

الجمهورية الجزائرية الديمقراطية الشعبية
République Algérienne Démocratique et Populaire
وزارة التعليم العالي والبحث العلمي
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique



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 réalisation d'une application mobile pour une école de formation

Préparé par : Bouanane Hadjer
Mehannaoui Hasna

Soutenu devant le jury

Président : Talai Meriem

M.A.A C.U.Abdelhafid Boussouf

Examineur : Abderrazak Samira

M.A.A C.U.Abdelhafid Boussouf

Encadreur : Boumassata Meriem

M.A.A C.U.Abdelhafid Boussouf

Année Universitaire : 2018/2019

Remerciements

*Tout d'abord, nous tient à remercier Allah,
Le clément Et le Miséricordieux de m'avoir donné le
courage d'accomplir ce Modeste travail.*

Mes sincères reconnaissances vont à mon encadreur

Boumassata Meriem

*qui m'a soutenue avec sa patience, ses Encouragements
et ses conseils.*

*Un grand merci sera également adressé aux membres du
Jury qui m'ont*

*Fait l'honneur de présider et d'examiner ce Modeste
travail.*

Notre reconnaissance s'adresse à nos familles surtout

mes parents qui ont su

*Nous apporter, sans relâcher leur soutien durant toutes
ces Langues années d'études.*

Enfin, nous remercions également, tous les enseignants,

*Qui ont assurés notre formation durant cycle
universitaire.*

Merci à tous.



Dedicace

*Je remercie **dieu** qui a toujours été à mes côtés.*

Je dédie ce modeste travail

*Aux fleurs de ma vie, **Ma mère**, ma raison d'être, ma*

raison de vivre,

La lanterne qui éclaire mon chemin

Et m'illumine de douceur et d'amour

***Mon père**, en signe d'amour, de reconnaissance et de*

gratitude pour tous Les soutiens et les sacrifices

dont il a

fait preuve à mon égard

*Mes dédicaces vont également à **mes frères** Et **mes sœurs***

Sans oublier de dédier ce mémoire à mon encadreur

Mme Boumassata Mereim,

*À mon binôme **hadjer**,*

*A tous **mes amies***

En témoignage de l'amitié sincère qui nous a liées Et des

bons moments passés

ensemble A tous les gens qui ont cru en moi et qui me

donnent l'envie D'aller en avant

Je veux remercier tous,

vos soutiens et vos encouragements me donnent

La force de continuer

Hasna

Dedicace

*Je remercie **dieu** qui a toujours été à mes côtés.*

Je dédie ce modeste travail

*Aux fleurs de ma vie, **Ma mère**, ma raison d'être, ma*

raison de vivre,

La lanterne qui éclaire mon chemin

Et m'illumine de douceur et d'amour

***Mon père**, en signe d'amour, de reconnaissance et de*

gratitude pour tous Les soutiens et les sacrifices

dont il a

fait preuve à mon égard

*Mes dédicaces vont également à **mes frères** Et **mes sœurs***

Sans oublier de dédier ce mémoire à mon encadreur

Mme Boumassata Mereim,

*À mon binôme **Hasna,***

*A tous **mes amies***

En témoignage de l'amitié sincère qui nous a liée Et des

bons moments passés

ensemble A tous les gens qui ont cru en moi et qui me

donnent l'envie D'aller en avant

Je veux remercier tous,

vos soutiens et vos encouragements me donnent

La force de continuer

Hadjer

Résumé

Les écoles de formation sont des établissements importants qui fournissent des services éducatifs efficaces grâce aux possibilités interactives qui aident l'apprenant à réussir dans ses études, en particulier dans le domaine de l'enseignement des langues étrangères et d'informatique, qui connaît une forte demande d'apprentissage au cours des dernières années.

L'objectif de ce mémoire concerne la conception et la réalisation d'une application mobile pour l'école de formation « EL Ameen School ». C'est une application qui permet de fournir un moyen de communication entre l'enseignant et l'étudiant pour partager des informations (cours, notes, annonces, etc.) et échanger des messages.

Pour atteindre ces objectifs, nous avons mené une étude conceptuelle en suivant le processus de développement logiciel 2TUP, utilisé UML comme langage de modélisation, le langage JAVA sous la plateforme ANDROID pour le développement et la programmation, et le SGBDR MySQL pour l'implémentation de la base de données.

Mots clés : Ecoles de formation, 2TUP, UML, JAVA, ANDROID, MySQL, Application mobile.

Abstract

Training schools are important institutions that provide effective educational services through interactive opportunities that help the learner to succeed in his studies, especially in the field of foreign language teaching and computer science, which is experiencing strong apprenticeship demand in recent years.

The purpose of this thesis concerns the design and implementation of a mobile application for the « EL Ameen School » training school. It is an application that provides a means of communication between the teacher and the student to share information (courses, notes, announcements, etc.) and exchange messages.

To achieve these objectives, we conducted a conceptual study following the 2TUP software development process, used UML as a modeling language, the JAVA language under the ANDROID platform for development and programming, and the MySQL SGBDR for the implementation of the database.

Keywords : Training schools, 2TUP, UML, JAVA, ANDROID, MySQL, Mobile application.

Sommaire

Table des figures	10
Liste des tableaux	14
Table des matières	15

INTRODUCTION GENERALE

1	19
1.1 INTRODUCTION	20
1.2 PRÉSENTATION DE L'ORGANISME D'ACCUEIL	20
1.2.1 DEFINITION DU CHAMP DE L'ETUDE	20
1.2.2 VUE GLOBALE SUR L'ÉCOLE « EL AMEED SCHOOL »	21
1.2.3 LES OBJECTIFS ÉDUCATIFS DE L'ÉCOLE	21
1.2.4 LA MEHTODOLOGIE DE L'ÉCOLE « EL AMEED SCHOOL»	22
1.2.5 LA STRUCTURE ADMINSTRATIVE DE L'ÉCOLE « EL AMEED SCHOOL»	22
1.2.6 LES FONCTIONS DES ACTEURS DE L'ÉCOLE	23
1.3 MOTIVATION	24
1.3.1 PROBLIMATIQUE	24
1.3.2 LA SOLUTION PROPOSEE : APPLICATION MOBILE « EL AMEED SCHOOL »	27
1.4 CONCLUSION	28
2	29
2.1 INTRODUCTION	30
2.2 LANGAGE MODILISATION UML	30
2.2.1 DEFINITION	30
2.2.2 HISTORIQUE	30
2.2.3 LES VUES UML	31

2.2.4	LES DIAGRAMMES UML	31
2.2.4.1	La vue statique (structurelle)	31
2.2.4.2	La vue dynamique (comportementale)	32
2.3	LE PROCESSUS UNIFIÉ (UP)	34
2.3.1	DEFINITION	34
2.3.2	LES CARACTÉRISTIQUES DU PROCESSUS UNIFIÉ	34
2.4	LE PROCESSUS 2TUP	35
2.4.1	DEFINITION	35
2.4.2	LES BRANCHES DU PROCESSUS 2TUP	35
2.4.2.1	La branche gauche (fonctionnelle)	35
2.4.2.2	La branche droite (architecture technique)	36
2.4.2.3	La branche du milieu	36
2.5	INFORMATIQUE MOBILE	37
2.5.1	EQUIPEMENT DE L'INFORMATIQUE MOBILE	37
2.5.1.1	Principaux équipements mobiles	37
2.5.2	SYSTEMES D'EXPLOITATION MOBILES	39
2.5.2.1	Définition	39
2.5.2.2	Les parts de marché des systèmes d'exploitation mobiles	40
2.5.3	ANDROID	40
2.5.3.1	DEFINITION	40
2.5.3.2	HISTORIQUE	40
2.5.3.3	LES OBJECTIFS D'ANDROID	42
2.5.4	ARCHITECTURE D'ANDROID	43
2.6	APPLICATION MOBILE	44
2.6.1	QU'EST-CE QU'UNE APPLICATION MOBILE ?	44
2.6.2	QU'EST-CE QU'UN SITE MOBILE ?	44
2.6.3	TYPES D'APPLICATIONS MOBILES	45
2.6.4	AVANTAGES ET INCONVÉNIENTS D'UNE APPLICATION MOBILE	45
2.7	CONCLUSION	46
3		47
3.1	INTRODUCTION	48
3.2	ETUDE PRÉLIMINAIRE	48
3.2.1	PRÉSENTATION DU PROJET	48
3.2.2	DESCRIPTION DU CONTEXTE	48
3.2.2.1	Identification des acteurs	48
3.2.2.2	Diagramme du contexte dynamique	49
3.2.2.3	Signification des messages	50

3.3	CAPTURE DES BESIONS FONCTIONNELS	52
3.3.1	DIAGRAMME DU CAS D'UTILISATION	53
3.3.2	DESCRIPTION DETAILLE DU CAS D'UTILISATION	54
3.3.2.1	Cas d'utilisation «Gérer enseignants »	54
3.3.2.2	Cas d'utilisation «Ajouter enseignant »	54
3.3.2.3	Cas d'utilisation «Modifier enseignant»	56
3.3.2.4	Cas d'utilisation «Supprimer enseignant»	57
3.3.2.5	Cas d'utilisation «Consulter enseignants»	59
3.3.2.6	Cas d'utilisation «Gérer étudiants »	61
3.3.2.7	Cas d'utilisation «Ajouter étudiant »	61
3.3.2.8	Cas d'utilisation «Modifier étudiant»	63
3.3.2.9	Cas d'utilisation «Supprimer étudiant»	64
3.3.2.10	Cas d'utilisation «Consulter étudiants»	66
3.3.2.11	Cas d'utilisation «Gérer formations »	68
3.3.2.12	Cas d'utilisation «Ajouter formation »	68
3.3.2.13	Cas d'utilisation «Modifier formation»	70
3.3.2.14	Cas d'utilisation «Supprimer formation»	71
3.3.2.15	Cas d'utilisation «Consulter formations»	73
3.3.2.16	Cas d'utilisation « Publier annonce »	75
3.3.2.17	Cas d'utilisation « Accéder aux ressources »	77
3.3.2.18	Cas d'utilisation « Evaluer le pourcentage de compréhension » 79	
3.3.2.19	Cas d'utilisation « Télécharger »	81
3.3.2.20	Cas d'utilisation « Consulter annonces »	83
3.3.2.21	Cas d'utilisation « Consulter notes »	84
3.3.2.22	Cas d'utilisation « Consulter évènements »	86
3.3.2.23	Cas d'utilisation « Participer »	88
3.3.2.24	Cas d'utilisation « Consulter Résultat »	90
3.3.2.25	Cas d'utilisation « Gérer ressources »	92
3.3.2.26	Cas d'utilisation «Ajouter ressource »	92
3.3.2.27	Cas d'utilisation « Retirer ressource »	93
3.3.2.28	Cas d'utilisation « Afficher notes »	96
3.3.2.29	Cas d'utilisation « Lancer événement »	97
3.3.2.30	Cas d'utilisation « Consulter Participations »	99
3.3.2.31	Cas d'utilisation « Afficher Résultat »	101
3.3.2.32	Cas d'utilisation « Consulter espace de communication » . .	103
3.3.2.33	Cas d'utilisation « Ecrire message »	105
3.3.2.34	Cas d'utilisation « Mettre à jour notes »	107

3.4	LE MODELE DE DOMAINE	109
3.5	CAPTURE DES BESIONS TECHNIQUES	110
3.5.1	LA CONFIGURATION METERAILLE DU SYSTEME	110
3.5.1.1	Développement du modèle de déploiement	110
3.5.1.2	Spécification d'architecture 2 tiers	110
3.5.2	CAPTURE DES SPECIFICATIONS LOGICIELLES	111
3.5.2.1	Identification des exploitants du système	111
3.5.2.2	Identification des cas d'utilisations techniques	111
3.5.2.3	Description des cas d'utilisation techniques	112
3.6	CONCLUSION	121
4		122
4.1	INTRODUCTION	123
4.2	PHASE D'ANALYSE	123
4.2.1	DÉCOUPAGE EN CATÉGORIES	123
4.2.1.1	Définition de catégorie	123
4.2.1.2	Répartition des classes candidates en catégories	124
4.2.1.3	Élaboration des diagrammes de classes préliminaire par catégorie	124
4.2.1.4	Dépendance entre catégories	126
4.2.2	DEVELOPPEMENT DU MODELE STATIQUE	126
4.3	PHASE DE CONCEPTION	128
4.3.1	CONCEPTION PRELIMINAIRE	128
4.3.1.1	Définition	128
4.3.1.2	Développement du modèle de déploiement	128
4.3.2	CONCEPTION DETAILLEE	129
4.3.2.1	Définition	129
4.3.2.2	Passage du diagramme de classe au modèle relationnel.	129
4.3.2.3	Tables de la base de données	130
4.3.2.4	Diagramme de classes détaillé	131
4.4	CONCLUSION	132
5		133
5.1	INTRODUCTION	134
5.2	PRESENTATION DES OUTILS DE DEVELOPPEMENT	134
5.2.1	Langage de programmation java	134
5.2.2	PHP	134
5.2.3	MySQL	135

5.2.4	Xampp server	135
5.2.5	Andriod studio	135
5.2.6	Le SDK Android	136
5.2.7	Framework « Laravel »	136
5.2.8	Bootstrap	136
5.2.9	Postman	137
5.2.10	Firestore	137
5.3	INTERFACES DE L'APPLICATION WEB	138
5.3.1	Fenêtre d'authentification	138
5.3.2	Fenêtre d'accueil	138
5.3.3	Fenêtre ajouter formation	139
5.3.4	Fenêtre gère étudiants	139
5.3.5	Fenêtre ajouter étudiant	139
5.3.6	Fenêtre modifier enseignant	140
5.4	INTERFACE D'APPLICATION MOBILE	140
5.4.1	Fenêtre d'accueil	140
5.4.2	Fenetre GPS	141
5.4.3	Fenêtre espace enseignant	142
5.4.4	Fenêtre espace étudiant	142
5.4.5	Fenêtre lancer évènement	143
5.4.6	Fenêtre consulter évènement	143
5.4.7	Fenêtre consulter participations	144
5.4.8	Fenêtre afficher résultat	144
5.4.9	Fenêtre consulter résultat	145
5.4.10	Fenêtre gérer ressources PDF	145
5.4.11	Fenêtre évaluer le pourcentage de compréhension	146
5.4.12	Fenêtre consulter espace de communication	146
5.5	CONCLUSION	147

CONCLUSION GENERALE

148

Bibliographie

149

Table des figures

1.1	Logo de l'école « El Ameed School »	20
1.2	Organigramme de l'école « El Ameed School »	25
1.3	Logo de l'application « El Ameed School »	26
2.1	Le processus de développement en Y.	36
2.2	Ordinateur portable.	37
2.3	Les smartphones.	37
2.4	Tablette.	38
2.5	PDA.	38
2.6	Logo Android OS.	39
2.7	Architecture d'Android.	43
3.1	Diagramme de contexte dynamique.	49
3.2	Diagramme des cas d'utilisations	52
3.3	Diagramme de séquence «Ajouter enseignant».	54
3.4	Diagramme de séquence «Modifier enseignant».	56
3.5	Diagramme de séquence «Supprimer enseignant».	57
3.6	Diagramme de séquence «Consulter enseignants».	58
3.7	Diagramme d'activité «Gérer enseignants ».	59
3.8	Diagramme de séquence «Ajouter étudiant ».	61
3.9	Diagramme de séquence «Modifier étudiant».	63
3.10	Diagramme de séquence «Supprimer étudiant».	64
3.11	Diagramme de séquence «Consulter étudiants».	65
3.12	Diagramme d'activité «Gérer étudiants ».	66
3.13	Diagramme de séquence «Ajouter formation ».	68
3.14	Diagramme de séquence «Modifier formation».	70
3.15	Diagramme de séquence «Supprimer formation».	71
3.16	Diagramme de séquence «Consulter formations».	72
3.17	Diagramme d'activité «Gérer formations ».	73
3.18	Diagramme de séquence «Publier annonce».	75

3.19	Diagramme d'activité «Publier annonce».	75
3.20	Diagramme de séquence «Accéder aux ressources».	77
3.21	Diagramme d'activité «Accéder aux ressources».	78
3.22	Diagramme de séquence «Evaluer le pourcentage de compréhension».	79
3.23	Diagramme d'activité «Evaluer le pourcentage de compréhension».	80
3.24	Diagramme de séquence «Télécharge».	81
3.25	Diagramme d'activité «Télécharge».	81
3.26	Diagramme de séquence «Consulter annonces».	82
3.27	Diagramme d'activité «Consulter annonces».	83
3.28	Diagramme de séquence «Consulter notes».	84
3.29	Diagramme d'activité «Consulter notes».	84
3.30	Diagramme de séquence «Consulter évènements».	86
3.31	Diagramme d'activité «Consulter évènements».	87
3.32	Diagramme de séquence «Participer».	88
3.33	Diagramme d'activité «Participer».	89
3.34	Diagramme de séquence «Consulter Résultat».	90
3.35	Diagramme d'activité «Consulter Résultat».	90
3.36	Diagramme de séquence « Ajouter ressource ».	92
3.37	Diagramme de séquence «Retirer ressource».	93
3.38	Diagramme d'activité «Gérer ressources».	94
3.39	Diagramme de séquence «Afficher notes».	95
3.40	Diagramme d'activité «Afficher notes».	96
3.41	Diagramme de séquence «Lancer événement».	97
3.42	Diagramme d'activité «Lancer événement».	98
3.43	Diagramme de séquence «Consulter Participations».	99
3.44	Diagramme d'activité «Consulter Participations».	100
3.45	Diagramme de séquence «Afficher Résultat ».	101
3.46	Diagramme d'activité «Afficher Résultat ».	102
3.47	Diagramme de séquence «Consulter espace de communication».	103
3.48	Diagramme d'activité «Consulter espace de communication ».	104
3.49	Diagramme de séquence «Ecrire message».	105
3.50	Diagramme d'activité «Ecrire message ».	106
3.51	Diagramme de séquence «Mettre à jour notes».	107
3.52	Diagramme d'activité «Mettre à jour notes ».	108
3.53	Le modèle de domaine.	108
3.54	Diagramme de déploiement.	109
3.55	Architecture 2 niveaux de notre système.	110

3.56	Modèle de spécification logicielle du système.	110
3.57	Diagramme de séquence «S’authentifier».	112
3.58	Diagramme d’activité «S’authentifier».	112
3.59	Diagramme de séquence «Ajouter compte».	114
3.60	Diagramme de séquence «Modifier compte».	116
3.61	Diagramme de séquence «Supprimer compte».	117
3.62	Diagramme de séquence «Consulter comptes».	118
3.63	Diagramme d’activité «Gérer comptes ».	119
4.1	Le découpage en catégories.	123
4.2	Découpage des classes dans les catégories <Formation>.	123
4.3	Découpage des classes dans les catégories <Personne >.	124
4.4	Découpage des classes dans les catégories <Communication >.	124
4.5	Dépendances entre les catégories.	125
4.6	Diagramme de classe de la catégorie « Formation ».	126
4.7	Diagramme de classe de la catégorie «Personne ».	126
4.8	Diagramme de classe de la catégorie «Communication».	127
4.9	Schéma du modèle de déploiement du système.	128
4.10	Diagramme de classes détaillé de notre système.	130
5.1	Logo java.	133
5.2	Logo Xampp.	134
5.3	Logo Android Studio.	135
5.4	Logo Laravel.	135
5.5	Logo Postman.	136
5.6	Logo Firebase.	136
5.7	Fenêtre d’authentification du directeur des études.	137
5.8	Fenêtre d’accueil.	137
5.9	Fenêtre ajouter formation.	138
5.10	Fenêtre gère étudiants.	138
5.11	Fenêtre ajouter étudiant.	139
5.12	Fenêtre de modification d’un enseignant.	139
5.13	Fenêtre d’accueil.	140
5.14	Fenêtre GPS.	140
5.15	Fenêtre espace enseignant.	141
5.16	Fenêtre espace étudiant.	141
5.17	Fenêtre lancer évènement.	142
5.18	Fenêtre de consultation d’un évènement.	142

5.19 Fenêtre de consultation de participations.	143
5.20 Fenêtre afficher résultat.	143
5.21 Fenêtre consulter résultat.	144
5.22 Fenêtre gérer ressources PDF.	144
5.23 Fenêtre d'évaluation du pourcentage de compréhension.	145
5.24 Fenêtre consulter espace de communication.	145

Liste des tableaux

2.1	Les versions d'Android.	41
3.1	Signification des messages.	51
3.2	Description textuelle du cas d'utilisation « Gérer enseignants ».	53
3.3	Description textuelle du cas d'utilisation «Ajouter enseignant ».	54
3.4	Description textuelle du cas d'utilisation «Modifier enseignant ».	55
3.5	Description textuelle du cas d'utilisation «Supprimer enseignant».	57
3.6	Description textuelle du cas d'utilisation «Consulter enseignants».	58
3.7	Description textuelle du cas d'utilisation « Gérer étudiants ».	60
3.8	Description textuelle du cas d'utilisation «Ajouter étudiant ».	61
3.9	Description textuelle du cas d'utilisation «Modifier étudiant ».	62
3.10	Description textuelle du cas d'utilisation «Supprimer étudiant».	64
3.11	Description textuelle du cas d'utilisation «Consulter étudiants».	65
3.12	Description textuelle du cas d'utilisation « Gérer formations ».	67
3.13	Description textuelle du cas d'utilisation «Ajouter formation ».	68
3.14	Description textuelle du cas d'utilisation «Modifier formation ».	69
3.15	Description textuelle du cas d'utilisation «Supprimer formation».	71
3.16	Description textuelle du cas d'utilisation «Consulter formations».	72
3.17	Description textuelle du cas d'utilisation «Publier annonce».	74
3.18	Description textuelle du cas d'utilisation «Accéder aux ressources».	76
3.19	Description textuelle du cas d'utilisation «Evaluer le pourcentage de compréhension».	79
3.20	Description textuelle du cas d'utilisation «Télécharger».	80
3.21	Description textuelle du cas d'utilisation «Consulter annonces».	82
3.22	Description textuelle du cas d'utilisation «Consulter notes».	83
3.23	Description textuelle du cas d'utilisation «Consulter évènements».	85
3.24	Description textuelle du cas d'utilisation «Participer».	88
3.25	Description textuelle du cas d'utilisation «Consulter Résultat».	89
3.26	Description textuelle du cas d'utilisation « Gérer ressources ».	91
3.27	Description textuelle du cas d'utilisation «Ajouter ressource ».	92

3.28	Description textuelle du cas d'utilisation «Retirer ressource».	93
3.29	Description textuelle du cas d'utilisation «Afficher notes».	95
3.30	Description textuelle du cas d'utilisation «Lancer événement ».	97
3.31	Description textuelle du cas d'utilisation «Consulter Participations».	99
3.32	Description textuelle du cas d'utilisation «Afficher Résultat ».	101
3.33	Description textuelle du cas d'utilisation «Consulter espace de communication ».	103
3.34	Description textuelle du cas d'utilisation «Ecrire message ».	105
3.35	Description textuelle du cas d'utilisation «Mettre à jour notes ».	107
3.36	Description textuelle du cas d'utilisation «S'authentifier ».	111
3.37	Description textuelle du cas d'utilisation « Gérer comptes ».	113
3.38	Description textuelle du cas d'utilisation «Ajouter compte ».	114
3.39	Description textuelle du cas d'utilisation «Modifier compte ».	115
3.40	Description textuelle du cas d'utilisation «Supprimer compte».	117
3.41	Description textuelle du cas d'utilisation «Consulter comptes».	118
4.1	Table de la base de données.	129

INTRODUCTION GENERALE

Les nouvelles technologies de communication connaissent une croissance exponentielle. Ils ont révolutionné notre façon de vivre et de voir les choses. Les appareils mobiles tels que les Smartphones ou les tablettes sont de plus en plus utilisés par tous les segments de la société, car la présence de ces nouvelles technologies offre de réelles opportunités dans différents contextes, ce qui a conduit à l'émergence de nombreuses applications mobiles.

Une application mobile est un logiciel développé pour être installé sur un appareil électronique mobile (Smartphone, tablette, etc.) afin de faciliter le quotidien de l'humain dans pratiquement tous les domaines. Le nombre de téléchargements d'applications mobiles est actuellement en forte hausse. Cette tendance va de pair avec la vente des Smartphones.

Parmi les établissements qui cherchent à améliorer leurs services en fournissant des services à distance via les technologies de communication (Smartphone, tablettes, etc.) sont les écoles de formation, qui représentent des établissements ou bien des centres de formation pour l'apprentissage de différentes sciences (les langues, l'informatique, etc.).

Dans ce contexte, nous avons proposé de développer une application mobile pour l'école de formation « EL Amed School », dans laquelle nous avons effectué notre stage. Cette école offre depuis plus de 4 ans des formations linguistiques et informatiques de qualité. Ces offres sont variées et s'adressent à un large public : travailleurs, étudiants ou professeurs universitaires, stagiaires postdoctoraux, candidats aux ordres professionnels et étudiants étrangers.

Au cours de notre stage nous avons observé un manque de moyens qui permettent à l'administration et aux enseignants de communiquer une information à l'étudiant à distance (faire des affichages, communiquer les résultats des tests et évaluations, lancer des événements à distance, etc.). L'école désire synchroniser ses services et fonctionnalités d'une manière performante, facile et efficace afin de satisfaire les étudiants et les enseignants de l'école. Pour cette raison l'école désire utiliser une application mobile pour garantir l'activité des enseignants et étudiants.

Pour remédier à ces problèmes nous avons essayé de développer une application mobile destinée à l'école de formation « El Amed School » afin de constituer une interface entre les étudiants et les enseignants de l'école dont le but est d'atteindre les objectifs suivants :

- Permettre à l'enseignant de gérer ses ressources (cours et multimédias), évaluations et affichages d'une manière facile et efficace, de lancer des évènements à distance, et de publier des annonces.
- Permettre à l'étudiant de consulter les cours et multimédias, les évènements, les annonces et les notes de tests, de participer aux événements, et d'évaluer le pourcentage de compréhension des cours.
- Fournir un espace communication entre les étudiants et les enseignants.

Nous avons essayé, en plus, de développer une application web pour l'administration de l'école qui permet de faciliter les tâches du directeur des études (créer des formations, mettre à jour des informations concernant les enseignants et les étudiants de l'école, publier des annonces, etc.).

Le reste du mémoire sera organisé en cinq chapitres comme suit :

Dans le premier chapitre, nous allons présenter l'école « El Ameen School » et nous allons analyser et critiquer les méthodes utilisées par l'école pour la communication entre administration, enseignants et étudiants, à fin de proposer une solution fiable et efficace.

Dans le deuxième chapitre, nous allons présenter le langage de modélisation UML et le processus de développement 2TUP, ainsi que le concept de l'informatique mobile, le système d'exploitation Android, et les applications mobiles.

Dans le troisième chapitre, nous allons décrire les besoins et les acteurs qui vont interagir avec notre système et présenter un recueil des besoins fonctionnels et techniques du système.

Dans le quatrième chapitre, nous allons présenter les phases d'analyse et de conception. Nous commençons par l'analyse, qui permet d'étudier précisément les spécifications fonctionnelles de manière à obtenir une idée de ce que va réaliser le système en terme de métier. Par la suite, nous passons à la phase de conception qui comprend la conception préliminaire et détaillée.

Dans le cinquième chapitre, nous allons présenter les environnements et les outils de développement de l'application mobile et de l'application web, ainsi que quelques interfaces de l'application.

Chapitre 1

Étude de l'existant et objectif du projet

1.1 INTRODUCTION

L'étude de l'existant est une étape indispensable dans tout projet informatique. Elle consiste en un travail de terrain au terme duquel on pourra bien connaître la situation actuelle de l'organisation.

Pour bien comprendre le problème que nous voulons résoudre et préciser le but que nous allons réaliser, on doit étudier le domaine de travail qui représente les écoles de formation. En ce qui nous concerne, notre étude de l'existant portera sur l'école privée «El Ameed School».

Dans ce chapitre, nous allons présenter l'école « El Ameed School », ses objectifs éducatifs, sa méthodologie et sa structure administrative. Ensuite, nous allons définir les fonctions de chaque acteur dans l'école. Et en fin, nous allons analyser et critiquer les méthodes utilisées par l'école pour la communication entre administration, enseignants et étudiants, à fin de proposer une solution fiable et efficace.

1.2 PRÉSENTATION DE L'ORGANISME D'ACCUEIL

1.2.1 DEFINITION DU CHAMP DE L'ETUDE

L'école de formation représente un établissement ou bien un centre de formation pour l'apprentissage de différentes sciences (les langues, l'informatique, etc.). En ce qui concerne les langues, l'école divise les étudiants en groupes selon plusieurs niveaux pour apporter le meilleur suivi linguistique possible.

Le système de l'école repose sur la gestion de différents éléments. Parmi ces éléments les enseignants, les étudiants, les cours et multimédia, la durée et le cout de chaque formation.

Le but essentiel de la gestion de tous ces éléments est :

- Fournir un programme au niveau des aspirations des cours inscrits.
- Donner une bonne organisation des groupes des étudiants pour chaque formation par exemple un fichier numérique précisé pour chaque étudiant.
- Assurer un bon accès à tous les cours.
- La communication est l'élément le plus important dans le développement des compétences de l'apprenant. Pour cette raison, toutes les écoles sont conçues pour assurer la communication entre les étudiants et les enseignants.
- planifier les inscriptions et les durées des formations.
- Contrôler les paiements des frais des formations.

1.2.2 VUE GLOBALE SUR L'ÉCOLE « EL AMEED SCHOOL »

Notre stage a été effectué dans l'école privée « El Ameer School », qui a été inaugurée en janvier 2015. L'école « El Ameer School » offre depuis plus de 4 ans des formations linguistiques et informatiques de qualité. Ces offres sont variées et s'adressent à un large public : travailleurs, étudiants ou professeurs universitaires, stagiaires postdoctoraux, candidats aux ordres professionnels et étudiants étrangers.

L'école est considérée comme un établissement privé. Elle est placée à un endroit stratégique à la périphérie de la ville de Mila. L'école comporte quatre salles où se déroulent les cours, en plus une salle informatique. Les cours sont donnés par des enseignants expérimentés et hautement qualifiés.



FIGURE 1.1 – Logo de l'école « El Ameer School ».

1.2.3 LES OBJECTIFS ÉDUCATIFS DE L'ÉCOLE

Depuis sa création, l'école « El Ameer School » vise à souligner un ensemble d'objectifs, qui sont représentés par :

- L'école El Ameer a pour mission de proposer des formations variées et de qualité en français, Arabe, anglais, Allemand, Espagnol, Turc et italien (en tant que langue secondaire) orienté pour les adultes spécialement.
- L'école s'engage à accompagner ses étudiants dans le développement des compétences linguistiques et interculturelles qui leurs permettront d'interagir efficacement avec le monde qui les entoure.
- L'école « El Ameer School » offre des cours de soutien spéciaux dans les trois langues principales, à savoir l'Arabe, l'Anglais et le Français.
- Qu'il soit au niveau débutant, intermédiaire ou avancé, qu'il désire étudier seul ou en groupe, l'étudiant trouve chez l'école El Ameer la formation linguistique qui convient le mieux à ses besoins.
- L'école propose aussi des formations en informatique.

1.2.4 LA METHODOLOGIE DE L'ÉCOLE « EL AMEED SCHOOL »

Dans cette structure, le nombre des étudiants dans chaque classe est limité, il varie entre six et douze personnes. De même, une homogénéité est présente, les groupes ont des âges et des niveaux relativement équivalents. Enfin, les cours et les activités linguistiques sont en rapport avec les aspirations personnelles et professionnelles des étudiants. Pour cela l'organisation de l'enseignement dans l'école se déroule comme suit :

1. L'enseignement est organisé en séquences comportant des séances.
2. Les séquences sont organisées autour de différents objectifs.
3. Des lectures et des devoirs pourront être effectués à la maison ou à l'école.
4. En cas d'absence, l'étudiant pourra rattraper le cours dans les plus brefs délais (auprès de ses professeurs et de ses camarades de classe).
5. L'étudiant est régulièrement évalué à travers ses performances dans diverses activités de production et compréhension orale ou écrite.
6. Les enseignants font des évaluations formatives – bilans toutes les quinzaines et proposent des activités de remédiation.
7. Les enseignants mettent en place la pédagogie différenciée pour répondre aux besoins de chaque étudiant.
8. Un examen de passage est obligatoire à la fin de chaque niveau.

1.2.5 LA STRUCTURE ADMINISTRATIVE DE L'ÉCOLE « EL AMEED SCHOOL »

L'école « EL Ameen School » se compose des acteurs suivants :

- Directeur.
- Directeur de l'étude.
- Secrétaire.
- Designer.
- Responsable de labo des langues.
- Responsable d'informatique.
- Responsable de Langue pour Adulte.
- Responsable de Cours de soutien.
- Enseignants de la langue Arabe.
- Enseignants de la langue Française.

- Enseignants de la langue Anglaise.
- Enseignants de la langue Espagnole.
- Enseignants de la langue Allemande.
- Enseignants de la langue Turque.
- Enseignants de la langue Italienne.
- Enseignants d'informatique.

1.2.6 LES FONCTIONS DES ACTEURS DE L'ÉCOLE

- **Le directeur**
 - Préparation du programme annuel d'activités.
 - Supervision des directions.
 - Programmation des moyens nécessaires pour suivre les étudiants.
 - Déterminer le type des formations à effectuer.
 - Préparation du programme pour les réunions de coordination mensuelles.
- **Le secrétaire**
 - Organisation des rendez-vous du directeur.
 - Accueil des visiteurs qui souhaitent s'inscrire et explication des formations de l'école.
 - Recevoir des appels téléphoniques et y répondre.
 - Classification, arrangement, et sauvegarde des copies de divers documents de la direction générale.
 - Préparation des réunions de la direction.
- **Designer**
 - Concevoir des logos et des symboles pour l'école.
 - Créer des vidéos et des designs utilisés dans la définition et l'annonce de l'école.
- **Le directeur de l'étude**
 - Suivi des activités qui concernent la branche des enfants.
 - Création de contenu de chaque formation.
 - Gestion des formations lancées au niveau de l'école ; choisir les dates de lancement et de fin de chaque formation, préciser le nombre des étudiants, le tarif et les délais d'inscription dans la formation.
 - Gestion des annonces de l'école.

— **Les responsables**

- Déterminer le contenu des cours présentées dans chaque formation et suivre leur progression.
- Superviser et diriger les enseignants de formation.

— **Les Enseignants**

- Superviser les cours et veiller à ce qu'elles soient présentées de manière fluide et compréhensible pour les étudiants.
- Suivi et évaluation des étudiants tout au long de la formation et réponses aux questions sur les cours fournies.

1.3 MOTIVATION

1.3.1 PROBLIMATIQUE

L'école désire synchroniser ses services et fonctionnalités d'une manière performante, facile et efficace afin de satisfaire les étudiants et les enseignants de l'école. Elle désire, aussi, garantir le maintien de l'activité pour détecter les problèmes existants. Actuellement, l'école utilise plusieurs systèmes informatiques qui contribuent à l'enseignement moderne et avancé, parmi ses systèmes :

- **Excel** : les enseignants utilisent le logiciel Excel, généralement pour afficher les notes des tests et les envoyer en message.
- **Scratch** : c'est un langage de programmation graphique manipulable et exécutable par le logiciel de même nom à vocation éducative, qui permet aux étudiants d'apprendre et d'exprimer leurs capacités à l'aide de techniques modernes.
- **Discord App** : Discord est un logiciel gratuit de voip (la voix sur IP) conçu à la base pour les communautés de joueurs. L'école El Amed utilise ce logiciel pour fournir des formations interactives surtout dans les formations de langues.

Au cours de la période de stage, nous avons analysé les différentes activités de l'école et nous avons observé les méthodes utilisées pour la réalisation de ces activités. En plus, nous avons interrogé le directeur des études de l'école qui nous a expliqué le mode de fonctionnement de l'école et quelques anomalies.

Après une étude approfondie et continue, nous avons pu observer les insuffisances suivantes :

1. Le volume important des informations traitées manuellement par l'administration et les responsables, ce qui provoque parfois des erreurs dans l'établissement des documents.

2. Les nouveaux étudiants, en particulier dans les branches linguistiques, souffrent d'un certain nombre de difficultés, en particulier de prononciation et de mémorisation de mots nouveaux, car il n'existe pas de système d'interprétation et il y a une difficulté à communiquer avec l'enseignant hors les séances de cours.
3. Pour les enseignants, il est difficile de donner des exemples abstraits et de les mettre en pratique pour l'étudiant.
4. Il n'y a pas de moyens qui permettent à l'administration et aux enseignants de communiquer une information à l'étudiant à distance (faire des affichages, communiquer les résultats des tests et évaluations, lancer des événements à distance, etc.).

Et pour cela nous avons assigné à notre étude les objectifs suivants :

- Fournir à l'administration un outil qui permet de gérer les formations (contenu, séances, enseignants, groupes d'étudiants, etc.) et les annonces.
- Permettre à l'enseignant de gérer ses cours, multimédias, évaluations et affichages d'une manière facile et efficace.
- Permettre à l'enseignant de lancer des événements à distance.
- Garantir la disponibilité des cours et des multimédia soutiens à tout moment pour l'étudiant.
- Améliorer la communication entre l'étudiant et l'enseignant en fournissant un espace commun (pour les questions, par exemple) pour assurer un cours intéressant et interactif.
- Permettre à l'étudiant de consulter les annonces de l'administration et des enseignants à tout moment.

La figure 1.2 représente l'organigramme de la structure administrative de l'école.

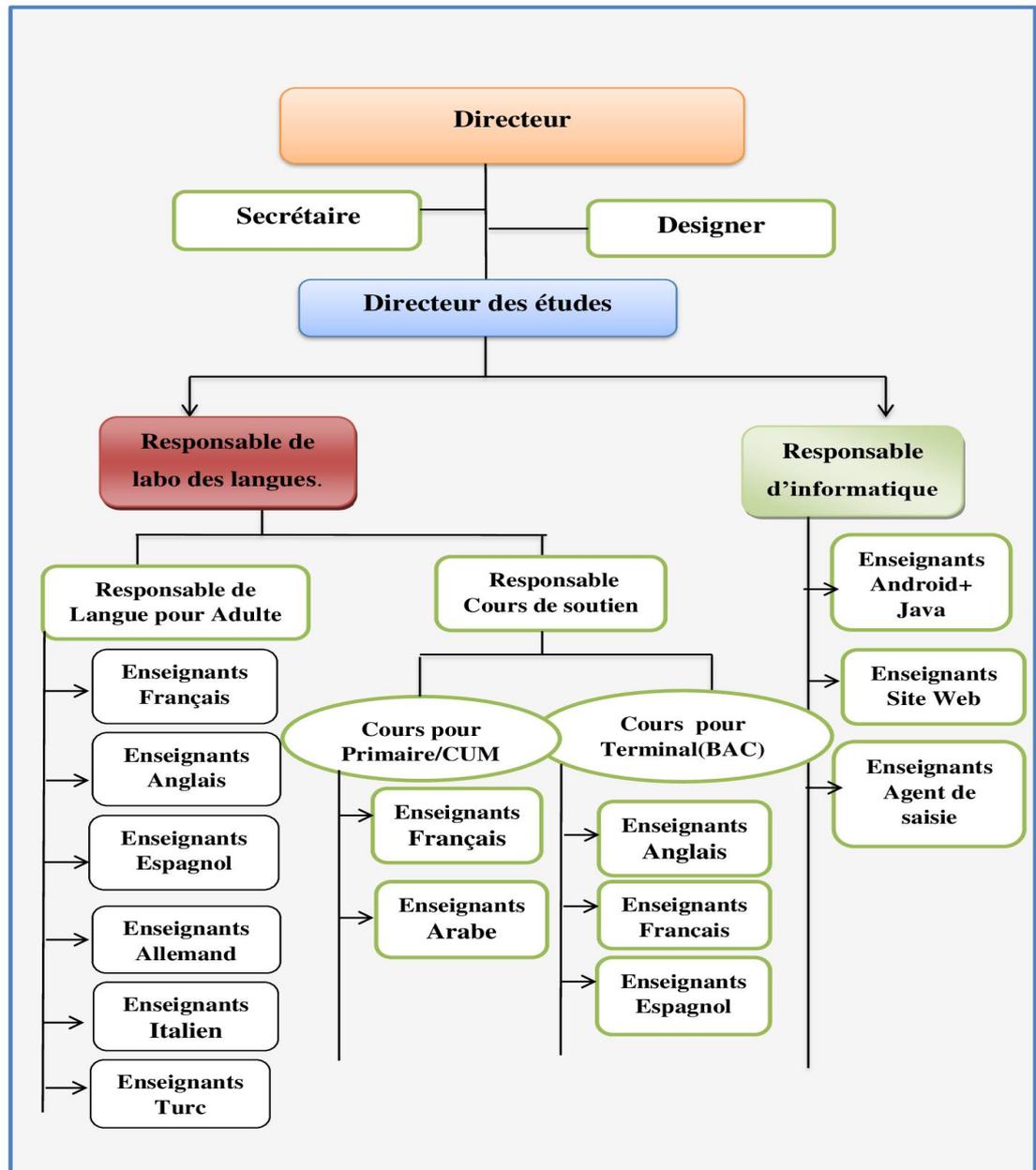


FIGURE 1.2 – Organigramme de l'école « El Amed School »

1.3.2 LA SOLUTION PROPOSEE : APPLICATION MOBILE « EL AMEED SCHOOL »

Notre projet consiste à réaliser un système informatique qui permet de traiter les problèmes de gestion de l'école et d'améliorer la méthode d'apprentissage de l'étudiant. Ce système permet de faciliter le travail de l'administration et de l'enseignant d'un côté, et de traiter le problème de communication entre administration, étudiants et enseignants d'un autre côté.

Le système que nous proposons se compose d'une application Android pour les étudiants et les enseignants, et une application web pour le directeur des études.



FIGURE 1.3 – Logo de l'application « El Ameed School ».

- **La partie Android : « El Ameed School »**

Cette application est adressée aux enseignants et aux étudiants. D'un côté, elle fournit à l'enseignant des fonctionnalités qui lui permettent de gérer ses cours et multimédias, d'afficher des annonces, de lancer des événements, et de communiquer les résultats des tests et évaluations aux étudiants. D'un autre côté, l'application permet aux étudiants de consulter les annonces, les cours, les multimédias, les notes, de participer aux événements, et d'évaluer les cours. En plus, l'application permet aux étudiants et leur enseignant d'accéder à un espace de communication commun (pour discuter les cours, poser des questions, etc.).

