

Institut Supérieur de Comptabilité et d'Administration des Entreprises
Nouakchott-Mauritanie



PROJET DE FIN D'ETUDES

Pour l'obtention de :

Licence en Développement Informatique

Thème :

**Application mobile pour une
bibliothèque d'audio**

Réalisé par :

Mohamed Abdellahi Mohamed	I18289
Maawiya Yahya Abdi	I18214
Mohamed Salem El Moctar Yahya	I18221

Encadré par :

Dr. Ahmed Sejad

Année Universitaire : 2022– 2023

Dédicaces

Toutes les lettres ne peuvent pas suffire à exprimer toute notre gratitude. Tous les mots ne peuvent pas véritablement exprimer l'amour, le respect et la reconnaissance que nous ressentons à votre égard. C'est pourquoi nous dédions ce souvenir à nos chers parents.

Aucune quantité d'encre au monde ne serait suffisante pour décrire pleinement nos sentiments envers vous. Vous avez toujours été notre source inépuisable de patience et de confiance.

Ce travail est le fruit de votre dévouement et de vos sacrifices, ainsi que du soutien et des encouragements constants que vous nous apportez. Nous espérons sincèrement que vous y trouverez la récompense de tout ce que vous avez semé, ainsi que le témoignage de notre fierté indéfectible envers vous.

Remerciement

Nous n'aurons pas commencé ce rapport sans remercier ALLAH, le plus puissant, le plus miséricordieux, qui nous a donné la santé et la volonté pour mener à bien ce projet, et qui est à l'origine de toute réussite dans nos vies.

Tout d'abord, nous tenons à remercier nos Parents et tous les membres de nos familles qui ont assuré le succès de notre formation à travers leur soutien par tous les moyens. Surtout nos mères pour leur soutien constant.

Nous adressons également nos remerciements à Dr. Ahmed Sejad, pour nous avoir donné l'opportunité de nous former et nous avoir aidés à réaliser ce projet

Nous remercions tous les Docteurs et Professeurs qui nous ont enseigné durant les années passées et la période actuelle. Nous saluons leurs grands efforts pour nous enseigner et leur excellente explication.

Et enfin, nous remercions tous nos amis et collègues qui nous ont aidés durant nos études, et plus particulièrement durant notre projet de fin d'études.

Table de Matières

Dédicaces	1
Remerciement.....	2
Résumé.....	6
Chapitre 1 : Introduction générale.....	7
Chapitre 2 : Présentation générale.....	8
Introduction.....	8
Cadre de projet.....	8
I.1 Etude de l'existence	8
I.2 Problématique	9
I.3 Objectifs.....	9
I.4 Solutions proposées.....	10
I.5 Les besoins fonctionnels	12
I.6 Les besoins non fonctionnels	12
Conclusion	13
Chapitre 3 : Environnement du projet	14
Introduction.....	14
Les technologies utilisées.....	14
• Dart	14
• Flutter.....	16
• Firebase	17
• HTML.....	18
• CSS.....	19
• Javascript.....	19
Chapitre 4 : Conception.....	21

Introduction.....	21
UML.....	21
Diagramme de classe	21
Diagramme de classe du sujet	22
Diagramme de cas d'utilisation	22
Diagramme de cas d'utilisation du sujet	23
Conclusion	23
Chapitre 5 : Réalisation	25
Chapitre 6 : Conclusion et perspectives	26
Bibliographie.....	27

LISTE DES FIGURES :

Figure 1 Logo Dart.....	14
Figure 2 Logo Flutter	16
Figure 3 Logo Firebase	17
Figure 4 Logo HTML	18
Figure 5 Logo CSS.....	19
Figure 6 Logo Javascript	19
Figure 7 diagramme de class du sujet	22
Figure 8 diagramme de cas d'utilisation du sujet (Administrateur)	23
Figure 9 diagramme de cas d'utilisation du sujet (Utilisateur)	23

Résumé

Le projet consiste en une application mobile de bibliothèque audio offrant des fonctionnalités essentielles.

Les utilisateurs peuvent chercher des livres, les lire et les sauvegarder en tant que favoris.

L'application permet également de consulter l'historique des livres consultés.

