

الجمهورية الجزائرية الديمقراطية الشعبية
RÉPUBLIQUE ALGÉRIENNE DÉMOCRATIQUE ET POPULAIRE
وزارة التعليم العالي والبحث العلمي
MINISTÈRE DE L'ENSEIGNEMENT SUPÉRIEUR ET DE LA RECHERCHE
SCIENTIFIQUE
جامعة عمار ثلجي بالأغواط
UNIVERSITE AMAR TELIDJI LAGHOUAT



FACULTE DES SCIENCES
DEPARTEMENT D'INFORMATIQUE

Projet de Fin d'Etudes

Licence

Domaine : Mathématiques et Informatique

Filière : Informatique

Option : Systèmes Informatiques

Par :

BARI Rahil

BENATALLAH Inès

Thème :

**Conception et réalisation d'une application mobile pour :
Cahier de Correspondance Scolaire**

Proposé par : Mr Tahar ALLAOUI

Année universitaire 2019/2020

إهداء

لله الحمد والشكر والمنة، خلقي ورزقني وعلمني وهداني وأنعم علي بأعظم نبي وأرحم أب وأحلى أم رحمها الله وغفر لها، فله الحمد على ذلك وله الشكر كذلك.

إلى أغلى وأحلى مخلوق في الوجود - أمي رحمة الله عليك أهديك فرحة تخرجي،

إلى من وضعتني على درب الحياة، رعتني، ربتي، علمتني، أدبتني وصنعتني تحت نظرها وسقتني بحبها حتى اكتمل عودي، إلى من وضع الحق سبحانه وتعالى الجنة تحت قدميها ووقرها في كتابه الكريم، إلى من ضحت بالنفس والنفيس من أجلي وكانت تنتظر يوم تخرجي بشوق وترى فيه أحلاهما وأمالها ونتيجة جهدها ونعبتها وسهرها، إلى من كان كل دُعائها لي شهادةً وغيتاً ... إليك سيدتي اهدي فرحتي، فأنت في نظري سيدة نساء هذا الزمان، إليك أمي يا ندى روعي وبلسمها، إليك أمي أيتها الروح الطاهرة يامن رحلت بلا عودة، أهديك تخرجي وكل عمل صالح نافع بَرَزَ مِنِّي عسى أن يُكْتَبَ من نور في صحائفك الطاهرة، تركتني في مُنتَصَفِ الطريق وحيدةً نائهةً فسلكته بِحُسْنِ تربيته وحَفِظْتَه فيه بِدُعَائِكَ الصالح لي، كَمْ هُوَ قَلْبِي مَحْزُونٌ يَتَعَذَّبُ وَيَقْطَعُ شَوْقاً إِلَى رُؤْيَاكَ بجانبي، تُبَارِكِينَ تَخْرُجِي ونجائي وذمومُ الفرحه والنجاح تتناثر كالذَرَرِ على وَجْهَتَايَ، لَكِنْ قَدَّرَ اللهُ ماضِي وإرادته نافذة، رضينا بما قضى وقدر ولا نقول إلا ما يُرضي ربنا. رَحِمَكَ اللهُ ماما، غَفَرَ اللهُ لكَ ماما، رَفَعَ اللهُ قَدْرَكَ ماما، جَمَعَنِي اللهُ وَإِيَّاكَ ماما في الفردوس الأعلى من الجنة مع خديجة وفاطمة وعائشة رضي الله عنهن.

إلى أبي وسندي ومن له بعد الله وأمي كل الفضل علي، أَنَاَ فكري وعطر سبرتي وسهر وقاسي وركب الصعاب من أجل أن يراني كالنجمة ساطعة في سماء النجاح لا أتخلف عن ركب الناجحين، اهدي إليك هذا العمل المتواضع.

إلى خالي الغالي صبيح محمد العيد،

إلى خالتي نصيرة ... لا أجد في قاموسي كلمات تفي بفضلها، خاصة ابتنتها سعاد، ساندتني، شجعتني، نصحتني وكلما خبت شُعلة قُؤَايَ أَجْدُهَا بجانبي تبعث في روح الأمل وتحثني على الصبر والمثابرة في العمل،

إلى ذو القلب الطيب الرحيم عمي صالح،

إلى الناصحتان المهمتان بشأني عمتي رقية وعمتي زينب،

إلى عمتي أم الخير،

إلى بقية أعمامي وعماتي وأخوالي وخالاتي، أبنائهم وبناتهم، إلى جميع أقاربي دون استثناء،

إلى أختي وصديقتي رفيقة المسيرة الدراسية شروق،

إلى صديقتي وزميلتي العزيزة هاجر، إلى زميلتي إيناس وشريكتي في مشروع التخرج،

إلى زميلاتي وزملائي في الدراسة،

إلى أساتذتي الأفاضل وأستاذاتي الفضليات، سهروا وتعبوا وبحثوا وحَضَرُوا الدروس من أجل أن يُمَلَأَ وعائي بالعلم والنور، شكرا لكم.

كامل الشكر والتقدير موصول للأستاذ المشرف على مشروع التخرج، الشكر موصول كذلك إلى رئيس لجنة المناقشة وأعضائها الموقرين وإلى كل الأساتذة الكرام، كما نُشِيدُ بالمسؤولين الإداريين وطاقمهم وإلى كل من كان سببا في نجاحنا.

إلى الجميع أهدي فرحة تخرجي ونجاعي.

راحيل باري

DEDICACES

A mes très chères Parents " Souad & Abdallah "

Aucune dédicace ne saurait être assez éloquente pour exprimer l'amour, l'estime, le dévouement et le respect que vous méritez pour tous vos sacrifices ,Votre soutien moral et matériel, vos encouragements, m'ont permis de réussir mes études. Que ce travail soit témoignage de ma reconnaissance et de mon amour sincère. Puisse Dieu, le tout puissant, vous préserver et vous accorder santé, longue vie et bonheur.

A mes très chères Sœurs " Amira " et " Joumana "
a mon cher Frère " Mouhamed "

En témoignage du respect, de l'amour et de l'affection que je porte pour vous. Je vous dédie ce travail avec tous mes vœux de bonheur, de santé et de réussite, surtout a toi *Amira* qui ma soutenu matériellement et moralement.

A ma chère Sœur " Sarah " et a mon beau Frère
" AbdelRahman "

Que dieu vous garde si tendre et aiment l'un envers l'autre et vous bénissent.

A mon patron "Mr Boumedienne AbdelMadjid "

Ce travail et le fuit de votre soutien vous m'avez redonné le courage et la force pour réaliser ce modeste travail que je vous dédie avec toute ma gratitude et mon respect.

Tues.

REMERCIEMENT

Nous tenons tout d'abord à remercier Allah notre seigneur le tout puissant et miséricordieux, qui nous a donné force et patience pour accomplir ce modeste travail. Ce mémoire n'aurait pas vu le jour sans la contribution et le concours de plusieurs personnes à qui nous voulions témoigner toute notre reconnaissance. Nous tenons à exprimer notre gratitude et nos vifs remerciements à l'encadreur de notre projet Mr Tahar ALLAOUI qui s'est montré patient durant toute la période du projet, il nous a orienté, aidé et encouragé pour bien mener notre travail à bon port malgré la situation exceptionnelle que nous avons traversés durant la pandémie du Covid19. Notre déférence s'adresse au corps professoral et administratif du département mathématiques et informatique de l'université Amar TELIDJI de Laghouat qui a contribué à la réussite de nos études universitaires et pour les plus belles années d'études qu'on a passé avec eux. Sans oublier bien sûr de remercier vivement le président et la membre du jury qui nous ont honorées d'accepter d'examiner notre travail et de l'enrichir par leurs propositions et conseils. Nous remercions profondément, nos parents, nos familles, nos amies et tous ceux qui nous ont soutenu de près ou de loin pour la réalisation de ce mémoire.

TABLE DES MATIÈRES

Dédicaces	i
Dédicaces	ii
Résumé	1
Introduction générale	1
1 généralités	2
1.1 Technologies mobiles : Vue générale	2
1.2 Introduction	2
1.3 Applications mobiles	2
1.3.1 Définition	2
1.3.2 Les types d'applications mobiles	3
1.3.3 Caractéristiques des applications mobiles	3
1.4 Plate-forme Android	3
1.4.1 Versions	5
1.4.2 Architecture	6
1.4.3 Avantages du système Android	7
1.5 Application Android	7
1.5.1 Présentation	7
1.5.2 Composant d'une application Android	7
1.6 Base de Données	8
1.7 Conclusion	8

2	analyse et conception	9
2.1	Introduction	9
2.2	Objectif de notre application	9
2.3	Langage de modélisation unifié	10
2.3.1	Définition	10
2.3.2	Les différents diagrammes UML	10
2.3.3	Pourquoi l'UML ?	13
2.4	Spécification des besoins	13
2.4.1	Besoins fonctionnels	13
2.4.2	Besoins non fonctionnels	14
2.4.3	Besoins techniques	14
2.5	Analyse des besoins	14
2.5.1	Identification des acteurs	14
2.5.2	Diagramme des cas d'utilisation	17
2.6	Conception de base de données	18
2.6.1	Diagramme de classes	18
2.6.2	Diagrammes de séquences	19
2.7	Conclusion	22
3	Réalisation et Développement	23
3.1	Introduction	23
3.2	Environnement et outils du travail	23
3.2.1	Environnement matériel	23
3.2.2	Environnement Logiciel	24
3.2.3	AndroidStudio	24
3.2.4	SDK	24
3.2.5	JDK	24
3.2.6	AVD	24
3.2.7	Firebase	24
3.3	Langage de developpement	26
3.3.1	Java	26
3.3.2	XML	26
3.4	Présentation des interfaces graphiques	26
3.4.1	Logo de l'application	26
3.4.2	Splash Screen	27
3.4.3	Interface de choix de l'utilisateur	28

3.4.4	Interface d'authentification	31
3.4.5	Interface principale de l'administrateur	34
3.4.6	Interface principale d'utilisateur	35
3.4.7	Navigation Drawer	37
3.4.8	Interface pour marquer l'absence	38
3.4.9	Interface d'évaluation	39
3.4.10	Interface Ajout élève	40
3.5	Conclusion	41
Conclusion générale et perspectives		42
Bibliographie		44

1.1	Logo du système Android	4
1.2	composants de la plateforme Android	6
2.1	Logo de l'UML	10
2.2	Hiérarchie des diagrammes UML.[1]	12
2.3	Diagramme des cas d'utilisation globale.	17
2.4	Diagramme de classes.	18
2.5	Diagramme de séquence «Authentification»	19
2.6	Diagramme de séquence «Ajout»	20
2.7	Diagramme de séquence «Evaluation».	21
2.8	Diagramme de séquence «Consultation»	22
3.1	Services de Firebase	25
3.2	Logo de l'application	26
3.3	Splash Screen	27
3.4	Choix Admin	28
3.5	Choix Enseignant	29
3.6	Choix Parent	30
3.7	Authentification Admin	31
3.8	Authentification Enseignant	32
3.9	Authentification Parent	33
3.10	Interface principale de l'administrateur	34
3.11	Interface principale d'utilisateur-Enseignant-	35
3.12	Interface principale d'utilisateur-Parent-	36
3.13	Navigation Drawer	37

3.14 Interface pour marquer l'absence	38
3.15 Interface d'évaluation	39
3.16 Interface Ajout élève	40

LISTE DES TABLEAUX

1.1	Les versions d'Android	5
2.1	Identification de l'acteur "Administrateur"	15
2.2	Identification de l'acteur "Enseignant"	16
2.3	Identification de l'acteur "Parent"	16
3.1	présentation des outils matériels	23

LISTE DES ABRÉVIATIONS

- API : Application Programming Interface.
- AVD : Android Vistual Device.
- BDD : Base de Données.
- IDE : Integrated Development Environment.
- JDK : Java Development Kit.
- NoSQL : Not only SQL.
- SDK : Software Développement Kit.
- UML : Unified Modeling Language.
- XML : Extensible Markup Language.

