Screen Design Standards and Detailed UI Design

Purpose

- The purpose of the screen design standards is to ensure consistency and simplicity in detailed UI design across all the displays
- Consistency contributes ease of learning, ease of use and remembering.
- Screen design standards also ensure quality when the standards are based on the User Profiles, contextual task analysis, usability goals setting.
- Standards can reduce time and cost(development of reusable code)

Description

- Define and document set of Screen Design Standards to be followed by all displays and interactions
- Some standards can be adopted directly or adapted from platform style guide(e.g Microsoft windows)
- Validated screen design standards are documented in product style guide

Issues that should be standardized

- Use of controls(e.g check boxes, option buttons, list boxes, combo boxes, push buttons)
- Location and format of standard display components(e.g title bar, status line, display body controls and navigation controls, action controls)
- Terminology
- Use of colors, fonts, styles
- Type, location, Format and wording of messages.

Roles and Resources

- Task leader: UI Designer
- Other resources: All team members previously participated.

A Step By Step Procedure

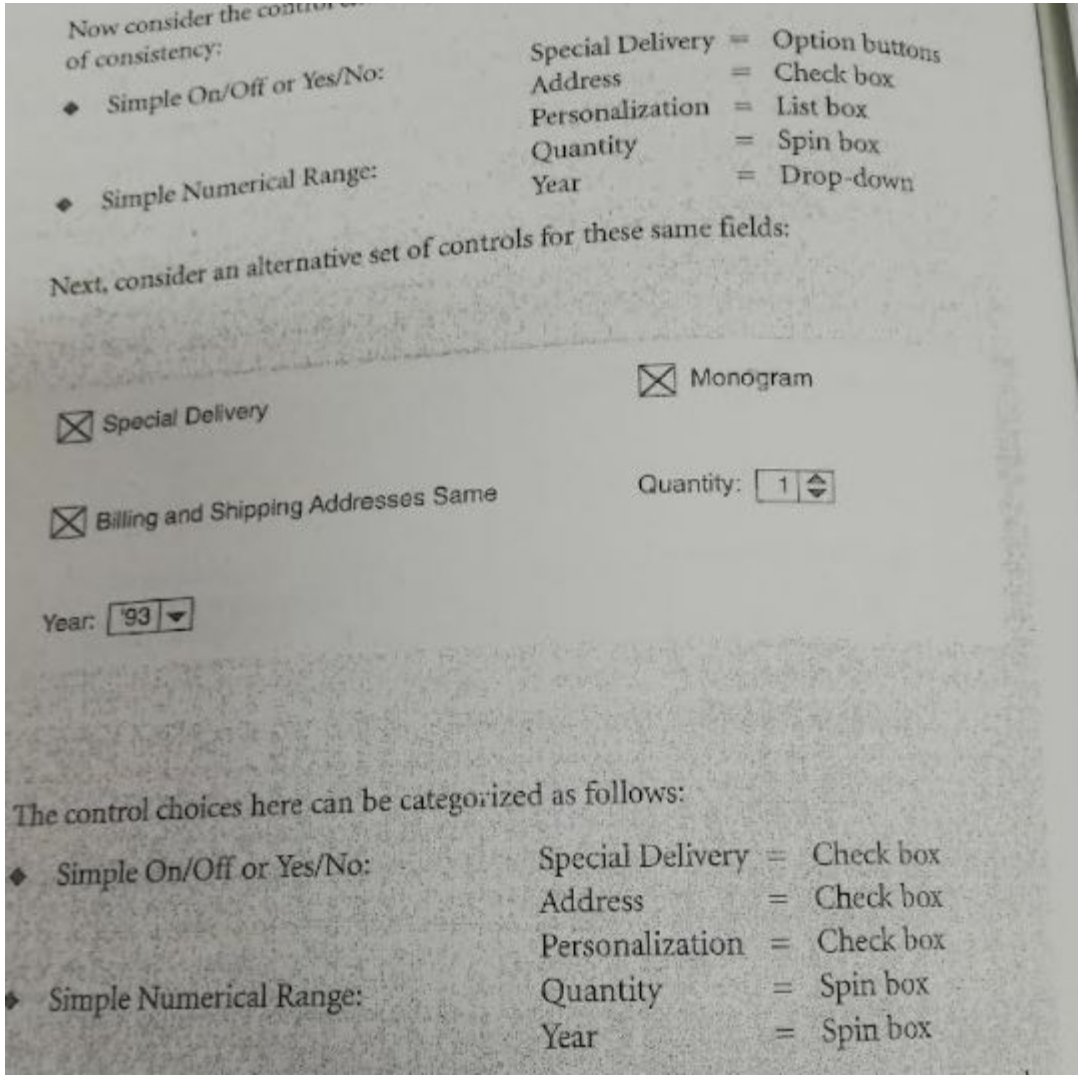
Screen Design Standards based on today's GUI platform standards can be grouped into the following categories:

- Control standards
- Product/process window standards
- Dialog box contents standards
- Message box contents standards
- Input device interactions standards
- Feedback standards

