

République Algérienne Démocratique et Populaire

Ministère de l'Enseignement Supérieur

Et de la Recherche Scientifique



Université Mohammed Sadik Ben Yahia –JIJEL

Faculté des Sciences exactes et d'informatique

Département d'informatique

*Mémoire de Fin d'Etudes en Vue de l'Obtention Du
Diplôme*

Master Académique En informatique

Option : réseaux et sécurité

Thème

*Réalisation d'une application mobile pour le
suivi des doctorants*

Réaliser par :

BOUHADICHE nesrine

Encadré par :

Dr. BOUREBIA soumia

Année Universitaire 2020/2021

Remerciement

Je tiens à remercier profondément ALLAH, le tout puissant de m'avoir donné le courage, la volonté, ainsi que la santé pour réaliser ce modeste travail.

Je tiens à exprimer mes vifs remerciements à Dr. BOUREBIA soumia, pour ses directives et ses orientations et sa disponibilité et pour la confiance qu'elle m'a accordée concernant mon projet de fin d'étude.

Je remercie les membres de jury qui ont fait l'honneur de participer au jugement de ce travail.

Je tiens à remercier toutes les personnes qui ont contribué au bon déroulement de ce projet de fin d'étude :

- A tous les enseignants de département d'informatique qui ont contribué à ma formation.*
- A mes parents.*
- A mes frères et sœurs.*
- A tous mes amis.*

Merci énormément...

Dédicace

Je dédie ce modeste travail à :

Mes très chers parents qui ont fait de moi ce que je suis aujourd'hui et qui ont veillé de guider mes pas durant toute ma vie par leurs aides, leurs grandes émotions, leurs sacrifices, et leur soutien et encouragement durant tous mes années d'apprentissage.

Mes très chers frères : Sabir et sa femme Amira ainsi mon frère Mouloud.

Mes très chères sœurs : Nezha, Wided, Mounia et Rahma.

Mes amis : Ines, Khaoula, Soumia, Ryma, Nabila et Meryama.

Mes collègues de la promotion.

Toute la promotion Informatique 2021.

Tous ceux qui j'aime et ceux que m'aiment.

Nesrine

Table des matières

Table des matières

Table des matières.....	i
Table des figures.....	v
Liste des tableaux.....	vii
Introduction générale.....	1
1 Notions de base	
1.1 Introduction	2
1.2 L'environnement universitaire en Algérie	2
1.2.1 Le système LMD	2
A. Licence.....	3
B. Master	3
C. Doctorat	4
1.3 Le système doctorat en Algérie	4
1.3.1 Inscription au doctorat	5
A. L'ancien Système.....	5
B. le Doctorat du LMD	5
1.3.2 Le Concours d'accès au doctorat.....	5
A. L'étude du dossier.....	5
B. Les épreuves écrites.....	5
1.3.3 Sujet et faisabilité de la thèse	5
1.3.4 Suivi de la Thèse entre le directeur, doctorant et le comité formation doctoral.....	6
A. Le Directeur de thèse.....	6
B. Le Doctorant	6
C. Le Comité de Formation Doctorale (CFD).....	7
1.3.5 Enseignement et encadrement	8
1.3.6 Protection du droit d'auteur.....	8
1.3.7 Thèse en cotutelle	9
1.4 Problématique.....	9
1.5 Travail à réaliser	9
1.6 Conclusion.....	10
2 Le développement mobile	
2.1 Introduction	11
2.2 Les terminaux mobiles	11
2.2.1 Définition.....	11
2.2.2 Types d'appareils mobiles.....	11

Table des matières

A. Smartphone	11
B. Tablette tactile	12
C. Ultraportable et miniportable :	13
2.3 Système exploitation mobile	13
2.3.1 Définition	13
2.3.2 Fonctionnalités d'un SE mobile.....	13
2.4 Le système d'exploitation Android	14
2.4.1 Présentation.....	14
2.4.2 Historique.....	14
2.4.3 Avantages et inconvénients d'Android.....	16
A. Avantages	16
B. Inconvénients	18
2.4.4 Architecture générale d'Android.....	19
a. Linux Kernel	19
b. Libraires	20
c. Android Runtime.....	20
d. Application Framework.....	20
e. Application.....	21
2.4.5 Outils de développement Android.....	21
A. le SDK Android.....	21
B. Les APIs	22
C. l'AVD Android	22
2.5 Les applications mobiles.....	23
2.5.1 définition.....	23
2.5.2 les types d'applications mobile.....	23
A. Application native	23
B. Application web	24
C. Application hybride	24
2.5.3 avantages et inconvénients d'un application mobile.....	24
A. Avantages.....	24
B. Inconvénients.....	25
2.5.4 Les critères indispensables d'une bonne application.....	25
2.5.5 Composants d'une application Android.....	27
2.5.6 Cycle de vie d'une activité.....	28
2.6 Conclusion.....	30
3 Présentation,conception et analyse	
3.1 Introduction	31

Table des matières

3.2 Présentation de l'application	31
3.2.1 Objectif.....	31
3.2.2 Apport de l'application sur le suivi des doctorants.....	32
3.2.3 Fonctionnement.....	32
3.2.4 Architecture.....	32
3.2.5 Les besoins fonctionnels.....	34
3.2.6 Les besoins non fonctionnels.....	34
3.3 Présentation d'UML	34
3.3.1 Définition.....	34
3.3.2 Les objectifs d'UML.....	35
3.3.3 Les diagrammes UML utilisés.....	35
a. Diagramme de cas d'utilisation	35
b. Diagramme de séquence	35
c. Diagramme de classe	36
3.4 Conception de l'application	37
3.4.1 Diagramme de cas d'utilisation	37
a. Identification des acteurs	37
b. Diagramme de cas d'utilisation de notre application	38
c. Description textuelle des cas d'utilisation.....	39
3.4.2 Diagrammes de séquences.....	45
3.4.3 Diagramme de classe.....	53
3.5 Conclusion.....	54
4 Implémentation et réalisation	
4.1 Introduction	55
4.2 Choix technique.....	55
4.2.1 Langage JAVA.....	55
4.2.2 XML.....	56
4.2.3 JSON.....	56
4.2.4 NoSQL.....	57
4.3 Environnement matériel.....	57
4.4 Environnement logiciel.....	58
4.4.1 Android studio.....	58
4.4.2 Android SDK.....	60
4.4.3 StarUML.....	60
4.5 Plateforme Firebase	61
4.5.1 Présentation.....	61
4.5.2 Les services Firebase utilisés.....	61

Table des matières

a. Firebase Realtime Database	61
b. Firebase Cloud Messaging (FCM) :	61
4.6 Bibliothèques utilisées	62
4.6.1 Jitsi meet Android SDK.....	62
4.6.2 Retrofit.....	62
4.6.3 QR generator.....	63
4.6.4 QR code scanner.....	63
4.7 Présentation de quelques interfaces de l'application	64
4.8 Les difficultés rencontrées	79
4.9 Conclusion.....	79
Conclusion générale.....	80

Table des figures

Figure 1.1: Architecture de système LMD.....	3
Figure 2.1: Exemple d'un smartphone.....	12
Figure 2.2 : Exemple d'une tablette tactile.....	12
Figure 2.3: Exemple d'un miniportable.....	13
Figure 2.4 : Architecture d'Android.....	19
Figure 2.5 : Application mobile.....	23
Figure 2.6 : Cycle de vie d'une activité Android.....	29
Figure 3.1: Architecture générale de notre application	33
Figure 3.2: Diagramme de cas d'utilisation.....	35
Figure 3.3 : Diagramme de séquence.....	36
Figure 3.4 : Diagramme de classe.....	36
Figure 3.5: Diagramme de CU globale de notre application.....	38
Figure 3.6 : Diagramme de séquence du CU « Créer compte »	45
Figure 3.7: Diagramme séquence du CU « S'authentifie »	46
Figure 3.8: Diagramme de séquence du CU « Utiliser messagerie »	47
Figure 3.9: Diagramme de séquence du CU « Gérer formation »	48
Figure 3.10: Diagramme de séquence du CU « Gérer réunion »	49
Figure 3.11: Diagramme de séquence du CU « Gérer liste des doctorants »	50
Figure 3.12: Diagramme de séquence du CU « Gérer publication »	51
Figure 3.13: Diagramme de séquence du CU « Consulter formation »	52
Figure 3.14: Diagramme de séquence du CU « Gérer QR code »	52
Figure 3.15 : Diagramme de classe de notre application.....	53
Figure 4.1: Fenêtre principale d'Android studio	59
Figure 4.2 : fenêtre bienvenue.....	64

Table des figures

Figure 4.3: Fenêtres inscription.....	65
Figure 4.4: Fenêtres connexion.....	66
Figure 4.5: Fenêtres l'envoi de QR code.....	67
Figure 4.6: Fenêtres réception, scan et confirmation.....	68
Figure 4.7: Fenêtres profils utilisateurs.....	69
Figure 4.8: Fenêtres ajoute des doctorants.....	70
Figure 4.9: Fenêtres suppression des doctorants.....	71
Figure 4.10: Fenêtres blocage des doctorants.....	72
Figure 4.11: Fenêtres messagerie.....	73
Figure 4.12: Fenêtres lancement de réunion.....	74
Figure 4.13: Fenêtres participation à la réunion.....	75
Figure 4.14: Fenêtres l'ajoute des formations.....	76
Figure 4.15: Fenêtres modification et suppression des formations.....	77
Figure 4.16: Fenêtres gérer publications.....	78
Figure 4.17: Fenêtres noter publications.....	79

Liste des tableaux

Liste des tableaux

Tableau 2.1: Comparaison entre les OS mobile.....	16
Tableau 2.2 : Les versions d'Android.....	18
Tableau 3.1 : Description textuelle de cas d'utilisation « créer compte »	39
Tableau 3.2: Description textuelle de cas d'utilisation « s'authentifie »	40
Tableau 3.3: Description textuelle de cas d'utilisation « participer au réunion »	40
Tableau 3.4: Description textuelle de cas d'utilisation « utiliser la messagerie »	40
Tableau 3.5: Description textuelle de cas d'utilisation « gérer formation et conférence ».....	41
Tableau 3.6: Description textuelle de cas d'utilisation « gérer réunion »	42
Tableau 3.7: Description textuelle de cas d'utilisation « gérer liste des doctorants »	42
Tableau 3.8: Description textuelle de cas d'utilisation « consulter progression »	43
Tableau 3.9 Description textuelle de cas d'utilisation « gérer publication »	43
Tableau 3.10: Description textuelle de cas d'utilisation « consulter formation »	44
Tableau 3. 11: Description textuelle de cas d'utilisation « générer code QR ».....	45

Introduction générale

L'université prend plus place dans l'espace numérique de jour en jour. Cette vision d'instauration ne se limite pas à l'utilisation des outils informatiques et la mise à disposition des praticiens les nouvelles technologies de communication et d'information mais, actuellement, l'université se crée une identité propre en faisant du numérique un cadre méthodologique pour ces pédagogies, et une orientation nouvelle pour faire face à des problèmes pointus.

Dans le cadre des réformes universitaires lancées au niveau d'Algérie, et dans un souci continu d'améliorer les méthodologies d'enseignement en favorisant l'accès aux nouvelles technologies de communication et d'information, nous avons mis en place une application mobile pour le suivi des doctorants qui offre aux doctorants et encadrants, la possibilité d'accéder à distance à une base de données, de partager les documents et participer aux forums.

Ce mémoire est organisé en quatre chapitres. Dans le premier chapitre, nous avons présenté une explication simplifiée du système doctoral en Algérie, en présentant brièvement la problématique et le travail à réaliser avec une organisation du manuscrit.

Dans le douzième chapitre, nous présentons les types d'appareils mobiles, les types d'applications mobiles et les différents systèmes d'exploitation mobiles et Android d'une façon Détaillée.

Dans le troisième chapitre nous exposons l'analyse, le détail et la conception de l'application ont se basant sur les diagrammes UML (diagramme de cas d'utilisation, diagramme de séquences et diagramme de classe).

Enfin et dans le quatrième chapitre, nous avons présenté l'application à travers une explication détaillée d'environnement de développement (matériel, logiciel, langages de programmation) et expliquer l'implémentation final d'application.

CHAPITRE 1

Notions de base

« C'est à force d'études et de travaux que l'homme réussit à distinguer qu'il a une âme différente de son corps ; et c'est à force de génie, la meilleure partie de lui-même, qu'il devient immortel par ses ouvrages. »

Pierre-Jules Stahl

1.1 Introduction

Le présent chapitre donne une vue générale sur le système universitaire en Algérie. Tout d'abord, le système LMD est expliqué. Ensuite, le système doctoral en Algérie (plus précisément le système doctoral au niveau de l'université de Jijel) est détaillé. Enfin, la problématique traitée et le travail réalisé dans ce mémoire sont exposés.

1.2 L'environnement universitaire en Algérie

A l'instar des pays de l'union européenne, les universités algériennes sont engagées, depuis l'année 2004 la mise en place d'une nouvelle architecture de l'enseignement supérieur conformément au système nommé LMD.

1.2.1 Le système LMD

Le LMD est un système d'enseignement supérieur préconisant une architecture des études en 3 cycles [2] :

- Un cycle qui couvre 6 semestres et qui mène au diplôme de Licence(L)
- Un cycle qui couvre 4 semestres et qui mène au diplôme Master(M)
- Un cycle de trois ans qui mène au diplôme Doctorat.

La figure 1 explique l'architecture du système LMD. Les diplômes et les formations obtenues sont mesurés en termes de crédit, plutôt qu'en années d'études. Trente crédits sont nécessaires pour accomplir un semestre (180 crédits pour accomplir Licence et 120 pour accomplir Master).

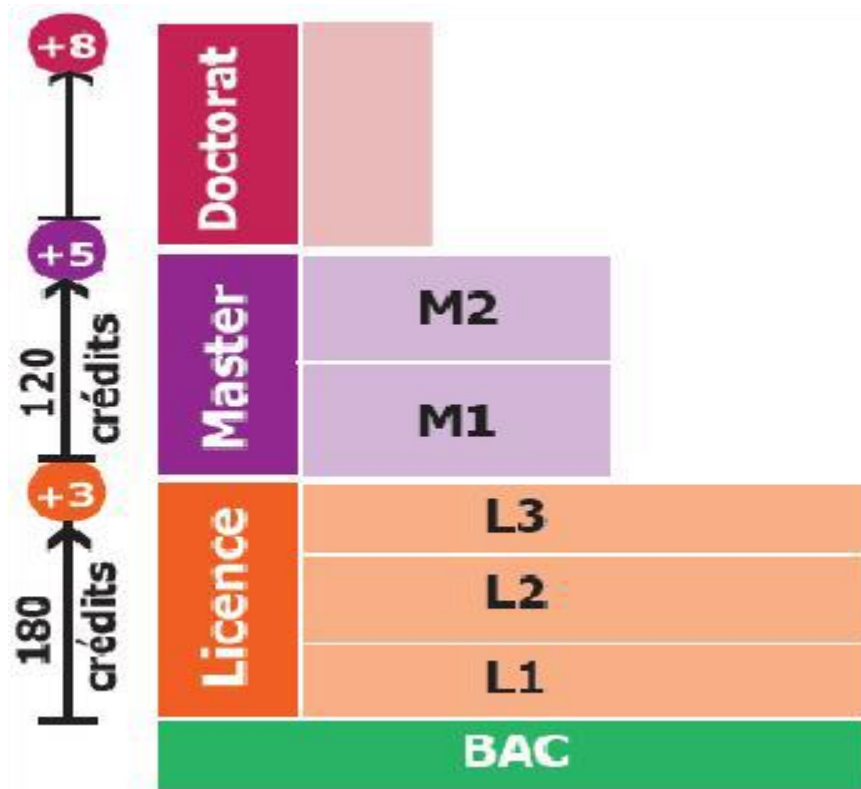


Figure 1.1 : Architecture de système LMD [1].

A. Licence

Le diplôme de licence est obtenu au bout de 6 semestres (3 ans). Il correspond à l'obtention de 180 crédits. La formation comprend des [2] :

- Enseignements théoriques.
- Enseignements méthodologiques.
- Enseignements professionnalisants.
- Enseignements transversaux (informatique, langues vivantes, culture...).
- Stages, Elle comprend aussi un dispositif d'accueil et de soutien (tutorat) destiné à aider l'étudiant à s'orienter et à réussir ses études.

B. Master

Après l'obtention d'une licence, le diplôme de master recherche ou professionnel, est obtenu au bout de 4 semestres (2 ans), qui correspondent à 120 crédits (soit 10 semestres et 300 crédits au total en comptant la licence). Le master, qui regroupe M1 et M2, sanctionne des parcours types comportant :

- Une voie à finalité professionnelle débouchant sur un master professionnel.
- Une voie à finalité recherche, débouchant sur un master recherche (académique).

Le master comprend [2] :

- Des enseignements théoriques.
- Des enseignements méthodologiques.
- Des enseignements professionnalisants.
- Des stages.
- Une initiation à la recherche.
- La rédaction d'un mémoire.

C. Doctorat

L'Université algérienne assure deux formations doctorales [2] :

- Doctorat science (ancien système) : L'accès à cette formation se fait après l'obtention du diplôme de magister ou d'un titre équivalent.
- Doctorat 3ème cycle (L.M.D) : L'accès à cette formation se fait par voie de concours.

1.3 Le système doctorat en Algérie

L'université algérienne assurera deux formations doctorales : celle de l'ancien système (Graduation-Magister-Doctorat) et celle du 3ème cycle du système LMD. Ce sont des formations par la recherche en vue de l'obtention du plus haut diplôme universitaire le doctorat [3].

- La formation doctorale est une formation à la recherche et par la recherche comportant un approfondissement des connaissances dans une discipline principale, une initiation aux techniques de raisonnement et d'expérimentation nécessaires dans les activités professionnelles et dans la recherche.
- Le doctorat permet l'acquisition de compétences scientifiques de haut niveau. Il correspond à la conduite d'un projet de recherche original et innovant. Sa préparation doit d'une part, s'inscrire dans le cadre des axes de recherches prioritaires nationales et d'autre part, être clairement définie dans ses buts comme dans ses exigences.
- La thèse de doctorat consiste en l'élaboration par le doctorant d'un travail de recherche original devant faire l'objet d'au moins une publication dans une revue scientifique reconnue.
- La période de la formation doctorale est aussi considérée comme une expérience professionnelle dans le secteur de la recherche et de l'innovation, au bout de laquelle le docteur est sensé avoir acquis non seulement des compétences scientifiques et techniques dans son thème de recherche mais aussi celles nécessaires à la gestion d'un projet en toute autonomie.
- Les modalités d'organisation de la formation doctorale sont fixées par voie réglementaire.

1.3.1 Inscription au doctorat

A. L'ancien Système

L'accès à la formation doctorale de l'ancien système se fait après l'obtention du diplôme de magister ou d'un titre équivalent. Le doctorant s'entend avec son encadreur sur le projet de thèse et le soumettent aux différents conseils scientifiques de l'établissement pour approbation.

B. le Doctorat du LMD

L'accès à la formation doctorale du système LMD se fait par voie de concours. Les modalités de son organisation et de sélection des candidats sont fixées par l'arrêtes n° 191 de 16 juillet 2012 modifiée et complétée par l'arrêtes n° 345 du 17 octobre 2012.

1.3.2 Le Concours d'accès au doctorat

Le concours s'effectue en deux étapes [3] :

A. L'étude du dossier

Cette étape permet d'apprécier le cursus universitaire du candidat et de procéder d'une première sélection en conformité avec les conditions d'accès définies par le comité de formation doctorale. Les dossiers de candidature jugés acceptables, seront classés en calculant pour chacun la moyenne pondérée.

B. Les épreuves écrites

Dans cette deuxième étape, chaque formation doctorale organise deux ou trois épreuves écrites d'une durée de 01h30 à 02 heures et qui se dérouleront en une seule journée. Elles porteront sur les spécialités de formation en master.

1.3.2 Sujet et faisabilité de la thèse

Le nom du directeur de thèse et le laboratoire d'accueil sont définis dans le dossier de candidature déposé auprès de l'administration. Le directeur de thèse et son laboratoire ou son équipe de recherche est responsables de la proposition du projet de recherche doctoral. Ils sont tenus d'assurer au projet :

- Un caractère original et formateur
- Sa réalisation dans les délais prévus par la réglementation.

Le choix du projet de recherche doctoral repose sur l'accord entre le doctorant et le directeur de thèse.

Le choix du sujet et les conditions de travail nécessaires à l'avancement de la recherche font l'objet, dans le respect de la présente charte, d'un accord librement conclu entre le doctorant et le directeur de thèse et est validé par le comité de formation doctorale.

Le sujet doit être soumis à l'agrément du conseil scientifique de l'établissement habilité qui en apprécie la conformité avec les axes de recherche prioritaires. Le sujet agréé fait l'objet d'un enregistrement sur le fichier central des thèses.

La durée de préparation du doctorat est fixée à trois années consécutives. Une prolongation d'une année à deux années supplémentaires peut être accordée par le chef d'établissement à titre exceptionnel sur proposition du conseil scientifique et après avis motivé du directeur de thèse et du comité de formation doctorale.

Pendant la durée de la thèse le doctorant non salarié bénéficie d'une bourse mensuelle dont le montant est fixé par la réglementation en vigueur.

1.3.4 Suivi de la thèse entre le directeur, doctorant et le comité formation doctoral

A. Le Directeur de thèse

Le directeur de thèse s'engage à consacrer le temps nécessaire pour l'encadrement scientifique des travaux de recherche du projet doctoral en fixant les périodes de rencontres de façon régulière et avec une fréquence adaptée. Il doit aussi définir et rassembler les moyens à mettre en œuvre pour permettre la réalisation du travail.

D'une autre part, le directeur de thèse s'assure de l'intégration du doctorant dans la structure de recherche et l'informe d'une part, de son droit d'accéder aux moyens techniques nécessaires à la recherche et aux ressources bibliographiques et d'autre part, de son devoir de participer aux formations spécifiques et complémentaires (indispensables à ses recherches ou à son devenir professionnel), aux séminaires, aux colloques et aux écoles d'été.

Le directeur de thèse doit veiller à ce que le doctorant fasse preuve d'initiative et gagne en autonomie tout au long du projet.

La codirection de thèse est possible lorsque le projet de recherche nécessite la coopération entre deux équipes (notamment thèses en cotutelles, thèse en entreprise, etc.) ou lorsqu'il fait appel aux compétences spécifiques de deux chercheurs d'une même équipe.

Enfin, le directeur de thèse donne son autorisation pour la soutenance. Le comité de formation doctorale propose les membres du jury [4].

B. Le Doctorant

Pendant toute la durée de l'élaboration de la thèse, le doctorant s'engage à :

- Consacrer le temps suffisant à l'avancement de ses travaux et à avoir des entretiens réguliers avec le directeur de thèse.
- Soumettre à son directeur de thèse, chaque année, un rapport sur l'état d'avancement de ses travaux avec les principales réalisations et de le présenter devant le comité de formation de 3ème cycle.
- S'intégrer dans la structure de recherche au sein de laquelle il effectue ces travaux de recherche et à respecter son règlement intérieur ainsi que les consignes de sécurité et d'hygiène.
- Faire un bon usage des moyens mis à sa disposition pour mener à bien ses recherches : les équipements scientifiques, les moyens informatiques, la documentation, l'accès à internet, etc.
- Assister à toutes les formations et activités programmées par le comité de formation doctorale et par le laboratoire de recherche d'accueil.
- Faire preuve d'une attitude d'autonomie, de responsabilité et de rigueur intellectuelle.
- Acquérir les compétences générales ou spécifiques que son directeur de thèse juge utiles.
- Respecter scrupuleusement les règles de confidentialité relatives aux méthodes, protocoles et résultats de sa recherche, et ne les communiquer ni oralement, ni par écrit, que sur autorisation de son directeur de thèse [4].

C. Le Comité de Formation Doctorale (CFD)

Chaque formation doctorale (FD) habilitée est gérée par son comité de formation doctorale. Ce comité est composé d'enseignants-chercheurs de rang magistral (appartenant à l'établissement habilité) ayant proposé l'ouverture de la formation et chargé de :

- Identifier les masters ouvrant droits à l'inscription au concours.
- Définir les conditions pédagogiques d'accès au concours permettant une présélection des candidatures.
- Procéder à l'étude des dossiers de candidature.
- Concevoir les épreuves écrites du concours.
- Veiller au respect des règles de l'anonymat dans l'organisation des épreuves du concours.
- Assurer l'organisation et le suivi du concours en coordination avec les services administratifs concernés, jusqu'à la proclamation des résultats.
- Assurer le suivi et l'évaluation des doctorants durant la formation.
- Se prononcer sur le sujet de recherche proposé par le directeur de thèse.

- Donner son avis sur la constitution du jury de soutenance de la thèse de doctorat et de proposer des rapporteurs.
- Initier toute forme de formation pour la recherche destinée aux doctorants (conférences, séminaires, ateliers).
- Organiser la mobilité des enseignants-chercheurs et des chercheurs intervenants dans la formation [4].

1.3.5 Procédures de médiation

En cas de litige entre le doctorant et le directeur de thèse et/ou le directeur du laboratoire, le comité de formation écoute les parties et propose un dénouement. En cas d'échec, il est fait appel aux structures scientifiques de la faculté [4].

1.3.6 Enseignement et encadrement

Le doctorant peut être appelé durant les deux premières années de son inscription à assurer des vacations de TP et/ou TD dans son domaine de compétence sous réserve de l'accord préalable de son directeur de thèse et dans la limite autorisée (3 h par semaine). Une demande d'autorisation devra parvenir à l'administration un mois avant le début des vacations, avec avis favorable du directeur de thèse et en précisant le nombre d'heures et le type de vacations prévus [4].

