



# ÉCOLE NATIONALE SUPÉRIEURE D'INFORMATIQUE ET D'ANALYSE DES SYSTÈMES - RABAT

RAPPORT DU PROJET DEV MOBILE

---

## TaskMaster

---

**Réalisé par :**  
BAZZAOUI Younes  
ECHAIB Ayoub  
TOUZANI Youssef

**Encadré par :**  
Mr. H.GUERMAH

# Remerciements

Nous tenons à exprimer notre profonde gratitude à toutes les personnes qui ont contribué à la réalisation de ce projet de développement mobile.

Nos remerciements les plus sincères vont particulièrement à Monsieur Hatim Guer-mah, enseignant du module de Développement Mobile, pour son encadrement rigoureux et son accompagnement tout au long du projet. Sa disponibilité, ses conseils techniques avisés et ses retours constructifs ont grandement enrichi notre apprentissage et nous ont permis d'avancer avec méthode et confiance.

Nous souhaitons également saluer la dynamique de collaboration au sein de notre groupe de trois membres. L'engagement collectif, la complémentarité de nos compétences et notre capacité à travailler en synergie ont été des éléments clés dans la réussite de ce projet.

Enfin, nous exprimons notre reconnaissance à nos familles et à nos proches pour leur soutien indéfectible, leur patience et leurs encouragements, qui ont été une source précieuse de motivation tout au long de cette expérience.

# Abstract

This project introduces a smart task management mobile application designed to help users efficiently organize their daily activities. The app enables task creation, editing, deletion, categorization, and supports recurring tasks and status tracking. Leveraging a large language model (LLM), the application analyzes user habits, task history, and schedules to detect interdependencies between tasks and intelligently suggest the most optimal execution order. Integration with Google Calendar allows for conflict management and ensures seamless synchronization with the user's agenda. The backend relies on Firebase for real-time data storage and synchronization. Additionally, smart notifications and secure authentication enhance usability and data protection. The core objective is to boost user productivity by automating priority management and reducing cognitive load through personalized AI-driven task planning.

# Introduction générale

L'application mobile TaskMaster a été conçue pour répondre aux besoins croissants d'organisation personnelle dans un monde où la charge cognitive et la pression du temps ne cessent d'augmenter. En s'inscrivant dans une logique de productivité intelligente, TaskMaster propose une plateforme simple et efficace qui aide les utilisateurs à gérer leurs tâches quotidiennes, à mieux structurer leur emploi du temps et à prendre des décisions plus éclairées quant à la priorisation de leurs activités.

Pensée pour aller au-delà des simples listes de tâches traditionnelles, TaskMaster intègre des fonctionnalités avancées qui permettent non seulement de créer, modifier et suivre l'évolution de ses tâches, mais aussi d'organiser celles-ci selon leur statut, leur catégorie ou leur fréquence. L'utilisateur peut ainsi visualiser de manière claire et structurée l'ensemble de ses obligations à court ou long terme.

Un des aspects clés de l'application est sa capacité à accompagner l'utilisateur dans sa prise de décision. En analysant les habitudes d'utilisation et les priorités déclarées, TaskMaster est capable de proposer des rappels intelligents et des suggestions de tâches à accomplir, contribuant ainsi à réduire la procrastination et à améliorer la gestion du temps. Grâce à un système de notifications dynamiques et personnalisées, l'utilisateur reste informé des tâches urgentes ou oubliées, favorisant un meilleur équilibre entre efficacité et bien-être mental.

L'application se distingue également par son orientation collaborative : elle est pensée pour un usage aussi bien individuel que collectif, dans des contextes professionnels ou académiques. Elle peut être utilisée aussi bien par un étudiant souhaitant structurer ses révisions que par un professionnel gérant plusieurs projets en parallèle.

Avec TaskMaster, l'objectif est clair : offrir une solution intelligente et accessible à tous pour faire face aux défis quotidiens de la gestion du temps et des priorités. Ce projet a permis, par ailleurs, d'explorer des problématiques concrètes de conception logicielle, de structuration de données et d'expérience utilisateur, tout en favorisant un travail collaboratif rigoureux au sein de notre groupe de trois étudiants.

# Table des matières

<b>Remerciements</b>	<b>1</b>
<b>Abstract</b>	<b>2</b>
<b>Introduction générale</b>	<b>3</b>
<b>1 Présentation du cadre général du projet</b>	<b>6</b>
1.1 Introduction . . . . .	6
1.2 Contexte du projet . . . . .	6
1.3 Problématique . . . . .	6
1.4 Solution . . . . .	6
1.5 Objectifs généraux . . . . .	7
1.6 Conclusion . . . . .	7
<b>2 Conception de l'application</b>	<b>8</b>
2.1 Introduction . . . . .	9
2.2 Analyse . . . . .	9
2.2.1 Besoins fonctionnels . . . . .	9
2.2.2 Besoins non fonctionnels . . . . .	9
2.3 Diagrammes fonctionnels (UML) . . . . .	10
2.3.1 Diagramme des cas d'utilisation (Use Case) . . . . .	10
2.3.2 diagramme de Séquence (Sequence diagram) . . . . .	11
2.3.3 diagramme de classe (class diagram) . . . . .	14
2.3.4 Conclusion . . . . .	14
<b>3 Réalisation de l'application</b>	<b>15</b>
3.1 Introduction . . . . .	15
3.2 Technologies utilisées . . . . .	15
3.3 Authentification des utilisateurs . . . . .	15
3.4 Création, modification et priorisation des tâches . . . . .	16
3.5 Interface utilisateur et expérience . . . . .	16
3.6 Statistiques et rapports . . . . .	16
<b>4 Analyse de la rentabilité de l'application</b>	<b>18</b>
4.1 Introduction . . . . .	19
4.2 Coûts estimés . . . . .	19
4.3 Sources de revenus envisagées . . . . .	19
4.3.1 Publicité via Google AdSense . . . . .	19
4.3.2 Abonnement Premium . . . . .	19
4.4 Perspectives de rentabilité . . . . .	20

