

Créez l'application Reciplease...

pour de bons petits plats !

Énoncé

La journée a été bien chargée. Vous avez avalé des heures de code, trois cours, discuté avec votre mentor, sorti votre chien et poncé Stack Overflow à cause de ce [\\$!%&! de bug](#). Il est déjà l'heure de préparer le dîner. Vous ouvrez votre frigo et là c'est la panne d'inspiration, le vide quantique, le néant abyssal. Et cette éternelle question dans votre tête :

Qu'est-ce que je vais bien pouvoir manger ce soir ?

Cette question existentielle, vous réalisez que vous vous la posez bien trop souvent. Et en bon programmeur, vous vous dites, il est temps d'être [DRY](#) (Don't Repeat Yourself). Vous prenez donc la décision de ne plus JAMAIS vous ennuyer avec cet épineux problème !

Et quelle est la réponse à tous les problèmes de la vie ?

Une app ?!!

Evidemment ! C'est donc armé d'une pomme et de votre mac préféré que vous vous lancez dans le développement de Reciplease...

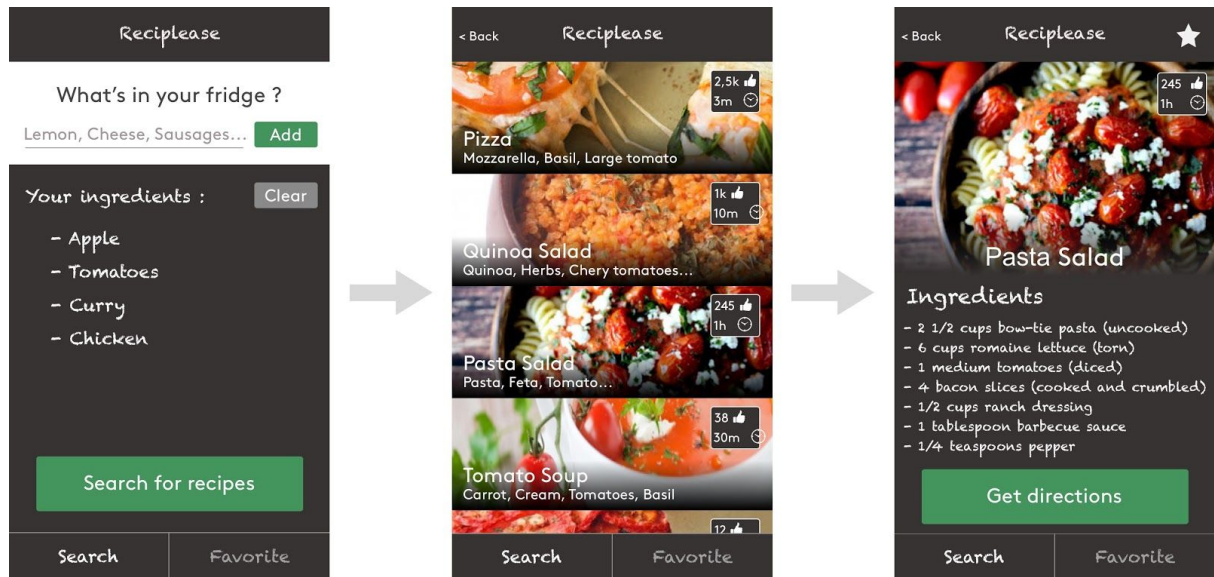
Description générale

Reciplease permet de rechercher des recettes à partir d'ingrédients. On peut également conserver ses recettes préférées en favoris.

L'application est donc divisée en deux parties : la recherche et les favoris. On peut naviguer entre les deux parties grâce à une barre d'onglets ("Tab bar").

1. La recherche

Voici comment se présente la recherche :



1/ L'utilisateur choisit ses ingrédients :

- Un champ de texte lui permet de rajouter ses ingrédients les uns après les autres.
- L'appui sur le bouton "Add" rajoute un ingrédient à la liste.
- L'appui sur le bouton "Clear" supprime tout les ingrédients de la liste.
- L'appui sur le bouton "Search for recipes" lance la recherche de recettes.