De plus, les utilisateurs ont la possibilité de télécharger à la fois l'audio et le PDF du livre pour une expérience de lecture flexible. Avec cette application conviviale, les amateurs de livres audios peuvent profiter d'une bibliothèque virtuelle complète et d'une lecture pratique à tout moment, n'importe où.

Chapitre 1 : Introduction générale

L'introduction générale du projet porte sur une application mobile de bibliothèque audio offrant une gamme de fonctionnalités pratiques pour les amateurs de livres audio.

L'application permet aux utilisateurs de chercher, lire et sauvegarder des livres, ainsi que de consulter leur historique de lecture.

De plus, les utilisateurs ont la possibilité de télécharger à la fois l'audio et le PDF du livre pour une expérience de lecture flexible.

Cette application est accompagnée d'une interface web dédiée à l'administrateur, qui lui permet de gérer les livres, les catégories et d'ajouter de nouvelles sources.

L'authentification est mise en place à l'aide de Firebase, où l'administrateur utilise une connexion par e-mail et mot de passe, tandis que les utilisateurs se connectent via leur numéro de téléphone et un code OTP (One Time Password).

Firebase est utilisé comme une source de données centralisée, incluant les collections de données pour les livres, les catégories et l'authentification.

La partie web est développée en HTML, CSS, JavaScript et Bootstrap, en utilisant Firebase comme backend.

Quant à l'application mobile, elle est développée en utilisant Flutter et Firebase pour offrir une expérience fluide et interconnectée entre les différents appareils.

Chapitre 2 : Présentation générale

Introduction

Dans ce chapitre nous allons commencer par une détermination du cadre du projet, ensuite la deuxième partie du chapitre sera une explication de la situation existante par rapport au sujet de notre projet, les problématiques, les objectifs, les solutions proposées, la troisième partie de ce chapitre est une énumération des besoins fonctionnels et non fonctionnels.

Cadre de projet

Le projet entre dans le cadre de présentation du mémoire de fin d'études pour l'obtention de la licence appliquée en Développement Informatique (DI), c'est un développement d'une application mobile bibliothèque audio et une application web pour gérer les livres de la bibliothèque.

I.1 Etude de l'existence

L'existence de ce projet est justifiée par la popularité croissante des livres audio et la demande croissante d'applications mobiles offrant une expérience de lecture pratique et flexible. Les livres audios permettent aux utilisateurs de profiter de la lecture tout en étant en déplacement, ce qui les rend populaires parmi les personnes occupées ou celles qui préfèrent écouter plutôt que lire.

L'application mobile de bibliothèque audio répond à ce besoin en offrant une gamme de fonctionnalités essentielles pour les utilisateurs. La possibilité de chercher des livres, de les lire et de les sauvegarder en tant que favoris permet aux utilisateurs de trouver facilement du contenu intéressant et de le conserver pour une utilisation ultérieure. L'historique de lecture fournit également un moyen pratique de suivre les livres déjà consultés.

I.2 Problématique

Comment concevoir et développer une application mobile de bibliothèque audio offrant des fonctionnalités essentielles pour les utilisateurs, tout en assurant une expérience fluide et interconnectée entre l'application mobile et l'interface web pour l'administrateur ?

- Comment créer une interface conviviale et intuitive pour permettre aux utilisateurs de chercher, lire et sauvegarder des livres audios facilement ?
- Comment mettre en place un système de gestion efficace des livres, des catégories et des nouvelles sources pour l'administrateur via l'interface web dédiée ?
- Comment garantir une expérience de lecture flexible en permettant aux utilisateurs de télécharger à la fois l'audio et le PDF des livres ?
- Comment assurer la sécurité et l'authentification des utilisateurs et de l'administrateur en utilisant Firebase comme source de données centralisée ?
- Comment optimiser les performances de l'application mobile pour une expérience fluide sur différents appareils et plateformes ?

La résolution de ces problématiques permettra de développer une application mobile de bibliothèque audio complète et conviviale, répondant aux besoins des utilisateurs et offrant une expérience de lecture pratique et flexible.

I.3 Objectifs

- Concevoir une interface utilisateur conviviale et intuitive pour permettre aux utilisateurs de chercher, lire et sauvegarder facilement des livres audios.
- Mettre en place un système de gestion efficace des livres, des catégories et des sources pour l'administrateur via l'interface web dédiée.
- Permettre aux utilisateurs de télécharger à la fois l'audio et le PDF des livres pour une expérience de lecture flexible.

