



BRIEF 1B 11/2022

DOCUMENTATION TECHNIQUE

Idriss KOURBANHOUSSEN (05)

Saia RALANDSON (12)

Ugo FOK-YIN (03)

DIA AFPAAR

Lien GitHub du projet: [DevIA05/1B \(github.com\)](https://github.com/DevIA05/1B)

1. Contexte du projet	2
2. Objectifs du projet	3
3. Ressources et outils utilisés	3
La plateforme GitHub permet aux programmeurs	5
informatiques de collaborer librement sur des	5
projets de code et nous a permis de planifier le projet	5
avec un scrumboard.	5
4. Maquette et architecture	5
5. Base de données	6
6. Arborescence du projet	7
7. Aperçu des fonctionnalités	8
A) Identification:	8
A.1 Redirection vers les pages d'accueil	9
A.2 Accès Refusé	9
B.1 Page d'accueil:	10
B.2 Gestionnaire de session:	10
B.3 Import massif de quizz:	12
Schéma explicatif de l'upload:	16
B.4 Ajouter du personnel	16
C)Fonctionnalités du collaborateur:	17
C.2 Réalisation du quiz d'évaluation	17
C.2.1 Coté Serveur	18
initQuiz()	18
extractData()	18
nextQuestion()	18
score()	18
C.2.2 Coté client	19
evalQuiz.html	19
updateTag	19
sendData()	19
D) Score	20
7. Conclusion et To do list	20

1.Contexte du projet

Vous êtes dév IA junior depuis peu dans cette entreprise et le service RH vous interpelle pour le projet QUIZ à peine amorcé par le SI :

Une application techniquement vieillissante (trop contraignante dit le SI) mais très efficace doit hélas être abandonnée. Le service RH souhaite une version moderne de cet outil sur son intranet, mais conserver de l'ancienne appli un maximum de data.

Les informations des salariés collaborateurs (utilisateurs de cette application) existent bien entendu sur notre logiciel RH.

2. Objectifs du projet

Objectif:

Notre objectif pour ce projet était de réaliser dans un premier temps une application permettant de mettre en place une session quizz, d'y répondre, ainsi qu'ajouter des utilisateurs et des quizz.

3. Ressources et outils utilisés

Framework Django:

Django est un framework de développement d'application web open source qui s'inspire du principe MVT.

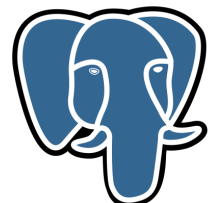
[Documentation de Django](#) | [Documentation de Django](#)



PostgreSQL:

PostgreSQL est un SGBD orienté objet puissant et open/source qui est capable de prendre en charge en toute sécurité les charges de travail de données les plus complexes.

[Documentation PostgreSQL 15.0](#)



PostgreSQL

HTML & CSS:



Bootstrap:

Bootstrap.css est un framework CSS qui organise et gère la mise en page d'un site web. Alors que le HTML gère le contenu et la structure d'une page web, le CSS s'occupe de la mise en page du site. Pour cette raison, les deux structures doivent coexister pour effectuer une action particulière.

[Get started with Bootstrap · Bootstrap v5.2](#)



JavaScript:

JavaScript (souvent abrégé en « JS ») est un langage de script léger, orienté objet, principalement connu comme le langage de script des pages web. Il permet d'interagir avec la page.



jQuery:

jQuery est une bibliothèque JavaScript libre et multiplateforme créée pour faciliter l'écriture de scripts côté client dans le code HTML des pages web

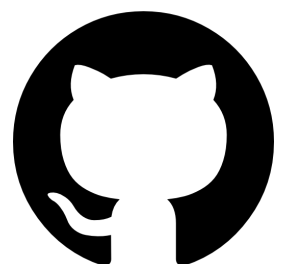


AJAX:

AJAX (*Asynchronous JavaScript + XML*) n'est pas une technologie en soi, mais un terme désignant une « nouvelle » approche utilisant un ensemble de technologies existantes dont la particularité est de permettre d'effectuer des requêtes au serveur web

GitHub:

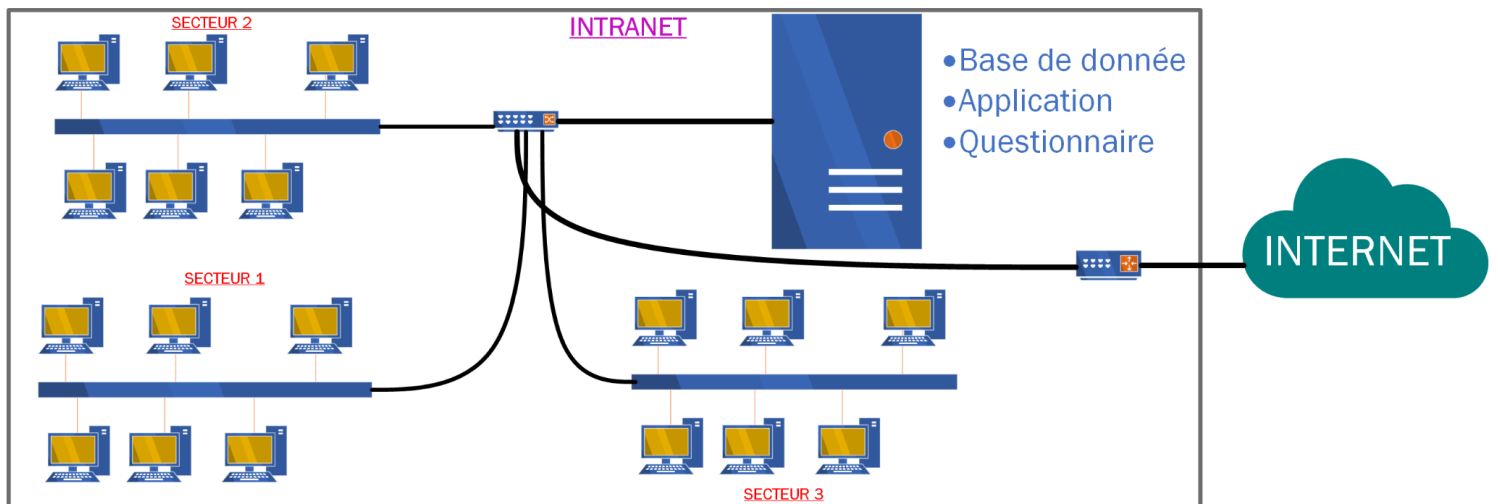
La plateforme GitHub permet aux programmeurs informatiques de collaborer librement sur des projets de code et nous a permis de planifier le projet avec un scrumboard.