- **Application web : « El Ameed Manager »**

C'est une application web adressée au directeur des études de l'école. Elle lui permet de créer des formations et de les mettre à jour, d'ajouter et de mettre à jour des informations concernant les enseignants et les étudiants de l'école, et de publier des annonces.

1.4 CONCLUSION

Dans ce chapitre, nous avons présenté l'organisme d'accueil « EL Ameen School » dans lequel nous avons effectué notre stage. Nous avons mis l'accent sur la structure administrative, les méthodes utilisées et les fonctionnalités de chaque acteur dans l'école. Puis nous avons discuté les problèmes existants avec la proposition d'une solution. Pour la réalisation de notre application nous allons avoir besoin d'une démarche de développement. Dans le chapitre suivant, nous allons présenter quelques concepts de base sur les applications mobiles et sur le processus et la démarche de développement qui nous aiderons tout au long de notre projet.

Chapitre 2

Concepts de Base

2.1 INTRODUCTION

Une application mobile est un logiciel développé pour être installé sur un appareil électronique mobile pour faciliter le quotidien de l'homme pratiquement dans tous les domaines.

Actuellement la conception et la réalisation des applications mobiles sont basées sur plusieurs méthodes et langages de modélisation. Il existe de nombreuses méthodes de conception et de développements disponibles, notre choix c'est porté vers la méthode 2TUP qui s'appuie sur le langage de modélisation UML tout au long de cycle de développement.

Dans ce chapitre, nous présentons quelques généralités sur le langage de modélisation UML, ses vues et ses diagrammes, ainsi que le processus 2TUP et ses phases. Ensuite, nous définissons le concept de l'informatique mobile, ses équipements et ses systèmes d'exploitation essentiellement l'Android. Et en fin, nous présentons les applications mobiles, leurs types, avantages et inconvénients.

2.2 LANGAGE MODILISATION UML

2.2.1 DEFINITION

UML (Unified Modeling Language) se définit comme un langage de modélisation graphique qui permet la spécification, la représentation et la construction des composants, dans le but de modéliser d'une manière claire et précise la structure et le comportement d'un système indépendant de toute méthode et tout langage de programmation [1].

2.2.2 HISTORIQUE

- Les années 1980 : Utilisation de méthodes adaptées à la programmation impérative (notamment Merise).
- Fin 80/Début 90 : la programmation par objet prend de l'importance.
- Mise en place de méthodes orientées objet. Plus de cinquante méthodes apparaissent entre 1990 et 1995 :
 - OOD : Object Oriented Design Booch, 1991.
 - HOOD : Hierarchical Object Oriented Design Delatte et al., 1993.
 - OOA : Object Oriented Analysis Schlaer, Mellor, 1992.
 - OOA/OOD : Coad, Yourdon, 1991.
 - OMT : Object Modeling Technique Rumbaugh, 1991.
 - OOSE : Object Oriented Software Engineering Jacobson, 1992.

- OOM : Object Oriented MeriseBouzeghoub, Rochfeld, 1993.
 - Fusion Coleman et al, 1994.
- 1994 : le consensus se fait autour de trois méthodes :
- OMT (Object Modeling Technique) de « James Rumbaugh » fournit une représentation graphique des aspects statique ,dynamique et fonctionnel d’un system.
 - OOD (Object Oriented Design) de « GradyBooch », définie pour le Département of Defense, introduit le concept de paquetage (package).
 - OOSE (Object Oriented Software Engineering) d’Ivra Jacobson fondel’analyse sur la description des besoins des utilisateurs.
- 1995 : Fusion des 3 principales méthodes pour définir un langage de modélisation commun : UML (Unified Modeling Language) [2].

2.2.3 LES VUES UML

Une façon de mettre en œuvre UML est de considérer différentes vues qui peuvent se superposer pour collaborer à la définition du système. Il y a cinq vues essentielles sur UML :

- **Vue logique (intégrité de conception) :** Perspective abstraite de la solution.
- **Vue des composants (intégrité de gestion du code) :** Perspective physique de l’organisation du code.
- **Vue des processus (intégrité d’exécution) :** Perspective sur les activités concurrentes et parallèles (tache et processus).
- **Vue des déploiements (intégrité de performance) :** Répartition du système (logiciel) à travers un réseau.
- **Vue des cas utilisation (intégrité de conception) :** Qui justifie les moyens rigoureux et systématiques de diriger la modélisation [3].

2.2.4 LES DIAGRAMMES UML

Les diagrammes sont des éléments graphiques décrivant le contenu des vues, qui sont des notions abstraites. Les diagrammes peuvent faire partie de plusieurs vues. UML comporte treize types de diagrammes qui représentent deux vues différentes du système.

2.2.4.1 La vue statique (structurelle)

Représente la structure statique du système en utilisant des objets, des attributs, des opérations et des relations. La vue statique comprend les diagrammes suivants :

— **Diagramme de classes (Class Diagram) :**

- Le diagramme de classes est un diagramme UML qui permet de modéliser les classes du system et leur relation (association, généralisation).
- Il fournit une représentation abstraite des objets du système qui vont interagir ensemble pour réaliser les fonctionnalités du système.
- Le diagramme de classes est considéré comme le plus important et le plus utilisé de la modélisation orientée objet.
- Le diagramme de classes présente la vue statique du système et ne tient pas compte des aspects dynamiques [2].

— **Diagramme d'objet (Object Diagram) :**

Ce type de diagramme UML montre des objets (instances de classes dans un état particulier) et des liens (relations sémantiques) entre ces objets. Les diagrammes d'objets s'utilisent pour montrer un contexte (avant ou après une interaction entre objets par exemple).

— **Diagramme de composants (Component Diagram) :**

Ce diagramme représente les différents constituants du logiciel au niveau de l'implémentation d'un système [4].

— **Diagramme de déploiement (Deployment Diagram) :**

Ce diagramme décrit l'architecture technique d'un système avec une vue centrée sur la répartition des composants dans la configuration d'exploitation [5].

— **Diagramme des paquetages (Package Diagram) :**

Ce diagramme donne une vue d'ensemble du système structuré en paquetage. Chaque paquetage représente un ensemble homogène d'éléments du système (classes, composant).

— **Diagramme de structure composite (Composite Structure Diagram) :**

Ce diagramme permet de décrire la structure interne d'un ensemble complexe Composé par exemple de classes ou d'objets et de composants techniques. Ce diagramme met aussi l'accent sur les liens entre les sous-ensembles qui collaborent [4].

2.2.4.2 La vue dynamique (comportementale)

Met l'accent sur le comportement dynamique du système en montrant la collaboration entre les objets et les modifications apportées à l'état interne des objets. Cette vue inclut les diagrammes suivants :

— **Diagramme des cas d'utilisation (Use Case Diagram) :**

- Est un diagramme UML. Il représente la structure des grandes fonctionnalités nécessaires aux utilisateurs du système.
- Le diagramme de cas d'utilisation permet de recueillir, d'analyser et organiser les besoins des utilisateurs.
- Le diagramme de cas d'utilisation est donc une vision orientée utilisateurs du système.
- Les éléments de modélisation des cas d'utilisation sont :
 1. **Acteur** : Entité externe qui agit sur le système.
 2. **Cas d'utilisation** : Ensemble d'actions réalisées par le système en réponse à une action d'un acteur.
 3. **Relations entre les cas d'utilisation** : UML définit trois types de relations standardisées entre cas d'utilisation, (Relation d'inclusion, Relation d'extension et Relation de généralisation/spécialisation).
 4. **Scénario** : Un cas d'utilisation est une abstraction de plusieurs chemins d'exécution. Une instance de cas d'utilisation est appelée : « scénario ». Il y'a trois types de scénario : scénario nominal, scénario alternatif et scénario d'erreur.

— **Diagramme d'états-transitions (State Machine Diagram) :**

Le diagramme d'états représente le cycle de vie commun aux objets d'une même classe. Ce diagramme complète la connaissance des classes en analyse et en conception [5].

— **Diagramme d'activité (Activity Diagram) :**

Il représente les règles d'enchaînement des activités et actions dans le système. Il permet d'une part de consolider la spécification d'un cas d'utilisation [6].

— **Diagramme de séquence (Sequence Diagram) :**

- Les diagrammes de séquences sont la représentation graphique des interactions entre les acteurs et les objets du système selon un ordre chronologique.
- Ils permettent de représenter un système pendant son exécution.
- On raisonne sur les objets.
- Les objets qui composent une application pendant son exécution et leurs échanges de messages permettent à l'application de réaliser les fonctionnalités pour laquelle elle est développée.
- Utilisés à tous les niveaux : l'analyse et la conception [2].

— **Diagramme de communication (Communication Diagram) :**

Représentation de façon simplifiée d'un diagramme de séquence se concentrant sur les échanges de messages entre les objets [5].

— **Diagramme global d'interaction (Interaction Overview Diagram) :**

Enchaînements possibles entre les scénarios préalablement identifiés sous forme de diagrammes de séquences (variante du diagramme d'activité).

— **Diagramme de temps (Timing Diagram) :**

Représentation des variations d'une donnée au cours du temps [6].

2.3 LE PROCESSUS UNIFIÉ (UP)

2.3.1 DEFINITION

C'est un processus de développement logiciel, il regroupe les activités à mener pour transformer les besoins d'un utilisateur en système logiciel. L'objectif d'un processus unifié est de maîtriser la complexité des projets informatiques en diminuant les risques [4].

2.3.2 LES CARACTÉRISTIQUES DU PROCESSUS UNIFIÉ

- **Itérative et incrémentale :** la méthode est itérative dans le sens où elle propose de faire des itérations lors de ses différentes phases, ceci garantit que le modèle construit à chaque phase ou étape soit affiné et amélioré. Chaque itération peut servir à ajouter de nouveaux incréments. Un incrément est une augmentation ou amélioration apportée à la construction en cours d'un système.
- **Centré sur l'architecture :** les modèles définis tout au long du processus de développement vont contribuer à établir une architecture cohérente et solide.
- **Conduite par les cas d'utilisation :** le but principal d'un système informatique est de satisfaire les besoins du client. Le processus de développement sera donc axé sur l'utilisateur. Les cas d'utilisation permettent d'illustrer ces besoins. Ils détectent puis décrivent les besoins fonctionnels (du point de vue de l'utilisateur), et leur ensemble constitue le modèle de cas d'utilisation qui dicte les fonctionnalités complètes du système.
- **Piloté par les risques :** il est piloté par les risques. Dans ce cadre, les causes majeures d'échec d'un projet logiciel doivent être écartées en priorité. Nous identifions une première cause provenant de l'incapacité de l'architecture technique à répondre aux contraintes opérationnelles, et une seconde cause liée à l'inadéquation du développement aux besoins des utilisateurs [5].

2.4 LE PROCESSUS 2TUP

2.4.1 DEFINITION

C'est un processus qui répond aux caractéristiques du processus unifié. Le processus 2TUP « 2 Track Unified Process » apporte une réponse aux contraintes de changement continu imposées aux systèmes d'information de l'entreprise. En ce sens, il renforce le contrôle sur les capacités d'évolution et de correction de tels systèmes.

2TUP est piloté par les risques, et utilise UML comme langage de modélisation car il y a une correspondance entre ses étapes et les différents diagrammes d'UML. Son idée de base est la suivante : tout développement imposé au système peut être configuré et traité en parallèle.

2Track signifie littéralement que le processus suit deux chemins. Il s'agit du « chemin fonctionnel » et d' « architecture technique », qui correspondent aux deux axes de changement imposés au système d'information.

2.4.2 LES BRANCHES DU PROCESSUS 2TUP

La réalisation du système 2TUP consiste à fusionner les résultats des deux branches en forme de Y :

2.4.2.1 La branche gauche (fonctionnelle)

Capitalise la connaissance du métier de l'entreprise. Elle constitue généralement un investissement pour le moyen et le long terme. Les fonctions du système d'information sont en effet indépendantes des technologies utilisées [7].

Cette branche comporte les étapes suivantes :

1. **capture des besoins fonctionnels** : Cette phase détaille la partie fonctionnelle indépendamment des technologies utilisées. Elle consiste à :
 - a) Identifier les cas d'utilisation.
 - b) Décrire les cas d'utilisation.
 - c) Structurer des cas d'utilisations dans des packages.
 - d) Identifier les classes candidates.
2. **Analyse** : L'étape qui consiste à étudier précisément la spécification fonctionnelle de manière à obtenir une idée de ce que va réaliser le système en termes de métier. Les résultats de l'analyse ne dépendent d'aucune technologie particulière. Elle comporte les étapes suivantes :
 - a) Découpage en catégorie.
 - b) Développement du modèle statique.

- c) Développement du modèle dynamique [8].

2.4.2.2 La branche droite (architecture technique)

Capitalise un savoir-faire technique. Elle constitue un investissement pour le court et moyen terme. Les techniques développées pour le système peuvent l'être en effet indépendamment des fonctions à réaliser [7].

Cette branche comporte les étapes suivantes :

1. **Capture des besoins techniques** : Cette phase comporte les étapes suivantes :
 - a) Spécification technique du point de vue matériel.
 - b) Spécification d'architecture.
 - c) Élaboration du modèle de spécification logicielle.
2. **La conception générique** : qui définit les composants nécessaires à la construction de l'architecture technique. Cette conception est complètement indépendante des aspects fonctionnels. Elle a pour objectif d'uniformiser et de réutiliser les mêmes mécanismes pour tout un système.

2.4.2.3 La branche du milieu

A l'issue des évolutions du modèle fonctionnel et de l'architecture technique, la réalisation du système consiste à fusionner les résultats des 2 branches. Cette fusion conduit à l'obtention d'un processus en forme de **Y**. Cette branche comporte les étapes suivantes :

1. **La conception préliminaire** : qui représente une étape délicate, car elle intègre le modèle d'analyse dans l'architecture technique de manière à tracer la cartographie des composants du système à développer. Elle comporte les étapes suivantes :
 - a) Développement du modèle de déploiement.
 - b) Développement du modèle d'exploitation.
2. **La conception détaillée** : cette phase s'appuie sur les catégories de conception organisées à la fois suivant les Frameworks techniques et les regroupements propres au métier. Elle comporte :
 - a) La conception des classes de système.
 - b) Le passage vers le modèle relationnel.
3. **Le codage** : produit des composants et tests au fur et à mesure que les unités de code sont réalisées.
4. **L'étape de recette** : qui consiste enfin à valider les fonctions du système développé [8].

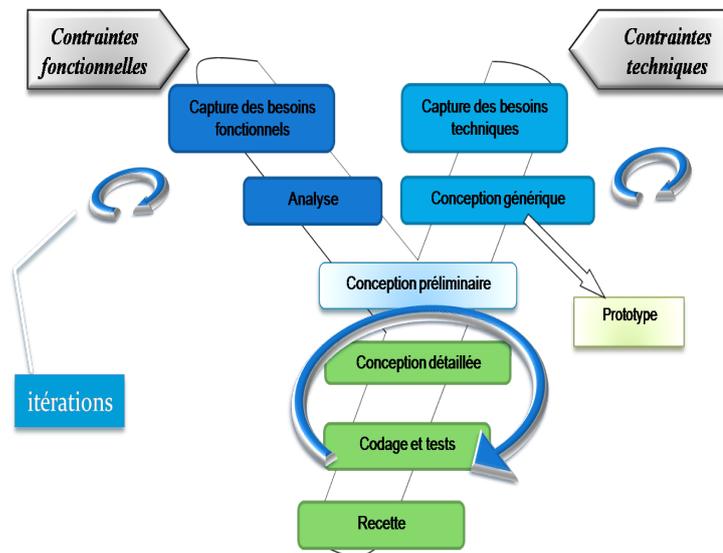


FIGURE 2.1 – Le processus de développement en Y.

2.5 INFORMATIQUE MOBILE

La mobilité caractérise ce qui peut se mouvoir ou être mû, ce qui peut changer de place, de position. Cette notion intuitive de l'activité humaine se traduit cependant par trois aspects différents, et autant d'approches, dans le domaine de l'informatique [9].

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 [10].

2.5.1 EQUIPEMENT DE L'INFORAMTIQUE MOBILE

2.5.1.1 Principaux équipements mobiles

Beaucoup de concepts de l'informatique mobile sont devenus une réalité. Cela est principalement le résultat de l'évolution énorme et la haute pénétration des technologies mobile. Aujourd'hui, les équipements mobiles sont plus puissants, plus riches en fonctionnalités et malgré cela ils coutent moins chères.

- **Les ordinateurs portables** : ce sont des ordinateurs personnels qui possèdent les mêmes fonctionnalités d'un ordinateur personnel de bureau mais qui ont une taille réduite (laptops, notebooks, netbooks). Ils sont conçus à l'origine pour remplacer les ordinateurs de bureau [11].



FIGURE 2.2 – Ordinateur portable.

- **Les smartphones :**Le smartphone ou «téléphone intelligent» désigne un téléphone mobile doté de fonctionnalités évoluées qui s'apparentent à celles d'un ordinateur : navigation sur Internet, lecture de vidéos, de musique, jeux vidéo, courrier électronique, vidéoconférence, bureautique légère [12].



FIGURE 2.3 – Les smartphones.

- **Historiques :**

la vague de smartphones arrivée en 2007 avec le premier iPhone n'est pas la première à avoir existé. L'iPhone et ses concurrents ont été imaginés à partir d'une autre génération de téléphones et assistants personnels, qu'ils sont en train d'écrasée commercialement aujourd'hui. D'autres appareils ont ainsi longtemps porté ce nom et offert des capacités très similaires, en fonctionnant sur des systèmes d'exploitation encore en circulation en 2012.

Les fonctions emblématiques du smartphone actuel, comme l'écran tactile, la connexion Internet, le GPS ou le clavier physique viennent directement d'appareils nés dans les années 1990. Des premiers Simon d'IBM en 1993 aux célèbres assistants personnels (PDA) de Palm en passant par le Nokia 9000 « Communicator », les fondations ont été posées petit à petit par des sociétés qu'on retrouve aujourd'hui sur le marché, sous leur nom ou celui de leur nouveau propriétaire. Les systèmes fournissant ces fonctions sont aussi majoritairement des évolutions des efforts menés dans les années 1990.

Et pour beaucoup, le téléphone portable est devenu un objet indispensable du quotidien. Pour communiquer, échanger avec les autres, mais aussi suivre l'actualité ou même se divertir [13].

— Les tablettes :

Une tablette est un équipement personnel mobile qui ressemble à un Smartphone mais qui possède un écran plus large. Avec son écran plus proche à ceux des ordinateurs portable, ce dernier c'est un écran tactile et aussi mieux adapté à certaines tâches comme la navigation internet, etc. La tablette cependant, partage la spécification technique, l'architecture et la même plateforme logicielles avec les smartphones.



FIGURE 2.4 – Tablette.

— PDA (Personal Digital Assistant) :

Un équipement portable qui offre un ensemble d'outils pour assister l'utilisateur dans les tâches de bureau et améliore sa productivité comme la gestion des contacts. Le PDA est doté d'un écran tactile. Fourni avec un stylo pour écrire du texte à la main [11].



FIGURE 2.5 – PDA.

2.5.2 SYSTEMES D'EXPLOITATION MOBILES

2.5.2.1 Définition

Tout comme un ordinateur dispose d'un système d'exploitation, les téléphones mobiles se composent également d'une plate-forme qui contrôle toutes ses fonctionnalités. Ceci est connu comme un système d'exploitation mobile. Généralement connu sous le nom D'OS (Operating System) mobile, il s'agit d'un système d'exploitation qui exploite un appareil mobile (Smartphone, tablette, etc.) [14].

2.5.2.2 Les parts de marché des systèmes d'exploitation mobiles

Le marché des téléphones mobile est aujourd'hui dominé par cinq grandes entreprises de technologie Smartphone qui sont Apple, RIM, Google, Microsoft et Nokia qui développent respectivement les systèmes d'exploitation iOS, BlackBerry OS, Android, Windows Phone et Symbian OS.

- **IOS** :le système d'exploitation d'Apple qui équipe exclusivement les iPhone et iPad [14].
- **Symbian OS (Nokia)** :Symbian est le système d'exploitation historique des premiers téléphones Nokia et Motorola.
- **Windows** :le système d'exploitation de l'entreprise américaine à l'origine du même système d'exploitation pour ordinateurs portables et fixes.
- **BlackBerry OS** :le système d'exploitation développé par BlackBerry qui équipe exclusivement les téléphones et smartphones BlackBerry.
- **Android** :le système d'exploitation de Google qui équipe la majorité des smartphones et tablettes d'aujourd'hui [15].



FIGURE 2.6 – Logo Android OS.

2.5.3 ANDROID

2.5.3.1 DEFINITION

Android est une pile logicielle open source créée pour un large éventail de périphériques avec différents facteurs de forme. Les principaux objectifs d'Android sont de créer une plate-forme logicielle ouverte disponible pour les opérateurs, les constructeurs OEM (Original Equipment Manufacturer) et les développeurs afin de concrétiser leurs idées innovantes et de produit réussi et réel qui améliore l'expérience mobile des utilisateurs [11].

2.5.3.2 HISTORIQUE

L'histoire d'Android commence en octobre 2003 lorsque Andy Rubin, Rich Miner, Nick Sears and Chris White Fondèrent une startup nommée AndroidInc en Californie. Ils voulaient au début développer les Systèmes d'exploitation pour les camera numériques. Le but final était de pouvoir fournir des applications tierces sur ces équipements. Comme le marché n'était pas important ils ont envisagé de développer un système pour mobile.

Google voulait que ses services soient utilisés sur des plateformes mobiles. Pour cela en 2005 Google rachète cette compagnie. Le 5 novembre 2007 Google annonce officiellement le développement d'Android dans le cadre d'un projet open source mené par un consortium de 34 compagnies de divers secteurs, l'OHA (Open Handset Alliance) [11].

Le tableau suivant résume l'évolution de cette plateforme de la version 1.0 à la version 9.0.

Version	Logo	Nom	Date	API
1.0	-	Aucun	11 novembre 2007	1
1.1	-	Aucun	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	5-6-7
2.2		Froyo	20 mai 2010	8
2.3		Gingerbread	6 décembre 2010	9-10
3.0-3.2		Honeycomb	22 février 2011	11-12-13

4.0		Icecream	19 octobre 2011	14-15
4.1-4.2-4.3		Jelly Bean	24 juillet 2013	16-17-18
4.4		Kitkat	31 octobre 2013	19-20
5.0-5.1		Lollipop	9 mars 2015	21-22
6.0		Marshmallow	5 octobre 2015	23
7.0-7.1		Nougat	22 août 2016	24-25
8.0		Oreo	22 août 2017	26-27
9.0		Pie	7 août 2018	28

TABLE 2.1 – Les versions d'Android.

2.5.3.3 LES OBJECTIFS D'ANDROID

1. Open Source

Android utilise des bibliothèques open source puissantes, comme par exemple SQLite.

2. Facile à développer

Toutes les API mises à disposition facilitent et accélèrent grandement le travail. Ces APIs sont très complètes et très faciles d'accès.

3. Flexible

Le système est extrêmement portable, il s'adapte à beaucoup de structures différentes.

(ex : les Smartphones, les tablettes, etc.)

4. Complémentaire

L'architecture d'Android est inspirée par les applications composites. Ces Applications se trouvent essentiellement sur internet et basé sur le but de partagée plusieurs composants pour obtenir un résultat surpuissant [14].

2.5.4 ARCHITECTURE D'ANDROID

La plateforme Android est une pile de couches logicielles. Elle est composée de couches suivantes :

- **Le noyau** :cette couche assure des fonctions de base tel que la gestion de la couche matérielle (pilotes), la gestion de la mémoire, la gestion des processus, les E/S(Binder).
- **La couche Hal** :couche mince au-dessus du noyau, la couche d'abstraction Hal défini une interface standard qui relie les couches supérieures avec les pilotes du matériel. Cette standardisation permet aux couches supérieures d'ignorer les détails du matériel ou de l'implémentation des pilotes et permet aussi aux développeurs de pilotes une marge plus importante. Les implémentations Hal sont conçu sous forme de modules (fichier .So) qui seront chargés par le système quand nécessaire.
- **La couche Librairies** :un ensemble de bibliothèques écrites en C/C++ qui sont derrière toutes les fonctionnalités d'Android .Cette couche permet aux couches supérieures d'accéder aux services du noyau.
- **L'environnement d'exécution/Android Runtime** :la couche runtime d'Android a été conçue pour correspondre aux besoins d'exécution dans les environnements embarqués.elle comprend deux éléments « la machine virtuelle ART/ex DALVIK » et « Bibliothèque de base /Core librairies ».
- **Application Framework** :écrite entièrement en java, cette couche offre un ensemble de services aux applications et fournie une abstraction pour les couches inférieures. Cet ensemble reflète toutes les capacités de la plateforme Android, il comprend deux éléments « **Services de base** » et « **Services matériels** ».
- **Applications** :cette couche regroupe l'ensemble des applications disponibles sur l'équipement mobile. Cet ensemble comprend les applications livré avec la plateforme « Natives » et les applications déployées par l'utilisateur. Android offre les mêmes APIs à toutes les applications, donc chaque application native peut être facilement remplacé et améliorée [11].

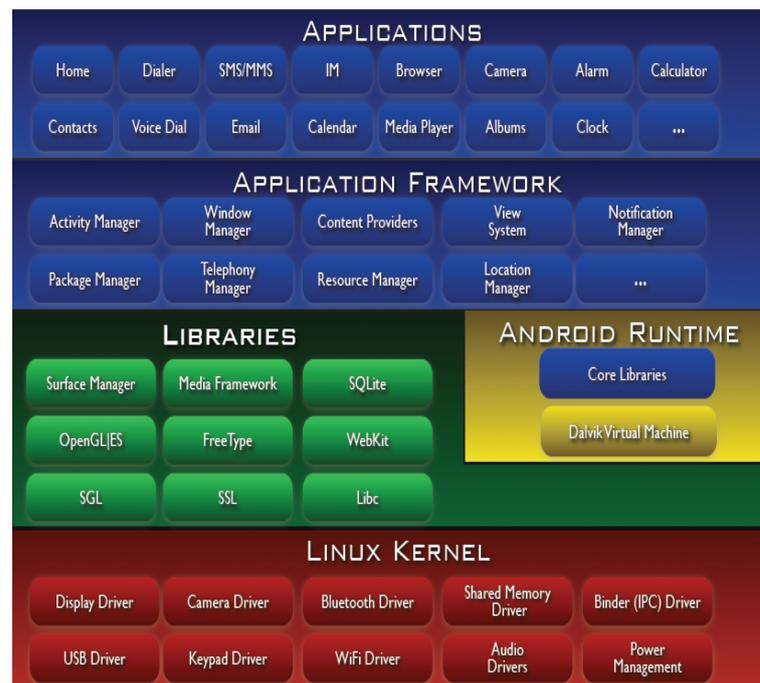


FIGURE 2.7 – Architecture d'Android.

2.6 APPLICATION MOBILE

Les applications pour mobiles sont entrées en scène en 2008 au moment où Apple a lancé son App Store, un service de téléchargement en ligne d'applications destinées à l'iPhone.

2.6.1 QU'EST-CE QU'UNE APPLICATION MOBILE ?

Une application mobile est une application développée spécifiquement pour une Plateforme, c'est un programme téléchargeable de façon gratuite ou payante et exécutable à partir du système d'exploitation du téléphone.

Elle permet généralement un accès plus confortable et plus efficace à des sites accessibles par ailleurs en versions mobile ou web. Reflet du succès commercial et technique pour les applications Android en ce moment.

2.6.2 QU'EST-CE QU'UN SITE MOBILE ?

Un site mobile est un site internet qui est optimisé pour les smartphones. Afin de consulter un site mobile, on doit être connecté sur internet et on y accède via le navigateur web du smartphone. Le référencement est primordial, de la même manière qu'un site web. Le langage de développement est également similaire à celui utilisé pour développer un site internet [14].

2.6.3 TYPES D'APPLICATIONS MOBILES

Il existe 3 types des applications mobiles :

- **Applications natives**

Une application native peut accéder à tous les API (Application Programming Interfaces) du système. Pour la développer on utilise souvent un ensemble d'outils et des ressources fournis par les mêmes fournisseurs du système. Cet ensemble d'outils est appelé « Kit de Développement » ou SDK (Software Development Kit).

Une application native est écrite à l'aide d'un langage de haut niveau comme java, C ou C++. Comme chaque système possède son propre SDK, il ne sera pas possible de porter une application native d'une plateforme à l'autre. Cela fait que la création ou la maintenance d'applications natives coutent plus et prennent beaucoup plus de temps [16].

- **Applications web**

Les langages web ont beaucoup évolué ces dernières années. HTML par exemple n'est plus un simple langage de description de document, il inclut maintenant des composants avancés d'interface graphique.

Les technologies du web actuelles permettent de créer des applications avancées. A la différence d'un site web, une application web apparait comme une application ordinaire qu'on peut lancer d'un simple raccourci et sans fournir d'URL [15].

- **Applications hybrides**

Les applications hybrides combinent les deux types, native et web. Le but est de bénéficier des avantages des deux types. Selon ce type, le développeur réalise une grande partie de l'application à l'aide des technologies web tout en profitant de l'accès direct aux API du système quand nécessaire à l'aide d'une partie native de l'application. Cette partie servira de pont entre les API du système et la partie Web. Les développeurs ont le choix de développer eux-mêmes ce pont, ou d'utiliser des solutions existantes et très répondues (Phone Gap) [16].

2.6.4 AVANTAGES ET INCONVINAIT D'UNE APPLICATION MOBILE

- **Les avantages d'une application mobile**

Le principal avantage d'une application mobile comparé à un site mobile est son ergonomie : elle est de bien meilleure qualité et de ce fait, l'expérience utilisateur est supérieure comparée à celle obtenue avec un site mobile. Ceci vient du fait qu'en développant une application mobile, le format, la navigation et le contenu sont adaptés au

Smartphone. L'application permet également d'utiliser et d'intégrer les fonctionnalités du téléphone ce qui peut rendre l'utilisation très riche.

Autre avantage, pour fonctionner une application, ce n'est pas nécessaire une connexion à internet.

Une application mobile est également plus appréciée du grand public car il est plus facile de trouver une application qu'un site mobile avec les « stores » proposés par Apple, Google, BlackBerry. Les applications permettent aussi d'alerter les mobinautes sur des évènements via des notifications [14].

- **Les inconvénients d'une application mobile**

Le principal inconvénient d'une application mobile c'est qu'elle doit respecter les règles définies par les différentes sociétés des plateformes mobiles.

Un autre inconvénient est le coût lié au développement d'une application. En moyenne, il est plus coûteux de développer une application mobile destinée à être utilisable sur plusieurs plateformes que de faire développer un site mobile.

Autre inconvénient, pour que l'utilisateur ait accès à la dernière version, il faut qu'il mette à jour depuis le store contrairement aux sites mobiles qui se mettent à jour automatiquement [14].

2.7 CONCLUSION

Dans ce chapitre, nous avons passé en revue le langage de modélisation UML, puis nous avons présenté le processus 2TUP que nous allons utiliser pour concevoir notre application. Ensuite, nous avons défini la technologie de l'informatique mobile et ses systèmes d'exploitation, en particulier le système Android. Nous avons terminé ce chapitre par une vue générale sur les applications mobiles et leurs types.

Ce chapitre prépare les différentes phases de conception qui sont l'objet des prochains chapitres.

Chapitre 3

Étude préliminaire et Capture des besoins

3.1 INTRODUCTION

Le but de ce chapitre est de réaliser les étapes de l'étude préliminaire, la capture des besoins fonctionnels et la capture des besoins techniques du processus de développement 2TUP. Ces étapes vont permettre de décrire les besoins et les acteurs qui vont interagir avec notre futur système et de présenter un recueil des besoins fonctionnels et techniques du système.

3.2 ETUDE PRELIMINAIRE

L'étude préliminaire est la phase d'analyse dans la méthode 2TUP qui précède directement l'analyse proprement dite effectuée au niveau de la branche fonctionnelle du cycle de développement en Y. Elle consiste à effectuer un premier repérage des besoins fonctionnels et opérationnels, en utilisant principalement le texte, ou des diagrammes très simples.

3.2.1 PRESENTATION DU PROJET

Dans le cadre de notre projet de fin d'études, nous avons eu comme tâche de concevoir et développer un système informatique au sein de l'école « El Ameed School ». Ce système est composé d'une application mobile et une application web.

L'application mobile est conçue pour faciliter les tâches d'enseignement et garantir la communication entre l'administration, les étudiants et les enseignants.

L'application web est conçue pour faciliter les fonctions de gestion de l'administration.

3.2.2 DESCRIPTION DU CONTEXTE

3.2.2.1 Identification des acteurs

Un acteur est un utilisateur humain, un dispositif matériel ou un système qui interagit directement avec le système étudié. Il représente un rôle joué par une entité externe au système, donc il peut consulter ou modifier directement l'état du système. Il émet ou reçoit des messages qui sont porteurs de données.

Nous avons identifié les acteurs suivants :

- **Directeur des études** :son rôle est de :
 - Gérer les enseignants.
 - Gérer les étudiants.
 - Gérer les formations.
 - Publier des annonces.

- **Enseignant** : son rôle est de :
 - Gérer ses ressources (cours et multimédia).
 - Gérer ses notes.
 - Afficher des notes.
 - Publier des annonces.
 - Lancer des évènements.
 - Consulter les participations.
 - Afficher les résultats.
 - Consulter l'espace de communication.
 - Ecrire un message.
- **Etudiant** : son rôle est de :
 - Accéder aux ressources.
 - Evaluer le pourcentage de compréhension.
 - Télécharger des ressources.
 - Consulter les annonces.
 - Consulter les évènements.
 - Participer aux évènements.
 - Consulter les résultats.
 - Consulter l'espace de communication.
 - Ecrire un message.

3.2.2.2 Diagramme du contexte dynamique

Le diagramme de contexte dynamique représente les messages échangés entre l'application et les acteurs identifiés (représentés de façon synthétique sur un diagramme). Chaque numéro, dans le diagramme, correspond à un message dans la table de messages représentée par le tableau 2.

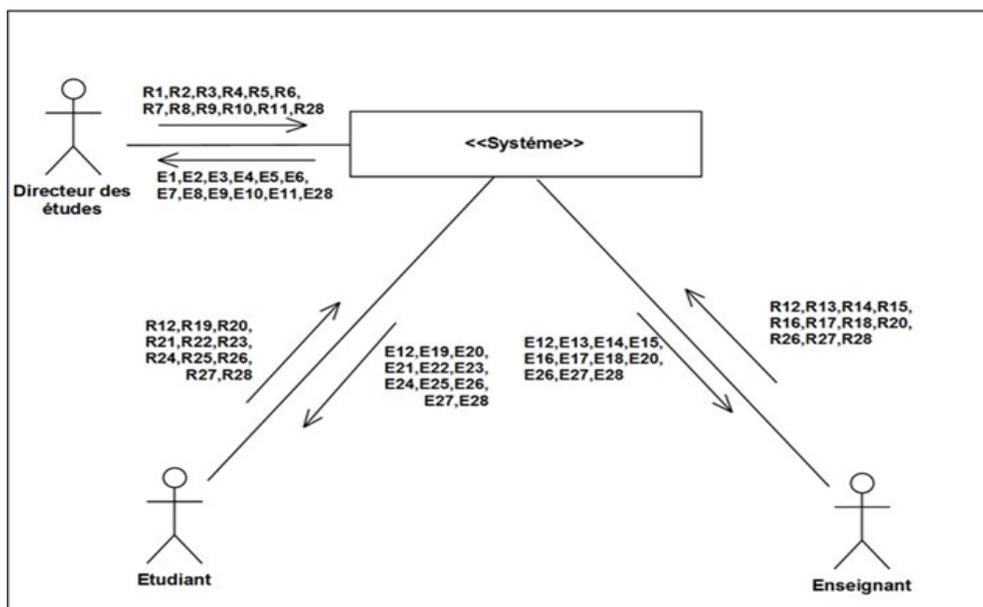


FIGURE 3.1 – Diagramme de contexte dynamique.

3.2.2.3 Signification des messages

Les messages émis	Les messages reçus
E1 : Confirmation d'ajout d'un enseignant.	R1 : Demande d'ajout un enseignant.
E2 : Confirmation de modification d'un enseignant.	R2 : Demande de modification d'un enseignant.
E3 : Confirmation de suppression d'un enseignant	R3 : Demande de suppression d'un enseignant.
E4 : Confirmation d'ajout d'un étudiant.	R4 : Demande d'ajout d'un étudiant.
E5 : Confirmation de modification d'un étudiant.	R5 : Demande de modification d'un étudiant.
E6 : Confirmation de suppression d'un étudiant.	R6 : Demande de suppression d'un étudiant.

<p>E7 : Affichage des informations des enseignants.</p> <p>E8 : Affichage des informations des étudiants.</p> <p>E9 : Confirmation d'ajout d'une formation.</p> <p>E10 : Confirmation de modification d'une formation.</p> <p>E11 : Confirmation de suppression d'une formation.</p> <p>E12 : Affichage des informations de formations.</p> <p>E13 : Confirmation de mise à jour des notes.</p> <p>E14 : Confirmation de publication d'une annonce.</p> <p>E15 : Confirmation d'ajout d'une ressource.</p> <p>E16 : Confirmation de suppression d'une ressource.</p> <p>E17 : Confirmation d'affichage des notes.</p> <p>E18 : Confirmation de lancement d'un évènement.</p> <p>E19 : Confirmation de participation dans un événement.</p> <p>E20 : Affichage du résultat d'un évènement.</p> <p>E21 : Confirmation d'évaluation du pourcentage de compréhension.</p>	<p>R7 : Demande de consultation des enseignants.</p> <p>R8 : Demande de consultation des étudiants.</p> <p>R9 : Demande d'ajout d'une formation.</p> <p>R10 : Demande de modification d'une formation.</p> <p>R11 : Demande de suppression d'une formation.</p> <p>R12 : Demande de consultation des formations.</p> <p>R13 : Demande de mise à jour des notes.</p> <p>R14 : Demande de publication d'une annonce.</p> <p>R15 : Demande d'ajout d'une ressource.</p> <p>R16 : Demande de suppression d'une ressource.</p> <p>R17 : Demande d'affichage de notes.</p> <p>R18 : Demande de lancement d'un évènement.</p> <p>R19 : Demande de participation dans un événement.</p> <p>R20 : Demande de consultation du résultat d'un évènement.</p> <p>R21 : Demande d'évaluation du pourcentage de compréhension.</p>
--	--

Les messages émis	Les messages reçus
E22 : Envoie de la ressource.	R22 : Demande de téléchargement d'une ressource.
E23 : Affichage des annonces	R23 : Demande de consultation des annonces.
E24 : Affichage des évènements.	R24 : Demande de consultation des évènements.
E25 : Affichage des ressources.	R25 : Demande d'accès aux ressources.
E26 : Affichage de l'espace de communication.	R26 : Demande de consultation de l'espace de communication.
E27 : Confirmation d'écriture d'un message.	R27 : Demande d'écriture d'un message.
E28 : Affichage de session.	R28 : Demande d'authentification.

TABLE 3.1 – Signification des messages.

3.3 CAPTURE DES BESOINS FONCTIONNELS

La capture des besoins fonctionnels est la première étape de la branche gauche du processus en Y. Elle formalise et détaille ce qui a été ébauché au cours de l'étude préliminaire.

Cette phase explique comment identifier les besoins exprimés selon le métier des utilisateurs, les reformuler, les structurer et les documenter avec UML. Elle s'appuie pour une large part sur la technique des cas d'utilisation.

3.3.1 DIAGRAMME DU CAS D'UTILISATION

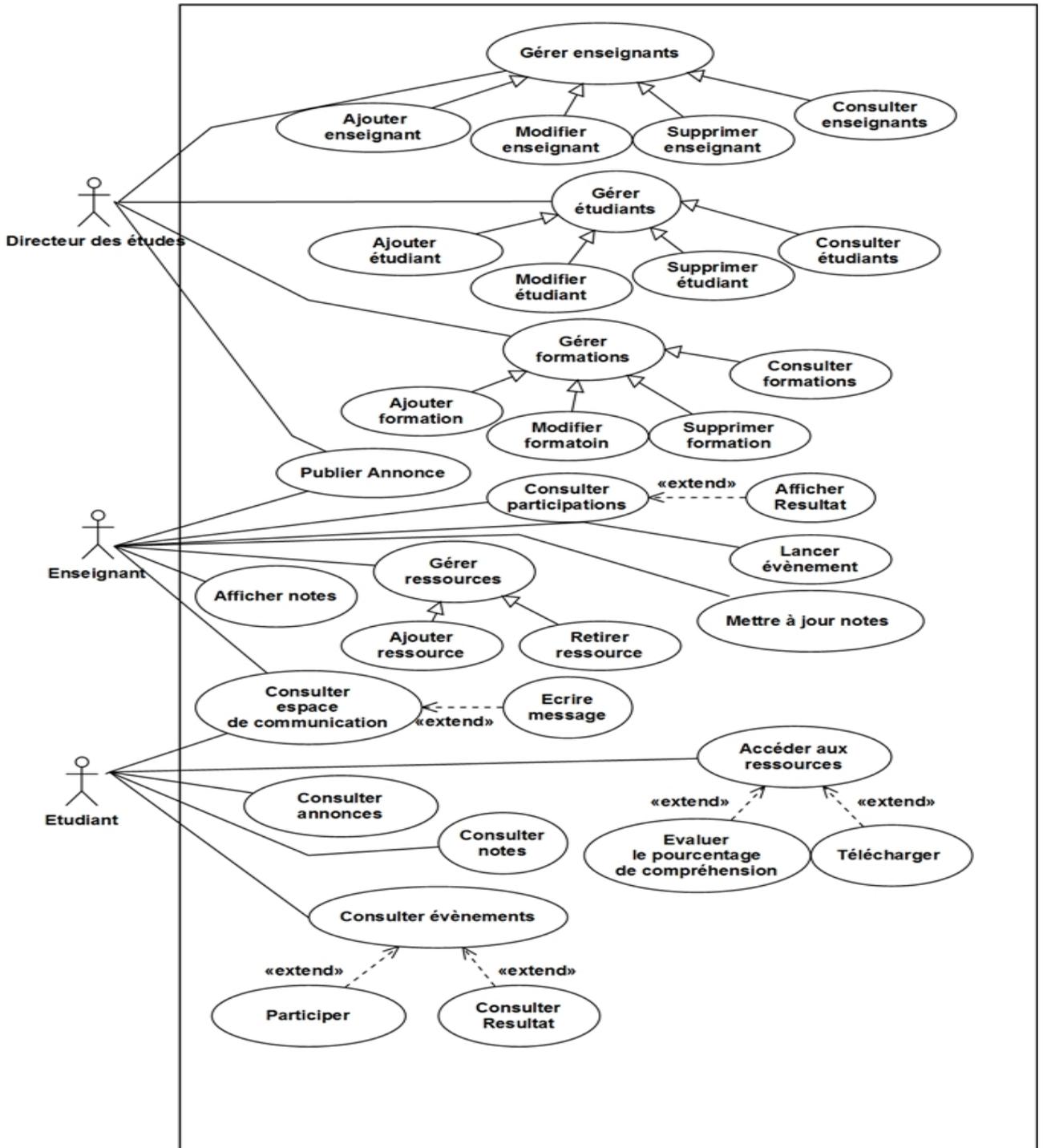


FIGURE 3.2 – Diagramme des cas d'utilisations

3.3.2 DESCRIPTION DETAILLE DU CAS D'UTILISATION

3.3.2.1 Cas d'utilisation «Gérer enseignants »

Nom	Gérer enseignants
Acteur	Directeur des études
But	Permet au directeur des études de consulter, d'ajouter, de modifier ou de supprimer un enseignant.
Pré-condition	Le directeur des études doit être authentifié.
Post-condition	/
Scénario nominal	<ol style="list-style-type: none"> 1. Le directeur des études demande de gérer les enseignants. 2. Le système affiche les options suivantes : <ul style="list-style-type: none"> • Consulter enseignants. • Ajouter enseignant. • Modifier enseignant. • Supprimer enseignant.

