- Stephanie arrives at the shop at 9.00am one Saturday and chooses a mountain bike
- Annie sees that its number is 468
- Annie enters this number into the system
- The system confirms that this is a woman's mountain bike and displays the daily rate (£2) and the deposit (£60) (**:Bike**)
- Stephanie says she wants to hire the bike for a week
- Annie enters this and the system displays the total cost £14 + £60 = £74 (**:Bike**)
- Stephanie agrees this
- Annie enters Stephanie's name, address and telephone number into the system (**:Customer**)
- Stephanie pays the £74
- Annie records this on the system and the system prints out a receipt (**:Payment collaborating with :Customer**)
- Stephanie agrees to bring the bike back by 5.00pm on the following Saturday.

Figure 6.2 *Scenario for the 'Issue bike' use case showing the objects that will carry out the different responsibilities*

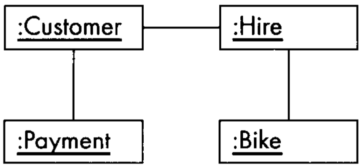

Figure 6.3 *Collaboration of objects required by the 'Issue bike' use case scenario in Figure 6.2*

Applying the CRC method to the Wheels classes in Figure 6.3 yields the set of responsibilities shown in Figure 6.4.

Deriving operations from CRC responsibilities

As we start to move from analysis to design we need more detail about how class responsibilities are going to be carried out. This means that we need to specify the responsibilities in terms of individual operations and attributes; we must ensure that each class has the data and operations needed to fulfil its responsibilities in the way that the user requires. We demonstrate this for the