

## UNIT 4 - DESIGN PROCESS

Design Rules? → essential guidelines & principles to improve usability & UX  
→ ensures consistency, efficiency & clarity in interaction b/w user & system

Principles that support usability

- ① **Learnability**: Predictability, Synthesizable, familiar, consistent, General.
- ② **Flexibility**: Dialogue Initiative, Multithreading, Task Migrability,  
(text, speak) Substitutivity, Customizability. (rearranging, apns on phone)
- ③ **Robustness**: Observability, Recoverability, Responsiveness, Task conformance

Principles of learnability :- easy for new users to understand.

Predictability - user should be able to anticipate results of their actions.

Synthesizability - what effect of past operation is on current operation

Generalizability - user should be able to apply knowledge.

Consistency - interface should be uniform manner throughout.

Familiarity - users prior knowledge should be applicable.

Example - Google search, Microsoft Office suite ie Word

Principles of flexibility :- multiple ways to interact.

Multithreading - multiple tasks to perform simultaneously.

Task Migrability - users can choose whether the system or they themselves will handle specific tasks. eg spelling check (red underline)

Example - Adobe Photoshop, Spotify.

Principles of Robustness :- support for mistakes & progress.

Example - Google Docs, MS Word.

Design standards → documented agreements used to ensure consistency & quality in UI. These are often established by industry leaders/ organizations such as ISO. That are:

- ↳ Improves product reliability & ease of use.
- ↳ Ensures accessibility for all users, incl. those with disabilities.
- ↳ Reduces the learning curve for new users by applying familiar design patterns.

Design Guidelines → guidelines are set of rules & recommendations and best practices that help for better design.

⇒ 8 Golden rules

The golden rule design - "understand your materials". ↗ i-phone

Understand computer - understand its limitations, capacity, tools & platforms

Understand people - understand psychological, social aspects, human errors.

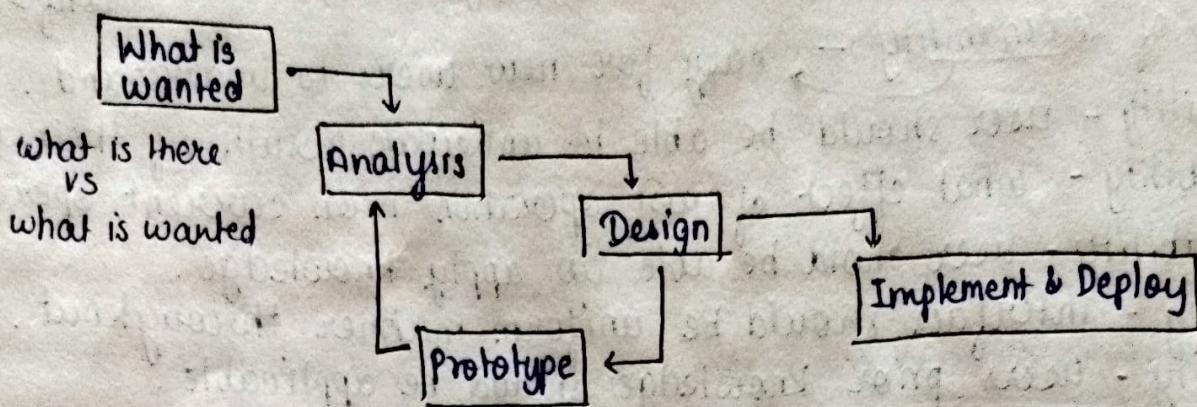
Design - process of creating a plan or blueprint of something, whether it's an object, a system or an experience.

It combines creativity with practical problem solving to enhance interaction with surroundings.

## What is Interaction Design? (IxD)

- focus on creating engaging interfaces that facilitate meaningful interactions between users and digital systems.
  - it aims to enhance UX by considering how users interact with technology, VCD, feedback, consistency, affordance, flexibility, understandability, goals, etc.
- Eg. - Mobile apps, websites

## Software Design Process



Types of Scenarios

- ① Goal & Task Based
- ② Elaborated
- ③ Full Scale Task

## User Focus

- User focus is a key principle in HCI, which means designing technology with the needs & preferences of users' mind. or
- Designing with user-centered approach means prioritizing user needs, behaviors & pain points. This can be achieved by:

### ① Know the users - Who are they?

Probably not like you! (design for others not you)  
Talk to Them  
Watch users  
Use Your Imagination

### ② Persona - fictional character that represents a specific segment of your target audience. It is created based on research & data gathered from actual users. Helps to Guide Design Decisions

Eg: Persona might include details like: User Representation

Name: Betty Age: 37 Job: Warehouse Manager Background - Has

worked in a company for 12 yrs, has 2 children & feels overwhelmed by new tech

### ③ Cultural Probes - tools used to gather qualitative data about users' lives, behaviours & environments without direct observation.

