

USN 

--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--



**Internal Assessment Test 3 – December 2022**

Sub:	User Interface Design					Sub Code:	18CS734	Branch:	ISE		
Date:	26/12/2022	Duration:	90 min's	Max Marks:	50	Sem/Sec:	VII A, B & C			OBE	
<b><u>Answer any FIVE FULL Questions</u></b>								MARKS	CO	RBT	
1	Explain Radio buttons and list boxes selection controls in detail.						10	CO3	L2		
2	Explain window operations in detail?						10	CO3	L2		
3	Discuss the purpose of Drop down or pop-up list boxes along with their advantages and disadvantages.						10	CO2	L2		
4	Give short notes on windows presentation styles and explain various window management techniques?						10	CO2	L2		
5	Explain the purpose of prototypes. Discuss any two kinds of prototypes with their importance to the system developers.						10	CO3	L3		
6	Explain Cognitive Walkthroughs, think Aloud evaluation and usability tests conducted in UID.						10	CO3	L2		

Faculty Signature

CCI Signature

HOD Signature

1 Explain Radio buttons and list boxes selection controls in detail.

Radio Buttons

• Description:

— A two-part control consisting of the following:

- Small circles, diamonds, or rectangles.
- Choice descriptions.

— When a choice is selected:

- The option is highlighted.
- Any existing choice is automatically highlighted and deselected.

• Purpose:

— To set one item from a small set of mutually exclusive options (2 to 8).

• Advantages:

- Easy-to-access choices.
- Easy-to-compare choices.
- Preferred by users.

Disadvantages:

- Consume screen space.
- Limited number of choices.

Proper usage:

- For setting attributes, properties, or values.
- For mutually exclusive choices (that is, only one can be selected).

- Where adequate screen space is available.
- Most useful for data and choices that are:
  - Discrete.
  - Small and fixed in number.
  - Not easily remembered.
  - In need of a textual description to meaningfully describe the alternatives.
- Most easily understood when the alternatives can be seen together and compared to one another.
- Never changed in content.
- Do not use:
  - For commands.
  - Singly to indicate the presence or absence of a state.

### Choice Descriptions

Provide meaningful, fully spelled-out choice descriptions clearly describing the values or effects set by the radio buttons.

Display in a single line of text.

Display using mixed-case letters, using the sentence style.

Position descriptions to the right of the button. Separate them by at least one space from the button.

When a choice is conditionally unavailable for selection, display the choice description grayed out or dimmed.

Include a none choice if it adds clarity.

### Size

Show a minimum of two choices, a maximum of eight.

### Defaults

When the control possesses a state or affect that has been predetermined to have a higher probability of selection than the others, designate it as the default and display its button filled in.

When the control includes choices whose states cannot be predetermined, display all the buttons without setting a dot, or in the indeterminate state.

When a multiple selection includes choices whose states vary, display the buttons in another unique manner, or in the mixed value state.

### Structure

A columnar orientation is the preferred manner of presentation.

Left-align the buttons and choice descriptions.

If vertical space on the screen is limited, orient the buttons horizontally.

Provide adequate separation between choices so that the buttons are associated with the proper description.

— A distance equal to three spaces is usually sufficient.

Enclose the buttons in a border to visually strengthen the relationship they possess.

### Organization

Arrange selections in expected order or follow other patterns such as frequency of occurrence, sequence of use, or importance.

— For selections arrayed top to bottom, begin ordering at the top.

— For selections arrayed left to right, begin ordering at the left.

If, under certain conditions, a choice is not available, display it subdued or less brightly than the available choices.

2 Explain window operations in detail?

### **Active Window**

- ✓ A window should be made active with as few steps as possible.
- ✓ Visually differentiate the active window from other windows.

### **General Guidelines**

- ✓ Design easy to use and learn windowing operations.
  - Direct manipulation seems to be a faster and more intuitive interaction style than indirect manipulation for many windowing operations.

- ✓ Minimize the number of window operations necessary to achieve a desired effect.
- ✓ Make navigating between windows particularly easy and efficient to do.
- ✓ Make the setting up of windows particularly easy to remember.
- ✓ In overlapping systems, provide powerful commands for arranging windows on the screen in user-tailorable configurations.

