

# Rapport itération 1 - équipe 3

---

LOG210-année-session-groupe (ex. LOG210-2020-été-g01)

## Coéquipiers

- Bédard, Tommy, AQ13400
- Abdelli, Pierre Amar, AQ48850
- Bewa, Lionel, AQ22600
- El-Safady, Sobhi, AP94320
- Korchi, Zakaria , AQ25210

## Chargés de laboratoire

- Valère K. Fami

# Grille de correction

---

► Cliquez ici pour lire les consignes obligatoires

- Tous vos diagrammes doivent être faits avec <https://plantuml.com/fr/>
- Les diagrammes doivent être visibles dans ce rapport
- Supprimer les textes explicatifs du gabarit (sauf ces consignes-ci)
- Vous devez exporter ce fichier en format PDF et l'ajouter dans votre dépôt

**⚠ Un travail qui contient trop d'erreurs fréquentes peut être refusé par le chargé de laboratoire. L'équipe peut reprendre le travail avec une pénalité de 25%. Vérifiez attentivement la grille de correction et les notes de cours**

Le travail sera noté selon la [grille de correction des rapports](#). Pour y accéder, vous devez utiliser votre adresse [etsmtl.net](mailto:etsmtl.net).

# Introduction

---

Dans cette première itération, l'objectif était de concevoir la structure générale du Système de gestion des apprentissages. L'évaluation du modèle du domaine, établir les diagrammes de séquences systèmes reliés au cas d'utilisation ainsi que leur diagramme de cas d'utilisation. Finalement, l'utilisation des analyse conceptuelles développer dans la première partie pour réaliser l'implémentation des cas d'utilisations CU01 et CU02.

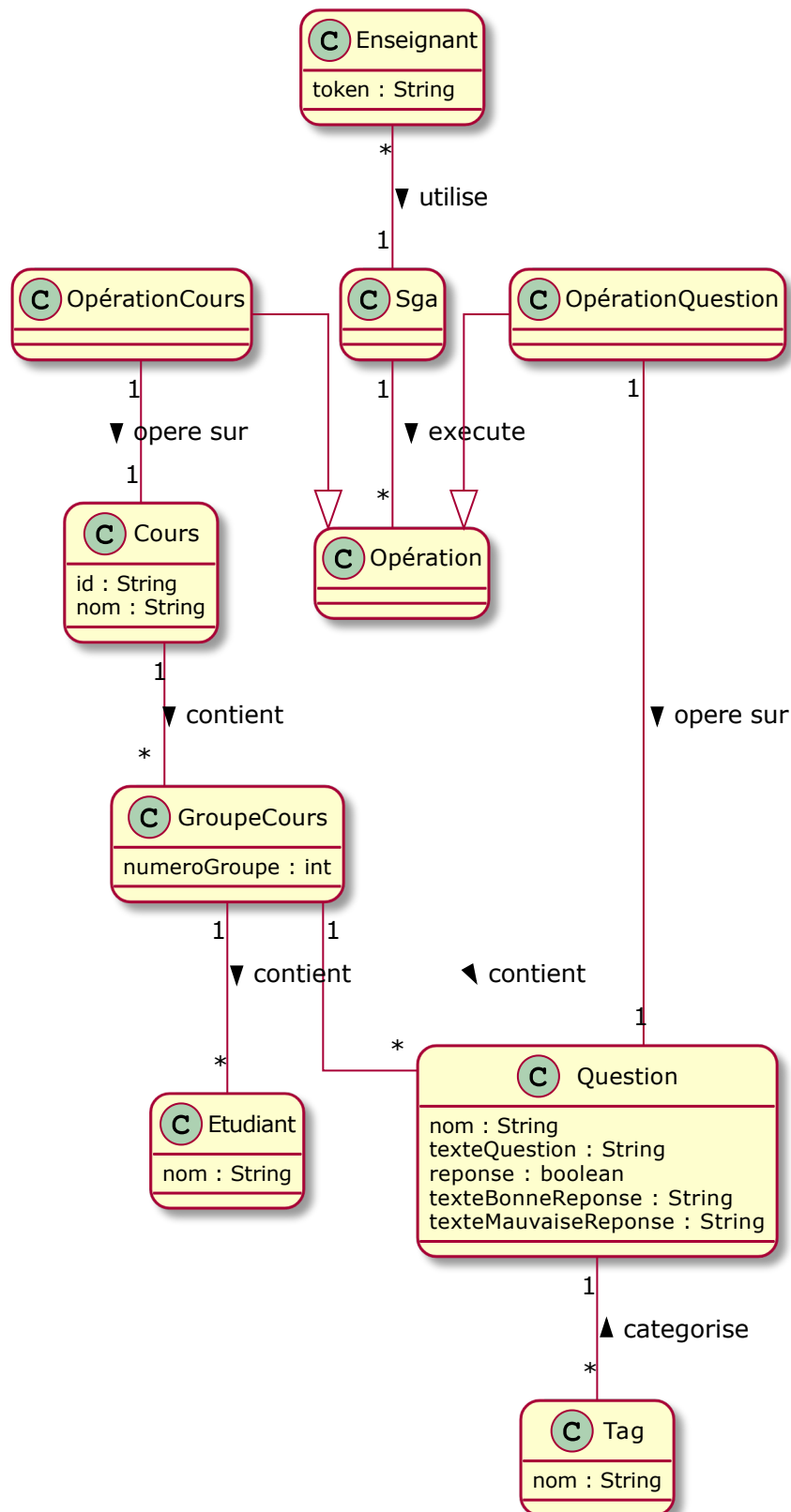
# Modèle du domaine

---

Le MDD est cumulatif : vous devez y ajouter des éléments à chaque itération (ou corriger les erreurs), selon la portée (et votre meilleure compréhension du problème) visée par votre solution. Utilisez une légende dans le MDD pour indiquer la couleur de chaque itération afin de faire ressortir les

changements. Voir les stéréotypes personnalisés : <https://plantuml.com/fr/class-diagram> et comment faire une légende avec couleurs en PlantUML.