Notre application vise à numériser le cahier de correspondance scolaire qui n'est autre chose qu'un livret ou outil de communication entre l'établissement scolaire et les parents, ces derniers doivent le consulter de manière régulière pour contrôler retards, absences ou autres remarques qui y sont notées, et recense toutes les informations au quotidien de l'élève au sein son l'établissement.

Le projet en question appelé « ClassBook » est l'étude, la conception et la mise en œuvre d'un système informatique sous Android pour Smartphone qui permet au parent de suivre de près et en temps réel la situation pédagogique quotidienne de son enfant et facilite aussi le lien entre l'administration de l'école et l'enseignant d'une part et le parent de l'élève d'autre part, il a pour objectif de responsabiliser certains parents qui ont tendance à laisser un peu trop de libertés à leurs enfants.

La numérisation du cahier de correspondance en question n'est pas encore utilisé à grande échelle dans les pays développés, chez nous en Algérie le suivie de l'élève sur le web est totalement absent. Dans ce mémoire nous avons proposés une conception UML, l'implémentation a été réalisée avec le langage java sous l'environnement Android Studio.

Avec l'indispensabilité de la communication et de l'information dans notre vie quotidienne et grâce à la généralisation des téléphones portables tactiles à écrans large ainsi qu'au développement des logiciels et des réseaux, la recherche d'information est accessible du bout des doigts par l'intermédiaire de téléphone ou pas à tout moment et en tous lieux. C'est important pour tout individu de se tenir au courant de ce qui se passe autour de lui et d'avoir accès aux informations dont il a besoin, quand il en a besoin facilement et plus rapidement.

Les usages mobiles ont évolués et l'application s'impose comme élément incontournable dans la réponse au besoin des utilisateurs ,elle a une rapidité d'accès à l'information ainsi que la possibilité de profiter de plusieurs fonctionnalités novatrices, ludique et intuitives qui repoussent les limites de l'expérience utilisateur comme : l'appareil photo, vidéo, le GPS... en plus elles offrent une interface plus spécifique et plus harmonieuse qui les rendre plus aisé que les sites mobiles dans leur utilisation. Pour cette raison l'idée de développement une application devient utile et conviviale.

Dans le domaine de l'éducation la communication entre les parents et les professeurs est essentielle pour le succès des enfants dans l'école pourtant, elle n'est pas toujours évidente, les outils sous la disposition des écoles sont peu pratiques et non sécurisés.

Il s'agit au cours de notre projet d'étudier et de développer une application sous Android comme une solution simple instantané et sécurisé qui réinvente la communication entre les parents et les enseignants en quelque clics.

Le présent mémoire est composé de trois chapitres :

- Le premier chapitre présente les différentes notions utilisées dans le développement des applications mobiles à savoir, le système d'exploitation Android, et les bases de données.
- Le deuxième chapitre expose les objectifs ainsi que la conception UML réalisée.
- Le troisième chapitre clôture ce mémoire dans lequel nous présentons la description de l'application développée, l'implémentation et la conclusion générale.

CHAPITRE 1

GÉNÉRALITÉS

1.1 Technologies mobiles : Vue générale

1.2 Introduction

La technologie mobile est un sujet d'actualité au sein de nombreux secteurs d'activité, pas simplement un terme à la mode. Il est extrêmement pertinent pour le domaine de développement d'application. Ce chapitre présente les différentes notions de l'environnement Android et ses fonctionnalités ainsi que les bases de données.

1.3 Applications mobiles

1.3.1 Définition

Une application mobile est une application embarquée qui est sensé s'exécuter sur une plateforme mobile à l'instar des Smartphones et les tablettes. L'application doit être téléchargée par l'utilisateur pour devenir exécutable à partir du système d'exploitation du téléphone exactement comme un logiciel ordinaire qui s'installe sur un ordinateur classique. Les applications mobiles ne sont pas transférables d'un système d'exploitation à l'autre en raison du langage informatique utilisé : si vous téléchargez une application Android, vous ne pourrez l'utiliser que sur un téléphone portable Android.

Elles permettent généralement un accès plus confortable et plus efficace à des sites ou services accessibles par ailleurs en versions mobile ou web.

1.3.2 Les types d'applications mobiles

Techniquement parlant, il y a trois types d'applications mobiles que tout utilisateur peut rencontrer :

1. **Applications mobiles web** : Toute application conçue avec HTML et CSS de plus opérationnelle sur navigateur internet pour un Smartphone est appelée application web. Peu importe la marque de votre Smartphone, vous pouvez accéder à l'application web par le biais de son navigateur et donc vous n'avez pas besoin de la télécharger. Vu qu'elle ne tient pas en compte les divergences persistantes entre les systèmes d'exploitations et les marques de Smartphone.
2. **Applications mobiles natives** : Il s'agit d'application conçue pour une grande partie de systèmes d'exploitation fiables par les Smartphones en se référant à un langage particulier à chacun d'eux. Ce mode d'application est accessible seulement sur les plateformes d'applications suivant ses particularités et ses formules. Ces plate-formes retirent 25 du prix de vente pour une application native payante.
3. **Applications mobiles hybrides** : Il s'agit d'une application mobile qui fusionne entre les caractéristiques de l'application web (développement en HTML 5) et celles de l'application native. De cette manière, l'application mobile sera accessible sur toutes les plateformes d'application.

1.3.3 Caractéristiques des applications mobiles

Pour dire qu'une application mobile est réussie, elle doit avoir certains critères très importants qui sont les suivants :

1. Rapidité et fluidité, un premier critère pour une application réussie ;
2. Le poids de l'application mobile, un facteur important pour les utilisateurs ;
3. L'ergonomie, un critère pour fidéliser l'utilisateur ;
4. Une bonne qualité d'affichage, un critère pour séduire l'utilisateur ;
5. Maintenance et évolutivité, la clé pour un bon suivi.

SubsectionMaintenance et évolutivité, la clé pour un bon suivi.

1.4 Plate-forme Android

Android est un système d'exploitation mobile apparu sur le marché en 2008 basé sur le noyau Linux et développé par Google. Le système a été conçu pour les Smartphones et tablettes

tactiles, les objets connectés et ordinateurs comme les télévisions, les voitures, les ordinateurs et les smartwatch.

L'interface Android est visuellement chargée et apporte une latitude plus importante à l'utilisateur pour personnaliser son interface comme par exemple la possibilité de remplacer les applications natives. En 2015, Android est le système d'exploitation le plus utilisé dans le monde avec plus de 80 % de parts de marché dans les smartphones.



FIGURE 1.1 – Logo du système Android

1.4.1 Versions

Le tableau ci-dessous montre les versions d'Android ainsi que leur date de lancement

TABLE 1.1 – Les versions d'Android

Nom	Version	Date lancement	API
Apple pie	1.0	23 septembre 2008	1
Bananas split	1.1	09 février 2009	2
Cupcake	1.5	27 avril 2009	3
Donut	1.6	15 septembre 2009	4
Eclair	2.0-2.1	26 octobre 2009	5-7
Froyo	2.2-2.2.3	20 mai 2010	8
Gingerbread	2.3-2.3.7	06 décembre 2010	9-10
Honeycomb	3.0-3.2.6	22 février 2011	11-13
Ice Cream Sandwich	4.0-4.0.4	18 octobre 2011	14-15
Jelly Bean	4.1-4.3.1	09 juillet 2012	16-18
Kit Kat	4.4-4.4.4	31 octobre 2013	19-20
Lollipop	5.0-5.1.1	12 novembre 2014	21-22
Marshmallow	6.0-6.0.1	05 octobre 2015	23
Nougat	7.0-7.1.2	22 août 2016	24-25
Oreo	8.0-8.1	21 août 2017	26-27
Pie	9	06 août 2018	28
Android10	10	03 septembre 2019	29

1.4.2 Architecture

La plate-forme Android est composée de quatre couches, elle est basée sur un noyau (kernel) linux 2.6.xx, au-dessus du kernel on retrouve le hardware abstraction layer, qui permet de séparer la plateforme logique du matériel.

Au-dessus de cette couche d'abstraction, on retrouve les bibliothèques (librairies) C/C++ utilisées par nombreux composants de la plateforme Android. Ensuite, on retrouve l'Android Runtime, cette couche contient les librairies cœurs du Frame-work ainsi que la machine virtuelle exécutant les applications.

Au-dessus la couche "Android Runtime" et des bibliothèques cœurs on retrouve le Frame-work permettant au développeur de créer des applications extrêmement riches et innovantes.

Enfin la dernière couche est la couche application.

Enfin la dernière couche est la couche application.

Le schéma suivant illustre les principaux composants de la plate-forme Android que l'on a expliqué brièvement.

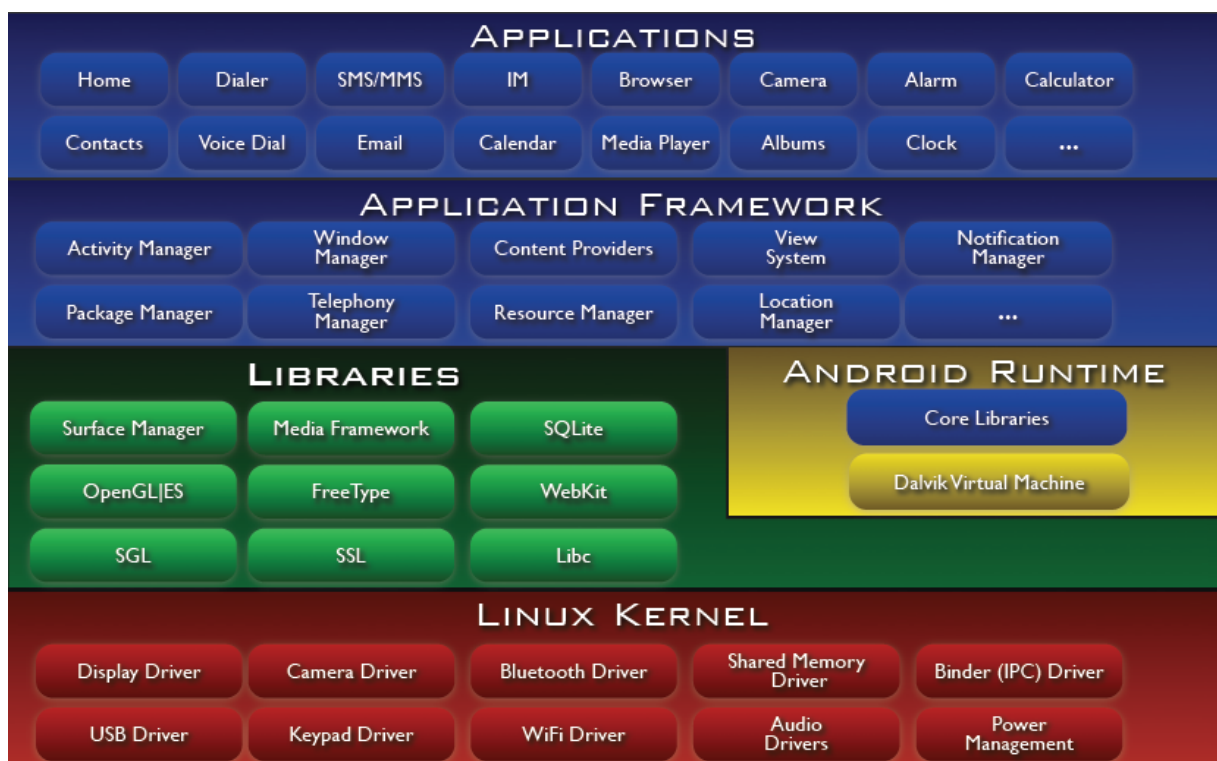


FIGURE 1.2 – composants de la plateforme Android

1.4.3 Avantages du système Android

Les plus grandes marques tels que HTC, Samsung, LG, ou encore Sony Ericsson ont été conquises par ses performances. Le Système d'exploitation Android équipe actuellement beaucoup de mobiles dans le monde. Son code source est accessible. Il est donc open-source, contrairement aux autres systèmes des grandes enseignes telles qu'Apple ou Microsoft et compatible avec des langages de programmation répandus (Java Development Kit).