- Assurer la sécurité et l'authentification des utilisateurs et de l'administrateur en utilisant Firebase comme source de données centralisée.
- Optimiser les performances de l'application mobile pour une expérience fluide sur différents appareils et plateformes.
- Offrir une expérience interconnectée entre l'application mobile et l'interface web pour permettre une gestion transparente des livres et des utilisateurs.
- Fournir une bibliothèque audio complète avec une large sélection de livres pour répondre aux intérêts et aux préférences des utilisateurs.
- Permettre aux utilisateurs de suivre leur historique de lecture pour une gestion personnalisée de leur expérience de lecture.
- Proposer une interface web intuitive pour l'administrateur afin de faciliter la gestion des livres, des catégories et des sources.
- Garantir une expérience utilisateur fluide et sans heurts lors de la recherche, de la lecture et de la sauvegarde des livres audio.
- Assurer la scalabilité du système pour gérer efficacement une grande quantité de livres et d'utilisateurs.
- Recueillir des commentaires et des évaluations des utilisateurs pour continuer à améliorer et enrichir l'expérience de la bibliothèque audio.

I.4 Solutions proposées

1. Conception d'une interface utilisateur conviviale : Créer une interface mobile attrayante et facile à utiliser, avec des fonctionnalités de recherche avancées, des filtres de catégories et une mise en page intuitive pour permettre aux utilisateurs de trouver rapidement les livres audios qu'ils recherchent.

2. Développement d'un système de gestion efficace : Mettre en place une interface web dédiée à l'administrateur, offrant des fonctionnalités de gestion des livres, des catégories et des sources. Permettre à l'administrateur d'ajouter, modifier et supprimer des livres, ainsi que de gérer les informations associées.

3. Téléchargement flexible du contenu : Implémenter la possibilité pour les utilisateurs de télécharger à la fois l'audio et le PDF des livres audio. Cela leur permettra de choisir le format de lecture qui correspond le mieux à leurs préférences et à leur situation.
4. Sécurité et authentification : Utiliser Firebase comme source de données centralisée et pour gérer l'authentification des utilisateurs et de l'administrateur. Mettre en place des mécanismes de sécurité robustes pour protéger les données sensibles et garantir que seules les personnes autorisées peuvent accéder à l'application et à ses fonctionnalités.
5. Optimisation des performances : Optimiser l'application mobile pour assurer une expérience fluide et réactive, quelle que soit la plateforme ou l'appareil utilisé. Réduire les temps de chargement, optimiser la consommation d'énergie et assurer une compatibilité avec différents systèmes d'exploitation et versions.
6. Expérience interconnectée : Assurer une communication fluide entre l'application mobile et l'interface web dédiée à l'administrateur. Permettre à l'administrateur de gérer les livres et les utilisateurs de manière transparente, en synchronisant les données entre les deux interfaces.
7. Bibliothèque audio complète : Offrir une vaste sélection de livres audio dans différentes catégories, genres et langues pour satisfaire les intérêts et les préférences des utilisateurs. Collaborer avec des éditeurs et des auteurs pour obtenir une bibliothèque riche et diversifiée.
8. Historique de lecture personnalisé : Permettre aux utilisateurs de consulter leur historique de lecture, y compris les livres consultés et les chapitres lus. Cela leur permettra de reprendre facilement leur lecture là où ils l'ont laissée et de gérer leur expérience de lecture de manière personnalisée.
9. Interface web intuitive pour l'administrateur : Concevoir une interface web conviviale et intuitive pour l'administrateur, avec des fonctionnalités de gestion claires et faciles à utiliser. Cela facilitera la gestion des livres, des catégories et des sources, ainsi que l'ajout de nouvelles fonctionnalités ou sources de contenu.

10. Tests, évaluations et améliorations continues : Effectuer des tests approfondis pour garantir la qualité de l'application et de l'interface web. Recueillir les commentaires et les évaluations des utilisateurs pour identifier les points d'amélioration et mettre en œuvre des mises à jour régulières afin d'enrichir l'expérience de la bibliothèque audio.

