

## Kajs Cars

This is mostly a collection of the results of doing the problem domain. I have added a small amount of explanatory text, so i hope you can make sense of it.

## FACTOR

- F: The system should be able to keep track of the cars in current station (the station at which the computer is accessed), this includes cars schedules, price and mileage. The system should also keep track customers who have made a reservation and which price group, customers who are currently renting a car and which one. For corporate customer is should also keep track of their account, which is just the amount if money they are due.
- A: The system should be available through a website where employees can also login to gain access to their extra features.
- C: The description doesn't give much information about this, although we can say that the employees have a varying amount of computer literacy, so nothing to complicated.
- T: As stated the system will be a website. Data should also be backed up on a local server so that the shop should could continue even without internet.
- O: Car, Customer, Station, Agreement, PriceGroup, Schedule
- R: The system is a tool that gives costumers and employees and overview of the cars, and helps in the rental process. Costumers can see which cars are available in the price ranges. Employees can see all the cars and their current status and when they should be services as well as help them fill out a lease properly.

## Class and Event Brainstorm

A quick brainstorm. Not everything was used, and i later realized i need more classes or events.

### Classes

- Customer
- Car
- Station
- Agreement
  - Lease
  - Reservation
- Price group
  - A
  - B
  - C

- D
- Schedule
  - Free
  - Repair
  - Other station
  - Rented

## Events

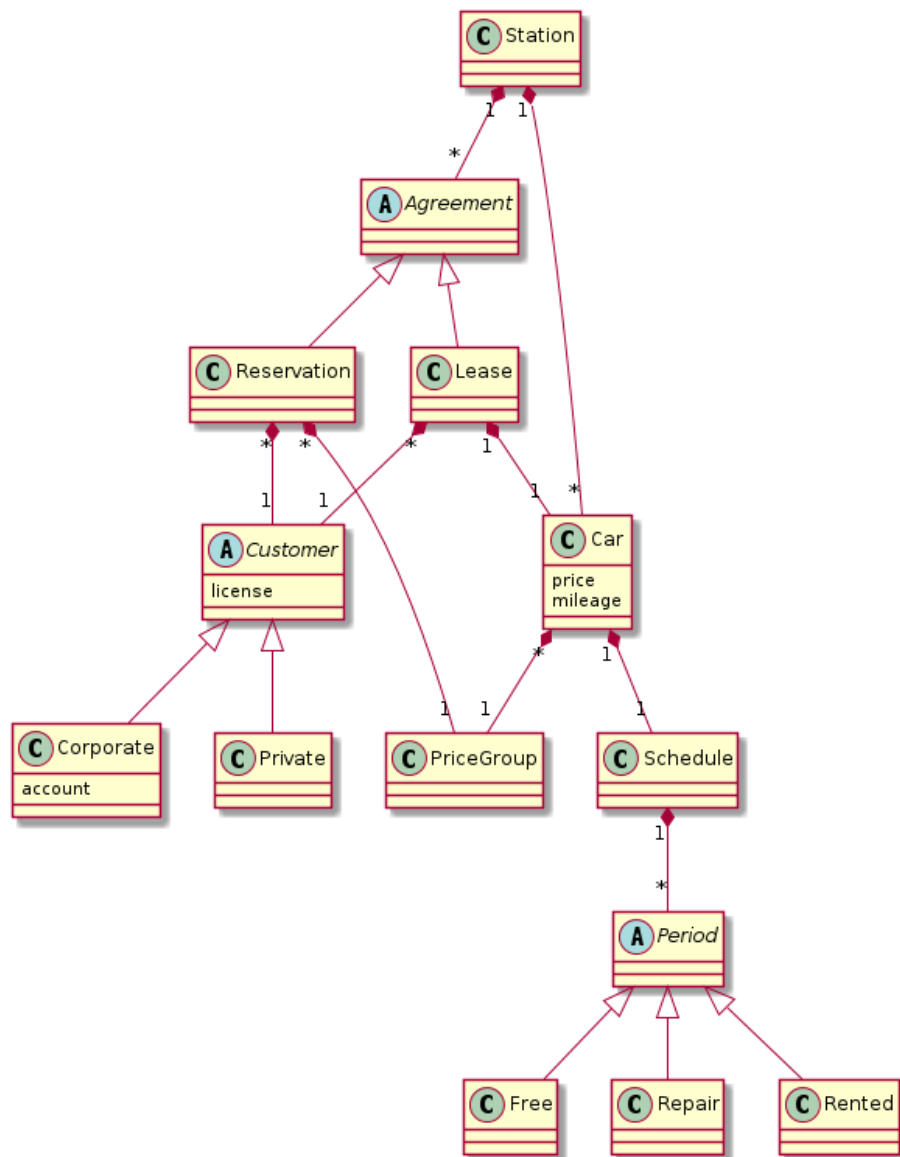
- Move
- Reserve
- Rent
- Deliver
- Repair