## Modèle du domaine (adapté au Cas d'utilisation ajouter et récupérer )



### CU01a - Ajouter cours

**Acteur principal:** Enseignant

**Préconditions:**

- L'enseignant est authentifié. SGA transfère l'information à SGB pour l'authentification. SGB retourne un token que SGA retourne à l'utilisateur. L'utilisateur doit nécessairement utiliser ce token pour faire ses requêtes à SGA..

**Garanties en cas de succès (postconditions):**

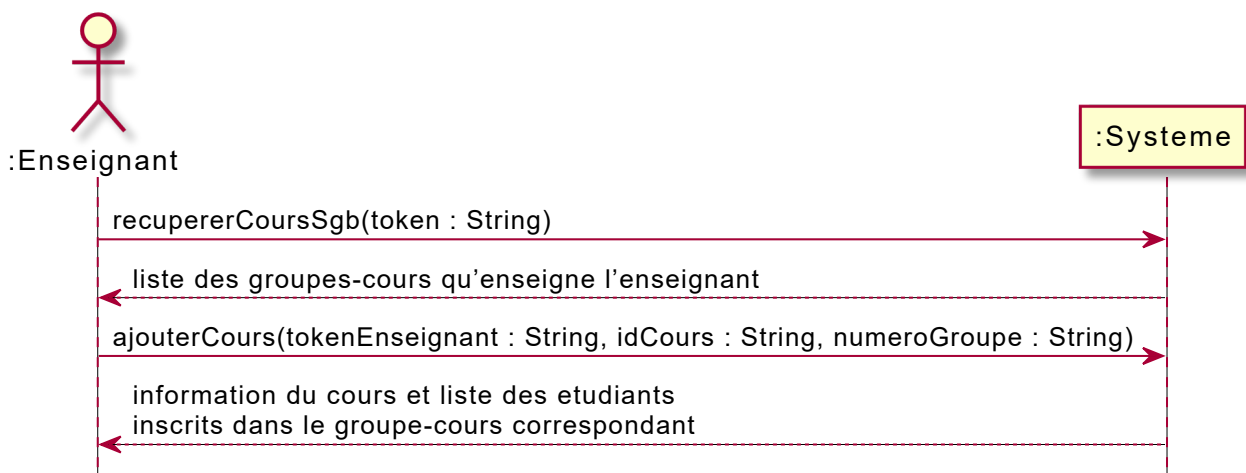
- Un nouveau cours est créé.
- L'enseignant est associé au cours
- Les étudiants inscrits dans le groupe-cours(SGB) sont associés au cours

**Scénario principal (succès):**

1. L'enseignant demande de créer un nouveau cours.
2. Le système affiche la liste des groupes-cours qu'enseigne l'enseignant (l'information provient du SGB).
3. L'enseignant choisit un groupe-cours dans la liste.
4. Le système affiche l'information du cours et affiche la liste des étudiants inscrits dans le groupe-cours correspondant (l'information provient du SGB).

**Extensions (ou scénarios alternatifs):**

- 3a. Un cours correspondant au groupe-cours sélectionné existe déjà.
1. Le système signale l'erreur et rejette la saisie.

**DSS CU01a - Ajouter cours****DSS Ajouter Cours****recupererCoursSgb(token : String)**

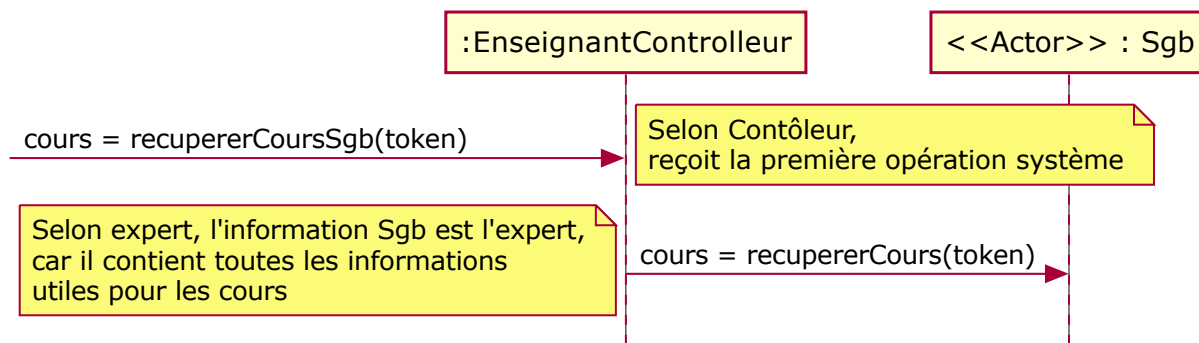
Si une opération a déjà été documentée dans un autre DSS, ajoutez un hyperlien vers le contrat précédent.

