**Détail et explication de tous les DP utilisés dans l’application**

* Decorator :(modèle de conception structurelle)

Un décorateur est le nom d'une des structures de patron de conception. Il permet d'attacher dynamiquement de nouvelles responsabilités à un objet. Les décorateurs offrent une alternative assez souple à l'héritage pour composer de nouvelles fonctionnalités.

* Abstract factory : (modèle de conception créative)

Le pattern Abstract Factory va nous permettre d’instancier des familles de produits dépendant les uns des autres sans qu’il soit nécessaire de préciser leur type concret. Elle fournit une interface pour créer des familles d'objets liés ou interdépendants sans avoir à préciser au moment de leur création la classe concrète à utiliser.

* Iterator : (modèle de conception comportementale)

Le modèle d'itérateur est un modèle de conception très couramment utilisé dans l'environnement de programmation Java et .Net. Ce modèle est utilisé pour obtenir un moyen d'accéder aux éléments d'un objet de collection de manière séquentielle sans avoir besoin de connaître sa représentation sous-jacente.

* Memento : (modèle de conception comportementale)

Capture et restaure l'état interne d'un objet. Le modèle memento est implémenté avec trois objets : le *créateur*, un *gardien* et un *souvenir*. Le créateur est un objet qui a un état interne. Le gardien va faire quelque chose à l'auteur, mais veut pouvoir annuler le changement. Le gardien demande d'abord à l'auteur un objet souvenir. Ensuite, il fait n'importe quelle opération (ou séquence d'opérations) qu'il allait faire. Pour revenir à l'état d'avant les opérations, il renvoie l'objet memento à l'expéditeur.

* State : (modèle de conception comportementale)

Il sert à représenter un automate d'état fini (ou graphe). C'est extrêmement simple en fait : il y a des états – les nœuds du graphe – et des évènements (ou transitions) – les arcs du graphe – qui permettent de passer d'un état à l'autre.

Reference : <https://sourcemaking.com/design_patterns>