[Main content](https://ng.cengage.com/static/nbreader/ui/apps/nbreader/nbreader.html" \l "header)

**Chapter**

**8**

**User Interface Design**

* **8.1**[User Interfaces](javascript://)
* **8.2**[Human-Computer Interaction](javascript://)
* **8.3**[Seven Habits of Successful Interface Designers](javascript://)
  + **8.3.1**[Understand the Business](javascript://)
  + **8.3.2**[Maximize Graphical Effectiveness](javascript://)
  + **8.3.3**[Think like a User](javascript://)
  + **8.3.4**[Use Models and Prototypes](javascript://)
  + **8.3.5**[Focus on Usability](javascript://)
  + **8.3.6**[Invite Feedback](javascript://)
  + **8.3.7**[Document Everything](javascript://)
* **8.4**[Guidelines for User Interface Design](javascript://)
  + **8.4.1**[Create an Interface That Is Easy to Learn and Use](javascript://)
  + **8.4.2**[Enhance User Productivity](javascript://)
  + **8.4.3**[Provide Flexibility](javascript://)
  + **8.4.4**[Provide Users with Help and Feedback](javascript://)
  + **8.4.5**[Create an Attractive Layout and Design](javascript://)
  + **8.4.6**[Enhance the Interface](javascript://)
  + **8.4.7**[Focus on Data Entry Screens](javascript://)
  + **8.4.8**[Use Validation Rules](javascript://)
  + **8.4.9**[Manage Data Effectively](javascript://)
  + **8.4.10**[Reduce Input Volume](javascript://)
* **8.5**[Source Document and Form Design](javascript://)
* **8.6**[Printed Output](javascript://)
  + **8.6.1**[Report Design](javascript://)
  + **8.6.2**[Report Design Principles](javascript://)
  + **8.6.3**[Types of Reports](javascript://)
* **8.7**[Technology Issues](javascript://)
  + **8.7.1**[Output Technology](javascript://)
  + **8.7.2**[Input Technology](javascript://)
* **8.8**[Security and Control Issues](javascript://)
  + **8.8.1**[Output Security and Control](javascript://)
  + **8.8.2**[Input Security and Control](javascript://)
* **8.9**[Emerging Trends](javascript://)
  + **8.9.1**[Modular Design](javascript://)
  + **8.9.2**[Responsive Web Design](javascript://)
  + **8.9.3**[Prototyping](javascript://)
* **8.10**[Summary](javascript://)
* [Chapter Review](javascript://)
  + [Key Terms](javascript://)

Change font size

[**help**](javascript://)

[Main content](https://ng.cengage.com/static/nbreader/ui/apps/nbreader/nbreader.html#header)

**8.1**User Interfaces

A [**user interface (UI)**](javascript://) describes how users interact with a computer system and consists of all the hardware, software, screens, menus, functions, output, and features that affect two-way communications between the user and the computer. The user interface is the key to [**usability**](javascript://), which includes user satisfaction, support for business functions, and system effectiveness.

Traditionally, a chapter on user interface design started with a discussion of output because output is what users touched, viewed, and needed to do their jobs. Today, the situation is different, for several important reasons:

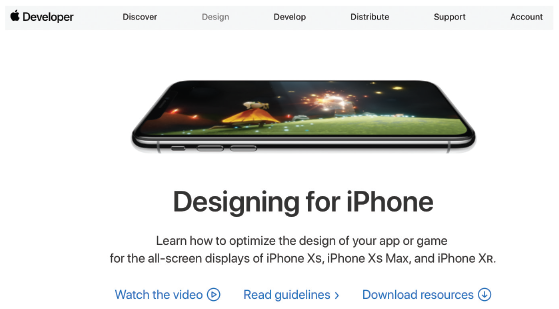
* Users can design their own output. System designers are more aware of user needs and desires. A system can maintain data integrity and still allow users to view, sort, filter, and examine data in any way that helps them do their jobs. There was a time when the MIS department made those choices and users had little or no say in the matter. Today, successful applications are designed quite differently—the system developer identifies user needs and then creates a design that will satisfy users *and* meet corporate requirements.
* Centralized IT departments no longer produce reams of printed reports. Those reports often gathered dust while sitting on top of file cabinets. While a few examples might persist, the overwhelming trend has been to customer-designed output. The customer might be an individual user, or a community of users, such as a department. As [Chapter 4](javascript://) pointed out, the IT team must understand user requirements before creating a solution.
* The user interface itself has evolved into a two-way channel, with powerful output capability, and most user information needs can be met with screen-generated data, which a user can print, view, or save. Well into the 1980s and beyond, a user interface was a blank character-based screen, which might or might not offer menu choices. If a user entered a command improperly, the system responded with an error message, which frustrated users and stifled productivity. Many hardware-centric vendors did not understand the importance of the user interface and its implications.

Apple was a pioneer in user interface development, introducing the [**graphical user interface (GUI)**](javascript://), complete with mouse and screen icons, in the early 1980s. At that point, not many companies were ready for this concept. When software giant Microsoft finally jumped on the GUI bandwagon with its Windows® operating system, the corporate doors swung open, and everyone from managers on down said, “How did we ever do without this?”

Many industry leaders believe that the best interfaces are the ones that users do not even notice—they make sense because they do what users expect them to do. For example, as shown in [Figure 8-1](javascript://), Apple believes that designing an exceptional user interface is essential to a successful app. Apple has long distinguished itself from its competitors by the intuitiveness of its products. Apple’s command of the market suggests that consumers are willing to pay a premium for products that “just work.”

**Figure 8-1**

Apple has long been a leader in creating elegant user interfaces for its products.



**Source:** Apple Inc.

When developing older systems, analysts typically designed all the printed and screen output first and then worked on the inputs necessary to produce the results. Often, the user interface mainly consisted of process-control screens that allowed the user to send commands to the system. That approach worked well with traditional systems that simply transformed input data into structured output.

As information management evolved from centralized data processing to dynamic, enterprise-wide systems, the primary focus also shifted—from the IT department to the users themselves. The IT group became a supplier of information technology, rather than a supplier of information. Today, the main focus is on users within and outside the company, how they communicate with the information system, and how the system supports the firm’s business operations.

In a [**user-centered**](javascript://) system, the distinction blurs between input, output, and the interface itself. Most users work with a varied mix of input, screen output, and data queries as they perform their day-to-day job functions. Because all those tasks require interaction with the computer system, the user interface is a vital element in the systems design phase.

User interface design requires an understanding of human-computer interaction and user-centered design principles, which are discussed in the next section.

Change font size

[**help**](javascript://)

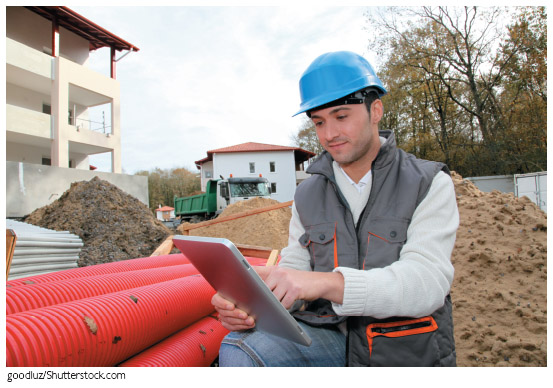
[Main content](https://ng.cengage.com/static/nbreader/ui/apps/nbreader/nbreader.html#header)

# 8.2Human-Computer Interaction

A user interface is based on basic principles of human-computer interaction. [**Human-computer interaction (HCI)**](javascript://) describes the relationship between computers and the people who use them to perform their jobs, like the worker shown in [Figure 8-2](javascript://). HCI concepts apply to everything from smartphones to global networks. In its broadest sense, HCI includes all the communications and instructions necessary to enter input to the system and to obtain output in the form of screen displays or printed reports.

**Figure 8-2**

HCI is essential to employee productivity, whether the work is done in a traditional office setting or on a construction site like the one shown in this figure.



Goodluz/ [Shutterstock.com](http://shutterstock.com/" \t "_blank)

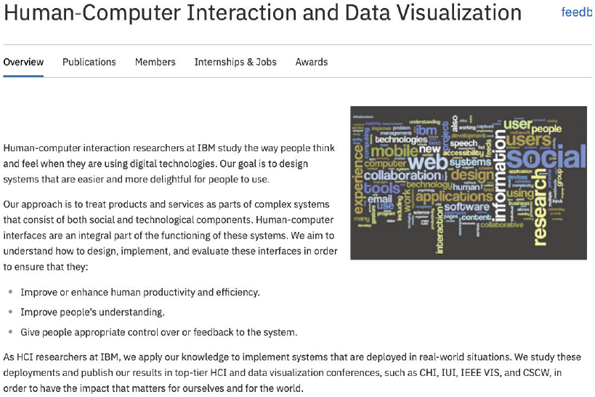
Early user interfaces involved users typing complex commands on a keyboard, which displayed as green text on a black screen. Then came the GUI, which was a huge improvement because it used icons, graphical objects, and pointing devices. Today, designers strive to translate user behavior, needs, and desires into an interface that users don’t really notice. IBM has stated that the best user interfaces are “almost transparent—you can see right through the interface to your own work.” In other words, a [**transparent interface**](javascript://) does not distract the user and calls no attention to itself.

A systems analyst designs user interfaces for in-house-developed software and customizes interfaces for various commercial packages and user productivity applications. The main objective is to create a user-friendly design that is easy to learn and use.

Industry leaders Microsoft and IBM both devote considerable resources to user interface research. [Figure 8-3](javascript://) describes IBM Research’s work on HCI. Their stated goal is to “design systems that are easier and more delightful for people to use.”

**Figure 8-3**

IBM’s research division is a leader in exploring human-computer interaction (HCI).



Enlarge Image

**Source:** IBM Corporation

Because HCI has a major impact on user productivity, it gets lots of attention—especially where multimillion-dollar issues are concerned. For example, in the article “Human-Computer Interaction in Electronic Medical Records: From the Perspectives of Physicians and Data Scientists” by E. Bologva et al. from Procedia Computer Science 100 (Elsevier, 2016), the authors describe how software usability has a major impact on the medical profession, and not everyone is happy about it—particularly physicians who often struggle with [**electronic health records (EHRs)**](javascript://) systems that are poorly designed. In her article, Ms. Gardner points out that physicians often multitask, answering a question about one patient while writing a prescription for another, and EHR software was not designed around that type of workflow.

**Case in Point 8.1**

### Casual Observer Software

Casual Observer Software’s main product is a program that monitors and analyzes user keystrokes and mouse clicks to learn more about the way employees use their computer systems. The problem is that some users feel this is an unwarranted intrusion into their privacy, and they prefer not to be observed. Some even fear that the data would be used for other reasons, including performance appraisal. You are a consultant who has been hired by a client firm that is trying to decide whether or not to use this software.

Before you advise the client, remember the Hawthorne effect, which suggests that employees might behave differently when they know they are being observed. Finally, think about the ethical issues that might be involved in this situation. What will you advise your client, and why?