**Contrat d'opération***PostCondition*

- aucun

## RDCU

### RDCU. recupererCoursSgb



`ajouterCours(tokenEnseignant : String, idCours : String, numeroGroupe : String)`

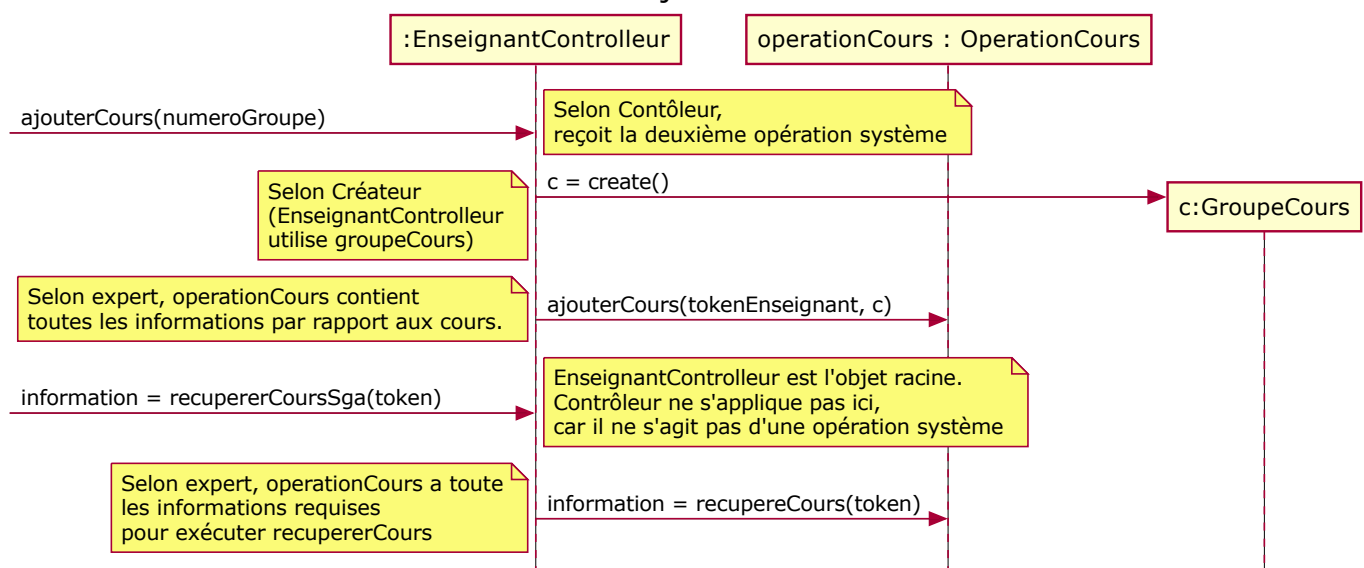
### Contrat d'opération

#### PostCondition

- Un nouveau cours est créé s'il n'existe pas déjà
- Un nouveau GroupeCours est créé
- le GroupeCours est associé au Cours
- L'enseignant est associé au cours
- Les étudiants inscrits dans le groupe-cours(SGB) sont associés au cours

## RDCU

### RDCU. ajouter cours



## CU01b - Récupérer cours

**Acteur principal:** Enseignant

**Préconditions:**

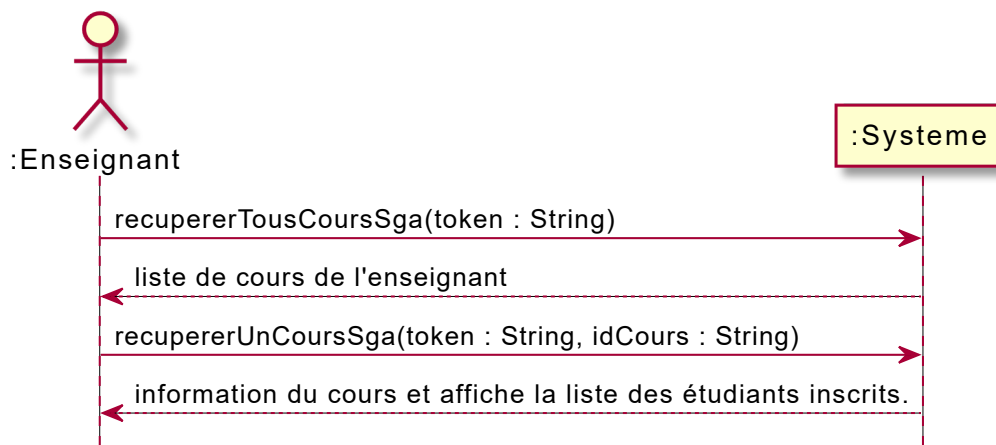
- L'enseignant est authentifié.

**Garanties en cas de succès (postconditions):**

- Aucun

**Scénario principal (succès):**

1. L'enseignant demande la liste de ses cours.
2. Le système affiche la liste de ses cours.
3. L'enseignant demande les détails d'un cours.
4. Le système affiche l'information du cours et affiche la liste des étudiants inscrits.

