

Révision

IFT2255 – Génie logiciel

Louis-Edouard LAFONTANT



Dans le processus unifié, à chaque phase, nous travaillons sur un flux à la fois



Le processus unifié suit un modèle itératif et incrémental. Le projet est découpé en plusieurs phases au travers desquelles on itère sur plusieurs flux de développement (à différent degré).



Quels sont les facteurs qui influencent le choix du modèle de développement? Illustrez-les avec 3 modèles de développement.

Le modèle de développement est principalement dirigé par la distance entre les acteurs (communication) et les contraintes sur les ressources. Si la communication est difficile, le projet suivra un modèle linéaire ou incrémental; sinon il sera itératif. En revanche, si les ressources sont limités, on cherche à minimiser les activités d'implémentation et de déploiement. Ceci se traduira par un modèle linéaire ou itératif.





Le développement en cascade suit un modèle linéaire



Le développement en cascade organise les flux en séquence et ne favorise pas les retours arrière dans la séquence. Ceci reflète les caractéristiques d'un modèle linéaire.



Nommez et décrivez brièvement les flux de développement?

- 1. Exigences:** Dans ce flux, on acquiert une compréhension du domaine et on détermine les besoins du client.
- 2. Analyse:** Dans ce flux, on analyse et affine les exigences afin de déterminer les contraintes physiques et conceptuelles qui encadreront et guideront la conception et l'implémentation.
- 3. Conception:** Dans ce flux, on affine l'analyse et on produit un design implémentable par les programmeurs, sans favoriser un langage de programmation.
- 4. Implémentation:** Dans ce flux, on traduit le design dans le langage de programmation choisi. Ici, on itère surtout sur le design et le code.
- 5. Tests:** Dans ce flux, on évalue les artéfacts produits jusqu'à présent.
- 6. Déploiement:** Dans ce flux, on installe le logiciel sur le serveur cible et on transmet tous les artéfacts au client.



UML est un langage de programmation.

Justifiez votre réponse.



UML est un langage de modélisation composé de 14 types de diagrammes (structurels et comportementaux). Il peut être couplé à des techniques de génération de code pour créer des programmes (exécutable).



Quelles sont les caractéristiques d'une bonne conception?

Une bonne conception est une **conception flexible et modulaire supportant des mode d'utilisation et favorisant l'extension et la réutilisation des modules**. Ce design est obtenu par un **faible couplage et une forte cohésion**, où le couplage représente le degré d'interaction entre les modules et la cohésion le degré d'interaction au sein du module.





Je cherche à identifier les besoins de 10,000 utilisateurs.
Comment procéder?

Pour procéder à la cueillette des besoins, nous avons plusieurs méthodes: entrevue, sondage, observation. Vu le grand nombre d'utilisateurs, il est préférable de **créer un sondage** pour récolter un maximum de réponses. Nous pouvons aussi faire des **observations passives d'un petit groupe** (± 100) d'utilisateurs représentatif, pour obtenir un feedback personnel et plus révélateur.



Quelle est la différence entre acteur principal et acteur secondaire?

La réalisation d'un cas d'utilisation nécessite parfois l'intervention de plusieurs acteurs. Cependant, ils ne jouent pas tous le même rôle dans leurs interactions avec le système.

Acteur principal: Initie le CU; bénéficiaire principal/direct du CU

Acteur secondaire: Contribue à la réalisation du CU





Je développe un dispositif pour surveiller la ligne de but.

Nommer 3 risques et 3 besoins non-fonctionnels

Risques

1. But accordé ou refusé par erreur;
2. Réinitialisation durant un match;
3. Augmentation du cout de base du sport;
4. Coupure de budget alloué pour le développement;
5. Réduction de l'enthousiasme pour le sport

Besoins non-fonctionnels

- **Fiabilité:** Taux de fonctionnement de 99.99%;
- **Utilisabilité:** Facile à configurer;
- **Compatibilité:** Compatible avec les autres dispositifs utilisés dans le sport;
- **Portabilité:** Réutilisable pour divers buts pour divers sports;
- **Performance:** Délai de réponse < 0.1s



