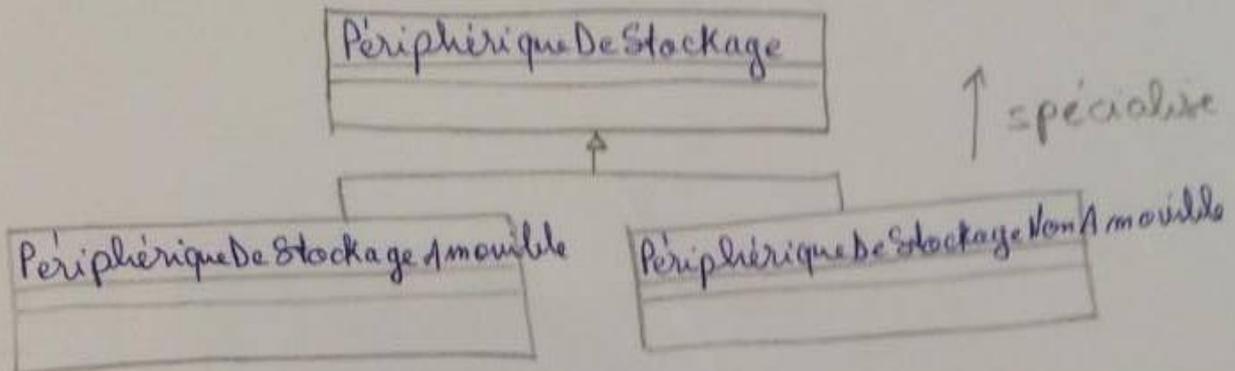
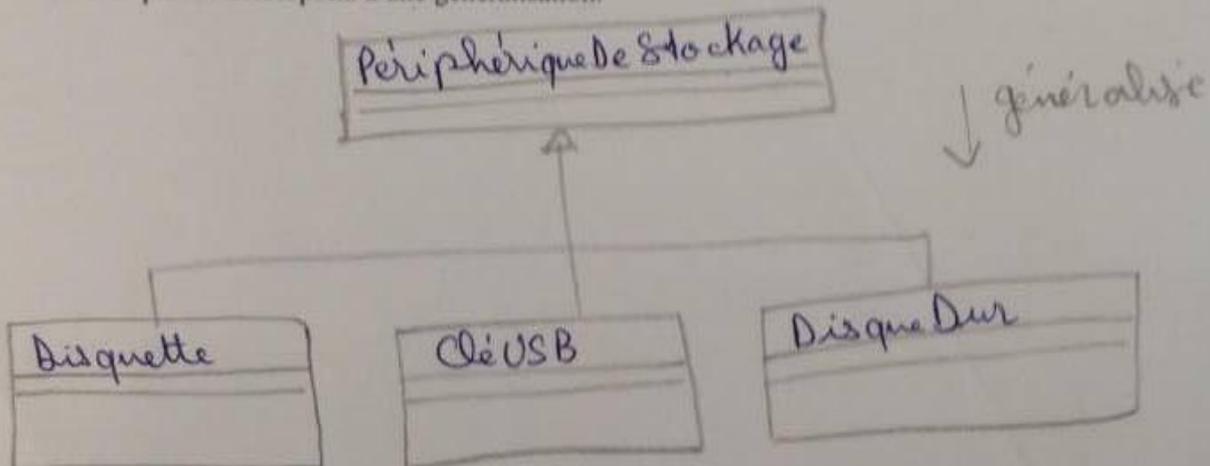


Corrigé (généralisation et spécialisation) :

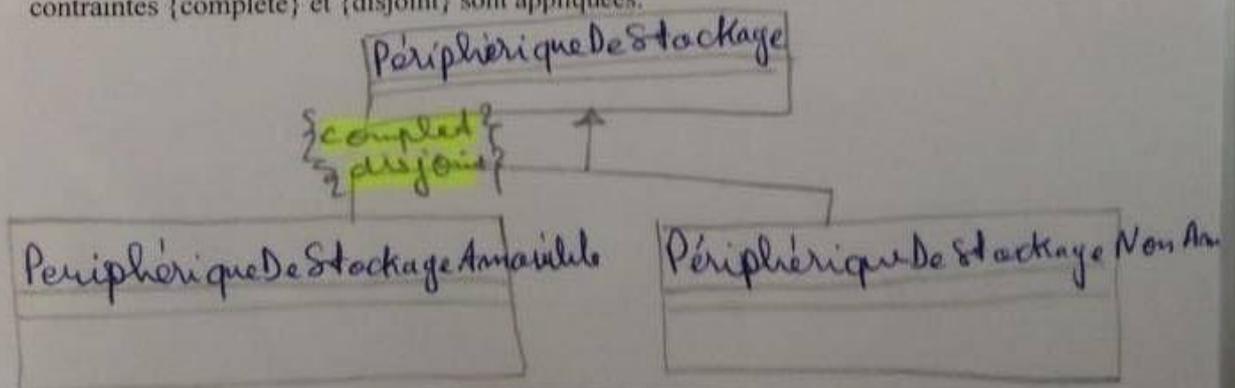
1. La première phrase correspond à une spécialisation. Les deux sous-classes **PériphériqueDeStockageAmovible** et **PériphériqueDeStockageNonAmovible** spécialisant la surclasse **PériphériqueDeStockage**.



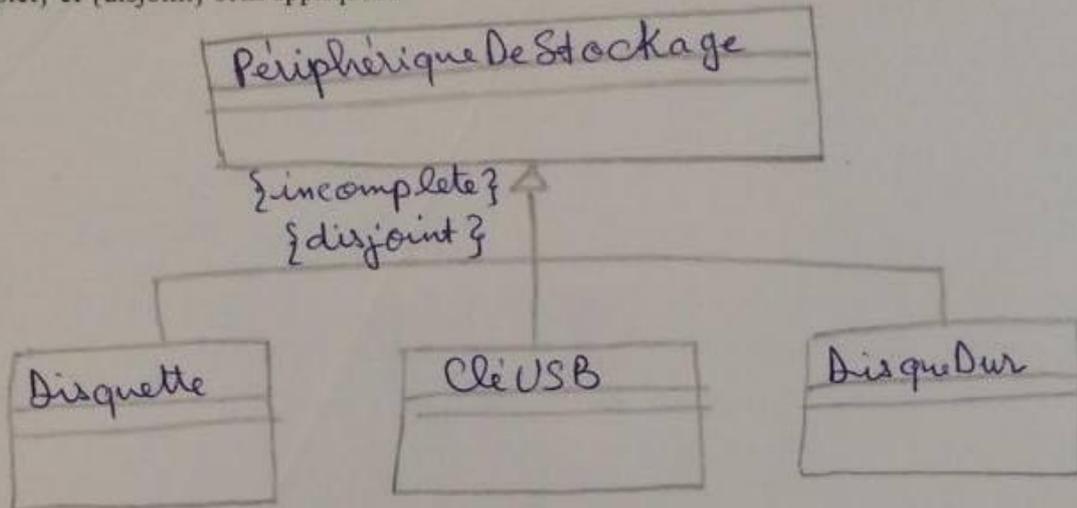
La seconde phrase correspond à une généralisation.



2. Dans le premier cas, l'ensemble des instances des deux sous-classes couvre les instances de la surclasse et les deux ensembles sont disjoints. De ce fait, les contraintes {complete} et {disjoint} sont appliquées.



Pour la second cas, l'arbre de généralisation est incomplet puisqu'il existe d'autres périphériques de stockage tels que les disques ZIP, etc. par ailleurs, les sous-classes sont disjointes (les deux sous-classes n'ont pas d'instances communes). De ce fait, les contraintes {incomplet} et {disjoint} sont appliquées.



#### Exercice N°6 :

Un message électronique comporte un titre que l'adresse du destinataire. Il est composé d'un entête et d'un corps. Il peut contenir éventuellement une ou plusieurs pièces jointes.

- 1) Tracez le diagramme de classes correspondant à cette description.
- 2) Pensez à distinguer les relations de composition des agrégations.

#### Corrigé (Agrégation et composition)

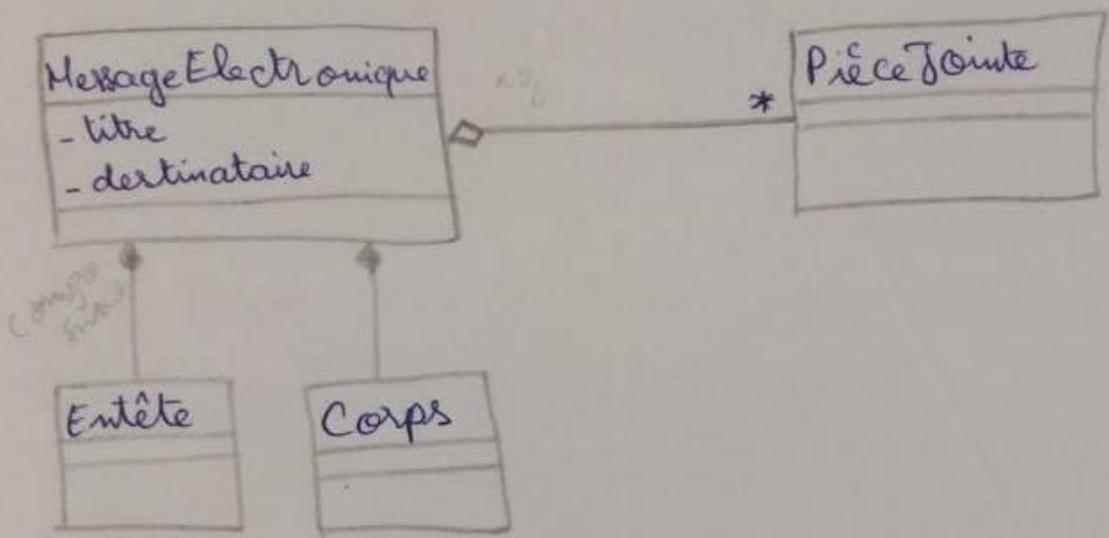
Il convient de déterminer la nature des relations entre les différentes classes du modèle. Les relations entre la classes **MessageElectronique** et les classes **Entête**, **Corps** et **PièceJointe** sont au minimum des agrégations. Afin de vérifier s'il s'agit de relations de composition, deux critères sont à vérifier :

- La cardinalité maximale au niveau du composé est de un : c'est le cas pour l'en-tête et le corps qui ne peuvent appartenir qu'à un seul message. La pièce jointe, quant à elle, peut être partagée par plusieurs messages.
- La suppression de l'objet composé entraîne celle de ses composants : c'est également le cas pour l'en-tête et le corps du message.

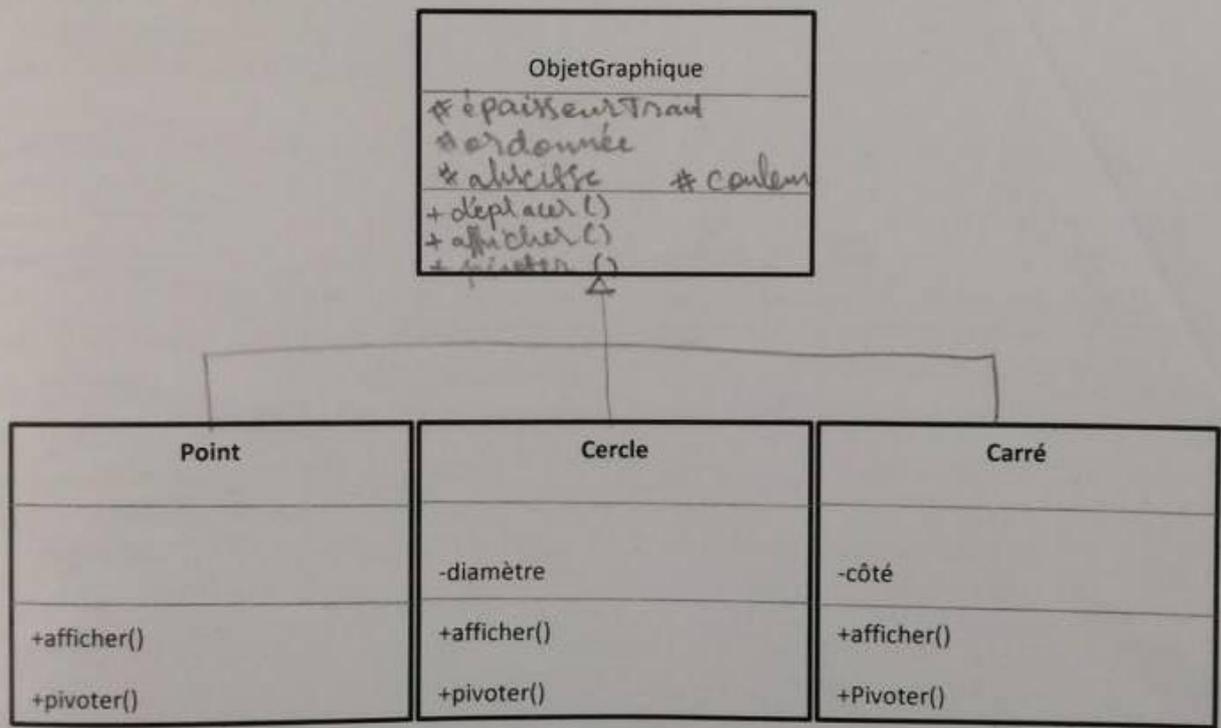
Les relations entre la classe **MessageElectronique** et les classes Entête et Corps sont des compositions.

Un message peut comporter plusieurs pièces jointes ou ne pas en avoir, aussi la cardinalité est \*.

Le titre et le destinataire sont des informations spécifiques à chaque message. Ce sont donc des attributs de la classe **MessageElectronique**.



Exercice N°7 :



Etant donné le diagramme de classe ci-dessus représentant des objets graphiques, répondez aux questions suivantes :

- 1- L'attribut **diamètre** est-il accessible pour un objet **Carré** ?
- 2- Un objet **Cercle** possède-t-il un attribut **couleur** ?
- 3- Peut-on appliquer la méthode **déplacer** à un objet **Point** ?
- 4- Grâce à quelle notion de l'approche objet, la méthode **pivoter** peut-elle être présente dans toutes les classes du diagramme et à quoi sert cette notion ?

- 5-
- Quelle hypothèse doit-on considérer pour rendre la classe **ObjetGraphique** abstraite ?
  - Quel est l'intérêt d'éviter que cette surclasse soit concrète ?
  - Quelle modification doit-on apporter au diagramme de classe ?
- 6- Proposez une extension du modèle permettant d'obtenir un objet graphique composé de plusieurs autres objets graphiques.

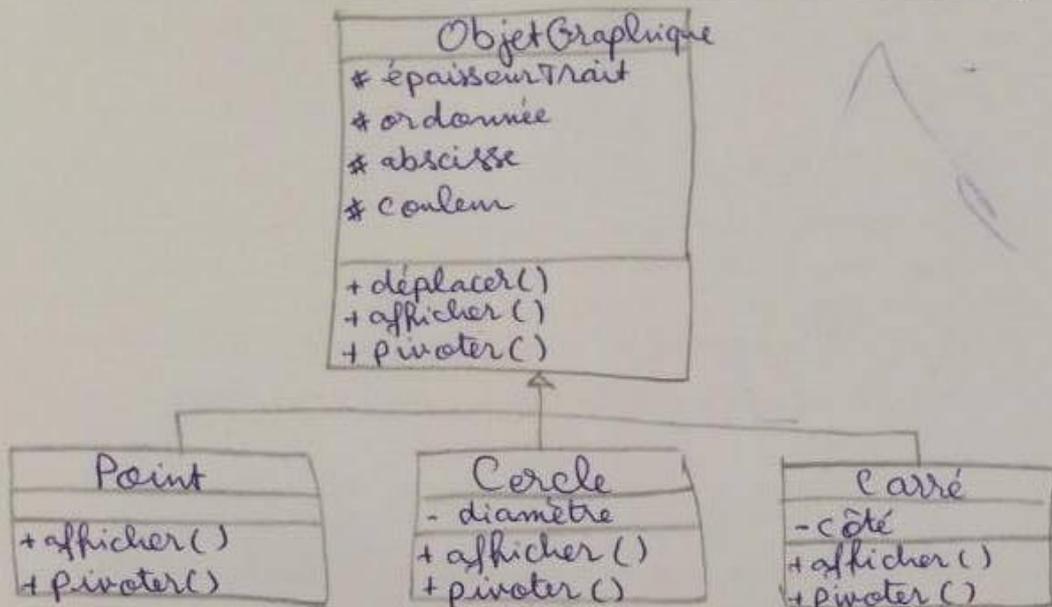
### Corrigé (Héritage, encapsulation et polymorphisme)

- L'attribut **Diamètre** est un attribut **privé** de la classe **Cercle**. Grâce au **principe d'encapsulation**, il n'est accessible qu'à des objets de sa classe. Un objet **Carré** n'a donc **pas accès à cet attribut**.
- La classe **Cercle** est une **sous-classe** de la classe **ObjetGraphique**. Elle **hérite par conséquent** de tous les **attributs de sa surclasse**. L'attribut **couleur** est donc un attribut de **Cercle** hérité de la classe **ObjetGraphique**.
- De la même manière que pour la question précédente, une méthode **est héritée** par les sous-classes d'une classe. La méthode **déplacer** est donc **héritée** dans la classe **Point**.
- La méthode **pivoter** est présente dans toutes les classes du diagramme et y possède un comportement différent grâce à **la notion de polymorphisme**.
- A.** une classe abstraite est une classe qui ne peut pas posséder d'instances directes. Afin de rendre la classe **ObjetGraphique** abstraite, il faut considérer que cette classe ne décrit que partiellement des objets graphiques. la solution consiste à rendre abstraites les méthodes **afficher** et **pivoter**. En effet, il n'est pas possible de dessiner ou de pivoter un objet graphique sans connaître sa nature précise, information qui est fournie par l'une des sous-classes d'**ObjetGraphique**. Par la suite, il est possible de créer de nouvelles sous-classes d'**ObjetGraphique** pour obtenir de nouveaux objets en plus des points, des cercles et des carrés.

**B.** il est préférable d'éviter que la classe **ObjetGraphique** soit concrète sinon il est obligatoire de devoir spécifier le comportement des méthodes **afficher** et **pivoter**, en donnant alors un comportement par défaut, ce qui ne constitue pas une solution satisfaisante.

