SYSTEM DESIGN

Presented by

Dr. K. Kissi Mireku

CENTRAL UNIVERSITY

FAITH • INTEGRITY • EXCELLENCE

System Design

- *Logical design.*
  - Database design
  - Entity Relationship Diagram
  - Use Case Diagrams
  - Context diagram
  - Input and output designs.
- *Physical design*
- *Documentation Control*
  - Program documentation
  - System documentation
  - Operations documentation
  - User documentation

System Design

- **System design** is the phase that bridges the gap between problem domain and the existing system in a manageable way.
  - *This phase focuses on the solution domain, i.e. "how to implement?"*
- It is the phase where the SRS document is converted into a format that can be implemented and decides how the system will operate.
- In this phase, the complex activity of system development is divided into several smaller sub-activities, which coordinate with each other to achieve the main objective of system development.

System Design

Inputs to System Design

System design takes the following inputs −

- Statement of work
- Requirement determination plan
- Current situation analysis
- Proposed system requirements including a conceptual data model, modified DFDs, and Metadata (data about data).

Outputs for System Design

- System design gives the following outputs −
- Infrastructure and organizational changes for the proposed system.
- A data schema, often a relational schema.
- Metadata to define the tables/files and columns/data-items.
- A function hierarchy diagram or web page map that graphically describes the program structure.
- Actual or pseudocode for each module in the program.
- A prototype for the proposed system.

System Design Types

- Systems design encompasses two major aspects of the new system:
  - *Logical systems design states what the system will do, with abstract specifications.*
  - *Physical systems design states how the system will perform its functions, with actual physical specifications.*

We also have the following design

- Architectural Design
- It is also known as high level design that focuses on the design of system architecture. It describes the structure and behavior of the system. It defines the structure and relationship between various modules of system development process.
- Detailed Design
- It follows Architectural design and focuses on development of each module.

System Design Types

Logical systems design

•Logical design pertains to an abstract representation of the data flow, inputs, and outputs of the system.

- It describes the inputs (sources), outputs (destinations), databases (data stores), procedures (data flows) all in a format that meets the user requirements.

• While preparing the logical design of a system, the system analyst specifies the user needs at level of detail that virtually determines the information flow into and out of the system and the required data sources.

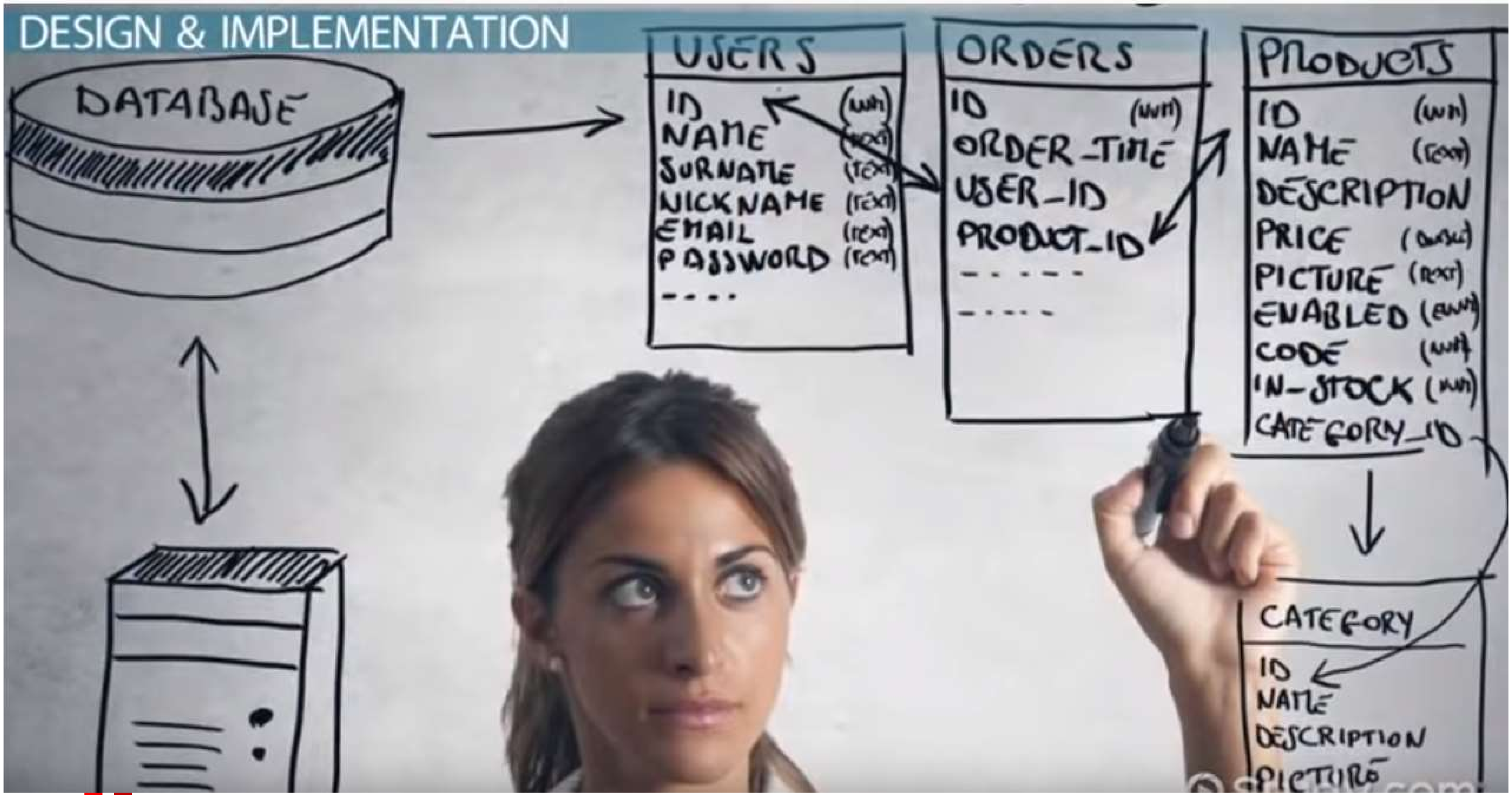
- Data flow diagram, E-R diagram modeling are used.