Change font size

[**help**](javascript://)

[Main content](https://ng.cengage.com/static/nbreader/ui/apps/nbreader/nbreader.html#header)

**8.3**Seven Habits of Successful Interface Designers

Although IT professionals have different views about interface design, most would agree that good design depends on seven basic principles. Successful interface designers use these basic principles as a matter of course—they become habits. These desirable habits are described in the following sections.

Change font size

[**help**](javascript://)

[Main content](https://ng.cengage.com/static/nbreader/ui/apps/nbreader/nbreader.html#header)

## 8.3.1Understand the Business

The interface designer must understand the underlying business functions and how the system supports individual, departmental, and enterprise goals. The overall objective is to design an interface that helps users to perform their jobs. A good starting point might be to analyze a functional decomposition diagram (FDD). As described in [Chapter 4](javascript://), an FDD is a graphical representation of business functions that starts with major functions and then breaks them down into several levels of detail. An FDD can provide a checklist of user tasks that must be included in the interface design.

Change font size

[**help**](javascript://)

[Main content](https://ng.cengage.com/static/nbreader/ui/apps/nbreader/nbreader.html#header)

## 8.3.2Maximize Graphical Effectiveness

Studies show that people learn better visually. The immense popularity of Apple’s iOS and Microsoft Windows is largely the result of their GUIs that are easy to learn and use. A well-designed interface can help users learn a new system rapidly and be more productive. Also, in a graphical environment, a user can display and work with multiple windows on a single screen and transfer data between programs. If the interface supports data entry, it must follow the guidelines for data entry screen design that are discussed later in this chapter.

Change font size

[**help**](javascript://)

[Main content](https://ng.cengage.com/static/nbreader/ui/apps/nbreader/nbreader.html#header)

## 8.3.3Think like a User

A systems analyst should understand user experience, knowledge, and skill levels. If a wide range of capability exists, the interface should be flexible enough to accommodate novices as well as experienced users.

To develop a user-centered interface, the designer must learn to think like a user and see the system through a user’s eyes. The user interface must be easy to learn, using terms and metaphors that are familiar to users. Users are likely to have real-world experience with many other machines and devices that provide feedback, such as automobiles, ATMs, and microwave ovens. Based on that experience, users will expect useful, understandable feedback from a computer system.

Carefully examine any point where users provide input or receive output. Input processes should be easy to follow, intuitive, and forgiving of errors. Predesigned output should be attractive and easy to understand, with an appropriate level of detail.

Change font size

[**help**](javascript://)

[Main content](https://ng.cengage.com/static/nbreader/ui/apps/nbreader/nbreader.html#header)

## 8.3.4Use Models and Prototypes

From a user’s viewpoint, the interface is the most critical part of the system design because it is where he or she interacts with the system—perhaps for many hours each day. It is essential to construct models and prototypes for user approval. An interface designer should obtain as much feedback as possible, as early as possible. Initial screen designs can be presented to users in the form of a **storyboard**, which is a sketch that shows the general screen layout and design. The storyboard can be created with software or drawn freehand. Users must test all aspects of the interface design and provide feedback to the designers. User input can be obtained in interviews, via questionnaires, and by observation. Interface designers also can obtain data, called [**usability metrics**](javascript://), by using software that can record and measure user interaction with the system.

Change font size

[**help**](javascript://)

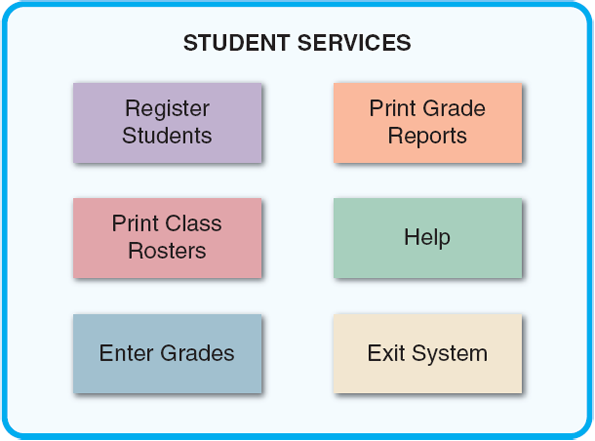
[Main content](https://ng.cengage.com/static/nbreader/ui/apps/nbreader/nbreader.html#header)

## 8.3.5Focus on Usability

The user interface should include all tasks, commands, and communications between users and the information system. The opening screen should show the main options ([Figure 8-4](javascript://) is an illustration). Each screen option leads to another screen, with more options. The objective is to offer a reasonable number of choices that a user easily can comprehend. Too many options on one screen can confuse a user—but too few options increase the number of submenu levels and complicate the navigation process. Often, an effective strategy is to present the most common choice as a default but allow the user to select other options.

**Figure 8-4**

The opening screen displays the main options for a student registration system. A user can click an option to see lower-level actions and menu choices.



Enlarge Image

Change font size

[**help**](javascript://)

[Main content](https://ng.cengage.com/static/nbreader/ui/apps/nbreader/nbreader.html#header)

## 8.3.6Invite Feedback

Even after the system is operational, it is important to monitor system usage and solicit user suggestions. The analyst can determine if system features are being used as intended by observing and surveying users. Sometimes, full-scale operations highlight problems that were not apparent when the prototype was tested. Based on user feedback, Help screens might need revision and design changes to allow the system to reach its full potential.

Change font size

[**help**](javascript://)

[Main content](https://ng.cengage.com/static/nbreader/ui/apps/nbreader/nbreader.html#header)

## 8.3.7Document Everything

All screen designs should be documented for later use by programmers. If a CASE tool or screen generator is being used, the screen designs should be numbered and saved in a hierarchy similar to a menu tree. User-approved sketches and storyboards also can be used to document the user interface.

By applying basic user-centered design principles, a systems analyst can plan, design, and deliver a successful user interface.

Change font size

[**help**](javascript://)

[Main content](https://ng.cengage.com/static/nbreader/ui/apps/nbreader/nbreader.html#header)

**8.4**Guidelines for User Interface Design

A system might have advanced technology and powerful features, but the *real* test is whether users like it and feel that it meets their needs. What follows is a set of general guidelines for successful user interface design. These guidelines are distilled from years of best practices in the industry. There is some overlap because many of the main guidelines share common elements.

Although there is no standard approach to interface design, these guidelines are a starting point suitable for traditional systems development. User interface development for web applications or for mobile apps has its own unique considerations, above and beyond these general guidelines. Perhaps the *most* important guideline is that not all of these recommendations must be followed—the best interface is the one that works best for the users.

Change font size

[**help**](javascript://)

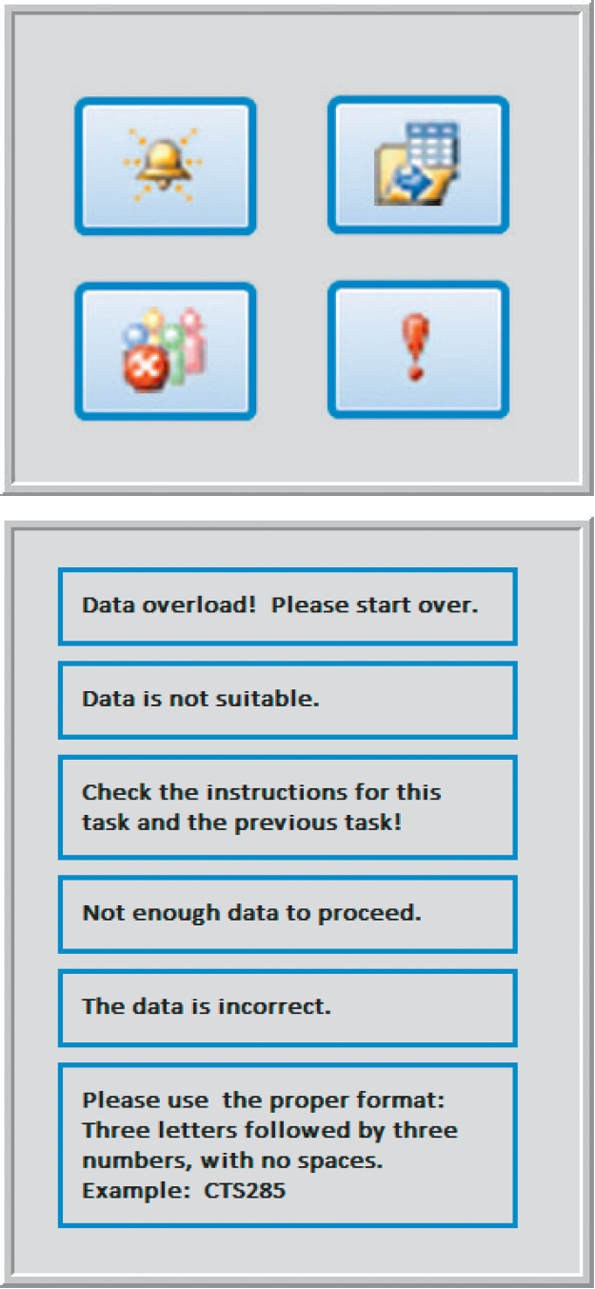
[Main content](https://ng.cengage.com/static/nbreader/ui/apps/nbreader/nbreader.html#header)

## 8.4.1Create an Interface That Is Easy to Learn and Use

1. Focus on system design objectives, rather than calling attention to the interface.
2. Create a design that is easy to understand and remember. Maintain a common design in all modules of the interface, including the use of color, screen placements, fonts, and the overall “look and feel.”
3. Provide commands, actions, and system responses that are consistent and predictable.
4. Allow users to correct errors easily.
5. Clearly label all controls, buttons, and icons.
6. Select familiar images that users can understand, and provide on-screen instructions that are logical, concise, and clear. For example, the top screen in [Figure 8-5](javascript://) shows four control buttons, but none of them has an obvious meaning. In the bottom screen, the first five messages provide little or no information. The last message is the only one that is easy to understand.

**Figure 8-5**

In the example at the top, the icons do not have a clear message. In the Help text examples at the bottom, only one message is understandable. The others would frustrate and annoy most users.



Enlarge Image

1. Show all commands in a list of menu items, but dim any commands that are not available to the user.
2. Make it easy to navigate or return to any level in the menu structure.

Change font size

[**help**](javascript://)

[Main content](https://ng.cengage.com/static/nbreader/ui/apps/nbreader/nbreader.html#header)

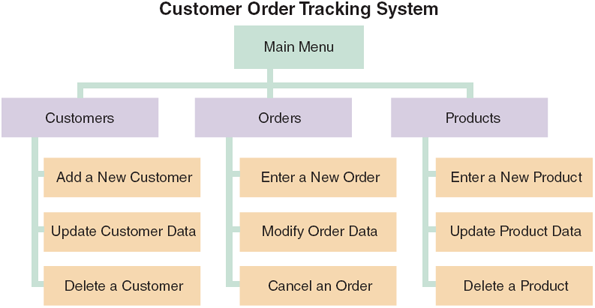
## 8.4.2Enhance User Productivity

The interface is where a user interacts with the system, so it can have a dramatic effect on productivity. If the interface empowers a user and enables him or her to handle more complex tasks, the user becomes more productive. Conversely, if the interface is difficult to work with, productivity declines.

