UNIVERSIDAD POLITECNICA DE VALENCIA

Laboratory

Laboratory Case

Rent a Vehicle

Software Engineering

Computer Science School
DSIC – UPV

Year 2015-2016

A group of young entrepreneurs, after a thorough market analysis, has decided that a very profitable business will be the rental of vehicles. Therefore, they have decided to create a company and asked an external software engineering company ISWSoft to develop the software to be used.

General description:

The company will offer rental vehicles to every person with a valid driver's license who requests it.

The company will have branch offices throughout the world. In each branch office, there will be vehicles to be rented. Vehicles will be classified according to its category. Each category will have a different daily rental cost.

When a customer makes a reservation of a vehicle on the Web site of the Company he(she) may register to the system (new customers) or provide his existing customer id. In order to make a customer registration, the customer must provide the following information: first name, last name, address, city, postal code, driver's license issue date as well as information about his(her) credit card (company, credit card number, expiration month and year and control code).
As part of a reservation, the customer must also indicate the category of vehicle he(she) is interested in, the pick-up date and place, and the return date and place. In addition to this information, the customer may also request additional equipment such as baby car seat, GPS, refrigerator, etc. Each additional equipment has a specific daily rental cost. There are two possible rental modalities to choose from when making a reservation: with unlimited mileage or with limited mileage plus additional cost per extra kilometer. The cost per extra kilometer is different for each category of vehicle. A customer is allowed to modify or remove a reservation at any time before the pick-up day.

At the time of picking a vehicle up, an employee of the company must verify the information given by the customer in the reservation, choose among the available vehicles in the requested category, verify the status of the vehicle in terms of damages before it is delivered to the customer, its mileage and fuel level. Every vehicle delivery is associated to the employee who has managed it just in case any future problems arise. If at the time of picking a vehicle up there are no available vehicles in the requested category, a vehicle of a higher category must be delivered to the customer without extra cost. In addition, the customers must undertake insurance. There are two types of insurance: full insurance or insurance for damage to third parties. In the latter case, the customers will pay any damages made to the rented vehicle.

Once a vehicle is returned, the employee must verify its state, verify that all the extra equipment is returned and register the new mileage and fuel level. The system will calculate the total amount of money to be paid and the employee must proceed with the payment transaction.
In case a reserved vehicle is not finally picked-up, the system will charge (the following day) a one-day full payment using the registered credit card and the vehicle will be again available.

In order to cope with high demand, vehicles may be transferred between branch offices. Once a week (every Friday evening), the administrator of the company requests to the system a list of possible transfers between branches to cope with the needs of the following week. The vehicles included in the list are those that minimize the travel

distance between branch offices and allow to cope with the existing needs. A vehicle will not be transferred from a branch office if there are not enough vehicles to cope with the reservations made in that branch for the following week. The administrator approves or rejects each proposed vehicle transfers. An employee from the origin branch office must physically transfer the vehicle to another destination branch office before Monday 9:00. When a transferred vehicle leaves a branch office, the system must record the employee who will transfer the vehicle, the transfer date and time and the mileage of the vehicle. As soon as the vehicle arrives to its destination branch office, another employee must record the arrival date and time and mileage of the vehicle.

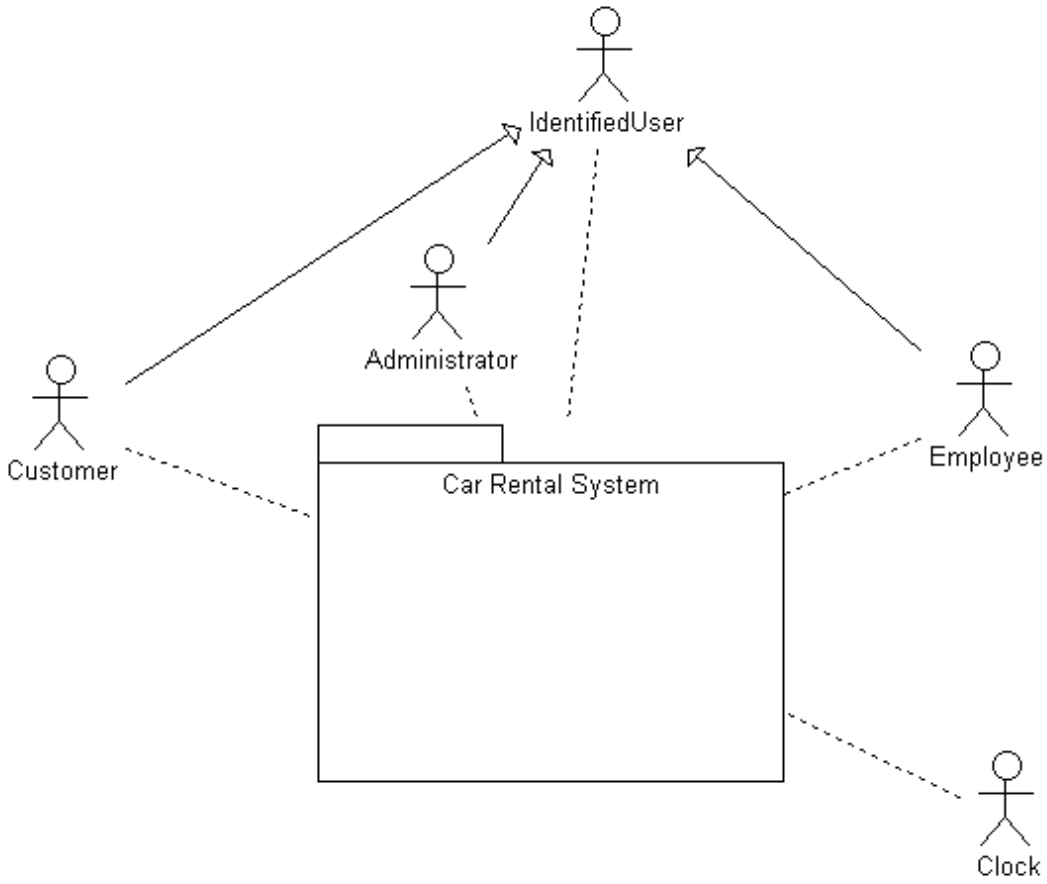
The system must manage all the functionality described above, In addition, the system must allow the administrator to perform CRUD (Create, Read, Update and Delete) operations for branch offices and customers. The employees may also update and delete reservations and rentals. The search operations with respect to vehicles, customers, reservations and rentals must also be supported. Finally, the system must also provide the following reports for every employee of a branch office:

- Existing branches
- Vehicles in a branch office organized per category
- Available vehicles in a branch office for a given date and organized per category.
- Vehicles needed in a given date to cope with the existing reservations.
- Unreturned vehicles.
- Rentals in a given branch office in a period of time

1. Use Cases Model

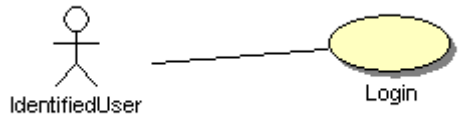
Car Rental System
(Use Cases Model)

Context Diagram

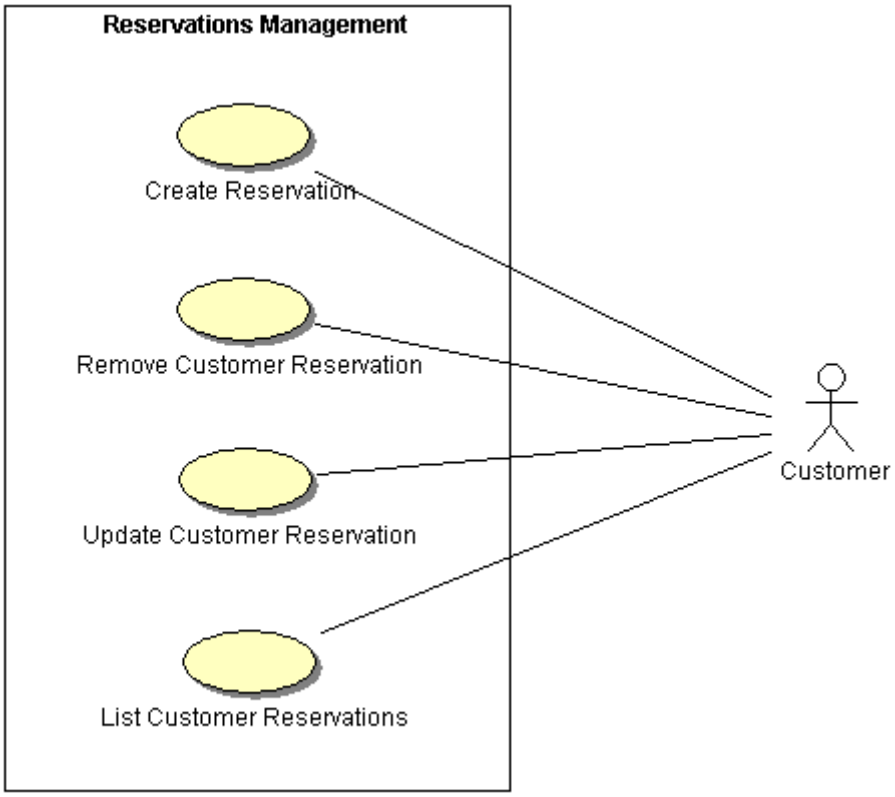


Initial Diagrams

