

## **Projet PhotoNum - Rapport**

### **I. Introduction**

Le projet consiste en la réalisation d'une application Java de gestion de commandes accédant via le protocole JDBC à une base de données sur un serveur distant.

Tout se fait à travers une interface en mode texte offrant un moyen d'interactivité (menus, affichage des résultats, trace des opérations, ...).

L'accès aux données suit le patron de composants DAO.

Notre projet est organisé de 3 packages: dao, jdbc et tables.

Le package tables contient toutes les classes métiers POJO.

Le package dao contient toutes les classes DAO.

Le package jdbc contient le squelette de l'application, et les autres classes nécessaires pour implémenter la connexion au serveur, les commandes JDBC, etc.

### **II. Modifications apportées depuis le rapport intermédiaire**

Nous avons modifié plusieurs tables et relations.

**LesClients**(idClient, adrMail, nom, prénom, mdp, #idAdr)

**LesAdresses**(idAdr, numRue, nomRue, ville, codePostal)

**LesFichierImages**(cheminAcces, infoPriseDeVue, resolutionImage, estPartagé, #idPhoto, #idClient, conservation, #idProprio, #idUtilisateur)

**LesPhotos**(idPhoto, parametresRetouche, #idPage)

**LesCommandes**(idCommande, dateCommande, prix, statut, #idAdr, #idClient, #codeProm)

**LesPromotions**(codeProm, taux, utilisé, #idCommande)

**LesArticles**(idArticle, prixImpression, quantité, #idCommande, #idImpr)

**LesImpressions**(idImpr, référence)

**LesPages**(idPage, texteDescriptif, miseEnForme, #idPhoto, #idImpr)

**LesAlbums**(idAlbum, référence, titre, #idPhotoCouverture, #idPage)

**LesCadres**(idCadre, référence, #idPage)

**LesCalendriers**(idCal, reference, #idPage)

**LesTirages**(idTir, référence, #idPhoto, #cheminAcces)

### III. Avancement du projet

Nous avons fait les changements suivants:

- Un client possède un seul prénom, pas plusieurs.
- L'adresse de domicile du client est son adresse de livraison.
- Nous n'avons pas géré les stocks et la notion de catalogue.

Les raisons des changements: difficulté de l'implémentation de cette dernière ou manque de temps.

Nous avons créé les classes d'objets POJO avec accesseurs et mutateurs pour chacune des tables citées précédemment.

Nous avons créé les classes DAO correspondantes et la superclasse DAO.

Nous avons implémenté une méthode pour chacune de ces fonctionnalités:

- La **création d'un nouveau client** et authentification d'un client existant en utilisant le ClientDAO.
- L'**upload de fichiers images** pour un client en utilisant la méthode create de FichierImagesDAO et PhotoDAO. Un fichier image peut être partagé ou non.
- Un fichier image est supprimé 10 jours après non-utilisation.
  - Il manque tout de même la notion de temps qui n'a pas été implémentée pour gérer la date de conservation, pour l'instant nous utiliserons la méthode diminuerConservation() présente dans FichierImages.java qui permet de diminuer de 1 la conservation de l'image et qui se verra lancer chaque matinée par le gérant.
- La **création d'impressions** (tirage, album, agenda, etc) en utilisant les méthodes create des classes DAO respectives à chaque classe.
- La **suppression de fichiers images** en utilisant la méthode delete de FichierImagesDAO.
- Les **différentes fonctions de consultation**:
  - Les détails de la commande d'un client avec son statut
  - Les informations d'un compte client avec sa liste de codes promo encore valables
  - La liste des impressions d'un client
  - La liste des images partagées
- La saisie d'une commande (~)
- Le **changement de l'état d'une commande**: "Prêt à l'envoi", "Envoyée". Une **facture** est générée lors du passage d'une commande à l'état "Envoyée".

Ces méthodes se trouvent dans le package jdbc.

Nous avons réalisé les **triggers** suivants:

- Suppression automatique après 10 jours de non-utilisation.

```
CREATE TRIGGER suppr_image
After UPDATE of conservation on LesFichierImages
Begin
    DELETE from LesFichierImages WHERE conservation = 0;
END;
```

- Trigger pour supprimer une promo après utilisation

```
CREATE TRIGGER suppPromo
after update of utilise on LesPromotions
BEGIN
    DELETE from LesPromotions WHERE utilise=1;
END;
```

#### IV. Tests et scénarios en supports de tests et de la démo

Pour tester notre application, il suffit de suivre les indications.

Tout d'abord, vous devez choisir entre lancer l'interface Client ou l'interface Gestion en tapant 1 ou 2.

Ensuite, vous devez choisir entre connexion ou création de nouveau compte.

Vous êtes ensuite redirigés vers le **Menu Principal**.

Le Menu Principal propose plusieurs fonctionnalités selon l'interface choisie.

- Ajout / suppression de fichiers images, création d'impressions, différentes consultations et saisie de commandes pour le côté Client
- Changement de l'état d'une commande et génération d'une facture pour le côté Gestion.

→ Vous choisissez la fonctionnalité voulue en suivant les indications. D'habitude, il suffit de taper un chiffre entre 1 et 6 pour naviguer entre ces différentes fonctionnalités.

#### Remarques:

- Si vous choisissez de consulter, vous êtes redirigés vers un autre menu, le **Menu Consultation**. Vous procédez de même.
- Si vous choisissez de créer une impression, il va falloir être patient car il y a plusieurs informations à entrer dans l'application (12 pages et leurs photos pour 1 Calendrier par exemple). Nous n'avons pas automatisé cela pour que ça ressemble plus au cas réel.
- Dernière remarque à savoir: à la fin de chaque fonctionnalité (ajout de fichier images par exemple, ou autre), vous êtes redirigé vers le Menu Principal jusqu'à ce que vous décidiez de quitter l'application en tapant '6'.

#### Important:

- Respecter les types demandés. (Pas de int si string demandé, ...)
- Il se peut que nous n'ayons pas traité quelques cas limites faute de temps.

#### V. La gestion de projet et l'organisation du travail en groupe

Voici quelques points qui détaillent notre organisation de travail au cours du projet:

- Nous avons utilisé GitHub. Grâce à des pushes et des pulls simples ou en récupérant manuellement c'est-à-dire en faisant un CTRL + C / CTRL + V, on récupère les différents fichiers dont on a besoin.
- Nous avons effectué plusieurs backups en cas de perte de fichiers.
- Nous nous sommes retrouvés en présentiel pendant les créneaux dédiés.
- Nous nous sommes répartis les tâches mais nous avons tous participé à l'implémentation des différentes fonctionnalités: DAO, JDBC, Triggers, SQL...

#### VI. Lien GitLab

<https://gitlab.com/l3miage-abouds/projetbdgroupe14>