



N° Réf :.....

Centre Universitaire
Abdelhafid Boussouf Mila

Institut des Sciences et Technologie

Département de Mathématiques et Informatique

Mémoire préparé en vue de l'obtention du diplôme de Master

En : Informatique

Spécialité: Sciences et Technologies de l'Information et de la
Communication (STIC)

Conception et implémentation d'une application mobile pour la gestion du Contrôle continu

Préparé par Laouar Sabrine.

Melghid Nour Elhouda.

Soutenue devant le jury

Président	Hettab Abdlkamel.	MAA .Université	C.U.Abd Elhafid Boussouf
Examineur	Selmane Samir.	MAA.Université	C.U.Abd Elhafid Boussouf
Encadreur	Bessouf Hakim.	MAA.Université	C.U.Abd Elhafid Boussouf

Année Universitaire : 2018/2019

Remerciement

C'est avec l'aide de Dieu qu'a vu le jour ce présent travail. En premier lieu, nous tenons à remercier toujours et par cette occasion et avant tout, «**ALLAH**», qui nous a donné la force et la patience d'accomplir ce modeste travail. Ensuite, ce travail n'aurait pas pu être achevé sans le soutien, les conseils et les encouragements de certaines personnes auxquelles nous tenons ici à exprimer nos sincères remerciements. Nous exprimons toute notre gratitude pour notre promoteur, Monsieur «**Mr Hakim Bessouf** » pour ses précieux conseils, sa disponibilité, la confiance qu'il nous a toujours témoignée et la sollicitude dont il nous a entourés, et ce tout au long de l'élaboration du présent travail. Nous n'oublions pas non plus nos enseignants, qui tout au long du cycle d'études au centre universitaire de MILA, particulièrement Monsieur «**Bencheikh elhocine Madjed** » nous ont transmis leur savoir. Nous adressons une pensée particulièrement affective à nos amis de CUM qui ont rendu agréables nos longues années d'études. Nous tenons enfin à remercier tous ceux qui ont collaboré de près ou de loin à l'élaboration de ce travail. spécialement «**Bendas Mohammed Amine** ».

Qu'ils acceptent nos humbles remerciements.

Sabrina et Chaima



Dédicace

*À mes très chers **Parents** pour leur affection, leur patience, leur soutien et leurs encouragements qui m'ont permis d'arriver au bout de ce travail.*

*À mes frères que je les aime énormément : **Abedassatar, Abderazak, Abderahman.***

*À mes soeurs : **Zeyneb, Hadir, Abir, Lamya & Maymouna.***

*À mes petits-enfants de la famille : **Didou, Bahaa & Ritale.***

*À ma belle **Imene Keriout.***

*À mon binôme **Sabrine** que j'estime beaucoup.*

*À tous mes amis surtout **Samir Boulekssoub.***

À tous mes collègues de promotion.

À toute ma famille.

Chaima

*À mes très chers **Parents** pour leur affection, leur patience, leur soutien et leurs encouragements qui m'ont permis d'arriver au bout de ce travail*

*À mes frères que je les aime énormément : **Morchid & Ramy.***

*À mes soeurs : **Marwa & Manar.***

*À ma belle **Imene Keriout.***

*À mon binôme **Chaima** que j'estime beaucoup.*

*À tous mes amis surtout **Khadidja, Nahla & Lina.***

*À **Bendjabeur Abderrahman (kheyrou).***

À tous mes collègues de promotion.

À toute ma famille.

Sabrine

Table des matières

Table des Matières	4
Table des figures	6
Liste des tableaux	12
Liste des abréviations	17
Résumé	18
Introduction générale	22
1 Étude préliminaire	24
Introduction	24
1.1 Élaboration du cahier des charges	25
1.1.1 La présentation du projet	25
1.1.2 Les grands choix techniques	26
1.1.3 Recueil des besoins fonctionnels	26
1.1.4 Recueil des besoins opérationnels	28
1.2 Description de contexte du système à réaliser	28
1.2.1 Identifications des acteurs	28
1.2.2 Identifications des messages	29
1.2.3 Le diagramme de contexte du système	30
Conclusion	32

2	Capture des besoins	33
	Introduction	33
2.1	Situation de l'étape dans le processus 2TUP	33
2.2	Capture des besoins fonctionnels	34
2.2.1	Identification des cas d'utilisation	35
2.2.2	Description des cas d'utilisation	38
2.2.3	Identification des classes candidates	60
2.2.4	Diagrammes de classes participantes	66
2.3	Capture des besoins techniques	79
2.3.1	La spécification technique du point de vue matériel	79
2.3.2	La capture des spécifications logicielles	80
	Conclusion	83
3	Phase d'analyse	85
	Introduction	85
3.1	Découpage en catégories	86
3.1.1	La répartition des classes candidates en catégories	86
3.1.2	Élaboration des diagrammes de classes préliminaires par catégorie	87
3.1.3	Dépendance entre catégories	89
3.2	Développement du modèle statique	90
3.2.1	Diagramme de classes de la catégorie « Acteurs »	90
3.2.2	Diagramme de classes de la catégorie « Session »	90
3.2.3	Diagramme de classes de la catégorie « Évaluation »	91
3.3	Développement du modèle dynamique	92
3.3.1	Diagrammes de séquence détaillés	92
	Conclusions	104
4	Conception et Réalisation	105
	Introduction	105
4.1	La conception	105
4.1.1	Dictionnaire de données	106
4.1.2	Diagramme de classe détaillé	107
4.1.3	Le modèle relationnel	108

4.1.4	Le passage de modèle objet au modèle relationnel	108
4.2	La réalisation	111
4.2.1	Environnement de développement	111
4.2.2	Les bases de données	112
4.2.3	Présentation de l'application réalisée	114
	Conclusion	125
	Conclusion générale	127
	Bibliographie	128
	Annexe	132
	Informatique mobile et les applications mobiles	132
	Conception	140

Table des figures

1.1	Situation de l'étude préliminaire dans 2TUP	25
1.2	Diagramme de contexte du système «Contrôle continu»	31
2.1	Situation du capture des besoins dans 2TUP	34
2.2	Diagramme de cas d'utilisation générale	37
2.3	Diagramme de séquence du cas S'inscrire	39
2.4	Diagramme de séquence du cas Ajouter groupe	41
2.5	Diagramme de séquence du cas Ajouter étudiant	43
2.6	Diagramme de séquence du cas Consulter test et corrigé type	44
2.7	Diagramme de séquence du cas Ajouter test et corrigé type	46
2.8	Diagramme de séquence du cas Créer session	48
2.9	Diagramme de séquence du cas Lancer session	50
2.10	Diagramme de séquence du cas Consulter notes	51
2.11	Diagramme de séquence du cas Partager notes	52
2.12	Diagramme de séquence du cas Ajouter éditeur	54
2.13	Diagramme de séquence du cas Modifier éditeur	56
2.14	Diagramme de séquence du cas Passer test	58
2.15	Diagramme de séquence du cas Consulter son test	59
2.16	Les responsabilités de la classe candidate « Utilisateur »	61
2.17	Les responsabilités de la classe candidate « Enseignant »	61
2.18	Les responsabilités de la classe candidate « Étudiant »	62
2.19	Les responsabilités de la classe candidate « Agent de MAJ »	62
2.20	Les responsabilités de la classe candidate « Groupe »	63
2.21	Les responsabilités de la classe candidate « Test »	63

2.22	Les responsabilités de la classe candidate « Question »	64
2.23	Les responsabilités de la classe candidate « Choix réponse »	64
2.24	Les responsabilités de la classe candidate « Session »	65
2.25	Les responsabilités de la classe candidate « Éditeur »	65
2.26	Diagramme de classes participantes du cas S’inscrire	66
2.27	Diagramme de classes participantes du cas S’authentifier	67
2.28	Diagramme de classes participantes du cas Ajouter groupe	68
2.29	Diagramme de classes participantes du cas Ajouter étudiant	69
2.30	Diagramme de classes participantes du cas Consulter test et corrigé type	70
2.31	Diagramme de classes participantes du cas Ajouter test et corrigé type	71
2.32	Diagramme de classes participantes du cas Créer session	72
2.33	Diagramme de classes participantes du cas Consulter session	73
2.34	Diagramme de classes participantes du cas Consulter notes	74
2.35	Diagramme de classes participantes du cas Ajouter éditeur	75
2.36	Diagramme de classes participantes du cas Modifier éditeur	76
2.37	Diagramme de classes participantes du cas Passer test	77
2.38	Diagramme de classes participantes du cas Consulter son test	78
2.39	Architecture à 2 niveaux	80
2.40	Modèle de spécification logicielle du système.	81
2.41	Diagramme de séquence du cas S’authentifier	82
2.42	Diagramme de séquence du cas Gérer l’intégrité des données	83
3.1	Situation de la phase d’analyse dans 2TUP.	86
3.2	Découpage en catégories	86
3.3	Diagramme des classes préliminaire de la Catégorie « Acteurs »	87
3.4	Diagramme des classes préliminaire de la Catégorie « Session »	87
3.5	Diagramme des classes préliminaire de la Catégorie « Évaluation » . .	88
3.6	Dépendance entre catégories	89
3.7	Diagramme de classes de la catégorie « Acteurs »	90
3.8	Diagramme de classes de la catégorie « Session »	90
3.9	Diagramme de classes de la catégorie « Évaluation »	91
3.10	Diagramme de séquence détaillé du cas Gérer test et corrigé type . . .	93
3.11	Diagramme de séquence détaillé du cas Consulter test et corrigé type .	94

3.12	Diagramme de séquence détaillé du cas Ajouter test et corrigé type . .	95
3.13	Diagramme de séquence détaillé du cas Modifier test et corrigé type .	96
3.14	Diagramme de séquence détaillé du cas Supprimer test et corrigé type	97
3.15	Diagramme de séquence détaillé du cas Archivé test et corrigé type . .	98
3.16	Diagramme de séquence détaillé du cas Gérer session	99
3.17	Diagramme de séquence détaillé du cas Consulter session	100
3.18	Diagramme de séquence détaillé du cas Créer session	101
3.19	Diagramme de séquence détaillé du cas Modifier session	102
3.20	Diagramme de séquence détaillé du cas Supprimer session	103
4.1	Situation de la conception dans le processus 2TUP	105
4.2	Diagramme de classe détaillé	107
4.3	Logo du langage java	111
4.4	Logo de l'IDE Android Studio	112
4.5	Logo du WampServer	113
4.6	Logo de Langage de Manipulation de Bases de Données MySQL	113
4.7	Logo d'éditeur de texte Notepad++	114
4.8	Logo des différentes applications réalisées	114
4.9	Espace d'inscription	115
4.10	Authentification	115
4.11	Accueil enseignant	115
4.12	Modifier profil	115
4.13	Gérer la liste des groupes enseignés	116
4.14	Ajouter groupe	116
4.15	Ajouter étudiant cas d'erreur 1	116
4.16	Ajouter étudiant cas d'erreur 2	116
4.17	Ajouter étudiant cas d'erreur 3	117
4.18	Ajouter étudiant "Ajout avec succès"	117

4.19	Ajouter groupe	
	"Liste des étudiants"	117
4.20	Modifier étudiant	117
4.21	Consulter étudiant	118
4.22	Modifier groupe	118
4.23	Consulter groupe	118
4.24	Gérer tests et	
	corrigés type	118
4.25	Ajouter test	119
4.26	Choisir style question	119
4.27	Ajouter question	119
4.28	Ajouter test	
	"Liste des questions"	119
4.29	Consulter test	120
4.30	Consulter question	120
4.31	Gérer session	120
4.32	Créer session	120
4.33	Consulter session	121
4.34	Gérer archive des tests	121
4.35	Espace d'inscription	122
4.36	Authentification	122
4.37	Accueil Agent de MAJ	122
4.38	Modifier profil	122
4.39	Gérer éditeur	123
4.40	Ajouter éditeur	123
4.41	Modifier éditeur	123
4.42	Consulter éditeur	123
4.43	Espace d'inscription	124
4.44	Authentification	124
4.45	Accueil Étudiant	124
4.46	Participation test	124

4.47	Passer test	
	« question 1 »	125
4.48	Passer test	
	« question 2 »	125
49	Architecture de la plateforme android	137
50	le cycle de vie d'une application Android	138
51	Diagramme de séquence de cas Modifier profil	141
52	Diagramme de séquence du cas Gérer la liste des groupes enseignés . .	143
53	Diagramme de séquence du cas Consulter groupe	144
54	Diagramme de séquence du cas Modifier groupe	146
55	Diagramme de séquence du cas Supprimer groupe	148
56	Diagramme de séquence du cas Imprimer	149
57	Diagramme de séquence du cas Importer la liste des étudiants	150
58	Diagramme de séquence du cas Gérer étudiants	152
59	Diagramme de séquence du cas Consulter étudiant	153
60	Diagramme de séquence du cas Modifier étudiant	155
61	Diagramme de séquence du cas Supprimer étudiant	157
62	Diagramme de séquence du cas Gérer tests et corrigés type	159
63	Diagramme de séquence du cas Modifier test et corrigé type	161
64	Diagramme de séquence du cas Supprimer test et corrigé type	163
65	Diagramme de séquence du cas Archiver test et corrigé type	165
66	Diagramme de séquence du cas Gérer session	167
67	Diagramme de séquence du cas Consulter session	168
68	Diagramme de séquence du cas Modifier session	170
69	Diagramme de séquence du cas Supprimer session	172
70	Diagramme de séquence du cas Gérer archive des tests	174
71	Diagramme de séquence du cas Consulter test de l'archive	175
72	Diagramme de séquence du cas Restaurer test de l'archive	177
73	Diagramme de séquence du cas Gérer éditeur de style de question . . .	179
74	Diagramme de séquence du cas d'utilisation Consulter éditeur	180
75	Diagramme de séquence du cas Supprimer éditeur	182
76	Diagramme de séquence du cas Consulter archive des tests	184

77	Diagramme de classes participantes du cas Modifier profil	185
78	Diagramme de classes participantes du cas Gérer la liste des groupe enseignés	186
79	Diagramme de classes participantes du cas Consulter groupe	187
80	Diagramme de classes participantes du cas Modifier groupe	188
81	Diagramme de classes participantes du cas Supprimer groupe	189
82	Diagramme de classes participantes du cas Gérer étudiants	190
83	Diagramme de classes participantes du cas Consulter étudiant	191
84	Diagramme de classes participantes du cas Modifier étudiant	191
85	Diagramme de classes participantes du cas Supprimer étudiant	192
86	Diagramme de classes participantes du cas Gérer tests et corrigés type .	193
87	Diagramme de classes participantes du cas Modifier test et corrigé type	194
88	Diagramme de classes participantes du cas Supprimer test et corrigé type	195
89	Diagramme de classes participantes du cas Archiver test et corrigé type	196
90	Diagramme de classes participantes du cas Gérer session	197
91	Diagramme de classes participantes du cas Modifier session	198
92	Diagramme de classes participantes du cas Supprimer session	199
93	Diagramme de classes participantes du cas Gérer archive des tests . . .	200
94	Diagramme de classes participantes du cas Consulter test de l'archive .	201
95	Diagramme de classes participantes du cas Restaurer test de l'archive .	202
96	Diagramme de classes participantes du cas Gérer éditeur de style question	203
97	Diagramme de classes participantes du cas Consulter éditeur	204
98	Diagramme de classes participantes du cas Supprimer éditeur	204

Liste des tableaux

1.1	Légende des messages de diagramme de contexte	31
2.1	Liste des cas d'utilisations fonctionnels	36
2.2	Description textuelle du cas d'utilisation « S'inscrire »	38
2.3	Description textuelle du cas d'utilisation « Ajouter groupe »	40
2.4	Description textuelle du cas d'utilisation « Ajouter étudiant ».	42
2.5	Description textuelle du cas d'utilisation « Consulter test et corrigé type »	44
2.6	Description textuelle du cas d'utilisation « Ajouter test et corrigé type »	45
2.7	Description textuelle du cas d'utilisation « Créer session »	47
2.8	Description textuelle du cas d'utilisation « Lancer session »	49
2.9	Description textuelle du cas d'utilisation « Consulter notes »	51
2.10	Description textuelle du cas d'utilisation « Partager notes »	52
2.11	Description textuelle du cas d'utilisation « Ajouter éditeur »	53

2.12	Description textuelle du cas d'utilisation	
	« Modifier éditeur »	55
2.13	Description textuelle du cas d'utilisation	
	« Passer test »	57
2.14	Description textuelle du cas d'utilisation	
	« Consulter son test »	59
2.15	liste des classes candidates	60
2.16	Description textuelle du cas d'utilisation	
	« S'authentifier »	81
2.17	Description textuelle du cas d'utilisation	
	« Gérer l'intégrité des données »	83
4.1	Dictionnaire de données	106
4.2	Équivalence entre les concepts objets et relationnels	109
4.3	Les relations de la base de données de notre système	110
4	Les versions d'Android.	135
5	Description textuelle du cas d'utilisation	
	« Modifier profil »	140
6	Description textuelle du cas d'utilisation	
	« Gérer la liste des groupes enseignés »	142
7	Description textuelle du cas d'utilisation	
	« Consulter groupe »	144
8	Description textuelle du cas d'utilisation	
	« Modifier groupe »	145
9	Description textuelle du cas d'utilisation	
	« Supprimer groupe »	147
10	Description textuelle du cas d'utilisation	
	« Imprimer »	149
11	Description textuelle du cas d'utilisation	
	« Importer la liste des étudiants »	150
12	Description textuelle du cas d'utilisation	
	« Gérer étudiants »	151

13	Description textuelle du cas d'utilisation	
	« Consulter étudiant »	153
14	Description textuelle du cas d'utilisation	
	« Modifier étudiant »	154
15	Description textuelle du cas d'utilisation	
	« Supprimer étudiant »	156
16	Description textuelle du cas d'utilisation	
	« Gérer tests et corrigés type »	158
17	Description textuelle du cas d'utilisation	
	« Modifier test et corrigé type »	160
18	Description textuelle du cas d'utilisation	
	Supprimer test et corrigé type	162
19	Description textuelle du cas d'utilisation	
	Archiver test et corrigé type	164
20	Description textuelle du cas d'utilisation	
	« Gérer sessions »	166
21	Description textuelle du cas d'utilisation	
	« Consulter session »	168
22	Description textuelle du cas d'utilisation	
	« Modifier session »	169
23	Description textuelle du cas d'utilisation	
	« Supprimer session »	171
24	Description textuelle du cas d'utilisation	
	« Gérer archive des tests »	173
25	Description textuelle du cas d'utilisation	
	« Consulter test de l'archive »	175
26	Description textuelle du cas d'utilisation	
	« Restaurer test de l'archive »	176
27	Description textuelle du cas d'utilisation	
	« Gérer éditeur de style de question »	178
28	Description textuelle du cas d'utilisation	
	« Consulter éditeur »	180

29	Description textuelle du cas d'utilisation	
	« Supprimer éditeur »	181
30	Description textuelle du cas d'utilisation	
	« Consulter archive des tests »	183

Liste des abréviations

[STIC] : Sciences et technologies de l'information et de la communication

[LMD] : Licence, Master, Doctorat

[2TUP] : 2 Track Unified Process

[UML] : Unified Modeling language

[MySQL] : My Structured Query Language

[XML] : Extensible Markup Language

[SDK] : Software Development Kit

[WAMP] : Windows Apache MYSQL PHP

[SGBD] : Système de gestion de base de données

[PHP] : Hypertext Preprocessor

[HTML] : Hypertext Markup Language

[CSS] : Cascading Style Sheets

Résumé

Le thème abordé dans ce projet de fin d'études consiste à la conception et l'implémentation d'une application mobile pour la gestion de contrôle continu des étudiants pendant un semestre d'une année universitaire. L'application réalisée a un objectif principal qui permet d'effectuer plusieurs contrôles continus pendant le semestre, et pour chaque contrôle une évaluation automatique et même immédiate, dans le but d'optimiser le temps de passage des tests et de corrections des copies.

Pour atteindre ces objectifs nous avons choisi de modéliser avec le formalisme UML2.0 et la méthode 2TUP comme une démarche de conception, notre choix s'est porté par rapport à sa stabilité et son indépendance vis-à-vis des langages d'implémentation. Afin de réaliser notre application, nous avons utilisé l'outil Android studio et «MySQL» comme SGBD . Quant à la réalisation des interfaces, nous avons utilisé le langage XML ainsi que JAVA.

Mots clés : Application mobile, contrôle continu, UML2.0, 2TUP, MySQL ,JAVA, ANDROID.

Abstract

Summary The theme of this graduation project is the design and implementation of a mobile application for the continuous monitoring management of students during a semester of an academic year. The application has one main objective : to achieve several continuous checks during the semester, and for each check an automatic and even immediate evaluation, in order to optimize the time required for testing and correction of copies. To achieve these objectives, we chose to model with the UML2.0 formalism and the 2TUP method as a design approach, our choice was based on its stability and independence from implementation languages. In order to create our application, we used the Android studio tool and "MySQL" as SGBD. As for the realization of the interfaces, we used XML and JAVA.

Keywords : Mobile application, continuous monitoring, UML2.0, 2TUP, MySQL ,JAVA, ANDROID.

الملخص

يُتمثل الموضوع الذي تم تناوله في مشروع نهاية الدراسات في تصميم وتنفيذ تطبيق للهاتف المحمول لإدارة التقييم المستمر للطلاب خلال فصل دراسي من السنة الجامعية. التطبيق الذي تم تنفيذه له هدف رئيسي والذي يجعل من الممكن تنفيذ العديد من الامتحانات خلال الفصل الدراسي، ولكل امتحان تقييم تلقائي وفوري، من أجل تحسين وقت اجتياز الاختبار وتصحيح النسخ.

ولتحقيق هذه الأهداف، اخترنا UML2.0 كإسلوب و2TUP كنهج تصميم، ولقد وقع اختيارنا هذا على أساس استقرار واستقلال هذه الأساليب مع لغات التنفيذ

من أجل تحقيق هذا التطبيق، استخدمنا Android studio وMysql كـ SGBD.

أما بالنسبة للغات البرمجة استعملنا XML و JAVA

الكلمات الرئيسية : تطبيقات الهاتف المحمول ، التحكم المستمر ، ANDROID ، JAVA ، MySQL ، 2TUP ، UML2.0.

Introduction Générale

Le nouveau système de l'enseignement supérieur a voulu apporter des réformes pour améliorer la performance de l'opération de formation et la qualité des formations. Le concept du contrôle continu des étudiants notamment dans le système LMD est l'un des concepts importants apportés par ces réformes.

L'évaluation continue des étudiants est comprise comme l'ensemble des procédures destinées à mesurer les résultats d'apprentissage des étudiants en matière de connaissances assimilées, de compréhension développée et de compétences acquises. qui sont appréciées semestriellement soit par un contrôle continu et régulier soit par un examen final ou bien par les deux modes combinés.

La note de contrôle continu doit refléter l'effort dépensé par l'étudiant tout au long de la période d'évaluation d'un côté pour encourager les gens qui travaillent et les récompenser et d'un autre côté pour motiver les gens qui travaillent moins.

La réalité des choses affirme qu'il est très difficile d'avoir une évaluation juste dans le contexte de ce travail, puisque L'évaluation Continue est soumise à des règles strictes telles que : le nombre élevé des étudiants dans les classes et le nombre important de paramètres qui rentre avec ces règles comme :

- Le facteur de temps (la chronologie de la durée qui ne doit pas dépasser 30 minutes).
- Le nombre des copies qui est grand.
- la correction et les notes qui doivent être assurées la semaine qui suit le contrôle ...etc

Au cœur de ces critiques, notre travail consiste à développer une application mobile destinée principalement d'un côté, à l'enseignant qui permet d'effectuer plusieurs contrôles continus pendant le semestre, et d'un autre côté, à l'étudiant qui doit avoir pour chaque contrôle une évaluation automatique et même immédiate, dont l'objectif est de faciliter certaines tâches :

- ★ Optimiser le temps de passage.
- ★ Réduire le temps de corrections des copies chez l'enseignant.
- ★ Donner une évaluation immédiate pour chaque étudiant.

Pour réaliser cette application, on a choisi le langage UML2.0 avec la démarche 2TUP pour la modélisation, ainsi on a utilisé JAVA sous plateforme Android Studio pour l'implémentation, avec MySQL pour la gestion des bases de données.

Nous avons organisé notre mémoire comme suit :

Le chapitre 01 : est un aperçu sur le sujet de notre étude ainsi que présentation du cahier des charges qui consiste à effectuer un premier repérage des besoins fonctionnels et opérationnels.

Le chapitre 02 : est consacré pour la capture des besoins qui comportent deux étapes : la capture des besoins fonctionnels et celles des besoins techniques.

Le chapitre 03 : est une phase d'analyse de notre système qui le découpe en catégories jusqu'à l'arrivée au modèle dynamique.

Le chapitre 04 : est dédiée à la phase de conception du système ainsi que la réalisation.

la phase de conception, qui présente la conception détaillée des classes et leurs associations qui construiront notre système et décrit le passage du modèle à objet au modèle relationnel.

La phase de réalisation dans laquelle nous définissons les outils de développement que nous avons utilisés et nous illustrerons également quelques interfaces de l'application mise en œuvre.

Enfin, nous achevons notre mémoire par une conclusion générale et quelques perspectives.

Étude préliminaire

Introduction

L'étude préliminaire est la première phase de notre processus de développement. Elle consiste à effectuer un repérage des besoins fonctionnels et opérationnels, en utilisant principalement des textes, ou des diagrammes très simples. Elle prépare les activités les plus formelles comme la capture des besoins fonctionnels et techniques.

Au niveau de ce chapitre nous commencerons à déterminer les besoins fonctionnels en considérant le système comme une boîte noire, afin d'étudier sa place. Après avoir identifié les acteurs qui interagissent avec le système, nous développerons un premier modèle UML au niveau contexte, pour pouvoir établir précisément les frontières fonctionnelles du système.

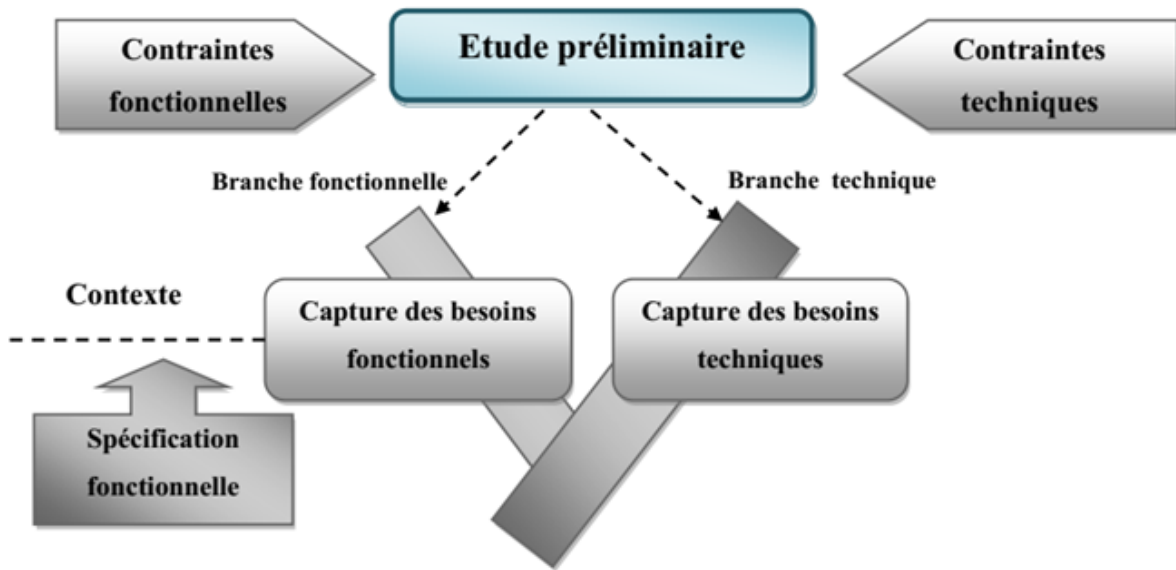


FIGURE 1.1 – Situation de l'étude préliminaire dans 2TUP

1.1 Élaboration du cahier des charges

Un cahier des charges est un document qui doit être respecté lors de la réalisation d'un projet. Il a pour objectif de définir et de décrire la spécifications d'un projet à réaliser. [1]

L'élaboration du cahier des charges s'effectue sur plusieurs étapes :

- la présentation du projet.
- Les grands choix techniques.
- Le recueil des besoins fonctionnels
- Le recueil des besoins opérationnels.

1.1.1 La présentation du projet

• Équipe du projet

Ce projet est à réaliser par les deux étudiantes Sabine LAOUR et NoorElhouda MELGHID qui sont inscrits dans la spécialité 'STIC' en informatique master2 à l'Institut de science et technologie du centre universitaire de Mila.

• Encadrement

Ce travail est dirigé par Mr. Hakim Bessouf. Maitre Assistant classe A au même établissement. C'est lui qui a développé l'idée du projet.

- **Cadre de réalisation**

Ce projet est réalisé dans le cadre de la préparation d'un diplôme master2 spécialité 'STIC' en informatique à l'institut des sciences et technologies du centre universitaire de Mila.

- **Problématique**

Dans le système LMD, les aptitudes et les acquisitions de connaissances sont appréciées semestriellement soit par un contrôle continu et régulier soit par un examen final ou bien par les deux modes combinés, et c'est ce qui est normalement en vigueur. et comme l'évaluation continue est soumise à des règles strictes telles que : la durée qui ne doit pas dépasser 30 minutes, la correction et les notes qui doivent être assurées la semaine qui suit le contrôle, elle est un peu difficile à effectuer de la manière qui permet une évaluation précise de l'étudiant. Ce dernier devrait obtenir une rétroaction instantanée à sa solution afin qu'il puisse se rendre compte des erreurs commises et les rectifier et avancer ainsi dans son apprentissage.

- **Objectif**

Notre travail consiste à concevoir et implémenter une application mobile qui permet d'effectuer plusieurs contrôles continus pendant le semestre, et avoir pour chaque contrôle une évaluation automatique et même immédiate, dont l'objectif d'optimiser le temps de passage et de corrections des copies et d'établir une évaluation exacte pour chaque étudiant.

1.1.2 Les grands choix techniques

Nous allons utiliser d'un certain nombre de techniques-clés qui sont principalement :

- le langage JAVA pour la programmation de la logique applicative de ce projet dans une plateforme Android.
- L'adoption d'une architecture client/serveur.
- Pour la modélisation on va utiliser la méthode de développement 2TUP à partir de langage standard UML.
- Le SGBD MYSQL pour le stockage et la gestion des données.

1.1.3 Recueil des besoins fonctionnels

Afin de répondre aux attentes des futurs utilisateurs de notre application nous avons effectué plusieurs recherches pour identifier au mieux les limites et besoins des différents acteurs.

Des interviews et des discussions régulières avec des enseignants nous ont permis de recueillir les besoins fonctionnels préliminaires suivants :

- **Contrôle continu**

C'est un test préparé par l'enseignant, et qui doit être passé par les étudiants, durant une séance de TD, TP afin d'établir une meilleure évaluation pour chaque étudiant.

- **Gestion des étudiants**

L'enseignant peut gérer les étudiants (consulter, ajouter, modifier, supprimer) qui lui sont affectés, selon leurs groupes, d'une façon globale (groupe) ou individuelle (étudiant).

- **Gestion des sessions**

L'enseignant peut gérer les sessions (créer, modifier, supprimer) en affectant à chaque session un ou plusieurs groupes, un test préparé auparavant, et une durée de passage de test.

- **Gestion des tests et corrigé type**

L'enseignant peut gérer les tests et leurs corrigés types (Consulter, ajouter, modifier, Supprimer, archiver) selon le choix du style de la question (QCM, calcul, ...).

- **Lancement de session**

Le lancement de la session est effectué en activant cette session préparée auparavant, par l'enseignant.

- **Gestion d'archives des tests**

L'enseignant peut gérer l'archive des tests (consulter ou supprimer des tests de l'archive) que l'étudiant peut consulter dans la plateforme elearning.

- **Consultation des notes**

L'enseignant peut consulte les notes des étudiants pour chaque groupe enseigné.

- **Passage du test**

C'est passer un test durant une séance. Il est effectué par l'étudiant, durant une session.

1.1.4 Recueil des besoins opérationnels

Les besoins opérationnels représentent les besoins non fonctionnels, qui caractérisent le système comme la performance ainsi que la sécurité et l'ergonomie du système.

Alors, pour notre application, on a tiré certains besoins pour avoir un fonctionnement de bonne qualité, on peut les résumés comme suit :

- **La sécurité** : les données des utilisateurs doit être protégée. Chaque utilisateur de l'application doit s'authentifier par un nom d'utilisateur et un mot de passe.
- **L'ergonomie** : les interfaces de l'application doivent être ergonomique. chaque interface de cette application doit être simple et claire, elle doit être bien organisé du point de vue graphique, le choix des couleurs, et des styles.
- **La disponibilité** : l'application doit être clair, facile et disponible pour utiliser par n'importe quel utilisateur.

1.2 Description de contexte du système à réaliser

Une fois ce premier recueil de besoins effectué, la description du contexte du système peut commencer. Elle consiste en trois activités successives :

- l'identification des acteurs,
- l'identification des messages,
- la réalisation des diagrammes de contexte. [2]

1.2.1 Identifications des acteurs

Un acteur représente l'abstraction d'un rôle joué par des entités externes (utilisateur, dispositifs matériels ou autres système) qui interagit directement avec le système étudié. [2]

On peut distinguer deux types des acteurs :

Acteur principal

L'acteur principal celui pour qui le cas d'utilisation produit un résultat observable.

Acteur secondaire

les acteurs secondaires sont souvent sollicités pour des informations complémentaires, ils peuvent uniquement consulter ou informer le système lors de l'exécution du

cas d'utilisation. [3]

Les différents acteurs qui doivent interagir dans notre système sont :

– **Acteurs principaux**

Ce sont les utilisateurs du système quelque soit (Enseignant, Agent de MAJ et Étudiant)

Enseignant :

c'est la personne dans laquelle il peut préparer, paramétrer et superviser le passage de l'examen (le type, la durée, ..., possibilité de superviser les écrans des étudiants).

Agent de MAJ de style de question :

c'est la personne dans laquelle il est responsable de développer les différents styles des question (QCM, Calcule, Image...), il permet d'éditer, consulter, modifier et supprimer ses styles.

Étudiant :

c'est la personne dans laquelle il doit passer l'épreuve dans un écran qui ne permet aucune autre action, et avoir la note juste après l'examen.

– **Acteurs secondaires**

Imprimante :

L'imprimante est utilisée par notre système, elle n'utilise pas le système et n'est pas l'initiatrice d'action de notre système, elle permet de consulter les données et les imprimer.

Plateforme E-Learning :

dans notre système elle désigne la plateforme du centre université de Mila. elle permet de faciliter d'une part l'accès à des ressources et à des services, d'autre part les échanges et la collaboration à distance.

De plus de ces deux acteurs secondaires il existe un acteur **fichier** pour l'importation de la liste des étudiants et un autre **application** pour le partage des notes à n'importe quelle application du téléphone (google, gmail ...).

1.2.2 Identifications des messages

Un message représente la spécification d'une communication unidirectionnelle entre objets qui transporte de l'information avec l'intention de déclencher une activité chez le récepteur.

Cette notion est utilisée également pour décrire les interfaces de plus haut niveau entre les acteurs et le système. [2]

- **Les messages émis par le système**

- Information des groupes enseignés avec les informations des étudiants.
- Information des tests et corrigés type.
- Information des sessions.
- Information des notes.
- Information de l'archive des tests.
- Information des styles de questions.
- Affichage des formulaires.
- Les notifications.

- **Les messages reçus par le système**

- Demande d'inscription.
- Demande d'authentification.
- Demande de modifier profil.
- Demande de consultation, d'ajout, de modification, de suppression et d'archivage.
- Demande d'importation, d'impression et de partage.
- Lancer session.
- Passer test.
- Demande d'éditer style question.

1.2.3 Le diagramme de contexte du système

La modélisation du contexte consiste à représenter de façon synthétique tous les messages échangés entre systèmes et acteurs sur un diagramme, que l'on peut qualifier de diagramme de contexte dynamique.

Ce diagramme dynamique est présenté grâce à un diagramme de collaboration UML comme suit :

- le système étudié est représenté par un objet central.
- Cet objet central entouré par un autre objet symbolisant les différents acteurs.
- Des liens relient le système à chaque acteur.
- Sur chaque lien sont montrés les messages en entrée et en sortie du système. [2]

Le schéma suivant représente le diagramme de contexte pour notre système :

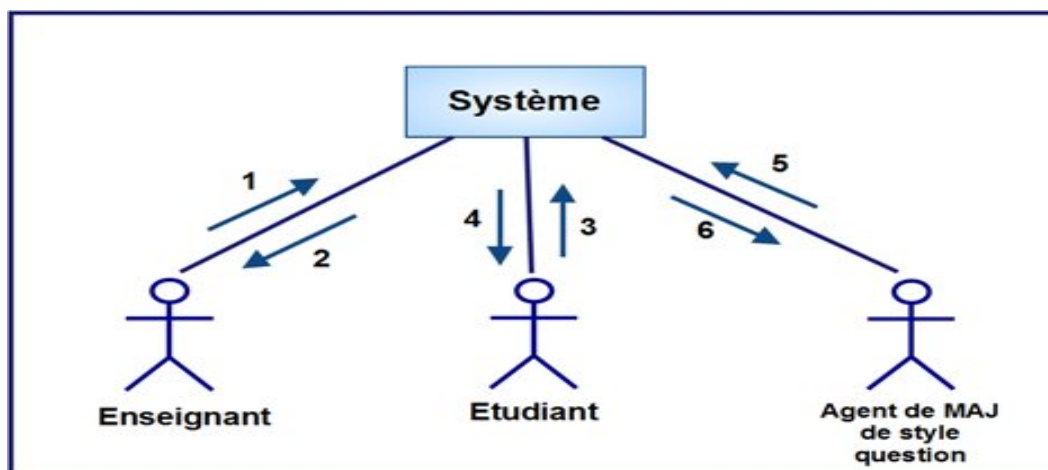


FIGURE 1.2 – Diagramme de contexte du système «Contrôle continu»

★ Légende des messages du diagramme de contexte dynamique

Messages émis	Messages reçus
2 : Système vers enseignant – Information des groupes enseignés – Informations des étudiants. – Information des tests et corrigés type. – Information des sessions. – Information des notes. – Information de l'archive des tests. – Affichage des formulaires. – Les notifications.	1 : Enseignant vers système – Demande d'inscription. – Demande d'authentification. – Demande de modifier profil. – Demande d'ajout, de modification. – Demande de suppression. – Demande de consultation, d'archivage. – Demande d'importation, d'impression. – Demande de partage. – Lancer session.
4 : Système vers étudiant – Afficher formulaire. – Afficher test. – Afficher résultat. – Les notifications.	3 : Étudiant vers système – Demande d'inscription. – Demande d'authentification. – Demande de modifier profil. – Demande de consultation – Passer test.
6 : Système vers agent de MAJ – Information des styles de questions	5 : Agent de MAJ vers système – Demande d'inscription. – Demande d'authentification. – Demande de modifier profil. – Demande d'éditer style question.

TABLE 1.1: Légende des messages de diagramme de contexte

Conclusion

Après avoir dégagé les besoins fonctionnels et opérationnels et tous les critères qu'on doit prendre en considération, l'étude préliminaire prépare les étapes plus formelles de capture des besoins fonctionnels et techniques que nous allons décrire au chapitre suivant.

Capture des besoins

Introduction

Après qu'on a défini les besoins fonctionnels nécessaires pour le fonctionnement de notre application, dans ce chapitre on présente ses besoins sous forme des cas d'utilisation par un diagramme de cas d'utilisation, ce qui nous donne une modélisation du système par cas d'utilisation. Ainsi, on va donner une fiche descriptive de chaque cas avec la description graphique correspondante.

2.1 Situation de l'étape dans le processus 2TUP

La deuxième étape du processus suivie c'est **Capture des besoins**, la figure suivante représente la situation de cette phase dans le processus 2TUP, elle se passe par deux branches d'une part une branche gauche présente capture des besoins fonctionnels et de l'autre par une branche à droite présente capture des besoins techniques :

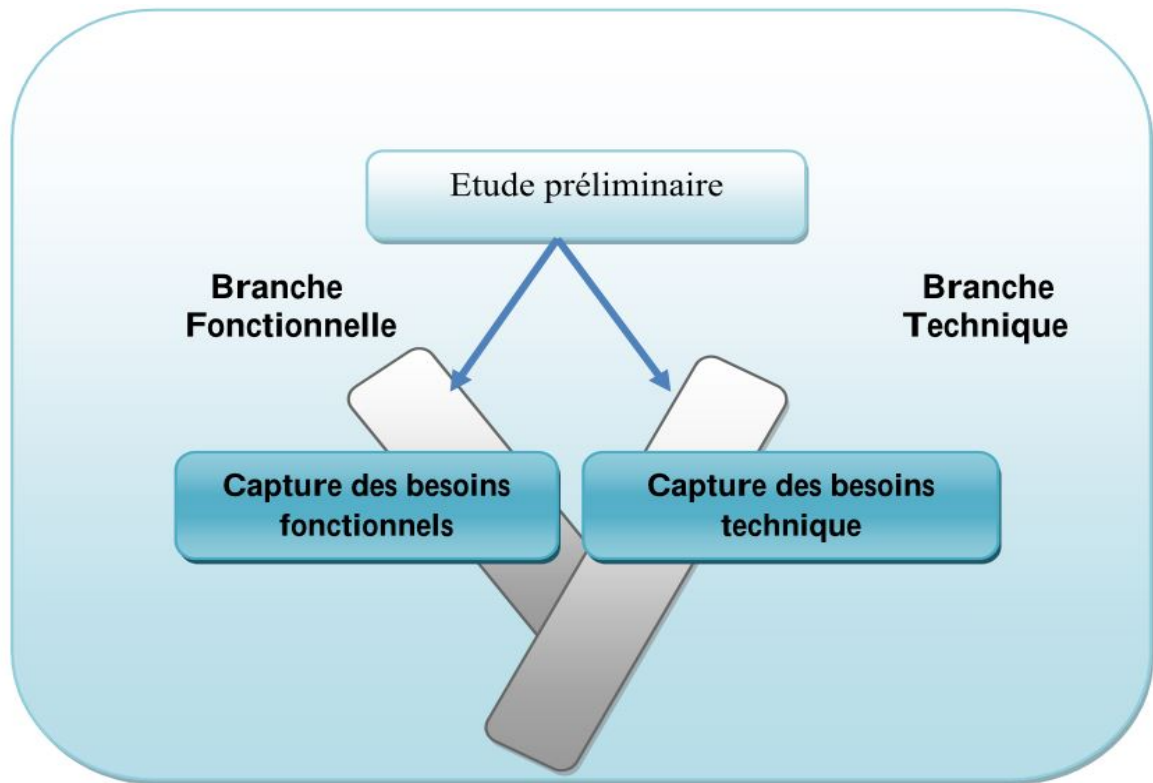


FIGURE 2.1 – Situation du capture des besoins dans 2TUP

2.2 Capture des besoins fonctionnels

La capture des besoins fonctionnels est la première étape de la branche gauche du cycle en Y de la méthode 2TUP. Elle formalise et détaille ce qui a été recueilli au cours de l'étude préliminaire. Cette phase représente un point de vue « fonctionnel » de l'architecture système. Elle est complétée au niveau de la branche droite du Y par la capture des besoins techniques et prépare l'étape suivante de la branche gauche : l'analyse. [2]

Dans cette phase Nous allons permet de :

- identifier les cas d'utilisation du système par ces acteurs.
- décrire les cas d'utilisation.
- identifier les classes candidates.
- élaborer les diagrammes des classes participantes des cas d'utilisation.

2.2.1 Identification des cas d'utilisation

Un cas d'utilisation (use Case) représente un ensemble de séquences d'actions réalisées par le système et produisant un résultat observable intéressant pour un acteur particulier.

Un cas d'utilisation modélise un service rendu par le système. Il exprime les interactions acteurs/système et apporte une valeur ajoutée « notable » à l'acteur concerné. [2]

En effet, ils sont des représentations fonctionnelles du système, ils permettent de modéliser les attentes des utilisateurs afin de réaliser une bonne délimitation du système et également d'améliorer la compréhension de son fonctionnement. La liste des cas d'utilisation de notre système est représentée dans le tableau suivant :

Num	Cas d'utilisation	Acteurs
1	S'inscrire	Utilisateur
2	S'authentifier	Utilisateur
3	Modifier profil	Utilisateur
4	Gérer la liste des groupes enseignés - consulter groupe - Ajouter groupe - Modifier groupe - Supprimer groupe	Enseignant
5	Imprimer	Enseignant
6	Importer la liste des étudiants	Enseignant
7	Gérer étudiants - consulter étudiant - Ajouter étudiant - Modifier étudiant - Supprimer étudiant	Enseignant
8	Gérer tests et corrigés type - Consulter test et corrigé type - Ajouter test et corrigé type - Modifier test et corrigé type - Supprimer test et corrigé type -Archiver test et corrigé type	Enseignant
9	Gérer sessions - Créer session - Consulter session - Lancer session - Modifier session - Supprimer session	Enseignant
10	Consulter notes	Enseignant
11	Partager notes	Enseignant
12	Gérer archive des tests - Consulter test de l'archive - Restaurer(désarchiver) test de l'archive	Enseignant
13	Gérer éditeur de style question - Consulter éditeur - Ajouter éditeur - Modifier éditeur - Supprimer éditeur	Agent de MAJ de style question
14	Passer test	Etudiant
15	Consulter son test	Etudiant
16	Consulter archive des tests	Etudiant

TABLE 2.1: Liste des cas d'utilisations fonctionnels

Diagramme de cas d'utilisation

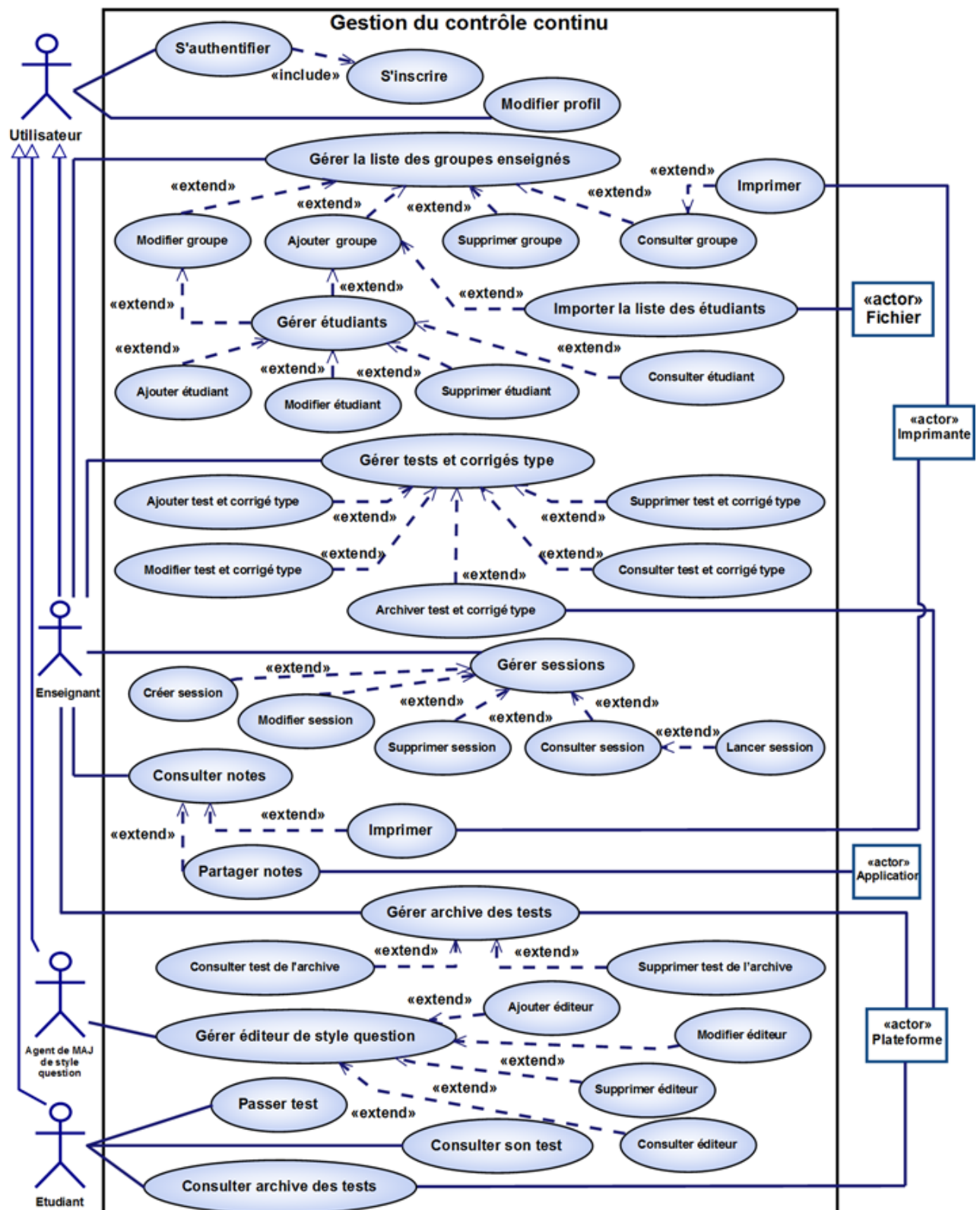


FIGURE 2.2 – Diagramme de cas d'utilisation générale

2.2.2 Description des cas d'utilisation

Nous décrivons chaque cas de façon textuelle en lui associant un nom, un objectif, les acteurs qui y participent, les préconditions et des scénarios. Puis, chaque scénario est illustré par un diagramme de séquence.

- « S'inscrire »

- Description textuelle

Cas d'utilisation	S'inscrire.
Acteurs	Utilisateur.
But	Ce cas permet à l'utilisateur de s'inscrire.
Poste condition	L'utilisateur est inscrit.
Scénario nominal	1-L'utilisateur demande d'inscrire dans l'application. 2-Le système affiche le formulaire d'inscription. 3-L'utilisateur remplit les champs du formulaire et confirme l'inscription. 4-Le système vérifie la validité des informations saisies et les enregistrées. 5- Le système affiche un message de confirmation d'inscription «Inscription avec succès ».
Scénario alternatif	• Les informations saisies sont invalide.le système affiche un message d'erreur. On reprend scénario nominal au point 2. • L'utilisateur existe déjà dans l'application.le système affiche un message d'erreur. On reprend scénario nominal au point 2.
Scénario optionnelle	• L'utilisateur peut annuler l'opération en cours à n'importe quel moment pourvue qu'il ne l'a pas validé.

TABLE 2.2: Description textuelle du cas d'utilisation

« S'inscrire »

– Diagramme de séquence

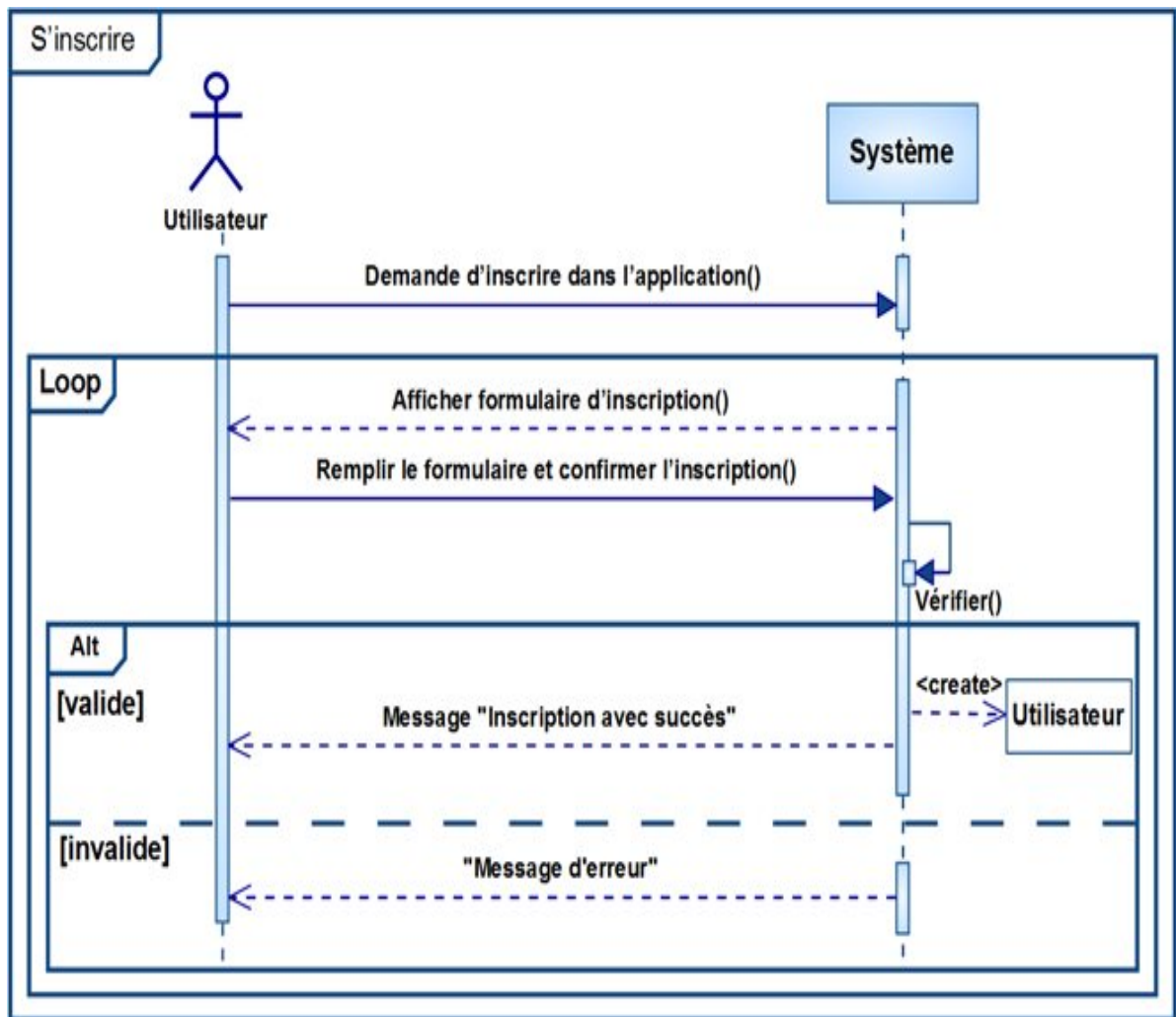


FIGURE 2.3 – Diagramme de séquence du cas S'inscrire

- « Ajouter groupe »

- Description textuelle

Cas d'utilisation	Ajouter groupe.
Acteur	Enseignant.
But	Ce cas permet à l'enseignant d'ajouter un nouveau groupe.
Pré condition	L'enseignant est dans son espace et il a choisi l'action « Ajouter groupe ».
Poste condition	Le nouveau groupe est ajouté à la liste des groupes enseignés.
Scénario nominal	1-Le système affiche l'interface d'ajout. 2-L'enseignant remplit les informations du groupe et confirme . l'opération. 3-Le système vérifie la validité des informations saisies et les enregistre. 4-Le système affiche un message de confirmation d'ajout «Ajout avec succès »
Scénario alternatif	• Les informations saisies sont invalide.le système affiche un message d'erreur On reprend scénario nominal au point 1.
Scénario optionnelle	• L'enseignant peut annuler l'opération en cours à n'importe quel moment pourvue qu'il ne l'a pas validé.

TABLE 2.3: Description textuelle du cas d'utilisation
« Ajouter groupe »

– Diagramme de séquence

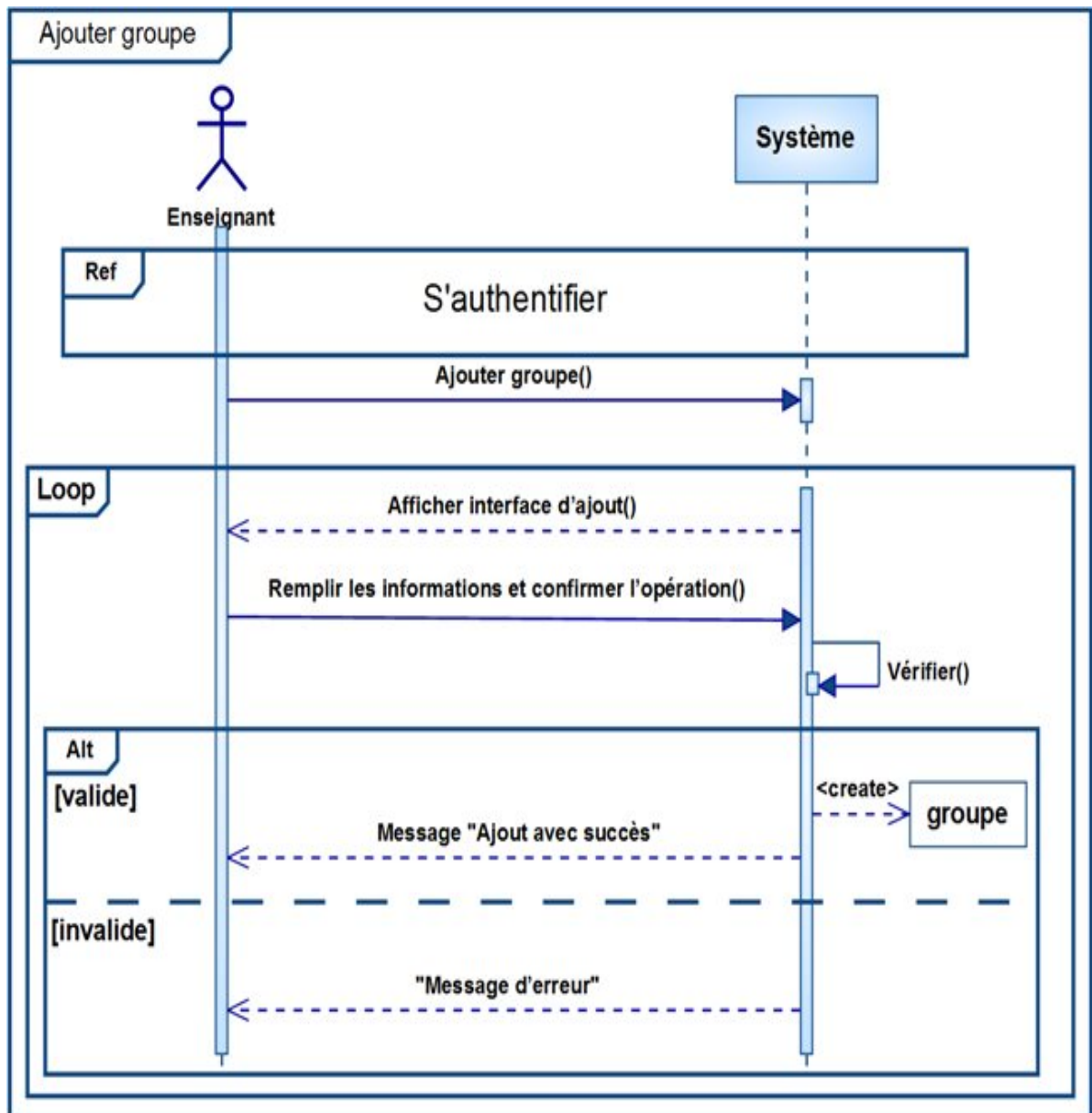


FIGURE 2.4 – Diagramme de séquence du cas Ajouter groupe

- « Ajouter étudiant »

- Description textuelle

Cas d'utilisation	Ajouter étudiant.
Acteur	Enseignant.
But	Ce cas permet à l'enseignant d'ajouter un nouveau étudiant.
Pré condition	L'enseignant est dans son espace et il a choisi l'action « ajouter étudiant ».
Poste condition	Un étudiant est ajouté et affecté à un groupe des groupes enseigné par lui.
Scénario nominal	1-Le système affiche un formulaire d'ajout. 2-L'enseignant rempli les champs du formulaire et confirme l'ajout. 3-Le système vérifie la validité des informations saisies et les enregistre. 4-Le système affiche un message de confirmation d'ajout «Ajout avec succès ».
Scénario alternatif	• Les informations saisies sont invalide.le système affiche un message d'erreur.On reprend scénario nominal au point 1. • L'étudiant existe déjà dans le groupe. le système affiche un message d'erreur. On reprend scénario nominal au point 1.
Scénario optionnelle	• L'enseignant peut annuler l'opération en cours à n'importe quel moment pourvue qu'il ne l'a pas validé.

TABLE 2.4: Description textuelle du cas d'utilisation
« Ajouter étudiant ».