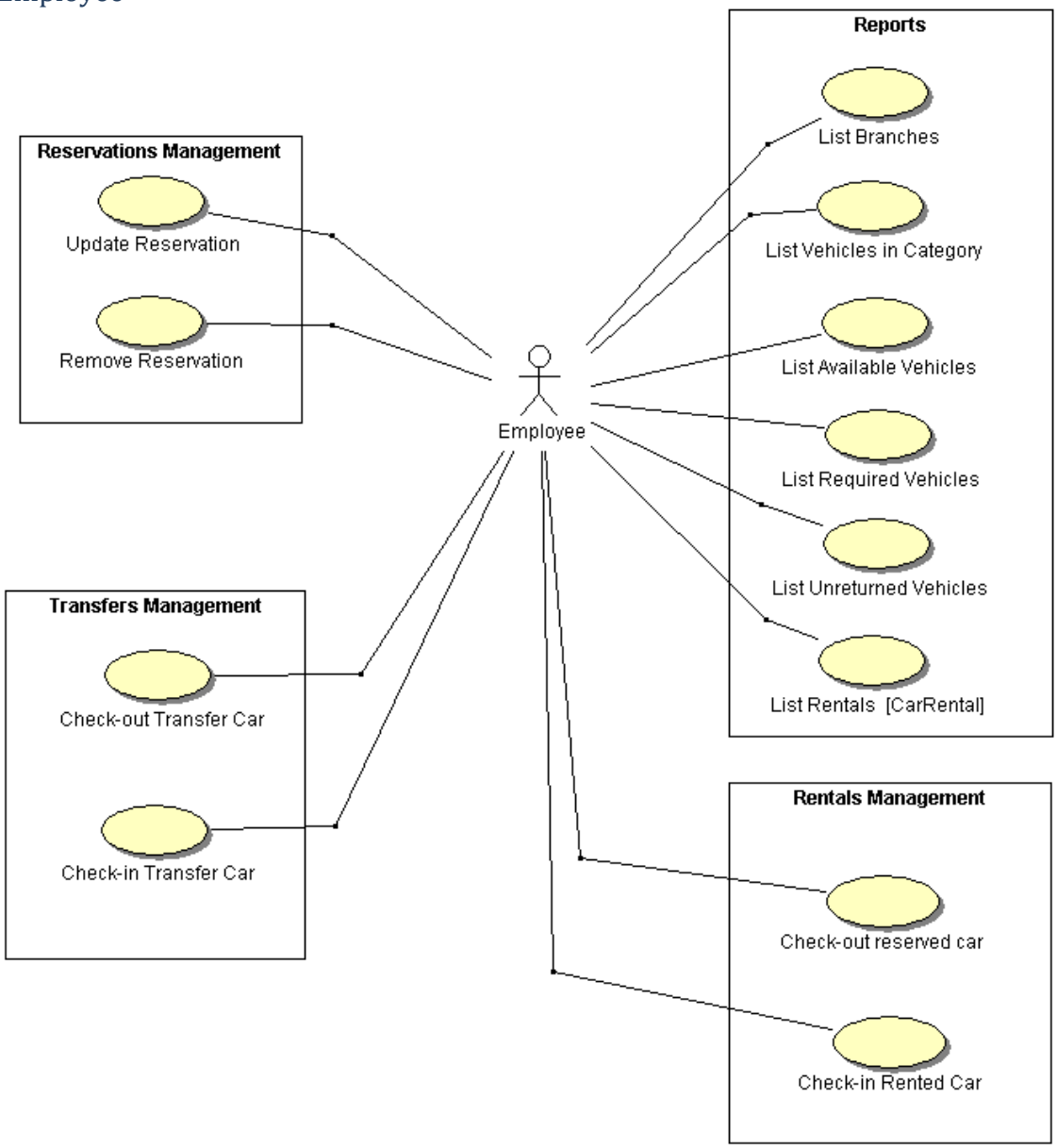
Identified User



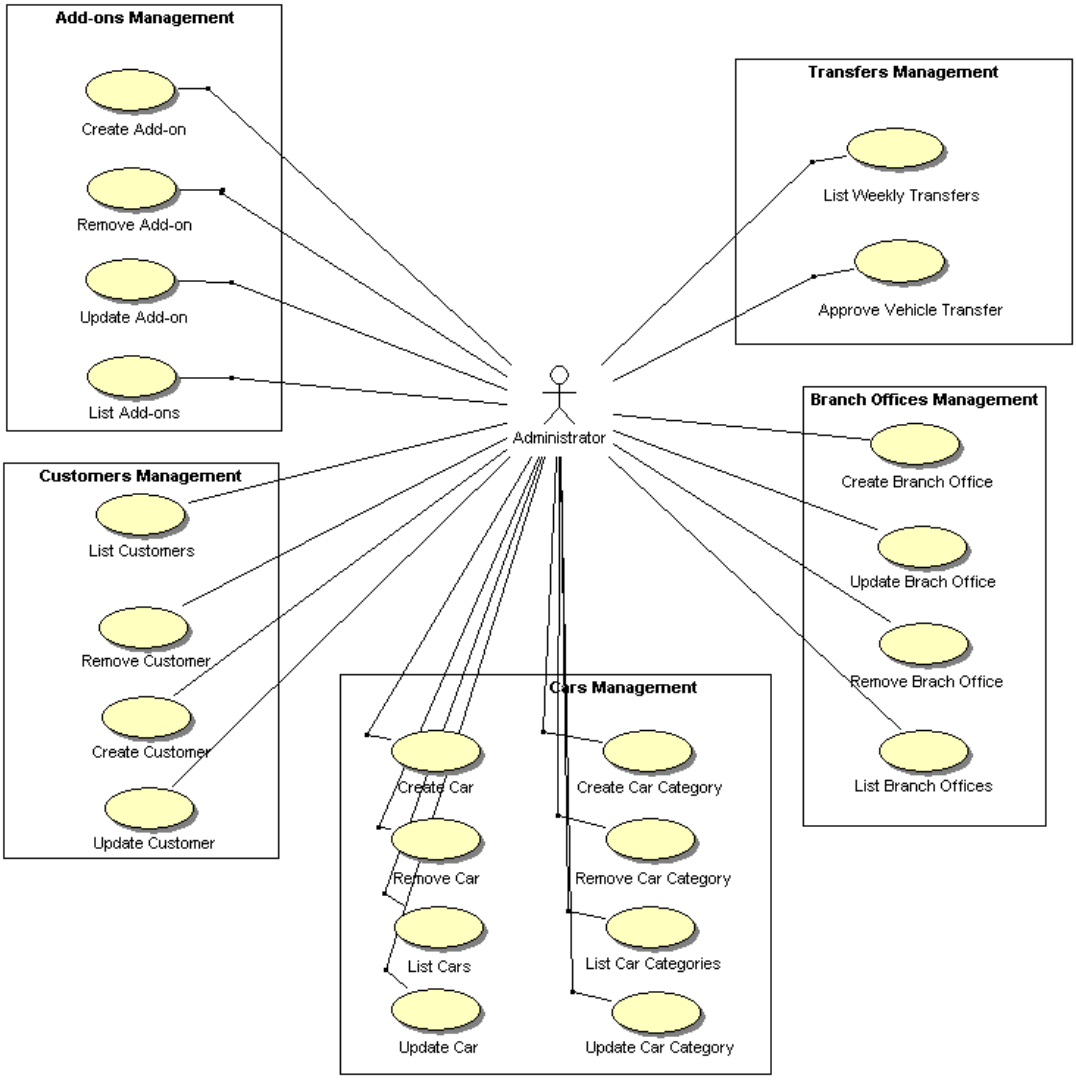
Customer



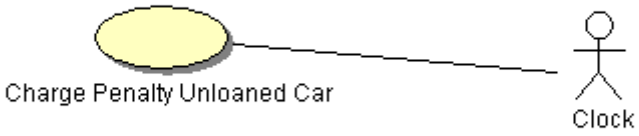
Employee


Reports
List Branches
List Vehicles in Category
List Available Vehicles
List Required Vehicles
List Unreturned Vehicles
List Rentals [CarRental]
Reservations Management
Update Reservation
Remove Reservation
Employee
Transfers Management
Check-out Transfer Car
Check-in Transfer Car
Rentals Management
Check-out reserved car
Check-in Rented Car