Bugroid est la mascotte de ce système d'exploitation, ce robot vert a conquis le monde car ce système d'exploitation a l'avantage de gérer plusieurs tâches à la fois. Ainsi, différentes applications peuvent être lancées en même temps sans craindre le bug de l'appareil, il permet aussi aux utilisateurs d'avoir des raccourcis sur leurs écrans d'accueils et bénéficie également de récentes mises à jour pour améliorer encore plus ses performances.

1.5 Application Android

1.5.1 Présentation

C'est une application mobile développée pour les Smartphones utilisant le système Android. Spécifiquement les applications Android sont de nature très variables : jeux, mobile commerce, utilitaire, service d'information... mais sa construction est à peu près similaire.

1.5.2 Composant d'une application Android

Se sont les éléments essentiels d'une application Android. Ces composants sont lâchement couplés par le fichier de **AndroidManifest.xml** qui décrit chaque composant de l'application et comment ils interagissent. Il existe quatre composants principaux.

1. **Activités (Activity)** : c'est la partie de l'application qui présente la vue pour l'utilisateur, manipule l'interaction de l'utilisateur avec l'écran du Smartphone elle est implémentée sous forme de sous-classe de la classe d'activité.
2. **Services** : Il s'agit d'un composant qui s'exécute en arrière-plan pour effectuer des opérations à long terme. Ils traitent les tâches de fond associées à une application.
3. **Récepteurs de diffusions (Broadcast Receivers)** : Ils gèrent la communication entre le système d'exploitation Android et les applications.
4. **fournisseurs de contenu (ContentProvider)** : Un fournisseur permet de fournir des données d'une application à d'autres sur demande. De telles requêtes sont traitées par les méthodes de la classe ContentResolver. Les données peuvent être stockées dans le système de fichiers, la base de données ou ailleurs.

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

- * **Fragment** : Il s'agit d'une partie de l'interface utilisateur dans une activité.
- * **Views** : Les éléments UI qui sont dessinés à l'écran, y compris les boutons, les listes des formulaires, etc...
- * **Layouts** : Afficher les hiérarchies qui contrôlent le format de l'écran et l'apparence des vues.
- * **Intent** : Permet d'envoyer un message pour un composant externe sans le nommer explicitement.
- * **Resources** : Il s'agit des éléments externe comme les images les constantes et les chaînes.
- * **Manifest** : Fichier de configuration de l'application permet de lancer les activités et de les ordonner.

1.6 Base de Données

Une base de données (que nous nommerons BDD par commodité) est une collection d'informations organisées afin d'être facilement consultables, gérables et mise à jour. Au sein d'une BDD, les données sont organisées en lignes, colonnes et tableaux. Elles sont indexées afin de pouvoir facilement trouver les informations recherchées à l'aide d'un logiciel informatique.

Dans le cas d'une grande BDD, les multiples utilisateurs doivent être en mesure de manipuler les informations qu'elle contient rapidement et n'importe quand, il est nécessaire que les données en provenance de plusieurs fichiers puissent être liées. C'est pourquoi différents types de bases de données ont été développés pour répondre à ces exigences : orientée texte, hiérarchique, réseau, relationnelle, orientée objet... etc.

Pour les larges ensembles de données distribuées les bases de données NoSQL sont utiles, une nouvelle technologie prometteuse parmi les SGBD non relationnels très utilisés par les développeurs d'applications mobile est :

1.7 Conclusion

Dans ce chapitre, nous avons mis le point sur les applications mobiles et leur types ensuite on a focalisé sur le système d'exploitation Android, car c'est ce dernier qu'on utilisera pour notre application et on a clôturé avec une idée générale sur les bases de données. Le prochain chapitre sera consacré à l'analyse et la conception de notre application mobile

2.1 Introduction

La conception est une phase très importante pour le développement d'une application car elle définit son fonctionnement et son efficacité, c'est pourquoi on va aborder dans ce chapitre cette partie essentielle pour notre projet on commençant en premier lieu par définir l'objectif de notre application et quelques généralités portant sur la méthode qui nous permet la modélisation de notre système en présentant le langage de modélisation unifié UML (UnifiedModeling Language) et montrer les diagrammes UML.

En second lieu, nous allons spécifier les besoins fonctionnels et non fonctionnels et que doit l'application fournir.

Et en dernier lieu on termine par parler de la conception de notre système en utilisant les diagrammes de cas d'utilisation, puis les diagrammes de séquence et à la fin le diagramme de classe.

2.2 Objectif de notre application

L'objectif de notre projet et de développer une application mobile permettant la communication entre les parents et les enseignants de leur(s) enfant(s) en donnant la chance aux parents de s'intégrer dans la vie scolaire de leur(s) enfant(s) et le(s) suivre en fournissant un accès à distance aux notes et aux absences. De plus, les rendez-vous pourront être directement pris via l'Internet.

Une application mobile qui remplace de ce fait le cahier de correspondance, trop souvent oublié ou perdu (parfois volontairement) par les élèves. Pour cela nous avons décomposé notre projet en deux parties.

La première partie concerne l'aspect administratif, celle qui traite la gestion de l'école, et

la seconde partie représente l'aspect utilisateur dédié aux professeurs et aux parents.

2.3 Langage de modélisation unifié

2.3.1 Définition

UML c'est l'acronyme anglais pour « Unified Modeling Language ». On le traduit par « Langage de modélisation unifié ». La notation UML est un langage visuel constitué d'un ensemble de schémas, appelés des diagrammes, qui donnent chacun une vision différente du projet à traiter. UML nous fournit donc des diagrammes pour représenter le logiciel à développer, son fonctionnement, sa mise en route, les actions susceptibles d'être effectuées par le logiciel, etc...

Ce langage ne préconise aucune démarche, ce n'est donc pas une méthode. Chacun est libre d'utiliser les types de diagramme qu'il souhaite, dans l'ordre qu'il veut. Il suffit que les diagrammes réalisés soient cohérents entre eux, avant de passer à la réalisation du logiciel.



FIGURE 2.1 – Logo de l'UML

2.3.2 Les différents diagrammes UML

UML propose 14 types de diagrammes dont 7 structurels et 7 comportementaux chacun d'eux étant dédié à la représentation des concepts particuliers d'un système logiciel, on va représenter ces deux grands groupes.

Diagrammes de structure ou diagrammes statiques

1. **Diagramme de classes (class diagram)** : représentation des classes intervenant dans le système.[1]
2. **Diagramme d'objets (object diagram)** : représentation des instances de classes (objets) utilisées dans le système.[1]

3. **Diagramme de composants (component diagram)** : représentation des composants du système d'un point de vue physique, tels qu'ils sont mis en œuvre (fichiers, bibliothèques, bases de données...)[1]
4. **Diagramme de déploiement (deployment diagram)** : représentation des éléments matériels (ordinateurs, périphériques, réseaux, systèmes de stockage...) et la manière dont les composants du système sont répartis sur ces éléments matériels et interagissent entre eux.[1]
5. **Diagramme des paquets (package diagram)** : représentation des dépendances entre les paquets (un paquet étant un conteneur logique permettant de regrouper et d'organiser les éléments dans le modèle UML), c'est-à-dire entre les ensembles de définitions.[1]
6. **Diagramme de structure composite (composite structure diagram)** : représentation sous forme de boîte blanche des relations entre composants d'une classe.[1]
7. **Diagramme de profils (profile diagram)** : spécialisation et personnalisation pour un domaine particulier d'un méta-modèle de référence d'UML.[1]

Diagrammes de comportement

1. **Diagramme des cas d'utilisation (use-case diagram)** : représentation des possibilités d'interaction entre le système et les acteurs (intervenants extérieurs au système), c'est-à-dire de toutes les fonctionnalités que doit fournir le système.[1]
2. **Diagramme états-transitions (state machine diagram)** : représentation sous forme de machine à états finis du comportement du système ou de ses composants.[1]
3. **Diagramme d'activité (activity diagram)** : représentation sous forme de flux ou d'enchaînement d'activités du comportement du système ou de ses composants.[1]
4. **Diagramme d'interaction ou diagrammes dynamique** : rassemblent :
 - **Diagramme de séquence (sequence diagram)** : représentation de façon séquentielle du déroulement des traitements et des interactions entre les éléments du système et/ou de ses acteurs.[1]
 - **Diagramme de communication (communication diagram)** : représentation de façon simplifiée d'un diagramme de séquence se concentrant sur les échanges de messages entre les objets.[1]
 - **Diagramme global d'interaction (interaction overview diagram)** : représentation des enchaînements possibles entre les scénarios préalablement identifiés sous forme de diagrammes de séquences (variante du diagramme d'activité).[1]

- **Diagramme de temps (timing diagram)** : représentation des variations d'une donnée au cours du temps (depuis UML 2.3).[1]

