

كلية العلوم  
+٠٢٤٧٧٠٢١ | +٢٤٠٥٥٠٤١  
FACULTÉ DES SCIENCES



UNIVERSITE IBN ZOHR  
**FACULTE DES SCIENCES**

Département Informatique  
Filière Sciences Mathématiques et Informatique

**PFE**  
**Présenté par : IAICH Ismail et El BARAGHI Ibrahim**

Pour l'obtention de la  
Licence en Sciences Mathématiques et Informatique

Conception et réalisation d'une application android pour la gestion des variétés  
végétales

Soutenu le 16/06/2022

**Encadré par : M. MACHKOUR Mustapha**

Année universitaire 2021-2022



# Dédicace

Toutes les lettres ne sauraient trouver les mots qu'il faut...

Tous les mots ne sauraient exprimer la gratitude, l'amour, Le respect, la reconnaissance...

## A nos très chers parents :

Chère parents comment on peut exprimer une dédicace à vous et nous savons qu'Aucune dédicace ne saurait être assez éloquente pour exprimer la profondeur De nos sentiments d'affection, d'estime et de respect pour l'amour dont vous avez toujours nous submergée, pour votre soutien, vos sacrifices et vos prières.

## A nos chers amis et collègues :

Un grand merci pour Qui font partie de ces personnes rares par leur gentillesse, leur tendresse et leurs grands cœurs. Merci pour votre inlassable soutien et pour tous les merveilleux moments que nous avons passé ensemble dans un environnement familial.

## A nos Professeurs et Encadrant et jurys:

Nous sommes très reconnaissant d'avoir votre aidé pendant toutes les préparations et notre parcours. Vous nous avez prodigués de conseils et de sagesse pour réussir notre trajet et consolider notre formation. Nous vous remercions Pour vos encouragements, votre temps et votre entière disponibilité

# Remerciements

AU début, Nous tenons tout d'abord à remercier DIEU le tout puissant et miséricordieux, Qui nous 'a donné la force et la patience et l'énergie d'accomplir ce travail. Nos remerciements et notre profonde gratitude s'adressent particulièrement à notre Encadrent :

Monsieur : MUSTAPHA MACHKOUR

Pour son encadrement de qualité, qui ne s'est pas attardée à nous 'orienter et à nous 'encourager, pour ses conseils et critiques constructives et pour son suivi et son patience durant la période de formation et de la préparation et à la réalisation de notre projet de fin d'étude.

Au terme de notre projet de fin d'études, on tient à témoigner de notre sincères remerciements à toutes les personnes qui ont contribué de près ou de loin au bon déroulement de notre projet de fin d'études et à l'élaboration de ce modeste travail.

A tous ceux qui ont contribué à la réussite de ce travail

# Résumé

Ce présent rapport est rédigé dans le cadre du projet de fin d'étude. Le projet à réaliser consiste à créer une application Android. Cette application permet de faciliter la réalisation de l'étude de la distinction des variétés soumises à homologation , et la gestion de catalogue officiel des variétés végétales de l'Office National de Sécurité Sanitaire des produits Alimentaires (ONSSA).

Enfin, pour réaliser cette application, on 'a utilisé Android Studio comme un environnement de développement intégré (IDE) et JAVA comme un langage de programmation.

Mots clés : Application android, variétés, catalogue officiel des variétés végétales, ONSSA, Android Studio, IDE, JAVA.

# **Abstract**

This report is written as part of the end of study project. The project to be carried out consists in creating an Android application. This application will facilitate the study of the distinction of varieties subject to registration, and the management of the official catalogue of plant varieties of the National Office of Food Safety (ONSSA).

Finally, we used Android Studio as an integrated development environment (IDE) and JAVA as a programming language to build this application.

**Keywords :** Android application, varieties, the official catalogue of plant varieties, ONSSA, Android Studio, IDE, JAVA .

# Table des matières

Dédicace .....	iii
Remerciements .....	iv
Résumé .....	v
Abstract.....	vi
Table des matières .....	vii
Liste des figures.....	ix
Liste des tableaux .....	x
Liste des abréviations .....	xi
Introduction générale.....	1
Chapitre 1 Contexte général du projet .....	3
1. Contexte du projet .....	3
2. Description du projet.....	3
3. Problématique et Objectifs du projet.....	3
4. Conduite du projet.....	4
5. Planification du projet .....	4
Chapitre 2 Analyse et conception.....	6
1. Présentation des cas d'utilisation .....	6
2. Diagrammes des séquences .....	8
3. Diagramme de classe.....	12
4. Modélisation des données .....	13
Chapitre 3 Etude technique et environnement .....	17
1. Développement mobile .....	17
2. Les besoins techniques .....	20
3. Architecture adoptée .....	20
4. Choix des langages.....	20
Chapitre 4 Réalisation, interfaces, tests .....	22

1. Environnement du travail.....	22
A. Android Studio:.....	22
B. XAMPP.....	27
C. Power AMC .....	30
D. Visual Studio Code: .....	31
2. Base de données de notre application.....	33
3. Présentation de l’application .....	33
CONCLUSION .....	46
Bibliographie .....	46



# Liste des figures

FIGURE 1 : DIAGRAMME DE GRANT .....	5
FIGURE 2 : SCHEMA DE METHODOLOGIE DE CONCEPTION .....	6
FIGURE 3 : DIAGRAMME DES CAS D'UTILISATION.....	7
FIGURE 4 : DIAGRAMME DE SEQUENCES DE CONNEXION .....	8
FIGURE 5 : DIAGRAMME DE SEQUENCE D'INSCRIPTION .....	9
FIGURE 6 : DIAGRAMME DE SEQUENCE D'AJOUTER UNE VARIETE .....	9
FIGURE 7: DIAGRAMME DE SEQUENCES D'AJOUTER UN ESPECE.....	10
FIGURE 8 : DIAGRAMME DE SEQUENCE D'ACCEPTATION DES VARIETES.....	10
FIGURE 9 : DIAGRAMME DE SEQUENCES DE MODIFICATION OU SUPPRESSION D'UNE VARIETE .....	11
FIGURE 10 : DIAGRAMME DE SEQUENCE DE GESTION DES COMPTES .....	12
FIGURE 11: DIAGRAMME DE CLASSE .....	13
FIGURE 12 : MODELE CONCEPTUEL DES DONNEES.....	14
FIGURE 13 : MODELE LOGIQUE DES DONNEES .....	15
FIGURE 14 : MODELE PHYSIQUE DES DONNEES .....	16
FIGURE 15 : DIFFERENTS PLATEFORME MOBILE .....	18
FIGURE 16 : PRESENTATION DES STATISTIQUES DES VENTES.....	19
FIGURE 17 : ANDROID STUDIO .....	22
FIGURE 18 : AVD MANAGER .....	23
FIGURE 19 : LE DEBUGGEUR.....	23
FIGURE 20 : L'INTERFACE DES FICHER EXECUTIVES .....	24
FIGURE 21 : L'INTERFACE INTUITIVE POUR LES FICHERS DE DESIGNE .....	25
FIGURE 22 : L'INTERFACE BRUTE POUR LES FICHERS DE DESIGNE.....	25
FIGURE 23 : XAMPP .....	27
FIGURE 24 : XAMPP CONTROLE PANEL .....	28
FIGURE 25 : ADMIN PAGE DE APACHE .....	29
FIGURE 26 : L'INTERFACE DE PHPMYADMIN .....	30
FIGURE 27: POWERDESIGNER.....	30
FIGURE 28 : L'INTERFACE DE POWERDESIGNER.....	31
FIGURE 29 : VISUAL STUDIO CODE .....	31
FIGURE 30 : L'INTERFACE DE VS CODE .....	32
FIGURE 31 : LE CYCLE DE VIE D'UNE ACTIVITE.....	34
FIGURE 32 : ECRAN DE CONNEXION .....	35
FIGURE 33 : ECRAN D'INSCRIPTION... ..	35
FIGURE 34 : ESPACE UTILISATEUR .....	36
FIGURE 35 : VOTRE PROFILE .....	36
FIGURE 36 : LIST DES VARIETES .....	37
FIGURE 37 : COMPARER DEUX VARIETES .....	38
FIGURE 38 : ESPACE AUTEUR .....	39
FIGURE 39 : NOTIFICATIONS .....	39
FIGURE 40 : AJOUTER UNE VARIETE .....	40
FIGURE 41 : AJOUTER UN ESPECE .....	41
FIGURE 42 : ESPACE CORRECTEUR .....	42
FIGURE 43 : NOUVELLES VARIETES .....	42
FIGURE 44 : GESTION DES VARIETES .....	42
FIGURE 45 : ESPACE ADMINISTRATEUR .....	43
FIGURE 46 : GESTION DES COMPTES .....	44
FIGURE 47 : DEMANDES DE CHANGEMENT DE TYPE DU COMPTE.....	45

# Liste des tableaux

TABLEAU 1: LES ACTEURS DE NOTRE APPLICATION .....	7
---	---

# Liste des abréviations

**IDE** : Integrated Development Environment

**MCD** : Modèle Conceptuel des Données

**MLD** : Modèle Logique des Données

**MPD** : Modèle Physique des Données

**ONSSA** : Office National de Sécurité Sanitaire des produits Alimentaires

**PHP** : Pre Hypertexte Processor

**SGBD** : Système de Gestion de Bases de Données

**SGBDR** : Système de Gestion de Base de Données Relationnelle

**SQL** : Structured Query Language

**UML** : Unified Modelling Language

**UPOV** : Union Internationale pour la Protection des Obtentions Végétale

**XML** : Extensible Markup Language

# Introduction générale

Les utilisateurs de smartphone consacrent 90% de leur temps aux applications mobile, en comparaison avec les sites web (Statistique Furry, 2015).

« Le mobile est le seul canal avec cette portée et cette profondeur d'engagement », note App Annie.

Les principaux chiffres à retenir de l'étude, à l'échelle mondiale :

-143 milliards de dollars de dépenses sur les boutiques d'applications (+20 % sur un an),

-218 milliards de téléchargements de nouvelles applications (+7 % sur un an),

-4,2 heures passées par jour en moyenne sur un mobile Android (+20 %).

Actuellement, le marché de la téléphonie mobile est en pleine révolution, du simple téléphone utilisé pour passer des appels aux téléphones avancés dotés de fonctions similaires à de vrais ordinateurs appelés téléphones intelligents. Ces téléphones intelligents sont légers, puissants et intelligents et remplaceront progressivement les équipements téléphoniques standards dans les magasins.

Le terme « application mobile » désigne un logiciel. Il s'agit plus précisément d'un programme, qui contient un fichier, pouvant être téléchargé depuis un téléphone mobile ou encore, depuis une tablette. Une fois installé, ce programme est exécuté par le système d'exploitation du smartphone ou de la tablette sur lequel il se trouve.

le fichier qui compose une application mobile est exécuté par le système d'exploitation de l'appareil sur lequel il se trouve. Nous retrouvons alors deux types de langage de développement spécifiques aux appareils mobiles, à savoir :

Le JAVA ou Kotlin, pour les smartphones et pour les tablettes Android, comme les dispositifs Samsung par exemple ,Et Objective C ou Swift pour IOS, autrement dit, pour tous les dispositifs Apple.

À noter : Il est possible de mettre en place une application destinée aux deux systèmes d'exploitation, soit Android et IOS. On parle alors « d'application hybride ».

Le présent travail constitue une contribution pour l'élaboration d'une application android conçue spécialement pour être utilisée comme support d'aide à la prise de décision lors de l'examen de la distinction des variétés soumises à l'homologation.

Ce rapport peut ainsi être subdivisé en quatre parties :

La première : consiste à la présentation du projet, des objectifs principaux puis les fonctionnalités de base.

La seconde : est consacrée à la conception de la solution proposée pour répondre au cahier des charges.

La troisième : nous présentons les différents langages choisis utilisé pour l'implémentation de notre application.

La quatrième : nous traitons l'environnements matériels, logiciels, suivie d'une description des différentes interfaces graphiques permettant l'utilisation de l'application.

Enfin ce mémoire est clôturé par une conclusion générale résumant les points essentiels du travail réalisé .

# Chapitre 1

## Contexte général du projet

### 1. Contexte du projet

Ce projet se situe dans le cadre de projet de fin d'études, qui est un élément indispensable pour valider les études du cycle licence au sein de la Faculté des Sciences d'Agadir (FSA) .

### 2. Description du projet

Il s'agit d'une application Android de taille égale à 14Mo qui fonctionne sur tous les téléphones mobiles Android (avec une version Android supérieure ou égale à 4.0) peut être utilisé par n'importe quelle personne avec des compétences informatiques ordinaires et de connaissances moyennes mais juste après l'inscription. Une personne peut créer un et un seul compte de type utilisateur utilisant son adresse email .

Il existe quatre types des comptes , chacune a des privilèges et des rôles :

- Administrateur : s'occupe de la gestion des comptes .
- Correcteur : s'occupe de la gestion des variétés.
- Utilisateur : un compte ordinaire avec les privilèges d'exploiter et de comparer les variétés existantes .
- Auteur : un utilisateur avec le privilège d'ajouter une nouvelle variété .

Les administrateurs peuvent changer le type de compte d'un certain utilisateur et la garantie quelque privilège .

**Public cible :** cette application concerne les obtenteurs qui souhaitent investir dans la production de nouvelles variétés ainsi que les scientifiques et les chercheurs dans l'amélioration génétique. Outre les utilisateurs habituels qui souhaitent améliorer leur connaissance des variétés existantes.

### 3. Problématique et Objectifs du projet

Au Maroc chaque année une centaine de variétés sont proposées pour inscription au catalogue officiel détenu par l'ONSSA. Depuis sa création en 1977, ce catalogue compte en 2021 plus de 4000 variétés enregistrées, toutes espèces confondues. Pour qu'une variété soit inscrite au catalogue, elle doit satisfaire à une série d'épreuves durant deux campagnes consécutives, pouvant qu'elle est bien distincte , homogène et stable et que sa valeur agronomique et technologique est égale ou supérieure à celles des variétés homologuées .

Dans ce contexte, l'exigence de distinction requière une comparaison avec toute autre variété préalablement inscrite au catalogue officiel national. Pour mener à bien cette évaluation, le service responsable de l'homologation doit disposer nécessairement d'un system informatique lui permettant d'accéder aux données résultantes des essais d'homologation des variétés existantes .

Le travail présenté dans ce mémoire s'agit de réaliser une application android qui pourrait faciliter la gestion des variétés et l'étude de la distinction des variétés soumis à l'homologation.

Les fonctionnalités qui devront être offertes par l'application pour chaque type de compte sont :

- Un utilisateur (ou auteur) peut exploiter les variétés existantes .
- Un auteur peut ajouter des variétés .
- Un utilisateur peut comparer deux variétés .
- Un correcteur peut supprimer et modifier les variétés .
- Un correcteur peut accepter ou refuser une variété ajouté par un auteur.
- Un administrateur peut changer le type d'un compte (alors ces privilèges) ou supprimer un compte .

## **4. Conduite du projet**

En ce qui concerne la conduite de projet et le partage des tâches entre nous durant ce projet : Nous avons commencé par une étude bibliographique sur le sujet, afin de comprendre l'objectif et le contexte générale, puis nous avons fait le choix sur les plateformes et les outils de développement qu'on va utiliser, en mettant le temps nécessaire à la prise en main de certains outils, tels que Android Studio, qui intègre, notamment, l'environnement de développement IntelliJ IDEA,

Notre travail a été fait en binôme, en partageant les tâches de temps en temps. Pour l'étude conceptuelle du projet, la modélisation des diagrammes et l'élaboration des diagrammes (de classe, de séquences, etc.) nous avons utilisé le logiciel PowerAMC. Pour la partie développement, nous nous sommes servi, essentiellement, de la logiciel Android Studio, dans laquelle nous avons développé par le langage JAVA. Ensuite, nous avons utilisé XAMPP pour la création du serveur local, gestion de scripts PHP et gestion de base de données (PHPMyAdmin).

## **5. Planification du projet**

Le plan comprend la prédiction du processus de l'ensemble du projet, Les étapes qui composent le cycle de développement. Par conséquent, nous avons utilisé le diagramme de Gantt qu'est un Outil de planification des tâches.

On présente dans la figure suivante le diagramme de Gantt de la planification élaboré qui représente nos tâches principales pour réaliser notre projet :

## planification du projet

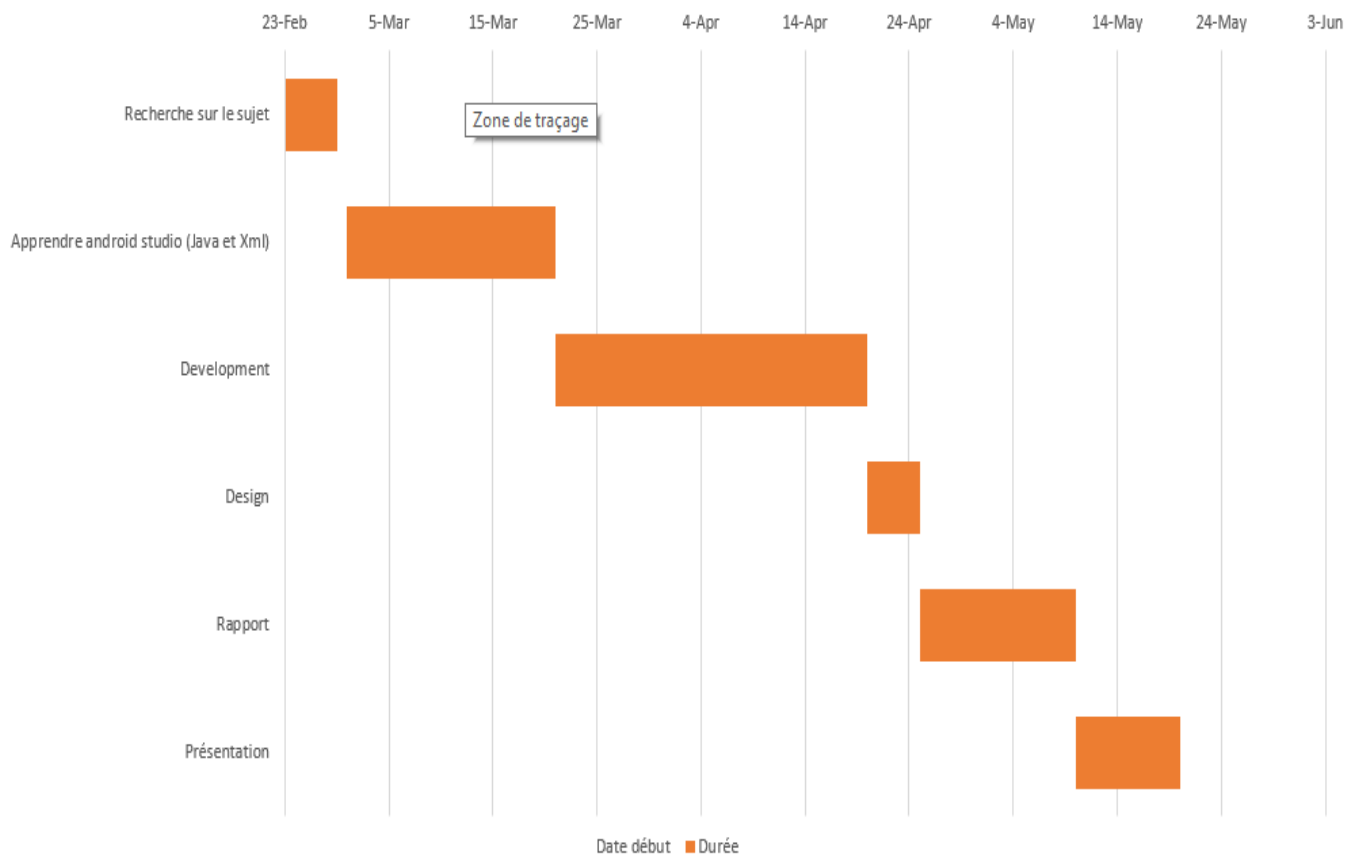


Figure 1 : Diagramme de Grant



# Chapitre 2

## Analyse et conception

Dans ce chapitre, nous commençons la partie liée à la conception. En montrant nos solutions sous forme de diagrammes UML qu'il répond bien aux besoins spécifiques et couvre le système (statique, dynamique, etc.).

