Projet Programmation Système A3

« Master Chef Info »

Table des matières

Table des illustrations :

I. Diagrammes UML fonctionnels

a. Diagrammes de cas d'utilisation

Le diagramme de cas d'utilisation permet de donner une vision globale du fonctionnement global d'un logiciel, ils sont utilisés pour des présentations pour la direction, mais aussi pour les équipes de développement.

Cela permet de représenter l'interaction entre l'utilisateur et le système.

Application :

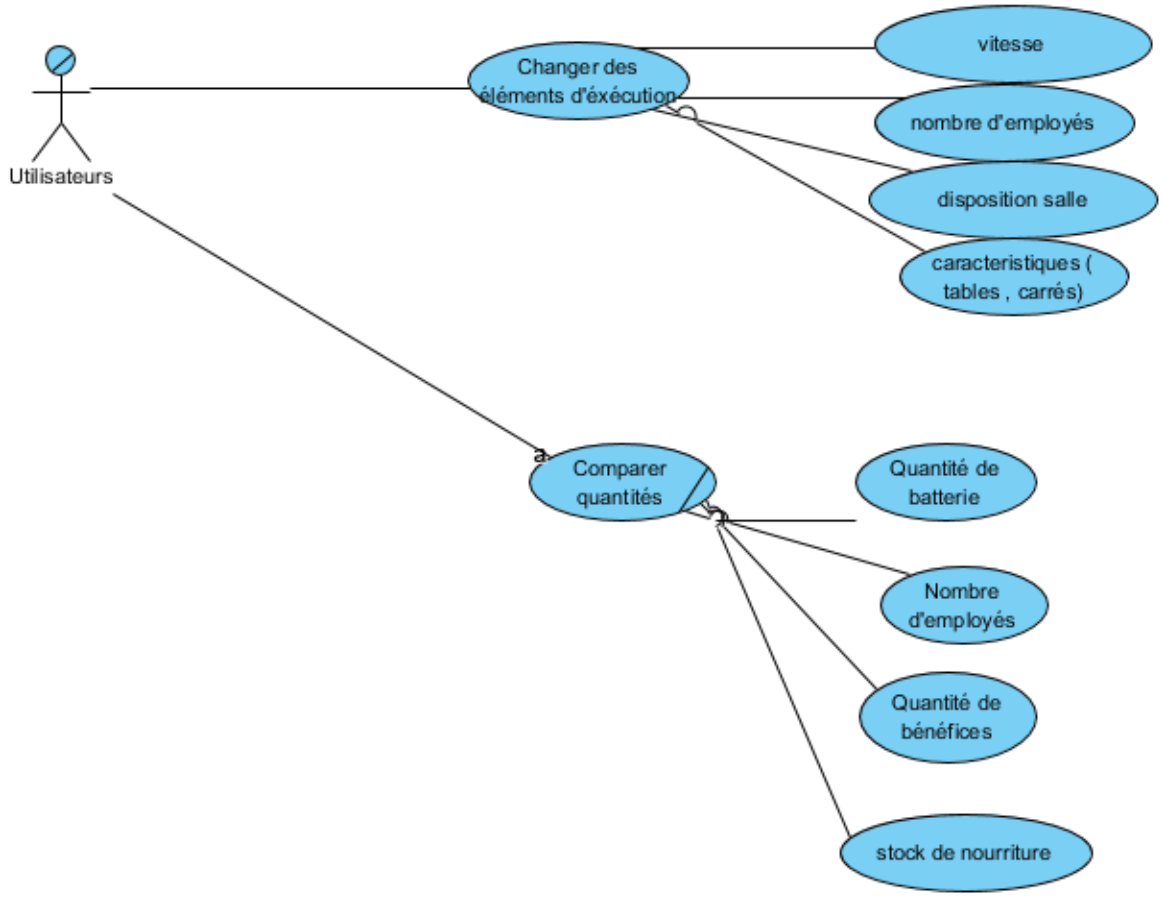


Figure 1- Diagramme cas utilisation - Application

L'utilisateur doit pouvoir modifier différents éléments d'exécution tel que la vitesse, le nombre d'employés, la disposition ainsi que les caractéristiques de la salle.

Il pourra aussi comparer les quantités de batteries, les employés ainsi que la pertinence de leurs dispositions, les bénéfices enregistrés par simulation et le stock de nourriture.

Salle de restaurant :

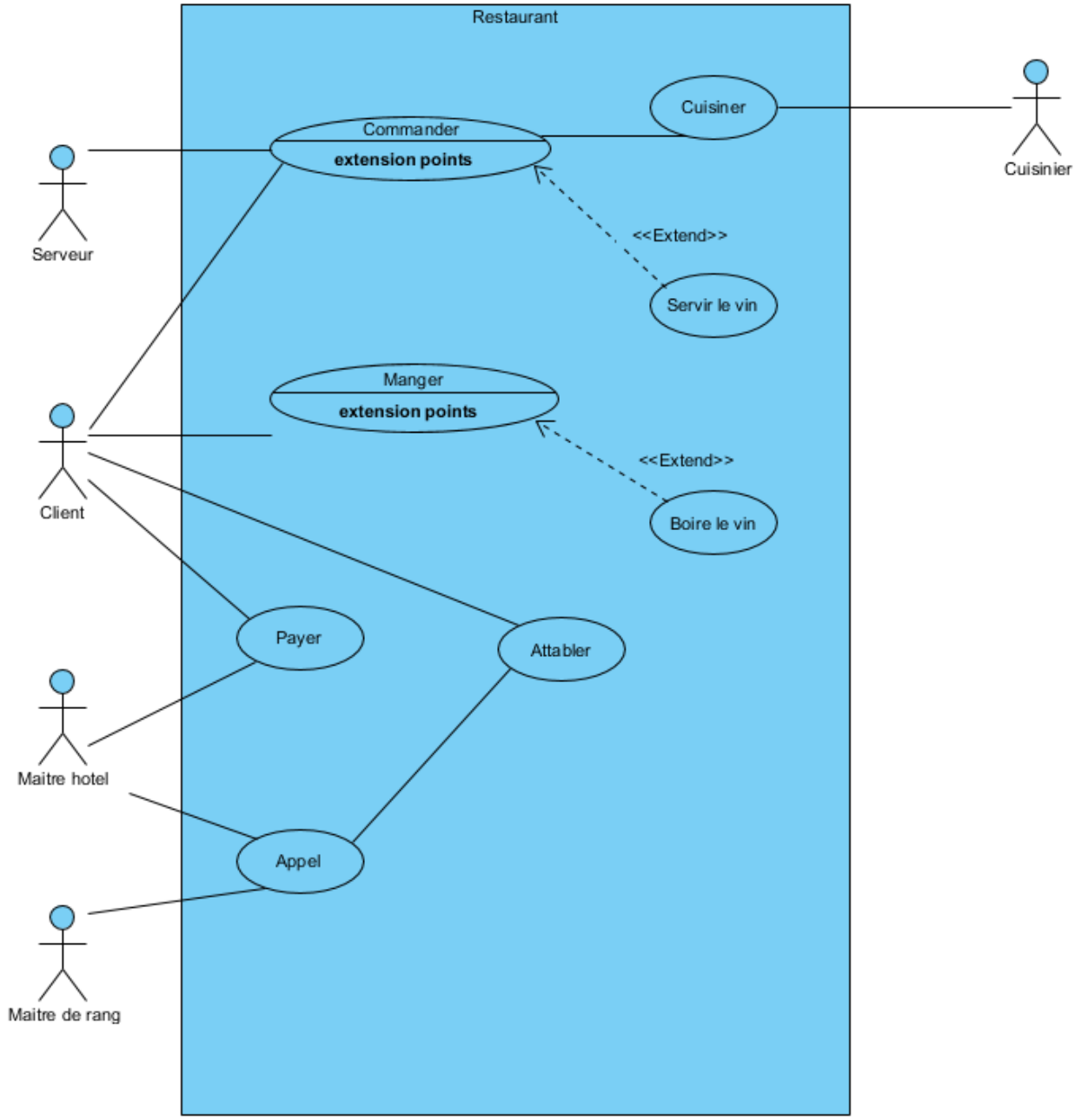
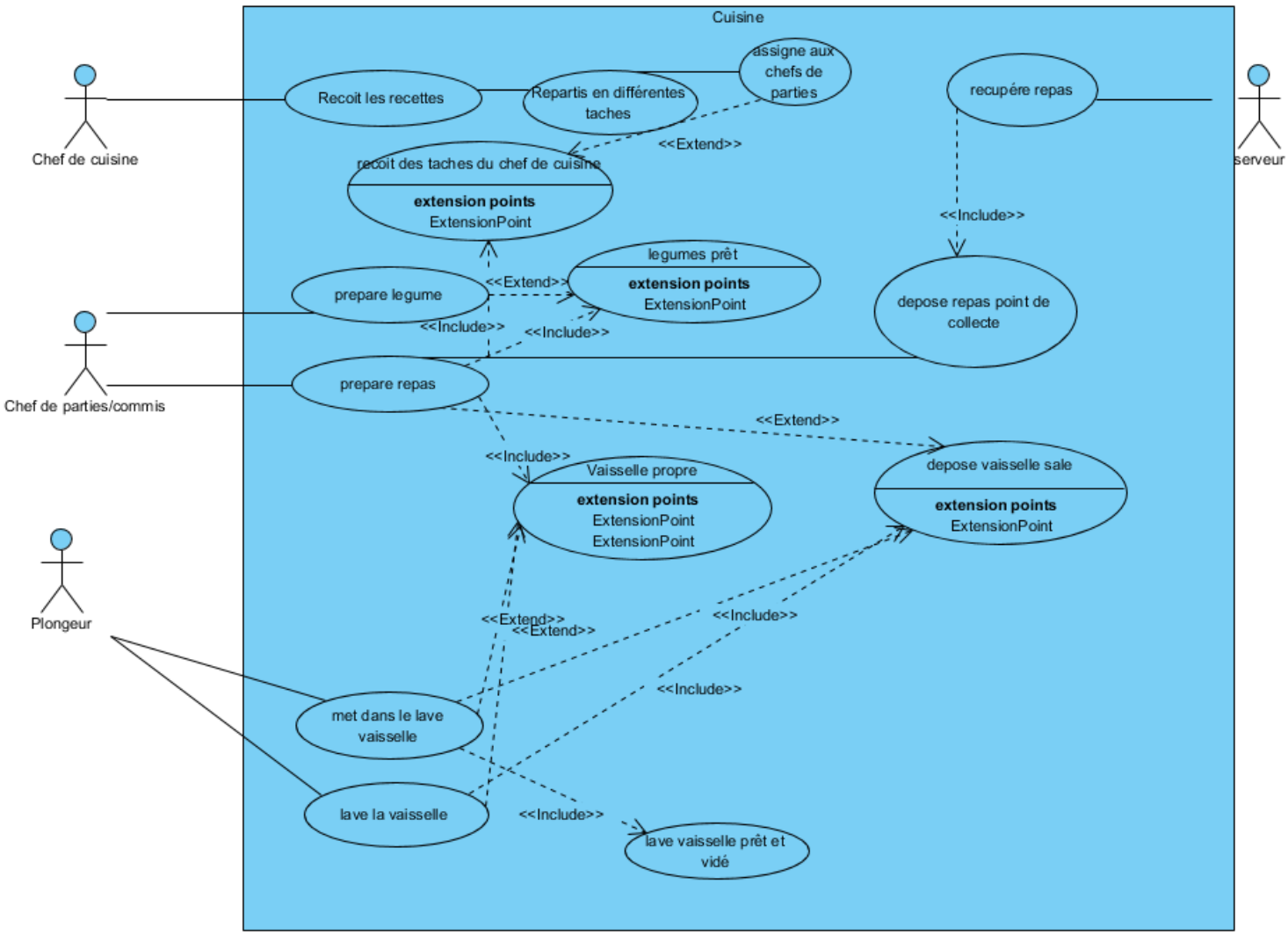


Figure 2 - Diagramme cas utilisation - Salle restaurant

Dans le restaurant, plusieurs acteurs entrent en action. En premier lieu on retrouve le client qui sera accueilli par le maître d'hôtel et qui devra être attablé grâce au maître de rang. Pour donner suite à cela le client va commander et le serveur va amener cette commande en cuisine pour les cuisiniers puissent effectuer la commande. Il y a également la possibilité que le client prenne du vin et dans ce cas le serveur amènera du vin.

