

# Ingénierie Logiciel







eCours : Cours et QCM en ligne

# Répartition du travail

- Equipe Rouge : Kévin Gesnouin, Loïc Lafontaine, Maxime Lavaste
- Equipe Bleu : Wu Shao, Steven Da Silva
- Equipe Verte : Boris Lenoir, Marco Ponti

# Organisation

**CodeFactory** Partagé en mode privé  
6 sur 6 sujets (2 non lus) ★

-  **GitHub**  
Par Marco Ponti - 1 message - 2 vues 🔊
-  **[IL] Réunion projet aujourd'hui (Boris Lenoir)**  
Par Marco Ponti - 1 message - 1 vue 🔊
-  **Manip pour utiliser le Git**  
Par Marco Ponti - 2 messages - 1 vue
-  **Salut la Plèbe <- URGENT !!! (1)**  
Par Marco Ponti - 1 message - 3 vues
-  **MAJ du jour (c'est juste un rappel de la journée à titre d'info)**  
Par Marco Ponti - 1 message - 3 vues 🔊
-  **Répartition en équipe + Redmine (1)**  
Par Marco Ponti - 1 message - 4 vues

**CodeFactory**








Ashmouth / IL Watch 0 Star 0 Fork 0

[Code](#) [Issues 0](#) [Pull requests 0](#) [Projects 0](#) [Wiki](#) [Pulse](#) [Graphs](#)

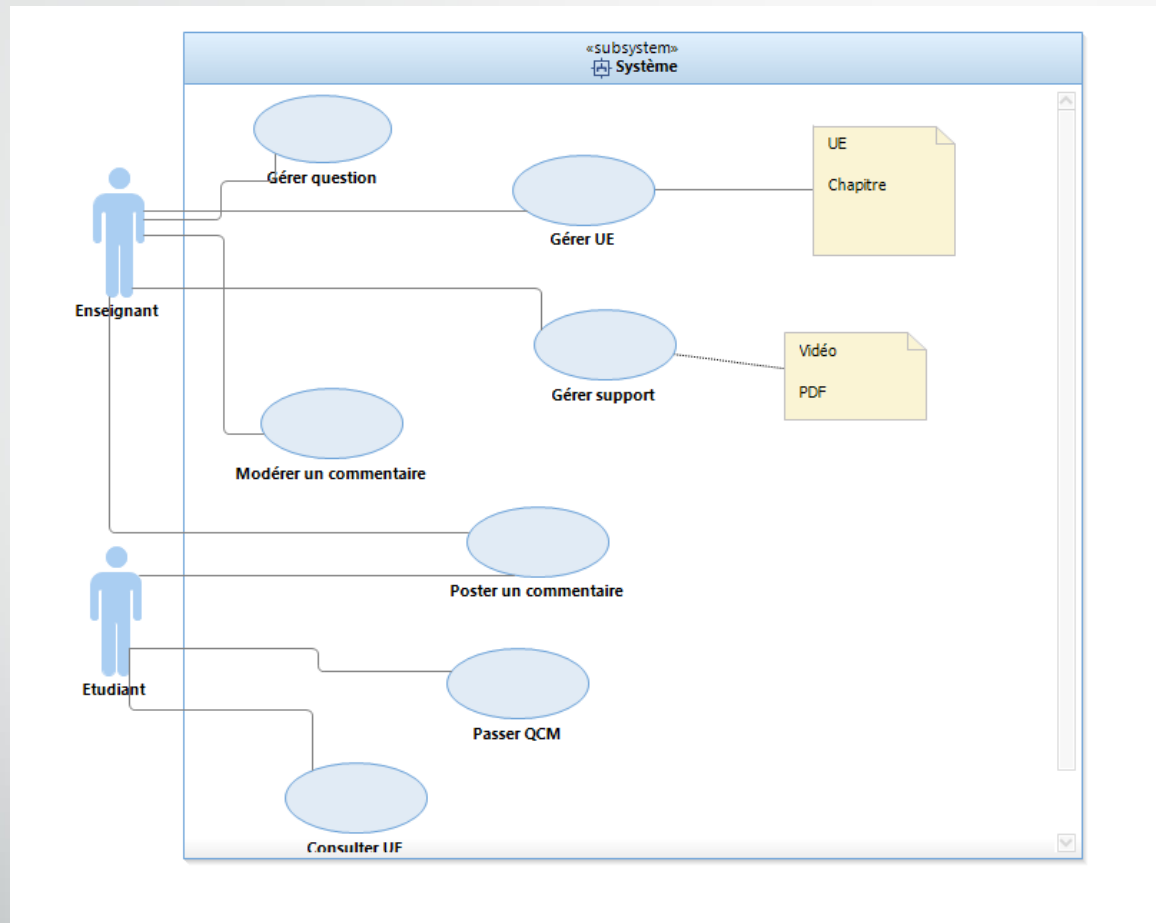
Projet IL (Code Factory)

