



# Application : Gestion de banque

APPLICATION JAVA

Réalisée par : Ouahib Salma |  
Encadrée par : Elfaker Abdellatif |  
Date : 14 avril 2024

## I. Introduction :

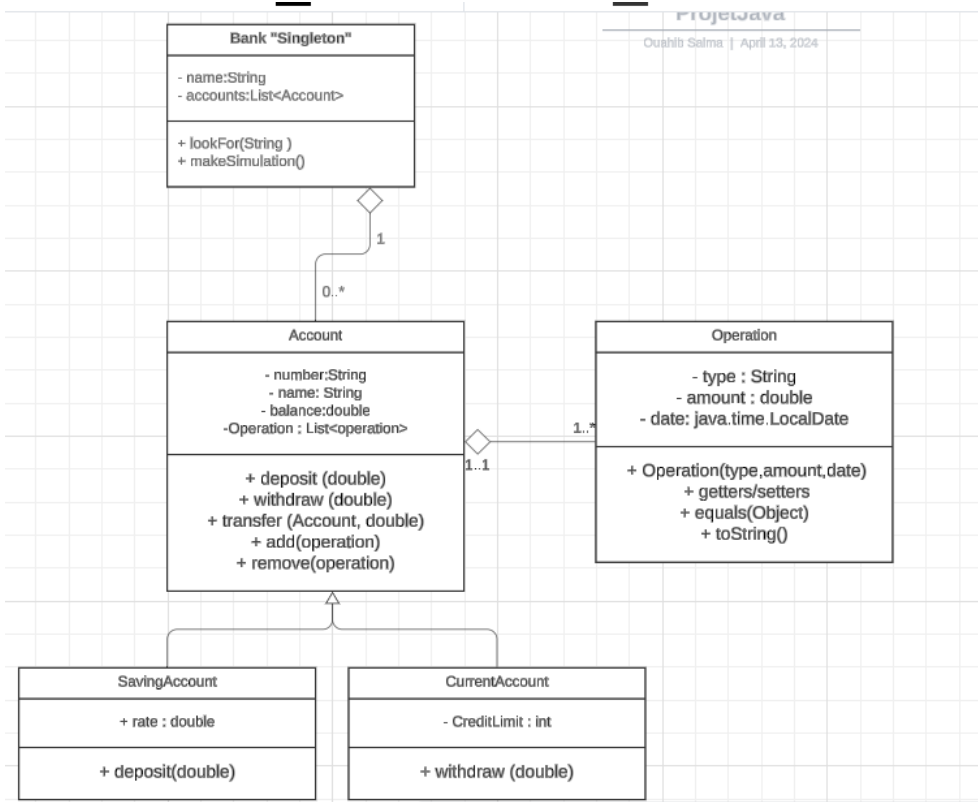
Lors de la gestion d'une banque, il est impératif de gérer un vaste ensemble de comptes, ainsi que les opérations qui y sont associées. Avec plusieurs individus effectuant différentes transactions, la gestion manuelle de ces opérations devient rapidement complexe. De plus, le nombre de clients peut être immense, ce qui rend difficile la tenue manuelle des registres et des détails des transactions.

Pour surmonter ces défis, une application développée en Java se révèle être une solution efficace. Cette application permet de rationaliser et de simplifier la gestion des comptes bancaires et des opérations qui y sont liées. Grâce à des fonctionnalités automatisées, telles que la saisie des transactions, le suivi des soldes, et la génération de rapports détaillés, cette application permet aux gestionnaires de banque de se concentrer sur des tâches plus stratégiques, tout en assurant une gestion précise et transparente des opérations bancaires.

En utilisant Java pour le développement de cette application, nous bénéficions de sa robustesse, de sa portabilité et de sa capacité à gérer des volumes de données importants de manière efficace. De plus, Java offre une large gamme de bibliothèques et de Framework qui facilitent le développement d'applications complexes, tout en garantissant la sécurité et la fiabilité des opérations bancaires.

En résumé, une application développée en Java offre une solution efficace pour gérer les opérations bancaires complexes, en automatisant les processus, en garantissant l'exactitude des données et en offrant des fonctionnalités avancées pour répondre aux besoins changeants du secteur bancaire.

