

Practical 1:Write down the problem statement for a E-Library Management System.

In the current scenario, managing a library involves manual labor which is time-consuming and error-prone. The tasks include tracking books issued, returned, and overdue, maintaining records of members, cataloging books, and managing fine collection.

This manual system often leads to mismanagement, loss of records, and difficulty in searching for books.

The aim of this project, Library Management System, is to develop a fully automated system that will handle all the aforementioned tasks digitally and efficiently. The system will provide an easy-to-use interface for librarians to manage books and members, and for members to search for books and view their borrowing history. The system will also generate reports for better decision-making.

The scope of the project is bounded to the following functionalities:

1. Book Management: Add, update, delete, and search for books.
2. Member Management: Register, update, and delete member information.
3. Borrowing and Returning Books: Issue and return books, and calculate overdue fines if any.
4. Report Generation: Generate reports on book inventory, popular books, active members, etc.

The system will not include advanced features like e-book management, online reservations, or integration with external databases.

The primary users of the system will be librarians and library members. The system will be designed to be user-friendly, reliable, and efficient, thereby improving the overall management of the library.

This project will contribute to the digital transformation of library management, making it more efficient and user-friendly.

The success of the project will be measured by the ease of use, reduction in manual errors, and improved library operations.

This online library management system was developed to solve the major drawbacks that faced the people responsible for the general managing of any small or large libraries due to using the existing systems.

We mainly focus on solution to the following limitations brought by the existing system.

The problem occurred before having computerized system includes:

File lose

When computerized system is not implemented file is always lost because of human environment. Sometimes due to some human error there may be a loss of records.

File damaged.

Online library will save us a great deal since physical interference is greatly reduced due to the nature of the digital world and thus files and general important information of the library is safer than ever before.

Difficult to search record

When there is no computerized system there is always a difficulty in searching of records if the records are large in number.

Space consuming

After the number of records become large the space for physical storage of file and records also increases if no computerized system is implemented.

Cost consuming

As there is no computerized system the to add each record paper will be needed which will increase the cost for the management of the library.

The project aim and objectives that will be achieved after completion of this project are discussed in this subchapter. The aims and objectives are as follows:

Online book issue

- A separate column for digital library
- Library user login page where student can find books issued by him/her and date of return.
- A search column to search availability of books
- Admin login page where admin can monitor the system.
- Advanced features to enhance general library functioning.

Library Management System refers to an application that is specifically designed to manage generally small and medium size libraries and to help mainly those in charge of the library at that particular time.

Library Management System is an application mainly used by librarian to manage the library using a computerized system where he/she can record various transactions like issue of books, return of books, addition of new books, addition of new students etc.

Books and student maintenance modules are also included in this system which would keep track of the students using the library and also a detailed description about the books a library contains. With this computerized system there will be no loss of book record or member record which generally happens when a non computerized system is used.

In addition, report module is also included in the Library Management System. If the user's position is admin, the user is able to generate different kinds of reports like lists of students registered, list of books, issue and return reports.

All these modules are able to help librarian to manage the library with more convenience and in a more efficient way as compared to library systems which are not computerized

Library Management System is a computerized system which helps user(librarian) to manage the library daily activity in electronic format. It reduces the risk of paper work such as file lost, file damaged and time consuming. It can help user to manage the transaction or record more effectively and time-Saving.

The project aim and objectives that will be achieved after completion of this project are discussed in this subchapter. The aims and objectives are as follows:

- Online book issue
- A separate column for digital library
- Library user login page where students can find books issued by him/her and date of return.
- A search column to search availability of books
- Admin login page where admin can monitor the system.
- Advanced features to enhance general library functioning.

Improvement in control and performance

The system is developed to cope up with the current issues and problems of library .The system can add user, validate user and is also bug free.

Save cost

After computerized system is implemented less human force will be required to maintain the library thus reducing the overall cost.

Save time

Librarian is able to search record by using few clicks of mouse and few search keywords thus saving his valuable time.

Option of online Notice board

Librarian will be able to provide a detailed description of workshops going in the college as well as in nearby colleges

Lecture Notes

Teachers have a facility to upload lectures notes in a pdf file having size not more than 10mb.

When a library management system will be implemented librarian and user will easily access library as searching and book transaction will be very faster .

The system should accurately performs member registration ,member validation , report generation, book transaction and search.

The system is designed for a user friendly environment so that student and staff of library can perform the various tasks easily and in an effective way.

- Organizational Requirement
- Implementation Requirements'

In implementing whole system it uses html in front end with php as server side scripting language which will be used for database connectivity and the backend part is developed using mysql.

PRACTICAL 2: Perform requirement analysis and develop Software Requirement Specification Sheet (SRS) E-Library Management System

Software Requirements Specification (SRS) for E-Library Management System

1. Introduction

1.1 Purpose The purpose of this document is to define the Software Requirements Specification (SRS) for the Library Management System (LMS). The system aims to digitize and streamline the library operations, minimizing manual labor and reducing errors.

1.2 Document Conventions This document follows IEEE standards for SRS documentation.

1.3 Intended Audience and Reading Suggestions The intended audience includes:

- Library administrators
- Software developers
- System testers
- End-users (students and faculty members)
- Management team

1.4 Product Scope The Library Management System will facilitate book cataloging, member management, book borrowing and returning, and report generation. The system will be user-friendly, efficient, and capable of handling small to medium-sized libraries.

1.5 References

- IEEE Std 830-1998: IEEE Recommended Practice for Software Requirements Specifications
 - MySQL documentation
 - PHP and HTML user guides
-

2. Overall Description

2.1 Product Perspective The LMS is a standalone web-based system that will be accessed by authorized users. It does not integrate with external databases but operates within the library's internal network.

2.2 Product Functions The LMS will provide functionalities such as:

1. **Book Management** – Add, update, delete, and search books.
2. **Member Management** – Register, update, and delete members.
3. **Borrowing and Returning** – Issue and return books with automated fine calculation.

4. **Report Generation** – Generate reports on book inventory, borrowing history, and active users.
5. **User Authentication** – Provide login functionalities for librarians and members.

2.3 User Characteristics

- **Librarians:** Manage books, members, and transactions.
- **Students/Faculty:** Search and borrow books, view their transaction history.
- **Administrators:** Monitor system usage and generate reports.

2.4 Constraints

- The system will be developed using HTML, PHP, and MySQL.
- It will not include e-book management or online reservations.
- Maximum file upload size for lecture notes: 10MB.

