	<h2 style="margin: 0;">EXAMEN</h2> <p>Semestre : 1 <input type="checkbox"/> 2 <input checked="" type="checkbox"/></p> <p>Session : Principale <input checked="" type="checkbox"/> Rattrapage <input type="checkbox"/></p>
<p>Module : Développement mobile Cross platform Enseignant(s) : Equipe mobile Classe(s) : 4 SIM 1,2,3,4,5</p>	
<p>Documents autorisés : OUI Nombre de pages : 4 Calculatrice autorisée : NON Internet autorisée : NON</p>	
<p>Date : 22/06/2023 Heure : 11h00 Durée : 01h30</p>	

*** La validation tient compte seulement de l'exécutable de l'application, l'étudiant sera noté pour chaque interface fonctionnelle.**

*** Il faut respecter les dispositions des composants graphiques, leurs formes et leurs tailles comme dans les figures en dessous.**

*** Pour exécuter votre projet sur un emulateur android, utilisez l'adresse suivante : 10.0.2.2**

*** Les ressources numériques sont : examen en version pdf et une collection postman et insomnia**

Vous êtes amenés à développer une application mobile android et iOS afin de faciliter la gestion des tâches « Task Manager ».

Pour ce faire, vous allez utiliser « Flutter » afin de réaliser les différentes interfaces/fonctionnalités de l'application comme suit :

Interface « Home » (8 pts) :

- L'application est représentée par un menu de type BottomNavigationBar contenant trois rubrique : Home (icône : home), Add (icône: add) et Cloud (icône: cloud).

L'interface Home contient (voir figure 1) :

- Une liste des tâches to do ou bien done
- Chaque élément de la liste contient :
 - . Titre de la tâche
 - . Description de la tâche
 - . Deadline (de type string)

. Icône pour l'état de la tâche (icône circle) : la couleur du cercle est en **vert** si la tâche est done, **rouge** sinon

- Les tâches sont récupérées à partir du stockage interne de l'appareil (sqflite)
- Si le stockage interne est vide, aucune liste ne sera affichée

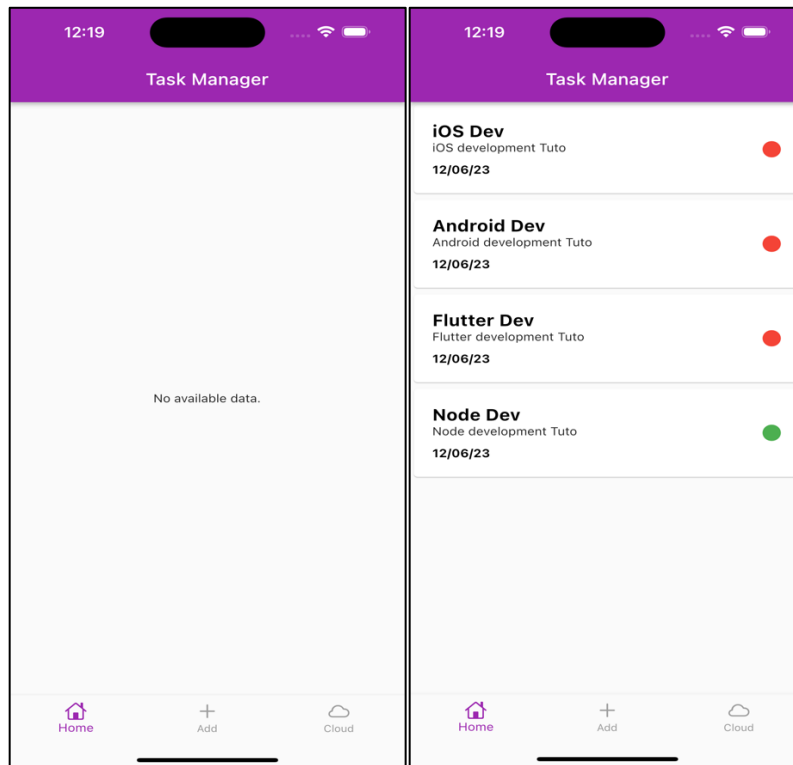


Figure 1: Interface Home

Interface « Add Task» (7 pts):

- Cette interface contient (voir figures 2 et 3) :

- Un champ pour le titre
- Un champ pour la description
- Un champ pour la date limite (deadline)
- Un bouton « Save » afin d'ajouter la tâche

En ajoutant une tâche, son état sera initialement todo donc affichée avec un cercle dont la couleur est **rouge**.

NB : Les champs de textes doivent être obligatoirement remplis pour l'ajout d'une tâche, un message d'erreur s'affiche sinon.

