

# Rapport itération 2 - équipe 3

---

LOG210-2021-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

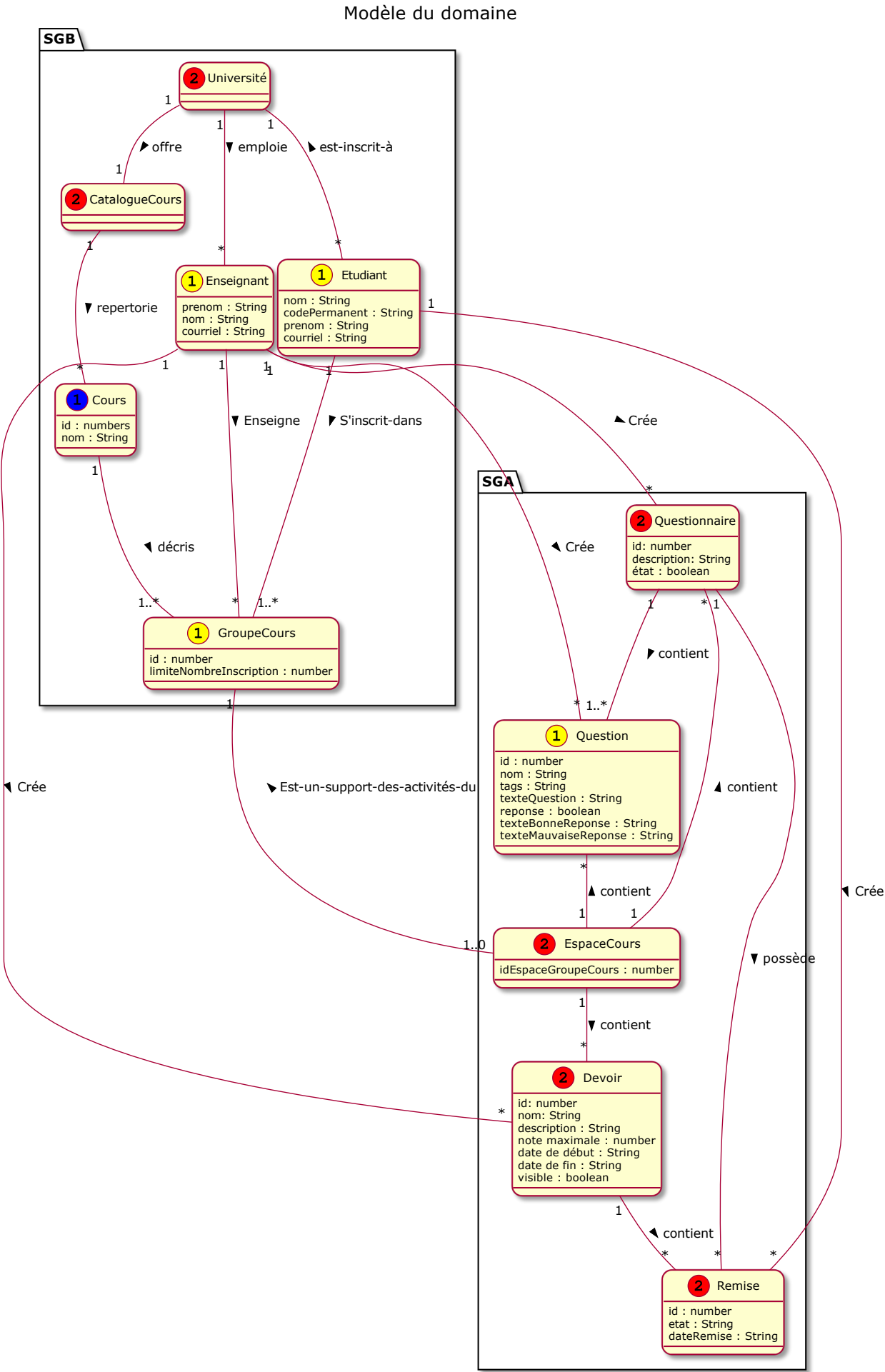
- Tous vos diagrammes doivent être faits avec <https://plantuml.com/fr/>
- Les diagrammes doivent être visibles dans ce rapport
- Supprimer les textes explicatifs du gabarit (sauf ces consignes-ci)
- Vous devez exporter ce fichier en format PDF et l'ajouter dans votre dépôt

# Introduction

---

L'objectif général de ce projet est de concevoir un système de gestion des apprentissages. Dans cette itération, les cas d'utilisations CU04 (a,b,c,d) ainsi que CU05 (a,b,c,d) ont été implémentés, testés et documentés. Ces cas correspondent à la gestion des devoirs ainsi que des questionnaires. Cette itération inclut notamment l'ajout de ces fonctionnalités, l'ajout de tests, l'incrémentation du MDD, l'ajout de DSS pour chaque cas d'utilisation ainsi que l'ajout de RDCU pour chacune des opérations système.

# Modèle du domaine



Type
1 Classe non modifiée première itération
1 Classe modifiée première itération
2 Nouvelle classe deuxième itération

## CU04a - Ajouter devoir

**Acteur principal:** Enseignant

**Préconditions:**

- L'enseignant est authentifié.

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

- Un nouveau devoir est créé et associé à un cours

**Scénario principal (succès):**

1. L'enseignant commence la création d'un devoir
2. Le système affiche les cours de l'enseignant ainsi que le nombre de devoirs associés à chaque cours
3. L'enseignant sélectionne un cours
4. Le système affiche tous les devoirs associés au cours.
5. L'enseignant crée un nouveau devoir avec une description et un nom, une note maximale, une date de début, une date de fin et un état visible ou non.
6. Le système confirme l'ajout du devoir et affiche tous les devoirs associés au cours.

On répète les étapes 5-6 tant qu'il y a un devoir à ajouter

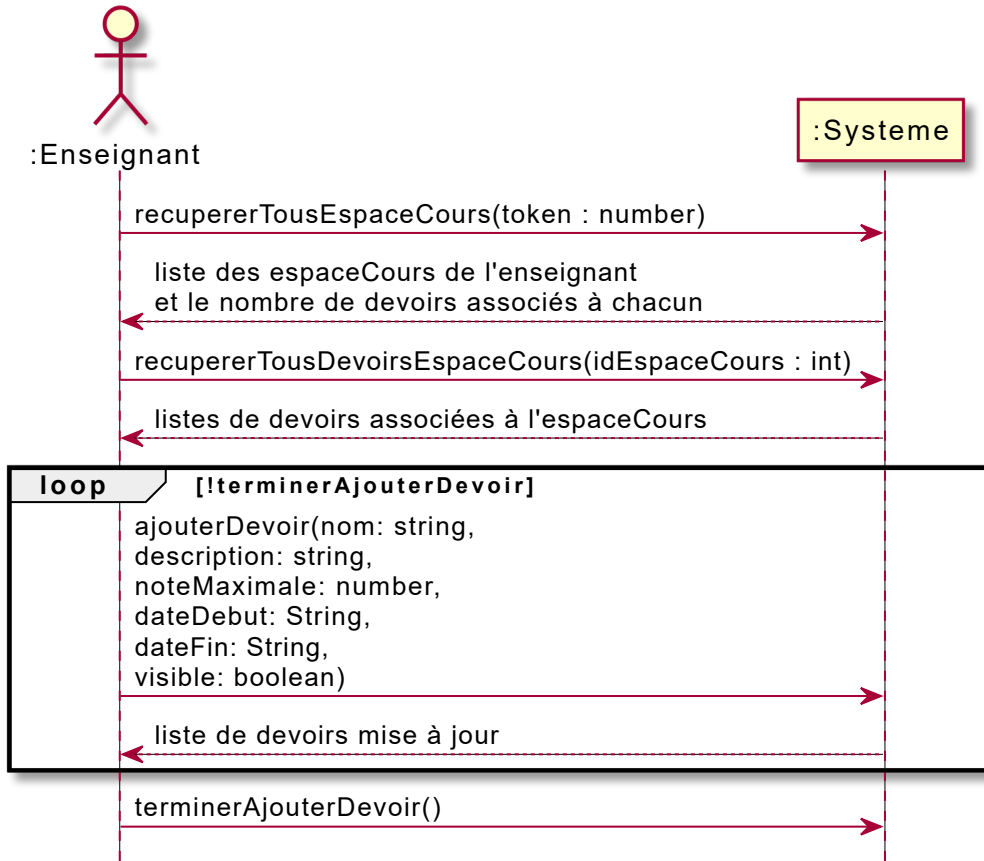
**Extensions (ou scénarios alternatifs):**

5a. La date de début est après la date de fin.