Cuisine :



Le chef de cuisine reçoit les recettes puis les reparties en différentes tâches qui correspondent aux étapes de la recette, il distribue ensuite ces tâches aux différents chefs/commis.

Les chefs de parties reçoivent leurs tâches du chef de cuisine ainsi ils préparent les repas.

Pour préparer les repas, il faut de la vaisselle propre ainsi que des légumes préparés, lorsque le repas est préparé il est disposé en point de collecte et est récupéré par le serveur.

Les légumes sont préparés par les chefs de parties, les commis et le plongeur.

La vaisselle est effectuée par les plongeurs, elle est effectuée à la main ou dans le lave-vaisselle.

Pour utiliser le lave-vaisselle, il doit être prêt et vide.

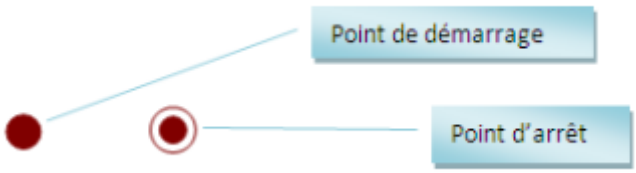
Ensuite la vaisselle étant effectuée elle est considérée comme propre et peut être utilisée.

b. Diagrammes d'activité

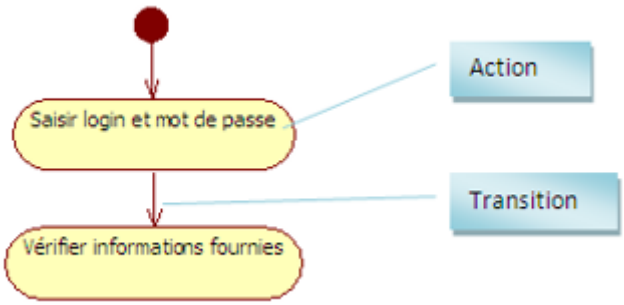
Un diagramme d'activité est un diagramme comportemental UML, il permet de modéliser le déclenchement des événements selon les états du système. Il permet également de modéliser des comportements. Ce type de diagramme est également utilisé pour décrire un workflow.

Un diagramme d'activité se caractérise par :

- Les Points de démarrage et d'arrêt
- Représenté par des cercles rouges

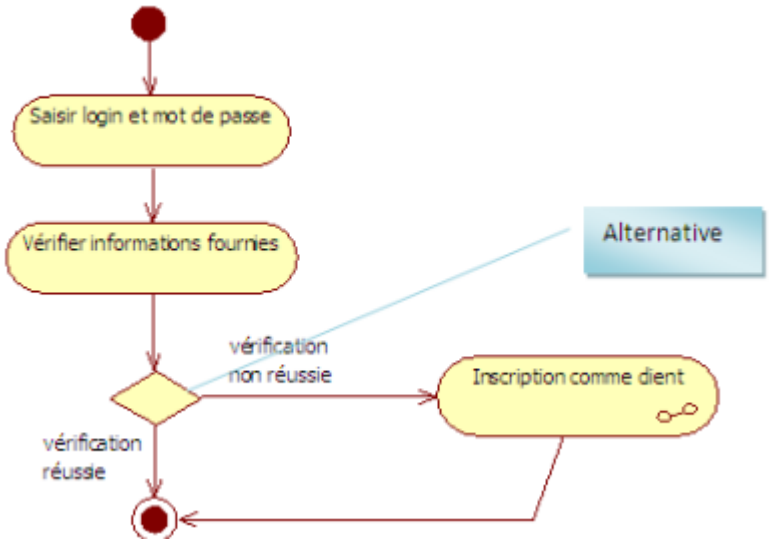


- Les actions et les transitions
- Le diagramme est organisé en actions réalisées par un acteur/le système.
- Toute les actions sont reliées par une flèche qui indique l'enchaînement des actions



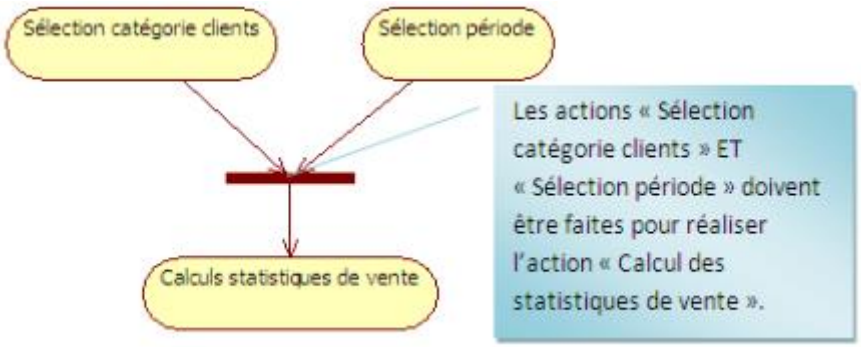
- L'alternative

- Ce type de représentation permet d'indiquer les différents scénarios du cas d'utilisation dans un diagramme



- La synchronisation

- Sert à indiquer qu'il faut avoir réaliser 2 actions pour réaliser la troisième



Voici dans notre projet, les différents diagrammes d'activité réalisés.

Client – Placement :

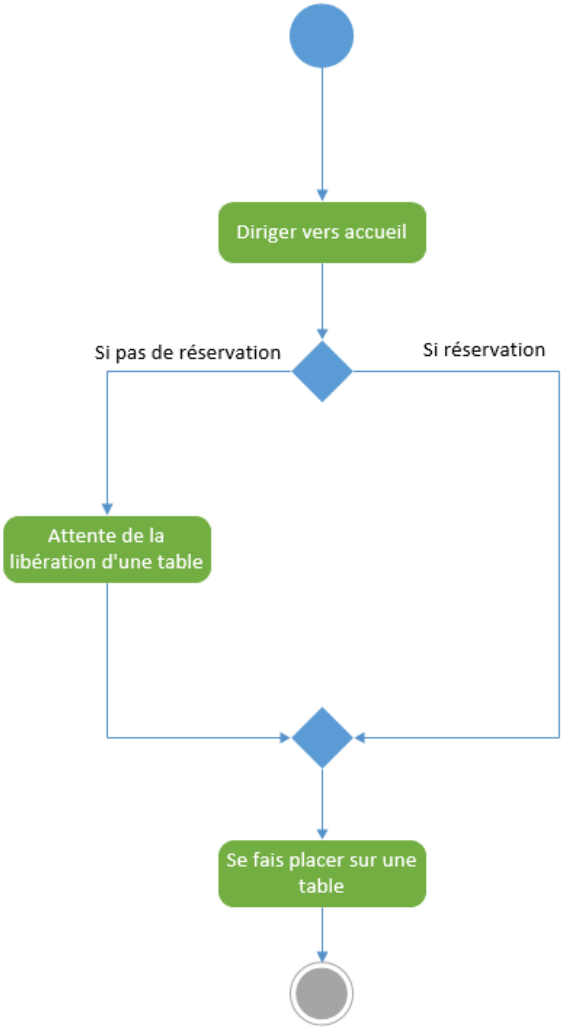


Figure 4 - Diagramme d'activités - Client - Placement

Lorsqu'un client entre dans le restaurant, il se dirige vers l'accueil.

S'il dispose d'une réservation, il peut directement être admis à table.

Dans le cas où il ne dispose pas de réservation, il doit attendre qu'une table adéquate se présente.

Ensuite, le client peut se faire placer sur une table.

Client – Commande :

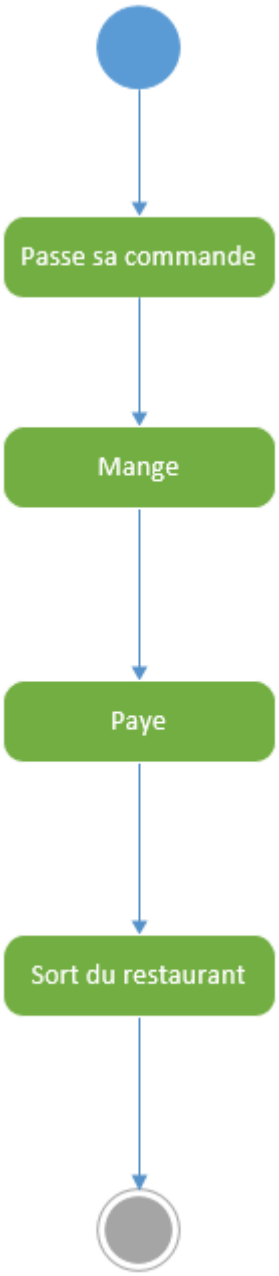


Figure 5 - Diagramme d'activités - Client-Commande

Le diagramme d'activité nous présente ici le rôle du client au niveau de la commande. Il va dans un premier temps passer sa commande, il va ensuite manger puis payer et enfin, il sort du restaurant.

Maitre-hôtel – Paiement :

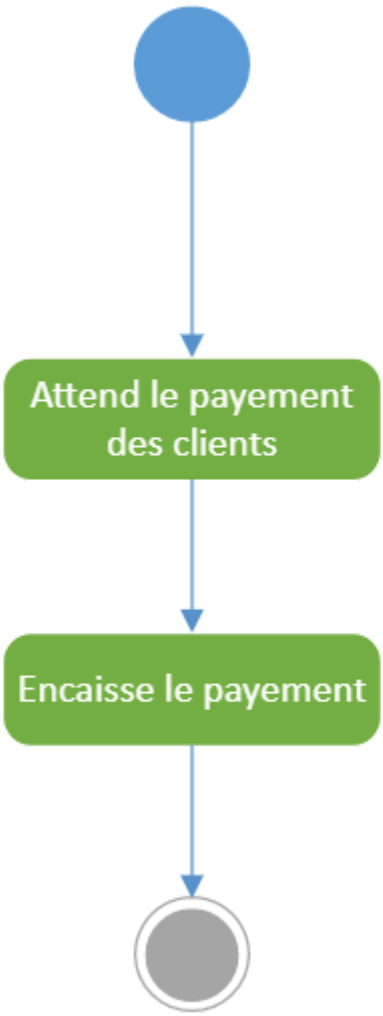


Figure 6 - Diagramme d'activités - Maître-hôtel- Paiement

Le diagramme d'activité nous présente ici le travail du maître d'hôtel lors du paiement d'un client, il attend à l'accueil le paiement des clients puis l'encaisse.

