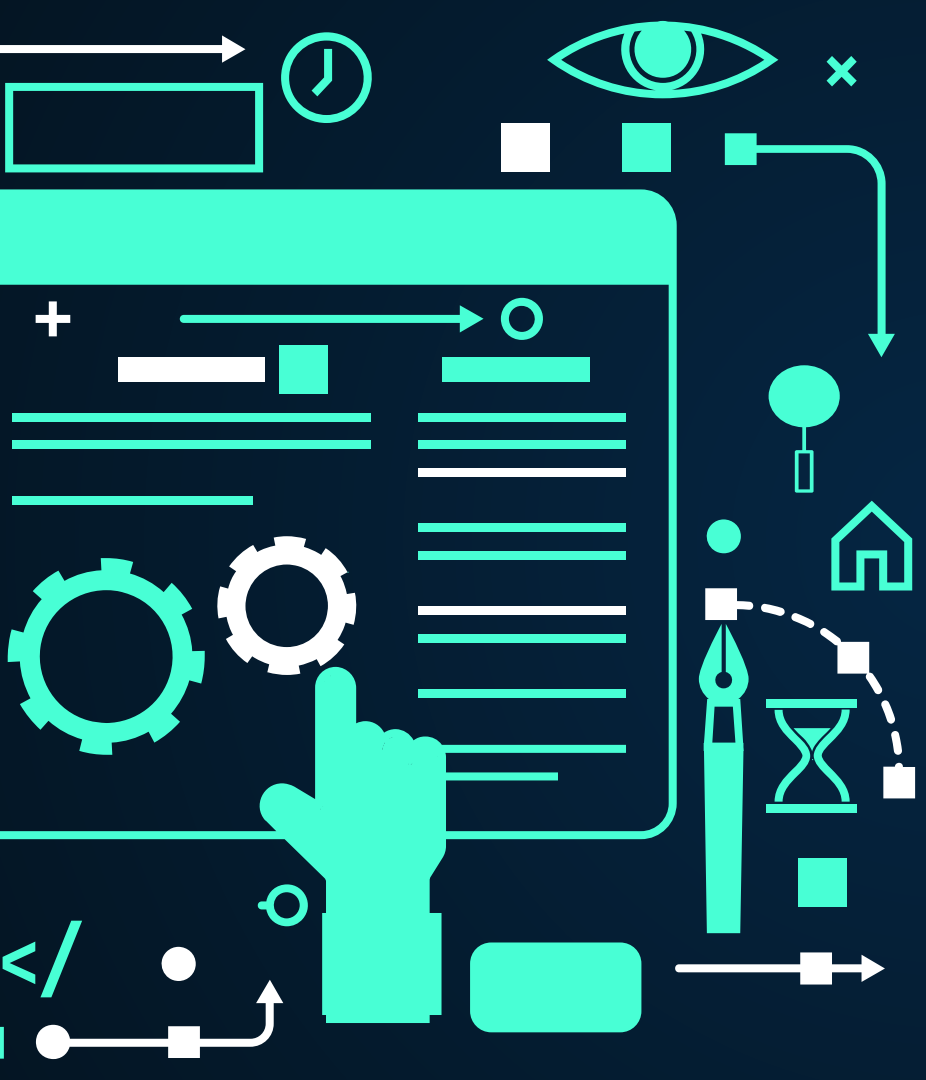


FITSIZE

Teva GEITNER
Jules GONZALEZ
Yaël PARA