7 commits 1 branch 0 releases 3 contributors MIT

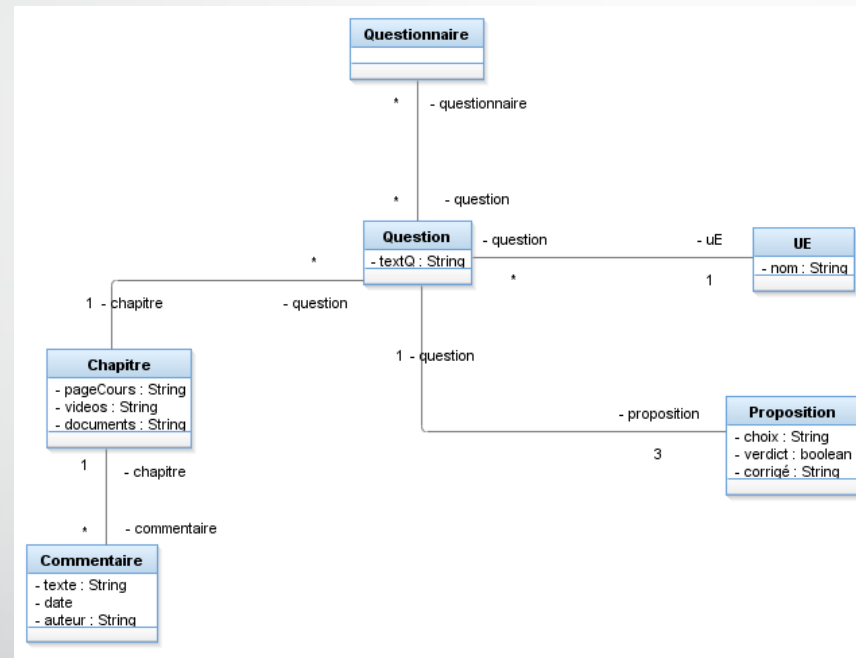
Branch: master [New pull request](#) [Create new file](#) [Upload files](#) [Find file](#) [Clone or download](#)

	Krihox debut diapo + screenshots a ajouter	Latest commit 72be042 2 hours ago
	Diaporama	debut diapo + screenshots a ajouter 2 hours ago
	Fiches_detaillees	premier commit, ajout fichiers + projet 3 hours ago
	Projet	ajout des diagrammes sequences concernant les commentaires + classe s... 2 hours ago
	Tests_validation	premier commit, ajout fichiers + projet 3 hours ago
	LICENSE	Initial commit 4 hours ago
	README.md	Initial commit 4 hours ago

# Diagramme Use Case



# Diagramme métier



# Fiche Détaillée Passer QCM

**Nom du Use Case :**

Passer QCM

**Date/Version/Auteurs :**

02/10 / v1 / Marco et Boris

**Description :**

Permet de passer un QCM

**Acteurs :**

Etudiant

**Pré-conditions :**

VIDE

**Scénario Nominal :**

- 1- L'étudiant choisit 'Passer QCM'
- 2- Le système affiche la liste des UE
- 3- L'étudiant choisit l'UE
- 4- Le système affiche la liste des chapitres
- 5- L'étudiant choisit le chapitre
- 6- Le système demande le nombre de questions
- 7- L'étudiant sélectionne le nombre de questions souhaité
- 8- Le système génère un QCM correspondant aux choix de l'étudiant
- 9- L'étudiant passe le QCM
- 10- Le système calcule le résultat, l'affiche et propose de repasser un QCM
- 11- L'étudiant refuse

**Post-conditions :**

VIDE]

# Fiche Détaillée Gérer Question

**Nom du Use Case :**

Gérer question

**Date/Version/Auteurs :**

02/10 / v1 / SHAO Wu - JESUS DA SILVA Steven

**Description :**

Permet aux enseignants d'ajouter, de modifier ou de supprimer une questions à une UE.