Eg. A disposable camera for user to take pic of their daily activities.  
A diary for user to jot down thoughts/feelings about their interactions with technology.

A voice recorder to capture spontaneous thoughts throughout the day.

## Scenarios

- Powerful tools used in interaction design to describe how users will interact with a system in specific situations.
- Scenarios are detailed narratives that outline a specific use case / interaction between a user & a system. They describe the context, actions, and outcomes of user interaction, often presented in stories.
- Purpose - helps designer think through UX & how user will engage in system
- Components - user context → user actions → system responses → outcomes
- Benefits of using Scenarios - Clarity, Identifying user needs, supporting interactive designs, highlight potential issues
- Eg. Brian & the Personal Movie Player - Brian wants to evaluate if the film "Moments of Significance" is suitable for his friend Alison, who typically avoids "arty" films. He decides to use his work comp.

and download the film because it has faster internet connection.

Narrative -

Decision to Download → Download the film → Transferring the film →  
Watching the film → Unexpected encounter → Invitation to cinema.  
Alison comes & connects speakers

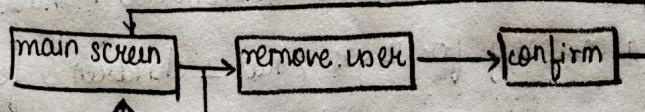
## Navigation Design

Process of navigating a system, application or websites using dialog box, widgets, hypertext or hypermedia, etc.

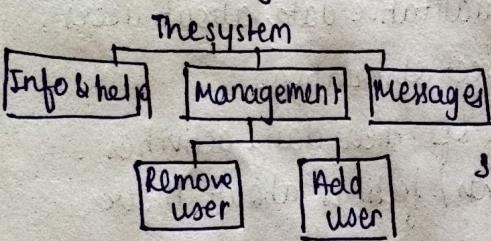
Constraints of ND

- ① Widgets - choose appropriate widget
- ② Screen/windows - find things on screen & understand logical grouping of buttons
- ③ Navigation within application - what will happen next, if button is pressed.
- ④ Environment - facebook is a social app, user must know its environment.

Example - GPS



→ Hierarchical Organisation



some messaging system like facebook, instagram or whatsapp

## Screen Design & Layout

- Involves arranging visual elements on the screen to facilitate interaction & make system easy to use. It focuses on layout, typography, color and user interface components
- Screen Design develop using different techniques such as CRT, LED & Touch Screen, etc.

Layout -

- ① Grouping & structure - logical grouping
- ② Order of groups & items - natural order for user
- ③ Decoration - design uses boxes that makes grouping clear
  - font style, text, colors that highlight grouping
- ④ Alignments - we read left to right, so normally left aligned
  - and numbers aligned at right / decimal point
- ⑤ White Space - spacing is imp.

- User wants on screen -
- ↳ flexible size of screen
  - ↳ Supported language
  - ↳ screen looks & Reliability
  - ↳ simple english
  - ↳ easy understandable symbols
  - ↳ simple & Attractive
- User does not want on screen -
- ↳ unclear option
  - ↳ Misleading Headlines
  - ↳ Inefficient Results
  - ↳ Improper Graphical Elements
  - ↳ Unnecessary Data
  - ↳ Poor Arrangements
  - ↳ Lack of Design features
  - ↳ Poor Quality Presentation
  - ↳ Too bright colors

## Prototyping techniques

Prototyping - process of creating early versions of a product to test ideas and see how they work. It helps designer figure out what works well & what needs improvement before making the final product.

Types:-

- ① Low-fidelity Prototypes -  
Simple & basic versions of product. They focus on layout & main function rather than detailed design  
features - Basic Design, Quick to make, focus on functionality.  
examples - Paper prototypes, Wireframes, Storyboard, sketches

- ② Medium-fidelity Prototypes -

more than low but less than high fidelity

features - Clickable areas, Basic Design, focus on user flow

examples - Clickable wireframes, storyboard-based prototypes.

- ③ High-fidelity Prototypes -

more detailed & look closer to final product. They include real design elements, colors, images & interactive features.

