Chapter 8: User Interface Design

# Teaching Tips and Strategies *(from Alan Dennis)*

This chapter probably presents the design of user interfaces somewhat differently than other textbooks you might have used. The first part of the chapter focuses on the overall design process, while the last half focuses on the detailed elements of the design. We have chosen to present interface design in this way, because in our opinion, it is critical to design the input, output and navigation elements as one integrated, coherent whole. We approach this using the traditional top-down approach commonly used in industry, but less commonly seen in the classroom. I usually spend two classes on this chapter. In this chapter, we focus on the *user* interfaces, but it is important to remember that system interfaces must also be designed (e.g., between two or more systems that exchange data).

I start by presenting the six fundamental principles for user interface design. I walk through them and remind students that these should be used a priori in the design of the system, as well as post-hoc after the interface design is finished as checklist to go back over the design. Many of these principles should be common sense, but I've found that before I explicitly mentioned them (and encouraged students to use them as an evaluation checklist), they were violated as often as they were followed. I sometimes present examples of screens that seem reasonable at first glance, but are really quite bad designs, and then ask students to assess them using the six principles. This helps drive home the fact that user interface design is not simple.

The design process is fairly straightforward, but once again, before I started presenting the process as a series of five steps, students often developed poor designs. The idea of developing a design and thinking about how the users will actually use it in the use scenarios before sitting down and writing programs is an extremely powerful concept that has greatly improved the quality of the designs produced by my students. While they used to design them on paper, it just never was the same as thinking about them in action. The CD Selections case is particularly useful in showing how the use scenarios can be used to improve the design.

I usually try to spend one class session on the second half of the chapter, but sometimes it takes more like a class and a half. There are lots of details here that can get tedious and dull, but nonetheless are important. We have tried to put most of the details in the tables so if you prefer you can focus on the principles and concepts, and leave the details to the students to read and use as they design the interface for the their project.

Some of the Your Turn boxes (8.7, 8.8, 8.8) can be used to stimulate class discussion by using them as assignments that students do in class or bring to class for discussion and critique by other students using the principles in each section. This may change the need to cover many minor details into a discussion of how to apply the details in the context of a user interface design.

# War Stories *(from Alan Dennis)*

**User Interface Evaluation**

Getting widespread evaluation of a user interface can be difficult. One of the consulting projects I worked on brought a set of twenty users into an e-JAD meeting room. The analysts demonstrated the initial interface design prototype (a series of mock-up screens). After each screen, the analysts paused while users typed their comments into the e-JAD system. Since the e-JAD system was anonymous, the users felt more comfortable complaining. And since everyone could type at the same time, no small set of users dominated the evaluation process. Once the users were finished, the analysts continued with the next screen, while another group of analysts read the users' comments to make sure they understood them. If they did not understand, they made sure to ask questions at the next break.

**Navigation Design in SAS**

One of the programs I love to hate is SAS, the statistical analysis system I use in my research. SAS has one of the worst user interfaces of any widely used commercial system I have used. The biggest problem is that SAS does not follow many of the Windows standards. For example, in SAS Control/F means add a footnote, not Find as it does in virtually every other Windows program (now how many times do you add a footnote to a SAS printout and why in the world would you want a hot-key for it, let alone a very commonly used one that means something else everywhere else). SAS does, of course, have a find command, but it has no hot key, meaning you have to use two mouse clicks to bring up the find menu. Once you do a find (or a replace), the find window disappears so if you want to find (or replace) the next occurrence, you have you again use two mouse clicks to bring up the box. And how many times do you want to repeat a find or replace?

# Answer to Your Turn 8-1: Web Page Critique

Student answers will vary, however, they should address each of the six principles of user interface design:

* Layout
* Content Awareness
* Aesthetics
* User Level
* Consistency
* Minimizing User Effort

# Answer to Your Turn 8-2: Personas and USD for the Web

Student answers will vary depending upon the university web site. However, student responses should include an outline of the steps a user would perform to accomplish a task; finding a course description, registering for a course, etc.

# Answer to Your Turn 8-3: Personas and USD for an ATM

Student answers will vary depending upon their choice of activity. Examples of use scenarios for an ATM are as follows:

Use Scenario: Authentication

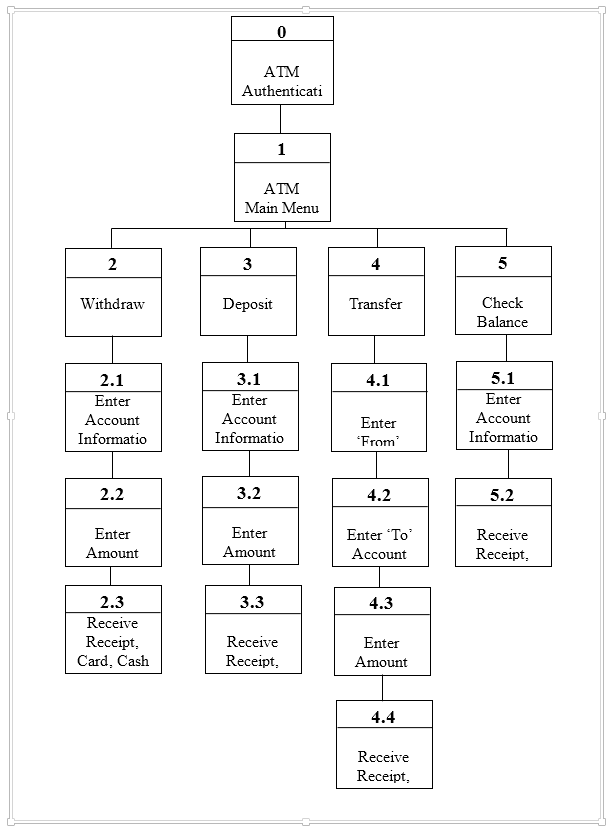
1. User inserts card into card reader
2. Upon prompt, user taps numeric keys for PIN input
3. User is granted access to account(s), or user is denied access.

Use Scenario: Withdrawing Cash from ATM

1. User is granted access to account
2. User will choose withdraw from a menu of available transaction types
3. User will choose account from which to withdraw
4. User will key in monetary amount to withdraw
5. User will remove card, money, and receipt

# Answer to Your Turn 8-4: Interface Structure Design

The following is an example, with DFD notations omitted.



# Answer to Your Turn 8-5: Interface Standards Development

Student answers will vary. The following is an example of an Interface Standard for an ATM.

Interface Standards – ATM

Interface Metaphor: Banking Account Access

Interface Objects: Accounts

Customer

Monetary Amounts

Interface actions: Screen – present options

Customer choice – withdraw, deposit, transfer, check balance

Verification of account status

Interface icons: Bank logo

Account icon – ledger sheet

Customer icon – person

Monetary icon – dollar bill

# Answer to Your Turn 8-6: Prototyping and Evaluation

Student answers will vary, however, as most ATMs do not employ a browser interface, the HTML prototype would not apply. As the system will primarily be used by customers, either a walk-through evaluation or an interactive evaluation would provide the most feedback from a user standpoint regarding the functionality of the new design.

# Answer to Your Turn 8-7: Design a Navigation System

Student answers will vary depending upon their familiarity with drawing software. Typically answers will include a menu based system with associated forms, one each for customer, product, and order with the ability to change, delete or search for a record.

