

ARCHITECTURE LOGICIEL

PROJET PROGRAMMATION CONCURENTE

Groupe 5 :

- KENGNI MIRES
- KOUEDOU MOUKAM CINDY LAURA
- MAKONGNE DEFFO JULIE MERVEILLE
- SIGNE TATSING CAMPBELL
- WEKO BRAYAN

SUPERVEURS :

- Mr DOMINGO PALAO
- Mr YOUNBI BRUCE

Année académique :
2024-2025

Table des matières

INTRODUCTION.....	2
PARTIE I- DESCRIPTION DES DIAGRAMMES DE CONCEPTION.....	3
I- DIAGRAMME D'ACTIVITE	3
1.MAITRE D'HOTEL.....	3
2. CHEF DE RANG	3
3.SERVEUR	4
4.COMMIS DE SALLE.....	4
5.CLIENT	4
6.CHEF DE CUISINE	5
7.CHEF DE PARTIE / CUISINIER.....	5
8.COMMIS DE CUISINE	5
9.PLONGEUR.....	5
II- DIAGRAMMES DE CAS D'UTILISATION.....	5
1.SALLE DE RESTAURATION.....	5
2.CUISINE.....	6
III- DIAGRAMMES DE SEQUENCE	7
1.SALLE DE RESTAURANT	7
2.CUISINE.....	7
IV- MODELE CONCEPTUEL DE DONNEES	9
PARTIE II: DESCRIPTION DES DESIGN PATTERN UTILISES.....	13
CONCLUSION.....	16

INTRODUCTION

Dans le cadre de la conception logicielle d'un système destiné à gérer les activités d'un restaurant, ce rapport a pour objectif de présenter une architecture détaillée et cohérente, reposant sur des principes fondamentaux d'ingénierie logicielle. Le développement d'un tel système nécessite une compréhension approfondie des processus métier, ainsi qu'une modélisation adaptée pour garantir l'efficacité et la fluidité des opérations. Ainsi, ce document propose une analyse des différents diagrammes de conception, allant des diagrammes d'activités aux modèles conceptuels de données. Chaque section met en lumière les interactions, les responsabilités et les flux d'information au sein du restaurant, que ce soit en salle ou en cuisine. Ce rapport vise à fournir une vision globale et détaillée des mécanismes sous-jacents pour une implémentation logicielle robuste et adaptée. Pour la création de ces diagrammes, nous avons utilisé **Visual Paradigm**, un logiciel puissant qui facilite la modélisation UML et la conception de systèmes. Grâce à ses fonctionnalités intuitives, Visual Paradigm permet de générer des diagrammes clairs et professionnels, favorisant ainsi une collaboration efficace et une documentation précise.

PARTIE I – PRESENTATION DE DIAGRAMMES DE CONCEPTION

I- DIAGRAMME D'ACTIVITE

Un **diagramme d'activité** est un type de diagramme UML (**Unified Modeling Language**) utilisée en modélisation pour décrire le déroulement d'un processus ou d'une série d'actions. Il met en évidence les différentes étapes, leurs enchaînements, les décisions à prendre, ainsi que les interactions entre les acteurs impliqués. Ce type de diagramme est particulièrement utile pour comprendre et optimiser les flux de travail. Dans le cadre de notre projet, le diagramme d'activité sera utilisé pour illustrer les tâches spécifiques de chaque employé du restaurant, telles que la prise de commandes, la préparation des plats, ou encore le service en salle. Il permettra de clarifier les responsabilités de chacun et d'assurer une coordination efficace entre les différentes fonctions au sein de l'établissement.

1. MAITRE D'HOTEL

Le maître d'hôtel, véritable coordinateur de la salle de restaurant, est chargé de l'accueil des clients, représentant ainsi la première impression que ces derniers auront de l'établissement. Dans le cadre de notre projet, il attribue une table aux clients et appelle le chef de rang pour les installer. Ce rôle est fixe et ne varie pas dans la salle. À son arrivée, le maître d'hôtel accueille les clients, leur demande le nombre de personnes et s'assure de la présence d'une réservation. Il attribue ensuite une table adaptée au nombre de convives, en veillant à ce que sa capacité soit suffisante. Responsable de la gestion des places, il prend les décisions nécessaires pour optimiser l'occupation de la salle, afin de maximiser le nombre de couverts. Enfin, il a également pour responsabilité de faire régler les clients à la fin de leur repas.

