Software Engineering Practicals

1. Study and implementation of Use Case Diagrams.

Aim :

- Aim to Implement of Use Case Diagrams.It serves as a crucial step in software development, aiding in the visualization and communication of system requirements.
- The primary aim is to identify actors, define use cases, establish relationships, and create clear diagrams that accurately represent system functionality.

Use_Case Diagram:

The book bank use cases are:

1. book_issue

2. book_return

3. book_order

4. book_entry

5. search book_details

Actors Involved:

1. Student

2. Librarian

3. Vendor

Usecase Name : Search Book_Details The librarian initiates this use case when any member returns or request the book and checking if the book is available.

Precondition: The librarian should enter all Book details.

Normal Flow: Build message for librarian who search the book.

Post Condition: Send message to respective member who reserved the book.

Usecase Name : Book_Issue Initiated by librarian when any member wants to borrow the desired book. If the book is available, the book is issued. Precondition: Member should be valid member of library.

Normal Flow: Selected book will be issued to the member. Alternative Flow: If book is not available then reserved book use case should be initiate.

Post Condition: Update the catalogue.

Usecase Name : Book_Order Initiated by librarian when the requested book is not available in the library at that moment. The book is reserved for the future and issued to the person when it is available.

**Precondition:** Initiated only when book is not available. Normal Flow: It reserved the book if requested.

Post Condition : Mention the entry in catalogue for reservation.

Usecase Name : Book_Return Invoked by the librarian when a member returns the book.

Precondition: Member should be valid member of library.

Normal Flow: Librarian enters bookid and system checks for return date of the book.

