

# 2JJAVA

# Documentation

## Sommaire

### Introduction

- Présentation du projet iStore
- Objectif de la documentation
- Technologie utilisée

### Architecture de l'application

- Structure et description des packages
  - com.iStore.config
  - com.iStore.dao
  - com.iStore.model
  - com.iStore.ui
  - com.iStore.ui.panels
  - com.iStore.utils

### Fonctionnalités clés

- Gestion des utilisateurs
- Gestion des articles et inventaires
- Sécurité des mots de passe

### Base de données

- Schema des tables

## Introduction

### *Présentation du projet iStore :*

iStore est une application conçue pour gérer plusieurs magasins et faciliter leur administration. Grâce à celle-ci, il est possible de gérer les utilisateurs qu'ils soient administrateurs ou employés, gérer les articles, les inventaires et les différents magasins. L'application propose une interface graphique intuitive afin de pouvoir y naviguer facilement.

### *Objectif de la documentation :*

Cette documentation a pour but de montrer la structure de l'application iStore, les fonctionnalités clés puis la base de données. Le code a été documenté grâce à JavaDoc (index.html dans le dossier doc).

### *Technologie utilisée :*

iStore a été entièrement fait en Java. Nous pouvons aussi retrouver du SQL pour les requêtes vers la base de données.

## Architecture de l'application

### *Structure et description des packages :*

com.iStore.config : Contient tout ce qui touche à la base de données. Il y a CreateTable.java pour la création des tables puis DatabaseConfig.java pour gérer la connexion à la base de données.

com.iStore.dao : Contient tout ce qui touche à l'accès à la base de données. Pour commencer, AdminDAO.java permet de gérer toutes les opérations d'administrateur comme celles liées aux magasins, aux articles, à la gestion des employés etc... puis nous avons UserDAO.java qui est là pour gérer les opérations d'employé comme l'authentification, la création de compte etc...

com.iStore.model : Contient le modèle de données représentant les différentes entités de l'application. On y retrouve les classes User.java, Store.java et Item.java.

com.iStore.ui : Contient tout ce qui touche à la gestion de l'interface graphique de l'application. GraphicInterface.java est la classe principale qui initialise la première fenêtre de l'application et gère la navigation entre les différents panels.

com.iStore.ui.panels : Contient tous les panels de l'application qui composent l'interface. Chaque panel représente une fonctionnalité spécifique comme l'authentification, l'inscription, la gestion des magasins, des utilisateurs etc...

com.iStore.utils : Contient des classes utilitaires utilisées à travers l'application. Il y a SessionManager.java qui est là pour enregistrer les données de l'utilisateur connecté et s'en servir dans différents panels, puis PasswordHash.java qui est là pour gérer le hash du mot de passe stocké dans la base de données.

# Fonctionnalités clés

## *Gestion des utilisateurs :*

iStore nous permet grâce à son interface de gérer les utilisateurs qu'ils soient des Administrateurs ou des Employés. Les deux rôles ont leurs droits spécifiques. Par exemple, les administrateurs peuvent créer de nouveaux comptes, mettre à jour les informations des utilisateurs quels qu'ils soient contrairement aux employés qui peuvent seulement visualiser les autres comptes. Ils peuvent seulement modifier ou supprimer leur propre compte. La gestion des rôles permet d'avoir une séparation des responsabilités entre les administrateurs et les employés.

## *Gestion des articles et inventaires :*

iStore nous permet aussi la gestion des magasins, des inventaires et des articles qui les composent. Toujours grâce à l'interface, il est possible de créer, modifier, supprimer ou encore mettre à jour les articles dépendant du magasin. Bien sûr, dépendant du rôle, nous avons plus ou moins de permission. Un administrateur pourra gérer tous les magasins et tous les articles, et un employé pourra gérer seulement le magasin auquel il est affilié.

## *Sécurité des mots de passe :*

Lors de la création des comptes, le mot de passe ne pouvait pas être stocké directement en brut dans la base de données. C'est pourquoi les mots de passe des utilisateurs sont d'abord hashé en utilisant l'algorithme SHA-512 avec un sel aléatoire. Seul le hash et le sel sont stockés dans la base de données pour ensuite pouvoir comparer le mot de passe.

# Base de données

