UNIVERSITE GASTON BERGER DE SAINT-LOUIS

UFR DE SCIENCES APPLIQUEES ET DE TECHNOLOGIE

Centre de formation professionnelle pluridisciplinaire

(cfpp)

LICENCE PROFESSIONNELLE GENIE INFORMATIQUE

(LPGI)

Rapport de Mémoire

SUJET : Conception et développement d’une application mobile pour des étudiants en difficulté

**Présenté et Soutenu par : Youssouf Issa Mahamat**

Sous la direction de : Pr. Jean Marie DEMBELE

# **Dédicace**

**Ce présent travail est dédié à :**

**Mon cher père, Issa Mahamat Denei,**

Tu as été ma lumière dans les moments les plus obscurs, et ton soutien inébranlable a été mon refuge lorsque les défis semblaient insurmontables. Chaque parole de sagesse, chaque geste d’amour et chaque sacrifice que tu as consentis ont laissé une empreinte indélébile dans mon cœur. Tu as été bien plus qu’un père : un guide, un protecteur et une source d’inspiration infinie. Sans toi, je ne serais pas là où je suis aujourd’hui. Merci d’avoir toujours cru en moi, même lorsque je doutais de mes propres forces.

**Ma chère mère Fatimé Mahamat Nour,**

Ton amour a été comme un soleil qui réchauffe même les jours les plus froids. Tes mots tendres, tes câlins réconfortants et ta douceur ont été un baume pour mon âme dans les moments de doute et de fatigue. Chacun de tes sacrifices, souvent silencieux, a été un pas de plus vers ma réussite. Et tes prières… elles ont été comme un bouclier invisible, me protégeant et me portant lorsque les épreuves semblaient trop lourdes. Tu es mon ange, ma force tranquille, et je te suis éternellement reconnaissant pour tout ce que tu as fait pour moi.

**Mes frères, Hamid Issa Mahamat et Al-khassim Issa Mahamat,**

Vous avez été bien plus que des frères : vous avez été mes alliés, mes complices et mes piliers. Votre soutien inconditionnel, votre présence rassurante et votre fraternité ont été des cadeaux précieux dans mon parcours. Chaque moment partagé, chaque rire et chaque encouragement ont renforcé notre lien et m’ont donné la force de continuer. Vous êtes ma fierté, ma famille, et je vous remercie du fond du cœur pour tout ce que vous avez fait pour moi.  
Mon Oncle Ali Souleymane Dabye ; Mon Oncle Tahir Hissein Nour ; Mon Oncle Ali Mahamat Nour ; Mon Oncle Youssouf Mahamat Nour ; Mon Oncle Issa Mahamat Nour; Mon Oncle Senoussi Moussa Abouna ; Mon Oncle Brahim Moussa Abouna ; Mon Grand frère Mahamat Brahim Issakha Kerima (Manakouri) ; Mon Oncle Mahamat Tahir Mahamat Denei ; Mon Oncle Hamit Mahamat Denei ; Mon Oncle Allathene Mahamat Abdramane ; Mon Oncle Youssouf Mahamat Wachil ; Mon Oncle Zakaria Mahamat Abbassi ; Mon Oncle Abakar Wachil ; Mon Oncle Djida Moussa Abouna.

# **Remerciements**

Je remercie avant tout Dieu le Tout-Puissant de m’avoir donné le courage, la patience, la volonté et la détermination nécessaires pour mener à bien ce travail. Sans Sa grâce, rien de tout cela n’aurait été possible.

Ma profonde gratitude va également à l’endroit des autorités académiques ainsi qu’au corps enseignant de l’Université Gaston Berger de Saint-Louis, en particulier ceux de l’UFR SAT. Je tiens à exprimer ma reconnaissance envers les professeurs qui ont marqué mon parcours universitaire et qui m’ont permis d’accumuler des connaissances scientifiques tout au long de ces années d’études. Parmi eux, je souhaite nommer :

Professeur Jean Marie DEMBELE ; Professeur Oumar DIOP ; Professeur Cyril Angelier LEPLAN ; Professeur David Célestin FAYE ; Professeur Cheikh Talibouya DIOP ; Professeur Cheikh Ba ; Professeur Babacar DIOUF ; Professeur Papa Alioune FALL ; Professeur Mouhamadou Lamine BALDE ; Professeur Mahamadou Traore ; Professeur Mohamed Ben MAAOUIA.

Vos enseignements, vos conseils et vos orientations ont été d’une richesse inestimable pour moi. Ce temple du savoir que vous incarnez a été un lieu d’apprentissage et d’épanouissement, et je vous en suis infiniment reconnaissant.

Un merci tout particulier va à mon encadreur, M. Jean Marie DEMBELE, qui a bien voulu m’accompagner tout au long de mes recherches. En tant que directeur de mémoire, vous avez fait preuve d’une disponibilité exceptionnelle, de patience et de rigueur. Vos suggestions, vos idées et vos orientations ont grandement contribué à la réussite de ce travail. Vos enseignements, tant en cours magistraux qu’aux travaux pratiques, ont enrichi mon parcours dans le domaine de l’informatique et m’ont permis de développer des compétences essentielles.

J’exprime ma profonde gratitude au Coordonnateur du CFPP, Pr. Oumar DIOP, pour son accompagnement et son engagement, ainsi qu’à Mme Ndéye Aïssatou SY SOW, assistante du Coordonnateur, pour son précieux soutien et sa disponibilité.

Je tiens également à remercier mes anciens, ceux qui m’ont accueilli et orienté dès mon arrivée à l’université. Un grand merci à Mahamat Hamid Ahmed et Lamine Mahamat Ali, qui ont été des guides précieux pour moi. Votre soutien et vos conseils ont été d’une grande aide pour m’adapter à la vie universitaire et pour avancer dans mon parcours.

Je n’oublie pas Dr Francis Mosnan, Ndeye Khady AIDARA, Acherif Ahmat Alifa, Mahamat Lamine Adji qui m’ont beaucoup aidé dans la réalisation de mon projet. Votre collaboration, vos idées et votre soutien ont été déterminants pour la réussite de ce travail. Merci pour votre générosité.

Je remercie également Dr Outman Ali Outman; Dr Ali Mahamat Saleh Cherif ; Dr Abdelaziz Adoum Bohor; Dr Garba Chaibo Abakar ; Dr Hassane Goudja IZADDINE ; Dr Adoum Mahadjir et tous mes chers compatriotes de l’université Gaston Berger de Saint-Louis pour vos prières, vos encouragements et vos conseils tout au long de ce parcours. Votre soutien moral a été une source de motivation et de réconfort pour moi. Je tiens également à remercier mes amies et sœurs : Leyla Issa Mahamat ; Mariam Issa ; Zara Adoum Ibni Adoum ; Balakhissa Abouna ; Mariam Hassan Mahamat; Achta Abakar Mahamat ; Fatime Ali Absakine ; Zeinab Ahmat Rachid ; Wazina Wardougou.

Je tiens à exprimer ma gratitude à ma famille, qui a toujours cru en moi et m’a soutenu dans les moments difficiles. Votre amour, vos sacrifices et vos prières ont été des piliers essentiels pour ma réussite…

Enfin, je ne saurais terminer mes remerciements sans mentionner les membres du jury pour l’honneur qu’ils ont accepté d’examiner mon travail.

# **Résumé de l'Application "Études Smart"**

L'application "Études Smart" est conçue pour aider les étudiants à mieux organiser leurs études en offrant des fonctionnalités de gestion de matières, de chapitres, et de rappels de révisions. Elle vise à améliorer la productivité et l'efficacité des utilisateurs dans leur apprentissage.

Fonctionnalités Principales :

* Ajout de Matières : Les utilisateurs peuvent ajouter des matières d'étude en Important leurs fichiers « les cours » PDF.
* Modifier et Suppression : Les utilisateurs ont la possibilité de modifier ou de supprimer les matières déjà ajoutées, avec des confirmations avant de procéder à ces actions.
* Planification des révisions : les utilisateurs peuvent planifier leurs cours en choisissant les matières qu’il a importées.
* Choisir une matière « chapitre » ;
* Sélectionner une date ;
* Sélectionner une heure ;

Pour planifier les notifications.

* Notifications : L'application peut envoyer des notifications pour rappeler aux utilisateurs de réviser leurs matières « cours » qu’il a programmées, avec la possibilité de planifier ces rappels.
* Interface Utilisateur : L'application possède une interface utilisateur intuitive, avec des formulaires « Importer » pour ajouter des matières et des listes affichant les matières ajoutées, accompagnées d'icônes et de boutons d'action pour une navigation fluide.
* IA : l’intelligence Artificielle juste l’utilisation fait échange avec IA (Intelligence Artificielle). Utilisateur pose des questions et l’IA répond à la question que l’utilisateur a posée.

Études Smart" est une application pratique et efficace pour les étudiants qui cherchent à optimiser leur temps d'étude et à améliorer leur performance académique. Elle combine des fonctionnalités de gestion des matières avec des outils de notification pour soutenir l'apprentissage actif et structuré.

**Table des matières**

[**Dédicace** II](#_Toc192888061)

[**Remerciements** III](#_Toc192888062)

[**Résumé de l'Application "Études Smart"** V](#_Toc192888063)

[**Tables des figures** VIII](#_Toc192888064)

[**Introduction Générale** 1](#_Toc192888065)

[**Chapitre I: Généralités sur les applications mobiles.** 4](#_Toc192888066)

[**Introduction** 4](#_Toc192888067)

[**I.1. Application Mobile** 4](#_Toc192888068)

[**I.2. Systèmes d’exploitation Mobiles** 5](#_Toc192888069)

[**I.2.1. IOS** 5](#_Toc192888070)

[**I.2.2. Android** 6](#_Toc192888071)

[**I.2.3. Nokia** 7](#_Toc192888072)

[**I.2.4. BlackBerry OS** 8](#_Toc192888073)

[**I.2.5. Windows Phone** 8](#_Toc192888074)

[**I.3. Stratégies de développement.** 9](#_Toc192888075)

[**I.3.1. Application Native** 10](#_Toc192888076)

[**I.3.2. Application Web** 11](#_Toc192888077)

[**I.3.3. Application Hybride** 12](#_Toc192888078)

[**I.4. Les statistiques sur le marché mondial des applications mobiles** 13](#_Toc192888079)

[**I.4.1. Croissance des téléchargements globaux** 13](#_Toc192888080)

[**I.4.2. Téléchargements sur Google Play Store et App Store** 14](#_Toc192888081)

[**I.4.3. Tendances par catégories** 14](#_Toc192888082)

[**I.4.4. Impact économique** 14](#_Toc192888083)

[**Chapitre II: Mise en œuvre de la solution** 15](#_Toc192888084)

[**II.1. Environnements de Développement** 15](#_Toc192888085)

[**II.2. Outils et le Langage Utilisées** 15](#_Toc192888086)

[**II.2.1. Android Studio** 15](#_Toc192888087)

[**II.2.2. Visual Studio Code** 16](#_Toc192888088)

[**II.3. Git** 16](#_Toc192888089)

[**II.4. GitHub** 17](#_Toc192888090)

[**II.5. Flutter** 18](#_Toc192888091)

[**II.5.1. Dart** 19](#_Toc192888092)

[**II.5.2. Firebase** 20](#_Toc192888093)

[**II.5.3. Principales fonctionnalités de Firebase :** 20](#_Toc192888094)

[**II.6. Choix des technologies.** 21](#_Toc192888095)

[**II.7. Ressources** 23](#_Toc192888096)

[**Chapitre III : Analyse et conception** 29](#_Toc192888097)

[**III.1. Langage de modélisation** 29](#_Toc192888098)

[**III.1.1. choix du langage de modélisation** 29](#_Toc192888099)

[**III.1.2. Outils de modélisation** 32](#_Toc192888100)

[**III.1.3. Diagrammes UML utilisé** 35](#_Toc192888101)

[**III.1.3.1 Diagramme de cas d’utilisation** 35](#_Toc192888102)

[**III.1.3.3. Diagrammes de séquence** 37](#_Toc192888103)