Alternative Flow: System checks for return date and if it returned late fine message will be displayed. Post Condition: Check the status of reservation.
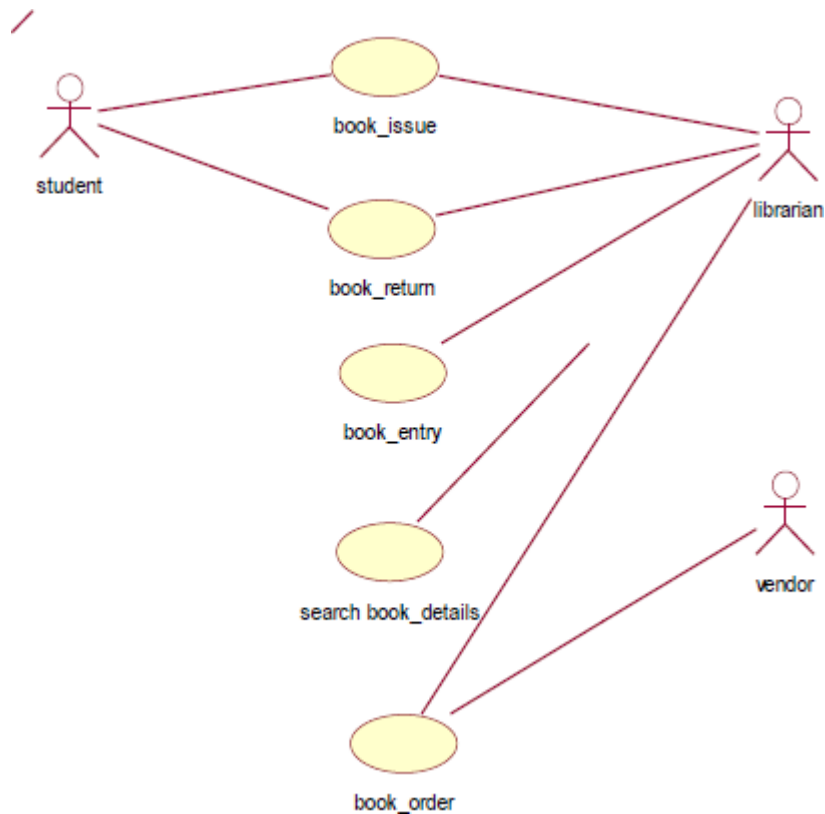
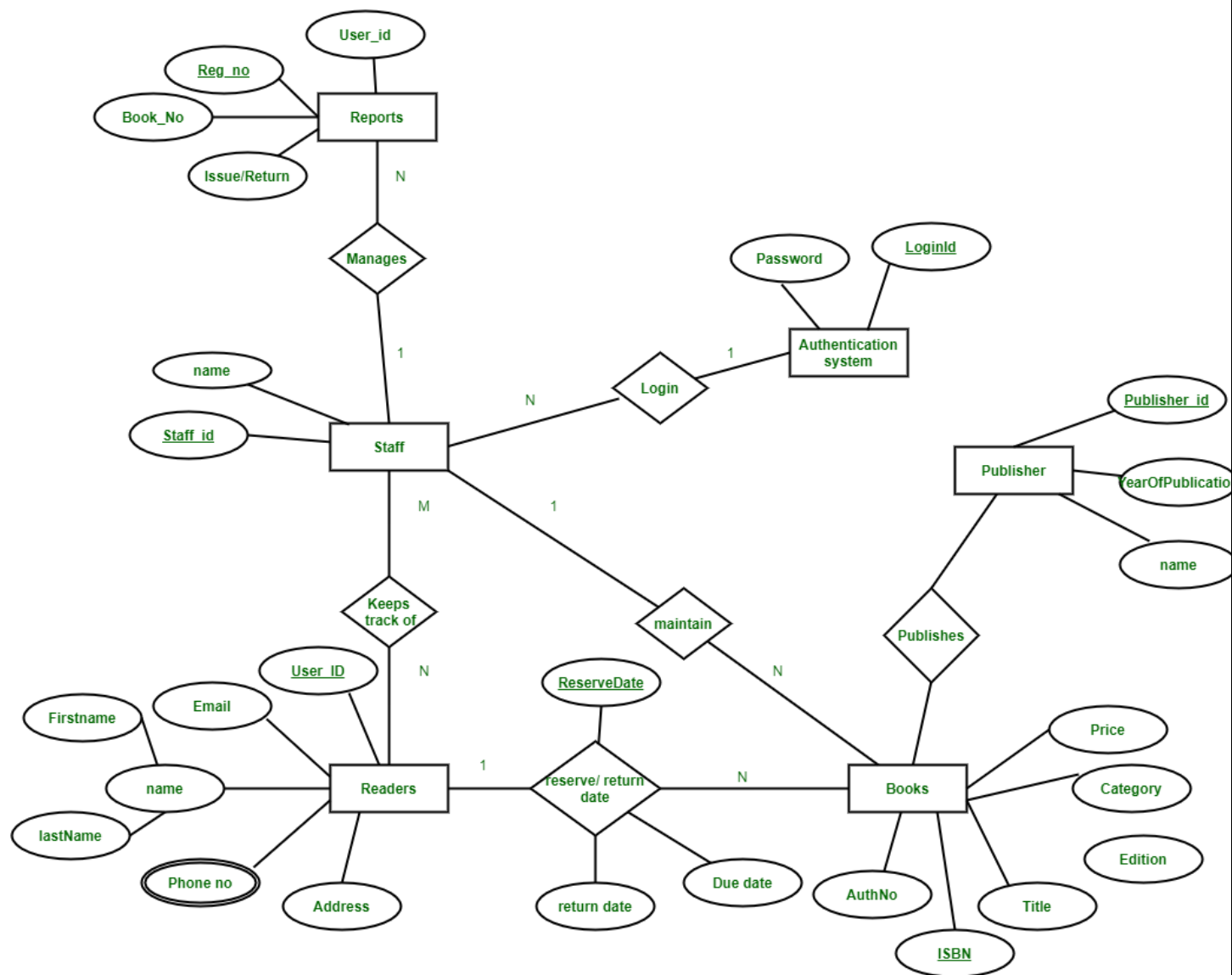2. **Study and implementation of Entity Relationship Diagrams.**

AIM :

- To Draw Entity Relationship Diagrams
- ER diagram is used to analyze the structure of the Database. It shows relationships between entities and their attributes.
- An ER Model provides a means of communication.