1. Organize tasks, commands, and functions in groups that resemble actual business operations. Group functions and submenu items in a multilevel menu hierarchy, or tree, that is logical and reflects how users typically perform the tasks. [Figure 8-6](javascript://) shows an example of a menu hierarchy for an order tracking system.

**Figure 8-6**

This menu hierarchy shows tasks, commands, and functions organized into logical groups and sequences. The structure resembles a functional decomposition diagram (FDD), which is a model of business functions and processes.



Enlarge Image

1. Create alphabetical menu lists or place the selections used frequently at the top of the menu list. No universally accepted approach to menu item placement exists. The best strategy is to design a prototype and obtain feedback from users. Some applications even allow menus to show recently used commands first. Some users like that feature, but others find it distracting. The best approach is to offer a choice and let users decide.
2. Provide shortcuts for experienced users so they can avoid multiple menu levels. Shortcuts can be created using hot keys that allow a user to press the Alt key + the underlined letter of a command.
3. Use default values if the majority of values in a field are the same. For example, if 90% of the firm’s customers live in Albuquerque, use Albuquerque as the default value in the City field.
4. Use a duplicate value function that enables users to insert the value from the same field in the previous record, but allow users to turn this feature on or off as they prefer.
5. Provide a fast-find feature that displays a list of possible values as soon as users enter the first few letters.
6. If available, consider a [**natural language**](javascript://) feature that allows users to type commands or requests in normal text phrases. For example, many applications allow users to request Help by typing a question into a dialog box. The software then uses natural language technology to retrieve a list of topics that match the request. Natural language technology is used in speech recognition systems, text-to-speech synthesizers, automated voice response systems, web search engines, text editors, and language instruction materials.

Change font size

[**help**](javascript://)

[Main content](https://ng.cengage.com/static/nbreader/ui/apps/nbreader/nbreader.html#header)

## 8.4.3Provide Flexibility

Suppose that a user wants a screen display of all customer balances that exceed $5,000 in an accounts receivable system. How should that feature be designed? The program could be coded to check customer balances against a fixed value of 5000, which is a simple solution for both the programmer and the user because no extra keystrokes are required to produce the display. However, that approach is inflexible. A better approach would be to let the user enter the amount. Or start with a [**default value**](javascript://) that displays automatically. Users can press ENTER to accept the value or type in another value. Often the best design strategy is to offer several alternatives, so users can decide what will work best for them.

Change font size

[**help**](javascript://)

[Main content](https://ng.cengage.com/static/nbreader/ui/apps/nbreader/nbreader.html#header)

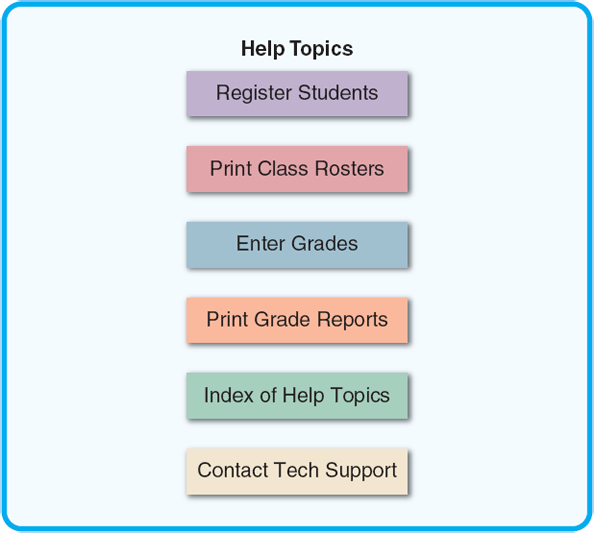
## 8.4.4Provide Users with Help and Feedback

This is one of the most important guidelines because it has a high impact on users. Never allow Help to slow a user down. Instead, make Help easy to find but not around when users don’t need it.

1. Ensure that help is always available on demand. Help screens should provide information about menu choices, procedures, shortcuts, and errors.
2. Provide user-selected help and context-sensitive help. User-selected help displays information when the user requests it. By making appropriate choices through the menus and submenus, the user eventually reaches a screen with the desired information. [Figure 8-7](javascript://) shows the main Help screen for the student registration system. [**Context-sensitive**](javascript://) help offers assistance for the task in progress.

**Figure 8-7**

The main Help screen for a student registration system.



Enlarge Image

1. Provide a direct route for users to return to the point from where help was requested. Title every help screen to identify the topic, and keep help text simple and concise. Insert blank lines between paragraphs to make Help easier to read, and provide examples where appropriate.
2. Include contact information, such as a telephone extension or email address if a department or help desk is responsible for assisting users.
3. Require user confirmation before data deletion (Are you sure?) and provide a method of recovering data that is deleted inadvertently. Build in safeguards that prevent critical data from being changed or erased.
4. Provide an “Undo” key or a menu choice that allows the user to undo the results of the most recent command or action.
5. When a user-entered command contains an error, highlight the erroneous part and allow the user to make the correction without retyping the entire command.
6. Use hypertext links to assist users as they navigate help topics.
7. Display messages at a logical place on the screen, and be consistent.
8. Alert users to lengthy processing times or delays. Give users an on-screen progress report, especially if the delay is lengthy.
9. Allow messages to remain on the screen long enough for users to read them. In some cases, the screen should display messages until the user takes some action.
10. Let the user know whether the task or operation was successful or not. For example, use messages such as Update completed, All transactions have been posted, or The ID Number was not found.
11. Provide a text explanation if an icon or image is used on a control button. This can be accomplished using a “mouse hover” to display a pop-up box with explanation when the mouse is moved over an icon or image.
12. Use messages that are specific, understandable, and professional. Avoid messages that are cute, cryptic, or vague, such as ERROR—You have entered an unacceptable value, or Error DE-4-16. Better examples are as follows:
    * Enter a number from 1 (low) to 5 (high)
    * Customer number must be numeric
    * Please re-enter a numeric value
    * Call the Accounting Department, Ext. 239 for assistance

Change font size

[**help**](javascript://)

[Main content](https://ng.cengage.com/static/nbreader/ui/apps/nbreader/nbreader.html#header)

## 8.4.5Create an Attractive Layout and Design

This is a subjective area because reasonable people can differ on what is attractive. The analyst should consider color, layout, and ease of use. Screen mock-ups and menu trees can be created and tried on users to get their input. If in doubt, err on the side of doing a bit less. For example, blinking messages initially may seem like a good idea, they might not be the best choice for the interface design. Also try to avoid too many fonts, styles, and sizes, which can be distracting. Each separate style should communicate something—a different level of detail, another topic, mandatory versus optional actions, and so on.

1. Use appropriate colors to highlight different areas of the screen; avoid gaudy and bright colors.
2. Use special effects sparingly. For example, animation and sound might be effective in some situations, but too many special effects can be distracting and annoying to a user, especially if he or she must view them repeatedly.
3. Use hyperlinks that allow users to navigate to related topics.
4. Group related objects and information. Visualize the screen the way a user will see it, and simulate the tasks that the user will perform.
5. Screen density is important. Keep screen displays uncluttered, with enough white space to create an attractive, readable design.
6. Display titles, messages, and instructions in a consistent manner and in the same general locations on all screens.
7. Use consistent terminology. For example, do not use the terms delete, cancel, and erase to indicate the same action. Similarly, the same sound always should signal the same event.
8. Ensure that commands always will have the same effect. For example, if the BACK control button returns a user to the prior screen, the BACK command always should perform that function throughout the application.
9. Ensure that similar mouse actions will produce the same results throughout the application. The results of pointing, clicking, and double-clicking should be consistent and predictable.
10. When the user enters data that completely fills the field, do not move automatically to the next field. Instead, require the user to confirm the entry by pressing the Enter key or Tab key at the end of every fill-in field.
11. Remember that users are accustomed to a pattern of red = stop, yellow = caution, and green = go. Stick to that pattern and use it when appropriate to reinforce on-screen instructions.
12. Provide a keystroke alternative for each menu command, with easy-to-remember letters, such as File, Exit, and Help.
13. Use familiar commands if possible, such as Cut, Copy, and Paste.
14. Provide a Windows look and feel in the interface design if users are familiar with Windows-based applications.
15. Avoid complex terms and technical jargon; instead, select terms that come from everyday business processes and the vocabulary of a typical user.

Change font size

[**help**](javascript://)

[Main content](https://ng.cengage.com/static/nbreader/ui/apps/nbreader/nbreader.html#header)

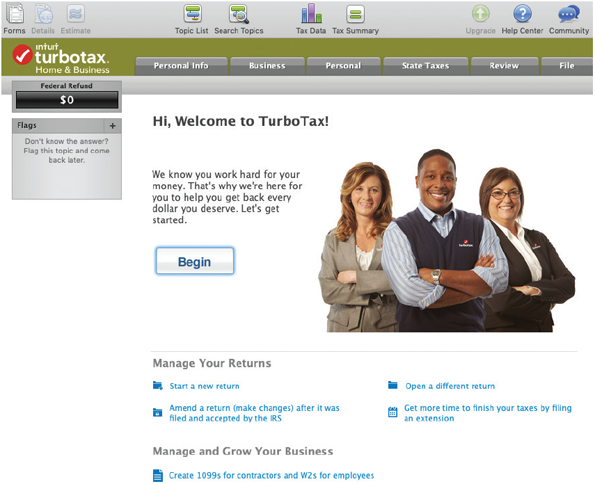
## 8.4.6Enhance the Interface

A designer can include many features, such as menu bars, toolbars, dialog boxes, text boxes, toggle buttons, list boxes, scroll bars, drop-down list boxes, option buttons, check boxes, command buttons, and calendar controls, among others. Screen design requires a sense of aesthetics as well as technical skills. User feedback should be obtained early and often as the design process continues.

1. The opening screen is especially important because it introduces the application and allows users to view the primary options. The starting point can be a [**switchboard**](javascript://) with well-placed command buttons that allow users to navigate the system. [Figure 8-8](javascript://) shows the switchboard of TurboTax introducing a tax preparation program. The main options are clearly displayed on an uncluttered screen. The addition of the picture showing tax professionals ready to help provides a sense of calm and confidence to the user—something particularly important for users who are likely to be confused and/or nervous when beginning the tax preparation process.

**Figure 8-8**

An example of a switchboard introducing TurboTax. The main options are clearly displayed on an uncluttered screen—something particularly important for users who are likely to be confused and/or nervous when beginning the tax preparation process.



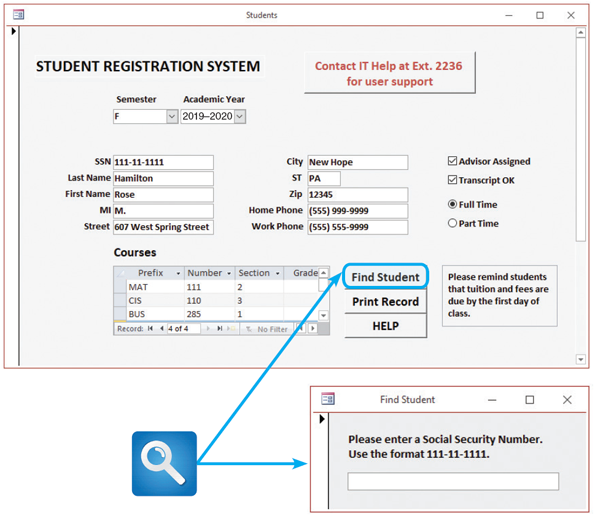
Enlarge Image

**Source:** Intuit

1. Use a [**command button**](javascript://) to initiate an action such as printing a form or requesting help. For example, when a user clicks the Find Student command button in [Figure 8-9](javascript://), a dialog box opens with instructions.

