**Documentation Technique du Projet iStore**

**1. Introduction**

Ce projet est une application de gestion de magasin nommée **iStore**. Il permet la gestion des utilisateurs, des magasins et des stocks d'articles via une interface graphique développée en **Java Swing** et une base de données **MySQL**.

**2. Architecture du Projet**

L'application est organisée en plusieurs packages :

* **Authentification** : Gestion de la connexion et de l'inscription des utilisateurs.
* **User** : Gestion des utilisateurs.
* **iStore** : Gestion des magasins et des articles.

**3. Packages et Classes Principales**

**3.1 Package Authentification**

Ce package contient les classes responsables de l'authentification et de la gestion des utilisateurs.

**Conn**

* **Rôle** : Gestion de la connexion à la base de données.
* **Méthodes principales** :
  + Connect(String email, String password): Authentifie un utilisateur.
  + singIn(String email, String pseudo, String password): Inscription d'un utilisateur.
  + whitelisted(String email): Vérifie si un email est sur liste blanche.
  + getConn(): Retourne l'instance de connexion à la base de données.

**Login**

* **Rôle** : Interface de connexion.
* **Attributs** : emailField, passwordField
* **Méthodes principales** :
  + Login(JFrame parent): Constructeur qui initialise l'interface de connexion.

**SignIn**

* **Rôle** : Interface de création de compte.
* **Attributs** : email, password, pseudo
* **Méthodes principales** :
  + SignIn(JFrame parent): Constructeur qui initialise l'interface d'inscription.

**HomeScreen**

* **Rôle** : Écran d'accueil de l'application.
* **Méthodes principales** :
  + HomeScreen(): Constructeur qui affiche les options de connexion et d'inscription.

**3.2 Package User**

Ce package contient la classe permettant de gérer les informations des utilisateurs.

**Utilisateur**

* **Rôle** : Représente un utilisateur et gère ses informations.
* **Attributs** : ID, email, pseudo, role
* **Méthodes principales** :
  + Utilisateur(String email): Constructeur qui initialise un utilisateur à partir de son email.
  + setInfo(): Récupère les informations de l'utilisateur depuis la base de données.
  + getID(), getEmail(), getPseudo(), getRole(): Accesseurs des attributs.

**3.3 Package iStore**

Ce package contient les classes principales pour la gestion des magasins et des articles.

**iStore**

* **Rôle** : Interface principale de gestion des magasins.
* **Méthodes principales** :
  + iStore(Utilisateur utilisateur): Constructeur initialisant l'interface principale.
  + addStore(), removeStore(): Ajout et suppression de magasins.
  + loadStoresFromDatabase(): Charge la liste des magasins depuis la base de données.

**ManageUsers**

* **Rôle** : Interface de gestion des utilisateurs.
* **Attributs** : utilisateur, userListModel, userList
* **Méthodes principales** :
  + ManageUsers(Utilisateur utilisateur): Constructeur initialisant l'interface de gestion des utilisateurs.
  + addUser(), deleteUser(): Ajout et suppression d'utilisateurs.
  + loadUsersFromDatabase(): Charge la liste des utilisateurs.

**ManageArticles**

* **Rôle** : Interface de gestion des articles.
* **Attributs** : utilisateur, articleListModel, articleList
* **Méthodes principales** :
  + ManageArticles(Utilisateur utilisateur): Constructeur initialisant l'interface de gestion des articles.
  + addArticle(), removeArticle(), updateStock(): Gestion des articles.
  + loadArticles(): Charge les articles en fonction du magasin sélectionné.

**Compte**

* **Rôle** : Interface de gestion du compte utilisateur.
* **Méthodes principales** :
  + pseudoChang(String newPseudo, int userID): Modifie le pseudo de l'utilisateur.
  + emailChang(String newEmail, int userID): Modifie l'email de l'utilisateur.
  + supprimerCompte(int userID): Supprime le compte utilisateur.

**4. Base de Données**

L'application utilise une base **MySQL** avec les principales tables suivantes :

* utilisateur(id, email, pseudo, password, role) : Stocke les utilisateurs.
* store(id, name) : Liste des magasins.
* inventory(id, store\_id) : Inventaire des magasins.
* items(id, name, storage, price, inventory\_id) : Articles en magasin.
* whitelisted(id, email) : Liste blanche d'emails autorisés.

**5. Fonctionnalités Principales**

* **Authentification** : Connexion et inscription sécurisées.
* **Gestion des utilisateurs** : Ajout, suppression et modification des utilisateurs.
* **Gestion des magasins** : Ajout et suppression des magasins.
* **Gestion des articles** : Ajout, suppression et mise à jour du stock des articles.
* **Interface graphique** : Interface Swing ergonomique et interactive.

**6. Conclusion**

Ce projet offre une gestion complète des magasins et des utilisateurs via une interface intuitive. Il est structuré en plusieurs classes bien définies pour faciliter la maintenance et l'évolution future.