– Diagramme de séquence

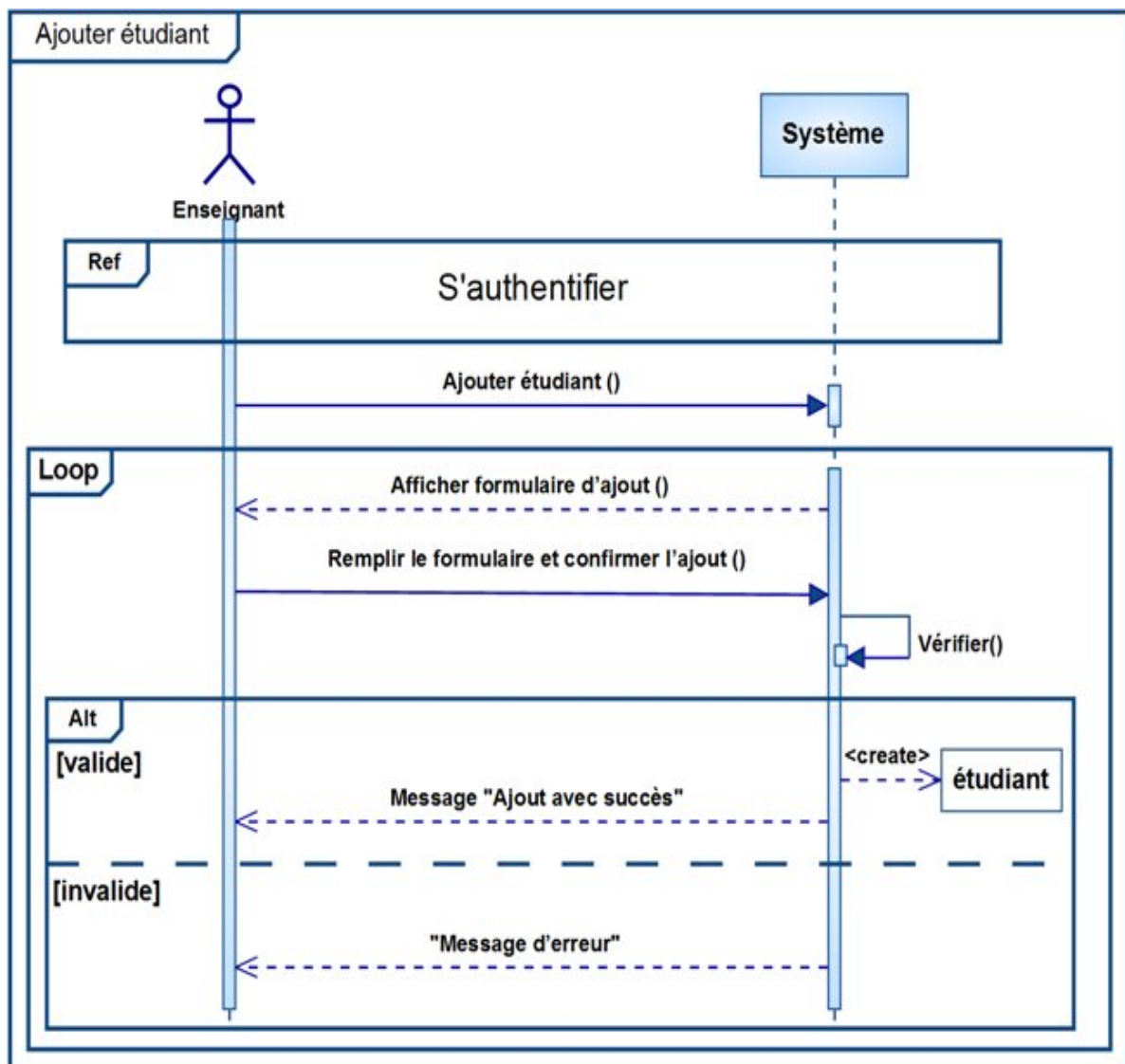


FIGURE 2.5 – Diagramme de séquence du cas Ajouter étudiant

- « Consulter test et corrigé type »
 - Description textuelle

Cas d'utilisation	Consulter test et corrigé type.
Acteur	Enseignant.
But	Ce cas permet à l'enseignant de consulter un test et son corrigé type.
Pré condition	L'enseignant est dans son espace et il a choisi l'action « Consulter test et corrigé type ».
Poste condition	Le test choisi avec son corrigé type est affiché.
Scénario nominal	1-Le système affiche la liste des tests. 2-L'enseignant choisit un des tests affichés. 3-Le système affiche le test choisi.

TABLE 2.5: Description textuelle du cas d'utilisation
« Consulter test et corrigé type »

- Diagramme de séquence

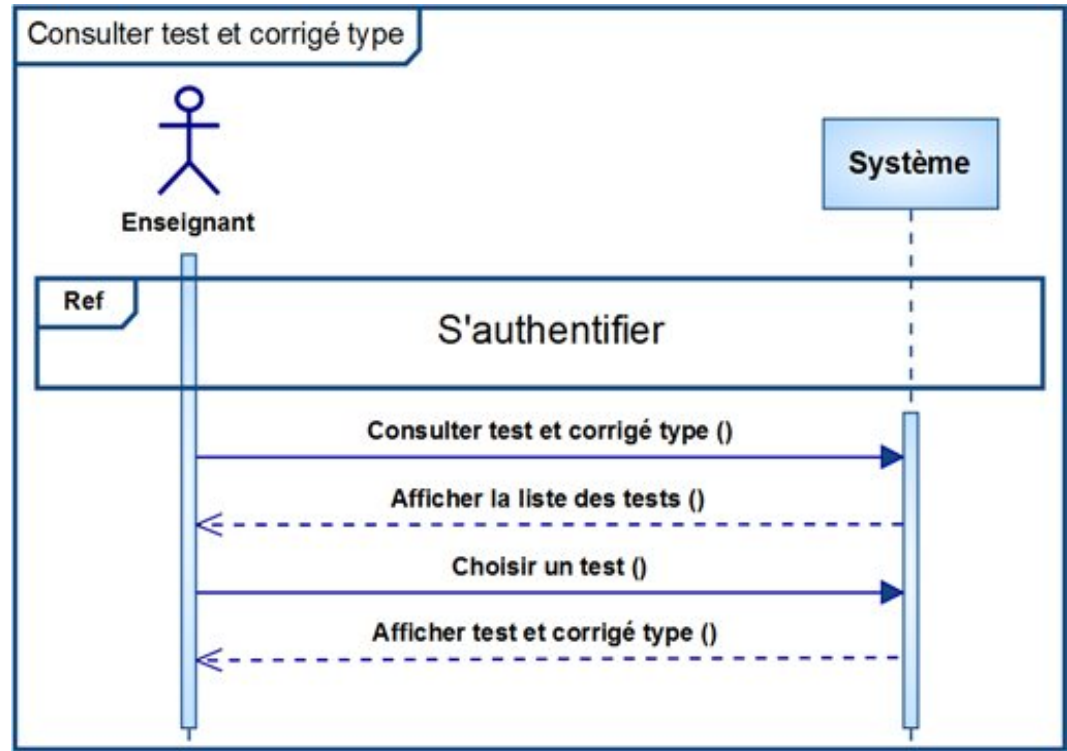


FIGURE 2.6 – Diagramme de séquence du cas Consulter test et corrigé type

- « Ajouter test et corrigé type »

- Description textuelle

Cas d'utilisation	Ajouter test et corrigé type.
Acteur	Enseignant.
But	Ce cas permet à l'enseignant d'ajouter un nouveau test.
Pré condition	L'enseignant est dans son espace et il a choisi l'action « ajouter test et corrigé type ».
Poste condition	Le nouveau test est enregistré.
Scénario nominal	1-Le système affiche un formulaire d'ajout. 2-L'enseignant saisit les informations du test et confirme l'ajout. 3-Le système vérifie la validité des informations saisies et les enregistre. 4-Le système affiche un message de confirmation d'ajout « Ajout test avec succès ». 5-L'enseignant demande l'ajout d'une question en choisissant le style(QCM, calcule, graphe, texte, ...). 6-Le système affiche l'interface de l'édition de la question selon le style choisi. 7-L'enseignant édite et valide la question et le corrigé type. 8-Le système vérifie la validité des informations saisies et les enregistre. 9-Le système affiche message de confirmation d'ajout « Ajout question avec succès ».
Scénario alternatif	• L'enseignant demande l'ajout d'une question avant de valider l'ajout d'un test. le système affiche un message d'information«l'ajout d'un test est obligatoire». On reprend scénario nominal au point 1. •L'enseignant demande l'ajout d'une nouvelle question, reprend le point 5. • Les informations saisies sont invalide.le système affiche un message d'erreur. On reprend scénario nominal au point 1 ou au point 6.
Scénario optionnelle	• L'enseignant peut annuler l'opération en cours à n'importe quel moment pourvue qu'il ne l'a pas validé.

TABLE 2.6: Description textuelle du cas d'utilisation
« Ajouter test et corrigé type »

– Diagramme de séquence

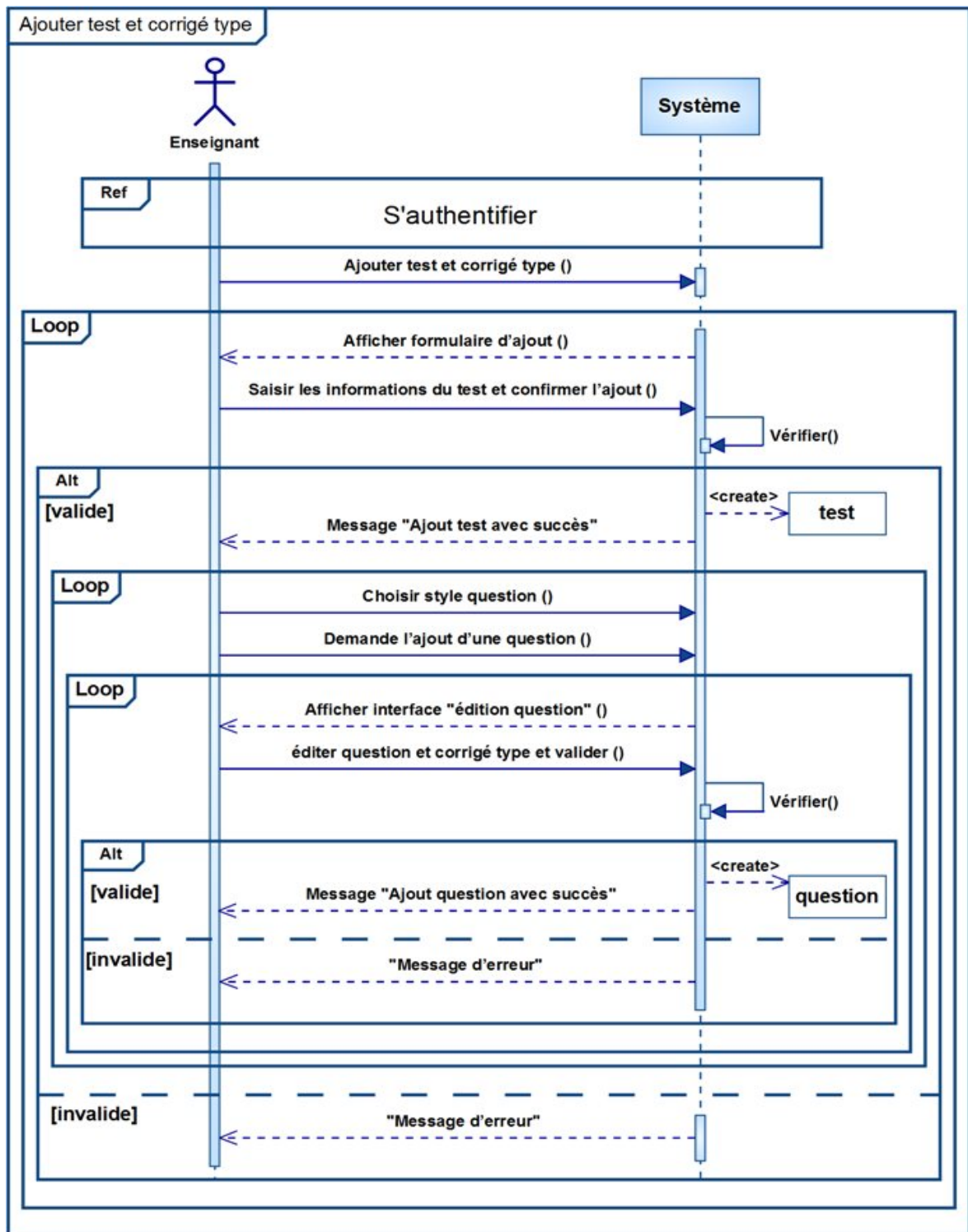


FIGURE 2.7 – Diagramme de séquence du cas Ajouter test et corrigé type

- « Créer session »

- Description textuelle

Cas d'utilisation	Créer session.
Acteur	Enseignant.
But	Ce cas permet à l'enseignant de créer une nouvelle session.
Pré condition	L'enseignant est dans son espace et il a choisi l'action « Créer session ».
Poste condition	La nouvelle session est créée.
Scénario nominal	1-Le système affiche l'interface de création. 2-L'enseignant choisi un test et les groupes dont ils veulent passer ce test. 3-L'enseignant remplit le nom et le temps de la session et confirme la création. 4-Le système vérifie la validité des informations et les enregistre. 4-Le système affiche un message de confirmation de création «Création avec succès ».
Scénario alternatif	• Les informations saisies sont invalide.le système affiche un message d'erreur. On reprend scénario nominal au point 1.
Scénario optionnelle	• L'enseignant peut annuler l'opération en cours à n'importe quel moment pourvue qu'il ne l'a pas validé.

TABLE 2.7: Description textuelle du cas d'utilisation
« Créer session »

– Diagramme de séquence

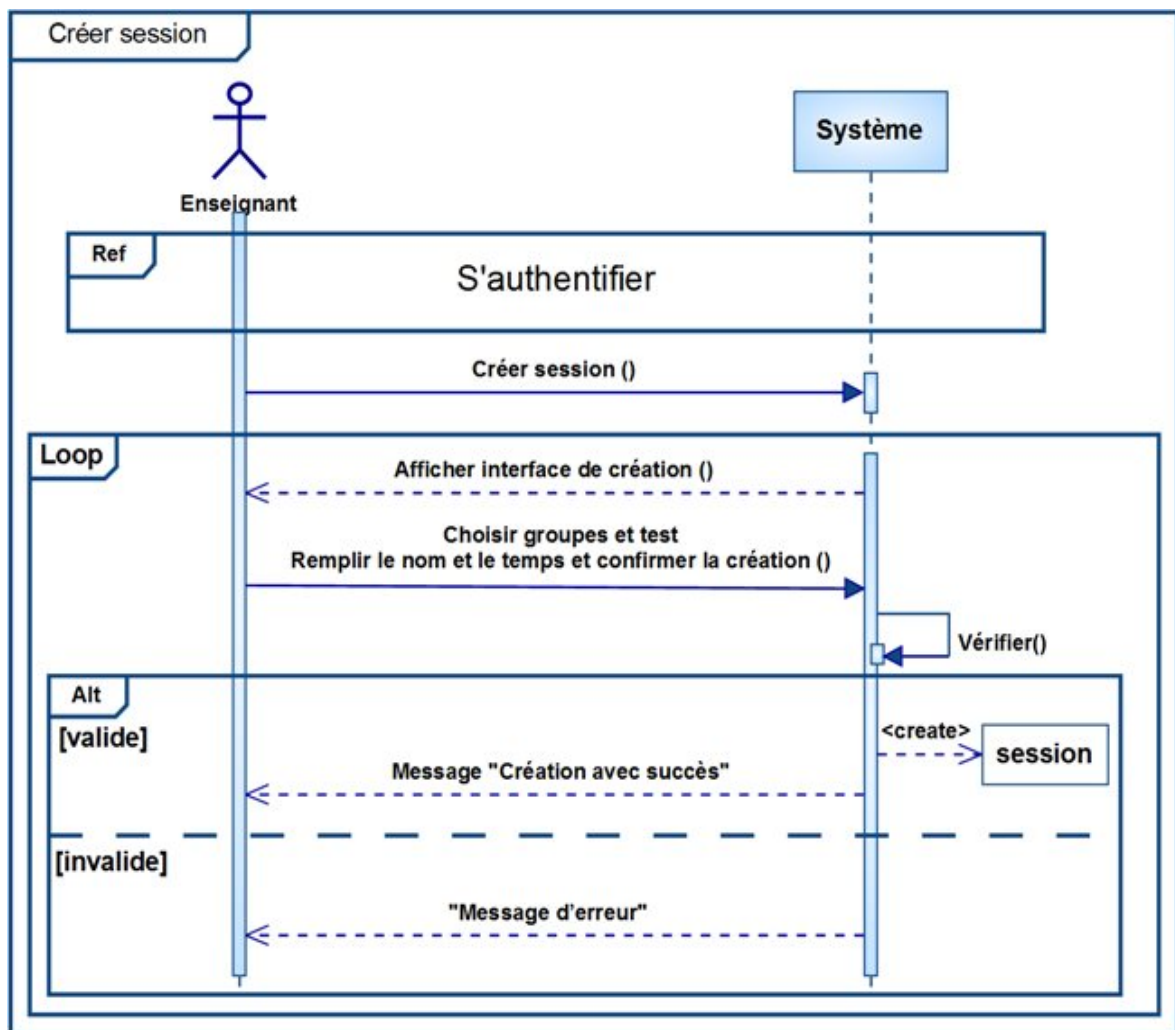


FIGURE 2.8 – Diagramme de séquence du cas Créer session

- « Lancer session »

- Description textuelle

Cas d'utilisation	lancer session.
Acteur	Enseignant.
But	Ce cas permet à l'enseignant de lancer une session.
Pré condition	L'enseignant est dans son espace et il a choisi l'action « Lancer session ».
Poste condition	Le test est lancé aux étudiants.
Scénario nominal	1-L'enseignant consulte la liste des sessions et choisit la session dont elle veut passer le test. 2-L'enseignant lance l'opération. 3-Le système affiche un message de confirmation de lancement. 4-L'enseignant confirme le lancement. 5-Le système lance la session concernée. 6-Le système affiche une notification « lancement avec succès ».
Scénario optionnelle	• L'enseignant peut annuler l'opération en cours à n'importe quel moment pourvue qu'il ne l'a pas validé.

TABLE 2.8: Description textuelle du cas d'utilisation
« Lancer session »

– Diagramme de séquence

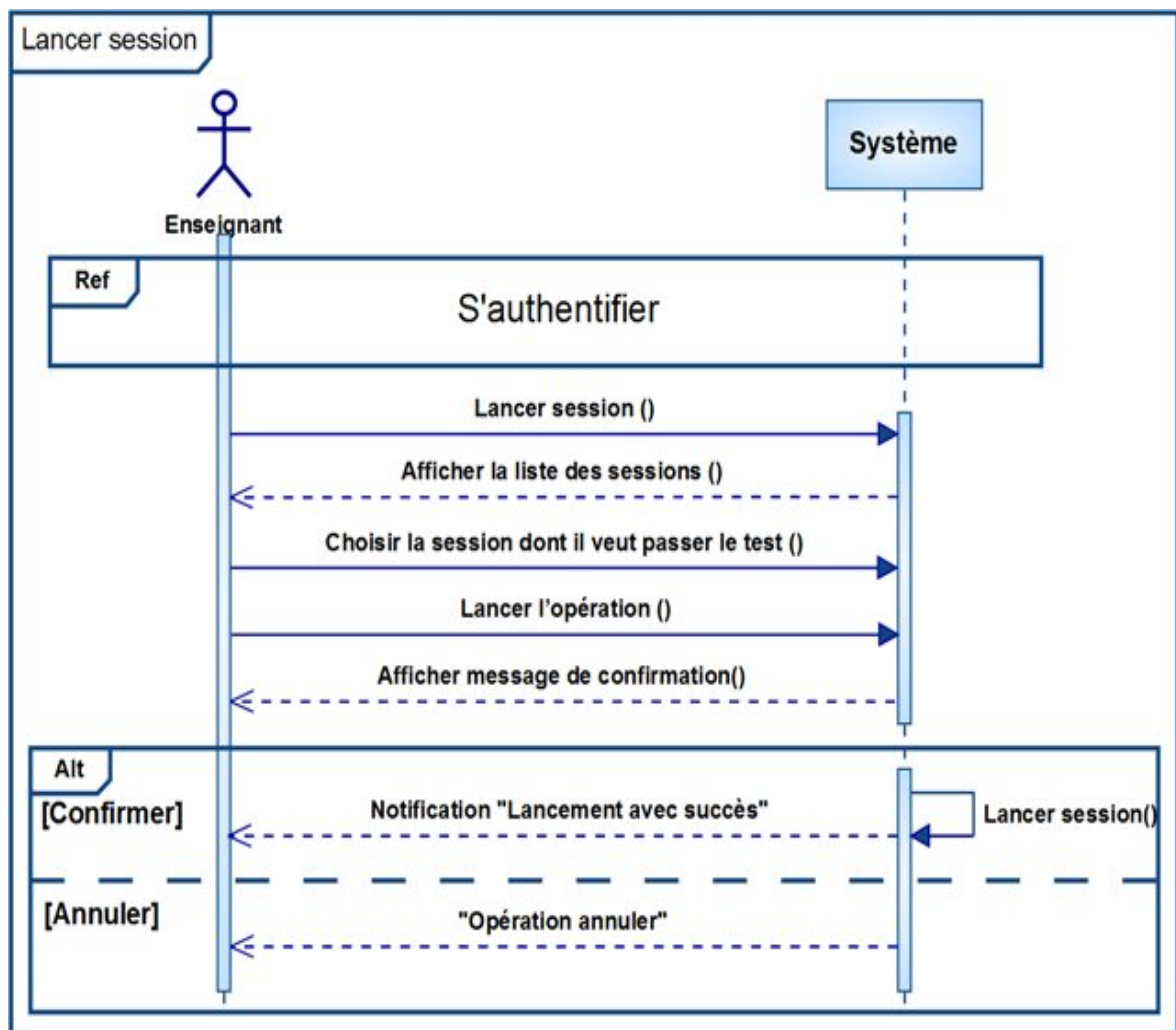


FIGURE 2.9 – Diagramme de séquence du cas Lancer session

- « Consulter notes »

- Description textuelle

Cas d'utilisation	Consulter notes.
Acteur	Enseignant.
But	Ce cas permet à l'enseignant de consulter les notes des étudiants.
Pré condition	L'enseignant doit être authentifié, il est dans son espace.
Poste condition	Les notes du groupe sélectionné sont affichées.
Scénario nominal	1-Le système affiche la liste des groupes enseignés. 2-L'enseignant choisit un groupe. 3-Le système affiche les notes de ce groupe.

TABLE 2.9: Description textuelle du cas d'utilisation
« Consulter notes »

- Diagramme de séquence

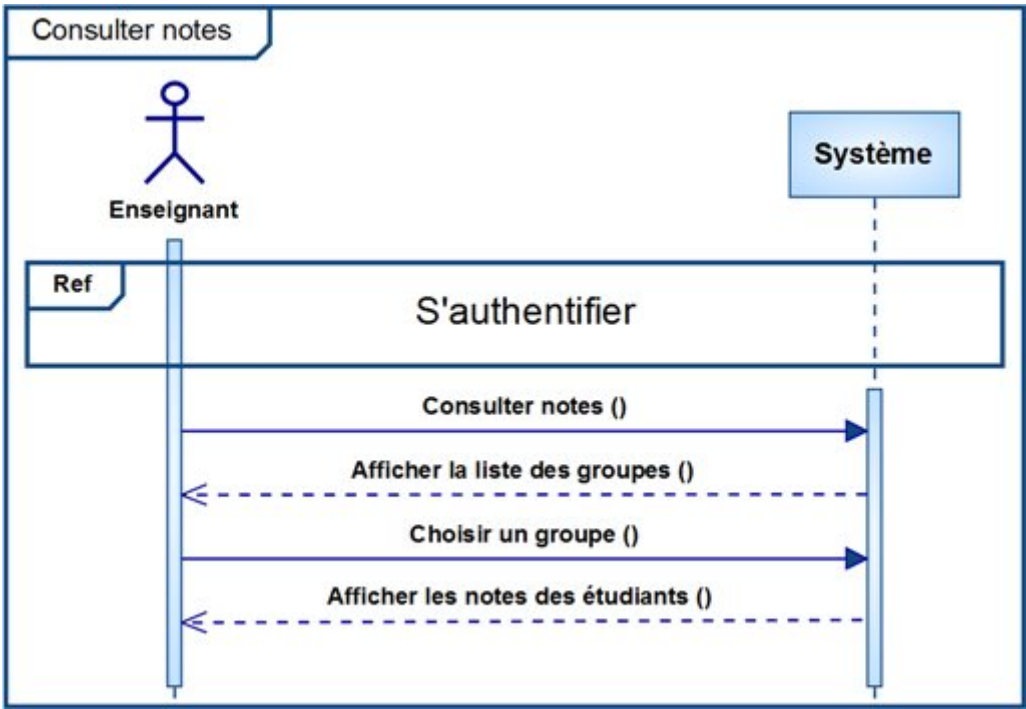


FIGURE 2.10 – Diagramme de séquence du cas Consulter notes

- « Partager notes »
 - Description textuelle

Cas d'utilisation	Partager notes .
Acteur	Enseignant (principal),application (secondaire).
But	Ce cas permet à l'enseignant de partager les notes.
Pré condition	L'enseignant est dans son espace,il a choisi l'action «partager notes».
Poste condition	Les notes sont partagées.
Scénario nominal	1-Le système affiche l'interface de partage. 2-L'enseignant choisit l'application de partage et confirme l'opération. 3-Le système affiche une notification «Partage avec succès ».
Scénario optionnelle	• L'enseignant peut annuler l'opération en cours à n'importe quel moment pourvue qu'il ne l'a pas validé.

TABLE 2.10: Description textuelle du cas d'utilisation
« Partager notes »

- Diagramme de séquence

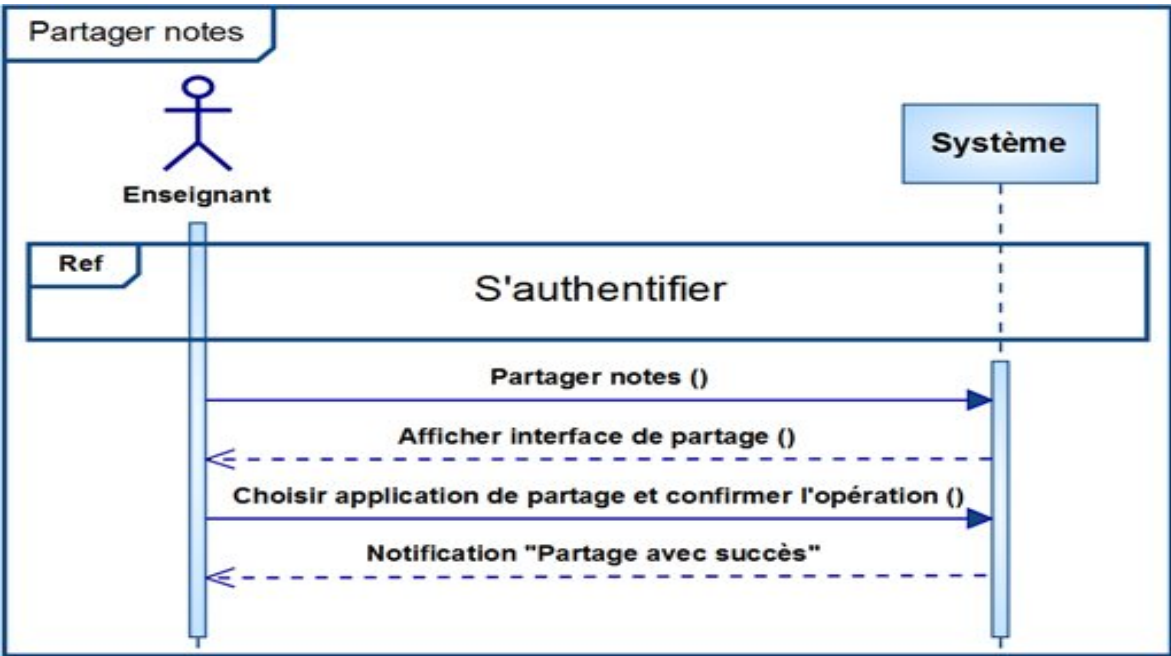


FIGURE 2.11 – Diagramme de séquence du cas Partager notes

- « Ajouter éditeur »

- Description textuelle

Cas d'utilisation	Ajouter éditeur.
Acteur	Agent de MAJ de style question.
But	Ce cas permet à L'agent d'ajouter un nouveau éditeur de style question.
Pré condition	L'agent est dans son espace et il a choisi l'action « ajouter éditeur ».
Poste condition	Le nouveau éditeur est ajouté à la liste des éditeurs.
Scénario nominal	1-Le système affiche l'interface d'ajout. 2-L'agent remplit le nom et édite le format d'éditeur de style question puis confirme l'opération. 3-Le système vérifie la validité des informations saisies et les enregistre. 4-Le système affiche un message de confirmation d'ajout « Ajout avec succès ».
Scénario alternatif	• Les informations saisies sont invalide.le système affiche un message d'erreur. On reprend scénario nominal au point 1.
Scénario optionnelle	• L'agent peut annuler l'opération en cours à n'importe quel moment pourvue qu'il ne l'a pas validé.

TABLE 2.11: Description textuelle du cas d'utilisation

« Ajouter éditeur »

– Diagramme de séquence

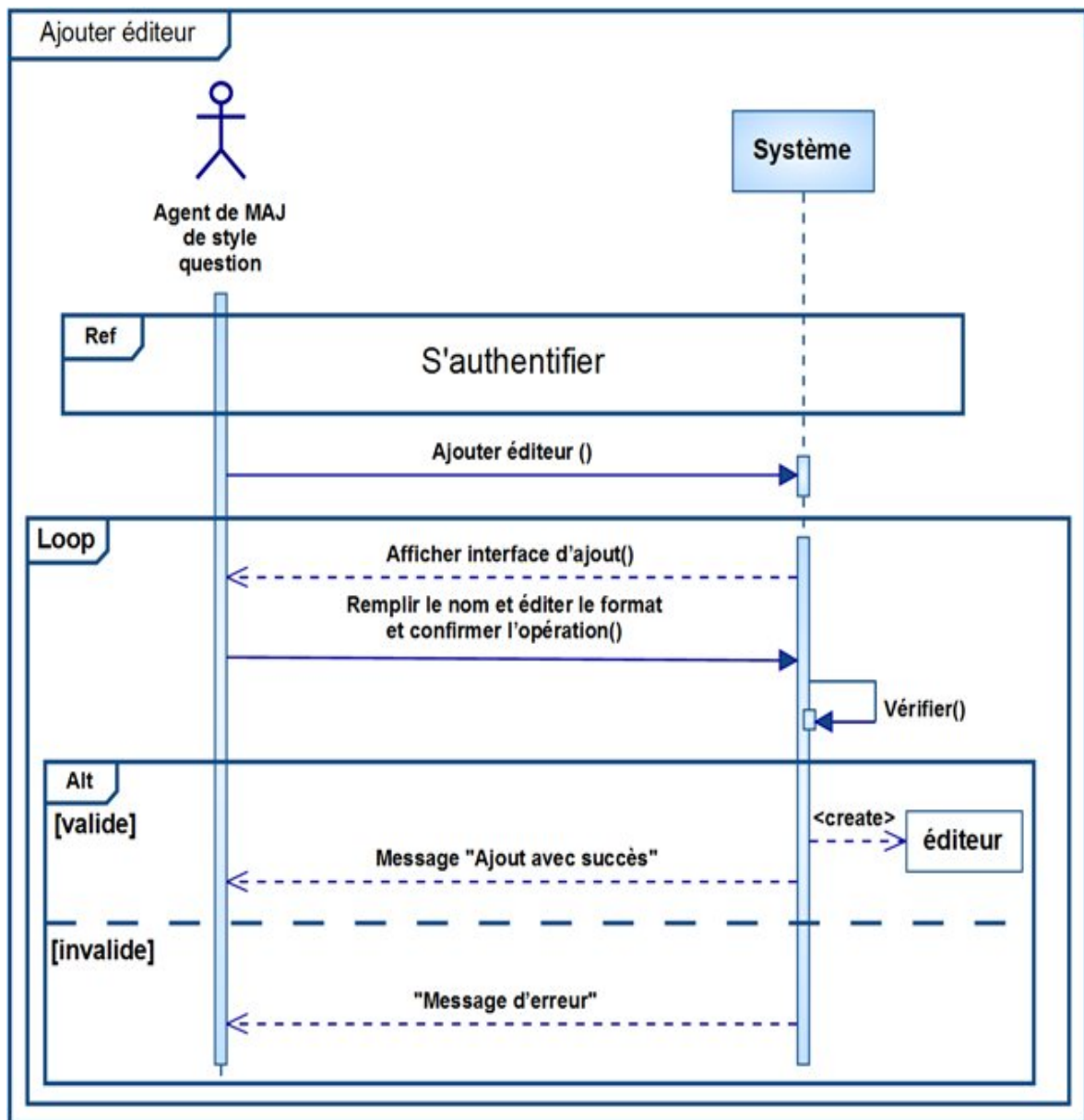


FIGURE 2.12 – Diagramme de séquence du cas Ajouter éditeur

- « Modifier éditeur »

- Description textuelle

Cas d'utilisation	Modifier éditeur.
Acteur	Agent de MAJ de style question.
But	Ce cas permet à l'agent de modifier le nom et le format d'éditeur de style question .
Pré condition	l'agent est dans son espace,et il a choisi l'action « Modifier éditeur ».
Poste condition	Le nom et le format d'éditeur sont modifiés.
Scénario nominal	1-l'agent consulte la liste des éditeurs et choisit l'éditeur à modifier. 2-Le système affiche l'interface de modification. 3-l'agent modifie les informations souhaitées et confirme. 4-Le système vérifie la validité des informations et enregistre les nouvelles informations. 5-Le système affiche un message de confirmation de modification « Modification avec succès ».
Scénario alternatif	• Les informations saisies sont invalide.le système affiche un message d'erreur. On reprend scénario nominal au point 2.
Scénario optionnelle	• L'agent peut annuler l'opération en cours à n'importe quel moment pourvue qu'il ne l'a pas validé.

TABLE 2.12: Description textuelle du cas d'utilisation
« Modifier éditeur »

– Diagramme de séquence

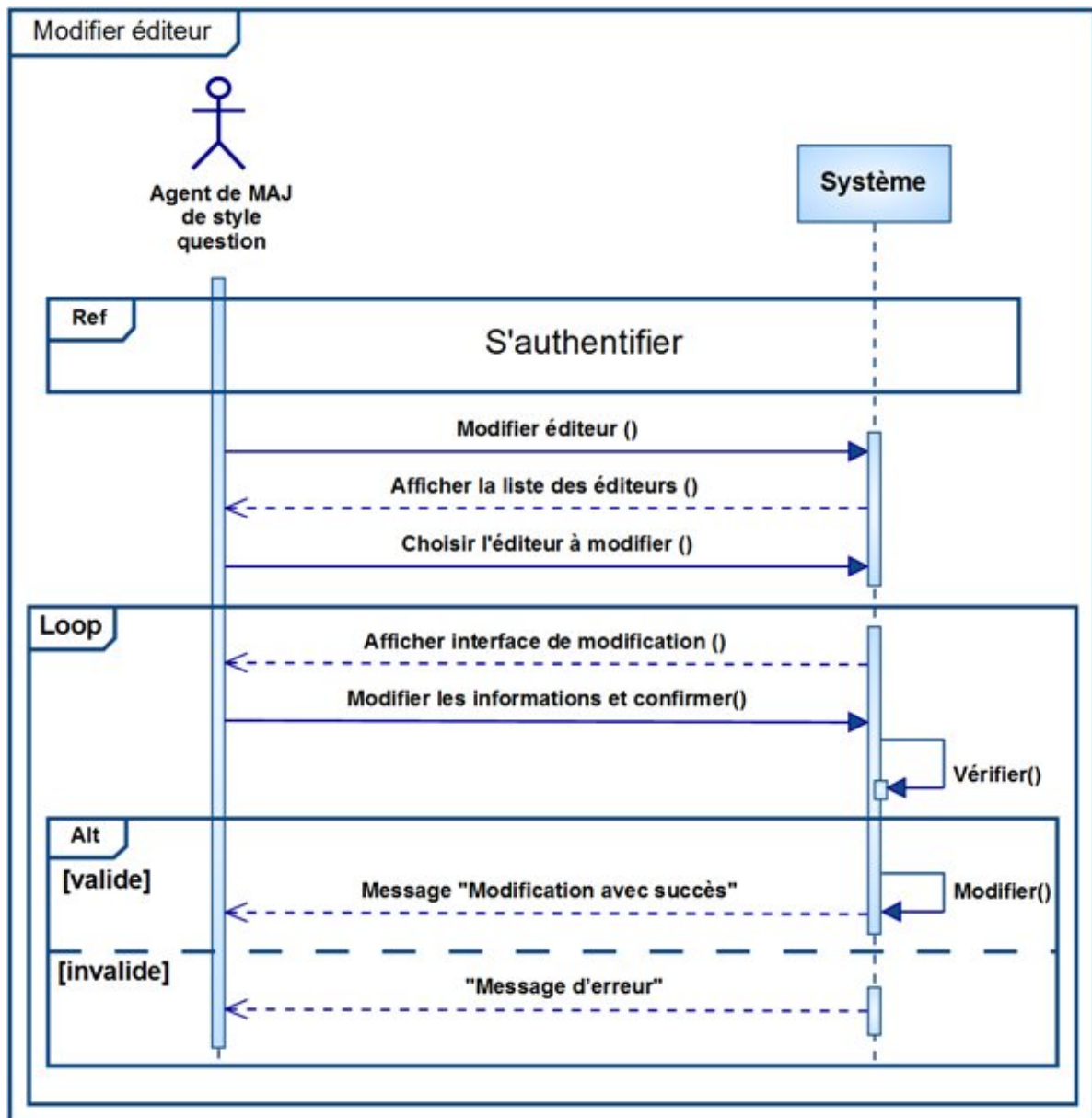


FIGURE 2.13 – Diagramme de séquence du cas Modifier éditeur

- « Passer test »

- Description textuelle

Cas d'utilisation	Passer test.
Acteur	Etudiant.
But	ce cas permet à l'étudiant de passer le test.
Pré condition	L'étudiant doit être authentifié, il est dans son espace.
Poste condition	Le test est passé.
Scénario nominal	1-L'étudiant consulte la liste des sessions et choisit une session active. 2-Le système affiche l'interface de passage du test. 3-L'étudiant répond aux questions puis valide ses réponses. 4-Le système affiche un message de confirmation « êtes-vous sûr de valider vos réponses ? ». 5-L'étudiant confirme la validation. 6-Le système enregistre le test. 7-Le système affiche une notification « Validation avec succès ».
Scénario alternatif	• Une fois la session est terminée le système effectuer une validation automatique des réponses et affiche un message d'information « le système effectuer une validation automatique ».
Scénario optionnelle	• Si l'étudiant essaye d'entrer à une session désactivé, le système affiche un message d'information « la session est désactivée ». • L'étudiant peut annuler l'opération lorsqu'il reçoit un message de confirmation on reprend le scénario nominal au point 3.

TABLE 2.13: Description textuelle du cas d'utilisation
« Passer test »

– Diagramme de séquence

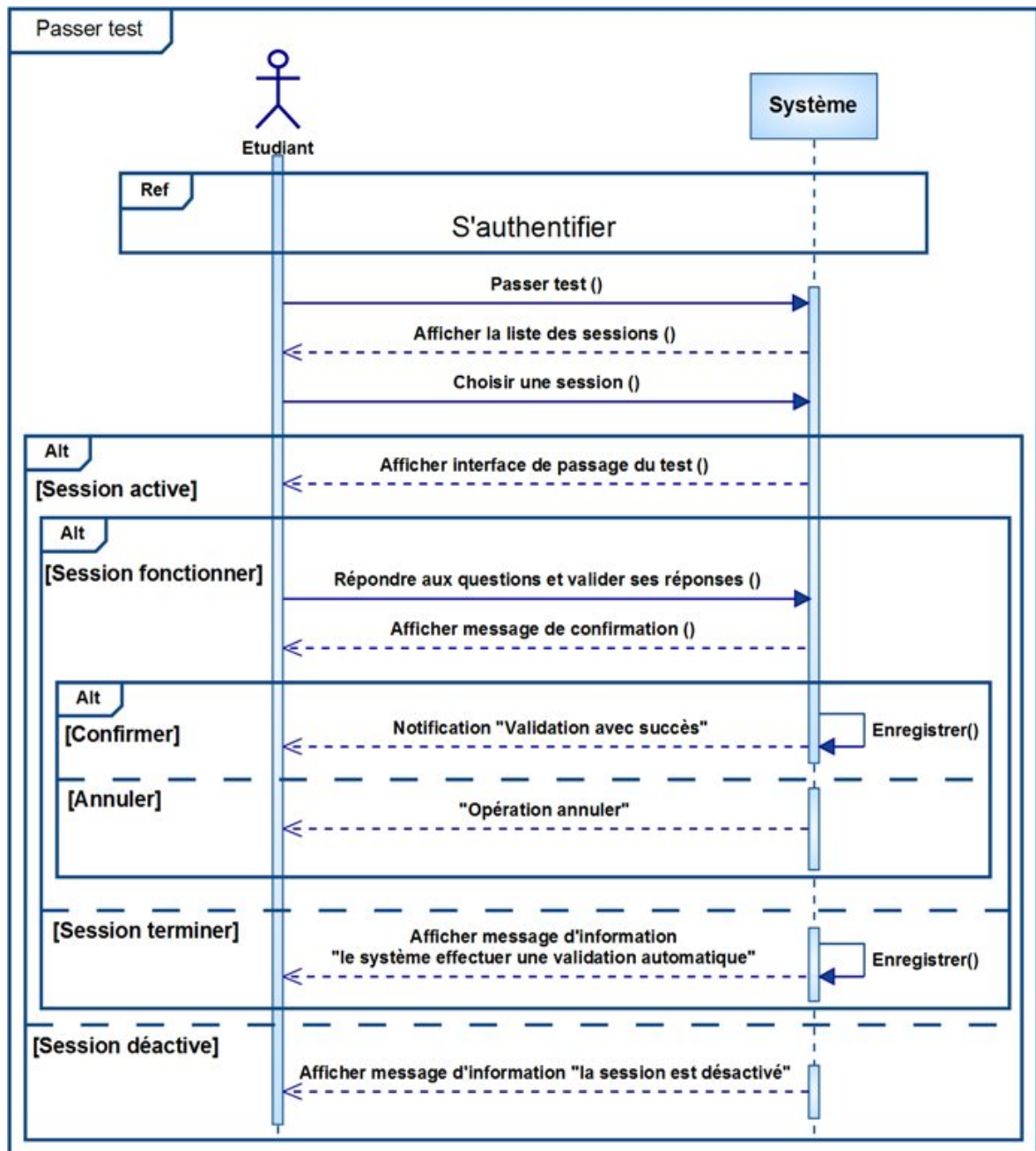


FIGURE 2.14 – Diagramme de séquence du cas Passer test

- « Consulter son test »

- Description textuelle

Cas d'utilisation	Consulter son test.
Acteur	Etudiant.
But	Ce cas permet à l'étudiant de consulter la correction de son test ainsi que la note totale.
Pré condition	L'étudiant doit être authentifié, il est dans son espace.
Poste condition	La réponse de l'étudiant et la correction sont affichées, avec l'évaluation de chaque question ainsi que note totale du test.
Scénario nominal	1-Le système affiche son test corrigé avec la note totale.
Scénario optionnelle	• Si l'étudiant demande de consulter son test avant de clôturer la session, le système affiche un message d'information « la session est encours de fonctionner ».

TABLE 2.14: Description textuelle du cas d'utilisation
« Consulter son test »

- Diagramme de séquence

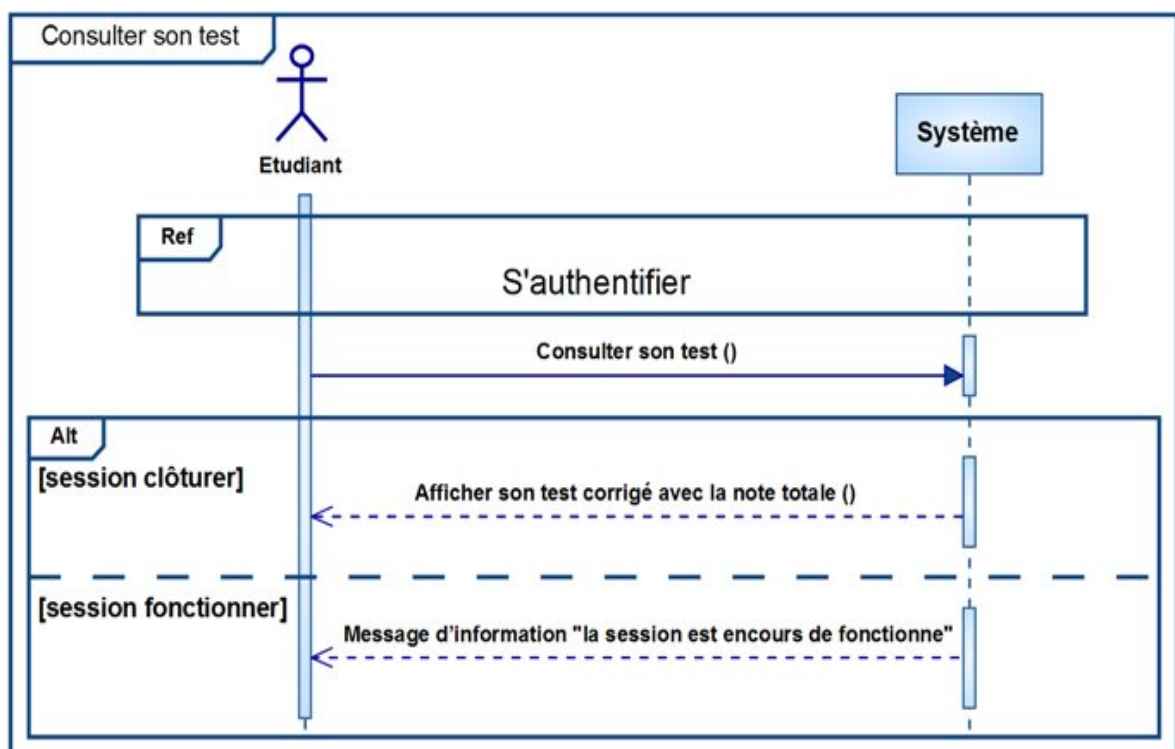


FIGURE 2.15 – Diagramme de séquence du cas Consulter son test

2.2.3 Identification des classes candidates

Cette phase va préparer la modélisation orientée objet en aidant à trouver les classes principales du futur modèle statique d'analyse. La technique utilisée pour identifier les classes candidates est la suivante :

- chercher les noms communs importants dans les descriptions textuelles des cas d'utilisation.
- vérifier les propriétés « objet » de chaque concept (identité, propriété, comportement), puis définir ses responsabilités. [2]

★ **Liste des classes candidates :**

le tableau ci-dessous résume la liste des classes candidates de notre système :

La liste des classes candidates
– Utilisateur. – Enseignant. – Étudiant. – Agent de MAJ. – Groupe. – Test. – Question. – Choix réponse. – Session. – Éditeur. – Note.

TABLE 2.15: liste des classes candidates

★ **Responsabilités des classes :**

Une responsabilité est une sorte de contrat, ou d'obligation, pour une classe. Elle se place à un niveau d'abstraction plus élevé que les attributs ou les opérations. En fait, on peut dire que les attributs, les opérations, et les associations représentent les propriétés élémentaires qui contribueront à remplir les responsabilités de la classe.[2]

Les figures suivantes représentent les responsabilités de chaque classe :

- **Classe Utilisateur**

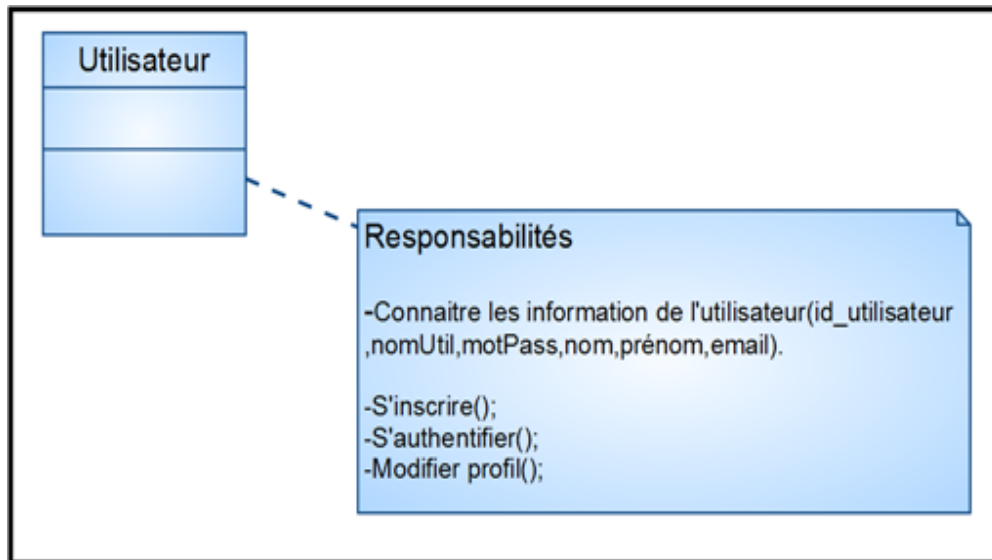


FIGURE 2.16 – Les responsabilités de la classe candidate « Utilisateur »

- **Classe Enseignant**

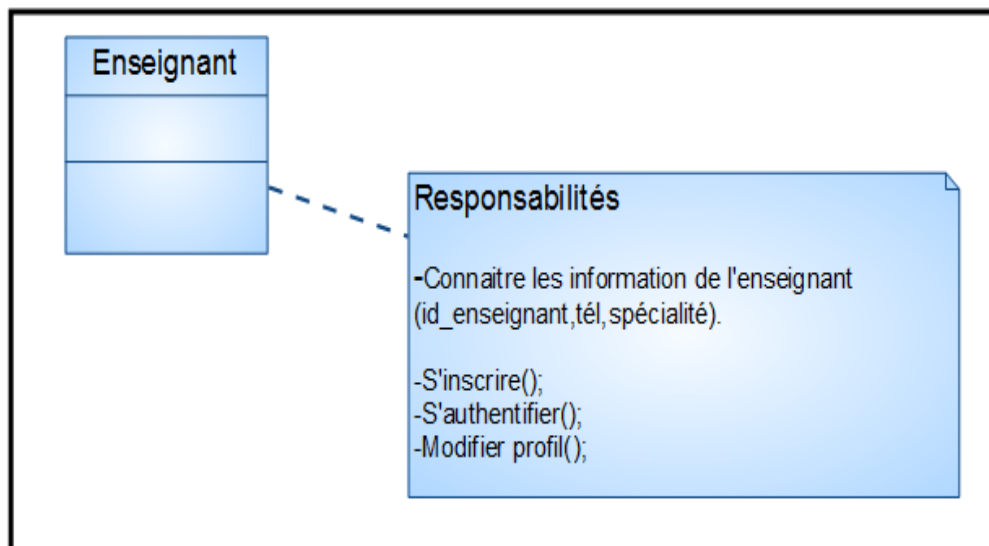


FIGURE 2.17 – Les responsabilités de la classe candidate « Enseignant »

- **Classe Etudiant**

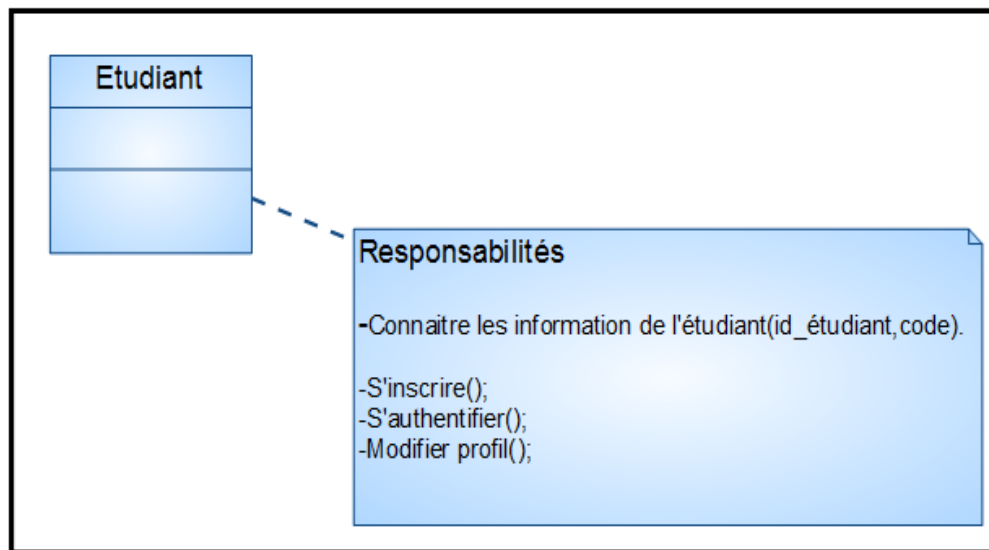


FIGURE 2.18 – Les responsabilités de la classe candidate « Étudiant »

- **Classe Agent de MAJ**

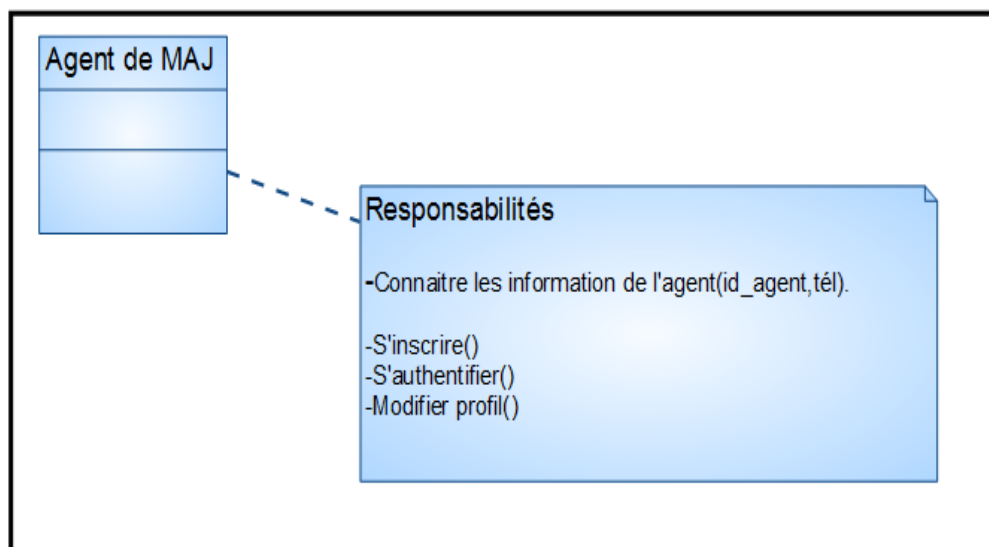


FIGURE 2.19 – Les responsabilités de la classe candidate « Agent de MAJ »

- Classe Groupe

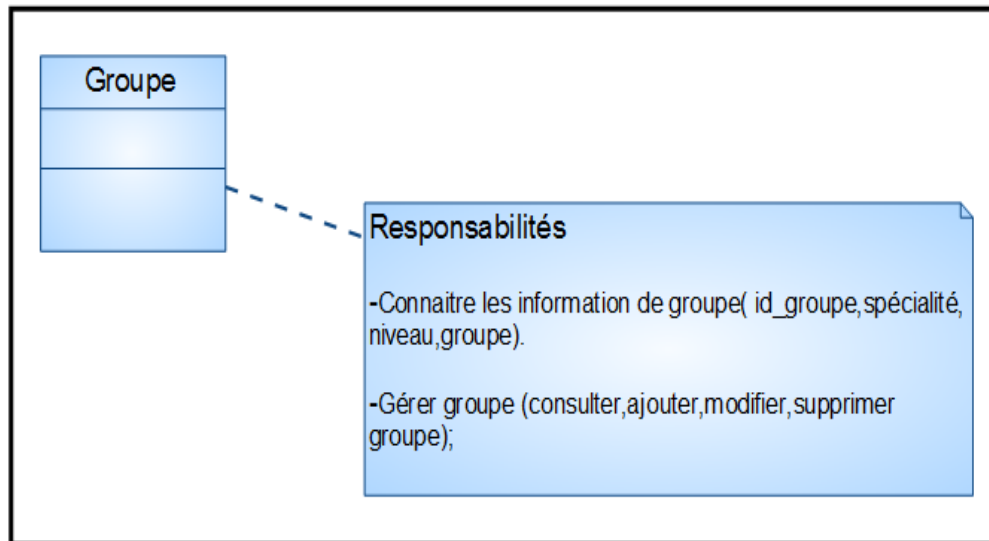


FIGURE 2.20 – Les responsabilités de la classe candidate « Groupe »

- Classe Test

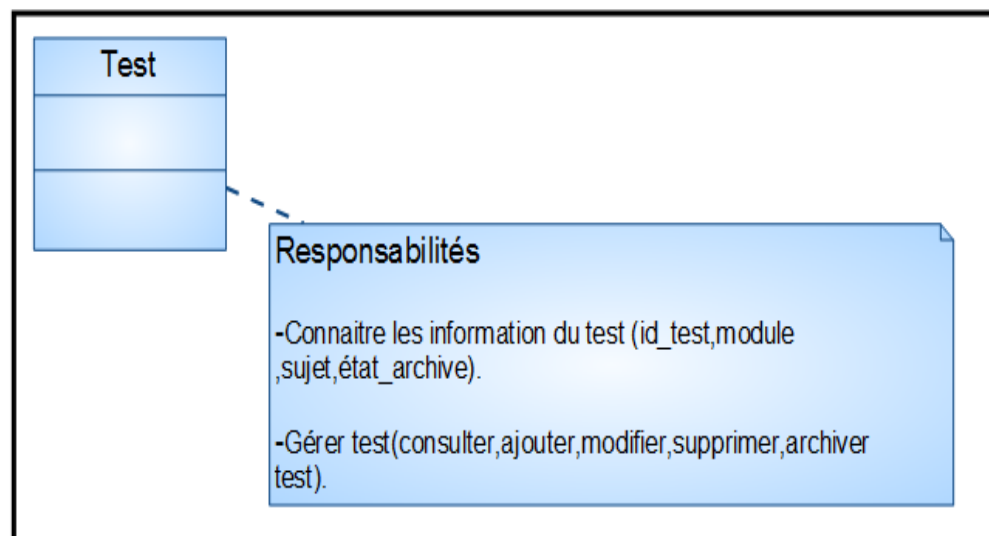


FIGURE 2.21 – Les responsabilités de la classe candidate « Test »

- Classe Question

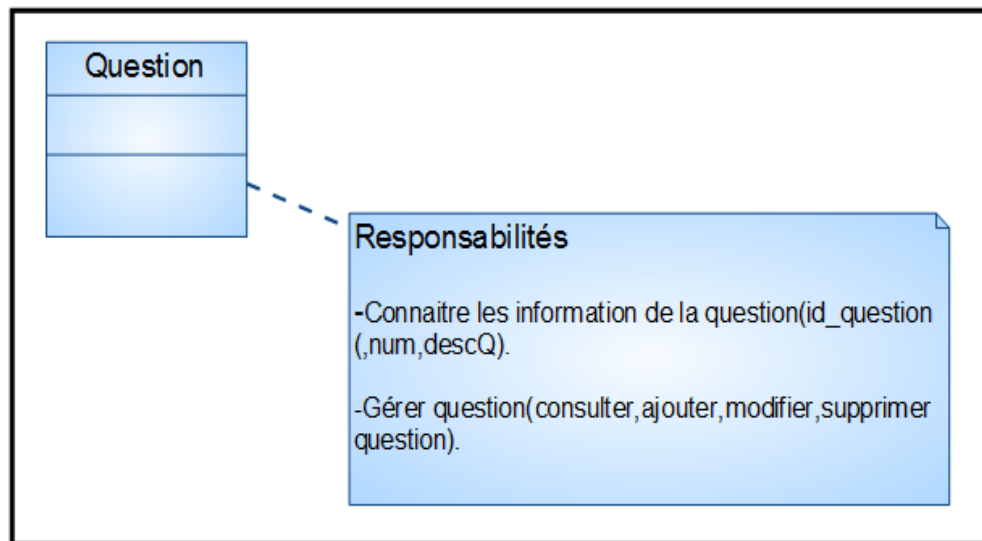


FIGURE 2.22 – Les responsabilités de la classe candidate « Question »

- Classe Choix réponse

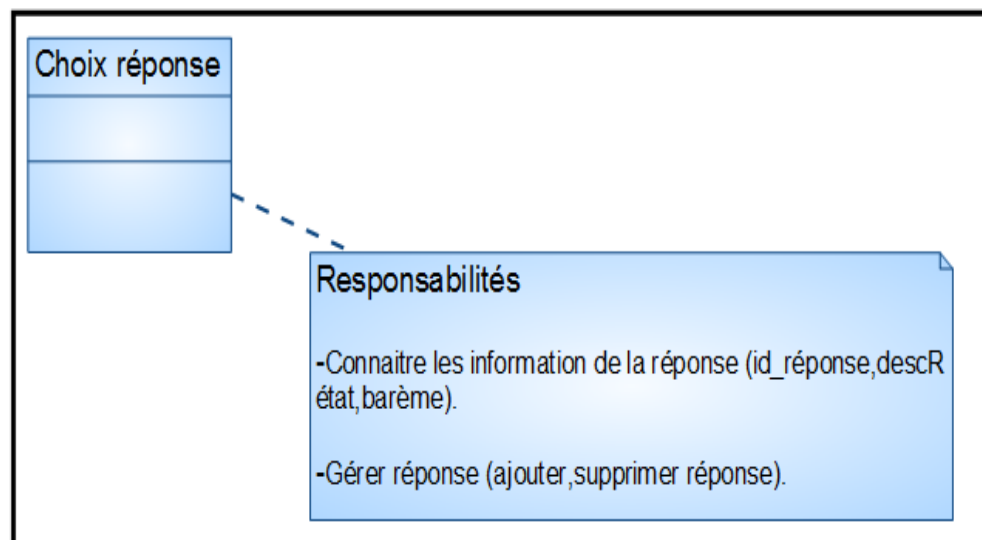


FIGURE 2.23 – Les responsabilités de la classe candidate « Choix réponse »

- Classe Session

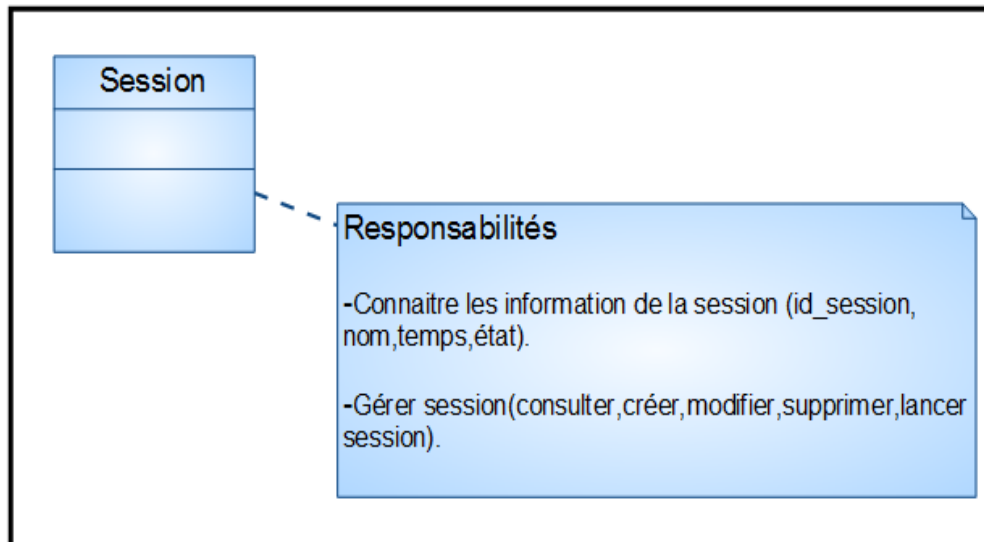


FIGURE 2.24 – Les responsabilités de la classe candidate « Session »