Maitre-hôtel – Placement :

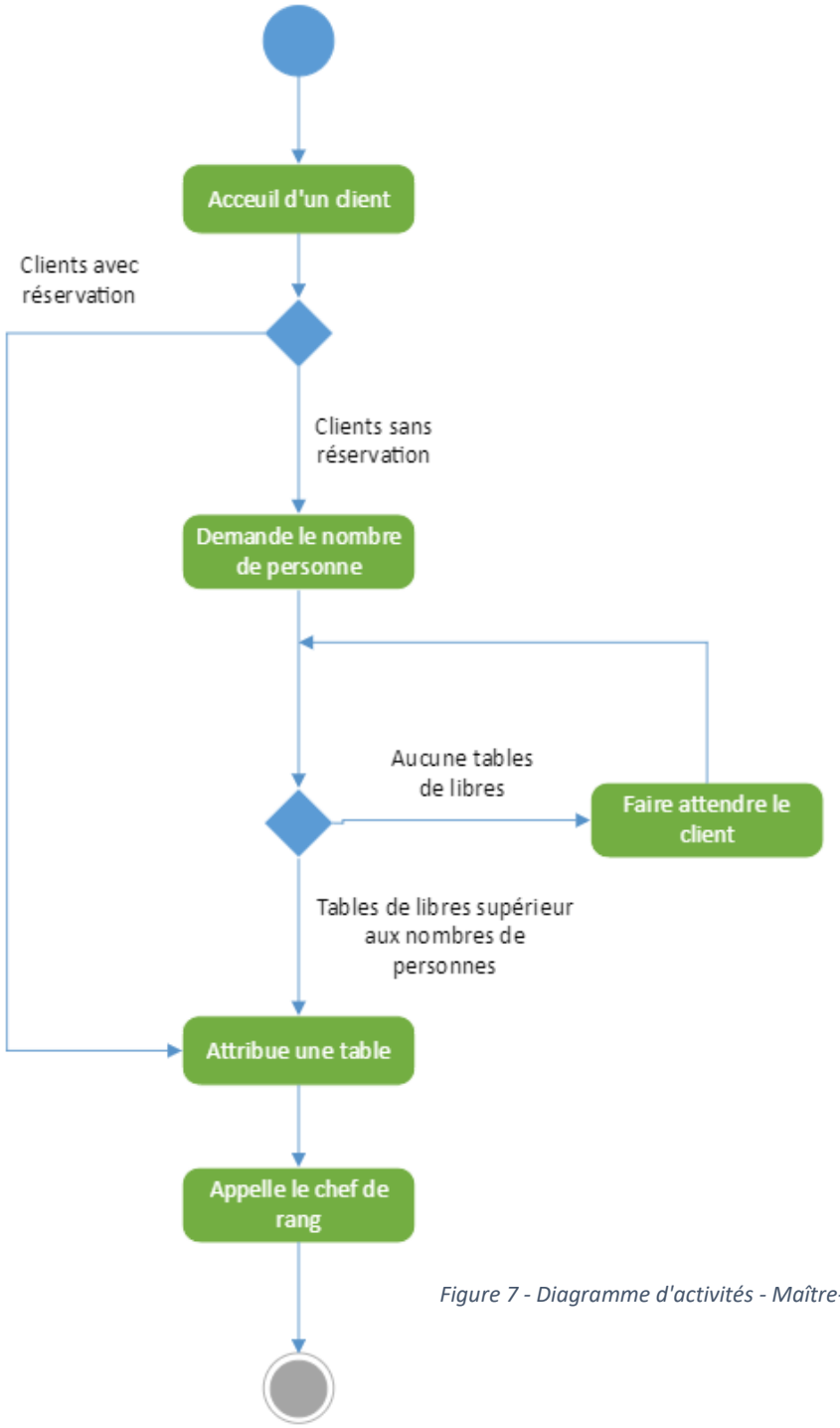


Figure 7 - Diagramme d'activités - Maître-hôtel - Placement

Le diagramme d'activité nous présente ici le travail du maître d'hôtel lors de l'arrivée d'un client. Il accueille le client et si celui-ci a réservé, il lui attribue une table et appelle le chef de rang. Sinon il demande le nombre de personne afin de leur attribuer une table. Si aucune table n'est disponible, il les fait patienter jusqu'à ce qu'il y ait une table de libre.

Chef de rang – Placement :

Figure 8 - Diagramme d'activités - Chef de rang - Placement

Le diagramme d'activité nous présente ici le travail du chef de rang au niveau du placement. Le chef de rang va aller chercher les clients à l'entrée qui ont déjà été accueillis par le maître d'hôtel et va ensuite les placer sur la table attribuée par ce dernier.

Chef de rang – Commande :

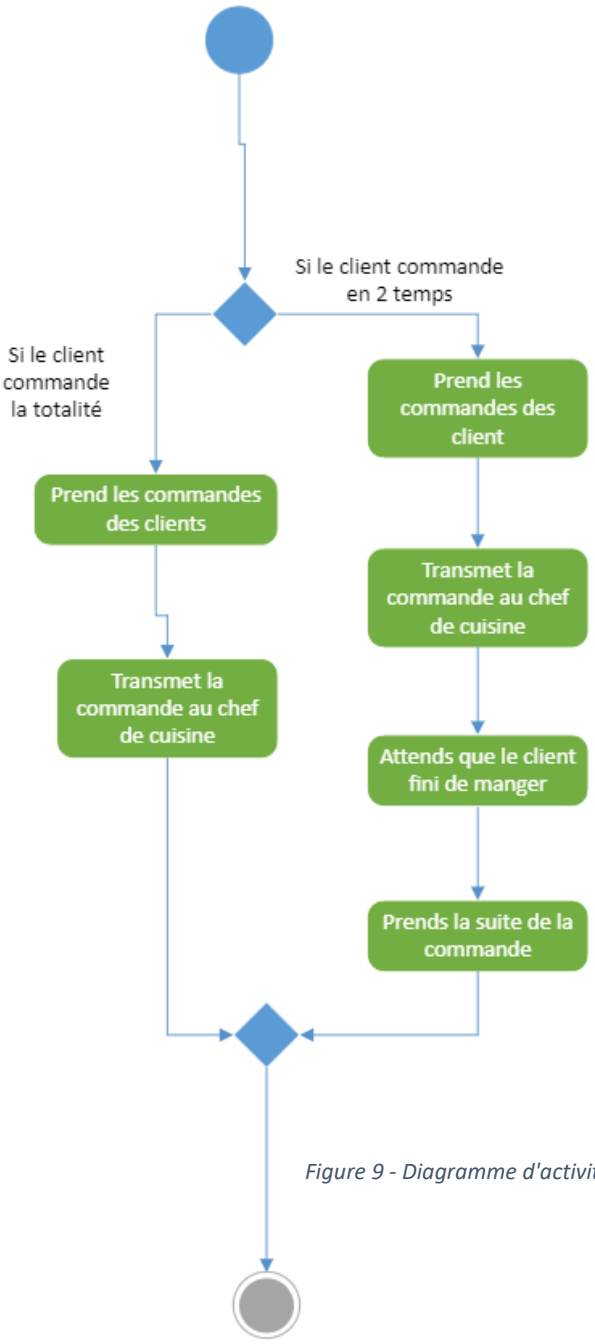


Figure 9 - Diagramme d'activités - Chef de rang - Commande

Le diagramme d'activité nous présente ici le travail du chef de rang au niveau des commandes. Il va dans un premier temps prendre la commande des clients pour ensuite les transmettre au chef de cuisine. Dans le cas où le client commande en deux fois, il va attendre que le client ait fini de manger pour prendre la suite de la commande.

Chef de rang – Carte :

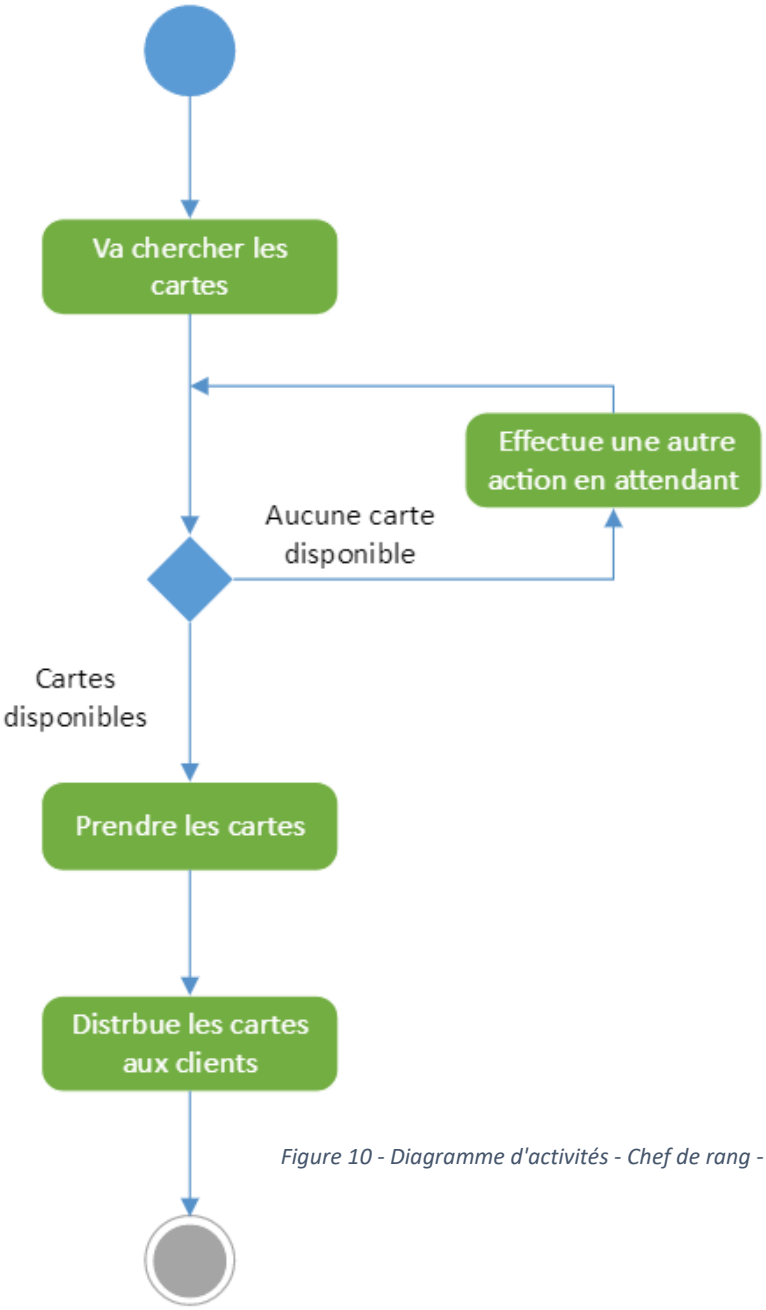


Figure 10 - Diagramme d'activités - Chef de rang - Carte

Le diagramme d'activité nous présente ici le travail du chef de rang au niveau des cartes. Il va dans un premier temps aller chercher les cartes à l'endroit où elles sont stockées, ensuite s'il y a des cartes disponibles, il les prendra pour les distribuer aux clients. Dans le cas où il n'y a pas de cartes de disponibles, il va effectuer une autre action et revenir plus tard pour refaire la routine précédente.

Serveur – Service :

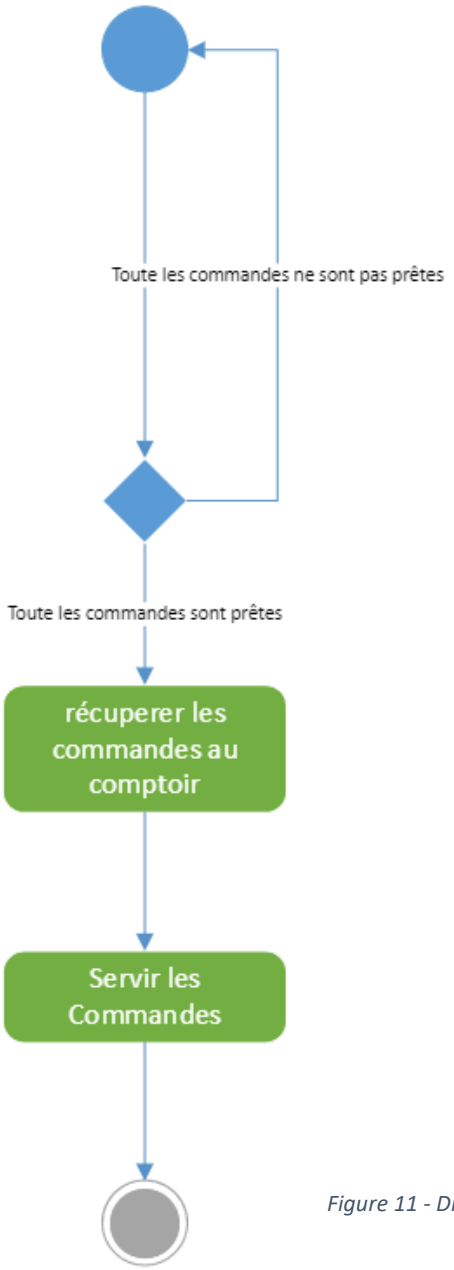


Figure 11 - Diagramme d'activités - Serveur - Service

Le diagramme d'activité nous présente ici le travail du serveur au niveau du service. Il va dans un premier temps vérifier si toute les commandes de la table sont prêtes, si c'est le cas, le serveur va récupérer les commandes au comptoir pour ensuite servir les clients sinon, il retourne à son état initial.