1. Le Système signale l'erreur et rejette la saisie.

## DSS CU04a - Ajouter devoir

## DSS Ajouter Devoir



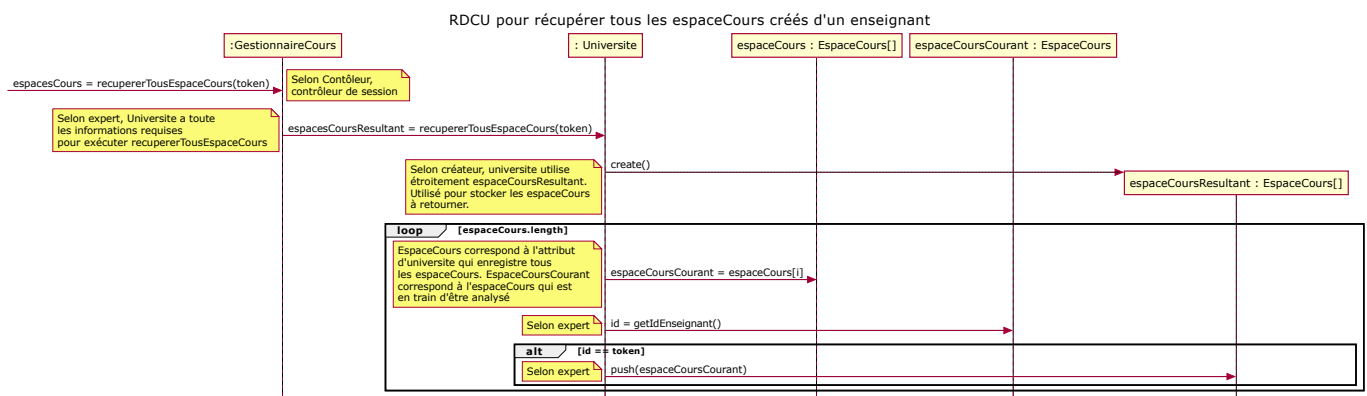
recupererTousEspaceCours(token : String)

### Contrat d'opération

PostCondition

- Aucune

### RDCU



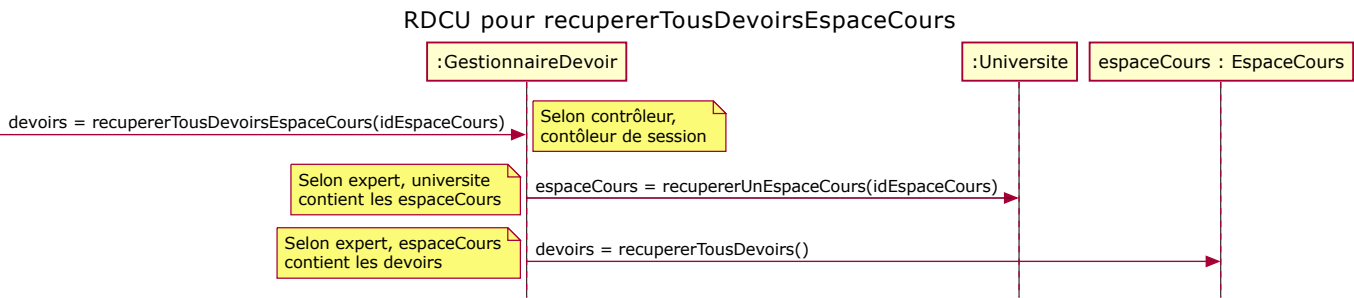
recupererTousDevoirsEspaceCours(idEspaceCours : number)

### Contrat d'opération

PostCondition

- Aucune

RDCU



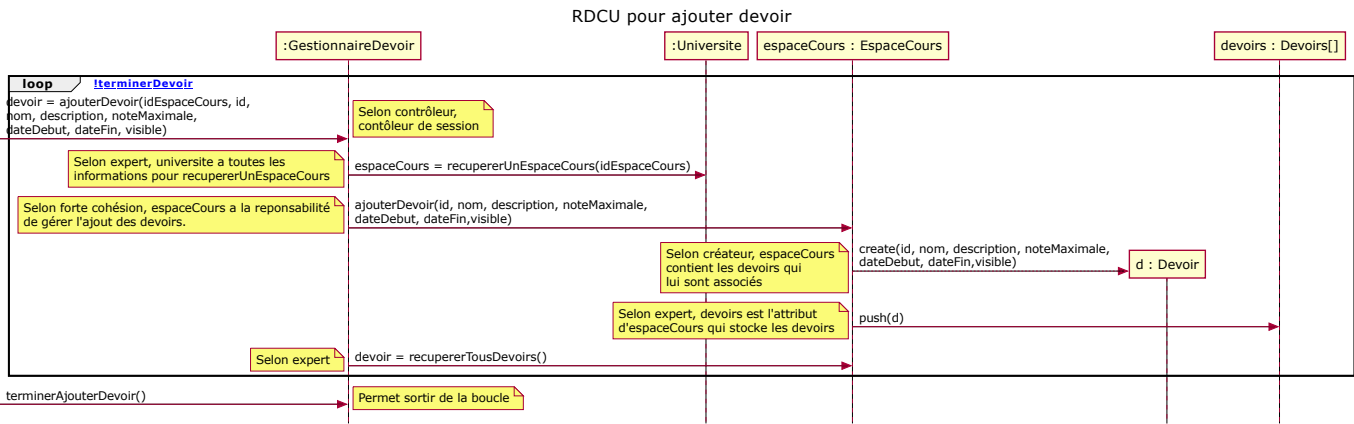
ajouterDevoir(nom: string, description: string, noteMaximale: number, dateDebut: String, dateFin: String, visible: boolean)

Contrat d'opération

PostCondition

- Une nouvelle instance d de Devoir a été créée
- Les attributs de d ont été initialisés
- d a été associé à un EspaceCours sur une base de correspondance avec idEspaceCours

RDCU



CU04b - Récupérer devoir

Acteur principal: Enseignant

Préconditions:

- L'enseignant est authentifié.

Garanties en cas de succès (postconditions):

- Aucune

Scénario principal (succès):

- L'enseignant commence la récupération d'un devoir

2. Le système affiche les cours de l'enseignant ainsi que le nombre de devoirs associés à chaque cours
3. L'enseignant sélectionne un cours
4. Le système affiche tous les devoirs associés au cours.
5. L'enseignant sélectionne un devoir
6. Le système affiche le détail du devoir
7. Le système affiche la liste des étudiants ayant fait le devoir ainsi que la note leur étant associée.

