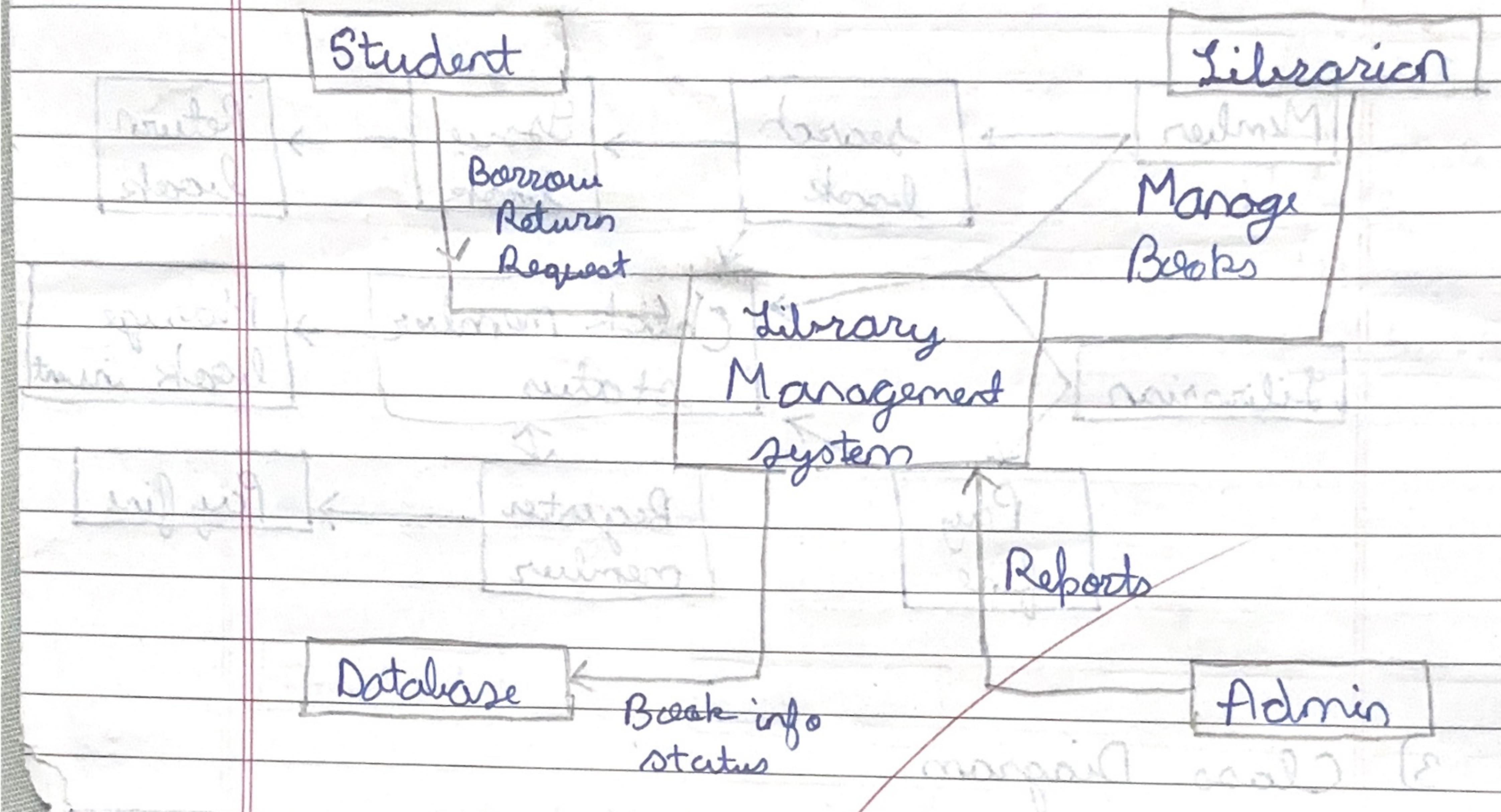
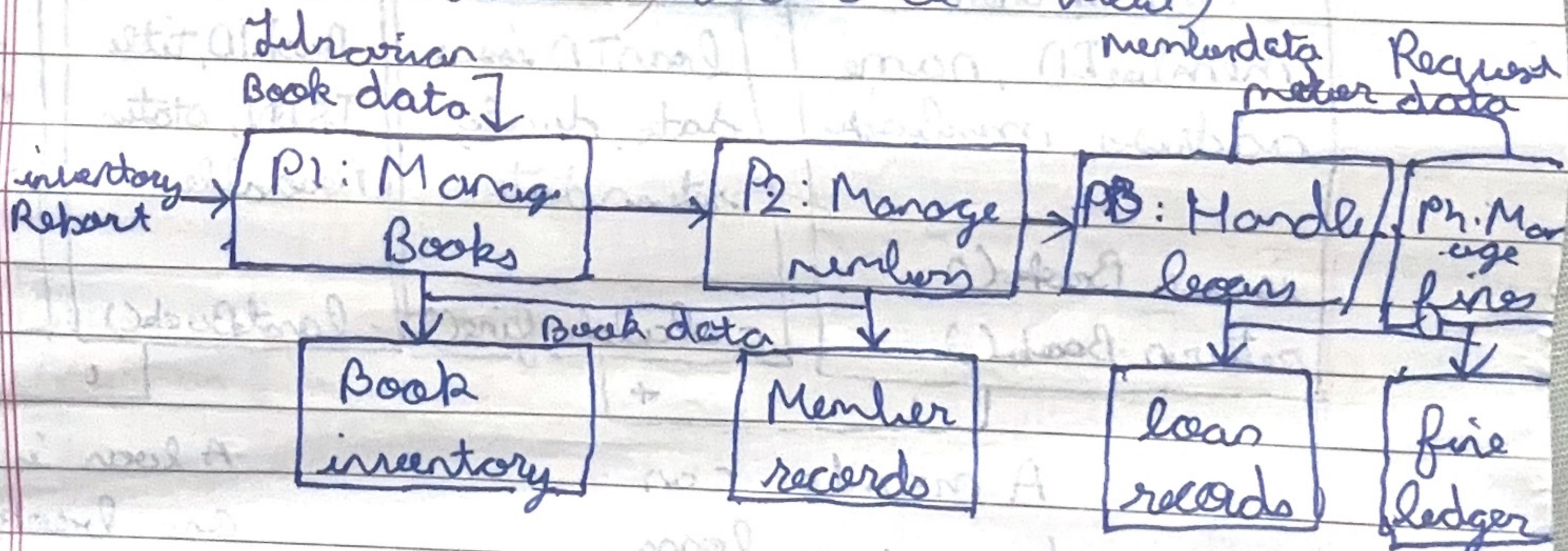


