

05/12/2018

# Dossier de conception .NET

RIVIERE Clément  
LAURAS Baptiste  
NERAUD Yohann  
BARRE-PITOUS Thomas

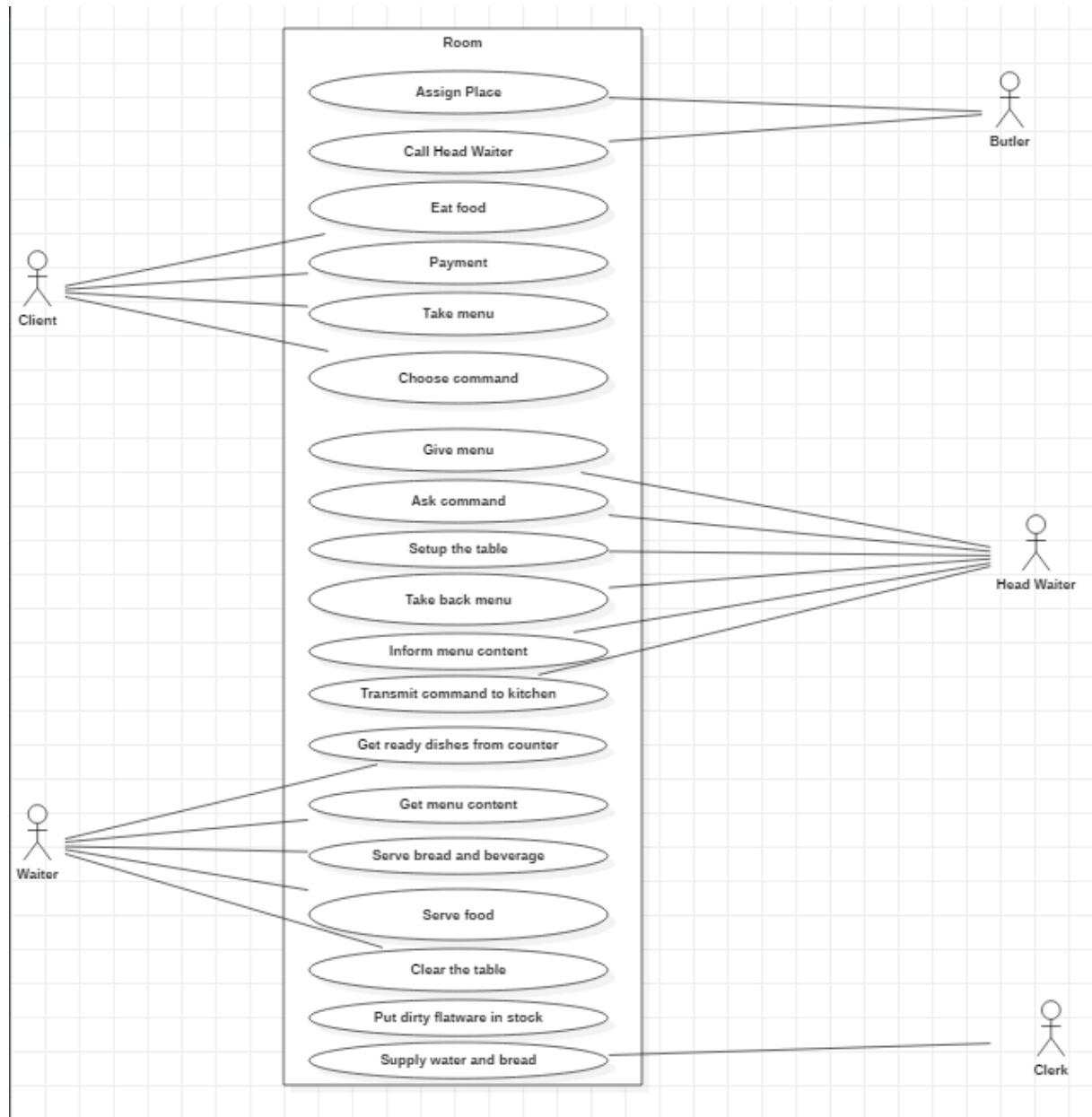
## Table des matières

<b>I. MODELISATION UML .....</b>	<b>3</b>
DIAGRAMMES DE CAS D'UTILISATION.....	3
SALLE .....	3
CUISINE .....	5
DIAGRAMMES D'ACTIVITE.....	7
MAITRE D'HOTEL .....	7
CHEF DE RANG .....	8
SERVEUR .....	9
COMMIS DE SALLE .....	10
CHEF CUISINIER.....	11
CUISINIER .....	12
COMMIS DE CUISINE .....	13
PLONGEUR.....	14
DIAGRAMMES DE CLASSES ET COMPOSANTS .....	15
CLASSES .....	15
COMPOSANTS .....	16
DIAGRAMMES DE SEQUENCE .....	17
PARTIE SALLE .....	17
PARTIE CUISINE .....	19
<b>II. DESIGNS PATTERNS.....</b>	<b>21</b>
DESIGN PATTERN SINGLETON .....	21
DESIGN PATTERN BRIDGE .....	22
DESIGN PATTERN OBSERVER .....	23
DESIGN PATTERN STRATEGY .....	24
DESIGN PATTERN BUILDER.....	25
<b>III. BASE DE DONNEES .....</b>	<b>26</b>
REALISATION DU MCD : .....	26

# I. Modélisation UML

Diagrammes de cas d'utilisation

Salle

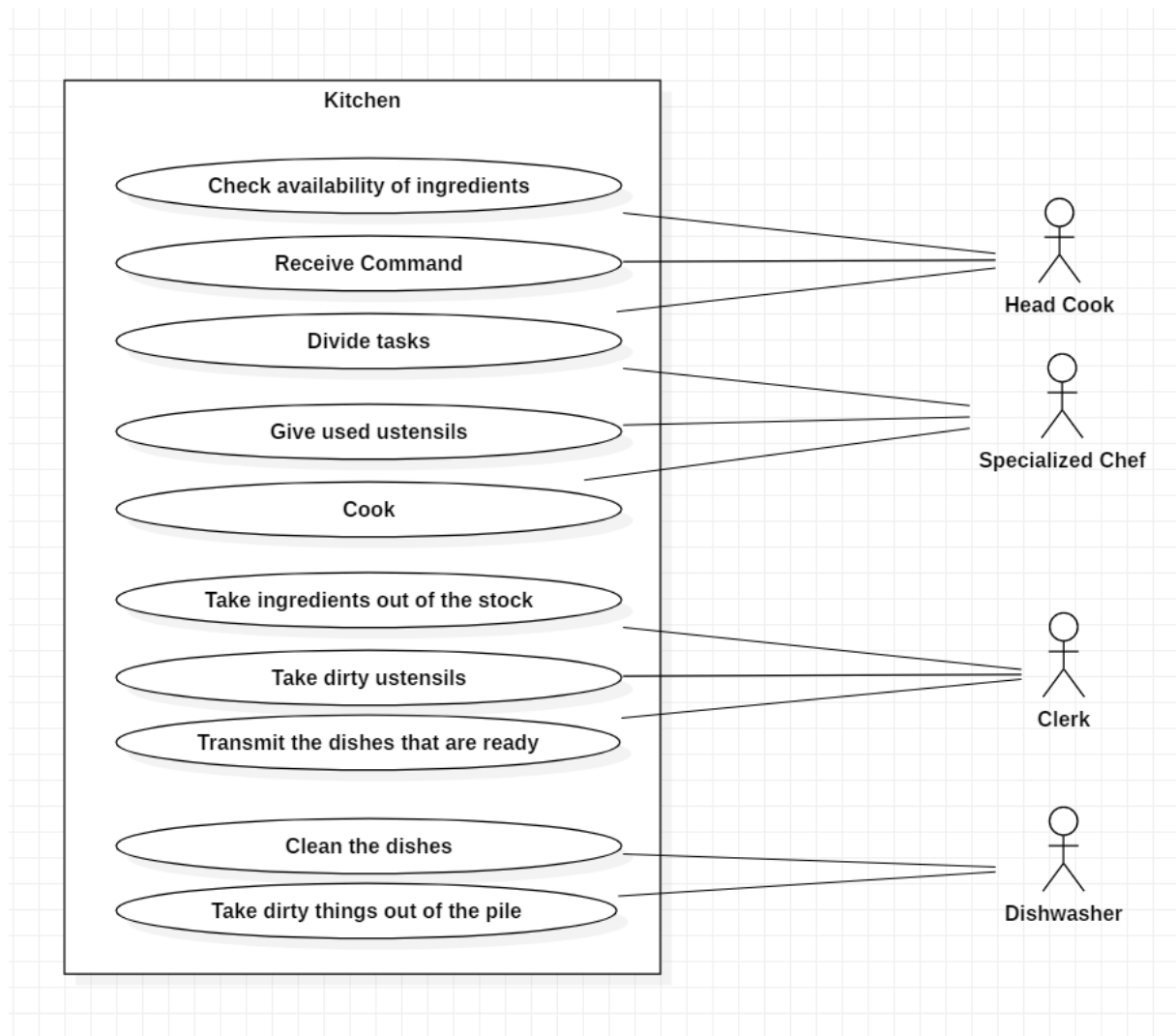


Ce diagramme définit l'ensemble des actions de nos acteurs dans un certain système. Ici, les actions sont aux nombres de 19 et elles appartiennent tous à au moins un personnage de la salle. Ici ne sont représentés que les acteurs travaillant dans l'espace de la salle, ceux n'ayant que des interactions avec ne sont pas représentés.