➢Logical design specifications include the design of

•*inputs*
•*processing*
•*outputs*
•*databases*
•*telecommunications*

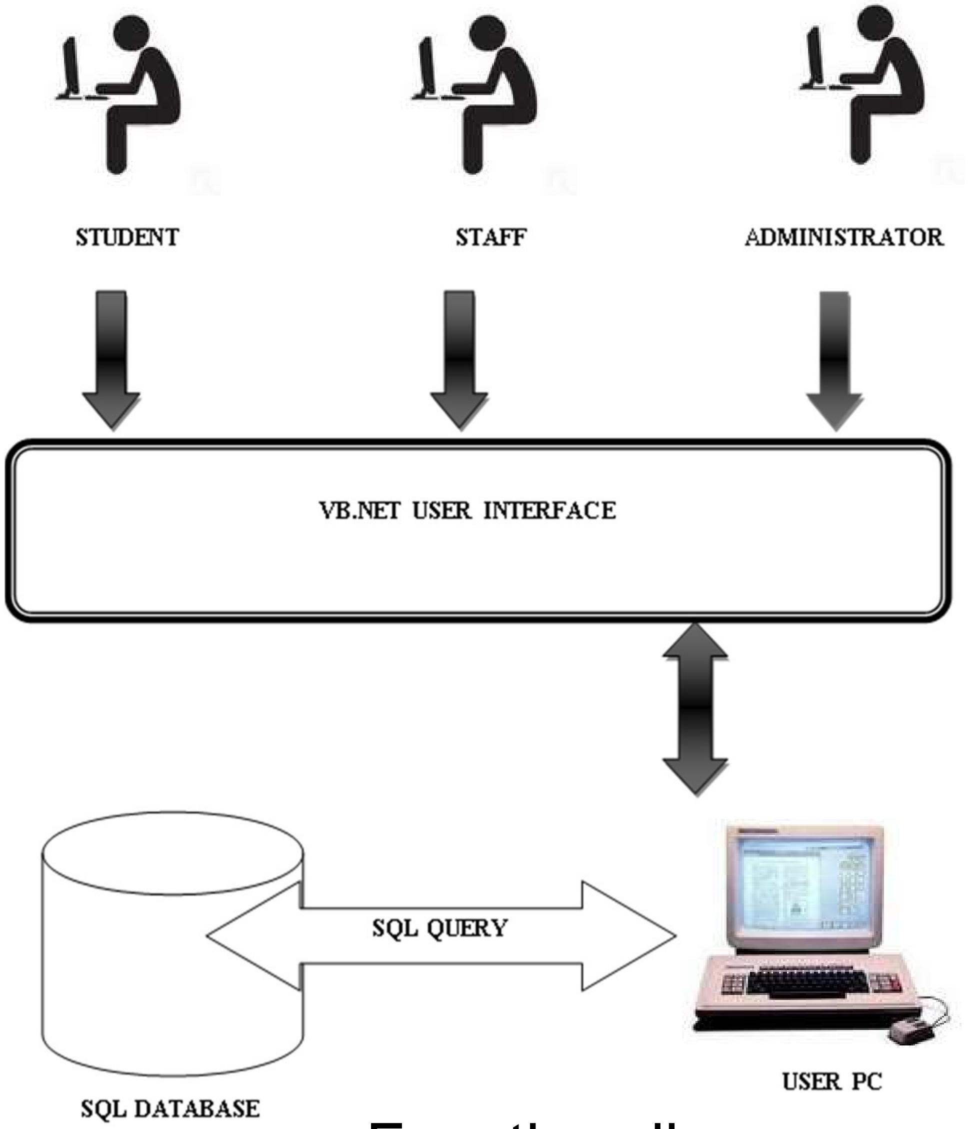
- telecommunications
- controls
- security
- and IS jobs.