## MSET Assignment 3A

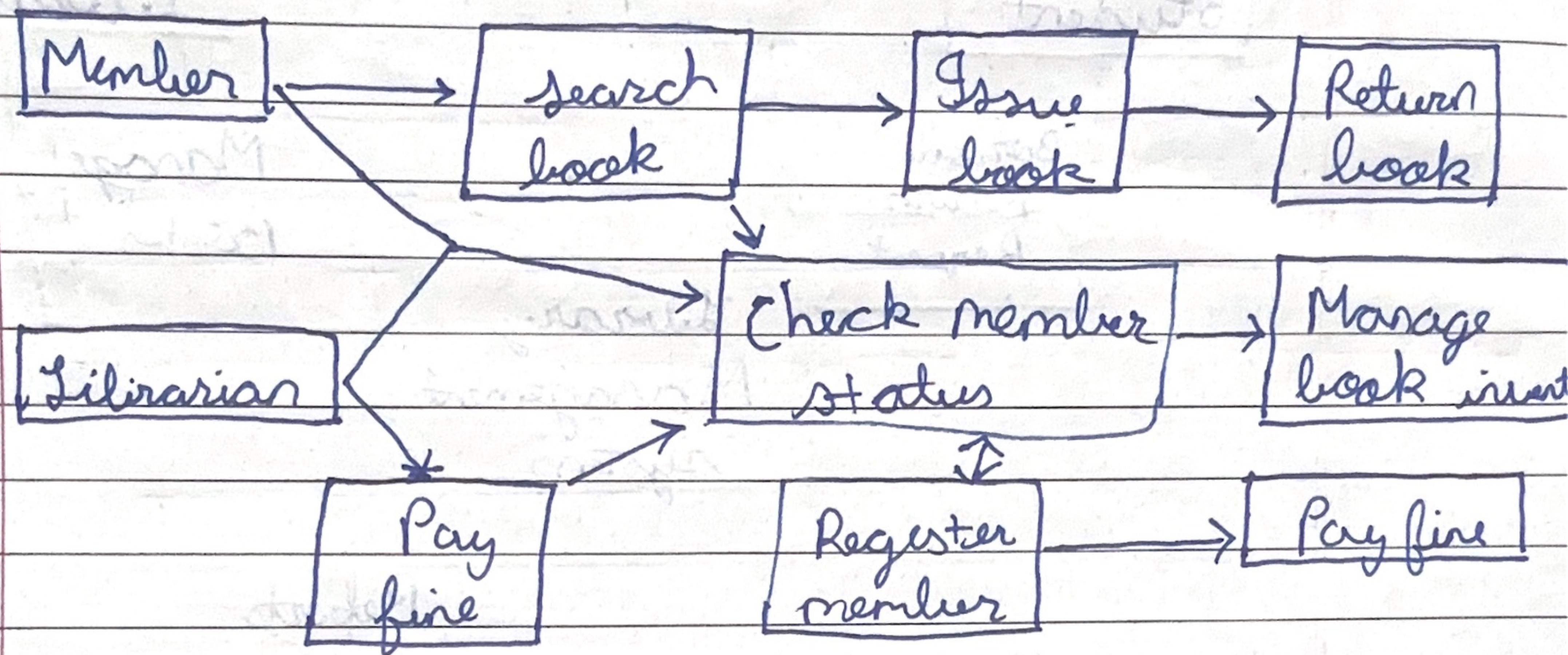
### 1] Level 0 DFD (Context Diagram)