Nous avons utilisé trois diagrammes: diagramme de cas d'utilisation, diagramme de séquence et diagramme de classe. Le schéma suivant représente notre méthodologie de conception:

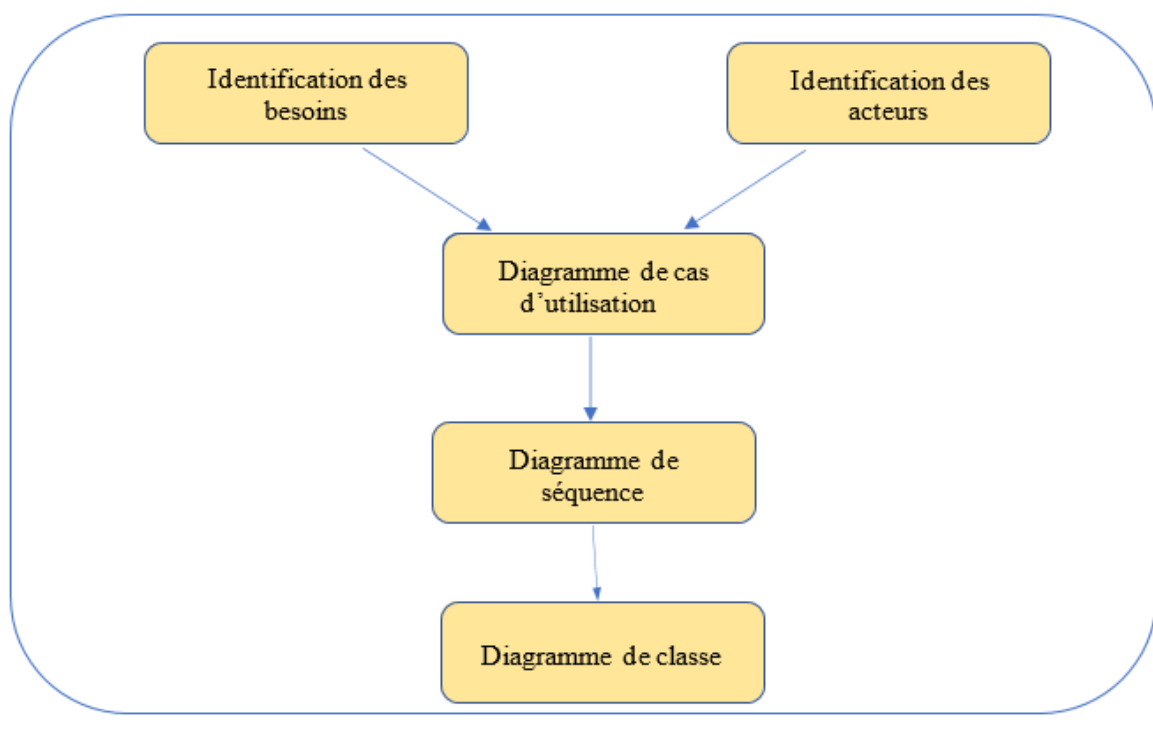


Figure 2 : Schéma de méthodologie de conception

### 1. Présentation des cas d'utilisation

A l'objectif de représenter correctement et clairement notre application, et pour limiter l'ambiguïté et les incompréhensions, UML (Unified Modeling Language) accompagne les différentes étapes de notre projet.

Au rôle de donner une vue du système dans son environnement extérieur, on utilise le diagramme de cas d'utilisation pour représenter la structure des grandes fonctionnalités nécessaires aux utilisateurs du système. Qui donne une vue de la relation entre l'utilisateur et les éléments que le système met en œuvre.

Les acteurs qui interagissent avec le système sont :

Tableau 1: les acteurs de notre application

Acteur	Cas d'utilisation
Administrateur	<ul style="list-style-type: none"> <li>■ Suppression des comptes</li> <li>■ Changement de type de compte</li> </ul>
Correcteur	<ul style="list-style-type: none"> <li>■ Acceptation des variétés</li> <li>■ Suppression des variétés</li> <li>■ Modification des variétés</li> </ul>
Auteur	<ul style="list-style-type: none"> <li>■ Ajoute des variétés</li> <li>■ Recherche des variétés</li> </ul>
Utilisateur	<ul style="list-style-type: none"> <li>■ Recherche des variétés</li> <li>■ Comparaison des variétés</li> </ul>

Le diagramme de cas d'utilisation est donnée par :

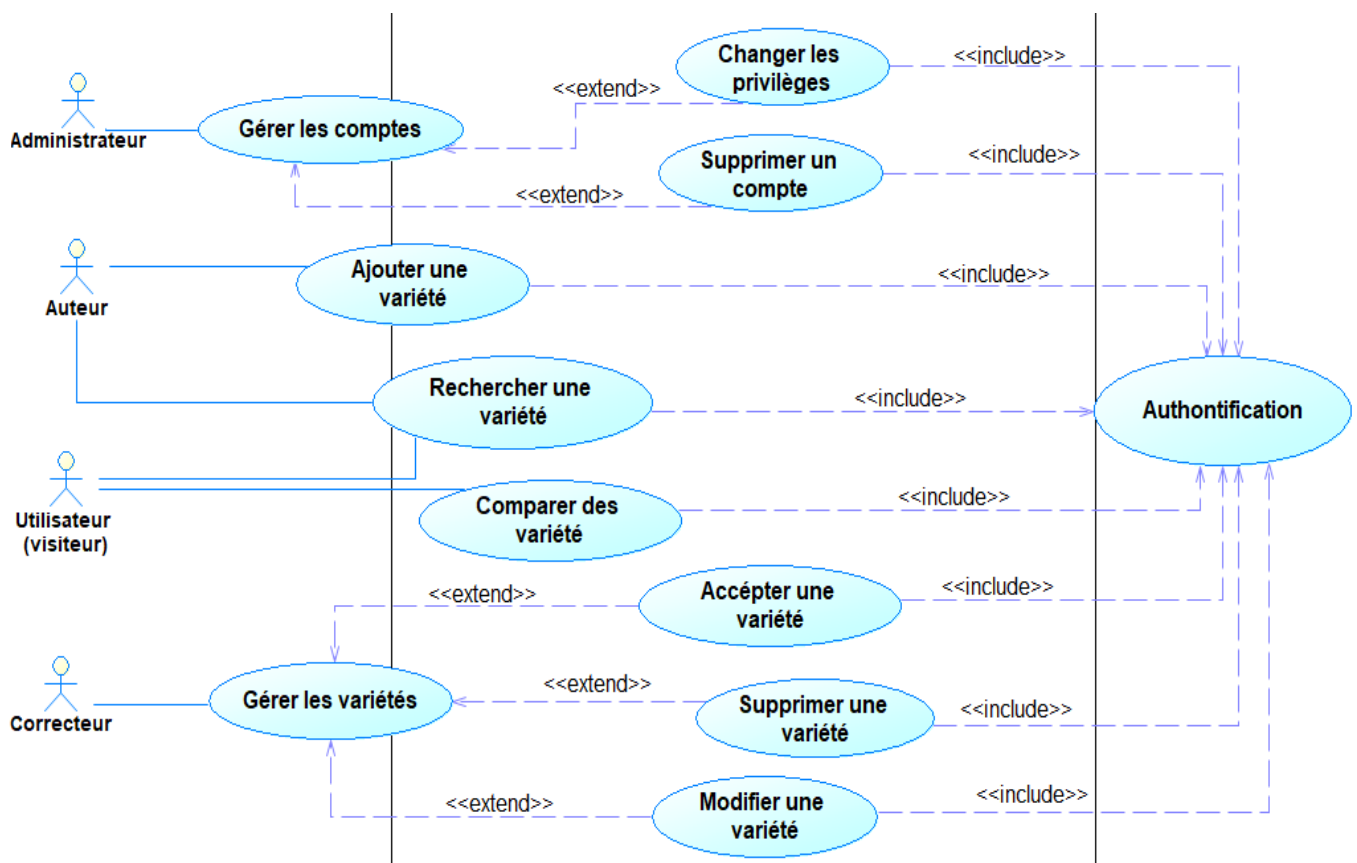


Figure 3 : Diagramme des cas d'utilisation

## 2. Diagrammes des séquences

Dans cette partie nous allons compléter les descriptions textuelles des diagrammes des cas d'utilisation par une représentation graphique UML, dont l'objectif est de représenter les interactions entre les objets en indiquant la chronologie des échanges

Nous avons 6 diagrammes de séquence :

Figure 4 : représente le scénario suivi par les différents type d'utilisateurs pour qu'il s'authentifie.

Figure 5 : représente le scénario suivi pour créer un compte.

Figure 6 : représente le scénario suivi par les auteurs pour ajouter une variété.

Figure 7 : représente le scénario suivi par les auteurs pour ajouter un espèce.

Figure 8 : représente le scénario suivi par les correcteurs pour accepter ou refuser une variété .

Figure 9 : représente le scénario suivi par les correcteurs pour modifier ou supprimer une variété

Figure 10 : représente le scénario suivi par les administrateurs pour la gestion des comptes.