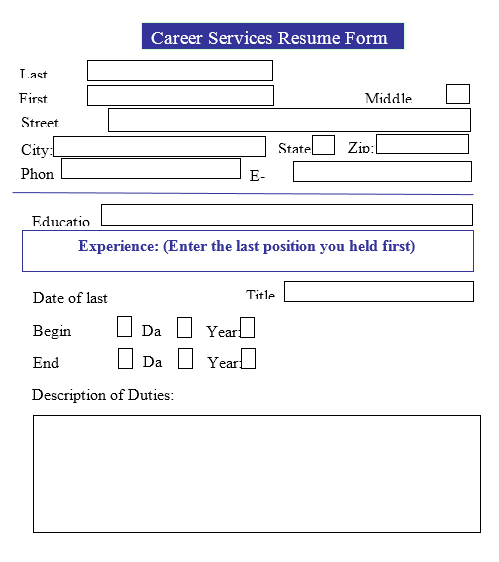


# Answer to Your Turn 8-8: Career Services

Student answers will vary. Typically answers will include online processing to ensure that each resume will be available to recruiters as soon as possible. Input format could be an online form that is basically a resume template, allowing the student to fill in text boxes for name, education, experience, etc. To minimize keystrokes, all dates might have a drop down list containing the numbers 1-31 for the day fields, a drop down list with 1-12 for the month fields, and a drop down list of a range of years for the year fields.

# Answer to Your Turn 8-9: Career Services

Student answers will vary. An example of an input form follows:



Validation checks should be identified for month/day combinations as well as recognizing leap years.

# Answer to Your Turn 8-10: Finding Bias

Student answers will vary depending upon the publications and graphs found.

# Answer to Concepts in Action 8-A: Interface Design Prototypes for a DSS Application

1. The team probably decided on a language prototype because it allows the user to see detailed examples of the interfaces to the system. The user does not have to use his imagination to determine what the interfaces will look like as they might with a story board or an html approach.
2. The trade-offs to using a language prototype is that it can be expensive and may take longer to develop.

# Answer to Concepts in Action 8-B: Public Safety Depends on a Good User Interface

Student answers will vary. However, steps should include:

* Involving the user in the decision-making process for selecting the system.
* Testing the new system under the conditions in which it will be used.
* Offering in-depth training for the end users.
* Offering training that simulates real-time conditions.

# Answer to Concepts in Action 8-C: Selecting the Wrong Students

Since the faculty assumed that the final report was in a rank order, they may have assumed that the order in which the students’ names appeared for evaluation was in rank order of grade point average or standardized test scores. It would seem that the faculty was not familiar with the system enough to use it accurately, thus errors may have been made in the input of their evaluations.

# Answer to Concepts in Action 8-D: Cutting Paper to Save Money

Reports that are most suited to an electronic format include: ad hoc reports, customized reports, reports that the user might interact with (search, delete, correct, etc.). Reports that are most suited to a paper format are those for which a permanent record are needed, reports that need signatures and reports that need to be portable (used in meetings).

# Solutions to End of Chapter Questions

1. *Explain three important user interface design principles.*

The authors list six principles of user interface design:

1. Layout - the interface should be a series of areas on the screen that are used consistently for different purposes.
2. Content Awareness - the user is always aware of where they are in the system and what information is being displayed.
3. Aesthetics - interfaces should look inviting and should be easy to use.
4. User Level - experiences users prefer ease of use, while inexperienced users prefer ease of learning.
5. Consistency - users can predict what will happen before a function is performed.
6. Minimize Effort - interface should be simple to use.
7. *What are three fundamental parts of most user interfaces?*
8. Navigation mechanism - the way the user gives instructions to the system and tells it what to do.
9. Input mechanism - the way in which the system captures information
10. Output mechanism - the way the system provides information to the user or to other systems.
11. *Why is content awareness important?*

Content awareness means that the interface makes the user aware of the information delivered through the interface with the least amount of user effort. This is important because if the user is constantly aware of where he is and what he is seeing, he will find the system much easier to use and his satisfaction will be high.

1. *What is white space, and why is it important?*

White space refers to areas on an interface that are intentionally left blank. The more white space on an interface, the less dense the information content. Designers need to try and strike a balance between information content and white space. Some white space is necessary to help the users find things on the interface. Generally, more experienced users need less white space than novice users.

1. *Under what circumstances should densities be low? High?*

Low densities are preferred by infrequent or novice users of an interface. These users will be unfamiliar with the interface and will be helped by having a balance of information and white space on the interface. High densities can be acceptable to experienced users of the interface, because they are highly familiar with the information on the interface and do not need as much white space to help them find what they are looking for.

1. *How can a system be designed to be used by both experienced and first time users?*

Experienced users prefer systems that focus on ease of use, while novice users prefer systems that are easy to learn. These two goals are not necessarily mutually exclusive. Generally, systems should be set up so that the commonly used functions can be accessed quickly, pleasing the experienced users. To assist the novice users, guidance should be readily available, perhaps through the “show me” functions that demonstrate menus and buttons.

1. *Why is consistency in design important? Why can too much consistency cause problems?*

Consistency means that all parts of the same system work in the same way. This enables the users to predict what will happen because a function in one part of the system works the same way in other parts of the system. Users will be confident as they work with different parts of the system if they can predict the behavior of functions throughout the system. The problem with too much consistency is that sometimes the users don’t differentiate forms or reports that look very similar to each other, and inadvertently use the wrong one. So, in these cases, there should be enough unique characteristics to distinguish each form and report from the others.

1. *How can different parts of the interface be consistent?*

The navigation controls can be consistent, using the same icon or command to trigger an action throughout the system. Terminology can be consistent throughout the interface. The content portion of the screen that contains forms and reports should also present consistently designed reports and forms. Messages and information in the status area should be specified consistently throughout the system.

1. *Describe the basic process of user interface design.*

First, identify ‘use cases’ that describe commonly used patterns of actions that users will perform. These use cases will be valuable in ensuring that the interface permits the users to enact these use cases quickly and smoothly. Next, develop the interface structure diagram, defining the basic structure of the interface (screens, forms, and reports) and how the interface components connect. Third, develop interface standards, the basic design elements that will be used throughout the interface. Fourth, create prototypes of the various interface components (navigation controls, input screens, output screens, forms, and reports). Finally, evaluate the prototypes and make changes as needed.

1. *What are personas and use case scenarios, and why are they important?*

Use cases describe commonly used patterns of actions that users will perform. Use cases describe how users will interact with the system. Use cases are developed for the most common ways of working through the system. These use cases will be valuable in ensuring that the interface permits the users to enact these use cases quickly and smoothly. Identified user groups can be represented with *personas*. The most common usage patterns of the system are depicted with *use scenarios* so that the interface can enable users to quickly and smoothly perform these scenarios.

1. *What is a site map and an interface structure diagram (ISD), and why are they used?*

A site map helps analysts clarify how all the information on the site fits together and helps establish the hierarchy of information on the site. An interface structure diagram shows all the screens, forms, and reports in the system, how they are related, and how the user moves from one to another. The diagram helps depict the basic components of the interface and how they work together to provide users the needed functionality. The structure of the interface depicted in the ISD can be examined using the use cases to see how well the use cases can be performed. This is an important early step in developing simple paths through the most common activities performed in the system.

