

**4A – IT**

**Semestre S7**

## **Programmation Application Mobile iOS**

**COMPTE RENDU DU TP  
TODOLIST**

**Réalisé par :**

**EL AMRI Mohamed.  
CHEIKHROUHOU Yacine.**

**Encadré par :**

**Pr. GINHAC Dominique**

**Année Universitaire : 2022/2023.**

## 1. Introduction :

L'objectif de ce TP est de développer une application complète de type Gestionnaire de tâches (Todo list). Le développement de cette application sera fait en respectant le modèle MVC et se décomposera en plusieurs phases, chacune d'elles permettant d'ajouter de nouvelles fonctionnalités. La réalisation de cette application nous a permis de revoir les connaissances vues en TD.

## 2. Choix de design :

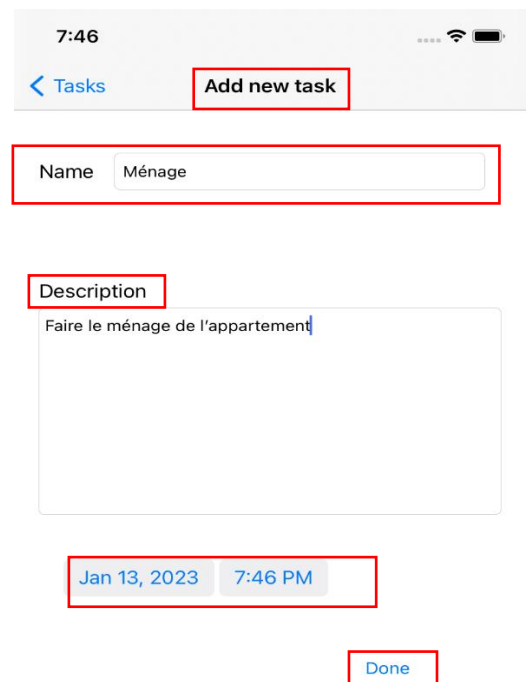
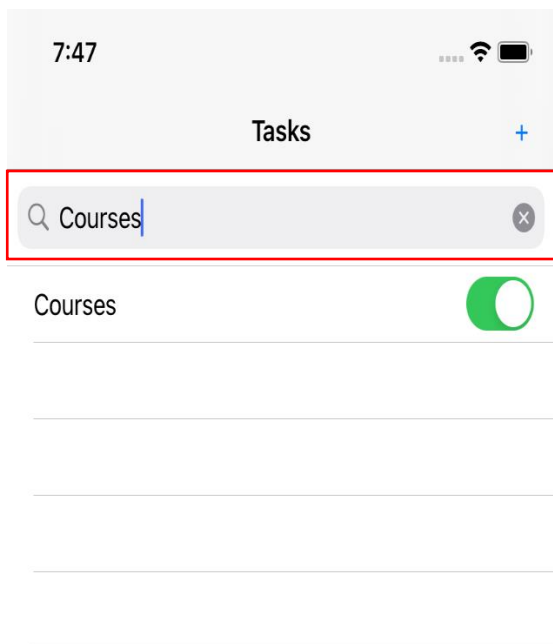
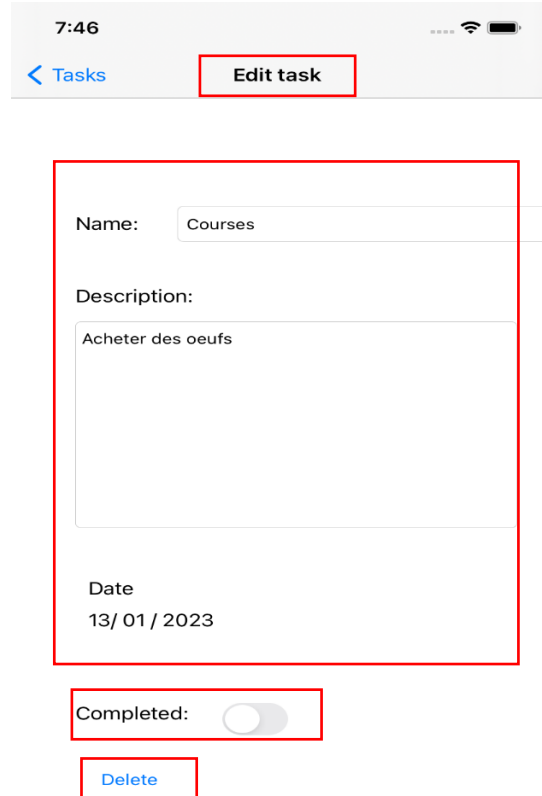
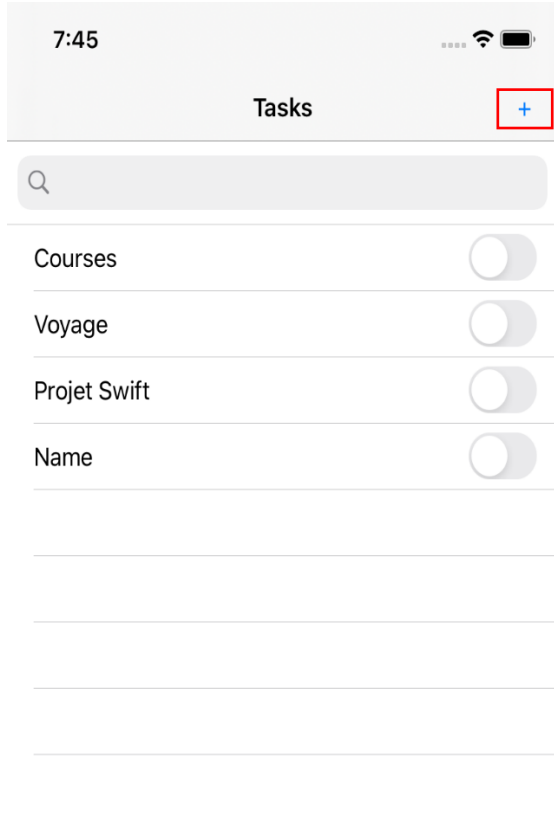
- Les vues sont implémentées en utilisant les fonctionnalités du Storyboard.
- Le Storyboard utilise un UINavigationController pour gérer la pile des vues et les retours en arrière. Après l'ajout ou la modification d'un Todo, le Navigation revient à la vue principale et rafraîchit le TableView.