Administrator

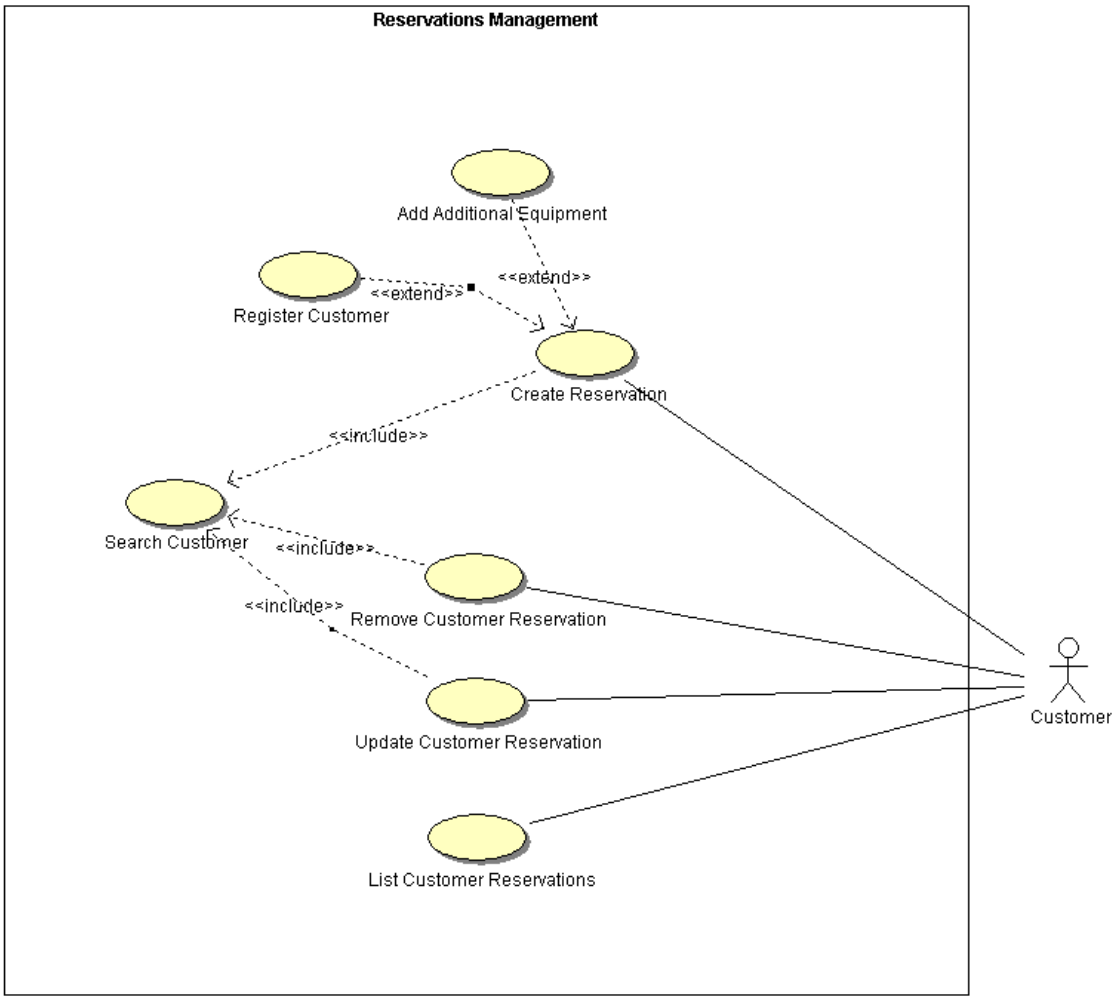


Automatic

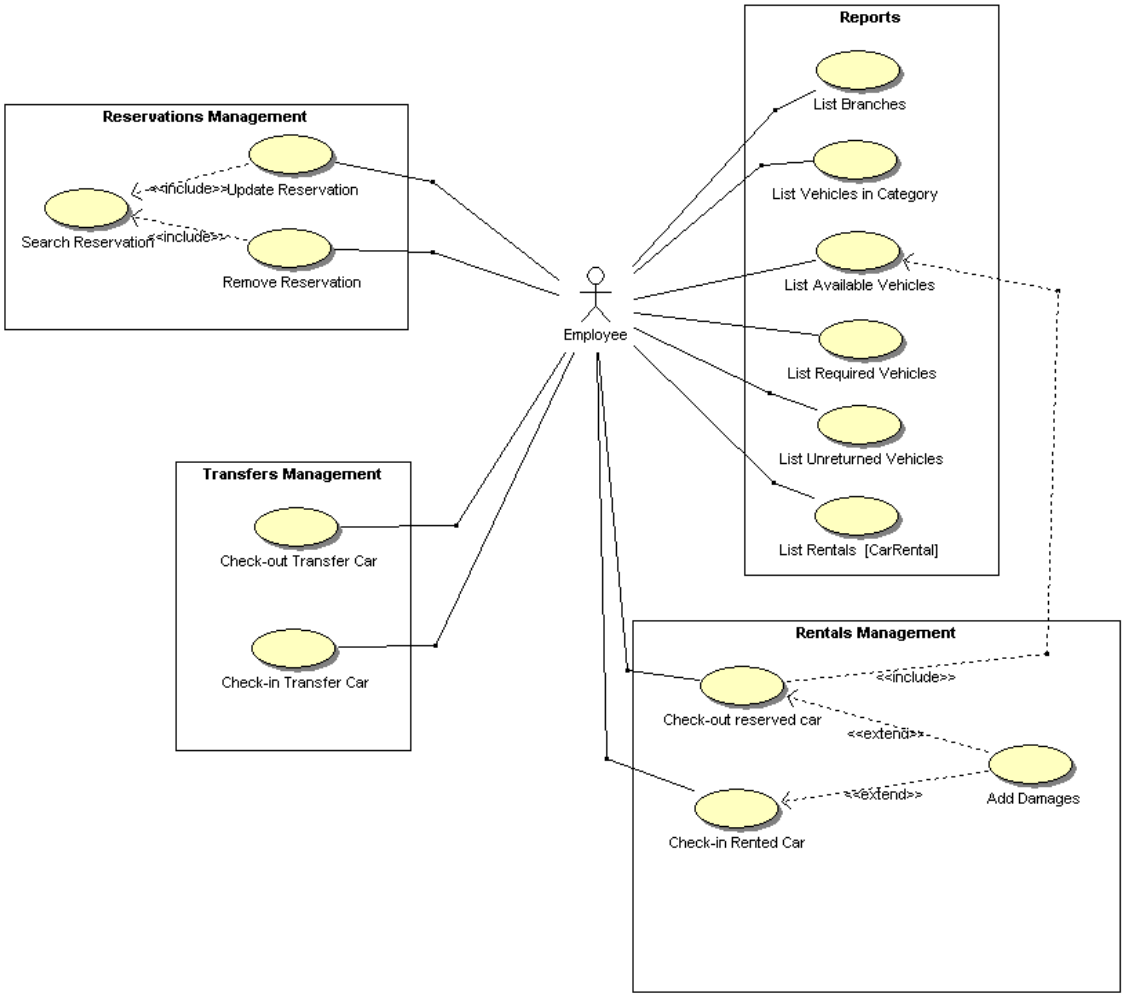


Structured diagrams

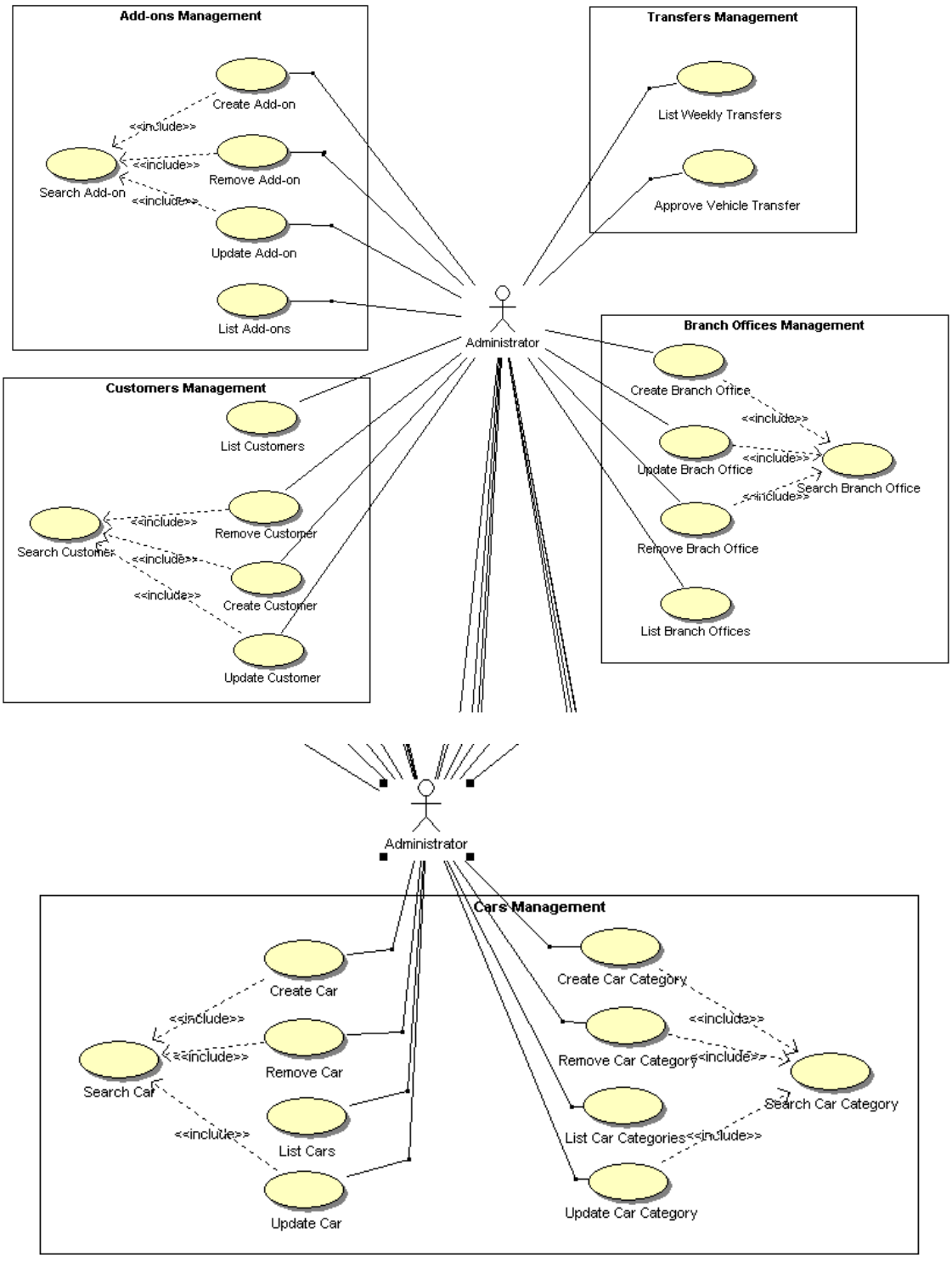
Customer



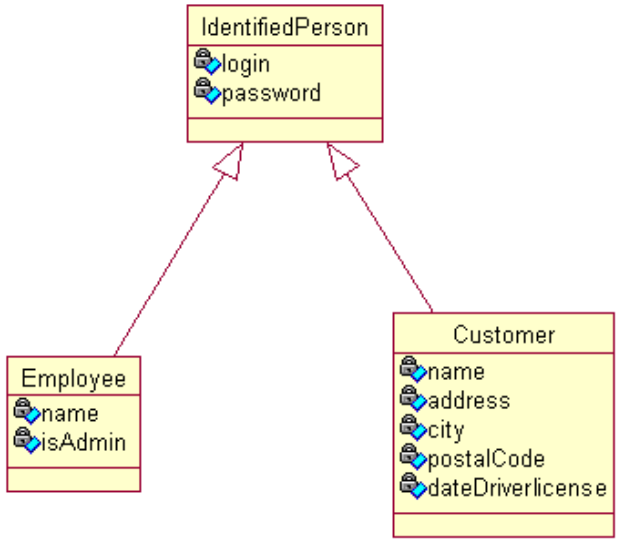
Employee

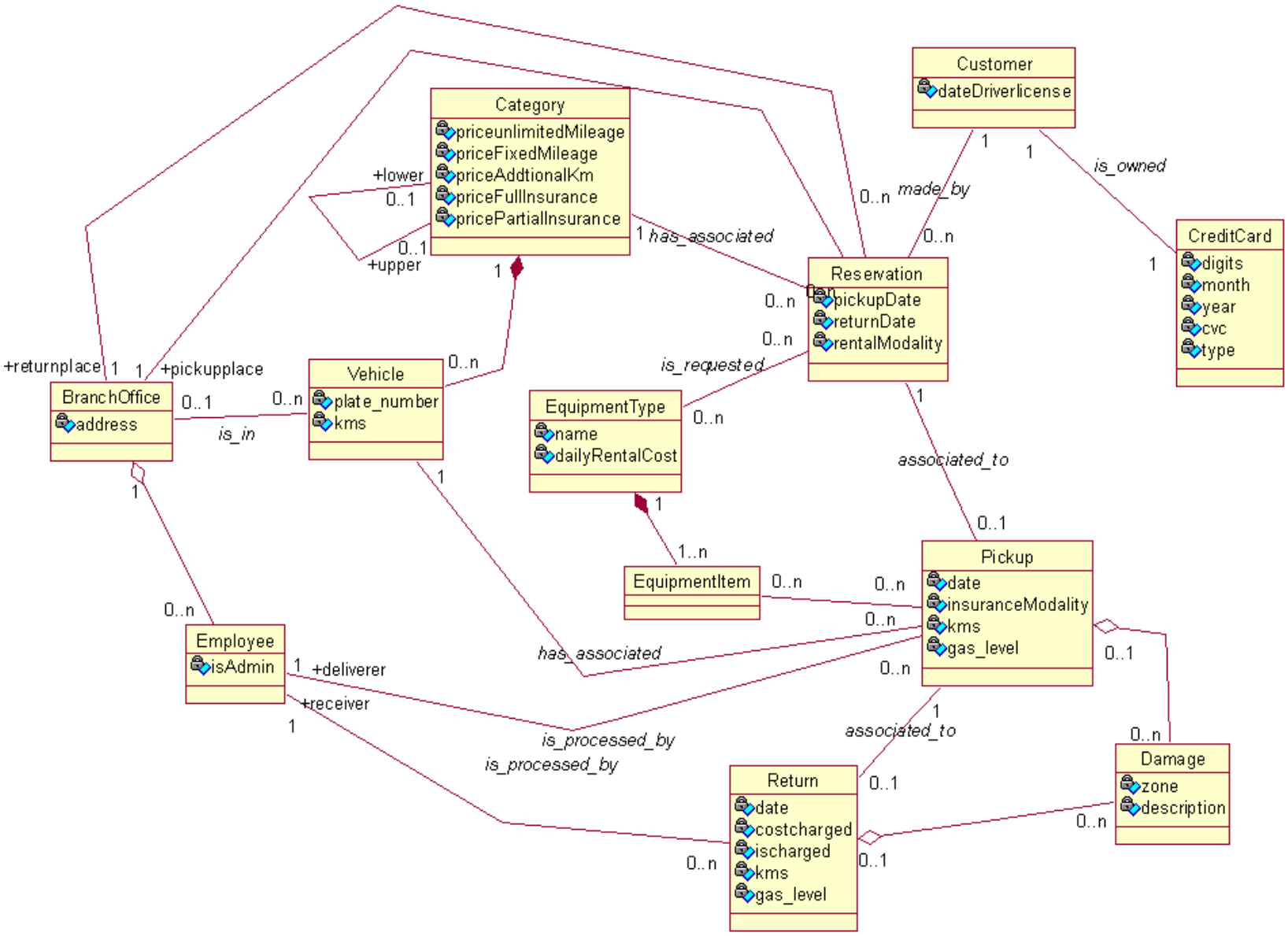

Reports
List Branches
List Vehicles in Category
List Available Vehicles
