



## Rapport – TaskMaster

Projet et rapport réalisé par :

**FARROUJ SALMA**

**ELBAROUDI DOUAE**

**TACHFINE HIBA**

**SALMA SOUIT**

Encadré par:

Mme Zineb

Filière :

4éme année IIR : ingénierie informatique et réseaux

Année universitaire 2025/2026

## Résumé du projet – TaskMaster

Le projet TaskMaster consiste en le développement d'une application mobile de gestion des tâches et des rappels intelligente. Conçue avec React Native et Expo, l'application fonctionne sur Android, iOS et Web, offrant une expérience utilisateur moderne et fluide. Contrairement aux solutions traditionnelles, TaskMaster agit comme un véritable compagnon de productivité qui organise vos activités par catégories et priorités, envoie des notifications pour vos rappels, et propose des suggestions intelligentes d'activités adaptées à votre emploi du temps.

L'application repose sur Firebase pour le backend et la persistance des données, assurant ainsi une synchronisation en temps réel et une authentification sécurisée. Elle intègre également un système de notifications locales via expo-notifications, une navigation optimisée avec React Navigation, et un système de thèmes dynamiques permettant de basculer entre les modes Clair et Sombre.

TaskMaster adopte une méthodologie agile simplifiée, favorisant un développement itératif et réactif. L'application répond ainsi aux besoins croissants d'organisation personnelle et de productivité en proposant une solution complète, évolutive et centrée sur l'utilisateur, accessible depuis n'importe quel appareil mobile.

# Contents

Chapitre 1 : Contexte général du projet.....	6
1.1 Introduction.....	6
1.2 Contexte du projet .....	6
1.3 Périmètre du projet.....	6
1.4 Problématique .....	6
1.5 Objectifs du projet.....	7
1.5.1 Cahier des charges .....	7
1.5.2 Étude de l'existant.....	7
1.5.3 Critique de l'existant .....	8
1.6 Benchmark .....	8
1.7 Méthodologies de gestion de projet .....	8
1.8 Diagramme de Gantt.....	9
1.9 Conclusion .....	9
Chapitre 2 : Analyse et conception.....	10
2.1 Introduction.....	10
2.2 Besoins fonctionnels.....	10
2.3 Besoins non fonctionnels.....	10
2.4 Diagramme de cas d'utilisation .....	11
2.5 Diagrammes de séquence .....	11
2.6 Diagramme de classes .....	12
2.7 Conclusion .....	12
Chapitre 3 : Réalisation.....	14
3.1 Introduction.....	14
3.2 Architecture de la solution .....	14
3.3 Outils et technologies utilisés.....	14
3.4 Interfaces de l'application .....	14
<b>3.5 Fonctionnalités principales implémentées .....</b>	<b>17</b>
3.6 Sécurité et gestion des accès .....	17
3.7 Tests et validation.....	18
3.8 Problèmes rencontrés et solutions apportées .....	18
3.9 Conclusion .....	18



## ***Remerciement***

Je tiens à exprimer ma sincère gratitude à toutes les personnes qui m'ont soutenue et aidée dans la réalisation de ce mémoire.Tout d'abord, un immense merci à mon encadrante, Mme Zineb , pour ses conseils précieux, son accompagnement et sa disponibilité.Un grand merci à mes amis et camarades de promotion pour leur aide, leur motivation et tous ces moments de partage qui ont rendu cette expérience plus agréable.Enfin, je me remercie moi- même pour avoir tenu bon jusqu'au bout, malgré le stress et les nuits blanches.

# Introduction Générale

À l'ère du numérique et de la mobilité, la gestion efficace du temps et des tâches quotidiennes constitue un enjeu majeur tant pour les étudiants que pour les professionnels. Face à la multiplication des responsabilités et à la dispersion des outils disponibles — qu'ils soient traditionnels (agendas papiers) ou numériques (applications souvent complexes, dépendantes d'une connexion internet ou non adaptées aux usages mobiles) —, le besoin d'une solution centralisée, intuitive et accessible partout et à tout moment se fait de plus en plus sentir.

C'est dans ce contexte que s'inscrit le projet **TaskMaster Mobile**, une application multiplateforme dédiée à la gestion intelligente des tâches et des rappels. Développée dans le cadre d'un projet de fin d'études en ingénierie informatique et réseaux, cette solution vise à offrir aux utilisateurs un environnement unique et moderne pour organiser leurs activités, programmer des alertes, prioriser leurs objectifs et bénéficier de suggestions adaptatives directement depuis leur smartphone.

Conçue avec une architecture mobile-first exploitant **React Native** pour l'interface et **Firebase** pour le backend et la persistance des données, TaskMaster Mobile se distingue par son ergonomie tactile, sa portabilité multiplateforme (Android, iOS, Web) et son fonctionnement intelligent. Elle s'adresse aussi bien aux novices qu'aux utilisateurs avertis, en proposant une expérience fluide, personnalisable et sécurisée, avec synchronisation en temps réel et notifications locales.