TABLE 3.2 – Description textuelle du cas d'utilisation « Gérer enseignants ».

3.3.2.2 Cas d'utilisation «Ajouter enseignant »

Nom	Ajouter enseignant
Acteur	Directeur des études
But	Permet au directeur des études d'ajouter un enseignant.
Pré-condition	Le directeur des études doit être authentifié.
Post-condition	Nouvel enseignant ajouté par le système.
Scénario nominal	<ol style="list-style-type: none"> 1. Le directeur des études demande d'ajouter un enseignant. 2. Le système affiche le formulaire correspondant. 3. Le directeur des études remplit le formulaire et valide. 4. Le système vérifie les informations entrées. 5. Le système enregistre le nouvel enseignant dans la base de données. 6. Le système affiche un message de confirmation.

Scénario Alternatif	<p>A1. Le directeur des études n'a pas rempli certains champs ou a saisi des informations non valides.</p> <ol style="list-style-type: none"> 1. Le système affiche un message d'erreur. 2. Le système retourne à l'étape numéro 2 du scénario nominal. <p>A2. L'enseignant existe déjà dans la base de données.</p> <ol style="list-style-type: none"> 1. Le système affiche un message d'erreur. 2. Le système retourne à l'étape numéro 2 du scénario nominal.
----------------------------	---

TABLE 3.3 – Description textuelle du cas d'utilisation «Ajouter enseignant ».

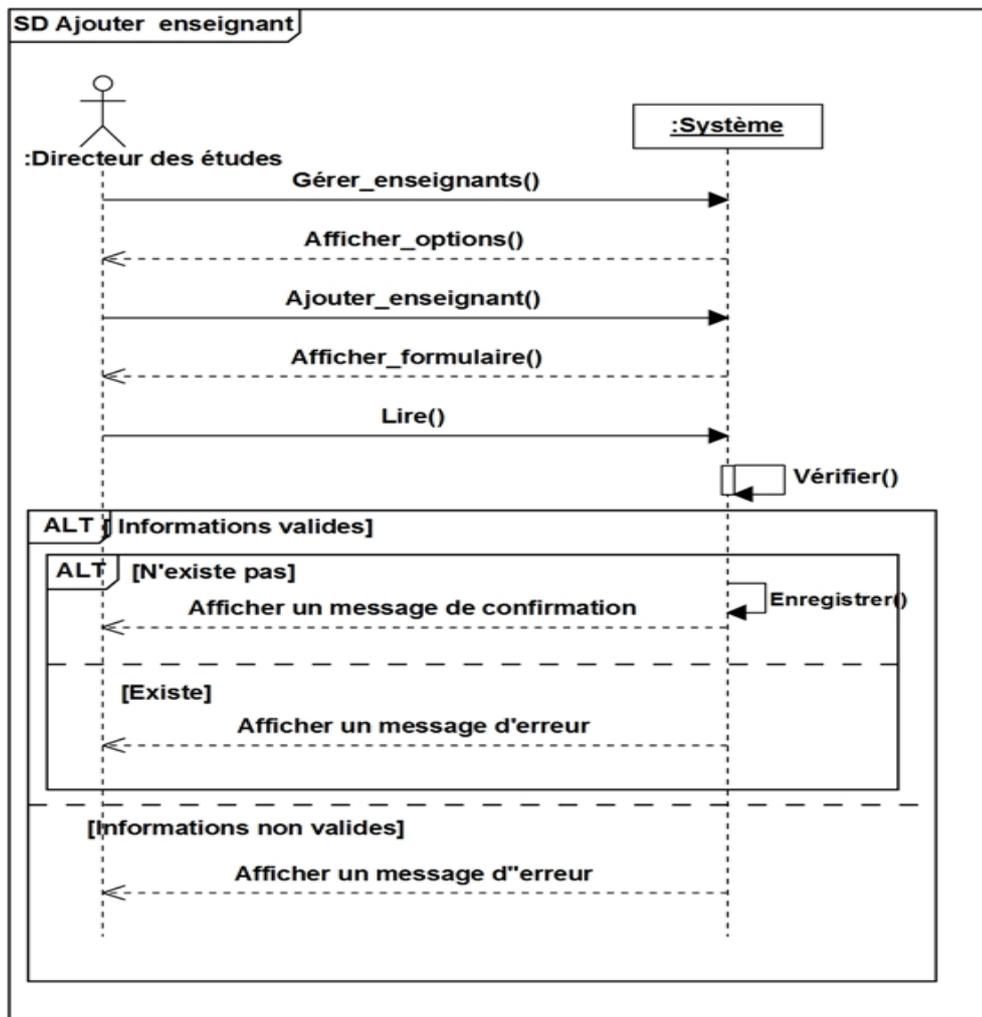


FIGURE 3.3 – Diagramme de séquence «Ajouter enseignant».

3.3.2.3 Cas d'utilisation «Modifier enseignant»

Nom	Modifier enseignant
Acteur	Directeur des études
But	Permet au directeur des études de modifier un enseignant.
Pré-condition	Le directeur des études doit être authentifié.
Post-condition	Les modifications sont enregistrées par le système.
Scénario nominal	<ol style="list-style-type: none"> 1. Le directeur des études demande de modifier un enseignant. 2. Le système affiche la liste des enseignants. 3. Le directeur des études sélectionne un enseignant. 4. Le système affiche les informations d'un enseignant. 5. Le directeur des études modifie et valide. 6. Le système vérifie les informations entrées. 7. Le système enregistre les modifications dans la base de données. 8. Le système affiche un message de confirmation.
Scénario Alternatif	<p>A1. Le directeur des études n'a pas rempli certains champs ou a saisi des informations non valides.</p> <ol style="list-style-type: none"> 1. Le système affiche un message d'erreur. 2. Le système retourne à l'étape numéro 4 du scénario nominal.

TABLE 3.4 – Description textuelle du cas d'utilisation «Modifier enseignant ».

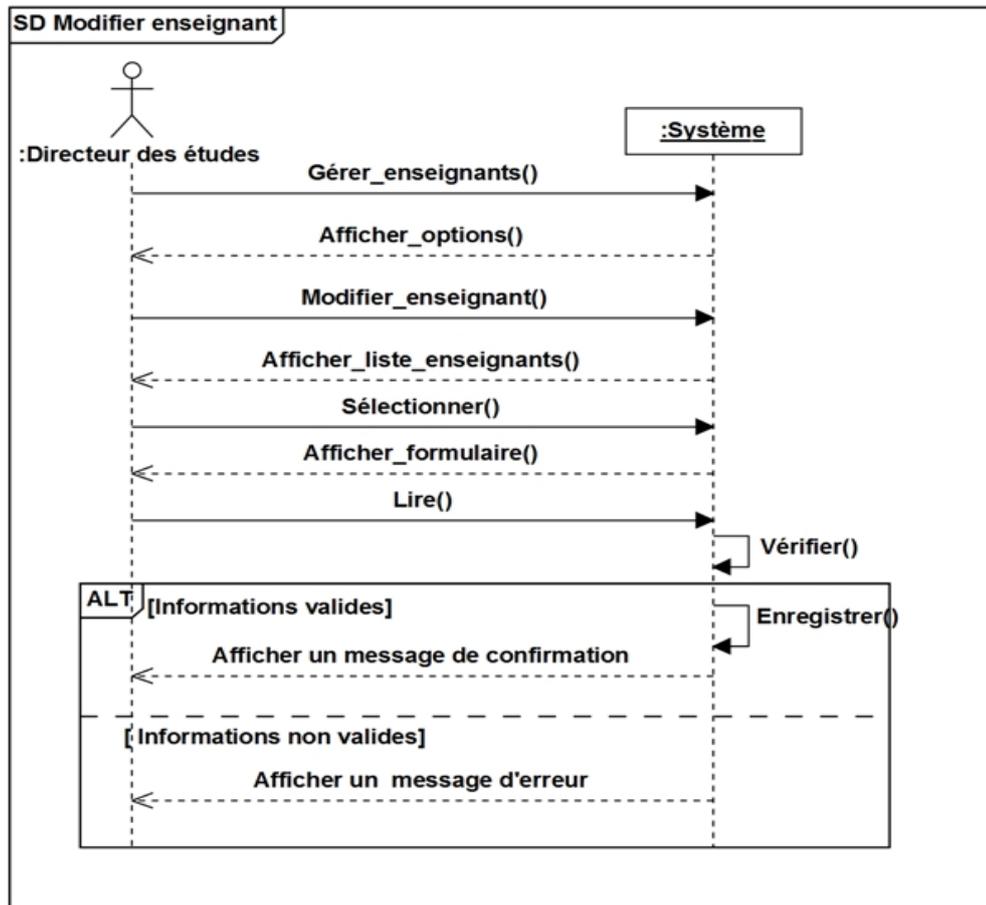


FIGURE 3.4 – Diagramme de séquence «Modifier enseignant».

3.3.2.4 Cas d'utilisation «Supprimer enseignant»

Nom	Supprimer enseignant
Acteur	Directeur des études
But	Permet au directeur des études de supprimer un enseignant.
Pré-condition	Le directeur des études doit être authentifié.
Post-condition	Enseignant supprimé du système.
Scénario nominal	<ol style="list-style-type: none"> 1. Le directeur des études demande de supprimer un enseignant. 2. Le système affiche la liste des enseignants. 3. Le directeur des études sélectionne un enseignant.

	<p>4. Le système demande de confirmer la suppression.</p> <p>5. Le directeur des études confirme la suppression.</p> <p>6. Le système supprime l’enseignant de la base de données.</p> <p>7. Le système affiche un message de confirmation.</p>
Scénario Alternatif	<p>A1. Le directeur des études ne confirme pas la suppression.</p> <p>1. Le système retourne à l’étape numéro 2 du scénario nominal.</p>

TABLE 3.5 – Description textuelle du cas d’utilisation «Supprimer enseignant».

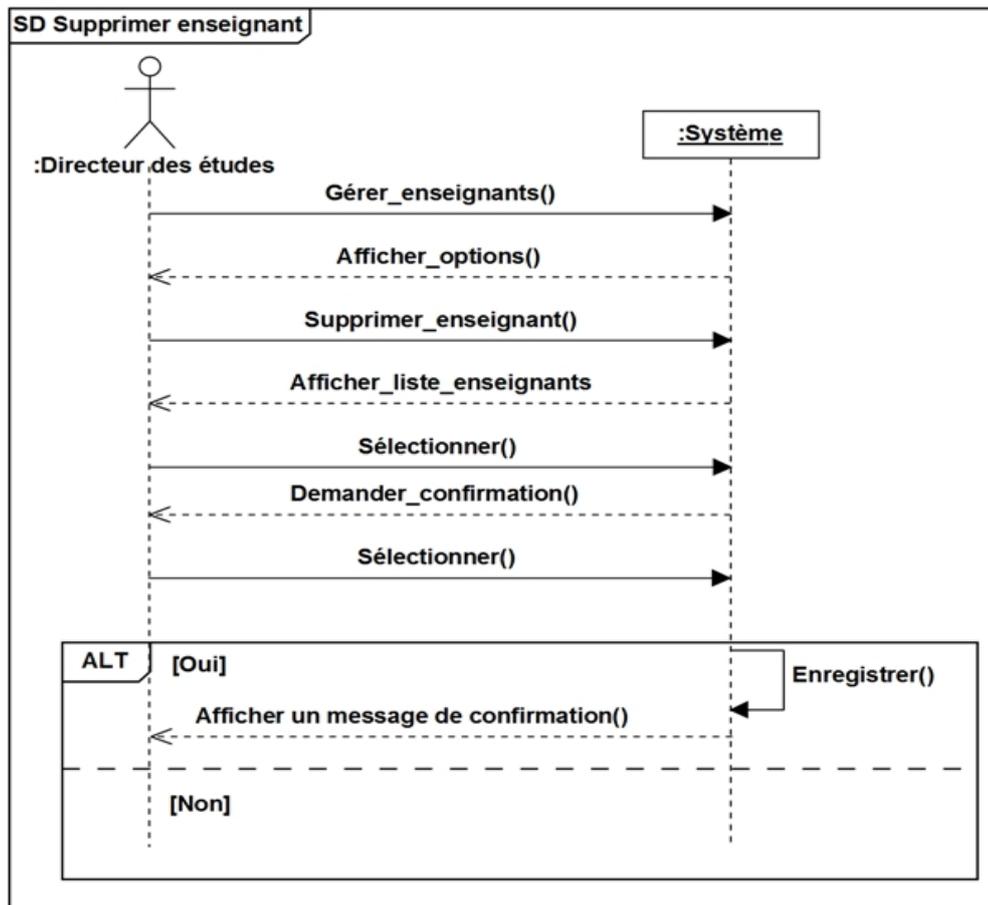


FIGURE 3.5 – Diagramme de séquence «Supprimer enseignant».

3.3.2.5 Cas d'utilisation «Consulter enseignants»

Nom	Consulter enseignants
Acteur	Directeur des études
But	Permet au directeur des études de consulter les informations des enseignants.
Pré-condition	Directeur des études doit être authentifié.
Post-condition	Informations des enseignants affichées.
Scénario nominal	<ol style="list-style-type: none"> 1. Le directeur des études demande de consulter les enseignants. 2. Le système affiche la liste des enseignants. 3. Le directeur des études choisit un enseignant. 4. Le système affiche les informations de l'enseignant.

TABLE 3.6 – Description textuelle du cas d'utilisation «Consulter enseignants».

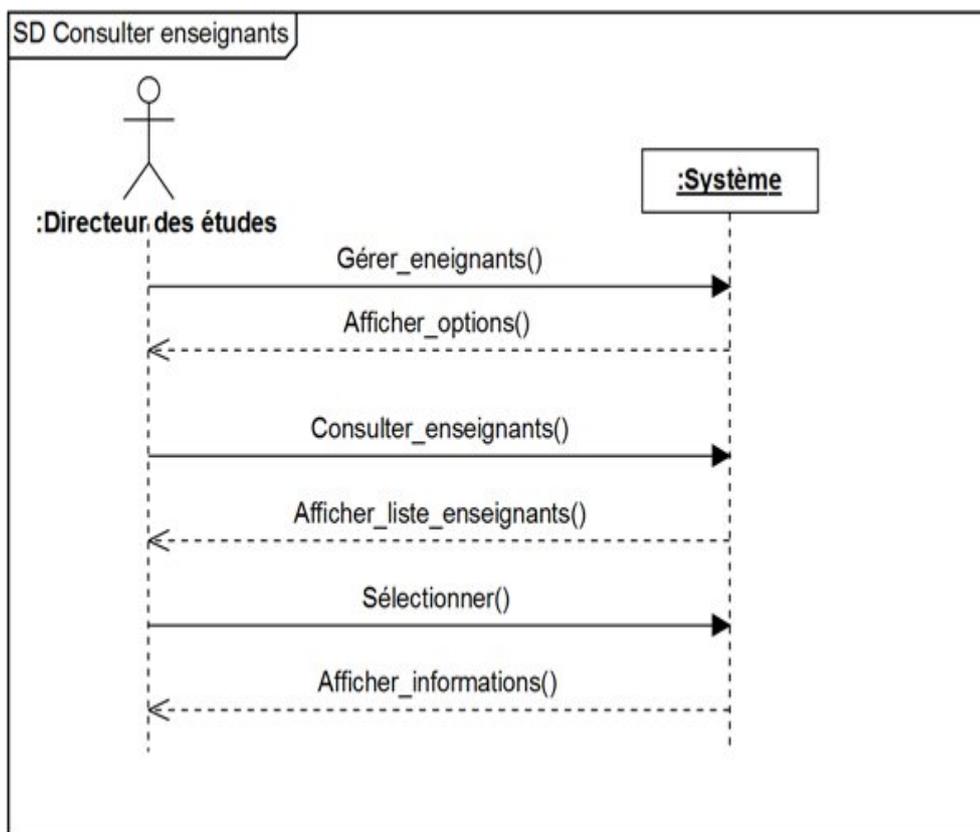


FIGURE 3.6 – Diagramme de séquence «Consulter enseignants».

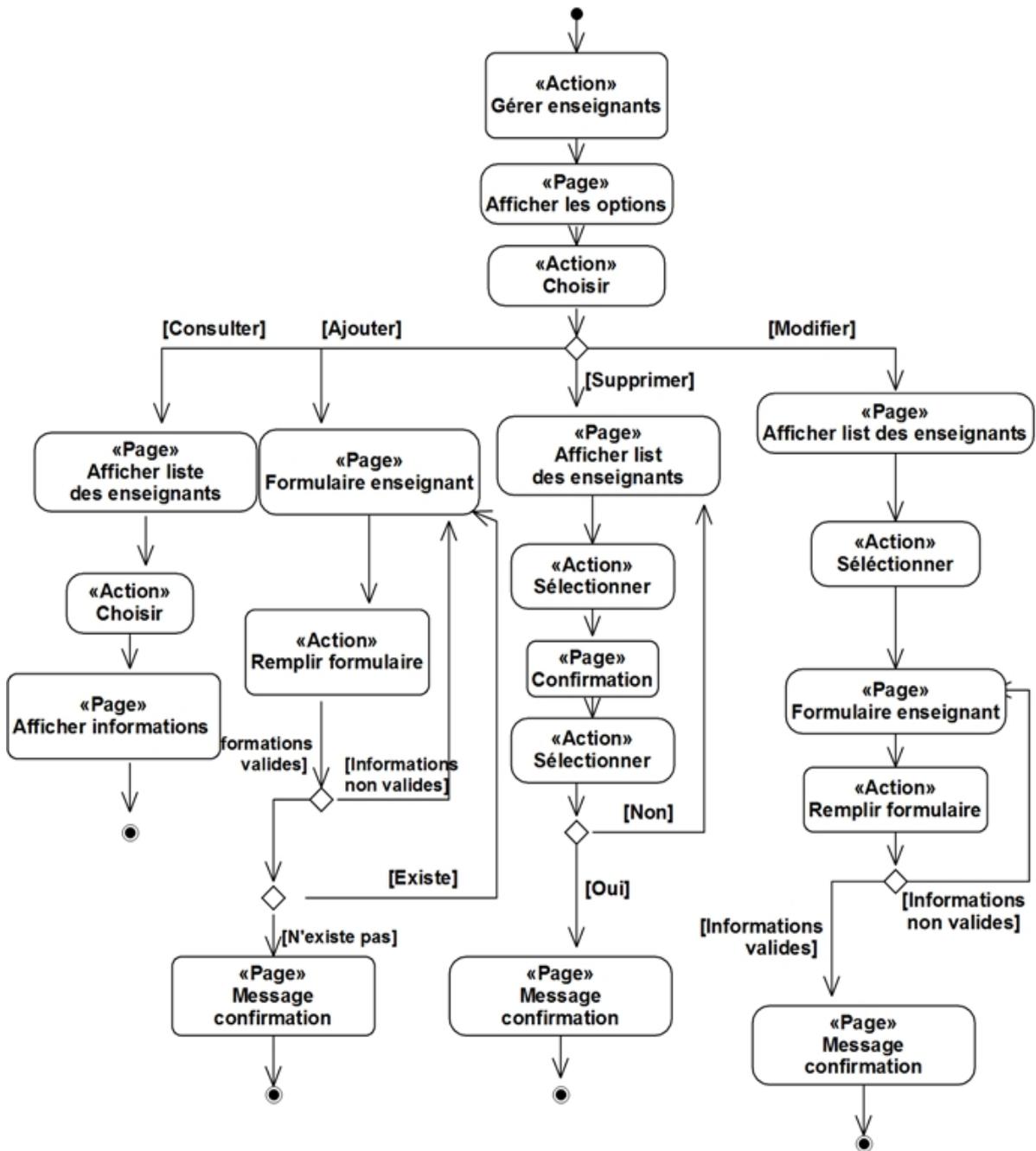


FIGURE 3.7 – Diagramme d’activité «Gérer enseignants ».

3.3.2.6 Cas d'utilisation «Gérer étudiants »

Nom	Gérer étudiants
Acteur	Directeur des études
But	Permet au directeur des études de consulter, d'ajouter, de modifier ou de supprimer un étudiant.
Pré-condition	Le directeur des études doit être authentifié.
Post-condition	/
Scénario nominal	<ol style="list-style-type: none"> 1. Le directeur des études demande de gérer les étudiants. 2. Le système affiche les options suivantes : <ul style="list-style-type: none"> • Consulter étudiants. • Ajouter étudiant. • Modifier étudiant. • Supprimer étudiant.

TABLE 3.7 – Description textuelle du cas d'utilisation « Gérer étudiants ».

3.3.2.7 Cas d'utilisation «Ajouter étudiant »

Nom	Ajouter étudiant
Acteur	Directeur des études
But	Permet au directeur des études d'ajouter un étudiant.
Pré-condition	Le directeur des études doit être authentifié.
Post-condition	Nouvel étudiant ajouté par le système.
Scénario nominal	<ol style="list-style-type: none"> 1. Le directeur des études demande d'ajouter un étudiant. 2. Le système affiche le formulaire correspondant. 3. Le directeur des études remplit le formulaire et valide. 4. Le système vérifie les informations entrées. 5. Le système enregistre le nouvel étudiant dans la base de données. 6. Le système affiche un message de confirmation.

Scénario Alternatif	<p>A1. Le directeur des études n'a pas rempli certains champs ou a saisi des informations non valides.</p> <ol style="list-style-type: none"> 1. Le système affiche un message d'erreur. 2. Le système retourne à l'étape numéro 2 du scénario nominal. <p>A2. L'étudiant existe déjà dans la base de données.</p> <ol style="list-style-type: none"> 1. Le système affiche un message d'erreur. 2. Le système retourne à l'étape numéro 2 du scénario nominal.
----------------------------	---

TABLE 3.8 – Description textuelle du cas d'utilisation «Ajouter étudiant ».

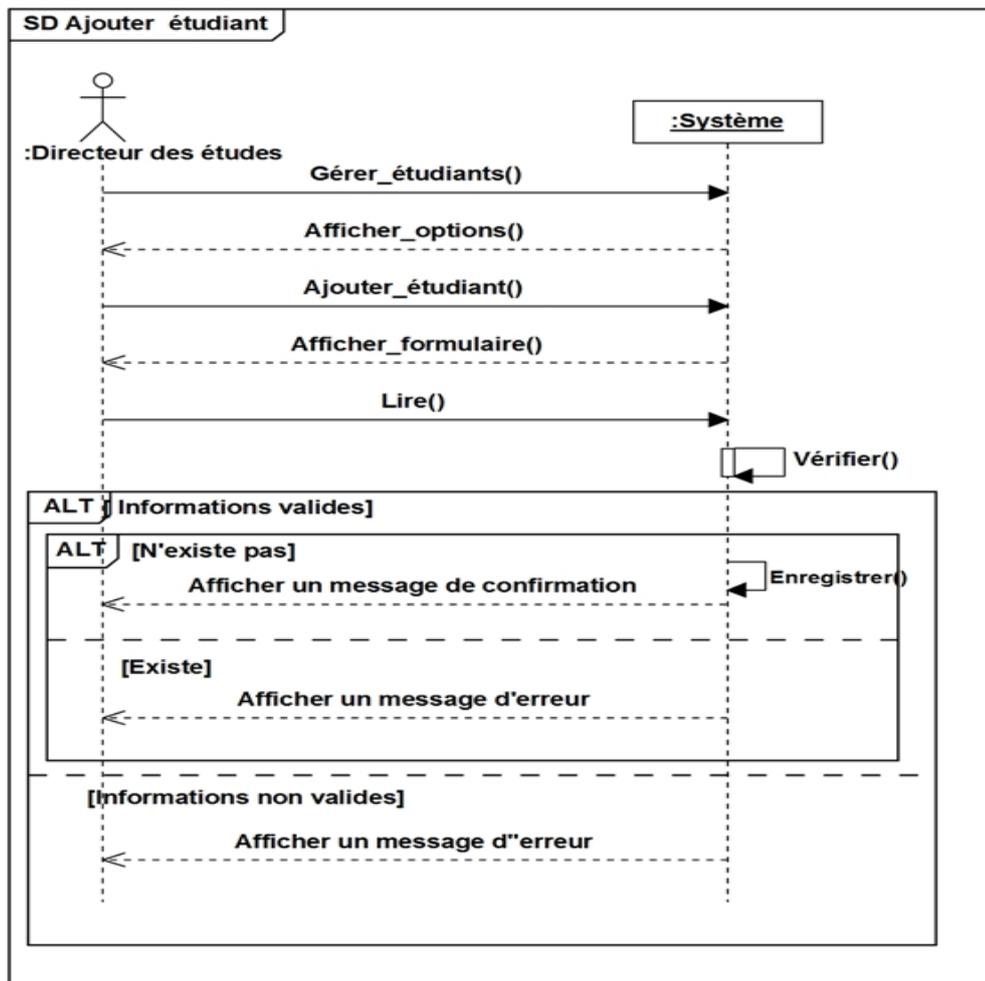


FIGURE 3.8 – Diagramme de séquence «Ajouter étudiant ».

3.3.2.8 Cas d'utilisation «Modifier étudiant»

Nom	Modifier étudiant
Acteur	Directeur des études
But	Permet au directeur des études de modifier un étudiant.
Pré-condition	Le directeur des études doit être authentifié.
Post-condition	Les modifications sont enregistrées par le système.
Scénario nominal	<ol style="list-style-type: none"> 1. Le directeur des études demande de modifier un étudiant. 2. Le système affiche la liste des étudiants. 3. Le directeur des études sélectionne un étudiant. 4. Le système affiche les informations d'un étudiant. 5. Le directeur des études modifie et valide. 6. Le système vérifie les informations entrées. 7. Le système enregistre les modifications dans la base de données. 8. Le système affiche un message de confirmation.
Scénario Alternatif	<p>A1. Le directeur des études n'a pas rempli certains champs ou a saisi des informations non valides.</p> <ol style="list-style-type: none"> 1. Le système affiche un message d'erreur. 2. Le système retourne à l'étape numéro 4 du scénario nominal.

TABLE 3.9 – Description textuelle du cas d'utilisation «Modifier étudiant ».

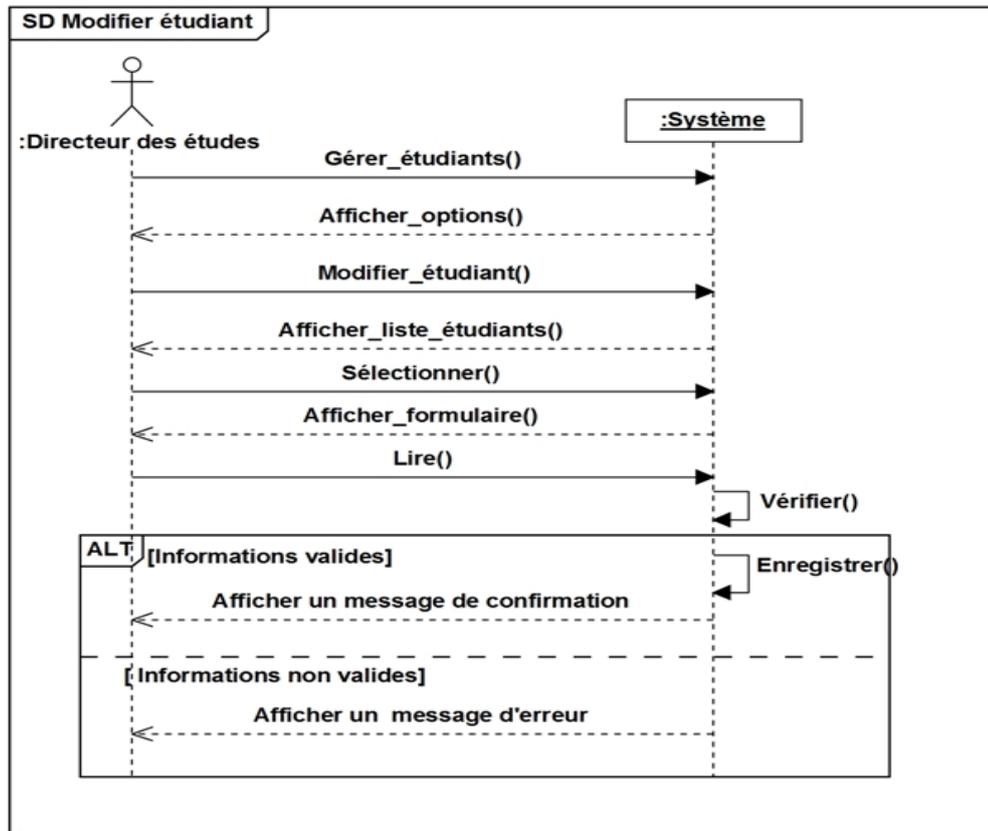


FIGURE 3.9 – Diagramme de séquence «Modifier étudiant».

3.3.2.9 Cas d'utilisation «Supprimer étudiant»

Nom	Supprimer étudiant
Acteur	Directeur des études
But	Permet au directeur des études de supprimer un étudiant.
Pré-condition	Le directeur des études doit être authentifié.

Post-condition	Etudiant supprimé du système.
Scénario nominal	<ol style="list-style-type: none"> 1. Le directeur des études demande de supprimer un étudiant. 2. Le système affiche la liste des étudiants. 3. Le directeur des études sélectionne un étudiant. 4. Le système demande de confirmer la suppression. 5. Le directeur des études confirme la suppression. 6. Le système supprime l'étudiant de la base de données. 7. Le système affiche un message de confirmation.
Scénario Alternatif	<p>A1. Le directeur des études ne confirme pas la suppression.</p> <ol style="list-style-type: none"> 1. Le système retourne à l'étape numéro 2 du scénario nominal.

TABLE 3.10 – Description textuelle du cas d'utilisation «Supprimer étudiant».

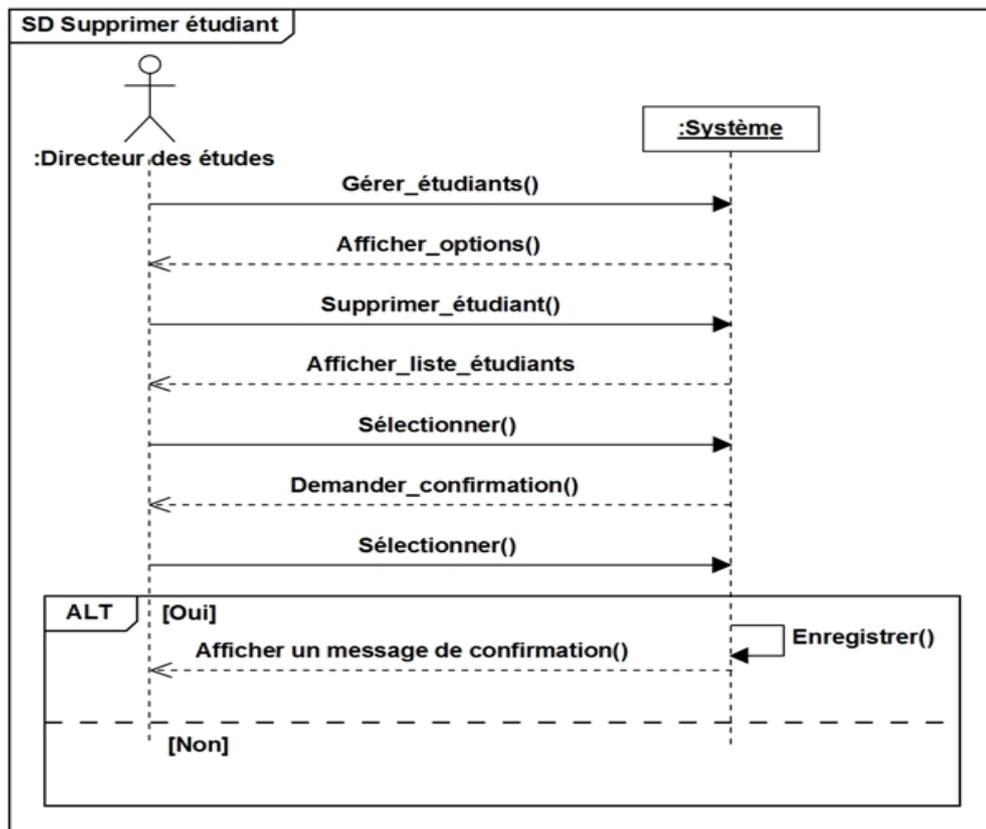


FIGURE 3.10 – Diagramme de séquence «Supprimer étudiant».

3.3.2.10 Cas d'utilisation «Consulter étudiants»

Nom	Consulter étudiants
Acteur	Directeur des études
But	Permet au directeur des études de consulter les informations des étudiants.
Pré-condition	Directeur des études doit être authentifié.
Post-condition	Informations des étudiants affichées.
Scénario nominal	<ol style="list-style-type: none"> 1. Le directeur des études demande de consulter les étudiants. 2. Le système affiche la liste des étudiants. 3. Le directeur des études choisit un étudiant. 4. Le système affiche les informations de l'étudiant.

TABLE 3.11 – Description textuelle du cas d'utilisation «Consulter étudiants».

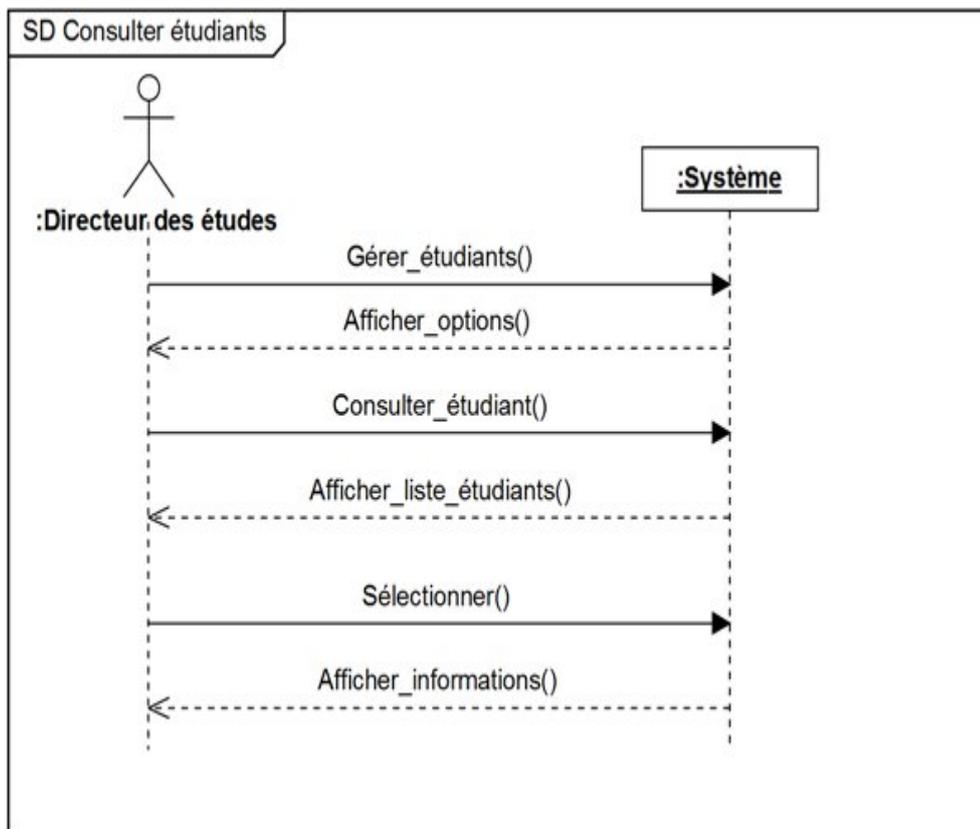


FIGURE 3.11 – Diagramme de séquence «Consulter étudiants».

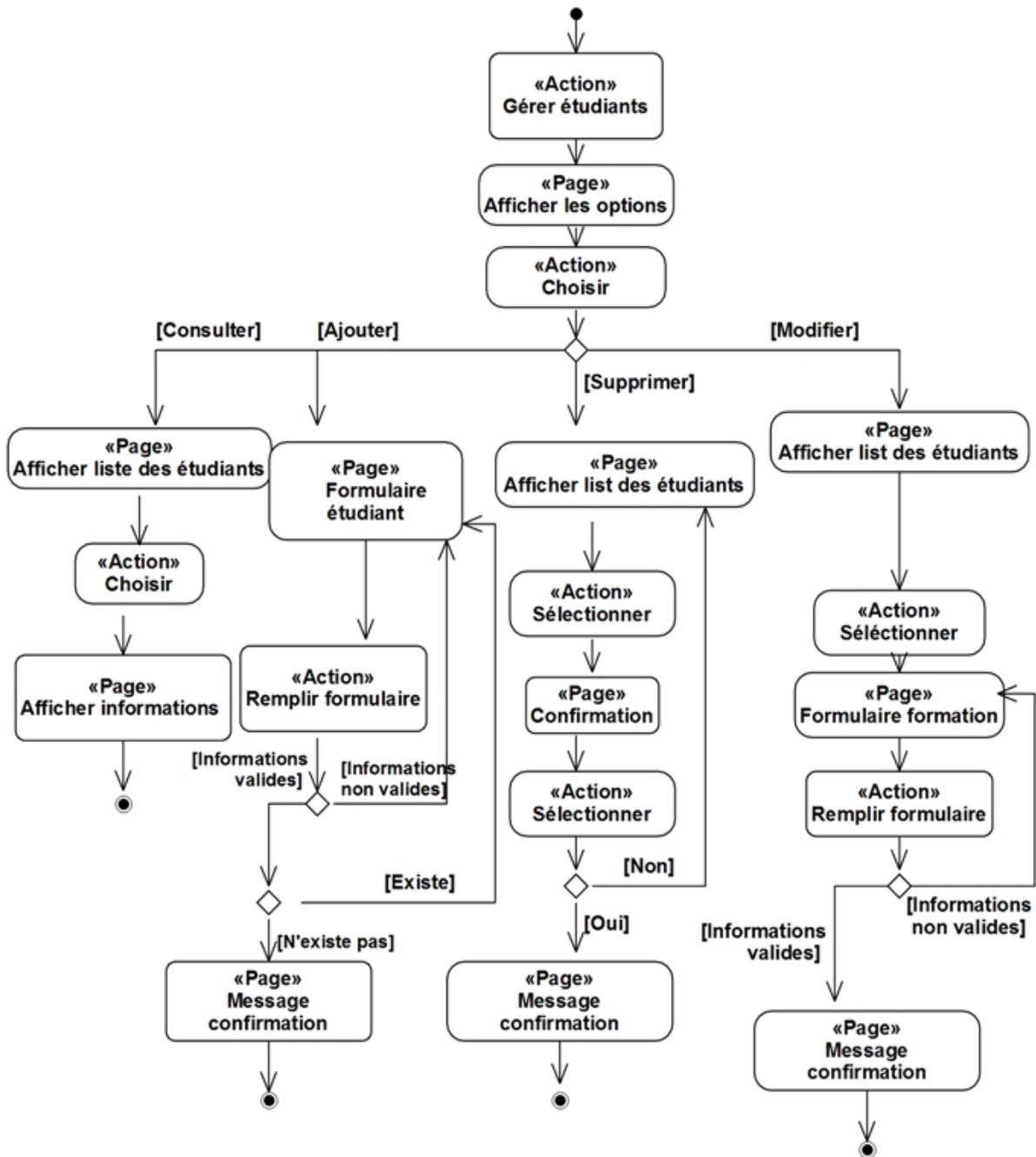


FIGURE 3.12 – Diagramme d’activité «Gérer étudiants ».

3.3.2.11 Cas d'utilisation «Gérer formations »

Nom	Gérer formations
Acteur	Directeur des études
But	Permet au directeur des études de consulter, d'ajouter, de modifier ou de supprimer une formation.
Pré-condition	Le directeur des études doit être authentifié.
Post-condition	/
Scénario nominal	<ol style="list-style-type: none"> 1. Le directeur des études demande de gérer les formations. 2. Le système affiche les options suivantes : <ul style="list-style-type: none"> • Consulter formations. • Ajouter formation. • Modifier formation. • Supprimer formation.

TABLE 3.12 – Description textuelle du cas d'utilisation « Gérer formations ».

3.3.2.12 Cas d'utilisation «Ajouter formation »

Nom	Ajouter formation
Acteur	Directeur des études
But	Permet au directeur des études d'ajouter une formation.
Pré-condition	Le directeur des études doit être authentifié.
Post-condition	Nouvelle formation ajoutée par le système.
Scénario nominal	<ol style="list-style-type: none"> 1. Le directeur des études demande d'ajouter une formation. 2. Le système affiche le formulaire correspondant. 3. Le directeur des études remplit le formulaire et valide. 4. Le système vérifie les informations entrées. 5. Le système enregistre la nouvelle formation dans la base de données. 6. Le système affiche un message de confirmation.