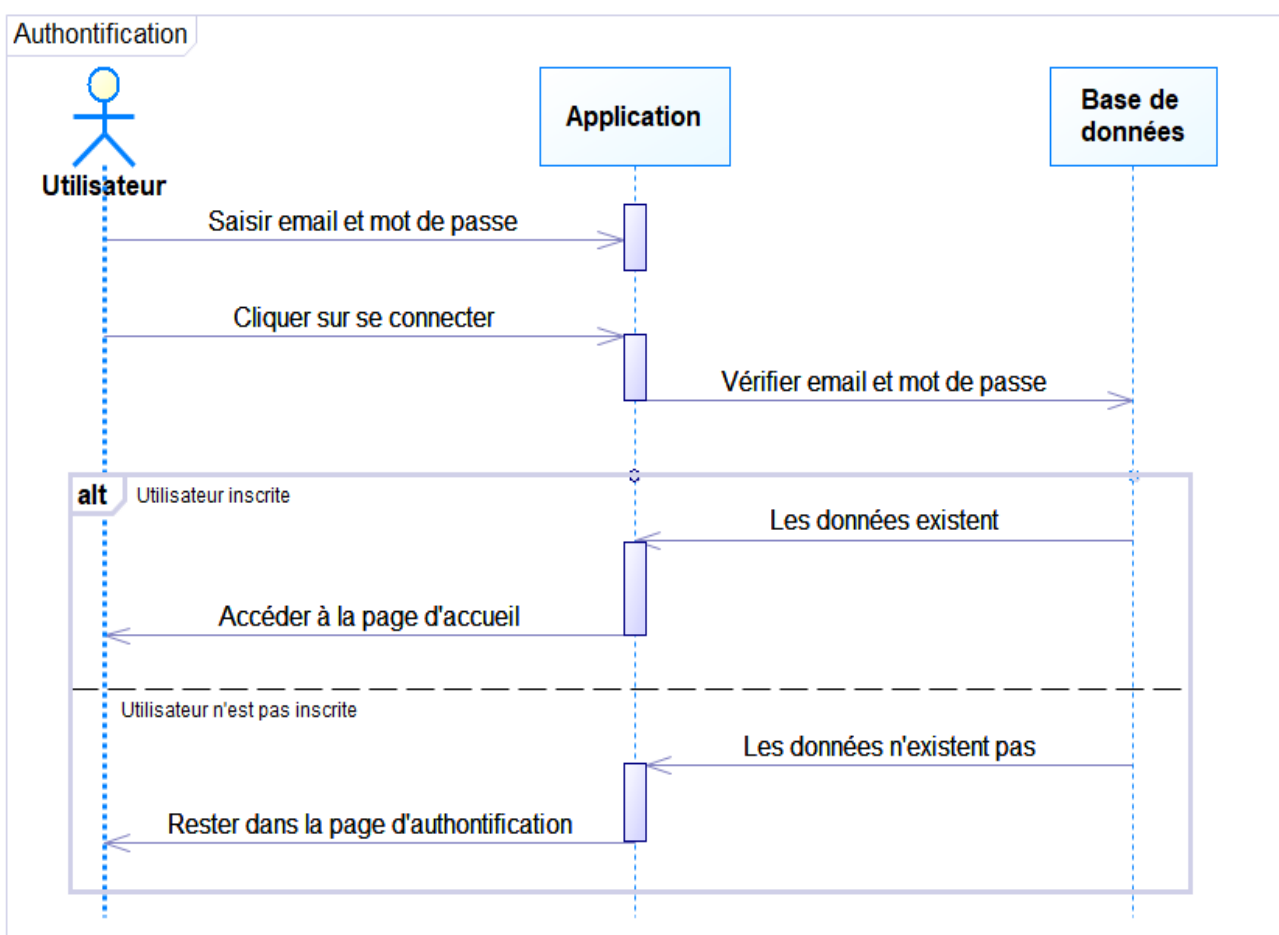


Figure 4 : Diagramme de séquences de connexion

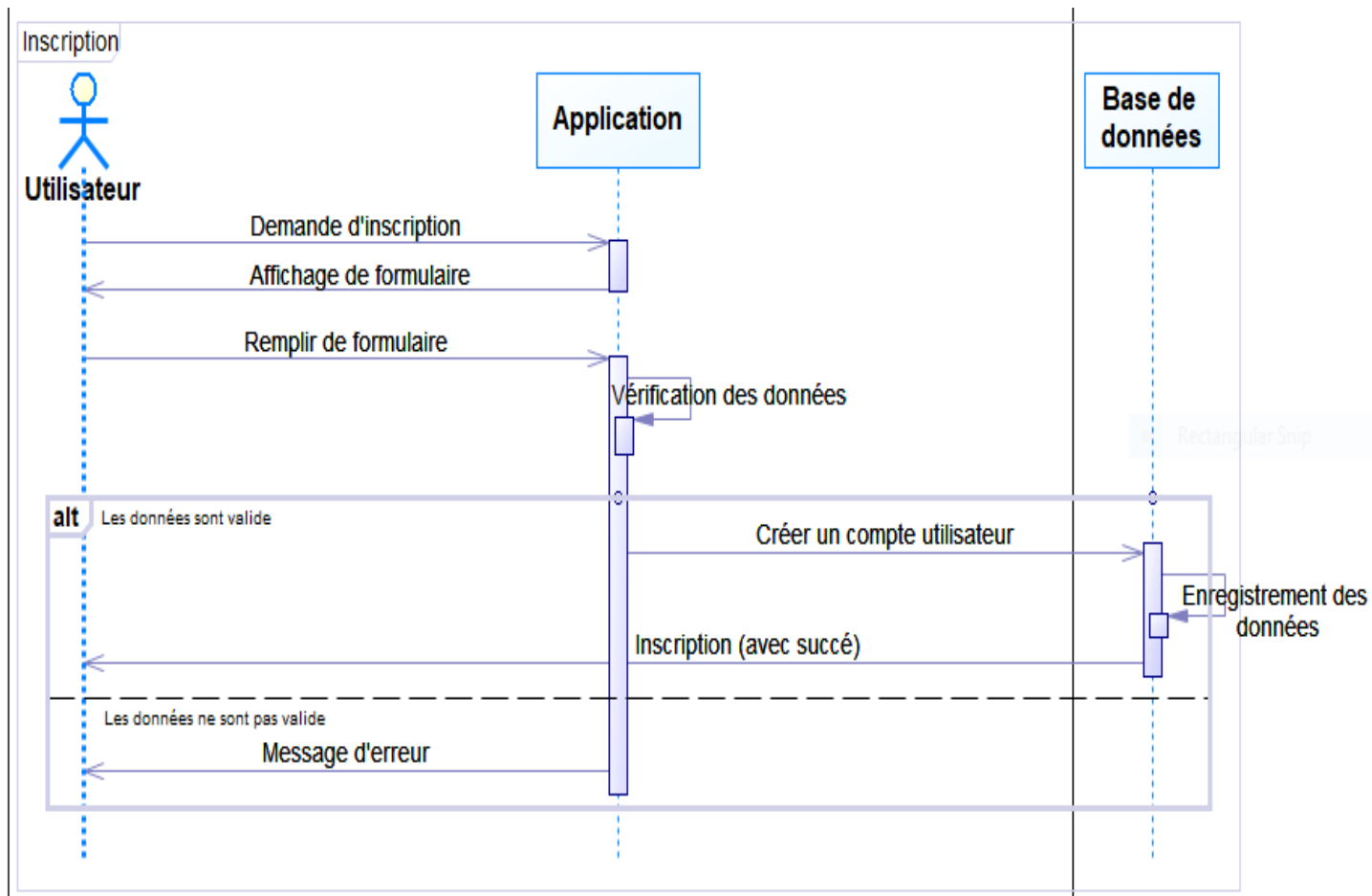


Figure 5 : Diagramme de séquence d'inscription

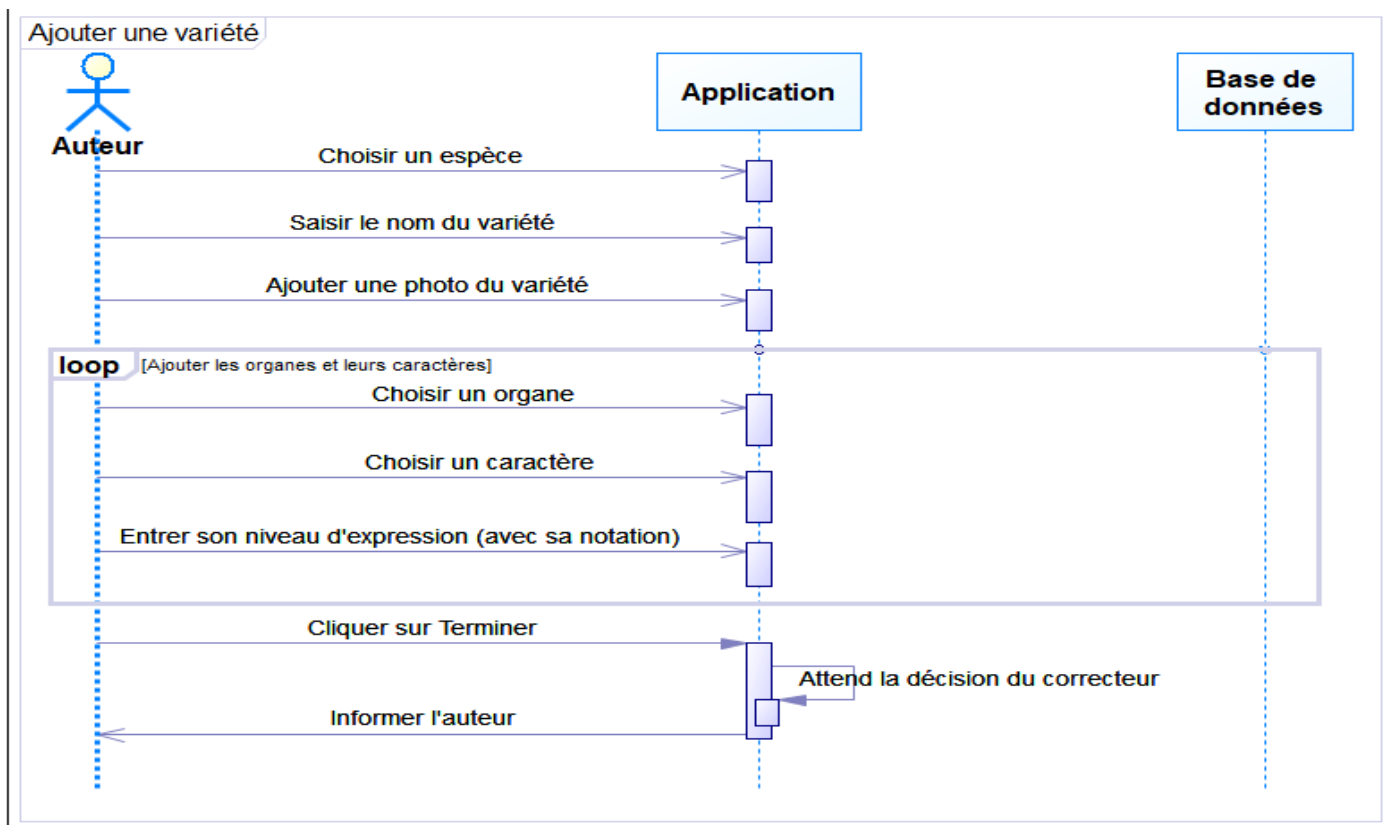


Figure 6 : Diagramme de séquence d'Ajouter une variété

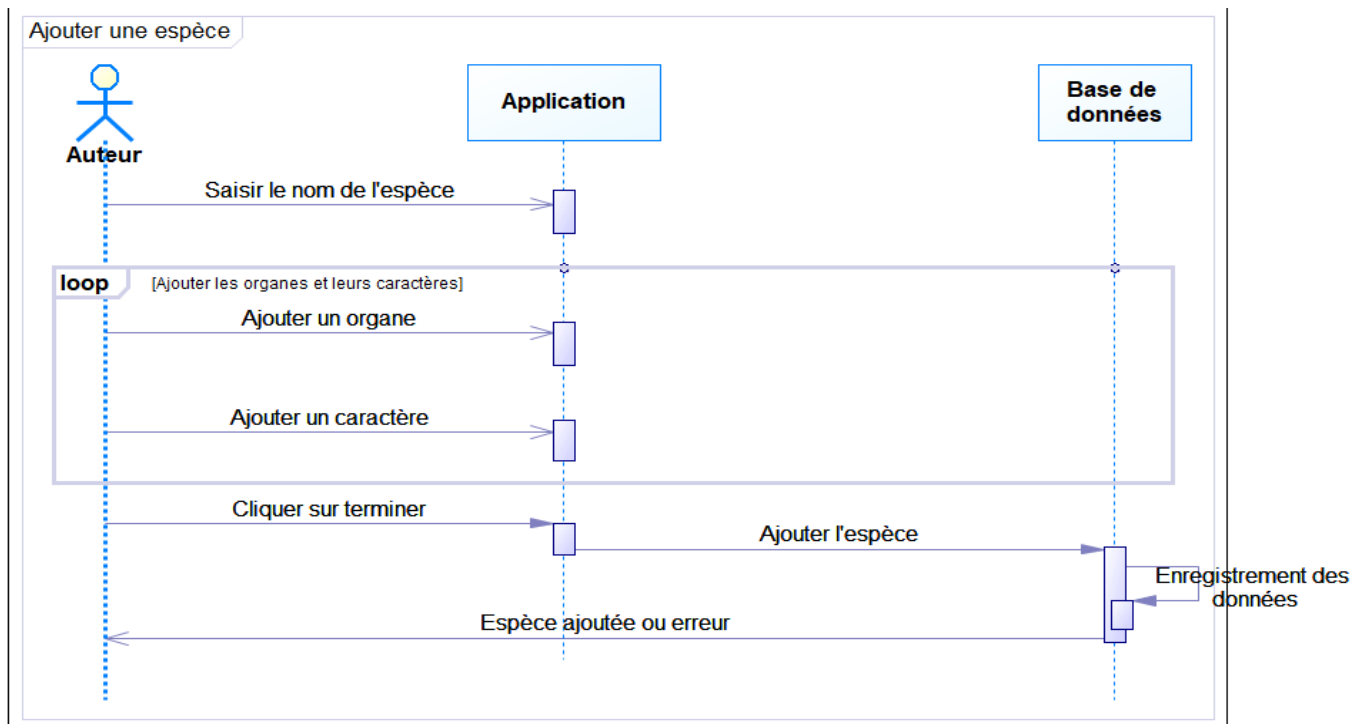


Figure 7: Diagramme de séquences d'ajouter un espèce

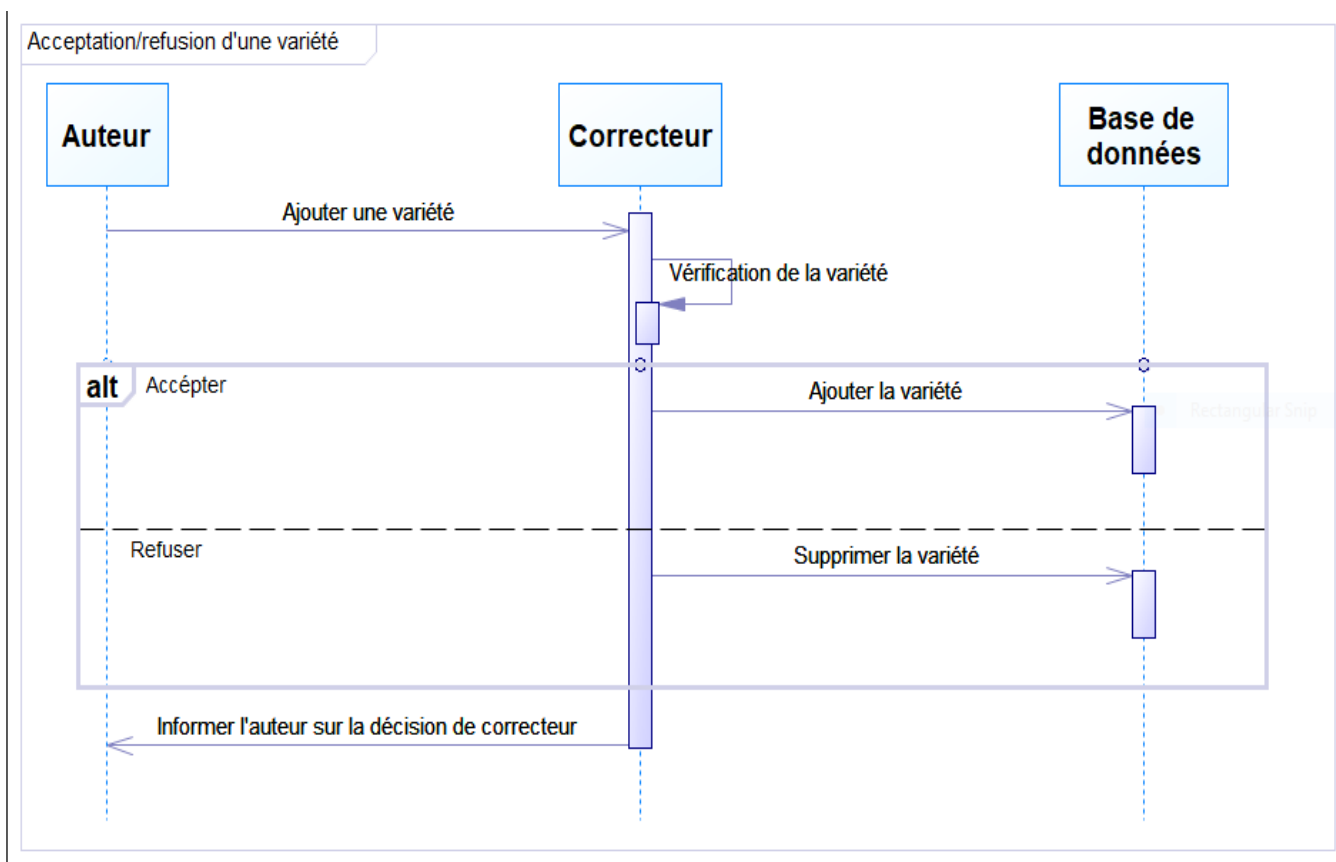


Figure 8 : Diagramme de séquence d'acceptation des variétés

## Modification des variétés

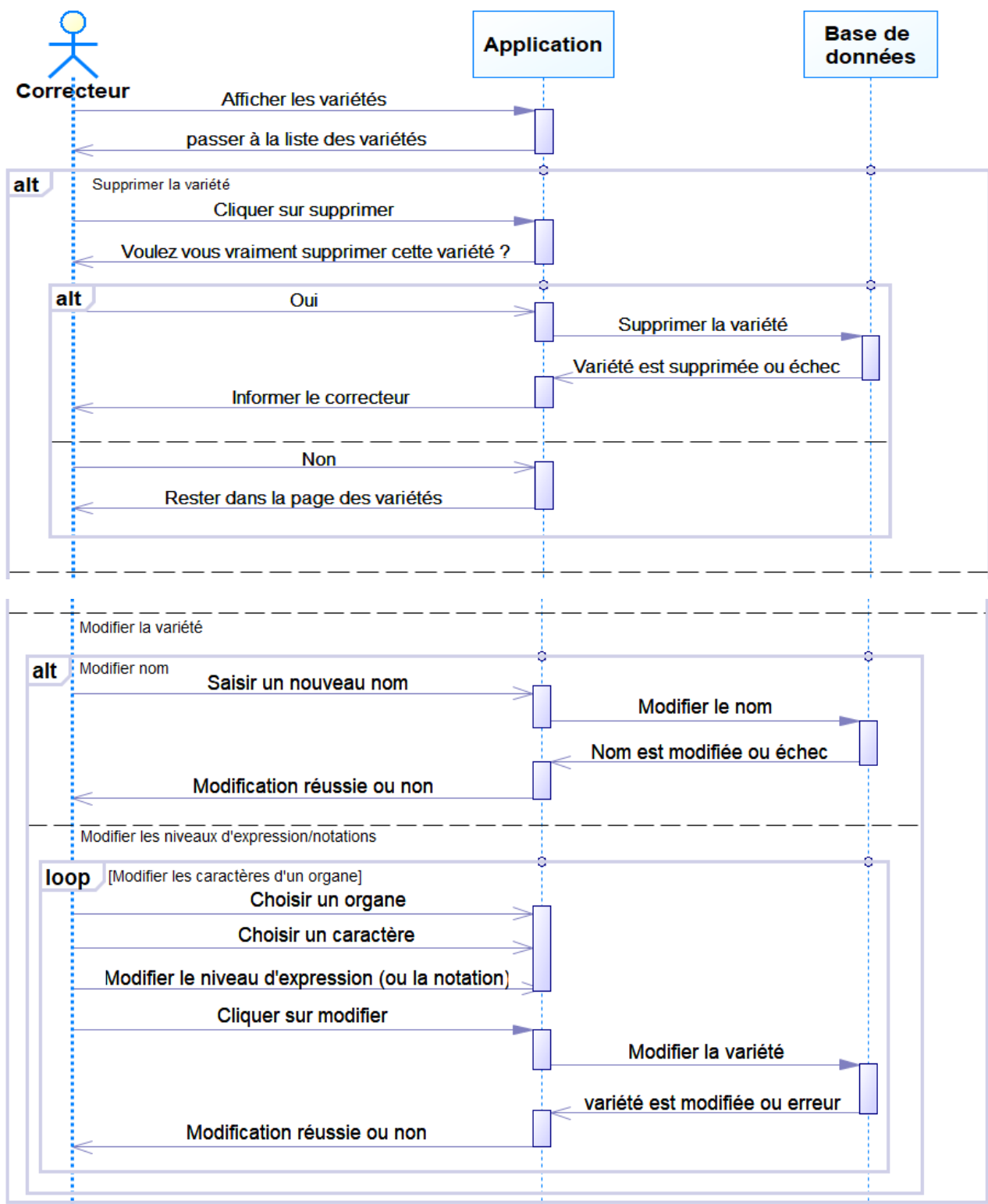


Figure 9 : Diagramme de séquences de modification ou suppression d'une variété

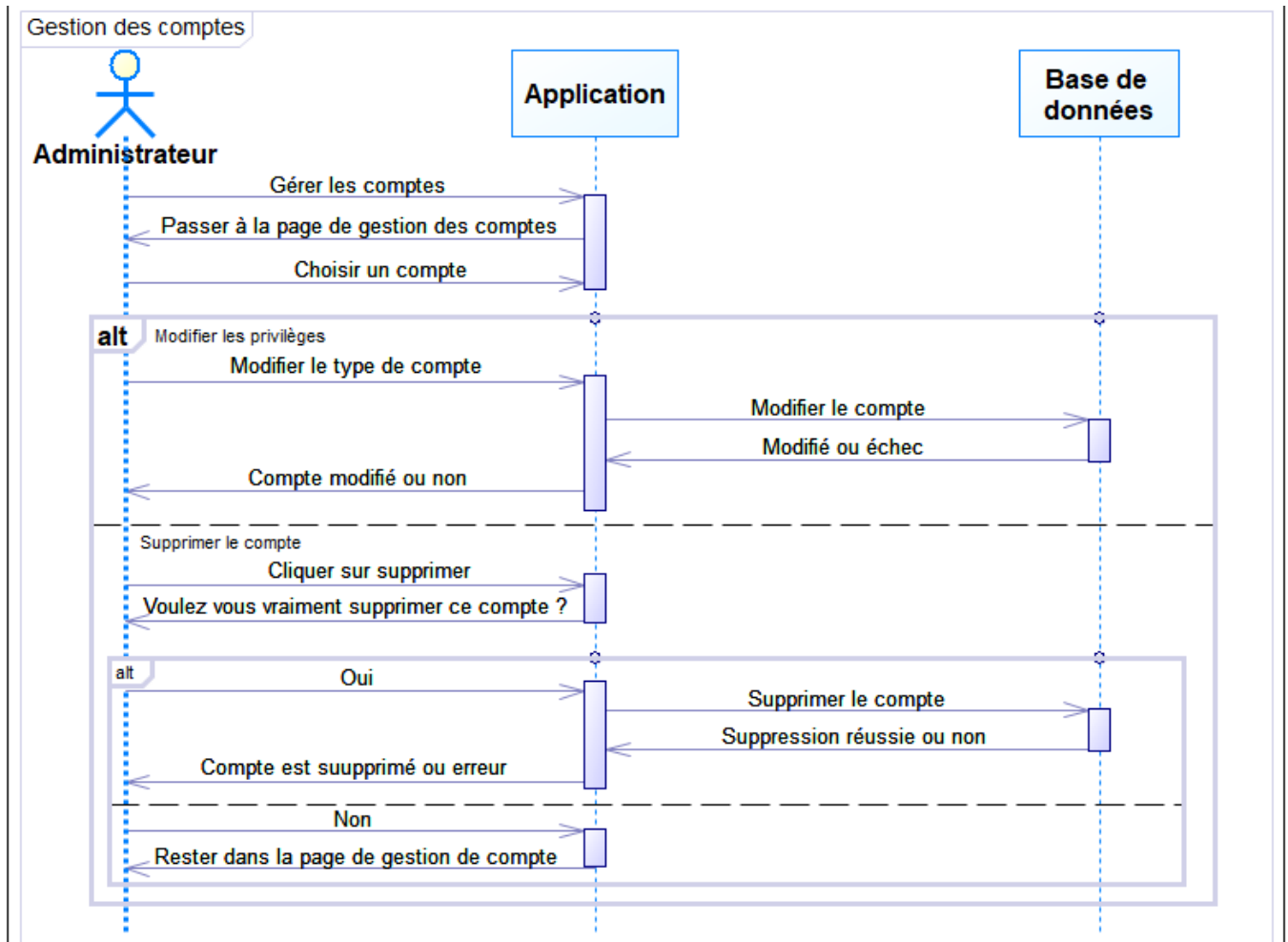


Figure 10 : Diagramme de séquence de gestion des comptes

### 3. Diagramme de classe

Le diagramme de classe représente les classes intervenant dans le système, c'est une Représentation statique des éléments qui composent un système et de leurs relations. Ce Représente les différentes classes qui modélisent notre application.

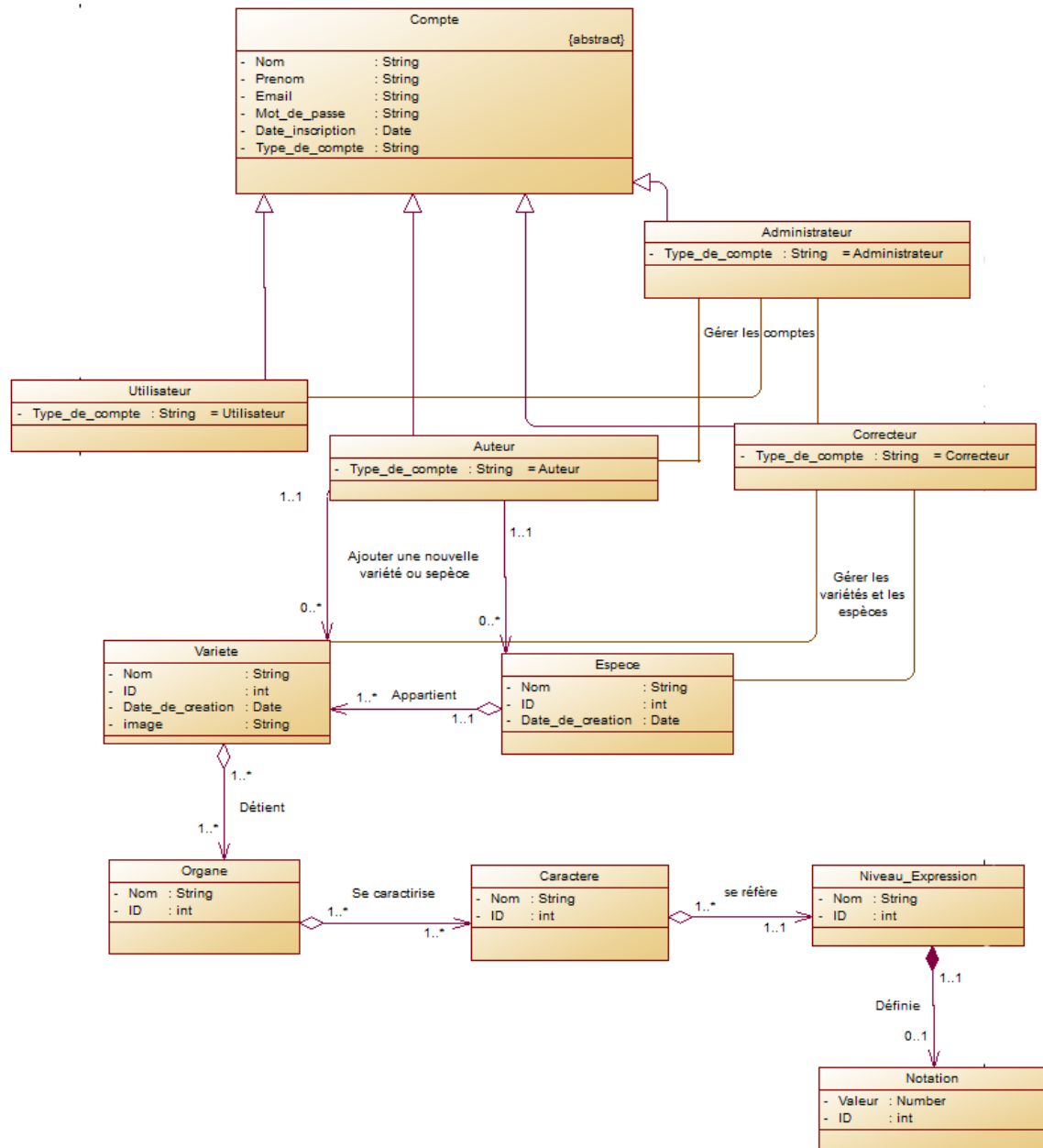


Figure 11: Diagramme de classe

## 4. Modélisation des données

On va utiliser la méthode de MERISE pour montrer les relations entre les tables au sein de la base, afin de faciliter l'organisation et la modélisation de la base selon un formalisme universel compréhensible par tous . pour ca on a les modèles suivants :

- Modèle conceptuel des données MCD :



Une représentation des données facilement compréhensible, permettant de décrire le système d'information à l'aide d'entités. Chaque entité représente un tableau dans notre base de donnée avec les relations entre eux .

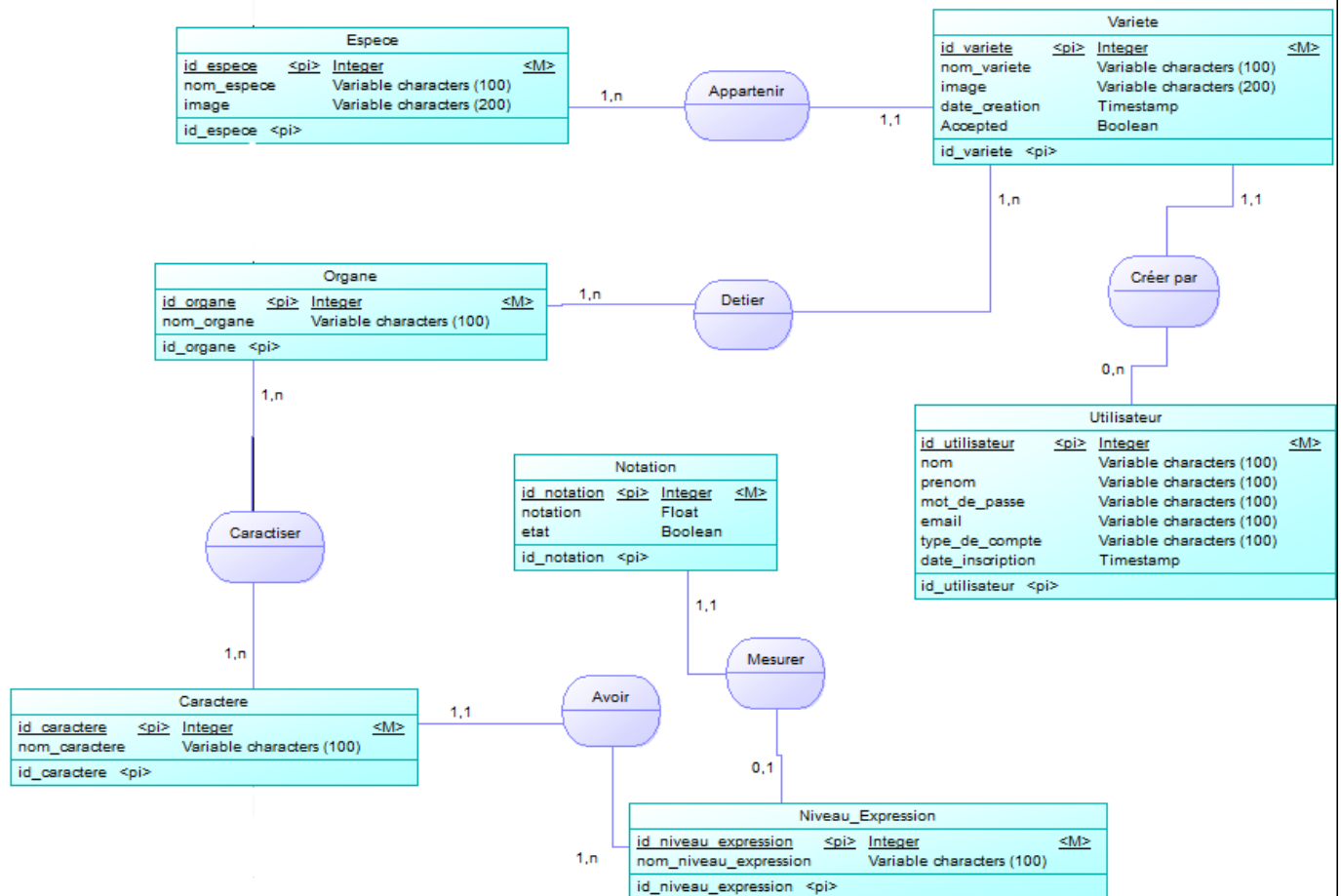


Figure 12 : Modèle conceptuel des données

#### - Modèle logique des données MLD :

Ce modèle décrit la structure de données utilisée sans faire référence à un langage de programmation . alors il précise le type de données utilisées lors de traitements .