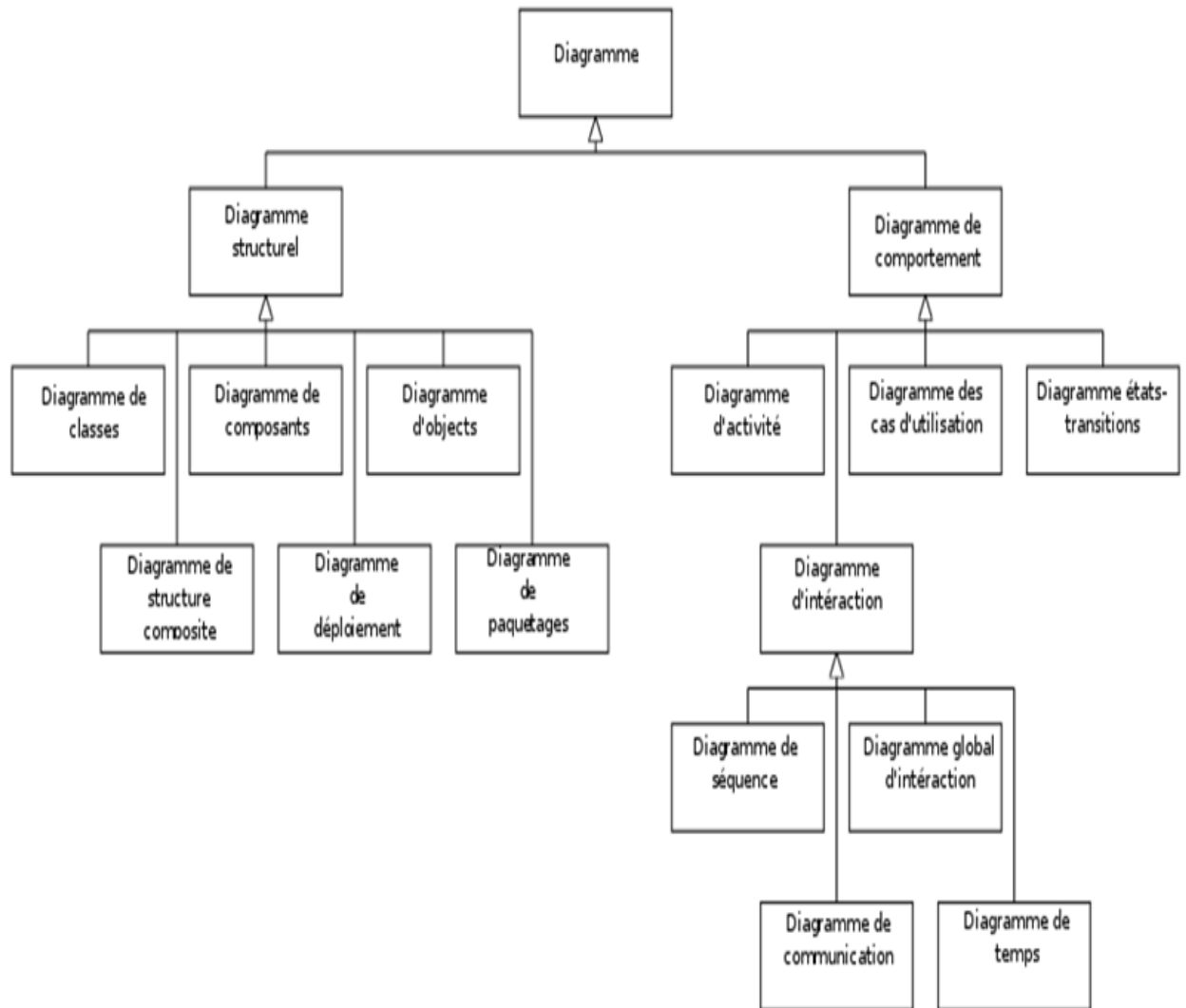


FIGURE 2.2 – Hiérarchie des diagrammes UML.[1]

2.3.3 Pourquoi l'UML ?

Ce qui nous a fait à utiliser l'UML c'est le fait qu'il procure à de nombreux avantages parmi eux :

1. **Facile à comprendre** : UML est un support de communication performant qui cadre l'analyse, il simplifie la compréhension du système et limite l'ambiguïté.
2. **Universel** : Son caractère polyvalent et sa souplesse en font un langage international très utilisé au monde.
3. **Limite les risques d'erreurs** : UML est un langage formel et normalisé, il permet un gain de précision et de stabilité.
4. **Flexible** : UML ne préconise aucune démarche, ce n'est donc pas une méthode. Chacun est libre d'utiliser les types de diagrammes qu'il souhaite, dans l'ordre qu'il veut.
5. **Lisible** : grâce aux diagrammes, UML est destiné à être compris par tout type de programmeurs et aide à expliquer les relations dans un programme d'une manière simple et illustre rapidement ses relations.

2.4 Spécification des besoins

Cette phase consiste à détailler l'ensemble des fonctionnalités que l'application doit fournir. Ces besoins seront présentés sous forme de besoins fonctionnels et non fonctionnels.

2.4.1 Besoins fonctionnels

Dans cette partie nous détaillons les fonctionnalités que le système doit fournir aux différents acteurs, l'application offre un ensemble de services scolaires administratifs et pédagogiques à chaque acteur.

1. **Administrateur** : Quand le Directeur d'établissement utilise notre application il peut :
 - Gérer les personnels de son établissement, les classes, les élèves et les parents etc... et ceci sur plusieurs années scolaires.
 - Gérer les comptes (ajouter, modifier et supprimer) des professeurs (ajoutés automatiquement) et des parents (ajoutés après demande).
2. **Professeurs** : Notre application propose aux enseignants de l'établissement les services suivants :
 - La gestion des notes : Ce service permet la saisie des notes pour chaque élève.
 - La gestion des appréciations : Envoyer une appréciation au parent d'un élève.
 - La gestion des absences : Permet l'édition de la feuille d'appel et la saisie des absences.

- Communication par messagerie avec les parents des élèves inscrits sur l'application.
- Gestion des rendez-vous : Concerne la consultation des rendez-vous et le profil du parent d'élève qui a envoyé la demande pour comprendre le sujet du rendez-vous.

3. **Parents** : L'application donne au parent la possibilité de :

- Suivre leur(s) enfant (s), leurs absences quotidienne et trimestrielle, leurs notes pour chaque matière etc. . .
- Envoyer des demandes de rendez-vous pour les enseignants de leur(s) enfant(s).
- Avoir les informations des enseignants de chaque matière étudiée par leur(s) enfant(s).
- Communiquer avec les enseignants en messagerie et prendre rendez-vous.

2.4.2 Besoins non fonctionnels

Cette partie décrit les objectifs liés aux performances du système et aux contraintes de son environnement. Notre futur système doit répondre aux besoins en matière de performance, de type de matériel ou de type de conception. Pour cela notre application mobile fournit les caractéristiques suivantes :

- Ergonomie ;
- Fiabilité et sans ambiguïté ;
- Rapidité de traitement ;
- Facilité d'utilisation ;
- Confidentialité des données personnelles ; ;
- Sécurité ;
- Convivialité des interfaces graphiques. ;

2.4.3 Besoins techniques

- Un Smartphone tournant sous le système d'exploitation Android de Google.
- Connexion Wifi ou de haut débit (3G,4G etc...).

2.5 Analyse des besoins

La première étape consiste à analyser la situation pour tenir compte des contraintes, des risques et de tout autre élément pertinent et assurer un processus qui répond à nos besoins. []

2.5.1 Identification des acteurs

Un acteur est un utilisateur type qui a toujours le même comportement vis-à-vis d'un cas d'utilisation, une même personne physique peut se comporter en autant d'acteurs différents

que le nombre de rôles qu'elle joue vis-à-vis du système. Notre système comporte les acteurs suivants :

- Administrateur.
- Enseignant.
- Parent.

L'Administrateur (responsable de l'établissement)

TABLE 2.1 – Identification de l'acteur "Administrateur"

Responsabilités	*Gérer les données de l'établissement et ajouter les classes, les élèves et les enseignants. *Saisir et gérer les comptes des utilisateurs. *Nettoyer le système en fin d'année scolaire, supprimer les données qui ne sont plus utiles.
Niveau de Compétence	Expert
Fréquence d'utilisation	Fréquente
Autorité	*Le responsable de l'établissement occupe le plus haut niveau d'autorité dans notre application.
Permissions	*L'administrateur a les pleins pouvoirs sur le système et a donc le plus haut niveau de privilège, il a accès aux informations personnelles de tous les élèves et leurs parents ainsi qu'aux enseignants.

L'enseignant

TABLE 2.2 – Identification de l'acteur "Enseignant"

Responsabilités	* Saisir les absences, les notes et les appréciations des élèves. * Consulter les demandes de rendez-vous avec les parents.
Niveau de Compétence	Intermédiaire.
Fréquence d'utilisation	Utilisation quotidienne et courte.
Autorité	*Les professeurs sont sous la responsabilité de l'administrateur (responsable de l'établissement).
Permissions	*Les privilèges des enseignants inclus l'accès aux données de leurs élèves. Ils peuvent uniquement modifier les notes des élèves qui suivent leur(s) cours. *Ils n'ont pas d'autorité sur les autres utilisateurs du système.

Le parent

TABLE 2.3 – Identification de l'acteur "Parent"

Responsabilités	* Consulter les notes et les absences de leur(s) enfant(s). * Établir des rendez-vous au besoin.
Niveau de Compétence	Novice
Fréquence d'utilisation	Selon le besoin.
Autorité	Les parents peuvent communiquer par messagerie avec les enseignants de leur(s) enfant(s).
Permissions	Les privilèges qu'ils ont se limitent à consulter les données relatives à leurs enfants. Ils ne peuvent pas modifier les informations personnelles. Ils peuvent contacter les professeurs de leurs enfants.

2.5.2 Diagramme des cas d'utilisation

Les cas d'utilisation constituent un moyen de recueillir et de décrire les besoins des acteurs du système. Ils peuvent être aussi utilisés ensuite comme moyen d'organisation du développement du logiciel, notamment pour la structuration et le déroulement des tests du logiciel.

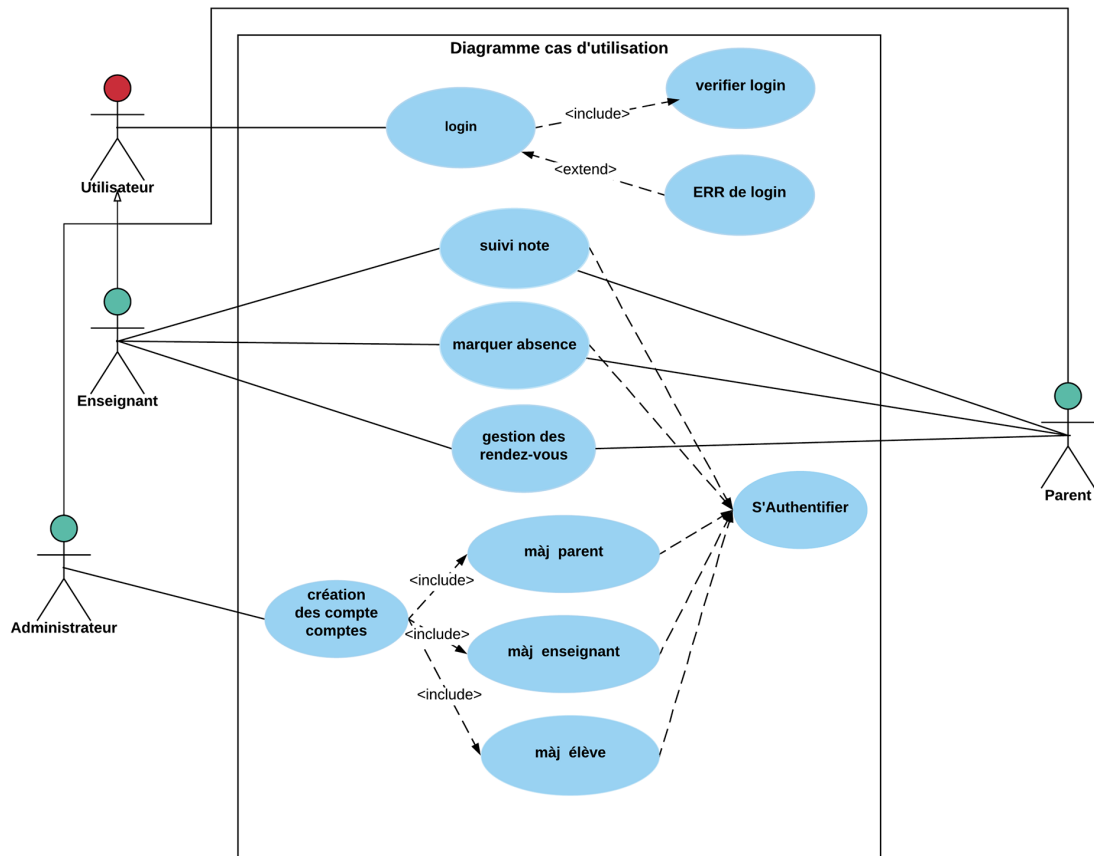


FIGURE 2.3 – Diagramme des cas d'utilisation globale.