- Classe Éditeur

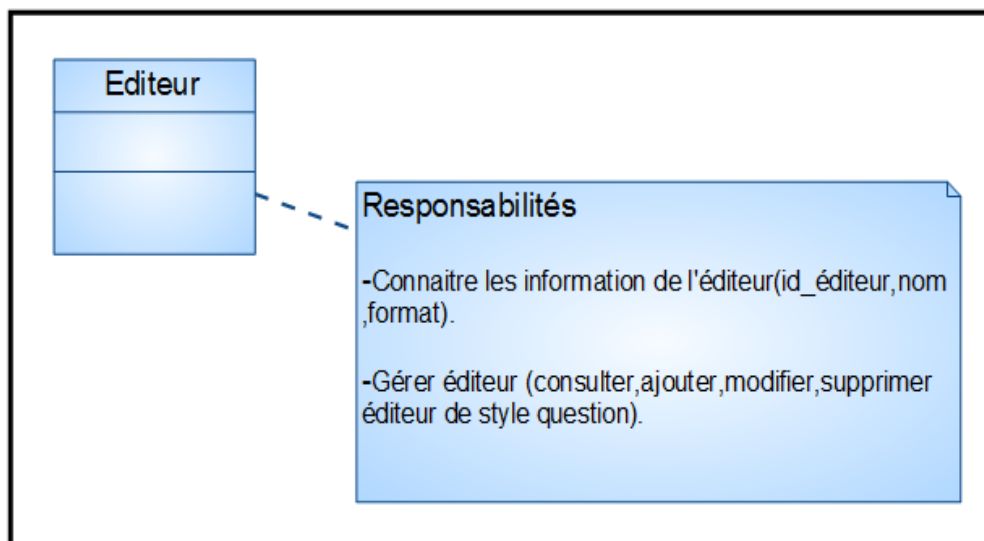


FIGURE 2.25 – Les responsabilités de la classe candidate « Éditeur »

2.2.4 Diagrammes de classes participantes

Afin de donner une meilleure présentation du fonctionnement du système, le diagramme de classes participantes est élaboré, en précisant les différentes méthodes et les entités utilisées. Les figures suivantes sont des captures pour chaque cas :

- «s’inscrire»

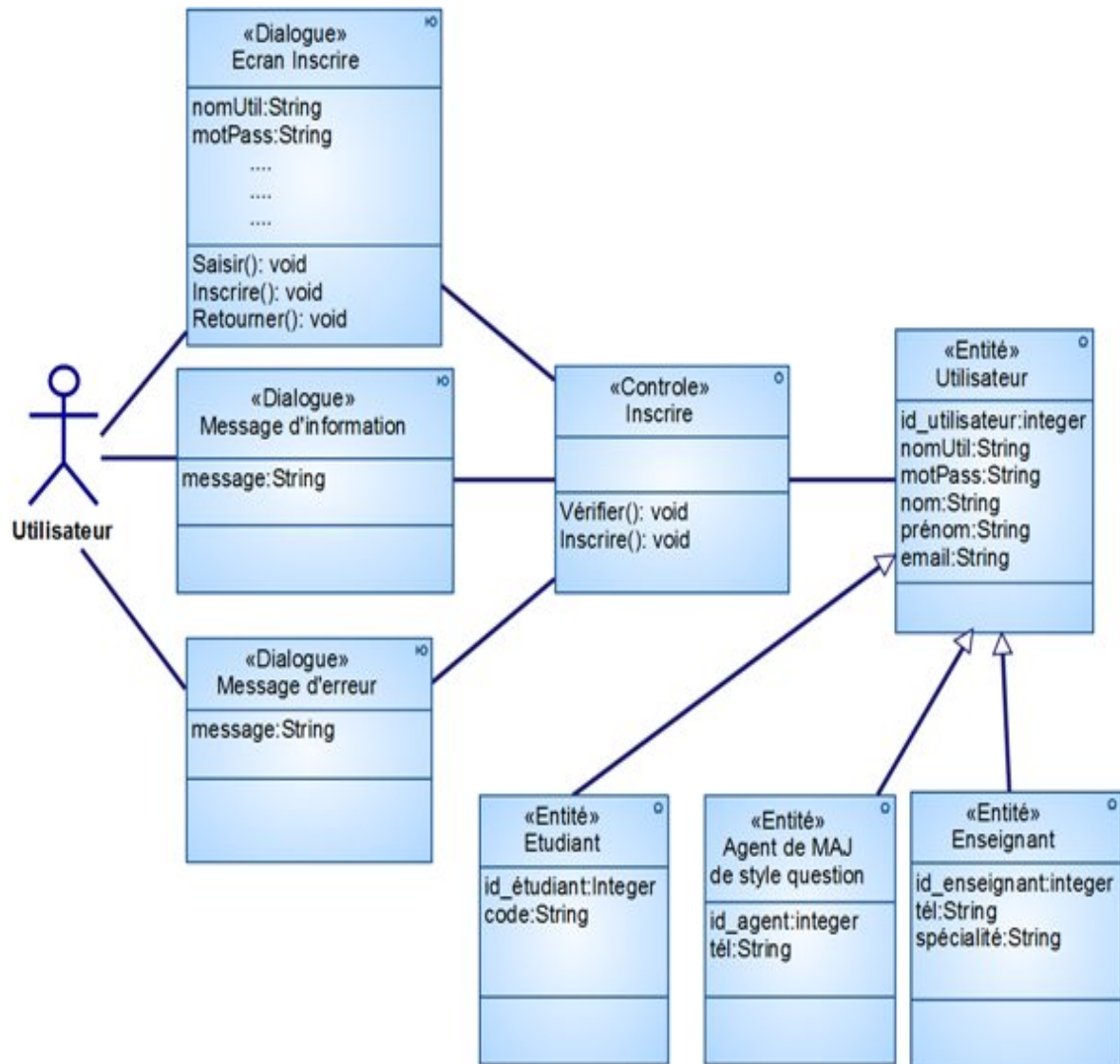


FIGURE 2.26 – Diagramme de classes participantes du cas S’inscrire

- «s'authentifier»

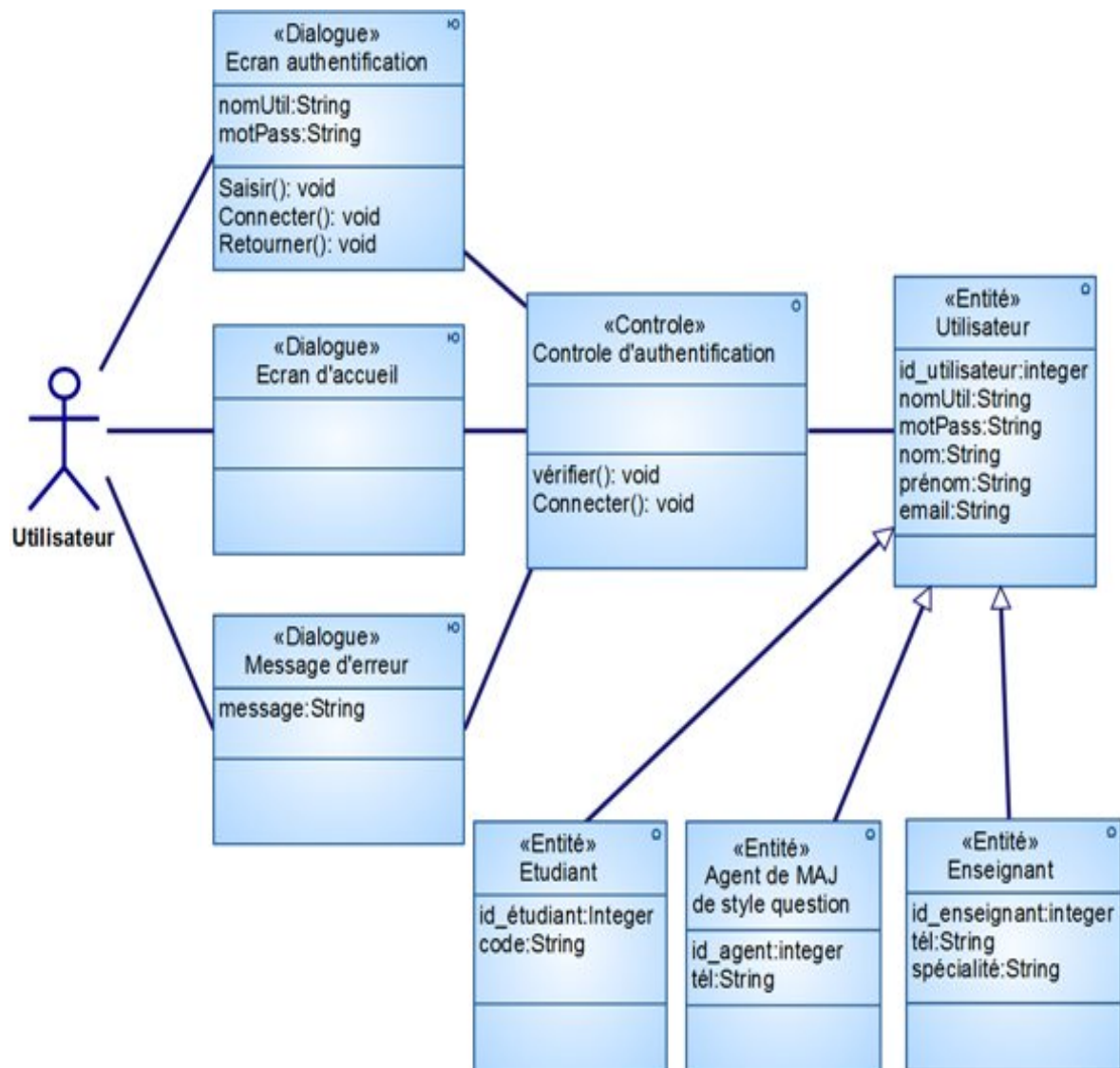


FIGURE 2.27 – Diagramme de classes participantes du cas S'authentifier

- « Ajouter groupe »

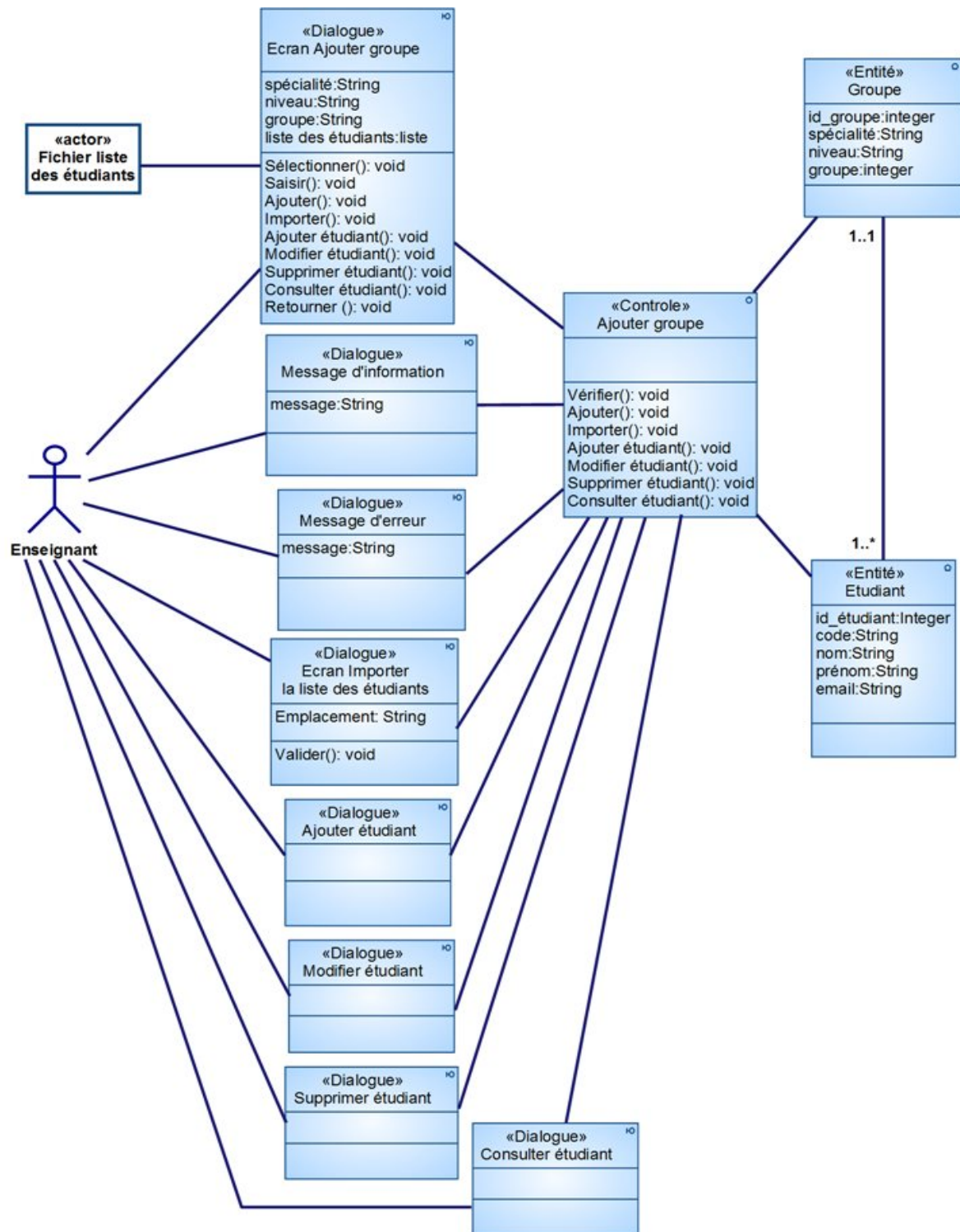


FIGURE 2.28 – Diagramme de classes participantes du cas Ajouter groupe

- « Ajouter étudiant »

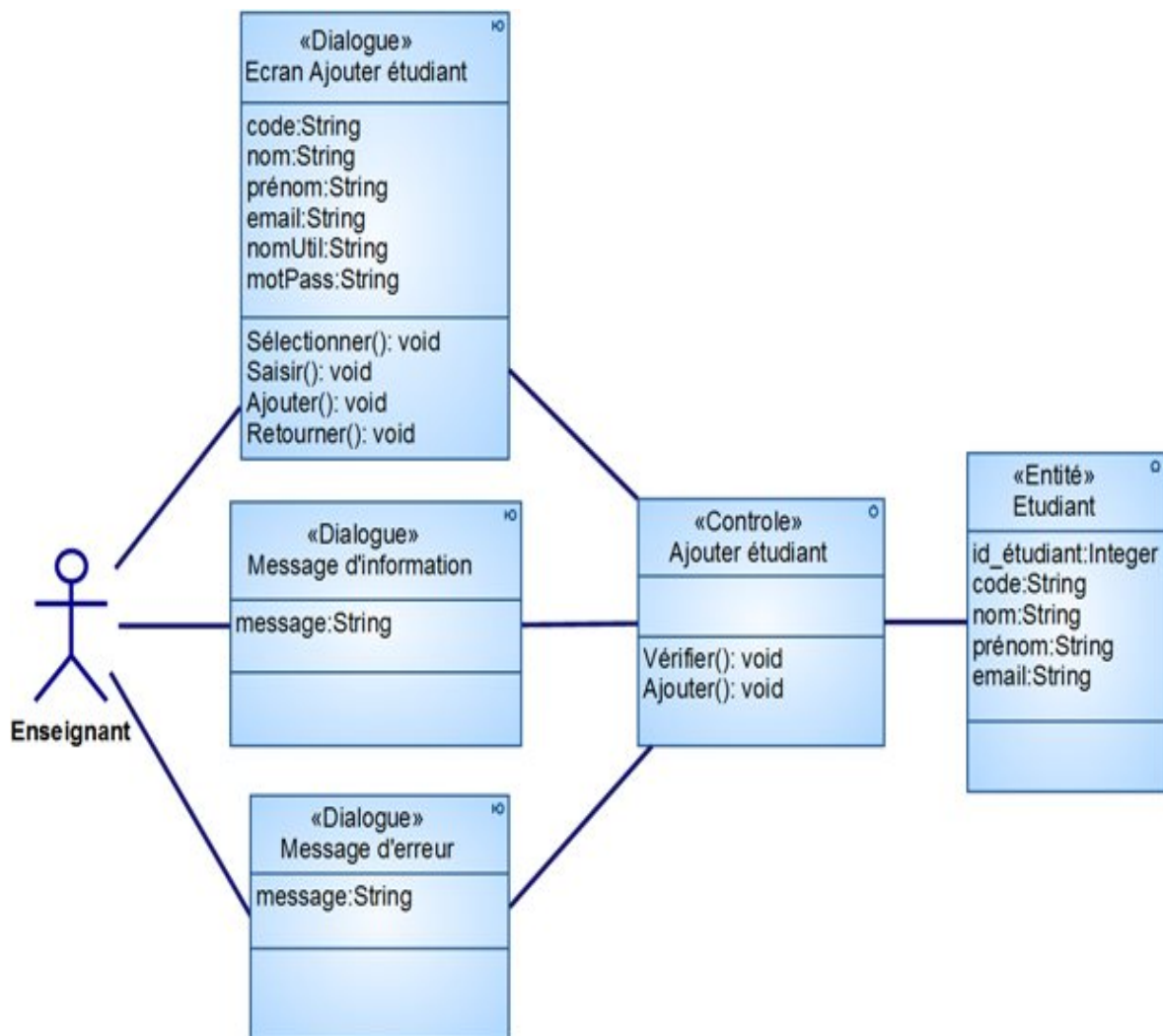


FIGURE 2.29 – Diagramme de classes participantes du cas Ajouter étudiant

- « Consulter test et corrigé type »

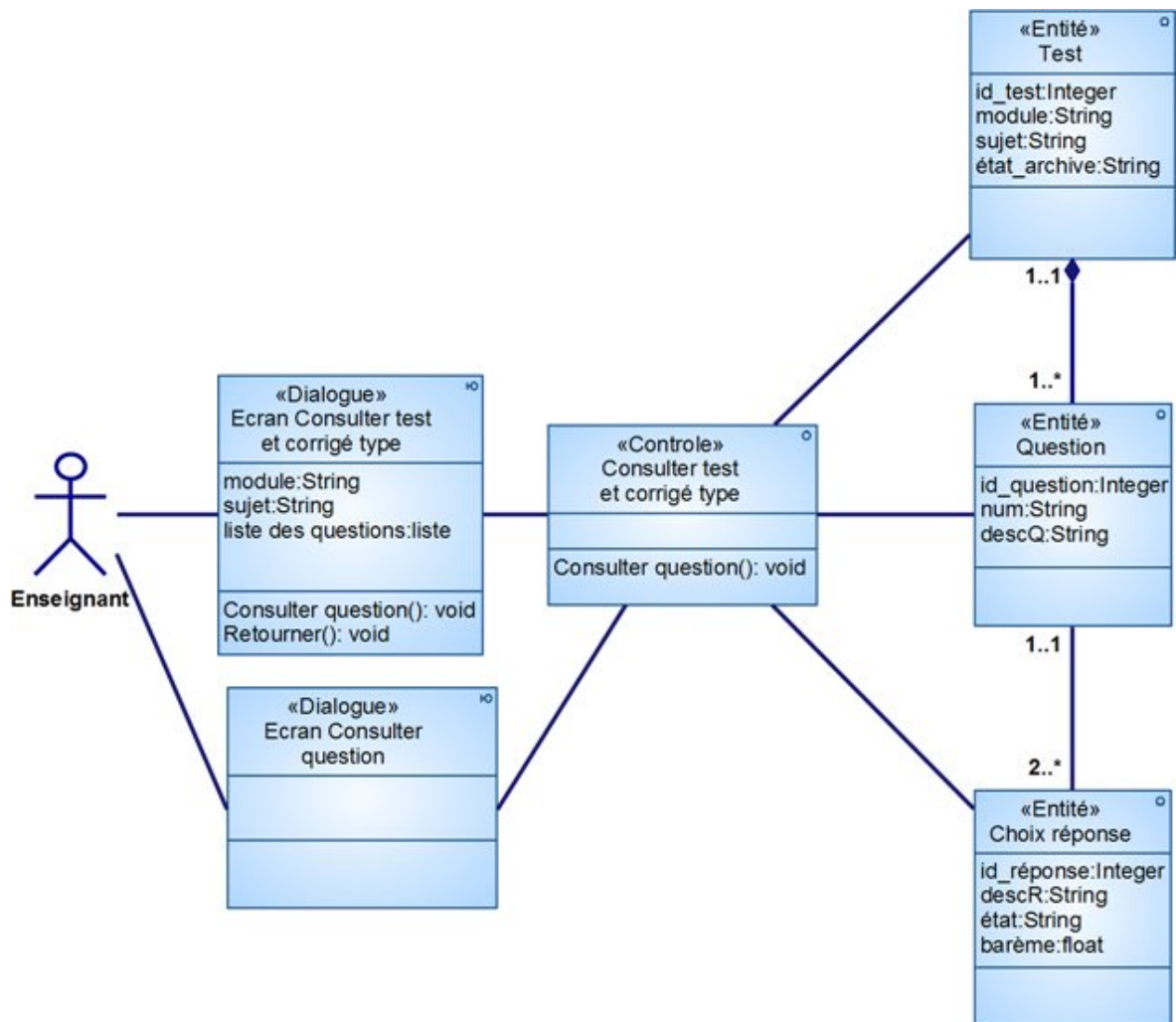


FIGURE 2.30 – Diagramme de classes participantes du cas Consulter test et corrigé type

- « Ajouter test et corrigé type »

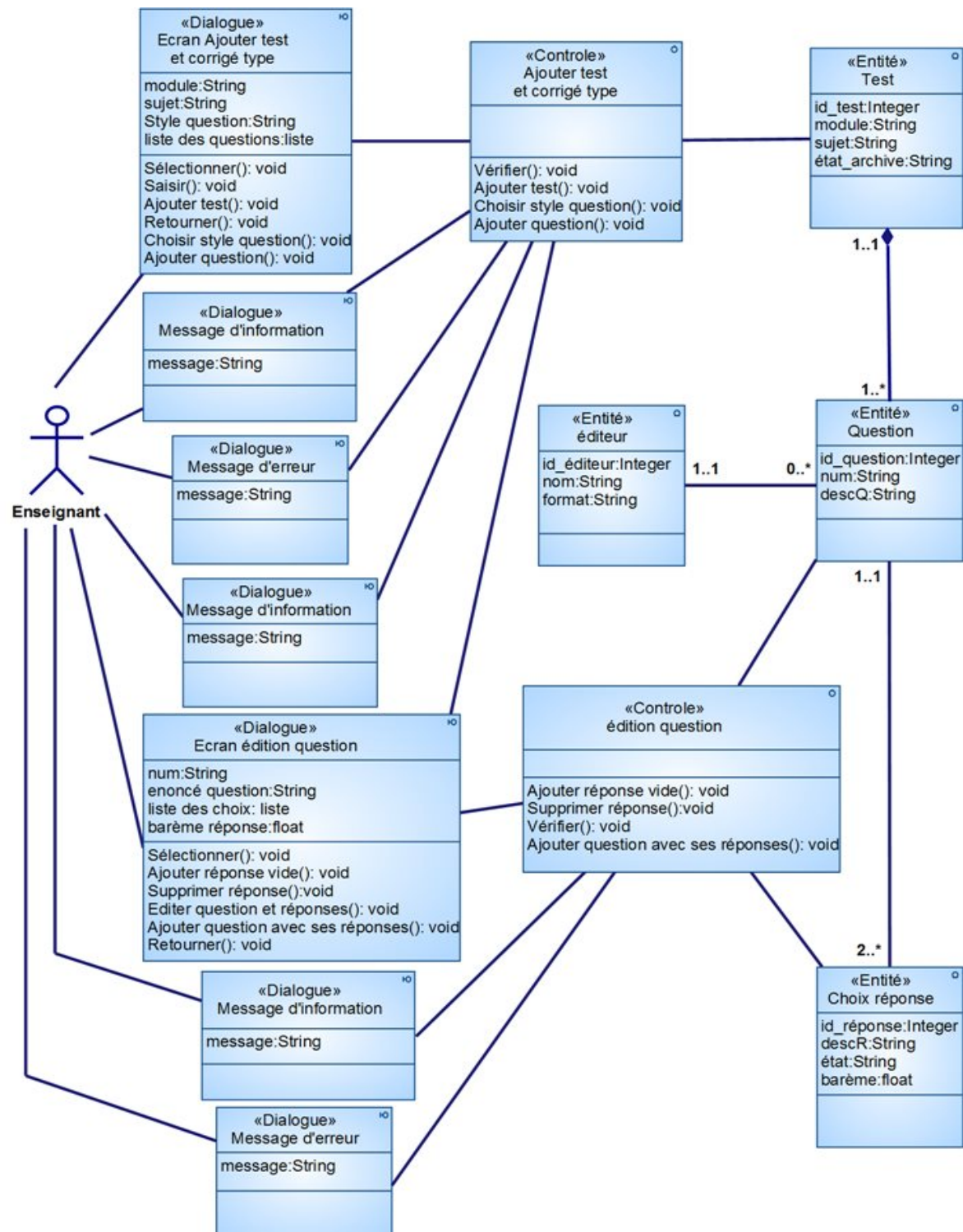


FIGURE 2.31 – Diagramme de classes participant du cas Ajouter test et corrigé type

- « Créer session »

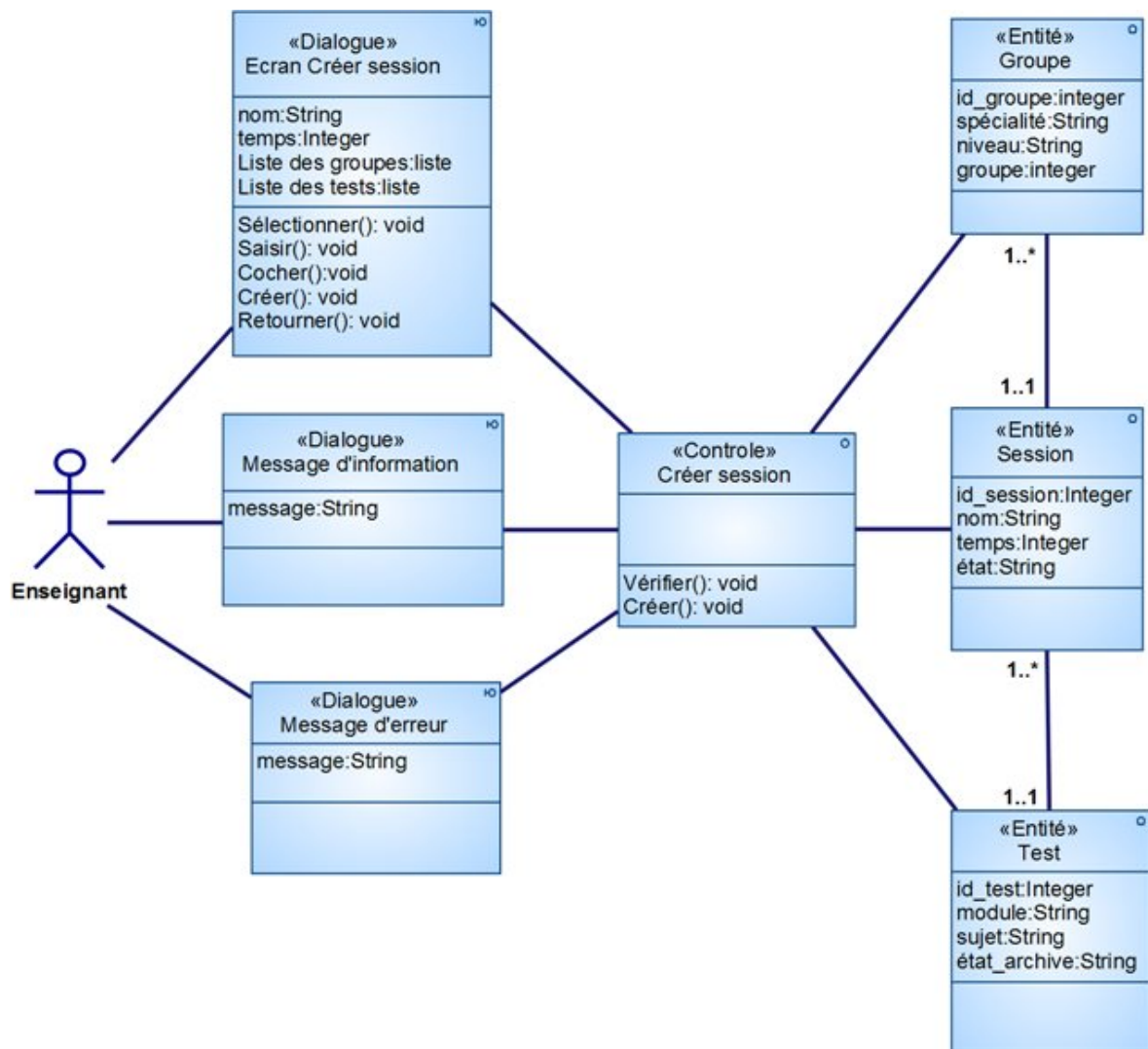


FIGURE 2.32 – Diagramme de classes participantes du cas Créer session

- « Consulter session »

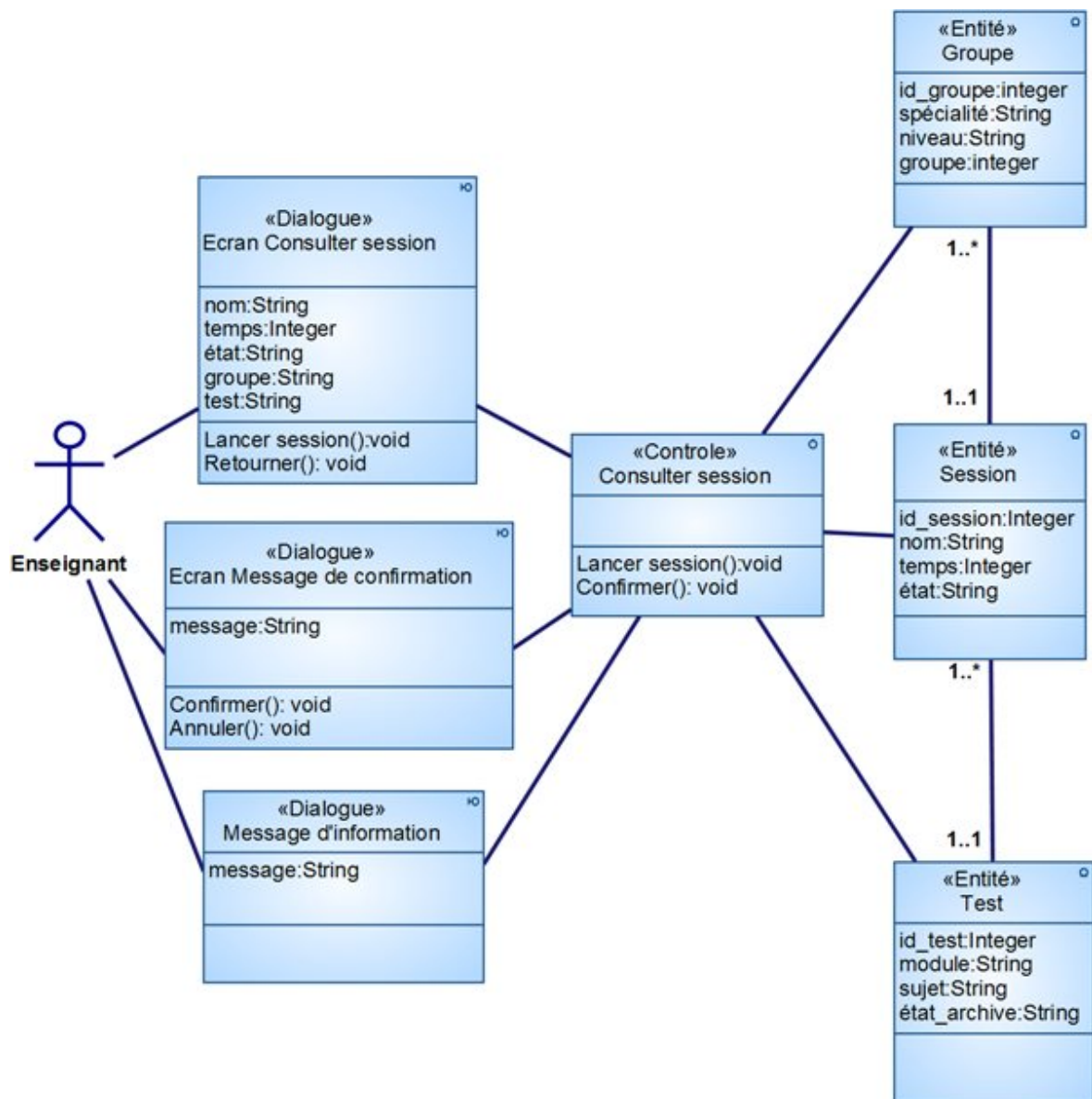


FIGURE 2.33 – Diagramme de classes participantes du cas Consulter session

- « Consulter notes »

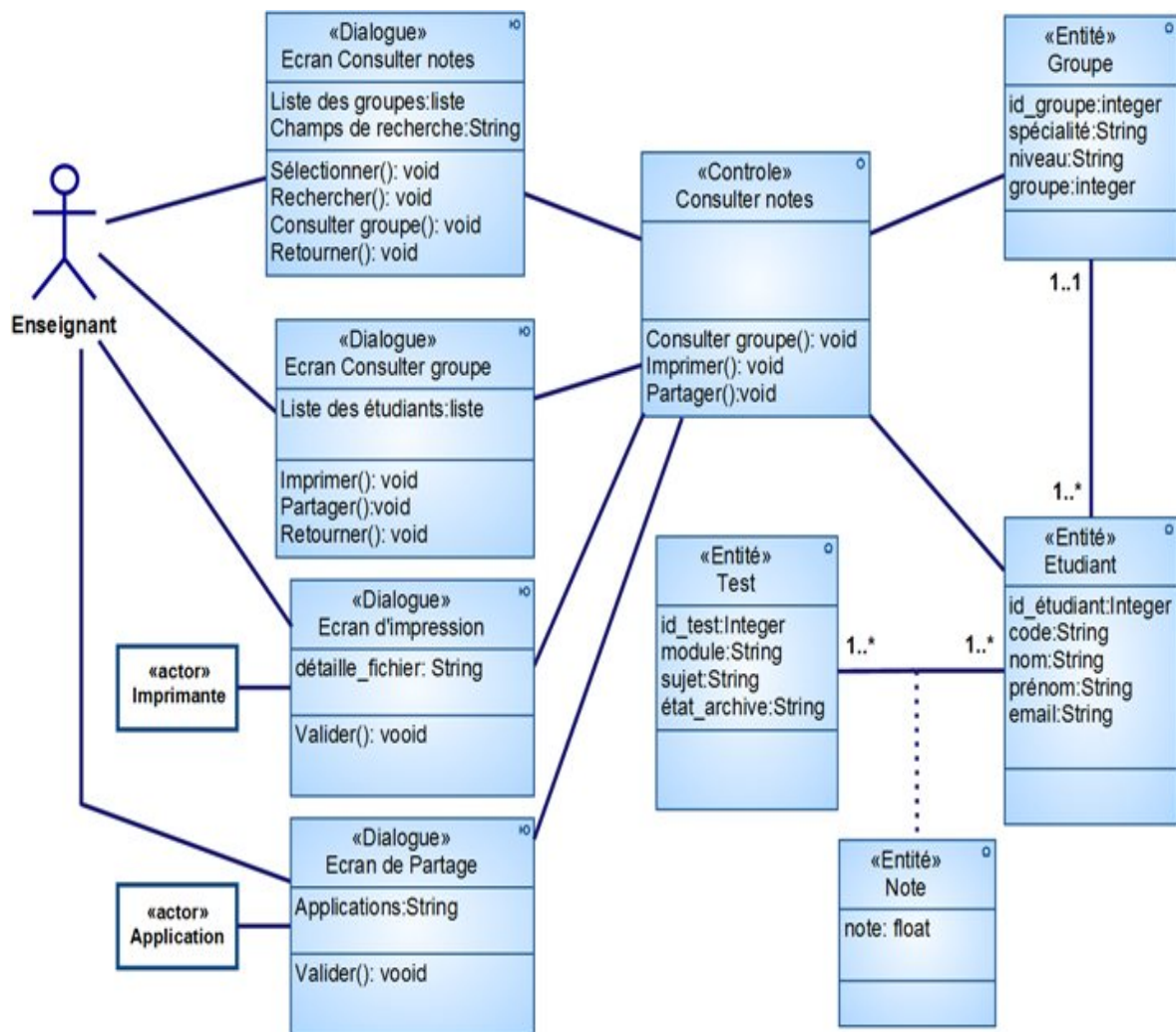


FIGURE 2.34 – Diagramme de classes participantes du cas Consulter notes

- « Ajouter éditeur »

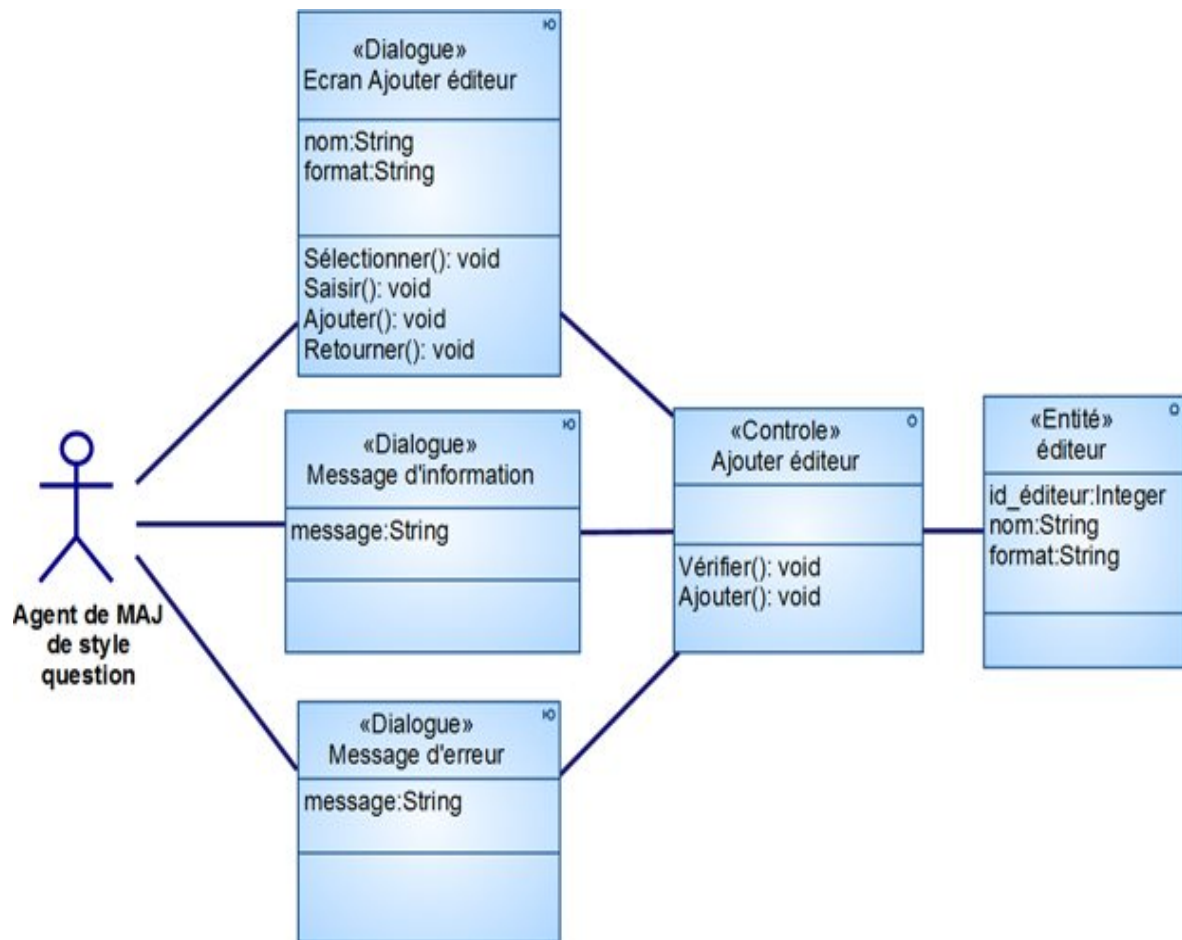


FIGURE 2.35 – Diagramme de classes participantes du cas Ajouter éditeur

- « Modifier éditeur »

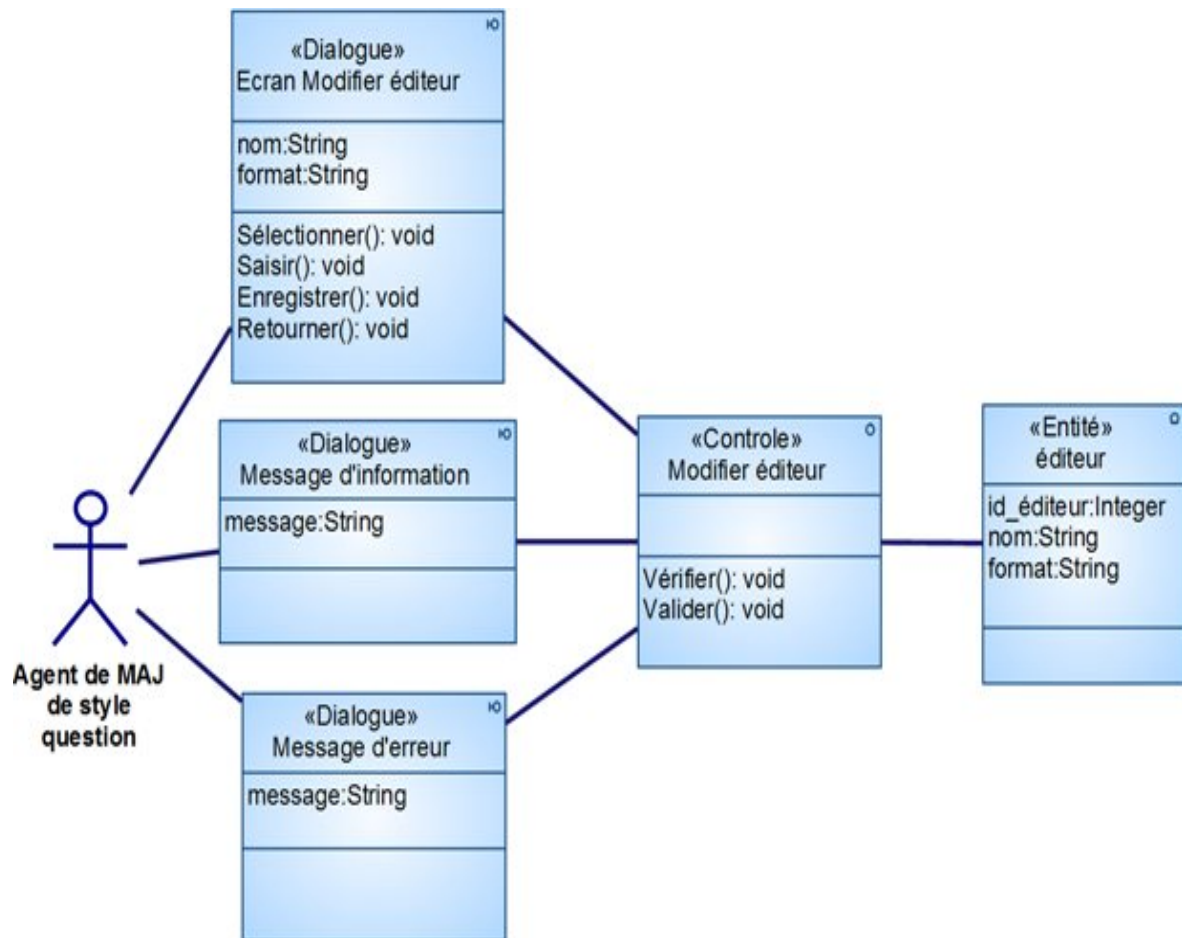


FIGURE 2.36 – Diagramme de classes participantes du cas Modifier éditeur

- « Passer test »

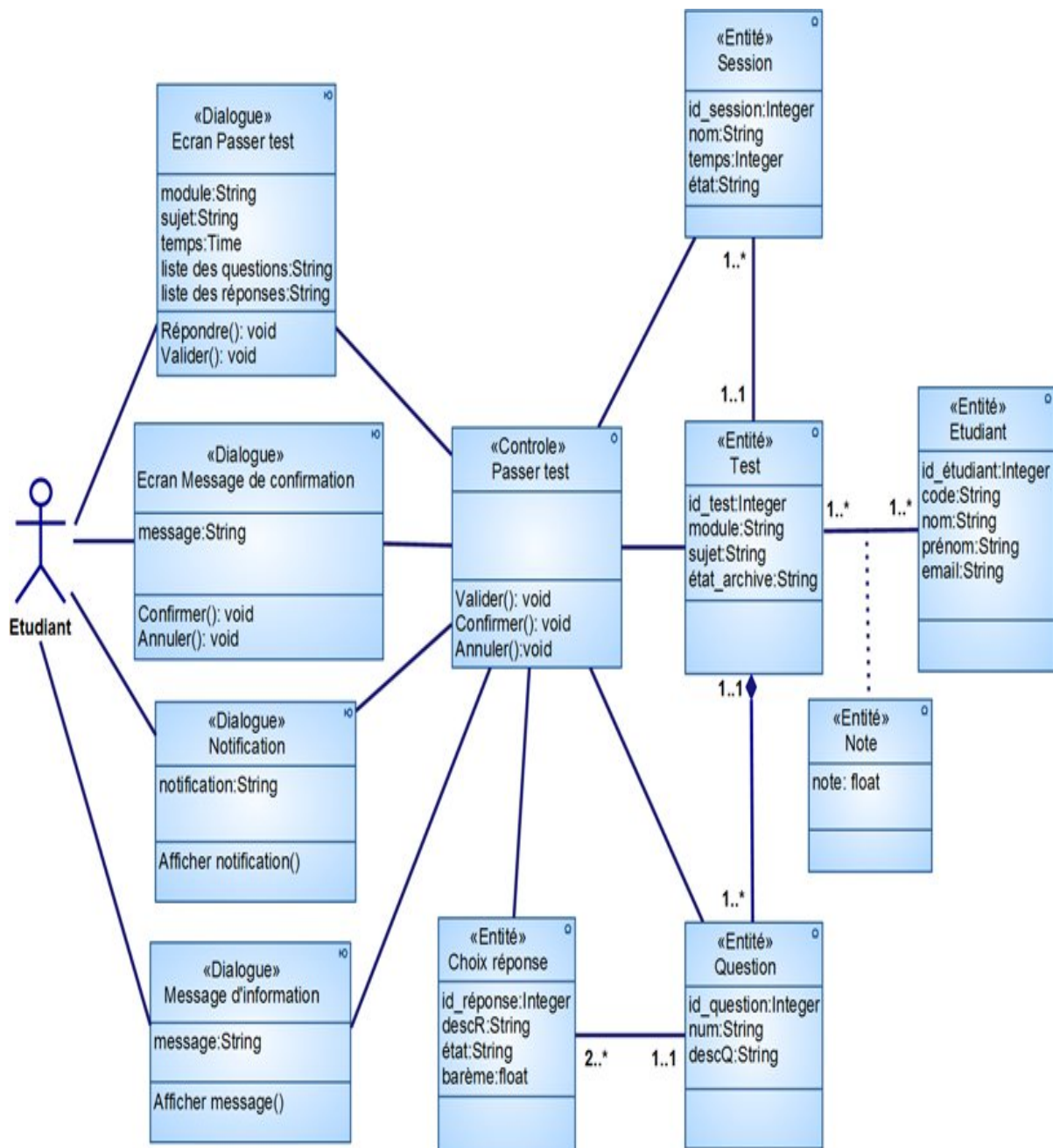


FIGURE 2.37 – Diagramme de classes participantes du cas Passer test

- « Consulter son test »

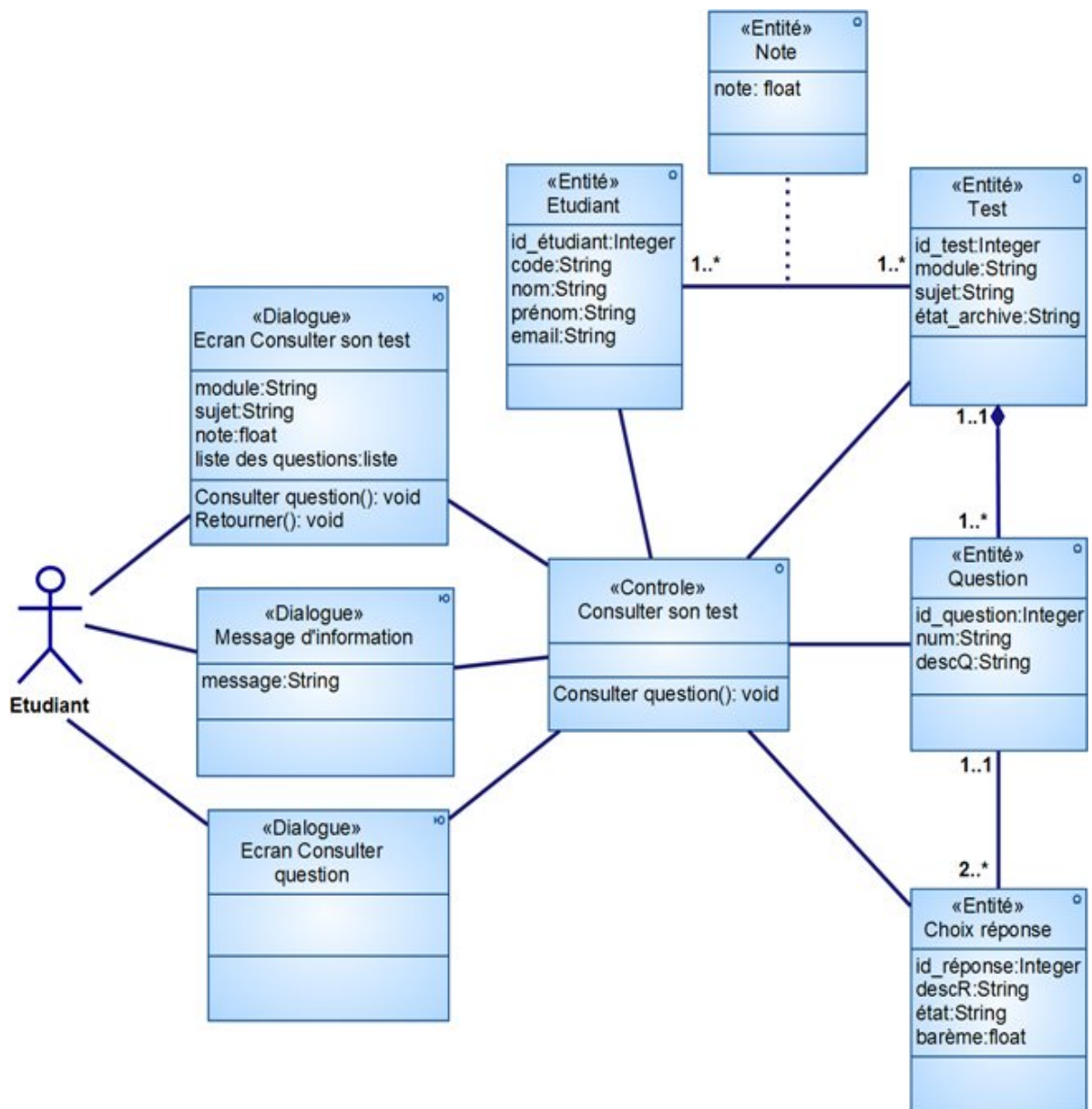


FIGURE 2.38 – Diagramme de classes participantes du cas Consulter son test

2.3 Capture des besoins techniques

Après avoir développé précédemment les besoins fonctionnels qui constituent la branche gauche du cycle Y, nous allons présenter la seconde branche du cycle en occurrence des besoins techniques du système.

La capture des besoins techniques couvre, par complémentarité avec celle des besoins fonctionnels toutes les contraintes qui ne traitent ni de la description du métier des utilisateurs, ni de la description applicative. Le modèle de spécification s'exprime suivant deux points de vue qui sont :

- la spécification technique du point de vue matériel.
- la capture des spécifications logicielles. [2]

2.3.1 La spécification technique du point de vue matériel

Les prérequis techniques ont été exprimés dans l'étude préliminaire, lors de l'expression des besoins opérationnels et de celle des choix stratégiques de développement. Ces choix impliquent des contraintes relatives à la configuration du réseau matériel. Elles sont de nature géographique, organisationnelle, et techniques. Elles concernent les performances d'accès aux données, la sécurité du système, la volumétrie. [2]

- **Style d'architecture en niveaux**

Le style d'architecture en niveaux spécifie le nombre de niveaux géographiques et organisationnels où vont se situer les environnements d'exécution du système.[2]

Dans notre système la configuration géographique impose le développement d'une solution client/serveur à deux niveaux.

- **Spécification de l'architecture à 2 niveaux**

une architecture à deux niveaux aussi appelée client-serveur, est une architecture où un poste client exécute une application se trouvant sur un serveur. Cela effectué comme un dialogue entre le client et le serveur dans lequel :

- **Client** : demande une ressource.
- **Serveur** : la lui fournit directement.

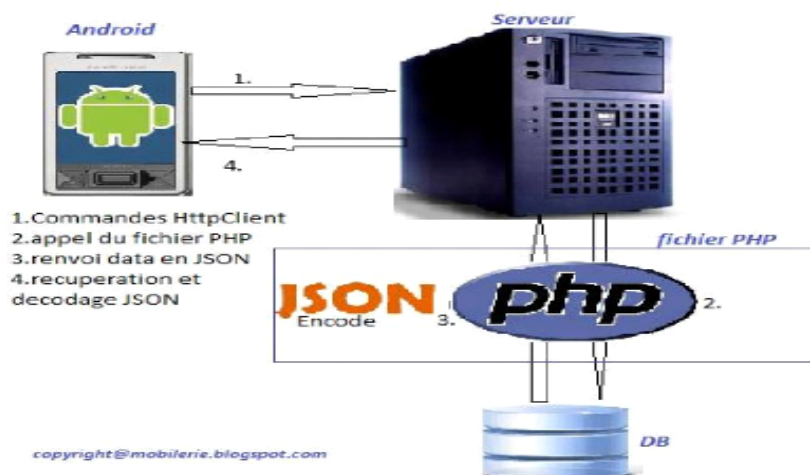


FIGURE 2.39 – Architecture à 2 niveaux

2.3.2 La capture des spécifications logicielles

Une fois que les spécifications techniques et d'architecture sont exprimées, on peut s'intéresser aux fonctionnalités propres du système technique en procédant à une spécification logicielle. Dans ce cas, on utilise les cas d'utilisation de manière différente que pour la spécification fonctionnelle. C'est pourquoi nous avons introduit le concept d'exploitant et de cas d'utilisation technique. [2]

- **Exploitant :**

l'exploitant est un acteur au sens d'UML, si ce n'est qu'il ne bénéficie que des fonctionnalités techniques du système.

- **Cas d'utilisation technique :**

un cas d'utilisation technique est destiné à l'exploitant, c'est une séquence d'action produisant une valeur ajoutée opérationnelle ou purement techniques.

Identification des exploitants du système

Les exploitants de notre système sont la majorité des acteurs de la branche fonctionnelle, sont donc utilisateurs dans la dimension technique. et qui sont : l'enseignant, l'étudiant et l'agent de MAJ.

Identification des cas d'utilisation technique

Les principaux cas d'utilisation techniques de notre système sont illustrés par la figure suivante :

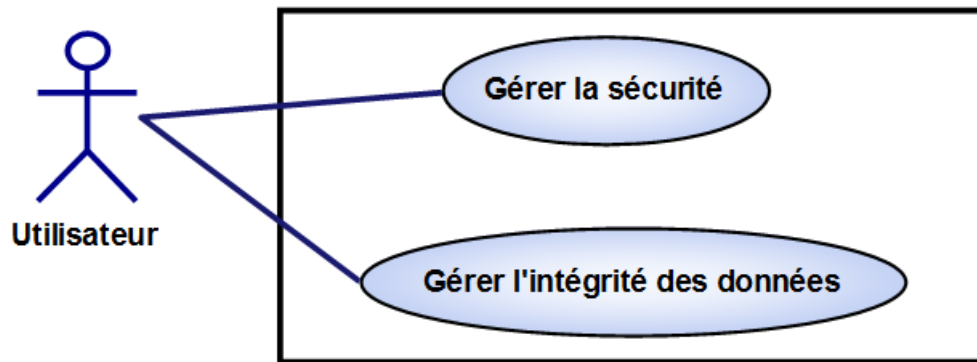


FIGURE 2.40 – Modèle de spécification logicielle du système.

Description des cas d'utilisation techniques :

✓ 1- « Gérer la sécurité »

• « S'authentifier »

– Description textuelle

Cas d'utilisation	S'authentifier.
Acteur	Utilisateur.
But	Ce cas permet de vérifier l'autorisation d'accès à des fonctionnalités de l'application.
Pré condition	L'utilisateur possède un compte.
Poste condition	L'utilisateur est authentifié.
Scénario nominal	1-L'utilisateur demande l'accès à l'application. 2-Le système affiche le formulaire d'authentification. 3-L'utilisateur introduit son identifiant et son mot de passe et confirme l'authentification. 4-Le système vérifie les informations d'authentification. 5-Le système renvoie l'interface correspondante.
Scénario alternatif	• L'identifiant et/ou mot de passe incorrect. Le système affiche un message d'erreur. Reprise de l'enchaînement du scénario nominal au point 2.
Scénario optionnelle	• L'utilisateur peut annuler l'opération en cours à n'importe quel moment pourvu qu'il ne l'a pas validé.

TABLE 2.16: Description textuelle du cas d'utilisation

« S'authentifier »

– Diagramme de séquence :

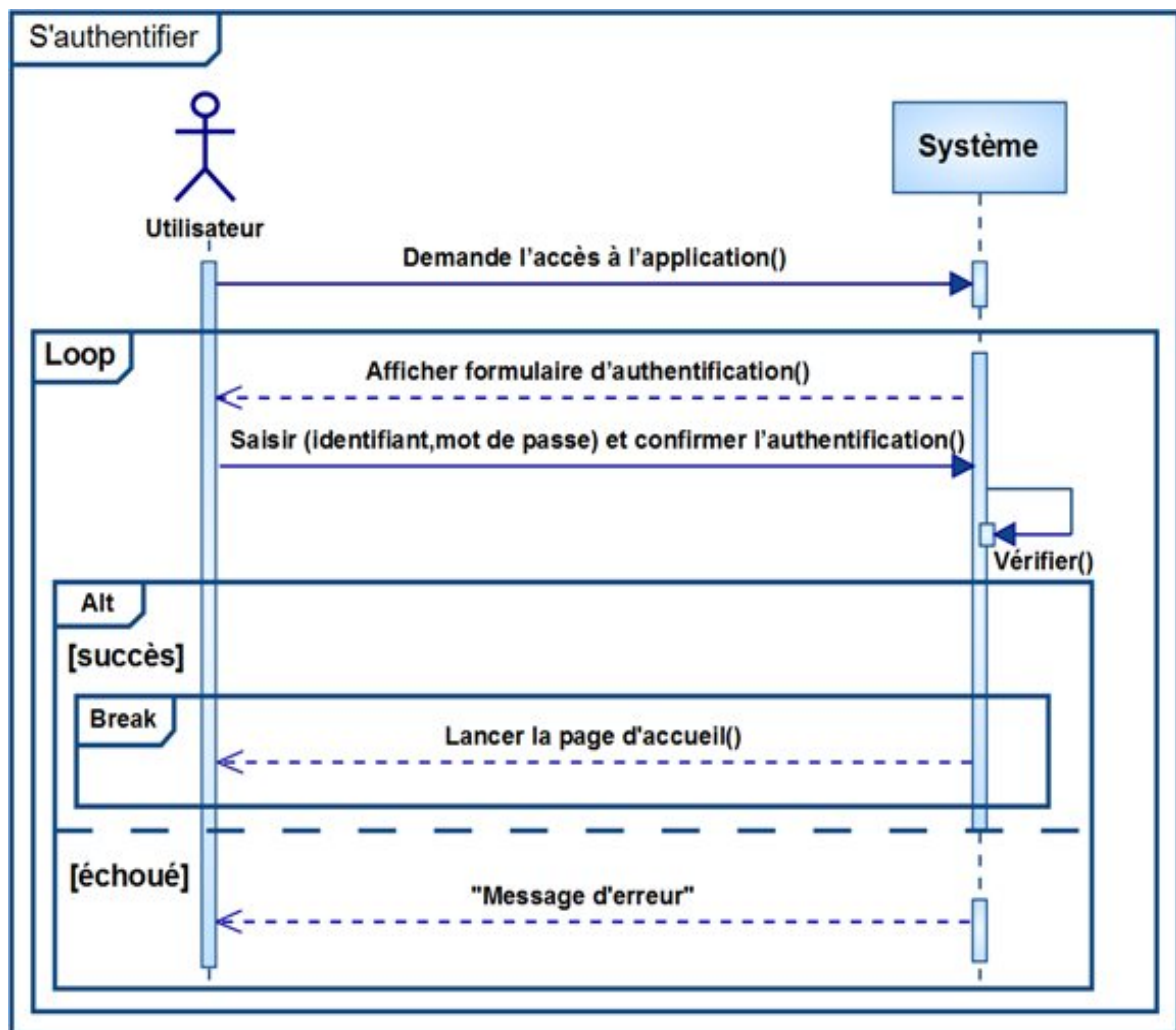


FIGURE 2.41 – Diagramme de séquence du cas S'authentifier

✓ 2- « Gérer l'intégrité des données »

– Description textuelle

Cas d'utilisation	Gérer l'intégrité des données.
Acteur	Utilisateur.
But	Vérifier l'intégrité des données.
Pré condition	L'utilisateur s'authentifie.
Scénario nominal	1-L'utilisateur saisit les données. 2-Le système contrôle la validité de chaque champ.
Scénario alternatif	• Le système affiche une notification « champs non valide ». Le scénario reprend au point 1.