features - Detailed design, Interactive elements, Takes more time

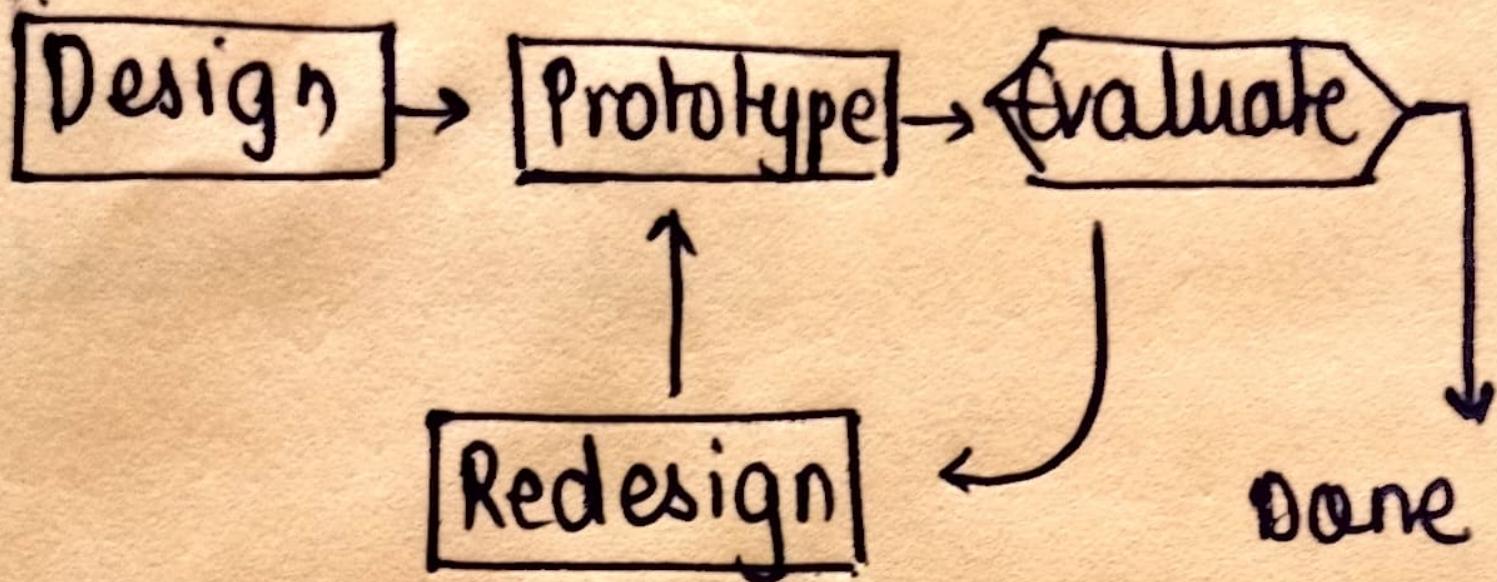
examples - Interactive Websites (prototype of online store), Mobile App Mockups (using tools like Adobe XD or Figma).

## Wireframing

Blueprint of website / app. It shows <sup>how</sup> everything will be laid out & how users will interact with it, but doesn't include detailed design or colors.

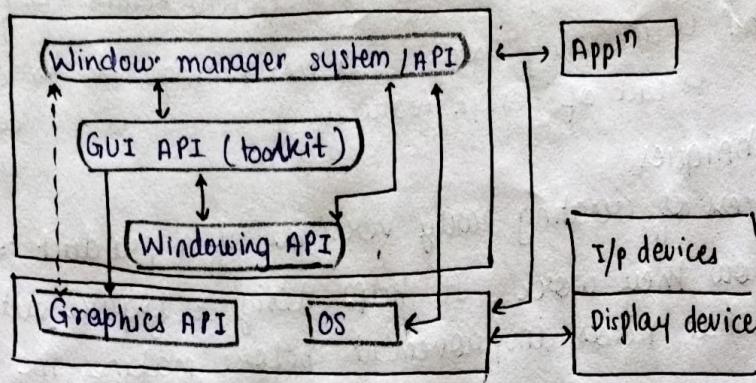
features - Simple Design, focus on layout, User flow.

Tools - figma, Adobe XD, Miro, MockFlow, etc.



## Understanding UI layer & Its execution framework.

UI layer is where user interacts with system. It includes all the visual components like buttons, forms and menus, as well as the logic that connects elements to the system's functionality.



## Model View Controller (MVC) framework.

Popular architectural pattern used to separate concerns in software development.  
It divides application into 3 parts :-

- ① Model    ② View    ③ Controller

**Model** - It is like brain of the application. It handles all data & logic.

→ It stores info, like user data / product details & knows how to retrieve or change that data.

→ Eg - In a game, model would keep track of players score & levels.

**View** - It is what users see on their screen. It is the UI.

→ It displays data <sup>from</sup> the model in a way that user can understand & interact with.

→ Eg - In a weather app, the view shows the current temperature & condition.

**Controller** - It manages the flow of info betw Model & View.

→ It takes user input (clicks / typing) updates Model based on i/p & tells View to update what user sees

→ Eg. Online store , Add to cart

## Example- Online food Ordering System

① Model - keeps track of restaurants, menus, food items & orders.

Tech used - MySQL, MongoDB, Node.js with Express, ASP.NET

② View - Displays menu, order forms & user profiles.

Tech used - React, Angular, Vue.js, HTML/CSS/JS

③ Controller - Takes user request, updates model & refreshes View

Tech used - Node.js, Flask, Ruby on Rails.