I.5 Les besoins fonctionnels

Recherche de livres audios : Les utilisateurs doivent pouvoir chercher des livres audios par titre, auteur, catégorie, etc.

Lecture de livres audio : Les utilisateurs doivent pouvoir lire les livres audios directement depuis l'application.

Sauvegarde des favoris : Les utilisateurs doivent pouvoir sauvegarder des livres audios en tant que favoris pour y accéder facilement.

Historique de lecture : Les utilisateurs doivent pouvoir consulter leur historique de lecture pour suivre les livres déjà consultés.

Téléchargement d'audio et de PDF : Les utilisateurs doivent pouvoir télécharger à la fois l'audio et le PDF des livres audio.

Interface d'administration : L'administrateur doit disposer d'une interface web dédiée pour gérer les livres, les catégories et les sources.

Authentification : Utiliser Firebase pour mettre en place un système d'authentification sécurisé pour les utilisateurs et l'administrateur.

I.6 Les besoins non fonctionnels

1. Fiabilité : L'application doit être fiable et fonctionner de manière stable, sans plantages fréquents ou erreurs majeures.
2. Gestion des erreurs : L'application doit être capable de gérer les erreurs de manière appropriée, en affichant des messages d'erreur clairs et en proposant des solutions ou des alternatives lorsque des problèmes surviennent.
3. Convivialité : L'interface utilisateur de l'application doit être conviviale et intuitive, avec une navigation facile et des

interactions simples pour permettre aux utilisateurs de trouver rapidement les fonctionnalités et d'accomplir leurs tâches sans confusion.

4. Sécurité : L'application doit garantir la sécurité des données des utilisateurs, en utilisant des mécanismes de chiffrement appropriés et en mettant en œuvre des protocoles de sécurité pour protéger les informations sensibles.
5. Maintenabilité : Le code source de l'application doit être bien structuré et documenté, facilitant ainsi la maintenance future et les mises à jour du logiciel.

Conclusion

En conclusion, l'application mobile de bibliothèque audio répond aux besoins des utilisateurs en offrant une interface conviviale pour la recherche, la lecture et la sauvegarde des livres audio. Avec une interface web dédiée pour l'administration et une attention portée à la fiabilité, la convivialité, la sécurité et la maintenabilité, l'application offre une expérience optimale.

Chapitre 3 : Environnement du projet

Introduction

L'environnement du projet se situe dans le domaine des applications mobiles et de la gestion de contenu. Avec une interface mobile intuitive et une interface web dédiée à l'administration, l'application de bibliothèque audio vise à offrir une expérience complète et pratique pour les amateurs de livres audio.

Les technologies utilisées

- **Dart**



Figure 1 Logo Dart

Dart est un langage de programmation moderne et orienté objet développé par Google. Il est conçu pour être à la fois simple à apprendre et puissant, offrant une syntaxe claire et concise. Dart prend en charge la programmation statique ainsi que la programmation dynamique, ce qui en fait un langage polyvalent pour différents types de développement.

Voici quelques caractéristiques clés de Dart :

1. **Typage fort** : Dart est un langage à typage statique, ce qui signifie que les variables et les expressions doivent être typées explicitement. Cela permet de détecter les erreurs

de type au moment de la compilation, ce qui conduit à un code plus sûr et plus robuste.

2. Garbage collector : Dart utilise un ramasse-miettes (Garbage collector) pour gérer automatiquement la mémoire et libérer les objets non utilisés. Cela simplifie la gestion de la mémoire et réduit les risques de fuites de mémoire.
3. Compilation JIT et AOT : Dart peut être compilé en utilisant soit une compilation juste-à-temps (JIT), ce qui permet une exécution rapide du code pendant le développement et les phases de test, soit une compilation anticipée (AOT), ce qui génère un code optimisé pour une exécution encore plus rapide.
4. Asynchronisme : Dart prend en charge les opérations asynchrones avec le concept d'avenirs (futures) et de flux (streams). Cela permet d'effectuer des opérations non bloquantes et d'améliorer les performances des applications qui nécessitent des interactions réseau ou des tâches intensives en calcul.
5. Bibliothèques et écosystème : Dart offre une large gamme de bibliothèques standard qui couvrent divers domaines tels que les opérations de fichier, la manipulation de chaînes, les opérations réseau, etc. De plus, Dart bénéficie d'un écosystème croissant de packages tiers, facilitant le développement d'applications en utilisant des fonctionnalités prêtes à l'emploi.

