**Make a reservation**

The client sends the reservation request to the reservation manager through one of the method offered by its view. The resource manager needs to ask to the car manager whether the reserved car is still available: if it is, a new reservation will be instantiate by the reservation manager that will also set the car and the user of the new reservation and start the reservation timer. If the car is no longer available an error will be notified to the clientApp.

**Start ride**

Once the client is at most five meters distant from the car he is reserved, he can send the request to unlock the car to the sever. This request is handled by the reservation manager who checks whether the client is actually close to the car. If this check goes well the reservation manager declares expired the reservation and unlocks the car. Then informs the ride manager that a reservation has just turned into a Ride. The ride manager is in charge of instantiating the new Ride and “waking-up” the CarApp on board the interested car. The new Ride will be instantiate only once the user will ignite the car. If the reservation manager detects that the user is more than five meters far from the car an error will be notified to the ClientApp.

**Available Cars Visualization**

The request is handled by the car manager who extracts all the cars from the database and sends them back to the clientApp. The ClientApp has an internal render that is able to filter the cars both on the user position and on a given position inserted by the user.

**End Ride**

This procedure can start only once the user has turned-off the car. He sends the request of terminating the ride through the on board CarApp. The ride can terminate only if the car is parked in a safe area, so the ride manager handles the request asking to the city manager whether the car’s position is actually in a safe area. The city manager, in turn, looks in the database for a safe area that matches the car’s position. If this search goes well then the ride manager queries the database in order to find the non-terminated ride of the given car and set the “terminated” attribute of this Ride to TRUE. Then the ride manager calculates the preliminary total fee, i.e. without taking into account eventual fines or discounts (the definitive total fee will be sent to the client app after three minutes from the ride termination), and send it back to the CarApp.