Scénario Alternatif	<p>A1. Le directeur des études n'a pas rempli certains champs ou a saisi des informations non valides.</p> <ol style="list-style-type: none"> 1. Le système affiche un message d'erreur. 2. Le système retourne à l'étape numéro 2 du scénario nominal. <p>A2. La formation existe déjà dans la base de données.</p> <ol style="list-style-type: none"> 1. Le système affiche un message d'erreur. 2. Le système retourne à l'étape numéro 2 du scénario nominal.
----------------------------	---

TABLE 3.13 – Description textuelle du cas d'utilisation «Ajouter formation ».

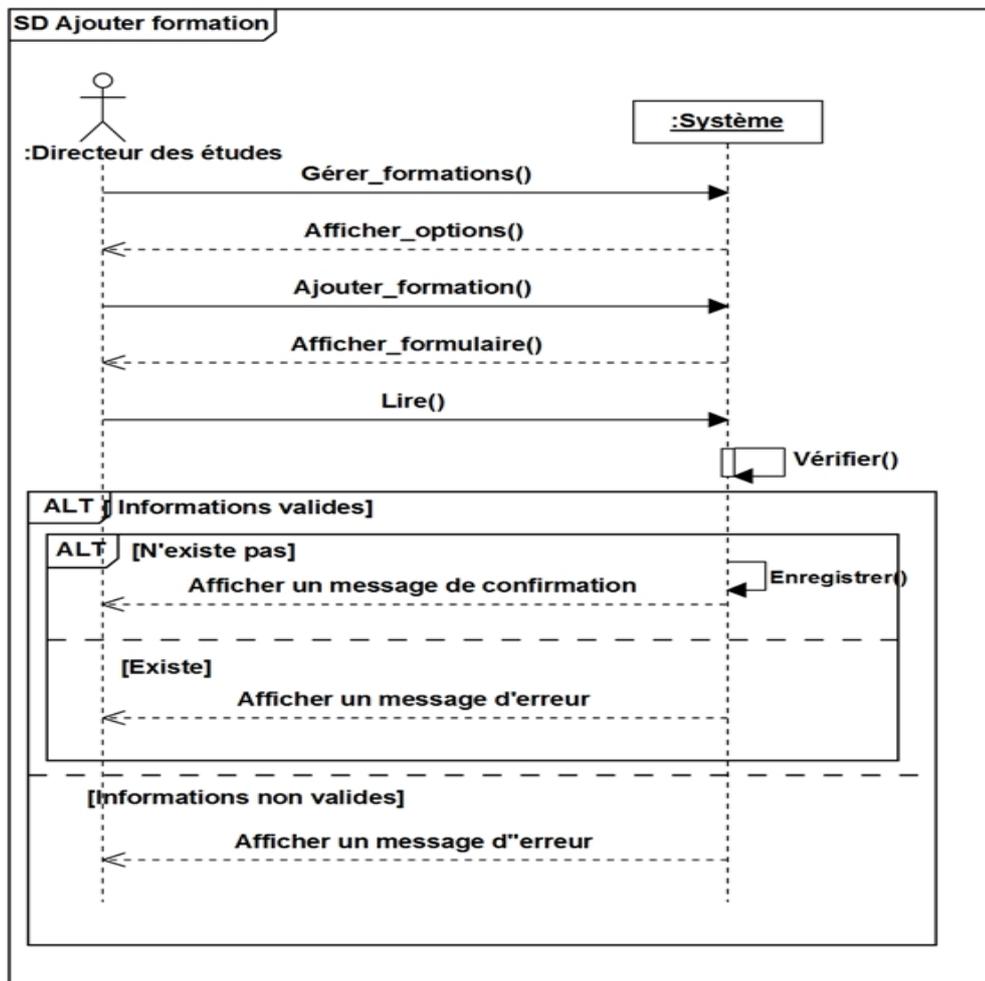


FIGURE 3.13 – Diagramme de séquence «Ajouter formation ».

3.3.2.13 Cas d'utilisation «Modifier formation»

Nom	Modifier étudiant
Acteur	Directeur des études
But	Permet au directeur des études de modifier une formation.
Pré-condition	Le directeur des études doit être authentifié.
Post-condition	Les modifications sont enregistrées par le système.
Scénario nominal	<ol style="list-style-type: none"> 1. Le directeur des études demande de modifier une formation. 2. Le système affiche la liste des formations. 3. Le directeur des études sélectionne une formation. 4. Le système affiche les informations de la formation. 5. Le directeur des études modifie les informations et valide. 6. Le système vérifie les informations entrées. 7. Le système enregistre les modifications dans la base de données. 8. Le système affiche un message de confirmation.
Scénario Alternatif	<p>A1. Le directeur des études n'a pas rempli certains champs ou a saisi des informations non valides.</p> <ol style="list-style-type: none"> 1. Le système affiche un message d'erreur. 2. Le système retourne à l'étape numéro 4 du scénario nominal.

TABLE 3.14 – Description textuelle du cas d'utilisation «Modifier formation ».

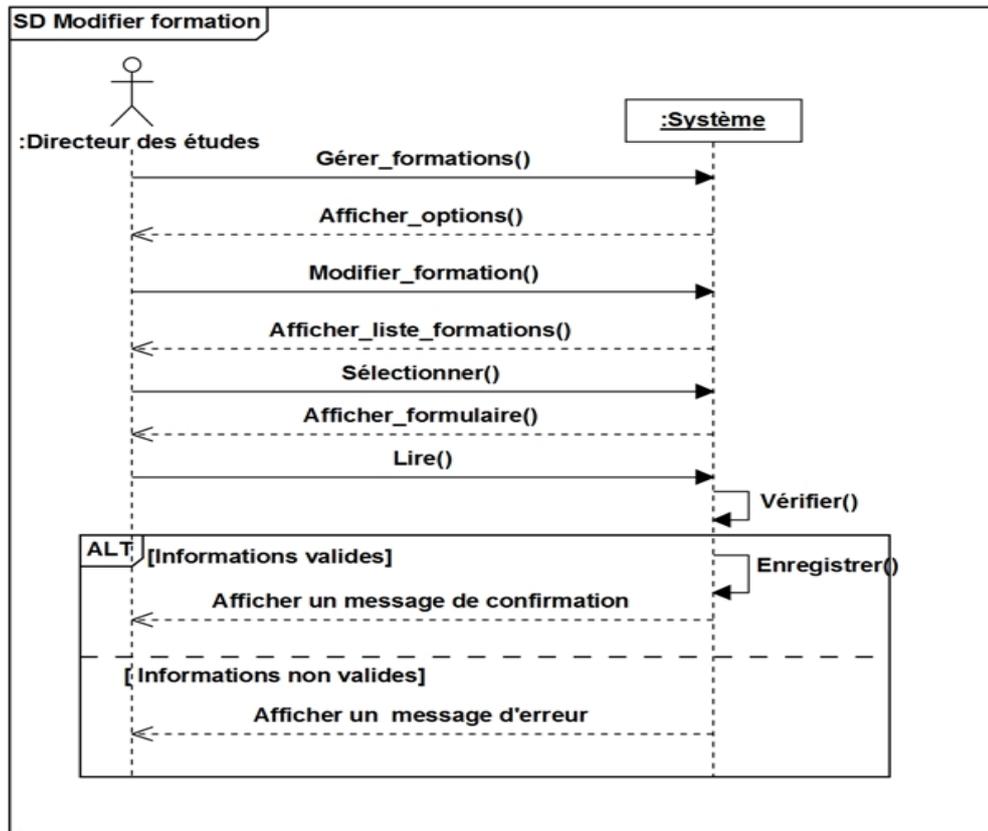


FIGURE 3.14 – Diagramme de séquence «Modifier formation».

3.3.2.14 Cas d'utilisation «Supprimer formation»

Nom	Supprimer formation
Acteur	Directeur des études
But	Permet au directeur des études de supprimer une formation.
Pré-condition	Le directeur des études doit être authentifié.
Post-condition	Formation supprimée du système.
Scénario nominal	<ol style="list-style-type: none"> 1. Le directeur des études demande de supprimer une formation. 2. Le système affiche la liste des formations. 3. Le directeur des études sélectionne une formation. 4. Le système demande de confirmer la suppression. 5. Le directeur des études confirme la suppression.

	<p>6. Le système supprime la formation de la base de données.</p> <p>7. Le système affiche un message de confirmation.</p>
Scénario Alternatif	<p>A1. Le directeur des études ne confirme pas la suppression.</p> <p>1. Le système retourne à l'étape numéro 2 du scénario nominal.</p>

TABLE 3.15 – Description textuelle du cas d'utilisation «Supprimer formation».

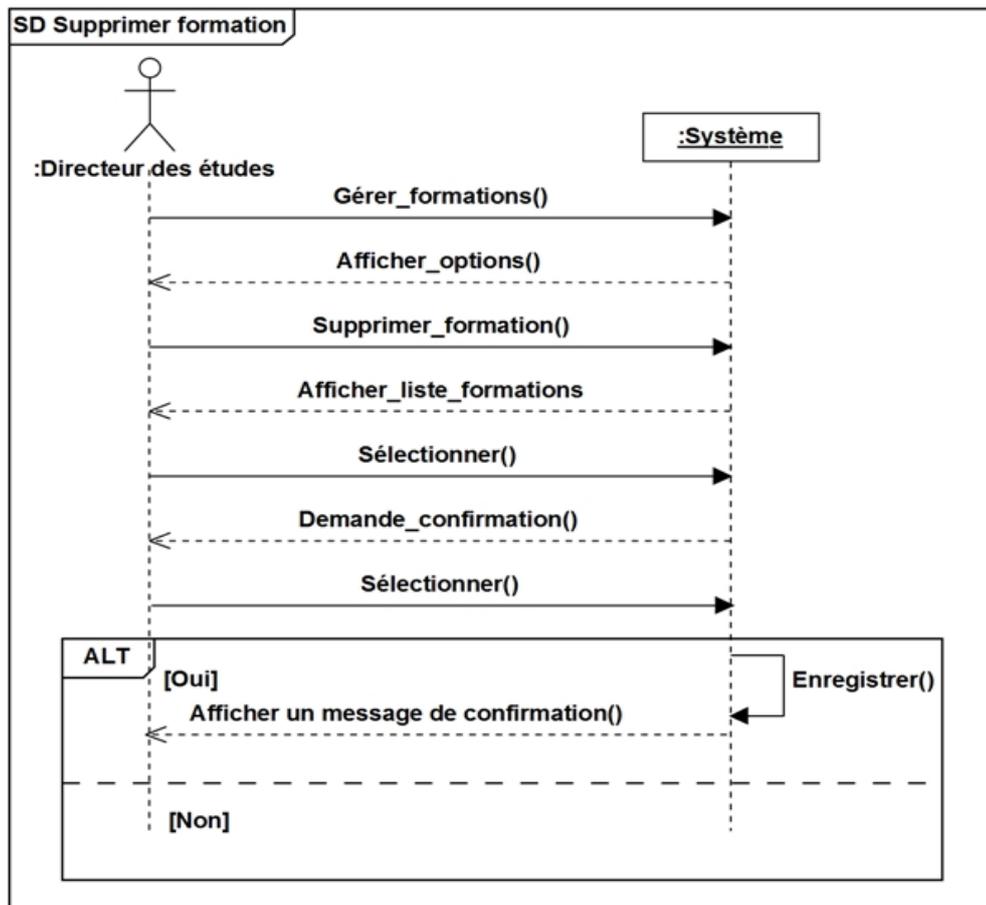


FIGURE 3.15 – Diagramme de séquence «Supprimer formation».

3.3.2.15 Cas d'utilisation «Consulter formations»

Nom	Consulter formations
Acteur	Directeur des études
But	Permet au directeur des études de consulter les formations..
Pré-condition	Directeur des études doit être authentifié.
Post-condition	Informations de formation affichées.
Scénario nominal	<ol style="list-style-type: none"> 1. Le directeur des études demande de consulter les étudiants. 2. Le système affiche la liste des formations. 3. Le directeur des études sélectionne une formation. 4. Le système affiche les informations de la formation.

TABLE 3.16 – Description textuelle du cas d'utilisation «Consulter formations».

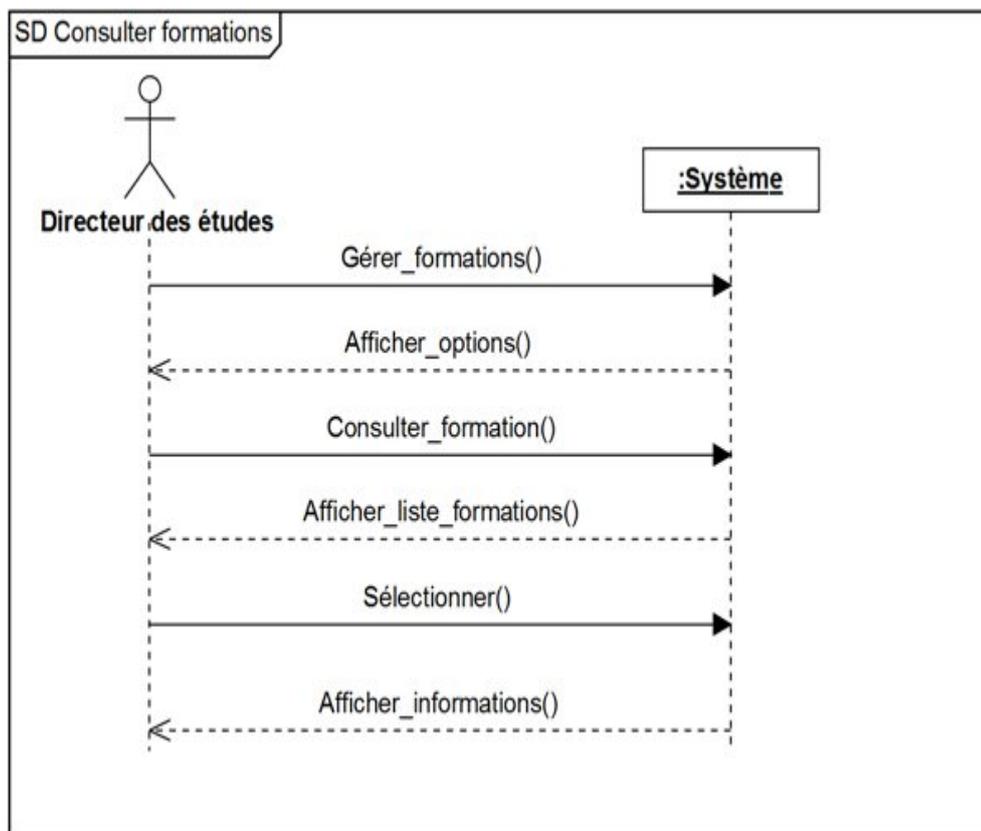


FIGURE 3.16 – Diagramme de séquence «Consulter formations».

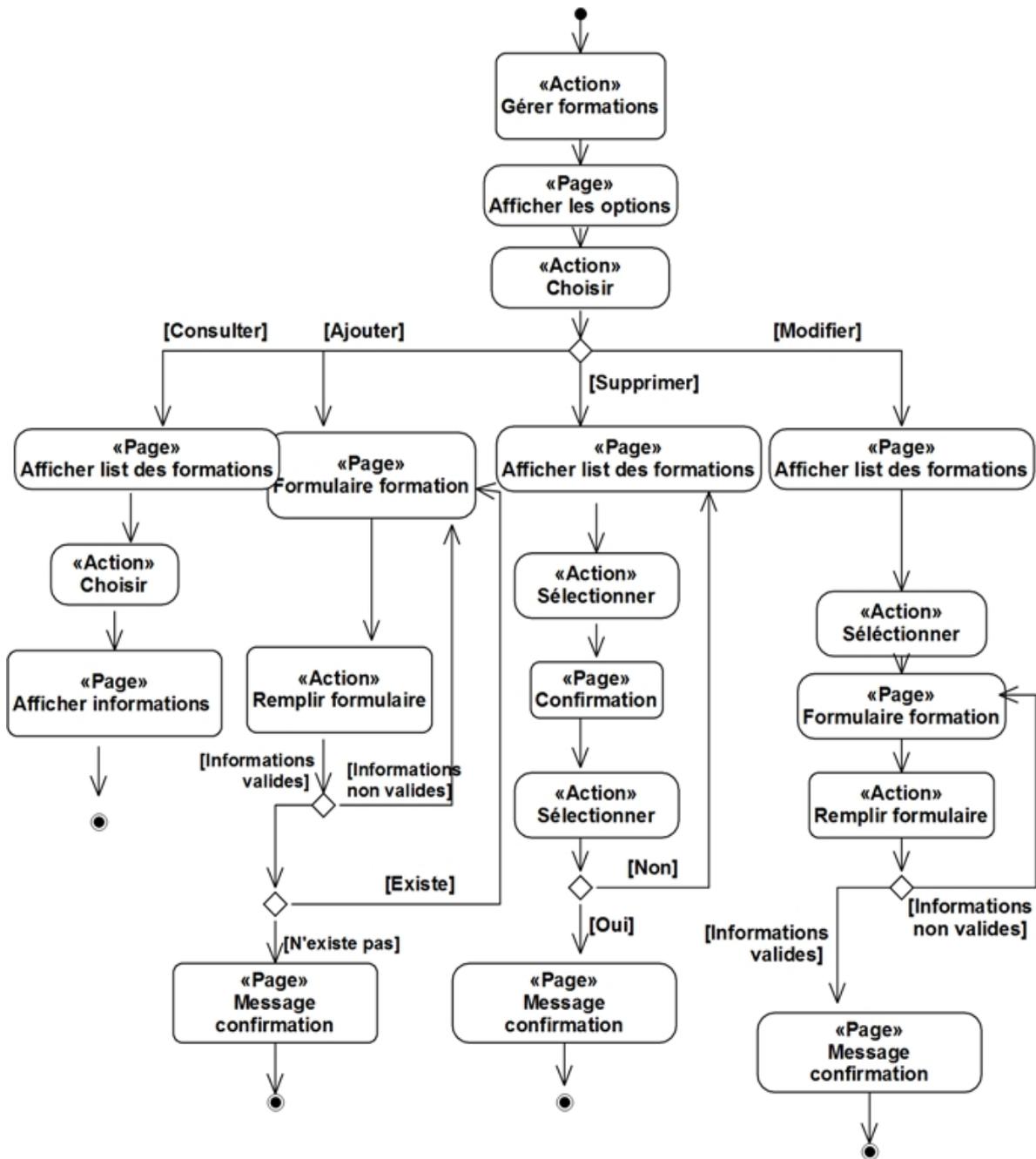


FIGURE 3.17 – Diagramme d’activité «Gérer formations».

3.3.2.16 Cas d'utilisation « Publier annonce »

Nom	Publier Annonce
Acteur	Directeur des études /Enseignant
But	Permet au directeur des études ou à l'enseignant de publier une annonce.
Pré-condition	Le directeur des études/enseignant doit être authentifié.
Post-condition	Nouvelle annonce ajoutée par le système.
Scénario nominal	<ol style="list-style-type: none"> 1. Le directeur des études/enseignant demande de publier une annonce. 2. Le système affiche un formulaire de publication. 3. Le directeur des études /enseignant remplit le formulaire et valide. 4. Le système vérifie les informations entrées. 5. Le système enregistre les informations dans la base de données. 6. Le système affiche un message de confirmation.
Scénario Alternatif	<p>A1. Le directeur des études/enseignant n'a pas rempli certains champs ou a saisi des informations non valides.</p> <ol style="list-style-type: none"> 1. Le système affiche un message d'erreur. 2. Le système retourne à l'étape numéro 2 du scénario nominal.

TABLE 3.17 – Description textuelle du cas d'utilisation «Publier annonce».

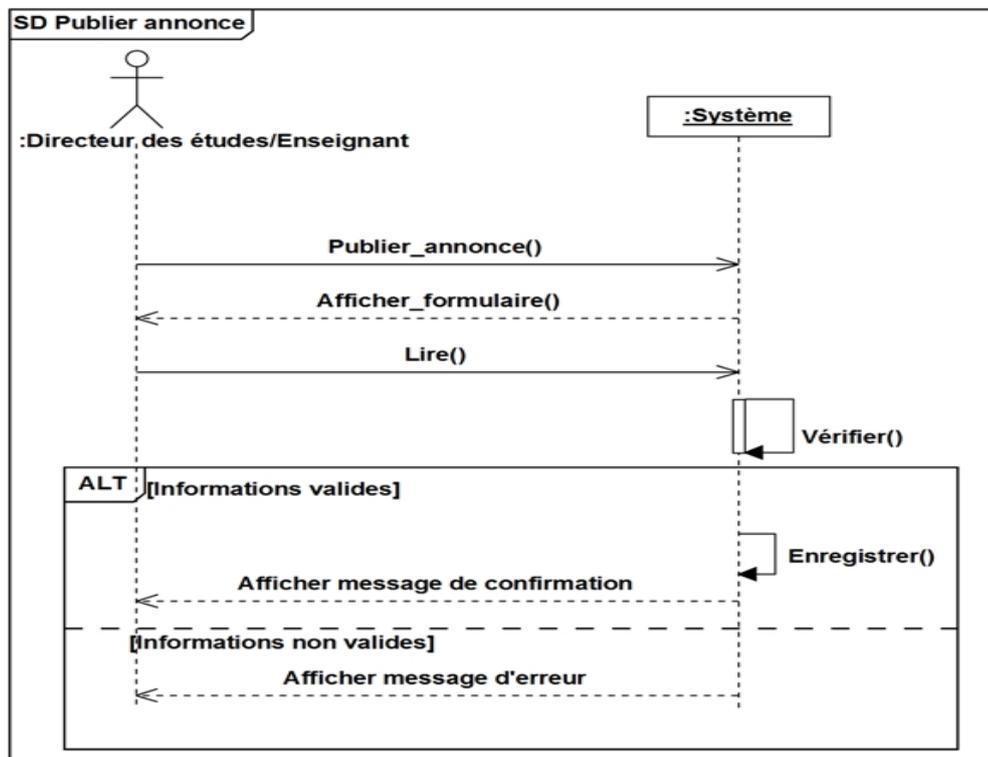


FIGURE 3.18 – Diagramme de séquence «Publier annonce».

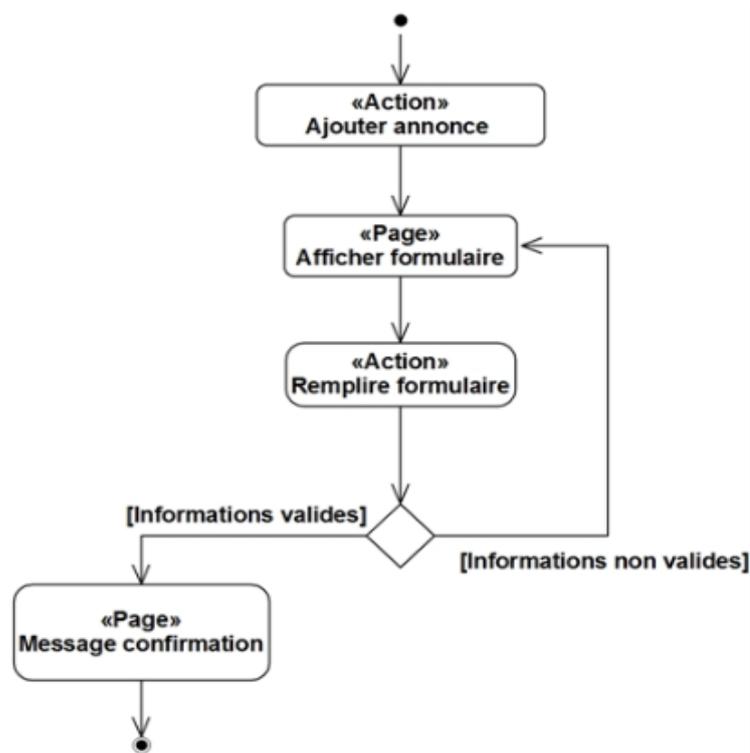


FIGURE 3.19 – Diagramme d’activité «Publier annonce».

3.3.2.17 Cas d'utilisation « Accéder aux ressources »

Nom	Accéder aux ressources
Acteur	Etudiant
But	Permet à l'étudiant d'accéder aux ressources (cours et multi-médias).
Pré-condition	L'étudiant doit être authentifié.
Post-condition	Contenu d'une ressource affiché.
Scénario nominal	<ol style="list-style-type: none"> 1. L'étudiant demande d'accéder aux ressources. 2. Le système affiche la liste des ressources de la formation (les cours et les multimédias). 3. L'étudiant sélectionne une ressource. 4. Le système affiche le contenu de la ressource. 5. Le système affiche les opérations « Télécharger » et « Evaluer le pourcentage de compréhension ».
Scénario Alternatif	<p>A1. L'étudiant choisit l'opération « Télécharger ».</p> <ol style="list-style-type: none"> 1. Le système fait référence au cas d'utilisation « Télécharger ». <p>A2. L'étudiant choisit l'opération « Evaluer le pourcentage de compréhension ».</p> <ol style="list-style-type: none"> 1. Le système fait référence au cas d'utilisation « Evaluer le pourcentage de compréhension ».

TABLE 3.18 – Description textuelle du cas d'utilisation «Accéder aux ressources».

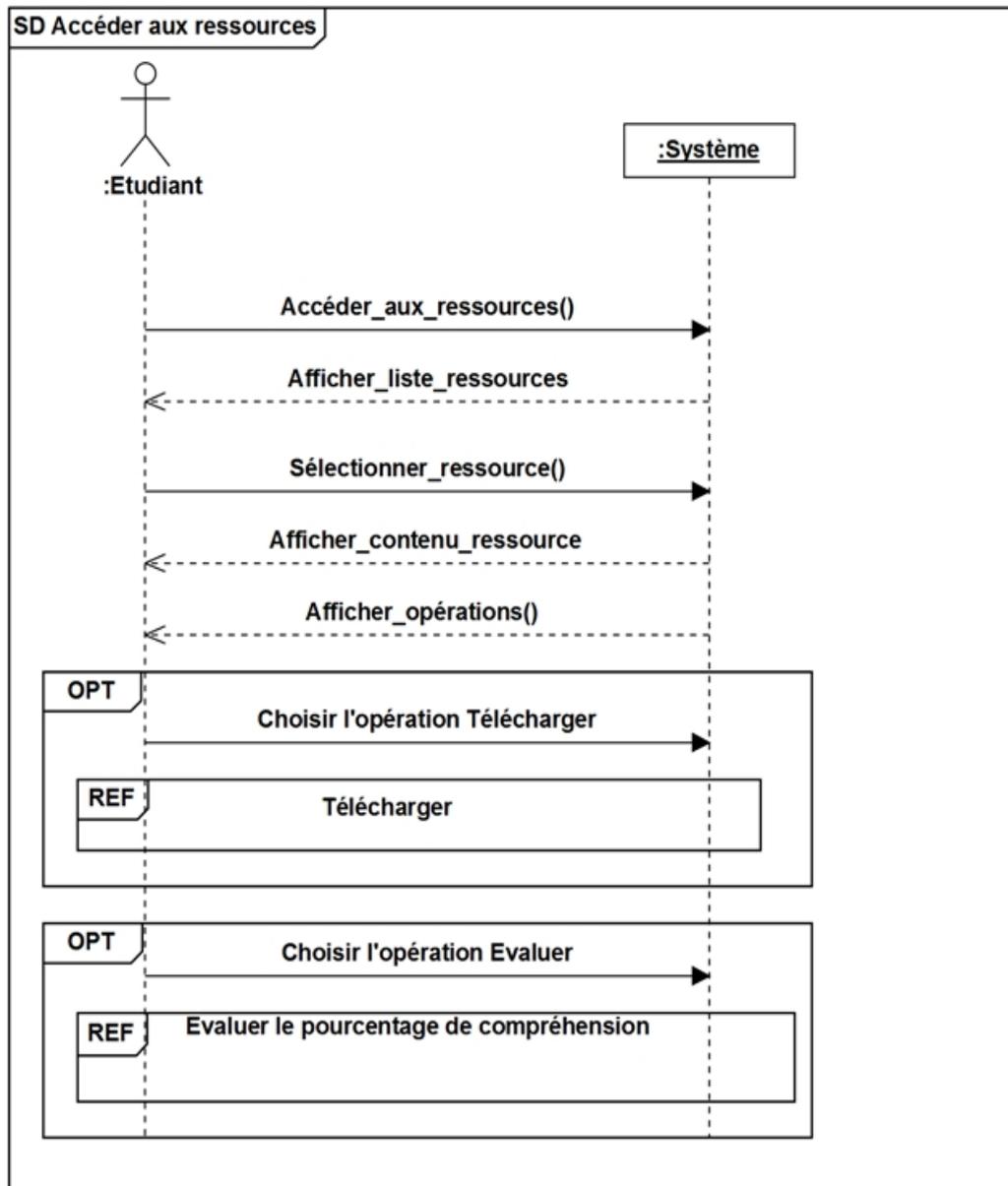


FIGURE 3.20 – Diagramme de séquence «Accéder aux ressources».

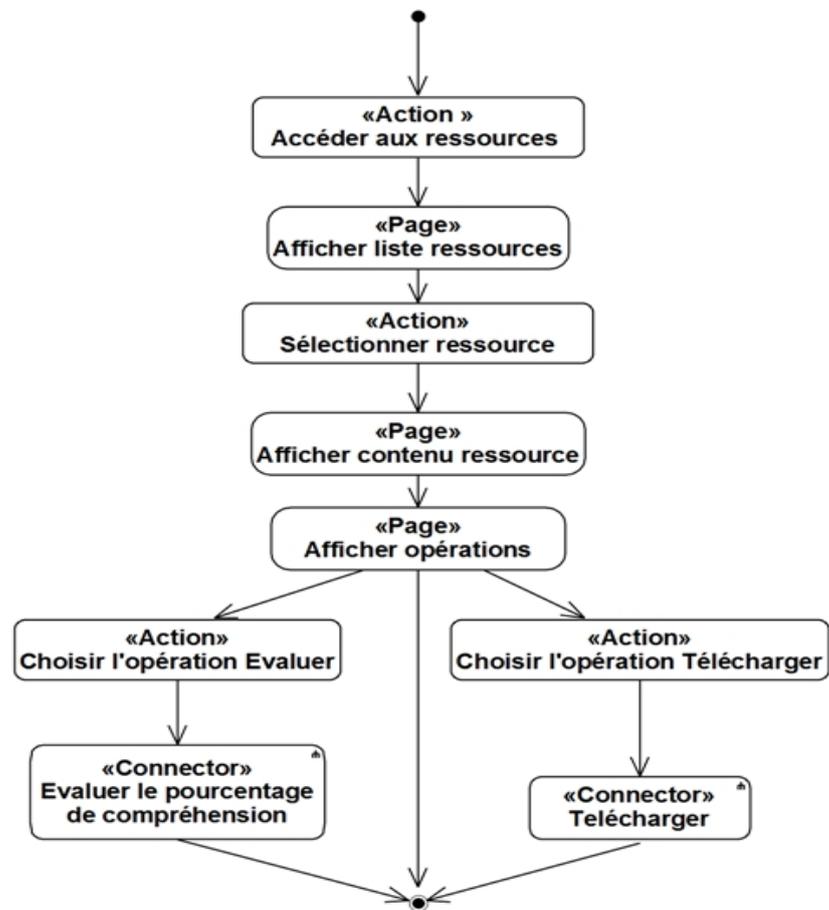


FIGURE 3.21 – Diagramme d’activité «Accéder aux ressources».

3.3.2.18 Cas d’utilisation « Evaluer le pourcentage de compréhension »

Nom	Evaluer le pourcentage de compréhension
Acteur	Etudiant
But	Permet à l’étudiant d’évaluer le pourcentage de compréhension d’une ressource (cours ou multimédia).
Pré-condition	L’étudiant doit être authentifié.
Post-condition	L’évaluation est enregistrée dans la base de données.
Scénario Nominal	<ol style="list-style-type: none"> 1. L’étudiant demande d’évaluer une ressource. 2. Le système affiche le champ correspondant. 3. L’étudiant remplit le champ et valide.

	4. Le système affiche un message de confirmation.
Scénario Alternatif	A1. L'étudiant n'a pas rempli le champ ou a saisi des informations non valides. 1. Le système affiche un message d'erreur. 2. Le système retourne à l'étape numéro 2 du scénario nominal.

TABLE 3.19 – Description textuelle du cas d'utilisation «Evaluer le pourcentage de compréhension».

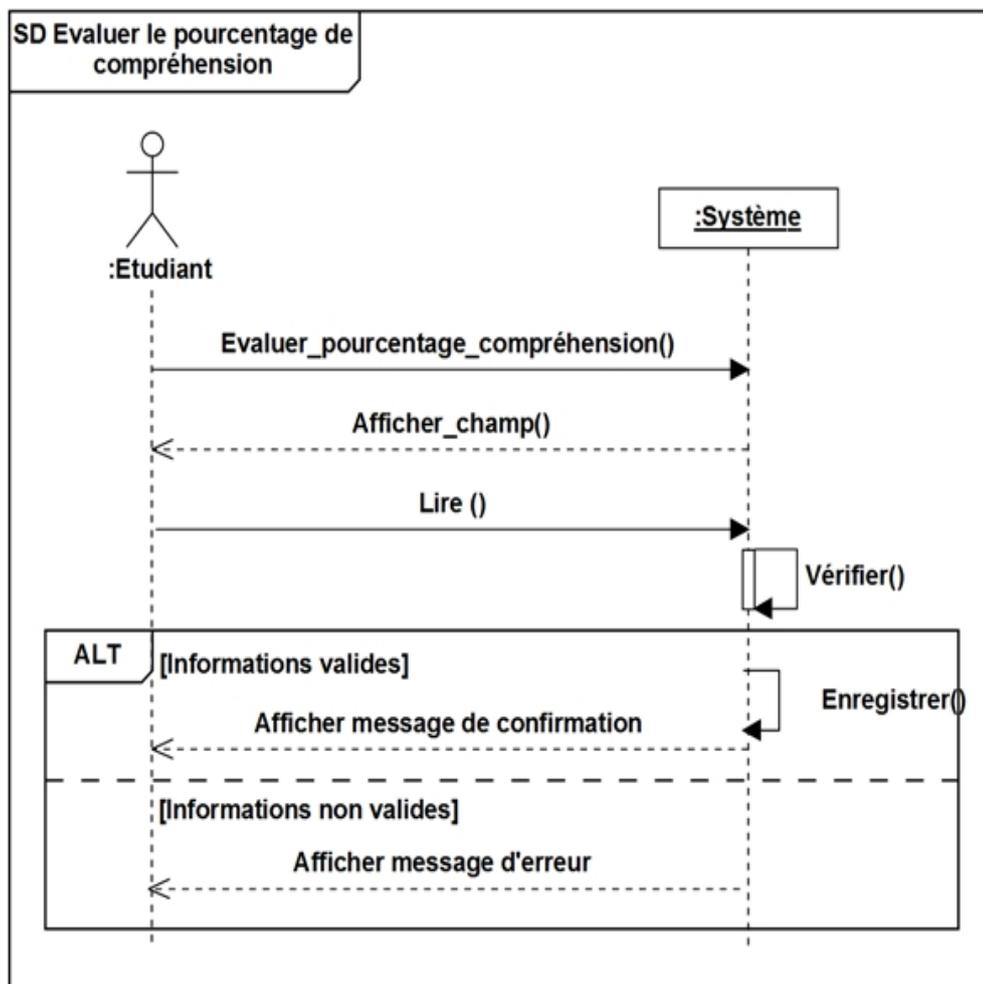


FIGURE 3.22 – Diagramme de séquence «Evaluer le pourcentage de compréhension».

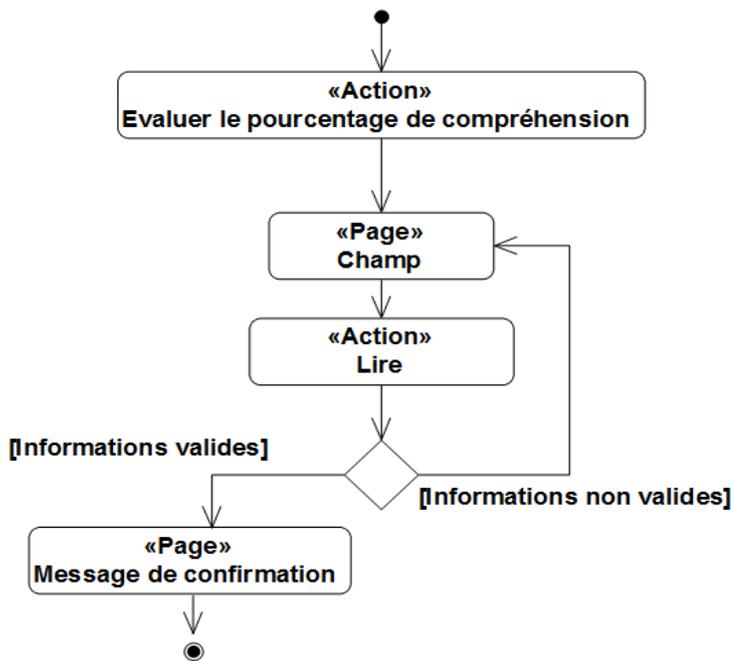


FIGURE 3.23 – Diagramme d’activité «Evaluer le pourcentage de compréhension».

3.3.2.19 Cas d’utilisation « Télécharger »

Nom	Télécharger
Acteur	Etudiant
But	Permet à l’étudiant de télécharger une ressource.
Pré-condition	L’étudiant doit être authentifié.
Post-condition	La ressource est téléchargée.
Scénario Nominal	<ol style="list-style-type: none"> 1. L’étudiant demande de télécharger une ressource. 2. Le système demande de sélectionner un dossier de sauvegarde. 3. L’étudiant sélectionne un dossier et valide. 4. Le système enregistre la ressource dans le dossier sélectionné. 5. Le système affiche un message de confirmation.

TABLE 3.20 – Description textuelle du cas d’utilisation «Télécharger».

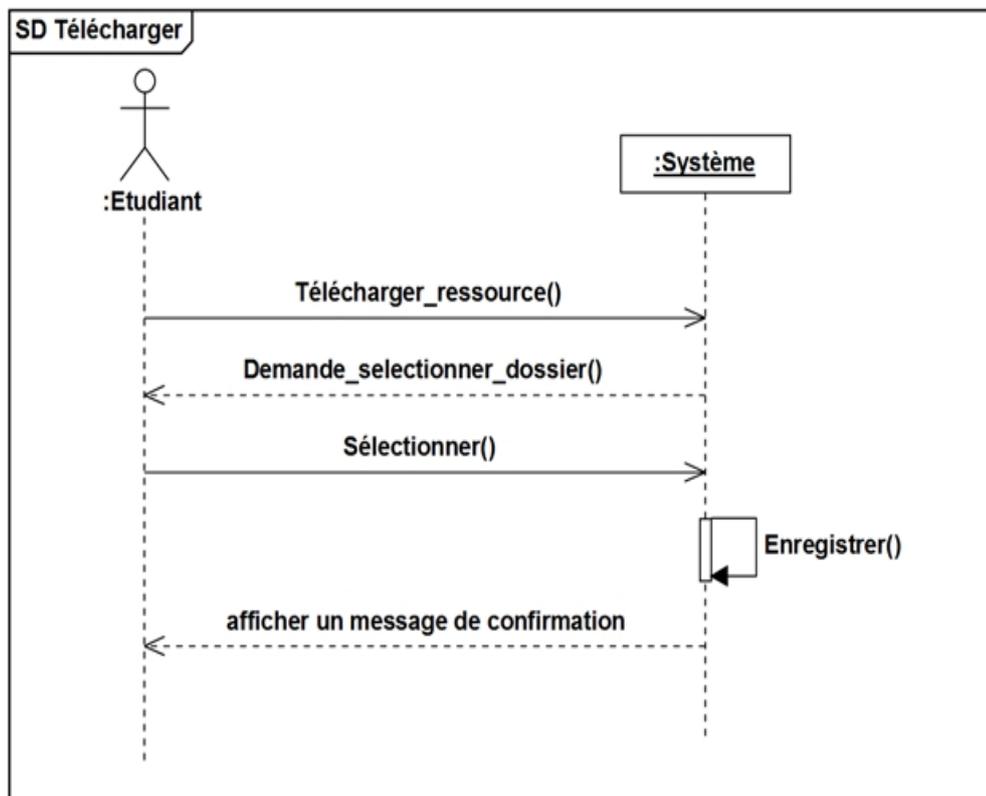


FIGURE 3.24 – Diagramme de séquence «Télécharge».

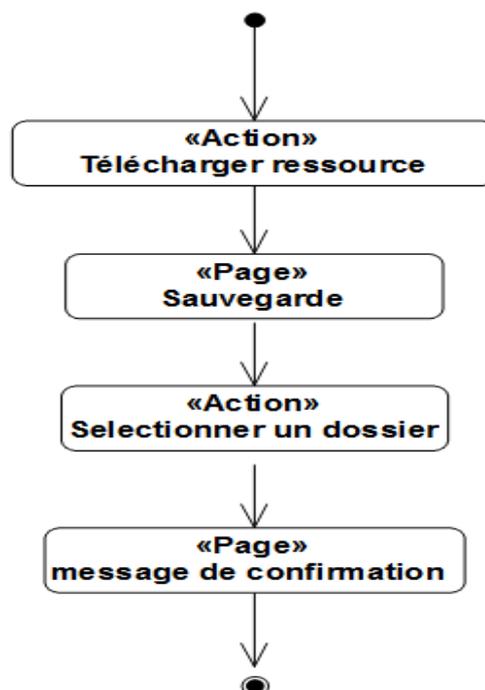


FIGURE 3.25 – Diagramme d’activité «Télécharge».

3.3.2.20 Cas d'utilisation « Consulter annonces »

Nom	Consulter annonces
Acteur	Etudiant
But	Permet à l'étudiant de consulter les annonces de l'administration et des enseignants.
Pré-condition	L'étudiant doit être authentifié.
Post-condition	Contenu d'une annonce affiché.
Scénario nominal	<ol style="list-style-type: none"> 1. L'étudiant demande de consulter les annonces. 2. Le système affiche la liste des annonces. 3. L'étudiant sélectionne une annonce. 4. Le système affiche le contenu de l'annonce.

TABLE 3.21 – Description textuelle du cas d'utilisation «Consulter annonces».