List Required Vehicles
List Unreturned Vehicles
List Rentals [CarRental]
Reservations Management
<<include>>Update Reservation
Search Reservation
<<include>>
Remove Reservation
Employee
Transfers Management
Check-out Transfer Car
Check-in Transfer Car
Rentals Management
<<include>>
Check-out reserved car
<<extend>>
Add Damages
<<extend>>
Check-in Rented Car

Administrator

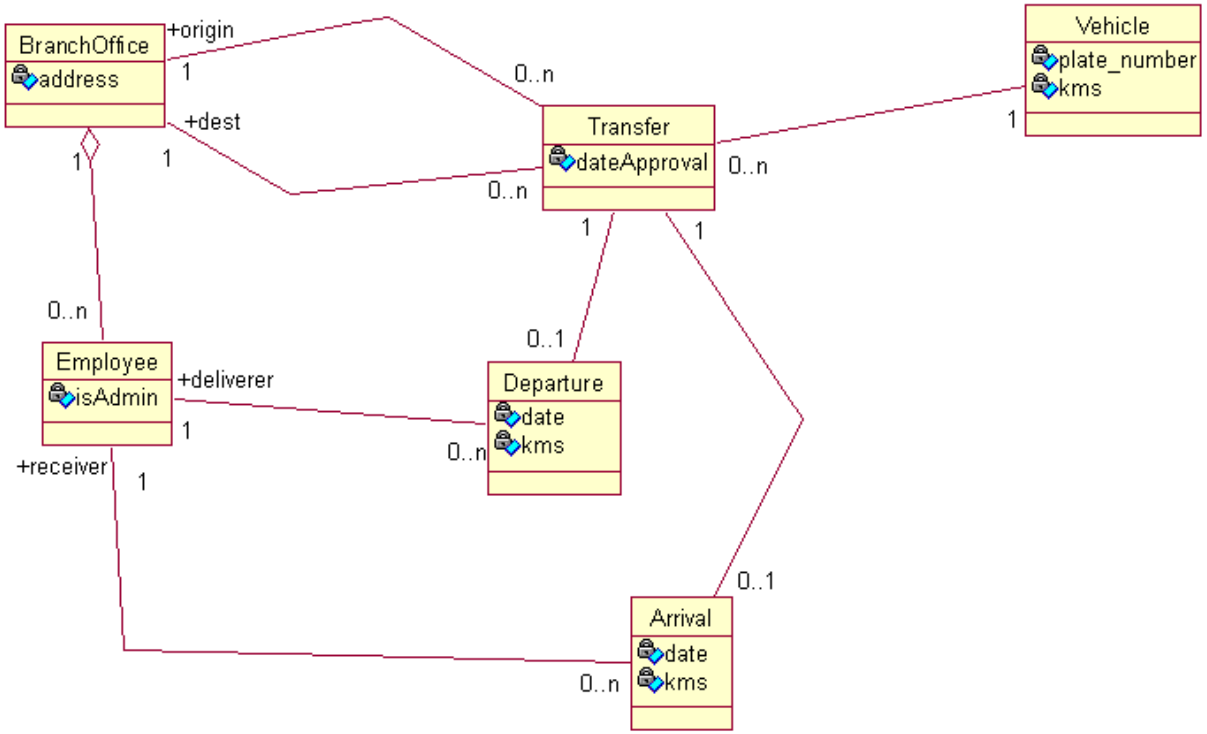


2. UML Class Diagrams




Customer
dateDriverlicense
Category
priceunlimitedMileage
priceFixedMileage
priceAddtionalKm
priceFullInsurance
pricePartialInsurance
CreditCard
digits
month
year
cvc
type
Reservation
pickupDate
returnDate
rentalModality
Vehicle
plate_number
kms
BranchOffice
address
EquipmentType
name
dailyRentalCost
EquipmentItem
Pickup
date
insuranceModality
kms
gas_level
Employee
isAdmin
Return
date
costcharged
ischarged
kms
gas_level
Damage
zone
description
made_by
is_owned
has_associated
is_requested
associated_to
is_in
is_processed_by
+lower
+upper
+returnplace
+pickupplace
+deliverer
+receiver

