

Mise en œuvre d'une application de gestion de tirages photographiques

1 Organisation du projet

Le projet BD a lieu du 13/03 au 31/03, les lundis après-midi, avec une soutenance le lundi 3/04. Il doit être réalisé par des groupes de **cinq** personnes. La présence est obligatoire dans les créneaux ADE. Les salles affectées à cette activité sont indiquées sur le planning ADE (merci de respecter ces salles ou bien d'informer de votre localisation, *par exemple sur le tableau des salles officielles*). Des permanences seront organisées par les enseignants pour chacune des demi-journées consacrées au projet. Un pointage des gens présents sera réalisé pour chaque demi-journée.

L'évaluation portera sur le développement réalisé (conception, spécification, qualité du code, couverture de tests), le rapport et la soutenance. En outre, la prise de recul de chaque groupe par rapport à ses réussites et ses lacunes sera également appréciée.

Tous les documents relatifs au projet ainsi que les sources de l'application devront être accessibles sur un dépôt Gitlab (<https://gricad-gitlab.univ-grenoble-alpes.fr> de préférence) ou Github.

Attention, ce projet est prévu pour un travail pendant les créneaux dédiés de l'emploi du temps, c'est-à-dire les lundis après-midi. Vous devez faire attention à ne pas passer trop de temps hors de ces créneaux et à ne surtout pas négliger d'autres cours. L'intérêt pédagogique de ce projet est essentiel, mais il n'est qu'une partie du contrôle continu du cours de BD.

2 Objectifs du projet

Le projet consiste en la réalisation du serveur (back) d'une application Java de gestion de commandes, accédant à une base de données client sur un serveur distant (ici Oracle 19c de l'UFR). L'accent est mis sur la définition et l'implémentation de micro-services d'accès aux données, et sur le maintien de la cohérence des données : L'interface extérieure se limitera à des tests via Postman. **L'évaluation ne tiendra pas compte de la présence d'une IHM éventuelle.**

Les technologies utilisées seront : Java, Spring Boot, JPA, Oracle (SQL), Postman, et StarUML pour la conception. Vous devrez suivre le modèle d'architecture logiciel étudié en cours et en TP (framework SpringBoot, Repositories, Objets DTO, mapping JPA).

3 Position du problème

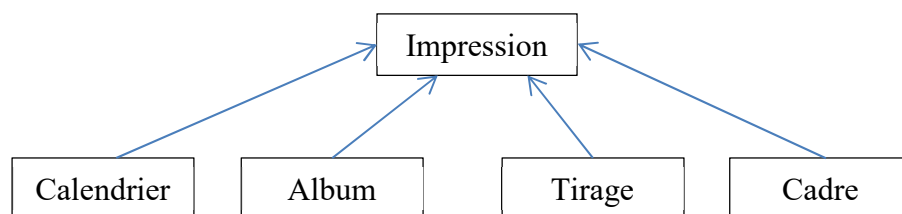
Une nouvelle société de tirage de photos numériques PhotoNum désire construire un système d'informations qui offre de nouveaux services aux clients tout en simplifiant la gestion du SI. Seule la partie Client sera développée dans ce projet.

Un client est décrit par son adresse mail, par ses nom et prénom(s), son adresse postale (utile pour l'expédition de ses commandes), ainsi qu'un mot de passe pour son authentification à la connexion au site PhotoNum. Chaque client doit télécharger ses fichiers images afin de pouvoir réaliser des impressions puis effectuer des commandes de ces impressions.

Une image est décrite par son chemin d'accès sur le serveur¹, son propriétaire, des informations de prise de vue, la résolution de l'image en million de pixels et une information précisant si le fichier est partagé ou non. Une image décrit l'objet physique (vous n'aurez pas à stocker de véritables photos, on en simulera seulement la présence virtuelle sur le serveur) et devra toujours être instanciée sous forme d'une photo dans les différents supports d'impression disponibles, appelés impressions dans la suite du document.

Le client organise ses photos dans des impressions : tirage, albums, calendriers, cadre. Une photo est donc une entité logique, liée à une image, et qui possède des paramètres de retouches d'image. Ces paramètres ne sont pas détaillés ici et seront gérés comme un seul élément (une chaîne de caractère par exemple). Une impression est donc une collection de photos organisées différemment selon le type spécifique d'impression. Une impression est identifiée de manière unique et possède un propriétaire :

- Un tirage est un ensemble de photos non ordonnées. Un tirage possède un format d'impression et une qualité de papiers (identique pour toutes les photos). Chaque photo peut être demandée en un certain nombre d'exemplaire.
- Un album permet d'organiser ses photos. Il possède une couverture avec une photo et un titre. Chaque page est composée de photos, chaque photo dispose d'un texte descriptif (en option). Les informations de mise en forme de chaque page ne sont là encore pas détaillées et placées dans un seul élément. Un album est disponible en différents formats et en différentes qualités.
- Un calendrier est un objet décoratif mural qui permet d'organiser des photos sur 12 pages. Chaque page correspond à un mois de l'année. L'organisation des photos sur une page est similaire à l'album.
- Un cadre est un objet décoratif qui peut contenir un ensemble de photos. La mise en page sera toujours simplifiée et placée dans un élément unique. Un cadre existe en différentes tailles et en différentes qualités.



Dans la suite du document, le terme *impression* sera utilisé pour désigner de façon générale les albums, tirages, calendriers ou cadres.

Un client peut décider de partager ses fichiers images. Dans ce cas, chaque fichier partagé est visible par tous les utilisateurs, qui peuvent alors l'utiliser dans des photos pour construire leurs propres impressions.

