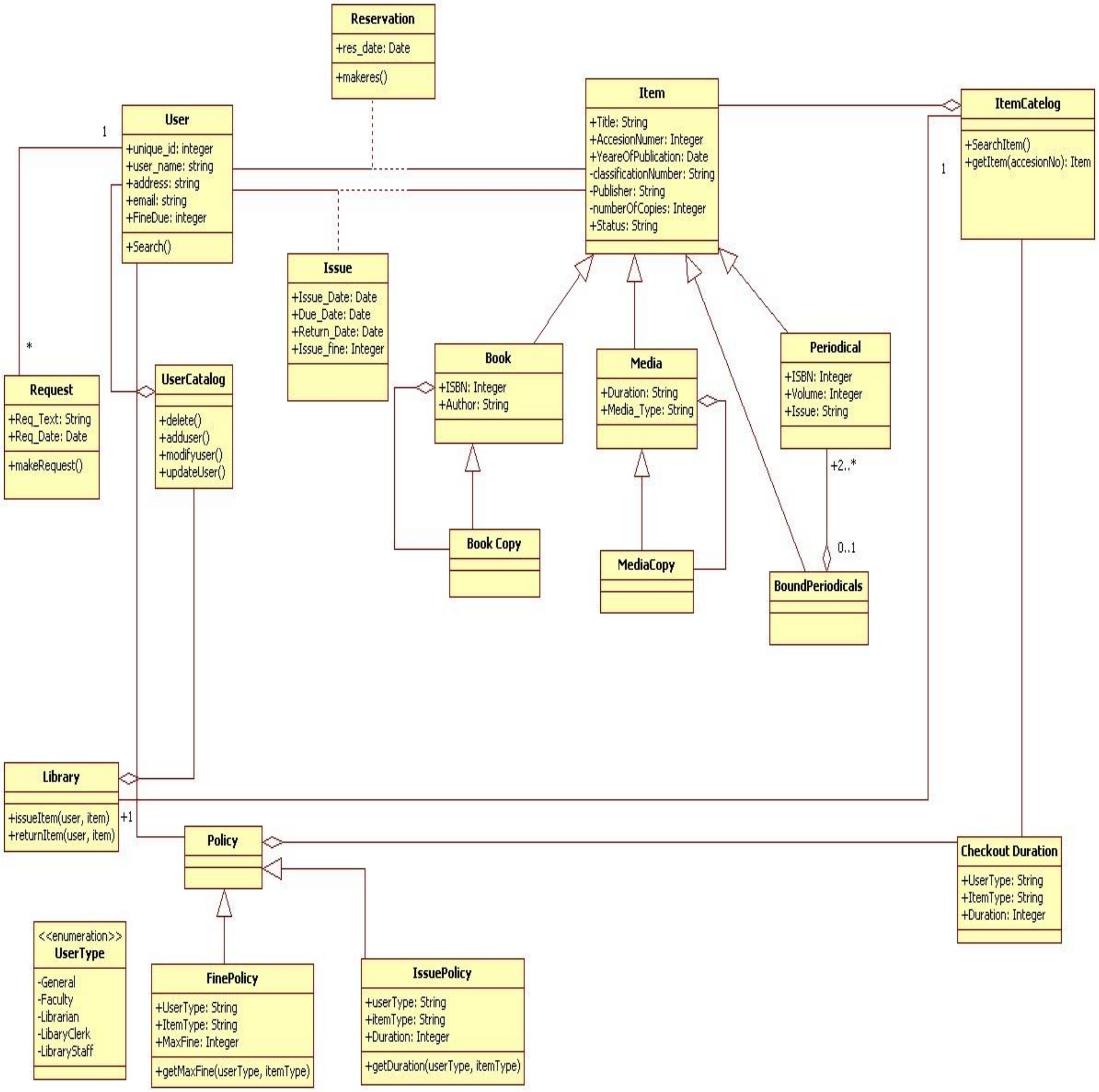
**Problem:** Class Diagram-Identify the Classes, Attributes, and Operations. Prepare the data dictionary for LIMS classes.

**Solution: 1.** Identify the nouns and choose the ones that can become classes. Identify the attributes of the classes based on description provided in Case Study or from drawing parallels from the Institute library. Identify the operations each class can perform.

## Program 3 : Class Diagram

**Problem:** Class Diagram-Identify the relationships (Associations, Generalization, Aggregation)

**Solution: 2.** Identify how the classes are associated and their cardinality. Identify Generalizations and Aggregations



## Data Dictornary for LIMS  classes :

**Book :** Represents a libray item which contains unique ISBN, Title, Author as its attributes**.**

**Book Copy :** Represents an instance of the item book

**Checkout Duration :** Represents the duration till which the user can hold a particular item. Attributes include Type of the User, type of the item and the duration.

**Issue** : A Library clerk can issue items to the user which contains an issue date, due date , return date, and fine.

**Item** : Item can be any library item which includes book, CD, DVD, Periodical, Bound Periodical etc..Each of these items must contain the following attributes, i.e the title, accession number, year of publication, publisher, classification number, number of copies, staus of each copy.

**Item Catelog :** Includes the functionalities that a user can perform. Functionalities include searching a library item, reserve an item, get a library item etc..

**Library** : Main class which is responsible for overall functionalities of LIMS.

**Media** : Represents a libray item which contains type of media, title & duration.

**Media Copy** : Represents an instance of an item media.

**Periodical :** Represents a libray item which contains title, ISBN, Volume number, issue number.

**Policy :** Librarian can create various policies for the user and items. Policies include fine policy, duration policy, issue policy.

**Reservation:** Users can reserve library items.Reservation is done based on the proprity of the user. Only 3 reservations can be made.