On a donc l'action du :

- Client :
  - Mange son repas
  - Paye
  - Prend la carte
  - Choisis sa commande
- Serveur :
  - Prend les commandes depuis le comptoir
  - Prend connaissance du contenu du menu
  - Sert les boissons et le pain
  - Sert les plats
  - Débarrasse la table
- Chef de rang :
  - Donne le menu
  - Prend la commande
  - Prépare la table
  - Reprend le menu
  - Informe les clients du contenu du menu
- Maitre d'hôtel :
  - Assigner une table
  - Appeler le chef de rang
- Commis de cuisine :
  - Met les couverts sales dans le stock
  - Réapprovisionne en eau et pain

## Cuisine



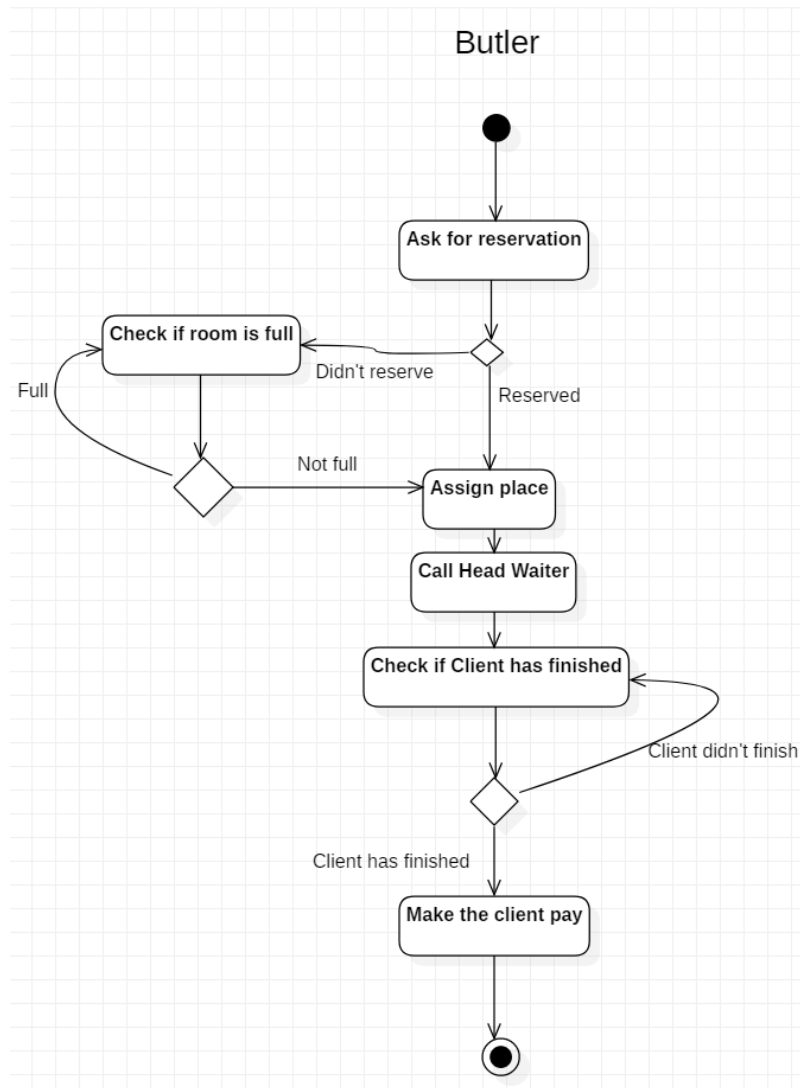
Ce diagramme définit l'ensemble des actions de nos acteurs dans un certain système. Ici, les actions sont aux nombres de 10 et elles appartiennent tous à au moins un personnage de la cuisine. Ici ne sont représentés que les acteurs travaillant dans l'espace cuisine, ceux n'ayant que des interactions avec ne sont pas représentés.

On a donc l'action de :

- Chef cuisinier :
  - Vérifier la disponibilité des ingrédients
  - Recevoir les commandes
  - Diviser les tâches
- Chef de partie :
  - Diviser les tâches
  - Libérer les ustensiles utilisés
  - Cuisiner
- Commis de cuisine :
  - Prendre des ingrédients du stock
  - Ramasser les ustensiles sales
  - Donner les plats prêts aux serveurs
- Plongeur :
  - Nettoyer les plats
  - Sortir les affaires sales de la pile
  - Utiliser le lave-linge
  - Utiliser la machine à laver

## Diagrammes d'activité

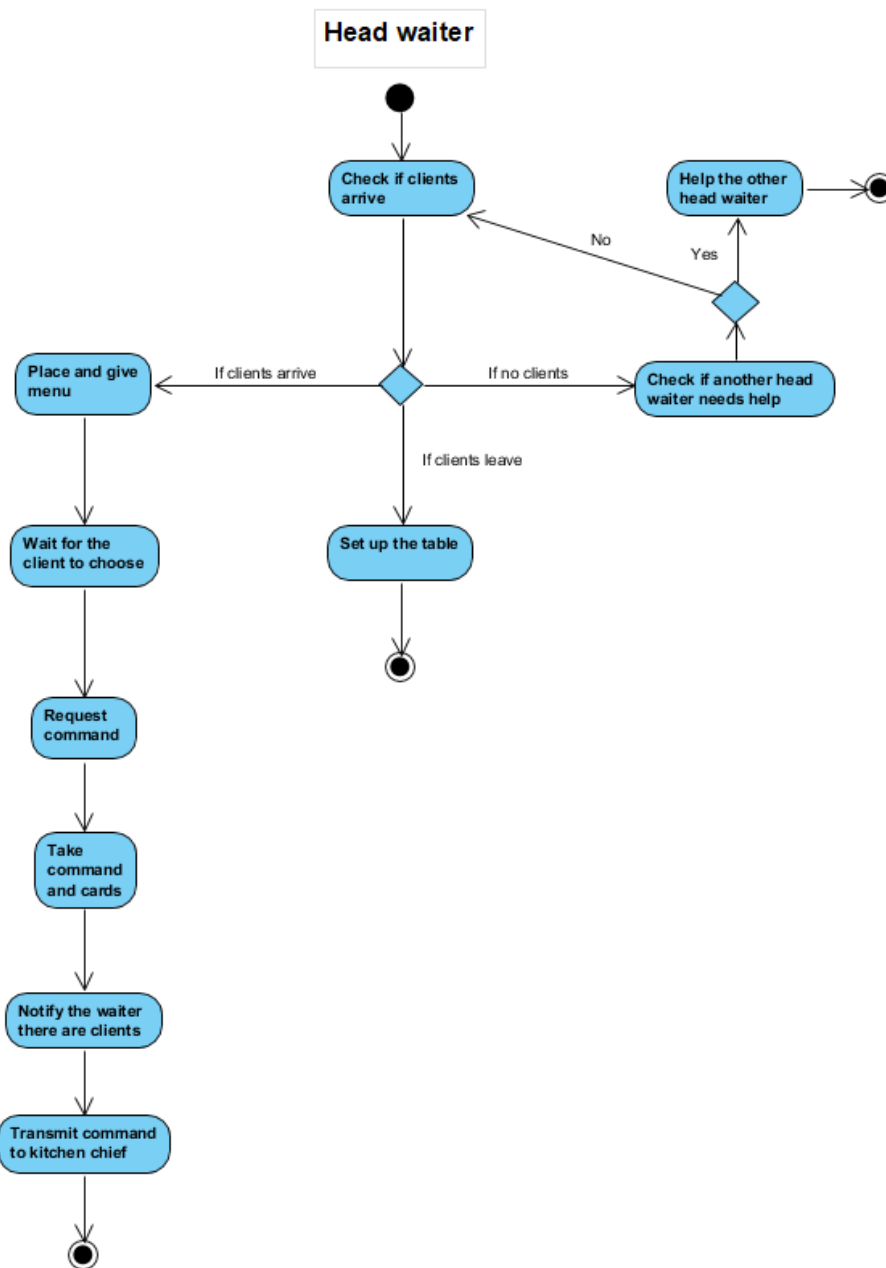
## Maître d'hôtel



Tout d'abord le maître d'hôtel va engager une rencontre avec le client, ce qui va démarrer ce diagramme. Il va d'abord demander si une réservation a été effectuée, dans le cas où le client y a fait recours il peut directement passer à l'étape suivante, cependant, si aucune réservation n'a été faite, le maître d'hôtel va d'abord vérifier que la salle n'est pas pleine, si une table est disponible, on passe à l'étape suivante, dans le cas contraire on retourne à l'étape précédente tant qu'aucune table n'est disponible.

Enfin notre personnage va en même temps, assigner une place ainsi qu'appeler un chef de rang. Une fois fini, il va attendre que les clients aient fini de manger pour prendre le paiement.