1. *Why are interface standards important?*

Interface standards are the basic design elements on which interfaces in the system are based. These standards help ensure consistency throughout the system.

1. *Explain the purpose and contents of interface metaphors, interface objects, and interface actions, interface icons, and interface templates.*
2. The interface metaphor provides a concept from the real world that helps the user understand the system and how it works. If the user understands the metaphor being used, he will probably be able to predict where to find things and how things will work even without actually using the system.
3. Interface objects are the fundamental building blocks of the system. Object names should be based on the most understandable terms.
4. Interface actions specify the navigation and command language style and the grammar of the system. Action terminology is also defined.
5. Interface icons are pictures that are used to represent objects and actions in the system, often shortcuts, that are available throughout the system.
6. The interface template defines the general appearance of all screens in the information system and all forms and reports that are used. The template consolidates all the other major interface design elements - metaphors, objects, actions, and icons.
7. *Why do we prototype the user interface design?*

An interface design prototype is a mock-up or a simulation of a computer screen, form, or report. Prototyping helps the users and programmers understand how the system will perform. Prototypes can be very useful in helping the users conceptualize how they will actually work with the system, and prototypes can help identify problems or misconceptions in the interface before it is actually implemented.

1. *Compare and contrast the three types of interface design prototypes.*

Storyboards are really just pictures or drawings of the interface and how the system flows from one interface to another. HTML prototypes are Web pages that show the fundamental parts of the system. Users can interact with the system by clicking buttons and entering data, moving from page to page to simulate navigating through the system. Language prototypes create models of the interface in the actual language that will be used to implement the system. These will show the user exactly what the interface will look like, which is not possible with the other two methods.

1. *Why is it important to perform an interface evaluation before the system is built?*

An interface assessment is important before the system is built because we need to do as much as we can to improve the interface design prior to implementation. Ideally, interface evaluation should be performed while the system is being designed—before it is built—so that any major design problems can be identified and corrected before the time and cost of programming have been spent on a weak design.

1. *Compare and contrast the four types of interface evaluation.*

These techniques vary in terms of the degree of formality and the amount of user involvement. Heuristic evaluation involves assessing the interface based on a checklist of design principles. This assessment is usually performed by team members, who independently assess the interface and then compare their assessments. Weaknesses that are common in all the evaluations then point to areas that need modification. Users are not involved in this process. In a walk-through evaluation, the users see the interface at a meeting presentation, and they are “walked-through” the parts of the interface. The interactive evaluation can be used when the prototype as been created as an HTML or language prototype. The users can actually interact with the interface as if they were using the system, and can give direct comments and feedback based on their experience. Problems or areas of confusion can be noted and corrected by the team. Formal usability testing has the users interacting with the interface without guidance from the project team. Every move made by the user is recorded and then analyzed later in order to improve the interface.

1. *Under what conditions is heuristic evaluation justified?*

Heuristic evaluation is probably justified in situations where the interface is well understood. When there is little uncertainly about how the interface should function, then it is probably sufficient to just assess it internally by comparison to a checklist of design principles. It would be dangerous to use this technique (which does not involve users) if there was uncertainty about what should appear in the interface or how it should function.

1. *What type of interface evaluation did you perform in the “Your Turn 8-1”?*

This is an example of heuristic evaluation, since the interface is being compared to a set of design principles.

1. *Describe three basic principles of navigation design.*
2. Prevent Mistakes - this principle is directed toward developing the navigation controls to help the user avoid making mistakes.
3. Simplify Recovery from Mistakes - this principle recognizes that mistakes will happen, and so is directed toward making it as easy as possible to recover from those mistakes.
4. Use Consistent Grammar Order - This principle states that the order of commands should be consistent throughout the system.
5. *How can you prevent mistakes?*

While it is impossible to completely prevent mistakes, there are some things that will help the user avoid mistakes. First, make sure all commands and actions are clearly labeled. Limit the number of choices that are presented to the user at one time to help reduce confusion. Never display a command or action that cannot be used. Also, give users a chance to confirm potentially destructive actions (such as deleting a record).

1. *Explain the differences between object-action order and action-object order.*

Commands given to the system usually follow a sequence of ‘specify the object, then specify the action’ or ‘specify the action, then specify the object.’ This is referred to as the grammar order of the commands. The designers should select the grammar order desired for the system and use it consistently.

1. *Describe four types of navigation controls.*

The text does not specifically list four types of navigation controls. The most common four are described here. Command languages are not common in modern systems but do exist. Menus are the most common. Direct manipulation and natural language voice recognition are becoming very common and improved greatly.

1. Languages - most often this navigation control refers to a command language, or a set of specials instructions that are used to instruct the system. In order to perform a task the user must know the correct command to give the system. Natural language interfaces free the user to give instructions in everyday terminology, but these types of systems are not common.
2. Menus - this navigation control presents the user with a list of options that can be performed as needed. Menu structures present the user with an organized set of commands to apply
3. Direct manipulation - this type of navigation control involves working directly with interface objects, such as dragging a file from one location to another.
4. Voice recognition - this navigation control involves giving instructions to the computer verbally. Some of the systems only recognize certain commands, while others recognize more natural speech. Progress is being made in this technology, but it is not yet common in systems.
5. *Why are menus the most commonly used navigation control?*

Menus are the most commonly used navigation control because they are much easier to learn than a language, and they are very simple to work with, enhancing the ease of use of the system.

1. *Compare and contrast four types of menus.*

Some of the more common types of menus include *menu bars*, *drop-down menus*, *popup menus*, *tab menus*, *icon tool bars*, and *image maps.* The menu bar is usually the main menu of the system. It consists of a list of commands across the top of the screen that is always displayed. The commands on the menu bar represent the main objects and/or actions of the system, and lead to other menus. Drop-down menus appear immediately below another menu. A series of commands are listed, and these lead to direct actions or other menus. The drop-down menu disappears after one use. Pop-up menus appear to ‘float’ on the screen, usually triggered by a right-click on the mouse. A series of commands that pertain to the work the users was doing are listed. Pop-up menus are often used to present an experienced user with shortcuts to common commands. Pop-up menus disappear after one use. A tab menu is a multi-page menu, each page represented by a tab on the menu. Each tab represents a set of related actions or settings. The tab menu will remain on the screen until the user closes it.

1. *Under what circumstances would you use a drop-down menu versus a tab menu?*

A drop-down menu is commonly used as the second-level menu, triggered when one of the main menu options is selected. The drop-down menu lists another set of more specific commands that will either lead directly to an action or to another, more detailed menu. The tab menu is chosen whenever the user needs to make multiple choices (such as specifying several settings) or perform several related commands. The tab menu stays open until the user has completed making the choices and closes the menu. Use a tab menu whenever the user needs to do several related tasks at one time.

1. *Describe five types of messages.*

Error messages are displayed when the user has done something that is not permitted or cannot be carried out. An error message should inform the user why the attempted action is illegal or incorrect. Confirmation messages are displayed whenever the user has entered a command that has major significance and may be destructive (such as shutting down the system or deleting a record.). The confirmation message is used to force the user to verify that the action is the correct one. Acknowledgment messages signify that an action or task is complete. These messages can be used to ensure that the user knows what the system is doing, but they can become very annoying if encountered frequently. Delay messages indicate that the system is performing a task and that the user should wait until the task is completed. These messages keep the user informed about the system status, and can be very helpful, especially to novice users who may not appreciate the time certain tasks require. Help messages provide the user with additional information, and are an important means of giving users instructions and guidance when needed. Even experienced users will need access to help for rarely used system functions.