2. CHEF DE RANG

Le chef de rang gère un ensemble de tables regroupées dans un secteur spécifique et agit sous l'autorité du maître d'hôtel. Il peut également superviser les commis affectés à son secteur. Pour notre projet, nous simplifions en attribuant un chef de rang unique par carré. Sa mission

principale est d'accueillir les clients, de les installer à leur table et de leur fournir les menus. Il prend ensuite les commandes, en une ou plusieurs étapes, avant de les transmettre au chef de cuisine. Son déplacement se limite entre le carré qu'il gère et la zone du maître d'hôtel. Lorsqu'il n'a pas de tâches immédiates, il peut assister un collègue, rester à l'accueil ou attendre dans son carré. Enfin, une fois les clients partis et la table nettoyée par le serveur, il prépare la table pour accueillir de nouveaux convives.

3. SERVEUR

Après avoir pris connaissance des commandes des clients, le serveur dépose sur la table le pain et l'eau en quantité ajustée au nombre de convives. Ensuite, il se positionne près du comptoir où les plats sont préparés, prêt à les récupérer pour les servir à la table correspondante. Il veille à ce que les plats identiques soient apportés simultanément. Pendant le repas, le serveur reste à la disposition des clients pour répondre à leurs éventuelles requêtes. Une fois le repas terminé, il débarrasse la table et transporte la vaisselle sale vers une zone désignée, en respectant une limite de cinq éléments à la fois.

Le serveur assure un service fluide pour les tables de la zone qu'il gère, contribuant à la satisfaction des clients. Après chaque plat, il récupère les assiettes et couverts pour les déposer à un point de collecte, où la cuisine s'occupe du nettoyage. Il se déplace uniquement dans la zone qui lui est attribuée, mais en l'absence de clients, il peut prêter main-forte aux collègues des zones voisines. Chaque zone est supervisée par deux serveurs.

4. COMMIS DE SALLE

Le commis de salle travaille exclusivement dans la salle, avec pour principales responsabilités la distribution du pain et des boissons aux clients. Il veille également à ce qu'aucun besoin ne soit laissé sans réponse, s'assurant que les clients disposent de tout ce qu'il leur faut pour leur repas. En tant qu'unique commis dans le restaurant, il intervient pour compléter le service des serveurs, en fournissant les éléments manquants ou en répondant aux demandes supplémentaires des clients.

5. CLIENT

Il peut choisir de réserver une table à l'avance ou de se présenter directement sans réservation. À son arrivée, c'est le maître d'hôtel qui lui attribue une table, tandis que le chef de rang s'occupe de l'installer. Une fois à table, le client passe sa commande, qui sera ensuite servie par un serveur. Chaque client peut avoir des préférences ou des comportements différents et pourra solliciter le serveur pendant le repas pour toute demande. Une fois le repas terminé, il effectue le paiement de l'addition auprès du maître d'hôtel.

6. CHEF DE CUISINE

Le chef de cuisine est responsable de la gestion globale de la cuisine. Il assure la coordination et la supervision de toutes les activités, en attribuant les différentes tâches aux chefs de parties. C'est lui qui garantit le bon déroulement du service pour que les plats arrivent à temps et dans de bonnes conditions pour les clients. Il n'y a qu'un seul chef de cuisine dans l'établissement. Le sous-chef, ou second, aide le chef dans ses fonctions et prend en charge ses responsabilités en son absence. Il occupe souvent également le rôle de chef de partie. Pour des raisons de simplification, la présence du sous-chef ne sera pas représentée dans ce modèle.