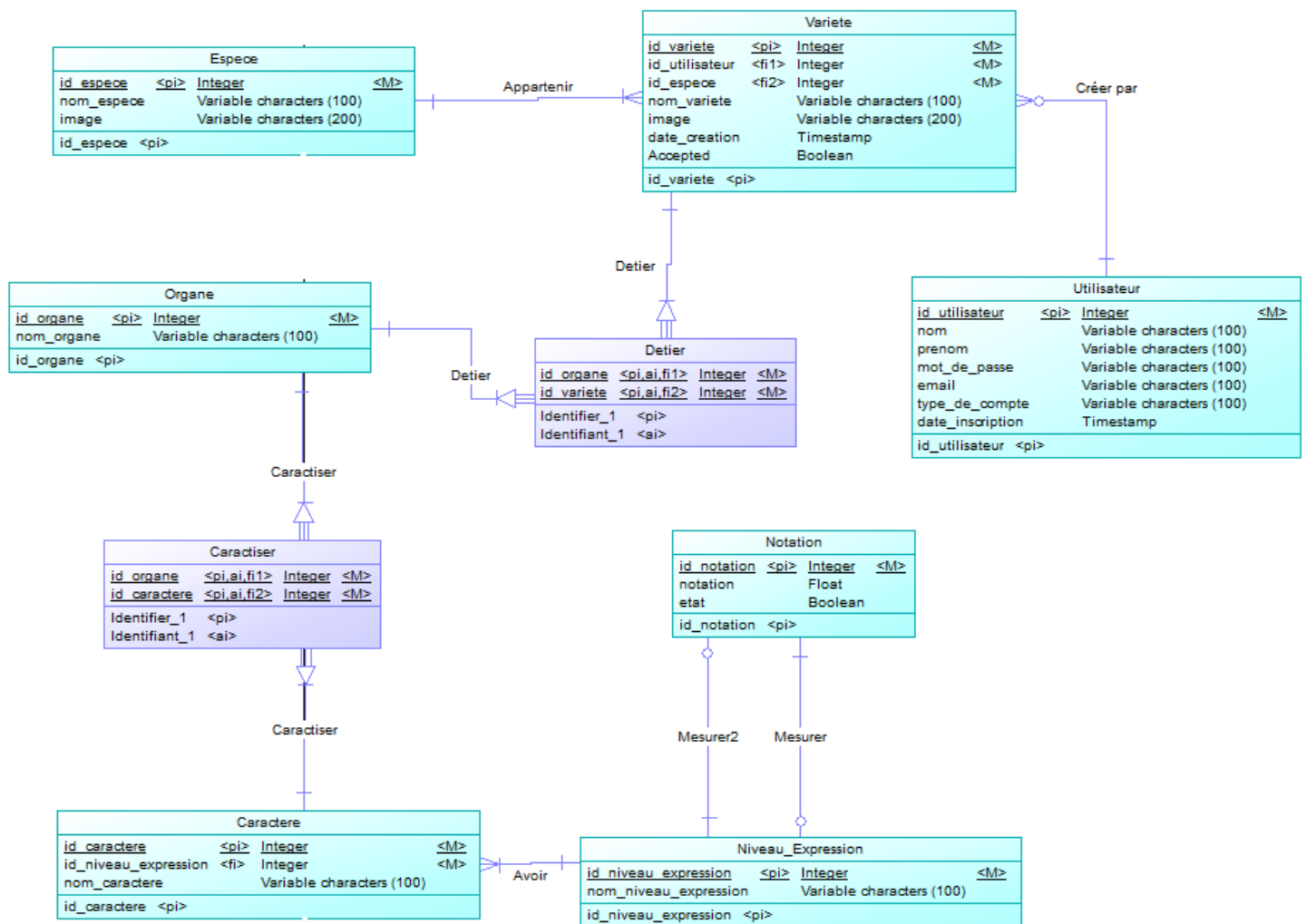


Figure 13 : Modèle logique des données

- **Modèle physique de données MPD :**

Ce modèle consiste à implanter une base de données dans un système de gestion de base de données relationnelle .



# Chapitre 3

## Etude technique et environnement

### 1. Développement mobile

Ces dernières années, le marché du mobile s'est développé rapidement. Depuis 2015, les utilisateurs surfent davantage sur des smartphones ou des tablettes que sur des ordinateurs. Nous n'utilisons plus de smartphones juste pour passer des appels. Une application mobile est développée pour atteindre un niveau de qualité, en termes d'ergonomie mais aussi de performance.

#### *a- Les points forts d'une application mobile*

**Un outil de fidélisation :** Assurer une conception ergonomique parfaite, adaptée aux applications mobiles. Par rapport aux sites mobiles, cela encourage les utilisateurs à rester. Soyez fidèle à l'application.

**Acquérir une vision :** le développement d'applications mobiles. Compte tenu de la taille du smartphone, du temps de charge et autres paramètres. L'application est répertoriée dans l'App Store en ligne. Ou des "stores" plus simples, ce qui leur permet d'avoir une meilleure visibilité du site Web et accès facile à l'application. Répondre à des besoins spécifiques.

**Un engagement de client plus fort :** Participation des clients. Désormais, nous ne devons plus nous concentrer uniquement sur les prix ou les fonctionnalités des produits, mais sur l'expérience client. Votre application est-elle facile à utiliser ? Est-ce convivial ? Avez-vous le bon outil interactif ? Si vous souhaitez augmenter l'engagement client, ce sont toutes des questions que vous devez vous poser.

**Une expérience client optimisée :** Les utilisateurs de smartphone consacrent 90% de leur temps aux applications mobiles, en comparaison avec les sites web (Statistique Furry, 2015).

#### *b- Les points faibles d'une application mobile*

- Le principal inconvénient d'une application c'est qu'elle doit respecter les règles définies par les différentes sociétés des plateformes mobiles (Apple pour les applications iOS, Google pour les applications Androïde, Windows pour les applications Windows phone, etc...).

- Un autre inconvénient est le coût lié au développement d'une application. En moyenne, il est plus coûteux de développer une application mobile destinée à être utilisable sur plusieurs plateformes (Afin d'être disponible pour un maximum de mobinautes) que de faire développer un site mobile ou un web App.
- Autre inconvénient, pour que l'utilisateur ait accès à la dernière version, il faut qu'il mette à jour depuis le store contrairement aux sites mobiles et web-app qui se mettent à jour automatiquement.

### *c- Différentes plateformes*

Tu as une idée d'application mobile puis tu as étudié sa cible tu as le développer, donc Il faut alors se lancer Mais pour rentabiliser votre temps et votre argent, il va falloir se poser une question dont une primordiale: celle du choix de la plateforme d'utilisation.

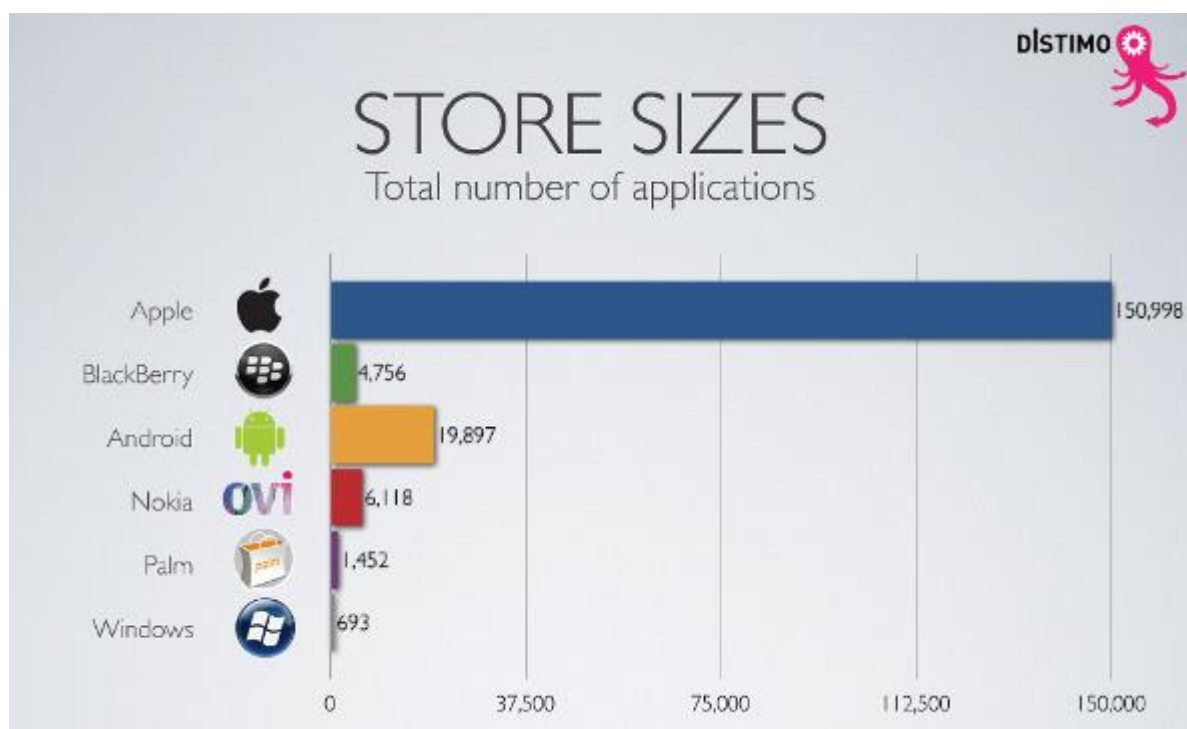


Figure 15 : différents plateforme mobile

#### ❖ BlackBerry OS :

Système d'exploitation propriétaire fonctionnant sur les Smartphones BlackBerry de la gamme BlackBerry, ce dernier il permet aux développeurs de mettre en place des applications. Toute application doit être signée numériquement par le compte RIM (Research In Motion) du développeur.

#### ❖ Android :

Android est un système d'exploitation mobile basé sur une version modifiée du noyau Linux et d'autres logiciels open source, principalement conçu pour les appareils mobiles à écran tactile tels que les téléphones intelligents et les tablettes.

Android est développé par une alliance de développeurs appelée "Open Mobile Alliance" et est sponsorisé par Google. C'est un logiciel libre et open source ; son code source s'appelle Android Open Source Project (AOSP), qui est principalement sous licence Apache.

Si vous souhaitez toucher un maximum d'utilisateurs, Android est la plateforme rêvée. Parmi les points positifs de la plateforme, on peut citer:

- Une très grande ouverture de la plateforme
- Des terminaux à des prix abordables
- Des facilités de développement et de déploiements d'application.

### ❖ iOS (Apple) :

iOS (anciennement iPhone OS) est un système d'exploitation mobile créé et développé spécifiquement par Apple Inc. pour son matériel. C'est un système d'exploitation qui prend en charge de nombreux appareils mobiles de l'entreprise (y compris l'iPhone et l'iPod Touch); les principales versions d'iOS sont publiées chaque année.

IOS est le deuxième système d'exploitation le plus répandu sur le marché. Après ANDROID, parmi des gros point négatif est le prix prohibitif des terminaux ainsi que la fermeture de l'OS qui impose des lignes de développement très précises et cadre les possibilités des applications (on ne peut pas faire tout ce que l'on veut).

### Des statistiques des ventes

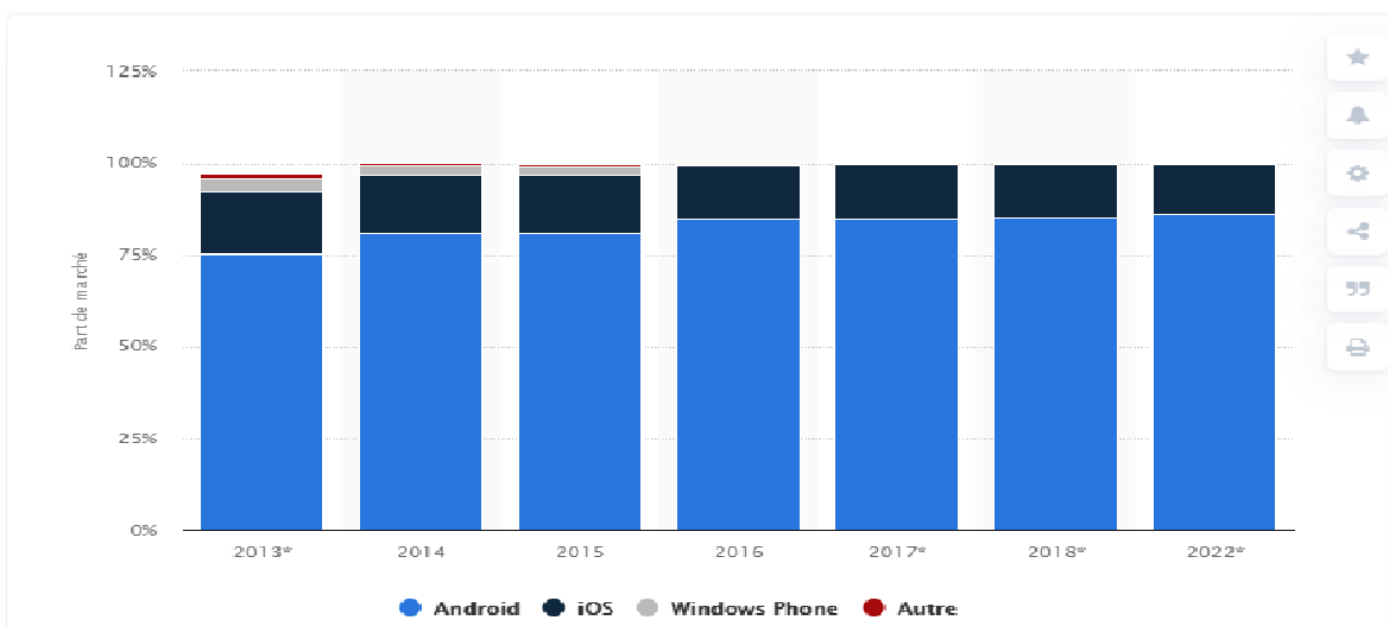


Figure 16 : présentation des statistiques des ventes

## 2. Les besoins techniques

Pour créer une application android vous devez avoir :

- Environnement de Développement (IDE)
- Un éditeur de texte pour la partie codage
- Un serveur locale (au cas où votre application est reliée à une base de données)
- Un téléphone mobile pour tester l'application ( ou utiliser l'appareil virtuel de l'IDE ).

Afin de mener à bien cette expérience, on utilise Android Studio comme Environnement de Développement.

## 3. Architecture adoptée

Tout projet Android doit respecter une hiérarchie bien précise qui permettra au compilateur de retrouver les différents éléments et ressources lors de la génération de l'application. Cette hiérarchie favorise la modularité des applications Android. A la création du projet, Android Studio crée automatiquement des dossiers pour contenir les fichiers de code Java, les fichiers XML, et les fichiers multimédias. L'explorateur de projet vous permettra de naviguer dans ces dossiers. Les dossiers que nous utiliserons le plus sont java et res. Le premier contient le code Java qui définit le comportement de l'application (situé dans le répertoire de votre projet sous app\src\main) et le second comporte des sous dossiers (dans app\src\main\res) où sont stockés les ressources qui définissent l'interface de l'application (l'apparence).

❖ La séparation entre fonctionnalité et apparence est un point essentiel de la philosophie Android.

## 4. Choix des langages

### ➤ JAVA

Le langage officiel pour le développement Android est Java .Java est un langage de programmation orienté objet inspiré du langage C++ il'est rapide, sécurisé et fiable , Les programmes créés sont portables. Java permet de créer des applications complètes. Il peut également servir à créer un petit module d'application, dit applet, à intégrer dans une page Web.

### ➤ XML

XML(Extensible Markup Language) Langage de balisage extensible, Les applications Android utilisent XML pour créer des fichiers de mise en page. Contrairement au HTML, XML est sensible à la casse, nécessite que chaque balise soit fermée et préserve les espaces.

### ➤ PHP

Permet la communication entre une application Android et une base de données. Est un langage de programmation libre, principalement utilisé pour produire des pages Web dynamiques via un serveur HTTP , mais pouvant également fonctionner comme n'importe quel langage interprété de façon locale .

### ➤ *SQL*

SQL (Structured Query Language) est un langage normalisé servant à exploiter des bases de données relationnelles. La partie langage de manipulation des données de SQL permet de rechercher, d'ajouter, de modifier ou de supprimer des données dans les bases de données relationnelles.

il y a plusieurs frameworks que vous pouvez utiliser dans le développement d'une application Android comme PhoneGap, React Native, Flutter ... . Mais dans notre projet il était inutile de les utiliser, Nous utilisons juste des outils intégrés dans Android Studio.



# Chapitre 4

## Réalisation, interfaces, tests

### 1. Environnement du travail

On a choisi nos outils de développement on se basant sur leur gratuité et l'open source. Effectivement, en implémentant une solution qui se base sur des technologies gratuites et open source, nous avons plus de chance d'éviter toutes sortes de problèmes liés aux licences, contrats, etc...

Dans cette partie pratique de la réalisation comporte on va faire une description des outils de développement utilisés pendant la réalisation de notre application mobile.

#### A. Android Studio:



Figure 17 : Android studio

Android Studio est un environnement de développement pour développer des applications mobiles Android. Il est basé sur IntelliJ IDEA et utilise le moteur de production Gradle

La première vision de l'interface d'Android studio peut sans doute vous effrayer par la tonne d'outil, d'option qu'elle propose. Mais Il n'est pas nécessaire de tout connaître de l'environnement. Comme l'environnement propose beaucoup d'outils et d'interfaces, On va présenter les interfaces et les outils qui sont essentiels à savoir maîtriser.

#### **Le premier outil AVD Manager :**

Vous pouvez accéder à cet outil en cliquant sur l'icône (en haut à droite, rectangle rouge) sur la barre des raccourcis de votre environnement de développement.

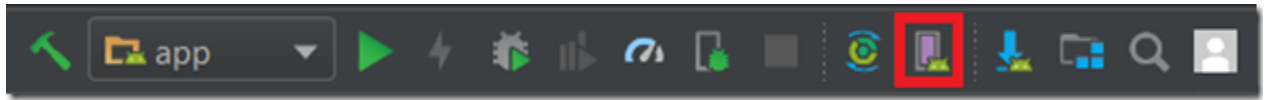


Figure 18 : AVD Manager

Il va vous permettre depuis son interface de créer et de démarrer en même instant autant d'émulateur que vous souhaitez. Les seules limites seront la mémoire vive (RAM) et la puissance de calcul de votre processeur de votre ordinateur.

### **Le deuxième outil le débugeur :**

Le débugeur permet dans 99% des cas de résoudre, les principales erreurs de codage (que l'on appelle bug). Sa fonction principale est de vous aider à repérer les erreurs de codage que vous avez pu effectuer. Vous pouvez démarrer votre application en mode debug (en utilisant le débugeur) grâce au raccourci ci-dessous :

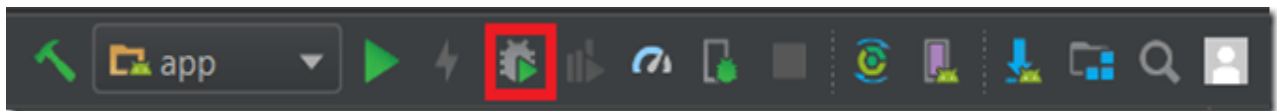


Figure 19 : le débugeur

Le débugeur sur Android studio propose plusieurs fonctionnalités. La plus essentielle reste tout de même le mode pas à pas. Cette fonctionnalité, vous permet de parcourir votre programme ligne par ligne et de trouver la ligne en question qui pose problème. C'est notamment utile, lorsque vous placez des breakpoints dans votre code (point d'arrêt en français).