1.3.7 Protection du droit d'auteur

Droit de citation les travaux scientifiques élaborés par le doctorant dans le cadre de la thèse de doctorat appartiennent de droit à l'institution auprès de laquelle il s'est inscrit et a effectué ses recherches. Celle-ci pouvant en disposer librement, à moins qu'elle n'y renonce au profit du candidat. Le doctorant reconnaît être lié par une obligation de secret à l'égard des tiers et s'engage à maintenir la confidentialité sur toutes les informations et matériels, sous quelque forme que ce soit, dont il aura connaissance au cours de la réalisation de sa thèse de doctorat et à l'occasion de son séjour au laboratoire - éventuellement en liaison avec d'autres organismes ou sociétés-, tant que ces informations ne seront pas du domaine public. Le doctorant s'engage à ne pas effectuer de communications ou publications écrites ou orales sans recevoir l'autorisation préalable du directeur de thèse. En outre, le doctorant s'engage à informer son directeur de thèse des résultats qu'il obtiendra au cours de la réalisation de ses travaux de recherche et, en cas de résultats valorisables, le doctorant s'engage à en informer son laboratoire d'accueil et à respecter les droits de propriété intellectuelle attachés aux dits résultats. Le doctorant s'efforce d'assurer que ses travaux de recherche sont utiles à la société et ne

doit pas reproduire des recherches effectuées ailleurs ou précédemment. Il doit éviter tout type de plagiat et respecter le principe de la propriété intellectuelle et de la propriété conjointe des données en cas de recherche effectuée en collaboration avec un ou plusieurs directeurs de thèse et/ou d'autres chercheurs [4].

1.3.8 Thèse en cotutelle

Si le doctorat est effectué dans le cadre d'un partenariat, cotutelle, avec un autre établissement, les parties doivent se conformer aux dispositions particulières mentionnées dans la convention de partenariat [4].

1.4 Problématique

Au printemps 2020, le confinement provoqué par la COVID-19 a conduit à la fermeture des universités en Algérie, au maintien à domicile des acteurs universitaires et à la mise en place d'une continuité pédagogique. Dès lors et soudainement, de nombreux enseignants ont eu à maintenir leurs activités professionnelles via les technologies de l'information et de la communication (TIC) tout en gérant simultanément leur vie privée.

La majorité des doctorants ont déclaré que la qualité de leur encadrement a été affectée par la pandémie de COVID-19. La fréquence diminuée des réunions ainsi que la difficulté de communication et partage des documents importants rendent l'expérience doctorale dure. Il est important donc que la communauté scientifique tire les leçons de cette expérience, afin de fournir la meilleure supervision possible dans ces doctorants. Dans ce contexte, nous avons essayé d'apporter une solution concrète aux problèmes cités ci-dessus, en créant une application mobile pour le suivi des doctorants, et en se basant sur les technologies mobiles qui peuvent être bénéfiques pour l'éducation en termes de : la portabilité, interactivité entre tous les membres, connectivité, individualité

1.5 Travail à réaliser

Le mobile Learning (ou m-Learning, ou « apprentissage nomade ») implique la combinaison des Technologies de l'Information et de la Communication pour l'Enseignement (TICE) et de la technologie mobile des Smartphones. En clair, il s'agit d'apprendre ou de se former à partir de son Smartphone ou de sa tablette, n'importe où et n'importe quand, plutôt que d'être obligé d'aller dans une salle de cours ou dans un bureau.

Ces enseignements peuvent prendre la forme de simples textes, de messages, d'images ou d'infographies, ou encore de capsules vidéo.

L'apprentissage prend alors les formes les plus diverses : on peut utiliser les appareils portables pour accéder aux ressources éducatives, se connecter avec les autres ou créer du contenu, aussi bien à l'intérieur qu'à l'extérieur de la salle de classe.

Dans ce mémoire, nous essaierons de combiner entre le m-Learning et le système doctoral en Algérie à travers la création d'une application mobile pour le suivi des doctorants qui est une structure qui ambitionne de centraliser toute information relative à la recherche entre le doctorant, Directeur de thèse et l'encadrement

Cette collecte d'information ne peut s'effectuer que par la collaboration de tous les acteurs de ce secteur qu'ils soient le doctorant, directeur de thèse et encadreur

Un travail collaboratif de tous les instants permettra aux différents acteurs d'alimenter et de maintenir à jour cette banque de données, chacun à son niveau et selon ses privilèges, une vue d'ensemble claire et réelle sur l'état de l'art et d'avancement des projets de recherche.

Cette plateforme a l'ambition d'être un outil fédérateur, entre doctorant et son directeur de thèse et l'encadreur. Ils seront à ce titre dotés d'outils de recherche et de veille qui permettront de localiser un matériel dont ils ont besoin, de consulter les productions (cours, articles, conférences, ...) et de se tenir informés de leur progression, Chaque doctorant disposera ainsi d'un suivi personnalisé. Les outils statistiques donneront des retours d'information sur une multitude d'indicateurs, tels que les états d'avancement, la progression et la pertinence des différents axes de recherche. Et de manière générale à vivre et interagir dans la communauté de la recherche.

Conclusion

Dans ce chapitre, nous avons essayé de donner les informations de bases nécessaires pour comprendre le système de doctorat en Algérie. Nous avons exposé la problématique de ce mémoire et qui porte sur le suivi des doctorants à distance durant la période de Covid-19. Et nous avons expliqué le travail à réaliser dans ce mémoire. Le prochain chapitre donne une vue générale sur la technologie qui sera utilisé pour le développement de notre solution.

CHAPITRE 2

le développement mobile

« Ce ne sont ni les chefs d'entreprise, ni les milliardaires, ni les hommes politiques qui changent le monde. Ce sont les scientifiques, les savants, les techniciens. Dans le passage du XXe au XXIe siècle, ce sont eux qui sont à l'origine des deux inventions les plus marquantes : le téléphone portable et Internet. »

Michel Butor

2.1 Introduction

Avec l'évolution de la technologie mobile et la tendance actuelle à l'utiliser, le smartphone fait désormais partie intégrante dans notre vie quotidienne. Ce dernier a permis de communiquer avec nos proches, de suivre des cours ou des réunions à distance, de faire des courses ou du sport, de gérer ses besoins administratifs, etc. Tous sa grâce au développement d'applications mobiles qui a très vite submergé. Aujourd'hui, l'utilisation des applications mobiles ne cesse de croître et leur développement devient de plus en plus important.

Ce chapitre aura pour objectif de présenter quelques notions sur les mobiles, les types d'appareils mobiles existants, les différents types d'applications mobiles et leurs caractéristiques et les différents systèmes d'exploitation mobiles.

2.2 Les terminaux mobiles

2.2.1 Définition

Un terminal mobile est un appareil portatif de poids léger utilisable d'une manière autonome, donnant l'accès sans fil aux plusieurs réseaux et permettant le traitement et l'échange de données. Il existe une grande variété de dispositifs mobiles, la section suivante donne un aperçu de quelques types existants.

2.2.2 Types d'appareils mobiles

A. Smartphone

Un smartphone désigne un téléphone portable multifonctions qui a la capacité de naviguer sur Internet, lire des musiques et des films, équipé d'une puce GPS, d'un écran tactile qui peut évoluer avec le temps à l'aide de mises à jour, et qui a la capacité de télécharger et installer de nouvelles applications [5].



Figure 2.1 : *Exemple d'un smartphone [7].*

Un smartphone permet de :

- Communiquer : en envoyant et recevant des mails, et en utilisant les réseaux tels que Facebook.
- Gérez le temps : consultez vos rendez-vous à venir, gérez vos tâches, etc.
- Multimédia : prendre des photos, regarder des vidéos et écouter de la musique
- Internet : naviguer sur Internet comme sur ordinateur.
- GPS : la plupart des appareils sont équipés de GPS pour trouver facilement leur route.

B. Tablette tactile

Une tablette tactile est un ordinateur portable très léger et sans clavier, Le texte est entré l'aide d'un stylo, en touchant l'écran, ou par oral. La puissance de calcul d'une tablette est inférieure de celle d'un ordinateur de bureau [6].



Figure 2.2 : *Exemple d'une tablette tactile [8].*

C. Ultraportable et miniportable

Un ultraportable est un ordinateur portable de taille et de poids réduit. Il pèse moins de 2 kg, n'a pas de lecteur DVD, dispose d'une batterie plus petite et d'un processeur moins puissant [6].



Figure 2.3 : *Exemple d'un miniportable* [9].

2.3 Système d'exploitation mobile

2.3.1 Définition

Un système d'exploitation mobile C'un logiciel permettant à un terminal mobile de fonctionner. Il permet de ce fait aux utilisateurs de gérer la connectivité sans fil (Réseau téléphonique mobile, Wifi, Bluetooth, GPS,...), passer un appel téléphonique, télécharger des applications ou encore paramétrer et personnaliser leurs terminaux.

Vu que les systèmes d'exploitation mobiles sont conçus pour fonctionner sur des appareils mobiles de petite taille et possédant une autonomie réduite, ils possèdent une gestion avancée de l'énergie et une capacité de fonctionner avec des ressources limitées [10].

2.3.2 Fonctionnalités d'un SE mobile

Un SE mobile regroupe un ensemble de fonctionnalités, dont [11] :

- La gestion de mémoire.
- La gestion des processus et l'ordonnancement.
- La gestion du système de fichiers.
- La gestion des E/S.
- La gestion des fonctionnalités multimédia.

Le marché des OS mobiles est quasi monopolistique actuellement, avec Android qui équipe plus de 8 téléphones sur 10 dans le monde et iOS utilisé par les possesseurs d'iPhone. Chaque système d'exploitation dispose son propre « Store » boutique en ligne où les nouvelles applications peuvent être téléchargées.

Dans ce mémoire nous avons choisis Android comme OS pour notre application et la section suivante détaille ce système.

2.4 Le système d'exploitation Android

2.4.1 Présentation

Android est un système d'exploitation open source développé par l'OHA (Open Handset Alliance) édité par Google pour appareils embarqués et/ou mobiles, comme les smartphones ou les tablettes. Android est basé sur un fork du noyau de Linux. Les applications sont écrites en Java, et fonctionnent au sein d'une machine virtuelle Dalvik. Cette machine virtuelle a été modifiée pour être adapté aux appareils de faible puissance. Ce système permet à n'importe qui de lancer sa propre version d'Android. De nombreuses personnes utilisent ainsi des "ROM custom", c'est à dire des versions modifiées par rapport au code de base fourni par l'éditeur. Toutefois, le code source de la version 3.0 (Honeycomb), dédiée aux tablettes, n'a jamais été fourni [12].

2.4.2 Historique

L'histoire d'Android commence en octobre 2003, où la société Android Inc. est créée. Officiellement, elle développe des logiciels pour mobiles. Mais en réalité, elle se préparait à sortir un tout nouveau système d'exploitation pour smartphones. En 2005, Google rachète cette entreprise, et sort une première bêta en novembre 2007, avant de lancer la version 1.0 en septembre 2008 avec le HTC Dream. À partir de ce moment-là, le rythme des nouvelles sorties est très élevé : pas moins de 11 versions différentes sont sorties en 3 ans [17].

- ❖ Les versions d'Android ont des noms de dessert ou plus généralement de sucreries depuis la sortie de la version 1.5 (Cupcake) et suivent une logique alphabétique (de A à Z) .

Le tableau 2.2 résume les différentes versions d'Android depuis 2008 jusqu'à 2021

Nom	Numéro de Version	Date de sortie
Pas de nom	1.0	23 Septembre, 2008
	1.1	9 février, 2009
Cupcake	1.5	27Avril, 2009
Donut	1.6	15 Septembre, 2009
Eclair	2.0	27 Octobre, 2009
	2.0.1	3 Décembre, 2009
	2.1	11 janvier, 2010
Froyo	2.2 – 2.2.3	20 Mai, 2010
Gingerbread	2.3 – 2.3.2	6 Décembre, 2010
	2.3.3 - 2.3.7	9 février, 2011
Honeycomb	3.0	22 février, 2011
	3.1	10 Mai, 2011
	3.2 - 3.2.6	15 juillet, 2011
Ice Cream Sandwich	4.0 – 4.0.2	18 Octobre, 2011
	4.0.3 - 4.0.4	16 Décembre, 2011
Jelly Bean	4.1 – 4.1.2	9 Juillet, 2012
	4.2 - 4.2.2	13 Novembre, 2012
	4.3 - 4.3.1	24 juillet, 2013
KitKat	4.4 – 4.4.4	31 Octobre, 2013
	4.4W - 4.4W.2	25 Juin 25, 2014
Lollipop	5.0 – 5.0.2	4 Novembre, 2014
	5.1 - 5.1.1	2 Mars, 2015
Marshmallow	6.0 – 6.0.1	2 Octobre, 2015

Nougat	7.0	22 août, 2016
	7.1 - 7.1.2	4 Octobre, 2016
Oreo	8.0	21 août, 2017
	8.1	5 Décembre, 2017
Pie	9	6 août, 2018
Android 10	10	7 Septembre, 2019
Android 11	11	8 Septembre, 2020
Android 12	12	18 février 2021

Tableau 2.2 *Les versions d'Android [13].*

2.4.3 Avantages et inconvénients d'Android

A. Avantages

La plateforme Android comprend un système d'exploitation basé sur le noyau Linux, une interface graphique, un navigateur Web et des applications pour l'utilisateur final qui peuvent être téléchargées pour être principalement utilisé pour les appareils à écran tactile, les téléphones portables et les tablettes.

Voici quelques raisons incontestables de choisir Android pour notre application mobile (avantages) [14] :

– Un système exploitation Open Source

C'est l'un des plus grands avantages du système d'exploitation Android. Comme le code source est ouvert à tous, les développeurs et les fabricants d'appareils peuvent facilement accéder au code source et apporter les modifications nécessaires en fonction de leur compatibilité matérielle si nécessaire. Cela rend le système d'exploitation hautement personnalisable et orienté vers la recherche. Même Google peut prendre des suggestions et des commentaires des

développeurs, des testeurs et des fabricants d'appareils et utiliser ces entrées pour améliorer le système d'exploitation Android, comme par exemple SQLite ou hyper file SQL pour les bases de données et OpenGL pour la gestion d'images 2D et 3D (pour faire des jeux).

– **OS Kernel Robuste**

Android s'appuie sur le noyau Linux 2.6 pour les services système de base tels que la sécurité, la gestion de la mémoire et des processus, le réseau et la gestion des drivers, le noyau est multitâche et multi-utilisateur

– **Un service performant**

Le système d'exploitation Android offre la possibilité d'exécuter plusieurs applications en même temps. De sorte qu'on peut aisément exécuter de nombreuses tâches en même temps avec des appareils Android. Le concept App Store permet aux utilisateurs d'utiliser plus facilement des logiciels allant des jeux aux applications bureautiques sans se déplacer sur Internet. Sans oublier de noter l'avantageuse capacité de ce système à se synchroniser avec le téléphone dans les cas où les appareils peuvent être modifiés. Google Play est également capable de détecter le type de version du système d'exploitation Android exécuté par l'appareil, récupérant ainsi uniquement les applications compatibles avec l'appareil. Cela profite à l'utilisateur car il n'y a pas de perte de temps grâce à une recherche de marché généralisée.

– **Développement des applications facile et abortable**

Le développement d'applications est extrêmement facile et abordable pour Android, toutes les API mises à disposition facilitent et accélèrent grandement le travail. Ces APIs sont très complètes et très faciles d'accès. De manière un peu caricaturale, on peut dire que vous pouvez envoyer un SMS en seulement deux lignes de code par exemple.

– **Facile à vendre**

Le Play Store (anciennement Android Market) est une plateforme immense et très visitée, c'est donc une mine d'opportunités pour quiconque veut diffuser une application dessus.

– **Flexible**

Le système est extrêmement portable, il s'adapte à beaucoup de structures différentes. Les smartphones, les tablettes, la présence ou l'absence de clavier ou de trackball, différents processeurs... On trouve même des fours à micro-ondes qui fonctionnent à l'aide d'Android ! Non seulement c'est une immense chance d'avoir autant d'opportunités, mais en plus Android est construit de manière à faciliter le développement et la distribution en fonction des composants en présence dans le terminal.

– **Complémentaire**

L'architecture d'Android est inspirée par les applications composites, et encourage par ailleurs leur développement. Ces applications se trouvent essentiellement sur internet et leur principe est que vous pouvez combiner plusieurs composants totalement différents pour obtenir un résultat surpuissant. Par exemple, si on combine l'appareil photo avec le GPS, on peut poster les coordonnées GPS des photos prises.

B. Inconvénients

Le principal inconvénient d'un système d'exploitation Android c'est qu'elle ouvert aux logiciels malveillants un grand risque pour les systèmes d'exploitation open source, c'est bien, mais cela ouvre également la plateforme Android à toutes sortes d'applications non sécurisées. Vous pouvez télécharger une application susceptible de compromettre les fonctionnalités de votre appareil. Ou pire encore, d'ouvrir des échappatoires pour des activités malveillantes telles que l'exploration de données.

Le revers de la médaille est une plus grande difficulté pour les développeurs pour créer des applications et les adapter à une gamme très variée du système d'exploitation. Ainsi Les surcouches ajoutées par les constructeurs amplifient les problèmes et retardent les mises à jour.

Aussi les considérations des performances qui nécessitent des périphériques assez puissants et rapides enfin La difficulté d'intégration pour les vendeurs et sa forte dépendance de Google [14].

2.4.4 Architecture générale d'Android

Le système d'exploitation d'Android est composé de 5 couches. Le schéma suivant illustre cette architecture [15][16][42].

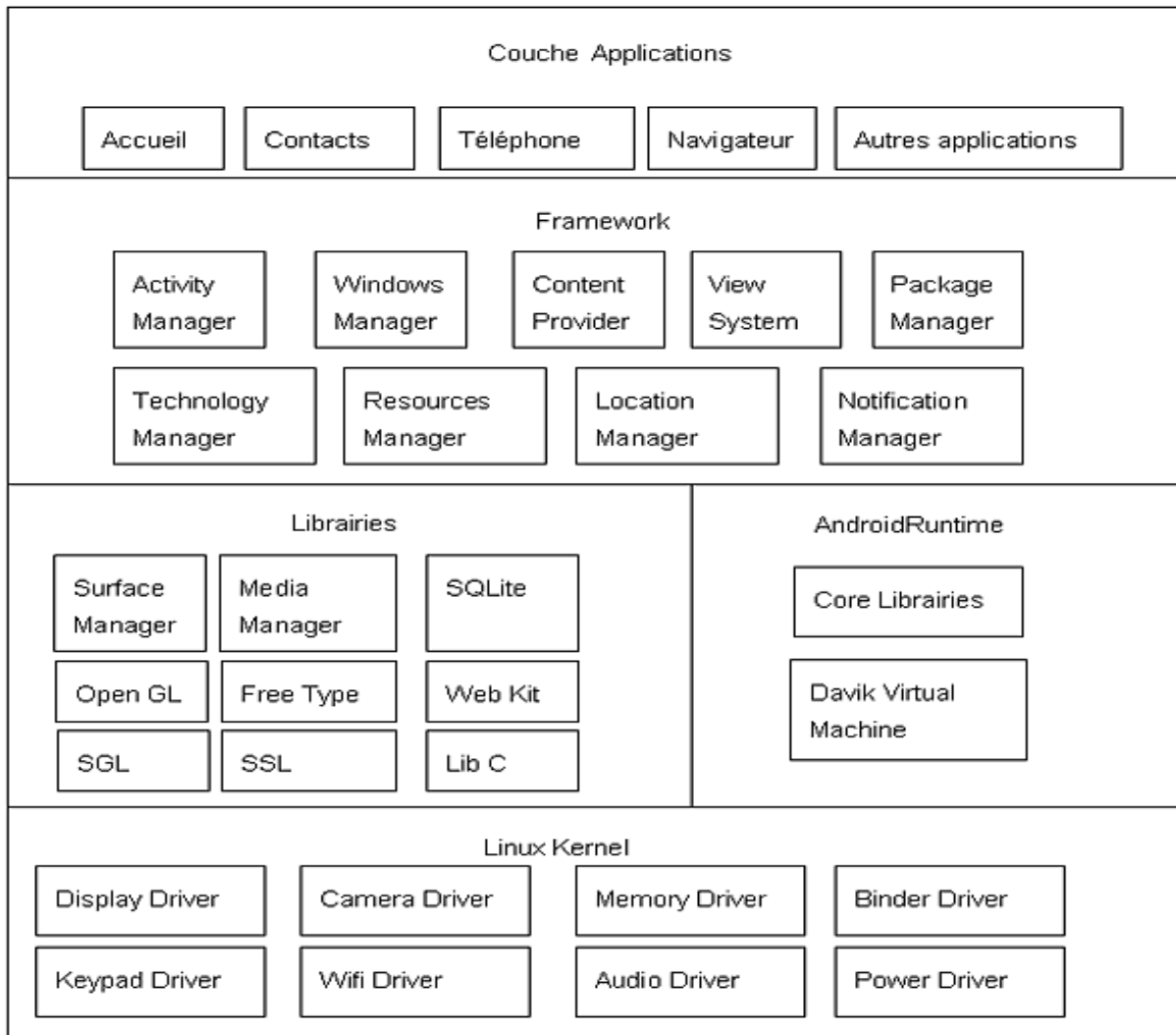


Figure 2.4 : Architecture d'Android [17].

a. Linux Kernel

Le noyau Linux modifié qui fonctionne comme HAL (Hardware Abstraction Layer ou la couche d'abstraction matérielle est une spécification et un utilitaire logiciel qui traque les

périphériques du système informatique), et fournit des pilotes de périphériques, gestion de la mémoire, gestion des processus, ainsi que des fonctionnalités de mise en réseau.

b. Libraires

La couche de bibliothèque est reliée par Java (qui s'écarte de la conception traditionnelle Linux). C'est dans cette couche que la librairie d'Android (Bionic) est située. Le gestionnaire s'occupe de la couche de l'interface utilisateur (UI) et des fenêtres.

- Surface Manager : affichage 2D/3D
- Media Framework : codecs basés sur librairie Open CORE
- SQLite : moteur de base de données relationnelle
- OpenGL|ES : implémentation OpenGL pour l'embarquer
- FreeType : rendu bitmap de polices vectorielles
- WebKit : moteur de rendu de pages Web
- SGL : moteur graphique 2D
- SSL : communications réseaux sécurisées
- Libc : dérivé de la librairie C (BSD) pour l'embarquer (Bionic libc).

c. Android Runtime

La couche d'exécution Android contient la machine virtuelle Dalvik et les bibliothèques de base (tels que Java ou IO). La plupart des fonctionnalités disponibles dans Android sont fournis par les bibliothèques de base. La VM Dalvik utilise des fonctionnalités principales de Linux comme la gestion de la mémoire et le multithreading, intrinsèque dans la langue Java. La VM Dalvik permet à toute application Android de fonctionner dans son propre processus, avec sa propre instance de la machine virtuelle Dalvik.

d. Application Framework

La couche Framework applicatif proposant des fonctionnalités de gestion de fenêtres, de téléphonie, de gestion de contenu. Le cadre Android comprend les services clés suivants :

- **Activity Manager** : Contrôle tous les aspects du cycle de vie de l'application et de la pile d'activité.

- **Fournisseurs de contenu** : Permet aux applications de publier et de partager des données avec d'autres applications.
- **Resource Manager** : Fournit l'accès à des ressources non intégrées au code telles que les chaînes, les paramètres de couleur et les mises en page de l'interface utilisateur.
- **Notifications Manager** : Permet aux applications d'afficher des alertes et des notifications à l'utilisateur.
- **View System** : Un jeu extensible de vues utilisé pour créer des interfaces utilisateur d'application.

e. Application

La couche application héberge l'interface API. Dans cette couche, le gestionnaire d'activités régit le cycle de vie des applications. Les fournisseurs de contenu permettent aux applications de soit accéder aux données depuis d'autres applications ou de partager leurs propres données. Le gestionnaire de ressources permet d'accéder à du code non-ressources (telles que des graphiques), tandis que le gestionnaire de notification permet aux applications d'afficher des alertes personnalisées. Au sommet de la couche application se trouvent respectivement les applications de base, ainsi que les applications de l'utilisateur.

Afin de développer des applications sous Android, un ensemble d'outils est nécessaire, la section suivante présente ses outils.

2.4.5 Outils de développement Android

Android est fourni avec un ensemble d'applications de base, dont un client de messagerie, le programme de SMS, calendrier, cartes, navigateur, contacts, et d'autres, Toutes les applications sont écrites en utilisant le langage de programmation Java [18].

A. le SDK Android

Les applications Android sont développées en Java, mais un appareil sous Android ne comprend pas le Java tel quel, il comprend une variante du Java adaptée pour Android. Un SDK, un kit de développement dans notre langue, est un ensemble d'outils permettant de développer

pour une cible particulière, Par exemple pour développer une console de jeu vidéo, on utilise un SDK spécifique pour développer des applications pour cette console. Le SDK Android est donc un ensemble d'outils que met à disposition Google afin de vous permettre de développer des applications pour Android.

B. Les APIs

Une API (Application Programming Interface) est un ensemble de définitions et de protocoles qui facilite la création et l'intégration de logiciels d'applications. Les API simplifient la façon dont les développeurs intègrent de nouveaux composants d'applications dans une architecture existante, elles facilitent la collaboration entre les équipes informatiques et métier.

Le SDK Android permet de télécharger les différentes versions de Google API pour intégrer des fonctionnalités liées aux services Google tels que : Google Maps, géolocalisation ou de la documentation au format Java Doc.

C. L'AVD Android

Un AVD (Android Virtual Device) est un émulateur de terminal sous Android, c'est-à-dire c'est un logiciel qui fait croire à la machine (ordinateur) qu'il est un appareil mobile sous Android.

Le SDK propose un émulateur Android. Il permet de lancer sur la machine du développeur un terminal virtuel représentant à l'écran un téléphone embarquant Android. C'est bien évidemment un outil indispensable pour le développement mobile, A chaque version d'Android est associée une version de l'émulateur, permettant au développeur de voir exactement à quoi ressemblera son application sur un matériel réel Rappelons cependant que l'émulateur ne propose pas toutes les fonctionnalités d'un vrai téléphone.

2.5 Les applications mobiles

2.5.1 définition

Une application mobile est un logiciel applicatif téléchargeable (gratuitement, payante ou mix avec des contenus ou fonctionnalités payantes) sur un appareil mobile (baladeur, smartphone, tablette tactile) via une plateforme de téléchargement adaptée au système d'exploitation (Google Play, App Store, Windows Store) [19].



Figure 2.5 : *Application mobile* [20].

2.5.2 les types d'applications mobile

Techniquement parlant, il y a trois types d'application mobile que tout utilisateur peut rencontrer [21] :

A. Application native

C'est une application conçue pour la plupart des systèmes d'exploitation qui sont fiables pour les smartphones, en référençant des langages propres à chaque système d'exploitation. Ce mode d'application n'est accessible que sur la plateforme d'application en fonction de sa particularité et de sa formule. Le développement d'applications natives doit s'appuyer sur la mémoire du smartphone sans manquer d'options liées au système d'exploitation. Le problème avec les applications natives est que les utilisateurs doivent disposer d'un système d'exploitation mobile spécifique pour les utiliser. Pour garantir une plus grande utilisation de ces applications mobiles,

veuillez envisager de lancer les mêmes applications mobiles compatibles avec n'importe quel système d'exploitation mobile.