## Chef de rang

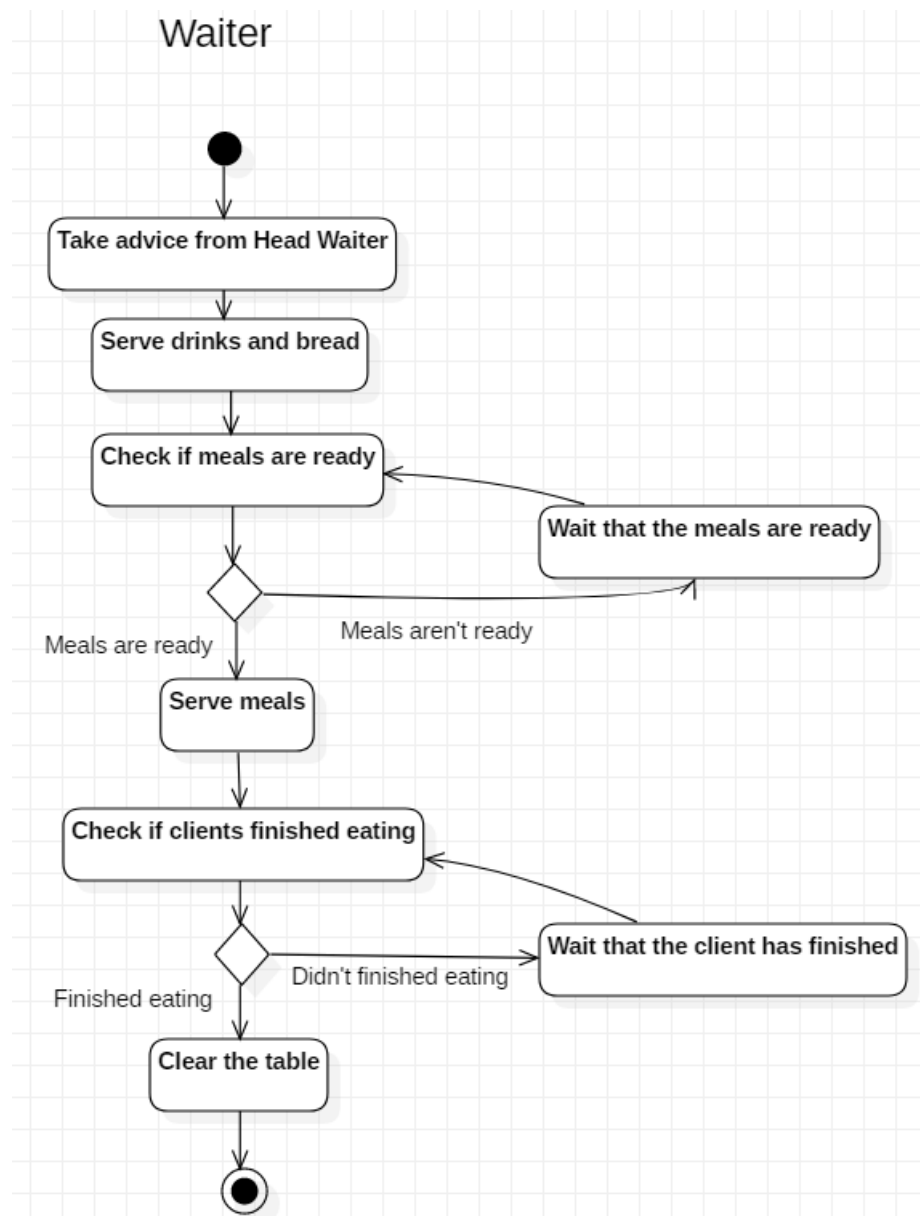


Il va d'abord vérifier si un client arrive, il y a plusieurs cas de figure :

- Un client arrive : Il va placer le client et donner le menu, attendre que le client choisisse sa commande, une fois que le client est prêt il va prendre la carte et la commande, en informer le serveur et prévenir le chef cuisinier.
- Un client part : Il va simplement préparer la table.
- Aucun client n'arrive : Il va vérifier si un autre chef de rang a besoin d'aide et éventuellement l'aider.

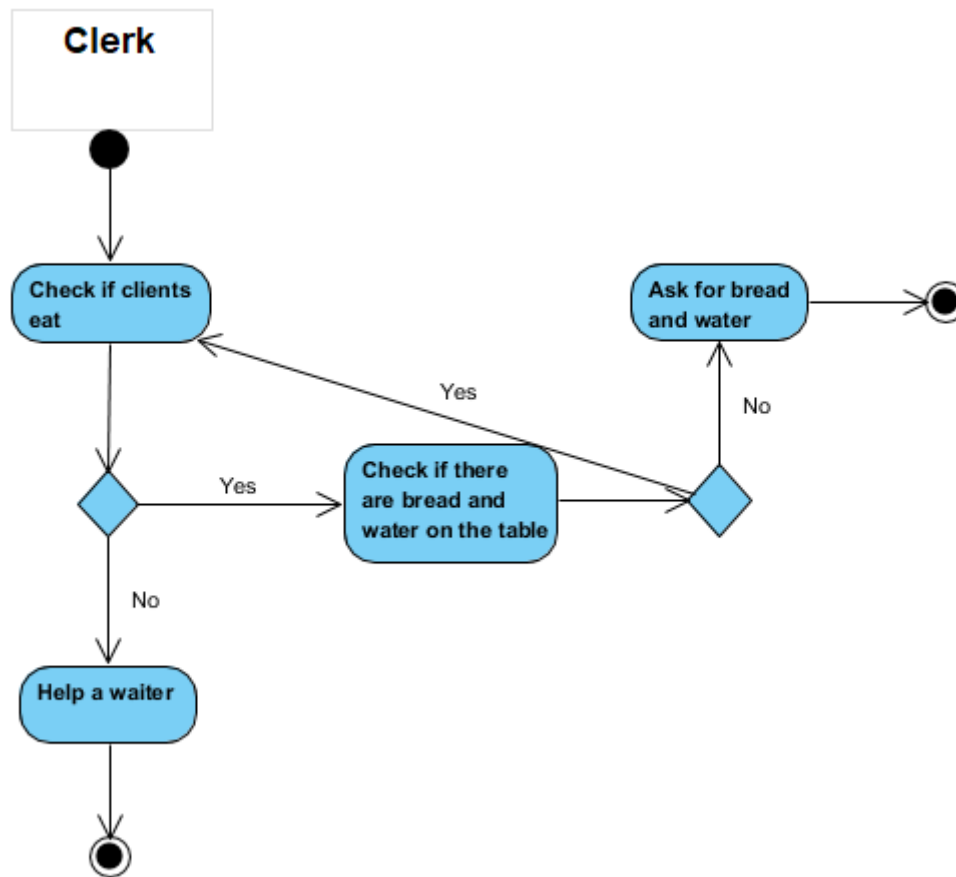


Serveur



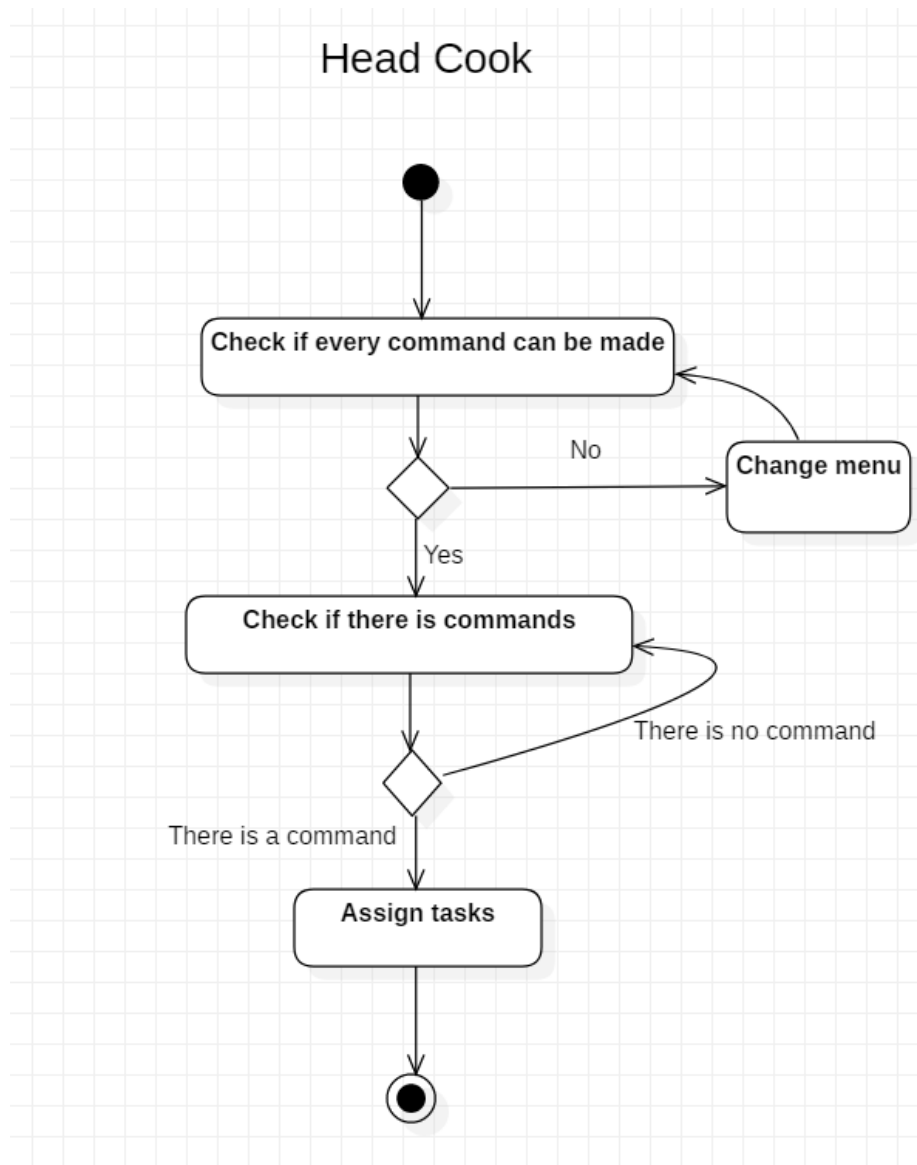
Le serveur va d'abord servir les boissons et le pain aux clients, puis attendre que les commandes prises par le chef de rang soient disponibles, afin de logiquement les apporter. Une fois que les clients ont terminé, il va débarrasser la table et déposer les couverts sales dans l'endroit prévu à cet effet.