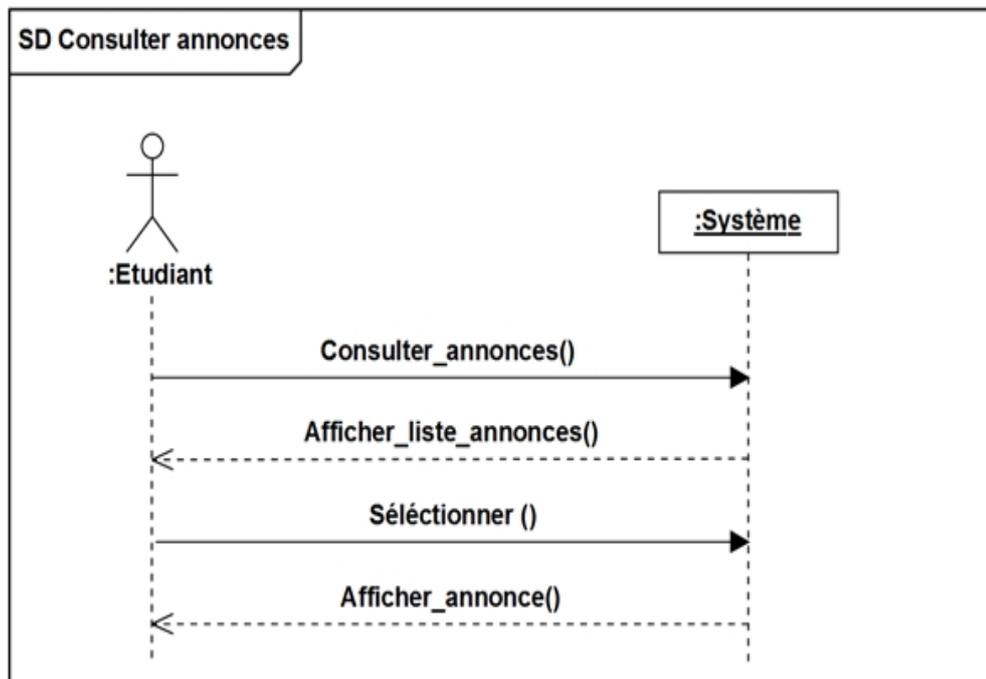


FIGURE 3.26 – Diagramme de séquence «Consulter annonces».

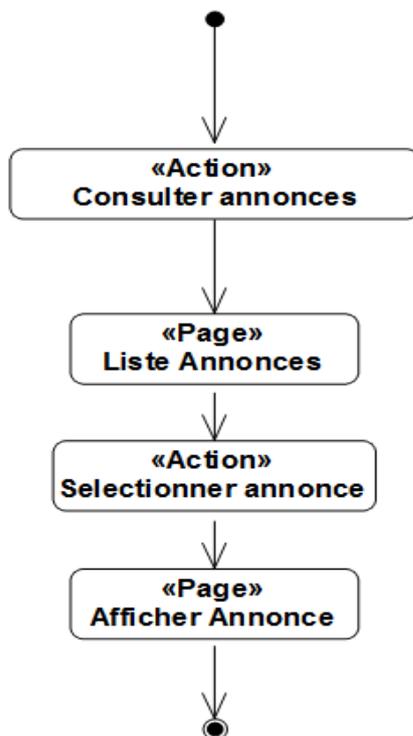


FIGURE 3.27 – Diagramme d’activité «Consulter annonces».

3.3.2.21 Cas d’utilisation « Consulter notes »

Nom	Consulter notes
Acteur	Etudiant
But	Permet à l’étudiant de consulter les notes affichées par l’enseignant.
Pré-condition	L’étudiant doit être authentifié.
Post-condition	Liste de notes affichée.
Scénario Nominal	<ol style="list-style-type: none"> 1. L’étudiant demande de consulter les notes. 2. Le système affiche la liste des affichages. 3. L’étudiant sélectionne un affichage. 4. Le système affiche la liste de notes.

TABLE 3.22 – Description textuelle du cas d’utilisation «Consulter notes».

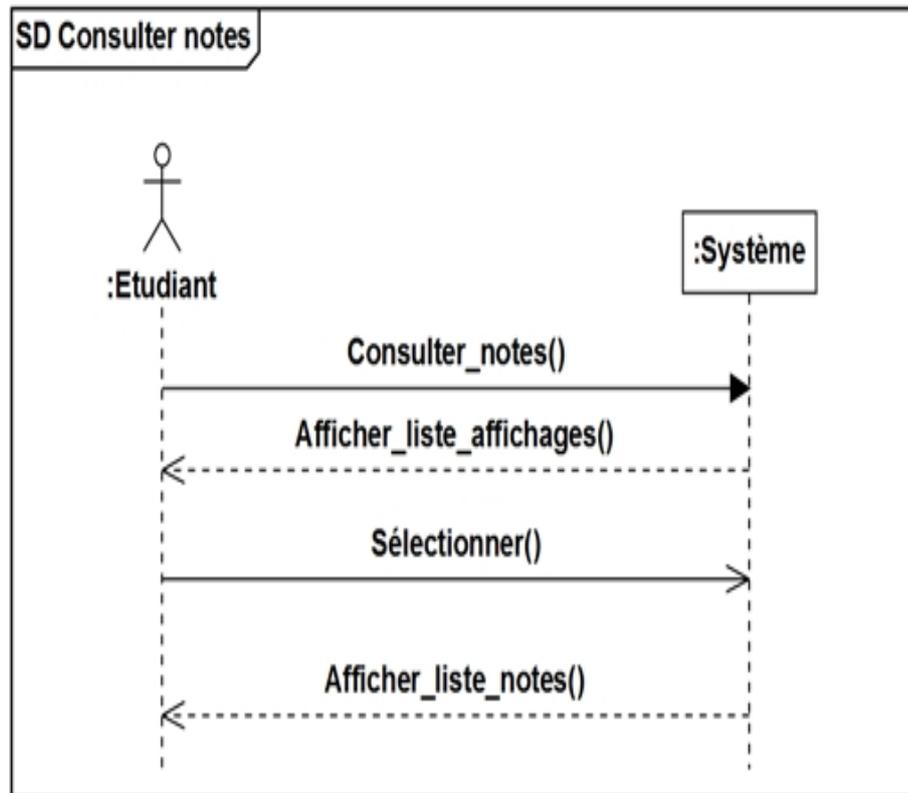


FIGURE 3.28 – Diagramme de séquence «Consulter notes».

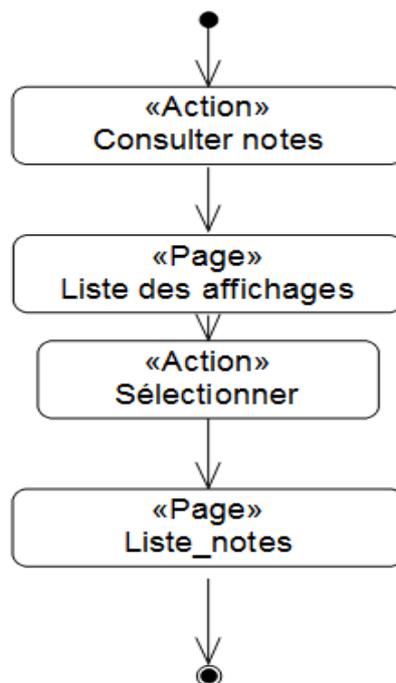


FIGURE 3.29 – Diagramme d’activité «Consulter notes».

3.3.2.22 Cas d'utilisation « Consulter évènements »

Nom	Consulter évènements
Acteur	Etudiant
But	Permet à l'étudiant de consulter les événements lancés par l'enseignant.
Pré-condition	L'étudiant doit être authentifié.
Post-condition	Contenu d'un événement affiché.
Scénario nominal	<ol style="list-style-type: none"> 1. L'étudiant demande de consulter les événements. 2. Le système affiche la liste des événements. 3. L'étudiant sélectionne un événement. 4. Le système affiche le contenu de l'événement. 5. Le système affiche les opérations « Consulter Résultat » et « Participer ».
Scénario Alternatif	<p>A1. L'étudiant choisit l'opération « Consulter Résultat ».</p> <ol style="list-style-type: none"> 1. Le système fait référence au cas d'utilisation « Consulter Résultat ». <p>A2. L'étudiant choisit l'opération « Participer ».</p> <ol style="list-style-type: none"> 1. Le système fait référence au cas d'utilisation « Participer ».

TABLE 3.23 – Description textuelle du cas d'utilisation « Consulter évènements ».

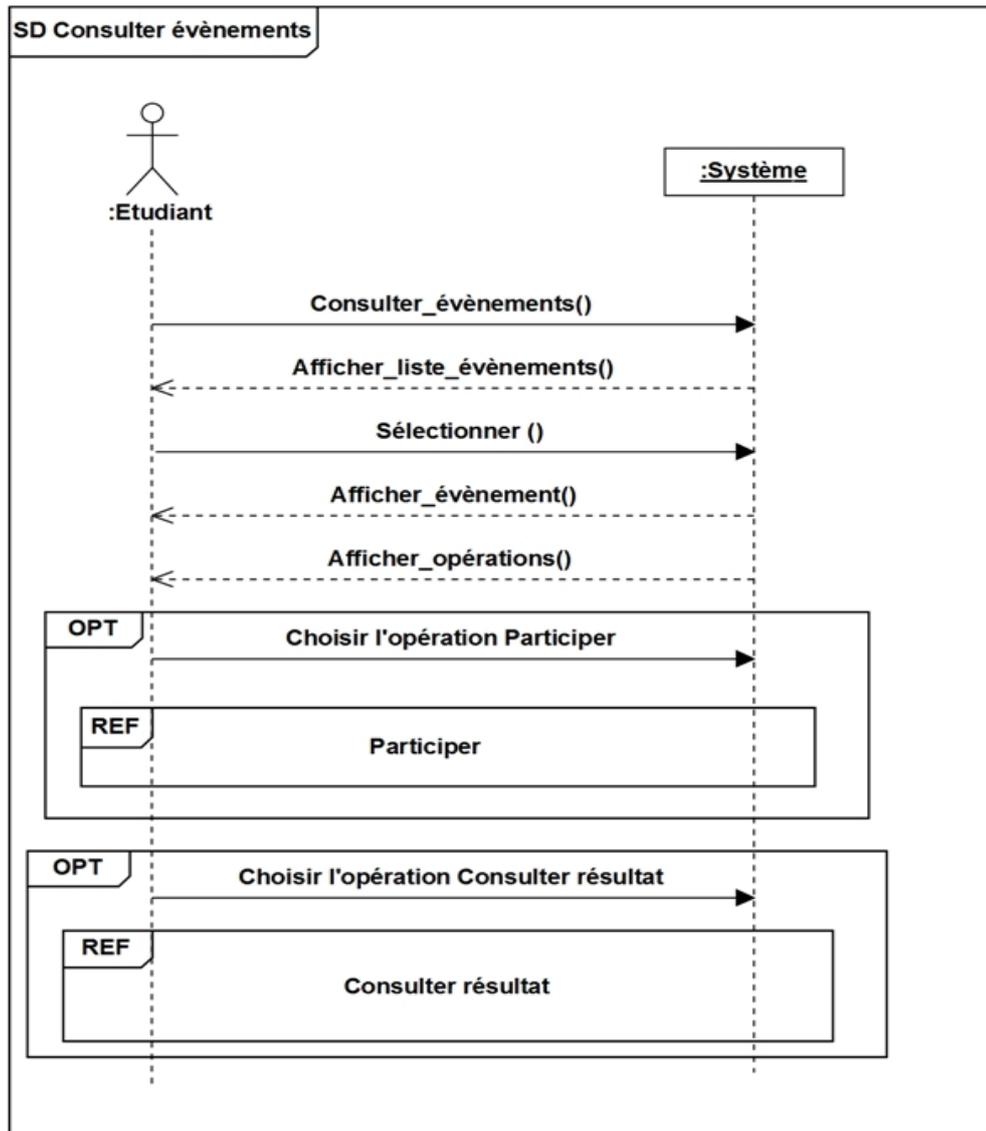


FIGURE 3.30 – Diagramme de séquence «Consulter évènements».

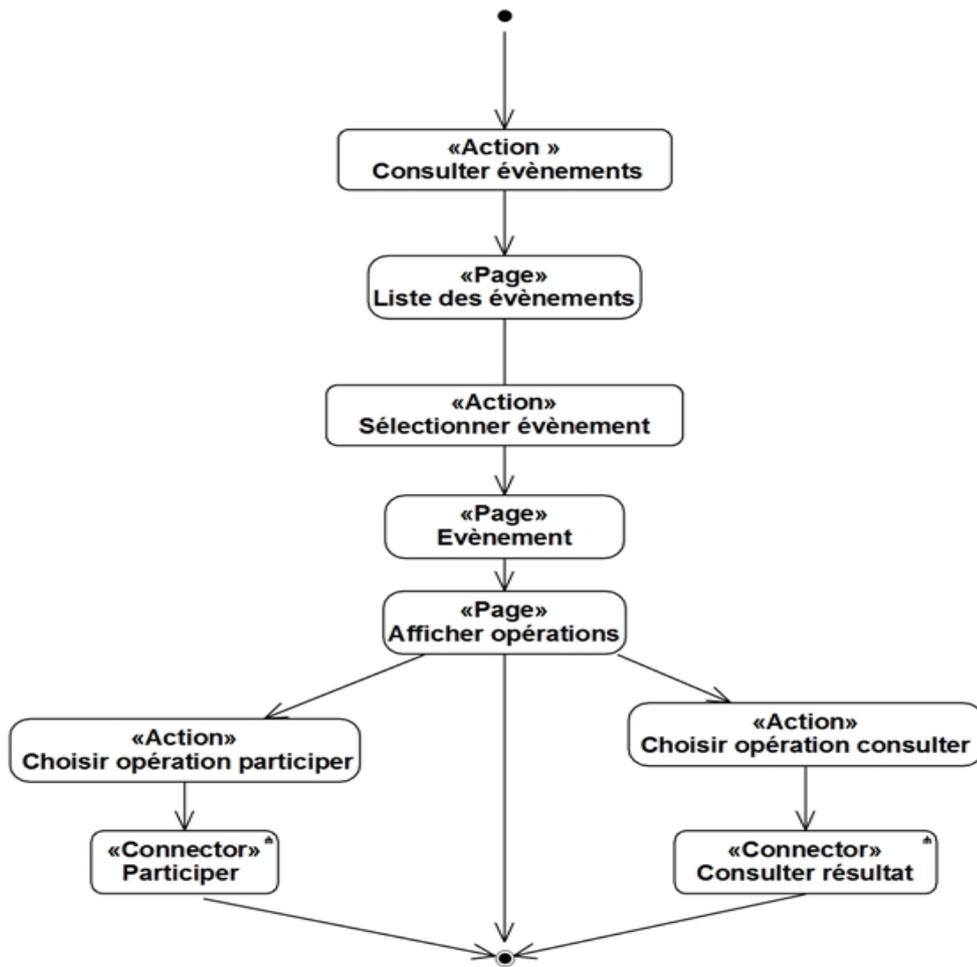


FIGURE 3.31 – Diagramme d’activité « Consulter évènements ».

3.3.2.23 Cas d’utilisation « Participer »

Nom	Participer
Acteur	Etudiant
But	Permet à l’étudiant de participer à un évènement.
Pré-condition	L’étudiant doit être authentifié.
Post-condition	Nouvelle participation ajoutée par le système.
Scénario Nominal	<ol style="list-style-type: none"> 1. L’étudiant demande de participer à un évènement. 2. Le système affiche le formulaire correspondant. 3. L’étudiant remplit le formulaire et valide.

	<p>4. Le système vérifie les informations entrées.</p> <p>5. Le système enregistre la nouvelle participation dans la base de données.</p> <p>6. Le système affiche un message de confirmation.</p>
Scénario Alternatif	<p>A1. L'étudiant n'a pas rempli certains champs ou a saisi des informations non valides.</p> <p>1. Le système affiche un message d'erreur.</p> <p>2. Le système retourne à l'étape numéro 2 du scénario nominal.</p>

TABLE 3.24 – Description textuelle du cas d'utilisation «Participer».

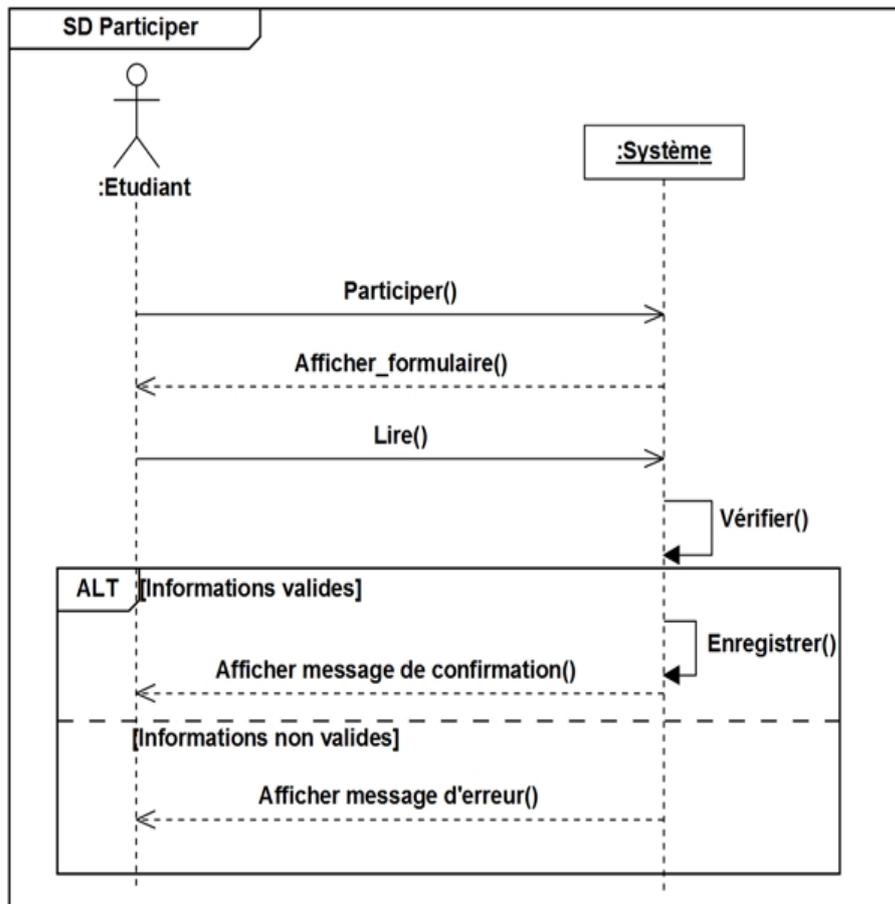


FIGURE 3.32 – Diagramme de séquence «Participer».

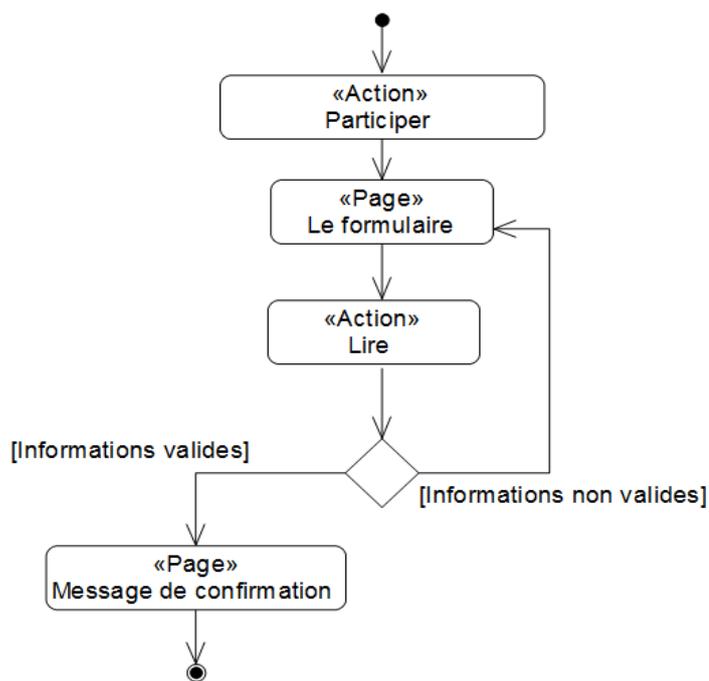


FIGURE 3.33 – Diagramme d’activité «Participer».

3.3.2.24 Cas d’utilisation « Consulter Résultat »

Nom	Consulter Résultat
Acteur	Etudiant
But	Permet à l’étudiant de consulter le résultat d’un évènement.
Pré-condition	L’étudiant doit être authentifié.
Post-condition	Résultat affiché.
Scénario Nominal	1. L’étudiant demande de consulter le résultat d’un évènement. 2. Le système affiche le résultat.

TABLE 3.25 – Description textuelle du cas d’utilisation «Consulter Résultat».

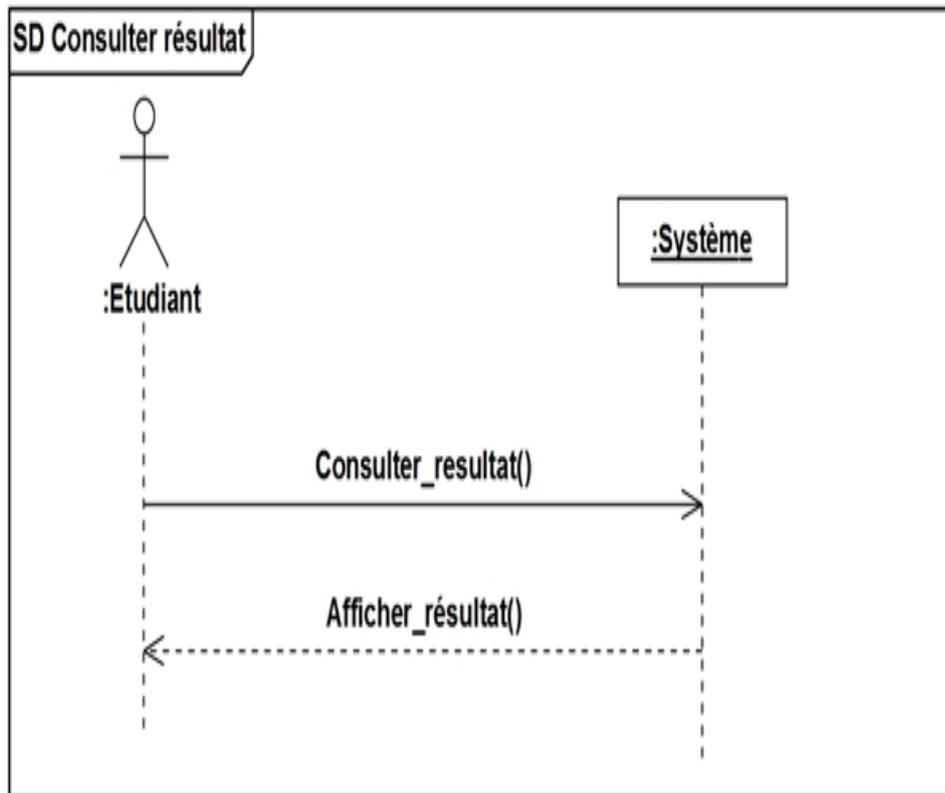


FIGURE 3.34 – Diagramme de séquence «Consulter Résultat».

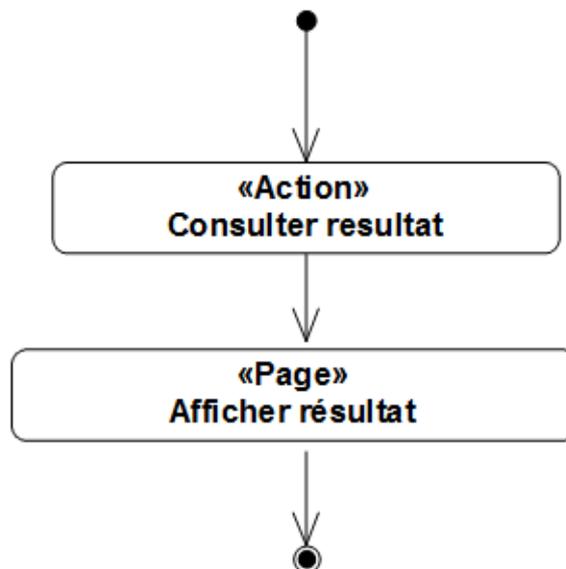


FIGURE 3.35 – Diagramme d’activité «Consulter Résultat».

3.3.2.25 Cas d'utilisation « Gérer ressources »

Nom	Gérer ressources
Acteur	Enseignant
But	Permet à Enseignant d'ajouter ou de retirer une ressource (cours ou multimédia).
Pré-condition	L'enseignant doit être authentifié.
Post-condition	/
Scénario nominal	<ol style="list-style-type: none"> 1. L'enseignant demande de gérer ses ressources. 2. Le système affiche les options suivantes : <ul style="list-style-type: none"> • Ajouter ressource. • Retirer ressource.

TABLE 3.26 – Description textuelle du cas d'utilisation « Gérer ressources ».

3.3.2.26 Cas d'utilisation «Ajouter ressource »

Nom	Ajouter ressource
Acteur	Enseignant
But	Permet à l'enseignant d'ajouter une ressource (cours ou multimédia).
Pré-condition	L'enseignant doit être authentifié.
Post-condition	Nouvelle ressource ajoutée par le système.
Scénario nominal	<ol style="list-style-type: none"> 1. L'enseignant demande d'ajouter une ressource. 2. Le système affiche le formulaire correspondant. 3. l'enseignant remplit le formulaire et valide. 4. Le système vérifie les informations entrées. 5. Le système enregistre la nouvelle ressource dans la base de données. 6. Le système affiche un message de confirmation.

Scénario Alternatif	<p>A1. L'enseignant n'a pas rempli certains champs ou a saisi des informations non valides.</p> <ol style="list-style-type: none"> 1. Le système affiche un message d'erreur. 2. Le système retourne à l'étape numéro 2 du scénario nominal.
----------------------------	--

TABLE 3.27 – Description textuelle du cas d'utilisation «Ajouter ressource ».

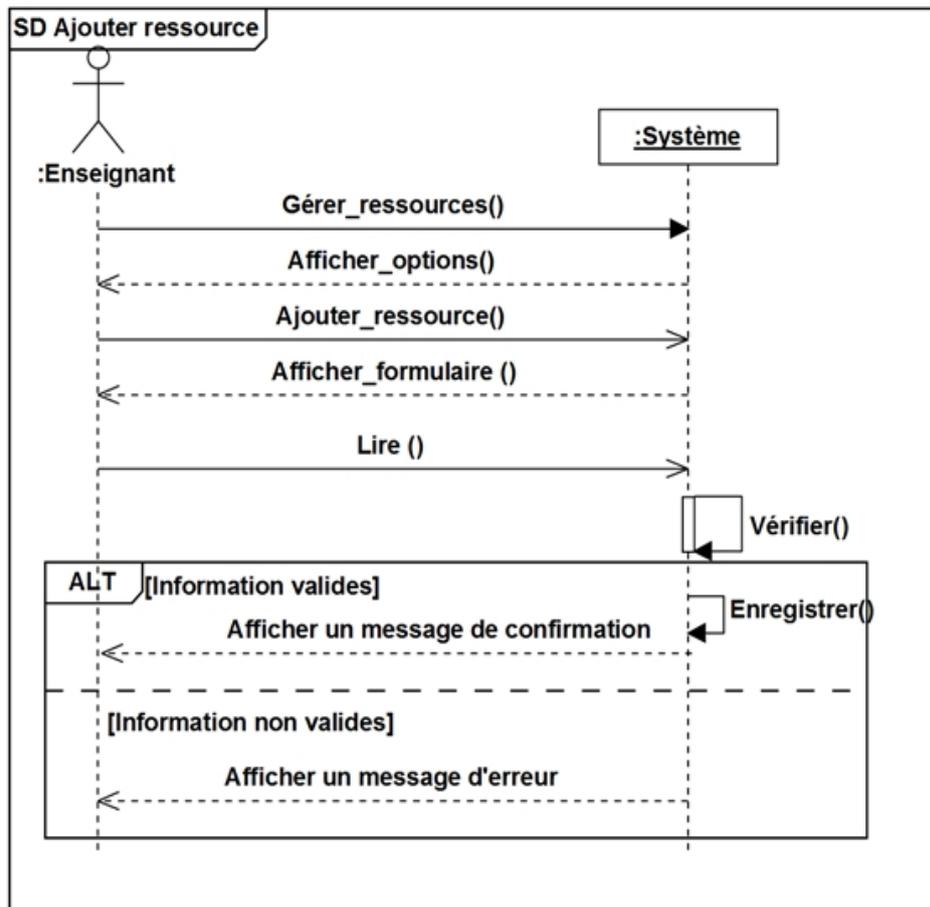


FIGURE 3.36 – Diagramme de séquence « Ajouter ressource ».

3.3.2.27 Cas d'utilisation « Retirer ressource »

Nom	Retirer ressource
Acteur	Enseignant
But	Permet à l'enseignant de retirer une ressource (cours ou multimédia).

Pré-condition	L'enseignant doit être authentifié.
Post-condition	Ressource supprimée du système.
Scénario nominal	<ol style="list-style-type: none"> 1. L'enseignant demande de retirer une ressource. 2. Le système affiche la liste des ressources. 3. L'enseignant sélectionne une ressource. 4. Le système demande de confirmer la suppression. 5. L'enseignant confirme la suppression. 6. Le système supprime la ressource de la base de données. 7. Le système affiche un message de confirmation.
Scénario Alternatif	<p>A1. L'enseignant n'a pas confirmé la suppression.</p> <ol style="list-style-type: none"> 1. Le système affiche un message d'erreur. 2. Le système retourne à l'étape numéro 2 du scénario nominal.

TABLE 3.28 – Description textuelle du cas d'utilisation «Retirer ressource».

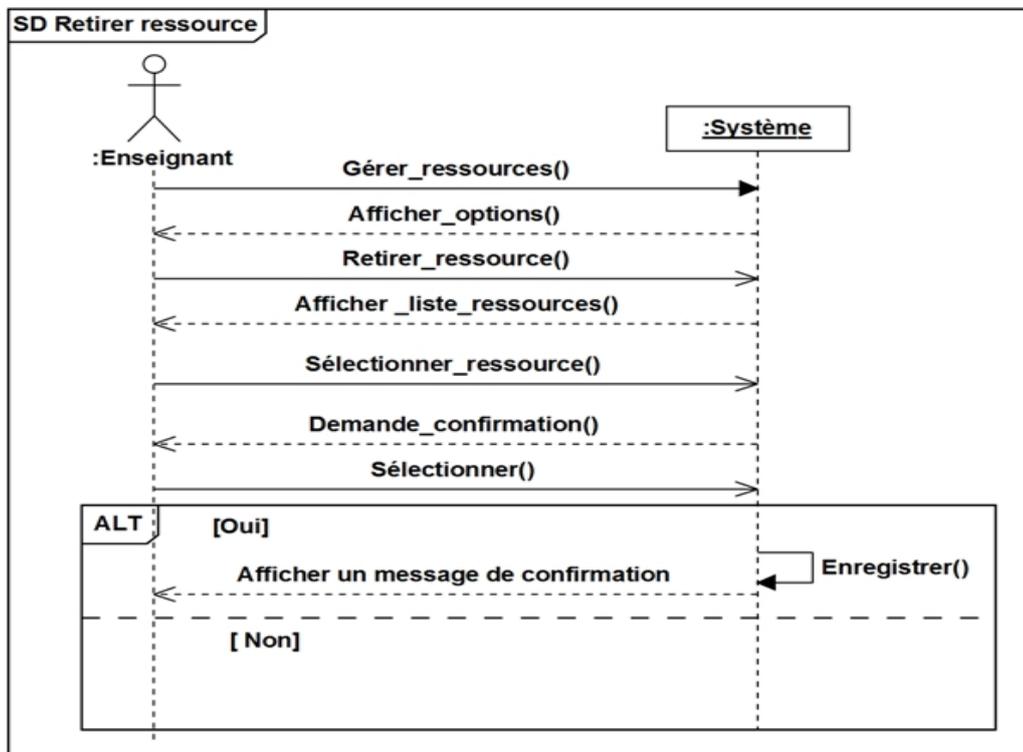


FIGURE 3.37 – Diagramme de séquence «Retirer ressource».

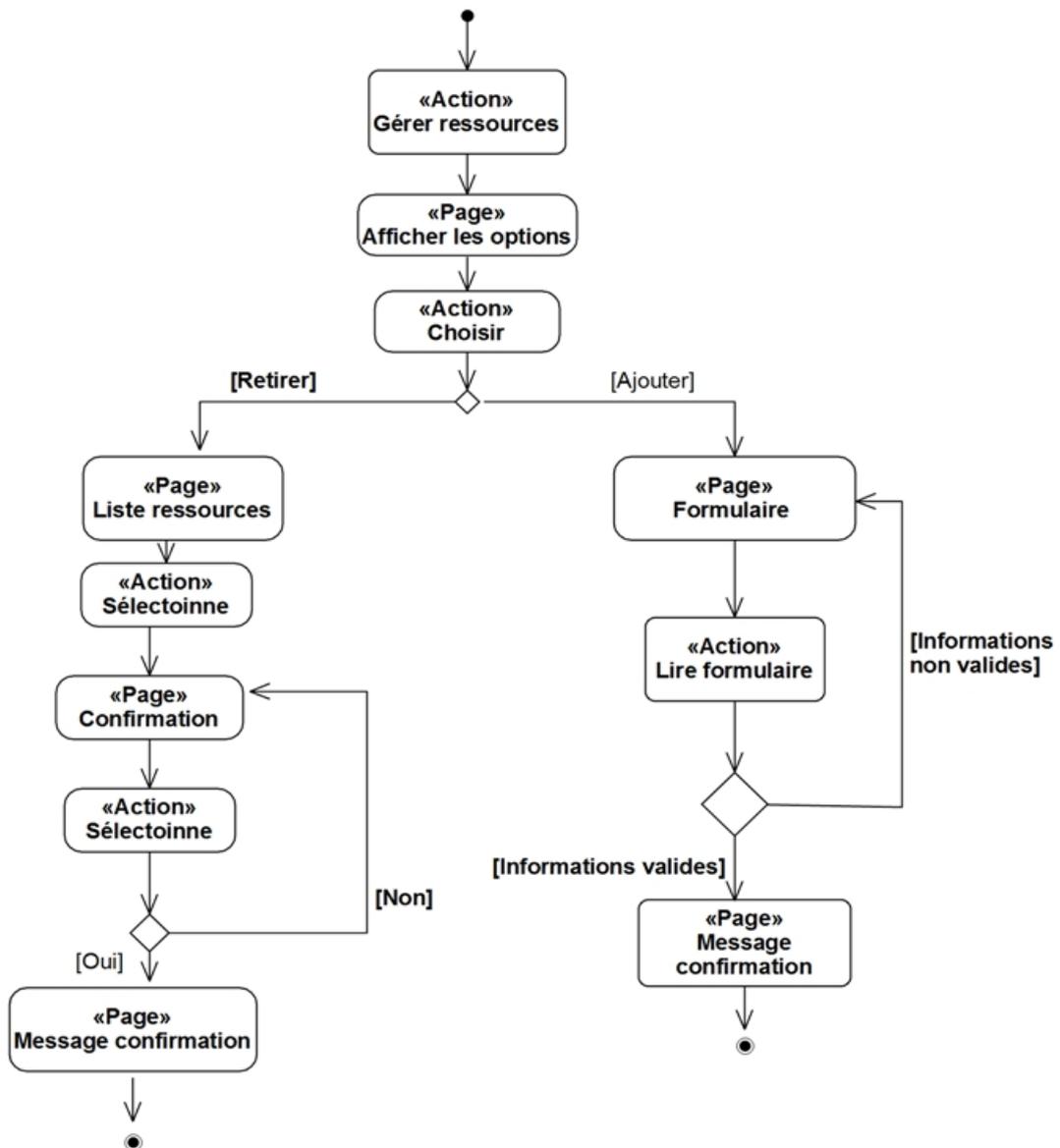


FIGURE 3.38 – Diagramme d’activité «Gérer ressources».

3.3.2.28 Cas d'utilisation « Afficher notes »

Nom	Afficher notes
Acteur	Enseignant
But	Permet à l'enseignant d'afficher les notes de tests aux étudiants.
Pré-condition	L'enseignant doit être authentifié.
Post-condition	Nouvel affichage ajouté par le système.
Scénario nominal	<ol style="list-style-type: none"> 1. L'enseignant demande d'afficher des notes. 2. Le système affiche la liste des fiches de notes. 3. L'enseignant sélectionne une fiche de notes et valide. 4. Le système enregistre le nouvel affichage dans la base de données. 5. Le système affiche un message de confirmation.

TABLE 3.29 – Description textuelle du cas d'utilisation «Afficher notes».

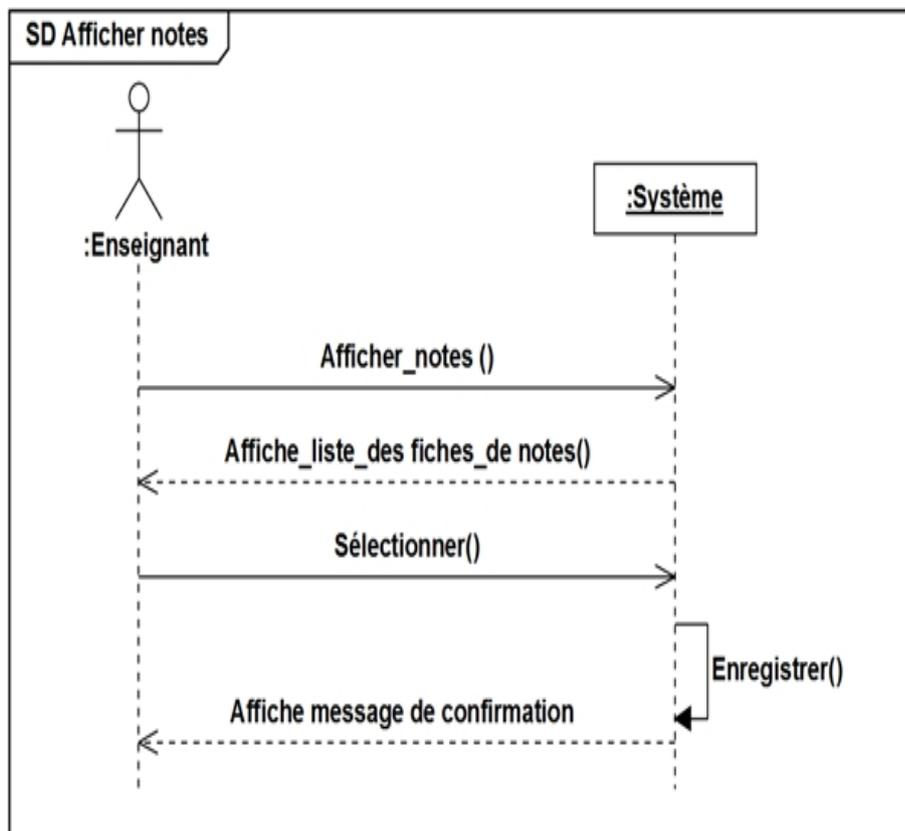


FIGURE 3.39 – Diagramme de séquence «Afficher notes».

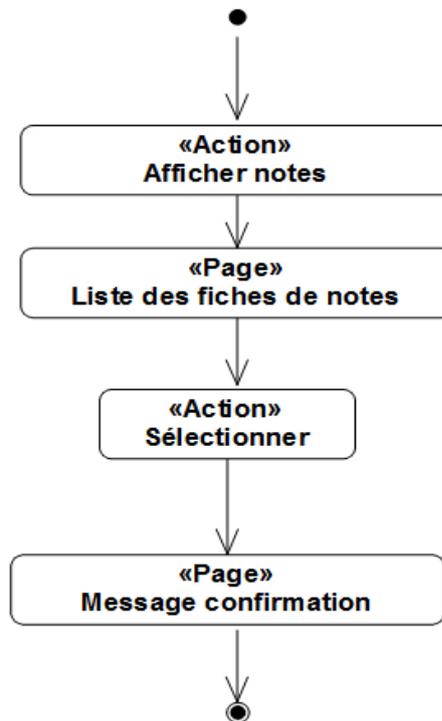


FIGURE 3.40 – Diagramme d'activité «Afficher notes».

3.3.2.29 Cas d'utilisation « Lancer événement »

Nom	Lancer événement
Acteur	Enseignant
But	Permet à l'enseignant de lancer un événement.
Pré-condition	L'enseignant doit être authentifié
Post-condition	Nouvel événement ajouté par le système.
Scénario nominal	<ol style="list-style-type: none"> 1. L'enseignant demande de lancer un événement. 2. Le système affiche le formulaire correspondant. 3. L'enseignant remplit le formulaire et valide. 4. Le système vérifie les informations entrées. 5. Le système ajoute un nouvel événement dans la base de données.

	6. Le système affiche un message de confirmation.
Scénario Alternatif	<p>A1. L'enseignant n'a pas rempli certains champs ou a saisi des informations non valides.</p> <p>1. Le système affiche un message d'erreur.</p> <p>2. Le système retourne à l'étape numéro 2 du scénario nominal.</p>

TABLE 3.30 – Description textuelle du cas d'utilisation «Lancer événement ».