## Event Table

	Car	Customer	Station	Agreement	Price Group	Schedule
Move	*		*			
Reserve		*		+	*	
Rent	*	*		*		*
Deliver	*	*				
Pay		*				
Cancel		+		+		
Repair	*					*

## Class Diagram

The main idea is that a car consist of a schedule and a price group. A customer can then make a reservation for a price group. Once they rent a car they can make a lease with a car in the same price group as their reservation.

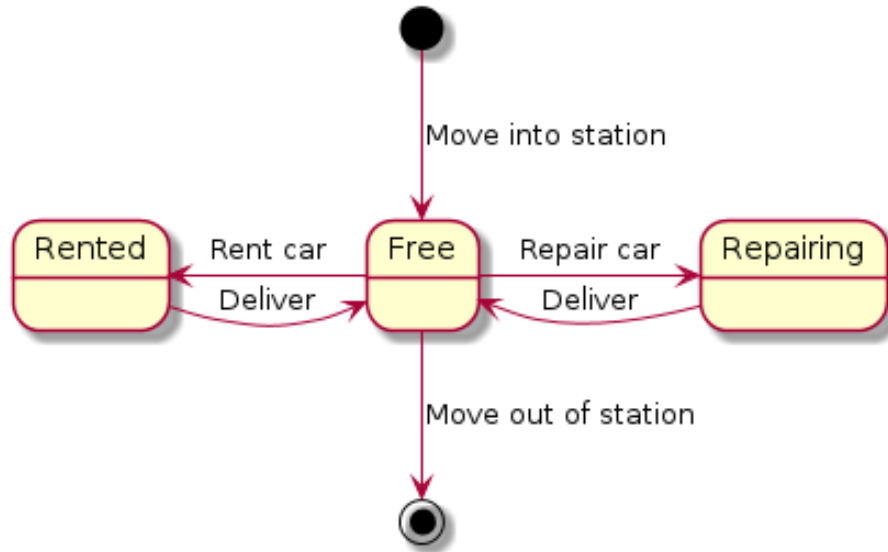


## Behaviour

Behaviour for what i think are the most important classes.

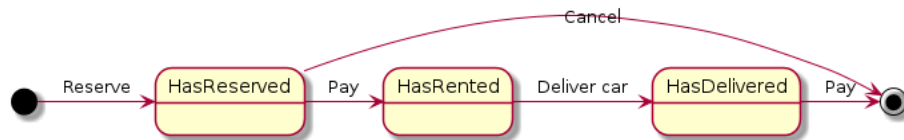
## Car

A car can be created by moving it into the station either from another station or from being bought. The car's "default" state is free, it can then either be rented or be repaired. The car ceases to exist when it is moved out of the station, either by selling it or moving it to another station.



## Private Customer

A customer can make a reservation for a price group. On the day that they need they can pay the deposit and rent the car. They then deliver the car, and lastly resolve any extra payments. Of course a customer can also cancel a reservation.



## Corporate Customer

As for a corporate customer I found it quite hard to model a state where they can have multiple concurrent reservations and rentals, without them being able to reserve and rent cars without necessarily delivering them.

