



MASTER 1 INFORMATIQUE I2A

Master DVL Master ITVL

FINANCE

HISTOIRE

GÉOGRAPHIE

INFORMATIQUE

MATHÉMATIQUES

SCIENCES POUR L'INGÉNIEUR

FRANÇAIS LANGUE ÉTRANGÈRE

ADMINISTRATION ÉCONOMIQUE ET SOCIALE

DIPLÔME D'ACCÈS AUX ÉTUDES UNIVERSITAIRES

MASTER MENTION INFORMATIQUE

Parcours Informatique Avancée et Applications (I2A)



Centre de Télé-enseignement
Universitaire

<http://ctu.univ-fcomte.fr>

FILIÈRE INFORMATIQUE

● **VVI7MPO**

Modélisation et programmation orientées objet - UML - TD

Mme DAMY - SYLVIE
sylvie.damy@univ-fcomte.fr



**UNIVERSITÉ DE
FRANCHE-COMTÉ**



UNIVERSITÉ
BOURGOGNE FRANCHE-COMTÉ

Cette brochure a été réalisée en L^AT_EX 2 ϵ .

Centre de Télé-enseignement. Université de Franche-Comté.

Contents

1	Diagramme de classe : Enoncés	1
1.1	Formes géométriques	1
1.2	Mails	1
1.3	Supports de stockage	2
1.4	Connexion à un ensemble de serveurs	2
1.5	Passage de diagrammes de classe UML à des modèles relationnels	2
1.6	Gestion d'une conférence scientifique	3
1.7	Articles et factures	4
1.8	Les vocabulaires du Dublin Core Metadata Initiative	4
1.9	Sauvegarde de fichiers	5
2	Diagrammes de classe : Corrigés des exercices	7
2.1	Formes géométriques	7
2.2	Mails	8
2.3	Supports de stockage	9
2.4	Connexion à un ensemble de serveurs	10
2.5	Passage de diagrammes de classe UML à des modèles relationnels	10
2.6	Gestion d'une conférence scientifique	13
2.7	Articles et factures	17
2.8	Les vocabulaires du Dublin Core Metadata Initiative	18
2.9	Sauvegarde de fichiers	19
3	DCU : Enoncés des exercices	21
3.1	L'authentification	21
3.2	FormaPem : L'étudiant	21
3.3	ReparAuto : Garage automobile	22
3.4	Gestion des cours en ligne	22
3.5	Le guichet automatique de banque	22
3.6	Gestion de biens immobiliers	23
3.7	Sauvegarde de fichiers	23
4	Diagrammes de cas d'utilisation : Corrigés des exercices	25
4.1	L'authentification	25
4.2	FormaPem : L'étudiant	26
4.3	ReparAuto : Garage automobile	27
4.4	Gestion des cours en ligne	28
4.5	Le guichet automatique de banque	32
4.6	Gestion de biens immobiliers	33
4.7	Sauvegarde de fichiers	34

5	Diagramme de Séquences : Enoncés des exercices	37
5.1	La lampe de chevet	37
5.2	Retrait d'espèces	37
5.3	GAB : Guichet automatique de banque	38
5.4	Gestion des cours en ligne	38
5.5	Gestion de biens immobiliers	38
5.6	Sauvegarde de fichiers	39
6	Diagramme de Séquences : Corrigés des exercices	41
6.1	La lampe de chevet	41
6.2	Retrait d'espèces	42
6.3	GAB : Guichet automatique de banque	43
6.4	Gestion des cours en ligne	44
6.5	Gestion de biens immobiliers	45
6.6	Sauvegarde de fichiers	46
7	Sujets d'examen	47
7.1	UML : Le coach sportif	47
7.2	Les vocabulaires du Dublin Core Metadata Initiative	49
7.3	Gestion de biens immobiliers	50
7.4	Le code de la route	51
7.5	Sauvegarde de fichiers	53
7.6	Bibliothèque	55
7.7	Centre de soins et dossiers médicaux	58
7.8	QCMs	62
7.9	Association d'étudiants	64
7.10	Suivi des étudiants	66
7.11	Taxis	68
7.12	Hospitalisations	69
7.13	Société de livraison express à domicile	71
7.14	Observatoire	73
7.15	Structures d'un établissement de recherche	75
7.16	Réservation de salles de cours et de matériel pédagogique	76
7.17	Diagramme de séquence : Gestion d'un emprunt dans une bibliothèque	76
7.18	Sujets sans corrigé	78

Chapter 1

Diagramme de classe : Énoncés

1.1 Formes géométriques

Considérons le diagramme de classes ci-dessous : Répondez aux questions suivantes :

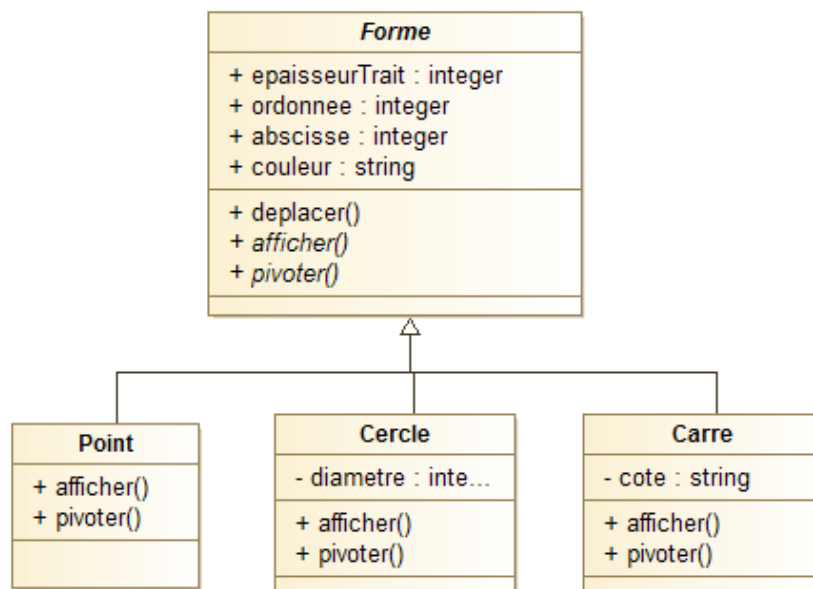


Figure 1.1 : Formes géométriques

1. L'attribut diamètre est-il accessible par 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. Comment modifier le modèle pour obtenir des formes graphiques composées de plusieurs formes ?

1.2 Mails

Un mail comporte un titre ainsi que l'adresse de son destinataire et d'une date. Il est composé d'un en-tête et d'un corps, et il peut éventuellement contenir une ou plusieurs pièces jointes.

L'en-tête comporte une date, et l'adresse retour. Le corps du mail comporte un contenu de type texte.

Tracez le diagramme de classe correspondant à cette description.

1.3 Supports de stockage

Considérons les informations suivantes :

- Les périphériques de stockage sont de deux types : amovibles ou fixes.
- Les disquettes, les clés USB sont des périphériques de stockage amovibles et les disques durs sont des périphériques de stockage fixes.

Définir un diagramme de classe correspondant à ces informations.

1.4 Connexion à un ensemble de serveurs

Définir un diagramme de classe permettant de décrire les informations pour la gestion de la connexion d'un utilisateur à un ensemble de serveurs.

L'utilisateur dispose d'un unique login, et mot de passe pour l'ensemble des serveurs. Chaque serveur a une adresse. Lorsque l'utilisateur se connecte sur un serveur, le système stocke la date et l'heure de connexion de l'utilisateur sur le serveur.

1.5 Passage de diagrammes de classe UML à des modèles relationnels

Proposez pour chacun des diagrammes de classes suivants un modèle relationnel correspondant.

1.5.1 Associations

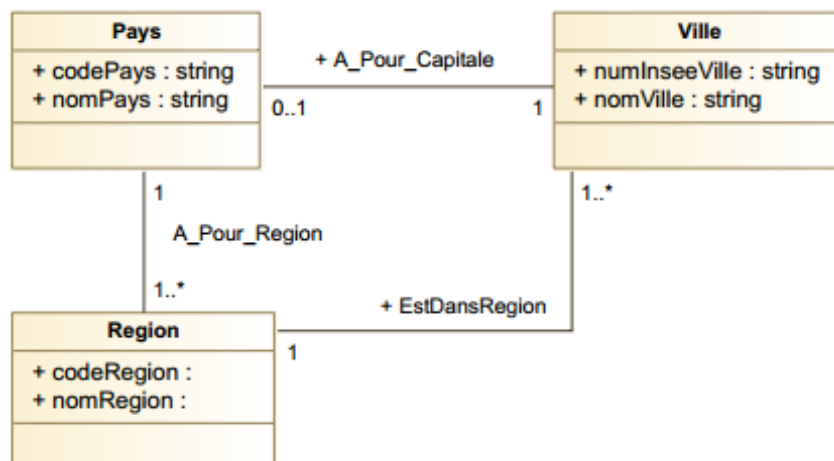


Figure 1.2 : Pays, capitales et régions



Figure 1.3 : Auteurs d'articles

1.5.2 Classes association

Reprenons ici le diagramme de classe créé dans l'exercice "connexion à un ensemble de serveurs".

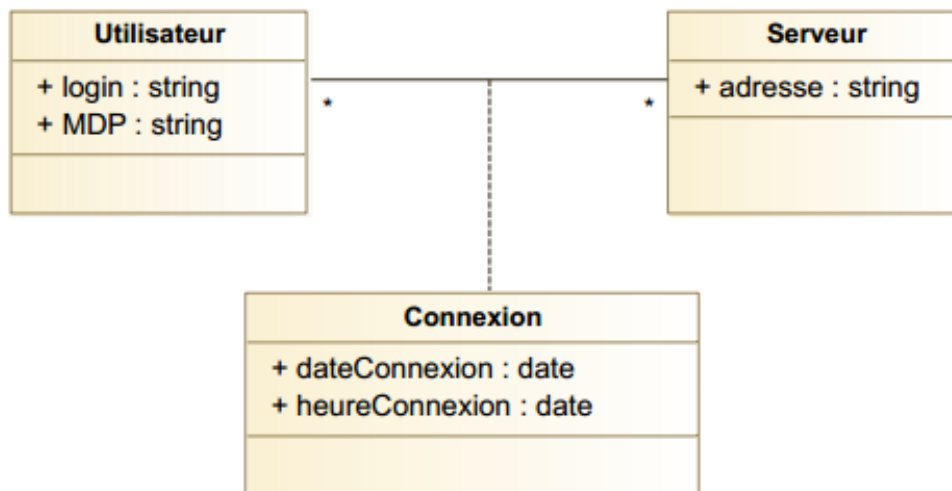


Figure 1.4 : Classe association

1.5.3 Héritage

Pour exprimer l'héritage dans un modèle relationnel il existe 3 méthodes qui présentent chacune des avantages et des inconvénients.

Représenter le diagramme de classe représentant des formes géométriques (figure 1.1) sous la forme d'un modèle relationnel en suivant chacune des solutions suivantes :

1. Traduire chaque classe en une table
2. Traduire chaque classe "feuille" en une table
3. Créer une seule table

1.6 Gestion d'une conférence scientifique

Nous voulons définir un diagramme de classe pour la gestion d'une conférence scientifique. Les informations concernant ce sujet sont les suivantes :

1. Une conférence se compose de plusieurs sessions. Une conférence a un titre, un lieu, une date de début et une date de fin.
2. Chaque session possède un titre, une date et une heure de début de session. Une session ne peut avoir lieu que pour une seule conférence.

3. Les participants participent à une session soit en tant qu'orateur soit en tant que public soit en tant que public et orateur.
 4. Un article est présenté lors d'une session. Il existe un ordre de présentation des articles dans une session. Une session peut être composée de la présentation de un à plusieurs articles, et un article n'est présenté que dans une seule session. Un article a un titre et un nombre de pages.
 5. Un article est écrit par un ou plusieurs auteurs. Un auteur a écrit au moins un article.
 6. Un article est soit un article court, soit un article long (il est obligatoirement l'un ou l'autre). Il est composé de sections numérotées et porte un titre. La destruction d'un article entraîne la destruction des sections le composant.
 7. Un auteur est une personne dont on veut le numéro de téléphone.
 8. Un participant est une personne dont on veut connaître un RIB.
 9. Une personne est composée d'un nom, d'un prénom, d'un organisme, d'un mail.
- Concevoir un diagramme de classe correspondant à cette description d'une conférence scientifique.

1.7 Articles et factures

L'objectif de cet exercice est de créer un diagramme de classe décrivant le modèle objet d'une application qui gère pour un magasin :

- les articles vendus,
- les factures.

Un article peut être *simple* ou *composite* :

- un article simple est composé d'un seul article comme par exemple un stylo. un article composite est composé de plusieurs articles simples. Par exemple trois stylos et deux ramettes de papier.

Chaque article est identifié par une référence. Il a un prix unitaire.

Un article composite ou lot a pour prix le total des prix des articles qu'il contient multiplié par un taux de remise.

Les articles simples peuvent être des stylos ou des ramettes. Un stylo a une couleur et un descriptif. Une ramette a un grammage, un nombre de pages et un descriptif.

Une facture comporte le nom du client et le détail : référence article et nombre d'articles achetés.

Nous ne gérons pas la clientèle du magasin, ni les ventes des articles. Ce qui nous intéresse est de pouvoir éditer une facture pour chaque vente.

Établissez le diagramme de classe correspondant à cet énoncé en justifiant vos choix.

1.8 Les vocabulaires du Dublin Core Metadata Initiative

L'objectif de cet exercice est de spécifier sous la forme d'un diagramme de classe, le modèle abstrait des vocabulaires utilisés dans les descriptions de métadonnées du Dublin Core.

- Un vocabulaire est un ensemble constitué d'un ou plusieurs termes. Chaque terme est membre d'un ou plusieurs vocabulaires.
- Un terme est une propriété, une classe, un schéma de codage de vocabulaire ou un schéma de codage de syntaxe.
- Une propriété peut avoir plusieurs domaines qui sont des classes.
- Une propriété peut avoir une ou plusieurs images qui sont elles-mêmes des classes.

- Une propriété peut être sous-propriété d'une ou plusieurs autres propriétés.
- Une ressource peut être membre d'un ou plusieurs schémas de codage de vocabulaire. C'est une instance d'une ou plusieurs classes.
- Une classe peut être sous-classe d'une ou plusieurs autres classes.
- Un schéma de codage de syntaxe est une classe.

Attention ce schéma ne correspond pas à ce que nous avons vu jusqu'à maintenant. Ici le détail des classes n'est pas décrit.

1.9 Sauvegarde de fichiers

Le logiciel de sauvegarde de l'entreprise "Dupont" stocke les informations dont il a besoin pour la gestion des sauvegardes dans une base de données "Sauve_Fiche". Cette base de données permet de savoir quels fichiers doivent être sauvegardés et quand ils ont été sauvegardés. On a besoin ici de connaître le nom des fichiers et leur chemin d'accès.

Différents types de sauvegardes peuvent être réalisés. Lors du lancement de la sauvegarde on enregistre la date et l'heure de lancement de la sauvegarde son type, le support utilisé. On distingue 2 types de sauvegarde :

- Complète : on sauvegarde tous les fichiers ;
- Incrémentale : on ne sauvegarde que les fichiers qui ont été modifiés depuis la dernière sauvegarde. Il est donc nécessaire de conserver la date de la dernière modification de chaque fichier lors de l'exécution de la sauvegarde.

Une sauvegarde peut être effectuée sur différents types de supports : bande magnétique, disque dur, Mais l'intégralité d'une sauvegarde est réalisée sur un unique type de support. Une première phase dite de préparation permet de réaliser les choix concernant la sauvegarde.

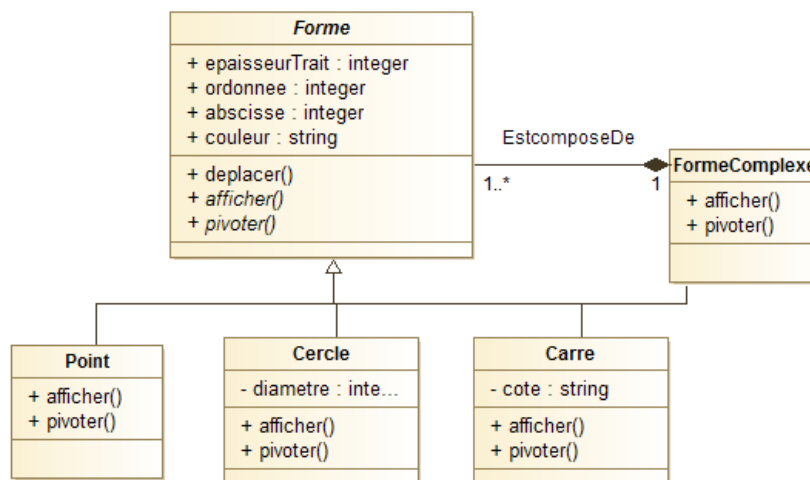
Lors de la sauvegarde, l'application récupère les dates et les heures de début et de fin de sauvegarde ainsi que la liste des fichiers à sauvegarder et stocke chacun des fichiers stockés sur un ou plusieurs supports. Chaque support de sauvegarde est identifié par un numéro. La sauvegarde est conservée durant 1 an dans un local dédié aux sauvegardes. Passé ce délai, le support peut être réutilisé pour d'autres sauvegardes. Les utilisateurs du système disposent d'un login et d'un mot de passe. Créez un diagramme de classe permettant de décrire les données de la base de données Sauve_Fiche qui permet d'enregistrer les informations concernant toutes les sauvegardes réalisées.

Chapter 2

Diagrammes de classe : Corrigés des exercices

2.1 Formes géométriques

1. L'attribut diamètre est-il accessible par un objet Carré ?
L'attribut **diamètre** est un attribut privé de la classe **Cercle**. Il n'est donc pas accessible par un objet qui n'appartient pas à cette classe.
2. Un objet Cercle possède-t-il un attribut couleur ?
La classe **Cercle** est une **sous-classe** de la classe **Forme**; un objet de la classe **Cercle** possède donc tous les attributs de la classe **Forme** et donc l'attribut **couleur**. Donc chaque objet **Cercle** possède un attribut **couleur**.
3. Peut-on appliquer la méthode déplacer à un objet Point ?
Tout comme pour les attributs, les méthodes d'une **super-classe** peuvent être appliquées sur un objet d'une sous-classe.
La méthode **déplacer**, qui est définie dans la super-classe **Forme**, peut donc être appliquée à un objet de la sous-classe **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 ?
La méthode **pivoter** est présente dans toutes les classes, et est redéfinie dans les classes **Point**, **Cercle** et **Carré** grâce à la notion de **polymorphisme**. Ici c'est la détermination du type du paramètre de la méthode lors de son appel, qui permet de choisir quelle implémentation sera choisie (dans quelle classe).
5. Comment modifier le modèle pour obtenir des formes graphiques composées de plusieurs formes ?
Une *forme graphique composée* aura les mêmes attributs et les mêmes méthodes qu'une forme graphique. On a donc un **héritage** : la classe **FormeComplexe** hérite de la classe **Forme**.
Par ailleurs un objet **FormeComplexe** est composé de plusieurs objets **Forme**. Et nous supposons que la destruction d'un objet **FormeComplexe** entraîne la suppression des objets **Forme** qui le composent. Nous avons donc une **composition**.



2.2 Mails

Un mail comporte un titre ainsi que l'adresse de son destinataire et d'une date. Il est composé d'un en-tête et d'un corps, et il peut éventuellement contenir une ou plusieurs pièces jointes. L'en-tête comporte une date, et l'adresse retour. Le corps du mail comporte un contenu de type texte.

Tracez le diagramme de classe correspondant à cette description.

Nous allons créer quatre classes :

- Mail
- Entête
- Corps
- PieceJointe.

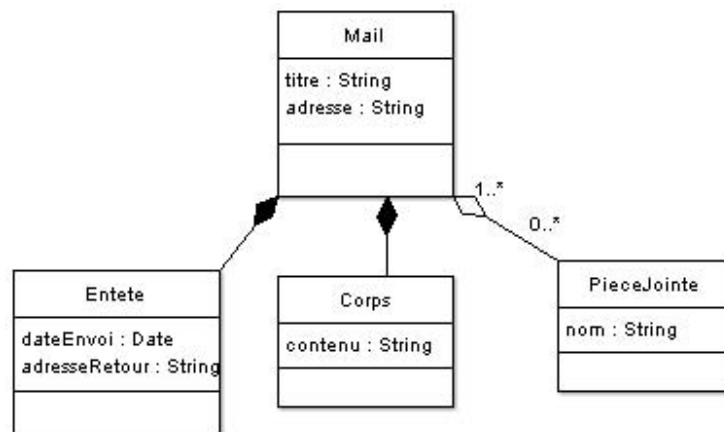
Nous définissons maintenant les relations entre ces différentes classes.

Les relations entre la classe **Mail** d'une part et les classes **Entête**, **Corps** et **PieceJointe** d'autre part sont au minimum des **agrégations**. En effet ces éléments n'ont pas lieu d'exister sans le mail. Pour savoir si nous avons des compositions il faut vérifier :

- la cardinalité maximale du côté du composé (ici mail) est au maximum de 1,
- la destruction d'un objet composé entraîne la destruction des composants.

Pour les classes **Entête** et **Corps** les deux conditions sont vérifiées on aura donc des **compositions**. Par contre pour la classe **Piecejointe** la cardinalité n'est pas de 1 ; une pièce jointe peut être utilisée dans plusieurs mails.

On obtient donc le diagramme de classe :



2.3 Supports de stockage

Considérons les informations suivantes :

- Les périphériques de stockage sont de deux types : amovibles ou fixes.
- Les disquettes, les clés USB sont des périphériques de stockage amovibles et les disques durs sont des périphériques de stockage fixes.

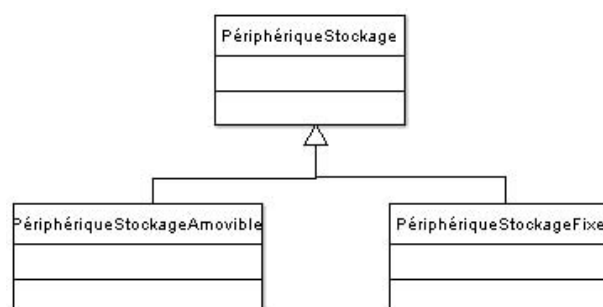
Définir un diagramme de classe correspondant à ces informations.

Les relations entre classes que nous traitons ici sont des relations **d'héritage**.

Nous définissons tout d'abord les sous-classes :

- `PériphériqueStockageAmovible`
- `PériphériqueStockageFixe`

de la classe `PériphériqueStockage`.

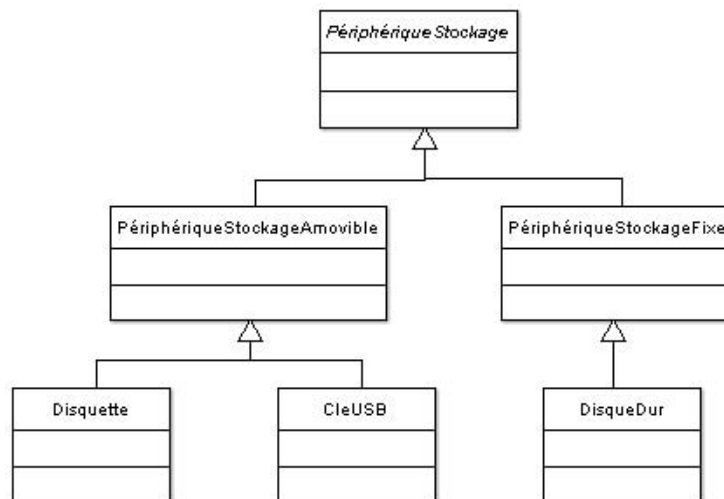


Ensuite nous pouvons définir les sous-classes :

- `Disquette`
- `CleUSB`

de la classe `PériphériqueStockageAmovible` et la classe `DisqueDur` comme sous-classe de la classe `PériphériqueStockageFixe`.

On remarquera que la classe `PériphériqueStockage` a été déclarée **abstraite**. On suppose ici qu'un périphérique est obligatoirement Fixe ou amovible, et donc que l'on aura pas besoin d'instancier la classe `PériphériqueStockage`.



2.4 Connexion à un ensemble de serveurs

Un utilisateur peut se connecter à plusieurs serveurs et un serveur peut accepter plusieurs utilisateurs. La date et l'heure concernant une connexion d'un utilisateur à un serveur. Ces deux attributs sont placés dans une classe association **Connexion** qui lie les classes **Utilisateur** et **Connexion**. On obtient ainsi le diagramme de classe cf. figure 2.1 .

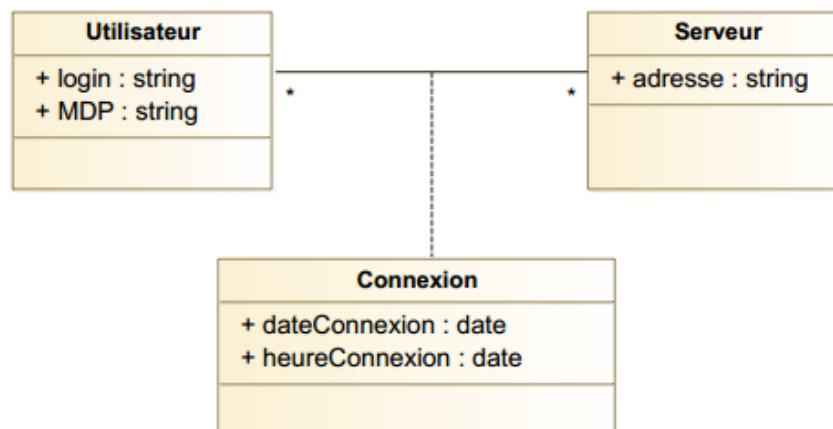


Figure 2.1 : Diagramme de classe : Connexion à un serveur

2.5 Passage de diagrammes de classe UML à des modèles relationnels

2.5.1 Associations

Pays, capitales et régions

Nous créons ici trois tables, une par classe.

- L'association **A_Pour_Capitale** est représentée par l'attribut **capitale** dans la table **Pays**. Il s'agit d'une clé étrangère qui a pour valeur le **numInsee** de la ville qui est capitale du pays.

- L'association `A_Pour_Region` est représentée par l'attribut `codePays` dans la table `Region`. Il s'agit d'une clé étrangère qui a pour valeur le `codePays` du pays dans lequel se trouve la région.
- L'association `Est_Dans_Region` est représentée par l'attribut `codeRegion` dans la table `Ville`. Il s'agit d'une clé étrangère qui a pour valeur le `codeRegion` de la région dans laquelle se trouve la ville.

Ce qui nous donne le modèle relationnel 2.2 .

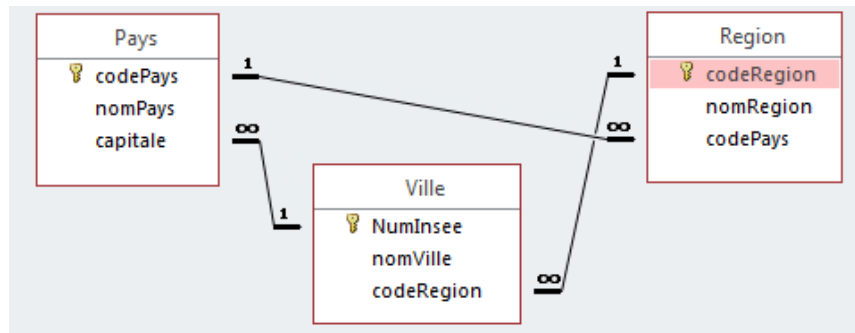


Figure 2.2 : Modèle relationnel : Pays, capitales et régions

Auteurs et articles

Pour représenter l'association entre les classes `Personne` et `Article` nous devons ici créer une table `Auteur`. En effet une personne peut rédiger plusieurs articles et un article peut être rédigé par plusieurs personnes. La multiplicité maximale est de * de chaque côté de l'association. On obtient ainsi le modèle relationnel suivant :

- Le fait qu'une personne soit auteur d'un article est représenté par l'attribut `numPersonne` de la table `Auteur`. Il s'agit d'une clé étrangère.
- Le fait qu'un article ait pour auteur une personne est représenté par l'attribut `DOIArticle` de la table `Auteur`. Il s'agit ici aussi d'une clé étrangère.
- la table `Auteur` a pour clé l'ensemble d'attributs (`numPersonne`, `DOIArticle`).

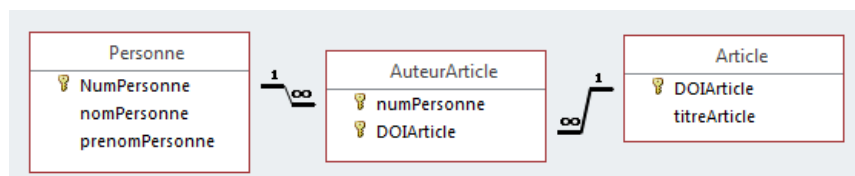


Figure 2.3 : Auteurs

2.5.2 Classes association

Les deux classes `Utilisateur` et `Serveur` sont traduites en tables ainsi que la classe association `Connexion` dans laquelle on trouvera en clés étrangères les clés des deux tables liées par la classe association. On obtient ainsi le modèle relationnel 2.4 . Un identifiant est ajouté dans les tables `Utilisateur` et `Connexion` en tant que clé primaire.

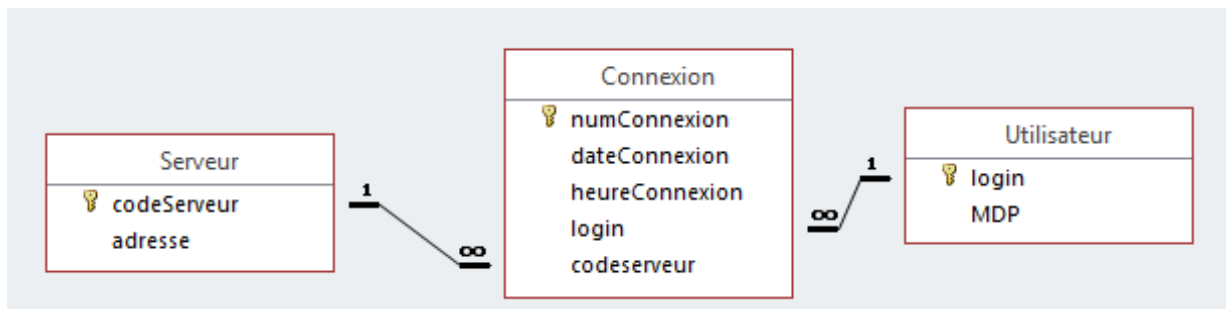


Figure 2.4 : Modèle relationnel issu d'une classe association

2.5.3 Héritage

Traduire chaque classe en une table

On crée ici 4 tables : **Forme**, **Point**, **Cercle** et **Carre**. On ajoute dans la table **Forme** un identifiant **numForme** en tant que clé primaire. Cet attribut est utilisé dans les autres tables comme clé étrangère (cf. figure 2.5). Pour retrouver toutes les informations sur une forme il faut réaliser une jointure de la table **Forme** avec la table **Point**, **Cercle** ou **Carre**. Cette solution suppose que toute création ou recherche de forme suppose d'utiliser deux tables.