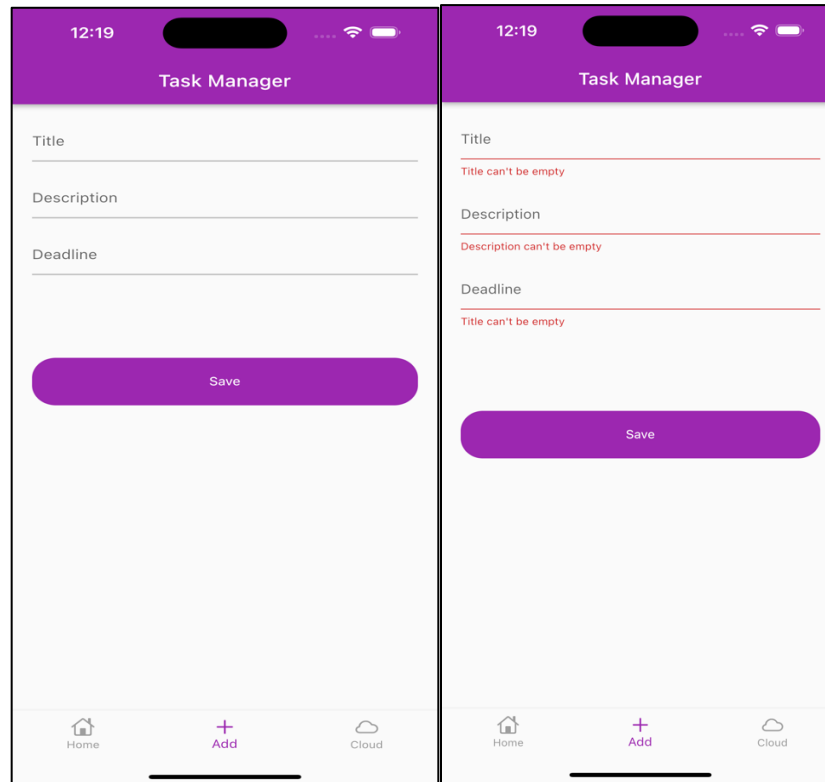


Figure 2 : Interface Add Task

En cas de succès, une boîte de dialogue sera affichée contenant le message suivant : « Task added successfully », avec le titre « Info » et l'icône « info » (Voir figure 3)

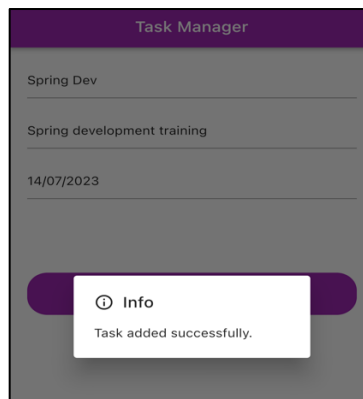


Figure 3: Task Added

Interface « Cloud » (5pts):

- Cette interface contient une liste des tâches récupérées à partir du serveur (Voir annexe), et un bouton permettant de synchroniser les tâches en local avec le serveur dont le titre est

« **Sync with cloud** » et l'icône « **sync** ».

- En cliquant sur le bouton, les tâches sauvegardées au niveau de la base des données interne seront sauvegardées au niveau de la base des données externe via un web service (Voir annexe).
- Une fois les données envoyées vers le serveur, la base de données locale sera vidée.

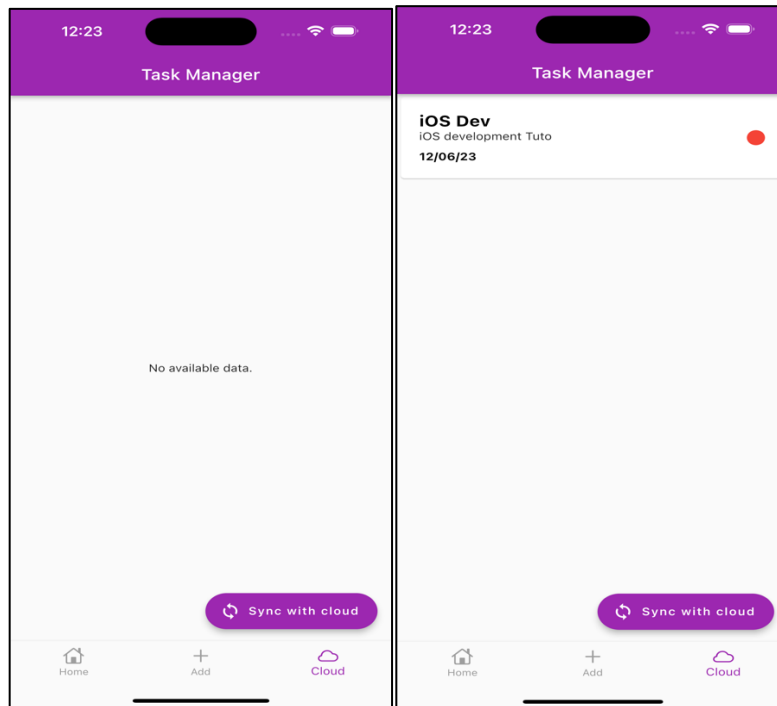


Figure 4 – Interface Cloud

Annexe

Ci-dessous les liens des web services à consommer :

Synchronise : méthode **POST** pour ajouter une tâche: <http://localhost:9090/tasks>

Liste des tâches : méthode **GET** pour récupérer la liste des tâches: <http://localhost:9090/tasks>