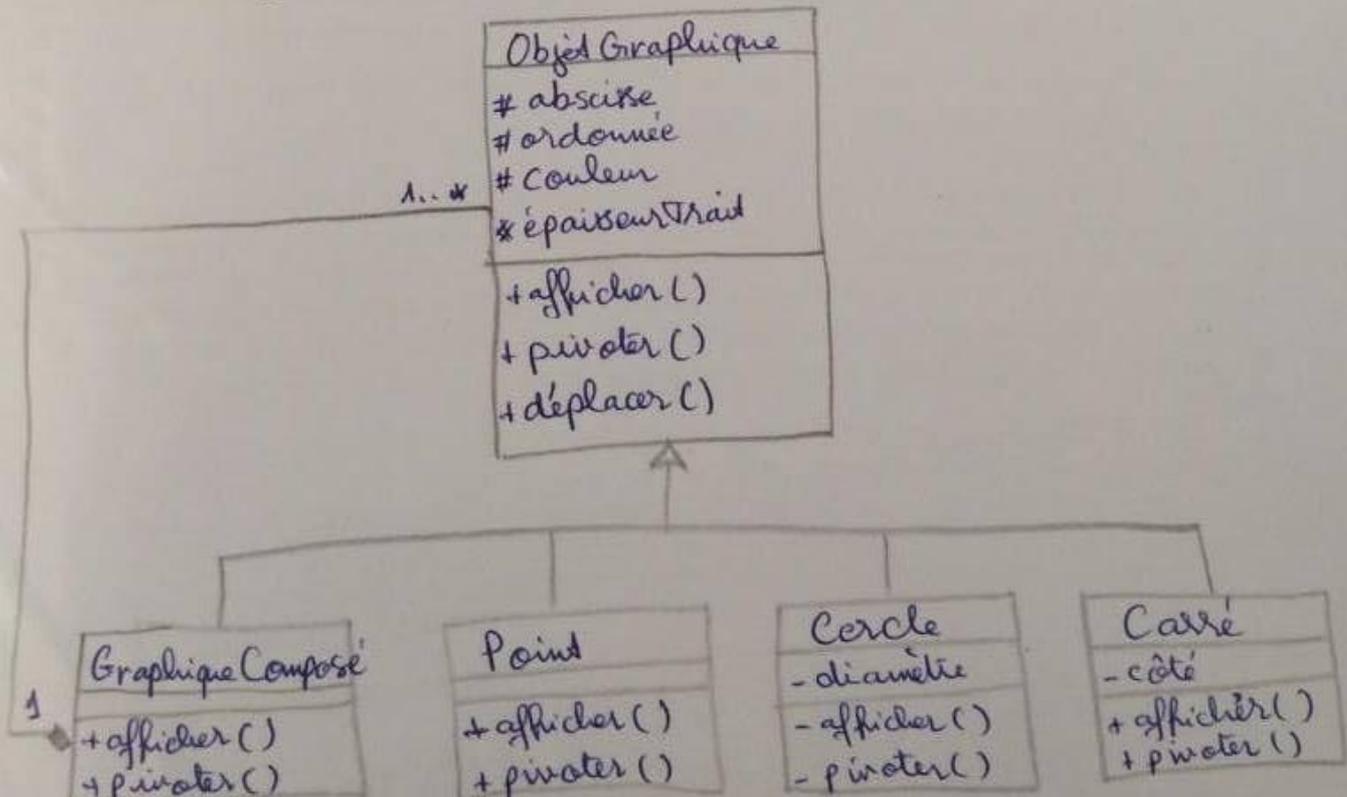
**C.** En UML, le nom d'une classe abstraite est mis en italique ou en conférant le stéréotype « abstract » à la classe. Les méthodes **afficher** et **pivoter** deviennent aussi abstraites puisqu'elles sont définies dans les sous-classes correspondantes.

Le diagramme ainsi modifié est le suivant :



6. Les objets graphiques composés sont représentés par une classe **GraphiqueComposé**. Chaque objet graphique composé est alors typé comme un objet graphique. La classe **GraphiqueComposé** est donc liée à la classe **ObjetGraphique** par une relation d'héritage. Un objet graphique composé est composé d'autres objets graphiques. La suppression de celui-ci entraîne la suppression de ses composants. Il y a donc d'une relation de composition entre les classes **GraphiqueComposé** et **ObjetGraphique**.

Le diagramme de classes est modifié en conséquence comme suit :



**Exercice N°1 :**

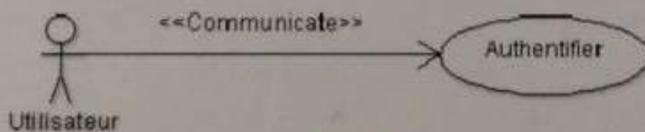
Il s'agit de décrire le cas d'utilisation de l'authentification d'un utilisateur à un système informatique. Cette authentification s'effectue de façon simple par la saisie d'un nom et d'un mot de passe.

- 1) Représentez, en UML, le cas d'utilisation de l'authentification avec acteur qui interagit avec ce dernier, sans spécifier les détails.
- 2) Introduisez, dans un diagramme du cas d'utilisation, la saisie du nom et celle du mot de passe ainsi que la vérification de ces données.
- 3) Ajoutez la saisie d'un code complémentaire après celle du mot de passe. Ce code complémentaire est optionnel et n'est destiné qu'aux utilisateurs ayant besoin d'une sécurité accrue.

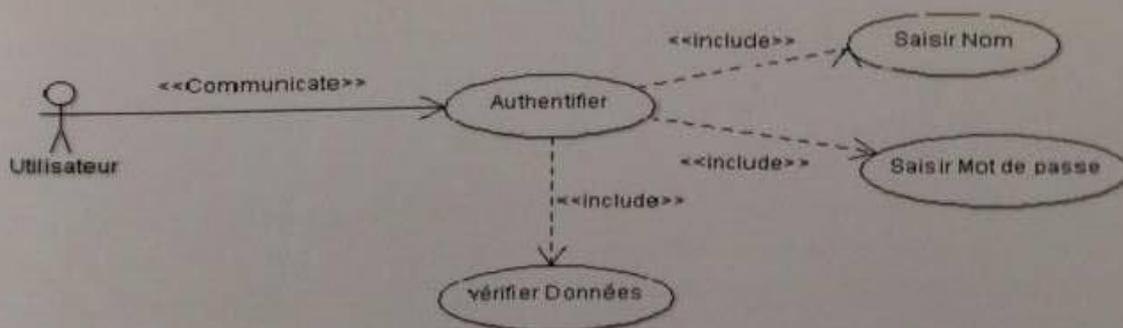
**Corrigé type :**

1. L'acteur primaire et le système sont assez faciles à identifier :
  - a. L'utilisateur constitue l'acteur primaire
  - b. Le système informatique ici ne possède qu'un seul cas d'utilisation à savoir **Authentifier**.

La représentation en UML se trouve ci-après :

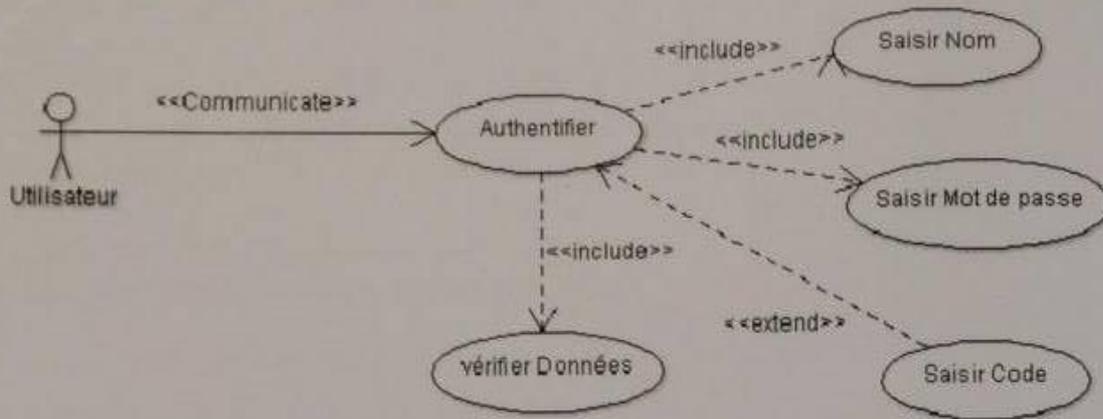


2. L'authentification requiert la saisie du nom et du mot de passe ainsi que la vérification de ces données. Celles-ci vont être représentées par un cas d'utilisation de type sous-fonction inclus dans le cas d'utilisation Authentifier. La relation utilisée est donc celle d'inclusion comme illustrée dans la représentation UML ci-après :



Notons que la relation d'inclusion est représentée par une flèche munie du stéréotype « include ».

3. La saisie du code complémentaire donne lieu à un nouveau cas d'utilisation de type sous-fonction qui étend le cas principal de façon optionnelle. La représentation est donnée à la figure suivante :



Notons que la relation d'extension est représentée par une flèche munie du stéréotype «extend». L'orientation de la flèche est inversée par rapport à celle de l'inclusion.

### Exercice N°2 :

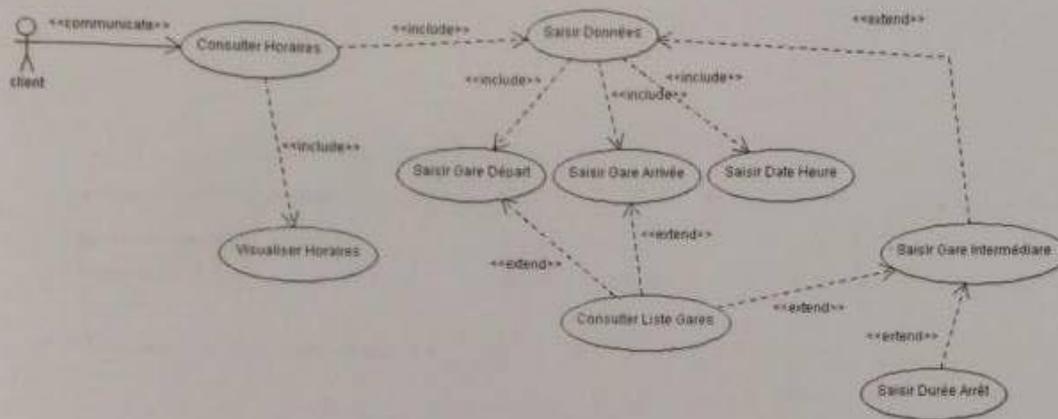
Le but de l'exercice est de décrire les exigences fonctionnelles d'un site Internet de consultation d'horaires de train, avec en option la possibilité d'acheter un billet correspondant à l'heure sélectionné.

- 1) Représentez en UML le cas d'utilisation de consultation avec la description de la requête (choix des gares de départ et d'arrivée, date et heure de départ, possibilité d'avoir une gare intermédiaire avec éventuellement une durée et la description de la visualisation des horaires calculés par le système. Pour le choix des gares, il est possible de consulter une base de données contenant toutes les gares du réseau ferroviaires).
- 2) Introduisez la possibilité d'acheter un billet correspondant à l'heure préalablement sélectionné parmi ceux visualisés. N'oubliez pas que le système demande le tarif choisi pour le billet. Ne décrivez pas les détails du choix du tarif ni du paiement, ni de l'envoi du billet.

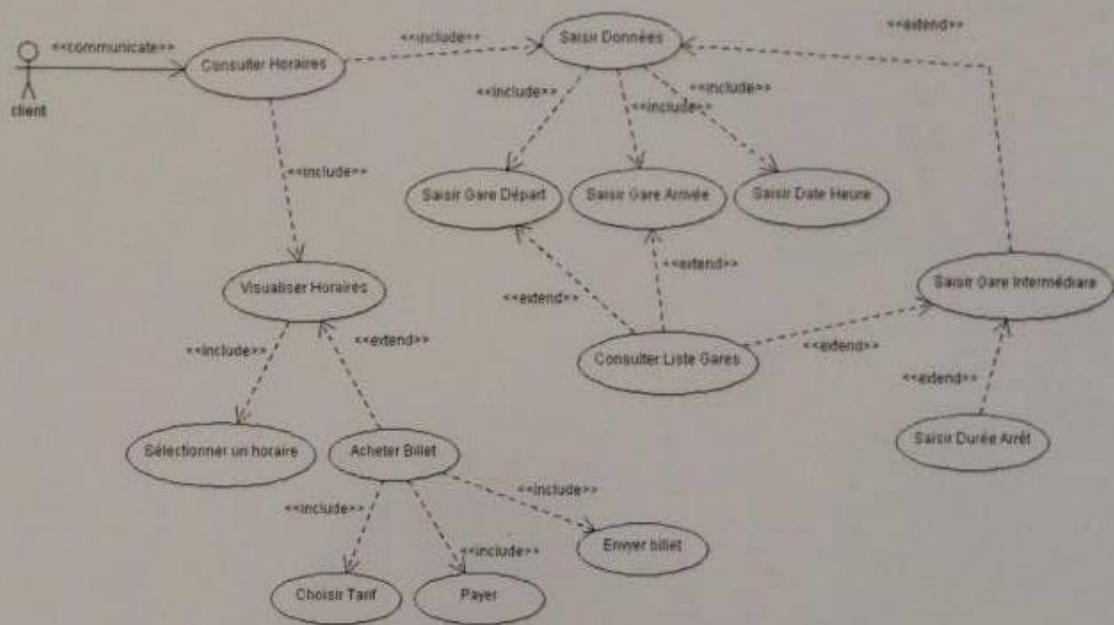
### Corrigé type :

1. Le cas d'utilisation fait apparaître le client (l'utilisateur qui consulte les horaires) et le cas d'utilisation **Consulter Horaires**. Celui-ci inclut les sous-fonctions **Saisir Données** et **Visualiser Horaires**. La sous-fonction Saisir Données est décrite en détail. Les données obligatoires donnent lieu à des sous-fonctions de saisie incluses et les données facultatives à des sous-fonctions de saisie qui étendent optionnellement. Il est important de noter que la sous-fonction de consultation de la liste des gares est une extension commune à la saisie des gares de départ et d'arrivée. Ceci montre que

les relations d'inclusion et d'extension servent à détailler et à factoriser des sous-fonctions communes. La représentation en UML est illustrée ci-après :



- La représentation est complétée par la possibilité d'achat d'un billet. La possibilité de sélection d'un horaire a également été ajoutée pour rendre possible l'achat du billet. Les trois sous-fonctions de l'achat du billet nécessitent une interaction avec le client, y compris l'envoi du billet qui nécessite de connaître l'adresse du client. Ceci justifie la présence de ces trois sous-fonctions dans le diagramme. La représentation UML est illustrée ci-après :



3. La représentation textuelle de la visualisation reprend les éléments du diagramme précédent. De surcroît, elle est plus précis car elle rend explicite l'ordre d'enclenchement des sous-fonctions ainsi que les conditions d'activation d'une extension.

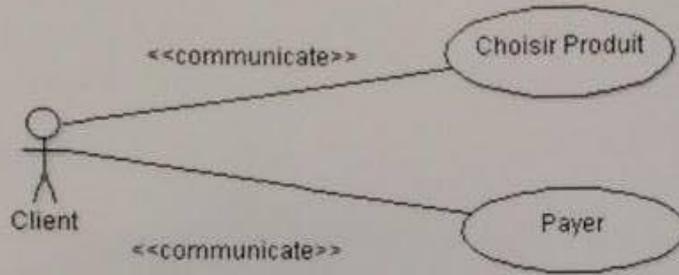
La représentation textuelle se trouve ci-après :

Cas d'utilisation	Visualiser Horaires
Opérations	
1	Afficher la liste des horaires
2	Boucler en 1
3	Afficher la liste des horaires avec la sélection
4	Boucler en 3
Extension	
1.A	Sélectionner un horaire
1.A.1	Prendre en compte cette sélection
1.A.2	Aller en 3
3.A	Acheter le billet correspondant à la sélection
3.A.1	Choisir le tarif
3.A.2	Effectuer le paiement
3.A.3	Envoyer le billet
3.A.4	Aller en 3

Il convient de noter que l'action de quitter la visualisation d'horaire n'a pas été représentée. En effet, c'est le bon sens d'avoir une telle action et elle n'a donc pas besoin d'être représentée.

**Exercice N°3 :**

Le client d'un distributeur de produits courants peut y trouver des produits alimentaires (pains, conserves, boissons, etc.) ainsi que d'autres types de produits courants (lessives, savons, etc.). Une fois qu'il a choisi les produits qu'il désire acheter, il doit ensuite payer ses achats. La figure suivante illustre le cas d'utilisation correspondant à cette description.

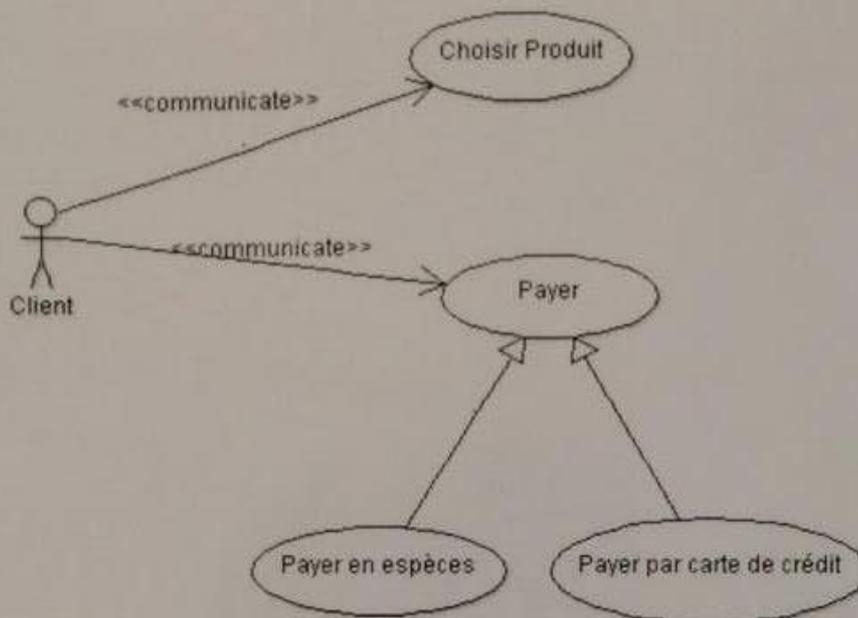


- 1) Il existe deux façons de payer les produits : soit en espèces soit par carte de crédit. Représentez, en complétant le diagramme précédent, ces deux possibilités.
- 2) Lors de l'achat d'un produit alimentaire et uniquement dans ce cas, le client vérifie la date de limite de consommation du produit. Représentez cette vérification, à partir du diagramme précédent.

**Corrigé type**

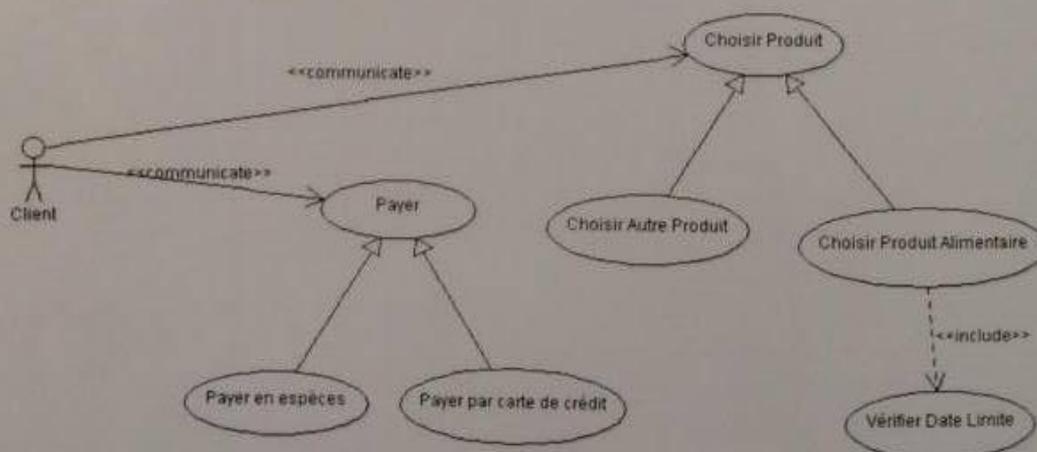
1. Il convient d'introduire les deux modes de paiement : espèce ou carte de crédit. Ceux-ci correspondent à deux cas d'utilisation spécialisant chacun le cas d'utilisation **Payer**. Ce dernier va alors devenir abstrait car il n'est plus mis en œuvre directement : sa mise en œuvre passe alors par l'un des deux sous-cas qui le spécialisent.