Commis de salle



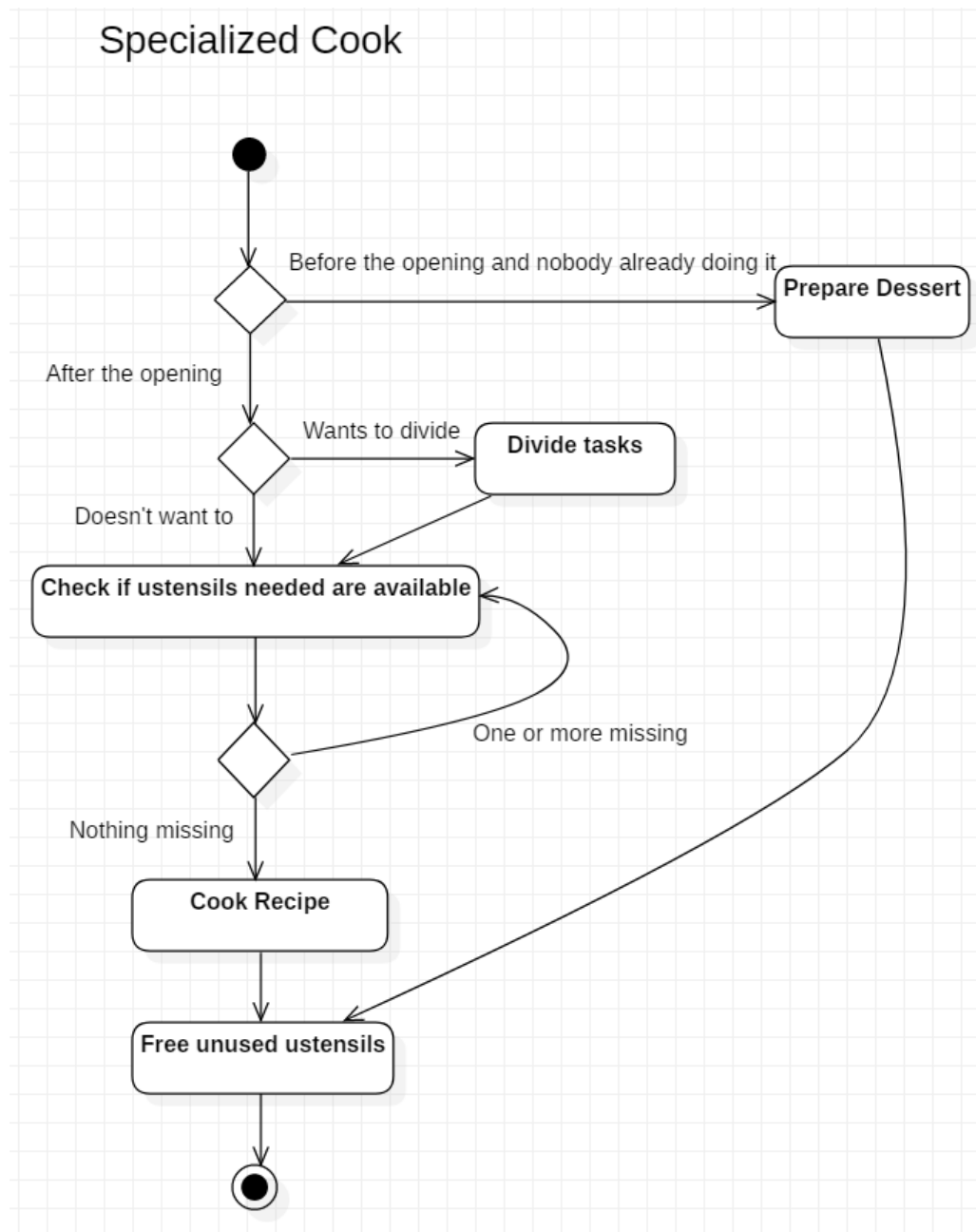
Tout d'abord il va vérifier que les clients sont encore en train de manger, s'ils le sont il va vérifier que tous les clients ont assez de pain et d'eau, sinon il va leur en proposer. Enfin, si tous les clients sont satisfaits, il peut aider un serveur.

Chef cuisinier



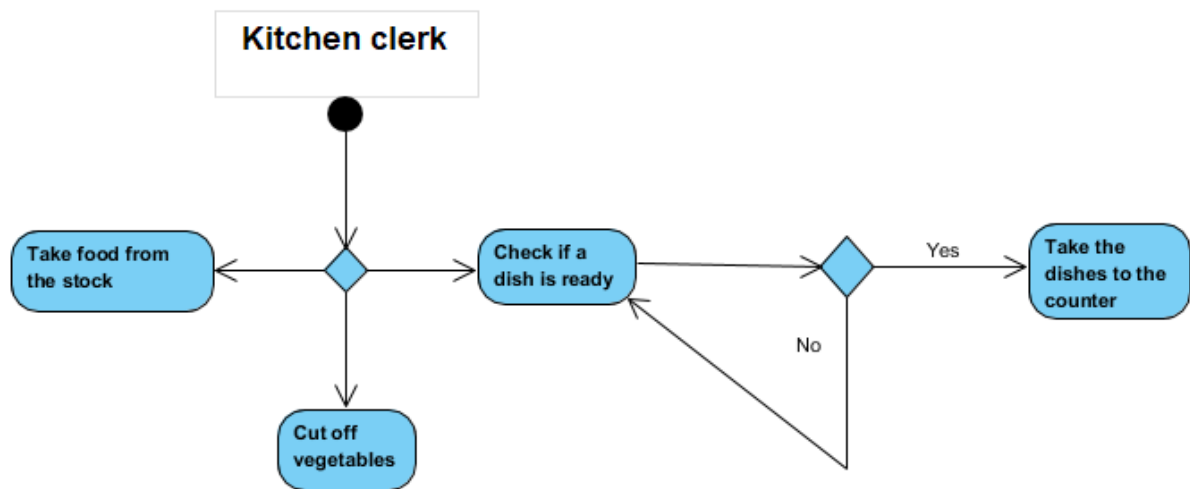
Son rôle est assez simple, il va tout simplement vérifier qu'aucun ingrédient ne manque dans les recettes mises à disposition sur le menu, sinon il va changer le menu. S'il n'y a aucun problème, il va prendre les commandes et assigner des tâches aux chefs de partie.

Cuisinier



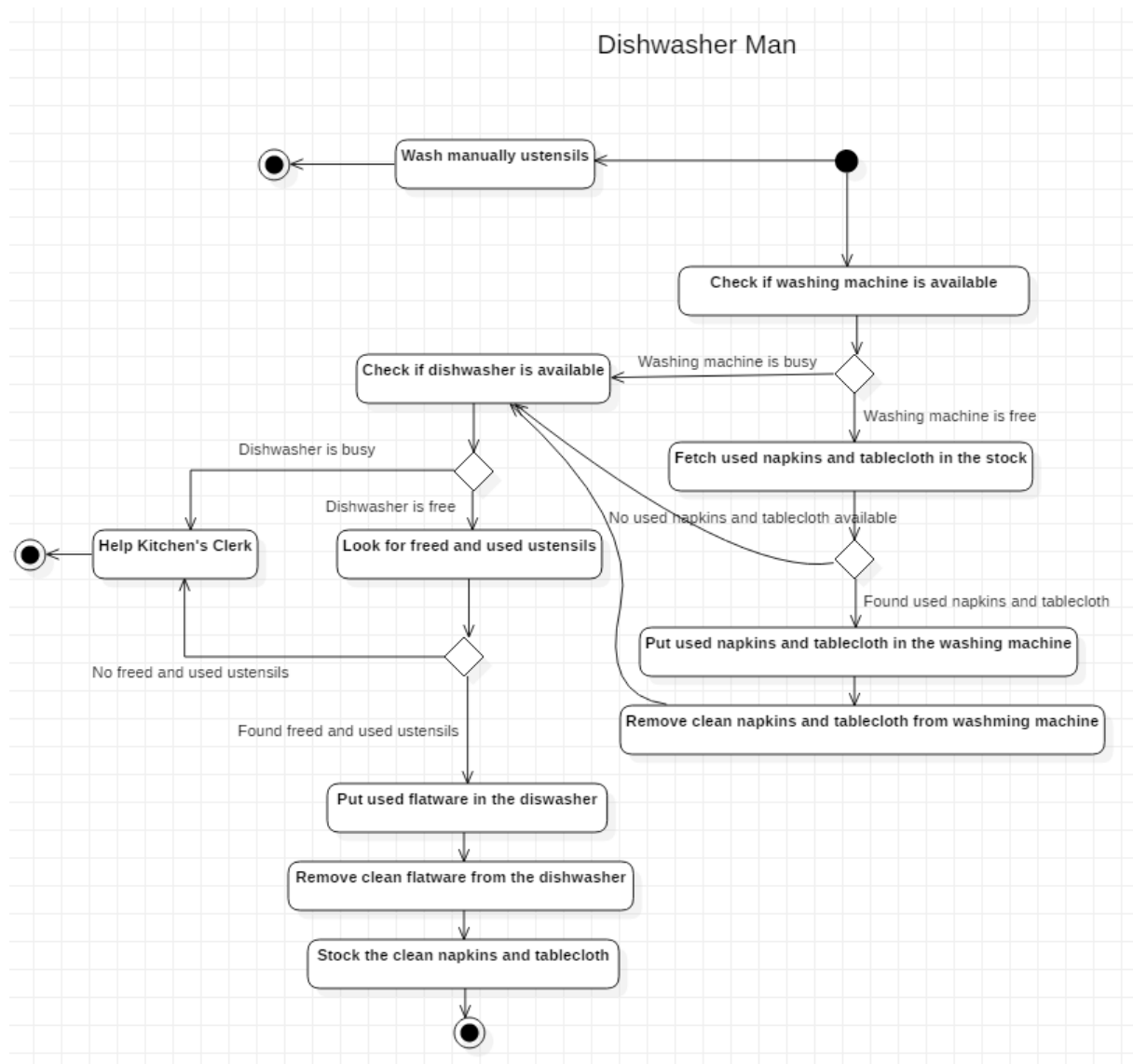
Le chef de partie (limité à 2) va d'abord préparer les desserts si aucun chef ne le fait avant l'ouverture du restaurant. Puis, s'il le veut, il peut diviser le travail qui lui est donné par le chef cuisinier pour accomplir une tâche. Il va ensuite vérifier que tous les ustensiles nécessaires pour effectuer une tâche sont présents, sinon il va attendre que le plongeur ait fini de les laver. Enfin une fois que tout est bon, il va préparer la recette et libérer les ustensiles dont il n'a plus besoin.

