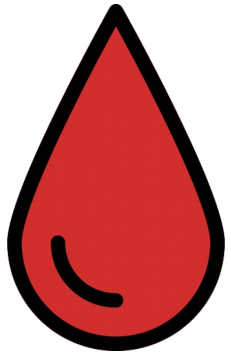
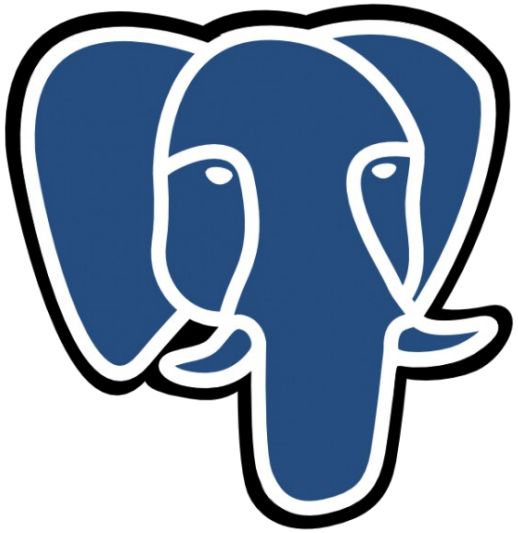


Kan Bankası

Backend

Flask

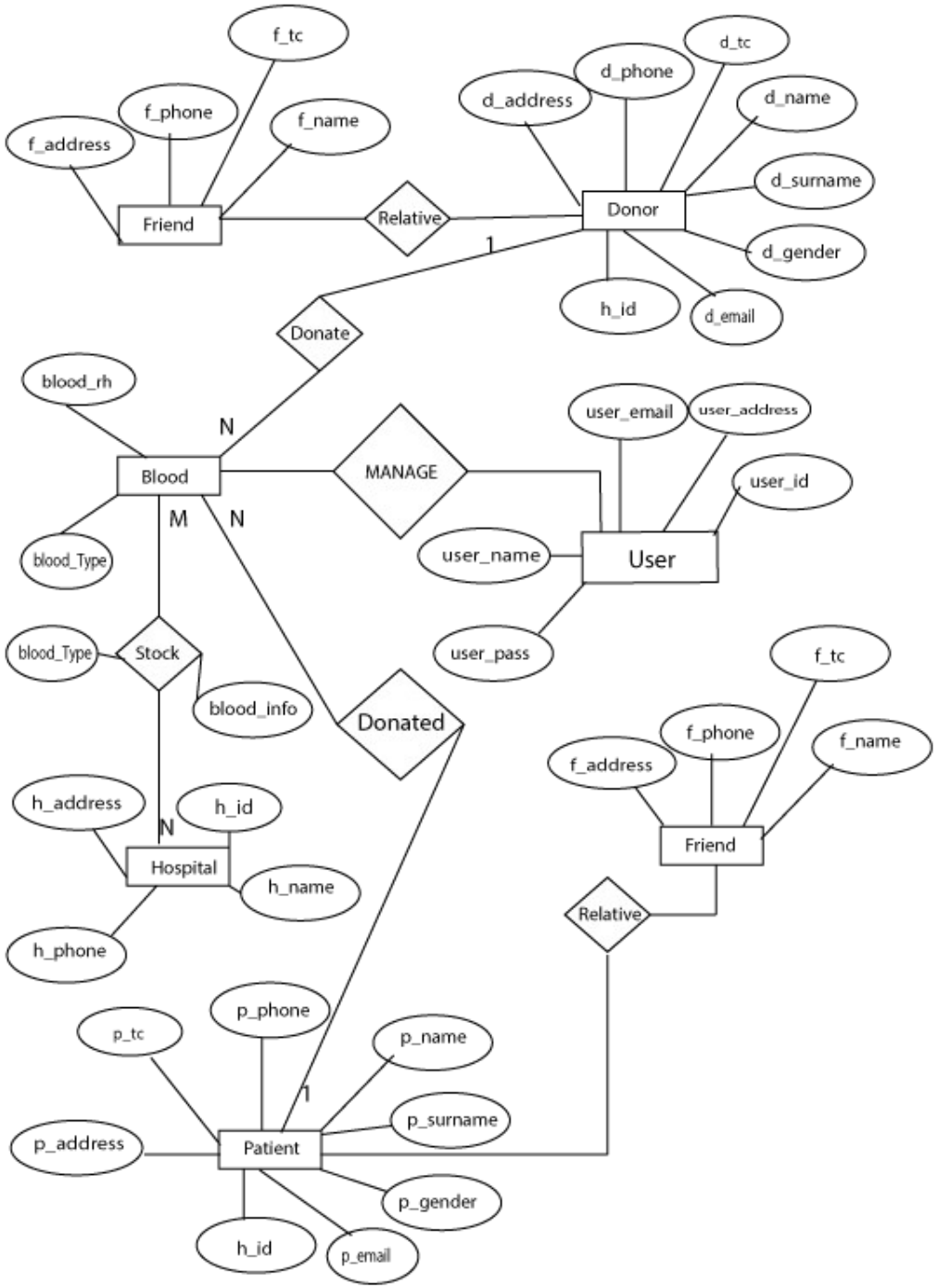


PostgreSQL

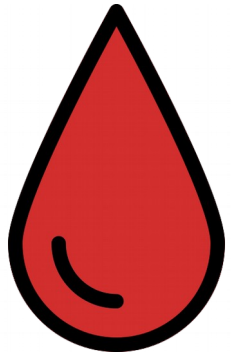
Frontend

Database

Er Diagram

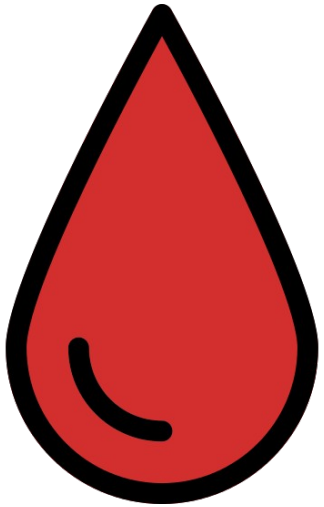


Kan Bankası

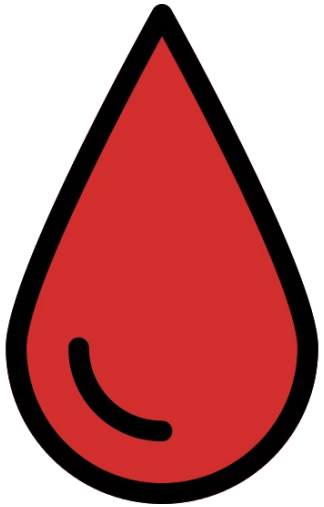


Amaç:

Kan veren/alan bilgisi düzenli olarak saklamak



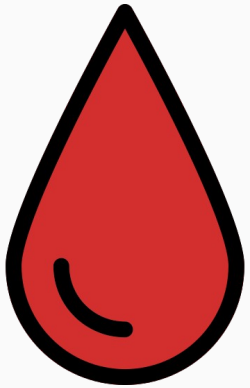
Karşılaştığım Sorunlar



Kan Grubuna Göre Tablonun Güncellenmesi

Ekle Düzenle Sil

○ A+ ○ A- ○ B+ ○ B- ○ AB+ ○ AB- ○ 0+ ○ 0-



Search

	İsim	Soyad	Cinsiyet	Tel	Kan
☐	Recep	Güneş	e	5501502645	3
☐	Salih	Turan	e	123456789	5
☐	Nazlı	Keskin	e	98765432	5
☐	Yeşim	Yüksel	k	451651651	4
☐	Aslı	Bozkurt	k	15151	6