4.5 Conclusion . . . . . 20

**Conclusion Générale 21**

# Chapitre 1

## Présentation du cadre général du projet

### 1.1 Introduction

Dans un monde où la productivité et la gestion efficace du temps sont devenues cruciales, disposer d'un outil simple, intuitif et efficace pour organiser ses tâches quotidiennes est essentiel. De nombreuses personnes, qu'elles soient étudiantes, professionnelles ou simplement soucieuses de mieux planifier leur journée, recherchent des moyens numériques pour structurer leurs activités. Le projet de To-Do List App naît de cette volonté d'offrir une solution légère et accessible permettant aux utilisateurs de lister, prioriser et suivre l'évolution de leurs tâches à tout moment.

### 1.2 Contexte du projet

Avec la multiplication des responsabilités personnelles, académiques et professionnelles, de nombreuses personnes se tournent vers des outils numériques pour mieux gérer leur temps. Le développement des technologies web et mobiles a démocratisé l'accès à des applications de gestion de tâches, mais beaucoup restent soit trop complexes, soit trop limitées. Ce projet s'inscrit dans un contexte où l'utilisateur recherche une solution à la fois simple, performante et intelligente, capable de s'adapter à ses besoins quotidiens. L'essor des intelligences artificielles, la généralisation des API comme Google Agenda, et la popularité de plateformes comme Firebase créent un environnement favorable au développement d'une application de type To-Do List enrichie, intuitive et connectée.

### 1.3 Problématique

Malgré la diversité des applications existantes, il subsiste un déséquilibre entre simplicité d'usage et richesse fonctionnelle. La plupart des applications de gestion de tâches imposent une courbe d'apprentissage trop élevée, ou à l'inverse, se contentent de fonctionnalités de base, sans réelle valeur ajoutée. De plus, la personnalisation, l'intégration intelligente et la synchronisation multi-appareils sont souvent absentes ou limitées. La problématique centrale de ce projet est donc la suivante : *comment concevoir une application de gestion de tâches à la fois simple, intelligente, personnalisable et connectée, qui réponde aux besoins quotidiens des utilisateurs tout en restant accessible à tous ?*

### 1.4 Solution

Pour répondre à cette problématique, nous proposons le développement d'une application mobile de type To-Do List, intégrant les fonctionnalités classiques de gestion de tâches (création, édition, suppression, priorisation) enrichies par des mécanismes d'intelligence artificielle. L'application permettra une organisation intuitive, une priorisation

intelligente des tâches selon les habitudes de l'utilisateur, et une synchronisation en temps réel grâce à Firebase. De plus, l'intégration avec des outils comme Google Agenda offrira une meilleure visualisation du planning global. L'approche retenue privilégie la simplicité d'usage, tout en apportant des fonctionnalités avancées adaptables selon le profil de l'utilisateur.

### 1.5 Objectifs généraux

L'objectif principal du projet est de concevoir une application mobile conviviale de gestion de tâches qui permet aux utilisateurs d'ajouter, modifier, supprimer et marquer des tâches comme terminées. L'application vise à offrir une interface claire et réactive, adaptée à une utilisation quotidienne, avec une expérience utilisateur fluide. Elle doit également permettre une organisation personnelle efficace des tâches à court ou long terme, tout en restant simple à utiliser, même pour un utilisateur non technique. En complément, l'intégration d'outils d'analyse et de suggestion intelligents permettra d'optimiser la planification et d'améliorer la productivité de l'utilisateur.

### 1.6 Conclusion

Ce chapitre a posé les bases contextuelles du projet. En partant d'un besoin réel de gestion simplifiée des tâches, il a mis en évidence la problématique rencontrée par les utilisateurs face aux solutions existantes. La solution proposée répond à ces enjeux par une application mobile intelligente, accessible et efficace. L'ensemble de ces éléments justifie le développement d'un tel outil et en guide les étapes de conception et de réalisation présentées dans les chapitres suivants.



## Chapitre 2

### Conception de l'application

## 2.1 Introduction

Ce chapitre présente la modélisation de notre application à travers des diagrammes UML et une conception technique adaptée. L'objectif est de structurer les différentes fonctionnalités de manière claire afin de faciliter le développement, l'intégration de l'IA et la maintenance de l'application. Nous mettons en évidence les choix technologiques, les architectures utilisées ainsi que les différentes interactions entre les composants logiciels. Cette approche permet de garantir une application robuste, évolutive et adaptée aux besoins des utilisateurs tout en facilitant la gestion du projet à chaque étape de son cycle de vie.