A Step By Step Procedure

- Draft Control standards

1.Draft Control standards



Now consider the control
of consistency:

- Simple On/Off or Yes/No:
- Simple Numerical Range:

Special Delivery	=	Option buttons
Address	=	Check box
Personalization	=	List box
Quantity	=	Spin box
Year	=	Drop-down

Next, consider an alternative set of controls for these same fields:

☒ Special Delivery

☒ Monogram

☒ Billing and Shipping Addresses Same

Quantity: 1

Year: '93

The control choices here can be categorized as follows:

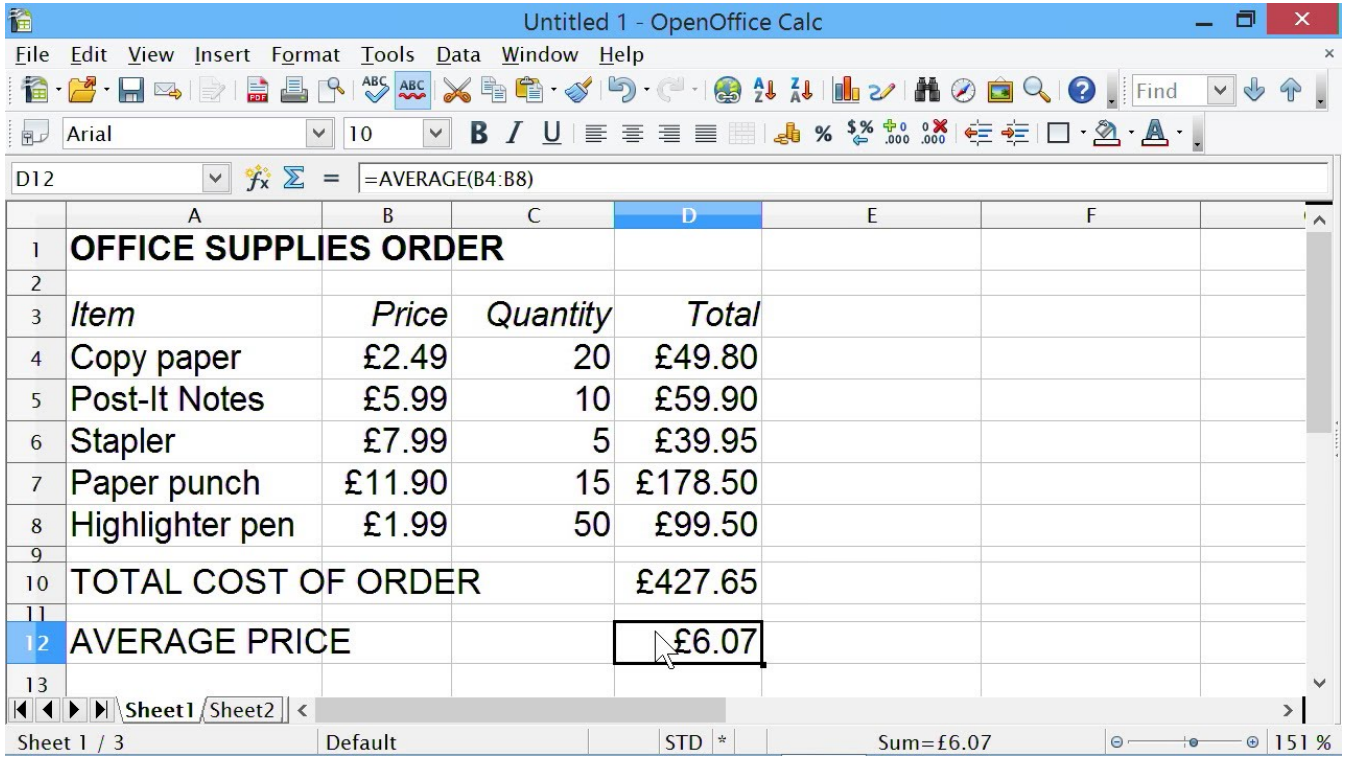
- Simple On/Off or Yes/No:
- Simple Numerical Range:

Special Delivery	=	Check box
Address	=	Check box
Personalization	=	Check box
Quantity	=	Spin box
Year	=	Spin box

2.Draft product/process window standards

- Product/process oriented
- Want to design a set of standards
- E.g how empty untitled worksheet will look as well as how created and saved worksheets will look, set of fields arranged in a particular order