### **l'interface pour les fichiers exécutifs**

Dans la vue de gauche vous pouvez apercevoir, la vue hiérarchique de votre application. les fichiers java constituent la partie exécutive de votre application. Lorsque vous double-cliquez sur le fichier, le fichier s'ouvre sur la vue de droite. Vous pouvez modifier ce fichier à votre convenance. Comme par exemple attribuer une fonctionnalité à un composant graphique.

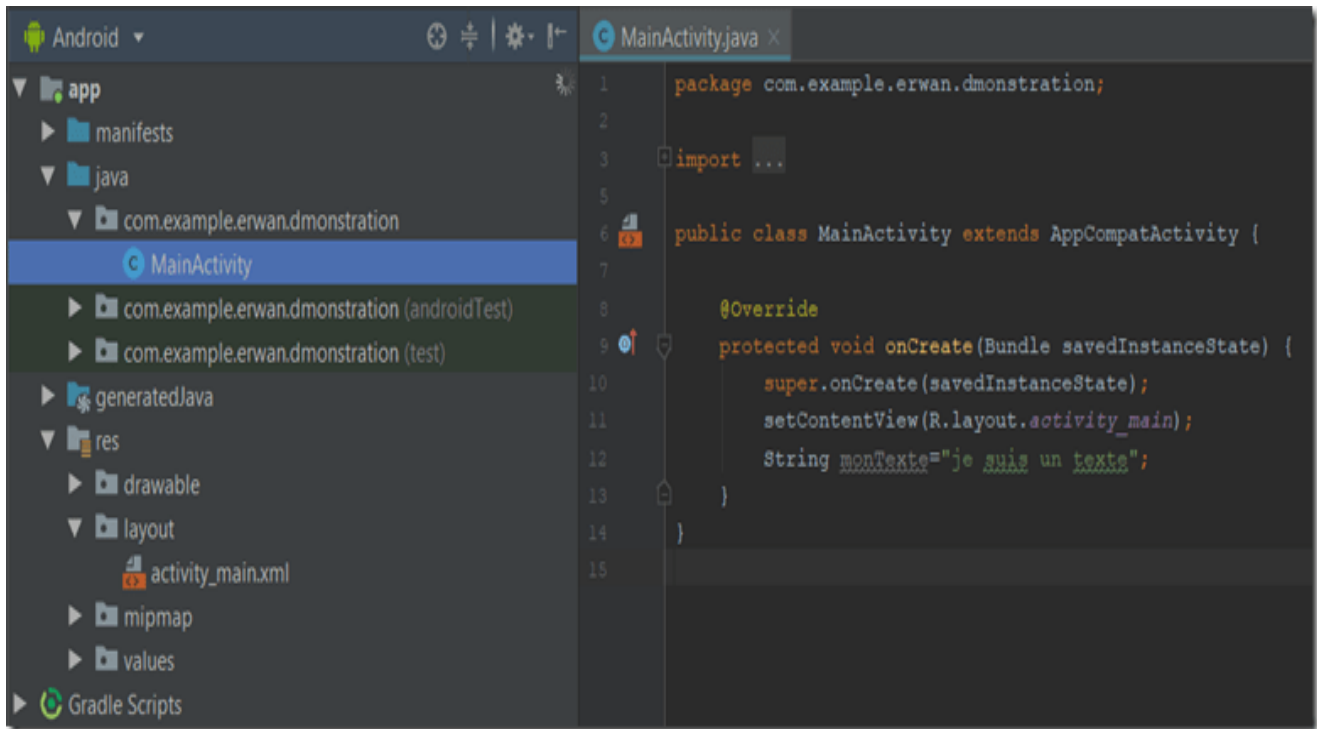


Figure 20 : l'interface des fichiers exécutifs

### ***l'interface intuitive pour le design de votre application***

Les fichiers layout sont la partie visible de votre application. Lorsque, vous ouvrez le fichier en double-cliquant, vous avez accès à deux onglets. Le premier l'onglet "Design", vous permet de voir un rapide aperçu de l'écran de ce que cela va donner. À gauche de l'aperçu, vous pouvez voir "la palette". Cette interface, vous propose un large choix de composants graphiques que vous pouvez implémenter. Un simple cliquer-déplacer d'un composant sur votre aperçu, vous permet de l'implémenter. À droite la vue "Attributs", permet de définir les caractéristiques du composant.

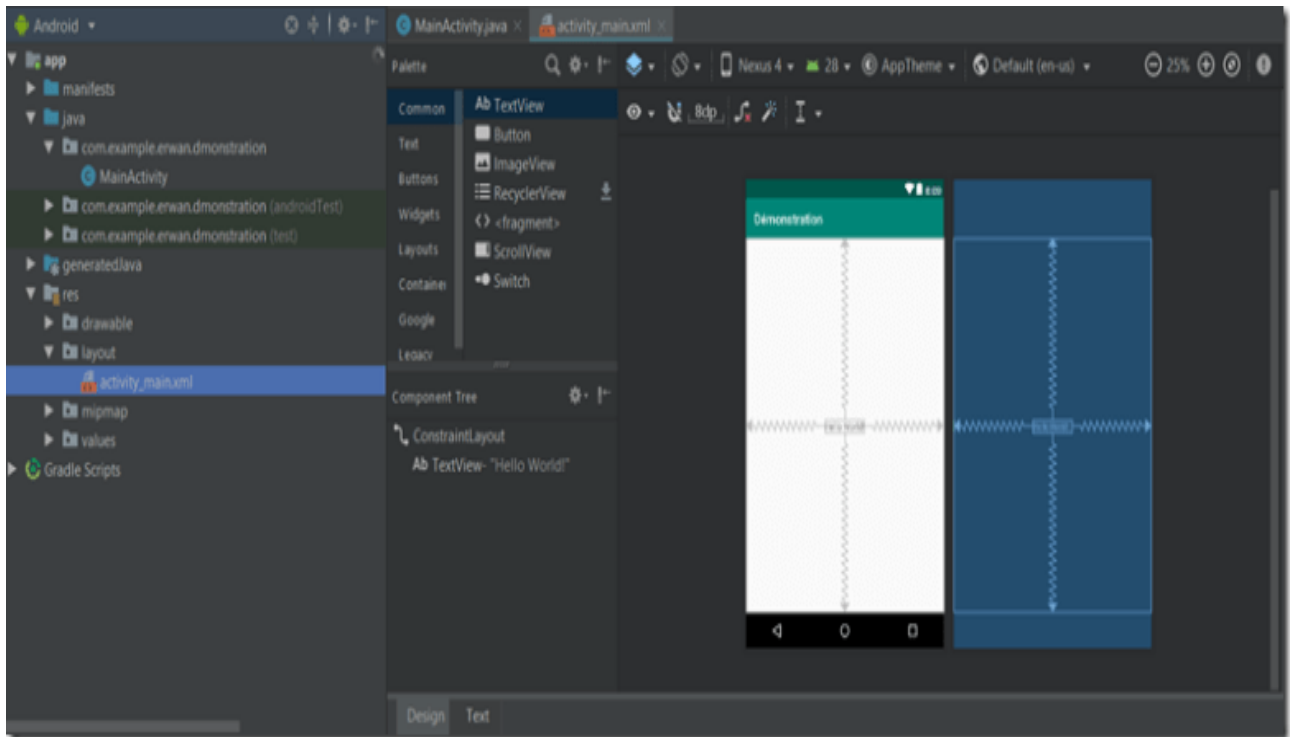


Figure 21 : l'interface intuitive pour les fichiers de désigne

### ***l'interface brute de texte pour le désigne de votre application***

Le deuxième onglet, l'onglet "texte" vous permet également de jouer sur le design de votre écran. Lorsque vous ajoutez un composant graphique dans l'onglet "Graphique", il vient s'implémenté également au format xml. De même, lorsque vous rajoutez un composant dans votre xml, vous l'apercevrez sur votre aperçu.

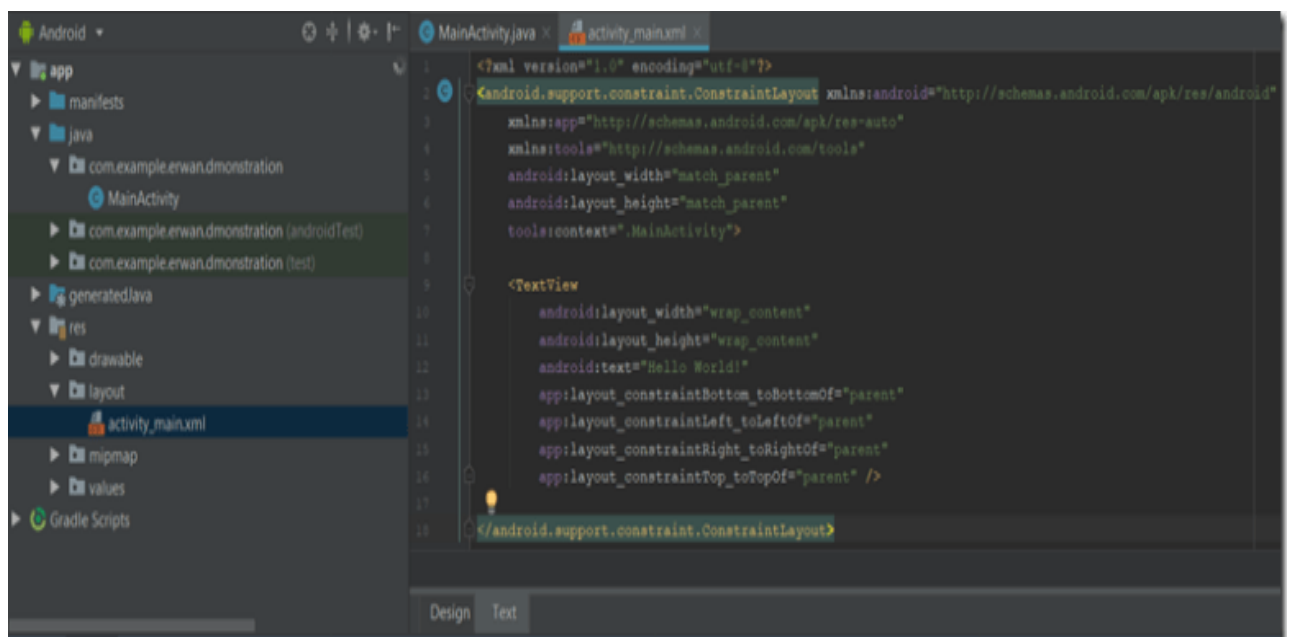


Figure 22 : l'interface brute pour les fichiers de désigne

## Lexique

### **Activity :**

Une activité représente un écran contenant une interface utilisateur. Une application est composée d'un ensemble d'activités. Les activités peuvent interagir entre elles, intra-application ou inter-applications.

### **apk file :**

C'est un fichier en format application package produit à l'issue de la compilation d'une application Android. C'est le fichier destiné aux utilisateurs. Ce fichier inclut le code de l'application sous forme d'un exécutable DVM (.dex), les ressources multimédias, et le AndroidManifest.xml.

### **Fragment :**

Un fragment représente un comportement ou une portion de l'interface utilisateur dans une activité. On peut combiner plusieurs fragments dans une même activité pour créer une interface à volets multiples et réutiliser un fragment dans plusieurs activités.

### **Intent :**

C'est un message qui permet d'activer un composant (une activité par exemple). C'est un message qui demande une action à un autre composant (de la même application ou d'une autre)

### **Layout :**

Le layout définit la structure visuelle d'une interface utilisateur. Les éléments d'un layout peuvent être définis soit dans un fichier XML ou bien à l'exécution à partir d'instructions dans le code. L'utilisation XML permet de mieux séparer l'apparence de l'application d'une part et le code qui définit son comportement d'autre part. Ceci permet de s'adapter plus facilement aux différentes cibles, il suffit de définir plusieurs fichiers XML pour un même code.

### **Manifest :**

Toute application Android doit posséder un fichier XML nommé AndroidManifest.xml. Ce fichier contient des informations essentielles sur l'application que le système doit connaître pour pouvoir la lancer. Ce fichier définit le nom du package de l'application, décrit les composants de l'application, détermine le processus qui accueillera ces composants, déclare les permissions dont l'application a besoin et celles que les autres doivent avoir pour interagir avec l'application, déclare la version Android minimale pour l'application, et liste les bibliothèques dont l'application a besoin.

### **Resources :**

Ce sont les ressources en relation avec la présentation de l'application (images, fichiers audio etc.). Pour chaque fichier de ressource qu'on inclut dans le projet, un identifiant unique est créé pour référencer cette ressource. Le fait d'avoir des ressources séparées du code nous donne la possibilité de proposer des versions différentes en fonction de la configuration de la cible. En

utilisant par exemple des images différentes selon la résolution de l'appareil, une mise en forme adaptée en fonction de l'orientation de l'affichage (portrait ou paysage), des fichiers de chaînes de caractères en plusieurs langues pour que l'interface s'affiche dans la langue de l'utilisateur, etc.

### **Service :**

Un service est un composant qui tourne en tâche de fond, pour effectuer de longues opérations ou exécuter des tâches pour un autre processus. Un service ne possède pas une interface utilisateur.

## **B. XAMPP**



Figure 23 : XAMPP

est un ensemble de logiciels libres. Le nom est un acronyme venant des initiales de tous les composants de cette suite. Ce dernier réunit le serveur Web Apache, la base de données relationnelle et système d'exploitation MySQL (ou MariaDB) ainsi que les langages scripts Perl et PHP. L'initiale X représente tous les systèmes d'exploitation possibles, à savoir Linux, Windows et Mac OS X.

XAMPP Control Panel vous permet de contrôler les composants de votre serveur de test. L'interface claire recueille toutes les actions et vous permet de démarrer ou d'arrêter certains modules en un clic. Cette interface propose des actions diverses :

**Config :** permet la configuration de votre XAMPP ainsi que des composants partiels

**Netstat:** affiche les processus courants sur l'ordinateur local

**Shell:** ouvre une cellule UNIX

**Explorer:** ouvre le dossier XAMPP dans Windows Explorer

**Services :** affiche les services fonctionnant en arrière-plan

**Aide :** propose des liens vers des forums pour utilisateurs

**Terminer :** ferme le XAMPP-Control-Panel

Vous pouvez lancer et stopper chaque module depuis cette interface dans la colonne « actions ». Les modules déjà démarrés sont surlignés en vert.

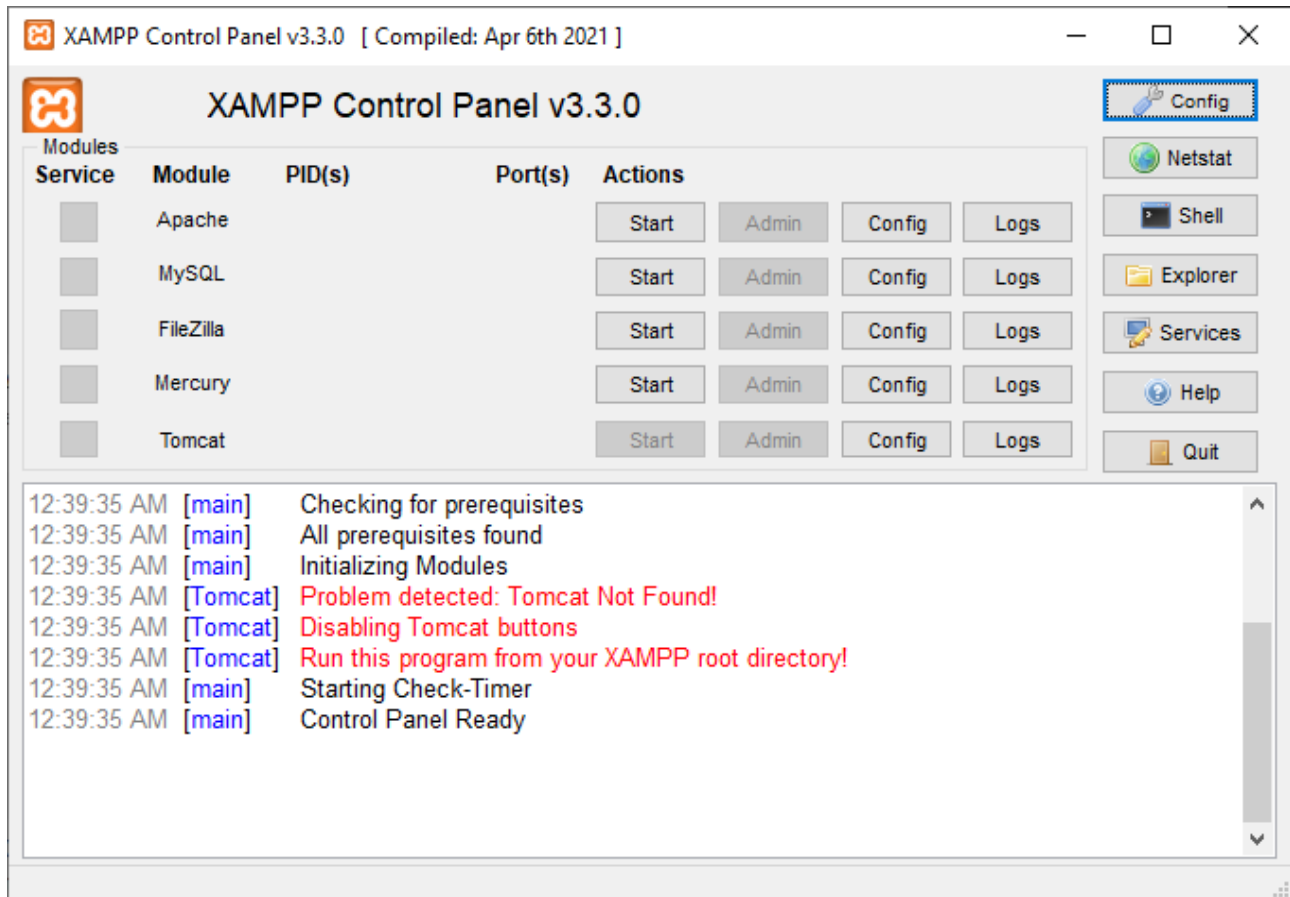


Figure 24 : XAMPP Contrôle Panel

Dans notre projet nous avons utilisé les deux modules Apache et MySQL.

- **Apache** : le serveur Web open source Apache est utilisé mondialement et permet de délivrer des contenus Web.
- **MySQL** : XAMPP se compose de l'un des systèmes de gestion de base de données relationnelle les plus populaires au monde. En combinaison avec le serveur Web Apache et le langage script PHP, MySQL sert à l'enregistrement de données pour des services Web. Les versions actuelles de XAMPP favorisaient MariaDB à l'insu de MySQL comme gestionnaire de base de données.

Une zone d'administration spécifique à chaque module de votre XAMPP est à disposition.

Cliquez sur le bouton Admin de votre serveur Apache pour parvenir à l'adresse URL de votre serveur Web. Dans ce cas, le Control-Panel lance votre navigateur Web standard. Vous êtes ensuite dirigé vers le tableau de bord de l'hébergeur local de votre XAMPP. Le tableau de bord comporte de nombreux liens vers des pages Web avec des informations utiles ainsi que le projet open source BitNami, qui contient diverses applications gratuites pour votre XAMPP comme WordPress ou d'autres systèmes de gestion de contenu. Vous pouvez de manière alternative accéder au tableau de bord sous `localhost/dashboard/`.