**Acteur :**

Enseignant

**Pré-conditions :**

L'enseignant est déjà connecté au système.

**Scénario Nominal :**

1. L'enseignant se connecte au site web.
2. Le système affiche la page web qui contient un bouton "gérer question".
3. L'enseignant clique sur le bouton "ajouter question".
4. Le système demande à l'enseignant de saisir un texte de question, l'UE à laquelle la question appartient, le chapitre concerné et au moins trois propositions de réponse possibles composé d'un texte, d'un verdict et d'un texte justificatif dont au moins une doit être valide.
5. L'utilisateur remplit les champs et valide.
6. Le système affiche un succès et retour à la page d'accueil.

**Post-conditions :**

Le système à mémoriser la question.

**Enchaînements alternatifs :**

A1. Un des champs n'est pas correctement rempli.

L'enchaînement démarre après le point 5 de la séquence nominale :

6. Le système détecte qu'un des champs n'est pas rempli, retour sur la séquence nominal 4.

A2. L'enseignant souhaite modifier une question

L'enchaînement démarre après le point 2 de la séquence nominale :

3. L'enseignant clique sur le bouton "modifier question"
4. Le système affichela liste des questions existantes
5. L'enseignant sélectionne la question qu'il veut modifier
6. Le système affiche le formulaire de la question avec les champs remplis
7. L'enseignant fait des modifications et valide.

(A2.1 Voir A1)

A3. L'enseignant souhaite supprimer une questions

L'enchaînement démarre après le point 2 de la séquence nominale :

3. L'enseignant clique sur le bouton "supprimer question"
4. Le système affichela liste des questions existantes
5. L'enseignant sélectionne la question qu'il veut supprimer
6. L'enseignant valide.