Draft product/process window standards



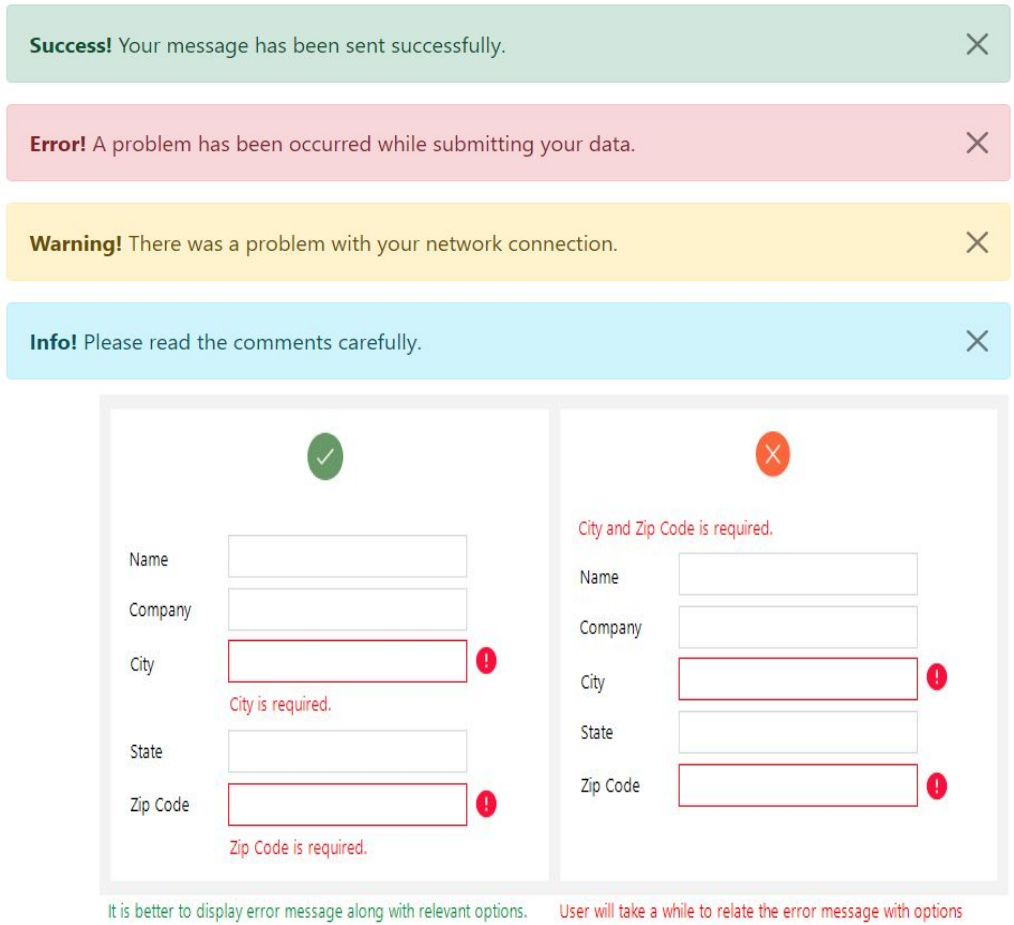
3.Draft Dialog box standards

- Set standards for dialog boxes.
- Required vs optional fields
- read only vs editable fields
- locating and titling push buttons

4.Draft message box standards

- Type of message boxes
  - Error
  - Warning
  - Status message
- Wording and syntax of message, placement of push buttons

Draft message box standards



5.Draft feedback standards

- How you will provide consistent feedbacks

Draft feedback standards. These standards define how you will consistently provide certain types of feedback. Depending on your product functionality, you may want to provide feedback on such things as selection, task completion, active process or product, and status. You can link visual cues (e.g., color, highlighting, blinking, shape, size) to types of feedback in a set of feedback standards. For example, you might use a small set of colors to indicate the status of something, or highlighting to indicate selection, or cursor shape to indicate task in progress versus task completion.

