

# Rapport Itération numéro i

---

## Identification des membres de l'équipe

Assurez-vous d'utiliser toujours le même compte GitHub pour accéder à ce projet.

### Membre 1

- Anna Andersen
- anna.andersen.1@ens.etsmtl.ca
- AQ80740
- aandersen-mal

### Membre 2

- Thomas Beetz
- Thomas.beetz.1@ens.etsmtl.ca
- AP59120
- Thomas-Beetz-Knowledge

### Membre 3

- Nick-Karl Chao
- nick-karl.chao.1@ens.etsmtl.ca
- AQ18340
- n3chao

### Membre 4

- Lucas Schwing
- lucas.schwing.1@ens.etsmtl.ca
- AR77560
- Lukip2603

### Membre 5

- Shi Yao Wang
- shi-yao.wang.1@ens.etsmtl.ca
- AQ87340
- syw1-art

## Exigences

Liste des exigences et personnes responsables de celles-ci.

Exigence	Responsable
CU01a	Anna Andersen, Lucas Schwing

Exigence	Responsable
CU01b	Nick-Karl Chao, Shi Yao Wang
CU02a	Anna Andersen, Lucas Schwing, Thomas Beetz

Modèle du domaine (MDD)

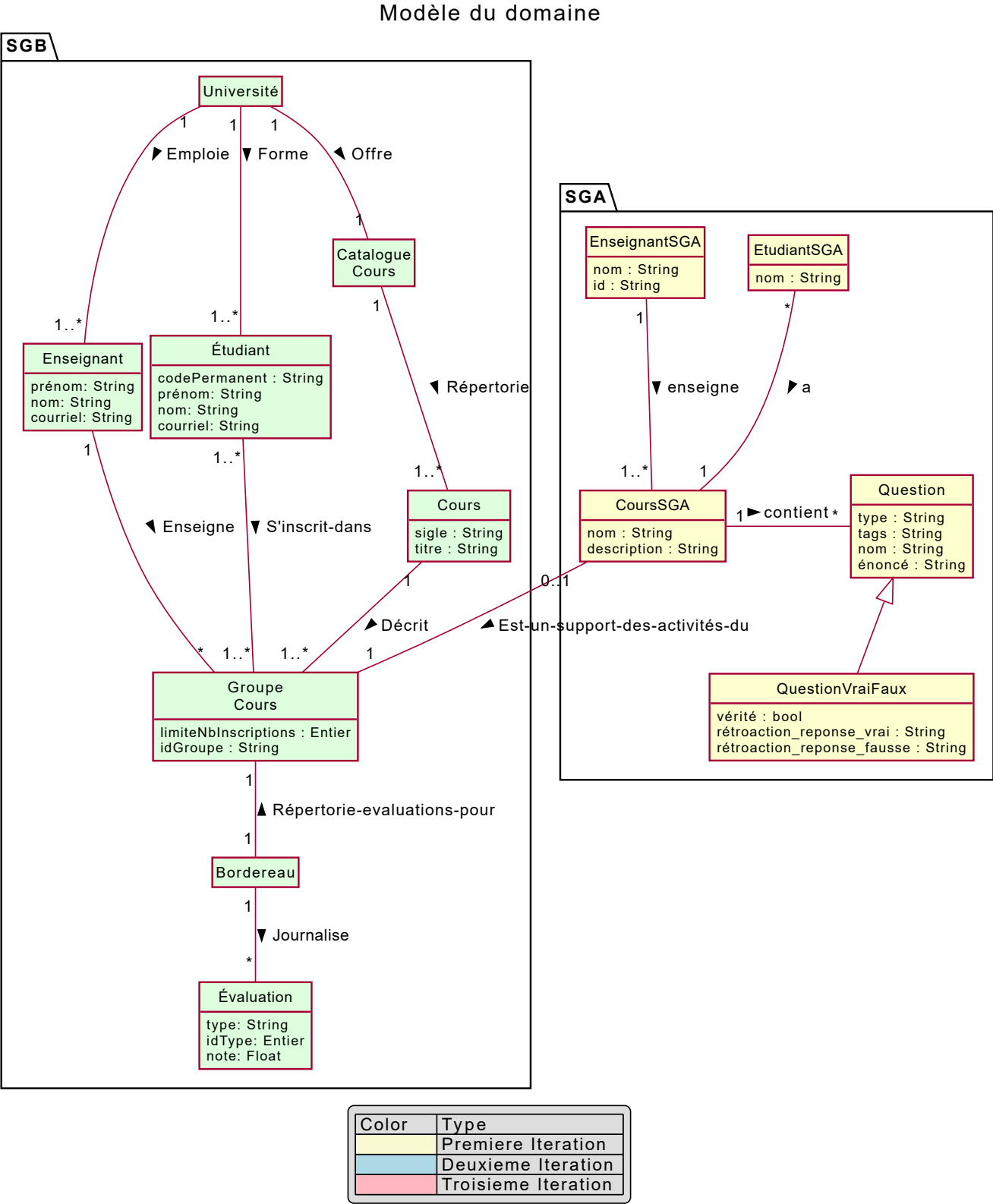
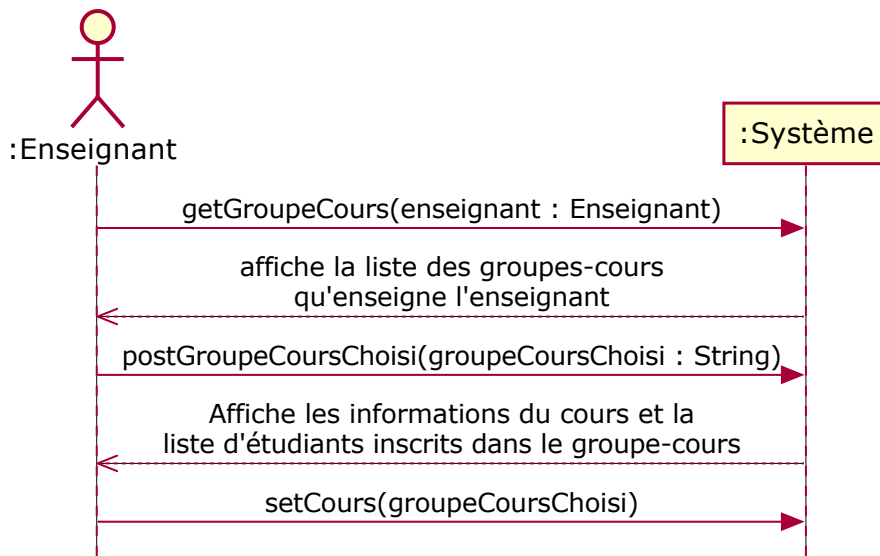


Diagramme de séquence système (DSS)

