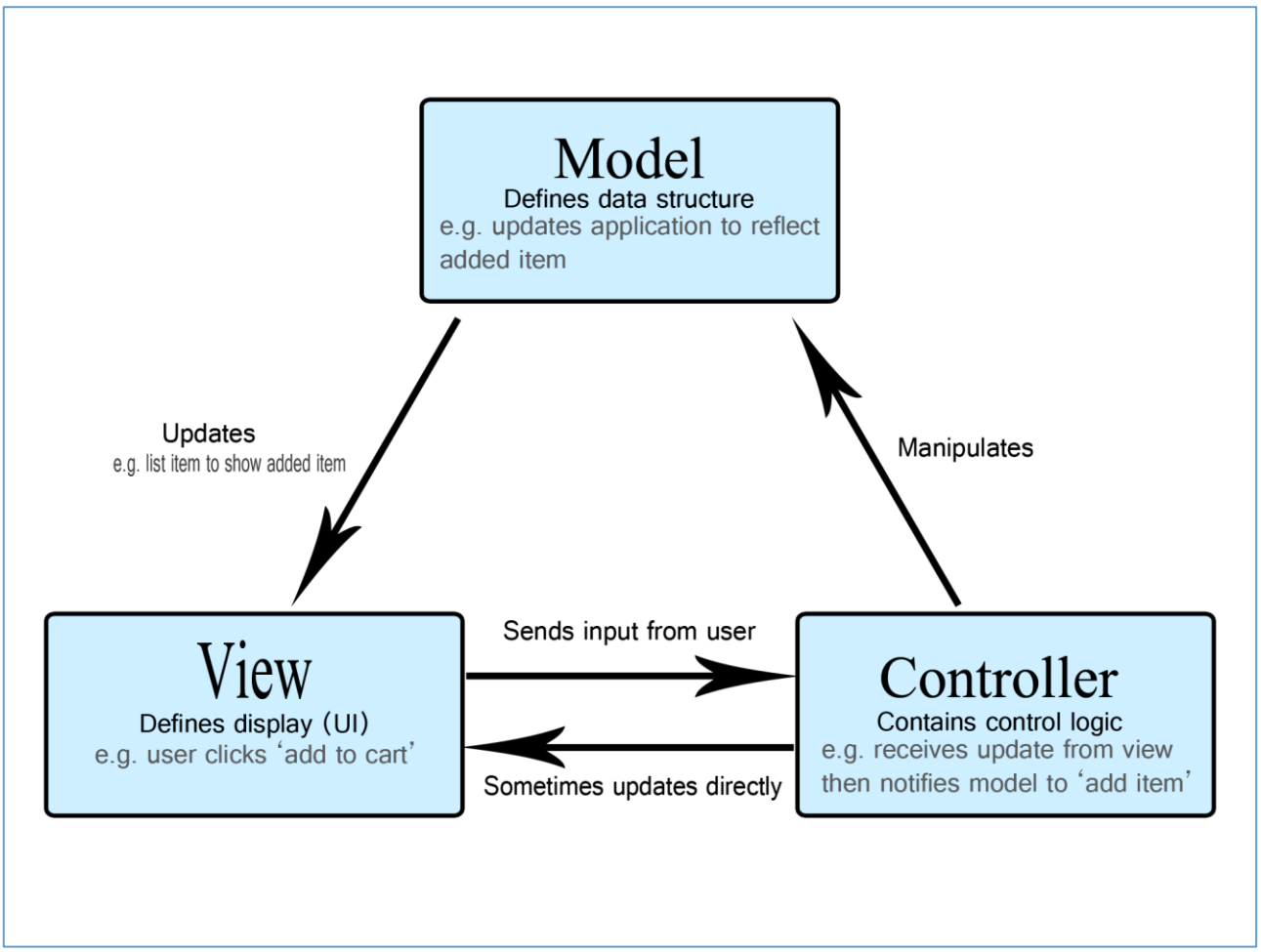
MVC : La couche Modèle

Modèle Vue Contrôleur

CONTENU

ARCHITECTURE MVC



DÉFINITION DE LA COUCHE « MODÈLE » DU PATTERN MVC

La couche Modèle d'une architecture MVC est un ensemble de composants qui contient les données ainsi que de la logique en rapport avec les données (validation, lecture et enregistrement). Il peut, dans sa forme la plus simple, contenir uniquement une simple valeur, ou une structure de données plus complexe. Le Modèle représente le contexte métier dans lequel s'inscrit l'application. Par exemple, pour un site e-commerce, la couche Modèle représentera les clients, les produits, les commandes etc…

En pratique, la couche modèle d'une architecture MVC est encapsulée au sein de la couche d'accès aux données d'une application n-tiers.

Ses principaux composants sont :

- **Les modèles**, qui sont une représentation « objet » de la structure d'une base de données.
- **Les classes d'accès aux données**, qui contiennent la logique d'interaction avec la base de données représentée par les modèles.