**DSS CU01b - Récupérer cours****DSS récupérer cours**

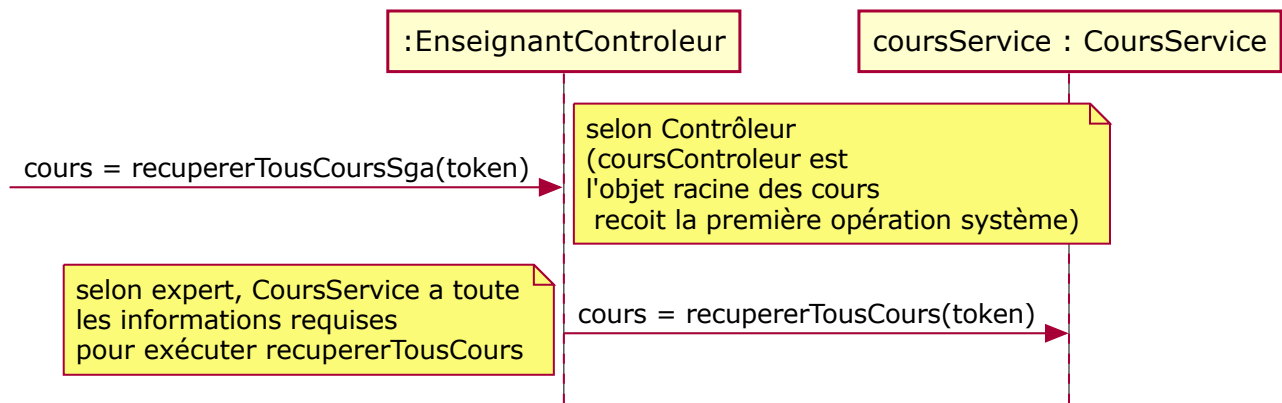
`recupererTousCoursSga(token : String)`

**Contrat d'opération***PostCondition*

- Aucun

**RDCU**

## RDCU pour récupérer tous les cours créés d'un enseignant



`recupererUnCoursSga(token : String, idCours : String)`

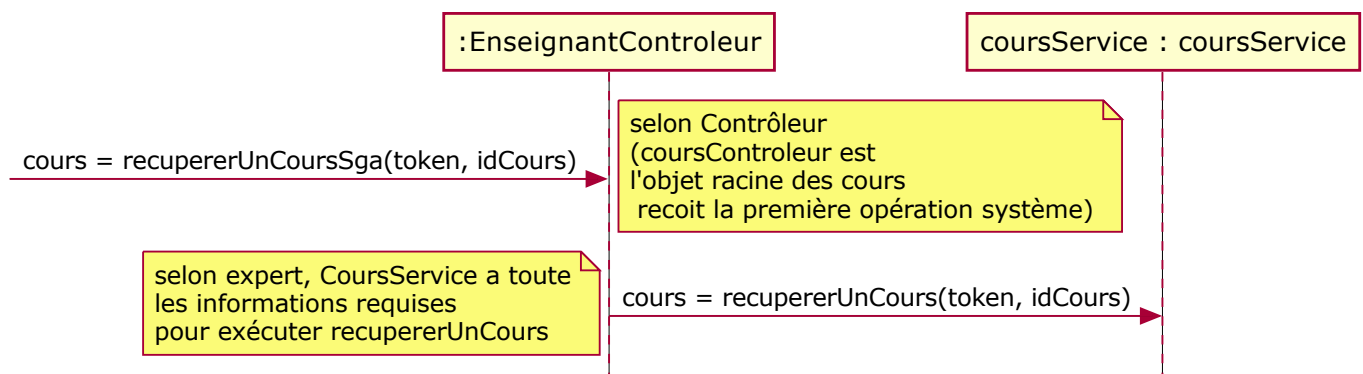
### Contrat d'opération

*PostCondition*

- Aucun

### RDCU

## RDCU pour récupérer un cours créé d'un enseignant



## CU01c - Retirer cours

**Acteur principal:** Enseignant

### Préconditions:

- L'enseignant est authentifié.

### Garanties en cas de succès (postconditions):

- Le cours n'existe plus dans le système SGA

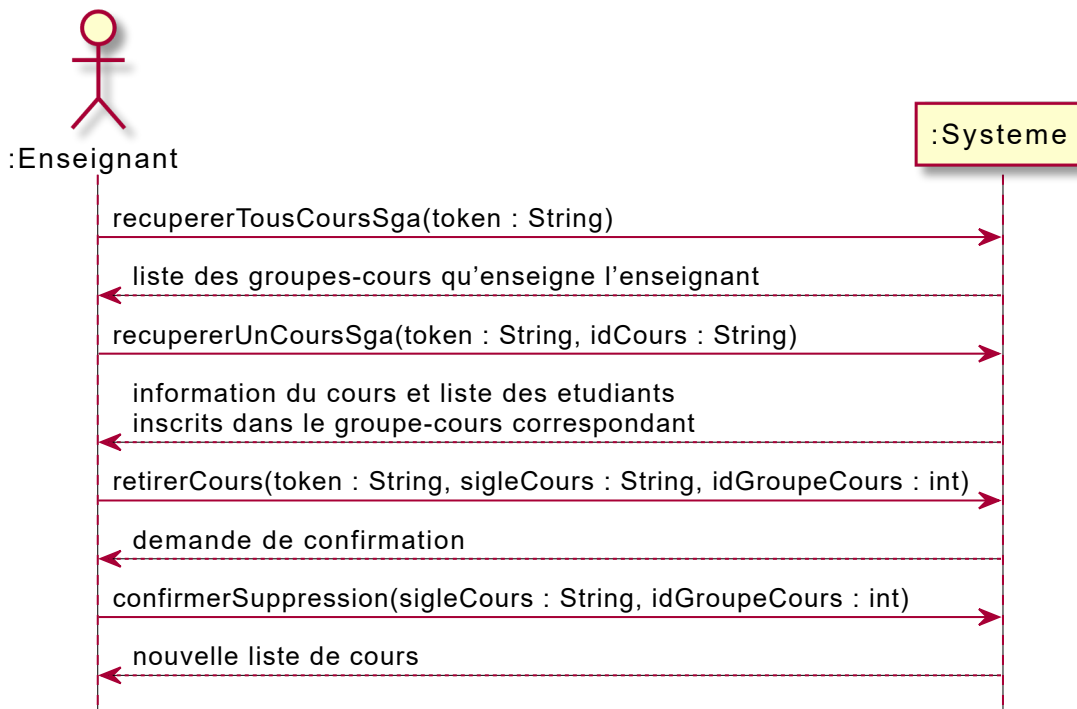
### Scénario principal (succès):

1. L'enseignant demande la liste de ses cours.
2. Le système affiche la liste de ses cours.
3. L'enseignant demande les détails d'un cours.