Si une erreur a lieu lors de la recherche, pensez à alerter l'utilisateur ! De plus, pendant la recherche, notifiez l'utilisateur que la recherche est en cours avec une `UIActivityIndicatorView`.

2/ Les résultats de la recherche sont affichés dans une table view

Chaque élément de la liste contient au moins :

- Le titre de la recette
- L'image si elle est présente (mettre une image par défaut sinon)
- La liste des ingrédients
- La durée d'exécution de la recette si elle est présente
- La note si elle est présente

En sélectionnant un élément de la liste, l'utilisateur accède au détail de cette recette.

Pour obtenir les éléments nécessaires au bon affichage de la vue "détail", vous allez devoir effectuer une requête supplémentaire. Comme pour le point précédent, alertez l'utilisateur en cas d'erreur et utilisez une `UIActivityIndicatorView` pendant la recherche.

3/ Le détail de la recette

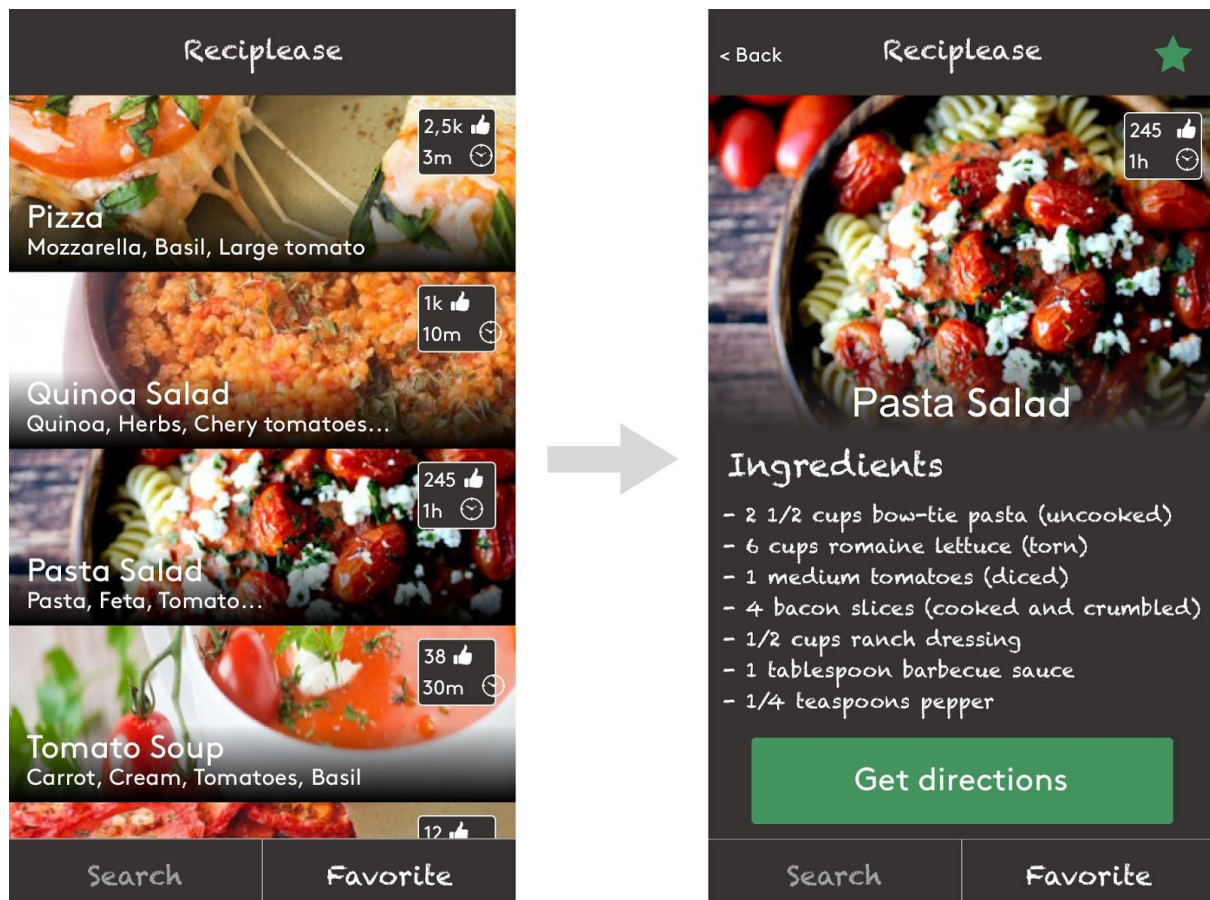
La page de détail de la recette doit contenir au moins :

