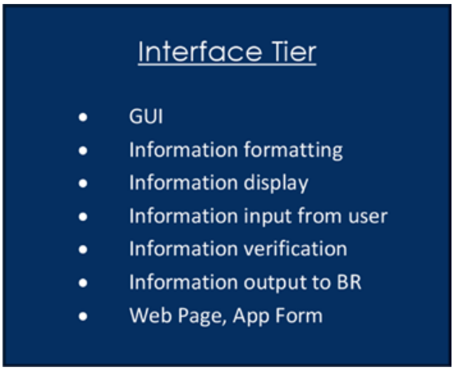
Interface Tier



Interface Tier

- GUI
- Information formatting
- Information display
- Information input from user
- Information verification
- Information output to BR
- Web Page, App Form

Figure 2-13. *Interface tier*

The interface tier (IT) displays information and provides the user the ability to interact with the application. Most interfaces provide a Graphical User Interface (GUI), which allows the user an attractive way to view information and interaction with the application. GUI interfaces provide common **objects**, including textboxes and buttons, which help the user to quickly adapt to new applications. In addition, pictures, images, icons, video, and sound can often be included to keep the user's interest. Menus and other navigation objects are also commonly included to help the user move through the application successfully.

> *Objects—Objects are blocks of code that have already been compiled for use within an application. Objects can be placed into a program by making an instance of the object. Objects contain methods and properties. Methods (or functions) are blocks of code that accomplish a task (such as placing items in a list box). Properties (or variables) are characteristics of an object that can be changed (such as a background color). Objects are usually well tested and error free. By reusing existing objects, the programmer can quickly create more reliable programs.*

This tier will display information using objects (such as labels and picture boxes) or scripting code (such as the `ajaxdemo.html` example in this chapter). The tier will also accept information from the user through interactive objects (such as textboxes and buttons). Static information can be provided from within the tier (via menus, logos, or footers). Dynamic information is usually provided to the tier from the business rules tier (such as the output from `myfirstprogram.php` shown in this chapter).

Some coding (shown in later chapters), which prepares information to be send to other tiers may be present in the interface tier. For example, JavaScript code that verifies that the user has entered all the required information or the proper information (numeric characters in an age textbox) is acceptable. Additional code may also prepare information received from the business rules tier to be displayed (such as converting numbers to text format) in the interface.

> *Verification/Verification Code—Verification code validates information. The code compares the information received to an expected standard format. For example, the code can verify that an e-mail address has both a @ and a . (period). If the information has both symbols, it could be considered "valid" (although we still are not sure that the e-mail address actually exists). If it does not have both symbols, the code is not valid. Information that is not valid usually will cause the program to display an error message to the user requesting valid information be reentered.*