place in the object as the invoked operation executes. The activation continues until the operation finishes processing when control returns to the object that sent the message. If an active object, in turn, sends a message, it remains active while it waits for a response. While it waits for a response it cannot, itself, do any computing. Showing activation is an optional feature of sequence diagrams.

- A lot of detail is omitted on a diagram that is drawn during analysis. It says nothing about how the Receptionist manages to send a message to the :Bike. In fact when we get to the design version of this diagram, we will use an interface object, which will offer options to the user and translate them into messages to objects.
- The analysis diagram also omits any detail about how the matching bike object is found. At this stage we can take it on trust that it is found and leave the detail of how this happens to the designer (see Chapter 9).

The next two steps are:

5	Stephanie says she wants to hire the bike for a week
6	Annie enters this and the system displays the total cost £14 + £60 = £74.

Using the getCharges() operation with the parameter (no.Days) (see Figure 6.10) we can ask :Bike to work out the cost for the required period. The system is organized in such a way that the customer has the opportunity to see the total cost before being committed to the transaction. Only once the customer has agreed to the cost do we record details about the customer and the hire.

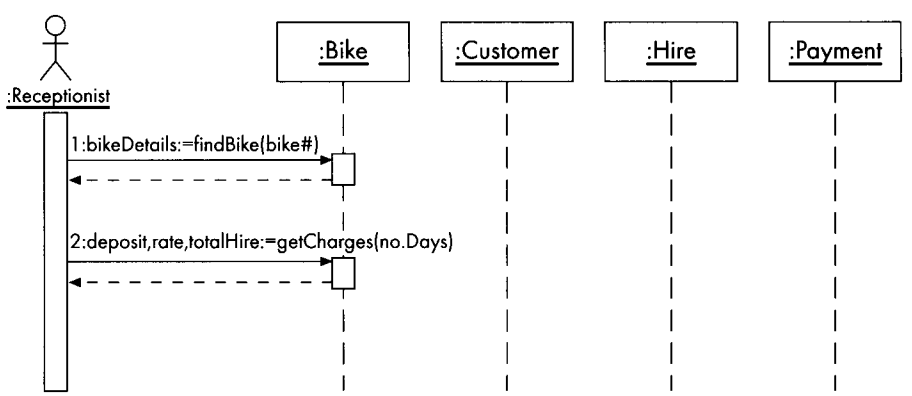

Figure 6.10 Fragment of sequence diagram