4. Le système affiche l'information du cours et affiche la liste des étudiants inscrits.
5. L'enseignant demande de supprimer le cours.
6. Le système demande une confirmation pour supprimer le cours.
7. L'enseignant confirme.
8. Le système supprime le cours et affiche la nouvelle liste de cours.

## DSS CU01c - Retirer cours

### DSS retirer cours



`recupererTousCoursSga(token : String)`

`recupererTousCoursSga`

`recupererUnCoursSga(token : String, idCours : String)`

`recupererUnCoursSga`

`retirerCours(token : String, idCours : String)`

#### Contrat d'opération

*PostCondition*

- Aucun

#### RDCU

## RDCU pour supprimerUneQuestionSga

:EnseignantControleur

confirmation = supprimerUneQuestionSga(sigleCours : String, idGroupeCours : int)

Selon Contôleur,  
reçoit la première opération système

confirmerSuppression(sigleCours : String, idGroupeCours : int)

## Contrat d'opération

## PostCondition

- Le cours n'existe plus dans operationQuestion

## RDCU

## RDCU pour confirmerSuppression

:EnseignantControleur

operationQuestion : OperationQuestion

confirmerSuppression(sigleCours : String, idGroupeCours : int)

Selon Contôleur,  
reçoit la deuxième opération systèmeChangement valeur attribut comme quoi la  
suppression a été confirmée par l'utilisateur

setSuppression(true)

Selon fabrication pure : operationQuestion  
ne représente pas un problème du domaine,  
mais permet de réduire le couplage et d'augmenter la cohésion

supprimerUneQuestionSga(question)

## CU02a - Ajouter question

**Acteur principal:** Enseignant**Préconditions:**

- L'enseignant est authentifié.

**Garanties en cas de succès (postconditions):**

- Une nouvelle question a été créée dans la banque pour le cours.

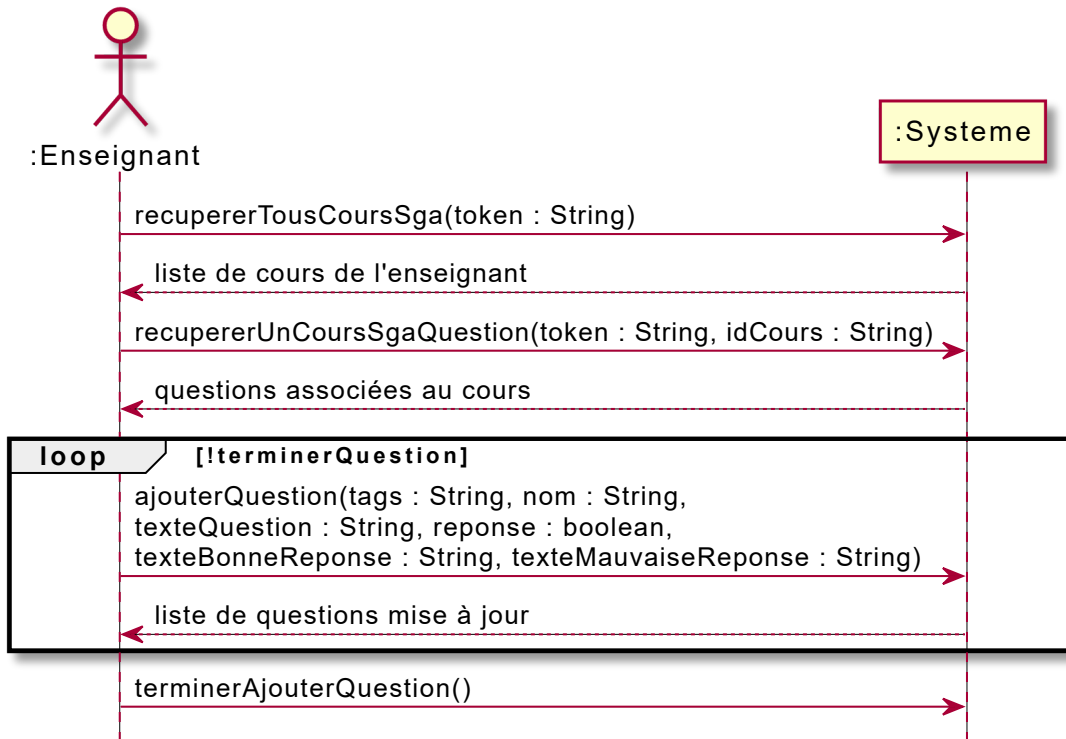
**Scénario principal (succès):**

1. L'enseignant commence la création de questions
2. Le système affiche les cours actifs de l'enseignant
3. L'enseignant sélectionne un cours
4. Le système affiche toutes les questions associées au cours.
5. L'enseignant ajoute une question de type vrai-faux en spécifiant une ou plusieurs catégories non hiérarchiques ("tags"), un nom (court) de la question, l'énoncé (le texte) de la question, la vérité (vrai ou faux) de l'énoncé, un texte de rétroaction pour la bonne réponse et un texte de rétroaction pour la mauvaise réponse.
6. L'enseignant répète l'étape 5 jusqu'à ce qu'il n'ait plus de questions à ajouter au cours.



**Extensions (ou scénarios alternatifs):**

- 5a. L'enseignant ajoute un autre type de question (défini par S4)
- 5b. Le nom de la question n'est pas unique.
1. Le Système signale l'erreur et rejette la saisie.