**Request :** User (Libray Staff) can request an item.

**User:** Are people who accesses the LIMS. Users are a General User or an Administrative user. General Users include students, staff. Administrative user are Libray staff, Library Clerk or an Librarian. Each user will have a unique user id, name, address, phone number, e-mail id, fine details.

**User Catelog:** User Catelog contains various functionalities that a Library can perform with respect to user on a LIMS. Functionalities include delete the user, add new user, deactivate the user account, re-activate account, invalid user.

## CRC (Class – Responsibilities – Collaborators) for significant classes

| USER | |
|---|---|
| | User Catalog<br>Reservation, Request<br>Item, Item Catalog<br>Issue<br>Policy |

| ITEM | |
|---|---|
| | User<br>Reservation<br>Item Catalog<br>Issue |

| ISSUE | |
|---|---|
| | User, User Catalog<br>Item, Item Catalog<br>Policy |

| USER CATALOG | |
|---|---|
| Create User<br>Modify User<br>Delete User<br>Get User<br>Search User | User<br>Policy<br>Library |

| ITEM CATALOG | |
|---|---|
| Create Item<br>Modify Item<br>Delete Item<br>Get Item<br>Search Item | Item<br>Policy<br>Library |

| LIBRARY | |
|---|---|
| Issue Item<br>Return Item | User, User Catalog<br>Item, Item Catalog<br>Policy<br>Issue |

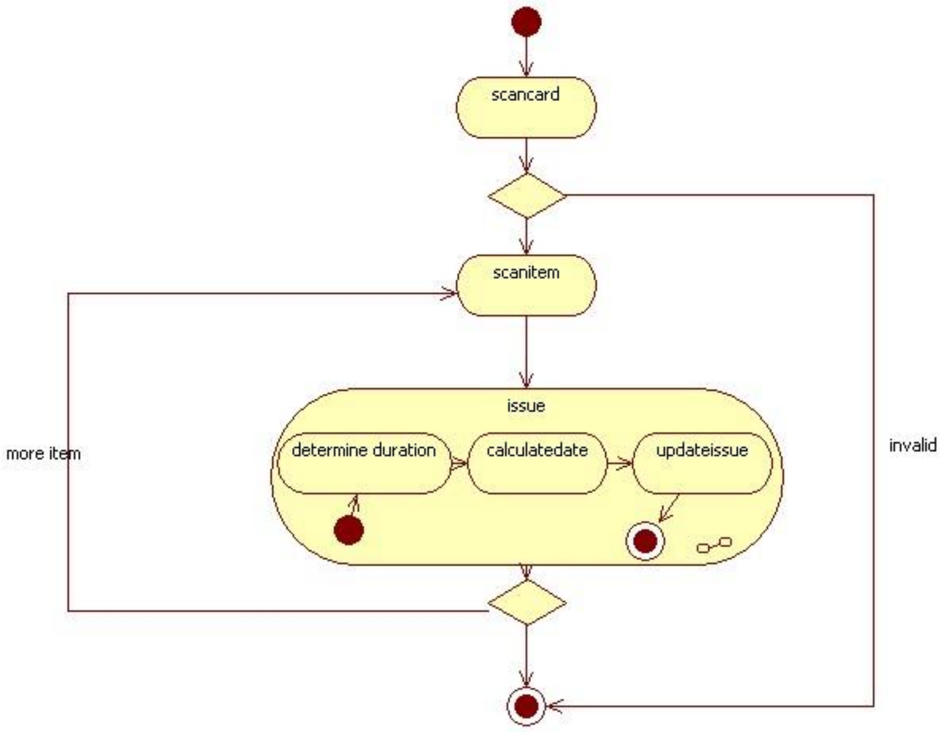
| REQUEST | |
|---|---|
| Make Request | User, User Catalog |

| RESERVATION | |
|---|---|
| Make Reservation | User, User Catalog<br>Item, Item Catalog |

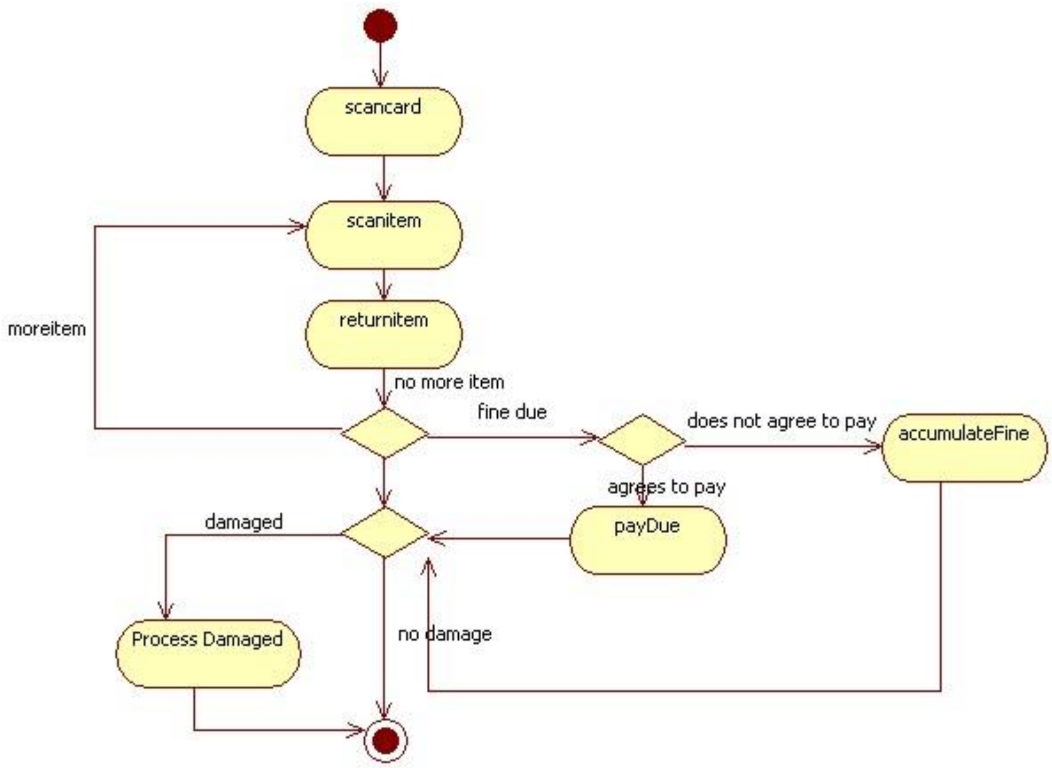
### Week 4- Program 3 : Activity Diagram - Issue