- 6.Document all Draft Standards

All screen design standards should be written down

Web Notes

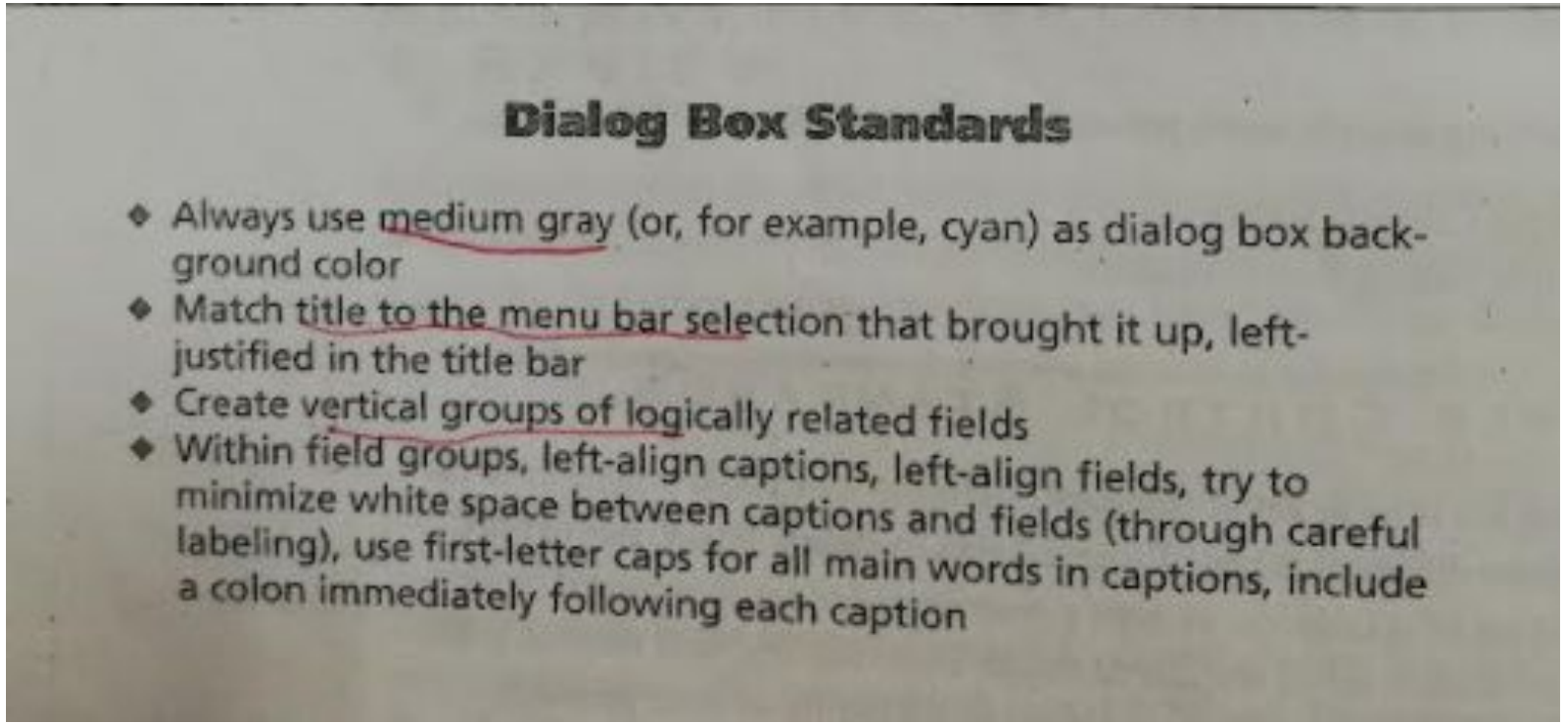
Web design techniques (both good and bad) tend to be copied. Perhaps other Web designers will copy your Screen Design Standards! And perhaps someday, we will have a set of universal Web Screen Design Standards supported by Web development tools, not unlike Microsoft Windows and Apple Macintosh standards. This would contribute greatly to the usability of the Web, just as the latter standards have done for traditional software.

Sample Work Product

Control Standards

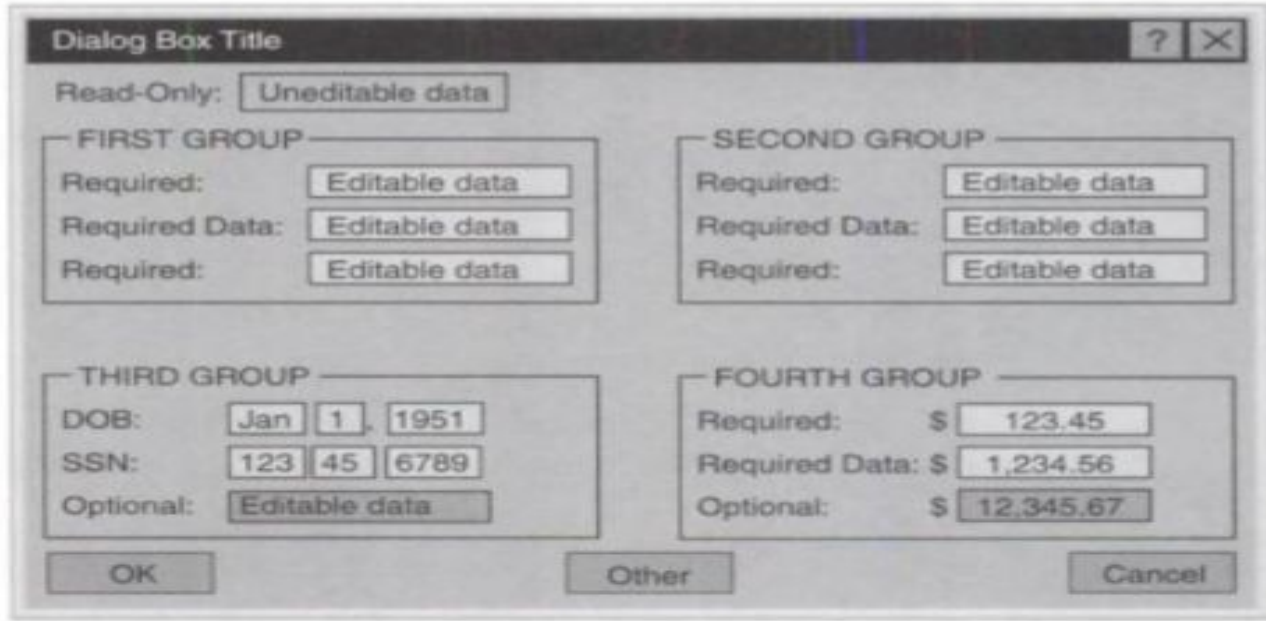
Menu Contents	Control
Navigational actions	Push buttons
Small number/choose one/fixed list	Option buttons
Small number/choose many/fixed list or yes/no, on/off	Check boxes
Large number or variable list/scarce screen space or to reduce clutter	Drop-down list box
Large number or variable list or choose many	Standard list box
Variable input	Text box
Very long lists/scarce screen space	Drop-down combo box
Very long lists	Standard combo box
Numbers or sequential lists	Spin button