Entities and their Attributes –
- **Book Entity :** It has authno, isbn number, title, edition, category, price. ISBN is the Primary Key for Book Entity.
- **Reader Entity :** It has UserId, Email, address, phone no, name. Name is composite attribute of firstname and lastname. Phone no is multi valued attribute. UserId is the Primary Key for Readers entity.
- **Publisher Entity :** It has PublisherId, Year of publication, name. PublisherID is the Primary Key.
- **Authentication System Entity :** It has LoginId and password with LoginID as Primary Key.
- **Reports Entity :** It has UserId, Reg_no, Book_no, Issue/Return date. Reg_no is the Primary Key of reports entity.
- **Staff Entity :** It has name and staff_id with staff_id as Primary Key.
- **Reserve/Return Relationship Set :** It has three attributes: Reserve date, Due date, Return date.
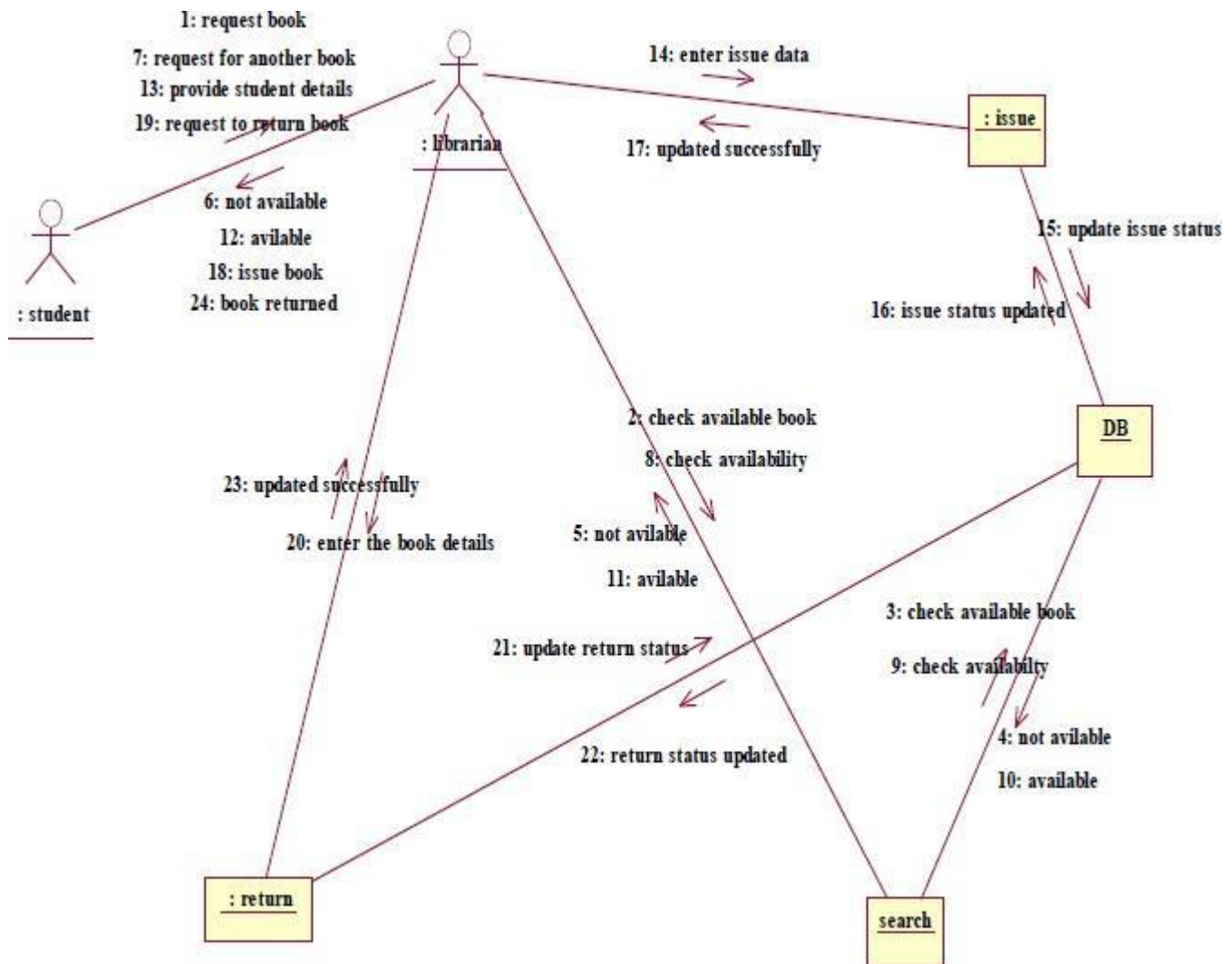
Relationships between Entities –
- A reader can reserve N books but one book can be reserved by only one reader. The relationship 1:N.
- A publisher can publish many books but a book is published by only one publisher. The relationship 1:N.
- Staff keeps track of readers. The relationship is M:N.
- Staff maintains multiple reports. The relationship 1:N.
- Staff maintains multiple Books. The relationship 1:N.
- Authentication system provides login to multiple staffs. The relation is 1:N.