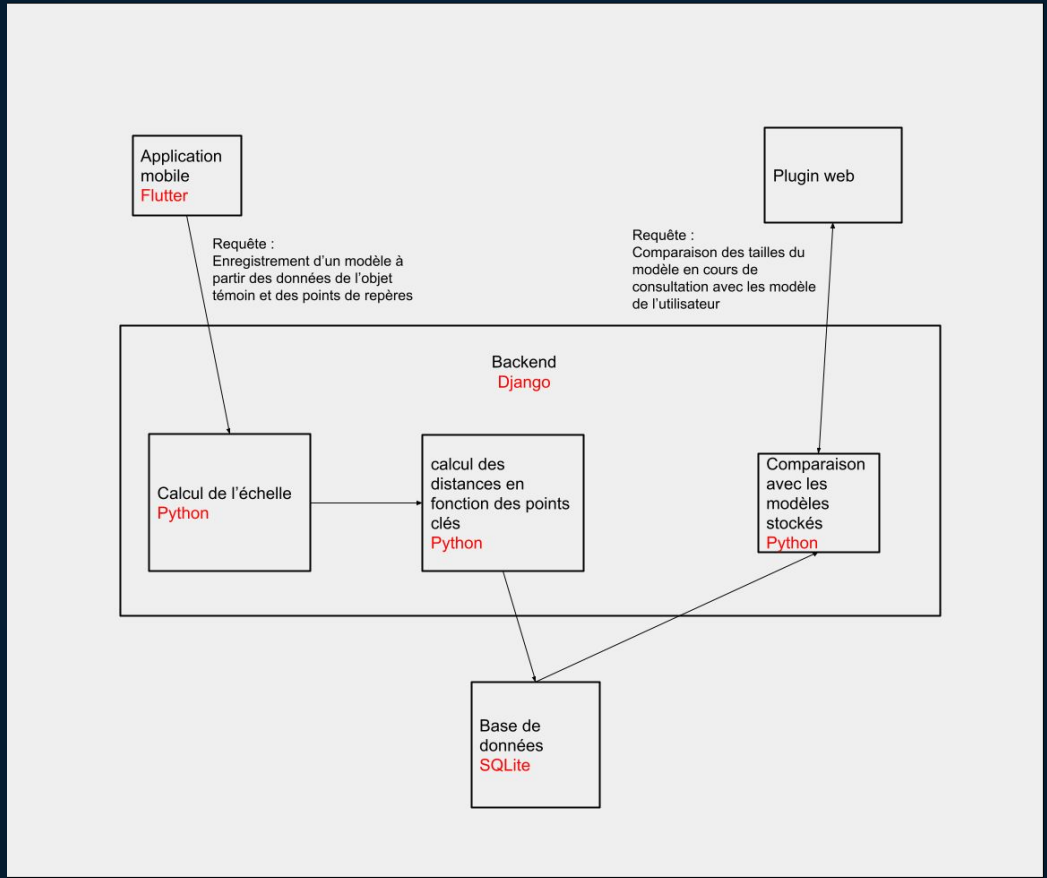
Porteurs du projet: Fidèle EYA'A - Jean-Marc INIKO
Tuteur : Didier DONSEZ