Sample Work Product



Dialog Box Standards

- Always use medium gray (or, for example, cyan) as dialog box background color
- Match title to the menu bar selection that brought it up, left-justified in the title bar
- Create vertical groups of logically related fields
- Within field groups, left-align captions, left-align fields, try to minimize white space between captions and fields (through careful labeling), use first-letter caps for all main words in captions, include a colon immediately following each caption



- Use group boxes, embed all-caps titles in upper left of group box (do not use all caps anywhere else)
- Order groups left to right, top to bottom according to natural order or expected frequency of use
- Use white space to set off and reinforce groups
- Right-justify or decimal-align numbers and currency *upon display* (allow left-justified input)
- Always place OK push button at lower-left edge, Cancel push button at lower-right edge, any other push buttons evenly spaced in between (push buttons are dark gray)
- Never use scroll bars in dialog boxes
- Background colors for fields:
  - Read-only—medium gray (or, for example, cyan)
  - Required—white (or, for example, yellow)
  - Optional—dark gray (or, for example, gray)
- Use consistent labels for common fields
  - SSN (Social Security Number)
  - DOB (Date of Birth)
- Use consistent display format for all dates: Jan 1, 2000 (or Jan 1)

Detailed UI Design

Purpose

PURPOSE

The design of the product user interface in all its complete detail is, of course, the ultimate purpose of the whole Usability Engineering Lifecycle; all the other tasks across the lifecycle are aimed at accomplishing this task as efficiently and effectively as possible. We want more than just a complete user interface design (which will happen with or without the Usability Engineering Lifecycle); we want a complete user interface design that optimizes user performance and satisfaction, and that is created through a cost-effective development process.

Step by Step Procedure

1. Complete the identification of all pathways between windows, dialog boxes and message boxes
2. Complete the design of menu bar and all other action controls
3. Complete the design of content of all windows, dialog boxes and message boxes
4. Complete the design of all interactions with input devices(mouse interactions, keyboard shortcuts/accelerators,

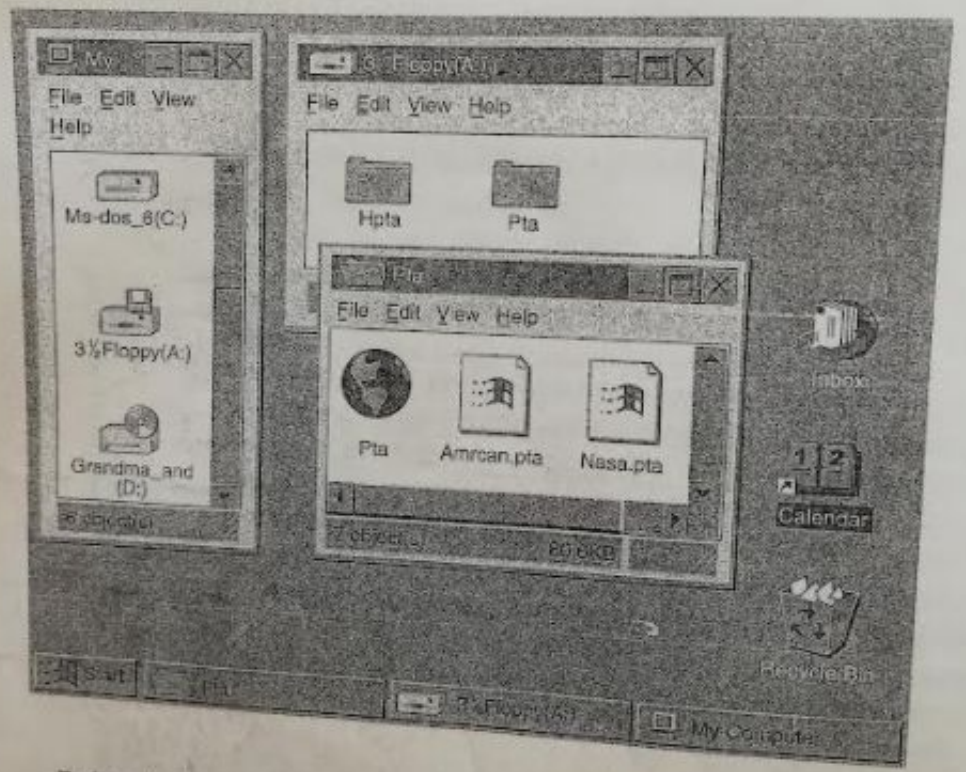
Sample Work product

Personal Travel Agent Application—
Detailed UI Design Specification

Following the Microsoft Windows standard, users launch the application from the desktop or operating system level. Users can store the application

"Detailed UI Design Specification," cont. next page

itself, as well as the primary products created with it (in this case, Trips), in any way they wish in their folder structure. In the following illustration, the user has stored the application and all current primary products (Trips) in the same folder (called "Pta," for Personal Travel Agent).