TABLE 2.17: Description textuelle du cas d'utilisation
« Gérer l'intégrité des données »

– Diagramme de séquence

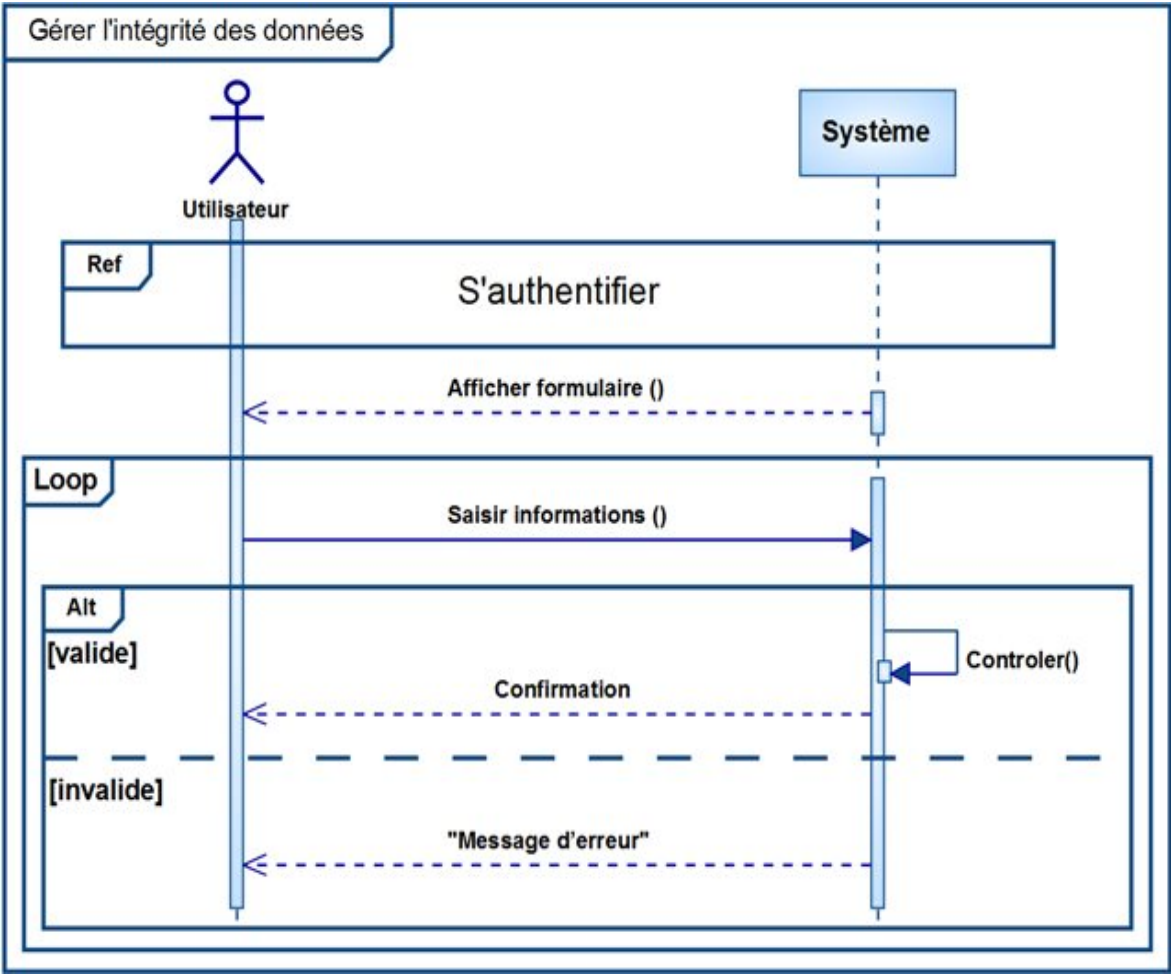


FIGURE 2.42 – Diagramme de séquence du cas Gérer l'intégrité des données

Conclusion

La phase de capture des besoins est une étape qui œuvre d'un côté pour compléter les recueils des besoins opérés pendant la phase d'étude préliminaire, et d'un autre côté, elle donne une première vue pour la prochaine étape. Cette dernière qu'elle sera le noyau du prochain chapitre.

Phase d'analyse

Introduction

La phase d'analyse représente la deuxième étape de la branche gauche du cycle en Y. Elle est consacrée à l'analyse d'objet de notre système juste après l'étude préliminaire et la capture des besoins fonctionnels et techniques. Elle consiste à étudier précisément les spécifications fonctionnelles afin d'obtenir une idée de ce que va réaliser le système en matière de métiers. Elle est constituée de 3 activités :

- le découpage en catégories ;
- le développement du modèle statique ;
- le développement du modèle dynamique ;

la figure suivante représente la situation de cette phase dans le processus 2TUP

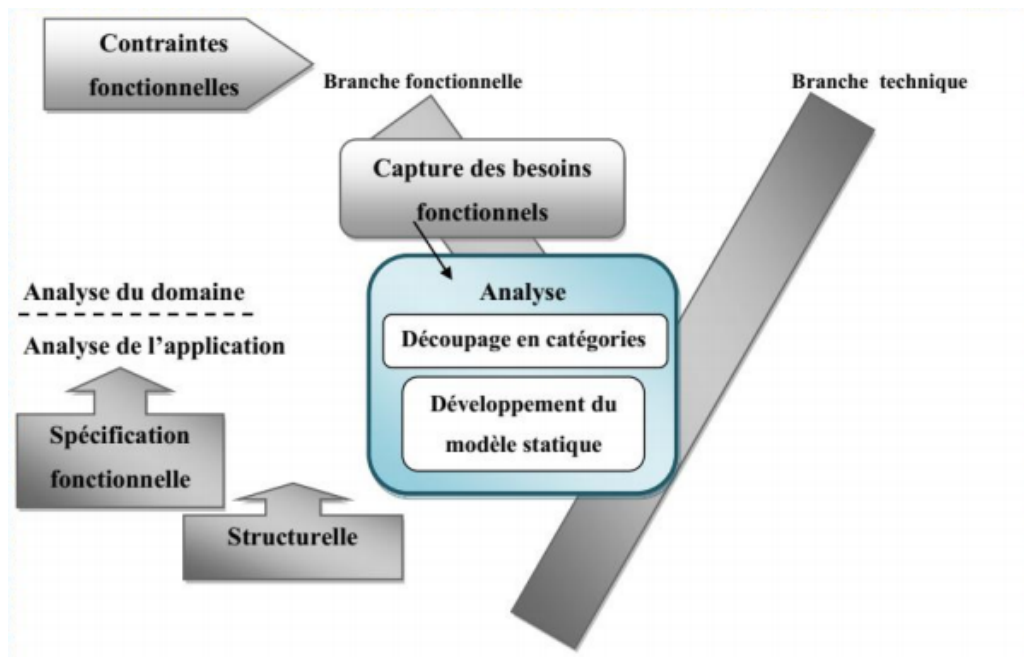


FIGURE 3.1 – Situation de la phase d'analyse dans 2TUP.

3.1 Découpage en catégories

Le découpage en catégories constitue la première activité de l'étape d'analyse, permet de déterminer les classes fondamentales du projet en utilisant les diagrammes de classes participantes dégagées dans l'étape de captures des besoins fonctionnels, Il utilise la notion de package pour définir des catégories de classes d'analyse et découper le modèle UML en blocs logiques les plus indépendants possibles. [2]

Le découpage en catégories se fait en 3 étapes :

3.1.1 La répartition des classes candidates en catégories

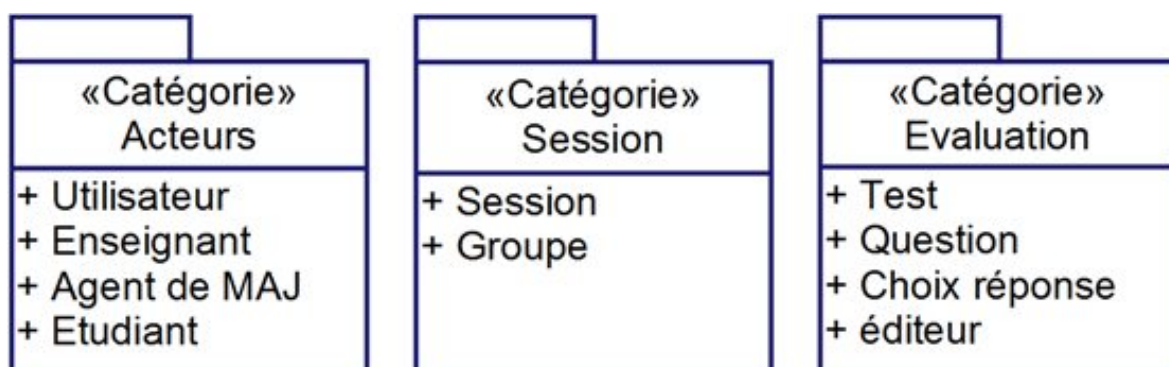


FIGURE 3.2 – Découpage en catégories

3.1.2 Élaboration des diagrammes de classes préliminaires par catégorie

- Catégorie « Acteurs »

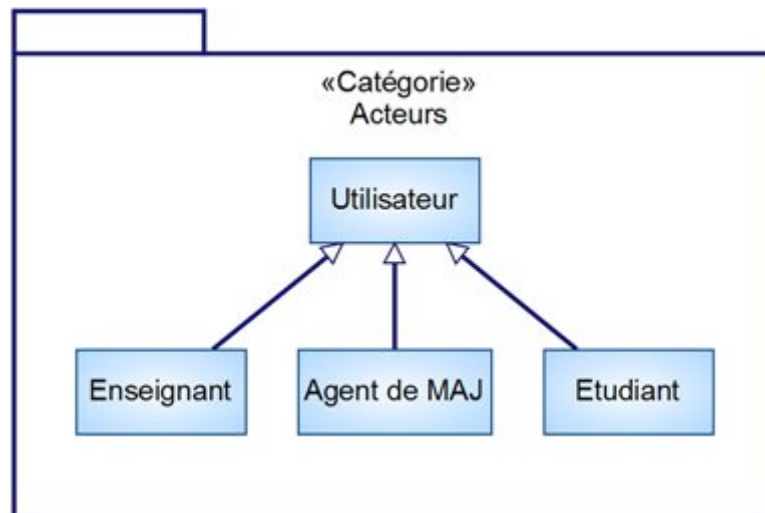


FIGURE 3.3 – Diagramme des classes préliminaire de la Catégorie « Acteurs »

- Catégorie « Session »

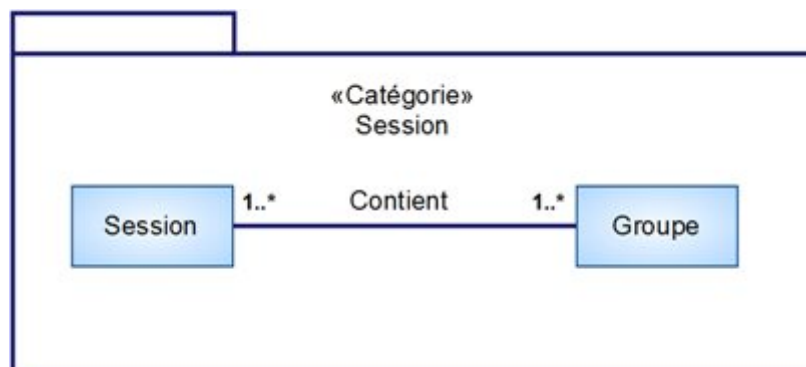


FIGURE 3.4 – Diagramme des classes préliminaire de la Catégorie « Session »

- Catégorie « Évaluation »

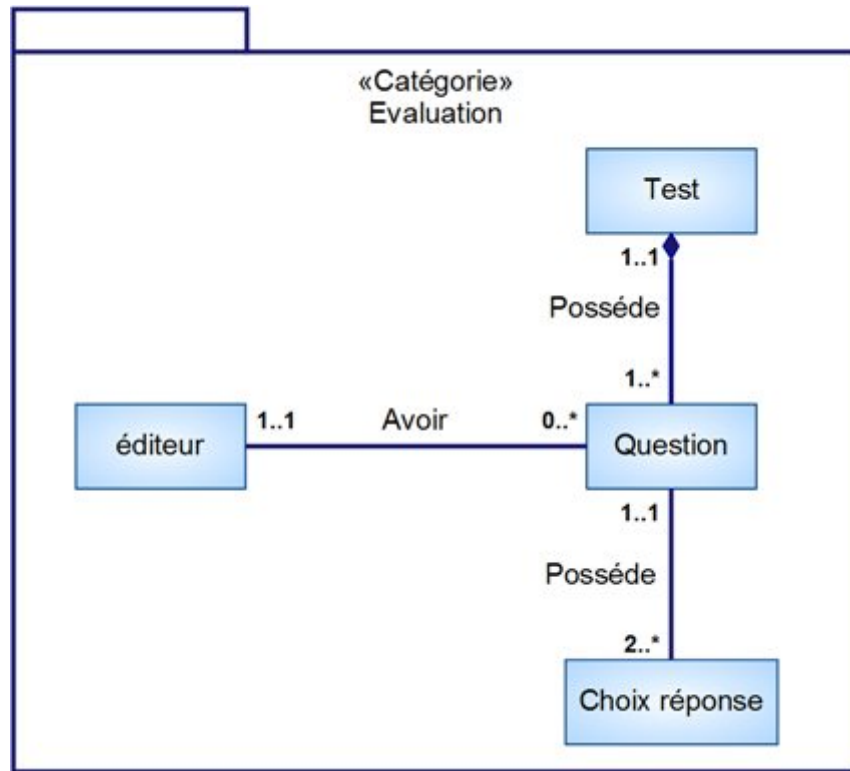


FIGURE 3.5 – Diagramme des classes préliminaire de la Catégorie « Évaluation »

3.1.3 Dépendance entre catégories

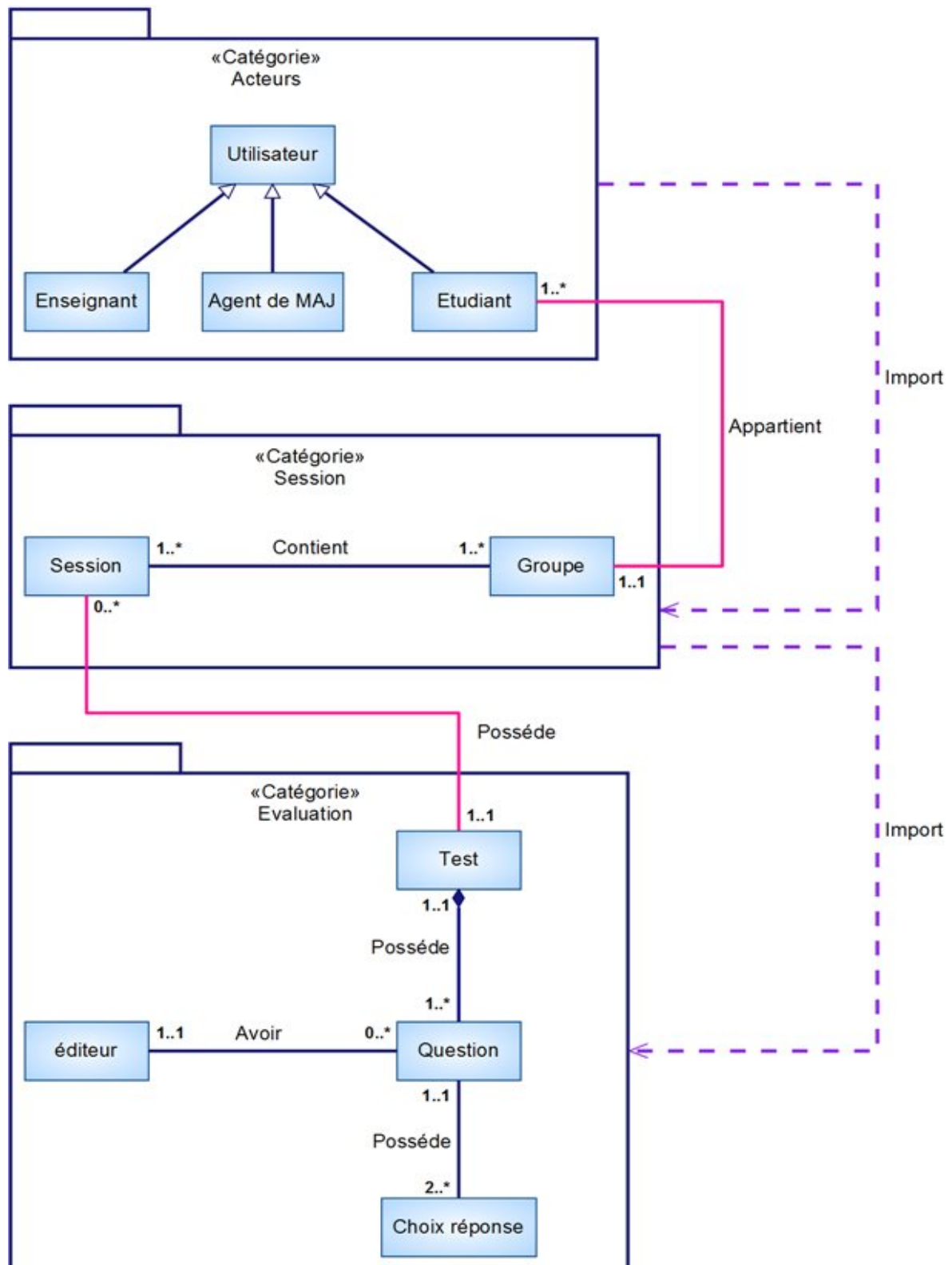


FIGURE 3.6 – Dépendance entre catégories

3.2 Développement du modèle statique

Le développement du modèle statique constitue la deuxième activité de l'étape d'analyse. Elle se situe sur la branche gauche du cycle en Y et succède au découpage en catégories. Il consiste à détailler, compléter et optimiser les diagrammes de classes participantes obtenus et réorganisés lors du découpage en catégories. [2]

3.2.1 Diagramme de classes de la catégorie « Acteurs »

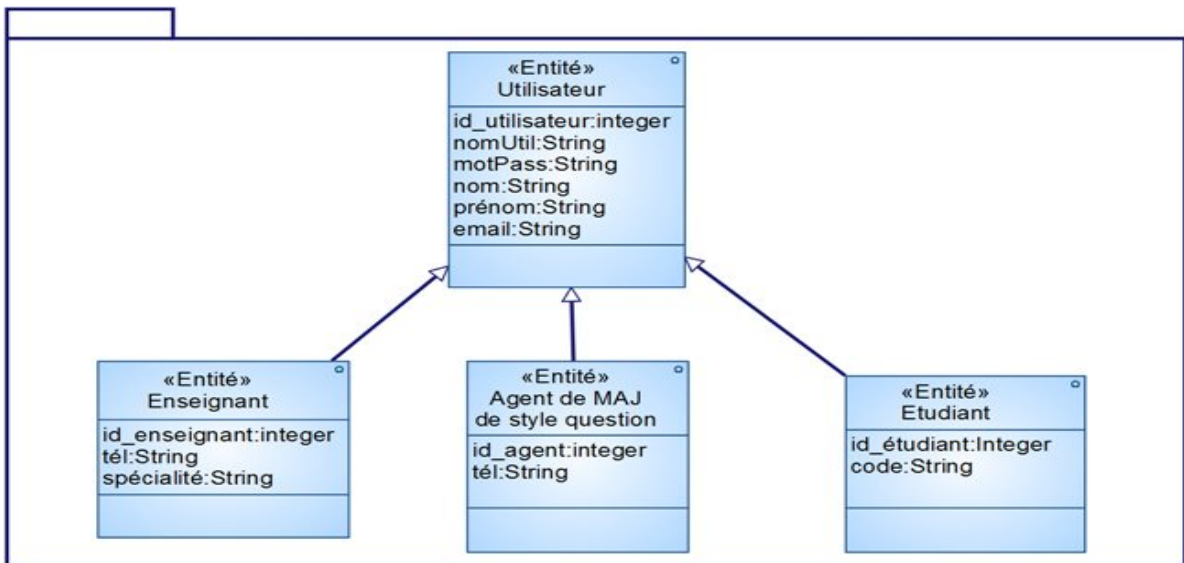


FIGURE 3.7 – Diagramme de classes de la catégorie « Acteurs »

3.2.2 Diagramme de classes de la catégorie « Session »

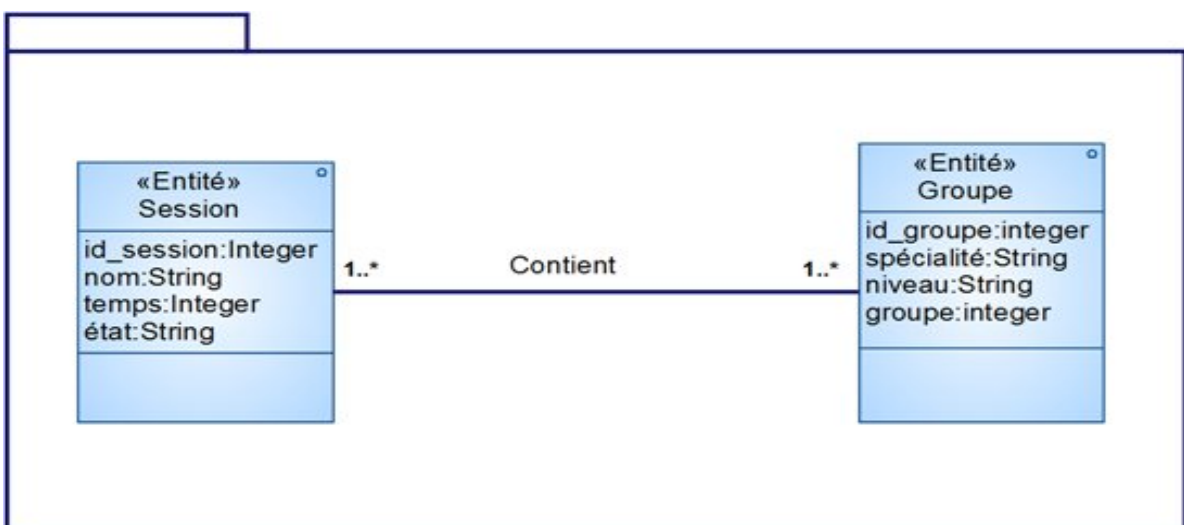


FIGURE 3.8 – Diagramme de classes de la catégorie « Session »

3.2.3 Diagramme de classes de la catégorie « Évaluation »

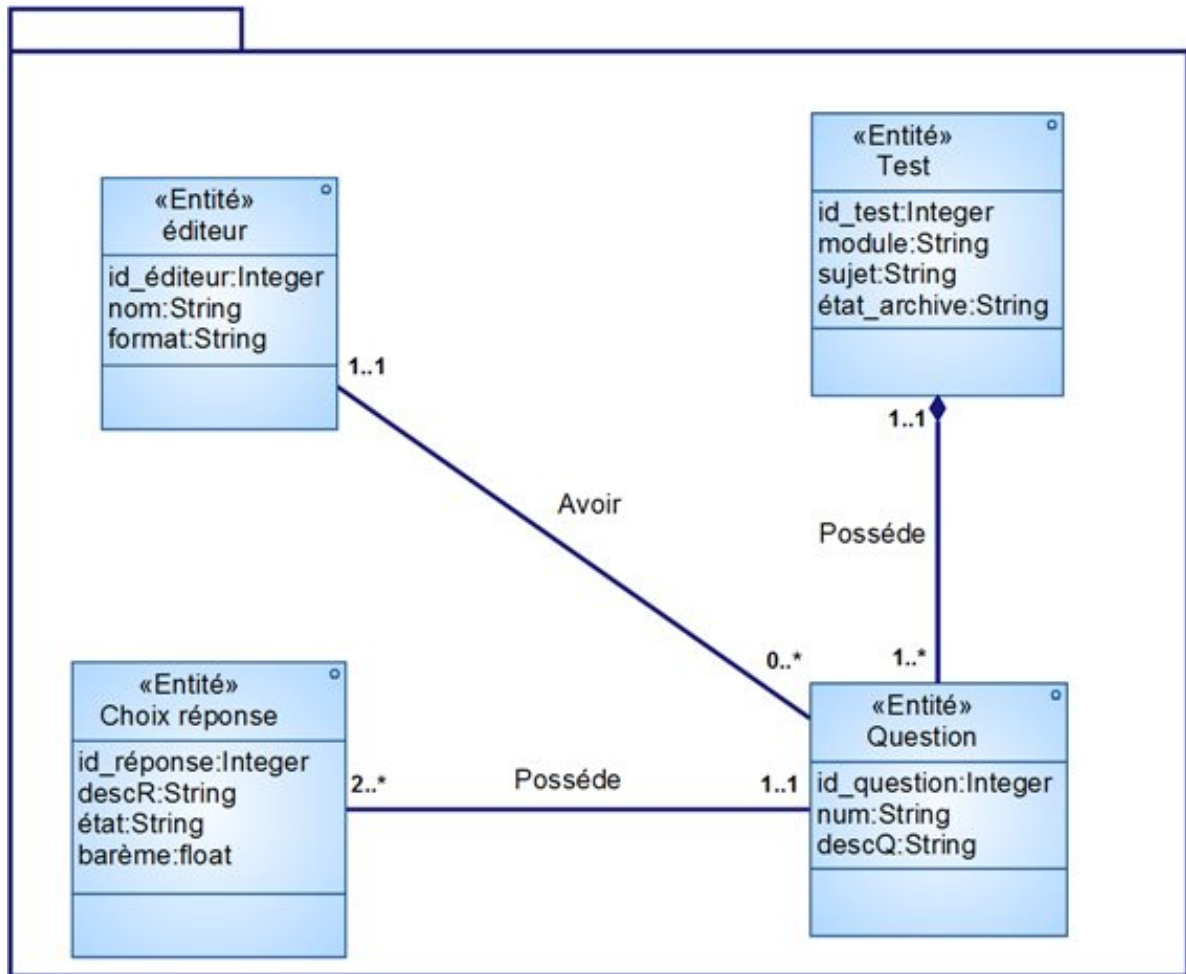


FIGURE 3.9 – Diagramme de classes de la catégorie « Évaluation »

3.3 Développement du modèle dynamique

Le développement du modèle dynamique constitue la troisième activité de l'étape d'analyse. Elle se situe sur la branche gauche du cycle en Y. Il s'agit d'une activité itérative, fortement couplée avec l'activité de modélisation statique. [2]

Le modèle dynamique est basé essentiellement sur deux types de diagrammes :

- ★ Le diagramme de séquence qui illustre les interactions entre les objets.
- ★ Le diagramme d'état de transition qui décrit l'évolution interne des objets d'une classe.

Lors de cette étape, nous allons décrire les différentes interactions et les cycles de vie des objets de notre système par le diagramme d'interaction.

3.3.1 Diagrammes de séquence détaillés

Durant la phase de la capture des besoins nous avons vu qu'un cas d'utilisation représenté par un diagramme de séquence décrit un ensemble des scénarios mais en considérant le système comme une boîte noire, alors pour ouvrir cette boîte le diagramme de séquence détaillé est élaboré. ce diagramme permet de remplacer le système par une collaboration d'objets dans chaque scénario.

Les figures suivantes représentent diagramme de séquence détaillé pour chaque cas :

- Cas d'utilisation : « Gérer test et corrigé type »

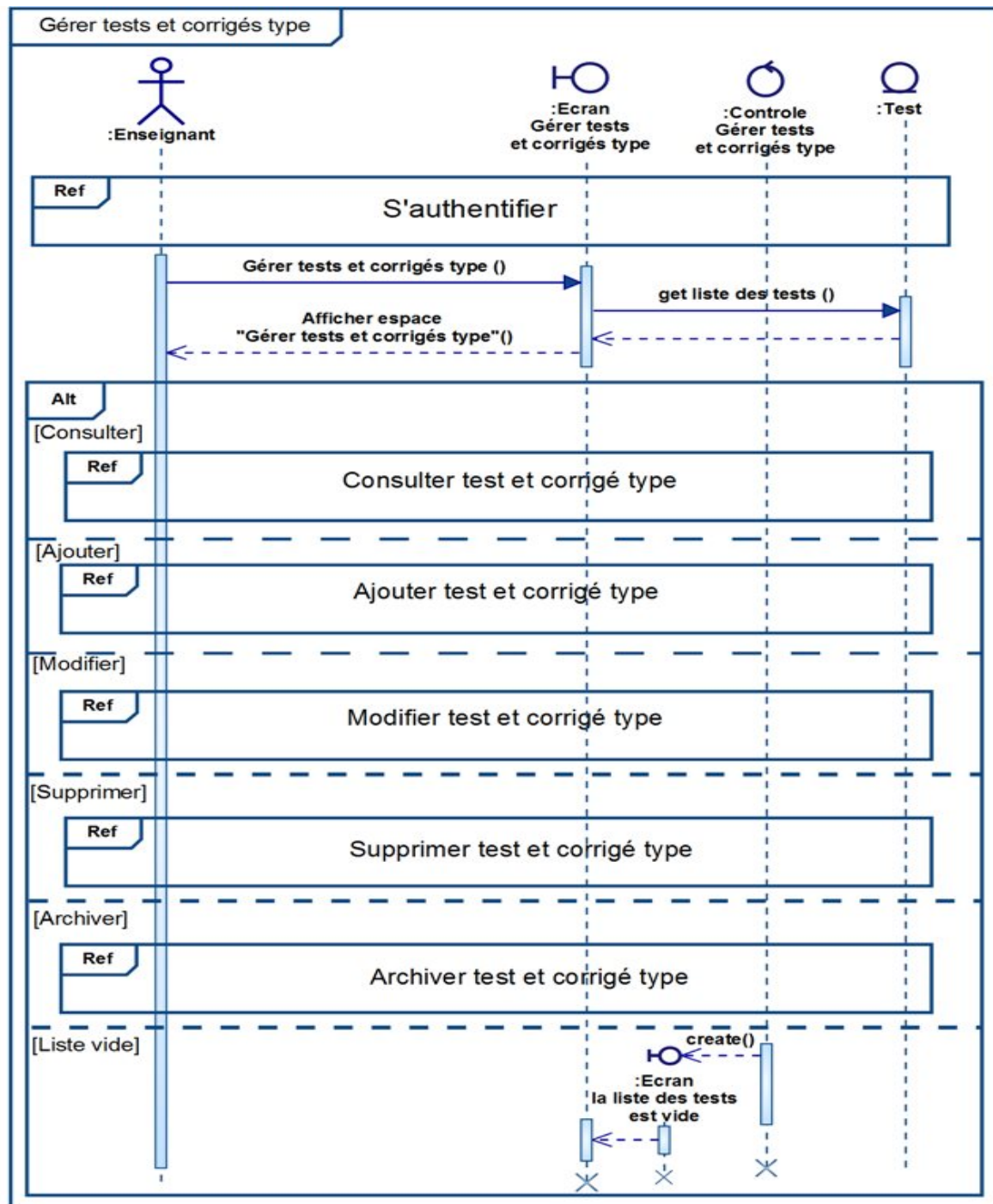


FIGURE 3.10 – Diagramme de séquence détaillé du cas Gérer test et corrigé type

• Cas d'utilisation : « Consulter test et corrigé type »

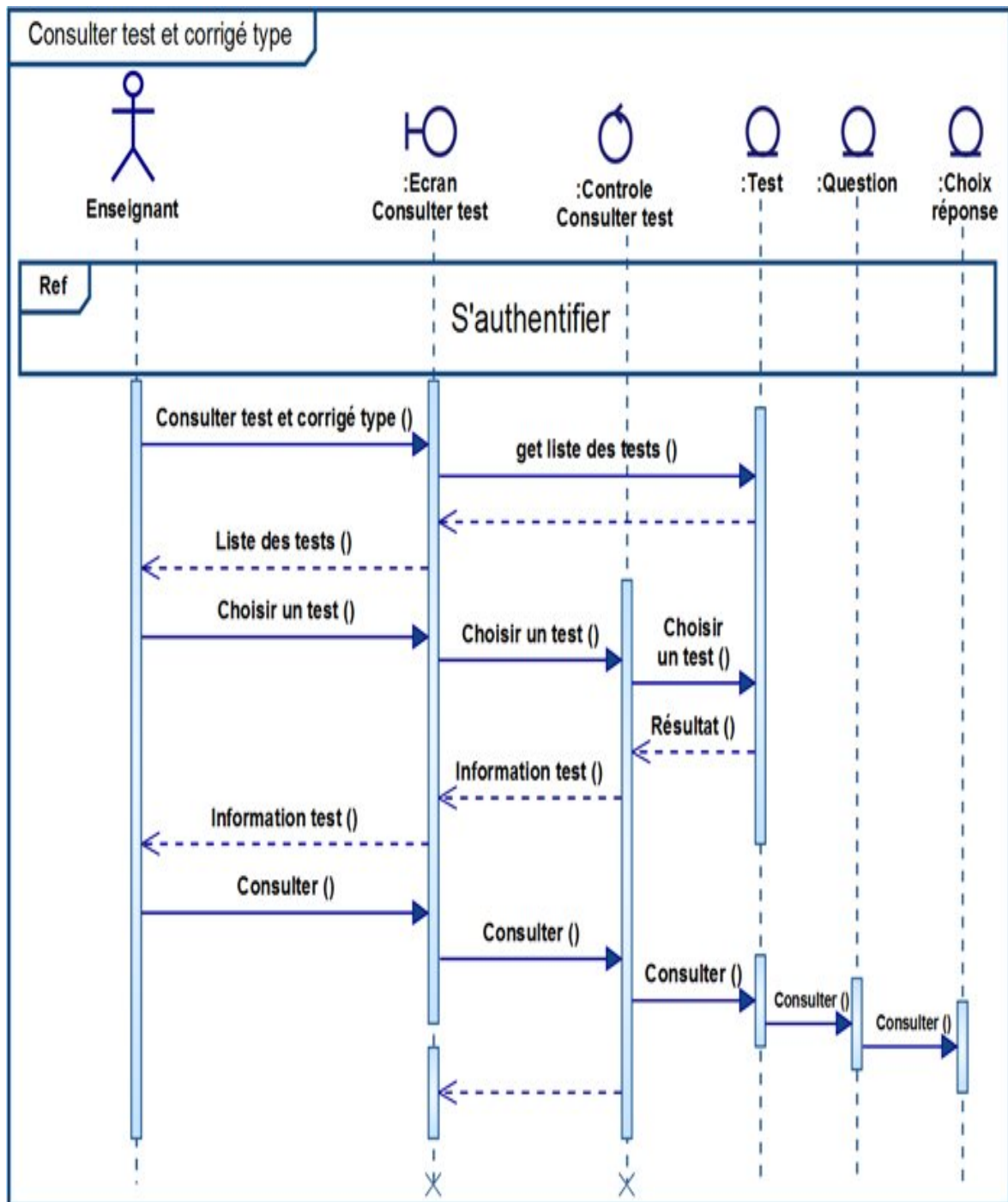


FIGURE 3.11 – Diagramme de séquence détaillé du cas Consulter test et corrigé type

• Cas d'utilisation : « Ajouter test et corrigé type »

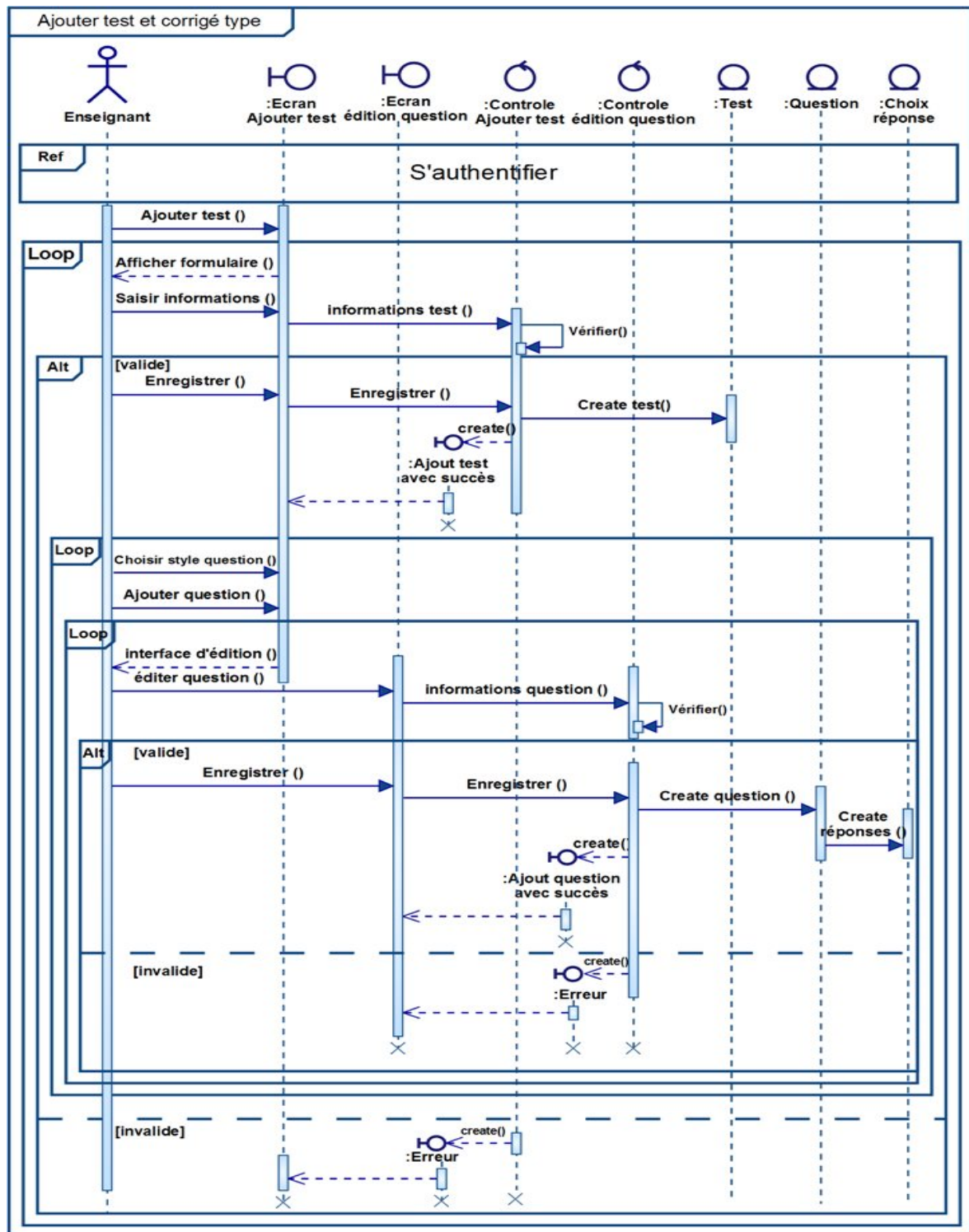


FIGURE 3.12 – Diagramme de séquence détaillé du cas Ajouter test et corrigé type

• Cas d'utilisation : « Modifier test et corrigé type »

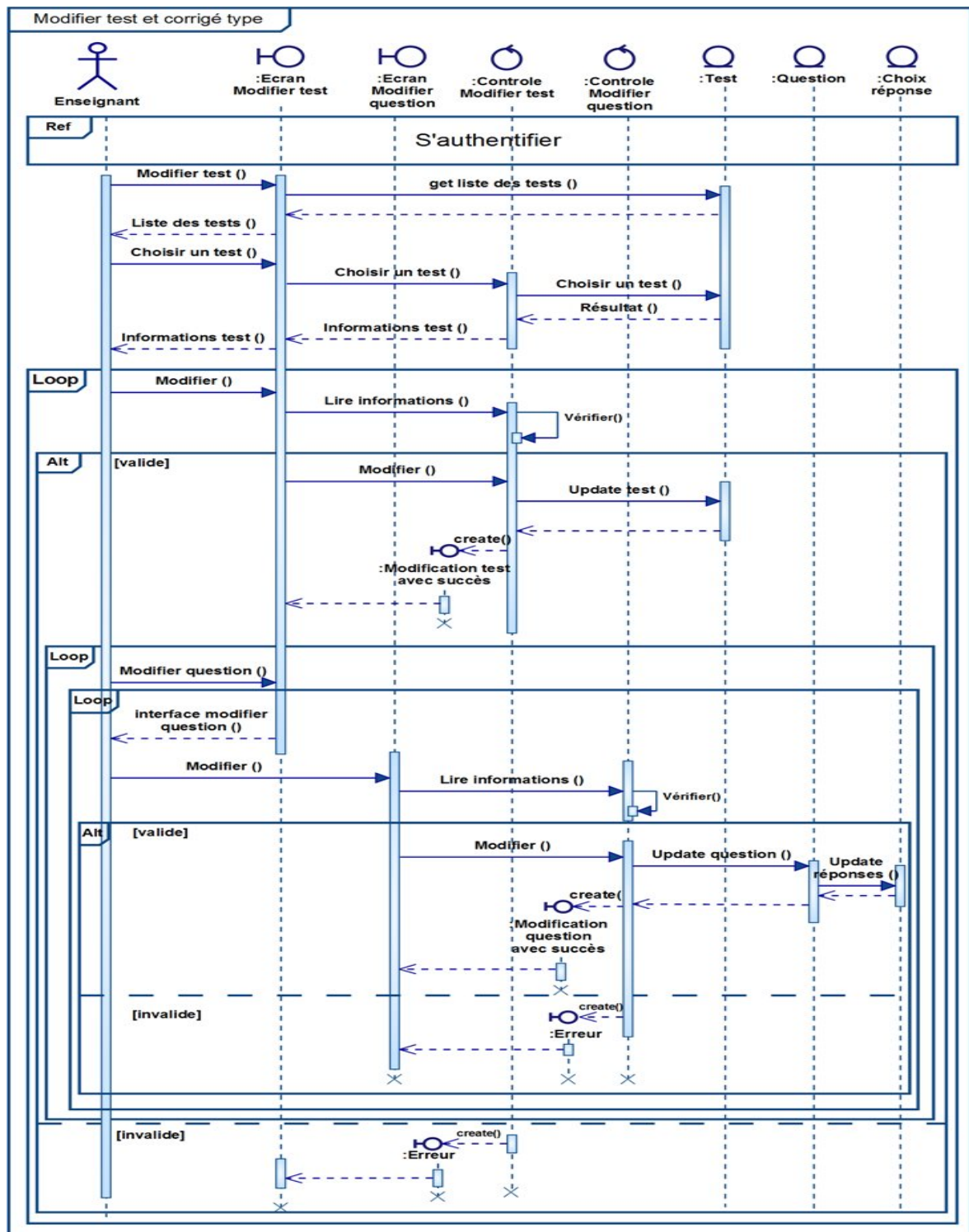


FIGURE 3.13 – Diagramme de séquence détaillé du cas Modifier test et corrigé type

• Cas d'utilisation : « Supprimer test et corrigé type »

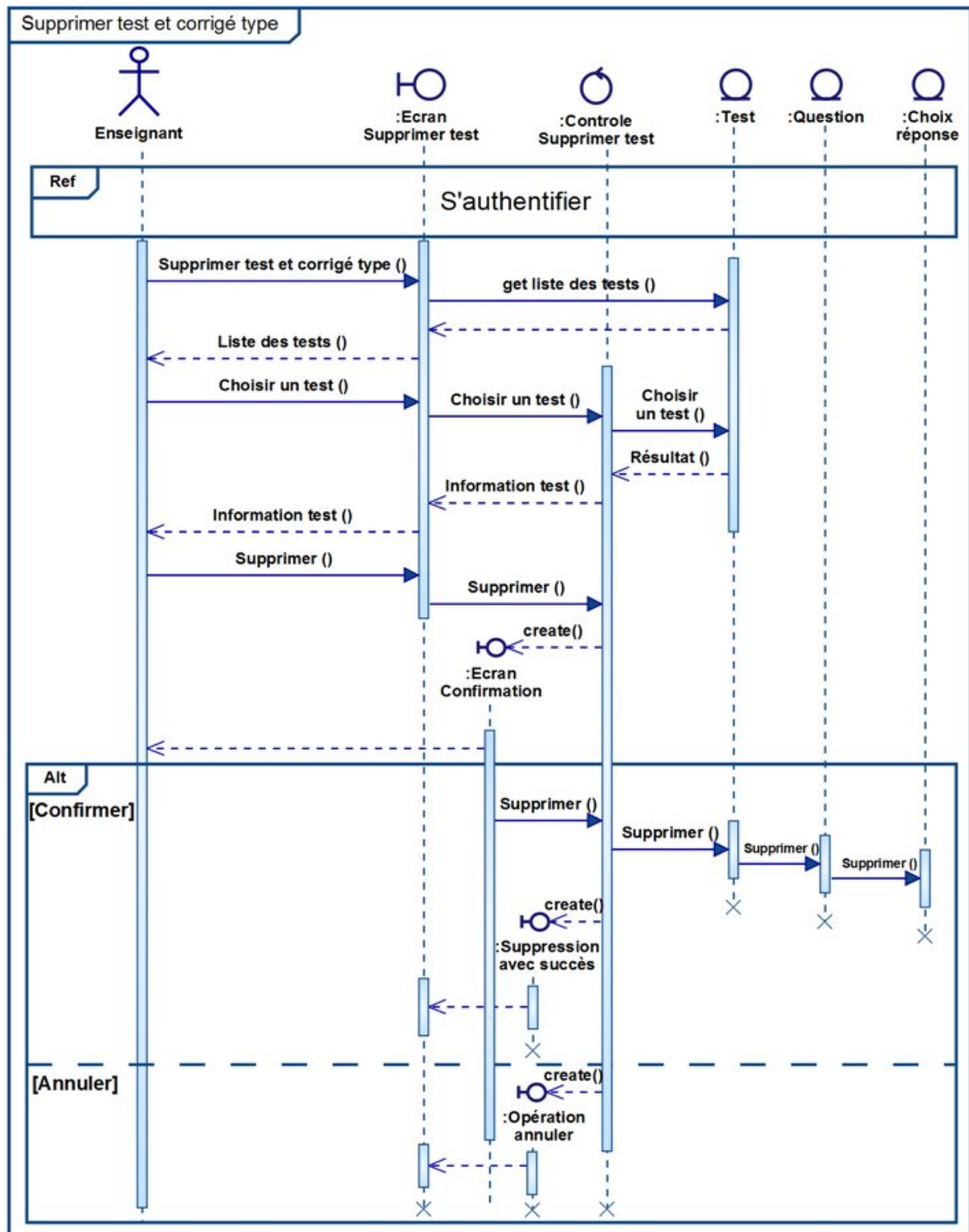


FIGURE 3.14 – Diagramme de séquence détaillé du cas Supprimer test et corrigé type

• Cas d'utilisation : « Archivé test et corrigé type »

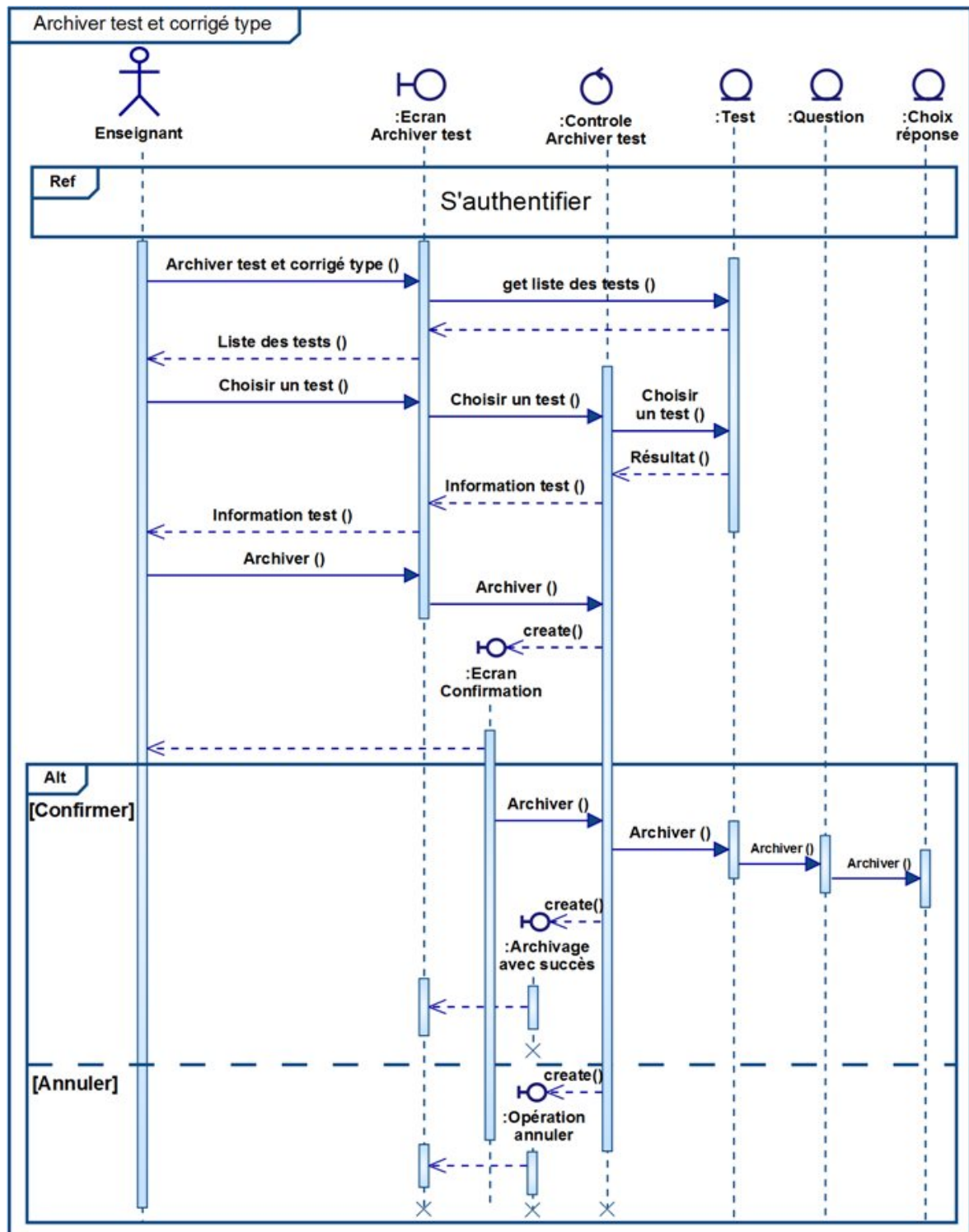


FIGURE 3.15 – Diagramme de séquence détaillé du cas Archivé test et corrigé type

• Cas d'utilisation : « Gérer session »

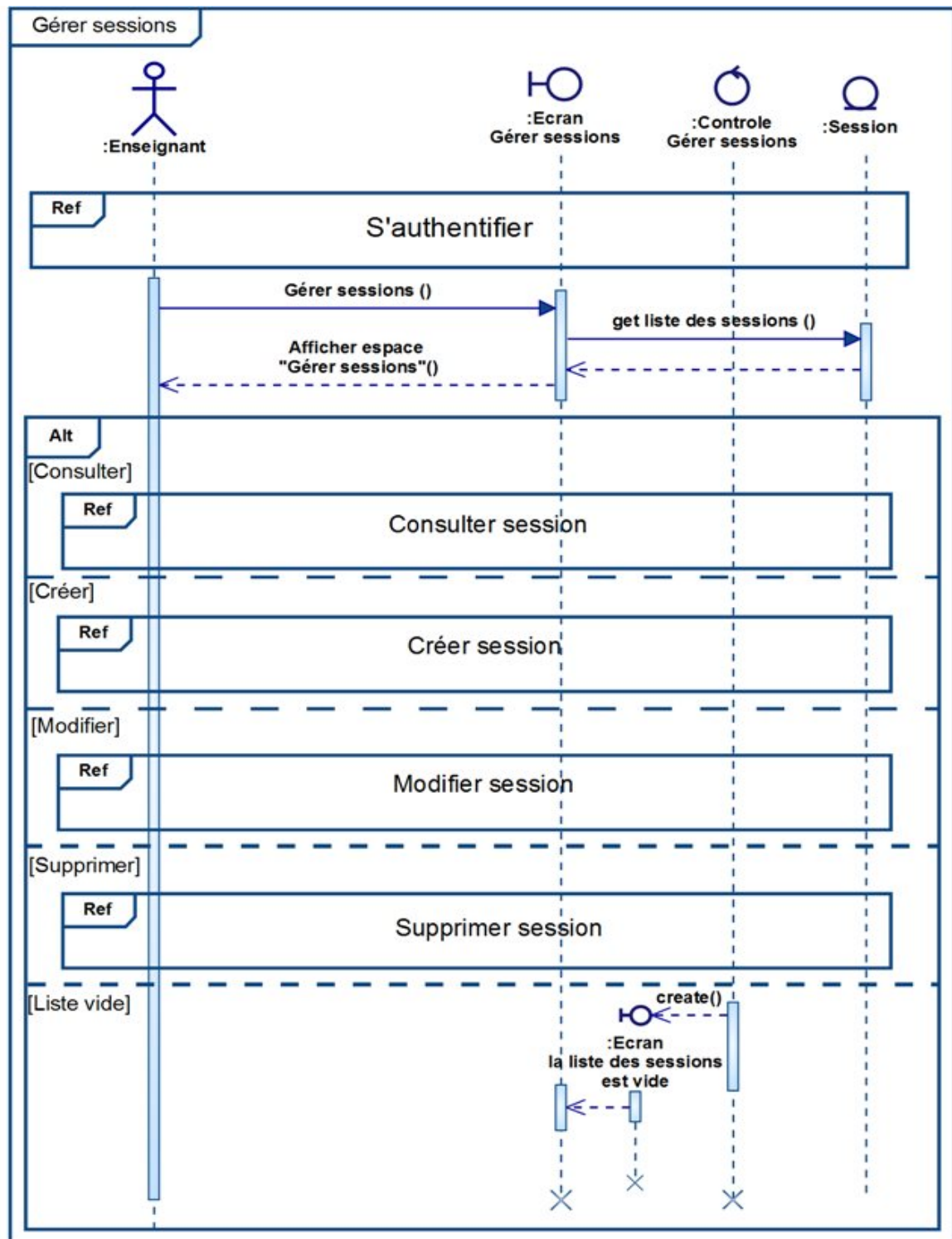


FIGURE 3.16 – Diagramme de séquence détaillé du cas Gérer session

• Cas d'utilisation : « Consulter session »

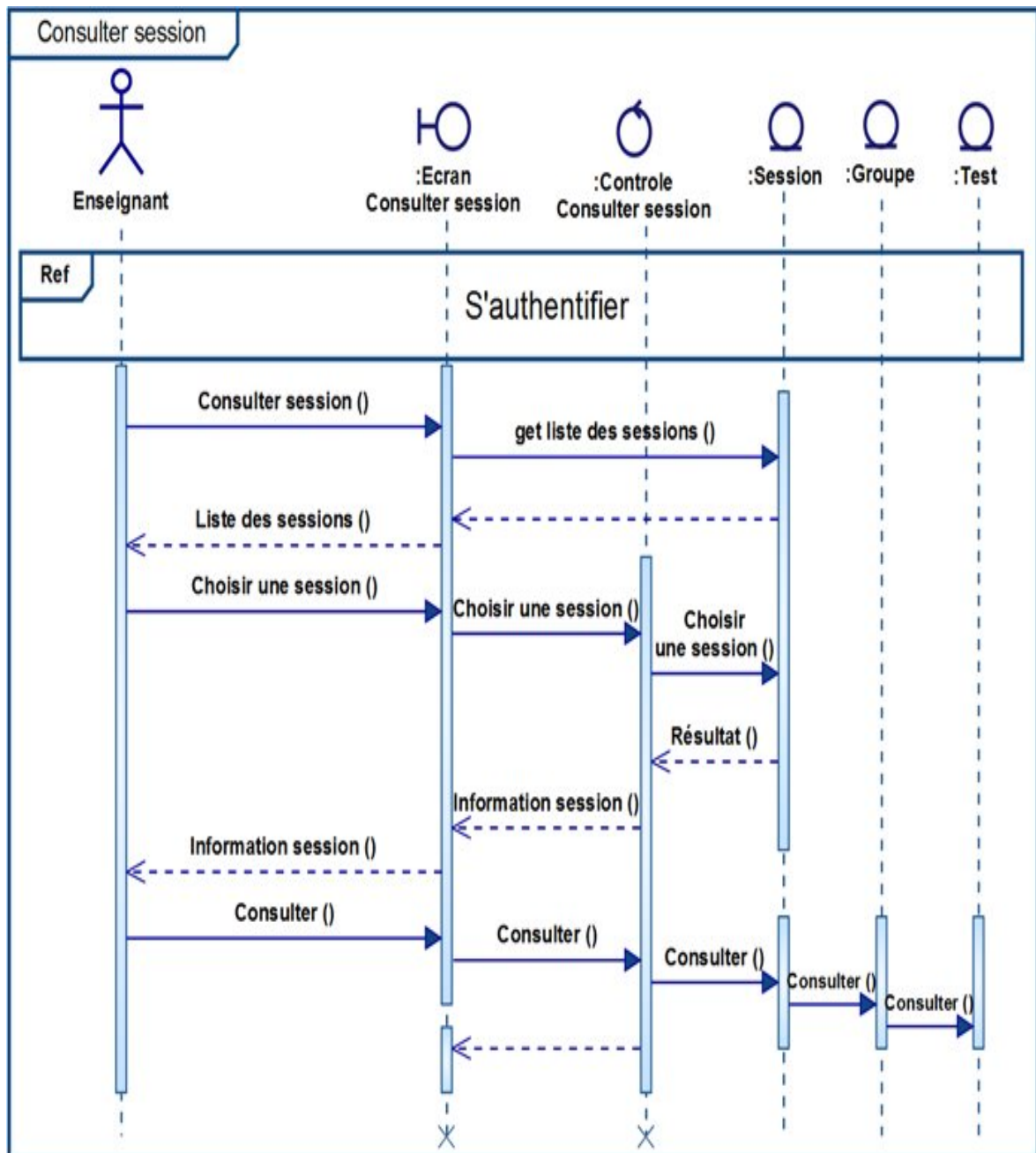


FIGURE 3.17 – Diagramme de séquence détaillé du cas Consulter session

• Cas d'utilisation : « Créer session »

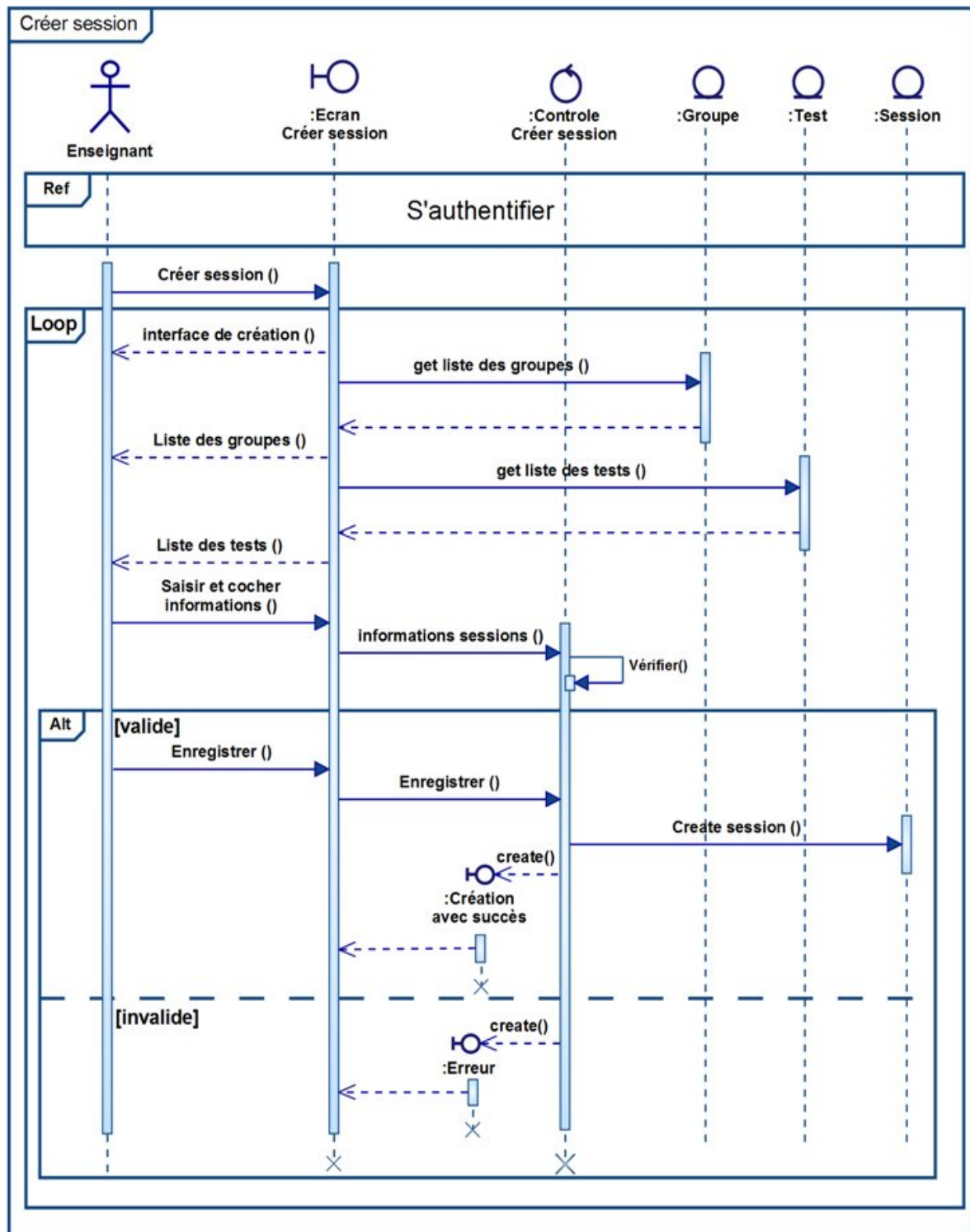


FIGURE 3.18 – Diagramme de séquence détaillé du cas Créer session

• Cas d'utilisation : « Modifier session »

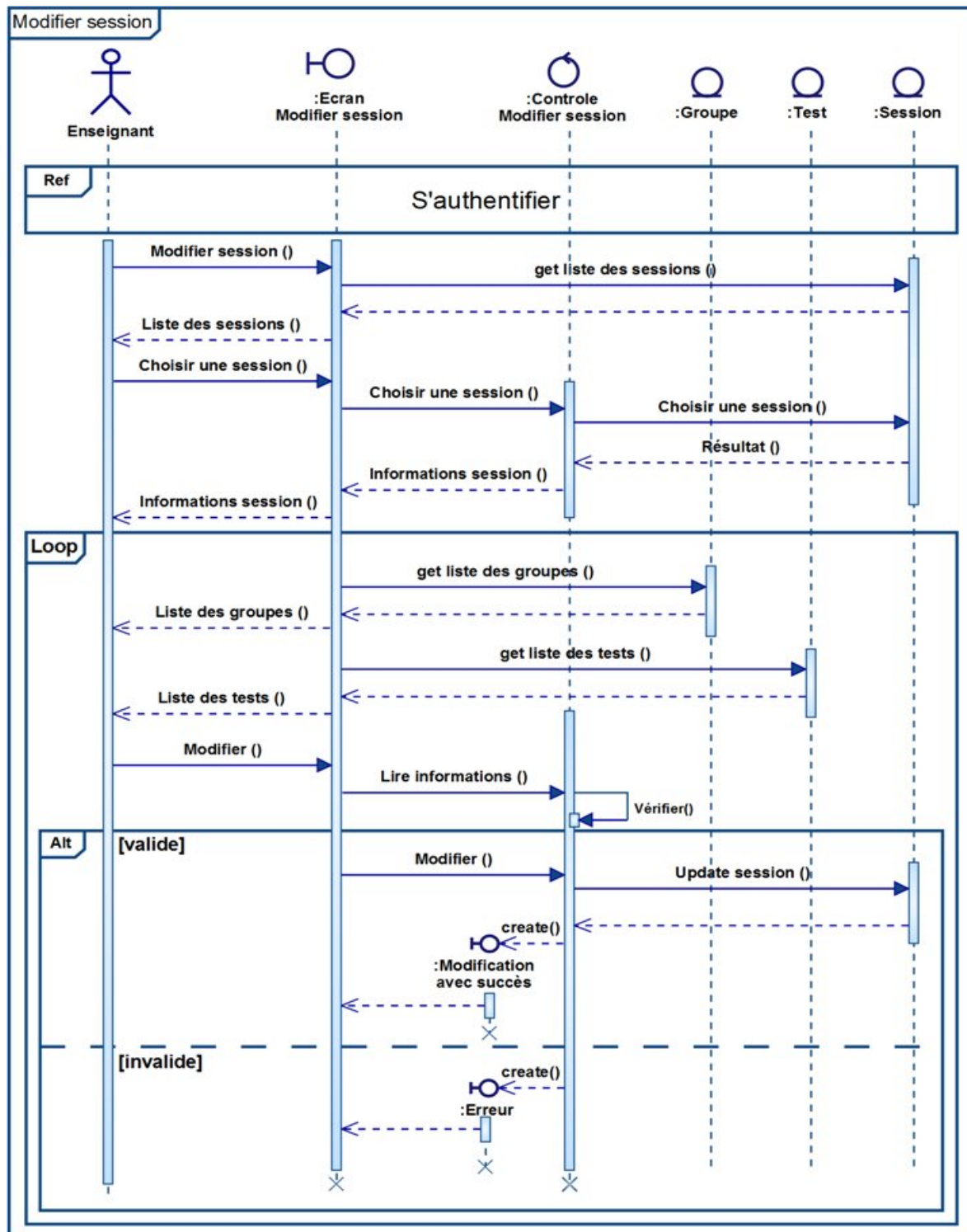


FIGURE 3.19 – Diagramme de séquence détaillé du cas Modifier session

• Cas d'utilisation : « Supprimer session »

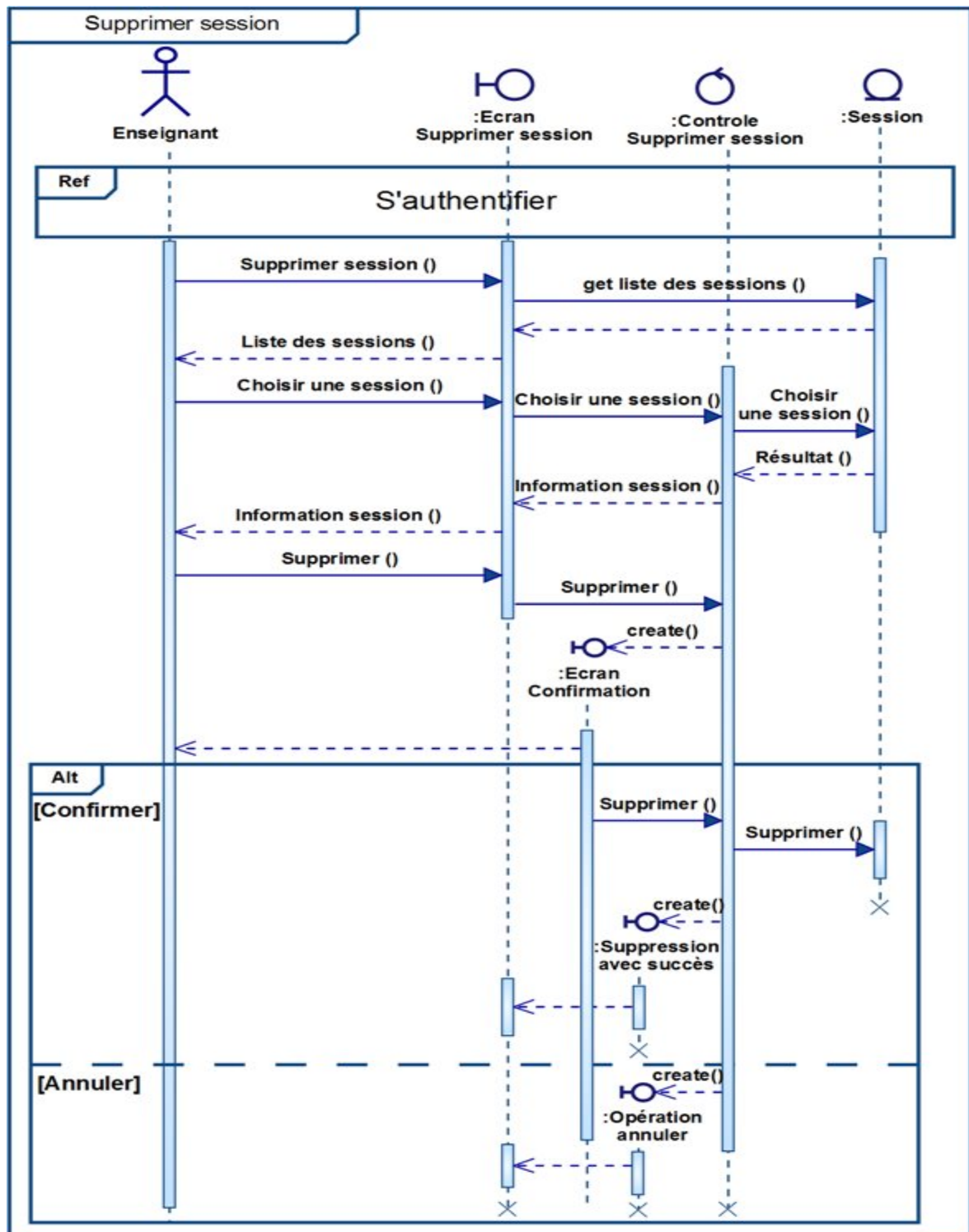


FIGURE 3.20 – Diagramme de séquence détaillé du cas Supprimer session

Conclusion

Au cours de ce chapitre nous avons présenté l'étape d'analyse. Ce stade nous a permis de passer de la structuration fonctionnelle à la structuration objet. La phase qui suit va nous permettre d'entamer la conception et la réalisation.