La représentation en UML est illustrée à la figure suivante où le cas d'utilisation **Payer** est représenté en italique pour montrer qu'il est abstrait :



2. Le choix du produit dépend maintenant de sa nature. Comme dans le point 1, deux sous-cas d'utilisation sont distingués : Choisir Produit Alimentaire et Choisir Autre Produit. Le cas des produits alimentaires inclut un nouveau cas appelé Vérifier Date Limite.

La représentation en UML est donnée par la figure suivante :



Cet exemple montre bien qu'un sous-cas d'utilisation ayant spécialisé un autre cas peut **enrichir son comportement**. Ici, la vérification de la date limite est un enrichissement du comportement du choix du produit. Cet enrichissement n'est appliqué qu'au choix des produits alimentaires. Dans cet exercice, à deux reprises, la spécialisation de cas d'utilisation existants est utilisée.

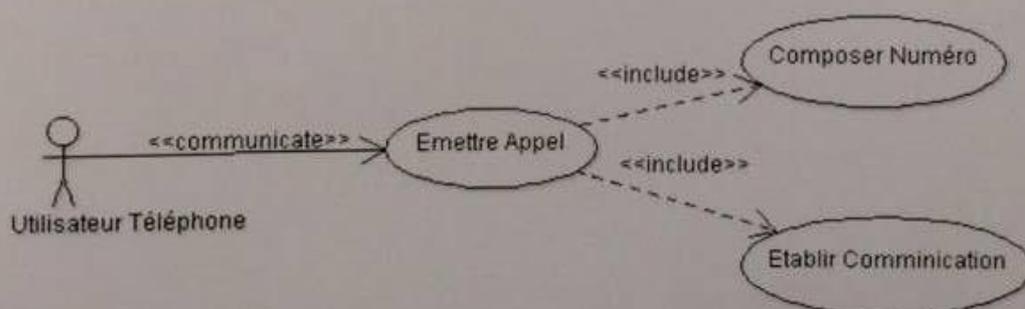
#### Exercice N°4 :

Le but de cet exercice est de décrire et de détailler le cas d'utilisation d'émission d'un appel téléphonique. Il convient de préciser qu'une phase de **numérotation** est nécessaire avant l'établissement de la communication avec le correspondant. Cette phase d'**établissement de la communication a toujours lieu**.

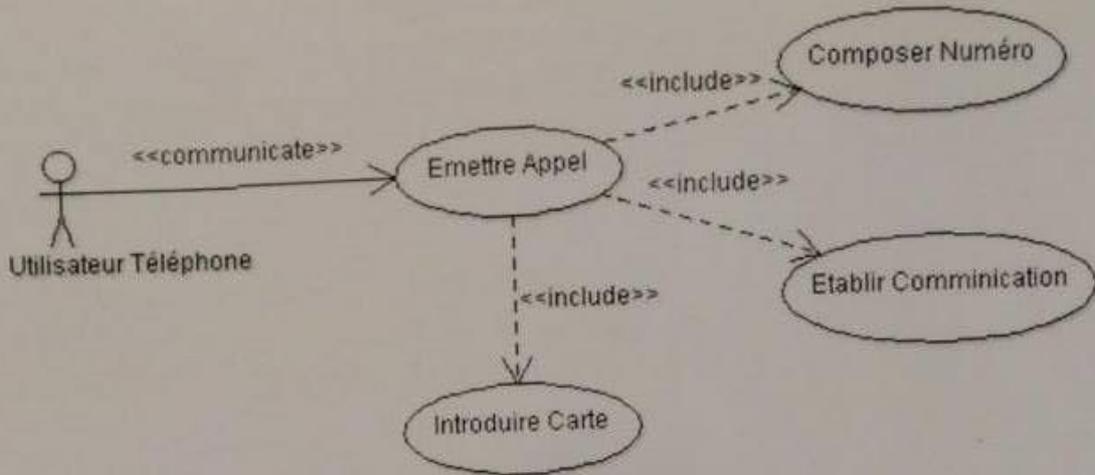
1. Représentez en UML le cas d'utilisation correspondant à l'émission d'un appel.
2. Le cas d'utilisation précédent s'applique bien à une ligne téléphone fixe. Adaptez-le pour une cabine téléphonique, où il est nécessaire d'introduire une carte téléphonique avant de pouvoir émettre un appel.
3. Modifiez également le cas d'utilisation pour un GSM. Il peut être alors nécessaire d'allumer au préalable celui-ci ou d'en déverrouiller le clavier.
4. Reprenez ces trois cas et généralisez-les par un cas d'utilisation abstrait.

#### Corrigé type :

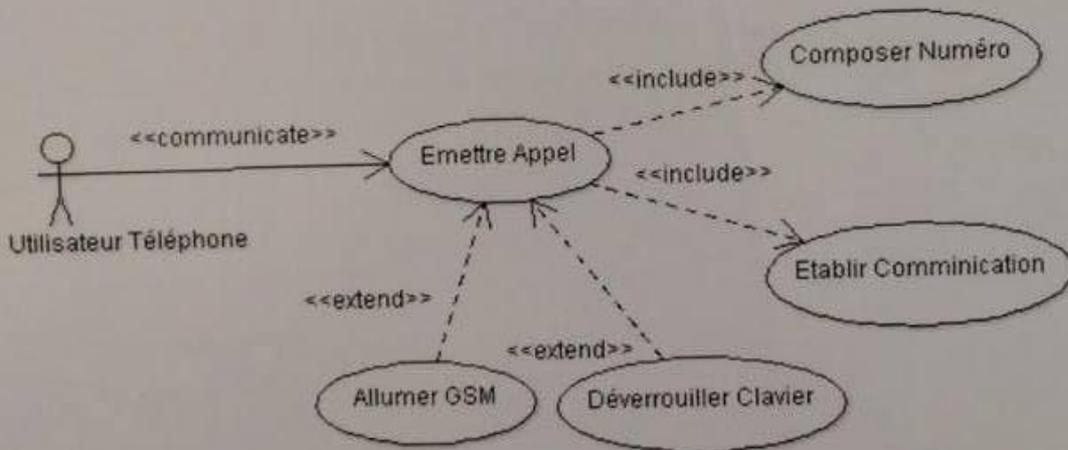
1. Il convient d'introduire le cas d'utilisation d'émission d'un appel ainsi que les sous-fonctions de numérotation et d'établissement de la communication comme illustré ci-après :



2. Dans le cadre d'un appel téléphonique émis depuis une cabine, le cas Emettre un appel inclut le cas d'utilisation d'insertion de la carte :



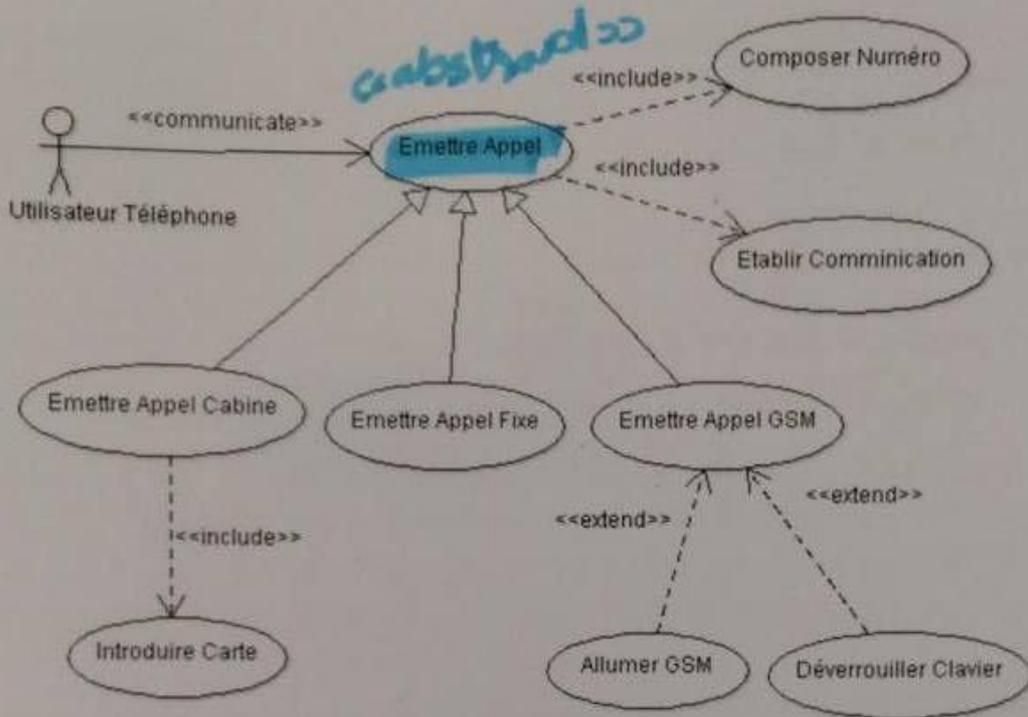
3. Dans le cadre d'un appel téléphonique émis depuis un GSM, deux extensions peuvent être activées si le téléphone doit être allumé ou s'il convient de déverrouiller le clavier :



4. Les trois cas d'utilisation précédents possèdent des similitudes comme le fait qu'ils représentent l'émission d'un appel téléphonique et qu'ils possèdent une relation d'inclusion avec les sous-fonctions de composition d'un numéro et d'établissement de la communication. Un cas d'utilisation abstrait d'émission d'un appel possédant une relation d'inclusion avec ces deux sous-fonctions constitue une généralisation de ces trois cas qui en sont alors des sous-cas.

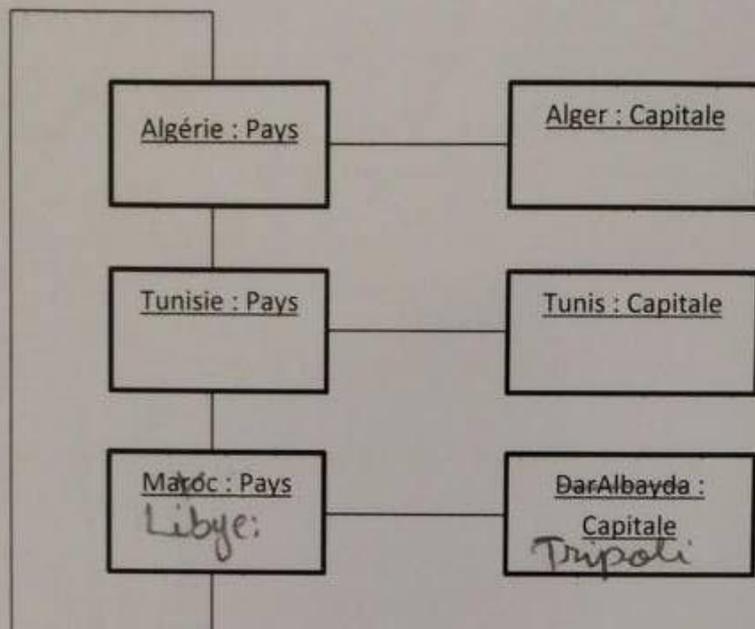
Ce cas d'utilisation est ensuite spécialisé en trois sous-cas : un premier correspondant aux appels depuis une cabine, un deuxième aux appels depuis un téléphone fixe et un troisième aux appels depuis un GSM. Ces sous-cas possèdent, si nécessaire, les relations d'inclusion et d'exclusion qui leur sont spécifiques.

La représentation UML est donnée à la figure suivante :



Exercice N°1 :

1) Tracez un diagramme de classes à partir du diagramme d'objets suivant :

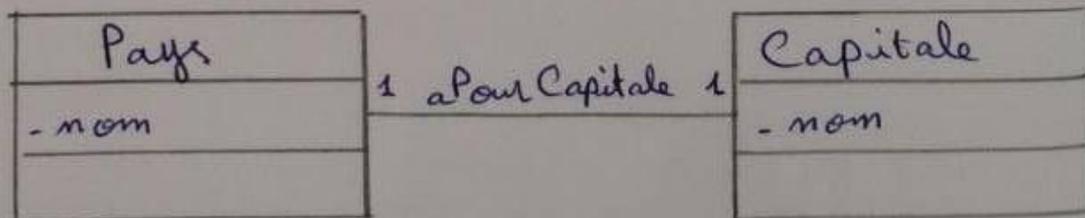


2) Complétez le modèle en proposant des attributs pour les différentes classes.

Corrigé (Diagramme d'objet)

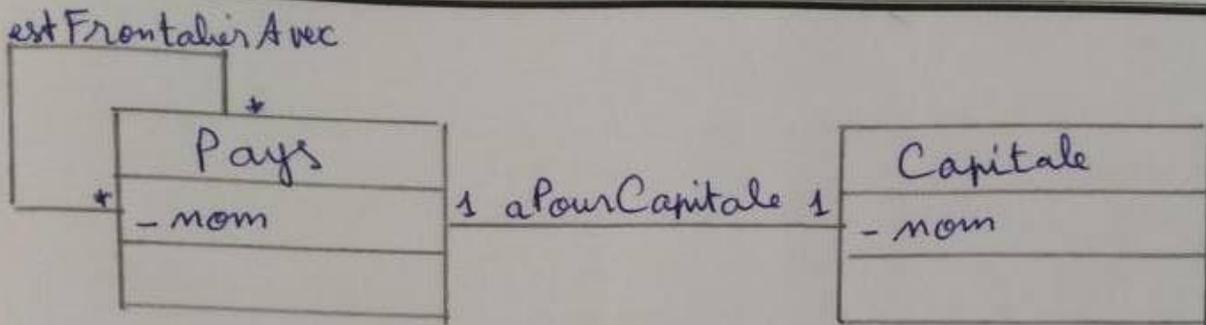
1. Les objets présents dans le diagramme d'objets appartiennent à deux classes **Pays** et **Capitale**. Deux types de liens apparaissent : un lien entre un pays et une capitale et un lien entre pays.

Le premier type de lien est représenté par une association appelée **aPourCapitale**. Un pays possède une seule capitale et une capitale n'appartient qu'à un seul pays, aussi la cardinalité retenue des deux côtés de l'association est 1.

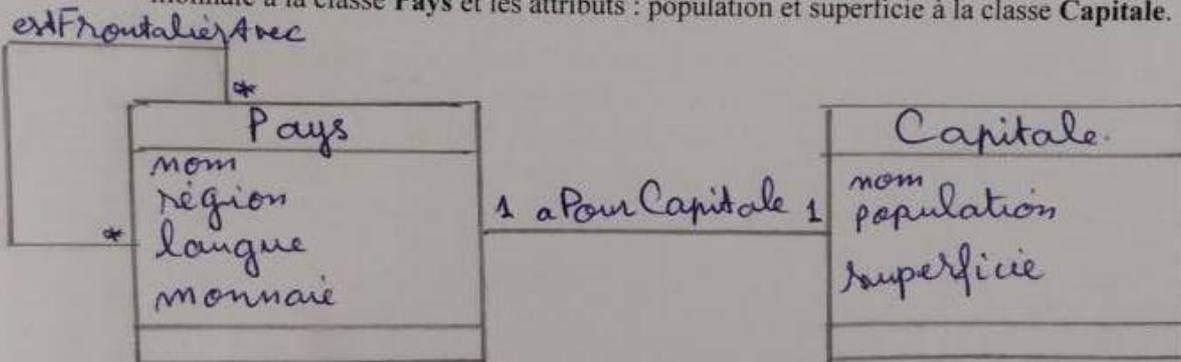


Le deuxième type de lien relie la classe Pays à elle-même, il est représenté par une association réflexive nommée **estFrontièreAvec**. Un pays peut être frontalier avec plusieurs pays ou ne partager aucune frontière, la cardinalité est donc \* des deux côtés de l'association.

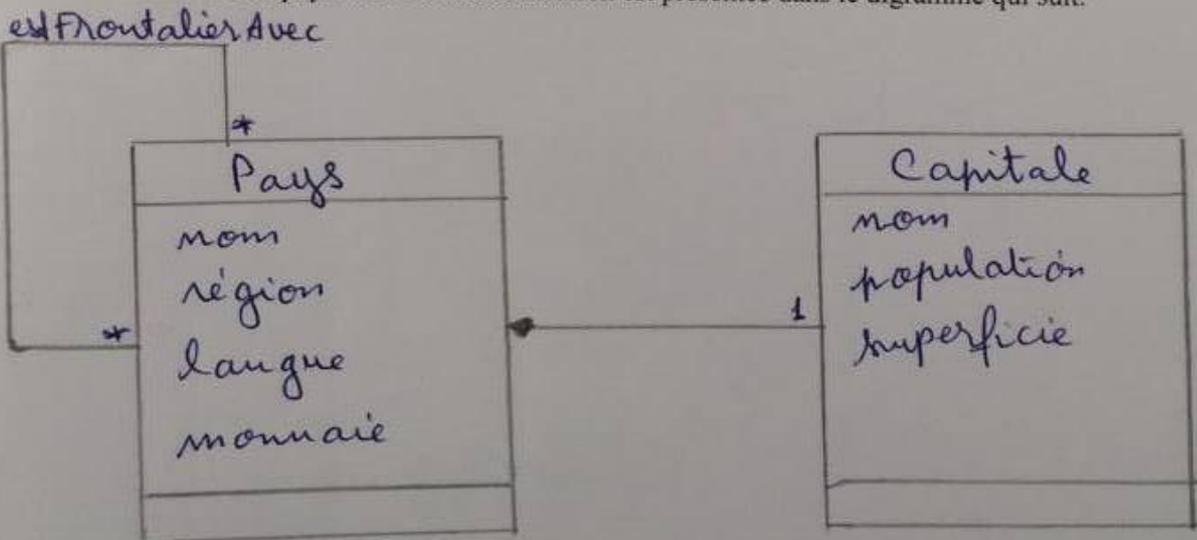
Le diagramme de classe obtenu est le suivant :



2. Le diagramme peut être complété en ajoutant des attributs tels que : région, langue, monnaie à la classe **Pays** et les attributs : population et superficie à la classe **Capitale**.



Notons que la relation entre un pays et sa capitale peut être représentée par une composition. En effet, une capitale fait partie d'un pays et son existence est inhérente à celle de son pays. Cette seconde solution est présentée dans le diagramme qui suit.