Dart est principalement utilisé avec le Framework Flutter pour le développement d'applications mobiles multiplateformes. Cependant, il peut également être utilisé pour le développement de serveurs, de scripts et d'applications web, offrant ainsi une polyvalence et une flexibilité aux développeurs.

- **Flutter**



Figure 2 Logo Flutter

Flutter est un Framework open-source développé par Google, utilisé pour créer des applications multiplateformes de haute qualité. Il permet aux développeurs de créer des interfaces utilisateur attrayantes et réactives, avec une seule base de code pour les applications iOS et Android. Grâce à son architecture de widgets personnalisables, Flutter offre une grande flexibilité et permet un développement rapide et efficace d'applications mobiles modernes.

Flutter est un Framework de développement d'applications multiplateformes révolutionnaire. Il offre la possibilité de créer des applications natives à partir d'une seule base de code, ce qui permet de gagner du temps et des ressources. Grâce à sa performance élevée, à sa réactivité et à son large éventail de widgets personnalisables, Flutter permet aux développeurs de créer des interfaces utilisateur modernes et attrayantes. Que ce soit pour iOS, Android ou même le web, Flutter offre une solution puissante et efficace pour le développement d'applications mobiles.

Effectivement, Flutter se base sur le langage de programmation Dart. Dart est un langage moderne, orienté objet et développé également par Google. Il est conçu pour être facile à apprendre, expressif et performant. Dart offre des fonctionnalités avancées telles que la compilation juste-à-temps (JIT) et la compilation anticipée (AOT), ce qui permet à Flutter d'exécuter des applications à grande vitesse. Grâce à Dart, Flutter bénéficie d'une syntaxe claire et concise, facilitant le développement d'applications mobiles réactives et performantes.

- **Firestore**



Figure 3 Logo Firestore

Firestore est une plateforme de développement d'applications mobiles et web proposée par Google. Elle offre un ensemble d'outils et de services cloud pour faciliter le développement, le déploiement et la gestion d'applications. Voici quelques-uns des principaux composants et fonctionnalités de Firestore :

1. Base de données en temps réel : Firestore propose une base de données en temps réel basée sur le modèle NoSQL, permettant de stocker et de synchroniser les données en temps réel entre les clients et le serveur.
2. Authentification : Firestore fournit des services d'authentification sécurisés, permettant aux utilisateurs de s'inscrire et de se connecter à l'application via différentes méthodes d'authentification, telles que l'authentification par e-mail/mot de passe, les comptes Google, Facebook, etc.
3. Stockage de fichiers : Firestore permet de stocker et de récupérer des fichiers tels que des images, des vidéos ou d'autres ressources dans le cloud, offrant ainsi une solution de stockage pratique et évolutive.
4. Hébergement : Firestore propose un service d'hébergement web pour déployer facilement des applications web statiques et dynamiques sans configuration complexe de serveur.
5. Messagerie Cloud : Firestore offre des fonctionnalités de messagerie push pour envoyer des notifications ciblées aux utilisateurs, améliorant ainsi l'engagement et la rétention des utilisateurs.
6. Analytics : Firestore fournit des outils d'analyse pour suivre et mesurer les performances et le comportement des utilisateurs dans l'application, offrant des informations précieuses pour prendre des décisions éclairées sur l'amélioration de l'expérience utilisateur.

7. Test et suivi des erreurs : Firebase propose des outils pour effectuer des tests de qualité, des tests de fonctionnement et le suivi des erreurs, permettant de détecter et de résoudre rapidement les problèmes dans l'application.
8. Machine Learning : Firebase propose des fonctionnalités de machine Learning telles que le kit ML Vision, le kit ML Natural Language, etc., permettant d'intégrer des capacités d'apprentissage automatique dans l'application.

Firebase est largement utilisé dans le développement d'applications mobiles et web, offrant une infrastructure solide et une intégration transparente avec d'autres services Google. Il permet aux développeurs de se concentrer sur la création d'applications sans avoir à se soucier de la gestion de l'infrastructure et des opérations backend.