**DSS CU02a - Ajouter question****DSS ajouter question**

`recupererTousCoursSga(token : String)`

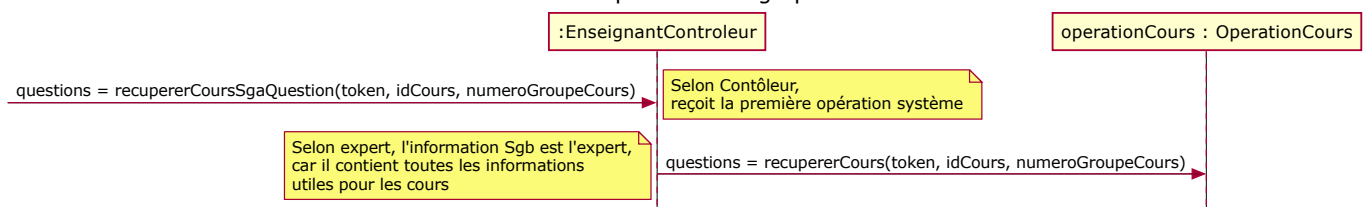
`recupererTousCoursSga`

`recupererUnCoursSgaQuestion(token : String, idCours : String)`

**Contrat d'opération**

*PostCondition*

- aucune

**RDCU****RDCU recupererCoursSgaquestion**

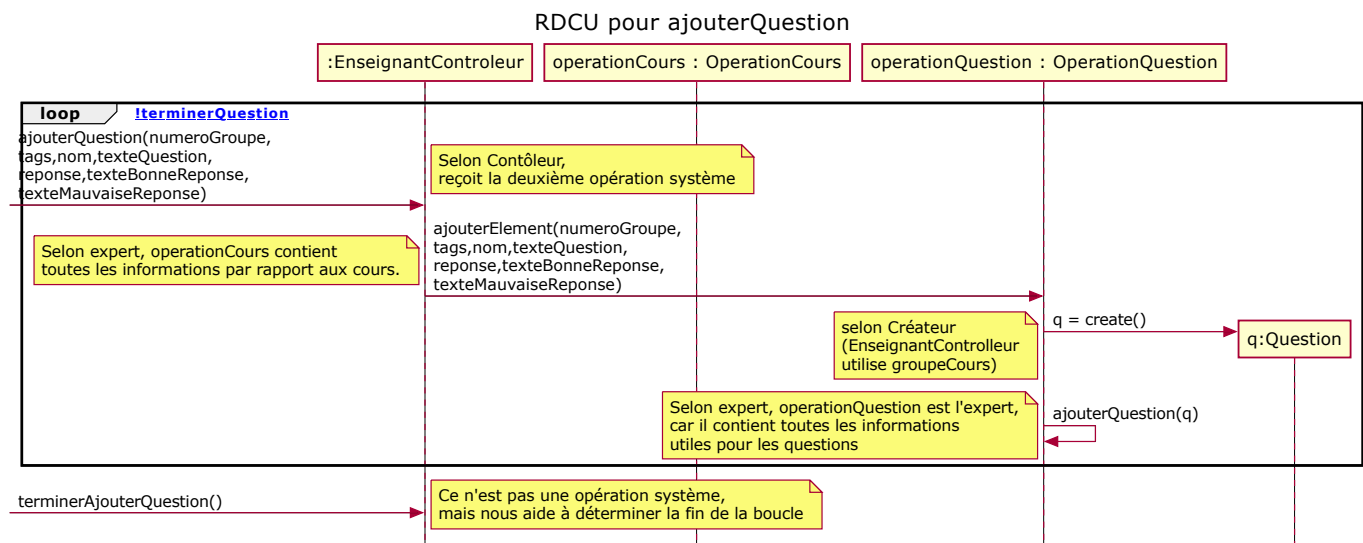
```
ajouterQuestion(numeroGroupe : int, tags : String, nom : String,
texteQuestion : String, reponse : boolean, texteBonneReponse : String,
texteMauvaiseReponse : String)
```

## Contrat d'opération

### PostCondition

- Une nouvelle question a été créée
- La nouvelle question a été associée à operationQuestion.
- La question a été associée a un GroupeCours

## RDCU



## CU02b - Récupérer question

**Acteur principal:** Enseignant

### Préconditions:

- L'enseignant est authentifié.

### Garanties en cas de succès (postconditions):

- Aucun

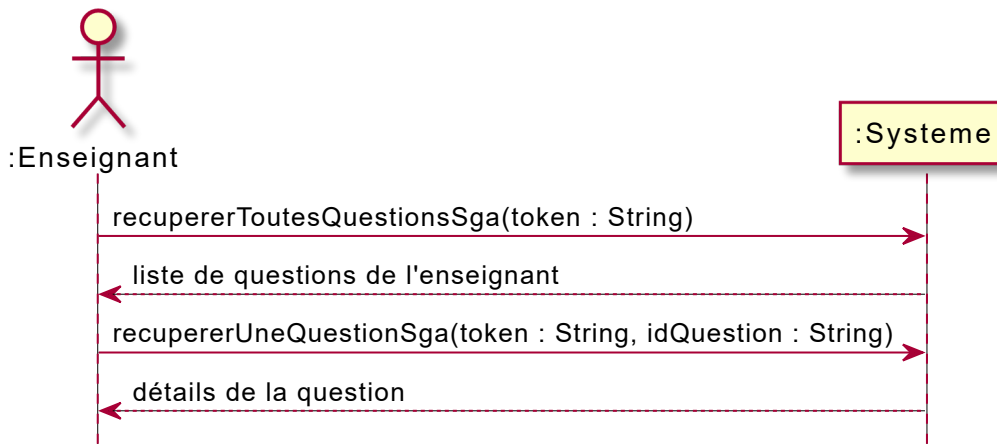
### Scénario principal (succès):