Conception et Réalisation

Introduction

Après la phase d'analyse étudiée précédemment, dans le présent chapitre nous entamons la phase de conception et de réalisation.

4.1 La conception

La conception est la phase ultime de la modélisation avec UML. Après la modélisation des besoins puis l'organisation de la structure de la solution, la conception vient construire et documenter précisément les classes, les interfaces, et les méthodes qui constituent le codage de la solution. le noyau de cette étape est de savoir transformer le modèle objet en modèle relationnel. [2]

Pour cette étape voici la figure qui représente la situation dans le processus 2TUP :

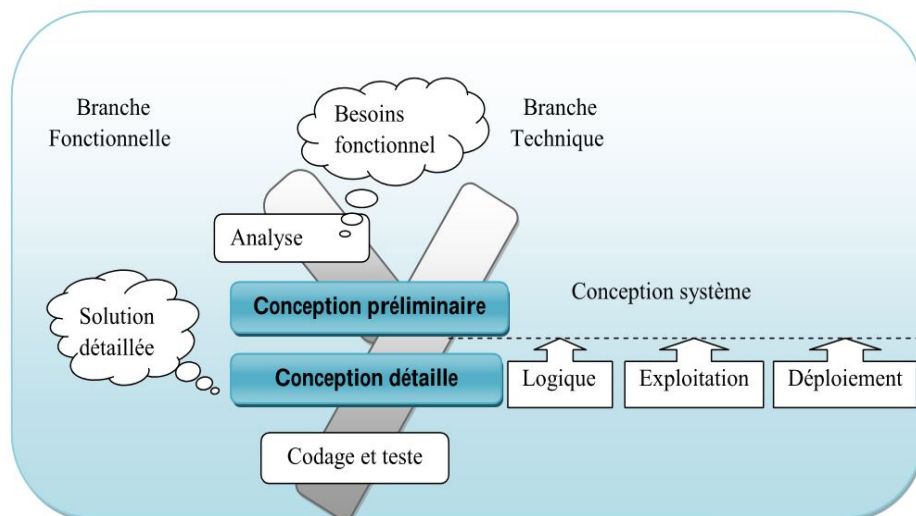


FIGURE 4.1 – Situation de la conception dans le processus 2TUP

4.1.1 Dictionnaire de données

- les classes et les attributs

Classes	Descriptions	Attributs	Types
Utilisateur	Identifiant utilisateur Nom utilisateur Mot de passe Nom utilisateur Prénom utilisateur Email utilisateur	id_utilisateur nomUtil motPass nom prénom email	int (11) varchar(20) varchar(20) varchar(20) varchar(20) varchar(40)
Enseignant	Identifiant enseignant Téléphone enseignant Spécialité enseignante	id_enseignant tél spécialité	int (11) varchar(20) varchar(20)
Agent de MAJ	Identifiant agent Téléphone agent	id_agent tél	int (11) varchar(20)
Etudiant	Identifiant étudiant Code étudiant	id_étudiant code	int (11) varchar(20)
Groupe	Identifiant groupe spécialité du groupe Niveau du groupe Numéro du groupe	id_groupe spécialité niveau groupe	int (11) varchar(20) varchar(20) int (2)
Test	Identifiant test Module du test Sujet du test L'état du test para port à l'archive	id_test module sujet état_archive	int (11) varchar(30) varchar(50) varchar(10)
Question	Identifiant question Numéro de question Description de la question	id_question num descQ	int (11) varchar(10) varchar(150)
Choix réponse	Identifiant réponse Description de la réponse L'état de la réponse Barème de la réponse	id_réponse descR état barème	int (11) varchar(150) varchar(10) float(5)
Session	Identifiant session Nom session Temps session L'état de la session	id_session nom temps état	int (11) varchar(20) int (10) varchar(10)
Éditeur	Identifiant éditeur Nom éditeur Format éditeur	id_éditeur nom format	int (11) varchar(20) varchar(20)
Note	Note de l'étudiant	note	float (5)

TABLE 4.1: Dictionnaire de données

4.1.2 Diagramme de classe détaillé

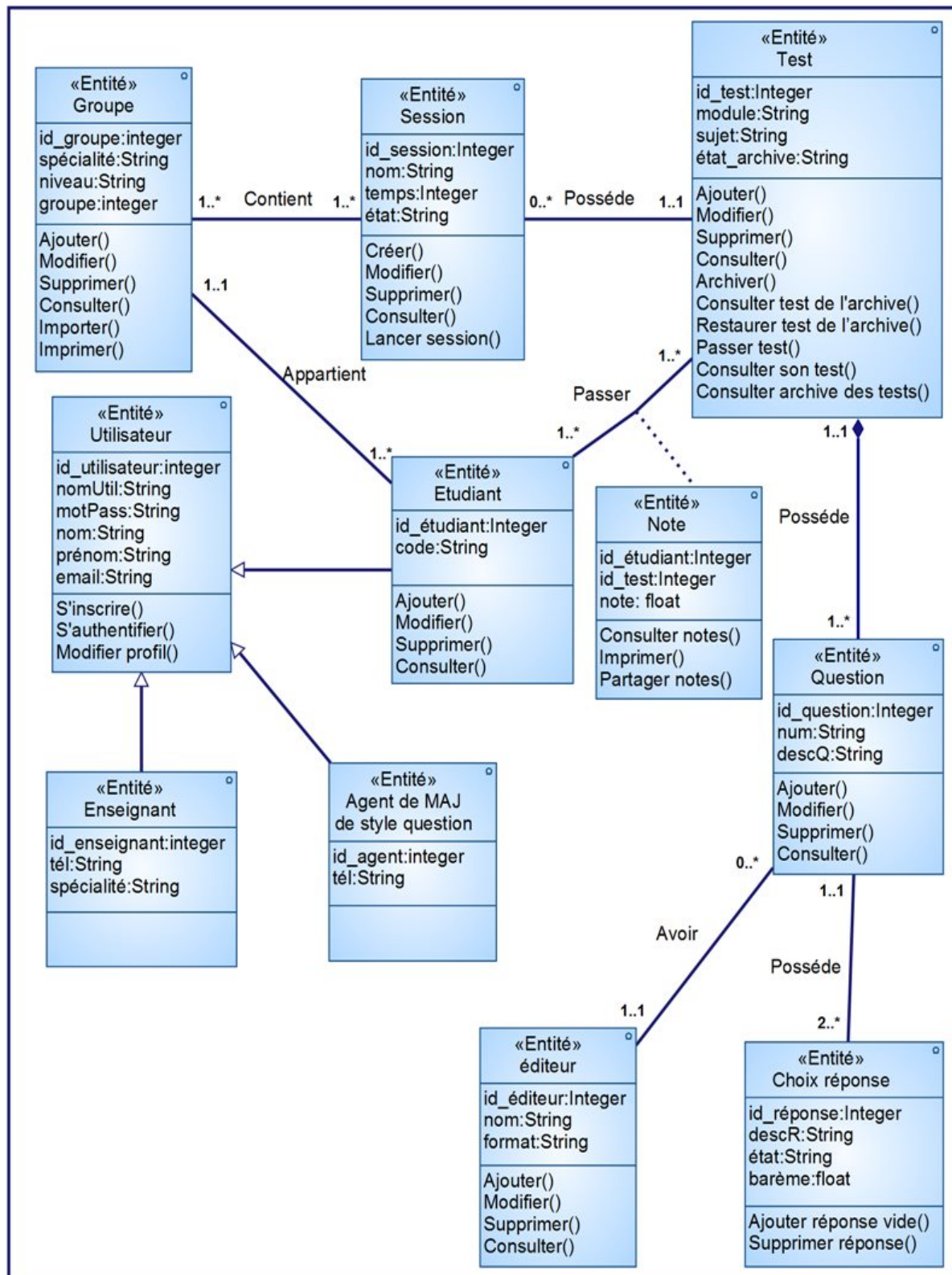


FIGURE 4.2 – Diagramme de classe détaillé

4.1.3 Le modèle relationnel

Un attribut : est un identifiant (un nom) décrivant une information stockée dans une base.

Le domaine d'un attribut : est l'ensemble, fini ou infini, de ses valeurs possibles.

Une relation : est un sous-ensemble du produit cartésien de n domaines d'attributs ($n > 0$).

Une relation : est représentée sous la forme d'une table à deux dimensions dans laquelle les n attributs correspondent aux titres des n colonnes.

Un schéma de relation : précise le nom de la relation ainsi que la liste des attributs avec leurs domaines.

Le degré : d'une relation est son nombre d'attributs.

Une occurrence : est un élément de l'ensemble figuré par une relation. Autrement dit, une occurrence est une ligne de la table qui représente la relation.

La cardinalité : d'une relation est son nombre d'occurrences.

Clé : on distingue deux types de clés :

clé primaire : ensemble d'attributs dont les valeurs permettent de distinguer les n -uplets les uns des autres (notion d'identifiant).

Clé étrangère : attribut qui est clé primaire d'une autre entité.[4]

4.1.4 Le passage de modèle objet au modèle relationnel

L'utilisation d'un SGBDR impose un enchaînement de représentation entre la structure des classes et la structure des données relationnelles, les deux structures ayant des analogies, les équivalences exprimées au tableau sont utilisées pour en réaliser le rapprochement. Une classe définit une structure de données à laquelle souscrivent des instances ; elle correspond donc à une table du modèle relationnel :

chaque attribut donne lieu à une colonne, chaque instance stocke les données dans une ligne et son ID sert de clé primaire.[5]

Modèle objet	Modèle relationnel
Classe	Table
Attribut de type simple	Colonne
Attribut de type composé	Colonne ou clé étrangère
Instance	T_uplet
ID	Clé primaire
Association	Clé étrangère
Héritage	Clé primaire identique sur plusieurs tables

TABLE 4.2: Équivalence entre les concepts objets
et relationnels

Les règles de passage

La réalisation du modèle relationnel peut être faite à partir de la description conceptuelle. Ce passage de modèle doit respecter un certain nombre de règles.

- **Transformation des classes :**

Chaque classe du diagramme UML devient une relation, il faut choisir un attribut de la classe pouvant jouer le rôle de clé.

- **Transformation des associations :**

Les règles de transformation des associations dépendent des multiplicités maximales des associations. Nous distinguons trois familles d'associations :

- **Association 1..* :**

Il faut ajouter un attribut de type clé étrangère dans la relation fils de l'association. L'attribut porte le nom de la clé primaire de la relation père de l'association.

- **Association *.* et classes-associations :**

La classe-association devient une relation. La clé primaire de cette relation est la concaténation des identifiants des classes connectées à l'association.

- **Association 1-1**

Il faut ajouter un attribut de type clé étrangère dans la relation dérivée de la classe ayant la multiplicité minimale égale à un. L'attribut porte le nom de la clé primaire de la relation dérivée de la classe connectée à l'association.

Si les deux multiplicités minimales sont à un, il est préférable de fusionner les deux classes en une seule.[6]

Les relations de la base de données

En se basant sur les règles ci-dessus, nous avons converti les classes entités et leurs associations, à des tables dans la base données.

NB : pour la notation, nous avons choisi de mettre en gras les clés primaires et de mettre # au début de chaque clé étrangère.

Les tables générées sont :

-**Enseignant** (**id_enseignant** , nom, prénom,email ,tél ,spécialité ,nomUtil ,motPass) ;

-**Agent** (**id_agent**,nom ,prénom ,email ,tél ,nomUtil ,motPass) ;

-**Etudiant** (**id_étudiant**,#id_groupe ,code, nom, prénom,email ,nomUtil ,motPass) ;

-**Groupe** (**id_groupe**,spécialité ,niveau ,groupe) ;

-**Test** (**id_test**,module,sujet ,état_archive) ;

-**Question** (**id_question**,#id_test ,#id_éditeur,num ,descQ) ;

-**Choix réponse** (**id_réponse**,#id_question,descR ,état,barème) ;

-**Session** (**id_session**,#id_test,#id_groupe,nom ,temps ,état) ;

-**Editeur** (**id_éditeur** ,nom ,format) ;

-**Note** (**id_test**,**id_étudiant** ,note) ;

TABLE 4.3: Les relations de la base de données de notre système

4.2 La réalisation

Après avoir réalisé la conception appropriée à notre application, nous allons dans cette partie décrire le processus de réalisation de notre application. Ceci en spécifiant l'environnement de développement, l'implémentation de la base de données et un aperçu sur les interfaces de notre application.

4.2.1 Environnement de développement

Afin de développer des applications sous Android un ensemble d'outils est nécessaire :

Le langage java

Java est un langage de programmation à usage général, évolué et orienté objet dont la syntaxe est proche du C. Ses caractéristiques ainsi que la richesse de son écosystème et de sa communauté lui ont permis d'être très largement utilisé pour le développement d'applications de types très disparates. Java est notamment largement utilisée pour le développement d'applications d'entreprises et mobiles. [7]



FIGURE 4.3 – Logo du langage java

Le langage XML

Le langage XML est un format général de documents orienté texte. Il s'est imposé comme un standard incontournable de l'informatique. Il est aussi bien utilisé pour le stockage de documents que pour la transmission de données entre applications. La force de XML réside dans sa capacité à pouvoir décrire n'importe quel domaine de données grâce à son extensibilité. Il va permettre de structurer et poser le vocabulaire et la syntaxe des données qu'il va contenir.[8]

Le SDK

SDK signifie Software Développement Kit, traduit en français par Kit de développement, c'est un ensemble d'outils d'aide à la programmation pour concevoir des logiciels, jeux, applications mobiles... etc, pour un terminal et/ou un système d'exploitation spécifique. [9]

L'IDE Android studio

Android Studio est un environnement de développement pour des applications android. Il permet principalement d'éditer des fichiers JAVA et les fichiers de configuration d'une application android. Il propose entre autres des outils pour gérer le développement d'applications multilingues et permet de visualiser la mise en page des écrans sur des écrans de résolution simultanément. [9]



FIGURE 4.4 – Logo de l'IDE Android Studio

4.2.2 Les bases de données

Pour implémenter notre base des données, nous avons utilisé l'environnement de création de base de données PHPMyAdmin et le système de gestion base de données MySQL.

Wampserver

Wampserver est une plateforme de développement Web sous Windows. Il permet de développer des applications Web dynamiques à l'aide du serveur Apache2, du langage de script PHP et d'une base de données MySQL. Il possède également PHPMyAdmin pour gérer plus facilement les bases de données.[10]



FIGURE 4.5 – Logo du WampServer

Langage de manipulation de bases de données MySQL

MySQL est un système de Gestion de Bases de Données (SGBD) client/serveur open source qui gère, stocke et manipule une grande quantité de données en les organisant sous forme de tables, il prend aussi en charge les fonctionnalités de protection et de sécurité et fournit un ensemble d'interfaces de programmation (dont une avec PHP) facilitant l'accès aux données. ce système repose sur Le langage SQL qui permet de décrire des tables et d'interagir avec elle (insertion de données et demande de données).[11]



FIGURE 4.6 – Logo de Langage de Manipulation de Bases de Données MySQL

PHP

PHP est une plate-forme composée d'un langage de programmation très complet et de nombreux outils pour le développement. Elle s'adapte très rapidement aux technologies émergentes et se voit de plus en plus utilisée dans des développements web dynamique professionnel et Open Source. Voici quelques-unes de ses caractéristiques principales :

- très bon compromis entre fiabilité et rapidité d'exécution ;
- Langage procédural et un langage orienté objet ;

- Outil très complet, dotait de nombreuses fonctionnalités, extensions et bibliothèques. [12]

Notepad++

Notepad++ est un éditeur de texte très léger, très puissant et libre (licence GPL). Il est parfait pour programmer avec des langages ne nécessitant pas d'environnement de développement (Html, Css, Javascript, Php) ou en ayant un peu pratique (python, processing), ou pour du traitement de données.[13]

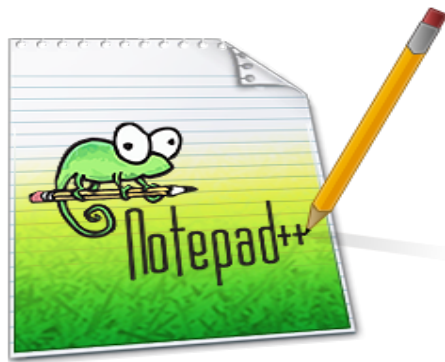


FIGURE 4.7 – Logo d'éditeur de texte Notepad++

4.2.3 Présentation de l'application réalisée

Pendant ce travail nous avons développé trois applications où chaque application relie principalement à son acteur. donc cette partie du mémoire, nous aide à présenter les principales fonctionnalités de chaque application qui fera réaliser par la description de quelques interfaces relatives à ces derniers.

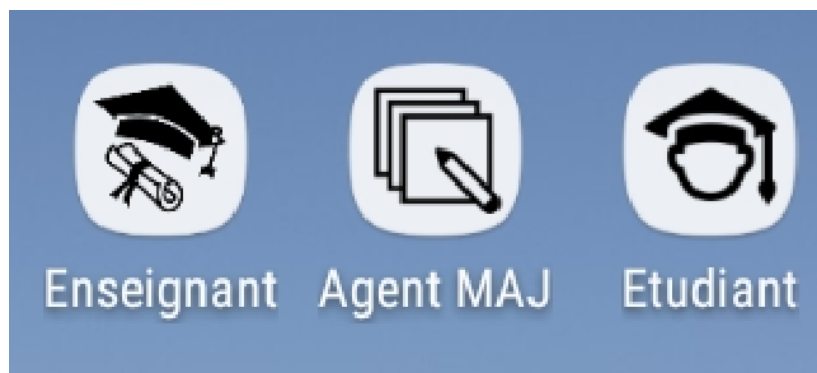


FIGURE 4.8 – Logo des différentes applications réalisées

★ Côté Enseignant

Espace d'inscription

Nom

Prénom

Email

Nom Utilisateur

Mot de passe

Confirmer mot de passe

Téléphone

Spécialité

Inscrire

FIGURE 4.9 – Espace d'inscription

Contrôle Continu

Nom Utilisateur

Mot de passe

Connecter

mot de passe oublié ?

inscrire

FIGURE 4.10 – Authentification

Contrôle Continu

Bienvenu au système de Gestion du contrôle continu

Espace enseignant

Gérer groupes enseignés

Gérer tests et corrigés type

Gérer sessions

Consulter notes

FIGURE 4.11 – Accueil enseignant

Modifier Profil

Enseignant

.....

.....

Modifier

Espace modifier profil

FIGURE 4.12 – Modifier profil



FIGURE 4.13 – Gérer la liste des groupes enseignés



FIGURE 4.14 – Ajouter groupe

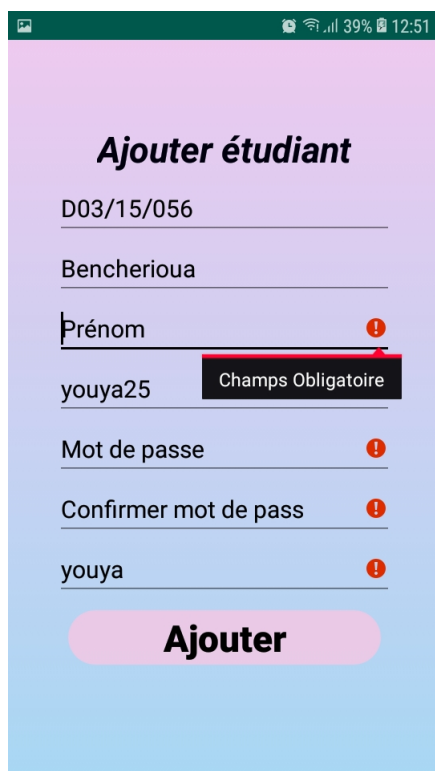


FIGURE 4.15 – Ajouter étudiant cas d'erreur 1



FIGURE 4.16 – Ajouter étudiant cas d'erreur 2

Ajouter étudiant

D03/15/056

Bencherioua

Prénom

youya25

Mot de passe

Confirmer mot de pass

youya

le signe @ est obligatoire dans le champ email

FIGURE 4.17 – Ajouter étudiant cas d'erreur 3

Ajouter étudiant

D03/15/056

Bencherioua

Aya

youya25

.....

youya1995@gmail.com

Ajouter

ajout avec succès

FIGURE 4.18 – Ajouter étudiant "Ajout avec succès"

Ajouter groupe

Informatique

2 éme année

1

Ajouter

Importer la liste des étudiants

Liste des étudiants

Recherche

D03/15/056

Bencherioua

Aya

FIGURE 4.19 – Ajouter groupe "Liste des étudiants"

Modifier étudiant

D03/15/056

Bencherioua

Aya

youya25

123456

youya1995@gmail.com

Modifier

FIGURE 4.20 – Modifier étudiant



FIGURE 4.21 – Consulter étudiant

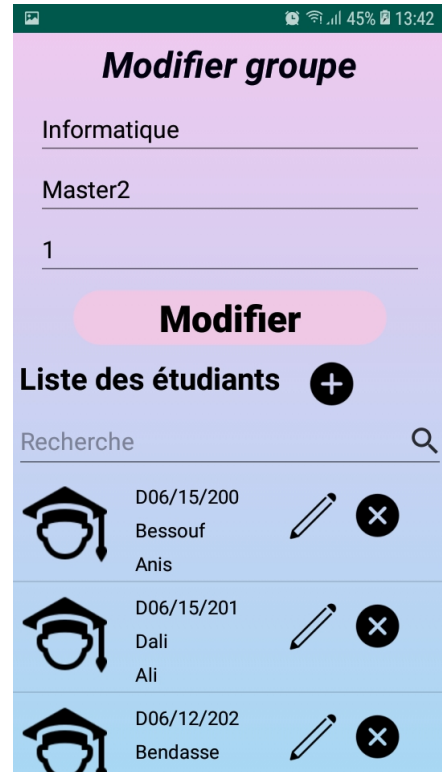


FIGURE 4.22 – Modifier groupe



FIGURE 4.23 – Consulter groupe

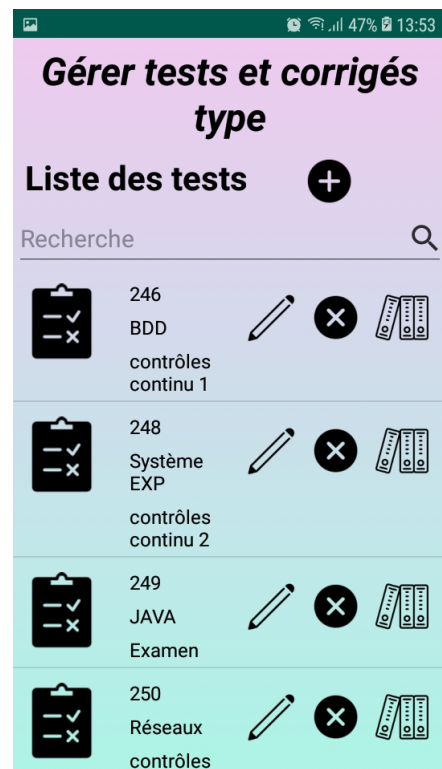


FIGURE 4.24 – Gérer tests et corrigés type



FIGURE 4.25 – Ajouter test



FIGURE 4.26 – Choisir style question

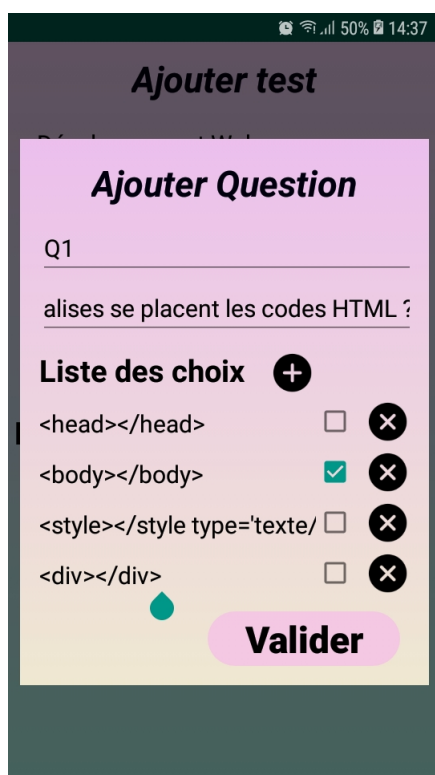


FIGURE 4.27 – Ajouter question

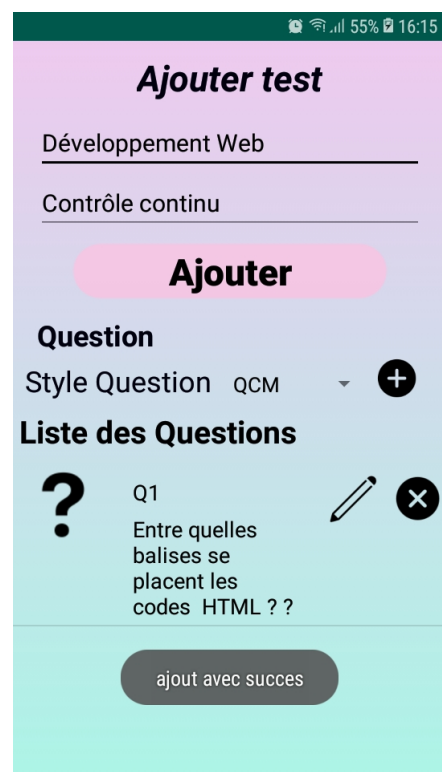


FIGURE 4.28 – Ajouter test "Liste des questions"

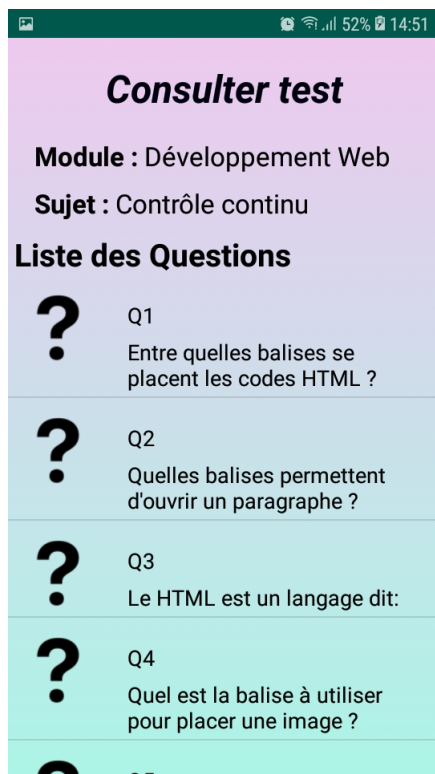


FIGURE 4.29 – Consulter test

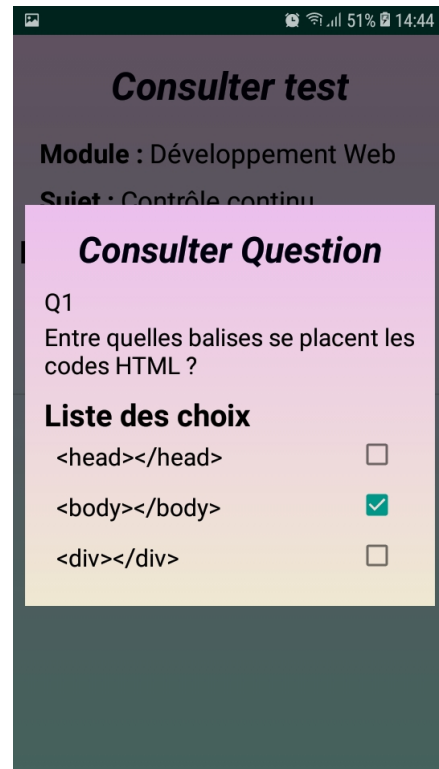


FIGURE 4.30 – Consulter question

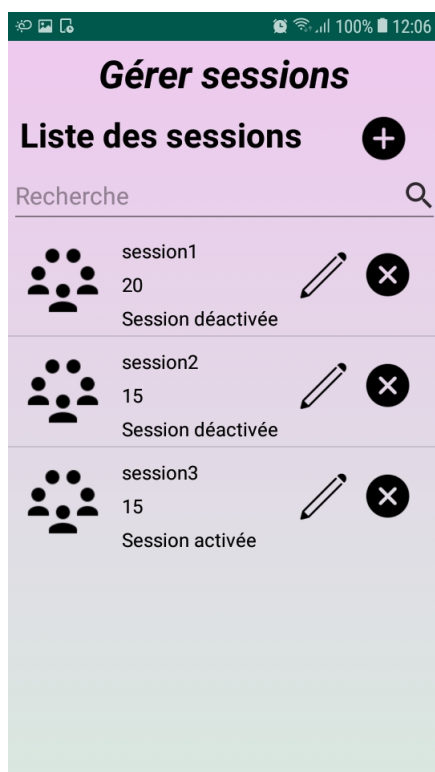


FIGURE 4.31 – Gérer session

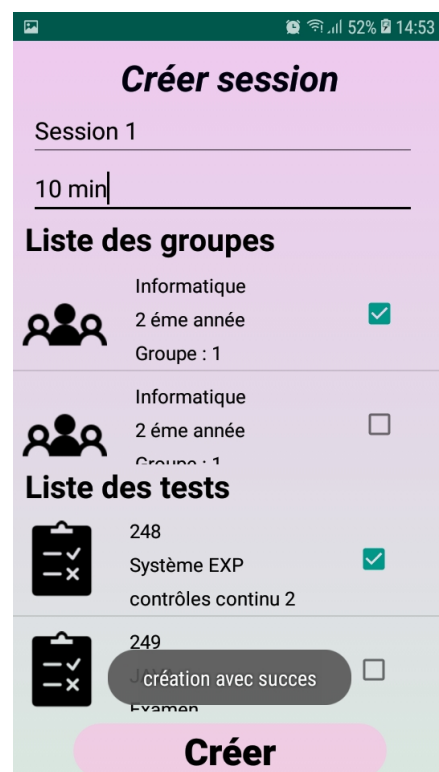


FIGURE 4.32 – Créer session

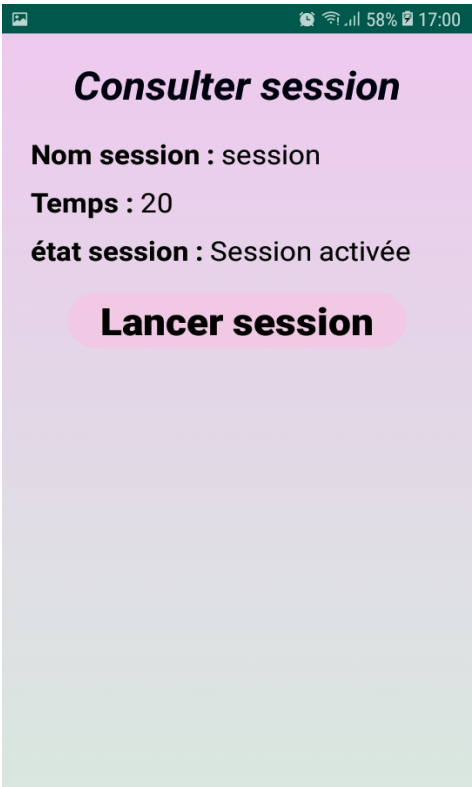
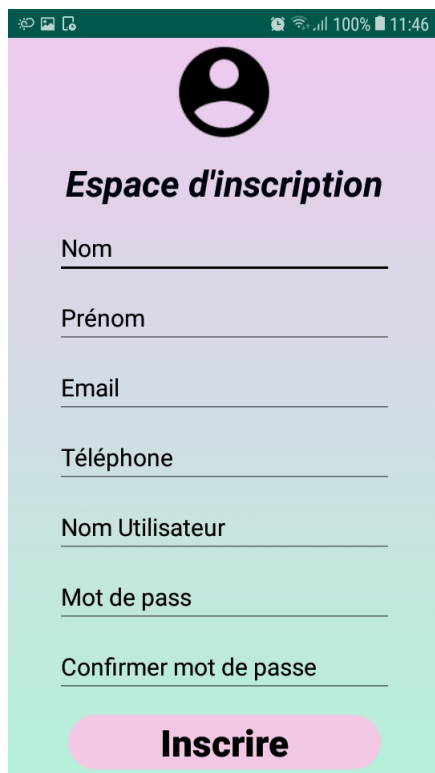


FIGURE 4.33 – Consulter session



FIGURE 4.34 – Gérer archive des tests

★ Coté Agent de MAJ de style question



Espace d'inscription

Nom

Prénom

Email

Téléphone

Nom Utilisateur

Mot de pass

Confirmer mot de passe

Inscrire

FIGURE 4.35 – Espace d'inscription



Contrôle Continu

Agent

.....

Connecter

mot de passe oublié ?

inscrire

FIGURE 4.36 – Authentification



Contrôle Continu

Bienvenu au système de Gestion du contrôle continu

Espace Agent MAJ de style question

Gérer éditeur de style question

succès

FIGURE 4.37 – Accueil Agent de MAJ



Modifier Profil

Agent

.....

.....

Modifier

Espace modifier profil

FIGURE 4.38 – Modifier profil

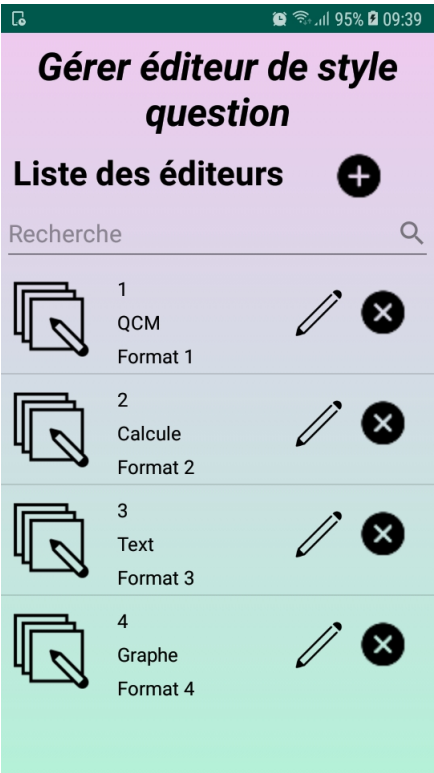


FIGURE 4.39 – Gérer éditeur

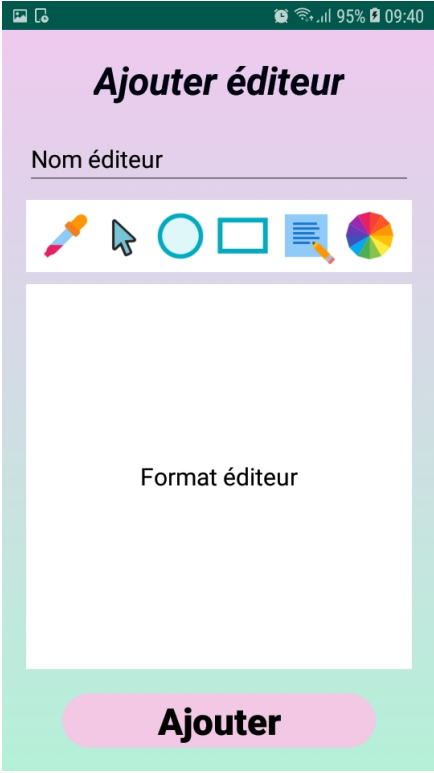


FIGURE 4.40 – Ajouter éditeur

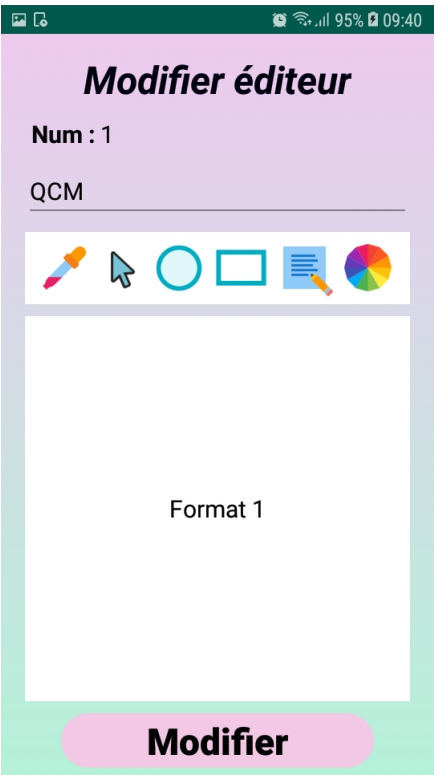


FIGURE 4.41 – Modifier éditeur

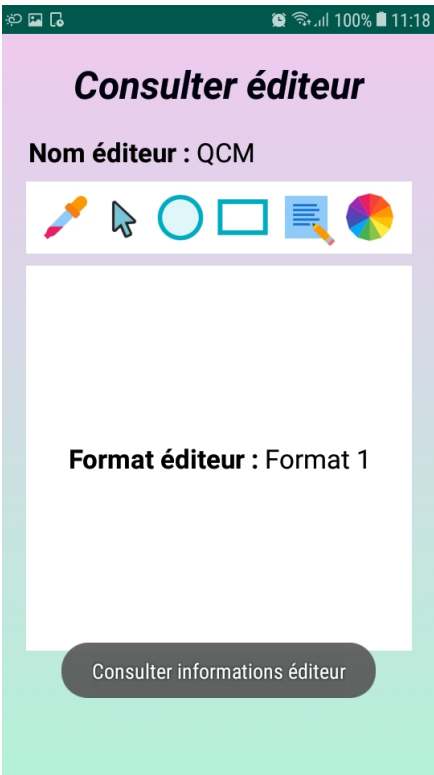


FIGURE 4.42 – Consulter éditeur

★ Côté Étudiant

Espace d'inscription
Choisir ton groupe

Informatique
Master2
Groupe : 2

Formulaire d'inscription

Code

Nom

Prénom

Nom utilisateur

Mot de passe

Confirmer mot de pass

Email

Inscrire

FIGURE 4.43 – Espace d'inscription

Contrôle Continu

Nom Utilisateur

Mot de pass

Connecter
mot de passe oublié ?

inscrire

FIGURE 4.44 – Authentification

Contrôle Continu

Bienvenu à l'application mobile du contrôle continu

Espace étudiant

Passer test

Consulter son test

Consulter archive des tests

FIGURE 4.45 – Accueil Étudiant

Participation test
Liste des sessions

Recherche

session1	20	Session désactivée
session2	15	Session désactivée
session3	15	Session activée

FIGURE 4.46 – Participation test

Passer test

Module : Développement Web

Sujet : Contrôle continu

Temps : 15

Questions

Q1 Entre quelles balises se placent les codes HTML ?

Réponses

<head></head>	☒
<body></body>	☐
	☐
<style></style>	☐

← →

Valider

FIGURE 4.47 – Passer test
« question 1 »

Passer test

Module : Développement Web

Sujet : Contrôle continu

Temps : 15

Questions

Q2 Quelles balises permettent d'ouvrir un paragraphe ?

Réponses

<p>	☐
<prg>	☒
<prg>	☐

← →

Valider

FIGURE 4.48 – Passer test
« question 2 »

Conclusion

Au cours de ce chapitre, et en premier lieu nous avons détaillé les différentes classes du système pour passer à la conception de la base de données, cette dernière étape est primordiale pour l'élaboration du modèle relationnelle qui définit l'ensemble des tables de la base de données de notre système. En second lieu nous avons présenté les aspects pratiques liés à la réalisation de notre projet, à savoir les outils de développement nécessaires. En dernier, nous avons illustré les interfaces que comprend notre application.

Conclusion Générale

Dans le tout de cette étude nous avons abordé le développement d'une application mobile pour la gestion de contrôle continu des étudiants dans le système LMD ; de deux points de vue modélisation et réalisation.

Sur le plan modélisation, nous nous sommes basés sur le processus 2TUP « 2 Track Unified Process ». La conception a été illustrée par les différents diagrammes UML2.0, qui nous ont permis de spécifier et documenter le système, notamment le diagramme des cas d'utilisation permet de capturer et représenter les besoins, le diagramme de classes a permis de représenter l'aspect statique du système. Et enfin, les diagrammes de séquence sont utilisés dans les phases de capture de besoins fonctionnels et d'analyse.

Sur le plan réalisation, nous avons choisi le langage Java sous la plateforme Android avec MYSQL pour la gestion de la base de données. En outre, ce projet nous a permis d'enrichir nos connaissances dans des domaines variés comme : la programmation mobile(Android), UML2.0, 2TUP, ... etc

À travers cette étude nous avons pu mettre en pratique et développer les connaissances que nous avons acquises durant notre cursus universitaire, et en particulier dans les systèmes d'information, les bases de données et le génie logiciel. Cependant nous avons senti une grosse pression de temps. Le temps relativement court ne nous a pas permis de faire tout ce qu'on a espéré faire. Cependant des perspectives d'amélioration de notre application restent envisageables telles que :

- ajouter d'autres styles de questions au test plus que le QCM (calcul, le complément des paragraphes par des mots ou bien par des images, l'ordre des phrases... etc)
- Relier notre application à la plateforme e-learning.
- Et principalement migrer notre base de données, d'une base de données centralisée à une base de données mobile.

Bibliographie

- [1] : **Zeteli Choayb,Bendas Dawoud.**« Conception et réalisation d'une application web pour la gestion des ressources humaines au sein de la direction d'Algérie Télécom ».Mémoire de Master professionnel , Sciences et technologies de l'information et de la communication(STIC).Mila :Centre Universitaire Abdel hafid Boussouf de Mila, 2018, 266 P.

- [2] : **Roques,P.et Vallée,F.**« UML 2 en action de l'analyse des besoins à la conception en java ».France :Eyrolles, 2001, 398 p.

- [3] : **Roques,P.**« Les cahiers du programmeur UML2 modélisé une application Web ».Paris :Eyrolles, 2008, 246 p.

- [4] : **Benabderrahmane,Fatiha.**« Le modèle relationnel » :Centre Universitaire Abdel hafid Boussouf de Mila,Cours, 2016, 28 p.

- [5] : **Guyot,J.** « Conception & réalisation des bases de données :de UML à SQL » :Editions systèmes et information, 2008 ,385 p.

- [6] : **Roques,P.et Vallée,F.**« UML 2 en action ».France :Eyrolles, 2008 ,394 p.

- [7] : **DOUDOUX,J.**« Développons en Java ».France :Eyrolles, 2008 ,394 p.

- [8] **CARTON,O.**« L’essentiel de XML » : l’Université Paris Diderot, cours de M2 Pro, 65 p.
- [9] **Dewas,Bilal.**« Développement d’application mobile » :Centre Universitaire Abdel hafid Boussouf de Mila, cours, 2017, 75 p.
- [10] **Romain Bourdon.** « WampServer, la plate-forme de développement Web sous Windows »[en ligne].Disponible sur : <[http ://www.wampserver.com/](http://www.wampserver.com/)> (Consulter le 21/05/2019).
- [11] **Rigaux,PH.**« Pratique de MySQL et PHP Conception et réalisation de sites web dynamiques »,DUNOD, 4ème édition, 2009.
- [12] **Ponçon G.**« Best practices PHP5 » : Eyrolles , 2005.
- [13] : **Boulmerka,Aissa.** « Développement d’applications web » :Centre Universitaire Abdel hafid Boussouf de Mila, Cours, 2016, 36 p.
- [14] : **Samuel,P.** « Réseaux et systèmes informatiques mobiles » :Fondements, architectures et applications[en ligne].Montréal :Edition revu et augmentée, 2003 , 601 Format.Disponible sur : < [https ://www.amazon.fr/Réseaux-systèmes-informatiques-mobiles - architectures/dp/2553010389](https://www.amazon.fr/Réseaux-systèmes-informatiques-mobiles-architectures/dp/2553010389)> consulté le (15/02/2019).
- [15] : **B.Bathelot.**«Définition :Application mobile»[en ligne].(MAJ le 8 Mar 2017).disponible sur :<[https ://www.définitions-marketing.com/définition/application mobile/](https://www.définitions-marketing.com/définition/application-mobile/)>.(Consulté le 16/02/2019).
- [16] :**Ferhat Yasser, Laaouar Djaafar.**« Réalisation d’une application mobile de gestion de la vie estudiantine pour les Smartphones Android ».Mémoire de Master professionnel, Sciences et technologies de l’information et de la communication(STIC).Mila :Centre Universitaire Abedlhafid boussouf de Mila, 2013, 158 p.

- [17] :**Guignard D,Chable J,.et all.**« Programmation Android : De la conception au déploiement, avec le SDK Google Android 2», Eyrolles , 2011.
- [18] :**M. Meie.**« Android application développent ».Etats Unis (Indiana), wiley publishing editions, 2009.

Annexe

Informatique mobile et les applications mobiles

Puisque notre objectif est de concevoir une application mobile dans le domaine concerné, dans ce contexte nous allons présenter les concepts liés à l'informatique mobile et aux applications mobiles, nous allons également décrire la conception du système d'exploitation qu'on doit utiliser pour développer notre application.

Informatique mobile

★ Définition

La mobilité est devenue au fil du temps une véritable philosophie de vie. Ainsi, elle incite à la redéfinition des habitudes de chacune et façonne graduellement notre relation à l'information, aux transports et au temps., ...

Le concept d'informatique mobile réfère à la capacité d'accéder à des services ou à des applications indépendamment de la localisation physique des utilisateurs, de leurs comportements ou de leurs mouvements.[14]

● Les smartphones

★ Définition

Un smartphone, c'est un concept composé de deux mots anglais *smart* et *phone* ce qui nous donne par traduction un téléphone intelligent qui regroupe les fonctions de téléphonie et celle d'assistants personnels (calendrier, agenda, courrier électronique). C'est un équipement personnel doté de capacités avancées de calcul et riche en fonctionnalités, ainsi il est doté d'un système d'exploitation sophistiqué proche à ceux des ordinateurs personnels et peut supporter plein des applications mobiles. [9]

Application mobile

★ Définition

Une application mobile est un programme applicatif développé qui peut être gratuite ou payante et qui est exécutable à partir du système d'exploitation d'un smartphone

ou d'une tablette. Elle est bien adaptée aux différentes plateformes des smartphones avec leurs contraintes et possibilités ergonomiques (écran tactile).[15]

Une application mobile peut être installée directement dans le smartphone après la fabrication, tandis que d'autres sont téléchargeables via internet et plus précisément de l'**APP Store**, **google Play**.

Les systèmes d'exploitation mobiles

• Définition

Un système d'exploitation mobile ou OS mobile ("Operating System") est un logiciel conçu pour fonctionner sur un appareil mobile, il s'agit d'un ensemble de programmes permettant, entre autres, la gestion des autres applications et interfaces, la connectivité sans fil ainsi que la communication avec le support physique.[16]

• Types de systèmes d'exploitation mobiles

Aujourd'hui il existe plusieurs systèmes d'exploitation qui rivalisent entre eux. Nous allons présenter quelques exemples.

★ Symbian

Symbian est le système d'exploitation utilisé sur la plupart des smartphones Nokia. Cet OS permet de télécharger des applications d'Ovi Store, l'application store développée par Nokia.

★ Apple iOS

Apple iOS est le système d'exploitation développé par Apple pour l'iPhone. Cependant, il a depuis lors aussi été utilisé pour l'iPod Touch, l'iPad, ainsi que pour l'Apple TV. Apple iOS ne permettant pas que ce système supporte des applications extérieures, une application store a été mis en ligne et offre aujourd'hui plus de 700 000 applications à télécharger sur votre appareil.

★ **Blackberry OS**

C'est l'OS développé par Research in Motion (RIM) pour leur ligne de Smartphones BlackBerry. La plateforme BlackBerry était autrefois bien connue pour son support des serveurs e-mails professionnels, ainsi que pour l'envoi et la réception d'e-mails grâce à la technique push (en temps réel) là où le réseau Internet mobile était disponible.

BlackBerry est également très populaire pour ses performances d'assistant personnel digital puisqu'il offre un carnet d'adresses assez poussé, un agenda très complet, la possibilité de faire des listes de tâches, etc.

★ **Windows Phone**

Windows Mobile, développé par Microsoft. Il est actuellement utilisé dans de nombreux smartphones de marques bien connues, telles que HTC, Samsung, LG et Dell. Tout comme Apple a créé sa propre application store, Windows Phone Marketplace est le magasin d'application en ligne développées par Microsoft dans le but de permettre aux utilisateurs de windows mobile phones de télécharger des applications et des jeux.

★ **Android**

Android est un SE basé sur Linux conçu principalement pour équipements à écran tactile comme les smartphones et les tablettes. Android est devenu la plateforme plus populaire pour mobile avec plus de 600000 applications sur le marché. Il est prévu que le store d'Android atteigne le chiffre d'un million d'application bien avant Apple store. Comme c'est la plateforme sur laquelle on va travailler dans ce projet, nous allons revoir cette plateforme plus tard en détails.

Découverte de la plateforme android

★ **Présentation d'android**

Android, un système d'exploitation conçu pour les équipements mobiles dont il est développé par Google. Contrairement à ios, android n'est pas attaché à une marque spécifiée de smartphones, on peut le trouver par exemple dans les marques de : Samsung, Acer, HTC, Alcatel, LG, Huawei, Sony...

Une autre définition dit qu'android est une pile de couches logicielles open source pour les équipements mobiles, cette pile comprend :

- Un système d'exploitation : version modifiée du noyau Linux 2.6.
- Des intergiciels (middlewares).
- Un ensemble d'application clés. [9]

★ Les version d'android

Annoncé officiellement en novembre 2007, le système Android a fait du chemin depuis ses débuts. Depuis la version 1.5, les versions possèdent des noms de desserts (en anglais).[9] Toutes les versions d'android sont résumées dans le tableau suivant :

Version	Nom de code	Date de sortie	Api
1.0	Aucun	11 novembre 2007	1
1.1	Petit four	22 octobre 2008	2
1.5	Cupcake	30 avril 2009	3
1.6	Donut	15 septembre 2009	4
2.0-2.1	Eclair	26 octobre 2009	5
2.2	Froyo	20 mai 2010	8
2.3	Gingerdroid	6 décembre 2010	10
3.0-3.2	Honeycomb	22 février 2011	11
4.0	Icecream sandwich	19 octobre 2011	15
4.1	Jelly Bean	9 juillet 2012	16
4.2		13 novembre 2012	17
4.3		24 juillet 2013	18
4.4	Kitkat	31 octobre 2013	19
5.0	Lollipop	3 novembre 2014	21
5.1		9 mars 2015	22
6.0	Marshmallow	5 octobre 2015	23
7.0 -7.1	Nougat	22 août 2016	24-25
8.0 -8.1	Oreo	21 août 2017	26

TABLE 4: Les versions d'Android.

★ Architecture de la plateforme android

La plate-forme Android est composée de différentes couches :

- **Applications**

Android est fourni avec un ensemble d'application dont un client email, une application SMS, un calendrier, un service de cartographie, un navigateur... toutes écrites en JAVA.

- **Framework de développement(Applications Framework)**

En fournissant une plateforme de développement ouverte, Android offre aux développeurs la possibilité de créer des applications extrêmement riches et innovantes. Les développeurs ont un accès complet à l'API même Framework utilisé par les applications de base.

L'architecture d'application est conçue pour simplifier la réutilisation des composants ; n'importe quelle application peut publier ses capacités et n'importe quelle autre application peut alors faire usage de ses capacités (soumis à des contraintes de sécurité appliquées par le Framework).

- **Bibliothèques (LIBRAIRIES)**

En interne, Android inclut un ensemble de bibliothèques C et C++ utilisées par de nombreux composants de la plateforme Android. Ces bibliothèques sont en réalité accessibles au développeur par l'intermédiaire du Framework Android. En effet, le Framework Android effectue, de façon interne, des appels à des fonctions C/C++ beaucoup plus rapides à exécuter que des méthodes Java standard. La technique Java native Interface (JNI) permet d'effectuer des échanges entre le code Java et le code C et C++.

- **Android Runtime**

Chaque application Android tourne dans son propre processus et avec sa propre instance de machines virtuelles Dalvik (Dalvik VM). Dalvik VM a été conçue pour optimiser l'exécution multiple de machines virtuelles. la Dalvik VM repose sur le noyau

Linux qui s'occupe des fonctions telles que les threads ou la gestion basse-niveau de la mémoire.

- **Linux Kernel**

Android est basé sur le noyau Linux (2.6). Il profite donc de ces caractéristique de stabilité et efficacité en matière de gestion mémoire, gestion de processus, modèle de sécurité, soutien de bibliothèque partagé, etc. en plus de ça le noyau d'Android est adapté au mobile par des modules spécifiques au contexte comme la gestion d'énergie.

[17]

La figure suivante résume cette architecture :



FIGURE 49 – Architecture de la plateforme android

★ Cycle de vie d'une application Android

La figure suivante présente le cycle de vie d'une application Android :

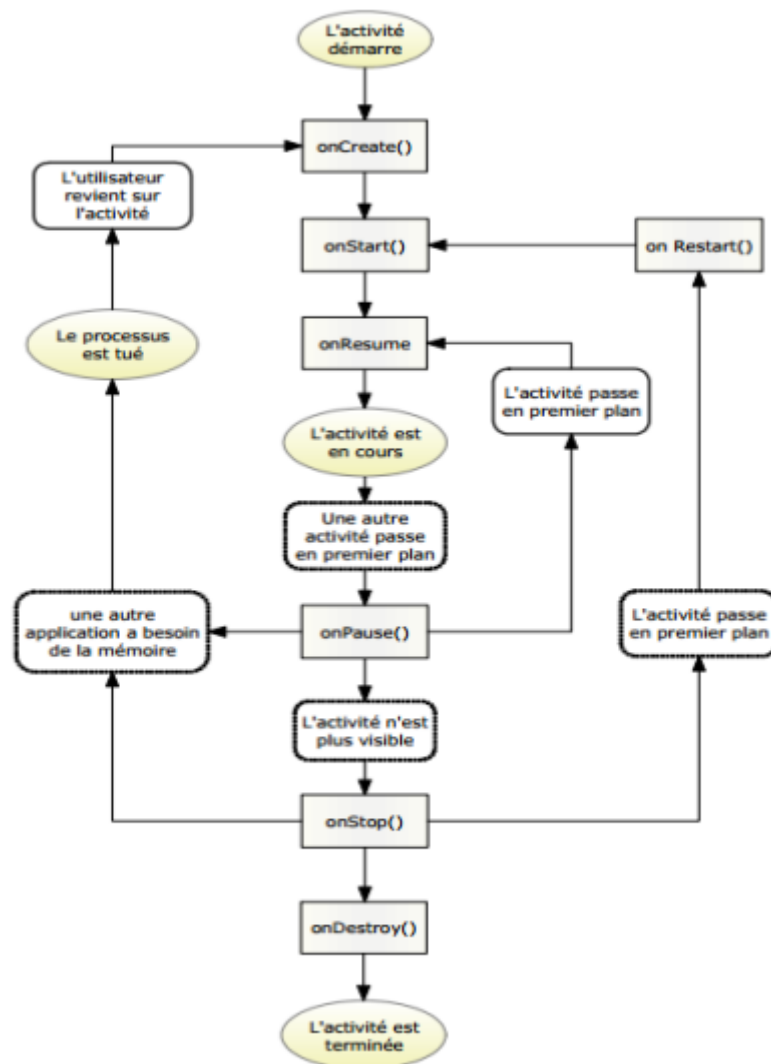


FIGURE 50 – le cycle de vie d'une application Android

Le fonctionnement du cycle de vie d'une application mobile qui sera illustré dans la figure si-dessus se fait comme suit :

- Au démarrage de l'activité ; la méthode `onCreate` est appelée. Pendant l'utilisation d'une activité, l'utilisateur presse la touche accueil, ou bien l'application téléphone, qualifiée comme prioritaire et qui interrompt son fonctionnement par un appel téléphonique entrant.