Showing 11 to 20 of undefined rows 10 rows per page

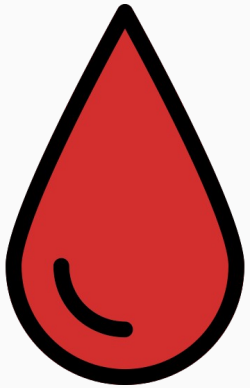
‹ ›

AjaxPrefilter

```
$('input[id="kanarama"]').on('click', function(e) {
    $.ajaxPrefilter(function( options ) {
    options.url = "/hasta/json?kan="+e.target.value
    });
    var $table = $('#table');
    $table.bootstrapTable('refresh');

});
```

Ekle Düzenle Sil

◉ A+ ○ A- ○ B+ ○ B- ○ AB+ ○ AB- ○ 0+ ○ 0-

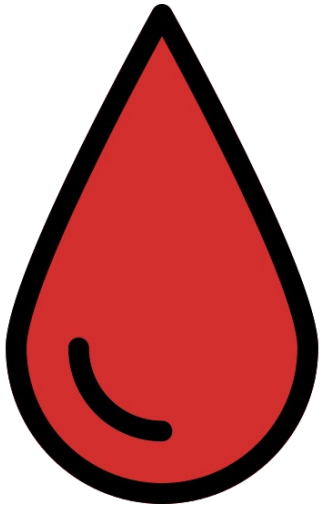


Search

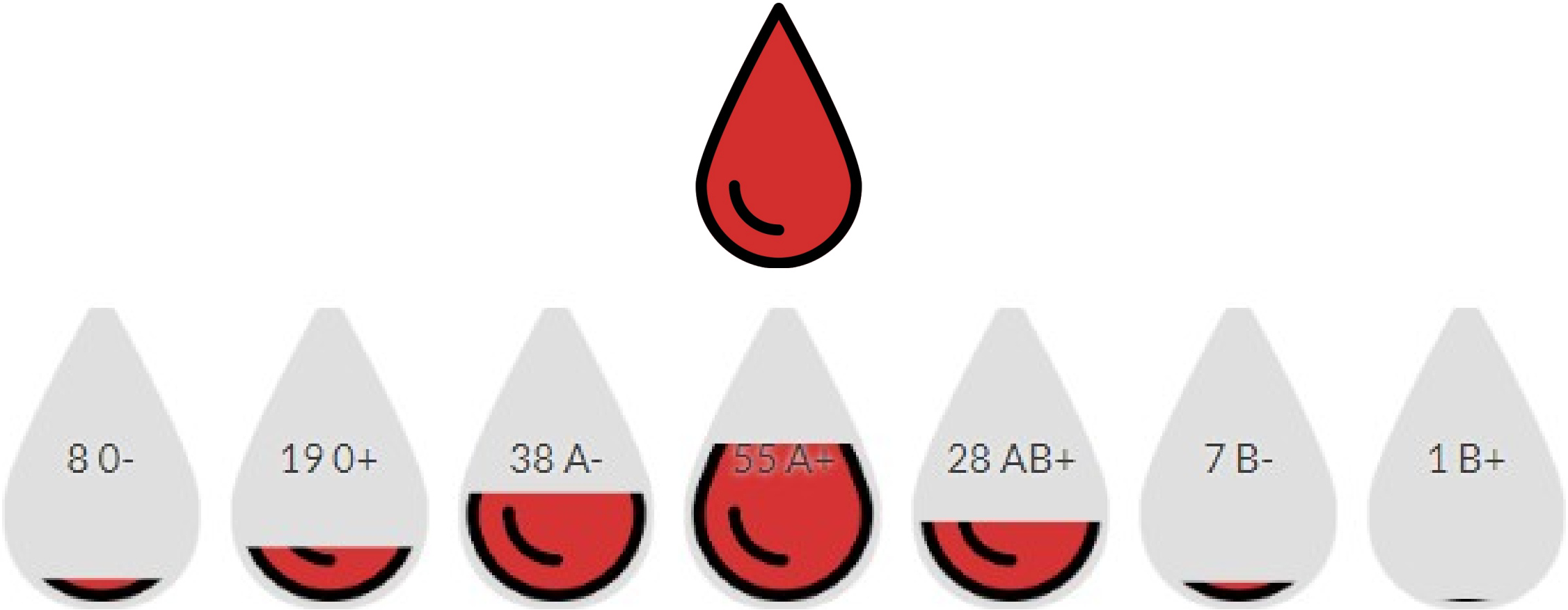
	İsim	Soyad	Cinsiyet	Tel	Kan
☐	A.Arif	Deneme	e	905531118273	1
☐	hasta1	hastasoyad	e	5531118273	1