Un seul DSS sera choisi et corrigé par l'auxiliaire d'enseignement

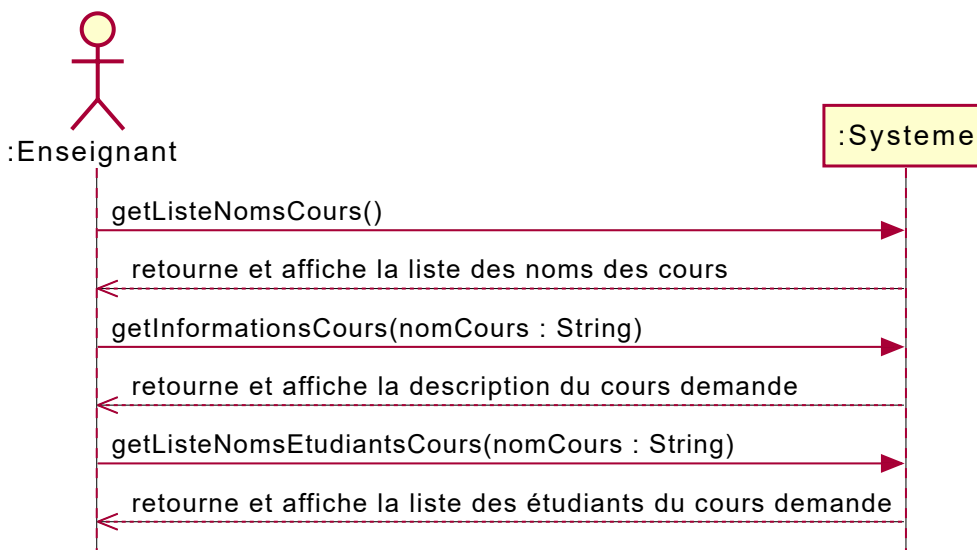
CU01a

### DSS CU01a - Ajouter cours



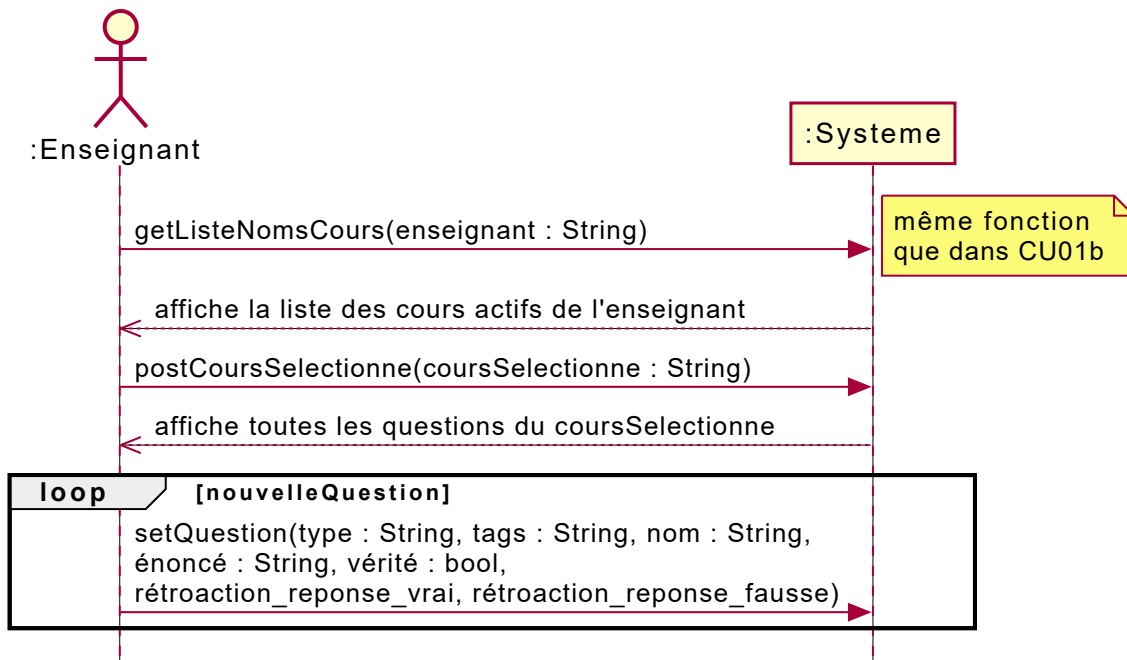
CU01b

### DSS CU01b - Récupérer cours



CU02a

## DSS CU02a



## Contrats

Si vous avez choisi un cas d'utilisation nécessitant un contrat, il faut le mettre dans cette section. Note: même s'il y a plusieurs contrats, un seul contrat sera choisi et corrigé par l'auxiliaire d'enseignement

CU01a

## Opération: setCours(groupeCoursChoisi)

Postconditions:

- une nouvelle instance c de Cours a été créée
- c a été associée a l'enseignant du cours
- c a été associé aux groupes d'étudiants inscrits

CU02a

## Opération: setQuestion

Postconditions:

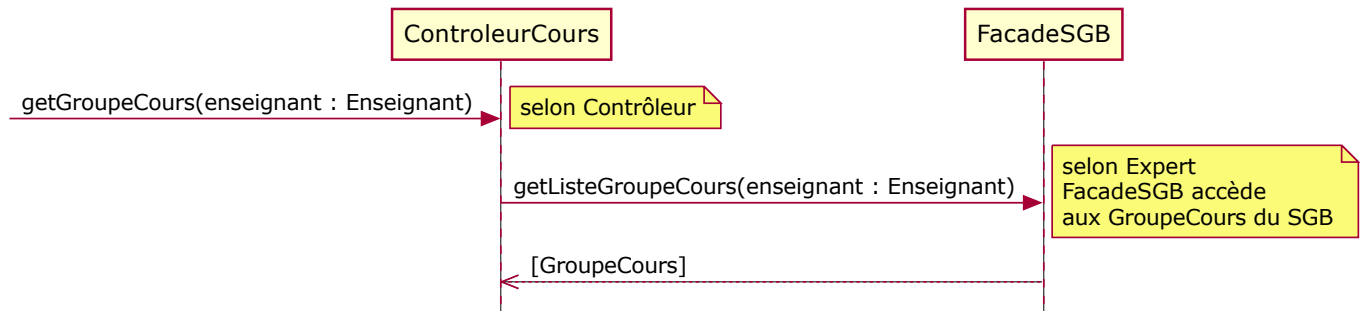
- une nouvelle instance q de Question a été créée
- q a été associée au cours

## Réalisation de cas d'utilisation (RDCU)

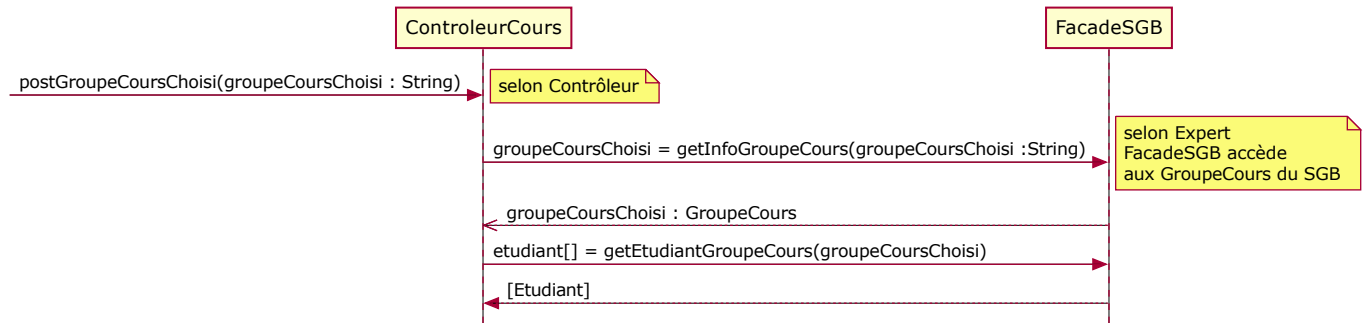
Chaque cas d'utilisation nécessite une RDCU. Note: une seule RDCU sera choisie et corrigée par l'auxiliaire d'enseignement