1. *What are the key factors in designing an error message?*

An error message should first identify the error. Some additional explanation of the problem is also usually provided. Then, the message should inform the user how to correct the problem. Finally, a button for user response is usually included that clears the message off the screen and enables the user to take the corrective action.

1. *What is context-sensitive help? Does your word processor have context-sensitive help?*

Context-sensitive help provides information that is dependent on what the user was doing when help was requested.

1. *Explain three principles in the design of inputs.*

The most significant input design principle is to capture data as close to its point of origin as possible. By electronically collecting the data at its point of origin, time delays are minimized and errors can be reduced. A second important input design principle is to minimize user keystrokes. Use source data automation techniques whenever possible. Only ask the user to enter new data into the system; use reference tables and lookups whenever possible. When the inputs have known values, use default values check boxes, radio buttons, or drop-down lists. Finally, use the appropriate mode of processing (online versus batch) for the application. Batch applications are generally simpler than online applications, but have the disadvantage of not updating the databases or files immediately. Online applications are more complex than batch, but are used when it is necessary to have immediate update of the databases or files.

1. *Compare and contrast batch processing and online processing. Describe one application that would use batch processing and one that would use online processing.*

There are two general approaches for entering inputs into a computer system: online processing and batch processing. With *online processing* (sometimes called *online transaction processing*), each input transaction is entered and processed at the time the event occurs. With *batch processing*, all the input transactions collected over some period are gathered together and processed at one time in a batch. An airline reservation system is a classic example of an online system, since the flight reservation is immediately reflected in the system database. Payroll systems are commonly batch applications, with payroll transactions accumulated over the pay period and processed as a batch at one time.

1. *Why is capturing data at the source important?*

Capturing data at the source has three advantages. First, it can reduce costs because work does not have to be duplicated. Second, it reduces delays in processing. Third, it reduces the likelihood of error.

1. *Describe four devices that can be used for source data automation.*

Bar code readers scan bar codes found on products to enter data directly into the system. Optical character readers can read and enter printed numbers and text. Magnetic stripe readers enter information from a stripe of magnetic material. Smart cards contain microprocessors, memory chips, and batteries to maintain information which then can be read by smart card readers.

1. *Describe five types of inputs.*

When it is necessary to enter text from the keyboard, a *text box* control is used. . Up-down numeric control is used when the user must enter numeric data. Check boxes are used whenever the user can choose one or more items from a known list. Radio buttons are used when the user needs to select one choice from a known set of options.

1. *Compare and contrast check boxes and radio buttons. When would you use one versus the other?*

Physically, check boxes are usually represented as small squares, and radio buttons are small circles. Operationally, they are used very differently. Check boxes are used when the user can select one or more choices from a list of options. Radio buttons are mutually exclusive. Only one button can be chosen at a time. Selecting one radio button removes the selection from any button previously selected. Use radio buttons when you want to force the user to make one choice. Use check boxes when the user can select multiple items from the list.

1. *Compare and contrast on-screen list boxes and drop-down list boxes. When would you use one versus the other?*

On-screen list boxes present the user with a list of choices that are always displayed. A drop-down list box displays the list of choices as needed. Generally, there is not enough screen space to use on-screen list boxes unless the list is quite short. Therefore, the drop-down list box is used to display longer lists temporarily and then disappear from the screen after the choice is made. The amount of available screen space dictates which type of list box will be used.

1. *Why is input validation important?*

Input is validated in order to try and reduce the amount of erroneous data that is entered into the system. Clearly, the quality of the information that comes out of a system is dependent on the quality of the input data. Therefore, we must do as much as is reasonable to assure high quality data is input in the system. The various techniques of data validation help us do that.

1. *Describe five types of input validation methods.*

Completeness checks are performed to verify that all required data items have been entered. However, when specific data is required, a completeness check will ensure that something is entered in every required field. Format checks ensures that data are of the right type (e.g., numeric) and in the right format (e.g., month, day, year). Range checks are commonly used when a numeric item falls within some expected range of values. A check digit check is used to validate numeric code fields. In these situations, an algorithm establishes a check digit for each occurrence of the numeric code. Whenever a numeric code is re-entered into the system its check digit is recalculated. If the calculated check digit does not match the expected check digit, there has probably been a data entry error in the code, and it needs to be re-entered. Consistency checks are performed when there is a relationship between field values that is known and can be checked. Database checks are used to compare an entry against a value stored in a file or database to ensure it is a valid value.

1. *Explain three principles in the design of outputs.*

First, it is important to understand how the report will be used. It is not enough just to know what data should appear on the report. The report designer needs to know how the user will utilize the report; what sequence or sorting arrangement is needed, what subtotals are needed, when the information is needed, etc. Second, the report design needs to manage the information load presented in the report. A report that dumps in ‘everything but the kitchen sink’ will probably not be useful to the recipient. Have the user specify the question(s) he wants to answer by using the report, and then provide just the information needed to answer those questions. Third, avoid presenting the information in a biased way. Bias can be unintentional, so carefully assess choices made on information sorting or structuring graphical outputs.

1. *Describe five types of outputs.*

Detail reports are reports that lists detailed information about all the items requested. Summary reports list summarized information about a large number of items. Exception reports list information only about items that meet some predefined criterion. (This definition was not included in Figure 8-22). Turnaround documents provide information about some system output (e.g., a bill), but also include a section that will re-enter the system as an input (e.g., a payment coupon). A graph is a depiction of numerical relationships and help users compare two or more items or understand how one has changed over time.

1. *When would you use electronic reports rather than paper reports and vice versa?*

Paper reports have the advantage of being permanent, easy to use, and portable (if they are small). Paper reports do not require the presence of a computer in order to be used. A report should be printed on paper if its content is fairly static and if it needs to be taken from place to place to be used, and computers are not readily available. Electronic reports store the reports on servers so that they can be readily viewed from any computer. Electronic reports are inexpensive that often many variations of the reports are created. Users can refer to the reports online or print them locally as needed. It is generally advantageous and less costly for users to print reports locally as needed rather than printing all reports centrally.

1. *What do you think are three common mistakes that novice analysts make in interface design?*

* Failing to focus on the most common paths through the interface
* Making the interface too crowded
* Failing to think about whether the primary users of the system are casual, occasional users or frequent, experienced users
* Being inconsistent from one place in the interface to another in terms of standard design features and terminology

1. *How would you improve the form in Figure 8-4?*

A user who has not seen this form before and does not know how it is used will find it difficult to make suggestions for improvement. The form is very dense, however, and so it might be useful to segment it into two pages that are logical subsets of the form content. Each page could be much less dense and therefore easier to use.

