

In the same way, the first part of the use case description for 'Issue bike' repeats the behaviour of the use case 'Handle enquiries', Annie always tells customers the daily hire rate and deposit for a bike before going ahead with the issuing. Rather than repeat a description of this behaviour in both use cases, we can remove it from 'Issue bike' and have an «include» relationship between 'Issue bike' and 'Handle enquiries'. Notice that the dashed arrow points from the main use case to the one to be included, e.g. from 'Issue bike' to 'Handle enquiries'.

*Extend.* The «extend» relationship is used as a way of specifying significant alternative behaviour in a use case. It usually documents functionality that the user can opt to use over and above the norm. The practice of using an «extend» relationship in this way is only for documenting important variations from the normal course of events. Minor variations can be covered in the extended use case description. We would use an «extend» relationship if we want to describe:

- Extra functionality that is available if required, for example printing a list rather than just viewing it on the screen.
- Behaviour done only under certain conditions, for example printing an extra receipt if the whole deposit is not returned.

If, therefore, we want to specify a chunk of behaviour that is additional or exceptional to the normal sequence of events in a use case, we can create a new use case for that behaviour and specify an «extend» relationship between the new and the original use case. In Figure 3.9 we have created a new use case 'Print receipt' and an «extend» relationship between this use case and 'Handle bike return'. What this means is that sometimes returning a bike might involve printing a receipt, although this is not what normally happens. Printing a receipt will only be necessary if the customer has kept the bike for more days than they originally paid for, or if the bike is returned damaged. By contrast, printing a receipt is always part of the 'Issue bike' use case, so we have specified an «include» relationship between 'Issue bike' and 'Print receipt'. Notice that now the dashed arrow points from the new extending use case to the main use case, i.e. from 'Print receipt' to 'Handle bike return'. There seems to be no particular reason for this change of direction, it's just the rule.

If we decide that part of the use case 'Issue bike' will quite often involve adding a new customer, or updating our existing customer details, then it will be sensible to specify an «extend» relationship between 'Issue bike' and 'Maintain customer list' (see Figure 3.10). An «extend» relationship is more appropriate here than an