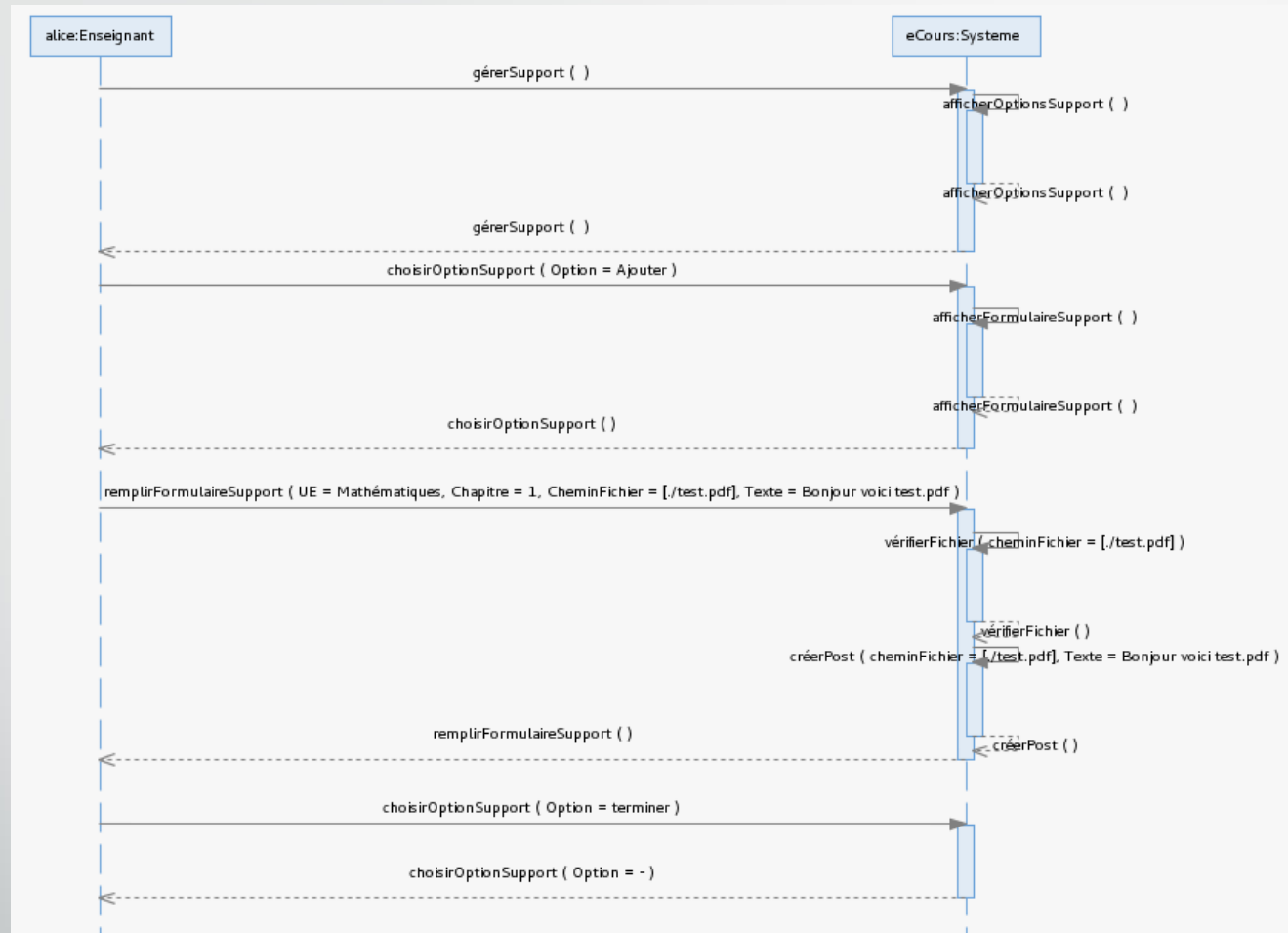
**Enchaînement exceptions:**

# Diagramme de séquence Gérer question

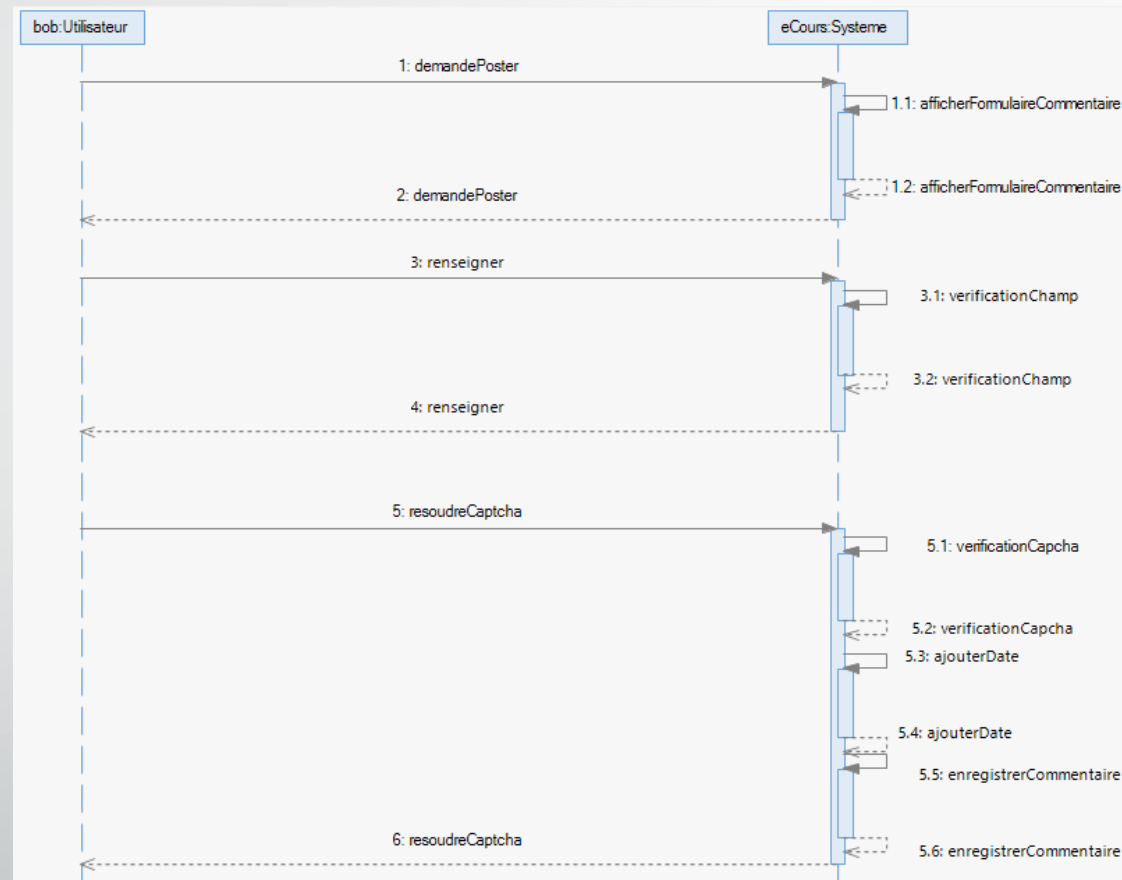




# Diagramme de séquence Gérer support



# Diagramme de séquence Poster Commentaire



**CodeFactory**

# Test de validation Poster Commentaire

Titre	Un utilisateur veut poster un commentaire	ID TVo8
Contexte	Un étudiant « etu »	
Entrée	∅	
Scénario	<ol style="list-style-type: none"><li>1. Le testeur choisit "Poster un commentaire"</li><li>2. Le testeur saisit un nom « etu » et son commentaire « Mon premier commentaire »</li><li>3. Le testeur résout le captcha « X23VR »</li></ol>	
Résultat attendu	Le commentaire est enregistré. On peut voir le nouveau commentaire dans la liste.	
Moyens de vérifications	Visuel, « mon premier commentaire » posté par « etu » apparaît dans la liste de commentaires	

```
System
+ choisirUE ( nomUE : String )
+ rechercheUE ( nomUE : String ) : UE
+ afficherListeChap ( ue : UE )
+ choisirChap ( nomChap : String )
+ rechercheChap ( nomChap : String ) : Chapitre
+ afficheSupport ( chap : Chapitre )
+ gererUE ( )
+ afficheListeUE ( )
+ ajouterUE ( nom : String, desc : String )
+ enregistrerUE ( nom : String, desc : String ) : boolean
+ gererQuestion ( )
+ afficherOptionsQuestion ( )
+ choisirOptionQuestion ( Option : String )
+ afficherFormulaireQuestion ( )
+ remplirFormulaireQuestion ( UE : UE, Chapitre : Chapitre, Texte : String [ * ], Propositions : String [ 3.. * ] )