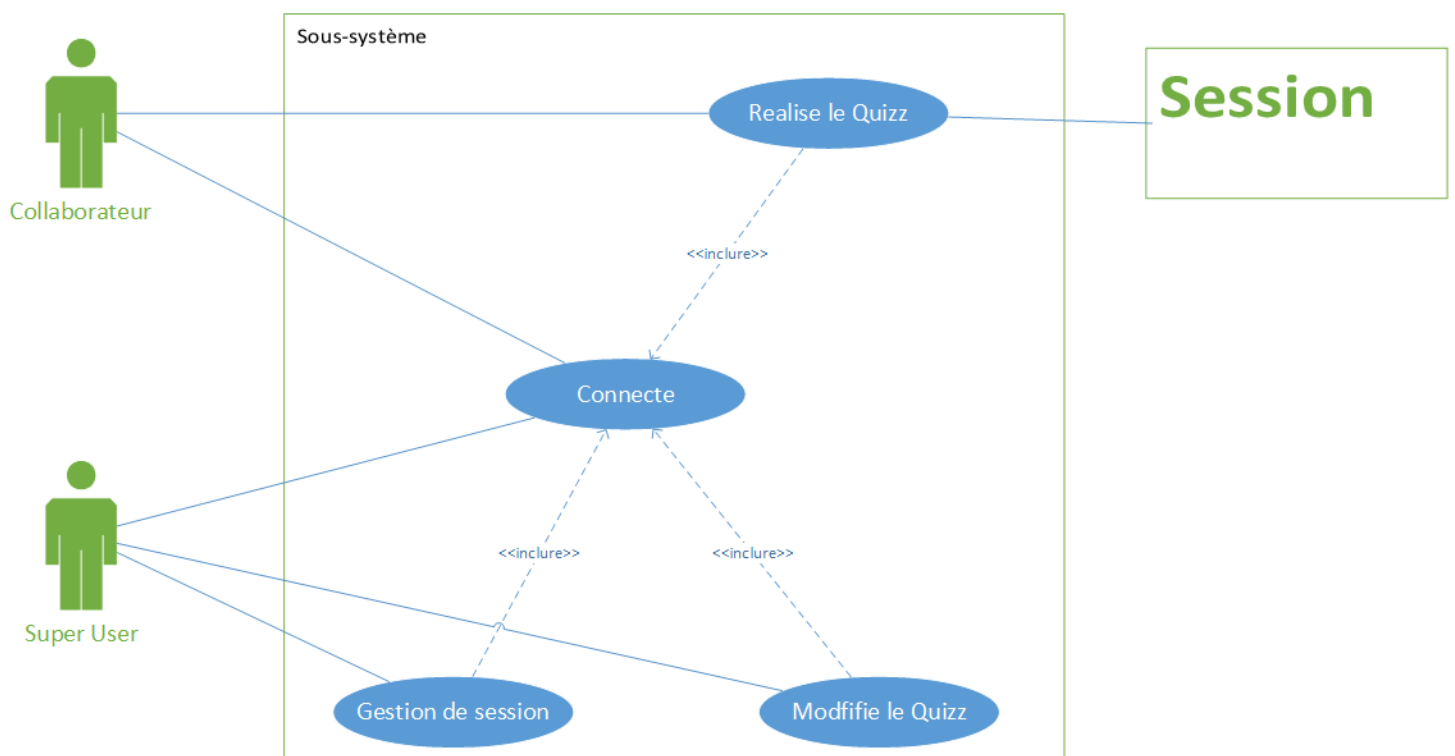
4. Maquette et architecture

Maquette: [FIGMA](#)

Architecture:

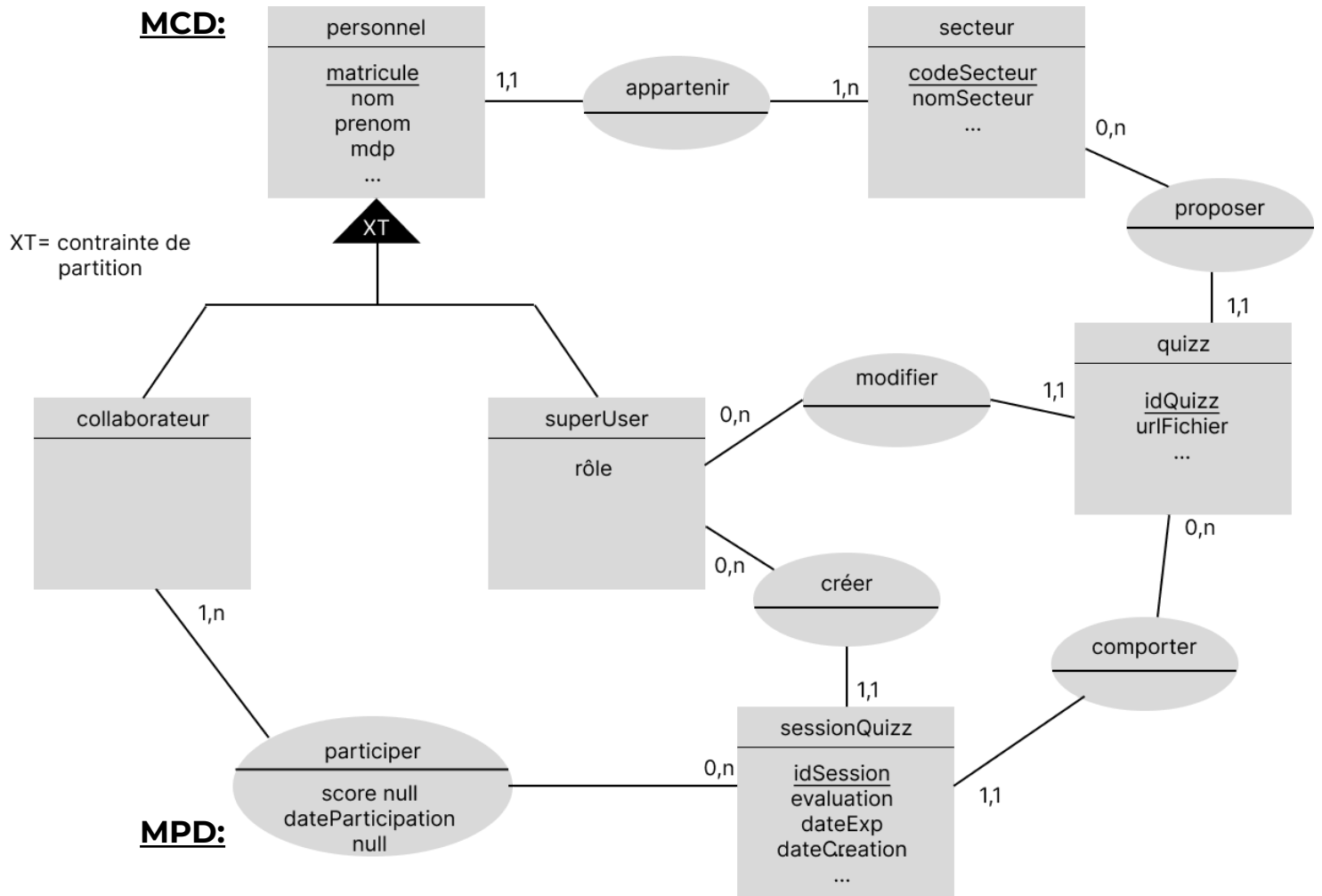


Graphe de dialogue:

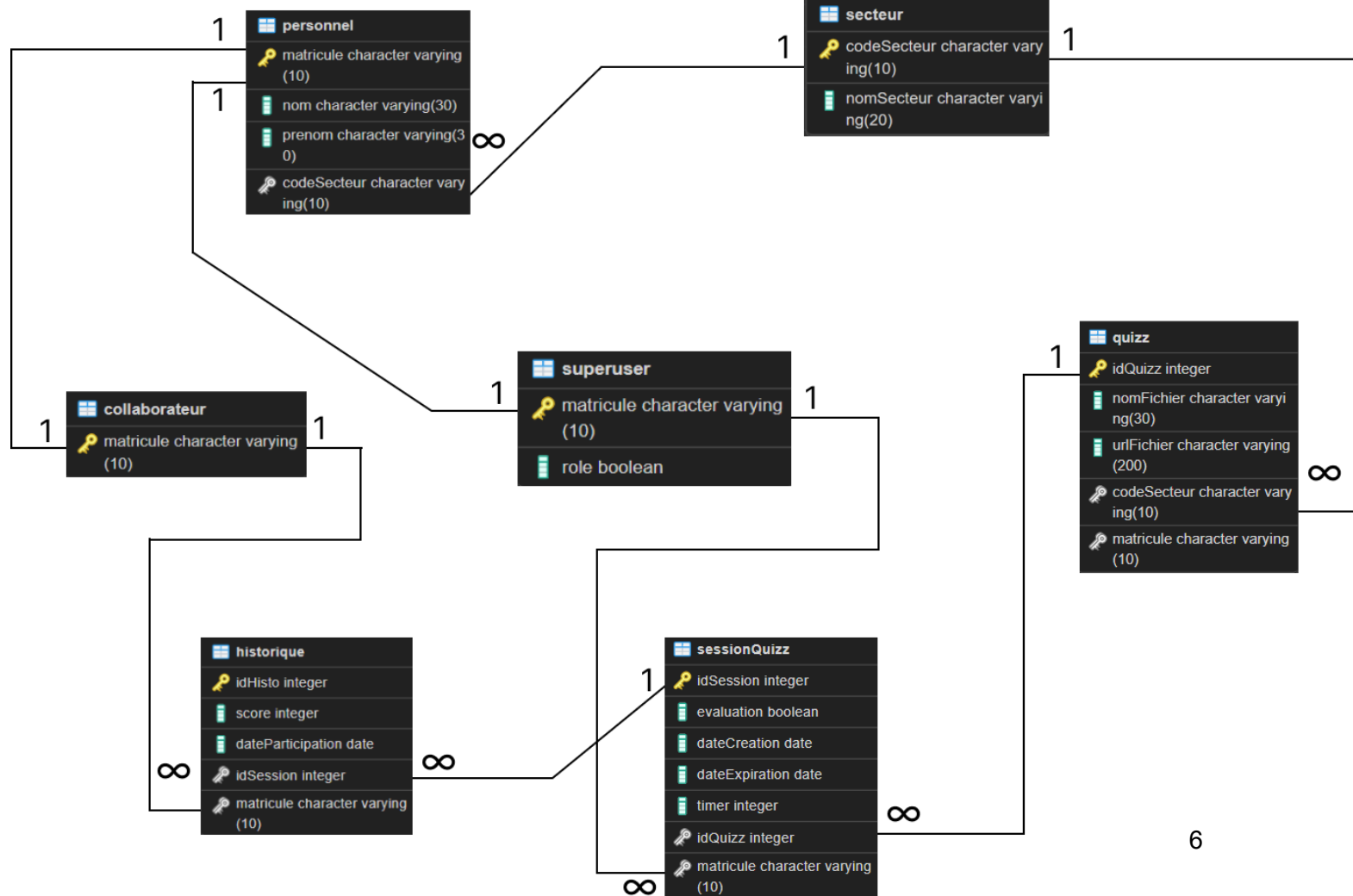


5. Base de données

MCD:



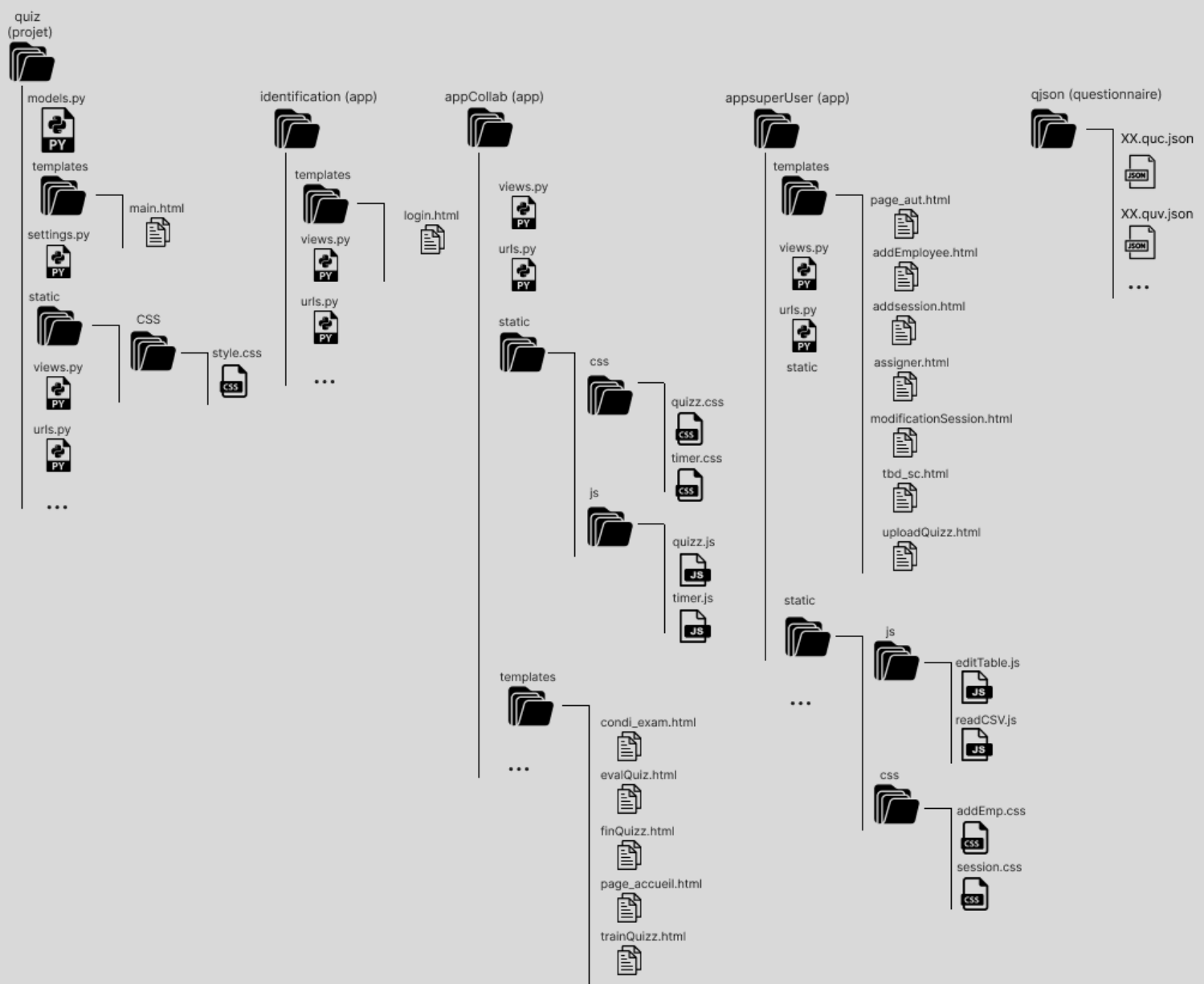
MPD:



Dictionnaire de données:

Attributs	Types de données	Valeur Null	Définition	Entité
Nom	Varcharr(30)		nom employé	Personnel
Prénom	Varcharr(30)		prénom employé	Personnel
Matricule	Varcharr(30)		numéro identifiant	Personnel
Rôle	booléen		rôle du super use 1= drh	Super User
CodeSecteur	Varcharr(30)		code identifiant d'un secteur	secteur
fURL	URL ou Varchar50		localisation du fichier (local)	Quizz
timer	int	x	temps de réalisation de la session	Session Quizz
dateExiration	Date		Date de fin de session	Session Quizz
Score	tinyInt	x	Le score du collaborateur de la session	Relation
DateParticipation	Date	x	Date à laquelle le collaborateur fait la session	Relation
Evaluation	booléen		Si evaluation ou non 1=eval	Session Quizz
NomDuFichier	Varcharr(30)		nom du fichier	Quizz
Mdp	Varcharr(30)		mdp du personnel	Personnel
IdQuizz	Integer		identifiant du Quizz	Quizz
IdSession	Integer		identifiant de la session	Session Quizz
nomSecteur	Varcharr(30)		nom du secteur	secteur

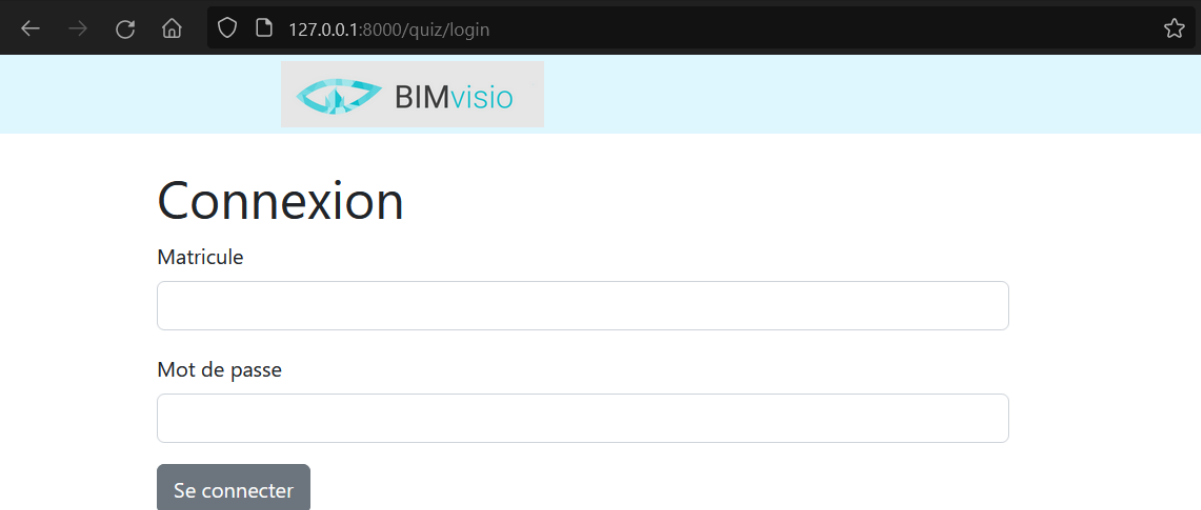
6. Arborescence du projet



7. Aperçu des fonctionnalités

A) Identification:

Permet à un utilisateur de s'identifier à partir de son mot de passe et de son matricule.




La table *Personnel*, *Superuser* et *Collaborateur* seront interrogées pour déterminer si l'utilisateur est présent et déterminer quel type d'utilisateur il s'agit. Il sera redirigé vers la page d'accueil concernant les collaborateurs (partie B) ou soit la page d'accueil concernant les superuser (partie C).

A.1 Redirection vers les pages d'accueil

L'interrogation et la redirection se fait dans la fonction `identification/views.py/login_user()`

A.2 Accès Refusé

L'accès aux pages du superuser et collaborateur est refusé pour les utilisateurs anonymes. J'utilise la décoration `login_required()` proposé par le système d'authentification de Django