Figure 4 Logo HTML

HyperText Mark up Langage, généralement abrégé HTML, est le langage de balisage conçu pour représenter les pages web. C'est un langage permettant d'écrire de l'hypertexte, d'où son nom. HTML permet également de structurer sémantiquement et logiquement et de mettre en forme le contenu des pages, d'inclure des ressources multimédias dont des images, des formulaires de saisie et des programmes informatiques. Il permet de créer des documents interopérables avec des équipements très variés de manière conforme aux exigences de l'accessibilité du web. Il est souvent utilisé conjointement avec le langage de programmation JavaScript et des feuilles de style en cascade

(CSS). HTML est inspiré du Standard Generalized Markup Language (SGML). Il s'agit d'un format ouvert.

- **CSS**



Figure 5 Logo CSS

Les feuilles de style en cascade, généralement appelées CSS de l'anglais Cascading Style Sheets, forment un langage informatique qui décrit la présentation des documents HTML et XML. Les standards définissant CSS sont publiés par le World Wide Web Consortium (W3C). Introduit au milieu des années 1990, CSS devient couramment utilisé dans la conception de sites web et bien pris en charge par les navigateurs web dans les années 2000.

- **Javascript**



Figure 6 Logo Javascript

JavaScript est un langage de programmation de scripts principalement employé dans les pages web interactives mais aussi pour les serveurs

avec l'utilisation (par exemple) de Node.js. C'est un langage orienté objet à prototype, c'est-à-dire que les bases du langage et ses principales interfaces sont fournies par des objets qui ne sont pas des instances de classes, mais qui sont chacun équipés de constructeurs permettant de créer leurs propriétés, et notamment une propriété de prototypage qui permet d'en créer des objets héritiers personnalisés. En outre, les fonctions sont des objets de première classe. Le langage supporte le paradigme objet, impératif et fonctionnel. JavaScript est le langage possédant le plus large écosystème grâce à son gestionnaire de dépendances NPM, avec environ 500 000 paquets en août 2017. JavaScript a été créé en 1995 par Brendan Eich. Il a été standardisé sous le nom d'ECMAScript en juin 1997 par Ecma International dans le standard ECMA-262. Le standard ECMA-262 en est actuellement à sa 8e édition. JavaScript n'est depuis qu'une implémentation d'ECMAScript, celle mise en œuvre par la fondation Mozilla. L'implémentation d'ECMAScript par Microsoft (dans Internet Explorer jusqu'à sa version 9) se nomme JScript, tandis que celle d'Adobe Systèmes se nomme Action Script. Avec les technologies HTML et CSS, JavaScript est parfois considéré comme l'une des technologies cœur du World Wide Web. Le langage JavaScript permet des pages web interactives, et à ce titre est une partie essentielle des applications web. Une grande majorité des sites web l'utilisent, et la majorité des disposent d'un moteur JavaScript dédié pour l'interpréter, indépendamment des considérations de sécurité qui peuvent se poser dans le cas échéant.

Chapitre 4 : Conception

Introduction

Dans ce chapitre, nous abordons l'étape cruciale de la conception de notre projet de bibliothèque audio. Nous utilisons l'UML pour modéliser notre système, en mettant l'accent sur les diagrammes de cas d'utilisation et de classe. Ces diagrammes nous permettent de définir les besoins et les fonctionnalités de notre application, en décrivant les interactions entre les utilisateurs et les différents composants. Cette phase de conception nous aide à structurer notre projet et à planifier l'implémentation de manière efficace.

UML



UML (Unified Modeling Language) se définit comme un langage de modélisation graphique et textuel, il est destiné à comprendre et décrire des besoins, spécifier et documenter les systèmes, et sert aussi à esquisser des architectures logicielles, concevoir des solutions et communiquer des points de l'UML unifie à la fois les notations et les concepts orientés objet. Il ne s'agit pas d'une simple notation graphique, car les concepts transmis par un diagramme ont une sémantique précise et sont porteurs de sens au même titre que les mots d'un langage.