## 2.2 Analyse

### 2.2.1 Besoins fonctionnels

**Gestion des utilisateurs** L'application doit permettre à un nouvel utilisateur de s'inscrire et à un utilisateur existant de se connecter à son compte. Elle doit également offrir une interface simple pour consulter et modifier les informations de son profil.

**Gestion des tâches** L'utilisateur doit pouvoir créer de nouvelles tâches, les visualiser sous forme de liste, consulter leurs détails, les modifier, les supprimer et les organiser en catégories pour une meilleure clarté.

**Fonctionnalités d'intelligence artificielle** L'application doit intégrer une assistance intelligente qui aide l'utilisateur à mieux gérer ses tâches. Elle doit aussi proposer des suggestions personnalisées et effectuer des analyses automatiques sur les tâches saisies.

**Statistiques et rapports** L'utilisateur doit avoir accès à des statistiques concernant l'utilisation de ses tâches, telles que le nombre de tâches réalisées. L'application doit également proposer un suivi de sa progression et une analyse de ses performances.

**Interface utilisateur** L'application doit proposer une navigation fluide entre les différentes vues. Les tâches doivent être affichées sous forme de liste. La création de tâches doit se faire via une modale de type bottom sheet, et chaque tâche doit avoir une vue de détail dédiée.

### 2.2.2 Besoins non fonctionnels

**Performance** L'application doit répondre rapidement aux actions de l'utilisateur, notamment lors du chargement ou de la manipulation des tâches. L'interface doit être réactive et les requêtes vers Firebase doivent être optimisées.

**Sécurité** L'authentification des utilisateurs doit être sécurisée. Les données personnelles doivent être protégées, et les sessions doivent être correctement gérées pour éviter les accès non autorisés.

**Fiabilité** Les données de l'utilisateur doivent être sauvegardées de manière fiable. L'application doit gérer les erreurs de manière appropriée et permettre une reprise en cas de panne.

**Maintenabilité** Le code de l'application doit être organisé de manière modulaire et documenté, afin de faciliter la compréhension, la correction des bugs et l'ajout futur de nouvelles fonctionnalités.

**Compatibilité** L'application doit fonctionner sur plusieurs versions d'Android. Elle doit aussi s'adapter à différentes tailles d'écran et gérer correctement les changements d'orientation (portrait ou paysage).

**Disponibilité** L'utilisateur doit pouvoir accéder à ses données même en l'absence de connexion Internet. L'application doit assurer une synchronisation automatique dès qu'une connexion est rétablie.

**Évolutivité** L'architecture de l'application doit permettre d'ajouter facilement de nouvelles fonctionnalités et de gérer un nombre croissant d'utilisateurs sans perte de performance.

**Expérience utilisateur** L'interface doit être intuitive, moderne et facile à prendre en main. L'utilisateur doit recevoir des retours visuels lors de ses actions, et l'application doit être accessible et responsive sur tous les types d'appareils.

## 2.3 Diagrammes fonctionnels (UML)

### 2.3.1 Diagramme des cas d'utilisation (Use Case)

Le diagramme suivant illustre les différentes interactions possibles entre les acteurs et le système. Il permet d'avoir une vue d'ensemble des fonctionnalités principales offertes par l'application.

