

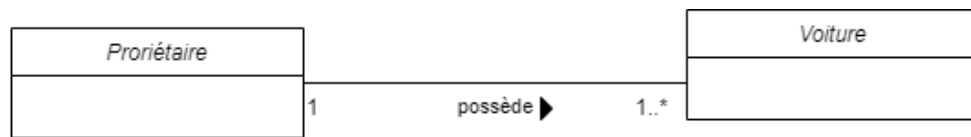
Exercices : Modèle du domaine partie 1

Question 1

Faire les associations et les multiplicités de ces modèles incomplets. Vous pouvez aussi mettre les associations de généralisation et des mots qui expliquent la relations (fait, paye, achète, enseigne, suis, etc.) avec une flèche noire ►

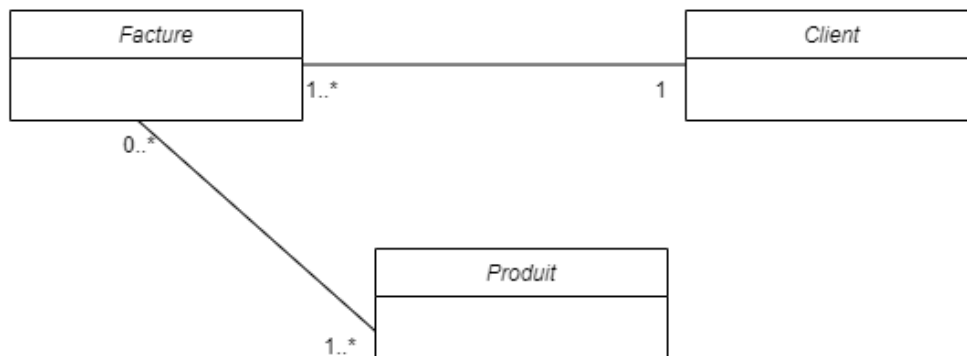
a)

Réponse



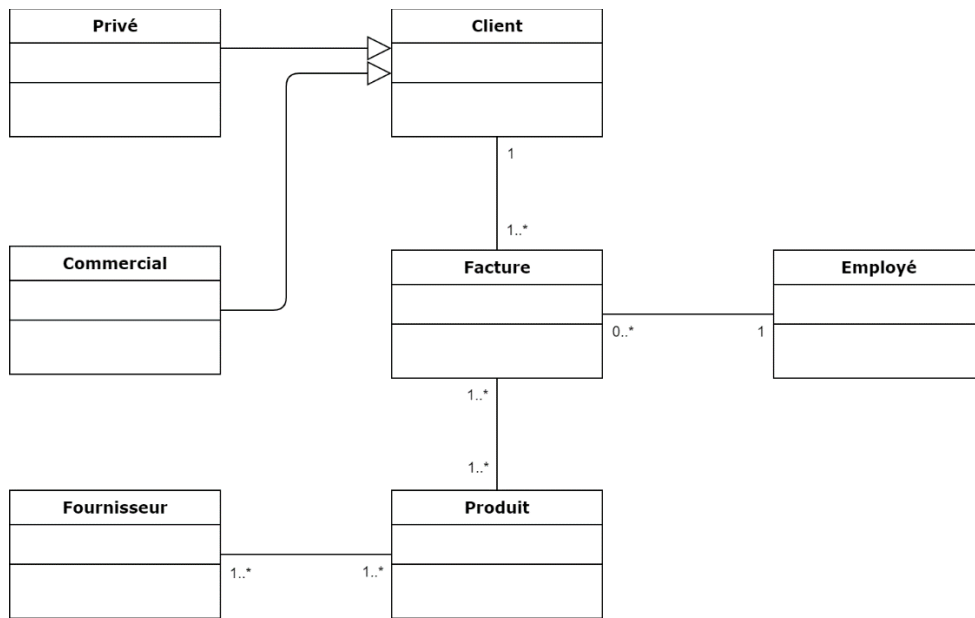
b)

Réponse :



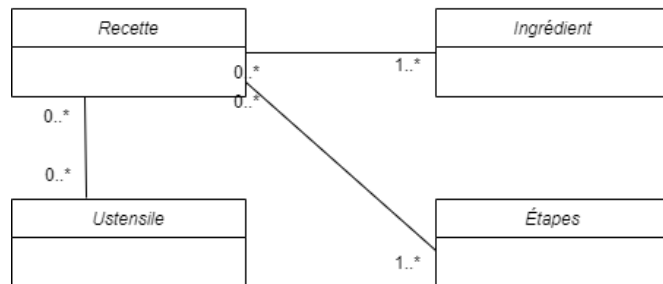
c)

Réponse :