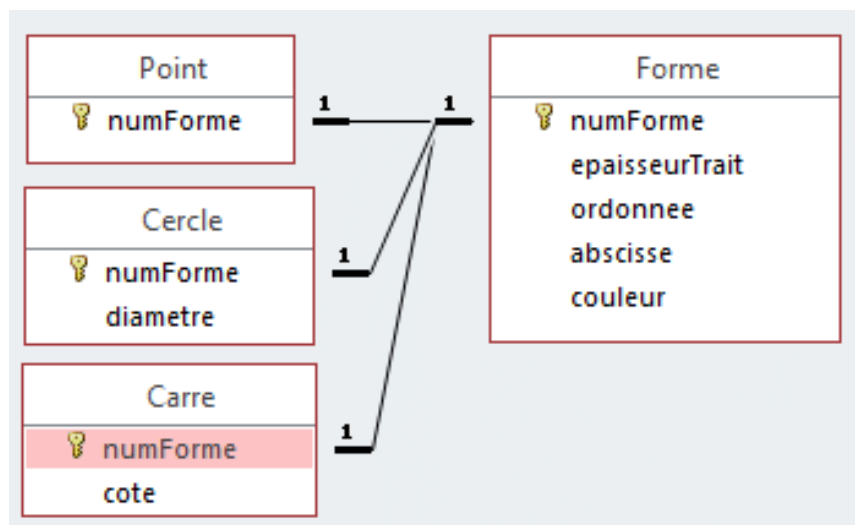


Figure 2.5 : Une table par classe

Traduire chaque classe "feuille" en une table

On crée ici (cf. figure 2.6) les tables **Point**, **Cercle** et **Carre** dans lesquelles on ajoute les attributs de la classe **Forme**. Cette solution est plus simple à utiliser car pour ajouter une forme il suffit d'ajouter un enregistrement dans la table adéquate. Et pour une recherche on travaille sur une seule table.

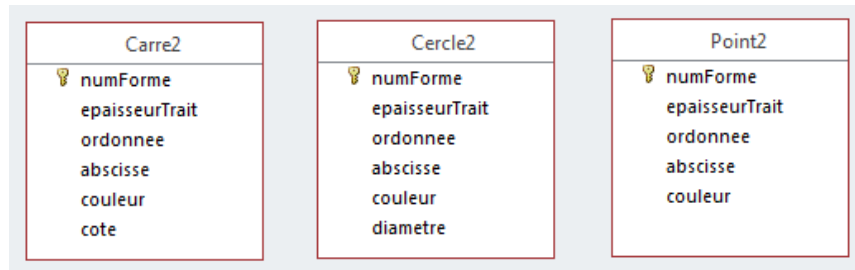


Figure 2.6 : Une table par classe "feuille"

Créer une seule table

On crée une seule table (cf. 2.7) qui contient les attributs définis dans chacune des classes et on ajoute un attribut `type_forme` qui permet de connaître le type de la forme afin de faciliter les traitements.

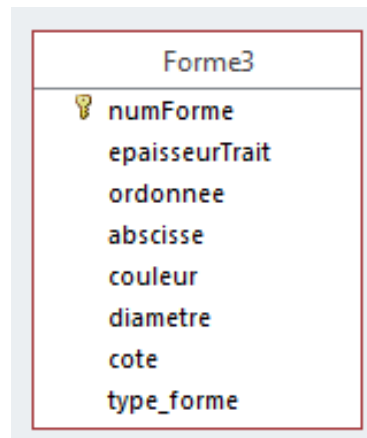


Figure 2.7 : Une seule table

2.6 Gestion d'une conférence scientifique

Concevoir un diagramme de classe pour la gestion d'une conférence scientifique.

Nous allons comme dans l'exemple de Formaperm reprendre une à une les informations données dans l'énoncé pour construire notre diagramme de classe.

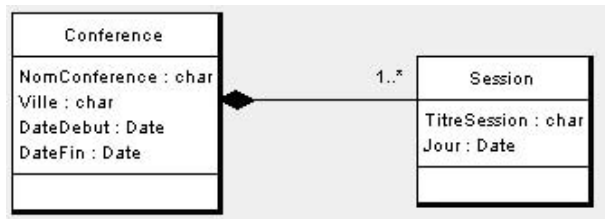
1. Une conférence se compose de plusieurs sessions. Une conférence a un titre, un lieu, une date de début et une date de fin.

Nous voyons apparaître ici deux classes de notre modèle : **Conférence** et **Session**. De plus on peut dire que la relation qui lie ces deux classes est au moins une **agrégation**. Une session ne peut être liée qu'à une seule conférence (Cf. Règle 2 de l'énoncé), et par ailleurs la suppression de la conférence entraîne la disparition de la session. Nous avons donc une **composition** entre ces deux classes.

2. Chaque session possède un titre, une date et une heure de début de session. Une session ne peut avoir lieu que pour une seule conférence.

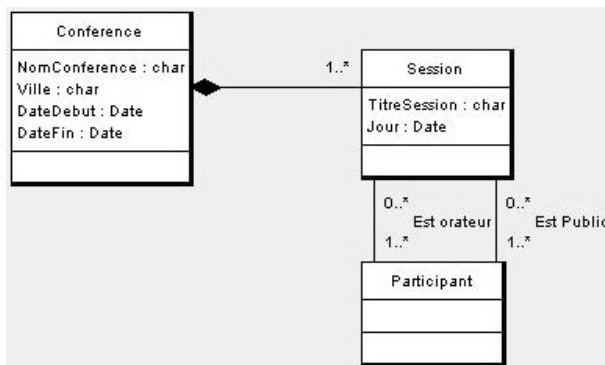
On obtient les deux classes suivantes :

3. Les participants participent à une session soit en tant qu'orateur soit en tant que public soit en tant que public et orateur.



Nous définissons ici une classe **Participant** qui a deux associations avec la classe **Session**. L'association **Est orateur** a une cardinalité 1..* du côté de la classe **Participant**. En effet une session peut avoir de 1 à plusieurs orateurs. D'autre part un participant peut être orateur dans zéro à plusieurs sessions, on a donc une cardinalité 0..* du côté de la classe **Session** sur le diagramme de classe.

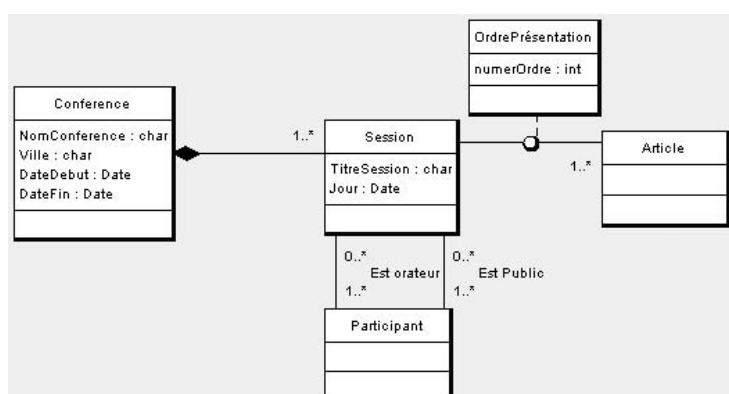
Pour ce qui est de l'association **Est public**, nous avons les mêmes cardinalités que pour l'association précédente.



4. Un article est présenté lors d'une session. Il existe un ordre de présentation des articles dans une session. Une session peut être composée de la présentation de un à plusieurs articles, et un article n'est présenté que dans une seule session. Un article a un titre et un nombre de pages.

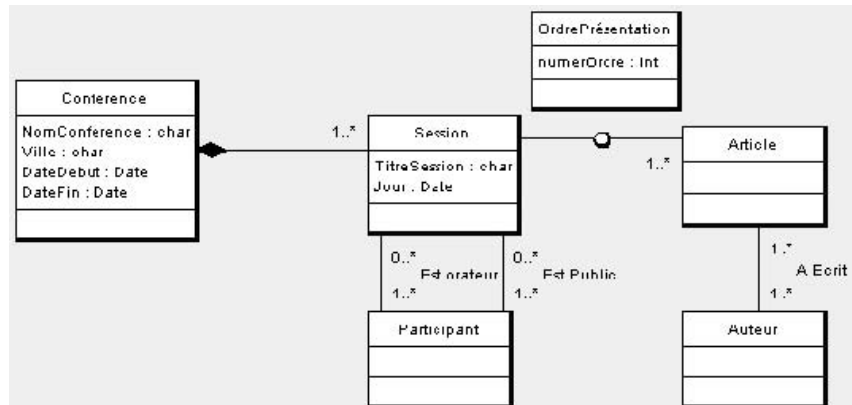
Nous avons une association **EstPrésenté** entre les classes **Session** et **Article**. Par ailleurs, nous définissons une **classe association** **OrdrePrésentation** qui donne le numéro d'ordre de présentation de l'article dans une session.

Nous avons donc les cardinalités 1 du côté de la classe **Session** et 1..* du côté de la classe **Article**.



5. Un article est écrit par un ou plusieurs auteurs. Un auteur a écrit au moins un article. Nous définissons une classe **Auteur**, et une association **A Ecrit** entre les classes **Auteur** et

Article. Les cardinalités sont de 1..* de chaque coté.

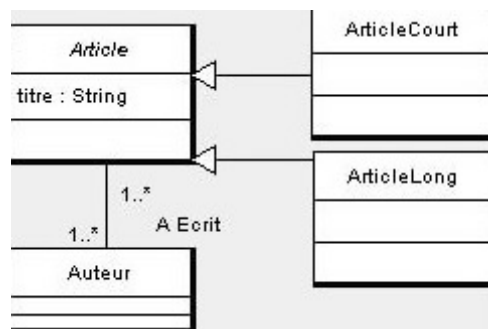


6. Un article est soit un article court, soit un article long (il est obligatoirement l'un ou l'autre). Il est composé de sections numérotées et porte un titre. La destruction d'un article entraîne la destruction des sections le composant.

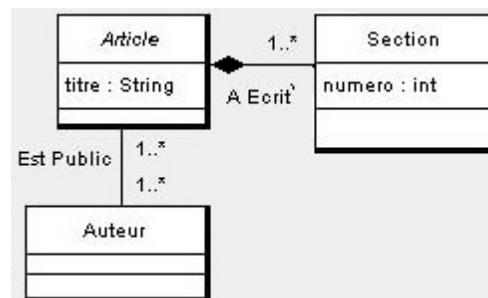
Nous complétons ici la classe `Article` en ajoutant l'attribut `titre`. De plus nous créons trois classes : `Section`, `ArticleLong` et `ArticleCourt`.

Les relations entre ces différentes classes sont :

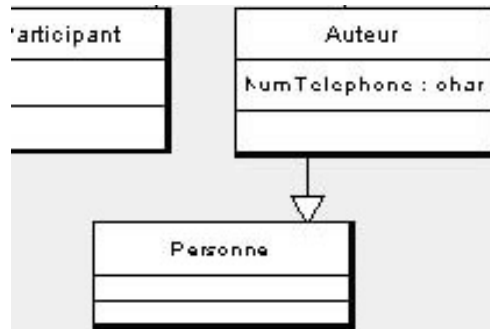
- **Héritage** entre `Article` et `ArticleCourt` et `ArticleLong`. Ces dernières classes sont des sous-classes de la classe `Article`. De plus un article étant obligatoirement de type long ou court, nous définirons la classe `Article` comme une **classe abstraite**.



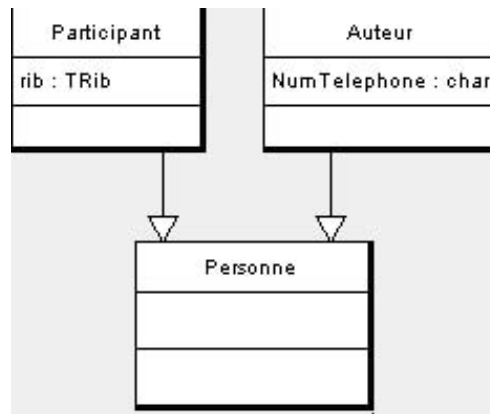
- **Agrégation** ou **composition** entre `Article` et `Section`. Une section ne correspond qu'à un seul article et la suppression de l'article entraîne celle des sections. On a donc une **composition**.



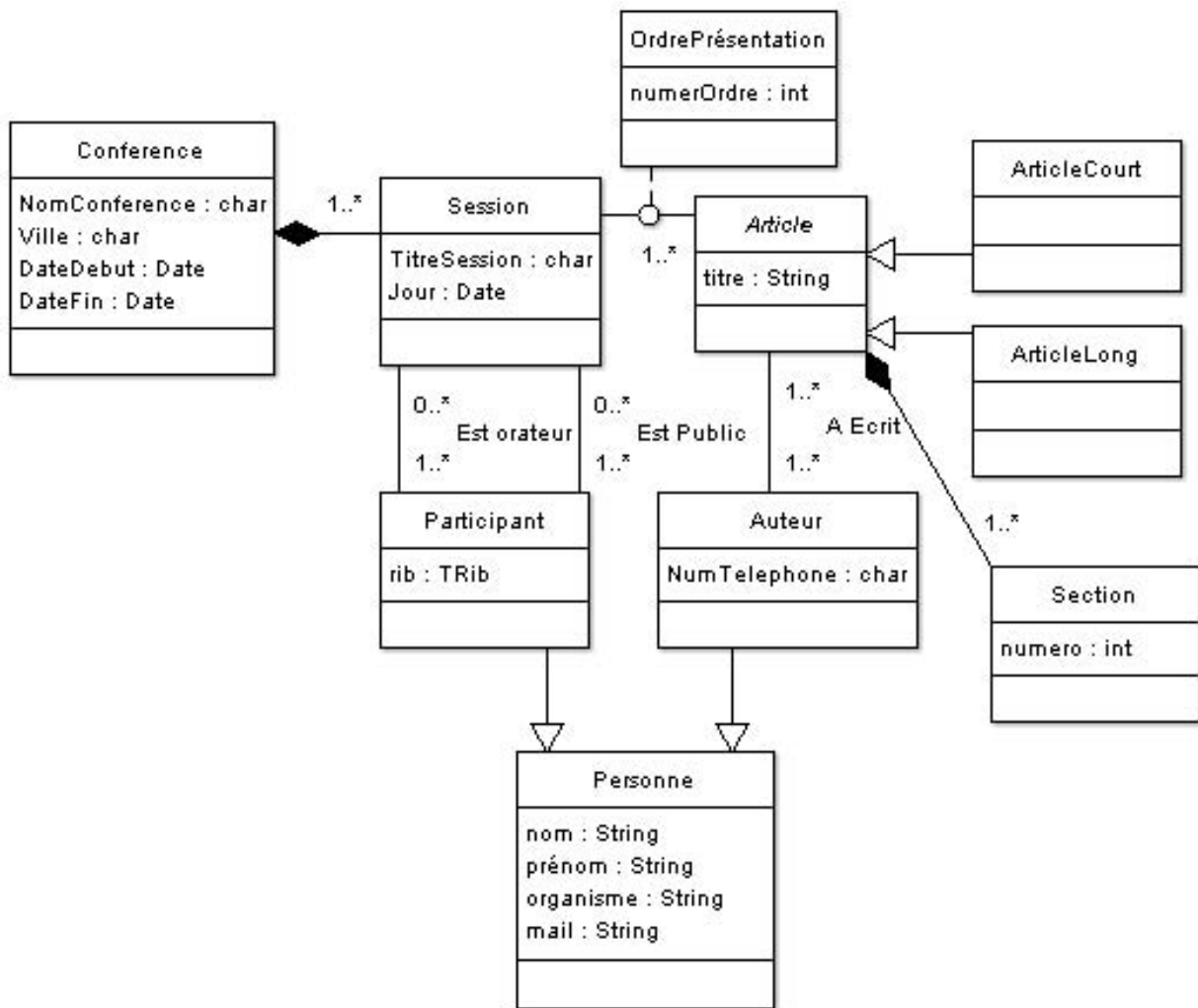
7. Un auteur est une personne dont on veut le numéro de téléphone. Nous définissons ici la classe `Personne` qui est une **super-classe** de `Auteur`. Nous avons comme attribut de cette classe `numTelephone`.



8. Un participant est une personne dont on veut connaître un RIB.
Nous avons une relation d'**héritage** entre les classes **Participant** et **Personne**.

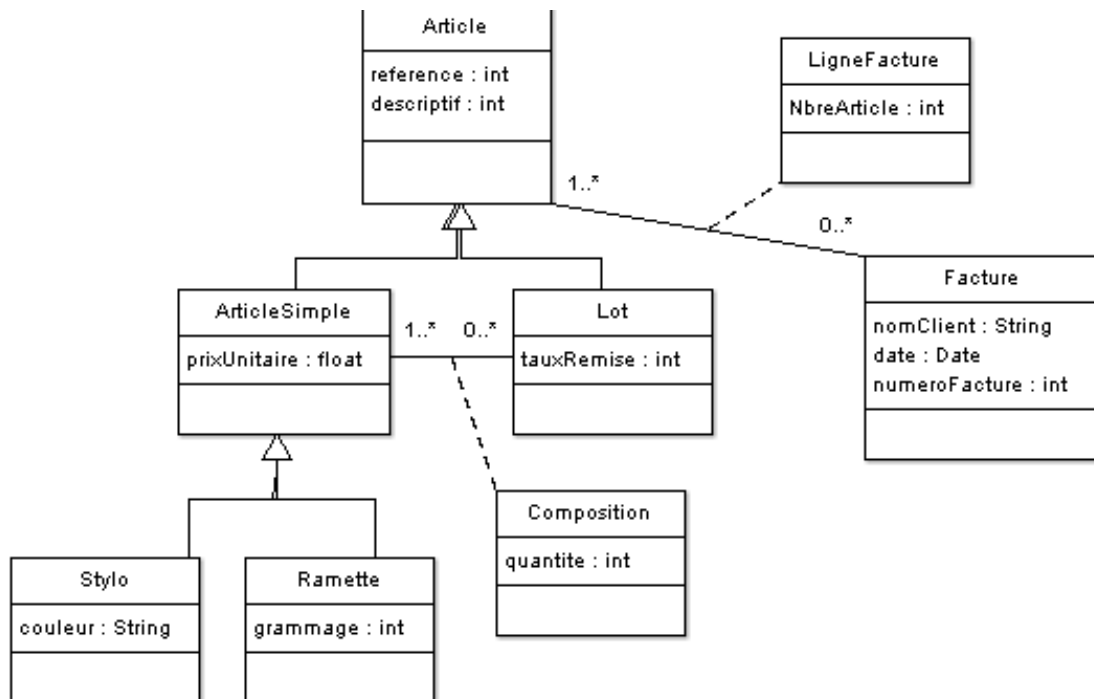
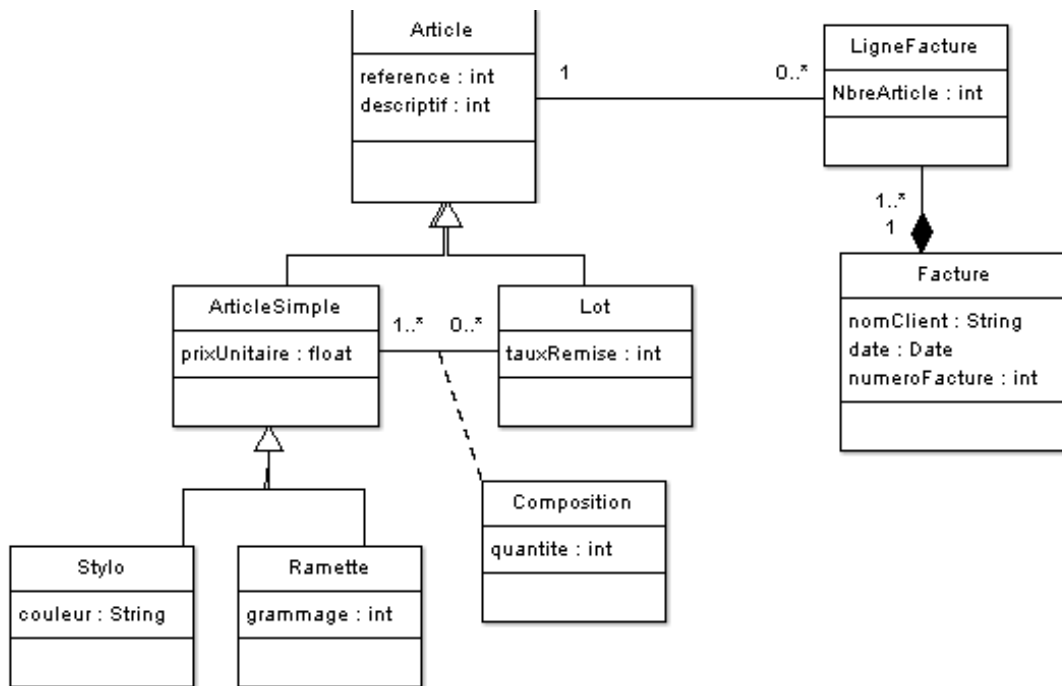


9. Une personne est composée d'un nom, d'un prénom, d'un organisme, d'un mail.
Nous obtenons pour finir le diagramme de classe :



2.7 Articles et factures

Diagramme de classe : Nous proposons ici deux solutions :



2.8 Les vocabulaires du Dublin Core Metadata Initiative

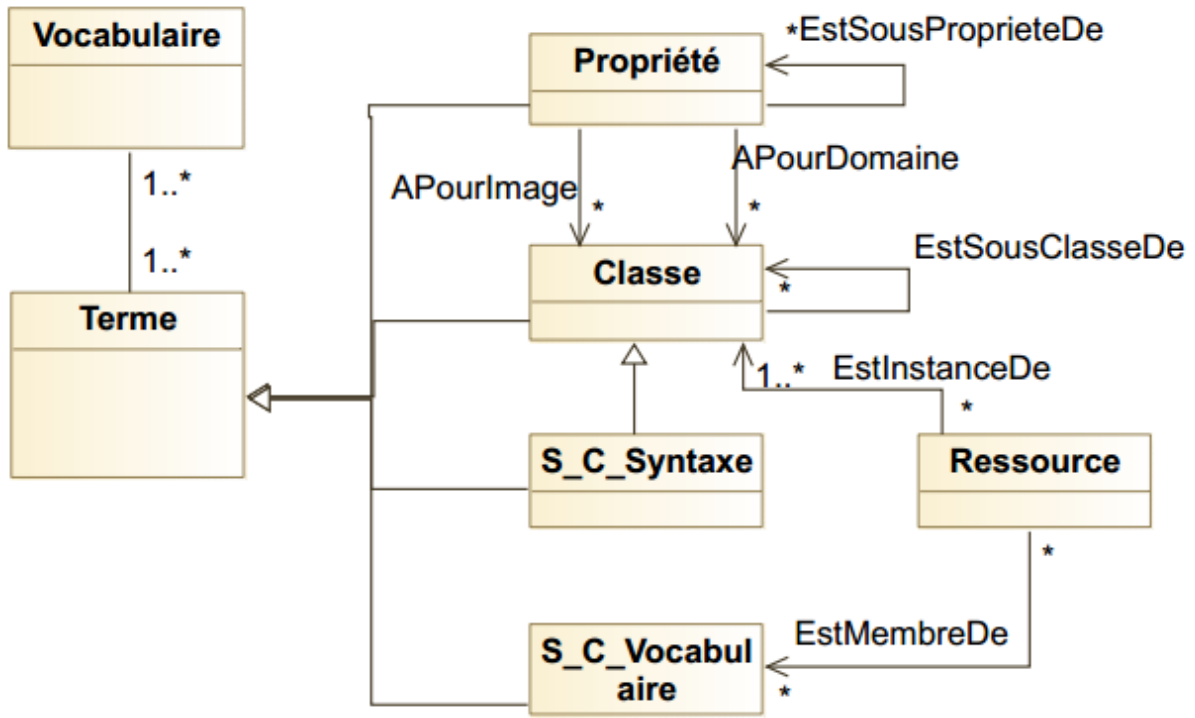
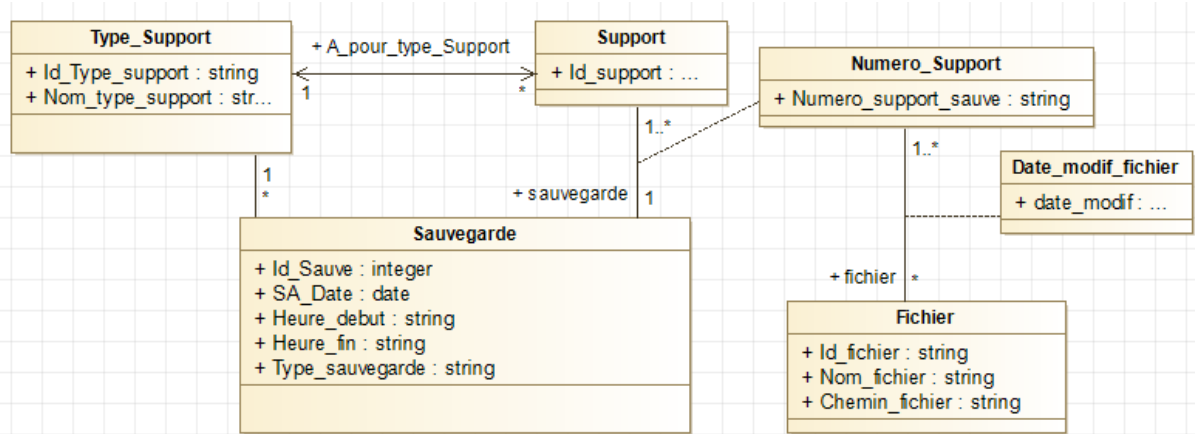


Figure 2.8 : Diagramme de classe DCMI

2.9 Sauvegarde de fichiers



Chapter 3

DCU : Enoncés des exercices

3.1 L'authentification

Nous nous proposons de décrire ici notre premier cas d'utilisation. Il s'agit de l'authentification d'un utilisateur sur un système informatique. L'utilisateur s'identifie en saisissant son nom et un mot de passe.

3.1.1 Premier cas d'utilisation : authentification simple

Décrire le diagramme de cas d'utilisation correspondant à l'authentification d'un acteur avec le système sans détail.

3.1.2 Deuxième cas d'utilisation : authentification plus détaillée

On détaille maintenant l'authentification avec la saisie du nom et du mot de passe de l'utilisateur ainsi que la vérification de ces données.

Décrire le diagramme de cas d'utilisation correspondant à cette authentification.

3.1.3 Troisième cas d'utilisation : authentification avec code optionnel

Nous ajoutons un dernier élément à notre authentification ; l'utilisateur peut ajouter un code supplémentaire pour avoir accès à des traitements supplémentaires après son authentification. écrire le diagramme de cas d'utilisation correspondant à cette dernière authentification.

Représentation textuelle du dernier cas d'utilisation

Représentez de façon textuel le cas d'utilisation détaillé de l'authentification.

3.2 FormaPem : L'étudiant

Construire le diagramme de cas d'utilisation correspondant aux étudiants du système FormaPerm. Un étudiant peut consulter ses informations personnelles, après s'être connecté (nous ne détaillons pas ici la connexion). Lors de la consultation il peut demander à modifier les informations le concernant. Il peut consulter ses résultats après s'être connecté. Et il peut afficher la liste de tous les cours (sans connexion) ou des cours auxquels il est inscrit (avec connexion). Lorsqu'il se connecte et affiche ses cours il peut alors choisir un cours et lister les documents liés à un cours et les télécharger éventuellement.

3.3 ReparAuto : Garage automobile

Le "Garage Dupont" est une entreprise qui entretient et répare des automobiles. Il dispose de deux logiciels : un logiciel de comptabilité et un logiciel de gestion des réparations appelé ReparAuto. Nous nous intéressons dans cet exercice à la description du logiciel de gestion des réparations.

ReparAuto communique avec le logiciel de comptabilité pour lui transmettre les réparations à facturer et ce dernier gère et imprime les factures de réparations.

Il est destiné en priorité au chef d'atelier, celui-ci peut y saisir les fiches de réparations et le travail effectué par les divers employés de l'atelier. Les magasiniers fournissent les pièces pour les véhicules qui ont une fiche de réparation ouverte; ils saisissent les pièces fournies dans ReparAuto depuis un terminal installé au magasin.

Enfin pour effectuer leur travail, les mécaniciens et autres employés de l'atelier vont chercher des pièces de rechange au magasin.

Lorsqu'une réparation est terminée, le chef d'atelier essaye la voiture. Si tout est en ordre, il met la voiture sur le parc clientèle et ferme la fiche de réparation informatisée. Les fiches de réparations fermées par le chef d'atelier sont ensuite exportées depuis ReparAuto par le comptable vers le logiciel de comptabilité.

Décrire le diagramme de cas d'utilisation du logiciel de gestion des réparations : ReparAuto.

3.4 Gestion des cours en ligne

Le département informatique de l'UFR Sciences et techniques de Besançon souhaite réaliser un logiciel de gestion des cours en ligne.

Le secrétariat s'occupe de la gestion des cours dans le logiciel. Il peut donc enregistrer des cours, supprimer un cours, inscrire des étudiants, inscrire un étudiant dans un cours, enlever un étudiant d'un cours. Quand le secrétariat inscrit un étudiant, il doit absolument l'inscrire à au moins un cours pour que l'inscription soit validée. Quand le secrétariat supprime un cours, il doit aussi enlever tous les étudiants de ce cours. Le secrétariat peut aussi envoyer des mails aux enseignants d'un cours. Pour réaliser toutes ces manipulations le secrétariat doit être logué.

Un enseignant d'un cours peut mettre à disposition des étudiants des documents, saisir des notes, et envoyer des mails aux étudiants ou au secrétariat. Il peut aussi préparer des tests électroniques qui pourront être passés par les étudiants à une période bien précise. Pour faire tout ceci, l'enseignant doit être logué.

Les étudiants pour leur part, peuvent consulter leurs notes et les documents de cours, envoyer des mails aux enseignants et passer des tests. Ils doivent bien sûr se loguer pour participer à chacune de ces activités, sauf la consultation de documents. Une fois sur la page du test, l'étudiant choisit de le démarrer ou de le quitter. Si l'étudiant quitte le test alors qu'il n'est pas terminé, un mail est envoyé à l'enseignant automatiquement et le test n'est plus disponible pour lui. Si l'étudiant complète le test de manière normale, les notes sont automatiquement entrées dans la base de données. L'étudiant ne peut alors plus réaliser le test (le test n'est plus disponible pour lui).

3.5 Le guichet automatique de banque

On considère un système simplifié de Guichet Automatique de Banque ou GAB. Ce système offre les services suivants :

1. Distribution d'argent à tout porteur de carte de crédit (carte Visa, ou de la banque), via un lecteur de carte et un distributeur de billets.

2. Consultation de solde de compte, dépôt en numéraire et dépôt de chèques pour les clients de la banque porteurs d'une carte de crédit de la banque.

Par ailleurs :

- Toutes les transactions sont sécurisées, via :
 - le Système d'Authentification Visa, pour les retraits effectués avec une carte Visa;
 - le Système d'Information de la banque, pour autoriser les transactions effectuées avec une carte de la banque, mais également pour accéder au solde des comptes.
- Le GAB nécessite des actions de maintenance, telles que le rechargement en billets du distributeur, la récupération des cartes avalées et des chèques déposés, etc.