CU01a

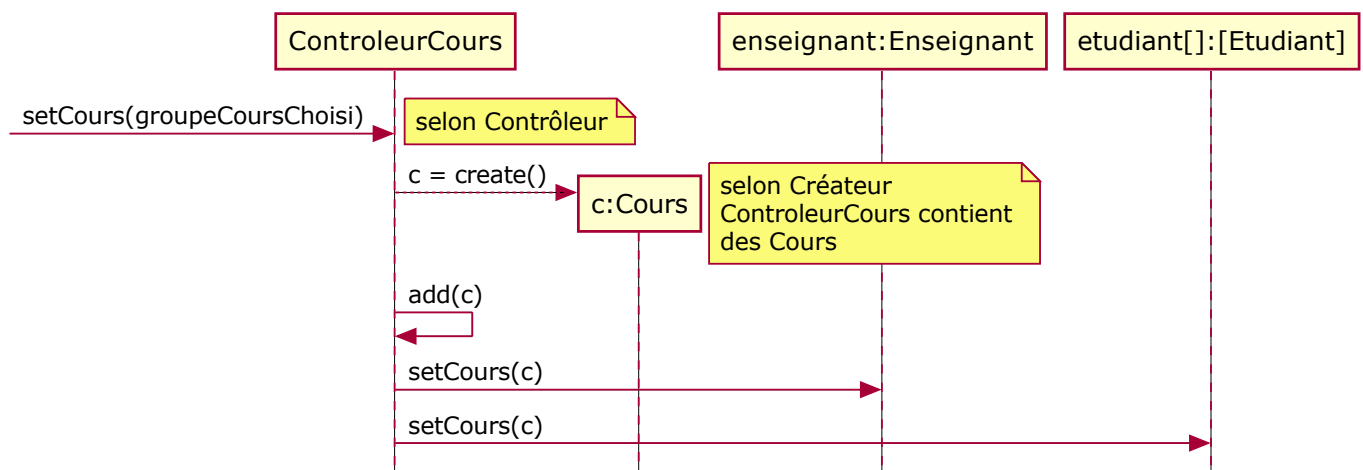
## RDCU getGroupeCours(enseignant : Enseignant)



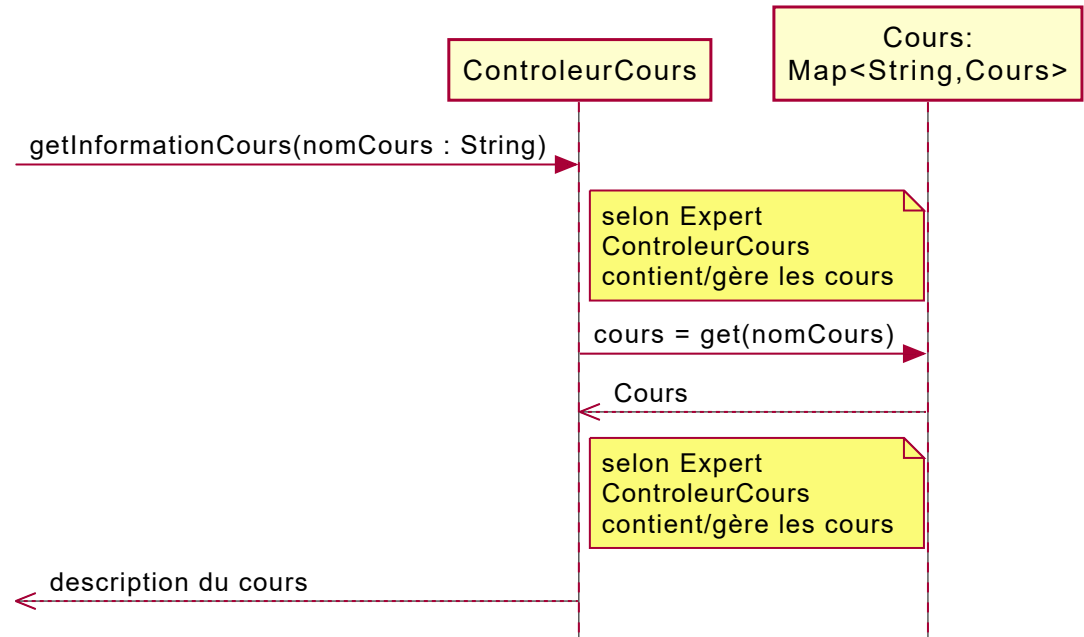
## RDCU postGroupeCoursChoisi(groupeCoursChoisi : String)