Commis de cuisine



Le commis va avoir plusieurs solutions dès le début, il a le choix de soit prendre des ingrédients depuis le stock, soit couper des légumes, soit attendre qu'un plat est prêt pour emmener la commande au comptoir.

## Plongeur



Tout d'abord le plongeur va en même temps laver à la main les ustensiles et commencer par regarder si le lave-linge n'est pas en route, si il l'est, le plongeur passe à la prochaine étape, sinon, il va aller chercher les affaires sales dans le stock. S'il n'y a rien également, le plongeur passe à la prochaine étape, sinon il prend les affaires sales, les met dans le lave-linge et les enlève une fois propre, ce qui l'emmène à l'étape suivante.

Ensuite, le plongeur va regarder si la machine à laver est disponible, si elle ne l'est pas il va partir aider le commis de cuisine, sinon il va aller logiquement chercher des ustensiles usés, si aucun n'est disponible il va encore une fois aller aider le commis, sinon il va mettre ces objets dans la machine, les enlever une fois propre et les mettre à disposition.

## Diagrammes de classes et composants

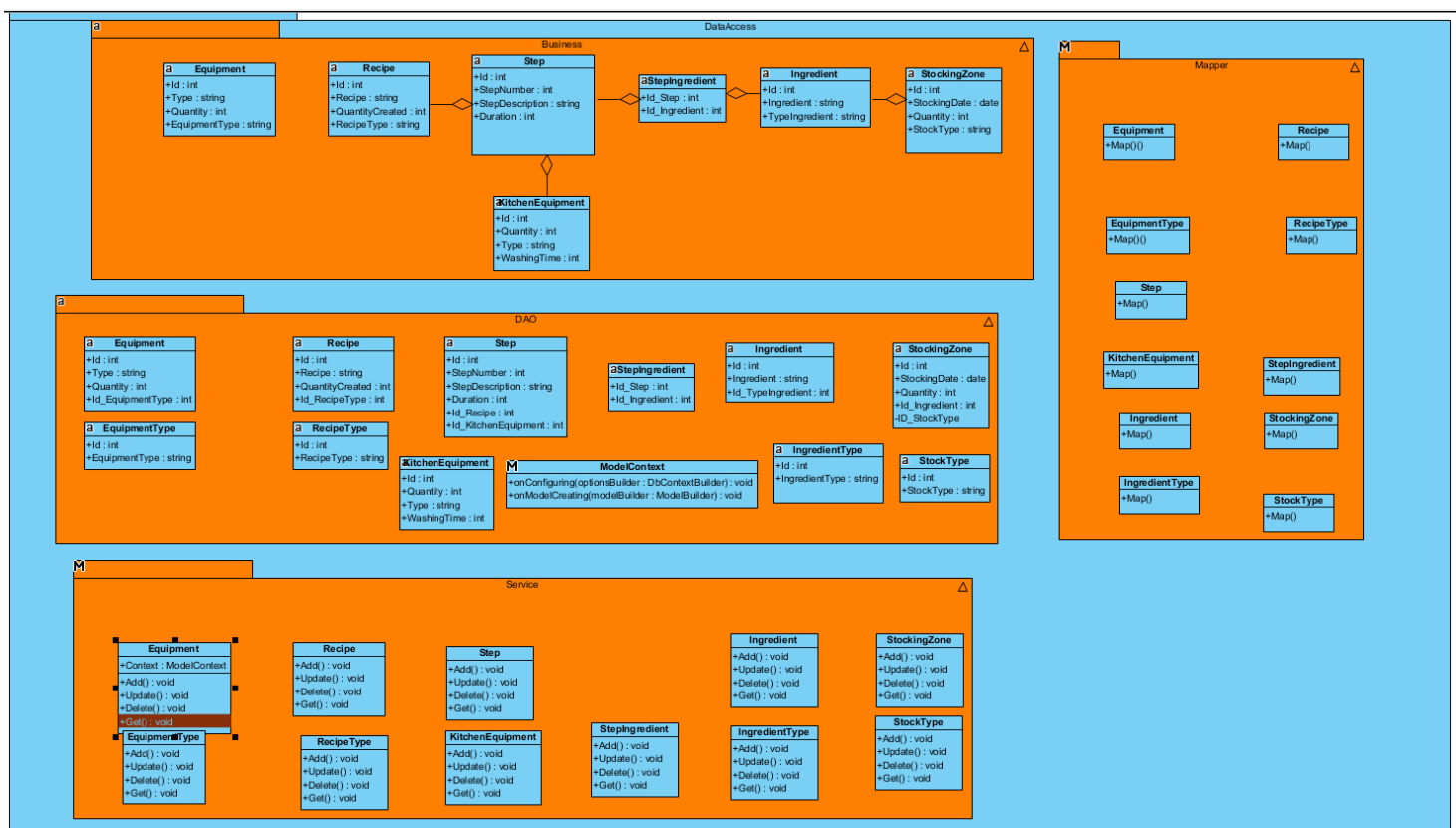
## Classes

## Cf. Diagramme de classe.vpp

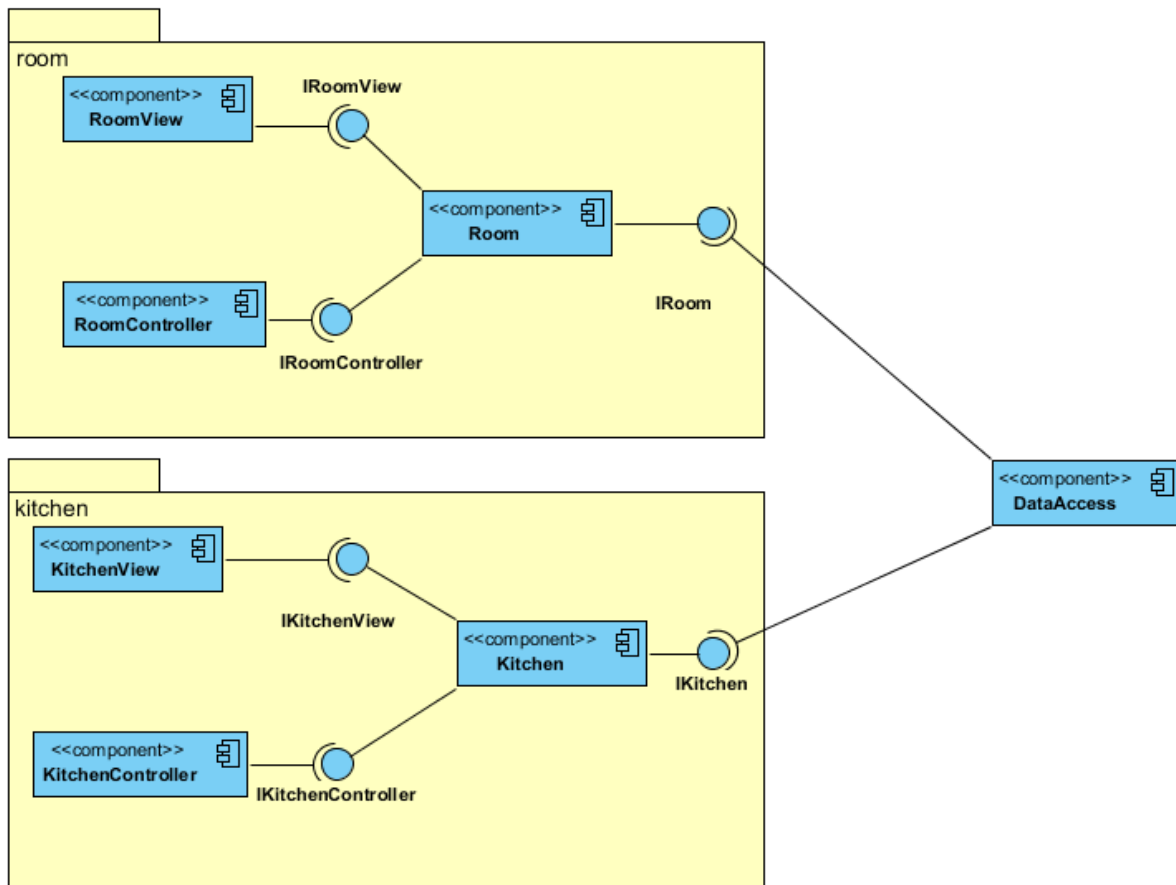
## Accès aux données

Pour accéder aux données nous avons utilisé une architecture en 4 couches :

- Persistance de données (Package service) : Cette couche s'occupe d'assurer l'exécution des 4 actions CRUD sur chacune des tables de la base de données.
- DAO : Elle est la représentation de la base de données sous forme d'objets. Elle contient donc une classe adaptée à chaque table pour récupérer les informations de la base de données.
- Métier : Cette couche contient une représentation de la base de données à un instant donné qui est manipulable par le développeur sans risquer d'affecter les données stockées dans la base de données.
- Mappeur : Cette couche va s'occuper de faire la translation en les objets du DAO et ceux de la couche Métier.



## Composants



Chacun des composants présents dans le diagramme correspond et doit fournir un service bien précis.

Le diagramme de composant permet de séparer chacune des couches pouvant être séparée du reste du programme afin de les réutiliser dans un autre programme.