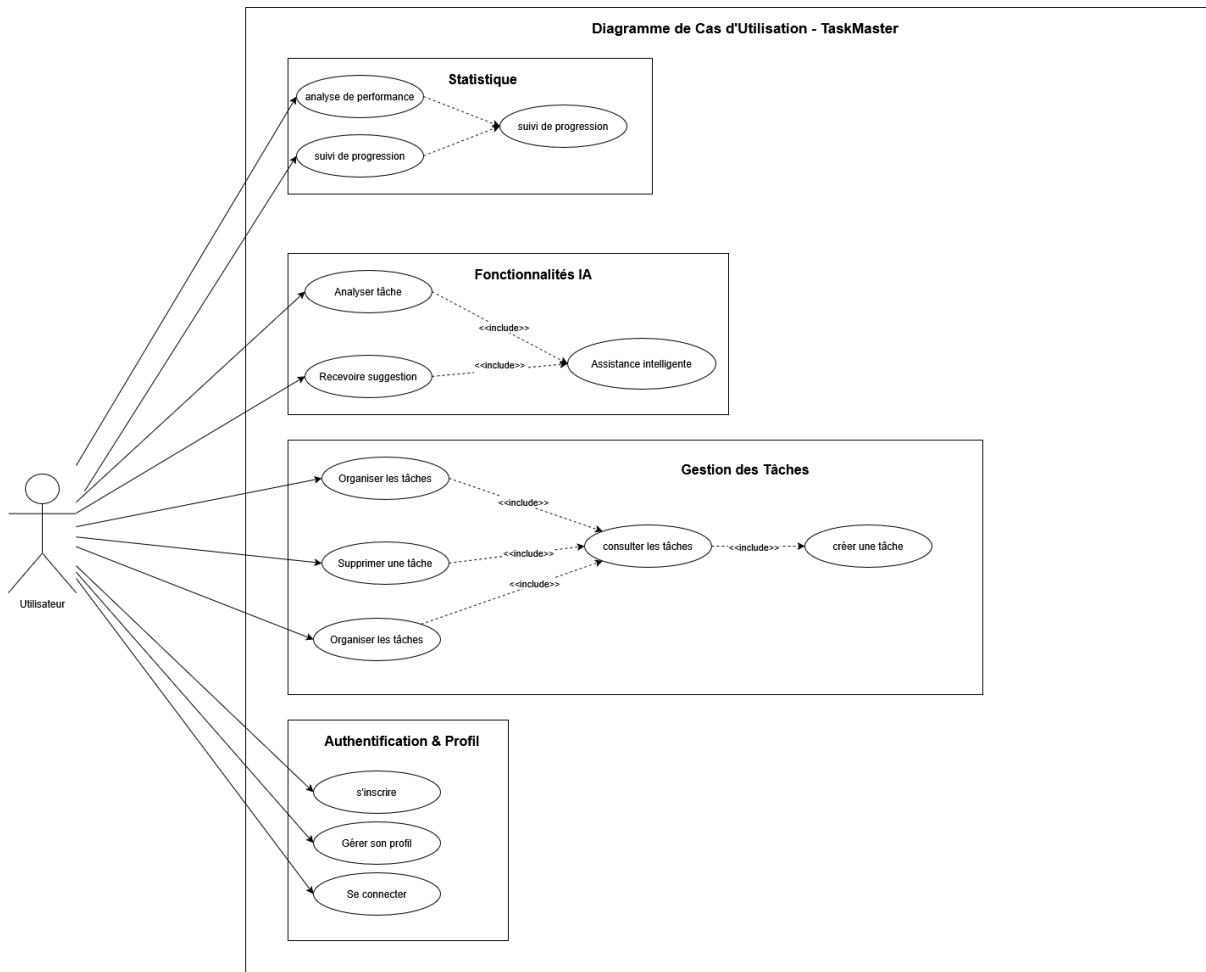


FIGURE 2.1 – Diagramme de cas d'utilisation du système

### 2.3.2 diagramme de Séquence (Sequence diagram)

Les diagrammes de séquence permettent de visualiser les interactions dynamiques entre les différents composants du système au fil du temps. Ils représentent le déroulement chronologique des messages échangés, depuis les actions de l'utilisateur jusqu'aux réponses générées par l'application. Dans cette section, deux diagrammes complémentaires sont présentés : un diagramme global et un diagramme détaillé.

**Diagramme de séquence global** Ce diagramme illustre le fonctionnement général de l'application, depuis le lancement jusqu'à la gestion complète des fonctionnalités. On y retrouve les principales activités de l'utilisateur : authentification, consultation des tâches, création et modification de celles-ci, interaction avec le calendrier, sollicitation des fonctionnalités d'intelligence artificielle, et visualisation des statistiques. Chaque interaction est représentée par une séquence de messages entre les composants : activités Android, base de données Firebase, modules IA, et système de statistiques. Ce schéma met en lumière la coordination entre les différentes couches de l'application.