On répète les étapes 5 à 7 tant que l'enseignant n'a pas terminé.

On répète les étapes 3 à 7 tant que l'enseignant n'a pas terminé.

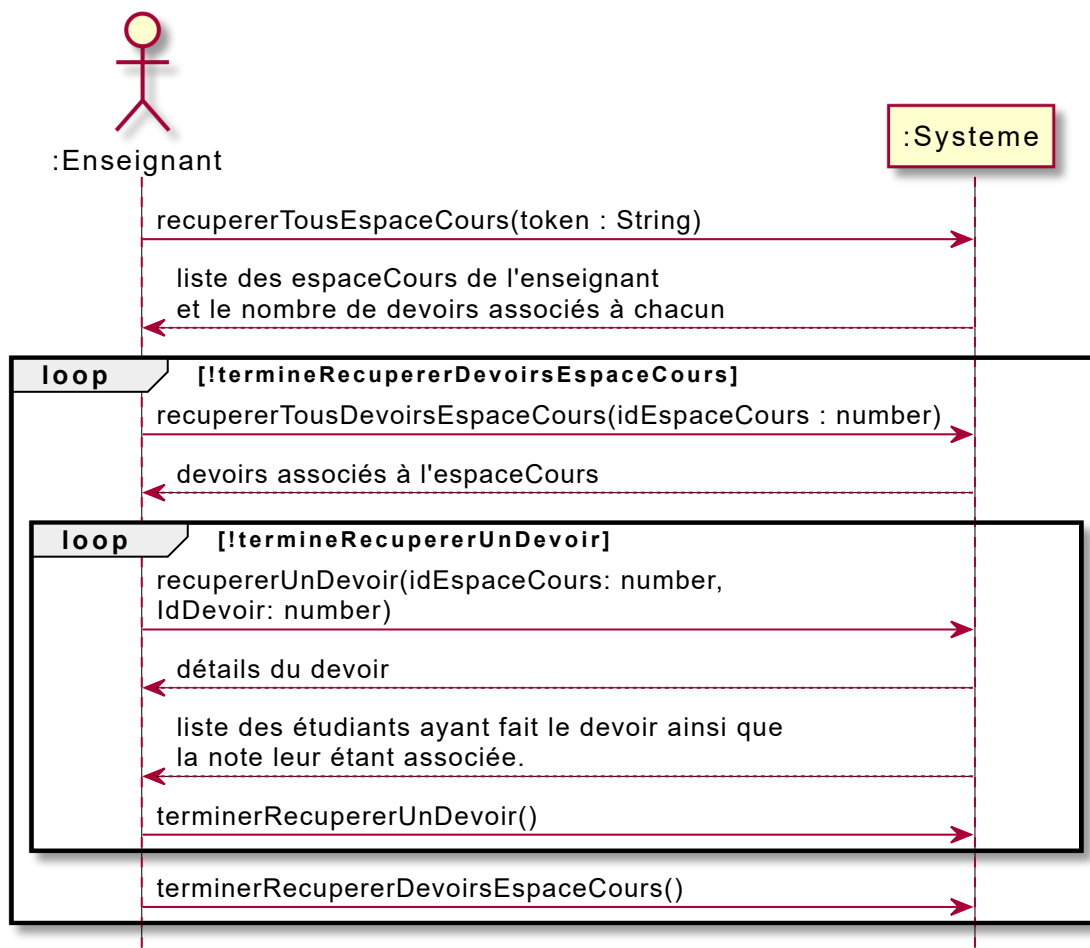
### Extensions (ou scénarios alternatifs):

7a. Le système affiche les étudiants par ordre alphabétique.

7b. Le système affiche les étudiants par ordre croissant de la note.

## DSS CU04b - Récupérer devoir

### DSS recupererDevoir



`recupererTousEspaceCours(token : String)`

`recupererTousEspaceCours`

`recupererTousDevoirsEspaceCours(idEspaceCours : number)`

recupererTousDevoirsEspaceCours

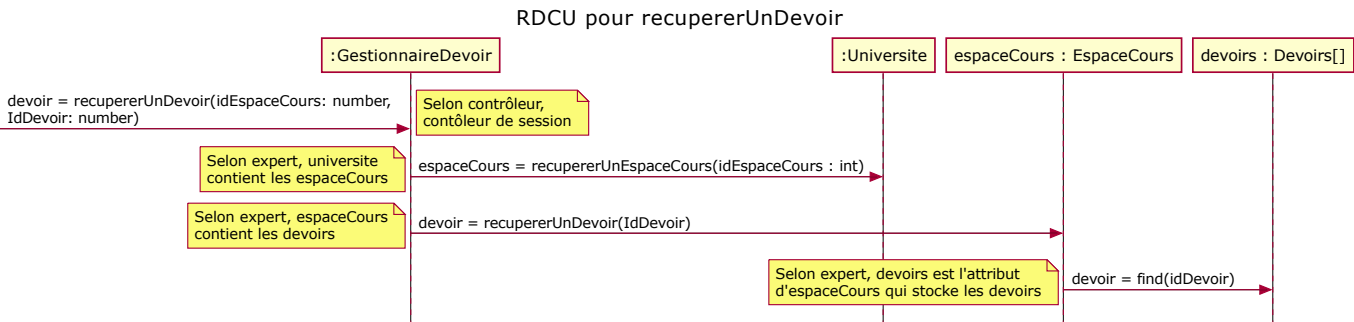
recupererUnDevoir(idEspaceCours: number, IdDevoir: number)

Contrat d'opération

PostCondition

- Aucune.

RDCU



CU04c - Modifier devoir

Acteur principal: Enseignant

Préconditions:

- L'enseignant est authentifié.

Garanties en cas de succès (postconditions):

- Un devoir est modifié

Scénario principal (succès):

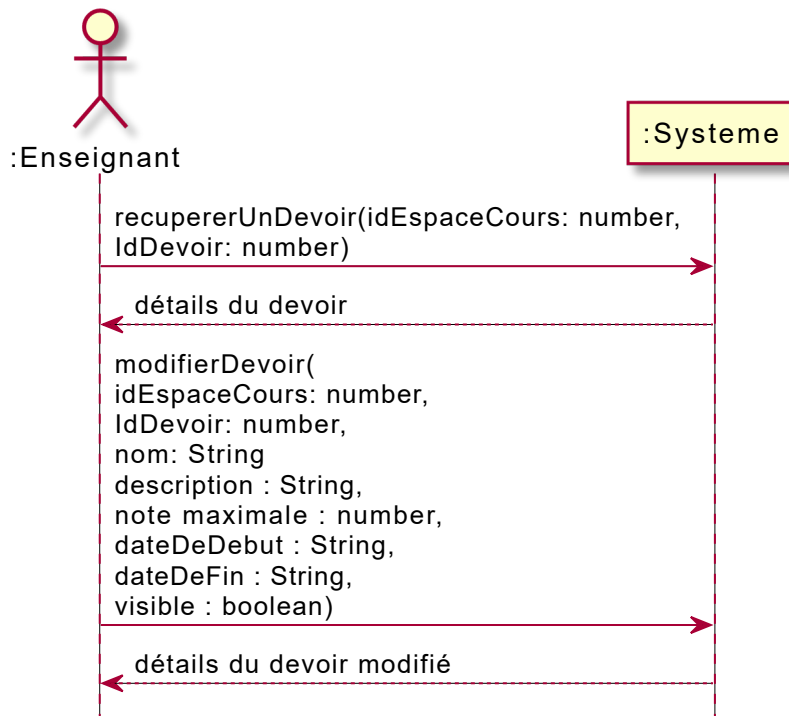
1. L'enseignant commence la modification d'un devoir
2. Le système affiche les valeurs actuelles du devoir à modifier.
3. L'enseignant effectue les modifications du devoir et les sauvegarde.
4. Le système affiche tout le devoir modifié

Extensions (ou scénarios alternatifs):

- 1a. Un devoir ne peut pas être modifié si des étudiants ont déjà commencé à réaliser celui-ci.