Showing 11 to 20 of undefined rows 10 rows per page

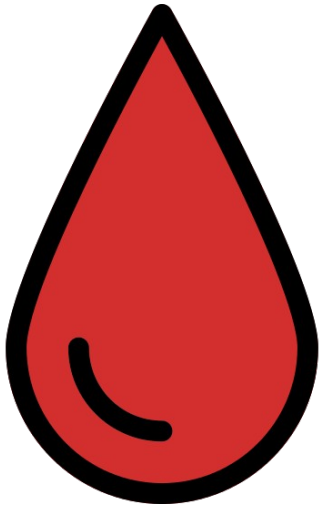
‹ ›



Stok Durumu

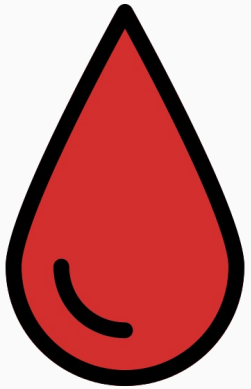


loading-bar.js



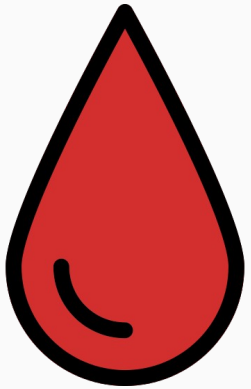
Validation

Validation Backend



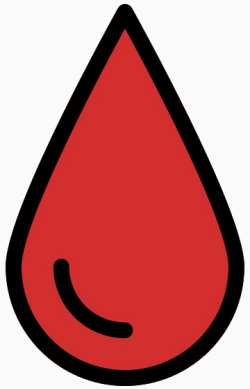
```
from flask_wtf import FlaskForm

from wtforms import StringField,
PasswordField, SubmitField,
BooleanField, RadioField, IntegerField

from wtforms.validators import
DataRequired, Length, Email,
EqualTo, NumberRange
```

Validation Backend

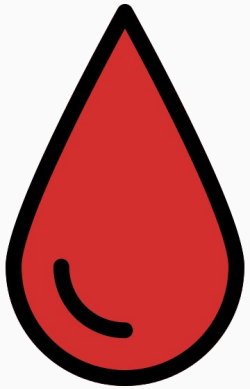


```
class Donor(FlaskForm):
  isim =
StringField('isim',validators=
    [DataRequired('isim gerekli'),
      Length(message=‘Uyari’,
      min=3, max=20)])
  tel  = IntegerField('tel',
validators=[DataRequired(‘Zorunlu
alan’)])
```

Validation Backend



```
if a=='create':
    form = Donor()
    if form.validate_on_submit():
      data=form.data
      return donorApi.create(data)
    else:
      return jsonify(form.errors),500
```



Validation Frontend

KAN BANKASI AnaSayfa Hakkında Ha

Donor

Ekle Düzenle Sil

	İsim	Soyad
☐	DENEME	DENEME
☐	ali	bak
☐	Crazy	Bilal
☐	Ahmet	Veli
☐	Asya	Yalçın
☐	TEST410	TEST303
☐	DENEME	DENEME
☐	DENEME	DENEME
☐	DENEME333	DENEME333
☐	TEST275	TEST32

Ekle

Donor Yakını

isim

isim gerekli

soyisim

soyisim gerekli

tel

tel gerekli

adres

adres gerekli

tc

tc hatali

Kan Grubu

Aç

hastane

email

Bu alan Zorunlu!