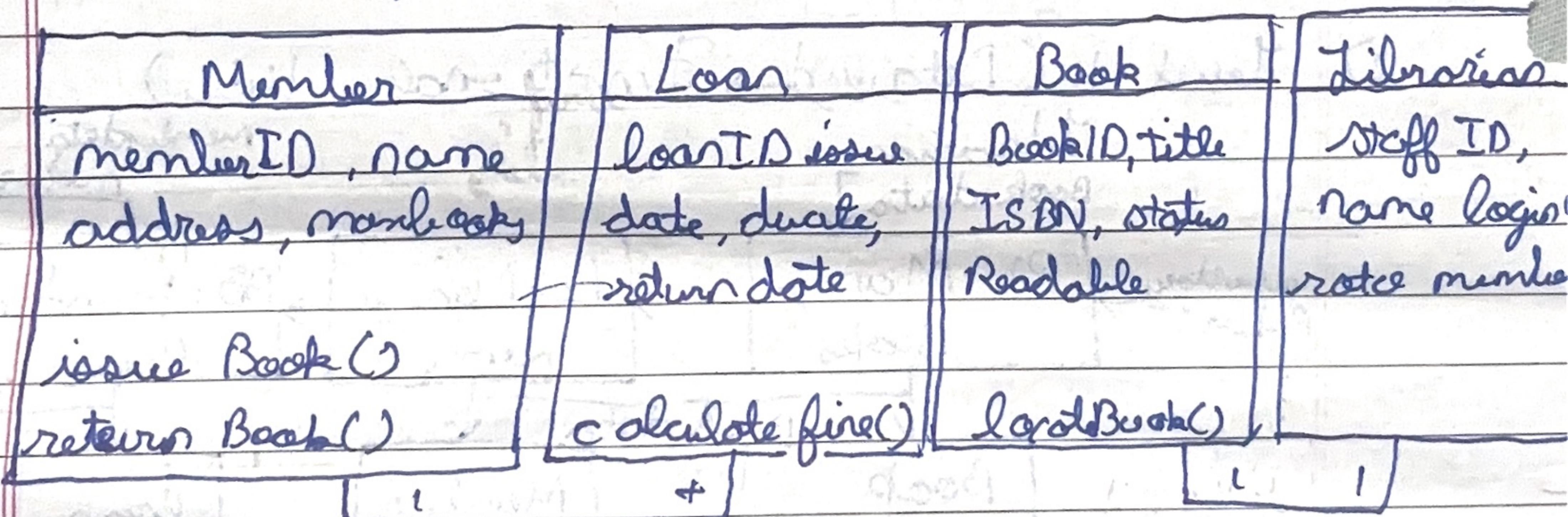
### 2) Level 1 (Detailed Functional View)