B. Application web

Toute application conçue avec HTML et CSS qui est plus exploitable sur le navigateur Web d'un smartphone est appelée une application Web. Quelle que soit la marque de votre smartphone, vous pouvez accéder à l'application Web via un navigateur, aucun téléchargement n'est donc nécessaire. Sans tenir compte des différences persistantes entre les systèmes d'exploitation et les marques de smartphones, l'application web manque d'ergonomie, et elle n'utilise pas la mémoire du smartphone, ce qui est désavantagé par rapport aux applications natives.

C. Application hybride

C'est une application mobile qui combine les caractéristiques d'une application Web (développement HTML 5) et d'une application native. Ainsi, les applications mobiles seront accessibles sur toutes les plateformes applicatives. Ce type d'application mobile minimise le coût et la durée de son développement, même si cela nuit à l'amélioration et à la qualité de l'application native. Veuillez noter que les applications hybrides ne sont accessibles que sur iPhone et Android.

2.5.3 Avantages et inconvénients d'une application mobile

A. Avantages

- Une parfaite ergonomie est assurée pour les applications mobiles en comparaison aux sites mobiles cela encourage les utilisateurs à demeurer fidèles aux applis. En effet, le développement d'application mobile tient compte la taille du smartphone, le temps de chargement et autres paramètres.
- Les applications mobiles favorisent l'intégration des options de téléphone et ainsi, l'expérience utilisateur devient plus développée.
- Pas besoin d'avoir accès à l'internet pour que l'application fonctionne.

- Facile à trouver sur les stores par rapport aux sites mobiles, les applications mobiles ont connu ainsi un usage plus répandu auprès des jeunes surtout qu'elles notifient sur les événements en cours [22].

B. Inconvénients

- La soumission aux normes et règles édictées par les sociétés des plateformes mobiles à savoir Apple, Google, Windows et autres.
- Un investissement lourd pour le développement d'une application mobile adaptée à chaque système d'exploitation mobile contrairement au cout qu'exige le développement d'un site mobile.
- Lors de toute mise à jour d'application mobile, le mobinaute se trouve dans l'obligation de la faire à travers le store alors que le site mobile se met à jour d'une manière automatique [22].

2.5.4 Les critères indispensables d'une bonne application

Le développement d'applications pour appareils mobiles nécessite de prendre en compte les contraintes et les fonctionnalités de ces appareils. Les appareils mobiles fonctionnent sur batterie et ont des processeurs moins puissants que les ordinateurs personnels et ont également plus de fonctionnalités telles que la détection d'emplacement et les caméras. Les développeurs doivent également prendre en compte un large éventail de tailles d'écran, de spécifications matérielles et de configurations [23]

Les caractéristiques prendre en compte lors de la création d'une application [24] :

– Les notifications intelligentes

Les notifications intelligentes permettent d'anticiper les besoins des utilisateurs. L'envoi d'un nombre adapté de notifications peut impressionner et fidéliser l'utilisateur. Par contre, l'envoi de trop de notifications pousse l'utilisateur à désactiver les notifications.

– Tenir compte de l'emplacement

Si l'on demande à l'utilisateur où il habite, il doit recevoir des informations et des offres plus spécifiques en fonction de sa localisation, afin que la demande soit plus convaincante.

– Interaction

Pour commencer, c'est recommandé d'ajouter des options pour partager le contenu de l'application sur les réseaux sociaux ; les gens aiment partager, si cette option est retirée, ils peuvent aller chercher ce contenu ailleurs s'ils peuvent le partager avec le monde. Surtout dans les applications commerciales, interagir avec l'utilisateur pour créer un sentiment de confiance. Grâce à cette confiance l'utilisateur entrera beaucoup plus facilement ses coordonnées bancaires dans l'application pour réaliser un achat.

– Conception

Le design ne consiste pas seulement à choisir les couleurs et les formes, le travail du concepteur de l'application est complexe et va de l'expérience utilisateur à l'interaction, en passant par les animations, l'étude des chemins comportementaux ou la conception des boutons, entre autres choses. Il faut de connaître les aspects du design valoriser car le design est essentiel pour l'utilisateur.

– Section contact

En cas d'erreur ou de problème le développeur doit être disponible pour permettre à l'utilisateur de signaler cela et en créant une section spéciale dans l'application pour les questions fréquemment posées.

– L'adaptation aux différents systèmes d'exploitation

iOS et Android sont les deux systèmes par excellence et chacun a ses particularités doit tenir compte lors de la création d'une application. Tout d'abord, en développant le code de l'application, on verra que les systèmes de création des deux systèmes sont différents : le programme change, le langage de programmation change, etc. L'idéal est d'avoir des concepteurs experts dans chaque logiciel, puisqu'ils connaissent le style de chaque système afin que l'application ait une fluidité avec le même, à un niveau visuel et de performance.

– Enregistrement facile et pratique

Si l'application exige que les utilisateurs s'inscrivent pour l'utiliser, Il vaut mieux leur faciliter la vie et leur donner la possibilité de le faire avec un compte qu'ils ont déjà créé sur les réseaux sociaux tels que Facebook et Google. L'option de récupération du mot de passe doit être

présentée par e-mail. Tous ces facteurs doivent être pris en compte pour créer l'écran de connexion

– **Connectivité des appareils**

Assurez-vous que toutes les sessions peuvent être connectées et que ce qui se passe sur un appareil est mis à jour automatiquement sur un autre. Cette partie est vitale, ne créez pas d'applications pour différents appareils ; Personne ne l'utilisera.

2.5.5 Composants d'une application Android

Les composants de l'application sont des blocs de construction de base d'une application Android. Ces composants sont faiblement couplés par l'organisation du fichier manifeste d'application. Les quatre principales composantes sont [25] :

1. Les Activités

Une activité est une composante principale de l'application Android. Elle représente l'implémentation et les interactions des interfaces. Plusieurs choix se proposent pour mettre en place l'interface visuelle : utiliser un fichier XML pour manipuler l'interface, créer les éléments de l'interface à l'intérieur du code.

2. Les Services

Un service est une sorte d'activité, mais ne possédant pas d'interface visuelle. Celle-ci est donc lancée en fond, et permet par exemple de jouer de la musique d'effectuer une vérification des mails toutes les x minute.

3. Les fournisseurs de contenus « ContentProvider »

Comme l'indique le nom, un ContentProvider permet de partager les données d'une application peuvent être stockées dans une base SQLITE, dans des fichiers ou sur web. Le but étant de permettre à d'autres applications de requêter ces données.

4. Les Broadcast receivers

Un broadcast receiver est une composante qui réagit à un événement système (par exemple batterie faible, Le téléphone reçoit un SMS...). Les broadcast receivers ne possèdent pas

Il existe d'autres composants qui seront utilisés dans la construction des entités susmentionnées, leur logique et leur câblage. Ces composants sont :

- **Les fragments** : Représentent une partie de l'interface utilisateur dans une activité.
- **Les Views** : Les éléments UI qui sont dessinés à l'écran, y compris les boutons, les listes des formulaires, etc.
- **Les Layouts** : Afficher les hiérarchies qui contrôlent le format d'écran et l'apparence des vues.
- **Les intents** : connexion Messages entre les composants.
- **Les Resources** : Les éléments externes, tels que les chaînes, les constantes et les images étirables.
- **Manifest** : le fichier de configuration de l'application.

2.5.6 Cycle de vie d'une activité

Pour développer d'une application sur Android, on doit comprendre le cycle de vie d'une activité [25].

- **L'état Active/courant (Running)** : l'activité se trouve au premier plan, et reçoit les interactions utilisateurs. Si l'appareil a besoin des ressources, l'activité se trouvant en bas du back est tuée.
- **L'état Paused (il est en pause)** : L'activité est visible mais l'utilisateur ne peut pas interagir avec. La seule différence avec l'état précédent est la non-réception des événements utilisateurs.
- **L'état Stopped** : L'activité n'est pas visible mais toujours en cours d'exécution. Toutes les informations relatives à son exécution sont conservées en mémoire.
- **L'état Dead** : l'activité est tuée, elle n'est plus en cours d'exécution, et disparaît de la back stack.
- **Back stack** : toutes les activités sont stockées dans une liste que l'on appelle généralement back stack.

Il existe trois boucles principales :

Lors de la création d'activité, la méthode **onCreate** est appelée. Cet appel est suivi par la méthode **onStart** afin de signifier le lancement effectif de l'application. Puis la méthode **onResume** est appelée afin d'exécuter tous les traitements nécessaires au fonctionnement de l'activité. Ces traitements devront être arrêtés lors d'appel à la méthode **onPause** est relancés si besoin lors d'un futur appel à la méthode **onResume**.

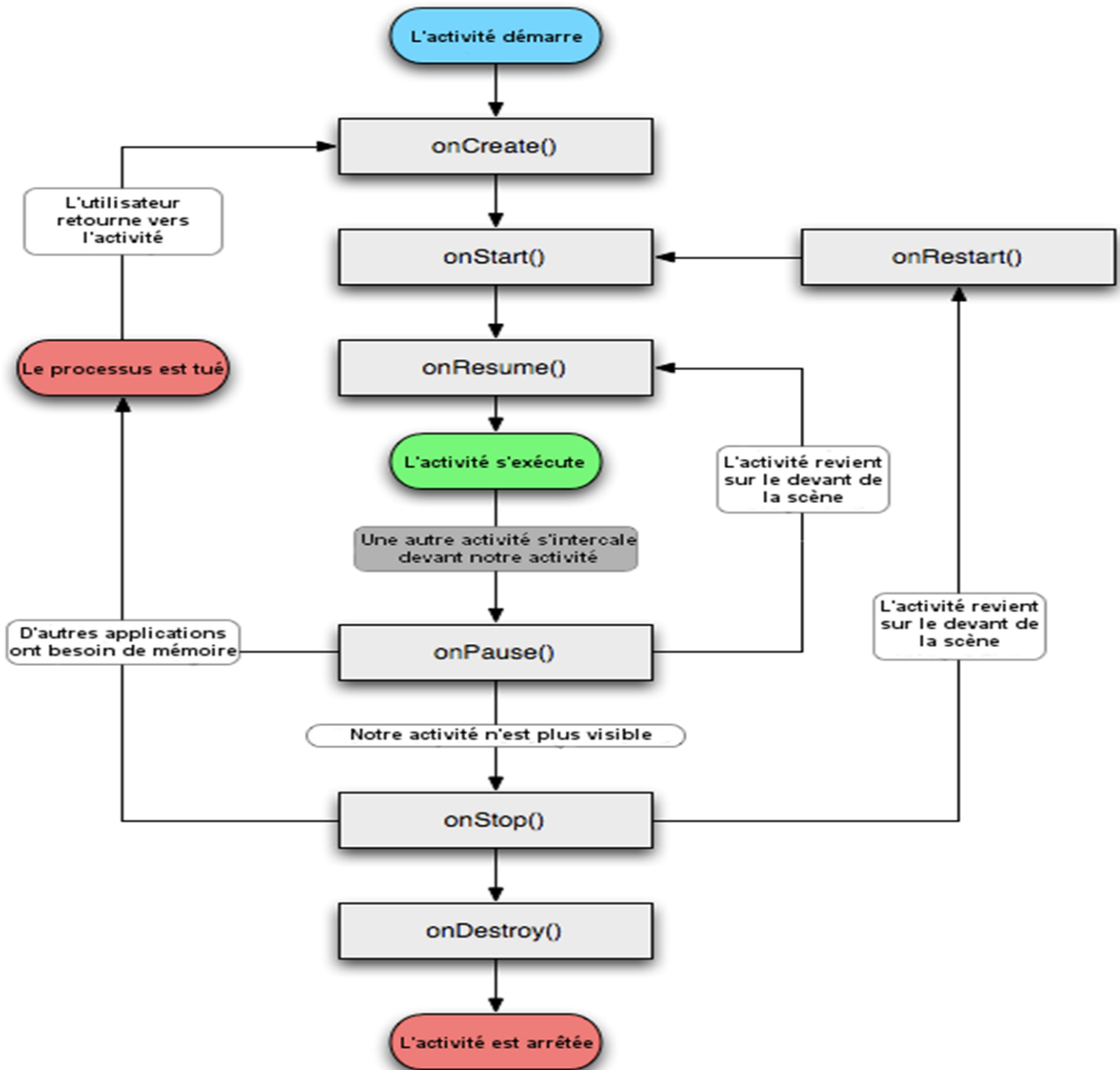


Figure 2.6 : Cycle de vie d'une activité Android [26].

2.6 Conclusion

Dans ce chapitre nous avons présenté brièvement le monde mobile, les principales plateformes mobiles, le système Android, son architecture et l'évolution de ses versions. En fin nous avons parlé sur les applications mobiles, ses avantages et ses inconvénients et les applications Android d'une manière précisée. Le chapitre suivant permet de détailler la conception de notre application.

CHAPITRE 3

Présentation, conception et analyse

« La conception de logiciel met en œuvre un ensemble d'activités qui à partir d'une demande d'informatisation d'un processus permettent la conception, l'écriture et la mise au point d'un logiciel »

3.1 Introduction

La complexité et la taille des logiciels nécessitent des outils, des techniques et des méthodes pour mener à terme la réalisation d'applications dans les meilleures conditions. Tout processus de développement d'applications ou systèmes informatiques doit suivre une démarche bien définie, Pour cela, la normalisation de leur conception et de leur développement s'impose.

Les méthodes de conception viennent à point nommé pour guider le concepteur dans sa tâche de conception. Elles imposent, une certaine discipline de travail qui, en retour, garantit la modularité et la justesse des applications développées, en conséquent nous avons opté pour une démarche de conception orientée objets, en se basant sur la modélisation UML.

Au niveau de ce chapitre, nous allons présenter notre application : objectif, fonctionnement et architecture, les besoins fonctionnels et non fonctionnels, Nous allons présenter aussi brièvement le model UML, les diagrammes utiliser d'une manière détaillée. Enfin, nous allons terminer avec une conception détaillée de notre application en s'aidant des principes diagrammes d'UML (cas d'utilisation, séquence et classe).

3.2 Présentation de l'application

3.2.1 Objectif

Nous souhaitons, à travers notre projet de développer une application « PhDmonitoring » qui permet de :

- Faciliter la communication entre le directeur de thèse et ses doctorants ou sera un espace privé pour eux.
- Faciliter la consultation de l'état d'avancement des doctorants (les formations, les publications, la rédaction, etc.) et l'envoi des remarques.
- Arrondir les distances et gagner du temps gras à la technologie de téléconférence.
- La possibilité d'un suivi permanant et continu.

3.2.2 Apport de l'application sur le suivi des doctorants

- Maximiser le nombre des doctorants cadré par un directeur de thèse.
- Faciliter les tâches de suivi des doctorants.
- Communication perfectible vis-à-vis des doctorants.
- Minimiser le nombre d'abandons.
- Augmenter le taux de réponses aux demandes et questions des doctorants.
- Une bonne solution pour la continuité de travail et de suivi dans les conditions actuelle qui travers le monde (universités fermées).

3.2.3 Fonctionnement

Notre application a pour but de rassembler les différents acteurs influant lors de la préparation des thèses des doctorants, elle permet aux directeurs de thèses de suivre la progression des étudiants, de partager avec des formations qui peuvent être des cours, des séminaires ou même leur proposer des séminaires d'une manière détaillée. Elle permet aux étudiants d'être corrigés, orientés et suivis au temps réel. Avec la fonctionnalité de réunion, l'étudiant est directement mis en relation avec son encadreur et son directeur de thèse en audiovisuel.

3.2.4 Architecture

L'architecture d'une application décrit les modèles et les techniques utilisés pour concevoir et créer une application. Elle fournit une feuille de route ainsi que les meilleures pratiques à suivre pour créer une application bien structurée.

La figure ci-dessous illustre l'architecture générale de notre application :



Figure 3.1 : *Architecture générale de notre application*

L'architecture est de type client-serveur 2-tier :

L'Application cliente envoie une requête au serveur, le serveur reçoit la requête, ce dernier traite les données, après renvoie la réponse au client, cette communication se fait à travers le protocole HTTP. L'échange de données est en format léger d'échange de données JSON.

Client Android : une application mobile développée en java sous Android studio.

Serveur : Firebase.

3.2.5 Les besoins fonctionnels

Après une étude détaillée de système, la description des exigences fonctionnelles des différents acteurs de l'application est comme suit :

- Gérer les formations pédagogiques.
- Gérer les publications (articles, titre-livres, etc.).
- Gérer les conférences.
- Gérer les réunions.
- Gérer la liste des doctorants.
- Gérer les progressions.
- Gestion de messagerie.

- Gestion notifications.

3.2.6 Les besoins non fonctionnels

Les besoins non fonctionnels décrivent toutes les contraintes techniques, ergonomiques et esthétiques auxquelles est soumis le système pour sa réalisation et pour son bon fonctionnement. Concernent notre application, nous avons les besoins suivants :

- La disponibilité : l'application doit être disponible pour être utilisé par n'importe quel utilisateur.
- La sécurité : l'accès des utilisateurs à l'espace d'application doit être protégé par un mot de passe et un identifiant donné par le système (authentification).
- La performance : l'application doit être performante pour répondre à toutes les exigences des utilisateurs d'une manière optimale.
- L'utilisation : l'application doit fournir une interface et facile à utiliser.
- La taille de l'application : la taille de l'application doit être légère en raison de la faible capacité des smartphones.

3.3 Présentation d'UML

La modélisation en pratique de n'importe quel système informatique est importante dans le développement des logiciels, dans notre projet nous avons choisi le langage de modélisation UML que nous allons exposer dans cette partie.

3.3.1 Définition

Le langage UML (Unified Modeling Language) est un langage de modélisation visuelle commun, riche sémantiquement et syntaxiquement. Il est destiné à l'architecture, la conception et la mise en œuvre des systèmes logiciels complexes par leurs structures aussi bien que leurs comportements. Il n'est pas un langage de programmation, mais il existe des outils qui peuvent être utilisés pour générer du code en plusieurs langages à partir des diagrammes UML. L'UML a une relation directe avec l'analyse et la conception orientées objet [27].

3.3.2 Les objectifs d'UML

- Raisonner sur le comportement du système.
- Détecter les erreurs et les omissions au début du cycle de vie.
- Présenter les conceptions proposées et communiquer avec les parties prenantes.
- Comprendre les exigences.
- Piloter la mise en œuvre.

On va expliquer brièvement les trois types de diagrammes UML utilisés dans la partie conception : diagramme de cas d'utilisation, de séquence et de classe.

3.3.3 Les diagrammes UML utilisés

a. Diagramme de cas d'utilisation

Fondamentalement, quatre éléments doivent être inclus dans un diagramme de cas d'utilisation. Il s'agit des acteurs, des systèmes, des cas d'utilisation et des relations. Les acteurs ne représentent personne ou objet qui interagit avec le système. Il peut s'agir d'êtres humains, d'autres ordinateurs ou d'autres systèmes logiciels. Les cas d'utilisation représentent les actions qui sont effectuées par un ou plusieurs acteurs dans un but particulier [27].

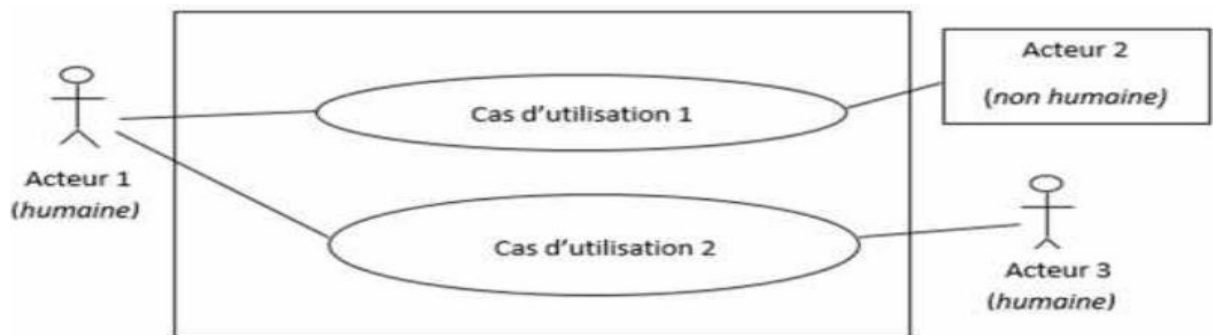


Figure 3.2 : Diagramme de cas d'utilisation

b. Diagramme de séquence

Une ligne de vie représente les instances typiques des composants ou des classes de votre système. Les messages sont affichés sous forme de flèches. Ils peuvent être complets, perdus ou trouvés ; synchrones ou asynchrones ; appel ou signal. Activer est utilisé pour indiquer l'activation d'un participant. Les objets sont des éléments de modèle qui représentent des

instances d'une ou plusieurs classes. L'acteur spécifie un rôle joué par un utilisateur ou tout autre système qui interagit avec le sujet [27].

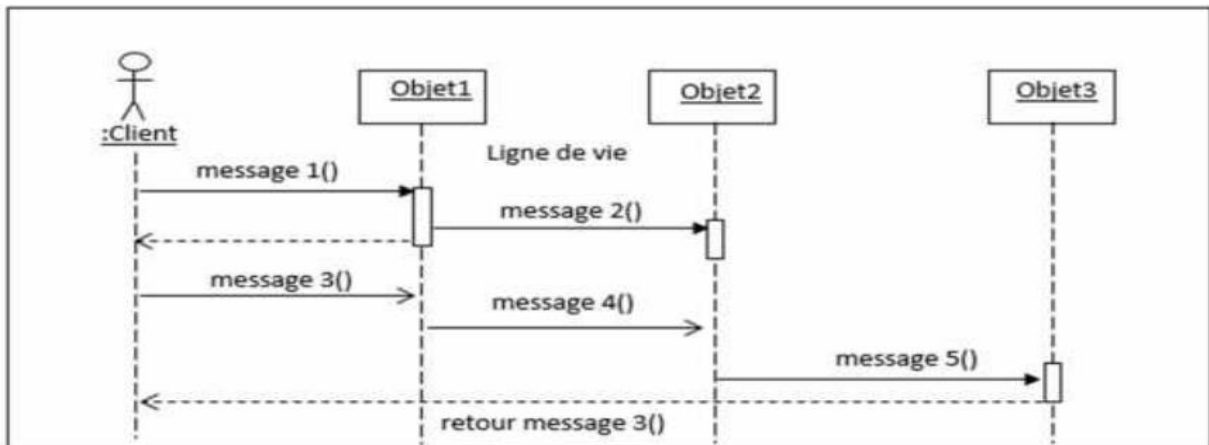


Figure 3.3 : Diagramme de séquence

a. Diagramme de classe

Dans un diagramme de classes UML, les classes représentent une abstraction d'entités ayant des caractéristiques communes. Les associations représentent les relations statiques entre les classes. L'agrégation est un type spécial d'association dans lequel les objets sont assemblés ou configurés ensemble pour créer un objet plus complexe. La généralisation est une relation dans laquelle un élément de modèle (l'enfant) est basé sur un autre élément de modèle (le parent). La relation de dépendance est une relation dans laquelle un aspect, le client, utilise ou dépend d'un autre aspect, le fournisseur [27].

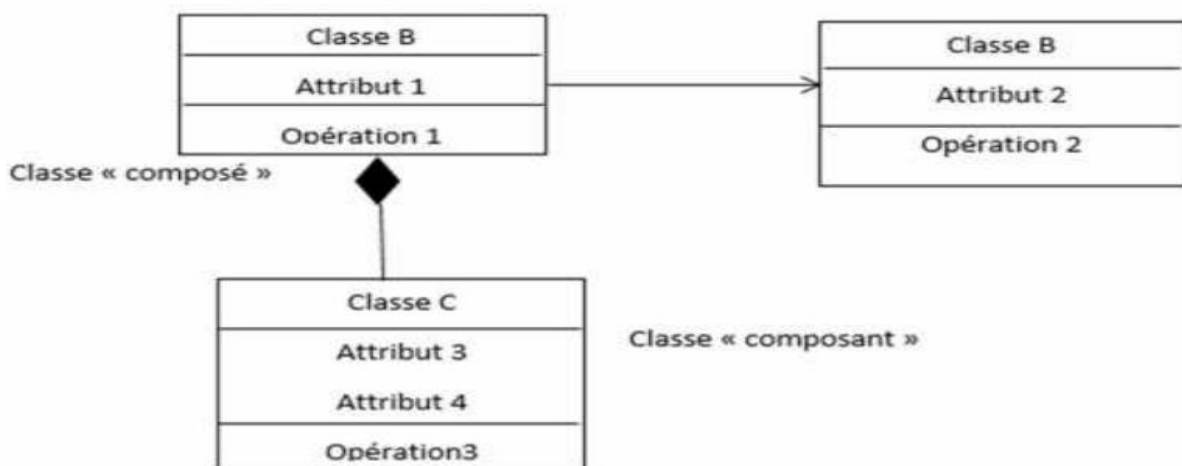


Figure 3.4 : Diagramme de classe

3.4 Conception de l'application

Pour décrire la conception de notre application, nous commençons par les diagrammes de cas d'utilisation, puis les diagrammes de séquence et à la fin le diagramme de classe

3.4.1 Diagramme de cas d'utilisation

a. Identification des acteurs : un acteur est une entité physique (personne) ou abstraite (logiciel) capable d'utiliser le système afin de répondre à un besoin bien défini. Les acteurs de notre système sont :

- **Le doctorant :** un étudiant ou personne faisant des études et des recherches doctorales dans le cadre d'une thèse, pour ambition de devenir docteur une fois sa thèse achevée. Son rôle :

- Gérer les publications (articles, titre-livres).
- Participer aux réunions.
- Utiliser la messagerie.
- Consulter des formations.
- Consulter des conférences.
- Générer un code QR.

- **Le directeur de thèse :** un professeur ou enseignant-chercheur qui supervise le travail d'un doctorant, il possède le privilège de plus haut niveau. Son rôle :

- Gérer les formations pédagogiques (cours, séminaire) et les conférences.
- Gérer les réunions.
- Participer aux réunions.
- Utiliser la messagerie.
- Gérer la liste des doctorants.
- Consulter les progressions.