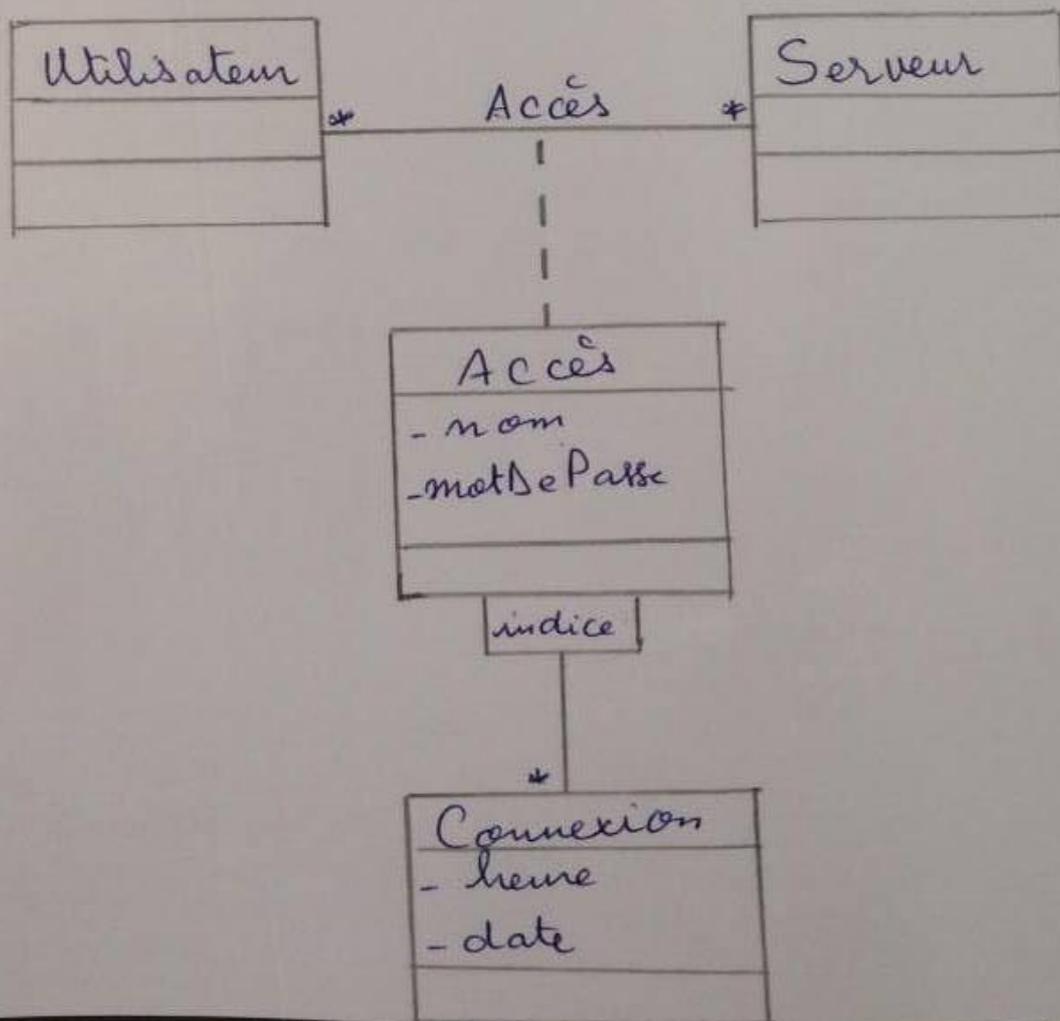
**Exercice N°2 :**

Le but est de décrire la gestion d'une connexion utilisateur à un serveur de bases de données. Un utilisateur de serveurs de bases de données se connecte avec un accès constitué d'un nom de compte et un mot de passe. La date et l'heure de la connexion doivent être stockées.

- 1) Proposez une représentation sous forme d'un diagramme de classes.
- 2) Complétez le modèle afin que chaque accès puisse disposer d'un espace de travail de taille fixe. Un espace de travail peut être partagé par plusieurs connexions.

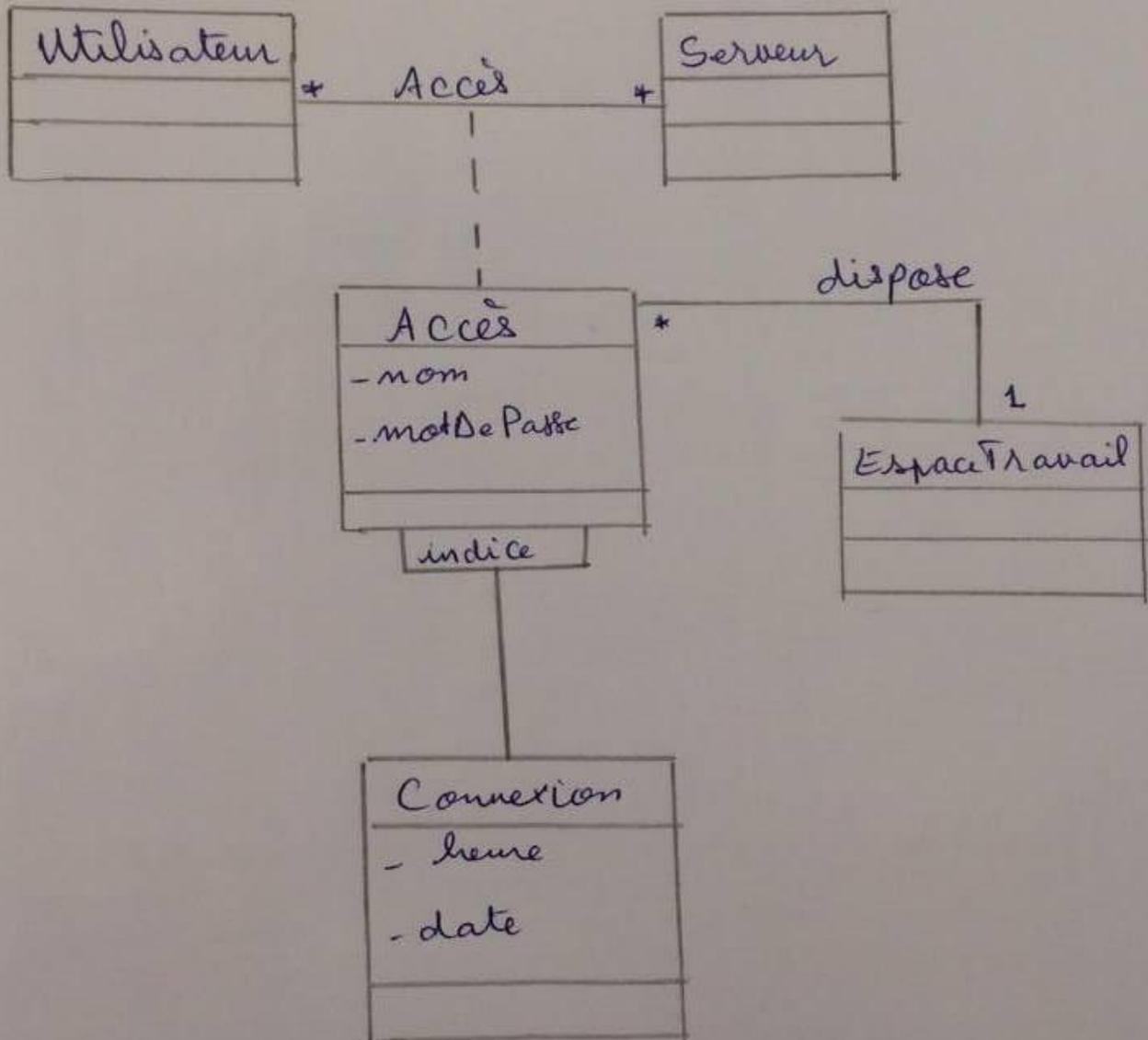
**Corrigé (Classe association) :**

1. Le lien entre un utilisateur et un serveur de base de données est décrit par une association. Un utilisateur peut se connecter à plusieurs serveurs et un serveur peut accepter plusieurs utilisateurs. Les informations date et heure concernent une connexion d'un utilisateur à un serveur, les informations nom et motDePasse sont relatives à un accès au serveur. Elles donnent lieu à des attributs qui vont être regroupés dans une classe d'association. Le diagramme de classe est le suivant :



2. Un accès dispose d'un espace de travail.

La question qui se pose est : sous quelle forme faut-il représenter l'espace de travail ? si la seule information à stocker est le nom de l'espace de travail, la solution la plus simple est de le représenter sous forme d'un attribut **nomEspace** de la classe **Accès**. Seulement, d'autres informations peuvent être intéressantes stocker, comme par exemple sa taille. Aussi, il est préférable de le représenter sous forme d'une classe, liée à la classe d'association **Accès** par une association avec les cardinalités correspondantes.



**Exercice N°3 :**

Il s'agit de modéliser une connexion à un site Web. La description du comportement du système est la suivante :

Un site Web est un ensemble de pages Web liées entre elles par des références appelées hyperliens. Ces pages sont conçues pour être consultées avec un navigateur. Chaque page possède sa propre configuration. La consultation d'une page s'effectue grâce à une URL qui correspond à l'adresse de la page.

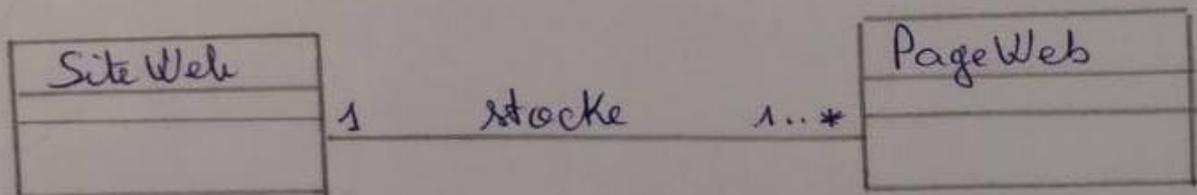
Un utilisateur peut, à tout moment, consulter une page Web en utilisant un navigateur. Cette requête est transmise au site Web correspondant. La date et l'heure de chaque connexion doivent être stockées.

Tracez le diagramme de classes correspondant à la description.

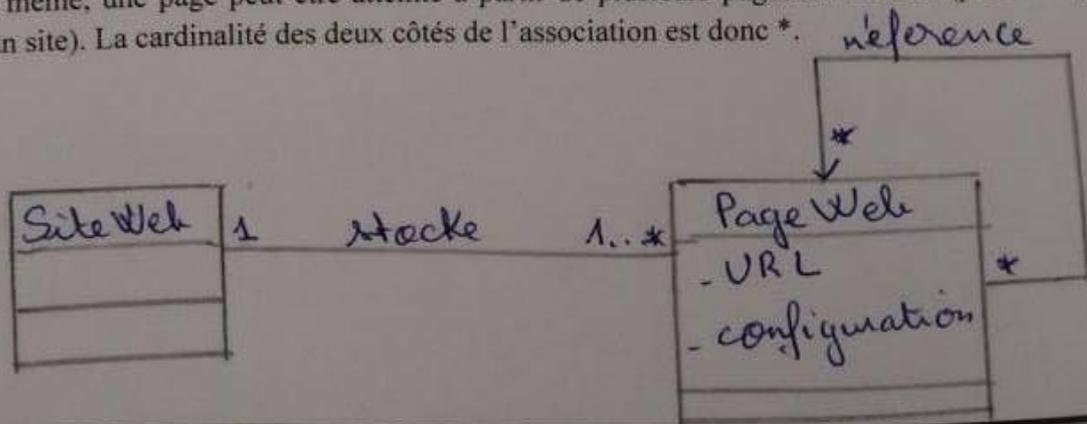
**Corrigé (site web)**

A partir de la phrase « un site web est un ensemble de pages web liées entre elles par des références appelées hyperliens », deux concepts importants ressortent et donnent lieu à des classes candidates : **SiteWeb** et **PageWeb**.

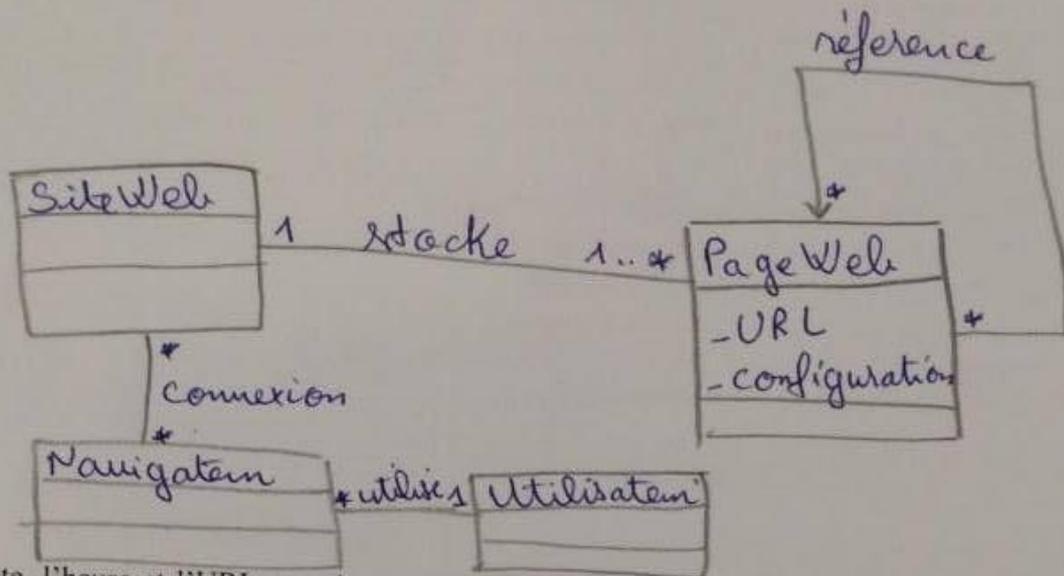
Ces deux classes sont liées par une association **stocke**. Un site peut stocker une à plusieurs pages et une page n'est stockée que dans un seul site. Le diagramme intermédiaire est le suivant :



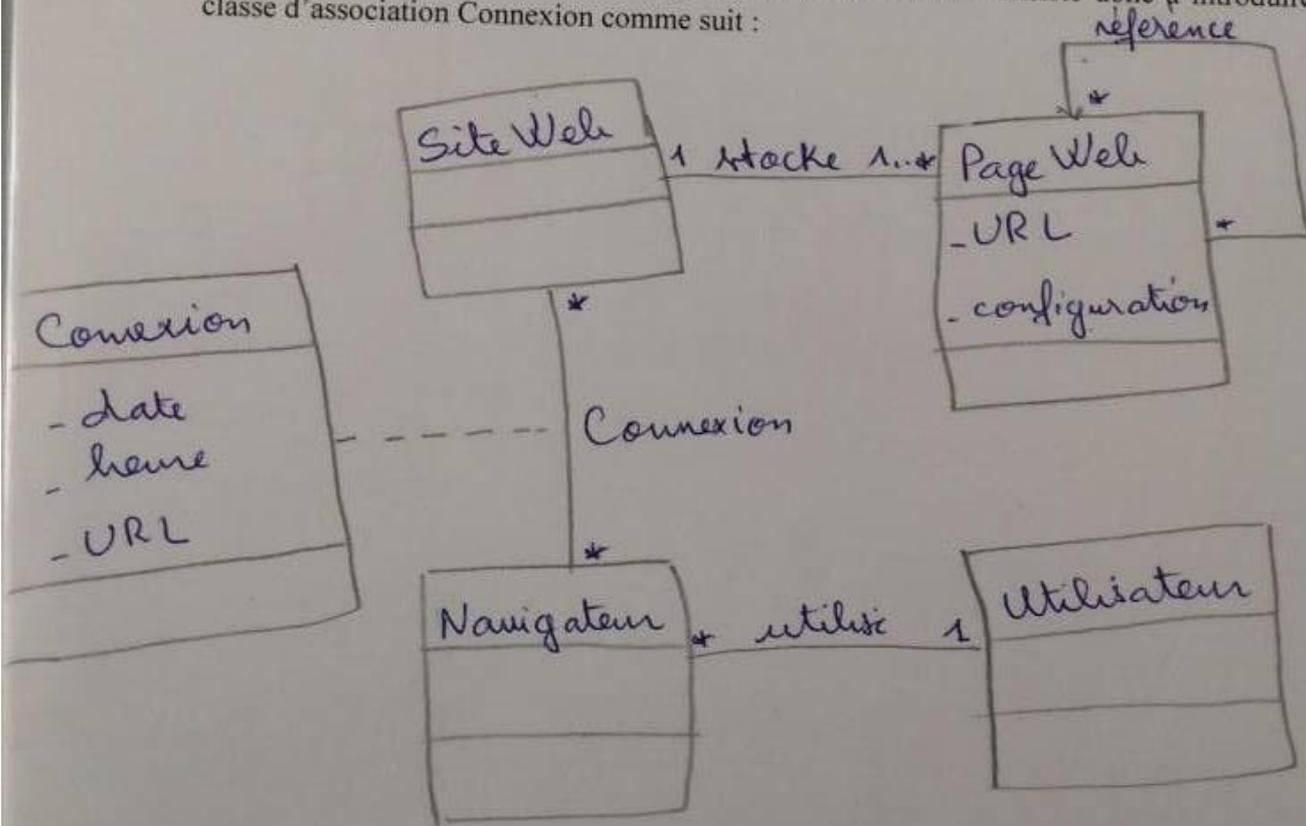
Les pages sont hyperliées entre elles. Ceci donne lieu à une association réflexive entre la classe PageWeb et elle-même. Une page peut être reliée à aucune ou plusieurs autres pages. De même, une page peut être atteinte à partir de plusieurs pages ou aucune (première page d'un site). La cardinalité des deux côtés de l'association est donc \*.



La modélisation de la connexion d'un utilisateur à une page web fait intervenir un navigateur. Les classes Navigateur et Utilisateur viennent s'ajouter au modèle. Un utilisateur utilise un ou plusieurs navigateurs. Un navigateur est connecté à des sites Web. L'URL et la configuration d'une page web donnent lieu à des attributs de la classe PageWeb. Le diagramme complété est le suivant :



La date, l'heure et l'URL soumise concernent une connexion entre un navigateur et un site web. Ce sont des informations sur l'association. La solution consiste donc à introduire une classe d'association Connexion comme suit :



**Exercice N°4 :**

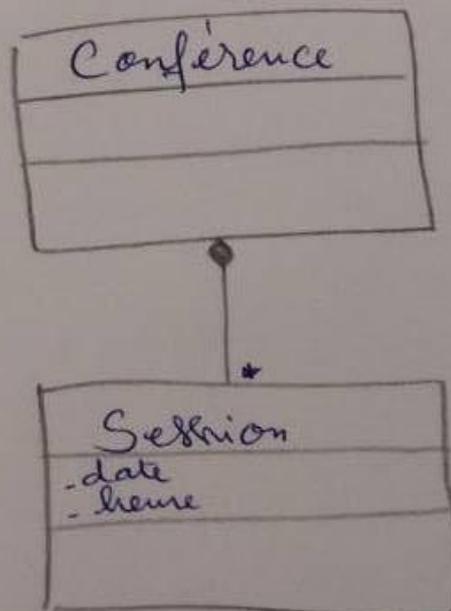
Il s'agit de modéliser un diagramme de classes pour la gestion d'une conférence scientifique. Les connaissances du domaine sont résumées sous forme de phrases comme suit :

- 1) La conférence se compose de plusieurs sessions.
- 2) Chaque session possède une date et une heure de début de session,
- 3) Les participants participent à une session soit en tant qu'orateur soit en tant que public.
- 4) Tous les participants doivent s'inscrire à la conférence. Une inscription peut être annulée ou confirmée.
- 5) Un article scientifique est présenté à une session.
- 6) Un article est soit un article long, soit un article court. Il est composé de sections numérotées et concerne un sujet donné.
- 7) Un auteur peut avoir un ou plusieurs articles présentés à la conférence.