2.5 Assumptions and Dependencies

- Users must have a stable internet connection to access the system.
 - Library staff must be trained to use the system.
-

3. Specific Requirements

3.1 Functional Requirements

- The system shall allow librarians to add, update, and delete books.
- The system shall allow students to search for books using keywords.
- The system shall record and track book borrowing and returning.
- The system shall calculate overdue fines automatically.
- The system shall provide login pages for both administrators and members.
- The system shall generate reports on library statistics.
- The system shall include an online notice board for events and announcements.
- The system shall allow faculty to upload lecture notes in PDF format.

Expanding Scope to Digital Library

- Include **e-book management** (upload, download, and view online).
- Support **multi-format digital content** (PDFs, audiobooks, research papers).
- Implement **access control** for premium or restricted content.
- Add **online reservation** for physical and digital resources.

New Functional Requirements

- Users should be able to **read e-books online** with annotation features.
- A **recommendation system** to suggest books based on user activity.
- Integration with **external academic databases** for research material.
- Implement a **citation generator** for research papers.
- **Updated Non-Functional Requirements**

4.Nonfunctional Requirements

4.1 Performance Requirements

The system shall accommodate high number of books and users without any fault. Responses to view information shall take no longer than 5 seconds to appear on the screen.

4.2 Safety Requirements

System use shall not cause any harm to human users.

4.3 Security Requirements

System will use secured database

Normal users can just read information but they cannot edit or modify anything except their personal and some other information.

System will have different types of users and every user has access constraints.

4.4 Error Handling

LMS product shall handle expected and non-expected errors in ways that prevent loss in information and long downtime period

5. Appendices

- Detailed database schema
- User interface design sketches
- System architecture diagram
- Use case diagrams
- Entity-relationship diagrams (ERD)
- Sequence diagrams

This SRS document serves as the foundation for the development and implementation of the Library Management System.

Use This link to see SRS with Black BOOK

<https://www.studocu.com/in/document/indian-institute-of-technology-kanpur/product-management/e-book-management-project-pdf-final/95199803>

Practical 3: Draw the function oriented diagram: Data Flow Diagram (DFD) and Structured chart.





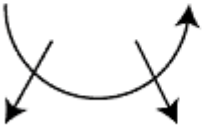
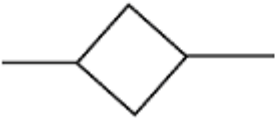
Structured Charts

It partitions a system into block boxes. A Black box system that functionality is known to the user without the knowledge of internal design.

Structured Chart is a graphical representation which shows:

- System partitions into modules
- Hierarchy of component modules
- The relation between processing modules
- Interaction between modules
- Information passed between modules

The following notations are used in structured chart:

| SYMBOL | DESCRIPTION |
|---|--------------|
|  | Module |
|  | Arrow |
|  | Data couple |
|  | Control Flag |
|  | Loop |
|  | Decision |

3. The structure charts is based on a library system.

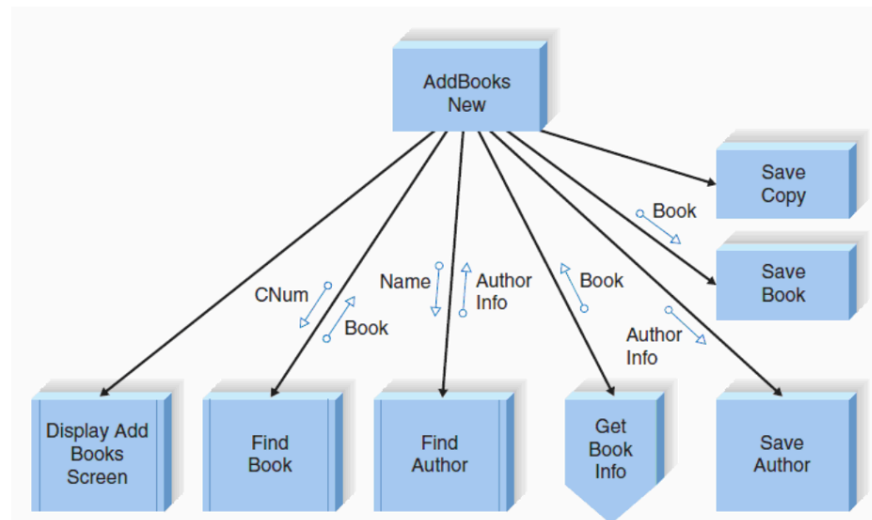
(1) (4 points) write down the name of each control module.

(2) (4 points) write down the name of each library module.





(3) (4 points) write down the name of each subordinate module.

(4) (4 points) write down the name of each module used as an off-page connector.

(5) (4 points) identify whether the structure chart is a transaction or transform structure chart.



A Data Flow Diagram (DFD) Notations:

| Symbol | Name | Meaning |
|---|-------------------|--|
|  | Rounded Rectangle | It represents functions which transforms input to output. The transformation name indicates its function. |
|  | Rectangle | It represents data stores. Again, they should give a descriptive name. |
|  | Circle | It represents user interactions with the system that provides input or receives output. |
|  | Arrows | It shows the direction of data flow. Their name describes the data flowing along the path. |
| "and" and "or" | Keywords | The keywords "and" and "or". These have their usual meanings in boolean expressions. They are used to link data flows when more than one data flow may be input or output from a transformation. |

A **Data Flow Diagram (DFD)** represents the movement of data within a system through inputs, processing, and outputs.

Inputs:

- **Book Request:** Student requests a book.
- **Library Card:** Student provides identity proof.

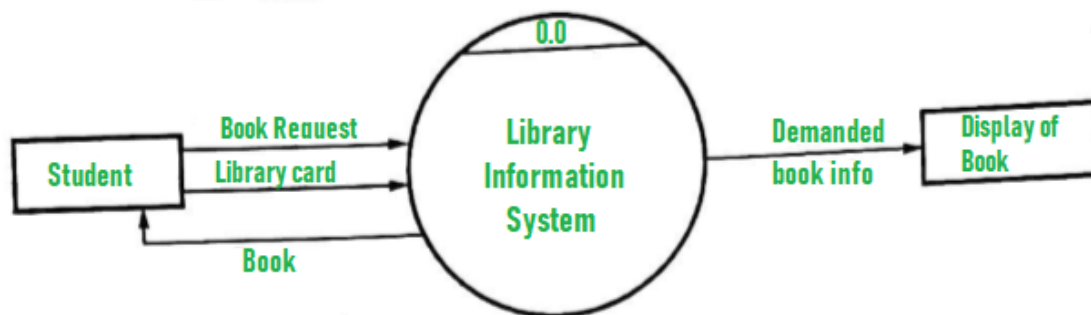
Processing:

- The system verifies the student's credentials.
- It checks book availability and reservation status.