- **L'encadrant :** enseignant assistant qui donne l'aide et partage les connaissances à ses doctorants. Son rôle :

- Participer aux réunions.
- Utiliser la messagerie.

b. Diagramme de cas d'utilisation de notre application

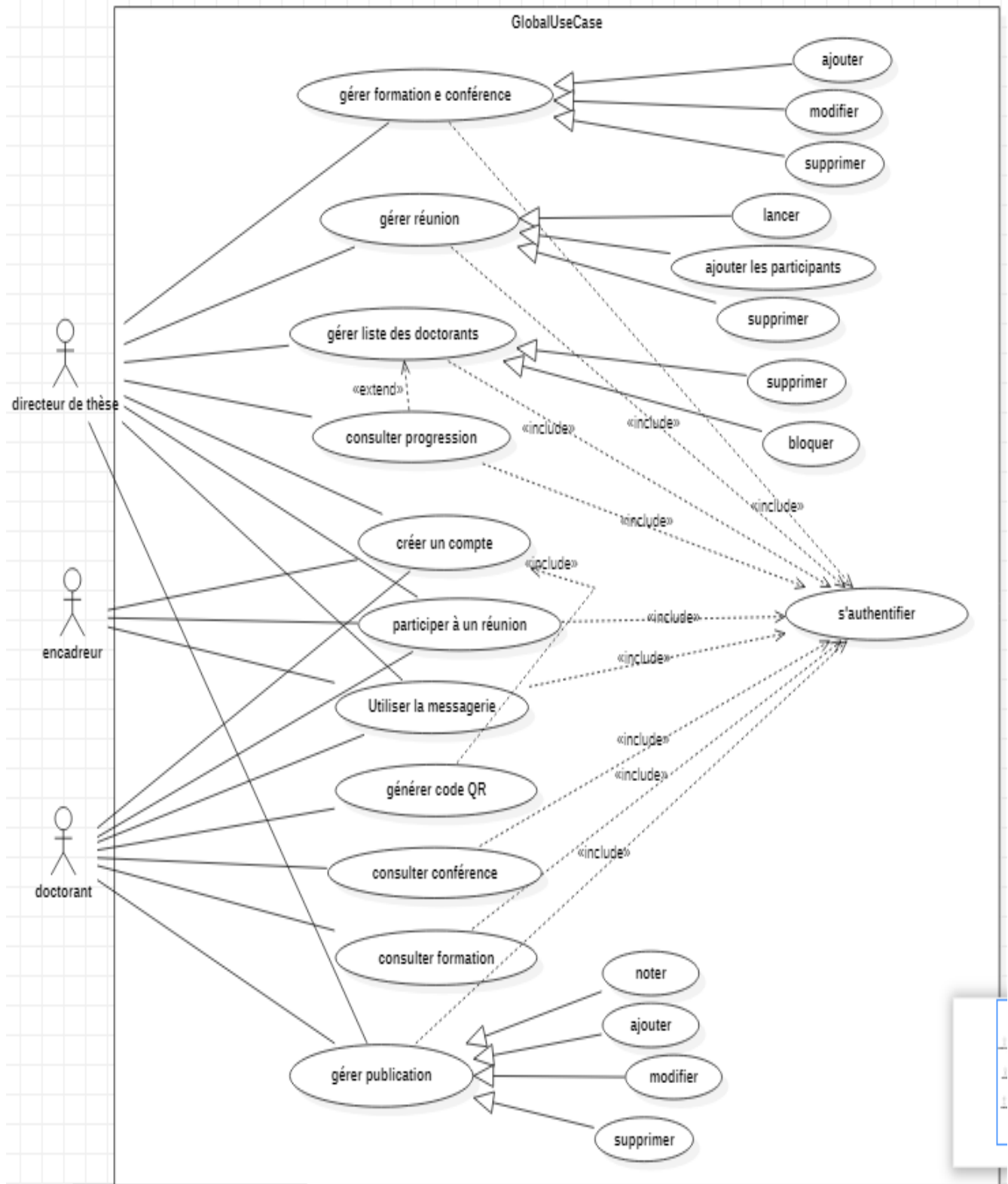


Figure 3.5 : Diagramme de CU globale de notre application

c. Description textuelle des cas d'utilisation

Chaque cas d'utilisation doit être associé une description textuelle des interactions entre l'acteur et le système et les actions que le système doit réaliser en vue de produire les résultats attendus Par les acteurs. La description textuelle du cas d'utilisation est illustrée dans les tableaux suivants :

❖ Cas d'utilisation « créer compte »

Cas d'utilisation	Créer compte
Acteur	Directeur de thèse, doctorant, encadrant.
Objectif	L'inscription pour avoir un compte.
Pré condition	Au cas que l'utilisateur c'est le doctorant, il génère un code QR à partir le code de confirmation, puis il l'envoie à son directeur avec un utile de messagerie (Gmail, Messenger, WhatsApp, etc.) et ce dernier qu'il va valider l'inscription de son doctorant.
Scénario nominal	L'utilisateur clique sur le bouton créer compte, le formulaire d'inscription sera affiché, il va remplir le formulaire et valider Le système enregistrer ses informations.
Scénario alternatif	Des champs vides : Remplir le formulaire et valider. Information incorrecte : Corriger les informations et valider.
Post condition	Un compte créé, un utilisateur inscrit.

Tableau 3. 1 : Description textuelle de cas d'utilisation « créer compte »

❖ Cas d'utilisation « s'authentifier »

Cas d'utilisation	S'authentifier
Acteur	Directeur de thèse, doctorant, encadrant.
Objectif	Valider l'accès à l'espace d'application.
Pré condition	Compte existant.
Scénario nominal	L'utilisateur demande l'accès au système. Le formulaire de saisie l'identificateur et du mot de passe sera affiché

	L'utilisateur saisit l'identifiant et le mot de passe, le système vérifie la validité du nom d'utilisateur et du mot de passe. L'accueil s'affiche.
Scénario alternatif	L'identificateur et le mode passe incorrecte : le système affiche le formulaire une autre fois pour corriger les informations.
Post condition	L'utilisateur authentifie.

Tableau 3. 2 : Description *textuelle* de cas d'utilisation « s'authentifie »

❖ Cas d'utilisation « **participer au réunion** »

Cas d'utilisation	Participer à la réunion
Acteur	Directeur de thèse, doctorant, encadrant.
Objectif	Effectuer une réunion.
Pré condition	Utilisateur s'authentifie. Seulement le directeur de thèse qui a l'autorisation de lancer la réunion.
Scénario nominal	Le directeur lance la réunion, il ajoute les participants. Lorsque la réunion est terminée, le directeur quitte la vidéoconférence.
Scénario alternatif	Action échouée : ressayer.
Post condition	Réunion effectuée.

Tableau 3. 3 : Description *textuelle* de cas d'utilisation « participer à la réunion »

❖ Cas d'utilisation « **utiliser la messagerie** »

Cas d'utilisation	Utiliser la messagerie
Acteur	Directeur de thèse, doctorant, encadrant.
Objectif	Etablir une discussion.
Pré condition	Utilisateur s'authentifie.
Scénario nominal	L'utilisateur ouvre l'onglet de messagerie, choisit le destinataire, puis envoyer un message tapé.
Scénario alternatif	Action échouée : ressayer.
Post condition	Une discussion effectuée.

Tableau 3. 4 : Description *textuelle* de cas d'utilisation « utiliser la messagerie »

❖ Cas d'utilisation « gérer formation et conférence »

Cas d'utilisation	Gérer formation et conférence.
Acteur	Directeur de thèse.
Objectif	Ajouter /modifier/ supprimer une formation ou conférence.
Pré condition	Utilisateur s'authentifie.
Scénario nominal	<p>Ce cas d'utilisation commence lorsque le directeur de thèse désire ajouter, modifier ou supprimer une formation/ conférence.</p> <p>Enchaînement (a) : Ajouter une formation/ conférence Le directeur saisit les informations sur la formation et sélectionner parmi ses doctorants qu'il est intéressé, le système sauvegarde et ajoute la formation à la liste des formations, puis une notification est envoyée.</p> <p>Enchaînement (b) : Modifier une formation/ conférence. Le directeur de thèse cherche une formation puis modifie ses informations.</p> <p>Enchaînement (c) : Supprimer une formation/ conférence. Le directeur cherche une formation et la supprime.</p> <p>Ce cas se terminent lorsqu'une formation est ajoutée, les informations mises à jour ou bien une formation supprimée.</p>
Scénario alternatif	Une des opérations est échouée : réessayer.
Post condition	Une formation ajoutée ou ses informations à jour ou supprimée.

Tableau 3. 5 : Description textuelle de cas d'utilisation « gérer formation et conférence »

❖ Cas d'utilisation « gérer réunion »

Cas d'utilisation	Gérer réunion
Acteur	Directeur de thèse.
Objectif	Lancer /ajouter participant / quitter.
Pré condition	Utilisateur s'authentifie.
Scénario nominal	<p>Ce cas commence lorsque le directeur de thèse envoie la demande de lancement d'une réunion</p> <p>Enchaînement (a) : lancé une réunion Le directeur clique sur le Buttons de lancement</p>

	Enchaînement (b) : ajouter participant le directeur afficher la liste des concernant et puis cliquer sur ajouter. Enchaînement (c) : Quitter. Le directeur de thèse quitte à la fin la réunion. Ce cas se terminé lorsqu'une réunion est terminée.
Scénario alternatif	Action échouée : ressayer.
Post condition	Réunion effectué.

Tableau 3. 6 : Description textuelle de cas d'utilisation « gérer réunion »

❖ Cas d'utilisation « gérer liste des doctorants »

Cas d'utilisation	Gérer liste des doctorants
Acteur	Directeur de thèse.
Objectif	Supprimer un doctorant/bloquer
Pré condition	Utilisateur s'authentifie.
Scénario nominal	Ce cas d'utilisation commence lorsque le directeur de thèse désire supprimer ou bloquer un doctorant. Enchaînement (a) : supprimer. Le directeur cherche un doctorant et supprimer-le. Enchaînement (b) : Bloquer. Le directeur cherche un doctorant et bloquer-le. Ce cas se terminé lorsqu'un doctorant supprimé ou bloqué.
Scénario alternatif	Une des opérations est échouée : réessayer.
Post condition	Un doctorant supprimé ou bloqué.

Tableau 3. 7 : Description textuelle de cas d'utilisation « gérer liste des doctorants »

❖ Cas d'utilisation « consulter progression »

Cas d'utilisation	Consulter progression
Acteur	Directeur de thèse.
Objectif	Afficher la progression d'un doctorant pour voir l'état d'avancement
Pré condition	Utilisateur s'authentifie, doctorant existé.

Scénario nominal	Le directeur de thèse affiche la liste de ses doctorants, puis affiche sa progression.
Scénario alternatif	Action échouée : ressayer.
Post condition	Progression consulté.

Tableau 3. 8 : Description textuelle de cas d'utilisation « consulter progression »

❖ **Cas d'utilisation « gérer publication »**

Cas d'utilisation	Gérer publication
Acteur	Doctorant, directeur de thèse
Objectif	Ajouter / modifier / supprimer / noter publication.
Pré condition	Utilisateur s'authentifie.
Scénario nominal	<p>Ce cas d'utilisation commence lorsque le doctorant désire ajouter, modifier supprimer une publication, ou le directeur désire à noter une publication.</p> <p>Enchaînement (a) : Ajouter Le doctorant saisit les informations, le système sauvegarde.</p> <p>Enchaînement (b) : modifier. Le doctorant cherche une publication puis modifie ses informations</p> <p>Enchaînement (c) : Supprimer. Le doctorant cherche une publication et la supprime.</p> <p>Enchaînement (d) : noter. Le directeur cherche une publication et la distribuer une note.</p> <p>Ce cas se terminé lorsqu'une publication est ajouté, les informations mises à jour, supprimé ou noté.</p>
Scénario alternatif	Une des opérations est échouée : réessayer.
Post condition	Une publication ajoutée ou ses informations à jour, supprimée ou notée.

Tableau 3. 9 Description textuelle de cas d'utilisation « gérer publication »

❖ **Cas d'utilisation « consulter formation »**

Cas d'utilisation	Consulter formation
Acteur	Doctorant.

Objectif	Consulter une formation.
Pré condition	Utilisateur s'authentifie.
Scénario nominal	Le doctorant consulte la liste des formations posté par son directeur de thèse dans la liste des formations.
Scénario alternatif	Opération échouée : ressayer.
Post condition	Formation consultée.

Tableau 3. 10 : Description textuelle de cas d'utilisation « consulter formation »

❖ **Cas d'utilisation « générer code QR »**

Cas d'utilisation	Gérer code QR
Acteur	Doctorant.
Objectif	Gérer un code QR à partir le code de confirmation.
Pré condition	Le code de confirmation est reçu.
Scénario nominal	Gérer le code QR, puis envoyer-le ou directeur de thèse avec un utile de communication (Gmail par exemple).
Scénario alternatif	Code non reçu : renvoyer. Code QR non généré : ressayer.
Post condition	Code de confirmation envoyé sous format d'un code QR.

Tableau 3. 11 : Description textuelle de cas d'utilisation « générer code QR »

Bien que la description textuelle des cas d'utilisation soit indispensable, il est toujours utile de la compléter par une description graphique sous la forme de diagrammes UML. Et pour cela nous avons établi les diagrammes de séquences pour chacun de ses cas d'utilisation.

3.4.2 Diagrammes de séquences :

❖ « Créer compte »

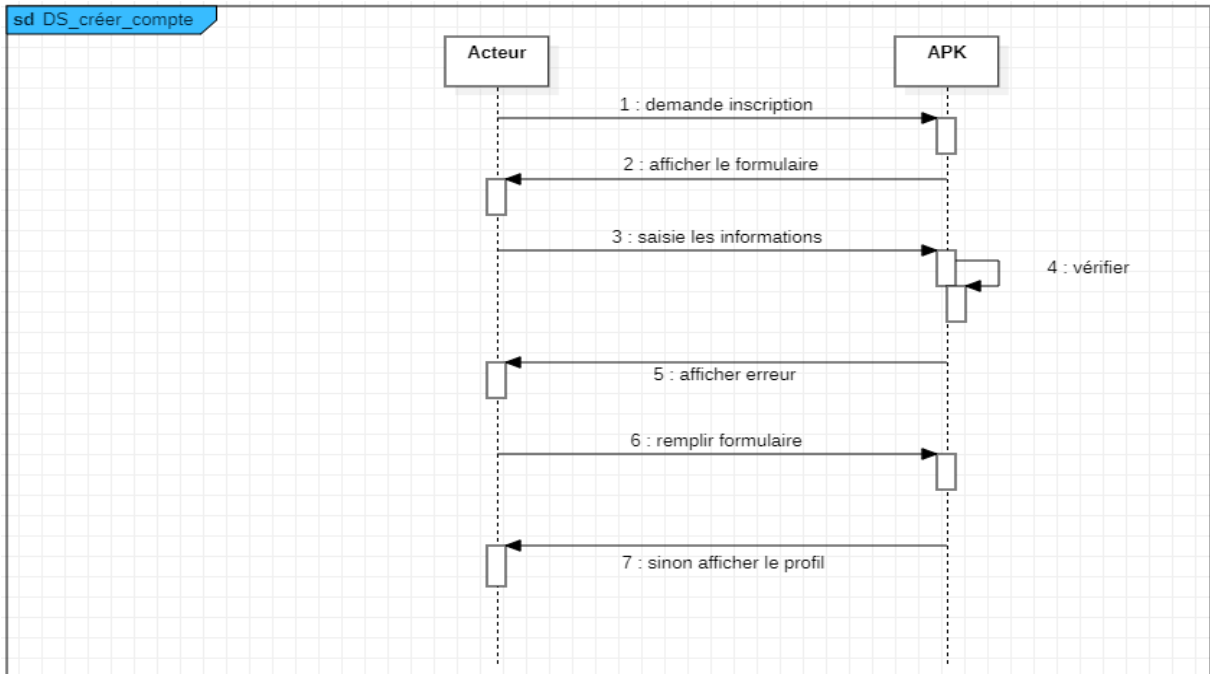


Figure 3.6 : Diagramme de séquence du CU « Créer compte »

❖ « S'authentifie »

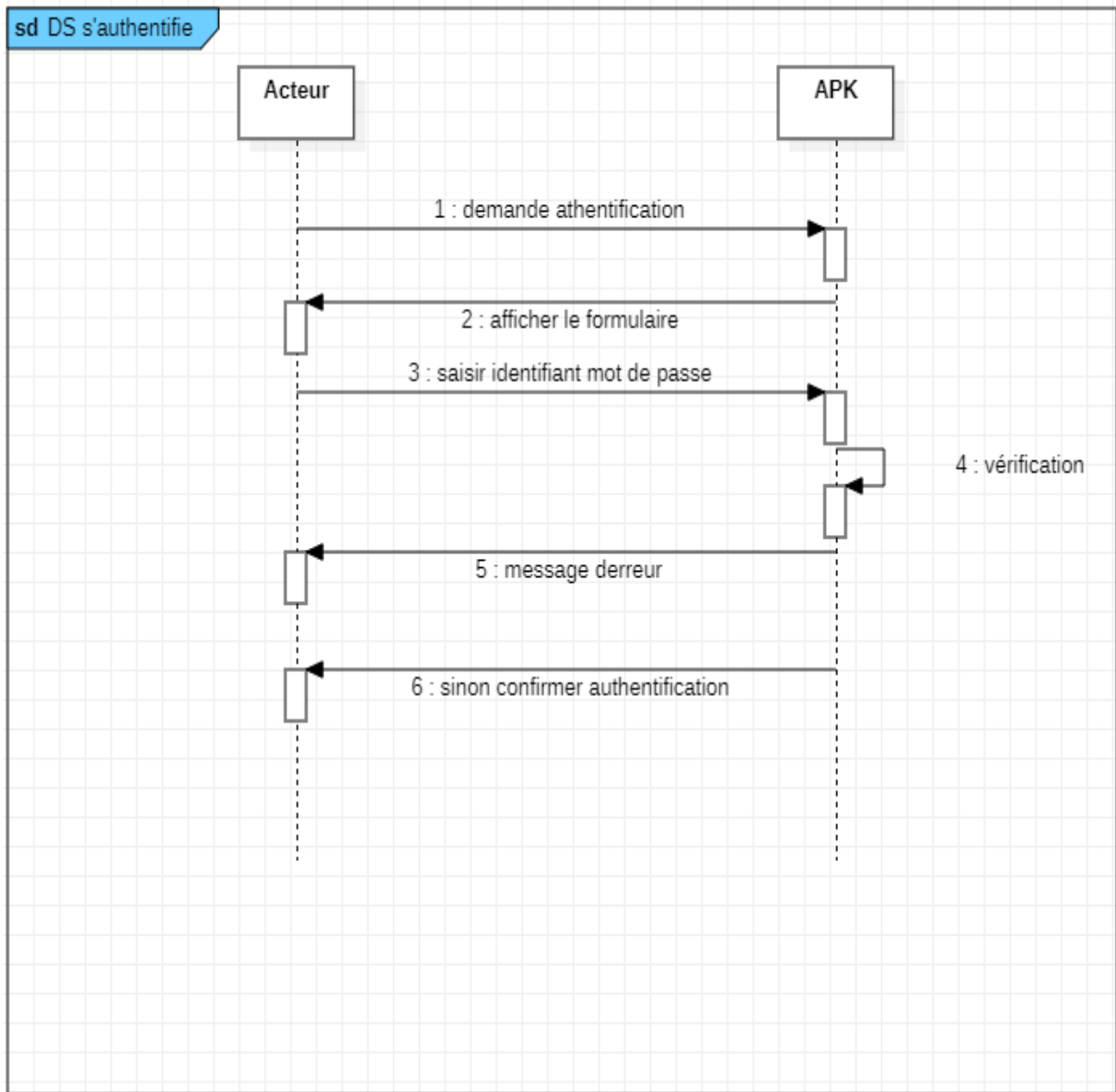


Figure 3.7 : Diagramme séquence du CU « S'authentifie »

❖ « Utiliser messagerie »

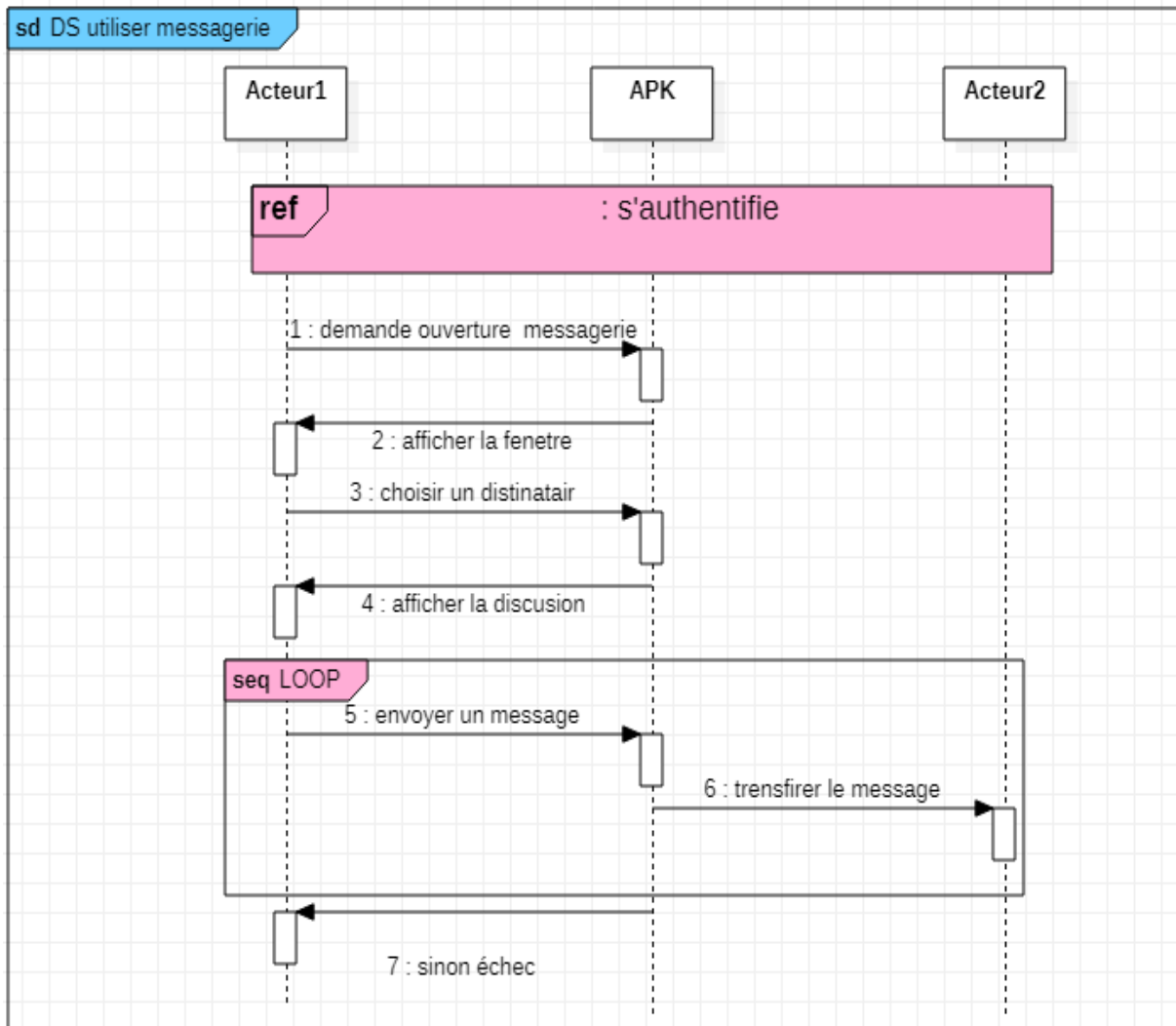


Figure 3.8 : Diagramme de séquence du CU « Utiliser messagerie »

❖ « Gérer formation »

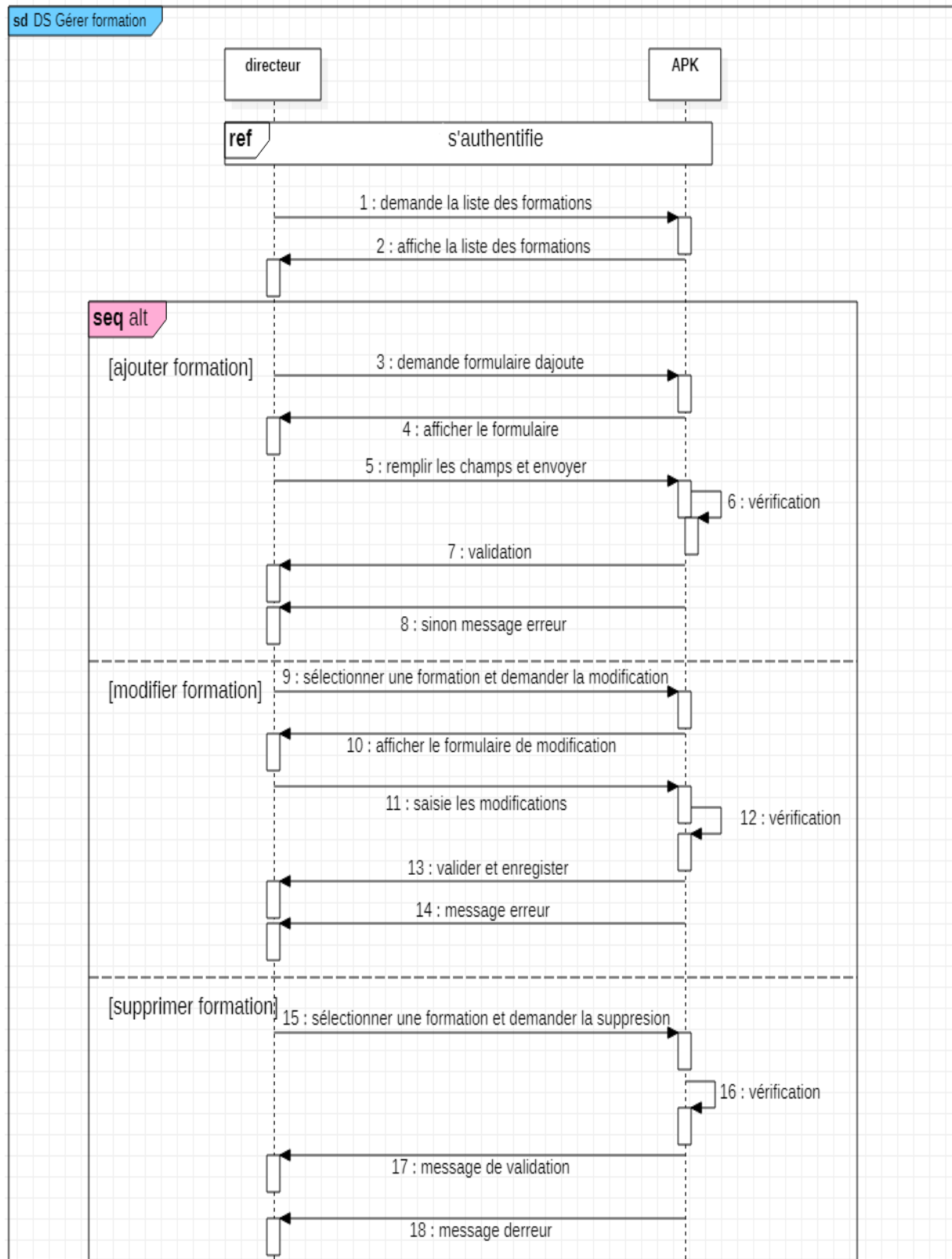


Figure 3.9 : Diagramme de séquence du CU « Gérer formation »