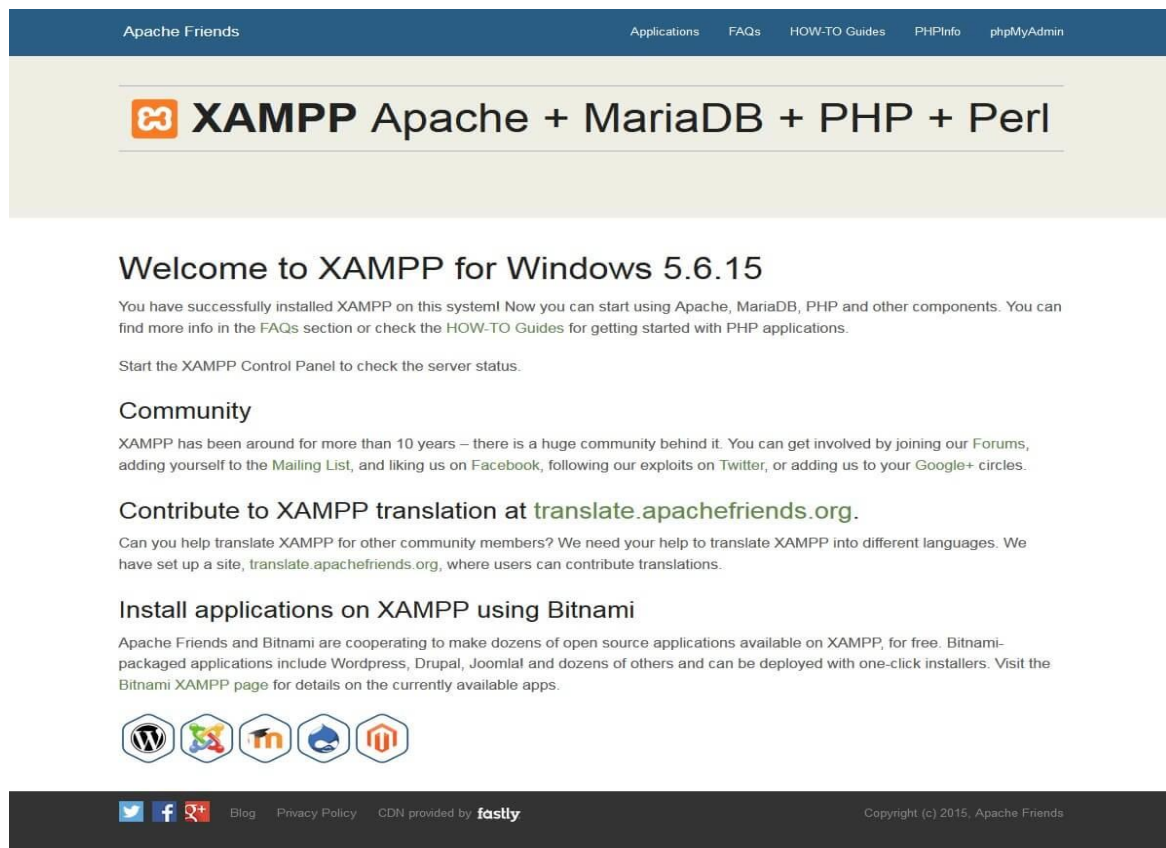


Figure 25 : Admin page de Apache

Utilisez le bouton d'administration de votre module de base de données pour ouvrir PhpMyAdmin. Ici, vous gérez les bases de données des projets que vous souhaitez tester sur votre XAMPP. Vous pouvez aussi accéder à l'interface d'administration pour votre base de données MySQL sous *localhost/phpmyadmin/*.



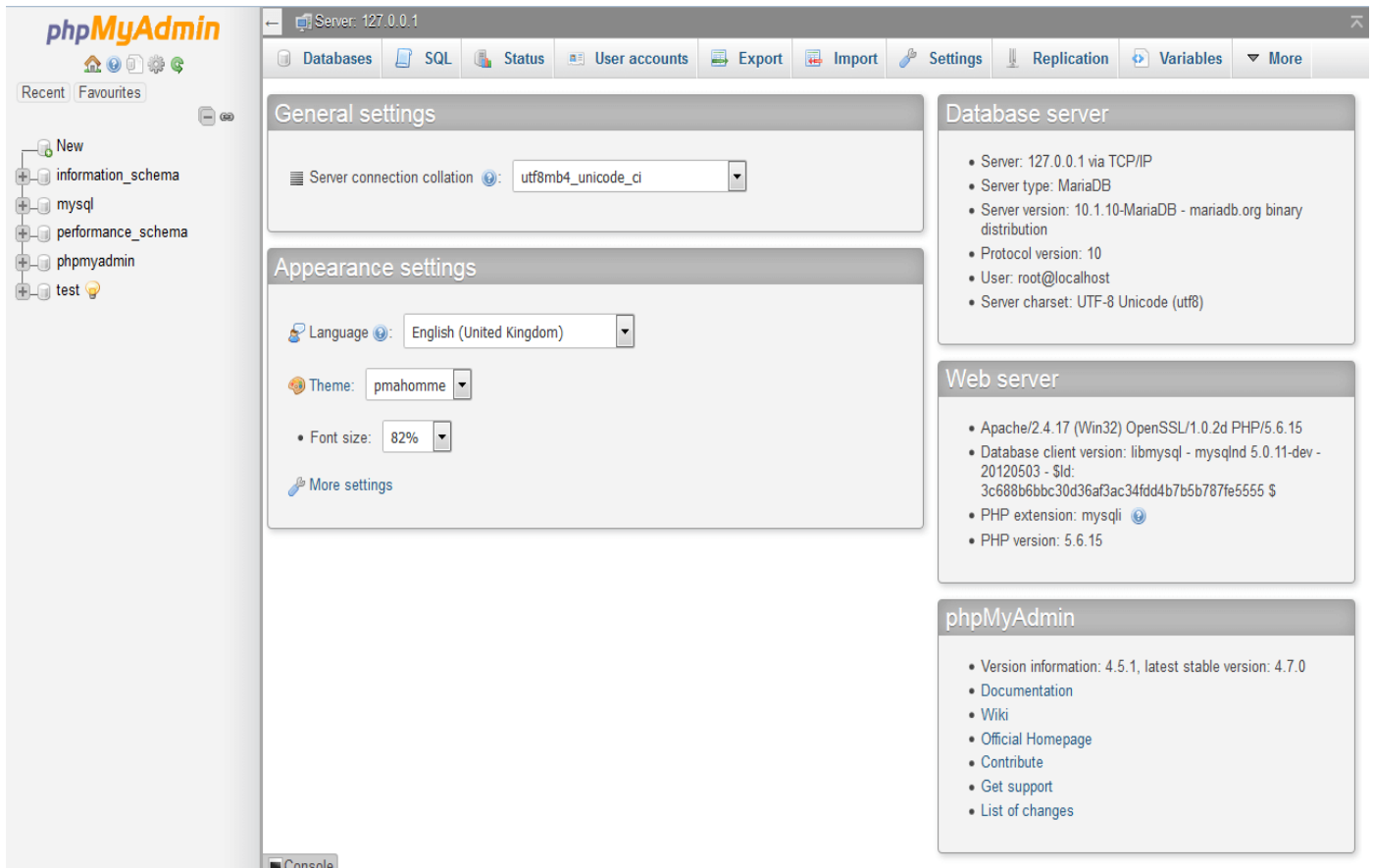


Figure 26 : L'interface de phpMyAdmin

### C. Power AMC



Figure 27: PowerDesigner

**PowerAMC** est un logiciel de modélisation. Il permet de modéliser les traitements informatiques et leurs bases de données associées. Créé par SDP sous le nom AMC\*Designor, racheté par Powersoft, ce logiciel est produit par Sybase depuis le rachat par cet éditeur en 1995 . Il permet de réaliser tous les types de modèles informatiques. A noter qu'il reste un des seuls qui permet de travailler avec la méthode MERISE. Selon Riff News, cela permet d'améliorer la modélisation, les processus, le coût et la production d'applications.

Tous les diagrammes réalisés dans ce projet sont réalisés avec powerAMC (sauf le diagramme de grant qui a été fait avec Excel.).

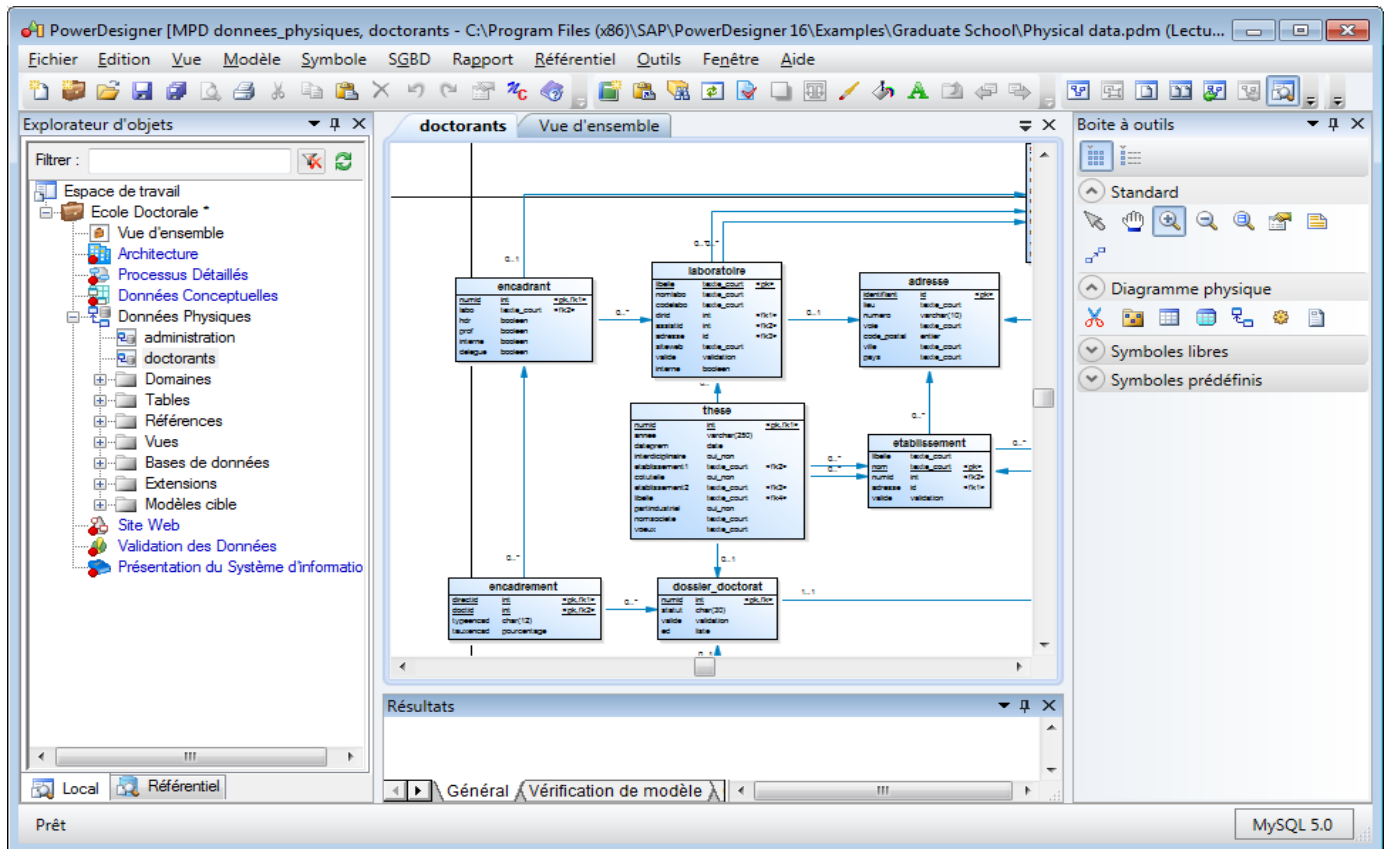


Figure 28 : L'interface de PowerDesigner

#### D. Visual Studio Code:

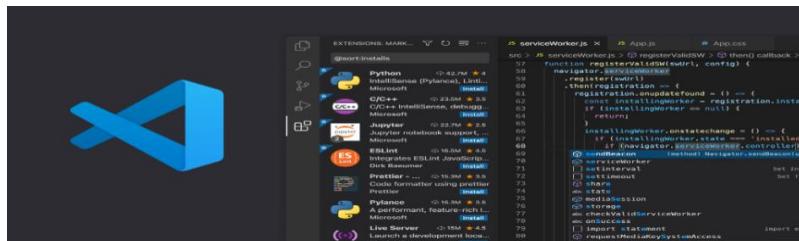


Figure 29 : Visual studio code

Est un éditeur de code extensible développé par Microsoft pour Windows, Linux et macOS. Il support la prise en charge du débogage, la mise en évidence de la syntaxe, la complétion intelligente du code, les snippets, la refactorisation du code et Git intégré.

Visual Studio Code peut être installé en tant que logiciel desktop sous Windows, Mac ou Linux. Il existe également la possibilité de l'utiliser directement comme application web dans le navigateur.

VSC propose différents éléments qui peuvent être intéressants pour de développeurs à tous niveau, si bien que, comparé à d'autres éditeurs de texte (e.g. Brackets), le niveau visé est plutôt intermédiaire/avancé. Néanmoins, VSC peut représenter un bon choix de départ même pour un

débutant dans la perspective d'atteindre ensuite une certaine expertise. De plus, les fonctionnalités d'édition avancées de VSC peuvent être également exploitées dans d'autres domaines tels que le formatage/nettoyage de fichiers textuels ou données brutes.

Parmi les caractéristiques principales du logiciel figurent :

**IntelliSense** : une technologie avancée qui propose, outre à la mise en évidence de la syntaxe et la complétion automatique du code, un système d'inférence articulé et basé directement sur la logique du code source ;

**Intégration native avec Git** : le logiciel implémente le système de gestion de versions Git directement dans l'interface de l'éditeur, ce qui représente un avantage pour pouvoir effectuer les opérations de versioning plus aisément ;

Ligne de commande intégrée : toujours dans l'interface de l'éditeur, il est possible de lancer la ligne de commande et exécuter toutes les commandes disponibles sur le système d'exploitation ;

**Eco-système d'extensions** : les extensions sont au coeur du projet et il existe même un système assez simple pour développer/publier ses propres extensions

**Debugging intégré** : pour les développeurs plus avancés, il existe également des fonctionnalités de debugging directement à l'intérieur de l'éditeur.

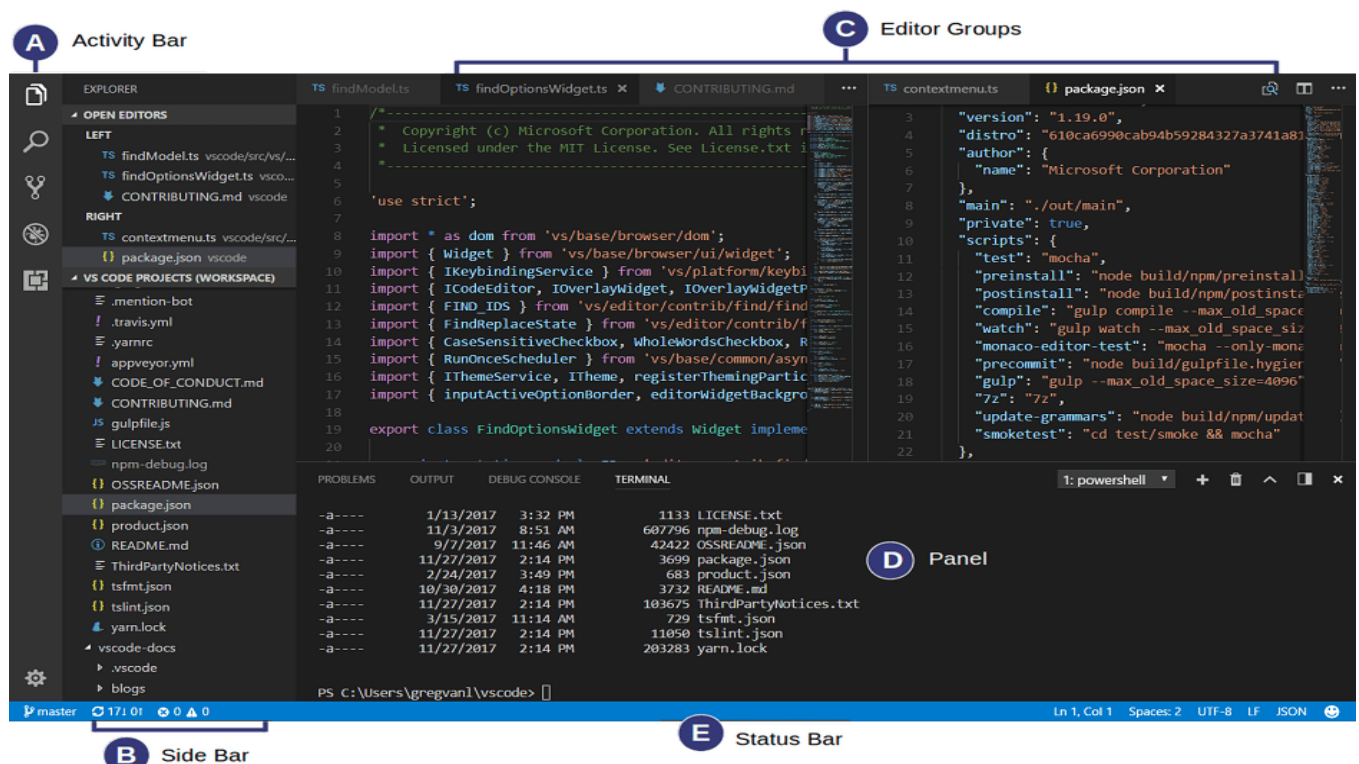


Figure 30 : L'interface de VS code

❖ Nous nous sommes servis de vscode pour les scripts php et sql.

## 2. Base de données :

La base de données nommée « PFE\_DB » est composée de 6 tables : Espèce, Variété, Caractère, Niveau d'expression, Notation, et Utilisateur.

Entre ces tables, il y a 6 associations résultant de trois types de relation : 1 à 1, 1 à plusieurs ou 0 à plusieurs .

Ces relations ont été représentées sous forme de phrases explicatives suivants :

- Chaque espèce peut avoir une à plusieurs variétés.
- Une variété appartient à une et une seule espèce.
- Chaque variété détient un à plusieurs organes.
- Un organe se présente chez une ou plusieurs variétés.
- Un organe se caractérise par un à plusieurs caractères.
- Un caractère définit un à plusieurs organes.
- Un caractère peut avoir un et un seul niveau d'expression .
- Un niveau d'expression définit zéro ou une seule notation .
- Une notation se réfère à un et un seul niveau d'expression .
- Un utilisateur (auteur) peut ajouter zéro à plusieurs variétés
- Une variété peut être ajoutée par un et un seul utilisateur (auteur) .

## 3. Présentation de l'application

Une activité (*activity*) représente un écran dans une application Android. l'emploi des *layouts* permet de garantir un rendu adapté selon le type d'appareil et les dimensions de l'écran.

Pour développer des activités, il est aussi nécessaire de comprendre que l'environnement Android se distingue du développement d'applications de bureau ou d'applications Web. En effet, l'environnement Android est généralement plus limité en ressources qu'un ordinateur classique.

Les activités sont des composants, c'est-à-dire qu'il n'est pas possible d'instancier directement une activité dans le code d'une application (avec le mot-clé *new* par exemple). Une activité est créée et gérée par le système. Un développeur doit donc être soucieux de respecter la façon dont le système va gérer les activités de l'application. Le système Android définit un cycle de vie d'une activité : cette dernière passe par des états différents. Une activité peut être prévenue du passage dans un des états suivants :

- initialisée (*initialized*)
- créée (*created*)
- démarrée (*started*)
- relancée (*resumed*)
- mise en pause (*paused*)
- arrêtée (*stopped*)
- détruite (*destroyed*)

Le schéma ci-dessous détaille les transitions possibles entre chaque état avec l'appel des méthodes de *callback*.

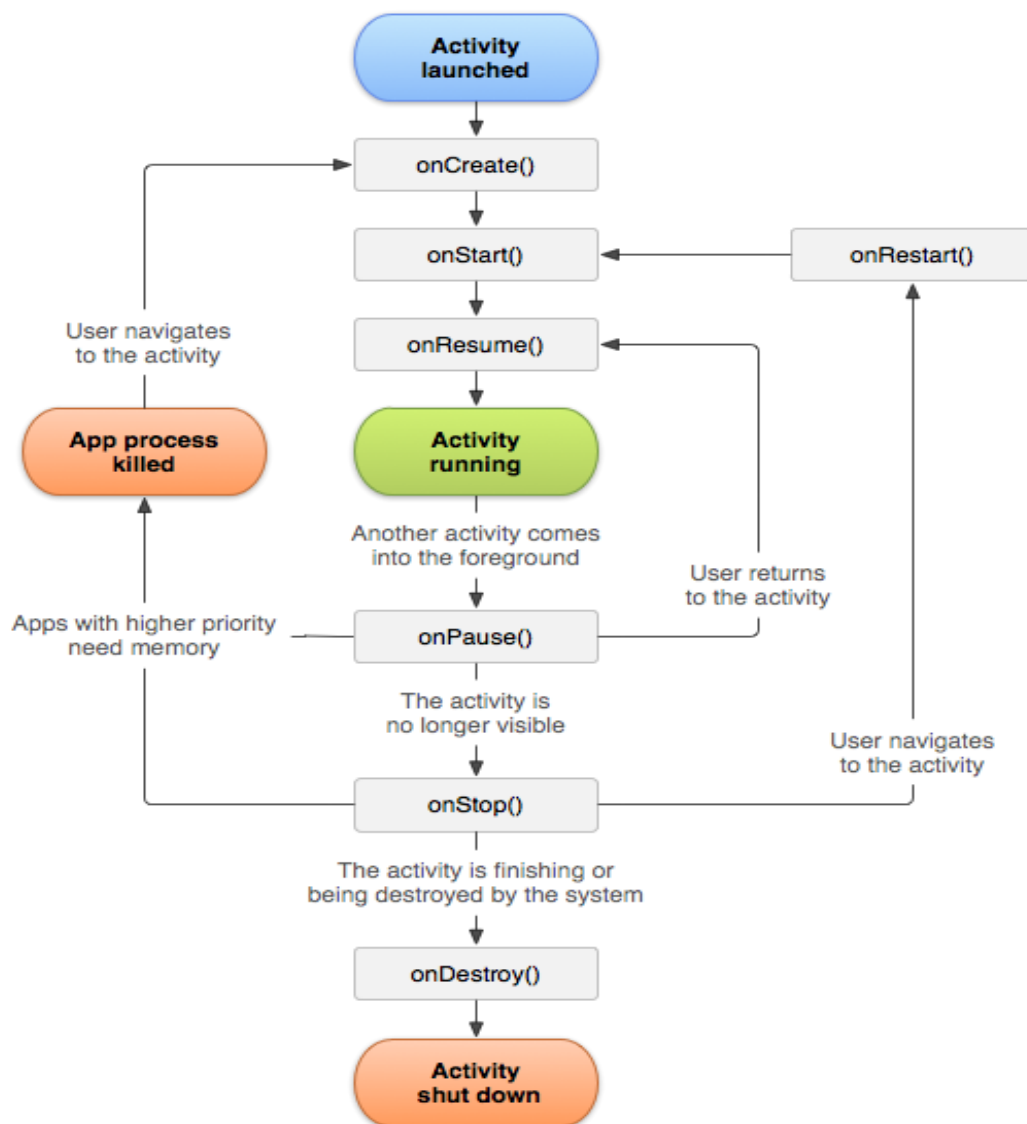


Figure 31 : Le cycle de vie d'une activité

❖ L'exécution de l'application est faite sur un appareil réel (nous n'avons pas utilisé l'émulateur).