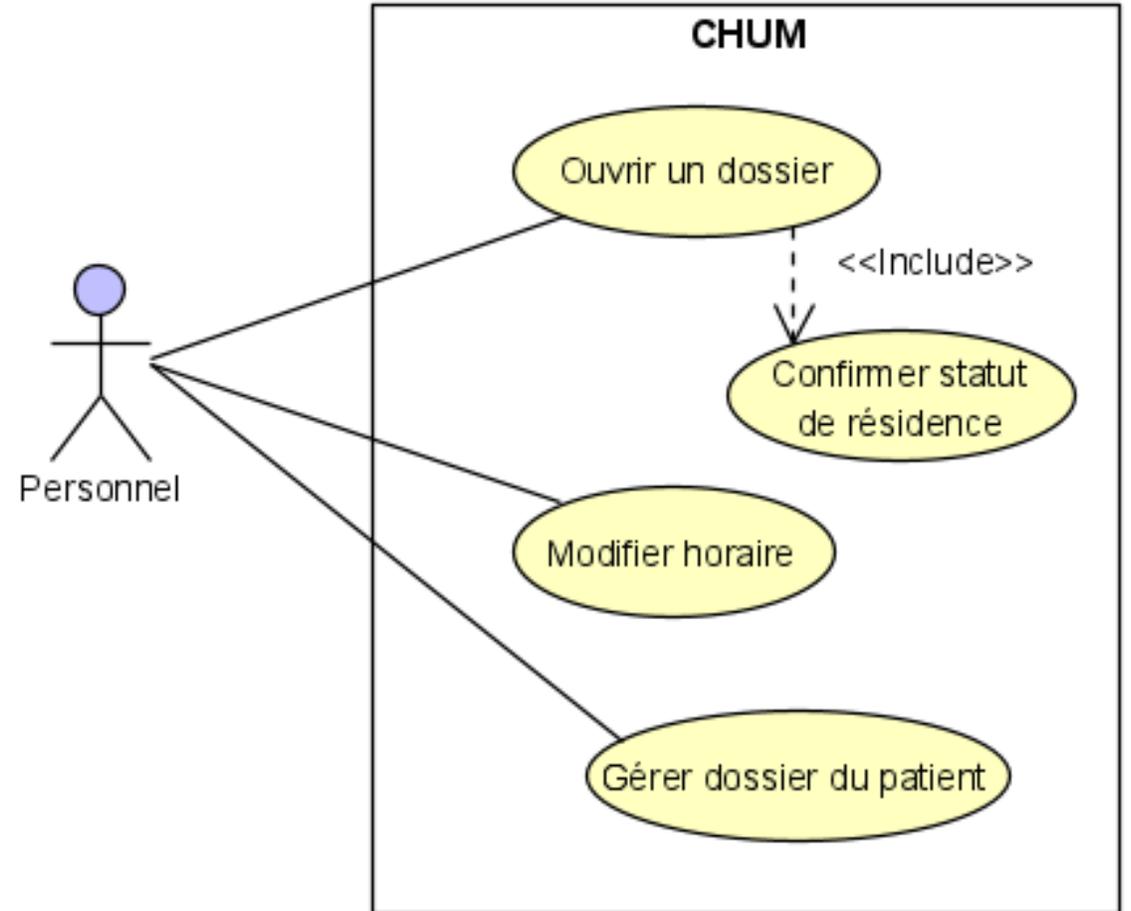
Les principes Agile visent surtout à inclure les actionnaires dans le développement



Les principes Agile visent à inclure les clients et utilisateurs qui sont généralement (dans nos pratiques passées) mis à l'écart.



Quel est le problème avec ce diagramme?



1. La relation *include* nécessite au minimum 2 CUs utilisant un troisième. Donc, sans autre cas, il faudrait jumeler *Ouvrir un dossier* et *Confirmer statut de résidence*.
2. Le cas *Gérer dossier du patient* est trop général. Il faudrait le décomposer en cas plus spécifique comme *Modifier dossier du patient* et *Archiver dossier du patient*.
3. Certains cas t.q. *Gérer dossier du patient* nécessitent probablement l'intervention du patient comme acteur secondaire. Il faudrait donc l'ajouter et l'associer à ces cas



Un scénario a toujours des pré/post conditions



Un scénario a toujours un but, des acteurs et un scénario principal. Les scénarios alternatifs ainsi que les pré/post conditions sont optionnelles. Les préconditions stipulent tout ce qui doit être absolument vrai pour entamer scénario principal. Les postconditions décrivent dans quel état le système doit absolument se trouver à la fin du scénario principal.