1. *Discuss three issues unique to developing touch screen user interfaces.*

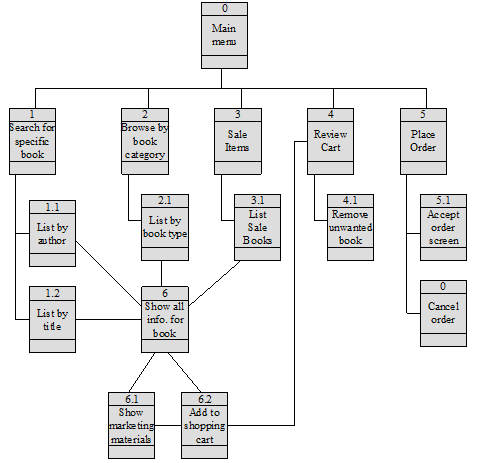
* Place navigation elements at the bottom of the screen with results at the top so that fingers do not obscure the content on the screen.
* Size the screen elements correctly for “fat fingers”
* Use standardized gesture interactions to enhance user ease of learning and ease of use.

# Solutions to End of Chapter Exercises

1. *Develop two personas and two use scenarios for a web site that sells some retail products (e.g., books, music, and clothes).*

|  |
| --- |
| Use scenario 1: Shopper looks for a specific item |
| 1. User searches for specific item 2. User looks at price and description to ensure the item is the one wanted 3. User places the item in shopping cart 4. User will want to complete the purchase by checking out; or will go on to look for other specific items |
| Use scenario 2: Shopper looks for an item on sale |
| 1. User goes to the sale area on the web site 2. User looks browses through sale items, reading descriptions of many 3. User may place item(s) in shopping cart or may just leave the sale area of the site 4. User will want to complete the purchase by checking out; or will go on to look for other items |

1. *Draw a site map and an ISD for a web site that sells some retail products (e.g., books, music, clothes).*



1. *Describe the primary components of the interface standards for a web site that sells some retail products (metaphors, objects, actions, icons, and templates).*

Metaphor – store with shopping cart

Objects – shopping cart; major classes of goods that are sold such as books

Actions – browse, select, purchase, modify, and check out

Icons – small image of shopping cart used for the shopping cart

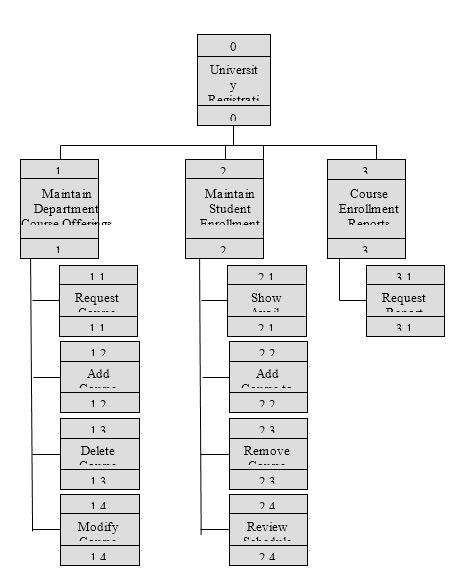
Template – navigation area and status area at top of screen; shopping cart in upper right corner; body of screen is the work area.

1. *Develop two use scenarios for the DFD in Exercise C in chapter 5.*

|  |
| --- |
| Use Scenario: Maintain department course offerings |
| 1. User requests course offering list. (1.1) 2. User obtains new course information and creates new course record. (1.2) 3. User obtains course number for course to be deleted. (1.3) 4. User obtains course number for course to be modified. (1.4) |

|  |
| --- |
| Use Scenario: Maintain student enrollments |
| 1. Student requests list of available courses. List of available courses is generated. (2.1) 2. Student adds course to current schedule. Fee payment status is checked and total hours enrolled is checked. If ok, course is added to student schedule. (2.2) 3. Student removes course from schedule. (2.3) 4. Student reviews current scheduled courses. (2.4) |

1. *Draw an ISD for the DFD in Exercise F in Chapter 4.*



1. *Develop the interface standards (omitting the interface template) for the DFD in Exercise F in Chapter 4.*

Interface Standards - University Registration System

Interface metaphor: Any university system requirement maintenance of student records, course offerings

Interface objects: Students - anyone who wants to register for a course

Staff - anyone who wants to add/delete/modify a course

* anyone who wants to view course report

Interface actions: Find - displays existing student schedule

Find – displays current course listing

Retrieve - displays current course availability

Schedule - confirms course registration

1. *Develop two use scenarios for the DFD in Exercise J in Chapter 4.*

|  |
| --- |
| Use Scenario: New customer rents video |
| 1. New customer present video to rent 2. User obtains customer information and enters in system. AVS card is issued to customer (2.1) 3. Video rental is processed (3.1) |

|  |
| --- |
| Use Scenario: Customer rents video |
| 1. Customer presents video to rent and AVS card 2. User enters customer ID from AVS card in system. Customer has unpaid fine. User collects fine from customer and enters payment in system (2.2) 3. User checks to be sure all overdue videos rented to customer are returned (3.1) 4. User processes video rental (3.1) |

1. *Develop the interface standards (omitting the interface template) for the DFD in Exercise J in Chapter 4.*

Interface Standards - A Video Store System

Interface metaphor: Video Rental Store

Interface objects: Customers - people who rent videos

Videos - videos available to rent

Rented video - video rented to a customer

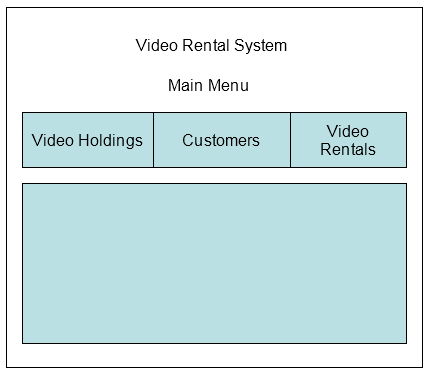
Overdue video - video that was not returned within its rental period

Fine - charge assessed for late, damaged, or lost videos

Interface actions: Rent - process a video rental

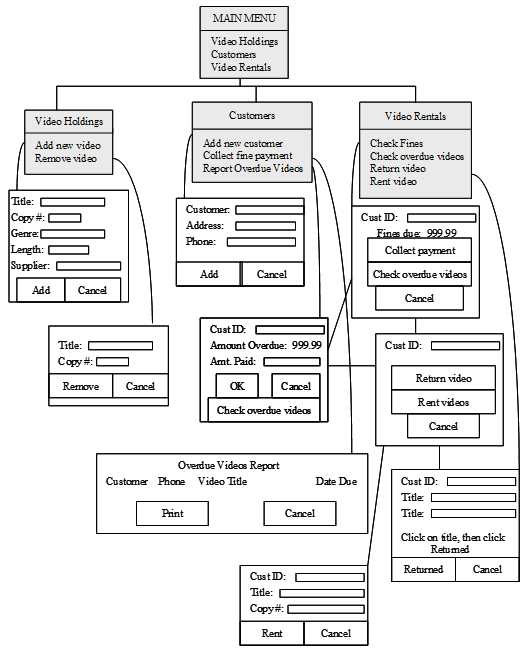
Interface icon: Use AVS logo on screen; buttons should look like VCR tapes

1. *Design an interface template for Exercise J in Chapter 4.*

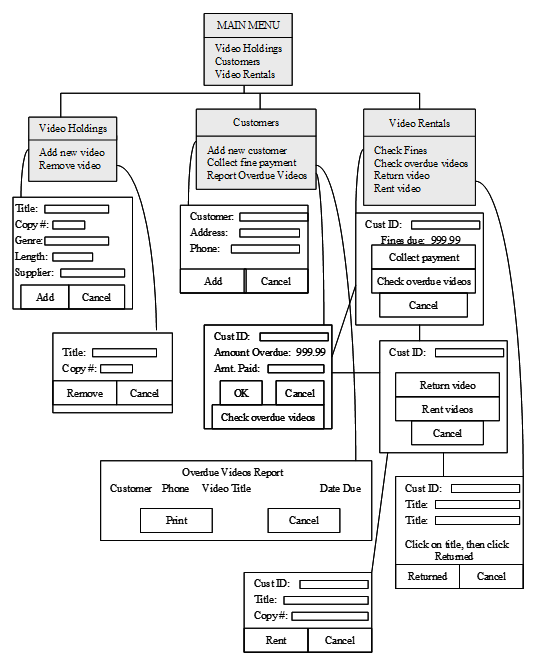


The above template shows the title area at the top of the screen. Navigation is done through the menu bar under the title. Work area is the main body of the screen.

