

Michel de Broux	Modifications de la conception	Charles Momin
Simon Lardinois	LSINF1225	Valentin Rombouts
Victor Lecomte	Groupe V	Harold Somers

Nous allons présenter ici les modifications que nous avons faites en passant de l'étape de conception à l'étape d'implémentation, et leur justification. Nous avons refait le schéma relationnel et le schéma UML pour les illustrer.

1 Internationalisation de la base de données

Comme mentionné dans le rapport 3 (implémentation Android), nous avons dû adapter la base de données à une approche multilingue. Nous avons déjà joint notre nouvelle base de données avec ses instructions de création ; nous joignons maintenant à ce rapport-ci le schéma relationnel refait (fichier `relationnel-corrige.pdf`). Nous avons décidé de garder les noms de champs existants en français pour mettre en évidence le fait que la structure principale du modèle présenté dans le rapport 1 n'a été en rien modifiée.

2 Modifications du schéma UML

Lors de l'implémentation, nous avons modifié quelques éléments mineurs du diagramme de classe : nous avons ajouté quelques opérations et modifié quelques autres. Nous joignons à ce rapport une version corrigée de l'ancien diagramme de séquence, `uml-dao-corrige.png` (principalement, nous avons enlevé les getters/setters non pertinent pour ce type de document) et une nouvelle version mettant en évidence les changements. Cette nouvelle version marque avec un + les opérations ajoutées, avec un - les opérations enlevées (rendues inutiles par d'autres) et avec un * les opérations modifiées.

Passons rapidement en revue les raisons pour les différents changements :

- **Utilisateur** : Nous avons ajouté le champ `motDePasse` pour permettre à l'utilisateur d'afficher le mot de passe actuel dans les paramètres d'utilisateur. De plus, nous imaginons qu'il sera utile plus tard pour effectuer des opérations en s'identifiant auprès du serveur central. Nous avons supprimé le champ `langue` car le système Android préconise que l'application s'adapte directement à la langue du téléphone.
- **Client** : L'opération `viderPanier` n'existe plus dans l'interface dans client, car elle est appelée automatiquement quand on appelle `confirmerPanier`. L'opération `ouvrirCommande` a été ajoutée pour envelopper une procédure qui devait auparavant se faire directement via `DAOCommande`.
- **Manager** : L'opération `changerGrade` peut se faire simplement et directement par un appel `setGrade` dans `DAOUtilisateur`. Nous avons gardé la classe Manager car le fait qu'un utilisateur l'instancie signifiait qu'il était manager, et parce que nous n'excluons pas qu'avec un ajout ultérieur de fonctionnalités, cette classe obtienne des méthodes propres et intéressantes comme celles de `Serveur`.
- **Détail** : Le détail doit contenir son `idDétail` pour que l'on puisse modifier des informations qui y sont liées dans la base de données et la `dateAjouté` pour qu'elle puisse être affichée dans l'interface.
- **Consommation** : Nous avons dû ajouter le `nomAffichage` pour la traduction de l'application tout en gardant le `nom` pour la base de données. Étant donné que nous n'avons besoin du

type que pour son icône, nous avons changé `type` en `icôneType`.

- **Ingrédient** : Nous avons ajouté `utilisationActuelle` pour retenir l'utilisation déduite par le panier tout en gardant la valeur réelle du stock pour l'afficher dans la consultation du stock. Nous avons ajouté deux opérations liées à cela : `getRestant` qui donne la différence entre le stock et l'utilisation panier, et `appliquerUtilisation` qui déduit l'utilisation du stock à la confirmation du panier.