❖ « Gérer réunion »

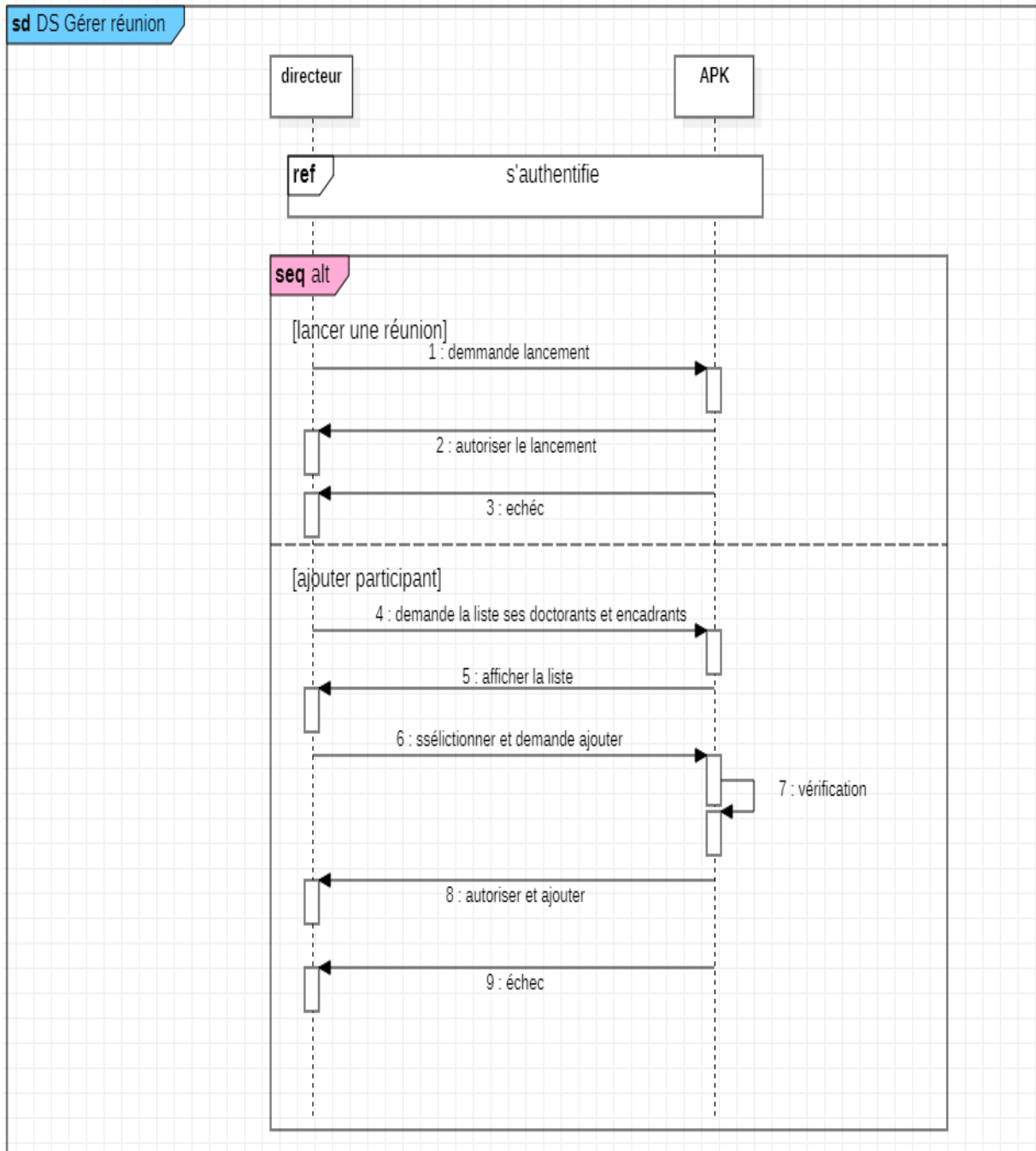


Figure 3.10 : Diagramme de séquence du CU « Gérer réunion »

❖ « Gérer liste des doctorants »

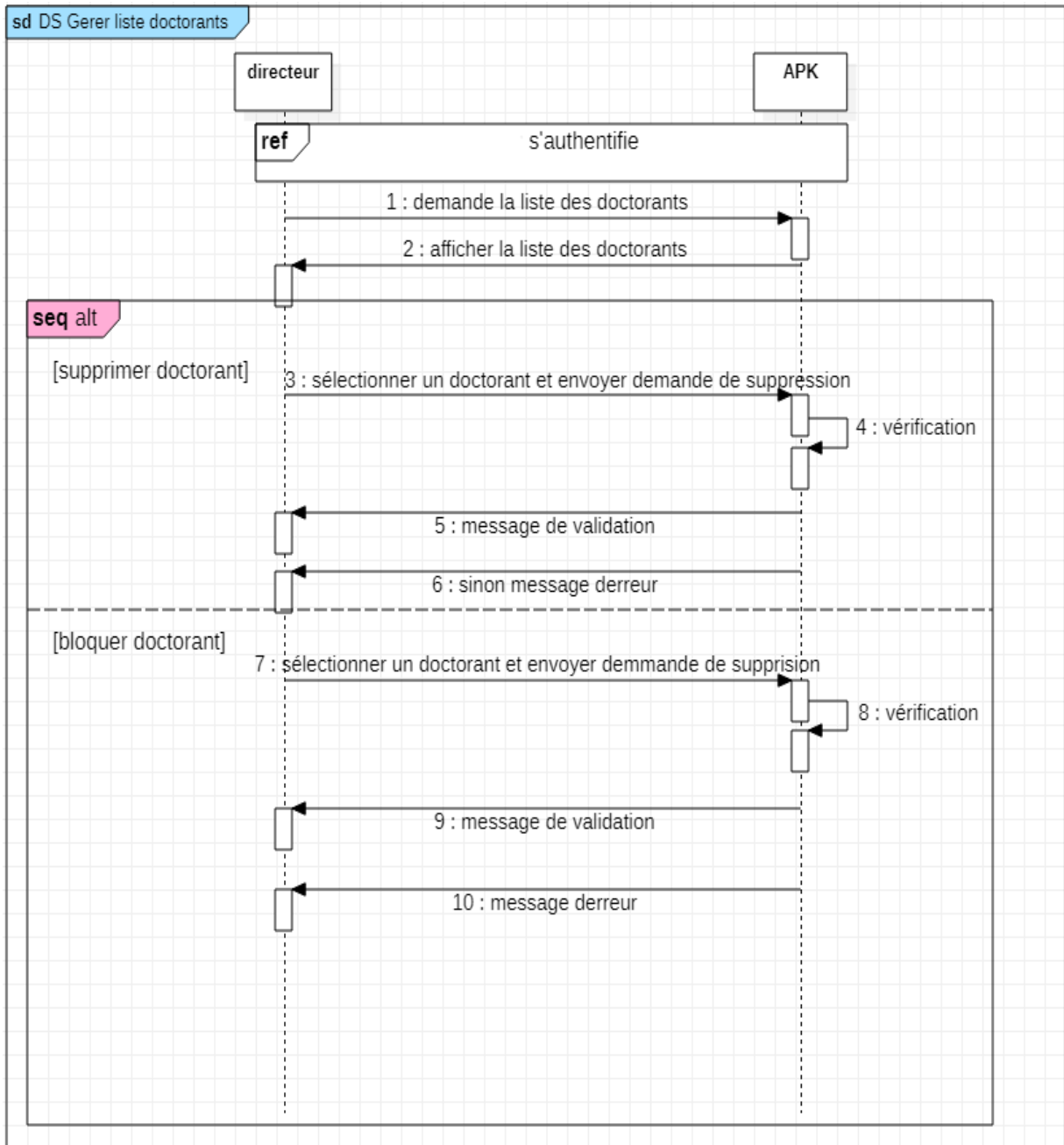


Figure 3.11 : Diagramme de séquence du CU « Gérer liste des doctorants »

❖ « Gérer publication »

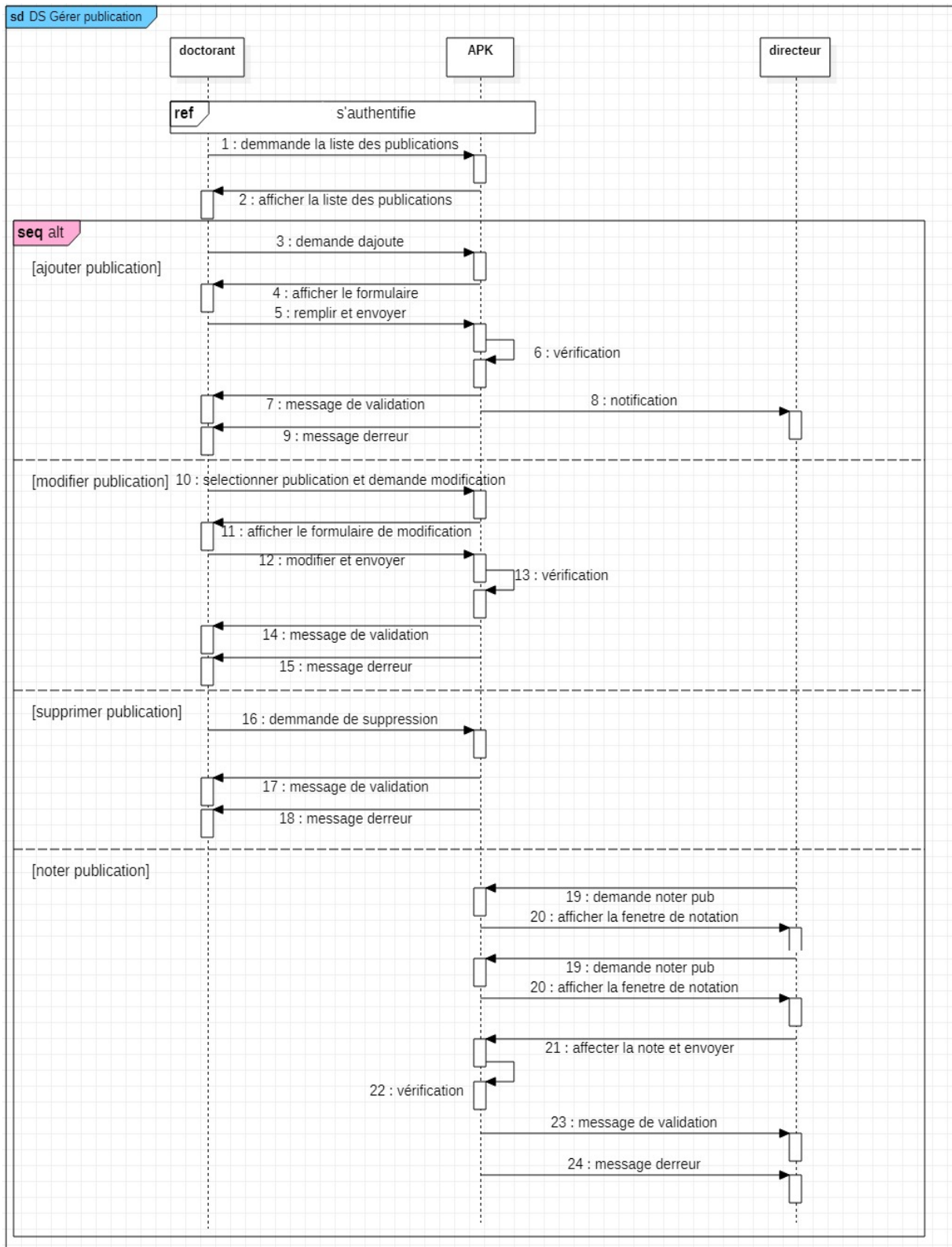


Figure 3.12 : Diagramme de séquence du CU « Gérer publication »

❖ « Consulter formation »

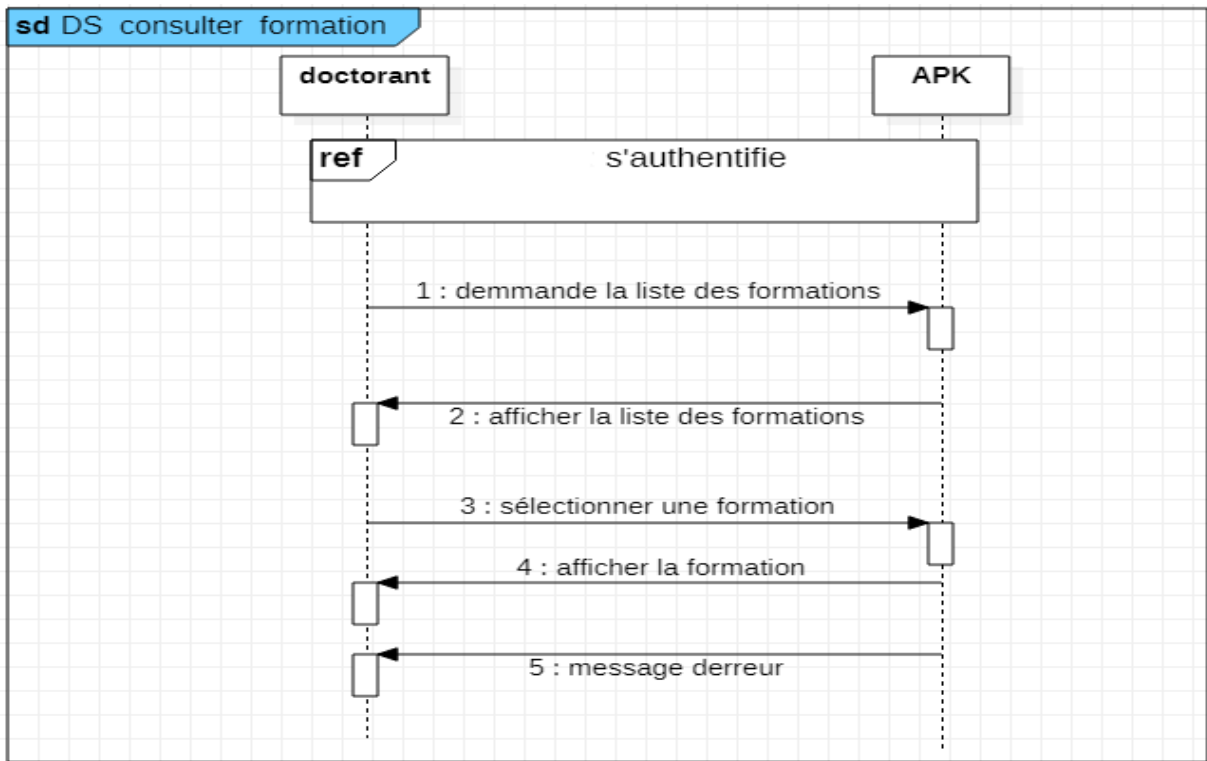


Figure 3.13 : Diagramme de séquence du CU « Consulter formation »

❖ « Gérer QR code »

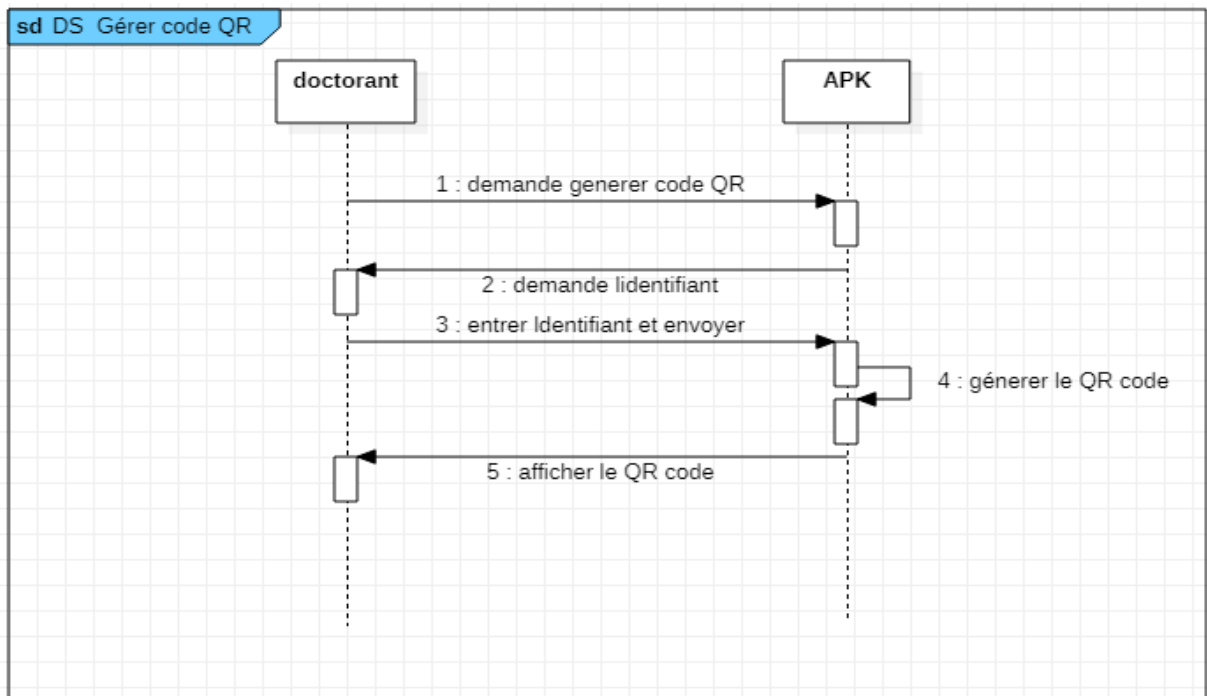


Figure 3.14 : Diagramme de séquence du CU « Gérer QR code »

3.4.3 Diagramme de classe :

La figure ci-dessous illustre le diagramme de classe général :

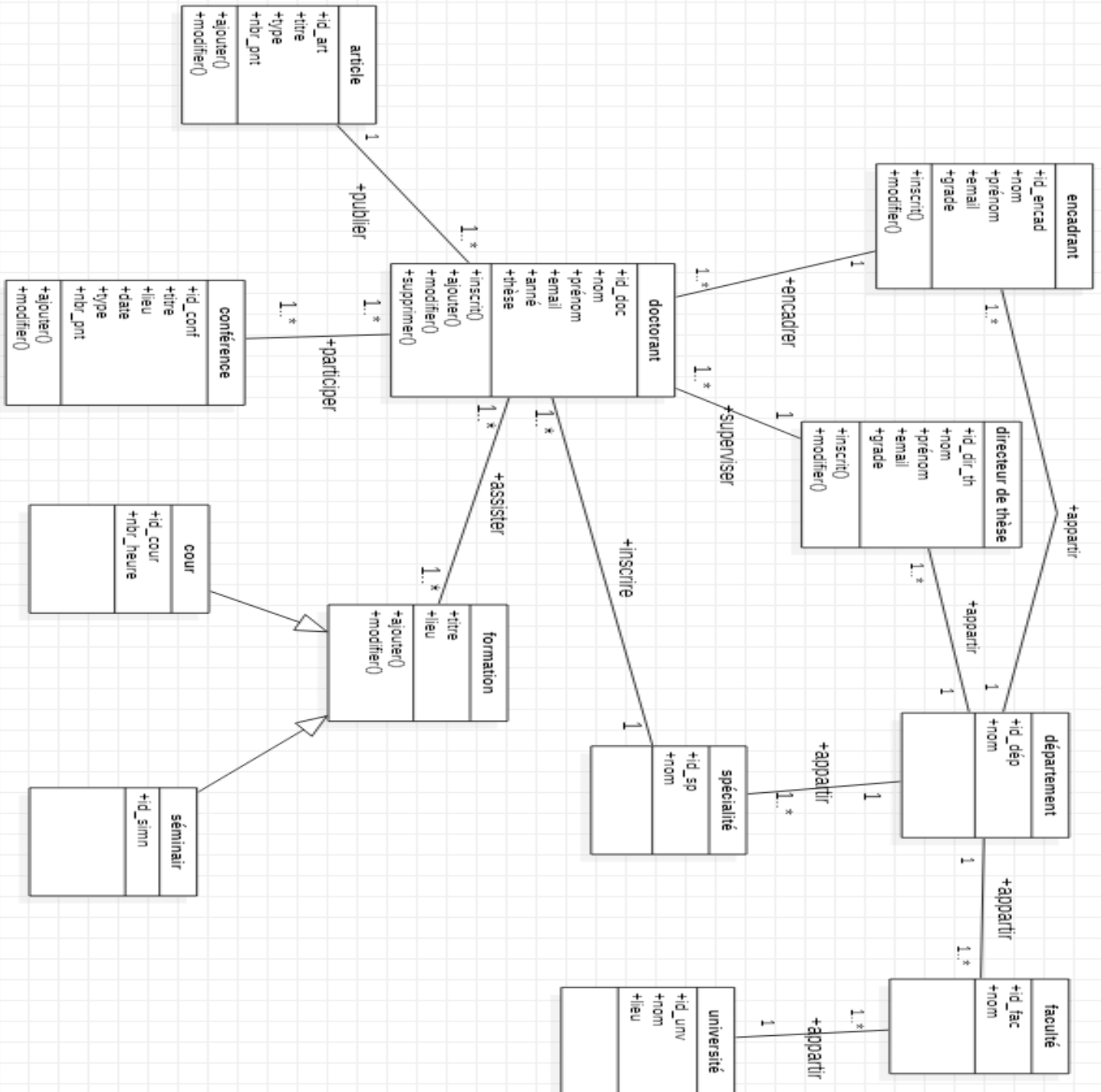


Figure 3.15 : Diagramme de classe de notre application

3.5 Conclusion

Dans ce chapitre nous avons présenté notre application, objectif, fonctionnement et architecture, les besoins fonctionnels et non fonctionnels. Nous avons présenté aussi le modèle UML et les diagrammes appliqués. En phase de conception, nous avons élaboré le diagramme de cas d'utilisation et sa fiche descriptive, les diagrammes de séquences et finalement le diagramme de classe.

Le chapitre suivant sera consacré à l'environnement technique de développement et à la réalisation de notre application.

CHAPITRE 4

implémentation et réalisation

« Le logiciel est une excellente combinaison entre l'art et l'ingénierie »

Bill Gates

4.1 Introduction

La réalisation et l'implémentation représentent l'étape la plus importante qui suit immédiatement la phase de conception. Le choix des outils de développement influence énormément sur le coût en temps de programmation, ainsi que sur la flexibilité du produit à réaliser. Cette phase consiste à transformer le modèle conceptuel établi précédemment en des composants logiciels formant notre système.

Dans ce chapitre, nous allons commencer par la description de l'environnement matériel et logiciel de travail puis, nous décrirons la phase d'implémentation en nous basant sur quelques interfaces.

4.2 Choix technique

4.2.1 Langage JAVA

JAVA est un langage de programmation orienté objet, développé par Sun Microsystems. Il permet de créer des logiciels compatibles avec de nombreux systèmes d'exploitation (Windows, Linux, Macintosh, Solaris). Java donne aussi la possibilité de développer des programmes pour téléphones portables et assistants personnels. Enfin, ce langage peut être utilisé sur internet pour des petites applications intégrées à la page web (applet) ou encore comme langage serveur [28][43].



Nous avons choisi l'outil java pour l'ensemble de ses caractéristiques qui ont largement contribué à son énorme succès, à savoir [29] :

- **Java est interprété** : la source est compilée en pseudo code ou byte code puis exécutée par un interpréteur Java " Java Virtual Machine (JVM)".
- **Java est portable** : il est multiplateforme (ne dépend d'aucune plate-forme). Il est possible d'exécuter des programmes Java sur tous les environnements qui possèdent une JVM.
- **Java est orienté objet** : Chaque fichier source contient la définition d'une ou plusieurs classes qui sont utilisées les unes avec les autres pour former une application.
- **Java est simple** : il est simple d'utilisation.

- **Java est fortement typé** : toutes les variables sont typées et il n'existe pas de conversion automatique qui risquerait une perte de données.
- **Java assure la gestion de la mémoire** : l'allocation de la mémoire pour un objet se fait automatique à sa création et Java récupère automatiquement la mémoire inutilisée grâce au garbage collector (GC) qui restitue les zones de mémoire laissées libres suite à la destruction des objets

4.2.2 XML

Le XML, pour Extensible Markup Language, désigne un langage informatique (ou métalangage) utilisé, entre autres, dans la conception des sites Web et pour faciliter les échanges d'informations sur Internet. Il s'agit, en quelque sorte, d'une version améliorée du langage HTML avec la création illimitée de nouvelles balises. XML permet la mise en forme de documents via l'utilisation de balises. Développé et standardisé par le World Wide Web Consortium à la fin des années 1990. Il se classe dans la catégorie des langages de description (il n'est ni un langage de programmation, ni de requêtes).



Le XML ne demande pas de connaissances techniques en codage pour être utilisé. Comme il est naturellement structuré, il est facile à lire et à comprendre. Pas de besoin d'un logiciel d'édition de code pour l'écrire. Il est universel, il ne rencontre pas de problème de compatibilité avec d'autres technologies. Enfin, il est extensible, puisqu'il est toujours possible d'ajouter de nouvelles balises au fil des besoins [30].

4.2.3 JSON

JSON (JavaScript Objet Notation) est un langage léger d'échange de données textuelles. Pour les ordinateurs, ce format se génère et s'analyse facilement. Pour les humains, il est pratique à écrire et à lire grâce à une syntaxe simple et à une structure en arborescence. JSON permet de représenter des données structurées (comme XML par exemple).



JSON est relativement facile à lire et à écrire, mais aussi facile à analyser et à générer pour les logiciels. Il est souvent utilisé pour sérialiser des données structurées et les échanger sur un réseau, généralement entre un serveur et des applications web [31].

