Below is represented the “bill's algorithm”. Before the driver ends the ride and exits the car, the system starts checking the state of sensors, the position of the car towards the position of the nearest safe area and last but not least the state of the battery (COME GESTIRE SCONTO 30%). The events generated and their consequences are discussed in the following table.

**[Legend]**

* + - *D*: driver;
    - *S*: System;
    - *C*: car;
    - *B*: battery;
    - LoP: list of passengers;
    - sA: safe area

|  |  |
| --- | --- |
| **Event** | **Consequences** |
| *D* exits C | *S* starts checking |
| Check the distance between the  sA and the current position | If((sA.nearest()-D.currPos())>3)  {D.applyTax()}  Else  {D.applyDiscount()} |
| *Check the number of passengers* | If(LoP.size()>2)  {D.applyDiscount()} |
| *Check the battery state* | If(B.getState()<20)  {D.applyTax()}  If(B.getState()>50)  {D.applyDiscount()} |
| *D* ends the rent | *C.* status = ”Ready” |
| Mettere sconto per ricarica. Discuti con Ciccio |  |

Below is represented the “reservation/rent algorithm”. The algorithm starts when a user clicks on the map provided by the system; immediately the controller of the system hides the selected car and starts the time of an hour (maximum amount of time that the user can wait before starting the rent). If the time exceeds the fixed constraint, the car returns available on the map, otherwise the status is "rented" (because it means that the user has pressed the start button and the ride can begin).

**[Legend]**

* + - *U*: user;
    - *R*: reservation;
    - *C*: car;
    - *S*: system;

|  |  |
| --- | --- |
| **Event** | **Consequences** |
| *U* selects car on the map | *C.status = “Reserved”*  *S.hideCar();* |
| Check the reservation’s time and compare it with the current time | If(system.getCurrTime()-R.time()>1)  {C.status=”Free”;  U.applyTax()  S.showCar()}}  Else  {C.status=”Rented”} |
| *Compare the positions* | If((U.position-C.position)<1)  {myApp.enableStartButton()} |
| *User starts the engine* | C.startCharge() |