1. *Draw an ISD for the DFD in Exercise J in chapter 4.*



1. *Design a storyboard for Exercise J in Chapter 4.*



1. *Develop an HTML prototype for Exercise E in this chapter.*

Student answers will vary depending upon their ideas of webpage design and their expertise in developing web pages.

1. *Develop an HTML prototype for Exercise F in this chapter.*

Student answers will vary depending upon their ideas of webpage design and their expertise in developing web pages.

1. *Develop the interface standards (omitting the interface template) for the DFD in Exercise L in Chapter 4.*

Interface Standards – University Library System

Interface metaphor: Library lending System – checking out and returning books.

Interface objects: Books – titles held by library

Borrowers – persons who borrow books from library

Borrowed books – title checked by borrower

Fine – charge assessed for book not returned within lending period

Special fine – charge assessed for damaging or losing book

Interface actions: Screen – verify that borrower has no outstanding fines before

checking out new books

Check out – borrower takes book for the lending period

Mark overdue – record expiration of lending term for book

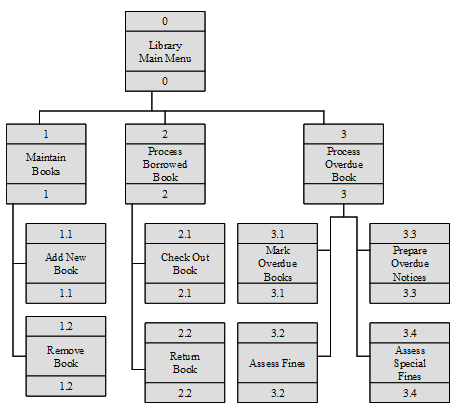
Assess fine – fine is added to borrower’s record for overdue book

Collect fine – fine is collected from borrower and removed from

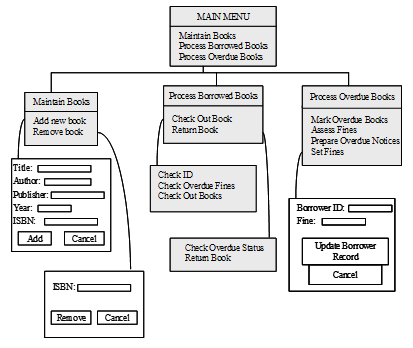
borrower’s record

Interface icons: Use University logo on screen; buttons should look like books

1. *Draw an ISD for the DFD in Exercise L in Chapter 4.*



1. *Design a storyboard for Exercise L in Chapter 4.*



1. *Develop two use scenarios for the DFD in Exercise L in Chapter 4.*

|  |
| --- |
| Use Scenario: Check out a book |
| 1. Borrower presents book to check out (2.1) 2. Borrower ID is checked for validity 3. Borrower record is checked for overdue books and fines 4. Book is checked out to customer |

|  |
| --- |
| Use Scenario: Book is returned |
| 1. Book is returned by borrower 2. Overdue file is checked. If book was overdue, it is removed from overdue file |

1. *Ask (*[*www.ask.com*](http://www.ask.com)*) is an Internet search engine that uses natural language. Experiment with it and compare it with search engines that use key words.*

On the Ask site, the user can phrase the request in any format: a sentence, a phrase, or a question. If the user is unsure of the key words that will produce the needed results, a natural language site may be beneficial. On the other hand, if the user knows the key words that are needed, a key word site may give the best results.

1. *Draw an ISD for “Your Turn 8-7” using the opposite grammar order from your original design (if you didn’t do it, draw two IDSs, one in each grammar order). Which is “best”? Why?*

Ex10s_1

Ex10s_2

The first ISD uses an “object-action” grammar order. The first level of the hierarchy lists the three main system objects, customers, products, and orders. The second ISD uses an “action-object” grammar order. The first level of this hierarchy lists the four major system actions, search, change, delete, and list.

Neither grammar order is inherently best; whatever seems most natural to the users is the preferred structure to use.

1. *Search the Internet for information on “Responsive Design.” Briefly summarize the meaning of this term and how it affects user interface design.*

A method of building websites that are highly functional on wide range of desktops and provide “app-like” experiences on mobile devises including smartphones, tablets, and e-readers.

# Answers to Textbook Minicases

*1. a. Experienced users take the position that once the system is learned, they should be able to use it quickly to complete the needed tasks. These users prefer a design that emphasizes the most commonly used functions with fewer options on menus. To maximize efficiency, these users like high information density, preferring a high ratio of information to white space on each page or form. For less frequently used functions, the users need guidance through unfamiliar parts of the system through “show me” or help functions.*

*1. b. Novice users will benefit from an interface that has high content awareness, so that they can easily identify the information on the screen and have a good sense of where they are in the system at all times. A key need for novice users is ease of learning the system, therefore, they like to see all available functions listed on menus. Novice users also benefit from low information density on the interface, preferring a balance between information and white space on the interface. These users will enjoy having guidance through all parts of the system through “show me” or help functions.*

*2. Several factors should be considered in making this decision. Among the factors are speed, cost, the uncertainty the team has about the interface design, and how widespread the use of the system will be. If the team has very little uncertainty about the interface design, then heuristic evaluation is probably acceptable. This process will check the interface design against the important design principles, and the team should be assured of a consistent interface. This method is the quickest and cheapest evaluation technique. If the team has little uncertainty about the interface design, but still wants to get user reaction in a quick and convenient forum, then the walkthrough evaluation is recommended. Whenever there is a situation in which the team has some uncertainty about the interface design, then it is essential to get the user’s response. The most convenient way to get this response is to use interactive evaluation. This enables the users to actually sit down and work with the system. Analysts can observe their interaction, look for trouble spots, and get the users’ immediate comment and feedback. Although this takes more time and cost, it is important whenever there might be questions about some of the design decisions. Usability testing is a very formal evaluation technique that is most commonly used by commercial software vendors. It is so time consuming and expensive that it is not generally warranted unless the software will be a commercial product, or it will be have widespread use throughout the organization.*

1. *Student designs will vary. The student’s interface design should reflect an attempt to prevent errors and make it easy to correct errors when they occur. Also, a consistent grammar order should be apparent in the design. Messages should be complete, informative, polite, and helpful.*
2. *Student designs will vary. The key elements to check for include use of radio buttons to indicate whether the new vehicle was an RV or trailer; also to indicate if the trade-in was an RV or trailer. The new vehicle names, models and manufacturer could be indicated with drop-down list boxes because of the known values for these fields. Salesperson name, plus option codes, descriptions, and prices may also be listed in drop-down list boxes (ten choices is probably too many to use check boxes).*
3. *Student designs will vary. The key element to check for is that lists are used for all of the information to be entered.*

# Supplemental Minicases

1. Charlie is a project manager for a new system under development for a national mortgage processing company. A common experience for IS managers today is to lose experienced IS staff members, who leave companies in search of more money and more interesting projects. This is exactly the situation Charlie finds himself in today. Two experienced members of his team have departed the organization within the last month. Now, Charlie must prepare for the arrival of his new team members, who are scheduled to join the team next week. Unfortunately for Charlie, the team members he lost had nine years of systems development experience between them, while his new staff members only have three years of experience between them. Charlie is very concerned about the lack of experience of the new team members.