4.2.4 NoSQL

Le NoSQL est un type de bases de données, dont la spécificité est d'être non relationnelles. Ces systèmes permettent le stockage et l'analyse du Big Data et les applications web en temps réel. Les bases de données relationnelles ne sont plus adaptées. Le terme NoSQL furent inventés en 1998 par Carl Stroz, afin de désigner sa base de données relationnelle légère et open source. Ce concept a ensuite été adopté et popularisé par les GAFAM tels que Google, Facebook ou Amazon confrontés à d'immenses volumes de données.



Une base de données NoSQL est une base de données “non relationnelle”. Il est possible d’y stocker des données sous une forme non structurée, sans suivre de schéma fixe. Les jointures ne sont plus nécessaires.

Les systèmes NoSQL sont compatibles avec une large variété de technologies permettant le stockage de données structurées, non structurées, semi-structurées ou polymorphique [32] [44].

4.3 Environnement matériel

Pour développer l'application, nous avons utilisé comme environnement matériel

Un ordinateur qui possède les caractéristiques suivantes :

Marque	MSI
Processeur	Intel(R) Core (TM) i7-10750H CPU @ 2.60GHz 2.59 GHz
Mémoire RAM	16,0 Go
Disque dur	512GB nMVE M2
Système d'exploitation	64 bits, processeur x64, Windows 10 Famille 20H2
Carte graphique	Nvidia GEFORCE RTX 2060 8GB
Ecran	17 pouces 144 Hz gaming

Et deux smartphones pour les tests qui possèdent les caractéristiques suivantes :

Marque	ACE Clever1	Xiaomi Redmi 8
Processeur	MTK 6762 Octa core(2.0GHz)	Octa-core (2x1.95 GHz Cortex)
Mémoire RAM	3 Go	4Go
Capacité	32Go	64Go
Version d'Android	8.1.0	9.0
Ecran	6.19 pouces	6.22 pouces
Batterie	4000mAh	5000 mAh

4.4 Environnement logiciel

4.4.1 Android studio

Android Studio est l'environnement de développement intégré (IDE) officiel pour le développement d'applications Android. Il est basé sur IntelliJ IDEA, un environnement de développement intégré Java pour les logiciels, et intègre ses outils d'édition de code et de développement. Pour prendre en charge le développement d'applications au sein du système d'exploitation Android, Android Studio utilise un système de génération basé sur Gradle, un émulateur, des modèles de code et une intégration GitHub. Chaque projet dans Android Studio a une ou plusieurs modalités avec du code source et des fichiers de ressources. Ces modalités incluent les modules d'application Android, les modules de bibliothèque et les modules de Google App Engine. Les applications construites dans Android Studio sont ensuite compilées au format APK pour être soumises au Google Play Store.

Le logiciel a été annoncé pour la première fois à Google I/O en mai 2013 et la première version stable a été publiée en décembre 2014. Android Studio est disponible pour les plates-formes de bureau Mac, Windows et Linux. Il a remplacé Eclipse Android Développement Tools (ADT) en tant que principal IDE pour le développement d'applications Android [33].

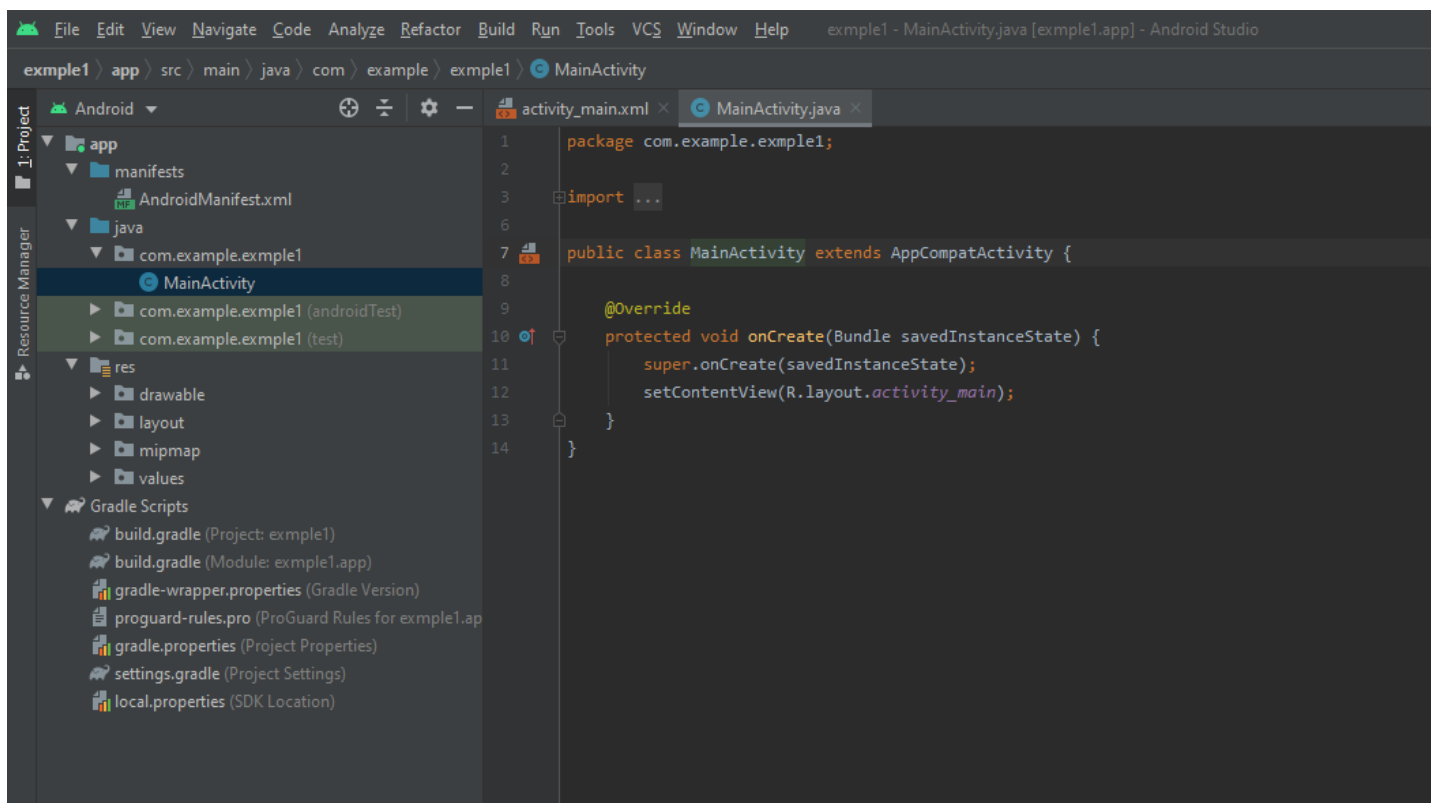


Figure 4.1: Fenêtre principale d'Android studio

Chaque projet dans Android Studio contient un ou plusieurs modules et chaque module d'application contient les fichiers suivants [34] :

- **Le Fichier manifeste** : Le fichier manifeste (AndroidManifest.xml) contient toutes les informations de l'application telles que les permissions, l'icône et le nom de l'application, Les composants.
- **Le répertoire Java** : ce répertoire contient tous les fichiers de code source de l'application, y compris les codes de teste JUnit.
- **Le répertoire res** : ce répertoire contient toutes les ressources de l'application comme les fichiers drawable, un ou plusieurs fichiers layout, les couleurs etc.
- **build.gradle(Project)** : ce fichier est utilisé pour définir toutes les dépendances et les configurations qui seront appliquées dans le projet.
- **build.gradle(app)** : ce fichier est utilisé pour définir les configurations et les dépendances qui seront appliquées dans le module application.

- **proguard-rules.pro** : ce fichier est pour définir toutes les règles proguard spécifique au projet.

4.4.2 Android SDK

Le kit de développement Android SDK (Software Development Kit en anglais) permet de créer des applications sur la plate-forme Android. Pour le développement natif Android, Esokia travaille avec Android Studio devenu l'IDE officiel pour Android. Développé sur le populaire IDE IntelliJ IDEA Java, Android Studio permet d'avoir un IDE totalement dédié aux développements Android. Il dispose d'une interface intuitive et simple d'utilisation pour les développeurs et représente donc un gain de temps considérable.

Un kit de développement SDK incorpore généralement un outil de débogage, un environnement de développement intégré (IDE) ou une notice d'utilisation pour faciliter le développement logiciel. Les améliorations apportées à Android SDK vont de pair avec l'évolution globale de la plate-forme Android [35] [45].

4.4.3 StarUML

StarUML est une plateforme de modélisation logicielle qui prend en charge UML (Unified Modeling Language). Il est basé sur UML version 1.4 et fournit onze types de diagrammes différents, et il accepte la notation UML 2.0. StarUML excelle dans la personnalisation de l'environnement de l'utilisateur et a une grande extensibilité dans ses fonctionnalités. StarUML offre une extensibilité et une flexibilité excellentes. Les utilisateurs peuvent créer leurs propres approches et cadres en fonction de leurs méthodologies. L'outil peut également être intégré à n'importe quel outil externe [36].



Le développement d'applications mobiles robustes et de haute qualité est une proposition qui exige une plate-forme de développement puissante et riche en fonctionnalités.

Firebase de l'écurie Google est une de ces plateformes qui a séduit les développeurs du monde entier, pour cela nous avons la choisi comme un serveur à notre application. On va présenter brièvement cette plateforme

4.5 Plateforme Firebase

4.5.1 Présentation

Firebase est une plateforme de développement d'applications mobiles développée par Firebase Inc. en 2011, puis acquis par Google en 2014. En octobre 2018, la plate-forme Firebase comptait 18 produits, utilisés par 1,5 million d'applications. Elle est fondamentalement un ensemble d'outils sur lesquels les développeurs peuvent compter, créant des applications et les développant en fonction de la demande. La fonctionnalité phare de Google Firebase réside dans sa base de données No SQL dite "temps réel". Une brique qui positionne l'offre de Google comme un successeur possible à Parse [37].



Firebase vise à résoudre trois problèmes principaux pour les développeurs :

- Créer une application, rapidement.
- Publiez et supervisez une application en toute confiance.
- Faire participer les utilisateurs.

Firebase met à disposition différents services, on va détailler les services qu'on a utilisés dans notre application.

4.5.2 Les services Firebase utilisés

a. Firebase Realtime Database : permet de stocker et de synchroniser des données entre les utilisateurs d'application en temps réel. Cela facilite l'accès des utilisateurs à leurs données depuis n'importe quel appareil : Web ou mobile, et aide les utilisateurs à collaborer les uns avec les autres. Chaque fois que les données sont mises à jour dans la base de données temps réel, celle-ci les stocke dans le cloud et informe simultanément tous les périphériques intéressés en millisecondes.

b. Firebase Cloud Messaging (FCM) : Firebase fournit une connexion fiable et à faible consommation de batterie entre le serveur et les périphériques, permet d'envoyer et de recevoir

gratuitement des messages de données (limite de 4 Ko) et des notifications sur iOS, Android et sur le Web.

L'envoi de messages instantanés ou à un moment ultérieur dans le fuseau horaire d'utilisateurs et l'envoi de données d'application personnalisées notamment la définition des sons, des dates d'expiration ou en encore des priorités est également possible. La fonctionnalité permet également de faire un suivi des évènements de conversion.

4.6 Bibliothèques utilisées

4.6.1 Jitsi meet Android SDK

Jitsi Meet est une application JavaScript WebRTC open source (Apache) qui utilise Jitsi Videobridge pour fournir des vidéoconférences de haute qualité, sécurisées et évolutives. Elle est disponible sur mobile depuis un certain temps déjà. L'application fonctionne à la fois sur Android et iOS, et bien qu'elle utilise `meet.jit.si` pour créer des conférences par défaut, elle fonctionne avec n'importe quel déploiement Jitsi Meet personnalisé, en spécifiant simplement l'URL complète. Jitsi Meet permet une collaboration très efficace.

Aujourd'hui, ces utilisateurs seront heureux d'apprendre que le SDK Jitsi Meet pour mobile. Ils peuvent diffuser leur bureau ou seulement certaines fenêtres. Jitsi Meet prend également en charge l'édition de documents partagés avec Etherpad. Il est disponible pour Android et iOS et l'application Jitsi Meet est maintenant une toute petite application qui utilise le SDK [38].

4.6.2 Retrofit

Retrofit est un client HTTP de type sécurisé pour Android et Java développé par Jake Wharton ainsi que l'ensemble du staff de Square. Cette librairie est basée elle-même sur le client REST OK Http (développée encore par Square !) et permet d'implémenter plus facilement et rapidement des requêtes réseau sur Android (Java ou Kotlin).

Retrofit permet évite ainsi d'installer manuellement toutes les parties nécessaires à l'exécution d'une requête, comme par exemple la gestion des réponses JSON ou la création

d'une AsyncTask. Cela permet un gain de temps conséquent et un code plus clair pour des performances équivalentes [39].

4.6.3 QR Generator

QR Code Generator API est une API REST multi-navigateurs qui obtient une entrée JSON avec du texte brut (comme expliqué ci-dessous), contenant un ou plusieurs champs selon le cas et renvoie une chaîne JSON qui contient une image encodée en base64 du QR Code. Cette chaîne encodée en base64 peut être la source de tout contrôle d'image au niveau de l'attribut src [40].

Un QR Code est une représentation visuelle d'une chaîne formée de caractères alphanumériques. Il est largement utilisé pour coder des informations de nombreux types tels que : texte brut, codes numériques, codes alphanumériques, liens Web, données de connexion Wi-Fi telles que SSID, mot de passe, données de paiement bitcoin, appels Skype, message WhatsApp, MMS et SMS, appel d'un numéro de téléphone, géolocalisation comme latitude et longitude et ainsi de suite [40].

4.6.4 QR code scanner

Code Scanner est une bibliothèque de scanner de code QR pour Android, basée sur ZXing avec des fonctionnalités supplémentaires telles que la mise au point automatique et contrôle de la lumière du flash l'orientations d'écran portrait et paysage, les caméras avant et arrière, le viseur personnalisable et la mise au point tactile. La bibliothèque prend en charge la numérisation d'un flux vidéo continu à partir d'une webcam ainsi que la numérisation d'images individuelles.

Le développement de cette bibliothèque est sponsorisé par nimiq, la première blockchain au monde basée sur un navigateur [41].

4.7 Présentation de quelques interfaces de l'application

Nous présentons dans cette section quelques captures d'écran des interfaces principales de notre application.