[**Chapitre IV : Intégration de l’intelligence Artificielle dans l’application Etudes\_Smart** 41](#_Toc192888104)

[**Chapitre V : Présentation des interfaces de l’application** 42](#_Toc192888105)

[**Conclusion Générale et Perspectives** 54](#_Toc192888106)

[**Références-Bibliographique** 1](#_Toc192888107)

# **Tables des figures**

[**Figure 1: image des téléphones Mobile** 5](#_Toc192888108)

[**Figure 2 : image des Tablettes** 5](#_Toc192888109)

[**Figure 3: L'icône représentant le système iOS** 6](#_Toc192888110)

[**Figure 4: L'icône représentant le système Android**. 7](#_Toc192888111)

[**Figure 5: Exemplaire d’un téléphone sous Symbian OS.** 7](#_Toc192888112)

[**Figure 6: Exemplaire d’un téléphone utilisant BlackBerry OS.** 8](#_Toc192888113)

[**Figure 7: L’icône représentant de Windows Phone** 9](#_Toc192888114)

[**Figure 8: les solutions pour le développement Mobile** 10](#_Toc192888115)

[**Figure 9: image de l’application Native** 11](#_Toc192888116)

[**Figure 10: Image de l’application Web** 12](#_Toc192888117)

[**Figure 11: image de l’application Hybride** 13](#_Toc192888118)

[**Figure 12: image android Studio** 16](#_Toc192888119)

[**Figure 13: image de Git** 17](#_Toc192888120)

[**Figure 14: image de GitHub** 18](#_Toc192888121)

[**Figure 15: image de Flutter** 18](#_Toc192888122)

[**Figure 16: image de Dart** 19](#_Toc192888123)

[**Figure 17: image de Firebase** 21](#_Toc192888124)

[**Figure 18 : image de SQLite.** 28](#_Toc192888125)

[**Figure 19 : image de Lucidchart.** 33](#_Toc192888126)

[**Figure 20: image de StarUML** 33](#_Toc192888127)

[**Figure 21: image d’AgroUML** 34](#_Toc192888128)

[**Figure 22 : image de Papyrus** 34](#_Toc192888129)

[**Figure 23: image de BOUML** 34](#_Toc192888130)

[**Figure 24: Diagramme de cas d’utilisation** 35](#_Toc192888131)

[**Figure 25 : Diagramme de classe** 36](#_Toc192888132)

[**Figure 26: Diagramme de séquence (Système de gestion des cours)** 37](#_Toc192888133)

[**Figure 27 : Diagramme de séquence (Système de gestion des planifications)** 38](#_Toc192888134)

[**Figure 28: diagramme de séquence (Système de gestion de l’intelligence Artificielle (IA))** 39](#_Toc192888135)

[**Figure 29 : Diagramme de séquence (Système de gestion des taches)** 40](#_Toc192888136)

[**Figure 30 : présentation la première interface** 44](#_Toc192888137)

[**Figure 31 : page de connexion** 45](#_Toc192888138)

[**Figure 32: page d’inscription.** 46](#_Toc192888139)

[**Figure 33 : page d’accueil** 47](#_Toc192888140)

[**Figure 34 : page des notifications** 48](#_Toc192888141)

[**Figure 35 : page de Gestion des taches.** 49](#_Toc192888142)

[**Figure 36 : page de la planification.** 50](#_Toc192888143)

[**Figure 37 : page de planification de révision.** 51](#_Toc192888144)

[**Figure 38 : page de Lecture.** 52](#_Toc192888145)

[**Figure 39 : page de l’Intelligence Artificiel** 53](#_Toc192888146)

# **Introduction Générale**

Depuis l’introduction des smartphones en 2007, avec la révolution initiée par l’iPhone, les appareils mobiles ont transformé notre manière de communiquer, de travailler et d’apprendre. Aujourd’hui, les smartphones occupent une place centrale dans la vie quotidienne, devenant des outils indispensables pour accéder à l’information, organiser ses tâches, et même poursuivre ses études.

Face à cette évolution, l’éducation ne fait pas exception. Avec des milliards de personnes utilisant des appareils mobiles chaque jour, les applications éducatives ont émergé comme des solutions modernes pour répondre aux besoins d’apprentissage et d’organisation. Toutefois, malgré la multiplication de ces applications, de nombreux étudiants continuent de rencontrer des défis dans la gestion de leurs études et la planification de leurs révisions.

Dans ce contexte, Études Smart se positionne comme une application innovante, pensée pour exploiter pleinement le potentiel des smartphones afin d’accompagner les étudiants dans leur parcours académique. Grâce à des fonctionnalités interactives, elle vise à améliorer l’organisation, la productivité et l’efficacité dans l’apprentissage, tout en s’adaptant aux besoins des utilisateurs modernes.

De plus, le choix de Flutter comme framework de développement garantit une expérience fluide et performante sur les deux principaux systèmes d’exploitation mobiles, Android et iOS, renforçant ainsi l’accessibilité de l’application.

**Contexte et Problématique**

Dans un environnement académique en constante évolution, les étudiants font face à de nombreux défis qui compromettent souvent leur réussite. Ces problèmes, bien qu’universels, sont accentués par le manque d’outils adaptés à leurs besoins spécifiques. Parmi les principales difficultés rencontrées, on peut identifier :

* Manque de support personnalisé

Les enseignants, bien qu’essentiels dans le parcours éducatif, sont souvent confrontés à des classes surchargées et à des responsabilités multiples. Cette surcharge limite leur capacité à fournir un suivi individualisé pour chaque étudiant. En conséquence, les besoins spécifiques de certains élèves, notamment ceux en difficulté, passent souvent inaperçus. Les étudiants se retrouvent livrés à eux-mêmes, sans orientation claire ni stratégie personnalisée pour surmonter leurs insuffisance.

* Difficulté dans la gestion des matières

L’organisation des cours, des chapitres et des séances de révision reste un véritable casse-tête pour de nombreux étudiants. Avec l’accumulation des contenus académiques et des délais serrés, il devient difficile de prioriser les matières ou de répartir le temps efficacement. L’absence d’un système centralisé pour regrouper, planifier et accéder aux contenus complique davantage cette tâche.

* Absence de rappels structurés

L’oubli des révisions est un problème courant, souvent dû à l’absence de rappels pratiques ou planifiés. Dans un monde où les distractions numériques sont omniprésentes, les étudiants ont besoin de systèmes automatisés pour les alerter des échéances importantes et les motiver à respecter leurs calendriers d’étude.

* Évaluation insuffisante des connaissances acquises

Après une séance de révision, il est crucial pour les étudiants de vérifier leur compréhension. Cependant, les outils pour s’autoévaluer sont rares, voire inaccessibles. Cela conduit à une illusion de maîtrise, où les étudiants pensent avoir compris une matière alors qu’ils rencontrent encore des lacunes importantes.

Dans ce contexte, l'émergence des technologies éducatives, notamment les applications mobiles, offre des opportunités considérables pour transformer ces défis en solutions concrètes. Études Smart se distingue par son approche innovante et ses fonctionnalités adaptées aux besoins des étudiants, allant de la gestion centralisée des matières à l'intégration de technologies modernes comme l'intelligence artificielle.

En s'appuyant sur Flutter, un framework performant et multi-plateforme, l'application garantit une expérience utilisateur fluide et accessible. Elle propose une solution complète qui combine planification structurée, rappels automatisés, apportant ainsi une véritable valeur ajoutée à l'apprentissage et à l'organisation des études.

Ce projet ambitionne non seulement de répondre aux problématiques actuelles des étudiants, mais aussi de les accompagner dans leur quête d'autonomie et d'efficacité académique. Études Smart s'inscrit donc comme un outil incontournable dans le paysage des applications éducatives, en contribuant à réinventer l'expérience d'apprentissage à l'ère numérique.

Ce travail est structuré en cinq chapitres, organisés comme suit :

Chapitre 1 : Généralités sur les applications mobiles  
Ce chapitre présente un aperçu des applications mobiles, leur évolution, leurs types et les technologies utilisées dans leur développement.

Chapitre 2 : Mise en œuvre de la solution

Cette section détaille le processus de développement de l’application, en mettant l’accent sur les outils et technologies adoptés.

Chapitre 3 : Analyse et conception

Ce chapitre traite de l’analyse des besoins, de la modélisation du système et de la conception des différentes composantes de l’application.

Chapitre 4 : Intégration de l’intelligence artificielle dans l’application Études Smart

Ici, nous explorons l’intégration de l’IA, ses fonctionnalités, son impact sur l’expérience utilisateur et les techniques mises en place pour son implémentation.

**Chapitre 5 : Présentation des interfaces de l’application**

Ce dernier chapitre décrit en détail les différentes interfaces de l’application, leur conception, leur ergonomie ainsi que leur rôle dans l’expérience utilisateur. Il met également en avant les choix graphiques et fonctionnels visant à optimiser l’accessibilité et l’efficacité de l’application.

# **Chapitre I: Généralités sur les applications mobiles.**

# **Introduction**

Les applications mobiles sont des logiciels conçus pour fonctionner sur des appareils mobiles tels que les smartphones, les tablettes et dans certains cas les montres connectées. Elles permettent aux utilisateurs d'accomplir diverses tâches, allant de la communication à la gestion, au divertissement, à l'apprentissage et bien plus encore. Avec l'évolution rapide de la technologie mobile et la prolifération des smartphones, les applications mobiles sont devenues un élément essentiel de la vie quotidienne. Elles jouent un rôle crucial dans différents domaines tels que l'éducation, la santé, la finance, le commerce, les médias sociaux et la productivité. Dans ce chapitre, nous présenterons en premier lieu les applications Mobiles et les systèmes d’exploitation Mobiles les plus répondus sur le marché. Nous aborderons par la suite les stratégies préconisées pour leur développement : [1]

# **I.1. Application Mobile**

De nos jours, les applications mobiles prennent une place de plus en plus importante dans notre vie quotidien tant les fonctionnalités qu’elles offrent nous facilitent grandement la vie. Dans cette section, nous aborderons les notions d’application mobiles et des systèmes d’exploitation sur lesquels repose leur fonctionnement.

**Définition**

Une application mobile est un logiciel applicatif développé pour installer sur un appareil mobile, généralement ***un téléphone mobile***, un ***téléphone intelligent*** ou ***une tablette numérique***. Les applications mobiles sont des programmes relativement légers, autonomes, utilisés pour des services de l’information, des medias sociaux, des jeux, etc…

Une application peut-être soit installée directement sur l’appareil dès sa construction en usine soit téléchargée depuis un magasin d’application dit « application store» telles que Google Play, l’App Store ou encore le Windows Phone Store. Une partie des applications sont payante tandis que d’autre ont gratuites. Il existe plusieurs systèmes d’exploitation mobiles.

|  |
| --- |
| **Figure 1: image des téléphones Mobile** |



**Figure 2 : image des Tablettes**

# **I.2. Systèmes d’exploitation Mobiles**

Un **système d’exploitation mobile** (ou Mobile Operating System, OS) est un logiciel essentiel qui agit comme un intermédiaire entre le matériel d'un appareil mobile (smartphone, tablette, montre connectée) et les logiciels ou applications installés sur celui-ci. Il remplit plusieurs fonctions fondamentales pour assurer le fonctionnement optimal de l'appareil et offrir une expérience utilisateur fluide.

## **I.2.1. IOS**

IOS est un système d'exploitation mobile développé par **Apple Inc.** exclusivement pour ses appareils, tels que l'iPhone, l'iPad et l'iPod Touch.

**Caractéristiques :**

Interface utilisateur fluide et intuitive.

⮚Écosystème fermé et sécurisé.

⮚App Store offrant des applications vérifiées.

⮚Mises à jour régulières et uniformes.

**Avantage :**

⮚Excellente optimisation pour le matériel Apple.

⮚Niveau élevé de sécurité et confidentialité.

**Inconvénients :**

⮚Moins de personnalisation pour les utilisateurs.

⮚Fonctionne uniquement sur les appareils Apple.