Serveur – Débarrasser :

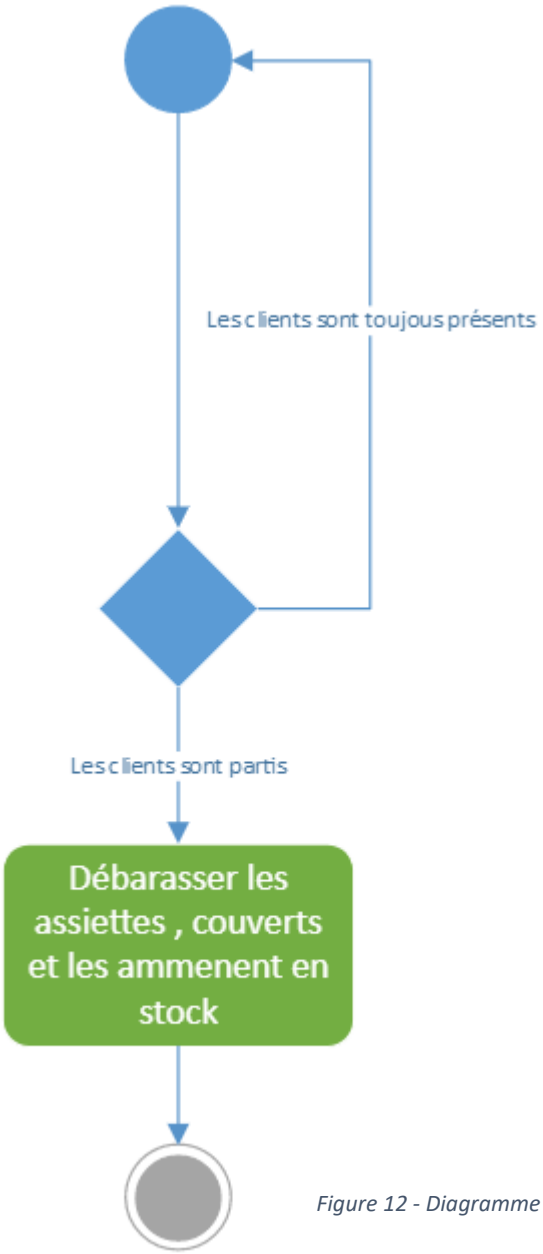


Figure 12 - Diagramme d'activités - Serveur - Débarrasser

Le diagramme d'activité nous présente ici le travail du serveur au niveau du débarrassage des tables. Il va dans un premier temps vérifier si les clients sont partis, si c'est le cas, le serveur va débarrasser la table et va stocker la vaisselle sale dans un endroit de stockage prévu à cet effet, sinon, il retourne à son état initial.

Commis salle :

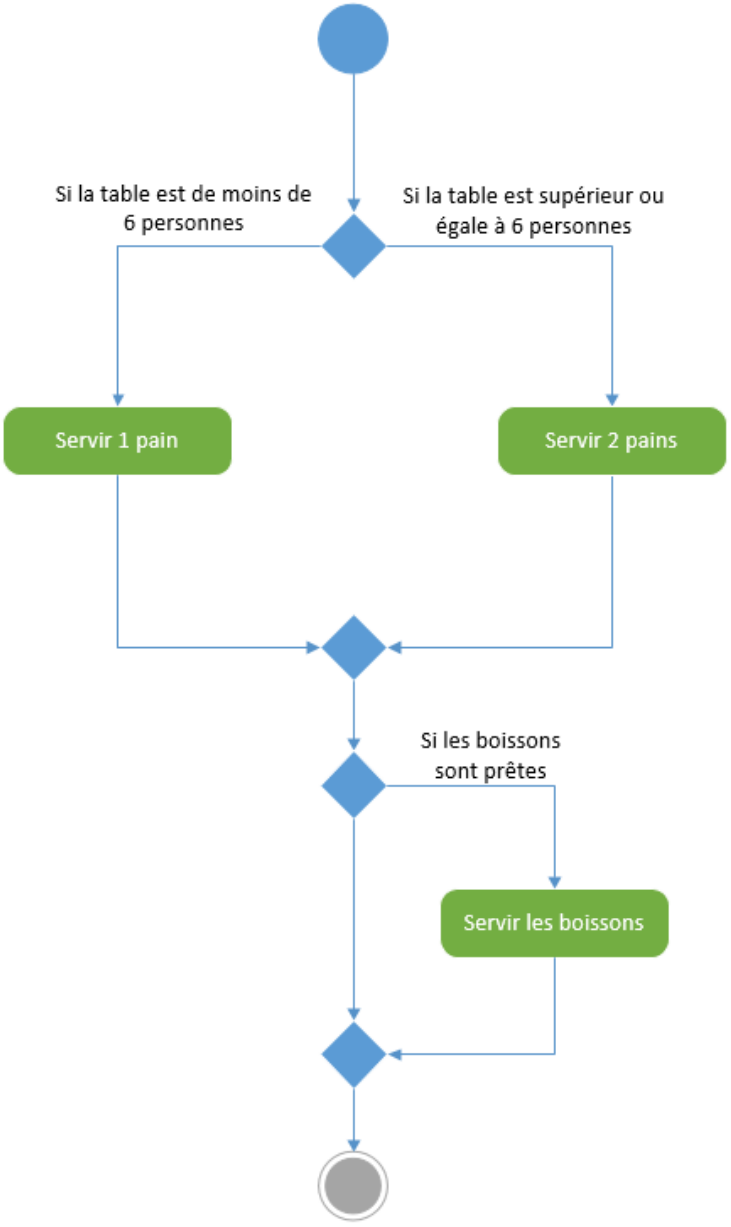


Figure 13 - Diagramme d'activités - Commis salle

Le diagramme d'activité nous présente ici le travail du commis de salle au niveau du débarrassage des tables. Il va dans un premier temps vérifier si la table comporte plus ou moins de 6 personnes, si c'est le cas, le commis va servir 2 pains sur la table, sinon, il sert 1 seul pain. Ensuite, il va vérifier si les boissons sont prêtes, si c'est le cas, il va servir les boissons sur la table, sinon, il va terminer son action.

Chef de cuisine :

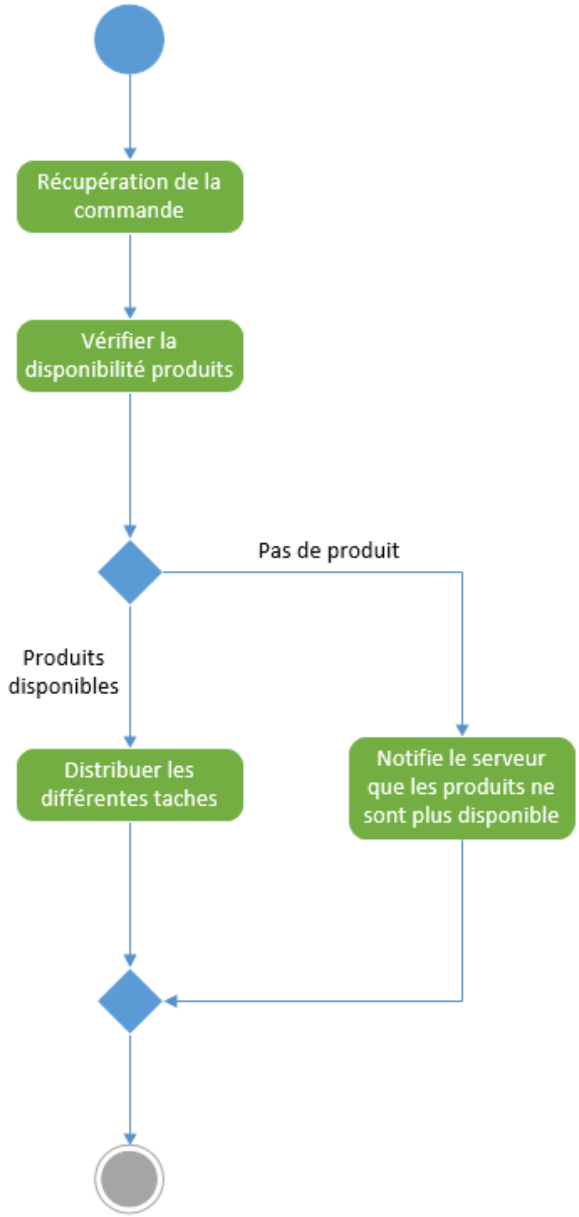


Figure 14 - Diagramme d'activités - Chef de cuisine

Le diagramme d'activité nous présente ici le travail du chef de cuisine. Celui-ci aura en charge de récupérer les commandes apportées par le serveur puis il aura la charge de vérifier si les produits sont disponibles ou non. Dans le cas où il n'y a plus de produit il avertira le serveur puis commencera un autre travail. Cependant si les produits sont disponibles il distribuera les tâches aux cuisiniers.

Cuisinier :

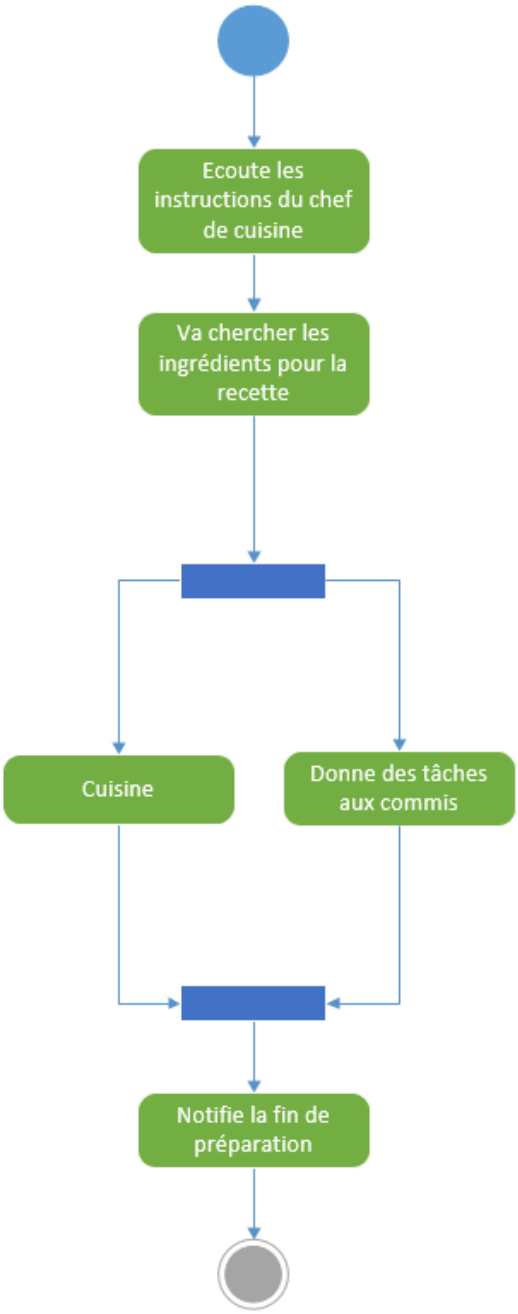


Figure 15 - Diagramme d'activités - Cuisinier

Le diagramme d'activité nous présente ici le travail du chef de partie. Directement sous les directives du chef de cuisine, il va donc écouter les instructions puis aller chercher les ingrédients utiles à la recette. Une fois qu'il aura les ingrédients, il va pouvoir confier des tâches aux commis, mais pourra aussi cuisiner au même moment. Finalement, il aura pour tâche d'avertir la fin de la préparation.