Outputs:

- **Issued Book:** If available, the book is issued.
- **Book Information:** The system displays details to help the student select a book.

Level 0 DFD –



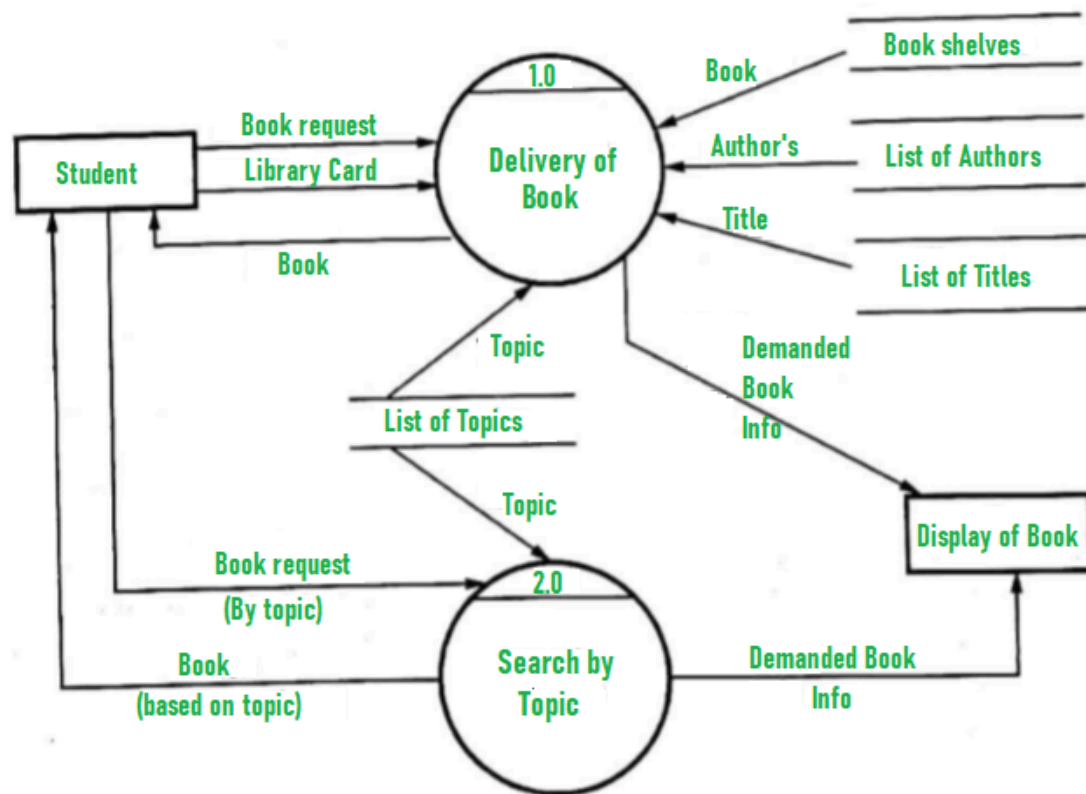
Level 1 DFD – E-Library Management System

Processes:

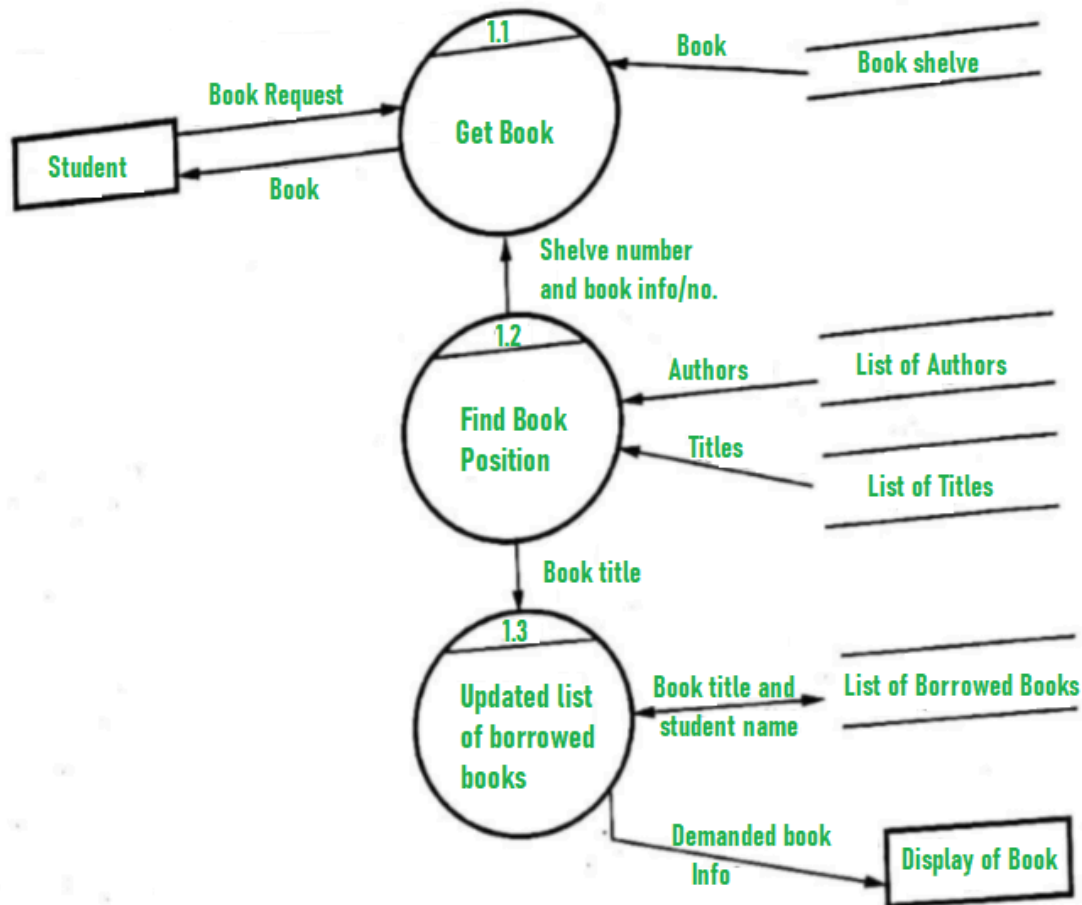
1. **Book Delivery** – Manages book issuing and returning.
2. **Search by Topic** – Allows users to find books based on topics.
3. **List Generation** – Displays **authors, titles, topics, and bookshelves** for book selection.