- Une fois l'appel téléphonique terminé, le système réveille l'activité précédemment mise en pause (appel de `onRestart`, `onStart`).

-Si l'activité reste trop longtemps en pause, le système à besoin de mémoire, il détruit l'activité (appel de `onDestroy`).

-`onPause` et `onResume` rajoutent un état à l'activité, puisqu'ils interviennent dans le cas d'activités partiellement visibles, mais qui n'ont pas le focus.

-La méthode `onPause` implique également que la vie de cette application n'est plus une priorité pour le système. Donc si celui-ci a besoin de mémoire, l'activité peut être fermée. Ainsi, il est préférable, lorsque l'on utilise cette méthode, de sauvegarder l'état de l'activité dans le cas où l'utilisateur souhaitera y revenir avec la touche accueil. **[18]**

Conception

Description des cas d'utilisation

- « Modifier profil »

- Description textuelle

Cas d'utilisation	Modifier profil.
Acteur	Utilisateur.
But	Ce cas permet à l'utilisateur de modifier les informations de son profil.
Pré condition	L'utilisateur doit être authentifié, il est dans son espace.
Poste condition	Les informations de l'utilisateur sont modifiées.
Scénario nominal	1-Le système affiche les informations de profil. 2-L'utilisateur modifie les informations souhaitées et confirme. 3-Le système vérifie la validité des informations saisies et enregistre les nouvelles informations. 4-Le système affiche un message de confirmation « Modification avec succès ».
Scénario alternatif	• Les informations saisies sont invalide.le système affiche un message d'erreur On reprend scénario nominal au point 1. • L'utilisateur existe déjà dans l'application.le système affiche un message d'erreur. On reprend scénario nominal au point 2.
Scénario optionnelle	• L'utilisateur peut annuler l'opération en cours à n'importe quel moment pourvue qu'il ne l'a pas validé.

TABLE 5: Description textuelle du cas d'utilisation

« Modifier profil »

– Diagramme de séquence

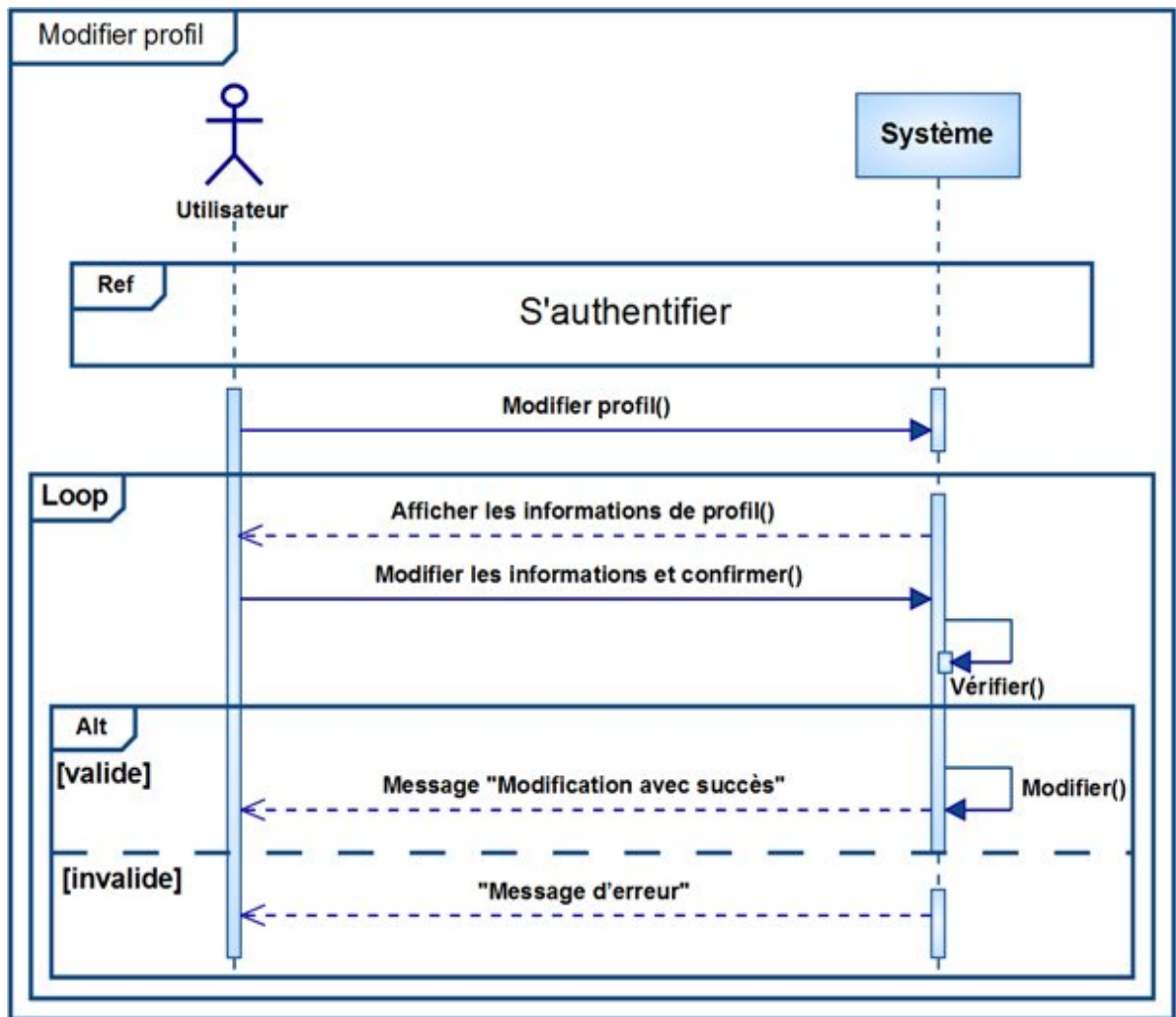


FIGURE 51 – Diagramme de séquence de cas Modifier profil

• « Gérer la liste des groupes enseignés »

– Description textuelle

Cas d'utilisation	Gérer la liste des groupes enseignés
Acteurs	Enseignant
But	Ce cas permet à l'enseignant de consulter, ajouter, modifier ou supprimer des groupes enseignés.
Pré-condition	L'enseignant doit être authentifié, il est dans son espace.
Scénario-nominal	L'enseignant choisit l'action à effectuer : 1-Consultation d'un groupe (voir le cas Consulter groupe). 2- L'ajout d'un groupe (voir le cas Ajouter groupe). 3- La modification des informations d'un groupe(voir le cas Modifier groupe). 4- La suppression d'un groupe (voir le cas Supprimer groupe).
Scénario optionnelle	• La liste des groupes est vide,le système affiche un message d'information « la liste est vide ».

TABLE 6: Description textuelle du cas d'utilisation

« Gérer la liste des groupes enseignés »

– Diagramme de séquence

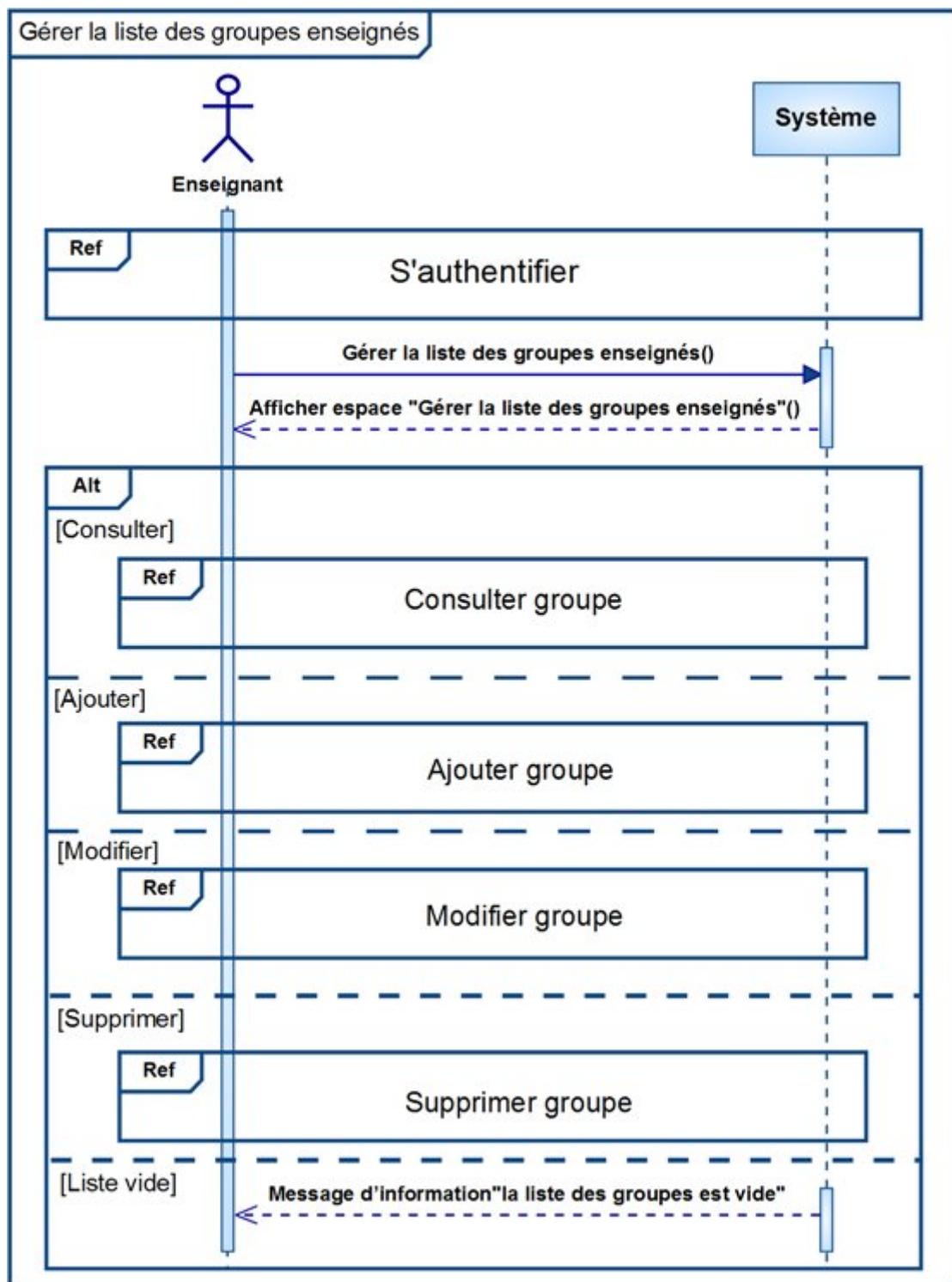


FIGURE 52 – Diagramme de séquence du cas Gérer la liste des groupes enseignés

- « Consulter groupe »

- Description textuelle

Cas d'utilisation	Consulter groupe.
Acteur	Enseignant.
But	Ce cas permet à l'enseignant de consulter les informations d'un groupe de ses groupes.
Pré condition	L'enseignant est dans son espace et il a choisi l'action « consulter groupe ».
Poste condition	Le groupe sélectionné est affiché avec sa liste d'étudiants.
Scénario nominal	1-Le système affiche la liste des groupes. 2-L'enseignant choisit un des groupes affichés. 3-Le système affiche les informations du groupe sélectionné.

TABLE 7: Description textuelle du cas d'utilisation

- « Consulter groupe »

- Diagramme de séquence

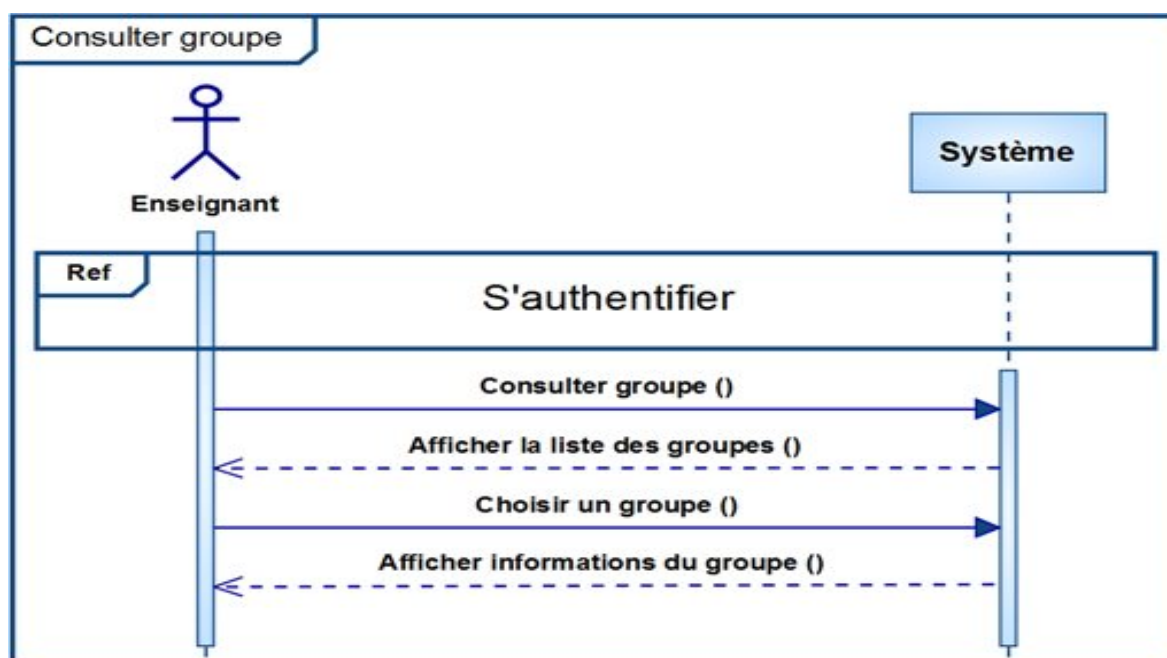


FIGURE 53 – Diagramme de séquence du cas Consulter groupe

- « Modifier groupe »

- Description textuelle

Cas d'utilisation	Modifier groupe.
Acteur	Enseignant.
But	Ce cas permet à l'enseignant de modifier les informations d'un groupe.
Pré condition	L'enseignant est dans son espace et il a choisi l'action « Modifier groupe ».
Poste condition	Les informations du groupe sont modifiées.
Scénario nominal	1-L'enseignant consulte la liste des groupes et choisit le groupe à modifier. 2-Le système affiche les informations du groupe. 3-L'enseignant modifie les informations souhaitées et confirme. 4-Le système vérifie la validité des informations saisies et enregistre les nouvelles informations. 5-Le système affiche un message de confirmation de modification « Modification avec succès ».
Scénario alternatif	• Les informations saisies sont invalide.le système affiche un message d'erreur On reprend scénario nominal au point 2.
Scénario optionnelle	• L'enseignant peut annuler l'opération en cours à n'importe quel moment pourvue qu'il ne l'a pas validé.

TABLE 8: Description textuelle du cas d'utilisation

« Modifier groupe »

– Diagramme de séquence

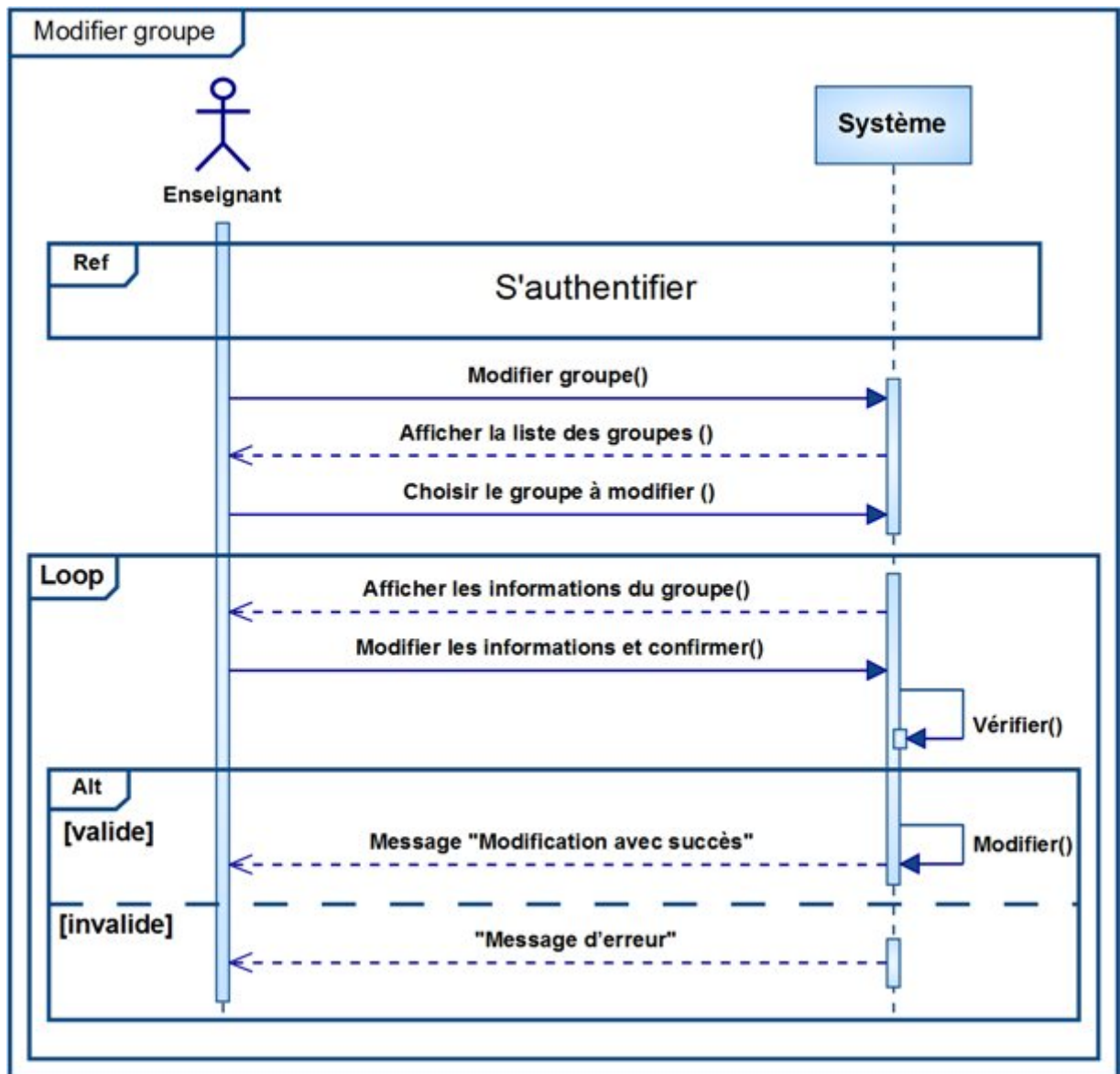


FIGURE 54 – Diagramme de séquence du cas Modifier groupe

- « Supprimer groupe »

- Description textuelle

Cas d'utilisation	Supprimer groupe.
Acteur	Enseignant.
But	Ce cas permet à l'enseignant de supprimer un groupe.
Pré condition	L'enseignant est dans son espace et il a choisi l'action « supprimer groupe ».
Poste condition	groupe supprimé de la liste des groupes enseignés.
Scénario nominal	1-L'enseignant consulte la liste des groupes et choisit le groupe à supprimer. 2-Le système affiche un message de confirmation de suppression. 3-L'enseignant confirme la suppression. 4-Le système supprime le groupe concerné. 5-Le système affiche une notification «Suppression avec succès »
Scénario optionnelle	• L'enseignant peut annuler l'opération lorsqu'il reçoit un message de confirmation.

TABLE 9: Description textuelle du cas d'utilisation

« Supprimer groupe »

– Diagramme de séquence

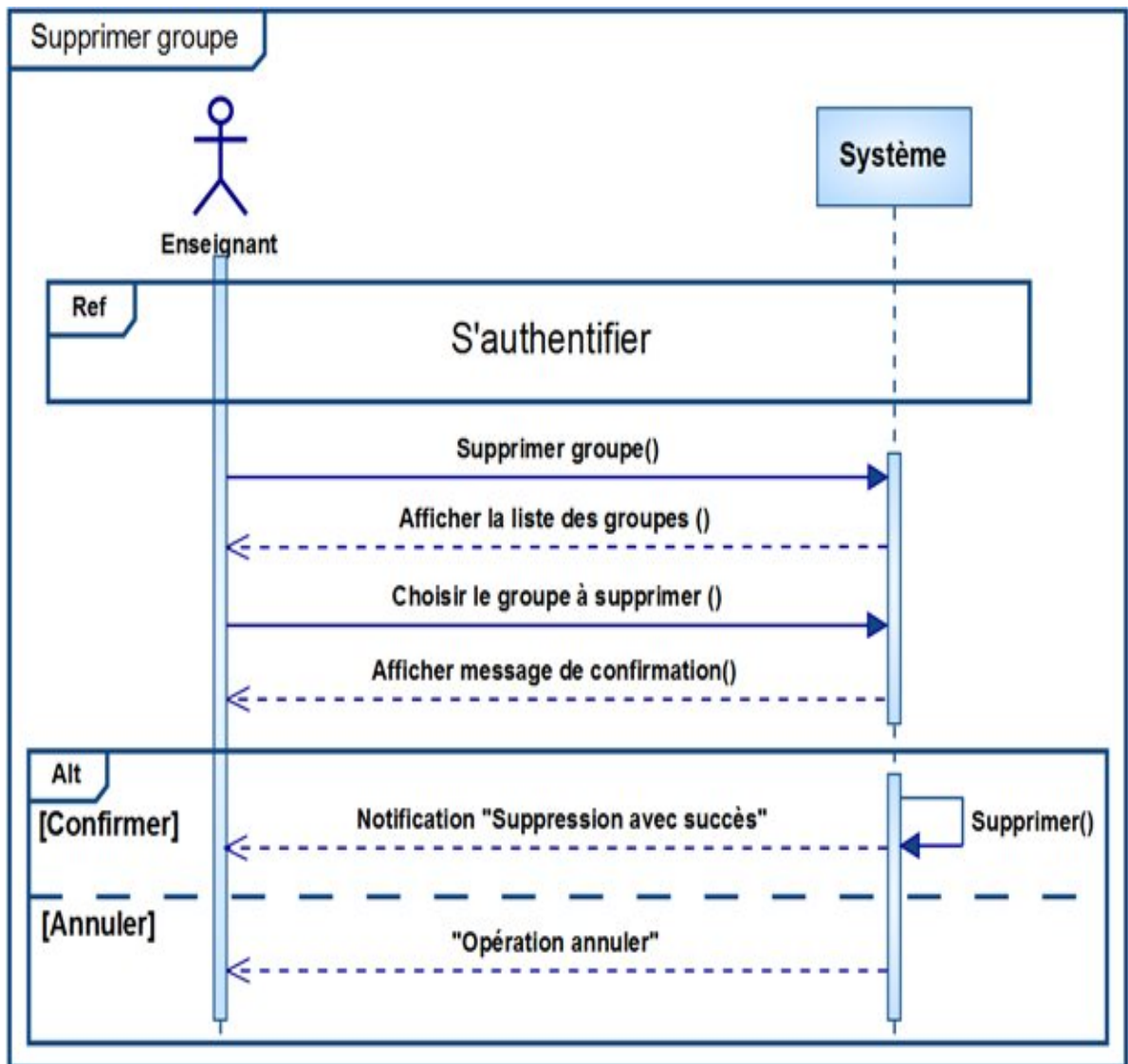


FIGURE 55 – Diagramme de séquence du cas Supprimer groupe

- « Imprimer »

- Description textuelle

Cas d'utilisation	Imprimer.
Acteur	Enseignant (principal), Imprimante (secondaire).
But	Ce cas permet à l'enseignant d'effectuer une impression.
Pré condition	L'enseignant est dans son espace et il a choisi l'action « imprimer ».
Poste condition	Réalisation de l'impression.
Scénario nominal	1-Le système affiche l'interface d'impression. 2-L'enseignant choisit les détails du fichier qu'il le besoin d'imprimer et confirme l'opération. 3-Le système affiche une notification « Impression avec succès »
Scénario optionnelle	• L'enseignant peut annuler l'opération en cours à n'importe quel moment pourvue qu'il ne l'a pas validé.

TABLE 10: Description textuelle du cas d'utilisation

- « Imprimer »

- Diagramme de séquence



FIGURE 56 – Diagramme de séquence du cas Imprimer

- « Importer la liste des étudiants »

- Description textuelle

Cas d'utilisation	Importer la liste des étudiants.
Acteur	Enseignant (principal), Fichier (secondaire).
But	Ce cas permet l'enseignant d'importer la liste des étudiants.
Pré condition	L'enseignant est dans son espace et il a choisi l'action « Importer la liste des étudiants ».
Poste condition	Importation de la liste des étudiants.
Scénario nominal	1-Le système affiche l'interface d'importation. 2-L'enseignant choisit le fichier à importer et confirme l'opération. 3-Le système affiche une notification «Importation avec succès ».
Scénario optionnelle	• L'enseignant peut annuler l'opération en cours à n'importe quel moment pourvue qu'il ne l'a pas validé.

TABLE 11: Description textuelle du cas d'utilisation
« Importer la liste des étudiants »

- Diagramme de séquence

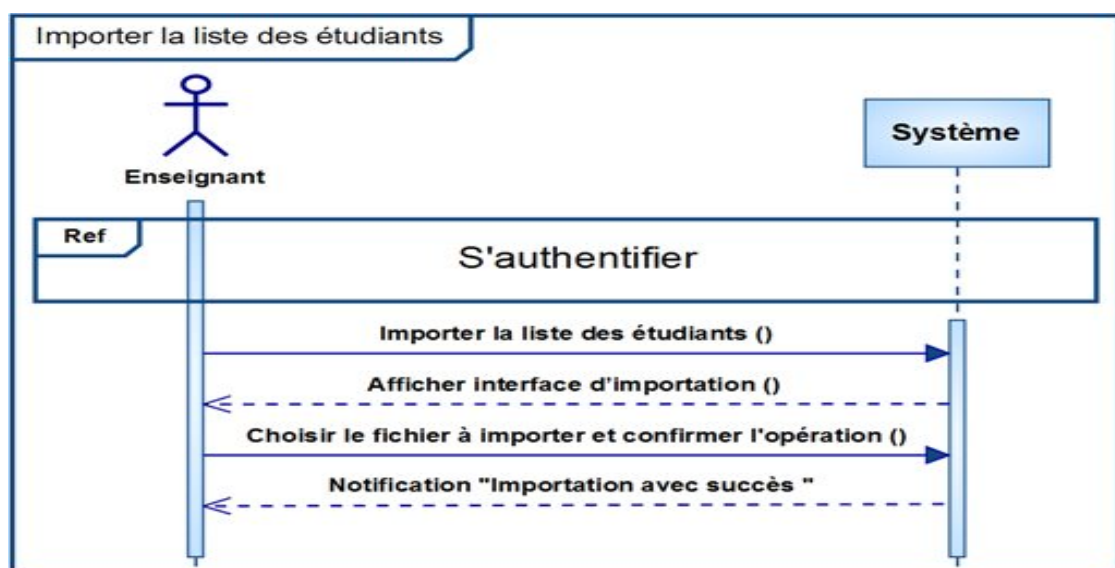


FIGURE 57 – Diagramme de séquence du cas Importer la liste des étudiants

- « Gérer étudiants »

- Description textuelle

Cas d'utilisation	Gérer étudiants.
Acteur	Enseignant.
But	Ce cas permet à l'enseignant de consulter, ajouter, modifier ou supprimer des étudiants.
Pré condition	L'enseignant est dans son espace, il a choisi l'action « Gérer étudiants ».
Scénario nominal	L'enseignant choisit l'action à effectuer : 1-Consultation d'un étudiant (voir le cas Consulter étudiant). 2-L'ajout d'un étudiant(voir le cas Ajouter étudiant). 3-La modification des informations d'un étudiant (voir le cas Modifier étudiant). 4-La suppression d'un étudiant (voir le cas Supprimer étudiant).
Scénario optionnelle	• La liste des étudiants est vide,le système affiche un message d'information « la liste est vide ».

TABLE 12: Description textuelle du cas d'utilisation

« Gérer étudiants »

– Diagramme de séquence

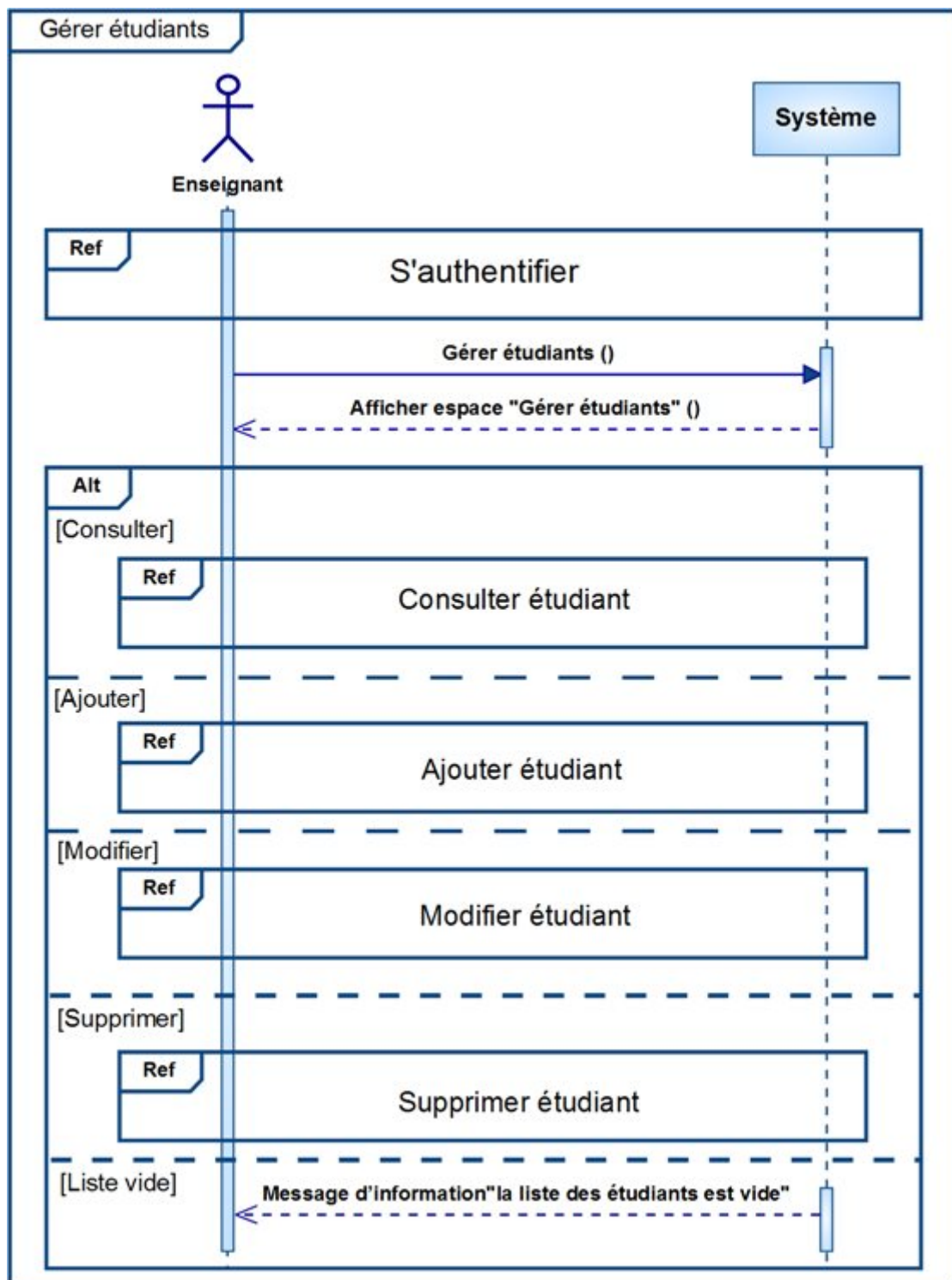


FIGURE 58 – Diagramme de séquence du cas Gérer étudiants

- « Consulter étudiant »

- Description textuelle

Cas d'utilisation	Consulter étudiant.
Acteur	Enseignant.
But	Ce cas permet à l'enseignant de consulter un étudiant.
Pré condition	L'enseignant est dans son espace et il a choisi l'action « consulter étudiant ».
Poste condition	L'étudiant choisi est affiché.
Scénario nominal	1-Le système affiche la liste des étudiants. 2-L'enseignant choisit un des étudiants affichés. 3-Le système affiche l'étudiant choisi.

TABLE 13: Description textuelle du cas d'utilisation

- « Consulter étudiant »

- Diagramme de séquence

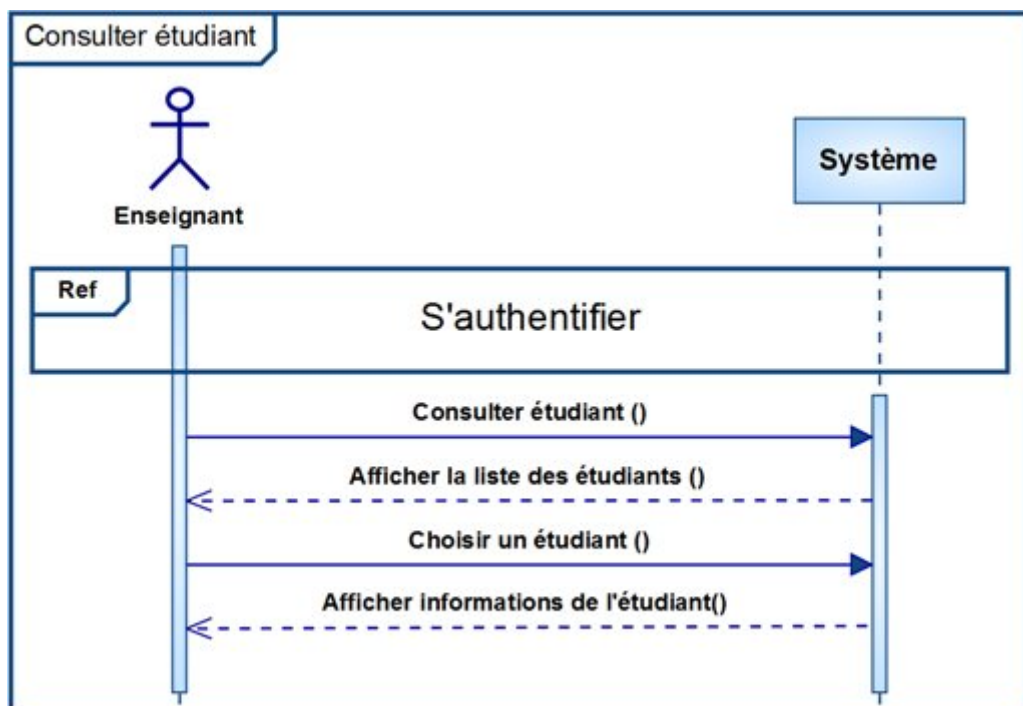


FIGURE 59 – Diagramme de séquence du cas Consulter étudiant

- « Modifier étudiant »

- Description textuelle

Cas d'utilisation	Modifier étudiant.
Acteur	Enseignant.
But	Ce cas permet à l'enseignant de modifier les informations d'un étudiant.
Pré condition	L'enseignant est dans son espace et il a choisi l'action « Modifier étudiant ».
Poste condition	Les informations de l'étudiant sont modifiées.
Scénario nominal	1-L'enseignant consulte la liste des étudiants et choisit l'étudiant à modifier. 2-Le système affiche les informations de l'étudiant. 3-L'enseignant modifie les informations souhaitées et confirme. 4-Le système vérifie la validité des informations saisies et enregistre les nouvelles informations. 5-Le système affiche un message de confirmation de modification « Modification avec succès ».
Scénario alternatif	• Les informations saisies sont invalide.le système affiche un message d'erreur.On reprend scénario nominal au point 2.
Scénario optionnelle	• L'enseignant peut annuler l'opération en cours à n'importe quel moment pourvue qu'il ne l'a pas validé.

TABLE 14: Description textuelle du cas d'utilisation

« Modifier étudiant »

– Diagramme de séquence

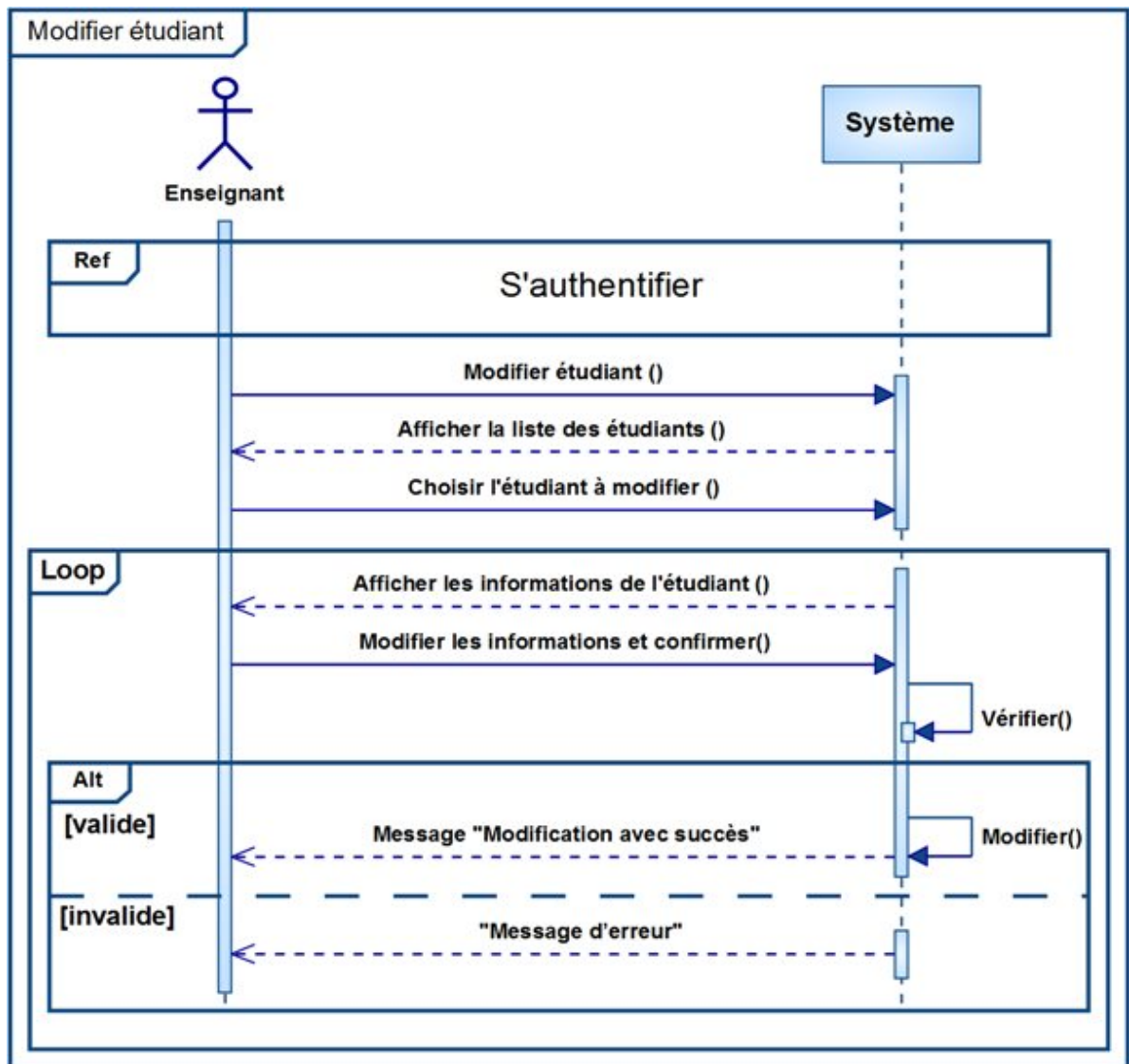


FIGURE 60 – Diagramme de séquence du cas Modifier étudiant

- « Supprimer étudiant »

- Description textuelle

Cas d'utilisation	Supprimer étudiant.
Acteur	Enseignant.
But	Ce cas permet à l'enseignant de supprimer un étudiant
Pré condition	L'enseignant est dans son espace et il a choisi l'action « supprimer étudiant ».
Poste condition	L'étudiant est supprimé .
Scénario nominal	1-L'enseignant consulte la liste des étudiants et choisit l'étudiant à supprimer. 2-Le système affiche un message de confirmation de suppression 3-L'enseignant confirme la suppression. 4-Le système supprime l'étudiant concerné. 5-Le système affiche une notification «Suppression avec succès ».
Scénario optionnelle	• L'enseignant peut annuler l'opération lorsqu'il reçoit un message de confirmation.

TABLE 15: Description textuelle du cas d'utilisation

« Supprimer étudiant »

– Diagramme de séquence

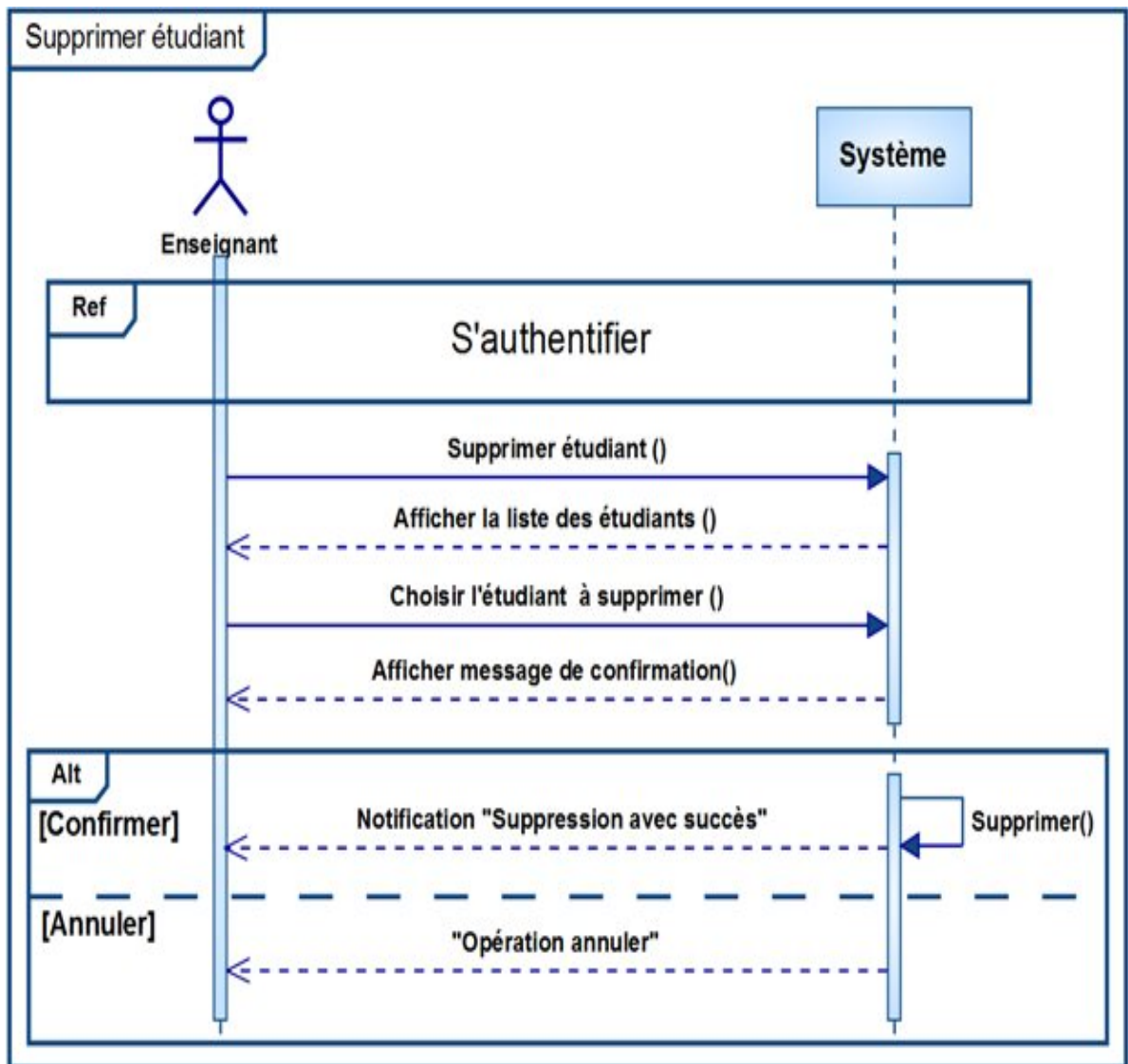


FIGURE 61 – Diagramme de séquence du cas Supprimer étudiant

- « Gérer tests et corrigés type »

- Description textuelle

Cas d'utilisation	Gérer tests et corrigés type
Acteurs	Enseignant
But	Ce cas permet à l'enseignant de consulter, ajouter, modifier, supprimer ou archiver des tests.
Pré-condition	L'enseignant doit être authentifié, il est dans son espace.
Scénario-nominal	L'enseignant choisit l'action à effectuer : 1-Consultation du test et corrigé type(voir le cas Consulter test et corrigé type). 2- L'ajout d'un test et corrigé type (voir le cas Ajouter test et corrigé type). 3-La modification des informations d'un test(voir le cas Modifier test et corrigé type). 4-La suppression d'un test(voir le cas Supprimer test et corrigé type). 5-L'archivage d'un test (voir le cas Archiver test et corrigé type).
Scénario optionnelle	• La liste des tests est vide,le système affiche un message d'information « la liste est vide ».

TABLE 16: Description textuelle du cas d'utilisation

« Gérer tests et corrigés type »

– Diagramme de séquence

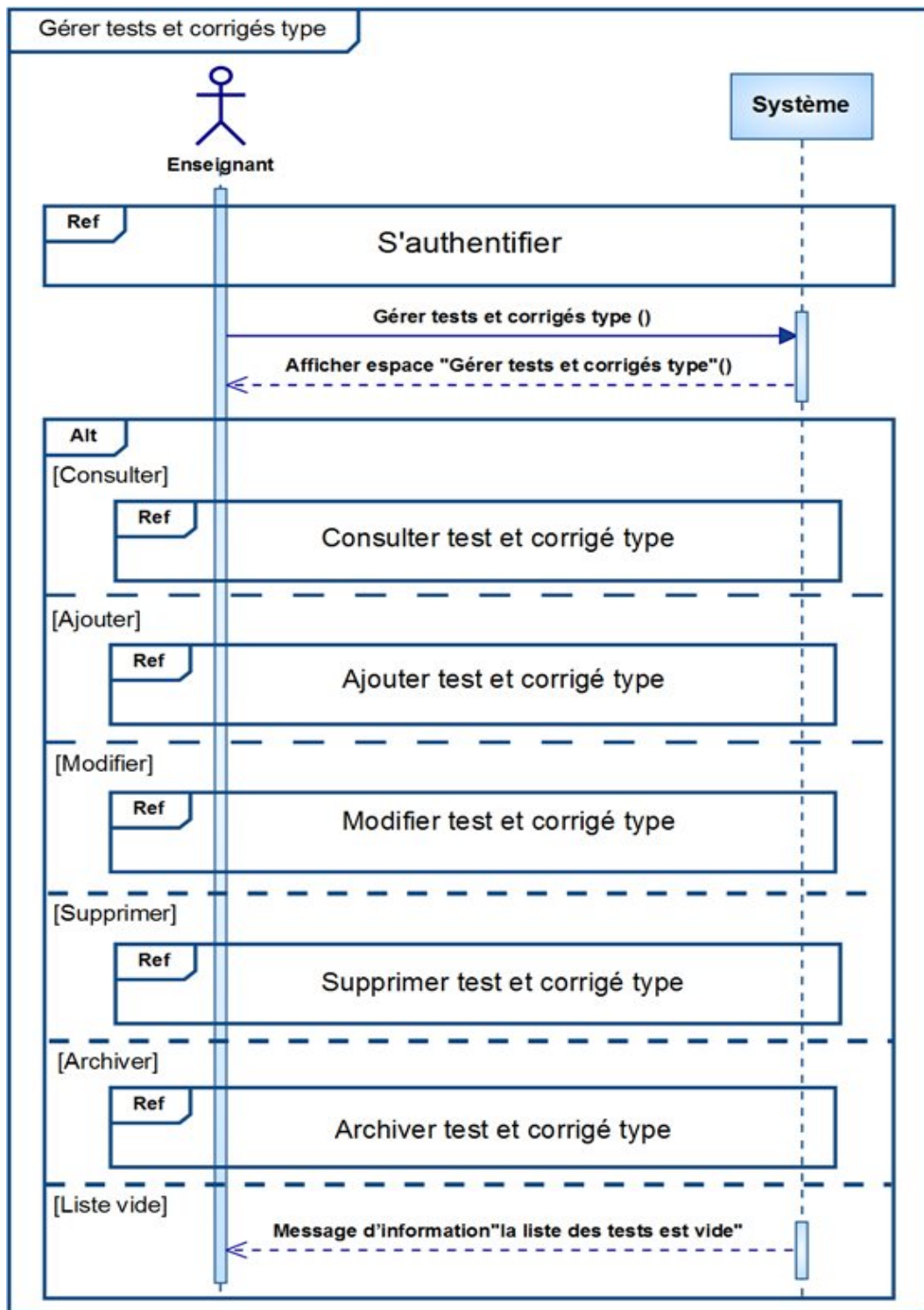


FIGURE 62 – Diagramme de séquence du cas Gérer tests et corrigés type

• « Modifier test et corrigé type »

– Description textuelle

Cas d'utilisation	Modifier test et corrigé type.
Acteur	Enseignant.
But	Ce cas permet à l'enseignant de modifier les informations d'un test.
Pré condition	L'enseignant est dans son espace et il a choisi l'action « modifier test et corrigé type ».
Poste condition	Les modifications effectuées sont enregistrées.
Scénario nominal	1-L'enseignant consulte la liste des tests et choisit le test à modifier. 2-Le système affiche l'interface de modification qui contient les informations du test avec ses questions et ses réponses. 3-L'enseignant modifie les informations souhaitées et confirme. 4-Le système vérifie la validité des informations saisies et enregistre les nouvelles informations. 5-Le système affiche un message de confirmation « Modification avec succès ».
Scénario alternatif	• Les informations saisies sont invalide.le système affiche un message d'erreur.On reprend scénario nominal au point 2.
Scénario optionnelle	• L'enseignant peut annuler l'opération en cours à n'importe quel moment pourvue qu'il ne l'a pas validé.

TABLE 17: Description textuelle du cas d'utilisation

« Modifier test et corrigé type »

– Diagramme de séquence

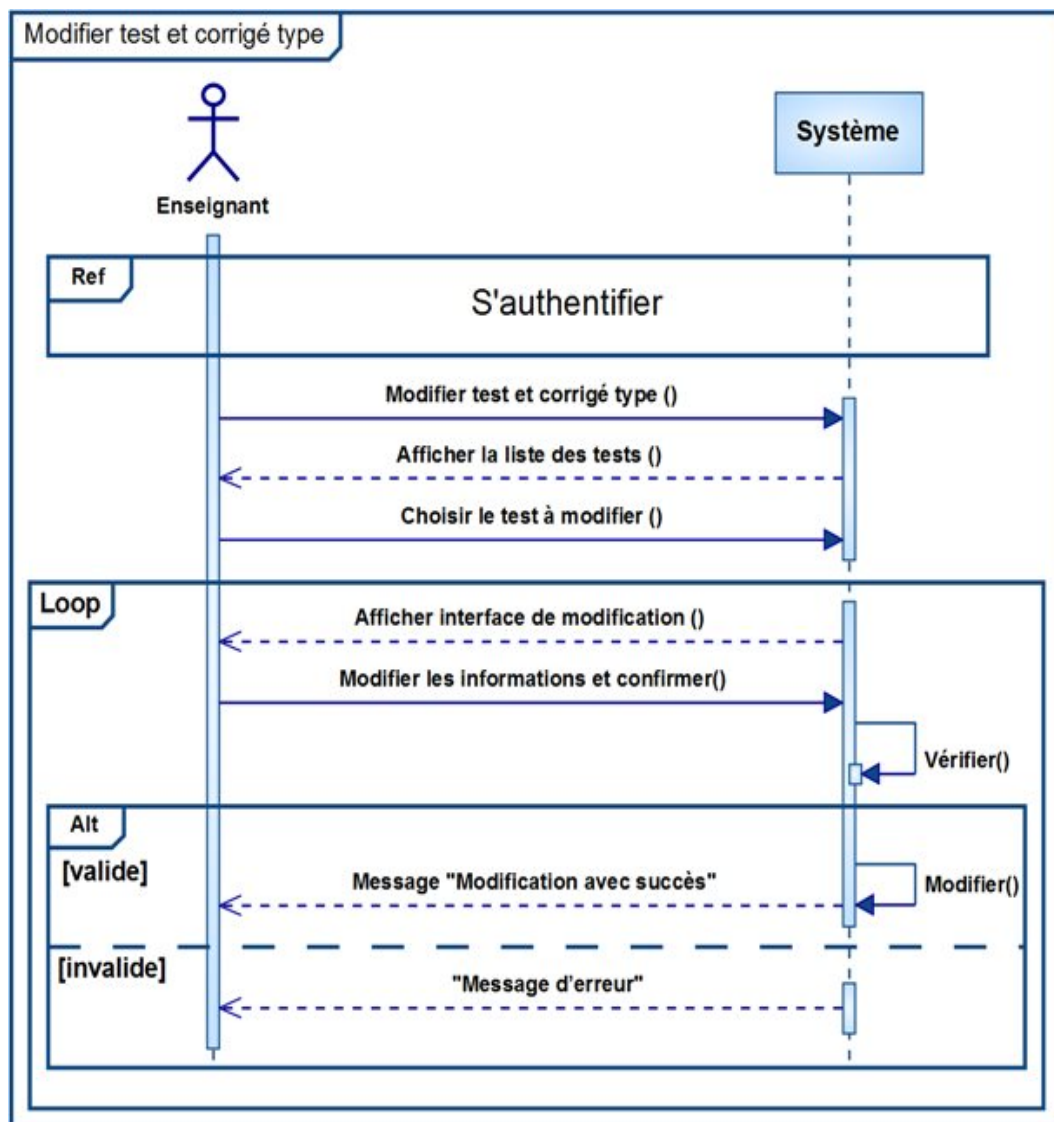


FIGURE 63 – Diagramme de séquence du cas Modifier test et corrigé type

- « Supprimer test et corrigé type »

- Description textuelle

Cas d'utilisation	Supprimer test et corrigé type.
Acteurs	Enseignant.
But	Ce cas permet à l'enseignant de supprimer un test.
Pré-condition	L'enseignant est dans son espace, il a choisi l'action «supprimer test et corrigé type.
Post-condition	Test supprimée du système.
Scénario nominal	1-L'enseignant consulte la liste des tests et choisit le test à supprimer. 2-Le système affiche un message de confirmation. 3-L'enseignant confirme la suppression. 4-Le système supprime le test sélectionné. 5-Le système affiche une notification «Suppression avec succès ».
Scénario optionnelle	• L'enseignant peut annuler l'opération lorsqu'il reçoit un message de confirmation.

TABLE 18: Description textuelle du cas d'utilisation

Supprimer test et corrigé type

– Diagramme de séquence

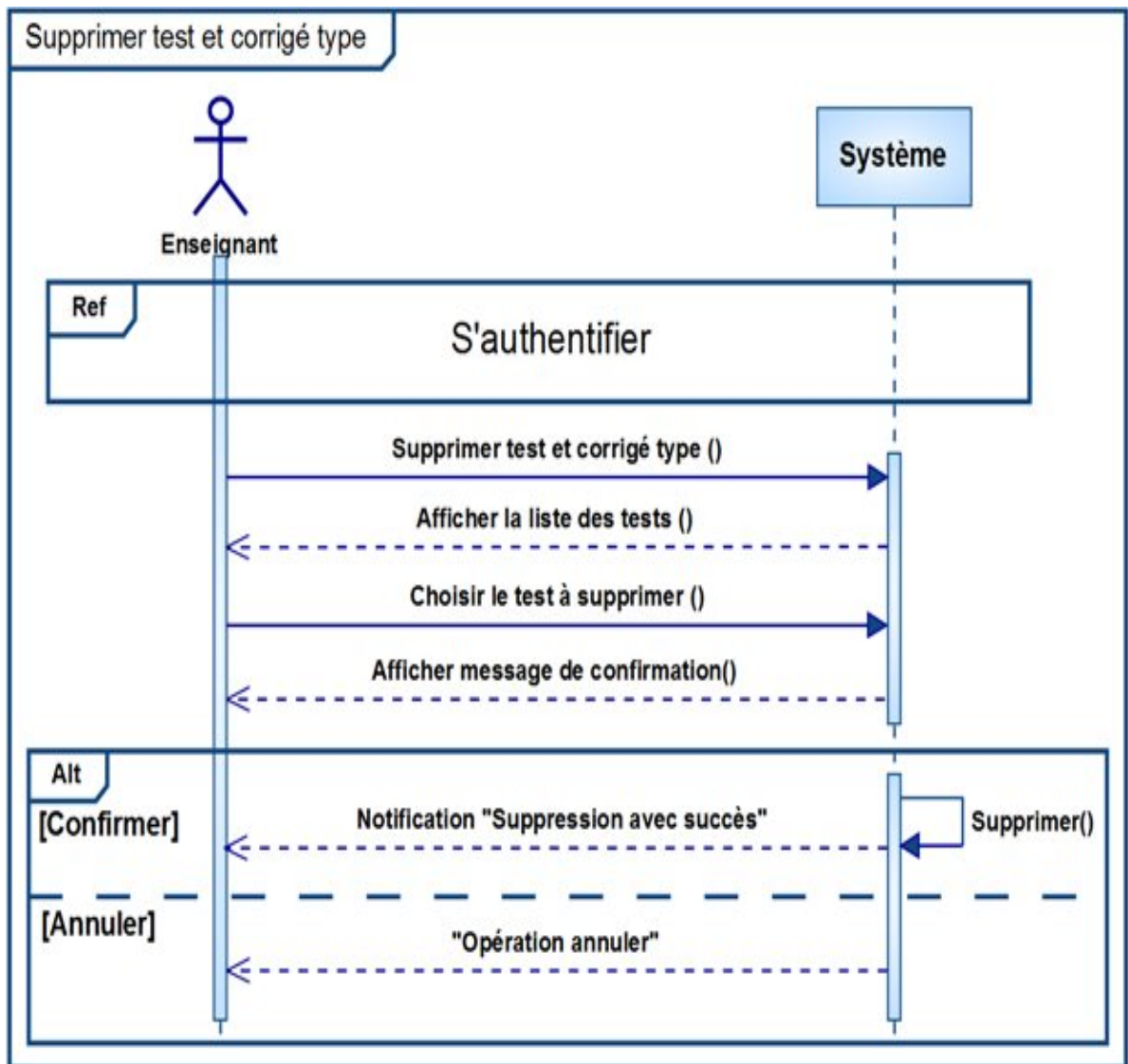


FIGURE 64 – Diagramme de séquence du cas Supprimer test et corrigé type

- « Archiver test et corrigé type »

- Description textuelle

Cas d'utilisation	Archiver test et corrigé type
Acteur principale	Enseignant(principal),Plateforme e-learning(secondaire)
objectif	Ce cas permet à l'enseignant d'archiver un test .
Pré condition	L'enseignant est dans son espace, il a choisi l'action « archiver test et corrigé type».
Post condition	Un test est archivé.
Scénario nominal	1-L'enseignant consulte la liste des tests et choisit le test à archivé. 2-Le système affiche un message de confirmation d'archivage. 3-L'enseignant confirme l'opération d'archivage. 4-Le système archivé le test concerné. 5-Le système affiche une notification «archivage avec succès ».
Scénario optionnelle	•L'enseignant peut annuler l'opération lorsqu'il reçoit un message de confirmation.

TABLE 19: Description textuelle du cas d'utilisation

Archiver test et corrigé type

– Diagramme de séquence

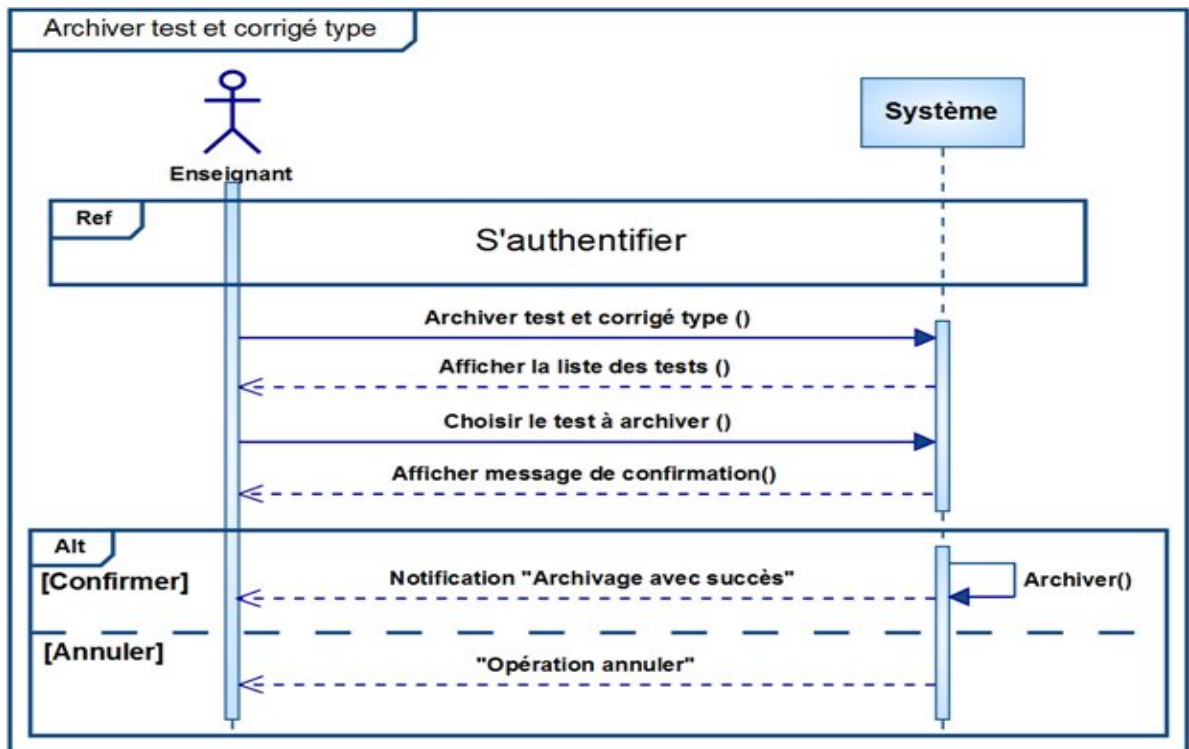


FIGURE 65 – Diagramme de séquence du cas Archiver test et corrigé type

- « Gérer sessions »

- Description textuelle

Cas d'utilisation	Gérer sessions.
Acteur	Enseignant.
But	Ce cas permet à l'enseignant de consulter, créer, modifier ou supprimer des sessions.
Pré condition	L'enseignant doit être authentifié, il est dans son espace.
Scénario nominal	L'enseignant choisit l'action à effectuer : 1-Consultation d'une session (voir le cas Consulter session). 2-Création d'une session (voir le cas Créer session) 3-Modification d'une session (voir le cas Modifier session) 4-Suppression d'une session (voir le cas Supprimer session)
Scénario optionnelle	• La liste des sessions est vide, le système affiche un message d'information « la liste est vide ».

