

Numdiag Questionnaire - Cahier des charges

Max Chateau

1 Présentation du projet

1.1 Motivation

Création d'une application pour créer et éditer des questionnaire à l'aide d'une interface utilisateur. L'administrateur pourra ainsi éditer des questionnaire qui seront soit publiés sous forme de questionnaire unitaires pour des démos, soit utilisés pour les audits de l'application PUMA.

Les questionnaires seront à destination d'utilisateurs, qui lorsqu'ils auront terminé de le remplir, obtiendront un score en fonction des réponses données, ainsi que des recommandations basées sur le score global et/ou sur certaines réponses de questions.

1.2 Principe

- Création de questions avec différents types de réponses (QCM, entiers, réponses libres), avec un coefficient qui affectera le score global en fonction de la réponse.
- Les questions seront regroupées dans une section, et ces sections composeront un questionnaire.
- Les questions peuvent avoir des dépendances avec des réponses (*ie* : une réponse pourra conditionner l'affichage ou non d'une question)

2 Détails techniques

2.1 Modèle de données

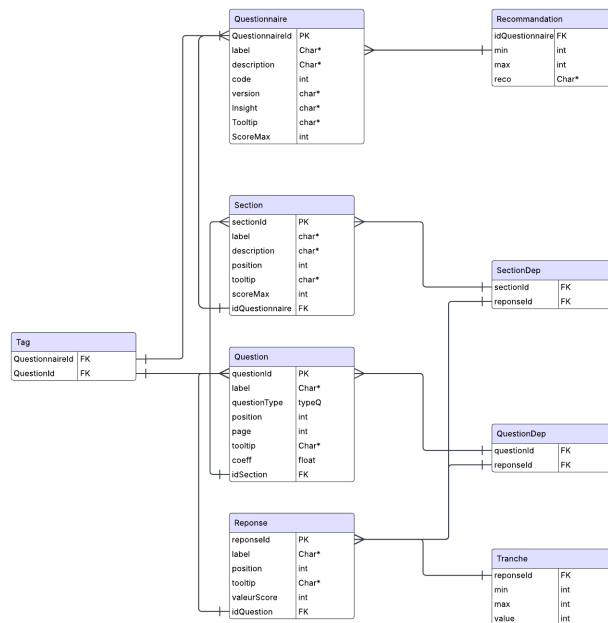


Figure 1: Modele Relationnel de Donnees

2.2 Technos Utilisées

Pour le front on va utiliser ReactJS + Vite. Pour le back on utilisera ExpressJS. La base de données sera hébergée sur PostgreSQL.

Ces 3 services seront séparés, et dockerisé, qui permettra d'utiliser docker-compose. On utilisera un unique repo Github pour ces 3 services.

2.3 Fonctionnement Global

La page admin affichera les différents questionnaires déjà créés, il y aura la possibilité d'en créer des nouveaux, de les supprimer, de les modifier ou de les dupliquer.

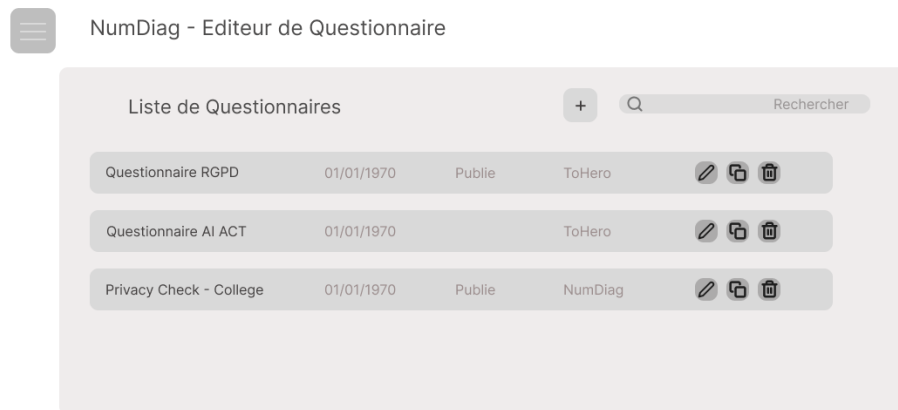


Figure 2: Page d'accueil des questionnaires

Lors de la modification d'un questionnaire, une nouvelle page s'ouvre pour afficher les sections, ainsi que les questions contenues dans chaque section. On peut également afficher les différentes caractéristiques de la section pour les modifier.