Data Stores:

- **Book Database** – Stores book details, availability, and locations.
- **User Database** – Maintains student and librarian records.
- **Transaction Records** – Logs book borrowing, returns, and overdue fines.



Level 1 DFD



Level 2 DFD

Out of scope:

Other activities like purchasing of new books, uploading notes pdf by teachers ,replacement of old books or charging a fine are not considered in the above system.

Note:

You could create DFD by Using these key points:

DFD Level 0 (Context Diagram)

This is the high-level overview of the system.

Entities & Data Flow:

- **User** → Requests books, borrows/returns, views records
- **Admin** → Manages users, books, and transactions
- **Library Database** → Stores book details, user data, transactions
-

DFD Level 1 (Detailed Process Breakdown)

This level expands the **E-Library Management System** into functional modules.

Processes & Data Flow:

① User Management

- User → (Login & Authentication) → System
- System → (Verify Credentials) → Library Database
- Library Database → (Send Response) → System

② Book Management

- Admin → (Add/Update/Delete Book) → Library Database
- User → (Search Book) → Library Database

③ Borrow & Return Process

- User → (Request Borrow) → System
- System → (Check Availability) → Library Database
- If Available → (Approve Borrow) → User & Update Database
- User → (Return Book) → System
- System → (Update Return Status) → Library Database

④ E-Resource Management

- Admin → (Upload Digital Books) → Library Database
- User → (Download E-Books) → Library Database

⑤ Reservation System

- User → (Request Reservation) → System
- System → (Check Book Availability) → Library Database
- If Not Available → (Add to Waitlist) → User

6 Reporting & Analytics

- Admin → (Generate Reports) → Library Database
- System → (Send Reports) → Admin

DFD Level 2 (Detailed Data Flow for Borrowing Process)

- **User requests to borrow a book**
- **System checks availability in Library Database**
- **If available:** Issue the book and update the database
- **If not available:** Notify the user and allow reservation

PRACTICAL 4. Draw the user's view analysis for the suggested system: Use case diagram.

What is a Use case diagram?

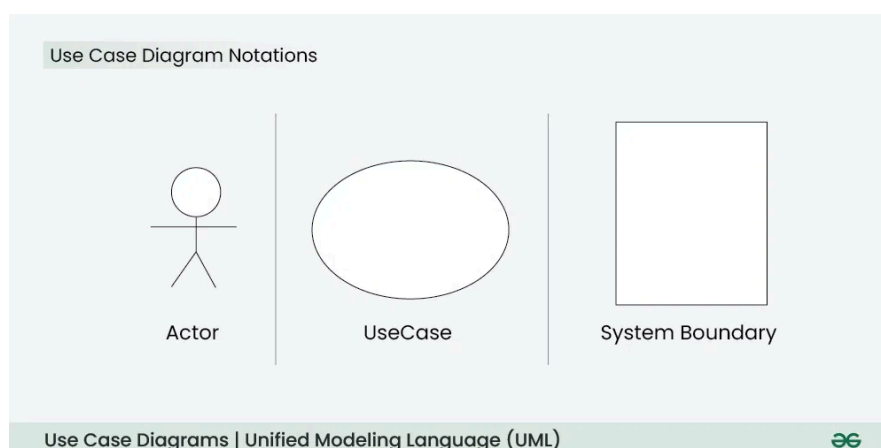
Use Case Diagram, referred to as a Behavior model or diagram. It simply describes and displays the relationship or interaction between the users or customers and providers of the application service or the system. It describes different actions that a system performs in collaboration to achieve something with one or more users of the system. A use-case diagram is used a lot nowadays to manage the system.

Use Case Diagram Notations

Use Case Diagram consists of the following components:

1. **Actor:** Actors are external entities that interact with the system. These can include users, other systems, or hardware devices. In the context of a Use Case Diagram, actors initiate use cases and receive the outcomes.
2. **Use Case:** Use cases are like scenes in the play. They represent specific things your system can do.
3. **System Boundary:** The system boundary is a visual representation of the scope or limits of the system you are modeling. It defines what is inside the system and what is outside.

Use Case Diagram for Library Management System



1. Actors:

- User (Staff or Student)
- Librarian

2. Use Cases:

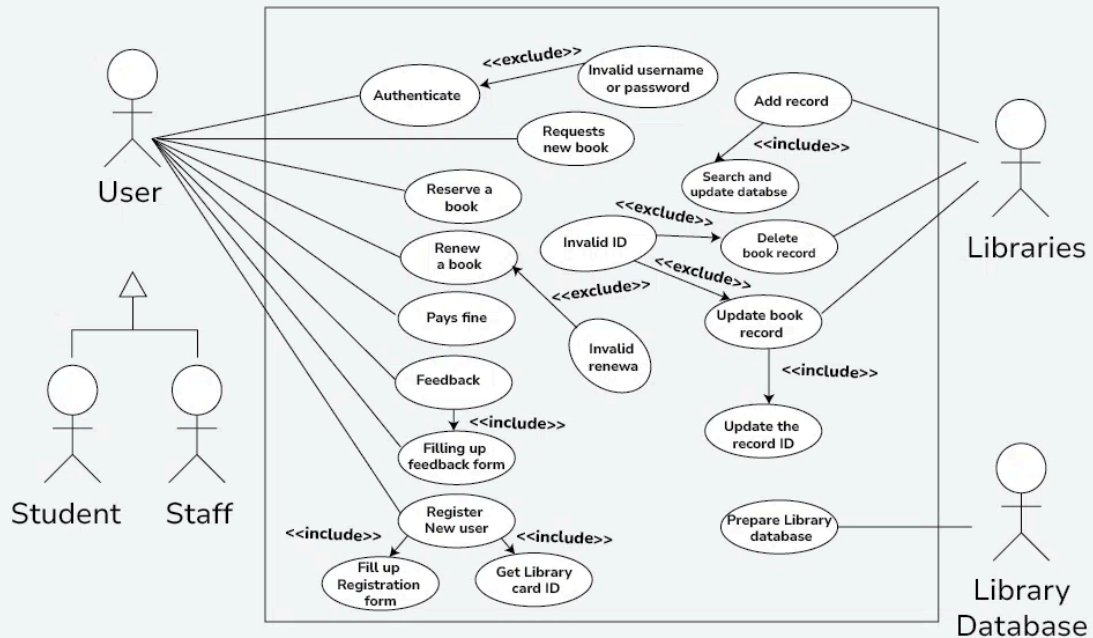
- Register New User
- Issue Library Card
- Request New Book
- Reserve Book
- Renew Book
- Pay Fine
- Fill Feedback Form
- Manage Records
- Delete Records
- Update Database

3. System Boundary:

- The system boundary will encompass all the use cases mentioned above.