## Opening a Window

- ✓ Provide an iconic representation or textual list of available windows.
  - If opening with an expansion of an icon, animate the icon expansion.
- ✓ When opening a window:
  - Position the opening window in the most forward plane of the screen.
  - Adapt the window to the size and shape of the monitor on which it will be presented.
  - Designate it as the active window.
  - Set it off against a neutral background.
  - Ensure that its title bar is visible.
- ✓ When a primary window is opened or restored, position it on top.
  - Restore all secondary windows to the states that existed when the primary window was closed.
- ✓ When a dependent secondary window is opened, position it on top of its associated primary window.
  - Position a secondary window with peer windows on top of its peers.
  - Present layered or cascaded windows with any related peer secondary windows.
- ✓ When a dependent secondary window is activated, its primary window and related peer windows should also be positioned at the top.
- ✓ If more than one object is selected and opened, display each object in a separate window.
- ✓ Designate the last window selected as the active window.
- ✓ Display a window in the same state as when it was last accessed.
  - If the task, however, requires a particular sequence of windows, use a fixed or consistent presentation sequence.
- ✓ With tiled windows, provide an easy way to resize and move newly opened windows.

## Sizing Windows

- ✓ Provide large-enough windows to:
  - Present all relevant and expected information for the task.
  - Avoid hiding important information.
  - Avoid crowding or visual confusion.
  - Minimize the need for scrolling.
    - But use less than the full size of the entire screen.
- ✓ If a window is too large, determine:
  - Is all the information needed?

- Is all the information related?
- ✓ Otherwise, make the window as small as possible.
  - Optimum window sizes:
    - For text, about 12 lines.
    - For alphanumeric information, about seven lines.
- ✓ Larger windows seem to have these advantages:
  - They permit displaying of more information.
  - They facilitate learning: Data relationships and groupings are more obvious.
  - Less window manipulation requirements exist.
  - Breadth is preferred to depth (based on menu research).
  - More efficient data validation and data correction can be performed.
- ✓ Disadvantages include:
  - Longer pointer movements are required.
  - Windows are more crowded.
  - More visual scanning is required.
  - Other windows more easily obscure parts of the window.
  - It is not as easy to hide inappropriate data.

## Window Placement

- Considerations:
  - In placing a window on the display, consider:
    - The use of the window.
    - The overall display dimensions.
    - The reason for the window's appearance.
- General:
  - Position the window so it is entirely visible.
  - If the window is being restored, place the window where it last appeared.
  - If the window is new, and a location has not yet been established, place it:
    - At the point of the viewer's attention, usually the location of the pointer or cursor.
    - In a position convenient to navigate to.
    - So that it is not obscuring important or related underlying window information.
  - For multiple windows, give each additional window its own unique and discernible location.
    - A cascading presentation is recommended.
  - In a multiple-monitor configuration, display the secondary window on the same monitor as its primary window.
  - If none of the above location considerations apply, then:
    - Horizontally center a secondary window within its primary window just below the title bar, menu bar, and any docked toolbars.
  - If the user then moves the window, display it at this new location the next time the user opens the window.
    - Adjust it as necessary to the current display configuration.

- Do not let the user move a window to a position where it cannot be easily repositioned.
- ✓ Dialog boxes:
  - If the dialog box relates to the entire system, center it on screen.
  - Keep key information on the underlying screen visible.
  - If one dialog box calls another, make the new one movable whenever possible.

### **Window Separation**

- Crisply, clearly, and pleasingly demarcate a window from the background of the screen on which it appears.
  - Provide a surrounding solid line border for the window.
  - Provide a window background that sets the window off well against the overall screen background.
  - Consider incorporating a drop shadow beneath the window.

### **Moving a Window**

- Permit the user to change the position of all windows.
- Change the pointer shape to indicate that the move selection is successful.
- Move the entire window as the pointer moves.
- If it is impossible to move the entire window, move the window outline while leaving the window displayed in its original position.
- Permit the moving of a window without its being active.