## 2) Use Case Diagram



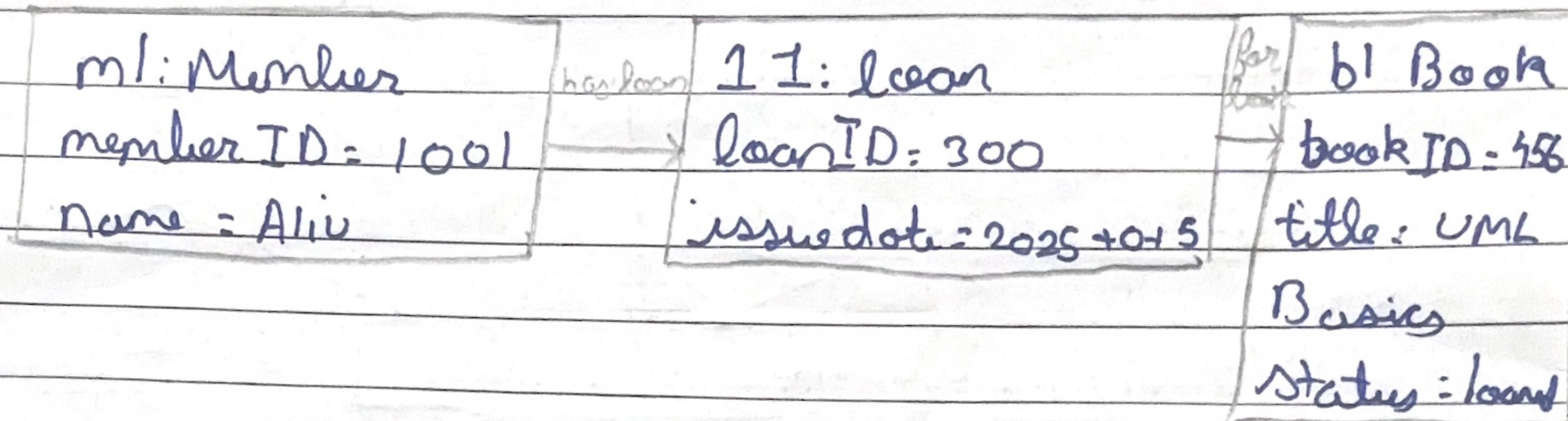
## 3) Class Diagram



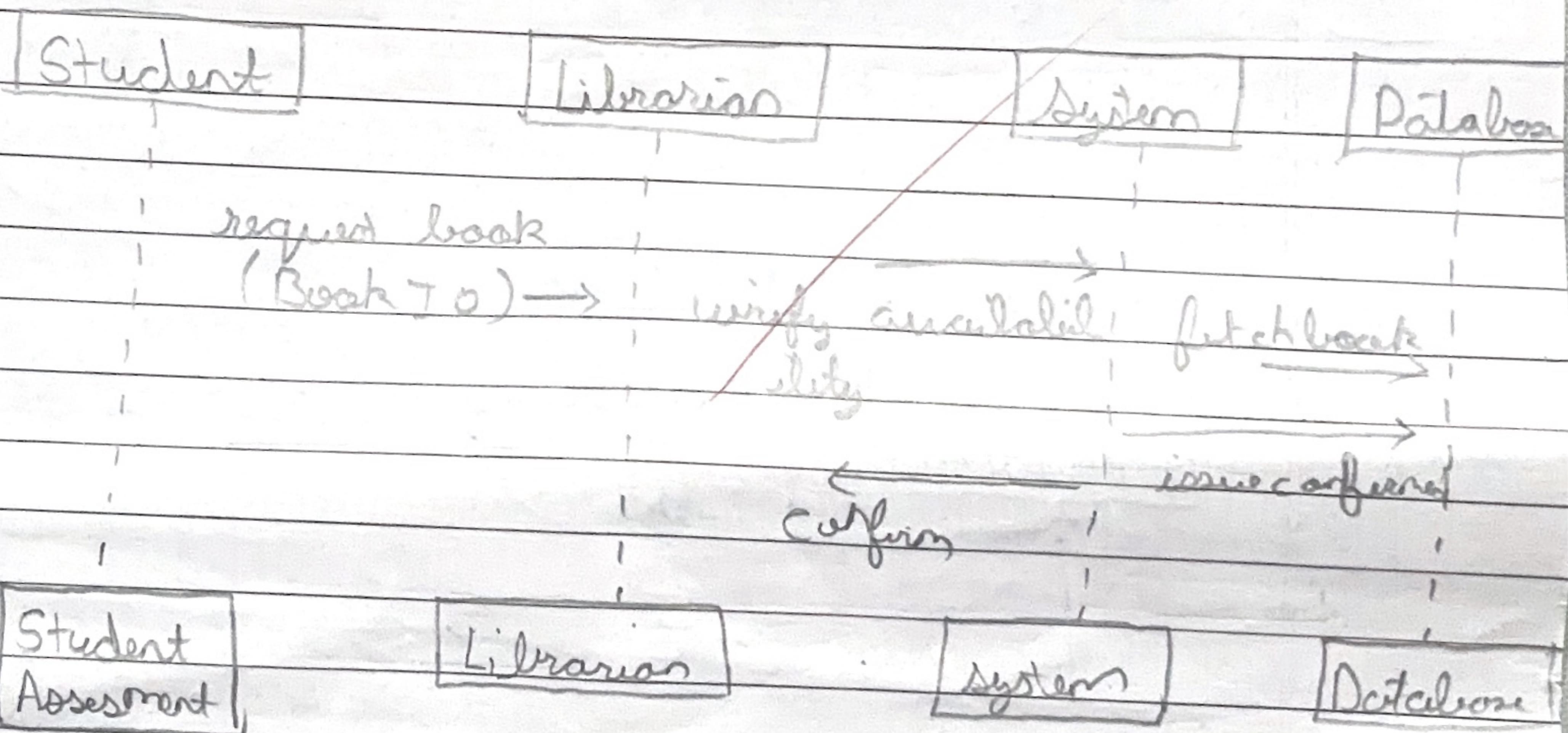
A member can have many loans

A loan is for one book

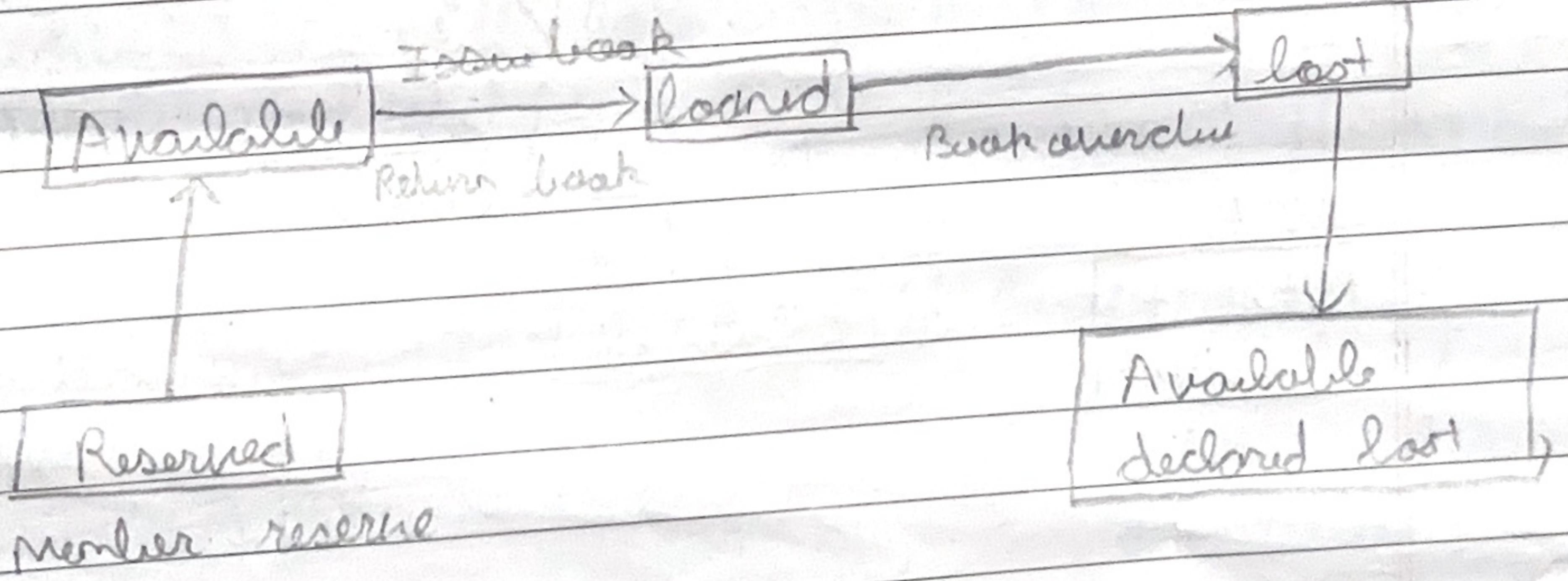
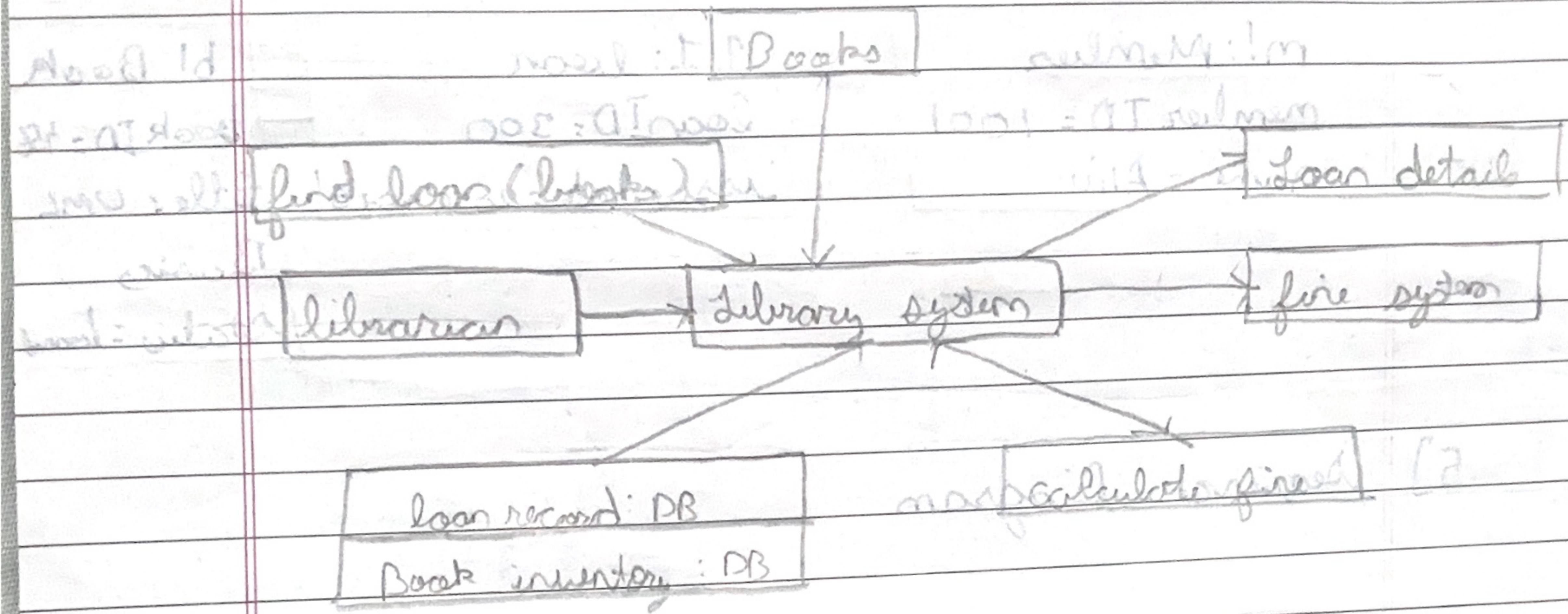
## 4] Object Diagram