Below is the use case diagram of a Library Management System

Use Case Diagram of Library Management System



Explanation of Use Case Diagram of a Library Management System

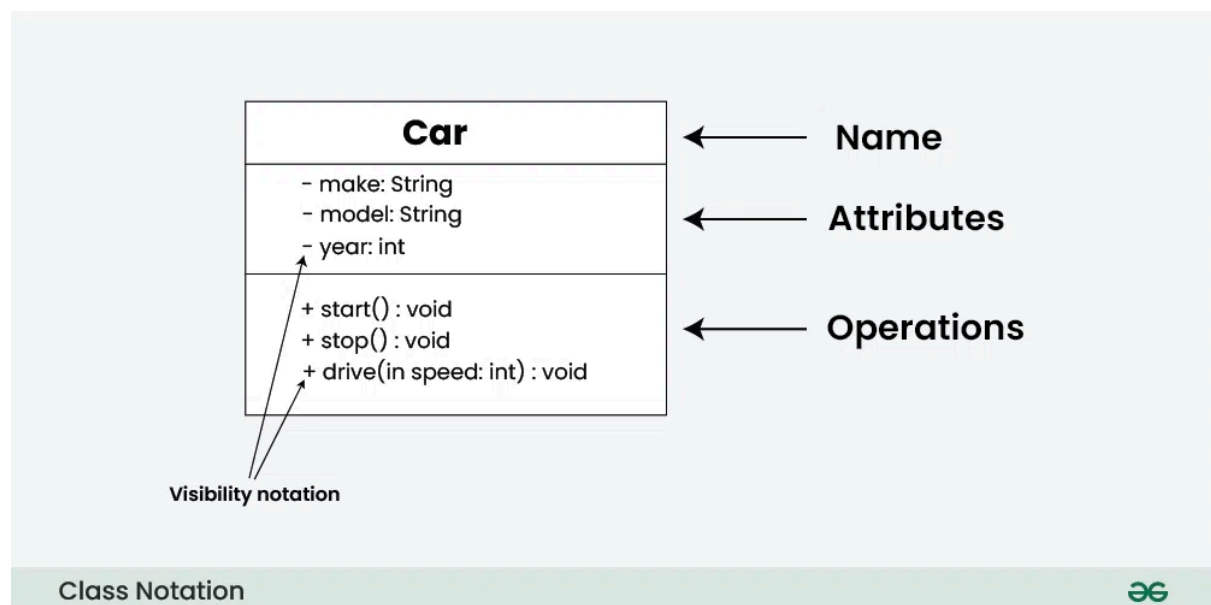
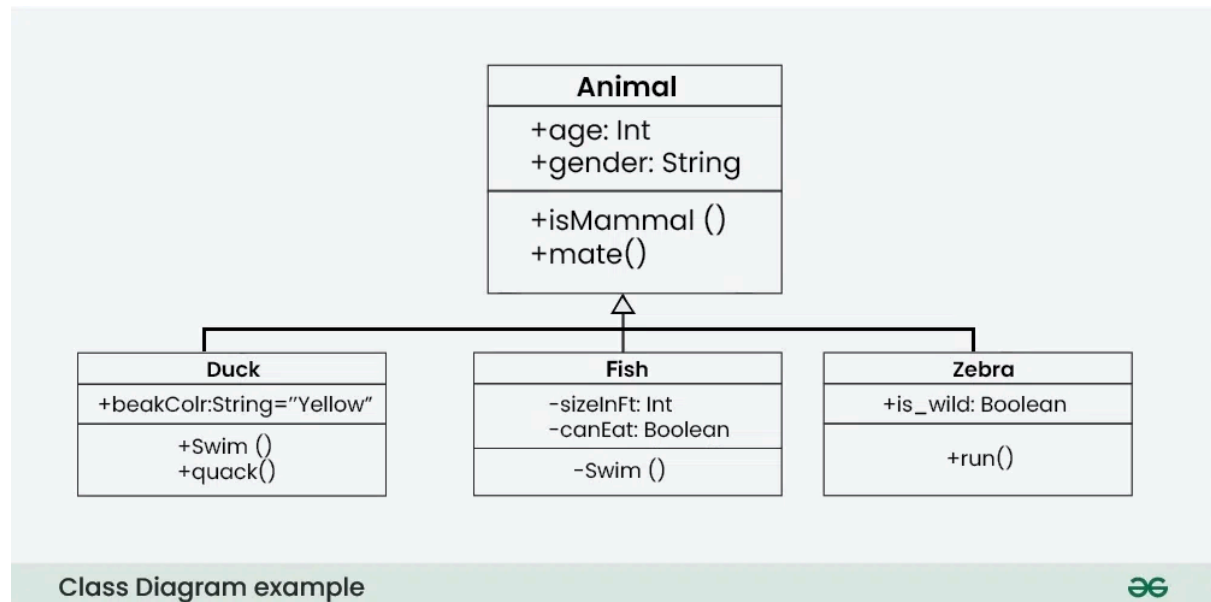
Here, we will understand the designing use case diagram for the library management system. Some scenarios of the system are as follows :

1. User who registers himself as a new user initially is regarded as staff or student for the library system.
 - For the user to get registered as a new user, registration forms are available that is needed to be fulfilled by the user.
 - After registration, a library card is issued to the user by the librarian. On the library card, an ID is assigned to cardholder or user.
2. After getting the library card, a new book is requested by the user as per there requirement.

3. After, requesting, the desired book or the requested book is reserved by the user that means no other user can request for that book.
4. Now, the user can renew a book that means the user can get a new due date for the desired book if the user has renewed them.
5. If the user somehow forgets to return the book before the due date, then the user pays fine. Or if the user forgets to renew the book till the due date, then the book will be overdue and the user pays fine.
6. User can fill the feedback form available if they want to.
7. Librarian has a key role in this system. Librarian adds the records in the library database about each student or user every time issuing the book or returning the book, or paying fine.
8. Librarian also deletes the record of a particular student if the student leaves the college or passed out from the college. If the book no longer exists in the library, then the record of the particular book is also deleted.
9. Updating database is the important role of Librarian.