### **Resizing a Window**

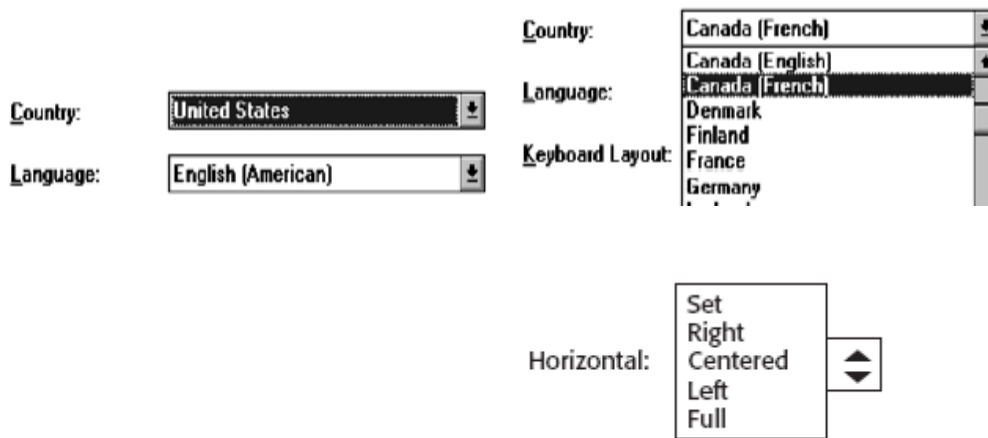
- Permit the user to change the size of primary windows.
  - Unless the information displayed in the window is fixed or cannot be scaled to provide more information.
- Change the pointer shape to indicate that the resizing selection is successful.
- The simplest operation is to anchor the upper-left corner and resize from the lower right corner.
  - Also permit resizing from any point on the window.
- Show the changing window as the pointer moves.
  - If it is impossible to show the entire window being resized, show the window's outline while leaving the window displayed in its original position.
- When window size changes and content remains the same:
  - Change image size proportionally as window size changes.
- ✓ If resizing creates a window or image too small for easy use, do one of the following:
  - Clip (truncate) information arranged in some logical structure or layout when minimum size is attained, or
  - When no layout considerations exist, format (restructure) information as size is reduced, or
  - Remove less useful information (if it can be determined), or

- When minimum size is attained, replace information with a message that indicates that the minimum size has been reached and that the window must be enlarged to continue working.
- Permit resizing a window without its being active.

### 3 Discuss the purpose of Drop down or pop-up list boxes along with their advantages and disadvantages.

- Description
  - A single rectangular control that shows one item with a small button to the right side.
  - The button provides a visual cue that an associated selection box is available but hidden.
  - When the button is selected, a larger associated box appears, containing a list of choices from which one may be selected.
  - Selections are made by using the mouse to point and click.
  - Text may not be typed into the control.
- Purpose:
  - To select one item from a large list of mutually exclusive options when screen space is limited.
- Advantages:
  - Unlimited number of choices.
  - Reminds users of available options.
  - Conserves screen space.
- Disadvantages:
  - Requires an extra action to display the list of choices.
  - When displayed, all choices may not always be visible, requiring scrolling.
  - The list may be ordered in an unpredictable way, making it hard to find items.
- Proper usage:
  - For selecting values or setting attributes.

- For choices that are mutually exclusive (only one can be selected).
- Where screen space is limited.
- For data and choices that are:
  - Best represented textually.
  - Infrequently selected.
  - Not well known, easily learned, or remembered.
  - Ordered in a unpredictable fashion.
  - Large in number.
  - Variable or fixed in list length.
- Use drop-down/pop-up lists when:
  - Screen space or layout considerations make radio buttons or single-selection list boxes impractical.
  - The first, or displayed, item will be selected most of the time.
- Do not use a drop-down list if it important that all options be seen together.



4 Give short notes on windows presentation styles and explain various window management techniques?

- The presentation style of a window refers to its spatial relationship to other windows.
- There are two basic styles, commonly called tiled or overlapping.

### Tiled Windows

- Tiled windows derive their name from common floor or wall tile. Tiled windows appear in one plane on the screen and expand or contract to fill up the display surface, as needed.
- Most systems provide two-dimensional tiled windows, adjustable in both height and width.





#### Advantages:

- The system usually allocates and positions windows for the user, eliminating the necessity to make positioning decisions.
- Open windows are always visible, eliminating the possibility of them being lost and forgotten.
- Every window is always completely visible, eliminating the possibility of information being hidden.
- They are perceived as fewer complexes than overlapping windows, possibly because there are fewer management operations or they seem less “magical.”
- They are easier, according to studies, for novice or inexperienced people to learn and use.
- They yield better user performance for tasks where the data requires little window manipulation to complete the task.