## II. Analyse et conception :



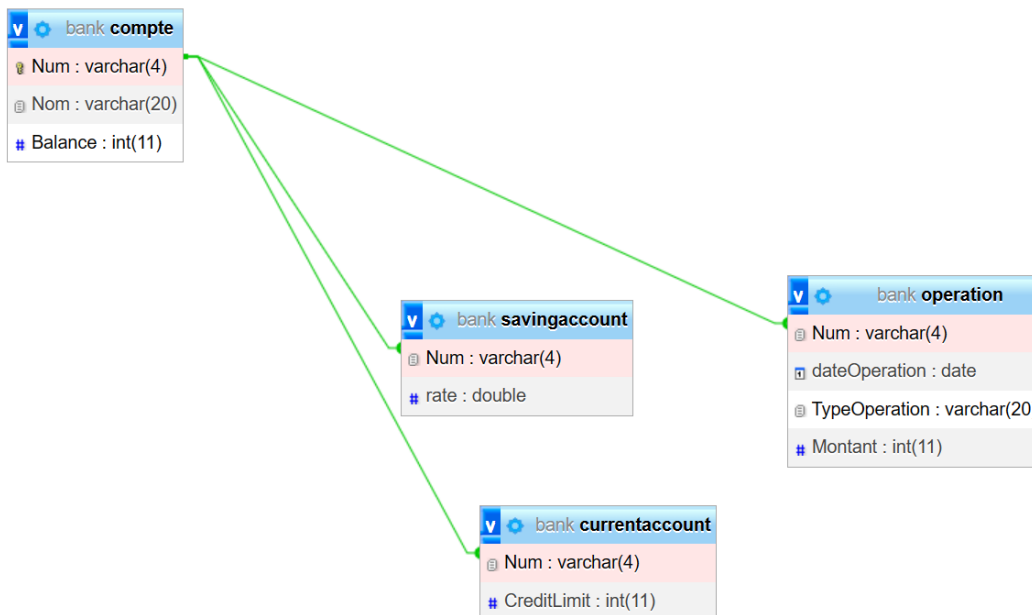
Nous avons un ensemble de classes qui modélisent une structure bancaire dans un système informatique.

La classe `Bank` et la classe `Account` sont liées par une relation de composition, ce qui signifie qu'une banque est composée de plusieurs comptes.

La classe `Operation` est également liée à la classe `Account` via une relation de composition, indiquant que les opérations (telles que les dépôts, les retraits, etc.) sont effectuées sur des comptes bancaires.

De plus, nous avons deux classes spécialisées, `SavingAccount` et `CurrentAccount`, qui héritent de la classe `Account`. Cela signifie que ces deux types de comptes partagent des caractéristiques et des fonctionnalités communes définies dans la classe `Account`, mais peuvent avoir des comportements spécifiques supplémentaires propres à chaque type de compte.

Le diagramme de classe nous aide à construire un schéma relationnel pour la modélisation des données pour un système bancaire dans une base de données MySQL utilisant XAMPP, prenant en compte les différentes entités (banque, compte, opération) et leurs interactions. Voici le schéma relationnel :



### III. Réalisation en Java :

- Le bouton CANCEL permet d'effacer le contenu des champs Login et Password
- Le bouton OK permet de valider les données entrées par l'utilisateur : On a deux types d'utilisateur Admin qui a comme Login a et comme password le vide. L'autre utilisateur client qui a comme Login v et comme password le vide.
- Le champ Status est mis à jour chaque fois qu'on clique sur OK

Après avoir entré des données valides, on autorise l'accès et les fenêtres Client et Admin (Amdin & Client frames) s'ouvrent