BIMvisio

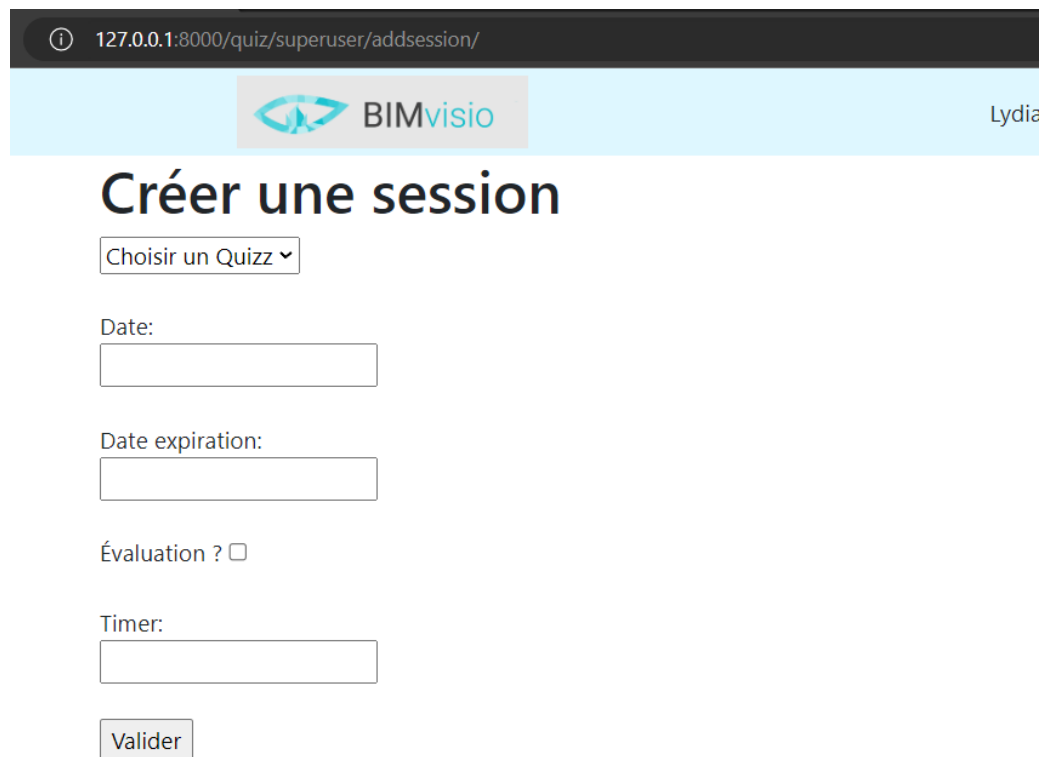
Lydia Langley 10

Deconnexion

Sessions en cours

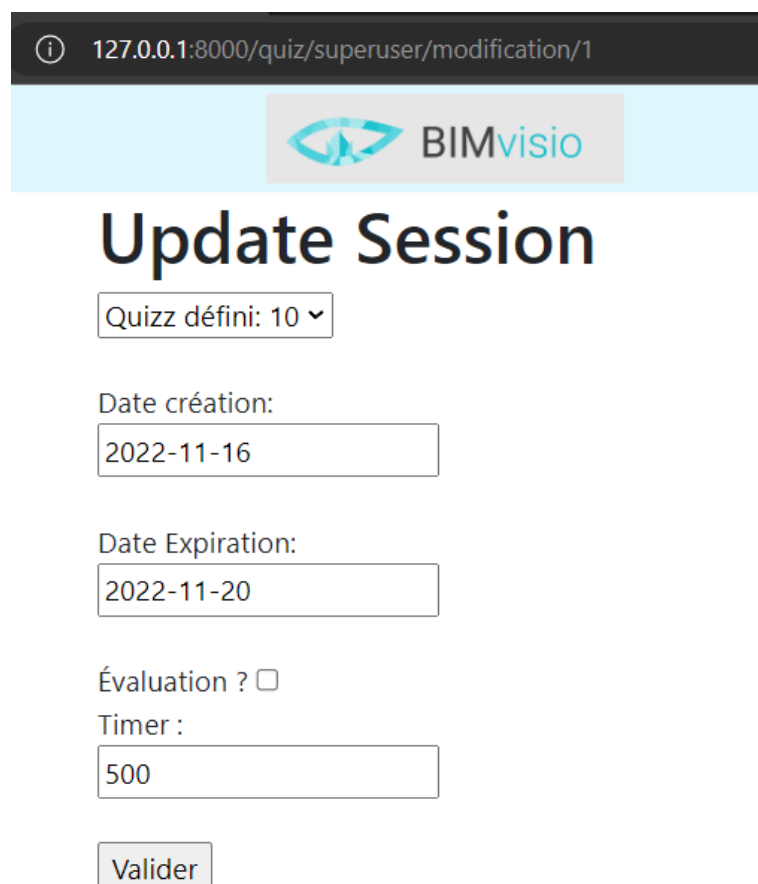
N° Session	N° Quizz	Date création	Date expiration	Type	Timer			
Session n°1	Quizz n°10	2022-11-16	2022-11-20	Entraînement	500	Assigner	Modifier	Effacer
Session n°4	Quizz n°14	2022-12-02	2022-12-25	Entraînement	500	Assigner	Modifier	Effacer
Session n°2	Quizz n°1	2022-11-17	2022-12-20	Évaluation	100	Assigner	Modifier	Effacer
Session n°3	Quizz n°14	2022-11-25	2022-11-30	Entraînement	400	Assigner	Modifier	Effacer
Session n°5	Quizz n°6	2022-01-01	2022-02-02	Évaluation	200	Assigner	Modifier	Effacer

Dans le gestionnaire de session on a la possibilité de voir les sessions en cours à partir de la base de données, créer de nouvelles sessions, les modifier ou supprimer et enfin assigner des collaborateurs aux sessions.



The screenshot shows a web browser window with the address bar displaying '127.0.0.1:8000/quiz/superuser/addsession/'. The page header features the BIMvisio logo and the name 'Lydia'. The main heading is 'Créer une session'. Below it is a dropdown menu labeled 'Choisir un Quizz'. There are three input fields: 'Date:', 'Date expiration:', and 'Timer:'. A checkbox labeled 'Évaluation ?' is present. At the bottom is a 'Valider' button.

La création de session se fait à partir d'un formulaire simple où l'on pourra récupérer les données saisies et les enregistrer dans la base de données



The screenshot shows a web browser window with the address bar displaying '127.0.0.1:8000/quiz/superuser/modification/1'. The page header features the BIMvisio logo. The main heading is 'Update Session'. Below it is a dropdown menu labeled 'Quizz défini: 10'. There are three input fields: 'Date création:' with the value '2022-11-16', 'Date Expiration:' with the value '2022-11-20', and 'Timer :' with the value '500'. A checkbox labeled 'Évaluation ?' is present. At the bottom is a 'Valider' button.

La modification de session fonctionne de la même manière et modifie les données dans la base de données après l'envoi du formulaire.

127.0.0.1:8000/quiz/superuser/assigner/1

Lydia Langley 10
Deconnexion

Assigner Collaborateurs à cette session

N° Session	N° Quizz	Date création	Date expiration	Type	Timer
Session n°1	Quizz n°10	2022-11-16	2022-11-20	Entraînement	500

Choix des collaborateurs à assigner

Matricule n°025Y	Assigner
Matricule n°187E	Assigner
Matricule n°133C	Assigner
Matricule n°151U	Assigner
Matricule n°169M	Assigner
Matricule n°043Q	Assigner
Matricule n°277Q	Assigner
Matricule n°097S	Assigner
Matricule n°079A	Assigner
Matricule n°259Y	Assigner

Collaborateurs actuellement assignés

Matricule	Score	Date Participation	
Matricule n°223O			Retirer
Matricule n°205W			Retirer
Matricule n°061I			Retirer
Matricule n°115K			Retirer
Matricule n°00			Retirer

Pour la page d'attribution des collaborateurs aux sessions, on affiche une liste des collaborateurs pas encore assigner à la session et la possibilité de les ajouter à la session puis une autre liste où l'on affiche les collaborateurs déjà assignés ainsi que leur score ou date de participation et un bouton pour les retirer de la session.