**Figure 3: L'icône représentant le système iOS**

## **I.2.2. Android**

Android est un système d'exploitation open source basé sur le noyau Linux, développé par Google. Il est utilisé par de nombreux fabricants de smartphones (Samsung, Huawei, etc...).

**Caractéristiques :**

Grande personnalisation.

Large choix d'appareils à différents prix.

Google Play Store avec une grande variété d'applications.

**Avantage :**

Accessibilité sur une large gamme d'appareils.

Système flexible pour les développeurs et utilisateurs.

**Inconvénients :**

Fragmentation des versions entre différents appareils.

Moins sécurisé par défaut qu’IOS.



**Figure 4: L'icône représentant le système Android**.

## **I.2.3. Nokia**

Nokia a utilisé plusieurs systèmes d'exploitation mobiles, notamment Symbian OS, avant de basculer vers Windows Phone. Symbian était l'un des premiers OS mobiles dominants avant l'essor d'iOS et Android.

**Caractéristiques :**

Interface simple et robuste.

Destiné aux appareils moins sophistiqués.

**Avantage :**

Consommation énergétique faible.

Large compatibilité avec les anciens téléphones Nokia.

**Inconvénients :**

Désuet face aux systèmes modernes.

Faible compatibilité avec les applications actuelles.

****

**Figure 5: Exemplaire d’un téléphone sous Symbian OS.**

## **I.2.4. BlackBerry OS**

BlackBerry OS est un système d'exploitation mobile développé par BlackBerry Limited pour ses smartphones orientés vers la sécurité et la productivité.

**Caractéristiques :**

Clavier physique intégré pour certains modèles.

Sécurité renforcée, idéal pour les professionnels.

**Avantage:**

Excellente sécurité des données.

Système de messagerie robuste (BlackBerry Messenger).

**Inconvénients :**

Déclin du marché des appareils BlackBerry.

Manques applications modernes.

****

**Figure 6: Exemplaire d’un téléphone utilisant BlackBerry OS.**

## **I.2.5. Windows Phone**

Windows Phone est un système d'exploitation mobile développé par Microsoft, basé sur une interface utilisateur de type "Metro".

**Caractéristiques :**

Intégration avec l'écosystème Microsoft (Office, OneDrive).

Interface basée sur des tuiles dynamiques.

**Avantage:**

Bonne intégration avec les outils professionnels.

Simplicité de l'interface utilisateur.

**Inconvénients :**

Support abandonné depuis 2017.

Faible écosystème d'applications comparé à Android et iOS.



**Figure 7: L’icône représentant de Windows Phone**

Chaque système d'exploitation a joué un rôle clé dans l'évolution des appareils mobiles, en apportant des innovations et des fonctionnalités uniques adaptées à différents besoins. Actuellement, iOS et Android dominent largement le marché mondial.

# **I.3. Stratégies de développement.**

La conception d’applications mobiles peut se faire suivant trois principales stratégies de développement : ***native***, ***web***, et ***hybride***. Chaque stratégie présente des caractéristiques uniques en termes de développement, d’utilisation et de performance. Voici un aperçu des trois approches :



**Figure 8: les solutions pour le développement Mobile**

## **I.3.1. Application Native**

Définition :

Une application native est une application développée spécifiquement pour un système d'exploitation mobile, tel qu'Android ou iOS. Elle utilise les langages de programmation et les outils spécifiques à la plateforme cibl.

Technologies utilisées :

Java, Kotlin, Python, Swift, Objective C, etc…

**Avantage:**

Performance optimale : Les applications natives tirent pleinement parti des capacités du matériel et des logiciels de l'appareil, garantissant des performances élevées.

Expérience utilisateur : Elles offrent une interface utilisateur fluide et bien intégrée, conforme aux normes de la plateforme.

Accès complet aux fonctionnalités de l'appareil : Les applications natives peuvent accéder aux fonctionnalités du téléphone, telles que l'appareil photo, le GPS, et les notifications push, sans restriction.

**Inconvénients :**

Coût élevé : Le développement d'applications distinctes pour plusieurs plateformes nécessite davantage de ressources.

Maintenance complexe : Les mises à jour doivent être effectuées séparément pour chaque plateforme.



**Figure 9: image de l’application Native**

## **I.3.2. Application Web**

Définition :

Une application web est une application accessible via un navigateur Internet. Elle est développée à l'aide des technologies web standard comme HTML, CSS, et JavaScript, et n'est pas installée directement sur l'appareil.

Technologies utilisées :

HTML5, CSS, JavaScript, etc...

**Avantage :**

Portabilité : Une seule application peut être utilisée sur plusieurs plateformes via un navigateur.

Facilité de mise à jour : Les mises à jour sont directement appliquées sur le serveur, sans intervention de l'utilisateur.

Coût réduit : Développement unique, sans besoin de créer des versions spécifiques pour chaque plateforme.

**Inconvénients :**

Performance limitée : Les applications web sont moins performantes que les applications natives, en particulier pour des fonctionnalités exigeantes.

Accès restreint : L'accès aux fonctionnalités natives de l'appareil (appareil photo, GPS, etc.) est limité.

Expérience utilisateur moins immersive : Les interfaces peuvent ne pas être aussi fluides que celles des applications natives.



**Figure 10: Image de l’application Web**

## **I.3.3. Application Hybride**

Définition :

Une application hybride combine les éléments des applications natives et web. Elle est développée en utilisant des technologies web (HTML, CSS, JavaScript) mais fonctionne à l'intérieur d'une enveloppe native, permettant ainsi un accès aux fonctionnalités de l'appareil. Des frameworks comme Flutter, React Native ou Ionic sont souvent utilisés pour ce type de développement.

* Technologies utilisées:

Flutter, React Native ou Ionic, Objective C, Swift, HTML5, etc.…

**Avantage:**

Développement rapide : Une base de code unique peut être utilisée pour plusieurs plateformes.

Coût modéré : Réduction des coûts de développement par rapport aux applications purement natives.

Accès aux fonctionnalités natives : Grâce à des plugins, les applications hybrides peuvent accéder à des fonctionnalités matérielles comme une application native.

**Inconvénients :**

Performance intermédiaire : Elles ne sont pas aussi rapides que les applications natives.

Limites de compatibilité : Certaines fonctionnalités avancées de l’appareil peuvent être difficiles à intégrer.

Dépendance aux frameworks : La performance et les capacités dépendent du framework choisi, ce qui peut limiter l'évolutivité.



**Figure 11: image de l’application Hybride**

# **I.4. Les statistiques sur le marché mondial des applications mobiles**

Le marché mondial des applications mobiles continue de croître de manière exponentielle, soulignant l'importance croissante des technologies mobiles dans la vie quotidienne des utilisateurs. Les statistiques récentes montrent une augmentation significative du nombre de téléchargements d'applications pour les différentes plateformes chaque année.

## **I.4.1. Croissance des téléchargements globaux**

En 2017, le nombre total de téléchargements d'applications mobiles a atteint 178 milliards.

Ce chiffre a augmenté pour atteindre 205 milliards en 2018.

En 2023, les téléchargements ont franchi la barre des 255 milliards et devraient encore croître pour atteindre environ 290 milliards en 2024, soit une augmentation de près de 63 % en sept ans (2017-2024).

## **I.4.2. Téléchargements sur Google Play Store et App Store**

Les deux plateformes majeures, Google Play Store (Android) et App Store (iOS), dominent le marché.

En 2019, les téléchargements combinés issus de ces deux plateformes ont atteint plus de 120 milliards, un record, sans inclure les réinstallations et les mises à jour.

En 2024, ce chiffre devrait dépasser les 140 milliards, renforçant leur position dominante sur le marché des applications mobiles.

## **I.4.3. Tendances par catégories**

Les applications les plus téléchargées incluent :

Applications de réseaux sociaux : WhatsApp, Facebook, Instagram, et TikTok figurent parmi les leaders.

Applications de divertissement : YouTube, Netflix, et Spotify continuent de connaître une popularité croissante.

Applications éducatives et de productivité : Avec l'essor de l'apprentissage à distance et du télétravail, les applications comme Duolingo, Zoom, et Microsoft Teams connaissent une adoption massive.

## **I.4.4. Impact économique**

Le marché des applications mobiles génère également des revenus significatifs, avec une croissance constante :

Les dépenses des utilisateurs dans les applications ont dépassé 200 milliards USD en 2023 et devraient atteindre 240 milliards USD en 2024.

Les revenus publicitaires des applications mobiles représentent une part croissante, contribuant à l'essor des modèles gratuits avec publicité intégrée.

# **Chapitre II: Mise en œuvre de la solution**

Dans ce chapitre, nous allons présenter les éléments essentiels qui ont contribué à la mise en œuvre de notre solution. Nous commençons par décrire l'environnement de travail dans lequel le projet a été développé, en mettant en avant les langages de programmation, les outils et les technologies utilisés. Ensuite, nous détaillerons les choix faits pour garantir une mise en œuvre efficace et adaptée aux besoins du projet.

# **II.1. Environnements de Développement**

Un environnement de développement, également connu sous le nom d’Integrated Development Environment (IDE), est une suite d'outils logiciels conçus pour assister les développeurs dans la création, l'écriture, le test et le débogage de leur code. Ces environnements offrent un espace de travail intégré, optimisé ainsi que le cycle de développement des applications.

# **II.2. Outils et le Langage Utilisées**

## **II.2.1. 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. Il peut être téléchargé sous les systèmes d’exploitation Windows, MacOS, Chrome OS et Linux. [2]

**Avantages**

Android Studio se veut léger et personnalisable. Il a en plus l’avantage de tourner sur l’ensemble des principaux systèmes d’exploitation (OS, Windows et bien sur Linux). Distribué sous licence Apache 2.0 il est, comme a l’habitude de Google, libre et open source.

Pour le développement mobile, Android Studio est une boite à outils complète. Il embarque ainsi un éditeur de texte, des outils de débogage ainsi qu’un émulateur. Tout cela, réuni en un même endroit, facilite l’écriture du code. De nombreux commentaires aident en plus le développeur à l’utiliser les bons éléments aux bons endroits. Un code d’erreur apparait enfin immédiatement si une fonction n’est pas utilisée de la façon attendue, limitant les pertes de temps inutiles.

Système d’exploitation : Windows, Linux, MacOs

Développeur : Google, JetBrains

Site Web: <https://developer.android.com/studio>



**Figure 12: image android Studio**

## **II.2.2. Visual Studio Code**

Visual Studio Code est un éditeur de code en termes simples. Visual Studio Code est « un éditeur gratuit qui aide le programmeur à écrire du code, aide au débogage et corrige le code à l’aide de la méthode intelli-sense ». En termes normaux, cela permet aux utilisateurs d’écrire le code de manière simple. Beaucoup de gens disent qu’il s’agit de la moitié d’un IDE et d’un éditeur, mais la décision appartient aux codeurs. Tout programme/logiciel que nous voyons ou utilisons fonctionne sur le code qui s’exécute en arrière-plan. Traditionnellement, le codage était utilisé dans les éditeurs traditionnels ou même dans éditeur de base comme le bloc-notes ! Ces éditeurs fournissaient un support de base aux codeurs. [3]

|  |  |
| --- | --- |
| Figure : image de Visual Studio code | Système d’exploitation : Windows, Linux, MacOs  Développeur : Microsoft  Site Web: <https://code.visualstudio.com> |

# **II.3. Git**

**Git e**st un système de contrôle de version distribuée. Il s'agit d'un logiciel permettant de suivre les modifications effectuées sur un ensemble de fichiers au fil du temps et de gérer les différentes versions de ces fichiers. Créé en 2005 par **Linus Torvalds** pour faciliter le développement du noyau Linux, Git est aujourd'hui l'un des outils de contrôle de version les plus populaires dans l'industrie du développement logiciel.

