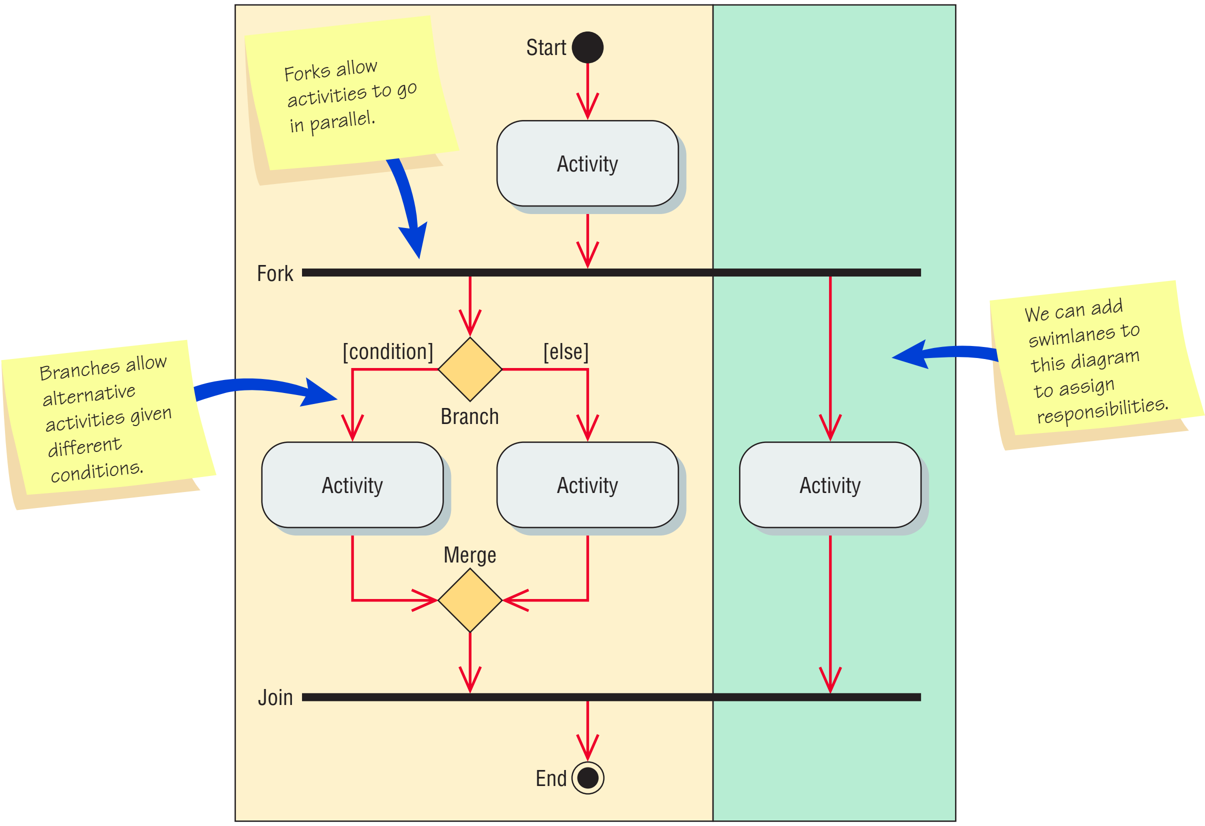
Forks allow activities to go in parallel.
Start
Activity
Fork
Branches allow alternative activities given different conditions.
[condition]
[else]
Branch
We can add swimlanes to this diagram to assign responsibilities.
Activity
Activity
Activity
Merge
Join
End

FIGURE 10.8

Specialized symbols are used to draw an activity diagram.

An arrow represents an event. Events represent things that happen at a certain time and place.

A diamond represents either a decision (also called a branch) or a merge. Decisions have one arrow going into the diamond and several going out. A guard condition, showing the condition values, may be included. Merges show several events combining to form one event.

A long, flat rectangle represents a synchronization bar. These are used to show parallel activities, and may have one event going into the synchronization bar and several events going out of it, called a fork. A synchronization in which several events merge into one event is called a join.

There are two symbols that show the start and end of the diagram. The initial state is shown as a filled-in circle. The final state is shown as a black circle surrounded by a white circle.

Rectangles surrounding other symbols, called swimlanes, indicate partitioning and are used to show which activities are done on which platform, such as a browser, server, or mainframe computer; or to show activities done by different user groups. Swimlanes are zones that can depict logic as well as the responsibility of a class.

You can see an example of swimlanes in Figure 10.9, which illustrates an activity diagram for the **Change Student Information** use case. It starts with the student logging onto the system by filling out a Web form and clicking the **Submit** button. The form is transmitted to the Web server, which then passes the data to the mainframe computer. The mainframe accesses the STUDENT database and passes either a "Not Found" message or selected student data to the Web server.

The diamond below the **Get Student Record** state indicates this decision. If the student record has not been found, the Web server displays an error message on the Web page. If the student record has been found, the Web server formats a new Web page containing the current student data in a Web form. The student may cancel the change from either the **Logon System** or the **Enter Changes** states, and the activity halts.

