

The interface must provide code with the ability to react to user interactions (clicking the Submit button), commonly called *user events*. Code may also be provided that prepares information provided by the user for use by other tiers (such as converting text entered by the user into number format for a calculation in the business rules tier).

*Event—Event-driven languages (such as PHP) can execute blocks of code when an event occurs. Events can be something that the user has done (such as clicking a button). Events can also be fired by the operating system. Programs provide listener code that “hears” an event. When an event occurs, the code provides an event method, which then executes. A program chooses which events to listen for by the presence, or lack of, listener code.*

The interface tier should not directly interact with a database management system or a database itself. By doing so, this would lock the tier into the database location and actual design of the database. The tier should not manipulate data (except for display purposes). Any accounting, mathematical calculations, or processing of data related to the application itself should be accomplished in the business rules tier.

*Database Management System (DBMS)—A Database Management System is software that allows a user or an application to create and define a database. It also provides the ability to insert, update, or delete information in the database.*

Do	Don't
Format data for display	Access data from database
Verify correct information from user	Calculate results
Respond to user events	Process information
Handle the unexpected (exceptions)	Verify user IDs and passwords
Format data for business rules tier	

## Do It

1. Give three examples of items that would be included in the interface tier.
2. Give three examples of items that would not be included in the interface tier.
3. Can some program code exist in the interface tier? If so, what tasks does this code provide?