DSS CU04c - Modifier devoir

## DSS modifierDevoir



`recupererUnDevoir(idEspaceCours: number, IdDevoir: number)`

RécupérerUnDevoir

`modifierDevoir(idEspaceCours: number, IdDevoir: number)`

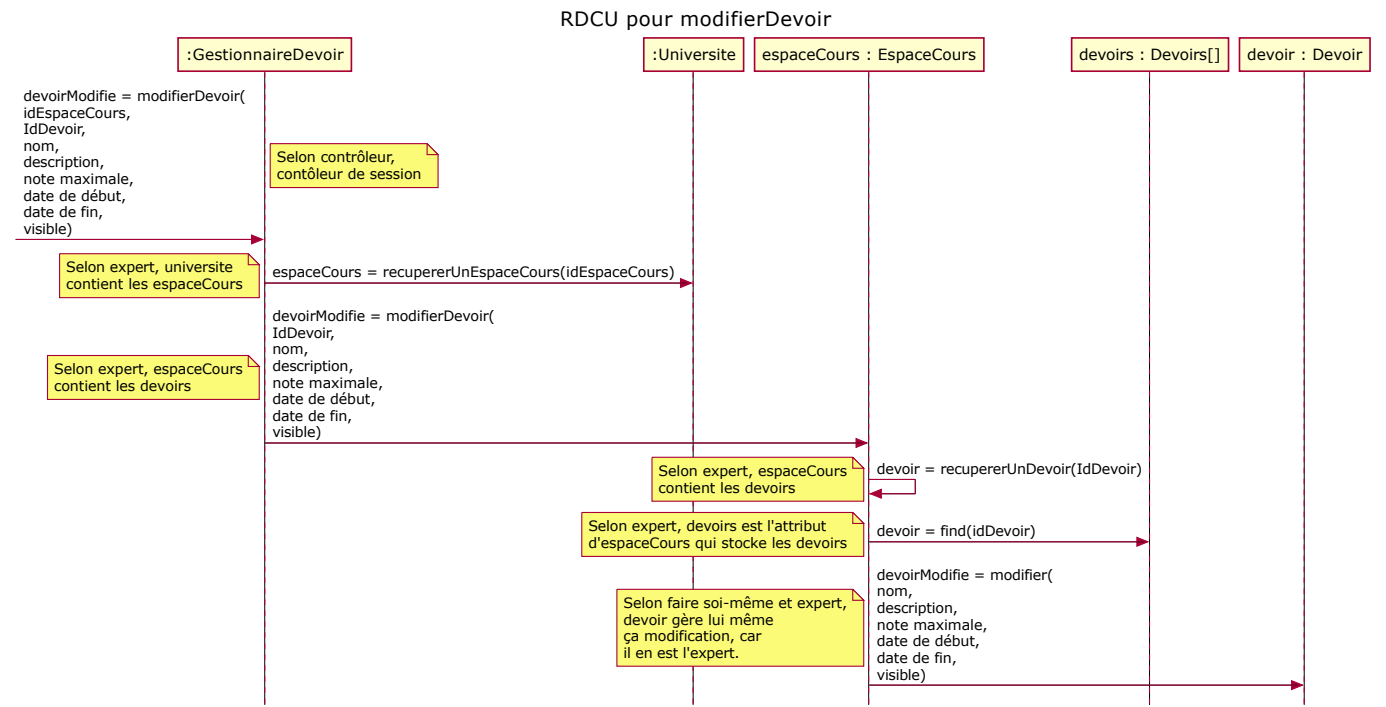
### Contrat d'opération

*PostCondition*

- Les attributs du devoir d ont été modifié

### RDCU





## CU04d - Supprimer devoir

**Acteur principal:** Enseignant

**Préconditions:**

- L'enseignant est authentifié.

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

- Un devoir a été supprimé pour le cours.

**Scénario principal (succès):**

1. L'enseignant commence la suppression d'un devoir
2. Le système affiche les valeurs du devoir à supprimer.
3. L'enseignant confirme la suppression du devoir

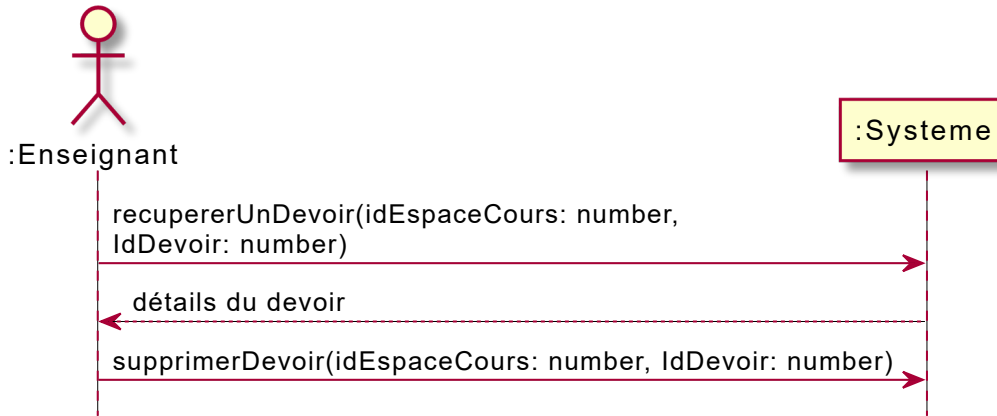
**Extensions (ou scénarios alternatifs):**

- 2a. Le système désactive la possibilité de suppression tant que le devoir a été utilisé par des étudiants.

---

## DSS CU04d - Supprimer devoir

## DSS supprimerDevoir



recupererUnDevoir(idEspaceCours: number, IdDevoir: number)

RécupérerUnDevoir

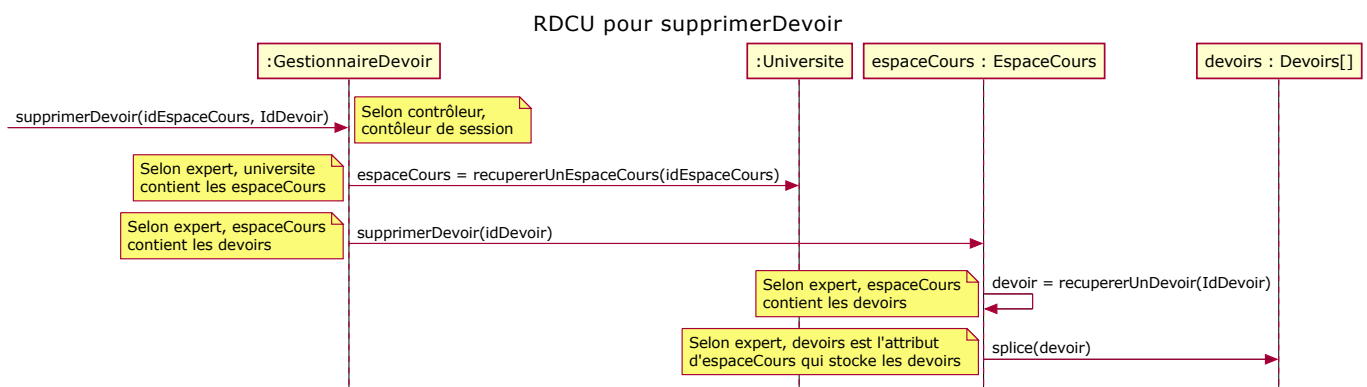
supprimerDevoir(idEspaceCours: number, IdDevoir: number)

### Contrat d'opération

*PostCondition*

- L'instance d de Devoir a été supprimée

### RDCU



## CU05a - Ajouter questionnaire

**Acteur principal:** Enseignant

**Préconditions:**

- L'enseignant est authentifié.

