

Parcours Développeur d'application iOS
Projet 10 – Reciplease
Bonus

Pour le projet 10 du parcours Développeur d'application iOS, il est demandé de créer une application de recherche de recette grâce à une API nommée Yummly. L'application nous permet de rajouter les ingrédients qui sont dans notre frigo et de nous trouver les recettes associées. Il est possible de voir les détails de la recette et de l'ajouter dans les favoris.

Pour ce projet il était demandé de rajouter un bonus. J'ai donc choisi d'intégrer une « Search Bar » sur l'écran des favoris afin de pouvoir rechercher directement une recette précise.

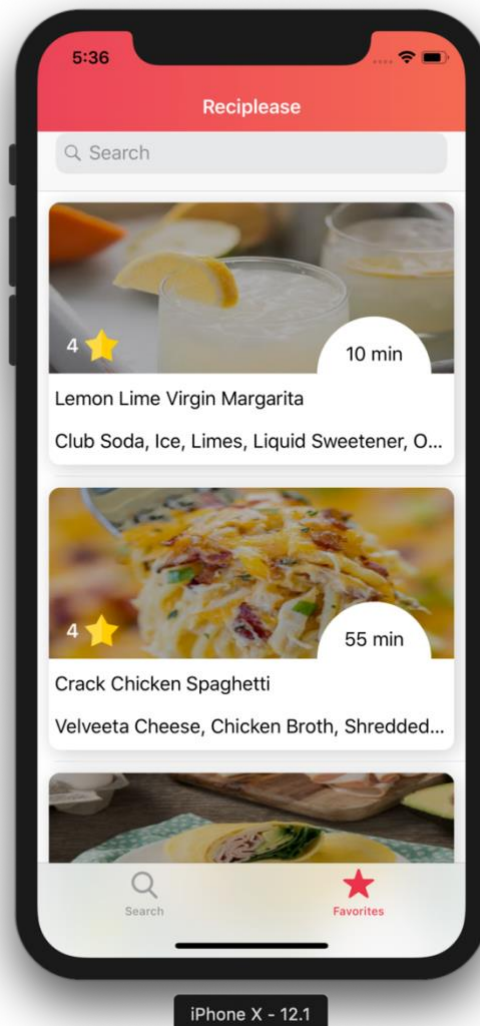


Figure 1 : L'écran des favoris avec la "Search Bar"

Pour ce faire il existe une solution native simple, en ajoutant le code suivant :

```
lazy var searchController: UISearchController = {  
    let searchController = UISearchController(searchResultsController: nil)  
    searchController.searchBar.delegate = self  
    searchController.definesPresentationContext = true  
    return searchController  
}()
```

Figure 2 : Le code pour ajouter une search bar

En ajoutant cette propriété au contrôleur qui contient au préalable un UITableView, on obtient directement le résultat de la figure 1. Cette mise en place est particulièrement simple et nous pouvons remercier Apple pour cette solution native.

Afin de rendre la search bar opérationnelle il suffit de spécifier à notre contrôleur qu'il doit suivre un protocole de type UISearchBarDelegate. Puis il suffit d'implémenter les différentes fonctions, comme sur la figure ci-dessous :

```
// MARK: - Extensions
extension FavoriteViewController: UISearchBarDelegate {
    func searchBar(_ searchBar: UISearchBar, textDidChange searchText: String) {
        if searchText.count == 0 {
            favoriteRecipes = Recipe.fetchAll()
        } else {
            favoriteRecipes = Recipe.fetch(with: searchText)
        }
        favoriteTableView.reloadData()
    }

    func searchBarCancelButtonClicked(_ searchBar: UISearchBar) {
        favoriteRecipes = Recipe.fetchAll()
        favoriteTableView.reloadData()
    }
}
```

Figure 3 : Le contrôleur suit désormais le protocole de la search bar

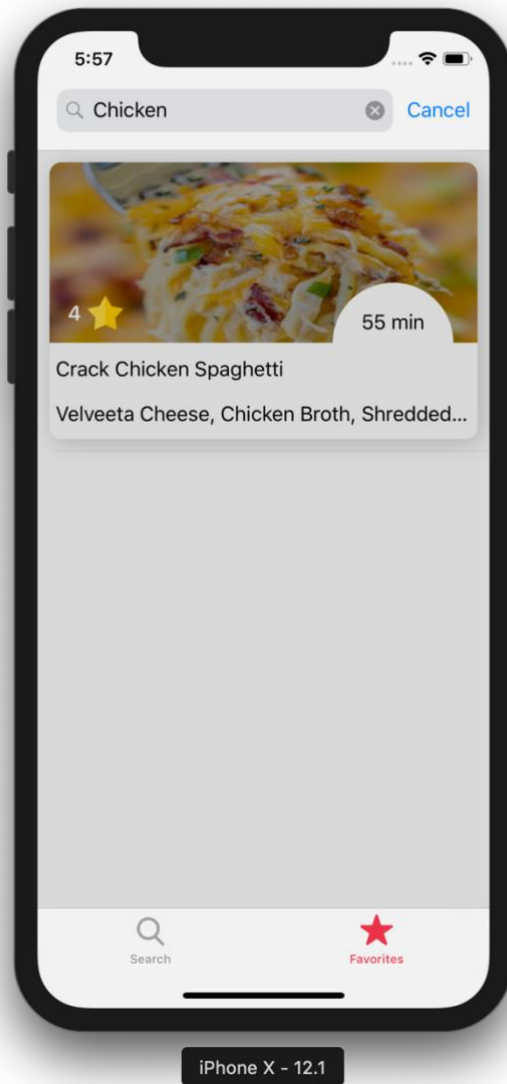
Pour pouvoir afficher en temps réel la liste des recettes filtrées, il nous suffit d'ajouter une condition. Lorsque le texte présent dans la search bar est vide, alors on recharge tous les éléments présents dans CoreData. Sinon, on va simplement utiliser la fonction fetch(with : String) de notre entité Recipe, pour pouvoir récupérer uniquement les recettes avec le nom correspondant.

```
static func fetch(viewContext: NSManagedObjectContext = AppDelegate.viewContext, with name: String) -> [Recipe] {
    let request: NSFetchRequest<Recipe> = Recipe.fetchRequest()
    request.predicate = NSPredicate(format: "name CONTAINS[cd] %@", name)
    guard let recipes = try? viewContext.fetch(request) else { return [] }
    return recipes
}
```

Figure 4 : La méthode fetch(with: String) de l'entité Recipe

Cette méthode va rechercher dans CoreData tous les éléments qui contiennent l'élément « name » passé en paramètre.

Puis il ne nous reste plus qu'à recharger les éléments du tableau pour avoir une liste qui se met à jour en temps réel.



iPhone X - 12.1

Figure 5 : L'écran des favoris avec un filtre "Chicken" ajouté grâce à la Search Bar