3.5.1 Question 1

Quels sont les utilisateurs du système ?

3.5.2 Question 2

Quels sont les cas d'utilisation ?

3.5.3 Question 3

Etablir le diagramme de cas d'utilisation de ce système.

3.6 Gestion de biens immobiliers

On souhaite créer un système de gestion de biens permettant à des personnes de proposer à la vente des biens ou d'en acheter.

Le site mis à disposition peut être consulté par tout visiteur, par contre pour pouvoir acheter ou mettre en vente un bien il faut s'inscrire. La description d'un bien est composée d'un type (Maison, Appartement), d'une localisation (quartier, ville, code postal), d'une surface, d'un nombre de pièces et d'un prix en euros. Un administrateur peut valider la description d'un bien ou au contraire l'annuler ; il valide par ailleurs une demande d'achat ou l'annule. Lorsque qu'une demande d'achat est validée, le client demandeur est alors prévenu par une alerte sur son compte et peut alors accéder aux données complètes de l'offre (détail sur le vendeur : téléphone,) et ainsi prendre contact avec le vendeur pour finaliser l'achat. Une personne qui réalise une demande d'achat doit se connecter, visualiser l'ensemble des offres correspondant à des critères de choix (Type, localisation, prix,), en choisir une et faire une offre de prix. Le dépôt d'une offre de vente suppose une connexion, la description de l'offre (Type, localisation, prix, contact)

Question 1 Quels sont les différents acteurs du système ? Question 2 Construire le diagramme de cas d'utilisation de ces différents acteurs

3.7 Sauvegarde de fichiers

Le logiciel de sauvegarde de l'entreprise "Dupont" stocke les informations dont il a besoin pour la gestion des sauvegardes dans une base de données "Sauve_Fiche". Cette base de données permet de savoir quels fichiers doivent être sauvegardés et quand ils ont été sauvegardés. On a besoin ici de connaître le nom des fichiers et leur chemin d'accès. Différents types de sauvegardes peuvent être réalisés. Lors du lancement de la sauvegarde on enregistre la date et l'heure de lancement de la sauvegarde son type, le support utilisé. On distingue 2 types de sauvegarde :

- Complète : on sauvegarde tous les fichiers ;

- Incrémentale : on ne sauvegarde que les fichiers qui ont été modifiés depuis la dernière sauvegarde. Il est donc nécessaire de conserver la date de la dernière modification de chaque fichier lors de l'exécution de la sauvegarde.

Une sauvegarde peut être effectuée sur différents types de supports : bande magnétique, disque dur, Mais l'intégralité d'une sauvegarde est réalisée sur un unique type de support. Une première phase dite de préparation permet de réaliser les choix concernant la sauvegarde.

Lors de la sauvegarde, l'application récupère les dates et les heures de début et de fin de sauvegarde ainsi que la liste des fichiers à sauvegarder et stocke chacun des fichiers stocke sur un ou plusieurs supports.

Chaque support de sauvegarde est identifié par un numéro. La sauvegarde est conservée durant 1 an dans un local dédié aux sauvegardes. Passé ce délai, le support peut être réutilisé pour d'autres sauvegardes.

Dans l'entreprise, seul l'informaticien est autorisé à gérer les sauvegardes. Il peut ainsi décrire de nouveaux types de supports, créer de nouveaux supports en supprimer. Il lance les sauvegardes et gère les supports de sauvegarde.

Le reste du personnel peut ajouter dans la liste des fichiers à sauvegarder ses propres fichiers et faire des recherches de fichiers dans les sauvegardes.

Quel que soit le traitement à réaliser l'utilisateur doit s'identifier login + mot de passe pour avoir accès à l'application.

3.7.1 Acteurs

Quels sont les différents acteurs du système ?

3.7.2 DCU

Construire le diagramme de cas d'utilisation de ces différents acteurs.

Chapter 4

Diagrammes de cas d'utilisation : Corrigés des exercices

4.1 L'authentification

4.1.1 Premier cas d'utilisation : authentification simple

Nous avons ici un acteur primaire qui est l'utilisateur. Le système ne possède ici qu'un seul cas d'utilisation : "Authentifier". Ce qui nous donne le diagramme de cas d'utilisation présenté dans la figure 4.1 .

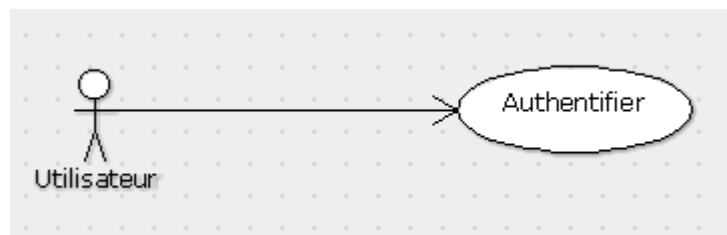


Figure 4.1 : Cas d'utilisation Authentification 1

4.1.2 Deuxième cas d'utilisation : authentification plus détaillée

Nous avons ici trois "sous-cas d'utilisation" que l'on peut lier au cas d'utilisation *Authentifier* par des relations d'inclusion (cf. figure 4.2).

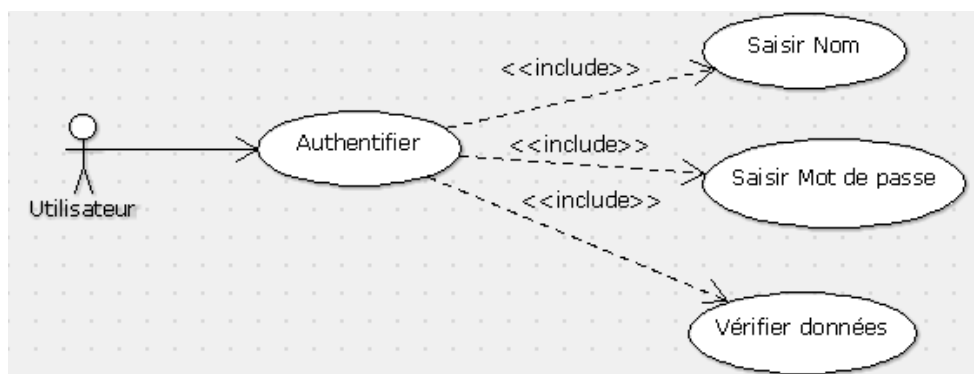


Figure 4.2 : Cas d'utilisation Authentification 2

4.1.3 Troisième cas d'utilisation : authentification avec code optionnel

Ici ce "sous-cas" d'utilisation est optionnel, nous sommes dans le cas d'une relation d'extension représentée dans la figure 4.3 .

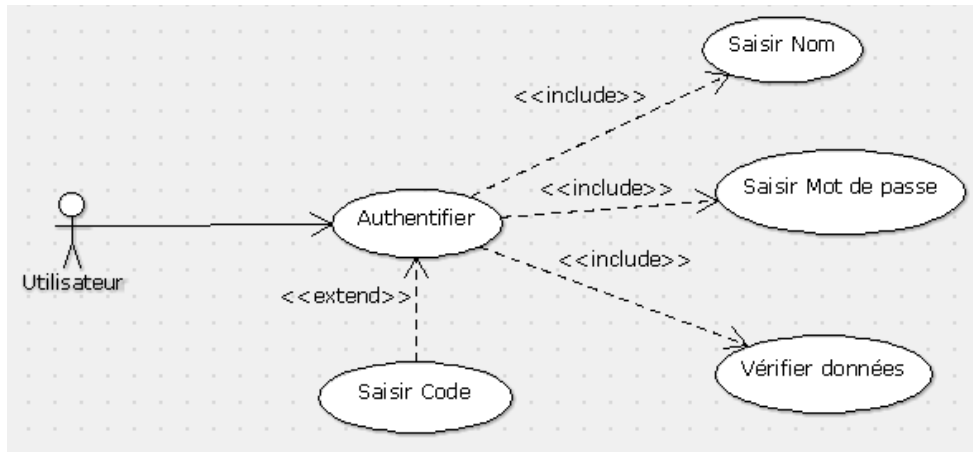


Figure 4.3 : Cas d'utilisation Authentification 3

Représentation textuelle du dernier cas d'utilisation

Reprenons le cas d'utilisation "authentification avec code optionnel", nous obtenons le tableau 4.1 .

Cas d'utilisation	Authentification avec code optionnel
Description	L'utilisateur s'authentifie en tapant son nom et son mot de passe. Ces données sont vérifiées et l'utilisateur peut en plus donner un code pour accéder à des informations complémentaires.
Acteur primaire	Utilisateur
Système	Authentification
Intervenants	Utilisateur et Base de données de login
Objectif acteur principal	Accéder au service
Sous-fonction	Vérifier le nom, vérifier le mot de passe
Conditions d'exécution	ne pas être connecté
Opérations	
1	Authentifier
2	Saisir nom
3	Saisir mot de passe
4	Vérifier données
Extensions	
1	Saisir code

Table 4.1 : Représentation textuelle du cas d'utilisation "Authentification avec code optionnel"

4.2 FormaPem : L'étudiant

Nous obtenons le diagramme suivant 4.4 .

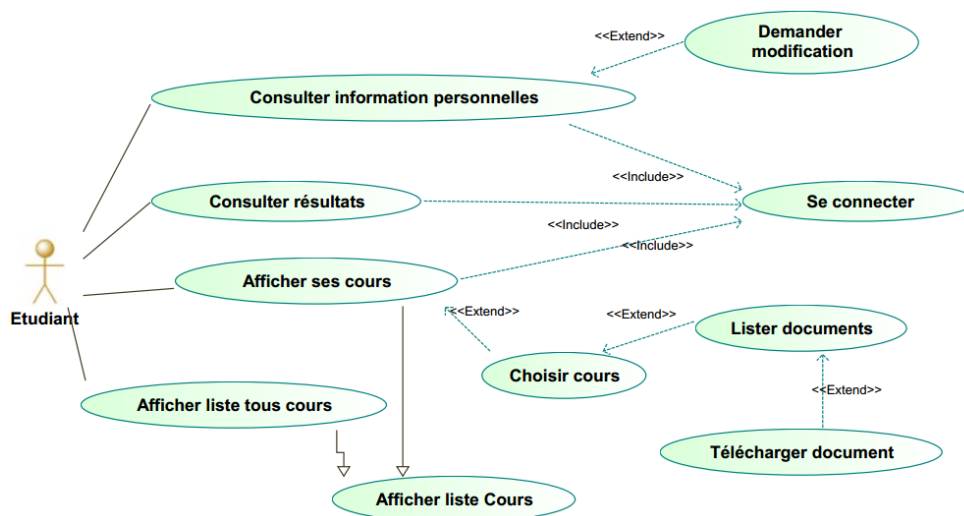


Figure 4.4 : Diagramme de cas d'utilisation Etudiant

4.3 ReparAuto : Garage automobile

Il convient tout d'abord de bien identifier qui sont les acteurs de ReparAuto. Le texte fait référence à un certain nombre d'entités ayant un lien avec notre logiciel :

- Chef d'atelier
- Mécanicien
- Autre employé
- Magasinier
- Comptable
- Le logiciel de comptabilité

Le logiciel de comptabilité est un acteur qui est lui-même un système, mais qui interagit avec le logiciel de gestion des réparations. Par contre les mécaniciens et autres employés n'interagissent pas directement avec notre système ; ils ne sont donc pas des acteurs principaux de notre système. Le diagramme de cas d'utilisation obtenu est le suivant (cf. figure 4.5).

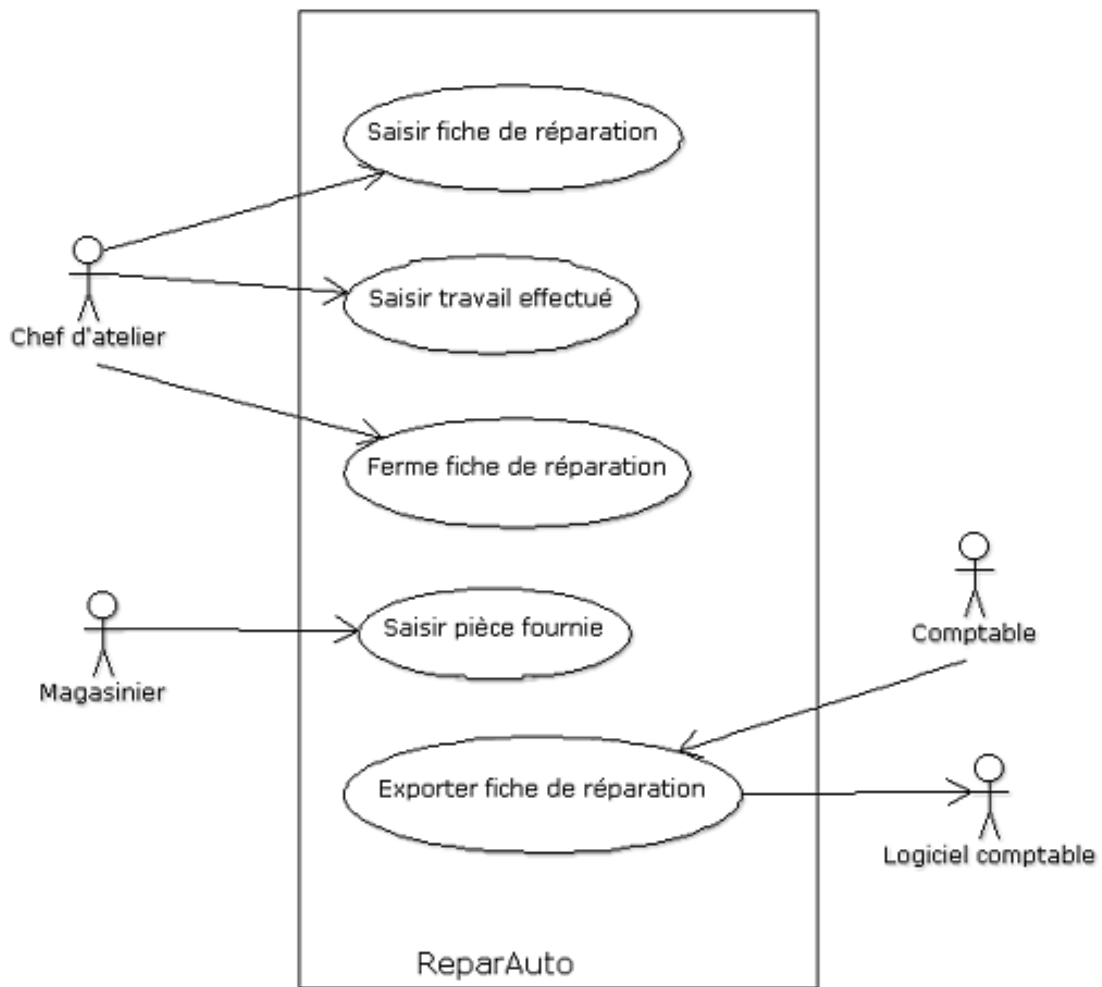


Figure 4.5 : Diagramme de cas d'utilisation de ReparAuto

4.4 Gestion des cours en ligne

L'énoncé fait apparaître les acteurs suivants :

- Secrétariat
- Enseignant
- Etudiant
- Et nous considérerons aussi l'utilisateur.

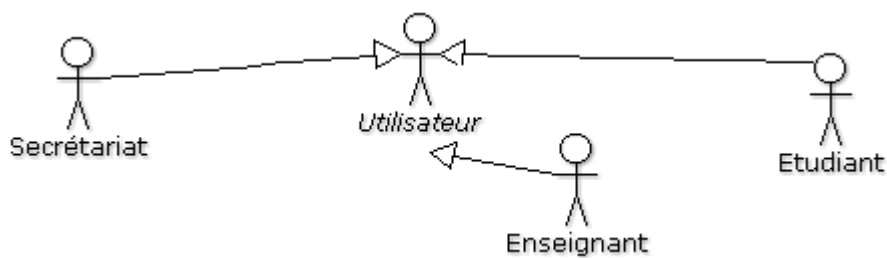


Figure 4.6 : Les acteurs de la gestion des cours

4.4.1 L'utilisateur

L'utilisateur est un acteur abstrait dont les autres acteurs héritent. Il peut se loguer et envoyer un mail, en effet ses cas d'utilisation seront réutilisés par tous les autres acteurs.

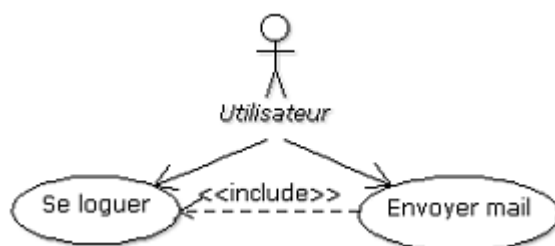


Figure 4.7 : L'acteur Utilisateur

4.4.2 Le secrétariat

Cet acteur peut être représenté par le diagramme de cas d'utilisation suivant : Nous trouvons ici

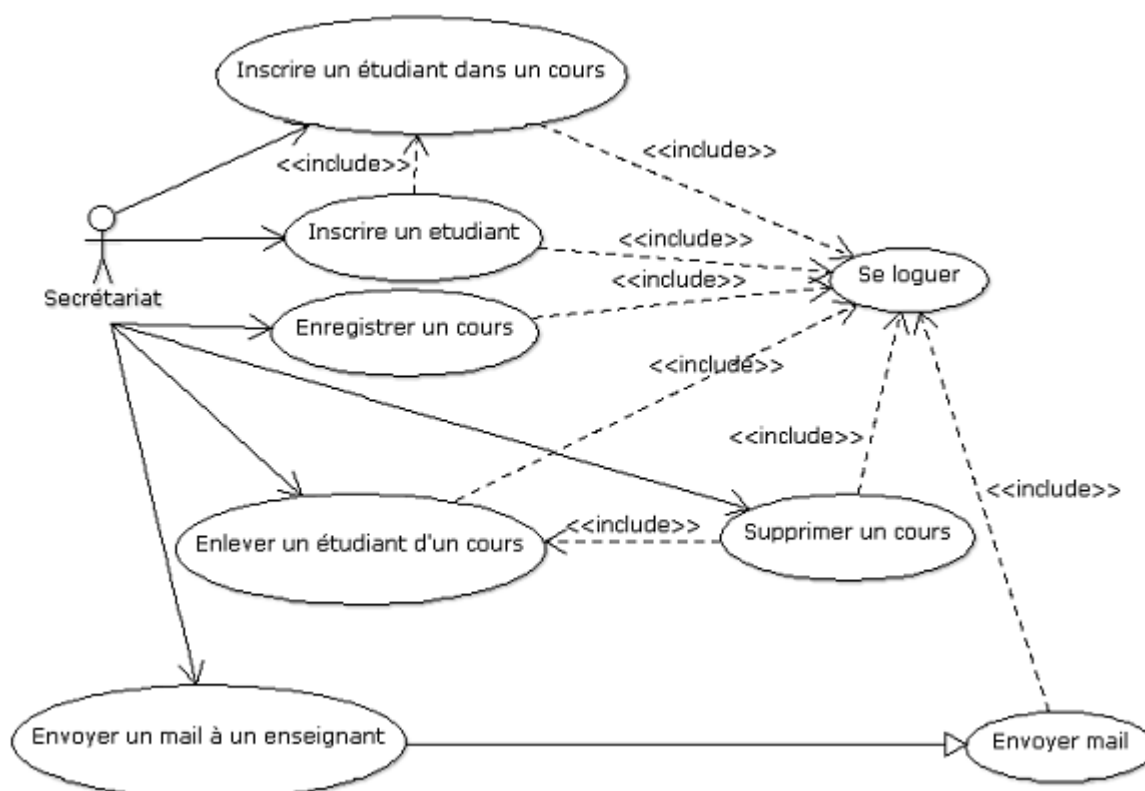


Figure 4.8 : Le secrétariat

6 nouveaux cas d'utilisation correspondant à ce que peut faire le secrétariat.

- Inscrire un étudiant : on a ici une relation `<< include >>` avec le cas d'utilisation "Inscrire un étudiant dans un cours".
- Envoyer un mail à un enseignant : une relation d'héritage lie ce cas d'utilisation à celui "envoyer un mail".

- Tous les cas d'utilisation utilisent le cas d'utilisation "Se loguer". Nous avons donc une relation d'inclusion.

4.4.3 L'étudiant

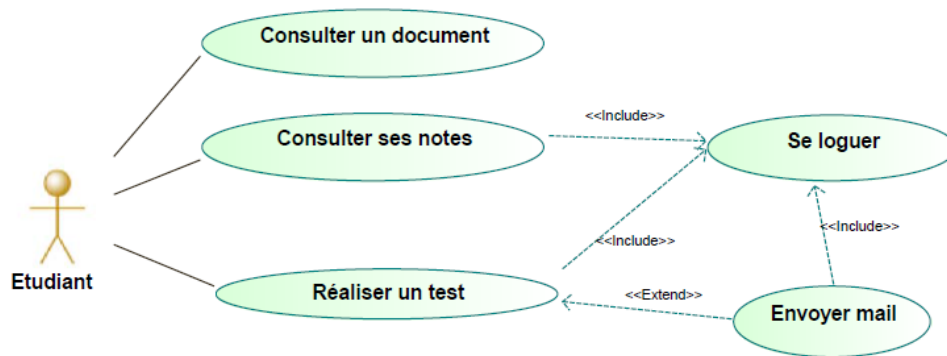


Figure 4.9 : Cas d'utilisation de l'étudiant

L'étudiant est décrit avec 3 cas d'utilisation. "Réaliser un test" a une relation d'extension avec "Envoyer un mail". Car ce cas d'utilisation ne sera utilisé que si le test ne se termine pas normalement.

4.4.4 L'enseignant

Le diagramme de cas d'utilisation contient 3 cas d'utilisation assez simples.

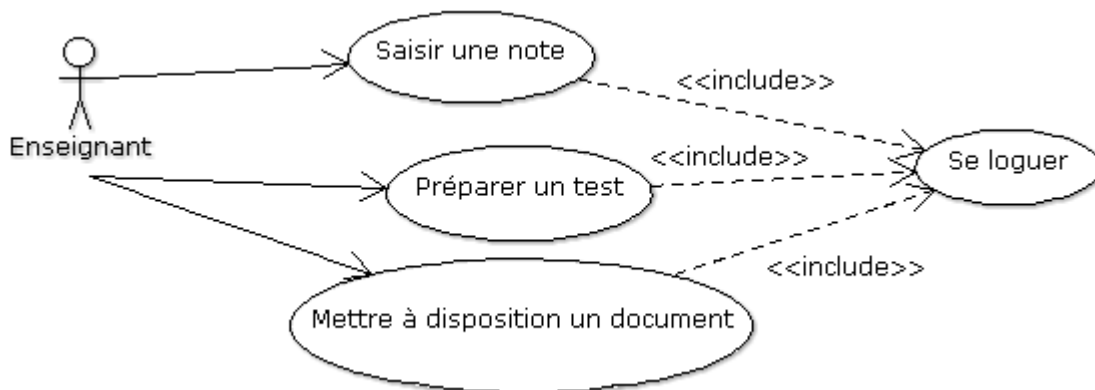


Figure 4.10 : Cas d'utilisation de l'enseignant

4.4.5 Diagramme complet

Ce qui donne le diagramme de cas d'utilisation complet présenté dans la figure 4.11 .

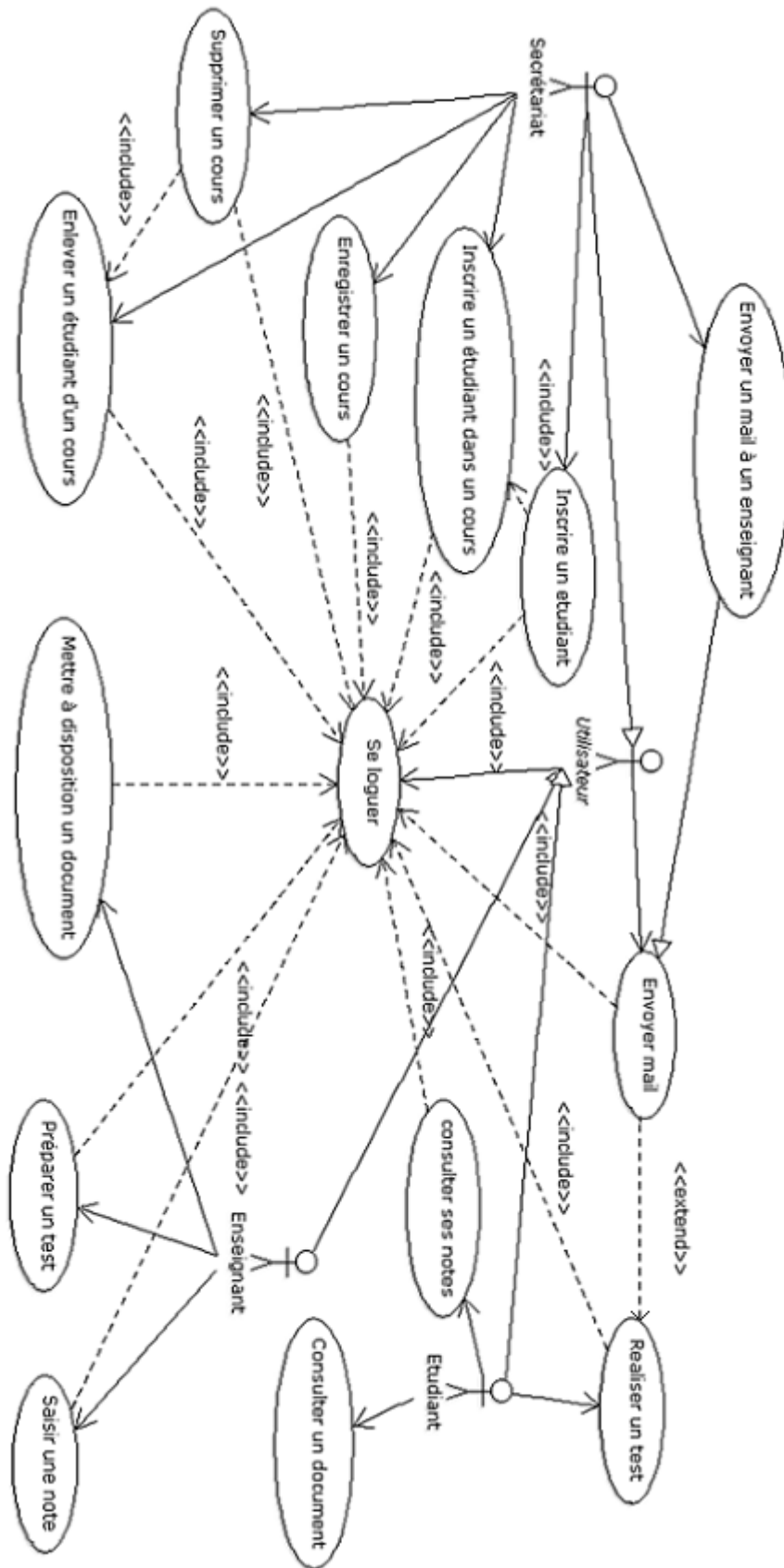


Figure 4.11 : Diagramme de cas d'utilisation complet

4.5 Le guichet automatique de banque

4.5.1 Question 1 : Utilisateurs du système

Nous distinguons ici 5 types d'utilisateurs du système. Les acteurs principaux :

- Porteur de carte non client,
- Porteur de carte client,
- Agent de maintenance.

Et les acteurs secondaires qui correspondent ici aux systèmes avec lesquels les utilisateurs principaux communiquent via le GAB.

- Le système d'authentification Visa,
- le système d'information de la banque.