1. L'enseignant commence la récupération de questions
2. Le système affiche toutes les questions de l'enseignant
3. L'enseignant sélectionne une question
4. Le système affiche les détails de la question

On répète les étapes 3 et 4 tant que l'enseignant n'a pas terminé

## DSS CU02b - Récupérer question

## DSS récupérer question



`recupererToutesQuestionsSga(token : String)`

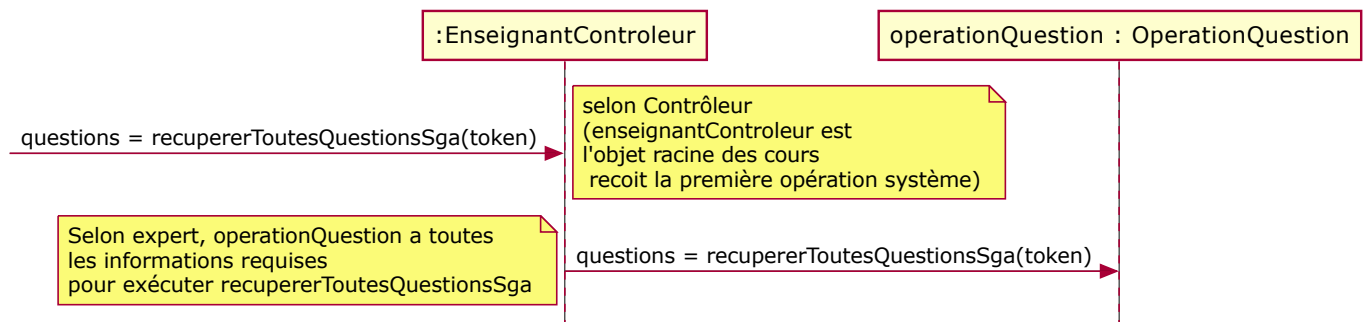
**Contrat d'opération**

*PostCondition*

- aucune

**RDCU**

RDCU pour récupérer toutes les questions créées d'un enseignant



`recupererUneQuestion(token : String, idQuestion : String)`

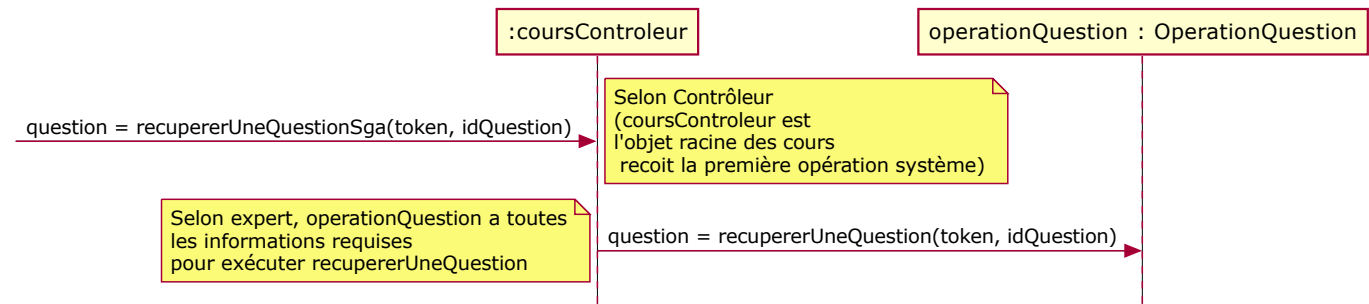
**Contrat d'opération**

*PostCondition*

- aucune

**RDCU**

RDCU pour récupérer une question créée d'un enseignant



CU02c - Modifier question

Acteur principal: Enseignant

Préconditions:

- L'enseignant est authentifié.

Garanties en cas de succès (postconditions):

- La question a été modifiée dans la banque pour le cours.

Scénario principal (succès):

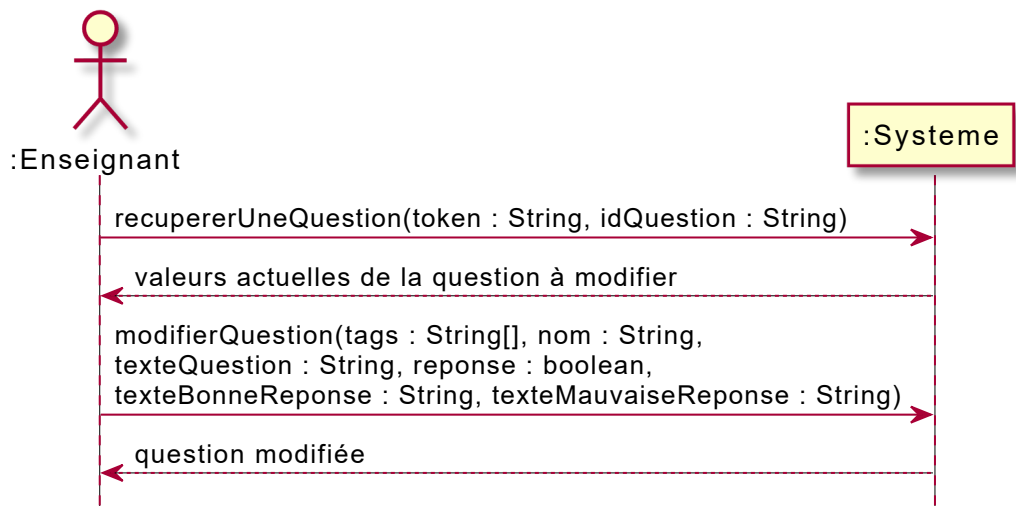
1. L'enseignant commence la modification d'une question
2. Le système affiche les valeurs actuelles de la question à modifier.
3. L'enseignant effectue les modifications de la question et les sauvegarde.
4. Le système affiche la question modifiée.