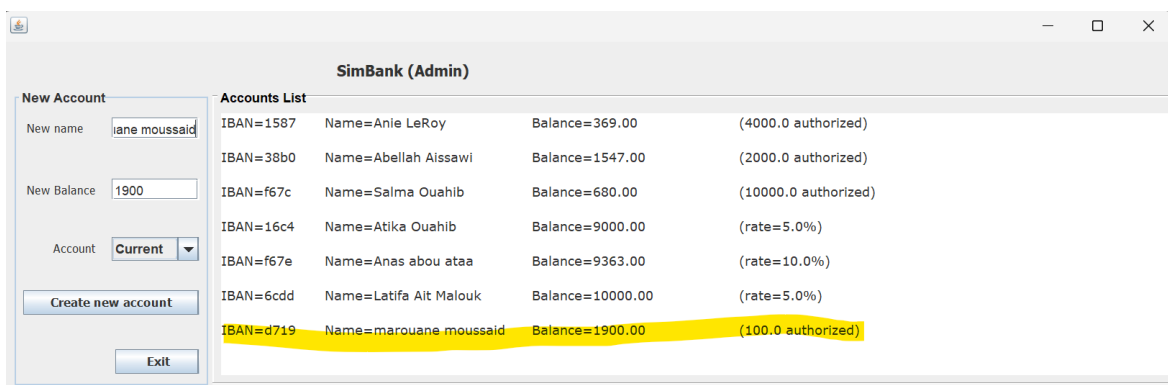
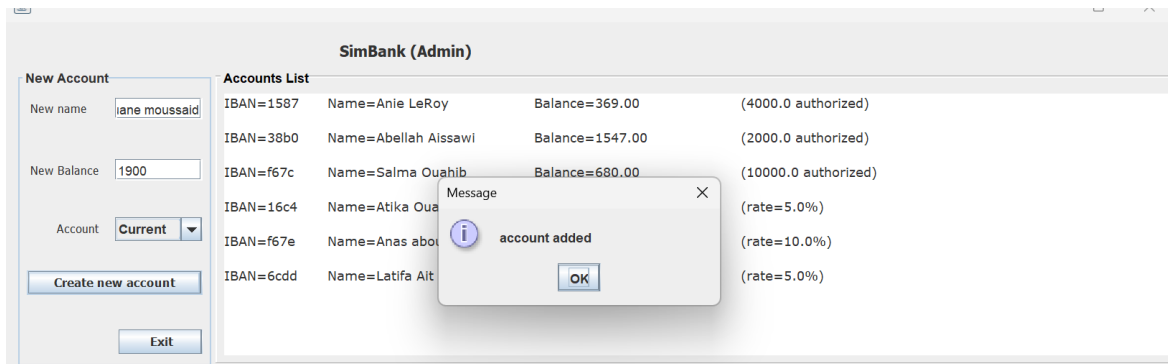
IBAN	Name	Balance	Notes
IBAN=1587	Name=Anie LeRoy	Balance=369.00	(4000.0 authorized)
IBAN=38b0	Name=Abellah Aissawi	Balance=1547.00	(2000.0 authorized)
IBAN=f67c	Name=Salma Ouahib	Balance=680.00	(10000.0 authorized)
IBAN=16c4	Name=Atika Ouahib	Balance=9000.00	(rate=5.0%)
IBAN=f67e	Name=Anas abou ataa	Balance=9363.00	(rate=10.0%)
IBAN=6cdd	Name=Latifa Ait Malouk	Balance=10000.00	(rate=5.0%)

Pour créer un nouveau compte, on remplit les champs Name et Balance, on choisit le type de compte et on clique sur Create new account.