System Design - Logical design

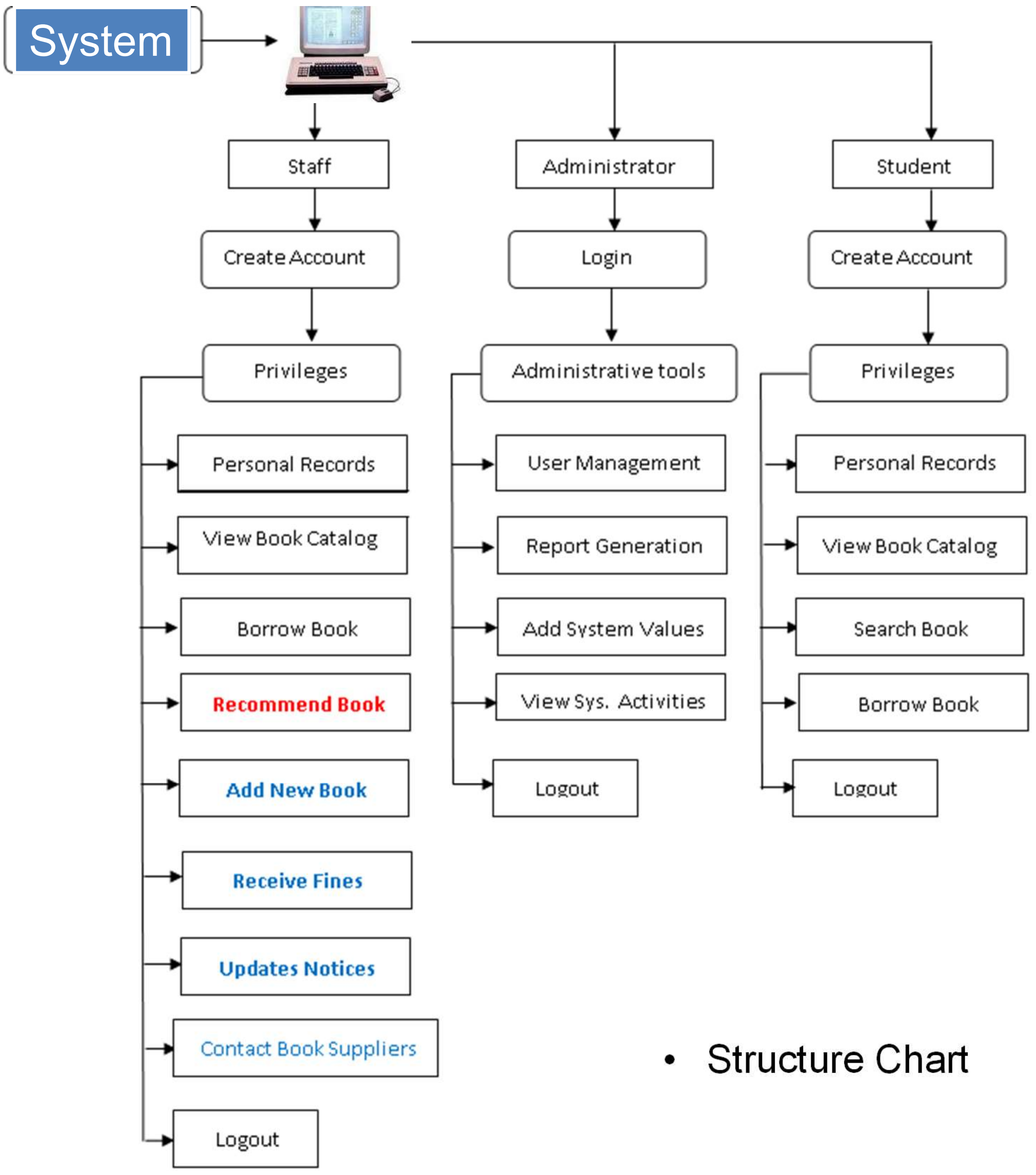


System Design Tool - Functional Req.

A functional requirement relates directly to a process the system has to perform as a part of supporting a user task and/or information it needs to provide as the user is performing a task



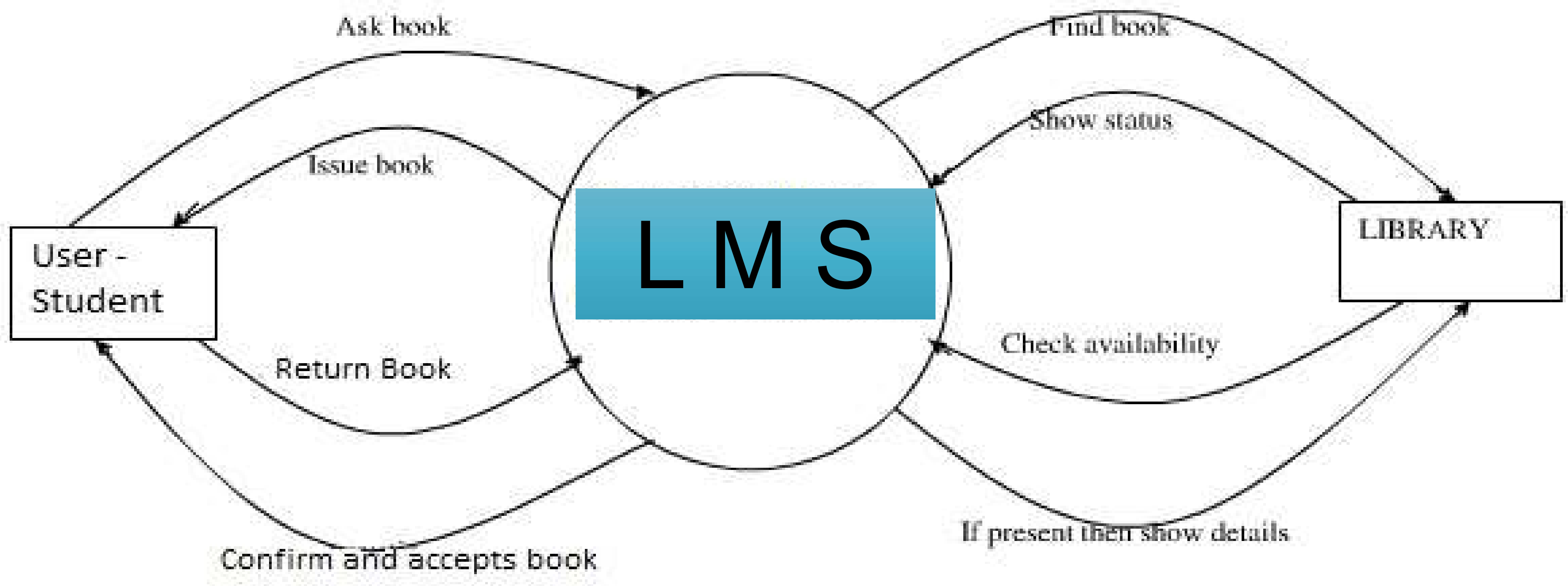
- Function diagram



- Structure Chart

System Design Tool - Context diagram

the context diagram shows the entire system in context with its environment.



System Design - Use Case Diagram

- What is a Use Case

  A formal way of representing how a business system interacts with its environment - The emphasis is on *what* a system does rather than *how.*

  Illustrates the activities that are performed by the users of the system

  A sequence of actions a system performs that yields a valuable result for a particular actor.

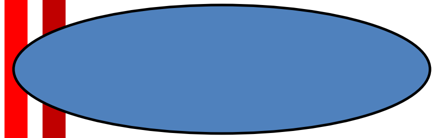
  A set of activities that produce some output.

- **Use case diagrams** describe what a system does from the standpoint of an external observer. Use case diagrams are closely connected to scenarios.