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

- Un nouveau questionnaire a été créé et associé à un cours
- Des questions ont été associées au questionnaire

**Scénario principal (succès):**

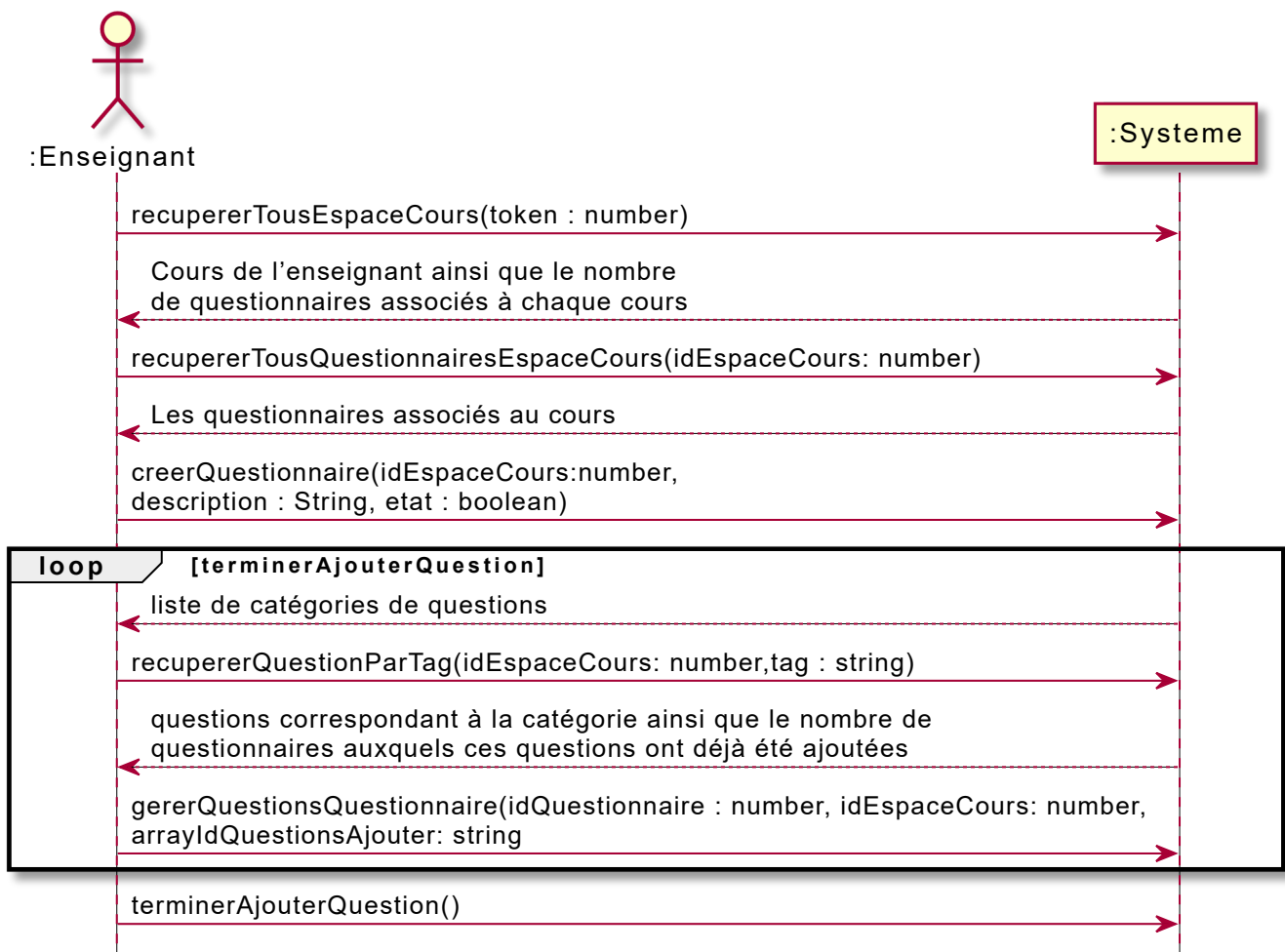
1. L'enseignant commence la création d'un questionnaire
2. Le système affiche les cours de l'enseignant ainsi que le nombre de questionnaires associés à chaque cours
3. L'enseignant sélectionne un cours
4. Le système affiche tous les questionnaires associés au cours.
5. L'enseignant crée un nouveau questionnaire avec une description et un état pour indiquer si le questionnaire est actif
6. Le système confirme la création du questionnaire
7. Le système affiche la liste de catégories de questions
8. L'enseignant sélectionne la catégorie de question
9. Le système affiche les questions correspondant à la catégorie ainsi que le nombre de questionnaires auxquels ces questions ont déjà été ajoutées.
10. L'enseignant sélectionne une ou plusieurs questions et les ajoute au questionnaire.

On répète les étapes 7-10 tant que l'enseignant n'a pas terminé l'ajout de questions.

### Extensions (ou scénarios alternatifs):

## DSS CU05a - Ajouter questionnaire

### DSS ajouterQuestionnaire



`recupererTousEspaceCours(token : String)`

recupererTousEspaceCours

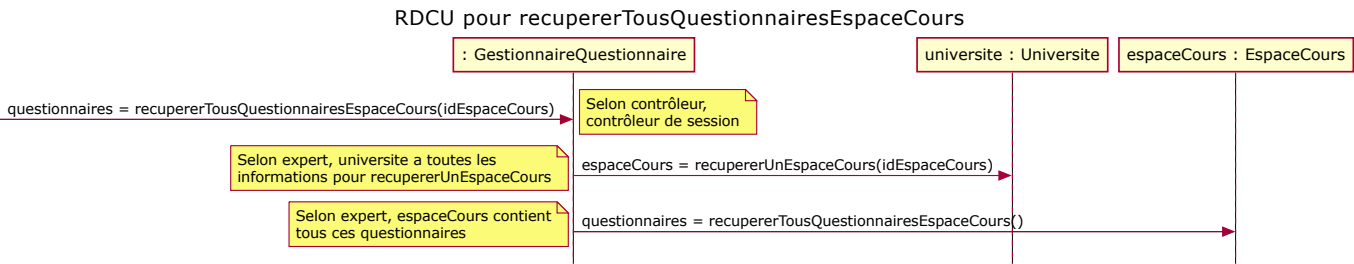
recupererTousQuestionnairesEspaceCours(idEspaceCours: number)