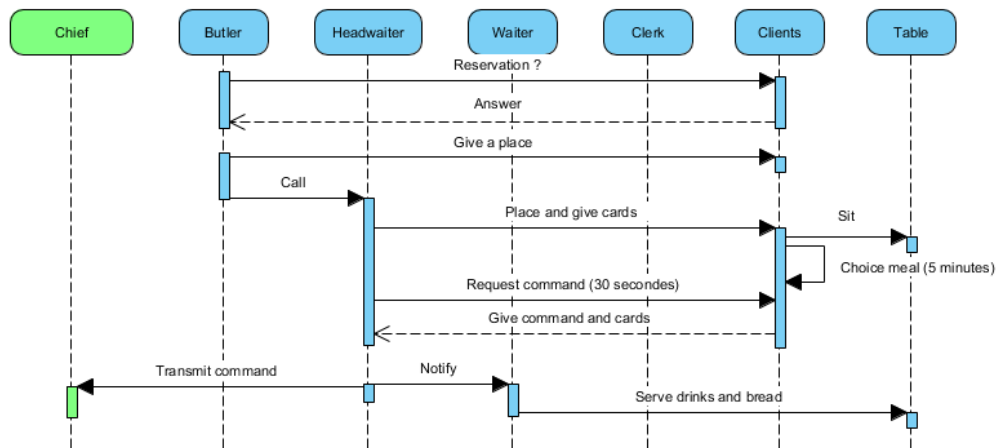
De ce fait, dans notre diagramme, nous pouvons récupérer les différentes parties qui composent notre programme. Comme nous l'apercevons, l'application en trois couches ainsi que le Design Pattern MVC ont permis de séparer directement les différents composants de notre programme.

Dans notre cas, si nous créons une nouvelle application ayant besoin d'accéder à la même BDD, il nous suffit de reprendre notre composant `DataAccess` et de le relier de manière adaptée à notre nouveau projet.



## Diagrammes de séquence

## Partie Salle

*Première partie*

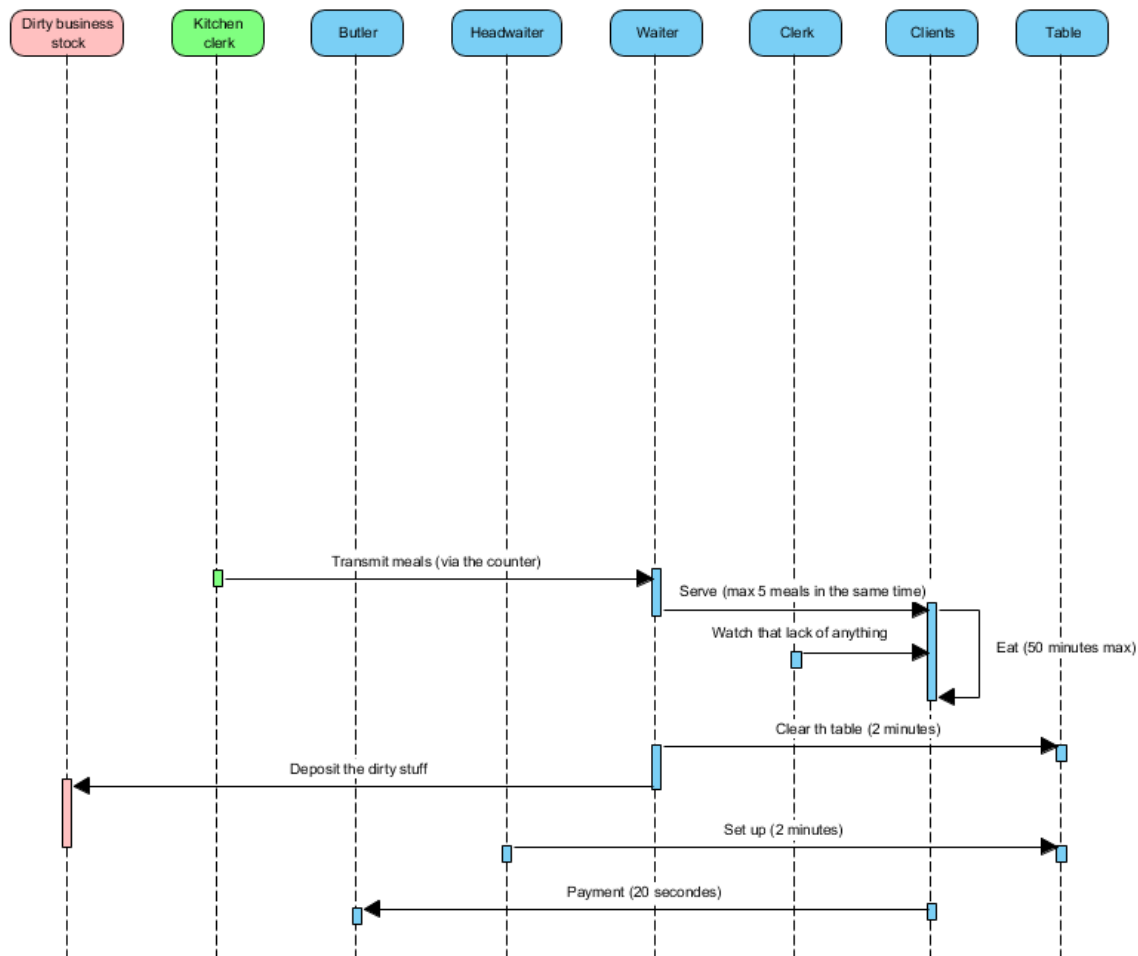
Cette première partie représente le scénario classique lors de l'arrivée des clients, principalement du point de vue de la salle :

Premièrement le maître d'hôtel accueille les clients en leur demandant s'ils ont réservé, puis attend une réponse.

Ce dernier attribue une table aux clients puis appelle le chef de rang afin de les placer et leur donner une carte du menu. Les clients choisissent leurs repas pendant 5 minutes et restituent leurs choix au chef de rang qui s'approche tout en rendant les cartes.

Le chef de rang quant à lui, interpelle un serveur pour qu'il puisse apporter de l'eau et du pain en fonction du nombre de personne présentes sur la table. Enfin, il transmet la commande des clients au chef cuisinier.

## Deuxième partie



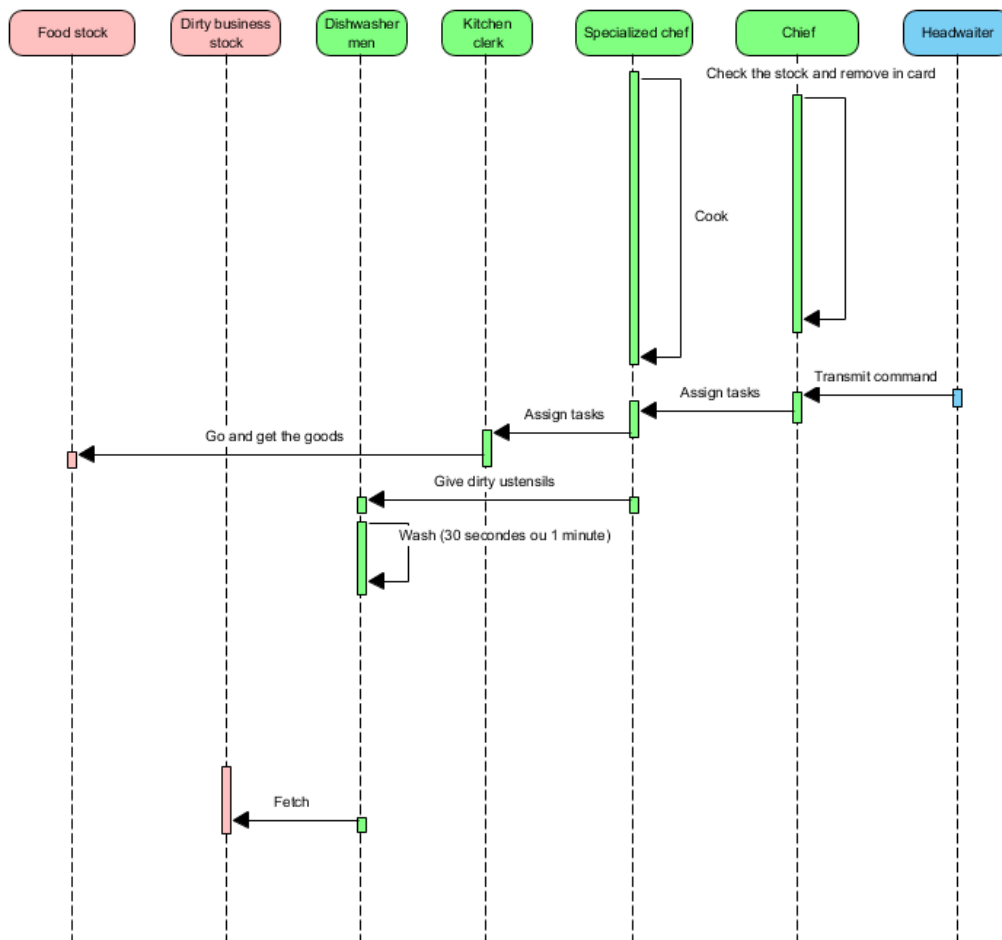
Cette seconde partie concerne seulement le moment où les plats sont prêts à être servis, jusqu'au départ des clients :

Premièrement, le commis de cuisine pose les plats sur le comptoir afin que le serveur puisse servir les clients. Ces derniers mangent pendant environ 50 minute, et peuvent demander si besoin au commis de salle de leur servir de l'eau, du pain et du vin.

Lorsque les clients ont fini de manger, le serveur peut desservir la table puis ramener les couverts ainsi que les tissus sales au lave-vaisselle et à la machine à laver.

Enfin, le chef de rang dresse la table laissée par les clients, puis ces derniers peuvent payer leur repas.