4.5.2 Cas d'utilisation

Les cas d'utilisation du GAB sont les suivants :

- Retirer argent
 - avec une carte client
 - avec une carte non client
- Consulter compte
- Déposer argent
 - numéraire
 - chèque
- Réaliser la maintenance
 - rechargement en billets du distributeur
 - récupération des cartes avalées
 - récupération des chèques déposés

4.5.3 Diagramme de cas d'utilisation du GAB

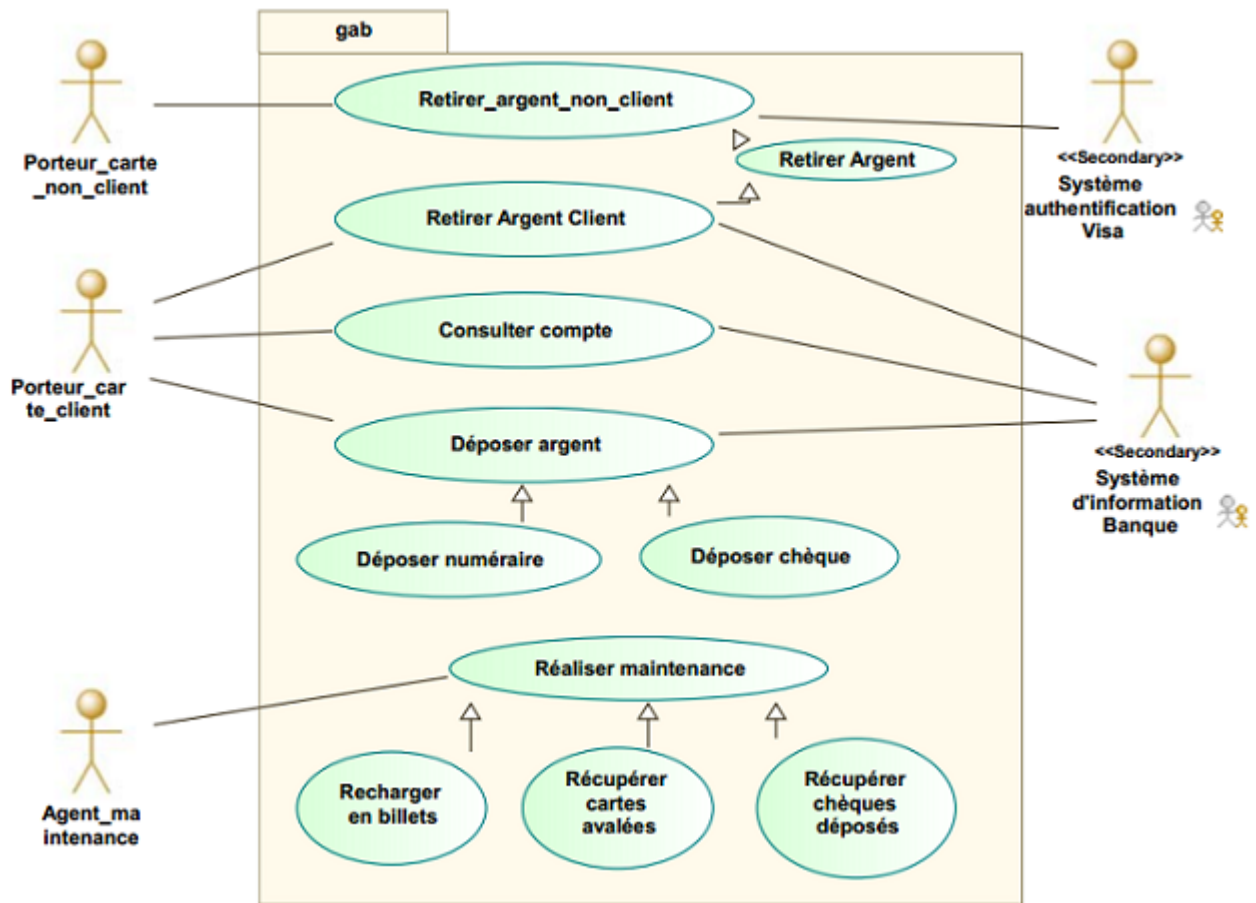


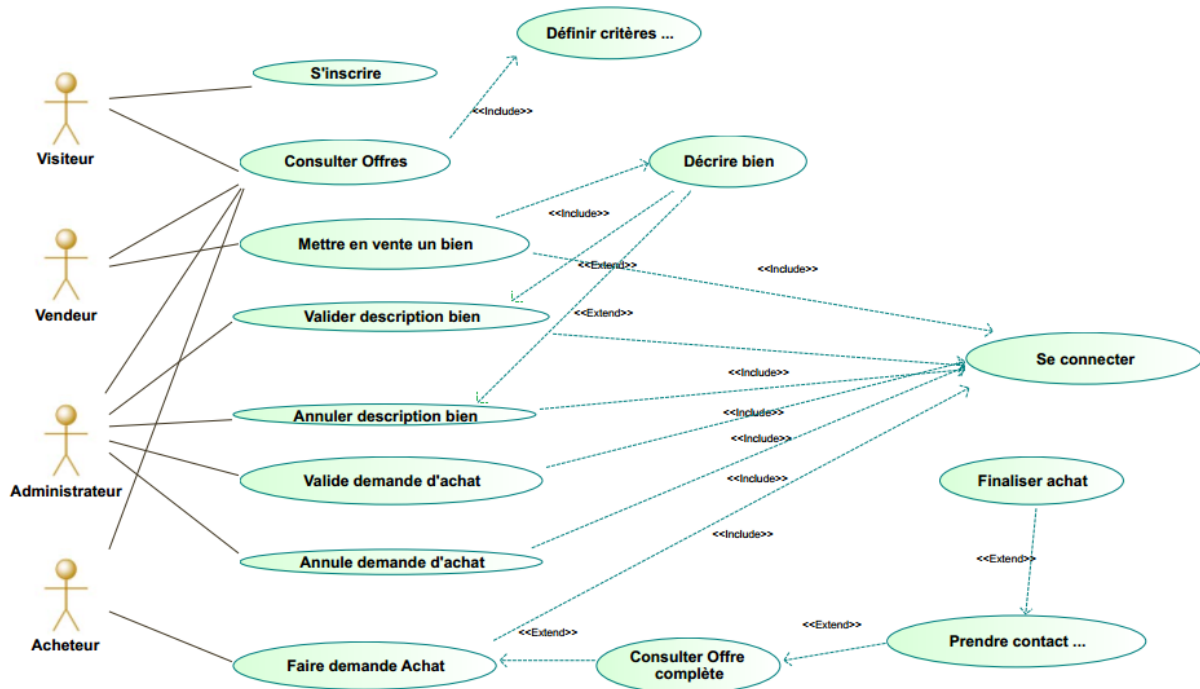
Figure 4.12 : Diagramme des cas d'utilisation du GAB

4.6 Gestion de biens immobiliers

Question 1 Quels sont les différents acteurs du système ? On distingue 4 types d'acteurs :

- Visiteur
- Vendeur
- Acheteur
- Administrateur

Question 2 Construire le diagramme de cas d'utilisation de ces différents acteurs



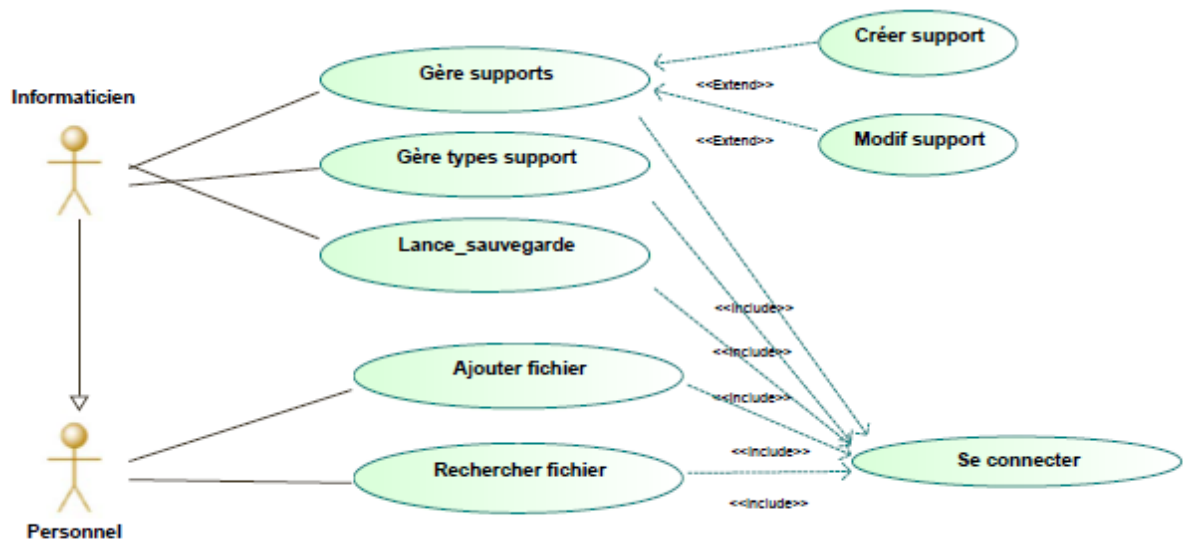
4.7 Sauvegarde de fichiers

4.7.1 Acteurs

On propose ici deux acteurs : informaticien et utilisateur.



4.7.2 DCU



Chapter 5

Diagramme de Séquences : Enoncés des exercices

5.1 La lampe de chevet

Considérons une lampe de chevet, les deux interactions liées au système que sont l'allumage de la lampe et son extinction.

5.1.1 Premier diagramme de séquences

Représenter sous la forme d'un diagramme de séquences les interactions entre la lampe et l'utilisateur pour l'allumage et l'extinction de la lampe. Ici nous avons deux participants :

- l'utilisateur qui est représenté par une instance de la classe `Utilisateur`
- la lampe, représentée par une instance de la classe `Lampe`.

5.1.2 Un diagramme de séquence avec d'autres participants

Prenons en compte maintenant deux autres participants à ces interactions : l'interrupteur et le réseau électrique. L'utilisateur n'interagit plus directement avec la lampe mais avec l'interrupteur qui lui-même interagit avec le réseau électrique qui pour finir interagit avec la lampe.

5.2 Retrait d'espèces

Un cas d'utilisation décrivant un retrait d'espèces propose les éléments suivants :

1. Le guichetier saisit le numéro de compte du client ;
2. L'application valide le compte auprès du système central ;
3. Le guichetier demande un retrait de 100 euros ;
4. Le système "guichet" interroge le système central pour s'assurer que le compte est suffisamment approvisionné ;
5. Le système central effectue le débit du compte ;
6. En retour, le système notifie au guichetier qu'il peut délivrer le montant demandé.

Donner le diagramme de séquences associé à cette description textuelle.

5.3 GAB : Guichet automatique de banque

Reprenons l'exemple vu dans les diagrammes de cas d'utilisation : le GAB. Créer un diagramme de séquences illustrant le scénario suivant :

1. Un client possédant une carte visa (non client de la banque) introduit sa carte bancaire dans le GAB.
2. Le GAB vérifie la conformité de la carte introduite.
3. Le GAB demande le code de la carte au client.
4. Le client saisit le bon code.
5. Le GAB vérifie que le code saisi est bel et bien correct.
6. Le GAB interroge le Système d'Autorisation Visa pour obtenir le solde maximum de retrait disponible.
7. Le GAB demande au client le montant à retirer.
8. Le client saisit le montant désiré.
9. Le GAB contrôle ce montant vis-à-vis du solde maximum de retrait disponible (le retrait est autorisé).
10. Le GAB demande au client s'il désire un ticket récapitulatif de la transaction.
11. Le client répond qu'il désire ce ticket.
12. Le GAB éjecte la carte bancaire du client.
13. Le client récupère sa carte bancaire.
14. Le GAB délivre les billets.
15. Le client récupère les billets.
16. Le GAB émet le ticket.
17. Le client récupère le ticket.
18. Le GAB enregistre la transaction dans sa base de données.

5.4 Gestion des cours en ligne

Reprenons le cas où un étudiant passe un test. Il doit bien sûr se loguer, une fois sur la page du test, l'étudiant choisit de le démarrer ou de le quitter. Si l'étudiant quitte le test alors qu'il n'est pas terminé, un e-mail est envoyé à l'intervenant automatiquement et le test n'est plus disponible pour lui. Si l'étudiant complète le test de manière normale, les notes sont automatiquement entrées dans la base de données. L'étudiant ne peut alors plus réaliser le test (le test n'est plus disponible pour lui).

5.5 Gestion de biens immobiliers

Reprenons le système de gestion de biens permettant à des personnes de proposer à la vente des biens ou d'en acheter.

Le site mis à disposition peut être consulté par tout visiteur, par contre pour pouvoir acheter ou mettre en vente un bien il faut s'inscrire. La description d'un bien est composée d'un type (Maison, Appartement), d'une localisation (quartier, ville, code postal), d'une surface, d'un nombre de pièces et d'un prix en euros.

Un administrateur peut valider la description d'un bien ou au contraire l'annuler ; il valide par ailleurs une demande d'achat ou l'annule. Lorsque qu'une demande d'achat est validée, le client demandeur est alors prévenu par une alerte sur son compte et peut alors accéder aux données complètes de l'offre (détail sur le vendeur : téléphone,) et ainsi prendre contact avec le vendeur

pour finaliser l'achat.

Une personne qui réalise une demande d'achat doit se connecter, visualiser l'ensemble des offres correspondant à des critères de choix (Type, localisation, prix,), en choisir une et faire une offre de prix. Le dépôt d'une offre de vente suppose une connexion, la description de l'offre (Type, localisation, prix, contact).

Modélisez la suite de traitements à réaliser pour qu'un client réalise l'achat d'un bien.

5.6 Sauvegarde de fichiers

Le logiciel de sauvegarde de l'entreprise "Dupont" stocke les informations dont il a besoin pour la gestion des sauvegardes dans une base de données "Sauve_Fiche". Cette base de données permet de savoir quels fichiers doivent être sauvegardés et quand ils ont été sauvegardés. On a besoin ici de connaître le nom des fichiers et leur chemin d'accès. Différents types de sauvegardes peuvent être réalisés. Lors du lancement de la sauvegarde on enregistre la date et l'heure de lancement de la sauvegarde son type, le support utilisé. On distingue 2 types de sauvegarde :

- Complète : on sauvegarde tous les fichiers ;
- Incrémentale : on ne sauvegarde que les fichiers qui ont été modifiés depuis la dernière sauvegarde. Il est donc nécessaire de conserver la date de la dernière modification de chaque fichier lors de l'exécution de la sauvegarde.

Une sauvegarde peut être effectuée sur différents types de supports : bande magnétique, disque dur, Mais l'intégralité d'une sauvegarde est réalisée sur un unique type de support. Une première phase dite de préparation permet de réaliser les choix concernant la sauvegarde.

Lors de la sauvegarde, l'application récupère les dates et les heures de début et de fin de sauvegarde ainsi que la liste des fichiers à sauvegarder et stocke chacun des fichiers stocke sur un ou plusieurs supports.

Chaque support de sauvegarde est identifié par un numéro. La sauvegarde est conservée durant 1 an dans un local dédié aux sauvegardes. Passé ce délai, le support peut être réutilisé pour d'autres sauvegardes.

Dans l'entreprise, seul l'informaticien est autorisé à gérer les sauvegardes. Il peut ainsi décrire de nouveaux types de supports, créer de nouveaux supports en supprimer. Il lance les sauvegardes et gère les supports de sauvegarde.

Le reste du personnel peut ajouter dans la liste des fichiers à sauvegarder ses propres fichiers et faire des recherches de fichiers dans les sauvegardes.

Quel que soit le traitement à réaliser l'utilisateur doit s'identifier (login + mot de passe) pour avoir accès à l'application.

Modélisez la suite de traitements à réaliser pour qu'un informaticien lance et finalise une sauvegarde de fichiers de type incrémentale.

Chapter 6

Diagramme de Séquences : Corrigés des exercices

6.1 La lampe de chevet

6.1.1 Premier diagramme de séquences

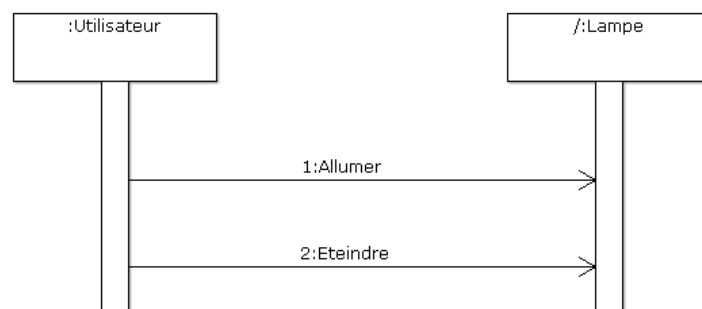


Figure 6.1 : Premier diagramme de séquence Lampe

6.1.2 Un diagramme de séquence avec d'autres participants

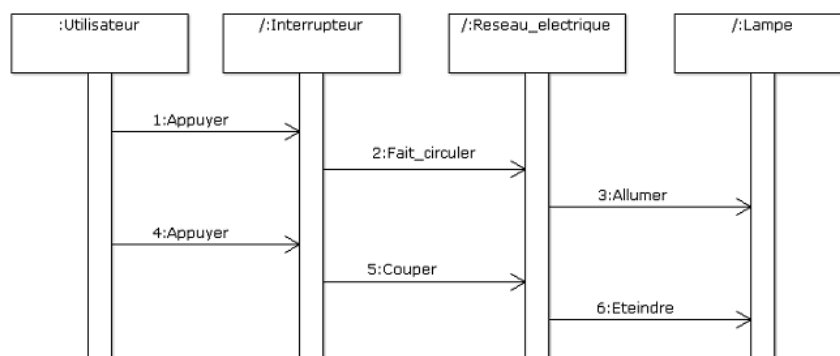
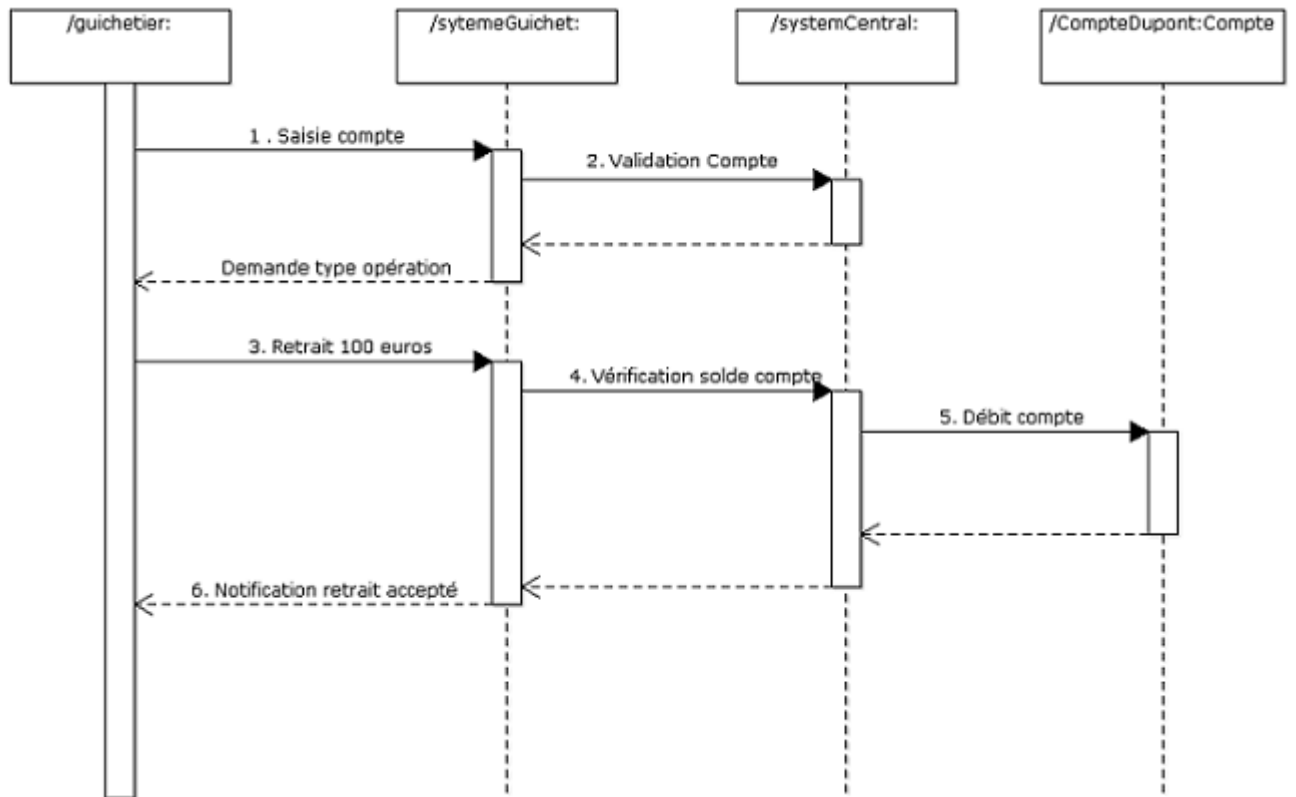
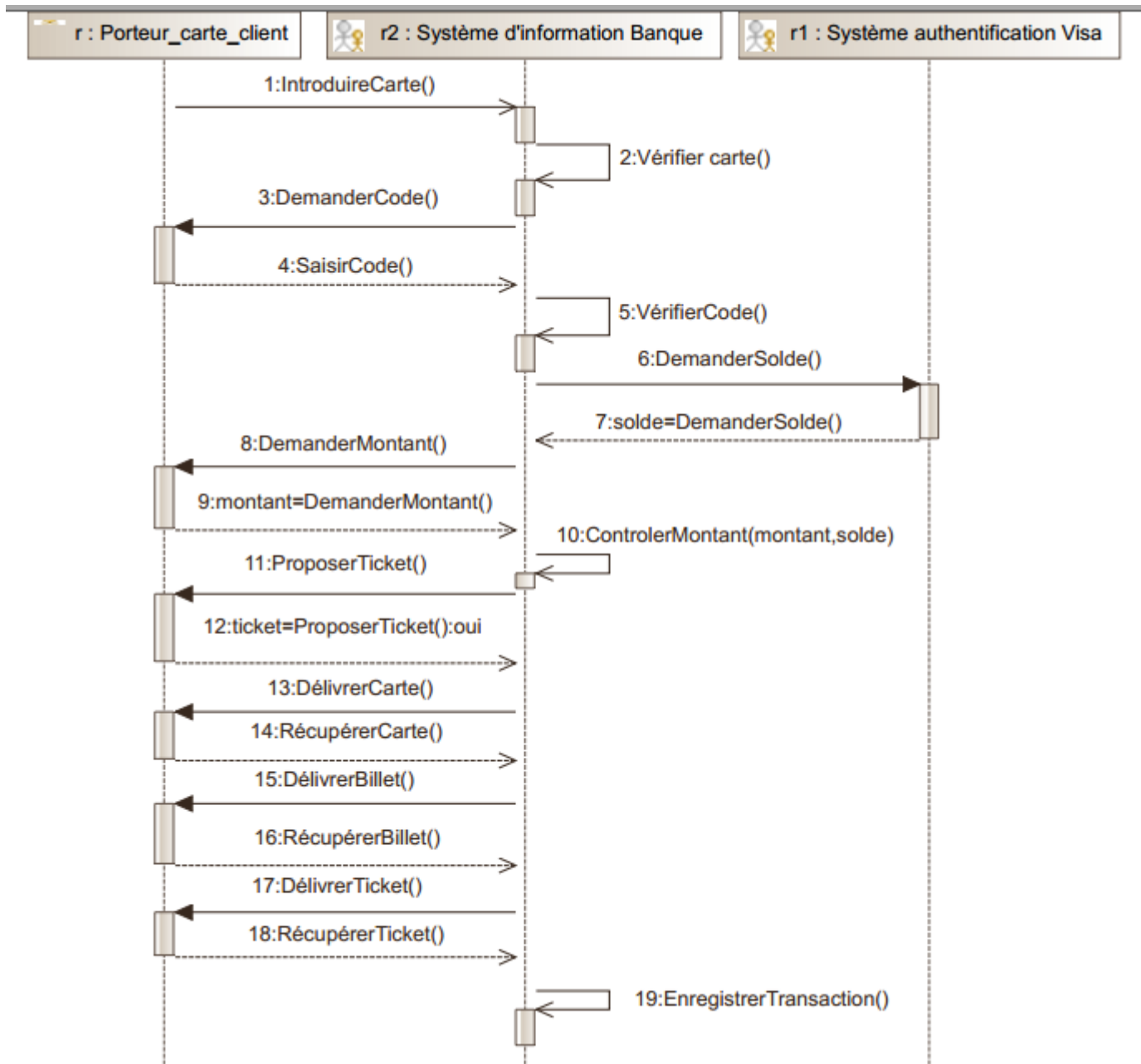


Figure 6.2 : DCU Lampe avec l'interrupteur et le réseau électrique

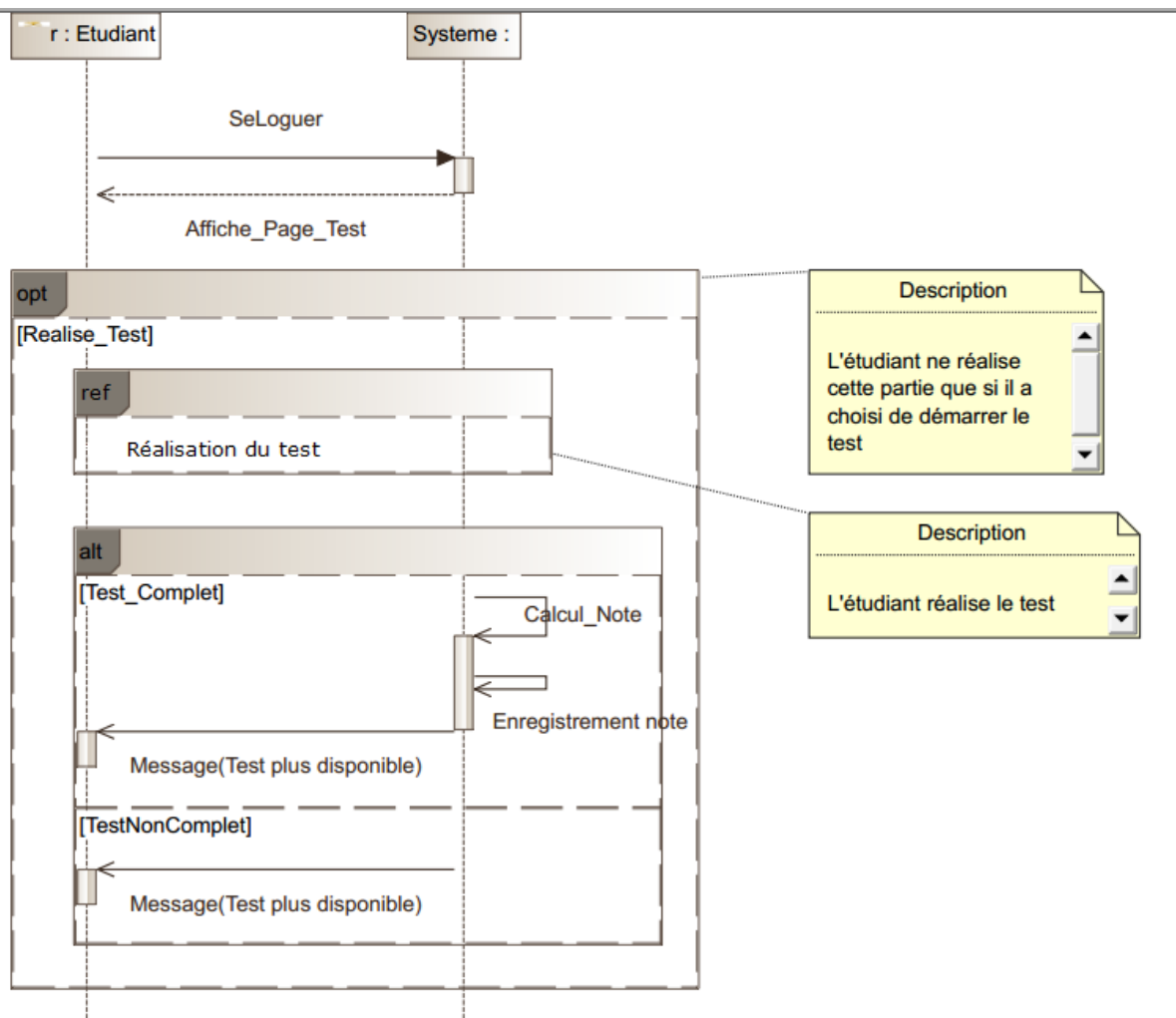
6.2 Retrait d'espèces



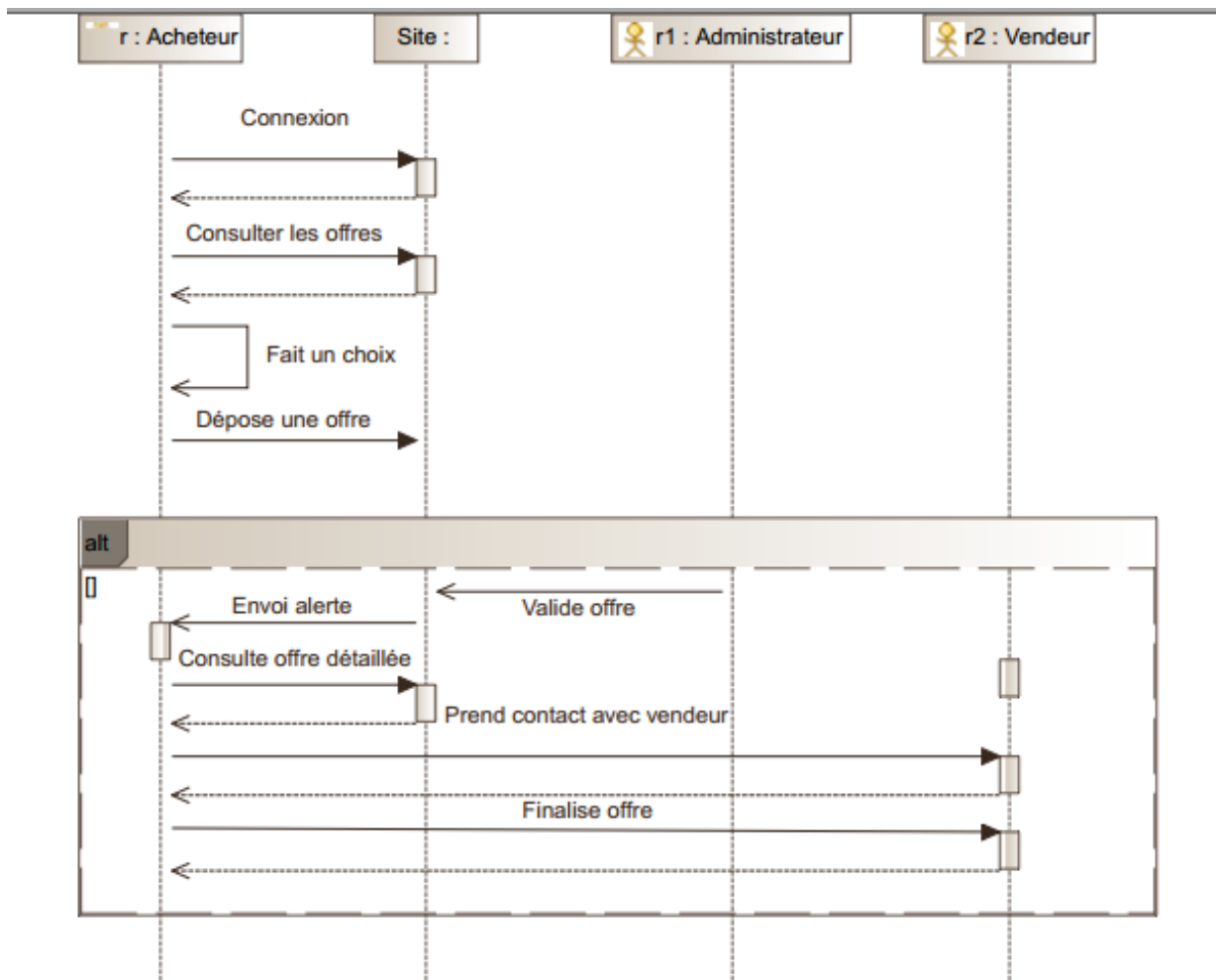
6.3 GAB : Guichet automatique de banque



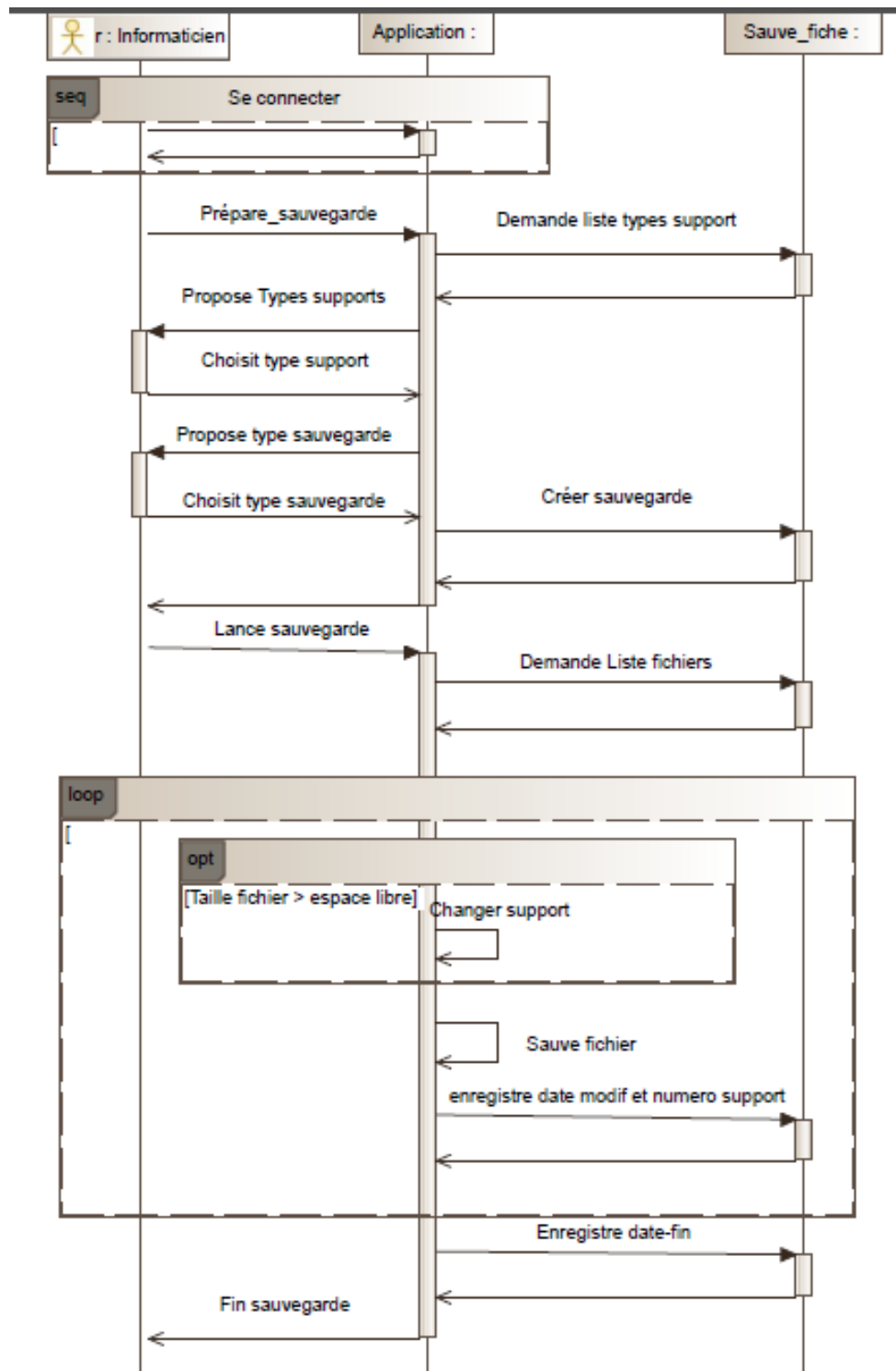
6.4 Gestion des cours en ligne



6.5 Gestion de biens immobiliers



6.6 Sauvegarde de fichiers



Chapter 7

Sujets d'examen

7.1 UML : Le coach sportif

Un coach sportif souhaite disposer d'une application lui permettant de gérer ses clients, leurs abonnements et leurs séances.