2.6 Conception de base de données

Cette étape est nécessaire pour n'importe quel projet dans cette phase qui 'est la conception de notre système, on utilisera les diagrammes de séquences pour représenter différents scénarios, ainsi que le diagramme de classes pour représenter les entités manipulées par les utilisateurs.

2.6.1 Diagramme de classes

Le diagramme de classe a pour objectif de décrire la structure des entités manipulées par les utilisateurs, Une classe est composée des attributs et opérations, et admet des associations et multiplicité.

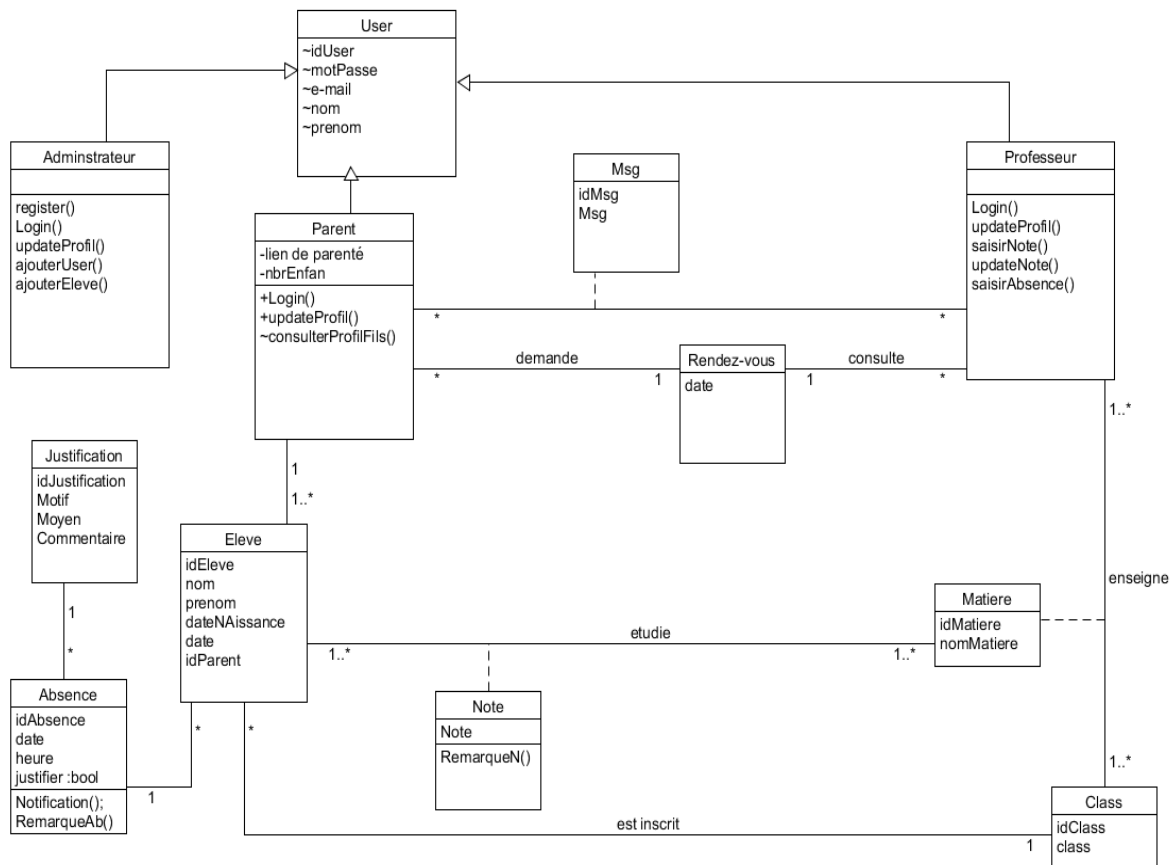


FIGURE 2.4 – Diagramme de classes.

2.6.2 Diagrammes de séquences

Le diagramme de séquence permet de représenter la succession chronologique des opérations réalisées par un acteur et indique les objets qui seront manipulés par l'acteur et les opérations.

- **Diagramme de séquence du cas «Authentification»** : L'utilisateur (administrateur, parent, enseignant) choisit l'espace concerné puis saisit les données d'authentification.

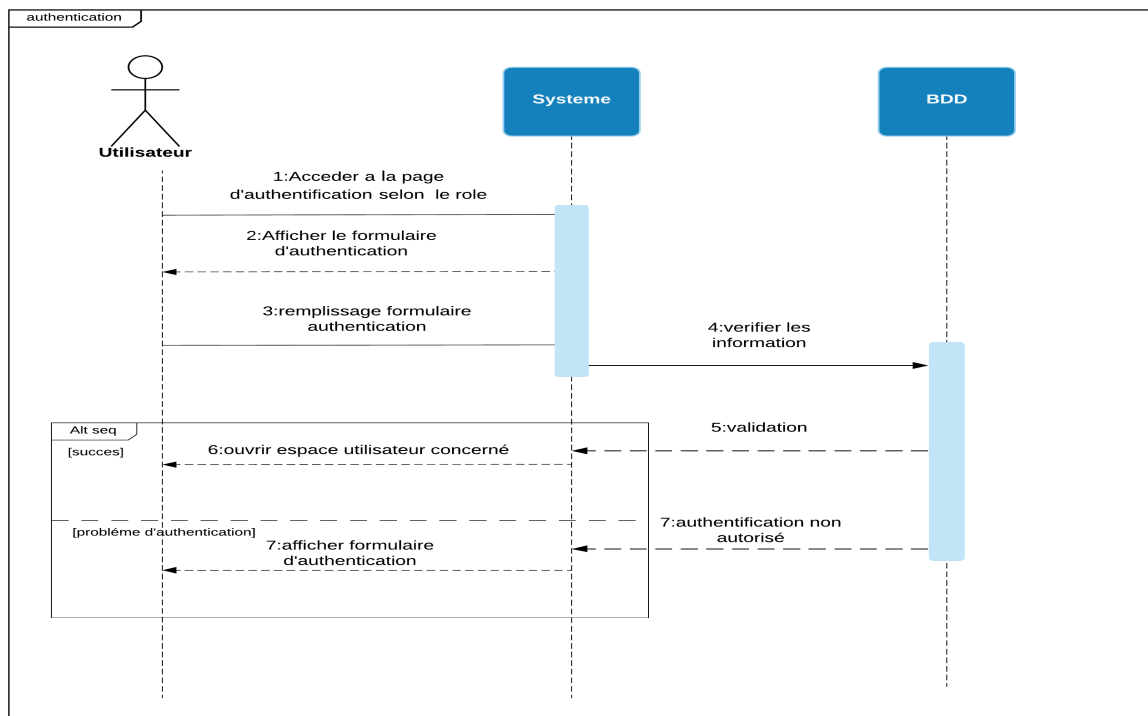


FIGURE 2.5 – Diagramme de séquence «Authentification»

- **Diagramme de séquence du cas «Ajouter un enseignant»** : L'administrateur peut ajouter des enseignants au système, il doit choisir la page de gestion des enseignants et remplir le formulaire des données personnelles de l'enseignant qui seront ajoutés au système.

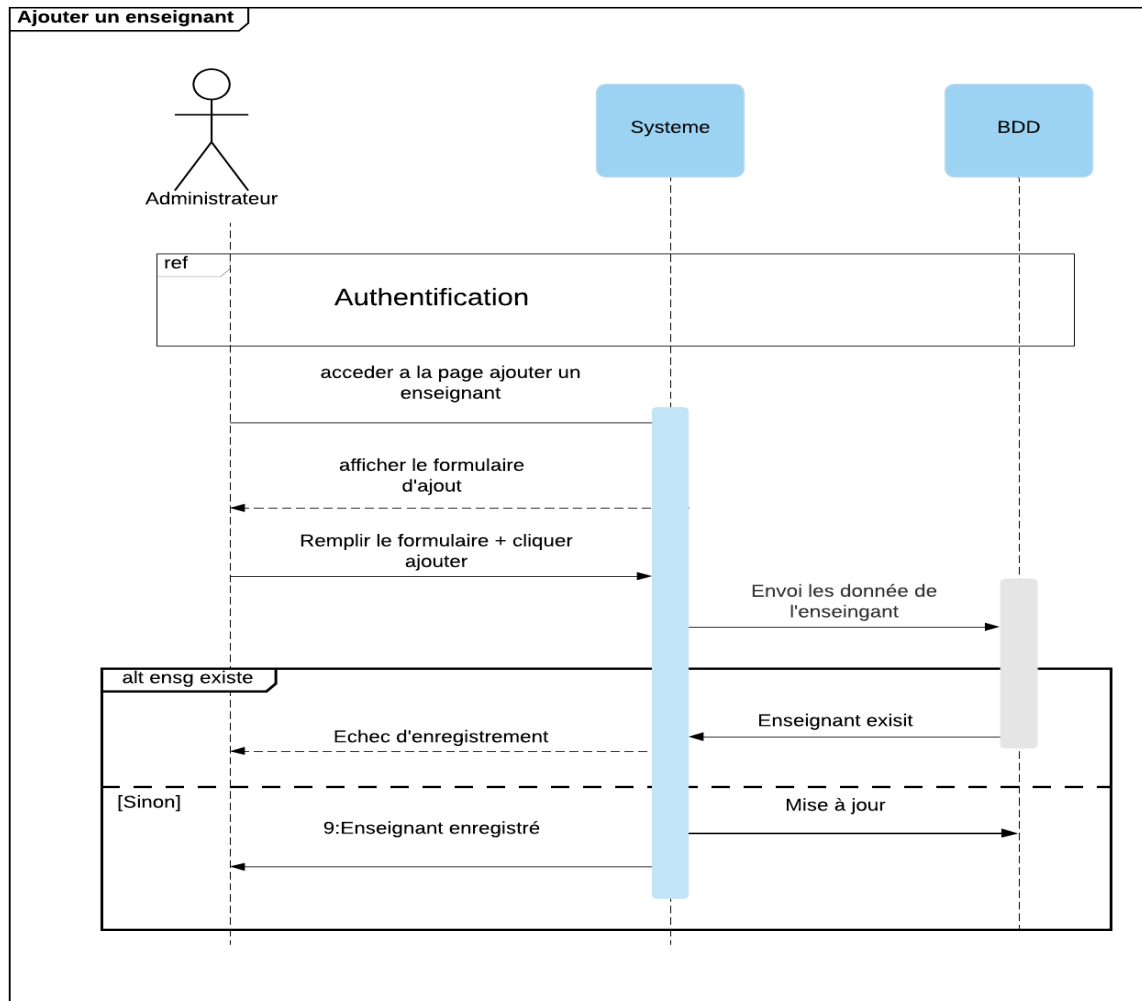


FIGURE 2.6 – Diagramme de séquence «Ajout»

- **Diagramme de séquence du cas «Evaluation des élèves»** : L'enseignant peut avoir plusieurs classes dans le même établissement le système lui affiche la liste des classes, l'enseignant choisit la classe voulue et demande la page des notes qui affichera tous les élèves appartenant à cette classe ainsi qu'un espace d'évaluation qu'il peut remplir.