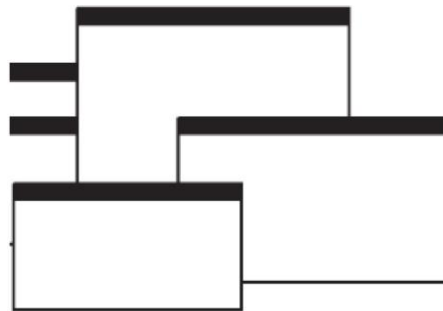
#### Disadvantages

- Only a limited number can be displayed in the screen area available.
- As windows are opened or closed, existing windows change in size. This can be annoying.
- As windows change in size or position, the movement can be disconcerting.

- As the number of displayed windows increases, each window can get very tiny.
- The changes in sizes and locations made by the system are difficult to predict.
- The configuration of windows provided by the system may not meet the user's needs.
- They are perceived as crowded and more visually complex because window borders are flush against one another, and they fill up the whole screen. Crowding is accentuated if borders contain scroll bars or control icons. Viewer attention may be drawn to the border, not the data.
- They permit less user control because the system actively manages the windows.

## Overlapping Windows

- Overlapping windows may be placed on top of one another like papers on a desk.
- They possess a three-dimensional quality, appearing to lie on different planes.



### Advantages:

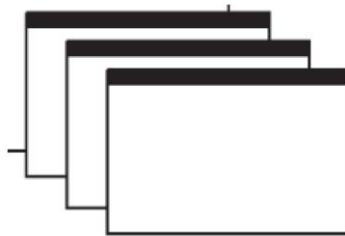
- Visually, their look is three-dimensional, resembling the desktop that is familiar to the user.
- Greater control allows the user to organize the windows to meet his or her needs.
- Windows can maintain larger sizes.
- Windows can maintain consistent sizes.
- Windows can maintain consistent positions.
- Screen space conservation is not a problem, because windows can be placed on top of one another.
- There is less pressure to close or delete windows no longer needed.
- The possibility exists for less visual crowding and complexity. Larger borders can be maintained around window information, and the window is more clearly set off against its background. Windows can also be expanded to fill the entire display.
- They yield better user performance for tasks where the data requires much window manipulation to complete the task.

## Disadvantages

- They are operationally much more complex than tiled windows. More control functions require greater user attention and manipulation.
- Information in windows can be obscured behind other windows.
- Windows themselves can be lost behind other windows and be presumed not to exist.
- That overlapping windows represent a three-dimensional space is not always realized by the user.
- Control freedom increases the possibility for greater visual complexity and crowding. Too many windows, or improper offsetting, can be visually overwhelming.

## Cascading Windows

- A special type of overlapping window has the windows automatically arranged in a regular progression.
- Each window is slightly offset from others, as illustrated in Figure



## Advantages

- No window is ever completely hidden.
- Bringing any window to the front is easier.
- It provides simplicity in visual presentation and cleanliness.

## Picking a Presentation Style

- Use tiled windows for:
  - Single-task activities.
  - Data that needs to be seen simultaneously.
  - Tasks requiring little window manipulation.
  - Novice or inexperienced users.
- Use overlapping windows for:
  - Switching between tasks.
  - Tasks necessitating a greater amount of window manipulation.
  - Expert or experienced users.
  - Unpredictable display contents.

5 Explain the purpose of prototypes. Discuss any two kinds of prototypes with their importance to the system developers.

- A prototype is primarily a vehicle for exploration, communication, and evaluation. Its purpose is to obtain user input in design, and to provide feedback to designers.

- A prototype is a simulation of an actual system that can be quickly created.
  - ✓ A prototype may be a rough approximation, such as a simple hand-drawn sketch, or it may be interactive, allowing the user to key or select data using controls, navigate through menus, retrieve displays of data, and perform basic system functions.
  - ✓ A prototype may have great breadth, including as many features as possible to present concepts and overall organization, or it might have more depth, including more detail on a given feature or task to focus on individual design aspects.

## Hand Sketches and Scenarios

- ✓ Description:
  - Screen sketches created by hand.
  - Focus is on the design, not the interface mechanics.
  - A low-fidelity prototype.
- ✓ Advantages:
  - Can be used very early in the development process.
  - Suited for use by entire design team.
  - No large investment of time and cost.
  - No programming skill needed.
  - Easily portable.
  - Fast to modify and iterate.
  - A rough approximation often yields more substantive critical comments.
  - Easier to comprehend than functional specifications.
  - Can be used to define requirements.
- ✓ Disadvantages:
  - Only a rough approximation.
  - Limited in providing an understanding of navigation and flow.
  - A demonstration, not an exercise.
  - Driven by a facilitator, not the user.
  - Limited usefulness for a usability test.
  - A poor detailed specification for writing the code.
  - Usually restricted to most common tasks.
- ✓ **Sketch Creation Process**
  - Sketch (storyboard) the screens while determining:
    - The source of the screen's information.
    - The content and structure of individual screens.
    - The overall order of screens and windows.
  - Use an erasable medium.
  - Sketch the screens needed to complete each workflow task.
  - Try out selected metaphors and change them as necessary.
  - First, storyboard common/critical/frequent scenarios.
    - Follow them from beginning to end.
    - Then, go back and build in exceptions.
  - Don't get too detailed; exact control positioning is not important, just overall order and flow.
  - Storyboard as a team, including at least one user.