Il souhaite ainsi disposer d'un système qui envoie au début de chaque semaine le planning des séances (jour, heure et lieu) à chaque client (sms et mail), la veille de chaque séance il envoie par sms un rappel de la séance à chaque client. Chaque mois le coach génère une facture pour chaque client (fichier pdf envoyé par mail) après validation des données mensuelles. Sur cette facture sont rappelés les dates des séances effectuées pendant le mois (date et heure), l'état de l'abonnement du client (nombre de séances réalisées depuis le début de l'abonnement, nombre de séances restantes, date de début et date de fin) et bien évidemment des informations sur le client telles que son nom et prénom, son adresse, son téléphone fixe et portable ainsi que son mail.

Plusieurs clients peuvent prendre un abonnement ensemble. Chacun a alors une quote part à cet abonnement. On peut avoir ainsi un abonnement de 50 séances dont la durée de validité est de 6 mois avec 3 clients ayant pour chacun pour quote part : 50, 25 et 25. La quote-part est un nombre qui exprime un pourcentage. Lorsque plusieurs clients partagent un abonnement, chacun reçoit sa propre facture.

L'abonnement correspond à un forfait, et a une date de début. Pour chaque abonnement on peut proposer différents lieux pour les séances et un lieu par défaut. Les autres lieux ne seront utilisés que pour des séances spécifiques suite à une modification de séance. Lors du démarrage de l'abonnement, des créneaux horaires (jour de la semaine et heure) sont fixés.

Le coach propose différents forfaits. Un forfait a un nombre total de séances, une durée de validité du forfait en mois et un tarif global.

Pour chaque client, le coach doit disposer d'un champ lui permettant de noter des remarques ("Problème de dos", "Préparation à un marathon", ...).

Un client doit pouvoir se connecter sur l'application (avec un login et un mot de passe), visualiser ses séances pour le mois en cours, visualiser l'état de son forfait. Il peut aussi demander l'annulation d'une séance (seulement si elle a lieu plus de 36 heures après la demande) et la modification d'une séance (lieu et/ou horaire).

Le coach doit pouvoir accéder, après connexion, à son planning pour la semaine, accepter une demande de modification de séance (changement d'horaire, de lieu), valider une demande d'annulation. Il doit aussi pouvoir accéder aux informations de chaque client ou de chaque abonnement, créer de nouveaux forfaits, de nouveaux clients et abonnements.

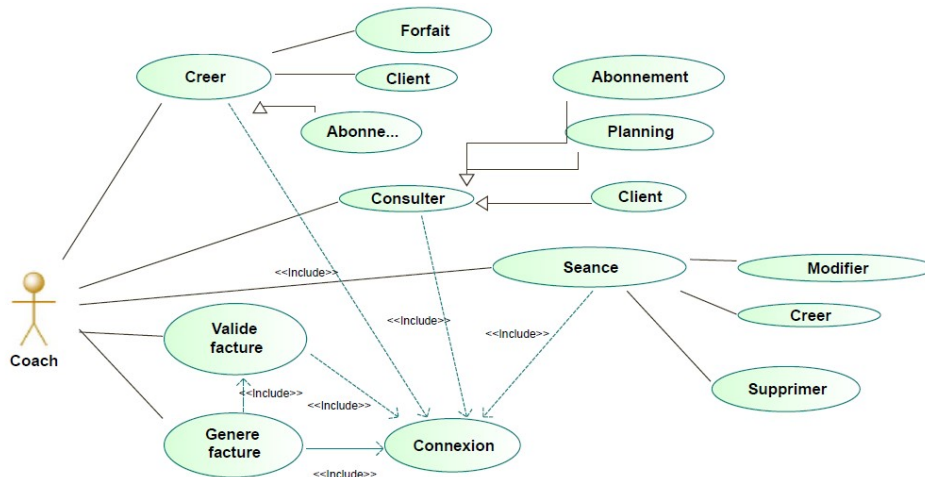
Une demande d'annulation réalisée plus de 36 heures à l'avance est automatiquement acceptée par le système qui traite alors l'annulation (suppression de la séance) et envoie un message d'annulation au client et un autre au coach. Dans le cas où la demande est effectuée moins de

36h avant la séance, le système demande au coach son accord pour l'annulation. Si le coach refuse la demande d'annulation de séance, la séance n'aura pas lieu mais sera due. Il faudra qu'elle apparaisse dans la facture avec le libellé "Non réalisée mais due". Dans le cas où le coach accepte la demande d'annulation de séance le traitement est le même que celui d'une demande d'annulation réalisée plus de 36h avant.

7.1.1 Diagramme de cas d'utilisation

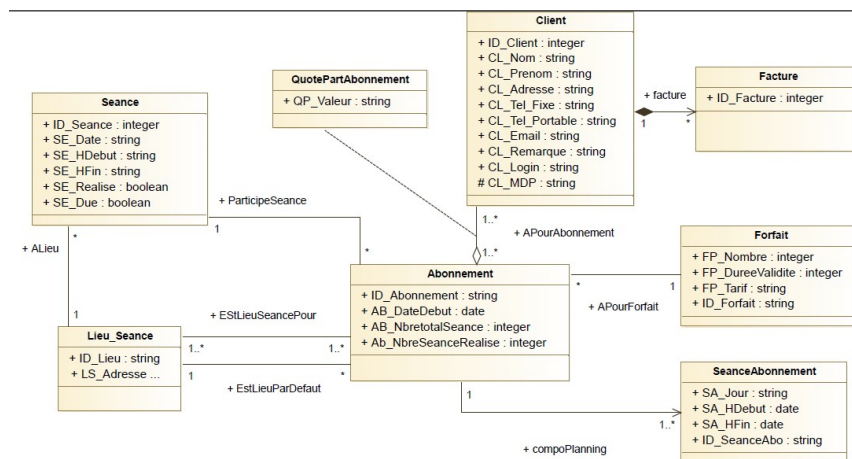
Quels sont les différents acteurs du système ?

Diagramme de cas d'utilisation de l'acteur Coach



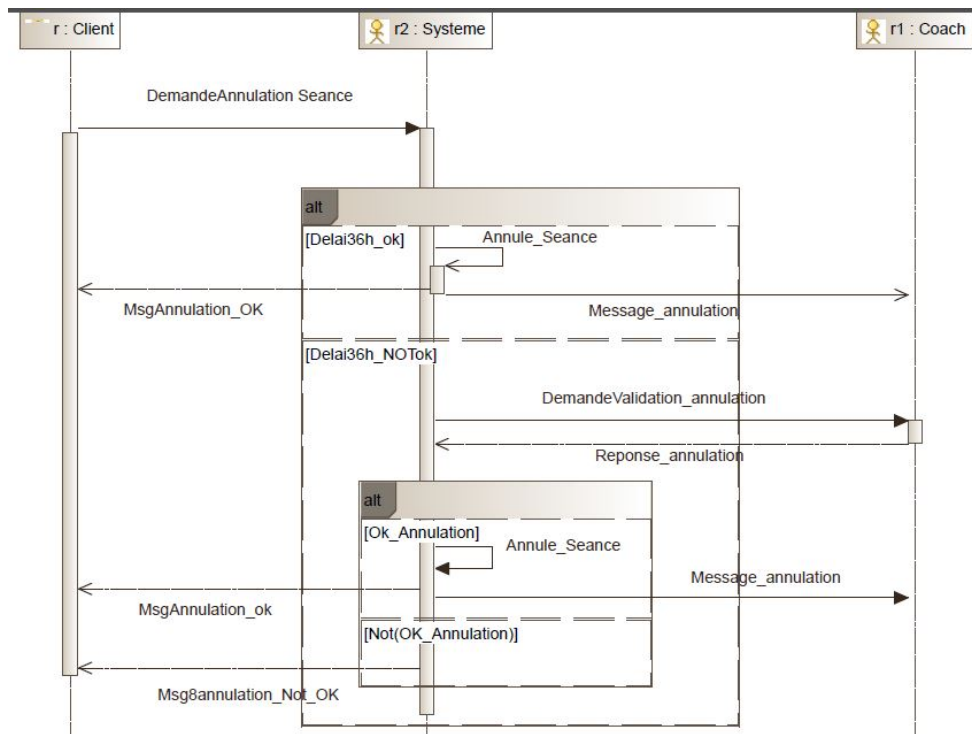
7.1.2 Diagramme de classe

Proposez un diagramme de classe décrivant les données du système "Coach".



7.1.3 Diagramme de séquence

Modélisez la suite de traitements à réaliser à partir d'une demande d'un client jusqu'à l'annulation d'une séance.

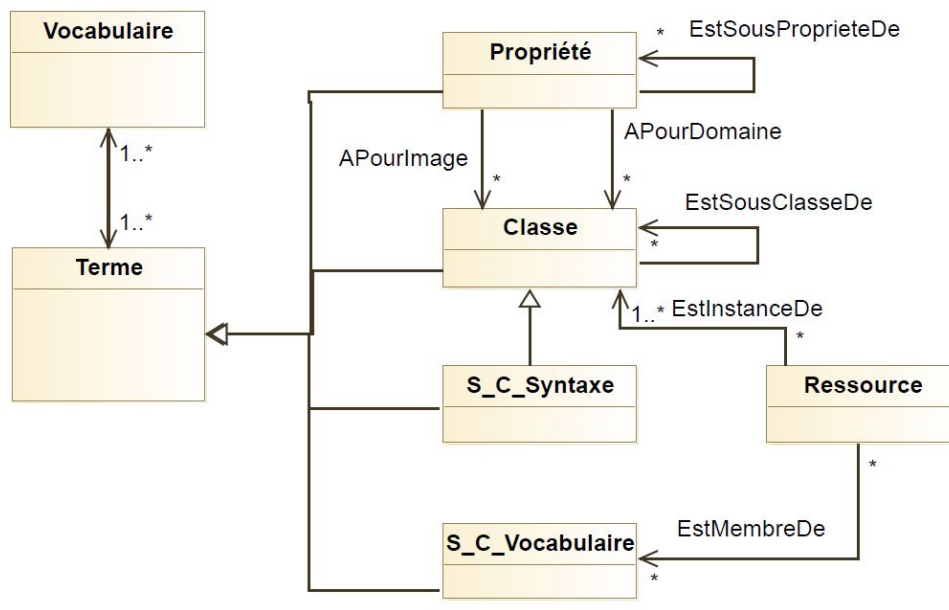


7.2 Les vocabulaires du Dublin Core Metadata Initiative

L'objectif de cet exercice est de spécifier sous la forme d'un diagramme de classe, le modèle abstrait des vocabulaires utilisés dans les descriptions de métadonnées du Dublin Core.

- Un vocabulaire est un ensemble constitué d'un ou plusieurs termes. Chaque terme est membre d'un ou plusieurs vocabulaires.
- Un terme est une propriété, une classe, un schéma de codage de vocabulaire ou un schéma de codage de syntaxe.
- Une propriété peut avoir plusieurs domaines qui sont des classes.
- Une propriété peut avoir une ou plusieurs images qui sont elles-mêmes des classes.
- Une propriété peut être sous-propriété d'une ou plusieurs autres propriétés.
- Une ressource peut être membre d'un ou plusieurs schémas de codage de vocabulaire. C'est une instance d'une ou plusieurs classes.
- Une classe peut être sous-classe d'une ou plusieurs autres classes.
- Un schéma de codage de syntaxe est une classe.

Attention ce schéma ne correspond pas à ce que nous avons vu jusqu'à maintenant. Ici le détail des classes n'est pas décrit.

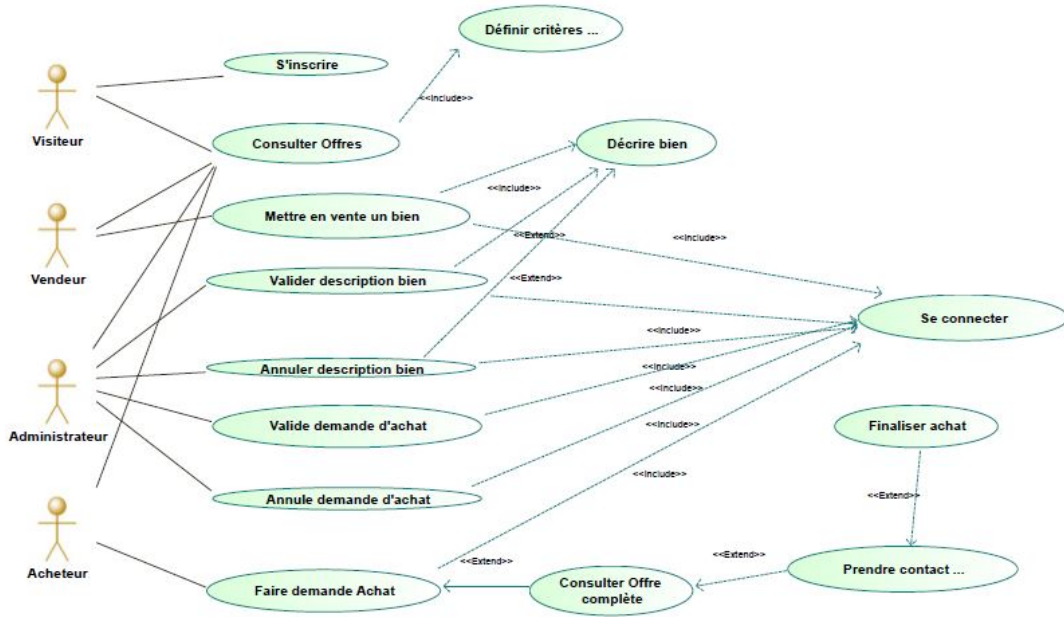


7.3 Gestion de biens immobiliers

On souhaite créer un système de gestion de biens permettant à des personnes de proposer à la vente des biens ou d'en acheter. Le site mis à disposition peut être consulté par tout visiteur, par contre pour pouvoir acheter ou mettre en vente un bien il faut s'inscrire. La description d'un bien est composée d'un type (Maison, Appartement), d'une localisation (quartier, ville, code postal), d'une surface, d'un nombre de pièces et d'un prix en euros. Un administrateur peut valider la description d'un bien ou au contraire l'annuler ; il valide par ailleurs une demande d'achat ou l'annule. Lorsque qu'une demande d'achat est validée, le client demandeur est alors prévenu par une alerte sur son compte et peut alors accéder aux données complètes de l'offre (détail sur le vendeur : téléphone,) et ainsi prendre contact avec le vendeur pour finaliser l'achat. Une personne qui réalise une demande d'achat doit se connecter, visualiser l'ensemble des offres correspondant à des critères de choix (Type, localisation, prix,), en choisir une et faire une offre de prix. Le dépôt d'une offre de vente suppose une connexion, la description de l'offre (Type, localisation, prix, contact)

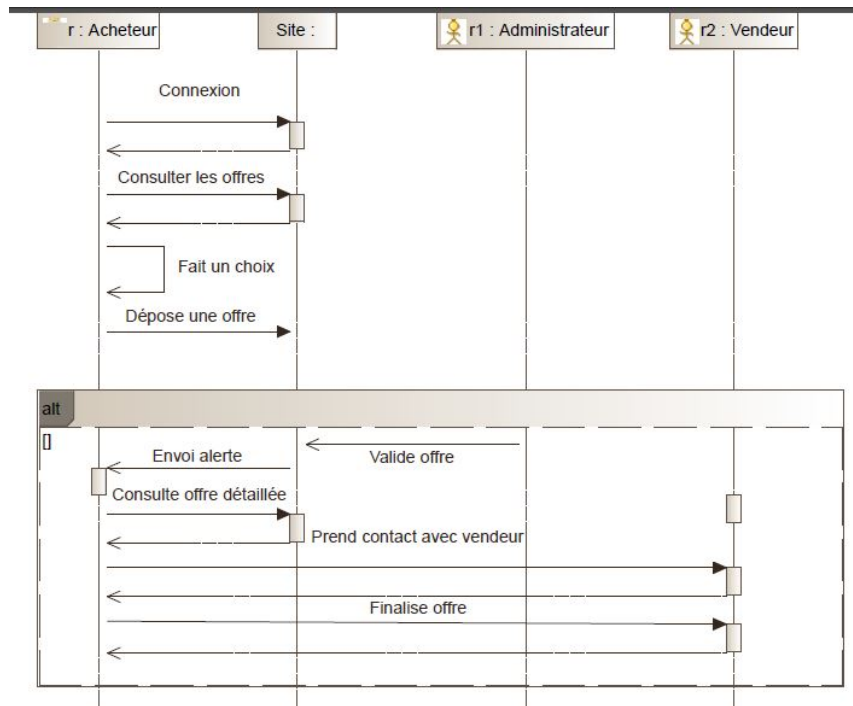
7.3.1 Diagramme de cas d'utilisation

Construire le diagramme de cas d'utilisation des différents acteurs.



7.3.2 Diagramme de séquence

Modélisez la suite de traitements à réaliser pour qu'un client réalise l'achat d'un bien.



7.4 Le code de la route

Une auto-école souhaite construire une application pour gérer le suivi de ses clients aux séances et aux examens du code de la route. Lorsqu'un client vient s'inscrire à l'auto-école on lui attribue automatiquement un numéro unique et on lui demande ses nom, prénom, adresse et date de

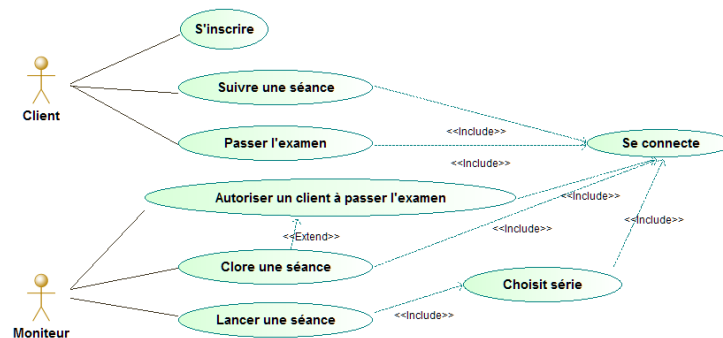
naissance. Une fois inscrit le client peut assister à autant de séances de code qu'il le souhaite. Pour chaque séance, qui a lieu à une date et une heure données, après s'être connecté, le moniteur d'auto-école affiche l'ensemble des DVD en choisit un et ensuite choisit une série de questions sur le DVD, il lance ensuite la séance une fois que les clients sont connectés. Une fois que les clients présents ont fini de répondre le moniteur clôt la séance. Chaque DVD est identifié par un numéro, a un nom d'éditeur et est composé de 6 séries, numérotées de 1 à 6. Chaque série est composée de 40 questions choisies dans un pool de questions. Chaque question est identifiée par un numéro et a un libellé (texte de la question), une réponse, un niveau de difficulté et un thème. Une même question peut apparaître dans plusieurs séries avec un numéro d'ordre pour chaque série. Une question appartient obligatoirement à au moins une série. On ne traite pas ici le détail des réponses. Lorsqu'un client assiste à une séance, après s'être connecté à l'application "CodeRoute", il répond à chacune des 40 questions posées et à la clôture de la séance il obtient son nombre de fautes (compris entre 0 et 40) une fois que l'application "CodeRoute" a calculé le nombre de fautes, elle l'affiche à chaque client. Lorsqu'un client a obtenu, au cours de ses quatre dernières séances un nombre de fautes inférieur ou égal à 5, le système prévient le moniteur d'auto-école qui peut alors autoriser le client à passer l'examen du code de la route à une date donnée (un seul examen pour une date donnée). La liste des sessions d'examen est affichée par l'application.

Les élèves ayant obtenu plus de 5 fautes à l'examen sont recalés et doivent assister de nouveau à des séances de code avant de pouvoir se représenter à l'examen. Le nombre de fautes obtenu à l'examen par chaque client est conservé.

7.4.1 Diagramme de cas d'utilisation

Quels sont les différents acteurs du système ?

Construire le diagramme de cas d'utilisation général de l'ensemble des acteurs

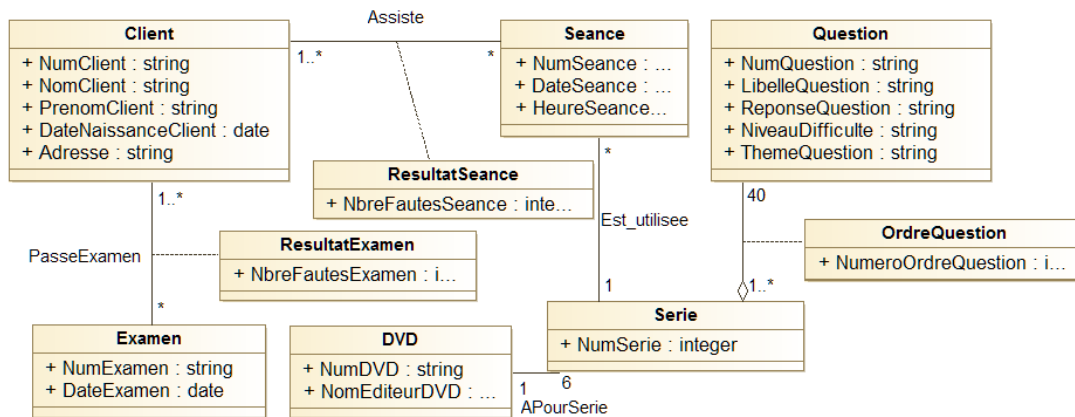


7.4.2 Diagramme de classe

Proposez un diagramme de classe décrivant les données du système "Code de la route".

7.4.3 Diagramme de séquence

Modélisez la suite de traitements qui sont réalisés lors d'une séance de code.



7.5 Sauvegarde de fichiers

Le logiciel de sauvegarde de l'entreprise "Dupont" stocke les informations dont il a besoin pour la gestion des sauvegardes dans une base de données "Sauve_Fiche". Cette base de données permet de savoir quels fichiers doivent être sauvegardés et quand ils ont été sauvegardés. On a besoin ici de connaître le nom des fichiers et leur chemin d'accès.

Différents types de sauvegardes peuvent être réalisés. Lors du lancement de la sauvegarde on enregistre la date et l'heure de lancement de la sauvegarde son type, le support utilisé.

On distingue 2 types de sauvegarde :

- Complète : on sauvegarde tous les fichiers ;
- Incrémentale : on ne sauvegarde que les fichiers qui ont été modifiés depuis la dernière sauvegarde. Il est donc nécessaire de conserver la date de la dernière modification de chaque fichier lors de l'exécution de la sauvegarde.

Une sauvegarde peut être effectuée sur différents types de supports : bande magnétique, disque dur, Mais l'intégralité d'une sauvegarde est réalisée sur un unique type de support. Une première phase dite de préparation permet de réaliser les choix concernant la sauvegarde.

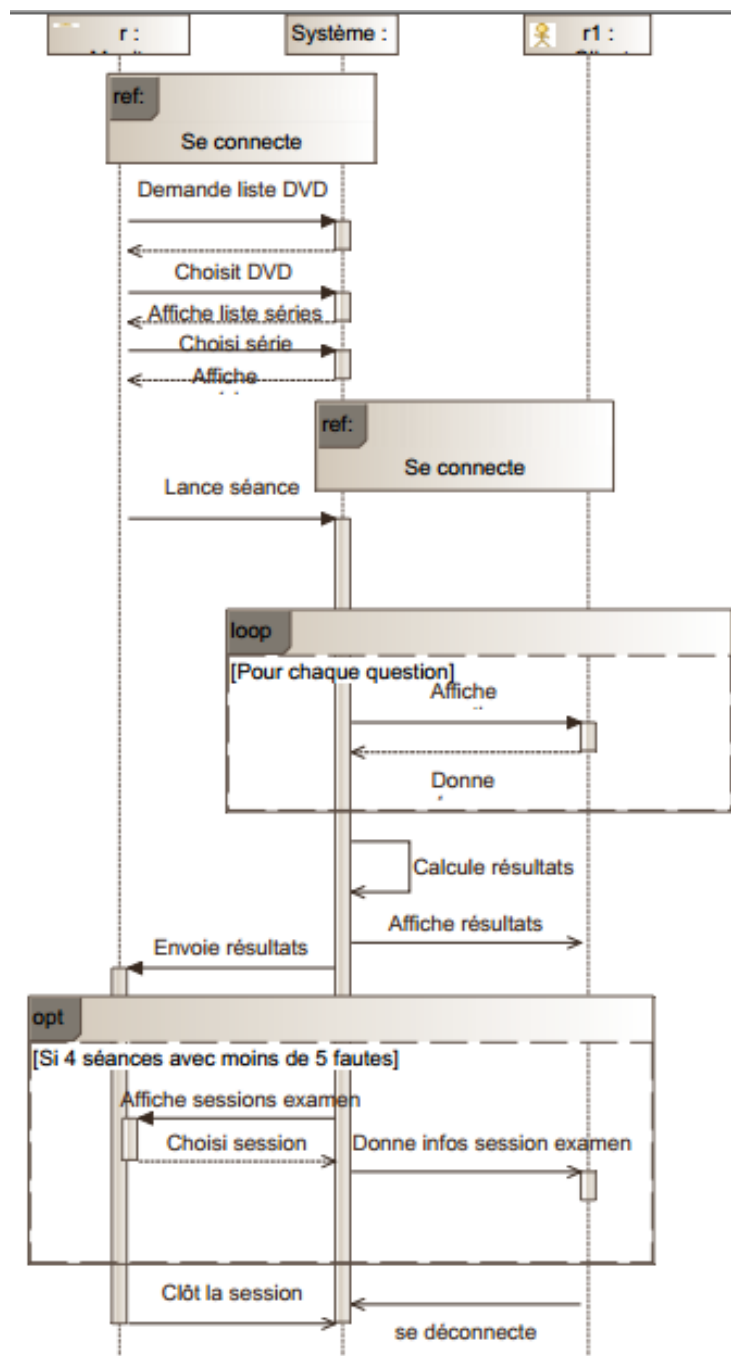
Lors de la sauvegarde, l'application récupère la date et l'heure de début de sauvegarde ainsi que la liste des fichiers à sauvegarder et stocke chacun des fichiers stocke sur un support. Si un seul support n'est pas suffisant pour sauvegarder tous les fichiers, un support "suivant" est utilisé pour la sauvegarde et éventuellement d'autres suivants. Chaque support de sauvegarde est identifié par un numéro. La sauvegarde est conservée durant 1 an dans un local dédié aux sauvegardes. Passé ce délai, le support peut être réutilisé pour d'autres sauvegardes.

A la fin de la sauvegarde on enregistre la date et l'heure.

Dans l'entreprise, seul l'informaticien est autorisé à gérer les sauvegardes. Il peut ainsi décrire de nouveaux types de supports, créer de nouveaux supports en supprimer. Il lance les sauvegardes et gère les supports de sauvegarde. Le reste du personnel peut ajouter dans la liste des fichiers à sauvegarder ses propres fichiers et faire des recherches de fichiers dans les sauvegardes. Quel que soit le traitement à réaliser l'utilisateur doit s'identifier (login + mot de passe) pour avoir accès à l'application.

7.5.1 Diagramme de classe

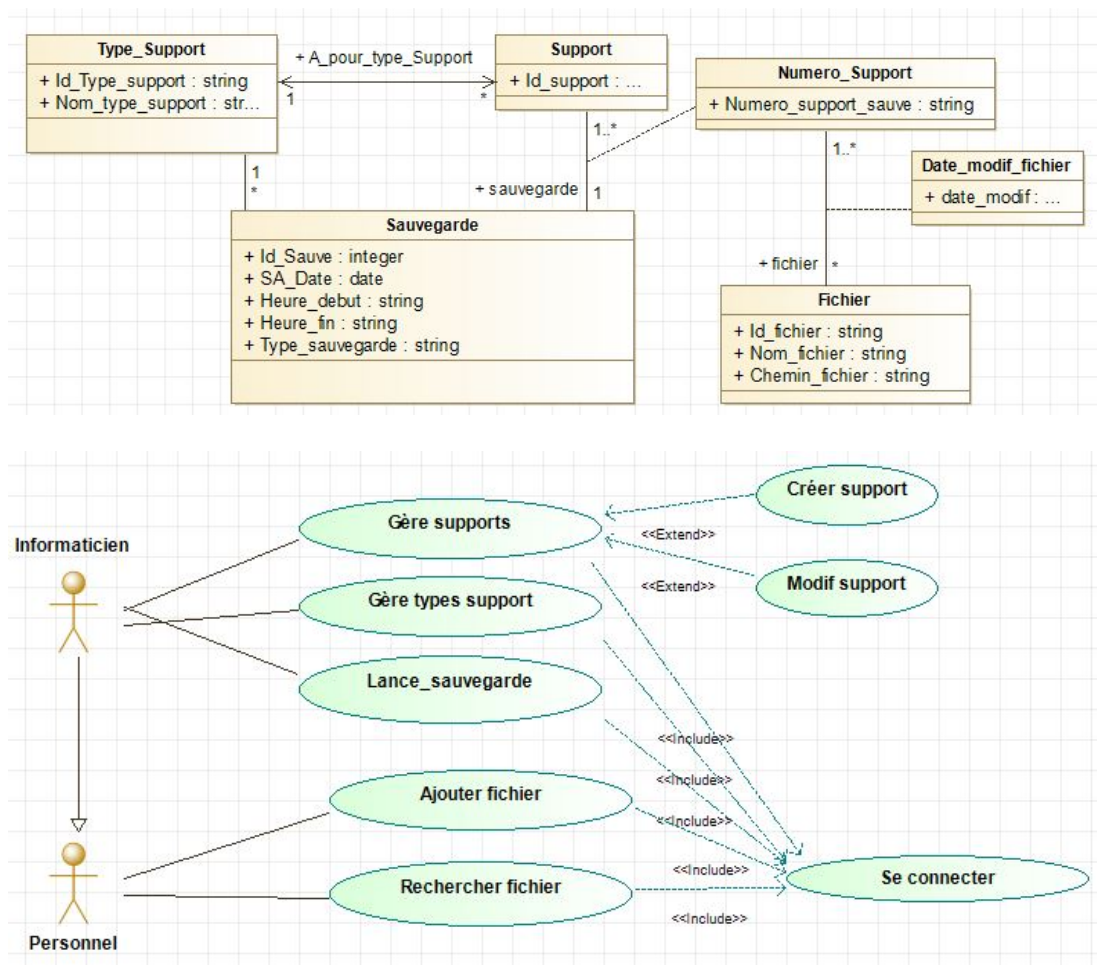
Créer un diagramme de classe permettant de décrire les données de la base de données "Sauve_Fiche" qui permet d'enregistrer les informations concernant toutes les sauvegardes réalisées.