cinsiyet

- ○ erkek
- ○ kadin

Not a valid choice ---

kan

○ A+ ◉ A- ○ B+ ○ B- ○ AB+ ○ AB- ○ 0+ ○ 0-

KAPAT Kaydet

Design Patterns

Singleton (Creational Patterns)
Adapter (Structural Patterns)
Mediator (Behavioral Patterns)

Singleton

```
class MetaSingleton(type):
    _instance = None
    def __call__(self):
        if self._instance is None:
            self._instance =
super().__call__()
        return self._instance
```

Singleton

```
class Database(metaclass=MetaSingleton):
    def __init__(self):
        self.con=None
        if self.con is None:
            self.con = psycopg2.connect(...)
            self.con.set_client_encoding('UTF8')
            ...
```

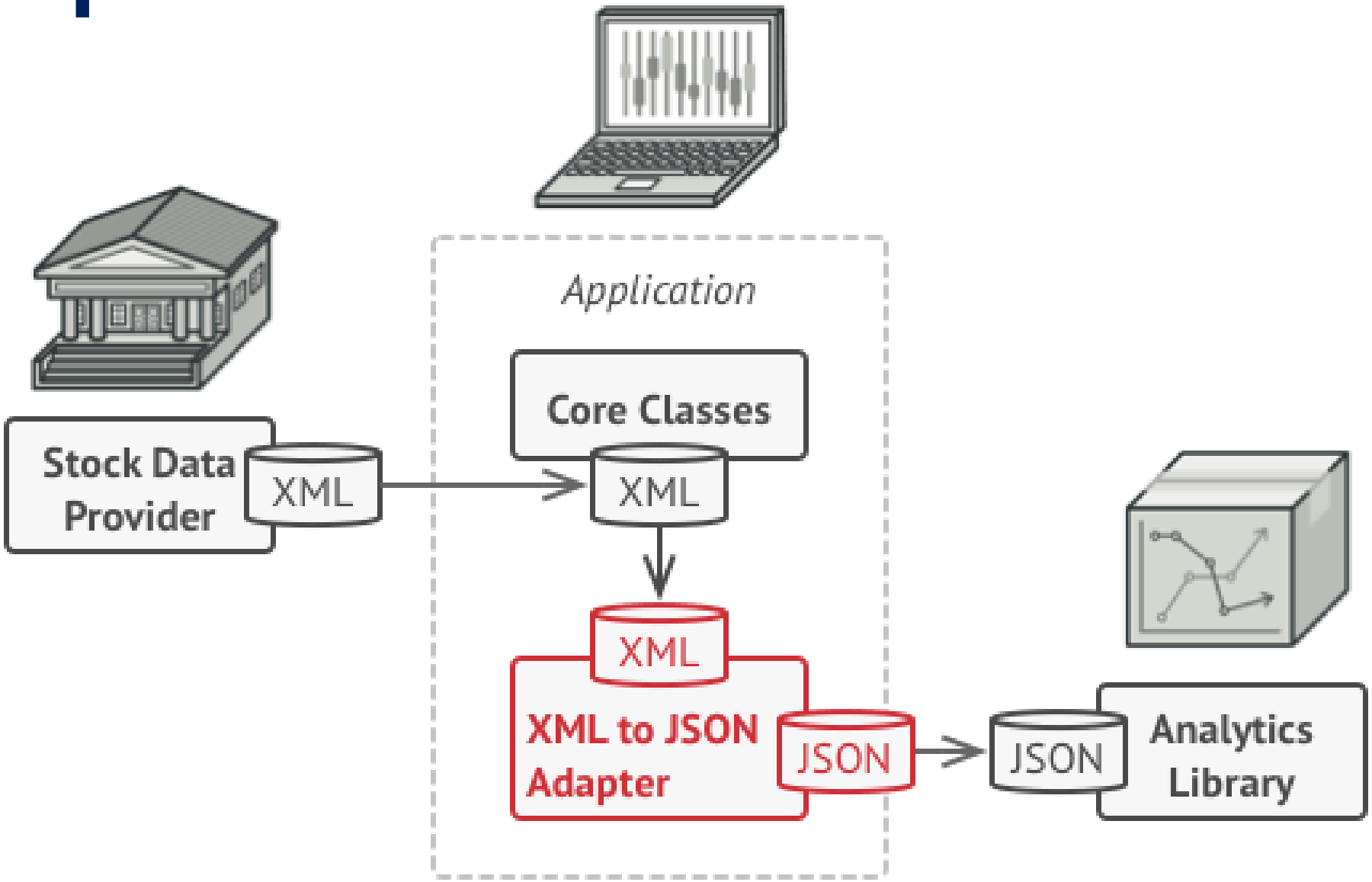
1

2

Singleton

```
def sqlrun(sorgu):
  db1 = Database()
  return
db1.execute(sorgu)
```

Adapter


Application
Core Classes
Stock Data Provider
XML
XML
XML
XML to JSON Adapter
JSON
JSON
Analytics Library

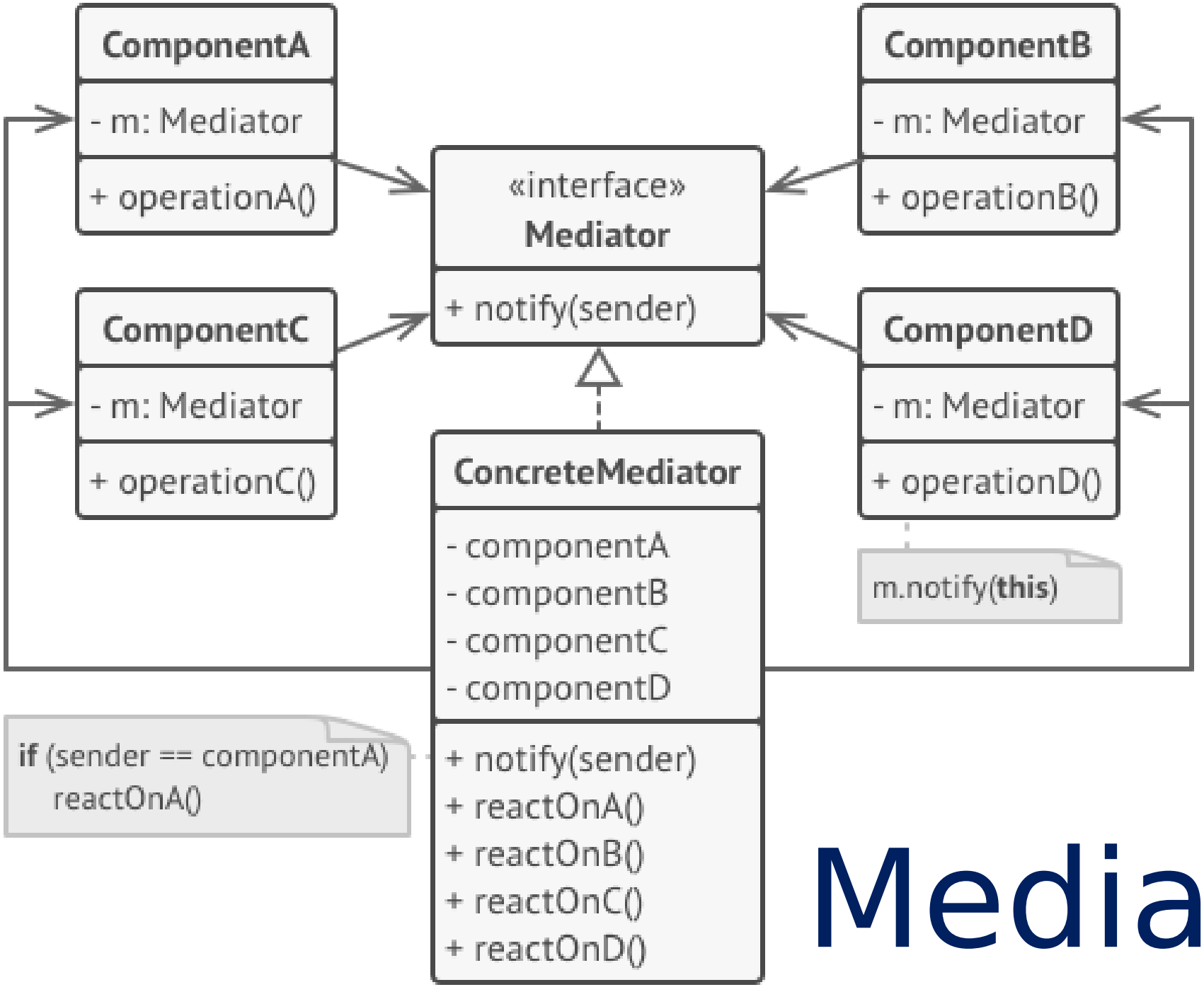
Adapter

```
from kanbankasi.sqlrun import
sqlrun

class Target:
    #interface
    def
list2json(self,sorgu):
        return sqlrun(sorgu)
```

```
class JsonDict(Target):
    def __init__(self, sqlrun):
        self.sqlrun = sqlrun

    def list2json(self,sorgu):
     rows=self.sqlrun(sorgu)
     return json.dumps(
     [dict(ix) for ix in rows]
     )
```