Git est un outil incontournable pour les développeurs modernes. Sa capacité à gérer efficacement les versions de code, à favoriser la collaboration et à s'adapter à divers flux de travail en fait un choix privilégié pour les projets individuels et collaboratifs. [4]

Développeur : Software Freedom Conservancy

Système d’exploitation : Multiplateforme

Type : Gestionnaire de code source décentralisé

Site : <https://git-scm.com/>



**Figure 13: image de Git**

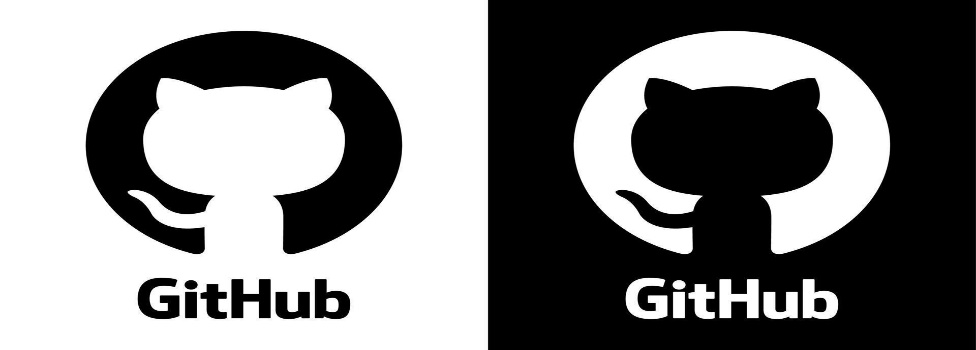
# **II.4. GitHub**

GitHub est une plateforme web d’hébergement et de gestion de code source pour les projets de logiciel. Elle utilise Git, un système de contrôle de version, pour permettre aux utilisateurs de collaborer à des projets logiciels en ligne. Les développeurs peuvent utiliser Github pour héberger leur code, gérer les taches et les bogues, ainsi que pour discuter avec d’autres développeurs et mainteneurs de projet. GitHub offre également des fonctionnalités pour la documentation de projet, le suivi des problèmes, le déploiement d’applications, et bien plus encore. De plus, GitHub est devenu une importante communauté de développeurs et un endroit où les entreprises peuvent trouver des talents et embaucher des développeurs [5]

Propriétaire : Microsoft

Type : Gestion de version collaboration

Site : <https://github.com>



**Figure 14: image de GitHub**

# **II.5. Flutter**

Flutter est un kit de développement de logiciel (SDK) d'interface utilisateur open-source créé par Google il est utilisé pour développer des applications multiplateformes pour Android, iOS, Linux, Mac, Windows, web… à partir d'une seule base de code. Dont la première version a été publiée sous forme de projet open source à la fin de l’année 2018. Flutter met à disposition une grande variété de bibliothèques d’éléments d’Interface Utilisateur standard pour Android et iOS. Et permet aussi de recharger la page automatiquement sans faire l’exécution du codes ce rechargement simplifie le débugge.

Les applications développées avec Flutter prennent l’aspect d’applications typiques des systèmes correspondants et se comportent également de manière similaire sans que le programmeur c’est-à-dire vous n’ait besoin de prêter attention à ces caractéristiques.

Propriétaire : Google

Type : **framework d'interface utilisateur (UI)** open-source

Site : <https://flutter.dev/>



**Figure 15: image de Flutter**

## **II.5.1. Dart**

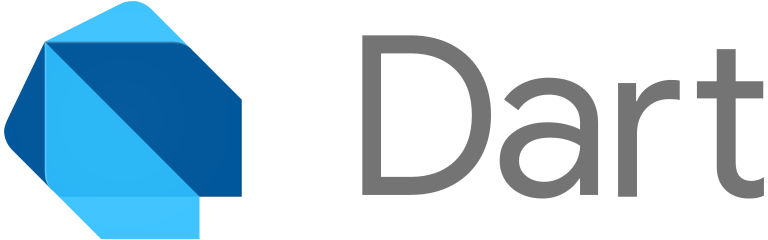
**Dart** est un langage de programmation open-source orienté objet, créé par **Google**, conçu principalement pour le développement d'applications côté client. Ce langage est utilisé pour développer des applications multiplateformes, notamment pour le web, les applications mobiles et les applications de bureau. La première version de Dart a été publiée en 2011. Depuis, il a évolué pour devenir un élément clé de l'écosystème de **Flutter**, offrant des performances et une simplicité optimales.

Dart a mis à disposition une syntaxe moderne et concise, proche des langages populaires comme Java, ce qui le rend accessible aux développeurs. Il prend en charge la compilation anticipée (Ahead-Of-Time) et la compilation Just-In-Time, offrant des performances élevées pour les applications mobiles et web. En outre, Dart facilite le rechargement à chaud (Hot Reload), ce qui permet aux développeurs de visualiser instantanément les modifications apportées au code sans redémarrer l'application, simplifiant ainsi le débogage et améliorant la productivité.

Les applications développées en Dart, notamment via Flutter, prennent un aspect natif sur les différentes plateformes et se comportent comme des applications typiques des systèmes correspondants. Cela permet aux développeurs de ne pas se soucier des spécificités des plateformes tout en garantissant une expérience utilisateur fluide et cohérente.

Propriétaire : Google

Site : <https://dart.dev>



**Figure 16: image de Dart**

## **II.5.2. Firebase**

Firebase est une plateforme de développement d'applications proposée par Google, conçue pour aider les développeurs à créer, améliorer et gérer des applications web et mobiles de manière plus efficace. Lancée initialement en 2011 en tant que solution indépendante, elle a été acquise par Google en 2014 et intégrée à son écosystème pour offrir un éventail d'outils et de services.

Firebase se distingue par sa capacité à fournir une infrastructure complète et prête à l'emploi, permettant de se concentrer sur les fonctionnalités principales de l'application sans devoir gérer des serveurs ou une infrastructure complexe.

## **II.5.3. Principales fonctionnalités de Firebase :**

* **Base de données en temps réel:**

Fournit une base de données NoSQL qui synchronise les données entre les utilisateurs en temps réel.

Idéal pour les applications interactives comme les messageries ou les jeux.

* **Cloud Firestore :**

Une base de données flexible et évolutive conçue pour les grandes applications. Prend en charge les requêtes complexes et les fonctionnalités hors ligne.

* **Authentification :**

Permet de gérer facilement les utilisateurs avec des options de connexion via e-mail, réseaux sociaux (Google, Facebook, etc.), ou téléphone.

* **Hébergement :**

Offre un hébergement rapide et sécurisé pour les sites web et applications web progressives (PWA).

* **Stockage en nuage :**

Solution de stockage pour gérer et partager des fichiers comme des images, vidéos, ou documents.

* **Fonctions cloud:**

Permet d'exécuter du code backend sans gérer de serveurs.

Utilisé pour les déclencheurs en réponse à des événements spécifiques (par exemple, une nouvelle base de données ou un téléchargement de fichier).

* **Notifications :**

Permet l'envoi de notifications push personnalisées aux utilisateurs via Firebase Cloud Messaging (FCM).

* **Analytique :**

Intégré des outils pour suivre et analyser le comportement des utilisateurs, optimisant ainsi les performances et l'engagement.

Site : <https://firebase.google.com>

Propriétaire : Google

Type : Plateforme de développement d'applications



**Figure 17: image de Firebase**

# **II.6. Choix des technologies.**

La réussite d’un projet de développement d’application repose en grande partie sur le choix des technologies. Pour l’application **Études\_Smart**, plusieurs critères ont guidé notre sélection : la simplicité de développement, la performance, la compatibilité multiplateforme, et la facilité de maintenance. Après une analyse approfondie, nous avons opté pour **Flutter** comme technologie principale pour le développement de l’application.

* **Présentation de Flutter**

Flutter est un **framework open-source** développé par Google, utilisé pour créer des applications multiplateformes à partir d’une base de code unique. Il permet de développer des applications natives performantes pour Android, iOS, Web, Windows, Mac, et Linux.

Les principales raisons du choix de Flutter sont les suivantes :

**Un développement rapide** : Grâce à la fonctionnalité de rechargement à chaud (Hot Reload), les modifications apportées au code sont instantanément visibles, ce qui accélère considérablement le processus de développement.

**Une base de code unique** : Flutter offre la possibilité de développer des applications pour plusieurs plateformes sans avoir à écrire un code séparé pour chacune d’elles, réduisant ainsi les efforts et le temps de développement.

**Des interfaces utilisateur riches** : Le framework propose une vaste bibliothèque de widgets personnalisables et conformes aux standards des plateformes (Material Design pour Android et Cupertino pour iOS), garantissant une expérience utilisateur cohérente.

**Performances élevées** : Grâce à son moteur de rendu en C++, Flutter permet de produire des applications rapides et fluides, proches des performances natives.

**Communauté active et documentation riche** : La large communauté de développeurs et la documentation exhaustive facilitent l’apprentissage et la résolution de problèmes.

**Compatibilité multi-plateforme**

L’application Etudes\_Smart vise à être accessible à un large public, quelle que soit la plateforme utilisée (Android, IOS, web). Flutter, avec son approche multiplateforme, nous permet de répondre à cet objectif sans multiplier les efforts de développement.

**Facilité d’intégration avec d’autres technologies**

Flutter offre une compatibilité transparente avec des bases de données locales comme SQLite (utilisée dans ce projet) ainsi que des services externes comme Firebase pour les notifications ou le stockage cloud. Cette flexibilité technologique nous a permis de construire une application robuste.

**Personnalisation de l’interface utilisateur**

Grâce aux widgets de Flutter, nous avons pu concevoir une interface utilisateur ergonomique et attrayante, adaptée aux besoins des étudiants. Par exemple, des widgets tels que **ListView (**affiche des listes déroulantes avec une grande quantité de données) et **Card (**ajoute un style visuel attrayant à vos éléments, parfait pour les informations structurées) Ont été utilisés pour afficher des listes de matières et des informations de manière claire et structurée.

**Gestion des performances hors ligne**

L’intégration de bases de données locales (comme SQLite) et d’autres outils natifs permet à l’application d’offrir une expérience fluide même en l’absence de connexion Internet, un critère essentiel pour les étudiants en déplacement ou dans des zones à connectivité limitée.

Le choix de Flutter comme technologie principale pour **Études\_Smart** s’est avéré pertinent et stratégique. Ses nombreux avantages en termes de rapidité, de compatibilité, et de performances nous ont permis de développer une application moderne et performante, adaptée aux besoins des étudiants. Grâce à Flutter, nous avons pu relever les défis du développement multiplateforme tout en assurant une expérience utilisateur de haute qualité.

**Tableaux de comparaison avec les autres technologies :**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Critères** | **Flutter** | **React Native** | **Swift (iOS)** | **Kotlin (Android)** |
| Développement multiplateforme | Android, IOS, Web, Desktop | Android, IOS, Web | Uniquement (IOS), | Uniquement (Android) |
| Langage Utilisé | Dart | javaScript (avec React) | Swift | kotlin |
| Intégration Base de données | SQLite, Firebase, … | SQLite, Firebase, … | Core Data, SQLite | Room Database, SQLite |

Ce tableau compare différentes technologies pour le développement d'une application mobile. Il met en évidence leur compatibilité multiplateforme, les langages utilisés et les options d'intégration des bases de données.

# **II.7. Ressources**

**Commandes Flutter essentielles :**

**1. Commandes de base :**

⮚flutter doctor :

Vérifie l’installation de flutter et détecte les éventuels problèmes de configuration.

⮚flutter create nom\_du\_projet

Cree un nouveau projet Flutter.

⮚flutter run

Exécute l’application Flutter sur l’appareil connecté ou l’émulateur actif.

⮚flutter devices

Liste tous les appareils connectés et disponibles pour le déploiement.

⮚flutter clean

Supprime les fichiers générés par Flutter pour repartir sur une base propre.

**2. Gestion des packages**

⮚flutter pub get

Télécharge les packages définis dans le fichier pubspec.yaml.

⮚flutter pub upgrade

Met à jour les packages vers leurs dernières version compatibles.

⮚flutter pub outdated

Vérifie si des packages utilisés ont des mis à jour disponibles.

⮚flutter pub add [nom\_du\_package]

Ajoute un nouveau package au fichier pubspec.yaml et l’installe.

⮚flutter pub remove [nom\_du\_package]

Supprime un package du fichier pubspec.yaml

**3. Installation**

Mise en place environnement Flutter

Voici les étapes pour mettre en place un environnement Flutter (sur Windows).