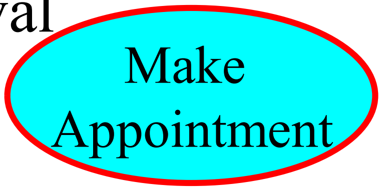
Use Case Diagrams

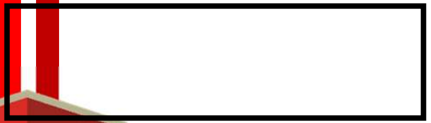
- The key elements of use case diagrams are:
  - **Use Cases** - modeling the functional goals of the system (describe scenarios that describe the interaction between users of the system (the actor) and the system itself).
  - **Actors** - classes of user (someone or something interacting with use case).
  - **Communication** - between actors and use cases, showing which use cases are instigated by which actors (often referred to as functional entitlement) –
  - **Dependencies** - between use cases that describe how the flow of one use case might rely on the flow of another

Use Case Diagram - Symbols

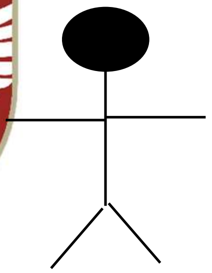


- Use Case - Represented by an oval

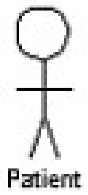


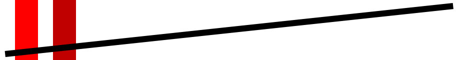


- Boundary - A boundary rectangle is placed around the perimeter of the system to show how the actors communicate with the system.

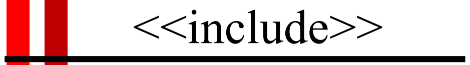


- Actor - Labelled using a descriptive noun or phras

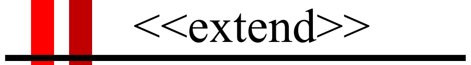




- Connection



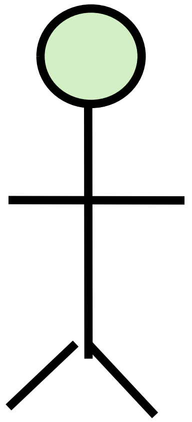
- Include relationship - Represents the inclusion of the functionality of one use case within another



- Extend relationship - Represents the extension of the use case to include optional functionality

Use Case Diagram - Actor

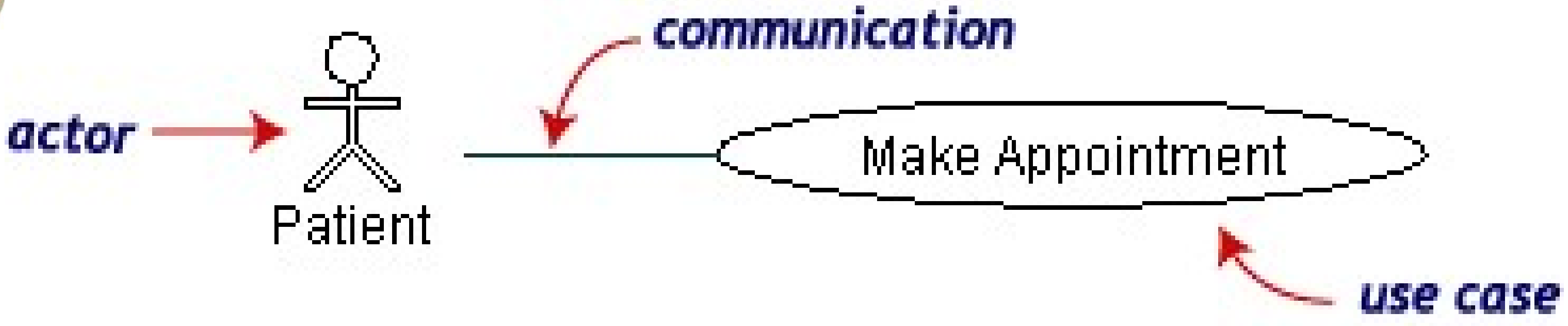
- A user or outside system that interacts with the system being designed in order to obtain some value from that interaction
- Actor *triggers* use case.
- Actor has responsibility towards the system (inputs), and Actor have expectations from the system (outputs).
- An Actor is outside or external the system.
- It can be a:
  - Human
  - Peripheral device (hardware)
  - External system or subsystem
  - Time or time-based event
- Represented by stick figure



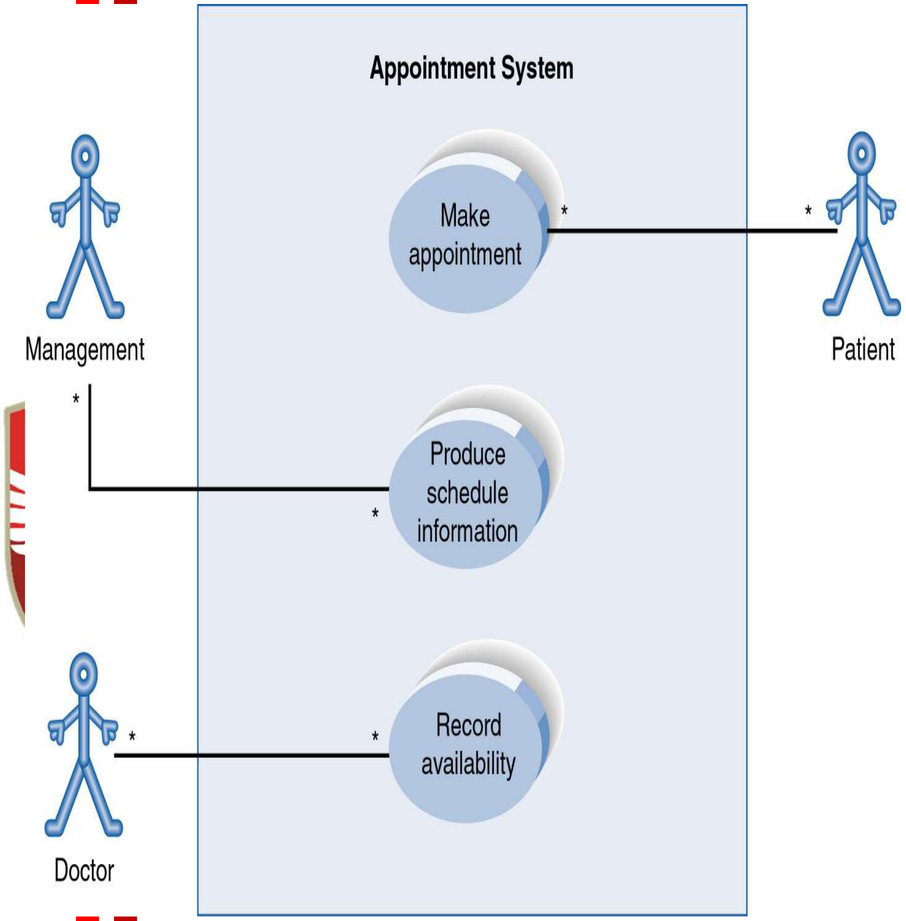
Use Case Diagram - Example

Actors are stick figures. Use cases are ovals. Communications are lines that link actors to use cases.

