UCAC | icam

MASTER CHEF INFO

PROJET PROGRAMMATION SYSTEME : GROUPE 7

Participant :

NGOUANA Ronny
FOUDA Anthony
BOUANGA Taviche
KEPYA Christian
SIPOUFO Loïc

Sous la supervision de :

Mr Odjong Humphrey

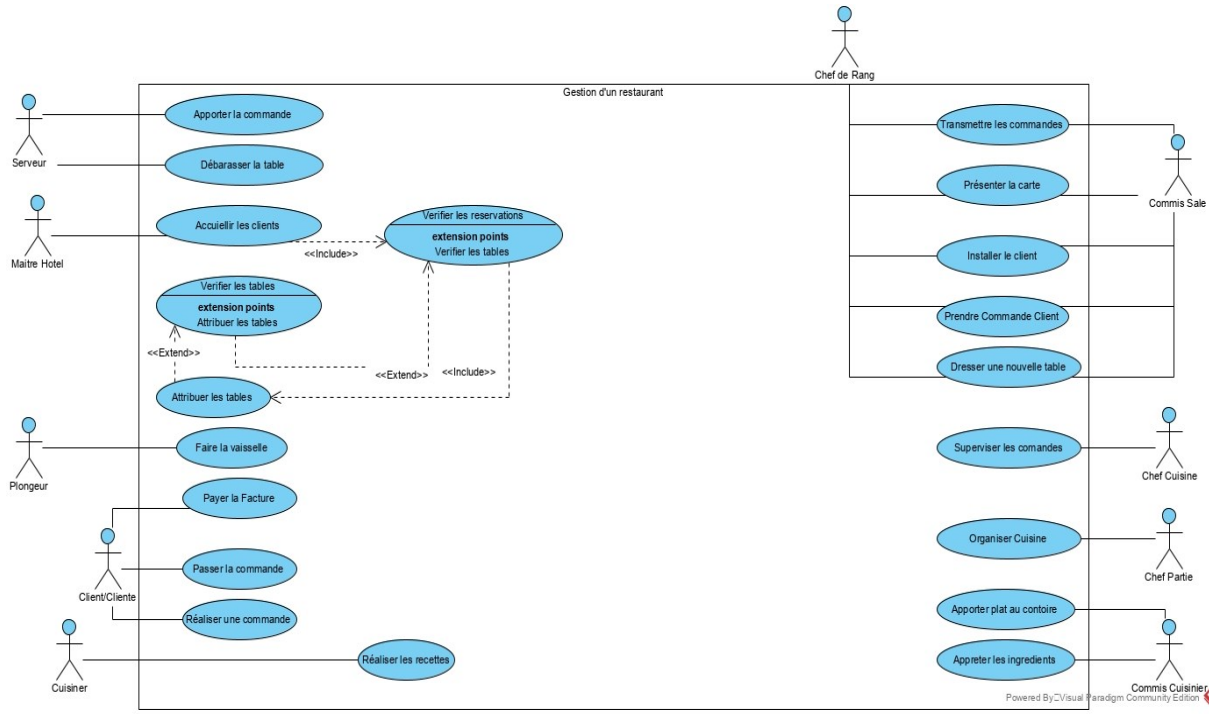
SOMMAIRE

I-DIAGRAMME

a-diagramme de cas d'utilisation

Un diagramme des cas d'utilisation UML résume certaines des relations entre les cas d'utilisation, des acteurs et des systèmes. Un diagramme des cas d'utilisation peut décrire les différents types d'utilisateurs d'un système et les différentes façons dont ils interagissent avec le système. Ce type de diagramme est généralement utilisé en conjonction avec le cas d'utilisation de données textuelles et sera souvent accompagné par d'autres types de diagrammes ainsi.

Ici il s question pour nous de faire une représentation du comportement fonctionnel de système et les interactions entre le système et nos différents acteurs. Ici nous essayé de mettre en évidence les actions et les fonctionnalités principale qu'aura notre système.

