Performance testing (PT) is not to be construed as features testing even though it has a definite link with the latter. In fact, PT begins from where the feature testing ends. It presupposes successful completion of features testing, meaning that all the desired functional requirements expected from the system are fully met. However, it is worth noting that conducting PT without features testing is often fraught with great difficulties. This is because the residual defects in the system, which otherwise would have been known in features testing, would give misleading results about the performance. Previous chapters emphasized the need for PT and various factors that have impact the performance of Web-based applications. This chapter provides an insight about the technology aspects, including the software languages, necessary for Web development. Needless to add, that technology as well as software languages have a critical impact on the performance of the system.

Client Server and Web-Based Technology

The evolution of networking in recent decades has set an interesting challenge for the growth of software science. This evolution is the result of fomenting many new as well as radical ideas over a period of time and manifesting them into invention of new
technologies. The ensuing section emphasizes the evolution of Web-based technology as a successor to client-server technology.

**Client Server Technology**

Client server technology (see *Client/server software architectures – An overview*, 2004) is the combination of two simple but distinct processes, that is, client and server process as shown in Figure 3.1. It works on the principle of the request-response mechanism wherein the client process requests a specific service from the server and the server process responds to the client by servicing the request. Client programs usually manage the user interface of the application. The client also manages the local resources such as the monitor, keyboard, workstation, and CPU. A significant volume of processing is done locally at the client side. A server fulfills the client’s request by performing the desired task. Such tasks include database retrieval and updates, servicing file requests, managing data integrity, and sending responses to client requests. The server process may run on another machine on the network.

In a broader vision, a server process acts as a software engine that manages shared resources such as databases, printers, modems, or high-powered processors. Client and server have fundamentally different requirements for computing resources such as processor speeds, memory, disk speeds and capacities, and input/output devices.

In such an environment, the performance of the overall system is the function of the performance of both client and server processes. Testing the performance of the system calls for testing not only the client and server but also the network connecting both the entities. The hardware platform and operating system of a client and server are not usually the same. Generally, their environment is heterogeneous and multivendor based. Client and server processes communicate through a well-defined set of standard application program interfaces (APIs) and Remote Procedure Calls (RPCs) (see Marshall, 2004), which enable smooth and interoperable data transfer.

*Figure 3.1. Simple client-server technology*
Applying Web-Based Collaborative Decision-Making in Reverse Logistics: The Case of Mobile Phones