Commis cuisine :

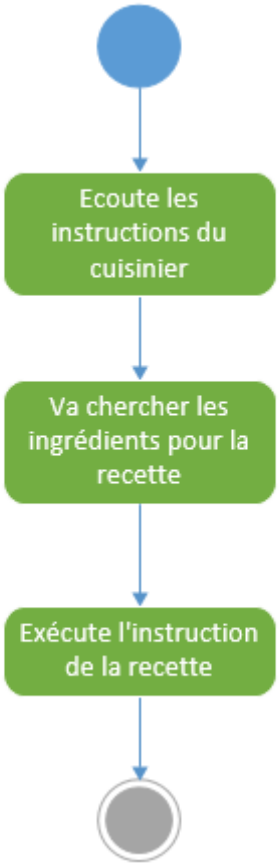


Figure 16 - Diagramme d'activités - Commis cuisine

Le commis de cuisine est sous les ordres du chef de partie. Quand ce dernier lui donnera une instruction le commis devra la réaliser, et ce, de la même manière que son supérieur.

Plongeur – Lave-vaisselle

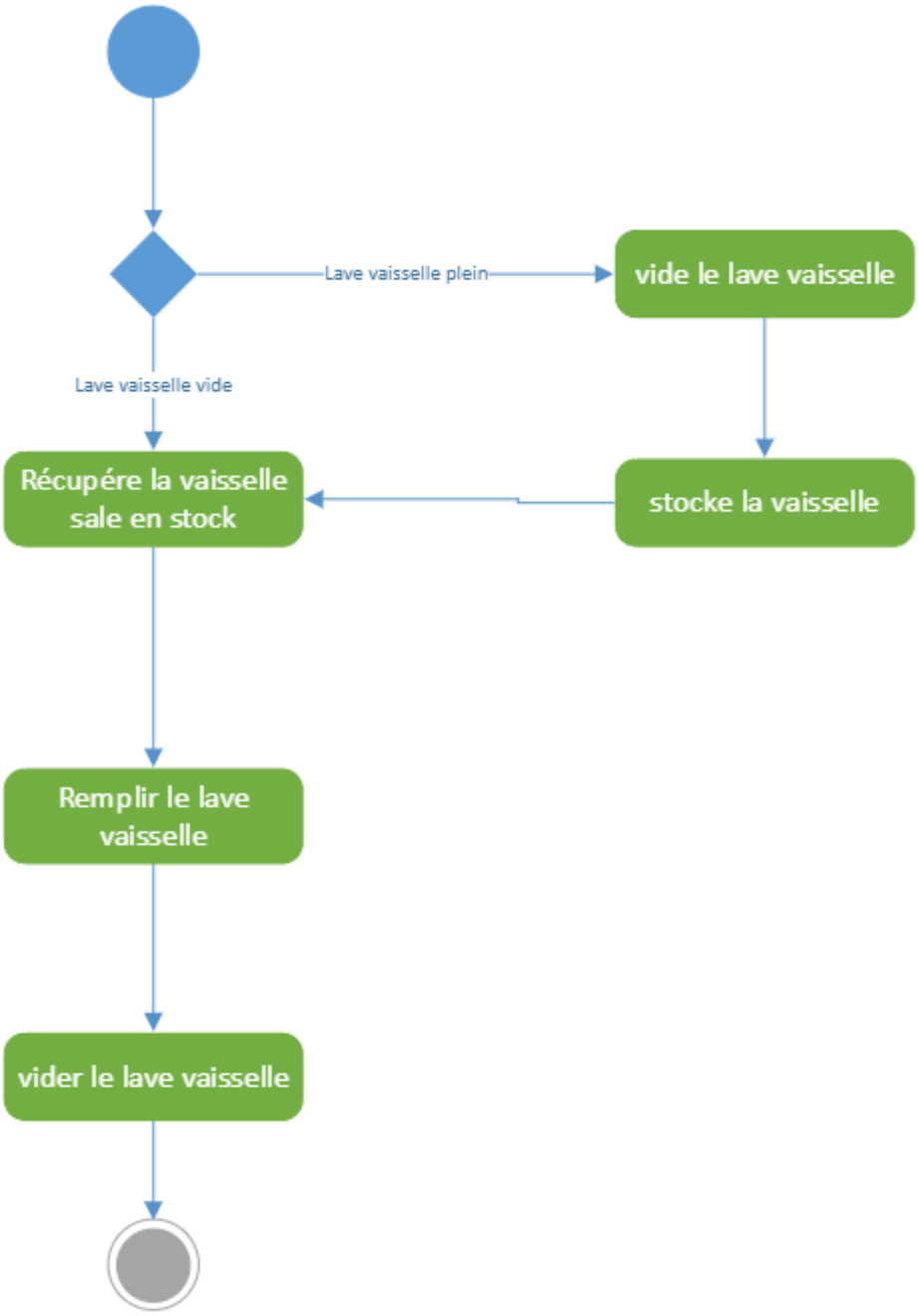


Figure 17 - Diagramme d'activités - Plongeur - Lave-vaisselle

Le diagramme d'activité nous présente ici le travail du plongeur au niveau de l'utilisation du lave-vaisselle. Il va dans un premier temps vérifier si le lave-vaisselle est vide, si c'est le cas, le plongeur va récupérer la vaisselle sale en stock puis remplir le lave-vaisselle et enfin le vider dès que celui-ci a terminé son cycle, sinon, il vide le lave-vaisselle et stock la vaisselle propre pour ensuite continuer son cycle.

Plongeur – Légume

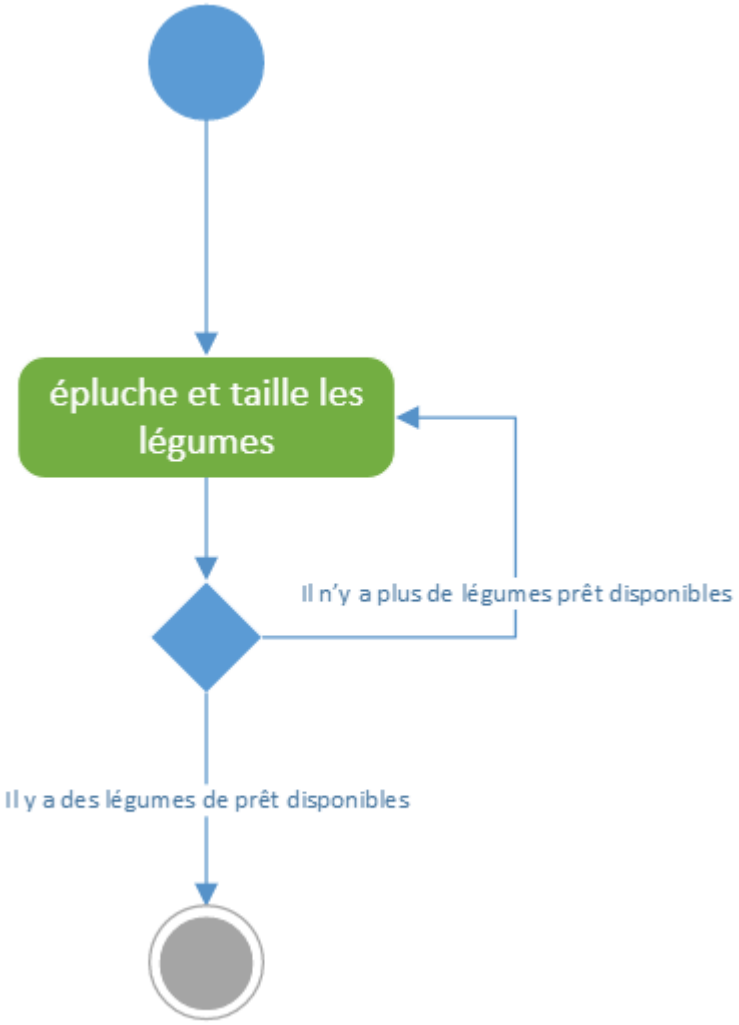


Figure 18 - Diagramme d'activités - Plongeur - Légume

Le diagramme d'activité nous présente ici le travail du plongeur au niveau de l'épluchage des légumes, Il va dans un premier temps vérifier s'il y a des légumes prêts de disponible. Si c'est le cas, le plongeur va terminer son action sinon, il épluche et taille les légumes pour les cuisines.

c. Diagrammes de composants

Si dessus le diagramme de composant du point de vue de l'application. Un tel diagramme permet de définir les composants et leurs dépendances dans l'environnement. On retrouvera :

- Les descriptions des implémentations du système
- Des groupes d'implémentations
- Des relations entre les diverses implémentations

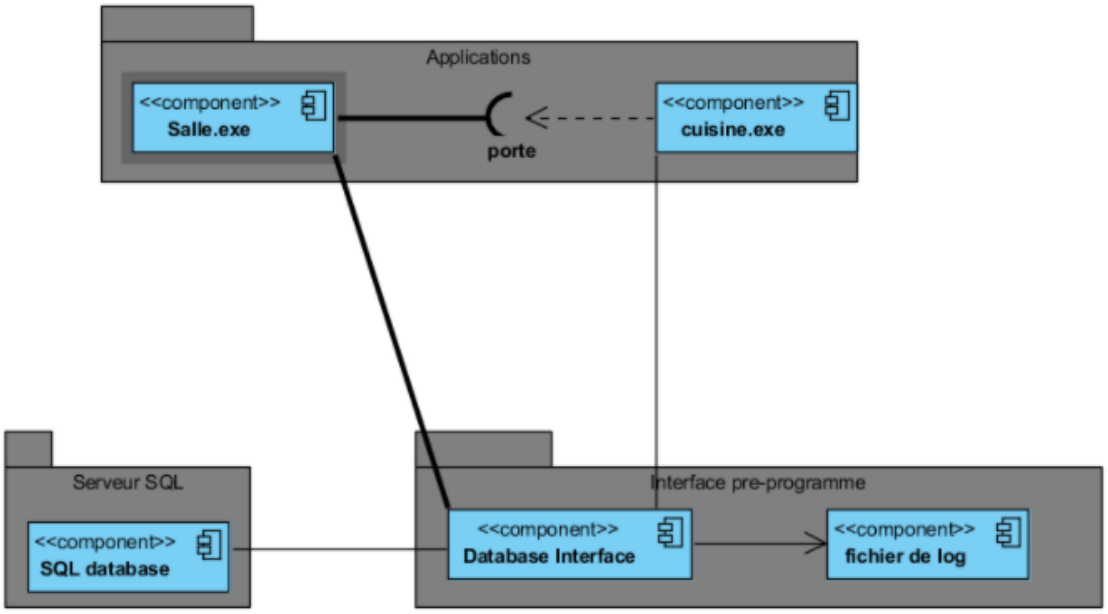


Figure 19 - Diagramme de composants

Ainsi, dans celui proposé, nous allons retrouver ce qui compose nos applications. Les application salle et cuisine communique en eux et cuisine est dépendant de salle pour avoir sens. En effet, s'il n'y a pas de commande ni de couvert et nappe à laver, la cuisine n'a pas de travail.

De plus, ces deux applications ont besoin d'avoir accès à la base de données, mais, au lieu de lire et d'écrire directement dedans, nous allons créer une interface afin de pouvoir travailler en déconnecter.

Finalement, les appels à la base de données seront stockés dans un fichier de log.

d. Diagrammes de classes

Un diagramme de classe décrit les responsabilités, le comportement et le type d'un ensemble d'objets.