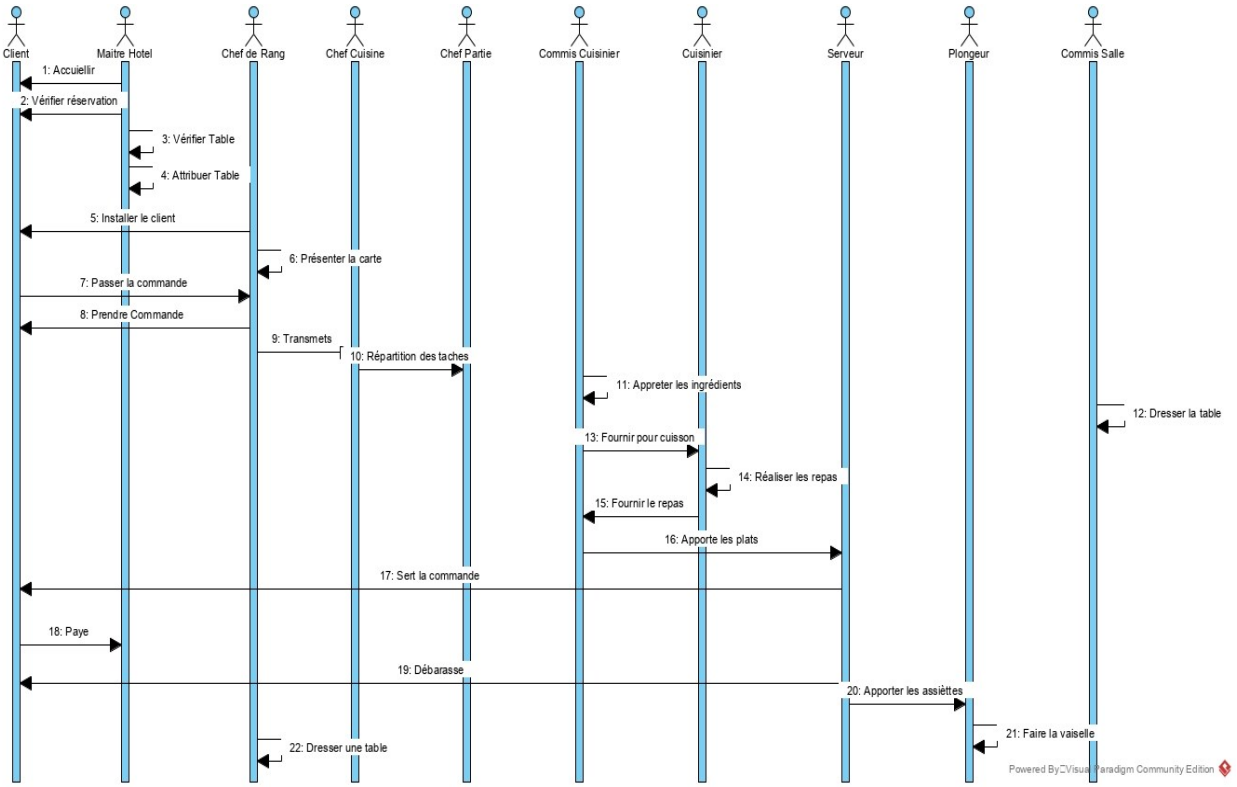


b-diagramme de séquence

Un Diagramme de Séquence est une forme de diagramme d'interaction, ce qui montre que les objets comme des lignes de sauvetage réduisant la page. Interactions représentées au fil du temps sont dessinées comme des connecteurs de message de la source Ligne de Vie à la Ligne de Vie cible. Les diagrammes de séquence sont bons à montrer les objets qui communiquent avec d'autres objets. Et quels sont les messages déclencher ces communications. Les diagrammes de séquence ne sont pas destinés à montrer logique procédurale complexe.

Ici nous essayons de schématiser comment notre système se comportera pendant son exécution, de schématiser l'ordre de séquençage et d'exécution de notre système. Là on peut constater qu'au

début de notre système le maitre hôtel accueille le client. Donc là nous présentons l'ordre chronologie de fonctionnement de notre système.



c-diagramme d'activité

Diagramme d'activité UML est un cas particulier d'un diagramme d'états dans lequel tous les états sont des États d'action et les transitions sont déclenchées par l'achèvement des actions dans l'état de la source. Utilisez un diagramme d'activité pour décrire le comportement interne d'une méthode et de représenter un flux entraîné par actions générées en interne.