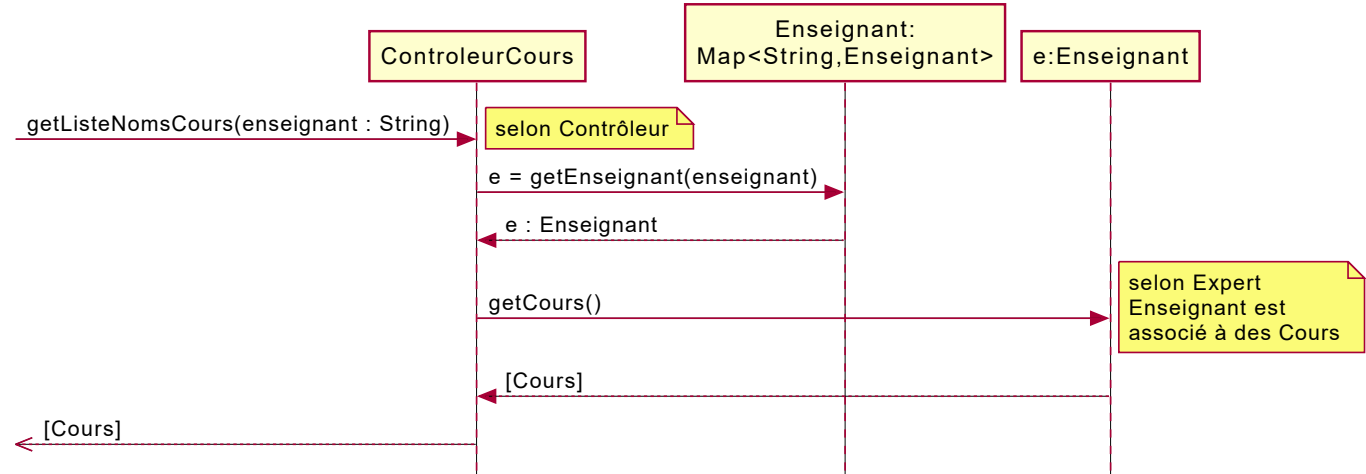
## RDCU setCours(groupeCoursChoisi)