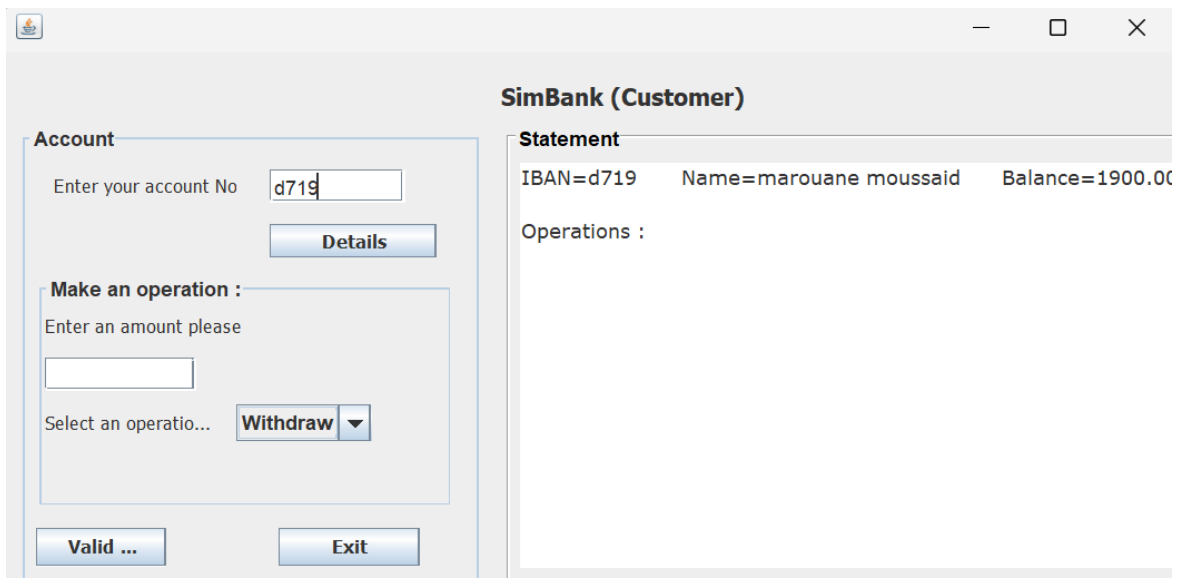
- Pour avoir des détails sur un compte, on saisit le numéro du compte et on clique sur Details.
- Si de plus on veut déposer ou retirer de l'argent, on saisit le montant, on clique sur Deposit or Withdraw et puis on clique sur Valider l'opération.

Aperçu d'utilisation :