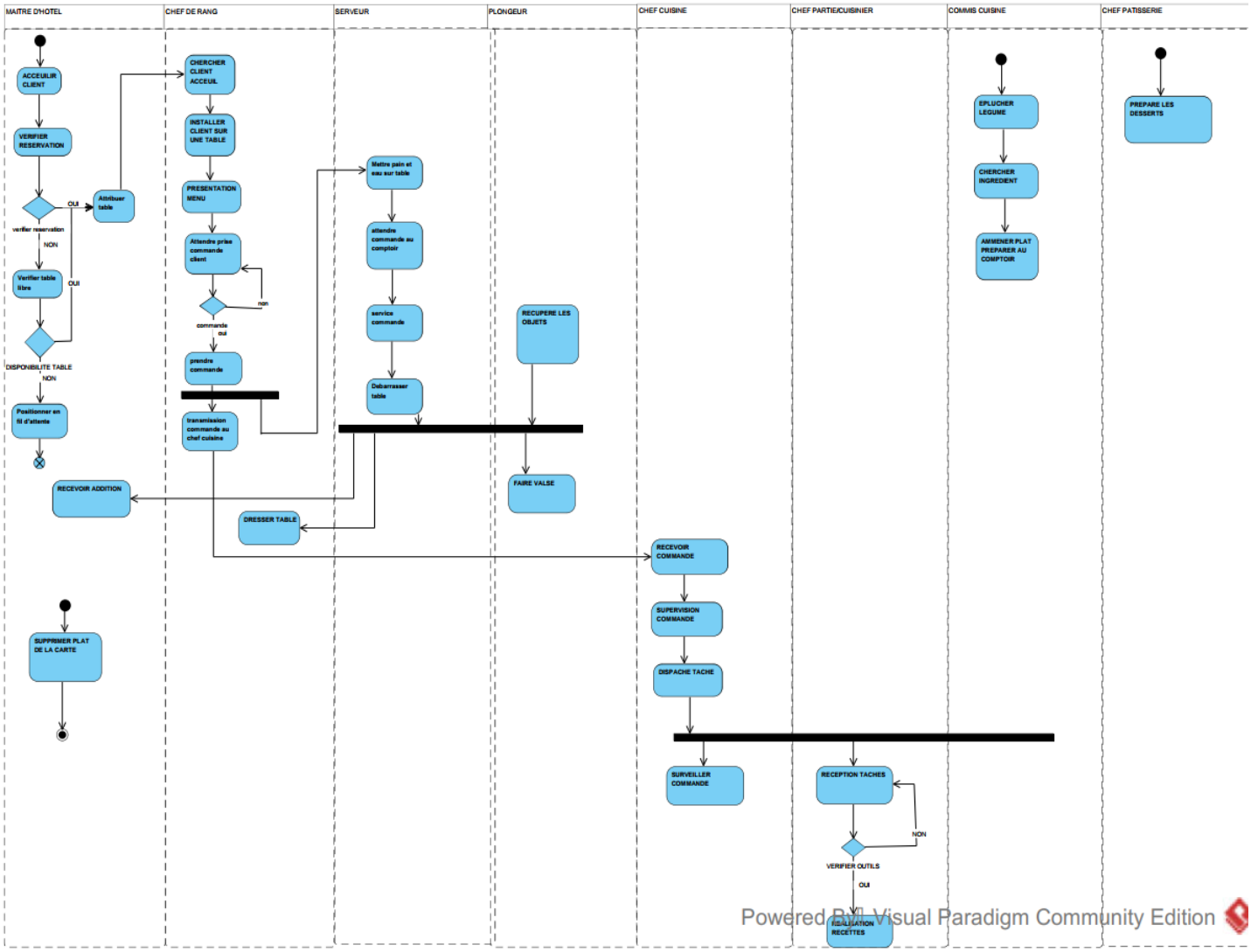
Dans notre diagramme, on peut énumère les diverses actions réaliser par les acteurs suivants :

- Le maitre d'hôtel accueil les clients, vérifie s'ils ont une réservation ou non. Décide de la table ou ils se mettront. Puis appelle le chef de rang pour qu'il puisse les accompagner.
- Le chef de rang quand il n'est pas occupé peut aller aider ses collègues en salle. En temps normal il accompagne les clients de l'entrée à leur table. Puis il donne les cartes au client et prend leur commande.
- Si le serveur n'est pas occupé il peut aider dans un autre carré. Il récupère les plats au comptoir puis les sert. Il réitère cette opération jusqu'à ce que des clients aient finis. Quand ils ont fini il ramasse les assiettes et les couverts, il les dépose dans la zone de stockage. Puis va redresser la table.

- Le chef de cuisine va recevoir les commandes puis les vérifier. Si jamais ces commandes sont validées et qu'elles sont réalisables on dispatche de façon optimal les commandes entres les différents chefs de partie. Si jamais elles ne sont pas validées, on supprime le plat du menu puis ont en informe le serveur.