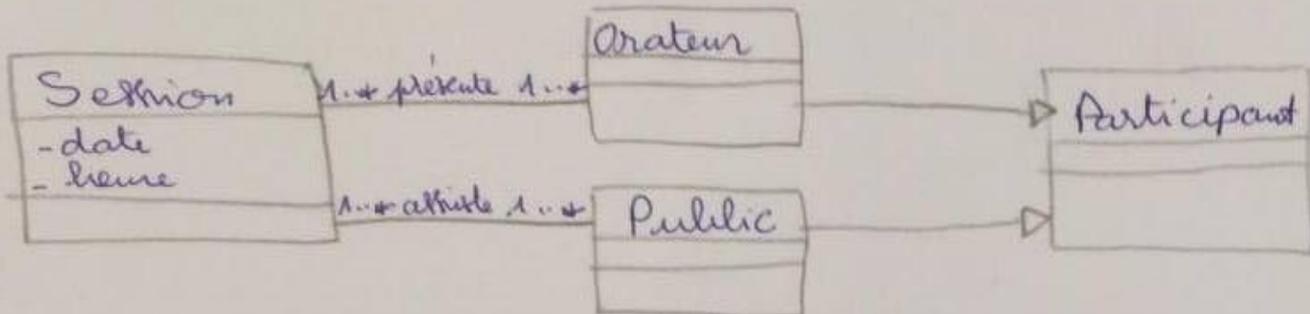
**Corrigé (Gestion d'une conférence scientifique) :**

1. La conférence se compose de plusieurs sessions.  
Cette phrase met en évidence deux concepts importants du domaine : conférence et session, qui sont représentés par des classes. D'après cette phrase, la relation qui lie ces deux classes est au moins une agrégation. Il reste à vérifier s'il s'agit d'une composition. L'énoncé ne précise pas si une session appartient à une seule conférence. En supposant que c'est le cas et étant donné qu'une session n'a pas lieu d'être sans conférence, la relation est une composition.
2. Chaque session possède une date et une heure de début de session. L'heure et la date de la session sont des notions représentant des valeurs concernant chacune des sessions. Elles sont représentées en tant qu'attributs de la classe **Session**.

Le diagramme initial est donc le suivant :



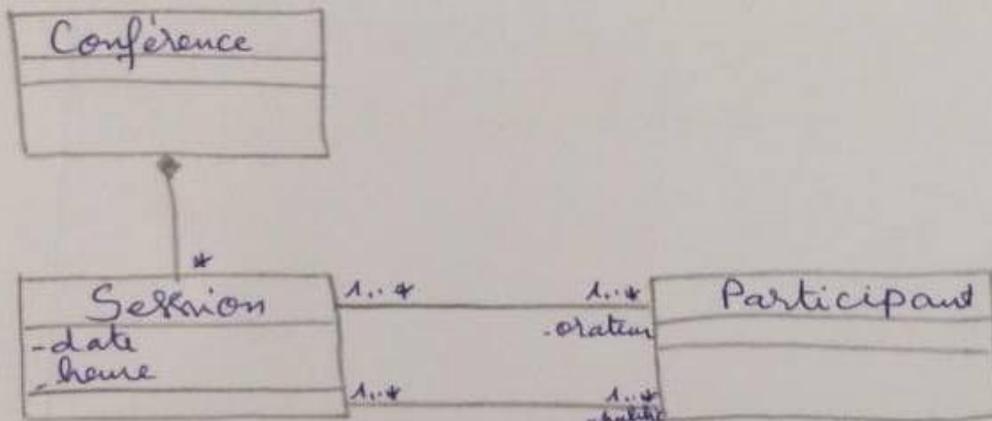
3. Les participants participent à une session soit en tant qu'orateur soit tant que public. Une modélisation pouvant venir à l'esprit consiste à créer deux classes **Orateur** et **Public** qui seraient des sous-classes de participant.



Cette solution est erronée puisque toutes les instances d'**Orateur** sont instances de **Public**, ce qui implique qu'il est inutile de faire de **Orateur** et **Public** deux classes distinctes.

Une solution correcte consiste à utiliser la notion de rôle dans une association. Ainsi, un participant peut jouer le rôle d'orateur ou de public ou les deux à la fois.

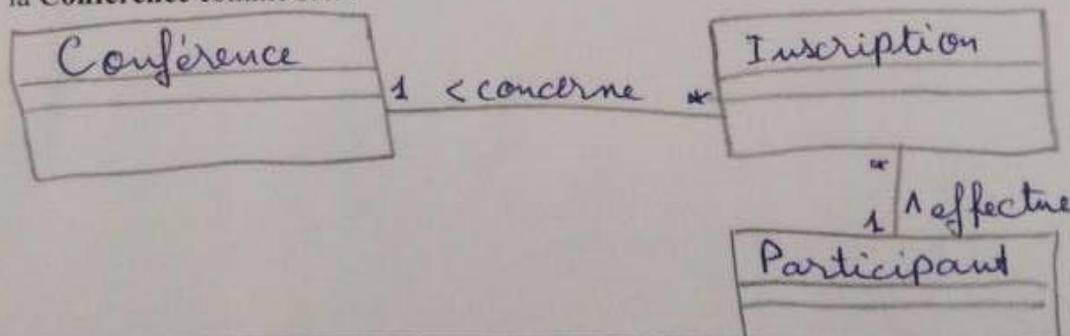
Le diagramme complété est le suivant :



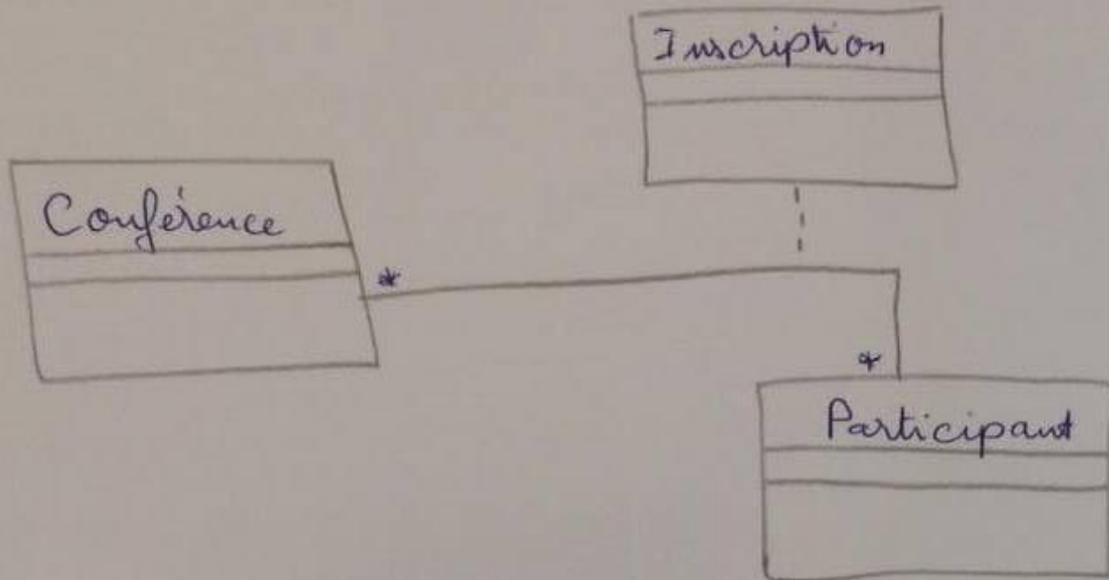
4. Tous les participants doivent s'inscrire à la conférence. Une inscription peut être annulée ou confirmée.

Cette phrase décrit un lien reliant un participant à une conférence à travers deux modélisations possibles.

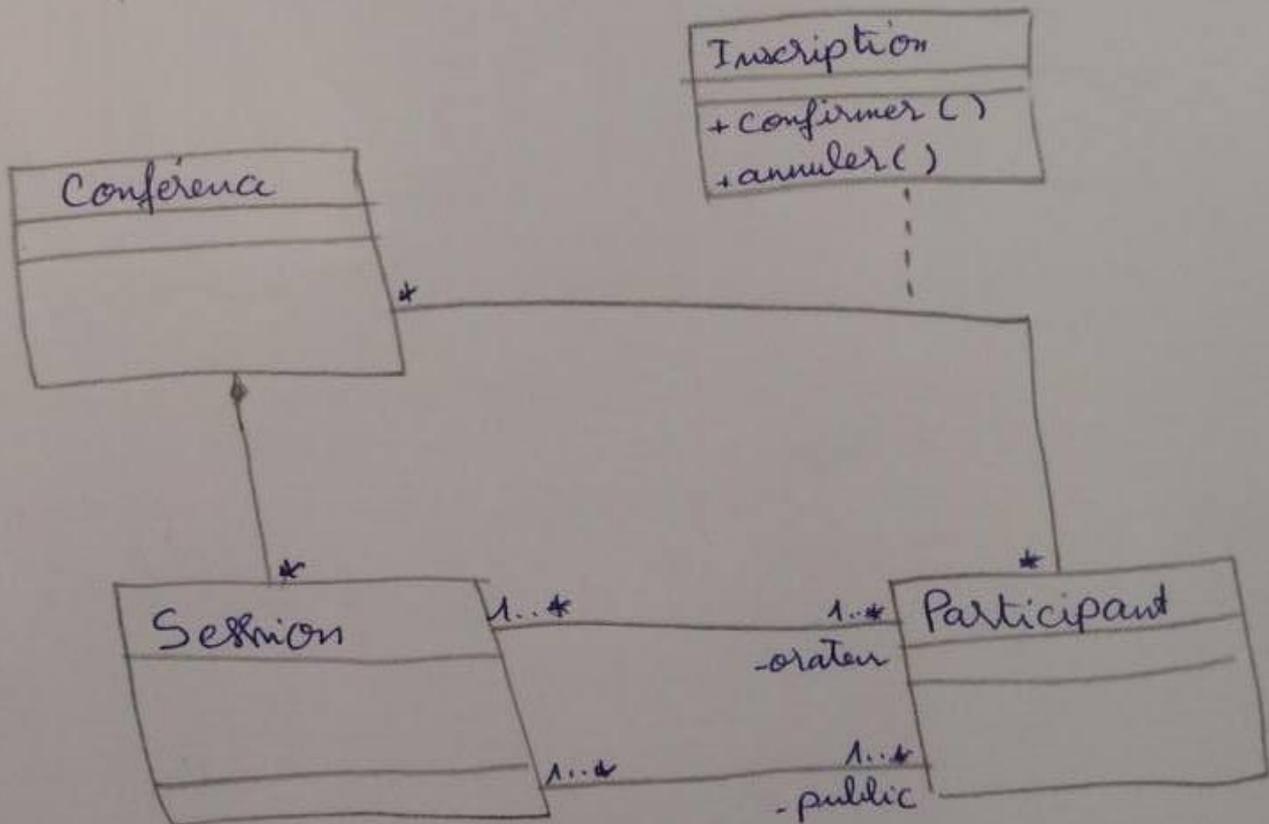
Dans la première, l'inscription est représentée par une classe reliée à **Participant** et à la **Conférence** comme suit :



Une seconde solution consiste à faire de l'inscription une classe d'association entre Participant et Conférence. Le diagramme est le suivant :



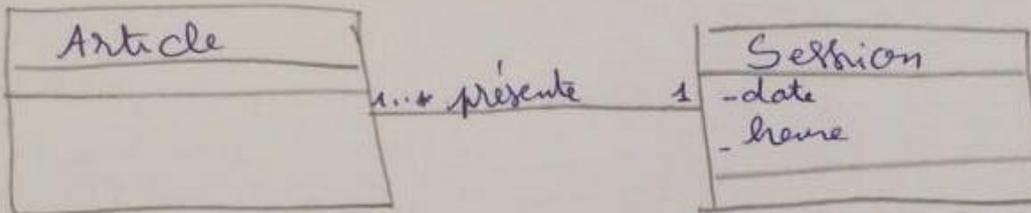
La confirmation ou l'annulation d'une inscription décrivent des opérations pouvant être appliquées à la classe **Inscription**. En modélisation statique, ceci se traduit par deux méthodes **confirmer** et **annuler** de la classe **Inscription**.



Les phrases 5,6, et 7 concernent les articles présentés à la conférence.

5. Un article scientifique est présenté à une session.

Ceci se traduit par une association **présenté**. Lors d'une session, un ou plusieurs articles peuvent être présentés, la multiplicité du côté de la classe Article est donc 1..\*.

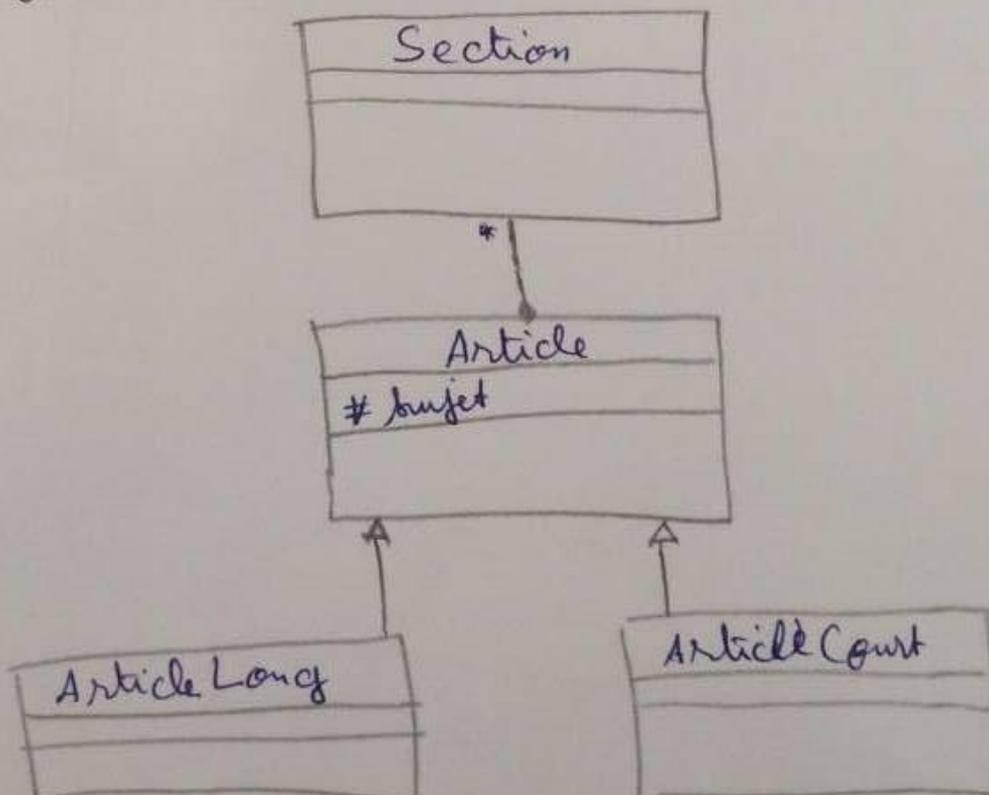


6. Un article soit un article long soit un article court. Il est composé de sections numérotées et concerne un sujet donné.

La modélisation de cette phrase s'effectue comme suit :

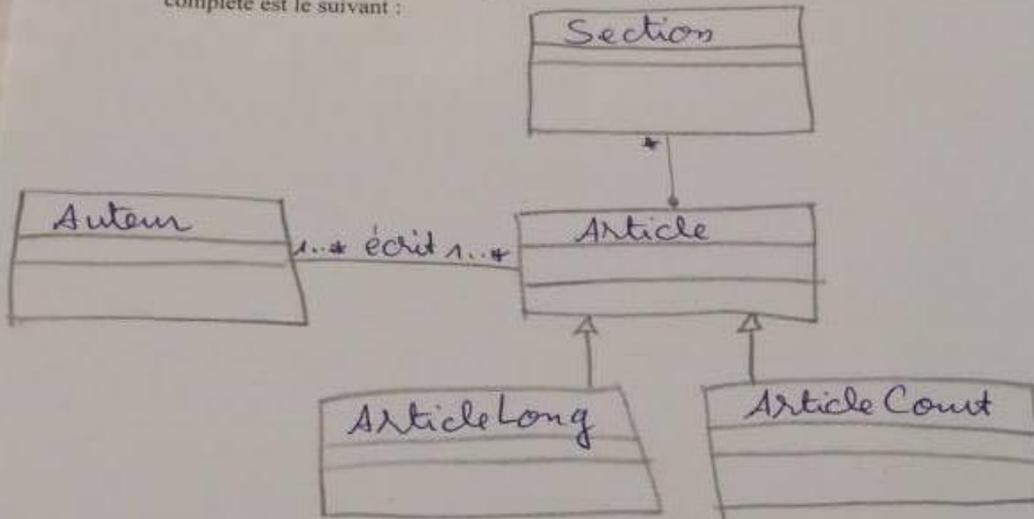
- **ArticleLong** et **ArticleCourt** sont des spécialisations de la classe Article qui devient abstraite.
- Un article est composé de plusieurs sections, il s'agit d'une relation de composition puisqu'une section n'appartient qu'à un seul article et son existence dépend de celle de l'article.
- Enfin, le sujet est une caractéristique d'article qui prend une valeur. Elle est modélisée comme attribut de la classe **Article**.

