

Etude de cas : Institut de beauté

ZENHEALTH est un professionnel en soins du corps qui dispose de plusieurs instituts de beauté. ZENHEALTH veut doter chacun de ses instituts d'une application pour satisfaire les clients. Les caractéristiques de cette application se présentent comme suit :

Un institut de beauté possède plusieurs cabines. Les clients peuvent réserver par téléphone ou à l'accueil de l'institut. Une cabine ne peut être réservée à plusieurs clients en même temps. Les clients s'installent généralement dans une cabine, et passent une commande. Une telle commande consiste en un ou plusieurs services assurés par l'institut. On suppose qu'à une cabine donnée correspond une commande, et qu'une cabine n'est servie que par une seule hôtesse. Il existe deux catégories d'hôtesse : hôtesse et gestionnaire. Chaque hôtesse a un nom et un prénom et est affectée à une ou plusieurs cabines pour toute une journée. Pour une bonne gestion de l'institut, on conservera la date et l'heure de la réservation, la date et l'heure de l'encaissement, le mode de paiement, et le montant total. Un service d'une commande est décrit par un libellé (soins de visage, massage, ...), un prix et le nombre d'interventions par jour (dû aux disponibilités des soignants par jour). Chaque service commandé sera décompté du nombre d'interventions par jour.

Travail demandé

Le but du projet est de développer une application Java dotée d'une interface textuelle (très simple) pour exécuter ses fonctionnalités, et accédant à une base de données via l'API JDBC en mode transactionnel pour éviter des situations d'incohérence.

Le travail se présente comme suit :

- Créer les tables ainsi que le jeu de données, sans oublier, d'ajouter les contraintes d'intégrité adéquates afin d'optimiser l'accès aux tables.
- L'application doit implémenter les fonctionnalités suivantes :
 1. Connecter un utilisateur (hôtesse/gestionnaire) à la base de données (connexion très simple).
 2. Donner le contrôle des transactions à l'application (*autocommit* désactivé).
 3. Une simple hôtesse doit pouvoir réaliser les opérations suivantes :
 - a. Réserver une cabine disponible pour une date et heure données.
 - b. Commander les services disponibles pour une réservation donnée.
 4. En plus des opérations d'une simple hôtesse, une gestionnaire doit pouvoir réaliser d'autres opérations, à savoir :
 - a. Affectation des hôtesse à des cabines.
 - b. Annulation d'une réservation non consommée.
 - c. Modifier le prix et/ou le nombre d'interventions pour un service donnée.
 - d. Calcul du montant total d'une réservation consommée et la mise à jour de la table RESERVATION pour l'encaissement.

Travail à rendre

Le projet est à réaliser en monôme.

Le projet sera évalué sur la base des critères suivants :

- Mode transactionnel pour communiquer avec la base de données, sachant que plusieurs hôtessees peuvent utiliser simultanément cette application.
- Fonctionnalités codées et opérationnelles.
- Clarté et qualité du code (ajout des commentaires).

Le rendu attendu comporte :

- Les fichiers code source.
- Un script sql sur toutes les données que vous avez exploitées.
- Un document *Read.me* (lisible) qui décrit les fonctionnalités implantées et comment les utiliser.

Préparer un fichier compressé au format ZIP dont le nom doit suivre le format :
Nom_Prenom.zip.

Attention : toute tentative de plagiat sera sévèrement sanctionnée.