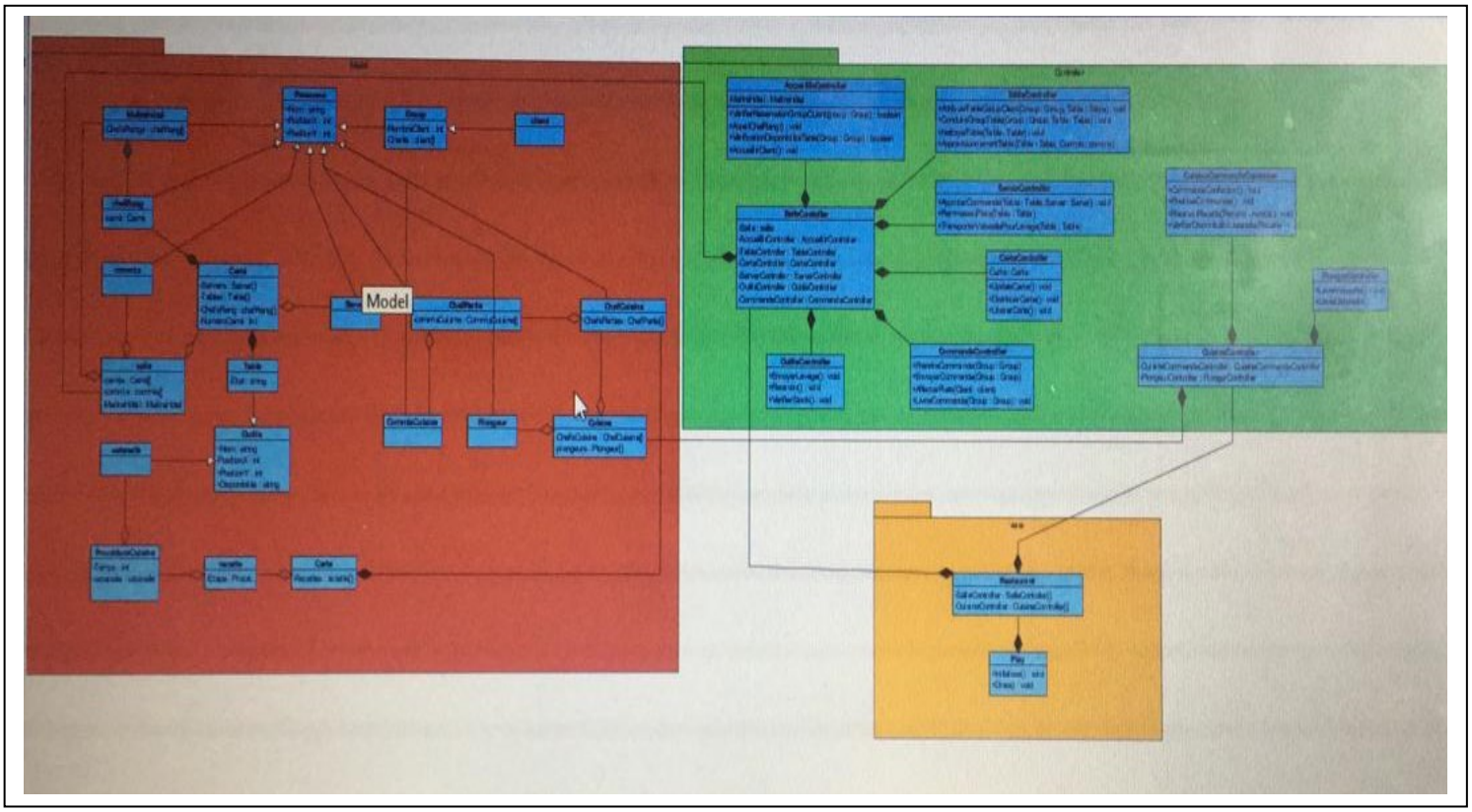
- Le chef de partie a pour rôle de récupérer l'ordre donné précédemment par le chef de cuisine. Si la main d'œuvre nécessaire est disponible on va alors ordonner la préparation du plat aux cuisiniers. Si jamais la main d'œuvre n'est pas disponible on laisse en file d'attente les ordres du chef.