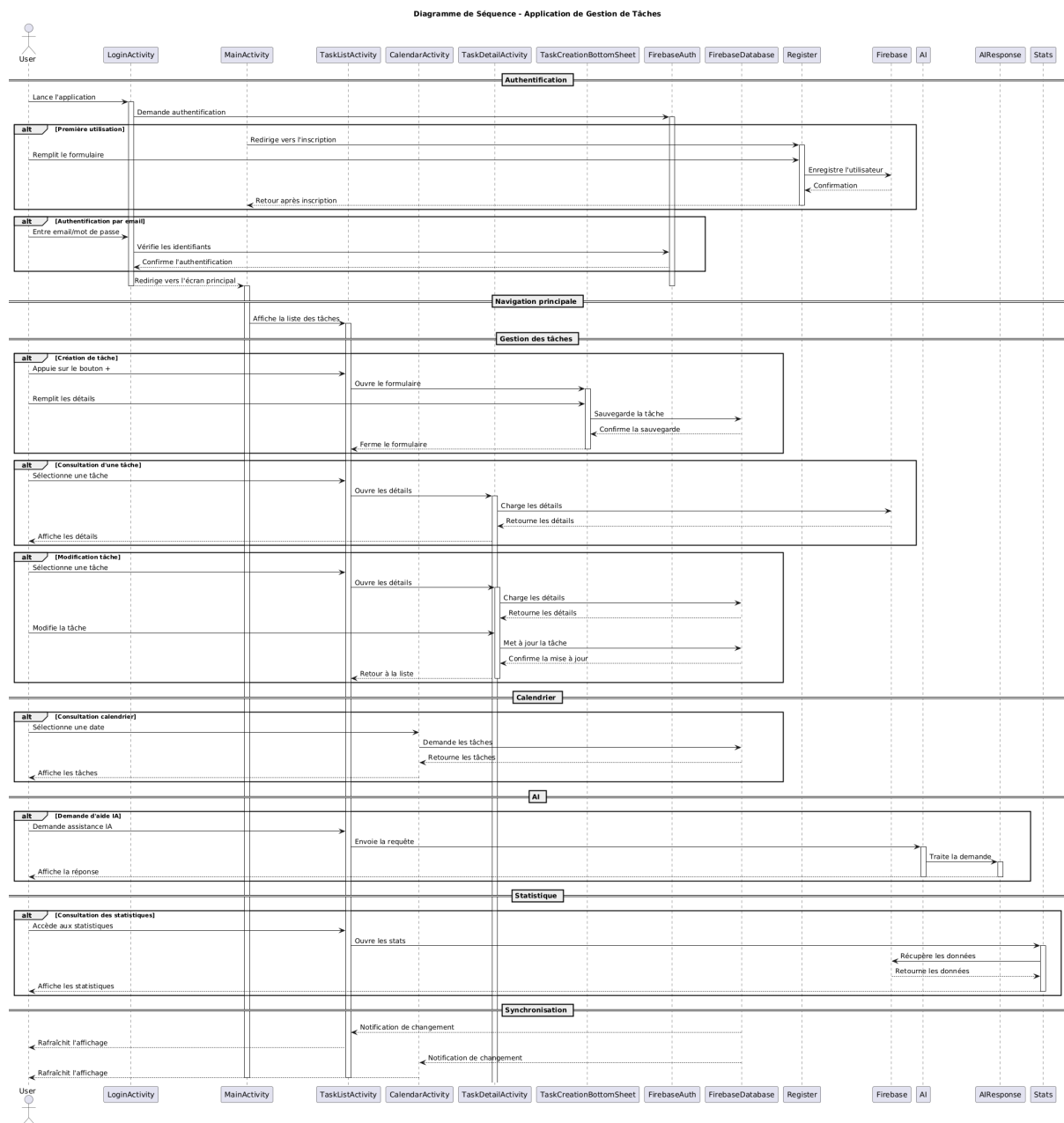
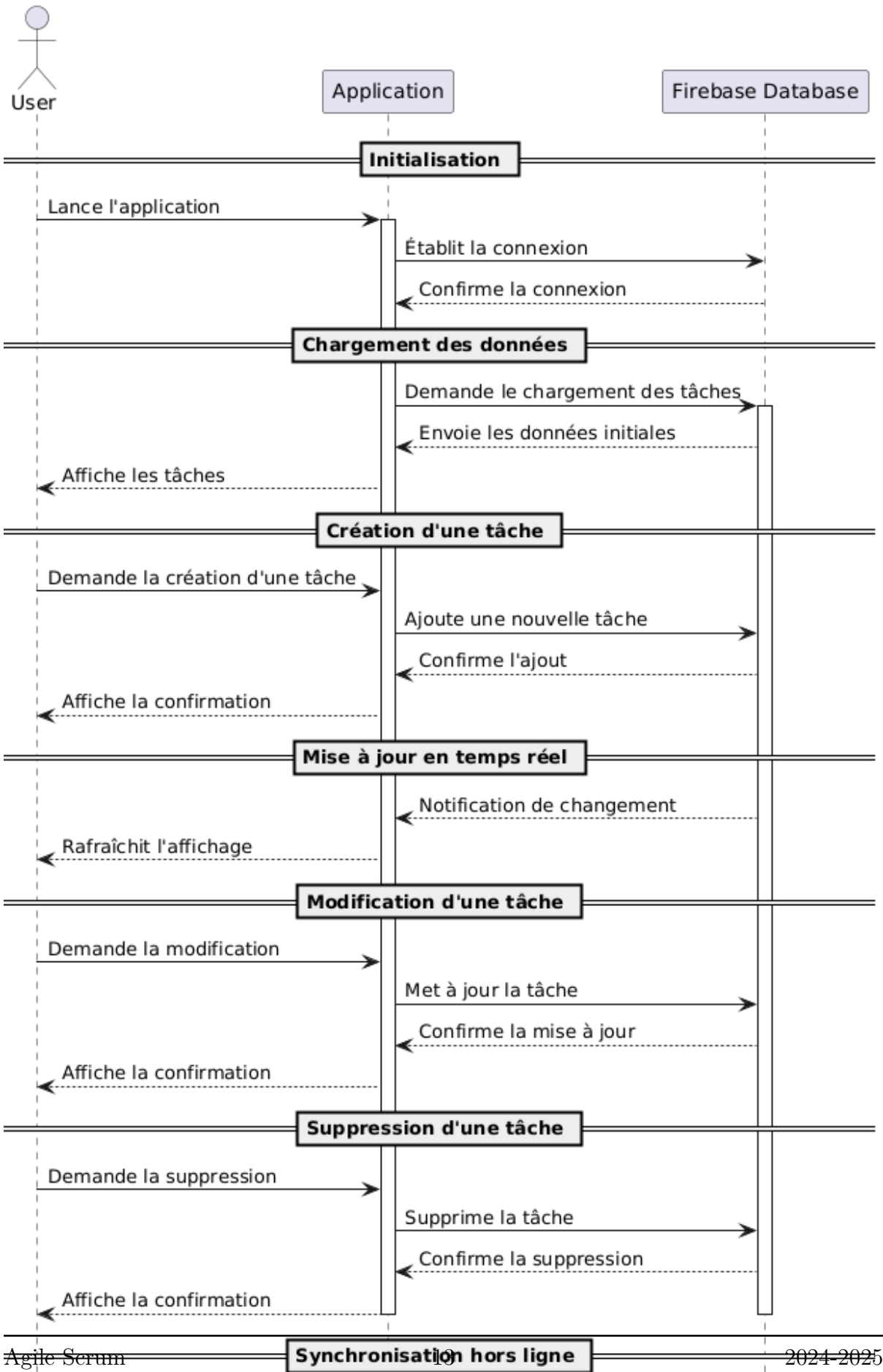