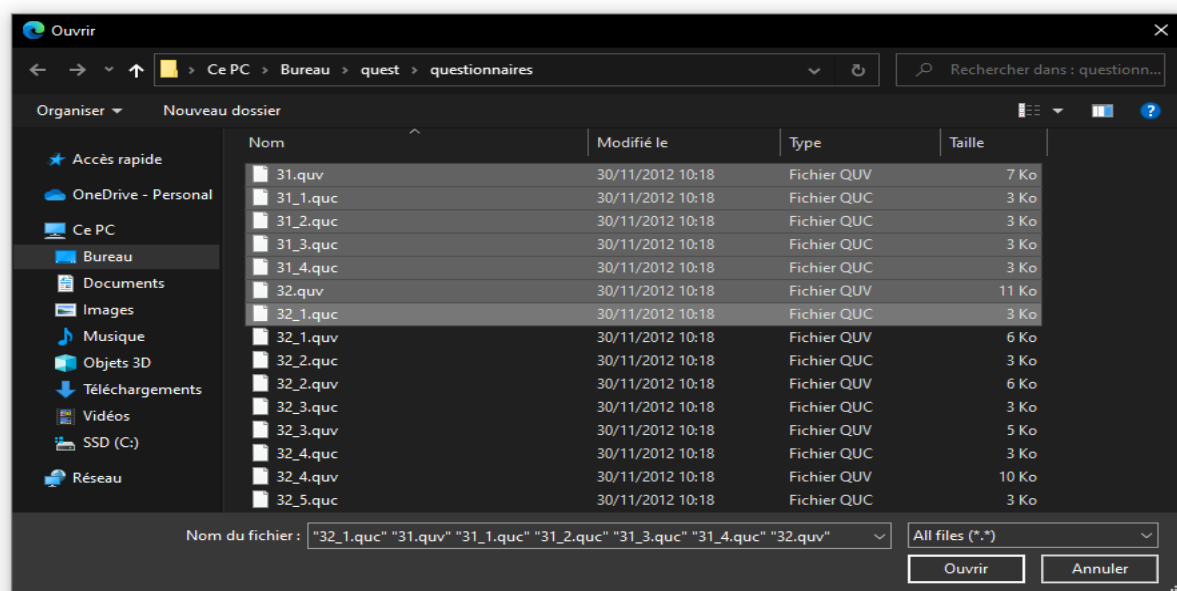
B.3 Import massif de quizz:

Lydia Lan

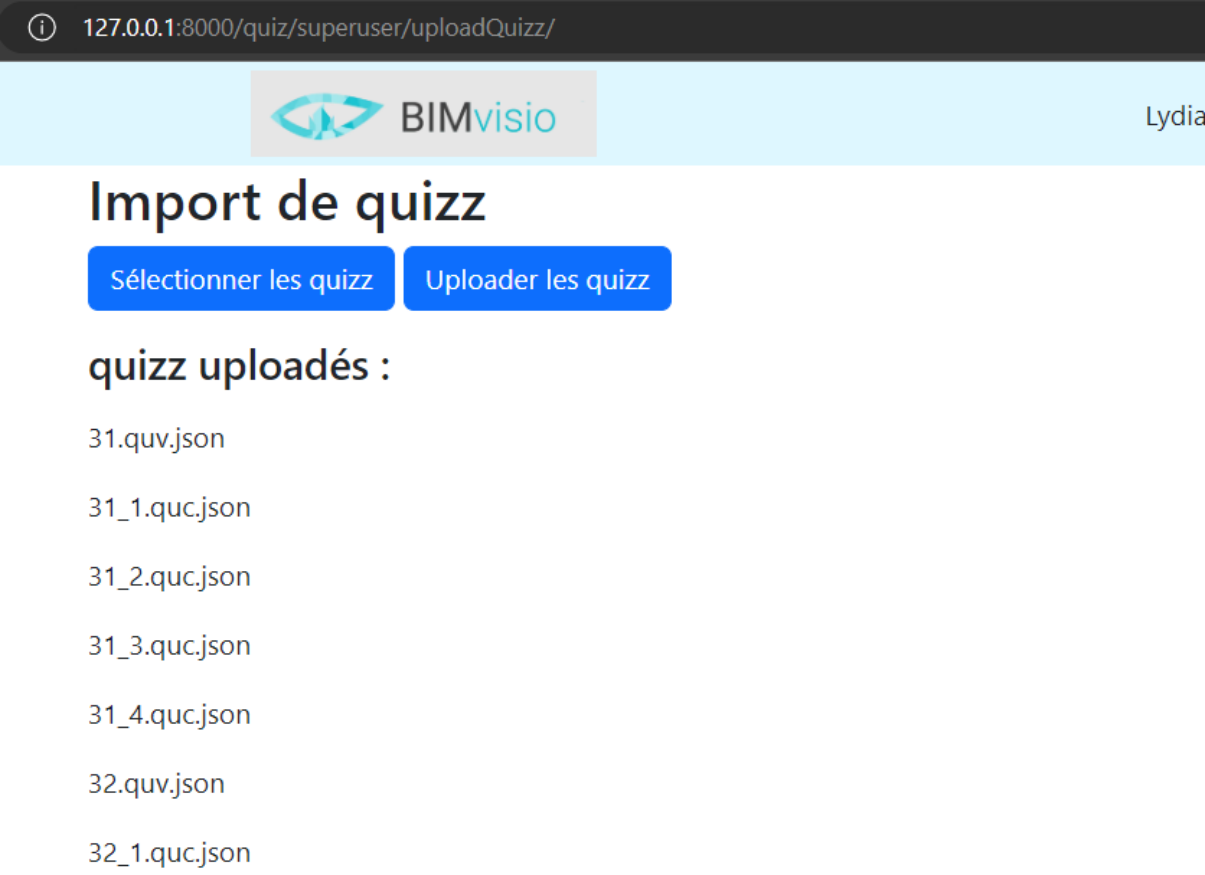
Import de quizz

Sélectionner les quizz
Uploader les quizz

Retour



Pour l'import de quizz, on a opté pour tout garder dans un fichier json à l'intérieur d'un dossier permanent et on enregistrera le chemin vers ce dossier ainsi que les noms des fichiers dans la table "quizz".