- The picture below is a **Make Appointment** use case for the medical clinic.
- The actor is a **Patient**. The connection between actor and use case is a **communication association** (or **communication** for short).



Use Case Diagram - Example



Use Case Name: Borrow Book

a. Description: The borrow book describes the scenario where the user after searching for a particular or view the book catalog, select a book to be lend from the library for sometimes.

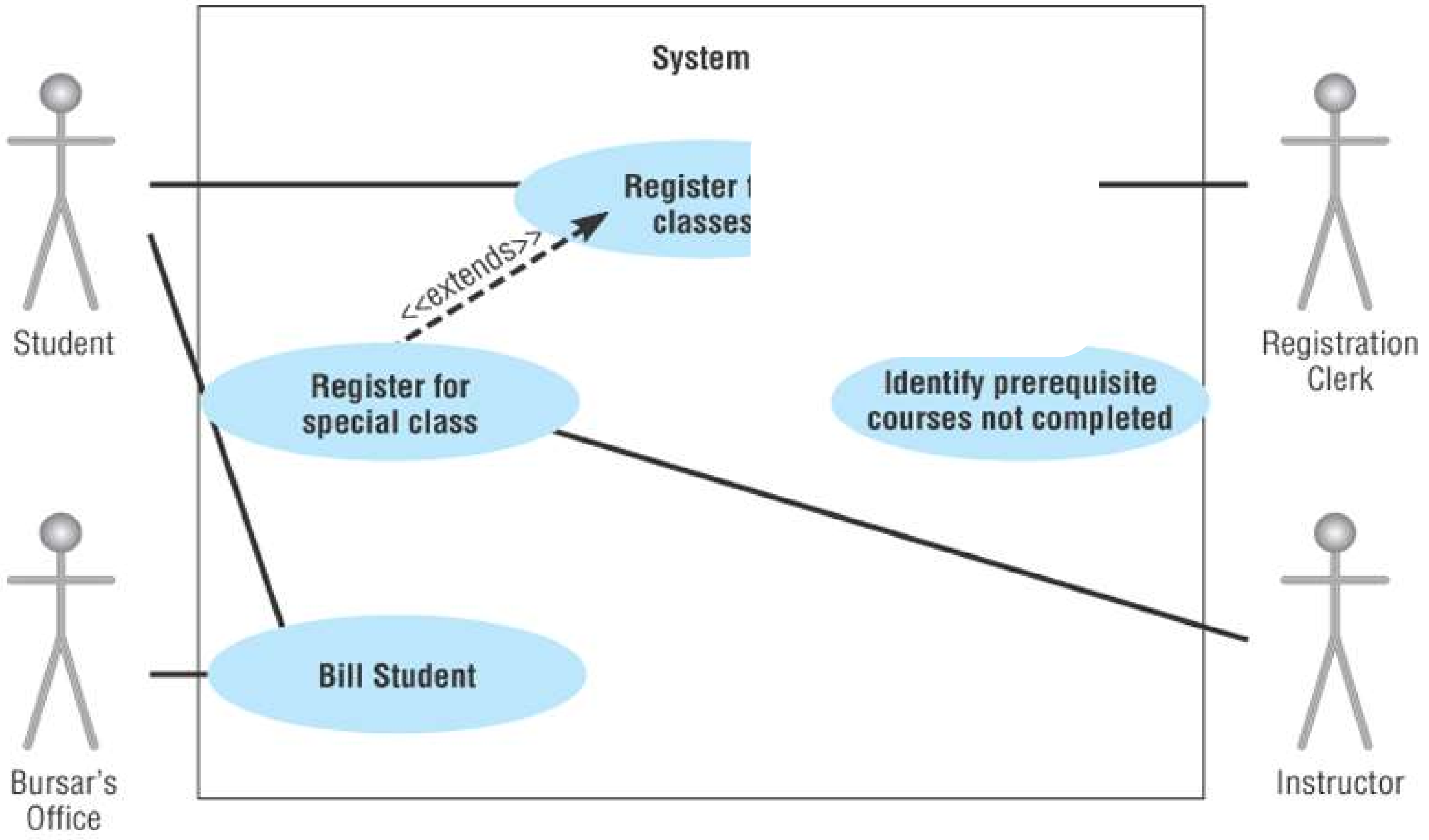
b. Actor: User (student or staff)

c. Input: The user clicks on the borrow book button from the book catalog windows and enter the book title of the book to be borrowed.

d. Output: The user will receive a confirmation of the availability of the book for borrowing and how to collect book and due date to return book or refused to be lend to user with explanatory of the refusal.