Salle restaurant :

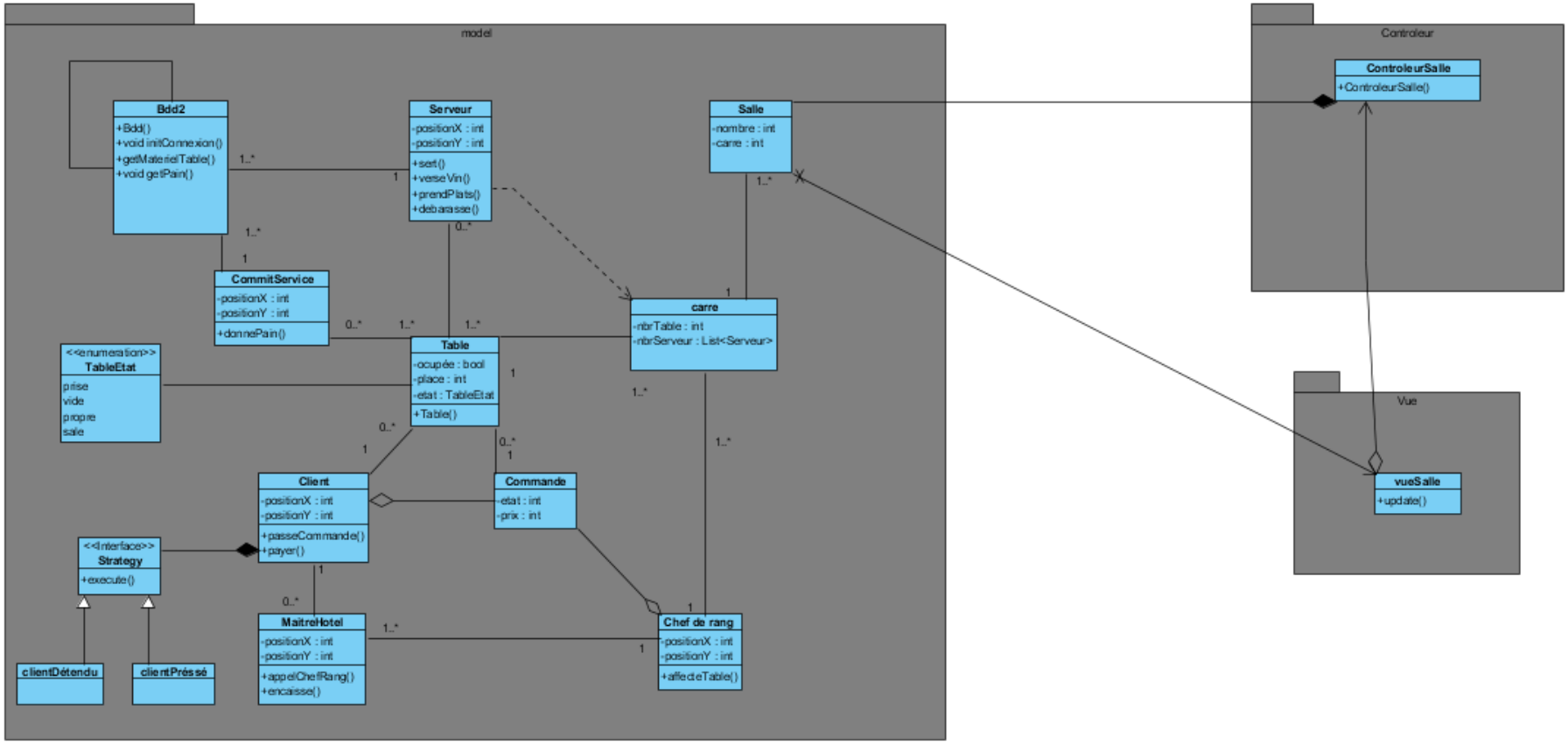
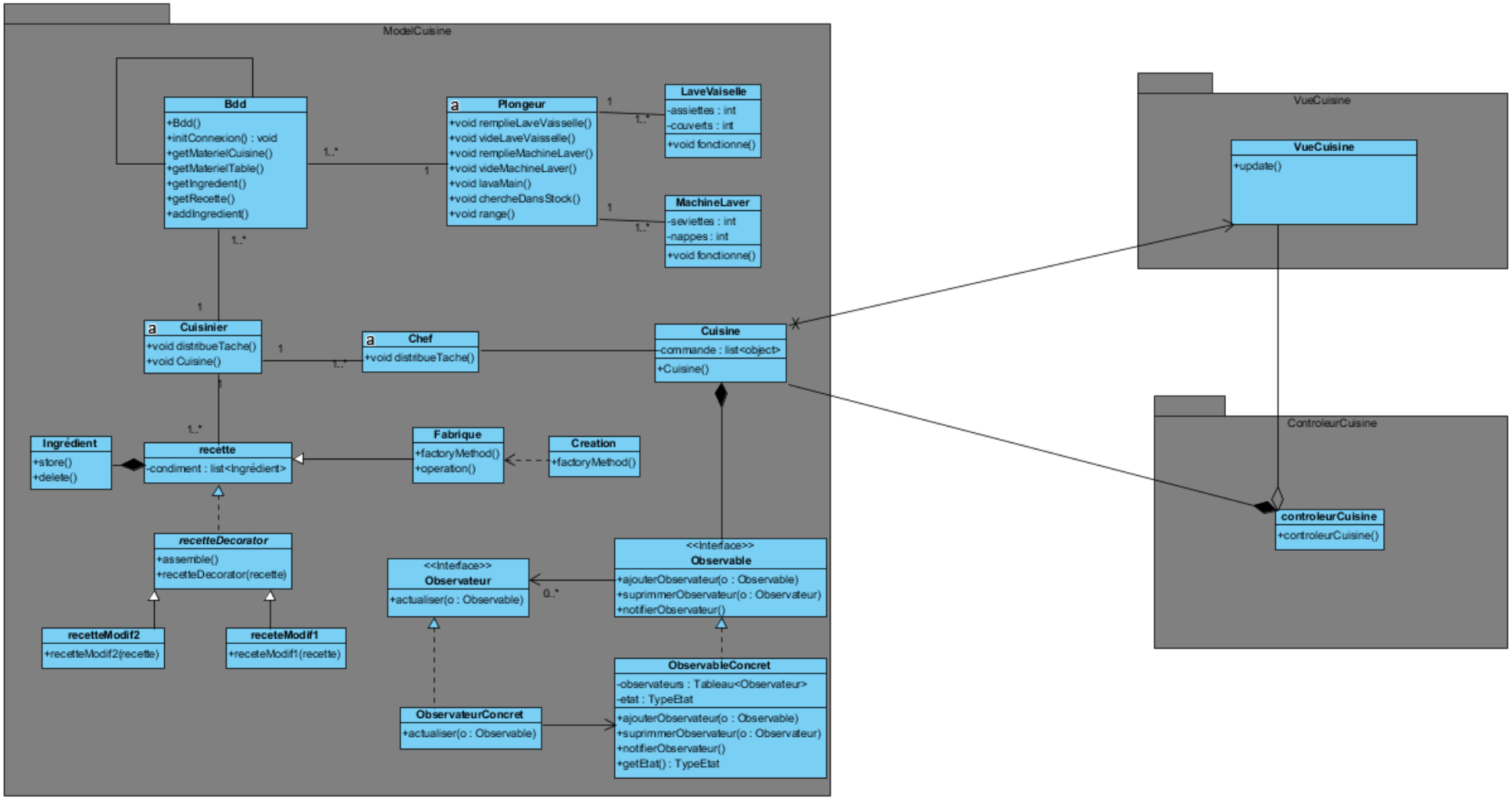


Figure 20 - Diagramme de classes - Salle restaurant

Cuisine



Figur

e. Diagrammes de séquences

Les diagrammes de séquences permettent de représenter les interactions entre les acteurs et le système et ce, dans un ordre chronologique.

Application :

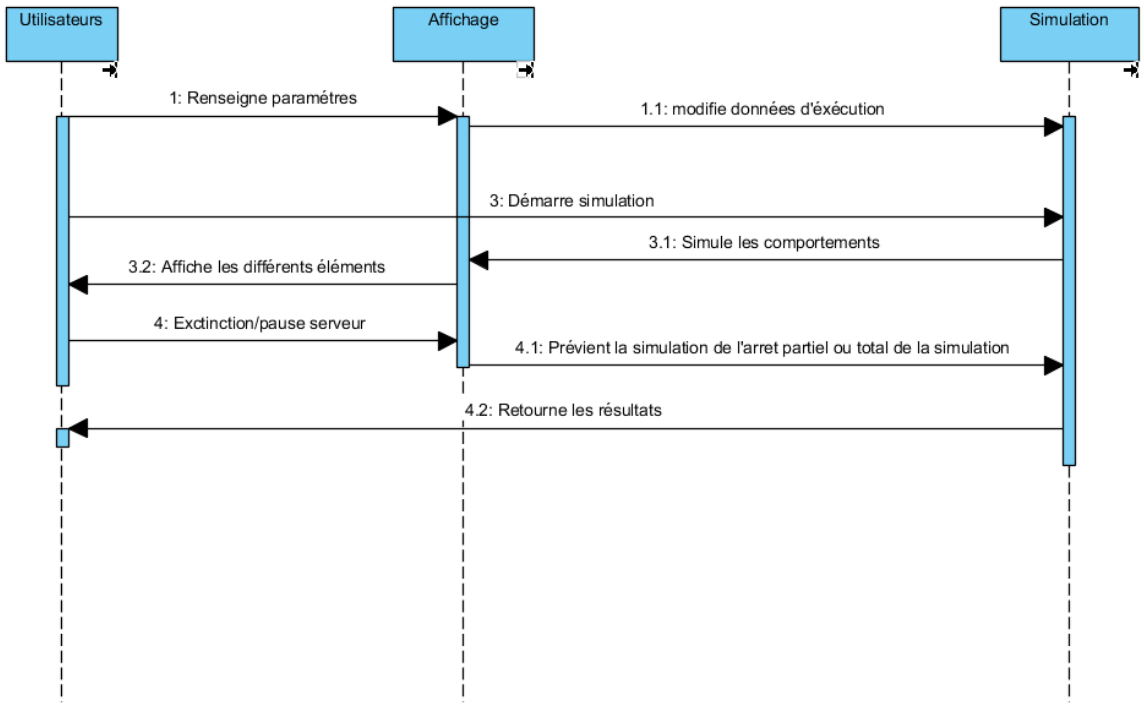


Figure 22 - Diagramme de séquences - Application

Ici le diagramme représente l'intégralité du programme. On y retrouve en 1er lieux les utilisateurs de l'application qui renseigneront des paramètres afin que le programme puisse modifier les paramètres. Quand l'utilisateurs à sélectionner les caractéristiques il peut démarrer la simulation que se déroulera et qui à la fin renverra les résultats.

Entre temps le client peut modifier la vitesse de l'application

Salle restaurant :

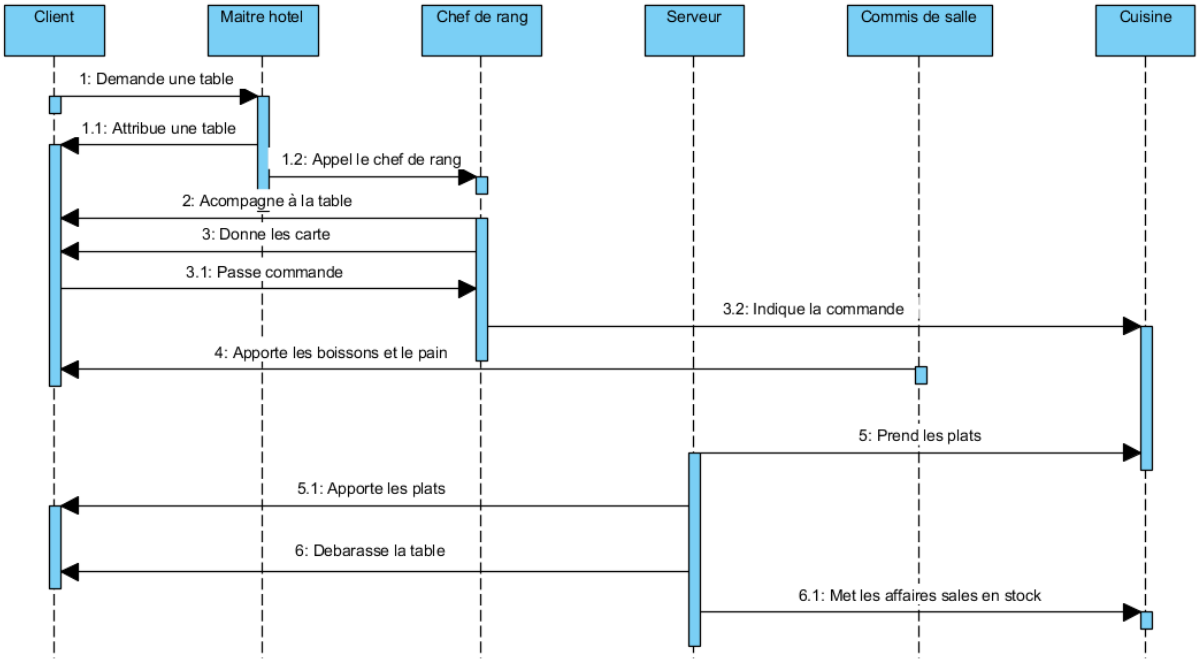


Figure 23 - Diagramme de séquences - Salle restaurant

Ce diagramme quant à lui nous présente le fonctionnement de la salle. Du point de vue du client, il va demander une table au maitre d'hôtel qui lui en affectera une. Au même moment de l'affectation, le maitre d'hôtel va demander au serveur de préparer la table si ce n'est pas encore le cas puis le serveur va distribuer carte, pain et eau. Après un temps de réflexion le client passe commande. Le serveur va envoyer la commande en cuisine qui sera préparer. Au même moment le commis ira servir les boissons. A la fin de la préparation le serveur prendra les plats e les emmènera à la table. Une fois le repas terminé le serveur débarrasse la table et met les affaires en stock.

