**System Design:**

**Definition:** System design is the process of defining the architecture, components, modules, interfaces, and data for a system to satisfy specified requirements. It is the phase that bridges the gap between the conceptualization of the system and its implementation.

The design phase in the Software Development Life Cycle (SDLC) is a crucial step where the system's architecture and specifications are defined in detail. This phase comes after the requirement analysis and before the implementation phase. The primary goal is to transform the requirements into a design that serves as a blueprint for the construction of the software.

**General Information about Design Phase:**

* **Key Activities:**
  + Architectural design
  + Database design
  + Interface design
  + Security design
  + Network design
* **Inputs:**
  + Requirements specification
  + System architecture guidelines
  + Functional specifications
* **Outputs:**
  + Detailed design specifications
  + System architecture diagram
  + Database schema
  + Interface prototypes
* **Review Process:**
  + Design documents are typically reviewed by stakeholders, including developers, architects, and sometimes end-users, to ensure that the proposed design aligns with the requirements and is feasible to implement.
* **Validation and Verification:**
  + Validation ensures that the design meets the specified requirements.
  + Verification involves checking that the design documents are consistent, complete, and accurate.
* **Iterative Nature:**
  + The design phase may involve iterations, especially when changes are required based on reviews, feedback, or evolving project needs.

**Types of System Design:**

1. **High-Level Design:**
   * *Description:* It defines the overall system architecture. This includes major components, their relationships, and the high-level flow of data and control.
   * *E-commerce Example:* High-level design for an e-commerce application involves defining components like user interface, product catalog, shopping cart, and payment processing.
2. **Low-Level Design:**
   * *Description:* It provides detailed information about each component defined in high-level design. It includes algorithms, data structures, database tables, and interfaces.
   * *E-commerce Example:* Low-level design for the payment processing component would detail algorithms for transaction validation, error handling, and database interactions.

**Logical System Design:**

**Definition:** Logical system design involves the conceptualization and specification of the data processing functionalities of a system. It focuses on what the system must do without specifying how it will be accomplished.

**Types of Logical System Design:**

1. **Data Flow Diagrams (DFD):**
   * *Description:* DFDs represent the flow of data within the system. It shows how data moves through processes, data stores, and external entities.
   * *E-commerce Example:* A DFD for an e-commerce application illustrates how user registration data flows from the user interface to the user database.
2. **Entity-Relationship Diagram (ERD):**
   * *Description:* ERDs model the entities within a system and their relationships. It helps define the structure of the database.
   * *E-commerce Example:* An ERD for an e-commerce application identifies entities like users, products, and orders and shows their relationships.

3. **Data Dictionary:**

* **Description:** A data dictionary is a centralized repository of metadata, containing definitions and descriptions of data elements used in a system. It provides a comprehensive view of the data's meaning, format, relationships, and usage within the system.
* **E-commerce Example:** In an e-commerce application's data dictionary, entries could include:

1. **User Table:**
   * *Fields:* user\_id, username, email, password\_hash
   * *Description:* Defines the structure of the user table with details about each field, such as the data type and constraints.
2. **Product Table:**
   * *Fields:* product\_id, product\_name, price, description
   * *Description:* Outlines the fields in the product table and their characteristics, facilitating a clear understanding of the product-related data.
3. **Order Table:**
   * *Fields:* order\_id, user\_id, order\_date, total\_amount
   * *Description:* Specifies the attributes and relationships in the order table, enabling developers to work with order-related data.

4. **Process Descriptions:**

* **Description:** Process descriptions provide detailed information about the steps, actions, and decisions involved in a specific process within a system. It outlines how data is transformed and manipulated during the execution of a particular operation.
* **E-commerce Example:** Consider the process of "Checkout" in an e-commerce application:

1. **Process: Checkout:**
   * *Steps:*
     + User clicks on "Checkout" after adding items to the cart.
     + System validates user authentication.
     + User selects shipping address and payment method.
     + System calculates the total amount.
     + User confirms the order.
     + System generates an order confirmation and deducts the items from inventory.
   * *Decisions:*
     + If user authentication fails, redirect to login.
     + If payment fails, prompt the user to retry or choose an alternative method.
   * *Data Manipulation:*
     + Update the order table with details of the transaction.
     + Deduct the purchased items from the product inventory.
     + Generate an order confirmation email for the user.

**Physical System Design:**

**Definition:** Physical system design involves specifying the physical configuration of the hardware and software components, their integration, and their deployment.

**Types of Physical System Design:**

1. **Database Design:**
   * *Description:* It involves defining the structure of the database, including tables, relationships, and constraints.
   * *E-commerce Example:* Designing the database for an e-commerce application includes tables for users, products, orders, and transactions.
2. **Network Design:**
   * *Description:* Network design defines how different components of the system will communicate, considering factors like bandwidth, latency, and security.
   * *E-commerce Example:* Network design for an e-commerce platform ensures secure communication between the web server and the database server.
3. **UI Design:**

* **Description:** UI (User Interface) design is the process of creating visually appealing and user-friendly interfaces for software applications. It involves designing the look, feel, and interaction of the application to enhance the user experience.
* **E-**commerce Example: Layout, Color scheme , Typography etc.