Extend Relationship Between Use Cases

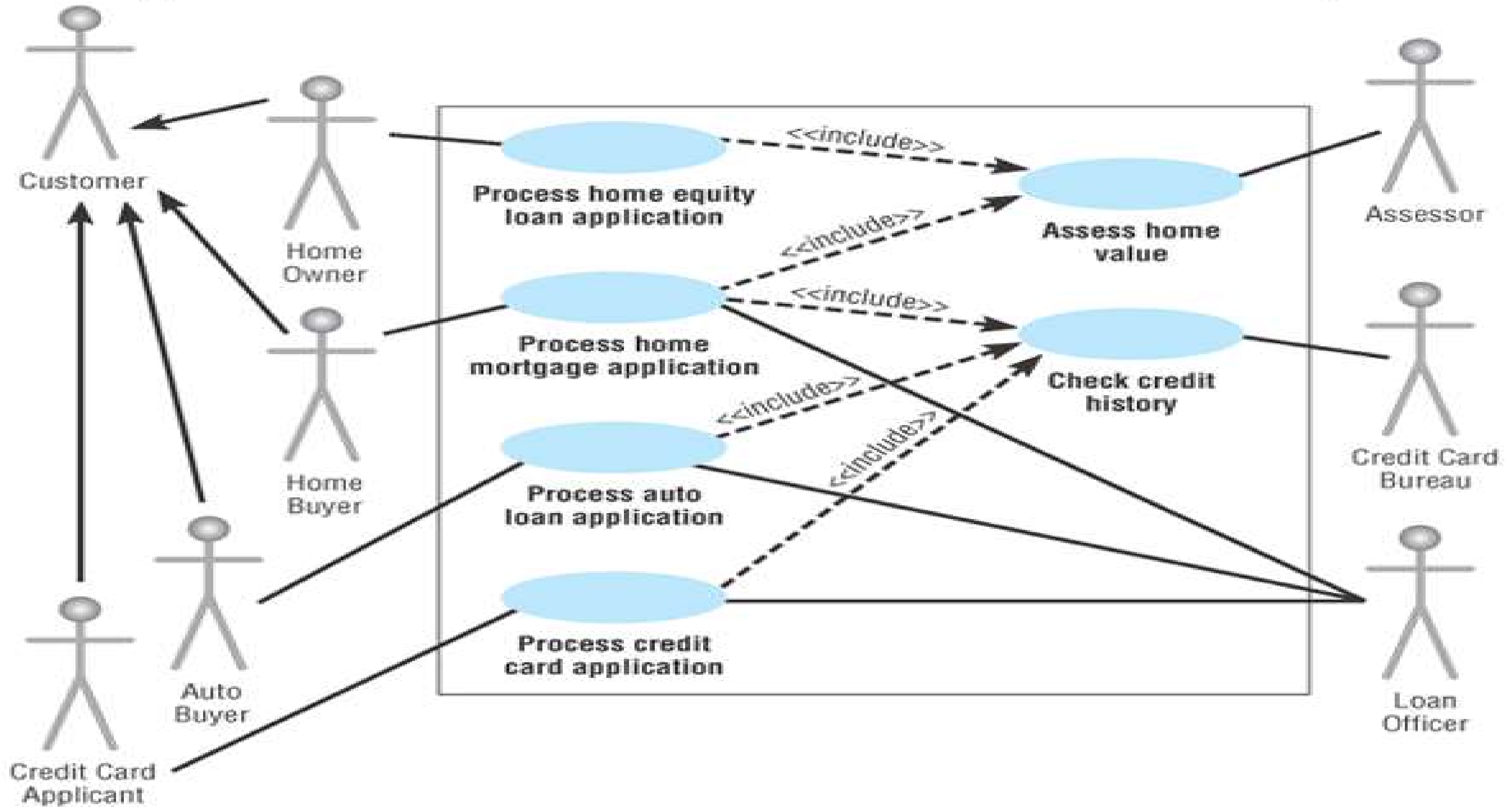
Figure 6.4 A Use Case Diagram for a University Registration System Represented with Microsoft's Visio



Careful: Arrows between Use Cases are NOT data flows!

Include Relationship Between Use Cases

Figure 6.6
Examples of Generalized Use Cases and Include Relationships



Logical Design – Database design

- **Database Management Tool for Database Management System.**
  - *Database Management tool is a graphical user interface for developing Database Servers.*
  - *Database Management System is a software package with computer programs that control the creation, maintenance, and use of a database.*
  - *It allows the development of databases for various applications by database administrators (DBAs) and other specialists.*
- A database is an integrated collection of data records, files, and other objects.
- A DBMS allows different user application programs to concurrently access the same database.

Logical Design – Database design

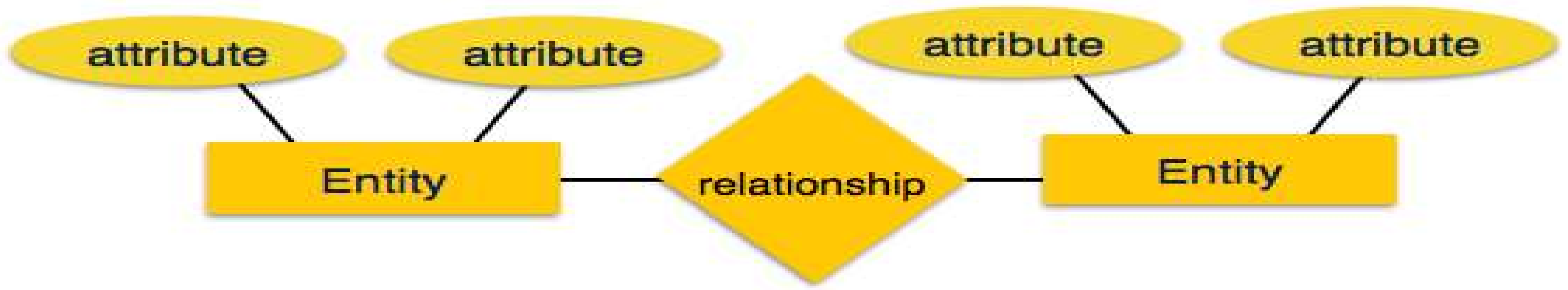
Database Management Tool	Database Server / DBMS
Microsoft Access interface	Microsoft Access Database
SQL Database	MySQL Server,
Microsoft SQL Management Studio	Microsoft SQL Server
phpMyAdmin	MySQL web based
Oracle interface	Oracle
Etc.	Etc.