Je développe, pour la ville, une application capable de localiser la poubelle la plus proche. Elle permet de distinguer le type de poubelle et leur capacité.

Nommez 3 besoins fonctionnels et 3 besoins non-fonctionnels pour cette application.

Besoins fonctionnels

1. Trouver la position de l'utilisateur (location)
2. Trouver les poubelles dans un rayon donné (25m, 50m)
3. Filtrer les poubelles par type (plastique, déchet) et capacité
4. Vérifier la capacité des poubelles (service de ramassage)
5. Produire un rapport sur l'usage des poubelles (efficacité)

Besoins non-fonctionnels

1. **Fiabilité:** 90% des poubelles suggérées doivent être accessibles
2. **Utilisabilité:** Facile à utiliser et configurer pour tous les âges
3. **Compatibilité:** Compatible avec les autres systèmes de la ville.
4. **Portabilité:** L'application est supporté par les systèmes d'exploitation populaires (ex: Android, iOS)
5. **Performance:** Délai de réponse < 0.1s



Quelles modifications faut-il apporter au diagramme pour supporter les cas suivants?

- Si les données d'inscription d'un nouvel étudiant ne sont pas valides, le flux est interrompu.
- Suite à l'analyse du profil de l'étudiant, le profil est ajouté à la liste des étudiants internationaux ou à la liste des étudiants réguliers.
- Le flux se termine lorsque l'inscription est ajoutée au dossier et que le profil de l'étudiant est catégorisé (dans une des listes).