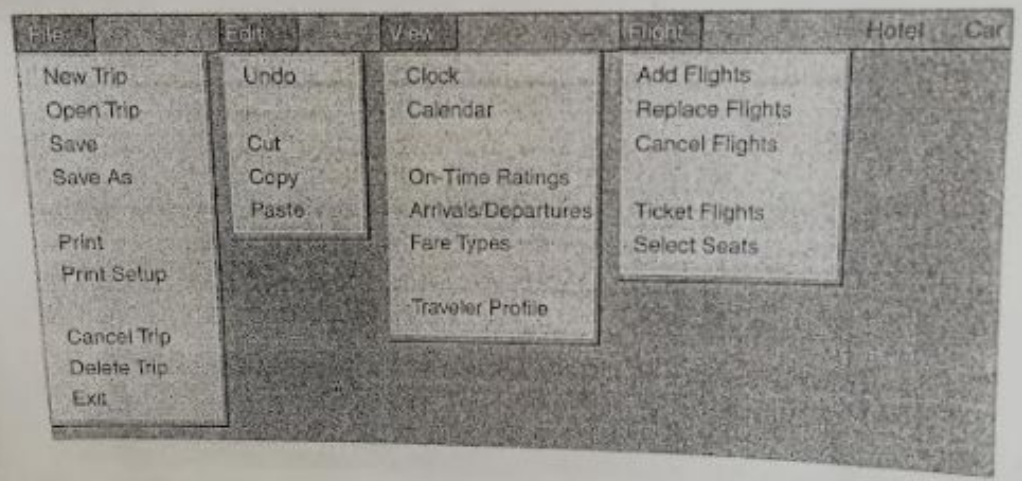


To launch the application, any standard Windows method may be used, including double-clicking with the mouse on the application icon or on any primary product (Trip) icon on the desktop. This opens the application window, with a primary product (Trip) window within it containing either an empty untitled trip or the saved trip that was double-clicked, respectively.

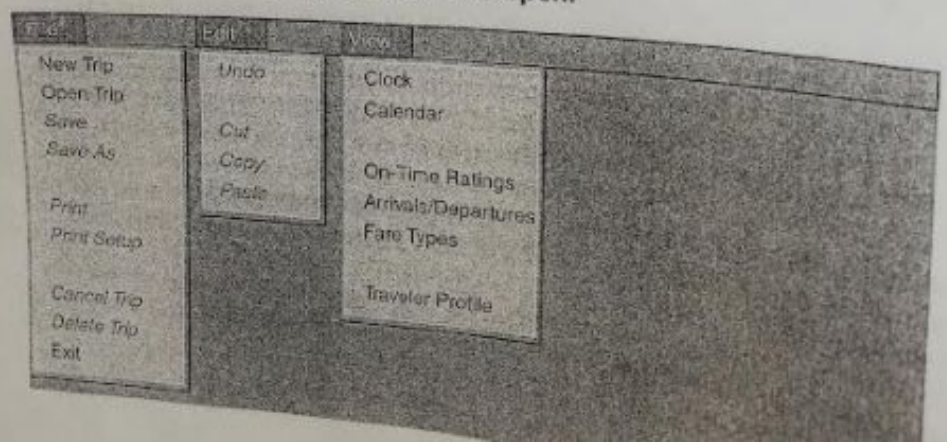
(Here, the specification would show the above mentioned configurations of windows.)

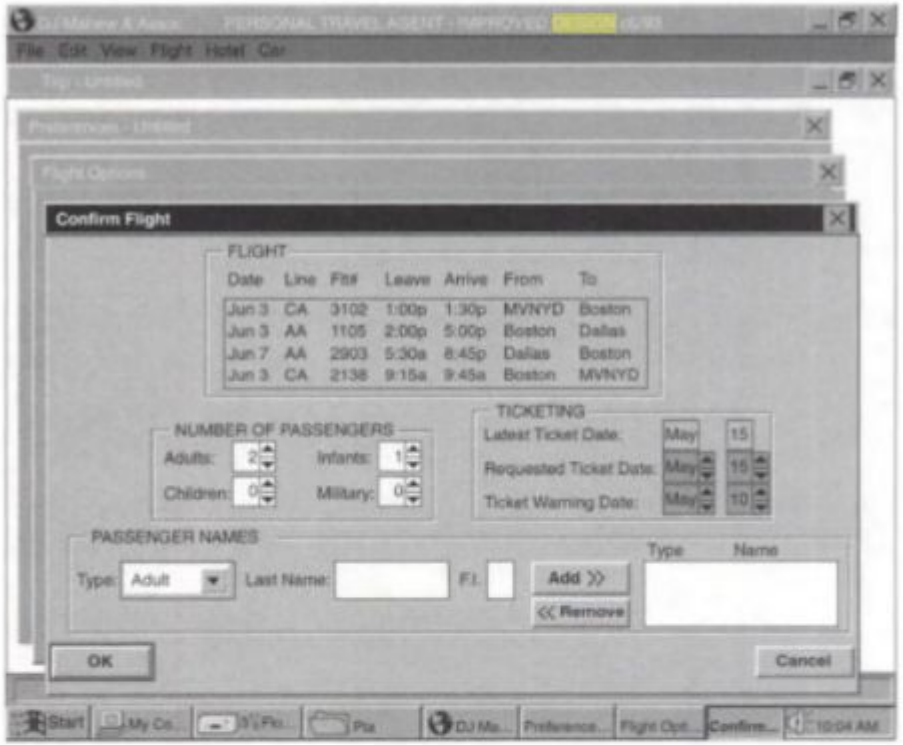
Only the application window contains a menu bar, and it is designed as follows. (The menu bar structure shown in this sample is not complete but would be completely specified in your Detailed User Interface Design.)