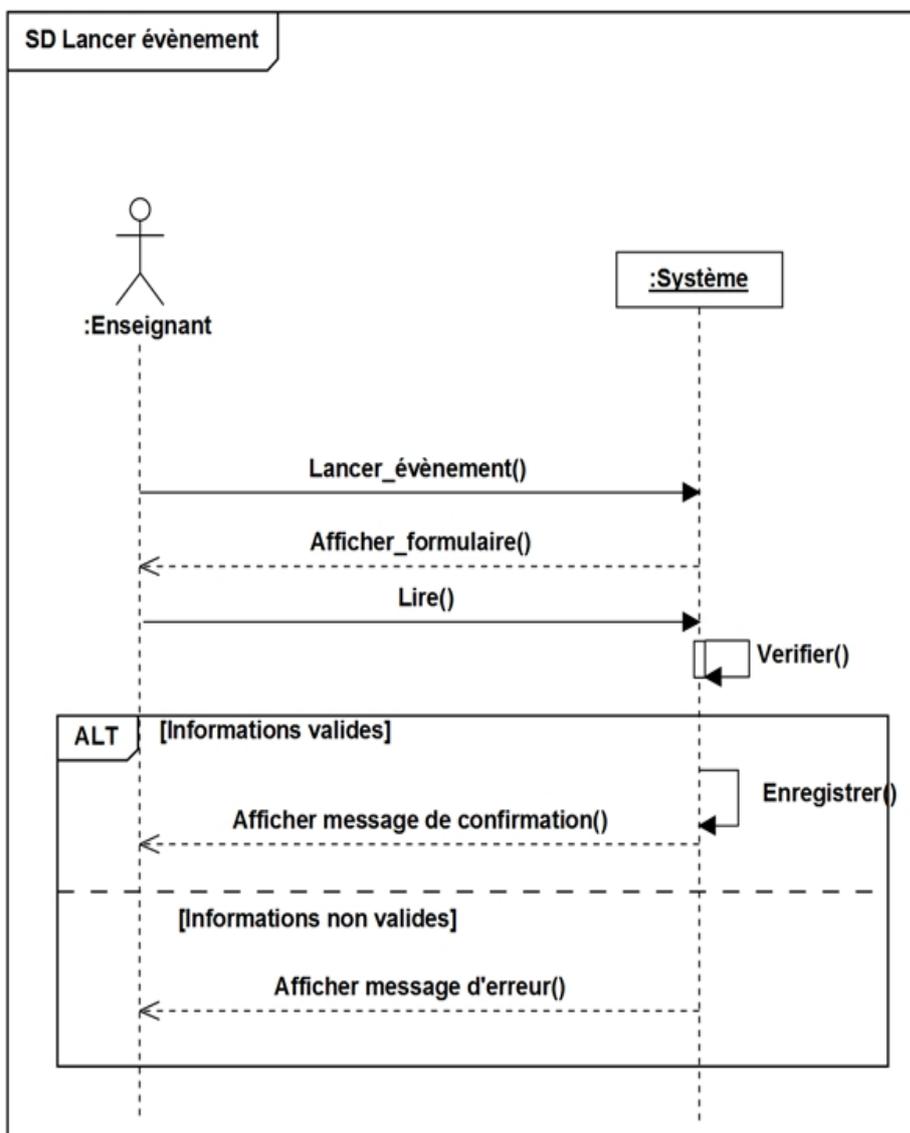


FIGURE 3.41 – Diagramme de séquence «Lancer événement».

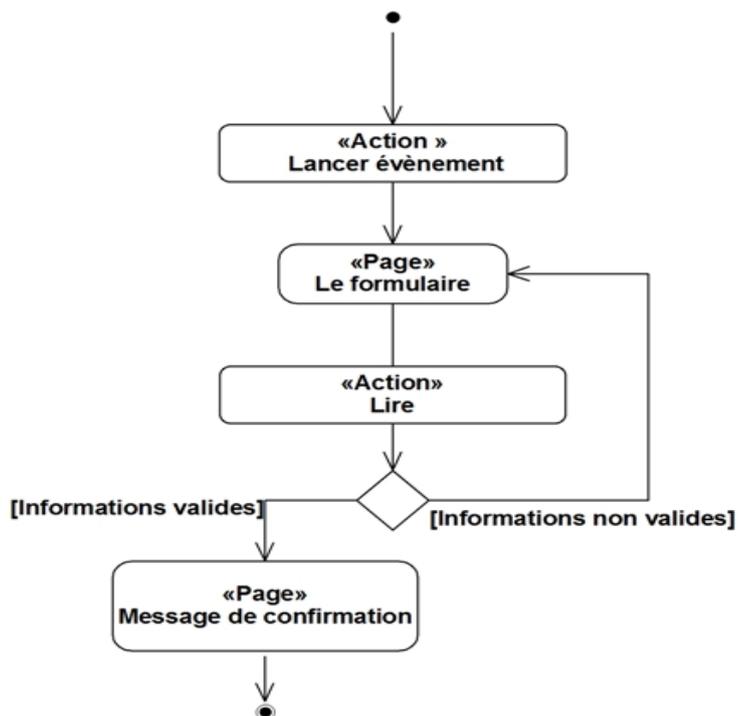


FIGURE 3.42 – Diagramme d’activité «Lancer événement».

3.3.2.30 Cas d’utilisation « Consulter Participations »

Nom	Consulter Participations
Acteur	Enseignant
But	Permet à l’enseignant de consulter les participations des étudiants aux événements.
Pré-condition	L’enseignant doit être authentifié.
Post-condition	Les participations à un événement sont affichées.
Scénario nominal	<ol style="list-style-type: none"> 1. L’enseignant demande de consulter les participations. 2. Le système affiche la liste des événements. 3. L’enseignant sélectionne un événement.

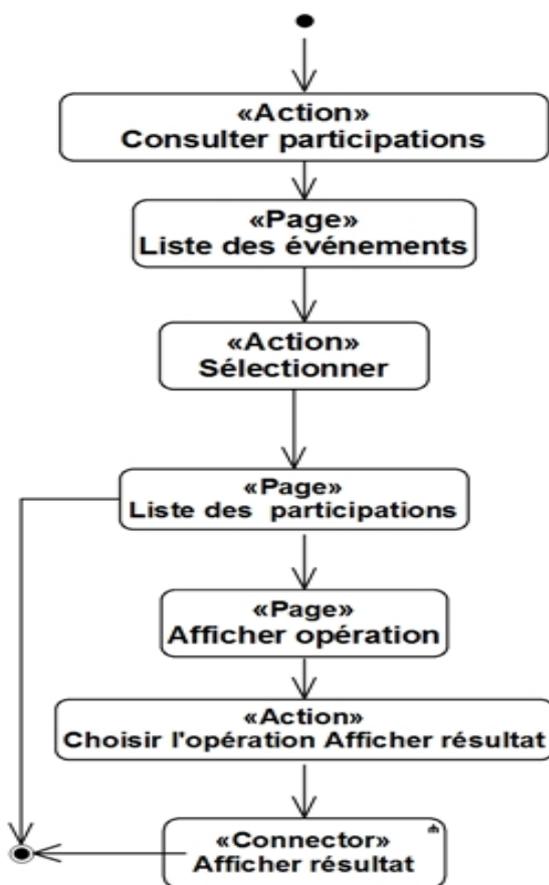


FIGURE 3.44 – Diagramme d’activité «Consulter Participations».

3.3.2.31 Cas d’utilisation « Afficher Résultat »

Nom	Afficher Résultat
Acteur	Enseignant
But	Permet à l’enseignant d’afficher le résultat d’un événement.
Pré-condition	L’enseignant doit être authentifié.
Post-condition	Un résultat est ajouté par le système.
Scénario nominal	<ol style="list-style-type: none"> 1. L’enseignant demande d’afficher le résultat d’un événement. 2. Le système affiche un champ. 3. L’enseignant remplit le champ et valide.

	<p>4. Le système vérifie les informations entrées.</p> <p>5. Le système ajoute le résultat de l'événement dans la base de données.</p> <p>6. Le système affiche un message de confirmation.</p>
Scenario Alternatif	<p>A1. L'enseignant n'a pas rempli le champ ou saisi une information non valide.</p> <p>1. Le système affiche un message d'erreur.</p> <p>2. Le système retourne à l'étape numéro 2 du scénario nominal.</p>

TABLE 3.32 – Description textuelle du cas d'utilisation «Afficher Résultat ».

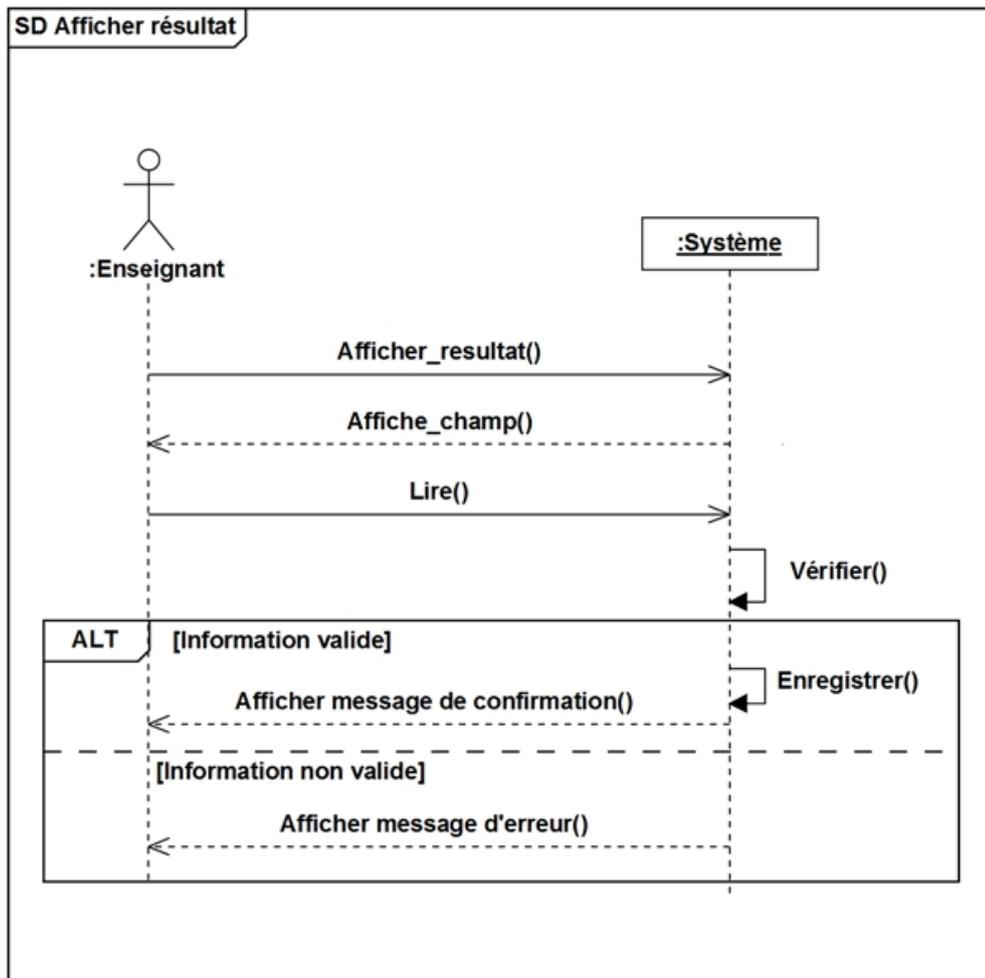


FIGURE 3.45 – Diagramme de séquence «Afficher Résultat ».

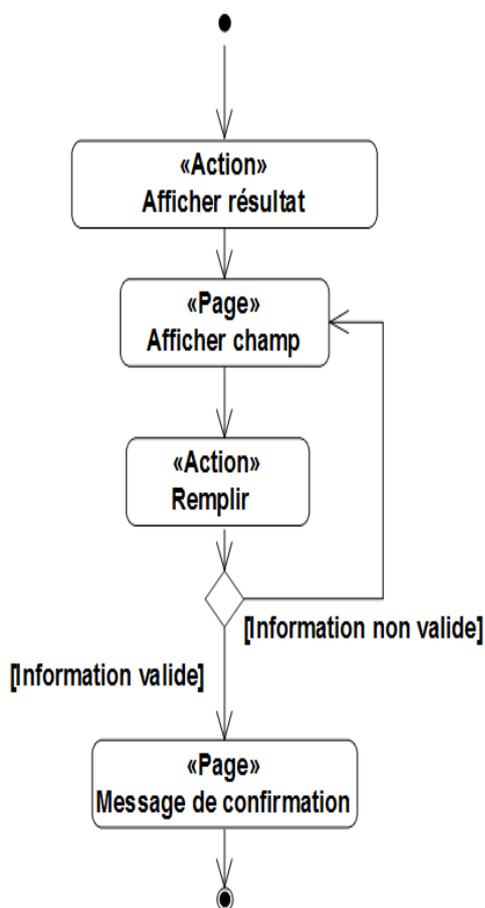


FIGURE 3.46 – Diagramme d’activité «Afficher Résultat ».

3.3.2.32 Cas d’utilisation « Consulter espace de communication »

Nom	Consulter espace de communication
Acteur	Enseignant/Étudiant
But	Permet à l’enseignant ou l’étudiant de consulter l’espace de communication.
Pré-condition	L’enseignant/ l’étudiant doit être authentifié.
Post-condition	Le contenu de l’espace de communication est affiché.
Scénario nominal	1. L’enseignant/ l’étudiant demande de consulter l’espace de communication. 2. Le système affiche les messages correspondant à la formation.

	3. Le système affiche l'opération « Ecrire message ».
Scenario Alternatif	A1. L'enseignant/l'étudiant choisit l'opération « Ecrire message ». 1. Le système fait référence au cas d'utilisation « Ecrire message ».

TABLE 3.33 – Description textuelle du cas d'utilisation «Consulter espace de communication ».

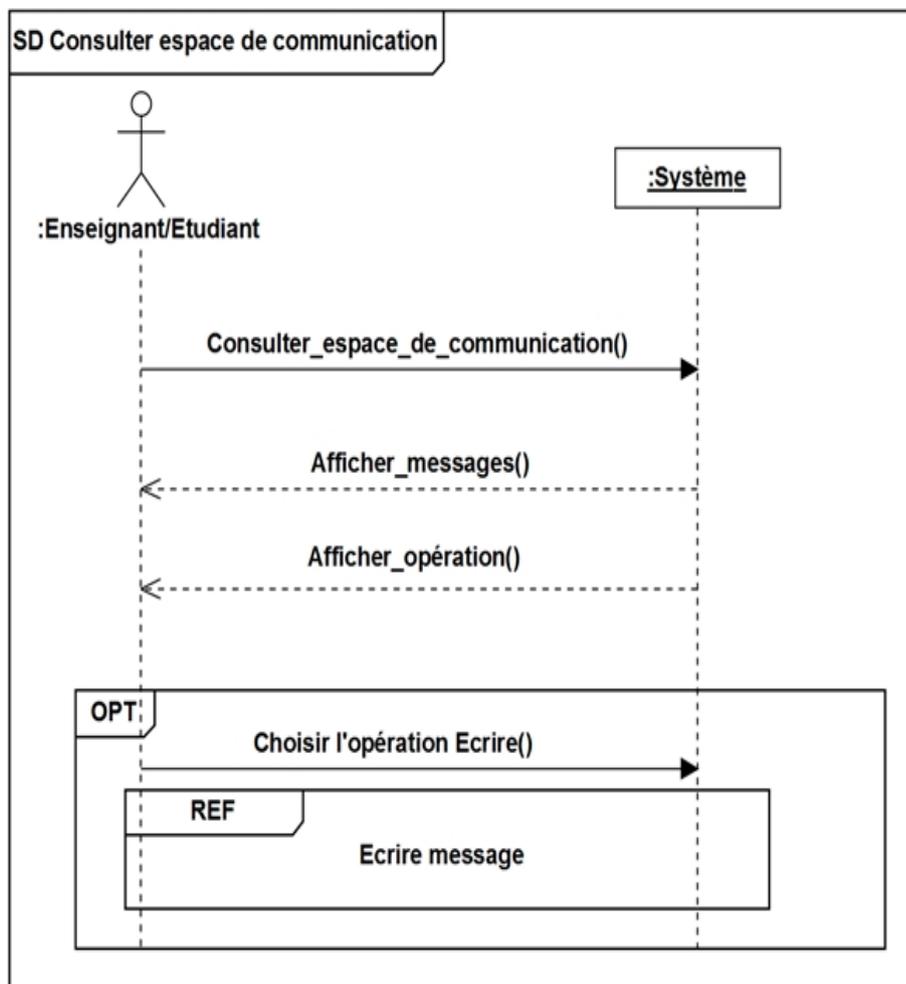


FIGURE 3.47 – Diagramme de séquence «Consulter espace de communication».

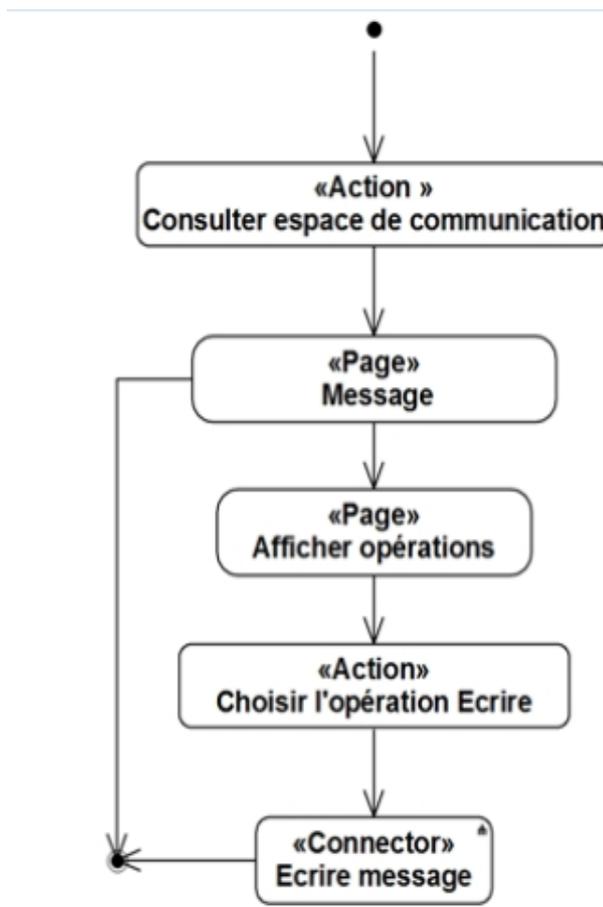


FIGURE 3.48 – Diagramme d’activité « Consulter espace de communication ».

3.3.2.33 Cas d’utilisation « Ecrire message »

Nom	Ecrire message
Acteur	Enseignant/Étudiant
But	Permet à l’enseignant ou l’étudiant d’écrire un message dans l’espace de communication.
Pré-condition	L’enseignant/ l’étudiant doit être authentifié.
Post-condition	Nouveau message ajouté par le système.
Scénario nominal	1. L’enseignant/ l’étudiant demande d’écrire un message. 2. Le système affiche un champ.

	<p>3. L'enseignant/ l'étudiant écrit le message et valide.</p> <p>4. Le système vérifie les informations entrées.</p> <p>5. Le système enregistre le message dans la base de données.</p> <p>6. Le système affiche un message de confirmation.</p>
Scenario Alternatif	<p>A1. L'enseignant /l'étudiant n'a pas rempli le champ ou a saisi des informations non valides.</p> <p>1. Le système affiche un message d'erreur.</p> <p>2. Le système retourne à l'étape numéro 2 du scénario nominal.</p>

TABLE 3.34 – Description textuelle du cas d'utilisation «Ecrire message ».

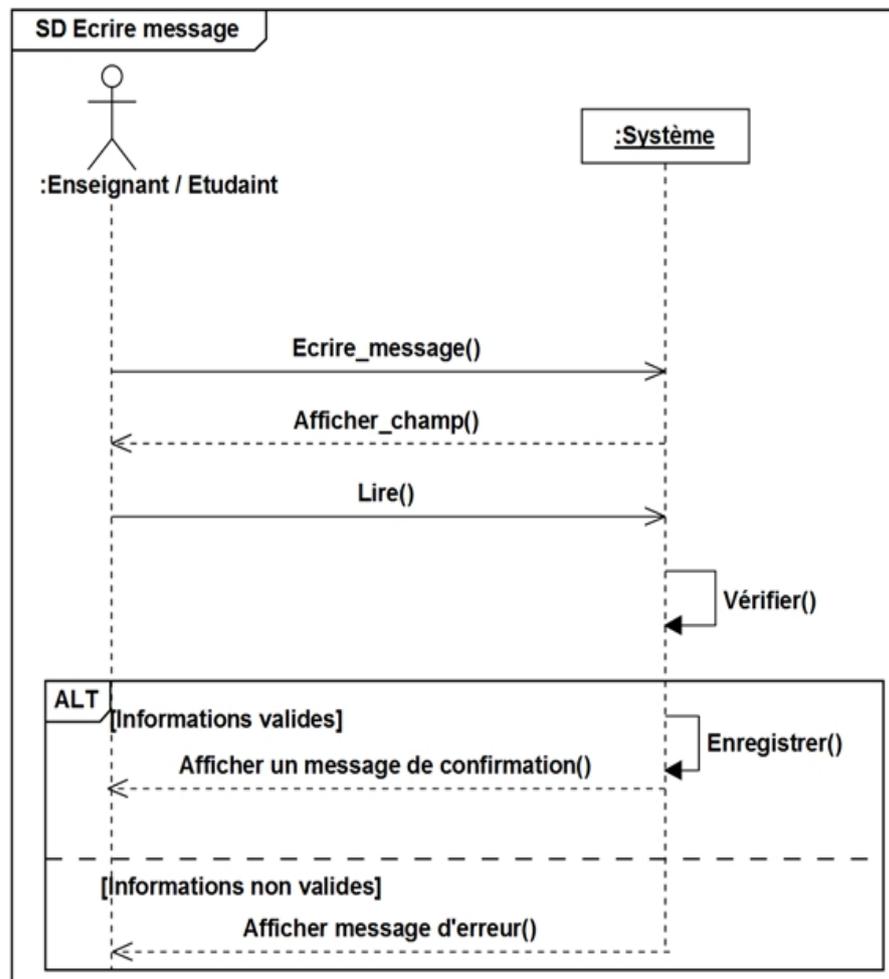


FIGURE 3.49 – Diagramme de séquence «Ecrire message».

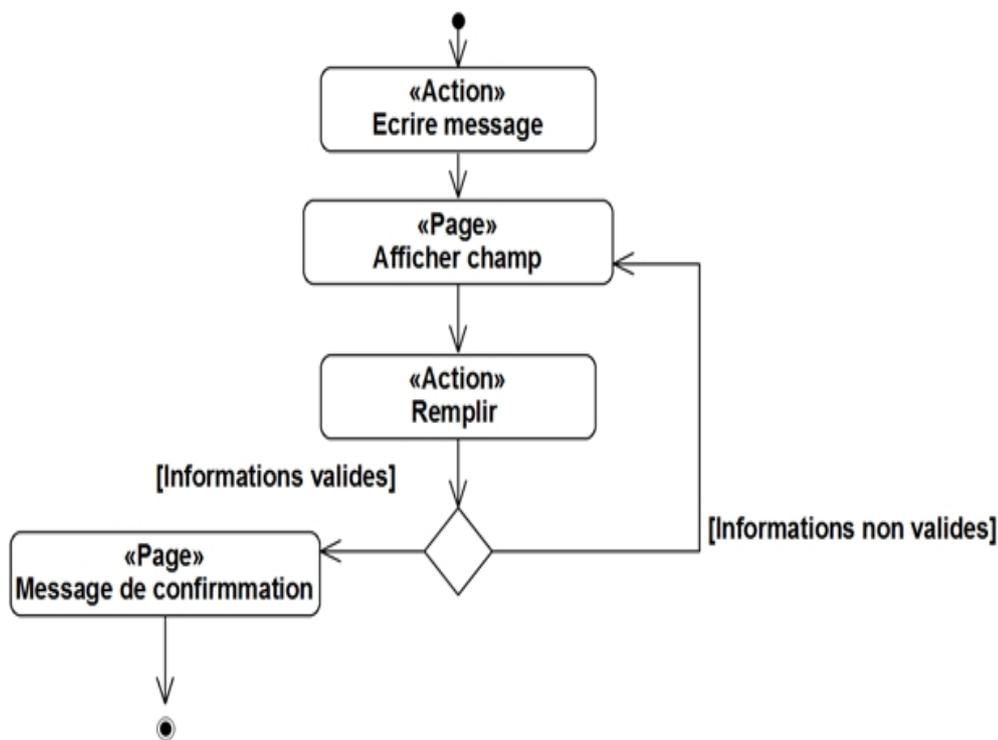


FIGURE 3.50 – Diagramme d’activité «Ecrire message ».

3.3.2.34 Cas d’utilisation « Mettre à jour notes »

Nom	Mettre à jour notes
Acteur	Enseignant
But	Permet à l’enseignant de mettre à jour ses notes.
Pré-condition	L’enseignant doit être authentifié.
Post-condition	La mise à jour est enregistrée par le système.
Scénario nominal	<ol style="list-style-type: none"> 1. L’enseignant demande de mettre à jour ses notes. 2. Le système affiche le formulaire des notes. 3. L’enseignant met à jour les notes. 4. Le système vérifie les informations entrées. 5. Le système enregistre la mise à jour dans la base de données. 6. Le système affiche un message de confirmation.

Scenario Alternatif	<p>A1. L'enseignant n'a pas rempli certains champs ou a saisi des informations non valides.</p> <ol style="list-style-type: none"> 1. Le système affiche un message d'erreur. 2. Le système retourne à l'étape numéro 2 du scénario nominal.
----------------------------	--

TABLE 3.35 – Description textuelle du cas d'utilisation «Mettre à jour notes».

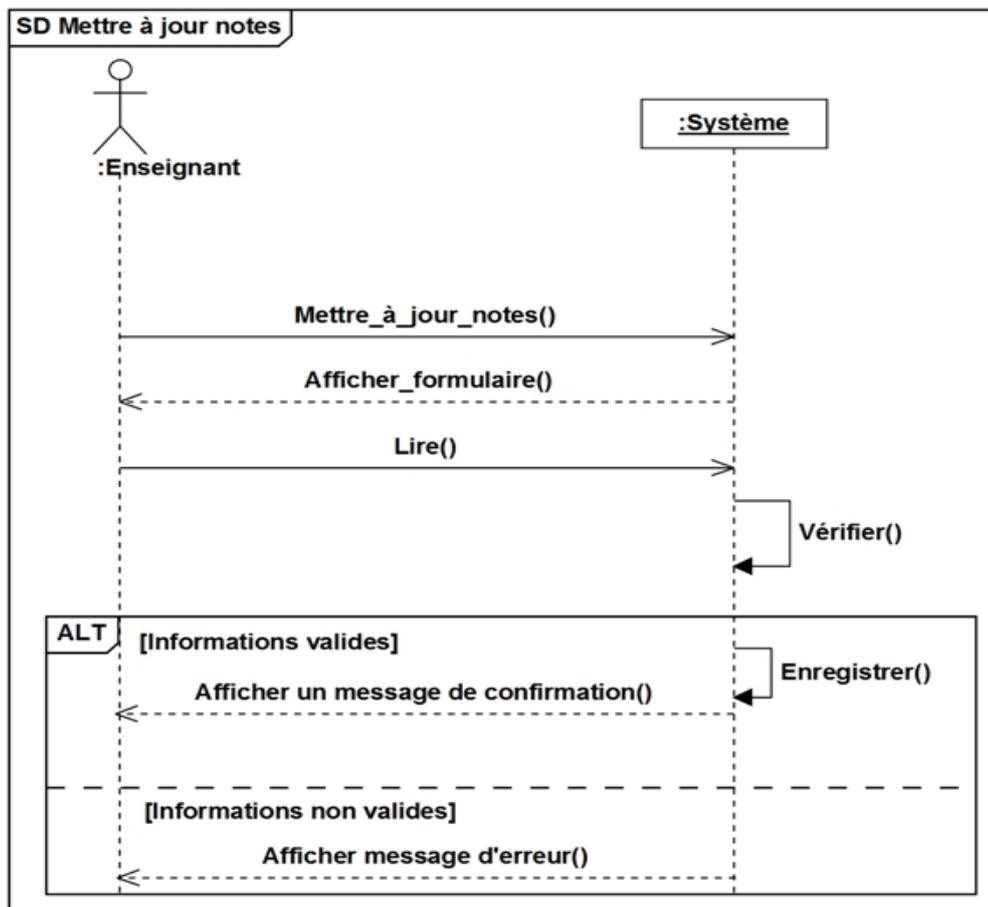


FIGURE 3.51 – Diagramme de séquence «Mettre à jour notes».

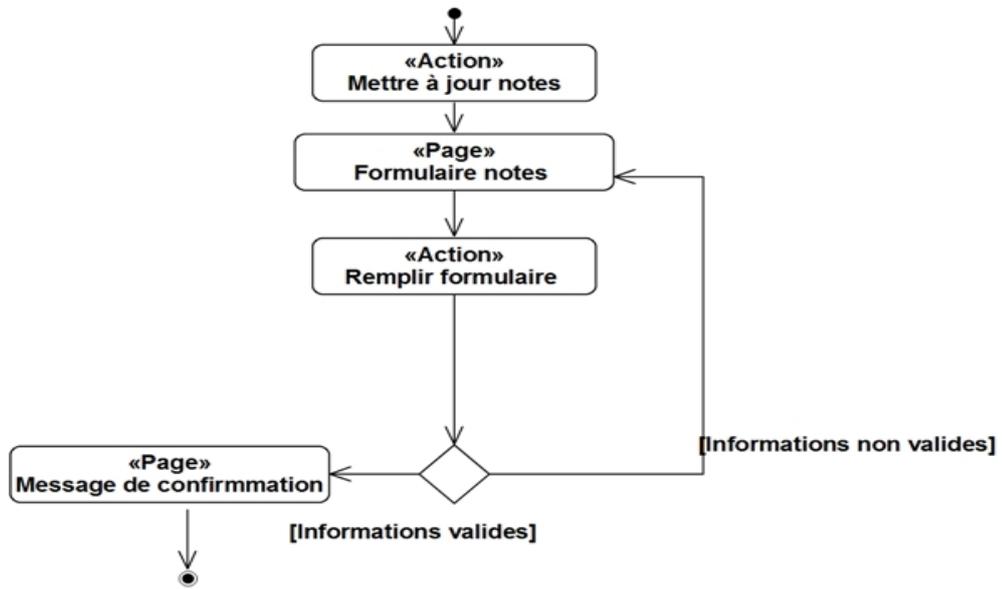


FIGURE 3.52 – Diagramme d’activité «Mettre à jour notes ».

3.4 LE MODELE DE DOMAINE

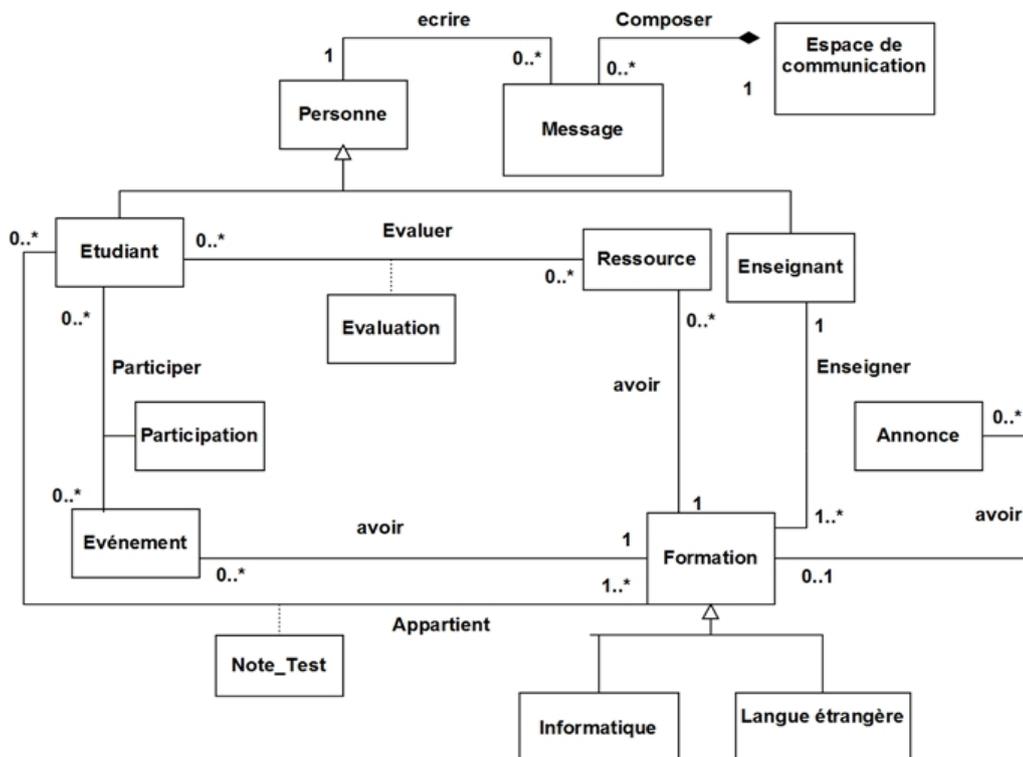


FIGURE 3.53 – Le modèle de domaine.

3.5 CAPTURE DES BESIIONS TECHNIQUES

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.

3.5.1 LA CONFIGURATION METERAILE DU SYSTEME

La configuration géographique du système impose le développement d'une solution (client/serveur). La configuration est schématisée par le diagramme de déploiement.

3.5.1.1 Développement du modèle de déploiement

Le diagramme de déploiement spécifie un ensemble de constructions qui peuvent être utilisées pour définir l'architecture d'exécution du système. Celle-ci représente l'affectation d'artefacts logiciels à des nœuds. Les nœuds sont connectés via des chemins de communication créant ainsi un réseau d'une complexité quelconque. Les nœuds sont en général définis d'une manière imbriquée et représentent soit des périphériques matériels, soit des environnements d'exécution de logiciels. Pour cela Nous suivons une architecture ci-dessous.

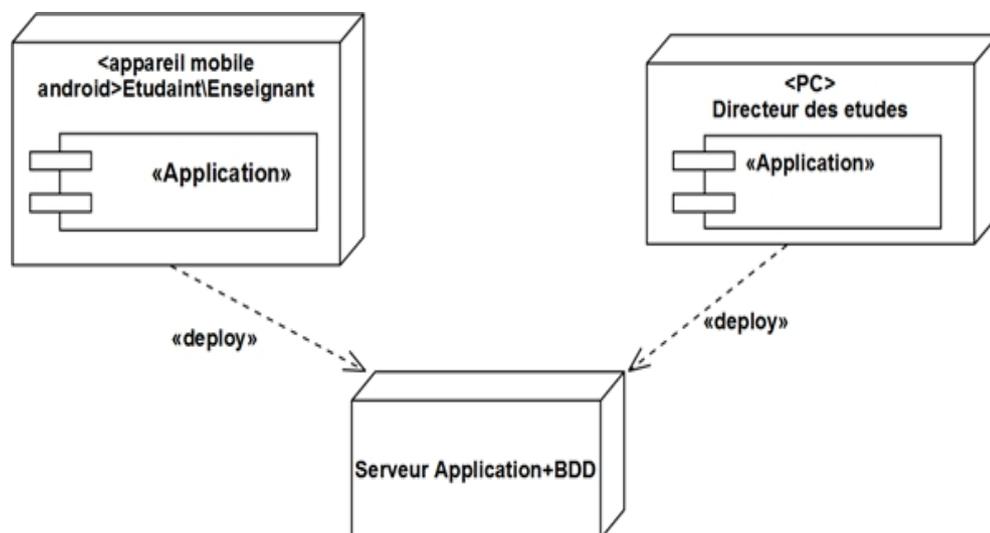


FIGURE 3.54 – Diagramme de déploiement.

3.5.1.2 Spécification d'architecture 2 tiers

Dans une architecture deux tiers, encore appelée client-serveur de première génération ou client-serveur de données, le poste client se contente de déléguer la gestion des données à un service spécialisé.

La gestion des données est prise en charge par un SGBD centralisé, s'exécutant le plus souvent sur un serveur dédié. Ce dernier est interrogé en utilisant un langage de requête qui, le plus souvent, est SQL. Le dialogue entre client et serveur se résume donc à l'envoi de requêtes et au retour des données correspondant aux requêtes [17].

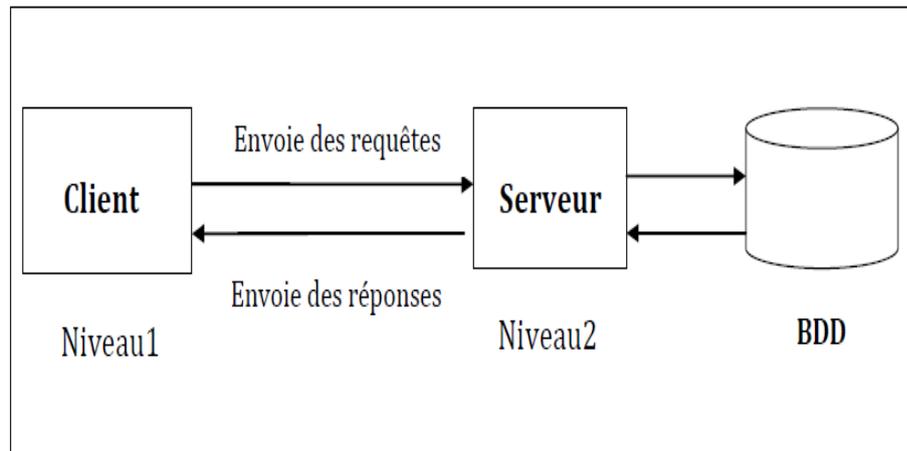


FIGURE 3.55 – Architecture 2 niveaux de notre système.

3.5.2 CAPTURE DES SPECIFICATIONS LOGICIELLES

3.5.2.1 Identification des exploitants du système

Un exploitant est un utilisateur dans son sens le plus général qui bénéficie des fonctionnalités techniques du système.

3.5.2.2 Identification des cas d'utilisations techniques

Destiné à l'exploitant, un cas d'utilisation technique est une séquence d'actions produisant une valeur ajoutée opérationnelle ou technique mais pas fonctionnelle.

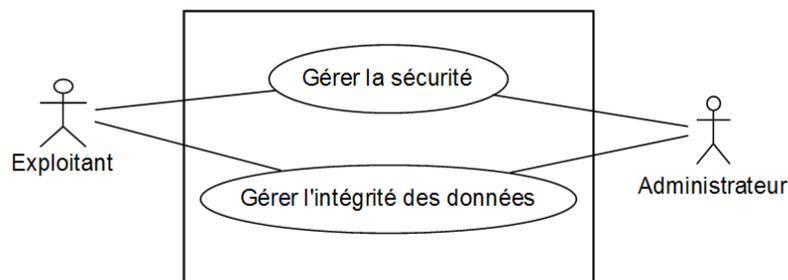


FIGURE 3.56 – Modèle de spécification logicielle du système.

3.5.2.3 Description des cas d'utilisation techniques

1. Cas d'utilisation «Gérer sécurité»

a) Cas d'utilisation « S'authentifier »

Nom	S'authentifier
Acteur	Administrateur/ Exploitant
But	Permet à L'administrateur/ L'exploitant d'accéder au système.
Pré-condition	L'administrateur/ L'exploitant possède un compte.
Post-condition	L'administrateur/ L'exploitant est authentifié par le système.
Scénario nominal	<ol style="list-style-type: none"> 1. L'administrateur/ L'exploitant demande de s'authentifier. 2. Le système affiche le formulaire d'authentification. 3. L'administrateur/ L'exploitant saisie les informations nécessaires et valide. 4. Le système vérifie les informations entrées. 5. Le système affiche la page d'accueil de l'application.
Scénario Alternatif	<p>A1. L'administrateur /L'exploitant n'a pas rempli certains champs ou a saisi des informations non valides.</p> <ol style="list-style-type: none"> 1. Le système affiche un message d'erreur. 2. Le système retourne à l'étape numéro 2 du scénario nominal.

TABLE 3.36 – Description textuelle du cas d'utilisation «S'authentifier ».

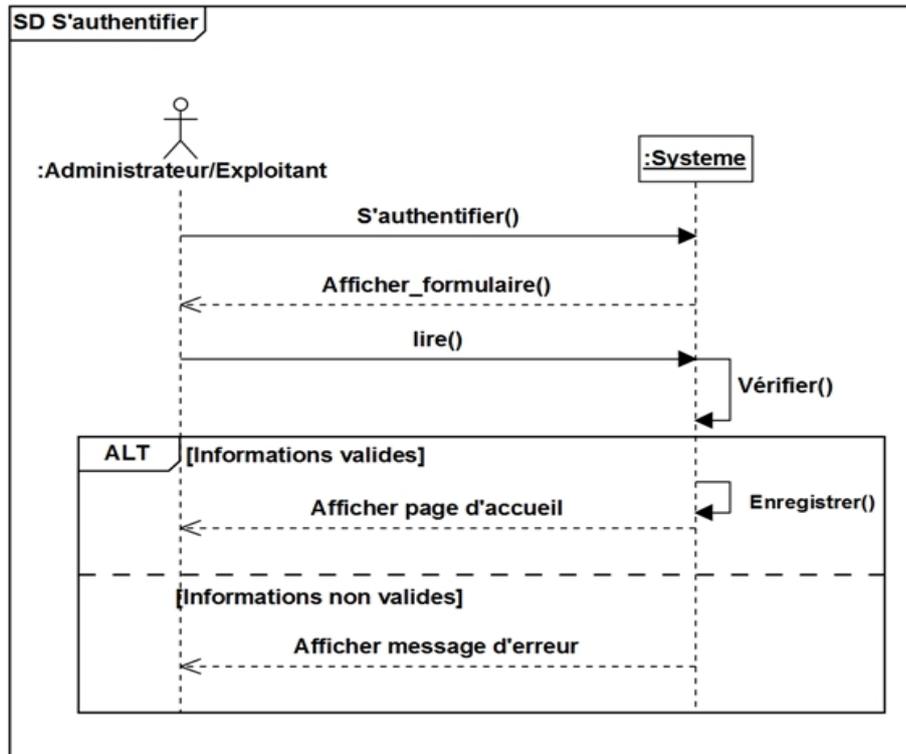


FIGURE 3.57 – Diagramme de séquence «S'authentifier».

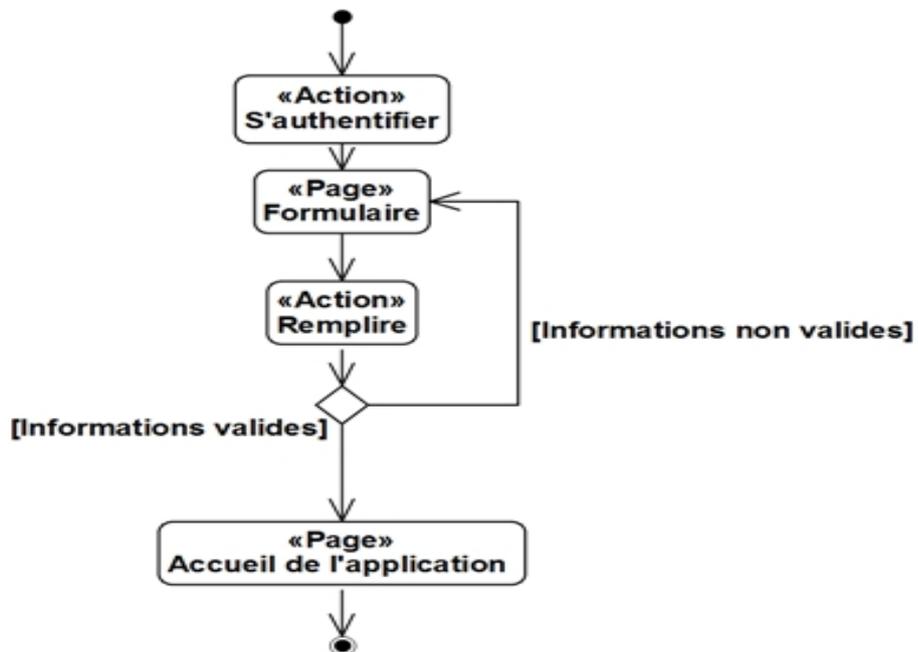


FIGURE 3.58 – Diagramme d'activité «S'authentifier».

b) Cas d'utilisation « Gérer comptes »

Nom	Gérer comptes
Acteur	Administrateur
But	Permet au directeur des études de consulter, d'ajouter, de modifier ou de supprimer un compte d'utilisateur.
Pré-condition	L'administrateur doit être authentifié.
Post-condition	/
Scénario nominal	<ol style="list-style-type: none"> 1. L'administrateur demande de gérer les comptes. 2. Le système affiche les options suivantes : <ul style="list-style-type: none"> • Consulter comptes. • Ajouter compte. • Modifier compte. • Supprimer compte.