Logical Design – Database design

- Creating a Database for the system using Microsoft Access
  - *Tables / Entities*
  - *Forms*
  - *Records*
  - *Data Types*
  - *Data / Attributes*

Entity-Relationship Diagram–ETD

- It is a technique used in database design that helps describe the relationship between various entities of an organization.
- ER Model is based on the notion of real-world entities and relationships among them.
- While formulating real-world scenario into the database model, the ER Model creates entity set, relationship set, general attributes and constraints.
- ER Model is based on −
  - ***Entities*** *and their attributes.*
  - ***Relationships*** *among entities.*

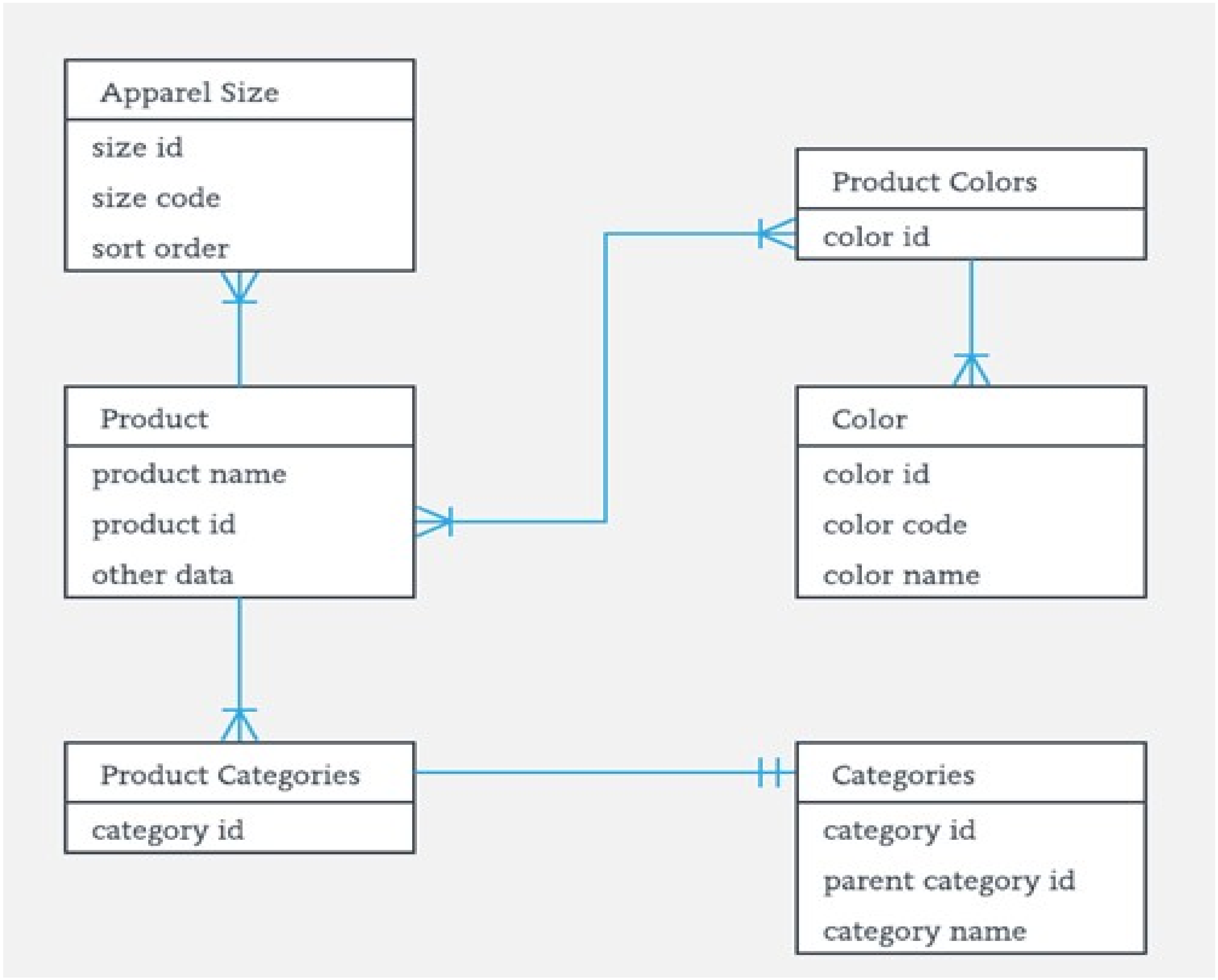


Entity Relationship Diagram – ETD

- **Entity** − An entity in an ER Model is a real-world entity having properties called **attributes**. Every **attribute** is defined by its set of values called **domain**.
- **Relationship** − The logical association among entities is called ***relationship***. Relationships are mapped with entities in various ways.
- Mapping Cardinalities defines the number of entities in one entity set, which can be associated with the number of entities of other set via relationship set.
  - *one to one*
  - *one to many*
  - *many to one*
  - *many to many*