When a document (Trip) window is open:



When no document (Trip) windows are open:





(Next the specification would describe all pathways through the application, showing the results of all possible picks within the menu bar and all displays, including the complete contents of all menu controls within each display. At some point in this complete and exhaustive specification, the illustration might look like the next illustration, showing a particular pathway initiated through the menu bar that takes the user through a set of cascading dialog boxes. Each dialog box would be illustrated in complete detail, with descriptions of how to interact with it, as in the following text).

The user fills in at least the required (light gray) fields and perhaps the optional (dark gray) ones. The controls for these fields operate in the standard Windows way.

The Passenger Type Name list box is filled in by successively filling in the Type, Last Name, and F. I. (First Initial) controls and then invoking (by any of the standard Windows methods) the Add push button. Passengers can be removed by selecting them in the Type Name list box and then invoking the Remove push button.

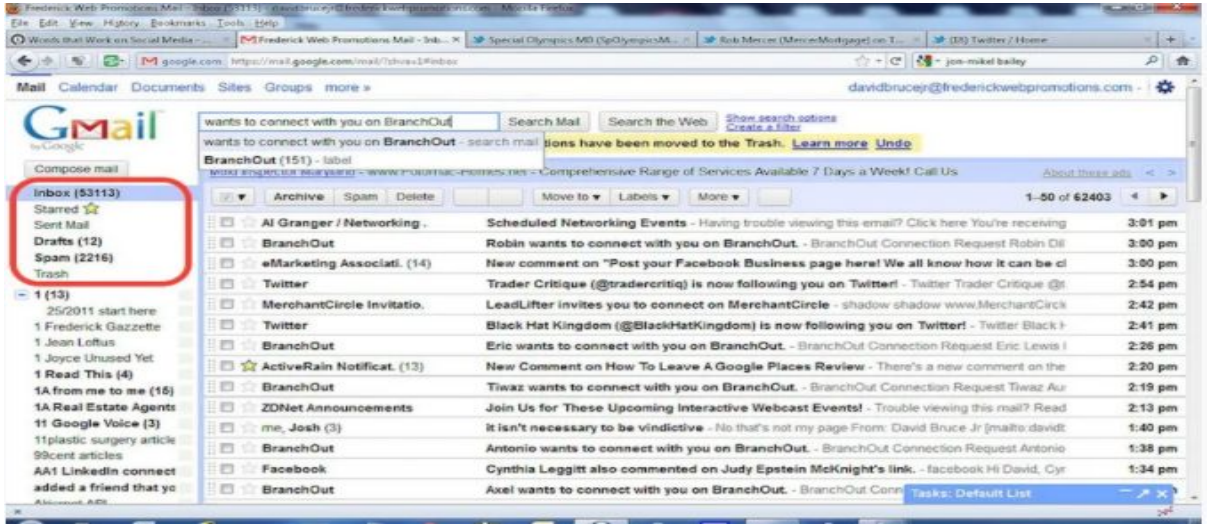
When all required fields are filled in, the user can click the OK button. This returns the user to the document (Trip) window, with all specified data from the dialog box sequence now filled in. If the user clicks the OK push button before all required fields are entered, a pop-up message box appears. (Here the specification would show this particular message box in an illustration and describe how to interact with it.)

At any time, the user can click the Cancel push button and return to the previous dialog box in the sequence. The menu bar is not active within the dialog box—only the push buttons are active. They are the only means of navigation out of the dialog box. The dialog box is also "modal," meaning the user cannot give focus to any other window simply by clicking on it. Again, only the push buttons within the dialog box provide navigation out of it.