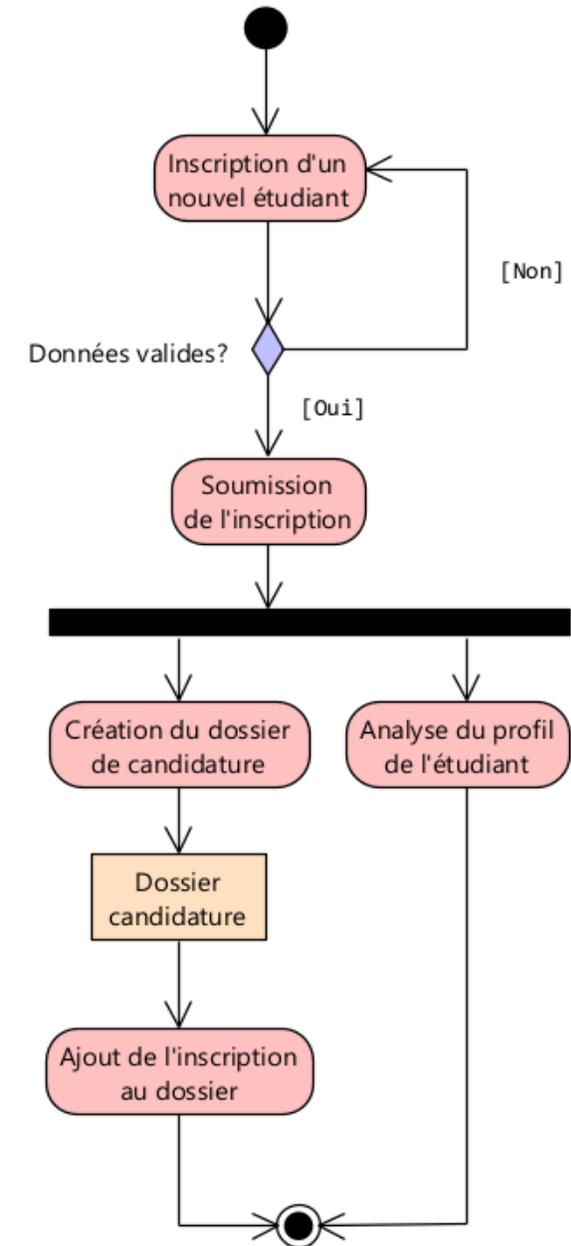
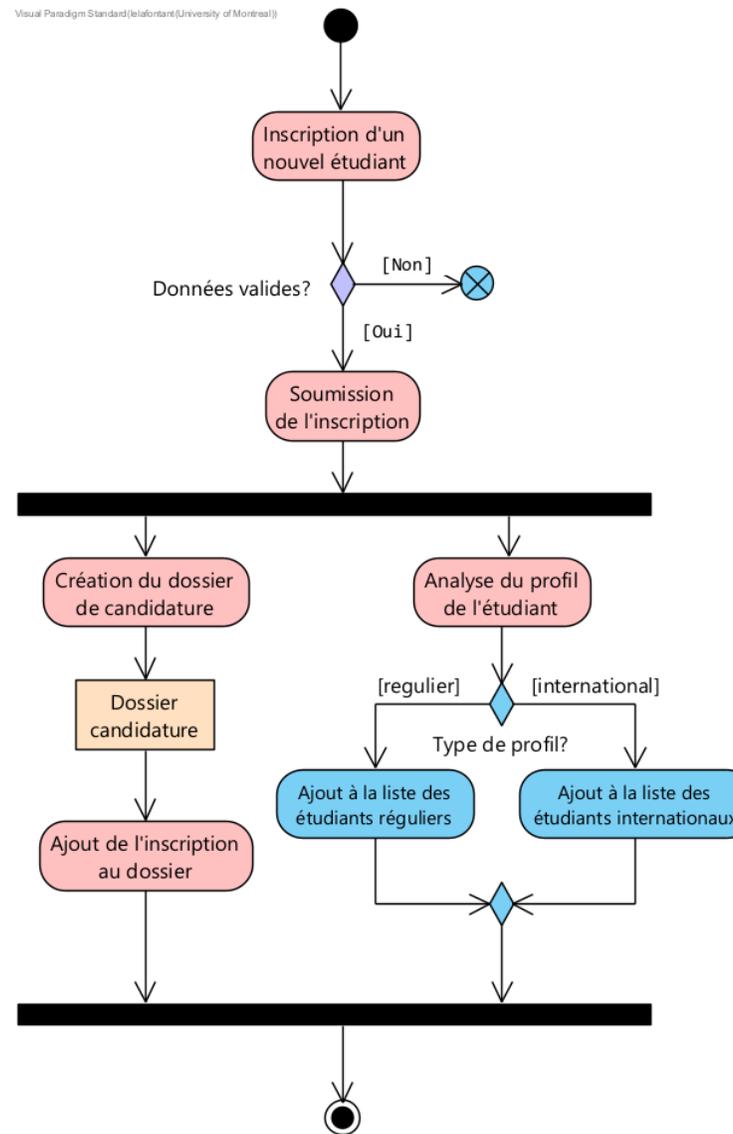