**Problem:** Identify the Activity Flow and prepare the Activity Diagram

## Solution:

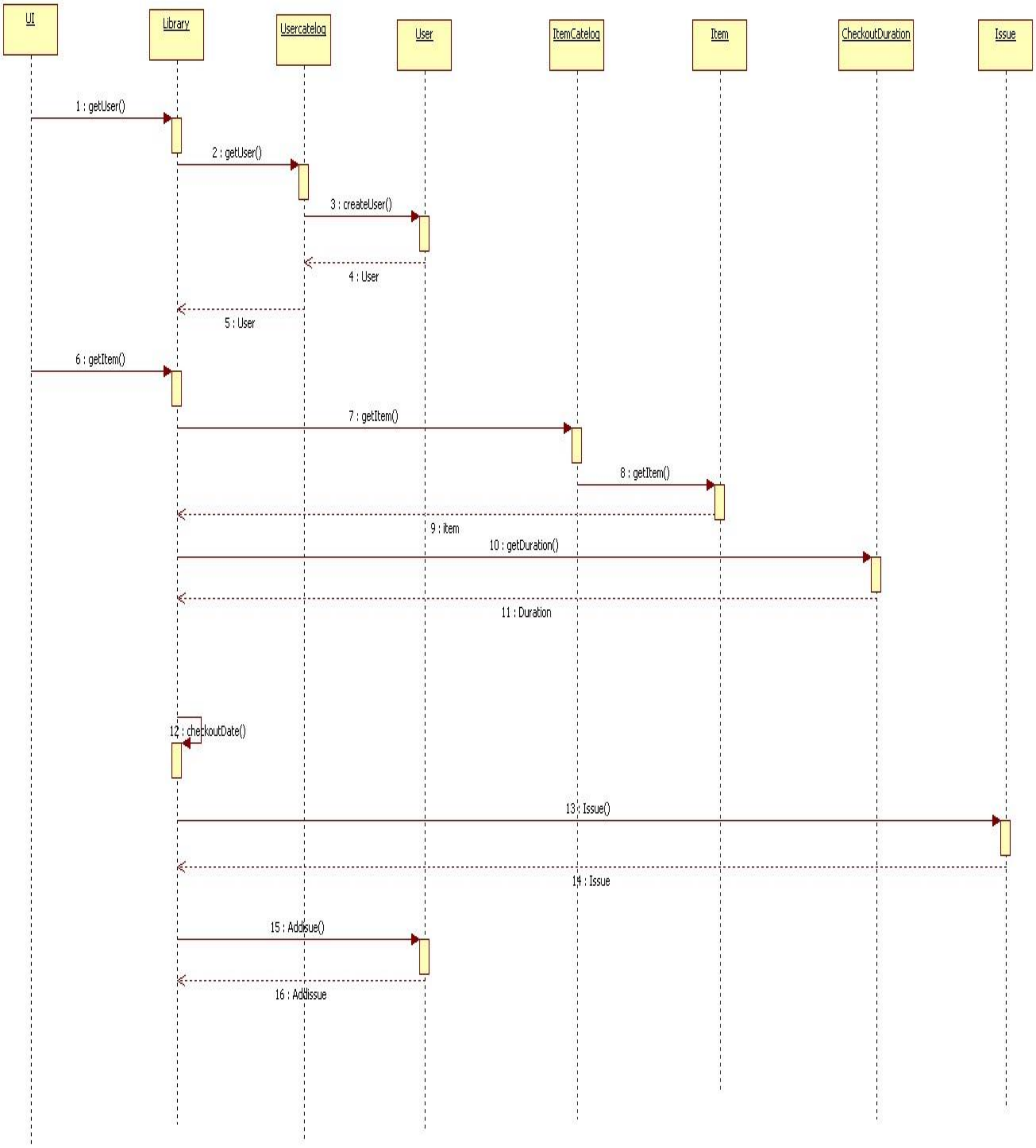


## Activity Diagram – Return

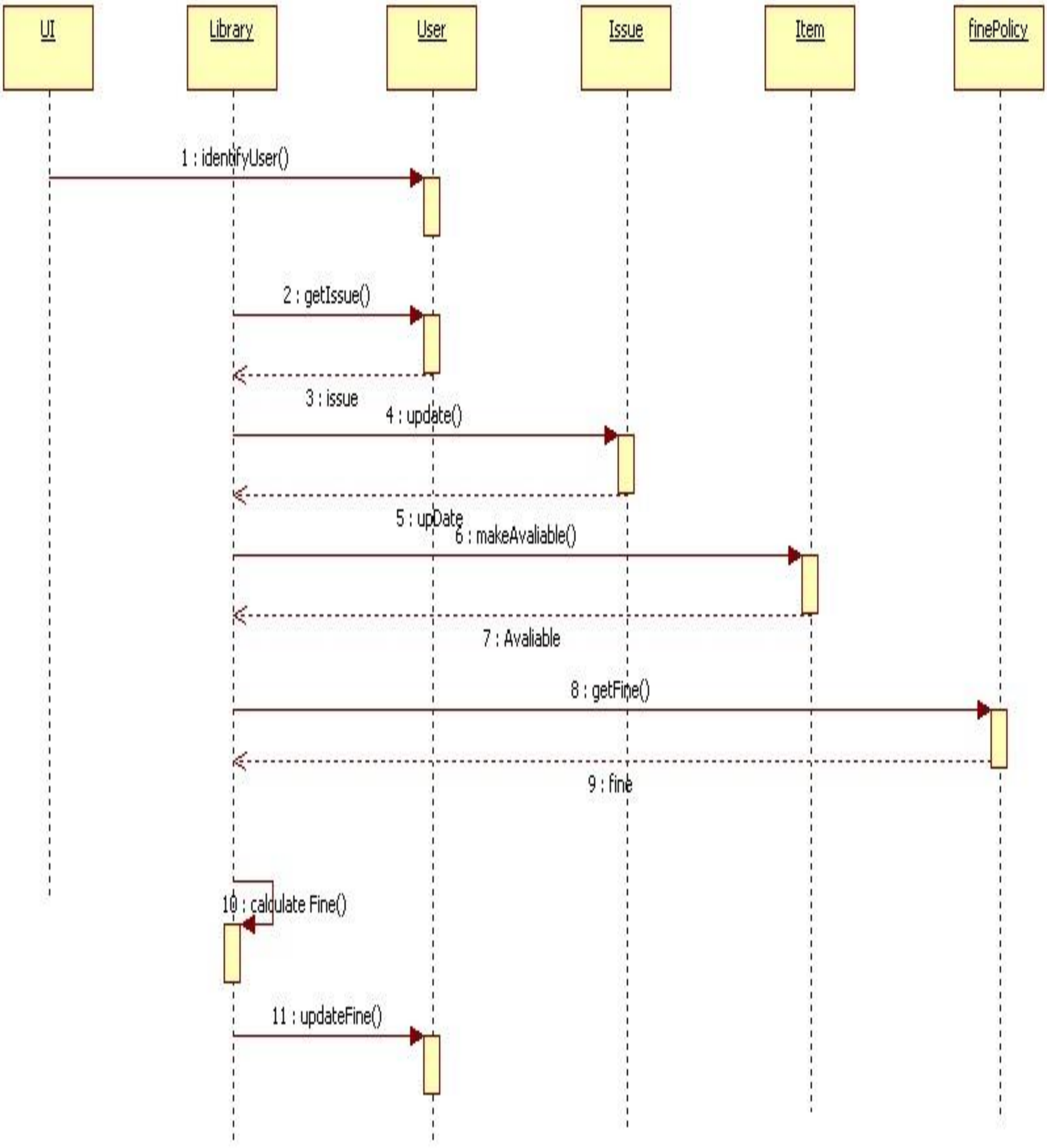


## Week 4- Program 5 : Sequence Diagram - Issue

**Problem:** Prepare the Sequence Diagram for the significant Use Cases
**Solution:**


UI
Library
Usercatelog
User
ItemCatelog
Item
CheckoutDuration
Issue
1 : getUser()
2 : getUser()
3 : createUser()
4 : User
5 : User
6 : getItem()
7 : getItem()
8 : getItem()
9 : item
10 : getDuration()
11 : Duration
12 : checkoutDate()
13 : Issue()
14 : Issue
15 : AddIssue()
16 : AddIssue