**⮚Télécharger et installer Flutter**

**Télécharger Flutter**

Rendez-vous sur le site officiel : [Flutter SDK](https://docs.flutter.dev/get-started/install/windows/mobile#download-then-install-flutter).

Téléchargez le fichier ZIP pour Windows.

**Extraire le SDK Flutter**

Extrayez le contenu du fichier ZIP dans un dossier (C:\src\flutter).

**Ajouter Flutter au PATH**

Accédez à "Paramètres système avancés" → Onglet "Avancé" → "Variables d'environnement".

Dans les variables utilisateur ou système, ajoutez le chemin du dossier bin de Flutter(C:\src\flutter\bin) au **PATH**.

**Installer les outils nécessaires**

**Installer Git**

Téléchargez et installez [Git](https://git-scm.com/downloads/guis?os=windows).

Exécuter ces deux commande pour configurer votre Git et le synchroniser avec votre compte GitHub ou GitLab : git config –global user.email"votre\_email@exemple.com" & git config --global user.name "nomutilisateur"

**Installer VS Code**

Téléchargez et installez [Visual Studio Code](https://code.visualstudio.com/).

Installez les extensions Flutter et Dart dans Visual Studio Code si vous l’utilisez

⮚ Configurer Android Studio

**Installer Android Studio**

Téléchargez et installez [Android Studio](https://developer.android.com/studio).

**Installer le SDK Android**

Ouvrez Android Studio, allez dans "More Actions" → "SDK Manager".

Si ce n'est le cas, installez **SDK Android**, **Android SDK command-line Tools**, **Android Emulator**, **Intel x86 Emulator Accelerator (HAXM installer)** et les outils requis (versions recommandées dans la documentation Flutter).

**Configurer un émulateur**

Dans Android Studio, allez dans "More Actions" → "Virtual Device Manager" → "Create Device".

Configurez un émulateur avec une image système (par exemple, Pixel 4 avec Android 12).

**Accepter la License d'Android**

Avec la commande flutter doctor --android-licences accepter la license d'Android dans votre installation.

Vérifier l’installation avec Flutter Doctor

Ouvrez un terminal (cmd ou PowerShell) et exécutez :

flutter doctor

Cette commande vérifie votre environnement et liste les éléments manquants.

Si des éléments sont signalés comme manquants, suivez les recommandations affichées.

Testez votre installation

Créez un nouveau projet Flutter :

flutter create mon\_projet

Accédez au répertoire du projet :

cd mon\_projet

Exécutez le projet sur un émulateur ou un appareil connecté :

flutter run

Conseils supplémentaires

**Activer le mode développeur sur un appareil Android**

Connectez un appareil Android à votre PC.

Activez le mode développeur et débogage USB dans les paramètres de l’appareil.

**Mise à jour régulière de Flutter**

Assurez-vous d'avoir la dernière version en exécutant :

flutter upgrade

**Problèmes de PATH ?**

Vérifiez si la commande flutter fonctionne dans le terminal. Si ce n’est pas le cas, assurez-vous que C:\flutter\bin est bien ajouté au **PATH**.

**Sqlite :**

Dans le cadre de ce projet, la gestion des données joue un rôle crucial, notamment pour organiser et stocker les informations liées aux matières, aux révisions, et aux fichiers associés. Pour répondre à ces besoins, nous avons opté pour **SQLite**, un système de gestion de bases de données relationnelles**(SGBDR)** léger et efficace, adapté aux applications mobiles.

**Présentation de SQLite**

SQLite est un **Système de Gestion de Bases de Données Relationnelles (SGBDR)** open-source, qui fonctionne directement à partir d'un fichier local. Contrairement à d'autres SGBDR comme MySQL ou PostgreSQL, SQLite est **sans serveur** et ne nécessite aucune configuration préalable.

Les principales caractéristiques de SQLite sont :

**Léger** : Une taille réduite, généralement inférieure à 1 Mo.

**Autonome** : Aucune dépendance externe, il fonctionne directement avec des fichiers locaux.

**Standard SQL** : Compatible avec la plupart des fonctionnalités du langage SQL.

**Portabilité** : La base de données est stockée dans un fichier unique, facilement transférable entre différents systèmes.

**Performances élevées** : Optimisées pour des opérations fréquentes de lecture/écriture dans des environnements embarqués.

**Pourquoi choisir SQLite ?**

Pour ce projet, plusieurs critères ont motivé le choix de SQLite :

**Facilité d'intégration avec Flutter** :   
SQLite dispose de bibliothèques Flutter bien documentées, telles que **sqflite**, qui simplifient son utilisation. Cela a permis d'intégrer la base de données directement dans l'application tout en assurant une gestion robuste des données.

**Léger et efficace** :   
Grâce à son architecture légère, SQLite n'alourdit pas l'application, un avantage essentiel pour les environnements mobiles où les ressources matérielles sont limitées.

**Autonome et sans serveur** :   
Contrairement aux solutions nécessitant des serveurs externes (comme Firebase Realtime Database ou MySQL), SQLite fonctionne en local, éliminant ainsi la nécessité d'une connexion Internet constante et notamment les coûts d'infrastructure.

**Portabilité** :   
Les bases de données SQLite étant des fichiers autonomes, elles peuvent être facilement sauvegardées, restaurées ou transférées, offrant une grande flexibilité dans la gestion des données.

En intégrant SQLite dans ce projet, nous avons pu bénéficier d'un système de gestion de bases de données performant, léger et adapté aux besoins des applications mobiles. Son utilisation a permis de structurer efficacement les données nécessaires au bon fonctionnement de l'application, tout en offrant une expérience utilisateur fluide et sans dépendances supplémentaires.



**Figure 18 : image de SQLite.**

**Tableaux de comparaison avec les autres technologies :**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Critère | SQLite | Firebase Realtime Database | MySQL | |  | | --- | |  |  |  | | --- | | PostgreSQL | |
| Type | |  | | --- | |  |  |  | | --- | | SGBDR embarqué | | Base de données NoSQL Cloud | SGBDR (serveur) | SGBDR (serveur) |
| Mode de fonctionnement | Local (sans serveur) | Cloud (nécessite internet) | Serveur nécessaire | Serveur nécessaire |
| Langage SQL | |  | | --- | |  |  |  | | --- | | Oui | | Non | Oui | Oui |
| Performance | Rapide en local | Rapide en cloud mais dépend du réseau | Bonne mais nécessite un serveur | Excellente mais nécessite un serveur |
| Sécurité | Bonne (Local) | Bonne (auth Firebase | Bonne (nécessite une gestion coté serveur) | Très bonne |

Ce tableau compare différentes technologies de bases de données en fonction de leurs caractéristiques. Il met en avant leur type, leur mode de fonctionnement, la compatibilité avec SQL, ainsi que leurs performances et leur niveau de sécurité.

# **Chapitre III : Analyse et conception**

**Introduction**

Ce chapitre est consacré à l'analyse et à la conception de notre application Études Smart. Nous y présentons les choix technologiques et méthodologiques pour la modélisation, ainsi que les outils utilisés pour concevoir une solution performante, robuste et maintenable. Le langage de modélisation adopté est détaillé, accompagné des diagrammes qui seront utilisés pour visualiser les aspects statiques (structurel) et dynamiques (comportemental) du système.

# **III.1. Langage de modélisation**

## **III.1.1. choix du langage de modélisation**

Dans le cadre de ce projet, plusieurs langages de modélisation informatique ont été étudiés pour choisir celui qui s’adapte le mieux à nos besoins. Parmi ces langues figurent : [8]

* UML (Unified Modeling language) : Un langage standard pour la modélisation orientée objet.
* BPMN (Busness Process Model and Notation) : Idéal pour modéliser des processus métier.
* ArchiMate : Un langage de modélisation d’architecture d’entreprise utilisé pour la description, l’analyse et la visualisation des systèmes complexes.

Apres une étude comparative, notre choix se porte sur le langage UML qui est un langage de modélisation unis qui permet de modéliser une application de façon standard dans le cadre de conception orienté objet. Il est complexe, il est à la fois une norme, un langage de modélisation objet, un support de communication et un cadre méthodologique. Il fournit un ensemble de notations visuelles pour représenter les différents perspectives d’un système, que ce soit sa structure, son représenter les différentes les aspects spécifiques du système.

Un **diagramme UML** est une représentation graphique d'un aspect particulier du système. UML propose plusieurs types de diagrammes regroupés en deux grandes catégories qui sont :

La Vues Statique et Vues Dynamique

**⮚Vues statiques** :

Les vues statiques se concentrent sur la structure du système, notamment les relations entre les objets, les classes, les composants, et la manière dont les différentes parties du système sont organisées. Voici les principaux diagrammes statiques utilisés :

**Diagrammes de classe** : Représentent les entités principales (ou classes) d'un système ainsi que leurs attributs, méthodes et relations (associations, héritages, agrégations, etc.).

Exemple : Une classe Utilisateur peut avoir des attributs comme nom, email et des méthodes comme s’inscrire () ou se connecter ().

**Les diagrammes d’objets**: Représentent les instances spécifiques des classes (les objets) à un moment donné, avec leurs valeurs réelles.

Exemple : Un objet de la classe Utilisateur peut avoir les valeurs nom = [youssouf@exemple.com](mailto:youssouf@exemple.com)

**Les diagrammes de cas d’utilisation** : Décrivent les fonctionnalités principales d'un système du point de vue des utilisateurs (appelés *acteurs*).

Exemple : Un cas d'utilisation peut être *Ajouté une matière* ou *Recevoir une notification de révision*.

**Les diagrammes de composants** : Montrez comment les différentes parties logicielles (ou composants) du système interagissent.

Exemple : Un composant pour gérer les notifications peut être relié à un composant pour planifier des rappels.

**Les diagrammes de déploiement** : Illustre comment le système est physiquement déployé sur le matériel, y compris les serveurs, les appareils mobiles, etc.

Exemple : L'application *Études Smart* déployée sur un smartphone avec un serveur distant pour stocker les données.

**Vues dynamiques :**

Les vues dynamiques se concentrent sur le comportement du système, c'est-à-dire la manière dont les objets interagissent entre eux et évoluent dans le temps. Voici les principaux diagrammes dynamiques utilisés :

**Les diagrammes de séquence** : Décrivent les interactions entre les objets et les acteurs sous forme de messages échangés dans un ordre chronologique.

Exemple : Lorsqu'un utilisateur planifie une révision, les messages peuvent inclure Saisir une date, Envoyer la notification, etc.

**Les diagrammes de collaboration**: Montrent les interactions entre objets ou composants mais se concentrent davantage sur la structure et les relations que sur la chronologie.

**Les diagrammes d’états-transition** : Représenter les différents états d'un objet et comment il passe d'un état à un autre en réponse à des événements.

Exemple : Une révision peut être dans l'état Planifié, puis passer à l'état Terminée après notification.

**Les diagrammes** **d’activités** : Décrivent les étapes successives d'un processus ou d'une activité.

Exemple : Le processus pour importer un fichier comprend les étapes suivantes : *Sélectionner un fichier 🡪 Valider 🡪 Charger le contenu*.

Toutefois, malgré l’omniprésence de ce dernier dans des projets informatique, UML ne présente de point forts mais aussi des points faibles.

**Les points forts :**

UML présente de nombreux avantages qui en font un langage incontournable dans les projets informatiques :

⮚Langage formel et normalisé:

Un gain de précision : Grace à ses notations standardisées, IML permet une description claire et précise des systèmes

Un gage de stabilité : La normalisation UML garantit une cohérence dans la modélisation des projets.

Utilisation d’outils : UML est largement pris en charge par une variété d’outils logiciels, ce qui facilite la creation, l’analyse et la gestion des diagrammes.

⮚ Support de communication performant :

UML offre un cadre structuré qui facilite la collaboration entre les parties prenantes (développeurs, analystes, clients, etc…).

Il permet de simplifier la compréhension de systèmes complexes grâce à des représentations visuelles abstraites mais claires.