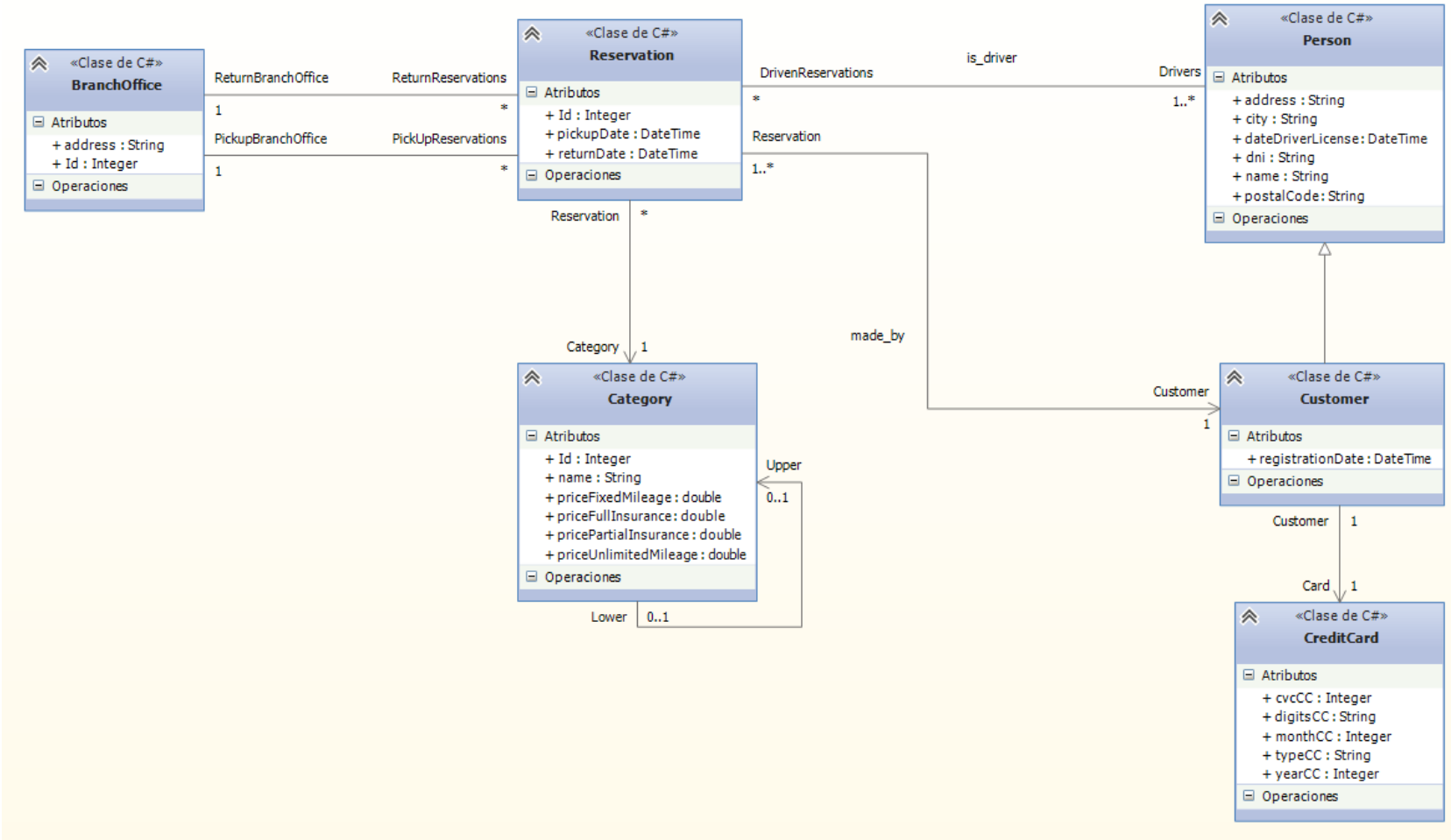
(Vehicle Transfer)



3. Implementation

Taking the previous analysis, it has been developed with Visual Studio 2015 Enterprise a simplified version applying the concepts to be studied in this course (code generation, DAOs, Entity Framework, etc). The implemented solution is available in a compressed file in poliformat and it may be downloaded, compiled and executed. This is a **reference implementation** that may be consulted during the development of the current laboratory assignment.

The developed version corresponds to the following simplified class diagram:


«Clase de C#»
BranchOffice
Atributos
+ address : String
+ Id : Integer
Operaciones
ReturnBranchOffice
1
ReturnReservations
*
PickupBranchOffice
1
PickUpReservations
*
«Clase de C#»
Reservation
Atributos
+ Id : Integer
+ pickupDate : DateTime
+ returnDate : DateTime
Operaciones
DrivenReservations
*
is_driver
Drivers
1..*
Reservation
1..*
Reservation
*
«Clase de C#»
Person
Atributos
+ address : String
+ city : String
+ dateDriverLicense : DateTime
+ dni : String
+ name : String
+ postalCode : String
Operaciones
Category
1
made_by
«Clase de C#»
Category
Atributos
+ Id : Integer
+ name : String
+ priceFixedMileage : double
+ priceFullInsurance : double
+ pricePartialInsurance : double
+ priceUnlimitedMileage : double
Operaciones
Upper
0..1
Lower
0..1
Customer
1
«Clase de C#»
Customer
Atributos
+ registrationDate : DateTime
Operaciones
Customer
1
Card
1
«Clase de C#»
CreditCard
Atributos
+ cvcCC : Integer
+ digitsCC : String
+ monthCC : Integer
+ typeCC : String
+ yearCC : Integer
Operaciones

and it implements the following use cases