Cash Manager Architectural Design

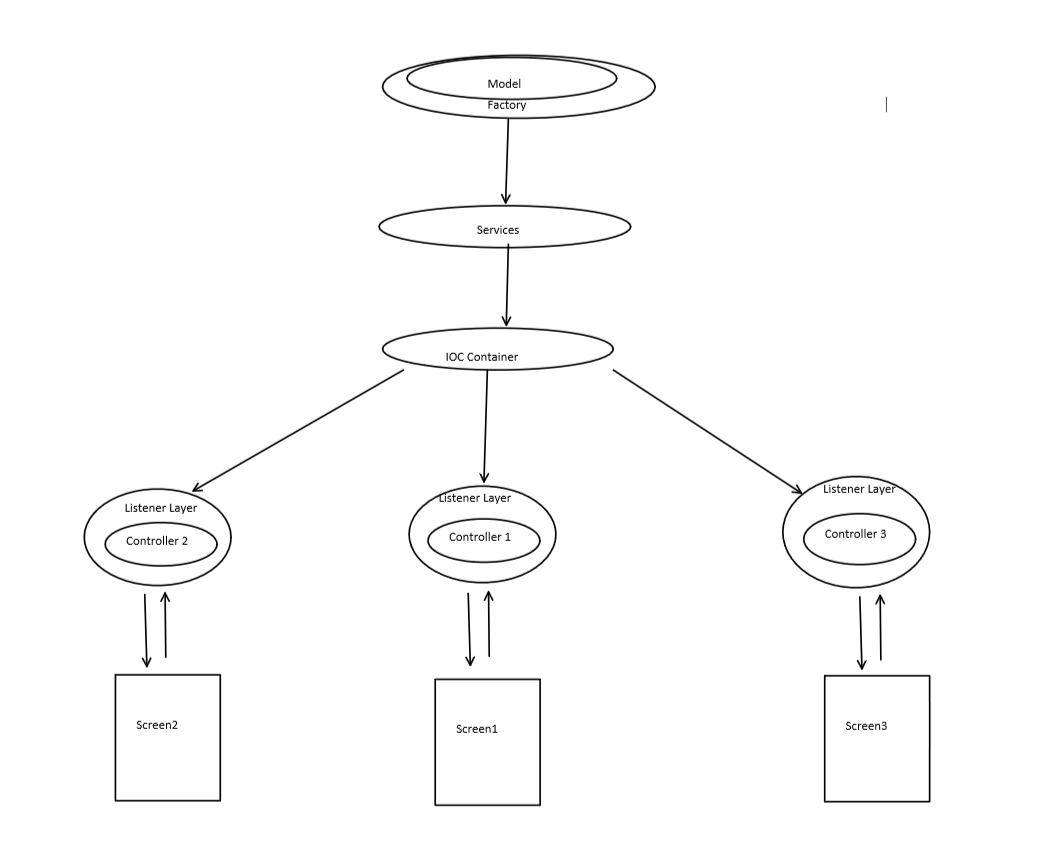


Fig1: Architecture du cash manager.

**Global architectural model:** MVC

**Détaille de chacun des composants :**

**« Screens » (représentent les vues)** :

* Responsable de la gestion de l’interaction entre l’utilisateur et l’application.
* Cree une représentation des donnes venant de l’application à l’utilisateur.
* Transmet les commandes aux contrôleurs.

**« Controlers » :**

* Recois les commandes venant de l’utilisateur à partir de la vue.
* Manipule les donnes avec une logique précise pour satisfaire l’utilisateur de l’application.
* Gere les autres composants de l’application et l’interaction qui existe entre elles.
* Grace a son implémentation du **« Listener Pattern »** Les contrôleurs peuvent communiquer entre eux pour mener à bien leurs fonctions de façon coordonnée.

**IOC Container:** (Inversion of Control Container)

* Responsable de l’injection de dépendances nécessaires au fonctionnement de l’application dans les contrôleurs.
* Gere la création d’instance de composants critiques aux fonctionnements de l’application.

**Services :**

* Responsable de l’Access au dépendances externes à l’application
* Gere l’Access a la base de données.
* Gere les paiements et la création de divers services consommée par l’application.

**Model :**

* Représentation des données utilisée par l’application.
* Grace aux **« Factory et au Builder Pattern »**, facilite la création, gestion et l’évolution de la structure des données consommées par l’application.

**NB :** The Singleton design pattern is used on the connection string.