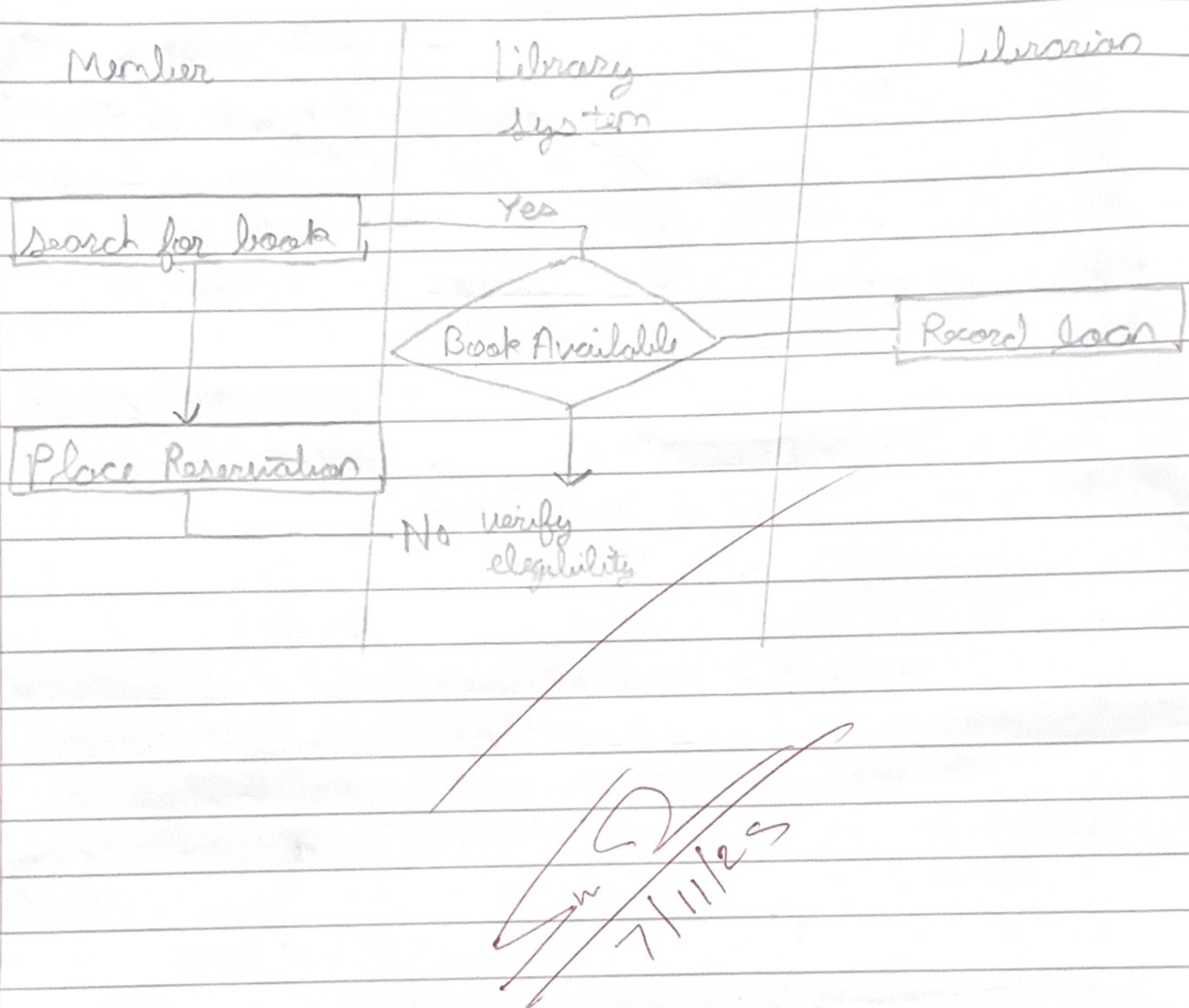
## 5] Sequence Diagram



## 6] Communication Diagram



## 8) Activity Diagram



## Mset Assignment

- 03(B)

- (1) Write a case study on requirement engineering and modelling including
- a) Topics, introduction, objective, phases, types, modeling techniques and OOM (object oriented model diagram).

