Rackakey

**Le 08.01.2019**

Présentation à Monsieur Carrel de la vision que j’ai du projet avec un schéma fait pour l’occasion (Schema\_Fonctionnement.vsdx):



Selon lui, on ne sait pas exactement encore comment cela va fonctionner au final (la communication entre le hub usb et le site).

La partie à gérer est celle du site avec Laravel avec ces points à prendre en compte :

* Historique des réservations des clés
* La clé usb doit être créée en tant que modèle logique (tout est simulé de ce côté-là)
* Il est possible d’utiliser le .env pour simuler un utilisateur connecté (🡪 A savoir qu’au final la connexion se fera au travers d’une interface prévue à cet effet comme c’est le cas pour l’intranet par exemple)
* Pour le point précédent, il faut quand même avoir une table dans la bdd
* Simuler une page où le professeur choisira les fichiers qu’il veut envoyer sur la clé usb. L’envoi en lui-même est simulé.
* Avoir une date de début et date de fin lorsqu’on fait une réservation (dans la table)
* Avoir des rôles (professeur, secrétariat et par ex admin)
* Le professeur peut :
  + Voir ses propres réservations
  + Ajouter une réservation
* Le/la secrétaire peut :
  + Supprimer une clé usb (lorsqu’elle a disparue par exemple)
  + Ajouter l’état « initialisé » à la clé usb
* L’admin peut faire toutes les actions des autres utilisateurs

A faire dans l’ordre :

* Créer les tables dans Laravel en utilisant les modèles à la Eloquent :

~~usbs~~

* + ~~reservationsUsbs (Voir dans la doc comment Eloquent veut qu’on lie tables intermédiaire n à n)~~
    - ~~reservation\_usb (table à créer selon éloquent) et y insérer reservation\_id et usb\_id et file\_id~~
  + ~~files (table regroupant~~
  + ~~users~~
  + ~~reservations~~
  + ~~status~~
  + roles
* Créer les seeds allant dans les bdd
* La vue principale a été créée.   
  Selon Monsieur Carrel il faut :
  + Faire en sorte que tous les éléments sur la page d’accueil soient dynamiques (tout récupérer de la bdd)   
    ~~🡪 Créer le numéro de port dans le modèle (table usbs)  
    🡪 Créer le rack dans le modèle (table usbs)~~

~~🡪 Créer le rack dans le modèle (table usbs)~~

* + ~~Ajouter un champ titre à la réservation (on pourra ainsi donner un nom à la réservation)~~
  + Ajouter le bouton « Ajouter une réservation » tout en haut de la page
  + La vue de « Ajouter une réservation » donnera la liste des réservations actuelles de l’utilisateur
* ~~La colonne « Action » ne devrait pas être affichée sur la page d’accueil pour les professeurs.  
  En effet, ceux-ci voient seulement les usbs en état disponibles. Ils peuvent seulement effectuer des actions sur la page où ils voient leur réservation.~~
* ~~La colonne « Action » sur la page d’accueil doit être affichée pour le rôle secretary. Elle leur permet d’activer une clé et de la rendre « Disponible » lorsque le status est « NotInitialized ».   
    
  Ceci évite par exemple qu’une personne vienne mettre sa clé usb personnelle et que cette dernière apparaisse disponible pour les autres. Les clés NotInitialized ne peuvent jamais être disponibles tant qu’un secrétaire/admin ne l’a pas initialisée (ce qui la mettre dans mode Disponible).~~
* ~~Il faut faire en sorte que tout le monde voit les clés~~
* Seules les actions vont changer.   
    
  Par exemple un prof ne peut pas supprimer ni initialiser une clé mais la secrétaire le peut. Seules les actions entre les personnes vont varier.
* La table statut doit contenir : active/inactive. Quand une clé est rentrée pour la première fois elle est inactive par défaut  
    
  Le statut en lui-même doit être « calculé » selon ces données :

*Si Rack = 0 (la clé n’est dans aucun rack = pas présente dans un hub)*

*Si clé a réservation existante*

*Si date\_returned dans la réservation est vide (clé pas encore retourné)*

*Status = Utilisée*

*Sinon date\_returned dans la réservation est rempli (clé retournée)*

*Status = Absente*

*Sinon (si clé a aucune réservation existante)*

*Status = Absente*

*Sinon si rack != 0 (clé est bien présente dans le rack)*

*Si clé a réservation existante*

*Si date\_returned dans la réservation est remplie*

*Status = Disponible*

*~~Si date\_returned clé n’est pas rempli~~*

*~~Status = Pas encore libérée par un prof~~*

*Sinon (si clé n’a aucune réservation existante)*

*Status = Disponible*

Suite à discussion avec Monsieur Carrel, le laisse tomber le schéma ci-dessus et je passe à celui qu’il a fait :

Si inRack = YES

Si inReservation = YES

Status = Présente (Commandée mais pas encore récupérée)

Si inReservation = NO

Status = Disponible

Si inRack = NO

Si inReservation = YES

Status = Utilisée (Pas dans le rack et a bel et bien une réservation)

Si inReservation = NO

Status = Absente (Ni dans le rack ni dans une réservation !!)

Ne pas oublier d’ajouter un champ dans la table réservation qui dit si elle est finie ou non !