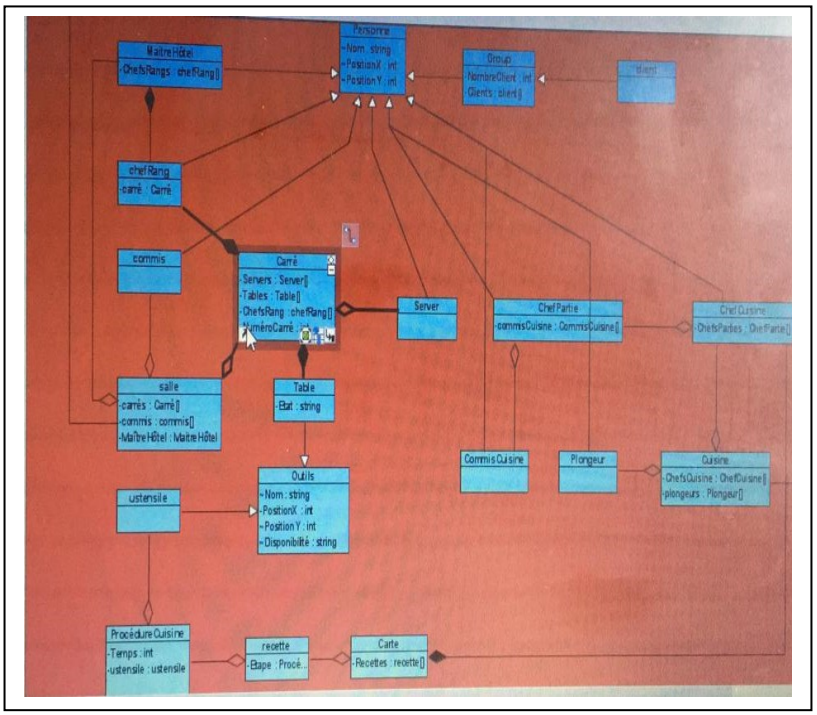
- Le commis de cuisine va récupérer un ordre du cuisinier. Si l'outil dont il a besoin est libre il va donc commencer la tache dont il est chargé. Si son outil n'est pas réutilisable il va le déposer dans la zone du « sale ». Si cet outil est assez propre pour être utiliser de nouveau il va donc le libérer suite à cela le commis de cuisine se charge d'amener les plats sur le comptoir.
- Le plongeur vérifie si les ustensiles sont sales, si oui ils les lavent puis les ramènent.

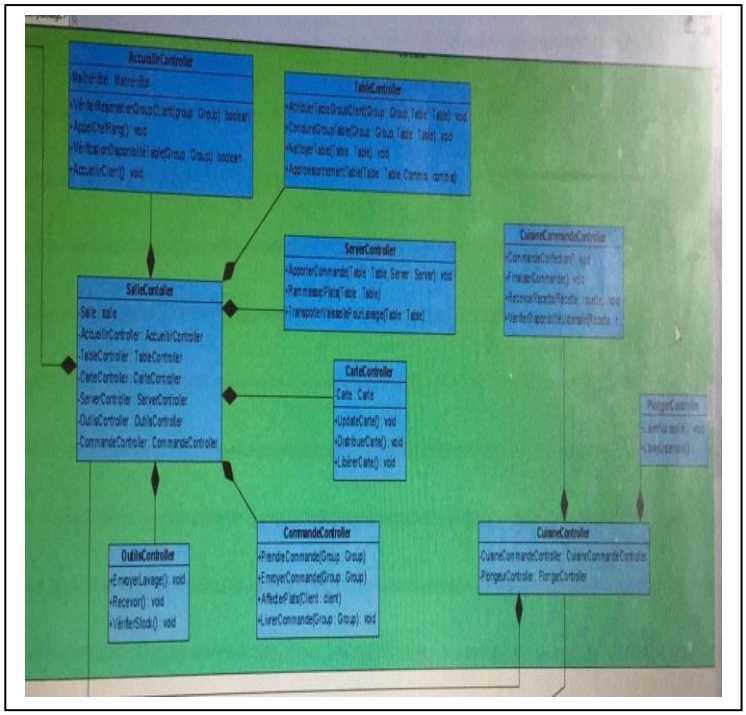


d-diagramme de classe

Le diagramme de classes est l'un des types les plus populaires en langage UML. Très utilisé par les ingénieurs logiciels pour documenter l'architecture des logiciels, les diagrammes de classes sont un type de diagramme de structure, car ils décrivent ce qui doit être présent dans le système modélisé.