Le diagramme obtenu est le suivant :

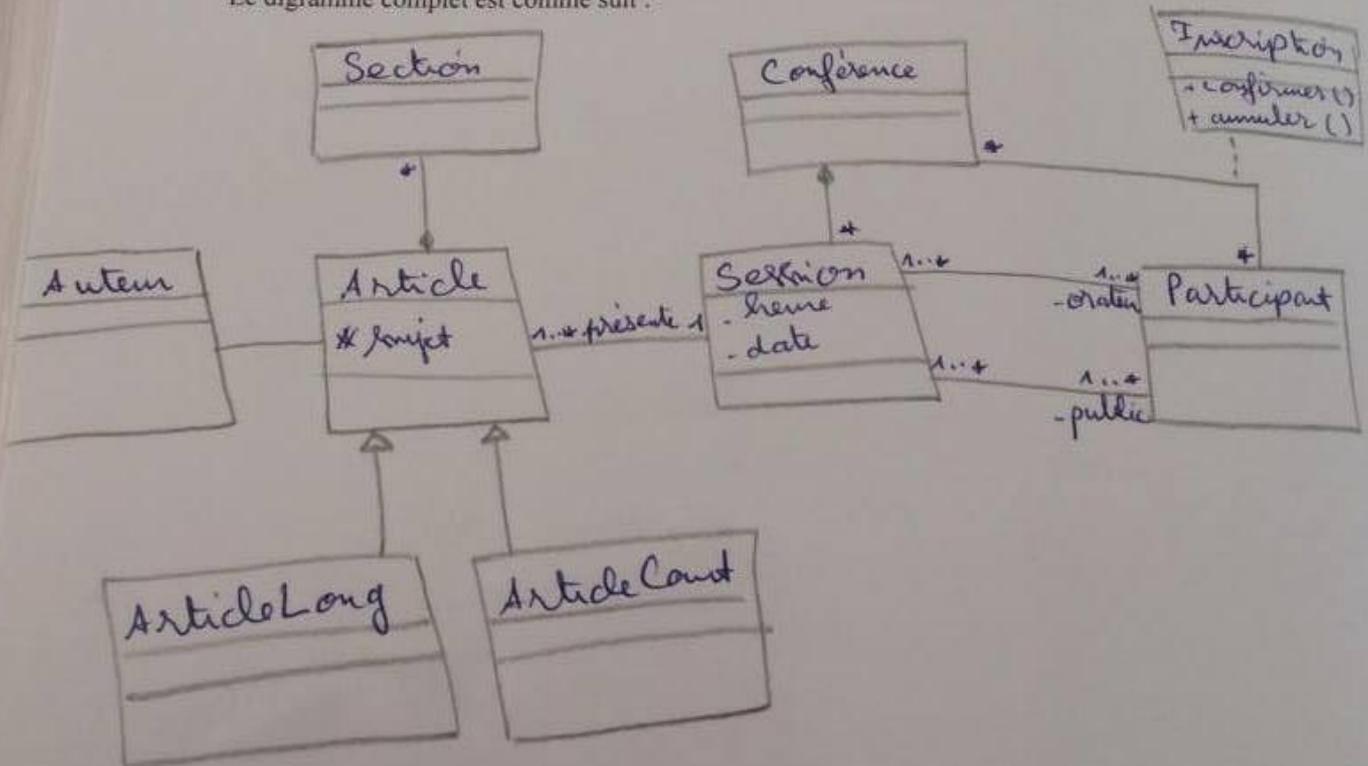


7. Un auteur peut avoir un ou plusieurs articles présentés à la conférence.

Enfin, cette phrase décrit le lien entre les classes Auteur et Article. Un auteur peut écrire plusieurs articles et un article peut être écrit par plusieurs auteurs. Le diagramme ainsi complété est le suivant :



Le diagramme complet est comme suit :



Plusieurs classes n'ont aucun attribut. Les attributs suivants peuvent être ajoutés au modèle :

**Classe Auteur**

- nom - prénom - numTéléphone - corrierElectronique - organisme

**Classe Article**

- titre - nombrePages

**Classe Section**

- numéroSection - titre

**Classe Participant**

- nom - prénom - corrierElectronique - organisme

**Classe Conférence**

- nom - dateDébut - dateFin - lieu

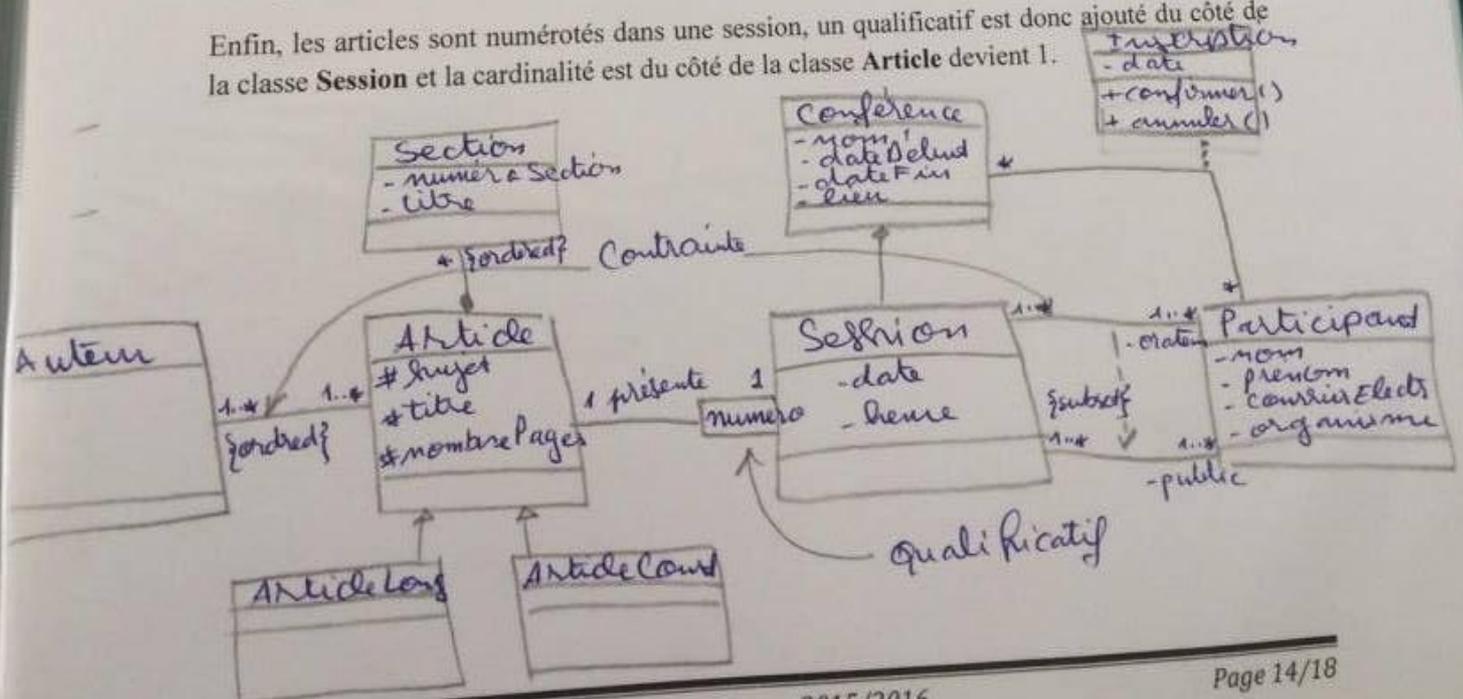
**Classe Inscription**

- date

Des contraintes et des qualificatifs peuvent être ajoutés au modèle. La première concerne les deux associations entre la classe Session et la classe Participant. Un orateur participe aussi à la session en tant que public, donc les occurrences des orateurs sont incluses dans celles de public. Ceci est représenté par la contrainte {subset} entre les deux associations.

Des contraintes d'ordre, notées {ordered} en UML, sont ajoutées pour les auteurs d'un article ainsi que ses sections.

Enfin, les articles sont numérotés dans une session, un qualificatif est donc ajouté du côté de la classe Session et la cardinalité est du côté de la classe Article devient 1.



**Exercice N°5 :**

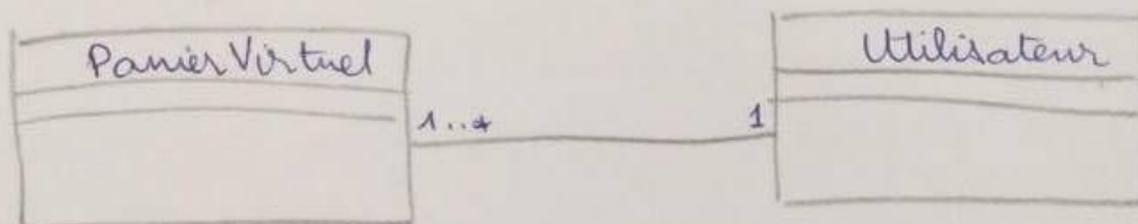
Voici la description d'un système de commerce électronique :

- Un utilisateur (client) qui se connecte à un site de commerce électronique choisit parmi les produits proposés.
- Il remplit au fur et à mesure un panier virtuel comportant les produits à acheter.
- A tout moment, il peut ajouter ou supprimer un produit au panier.
- Le paiement s'effectue à l'aide d'une carte de crédit. Celle-ci n'appartient pas nécessairement à l'utilisateur.
- A la fin de la transaction, l'utilisateur peut soit valider sa commande ou l'annuler.

- 1) Décrivez ce système à l'aide d'un diagramme de classes.
- 2) Complétez le diagramme en ajoutant le concept de client privilégié. Ce type de clients bénéficie d'une réduction pour ces achats et les informations de ses cartes de crédit sont sauvegardées par le système.

**Corrigé (commerce électronique)**

1. L'utilisateur manipule le panier en y ajoutant ou en enlevant des produits disponibles sur le site. Le lien entre les classes **Utilisateur** et **Panier Virtuel** est représenté par une association. Un utilisateur peut avoir plusieurs paniers virtuels.

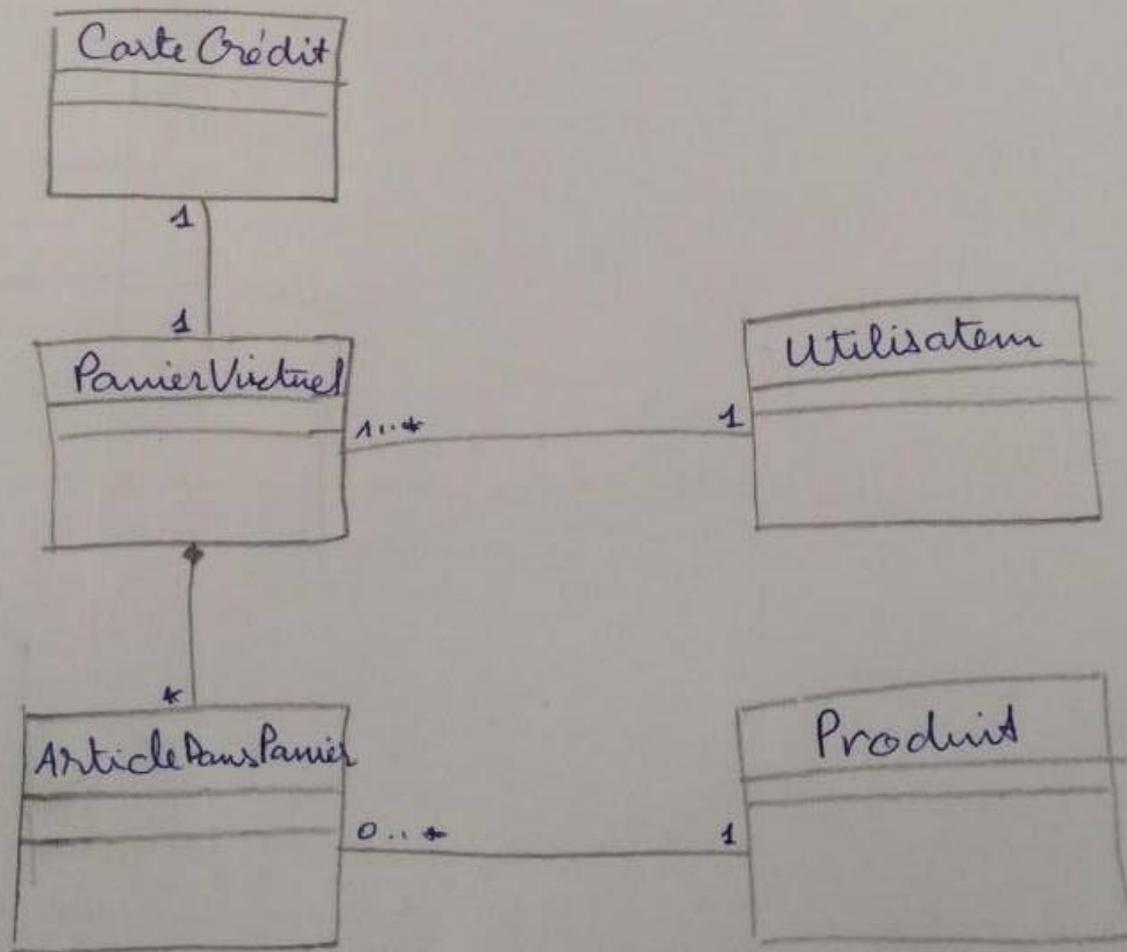


Le panier virtuel contient des produits. La solution la plus simple consiste à représenter ce lien par une relation de composition entre le panier et les produits qu'il contient. Cette relation ne peut pas relier le panier directement à la classe **Produit** car il faut distinguer les produits disponibles et pour lesquels il faut stocker des informations générales telles que leur désignation et les produits à mettre dans le panier avec des informations telles que la qualité et le prix. Les deux classes correspondantes sont **Produits** et **ArticleDansPanier**.

Pour chaque instance de la classe **Produit**, il peut y avoir plusieurs instances de **ArticleDansPanier**. Le diagramme est complété en conséquence comme suit :



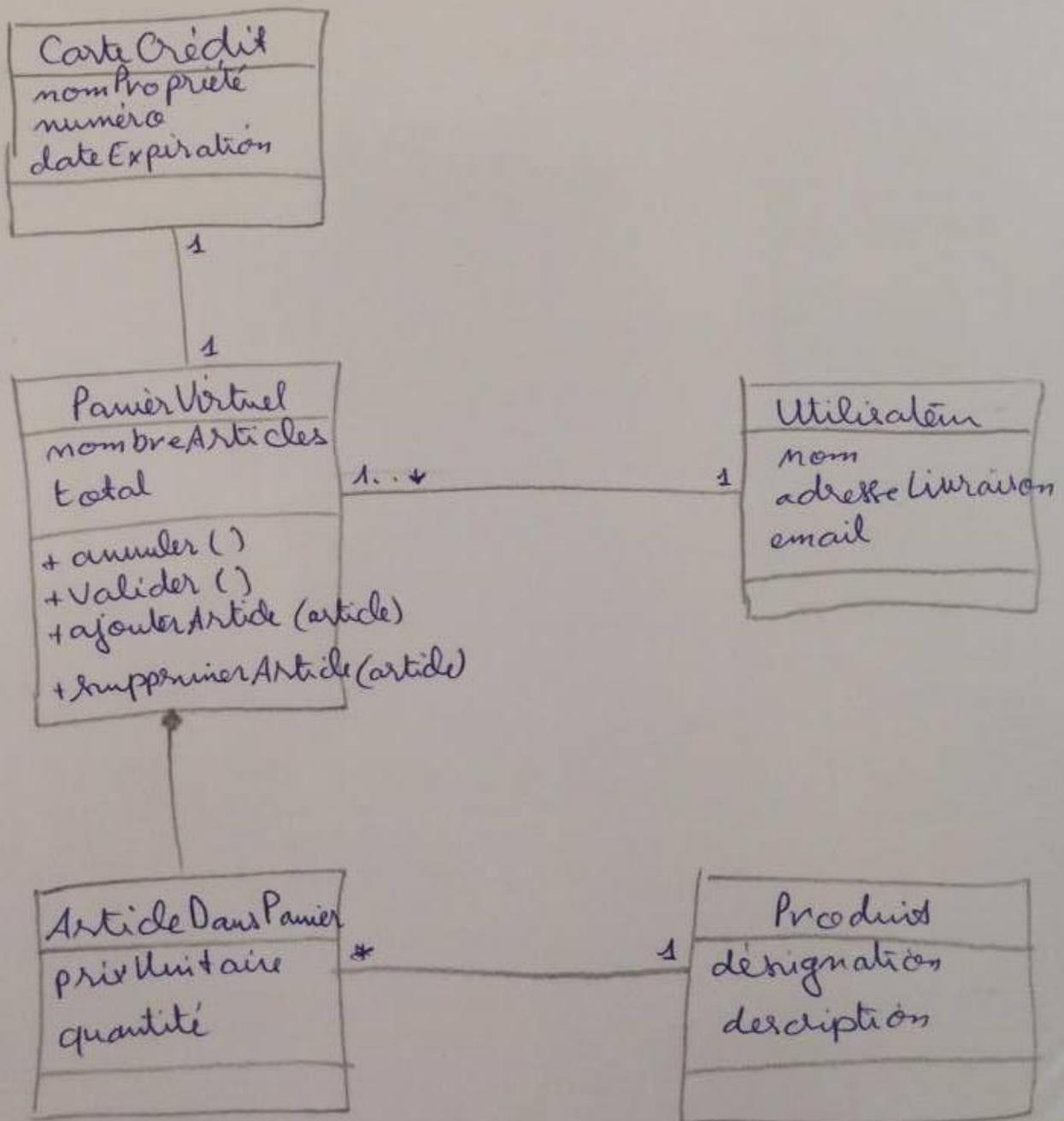
La carte de crédit n'appartient pas nécessairement à l'utilisateur. Elle concerne un achat, c'est-à-dire un panier. La classe correspondante **CarteCrédit** est liée par conséquent à la classe **PanierVirtuel**.



Des attributs ainsi que des méthodes peuvent être ajoutés au modèle pour les compléter.

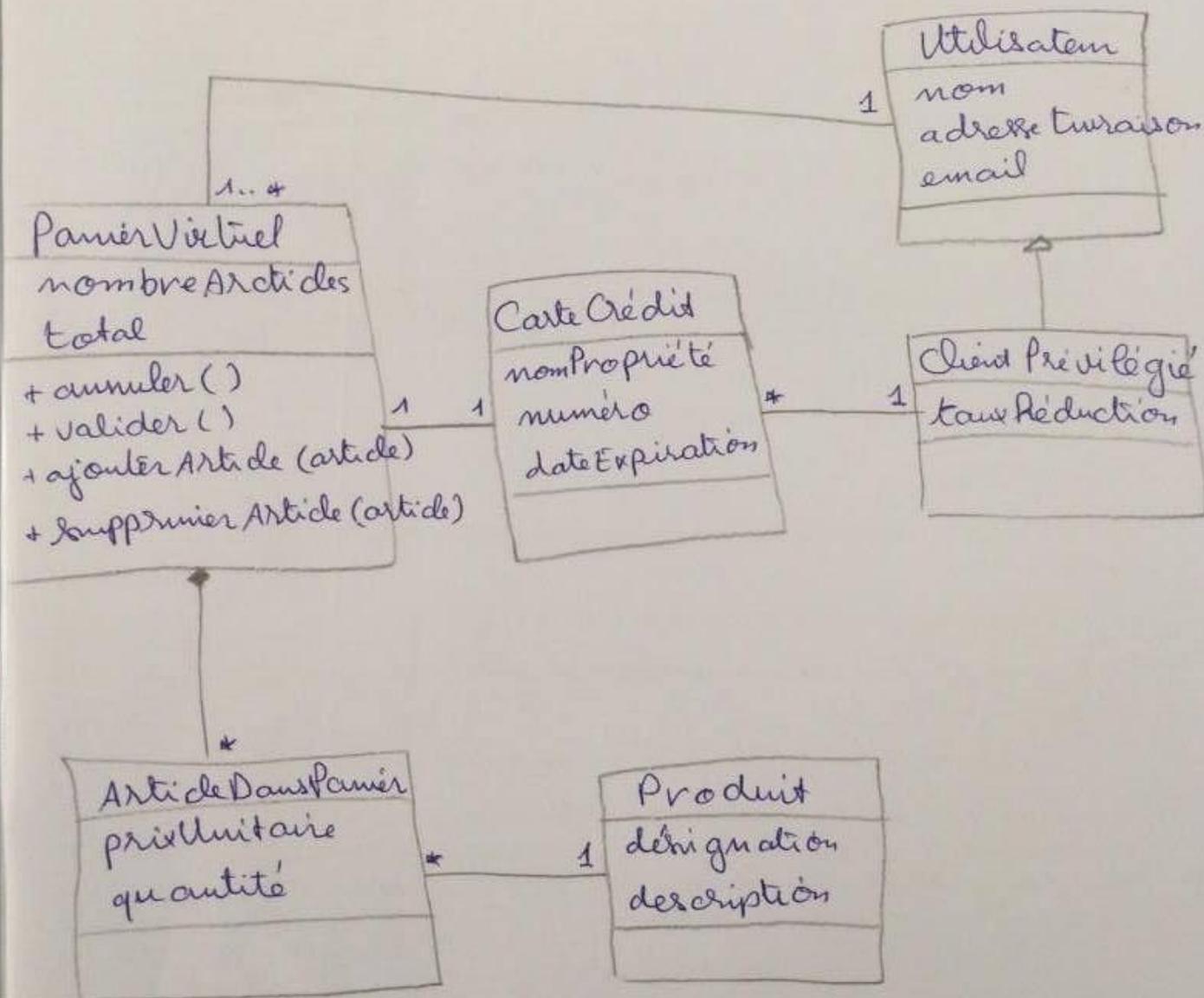
D'après l'énoncé, un client peut soit valider ou annuler une commande. Les méthodes correspondantes doivent être ajoutées à la classe PanierVirtuel. L'ajout et la suppression de produits du panier représentés par des méthodes ajouterArticle et supprimerArticle dans la classe PanierVirtuel.