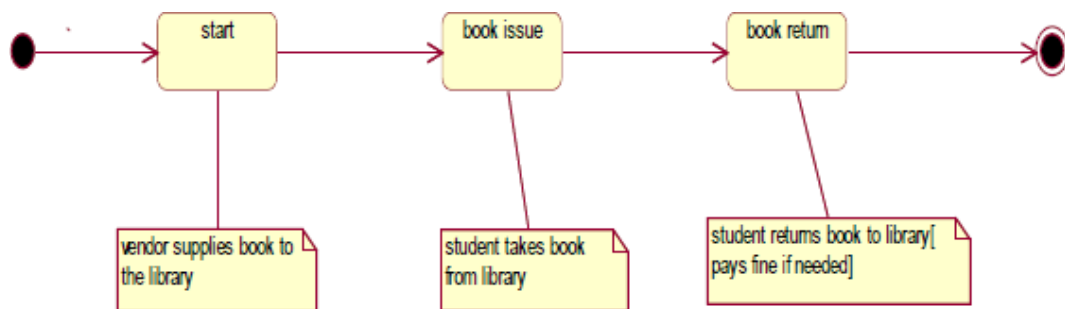## 3. Study and implementation of Sequence Diagrams.
Aim:
- To design a Sequence diagram for Book Bank Management System
- To show how objects in a system work together over time.

1: request book
7: request for another book
13: provide student details
19: request to return book

14: enter issue data

: issue

17: updated successfully

: librarian

6: not available
12: avilable
18: issue book
24: book returned

15: update issue status

: student

16: issue status updated

2: check available book
8: check availability

DB

23: updated successfully

20: enter the book details

5: not avilable

11: avilable

3: check available book

9: check availabilty

21: update return status

22: return status updated

4: not avilable

10: available

: return

search

4. Study and implementation of State Transition Diagrams.

  Aim:

- **To draw State Transition Diagrams**

- It consists of state, events and activities.

- State diagrams are a familiar technique to describe the behavior of a system.

- To describe all of the possible states that a particular object can get into and how the object's state changes as a result of events that reach the object.

5. Study and implementation of Data Flow Diagrams.

- To draw Data Flow Diagram (DFD)
- To depicts the flow of information and the transformation applied when data moves in and out of a system.
- To describe overall system is represented and described using input, processing, and output in the DFD.

The inputs can be:
- **Book request** when a student requests for a book.
- **Library card** when the student has to show or submit his/her identity as proof.

The overall processing unit will contain the following output that a system will produce or generate:
- The book will be the output as the book demanded by the students will be given to them.
- Information on the demanded book should be displayed by the library information system that can be used by the student while selecting the book which makes it easier for the student.