7.5.2 Diagramme de cas d'utilisation

7.5.3 Diagramme de séquence

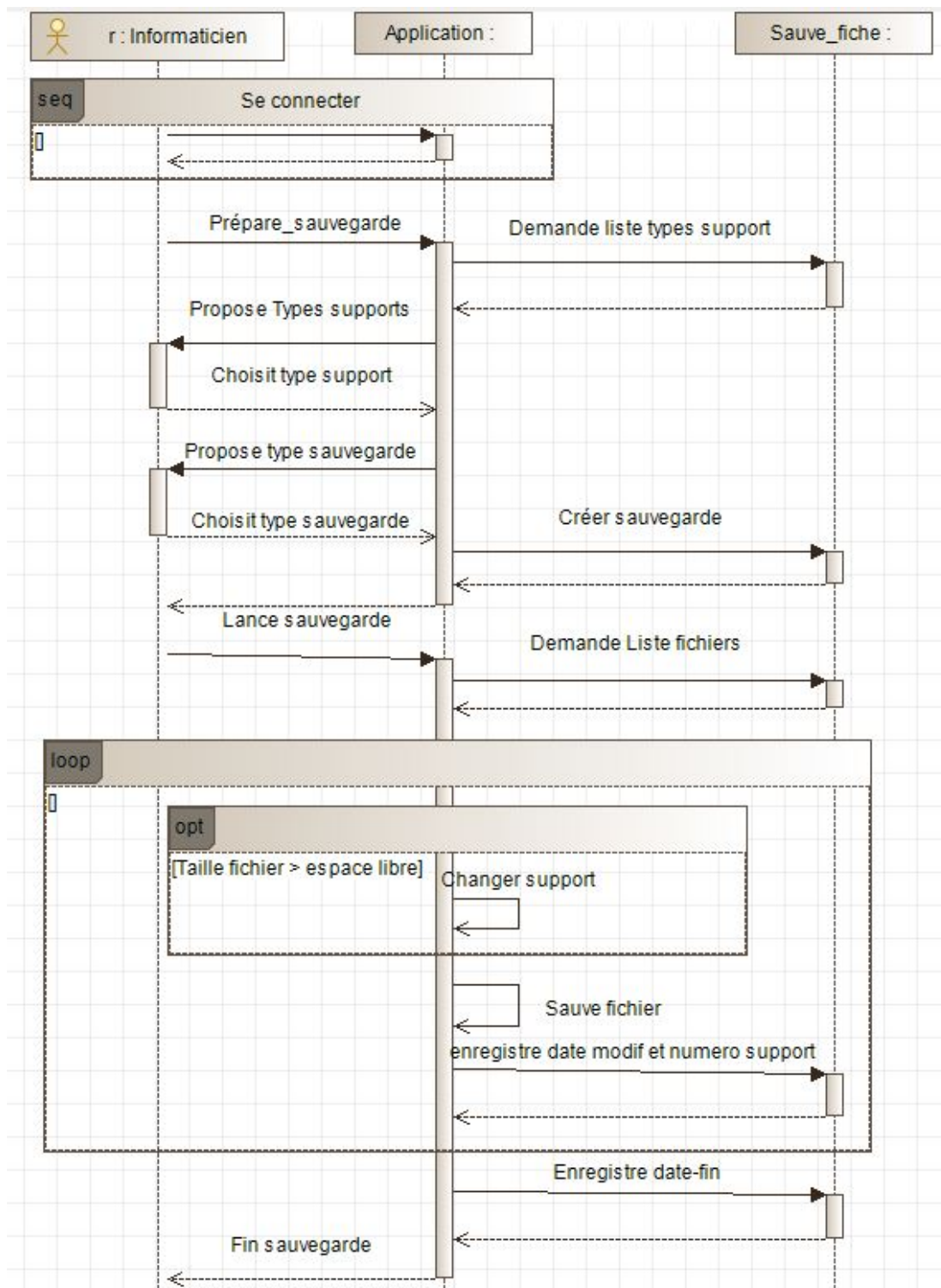
Modélisez la suite de traitements à réaliser pour qu'un informaticien lance et finalise une sauvegarde de fichiers de type incrémentale.



7.6 Bibliothèque

On souhaite gérer les emprunts d'ouvrages d'une bibliothèque. Le système à concevoir doit assister le bibliothécaire dans sa tâche.

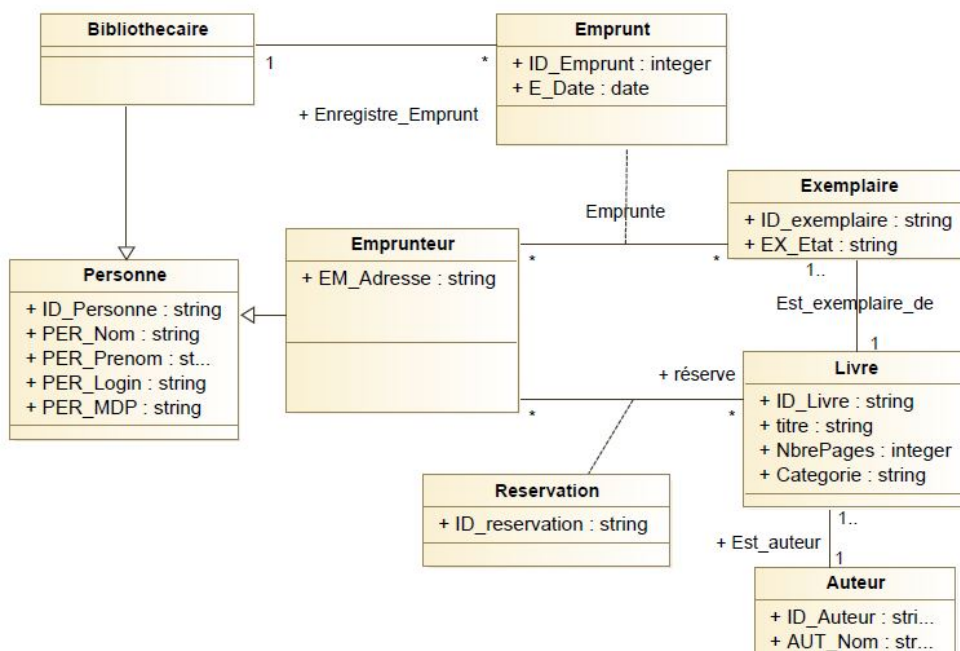
1. Le bibliothécaire est un employé de la bibliothèque. Le système enregistre son identifiant, son nom et son prénom, un login et un mot de passe.
2. Le bibliothécaire gère les emprunteurs. Il peut en ajouter, modifier ou supprimer. Un emprunteur est décrit par un code, un nom, un prénom, un login, un mot de passe, une adresse et un mail.
3. Le bibliothécaire prête des exemplaires de livres à des emprunteurs. Il enregistre alors la date de l'emprunt, le code de l'emprunteur, le code de l'exemplaire du livre emprunté. Un emprunt dure 2 semaines à partir de la date d'emprunt.
4. Le bibliothécaire doit gérer l'achat de nouveaux livres. Un livre outre son code est décrit par un titre, un auteur, un nombre de pages et une catégorie.
5. Un exemplaire d'un livre peut être supprimé lors qu'il est en mauvais état. Et un livre peut être supprimé s'il n'est plus demandé.
6. Un emprunteur peut réserver un livre qui est indisponible (tous les exemplaires du livre sont indisponibles).
7. Lorsqu'un livre devient disponible (un exemplaire est rendu), l'emprunteur qui l'avait réservé doit être averti. La réservation est annulée quand le livre est emprunté.



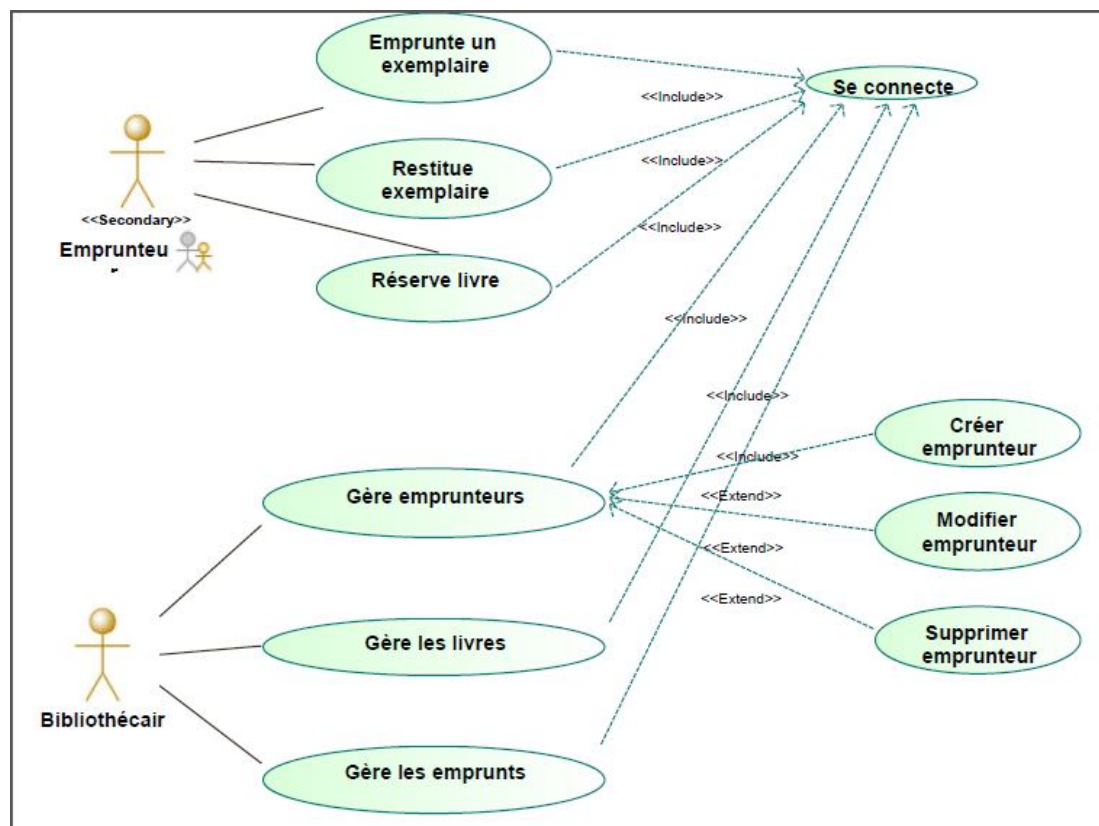
8. Le bibliothécaire doit pouvoir réaliser la mise à jour (ajout, suppression et modification) des informations relatives aux exemplaires et aux livres.
9. Il doit pouvoir aussi mettre à jour des informations relatives aux prêts et réservations.
10. Le bibliothécaire peut obtenir la liste des livres empruntés dont la date de retour est dépassée.

7.6.1 Diagramme de classe

Créez un diagramme de classe permettant de décrire les données de la base de données "Bibliothèque" qui permet d'enregistrer les informations concernant la gestion des emprunts de la bibliothèque.



7.6.2 Diagramme de cas d'utilisation



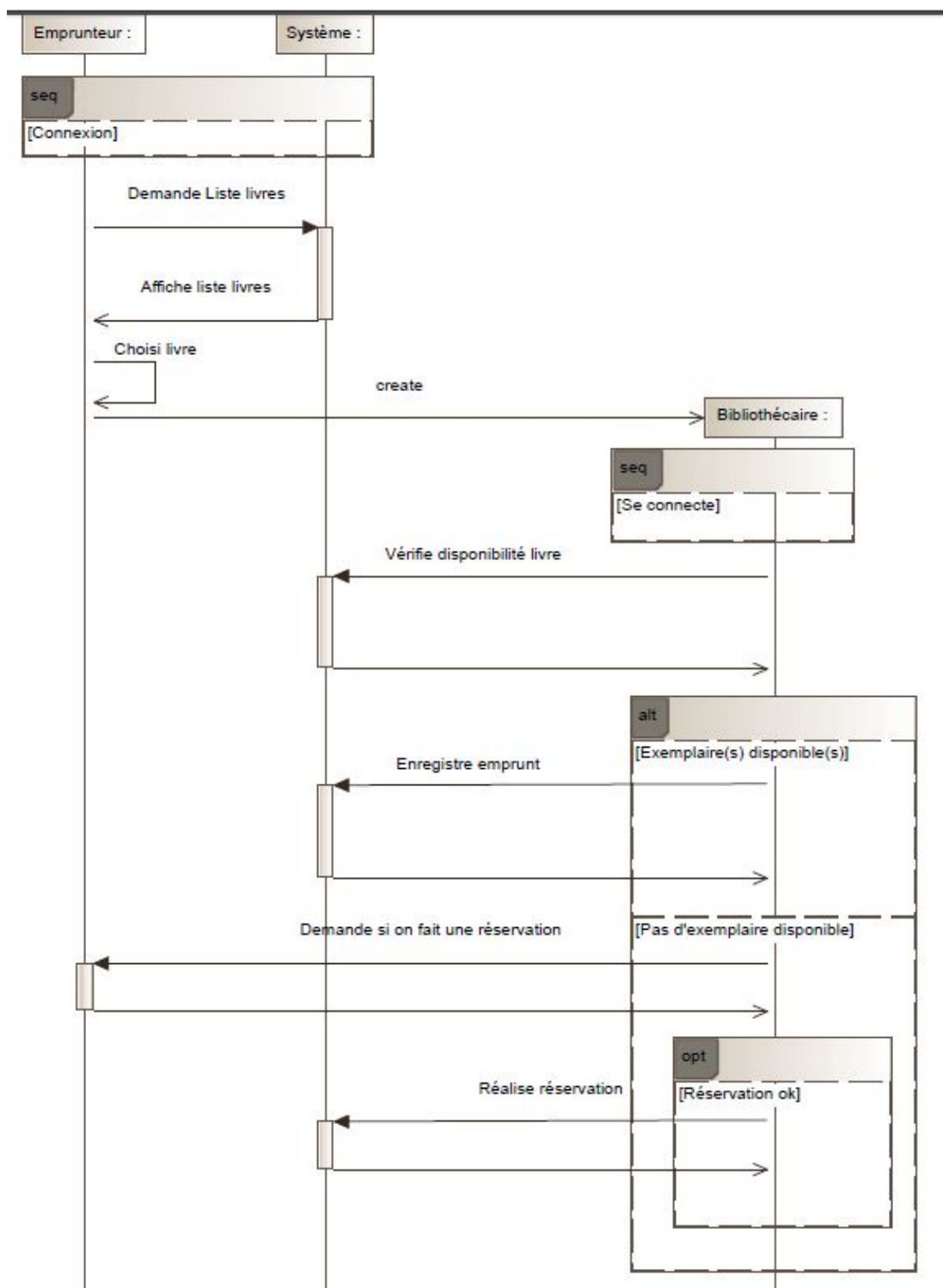
7.6.3 Diagramme de séquence

Modélisez la suite complète de traitements à réaliser pour emprunter un livre. Tous les cas sont à envisager (livre disponible ou pas). L'emprunteur se connecte, choisi son livre, le demande au bibliothécaire, celui-ci vérifie si un exemplaire est disponible, si oui il enregistre l'emprunt, sinon il peut proposer à l'emprunteur une réservation.

7.7 Centre de soins et dossiers médicaux

On souhaite gérer les dossiers médicaux d'un centre de soins.

1. Le personnel du centre de soins, peut être médical ou administratif. Il peut accéder aux dossiers médicaux des patients et leur proposer un rendez-vous avec un médecin. La consultation d'un dossier médical est limitée pour le personnel administratif aux informations administratives, le personnel médical peut de son côté accéder à toutes les informations médicales concernant le patient, effectuer un diagnostic, rédiger un rapport de consultation, consulter des rapports de consultation précédents et faire des prescriptions.
2. Afin d'accéder à l'application le personnel du centre doit se connecter. Chaque membre du centre de soins dispose d'un login et d'un mot de passe, et est identifié en tant que 'Personnel médical' ou 'personnel administratif'. Pour le personnel médical on dispose de plus de ses nom, prénom et de son identifiant professionnel.
3. Pour consulter son dossier médical le patient dispose d'un code d'accès à son dossier, son login est son numéro de sécurité sociale. On dispose de plus pour le patient de ses nom et prénom, de son sexe et de sa date de naissance.



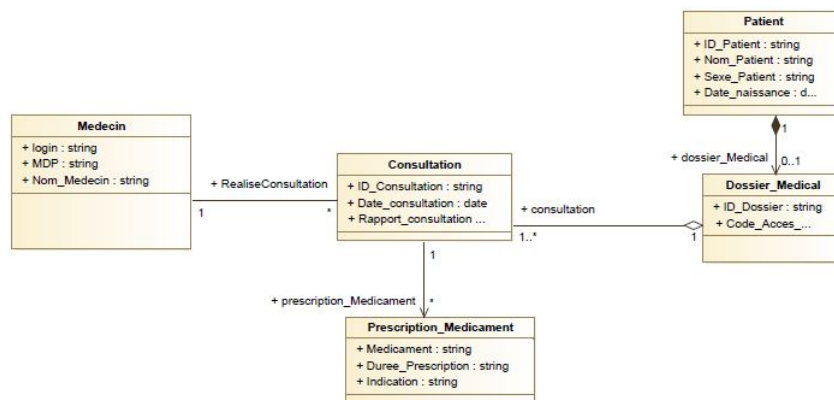
4. Les patients suivent des traitements, demandent des rendez-vous, effectuent des consultations chez les médecins et passent des examens et des tests dans les laboratoires d'analyse. Ils peuvent consulter leur dossier : parties administrative et médicale.
5. Une consultation est effectuée par un médecin sur un patient à une date et heure donnée. A l'issue de la consultation le médecin rédige un rapport de consultation et peut faire des prescriptions.
6. Une prescription peut être une prescription de médicament, une prescription d'analyse

ou une prescription d'opération chirurgicale. Une prescription de médicament décrit la prescription d'un médicament avec une durée de prise et une posologie (indications de prise).

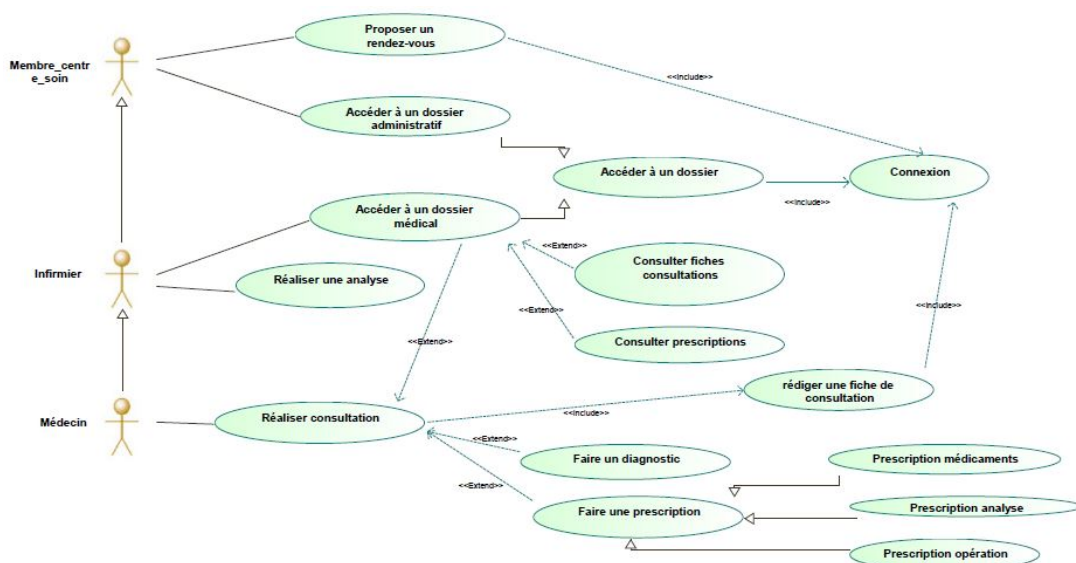
7. Les éléments concernant les consultations d'un patient consultation sont liés à ce patient et ne peuvent exister sans lui, de même pour les prescriptions liées à une consultation.
8. Lors de la consultation de la partie médicale d'un dossier, le personnel médical peut consulter les fiches de consultations ainsi que les prescriptions.

7.7.1 Diagramme de classe

Créer un diagramme de classe permettant de décrire les données de la base de données "Dossier médical" qui permet d'enregistrer les informations concernant la gestion des dossiers médicaux des patients du centre médical et donc principalement les informations concernant la consultation d'un patient auprès d'un médecin avec les prescriptions de médicaments. On ne gère pas ici les résultats des examens et des tests en laboratoire.

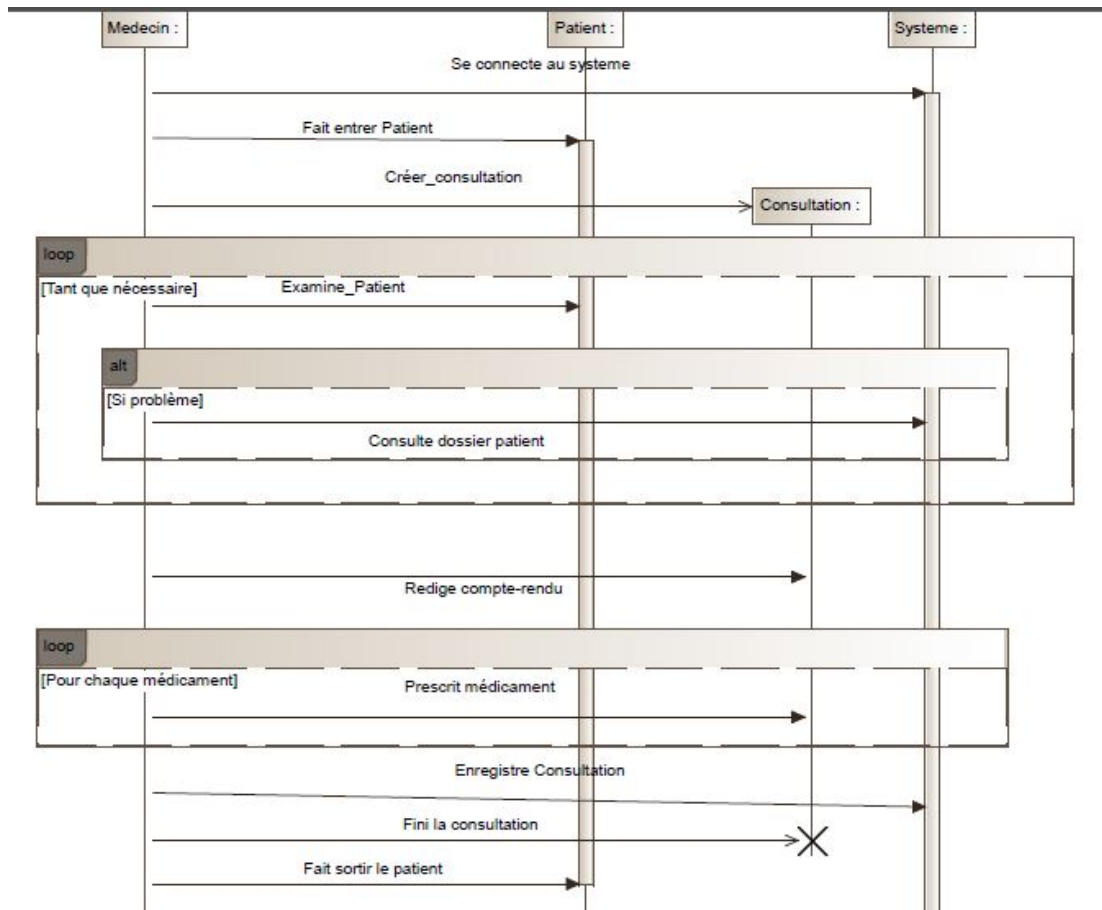


7.7.2 Diagramme de cas d'utilisation



7.7.3 Diagramme de séquence

Modélisez la suite complète de traitements à réaliser pour la réalisation d'une session de consultations par un médecin. Celui-ci doit pour commencer, se connecter au système, ensuite pour chaque patient, il le fait entrer, ouvre son dossier médical et commence la consultation. Si le patient n'a pas encore de dossier médical il le crée. Ensuite il l'examine, en cas de doute il consulte son dossier et continue la consultation. Lorsque le médecin considère que la consultation est faite, il rédige son compte-rendu et ensuite prescrit 0 ou plusieurs médicaments. Enfin il enregistre la consultation et reconduit le patient et passe ensuite au patient suivant.



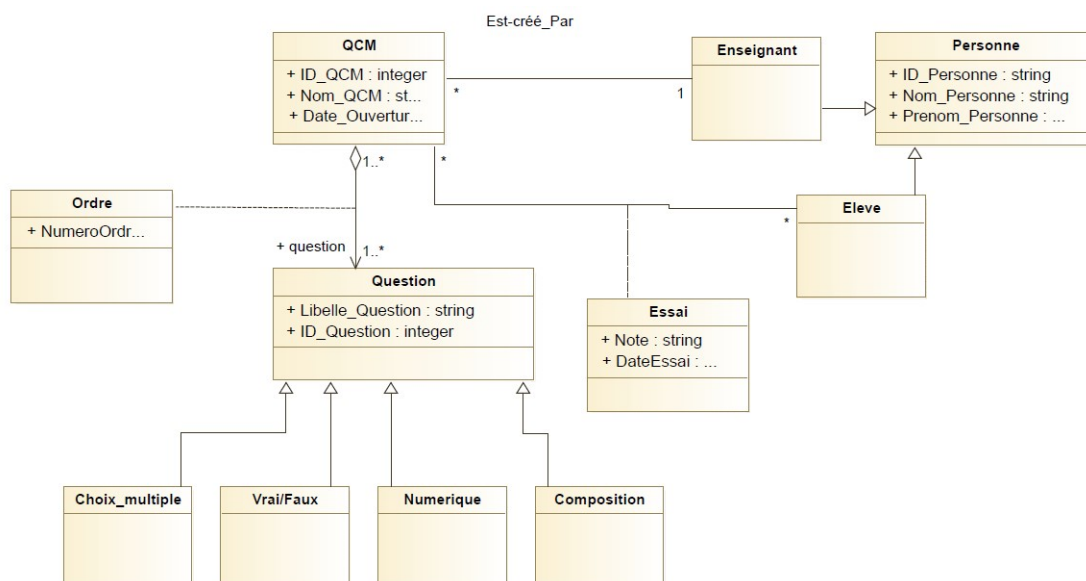
7.8 QCMs

On souhaite gérer les QCMs d'une formation en créant une application permettant de gérer les questions et de créer des contrôles sous la forme de QCM.

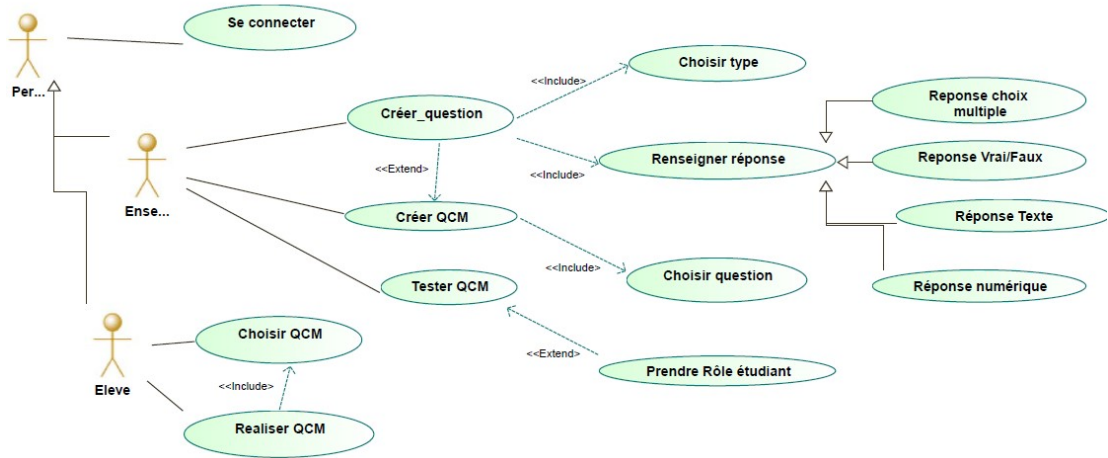
1. Les enseignants et les élèves sont décrits par un identifiant, un nom et un prénom, un login et un mot de passe. Quelle que soit l'opération à réaliser sur le système il faut se connecter.
2. L'enseignant peut créer des QCMs. Pour cela il peut créer des questions qui sont décrites par un texte et un type.
3. Une question peut être à choix multiples, de type Vrai/faux, de type texte à composer ou numérique.
4. Pour créer un QCM, l'enseignant lui donne un titre, une date d'ouverture et une date de fermeture. Il peut ensuite y ajouter des questions en précisant dans quel ordre elles apparaîtront.
5. Une question peut être utilisée dans plusieurs QCMs. Elle appartient à au moins un QCM.
6. L'élève peut réaliser à des dates différentes un même QCM, il obtient à chaque essai une note.
7. Lorsque l'élève veut réaliser un QCM, il doit se connecter, choisir dans la liste de QCMs disponibles celui qui l'intéresse, le démarrer et ensuite répondre à chaque question. Lorsqu'il a choisi sa réponse il peut cliquer sur le bouton suivant et passer ainsi à la question suivante. A la fin du QCM, il peut relire une fois l'ensemble du QCM ou valider ses réponses. S'il valide ses réponses il obtient sa note finale.
8. Pour tester un QCM, l'enseignant peut prendre le rôle "élève".

7.8.1 Diagramme de classe

Créez un diagramme de classe permettant de décrire les données de la base de données "QCM" qui permet d'enregistrer les informations concernant la gestion des QCMs. Nous ne traitons pas ici le détail des réponses aux questions (lors de la création d'une question ou de la réalisation d'un QCM par un élève).