If the student enters changes on the Web form and clicks the **Submit** button, the change data is transmitted to the server and a program starts running that validates the changes. If there are errors, an error message is sent to the Web page. If the data are valid, the student record is updated and a Change Student Journal Record is written. After a valid update, a confirmation Web page is sent to the browser and the activity terminates.

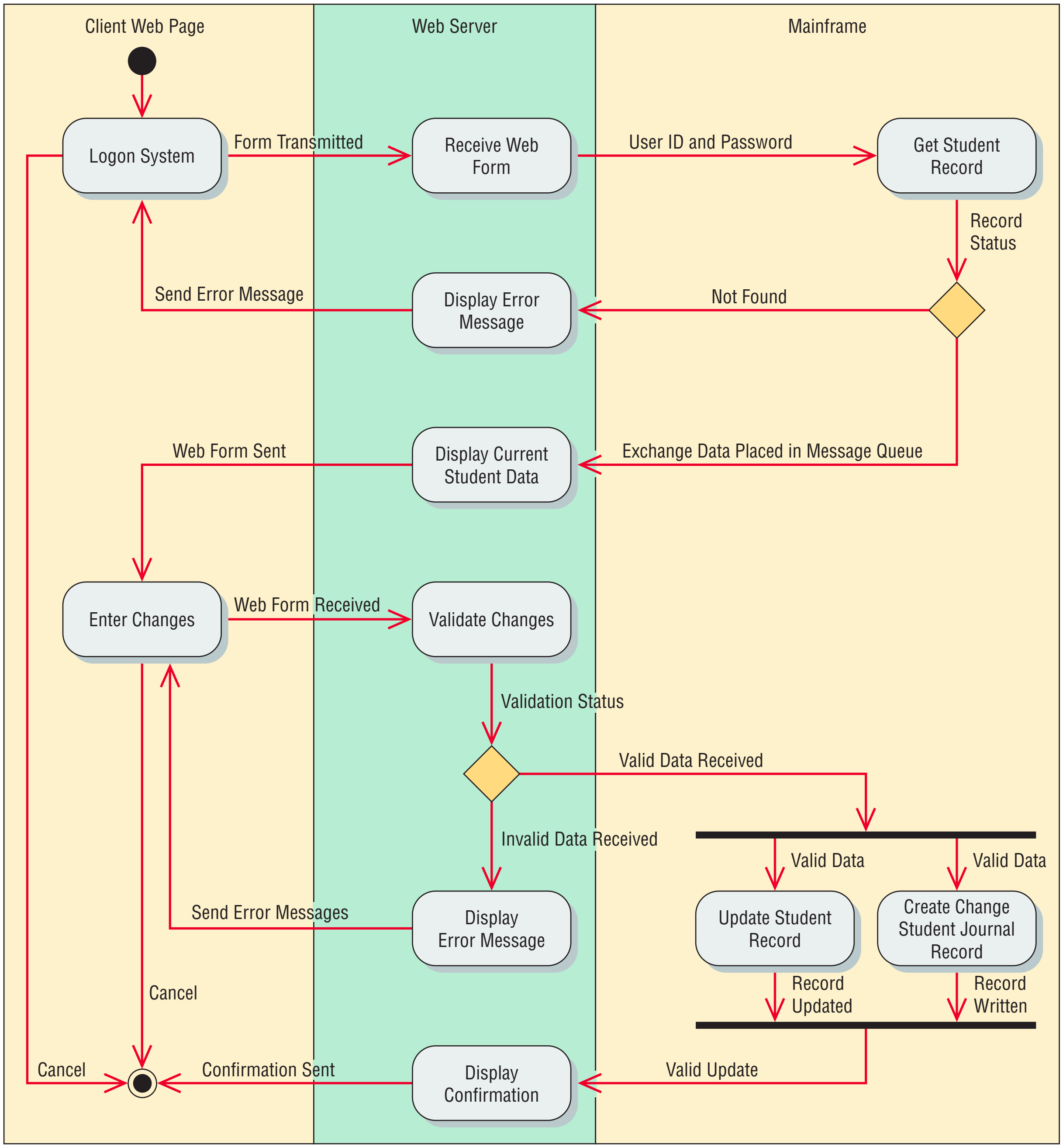

FIGURE 10.9

This activity diagram shows three swimlanes: Client Web Page, Web Server, and Mainframe.

Creating Activity Diagrams

Activity diagrams are created by asking what happens first, what happens second, and so on. You must determine whether activities are done in sequence or in parallel. If physical data flow diagrams (as described in Chapter 7) have been created, they may be examined to determine the sequence of activities. Look for places where decisions are made, and ask what happens for each of the decision outcomes. Activity diagrams may be created by examining all the scenarios for a use case.

Each path through the various decisions included on the use case is a different scenario. In the main path would be **Logon System, Receive Web Form, Get Student Record, Display Current Student Data, Enter Changes, Validate Changes, Update Student Record, Create Change Student Journal Record,** and **Display Confirmation.**

This isn't the only scenario that comes from this use case. Other scenarios may occur. One possibility could be **Logon System, Receive Web Form, Get Student Record,** and **Display Error Message.** Another scenario could be **Logon System, Receive Web Form, Get Student Record, Display Current Student Data, Enter Changes, Validate Changes,** and **Display Error Message.**