## Partie Cuisine

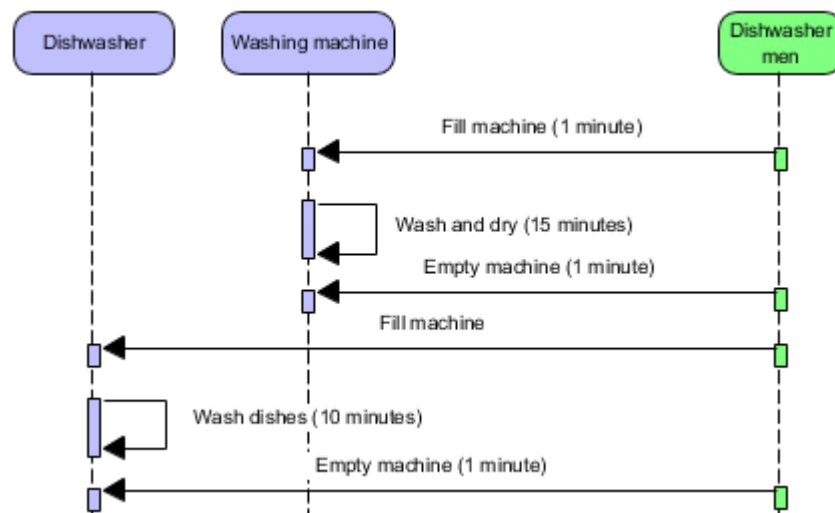
*Préparation de la nourriture*

Cette partie concerne la préparation des repas des clients :

Premièrement, le chef de rang donne la commande au chef cuisinier. Ce dernier va assigner différentes tâches aux deux chefs de partie, ils vont donc demander aux commis de cuisine d'aller chercher de la nourriture dans la réserve.

Lorsque les ustensiles ont été utilisés, les chefs de partie les donnent au plongeur, qui s'occupera de les laver à la main en fonction de la taille de l'ustensile.

Enfin, le plongeur récupère les plats ainsi que les affaires sales au comptoir des plats sales.

*Nettoyage du matériel*

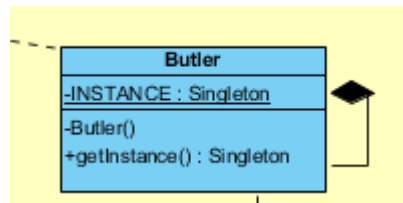
Cette partie concerne uniquement le lavage des « choses » sales :

Le plongeur se verra en permanence aller remplir la machine à laver ainsi que le lave-vaisselle, ces derniers ont le même principe de fonctionnement et vont donc être en marche pendant 15 et 10 minutes. Enfin il récupère les affaires propres.

## II. Designs patterns

### Design Pattern Singleton

Nous avons utilisé le DP singleton sur la classe Butler, en effet cette classe représente le Maître d'hôtel qui se doit d'être unique dans le restaurant. Le Singleton nous permet donc de nous assurer que ce dernier ne sera jamais instancié plus d'une fois.



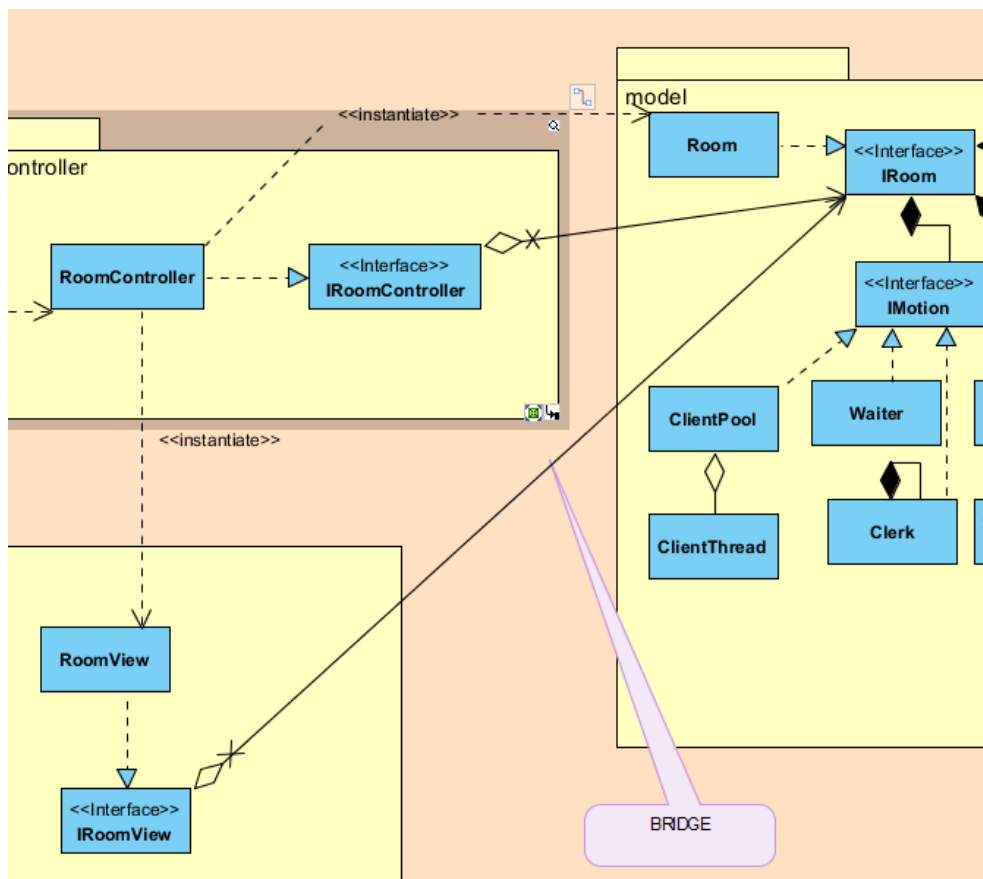
## Design Pattern Bridge

L'objectif du DP Bridge est de découpler une abstraction de sa mise en œuvre de sorte que les deux classes concernées peuvent varier indépendamment sans modifier une classe voisine.

Dans notre cas, nous avons choisi de placer ce dernier entre les différentes interfaces implémentées par le contrôleur, le modèle et la vue. Cela permet, par exemple, de pouvoir ajouter une deuxième vue de la cuisine. Un graphique pour l'utilisateur, mais également une console qui afficherait toutes les modifications dans le programme.

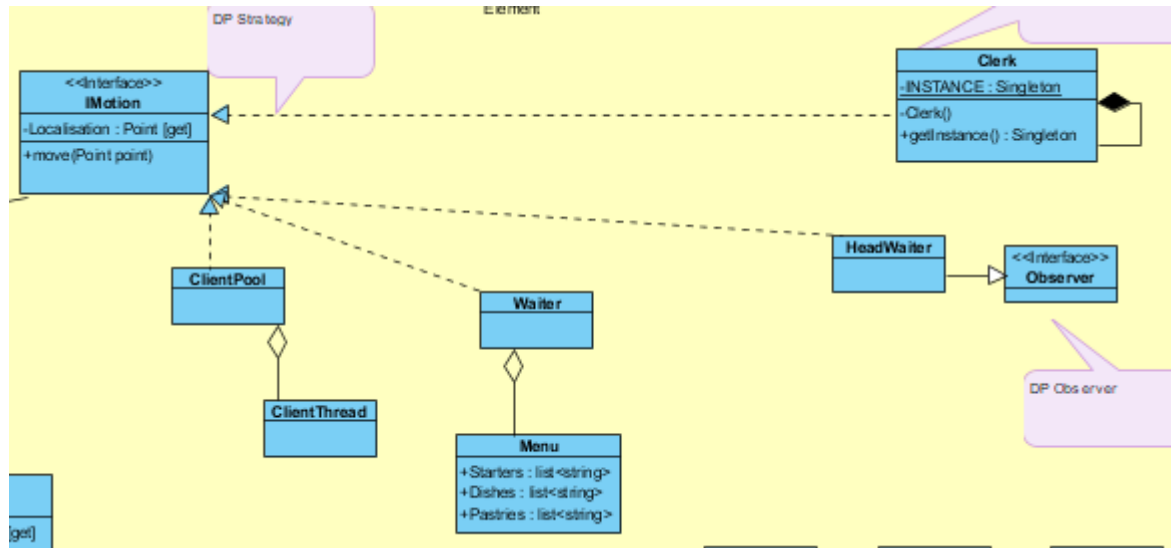
Si jamais il y a besoin de voir un problème non visible dans l'interface graphique, on pourrait le voir en console. Les deux interfaces implémentant les méthodes du modèles à travers l'interface IRoom mais également celle d'une vue classique, à travers IRoomView, ajouter la console est très simple, puisqu'il suffit de créer une classe implémentant IRoomView.

De même côté modèle ou contrôleur, si nous souhaitons ajouter des classes ou méthodes. Les méthodes "génériques" seront toujours utilisables à travers les interfaces par les autres parties du code.



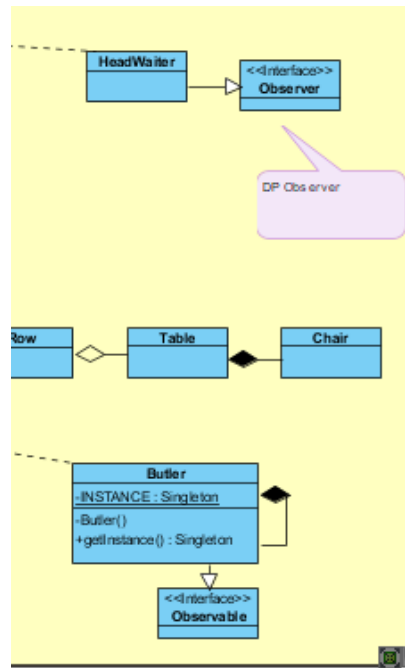
## Design Pattern Observer

Pour l'implémentation du design pattern Observer/Observable nous avons ici choisi de l'utiliser pour toutes les parties nécessitant de renvoyer une notification à un autre objet à chaque changement. En effet, ce design permet entre autres de notifier un observateur lorsqu'un observable à un event. Nous l'avons donc à un endroit : Pour le maitre d'hôtel qui a besoin d'appeler un chef de rang dès qu'un client a été assigné à une place.