FIGURE 2.2 – sequence diagram

**Diagramme de séquence détaillé** Le second diagramme se concentre sur les interactions spécifiques entre l'application et la base de données Firebase. Il détaille les opérations de chargement des données, de création, de mise à jour, de suppression et de synchronisation en temps réel. Ce diagramme montre également comment l'application gère automatiquement la synchronisation hors ligne grâce à Firebase. Ce niveau de détail permet de mieux comprendre le comportement de l'application dans un environnement connecté ou non, tout en assurant la cohérence des données utilisateur.

Diagramme de Séquence - Interactions avec la Base de Données



### 2.3.3 diagramme de classe (class diagram)

Le diagramme de classe ci-dessous présente la structure interne de l'application. Il décrit les principales classes utilisées, leurs attributs, leurs méthodes, ainsi que les relations entre elles. On y retrouve les activités principales de l'application, les classes liées à la gestion des tâches, à Firebase, à l'interface utilisateur, ainsi qu'aux fonctionnalités d'intelligence artificielle.

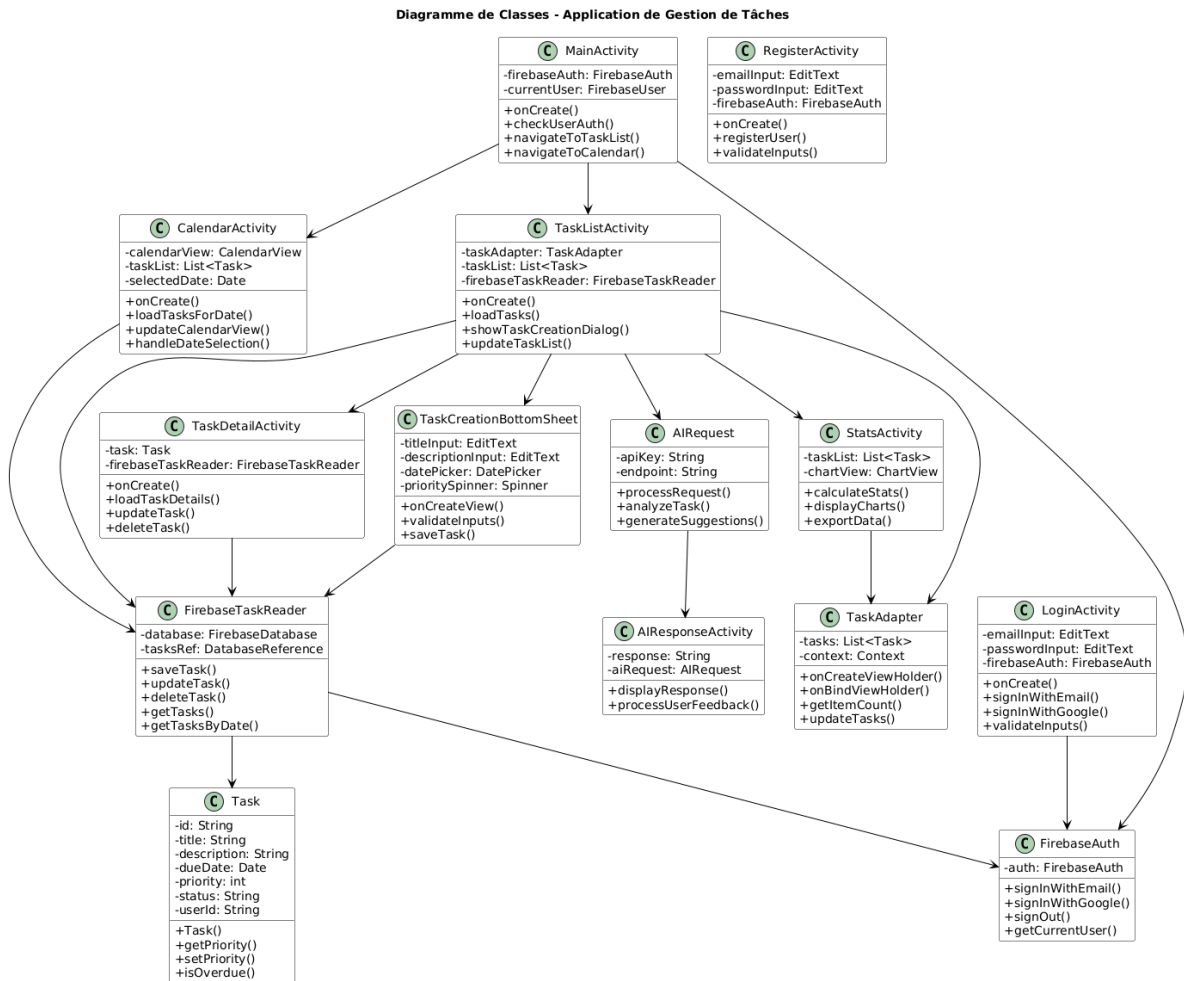


FIGURE 2.4 – class diagram

### 2.3.4 Conclusion

Ce chapitre nous a permis de poser les bases de l'architecture et du fonctionnement interne de notre application. À travers l'analyse des besoins, nous avons identifié les fonctionnalités essentielles à implémenter, tant sur le plan fonctionnel que non fonctionnel. Les diagrammes UML ont apporté une vision structurée des cas d'utilisation et des interactions entre les différents composants du système. Cette étape de conception joue un rôle fondamental dans la réussite du projet, car elle guide le développement en assurant cohérence, évolutivité et maintenabilité. Elle permet également de mieux anticiper les contraintes techniques et de garantir une expérience utilisateur optimale.

# Chapitre 3

## Réalisation de l'application

### 3.1 Introduction

Ce chapitre présente la phase de développement de l'application à partir des éléments définis lors de la conception. Il décrit les outils technologiques utilisés ainsi que les principales fonctionnalités implémentées. Cette étape concrétise les choix techniques et fonctionnels pour aboutir à une application mobile opérationnelle, stable et orientée utilisateur.

### 3.2 Technologies utilisées

Le développement de l'application *Task Master* s'appuie sur les technologies suivantes :

- **Android Studio** : utilisé comme environnement de développement principal, il offre un ensemble d'outils puissants pour construire des applications Android performantes.
- **Java** : langage de programmation retenu pour sa compatibilité avec Android et sa stabilité, permettant d'implémenter les interfaces et la logique métier.
- **Firebase Realtime Database** : cette base de données NoSQL permet un enregistrement et une synchronisation en temps réel des tâches sur tous les appareils connectés.
- **Firebase Authentication** : utilisé pour gérer de manière sécurisée les comptes utilisateurs et l'authentification par email ou via Google.
- **Firebase Cloud** : plateforme assurant une infrastructure évolutive, centralisée et fiable pour le backend de l'application.