Diagramme de classe

Le diagramme de classe est une représentation visuelle des classes, des attributs et des relations entre les objets dans un système. Il permet de modéliser la structure statique du système en identifiant les classes,

leurs attributs et leurs relations, ainsi que les méthodes associées à chaque classe. Ce diagramme permet de comprendre la structure du système et de visualiser les relations entre les différentes entités du système.

Diagramme de classe du sujet

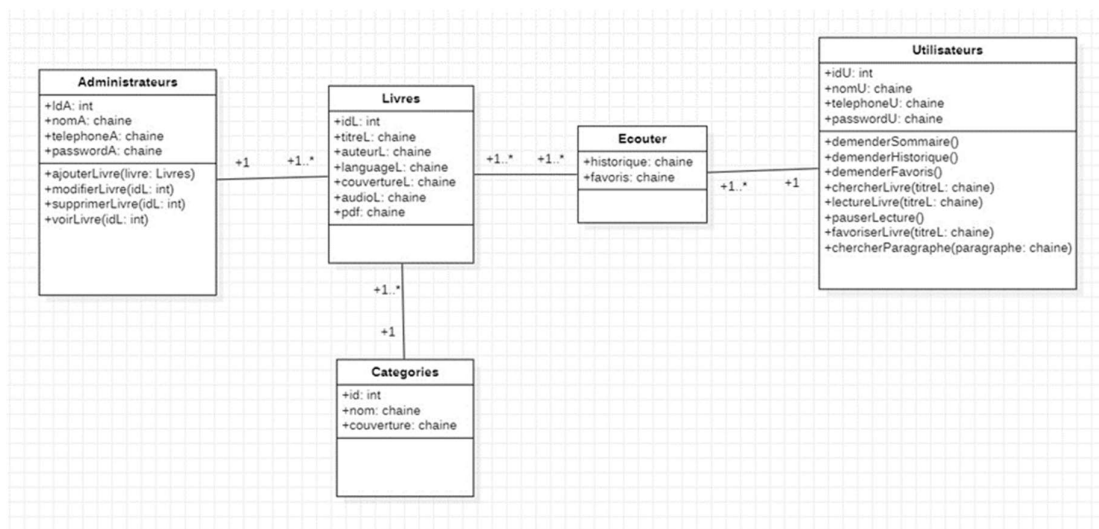


Figure 7 diagramme de class du sujet

Diagramme de cas d'utilisation

Le diagramme de cas d'utilisation représente les interactions entre les acteurs (utilisateurs et administrateurs) et le système. Il montre comment les acteurs utilisent les fonctionnalités du système pour atteindre leurs objectifs. Ce diagramme permet de comprendre les besoins et les fonctionnalités du système du point de vue des utilisateurs. Il identifie les différents cas d'utilisation et les actions

associées à chacun d'eux, permettant ainsi de définir les principales fonctionnalités du système.

Diagramme de cas d'utilisation du sujet

Système

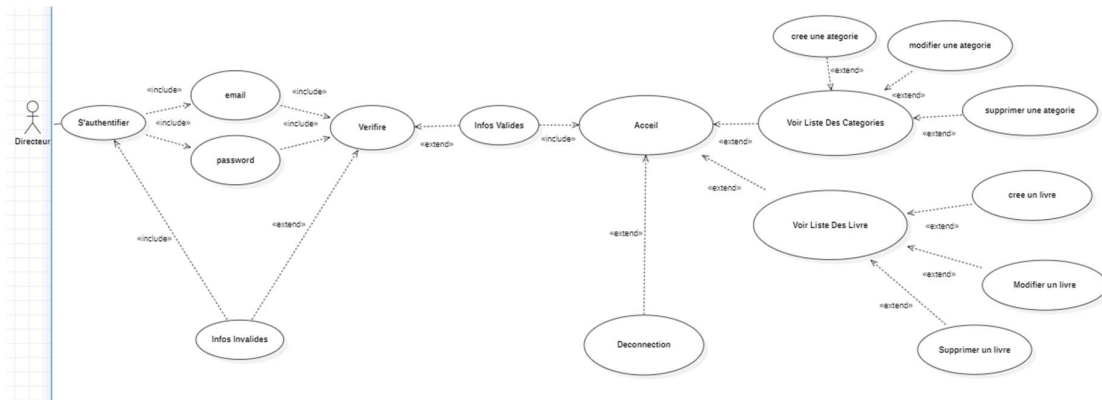


Figure 8 diagramme de cas d'utilisation du sujet (Administrateur)