1. Fenêtre « bienvenue »



Figure 4.2 : *fenêtre bienvenue*

2. Fenêtres d'inscription

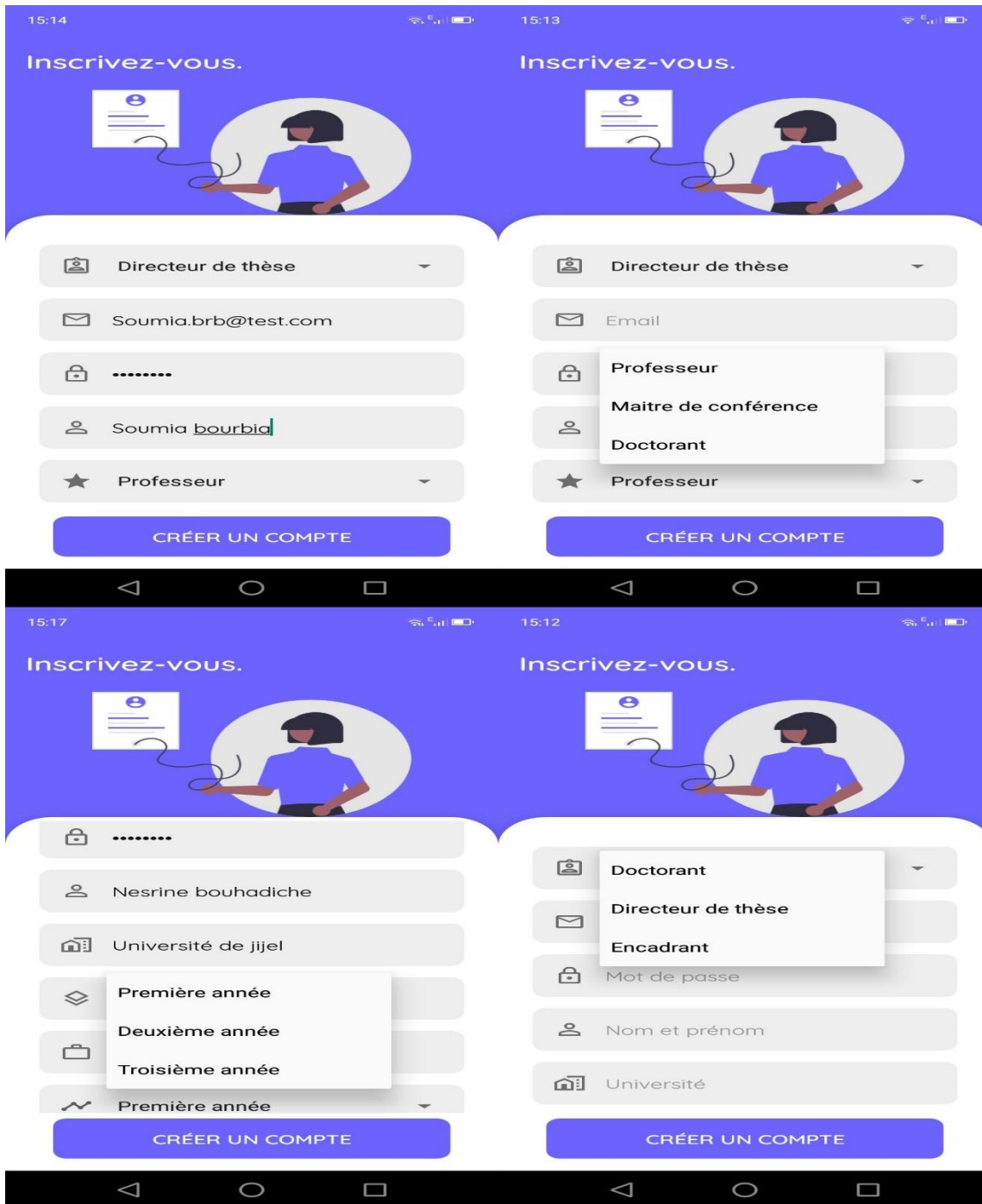


Figure 4.3: Fenêtres inscription

3. Fenêtres de connexion

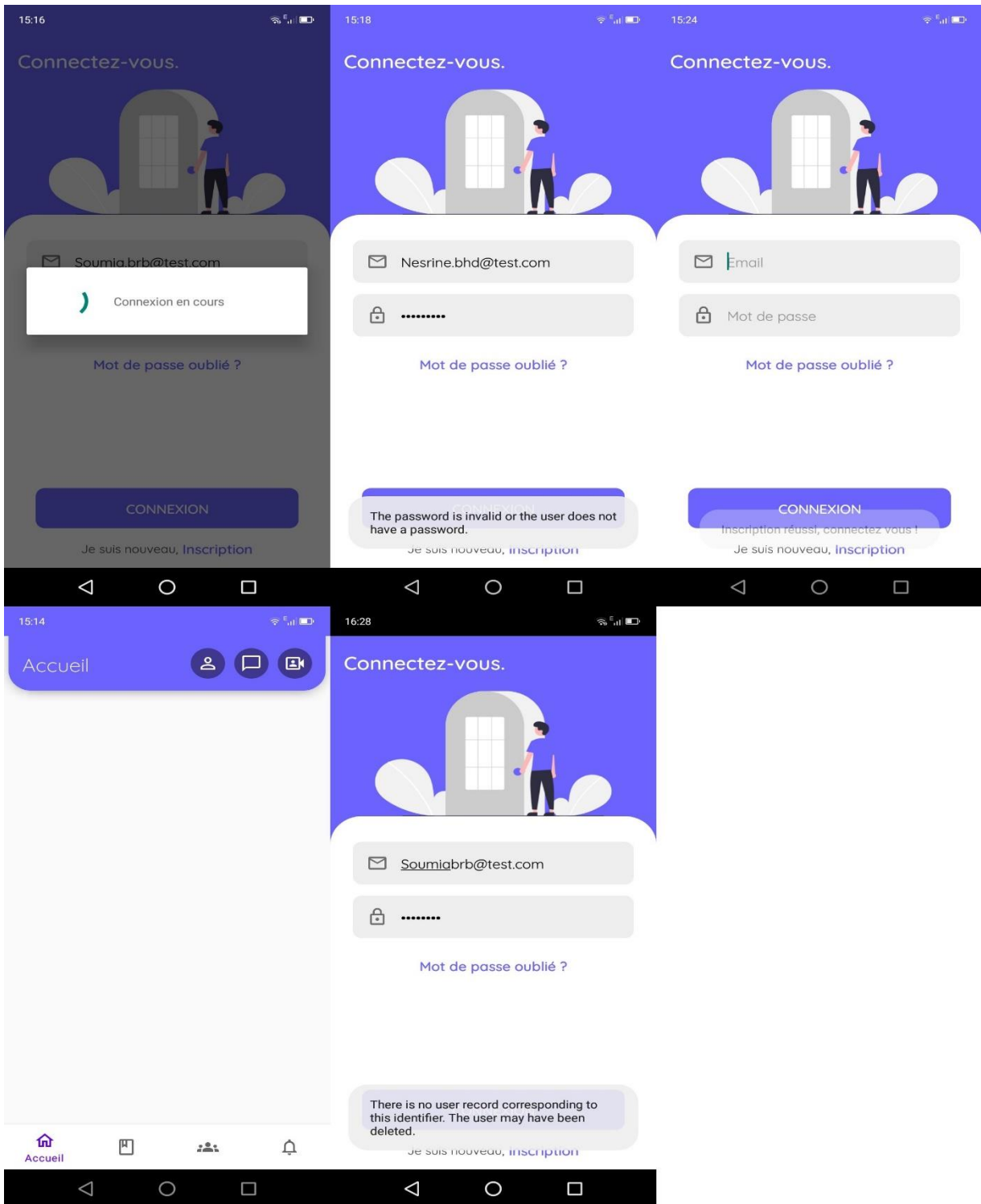


Figure 4.4: Fenêtres connexion

4. Fenêtres de génération de QR code

a. Coté envoie doctorant

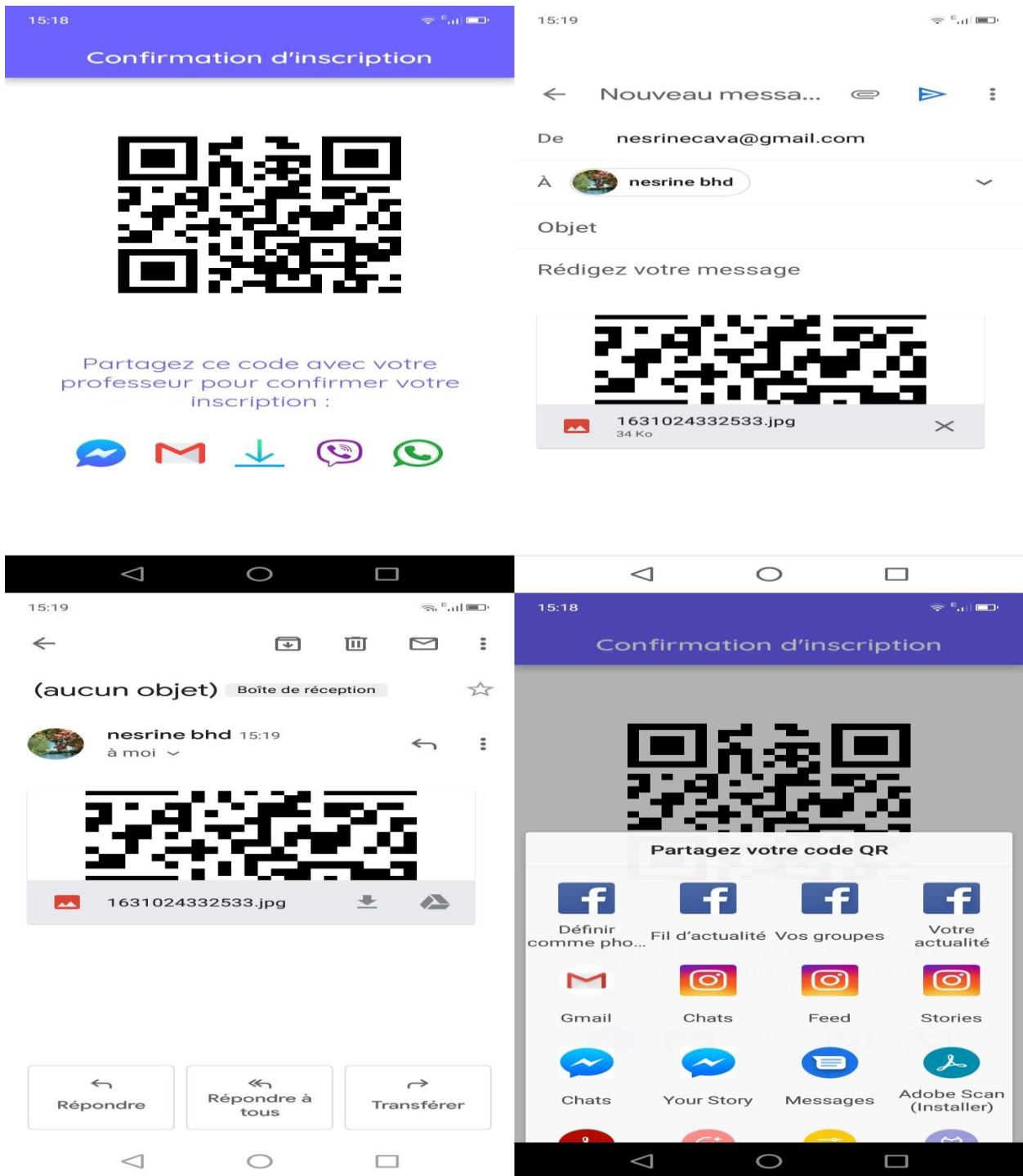


Figure 4.5: Fenêtres l'envoi de QR code

b. Coté réception : directeur de thèse

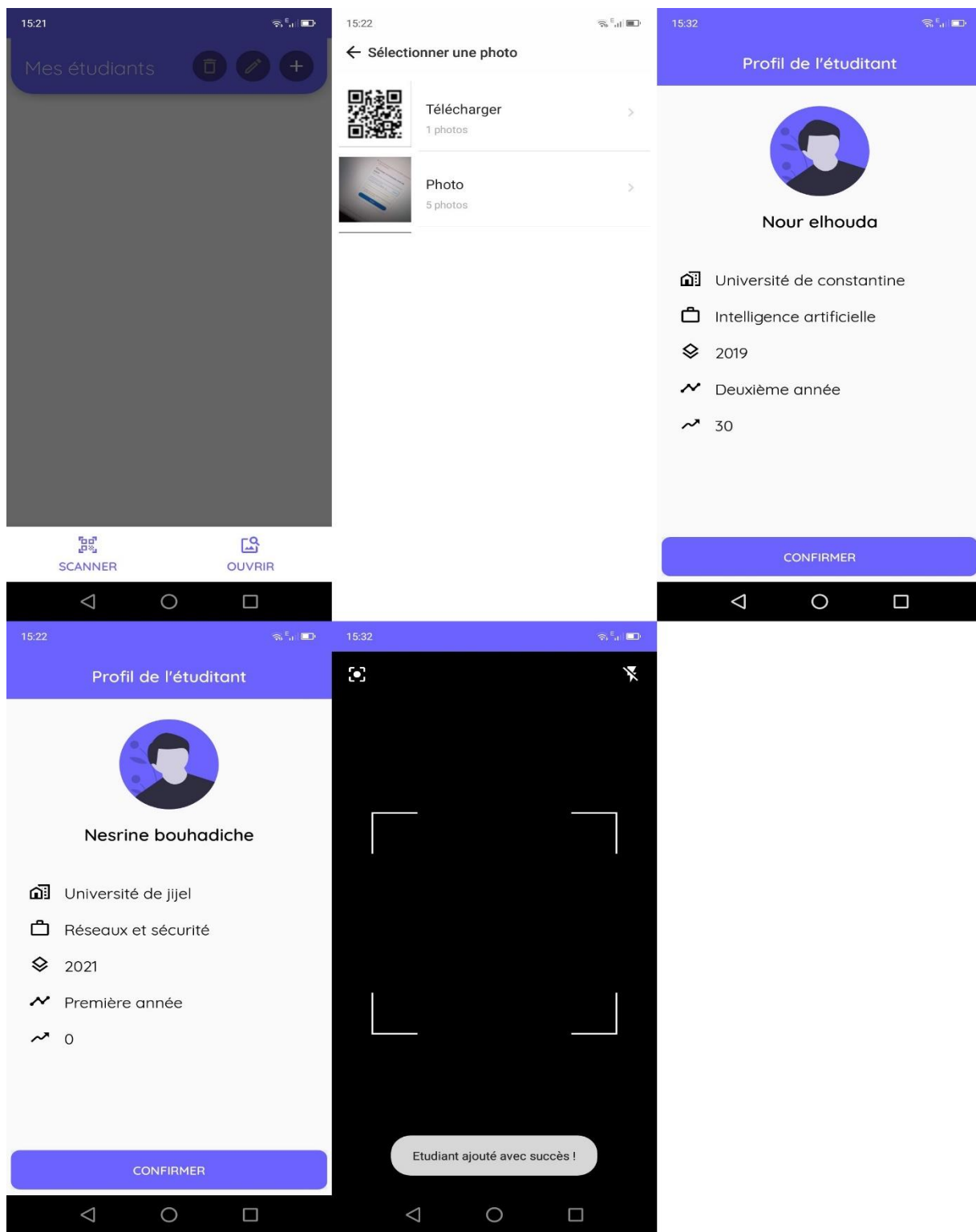


Figure 4.6: Fenêtres réception, scan et confirmation

5. Fenêtres des profils

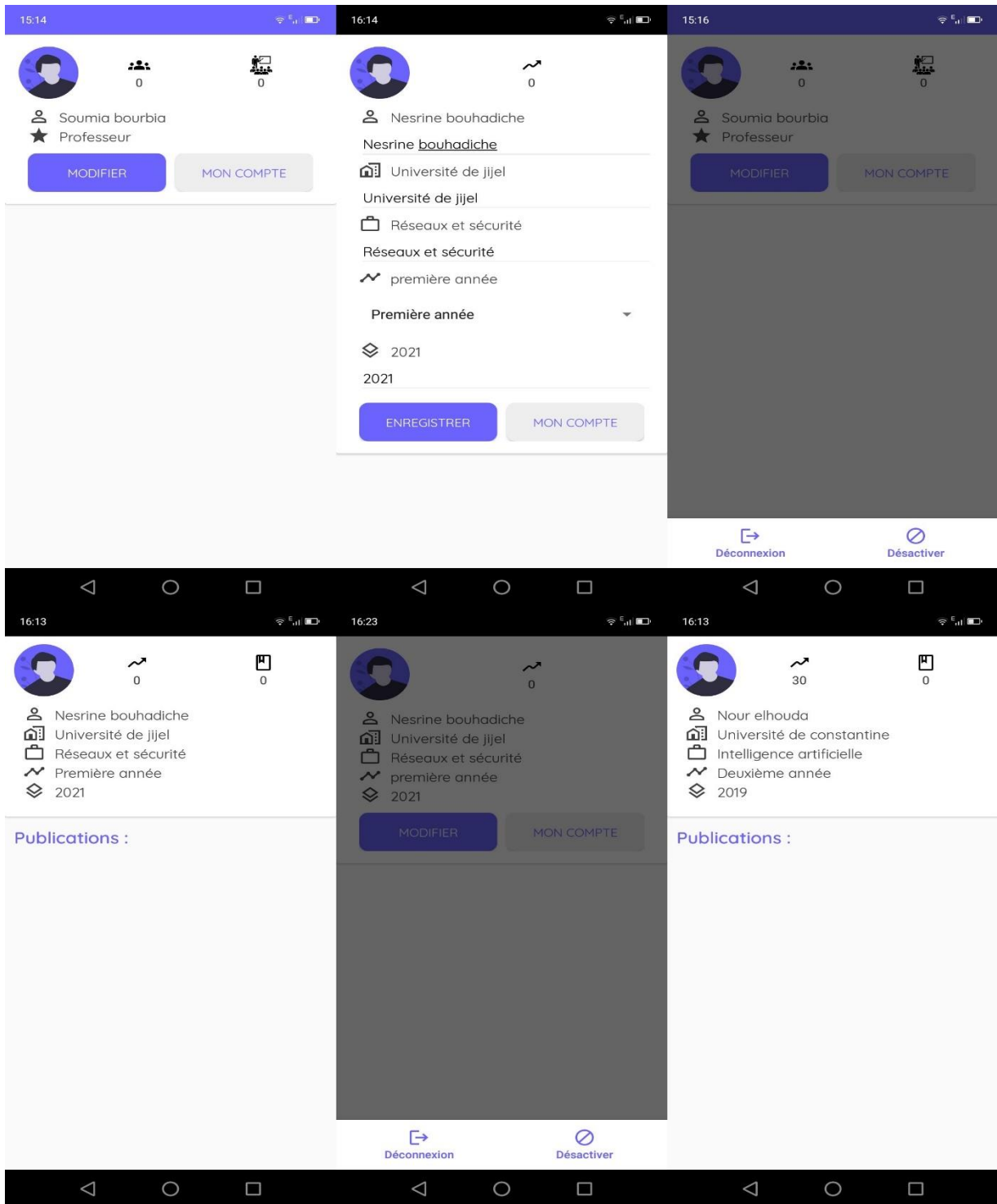
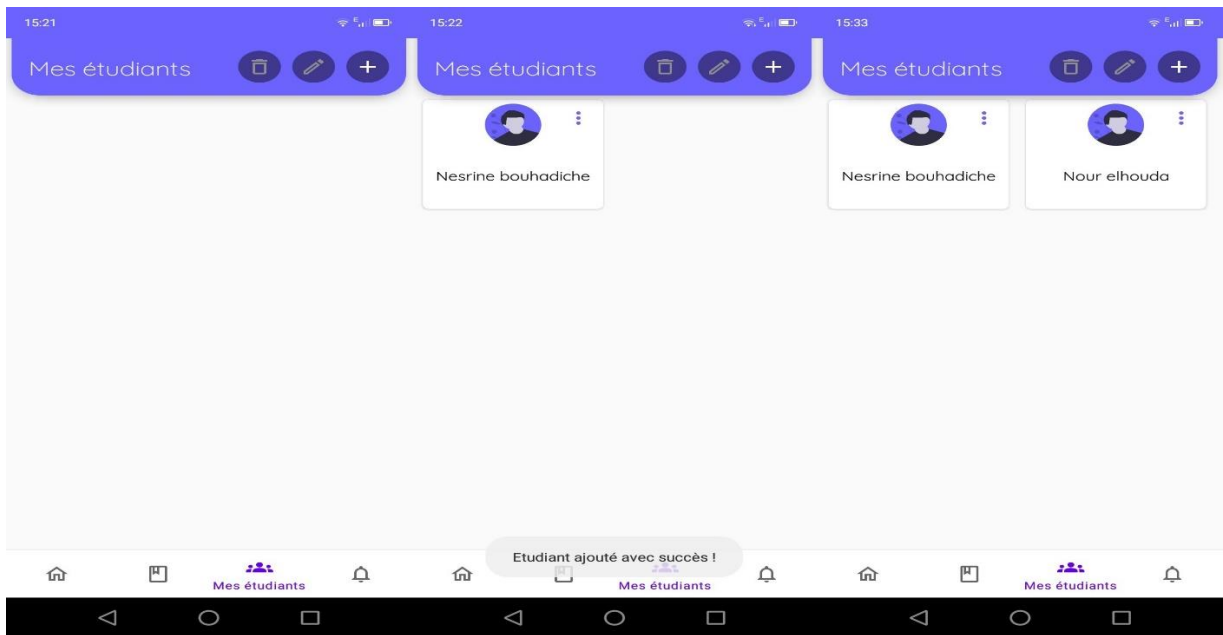