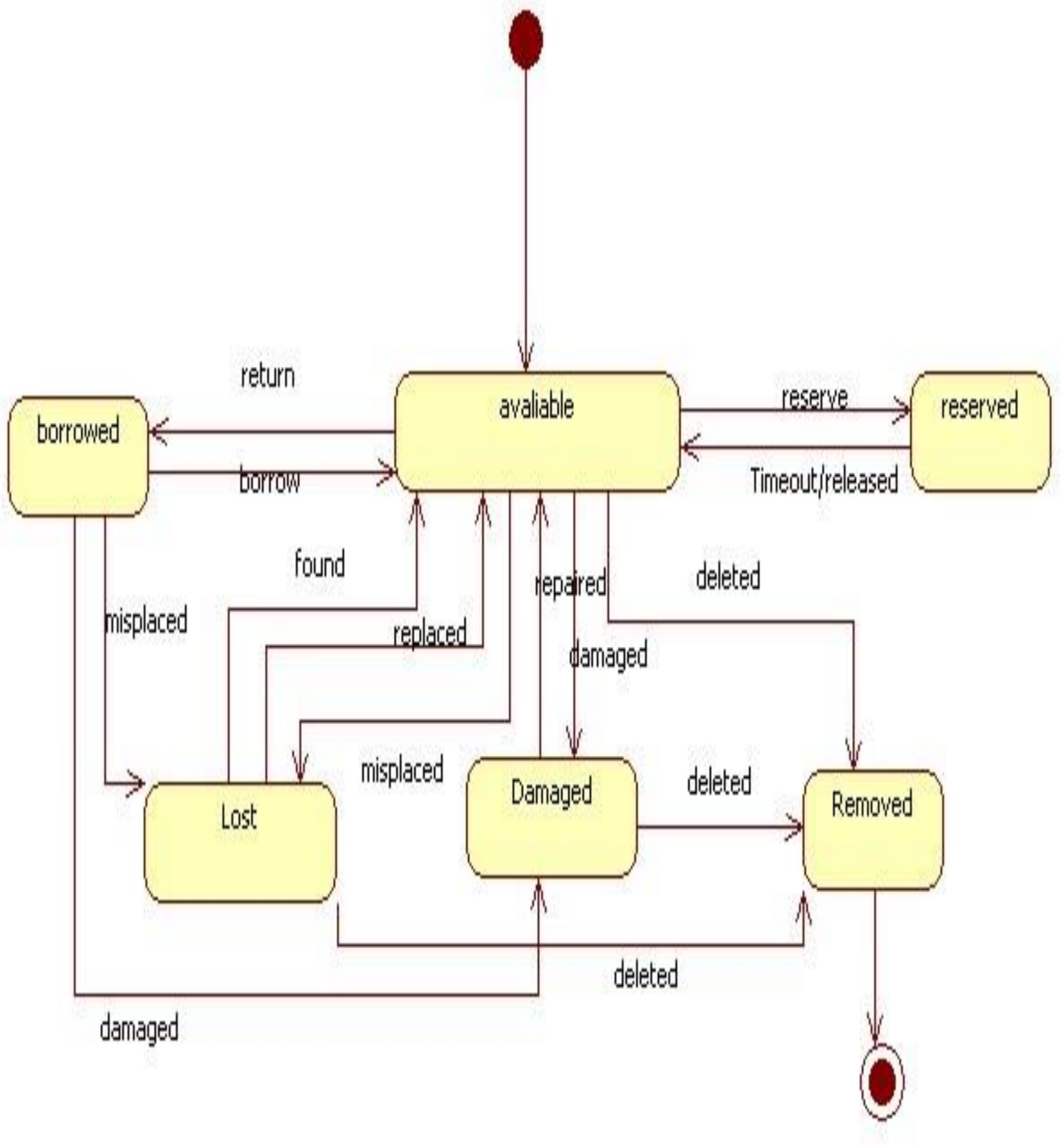
## Sequence Diagram – Return



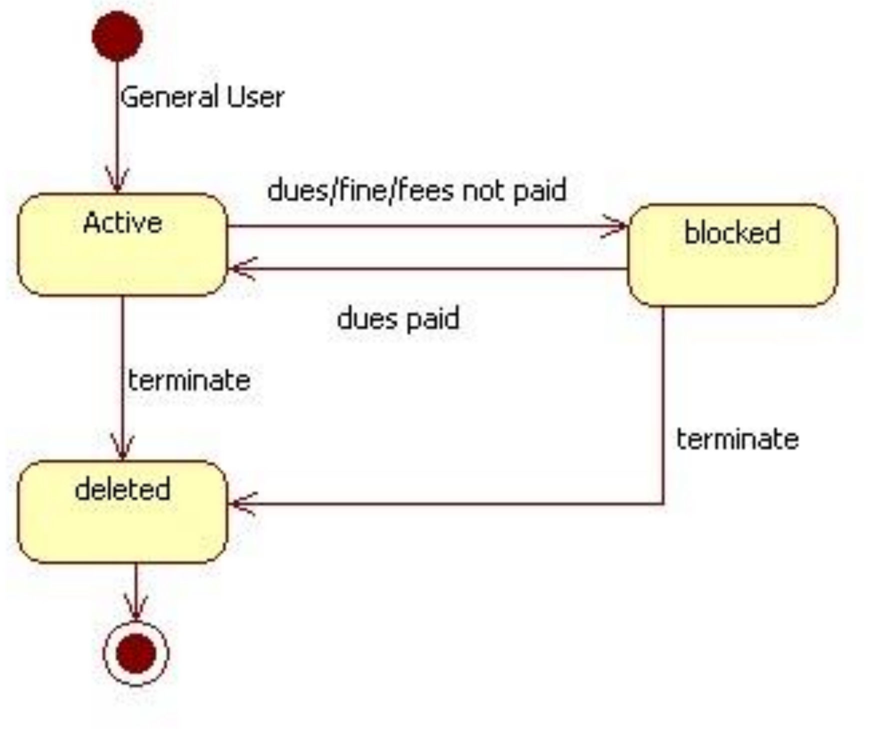
## Week 4 Program 5 : State Diagram : Item

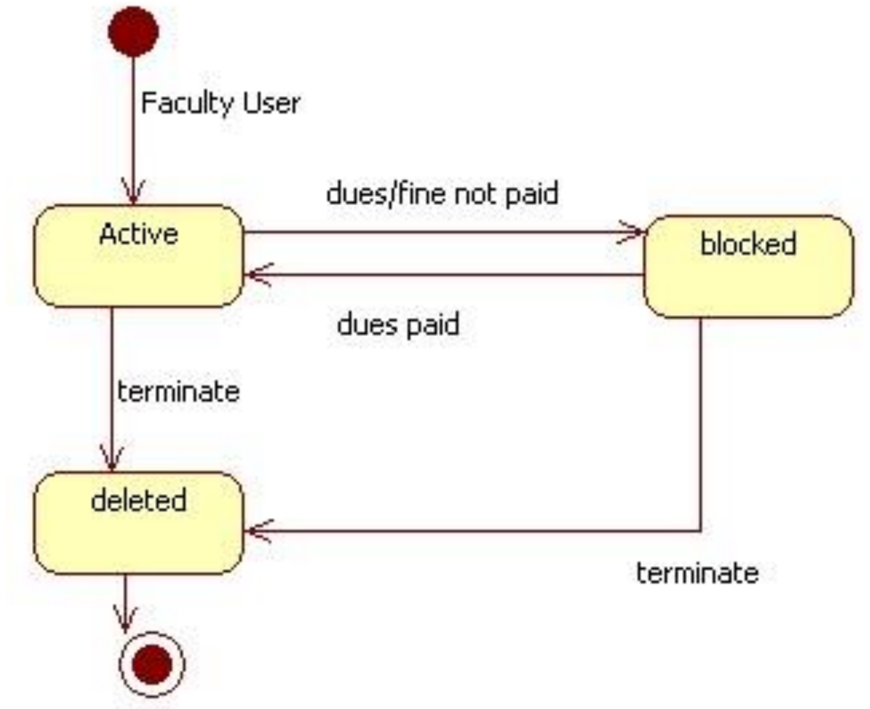
**Problem:** Prepare the State Diagram for the significant Classes