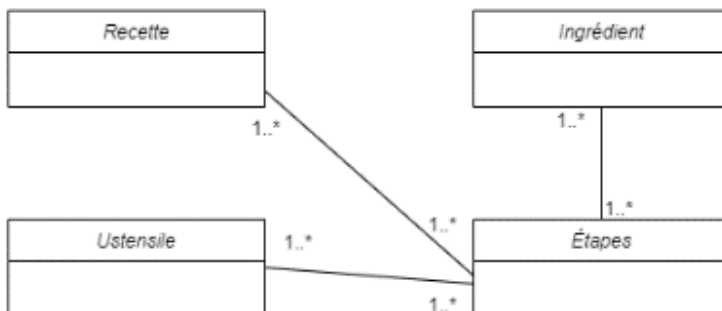


d)

Réponse :



Ou bien :



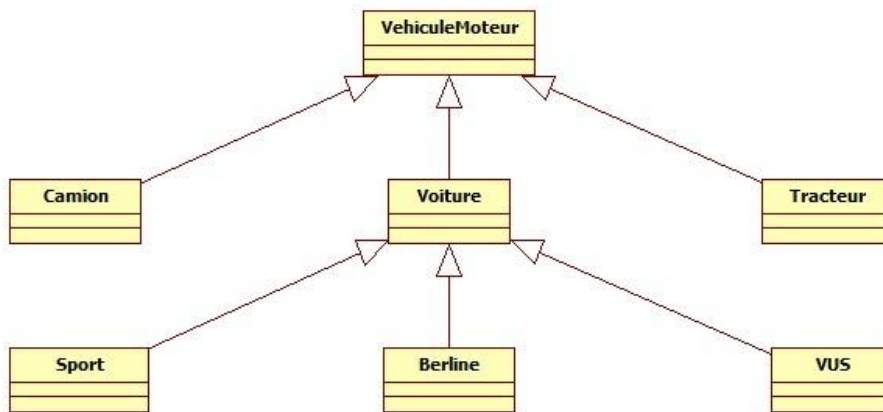
Question 2

- Discutez de vos réponses avec vos collègues et le prof

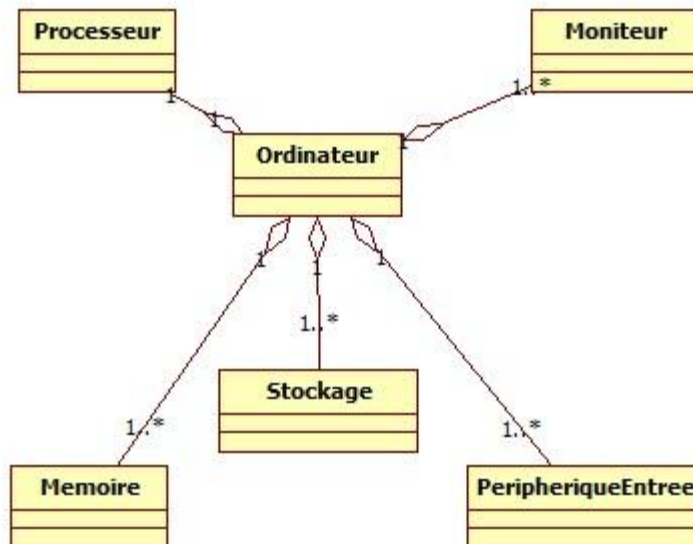
Question 3

Conceptualiser (Modèle du domaine) les situations suivantes :

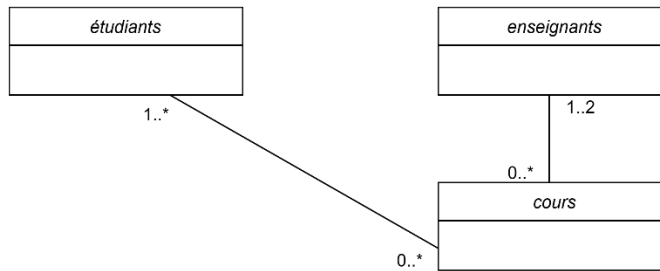
- a) Une compagnie possède plusieurs types de véhicules à moteur : des camions, des voitures et des tracteurs. Pour les voitures, il existe des modèles sport, des berlines et des VUS.