TABLE 3.37 – Description textuelle du cas d'utilisation « Gérer comptes ».

c) Cas d'utilisation «Ajouter compte »

Nom	Ajouter compte
Acteur	Administrateur
But	Permet à l'administrateur d'ajouter un compte d'utilisateur.
Pré-condition	L'administrateur doit être authentifié.
Post-condition	Nouveau compte ajoutée par le système.
Scénario nominal	<ol style="list-style-type: none"> 1. L'administrateur demande d'ajouter un compte. 2. Le système affiche le formulaire correspondant. 3. L'administrateur remplit le formulaire et valide. 4. Le système vérifie les informations entrées. 5. Le système enregistre un Nouveau compte dans la base de données. 6. Le système affiche un message de confirmation.

Scénario Alternatif	<p>A1. L'administrateur n'a pas rempli certains champs ou a saisi des informations non valides.</p> <ol style="list-style-type: none"> 1. Le système affiche un message d'erreur. 2. Le système retourne à l'étape numéro 2 du scénario nominal. <p>A2. Le compte existe déjà dans la base de données.</p> <ol style="list-style-type: none"> 1. Le système affiche un message d'erreur. 2. Le système retourne à l'étape numéro 2 du scénario nominal.
----------------------------	---

TABLE 3.38 – Description textuelle du cas d'utilisation «Ajouter compte ».

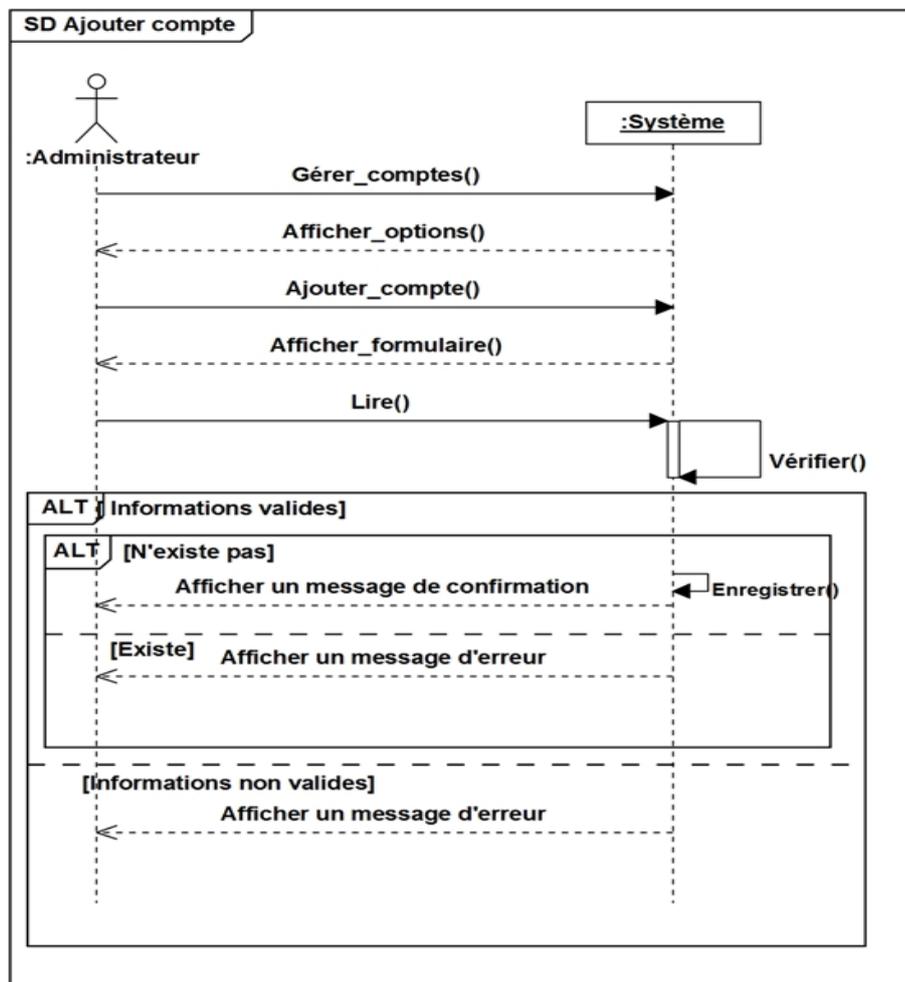


FIGURE 3.59 – Diagramme de séquence «Ajouter compte».

d) Cas d'utilisation «Modifier compte»

Nom	Modifier compte
Acteur	Administrateur
But	Permet à L'administrateur de modifier un compte utilisateur.
Pré-condition	L'administrateur doit être authentifié.
Post-condition	Les modifications sont enregistrées par le système.
Scénario nominal	<ol style="list-style-type: none"> 1. L'administrateur demande de modifier un compte. 2. Le système affiche la liste des comptes. 3. L'administrateur sélectionne un compte. 4. Le système affiche les informations d'un compte. 5. L'administrateur modifie et valide. 6. Le système vérifie les informations entrées. 7. Le système enregistre les modifications dans la base de données. 8. Le système affiche un message de confirmation.
Scénario Alternatif	<p>A1. L'administrateur n'a pas rempli certains champs ou a saisi des informations non valides.</p> <ol style="list-style-type: none"> 1. Le système affiche un message d'erreur. 2. Le système retourne à l'étape numéro 4 du scénario nominal.

TABLE 3.39 – Description textuelle du cas d'utilisation «Modifier compte ».

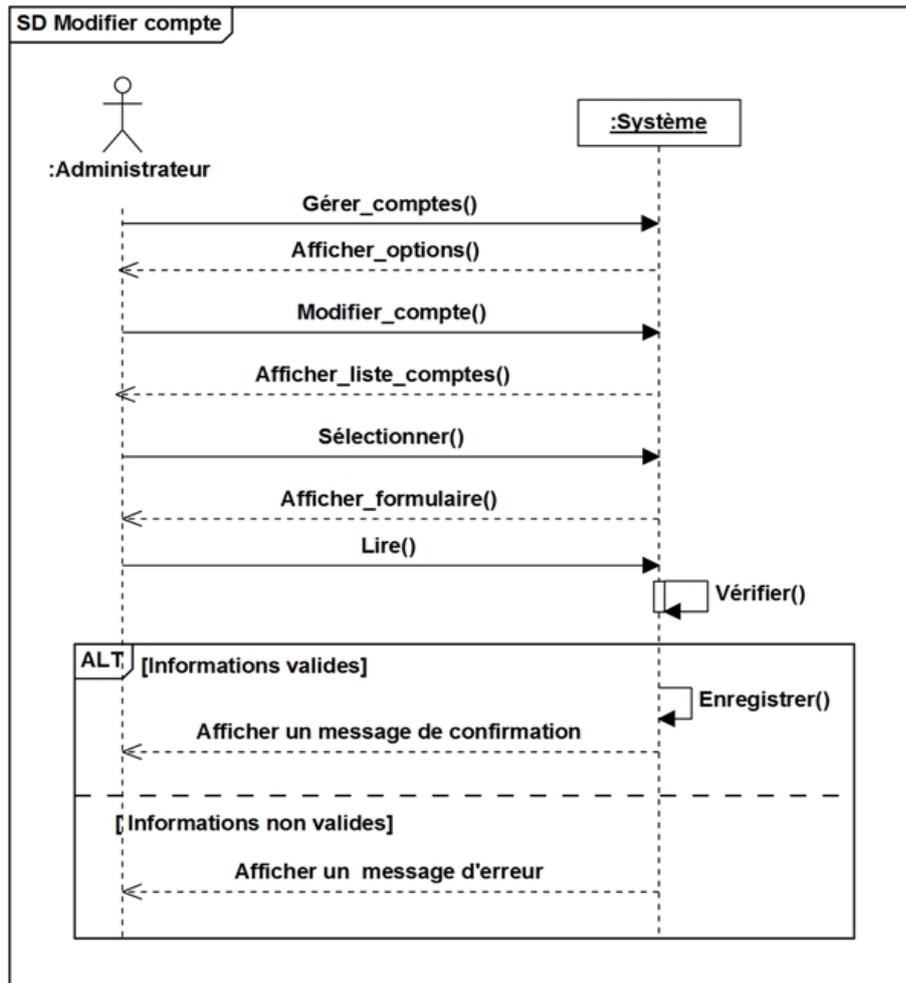


FIGURE 3.60 – Diagramme de séquence «Modifier compte».

e) Cas d'utilisation «Supprimer compte»

Nom	Supprimer compte
Acteur	Administrateur
But	Permet à L'administrateur de supprimer un compte utilisateur.
Pré-condition	L'administrateur doit être authentifié.
Post-condition	Compte supprimée du système.
Scénario nominal	1. L'administrateur demande de supprimer un compte. 2. Le système affiche la liste des comptes.

	<p>3. L'administrateur sélectionne un compte.</p> <p>4. Le système demande de confirmer la suppression.</p> <p>5. L'administrateur confirme la suppression.</p> <p>6. Le système supprime le compte de la base de données.</p> <p>7. Le système affiche un message de confirmation.</p>
Scénario Alternatif	<p>A1. L'administrateur ne confirme pas la suppression.</p> <p>1. Le système retourne à l'étape numéro 2 du scénario nominal.</p>

TABLE 3.40 – Description textuelle du cas d'utilisation «Supprimer compte».

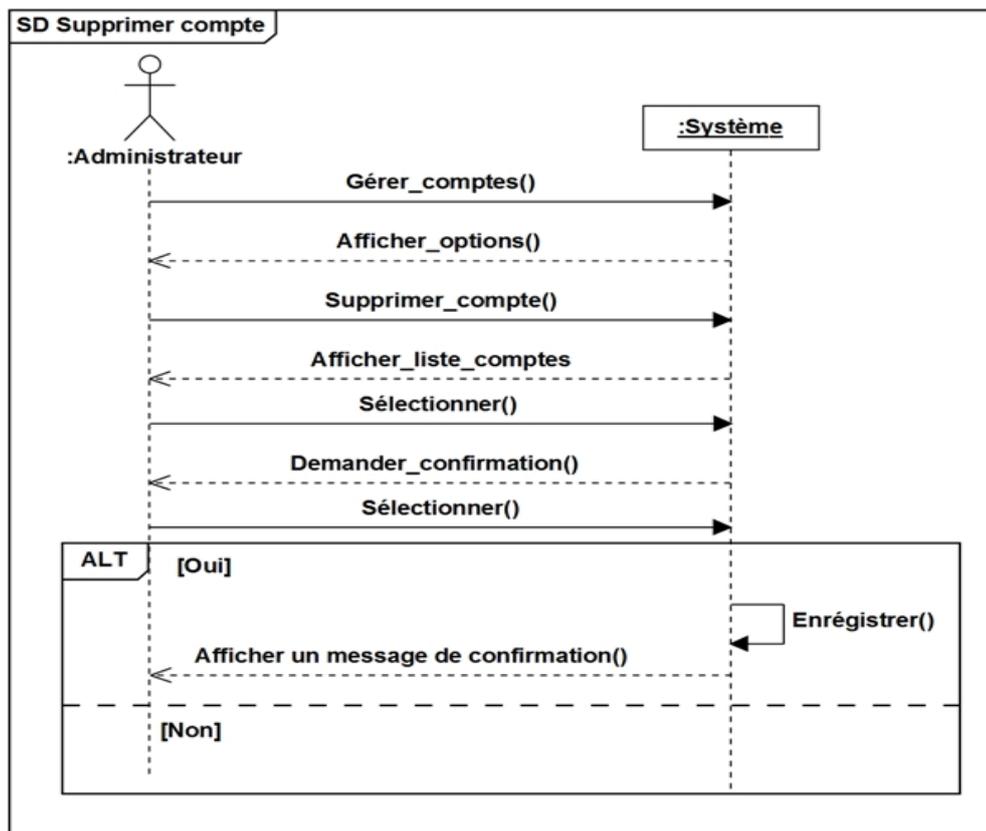


FIGURE 3.61 – Diagramme de séquence «Supprimer compte».

f) d'utilisation «Consulter comptes»

Nom	Consulter comptes
Acteur	Administrateur
But	Permet à l'administrateur de consulter les comptes des utilisateurs.
Pré-condition	L'administrateur doit être authentifié.
Post-condition	Informations d'un compte affichées.
Scénario nominal	<ol style="list-style-type: none"> 1. L'administrateur demande de consulter les comptes. 2. Le système affiche la liste des comptes. 3. Le directeur des études choisit un compte. 4. Le système affiche les informations du compte.

TABLE 3.41 – Description textuelle du cas d'utilisation «Consulter comptes».

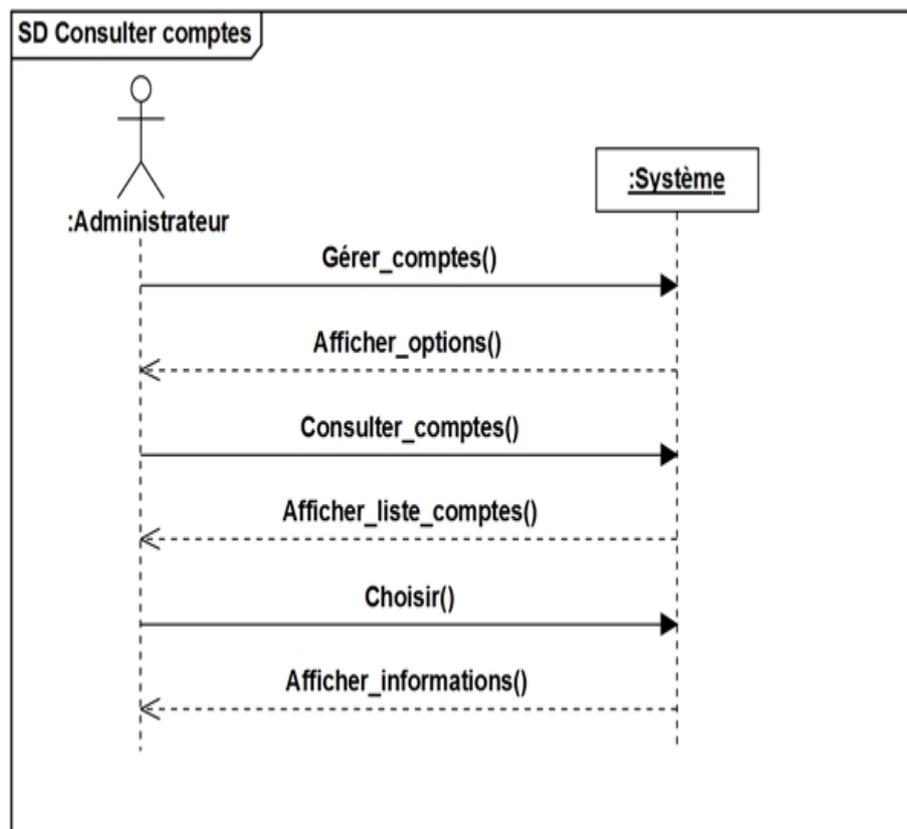


FIGURE 3.62 – Diagramme de séquence «Consulter comptes».

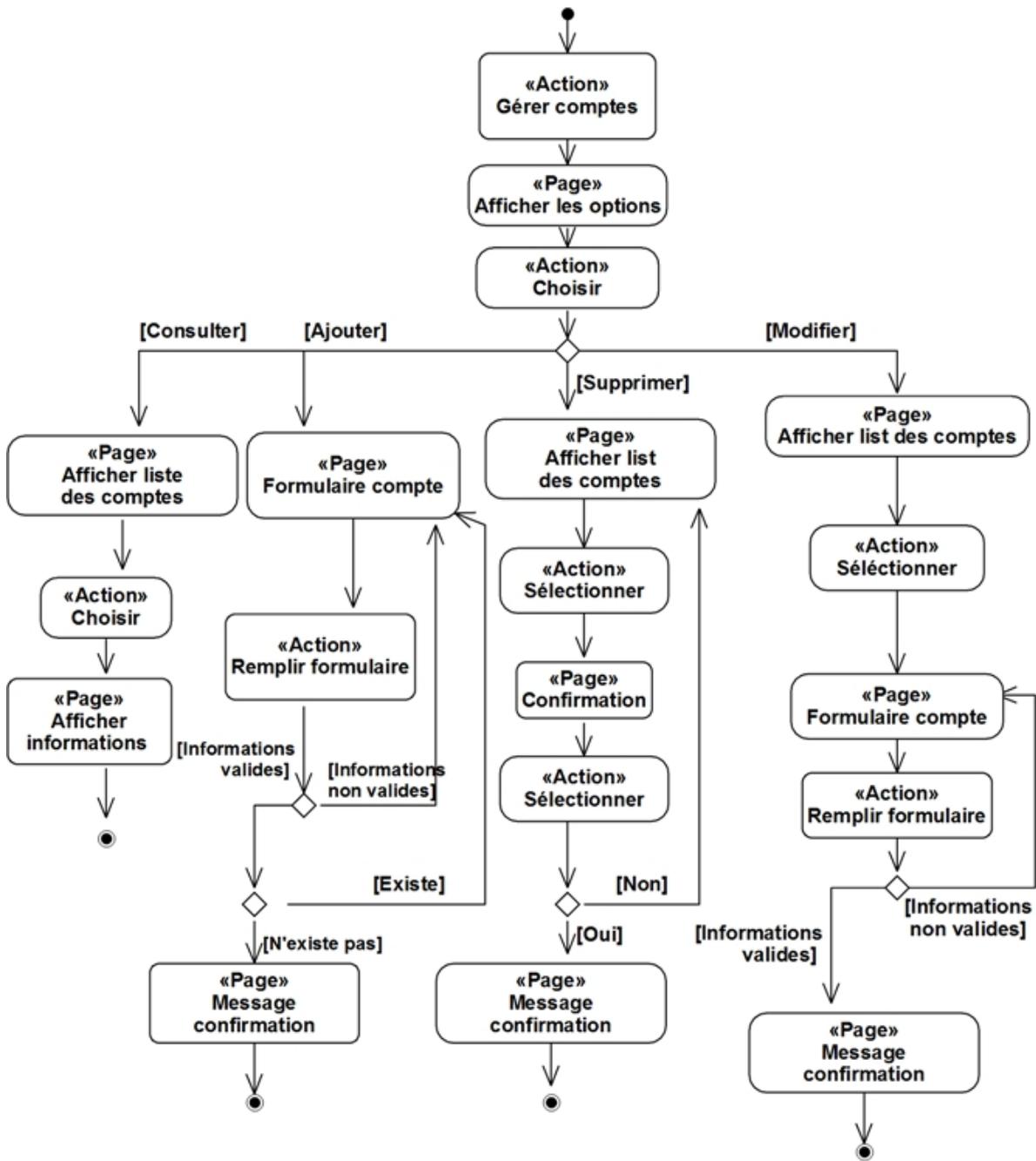


FIGURE 3.63 – Diagramme d’activité «Gérer comptes ».

3.6 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 les prochaines étapes d'analyse et conception.

Chapitre 4

Analyse et conception

4.1 INTRODUCTION

Ce chapitre est consacré à la phase d'analyse et celle de la conception. Nous commençons par l'analyse qui représente la deuxième étape de la branche gauche du cycle en Y. Par la suite, nous passons à la phase de conception qui comprend la conception préliminaire et détaillée.

4.2 PHASE D'ANALYSE

Elle consiste à étudier précisément les spécifications fonctionnelles de manière à obtenir une idée de ce que va réaliser le système en terme de métier. Elle est constituée de 3 activités :

- Le découpage en catégories.
- Le développement du modèle statique.
- Le développement du modèle dynamique.

4.2.1 DÉCOUPAGE EN CATÉGORIES

Le découpage en catégories constitue la première activité de l'étape d'analyse. Il 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.

4.2.1.1 Définition de catégorie

Une catégorie est un regroupement logique de classes à fort cohérence interne et faible couplage externe, dans la notion UML elle est représentée graphiquement par des stéréotypes des package.

4.2.1.2 Répartition des classes candidates en catégories

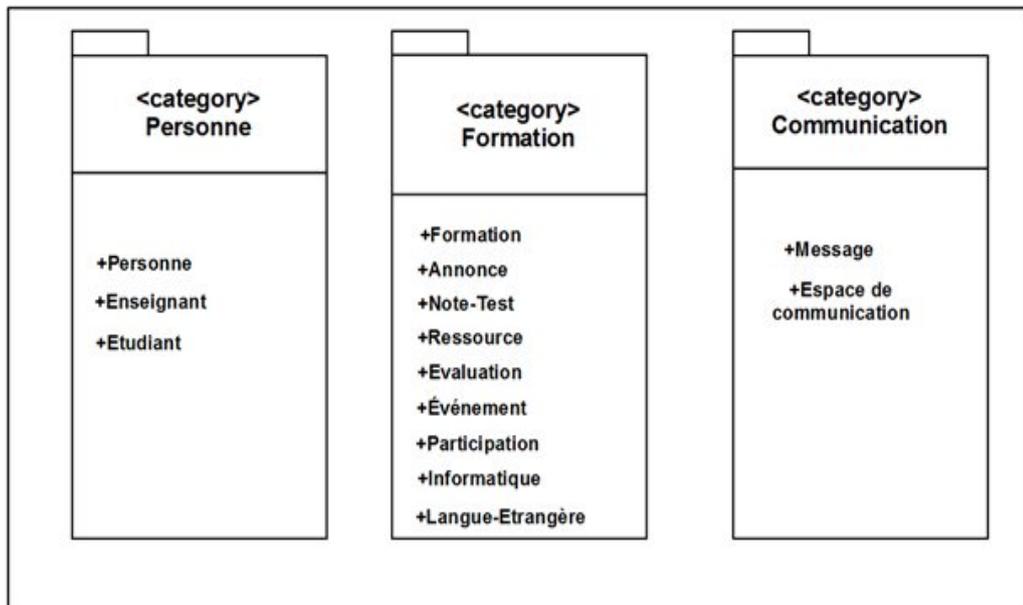


FIGURE 4.1 – Le découpage en catégories.

4.2.1.3 Élaboration des diagrammes de classes préliminaire par catégorie

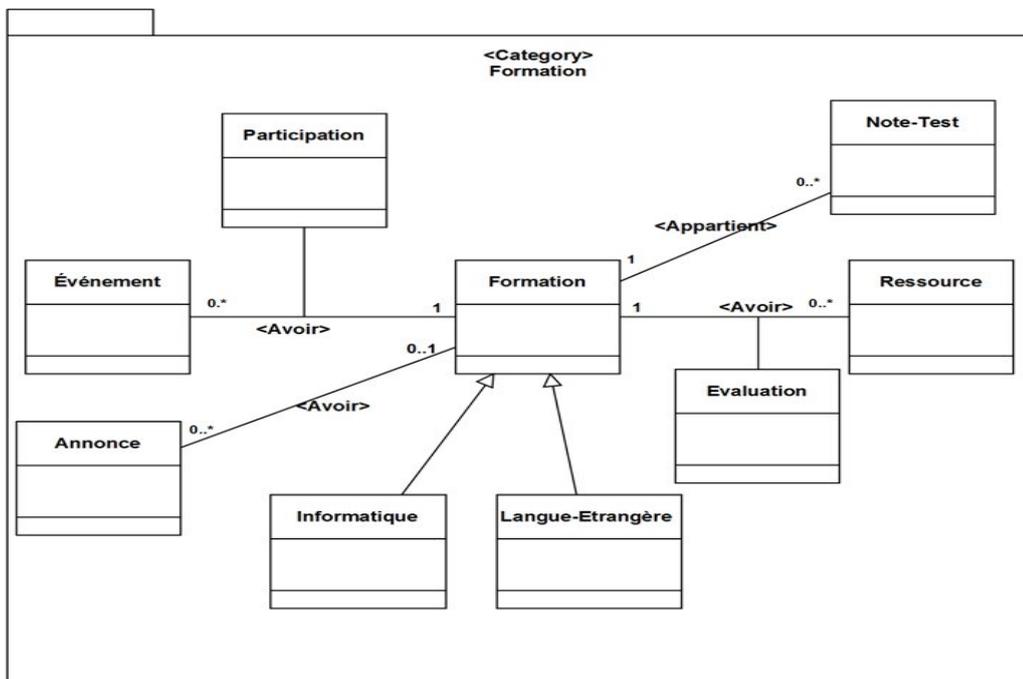


FIGURE 4.2 – Découpage des classes dans les catégories <Formation>.

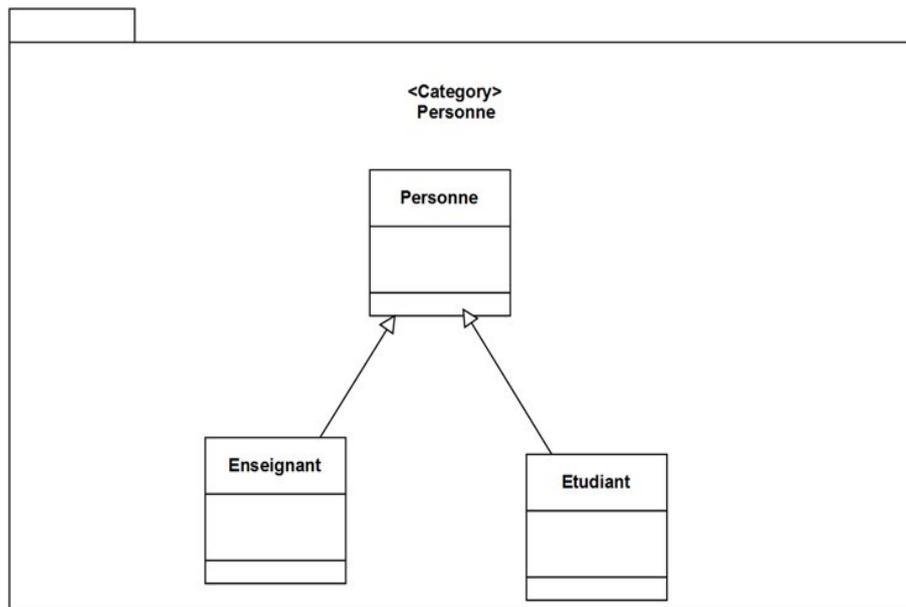


FIGURE 4.3 – Découpage des classes dans les catégories <Personne >.

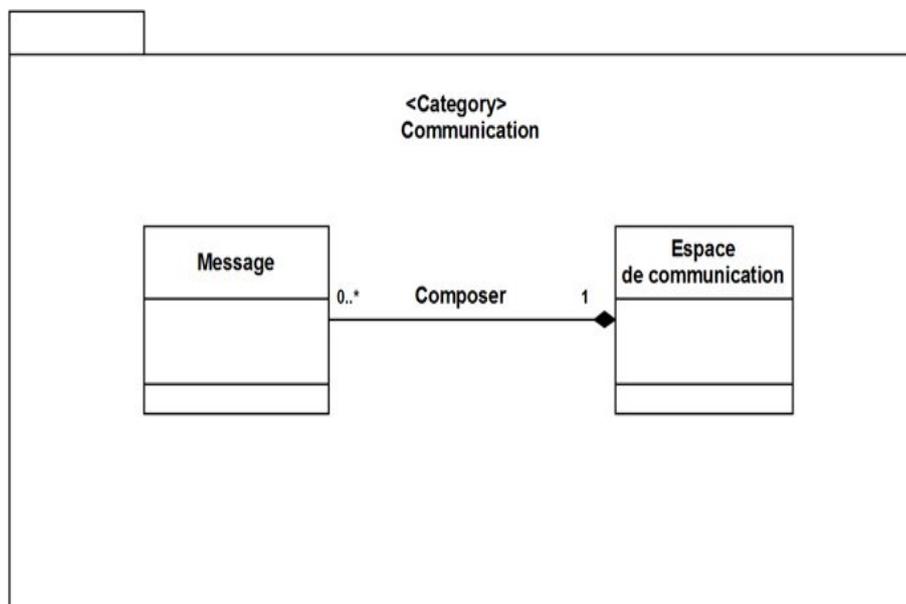


FIGURE 4.4 – Découpage des classes dans les catégories <Communication >.

4.2.1.4 Dépendance entre catégories

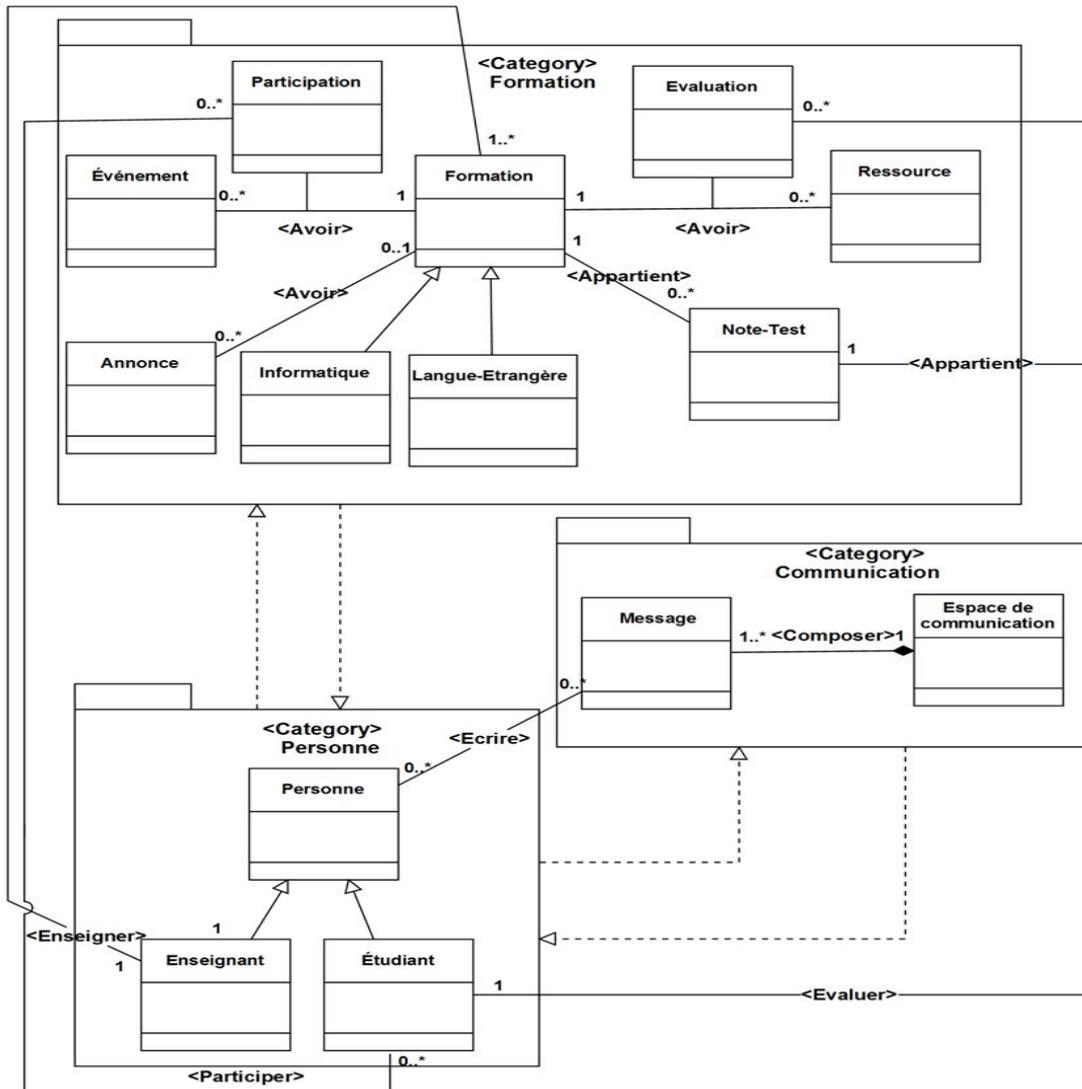


FIGURE 4.5 – Dépendances entre les catégories.

4.2.2 DEVELOPPEMENT DU MODELE STATIQUE

Le développement du modèle statique constitue la deuxième activité de l'étape d'analyse. Lors de cette étape, nous reprenons les diagrammes organisés lors du découpage en catégories afin de les affiner en leur ajoutant des attributs.

• Diagramme de classe de la catégorie « Formation »

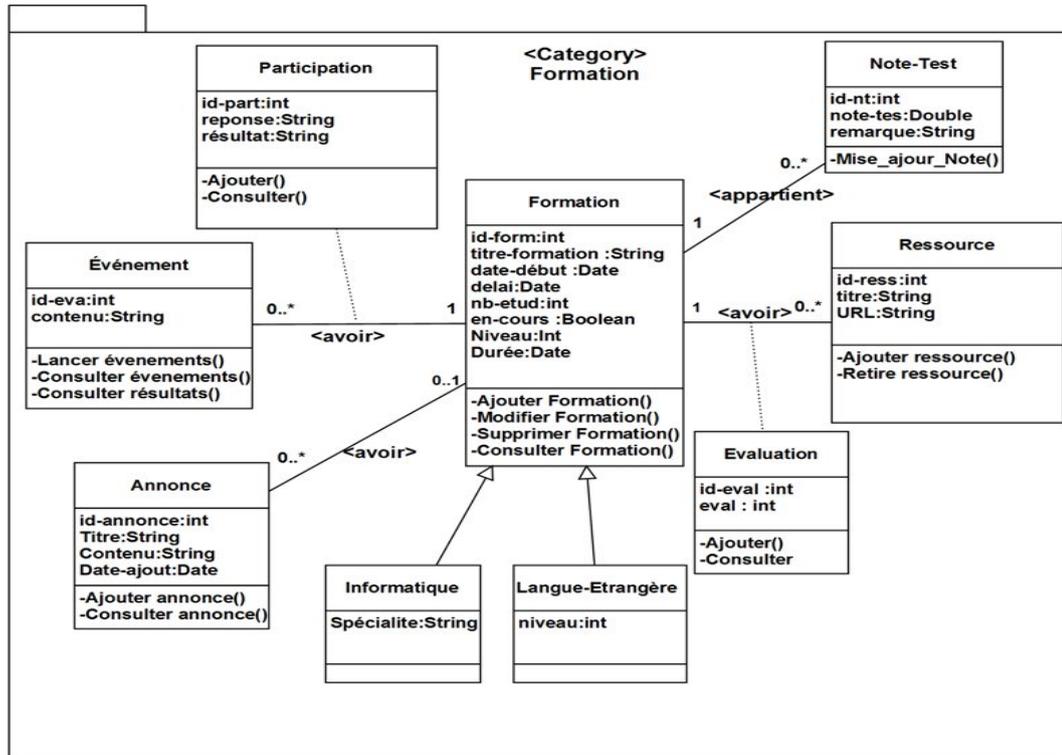


FIGURE 4.6 – Diagramme de classe de la catégorie « Formation ».

• Diagramme de classe de la catégorie « Personne »

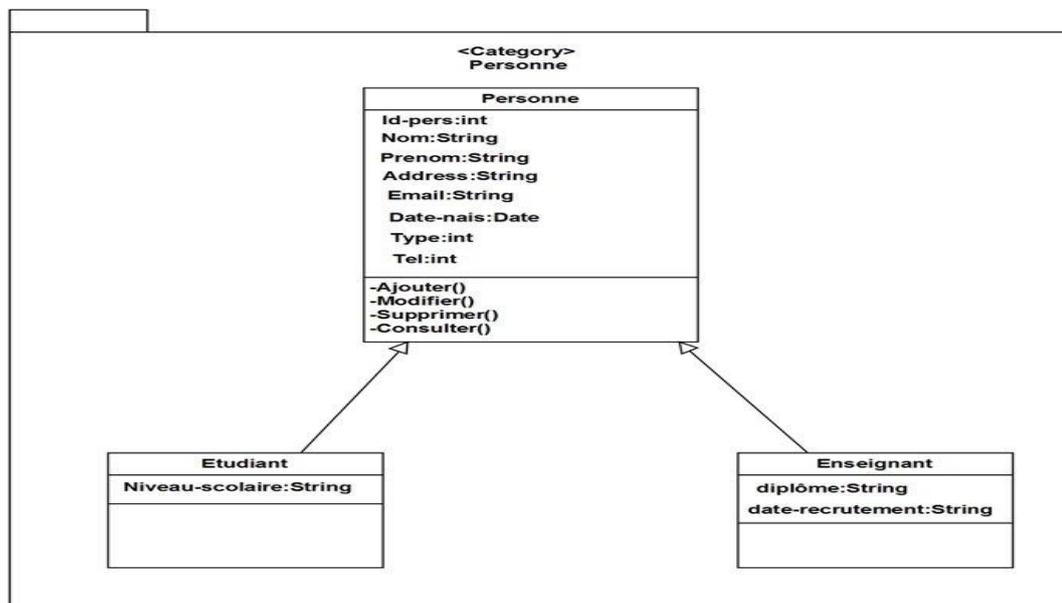


FIGURE 4.7 – Diagramme de classe de la catégorie « Personne ».

- **Diagramme de classe de la catégorie « Communication »**

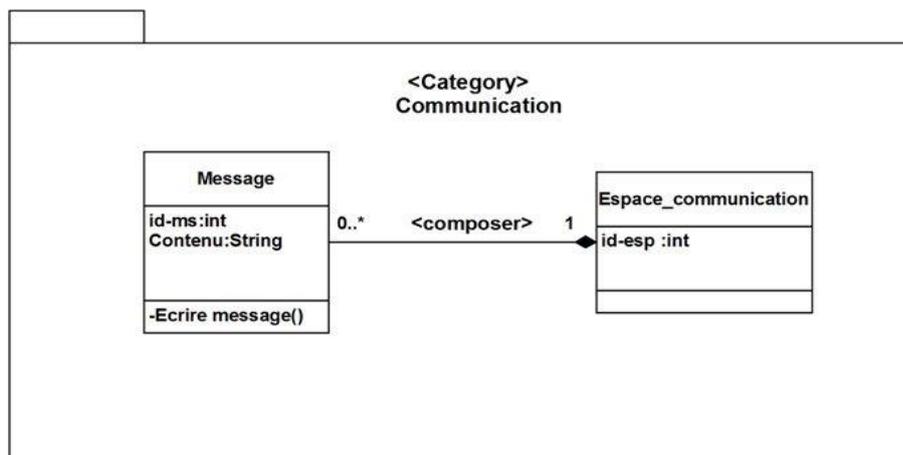


FIGURE 4.8 – Diagramme de classe de la catégorie «Communication».

4.3 PHASE DE CONCEPTION

Nous arrivons maintenant à la dernière phase de la modélisation avec UML après la modélisation des besoins puis l’organisation de la structure de la solution. La conception consiste à construire et à documenter précisément les classes et les tables. Dans cette partie nous allons effectuer une conception préliminaire et une conception détaillée.

4.3.1 CONCEPTION PRELIMINAIRE

4.3.1.1 Définition

La conception préliminaire est certainement l’étape la plus délicate du processus 2TUP car elle en représente le cœur. C’est en effet à cette occasion que s’effectue la fusion des études fonctionnelles et techniques. En conséquence, cette étape permet de passer de l’analyse objet à la conception, c’est-à-dire adapter la conception aux spécifications fournies par l’analyse et intégrer les fonctions métier et applicatives du système dans l’architecture technique[19].

4.3.1.2 Développement du modèle de déploiement

Dans cette étape on présente l’architecture physique supportant l’exploitation du système, le déploiement d’une solution client/serveur se construit sur la définition des postes de travail.

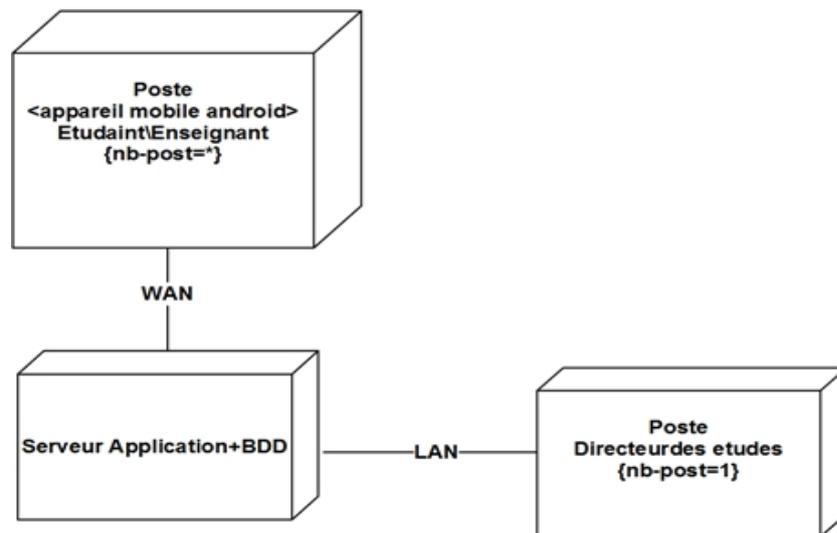


FIGURE 4.9 – Schéma du modèle de déploiement du système.

4.3.2 CONCEPTION DETAILLEE

4.3.2.1 Définition

L'étape de conception détaillée est l'avant dernière étape avant le codage. A cette étape, on commence à raffiner les choix de conception en incluant les spécificités du langage ou environnement de développement [18].

4.3.2.2 Passage du diagramme de classe au modèle relationnel.

Le passage du modèle conceptuel au modèle relationnel doit respecter les règles de transformation suivantes :

- **Règle 1** :(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.
- **Règle 2** :(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.
- **Règle 3** :(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.
- **Règle 4** :(cas héritage) transformer chaque sous classe en une relation, la clé primaire de la super classe devient clé primaire de chaque sous classe.

- **Règle 5** : (cas de composition) la clé primaire de la classe composée devient clé étrangère de la classe composant.