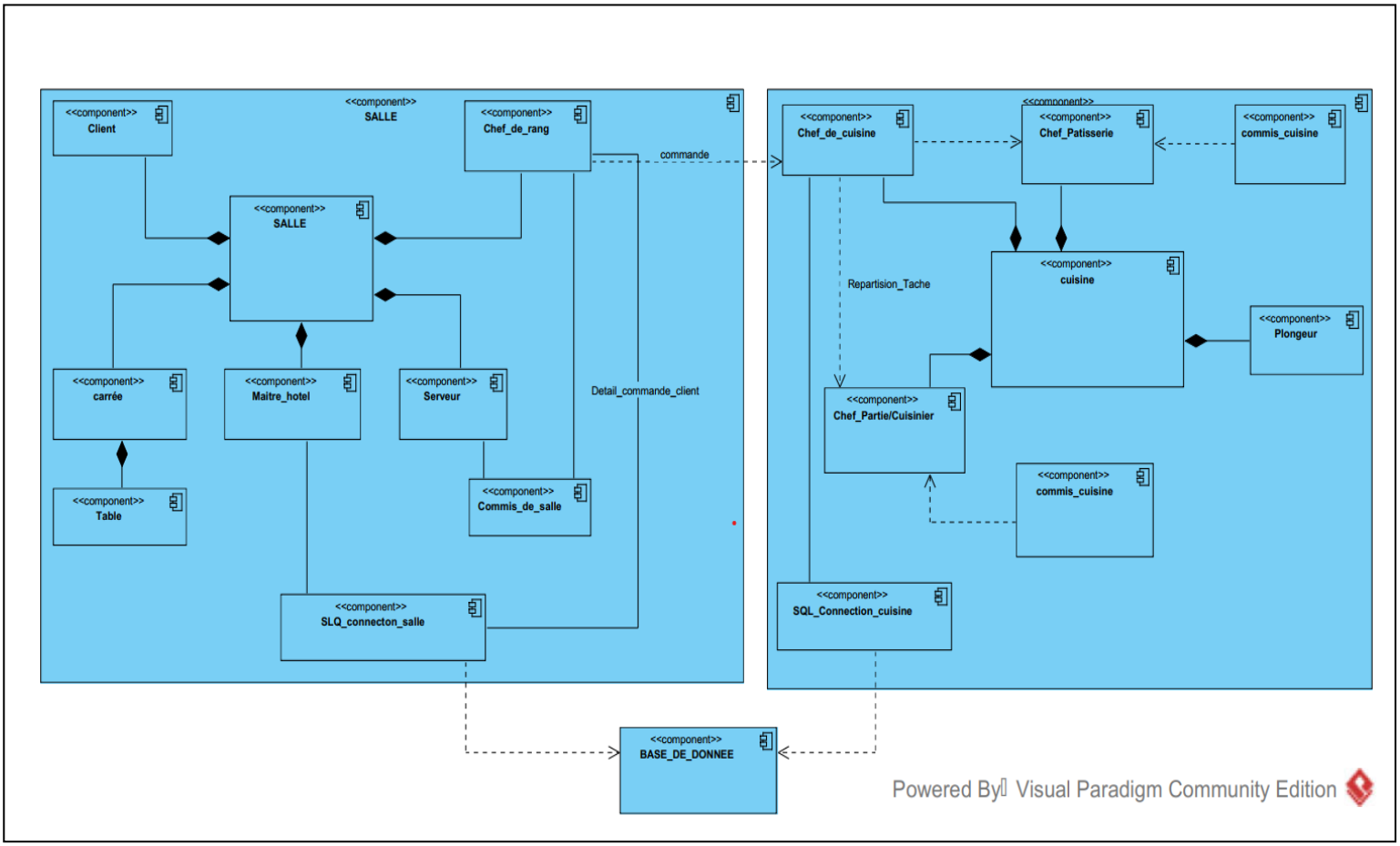






e-un diagramme de composants

Un diagramme de composants a pour objectif d'illustrer la relation entre les différents composants d'un système. Dans le cadre de l'UML, le terme « composant » fait référence à un module de classes qui représentent des systèmes ou des sous-systèmes indépendants ayant la capacité de s'interfacer avec le reste du système Diagramme de Composant. Les diagrammes de composants UML permettent de simplifier les processus, même les plus complexes.



Le but est de décrire les choix d'implémentation et les dépendances de compilation et d'implémentation entre les composants du système.

