

Compte Rendu Projet SWIFT

Objectif

L'objectif est de développer une application complète de type Gestionnaire de tâches (Todo list). Le développement de cette application a été fait en respectant le modèle MVC. Pour ce faire l'outil XCode servi de plateforme de développement.

1. Conception du modèle de données

Nous créons dans un premier temps une classe Todo qui servira de modèle de données pour notre application. Une tâche possède un nom et une description.

2. Création du Storyboard et Programmation de l'application

Notre application est conçue sur le principe d'un TableView. Pour notre première vue, nous ajoutons directement dans le MainStoryboard un Table View avec les contraintes correspondantes pour un bon affichage. On insère aussi un cellule prototype incluant deux Label permettant d'afficher le nom et la description de la tâche(avec de bonnes contraintes).

L'affichage des différentes tâches se fera l'aide de l' UITableViewDataSource. Cette fournit le nombre de lignes et de sections. `func tableView(UITableView, numberOfRowsInSection: Int) -> Int` Indique à la source de données de renvoyer le nombre de lignes dans une section donnée d'une vue de table. On n'oubliera pas de lier notre cellule prototype à la classe de type TableViewCell crée au préalable. Ans le ViewController on crée un tableau de type Todo qui contiendra nos différentes objets. Ce tableau sera rempli à l'aide d'une boucle for dans le `viewDidLoad()`.

`myTableView.dataSource = self` sera inscrit dans le `viewDidLoad()` et permettra d'indiquer la source de données.

On doit dès à présent piloter la vue permettant d'afficher le détail d'un tâche. Pour ce faire on crée une nouvelle classe qui va hériter de la classe `UIViewController()`. On associe la vue à la classe qu'on vient de créer.

On veut présentement envoyer des données vers le ViewController de destination. La méthode utilisé est la méthode `prepare(for segue)` qui va préparer les instructions qui vont être exécuter au moment ou la deuxième vue va s'afficher. On s'intéresse à la variable destination de cette méthode (`segue.destination`).

3. Ajout de la date

On ajoute un `Date Picker()`. On modifie le modèle de données en ajoutant un attribut de type `Date`, pour récupérer la date courante. On effectue aussi le tri en fonction de la date à la création et à l'ajout d'une tâche.

4. Ajout de la bare de recherche

On ajoute un `UISearchBarDelegate` afin de remplir la fonction `searchBar ()`. Cette fonction sera appelé pendant l'écriture un élément à l'intérieur de la search Bar. On crée ainsi un tableau `dataRecherche` qui nous servira de nouveau tableau.

Problèmes rencontrés

- Disponibilité restreinte des salles de TP