# Tapio Back End Case Exercise

Develop a feature and the DB architecture to support it from the already existing models and components of the platform. The approximative design of what the front end would look like is provided.

## Requirements

Create a projection tool to allow our experts to design reduction strategies for our clients. The usual carbon report would have several sources attached to it. The tool would allow to register potential modification to those sources.

Implementation should:

- include additional/modified models
- include endpoints (DRF) for specific data
- be implemented using the Django framework
- pay extra care to optimization

## Specifications

- Allow to plan a modification to a `source` (either by applying a ratio to the `value` or changing the `emission_factor` (EF) value)
- Modifications could be in series **e.g.** I first reduce my value by 2 then I change the EF from 42 to 3.14 (switch from diesel to electric let's say)
- Provide information regarding the delta in `total_emission` regarding the `source`
- Provide information regarding the delta in `total_emission` for the `report`
- For `source`s with `lifetime`s (capital goods) the amortization should be considered:
  - `total_emission` are divided along the `lifetime` of the `source`
  - after `lifetime` years the `total_emission` displaied is 0
  - making a modification on a `source` with `lifetime` means that the original `source` could already be amortized **e.g**. bought 1 car (lifetime 5 years) in 2020, if I buy another one in 2022 (with a modification) both will be showed in the total emissions displaied for my modification. If it's in 2028, only the second one will be showed
- When retrieving an information (by `source` or `report`) we should be able to specify a year (attention to `lifetime`)
- We should be able to retrieve data for a range of years (by `source` or `report`) within a dict with the year as a key and the emissions as a value

### Bonus

- We could have several reduction strategies by `report`
- New `source` could be added in reduction strategies
- Modifications could be progressive, the growth should be partially showed when the `year` fits the time the growth started. **e.g.** I'll double my `source` by 2024
- Script to generate a dummy DB

## Resources

### Models

```
class Report(models.Model):
    """
    The Report is the sum of all the emissions. It should be done once a year
    """
    name = models.CharField(max_length=200, blank=True, null=True)
    date = models.DateField()


class Source(models.Model):
    """
    An Emission is every source that generates GreenHouse gases (GHG).
    It could be defined as source x emission_factor = total
    """
    report = models.ForeignKey(Report, on_delete=models.CASCADE, blank=True,
null=True)
    description = models.CharField(max_length=250, blank=True, null=True)
    value = models.FloatField(blank=True, null=True)
    emission_factor = models.FloatField(blank=True, null=True)
    total_emission = models.FloatField(blank=True, null=True, help_text=_("Unit in
kg"))
    lifetime = models.PositiveIntegerField(blank=True, null=True)
    acquisition_year = models.PositiveSmallIntegerField(blank=True, null=True)
```

**Front End**

Stratégie Stats Source Modification