**Solution:**



## Week 5- Program – 6- State  Diagram : User

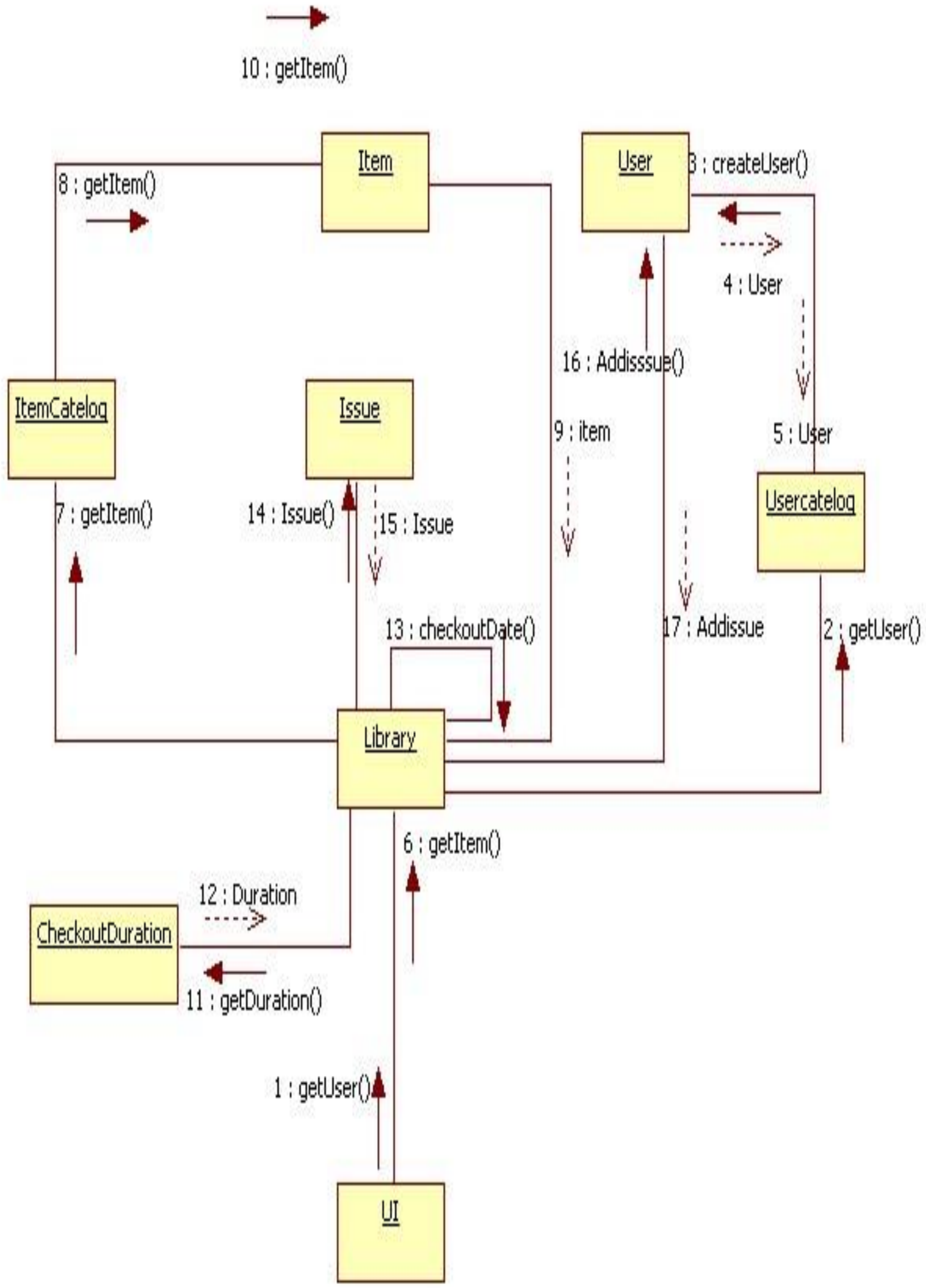




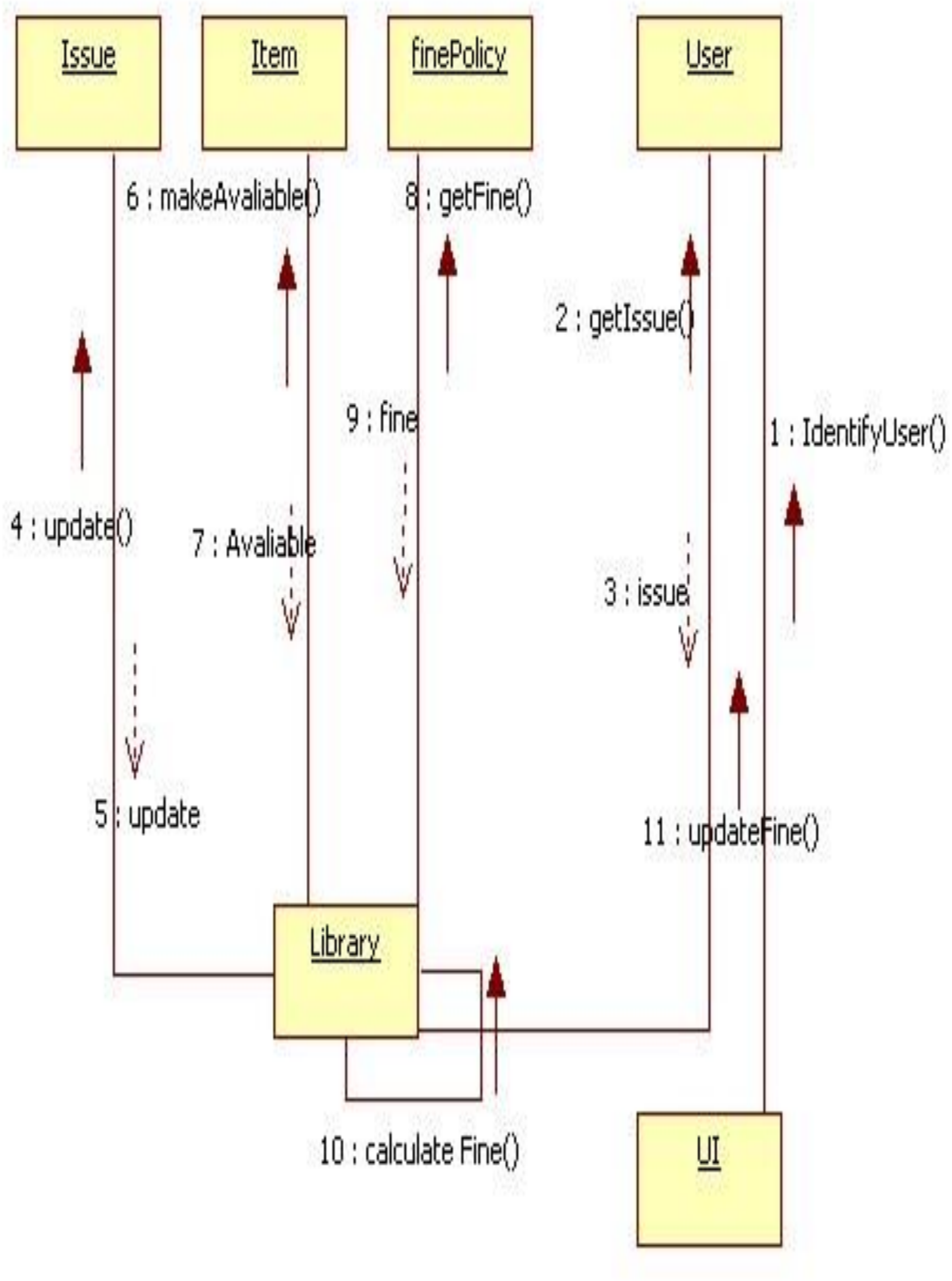
## Week 6 - Program 7 : Collaboration Diagram – Issue

