



Reciplease

Bonus

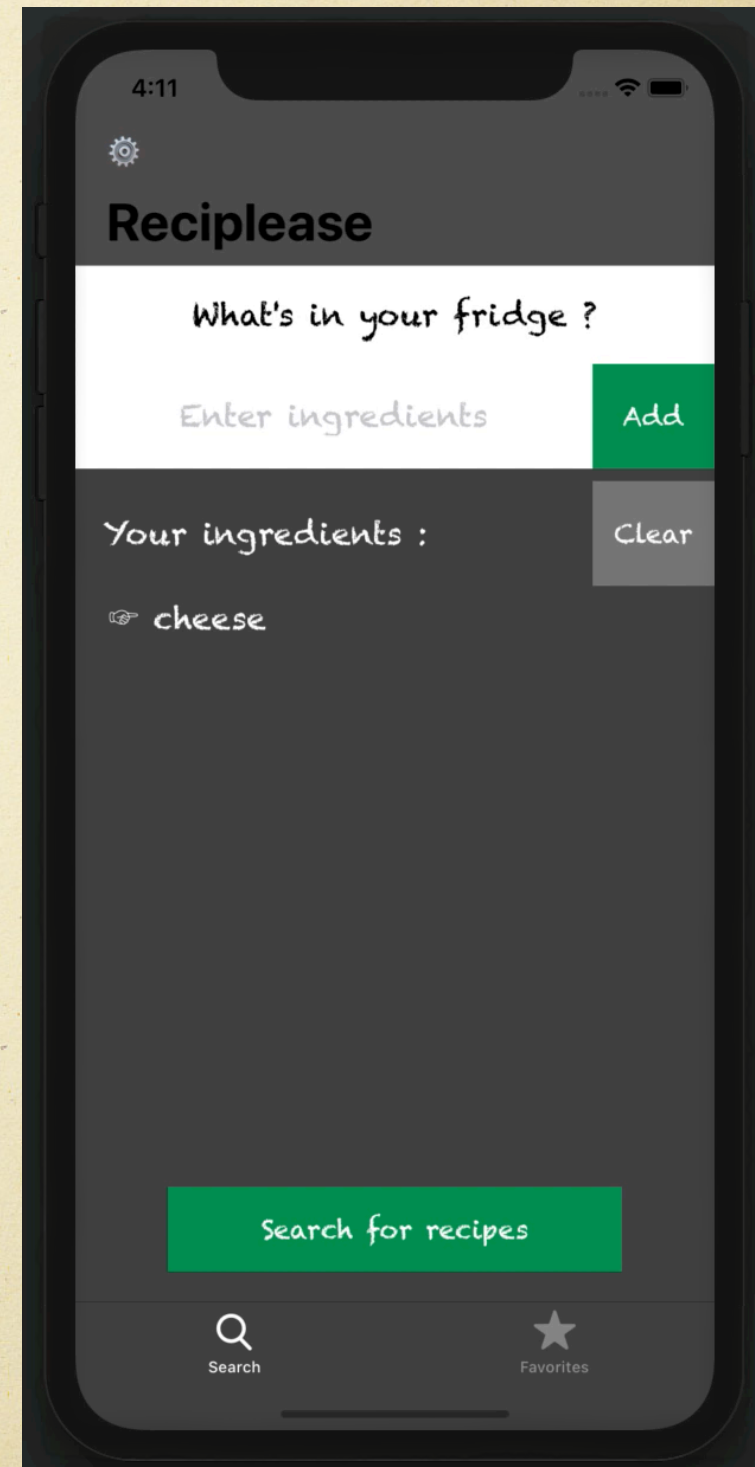
Reciplease est une application de recherche de recettes par ingrédients

- Pour affiner les recherches de recettes et avoir une interface utilisateur plus agréable à utiliser, j'ai intégré les bonus suivants:
- Une animation pour les tableView qui font apparaitre la liste des recettes.
- Une vue paramètres qui permet d'affiner la recherche des recettes en fonction des allergies auxquelles on peut être sujet.