Diagramme modifié



Décrivez les classes intervenantes dans ce diagramme de séquence



5 min

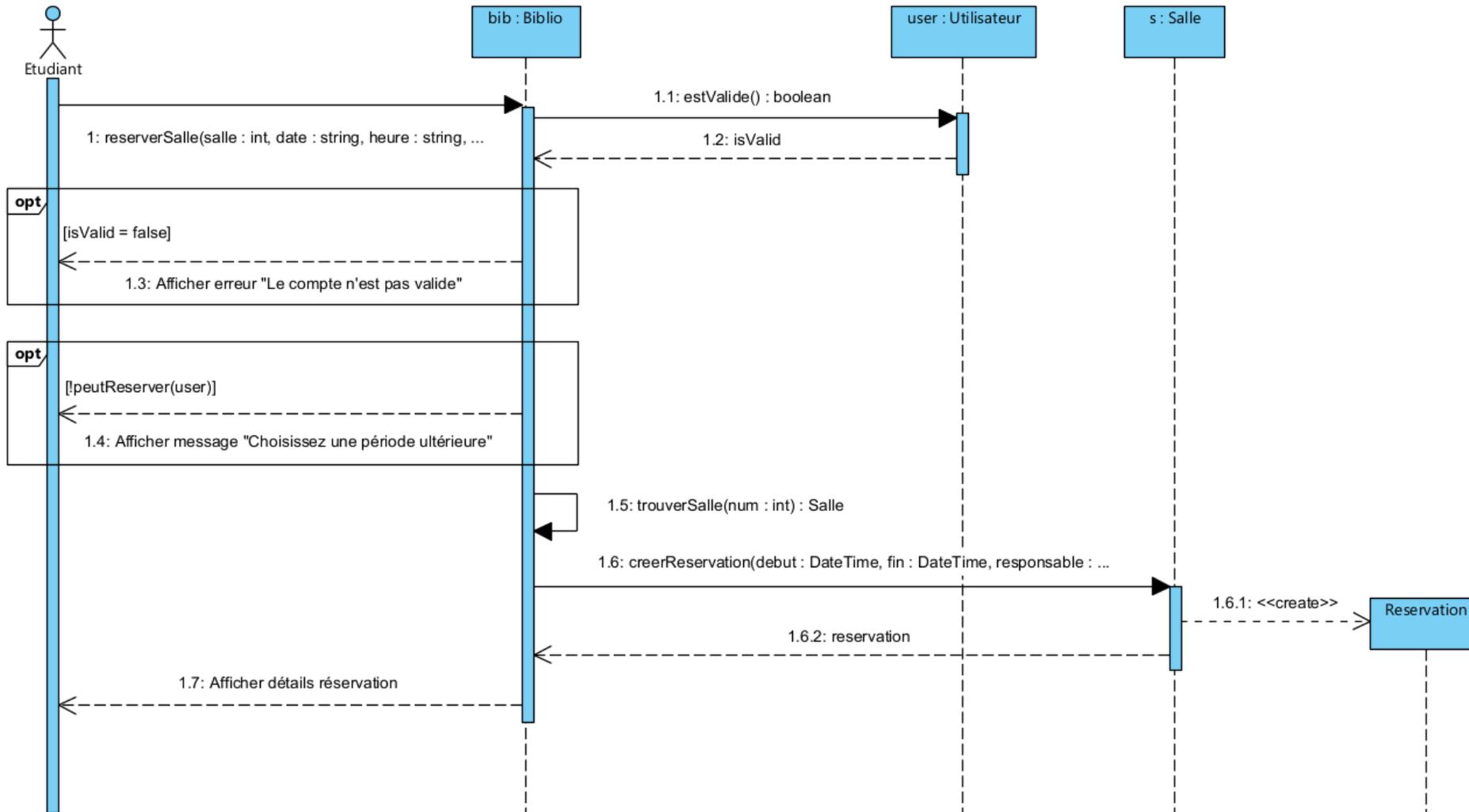
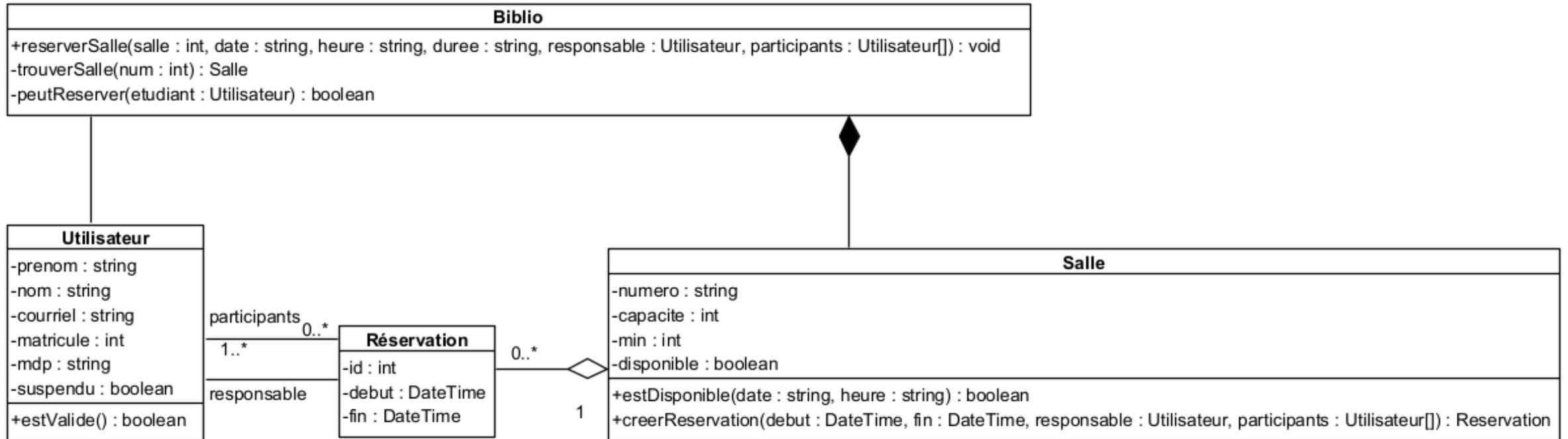