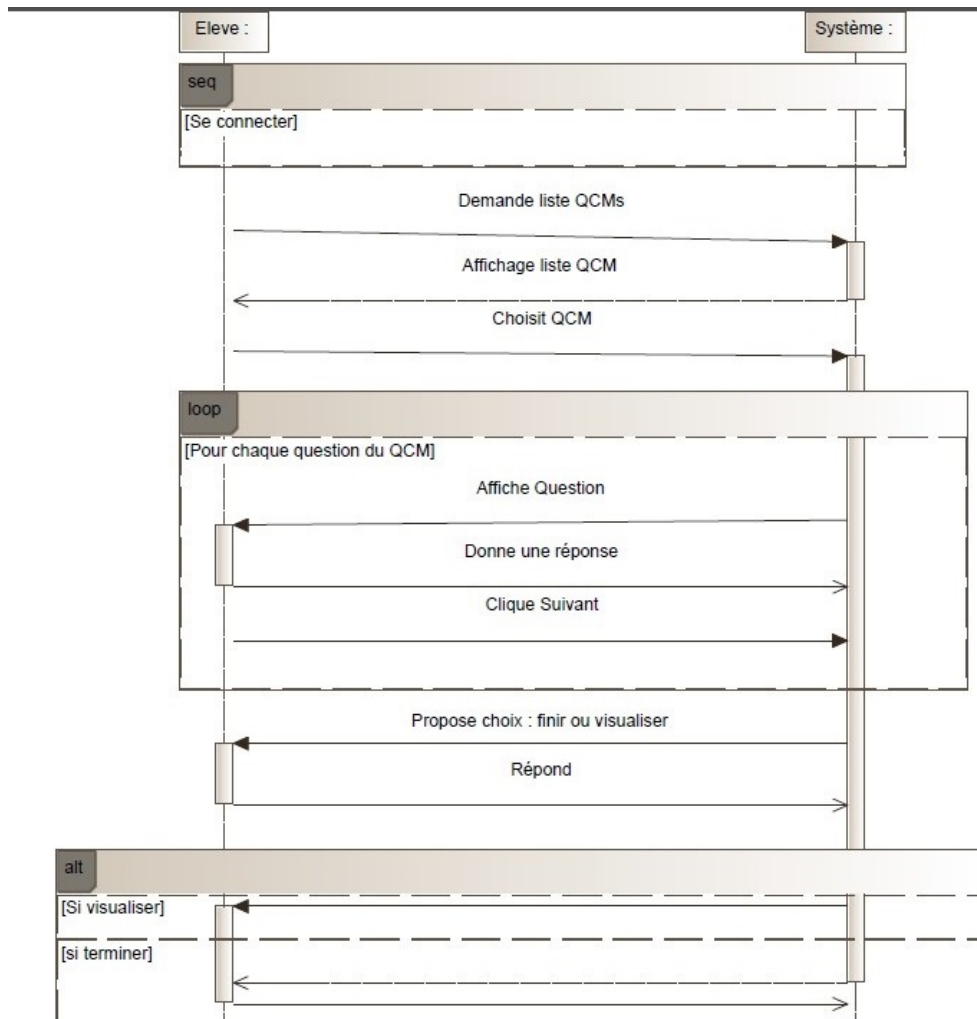


7.8.2 Diagramme de cas d'utilisation



7.8.3 Diagramme de séquence

Modélisez la suite complète de traitements à réaliser pour qu'un élève réalise un QCM.



7.9 Association d'étudiants

Une association d'étudiants souhaite développer une application informatique permettant de suivre les projets de ses membres. L'objectif est de pouvoir dire quels projets sont en cours, qui s'en occupe, qui intervient et avec quel rôle.

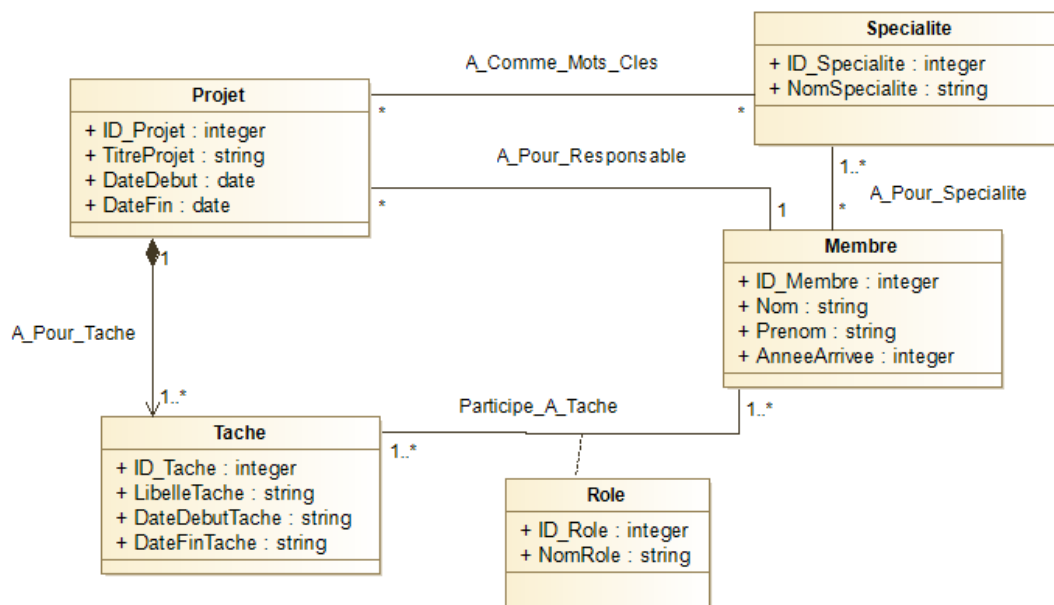
Un membre de l'association est décrit par ses nom et prénom et son année d'arrivée dans l'association, ainsi que ses spécialités. Un projet a un responsable qui est un membre de l'association, un titre, une date de début et de fin et on peut lui associer quelques mots clés qui correspondent à des spécialités. Un projet se décompose en tâches auxquelles sont associés un ou plusieurs membres de l'association. Ces tâches ont une date de début et de fin et un Libellé. Chaque membre participe à une tâche avec un rôle spécifique (trésorier, secrétaire, développeur, ...). Certains membres de l'association peuvent se connecter à l'application de gestion des projets, les autres sont simplement visiteurs.

Un visiteur peut consulter la liste des projets. Pour chaque projet il peut consulter ses tâches et les membres associés à chaque tâche, ainsi que le responsable du projet. Il peut faire une recherche par spécialité des projets et des membres. Il peut aussi rechercher un membre et visualiser l'ensemble des rôles et responsabilité de ce membre.

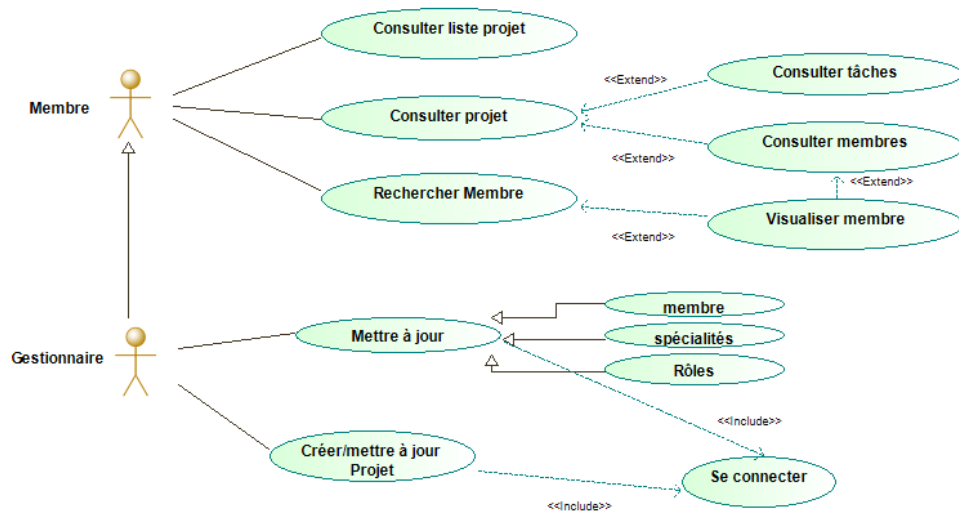
Une personne connectée peut mettre à jour la liste des membres, des spécialités, des rôles. Elle peut aussi créer ou modifier un projet. Cette création suppose la saisie des informations de description du projet : titre, date de début et de fin, choix du responsable et association de mots clés. Une vérification a alors lieu pour vérifier que le projet n'a pas déjà été créé. Ensuite, si le projet n'était pas encore créé, il faut décrire chaque tâche du projet. Pour chacune on saisit son nom, ses dates de début et fin et on recherche dans la liste des membres ceux qui y participeront en leur associant un rôle.

7.9.1 Diagramme de classe

Créez un diagramme de classe permettant de décrire les données de la base de données "Association_Projets".

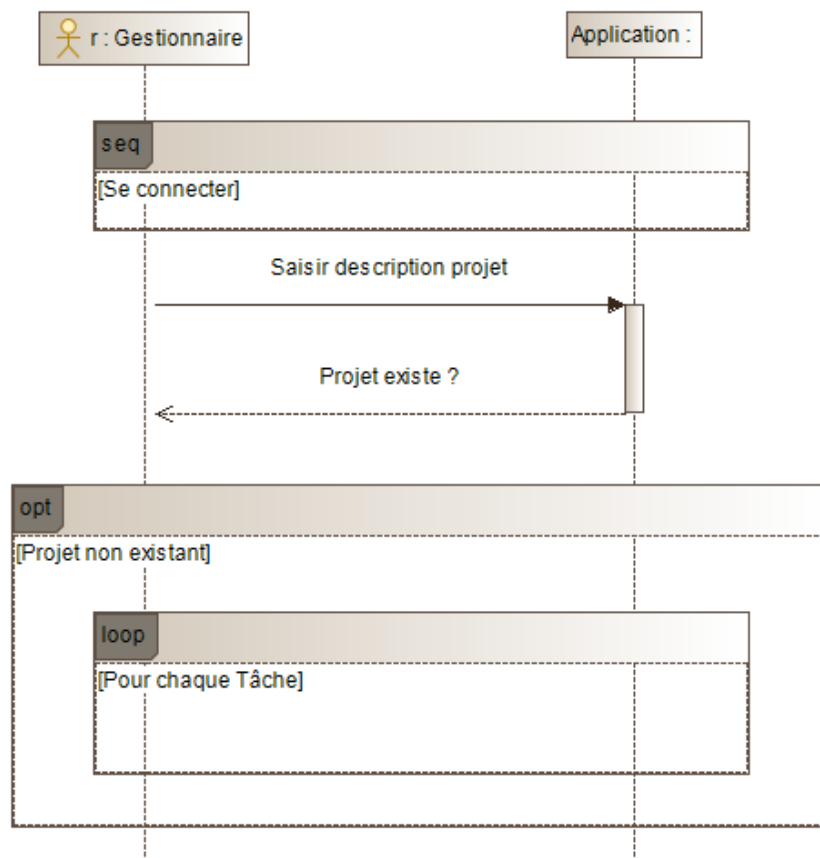


7.9.2 Diagramme de cas d'utilisation



7.9.3 Diagramme de séquence

Modélisez la suite de traitements à réaliser pour créer un projet.



7.10 Suivi des étudiants

On désire réaliser un système de gestion de suivi des étudiants d'une école.

Tout étudiant est caractérisé par un identifiant, son nom, son prénom, sa date de naissance, et son adresse. Un étudiant est inscrit dans une année de diplôme et n'appartient qu'à un seul groupe. Chaque année de diplôme peut avoir 1 à plusieurs groupes en fonction du nombre d'étudiants et est composée de différents modules.

Un module, décrit par un nom, un objectif et une masse horaire, est composé de séances. Chaque séance a lieu à une date donnée, dans une salle donnée et possède un objectif pédagogique. Les séances sont classées en 3 types : "Cours", "Travaux Dirigés" et "Travaux Pratiques".

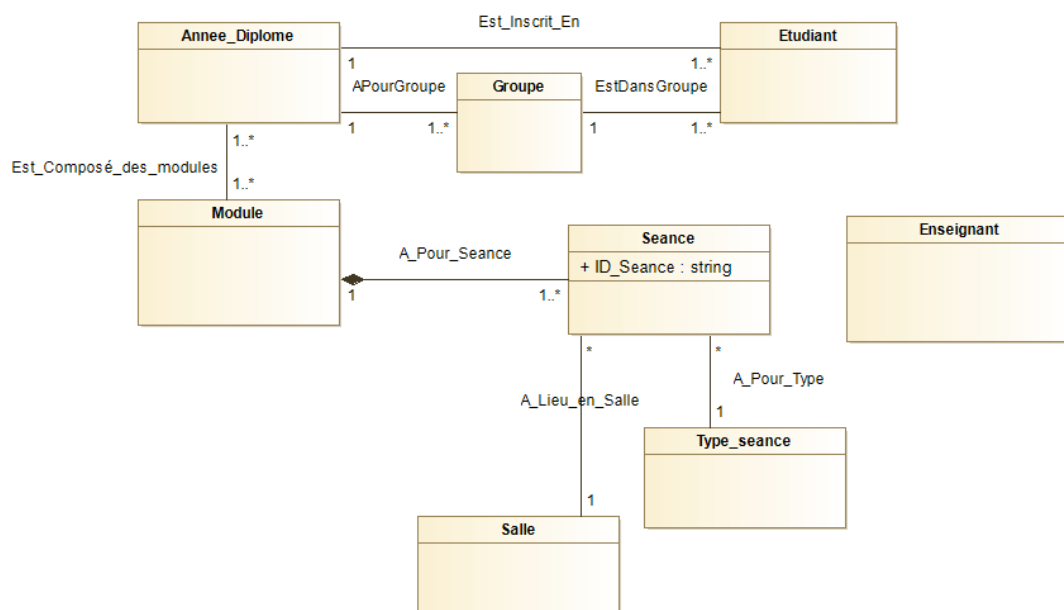
Un enseignant peut se connecter au système de suivi des étudiants à partir de son mobile ou d'un ordinateur sur le réseau de l'école. Ensuite il choisit l'année de diplôme, le groupe, le module concerné et le type de la séance. Enfin il saisit l'objectif de la séance, le volume horaire réalisé et enregistre les étudiants absents à la séance.

L'application sera utilisée par les différents types d'utilisateurs qui doivent s'identifier pour utiliser le système :

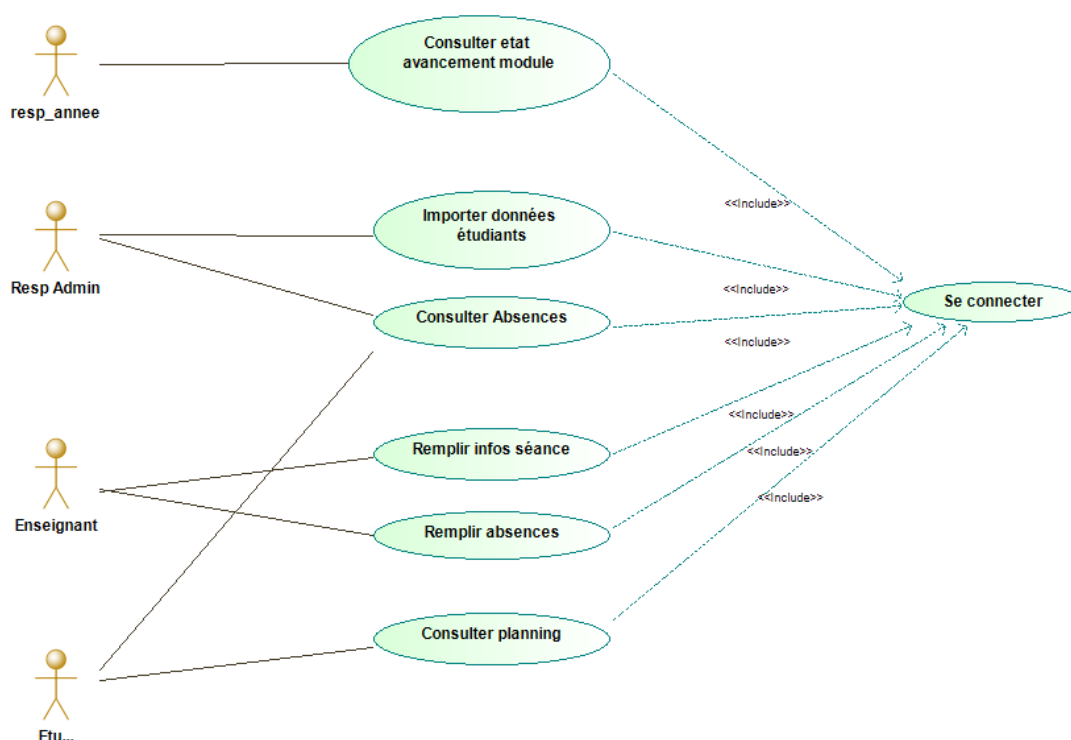
1. Le responsable de l'année qui gère les formateurs et les modules et peut consulter l'état d'avancement des modules.
2. Le responsable administratif qui gère les informations concernant les étudiants. Il peut en particulier importer les informations concernant les étudiants depuis l'application "Apogée". Il peut de plus consulter les absences des étudiants.
3. L'enseignant remplit les informations concernant la séance et les absences des étudiants.
4. Les étudiants peuvent accéder au système pour consulter leur planning et leurs absences.

7.10.1 Diagramme de classe

Créez un diagramme de classe permettant de décrire les données de la base de données "Suivi Etudiants" qui permet d'enregistrer les informations concernant la gestion des diplômes, module et étudiants de l'école.

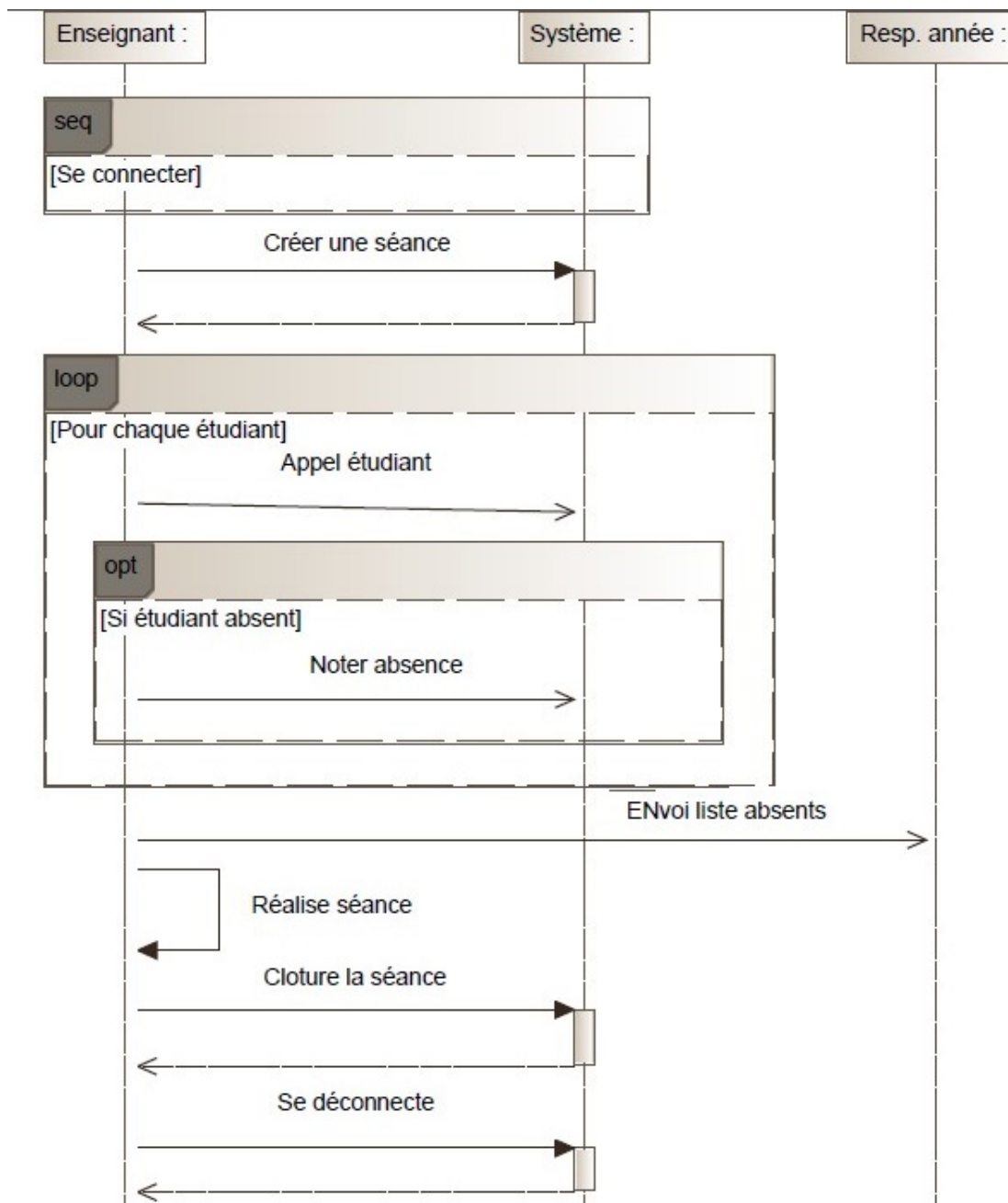


7.10.2 Diagramme de cas d'utilisation



7.10.3 Diagramme de séquence

Modélisez la suite complète de traitements à réaliser pour qu'un enseignant réalise une séance, de la description de la séance dans le système, à l'appel des étudiants, la saisie des absences, l'envoi des absences au responsable de l'année et enfin la clôture de la séance.



7.11 Taxis

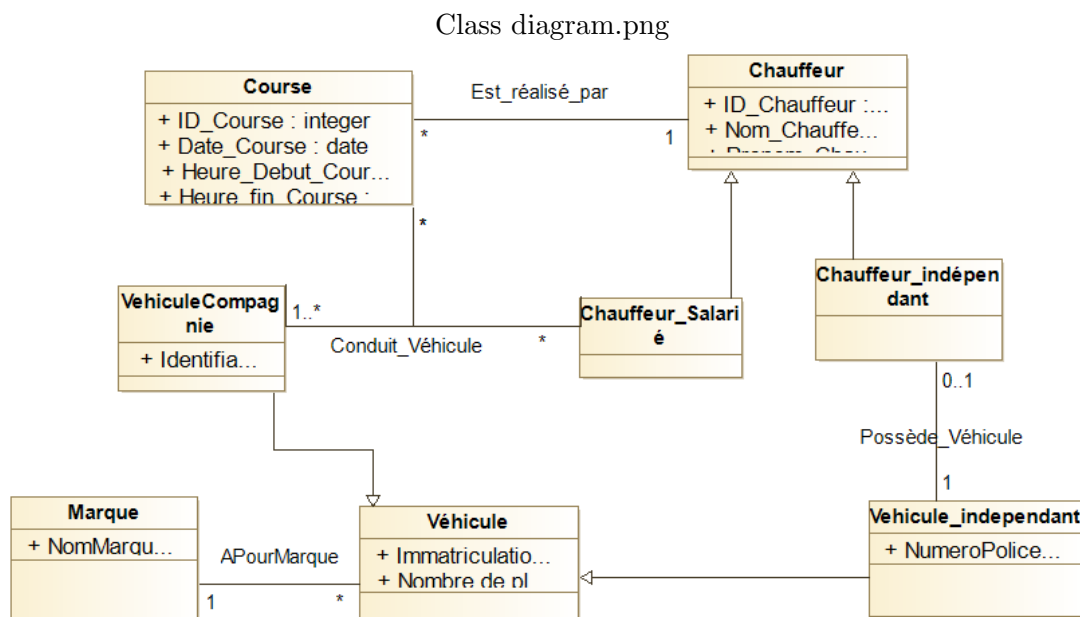
Une compagnie de taxis souhaite gérer ses véhicules, chauffeurs et courses en créant une application informatique.

1. Une course a une date, une heure de début et de fin. Elle est réalisée par un chauffeur.
2. Un chauffeur a un nom et un prénom. Il peut être chauffeur salarié ou chauffeur indépendant.
3. Un véhicule a une immatriculation, une marque et un nombre de places.
4. La compagnie de taxis possède des véhicules qui sont utilisés par les chauffeurs salariés. Les chauffeurs indépendants possèdent leur propre véhicule.
5. Les véhicules de la compagnie possèdent un identifiant spécifique, alors que les véhicules des chauffeurs indépendants ont un numéro de la police d'assurance du véhicule.

6. compagnie est utilisé.
7. La compagnie de taxis gère les demandes des clients. En fonction du point de départ de la course demandée, elle contacte les chauffeurs salariés en service. Si aucun n'est disponible elle contacte les chauffeurs indépendants.
8. Lorsqu'un chauffeur prend en charge un client, il se rend sur le point de départ à l'horaire donné. Il récupère le lieu à rejoindre et tape les coordonnées dans l'outil de navigation du véhicule et suit les instructions jusqu'à arriver à destination. Il indique alors le coût de la course au client et encaisse le prix de la course. Il peut alors se mettre en état de disponibilité pour une prochaine course ou se mettre en pause.
9. Le paiement d'une course peut être réalisé par carte, en liquide ou via un coupon de course (qui paie la course dans son intégralité).

7.11.1 Diagramme de classe

Créez un diagramme de classe permettant de décrire les données de la base de données "TAXIS" qui permet d'enregistrer les informations concernant la gestion des taxis, leurs chauffeurs et les courses pour une compagnie de taxis.



7.11.2 Diagramme de séquence

Créez un diagramme de séquence qui modélise la suite complète de traitements à réaliser de la demande d'un client au paiement de la course.

7.12 Hospitalisations

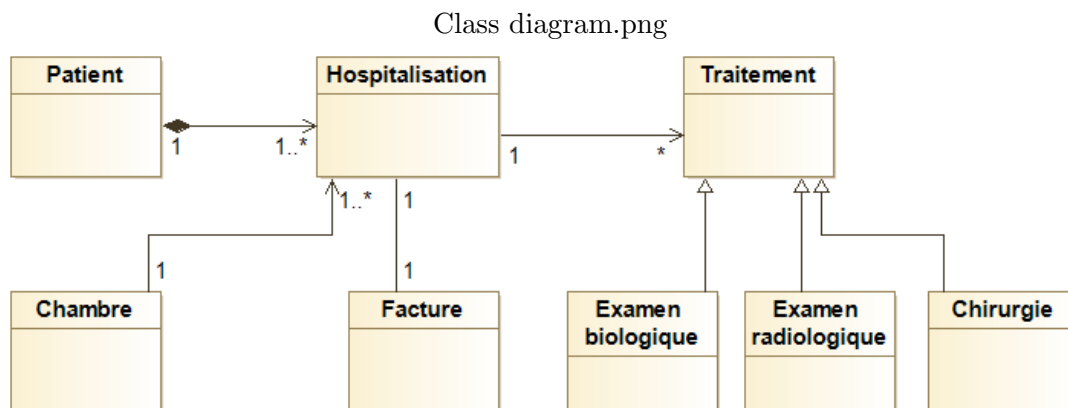
On souhaite gérer les hospitalisations dans un hôpital en créant une application permettant de gérer les patients et les différents types d'hospitalisation.

1. Un patient est décrit par un identifiant, un nom et prénom, une date de naissance, une adresse, un téléphone et une adresse mail. Un patient peut consulter son dossier d'hospitalisation et modifier la partie "état civil".

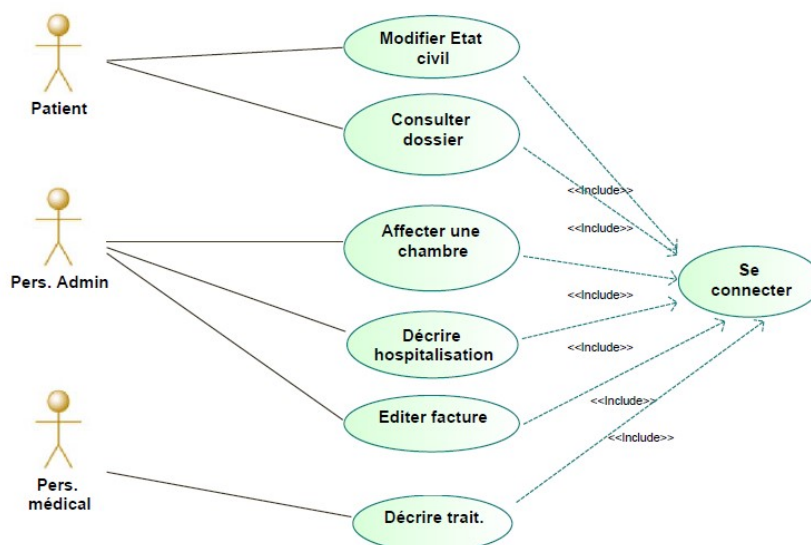
2. Une hospitalisation correspond à un patient. Elle est décrite par un personnel administratif avec une date de début et de fin, un motif d'admission. Elle donne lieu à plusieurs traitements. Une facture est éditée pour chaque hospitalisation, elle regroupe les frais liés à chaque traitement réalisé lors de l'hospitalisation. Lors d'une hospitalisation, une chambre est affectée. Une chambre a un numéro, un étage et un prix journalier.
3. Un traitement est décrit par démarre à une date et a un prix. Il peut être de différents types : Examen biologique, Examen radiologique ou Chirurgie. Un examen biologique a une description et un résultat. Un examen radiologique a une description, un résultat et une image. Une chirurgie est décrite par une description et les noms d'un chirurgien et d'un anesthésiste.
4. Lorsqu'un patient est hospitalisé, les détails de l'hospitalisation sont saisis, une chambre est sélectionnée parmi les chambres disponibles. A chaque traitement le détail est renseigné (en fonction du type de traitement) par un personnel médical. Lors de la sortie, un personnel administratif édite une facture qui reprend les frais liés à chaque traitement et à la chambre. Enfin la chambre est libérée.

7.12.1 Diagramme de classe

Créez un diagramme de classe, (4 points) permettant de décrire les données de la base de données "Hospitalisation" qui permet d'enregistrer les informations concernant la gestion des hospitalisations.