TABLE OF CONTENTS

CONTEXTE

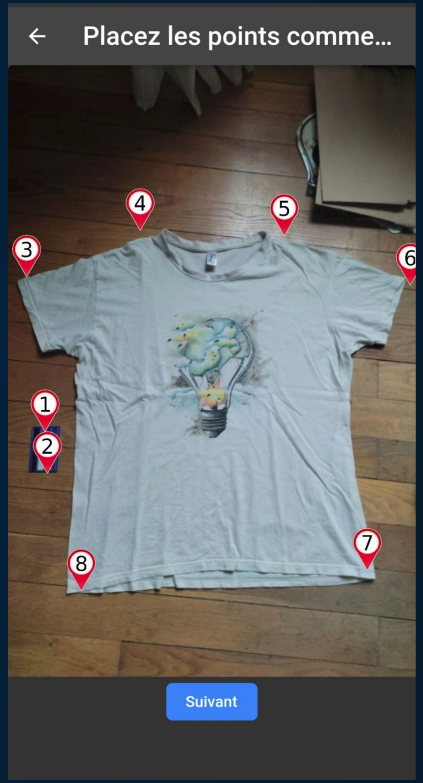
FIT SIZE

ARCHITECTURE TECHNIQUE



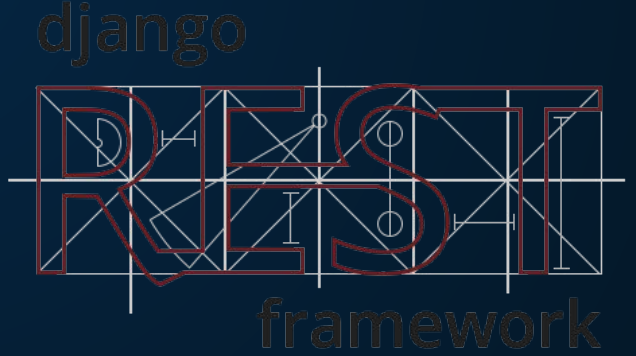
RÉALISATIONS TECHNIQUES

FRONTEND

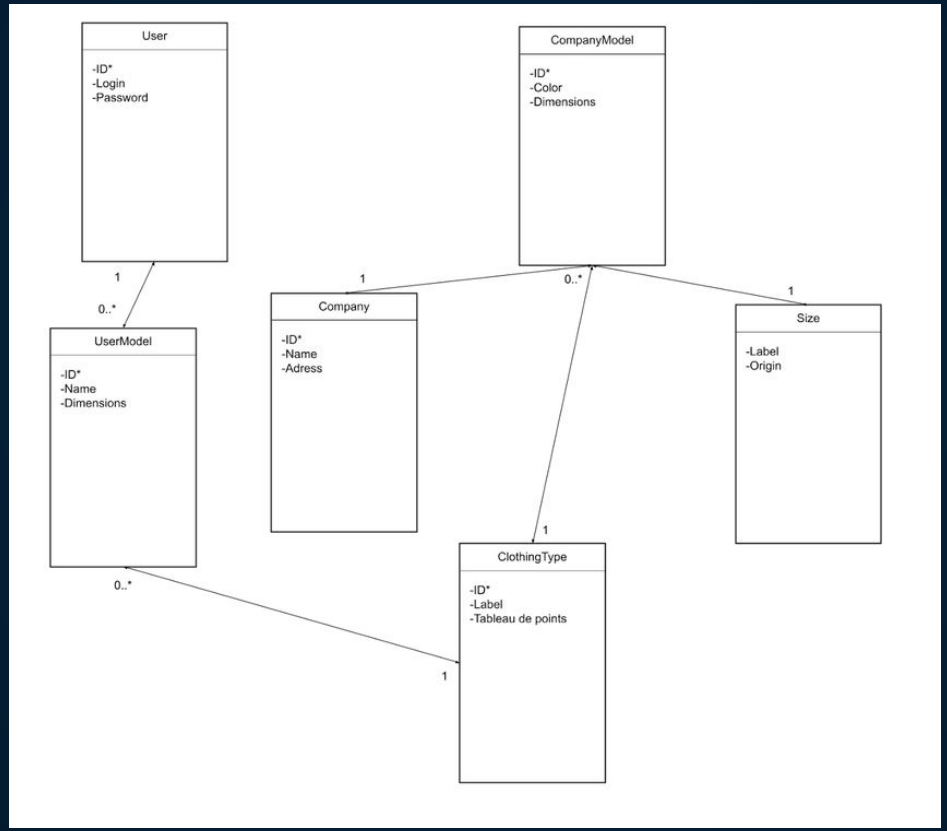


RÉALISATIONS TECHNIQUES

BACKEND

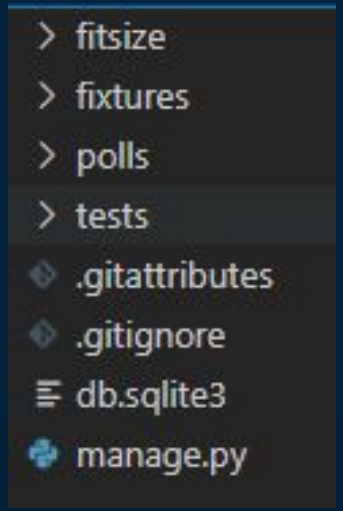


RÉALISATIONS TECHNIQUES



RÉALISATIONS TECHNIQUES

ARCHITECTURE



RÉALISATIONS TECHNIQUES

MODEL

```python
class UserModel(models.Model):
    name = models.CharField(max_length=100, blank=True)
    dimensions = models.CharField(max_length=100, blank=True)
    user = models.ForeignKey(
        'User', on_delete=models.CASCADE, blank=True, null=True)
    clothingtype = models.ForeignKey(
        'ClothingType', on_delete=models.CASCADE, blank=True, null=True)

    def __str__(self):
        return "USERMODEL : name : " + self.name
```

RÉALISATIONS TECHNIQUES

SERIALIZER

```python
class UserModelSerializer(ModelSerializer):

    class Meta:
        model = UserModel
        fields = ('id', 'name', 'dimensions', 'user', 'clothingtype')

    def to_representation(self, instance):
        self.fields['user'] = UserSerializer(read_only=True)
        self.fields['clothingtype'] = ClothingTypeSerializer(read_only=True)
        return super(UserModelSerializer, self).to_representation(instance)
```