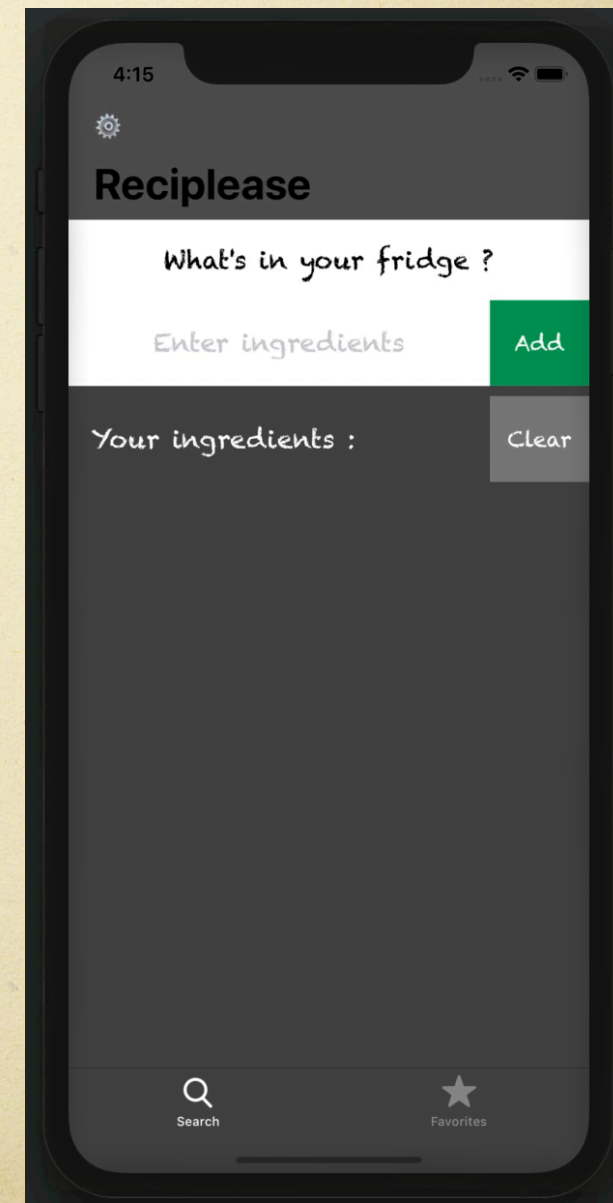
La tableView animée

```
func tableView(_ tableView: UITableView, willDisplay cell: UITableViewCell,
forRowAt indexPath: IndexPath) {
    cell.transform = CGAffineTransform(scaleX: 0.8, y: 0.8)
    UIView.animate(withDuration: 0.4) {
        cell.transform = CGAffineTransform.identity
    }
}
```



Le choix des allergies

- La vue paramètre qui permet de choisir les allergies est une present modaly qui grâce à des switches, rajoute une ou plusieurs allergies dans la requête de recherche des recettes.
- Ces réglages sont sauvegardés avec UserDefaults et à chaque fois que la vue apparait, les switches sont mis à jour en fonction des allergies que l'on a choisi.



Sauvegarde des paramètres dans la classe SettingService avec UserDefaults

```
class SettingService {  
    private struct Keys {  
        static let eggAllergy = "eggAllergy"  
        static let glutenAllergy = "glutenAllergy"  
        static let peanutAllergy = "peanutAllergy"  
    }  
  
    static var eggAllergy: String {  
        get {  
            return UserDefaults.standard.string(forKey: Keys.eggAllergy) ?? ""  
        }  
        set {  
            UserDefaults.standard.set(newValue, forKey: Keys.eggAllergy)  
        }  
    }  
  
    static var glutenAllergy: String {  
        get {  
            return UserDefaults.standard.string(forKey: Keys.glutenAllergy) ?? ""  
        }  
        set {  
            UserDefaults.standard.set(newValue, forKey: Keys.glutenAllergy)  
        }  
    }  
  
    static var peanutAllergy: String {  
        get {  
            return UserDefaults.standard.string(forKey: Keys.peanutAllergy) ?? ""  
        }  
        set {  
            UserDefaults.standard.set(newValue, forKey: Keys.peanutAllergy)  
        }  
    }  
}
```


Choix des allergies dans le viewController synchronisées avec les switchs

```
// MARK: - Choice allergy with switch

private func switchEggAllergy() {
    guard eggAllergySwitch.isOn else {
        SettingService.eggAllergy = ""
        return
    }
    SettingService.eggAllergy = eggAllergy
}

private func switchGlutenAllergy() {
    guard glutenAllergySwitch.isOn else {
        SettingService.glutenAllergy = ""
        return
    }
    SettingService.glutenAllergy = glutenAllergy
}

private func switchPeanutAllergy() {
    guard peanutAllergySwitch.isOn else {
        SettingService.peanutAllergy = ""
        return
    }
    SettingService.peanutAllergy = peanutAllergy
}

// MARK: - Update status of switch every time the view appears with allergy saved

private func allergySwitchState(allergySaved: String, allergyValue: String,
                                selectSwitch: UISwitch) {
    guard allergySaved == allergyValue else {
        selectSwitch.isOn = false
        return
    }
    selectSwitch.isOn = true
}
```