- Only develop online prototypes when everyone agrees that a complete set of screens has been satisfactorily sketched.

### **Interactive Paper Prototypes**

- ✓ Description:
  - Interface components (menus, windows, and screens) constructed of common paper technologies (Post-It notes, transparencies, and so on).
  - The components are manually manipulated to reflect the dynamics of the software.
  - A low-fidelity prototype.
- ✓ Advantages:
  - More illustrative of program dynamics than sketches.
  - Can be used to demonstrate the interaction.
  - Otherwise, generally the same as for hand-drawn sketches and scenarios.
- ✓ Disadvantages:
  - Only a rough approximation.
  - A demonstration, not an exercise.
  - Driven by a facilitator, not the user.
  - Limited usefulness for usability testing.

### **Programmed Facades**

- Description:
  - Examples of finished dialogs and screens for some important aspects of the system.
  - Created by prototyping tools.
  - Medium-fidelity to high-fidelity prototypes.
- Advantages:
  - Provide a good detailed specification for writing code.
  - A vehicle for data collection.
- Disadvantages:
  - May solidify the design too soon.
  - May create the false expectation that the “real thing” is only a short time away.
  - More expensive to develop.
  - More time-consuming to create.
  - Not effective for requirements gathering.
  - Not all of the functions demonstrated may be used because of cost, schedule limitations, or lack of user interest.
  - Not practical for investigating more than two or three approaches.

### **Prototype-Oriented Languages**

- ✓ Description:
  - An example of finished dialogs and screens for some important aspects of the system.

- Created through programming languages that support the actual programming process.
- A high-fidelity prototype.
- Advantages:
  - May include the final code.
  - Otherwise, generally the same as those of programmed facades.
- Disadvantages:
  - Generally the same as for programmed facades.

6 Explain Cognitive Walkthroughs, think Aloud evaluation and usability tests conducted in UID.

### Cognitive Walkthroughs

#### Description:

— Reviews of the interface in the context of tasks users perform.

#### Advantages:

- Allow a clear evaluation of the task flow early in the design process.
- Do not require a functioning prototype.
- Low cost.
- Can be used to evaluate alternate solutions.
- Can be performed by developers.
- More structured than a heuristic evaluation.
- Useful for assessing “exploratory learning.”

#### Disadvantages:

- Tedious to perform.
- May miss inconsistencies and general and recurring problems.

#### Guidelines:

— Needed to conduct the walkthrough are:

- A general description of proposed system users and what relevant knowledge they possess.
- A specific description of one or more core or representative tasks to be performed.
- A list of the correct actions required to complete each of the tasks.

— Review:

- Several core or representative tasks across a range of functions.
  - Proposed tasks of particular concern.
- Developers must be assigned roles of:
- Scribe to record results of the action.
  - Facilitator to keep the evaluation moving.

— Start with simple tasks.

— Don't get bogged down demanding solutions.

— Limit session to 60 to 90 minutes.

### Think-Aloud Evaluations

#### Description:

— Users perform specific tasks while thinking out loud.

— Comments are recorded and analyzed.

#### Advantages:

— Utilizes actual representative tasks.

— Provides insights into the user's reasoning.

Disadvantages:

— May be difficult to get users to think out loud.

Guidelines:

— Develop:

- Several core or representative tasks.

- Tasks of particular concern.

— Limit session to 60 to 90 minutes.

Usability Test

Description:

— An interface evaluation under real-world or controlled conditions.

— Measures of performance are derived for specific tasks.

— Problems are identified.

Advantages:

— Utilizes an actual work environment.

— Identifies serious or recurring problems.

Disadvantages:

— High cost for establishing facility.

— Requires a test conductor with user interface expertise.

— Emphasizes first-time system usage.

— Poorly suited for detecting inconsistency problems.