7.12.2 Diagramme de cas d'utilisation



7.13 Société de livraison express à domicile

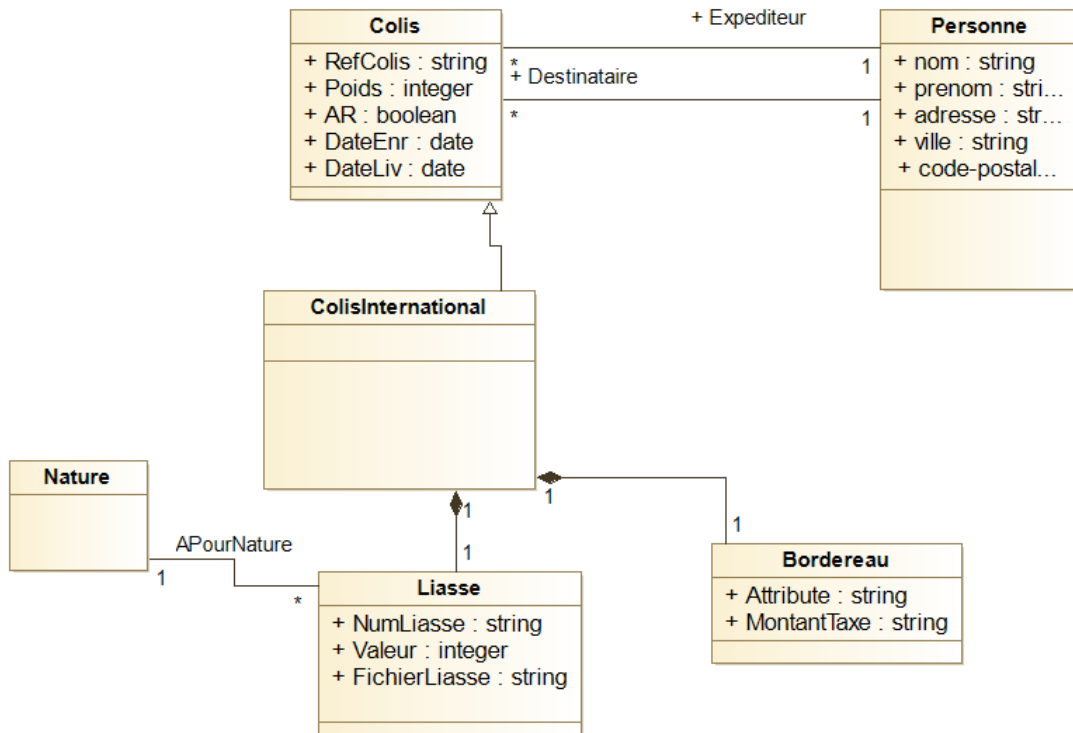
La société de livraison express à domicile, LED, reçoit chaque jour, via son service Réception, des demandes de livraison de colis en France et à l'international. Le service Réception enregistre le colis, c'est-à-dire : la référence du colis, les coordonnées de l'expéditeur et du destinataire (qui se composent d'un nom, prénom, adresse, ville, code postal et pays), le poids du colis, le choix ou non d'un accusé de réception et les dates d'enregistrement du colis et de livraison estimée. Il imprime enfin un récépissé pour le client. Dans le cas d'un colis à destination de la France, le service Réception gère la facturation et une fois le paiement effectué, il transmet le colis au service Logistique pour son acheminement.

Le tarif de livraison d'un colis est calculé en fonction du poids du colis, de sa destination et du choix ou non d'un envoi avec accusé de réception (un forfait de 15 est demandé si le client demande un envoi avec accusé de réception).

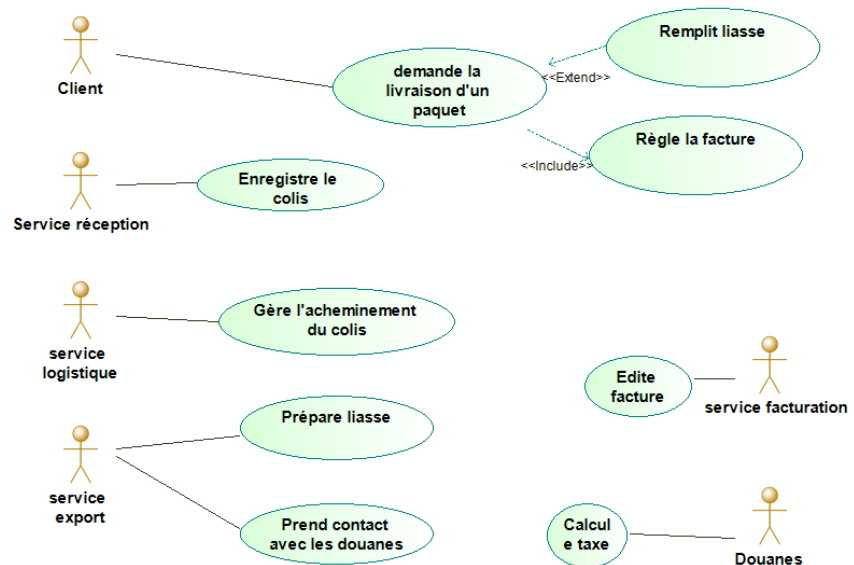
Un colis à destination de l'international, doit respecter la réglementation douanière et donc faire l'objet de démarches plus lourdes qui rallongent son délai d'acheminement de 48h au moins et est sur-facturé de 10%. Le client doit remplir et signer une liasse de transport qui précise la nature (liste définie par la réglementation douanière) et la valeur du contenu du colis à acheminer. Le colis, accompagné de la liasse, est transmis au service Export de l'entreprise. La liasse est enregistrée avec un numéro, une signature électronique du client et un fichier scanné (la liasse est directement scannée par le service Export). Le service Export s'occupe alors des formalités avec les douanes Françaises. Le colis ne peut être acheminé avant l'accord des douanes qui se matérialise par un bordereau avec la référence du colis à acheminer et le montant de la taxe à la charge du client. Le service Export de l'entreprise transmet alors l'information au service de facturation. Celui-ci émet ensuite la facture finale à destination du client. Enfin, après le règlement de la facture par le client, le service Export est informé et transmet le colis avec le bordereau des douanes au service Logistique qui se charge de la livraison.

7.13.1 Diagramme de classe

Créez un diagramme de classe permettant de décrire les données de la base de données "Colis" qui permet d'enregistrer les informations concernant la gestion des colis à traités par la société LED.



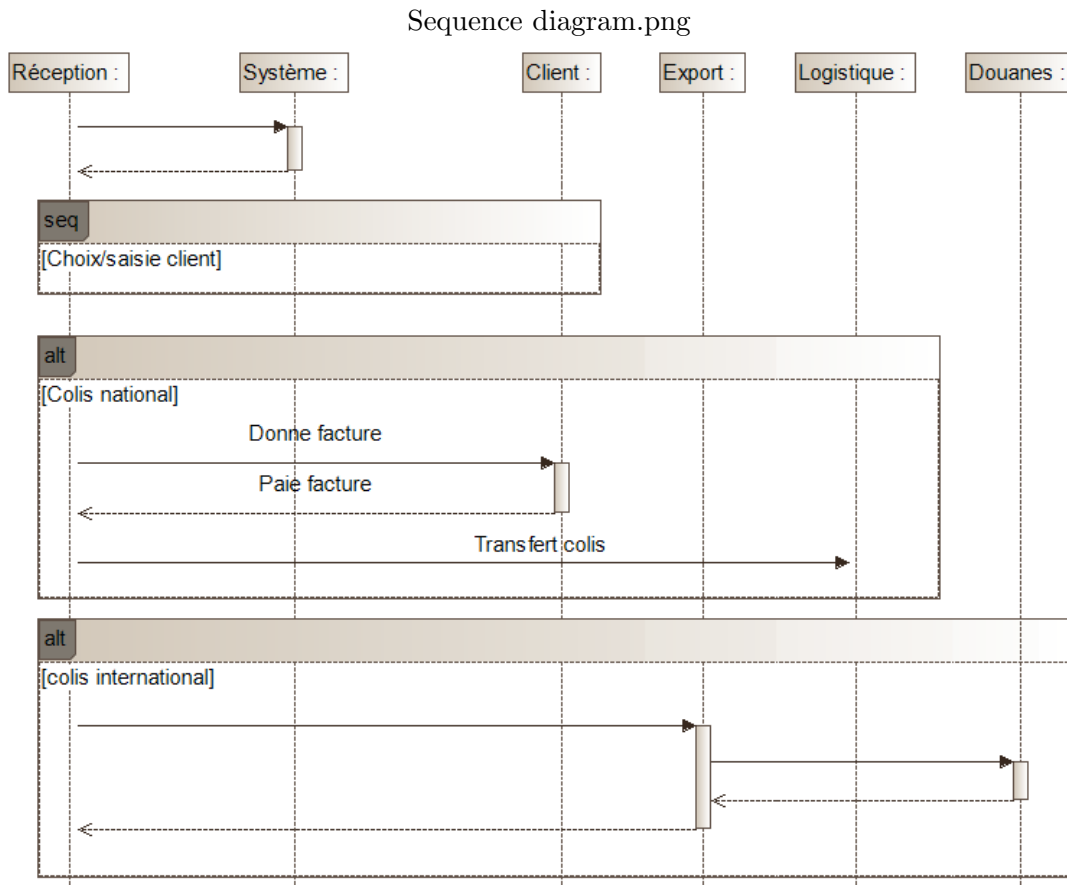
7.13.2 Diagramme de cas d'utilisation



Use Case diagram.png

7.13.3 Diagramme de séquence

(3 points) Modélisez la suite complète de traitements à réaliser de la réception d'un colis par le service Réception à son transfert au service Logistique.



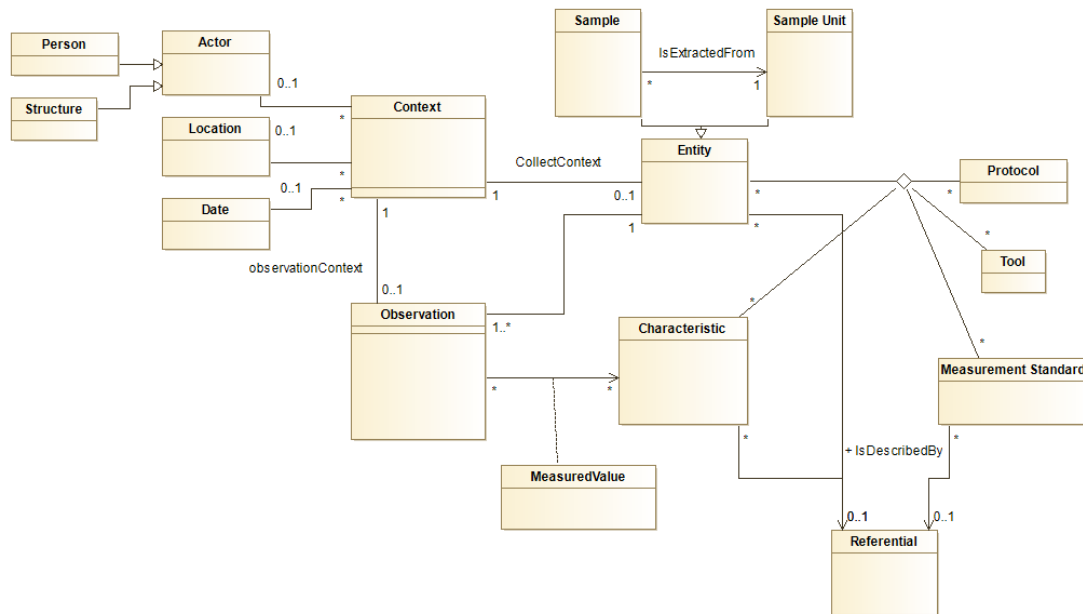
7.14 Observatoire

Pour représenter un observatoire environnemental, un modèle d'observation est créé. Une observation porte sur une entité, qui peut être un échantillon ou un prélèvement. Les entités sont caractérisées par un identifiant et un nom provenant d'un référentiel. Elles sont collectées dans un contexte de collecte. Un prélèvement est issu d'un échantillon. On peut extraire plusieurs échantillons d'un prélèvement. Une observation correspond à la mesure d'une ou plusieurs caractéristiques sur une entité. Elle donne une valeur pour chaque caractéristique mesurée. L'observation a un contexte d'observation. Un contexte décrit l'acteur qui a réalisé l'observation ou la collecte, un lieu et une date. L'acteur peut être une personne ou une structure. Un contexte de collecte ou d'observation correspond à une unique entité et n'existe que par elle.

Pour décrire comment sera effectué l'ensemble des observations sur une entité on décrit pour chaque caractéristique le protocole, l'outil et l'unité de mesure utilisés. Les listes de protocoles connus, d'outils et d'unités sont décrits par des référentiels. Ils peuvent être utilisés pour différentes observations.

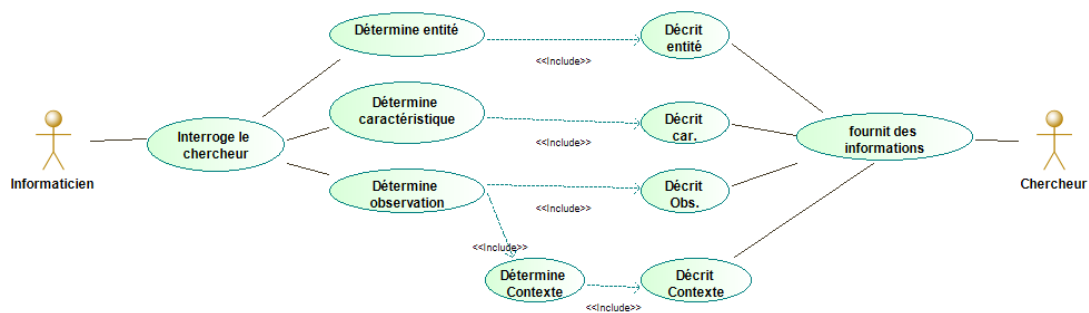
7.14.1 Diagramme de classe

Créer un diagramme de classe permettant de décrire les données de la base de données "Observation" qui permet d'enregistrer l'ensemble des informations concernant les observations sur des entités.



7.14.2 Diagramme de cas d'utilisation

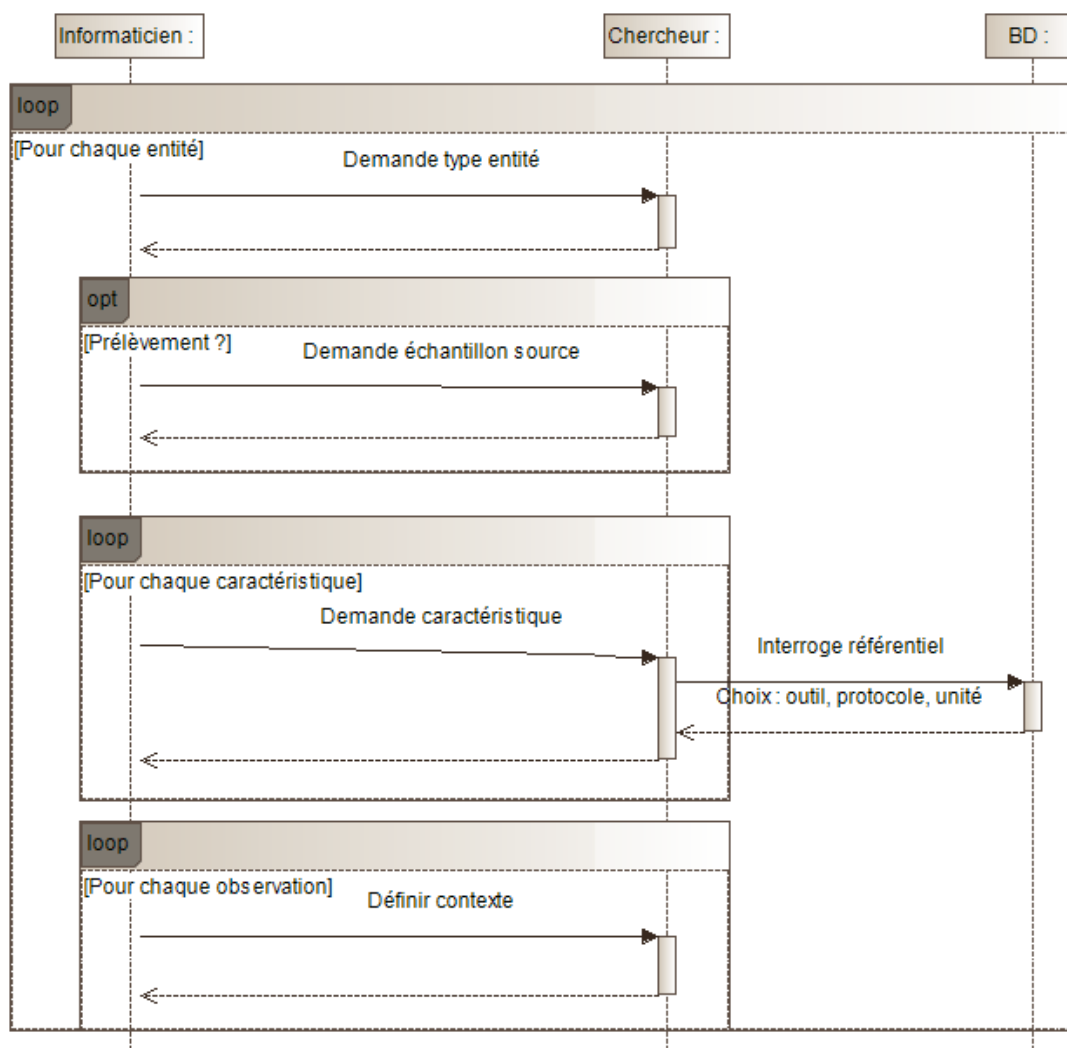
Quels sont les acteurs de ce système. Décrire un diagramme de cas d'utilisation donnant les principales fonctionnalités du système.



7.14.3 Diagramme de séquence

Construire le diagramme de séquence correspondant à la construction d'une base de données d'observation en utilisant les concepts présentés. Pour construire un observatoire l'informaticien interroge selon une procédure prédéfinie le chercheur. L'informaticien demande au chercheur de définir les entités qu'il souhaite observées. Pour chaque entité il définit s'il s'agit d'un échantillon ou d'un prélèvement. Dans le dernier cas il demande au chercheur de préciser de quel échantillon est issu le prélèvement. Pour chaque entité, l'informaticien demande au chercheur quelle caractéristique sera mesurée. Et pour chaque caractéristique il choisit dans les listes

adéquates le protocole, l'outil et l'unité choisie pour la mesure. Enfin, l'informaticien construit les observation en demandant au chercheur de décrire pour chacune, l'acteur, le lieu et la date.



7.15 Structures d'un établissement de recherche

Nous nous proposons ici de décrire un diagramme de classe permettant de décrire les structures composant un établissement ainsi que leurs historiques.

Une structure a un identifiant unique qui permet de l'identifier durant tout son cycle de vie (modification, ...). Elle a par ailleurs un nom cours, un nom long et une année de création. Chaque structure a un niveau qui peut évoluer dans le temps.

Une structure de niveau $i+1$ appartient à une structure de niveau i (pour $i \geq 1$).

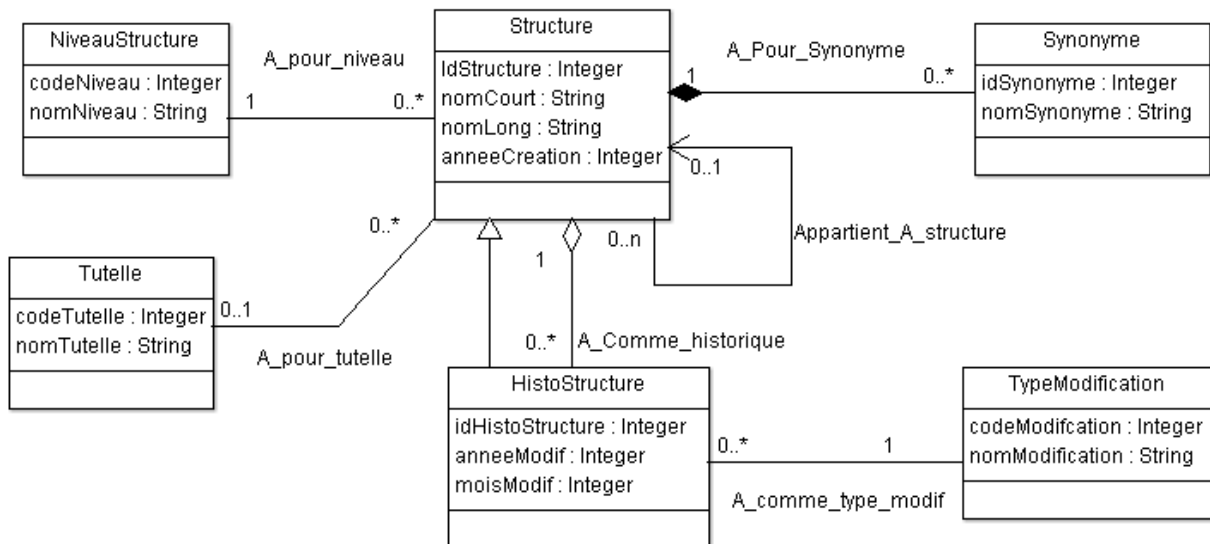
Une structure peut avoir au plus une tutelle.

Un niveau de structure est caractérisé par un code et un nom (Etablissement, laboratoire, équipe, ...). Plusieurs structures peuvent avoir le même niveau. Une tutelle a un code et un nom. Plusieurs structures peuvent avoir une même tutelle.

Un structure peut avoir plusieurs noms, nous gérons donc la liste des synonymes de la structure. Ceux-ci n'ont aucune existence en dehors de la structure. Ils sont décrits par un code et un nom. Dans le temps les structures évoluent et l'on veut garder l'historique de chaque structure. Toute

structure créée sera conservée dans la base de données.

L'évolution d'une structure peut correspondre à différents types de modification : Changement de nom, de tutelle, fusion avec une autre structure, suppression, ... L'évolution d'une structure prend effet à une date donnée (année et mois). On souhaite garder toutes les informations concernant la structure avant son changement. Un historique dépend d'une structure et n'a pas de raison d'exister sans la structure. On peut avoir pour une même structure plusieurs historiques.



7.16 Réserve de salles de cours et de matériel pédagogique

On désire gérer la réservation des salles de cours et du matériel pédagogique (ordinateur portable ou/et vidéo-projecteur) de la faculté des sciences.

Les enseignants sont les seuls à pouvoir réaliser des réservations (sous réserve de disponibilité de la salle ou du matériel).

On propose différents types de consultations :

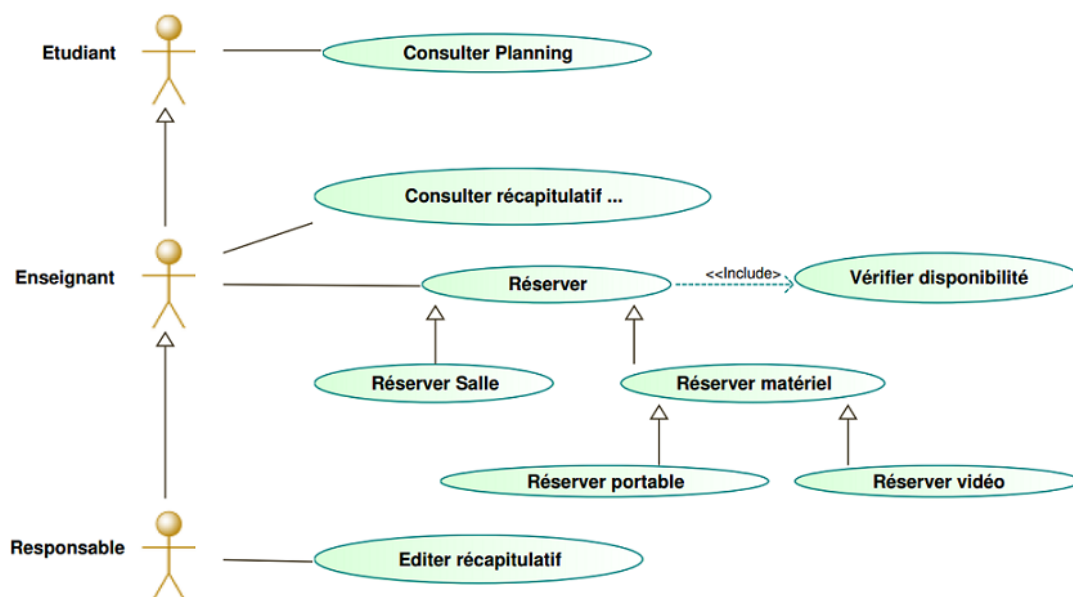
- Le planning des salles peut être consulté par tout le monde (enseignants et étudiants).
- Le récapitulatif horaire par enseignant (calculé à partir du planning des salles) ne peut être consulté que par les enseignants.

Pour chaque formation il existe un enseignant responsable qui peut éditer le récapitulatif horaire de la formation.

Modéliser cette gestion des salles de cours et du matériel par un ensemble de diagrammes de cas d'utilisation, et donnez la représentation textuelle du cas d'utilisation lié à l'acteur "Enseignant".

7.17 Diagramme de séquence : Gestion d'un emprunt dans une bibliothèque

Nous souhaitons ici décrire les traitements réalisés lorsqu'un adhérent veut emprunter un exemplaire d'une oeuvre donnée.

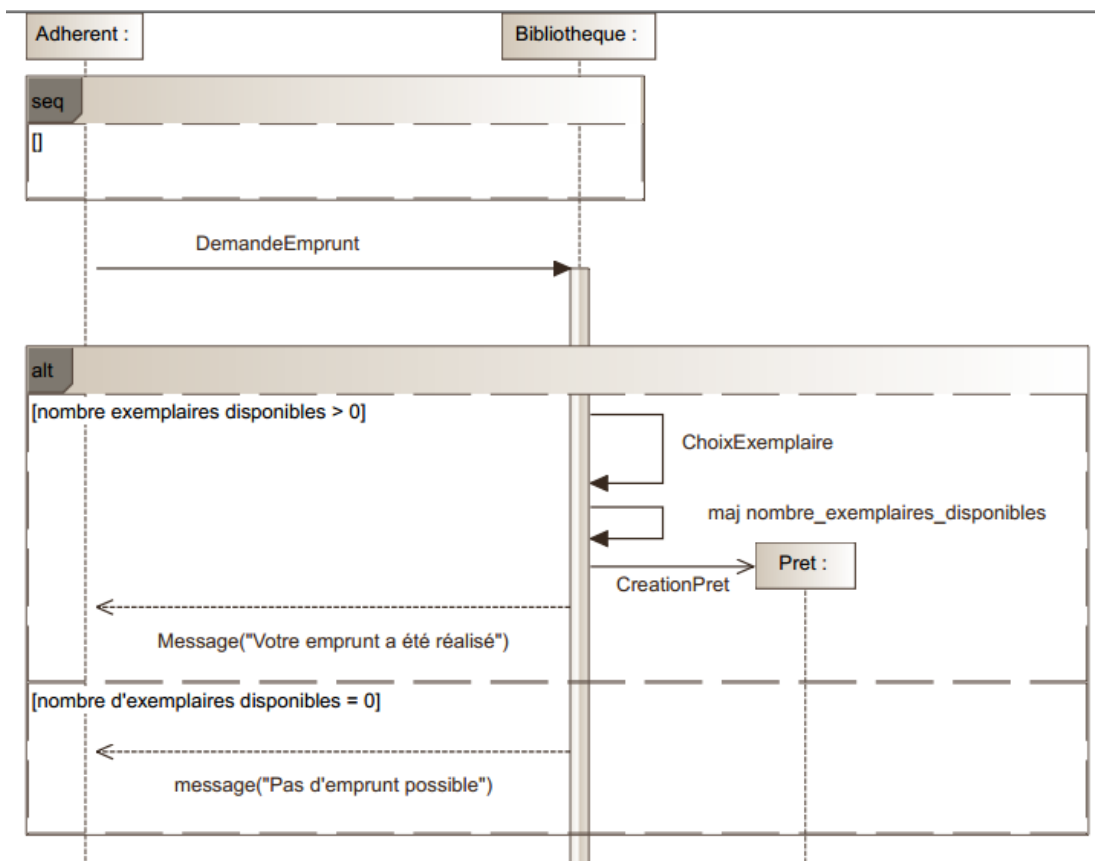


L'adhérent, après s'être connecté demande l'emprunt d'une oeuvre présente dans le catalogue de la bibliothèque.

Le système vérifie si il reste au moins un exemplaire disponible dans la bibliothèque.

- Si il n'y a pas d'exemplaire disponible le système retourne à l'adhérent le message *"Pas d'emprunt possible"*.
- Dans le cas où au moins un exemplaire est disponible, le système sélectionne un exemplaire pour l'emprunt, crée l'emprunt, met à jour le nombre d'exemplaires disponibles et renvoie à l'adhérent le message *"Votre emprunt a été réalisé"*.

Etablir un diagramme de séquence qui décrit l'emprunt d'un exemplaire d'une oeuvre donnée. Nous supposons ici que le traitement de connexion est connu.



7.18 Sujets sans corrigé

7.18.1 Diagramme de séquence du publiphone

On considère un système simplifié de publiphone à pièces, dont voici les différentes caractéristiques :

- Le prix minimal d'une communication interurbaine est de 0,2 euros,
- Après l'introduction de la monnaie, l'utilisateur a deux minutes pour composer son numéro (ce délai est décompté par le standard),
- La ligne peut être libre ou occupée,
- Le correspondant peut raccrocher le premier,
- Le publiphone consomme de l'argent dès que l'appelé décroche et à chaque unité de temps (UT) générée par le standard. Supposons ici que l'unité de temps considérée par le standard est la minute,
- On peut ajouter des pièces à tout moment (0,2 euros pour bénéficier d'une minute de communication supplémentaire),
- Lors du raccrochage, le solde de monnaie est rendu.

Etablir un diagramme de séquence qui décrit l'utilisation du publiphone. On suppose que l'utilisateur du publiphone ne possède que des pièces de 0,2 euros qu'il ajoute les unes après les autres quand c'est nécessaire. On suppose également que c'est l'utilisateur du publiphone qui raccroche en premier.

7.18.2 Auto-école

Une auto-école souhaite construire une application pour gérer le suivi de ses clients aux séances et aux examens du code de la route. Lorsqu'un client vient s'inscrire à l'auto-école on lui attribue automatiquement un numéro unique et on lui demande ses nom, prénom, adresse et date de naissance. Une fois inscrit le client peut assister à autant de séances de code qu'il le souhaite.

Pour chaque séance, qui a lieu à une date et une heure données, après s'être connecté, le moniteur d'auto-école affiche l'ensemble des DVD en choisit un et ensuite choisit une série de questions sur le DVD, il lance ensuite la séance une fois que les clients sont connectés. Une fois que les clients présents ont fini de répondre le moniteur clôt la séance. Chaque DVD est identifié par un numéro, a un nom d'éditeur et est composé de 6 séries, numérotées de 1 à 6. Chaque série est composée de 40 questions choisies dans un pool de questions. Chaque question est identifiée par un numéro et a un libellé (texte de la question), une réponse, un niveau de difficulté et un thème. Une même question peut apparaître dans plusieurs séries avec un numéro d'ordre pour chaque série. Une question appartient obligatoirement à au moins une série. On ne traite pas ici le détail des réponses. Lorsqu'un client assiste à une séance, après s'être connecté à l'application "CodeRoute", il répond à chacune des 40 questions posées et à la clôture de la séance il obtient son nombre de fautes (compris entre 0 et 40) une fois que l'application "CodeRoute" a calculé le nombre de fautes, elle l'affiche à chaque client. Lorsqu'un client a obtenu, au cours de ses quatre dernières séances un nombre de fautes inférieur ou égal à 5, le système prévient le moniteur d'auto-école qui peut alors autoriser le client à passer l'examen du code de la route à une date donnée (un seul examen pour une date donnée). La liste des sessions d'examen est affichée par l'application.

Les élèves ayant obtenu plus de 5 fautes à l'examen sont recalés et doivent assister de nouveau à des séances de code avant de pouvoir se représenter à l'examen. Le nombre de fautes obtenu à l'examen par chaque client est conservé.