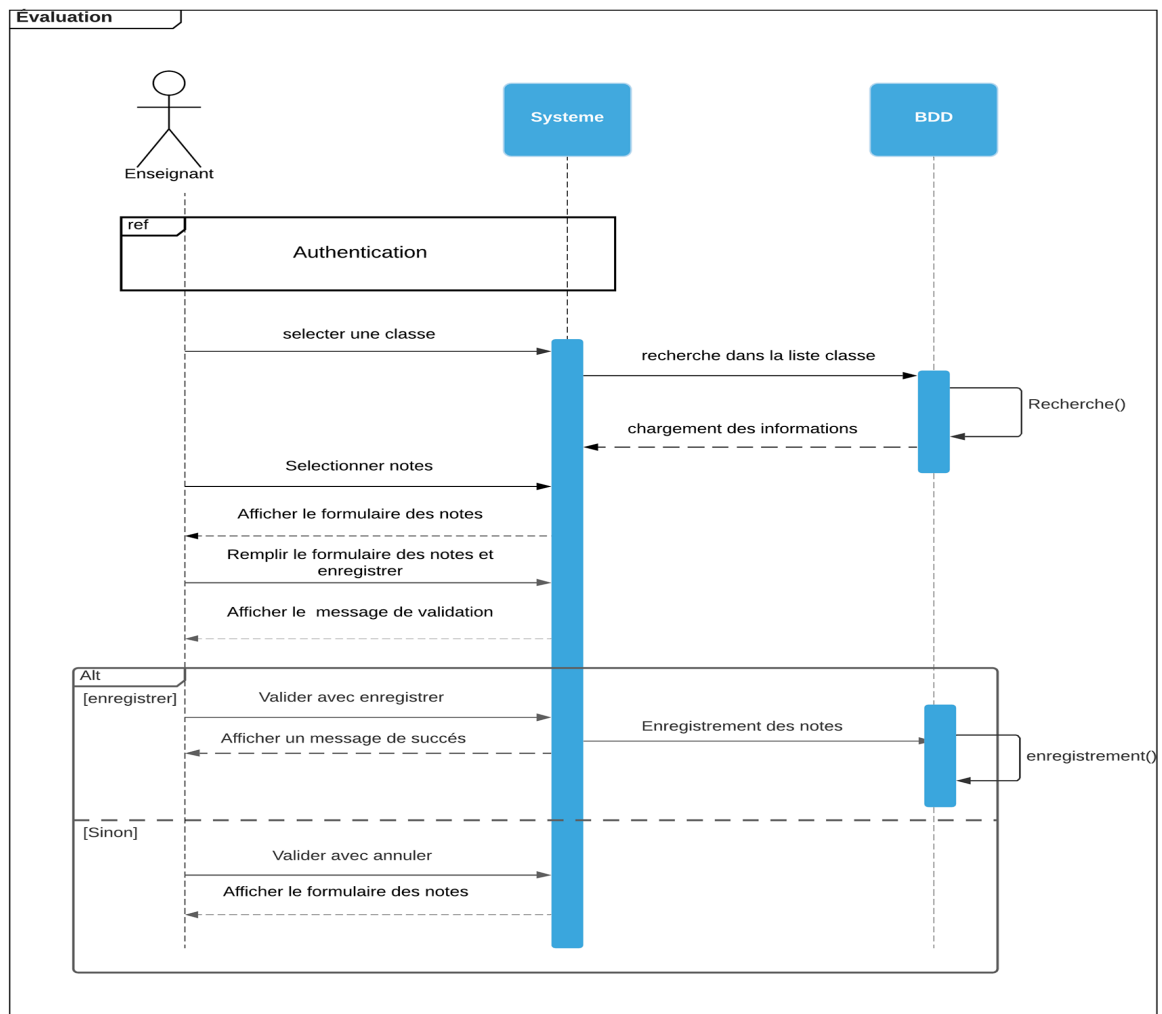


FIGURE 2.7 – Diagramme de séquence «Evaluation».

- **Diagramme de séquence du cas «Consultation des absences»** : Un parent peut avoir plus d'un enfant dans la même école, le système affiche la liste de ses enfants inscrits, le parent choisit l'élève voulu et demande la page des absences.

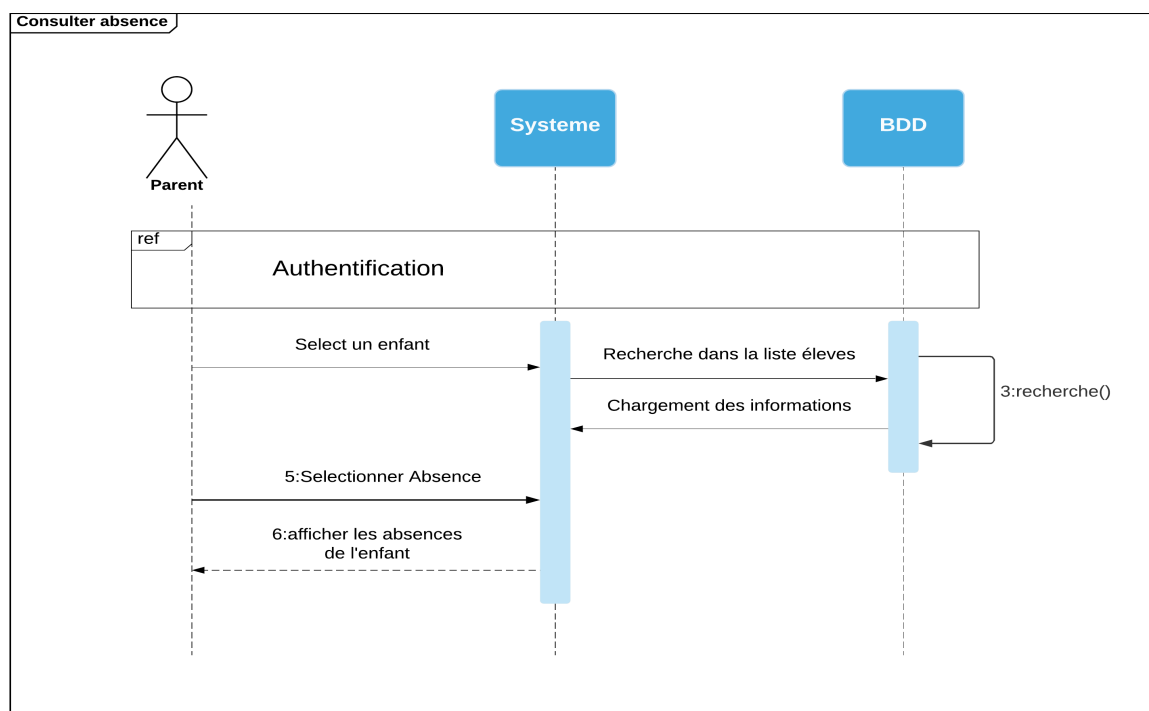


FIGURE 2.8 – Diagramme de séquence «Consultation»

2.7 Conclusion

Dans ce chapitre, nous avons commencé par présenter l'objectif de notre application tout en utilisant la démarche du processus unifié UML, nous avons spécifié besoins attendus de notre application et enfin nous avons présenté la phase d'analyse et la conception de notre système en détaillant les différents diagrammes UML appropriés jugés indispensables à notre projet. Le chapitre suivant mettra en évidence, le fruit de ce passage et les différents résultats du développement de notre application.

Le chapitre suivant sera consacré à l'étape finale de notre projet qui est l'étape de réalisation et d'implémentation.

CHAPITRE 3

RÉALISATION ET DÉVELOPPEMENT

3.1 Introduction

Durant cette dernière étape nous entamons la phase de réalisation après avoir complété l'étude conceptuelle dans le chapitre précédent, nous abordons la partie implémentation dans ce qui suit.

Ce chapitre va décrire les environnements matériels et logiciels qui nous ont permis de réaliser notre projet. Nous passons ensuite à la présentation des interfaces graphiques de l'application.

3.2 Environnement et outils du travail

Nous allons présenter l'environnement, les outils de développement utilisés ainsi que le langage de programmation et la base de données NoSql Firebase. Ces outils nous ont permis d'atteindre l'objectif escompté.

3.2.1 Environnement matériel

Notre application a été réalisée dans deux pc portable et un Smartphone pour l'exécution. Les caractéristiques de ses outils sont présentées dans ce tableau :

TABLE 3.1 – présentation des outils matériels

Caractéristiques	Lap-Top1	Lap-Top2	Smartphone
Marque	HP	Alfatron	Condor M1
RAM	12Go	8Go	4Go
Processeur	Intel® core™ i7-7500U.	Intel® core™ i5-7200U.	rev2(AArch64)
Disque dur	12Go	4Go	32Go
O.S	Windows10	Windows10	Android07

3.2.2 Environnement Logiciel

Les outils de développements que nous avons utilisés sont les suivants :

3.2.3 AndroidStudio

C'est un environnement de développement pour développer des applications mobiles Android, permet principalement d'éditer les fichiers Java/Kotlin et les fichiers de configuration XML d'une application Android.

Il propose entre autres des outils pour gérer le développement d'applications multilingues et permet de visualiser rapidement la mise en page des écrans sur des écrans de résolutions variées simultanément. Il intègre par ailleurs un émulateur permettant de faire tourner un système Android virtuel sur un ordinateur.[wikipedia].

3.2.4 SDK

Signifie Software Development Kit, c'est l'outil le plus important facile à installer, Le kit de développement d'Android est un ensemble complet d'outils de développement il inclut les utilitaires nécessaires pour aider les développeurs Android.[3]

3.2.5 JDK

Signifie Java Development Kit, Afin de développer des programmes Java, le programmeur doit disposer d'un compilateur Java et des bibliothèques de compilation, ainsi que des bibliothèques de support pour les tâches de programmation régulière, telles que le débogage. Le JDK contient un compilateur, des bibliothèques de programmes et le JRE afin de présenter aux programmeurs une plateforme permettant de compiler et d'exécuter leurs programmes. [4]

3.2.6 AVD

Signifie L'Android Virtual Device est un émulateur de terminal sous 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 en temps réel exactement à quoi ressemblera son application sur un matériel réel.[3]

3.2.7 Firebase

C'est l'un des Cloud les plus performants développés par Google. Firebase est un ensemble de plusieurs services pour n'importe quel type d'application (Android, iOS, Javascript, Node.js, Java, Unity, PHP, C++ ...) , Il propose d'héberger en NoSQL (not only SQL) et en temps réel

des bases de données, du contenu, de l'authentification sociale, et des notifications, ou encore des services, tel que par exemple un serveur de communication temps réel

Firebase a été décomposée en plusieurs produits extrêmement riches et adaptés au monde du mobile, ceux que nous avons utilisés sont considérés comme principaux, et sont décrits ci-après :

- **Realtime Database** : La base de données Firebase Realtime Database est une base de données NoSQL hébergée dans le cloud qui permet de stocker et de synchroniser des données entre utilisateurs en temps réel [2].
- **Authentication** : Solution permettant de créer et gérer facilement des moyens d'authentification variés (Google, Facebook, Email, etc...) dans le but de sécuriser l'accès à une application mobile et authentifier les utilisateurs [2].