1. **Level 0 DFD –**
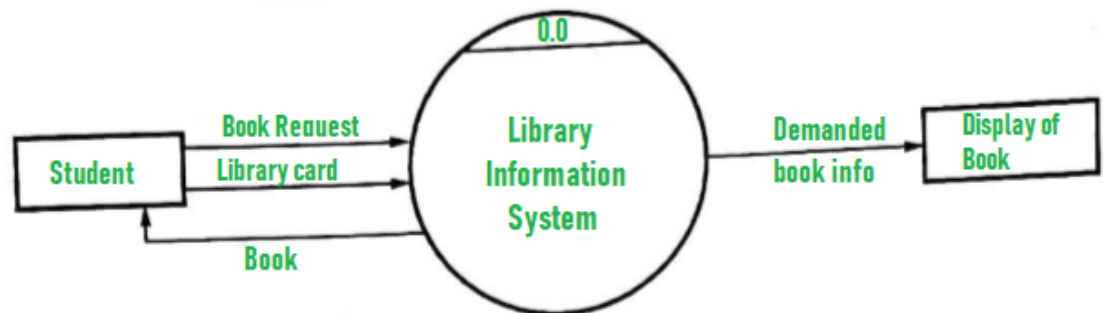
2. **Level 1 DFD –** At this level, the system has to show or exposed with more details of processing. The processes that are important to be carried out are:
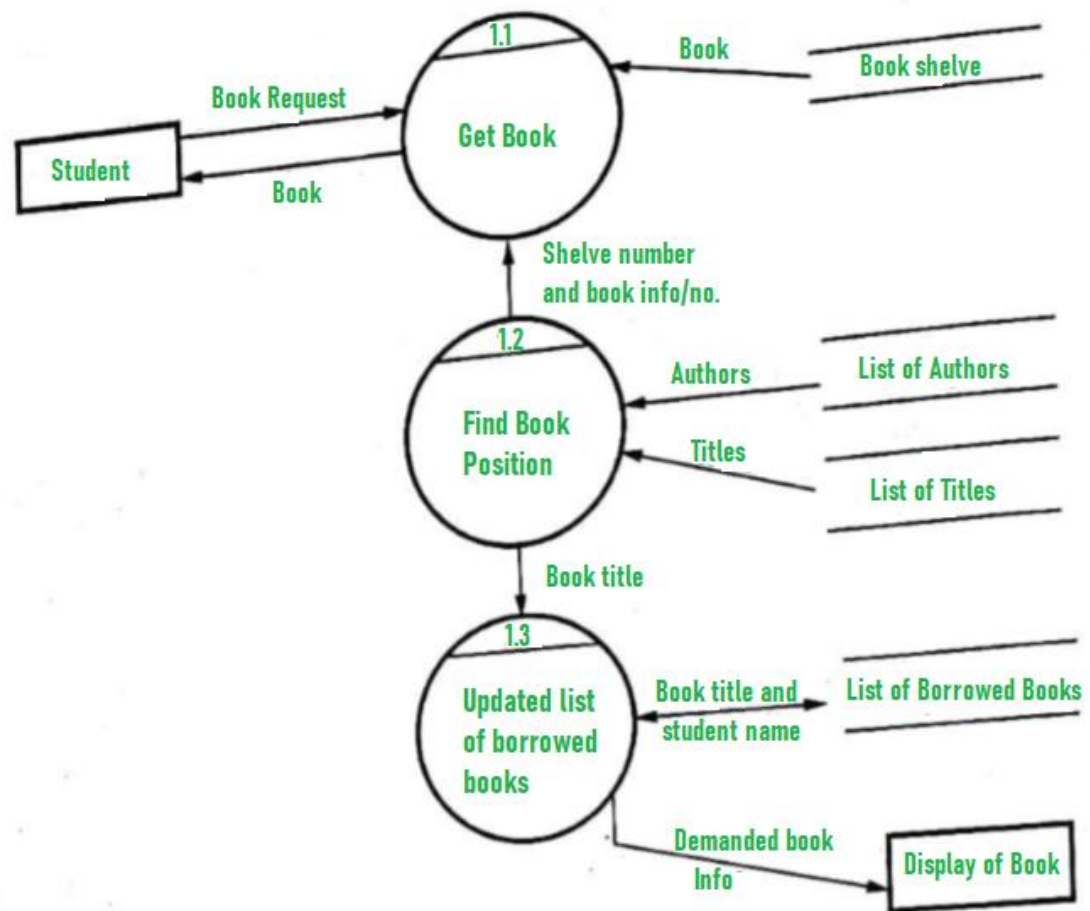   - Book delivery
   - Search by topic

List of authors, List of Titles, List of Topics, the bookshelves from which books can be located are some information that is required for these processes. **Data store** is used to represent this type of information.
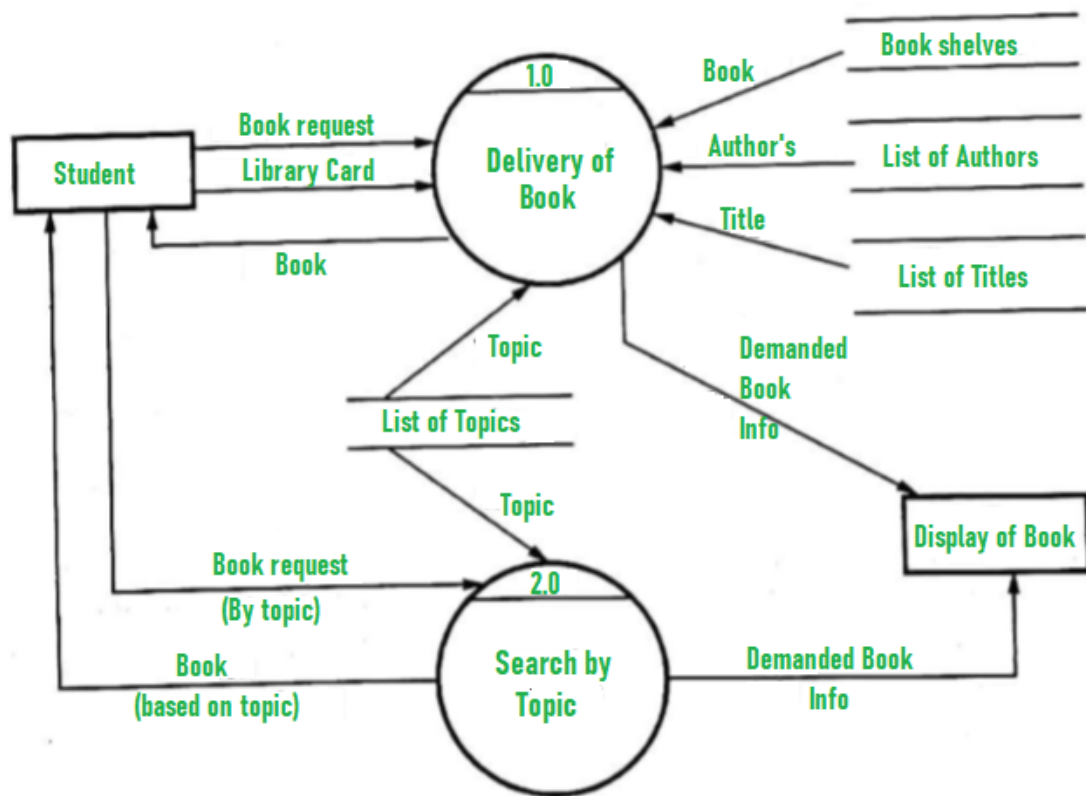
Level 1 DFD