PRACTICAL 5. Draw the structural view diagram for the system: Class diagram.

Class Diagram Notation

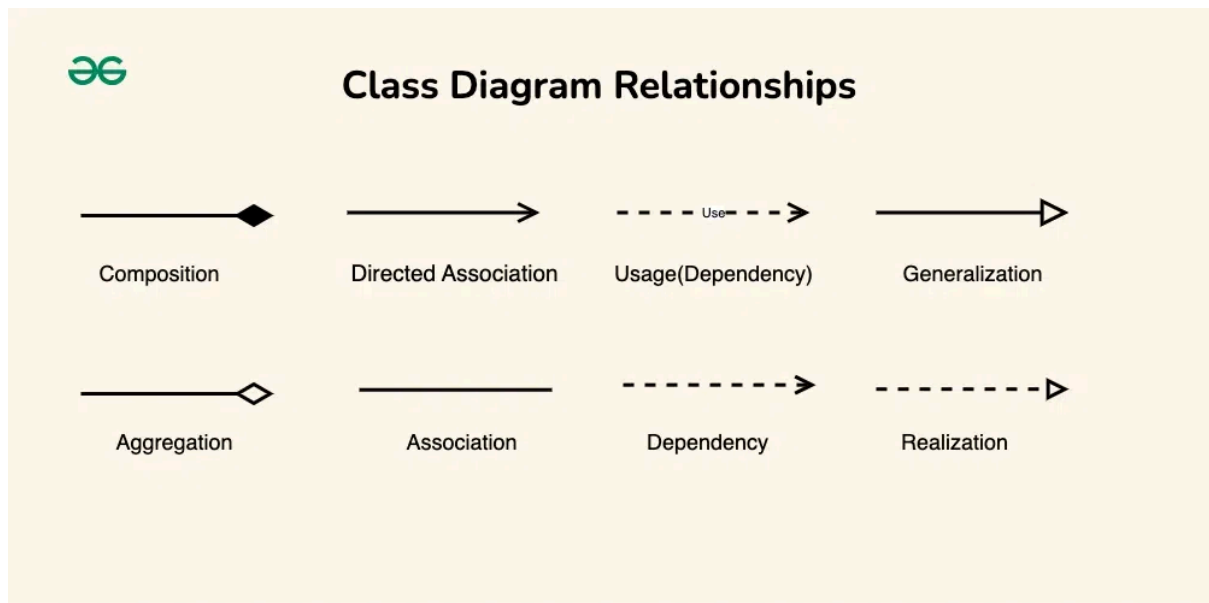


- **Class Name:** Written in the top compartment, centered, and bold.
- **Attributes:** Listed in the second compartment with visibility and data type.
- **Methods:** Listed in the third compartment with visibility, return type, and parameters.
- **Visibility Notation:**

- **+ Public** (visible to all)
- **- Private** (visible only within the class)
- **# Protected** (visible to subclasses)
- **~ Package/default** (visible to classes in the same package)

Parameter Directionality

- **In (Input):** Arrow points towards the receiving class.
 - **Out (Output):** Arrow points away from the receiving class.
 - **InOut (Input and Output):** Arrow points both towards and away from the receiving class.
-



Relationships between Classes

- 1. Association:**
 - **Bi-directional relationship.**
 - **Solid line between classes.**
 - **Example:** Library ↔ Book (Multiple books belong to one library).
- 2. Directed Association:**
 - **One-way relationship.**
 - **Solid line with an arrow indicating direction.**
 - **Example:** Teacher → Course (Teacher teaches the course).
- 3. Aggregation:**
 - **Whole-part relationship (weaker form).**

- Diamond on the side of the whole class.
 - Example: Company ◇ Employee (Employees exist independently of the company).
4. **Composition:**
- Strong ownership relationship.
 - Filled diamond on the side of the whole class.
 - Example: ContactBook ◆ Contact (Contacts don't exist without the contact book).
5. **Generalization (Inheritance):**
- “Is-a” relationship.
 - Solid line with a hollow arrow pointing from child to parent class.
 - Example: BankAccount → (SavingsAccount, CurrentAccount).
6. **Realization (Interface Implementation):**
- Class implements an interface.
 - Dashed line with an open arrowhead pointing to the interface.
 - Example: Person and Corporation realizing Owner interface.
7. **Dependency Relationship:**
- Loosely coupled connection.
 - Dashed arrow indicating a class depends on another.
 - Example: Person ---► Book (Person reads the Book).
8. **Usage (Dependency) Relationship:**
- Client class uses the supplier class's functionality.
 - Dashed arrow from client to supplier.
 - Example: Car ---► FuelTank (Car uses FuelTank services).

PRACTICAL 6: Draw the behavioral view diagram for Library Management System :Activity diagram

Sequence Diagrams – Unified Modeling Language (UML)

1. What are Sequence Diagrams?

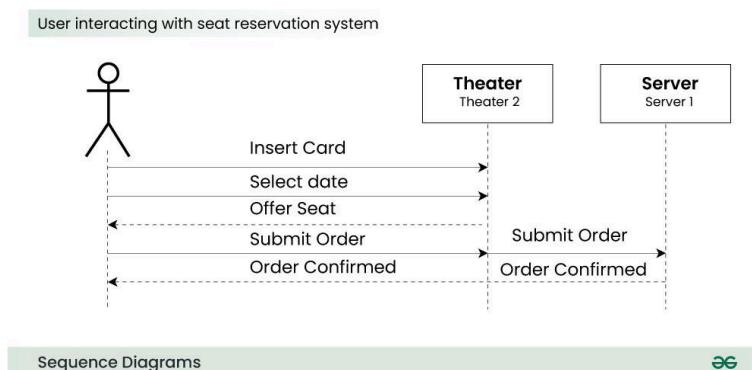
- Sequence diagrams are UML diagrams that represent the interactions between objects or components in a system over time.
- They focus on the order and timing of messages exchanged, illustrating the sequence of operations.