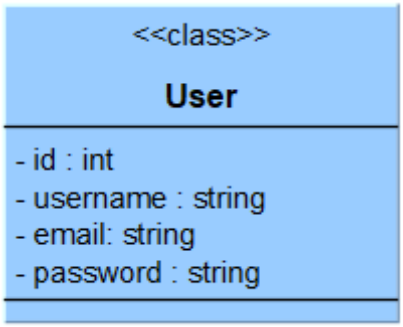
STRUCTURE D'UNE COUCHE « MODÈLE » CLASSIQUE

LES MODÈLES

Les modèles (Model) sont un ensemble de classes qui représentent généralement la structure d'une base de données.

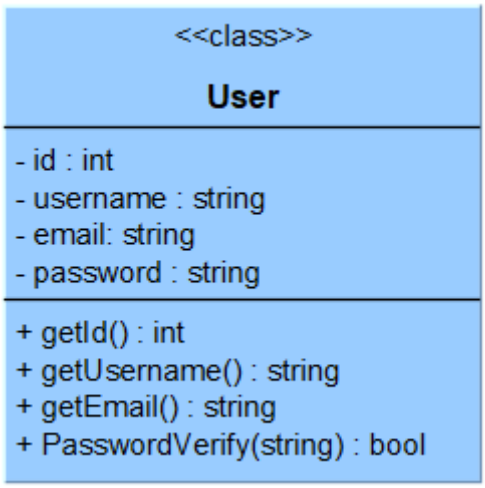
Concrètement, un modèle contient les attributs caractérisant un objet métier et est généralement consommé par la Vue (View). Dans le cas d'une application couplée à une base de données relationnelle, un modèle peut représenter la vue objet d'une table.

Exemple d'un modèle représentant un utilisateur :



Un modèle est une classe dont les méthodes se limitent généralement au fonctionnement interne du modèle comme la lecture ou la validation des données.

Dans notre modèle User, on pourrait par exemple implémenter des accesseurs ainsi qu'une méthode pour vérifier la correspondance d'un mot de passe fourni avec celui stocké :



Vous l'avez bien compris, un modèle est une représentation d'un objet métier et… juste une représentation.

Un modèle ne doit en aucun cas interagir directement avec la base de données.

Ce rôle d'interaction avec un SGBD est confié aux classes d'accès aux données.

LES CLASSES D'ACCÈS AUX DONNÉES

Les classes d'accès aux données (DAO « Data Access Object ») contiennent la logique d'interaction avec la base de données et sont consommées par les contrôleurs (couche de traitement).

Concrètement, les classes d'accès aux données contiennent les méthodes permettant d'interagir avec la base de données. Elles peuvent être divisées en 2 catégories :

- Les classes de connexion à une base de données dont le rôle est d'établir et de représenter une connexion vers une base de données.
- Les classes de requêtes dont le rôle sera d'interroger une base de données représentée par une classe de connexion.

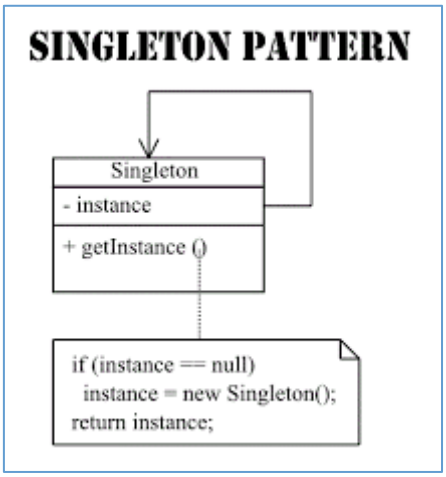
Les résultats des requêtes seront stockés dans les modèles correspondant et seront renvoyés vers la Vue par le contrôleur.

LES CLASSES DE CONNEXION

Les classes de connexion représentent une connexion vers une base de données. Elles contiennent au minimum la liaison vers la base de données.

Dans les applications Web, il est d'usage d'utiliser le patron de conception « Singleton » pour s'assurer que l'application n'établit pas de multiples connexions à la base de données, ce qui pourrait nuire aux performances.

Le pattern Singleton garantit qu'une seule instance d'une classe n'est active à un instant T. Le constructeur d'une classe Singleton est privé, l'instance unique est stockée dans un attribut statique privé et l'accès cette l'instance unique se fait au moyen d'une méthode statique publique nommée « getInstance() » par convention (mais il est possible de la nommer comme bon vous semble).



Le patron de conception Singleton ne fait pas l'unanimité au sein de la communauté des développeurs. De ce fait, d'autres approches existent.

