

Youcef Bekkar
Hermann Kao
Hareish Jeyaku
27 septembre 2022

Compte rendu - What'sTheWeather

Ce compte rendu est écrit dans le cadre d'une projection dans notre futur carrière au sein de l'entreprise What'sTheWeather et une réflexion à notre potentiel futur au sein de celle-ci.

Dans cette optique nous verrons quels seront nos objectifs de carrière au sein de ladite entreprise et les étapes envisagées pour atteindre ces objectifs. Étant un PPP, ce compte rendu sera assez personnel centré sur nos qualités, nos défauts, nos espérances ainsi que nos visions; tout ceci appuyé sur notre expérience de développement de l'application météo.

Le compte rendu du projet sera expliqué en trois grandes étapes; la préparation, la réalisation du projet et enfin le rendu final.

La préparation

Dans un premier temps le groupe à été créé par Hareish qui a mis en relation Youcef et Hermann, après la création d'un groupe de discussion nous nous fait une mise au point sur nos compétences; notre savoir-être; nos qualités propres, nos défauts également, nos limites potentielles, notre limite absolue et enfin nos goûts et intérêts. Ensuite nous avons vu quelles étaient nos savoirs, nos connaissances acquises jusqu'aujourd'hui, nos formations, certifications, expériences professionnelles, projets d'écoles, projets personnels; et enfin quels étaient nos savoir-faire. Pour ma part on a vu que j'avais déjà fait une licence Maths - Info donc j'ai déjà codé en Java et C, ce qui m'as assez facilité en Swift qui est de la

programmation objet compilé, de plus j'avais fait le projet cat-tinder pour l'ETNA en Xcode Swift (no-code). Concernant Hermann et Hareish, ils partaient de zéro.

A la suite de cette mise au point, nous avons décidé tous les trois de faire le projet en langage Swift.

Étrangère pour tout les trois, on a donc dû, pour se préparer à ce projet, se former à cette technologie. Le choix de ce langage à été motivé de part sa vitesse de compilation et d'exécution, du fait qu'il est open source et pour le côté sécurité que Apple a de base.

La préparation et formation personnelle étant faite, le projet à pu débuter sans encombre.

L'appli étant scindé en 3 écrans, le travail a donc été repartit de telle sorte a ce que chacun fasse une partie soit un écran.

La réalisation du projet

Pour se lancer dans la réalisation du projet, on s'est entouré principalement de la doc Xcode, Swift et l'API openweathermap, des formation gratuite en ligne de création d'application mobile, mais aussi notre réseau de connaissance. En discutant avec des master de l'ETNA, et des personnes compétentes en Swift dans nos entreprises respectives on a pu avancer dans cette réalisation. Malgré ces personnes compétentes et mes recherches, je n'arrivait toujours pas à solutionner mon problème de géolocalisation ... L'application d'Hermann elle contient la géolocalisation donc je ne m'étais pas trop attardé sur mon erreur et ai plutôt demandé à Hermann comment sa géolocalisation fonctionnait. En outre de tout cela, en utilisant Xcode, nous avons vu que le no code était possible, et qu'en faisant la conception de l'application nous avons vu que le code Swift était auto-généré. À la suite de ceci Hermann à suggéré de ne pas utiliser le no code, pour pouvoir bien expliqué le projet dans les détails lors de la soutenance. Sa suggestion à été adoptée.

Les points travaillés pour le développement de l'application ont été UI/UX notamment pour le front; à savoir le placement des boutons et l'organisations des icônes.

La réalisation du projet - Conclusion

Dans un premier temps nous avons étudié le projet « What's The Weather » dans le design et technique. Nous avons décidé de faire iOS (Swift) car Le langage Swift a été conçu pour iOS et les autres systèmes d'exploitation des appareils Apple ainsi que pour Linux.

C'est un Langage open-source. Il s'agit d'une technologie ouverte pour tous les programmeurs du monde entier. Pour la sécurité notre application le codage en Swift offre une cohérence et une sécurité accrue. En raison des mesures de protection ajoutées, la prévention des erreurs devient beaucoup plus facile et la lisibilité s'améliore considérablement.

La Vitesse est plus de deux fois plus que l'Objective-C. Si l'on compare avec [Python](#), la vitesse est plus élevée. Si on le compare à [Python](#), la différence est encore plus radicale – il est 8,5 fois plus rapide. Nous avons privilégié cette technologie pour démarrer notre application « What's The Weather ».

Nous avons utilisé Swift UI comme framework pour créer notre application, c'est plus simple comme méthode pour développer notre application.

Le choix de Design pattern était très important pour prendre la direction pour notre Groupe. Nous avons fait le choix de faire Design pattern MVVM cela nous a permis des modèles de conception sont des solutions réutilisables aux problèmes courants du code.

Pour un expérience utilisateur efficace et rendant son interface facile d'utilisation, nous avons travaillé sur les points UI/UX pour développeur notre application :

- Structure de notre application
- Design visuel
- Bouton d'action

Pour mieux répartir le travail nous avons organisé le travail de groupe en découpant plusieurs étapes:

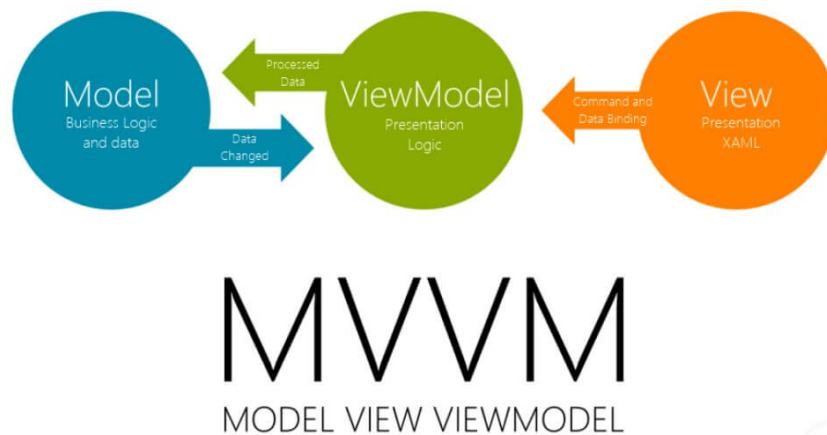
- Prise en main d'outil (Xcode, Swift UI).
- Découpages des écrans pour que chaque membre de l'équipe peut se concentrer sur le travail à réaliser.
- Liste des fonctionnalités pour chaque écran .
- Assemblages de travail chaque personne pour regrouper de travail.

Concernant les API, cela à été réparti comme suit :

- Auto-complétion des ville : Api Teleport
- Météo : Api Openweathermap

Concernant la géolocalisation, elle est dû à un module natif de chez Apple.

Afin de visualiser l'architecture on a le schéma suivant :



Cette architecture est un design pattern qui vise à séparer la logique de présentation d'une app en trois couches :

- Model
- View
- ModelView

Ici le ViewModel reçoit les Maj de la View, traite les données et renvoie les Maj à la View.

Pour notre app cela ressemblerait plutôt à ça :

.

- └── app
 - └── models - protocoles et struc definissant les types de données
 - └── ressources - images et medias
 - └── stores - gestion du state
 - └── styles - styles custom
 - └── views - tout ce qui est visuel
 - └── components - composants des ecrans
 - └── screens - ecrans

Pour conclure, ce projet à été très bénéfique pour nous trois qui travaillons sur Mac, et donc sous MacOS, elle nous a permis de nous familiariser avec le langage Swift, et nous a même séduit.