Entity - Relationship Diagram

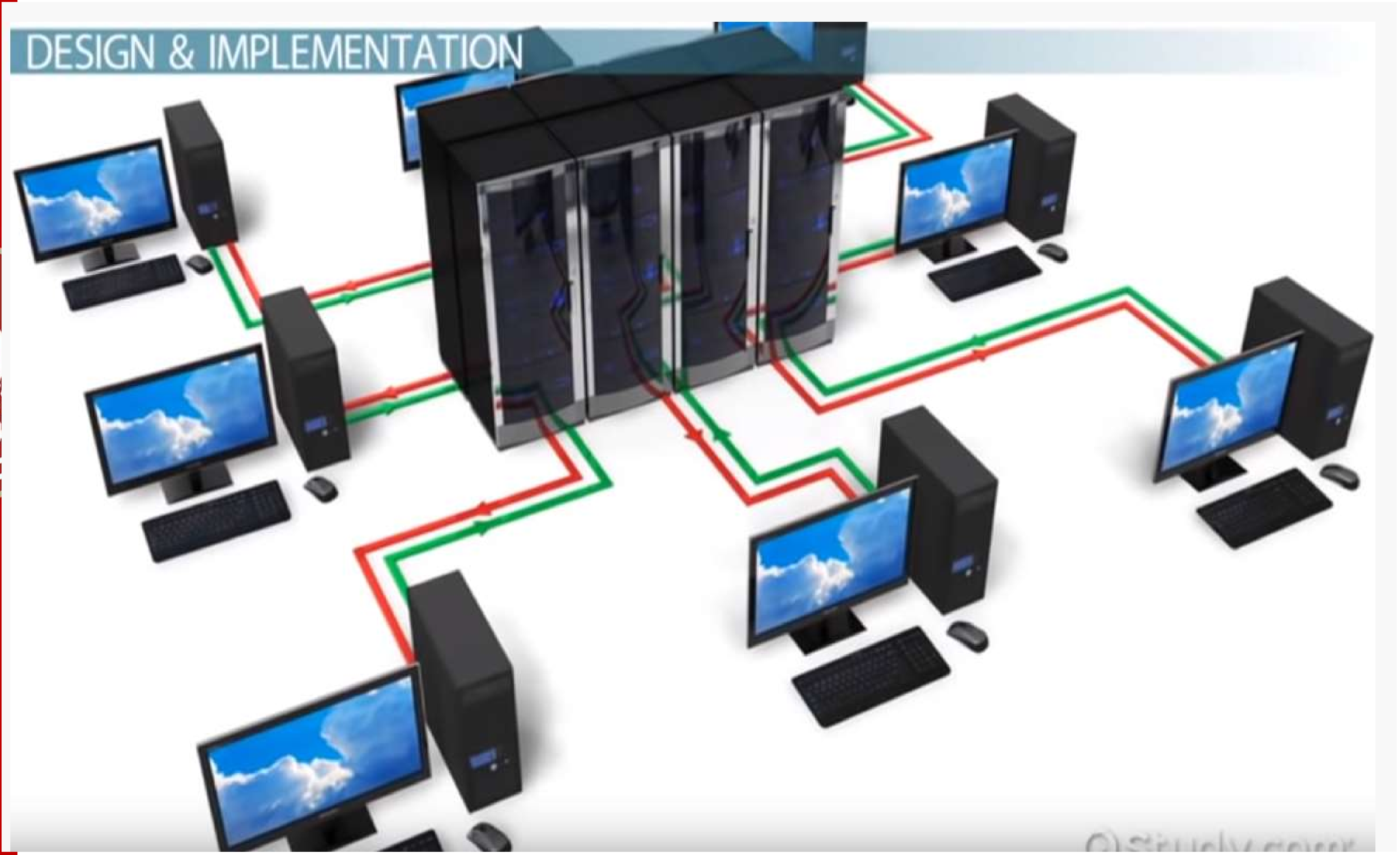
- Tool that graphically shows connections between entities in system



Physical design

- Physical design relates to the actual input and output processes of the system. It focuses on how data is entered into a system, verified, processed, and displayed as output.
- It produces the working system by defining the design specification that specifies exactly what the candidate system does. It is concerned with user interface design, process design, and data design.
- It consists of the following steps −
  - *Specifying the input/output media, designing the database, and specifying backup procedures.*
  - *Planning system implementation.*
  - *Devising a test and implementation plan, and specifying any new hardware and software.*
  - *Updating costs, benefits, conversion dates, and system constraints.*

Phyiscal design



Documentation Control

- Documentation is a process of recording the information for any reference or operational purpose. It helps users, managers, and IT staff, who require it. It is important that prepared document must be updated on regular basis to trace the progress of the system easily.
- After the implementation of system if the system is working improperly, then documentation helps the administrator to understand the flow of data in the system to correct the flaws and get the system working.
- Programmers or systems analysts usually create program and system documentation. Systems analysts usually are responsible for preparing documentation to help users learn the system. In large companies, a technical support team that includes technical writers might assist in the preparation of user documentation and training materials.

Documentation Control

Advantages

- It can reduce system downtime, cut costs, and speed up maintenance tasks.
- It provides the clear description of formal flow of present system and helps to understand the type of input data and how the output can be produced.
- It provides effective and efficient way of communication between technical and nontechnical users about system.
- It facilitates the training of new user so that he can easily understand the flow of system.
- It helps the user to solve the problems such as troubleshooting and helps the manager to take better final decisions of the organization system.
- It provides better control to the internal or external working of the system.

Documentation - Types

- When it comes to System Design, there are following four main documentations −
  - ❑Program documentation
  - ❑System documentation
  - ❑Operations documentation
  - ❑User documentation
- **Program Documentation**
- It describes inputs, outputs, and processing logic for all the program modules.
- The program documentation process starts in the system analysis phase and continues during implementation.
- This documentation guides programmers, who construct modules that are well supported by internal and external comments and descriptions that can be understood and maintained easily.

Documentation - Types

Operations Documentation

- Operations documentation contains all the information needed for processing and distributing online and printed output. Operations documentation should be clear, concise, and available online if possible.
- It includes the following information −
- Program, systems analyst, programmer, and system identification.
- Scheduling information for printed output, such as report, execution frequency, and deadlines.
- Input files, their source, output files, and their destinations.
- E-mail and report distribution lists.
- Special forms required, including online forms.
- Error and informational messages to operators and restart procedures.
- Special instructions, such as security requirements.

Documentation - Types

- User Documentation
- It includes instructions and information to the users who will interact with the system.
  - For example, user manuals, help guides, and tutorials. User documentation is valuable in training users and for reference purpose. It must be clear, understandable, and readily accessible to users at all levels.

A user documentation should include −

- A system overview that clearly describes all major system features, capabilities, and limitations.
- Description of source document content, preparation, processing, and, samples.
- Overview of menu and data entry screen options, contents, and processing instructions.
- Examples of reports that are produced regularly or available at the user's request, including samples.
- Security and audit trail information.
- Explanation of responsibility for specific input, output, or processing requirements.
- Procedures for requesting changes and reporting problems.
- Examples of exceptions and error situations.
- Frequently asked questions (FAQs).
- Explanation of how to get help and procedures for updating the user manual.

Documentation - Types

- **System Documentation**
- System documentation serves as the technical specifications for the IS and how the objectives of the IS are accomplished. Users, managers and IS owners need never reference system documentation. System documentation provides the basis for understanding the technical aspects of the IS when modifications are made.
  - *It describes each program within the IS and the entire IS itself.*
  - *It describes the system's functions, the way they are implemented, each program's purpose within the entire IS with respect to the order of execution, information passed to and from programs, and overall system flow.*
  - *It includes data dictionary entries, data flow diagrams, object models, screen layouts, source documents, and the systems request that initiated the project.*
  - *Most of the system documentation is prepared during the system analysis and system design phases.*
  - *During systems implementation, an analyst must review system documentation to verify that it is complete, accurate, and up-to-date, and including any changes made during the implementation process.*