Contrat d'opération

PostCondition

- Aucune

RDCU



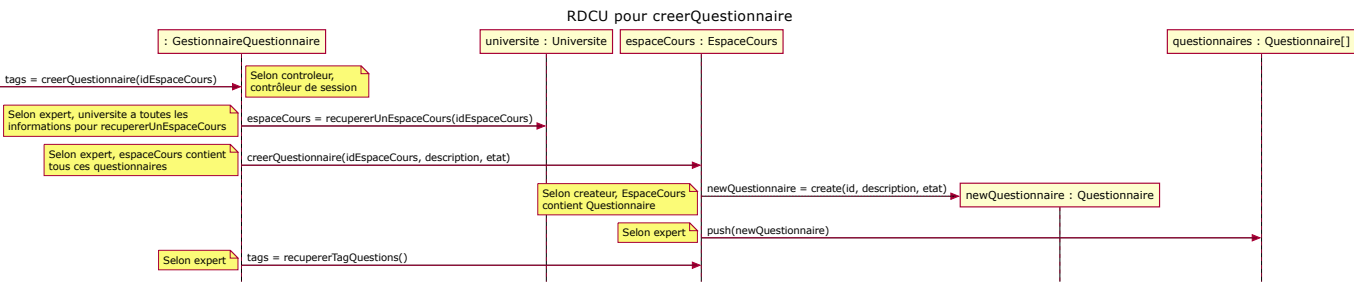
creerQuestionnaire(idEspaceCours:number, description : String, etat : boolean)

Contrat d'opération

PostCondition

- Une nouvelle instance newQuestionnaire de Questionnaire a été créée
- Les attributs de newQuestionnaire ont été initialisés
- newQuestionnaire a été associée à son espaceCours sur la base de correspondance de idEspaceCours

RDCU



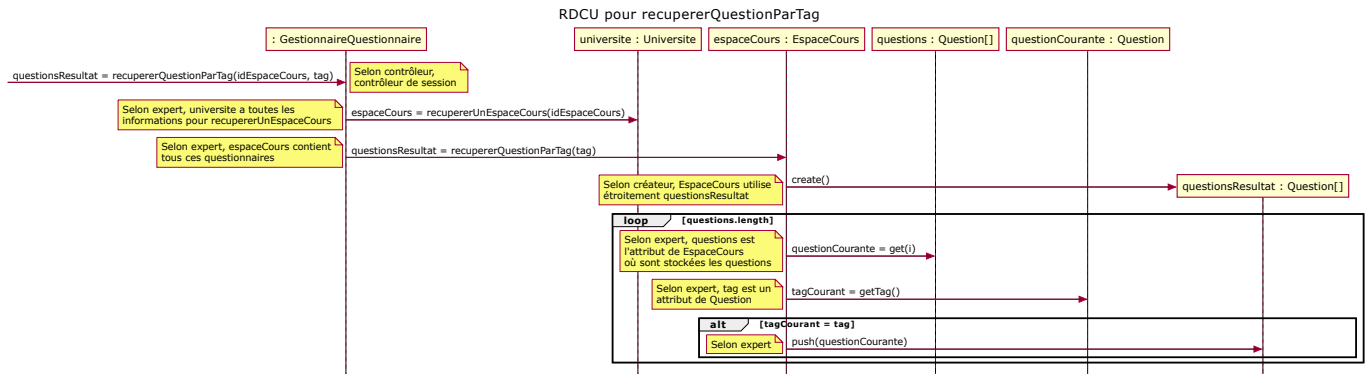
recupererQuestionParTag(idEspaceCours: number,tag : string)

Contrat d'opération

PostCondition

- Aucune

RDCU



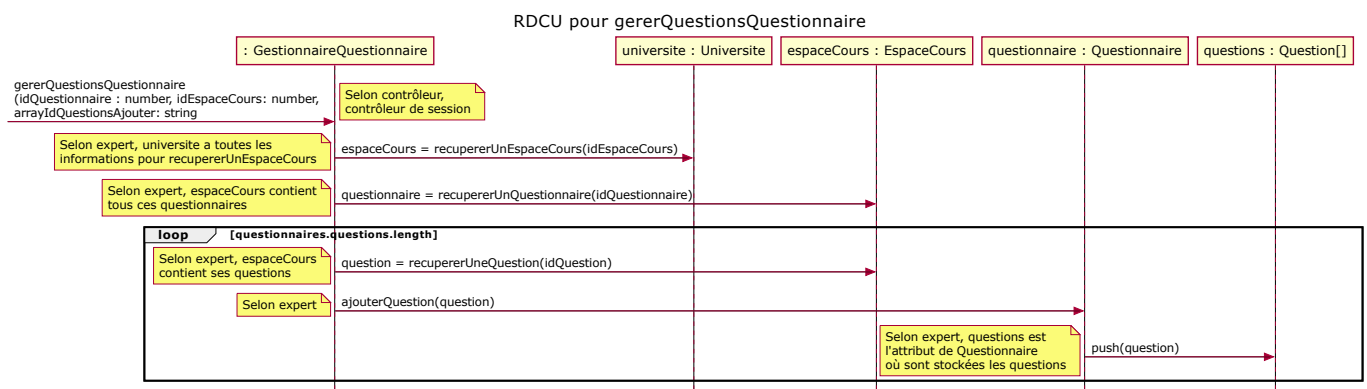
gererQuestionsQuestionnaire(idQuestionnaire : number, idEspaceCours: number, arrayIdQuestionsAjouter: string)

### Contrat d'opération

#### PostCondition

- Les questions basé sur une correspondance avec idQuestionsAjouter ont été associées au questionnaire sur une base de correspondance avec idQuestionnaire

### RDCU



## CU05b - Afficher questionnaire

**Acteur principal:** Enseignant

### Préconditions:

- L'enseignant est authentifié.
- Il existe un cours, un questionnaire, etc.

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

- Aucune

### Scénario principal (succès):

- L'enseignant commence la gestion des questionnaires
- Le système affiche les cours de l'enseignant ainsi que le nombre de questionnaires associés à chaque cours