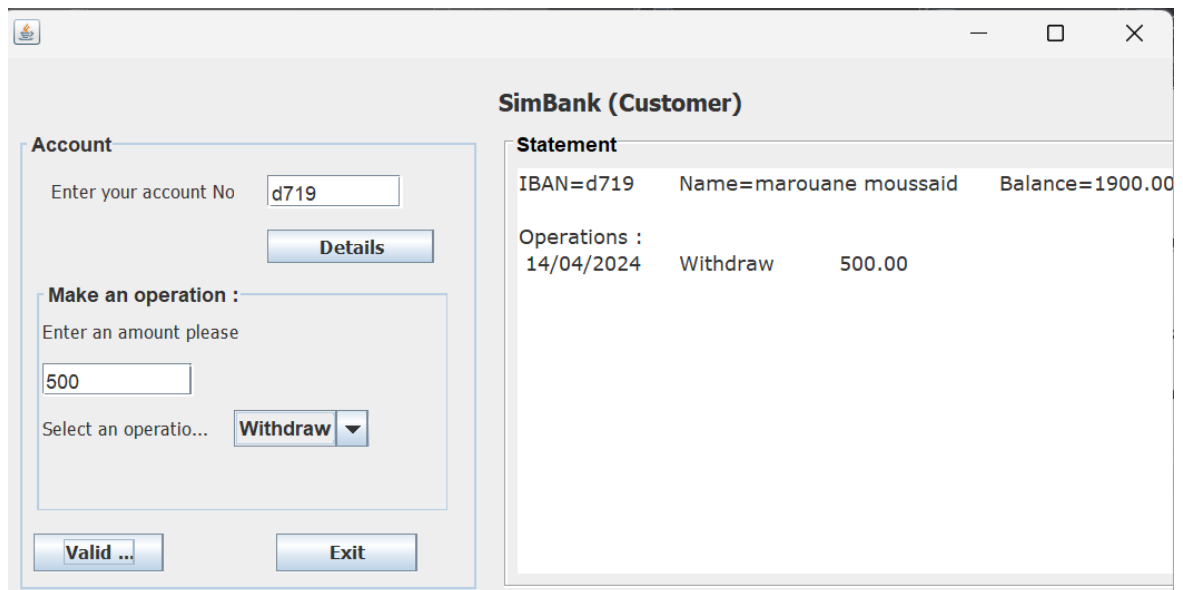
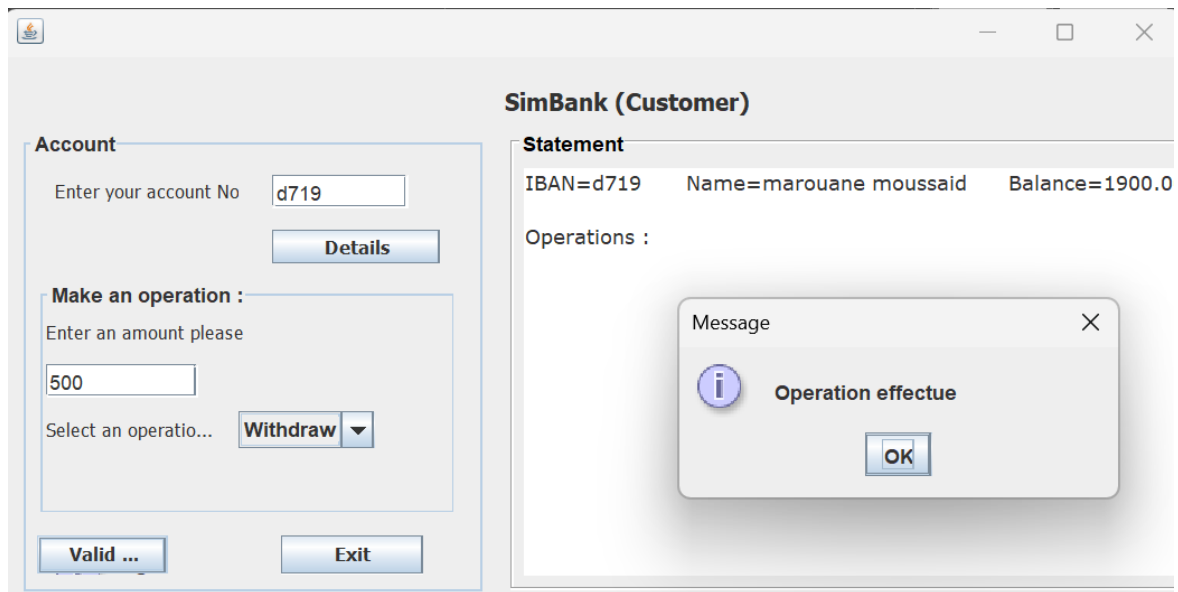
Ajout d'un compte :



On peut aussi ajouter une opération pour un compte :



C'est un nouveau compte qui n'a aucune opération, voici l'ajout d'une opération :



#### IV. Conclusion :

Ce projet nous a permis d'utiliser et d'implémenter différentes classes en approfondissant notre compréhension du langage Java et nous a offert une expérience pratique en utilisant les principes de la programmation orientée objet (POO) avec Java. L'implémentation des classes nous a permis de développer une application représentant la gestion d'une banque. De plus, nous avons utilisé des classes provenant de packages tels que 'io', 'util' et 'sql'.

Nous avons créé diverses classes pour modéliser les entités clés d'une application bancaire, comme la classe Account, CurrentAccount, SavingAccount, Operation, etc. Cette approche nous a permis de structurer notre code de manière modulaire et efficace.

Et aussi, l'utilisation des packages standards tels que `'io'` (pour les opérations d'entrée/sortie), `'util'` (pour les utilitaires) et `'sql'` (pour les interactions avec les bases de données) a enrichi notre projet. Par exemple, nous avons pu lire et écrire des données à partir de fichiers grâce à `'io'`, utiliser des collections de données optimisées provenant de `'util'`, et interagir avec une base de données pour stocker les informations des clients et des transactions grâce à `'sql'`.

Ce projet nous a donc permis d'acquérir une expérience pratique et concrète dans le développement logiciel en Java, en mettant en œuvre des concepts avancés tels que l'héritage, l'encapsulation et le polymorphisme. De plus, cela nous a sensibilisés à l'importance de l'organisation du code et de l'utilisation efficace des ressources fournies par les packages Java standard.