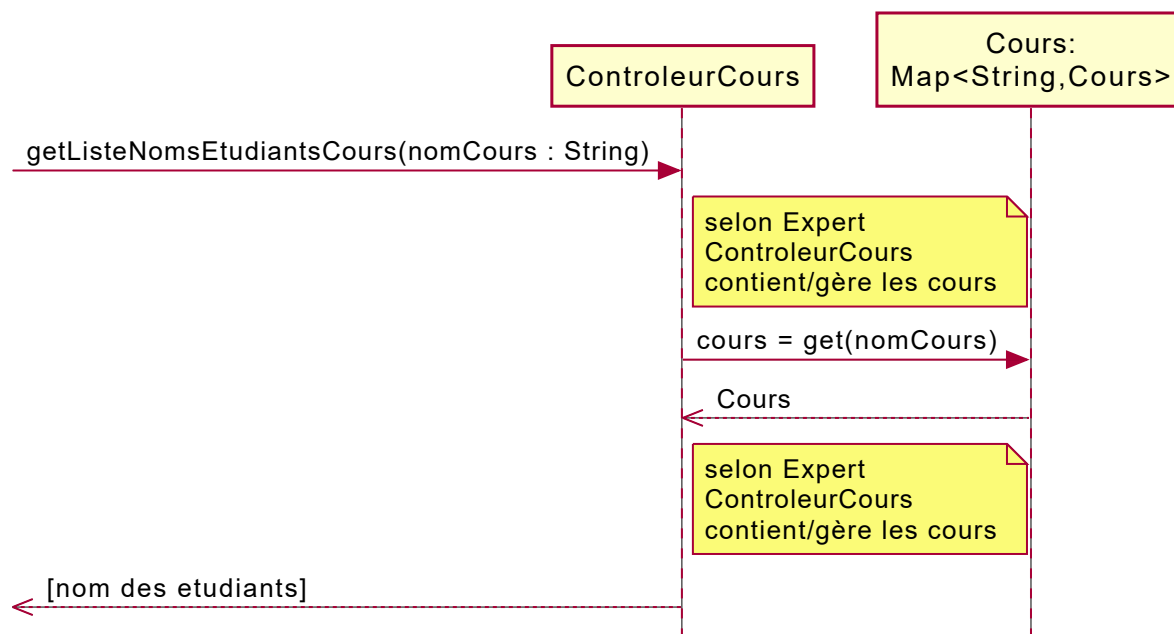
RDCU CU01b



RDCU CU01b

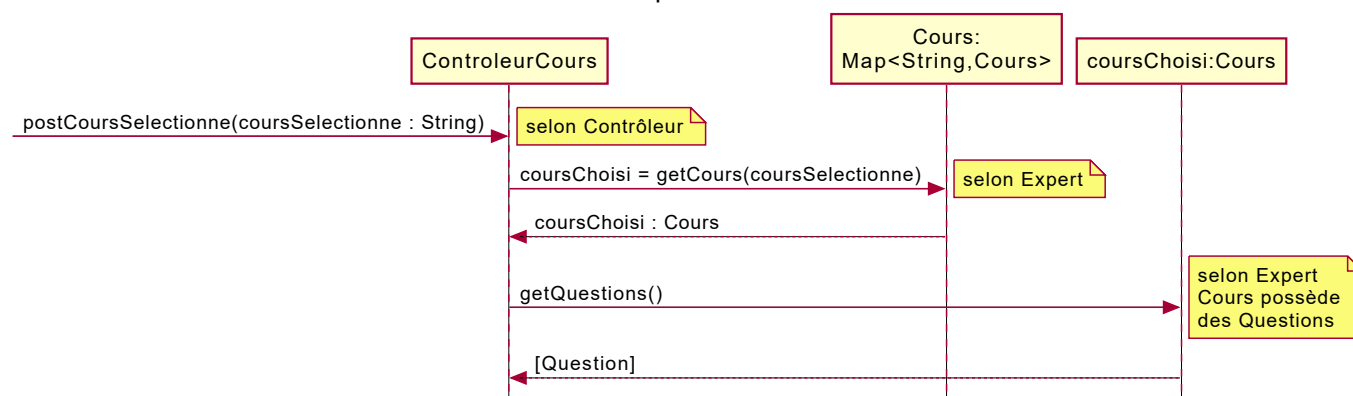


## RDCU CU01b

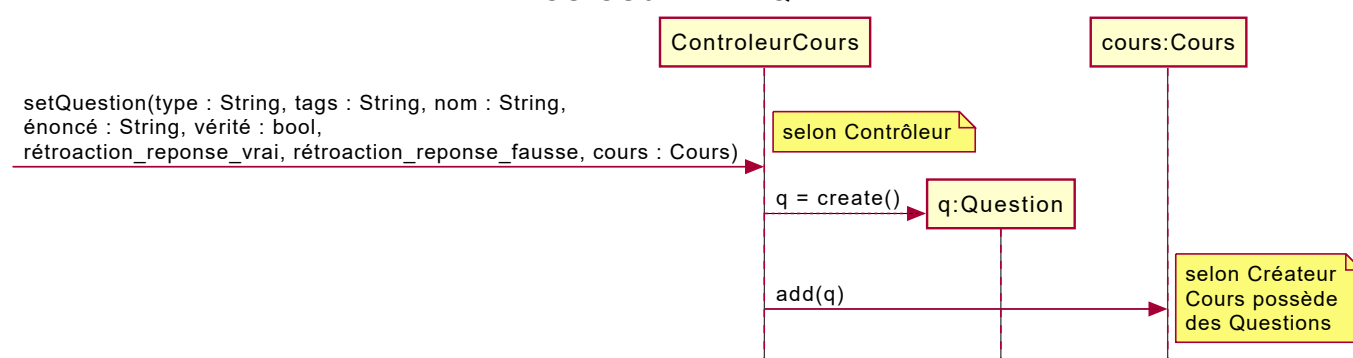


## CU02a

## RDCU CU02a - postCoursSelectionne



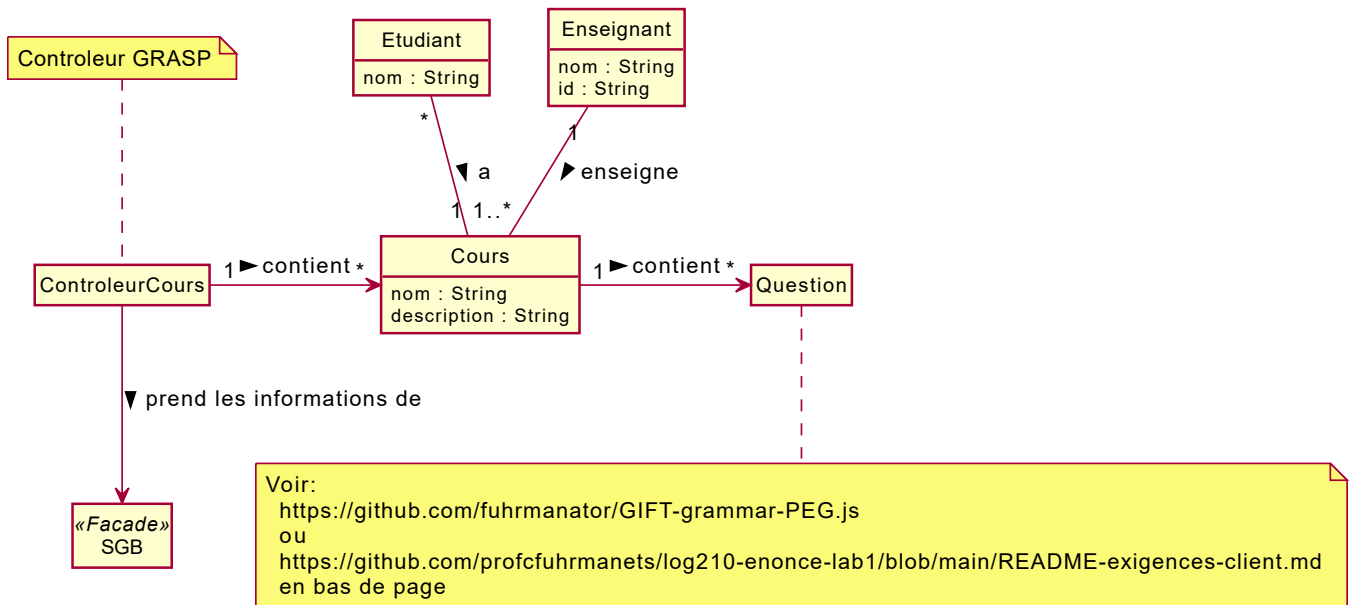
## RDCU CU02a - setQuestion



## Diagramme de classe logicielle (DCL)

Facultatif, mais fortement suggéré Ce diagramme vous aidera à planifier l'ordre d'implémentation des classes. Très utile lorsqu'on utilise TDD.

## Diagramme de classe logicielle



## Vérification finale

- ☒ Vous avez un seul MDD
  - ☒ Vous avez mis un verbe à chaque association
  - ☒ Chaque association a une multiplicité
- ☒ Vous avez un DSS par cas d'utilisation
  - ☒ Chaque DSS a un titre
  - ☒ Chaque opération synchrone a un retour d'opération
  - ☒ L'utilisation d'une boucle (LOOP) est justifiée par les exigences
- ☒ Vous avez autant de contrats que d'opérations système (pour les cas d'utilisation nécessitant des contrats)
  - ☒ Les postconditions des contrats sont écrites au passé
- ☒ Vous avez autant de RDCU que d'opérations système
  - ☒ Chaque décision de conception (affectation de responsabilité) est identifiée et surtout **justifiée** (par un GRASP ou autre heuristique)
  - ☐ Votre code source (implémentation) est cohérent avec la RDCU (ce n'est pas juste un diagramme)
- ☒ Vous avez un seul diagramme de classes
- ☒ Vous avez remis la version PDF de ce document dans votre répertoire
- ☒ Vous avez regardé cette petite présentation pour l'architecture en couche et avez appliqué ces concepts