## Design Pattern Strategy

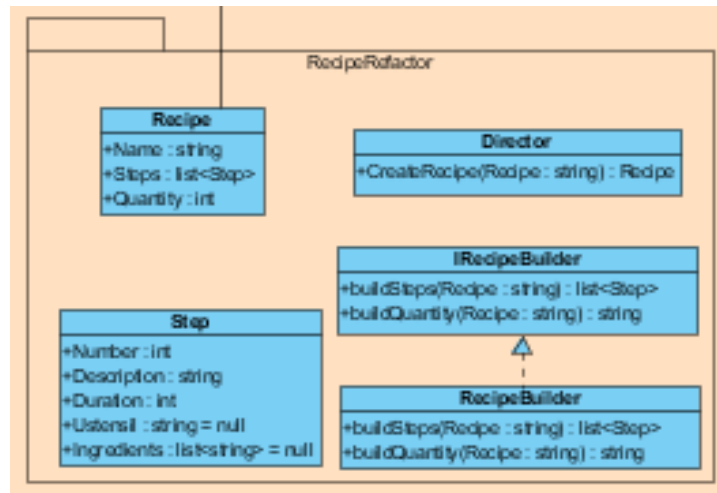
Pour l'implémentation du design pattern Strategy nous avons choisi de l'utiliser pour les objets que nous avons découpé en deux parties : Motion (Pouvant bouger) et Motionless (Immobile). On peut voir l'implémentation de Strategy dans le contexte de la cuisine et de la salle. Cela permet d'avoir un choix d'algorithme en fonction de l'objet que l'on va utiliser.





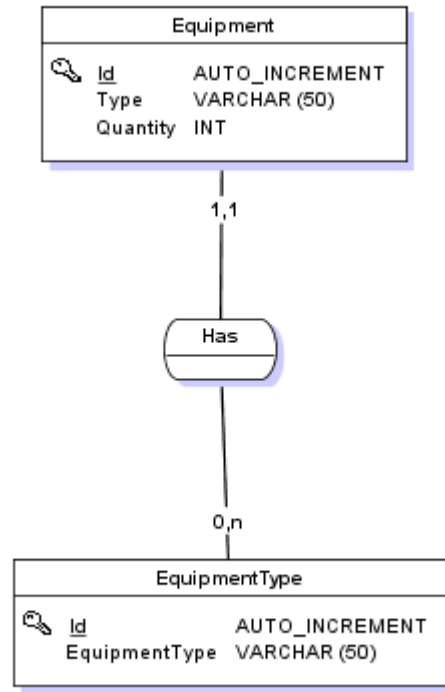
## Design Pattern Builder

Nous avons utilisé le design pattern Builder pour créer un objet Recette qui contiendra une représentation d'une recette contenue dans la BDD. Il permet de découper en étapes simples les étapes de construction d'un objet complexe. Il permet donc d'éviter de définir un constructeur complexe et de déléguer la création de l'objet à un système externe à ce dernier. Nous l'avons utilisé pour construire un objet Recipe qui contient une version facilement manipulable pour un développeur d'une recette donnée. En effet la construction de cet objet nécessite le questionnement de 5 tables de la base de données.



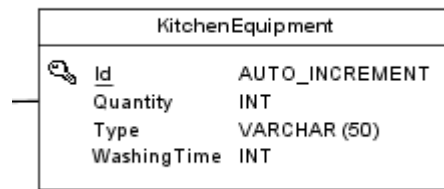
### III. Base de données

Réalisation du MCD :

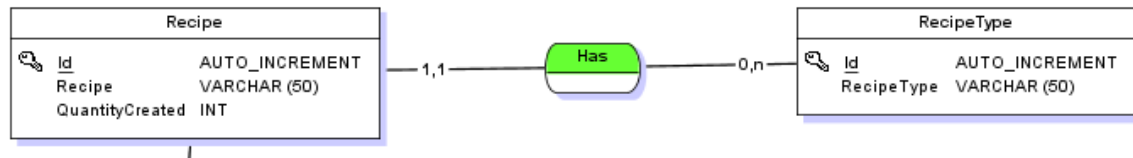


Pour gérer les équipements de la salle (carafe, corbeille à pain...) et ceux communs à la salle et la cuisine (couverts, assiettes...), nous avons créé une table commune pour ces deux types car les éléments possédaient les mêmes attributs, nous conservons la différence entre les objets de salle et ceux communs à la cuisine et la salle.

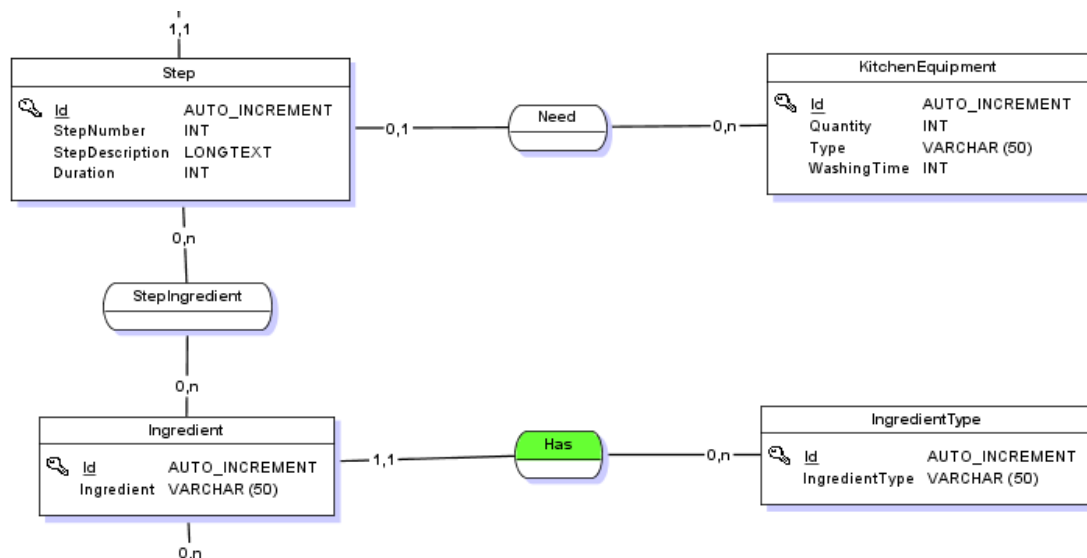
Nous avons séparé les équipements de la cuisine car ces derniers possédaient un attribut supplémentaire : le temps de lavage. Ces objets sont stockés dans la table suivante :



Pour le stockage des recettes nous avons commencé par créer une table recette qui contient le nom de la recette et la quantité qu'elle crée. Nous séparons les différentes recettes avec un type de recette pour différencier les entrées, les plats et les desserts. Voici les tables correspondantes :

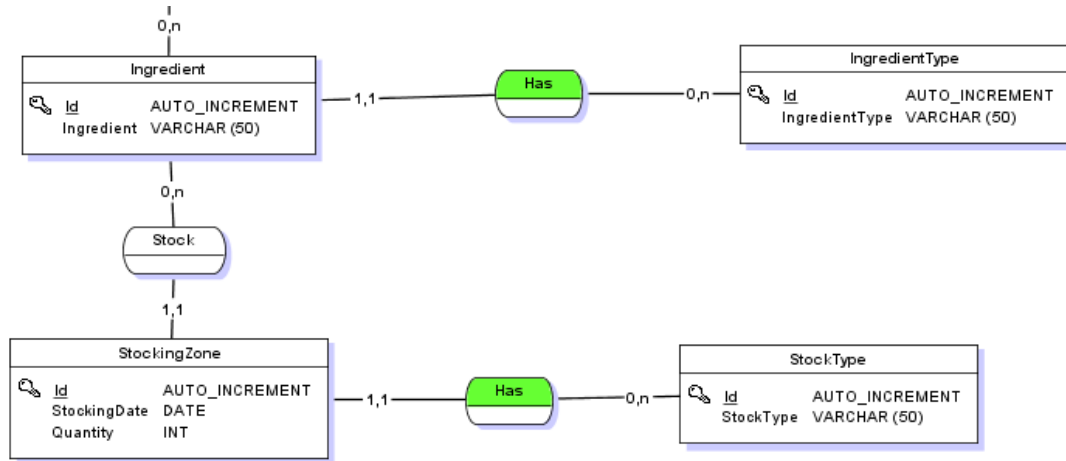


Nous avons ensuite créé une table pour stocker les étapes de la recette, une étape est une action à réaliser dans la recette, elle comporte donc une description, une durée et un numéro d'étape. Nous avons mis la table étape en relation avec la table des équipements de cuisine, ainsi nous savons pour chaque étape quel outil utiliser. Nous l'avons également mis en relation avec les ingrédients afin de pouvoir récupérer les ingrédients nécessaires à chaque étape.



La table Type ingrédient permet de différencier si un ingrédient est à stocker au frais, en température ambiante ou en congélateur.

Pour gérer le stock des ingrédients nous avons créé une table zone de stockage qui contient la date d'ajout en stock d'un ingrédient, la quantité stockée et est en relation avec le type de stock pour dire si le produit est au congélateur, frigo ou à température ambiante.



(Cf. « MCD.png » et « MLD.png »)