Topic: Requirement engineering and modelling introduction

RE is the process of identifying, analyzing, documenting and validating the needs and expectations of stakeholders for a software system. It ensures that final product meets user requirements and performs the desired functions effectively.

Objectives -

- i. To understand the needs of users and stakeholders.
- ii. To define clear and complete functional and non-functional requirements.
- iii. To reduce development cost and avoid rework.
- iv. To provide a basis for system analysis, design and testing.
- v. To ensure the developed system meets business goals.

Phases -

- i. Requirement Elicitation  
gathering requirements through interviews, questionnaires etc.

i. 2. Requirements Analysis -  
analyzing collected requirements for feasibility  
and consistency.

iii. Requirements Specification -  
documenting the requirements in Software  
Requirement Specification.

iv. Requirements Validation -  
Ensuring correctness and verifying with  
stakeholders.

v. Requirements Management -  
Handling changes in requirements during  
development.

Types -

i. Functional Requirements -  
describes what the system should do (login).

ii. Non-functional Requirements -  
Describe system quality attributes (security,  
performance).

Modelling Techniques -

i. Structural - (DFD)  
Shows how data flows in the system.

ii. Behavioral - (Use Case Diagram)  
Shows interactions and operations.

iii. Architectural - (Deployment Diagram)  
Shows physical structure.

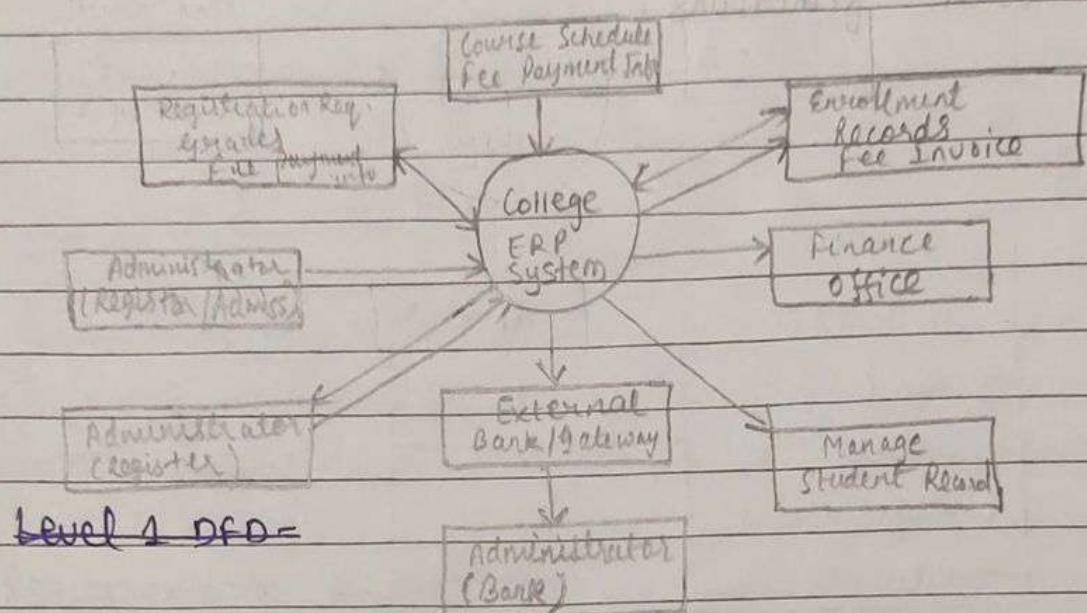
## DOM -

DOM represents the system in terms of objects, which are entities with attributes and functions.

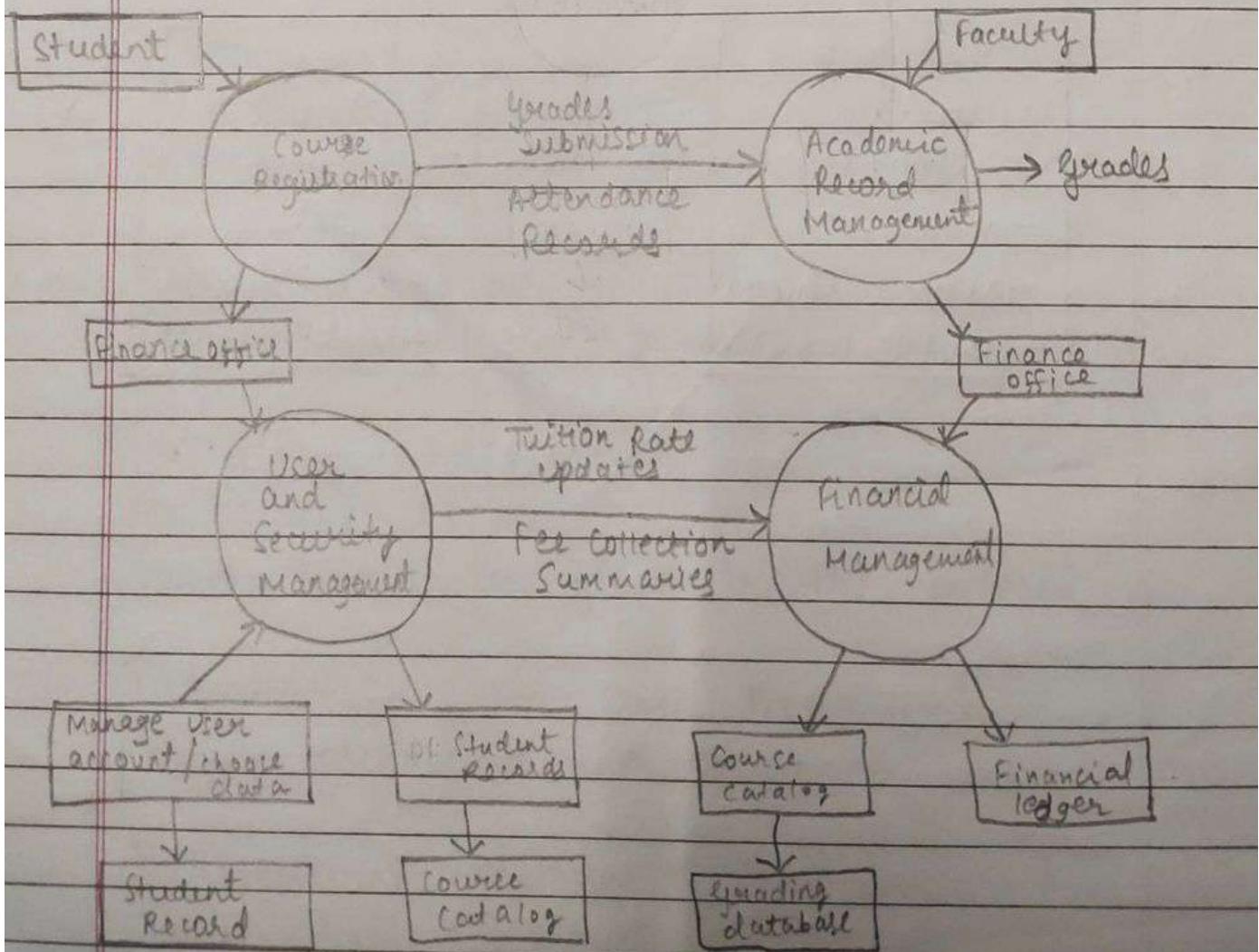
Student		fee
- StudentID		- feeID
- name		- amount
- course		- status
+ register()		+ payFee()
+ viewResult()		+ viewReceipt()

Attendance	Exam
- attID	- examID
- date	- marks
- status	- grade
+ markAttendance()	+ recordMarks()
+ viewAttendance()	+ viewResult()

b) College ERP system -  
Level 0 DFD - College Context Diagram -



Level 1 - DFD (College ERP Detailed Functional View)



## Assignment - 02

- Q1. Define requirement engineering. Explain its importance in software development life cycle.

A systematic and strict approach to the definition creation and verification of requirements for a software system is known as requirement engineering.

Importance in SDLC -

- i) It provides a clear understanding of what system should do, which helps in designing and developing the software accurately.
- ii) It reduces project failures by properly gathering and analyzing requirements.
- iii) It improves the communication by bridging the gap between stakeholders, developers and users by clearly defining expectation.
- iv) As when requirements are well defined, it ensures quality and user satisfaction.
- v) It also helps in project planning by estimating cost, effort and time.

2. What is feasibility study? Describe the 5 types with example.

A feasibility study is an analysis used to determine whether a proposed software project is practical and achievable. It helps decide if the project should proceed or not based on factors like time, cost, resources and technology.

Types -

i) Technical Feasibility -

Checks whether the required technology, tools, and resources are available to develop the system. For eg.,

If a project needs AI tools, it checks whether the organization has necessary hardware and software.

ii) Operational Feasibility -

It determines whether the system will work effectively in the existing environment and meet user requirements. For eg.,

Checking if employees can easily adopt to

iii) Economic Feasibility -

Evaluates whether the project is financially viable - whether benefits outweigh the costs.

For eg.,

Developing a new software system should save more money than it costs to build and maintain.

iv) Legal feasibility -

Ensures project follows all legal and regulatory requirements. For eg., A healthcare software must comply with patient data protection laws.

v) Schedule Feasibility -

checks whether the project can be completed within the required time frame.

For eg.,

Determining if a mobile app can be developed and tested before the product launch date.

Q3. Explain the term Requirement elicitation.

Requirement elicitation is the process of gathering and understanding the needs and expectations of user and stakeholders for a software system. It is first step in ~~in~~. It involves identifying stakeholders, gathering requirements, analyzing requirements, documenting and validating requirements.

Its common techniques are -

Interviews, surveys, focus groups, observation, prototyping, brainstorming / task analysis.

Q4. What is requirement specification? Describe its type in detail.

It is the process of documenting the user and system requirements in a clear, detailed and organized manner. It serves as a reference for designers, developers and testers.

Types -

i) Functional -

Define what the system should do - its functions and features.

For eg., the system shall allow user to login using a password.

ii) Non-Functional -

Define how the system performs its functions - quality attributes like performance, security, reliability, etc. For eg., the system should respond within two seconds.

iii) User-requirements -

High-level statements of what the user expect the system to do.

For eg., the user should be able to view previous orders.

iv) System Requirements -

Detailed description of system functionalities, hardware and software specifications.

For eg., system shall support windows and Linux platforms.

- Q5. Difference between verification and validation and mention two techniques used for each.

Verification

Focus: Are requirements correctly specified?

Techniques: Reviews, walk-throughs, test cases

Goal: It checks whether the product is built correctly as per requirements and design.

It is performed during development phase.

Validation

Does software meet user needs?

Prototyping, simulation, testing.

It checks whether the right product is built that meets user needs.

It is performed during testing or after development.

The techniques used for verification is  
Reviews, walk throughs and for  
Validation - prototyping and simulation.