**Figure 8-9**

A data entry screen for a student registration system. This screen uses several design features that are described in the text. When a user clicks the Find Student command button, a dialog box is displayed with instructions.



Enlarge Image

1. If a software package is being used, check to see if it allows the creation of customized [**menu bars**](javascript://) and toolbars. If so, consider these options.
2. Add a shortcut feature that lets a user select a menu command either by clicking the desired choice or by pressing the Alt key + the underlined letter. Some forms also use a [**toolbar**](javascript://) that contains icons or buttons that represent shortcuts for executing common commands.
3. If variable input data is needed, provide a [**dialog box**](javascript://) that explains what is required.
4. A [**toggle button**](javascript://) makes it easy to show on or off status—clicking the toggle button switches to the other state.
5. Use [**list boxes**](javascript://) that display the available choices. If the list does not fit in the box, a [**scroll bar**](javascript://) allows the user to move through the available choices. Also, be sure to provide another way to enter data that does not align with a specific list choice.
6. Use an [**option button**](javascript://), or [**radio button**](javascript://), to control user choices. For example, if the user can select only one option, display a suitable message (Choose one item), but if there is no restriction, display a different message (Choose all that apply). Use a black dot to show selected options.
7. If [**check boxes**](javascript://) are used to select one or more choices from a group, show the choices with a checkmark or an X.
8. When dates must be entered, use a [**calendar control**](javascript://) that allows the user to select a date that the system will use as a field value.

Change font size

[**help**](javascript://)

[Main content](https://ng.cengage.com/static/nbreader/ui/apps/nbreader/nbreader.html#header)

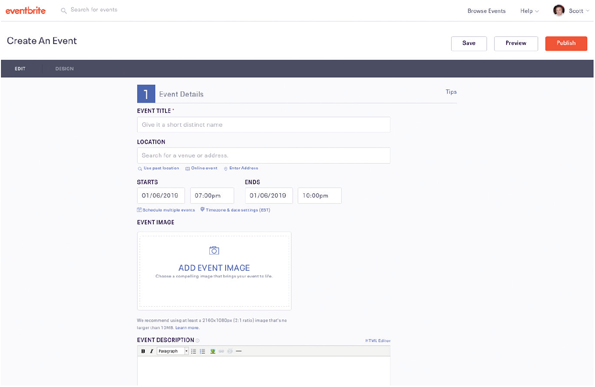
## 8.4.7Focus on Data Entry Screens

Data entry is especially important because it is in the job description of so many users.

1. Whenever possible, use a data entry method called [**form filling**](javascript://), where a blank form that duplicates or resembles the source document is completed on the screen.
2. Restrict user access to screen locations where data is entered. For example, in the Eventbrite event management data entry when the screen in [Figure 8-10](javascript://) appears, the system should position the insertion point in the first data entry location. After the operator enters an event title, the insertion point should move automatically to the entry location for the next field (Location). A user should be able to position the insertion point only in places where data is entered on the form.

**Figure 8-10**

In this data entry screen for creating events, the system generates start and end dates and times automatically, but these can be changed by the user at any time. A red asterisk, as shown beside Event Title, indicates required fields. Gray text within the data entry field lets the user know what information to provide. This text is replaced with user input.



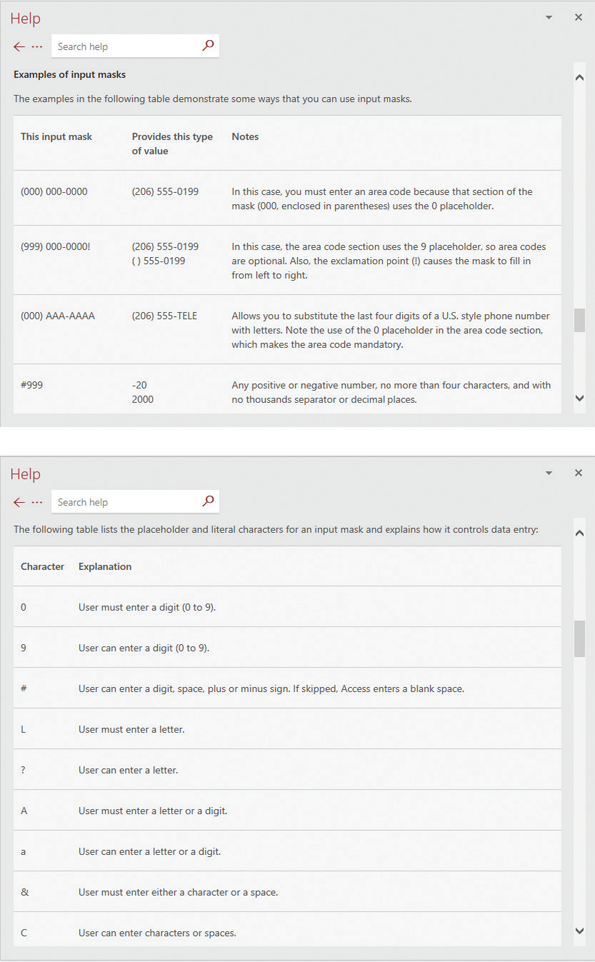
Enlarge Image

**Source:** Eventbrite

1. Provide a way to leave the data entry screen at any time without entering the current record, such as a “Cancel” button. Since the application shown in [Figure 8-10](javascript://) is web based, this can be accomplished by selecting the “Back” button of the browser (not shown).
2. Provide a descriptive caption for every field, and show the user where to enter the data and the required or maximum field size. Typically, white boxes show the location and length of each field. Other methods used to indicate field locations are video highlighting, underscores, special symbols, or a combination of these features.
3. Provide a means for users to move among fields on the form in a standard order or in any order they choose. In a GUI, the user can override the standard field order and select field locations using the mouse or arrow keys.
4. Allow users to add, change, delete, and view records. Messages such as Apply these changes? (Y/N) or Delete this record? (Y/N) should require users to confirm the actions. Highlighting the letter N as a default response will avoid problems if the user presses the Enter key by mistake.
5. Design the screen form layout to match the layout of the source document. If the source document fields start at the top of the form and run down in a column, the input screen should use the same design.
6. Display a sample format if a user must enter values in a field in a specific format. For example, provide an on-screen instruction to let users know that the date format is MMDDYY, and provide an example if the user must enter separators, such as slashes.
7. In addition to the sample format in the preceding rule, it might be better to use an [**input mask**](javascript://), which is a template or pattern that restricts data entry and prevents errors. Microsoft Access provides standard input masks for fields such as dates, telephone numbers, postal codes, and Social Security numbers. In addition, custom input masks can be created, as shown in [Figure 8-11](javascript://). Note that a mask can have a specific format. For example, if a user enters text in lowercase letters, the input mask will capitalize the first letter automatically.

**Figure 8-11**

Microsoft Access provides various input masks for dates, phone numbers, and postcodes, among others. In addition, it is easy to create a custom mask using the characters shown here.



Enlarge Image

**Source:** Microsoft Corporation

1. Require an ending keystroke for every field. Pressing the Enter key or the Tab key should signify the end of a field entry. Avoid a design that moves automatically to the next item when the field is full. The latter approach requires an ending keystroke only when the data entered is less than the maximum field length. It is confusing to use two different data entry procedures.
2. Do not require users to type leading zeroes for numeric fields. For example, if a three-digit project number is 045, the operator should be able to type 45 instead of 045 before pressing the Enter key. An exception to that rule might occur when entering a date, where a leading zero is needed to identify single-digit months or days, such as 06-04-2013.
3. Do not require users to type trailing zeroes for numbers that include decimals. For example, when a user types a value of 98, the system should interpret the value as 98.00 if the field has been formatted to include numbers with two decimal places. The decimal point is needed only to indicate nonzero decimal places, such as 98.76.
4. Display default values so operators can press the Enter key to accept the suggested value. If the default value is not appropriate, the operator can change it.
5. Use a default value when a field value will be constant for successive records or throughout the data entry session. For example, if records are input in order by date, the date used in the first transaction should be used as the default date until a new date is entered, at which time the new date becomes the default value.
6. Display a list of acceptable values for fields, and provide meaningful error messages if the user enters an unacceptable value. An even better method, which was described under Rule 5: Enhance the Interface, is to provide a drop-down list box containing acceptable values that allows the user to select a value by clicking.
7. Provide users with an opportunity to confirm the accuracy of input data before entering it by displaying a message such as, Add this record? (Y/N). A positive response (Y) adds the record, clears the entry fields, and positions the insertion point in the first field so the user can input another record. If the response is negative (N), the current record is not added and the user can correct the errors.

Data entry screens should also anticipate future needs. Consider a parts inventory database that contains a one-character field for category, such as electrical, mechanical, or hydraulic. The design works well, but what if the company decides to break these overall groups down into more specific segments? A better design would anticipate possible expansion to two or more characters. For example, in 1999, there was widespread concern about what was called the Y2K issue because many older programs used only two characters to store the year and might not recognize the start of a new century.

Change font size

[**help**](javascript://)

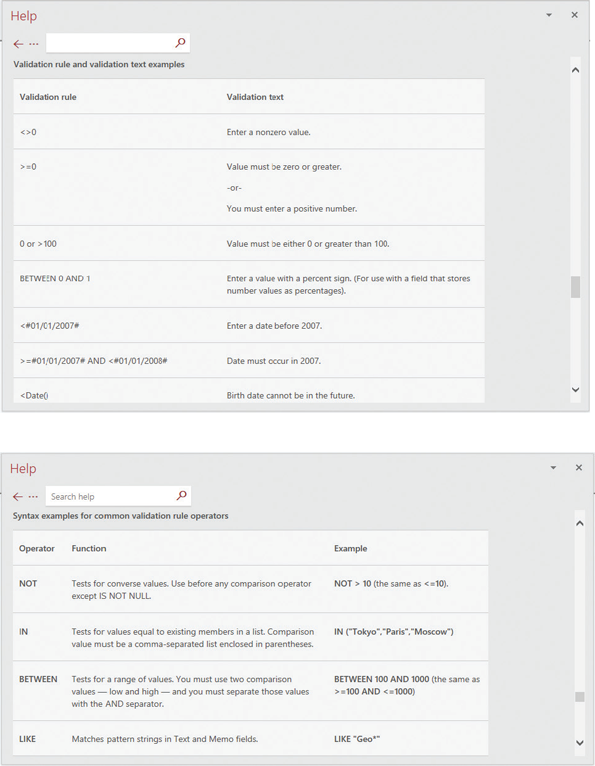
[Main content](https://ng.cengage.com/static/nbreader/ui/apps/nbreader/nbreader.html#header)

## 8.4.8Use Validation Rules

Reducing input errors improves data quality. One way to reduce input errors is to eliminate unnecessary data entry. For example, a user cannot misspell a customer name if it is not entered or is entered automatically based on the user entering the customer ID. Similarly, an outdated item price cannot be used if the item price is retrieved from a master file instead of being entered manually. The best defense against incorrect data is to identify and correct errors before they enter the system by using data validation rules, as shown in [Figure 8-12](javascript://). A [**data validation rule**](javascript://) improves input quality by testing the data and rejecting any entry that fails to meet specified conditions. The design can include at least eight types of data validation rules.