### 3.3 Authentification des utilisateurs

L'application propose deux méthodes d'authentification : par email et mot de passe, ou via un compte Google. L'authentification classique permet aux utilisateurs de créer un compte en renseignant leur adresse email, validée via un lien de confirmation, puis de se connecter avec un mot de passe sécurisé. En cas d'oubli, une procédure de récupération est disponible. La session est conservée de manière sécurisée tant que l'utilisateur ne choisit pas de se déconnecter.

Parallèlement, l'intégration de l'API Google Sign-In facilite une connexion rapide et sécurisée à l'application. Les tokens d'authentification sont gérés automatiquement et les données utilisateur sont synchronisées avec Firebase. Des pistes d'amélioration ont été identifiées, notamment l'ajout d'une authentification à deux facteurs, une meilleure gestion des erreurs, et la déconnexion automatique après inactivité prolongée.



## 3.4 Création, modification et priorisation des tâches

La gestion des tâches constitue le cœur de l'application. La création de tâches s'effectue via une interface intuitive sous forme de BottomSheet, permettant à l'utilisateur de saisir un titre, une description, une date d'échéance, une catégorie et un niveau de priorité. Les champs sont vérifiés avant validation, et les données sont stockées en temps réel dans Firebase.

Chaque tâche peut être consultée et modifiée depuis une vue détaillée. L'utilisateur peut modifier ses propriétés à tout moment, avec une mise à jour immédiate dans la base de données. L'application prend également en charge l'historique des modifications et la gestion des conflits de synchronisation.

Une fonctionnalité de priorisation intelligente est intégrée. Elle repose sur plusieurs critères : la date limite, l'importance déclarée par l'utilisateur, l'historique des tâches réalisées, ainsi que des suggestions générées automatiquement. Le système s'adapte progressivement au comportement de l'utilisateur pour lui proposer une organisation personnalisée.

## 3.5 Interface utilisateur et expérience

L'interface de Task Master est conçue selon les principes du Material Design, assurant une navigation fluide et cohérente entre les écrans. Les transitions sont animées, les éléments bien hiérarchisés, et l'utilisateur peut choisir entre un thème clair ou sombre selon ses préférences. L'expérience utilisateur est au cœur de la conception, avec une ergonomie optimisée pour les interactions mobiles.

En complément, des fonctionnalités d'intelligence artificielle viennent enrichir l'expérience. L'IA analyse les habitudes de l'utilisateur pour ajuster la priorisation des tâches, proposer des suggestions, et même anticiper des dates d'échéance potentielles. Ces apports contribuent à une gestion plus efficace du temps et des priorités.

## 3.6 Statistiques et rapports

Un tableau de bord intégré permet à l'utilisateur de suivre sa progression dans le temps. Il affiche des graphiques de performance, des indicateurs de productivité, ainsi que l'historique des tâches accomplies. Ces éléments offrent une vision claire de l'activité et motivent l'utilisateur à atteindre ses objectifs.

Par ailleurs, l'application intègre une fonction d'analyse de données. Elle permet de détecter les périodes de haute ou faible productivité, d'identifier les tâches fréquemment reportées, et de générer des rapports personnalisés. Ces rapports peuvent être exportés pour une utilisation externe ou à des fins de suivi personnel.