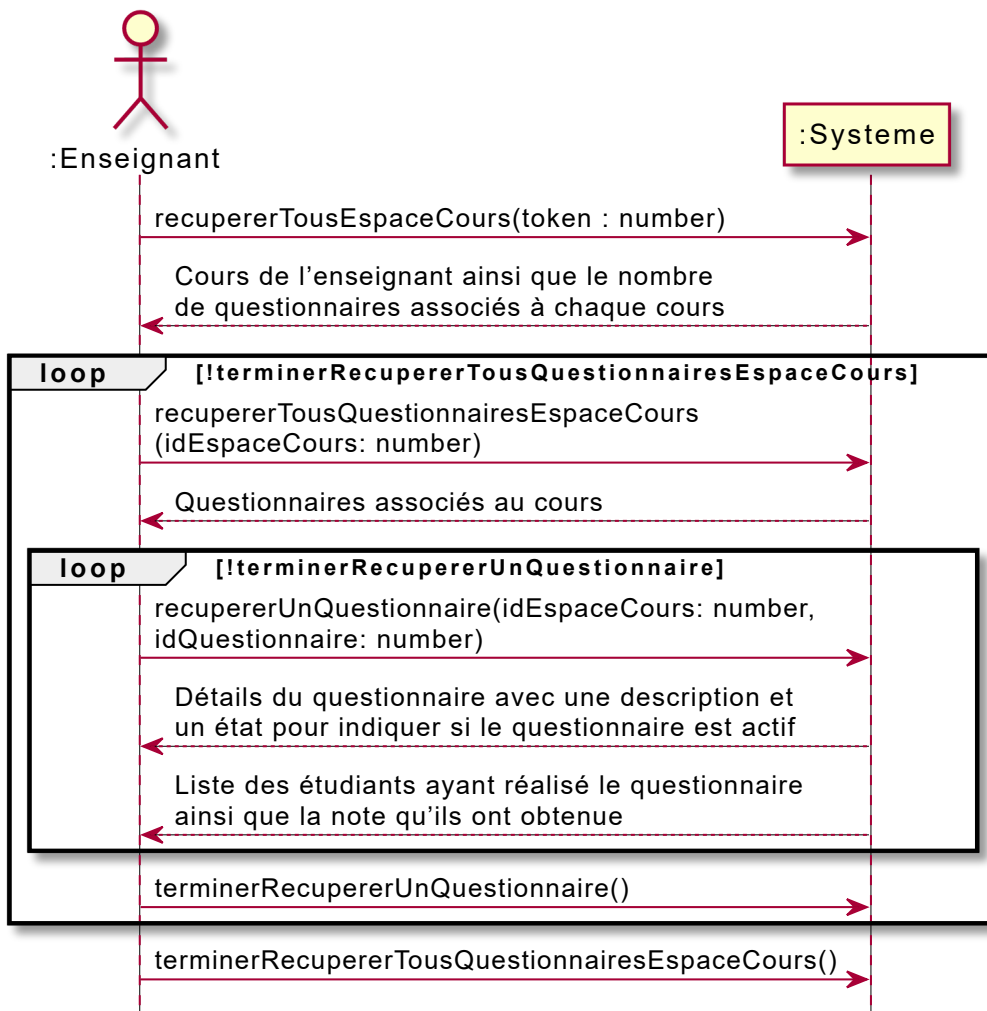
3. L'enseignant sélectionne un cours
4. Le système affiche tous les questionnaires associés au cours.
5. L'enseignant sélectionne un questionnaire
6. Le système affiche les détails du questionnaire avec une description et un état pour indiquer si le questionnaire est actif
7. Le système affiche la liste des étudiants ayant réalisé le questionnaire ainsi que la note qu'ils ont obtenue.

On répète les étapes 5 à 7 tant que l'enseignant n'a pas terminé.

On répète les étapes 3 à 7 tant que l'enseignant n'a pas terminé.

## DSS CU05b - Afficher questionnaire

### DSS afficherQuestionnaire



**recupererTousEspaceCours(token : String)**

recupererTousEspaceCours

**recupererTousQuestionnairesEspaceCours(idEspaceCours: number)**

recupererTousQuestionnairesEspaceCours

recupererUnQuestionnaire(idEspaceCours: number, idQuestionnaire: number)

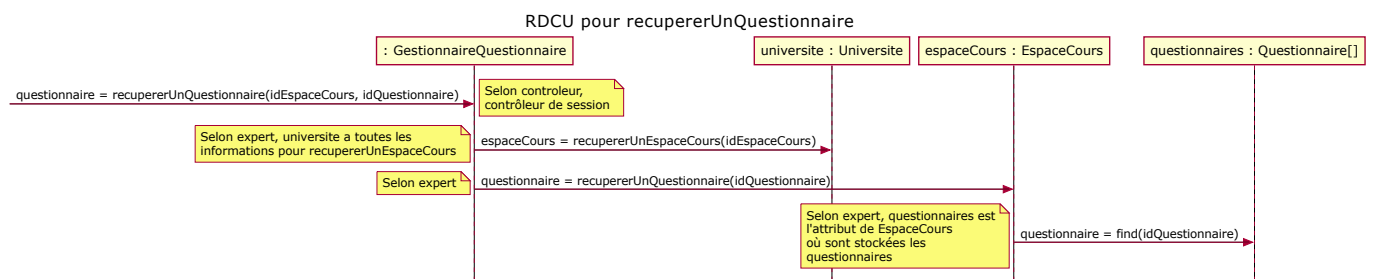
## Contrat d'opération

### PostCondition

- Aucune

## RDCU

Ici pour se qui est de l'affichage des détails du questionnaire, il suffit de récupérer les attributs du questionnaires. Pour la liste d'étudiants il faut récupérer la remise et ensuite récupérer les détails des étudiants. Ces opérations ne sont pas démontrées dans le RDCU car cela ce passe dans le "front-end".



## CU05c - Modifier questionnaire

**Acteur principal:** Enseignant

### Préconditions:

- L'enseignant est authentifié.
- Il existe un cours, un questionnaire, etc.

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

- Un questionnaire a été modifié
- Les questions associées au questionnaire ont été modifiées

### Scénario principal (succès):

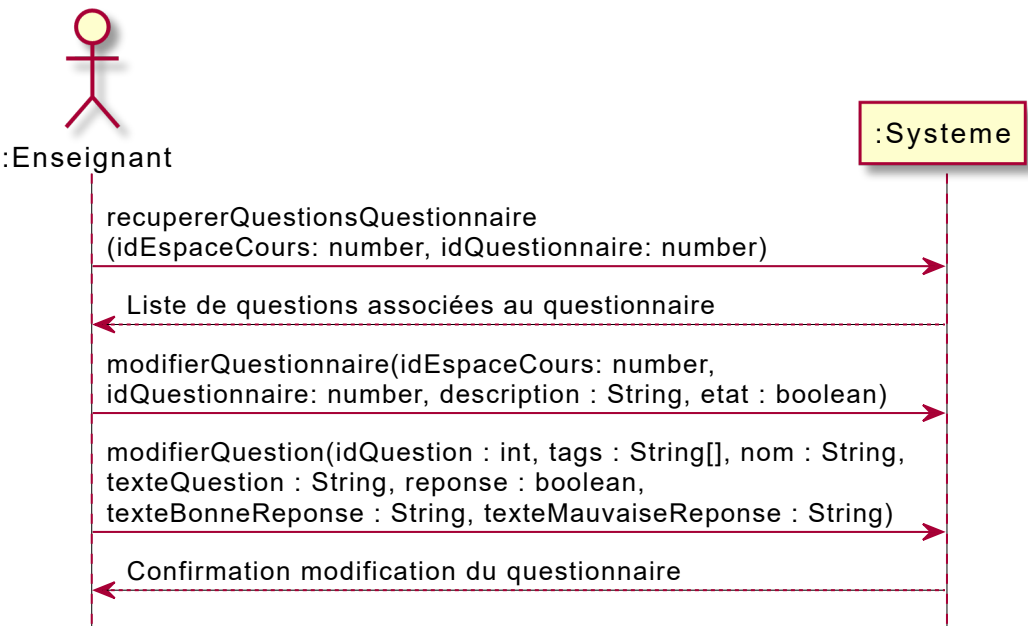
1. L'enseignant commence la modification d'un questionnaire
2. Le système affiche la liste de questions associées au questionnaire
3. L'enseignant modifie la description et/ou l'état du questionnaire
4. L'enseignant modifie les questions associées au questionnaire
5. Le système confirme la modification du questionnaire

### Extensions (ou scénarios alternatifs):

- 3a. 4a. L'enseignant ajoute une question au questionnaire.
- 4b. L'enseignant supprime une question du questionnaire.