Le diagramme complété est le suivant :



2. Le concept de client privilégié sera représenté par une classe. Un client privilégié est un type particulier d'utilisateur du site sur lequel d'autres informations sont à sauvegarder, dont les informations concernant ses cartes de crédits. Puisqu'un client peut posséder plusieurs ces informations ne peuvent pas être représentées par des attributs. De plus, la classe **CarteCrédit** est déjà présente dans le modèle. Il suffit donc de la relier à la classe **ClientPrivilégié**.

Les clients privilégiés bénéficient d'une réduction. Il est intéressant de garder le taux de réduction en tant qu'attribut de la classe si celui-ci diffère pour chaque client.



**Exercice N°1 :**

Le but de l'exercice est de décrire les interactions entre un utilisateur et le site Web d'une société de chemins de fer.

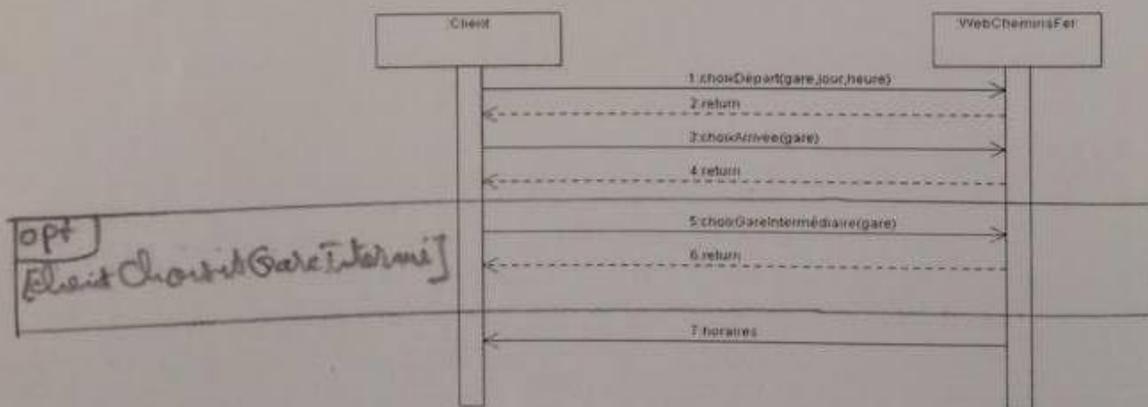
- 1) Représentez à l'aide d'un diagramme de séquence, les interactions suivantes :
  - a. Le choix de la gare, de la date et de l'heure du départ,
  - b. Le choix de la gare d'arrivée,
  - c. Choix optionnel d'une gare intermédiaire,
  - d. Obtention des horaires.
- 2) Ajoutez la possibilité d'acheter un billet correspondant à l'horaire préalablement sélectionné parmi ceux visualisés. L'utilisateur confirme ensuite l'acquisition définitive avant de recevoir le billet pour impression. Aucune trace informatique du billet n'est conservée.

**Corrigé :**

1. Le diagramme de séquence illustrant la consultation des horaires est tracé ci-après. Des paramètres sont utilisés pour transmettre les informations nécessaires comme la gare, la date et l'heure de départ associées au message **choixDépart**. Les trois messages **choixDépart**, **choixArrivée** et **choixGareIntermédiaire** sont de type synchrone. Ils sont suivis, par conséquent, d'un message de retour provenant du destinataire.

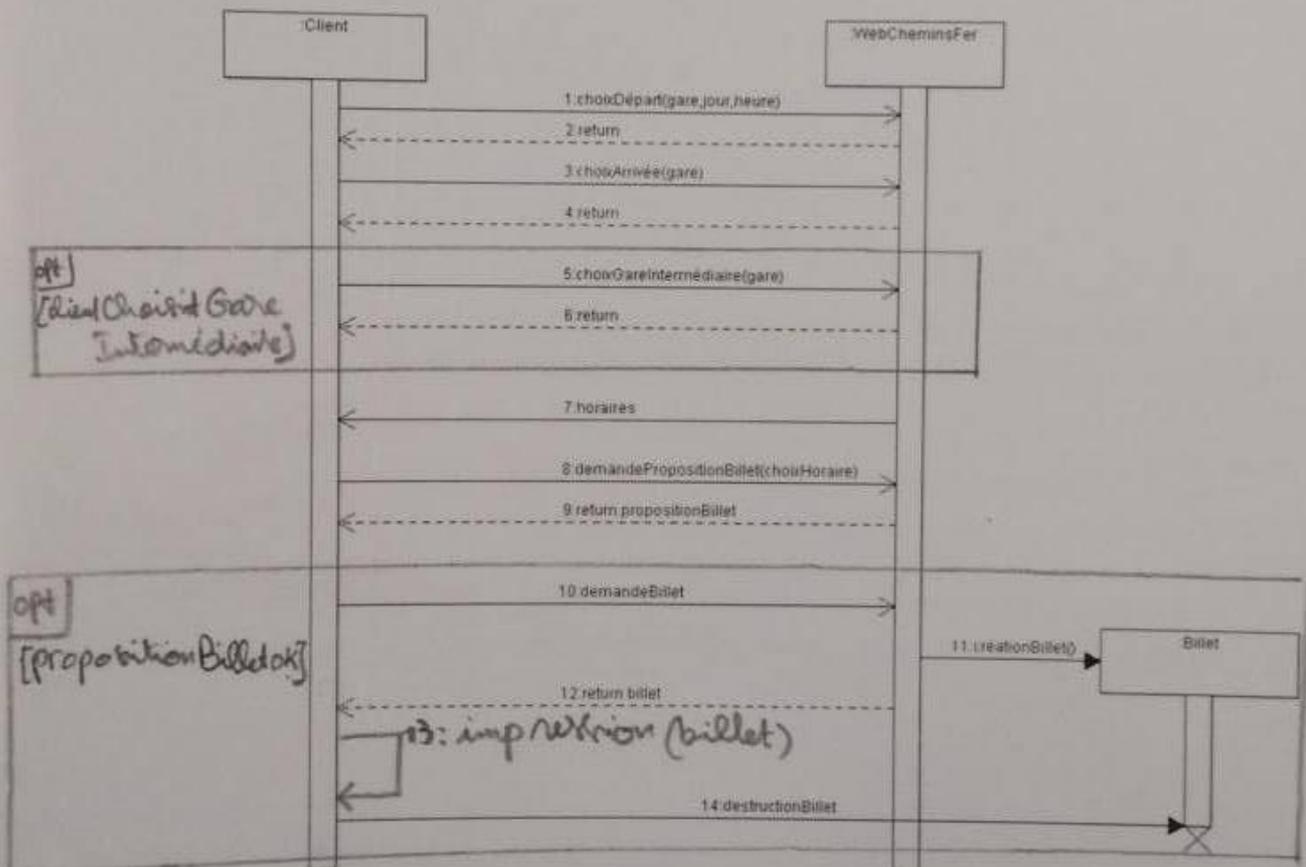
Le choix d'une gare intermédiaire est facultatif. Il n'a lieu que si l'utilisateur le désire, ce qui est illustré par la condition de garde **clientChoisitIntermédiaire**.

Enfin, le message qui contient l'information des horaires, appelé **horaires**, est renvoyé au client en mode asynchrone.



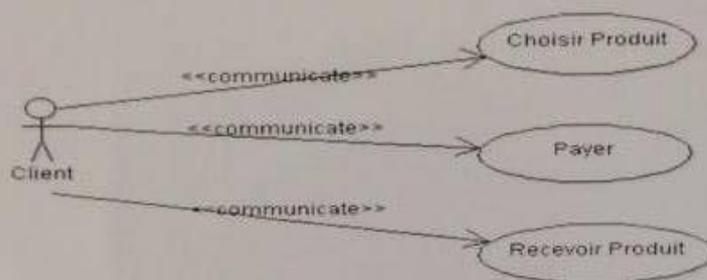
2. Le diagramme de séquences intégrant la possibilité d'acquisition d'un billet correspondant à l'un des horaires renvoyé est illustré ci-après. L'utilisateur choisit un horaire et reçoit une proposition de billet. Si cette proposition lui convient (représentée par la condition de garde **propositionBilletOK**), l'utilisateur demande alors le billet. Ceci provoque la création d'une nouvelle instance de la classe **Billet** par le site Web, instance qui est ensuite retournée à l'utilisateur. L'utilisateur imprime le billet puis le détruit.

L'impression est représentée par un message de l'utilisateur à lui-même. En effet, comme l'imprimante est absente du diagramme de séquences, l'utilisation d'un message à soi-même sert à montrer que l'utilisateur doit réaliser cette opération. C'est alors à charge pour lui d'envoyer les messages nécessaires, messages qui ne sont pas représentés dans le diagramme.



**Exercice N°2 :**

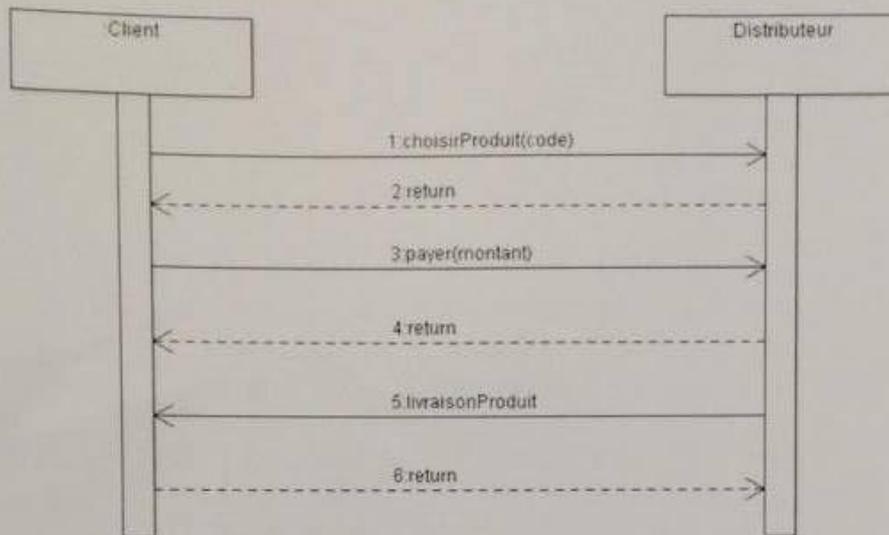
Le but de l'exercice est de montrer comment le diagramme de séquence constitue un support pour décrire les objets composant un système. Le système considéré est un distributeur automatique de produits alimentaires. L'utilisateur communique avec le système au travers de trois cas d'utilisation : **Choisir Produit**, **Payer** et **Recevoir Produit**. Le diagramme de cas d'utilisation ci-après montre ces trois cas :



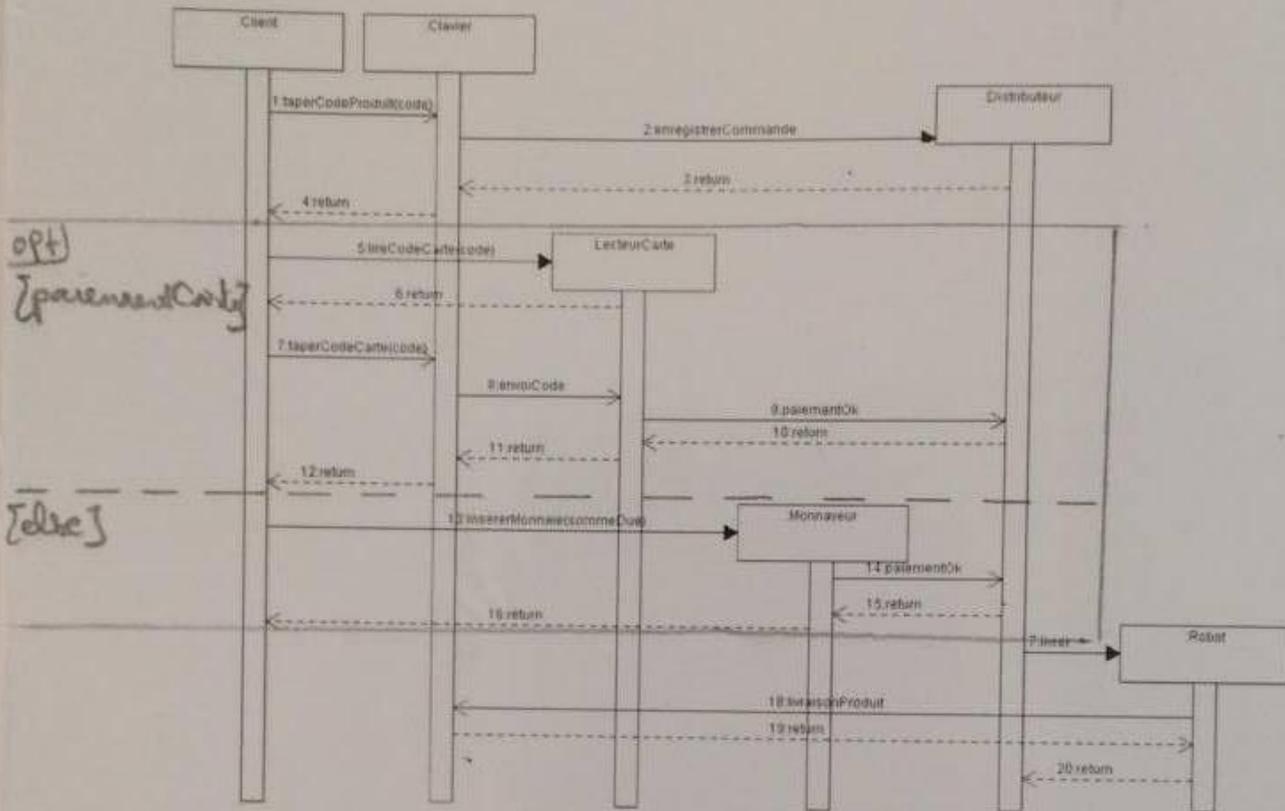
- 1) Elaborez un diagramme de séquences qui illustre les cas d'utilisation, en respectant la séquence globale basée sur l'achat d'un produit : choix, paiement puis réception du produit.
- 2) Décomposez les messages de chaque diagramme de séquences pour découvrir des objets du système. Par exemple, le choix du produit s'effectue en tapant sur un clavier.

**Corrigé :**

1. Le diagramme de séquence ne pose pas de problème particulier. Il montre les différentes interactions formées par des messages synchrones.

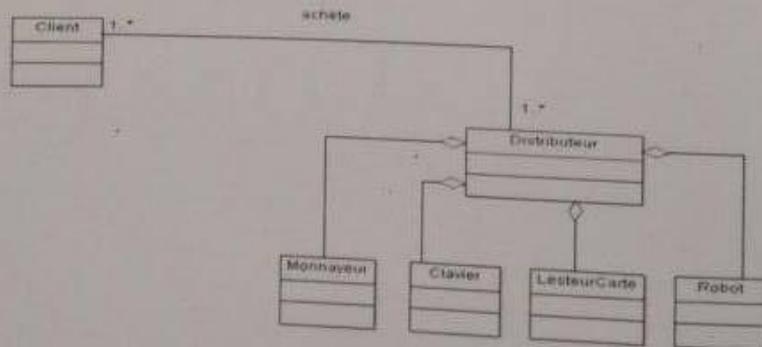


2. Les différents messages du diagramme précédent ont été décomposées comme suit :
- choisirProduit** donne lieu au fait qu'il faut taper le code du produit sur le clavier du distributeur et que le clavier doit fournir cette information au distributeur.
  - payer** donne lieu à deux possibilités :
    - soit le paiement par carte qui fait intervenir un lecteur de cartes et le clavier.
    - soit le paiement en espèces qui fait intervenir un monnayeur.
  - livrer** s'effectue grâce à un robot qui prend les produits et les apporte au client.



Ainsi la décomposition des messages a fait apparaître quatre nouveaux objets : le clavier, le lecteur de cartes, le monnayeur et le robot. Ceux-ci apparaissent sur le diagramme de classe ci-après.

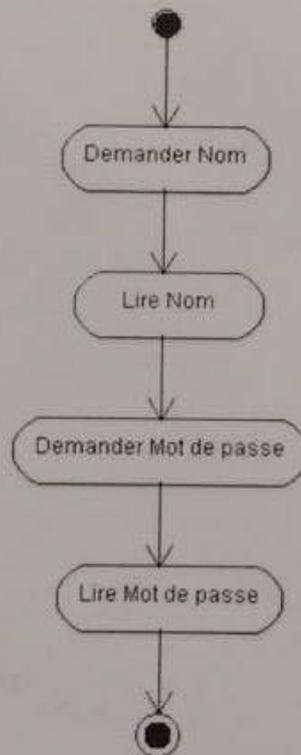
Il est, bien sûr, possible de continuer la décomposition. Par exemple, le clavier est composé de touches. Rappelons que le but est de trouver une décomposition qui correspond aux besoins de modélisation du système.



- 1) Représentez en UML les activités nécessaires que doit réaliser le système d'information pour authentifier un utilisateur en faisant l'hypothèse que l'utilisateur saisisse correctement son nom et son mot de passe. On suppose que ces deux étapes sont réalisées chronologiquement.
- 2) Ajoutez une alternative pour tester si le nom et le mot de passe sont corrects.