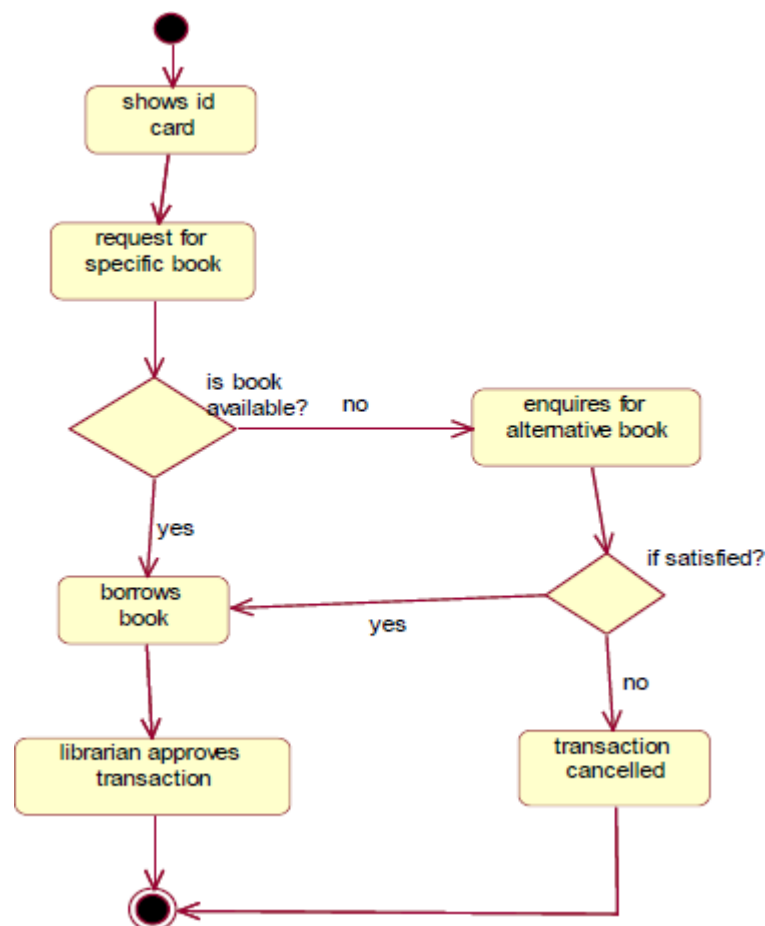1. **Level 2 DFD** –

Level 2 DFD

6. Study and implementation of Activity Diagrams.

Aim:

- To design a Sequence diagram for Book Bank Management System

Activity Diagram:

Activity diagrams are graphical representations of workflows of stepwise activities and actions with support for choice, iteration and concurrency. In the Unified Modeling Language, activity diagrams an be used to describe the business and operational step-by-step workflows of components in a system. An activity diagram shows the overall flow of control. An activity is shown as an rounded box containing the name of the operation. This activity diagram describes the behaviour of the system.