4.3.2.3 Tables 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. Les tables générées sont :

Classes	Attributs
Etudiant	(Id-pers ,niveau-scolaire, # Id-form).
Enseignant	(Id-pers , diplôme, date-recrutement, # Id-form).
Formation	(Id-form , titre-formation, date-début, type, délai, nb-etud,en-cours).
Informatique	(Id-form , spécialité).
Langue-Etrangère	(Id-form , niveau).
Ressource	(Id-Res , URL, titre , # Id-form).
Note-Test	(Id-nt , note-test, remarque, # Id-Etud, # Id-form).
Événement	(Id-eva , contenu, # Idform).
Evaluation	(Id-eval , eval, # id-Res, # Id-Etud).
Participation	(Id-part , réponse, résultat,# id-eva, # Id-Etud).
Annonce	(Id-ann , titre, contenu, Date-ajout).
Message	(Id-msg , contenu,# Id-esp).
Personne	(Id-pers ,Nom, prénom, email, Adresse, date-naissance, type,# Id-form).
Espace-communication	(Id-esp).

TABLE 4.1 – Table de la base de données.

4.3.2.4 Diagramme de classes détaillé

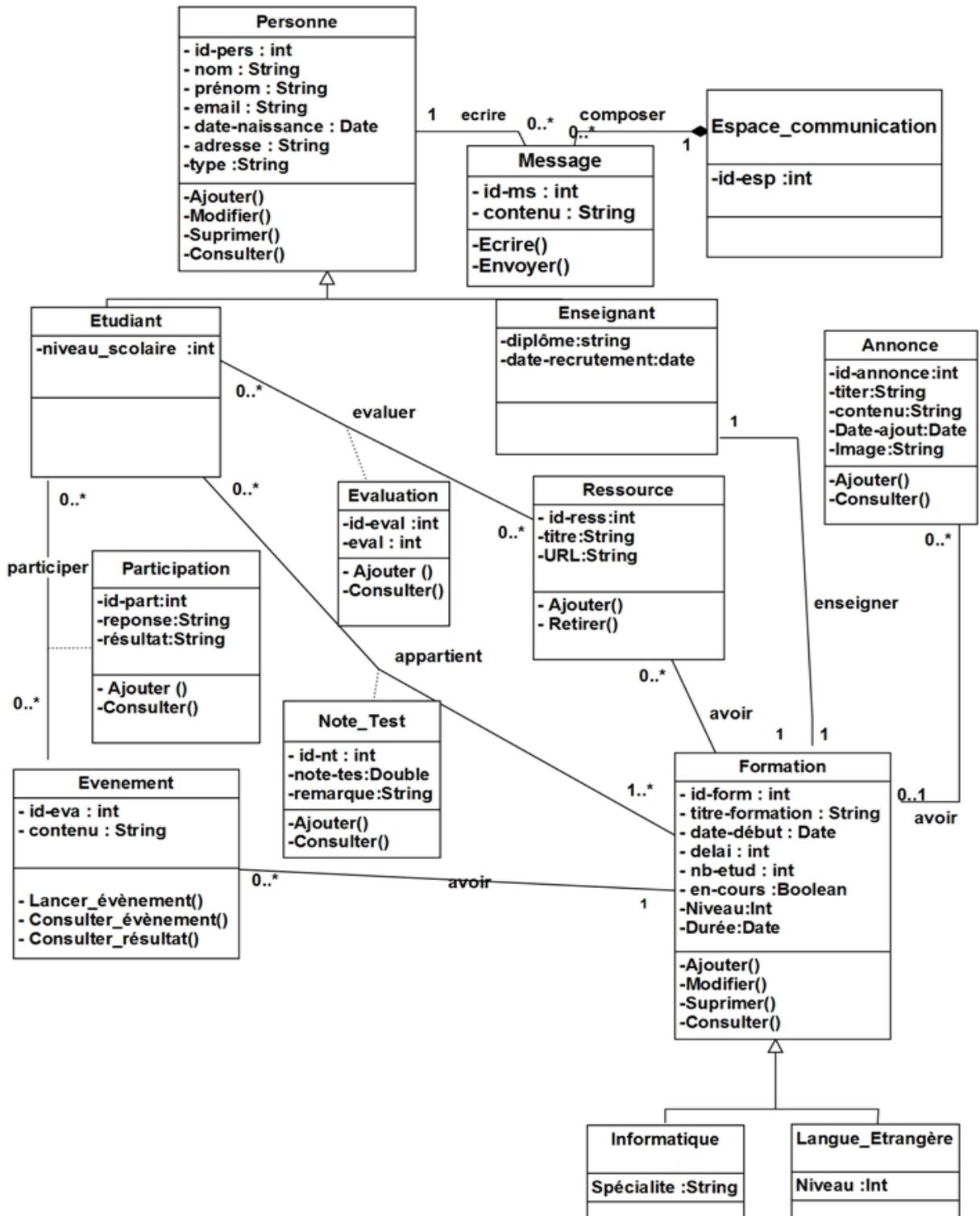


FIGURE 4.10 – Diagramme de classes détaillé de notre système.

4.4 CONCLUSION

Au cours de ce chapitre, nous avons présenté l'étape d'analyse qui nous a permis de passer d'une structuration fonctionnelle via les cas d'utilisations et les packages à une structuration objet via les classes et les catégories. En plus nous avons étudié la conception préliminaire et détaillée du système.

Dans le chapitre suivant, nous allons présenter la phase de réalisation de notre application.

Chapitre 5

Implémentation

5.1 INTRODUCTION

Cette partie concerne l'implémentation du projet où on convertit les diagrammes UML à un code source exécutable. Nous allons présenter les environnements de développement de l'application mobile et du site web tel que Laravel et Android Studio, les langages de programmation Java, SQL et PHP. Ensuite, nous allons présenter quelques fenêtres de notre application mobile et site web.

5.2 PRESENTATION DES OUTILS DE DEVELOPPEMENT

5.2.1 Langage de programmation java

Java est à la fois un langage de programmation et un environnement d'exécution. Le langage java a la particularité principale d'être portable sur plusieurs systèmes d'exploitation. Lors de la création du langage java, il avait été décidé que ce langage devait répondre aux 5 objectifs suivants : utiliser une méthode orientée objet, permettre à un même programme d'être exécuté sur plusieurs systèmes d'exploitation différents, pouvoir utiliser les réseaux informatiques, pouvoir exécuter du code distant de manière sûre, être facile à utiliser et posséder les points forts des langages de programmation orientés objet comme C++ [19].



FIGURE 5.1 – Logo java.

5.2.2 PHP

PHP (Hypertext PreProcessor) est un langage de programmation. Sa principale application se situe au niveau de la gestion des sites web dynamiques. On peut par exemple lui faire créer le contenu de pages HTML suivant différents paramètres : l'âge d'un visiteur, sa catégorie socioprofessionnelle, des mots-clés qu'il aura indiqués dans un moteur de recherche, des actualités du jour, etc. Les capacités de PHP ne s'arrêtent pas à la création de pages web. Il est aussi possible de manipuler des images, de créer des fichiers PDF, et de se connecter à des bases de données [21].

5.2.3 MySQL

Est un système de gestion de base de données. Son rôle est de stocker et de gérer une grande quantité de données en les organisant sous forme de tables, et de permettre la manipulation de ces données à travers le langage de requête SQL. On ne s'occupe plus alors de la manière dont les données sont stockées sur le disque dur, de simples instructions permettent d'ajouter, de supprimer, de mettre à jour et surtout de rechercher des données dans une base de données [19].

5.2.4 Xampp server

XAMPP(X(cross) Apache MariaDB Perl PHP) est un ensemble de logiciels permettant de mettre en place facilement un serveur Web et un serveur FTP. Il s'agit d'une distribution de logiciels libres offrant une bonne souplesse d'utilisation, réputée pour son installation simple et rapide. Ainsi, il est à la portée d'un grand nombre de personnes puisqu'il ne requiert pas de connaissances particulières et fonctionne, de plus, sur les systèmes d'exploitation les plus répandus [22].



FIGURE 5.2 – Logo Xampp.

5.2.5 Andriod studio

Android Studio est l'IDE officiel du système d'exploitation Android de Google, basé sur IntelliJ IDEA (Integrated Development Environment Android) et conçu spécifiquement pour le développement Android. En plus du puissant éditeur de code et développeur IntelliJ Android Studio offre encore plus de fonctionnalités qui améliorent la productivité lors de la création de logiciels Android des applications, telles que :

- Un système de construction flexible basé sur Gradle.
- Un émulateur rapide et riche en fonctionnalités.
- Un environnement unifié dans lequel vous pouvez développer pour tous les appareils Android.
- Instant Run pour appliquer des modifications à votre application en cours d'exécution sans créer de nouveau fichier APK.

- Modèles de code et intégration de GitHub pour vous aider à créer des fonctionnalités d'application courantes et à importer des exemples.
- Prise en charge intégrée de Google Cloud Platform, facilitant l'intégration de Google Cloud Messaging et App Engine [23].

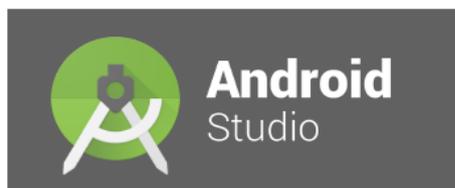


FIGURE 5.3 – Logo Android Studio.

5.2.6 Le SDK Android

C'est un ensemble d'outils permettant aux développeurs de créer des applications pour Android. Le SDK est disponible en téléchargement libre. Android SDK est un outil indispensable pour développer des applications Android. Un SDK (Software Development Kit) en général est un kit de développement ou trousse de développement logiciel. C'est un ensemble d'outils permettant aux développeurs de créer des applications de type défini (par exemple pour iOS, Android ou Symbian) [19].

5.2.7 Framework « Laravel »

C'est un Framework PHP open sources basé sur l'architecture MVC (Model View Controller). Sortie en juin 2011, il est actuellement l'un des Framework PHP les plus populaires [20].



FIGURE 5.4 – Logo Laravel.

5.2.8 Bootstrap

Bootstrap est un Framework développé par l'équipe du réseau social Twitter. Proposé en open source (sous licence MIT (Massachusetts Institute of Technology)), ce Framework utilisant les langages HTML, CSS et JavaScript fournit aux développeurs des outils pour créer

un site facilement. Ce Framework est pensé pour développer des sites avec un design responsive, qui s'adapte à tout type d'écran, et en priorité pour les Smartphones. Il fournit des outils avec des styles déjà en place pour des typographies, des boutons, des interfaces de navigation et bien d'autres encore. On appelle ce type de Framework un "Front-End Framework" [21].

5.2.9 Postman

Postman est un environnement complet de développement d'API : la seule solution de bout en bout avec collaboration, documentation, simulacres et surveillance. C'est la chaîne d'outils indispensable aux développeurs d'API. Nous avons utilisé Postman pour tester les différentes routes de l'API [23].



FIGURE 5.5 – Logo Postman.

5.2.10 Firebase

Firebase est un ensemble de services d'hébergement pour n'importe quel type d'application (Android, iOS, Java, PHP, C++ ...). Il propose d'héberger en NoSQL et en temps réel des bases de données.

Les applications qui utilisent Firebase intègrent une bibliothèque qui permet les diverses interactions possibles [24].



FIGURE 5.6 – Logo Firebase.

5.3 INTERFACES DE L'APPLICATION WEB

5.3.1 Fenêtre d'authentification

Cette fenêtre permet au directeur des études de s'authentifier.

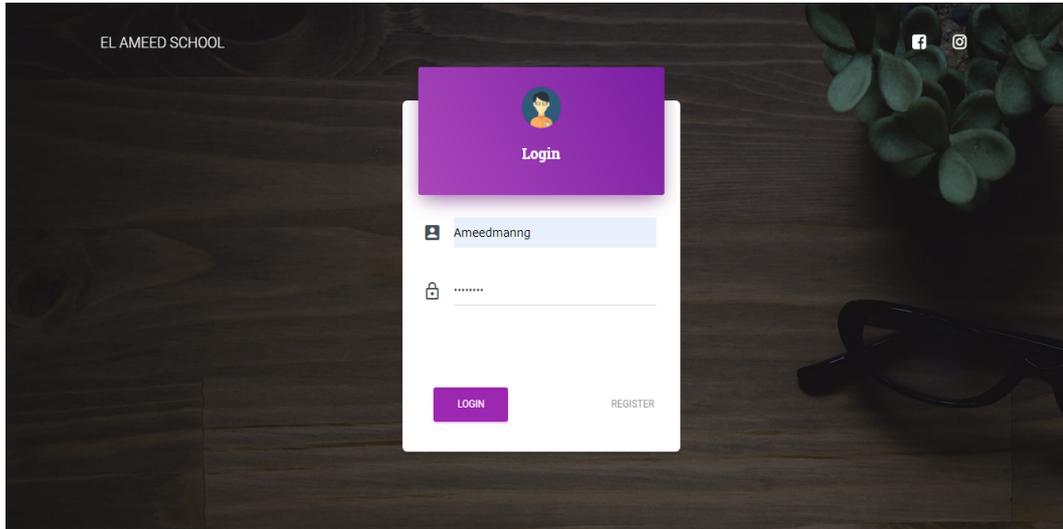


FIGURE 5.7 – Fenêtre d'authentification du directeur des études.

5.3.2 Fenêtre d'accueil

Cette fenêtre représente la fenêtre d'accueil de l'application web.

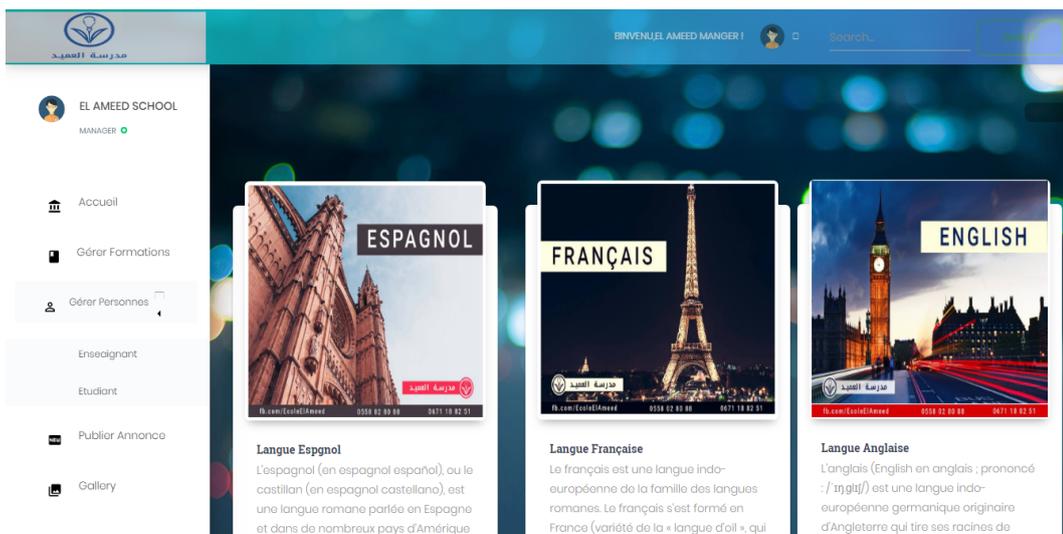


FIGURE 5.8 – Fenêtre d'accueil.

5.3.3 Fenêtre ajouter formation

Cette fenêtre permet au directeur des études d'ajouter une nouvelle formation.

FIGURE 5.9 – Fenêtre ajouter formation.

5.3.4 Fenêtre gère étudiants

profile	email	nom et prenom	Action
	hadjerbouanane001@gmail.com	hadjer bouanane	+ ✎ 🗑️
	rihab@gmail.com	Rihab bokhech	+ ✎ 🗑️
	hasna@gmail.com	hasna mehannoui	+ ✎ 🗑️
	sayfi@gmail.com	walid sayfi	+ ✎ 🗑️
	donya@gmail.com	Donya bochar	+ ✎ 🗑️
	adam@gmail.com	Adam hajazi	+ ✎ 🗑️

FIGURE 5.10 – Fenêtre gère étudiants.

5.3.5 Fenêtre ajouter étudiant

Cette fenêtre permet au directeur des études d'ajouter un étudiant à une formation.

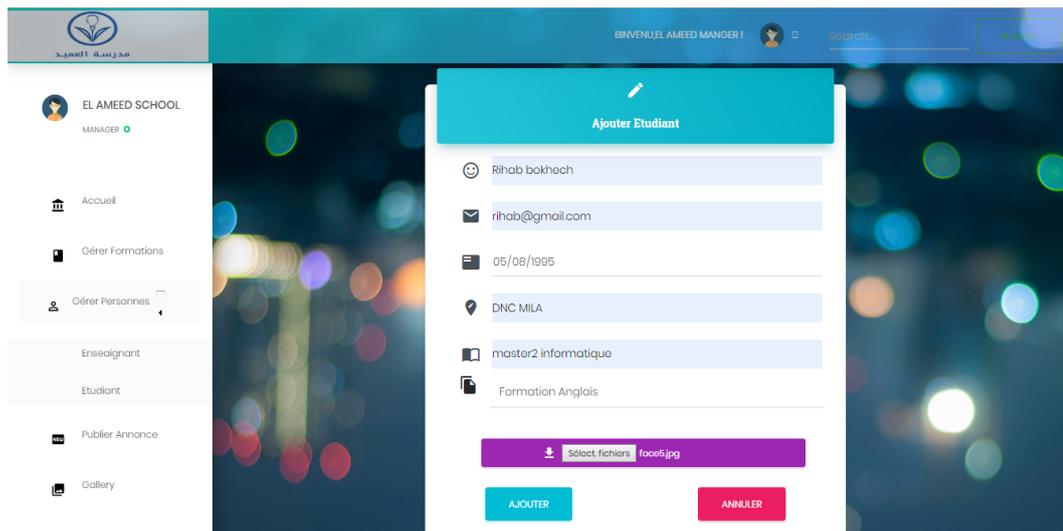


FIGURE 5.11 – Fenêtre ajouter étudiant.

5.3.6 Fenêtre modifier enseignant

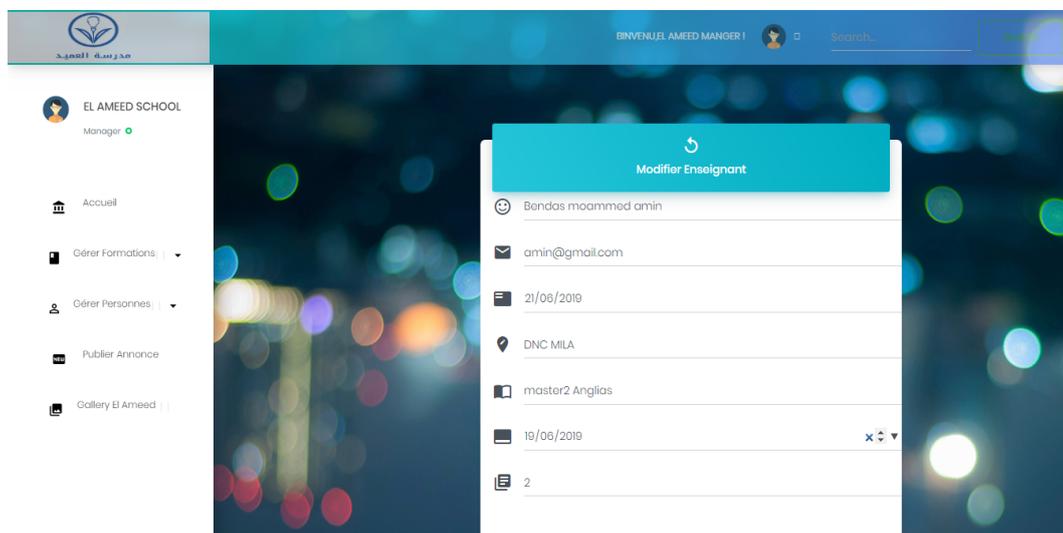


FIGURE 5.12 – Fenêtre de modification d'un enseignant.

5.4 INTERFACE D'APPLICATION MOBILE

5.4.1 Fenêtre d'accueil

Cette fenêtre représente la fenêtre d'accueil de l'application mobile.

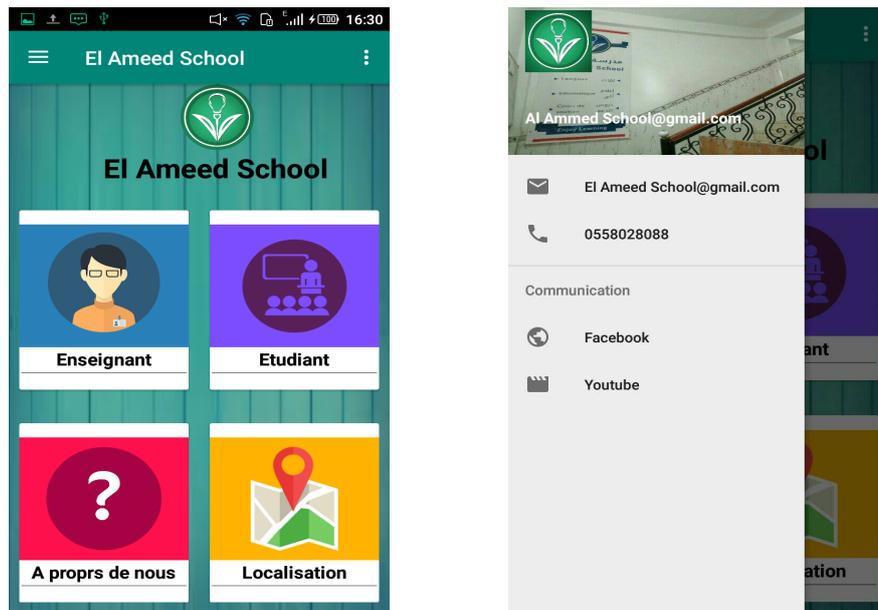


FIGURE 5.13 – Fenêtre d'accueil.

5.4.2 Fenetre GPS

Cette fenetre permet aux étudiants et enseignants de localiser l'école.



FIGURE 5.14 – Fenêtre GPS.

5.4.3 Fenêtre espace enseignant

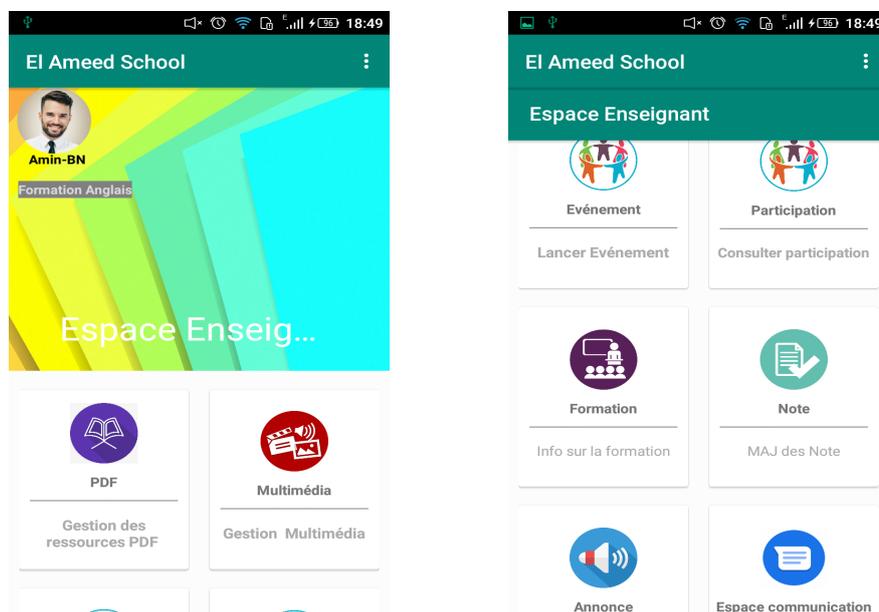


FIGURE 5.15 – Fenêtre espace enseignant.

5.4.4 Fenêtre espace étudiant

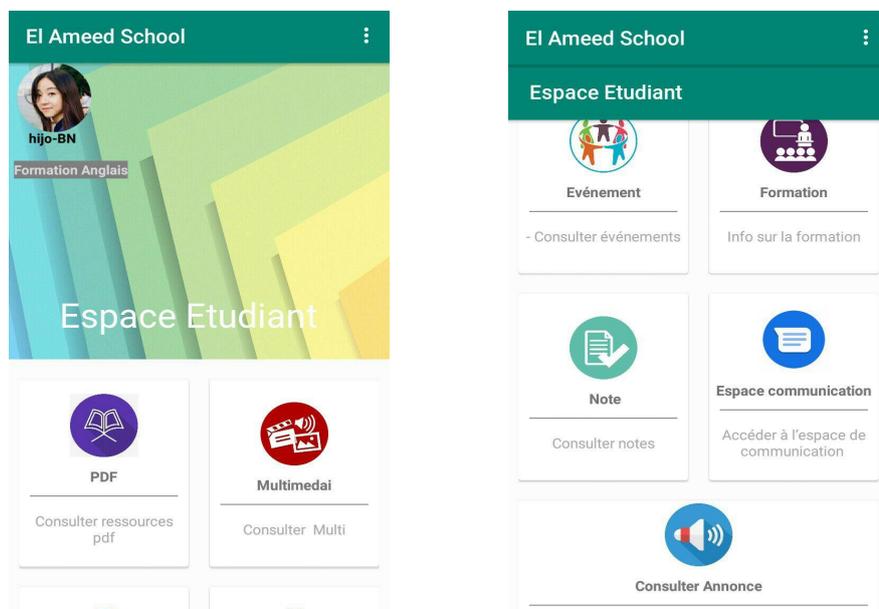


FIGURE 5.16 – Fenêtre espace étudiant.

5.4.5 Fenêtre lancer évènement

Cette fenêtre permet l'enseignant de lancer un évènement.

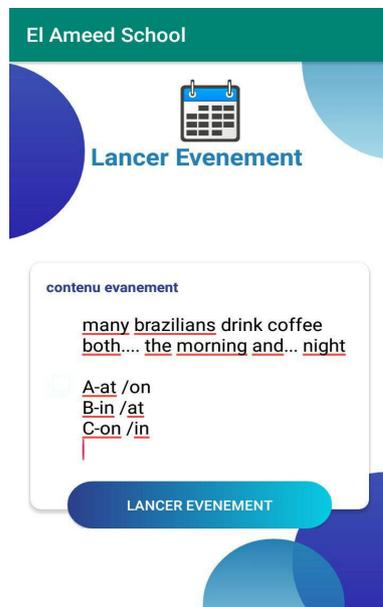


FIGURE 5.17 – Fenêtre lancer évènement.

5.4.6 Fenêtre consulter évènement

Cette fenêtre permet à l'étudiant de consulter le contenu d'un évènement.

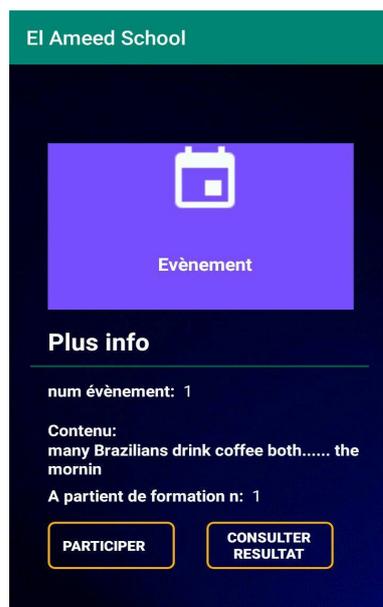


FIGURE 5.18 – Fenêtre de consultation d'un évènement.

5.4.7 Fenêtre consulter participations

Cette fenêtre permet à l'enseignant de consulter la liste de participations des étudiants à un évènement.



El Aameed School		
Liste de participations		
	hadjer bouanane A	num eva: 1
	Rihab bokhech B	num eva: 1
	Donya bochar C	num eva: 1
	walid sayfi C	num eva: 1
	hasna mehnnoui .	num eva: .

FIGURE 5.19 – Fenêtre de consultation de participations.

5.4.8 Fenêtre afficher résultat

Cette fenêtre permet à l'enseignant d'afficher le résultat d'un évènement.

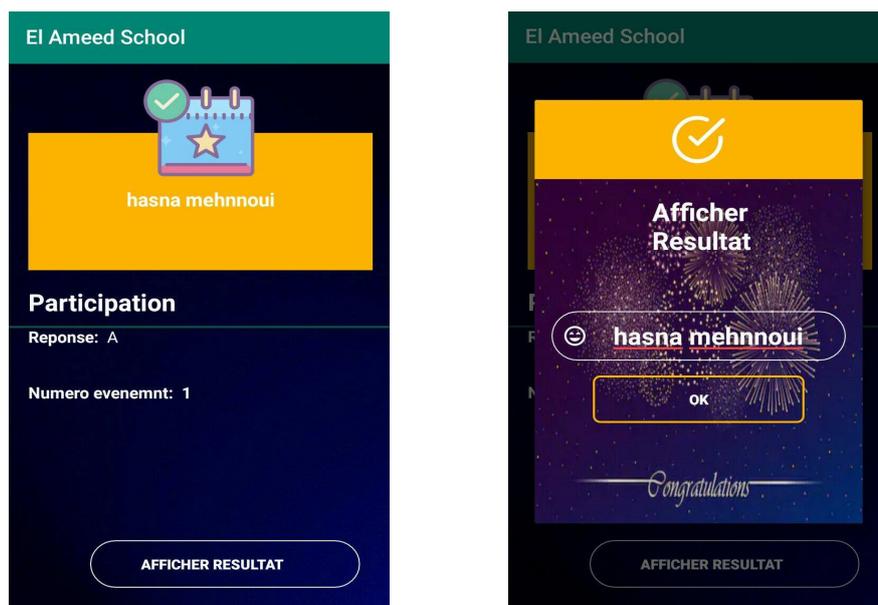


FIGURE 5.20 – Fenêtre afficher résultat.

5.4.9 Fenêtre consulter résultat

Cette fenêtre permet à l'étudiant de consulter le résultat d'un événement.

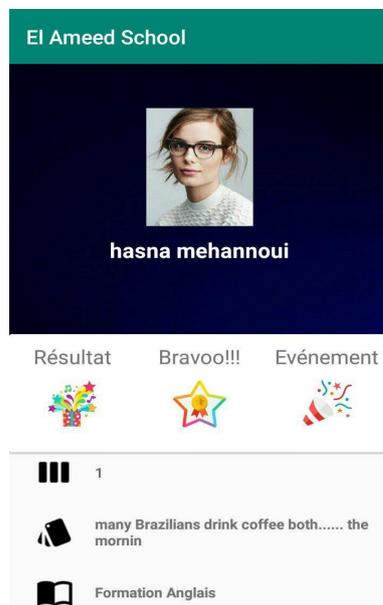


FIGURE 5.21 – Fenêtre consulter résultat.

5.4.10 Fenêtre gérer ressources PDF

Cette fenêtre permet à l'enseignant d'ajouter ou de supprimer une ressource PDF.

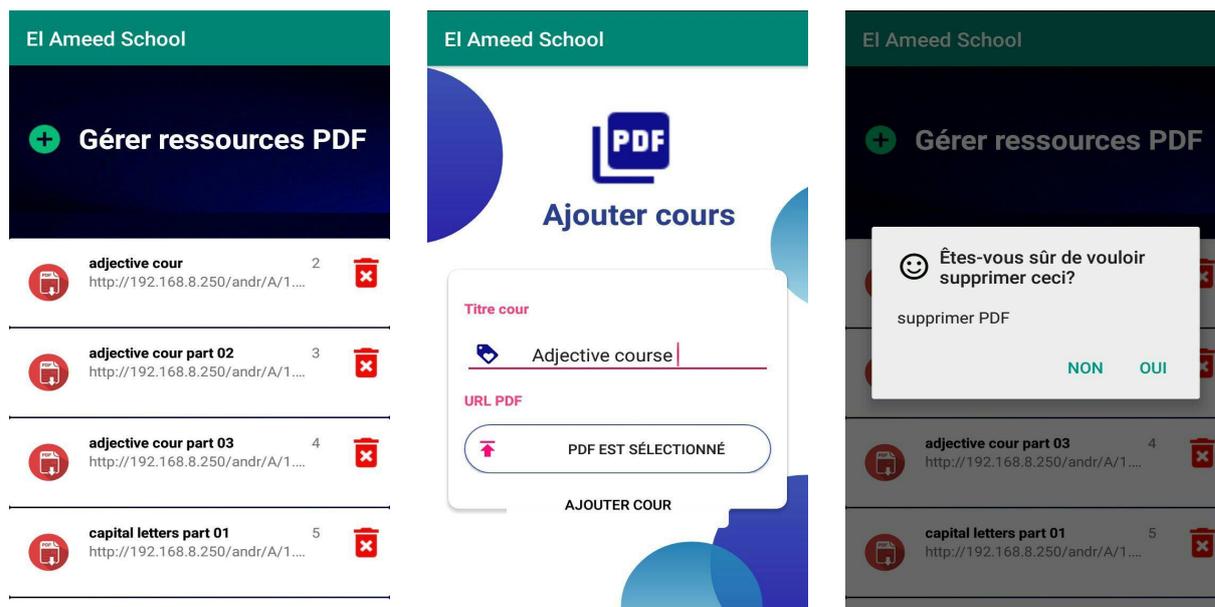


FIGURE 5.22 – Fenêtre gérer ressources PDF.

5.4.11 Fenêtre évaluer le pourcentage de compréhension

Cette fenêtre permet à l'étudiant d'évaluer le pourcentage de compréhension d'un cours.



FIGURE 5.23 – Fenêtre d'évaluation du pourcentage de compréhension.

5.4.12 Fenêtre consulter espace de communication



FIGURE 5.24 – Fenêtre consulter espace de communication.

5.5 CONCLUSION

Dans ce chapitre, nous avons présenté les outils et les logiciels utilisés pour le développement de notre application. Ensuite, nous avons présenté quelques interfaces de l'application. La phase d'implémentation, nous a permis de matérialiser les résultats des phases précédentes. Elle nous a permis également d'apprendre des langages et des outils nécessaires pour le développement des applications web et mobile.

CONCLUSION GENERALE

Le travail présenté dans ce mémoire de fin d'études se résume en la réalisation d'un système basé sur une application mobile sous Android et une application web pour l'école de formation « EL Ameed School ».

L'application mobile est adressée aux enseignants et étudiants de l'école. Elle leur permet de communiquer les différentes informations (cours, affichages, événements, participations, messages, etc.) d'une manière facile et efficace. L'application web est adressée au directeur des études de l'école, elle lui permet de gérer les informations utilisées par l'application mobile.

Pour le développement de notre application nous avons suivi le processus de développement 2TUP et le langage de modélisation UML. Sur le plan réalisation, nous avons choisi le langage Java sous la plateforme Android, le système de gestion de base de données MYSQL, et le langage PHP sous l'environnement Laravel.

La période passée au développement de notre application, nous a été d'un apport considérable. En effet, c'est une expérience qui nous a permis d'enrichir nos connaissances dans de domaines très variés comme : UML, 2TUP, Android, etc. D'autre part, ce projet a été une expérience réussie qui nous a permis de bénéficier d'un stage pratique qui a contribué à l'augmentation de nos connaissances dans le domaine de la recherche d'information.

Dans les futurs travaux, nous souhaitons rajouter de nouvelles fonctionnalités à notre système tel que : les statistiques coté administration, la gestion de l'historique, la compatibilité avec tous les types des dispositifs Android tel que : Smart Watch, Smart TV, etc.

Bibliographie

- [1] OGOR, Robert. *Modélisation avec UML* [en ligne] ENST BRETAGRE : 2003, 82p. Format PDF. Disponible sur : <<http://lamia.univ-ag.fr/~mperouma/telechargements/CoursENSTBr.pdf>> (Consulté le 26/01/2019).
- [2] MOHAMMED AMIN BENDAS, Rakib : *Applications Mobiles pour le suivi des élèves*. [en ligne]. Sciences et Technologies de l'Information et de la Communication (STIC), Centre Universitaire Abdelhafid Boussouf Mila, 2016/2017, 125p. Disponible sur : <<http://bib.centre-univ-mila.dz/index.php>> (Consulté le 26/01/2019).
- [3] OLIVER, Guibert. *Analyse et conception des systèmes informatiques*, LABRI. [en ligne]. Département Informatique de l'IUT de l'Université Bordeaux, 2007, 151p. Format PDF. Disponible sur : <<https://www.labri.fr/perso/guibert/DocumentsEnseignement/v2.pdf>> (Consulté le 26/01/2019).
- [4] GABAY, Joseph. *UML 2 Mise en oeuvre guidée avec études de cas analyse et conception*. [en ligne]. Paris, Dunod, 2008, 242p. Format PDF. Disponible sur : <<https://elearn.univ-ouargla.dz/2013-2014courses/.../UML2Analyseetconception.pdf?>> (consulté le 26/01/2019).
- [5] BOURAOUI NASSIM, ZOUAGHI RADOUANE. *Conception et réalisation d'une application de gestion des stagiaires au sein d'un centre de formation professionnelle*. [en ligne]. Sciences et Technologies de l'Information et de la Communication, Centre Universitaire Abdelhafid Boussouf Mila 2013/2014, 172p. Disponible sur : <<http://bib.centre-univ-mila.dz/index.php>> (Consulté le 26/01/2019).
- [6] LAURENT, DeBrauwere. FIENST, Vander Hyde. *UML 2 initialisation, exemples et exercices corrigés*. [en ligne]. 2eme édition : CNI, 2012, 150p. Disponible sur : <<https://livre.fnac.com/a4785634/Laurent-Debrauwere-UML-2>> (Consulté le 24/01/2019).
- [7] KAZI AOUELBASSIM, ROSTANE ZAKARIA. *Suivie des enseignements du LMD par application de la méthode 2TUP*. [en ligne], informatique industrielle, Université Abou

- Bekar Belkaid Tlemcen : Faculté des Sciences de l'ingénieur Département de l'informatique, 2007,114p. Disponible sur :<<ftp://developpez.com/.../rapport/Application%20de%20la%20méthodologie%20TUP>>(Consulté le 25/01/2019).
- [8] PASCAL Roques, FRANCK Vallée. *UML 2 en action de l'analyse des besoins à la conception* .[en ligne].4e édition.Paris : EYROLLES,février 2007 ,394p.PDF.Disponible sur :<<https://www.eyrolles.com/Informatique/Livre/uml-2-en-action-9782212121049/>>(Consulté le 27/01/2019).
- [9] *Etudier l'informatique mobile* [en ligne] .(modifié le 2019)Disponible sur :<<https://www.etudier.com/dissertations/l'Informatique-Mobile/391406.html>>(Consulté le 26/01/2019).
- [10] FERHAT YASSER ,LAAOUAR DJAAFAR.*Réalisation d'une application mobile de gestion de la vie estudiantine pour les Smartphones Android*. [en ligne]. Sciences et Technologies de l'Information et de la Communication (STIC) ,Centre Universitaire de Mila, 2012 / 2013 ,158p.Disponible sur :<<http://bib.centre-univ-mila.dz/index.php>>(Consulté le 27/01/2019)
- [11] Douas, bilal. *Développement des applications mobiles : la plateforme Android*. Mila : centre universitaire de Mila, 2016-2017,14p.
- [12] TECH SMARTPHONE. *Définition des smartphones* [en ligne] .Disponible sur :<<https://www.futura-sciences.com/tech/definitions/smartphone-smartphone1954/>> (Consulté le 25/01/2019).
- [13] FUTURA TECH. *Smartphones : la guerre des systèmes d'exploitation mobiles* [en ligne]. (Créé le 2007, modifié le 2019) Disponible sur :<<https://www.futura-sciences.com/tech/dossiers/telecoms-smartphones-guerre-systemes-exploitation-mobiles-1487/page/1/>>(Consulté le 25/01/2019).
- [14] M HAFFAF .*Chapitre 1 Applications et Systèmes d'exploitation mobiles* .[en ligne] ,2007 ,12p .Disponible sur :<<dspace.univtlemcen.dz/bitstream/112/8275/4/Chapitre-I.pdf>> (Consulté le 25/01/2019).
- [15] LEPTIDIGITALE. *Les parts de marché des systèmes d'exploitation mobiles iOS, Android et Windows en 2017*. [en ligne].(modifié le 2019)Disponible sur :<<https://www.leptidigital.fr/technologie/parts-demarche-systemes-exploitation-mobiles-ios-android-windows-12>> (consulté le 27/01/2019).

- [16] DOUAS, bilal. *Développement des applications mobiles : développement mobile sous Android*. Mila : centre universitaire de Mila, 2016-2017,72p.
- [17] MARIE, pascal Delamare. *Les différentes architectures client /serveur*. [en ligne]. Lycée Collège Raymond Poincaré Bar le Duc, 3p.Disponible sur :<mariepascal.delamare.free.fr/IMG/pdf/leClientServeur1.pdf >(Consulté le 31/05/2019).
- [18] Pascal ROQUES ,Frank VALLEE .« UML en action de l’analyse des besoins a la conception en java »,édition Eyroles,2000.
- [19] ZEMMOURI AMEL, BOULAICHE NASSIMA. *Conception et réalisation d’une application mobile pour la gestion des services bancaires*. [en ligne] .Stic.Mila : Centre Universitaire de Mila, 2015 / 2016 ,129 p. Disponible sur :<mixlr.com/bibliothèque-centre-univ-mila/>(consulté le 19/03/2019).
- [20] BOUKHCHEM, Nadhir. *Développement Web Avancé. Les frameworks de développement Web*. Mila : centre universitaire de Mila, 2017-2018,13p.
- [21] BENCHAOUI Safa ,BOULKELIA Sarra . *Développent d’une application web Pour la gestion locative au niveau de l’OPGI de Mila*. [en ligne] .Stic.Mila : Centre Universitaire de Mila, 2017 / 2018 ,150 p. Disponible sur :<mixlr.com/bibliothèque-centre-univ-mila/>(consulté le 27/03/2019).
- [22] Des Geeks et des lettres. XAMPP : plateforme pour héberger son propre site web . [en ligne].(mise à jour 2019) Disponible sur :<<https://desgeeksetdeslettres.com/programmation-java/xampp-plateforme-pour-heberger-son-propre-site-web>> (consulté le 27/03/2019).
- [23] BOUDAD Mohammed . *Study and development of a regular carpooling platform in Algeria*. [en ligne] .Stic.Mila : Centre Universitaire de Mila, 2017 / 2018 ,93 p. Disponible sur<mixlr.com/bibliothèque-centre-univ-mila/>(consulté le 27/05/2019).
- [24] WIKIPEDIA. [en ligne].(mise à jour 2019) Disponible sur :<<https://fr.wikipedia.org/wiki/Firebase> >(consulté le 27/05/2019).