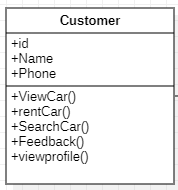
***OCL***



Invariant: Id

Description: Ensures that each customer has a unique identifier.

Expression:

for all customers c1 and c2, if c1 is not equal to c2, then c1's ID is not equal to c2's ID.

inv Name: self.name <> null and not self.name.isEmpty()

inv ValidPhone: self.phone.matches('[0-9]+')

**Operation constraints**

context Customer::viewCar()

pre: self.loggedIn = true

context Customer::rentCar(car: Car)

pre: self.loggedIn = true and car.available = true and car .size()>0

context Customer::search

pre: not keyword.isEmpty() and car.size()>0

context Customer::search

pre: not keyword.isEmpty() and car.size()>0

context Customer::feedback(message: String)

pre: not message.isEmpty()

A diagram of a graph

Description automatically generated

context Customer

inv: self.role <> null

context Customer :: viewCar()

pre: self.role = 'admin || user'

**Operation constraints**

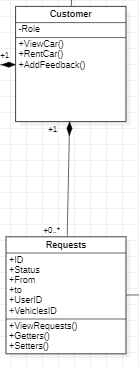
context Customer:: addCar()

pre: self.role = 'admin'

context Customer::addFeedback(description: String)

pre: self.role = 'customer'

post: Feedback.allInstances()->exists(f | f.description = description)



context Requests

inv :self.status = Accept or self.status = 'rejected'

context customer

inv : self.requests -> not Empty()

A diagram of a vehicle

Description automatically generated with medium confidence

Context : No of passengers

Invariant: No of passengers -----> size() <6

context Vichelle

inv:

-- This invariant ensures that when adding a vehicle,

-- it must have a license.

self.car.licence <> null

context Vichelle

inv:

-- This invariant ensures that the licence attribute

-- of Vichelle instances has a size of 3.

self.licence->notEmpty() implies self.licence->size() = 3

context Customer

pre rentcar:

self.licence->notEmpty()