The screenshot shows a web browser window with the address bar displaying "127.0.0.1:8000/quiz/superuser/uploadQuizz/". The page header features the "BIMvisio" logo and the name "Lydia" on the right. The main heading is "Import de quizz". Below it are two blue buttons: "Sélectionner les quizz" and "Uploader les quizz". Under the heading "quizz uploadés :", a list of uploaded files is shown: 31.quv.json, 31_1.quc.json, 31_2.quc.json, 31_3.quc.json, 31_4.quc.json, 32.quv.json, and 32_1.quc.json.

```
<form method="POST" enctype="multipart/form-data">
.....
<input type="file" id='upQuizz' name="document" multiple>
```

Sur le template html on précise que le formulaire d'upload de fichier utilise la méthode POST et prends en compte l'upload de plusieurs fichiers en même temps avec l'attribut "multiple"

```
def uploadQuizz(request):
    monRepertoireQ = 'questionnaires/'
    monRepertoireTemp = 'qjsonTemp/'
    monRepertoire='qjson/'
```

Dans la view on mets dans des variables le nom des chemins d'accès aux dossiers qu'on a besoin

```

if request.method == 'POST':
    for f in request.FILES.getlist('document'):
        quizz=f
        fs=FileSystemStorage()
        fs.save(quizz.name,quizz)
        exec(open("script/xmljson.py").read())

```

Si il ya une requête de type post, on va donc faire une boucle afin de récupérer les fichiers uploadés et on va utiliser le module FileSystemStorage de django et avec la méthode .save() on sauvegarde les fichiers dans un dossier “questionnaires” défini dans le settings.py :

```

MEDIA_ROOT = os.path.join(BASE_DIR, 'questionnaires')

```

Une fois la boucle terminée et que tous les fichiers sont arrivés dans le dossier “questionnaires” on exécute le script xmljson.py afin de lire les fichiers xml puis extraire les données sous forme de dictionnaire puis enfin les sauvegarder en format json.

Dans le xmljson.py on va récupérer les fichiers dans le dossier “questionnaires”

```

def getListeFichiers(dossier) :
    listeFichiers = []
    for (repertoire, sousRepertoires, fichiers) in walk(dossier):
        listeFichiers.extend(fichiers)
        break
    return listeFichiers

monRepertoire = 'questionnaires/'
listeFichiers = getListeFichiers(monRepertoire)

```

On va ensuite définir une fonction qui va permettre de lire tous les fichiers en utilisant le module xmltodict

```

def traiterUnFichier(fichierQuizz):
    with open(fichierQuizz,'r',encoding="ISO-8859-1") as myfile:
        obj = xmltodict.parse(myfile.read())
    return obj

```

On lance une boucle afin de transformer tous les fichiers quv et quc au format json et on les enregistre dans 2 dossiers.

```

for fich in range(len(listeFichiers)):
    json_data =json.dumps(traiterUnFichier(monRepertoire+listeFichiers[fich]))
    pathOutput = "qjson/"+listeFichiers[fich] +".json"
    with open(pathOutput, "w") as outfile:outfile.write(json_data)
    pathOutputTemp = "qjsonTemp/"+listeFichiers[fich] +".json"
    with open(pathOutputTemp, "w") as outfile:outfile.write(json_data)

```

Le script xmljson.py se termine ici et on retourne donc à la view.

```

listeFichiers=[f for f in listdir(monRepertoireTemp) if
isfile(join(monRepertoireTemp,f))]
urlFichiers=[monRepertoire + f for f in listeFichiers]

```

On enregistre dans une variable une liste de tous les noms de fichiers contenu dans “qjsonTemp” et dans une autre variable le chemin d’accès vers ces fichiers en concaténant le nom du dossier permanent avec la liste des fichiers

```

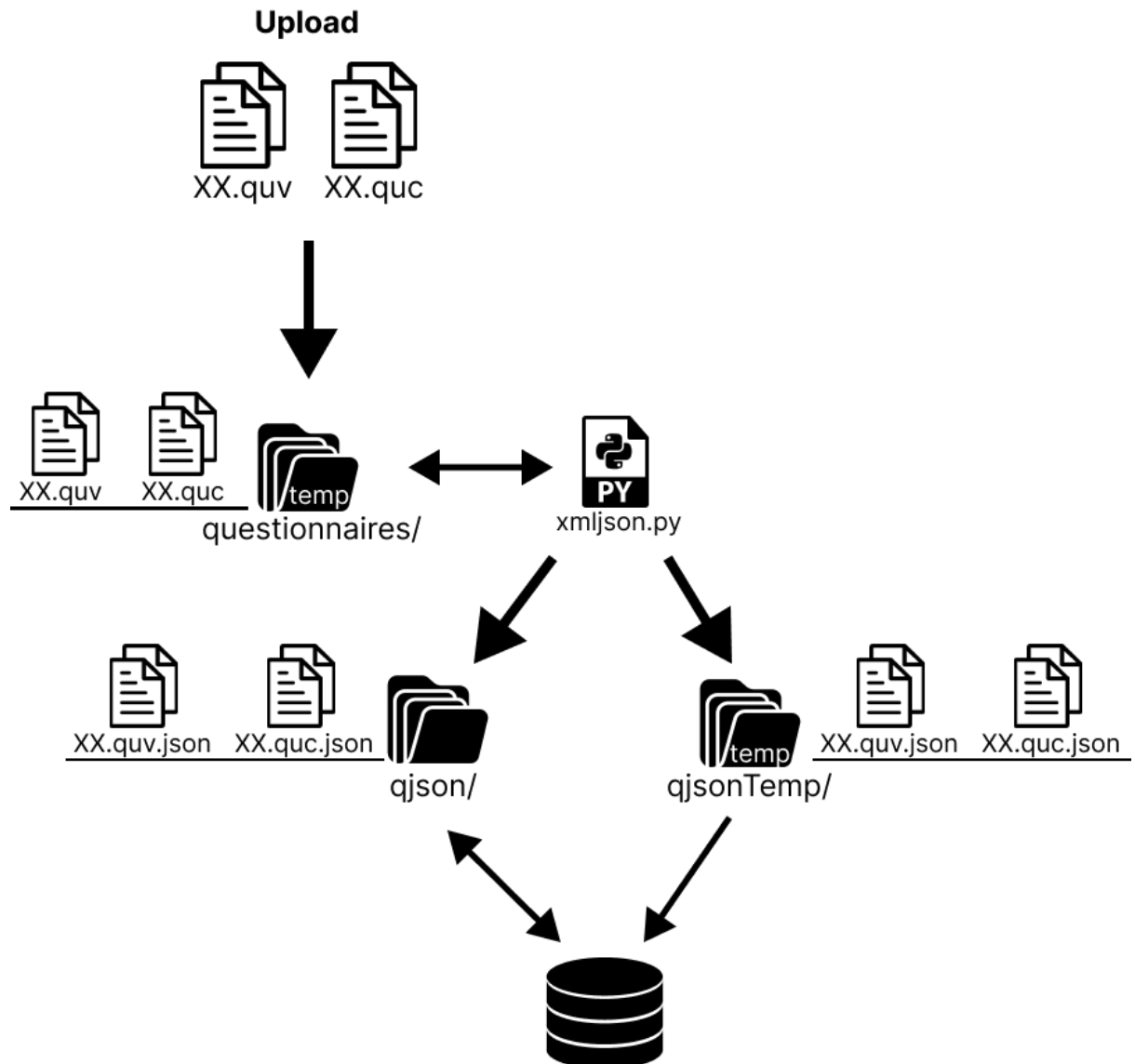
for i in range(len(listeFichiers)):
    nQ= Quizz(nomfichier=listeFichiers[i], urlfichier =
urlFichiers[i])
    nQ.save()