## Conclusion

La phase de réalisation a permis de transformer la conception théorique de l'application en une solution concrète et fonctionnelle. Grâce à un ensemble d'outils modernes et une architecture bien pensée, les principales fonctionnalités ont été mises en œuvre dans le respect des besoins initiaux. L'intégration de Firebase et l'ajout de modules intelligents

apportent à l'utilisateur une expérience fluide, personnalisée et efficace. Cette base solide ouvre la voie à des évolutions futures du projet.

## Chapitre 4

### Analyse de la rentabilité de l'application

## 4.1 Introduction

La question de la rentabilité est essentielle pour assurer la viabilité et la pérennité d'une application mobile. Dans le cas de notre projet, l'investissement initial est très limité puisque le développement a été entièrement réalisé par notre propre équipe, sans recours à des prestataires externes ni à des services payants. De plus, les outils et services utilisés, tels que Firebase et les bibliothèques Android, sont disponibles gratuitement dans leurs versions de base.

Malgré ces faibles coûts, l'application offre un potentiel réel de génération de revenus grâce à des modèles économiques simples et efficaces. Deux sources principales de monétisation ont été envisagées : l'intégration de la publicité et la mise en place d'un système de comptes Premium.

## 4.2 Coûts estimés

Le coût de développement est considéré comme quasi nul, car :

- Le développement a été effectué en interne, sans coût salarial.
- Aucun outil ou API payant n'a été utilisé.
- Firebase est utilisé dans sa version gratuite, suffisante pour un volume moyen d'utilisateurs.

Les seuls coûts à prévoir à l'avenir pourraient concerner :

- L'hébergement si la base d'utilisateurs devient très importante.
- L'achat d'un nom de domaine et la publication sur le Play Store.

## 4.3 Sources de revenus envisagées

### 4.3.1 Publicité via Google AdSense

L'application peut intégrer de la publicité discrète via Google AdSense (ou AdMob pour Android), par exemple dans la vue des tâches ou en bas de l'écran d'accueil. Cela permettrait de générer des revenus passifs proportionnels au trafic utilisateur, sans impacter l'expérience.

**Avantages :**

- Facile à intégrer.
- Aucun frais initial.
- Source de revenus dès les premiers utilisateurs.

### 4.3.2 Abonnement Premium

Un second modèle de monétisation consiste à proposer deux types de comptes : **utilisateur standard** et **utilisateur Premium**. L'accès Premium pourrait être proposé sous forme d'abonnement mensuel ou annuel.

**Fonctionnalités réservées aux utilisateurs Premium :**

- Accès complet aux fonctionnalités d'intelligence artificielle (priorisation avancée, suggestions, analyse personnalisée).
- Statistiques détaillées de productivité.
- Synchronisation multi-appareils étendue.

- Suppression des publicités.
- Mode hors ligne amélioré.

### 4.4 Perspectives de rentabilité

Compte tenu du faible coût initial et du potentiel de monétisation, l'application peut atteindre rapidement un seuil de rentabilité si elle attire une base d'utilisateurs suffisante. L'ajout progressif de nouvelles fonctionnalités Premium et l'amélioration de l'expérience utilisateur contribueront à fidéliser les utilisateurs et à convertir certains d'entre eux en clients payants.

### 4.5 Conclusion

L'application présente un modèle économique simple, basé sur des coûts de développement très faibles et des sources de revenus accessibles. Grâce à une combinaison entre publicité et abonnement Premium, elle peut générer des revenus tout en maintenant une version gratuite fonctionnelle pour les utilisateurs. Ce modèle assure à la fois l'accessibilité du service et la viabilité financière du projet à moyen et long terme.

# Conclusion Générale

Ce projet de développement d'une application de gestion intelligente des tâches a permis de mettre en pratique l'ensemble des compétences acquises au cours de notre formation, tant sur le plan technique que méthodologique. De la phase d'analyse des besoins à la conception UML, en passant par le développement et la mise en œuvre des fonctionnalités, chaque étape a été abordée de manière structurée et rigoureuse.

L'application développée se distingue par sa simplicité d'utilisation, son interface fluide, et l'intégration de fonctionnalités avancées comme la priorisation intelligente via un module d'IA, la synchronisation en temps réel avec Firebase, ou encore l'affichage de statistiques personnalisées. Elle répond ainsi à un besoin concret : celui de mieux organiser ses tâches quotidiennes dans un environnement mobile, accessible et intelligent.

Sur le plan technique, le projet a été réalisé en utilisant des outils modernes et gratuits, permettant un excellent rapport qualité/coût. Cette approche démontre qu'il est possible de créer une application complète, stable et évolutive sans investissements lourds.

Enfin, une réflexion sur la rentabilité et les perspectives d'évolution a été menée, ouvrant la voie à une éventuelle commercialisation via des modèles freemium et des publicités intégrées. Ce travail constitue une base solide pour de futures améliorations ou extensions fonctionnelles.

Nous pouvons conclure que les objectifs fixés au départ ont été atteints, et que ce projet représente une expérience riche et valorisante sur les plans personnel, académique et professionnel.