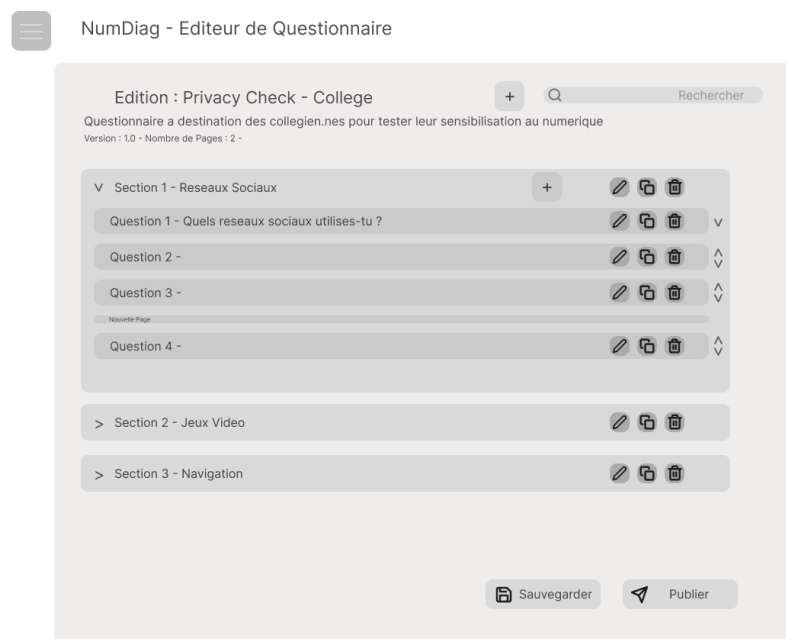


Figure 3: Vue de modification d'un questionnaire

L'edition d'une question vient se faire dans une pop-up qui va demander les differentes informations qui sont les reponses, le score, les indications.



Figure 4: Vue de modification d'une question

2.4 Fonctionnement Technique

Une creation va envoyer les informations au back afin de verifier s'il est possible de creer l'element demande. Lors d'une demande d'edition, le serveur va recuperer les toutes les informations du questionnaire associe demande dans la database, et les stocker en tampon, puis les envoyer a l'admin.

L'admin va faire les modifications de son cote, puis les envoyer au serveur, il va alors comparer avec ce qu'il a de son cote pour verifier si la modification est possible.

Quand l'admin a termine toutes les operations, le serveur va envoyer toutes les modifications dans la database.

Aucun element de la base de donnees ne sera fait directement dans le front, le serveur back fera office d'intermediaire pour transmettre les infos

2.5 Les routes

Les infos relatives aux differents appels API seront stockes dans le body des requetes

On implementera une architecture CRUD et REST pour toutes les routes de l'application

Les routes GET :

- url/questionnaire : renvoie les questionnaires avec toutes les infos ()
- url/questionnaire/:idquestionnaire : renvoie les les infos d'un questionnaire specifique
- url/questionnaire/:idquestionnaire/section : renvoie toutes infos de toutes les section d'un questionnaire

Les routes POST :

- url/login : Permet a l'utilisateur de se connecter (mail + password)
- url/register : Permet a l'utilisateur de s'enregistrer (mail + password + ...)
- url/questionnaire : Permet de creer un questionnaire (toutes les infos indiquees dans la database)
- url/questionnaire/:idquestionnaire/section : Permet de rajouter une section a unn questionnaire
- url/questionnaire/:idquestionnaire/section/:idsection/question : Une question a une section d'un questionnaire
- url/questionnaire/:idquestionnaire/section/:idsection/question/:idquestion/reponse : Permet de rajouter une reponse a une question
- url/document : creer un nouveau document
- url/document/:idDocument : permet d'editer un document (tag associe, description, etc...)
- url/createTag : permet de creer un tag
- url/linkTag : permet de lier un tag a une question