FIGURE 3.1 – Services de Firebase

3.3 Langage de developpement

3.3.1 Java

Java est un langage de programmation inspiré du langage C++, avec un modèle de programmation orienté objet. Java permet de créer des applications complètes. Les programmes créés sont portables. Le programme source est compilé dans un « code », qui peut être exécuté sur un serveur ou un client doté d'une machine virtuelle Java.[5]

3.3.2 XML

XML signifie Extensible Markup Language, c'est un langage de balisage semblable au HTML utilisé pour décrire les données. Xml est bien lisible à la fois par l'homme et la machine. De plus, il est évolutif et simple à développer. Sous Android, nous utilisons xml pour concevoir nos mises en page, car xml est un langage léger, donc il ne rend pas notre mise en page lourde.[3]

3.4 Présentation des interfaces graphiques

Notre application « ClassBook » contient trois espaces réservés consécutivement à l'Administrateur, à l'enseignant et au parent.

Nous présentons quelques interfaces de l'application tout en citant les détails de chaque capture écran.

3.4.1 Logo de l'application

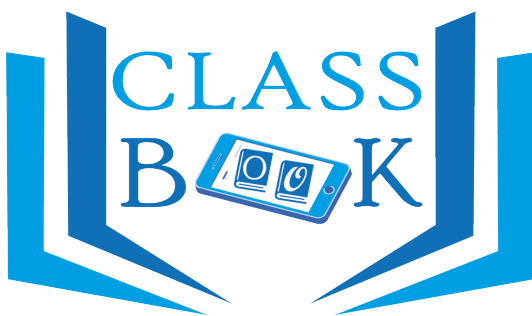


FIGURE 3.2 – Logo de l'application

3.4.2 Splash Screen

L'interface animée apparaît au lancement de l'application.



FIGURE 3.3 – Splash Screen

3.4.3 Interface de choix de l'utilisateur

La première étape avant d'accéder à l'interface d'inscription est de choisir le type de l'utilisateur de l'application, cette interface est un écran diapositive (Onboarding screen).



FIGURE 3.4 – Choix Admin

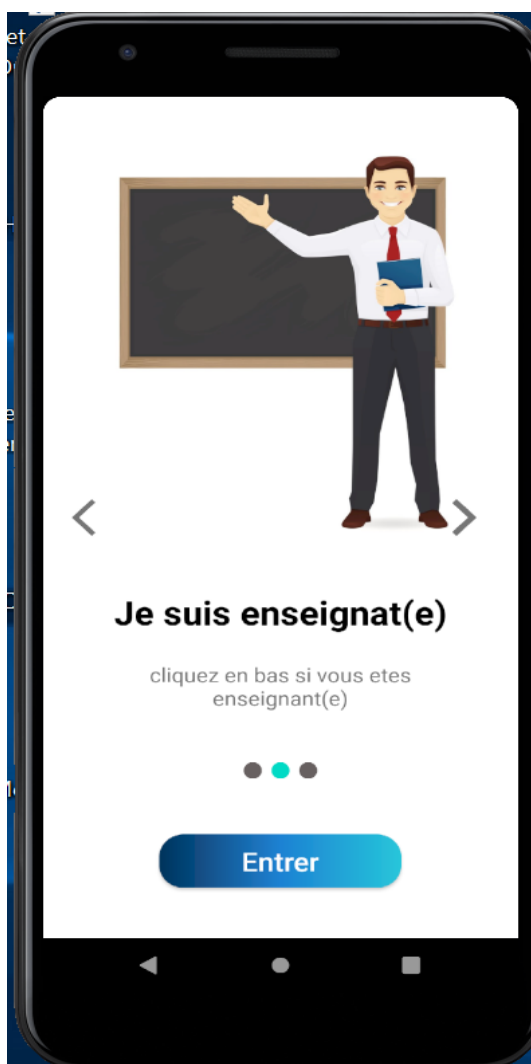


FIGURE 3.5 – Choix Enseignant

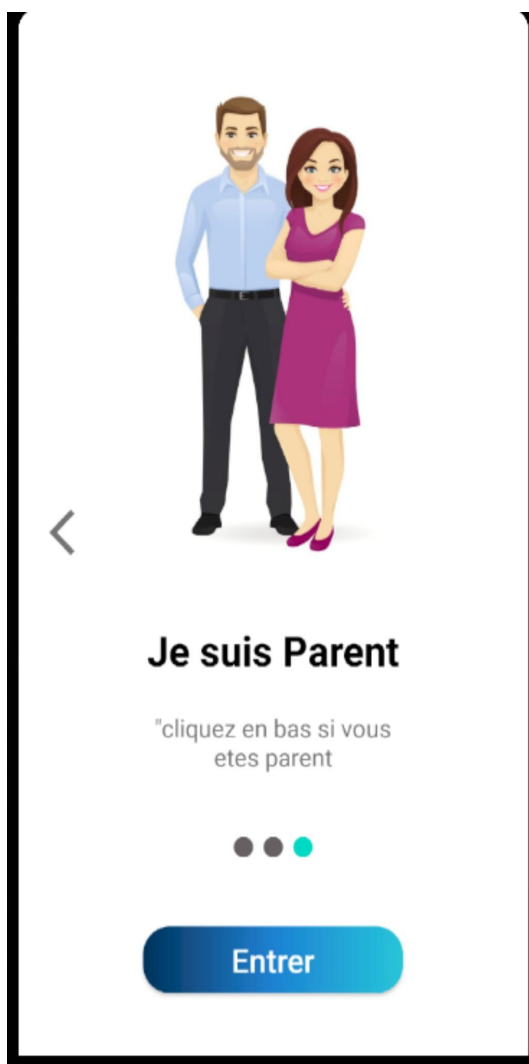


FIGURE 3.6 – Choix Parent

3.4.4 Interface d'authentification

Chaque utilisateur a sa propre interface d'authentification, elle lui permet d'accéder à son propre espace.

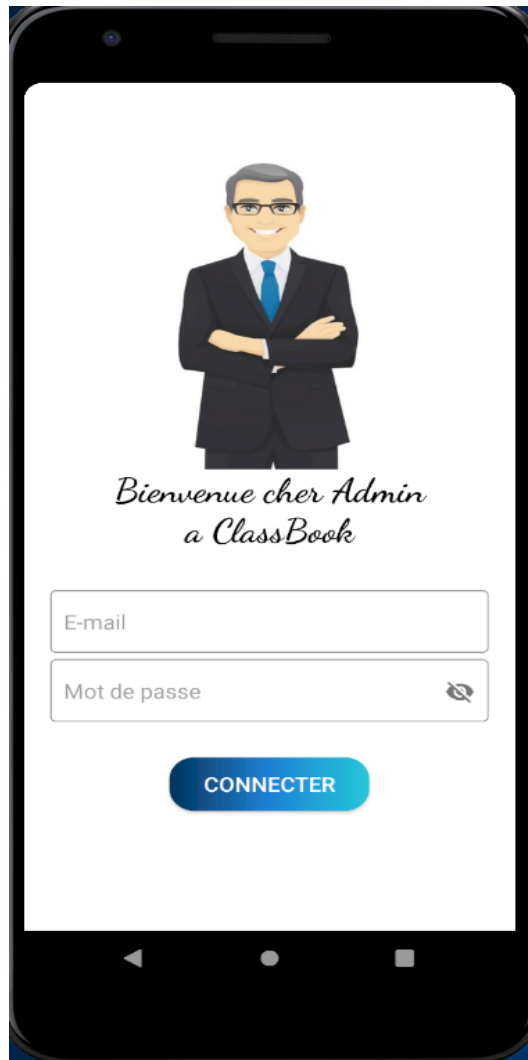


FIGURE 3.7 – Authentification Admin



FIGURE 3.8 – Authentification Enseignant



FIGURE 3.9 – Authentification Parent

3.4.5 Interface principale de l'administrateur

Cette interface permet à l'administrateur de choisir une catégorie (enseignant, parent, élève ou classe) pour création-màj-suppression ou affichage de la liste.

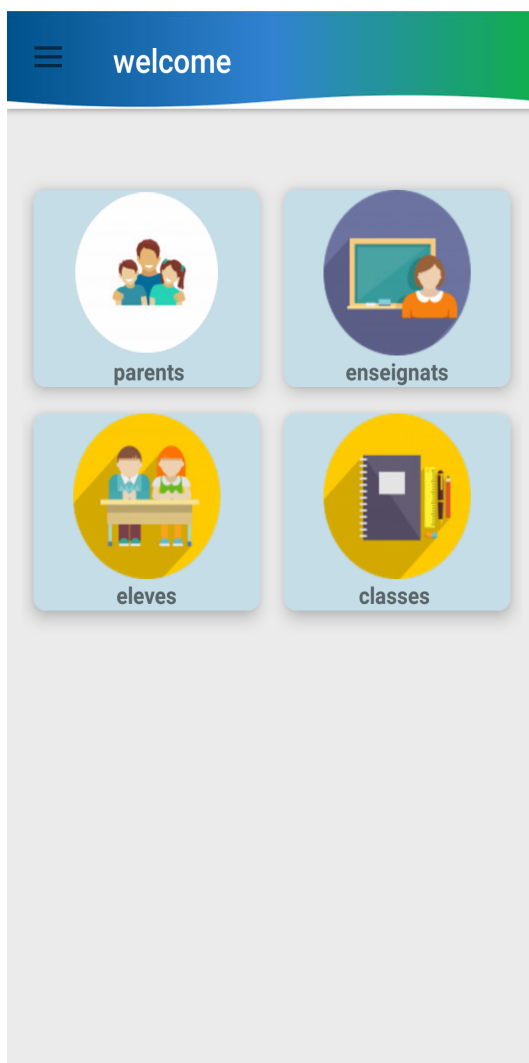


FIGURE 3.10 – Interface principale de l'administrateur

3.4.6 Interface principale d'utilisateur

Chaque utilisateur a sa propre interface d'authentification, elle lui permet d'accéder à son Interfaces, l'une pour l'enseignant la seconde pour le parent, donnent un accès facile aux informations et aux taches selon besoins.