## 3. Choix de programmation :

- La liste des Todo est séparée en une liste logique et une liste affichée (car la barre de recherche filtre les résultats).
- Si la barre de recherche est utilisée, le TableView affiche une sous-liste de l'ensemble des Todo. Le filtre de recherche est réinitialisé après l'ajout d'un nouveau Todo.
- Pour les textes, nous avons adopté le style graphique des champs de texte pour les rendre plus visibles sur l'écran.
- Nous avons utilisé un bouton Switch sur le (TableViewCell) pour représenter la complétion d'un Todo.
- Nous avons affiché la date dans la vue détaillée d'un Todo, et les tâches sont triées par ordre chronologique, et la date d'un Todo est définie lors de sa création.
- L'application Storyboard utilise des vues contrôlées par un ViewController chacune, qui sont gérées par une pile dans le UINavigationController du Storyboard.
- L'utilisation d'une vue pour chaque aspect de l'interface permet d'ajouter des fonctionnalités à chaque vue en regroupant le code dédié à chaque vue dans son ViewController, et d'ajouter le code de gestion des outlets et des événements.
- Pour correspondre aux vues, il y a un ViewController principal, un TodoViewController pour la vue détaillée d'un Todo, et un AddViewController pour la vue d'ajout d'un Todo.
- Nous avons utilisé une classe dédiée ToggledTableViewCell pour les TableViewItem qui ont un bouton Switch.

- Les contrôleurs des vues secondaires ont tous une référence vers le ViewController principal, qui est initialisée par celui-ci lors de la création des vues.

#### 4. Différentes vues lors de la simulation :



7:47

Tasks

+

Courses

Voyage

Projet Swift

Name

Ménage