Diagramme de classes (minimal)

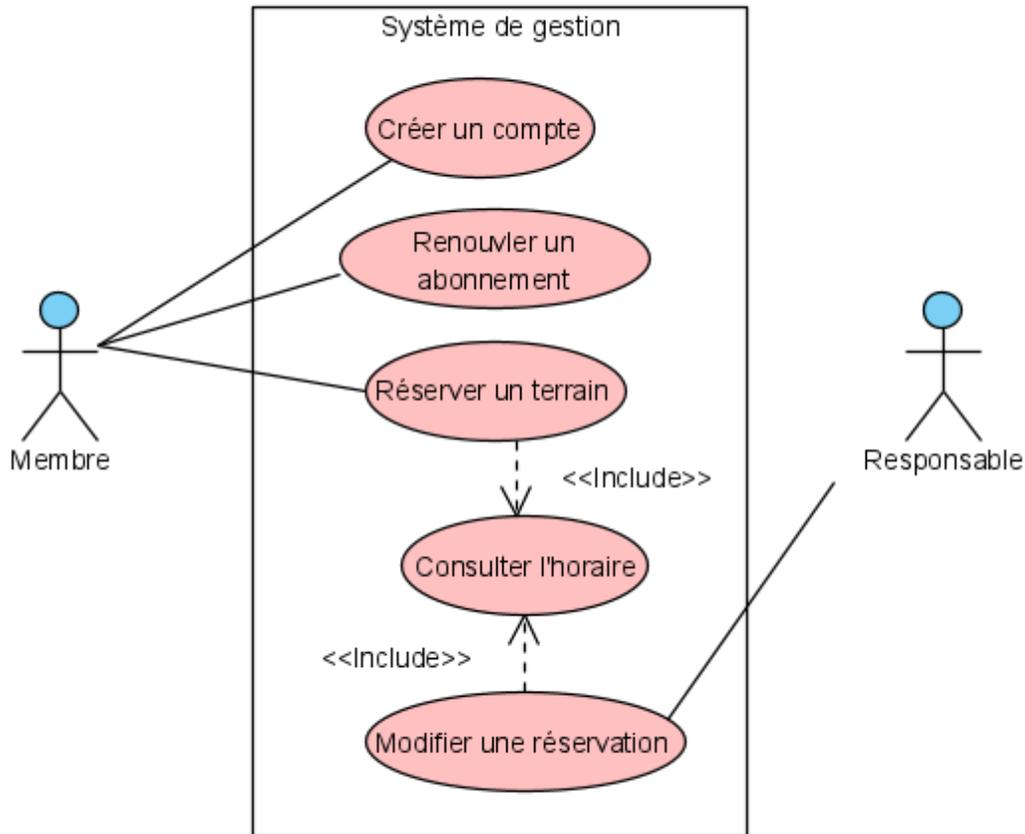




Construisez le diagramme de CU et le diagramme de classes

Considérons un système de gestion des terrains d'un club de badminton. Pour réserver un terrain, un membre du club donne son code de membre. Si son abonnement n'est plus valable, il doit le renouveler avant de procéder à la réservation, en payant les frais de membre. Pour faire la réservation, il peut commencer par demander l'horaire disponible pour la semaine courante ou bien indiquer directement le jour et l'heure à laquelle il veut jouer. Lorsqu'il trouve un horaire qui lui convient, il confirme son choix. Si un terrain à nettoyer ou à réserver pour une compétition est déjà réservé par un membre du club, un responsable peut réorganiser les réservations en plaçant les réservations des membres dans d'autres terrains. Un terrain est identifié par un numéro et la salle où se trouve le terrain.