**Figure 8-12**

Microsoft Access provides validation rules that can improve data quality by requiring the input to meet specific requirements or conditions.



Enlarge Image

**Source:** Microsoft Corporation

1. A [**sequence check**](javascript://) can be used when the data must be in some predetermined sequence. If the user must enter work orders in numerical sequence, for example, then an out-of-sequence order number indicates an error, or if the user must enter transactions chronologically, then a transaction with an out-of-sequence date indicates an error.
2. An [**existence check**](javascript://) can apply to mandatory data items. For example, if an employee record requires a Social Security number, an existence check would not allow the user to save the record until he or she enters a suitable value in the Social Security number field.
3. A [**data type check**](javascript://) can test to ensure that a data item fits the required data type. For example, a numeric field must have only numbers or numeric symbols, and an alphabetic field can contain only the characters A through Z (or a through z).
4. A [**range check**](javascript://) can be used to verify that data items fall between a specified minimum and maximum value. The daily hours worked by an employee, for example, must fall within the range of 0 to 24. When the validation check involves a minimum or a maximum value, but not both, it is called a [**limit check**](javascript://). Checking that a payment amount is greater than zero, but not specifying a maximum value, is an example of a limit check.
5. A [**reasonableness check**](javascript://) identifies values that are questionable but not necessarily wrong. For example, input payment values of $0.05 and $5,000,000.00 both pass a simple limit check for a payment value greater than zero, and yet both values could be errors. Similarly, a daily-hours-worked value of 24 passes a 0 to 24 range check; however, the value seems unusual, and the system should verify it using a reasonableness check.
6. A [**validity check**](javascript://) can be used for data items that must have certain values. For example, if an inventory system has 20 valid item classes, then any input item that does not match one of the valid classes will fail the check. Verifying that a customer number on an order matches a customer number in the customer file is another type of validity check. Because the value entered must refer to another value, that type of check also is called referential integrity, which is explained in [Chapter 9](javascript://), Data Design. Another validity check might verify that a new customer number does not match a number already stored in the customer master file.
7. A [**combination check**](javascript://) is performed on two or more fields to ensure that they are consistent or reasonable when considered together. Even though all the fields involved in a combination check might pass their individual validation checks, the combination of the field values might be inconsistent or unreasonable. For example, if an order input for 30 units of a particular item has an input discount rate applicable only for purchases of 100 or more units, then the combination is invalid; either the input order quantity or the input discount rate is incorrect.
8. [**Batch controls**](javascript://) are totals used to verify batch input. Batch controls might check data items such as record counts and numeric field totals. For example, before entering a batch of orders, a user might calculate the total number of orders and the sum of all the order quantities. When the batch of orders is entered, the order system also calculates the same two totals. If the system totals do not match the input totals, then a data entry error has occurred. Unlike the other validation checks, batch controls do not identify specific errors. For example, if the sum of all the order quantities does not match the batch control total, the only thing known is that one or more orders in that batch were entered incorrectly or not input. The batch control totals often are called [**hash totals**](javascript://) because they are not meaningful numbers themselves but are useful for comparison purposes.

Change font size

[**help**](javascript://)

[Main content](https://ng.cengage.com/static/nbreader/ui/apps/nbreader/nbreader.html#header)

## 8.4.9Manage Data Effectively

In addition to its effect on users, data management impacts company efficiency, productivity, and security. To reduce input errors, the system should enter and verify data as soon as possible, and each data item should have a specific type, such as alphabetic, numeric, or alphanumeric, and a range of acceptable values.

It is important to collect input data as close to its source as possible. For instance, using barcode scanners rather than manual forms on a warehouse freight dock, or having salespeople use tablets to record orders rather than filling in source documents. The easiest, most accurate, and least expensive data input strategy is automated data capture.

In an efficient design, data is entered only once. For example, if input data for a payroll system also is needed for a human resources system, the analyst could design an interface to transfer data automatically, or a central data storage area could be created that both systems can access. [Chapter 9](javascript://) describes normalization, which is a set of rules that can help avoid data design problems. A secure system also includes [**audit trails**](javascript://) that can log every instance of data entry and changes. For example, the system should record when a customer’s credit limit was set, by whom, and any other information necessary to construct the history of a transaction.

Change font size

[**help**](javascript://)

[Main content](https://ng.cengage.com/static/nbreader/ui/apps/nbreader/nbreader.html#header)

## 8.4.10Reduce Input Volume

This is the final guideline, but in some ways it should be the first because it affects all the rest. When input volume is reduced, unnecessary labor costs are avoided, which in turn gets the data into the system more quickly and decreases the number of errors. Therefore, the analyst should start by reducing the number of data items required for each transaction.

1. Input necessary data only. Do not input a data item unless it is needed by the system. A completed order form, for example, might contain the name of the clerk who took the order. If that data is not needed by the system, the user should not enter it.
2. Do not input data that the user can retrieve from system files or calculate from other data. This reduces input errors and data inconsistencies.
3. Do not input constant data. If orders are in batches with the same date, then a user should enter the order date only once for the first order in the batch. If orders are entered online, then the user can retrieve the order date automatically using the current system date.
4. Use codes. Codes are shorter than the data they represent, and coded input can reduce data entry time. Codes are discussed more in [Chapter 9](javascript://).

**Case in Point 8.2**

### Boolean Toys

Suppose you are a systems analyst studying the order processing system at Boolean Toys, a fast-growing developer of software for preschool children. You know that many data entry users have complained about the input screens. Some users would prefer to rearrange the order of the fields, others would like to change the background color on their screens, and still others want shortcuts that would allow them to avoid a series of introductory screens.

What if Boolean’s users could customize their own data entry screens without assistance from the IT staff by using a menu-driven utility program? What would be the pros and cons of such an approach? When should a systems analyst decide a design issue, and when should users be allowed to select what works best for them?

Change font size

[**help**](javascript://)

[Main content](https://ng.cengage.com/static/nbreader/ui/apps/nbreader/nbreader.html#header)

**8.5**Source Document and Form Design

No matter how data enters an information system, the quality of the output is only as good as the quality of the input. The term [**garbage in, garbage out (GIGO)**](javascript://) is familiar to IT professionals, who know that the best time to avoid problems is when the data is entered. The main objective is to ensure the quality, accuracy, and timeliness of input data. Unfortunately, the dream of a “paperless office” has never been realized. Even with RFID technology and automated data capture, we still enter data on source documents and forms, and instead of a human-computer interface, systems analysts must deal with the challenge of a human-paper interface.

A [**source document**](javascript://) collects input data, triggers or authorizes an input action, and provides a record of the original transaction. During the input design stage, the analyst develops source documents that are easy to complete and use for data entry. Source documents generally are paper based but also can be provided online. Either way, the design considerations are the same.

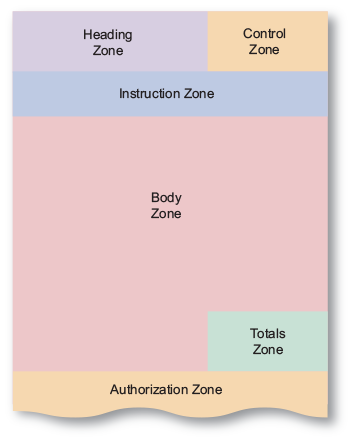
Consider a time when it was a struggle to complete a poorly designed form. There might have been insufficient space, confusing instructions, or poor organization—all symptoms of incorrect [**form layout**](javascript://).

Good form layout makes the form easy to complete and provides enough space, both vertically and horizontally, for users to enter the data. A form should indicate data entry positions clearly using blank lines or boxes and descriptive captions. Also consider using check boxes whenever possible, so users can select choices easily. However, be sure to include an option for any input that does not match a specific check box.

The placement of information on a form also is important. Source documents typically include most of the zones shown in [Figure 8-13](javascript://). The heading zone usually contains the company name or logo and the title and number of the form. The control zone contains codes, identification information, numbers, and dates that are used for storing completed forms. The instruction zone contains instructions for completing the form. The main part of the form, called the body zone, usually takes up at least half of the space on the form and contains captions and areas for entering variable data. If totals are included on the form, they appear in the [**totals zone**](javascript://). Finally, the [**authorization zone**](javascript://) contains any required signatures.

**Figure 8-13**

Source document zones.



Information should flow on a form from left to right and top to bottom to match the way users read documents naturally. That layout makes the form easy to use for the individual who completes the form and for users who enter data into the system using the completed form.

The same user-friendly design principles also apply to printed forms such as invoices and monthly statements, except that heading information usually is preprinted. Column headings should be short but descriptive, avoiding nonstandard abbreviations, with reasonable spacing between columns for better readability.

The order and placement of printed fields should be logical, and totals should be identified clearly. When designing a preprinted form, contact the form’s vendor for advice on paper sizes, type styles and sizes, paper and ink colors, field placement, and other important form details. The goal is to design a form that is attractive, readable, and effective.

Layout and design also are important on web-based forms. There are many resources that will help with designing efficient, user-friendly forms. These include websites that must conform to the U.S. Federal Government’s accessibility guidelines, which can be found online at [http://www.section508.gov](http://www.section508.gov/" \t "_blank).

Change font size

[**help**](javascript://)

[Main content](https://ng.cengage.com/static/nbreader/ui/apps/nbreader/nbreader.html#header)

**8.6**Printed Output

Before designing printed output, there are several questions to consider:

* Why is this being delivered as printed output, rather than screen-based information, with an option for users to view, print, or save as needed?
* Who wants the information, why is it needed, and how will it be used?
* What specific information will be included?
* Will the printed output be designed for a specific device?
* When and how will the information be delivered, and how often must it be updated?
* Do security or confidentiality issues exist? How will they be managed?

The design process should not begin until these questions have been answered. Some information probably was gathered during the systems analysis phase. To gain more understanding, the analyst should meet with users to find out exactly what kind of output *they* are requesting. Prototypes and mock-ups can be used to obtain feedback throughout the design process.

Change font size

[**help**](javascript://)

[Main content](https://ng.cengage.com/static/nbreader/ui/apps/nbreader/nbreader.html#header)

## 8.6.1Report Design

Although many organizations strive to reduce the flow of paper and printed reports, few firms have been able to eliminate printed output totally. Because they are portable, printed reports are convenient and even necessary in some situations. Many users find it handy to view screen output and then print the information they need for a discussion or business meeting. Printed output also is used in [**turnaround documents**](javascript://), which are output documents that are later entered back into the same or another information system. In some areas, the telephone or utility bill, for example, might be a turnaround document printed by the company’s billing system. When the required portion of the bill is returned with payment, the bill is scanned into the company’s accounts receivable system to record the payment accurately.