Activity Diagram:

Activity diagrams are graphical representations of workflows of stepwise

activities and actions with

support for choice, iteration and concurrency. In the Unified Modeling Language, activity diagrams can be used to describe the business and operational step-by-step workflows of components in a

system. An activity diagram shows the overall flow of control. An activity is shown as an rounded box containing the name of the operation.
This activity diagram describes the behaviour of the system.

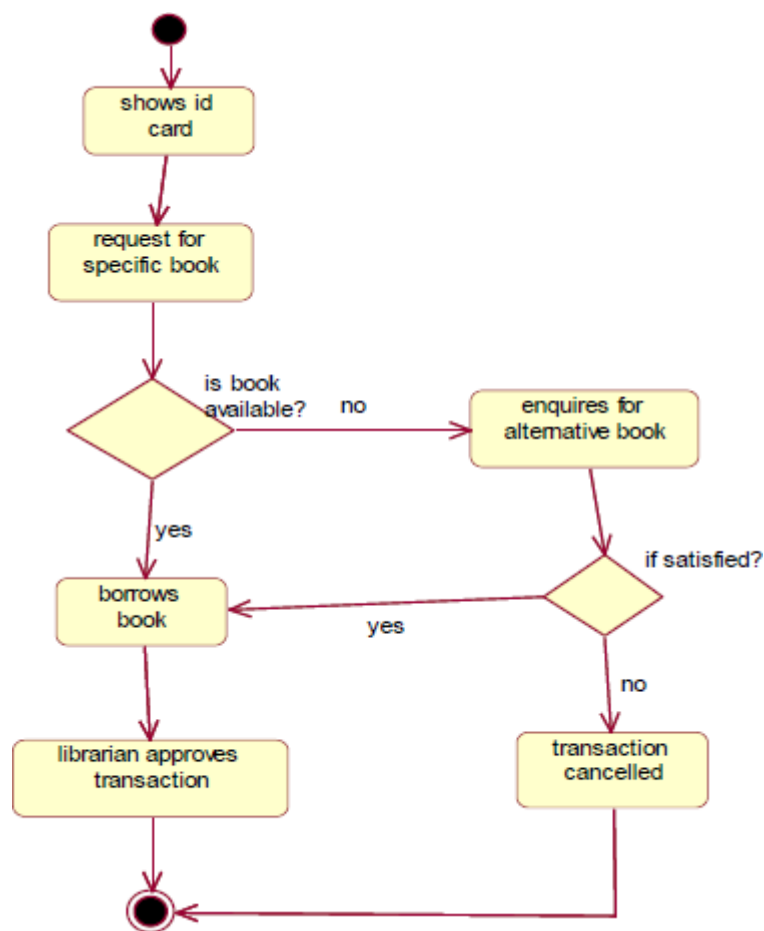## 7. Study and implementation of Class Diagrams.
Aim: To draw Class Diagrams

Class Diagram:

The class diagram, also referred to as object modeling is the main static analysis diagram. The main task of object modeling is to graphically show what each object will do in the problem domain. The problem domain describes the structure and the relationships among objects.

The ATM system class diagram consists of four classes:

1. Student

2. Book

3. Issue

4. Return

5. Vendor

6. Details

1) *Student:*

It consists of twelve attributes and three operations. The attributes are enrollno, name, DOB, fathername, address, dept name, batch and book limits. The operations of this class are addStInfo(), deleteStInfo(), modifyStInfo().

2) *Book:*

It consists of ten attributes and four operations. This class is used to keep book information such as author, title, vendor, price, etc

3) *Issue:*

It consists of eight attributes and two operations to maintain issue details such as, issue date, accno of issued book, name of the student who borrowed book.

