

# DESIGN PATTERNS

## A. SINGLETON

Le design pattern singleton permet de faire en sorte qu'une classe n'a qu'une seule instance.

### **Application**

Le singleton : utilisé dans la partie « Model » dans le MVC puisqu'une seule instance est nécessaire pour les données à afficher.

## B. STRATEGY

Le design pattern strategy permet de définir une famille d'algorithmes, de placer chacun d'eux dans une classe distincte.

### **Application**

Le strategy : utilisé dans le menu, puisqu'il y a plusieurs des recettes pour les entrées, les plats et les desserts. En outre, le design pattern strategy permettra de choisir les recettes en fonction des ingrédients.

## C. OBSERVER

Le design pattern observer : Il définit le mécanisme de souscription pour notifier à plusieurs objets tout événement survenant dans l'objet qu'ils observent.

### **Application**

L'observer s'utilisera dans la partie « vue » dans le MVC et chaque fois qu'une modification est présentée sera notifiée et une mise à jour sera effectuée dans le système.

## D. FAÇADE

La façade fournit une interface unifiée à un ensemble d'interfaces d'un sous-système. Façade définit une interface de niveau supérieur facilitant l'utilisation du sous-système.

### **Application**

La façade s'utilisera pour interagir d'une façon moins complexe avec la classe de la salle de restauration et de la cuisine et simplifier l'utilisation de leurs méthodes.

## E. BUILDER

Le design pattern builder : Il est utilisé pour la création d'une variété d'objets complexes à partir d'un objet source.

### **Application**

Le builder s'utilisera pour prendre les commandes. Les commandes sont un processus de construction qui peut être représentée en différents façons. En outre, les commandes sont constituées d'une entrée, d'un plat et d'un dessert.