Modélisation: Cas d'utilisation + scénarios



CU: Réserver un terrain

Acteurs: Membre (principal)

Précondition: Le membre est connecté au système

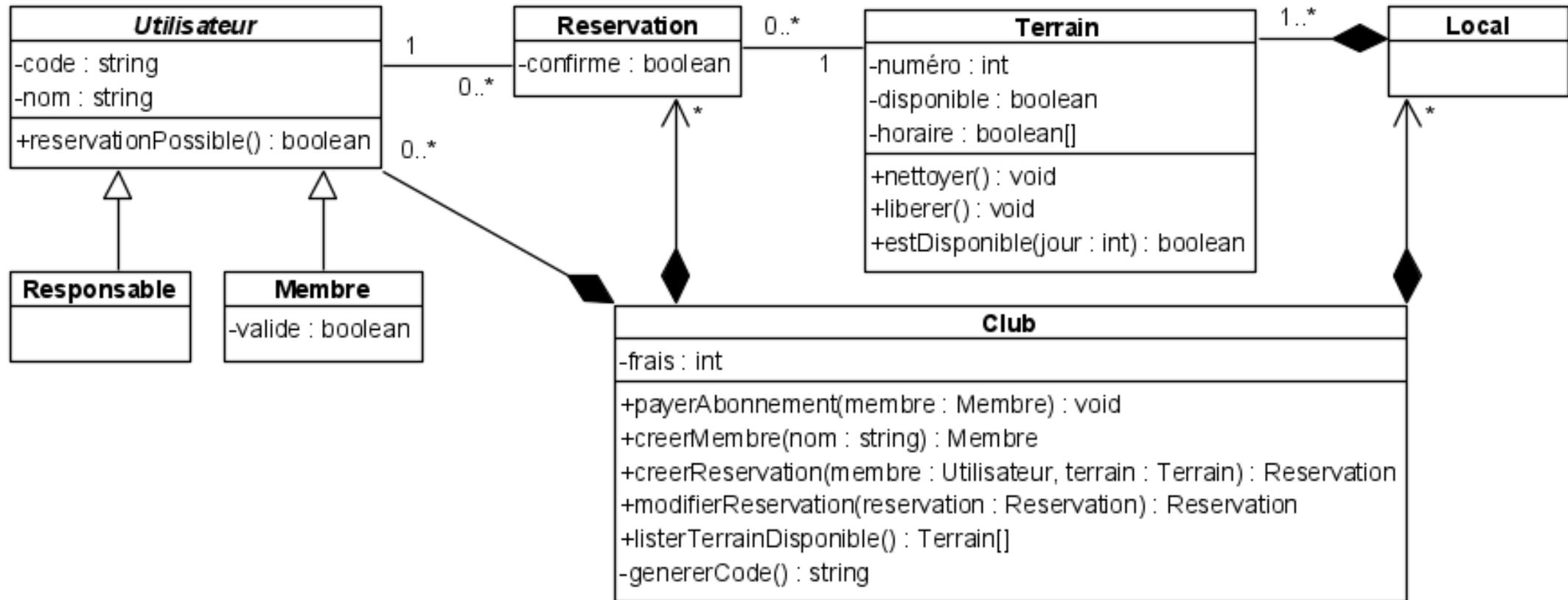
Scénario principal

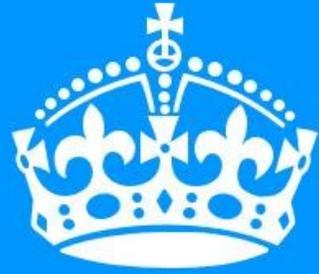
1. Le membre choisit l'option « Réserver un terrain »
2. Le système demande le code d'accès
3. Le membre fournit son code de membre
4. Le système valide le code du membre
5. Le membre consulte les plages disponibles
6. Le membre fournit la date et l'heure de la réservation
7. Le système valide le choix du membre
8. Le système affiche le numéro du terrain réservé

Scénario alternatif

- 5a. L'abonnement du membre n'est pas valable
- 5a.1 Le système affiche le message d'erreur et invite l'utilisateur à renouveler son abonnement
 - 5a.2 Le scénario se termine

Modélisation: Diagramme de classes





**KEEP
CALM
AND
ACE YOUR
EXAMS**