Extensions (ou scénarios alternatifs):

- 3a. Le nom (modifié) de la question n'est pas unique.
- 1. Le Système signale l'erreur et rejette la saisie.

DSS CU02c - Modifier question

DSS modifier question



```
recupererUneQuestion(token : String, idQuestion : String)
```

```
recupererUneQuestion
```

```
modifierQuestion(tags : String[], nom : String, texteQuestion : String,
reponse : boolean, texteBonneReponse : String, texteMauvaiseReponse :
String)
```

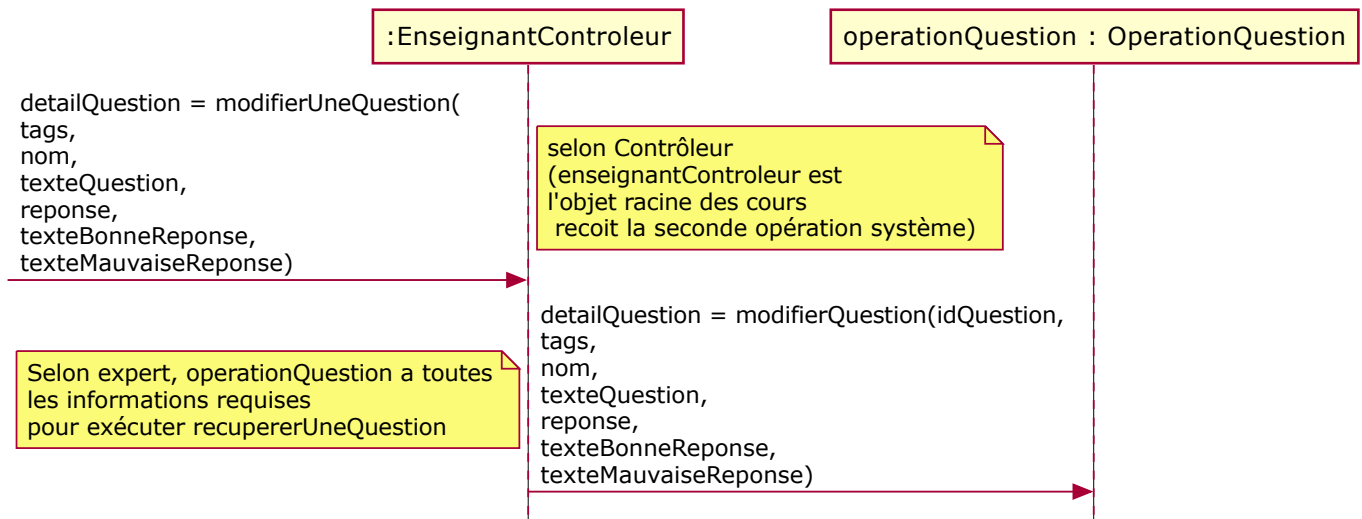
### Contrat d'opération

*PostCondition*

- La question a été modifiée dans operationQuestion

### RDCU

#### RDCU modifierUneQuestion



## CU02d - Supprimer question

**Acteur principal:** Enseignant

**Préconditions:**

- L'enseignant est authentifié.

**Garanties en cas de succès (postconditions):**

- Une question a été supprimée de la banque pour le cours.

**Scénario principal (succès):**

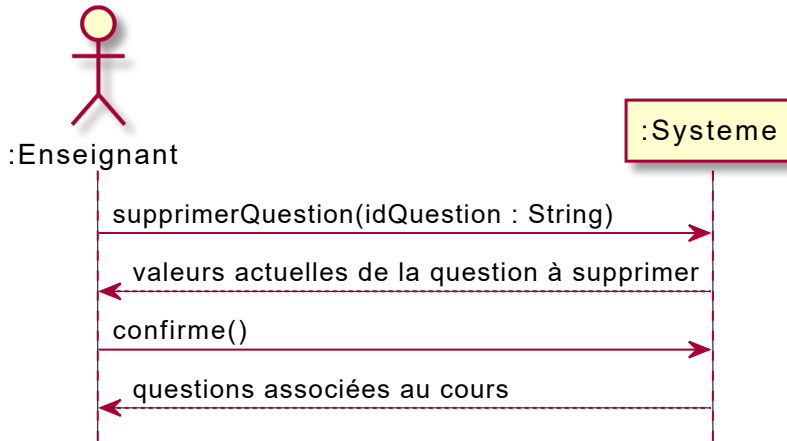
1. L'enseignant commence la suppression d'une question
2. Le système affiche les valeurs actuelles de la question à supprimer.
3. L'enseignant confirme la suppression de la question

**Extensions (ou scénarios alternatifs):**

2a. Le système affiche la liste des questionnaires utilisant cette question et désactive la possibilité de suppression tant que la question est utilisée dans un questionnaire.

## DSS CU02d - Modifier question

### DSS supprimer question



`recupererUneQuestion(token : String, idQuestion : String)`

`recupererUneQuestion`

`supprimerUneQuestion(idQuestion : String)`

### Contrat d'opération

*PostCondition*

- Une question a été supprimée dans operationQuestion

### RDCU

### RDCU pour supprimer une question créée d'un enseignant