#### a) Interface d'authentification :

L'interface ci-dessous constitue la fenêtre d'accueil (Main Activity) de notre application . A partir de cette interface vous pouvez accéder à votre espace personnel , en saisissant un e-mail et un mot de passe

#### b) Interface d'inscription :

Cette interface est l'interface d'inscription des utilisateurs, Pour qu'un utilisateur peut s'inscrire il est obligatoire d'entrer son Nom, Prénom, Adresse e-mail et Mot de passe . Après l'inscription, l'utilisateur aura un compte de type utilisateur (utilisateur régulier).



Figure 32 : Ecran de connexion

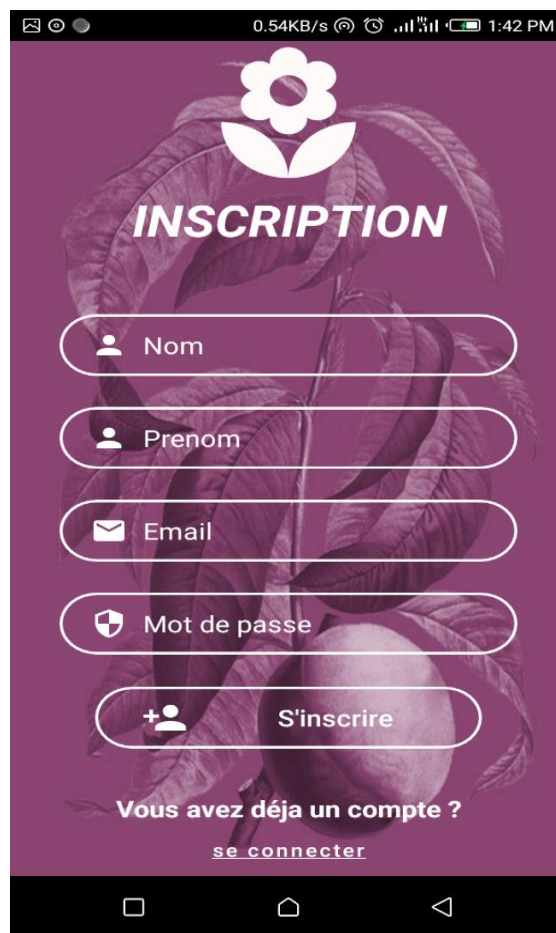


Figure 33 : Ecran d'inscription

### c) Espace Utilisateur :

L'inscription permet de créer un compte de type « Utilisateur », un utilisateur possède les privilèges suivants :

- Navigué dans la liste des variétés par un clic sur le button « liste des variétés »
- Comparer deux variétés d'une espèce par le clic sur le button « comparer » et choisit les variétés à comparer .
- Envoyer une demande de changement de type de compte à un compte de type « Auteur » et donc avoir le privilège d'ajouter une nouvelle variété ou espèce , par le clic sur le button « demande de privilège ».
- Consulter ou changer les informations du compte par le clic sur « votre profil ».

❖ Le type de compte sera changé juste si un administrateur accepte la demande .

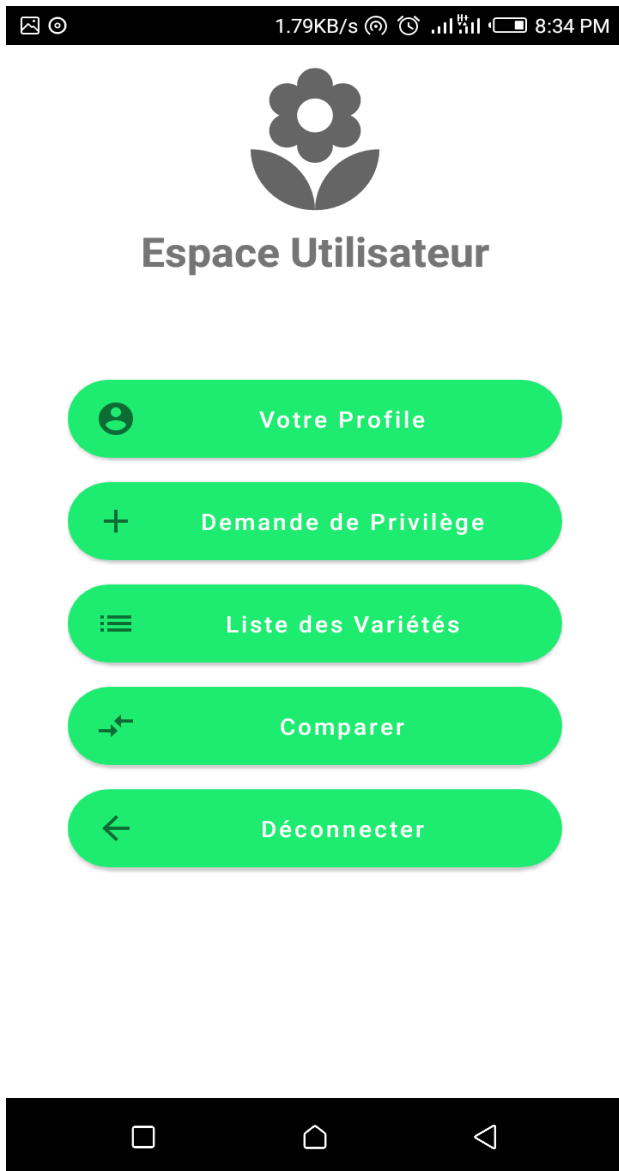


Figure 34 : Espace Utilisateur

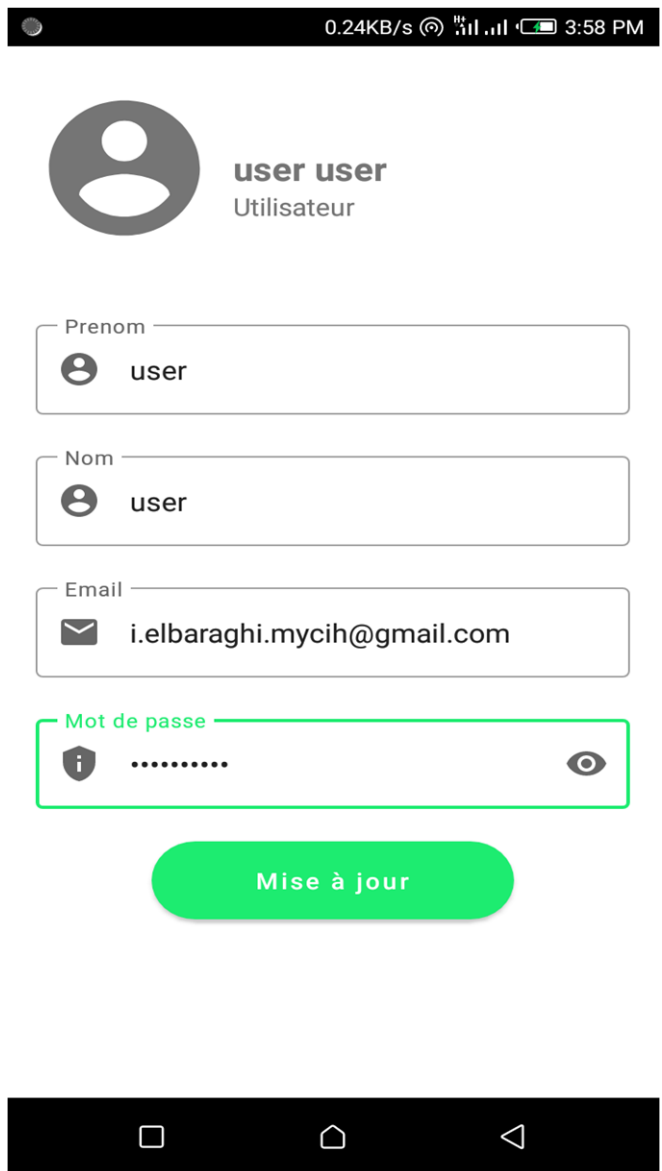


Figure 35 : Votre Profile



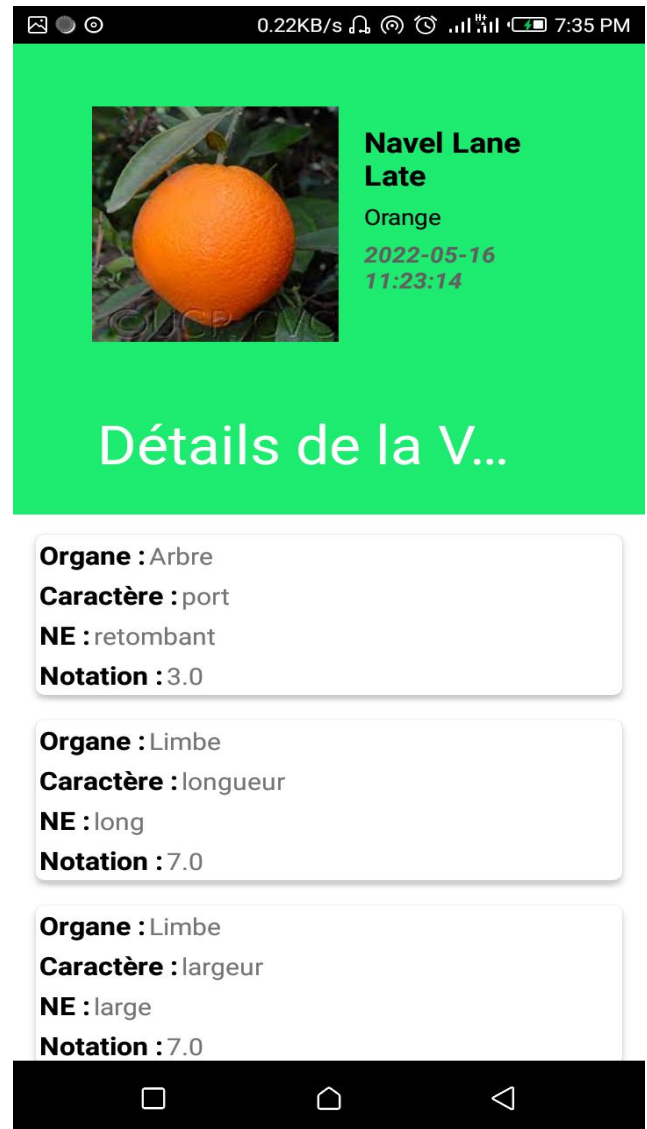
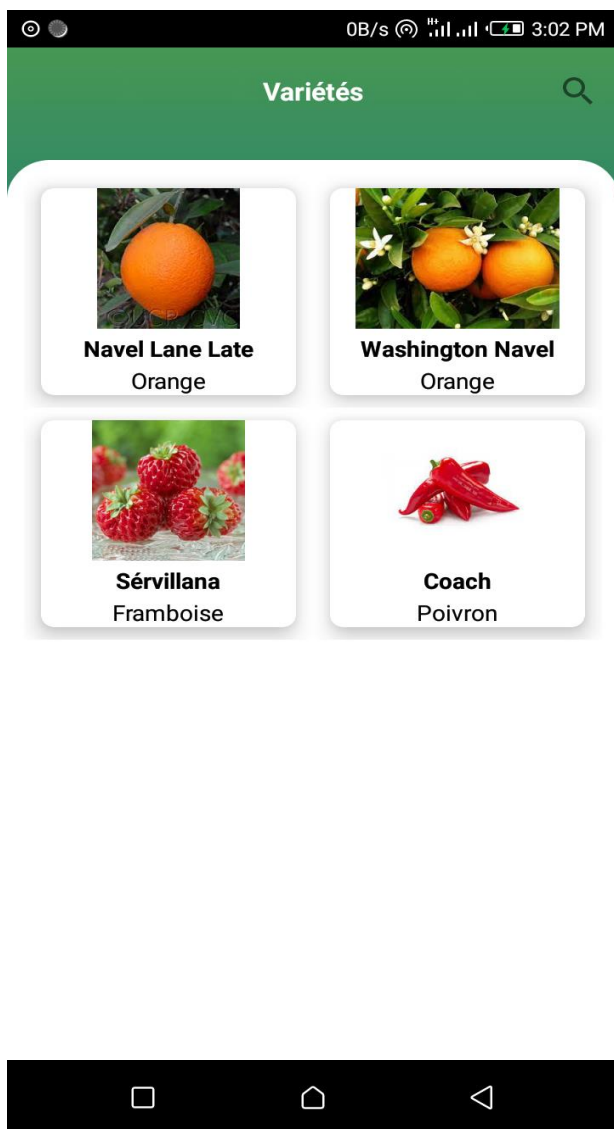


Figure 36 : List des variétés




0B/s 3:03 PM

# **Comparer les variétés**

Espèce  
 Orange

Variété 1  
 Navel Lane Late

Variété 2  
 Washington Navel


**COMPARER**

□ ◀ ▶

0B/s 3:02 PM

## Résultat de comparaison

Organe: Caractère	NE_Navel Lane Late (note)	NE_Washington Navel (note)
Arbre: port	retombant(3.0)	étalé(1.0)
Limbe: longueur	long(7.0)	long(7.0)
Limbe: largeur	large(7.0)	large(7.0)
Limbe: Rapport long/larg	a mesurer(1.0)	a mesurer(1.0)

◀ ◻ ◻

Figure 37 : Comparer deux variétés

#### d) Espace Auteur

Les utilisateurs qui ont un type de compte auteur passent après l'authentification à l'espace d'auteur où il trouvera les quatre buttons :

- Liste des variétés (Chercher une variété) : Comme dans le cas d'un compte Utilisateur.
- Ajouter variété : L'auteur peut ajouter une nouvelle variété (à une espèce déjà existant) ou ajouter une espèce dans cet écran-là. La variété ne sera pas ajoutée aux listes des variétés qu'après un correcteur lui accepter .
- Votre profil : Comme le cas d'utilisateur.
- Déconnecter : Revenir à l'interface d'authentification.
- L'icône d'alarme : notifications sur les variétés ajoutées par l'auteur.