Adapter

```
def client_code(target,sorgu):
    print(target.list2json(sorgu))


target = Target()
client_code (target,"select * from user1")
adapter = JsonDict(sqlrun)
client_code (adapter,"select * from user1")
```


ComponentA
- m: Mediator
+ operationA()
ComponentB
- m: Mediator
+ operationB()
«interface»
Mediator
+ notify(sender)
ComponentC
- m: Mediator
+ operationC()
ComponentD
- m: Mediator
+ operationD()
m.notify(this)
ConcreteMediator
- componentA
- componentB
- componentC
- componentD
+ notify(sender)
+ reactOnA()
+ reactOnB()
+ reactOnC()
+ reactOnD()
if (sender == componentA)
reactOnA()

Mediator

Mediator

```
class BaseComponent:
    def __init__(self, mediator = None):
        self._mediator = mediator
    @property
    def mediator(self):
        return self._mediator
    @mediator.setter
    def mediator(self, mediator):
        self._mediator = mediator
```

Mediator

```
class ComponentDonor(BaseComponent):
    def __init__(self,isim):
        self.name=isim
    def list(self,offset,limit):
        return self.mediator.crud(self,
"list",offset=offset,limit=limit)
    def create(self,data):
        return self.mediator.crud(self, "create",data=data)
    def update(self,data):
        return self.mediator.crud(self, "update",data=data)
    def delete(self,data):
```