TABLE 20: Description textuelle du cas d'utilisation

« Gérer sessions »

– Diagramme de séquence

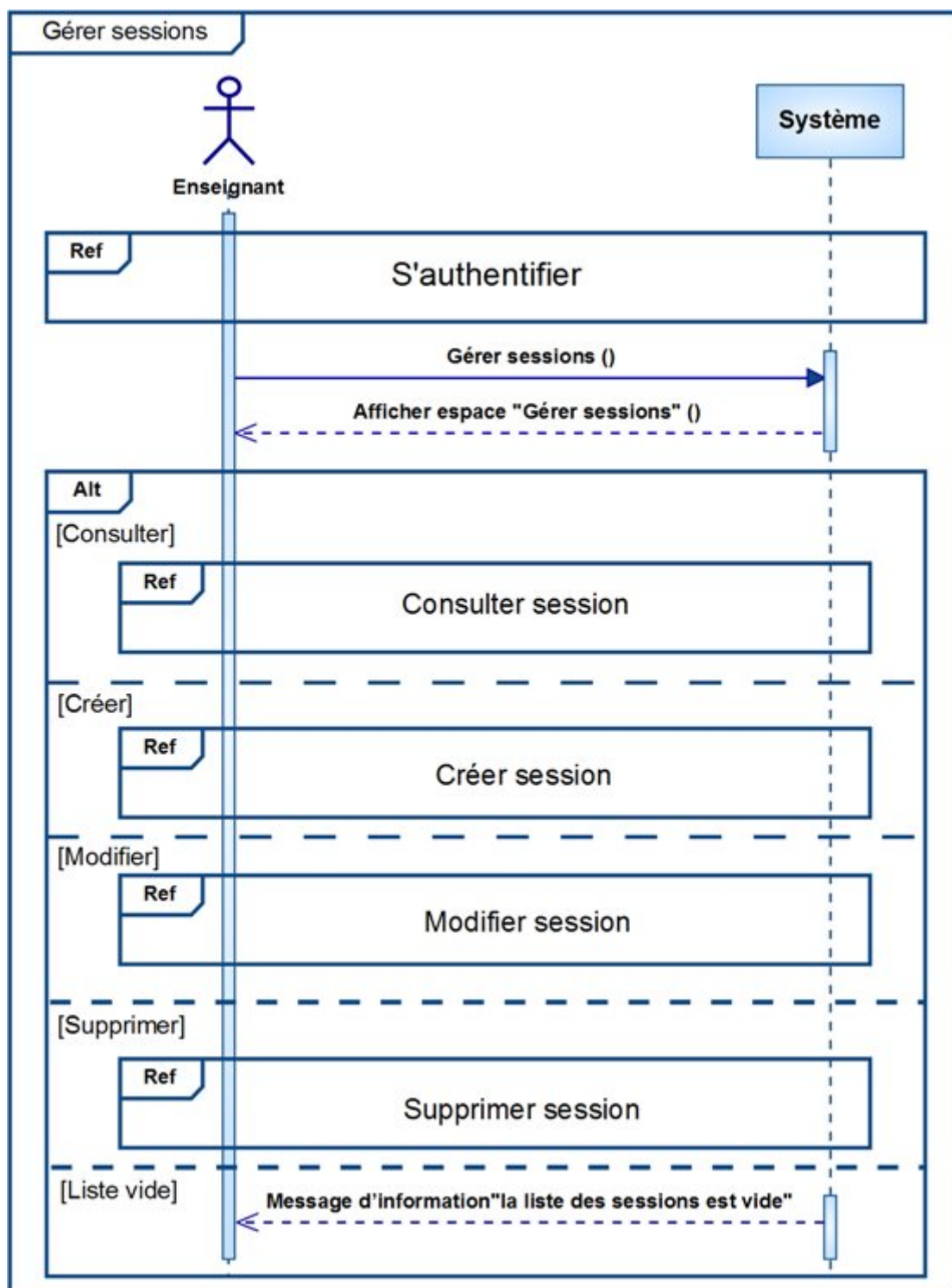


FIGURE 66 – Diagramme de séquence du cas Gérer session

- « Consulter sessions »

- Description textuelle

Cas d'utilisation	Consulter session.
Acteur	Enseignant.
But	Ce cas permet à l'enseignant de consulter une session.
Pré condition	L'enseignant est dans son espace et il a choisi l'action «consulter session ».
Poste condition	La session choisie avec ces différentes informations est affichée.
Scénario nominal	1-Le système affiche la liste des sessions. 2-L'enseignant choisit une des sessions affichées. 3-Le système affiche la session choisie.

TABLE 21: Description textuelle du cas d'utilisation

- « Consulter session »

- Diagramme de séquence

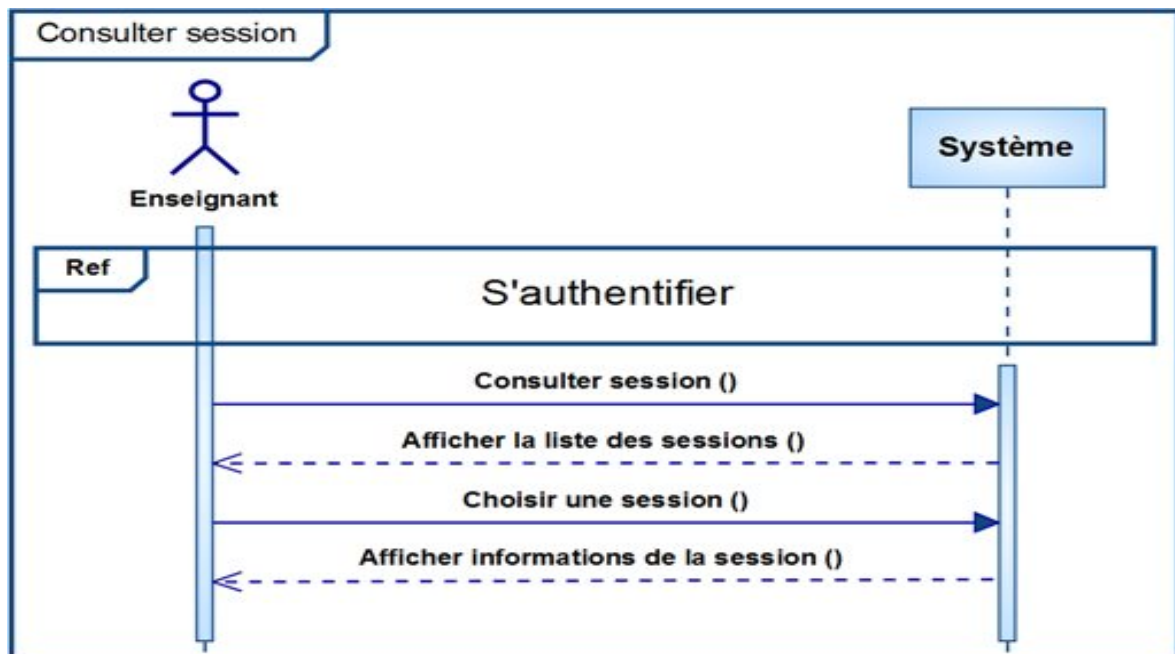


FIGURE 67 – Diagramme de séquence du cas Consulter session

- « Modifier session »

- Description textuelle

Cas d'utilisation	Modifier session.
Acteur	Enseignant.
But	Ce cas permet à l'enseignant de modifier les informations d'une session .
Pré condition	L'enseignant est dans son espace et il a choisi l'action « Modifier session ».
Poste condition	Les informations de la session sont modifiées.
Scénario nominal	1-L'enseignant consulte la liste des sessions et choisit la session à modifier. 2-Le système affiche les informations de la session. 3-L'enseignant modifie les informations souhaitées et confirme. 4-Le système vérifie la validité des informations saisies et enregistre les nouvelles informations. 5-Le système affiche un message de confirmation de modification « Modification avec succès ».
Scénario alternatif	• Les informations saisies sont invalide.le système affiche un message d'erreur. On reprend scénario nominal au point 2.
Scénario optionnelle	• L'enseignant peut annuler l'opération en cours à n'importe quel moment pourvue qu'il ne l'a pas validé.

TABLE 22: Description textuelle du cas d'utilisation

« Modifier session »

– Diagramme de séquence

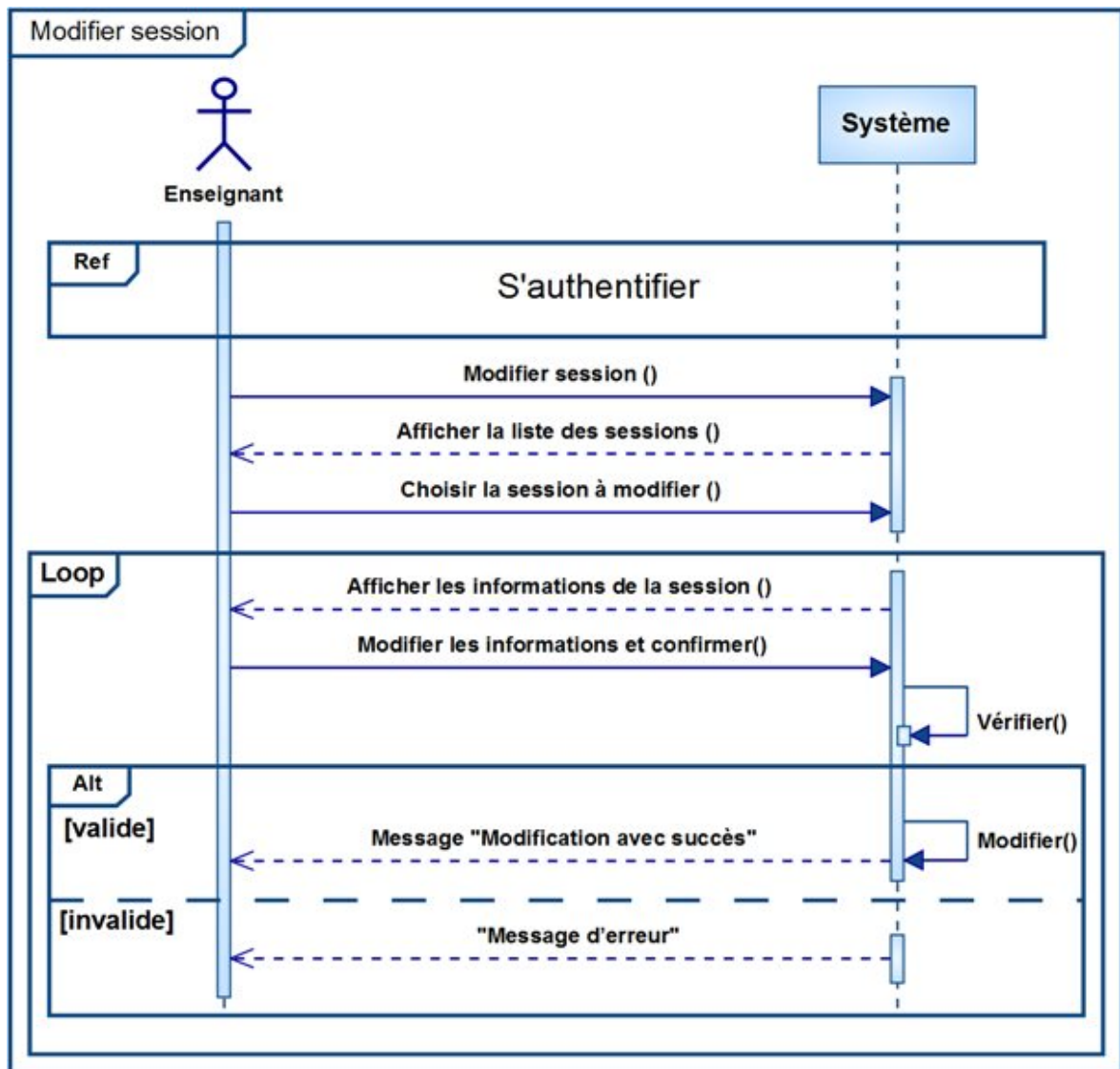


FIGURE 68 – Diagramme de séquence du cas Modifier session

- « Supprimer session »

- Description textuelle

Cas d'utilisation	Supprimer session.
Acteur	Enseignant.
But	Ce cas permet à l'enseignant de supprimer une session
Pré condition	L'enseignant est dans son espace et il a choisi l'action « supprimer session ».
Poste condition	La session est supprimée.
Scénario nominal	1-L'enseignant consulte la liste des sessions et choisit la session à supprimer. 2-Le système affiche un message de confirmation de suppression. 3-L'enseignant confirme la suppression. 4-Le système supprime la session concerné. 5-Le système affiche une notification «Suppression avec succès »
Scénario optionnelle	• L'enseignant peut annuler l'opération lorsqu'il reçoit un message de confirmation.

TABLE 23: Description textuelle du cas d'utilisation

« Supprimer session »

– Diagramme de séquence

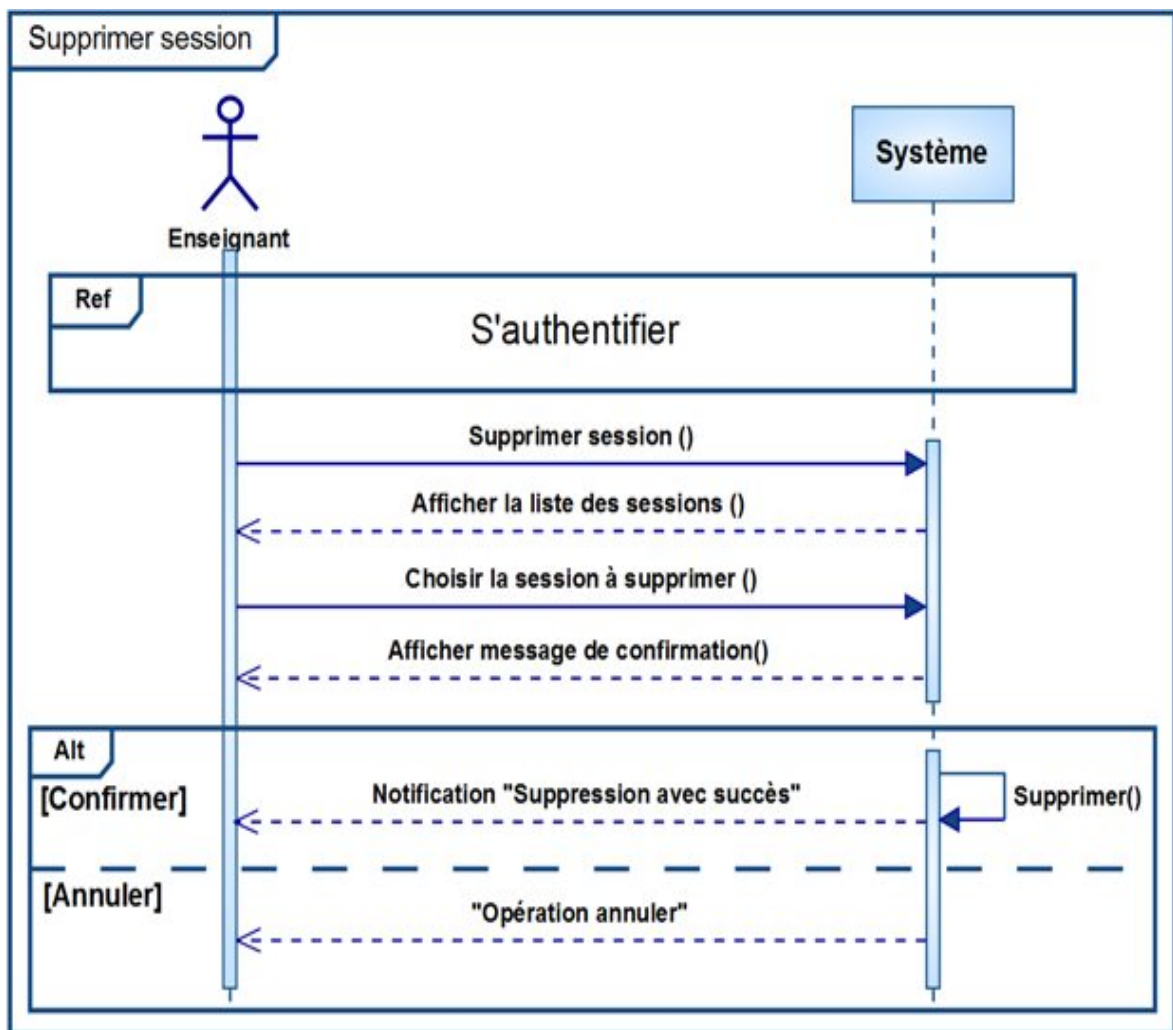


FIGURE 69 – Diagramme de séquence du cas Supprimer session

- « Gérer archive des tests »

- Description textuelle

Cas d'utilisation	Gérer archive des tests.
Acteur	Enseignant.
But	Ce cas permet à l'enseignant de consulter ou supprimer des tests de l'archive.
Pré condition	L'enseignant doit être authentifié, il est dans son espace.
Scénario nominal	L'enseignant choisit l'action à effectuer : 1-la consultation d'un test de l'archive(voir le cas Consulter test de l'archive). 4-la restauration d'un test de l'archive (voir le cas Restaurer test de l'archive).
Scénario optionnelle	• La liste des tests archivés est vide,le système affiche un message d'information « la liste est vide ».

TABLE 24: Description textuelle du cas d'utilisation

« Gérer archive des tests »

– Diagramme de séquence

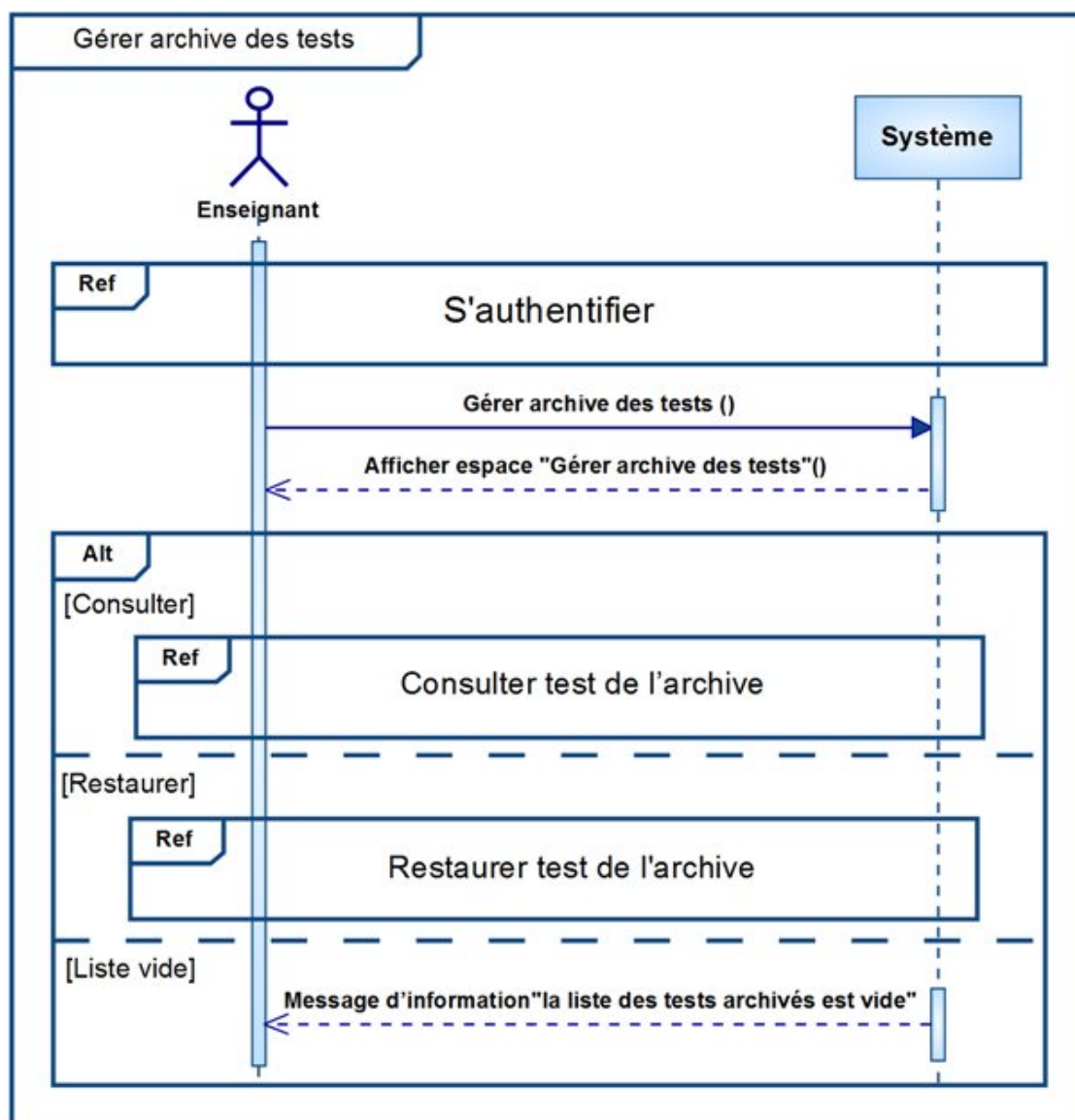


FIGURE 70 – Diagramme de séquence du cas Gérer archive des tests

- « Consulter test de l'archive »

- Description textuelle

Cas d'utilisation	Consulter test de l'archive.
Acteur	Enseignant.
But	Ce cas permet à l'enseignant consulter un test de l'archive.
Pré condition	L'enseignant est dans son espace,et il a choisi l'action « consulter test de l'archive ».
Poste condition	Le test archivé choisi est affiché.
Scénario nominal	1-Le système affiche la liste des tests archivés. 2-L'enseignant choisit un des tests affichés. 3-Le système affiche le test choisi.

TABLE 25: Description textuelle du cas d'utilisation
« Consulter test de l'archive »

- Diagramme de séquence

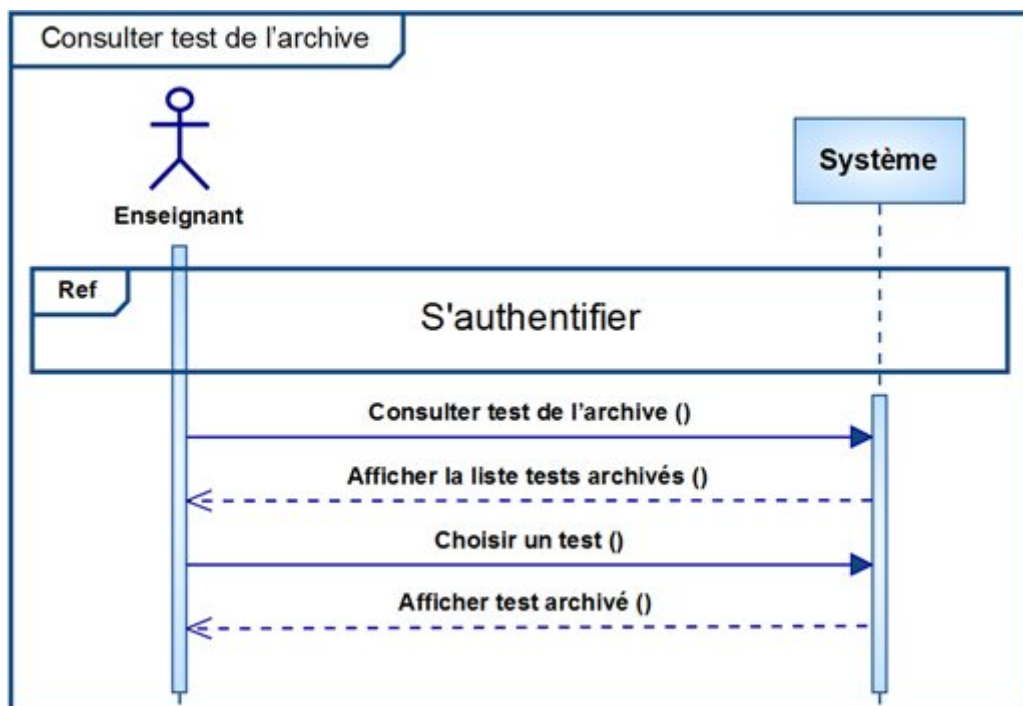


FIGURE 71 – Diagramme de séquence du cas Consulter test de l'archive

- « Restaurer test de l'archive »

- Description textuelle

Cas d'utilisation	Restaurer test de l'archive.
Acteur	Enseignant.
But	Ce cas permet à l'enseignant de restaurer un test de l'archive.
Pré condition	L'enseignant est dans son espace, et il a choisi l'action « Restaurer test de l'archive ».
Poste condition	Le test archivé est restauré.
Scénario nominal	1-L'enseignant consulte la liste des tests archivés et choisit le test à restaurer. 2-Le système affiche un message de confirmation de Restauration. 3-L'enseignant confirme la restauration. 4-Le système restaurer le test concerné. 5-Le système affiche une notification « Restauration avec succès »
Scénario optionnelle	• L'enseignant peut annuler l'opération lorsqu'il reçoit un message de confirmation.

TABLE 26: Description textuelle du cas d'utilisation

« Restaurer test de l'archive »

– Diagramme de séquence

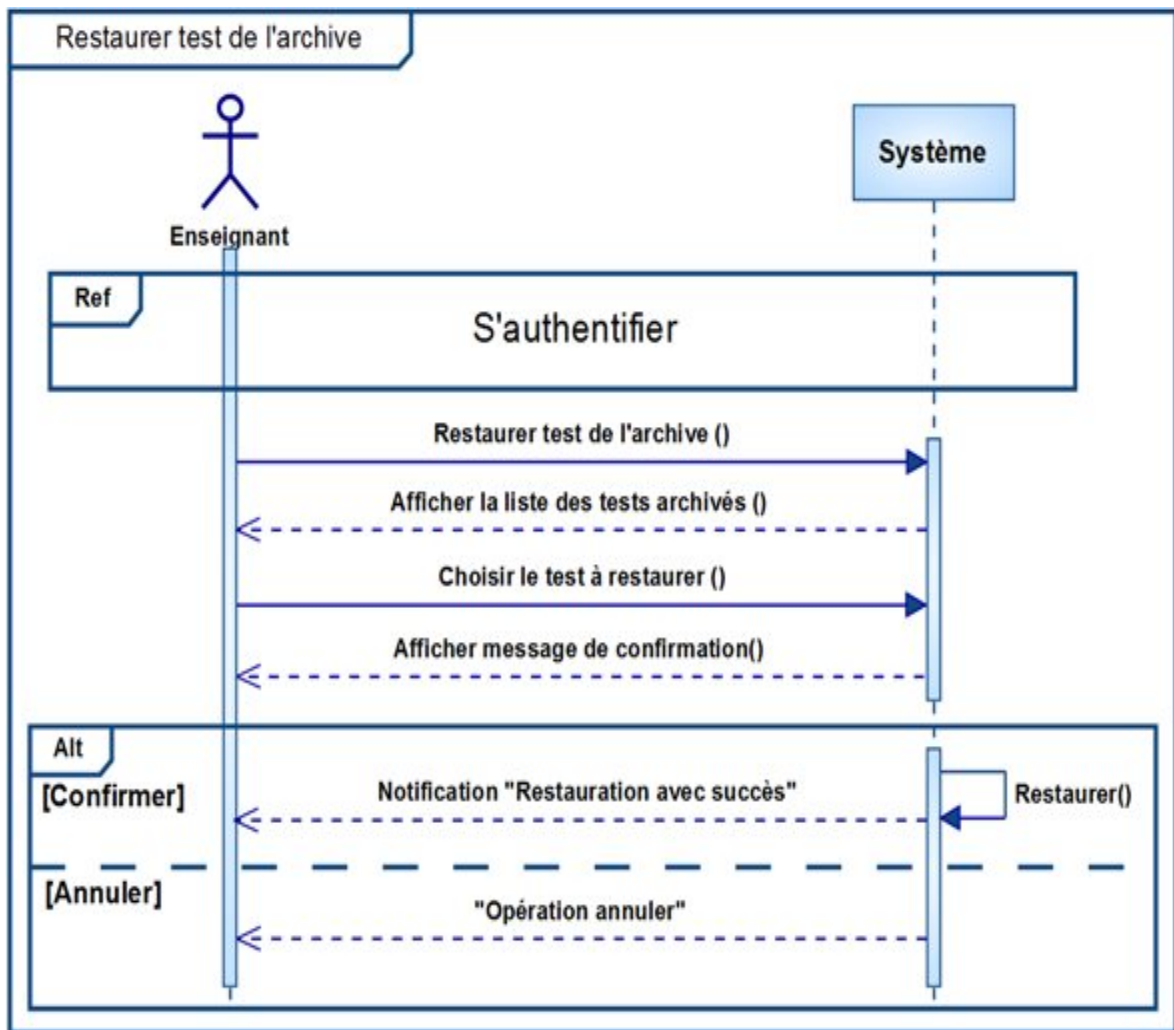


FIGURE 72 – Diagramme de séquence du cas Restaurer test de l'archive

- « Gérer éditeur de style de question »

- Description textuelle

Cas d'utilisation	Gérer éditeur de style de question.
Acteur	Agent de MAJ de style question.
But	Ce cas permet à l'agent de consulter, ajouter, modifier ou supprimer des styles des questions.
Pré condition	L'agent doit être authentifié, il est dans son espace.
Scénario nominal	L'agent choisit l'action à effectuer : 1-La consultation d'un éditeur (voir le cas Consulter éditeur). 2-L'ajout d'un éditeur (voir le cas Ajouter éditeur). 3-La modification des informations d'un éditeur(voir le cas Modifier éditeur). 4-La suppression d'un éditeur (voir le cas Supprimer éditeur).
Scénario optionnelle	• La liste des éditeurs est vide,le système affiche un message d'information « la liste est vide ».

TABLE 27: Description textuelle du cas d'utilisation

« Gérer éditeur de style de question »

– Diagramme de séquence

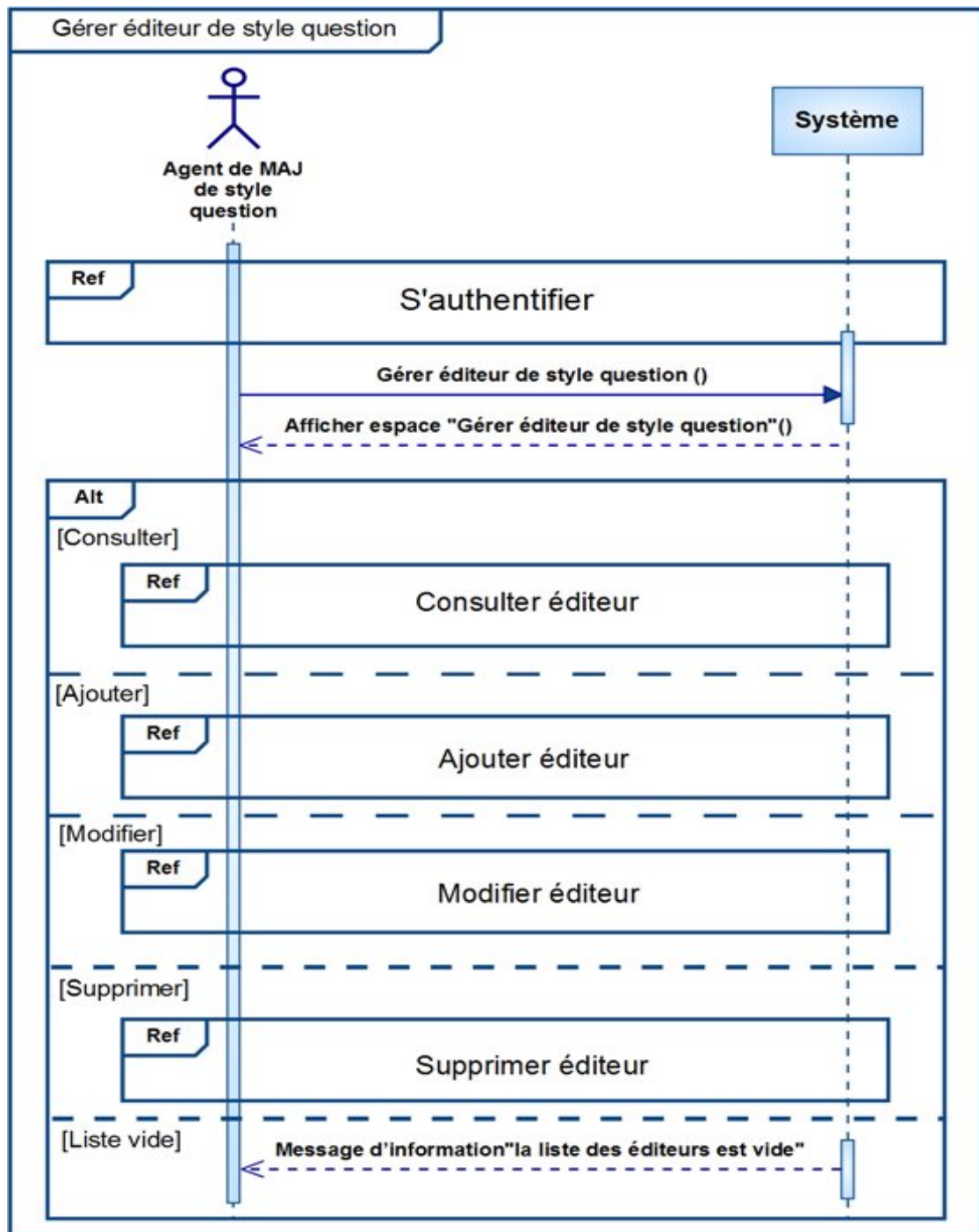


FIGURE 73 – Diagramme de séquence du cas Gérer éditeur de style de question

- « Consulter éditeur »

- Description textuelle

Cas d'utilisation	Consulter éditeur.
Acteur	Agent de MAJ de style question.
But	Ce cas permet à l'agent de consulter un éditeur de style question.
Pré condition	L'agent est dans son espace, et il a choisi l'action « Consulter éditeur ».
Poste condition	L'éditeur de style question choisi est affiché.
Scénario nominal	1-Le système affiche la liste des éditeurs . 2-L'agent choisit un des éditeurs affichés. 3-Le système affiche l'éditeur choisi.

TABLE 28: Description textuelle du cas d'utilisation

- « Consulter éditeur »

- Diagramme de séquence

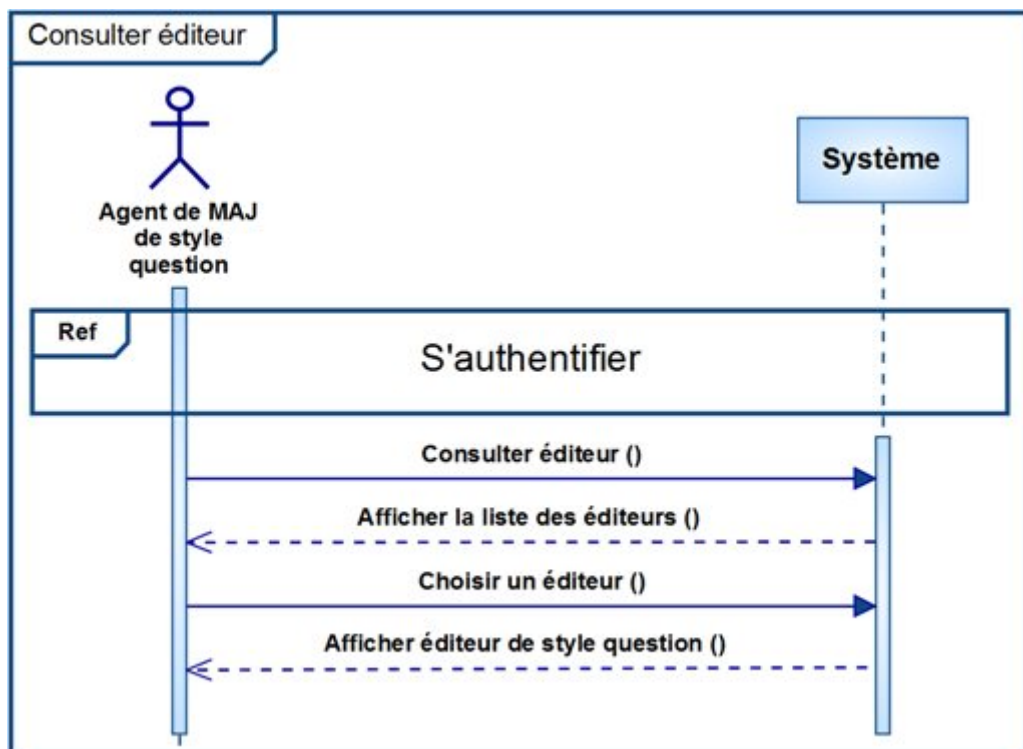


FIGURE 74 – Diagramme de séquence du cas d'utilisation Consulter éditeur

- « Supprimer éditeur »

- Description textuelle

Cas d'utilisation	Supprimer éditeur.
Acteur	Agent de MAJ de style question.
But	Ce cas permet à l'agent de supprimer un éditeur de style question.
Pré condition	L'agent est dans son espace, et il a choisi l'action « supprimer éditeur ».
Poste condition	L'éditeur de style question est supprimé.
Scénario nominal	1-L'agent consulte la liste des éditeurs et choisit l'éditeur à supprimer. 2-Le système affiche un message de confirmation de suppression 3-L'agent confirme la suppression. 4-Le système supprime l'éditeur concerné. 5-Le système affiche une notification «Suppression avec succès ».
Scénario optionnelle	• L'agent peut annuler l'opération lorsqu'il reçoit un message de confirmation.

TABLE 29: Description textuelle du cas d'utilisation

« Supprimer éditeur »

– Diagramme de séquence

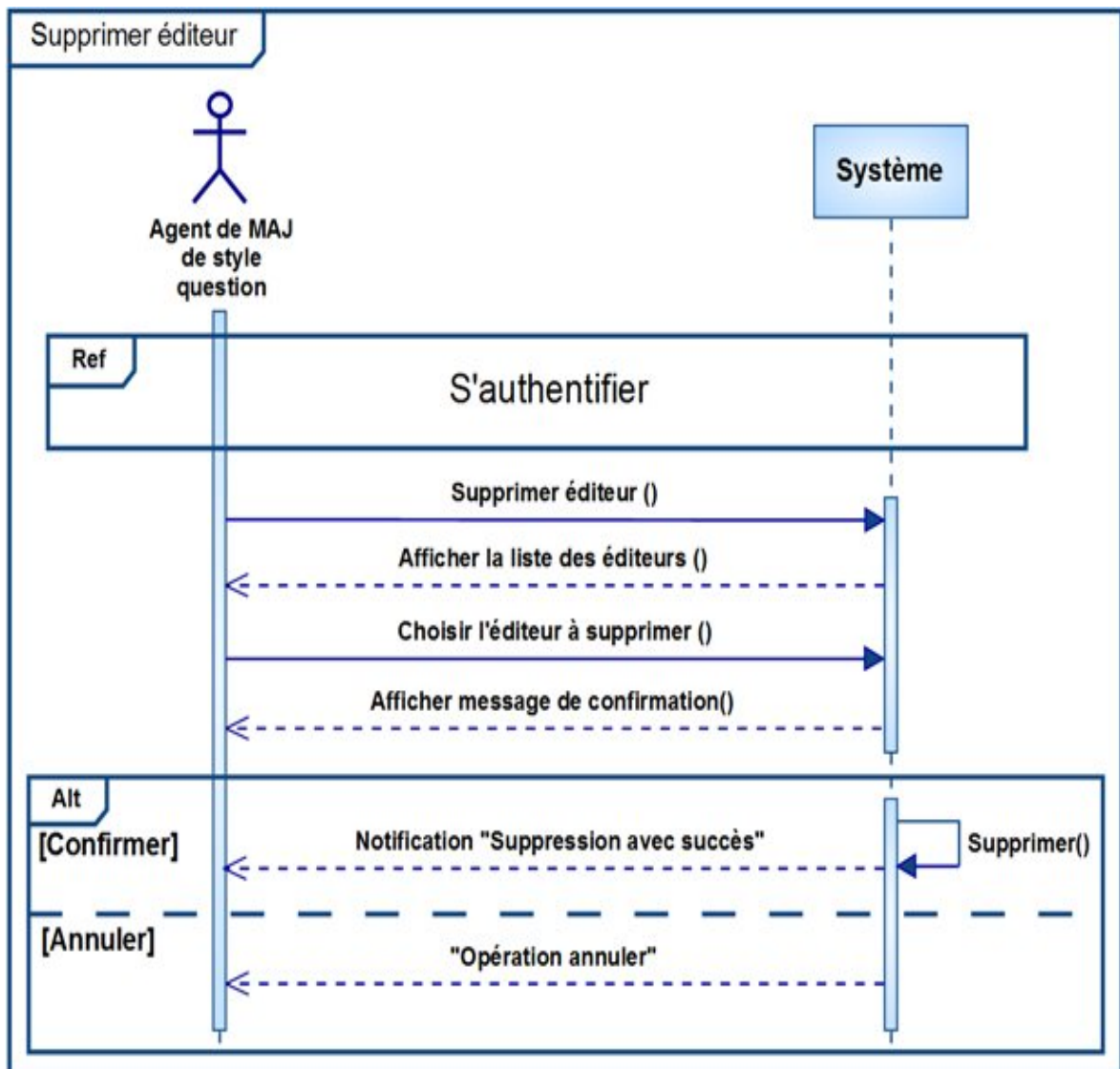


FIGURE 75 – Diagramme de séquence du cas Supprimer éditeur

- « Consulter archive des tests »

- Description textuelle

Cas d'utilisation	Consulter archive des tests.
Acteur	Etudiant (principal), Plateforme e-learning (secondaire).
But	Ce cas permet d'orienter l'étudiant vers les tests archivés dans la plateforme e-learning.
Pré condition	L'étudiant doit être authentifié.
Poste condition	La redirection vers les tests archivés dans la plateforme e-learning est effectuée.
Scénario nominal	1-L'étudiant demande de consulter l'archive des tests. 2-Le système affiche un message d'orientation vers la plateforme e-learning. 3-L'étudiant confirme l'opération 4-Le système redirige l'étudiant vers les tests archivés dans la plateforme e-learning. 5-Le système affiche une notification «Orientation avec succès».
Scénario optionnelle	•L'étudiant peut annuler l'opération lorsqu'il reçoit un message d'orientation.

TABLE 30: Description textuelle du cas d'utilisation

« Consulter archive des tests »

– Diagramme de séquence

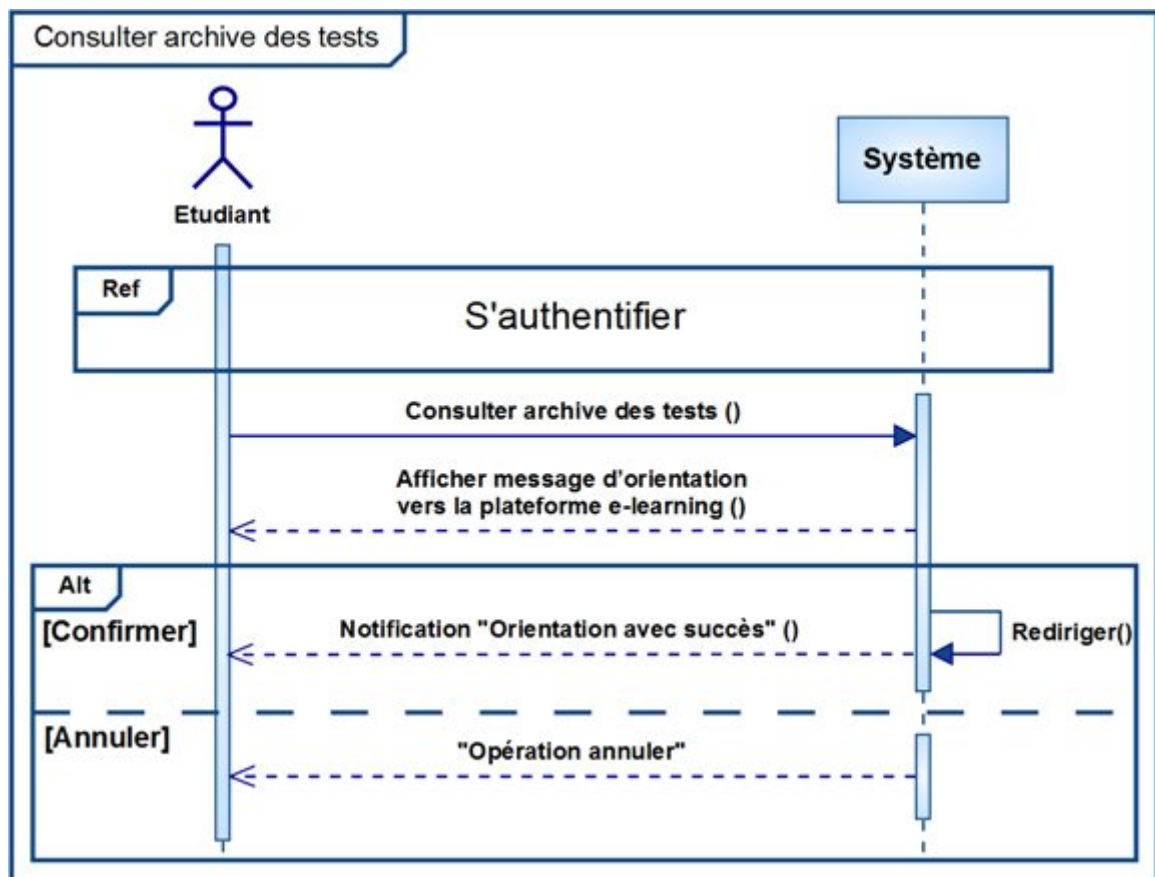


FIGURE 76 – Diagramme de séquence du cas Consulter archive des tests

Diagrammes de classes participantes :

Les figures suivantes représentent les diagrammes de classes participantes relia à chaque cas d'utilisation.

- « Modifier profil »

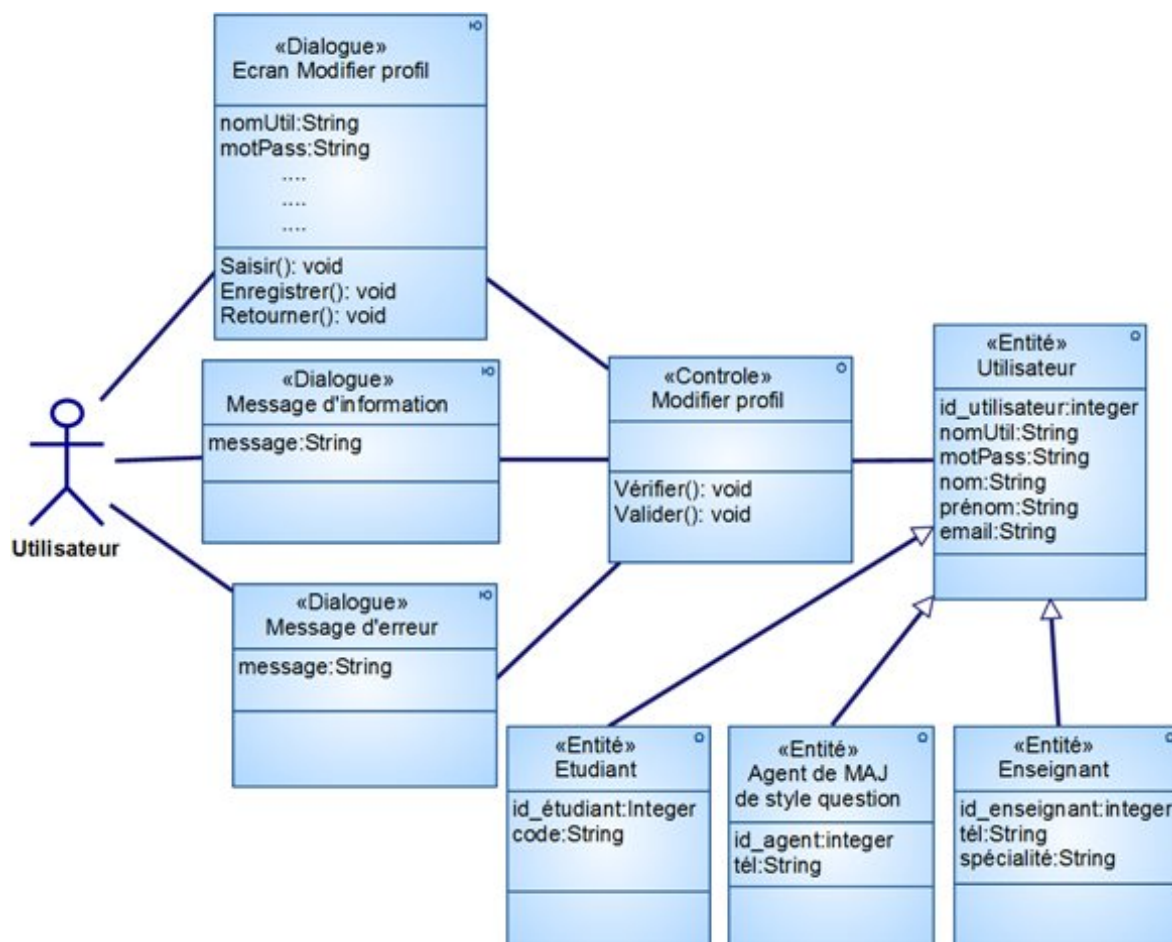


FIGURE 77 – Diagramme de classes participantes du cas Modifier profil

- « Gérer la liste des groupes enseignés »

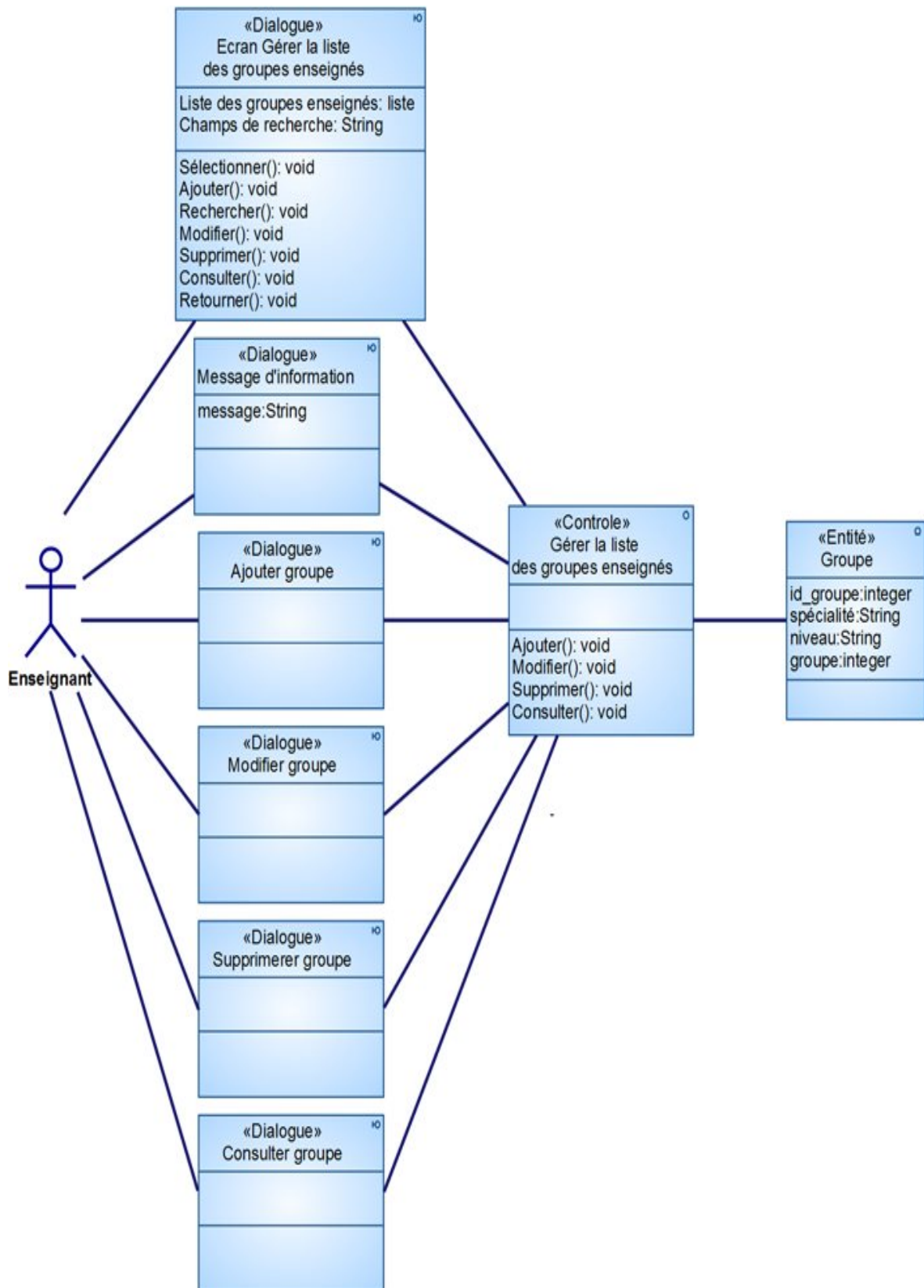


FIGURE 78 – Diagramme de classes participantes du cas Gérer la liste des groupe enseignés

- « Consulter groupe »

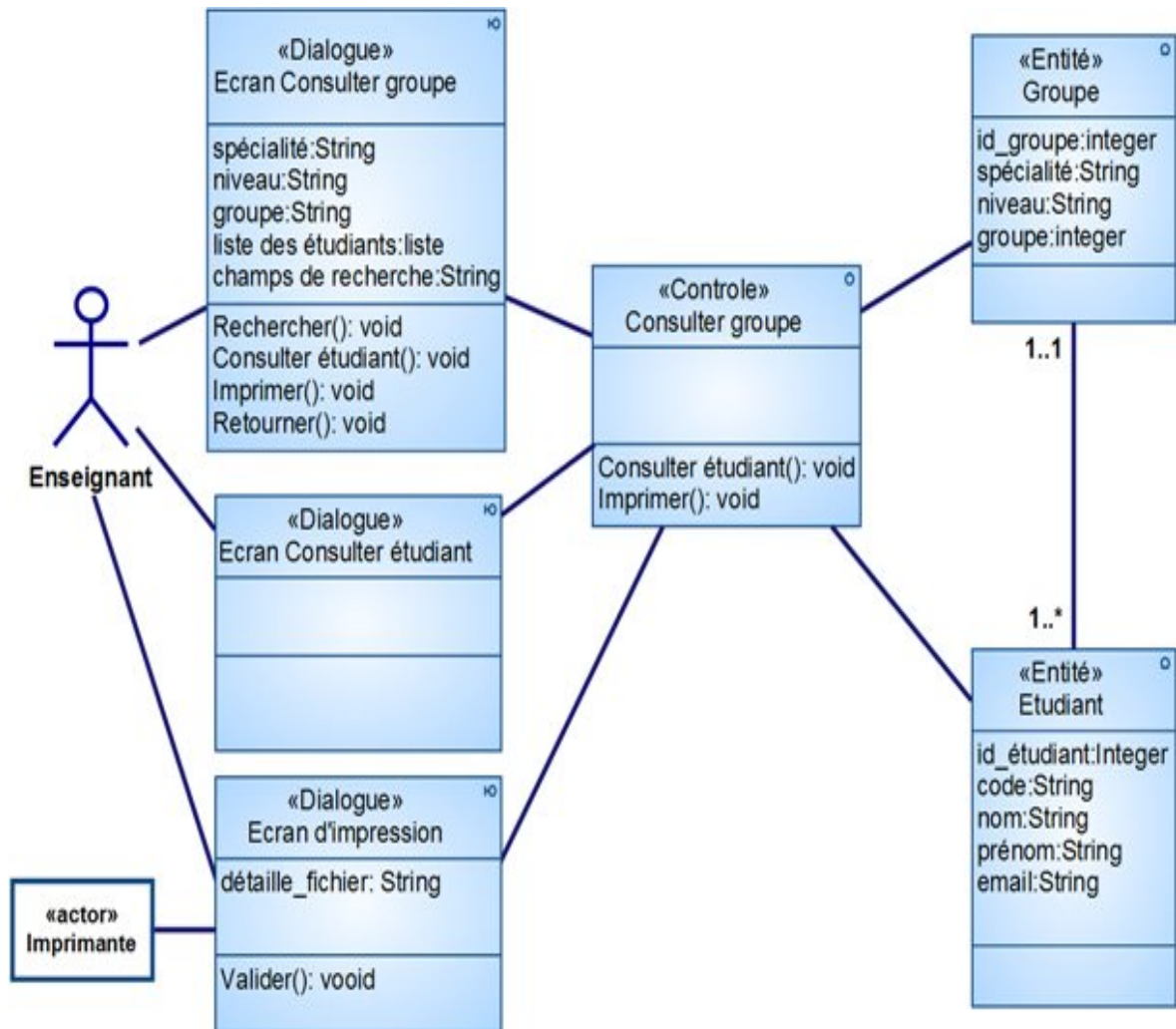


FIGURE 79 – Diagramme de classes participantes du cas Consulter groupe

- « Modifier groupe »

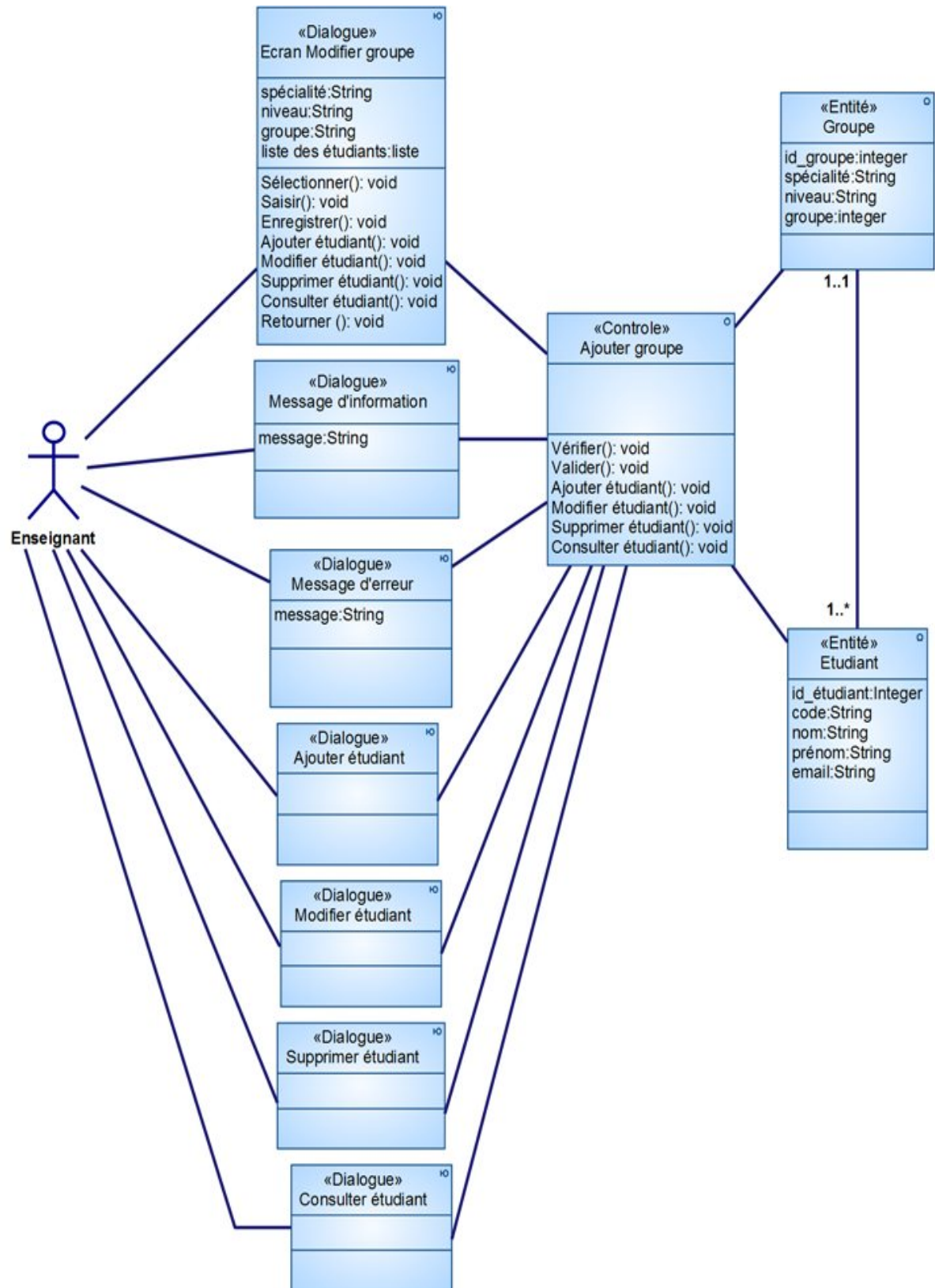


FIGURE 80 – Diagramme de classes participantes du cas Modifier groupe

- « Supprimer groupe »