**Les points faibles :**

Malgré ses avantages, UML présente également certaines limites :

⮚Complexités :

La diversité () des diagrammes et la richesse des fonctionnalités peuvent rendre UML difficile à maitriser pour les débutants.

⮚Temps et efforts :

La creation et la maintenance des diagrammes UML peuvent demander beaucoup de temps, surtout pour les projets en évolution rapide.

⮚ Flexibilité limitée :

UML, en tant que standard, peut être perçu comme trop rigide () dans certains contextes spécifiques ou des adaptations sont nécessaires.

⮚ Cout des outils :

Bien que certains outils UML soient gratuits, les versions professionnelles et plus complètes peuvent être couteuses.

## **III.1.2. Outils de modélisation**

Les outils de modélisation jouent un rôle dans la conception et l’analyse des systèmes informatiques. Ils permettent de représenter graphiquement les aspects structurels d’outils de modélisation : [9]

⮚ **Outils de modélisation graphique :**

Ces outils permettent de créer des diagrammes pour représenter les différents aspects d’un système (structure, comportement, interactions, etc…).

**Lucidchart :**

Un outil en ligne pour créer des diagrammes UML, des flux de travail, et d’autres représentations visuelles.



**Figure 19 : image de Lucidchart.**

**⮚Outils spécialisés pour UML :**

Ces outils sont conçus spécifiquement pour créer et manipuler des diagrammes UML.

**StarUML :**

StarUML est un outil de modélisation UML puissant et populaire. Il est utilisé par les développeurs pour concevoir des logiciels grâce à une interface intuitive et des fonctionnalités avancées.

****

**Figure 20: image de StarUML**

**AgroUML :**

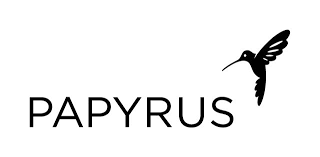
AgroUML est un outil open-source pour la modélisation UML. Il est conçu pour créer et éditer des diagrammes UML de manière complète, tout en restant gratuit et accessible.



**Figure 21: image d’AgroUML**

**Papyrus**

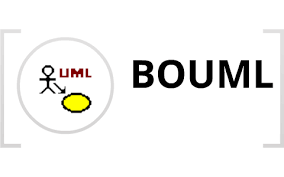
Papyrus est un outil open-source basé sur la plateforme Eclipse pour la modélisation UML. Il est souvent utilisé dans des projets complexes ou académiques



**Figure 22 : image de Papyrus**

**BOUML :**

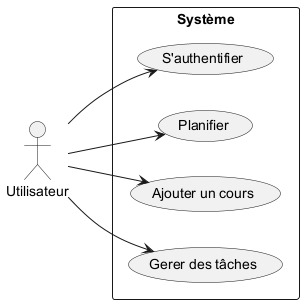
BOUML est un outil UML rapide et léger, connu pour sa simplicité et son efficacité dans les projets de modélisation.



**Figure 23: image de BOUML**

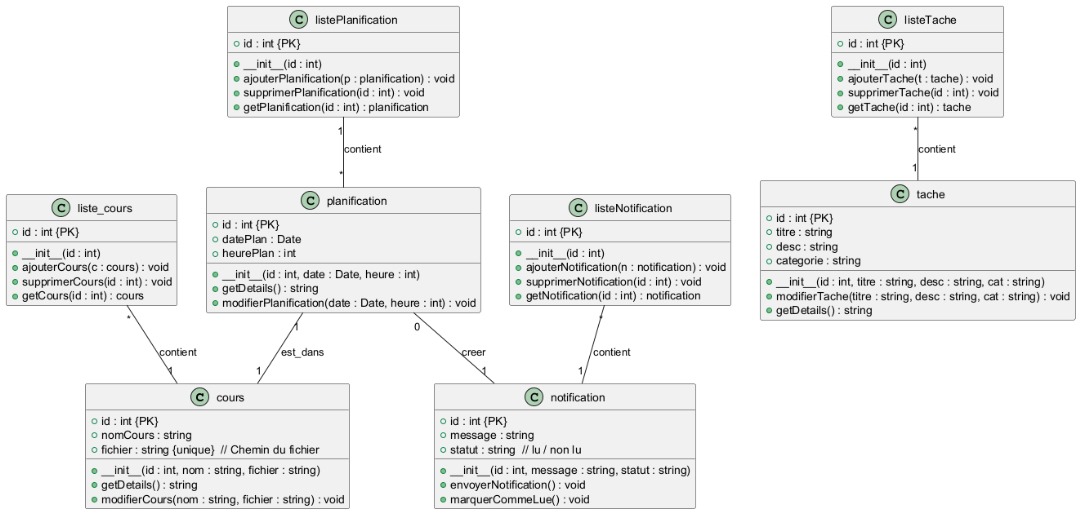
**III.1.3. Diagrammes UML utilisé**Dans cette section, nous présentons les trois(3) principaux diagrammes UML utilisés pour modéliser l’application Études\_Smart :

## **III.1.3.1 Diagramme de cas d’utilisation**



**Figure 24: Diagramme de cas d’utilisation**

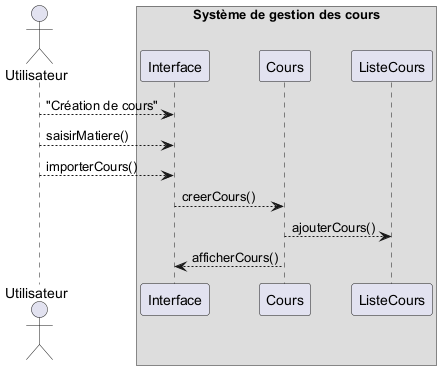
**III.1.3.2. Diagramme de classe**

****

**Figure 25 : Diagramme de classe**

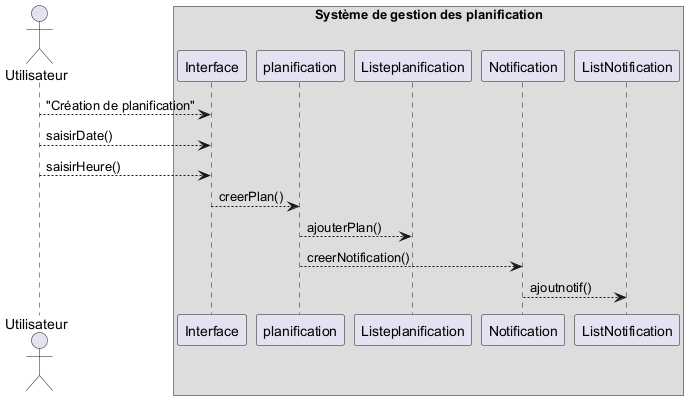
## **III.1.3.3. Diagrammes de séquence**

* **Système de gestion des cours**

****

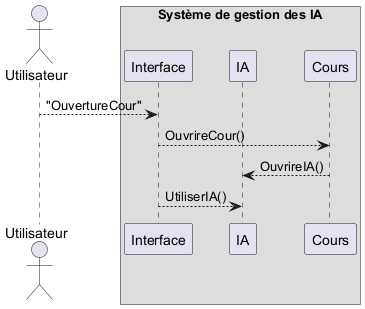
**Figure 26: Diagramme de séquence (Système de gestion des cours)**

**Système de gestion des planifications**

****

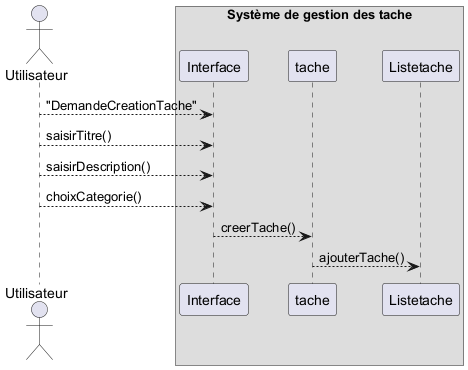
**Figure 27 : Diagramme de séquence (Système de gestion des planifications)**

* **Système de gestion de l’intelligence Artificielle (IA)**

****

**Figure 28: diagramme de séquence (Système de gestion de l’intelligence Artificielle (IA))**

**Système de gestion des taches**

****

**Figure 29 : Diagramme de séquence (Système de gestion des taches)**

L’utilisation des diagrammes de cas d’utilisation, diagramme des séquences et diagramme de classe a permis de mieux structurer et comprendre la fonctionnement de l’application Etudes Smart. Ces modèles facilitent le développement et garantissent une architecture claire et maintenable.

# **Chapitre IV : Intégration de l’intelligence Artificielle dans l’application Etudes\_Smart**

L’intégration de l’intelligence artificielle (IA) dans **Études Smart** représente une avancée majeure pour améliorer l’expérience utilisateur et optimiser les révisions des étudiants. Grâce à cette technologie, l’application peut **répondre aux questions des utilisateurs en temps réel**.

L’objectif de cette section est d’expliquer comment l’IA a été intégrée à l’application et les bénéfices qu’elle apporte.

Après une analyse comparative basée sur des critères tels que **la facilité d’intégration, la pertinence des réponses et les coûts d’utilisation**, nous avons opté pour **une API de génération de texte** qui permet d’obtenir des réponses intelligentes et adaptées aux besoins des étudiants.

L’intégration de l’IA repose sur une **communication entre l’application et une API externe** via des requêtes réseau. Ce processus comprend plusieurs étapes essentielles :

* **Connexion à l’API** : L’application établit une connexion sécurisée avec l’API en envoyant une requête contenant la question ou le texte à analyser.
* **Traitement par l’IA** : L’API utilise un modèle de traitement du langage pour analyser la demande et générer une réponse pertinente.
* **Affichage de la réponse** : L’application récupère le résultat et l’affiche à l’utilisateur sous forme de texte structuré.

L’intelligence artificielle apporte une réelle **valeur ajoutée** à l’application **Études Smart** en offrant un accompagnement intelligent et personnalisé aux étudiants. Grâce à l’IA, les utilisateurs bénéficient d’un outil interactif qui leur permet d’optimiser leurs révisions et d’améliorer leur compréhension des cours.

L’avenir de cette technologie dans l’application repose sur l’amélioration continue des fonctionnalités et l’intégration de nouvelles solutions adaptées aux besoins des étudiants.

# **Chapitre V : Présentation des interfaces de l’application**

**Objectifs du Projet**

L’application Études Smart vise à répondre aux besoins spécifiques des étudiants en proposant une plateforme numérique intuitive et efficace. Les objectifs principaux du projet sont les suivants :

* Faciliter l’organisation et la gestion des études

En offrant une interface utilisateur simple, fluide et conviviale, Études Smart accompagne les étudiants dans la planification et l'organisation de leurs études. L'objectif est de réduire la complexité souvent associée à la gestion de multiples matières, chapitres, et délais.

* Proposer des fonctionnalités personnalisées

Études Smart met à disposition des outils interactifs permettant aux utilisateurs d'importer leurs cours sous différents formats, de planifier leurs révisions et de structurer leurs activités académiques selon leurs besoins spécifiques.

* Améliorer l’expérience d’apprentissage

Grâce à l’intégration de technologies modernes, telles que des notifications pour les rappels et l’intelligence artificielle (IA), Études Smart vise à rendre l’apprentissage plus dynamique, engageant, et stimulant pour les étudiants.

* Encourager l’autonomie et optimiser la gestion du temps

L’application permet aux utilisateurs de suivre leur progression académique, d’identifier leurs forces et leurs insuffisances, et de mieux gérer leur temps grâce à une planification structurée et des outils de suivi intégrés.

Ces objectifs reflètent la vision globale d’Études Smart : transformer les défis académiques des étudiants en opportunités d’apprentissage et de développement personnel.

**Solution Apportée**

Pour répondre aux problématiques identifiées et atteindre ses objectifs, l’application Études Smart propose une solution complète qui regroupe plusieurs fonctionnalités innovantes et pratiques :

* Gestion de matières

Les étudiants peuvent importer leurs cours au format PDF ou autres types de fichiers directement depuis leur appareil.

Chaque matière est ajoutée sous forme d’une fiche détaillée, avec des options pour consulter, modifier ou supprimer les informations selon les besoins.