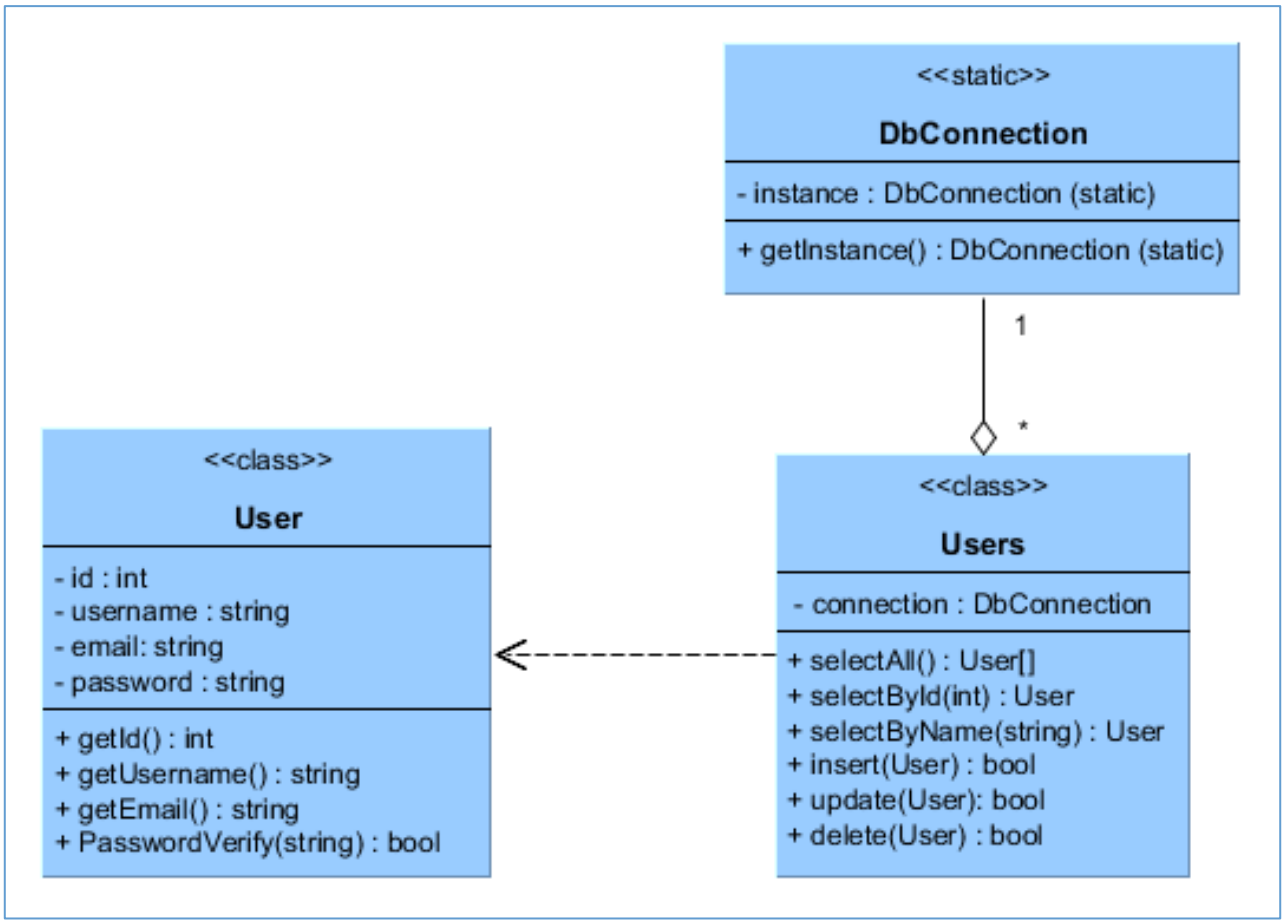
Par exemple, il est possible d'utiliser l'injection de dépendance. Dans ce cas, l'objet de connexion est stocké dans un gestionnaire de connexion et est « injecté » dans le constructeur des classes de requêtes. Cette approche est légèrement plus complexe à construire mais permet plus de souplesse lorsque l'application sera amenée à évoluer.

LES CLASSES DE REQUÊTES

Les classes de requêtes sont les plus complexes à implémenter. Leur rôle central impose une certaine rigueur puisque qu'elles assurent le lien entre les différentes couches de l'application. Elles sont principalement consommées par les contrôleurs. Leurs méthodes envoient des requêtes à la base de données et retournent les résultats dont le format dépendra du type de requêtes envoyées.

- Les requêtes de lecture renvoient un modèle (ou une collection de modèles) représentant le résultat.
- Les requêtes d'écriture renvoient le nombre d'enregistrements affectés ou, selon l'implémentation, un booléen qui indiquera si l'opération a réussi.

Reprenons notre utilisateur et complétons le diagramme en y ajoutant une classe de connexion et la classe qui gérera les requêtes vers la base de données où l'ensemble des utilisateurs est stocké.



La classe « Users » interroge la base de données avec ses méthodes en utilisant la connexion « DbConnection ».

Les méthodes de « lecture » de la classe « Users » retournent des instances de la classe « User » qui seront manipulées par le contrôleur et la vue.

Il reste maintenant à coder tout çà 😉

--- FIN DU DOCUMENT ---