Cuisine :

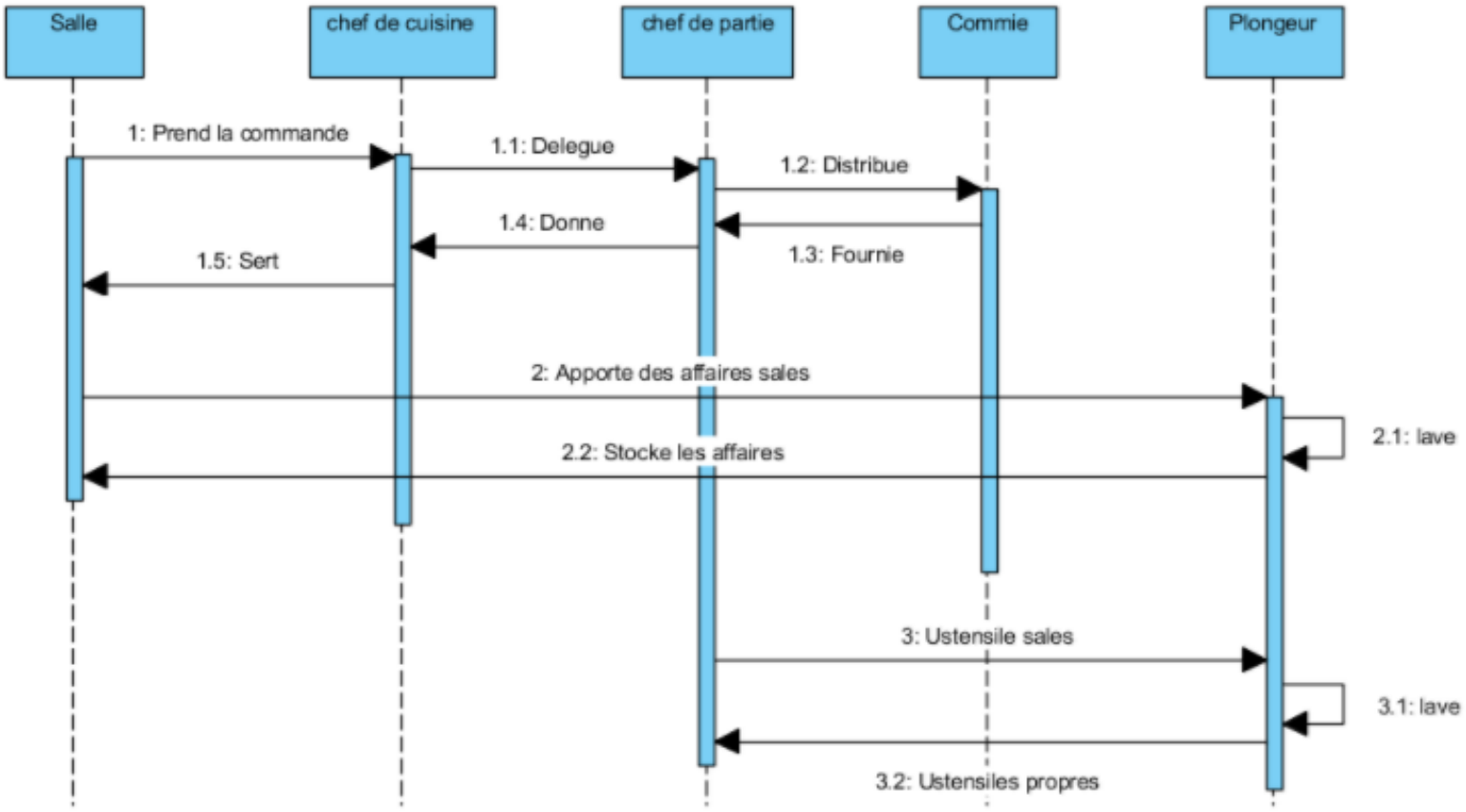


Figure 24 - Diagramme de séquences - Cuisine

Ce diagramme de séquence nous révèle le fonctionnement de la cuisine. On peut distinguer 3 parties :

- La 1ère qui consiste en la réception, conception et service du plat commandé par un client en salle

- La 2ème qui est pour le plongeur la tache de nettoyage des couverts ainsi que la nappe utilisée lors d'un précédent repas

- La 3ème qui ressemble à la précédente mais qui cette fois ci s'applique sur la batterie utilisée par les cuisiniers

II. Détails et explications des DP utilisés

a. Design pattern singleton

L'objectif du singleton est de ne permettre l'instanciation d'une classe à un seul objet. Concrètement cela se traduit par une classe contenant une méthode. Cette dernière crée une instance s'il n'en existe pas. Sinon elle renvoie la référence de l'objet.

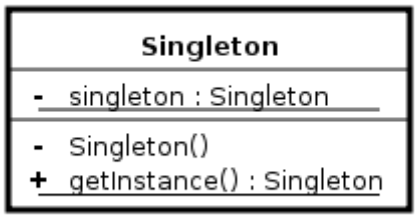


Figure 25 - Design Pattern - Singleton

Pour le projet nous allons utiliser ce design pattern à plusieurs reprises. En effet il peut nous servir pour la connexion à la base de données, création d'un rôle unique comme le maitre d'hôtel.

b. Design pattern factory

Le patron de conception factory va permettre d'instancier des objets à partir d'un type abstrait.

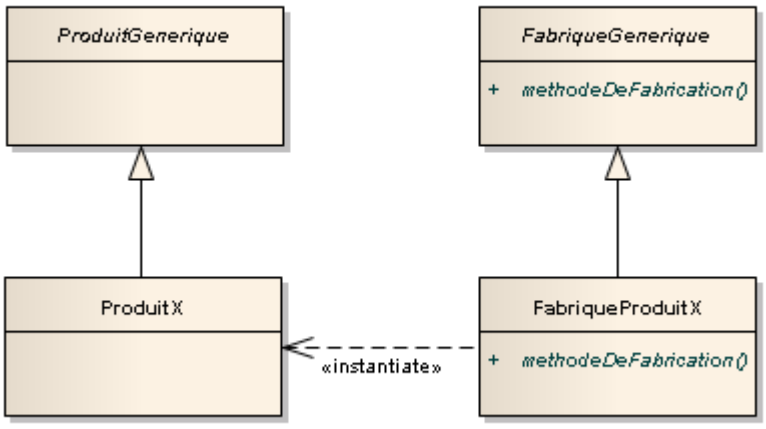


Figure 26 - Design Pattern - Factory

Dans notre solution, nous allons l'utiliser afin d'instancier plusieurs plats.

Design pattern decorator

Il va permettre d'ajouter des fonctionnalités à une classe et, contrairement à l'héritage, de manière dynamique.

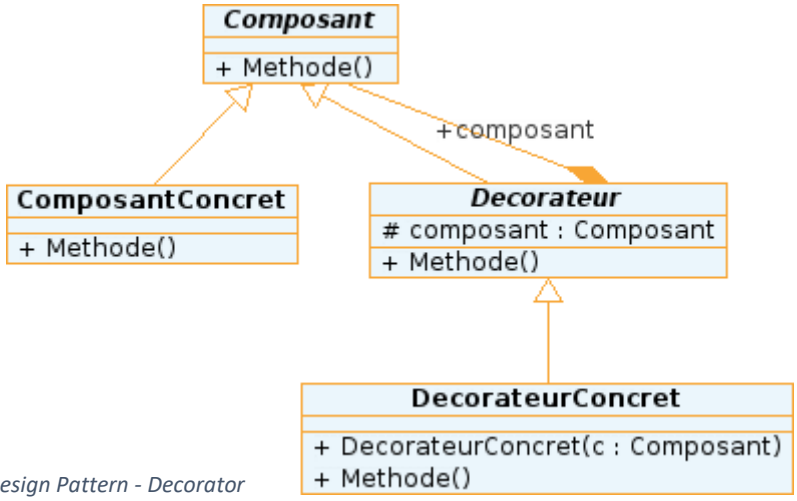


Figure 27 - Design Pattern - Decorator

L'utilisation d'un tel design pattern peut être intéressant si des plats se ressemble. Sur une carte des desserts il est possible de voir une crêpe à laquelle on y ajouter des ingrédients tel que de la chantilly et ce, grâce à un decorator.

c. Design pattern iterator

La conception de se design pattern va permettre d'accéder aux éléments d'une liste de manière itérative.

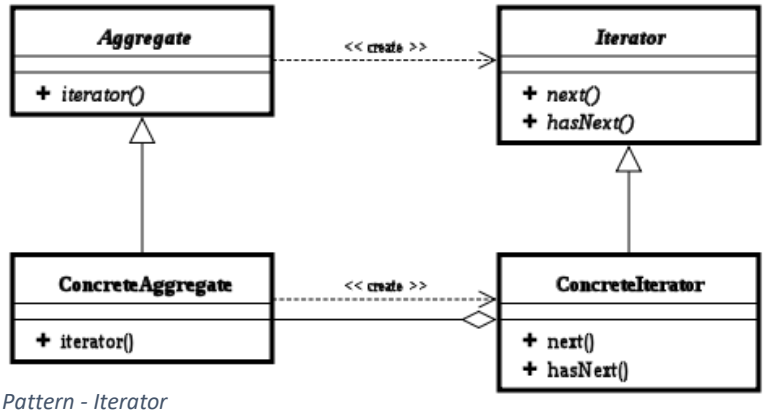


Figure 28 - Design Pattern - Iterator

Ce design pattern va permettre de savoir la commande suivante.

d. Design pattern strategy

Ce design pattern va permettre de créer des objets qui, selon le comportement, change.

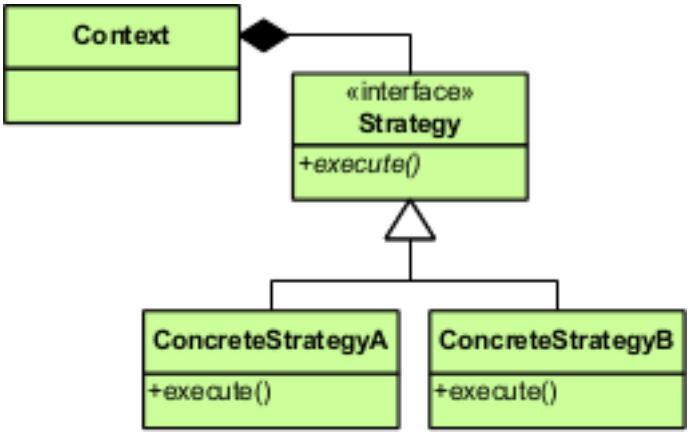


Figure 29 - Design Pattern - Strategy

Pour notre projet, il va nous servir pour créer différent client, celons leurs gouts.

e. Design pattern observer

Le design pattern observer va permettre de voir si un objet observé est modifié.