To bring the new team members up to speed as quickly as possible, Charlie plans to have them spend several days reviewing the Project Binder, which summarizes all pertinent project information. He also needs to get the team moving ahead on the project as quickly as he can. The next task in the project workplan is the development of the user interface design. This is a good area to involve the new team members. In anticipation of their arrival, Charlie plans to develop a document that outlines the process that is followed in developing a user interface. This will help the new team members understand the steps that will be followed for the work they will be performing.

Prepare this document outlining the five-step process that is used in user interface design. Remember that this document will be used by the new team members to help them understand what is expected in each step of the user interface design process.

*Answer: Student answers should list and explain in clear terms the following five steps:*

1. *Identify ‘use cases’ that describe commonly used patterns of actions that users will perform. These use cases will be valuable in ensuring that the interface permits the users to enact these use cases quickly and smoothly.*
2. *Develop the interface structure diagram, defining the basic structure of the interface (screens, forms, and reports) and how the interface components connect.*
3. *Develop interface standards, the basic design elements that will be used throughout the interface.*
4. *Create prototypes of the various interface components (navigation controls, input screens, output screens, forms, and reports).*
5. *Evaluate the prototypes and make changes as needed.*
6. John is a systems analyst on the Holiday Travel Vehicle project team (see Minicase 4, this chapter). The team is currently developing the system’s user interface design. John has been given the task of developing a plan for prototyping the user interface, and he is struggling to determine the best method of prototyping to use. All of John’s prior experience with user interface prototyping has used the storyboard technique. John’s project manager specifically requested that he consider all types of prototyping (storyboarding, HTML prototypes, and language prototypes) before developing his prototyping plan. The system under development will be a significant departure from the interface that the users are accustomed to, and the manager wants to be sure that the interface design is clear and acceptable to the users before it is implemented.

Prepare a summary of the three major types of prototyping that can be used in a project. Identify the strengths and weaknesses of each method. Discuss the criteria that John should use to select a prototyping approach to use in the project.

*Answer: Storyboards are paper-based representations of the screens and the flow of control from one screen to another in the system. Storyboards are simple and inexpensive to create, but do not give an accurate picture of what the screens will actually look like. HTML prototypes develop the interface as a series of web pages. The web pages can be linked so that the users can navigate throughout the system, and space can be set aside for simulated data entry as well. HTML* prototypes *can be developed fairly quickly, but do not represent the actual appearance of the user interface (unless the system will be a web-based implementation). Language prototypes build the prototype screens using the actual language that will be used to implement the system. Therefore, these prototypes accurately represent the actual user interface users will see. These prototypes can also be set up to allow data entry and navigation between screens, so the users understand the interaction that will take place with the interface. These prototypes take the most time to develop and are therefore the most expensive, but have the advantage of depicting the system exactly as it will look upon implementation.*

*John’s decision criteria should be based on three factors: speed, cost, and the degree of uncertainty the team has about the interface. If the user interface is well understood, then storyboarding is suggested because of its speed and low cost. If all or part of the interface is not well understood, HTML or language prototypes should be used. The more uncertainty there is about the interface, the more the use of language prototypes is justified. In these cases, the extra time and expense associated with language prototypes is warranted because of the accurate detail these prototypes will present to the users.*

1. Prepare a written summary of the input design principles that you implemented in your prototyped input design in the textbook Minicase #4. Also, discuss the validation that you will implement in your input processing.

*Answer: The principle of on-line versus batch processing was not addressed in this scenario, but it seems likely that the data entry form the students design will be used to gather transaction data that will be processed as a batch at a later time. The need for real-time inventory records seems minimal in this setting. The scenario does reflect the principle of capturing the data at the source. Since the ‘data entry’ mechanism will now be the salespeople themselves, the need to minimize keystrokes is critical. It is important that the student’s design reflect an attempt to reduce keystrokes and increase accuracy through the use of radio buttons, check boxes, and drop-down lists. Invoice numbers should be automatically assigned by the system, and the current date should be automatically inserted. There are also several situations where references to database tables can be used to complete some information. For example, if the customer has purchased from Holiday Travel Vehicles in the past, the customer information should be able to be pulled up through the entry of the customer number or customer name. The entry of the new vehicle serial number should be sufficient to pull up the name, model, manufacturer, year, and base cost. Selection of an option code from a list should bring up the description and price. There should be automatic totaling performed for the options selected and automatic calculation of taxes. The final price of the purchase should be automatically calculated as well. If the students use the above design principles, the need for additional validation will be limited. Text and numeric field format checks should be performed where appropriate. Also, a completeness check should be performed for all required fields. Since much of the data can be supplied through database checks, there will not be a need for additional validation.*

1. Prepare the design and prototype of the printed Sales Invoice that will be printed upon completion of the sales transaction at Holiday Travel Vehicles. This will be the document that the customer takes with him/her summarizing the sale. Also, design and prototype an output report from the system that provides the owner of Holiday Travel Vehicles information about the prior week’s sales activity. This report should be available in two forms: one that groups the sales by vehicle manufacturer, and the other that groups the sales by salesperson. The owner wants to use the reports to quickly assess the sales performance of the various vehicles offered for sale and the performance of the various salespeople on staff.

*Answer: Student report designs will vary. Look for report designs that are organized thoughtfully with consideration to how the user will want the information presented.*

1. MER’s Fine Timepieces is a thriving business that specializes in sales and service of fine watches and clocks. Although the market for timepieces has changed as many people purchase inexpensive, ‘throw-away’ clocks and watches, there are still a large number of people who prefer quality timepieces, and they remain popular as gifts. A unique aspect of the business is the repair service that is available. MER’s Fine Timepieces employs skilled watchmakers and clockmakers who can repair all varieties of watches and clocks, including heirloom pieces.

To facilitate the maintenance of watch and clock repair records for the business, a new tracking system is under development. This system will enable the clerks at MER’s Fine Timepieces to record complete information on the repair item when it is brought into the store. When a watch- or clock-maker begins work on the item, he/she can update the repair record as work is performed. When the repair is complete, all work performed and labor hours are recorded on the repair record. When the customer returns for the item, the repair record is closed. The following specification has been prepared for the system’s user interface.

Main Menu: From this screen, the user can: enter a new service record, update a service record, close a service record, and check the status of an item being repaired.