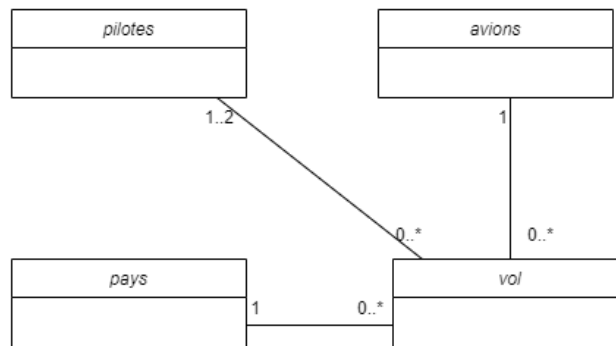
- b) Pour constituer un ordinateur de bureau, il faut, entre autres, un processeur, un ou des périphériques d'entrée, du stockage et un ou des écrans.



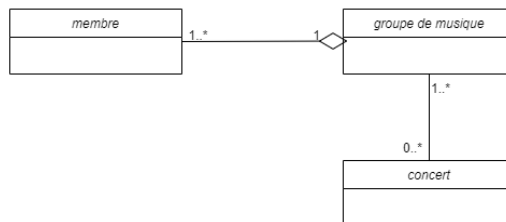
- c) Dans un système scolaire, on a des étudiants qui suivent des cours qui sont enseignés par un ou 2 enseignants (Team teaching). Les cours n'ont pas lieu sans enseignants ni sans étudiants.



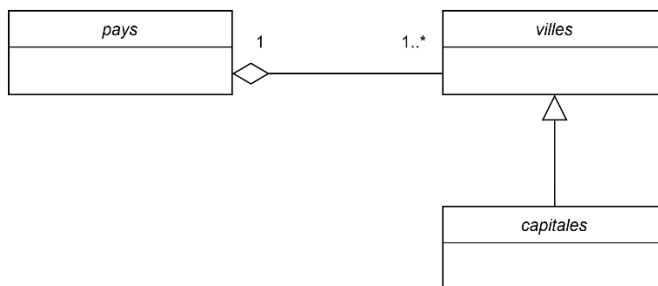
- d) Réfléchir aux concepts, entités et relations dans un aéroport. Concentrez-vous sur les vols, les avions, les pays et les pilotes et non sur la logistique de douane ou de valises ou d'employés.



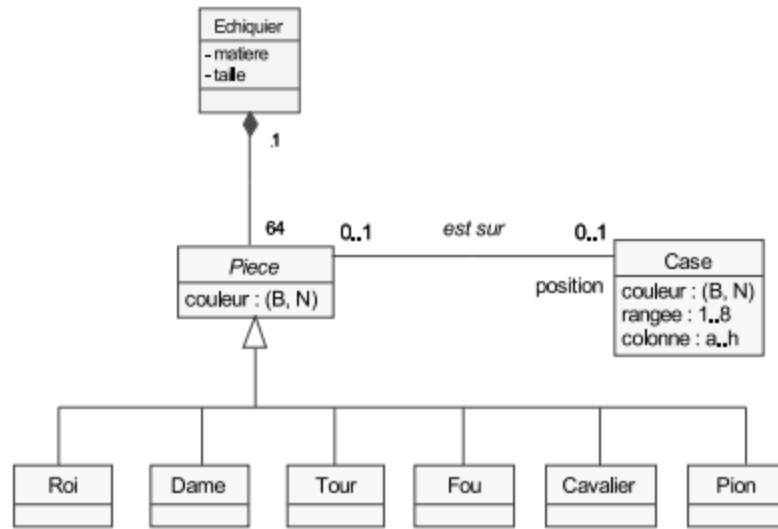
- e) Faire le modèle du domaine d'un groupe de musique avec des musiciens et des concerts.



- f) Faire le modèle pour les concepts de pays, ville et capitale. Ajouter les attributs nom, langue, monnaie, nombre d'habitants, superficie aux endroits appropriés.



- g) Faire le modèle d'un jeu d'échec, qui contient des cases, des pions (pions, roi, dame, tour, fou, chevalier).



Question 4

Voici un scénario. Extraire votre compréhension des concepts en un modèle du domaine. Inclure les entités, les associations et les multiplicités qui vous semblent logiques. On ne les a pas vus en détail, mais les attributs aussi peuvent être ajoutés (détails sous la classe). Certaines questions ouvertes peuvent demeurer!

Scénario :

Vous voulez aider un ami qui aimerait une application pour son entreprise en démarrage. Il élève et vend des **Pokémons**. Il voudrait permettre à ses **clients** de pouvoir se connecter à l'application pour voir le **catalogue** de Pokémons et de **produits** qu'il a en disponibilité.

Les **utilisateurs non connectés** peuvent quand même voir le catalogue. Par contre pour ajouter des **items** au **panier**, il faudra que l'utilisateur crée un **compte**. Il y a deux façons pour ajouter des Pokémons au panier soit directement à partir du catalogue ou en visualisant une **fiche** d'un Pokémon.