La commande d'un client est composée d'une date de commande et d'un ensemble d'articles (au minimum 1) et du prix total. Chaque article est constitué d'une impression avec son prix et sa quantité. Le prix d'un article est bien sur celui au moment de la commande. Un client peut créer plusieurs adresses et en choisir une pour la livraison.

Chaque type d'impression possède un prix en fonction de son format et de sa qualité. Un tirage photo, par exemple, précise un prix par photo en fonction de la qualité et du format. La société possède un catalogue qui répertorie tous les types d'impression avec leur tarif. Cette notion de

¹ Le dépôt de fichiers sera entièrement simulé, seul le chemin (fictif) du fichier sera stocké et aucun fichier ne sera uploadé sur un quelconque serveur.

qualité et de format est vu dans l'entreprise comme une référence dans ce catalogue (10x15MAT, 10x13BRIL, CALA4BRIL, CADA3MAT, etc.)

Une commande n'est possible que sur des impressions dont toutes les images utilisées sont soit propriété du client, soit partagées par une autre personne. Une fois validée, une commande ne change pas même si des fichiers partagés changent d'état.

4 Cahier des charges de l'application

La liste ci-dessous énumère les fonctionnalités essentielles que doit proposer l'application.

1. Création d'impression (tirage, album, agenda, etc) à partir du catalogue PhotoNum. Les photos qui composent l'impression seront créées à partir de fichiers images téléchargés ou bien à partir des fichiers images partagés par d'autres utilisateurs. On suppose les images déjà uploadées.
2. Modification ou suppression de fichiers images. Un fichier partagé ne peut être supprimé ou départagé. Un fichier peut être supprimé uniquement s'il n'est lié à aucune photo dans une impression en cours.
3. Différentes fonctions de consultation : détails de la commande d'un client avec son statut, les informations d'un compte client, la liste des impressions d'un client avec possibilité de détailler le contenu, la liste des images partagées.
4. Saisie d'une commande (choix des impressions, des formats relatifs, des quantités). La commande sera effective lorsqu'elle sera validée ; à cet instant, la commande sera associée au statut "en cours".

Initialisation du projet :

Toutes les images doivent exister au préalable dans la base, l'interface client n'en crée pas de nouvelles. Vous devrez donc les initialiser "en dur" via un script ou au démarrage de l'application.

Tous les clients peuvent exister au préalable de la même façon que pour les images. Vous pouvez aussi optionnellement les créer par l'interface client si vous en avez le temps.

5 Modélisation et conception

Vous donnerez un modèle conceptuel (un ou plusieurs diagrammes de classes UML), et son implémentation en classes d'entité Java avec le mapping JPA. Les choix conceptuels et d'implémentation devront être argumentés au besoin.

Cette question fait l'objet d'un rendu intermédiaire. Les diagrammes UML seront réalisés avec l'application StarUML.

6 Réalisation de l'application

Vous devrez réaliser la partie serveur (back) d'une application Java qui met en œuvre les fonctionnalités énoncées dans la section 4. Les interactions avec le serveur se feront par le biais de Postman.

Le contrat de l'API devra être spécifié formellement, et un jeu de tests complet devra être produit pour montrer la validation de contrat. Ces tests seront réalisés sous Postman.

Vous disposerez d'un modèle d'architecture complet et fonctionnel permettant la réalisation de test sous H2 et l'exécution d'une application (back)accédant à la base Oracle de l'UFR. L'architecture suivra le modèle vu en TP.

7 Rapport final et soutenance

Les documents (au format PDF) doivent répondre aux différentes questions dans un français correct.

Rapport intermédiaire. Le contenu du rapport décrit les points suivants :

- Le schéma conceptuel des données (UML) en argumentant vos choix si plusieurs solutions apparaissent (fichiers StarUML fournis),
- L'implémentation en classes d'entité JAVA,
- Une première version du mapping JPA pour assurer la persistance,
- La justification pour les choix de conceptions/implémentations.

Rendu final. Vous devrez faire l'effort d'être à la fois précis et concis. Le contenu du rendu doit comprendre un rapport et les sources du projet :

- Le rapport doit rappeler les objectifs du projet ainsi que l'organisation du document dans une brève introduction,
- Indiquer les modifications appliquées sur la partie conceptuel/implémentation du modèle de données depuis le rapport intermédiaire (UML, Java, JPA),
- Décrire brièvement l'état d'avancement du projet (fonctionnalités réalisées et non réalisées coté back),
- Présenter la gestion de projet et l'organisation du travail en groupe.
- Les sources et le projet seront disponible sur un dépôt Gitlab ou Github.
- Les sources devront intégrer la spécification du contrat d'API au format OpenAPI v3 et le jeu de test pour valider le contrat.

Soutenance. Vous aurez 15 minutes à votre disposition pour nous faire une démonstration de votre serveur, et répondre aux questions. Tous les membres d'un groupe sont conviés à la soutenance (sous réserve d'absence exceptionnelle justifiée). Chaque groupe est convoqué à une heure précise. Les horaires seront transmis au plus tard le 29 mars sur Moodle.

Chaque groupe dispose de 15 minutes

- Pour présenter le modèle de données, l'architecture logicielle, la gestion de projet,
- Pour montrer le respect du contrat de l'API,
- Pour répondre aux questions des enseignants.

8 Dates importantes

Les dates clés du planning sont les suivantes :

- **Vendredi 10 mars** : Mise en ligne du projet
- **Dimanche 12 mars** : deadline pour la définition des groupes
- **Lundi 13 mars** : première séance Projet.
- **Vendredi 17 mars 22h** : rapport intermédiaire Modèle de donnés
- **Lundi 20 et 27 mars** : séance Projet en présentiel
- **Dimanche 2 avril 20h** : dernier commit sur le dépôt

- **Lundi 3 avril : soutenances.**