Enter a New Service Record: When an item is brought in for repair, the clerk will record as much information about the item as possible. The system will assign a service ticket ID to the item. Complete customer information is recorded, including name, address, and home and business phone numbers. The date the item was received is entered, along with a date promised (if any).

The type of item should be specified. If the item is a clock, it can be categorized as one of the following clock types: wall-chime, wall-no chime, grandfather/grandmother, mantel-chime, mantel-no chime, 400-day, cuckoo, or other. The most common clock manufacturers are Ansonia and Sliegh, but there can be many other clock manufacturers. The actual manufacturer’s name should be recorded if known. If the item is a watch, it can be categorized as one of the following watch types: pocket, man’s standard wrist, woman’s standard wrist, man’s quartz wrist, woman’s quartz wrist, and chronograph/chronometer. The most common watch manufacturers are Bulova, Citizen, Hamilton, Wittnauer, and Rolex, but there can be other watch manufacturers. If the item is an antique or heirloom, it should be designated as such on the repair record.

In some cases, an estimate of the repair cost is made and entered into the system. The problem being experienced is generally that the clock or watch runs fast, runs slow, or won’t run at all. If the item runs fast or slow, the clerk will ask for an estimate of the number of minutes a day it runs fast or slow. If the item is a clock, another problem can be that it won’t chime. Details on the problem should be entered into the system by the clerk.

When the new service record is complete, an internal status of ‘in queue’ is assigned to the item. This designates that the item has been received but is not yet being worked on by a clock- or watch-maker.

Update a Service Record: When a clock- or watch-maker is ready to work on an item, he/she brings up the item’s service record to review the problem description using the service ticket ID. The item’s status is changed to ‘under repair.’

When the repair procedures have been performed, the clock or watch is moved to a special holding area and the item’s status is changed to ‘timing.’ The item will monitored for proper performance for at least three days. At this time, the watch- or clock-maker will record the services performed on the item (there may be more than one). The categories of services performed are: clean, overhaul, adjust timing, and other. An area for comments should be available to record other information about the repair. The number of labor hours should be recorded, along with the watch- or clock-maker’s initials and the completion date.

When the clock or watch has passed the timing tests, its status is changed to ‘complete.’

Close a Service Record: When the customer comes in to pick up an item (or the item is delivered in the case of grandfather/grandmother clocks), a clerk will close the service record. This involves collecting payment for the item, marking the service record as paid, and recording the date the item was returned to the customer.

Check Status: At any time, a customer may call to inquire about the status of a watch or clock being repaired. The clerk should be able to enter either the service ticket ID (if available) or the customer name to retrieve status information on the item. The clerk should be able to inform the customer if the item is in queue, under repair, timing, or complete.

Develop prototypes of the screens that will be needed for this scenario. You will need screens for the main menu plus screens for each of the menu options. Make use of GUI screen features as much as possible.

*Answer: Student designs will vary. There are several key features to watch for in this problem. First, the service record information on clocks and watches is similar, but each has some unique features. This is a good opportunity to use a tab menu (one for clocks, one for watches) as a way of simplifying data entry. There are ample opportunities to use drop-down lists or combo boxes (e.g., clock manufacturer, watch manufacturer, clock type, watch type). Radio buttons would be a good choice to indicate the item’s repair status, and also to designate the item as antique or heirloom if appropriate. A spinner could be used to indicate the number of minutes an item runs fast or slow per day. Check boxes would be appropriate for the services performed since there may be more than one. A large text box with scroll bars should also be provided for additional free-form comments about the repair.*

# Experiential Exercises

1. Purpose: to become familiar with the principles of user interface design.

Divide your class into small groups. Have each group evaluate a system that students use on campus, using the principles for user interface design as their guidelines. Systems with which the students are familiar would be best, such as the registration system or the placement system. In class, have each group share their assessment of the user interface. Probe for suggestions for improvement that are based on the principles of user interface design.

1. Purpose: to become familiar with the principles of user interface design.

Divide your class into small groups. Have each group evaluate two web sites that are used to transact business. As a class, select one site that all groups will evaluate (such as Dell Computers), and then let each group select the other site. Use the principles for user interface design as the evaluation guidelines. In class, have each group share their assessment of the user interfaces. Do the groups seem to share the same opinion of the site that all evaluated? If not, where do the differences come from? Can the class identify any common strengths or weaknesses observed on the web sites?

1. Purpose: to become familiar with evaluating user interface design.

Have students work in small groups. Each group should identify two web sites for companies that are competitors (e.g., two sites that sell books; two sites that sell computers, etc). First, have the students develop two use scenarios that would be common for visitors to those sites. Then, have the students draw an interface structure diagram for each site. Have the students work through the use scenarios on each site. Based on their experience and on the structures of the sites, have the students evaluate the sites’ user interface and report their findings to the class.

1. Purpose: to become familiar with usability testing.

Have students design a usability testing session for an on-campus system. Develop a set of tasks to be performed using the system. In a lab setting, bring in a several students who have not used the computer system. Give them a general orientation to the software, and then give them their tasks to perform. Since you will probably not have the means to capture every keystroke, videotape the session so that the students can review the user’s interaction with the system. It would be helpful to debrief the student on tape following the session so that they can discuss their success or frustration with the tasks.

1. Purpose: to evaluate navigation methods.

Have students work with a partner in the exercise. Each pair of students should evaluate your university's Web site and two other, commercial web sites with respect to the navigation methods that are employed. Have students use the principles of navigation design to assist in their evaluations. In class, discuss the consensus of opinion regarding your university's web site. The other sites the student's visit may suggest particularly confusing or difficult navigation controls. If the students identify any such examples, discuss how the principles of navigation design could be used to correct the situation.

1. Purpose: to understand the linkage between source document design and input form design.

Have students work in groups of four and generate a list of all the information that is needed to complete the registration process as your university. Then, have each group split into pairs. Have one pair design a source document that could be used to capture that information. Have the other pair prototype an input form that could be used to enter the data into a computer system. Do not allow the pairs to communicate with each other during this step. When they are finished, have the group compare the source document design with the input form design. Chances are good that the pairs of students will have made some different assumptions, and that it would be difficult to enter the data from the source document into input form without some skipping around. This should trigger some good discussion of the need to coordinate these activities.

1. Purpose: to understand the various input validation method that can be employed in an input process.

Have the student groups from Exercise 2 above develop a written summary of how each field in their student registration input form will be validated. Establish a requirement that every field must have at least one validation method applied to it. For those fields for which multiple validation methods are possible, have the students list every one that could be used. Have the students develop error messages for each validation technique applied. This exercise will help the students appreciate the care that must be taken in developing an input process that conforms to the principles of good input design.

1. Purpose: to practice the principles of good output design.

A report with which your students should be familiar is the Degree Audit report (another name may be used at your institution). This report is suggested because it is notorious at many schools for being cumbersome and confusing. It can serve as a good example of a report that is designed for multiple, conflicting purposes and therefore is practically unusable.

Have the students bring in a copy of their reports. Have them generate a list of the questions that can be answered by referring to the report. This will help them appreciate the multiple purposes for which the report is designed. Second, for each question that can be asked, have them identify the information presentation that they would find most useful to answer just that question. Finally, have the class consider how many different, more specialized reports could be produced that would be more helpful and less confusing to students. Discuss why the university chose to create the Degree Audit in its current form.