FIGURE 3.11 – Interface principale d'utilisateur-Enseignant-



FIGURE 3.12 – Interface principale d'utilisateur-Parent-

3.4.7 Navigation Drawer

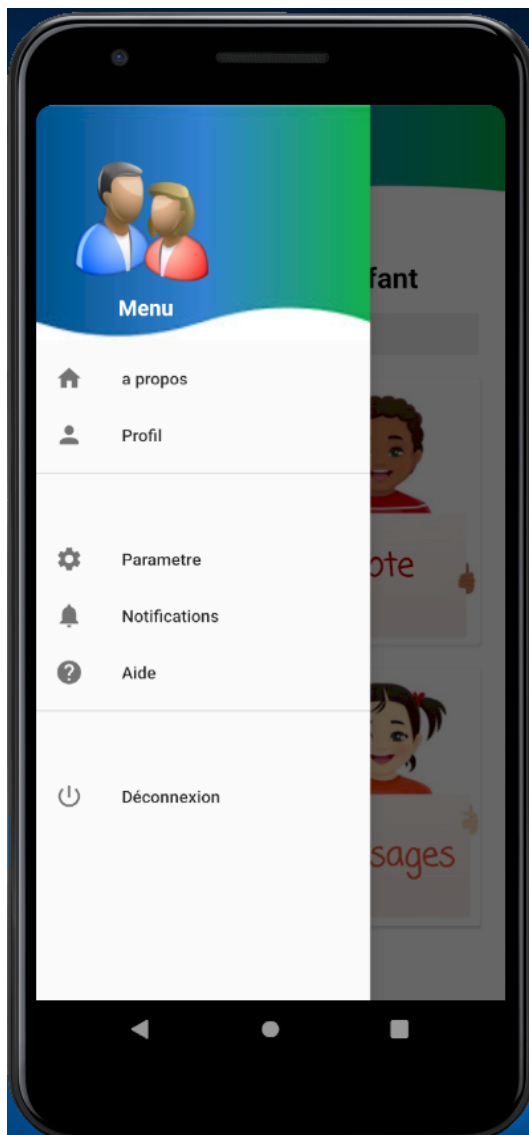


FIGURE 3.13 – Navigation Drawer

3.4.8 Interface pour marquer l'absence

Cette interface permet à l'enseignant de marquer l'absence des élèves de leur classe.

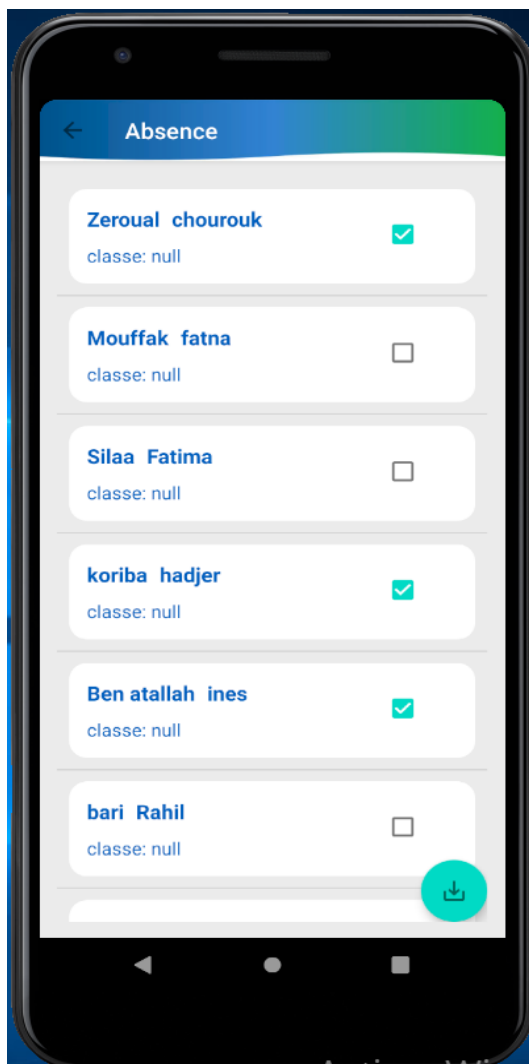


FIGURE 3.14 – Interface pour marquer l'absence

3.4.9 Interface d'évaluation

Cette interface permet à l'enseignant d'évaluer et de noter les élèves de sa classe.

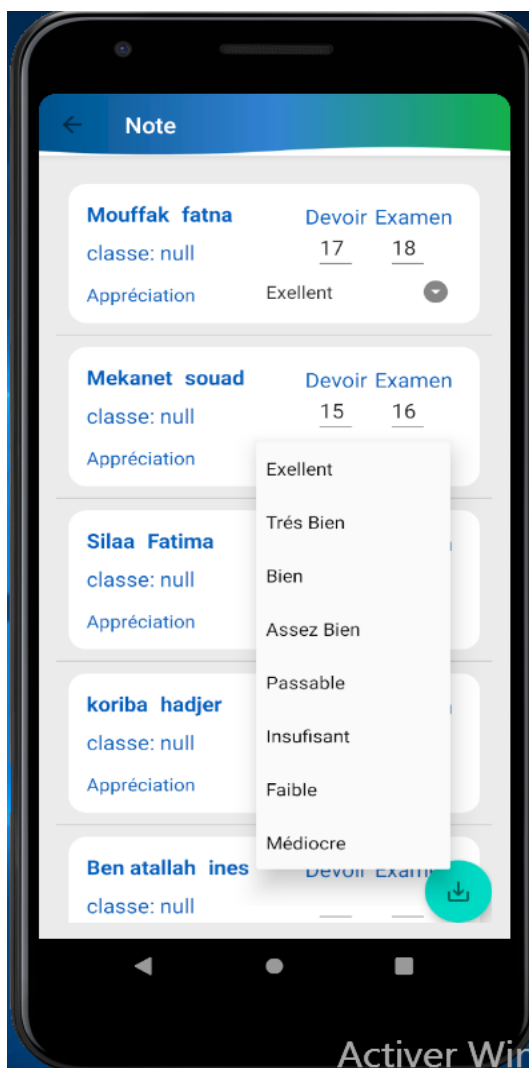
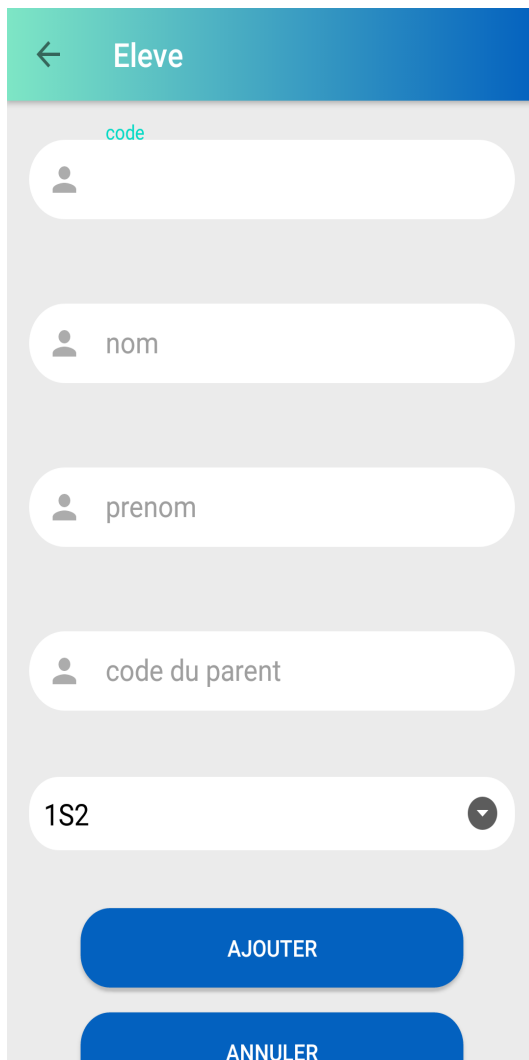


FIGURE 3.15 – Interface d'évaluation

3.4.10 Interface Ajout élève

Par le biais de cette interface l'Administrateur ajoute(crée) un nouveau élève inscrit à l'établissement.



The image shows a mobile application interface for adding a student. At the top, there is a blue header bar with a back arrow and the title 'Eleve'. Below the header, there are five input fields, each with a person icon on the left and a label on the right: 'code', 'nom', 'prenom', 'code du parent', and a dropdown menu currently showing '1S2'. At the bottom of the form, there are two blue buttons: 'AJOUTER' and 'ANNULER'.

FIGURE 3.16 – Interface Ajout élève

3.5 Conclusion

Dans ce chapitre, nous avons décrit les aspects pratiques liés à la réalisation de notre application «ClassBook », à savoir l'environnement, les outils de développement, ainsi que l'explication de Firebase, suivi des détails sur les langages utilisés, et à la fin nous avons présenté les principales interfaces de notre application mobile.

CONCLUSION GÉNÉRALE ET PERSPECTIVES

- **Conclusion :**

Ce projet de fin d'étude consiste à réaliser une application mobile sur le cloud relative au Cahier de Correspondance Scolaire. Au cours de ce mémoire, nous avons présenté les différentes étapes de la conception et la réalisation de notre application.

La réalisation de ce projet nous a permis d'apprendre à s'organiser, à travailler en groupe (étant donné que tout travail informatique parfait est l'œuvre d'une équipe) et surtout l'amélioration nos connaissances et nos compétences dans les domaines de la conception, la programmation et le design.

Nous avons appris à mieux manipuler les langages Java, Xml , les bases de données NoSql, le Firebase et l'AndroidStudio en général. En effet, ce travail étant une nouvelle expérience très marquante qui va nous permettre de mieux s'imprégner dans l'univers du développement et la programmation, ce n'est donc pas un modèle parfait ou typiquement professionnel, c'est pourquoi nous restons ouverts et acceptons toute critique constructive, suggestion ou remarque nous permettant de progresser davantage dans le développement et la conception.

- **Perspectives :**

Malgré les problèmes de configuration et de développement qu'on a rencontré avec Firebase et AndroidStudio etc.. , nous avons pu réaliser la totalité de l'application mobile en faisant fonctionner les différentes parties de cette dernière.

Pour un meilleur accompagnement scolaire des élèves, nous envisageons d'améliorer dans un avenir proche notre application en lui ajoutant par exemple :

- * Numérisation de l'inscription des élèves.
- * Téléconférence entre association des parents d'élèves, parents, enseignants et administration.
- * Génération de données statistiques instantanés sur les retards, l'absentéisme, le niveau scolaire par élève, par classe et global. etc...
- * Envoi des relevés de notes et bulletins au format PDF par mobile.

En outre, et en terme de perspective nous souhaitons que notre application soit redéveloppé en utilisant l'outil intelligence artificielle qui a conquis presque tous les secteurs et l'utiliser par exemple dans les cours e-learning à distance durant une pandémie ou absence générale suite à désastre naturelle quelconque car cela facilite le suivi de chaque élève, aussi on demande à améliorer et adapter l'application pour un usage général dans les établissements scolaires afin qu'ils en tirent le meilleur profit de la technologie de pointe.

BIBLIOGRAPHIE

- [1] UML2 Analyse ET Conception Mise en œuvre guidée avec étude de cas, Joseph Gabay, David Gbay, Editions DUNOD 2008
- [2] [https ://firebase.google.com/ ?](https://firebase.google.com/?)
- [3] FREDERIC ESPIAU, « Créez des applications pour Android », Openclassroom, 2012.
- [4] CYRILLE HERBY, « Apprenez à programmer en Java », OpenClassrooms - ex-Site du Zéro ,2e édition, 2012.
- [5] [https ://www.lemagit.fr/definition/Java](https://www.lemagit.fr/definition/Java)