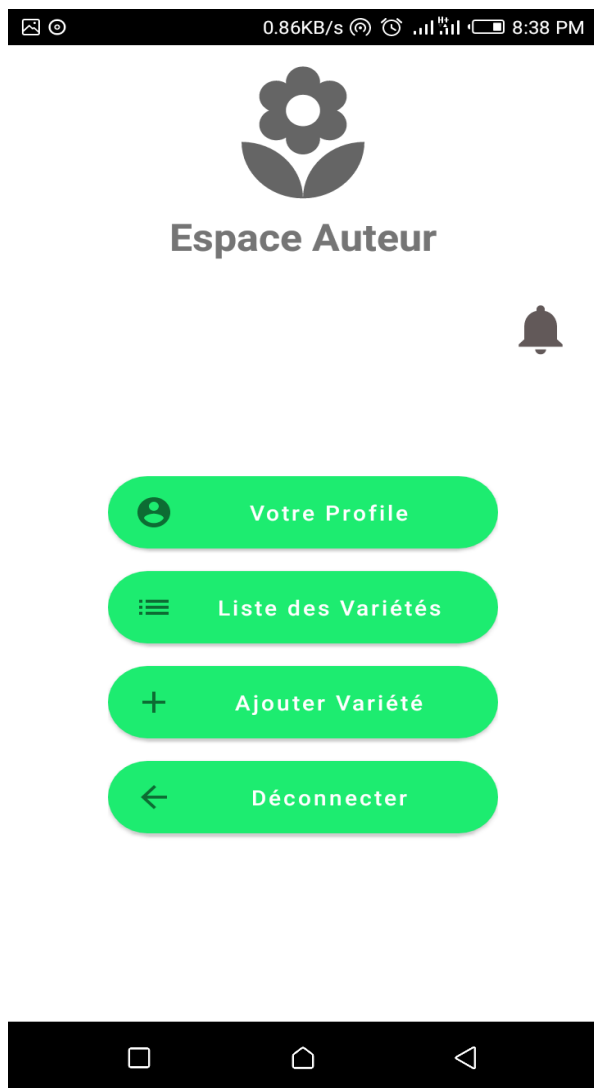


Figure 38 : Espace Auteur



Figure 39 : Notifications

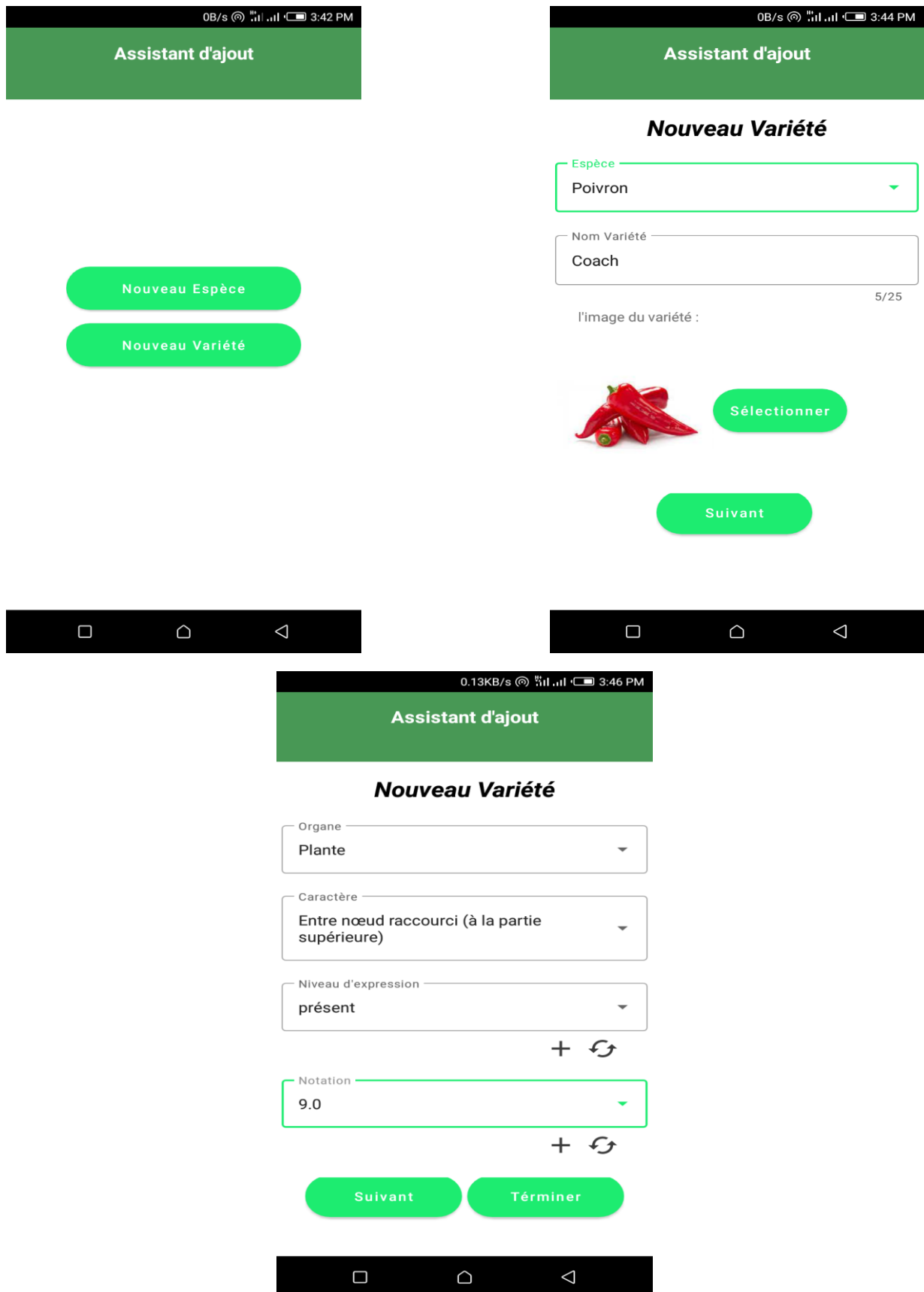


Figure 40 : Ajouter une variété

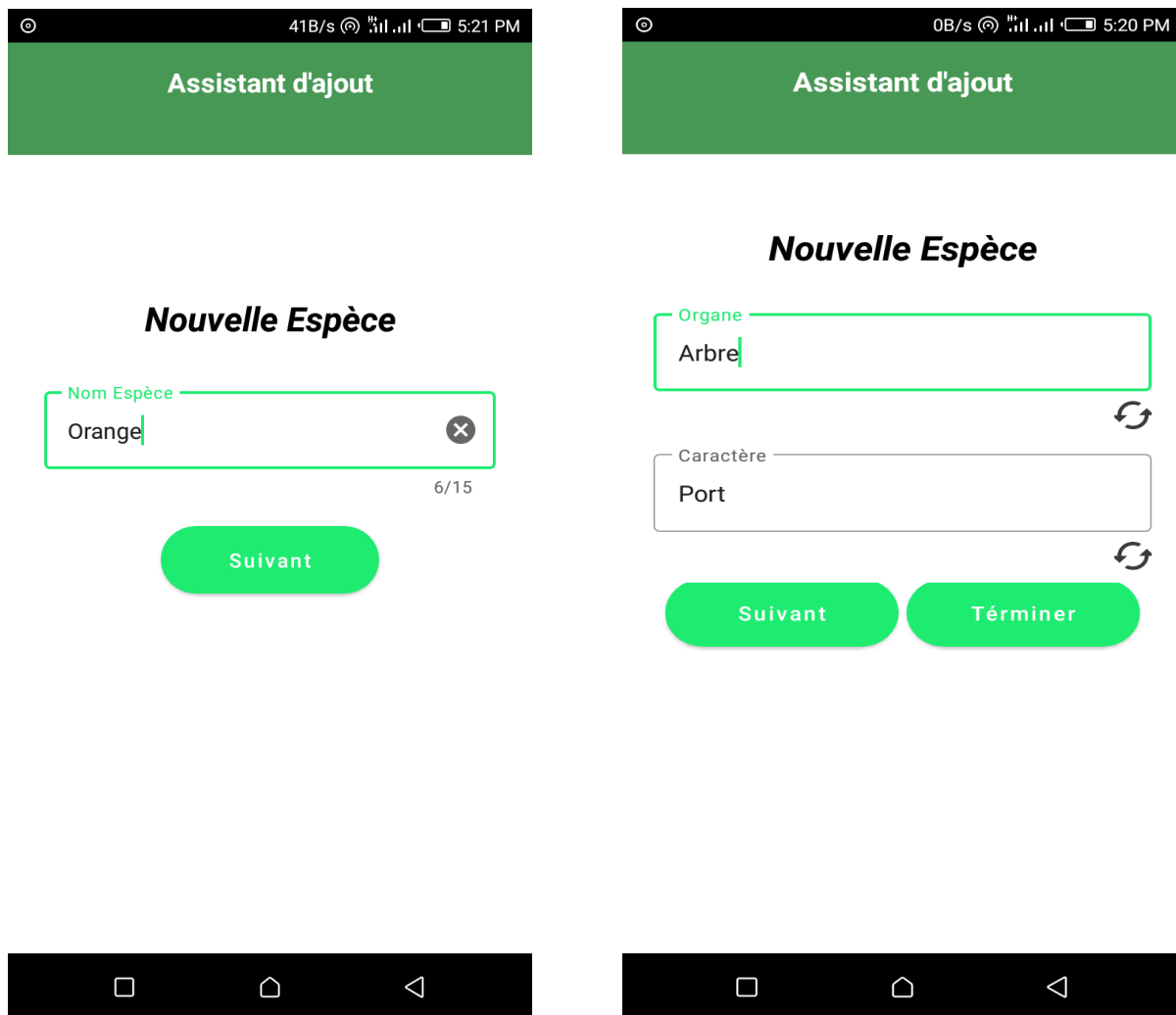


Figure 41 : Ajouter un espèce

#### e) Espace Correcteur :

Les correcteurs sont choisis par les administrateurs pour gestionner les variétés et contrôler l'ajoute des nouvelles variétés et espèces, ils vérifient les données des nouvelles variétés avant de l'accepter ou le refuser. Les correcteurs ont aussi le droit de supprimer ou modifier les variétés existantes.

- Gestion variétés : modifier ou supprimer une variété.
- Nouvelles variétés : Liste des nouvelles variétés, le correcteur l'accepte ou le refuse.

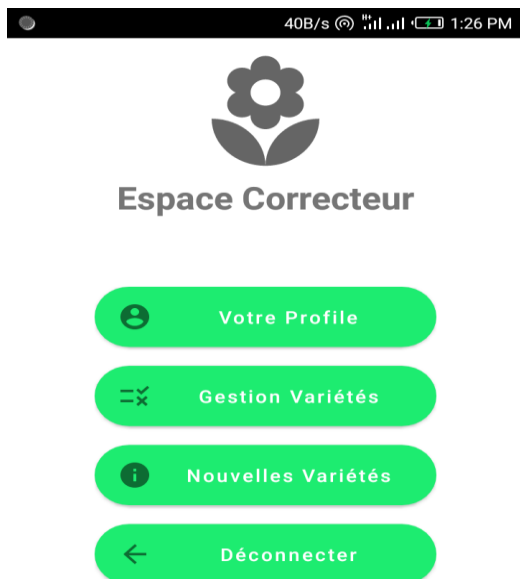


Figure 42 : Espace correcteur

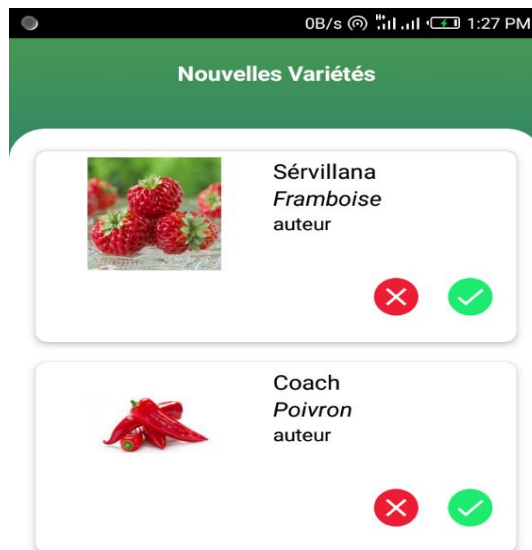


Figure 43 : Nouvelles variétés

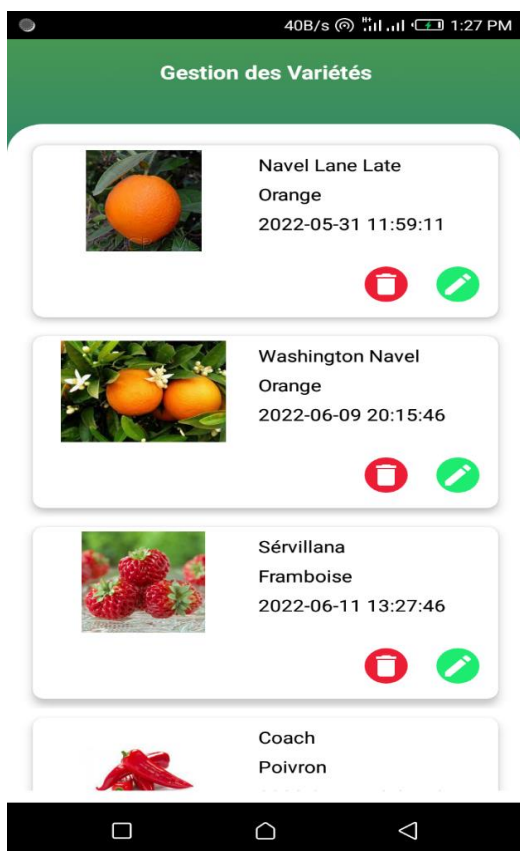
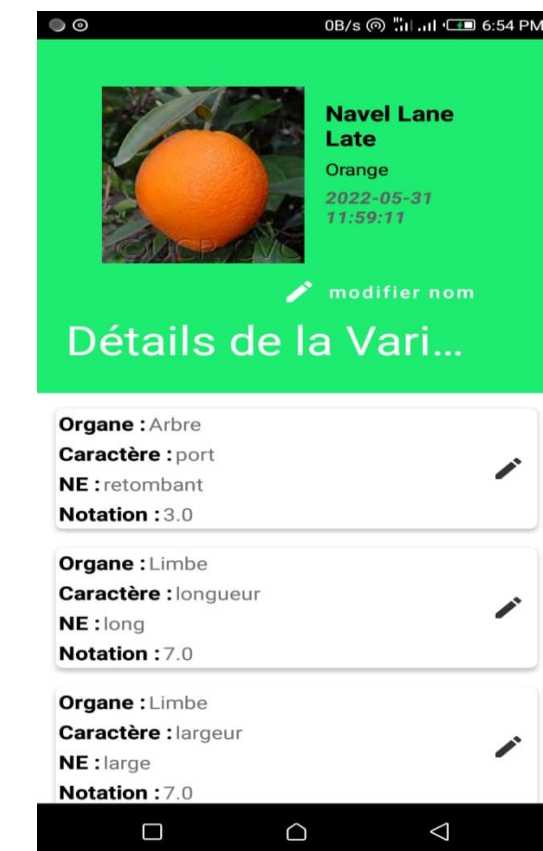


Figure 44 : Gestion des variétés



#### f) Espace Administrateur :

Les administrateurs de l'application sont les responsables de gestion des comptes (supprimer ou modifier le type d'un compte) , ils peuvent accepter ou refuser la demande de devenir un auteur envoyé par un utilisateur . Les correcteurs sont choisis par les administrateurs.

- Gestion utilisateurs : changer le type d'un compte ou le supprimer .
- Liste des demandes : accepter ou refuser les demandes de devenir un auteur .

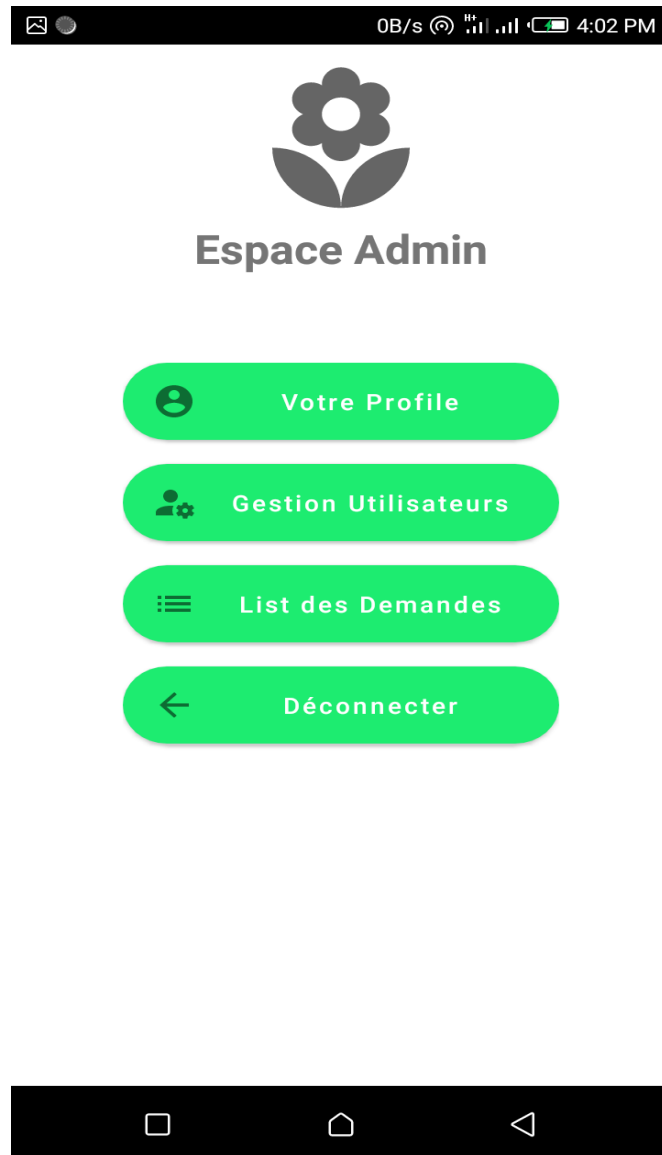


Figure 45 : Espace administrateur

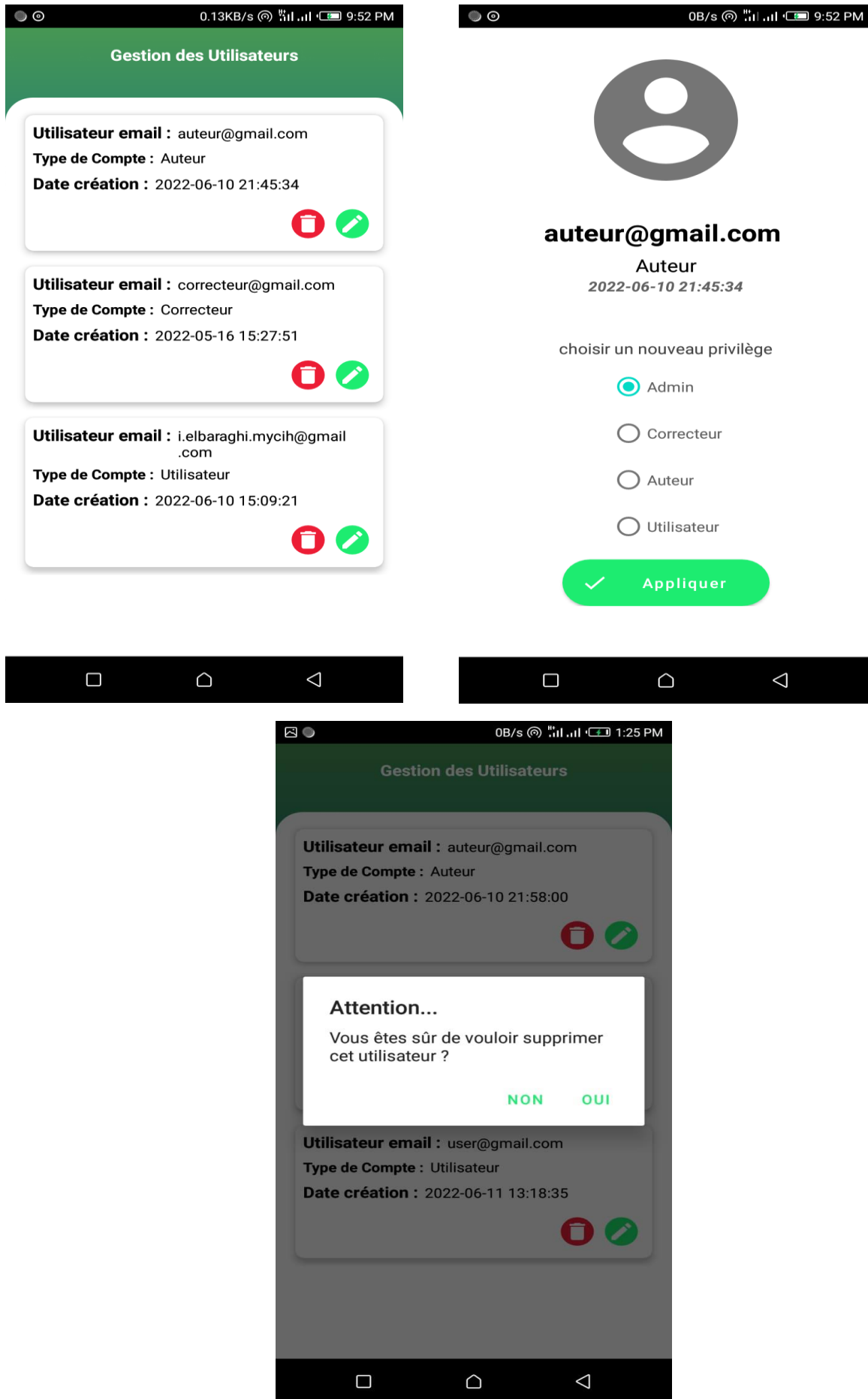


Figure 46 : Gestion des comptes

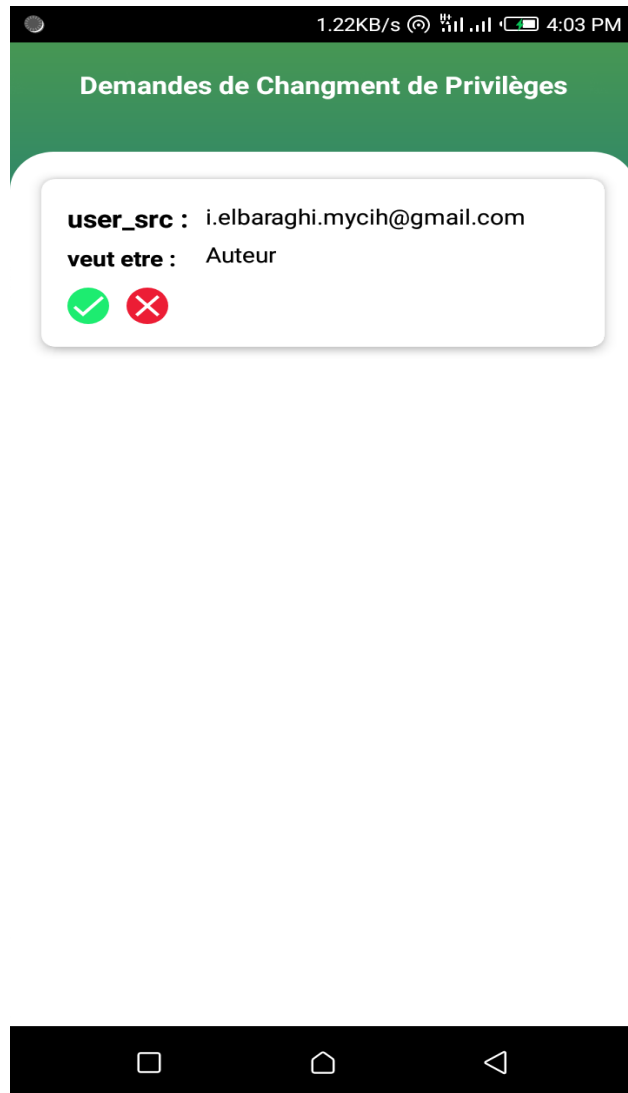


Figure 47 : Demandes de changement de type du compte



## **Conclusion**

Notre projet a été réalisé dans le cadre d'un projet de fin d'études au sein du Département d'Informatique de la Faculté des Sciences d'Agadir. Notre travail se résume en adaptation d'une application android. Ce document décrit d'abord le contexte du travail et les objectifs du projet. Nous avons déterminé les principales fonctions de l'application à générer. Puis nous avons introduit les différentes étapes de la conception, en partant de l'analyse des besoins et de la spécification des besoins, puis nous avons utilisé UML pour modéliser les fonctions identifiées à la fin de cette spécification pour répondre aux besoins des utilisateurs. De plus, nous avons présenté la généralité du développement mobile et les principaux systèmes d'exploitation mobiles, puis décrit les différentes technologies considérées dans notre travail.

Enfin, nous présentons la partie implémentation en citant différentes technologies utilisées pour développer notre application, comme Android Studio comme IDE. Pour notre langage de développement, nous avons choisi JAVA. Ce projet a fait l'objet d'une expérience à la fois intéressante et enrichissante, qui Nous a permis d'améliorer nos connaissances et nos compétences dans le Domaine du développement des application android.

# Bibliographie

## 1. Mémoire :

-Contribution à l'élaboration d'un système informatique pour l'étude de la distinction entre les variétés soumises à l'homologation

## 2. Web :

<https://blog.agendize.com/fr/5-avantages-application-mobile-entreprise>

<http://www.onssa.gov.ma/>

<https://developer.android.com/>

<https://upov.int/portal/index.html.fr>

<https://www.freshvegetables.co.nz/>

<https://dev.java/>

<https://www.vegetables.co.nz/>

<https://www.php.net/>

<https://www.blw.admin.ch/>

<https://www.xml.com/>

<https://www.tomatoesnz.co.nz/>