```

Pour sauvegarder tout ça dans la base de données on lance une boucle for et on indique dans un variable que pour chaque itération on va sauvegarder le nom du fichier ainsi que l’url des fichiers dans la table Quizz. Au final on obtiendra ce genre de table dans la bdd:

idQuizz [PK] integer	nomFichier character varying (30)	urlFichier character varying (200)
1	31.quv.json	qjson/31.quv.json
2	31_1.quc.json	qjson/31_1.quc.json
3	31_2.quc.json	qjson/31_2.quc.json
4	31_3.quc.json	qjson/31_3.quc.json
5	31_4.quc.json	qjson/31_4.quc.json
6	32.quv.json	qjson/32.quv.json
7	32_1.quc.json	qjson/32_1.quc.json
8	32_1.quv.json	qjson/32_1.quv.json

Schéma explicatif de l'upload:



B.4 Ajouter du personnel

À partir de la page, on importe un fichier au format csv qui sera lu et affiché dans une table. On pourra modifier les données ou les enregistrer dans la base de données. Le fichier d'exemple comportant les collaborateurs a ajouté n'étant pas normalisé pour l'instant on enregistre les données (les collaborateurs) dans le secteur marketing. On considère aussi que les données correspondent aux collaborateurs.

Ajout d'employé

Parcourir... MKT_secteur (marketing).csv Envoyer

Prénom	Nom	Matricule	Password	
Jeanne Melissa	CADY	025Y	025Y	Edit Delete
Jean Terence	DEVEQUI	187E	187E	Edit Delete
Jean Ugo	FOK YIN	223O	223O	Edit Delete
Jean Pascal	INDICE	205W	205W	Edit Delete
Jean Idriss	KOURBANHOUSSEN	061I	061I	Edit Delete

La partie édition des données depuis le tableau se fait dans le script *superuser/statis/js/editTable.js*

La partie récupération des données depuis la table se fait dans le script *superuser/statis/js/readCSV.js*

C)Fonctionnalités du collaborateur:

C.2 Réalisation du quiz d'évaluation

La fonctionnalité quiz d'évaluation envoie question par question à l'utilisateur et à chaque validation de la question, on envoie la question suivante et on vérifie si son choix de réponse est correct. À la fin du questionnaire il est redirigé vers la page de score.

← → 127.0.0.1:8000/quiz/collaborateur/eval-quiz BIMvisio Garrett Lyon 00 Deconnexion

QUIZ

0:16

01/17 - Élément d'échafaudage de pieds

Un camarteau est :

☐ Un outil spécial de coffreur ☐ Une situation de travail difficile à vivre ☐ Une semelle d'échafaudage tubulaire

Valider

Le compte à rebours est réalisé par Mateusz Rybczonec, le code se trouve dans les fichiers *appColob/static/js/timer.js* et *appCollab/static/css/timer.css*

C.2.1 Coté Serveur

initQuiz()

Dans un premier temps on initialise les éléments permettant la mise en place du quizz.

- On charge le questionnaire:

```
with open(file) as f: data = json.load(f)
```

- On fait appel au dictionnaire de *session* de l'objet *request* pour stocker des données côté serveur. On stocke ainsi les éléments permettant:
 - de suivre les points,
 - d'envoyer les questions au fur et à mesure

Le stockage d'une valeur se fait comme ceci:

```
request.session['nom-de-la-variable'] = valeur
```

Pour suivre la question à envoyer on utilise une variable `numQuestion` qui représente l'indice ou l'index de la question dans l'objet contenant le quiz

`extractData()`

Cette fonction permet de récupérer le titre, l'intitulé, la durée, les choix de la question courante ainsi que le numéro de question (`numQuestion`).

`nextQuestion()`

Cette fonction est appelée par le client en envoyant des données, les numéros des cases cochées (contenu dans une liste) représentant les choix de la question.

Puis on envoie en réponse la question suivante.

Pour envoyer la question suivante on incrémente `numQuestion`

Quand `numQuestion` n'est plus inférieur à au nombre de question présent dans le questionnaire alors on envoie comme réponse seulement le `numQuestion` (et non plus la question suivante)

`score()`

Accorde le point si:

- le nombre de bonne réponse correspond au nombre de case coché par l'utilisateur
- et si:
 - le numéro de la case à cocher correspond au numéro de la bonne réponse

puis calcule le score en fonction du coefficient

C.2.2 Coté client

evalQuiz.html

Les données envoyées par le serveur (par la fonction `initQuiz()` puis par `nextquestion()`) sont récupérés puis assignées de la ligne 41 à 46 dans le fichier `appCollab/template/evalQuiz.html`

updateTag

Chaque choix est matérialisé par une case à cocher dont le nom est un numéro, un indice qui est un vis à vis de l'indice des choix de la question. Ces *check box* sont générées par du javascript car il n'y a pas le même nombre de choix dans chaque question. Le remplissage du titre, intitulé et la durée se font dans la fonction `quizz.js/updateTag()`

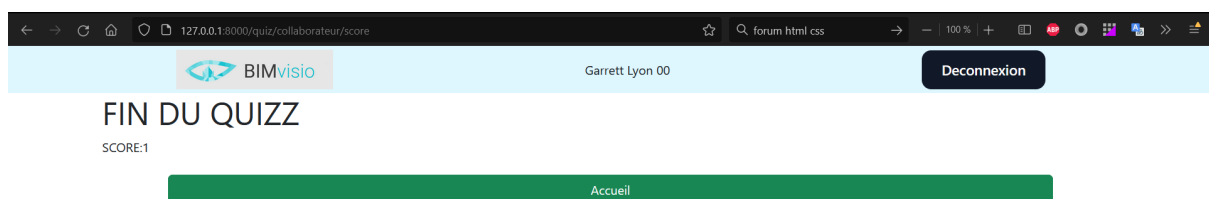
sendData()

Les données sont ensuite envoyées en cliquant sur le bouton *valider* qui fait appel à la fonction `appCollab/template/evalQuiz.html/sendData()` ligne 53 à 93. Cette fonction fait appel à *ajax* qui envoie et réceptionne la réponse. Lorsque l'on reçoit la réponse qui est la question suivante, on met à jour les éléments de notre page: titre, intitulé, durée, les choix ainsi que le `numQuestion`

La mise à jour des champs se fait ligne 73 à 82

Si le `numQuestion` n'est plus inférieur à au nombre de question présent dans le questionnaire alors on redirige l'utilisateur vers la page *score*

D) Score



Sur cette page on récupère le score stocké dans le dictionnaire de session de l'utilisateur et on l'inscrit dans la balise correspondante.

7. Conclusion et To do list

- Finir la stylisation des templates
- Mettre en place la réalisation d'un quiz d'entraînement
- Persistence des données
- Créer des questions depuis un formulaire
- Affichage des données en fonction du rôle du superuser
- Ajouter la fonction permettant de fermer le pop-up dans la page de *login*
- Enregistrer le score dans la table *Participer*