* Planification des révisions

Les utilisateurs peuvent planifier leurs révisions de manière structurée en sélectionnant une matière spécifique et en associant une date et une heure pour planifier son cours et recevoir de notifications pour commencer de réviser son cour.

Cette fonctionnalité inclut une visualisation claire des sessions de révision prévues, aidant ainsi à éviter les oublis ou les chevauchements.

* Notifications intelligentes

Études Smart envoie des rappels automatiques pour alerter les étudiants des séances de révision programmées.

Ces notifications sont paramétrables pour correspondre aux préférences individuelles des utilisateurs, garantissant une régularité dans les révisions.

* Intelligence Artificielle (IA)

Études Smart intègre une IA permettant à l’utilisateur de poser des questions Cette fonctionnalité aide les étudiants à se concentrer sur les points clés du contenu, tout en économisant du temps lors des révisions.

* Interface utilisateur conviviale

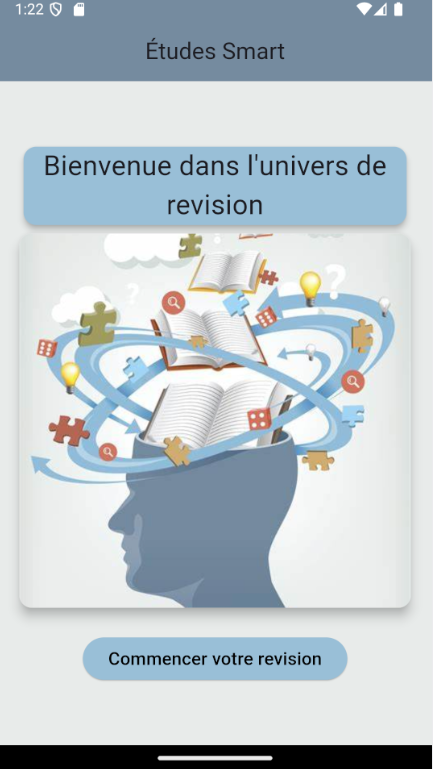
L’application est conçue pour être intuitive et agréable à utiliser, avec des listes organisées, des icônes explicites, et des boutons d’action clairs.

La navigation fluide entre les différentes sections garantit une expérience utilisateur optimale, même pour les débutants.

On va commencer par la présentation la première interface de l’application qui affiche à

l’utilisateur l’accède à notre application.

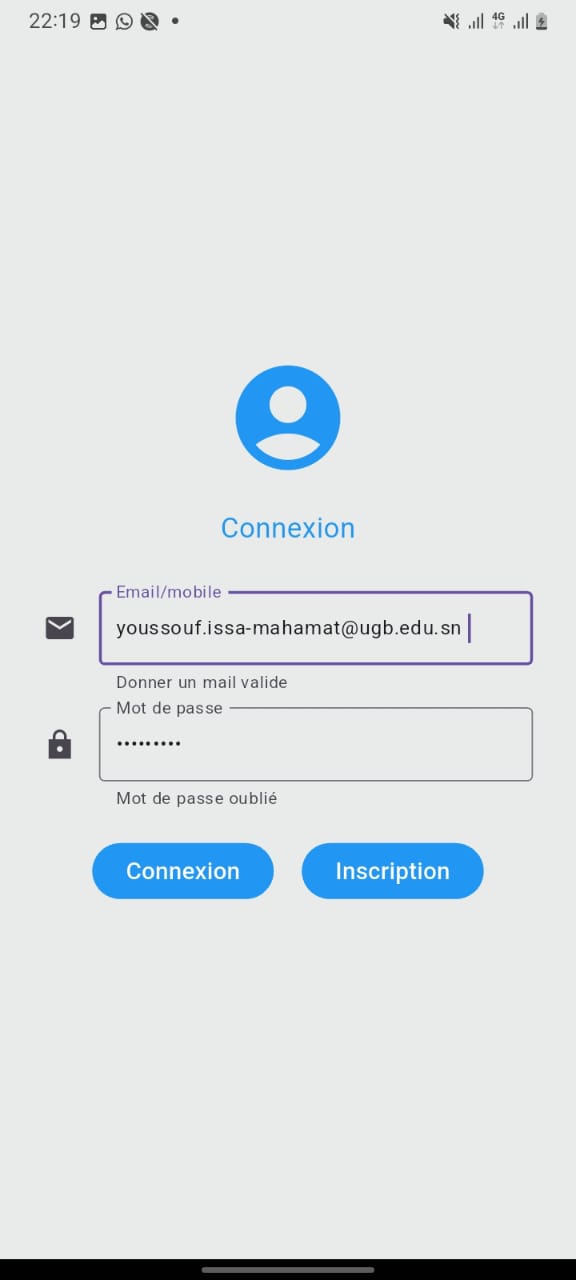
Cet écran permet à l’utilisateur d’accéder à l’application. Il affiche un bouton pour commencer votre révision et peut inclure une brève présentation de l’application



**Figure 30 : présentation la première interface**

**Page de connexion**

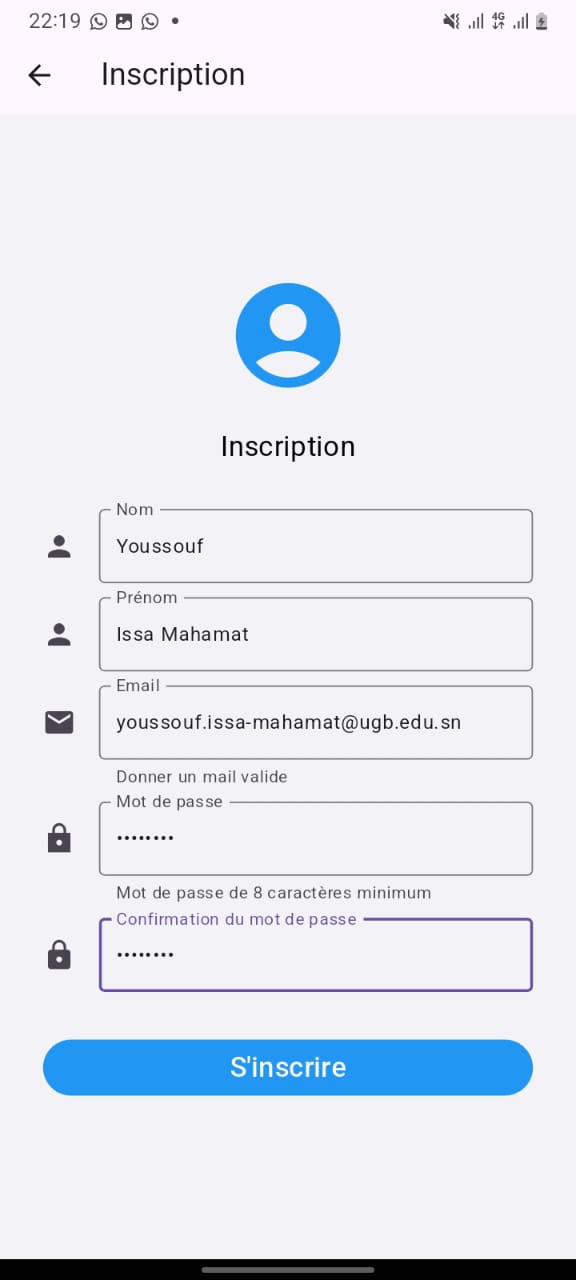
Permet aux utilisateurs existant de se connecter avec leur email et mot de passe. Possibilité d’afficher un message d’erreur en cas de mauvais identifiants.



**Figure 31 : page de connexion**

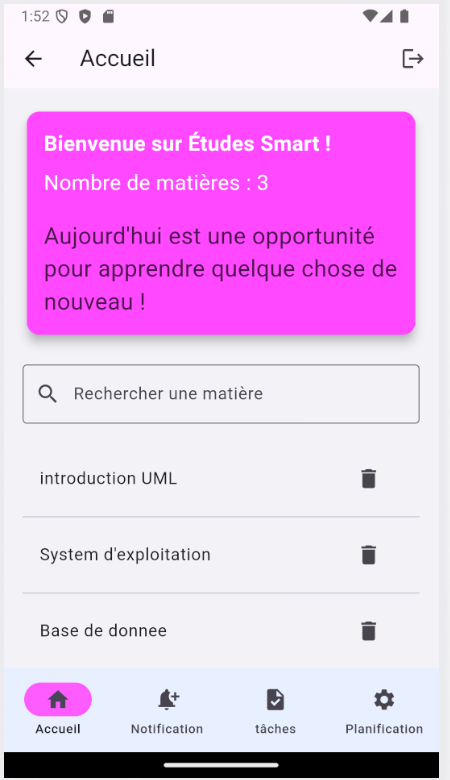
**Page d’inscription**

Permet aux niveaux utilisateurs de créer un compte en renseignant leur nom, prénom, email et mot de passe et confirmation du mot de passe. Un bouton permet de valider l’inscription et de rediriger vers la page d’accueil après succès.



**Figure 32: page d’inscription.**

Cette représente la page d’accueil de l’application. Elle offre une vue d’ensemble des matières enregistrées et permet aux utilisateurs d’y accéder facilement



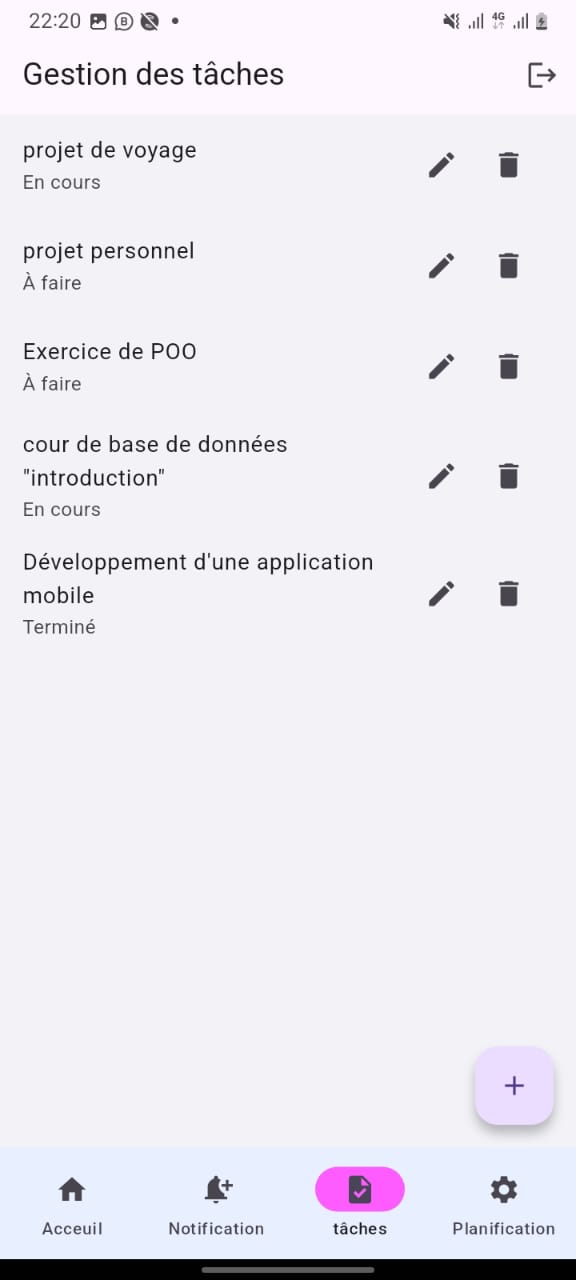
**Figure 33 : page d’accueil**

Page des  notifications : affiche les rappels programmés pour aider l’utilisateur à respecter son planning. Indique la matière et l’heure de la révision prévue



