User Interface Design: -
- The main goal of UI is to display & obtain needed information in an accessible, efficient manner.
- A well defined UI has visual appeal that motivates users to use application
- To view UI design as a creativity process it constituents are the following
  1. A curious & imaginative mind
  2. A broad Background & fundamental knowledge of existing tools & methods
  3. An enthusiastic desire to do a complete & thorough job of discovering solutions once a problem has been defined
  4. Being able to deal with uncertainty & ambiguity & to defer premature closure

Designing View Layer classes: -
- The view layer objects are responsible for two major aspects of applications:
  1. Input – responding to user interaction
  2. Output – displaying or printing business objects
- The process of designing view layer classes is divided into four major activities
  1. Macro level UI design process – identifying view layer objects
  2. Micro level UI design activities – constitute of mainly two steps such as
    i. Designing the view layer objects by applying design axioms & corollaries
    ii. Prototyping the view layer interface
  3. Testing usability & user satisfaction
  4. Refining and iterating the design

Macro–Level Process: -
- The main goal of this level is identifying view class by analyzing use cases. The interface object handles all communication with actor but processes no business rules or object storage activities
- The view layer macro process consists of two steps as shown in diagram
  1. Determine if class interacts with a human actor
    a. Identify view (interface) objects for class
    b. Define relationships among the view (interface) objects
  2. Iterate and refine
- The advantage of utilizing use cases in identifying & designing view layer objects is that the focus centers on the user and including users as part of the planning & design is the best way to ensure accommodating them

Micro – Level Process: -
- A user-centered interface replicates the user's view of doing things by providing the outcomes users expect for any action. The micro level process of designing view objects has two steps
  1. For very interface object identified in macro UI design process, apply micro – level UI design rules & corollaries to develop the UI
  2. Iterate and Refine
- UI design rule 1: Making the interface simple (Application of corollary 2) – Labels, static text, check boxes, group boxes & option buttons often clutter the interface & take up twice the room mandated by actual data.
- The following factors should be considered while evaluating the impact:
  Every additional feature potentially affects performance, complexity, stability, maintenance & support costs of an application
  It is harder to fix a design problem after release of product because users may adapt or even become dependent on a peculiarity in design
  Simplicity is different from being simplistic. Making something simple to use often requires a good deal of work & code
Features implemented by a small extension in application code do not necessarily have a proportional effect in a UI

- **UI design rule 2**: Making the interface Transparent and Natural (Application of Corollary 4) – It implies that there should be strong mapping between user’s view of doing things & UI classes. A metaphor or analog, relates two otherwise unrelated things by using one to denote the other. A goal of UI design is to make user interaction with computer as simple and natural as possible

- **UI design rule 3**: Allowing Users to be in Control of s/w. A mode is a state that excludes general interaction or otherwise limits the user to specific interactions. Some of ways to put users in control are the following:
  - Make the interface forgiving
  - Make the Interface Visual
  - Provide immediate feedback
  - Avoid modes
  - Make the interface consistent

There can be some of the modes that can be used in the user interface – Modal dialog, spring loaded modes and tool–driven modes

**Purpose of a View Layer Interface:**

- UI can employ one or more windows. Each window should serve a clear, specific purpose. Windows commonly are used for the following purposes
  - **Forms & data entry windows** – provide access to data that users can retrieve, display & change in the application
  - **Dialog Boxes** – display status information or ask users to supply information or make a decision before continuing with a task. A typical feature of a dialog box is OK button
  - **Application windows (main windows)** – is a container of application objects or icons. In other words, it contains an entire application with which users can interact

**Guidelines for Designing forms & Data Entry windows:** First identify the information want to display or change on designing, then identify the tasks users need to work with data entry window

- Use an existing paper form such as printed invoice as starting point of design
- If too much information to be fit on a screen, consider using a main window with optional smaller windows that users can display on demand or; using a window with multiple pages
- Put required or frequently entered information toward top & left side of form, entering optional or seldom –entered information towards bottom & right side
- When information is positioned vertically, align fields at their left edges (in western countries)
- Arrange controls in the sequence users expect to enter data
- Put similar or related information together use visual effects to emphasize the grouping

**Guidelines for Designing Dialog Boxes & Error Messages:** A dialog box provides an exchange of information or a dialog between the user & application. If dialog box is for an error message, then

- The error message should be positive E.g.: Enter following date format – mm/dd/yyyy
- The error message should be constructive E.g.: Press the undo button and try again
- Error message should brief and meaningful E.g.: Error – unexpected data format
- Orient the controls in dialog box in the direction people read

**Guidelines for Designing Command Buttons Layout:** Layout major command button either stacked along the upper – right border of dialog box or lined up across the bottom of dialog box. Positioning buttons on left border is very popular in Web interfaces. For easy readability, make buttons a consistent length. Consistent visual and operational styles will allow users to transfer their knowledge and skills more easily

**Guidelines for Designing Application windows:** An app., window usually contains common drop–down menus. While it is not required for all applications, then get the generally followed pattern of menus – file, edit, view & other command menus, window, help and toolbars & status bars

**Guidelines for Using colors:** As general practice, do not let color be only visual cue, use an animated button, a sound package or a message box and use color in effective manner

- Use identical or similar colors to indicate related information
- For an object background, use a contrasting but complementary color
- Use bright colors to call attention to certain elements on screen and use dim colors to make other elements less noticeable
- Use colors consistently within each window and among all windows in application
- Using too many colors can be visually distracting & make application less interesting
- Allow the user to modify the color configuration of an application

**Guidelines for Using Fonts:** Consistency is the key to an effective use of fonts and color in our interface. Most commercial applications use 12 –point system font for menus & 10 –point system font in dialog boxes. The following guidelines can help use fonts to best convey information
◊ Use commonly installed fonts, not specialized fonts that user might not have on their machines
◊ Use bold for control labels, so they will remain legible when the object is dimmed
◊ Use fonts consistently within each form & among all forms in our application
◊ Using too many font styles, sizes & colors can be visually distracting and should be avoided
◊ To emphasize text, increase its font size relative to other words on form or use a contrasting color. Avoid underlines; they can be confusing & difficult to read on screen

**Prototyping the User Interface:**

- **Rapid prototyping** encourages the incremental development approach, “grow, don’t build”. Prototyping involves a number of iterations, where we add a little more to the application and we understand the problem a little better, we make more improvements
- With use of **CASE tools**, operational s/w using visual prototyping or normal development tools.
- **Visual & rapid prototyping** is a valuable asset in many ways. First, it provides and effective tool for communicating the design. Second, it can help you define task flow and better visualize the design. Finally, it provides a low–cost vehicle for getting user input on a design.
- **Creating a user interface** generally consists of three steps
  1. Create the user interface objects (such as buttons, data entry fields)
  2. Link or assign the appropriate behaviors or actions to these user interface objects & their events
  3. Test, debug, then add more by going back to step 1

**General Considerations:**


**OOSDLC:**

- Object Oriented Analysis (OOA) – Actors, classes, object Design + Documentation. – Transformation of design in terms of classes, objects & interface.
- Object Oriented Design (OOD) – Operation + method + Object
- Object Oriented Implementation – CBD, RAD
- Prototype

**Objects relationships** – Associative, cardinality, Aggression

**Classes** – Universal entities, simplicity establishing related features.