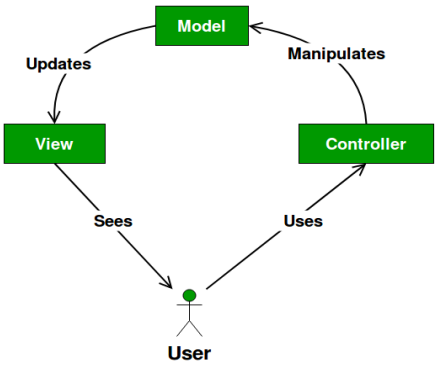
Ⅱ-DP (Design Patterns) -MVC

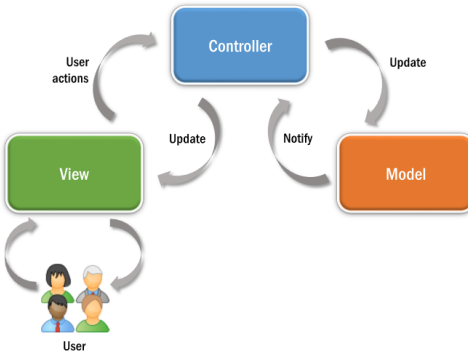
1-MVC

Le modèle de conception MVC (Model View Controller) spécifie qu'une application se compose d'un modèle de données, d'informations de présentation et d'informations de contrôle. Le modèle exige que chacun d'eux soit séparé en différents objets. MVC est plus un modèle architectural, mais pas pour une application complète. MVC concerne principalement la couche d'interface utilisateur / d'interaction

d'une application. Vous aurez toujours besoin d'une couche de logique métier, peut-être d'une couche de service et d'une couche d'accès aux données. Il est composé de trois types de modules ayant trois responsabilités différentes à savoir :

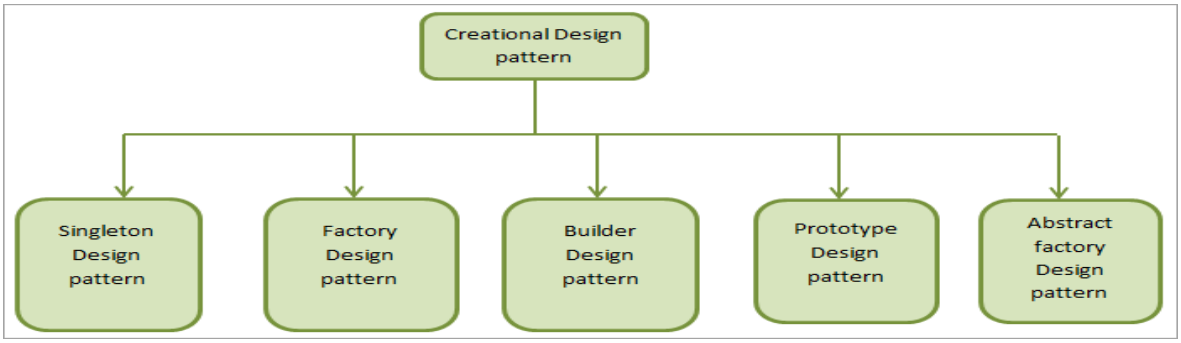
- Un modèle (Model) : les objets modèles sont les parties de l'application qui implémentent la logique du domaine des données de l'application. Souvent, ils récupèrent l'état du modèle et le stockent dans une base de données.
- Une vue (View) : les vues sont les composants qui affichent l'interface utilisateur de l'application (interface utilisateur). En général, cette interface utilisateur est créée à partir des données du modèle.
- Un contrôleur (Controller) : Les contrôleurs sont les composants qui gèrent les interventions de l'utilisateur, exploitent le modèle et finalement sélectionnent une vue permettant de restituer l'interface utilisateur. Dans une application MVC, la vue affiche uniquement des informations ; le contrôleur gère les entrées et interactions des utilisateurs, et y répond.





Dans la réalisation de notre application, le Design pattern MVC est utilisé dans le cadre structurel de l'application. Il permet de séparer les différents composants pour l'évolutivité du programme.

On peut repartir les divers éléments du Design patterns comme suite



2-observer

Un modèle de conception d'observateur est utile lorsque vous êtes intéressé par l'état d'un objet et que vous souhaitez être averti en cas de changement. Dans le modèle observateur, l'objet qui surveille l'état d'un autre objet est appelé Observateur et l'objet observé est appelé Sujet.