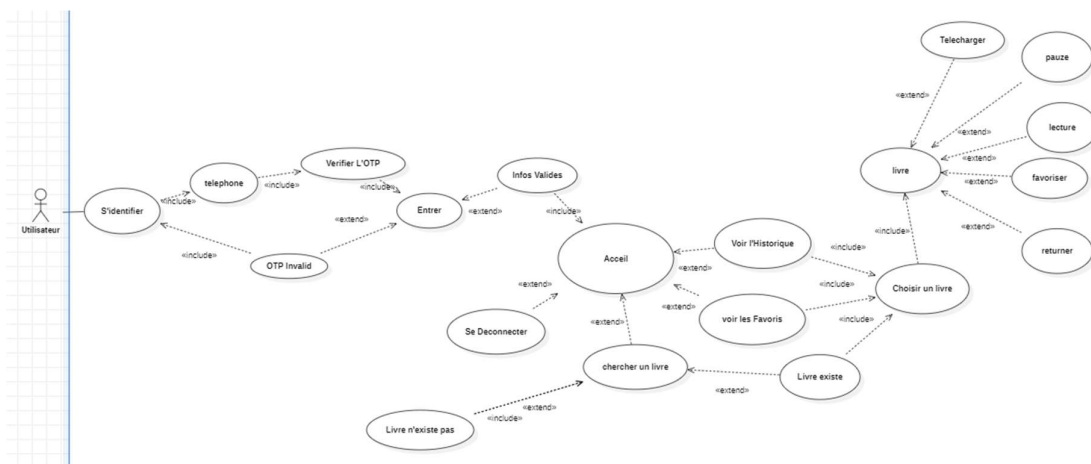


Figure 9 diagramme de cas d'utilisation du sujet (Utilisateur)

Conclusion

Dans cette phase de conception de la bibliothèque, nous avons utilisé une approche méthodique pour modéliser et documenter les différentes composantes de l'application. Les diagrammes de conception créés

fournissent un modèle graphique clair, prêt à être utilisé comme base pour le développement du système.

Chapitre 5 : Réalisation

Chapitre 6 : Conclusion et perspectives

En conclusion, ce projet de développement d'une application mobile de bibliothèque audio a été une expérience enrichissante. Nous avons réussi à concevoir et à mettre en œuvre une solution complète, offrant aux utilisateurs une expérience de lecture audio pratique et flexible.

Grâce à l'utilisation du Framework Flutter et du langage de programmation Dart, nous avons pu créer une application mobile, compatible avec les appareils Android. La puissance de Flutter nous a permis de développer une interface utilisateur attrayante et réactive, offrant une expérience fluide aux utilisateurs.

L'intégration de Firebase avec Flutter a été essentielle pour la gestion des données, l'authentification des utilisateurs et le stockage des fichiers. Firebase nous a fourni une infrastructure robuste et évolutive pour notre application, facilitant la gestion des livres, des catégories et des utilisateurs.

De plus, si nous avions disposé de plus de temps, nous aurions pu réaliser la deuxième partie de notre projet, qui consistait à permettre aux aveugles d'interagir avec l'application. Cela aurait été une étape importante pour rendre notre application encore plus inclusive et accessible à tous.

Nous avons pu appliquer les compétences et connaissances acquises au cours de nos trois années de licence pour mener à bien ce projet. Les concepts de développement logiciel, la programmation orientée objet et les bonnes pratiques de conception ont tous été mis en œuvre pour créer une application de qualité.

En somme, ce projet a été une réussite, offrant une solution complète et conviviale aux amateurs de livres audio. Il démontre l'efficacité et les possibilités offertes par Flutter et Firebase dans le développement d'applications mobiles modernes. Ce projet constitue une base solide pour des améliorations futures et des fonctionnalités supplémentaires, permettant de continuer à répondre aux besoins des utilisateurs dans le domaine de la lecture audio.

Bibliographie

<https://flutter.dev/>

<https://pub.dev/>

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

<https://firebase.google.com/>

<https://getbootstrap.com/docs/4.0/getting-started/download/>

<https://elzero.org/>

<https://www.w3schools.com/>

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