+ validerQuestion ( UE : UE, Chapitre : Chapitre, Texte : String, Propositions : String [ 3.. * ] )
+ passerQCM ( )
+ afficherSelectionNombreQuestions ( )
+ choisirNombreQuestion ( Nombre : Integer )
+ genererQCM ( UE : UE, Chapitre : Chapitre, NombreQuestion : Integer )
+ afficherQuestion ( NumQuestion : Integer )
+ repondreQuestion ( Réponse : Boolean [ 3.. * ] )
+ verifierRéponse ( NumQuestion : Integer, Réponse : Boolean [ 3.. * ] ) : Boolean
+ afficherRésultat ( Validité : Boolean, NumQuestion : Integer )
+ passerQuestionSuivante ( )
+ calculerNote ( Réponses : Boolean [ 3.. * ] )
+ gererSupport ( )
+ afficherListeCours ( )
+ afficherOptionsSupport ( )
+ choisirOptionSupport ( Option : String )
+ afficherFormulaireSupport ( )
+ remplirFormulaireSupport ( UE : UE, Chapitre : Chapitre, CheminFichier : String, Texte : String )
+ verifierFichier ( cheminFichier : String [ * ] )
+ creerPost ( cheminFichier : String, Texte : String [ * ] )
+ afficherFormulaireCommentaire ( )
+ remplirFormulaireCommentaire ( User : String, Texte : String, Captcha : String )
+ enregistrerCommentaire ( User : String, Texte : String )
+ demandeSuppressionCommentaire ( id : int )
+ afficherFormulaireConfirmation ( )
+ confirmer ( )
+ supprimerMessage ( id : Integer )
+ ajouterChapitre ( )
+ demandeNouveauNom ( )
+ donnerNom ( nom : String )
+ enregistrerChapitre ( nom : String )
+ supprimerChapitre ( nom : String )
+ afficheFormulaireChapitre ( )
+ choisirModifierChapitre ( nom : String )
+ modifierNomChapitre ( nom : String )
```

**CodeFactory**