RÉALISATIONS TECHNIQUES

ROUTER ET VIEWSET

```
router = routers.SimpleRouter()
router.register(r'user', UserViewSet)
router.register(r'usermodel', UserModelViewSet)
router.register(r'company', CompanyViewSet)
router.register(r'companymodel', CompanyModelViewSet)
router.register(r'size', SizeViewSet)
router.register(r'clothingtype', ClothingTypeViewSet)

urlpatterns = [
    re_path('^', include(router.urls)),
]
```

```
class UserModelViewSet(ModelViewSet):
    serializer_class = UserModelSerializer
    queryset = UserModel.objects.all()

    @action(methods=['post'], detail=False, url_path='savedimensions',url_name="Save dimensions")
    def saveFromKeyPoints(self, request, *args, **kwargs) :
```

RÉALISATIONS TECHNIQUES

TESTS ET FACTORY BOY

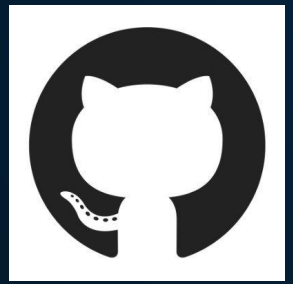
```
Found 31 test(s).
Creating test database for alias 'default'...
System check identified no issues (0 silenced).
...............................
----------------------------------------------------------------------
Ran 31 tests in 0.368s

OK
Destroying test database for alias 'default'...
```

```python
def testDelete(self):
    """DELETE to destroy a UserModel."""
    usermodel = UserModelFactory()
    usermodel.user.save()
    usermodel.clothingtype.save()
    usermodel.save()
    self.assertEqual(UserModel.objects.count(), 1)
    response = self.client.delete(
        self.list_url + str(usermodel.id) + '/')
    self.assertEqual(response.status_code, status.HTTP_204_NO_CONTENT)
    self.assertEqual(UserModel.objects.count(), 0)
```

GESTION DE PROJET

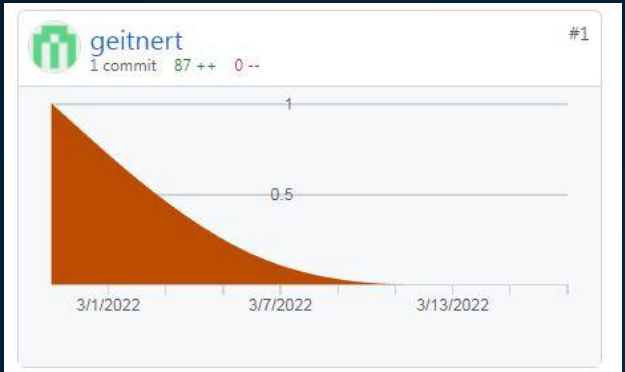
Extrait du backlog

Application Flutter	Pages d'aceuilles de l'application	3
	Menu de choix d'un objet témoins et explication du procédé a l'utilisateur	5
	Prise de photo sur l'application (Habit + objet témoin)	9



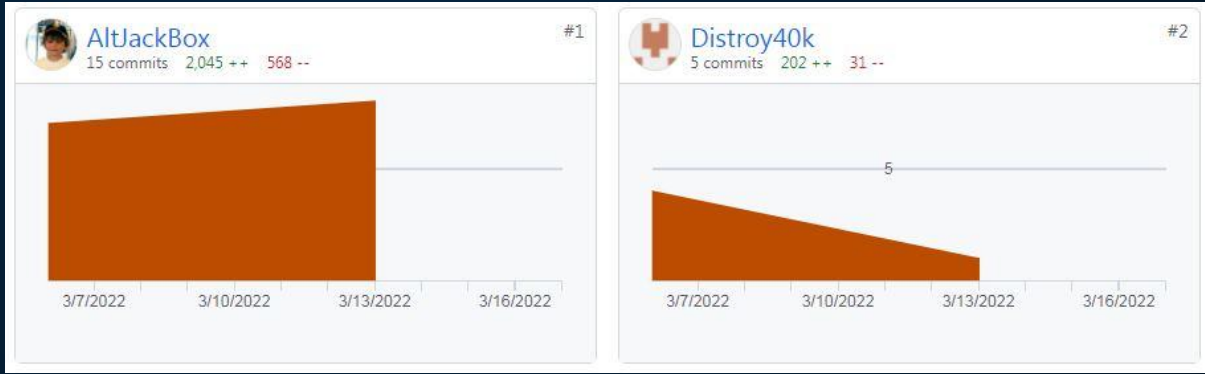
OUTILS

MÉTRIQUES LOGICIELS



Frontend

Backend



CONCLUSION

MERCI POUR VOTRE ATTENTION
DES QUESTIONS ?