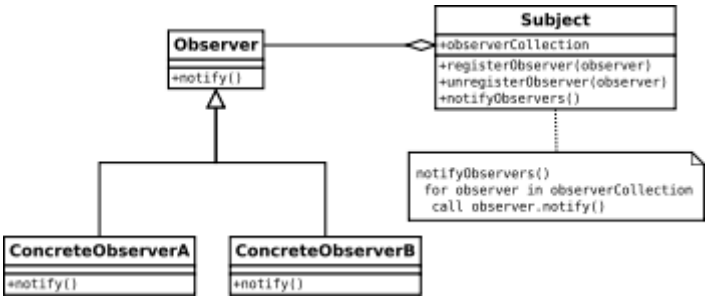


Figure 30 - Design Pattern - Observer

Cela va nous servir pour savoir si la cuisine à fini ou non un plat

f. Design pattern MVC

Ce design pattern signifie Model vue contrôleur :

- La vue correspond à une IHM
- Le model va, généralement, contenir les donné. C'est ce que l'on nomme la couche métier
- Le contrôleur va faire le lien entre la vue et le model. Il va contrôler les donner.

Il est le résultat de plusieurs autres design pattern. Observer, stratégie et composite

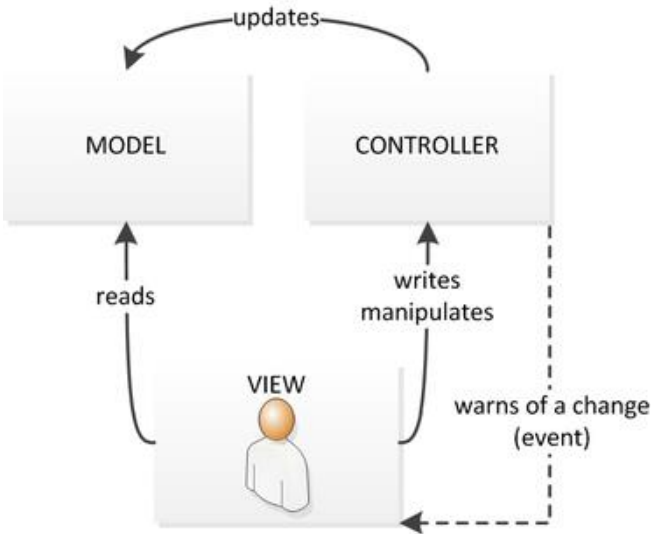


Figure 31 - Design Pattern - MVC

L'utilisation de ce design pattern va nous permettre de mieux structurer notre code du fait de la séparation entre la vue et le contrôleur

III. MDC et script SQL

a. MCD

Afin de réaliser notre projet, nous avons dû élaborer un MCD pour la gestion de la base de données. Un MCD (Modèle conceptuel de données) est une représentation logique de l'organisation des données et des relations entre elles. Son objectif est de décrire de manière formelle les données utilisées par le SI. Un MCD est composé par :

- Les entités
- Il s'agit de la représentation d'un élément possédant un rôle dans le système que l'on souhaite décrire (exemple : ingrédients)
- Une entité compote une ou plusieurs propriétés qui décrive les données stockées

- Les relations
- Il s'agit d'un lien entre les entités
- Peut être relié à elle-même (réflexive) ou à 2 entités (binaires)

- Les cardinalités
- Donne une indication sur le nombre minimum et maximum ou une entité peut participer à une association

Dans notre projet, voici le MCD élaboré grâce au logiciel Visual Paradigm :

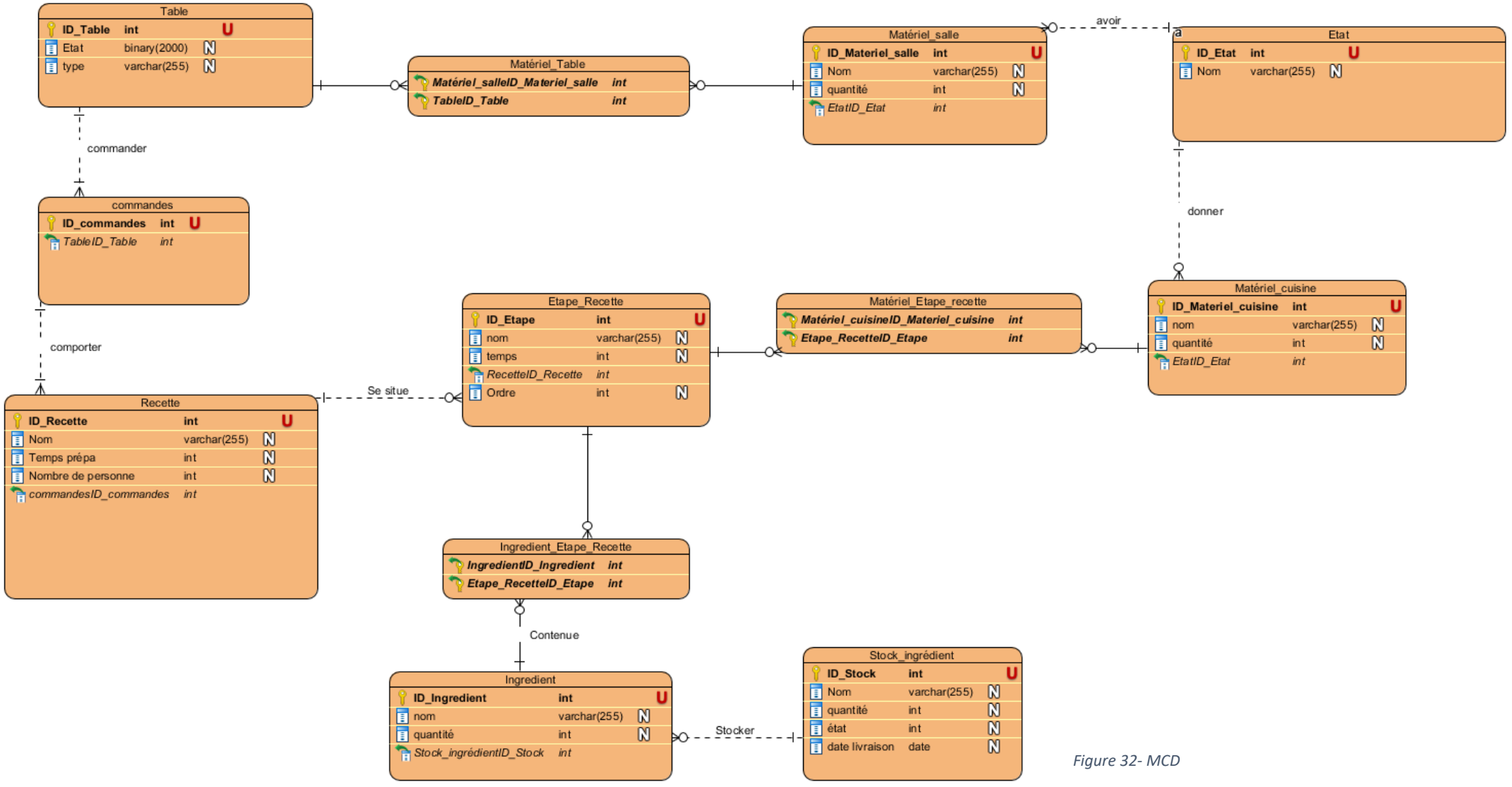


Figure 32- MCD

b. Script SQL

Le Logiciel permet également de générer un script SQL de notre MCD afin de remplir notre base de données sur SQL server.

```
use restaurant;

CREATE TABLE commandes (ID_commandes int NOT NULL, TableID_Table int NOT NULL, PRIMARY KEY (ID_commandes));

CREATE TABLE Etape_Recette (ID_Etape int NOT NULL, nom varchar(255) NULL, temps int NULL, RecetteID_Recette int NOT NULL, Ordre int NULL, PRIMARY KEY (ID_Etape));

CREATE TABLE Etat (ID_Etat int IDENTITY NOT NULL, Nom varchar(255) NULL, PRIMARY KEY (ID_Etat));

CREATE TABLE Ingredient (ID_Ingredient int NOT NULL, nom varchar(255) NULL, quantité int NULL, Stock_ingrédientID_Stock int NOT NULL, PRIMARY KEY (ID_Ingredient));

CREATE TABLE Ingredient_Etape_Recette (IngredientID_Ingredient int NOT NULL, Etape_RecetteID_Etape int NOT NULL, PRIMARY KEY (IngredientID_Ingredient, Etape_RecetteID_Etape));

CREATE TABLE Matériel_cuisine (ID_Materiel_cuisine int NOT NULL, nom varchar(255) NULL, quantité int NULL, EtatID_Etat int NOT NULL, PRIMARY KEY (ID_Materiel_cuisine));

CREATE TABLE Matériel_Etape_recette (Matériel_cuisineID_Materiel_cuisine int NOT NULL, Etape_RecetteID_Etape int NOT NULL, PRIMARY KEY (Matériel_cuisineID_Materiel_cuisine, Etape_RecetteID_Etape));

CREATE TABLE Matériel_salle (ID_Materiel_salle int NOT NULL, Nom varchar(255) NULL, quantité int NULL, EtatID_Etat int NOT NULL, PRIMARY KEY (ID_Materiel_salle));

CREATE TABLE Matériel_Table (Matériel_salleID_Materiel_salle int NOT NULL, TableID_Table int NOT NULL, PRIMARY KEY (Matériel_salleID_Materiel_salle, TableID_Table));

CREATE TABLE Recette (ID_Recette int NOT NULL, Nom varchar(255) NULL, [Temps prépa] int NULL, [Nombre de personne] int NULL, commandesID_commandes int, PRIMARY KEY (ID_Recette));

CREATE TABLE Stock_ingrédient (ID_Stock int NOT NULL, Nom varchar(255) NULL, quantité int NULL, état int NULL, [date livraison] date NULL, PRIMARY KEY (ID_Stock));

CREATE TABLE [Table] (ID_Table int NOT NULL, Etat binary(2000) NULL, type varchar(255) NULL, PRIMARY KEY (ID_Table));

ALTER TABLE Ingredient_Etape_Recette ADD CONSTRAINT FKIngredient829388 FOREIGN KEY (Etape_RecetteID_Etape) REFERENCES Etape_Recette (ID_Etape);

ALTER TABLE Matériel_Etape_recette ADD CONSTRAINT FKMatériel_E519515 FOREIGN KEY (Matériel_cuisineID_Materiel_cuisine) REFERENCES Matériel_cuisine (ID_Materiel_cuisine);

ALTER TABLE Matériel_Etape_recette ADD CONSTRAINT FKMatériel_E639671 FOREIGN KEY (Etape_RecetteID_Etape) REFERENCES Etape_Recette (ID_Etape);

ALTER TABLE Matériel_Table ADD CONSTRAINT FKMatériel_T959896 FOREIGN KEY (Matériel_salleID_Materiel_salle) REFERENCES Matériel_salle (ID_Materiel_salle);

ALTER TABLE Matériel_Table ADD CONSTRAINT FKMatériel_T550983 FOREIGN KEY (TableID_Table) REFERENCES [Table] (ID_Table);

ALTER TABLE Matériel_salle ADD CONSTRAINT avoir FOREIGN KEY (EtatID_Etat) REFERENCES Etat (ID_Etat);

ALTER TABLE commandes ADD CONSTRAINT commander FOREIGN KEY (TableID_Table) REFERENCES [Table] (ID_Table);

ALTER TABLE Recette ADD CONSTRAINT comporter FOREIGN KEY (commandesID_commandes) REFERENCES commandes (ID_commandes);

ALTER TABLE Ingredient_Etape_Recette ADD CONSTRAINT Contenue FOREIGN KEY (IngredientID_Ingredient) REFERENCES Ingredient (ID_Ingredient);

ALTER TABLE Matériel_cuisine ADD CONSTRAINT donner FOREIGN KEY (EtatID_Etat) REFERENCES Etat (ID_Etat);

ALTER TABLE Etape_Recette ADD CONSTRAINT [Se situe] FOREIGN KEY (RecetteID_Recette) REFERENCES Recette (ID_Recette);

ALTER TABLE Ingredient ADD CONSTRAINT Stocker FOREIGN KEY (Stock_ingrédientID_Stock) REFERENCES Stock_ingrédient (ID_Stock);
```