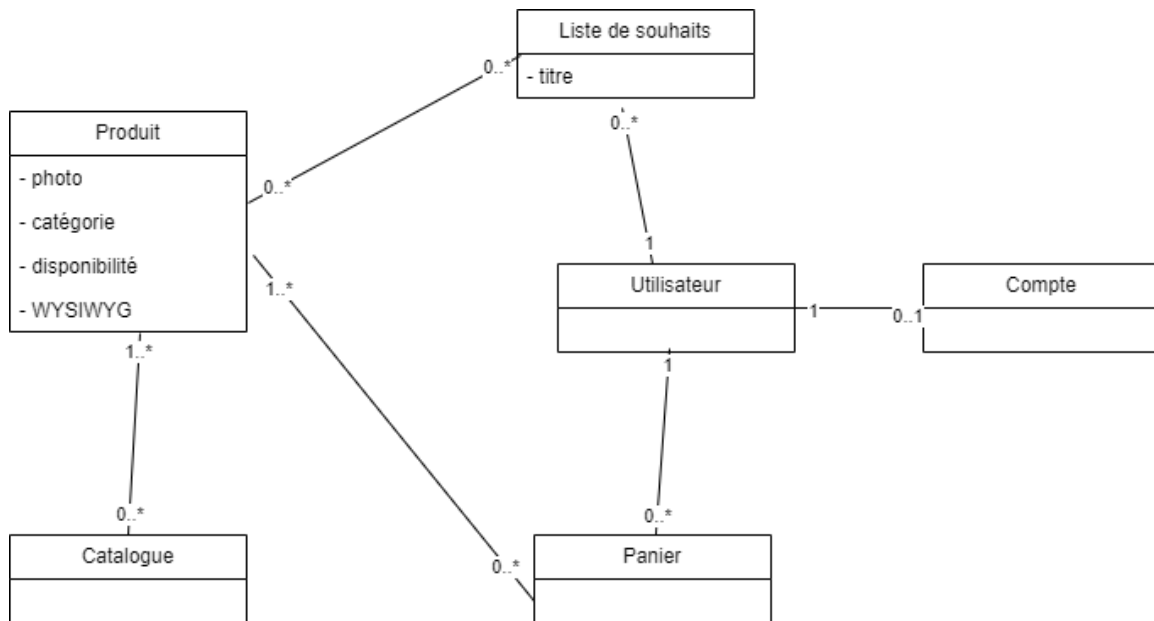
L'utilisateur pourra aussi gérer des **listes de souhaits** qu'il pourra gérer à sa guise en y ajoutant ou y retirant des items qu'il aimerait suivre. Chaque liste de souhaits a un **titre** pour faciliter la classification des items.

Votre ami doit aussi pouvoir gérer le catalogue en y faisant les modifications usuelles que l'on retrouve sur ce genre d'application soit la gestion des produits et des utilisateurs. Il lui sera aussi possible de **catégoriser** les produits pour en faciliter la recherche.

Les items peuvent être WYSIWYG (*what you see is what you get*), i.e. que le Pokémon qu'il voit sur le site sera celui qui lui sera vendu. Ainsi, il faudra que votre ami insère une **photo** lorsqu'il entrera un item dans cette catégorie.

Concernant les utilisateurs, il ne veut pas gérer les mots de passe de tous et de chacun, il aimerait que ceux-ci utilisent un système d'authentification tierce tel que Google, Microsoft, etc.

Votre ami pourrait écrire un roman sur tous les besoins qu'il a, mais vous vous limitez à ce qui est écrit pour cette première ébauche.



Détails de la réflexion :

- « Fiche » n'est pas choisi dans les entités, puisqu'elle est une représentation visuelle du produit lui-même. Elle présente les attributs et met un bouton d'action, ce n'est donc pas une entité statique différente du produit.
- Le système d'authentification n'a pas été représenté non-plus, il ne serait pas implémenté du côté de l'application et on ne manipule pas ces données. C'est pourquoi, même si l'utilisateur a un mot de passe, on ne le gèrera pas dans le code de classe du côté de l'application, alors on ne le met pas.
- L'utilisateur connecté à un compte, celui non-connecté n'a pas de compte. Ici, la supposition a été faite qu'un utilisateur ne peut pas avoir 2 comptes, ça serait une question ouverte.
- Il peut y avoir beaucoup plus d'attributs qui n'ont pas été nommés, mais qui seraient logiques à proposer au client dans les entités.
- Les multiplicités ont été attribuées pour permettre toutes les possibilités, les possibilités minimales et certaines suppositions ont été faites pour l'unicité (par exemple, qu'un panier ne peut appartenir qu'à un seul utilisateur).