Designers use a variety of styles, fonts, and images to produce reports that are attractive and user-friendly. Whether printed or viewed on-screen, reports must be easy to read and well organized. Rightly or wrongly, some managers judge an entire project by the quality of the reports they receive.

Database programs such as Microsoft Access include a variety of report design tools, including a Report Wizard, which is a menu-driven feature that designers can use to create reports quickly and easily. Many online web-based database systems also provide similar report design guidelines.

Although the vast majority of reports are designed graphically, some systems still produce one or more [**character-based reports**](javascript://) that use a character set with fixed spacing. Printing character-based reports on high-speed impact printers is a fast, inexpensive method for producing large-scale reports, such as payroll or inventory reports, or registration rosters at a school. This is especially true if multiple copies are required.

Users should approve all report designs in advance. The best approach is to prepare a sample report, called a [**mock-up**](javascript://), or prototype, for users to review. The sample should include typical field values and contain enough records to show all the design features. Depending on the type of printed output, a Microsoft Word document can be created, or a report generator used, to create mock-up reports.

Change font size

[**help**](javascript://)

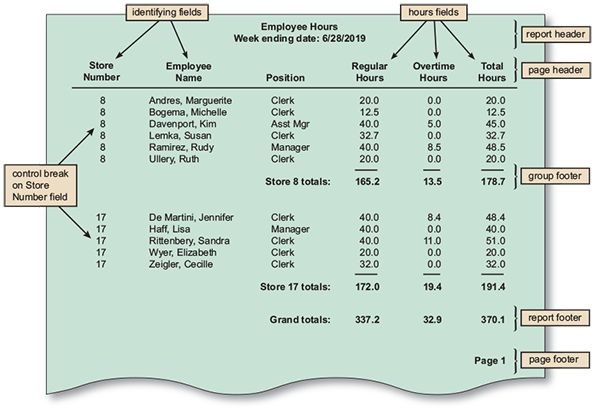
[Main content](https://ng.cengage.com/static/nbreader/ui/apps/nbreader/nbreader.html#header)

## 8.6.2Report Design Principles

Printed reports must be attractive, professional, and easy to read. For example, a well-designed report should provide totals and subtotals for numeric fields. In the report shown in [Figure 8-14](javascript://), note that when the value of a control field, such as Store Number, changes, a [**control break**](javascript://) occurs. A control break usually causes specific actions, such as printing subtotals for a group of records. That type of detail report is called a [**control break report**](javascript://). To produce a control break report, the records must be arranged, or sorted, in [**control field order**](javascript://).

**Figure 8-14**

The Employee Hours report is a detailed report with control breaks, subtotals, and grand totals. Note that a report header identifies the report, a page header contains column headings, a group footer contains subtotals for each store, a report footer contains grand totals, and a page footer identifies the page number.



Enlarge Image

Good report design requires effort and attention to detail. To produce a well-designed report, the analyst must consider design features such as report headers and footers, page headers and footers, column headings and alignment, column spacing, field order, and grouping of detail lines.

### Report Headers and Footers

Every report should have a report header and a report footer. The [**report header**](javascript://), which appears at the beginning of the report, identifies the report and contains the report title, date, and other necessary information. The [**report footer**](javascript://), which appears at the end of the report, can include grand totals for numeric fields and other end-of-report information, as shown in [Figure 8-14](javascript://).

### Page Headers and Footers

Every page should include a [**page header**](javascript://), which appears at the top of the page and includes the column headings that identify the data. The headings should be short but descriptive. Avoid abbreviations unless the users will understand them clearly. Either a page header or a [**page footer**](javascript://), which appears at the bottom of the page, is used to display the report title and the page number.

Database programs such as Microsoft Access make it easy to create groups and subgroups based on particular fields. The report can also calculate and display totals, averages, record counts, and other data for any group or subgroup. For example, a large company might want to see total sales and number of sales broken down by product within each of the 50 states. The information shown in [Figure 8-15](javascript://) is part of Access’ online help that refers to a step-by-step process for creating multilevel grouping.

**Figure 8-15**

Microsoft Access includes an easy-to-use tool for grouping data.



Enlarge Image

**Source:** Microsoft Corporation

### Repeating Fields

Report design is an art, not a science. User involvement is essential, but users often don’t know what they want without seeing samples. For example, consider the issue of repeating fields. The sample report in [Figure 8-14](javascript://) repeats the store number on every row. Is that a good thing? The best advice is to ask users what they think and be guided accordingly. A similar issue exists with regard to the overtime hours column. Is it better to print the zero-overtime data, or only print actual hours, so the data stands out clearly? Again, the best answer is usually the one that works best for users.

### Consistent Design

Look and feel are important to users, so reports should be uniform and consistent. When a system produces multiple reports, each report should share common design elements. For example, the date and page numbers should print in the same place on each report page. Abbreviations used in reports also should be consistent. For example, when indicating a numeric value, it is confusing for one report to use #, another NO, and a third NUM. Items in a report also should be consistent. If one report displays the inventory location as a shelf number column followed by a bin number column, that same layout should be used on all inventory location reports.

Change font size

[**help**](javascript://)

[Main content](https://ng.cengage.com/static/nbreader/ui/apps/nbreader/nbreader.html#header)

## 8.6.3Types of Reports

To be useful, a report must include the information that a user needs. From a user’s point of view, a report with too little information is of no value. Too much information, however, can make a report confusing and difficult to understand. When designing reports, the essential goal is to match the report to the user’s specific information needs. Depending on their job functions, users might need one or more of the reports described in the following sections.

### Detail Reports

A [**detail report**](javascript://) produces one or more lines of output for each record processed. Because it contains one or more lines for each record, a detail report can be quite lengthy. Consider, for example, a large auto parts business. If the firm stocks 3,000 parts, then the detail report would include 3,000 detail lines on approximately 50 printed pages. A user who wants to locate any part in short supply has to examine 3,000 detail lines to find the critical items. A better alternative might be an exception report.

### Exception Reports

An [**exception report**](javascript://) displays only those records that meet a specific condition or conditions. Exception reports are useful when the user wants information only on records that might require action but does not need to know the details. For example, a credit manager might use an exception report to identify only those customers with past-due accounts, or a customer service manager might want a report on all packages that were not delivered within a specified time period.

### Summary Reports

Upper-level managers often want to see total figures and do not need supporting details. A sales manager, for example, might want to know total sales for each sales representative but not want a detail report listing every sale made by them. In that case, a [**summary report**](javascript://) is appropriate. Similarly, a personnel manager might need to know the total regular and overtime hours worked by employees in each store but might not be interested in the number of hours worked by each employee.

**Case in Point 8.3**

### Lazy Eddie

Lazy Eddie is a furniture chain specializing in recliners. Their management has asked you to review the large number of printed reports that are distributed to Lazy Eddie’s 35 store managers. Management is not sure that the managers actually read or use the reports, even though the store managers say they want them. Store visits have shown many of the reports end up stacked on top of filing cabinets, seemingly untouched.

To determine if store managers really use the printed reports, management has asked you to create a procedure that requires users to review and justify their information needs. You could design a form that asks if the information still is required, and why. You could try to get users to decide if a report is worth the cost of producing it. How do you proceed?

Change font size

[**help**](javascript://)

[Main content](https://ng.cengage.com/static/nbreader/ui/apps/nbreader/nbreader.html#header)

**8.7**Technology Issues

Unlike early innovations such as the mouse and the inkjet printer, most technology advances today affect both output *and* input. In a very real sense, output and input have become interdependent, as they are in a user interface, and it is difficult to cite examples of changes in one that would not cause, or at least encourage, changes in the other. For example, new touch-screen input technology generates output that must be properly designed and sized for a particular device, which might be a smartphone, a tablet, or a 23-inch desktop monitor.

The following sections discuss output and input technology separately, but interface designers should always be alert to the possible opportunities, or potential problems, of input/output linkage.

Change font size

[**help**](javascript://)

[Main content](https://ng.cengage.com/static/nbreader/ui/apps/nbreader/nbreader.html#header)

## 8.7.1Output Technology

Although business information systems still provide most output as screen displays and printed matter, technology is having an enormous impact on how people communicate and obtain information. This trend is especially important to firms that use information technology to lower their costs, improve employee productivity, and communicate effectively with their customers.

In addition to screen output and printed matter, output can be delivered in many ways. The system requirements document probably identified user output needs. Now, in the systems design phase, the analyst creates the actual forms, reports, documents, and other types of output that might be accessed from workstations, notebooks, tablets, smartphones, and other devices. How the information will be used, stored, and retrieved must also be considered. The following subsections explain various output types and technologies.

### Internet-Based Information Delivery

Millions of firms use the Internet to reach new customers and markets around the world. To support the explosive growth in e-commerce, web designers must provide user-friendly screen interfaces that display output and accept input from customers. For example, a business can link its inventory system to its website so the output from the inventory system is displayed as an online catalog. Customers visiting the site can review the items, obtain current prices, and check product availability.

Another example of web-based output is a system that provides customized responses to product or technical questions. When a user enters a product inquiry or requests technical support, the system responds with appropriate information from an on-site knowledge base. Web-based delivery allows users to download a universe of files and documents to support their information needs. For example, the web provides consumers with instant access to brochures, product manuals, and parts lists, while prospective homebuyers can obtain instant quotes on mortgages, insurance, and other financial services.

To reach prospective customers and investors, companies also use a live or prerecorded [**webcast**](javascript://), which is an audio or video media file distributed over the Internet. Radio and TV stations also use this technique to broadcast program material to their audiences.

### Email

Email is an essential means of internal and external business communication. Employees send and receive email on local or wide area networks, including the Internet. Companies send new product information to customers via email, and financial services companies use email messages to confirm online stock trades. Employees use email to exchange documents, data, and schedules and to share business-related information they need to perform their jobs. In many firms, email has virtually replaced traditional memos and printed correspondence.

### Blogs

Web-based logs, called [**blogs**](javascript://), are another form of web-based output. Because blogs are journals written from a particular point of view, they not only deliver facts to web readers but also provide opinions. Blogs are useful for posting news, reviewing current events, and promoting products.

### Instant Messaging

This popular form of communication is another way for individuals and companies to communicate effectively over the Internet. Although some users feel that it can be a distraction, others like the constant flow of communication, especially as a team member in a collaborative situation.

### Wireless Devices

Messages and data can be transmitted to a wide array of mobile devices, including tablet computers, smartphones, and similar wireless products that combine portable computing power, multimedia capability, and Internet access.

### Digital Audio, Images, and Video

Sounds, images, and video clips can be captured, stored in digital format, and transmitted as output to users who can reproduce the content.

Audio or video output can be attached to an email message or inserted as a clip in a Microsoft Word document. Businesses also use automated systems to handle voice transactions and provide information to customers. For example, using a telephone keypad, a customer can confirm an airline seat assignment, check a credit card balance, or determine the current price of a mutual fund.

If a picture is worth a thousand words, then digital images and video clips certainly are high-value output types that offer a whole new dimension. For example, an insurance adjuster with a digital camera phone can take a picture, submit the image via a wireless device, and receive immediate authorization to pay a claim on the spot. If images are a valuable form of output, video clips are even better in some situations. For example, video clips provide online virtual tours that allow realtors to show off the best features of homes they are marketing. The user can zoom in or out and rotate the image in any direction.