- Le titre de la recette
- La liste complète des ingrédients avec le détail des portions
- L'image si présente (mettre une image par défaut sinon)
- La durée d'exécution de la recette si présente
- La note si présente
- Un bouton pour accéder à la liste détaillée des instructions

Sur cette page, un bouton permet de sauvegarder la recette dans ses favoris.

2. Favoris

Voici comment se présentent les favoris :



Les interfaces sont très similaires à la section recherche. Les seules différences sont :

- La liste n'est composée que des favoris (évidemment)
- Si la liste est vide, laissez un message à l'utilisateur pour lui expliquer comment la remplir.
- Dans la vue de détail, le bouton pour mettre en favori est sélectionné. Si l'utilisateur tape à nouveau dessus, la recette ne fait plus partie des favoris.

Yummly

C'est bien beau tout ça ? Mais quels sont ces requêtes ? Où puis je trouver les recettes ?

Grâce à [yummly.com](https://www.yummly.com) ! Ce site combine les recettes de très nombreux sites de recettes. Et ils ont une [API](#) ! Vous allez pouvoir puiser dans leur gigantesque base de données !

L'utilisation de l'API est payante pour un usage commercial. Ils proposent un essai de 14 jours que vous pouvez étendre en tant qu'étudiant. Donc ne vous privez pas et postulez tout de suite pour un compte étudiant (Recherchez "Academic Plan").

Vous allez utiliser la fonction de recherche (`Search Recipe`). Celle-ci a plusieurs options, celle qui nous intéresse ici est la recherche par ingrédient.

Pour avoir les détails d'une recette en particulier il vous faudra utiliser la fonction `Get Recipe`.

La recherche de recettes fonctionne mieux en anglais. Pour inciter l'utilisateur à utiliser l'anglais, je vous suggère de designer votre application directement en anglais.

Lorsque l'utilisateur veut accéder aux instructions de la recette via le bouton `Get directions`, vous n'avez qu'à le rediriger vers le site web où se trouve la recette en ouvrant Safari à la bonne page. La fonction `Get Recipe` vous renvoie l'URL du site web.

Design

Je vous ai fourni au dessus un exemple de design de l'application. Ce n'est qu'un exemple. Vous êtes libre de faire le design de votre choix

tant que les instructions ci-dessus sont respectées. Bien sûr, comme d'habitude, votre design devra être propre et clair sur toutes les tailles d'écran et dans au moins une orientation.

Contraintes

- Le code est sur **GitHub** avec un historique de commits cohérent.
- Le code est **clair et lisible**.
- Le code est écrit en **anglais** : commentaires, variables, fonctions...
- Le projet ne contient **aucun warning ni erreur**.
- Vous avez respecté le modèle **MVC** scrupuleusement.
- Votre application doit contenir **des tests unitaires** qui couvriront la majorité de la logique de votre code.
- L'application doit s'afficher correctement sur **toutes les tailles d'iPhone en mode portrait**.
- L'application doit supporter iOS 11 et les versions supérieures.
- Vous utiliserez `Alamofire` pour vos requêtes et vous l'installerez en utilisant `Cocoa Pods`.
- Vous sauvegarderez les recettes en favoris en utilisant Core Data.

Livrables

Pour ce projet vous reproduirez le schéma du cours sur les MVCs pour expliquer comment les différents MVCs de votre application discutent entre eux.

Vous fournirez donc un dossier .zip contenant :

- Le projet Xcode
- Un fichier PDF décrivant brièvement le bonus que vous avez choisi et la façon dont vous l'avez intégré.

Soutenance

La présentation de votre projet : 15-20 minutes

Un moment questions/réponses avec l'évaluateur : 10 minutes

Compétences à valider

- Comprendre et utiliser le delegate pattern
- Sauvegarder des données avec Core Data
- Gérer une Table View
- Utiliser une librairie open source