2. Why use Sequence Diagrams?

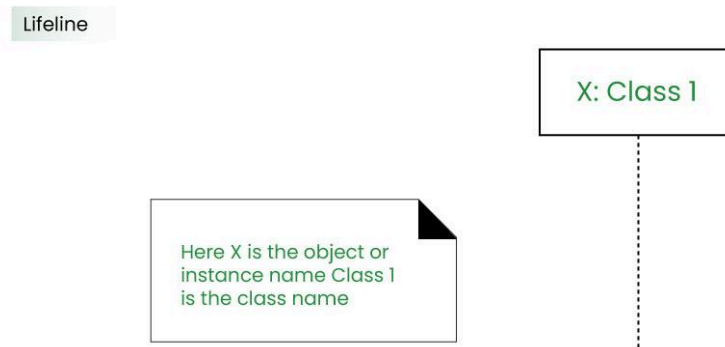
- **Visualizing Dynamic Behavior:** Depicts sequential interactions, making dynamic processes easy to understand.
- **Clear Communication:** Simplifies complex interactions for better team communication.
- **Use Case Analysis:** Illustrates how specific processes are executed within a system.
- **System Architecture Design:** Helps in defining communication among system components.
- **Documenting System Behavior:** Serves as a reference for developers and maintenance teams.
- **Debugging and Troubleshooting:** Identifies bottlenecks or errors in system processes.

3. Sequence Diagram Notations

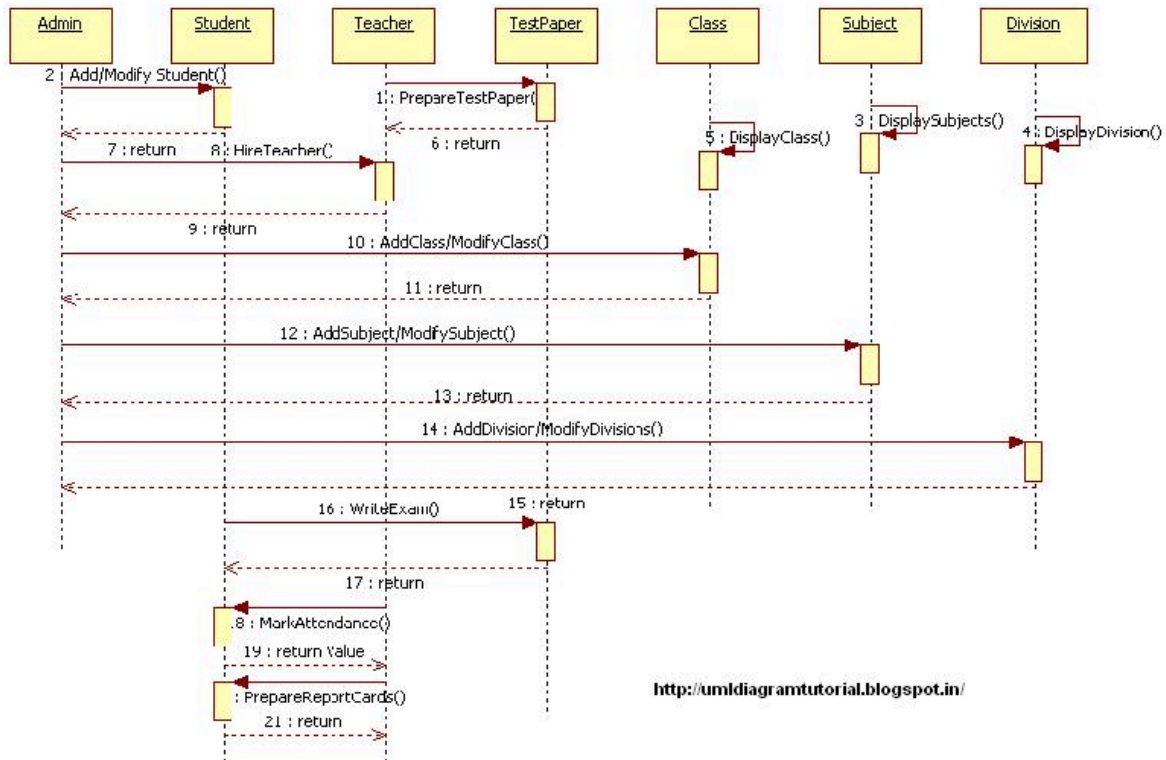
- **Actors:** External entities interacting with the system (e.g., users). Represented by a stick figure.



- **Lifelines:** Depicts an object's existence over time. Shown as a rectangle (head) atop a vertical dashed line (stem).



- **Messages:** Show communication between objects, including:
 - **Synchronous Message:** Waits for a reply (solid arrow).
 - **Asynchronous Message:** Does not wait for a reply (lined arrow).
 - **Create Message:** Instantiates a new object (dotted arrow labeled "create").
 - **Delete Message:** Destroys an object (arrow terminating with an "X").
 - **Self Message:** Object sends a message to itself (U-shaped arrow).
 - **Reply Message:** Return message from receiver to sender (dotted line with open arrow).
 - **Found Message:** Unknown source sends a message (arrow from endpoint to lifeline).
 - **Lost Message:** Recipient is unknown (arrow from lifeline to endpoint).
- **Guards:** Conditional logic that controls the flow of messages.



4. How to Create Sequence Diagrams?

1. **Identify the Scenario:** Define the specific interaction or use case.
2. **List Participants:** Determine all involved objects or actors.
3. **Define Lifelines:** Draw dashed vertical lines for each participant.
4. **Arrange Lifelines:** Organize lifelines horizontally in the sequence of interactions.
5. **Add Activation Bars:** Show the time when an object is active.
6. **Draw Messages:** Represent communication between participants using arrows.
7. **Include Return Messages:** Show replies with dashed arrows.
8. **Indicate Timing and Order:** Number the messages to show sequence.
9. **Include Conditions and Loops:** Use combined fragments for conditions and loops.
10. **Parallel Execution:** Represent concurrent activities with parallel dashed lines.
11. **Review and Refine:** Ensure clarity and correctness.
12. **Add Annotations:** Provide context or comments as needed.
13. **Document Assumptions:** Note any relevant constraints or assumptions.
14. **Use Tools:** Utilize UML modeling tools for professional diagrams.

5. Use Cases of Sequence Diagrams

- **System Behavior Visualization:** Shows dynamic behavior and message flow.
- **Software Design and Architecture:** Assists in designing interactions between components.
- **Communication and Collaboration:** Facilitates understanding among stakeholders.

- **Requirements Clarification:** Clarifies interactions for better requirement analysis.
- **Debugging and Troubleshooting:** Visualizes control flow to identify issues.