## Assignment - 04

1. Explain the different levels of testing: unit integration system and acceptance testing with examples.

### Unit Testing -

Its goal is to verify that individual components or units of code work correctly in isolation. Common tools include JUnit, Test, NUnit. For eg:

function calculateTotal(items) {  
 return

items.reduce((sum, item) =>

sum + (item.price \* item.quantity));  
 };

Testing a function that calculates the total price of items in a shopping cart.

### Integration Testing -

After individual units are tested, integration testing ensures that different modules work together correctly when combined.

This phase catches issues in module interfaces and data flow. Common tools include Selenium, TUnit, Postman. For eg: checking if the login module correctly passes user credentials to the dashboard, and whether the dashboard displays the right user information after successful authentication.

## System Testing -

Tests the entire application as a complete, integrated system. For eg., Verify an e-commerce application's complete order flow - from browsing products and adding to cart, through checkout and payment processing, to order confirmation.

## Acceptance Testing -

Validates software meets business requirements and is ready for delivery to end users.

It includes alpha testing and beta testing. For eg.,

to check if a registered user can successfully log in using a valid username and password. The system should verify the credentials and redirect the user to the dashboard.

2. Differentiate between SDLC and STLC with a neat diagram.

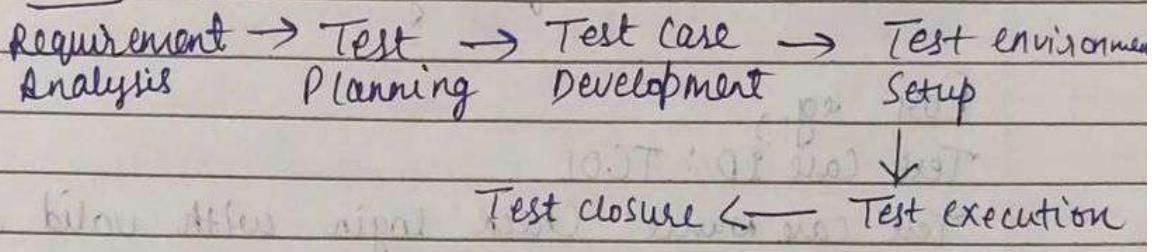
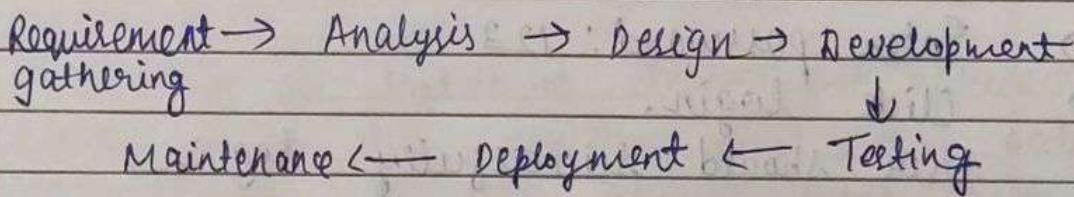
SDLC

STLC

- |   |  |
|---|--|
| i. A process that defines all phases of software development, from requirements gathering to test closure, to deployment & maintenance. | ii. It focuses on building the software.                 |
| i. A process that defines all phases of software testing, from requirement analysis to deployment & maintenance.                        | ii. It focuses on verifying and validating the software. |

- |   |  |
|---|--|
| i. A process that defines all phases of software development, from requirements gathering to test closure, to deployment & maintenance. | ii. It focuses on building the software.                 |
| i. A process that defines all phases of software testing, from requirement analysis to deployment & maintenance.                        | ii. It focuses on verifying and validating the software. |

- |      |  |  |
|------|--|--|
| iii. | Its objective is to deliver a working software product that meets user requirements. | iii. To ensure the product is defect-free and high quality before release. |
| iv.  | Covers the entire lifecycle of the software.   | iv. Part of SDLC, focused only on testing.                                 |

STLC -SDLC -

- Q. What is a Test Case? Explain its template and give an example.

## Test Case -

Specific steps & conditions designed to validate a particular function or feature works correctly.

Template -

Test Case ID - unique no. for identification.

Test case Name / Title - what functionality is being tested.

Module Name - Part of application tested.

Preconditions - requirements before executing test.

Test steps - Step by step procedure to perform

Test Data - i/p values used.

Expected result - what should happen.

Actual result - what actually happened.

Status - Pass / Fail.

Comments - notes if any.

For eg.,

Test Case ID : TC01

Test case name : Check login with valid details

Steps -

- Open login page
- Enter username : abc
- Enter password : 123
- Click login.

User should successfully log in.

4. What is Requirement Traceability Matrix (RTM)? Explain its purpose and give a sample format.

A RTM is a document used in software testing to map and track each requirement with its corresponding test cases.

Purpose-

- i. Ensures all requirements are tested.
- ii. Avoids missing functionality.
- iii. Helps track changes.
- iv. Links requirement → test case → defects.

Sample Format -

Req ID	Requirement	Test Case ID	Status
RQ01	Login	TCD1, TCD2	Pass
RQ02	Sign-up	TCD3	Pass

5. What is Test-Driven Development (TDD)? Explain the TDD cycle with an example.

TDD is a software development approach where tests are written first, then code is written to pass those tests, and finally the code is improved.

TDD follows three steps - Red → Green → Refactor

i. Red -

Write a test case → test fails (because code is not written yet).

## ii. Green:

Write minimum code required to make the test pass.

## iii. Refactor:

Improve the code → keep all tests passing.

a) → for eg.,  
Red -

write test: add(2, 3) should return 5.  
→ Test fails.

b) → Green - write code.

function add(a, b) { return a + b; }

→ Test passes.

c) → Refactor -

clean code if needed.

→ All tests still pass.