

# Projet lien SGBD-LOO

## Romain BACHELIER & Pierre PALLIER

### Sujet 2

#### Base de données

Nous avons choisi d'avoir les tables suivantes dans la base de données :

- **user** : les utilisateurs inscrits et leurs informations.
- **doctor** : les médecins enregistrés et leurs informations.
- **medical\_tests**: les analyses possibles et leurs informations.
- **authorizations** : association entre médecins et tests, pour savoir qui est autorisé à réaliser les analyses.
- **schedule** : la liste des rendez-vous, qui contient des clés étrangères vers un médecin, un utilisateur, et une analyse. Nous avons choisi d'utiliser des Timestamp pour les dates, et un booléen pour indiquer si la visite a été payée.

Malheureusement, nous n'avons pas réussi à résoudre un problème à temps : la base de données insère les dates dans la table schedule avec une heure de décalage.

#### Java

##### **Authentication :**

Les utilisateurs commencent par s'authentifier en fournissant l'adresse mail associé à leur compte (bouton « Log in »). Si l'adresse fournie ne correspond à aucun utilisateur en base de données, un message d'erreur apparaît. Nous avons choisi l'adresse mail comme identification afin d'éviter les situations de noms et prénoms similaires, l'adresse mail étant unique.

S'ils n'ont pas encore de compte, ils peuvent en créer un (bouton « Sign in ») en renseignant leur nom, prénom, numéro de sécurité sociale, adresse, et email. S'il existe déjà une adresse mail ou un numéro de sécurité sociale identique dans la base de données, le compte n'est pas créé et un message d'erreur informe l'utilisateur. L'adresse mail doit être unique pour l'identification, et le numéro de sécurité sociale pour le réalisme.

Une fois les informations rentrées correctement, une fenêtre informe l'utilisateur du succès de la création du compte, et une nouvelle ligne est ajoutée dans la table « user » de la base de données contenant les informations rentrées. L'utilisateur peut donc se connecter avec le compte qu'il vient de créer.

Si des champs sont laissés vides lors de l'inscription, un message d'erreur apparaît et l'inscription ne se fait pas.

##### **Faire une réservation :**

Les utilisateurs peuvent réserver un rendez-vous pour les analyses disponibles (bouton « Book appointment »). Le choix de l'analyse se fait par un menu déroulant qui recense les analyses présentes dans la base de données.

Une fois l'analyse sélectionnée, un algorithme va parcourir la liste des médecins autorisés à réaliser cette analyse, et va sélectionner le premier rendez-vous disponible. Si tous les médecins sont libres, le médecin dont le dernier rendez-vous est le plus lointain sera sélectionné, et le rendez-vous sera créé pour l'heure actuelle. S'il n'existe aucun rendez-vous enregistré, le médecin en tête de liste sera choisi.

Une fois l'heure du rendez-vous et le médecin choisis, un nouvel objet Schedule est créé à partir de ces informations, puis inséré dans la base de données. Une fenêtre de confirmation informe l'utilisateur de la date et l'heure de son rendez-vous (qui est décalée d'une heure par rapport à l'information rentrée dans la base de données à cause du bug mentionné plus haut).

Nous avons choisi un format d'heure 'aaaa-MM-jj hh:mm:ss' afin de simplifier la gestion de dates, puisqu'il s'agit du format par défaut des objets Timestamp.

### **Voir les réservations et payer :**

L'utilisateur peut consulter ses rendez-vous dans un tableau (bouton « View appointments »). Il peut y voir le nom du médecin qui va réaliser l'analyse, le type et la date de l'analyse, ainsi que si la visite a été payée ou non.

En cliquant sur une ligne et sur le bouton « Pay » (si la visite n'est pas déjà payée), l'utilisateur va pouvoir payer pour sa visite. Une fenêtre va lui demander ses informations de paiement, qui vont ensuite être transmises à un service de paiement (pour l'exercice, nous avons un placeholder qui va juste valider les informations peu importe les données reçues).

Une fois le paiement validé, le booléen « paid » dans la table Schedule va être mis à jour, et la vue se rafraîchit pour afficher les données mises à jour.