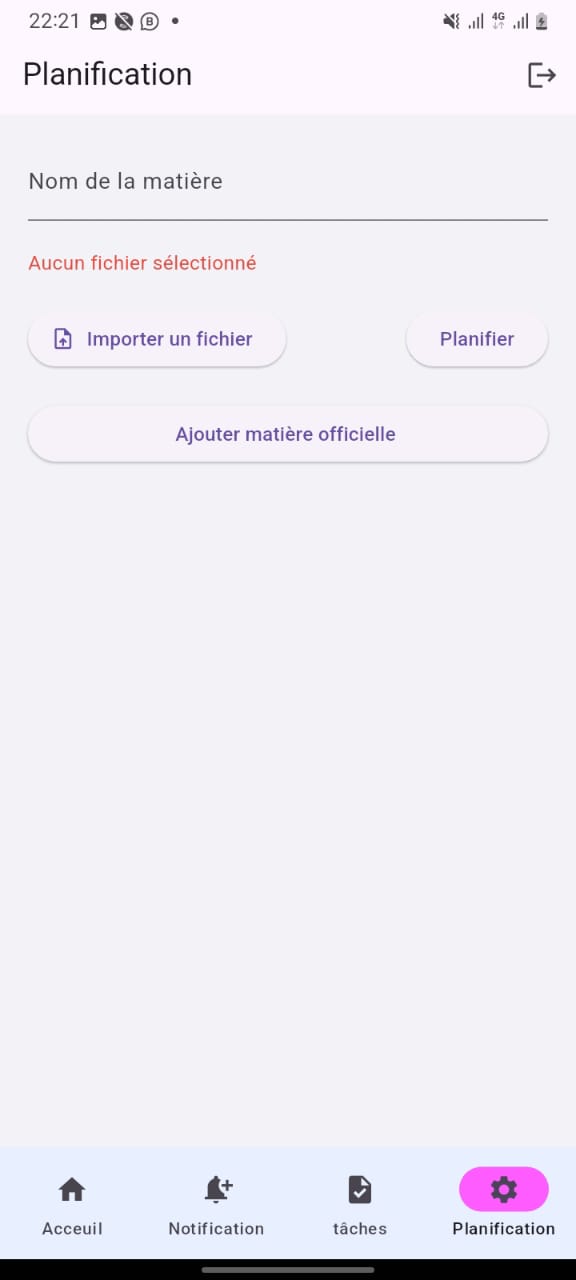
**Figure 34 : page des notifications**

Cette capture d’écran présente la page de gestion des taches de l’application. Elle permet à l’utilisateur d’organiser ses taches selon leur statut : « A faire », « En Cours » et Terminé ». Cela aiderait à mieux gérer le travail à accomplir.



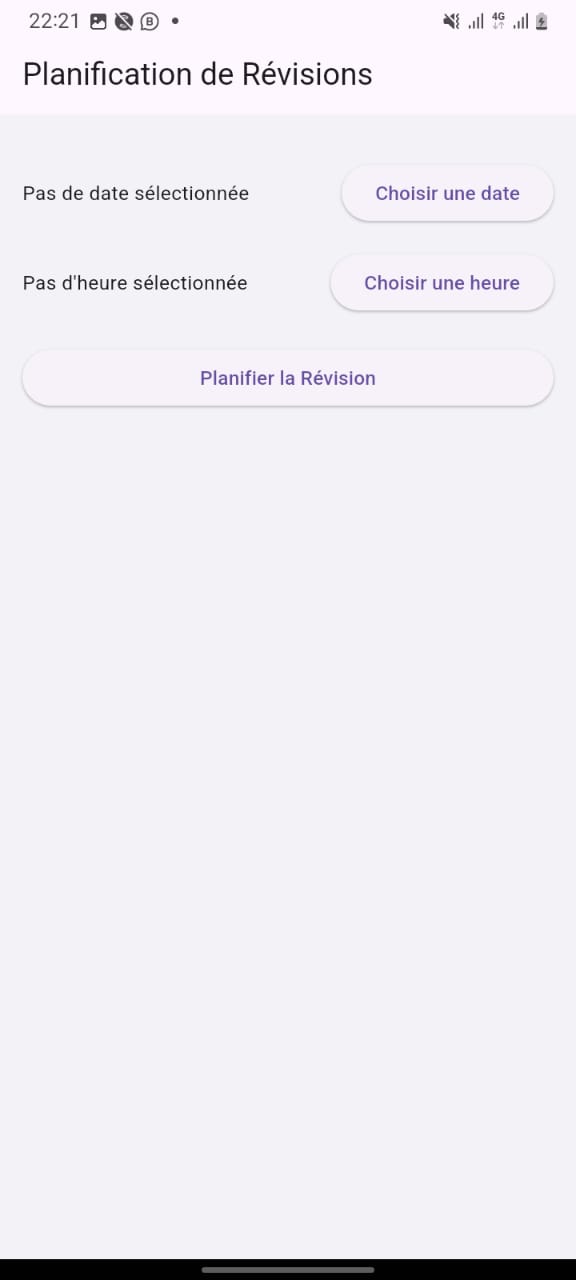
**Figure 35 : page de Gestion des taches.**

Cette interface simple permet aux utilisateurs de créer leur propre emploi du temps en quelque clics . Il suffit d’entrer le nom de la matière, d’importer les documents et de définir les dates de révision.



**Figure 36 : page de la planification.**

Cette interface permet aux utilisateurs de créer un planning de révision personnalisé. Ils peuvent définir la date et l’heur auxquelles ils souhaitent reviser.



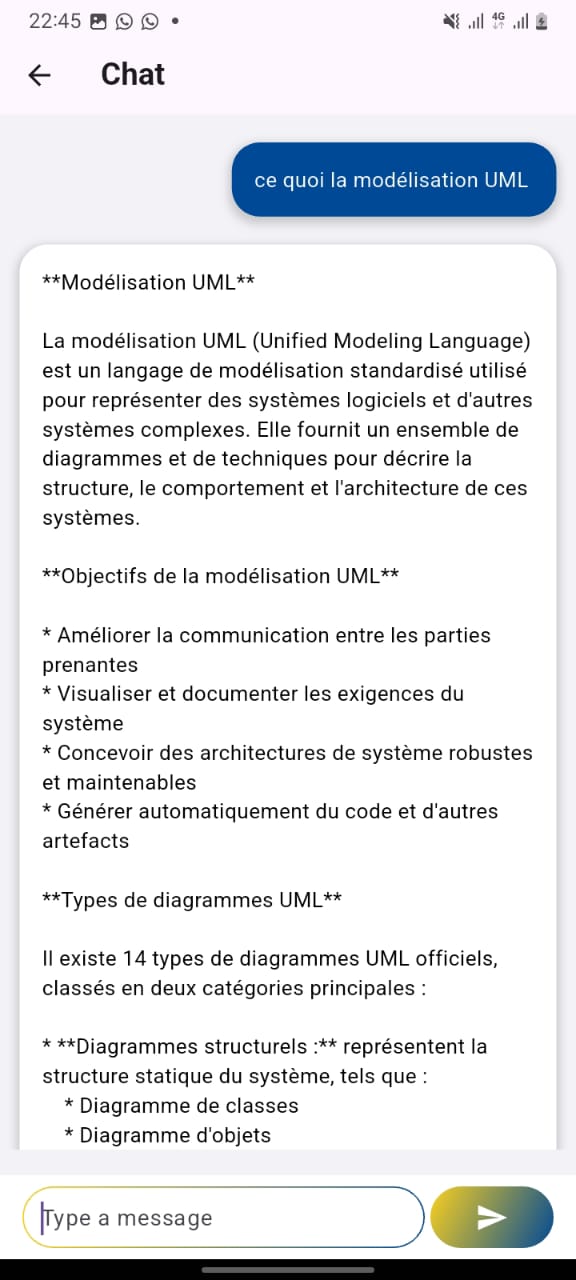
**Figure 37 : page de planification de révision.**

Cette interface est conçue pour faciliter la lecture de documents PDF sur un appareil mobile. Elle offre une expérience de lecture simple et intuitive, permettant aux utilisateurs de parcourir facilement les pages, de zoomer sur le texte et d’accéder rapidement aux informations.



**Figure 38 : page de Lecture.**

Cette partie de chat pour l’intelligence Artificiel si l’utilisateur demande une question l’IA répond automatiquement. Juste espace d’échange l’utilisateur avec IA. Champ de saisie dédié aux questions et un bouton d’envoi .



**Figure 39 : page de l’Intelligence Artificiel**

# **Conclusion Générale et Perspectives**

L’objectif de notre travail était de concevoir et développer une application mobile multiplateforme permettant aux étudiants d’optimiser la gestion de leurs révisions. Études Smart se veut être un outil innovant, offrant des fonctionnalités telles que l’importation et l’organisation des matières, la planification des révisions avec rappels automatiques, ainsi que l’intégration de l’intelligence artificielle pour générer des résumés et répondre aux questions des utilisateurs.

Tout au long du projet, nous avons étudié les différentes technologies adaptées au développement mobile, et avons choisi Flutter pour garantir une application fluide et performante sur Android et iOS. Nous avons également intégré Firebase pour la gestion des données et les notifications, et utilisé SQLite pour le stockage local des informations.

La réalisation de ce projet nous a permis d’acquérir des compétences approfondies en développement mobile, modélisation UML, gestion des bases de données et intégration de systèmes avancés comme l’IA. Nous avons également relevé plusieurs défis techniques, notamment l’optimisation des performances, la gestion des notifications et l’amélioration de l’expérience utilisateur.

**Perspectives d’Amélioration**

Afin de rendre Études Smart encore plus efficace et adaptée aux besoins des étudiants, plusieurs axes d’amélioration peuvent être envisagés :

* Amélioration par l’intelligence artificielle

Intégrer une IA plus avancée pour analyser et suggérer des plans de révisions optimisés en fonction du niveau de l’étudiant.

* Collaboration et partage

Ajouter une fonctionnalité permettant aux étudiants de partager leurs notes et plannings de révisions avec leurs camarades.

* Développement d’une version web

Étendre l’application à une version web, pour permettre une accessibilité depuis un ordinateur.

* Ajout d’une fonctionnalité de suivi des performances

Permettre aux utilisateurs de suivre leurs progrès académiques via des statistiques et des rapports détaillés.

Avec Études Smart, nous avons développé une solution qui répond aux défis des étudiants en matière d’organisation et de productivité. Grâce à ses fonctionnalités innovantes et son intégration avec des technologies modernes, elle constitue une véritable aide numérique pour l’apprentissage.

En poursuivant son amélioration, cette application pourrait devenir un outil incontournable pour les étudiants du monde entier, en leur offrant une gestion efficace du temps et des révisions adaptées à leurs besoins.

# **Références-Bibliographique**

[1] Cour de Professeur Massa Mbaye « installation de Flutter et configuration de différente » (consulté le 04 octobre 2024)

[2] <https://developer.android.com/studio> « installation de android studio » (consulté le 04 octobre 2024)

[3] <https://code.visualstudio.com> « installation de vs code » (consulté le 04 octobre 2024)

[4] <https://git-scm.com/> consulté en novembre 2024

[5] <https://github.com> consulté en novembre 2024

[6] <https://codabee.com/flutter> consulté en décembre 2024

[7] <https://flutter.dev/> consulté en décembre 2024

[8] Cour de Professeur David Célestin FAYE « Modélisation UML » consulté en janvier

[9] <https://en.wikipedia.org/wiki/List_of_Unified_Modeling_Language_tools> « liste de UML complet »

* https://www.youtube.com/watch?v=sIiEf457bS0 “videos YouTube cote firebase” consulté le 25 décembre 2024
* <https://formation.drissas.com/flutter-revolution/> “ consulté le 25 décembre 2024
* <https://cours-informatique-gratuit.fr/dictionnaire/> «  Dictionnaire mobile »
* <https://www.google.com/search?q=list+api&oq=list+api&gs_lcrp=EgZjaHJvbWUyCQgAEEUYORiABDIHCAEQABiABDIHCAIQABiABDIICAMQABgWGB4yCAgEEAAYFhgeMgYIBRBFGDwyBggGEEUYPDIGCAcQRRg80gEINzEyMGowajeoAgCwAgA&sourceid=chrome&ie=UTF-8> « Liste des API » consulté le 2 janvier 2025
* <https://www.youskil.com/maitriser-la-consommation-dapi-rest-en-flutter-guide-pratique/> « site pour les API de IA » consulté le 4 janvier 2025
* <https://plantuml.com/fr/> consulté le 06 janvier 2025