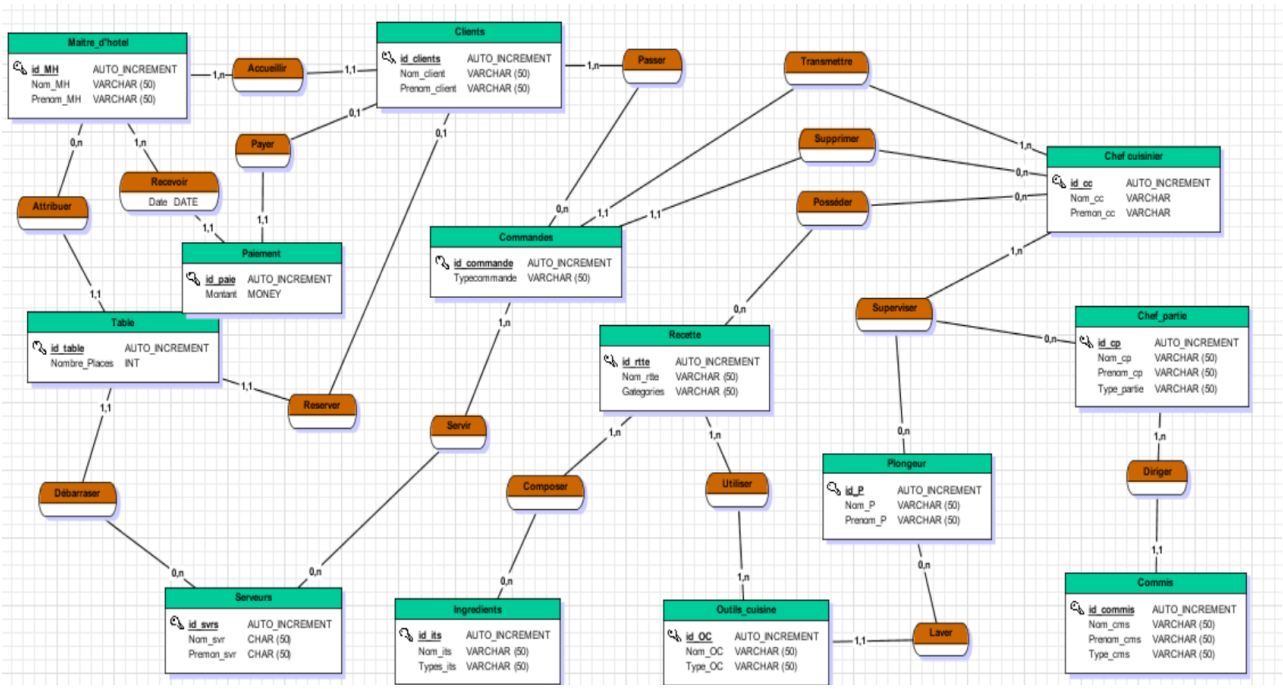
Ici nous allons l'utiliser dans la surveillance et la supervision des éléments dans le programme, par exemple la gestion de l'arrivée d'un nouveau client est gérée par un observer.

3-singleton

Le modèle Singleton restreint l'instanciation d'une classe et garantit qu'une seule instance de la classe existe. Cela semble être un modèle de conception très simple, mais en ce qui concerne la mise en œuvre, il comporte de nombreux problèmes de mise en œuvre. On implémente le singleton en écrivant une classe contenant une méthode qui crée une instance uniquement s'il n'en existe pas encore. Sinon elle renvoie une référence vers l'objet qui existe déjà.

Dans notre cas, on l'utilisera dans la mesure où il n'y a qu'une seule instance de l'objet qui doit exister au moment de l'exécution. Mais également il sera utilisé pour le DP factory.

Ⅲ-MDC



La table centrale de notre MCD est celle des recettes. Celle-ci dispose de colonnes permettant de catégoriser, de différencier selon leur type, leur nom nous permettant de définir si une recette est un plat ou une entrée ou un dessert. Enfin d'être mise sur pied ces recette nécessitent des ustensiles connus sous le nom d'outils cuisine. Ces outils cuisine englobe les ustensiles tels que les couverts. En

plus des ustensiles de cuisines nous allons utiliser aussi des ingrédients pour la mise sur pied des différentes recettes d'où nous avons créé une table à cet effet. De plus pour connaitre les ingrédients utilisés par un recette il y a une table intermédiaire entre celle des ingrédients et celle des recettes. Celle si contient l'identifiant des recettes, celui des aliments ainsi que la quantité nécessaire à la recette. Les commandes sont reliées aux tables qui sont elles-mêmes reliées aux carrés. Une commande comprend l'identifiant de la table et ceux des recettes commandées. En plus nous avons à l'intérieur de notre MCD des tables permettant de mettre en exergue les interactions entre les différents acteurs du système.