4) *Return:*

It consists of eight attributes and two operations to maintain issue details such as,
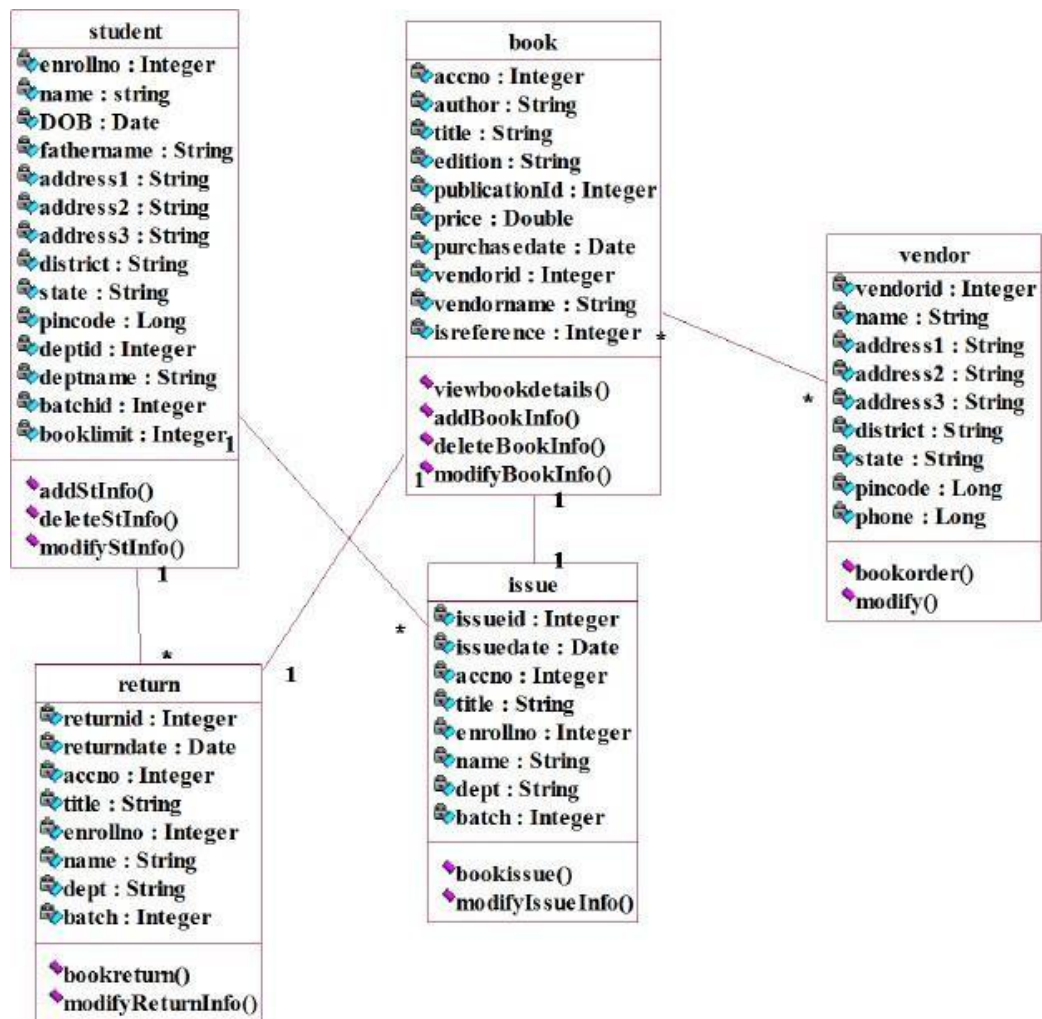
issue date, accno of issued book, name of the student who borrowed book.

5) *Students:*

The attributes of this class are name, dept ,year ,bcode no The operation is display students().

6) *Detail:*

The attributes of this class are book name, author, bcode no The operations are delete details().

student
- enrollno : Integer
- name : string
- DOB : Date
- fathername : String
- address1 : String
- address2 : String
- address3 : String
- district : String
- state : String
- pincode : Long
- deptid : Integer
- deptname : String
- batchid : Integer
- booklimit : Integer

- addStInfo()
- deleteStInfo()
- modifyStInfo()

book
- accno : Integer
- author : String
- title : String
- edition : String
- publicationId : Integer
- price : Double
- purchasedate : Date
- vendorid : Integer
- vendorname : String
- isreference : Integer

- viewbookdetails()
- addBookInfo()
- deleteBookInfo()
- modifyBookInfo()

vendor
- vendorid : Integer
- name : String
- address1 : String
- address2 : String
- address3 : String
- district : String
- state : String
- pincode : Long
- phone : Long

- bookorder()
- modify()

issue
- issueid : Integer
- issuedate : Date
- accno : Integer
- title : String
- enrollno : Integer
- name : String
- dept : String
- batch : Integer

- bookissue()
- modifyIssueInfo()

return
- returnid : Integer
- returndate : Date
- accno : Integer
- title : String
- enrollno : Integer
- name : String
- dept : String
- batch : Integer

- bookreturn()
- modifyReturnInfo()

8. Study and implementation of Component Diagrams.

Aim:

To **implement of Component Diagrams**

Component Diagram

Deployment diagrams are used to visualize the topology of the physical components of a system where the software components are deployed.