## DSS CU05c - Modifier questionnaire

DSS modifierQuestionnaire



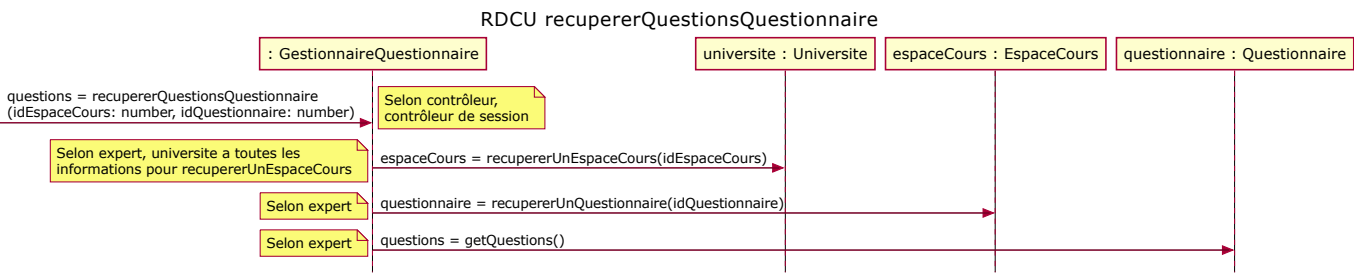
`recupererQuestionsQuestionnaire(idEspaceCours: number, idQuestionnaire: number)`

Contrat d'opération

PostCondition

- Aucune

RDCU



`modifierQuestionnaire(idEspaceCours: number, idQuestionnaire: number, description : String, etat : boolean)`

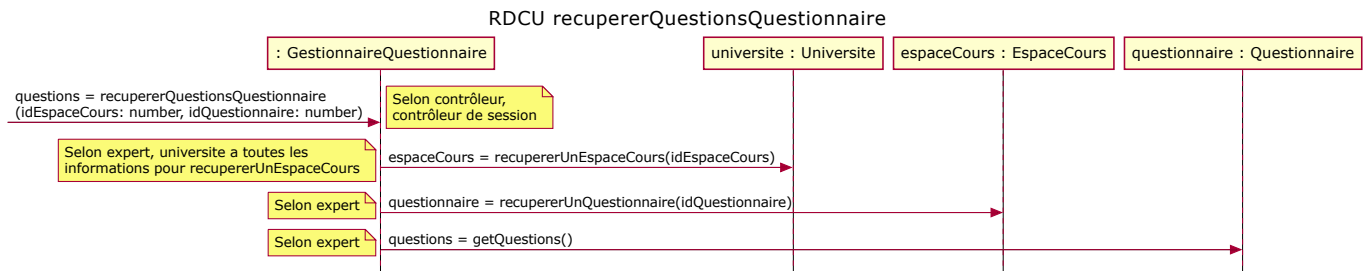
Contrat d'opération

PostCondition

- Les attributs du Questionnaire "questionnaire" ont été modifiés



## RDCU



```

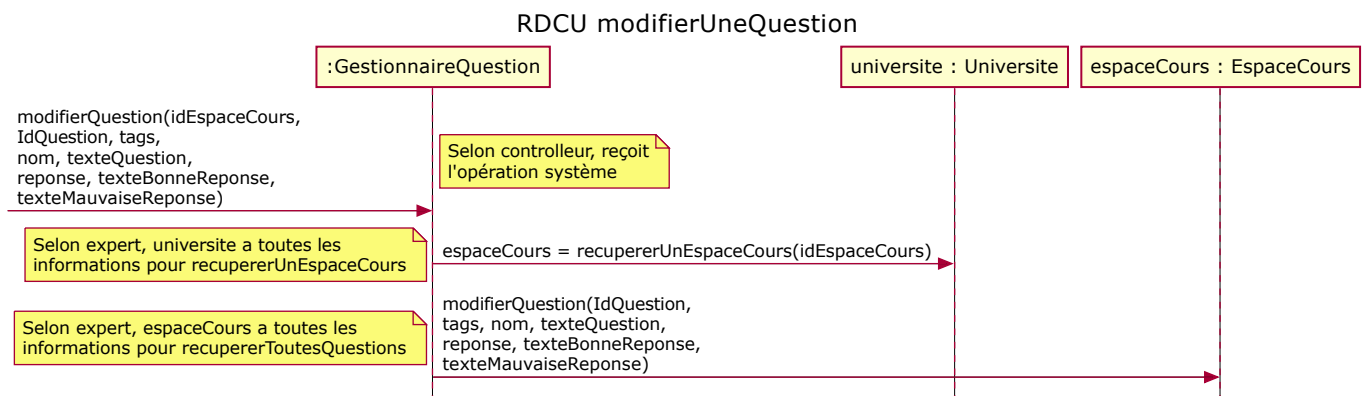
modifierQuestion(tags : String[], nom : String, texteQuestion : String,
reponse : boolean, texteBonneReponse : String, texteMauvaiseReponse :
String)
  
```

## Contrat d'opération

### PostCondition

- Les attributs de la Question "question" ont été modifiés

## RDCU



## CU05d - Supprimer questionnaire

**Acteur principal:** Enseignant

### Préconditions:

- L'enseignant est authentifié.
- Il existe un cours, un questionnaire, etc.

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

- Un questionnaire associé à un cours est supprimé

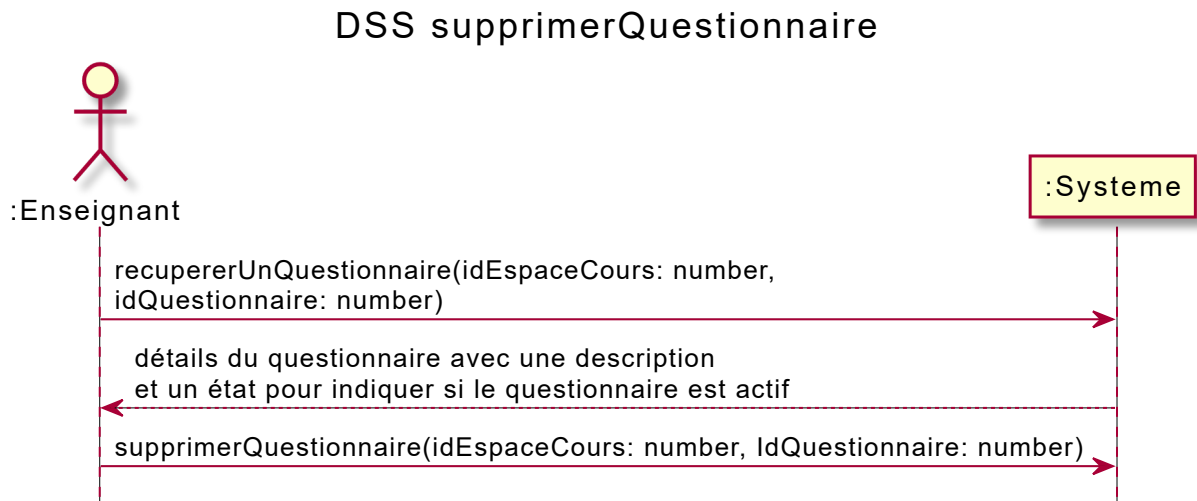
### Scénario principal (succès):

1. L'enseignant commence la suppression d'un questionnaire
2. Le système affiche les détails du questionnaire avec une description et un état pour indiquer si le questionnaire est actif
3. L'enseignant supprime le questionnaire

Extensions (ou scénarios alternatifs):

1a. Le système désactive la possibilité de suppression du questionnaire aussitôt qu'un étudiant à réalisé celui-ci dans le cadre du cours.

DSS CU05d - Supprimer questionnaire



recupererUnQuestionnaire(idEspaceCours: number, idQuestionnaire: number)

recupererUnQuestionnaire

supprimerQuestionnaire(idEspaceCours: number, IdQuestionnaire: number)

Contrat d'opération

PostCondition

- L'instance "questionnaire" a été supprimé sur une base de correspondance avec idQuestionnaire

RDCU

