
DOCUMENTATION - DEVELOPPEUR

Table des matières

▸ Informations sur le projet	Page N°1
▸ Identifiants de connexion	Page N°1
▸ Info & structure de la base de données	Page N°2
▸ Les fonctions de la passerelle	Page N°3

→ Informations

Seule la demande de PDF par utilisateur fonctionne.

→ Identifiants

Administrateur ayant tous les privilèges :

Username : *admin*

Mot de passe : *admin*

Infos : *Admin en administration*

Pharmacien :

Username : *arondelaere*

Mot de passe : *mdptropdur*

Infos : *Pharmacien en radiologie*

Infirmier :

Username : *lpodevin*

Mot de passe : *mdptropdur*

Infos : *Infirmier en immunologie*

→ Infos & structure de la base de données

INFOS

- SGBD utilisé: PgAdmin
- IP: 192.168.1.245:5432
- User: rondelaere | Pass: rondelaere
- Database name: slam2024hopitalpharmacie_rondelaere

STRUCTURES

demandes(id,qtte,date,#auteur,#produit)

- ➡ auteur : clé étrangère sur username de la table users
- ➡ produit: clé étrangère sur id de la table produit

produit(id,libelle,prix,qttestock,localisation,seuil)

profession(id,libelle)

qtteStockService(id,qttestock,#produit,#service)

- ➡ produit: clé étrangère sur id de la table produit
- ➡ service: clé étrangère sur id de la table services

services(id,libelle)

sorties(id,auteur,date,qtte,#idproduit)

- ➡ idproduit: clé étrangère sur id de la table produit

users(username,nom,prenom,mail,mdp,#profession,#service)

- ➡ profession: clé étrangère sur id de la table profession
- ➡ service: clé étrangère sur id de la table services

→ Fonctions de la passerelle

- **connexionBDD()**

- DESC: *se connecter à la base de donnée*
- RETOUR: *aucun*

- **donnerTousLesProduits()**

- DESC: *montrer la liste de produit*
- PARAM: *aucun*
- RETOUR: *ArrayList <Produit>*

- **donnerUnProduit()**

- DESC: *donner un produit particulier*
- PARAM: *unId*
- RETOUR: *Produit*

- **ajouterProduit()**

- DESC: *ajouter un produit à la liste de produit*
- PARAM: *Produit unProduit*
- RETOUR: *Boolean*

- **modifierProduits()**

- DESC: *modifier un produit dans la liste lesProduits*
- PARAM: *Produit unProduit*
- RETOUR: *Boolean*

- **supprimerProduits()**

- DESC: *supprimer un produit dans la liste lesProduits*
- PARAM: *Produit unProduit*
- RETOUR: *Boolean*

- **retirerStock()**

- DESC: *mettre à jour le stock d'un produit*
- PARAM: *Produit unProduit, int pQtte, String pAuteur*
- RETOUR: *Boolean*

- **retirerStockLog()**

- DESC: *retire les stocks et stocke l'info dans la BDD*
- PARAM: *Produit unProduit, int pQtte, String pAuteur*
- RETOUR: *Boolean*

- **getNBProduits()**

- DESC: *renvoie le nombre de produits différents*
- PARAM: *aucun*
- RETOUR: *int*

- **verifLogin()**

- DESC: *vérifie les identifiants que l'utilisateur entre*
- PARAM: *String pUsername, String pMdp*
- RETOUR: *Boolean*

- **hashPassword()**

- DESC: *hass le mot de passe*
- PARAM: *String pMdp*
- RETOUR: *aucun*

- **getNom()**

- DESC: *renvoie le nom de l'utilisateur*
- PARAM: *String pUsername*
- RETOUR: *String*

- **getPrenom()**

- DESC: *renvoie le prénom de l'utilisateur*
- PARAM: *String pUsername*
- RETOUR: *String*

- **getProfession()**

- DESC: *renvoie la profession de l'utilisateur*
- PARAM: *String pUsername*
- RETOUR: *String*

- **getService()**

- DESC: *renvoie le service de l'utilisateur*
- PARAM: *String pUsername*
- RETOUR: *String*

- **donnerStockService()**
 - DESC: *renvoie le stock d'un service*
 - PARAM: *String pService*
 - RETOUR: *ArrayList*
- **getNameProduitWithId()**
 - DESC: *donne le nom d'un produit grâce à son id*
 - PARAM: *int pId*
 - RETOUR: *String*
- **getIdProduitWithName()**
 - DESC: *donne l'id d'un produit grâce à son nom*
 - PARAM: *String pName*
 - RETOUR: *int*
- **ajouterDemande()**
 - DESC: *demande un renouvellement de stock d'un produits*
 - PARAM: *Produit unProduit, int pQtte, String pAuteur*
 - RETOUR: *Boolean*
- **getAllUsername()**
 - DESC: *renvoie tous les noms d'utilisateur*
 - PARAM: *aucun*
 - RETOUR: *ArrayList<String>*
- **getAllDemandesWithIDUser()**
 - DESC: *renvoie toutes les demandes de renouvellement de stock grâce à l'id de l'utilisateur*
 - PARAM: *String pUser*
 - RETOUR: *ArrayList<Demandes>*
- **getAllDemandesWithDate()**
 - DESC: *renvoie toutes les demandes de renouvellement de stock faites un jour précis*
 - PARAM: *Date date*
 - RETOUR: *ArrayList<Demandes>*

Pour que l'application soit ergonomique elle doit:

- répondre aux besoins des utilisateurs de l'application
- rendre l'application désirable
- assurer la portabilité de l'application
- assurer la crédibilité de l'application

Pour qu'elle soit fonctionnelle, on doit:

- créer un seuil critique où le stock de médicaments se remet automatiquement
- créer un pdf à chaque commande passée
- créé un moyen de valider ou annuler des commandes des services
- créé un moyen pour que les infirmier puissent gérer complètement

Pour qu'elle doit être sécuré, il faut:

- que les injections sql ne marchent pas
- hacher les mots de passe dans la base de données
- que les utilisateurs aient des accès limités à l'application

Aide administration:

- Exporter BDD (pgAdmin): `pg_dump -h 192.168.1.245 -U rondelaere -p 5432 -d slam2024hopitalpharmacie_rondelaere > bdd.sql`