Figure 4.7: Fenêtres profils utilisateurs

6. Fenêtres « gérer liste des doctorants »

a. L'ajoute



–

Figure 4.8: Fenêtres ajoute des doctorants

a. Le blocage et la suppression

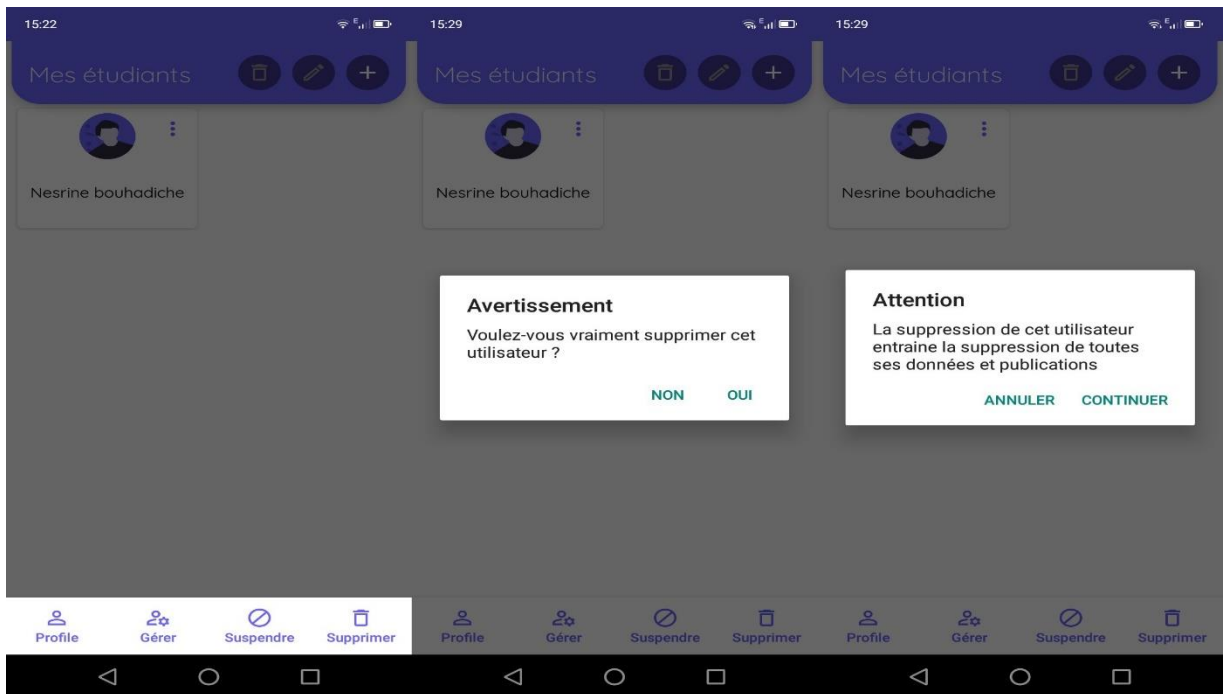


Figure 4.9: Fenêtres suppression des doctorants

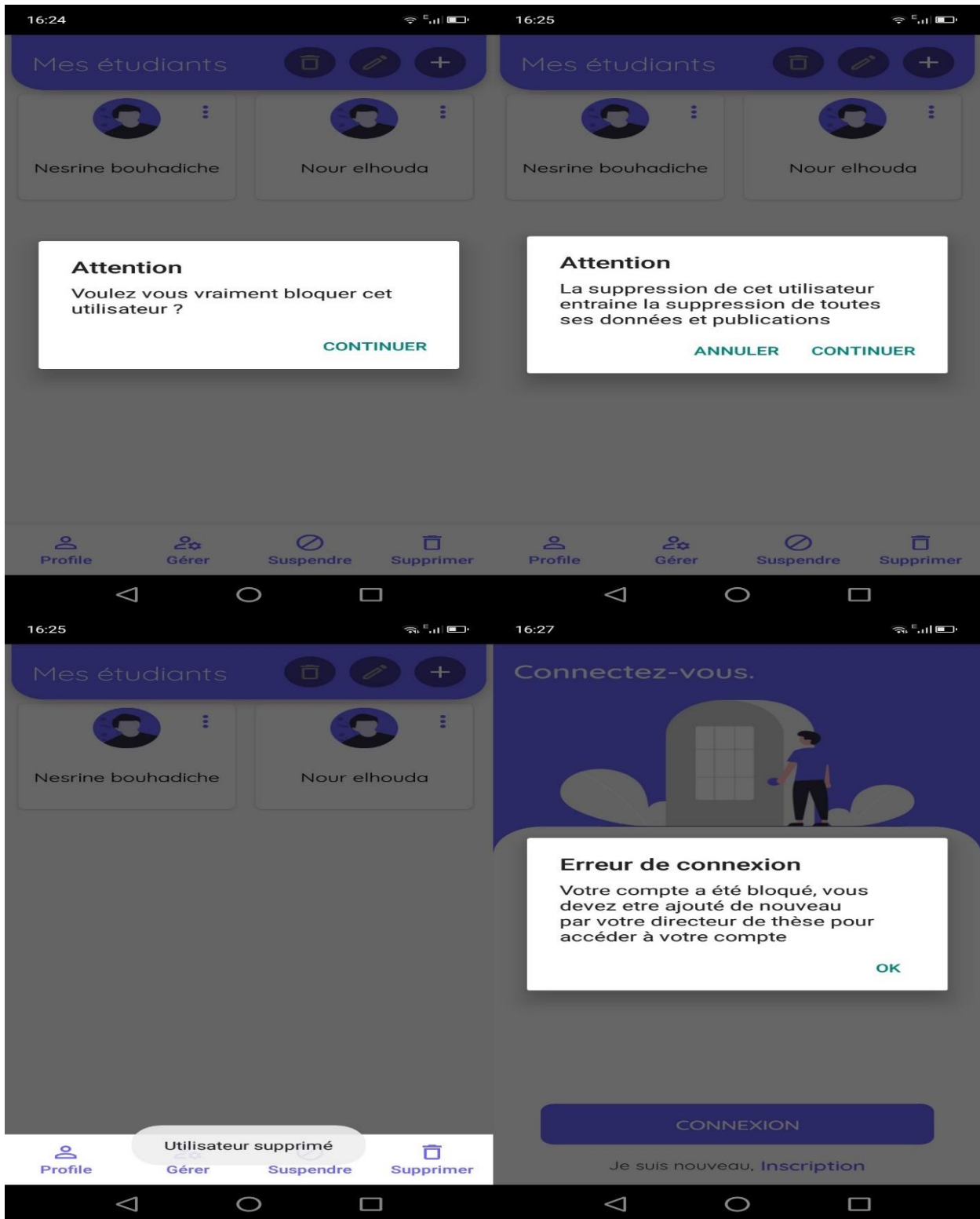


Figure 4.10: Fenêtres blocage des doctorants

7. Fenêtre de messagerie

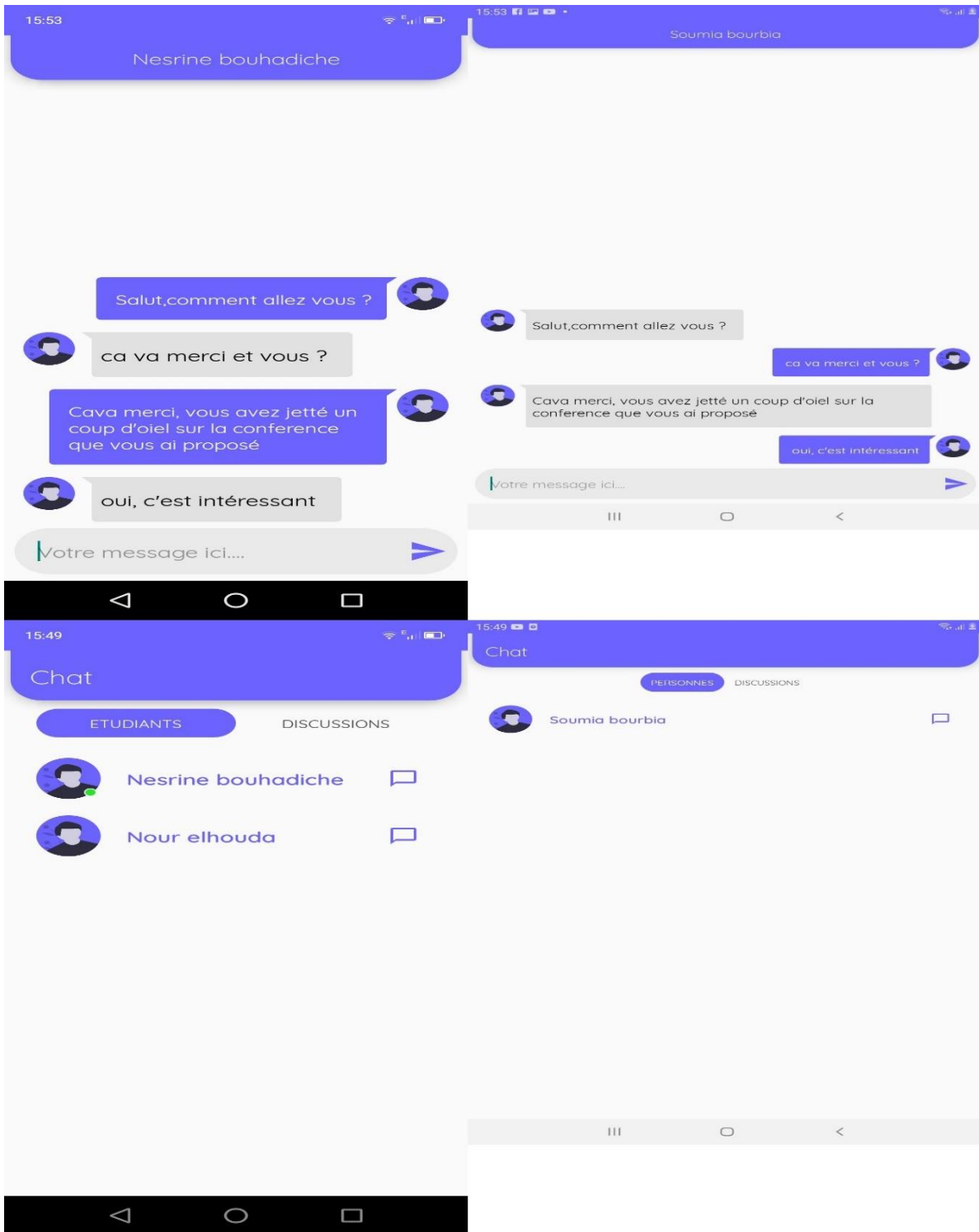


Figure 4.11: Fenêtres messagerie

8. Fenêtres de réunion

a. Lancement

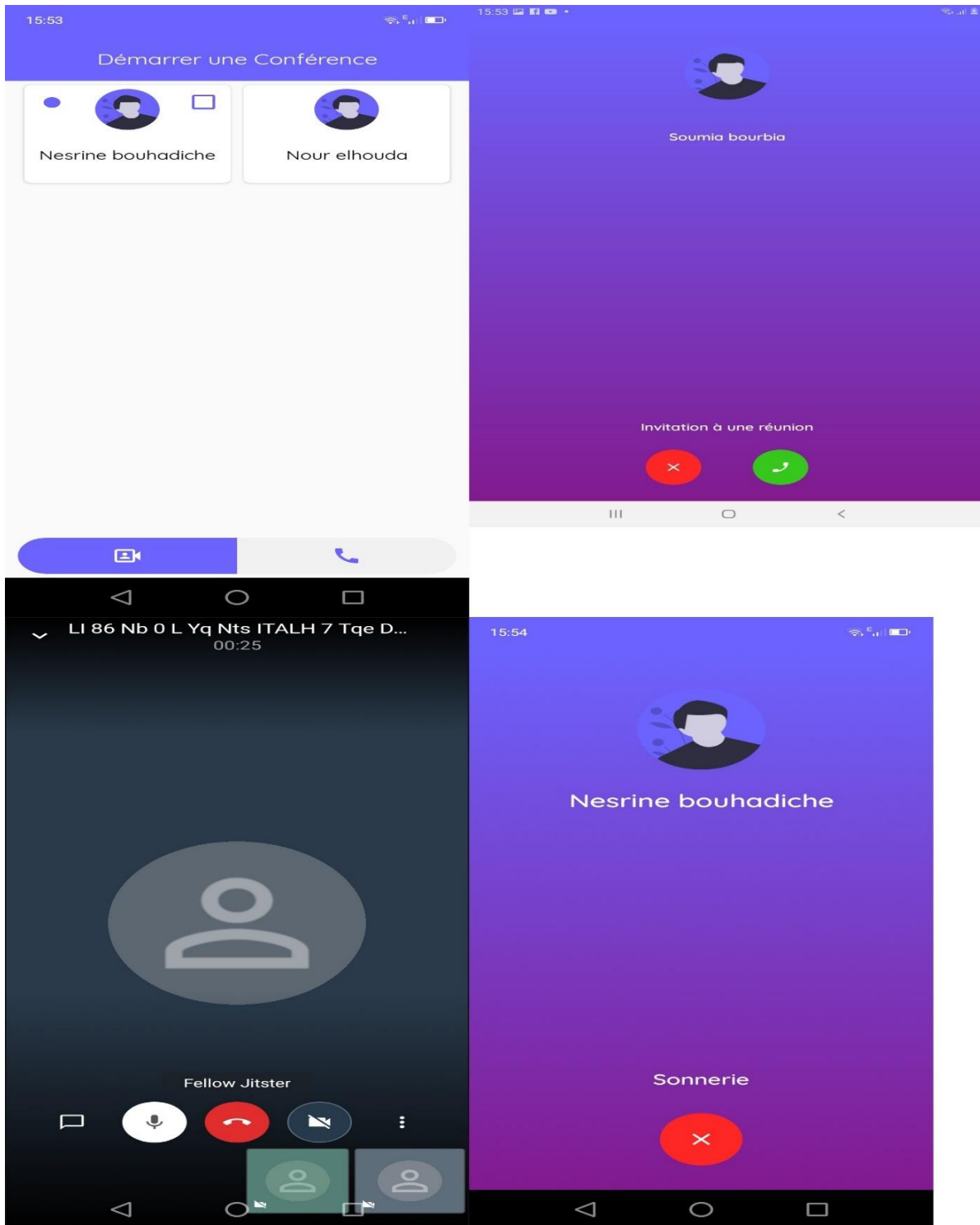


Figure 4.12: Fenêtres lancement de réunion

b. Participation

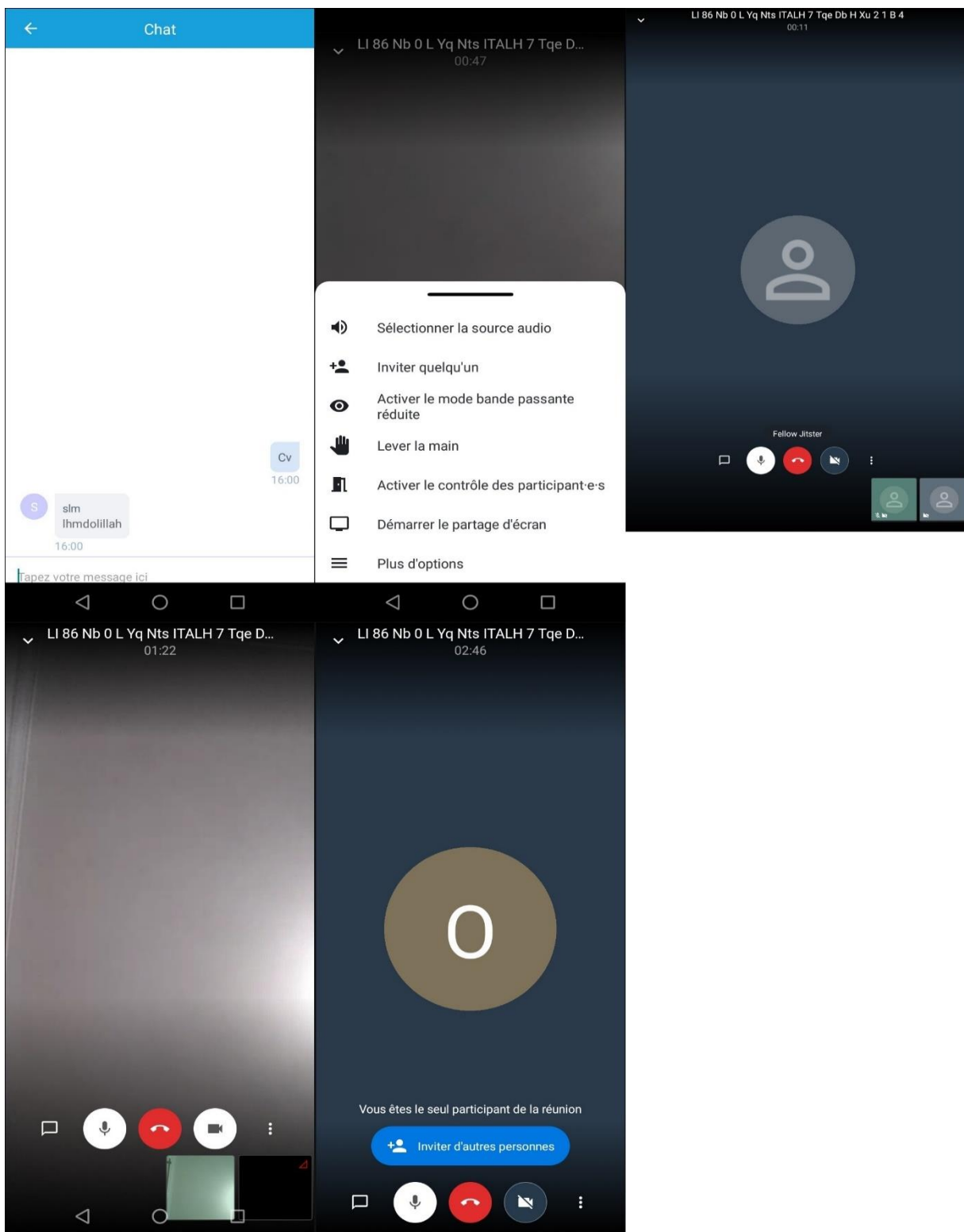


Figure 4.13: Fenêtres participation au réunion

9. Fenêtres pour gérer les formations

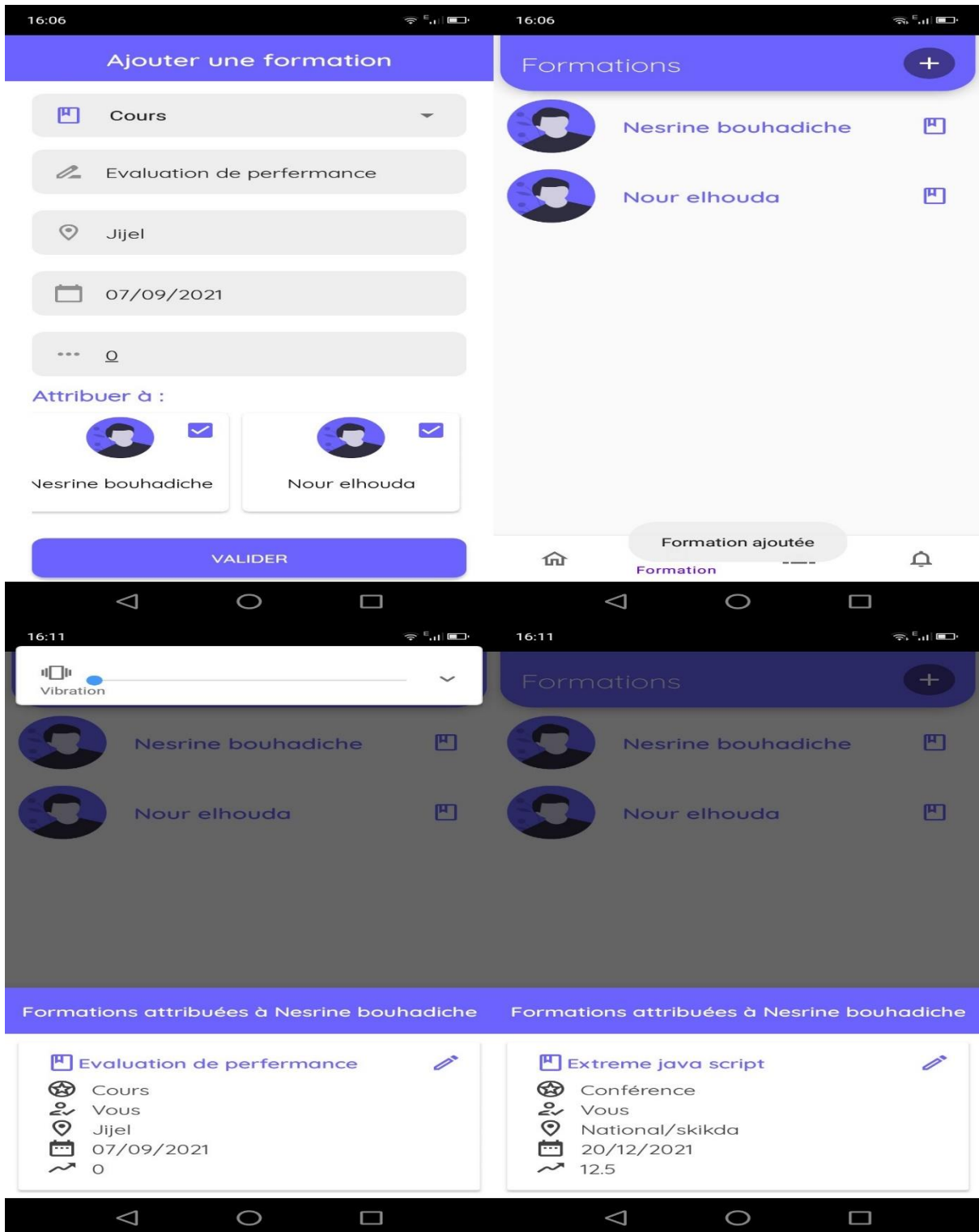


Figure 4.14: Fenêtres l'ajoute des formations

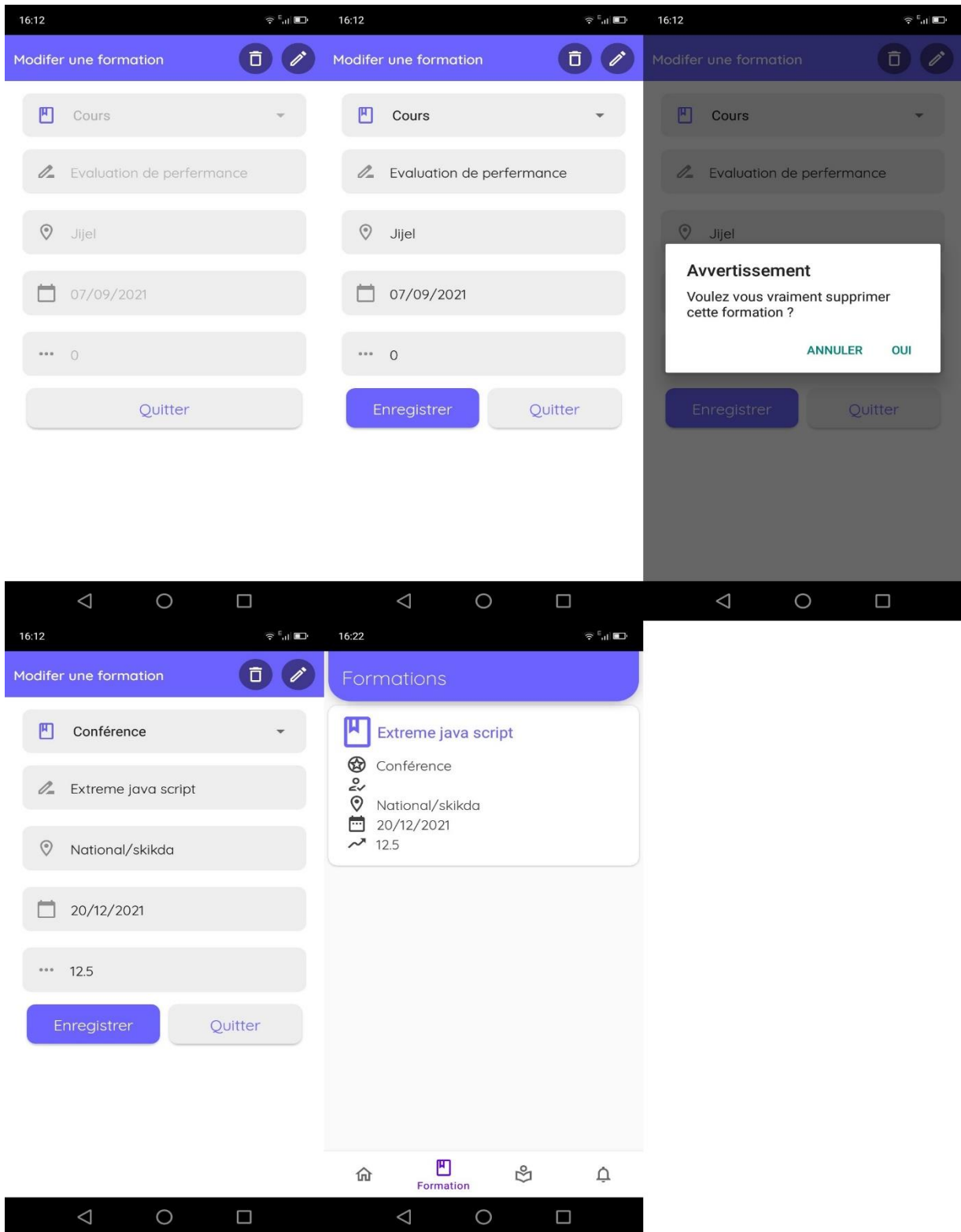


Figure 4.15: Fenêtres modification et suppression des formations

10.Fenêtres pour gérer les publications

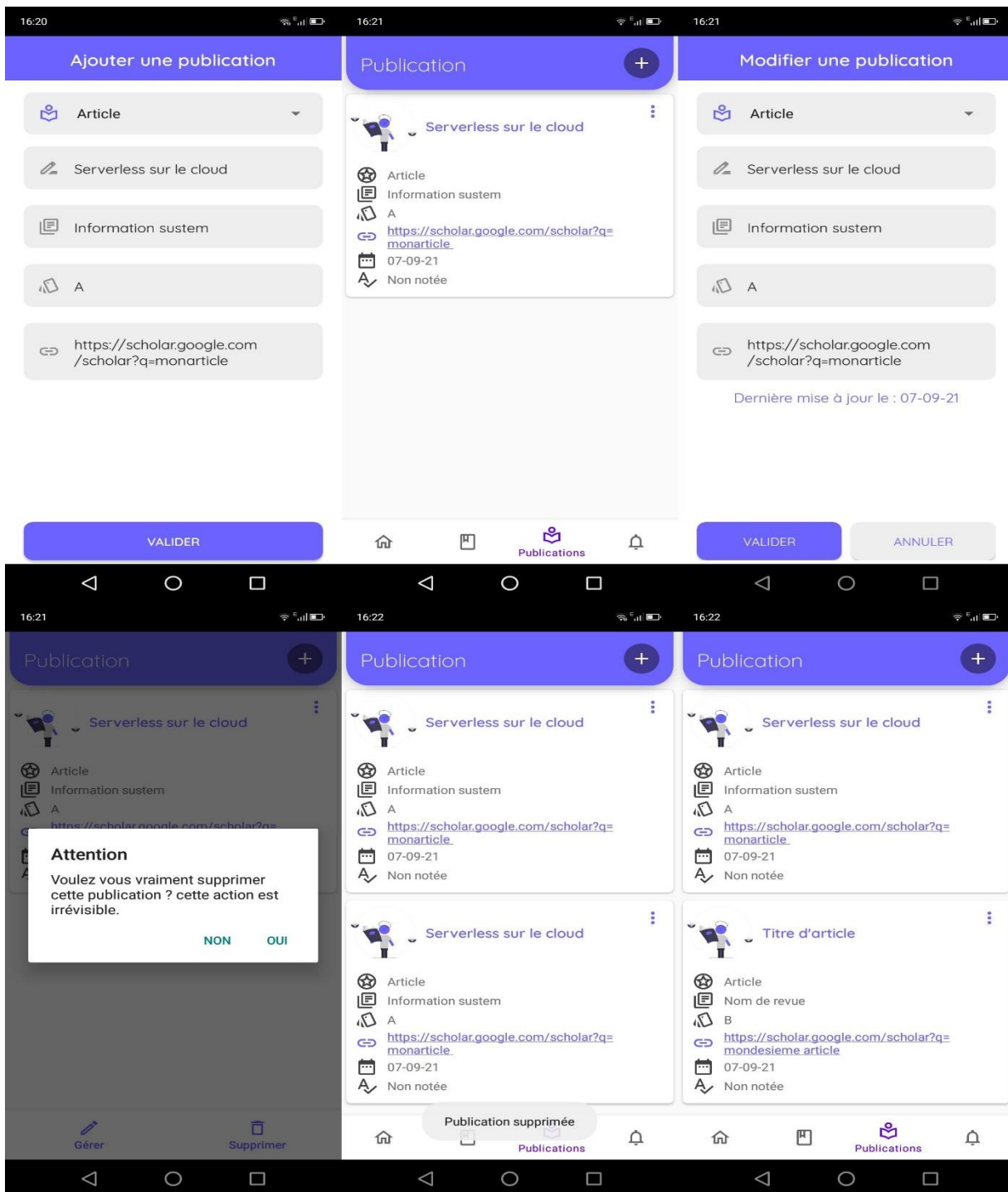


Figure 4.16: Fenêtres gérer publications

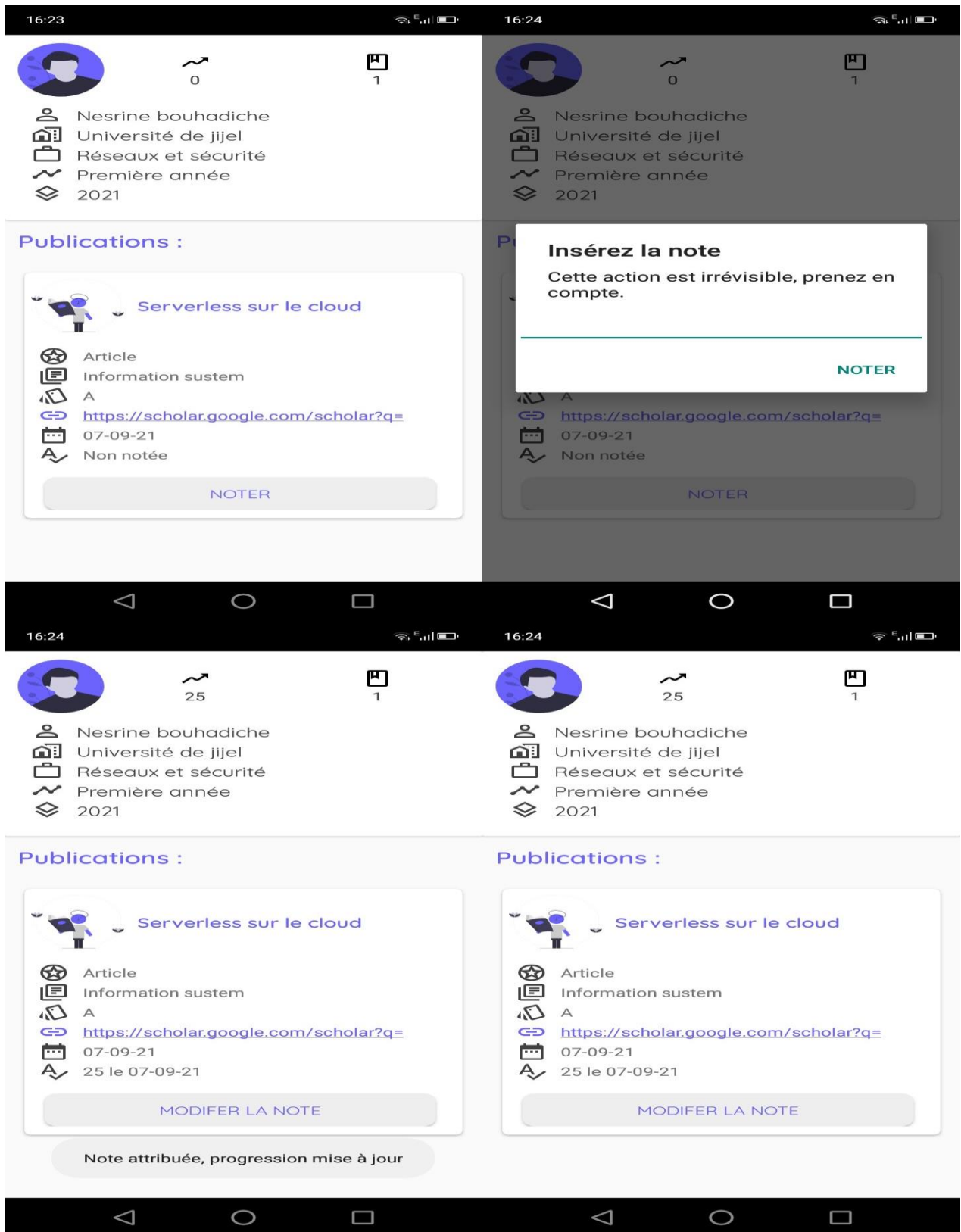


Figure 4.17: Fenêtres noter publications

4.8 Les difficultés rencontrées

Au cours de la réalisation de ce projet, nous avons rencontré des difficultés, dont le suivant :

1. Nous avons utilisé de nouveaux outils que nous n'avons pas utilisés auparavant. Au début, il y avait une certaine ambiguïté sur la façon dont cela fonctionnerait, mais avec le temps, nous nous y sommes habitués.
2. Le manque de documentation de certaines apis qui étaient indispensable au fonctionnement de l'application, comme l'api Jitsi meet.

4.9 Conclusion

Ce dernier chapitre a été consacré à la présentation de la réalisation de notre application. nous avons décrit brièvement les langages de développement, l'environnement matériel et logiciel, les outils et les différentes bibliothèques manipulés. A la fin, nous avons présenté une vue de l'application via quelques interfaces.

Conclusion générale

La pandémie de COVID-19 pose des problèmes aux doctorants et à leurs superviseurs surtout au niveau de processus de suivi des doctorants. La fréquence de la supervision avait diminué pendant la pandémie. Les rencontres de recherche ainsi que les réunions entre communauté scientifique sont devenues une difficulté. Le recours vers des solutions virtuels est donc indispensable. La communauté scientifique a besoin de se doter des moyens nécessaires pour garantir le meilleur encadrement possible des doctorants. Une supervision plus fréquente et l'utilisation d'un éventail diversifié de plateformes de réunion sont utiles.

Notre travail consiste à la réalisation d'une application mobile « PhDmonitoring » pour le suivi à distance des doctorants. L'application permet principalement de faciliter la communication entre directeurs de thèse et leurs doctorants en créant un espace d'échange entre les deux. L'application également offre la possibilité de planifier et d'organiser des réunions à distance.

Le travail réalisé m'a permis d'améliorer et d'enrichir mes compétences dans le développement mobile, j'ai acquis de l'expérience avec un ensemble de langages et technologies comme Java, XML, Firebase, .

Comme perspectives, j'ai l'intention d'ajouter certaines fonctionnalités telles que l'enregistrement des réunions.

Bibliographie

Bibliographie

- [1] Système LMD.
http://univbejaia.dz/portes_ouvertes/index.php?option=com_content&view=article&id=26&Itemid=116
- [2] Descriptif général du système lmd. <https://univ-biskra.dz/index.php/fr/31-formation-superieure/articles2/322-descriptif-general-du-systeme-lmd>
- [3] CARNET DU DOCTORANT. <https://www.univ-boumerdes.dz/recherche/Charte%20pour%20l%27accompagnement%20du%20doctorant%20de%20l%27université.pdf>
- [4] Charte de la thèse doctorat LMD.
<https://view.officeapps.live.com/op/view.aspx?src=http%3A%2F%2Fwww.ensa.dz%2Fwp-content%2Fuploads%2F2016%2F03%2FCHARTE-DE-LA-THESE-doctorat-LMD.docx>
- [5] Les smartphones. <https://cours-informatique-gratuit.fr/cours/les-smartphones/>
- [6] Appareil mobile. https://fr.wikipedia.org/wiki/Appareil_mobile
- [7] <https://www.01net.com/tests/test-du-samsung-galaxy-s20-fe-performant-endurant-et-abordable-ce-smartphone-va-cartonner-6535.html>
- [8] Tablette Tactile. <https://www.tablette-tactile.net>
- [9] Ordinateur / PC Portable Mini Netbook Android. <https://www.123comparer.fr/PC-portable/236948.html>
- [10] Système d'exploitation : définition, traduction et acteurs. <https://www.journaldunet.fr/web-tech/dictionnaire-du-webmastering/1203371-systeme-d-exploitation-definition-traduction-et-acteurs/>
- [11] Mohamed, B. Les Systèmes d'exploitation mobile. <https://fr.slideshare.net/mohamedBouraoui5/les-systmes-dexploitation-mobile>
- [12] la programmation sous android. <http://dspace.univtlemcen.dz/bitstream/112/11054/2/Microsoft%20Word%20-%20-%20chapitre-02-la-programmation-sous-android%20%281%29.pdf>
- [13] Lee, Wei-Ming, Beginning android 4 application Development. John Wiley & Sons, 2012.
- [14] Système d'exploitation Android : avantages et inconvénients. <https://www.dupontfivestar.fr/systeme-dexploitation-android-avantages-et-inconvenients/>
- [15] Android – Architecture. <https://astuces-informatique.com/android-architecture/>
- [16] ARCHITECTURE ANDROID. http://www.josephazar.net/blog/android_arch.php

Bibliographie

- [17] Architecture du système d'exploitation Android. <https://www.editions-eni.fr/open/mediabook.aspx?idR=a1cb1b84d376f88bd4145127075aed6f>
- [18] REMACI, Z, Y., GHITRI, S. Mémoire de fin d'études « Développement d'une application mobile Le jeu « smile » ».
- [19] Application mobile. <https://www.wearecom.fr/dictionnaire/application-mobile/>
- [20] Développement d'applications mobiles au Maroc. <http://www.riadmehdi.net/developpement-dapplications-mobiles-maroc/>
- [21] Applications-mobile. <https://www.taktilcommunication.com/blog/applications-mobile/definition-typologie-applications-mobiles.html>
- [22] Application mobile Vs site mobile : Avantages et inconvénients. <https://www.taktilcommunication.com/blog/applications-mobile/site-mobile-vs-application-mobile-avantages-et-inconvenients.html>
- [23] Mobile app. https://en.wikipedia.org/wiki/Mobile_app
- [24] Les critères indispensables d'une bonne application. <https://fr.yeeply.com/blog/criteres-bonne-application/>
- [25] Picard, Gauthier. "Android™ Programming."
- [26] https://profgra.org/lycee/activite_Android_activities.html
- [27] Eriksson, Hans-Erik, et al. *UML toolkit*. Vol. 1. New York: Wiley, 1998.
- [28] Java : qu'est-ce que c'est. <https://www.futura-sciences.com/tech/definitions/internet-java-485/>
- [29] Les caractéristiques de JAVA. <https://latrach.net/les-caracteristiques-de-java>
- [30] XML (Extensible Markup Language) : définition et description simple. <https://www.journaldunet.fr/web-tech/dictionnaire-du-webmastering/1203615-xml-extensible-markup-language-definition-traduction/>
- [31] Json : définition et présentation de ce format de données. <https://www.journaldunet.fr/web-tech/dictionnaire-du-webmastering/1445308-json-definition-et-presentation-de-ce-format-de-donnees/>
- [32] NoSQL : Tout comprendre sur les bases de données non relationnelles. <https://datascientest.com/nosql>
- [33] DiMarzio, Jerome. *Beginning Android Programming with Android Studio*. John Wiley & Sons, 2016.
- [34] La Structure du projet d'une application Android sous android Studuio. <https://androidtutos.com/structure-basique-dun-projet-android/>
- [35] Développement d'Applications Mobiles Android [Android SDK].<https://esokia.com/fr/expertises/developpement-android>
- [36] What is StarUML. https://documentation.help/StarUML/what_is_staruml.htm

Bibliographie

- [37] A quoi sert Firebase, la plateforme mobile de Google? <https://junto.fr/blog/firebase/>
- [38] Introducing the Jitsi Meet mobile SDK. <https://jitsi.org/blog/introducing-the-jitsi-meet-mobile-sdk/>
- [39] Premiers pas avec le client HTTP Retrofit 2. <https://www.moyens.net/android/premiers-pas-avec-le-client-http-retrofit-2/>
- [40] QR Code Generator API. <https://appsource.microsoft.com/enus/product/webapps/devisssoftware.qrcodeme?tab=overview>
- [41] Android QR Code Scanner Library. <https://androiddvlpr.com/android-qr-code-scanner-library/>
- [42] M. Farinone. Présentation de la Platform Android. 2011. <http://cedric.cnam.fr/farinone/SETMO/seance1PourSETMO.pdf>.
- [43] E.PUYBARET. Les cahiers du programmeur. java 1.4 et 5.0 (3e édition), EYROLLES, 2004.
- [44] Vaish, Gaurav. Getting started with NoSQL. Vol. 2103. Packt Publishing, 2013.
- [45] DiMarzio, Jerome JF. Android™: A programmer's Guide. 2020.

Résumé

Ce mémoire a été rédigé dans le cadre du projet de fin d'études pour l'obtention du diplôme Master en Informatique option : Réseaux et Sécurité de l'université de Jijel. L'objectif de ce projet est de développer une application mobile « PhDmonitoring » visant à faciliter le suivi pédagogique des doctorants. L'application permet aux directeurs de thèses de suivre la progression de ses doctorants, de partager avec eux des formations (cours, séminaires), etc. Et permet aux doctorants d'être encadrés, orientés et suivis en temps réel. La conception de l'application a été réalisée à l'aide du langage UML. Tandis que, la réalisation a été effectuée en utilisant les deux langages JAVA et XML sous Android studio comme environnement de développement. Le service Firebase fournie par google comme système de gestion de base de données NoSQL et un ensemble des bibliothèques comme : Retrofit, QR code scanner, Jitsi meet Android SDK ont été combiné pour former l'application.

Mots clés : Doctorat, suivi, application mobile, Android, Firebase, JAVA.

Abstract

This thesis was written as part of the graduation project to obtain the Master's degree in Computer Science option: Networks and Security at the University of Jijel. The objective of this project is to develop a mobile application "PhDmonitoring" to facilitate the monitoring of doctoral students. The application allows supervisors to monitor the progress of their students, to share courses with them (courses, seminars), etc. It also allows PhD students to be supervised, oriented and followed in real time. The conception of the application has been realized with the help of the UML language. While, the realization has been done using both JAVA and XML languages under Android studio as a development environment. The Firebase service provided by google as a NoSQL database management system and a set of libraries like: Retrofit, QR code scanner, Jitsi meet Android SDK were combined to form the application.

Keywords : PhD, tracking, mobile application, Android, Firebase, JAVA.