### Automated Fax Systems

An [**automated fax**](javascript://) or [**faxback**](javascript://) system allows a customer to request a fax using email, via the company website, or by telephone. The response is transmitted in a matter of seconds back to the user’s fax machine. Although most users prefer to download documents from the web, many established organizations still offer an automated faxback service as another way to provide immediate response 24 hours a day to a certain set of customers. Certain industries in particular, such as drug stores and doctor’s offices, insurance companies, and real estate brokers, still rely on fax machines as a primary way of communication.

### Podcasts

A [**podcast**](javascript://) is a specially formatted digital audio file that can be downloaded by Internet users from a variety of content providers. Many firms use podcasts as sales and marketing tools and to communicate with their own employees. Using software such as iTunes, users can receive a podcast, launch the file on their computer, and store it on their portable player. Podcasts can include images, sounds, and video.

### Computer Output to Digital Media

This process is used when many paper documents must be scanned, stored in digital format, and retrieved quickly. For example, if an insurance company stores thousands of paper application forms, special software can treat the documents as data and extract information from a particular column or area on the form. Digital storage media can include magnetic tape, CDs, DVDs, and high-density laser disks.

### Specialized Forms of Output

An incredibly diverse marketplace requires many forms of specialized output and devices. For example:

* Portable, web-connected devices that can run multiple apps, handle multimedia output, and provide powerful, multipurpose communication for users
* Retail point-of-sale terminals that handle computer-based credit card transactions, print receipts, and update inventory records
* Automatic teller machines (ATMs) that can process bank transactions and print deposit and withdrawal slips
* Special-purpose printers that can produce labels, employee ID cards, driver’s licenses, gasoline pump receipts, and, in some states, lottery tickets
* Plotters that can produce high-quality images such as blueprints, maps, and electronic circuit diagrams
* Electronic detection of data embedded in credit cards, bank cards, and employee identification cards

Change font size

[**help**](javascript://)

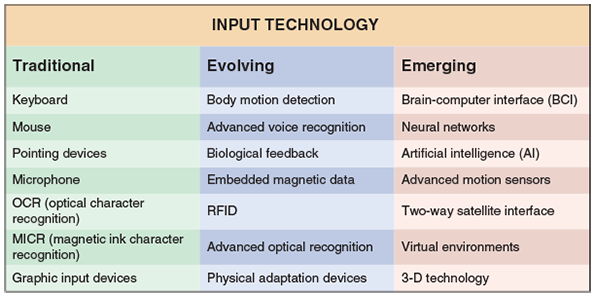
[Main content](https://ng.cengage.com/static/nbreader/ui/apps/nbreader/nbreader.html#header)

## 8.7.2Input Technology

Input technology has changed dramatically in recent years. In addition to traditional devices and methods, there has been a rapid expansion of new hardware and ways to capture and enter data into a system, some of which are shown in [Figure 8-16](javascript://). Businesses are using the new technology to speed up the input process, reduce costs, and capture data in new forms, such as digital signatures.

**Figure 8-16**

Input devices can be very traditional or based on the latest technology.



Enlarge Image

Input methods should be cost-efficient, timely, and as simple as possible. Systems analysts study transactions and business operations to determine how and when data should enter the system. Usually, the first decision is whether to use batch or online input methods. Each method has advantages and disadvantages, and the systems analyst must consider the following factors.

### Batch Input

Using [**batch input**](javascript://), data entry usually is performed on a specified time schedule, such as daily, weekly, monthly, or longer. For example, batch input occurs when a payroll department collects time cards at the end of the week and enters the data as a [**batch**](javascript://). Another example is a school that enters all grades for the academic term in a batch.

### Online Input

Although batch input is used in specific situations, most business activity requires [**online data entry**](javascript://). The online method offers major advantages, including the immediate validation and availability of data. A popular online input method is [**source data automation**](javascript://), which combines online data entry and automated data capture using input devices such as [**radio frequency identification(RFID) tags**](javascript://), magnetic data strips, or even smartphones. Source data automation is fast and accurate and minimizes human involvement in the translation process.

Many large companies use a combination of source data automation and a powerful communication network to manage global operations instantly. Some common examples of source data automation are as follows:

* Businesses that use point-of-sale (POS) terminals equipped with bar code scanners and magnetic swipe scanners to input credit card data
* Automatic teller machines (ATMs) that read data strips on bank cards
* Factory employees who use magnetic ID cards to clock on and off specific jobs so the company can track production costs accurately
* Hospitals that imprint bar codes on patient identification bracelets and use portable scanners when gathering data on patient treatment and medication
* Retail stores that use portable bar code scanners to log new shipments and update inventory data
* Libraries that use handheld scanners to read optical strips on books

### Trade-Offs

Although online input offers many advantages, it does have some disadvantages. For example, unless source data automation is used, manual data entry is slower and more expensive than batch input because it is performed at the time the transaction occurs and often done when computer demand is at its highest.

The decision to use batch or online input depends on business requirements. For example, hotel reservations must be entered and processed immediately, but hotels can enter their monthly performance figures in a batch. In fact, some input occurs naturally in batches. A cable TV provider, for example, receives customer payments in batches when the mail arrives.

Change font size

[**help**](javascript://)

[Main content](https://ng.cengage.com/static/nbreader/ui/apps/nbreader/nbreader.html#header)

**8.8**Security and Control Issues

A company must do everything in its power to protect its data. This includes not only the firm’s own information but that of its customers, employees, and suppliers. Most assets have a value, but corporate data is priceless because without safe, secure, and accurate data, a company cannot function.

The following sections discuss output and input data security and control.

Change font size

[**help**](javascript://)

[Main content](https://ng.cengage.com/static/nbreader/ui/apps/nbreader/nbreader.html#header)

## 8.8.1Output Security and Control

Output must be accurate, complete, current, and secure. Companies use various [**output control**](javascript://) methods to maintain output integrity and security. For example, every report should include an appropriate title, report number or code, printing date, and time period covered. Reports should have pages that are numbered consecutively, identified as Page nn of nn, and the end of the report should be labeled clearly. Control totals and record counts should be reconciled against input totals and counts. Reports should be selected at random for a thorough check of correctness and completeness. All processing errors or interruptions must be logged so they can be analyzed.

[**Output security**](javascript://) protects privacy rights and shields the organization’s proprietary data from theft or unauthorized access. To ensure output security, several important tasks must be performed. First, limit the number of printed copies and use a tracking procedure to account for each copy. When printed output is distributed from a central location, specific procedures should be used to ensure that the output is delivered to authorized recipients only. That is especially true when reports contain sensitive information, such as payroll data. All sensitive reports should be stored in secure areas. All pages of confidential reports should be labeled appropriately.

As shown in [Figure 8-17](javascript://), it is important to shred sensitive reports, out-of-date reports, and output from aborted print runs. Blank check forms must be stored in a secure location and be inventoried regularly to verify that no forms are missing. If signature stamps are used, they must be stored in a secure location away from the forms storage location.

**Figure 8-17**

To maintain output security, it is important to shred sensitive material.



Gang Liu/ [Shutterstock.com](http://shutterstock.com/" \t "_blank)

In most organizations, the IT department is responsible for coordinating output control and security measures. Systems analysts must be concerned with security issues as they design, implement, and support information systems. Whenever possible, security should be designed into the system by using passwords, shielding sensitive data, and controlling user access. Physical security always will be necessary, especially in the case of printed output that is tangible and can be viewed and handled easily.

Enterprise-wide data access creates a whole new set of security and control issues. Many firms have responded to those concerns by installing diskless workstations. A [**diskless workstation**](javascript://) is a network terminal that supports a full-featured user interface but limits the printing or copying of data, except to certain network resources that can be monitored and controlled. This concept would typically preclude the use of portable storage devices, such as USB thumb drives.

Change font size

[**help**](javascript://)

[Main content](https://ng.cengage.com/static/nbreader/ui/apps/nbreader/nbreader.html#header)

## 8.8.2Input Security and Control

[**Input control**](javascript://) includes the necessary measures to ensure that input data is correct, complete, and secure. Input control must be the focus during every phase of input design, starting with source documents that promote data accuracy and quality. When a batch input method is used, the computer can produce an input log file that identifies and documents the data entered.

Every piece of information should be traceable back to the input data that produced it. That means the analyst must provide an audit trail that records the source of each data item and when it entered the system. In addition to recording the original source, an audit trail must show how and when data is accessed or changed, and by whom. All those actions must be logged in an audit trail file and monitored carefully.

A company must have procedures for handling source documents to ensure that data is not lost before it enters the system. All source documents that originate from outside the organization should be logged when they are received. Whenever source documents pass between departments, the transfer should be recorded.

[**Data security**](javascript://) policies and procedures protect data from loss or damage, which is a vital goal in every organization. If the safeguards are not 100% effective, data recovery utilities should be able to restore lost or damaged data. Once data is entered, the company should store source documents in a safe location for some specified length of time. The company should have a [**records retention policy**](javascript://) that meets all legal requirements and business needs.

Audit trail files and reports should be stored and saved. Then, if a data file is damaged, the information can be used to reconstruct the lost data. Data security also involves protecting data from unauthorized access. System sign-on procedures should prevent unauthorized individuals from entering the system, and users should change their passwords regularly. Having several levels of access also is advisable. For example, a data entry person might be allowed to view a credit limit but not change it. Sensitive data can be encrypted, or coded, in a process called **encryption**, so only users with decoding software can read it.

Change font size

[**help**](javascript://)

[Main content](https://ng.cengage.com/static/nbreader/ui/apps/nbreader/nbreader.html#header)

**8.9**Emerging Trends

The user interface continues to evolve. The widespread use of mobile devices such as the iPad and other tablets, coupled with the ubiquity of the iPhone and other smartphones, has greatly influenced user interface design. Indeed, key user experience lessons from these devices have been reflected back into the user interface for mainstream operating systems such as Mac OS X and Microsoft Windows.

It is difficult to predict the future, but the introduction of the Apple Watch, Google’s Wear OS, Fitbit’s fitness trackers, and other wearable devices promises to further shake up user interface design principles. Advanced technology will support the evolution, but the *real* driving force will be user empowerment, which results in customer satisfaction, increased productivity, and bottom-line savings.

Change font size

[**help**](javascript://)

[Main content](https://ng.cengage.com/static/nbreader/ui/apps/nbreader/nbreader.html#header)

## 8.9.1Modular Design

In a **modular design**, individual components, called [**modules**](javascript://), are created that connect to a higher-level program or process. In a structured design, each module represents a specific process, which is shown on a DFD and documented in a process description. If an object-oriented design is being used, as described in [Chapter 6](javascript://), code modules represent classes. Modular design is explained in more detail in [Chapter 11](javascript://), which describes systems implementation.

Modules should be designed to perform a single function. Independent modules provide greater flexibility because they can be developed and tested individually and then combined or reused later in the development process. Modular design is especially important in designing large-scale systems because separate teams of analysts and programmers can work on different areas and then integrate the results.