Mediator

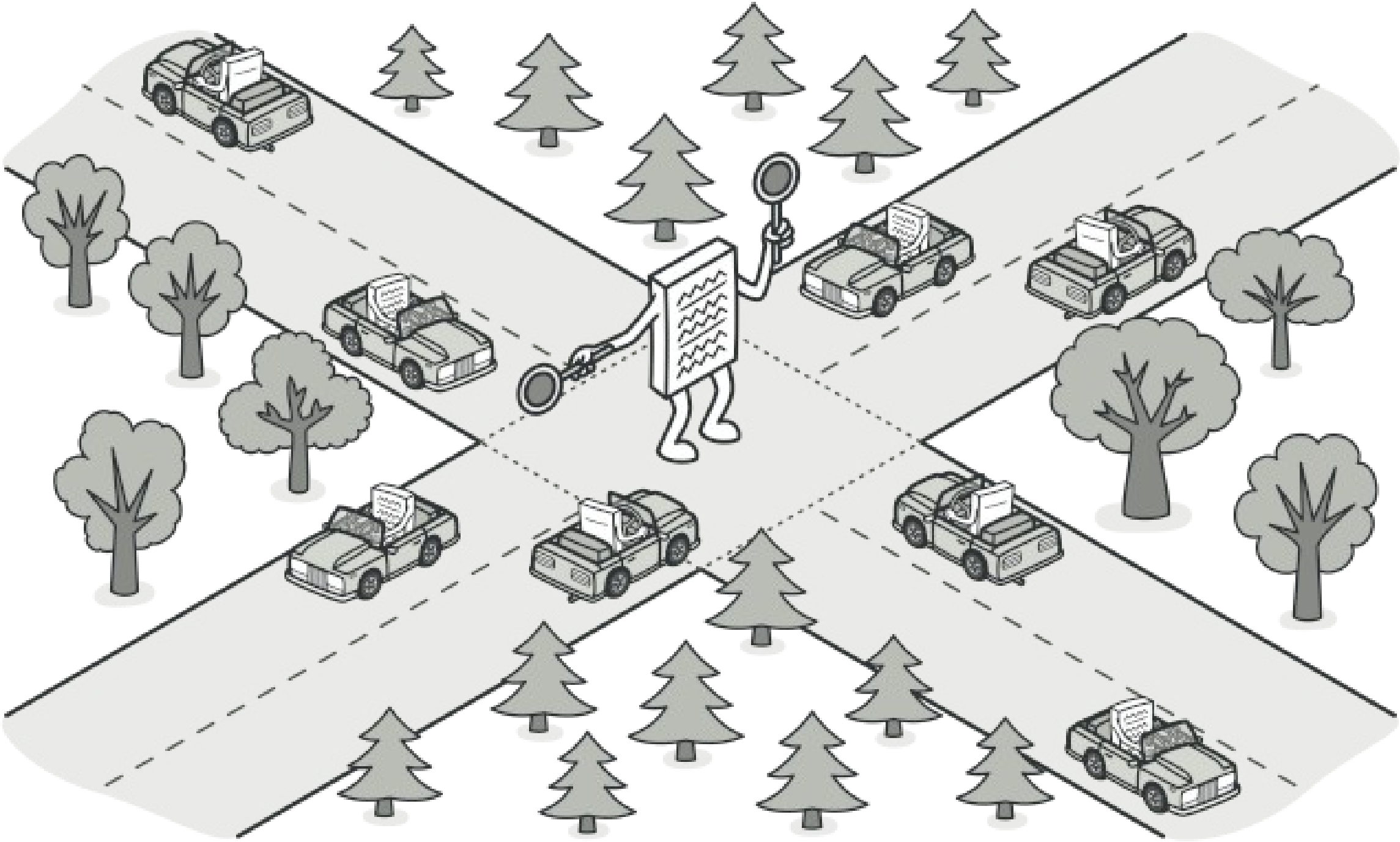
```
def _list(self,offset,limit):
    sorgu="select * ... offset "+ str(offset) +" rows fetch first
... "
    return rowsjson(sorgu)
def _create(self,data):
    pass
def _update(self,data):
    pass
def _delete(self,data):
    pass
```

Mediator

```
class ConcreteMediator(Mediator):
    def __init__(self, donor, hasta):
        self._component1 = donor
        self._component1.mediator = self
        self._component2 = hasta
        self._component2.mediator = self
```

Mediator

```
def crud(self, sender, event, **param):
    if event == "list":
        if sender.name=="donor":
            return self._component1._list(param['offset'],param['limit'])
        if sender.name=="hasta":
            return self._component2._list(param['offset'],param['limit'])
    elif event == "create":
        if sender.name=="donor":
            data=param['data']
            return self._component1._create(data)
        if sender.name=="hasta":
            data=param['data']
            return self._component2._create(data)
```



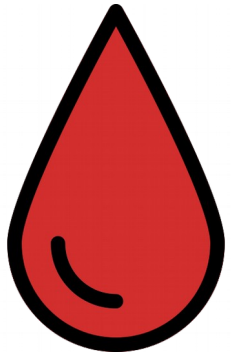
Mediator

```
import kanbankasi.ApiMediator as
api


donorApi=api.ComponentDonor('d
onor')
hastaApi=api.ComponentHasta('ha
sta')
```

Mediator

```
if a=='update':
   data =request.data
   data=json.loads(data)
   return donorApi.update(data)
if a=='delete':
   data =request.data
   data=json.loads(data)
   return donorApi.delete(data)
```

Teşekkürler



github.com/
mzuvin/kanbagisi