**Problem:** Prepare the Collaboration Diagram for the significant Use Cases

**Solution:**



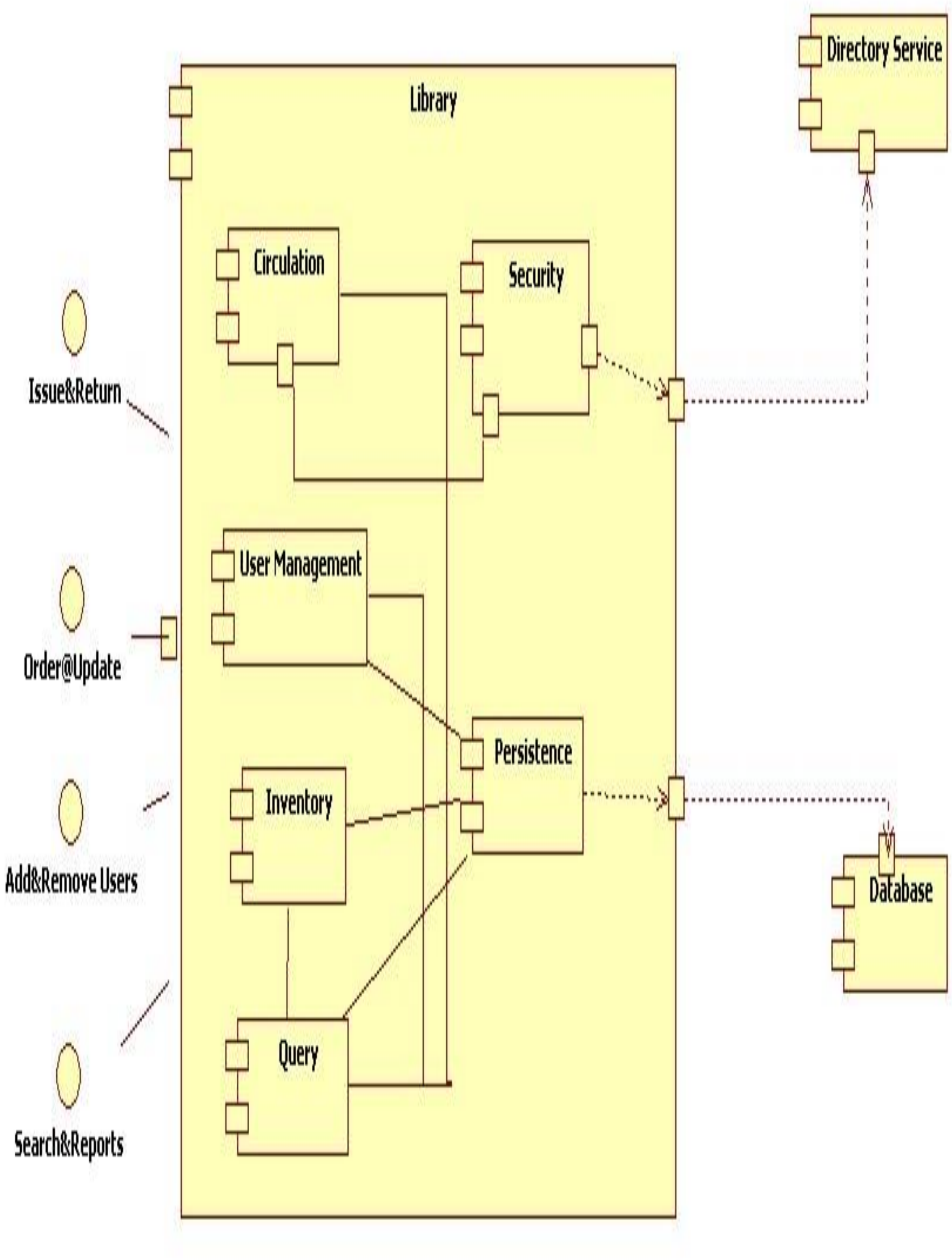
## Collaboration Diagram – Return



## Week 7- Program 8 : Component Diagram

**Problem:** Prepare the Component Diagram

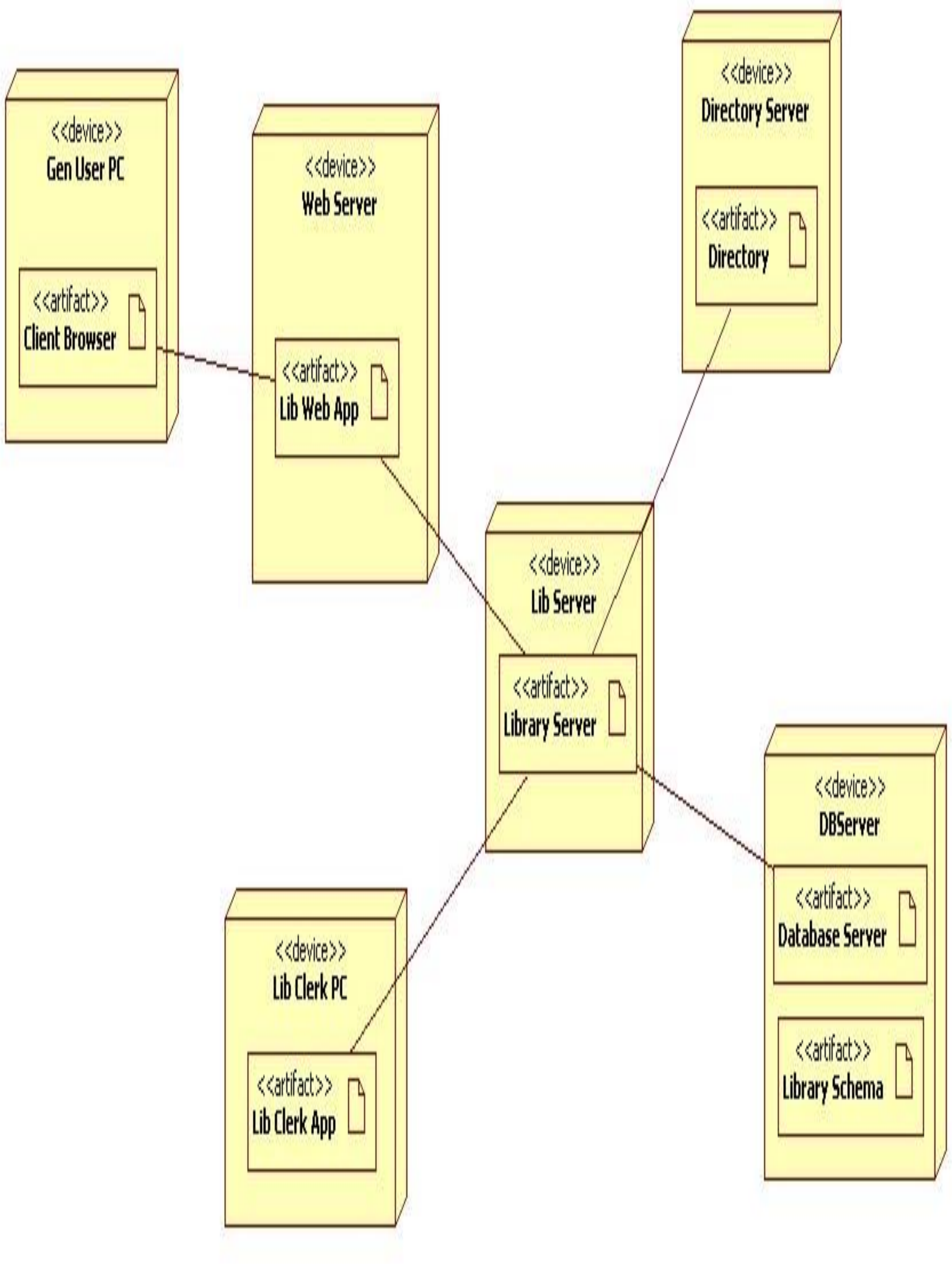
**Solution:**



# Week 7 –Program 9 : Deployment Diagram

**Problem:** Prepare the Deployment Diagram

**Solution:**



## Week 8: Application of SOLID and GRASP Principles

GRASP (General Responsibility Assignment Software Patterns) is an acronym created by Craig Larmanto encompass nine object-oriented design principles related to creating responsibilities for classes**.**

Creator
•Problem: Who should be responsible for creating a new instance of s ome class?if done poorly, this choice can affect coupling, clarity, enca psulation, and reusability.

•Solution: Assign class B the responsibility to create an instance of cl ass A if one of the below is true (the more the better). If more than on e option applies, usually prefer a classB which aggregates or containA . –B contains or is composed of A.
–B records A.
–B closely uses A. –
B has the initializing data for A that will be passed to A when it is created.
•Thus B is an Expert with respect to creating A.

Information Expert
•Problem: What is a general principle of assigning responsibilities to objects?
•Solution: Assign a responsibility to the information expert—
the class that has the information necessary
to fulfill the responsibility.

Single-responsibility Principle

S.R.P for short - this principle states that: A class should have one and only one reason to change, meaning that a class should have only one job.

The single responsibility principle is a computer programming principle that states that every module or class should have responsibility over a single part of the functionality provided by the software, and that responsibility should be entirely encapsulated by the class. All its services should be narrowly aligned with that responsibility, that is "A class should have only one reason to change

Week 9-12 Mini-Project on Design Patterns

**- Pro** ITEM _ TYPE_NAME CHAR (20))