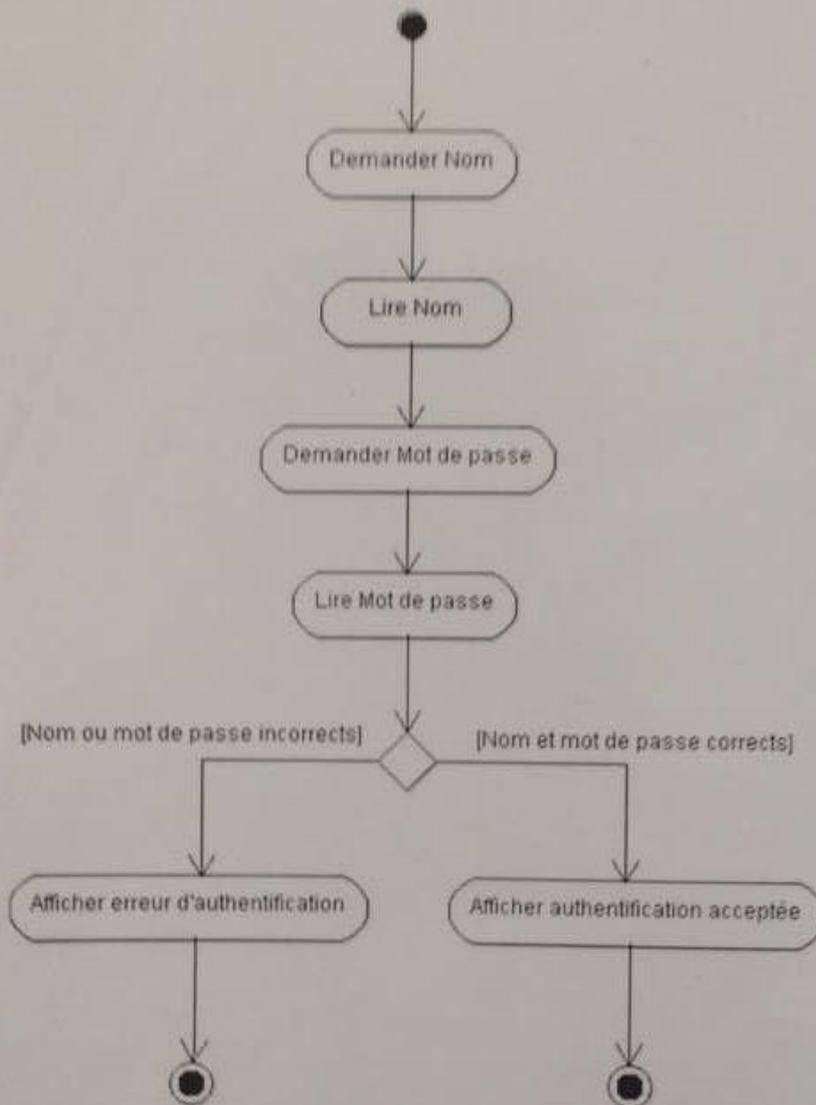
**Corrigé**

1. Le diagramme d'activité est très simple, le système doit demander à l'utilisateur son nom, attendre la réponse puis faire de même avec le mot de passe. Le diagramme est le suivant :



2. Le test d'exactitude du nom et du mot de passe s'effectue à l'aide d'une alternative et de deux conditions de garde exclusives.

Le diagramme d'activités modifié est illustré ci-après.



**Exercice N°2 :**

Un virement international dans l'Union Européenne requiert les indications suivantes :

- Nom et adresse du bénéficiaire,
- Montant du virement,
- Code BIC de la banque qui est un identifiant unique de 8 caractères,
- Numéro du compte bancaire au format IBAN qui est une représentation unifiée,
- Message à l'attention du bénéficiaire,

Par ailleurs, il existe trois options distinctes laissées au choix du donneur d'ordre du virement :

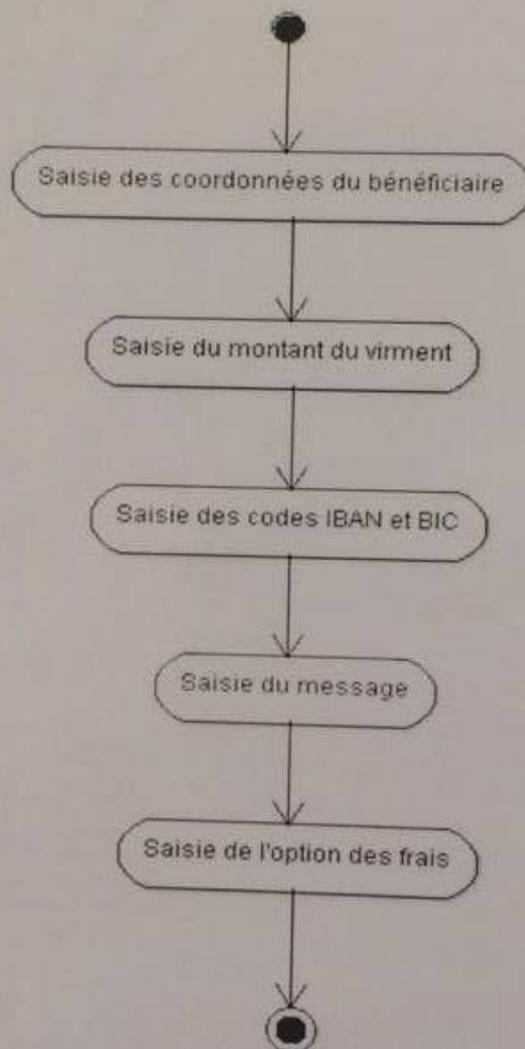
- L'option partagée où le donneur d'ordre et le destinataire paient respectivement les frais de leur banque respective,
- L'option frais à charge du donneur d'ordre où le donneur d'ordre paie l'ensemble des frais,
- L'option frais à charge du bénéficiaire où le bénéficiaire paie l'ensemble des frais.

Le système pris en compte est le site Web de la banque qui offre la possibilité à ses utilisateurs de saisir un virement européen en ligne.

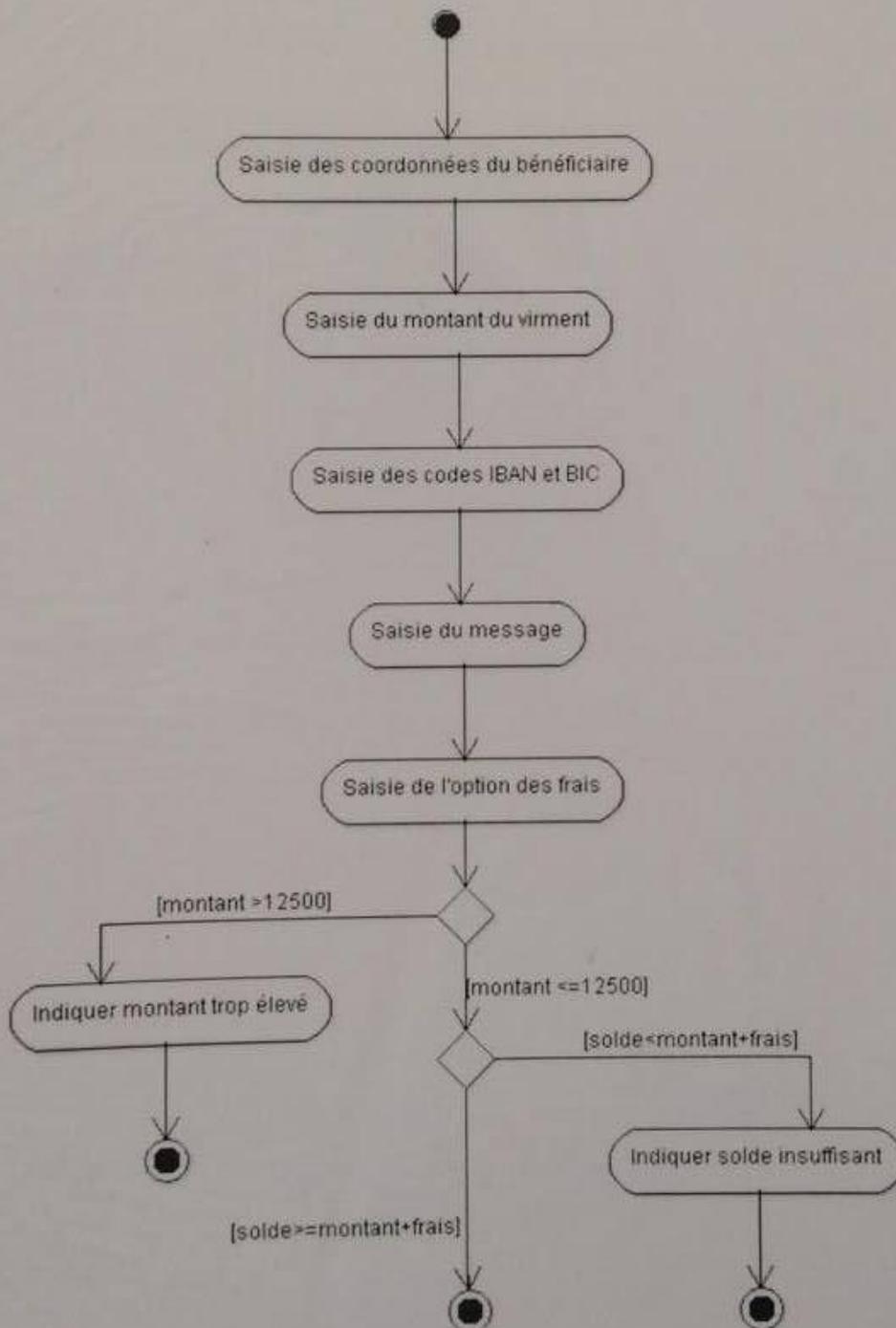
- 1) Représentez en UML les différentes saisies que doit réaliser le système auprès de l'utilisateur pour préparer un virement européen.
- 2) Le montant d'un virement européen ne peut pas dépasser 12 500 Euro, sinon il convient d'utiliser une autre procédure dite du virement international qui demande d'autres informations et dont le coût est plus élevé. Modifiez le diagramme pour que le système prévienne l'utilisateur si le montant dépasse 12 500 Euros. Prenez également en compte le cas où le total du montant et des frais dépasse le solde de son compte.
- 3) Augmentez le diagramme précédent en incluant :
  - a. Le calcul des frais (en interrogeant la banque du bénéficiaire si nécessaire).
  - b. Le virement en lui-même.
  - c. La mise à jour du solde du compte du donneur d'ordre et du bénéficiaire.

**Corrigé :**

1. Le premier diagramme est simple, le système demande à l'utilisateur de saisir les différentes informations. Il est tracé ci-après.

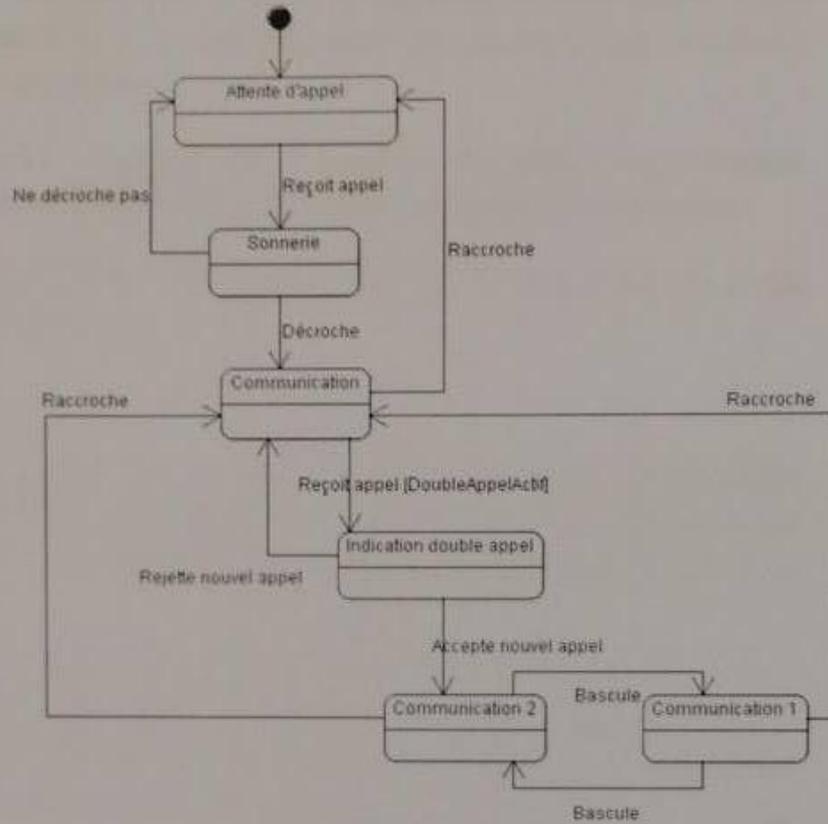


2. Les deux tests relatifs au montant du virement sont exécutés à la suite de la saisie. Les frais du virement sont pris en compte. Le diagramme d'activités incluant les deux tests est illustré ci-après.



3. Pour calculer les frais dans le cas d'un virement où tous les frais sont à la charge du bénéficiaire, il est nécessaire d'en interroger la banque. Par ailleurs, lors de la transmission du virement, c'est la banque du bénéficiaire qui le reçoit. Par conséquent, le diagramme doit contenir deux travées : l'une relative à la banque du donneur d'ordre et l'autre à la banque du bénéficiaire.

Le calcul du montant à virer dépend de l'option. Ce calcul prend en compte le fait que la banque du bénéficiaire va déduire ses propres frais. Quand tous les frais sont à la charge du bénéficiaire, les frais d'émission sont déduits du montant à virer. Quand tous les frais sont à la charge du donneur d'ordre, les frais de réception sont ajoutés au montant à virer. Le diagramme complet est illustré ci-après.



**Exercice N°1 :**

Le but de l'exercice est de décrire les différents états de la situation professionnelle d'une personne et les transitions correspondantes. La personne peut, par exemple, être étudiante, salariée, sans activité, indépendante ou retraitée.

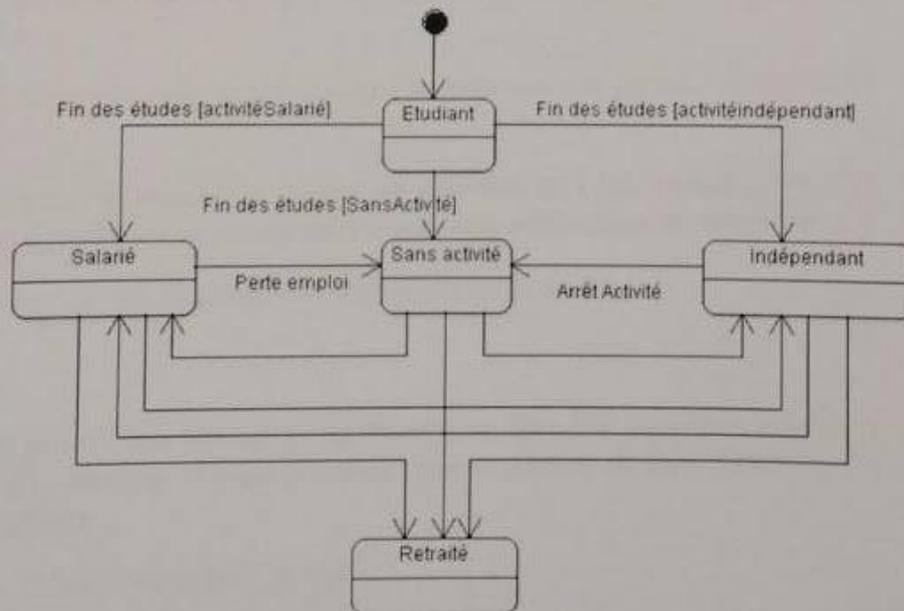
Au début de sa situation professionnelle, une personne est étudiante. Ne prenez pas en compte les activités simultanées comme la possibilité d'être simultanément salarié et indépendant.

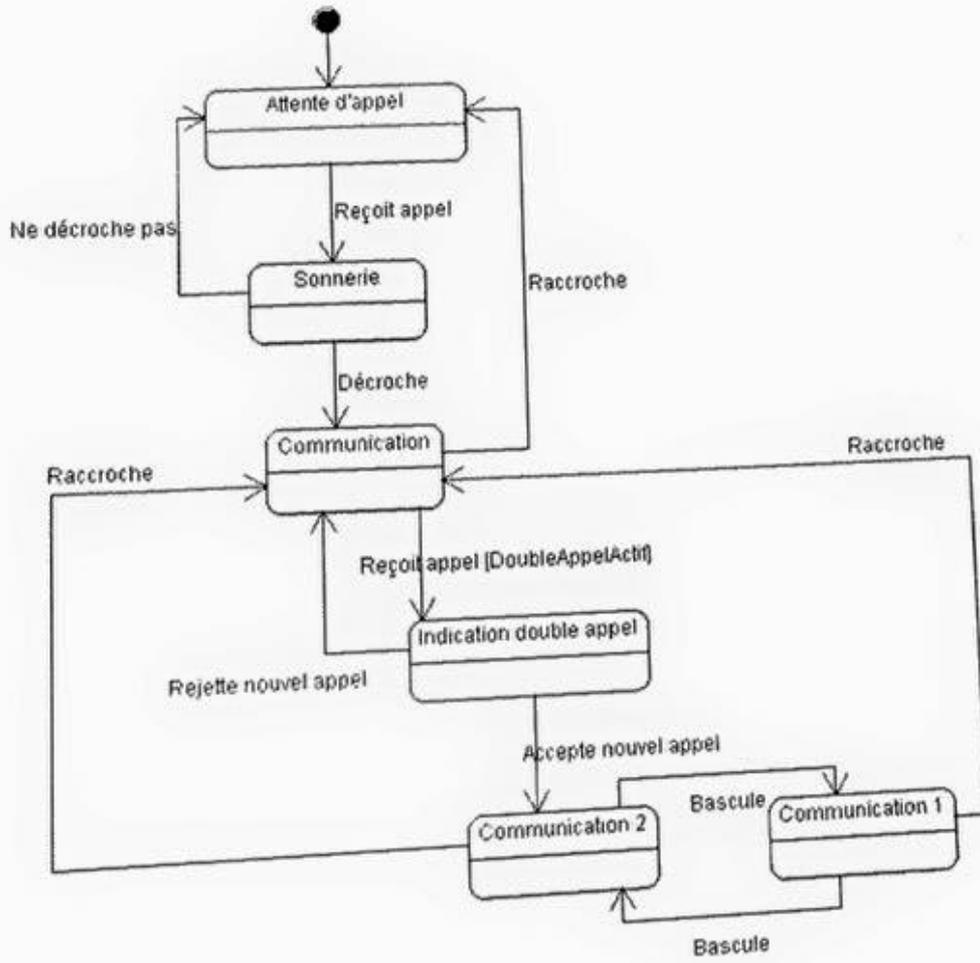
Construisez le diagramme d'états-transitions correspondant (utilisez les conditions de garde pour différencier les possibilités multiples).

**Corrigé :**

Le diagramme d'états-transitions est représenté ci-après. Après l'état initial, la personne devient étudiante. A la fin des études, la personne est soit salariée, soit indépendante, soit sans activité. Ensuite, elle peut passer d'une activité à une autre ou devenir sans activité au cours de sa carrière professionnelle.

Tous les événements n'ont pas été explicités sur le diagramme car ils sont directement liés au changement d'états. Par exemple, pour la transition située entre l'état **Indépendant** et l'état **Salarié**, comme entre l'état **Sans activité** et l'état **Salarié**, l'événement associé est simplement **Devenir Salarié**. Quant au passage à l'état **Retraité**, il intervient quand l'âge de la retraite est atteint.





**Exercice N°2 :**

Le but de l'exercice est de s'intéresser aux changements d'états d'un téléphone portable lors de la réception d'appels, notamment en prenant en compte la possibilité de gérer deux conversations en alternant (principe du double appel).

Elaborez un diagramme d'états-transitions représentant les différents états que peut prendre le téléphone portable (le téléphone est bien sûr allumé).

**Corrigé :**

Le diagramme montrant les états et transitions d'un téléphone portable lors de la réception d'appels est illustré ci-après. Le double appel est pris en compte de façon optionnelle grâce à une condition de garde. En effet, si un appel est reçu alors qu'une communication est en cours, le téléphone ne reçoit l'événement que si le double appel est actif. Dans le cas contraire, c'est le réseau qui indique l'occupation à l'appelant (ou le renvoie vers la messagerie vocale).

Lorsque l'appelé a accepté deux appels simultanés, le téléphone est dans l'un des deux états **Communication 2** ou **Communication 1** avec des événements de bascule entre les deux. Si un événement **Raccroche** se produit, le téléphone passe à nouveau dans l'état **Communication**.