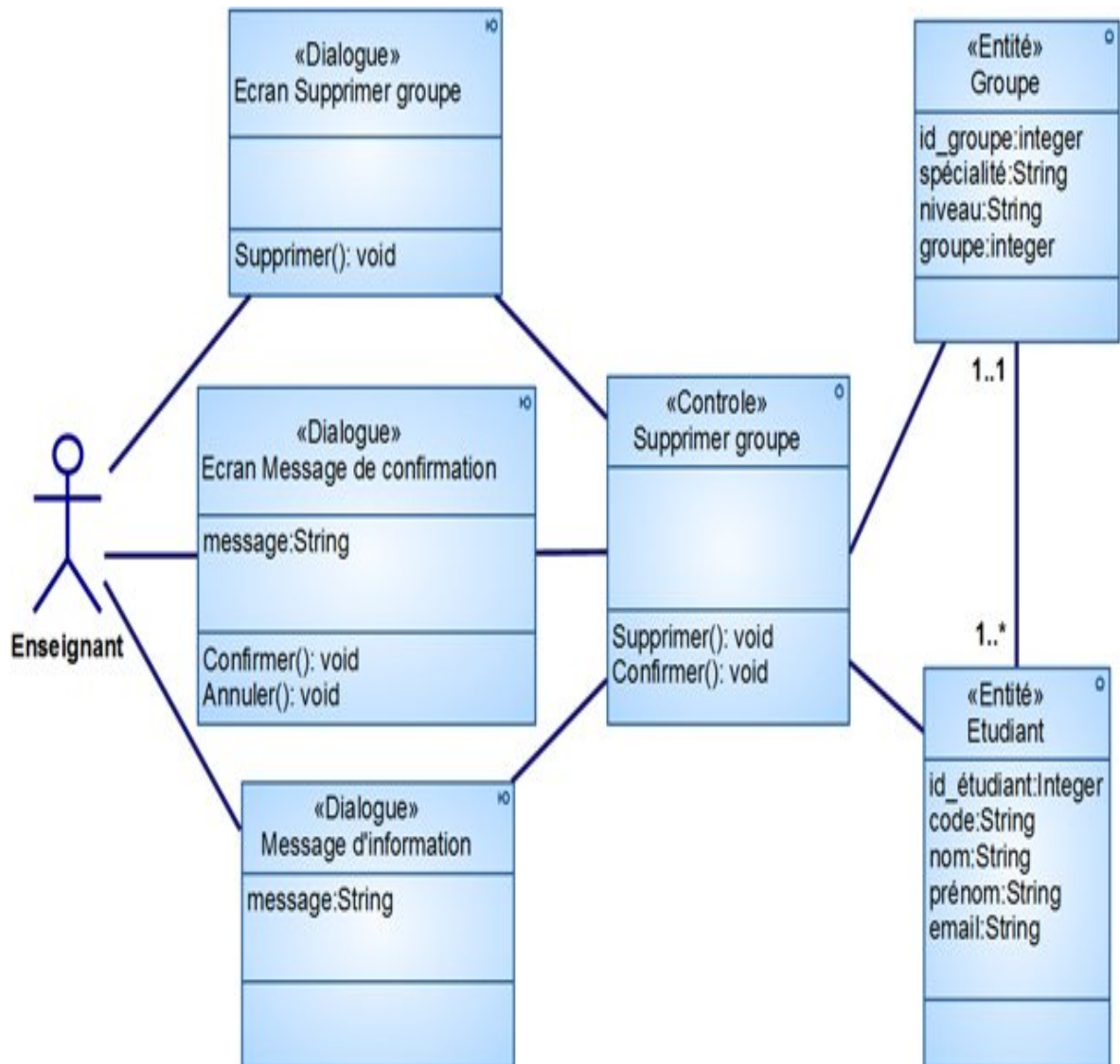


FIGURE 81 – Diagramme de classes participantes du cas Supprimer groupe

• « Gérer étudiants »

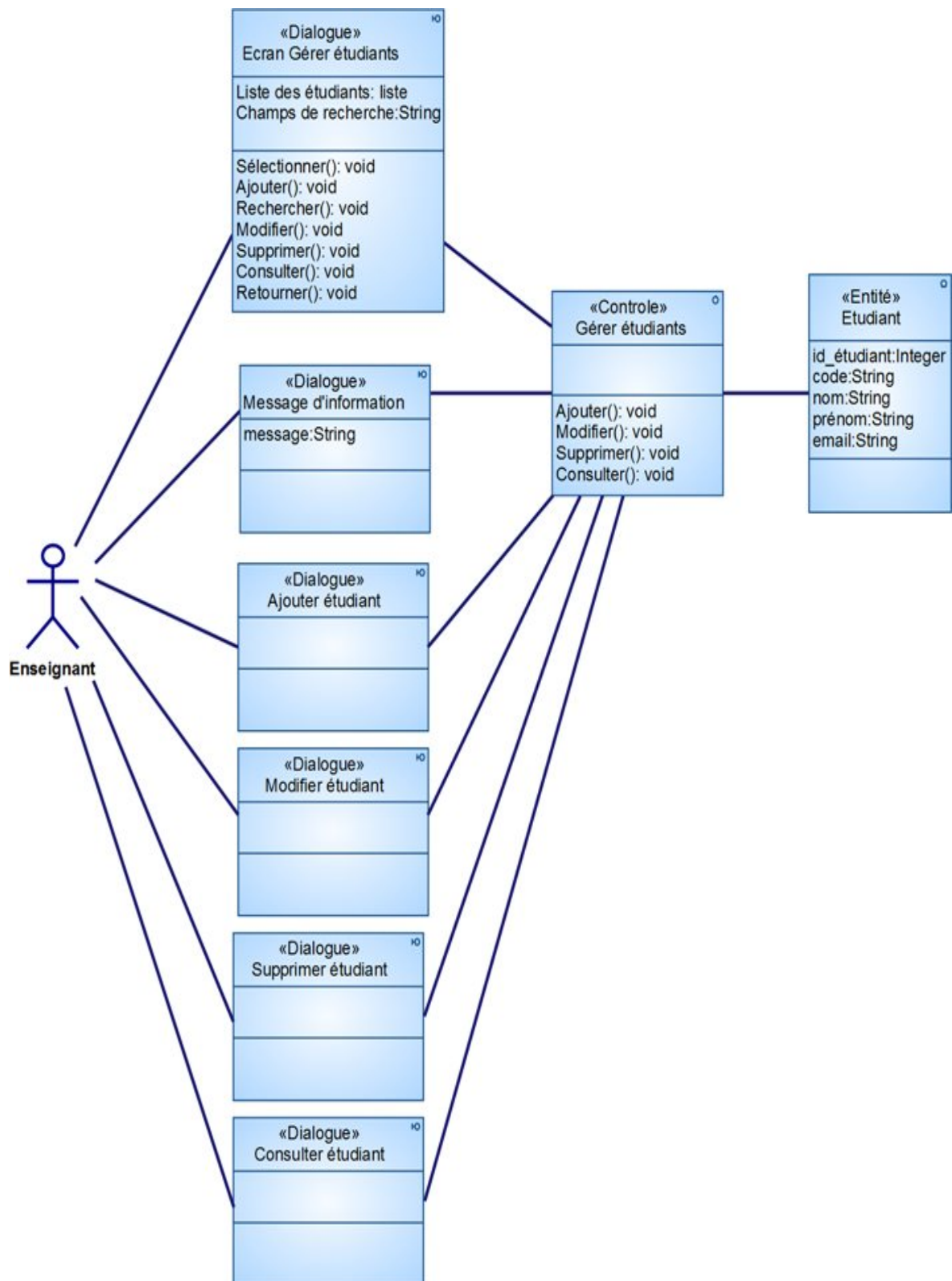


FIGURE 82 – Diagramme de classes participantes du cas Gérer étudiants

- « Consulter étudiant »

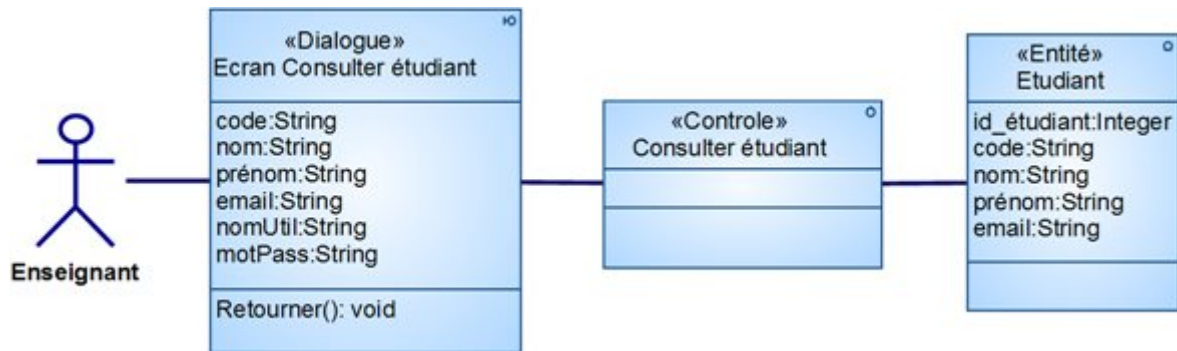


FIGURE 83 – Diagramme de classes participantes du cas Consulter étudiant

- « Modifier étudiant »

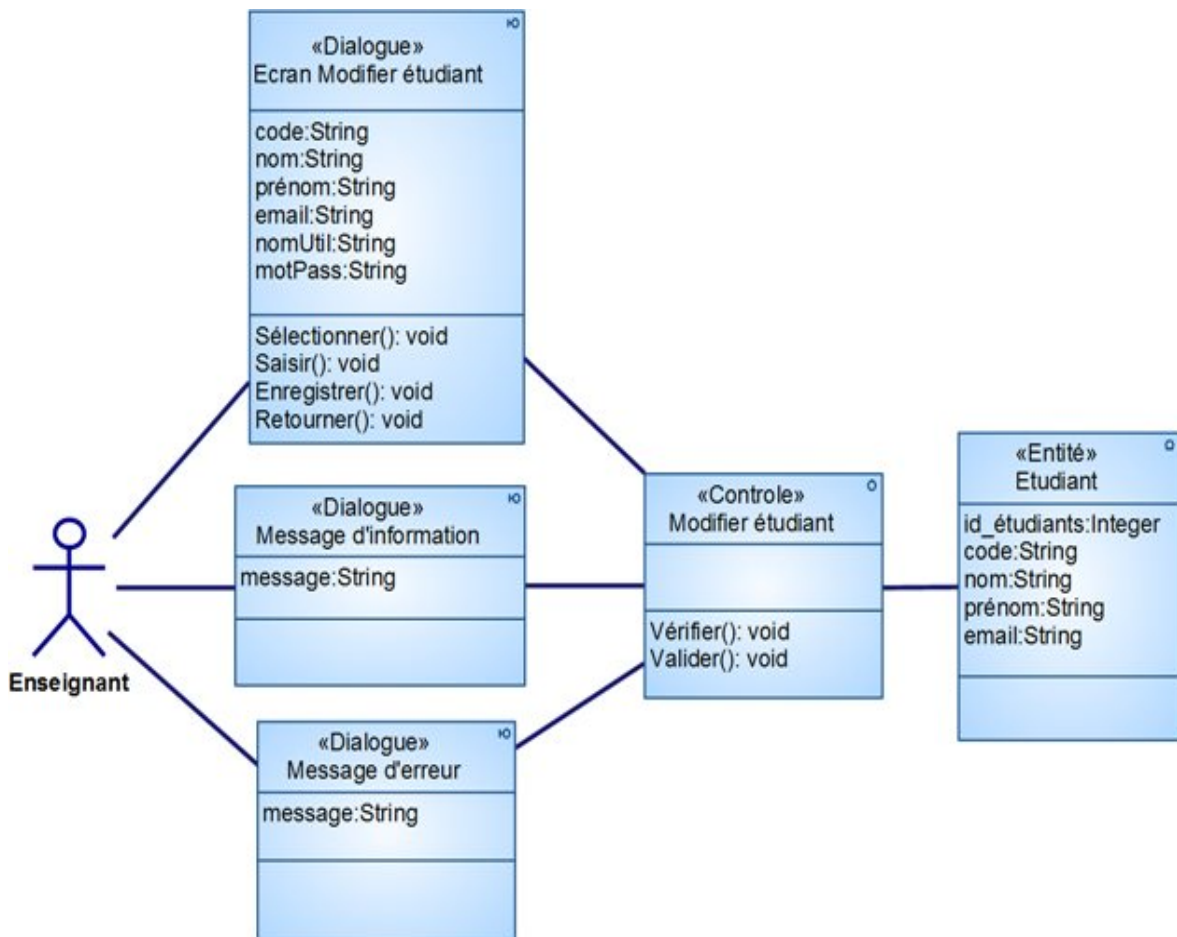


FIGURE 84 – Diagramme de classes participantes du cas Modifier étudiant

- « Supprimer étudiant »

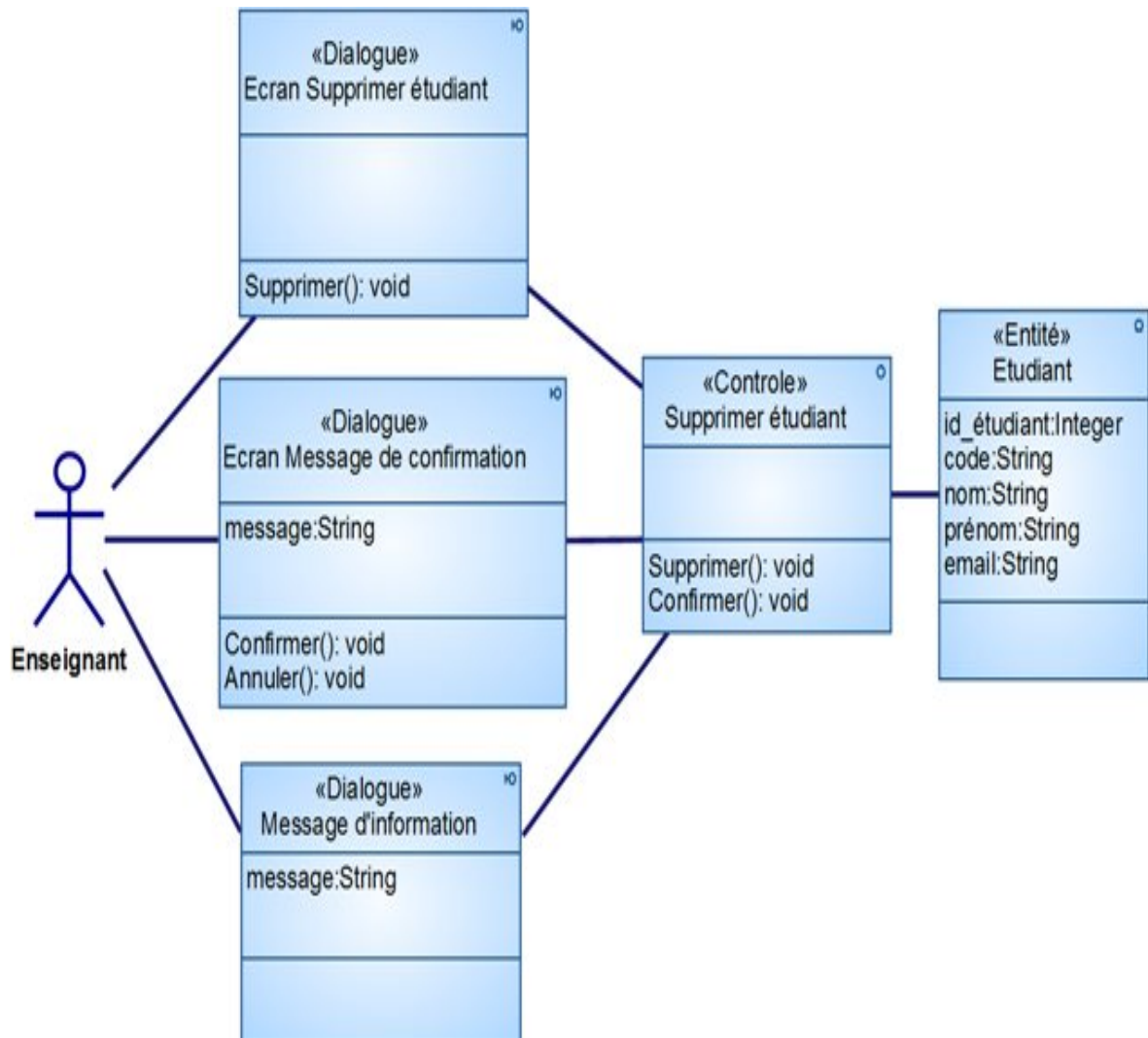


FIGURE 85 – Diagramme de classes participantes du cas Supprimer étudiant

• « Gérer tests et corrigés type »

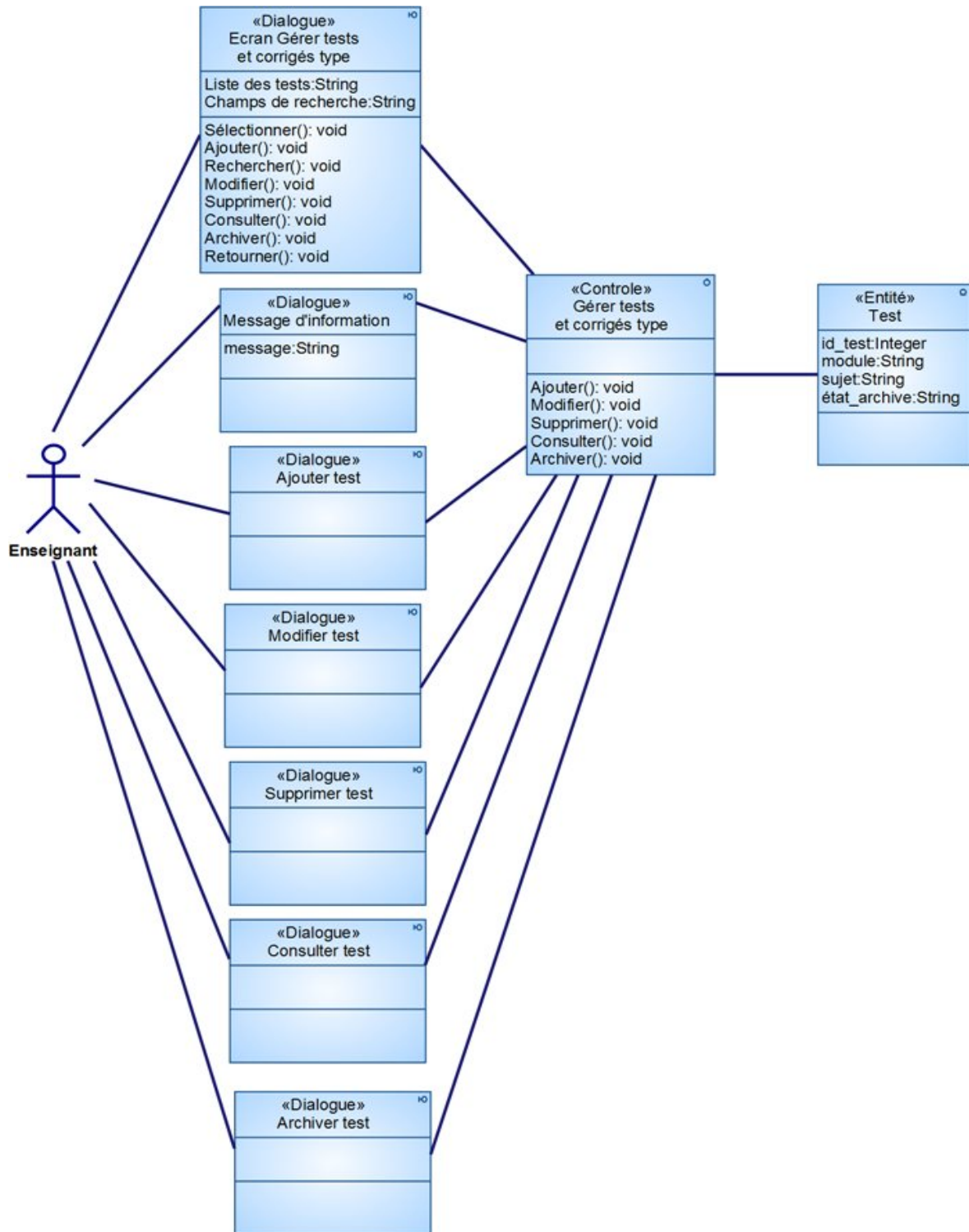


FIGURE 86 – Diagramme de classes participantes du cas Gérer tests et corrigés type

- « Modifier test et corrigés type »

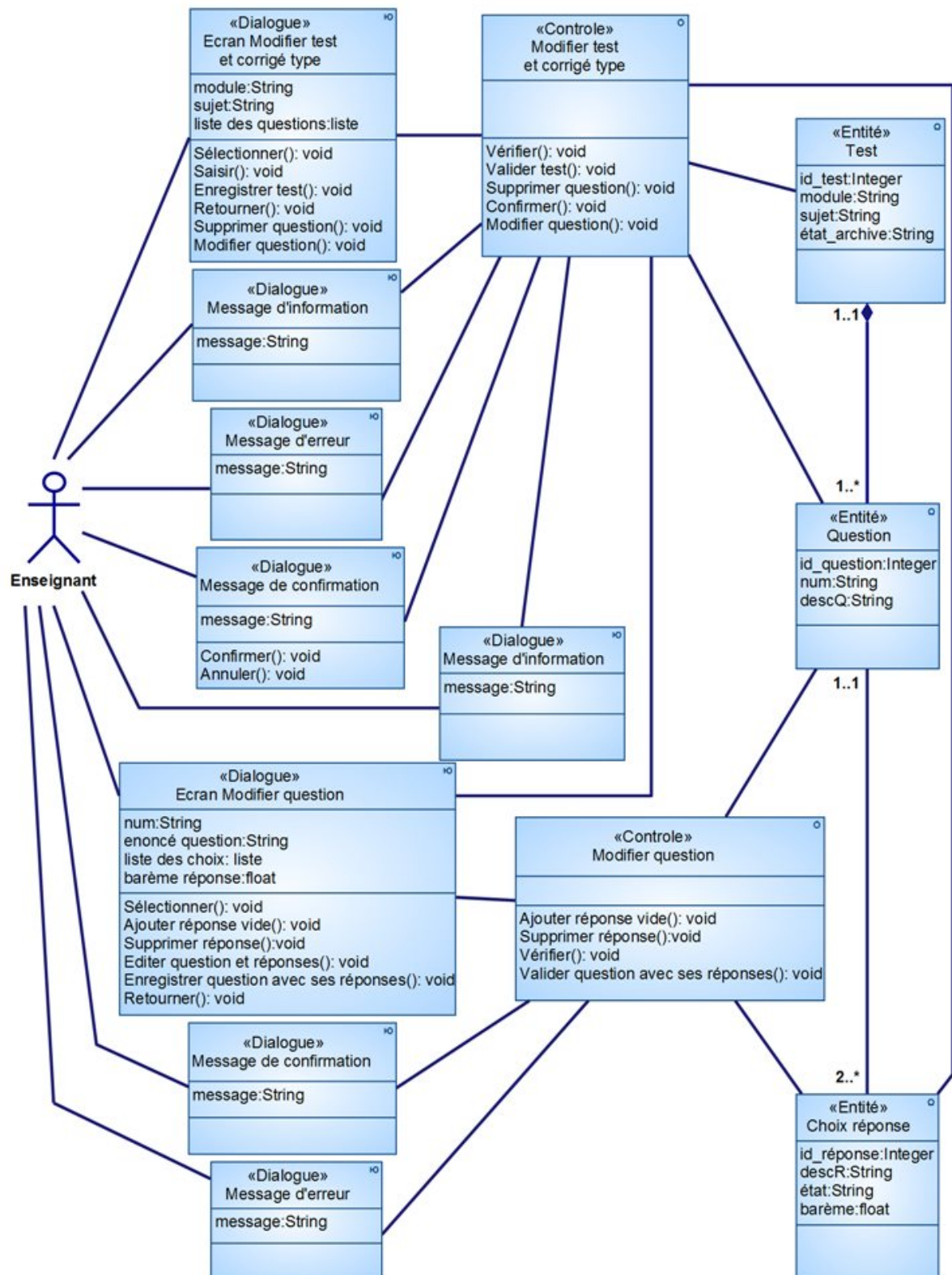


FIGURE 87 – Diagramme de classes participantes du cas Modifier test et corrigé type

- « Supprimer test et corrigés type »

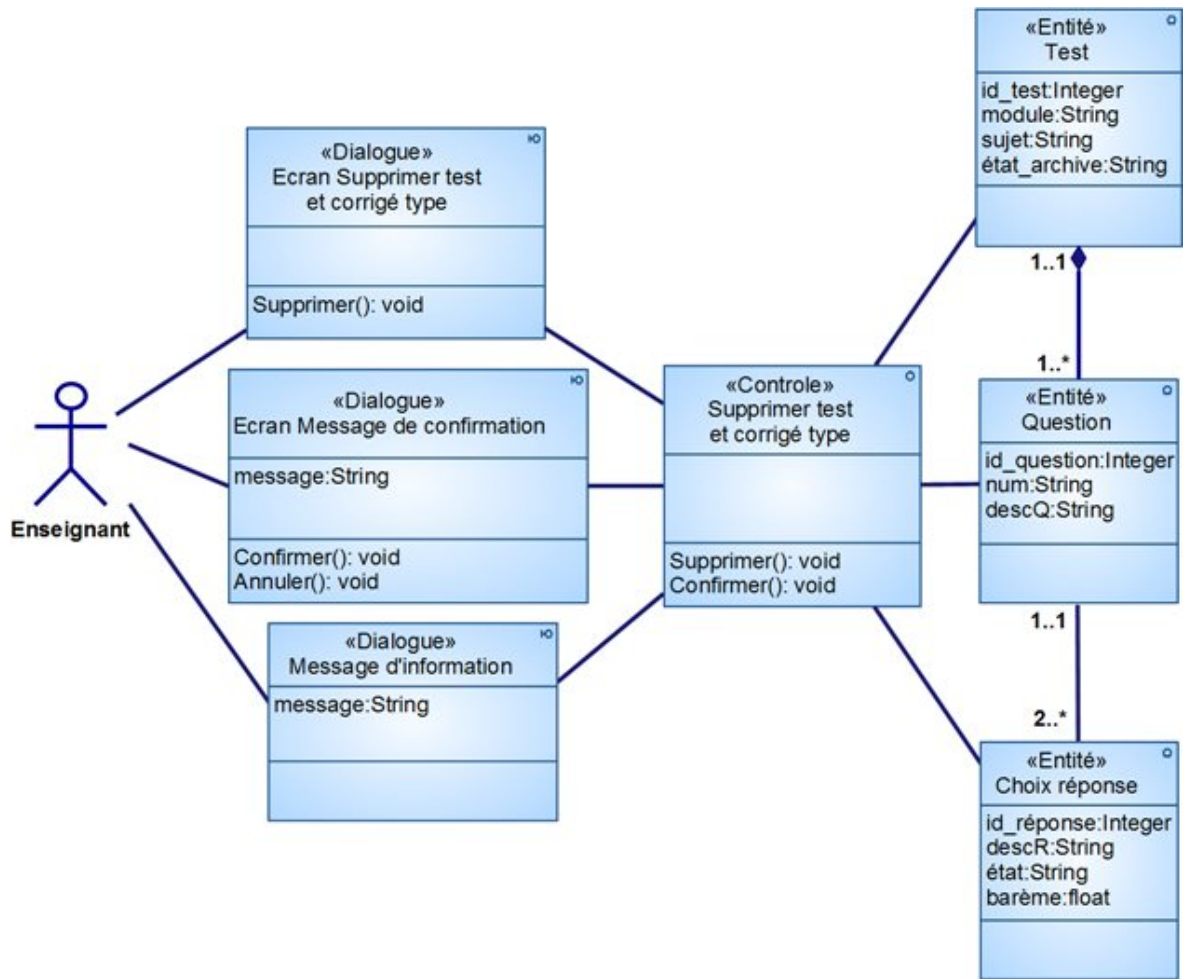


FIGURE 88 – Diagramme de classes participantes du cas Supprimer test et corrigé type

- « Archiver test et corrigés type »

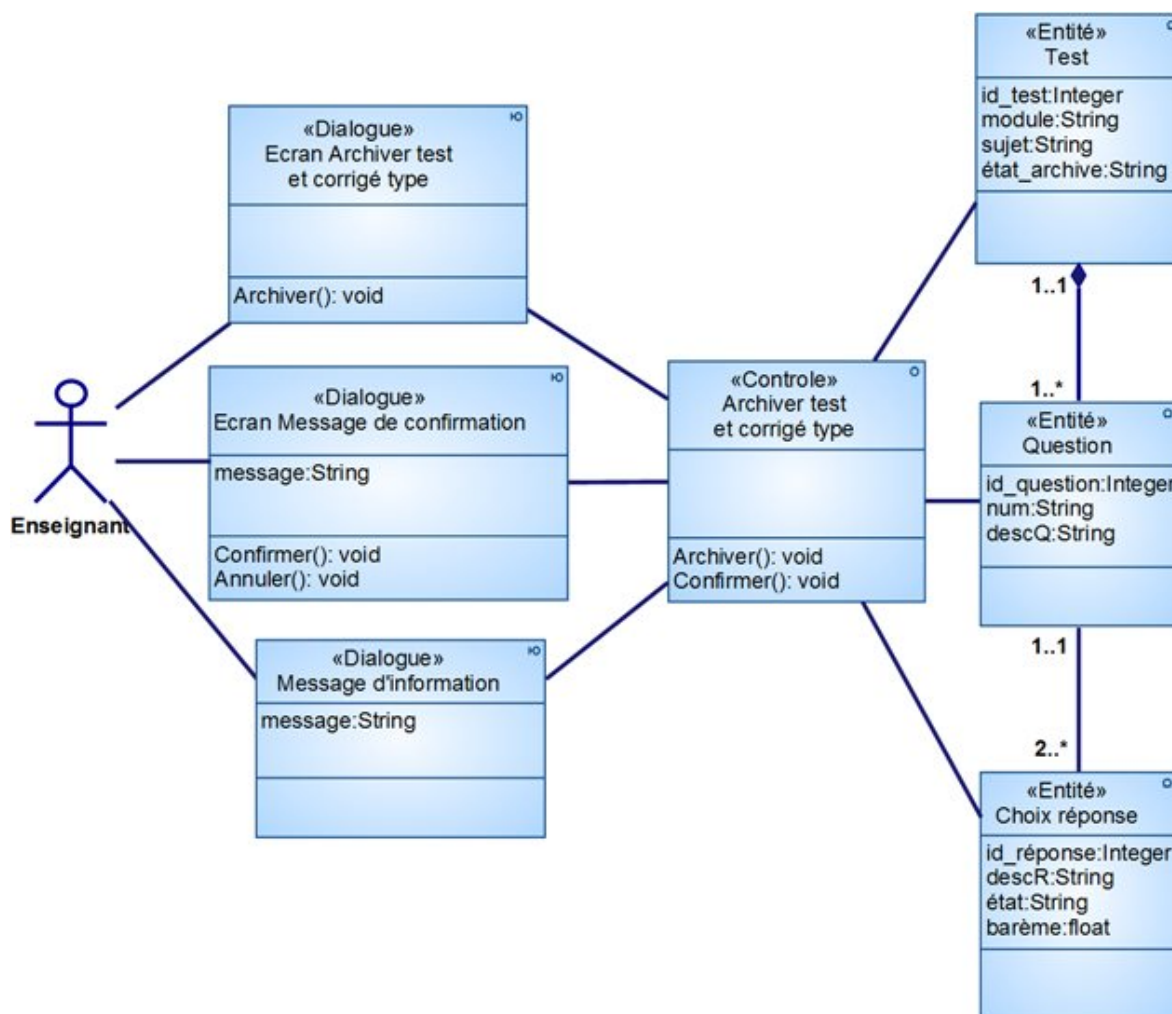


FIGURE 89 – Diagramme de classes participantes du cas Archiver test et corrigé type

- « Gérer session »

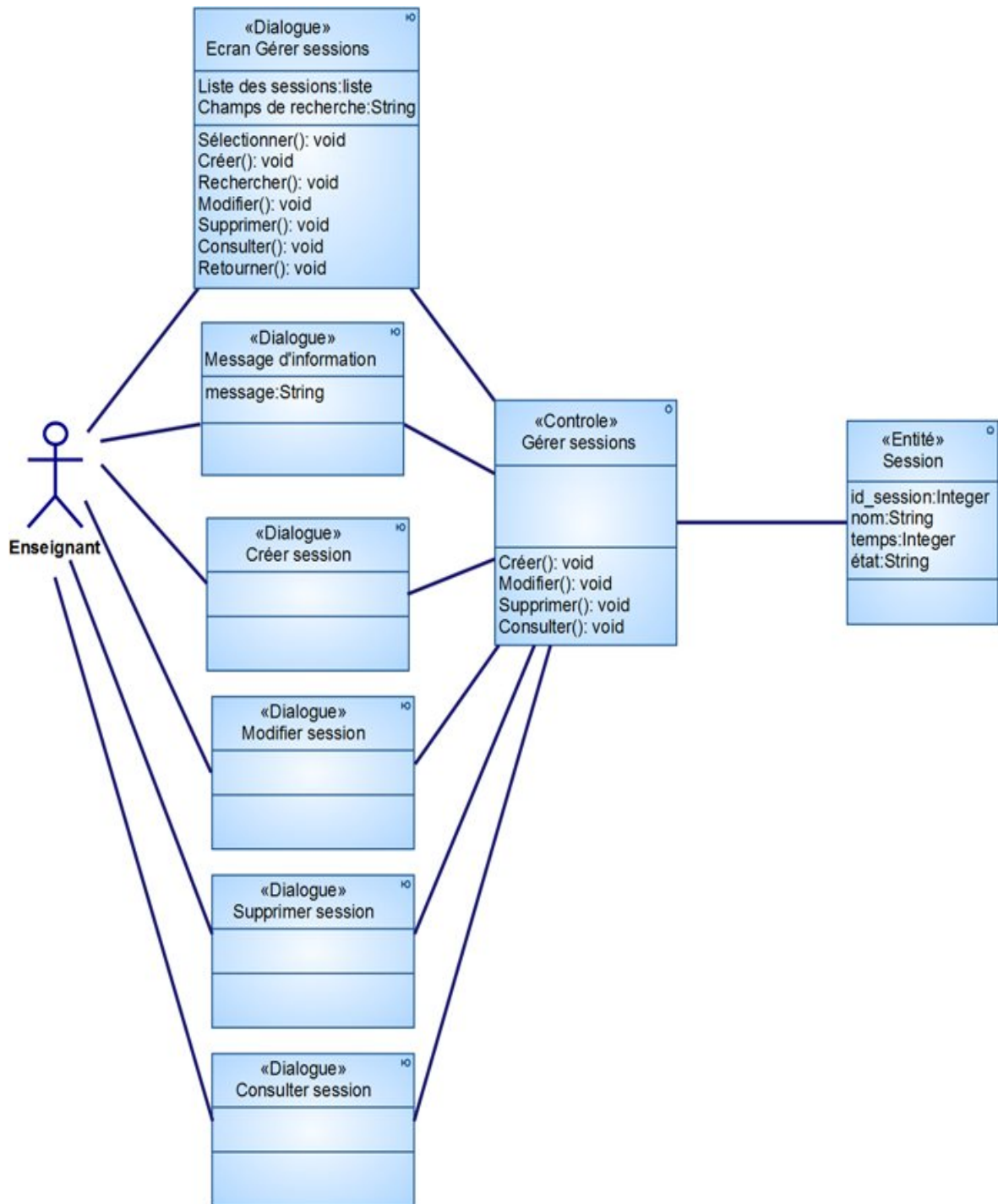


FIGURE 90 – Diagramme de classes participantes du cas Gérer session

- « Modifier session »

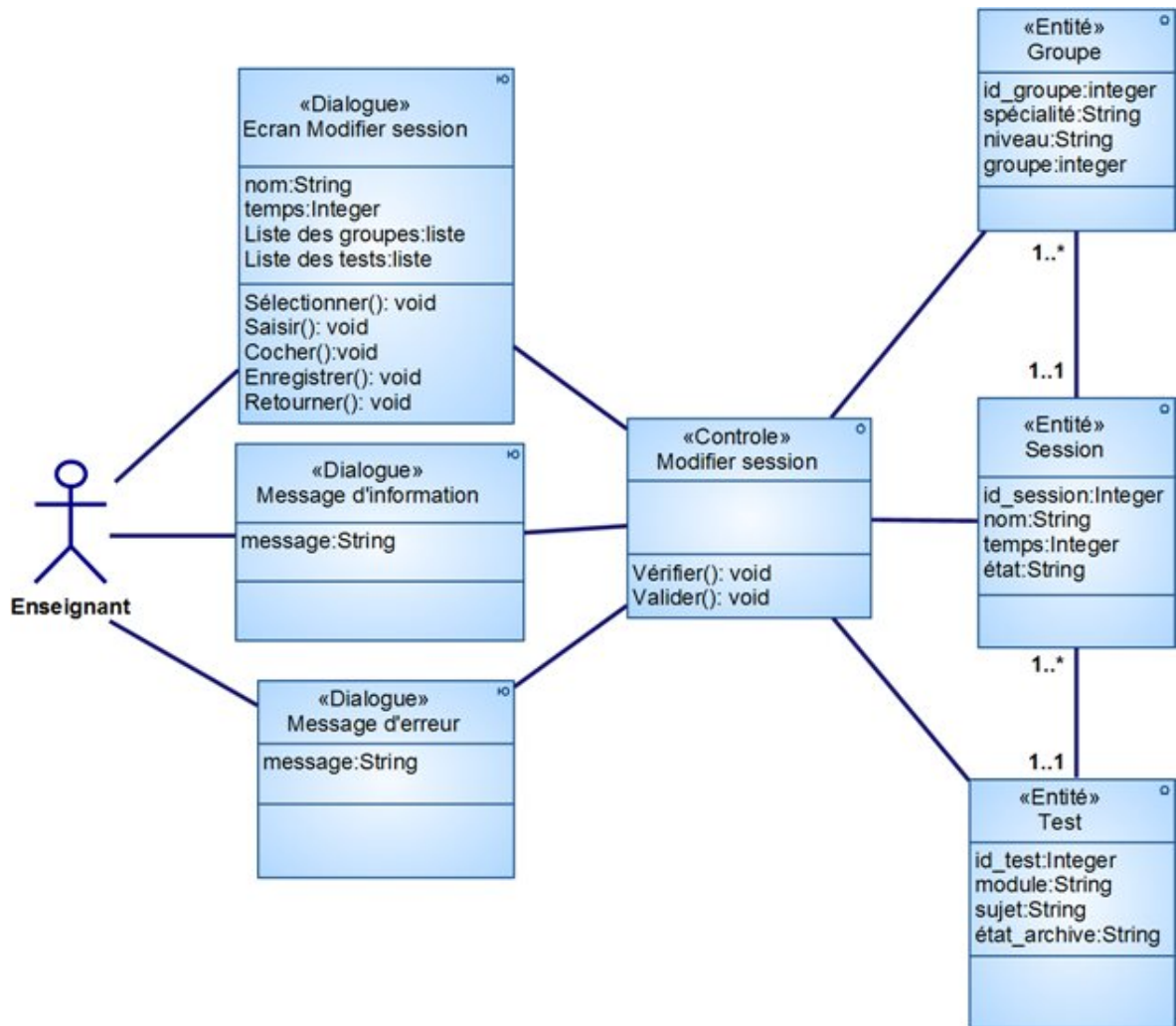


FIGURE 91 – Diagramme de classes participantes du cas Modifier session

- « Supprimer session »

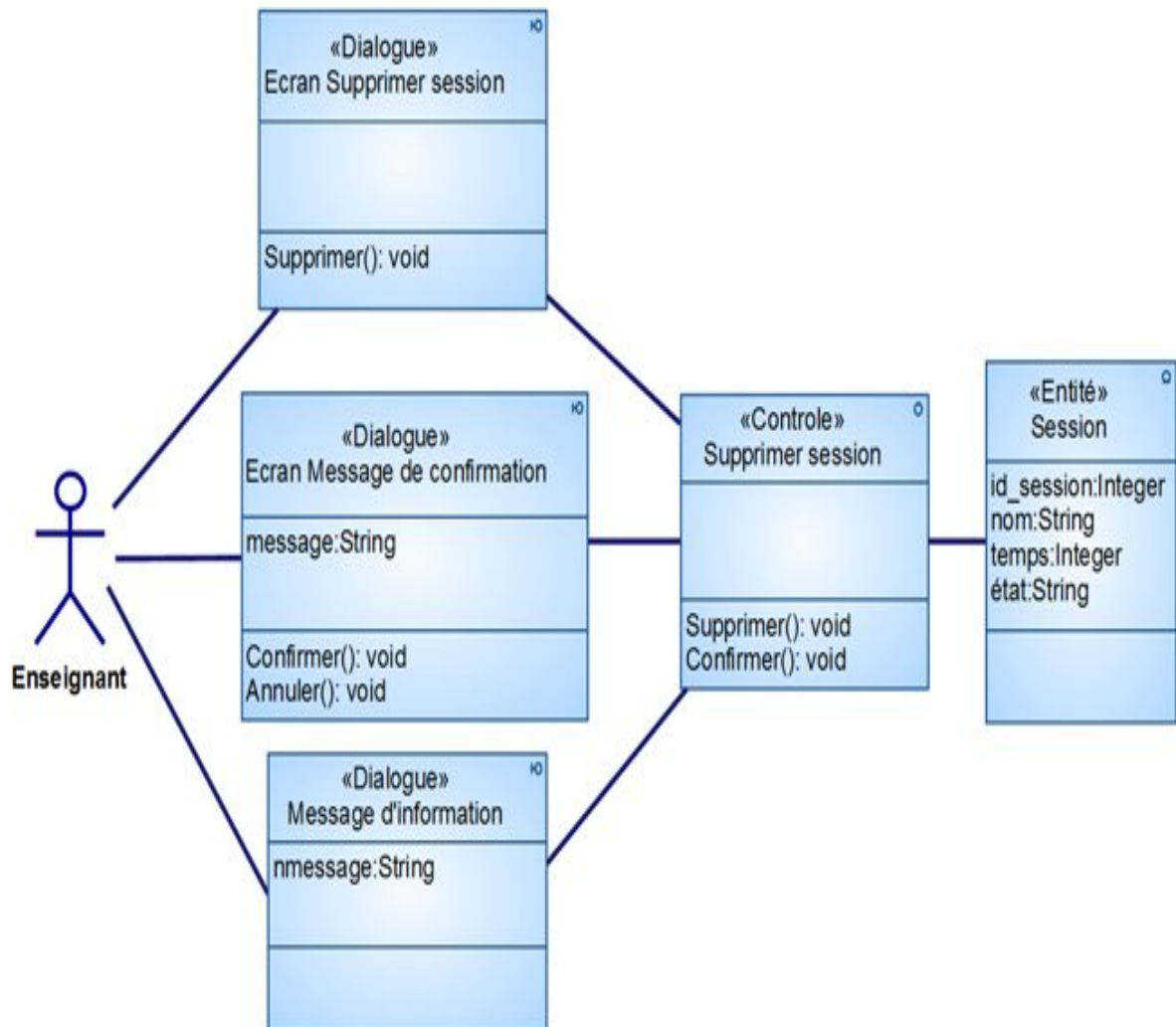


FIGURE 92 – Diagramme de classes participantes du cas Supprimer session

- « Gérer archive des tests »

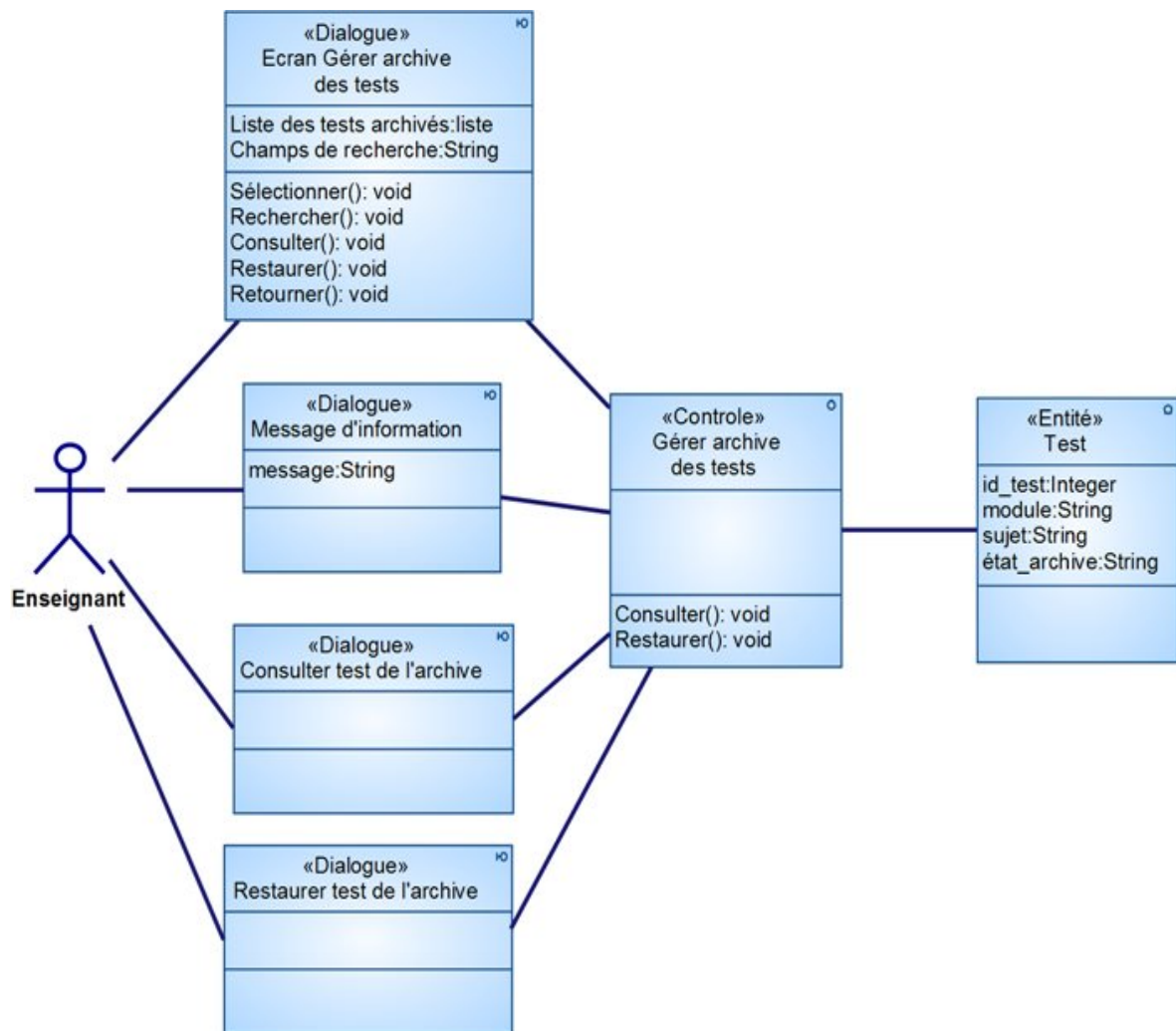


FIGURE 93 – Diagramme de classes participantes du cas Gérer archive des tests

- « Consulter test de l'archive »

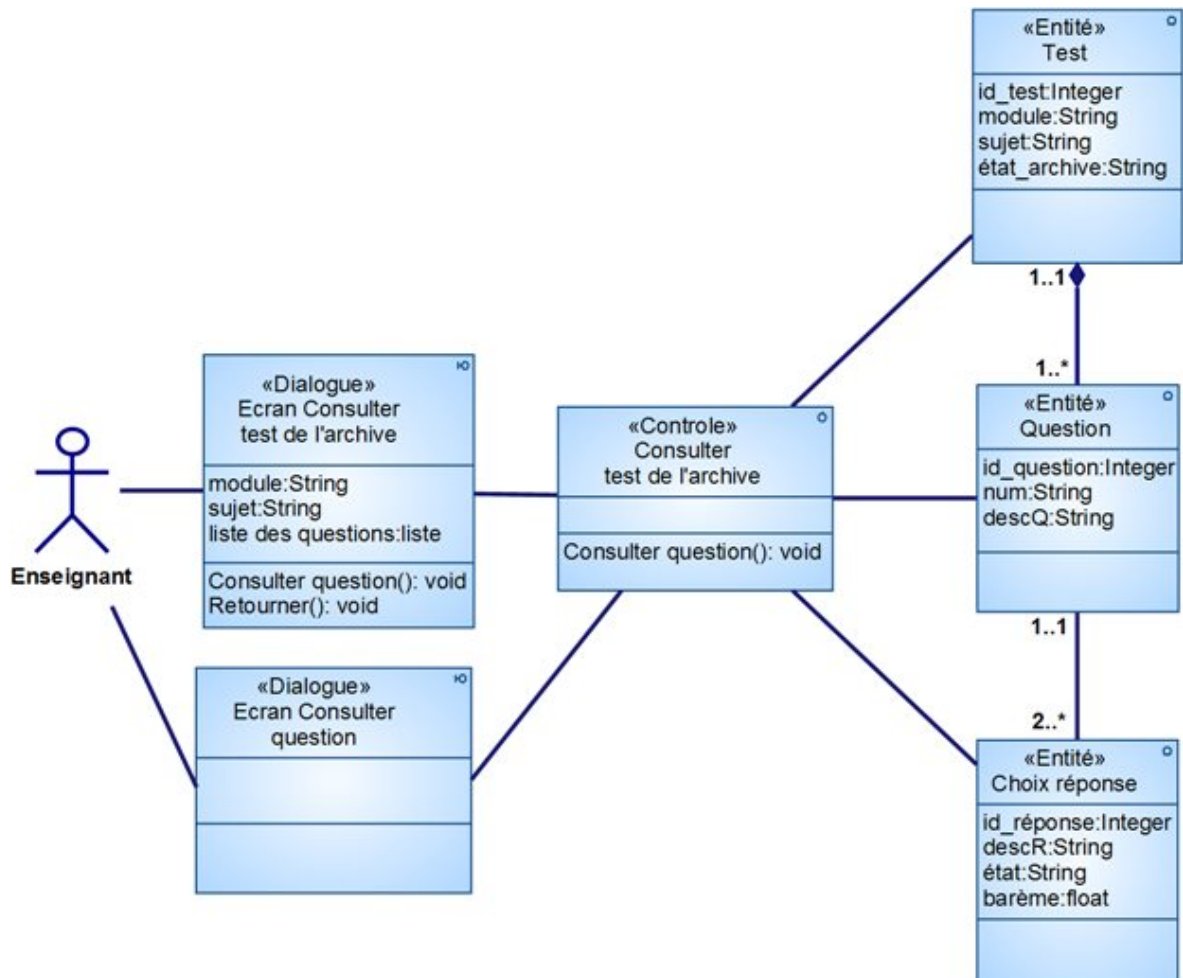
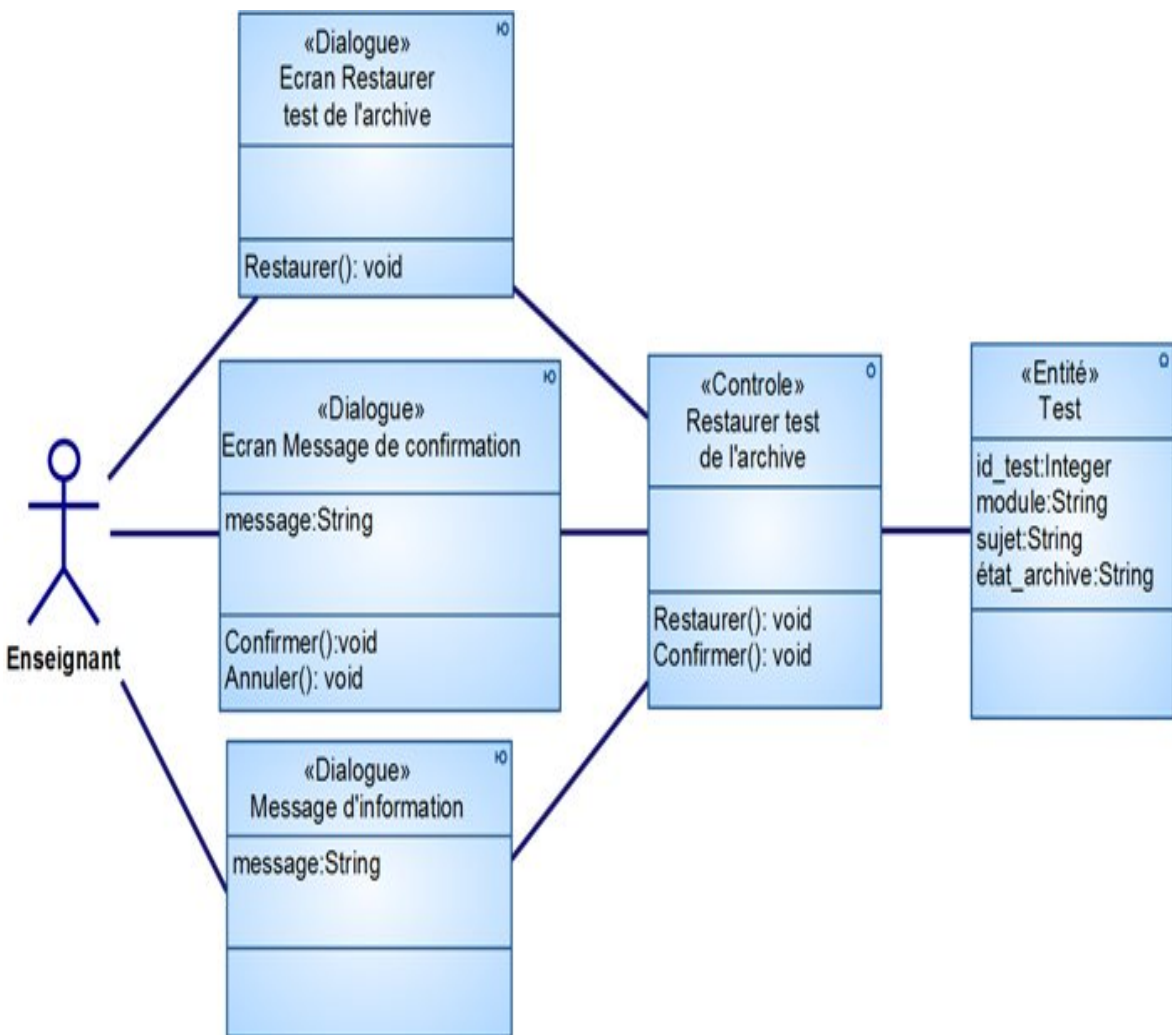


FIGURE 94 – Diagramme de classes participantes du cas Consulter test de l'archive



- « Gérer éditeur de style question »

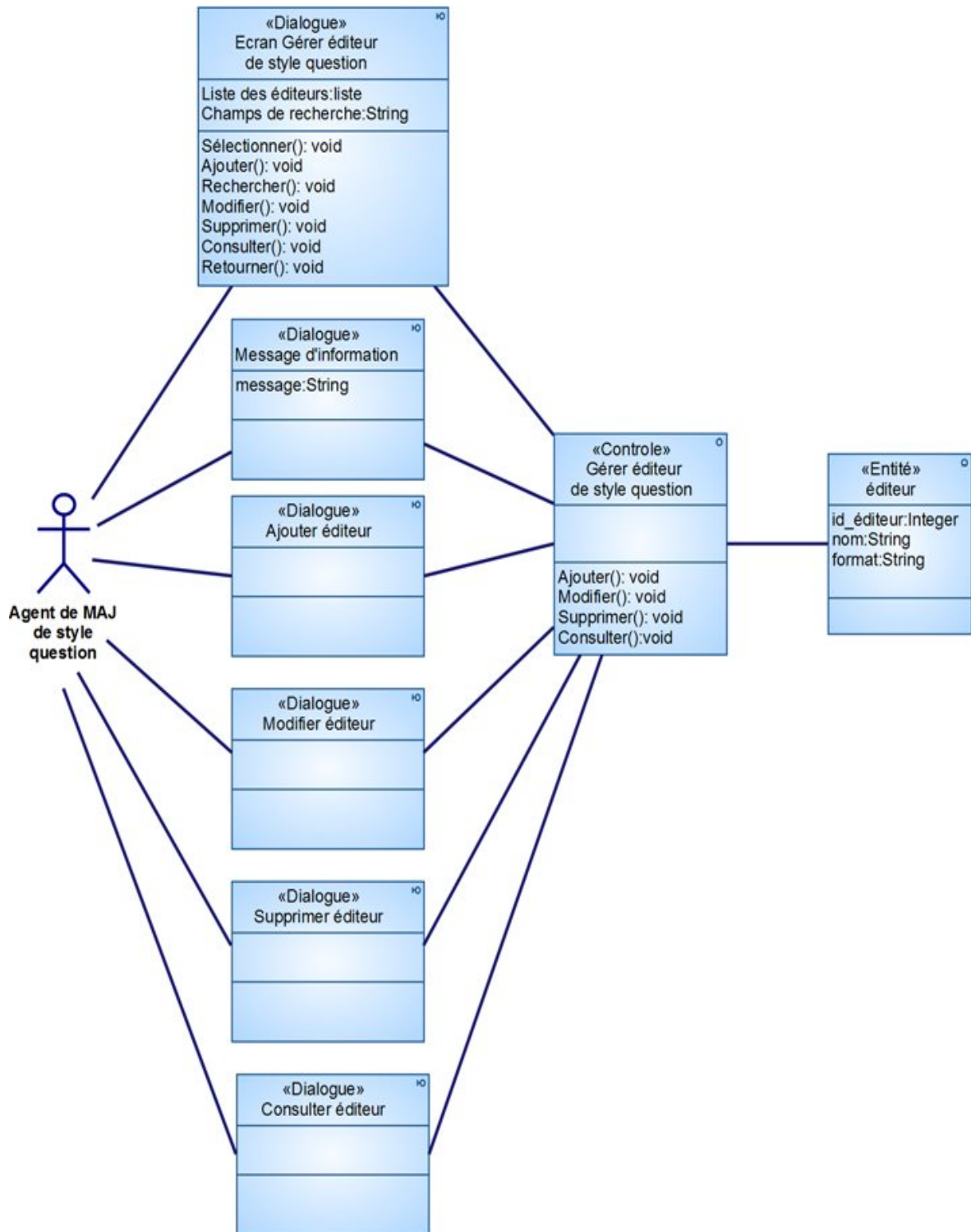


FIGURE 96 – Diagramme de classes participantes du cas Gérer éditeur de style question

- « Consulter éditeur »

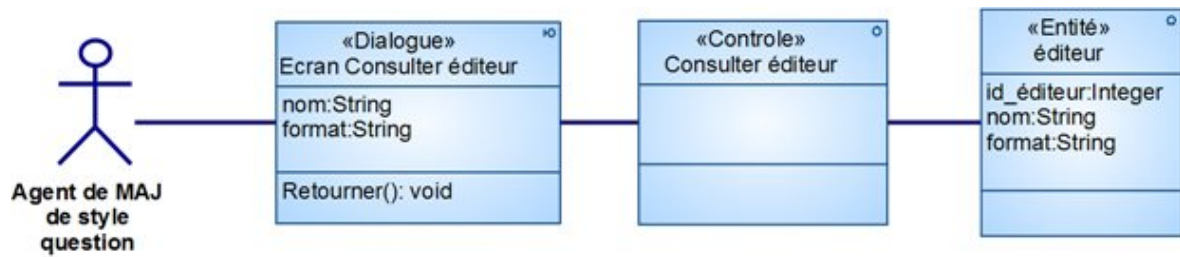


FIGURE 97 – Diagramme de classes participantes du cas Consulter éditeur

- « Supprimer éditeur »

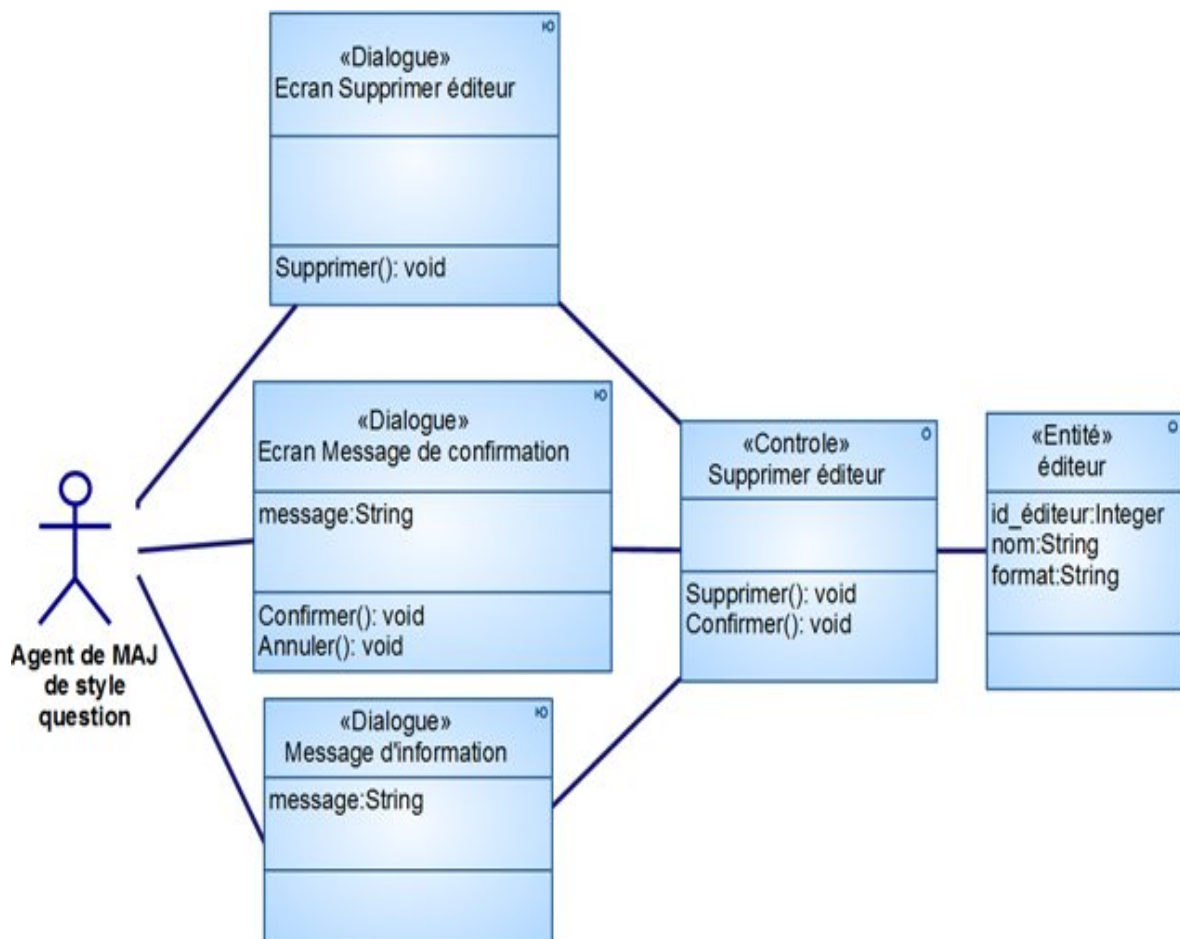


FIGURE 98 – Diagramme de classes participantes du cas Supprimer éditeur