Change font size

[**help**](javascript://)

[Main content](https://ng.cengage.com/static/nbreader/ui/apps/nbreader/nbreader.html#header)

## 8.9.2Responsive Web Design

Today’s content is viewed by users on multiple devices: computers, tablets, and smartphones. Each device has its own form factors that can limit the user experience. For example, the physical dimensions of a smartphone preclude the use of wide menus—without forcing the user to scroll horizontally too much.

Responsive web design is an emerging trend that renders web content properly, independently of the device in use. This means the developer only has to focus on essential user interface issues; how the GUI artifacts are presented on the device is handled automatically by the underling framework. Responsive web design addresses various nonfunctional attributes, including usability, performance, and maintainability.

Responsive web design relies on a number of underlying technologies, including CSS3, flexible images, and fluid grids. Page elements are expressed in relative terms (e.g., percentages), rather than absolute terms (e.g., pixels). This allows the content to “flow” properly in the user interface, irrespective of the display device.

Change font size

[**help**](javascript://)

[Main content](https://ng.cengage.com/static/nbreader/ui/apps/nbreader/nbreader.html#header)

## 8.9.3Prototyping

[**Prototyping**](javascript://) produces an early, rapidly constructed working version of the proposed information system, called a **prototype**. Prototyping, which involves a repetitive sequence of analysis, design, modeling, and testing, is a common technique that can be used to design anything from a new home to a computer network. For example, engineers use a prototype to evaluate an aircraft design before production begins, as shown in the wind tunnel testing in [Figure 8-18](javascript://).

**Figure 8-18**

Wind tunnel testing is a typical example of prototyping.



Courtesy of NASA

User input and feedback is essential at every stage of the systems development process. Prototyping allows users to examine a model that accurately represents system outputs, inputs, interfaces, and processes. Users can “test-drive” the model in a risk-free environment and either approve it or request changes. In some situations, the prototype evolves into the final version of the information system. In other cases, the prototype is intended only to validate user requirements and is discarded afterward.

Perhaps the most intense form of prototyping occurs when agile methods are used. As described in [Chapter 1](javascript://), agile methods build a system by creating a series of prototypes and constantly adjusting them to user requirements. As the agile process continues, developers revise, extend, and merge earlier versions into the final product. An agile approach emphasizes continuous feedback, and each incremental step is affected by what was learned in the prior steps.

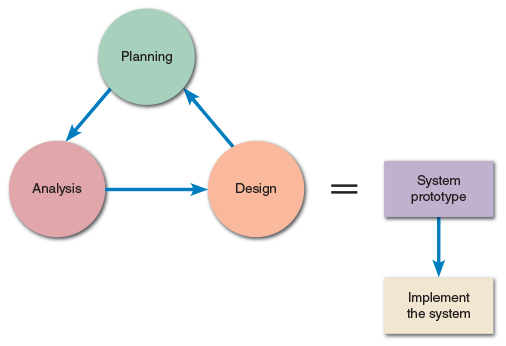
Systems analysts generally use two prototyping methods: system prototyping and design prototyping.

### System Prototyping

[**System prototyping**](javascript://) produces a full-featured, working model of the information system. A system prototype that meets all requirements is ready for implementation, as shown in [Figure 8-19](javascript://). Because the model is “on track” for implementation, it is especially important to obtain user feedback and to be sure that the prototype meets all requirements of users and management.

**Figure 8-19**

The end product of system prototyping is a working model of the information system, ready for proper implementation.



### Design Prototyping

Systems analysts also use prototyping to verify user requirements, after which the prototype is discarded and implementation continues. The approach is called [**design prototyping**](javascript://), or [**throwaway prototyping**](javascript://). In this case, the prototyping objectives are more limited but no less important. The end product of design prototyping is a user-approved model that documents and benchmarks the features of the finished system. Design prototyping makes it possible to capture user input and approval while continuing to develop the system within the framework of the SDLC. Systems analysts typically use design prototyping as they construct outputs, inputs, and user interfaces.

### Trade-Offs

Prototyping offers many benefits, including the following:

* Users and systems developers can avoid misunderstandings.
* System developers can create accurate specifications for the finished system based on the prototype.
* Managers can evaluate a working model more effectively than a paper specification.
* Systems analysts can use a prototype to develop testing and training procedures before the finished system is available.
* Prototyping reduces the risk and potential financial exposure that occur when a finished system fails to support business needs.

Although most systems analysts believe that the advantages of prototyping far outweigh any disadvantages, the following potential problems should be considered:

* The rapid pace of development can create quality problems, which are not discovered until the finished system is operational.
* Other system requirements, such as reliability and maintainability, cannot be tested adequately using a prototype.
* In very complex systems, the prototype can become unwieldy and difficult to manage.
* A client or user might want to adopt the prototype with few to no changes, mistakenly thinking that the prototype will meet their needs though it may need further customization, leading to increased maintenance costs later in the SDLC.

### A Question of Ethics

* [iStock.com](http://istock.com/" \t "_blank)/faberfoto\_itOne of the systems analysts on the project team thought that he did a good job of designing the company’s tech support webpage, but his supervisor isn’t so sure. His supervisor is concerned that the design is very similar to a page used by the company’s major competitor, and she asked him whether he had used any HTML code from that site in his design. Although the analyst didn’t copy any of the code, he did examine it in his web browser to see how they handled some design issues.

The supervisor asked the analyst to investigate webpage copyright issues and report back to her. In his research, the analyst learned that outright copying would be a copyright violation, but merely viewing other sites to get design ideas would be permissible. What is not so clear is the gray area in the middle. The analyst asked you, as a friend, for your opinion on this question: Even if no actual copying is involved, are there ethical constraints on how far you should go in using the creative work of others? How would you answer?

Change font size

[**help**](javascript://)

[Main content](https://ng.cengage.com/static/nbreader/ui/apps/nbreader/nbreader.html#header)

**8.10**Summary

The chapter began with a discussion of user interface design and HCI concepts. A GUI uses visual objects and techniques that allow users to communicate effectively with the system. User-centered design principles include understanding the business, maximizing graphic effectiveness, thinking like a user, using models and prototypes, focusing on usability, inviting feedback, and documenting everything.

When designing the user interface, it should be transparent; create an interface that is easy to learn and use; enhance user productivity; make it easy to obtain help or correct errors; minimize input data problems; provide feedback; create an attractive layout and design; and use familiar terms and images. Control features, such as menu bars, toolbars, drop-down list boxes, dialog boxes, toggle buttons, list boxes, option buttons, check boxes, and command buttons can also be added. Controls are placed on a main switchboard, which is like a graphical version of a main menu.

The discussion of input design began with a description of source documents and the various zones in a document, including the heading zone, the control zone, the instruction zone, the body zone, the totals zone, and the authorization zone. The discussion of data entry screen design explained the use of input masks and validation rules to reduce data errors. Input masks are like templates that only permit certain combinations of characters, and data validation rules can provide checks to ensure that inappropriate data is prevented from entering the system. These checks can include data sequence, existence, range and limit, reasonableness, and validity, among others.

The chapter described various types of printed reports, including detail, exception, and summary reports. The features and sections of reports, including control fields, control breaks, report headers and footers, page headers and footers, and group headers and footers were explained. Other types of output, such as web-based information delivery, audio output, instant messaging, podcasts, email, and other specialized forms of output were also discussed.

Batch and online input methods were also described, as were input media and procedures, and input volume. Input methods include data capture and data entry. Data capture, which may be automated, involves identifying and recording source data. Data entry involves converting source data into a computer-readable form and entering it into the system. New technology offers optical and voice recognition systems, biological feedback devices, motion sensors, and a variety of graphical input devices.

Security and control were discussed. Output control includes physical protection of data and reports and control of unauthorized ports or devices that can extract data from the system. Input controls include audit trails, encryption, password security, data security, and the creation of access levels to limit persons authorized to view or use data.

Finally, the emerging trends of modular design, responsive web design, and prototyping were discussed.

Change font size

[**help**](javascript://)

[Main content](https://ng.cengage.com/static/nbreader/ui/apps/nbreader/nbreader.html#header)

# Chapter Review

## **Key Terms**

* [**audit trails**](javascript://)
* [**authorization zone**](javascript://)
* [**automated fax**](javascript://)
* [**batch**](javascript://)
* [**Batch controls**](javascript://)
* [**batch input**](javascript://)
* [**blogs**](javascript://)
* [**calendar control**](javascript://)
* [**character-based reports**](javascript://)
* [**check boxes**](javascript://)
* [**combination check**](javascript://)
* [**command button**](javascript://)
* [**context-sensitive**](javascript://)
* [**control break**](javascript://)
* [**control break report**](javascript://)
* [**control field order**](javascript://)
* [**data security**](javascript://)
* [**data type check**](javascript://)
* [**data validation rule**](javascript://)
* [**default value**](javascript://)
* [**design prototyping**](javascript://)
* [**detail report**](javascript://)
* [**dialog box**](javascript://)
* [**diskless workstation**](javascript://)
* [**electronic health records (EHRs)**](javascript://)
* **encryption**
* [**exception report**](javascript://)
* [**existence check**](javascript://)
* [**faxback**](javascript://)
* [**form filling**](javascript://)
* [**form layout**](javascript://)
* [**garbage in, garbage out (GIGO)**](javascript://)
* [**graphical user interface (GUI)**](javascript://)
* [**hash totals**](javascript://)
* [**human-computer interaction (HCI)**](javascript://)
* [**input control**](javascript://)
* [**input mask**](javascript://)
* [**limit check**](javascript://)
* [**list boxes**](javascript://)
* [**menu bars**](javascript://)
* [**mock-up**](javascript://)
* **modular design**
* [**modules**](javascript://)
* [**natural language**](javascript://)
* [**online data entry**](javascript://)
* [**option button**](javascript://)
* [**output control**](javascript://)
* [**output security**](javascript://)
* [**page footer**](javascript://)
* [**page header**](javascript://)
* [**podcast**](javascript://)
* **prototype**
* [**prototyping**](javascript://)
* [**radio button**](javascript://)
* [**radio frequency identification(RFID) tags**](javascript://)
* [**range check**](javascript://)
* [**reasonableness check**](javascript://)
* [**records retention policy**](javascript://)
* [**report footer**](javascript://)
* [**report header**](javascript://)
* [**scroll bar**](javascript://)
* [**sequence check**](javascript://)
* [**source data automation**](javascript://)
* [**source document**](javascript://)
* **storyboard**
* [**summary report**](javascript://)
* [**switchboard**](javascript://)
* [**system prototyping**](javascript://)
* [**throwaway prototyping**](javascript://)
* [**toggle button**](javascript://)
* [**toolbar**](javascript://)
* [**totals zone**](javascript://)
* [**transparent interface**](javascript://)
* [**turnaround documents**](javascript://)
* [**usability**](javascript://)
* [**usability metrics**](javascript://)
* [**user-centered**](javascript://)
* [**user interface (UI)**](javascript://)
* [**validity check**](javascript://)
* [**webcast**](javascript://)

Change font size

[**help**](javascript://)

[Main content](https://ng.cengage.com/static/nbreader/ui/apps/nbreader/nbreader.html#header)