Five Ways You Can Achieve Consistency in Your Work

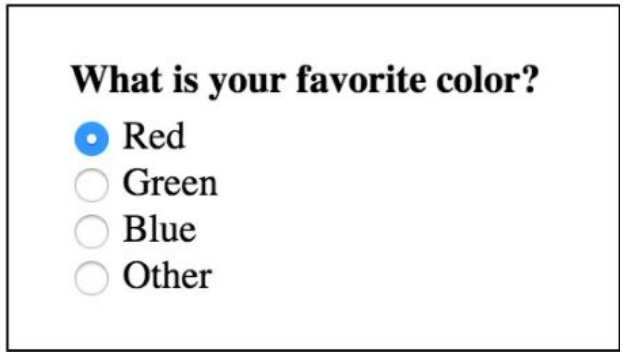
Your Choice of Language

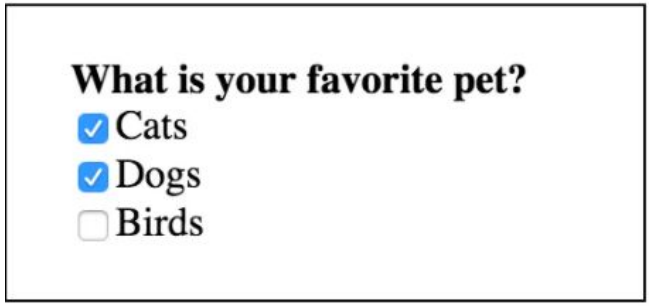
1. When things mean the same or perform the same operation they should be represented in the same way, as is the case in Google's email facility, Gmail. Based on the organization style of client email applications, Gmail’s folders are labelled 'Inbox', 'Drafts', 'Sent Mail', etc. The language used for these folders shows familiarity and consistency to anyone who have used email applications in the past



Apply UI Elements as They are Originally Defined

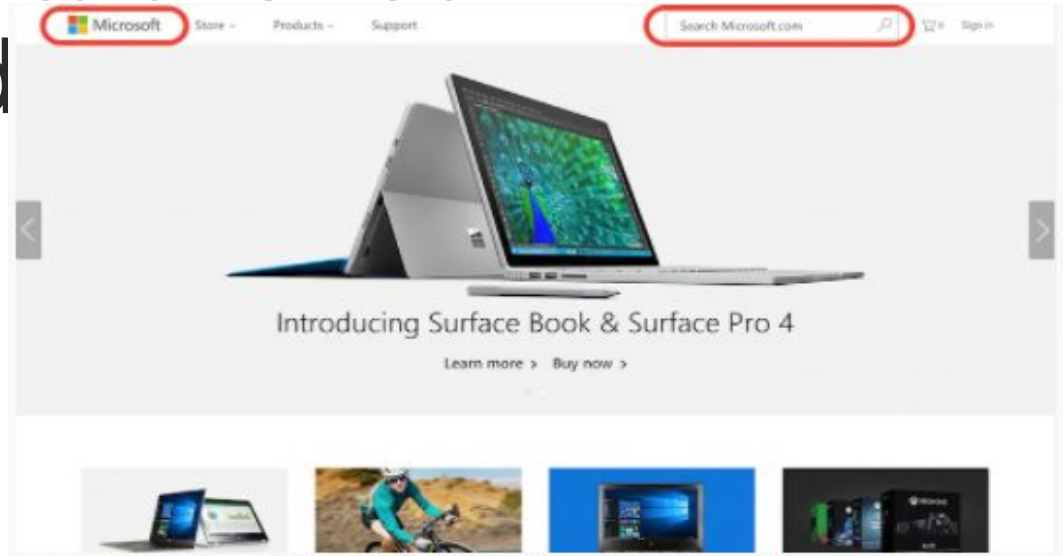
- For instance, radio buttons are meant to be used when there is only one option allowed. Checkboxes on the other hand should be used only when the user is allowed more than one option.





Consider Various Well-established Conventions When Decid

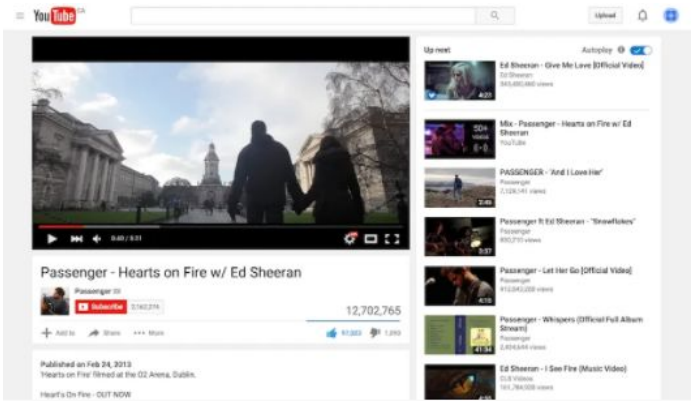
- Humans have a strong memory
- for where things are visually
- located on the screen



Author/Copyright holder: Microsoft Corporation. Copyright terms and licence: Fair Use.

Design for your User's Expectations

- For example, an airline site should have a ticket-booking system, while a music-sharing site should have a media player
- A video-sharing website like YouTube is obviously expected to have a video player. This is a great example of consistency in that the features and functionality of the site supports what the user expects.



Create Consistent Visual Elements throughout Your Site

- A bad example of consistency in the choice of colors and fonts can be seen in the
- 2013 version of Google's Gmail mobile user interface.