Ce rapport détaille les différentes étapes du projet, depuis l'analyse des besoins spécifiques au mobile et la conception technique jusqu'à la réalisation et aux tests multiplateformes. Il met en lumière les choix technologiques modernes, les défis relevés liés au développement cross-platform et les résultats obtenus, illustrant ainsi comment une approche structurée, agile et centrée sur l'expérience utilisateur mobile peut mener au développement d'un outil à la fois simple, performant et utile au quotidien, où que l'on soit.

# Chapitre 1 : Contexte général du projet

## 1.1 Introduction:

Le projet **TaskMaster** consiste en la création d'une **application mobile multiplateforme** destinée à la gestion intelligente des tâches quotidiennes et des rappels. Cette application vise à aider les utilisateurs à organiser leurs activités, planifier leurs journées, définir des rappels et recevoir des suggestions intelligentes pour améliorer leur productivité, directement depuis leur smartphone.

## 1.2 Contexte du projet

De nombreuses personnes, qu'elles soient étudiants ou professionnels, rencontrent des difficultés à gérer efficacement leurs tâches et leur temps depuis leurs appareils mobiles. Les solutions existantes sont souvent trop complexes, dépendantes d'une connexion internet constante, ou manquent de fonctionnalités adaptatives.

Le projet TaskMaster répond à ce besoin en proposant une application mobile simple, intuitive et sécurisée, permettant aux utilisateurs de centraliser la gestion de leurs tâches et rappels avec une expérience moderne et responsive.

## 1.3 Périmètre du projet

Le projet TaskMaster couvre l'ensemble des besoins liés à la gestion mobile des tâches quotidiennes et des rappels :

- **Création et gestion des comptes utilisateurs** via Firebase Authentication
- **Ajout, modification et suppression des tâches** avec catégorisation et priorités
- **Programmation et gestion des rappels** avec notifications locales
- **Suggestions intelligentes** d'activités selon le temps libre (jours vides ou week-ends)
- **Interface adaptive** avec mode Clair/Sombre
- **Tableau de bord synthétique** avec statistiques en temps réel
- **Synchronisation des données** via Firebase Firestore

## 1.4 Problématique

Les utilisateurs rencontrent souvent des difficultés à organiser leur journée et à ne pas oublier des tâches importantes depuis leurs appareils mobiles. Les solutions existantes sont parfois trop complexes, trop dépendantes d'internet, ou ne s'adaptent pas à tous les types d'utilisateurs et à leurs préférences visuelles.

La problématique centrale de ce projet est donc :

**Comment créer une application mobile multiplateforme, performante et intuitive qui permette aux utilisateurs de gérer efficacement leurs tâches quotidiennes depuis leur smartphone, d'éviter les oubliés grâce à des notifications intelligentes, et d'améliorer leur productivité avec des suggestions adaptatives, le tout avec une interface moderne et personnalisable ?**

## 1.5 Objectifs du projet

Le projet TaskMaster poursuit plusieurs objectifs essentiels :

- **Développer une application mobile multiplateforme** avec React Native et Expo
- **Faciliter l'organisation et le suivi des tâches** depuis n'importe quel appareil
- **Réduire les oubliés** grâce aux notifications locales programmées
- **Encourager la productivité** via des suggestions intelligentes basées sur le temps libre
- **Proposer une interface moderne, fluide et personnalisable** (mode Clair/Sombre)
- **Garantir la sécurité et la fiabilité** des données avec Firebase
- **Créer une application structurée, maintenable et facilement extensible**

### 1.5.1 Cahier des charges

- Le cahier des charges se divise en **besoins fonctionnels** et **besoins non fonctionnels** :
  - **Besoins fonctionnels** :
    - Authentification sécurisée des utilisateurs (création de compte, connexion)
    - Gestion complète des tâches (CRUD : création, lecture, mise à jour, suppression)
    - Catégorisation et priorisation des tâches
    - Gestion des rappels avec notifications locales
    - Suggestions intelligentes d'activités selon les créneaux libres
    - Tableau de bord avec statistiques personnelles
    - Interface adaptable (mode Clair/Sombre)
  - **Besoins non fonctionnels** :
    - **Performance** : Temps de réponse rapide et expérience fluide
    - **Sécurité** : Protection des données utilisateur avec Firebase

- **Compatibilité** : Application fonctionnelle sur Android, iOS et Web
- **UX/UI** : Interface intuitive, moderne et responsive
- **Maintenabilité** : Code structuré, modulaire et bien documenté
- **Accessibilité** : Support des différentes tailles d'écran et préférences utilisateur

### 1.5.2 Étude de l'existant

Aujourd'hui, plusieurs applications mobiles de gestion des tâches existent, telles que **Todoist**, **Microsoft To Do**, **Google Tasks**, ou [Any.do](#). Ces solutions permettent la gestion de tâches et de rappels mais présentent souvent des limitations : interfaces parfois surchargées, fonctionnalités payantes, ou dépendance excessive à une connexion internet pour une utilisation optimale

### 1.5.3 Critique de l'existant

L'analyse des solutions existantes montre plusieurs limites spécifiques au mobile :

- **Dépendance à une connexion Internet** constante pour certaines fonctionnalités
- **Interfaces parfois complexes** ou peu adaptées aux petites écrans
- **Absence de suggestions intelligentes** véritablement adaptatives
- **Manque de personnalisation** visuelle (thèmes fixes)
- **Expérience utilisateur inégale** entre les différentes plateformes
- **Fonctionnalités avancées souvent payantes**

Ces lacunes justifient le développement d'une application mobile moderne, légère, personnalisable et centrée sur l'expérience utilisateur multiplateforme.

## 1.6 Benchmark

Un benchmark des solutions mobiles existantes a été réalisé :

- **Todoist Mobile** : Interface riche et fonctionnalités avancées, mais version gratuite limitée
- **Microsoft To Do Mobile** : Intégration avec l'écosystème Microsoft, design épuré
- **Google Tasks** : Simplicité extrême mais fonctionnalités très basiques
- **[Any.do](#)** : Design moderne mais nombreuses fonctionnalités premium

Ce benchmark confirme l'opportunité de créer une application mobile qui combine simplicité d'utilisation, personnalisation visuelle, suggestions intelligentes et gratuité des fonctionnalités essentielles

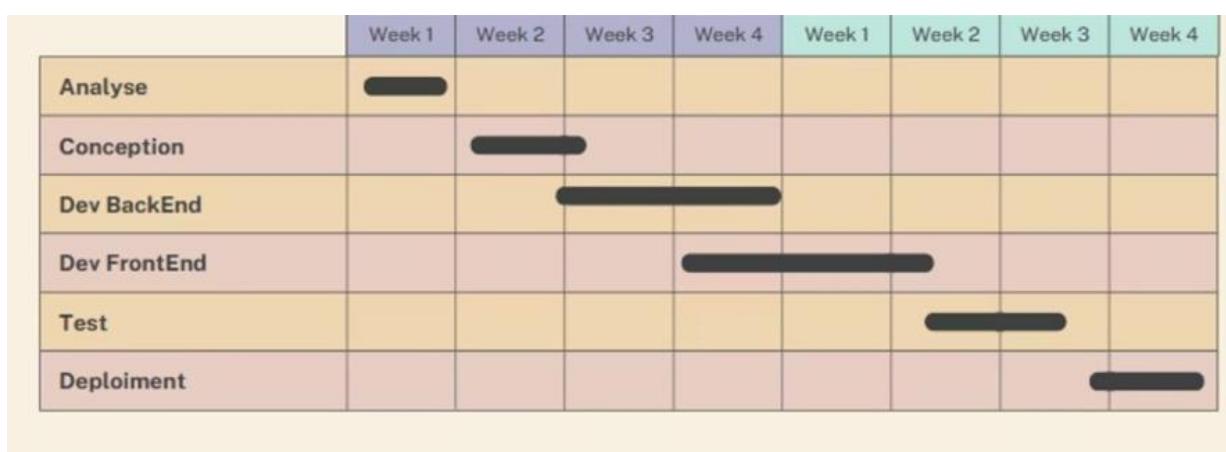
## 1.7 Méthodologies de gestion de projet

Pour le développement de TaskMaster, une méthodologie Agile simplifiée a été adoptée, adaptée au développement mobile :

- **Sprints hebdomadaires** avec objectifs précis
- **Analyse et spécification** des besoins mobiles spécifiques
- **Conception d'interface mobile** avec maquettes responsive
- **Développement multiplateforme** avec React Native
- **Tests sur différents appareils** (Android, iOS, émulateurs)
- **Intégration continue** avec Expo et Firebase
- **Retours utilisateurs itératifs** sur l'expérience mobile

Des réunions régulières permettent de suivre l'avancement, d'ajuster les priorités et de valider les fonctionnalités spécifiques au mobile.

## 1.8 Diagramme de Gantt



## 1.9 Conclusion

Le projet **TaskMaster** s'inscrit dans une démarche de modernisation de la gestion quotidienne des tâches sur mobile, en offrant une application multiplateforme intuitive, personnalisable et intelligente. Après avoir identifié les besoins spécifiques au mobile, analysé les solutions existantes et défini des objectifs précis, nous avons pu cerner les enjeux techniques et ergonomiques propres au développement mobile. L'approche agile adoptée permettra d'assurer une réalisation progressive et adaptée aux contraintes des différentes plateformes.

## Chapitre 2 : Analyse et conception

- **Introduction**

L'étape d'analyse et de conception constitue la base technique du projet **TaskMaster**. Elle permet de traduire les besoins exprimés dans le cahier des charges en modèles fonctionnels et techniques adaptés au développement mobile avec React Native et Firebase. Ce chapitre présente les besoins identifiés, les choix de conception, les diagrammes d'analyse, ainsi que les décisions architecturales indispensables pour assurer une application mobile cohérente, performante et évolutive

- **Besoins fonctionnels**

Les besoins fonctionnels de l'application mobile TaskMaster se concentrent sur les principales fonctionnalités nécessaires pour une gestion efficace et intuitive des tâches depuis un smartphone :

- **Gestion des utilisateurs** : Création de compte via Firebase Authentication, authentification sécurisée, et gestion du profil personnel avec accès à l'historique des tâches et rappels
- **Gestion des tâches** : Création, modification et suppression de tâches avec catégorisation (travail, personnel, courses, etc.), définition des priorités (haute, moyenne, basse) et indication du statut (à faire, en cours, terminé)
- **Gestion des rappels** : Programmation de notifications locales avec choix de date, heure et récurrence, accessible même hors ligne
- **Suggestions intelligentes** : Proposition automatique d'activités selon les créneaux libres détectés dans l'emploi du temps, avec catégories personnalisables (loisirs, sport, tâches ménagères, etc.)
- **Tableau de bord mobile** : Vue synthétique optimisée pour mobile présentant les tâches du jour, les rappels à venir et les statistiques de productivité
- **Personnalisation** : Basculer entre les modes Clair et Sombre, ajuster les préférences de notifications, personnaliser les catégories

- **Besoins non fonctionnels**

Les besoins non fonctionnels de TaskMaster visent à garantir la qualité, la performance et la sécurité de l'application mobile sur le long terme :

**Performance mobile** : Temps de chargement rapide (< 2 secondes), animations fluides, consommation batterie optimisée

**Compatibilité multiplateforme** : Fonctionnement identique sur Android (8+), iOS (12+) et navigateurs web modernes

**Sécurité** : Authentification robuste via Firebase, chiffrement des données, protection contre les injections

**Expérience utilisateur mobile** : Interface tactile intuitive, navigation gestuelle, design responsive adapté aux différentes tailles d'écran

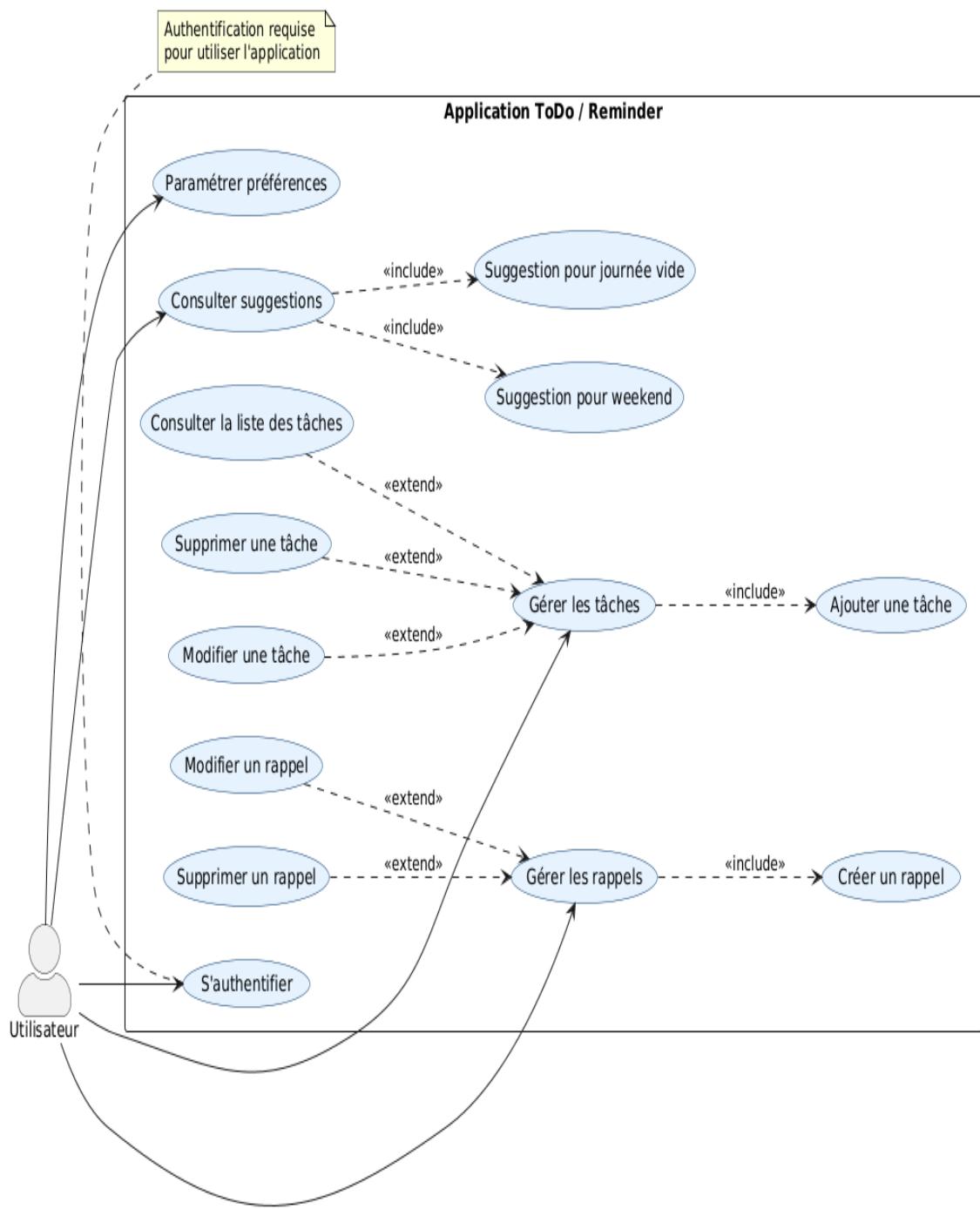
**Mode hors ligne** : Fonctionnalités essentielles disponibles sans connexion internet

**Maintenabilité** : Architecture modulaire avec séparation claire des couches, code bien documenté

**Accessibilité** : Support du VoiceOver/TalkBack, contrastes adaptés, tailles de texte ajustables

- **Diagramme de cas d'utilisation**

Ce diagramme de cas d'utilisation illustre les principales fonctionnalités de l'application **ToDo / Reminder** ainsi que les interactions possibles avec l'utilisateur. Il met en évidence l'authentification obligatoire, la gestion des tâches et des rappels, la consultation de la liste des tâches, le paramétrage des préférences et les suggestions automatiques. Les relations « include » et « extend » permettent de structurer les fonctionnalités principales et optionnelles, offrant une vision globale et cohérente du fonctionnement de l'application



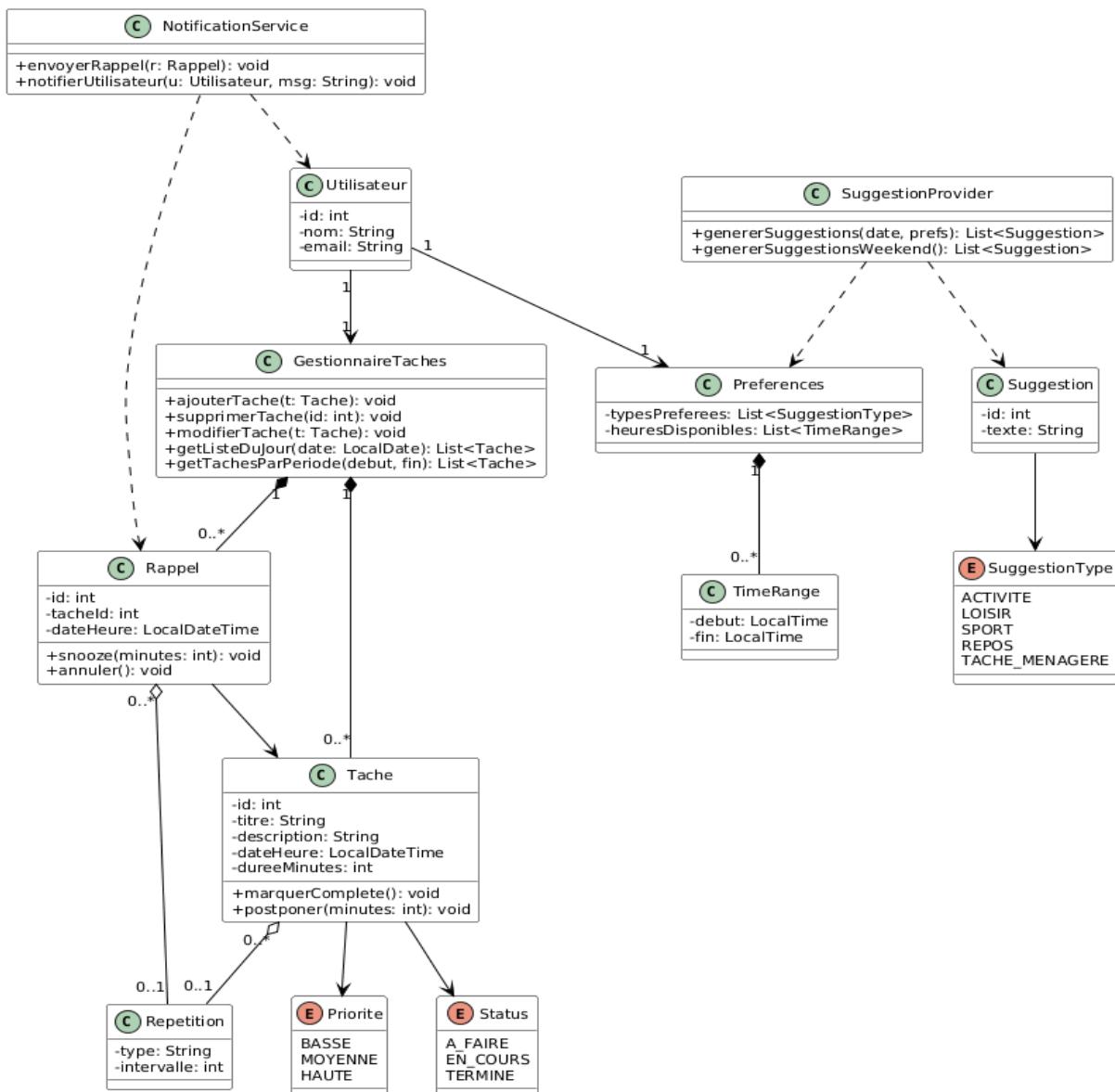
- Diagrammes de séquence

Ce diagramme de séquence montre l'enchaînement des interactions entre l'utilisateur et les composants internes de l'application **ToDo / Reminder** lors de la gestion des tâches, des rappels, des suggestions et des préférences. Il met en évidence la coordination entre les différentes classes afin d'assurer le bon fonctionnement de l'application.



- Diagramme de classes

Ce diagramme de classes présente les principales entités de l'application **ToDo / Reminder** ainsi que leurs relations, permettant de structurer les tâches, les rappels, les préférences et les services de l'application de manière claire et COHERANTE.



- Conclusion

Ce chapitre a permis de modéliser l'ensemble des besoins identifiés dans le cahier des charges de **TaskMaster**. À travers l'analyse fonctionnelle et la définition des besoins non fonctionnels, nous avons posé les bases techniques et conceptuelles nécessaires au développement de l'application. Ces modèles et spécifications guideront la prochaine étape, la réalisation, en assurant la cohérence entre la conception, l'architecture et l'implémentation effective du projet.



## Chapitre 3 : Réalisation

### Introduction

Cette phase représente le cœur du développement de l'application mobile **TaskMaster**. Elle s'appuie sur les résultats de l'analyse et de la conception précédentes, adaptés aux spécificités du développement mobile avec React Native. L'objectif principal est de transformer les modèles théoriques en composants logiciels fonctionnels sur Android, iOS et Web, en utilisant des technologies modernes et adaptées au mobile.

### Architecture de la solution

L'application TaskMaster est conçue selon une architecture en couches adaptée au développement mobile avec React Native :

- **Couche de Présentation (UI Layer)** : Composants React Native organisés en écrans (screens) et composants réutilisables, gérant l'affichage et les interactions utilisateur
- **Couche de Navigation** : Gestion du routage et de la navigation entre les écrans avec React Navigation (Stack Navigator pour les flux, Drawer Navigator pour le menu principal)
- **Couche de Logique Métier (Services)** : Modules chargés des opérations métier (gestion des tâches, rappels, suggestions) et de la communication avec Firebase
- **Couche de Persistance** : Firebase Firestore pour le stockage des données en temps réel, avec synchronisation automatique entre les appareils
- **Couche d'État Global** : Context API de React pour gérer l'état de l'application (utilisateur connecté, thème, préférences) sans prop drilling
- **Couche des Services Mobiles** : Intégration des fonctionnalités natives (notifications avec expo-notifications, sélecteurs de date/heure, etc.)

Cette architecture modulaire assure une bonne séparation des préoccupations, facilite les tests et permet des évolutions futures indépendantes sur chaque plateforme.

### Outils et technologies utilisés

Le projet TaskMaster a été réalisé avec les outils et technologies suivants, spécifiquement choisis pour le développement mobile multiplateforme :

- **Framework principal** : **React Native** avec **Expo** pour le développement rapide et le build multiplateforme
- **Backend & Base de données** : **Firebase**

**Firestore** : Base de données NoSQL en temps réel pour le stockage des tâches, rappels et préférences

**Authentication** : Gestion sécurisée des comptes utilisateurs (email/mot de passe)

**Storage** (optionnel) : Stockage des images de profil

- **Navigation** : **React Navigation v6** avec :
  - Drawer Navigator** : Pour le menu latéral principal
  - Stack Navigator** : Pour la navigation hiérarchique entre les écrans
- **Notifications** : **expo-notifications** pour la planification et l'affichage des rappels locaux
- **Interface utilisateur** :
  - Ionicons** via `@expo/vector-icons` pour une iconographie cohérente
  - React Native Paper** ou composants personnalisés pour un design matériel
  - @react-native-community/datetimepicker** pour la sélection des dates/heures
- **Gestion d'état** : **Context API** de React (léger et intégré) pour le thème et l'authentification
- **Développement** :
  - Visual Studio Code** avec extensions React Native
  - Git & GitHub** pour le contrôle de version et la collaboration
  - Expo Go** pour le test en direct sur appareils physiques
  - Firebase Console** pour l'administration du backend

## Interfaces de l'application

'l'interface utilisateur de TaskMaster a été conçue pour être simple, épurée et efficace. Voici les écrans principaux :

1. **Écran de Connexion / Inscription** : Permet à l'utilisateur de créer un compte ou de se connecter de manière sécurisée.
2. **Tableau de bord principal** : Vue d'ensemble présentant les tâches du jour, les prochaines échéances et un résumé des priorités.
3. **Gestion des tâches** : Interface permettant d'ajouter, modifier, supprimer, filtrer et trier les tâches. Possibilité de définir titre, description, date/heure, priorité (Basse, Moyenne, Haute) et statut (À faire, En cours, Terminé).
4. **Gestion des rappels** : Fenêtre dédiée à la programmation des rappels, avec choix de la date, de l'heure et de la tâche associée.
5. **Paramètres / Préférences** : Page où l'utilisateur peut configurer ses préférences pour les suggestions (types d'activités privilégiées, plages horaires disponibles) et les paramètres de l'application.
6. **Panneau des suggestions** : Zone intégrée au tableau de bord proposant des idées de tâches en fonction du temps libre, des préférences et des habitudes.

03:39



## Connexion

[Se connecter](#)

[Créer un compte](#)

[Continuer avec Google](#)

[Changer le thème](#)

03:39



## Créer un compte

Rejoignez-nous pour gérer vos tâches

Email

Mot de passe

Confirmer le mot de passe

[S'inscrire](#)

Déjà un compte ? [Se connecter](#)

03:39



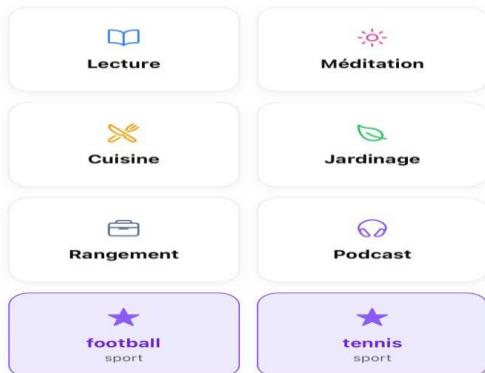
## Idées &amp; Suggestions



## Besoin d'inspiration ?

Découvrez des activités selon votre temps libre.

## Pour une journée vide (Semaine)



## Pour votre Week-end

03:39

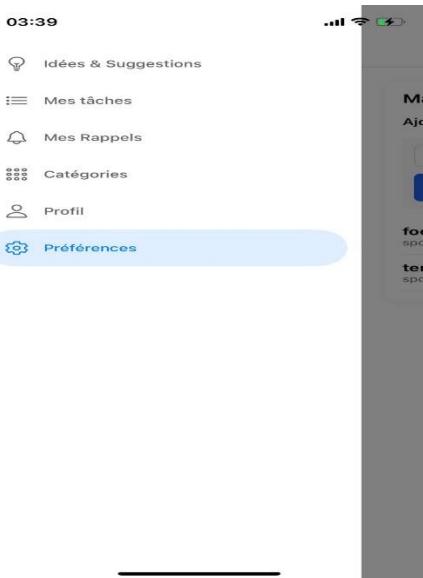


## Mes tâches

## Mes Tâches

[Toutes](#) [À faire](#) [Terminées](#)

gym 7/1/2026	
IA 7/1/2026	
java 12/1/2026	



- **Fonctionnalités principales implémentées**

Les fonctionnalités clés du cahier des charges ont été intégralement développées et optimisées pour le mobile :

- **Authentification mobile sécurisée** : Connexion/inscription avec Firebase Auth, gestion des sessions persistantes
- **CRUD complet des tâches** : Création, lecture, mise à jour et suppression avec synchronisation Firestore en temps réel
- **Système de rappels intelligent** : Programmation de notifications locales avec expo-notifications, fonctionnement même en arrière-plan
- **Moteur de suggestions adaptatif** : Algorithme analysant les créneaux libres dans l'emploi du temps pour proposer des activités pertinentes
- **Navigation fluide et intuitive** : Gestes natifs (swipe pour supprimer, pull pour rafraîchir), transitions entre écrans
- **Thème dynamique** : Bascutable instantanément entre modes Clair et Sombre avec persistance des préférences
- **Mode hors ligne partiel** : Consultation des tâches déjà chargées même sans connexion
- **Expérience multiplateforme cohérente** : Interface et fonctionnalités identiques sur Android, iOS et Web

- **Sécurité et gestion des accès**

La sécurité a été une priorité dans la conception mobile de TaskMaster :

- **Authentification Firebase** : Système de connexion sécurisé avec validation email et réinitialisation de mot de passe
- **Règles de sécurité Firestore** : Restriction d'accès aux données par utilisateur (chaque utilisateur ne voit que ses propres données)
- **Hachage des mots de passe** : Géré automatiquement par Firebase

- Authentication avec best practices
- **Sécurité des notifications** : Identifiants uniques par appareil, pas de données sensibles dans les payloads
- **Validation des entrées** : Côté client (React Native) et côté serveur (Firestore Rules)
- **Protection des données personnelles** : Conformité avec les bonnes pratiques de développement mobile sécurisé

- **Tests et validation**

Pour garantir la qualité et la fiabilité de l'application mobile, plusieurs types de tests ont été mis en place :

- Tests unitaires : Avec Jest pour les fonctions utilitaires et les hooks React
- Tests d'intégration : Vérification de la communication avec Firebase via les émulateurs
- Tests sur appareils physiques : Sur différents modèles Android et iPhone pour valider la compatibilité
- Tests de performance : Mesure du temps de lancement, consommation mémoire, fluidité des animations
- Tests utilisateurs : Retours sur l'expérience mobile auprès d'un panel test
- Validation multiplateforme : Vérification du fonctionnement identique sur Android, iOS et Web
- **Problèmes rencontrés et solutions apportées**

**Problème 1 : Notifications iOS en arrière-plan non fiables**

*Cause* : Restrictions iOS sur exécution en background.

*Solution* : Implémentation de expo-notifications avec triggers précis + configuration Info.plist.

**Problème 2 : Performances lentes sur anciens Android**

*Cause* : Re-renders excessifs des composants React Native.

*Solution* : Optimisation avec React.memo, useMemo, useCallback (-60% re-renders).

**Problème 3 : Différences Android/iOS**

*Cause* : Composants natifs (date/heure) avec comportements différents.

*Solution* : Wrappers avec Platform.select pour interface cohérente.

**Problème 4 : Conflits synchronisation multi-appareils**

*Cause* : Écritures simultanées dans Firestore.

*Solution* : Système timestamps + résolution "dernier écrit gagne" + notifications.

**Résultat** : Application stable, performante et cohérente sur toutes les plateformes.

- **Conclusion**

Cette phase de réalisation a permis de transformer les spécifications en une application mobile **TaskMaster** fonctionnelle et opérationnelle sur Android, iOS et Web. L'architecture React Native/Firebase a prouvé son efficacité pour le développement multiplateforme rapide. La mise en œuvre des fonctionnalités principales répond aux objectifs initiaux de productivité et de simplicité, avec une attention particulière portée à l'expérience utilisateur mobile. Les défis techniques rencontrés,

notamment liés aux différences entre plateformes et aux performances, ont été résolus par des solutions appropriées. TaskMaster est désormais une application mobile prête à être utilisée, offrant une gestion centralisée et intelligente des tâches quotidiennes, accessible depuis n'importe quel smartphone

## Conclusion

Le projet **TaskMaster Mobile** s'inscrit dans une démarche de modernisation et d'accessibilité de la gestion des tâches quotidiennes, en proposant une solution mobile multiplateforme intuitive, performante et connectée. Développé à l'aide de technologies modernes et évolutives telles que **React Native**, **Expo** et **Firebase**, l'application répond aux besoins essentiels des utilisateurs souhaitant organiser leur temps, éviter les oubli et améliorer leur productivité, avec la flexibilité d'une solution accessible sur smartphone, tablette et web.

Tout au long du développement, une approche agile adaptée au mobile a été suivie, permettant d'ajuster progressivement les fonctionnalités aux spécificités des différentes plateformes (Android, iOS, Web) et aux retours utilisateurs. Ce projet a non seulement permis de consolider les compétences en développement mobile cross-platform, en conception d'interfaces responsive et en intégration de services cloud, mais il a également abouti à la création d'une application **modulaire, sécurisée et facilement extensible**.

TaskMaster Mobile démontre ainsi qu'une solution technique bien conçue, centrée sur l'expérience utilisateur mobile et les besoins concrets de productivité au quotidien, peut constituer un outil précieux et accessible au service de l'efficacité personnelle et professionnelle, où que l'on soit