6. Challenges of Using Sequence Diagrams

- **Complexity and Size:** Large systems lead to complex diagrams that are hard to read.
- **Abstraction Level:** Balancing detail without overwhelming the reader is challenging.
- **Dynamic Nature:** Frequent updates are required as systems evolve.
- **Ambiguity in Messages:** Messages might be unclear, leading to misunderstandings.
- **Concurrency and Parallelism:** Representing simultaneous interactions can be complicated.
- **Real-Time Constraints:** Capturing precise timing requirements is difficult.

PRACTICAL 7: Prepare time line chart/Gantt Chart/PERT Chart

Project Evaluation and Review Technique (PERT)

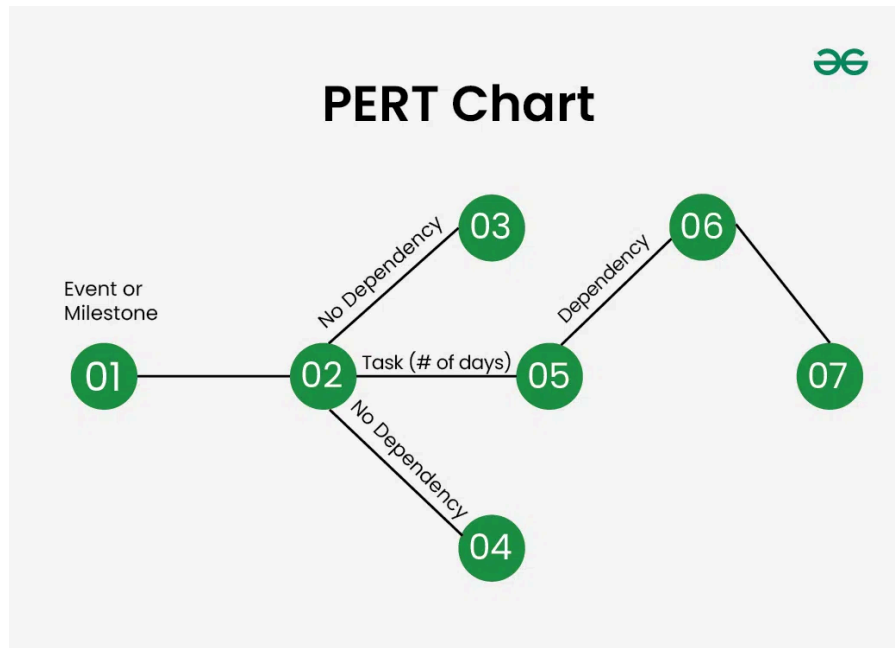
PERT is a project management technique used to plan, schedule, and coordinate tasks within a project. It helps visualize the sequence and timing of activities, ensuring that all tasks are integrated and managed efficiently. PERT is especially useful for complex and research-based projects with uncertain activity durations.

PERT Chart

A PERT chart is a visual representation of a project's tasks in a flowchart format. It shows the sequence of tasks, their dependencies, and the estimated duration for each activity. The main purpose of a PERT chart is to identify the **critical path**, which determines the minimum time required to complete the project.

Components of a PERT Chart

1. **Nodes:** Represent tasks or milestones, often labeled with the task name and duration.
2. **Arrows:** Indicate the sequence and dependencies between tasks.
3. **Time Estimation:** Three types are used:
 - **Optimistic Time (O)** – The shortest time to complete a task.
 - **Pessimistic Time (P)** – The longest time to complete a task.
 - **Most Likely Time (M)** – The most probable time required.
4. **Critical Path:** The longest sequence of dependent tasks that determines the project's shortest possible duration. Delays in these tasks directly impact the project's completion date.
5. **Milestones:** Significant events or deadlines that indicate major progress points in the project.



How to Create a PERT Chart

1. **Identify Project Tasks** – List all tasks required for project completion.
2. **Define Task Dependencies** – Determine the order of tasks and their dependencies.
3. **Estimate Timeline** – Use optimistic, pessimistic, and most likely times for accurate duration estimates.
4. **Calculate Critical Path** – Identify the longest path of dependent tasks that determines the project's shortest completion time.
5. **Manage Task Progress** – Track progress, address delays, and adjust timelines as needed.

PERT Chart vs. Gantt Chart

| Feature | PERT Chart | Gantt Chart |
|------------------------|--|---------------------------------------|
| Purpose | Visualize task dependencies and sequence | Schedule tasks over time |
| Focus | Task relationships and workflow | Task duration and timeline |
| Representation | Network diagram with nodes and arrows | Horizontal bar chart |
| Time Estimation | Uses optimistic, pessimistic, and likely times | Shows start and end dates |
| Complexity | More complex, suitable for large projects | Simpler, easier to read |
| Flexibility | Adaptable to changing paths | Less flexible once established |
| Best Use Case | Research and development projects | Construction and production schedules |

A **Gantt Chart** is a project management tool that displays tasks or activities along a timeline. It visually represents the start and end dates of tasks, their duration, and the overall project schedule. It is particularly useful for tracking progress and ensuring that tasks are completed on time.

Components of Gantt Charts

- Timeline:**
 - Shows the project duration, divided into units like days, weeks, or months.
 - Helps visualize the time allocated for each task.
- Tasks/Activities:**
 - Lists all the tasks required to complete the project.
 - Usually arranged in sequential order from top to bottom.
- Bars:**
 - Horizontal bars represent the duration of each task.
 - The length of the bar corresponds to the time required to complete the task.
- Dependencies:**
 - Arrows connecting tasks show dependencies, indicating that one task must be completed before another can start.
- Milestones:**
 - Special markers that indicate significant events or deadlines.
 - Typically shown as diamonds on the timeline.
- Progress Indicators:**

- Shaded portions within the task bars indicate the percentage of work completed.
 - Useful for tracking the progress of each task.
7. **Task Owners/Resources:**
- Specifies who is responsible for each task.
 - Helps in resource allocation and accountability.
8. **Start and End Dates:**
- Displayed at the beginning and end of each task bar.
 - Clearly shows the timeline for task completion.

Advantages of Gantt Charts

- **Visual Clarity:** Provides an easy-to-understand overview of the project schedule.
- **Task Tracking:** Helps monitor the progress of tasks and identify delays.
- **Resource Allocation:** Clarifies who is responsible for each task.
- **Dependency Management:** Shows task dependencies to avoid scheduling conflicts.

Challenges of Gantt Charts

- **Complexity in Large Projects:** Can become difficult to manage for projects with many tasks.
- **Static Representation:** Requires frequent updates if project timelines change.
- **Time-Consuming Setup:** Initial creation and maintenance can be time-consuming.

Gantt Chart:

