



# Application Mobile: Gloire- ToDoList



## 1. Introduction

### 1.1. Contexte et Motivation

Dans notre quotidien personnel ou professionnel, il devient essentiel de gérer efficacement nos tâches. Le but de ce projet est de proposer une **application mobile simple, rapide et intuitive** permettant à tout utilisateur de créer, suivre, modifier et supprimer ses tâches.

Cette application a été développée en utilisant **React Native** avec **TypeScript** et l'environnement **Expo**, afin de cibler à la fois les plateformes Android et iOS.



## 2. Objectifs du projet

- Permettre la **création rapide de tâches**.
- Offrir une **interface épurée et ergonomique**.
- Autoriser la **modification ou suppression** des tâches.
- Permettre à l'utilisateur de **cocher les tâches réalisées**.
- Assurer une **persistance locale** des données (même après redémarrage de l'application).
- Démontrer l'utilisation de **React Native avec TypeScript**, l'architecture des composants, la gestion d'état, et le stockage local.



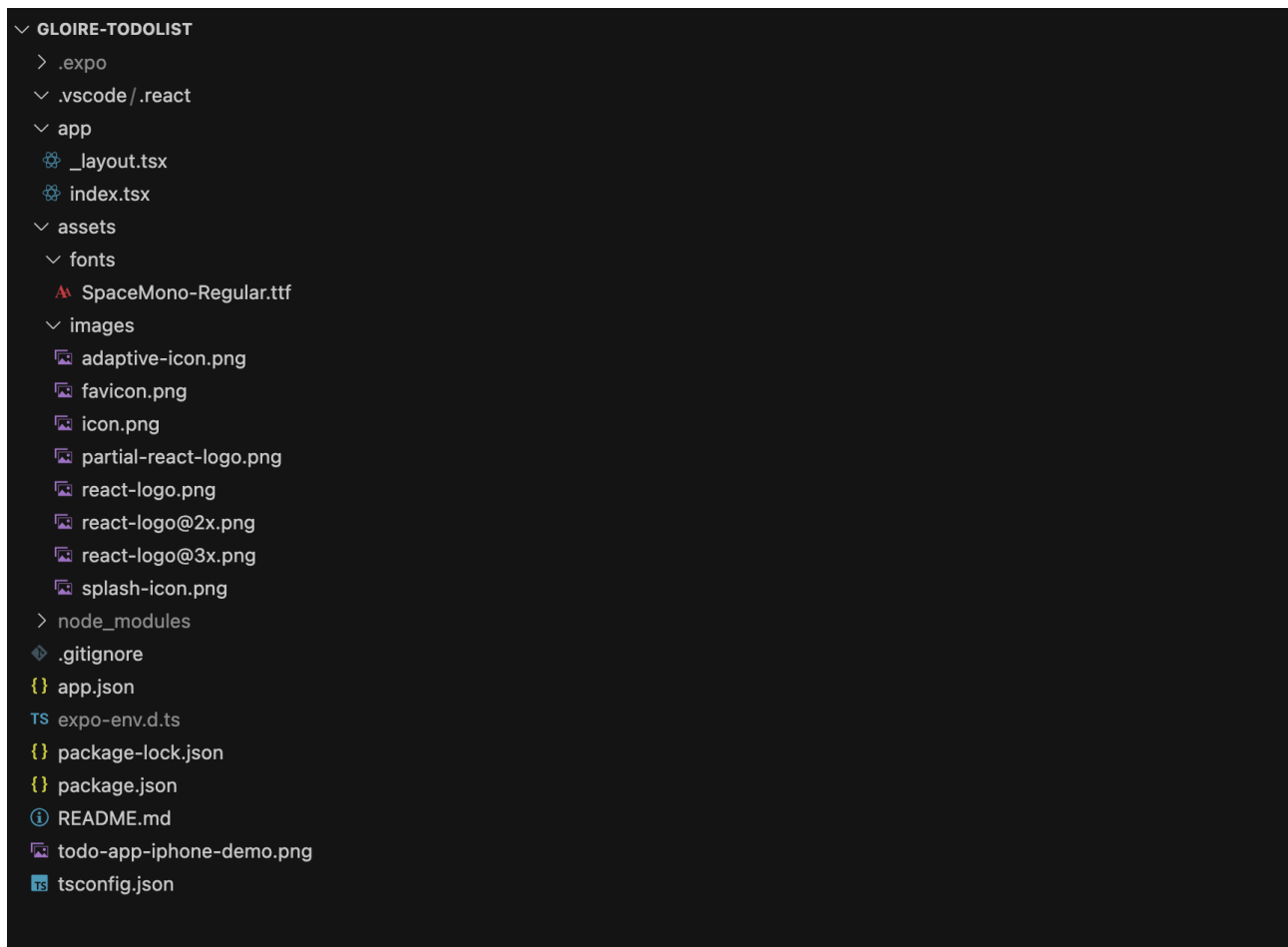
## 3. Technologies et outils utilisés

Outil / Technologie	Rôle
React Native	Framework pour le développement mobile multiplateforme
TypeScript	Typage statique pour une meilleure fiabilité du code
Expo	Plateforme de build et test simplifié pour React Native
AsyncStorage	Stockage local pour persister les données

<b>React Hooks</b>	Gestion de l'état fonctionnel dans les composants
<b>Modales React</b>	Interaction avec l'utilisateur pour ajouter ou modifier les tâches

## **4. Architecture de l'application**

### **4.1. Structure des dossiers**



## **5. Fonctionnalités détaillées**

### **5.1. Ajout d'une tâche**

- L'utilisateur saisit un texte dans un champ.
- Il valide, et la tâche s'ajoute à la liste avec un identifiant unique.
- La liste se met à jour automatiquement.

11:13



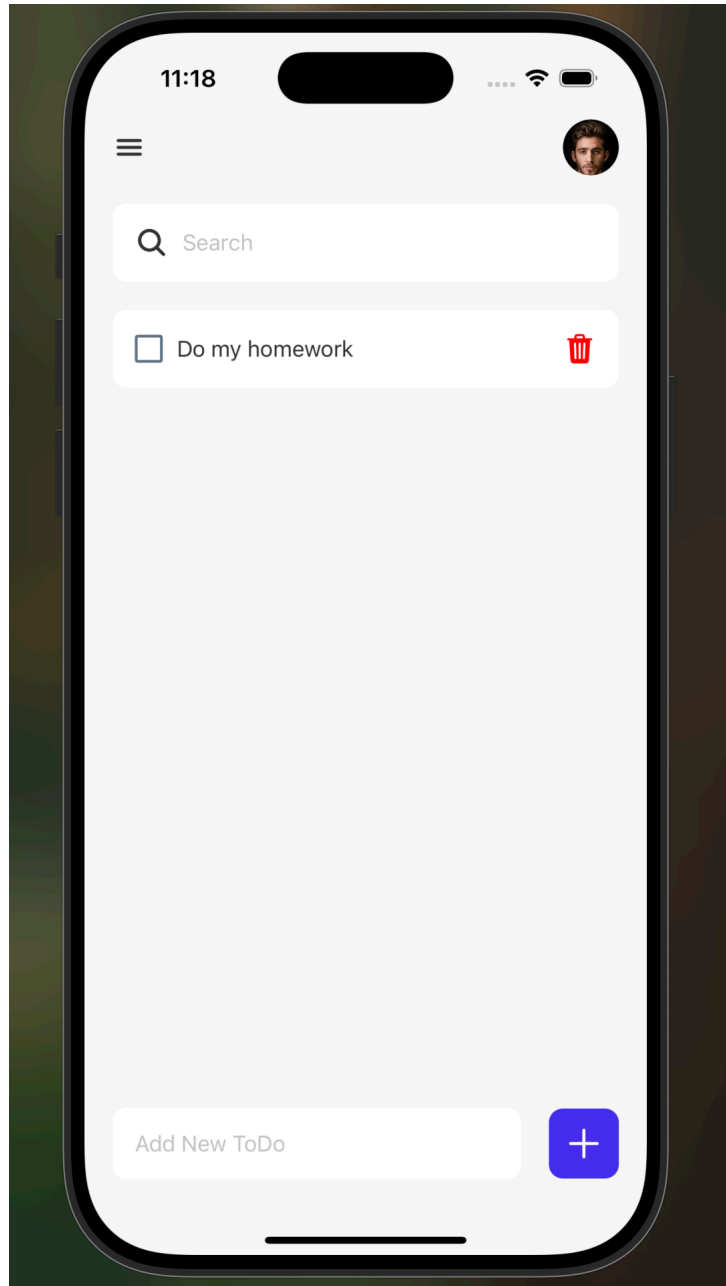
 Search

Add New ToDo



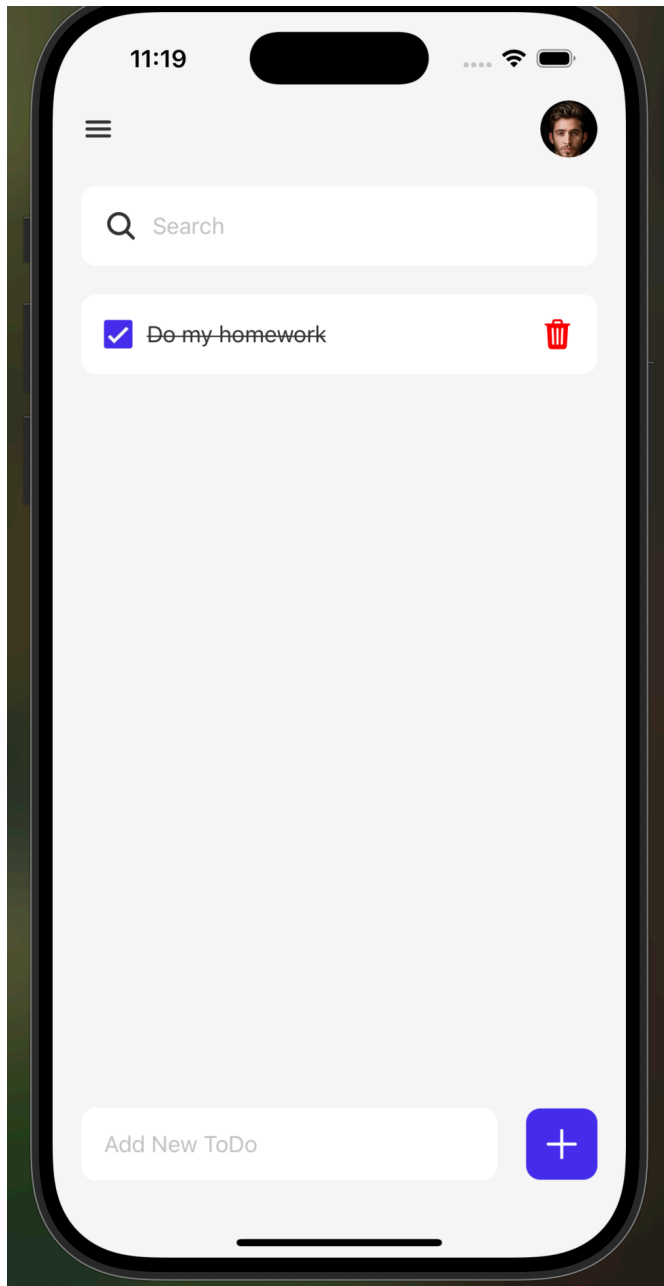
## 5.2 Suppression d'une tâche

- Depuis la page principale, un bouton corbeille qui permet de supprimer définitivement la tâche.



### ✓ 5.3. Marquage comme complétée

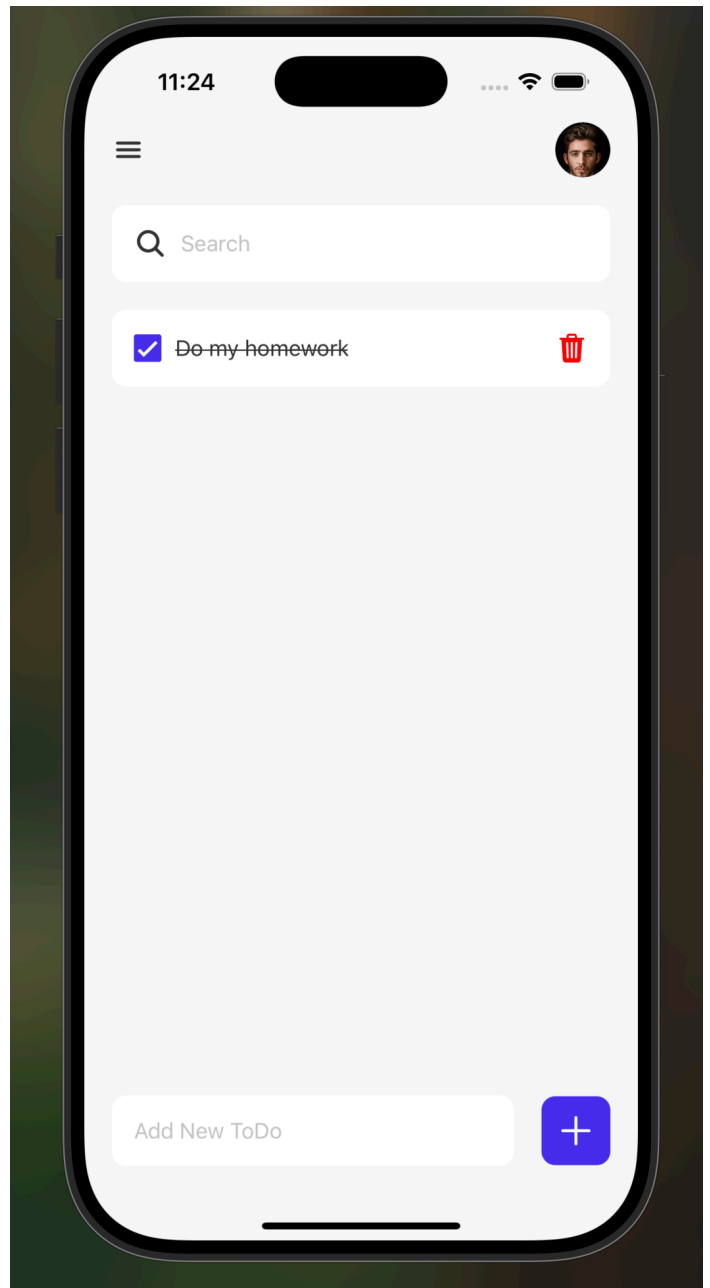
- Chaque tâche possède une case à cocher.
- Une tâche cochée est affichée en texte barré.





## 5.5. Sauvegarde locale avec AsyncStorage

- Toutes les tâches sont enregistrées dans la mémoire de l'appareil.
- Elles sont rechargées automatiquement au lancement de l'application.

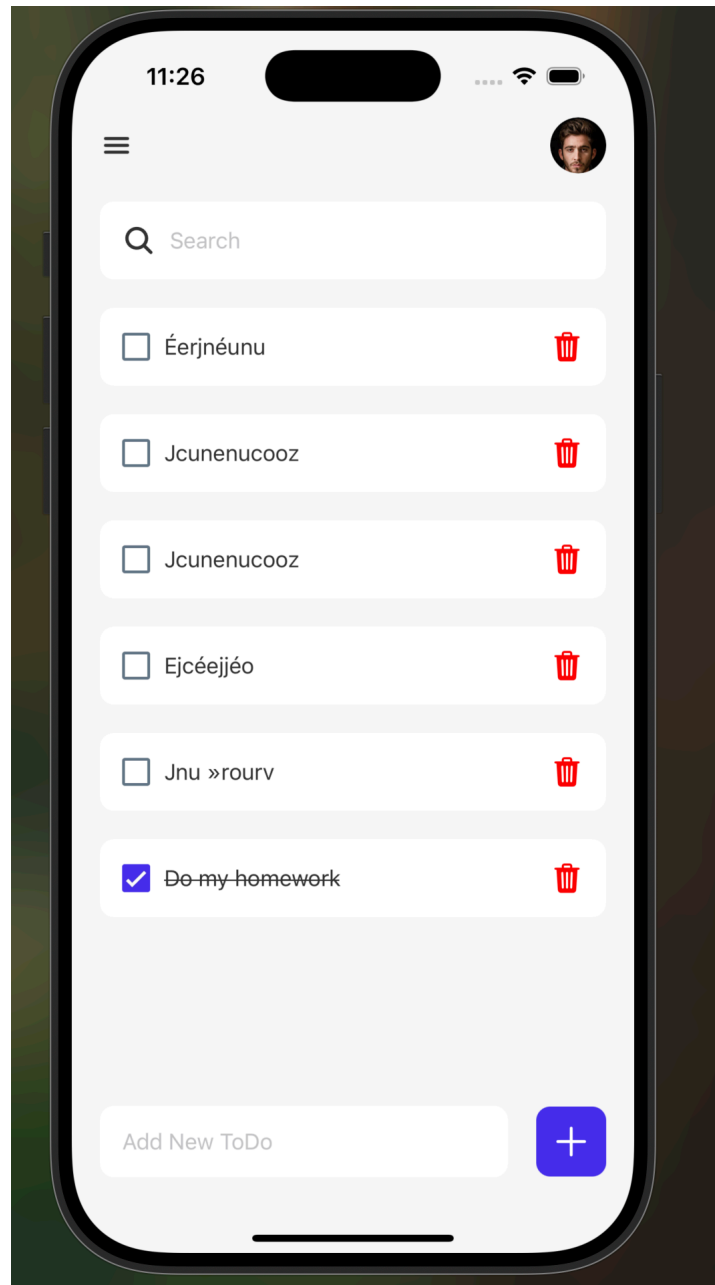


## 6. Design et expérience utilisateur

L'interface suit une **approche minimaliste et intuitive**, pensée pour une utilisation rapide.

## **Couleurs et design :**

- Fond blanc et gris clair
- Icônes sobres
- Tâches bien espacées et lisibles
- Actions accessibles en un seul geste



## **7. Fonctionnement interne**



## 7.1. Gestion d'état

L'état de l'application est centralisé dans un **hook personnalisé** (`useTaskStore.ts`). Il contient :

- La liste des tâches
- Les méthodes pour ajouter, modifier, supprimer et marquer comme terminé

## 7.2. Persistance des données

- Les tâches sont enregistrées sous forme de JSON via `AsyncStorage`.
- Un effet `useEffect` est utilisé pour recharger les données au démarrage.



## 8. Sécurité et limitations

- Les données sont stockées **localement**, donc aucune synchronisation entre plusieurs appareils.
- Pas de système de comptes utilisateurs.
- Pas de notifications pour les rappels.

Ces limites permettent de maintenir l'application **légère et rapide**, idéale pour un usage simple.



## 9. Améliorations possibles

Amélioration	Description
Notifications	Rappels pour les tâches proches de l'échéance
Priorisation	Ajouter un niveau de priorité (faible, moyen, élevé)
Catégories	Permettre de regrouper les tâches par thème
Thème sombre	Choix entre thème clair et sombre
Sauvegarde cloud (Firebase)	Permettre la sauvegarde en ligne et la synchronisation



## 10. Tests et validation

- Test de chaque fonctionnalité réalisée sur IOS via Xcode.
- Comportement vérifié :
  - Ajout/mise à jour/suppression d'une tâche
  - Persistance après redémarrage
  - Affichage fluide des tâches
- Résultat : application stable, fluide et fonctionnelle sur appareils récents.



## **11. Résumé des captures d'écran à intégrer**

N°	Description
1	Interface initiale avec champ d'ajout vide
2	Ajout d'une tâche et son apparition dans la liste
3	Tâche cochée comme "faite" (barrée/atténuée)
4	Fenêtre modale d'édition d'une tâche
5	Suppression d'une tâche
6	Interface après rechargement de l'app (avec persistance)
7	Vue d'ensemble avec plusieurs tâches ajoutées



## **12. Conclusion**

Ce projet démontre de manière claire comment concevoir une application mobile multiplateforme avec React Native en intégrant des fonctionnalités essentielles : interface conviviale, gestion d'état, persistance des données, modales interactives et manipulation d'une liste dynamique.

Il constitue une **excellente base pédagogique** pour tout développeur souhaitant se lancer dans le développement mobile, tout en offrant un outil réellement utile au quotidien.

**BOUNGOU MBIMI Gloire Bryan**