7. CHEF DE PARTIE / CUISINIER

Les chefs de partie sont responsables d'une section spécifique de la cuisine et supervisent les commis, apprentis et stagiaires travaillant sous leur direction. Le nombre de chefs de partie peut varier en fonction de la taille du restaurant, allant jusqu'à six dans les grandes brigades pour gérer les domaines comme les sauces, la rôisserie, le poisson, les entremets, le garde-manger et la pâtisserie. Ce sont eux qui préparent les plats que les clients dégusteront. Dans notre restaurant, il y a deux chefs de partie qui se partagent l'ensemble des tâches en cuisine.

8. COMMIS DE CUISINE

Le commis de cuisine soutient les chefs en allant chercher les ingrédients nécessaires dans le stockage, en préparant les fruits et légumes (comme les éplucher), et en fournissant les plats déjà préparés au comptoir correspondant pour la suite du service.

9. PLONGEUR

Les plongeurs et aides de cuisine jouent un rôle essentiel en s'occupant du nettoyage du matériel de cuisine ainsi que de la vaisselle. Ils peuvent également être amenés à nettoyer certaines zones de la cuisine ou à aider à la préparation des légumes, comme l'épluchage et la découpe. Dans notre établissement, un plongeur est présent pour assurer ces tâches.

II- DIAGRAMMES DE CAS D'UTILISATION

1- SALLE DE RESTAURATION

Dans la salle de restauration, le fonctionnement est orchestré de manière coordonnée pour garantir une expérience fluide aux clients. Lorsqu'un groupe arrive, il est accueilli par le

maître d'hôtel, qui leur attribue une table adaptée. Le chef de rang prend ensuite le relais en installant les clients et en leur présentant les menus disponibles. Une fois les choix effectués, il enregistre la commande et la transmet à la cuisine pour préparation. Pendant ce temps, le serveur met en place les éléments de base, comme le pain et l'eau, puis reste disponible pour d'éventuelles demandes. Tout au long du repas, le serveur veille à débarrasser la table après chaque plat et reste attentif aux besoins des clients. Une fois le repas terminé, le règlement est effectué directement auprès du maître d'hôtel. Ce processus reflète une organisation où chaque membre du personnel a un rôle précis pour assurer un service efficace.

2- CUISINE

Dans la cuisine, l'organisation est essentielle pour garantir une préparation efficace des repas. Avant l'arrivée des clients, une partie des plats, notamment les desserts, est préparée en avance pour anticiper les commandes. Lorsque celles-ci arrivent, le chef de cuisine joue un rôle central en répartissant les tâches entre les différents cuisiniers, en fonction de leurs compétences et de la complexité des plats. Chaque cuisinier s'appuie sur les ingrédients fournis ou préalablement préparés par le commis, tout en utilisant les ustensiles disponibles. Une fois la tâche terminée, les outils utilisés sont libérés et transmis au plongeur. Ce dernier assure leur nettoyage ainsi que celui de la vaisselle et du linge sale, garantissant ainsi que tous les équipements restent opérationnels pour les prochaines préparations. Ce fonctionnement fluide reflète une coordination précise entre les différents membres de l'équipe en cuisine.

Tableau 1: Etapes du Diagramme de cas d'utilisation

Acteur	Description	Cas d'utilisation
Client	Personne qui consomme des plats et des boissons au restaurant.	- Passer une commande - Consulter le menu - Faire une réservation
Maitre d'hôtel	Responsable de l'accueil des clients et de la supervision du personnel de salle.	- Gérer les réservations

Chef de rang	Serveur principal d'une section, chargé de prendre les commandes.	-Prendre la commande - Servir les plats
Serveur	Membre de l'équipe de service, responsable de la livraison des plats.	- Apporter les commandes - Encaisser les paiements
Commis de salle	Aide les serveurs et le chef de rang, s'occupant de la mise en place.	- Préparer les tables - Aider au service
Chef de cuisine	Responsable de l'ensemble de la cuisine et de la gestion du personnel.	- Élaborer le menu - Superviser la cuisine
Chef de partie cuisine	Responsable d'une section spécifique de la cuisine.	- Préparer les plats - Superviser les commis de cuisine
Commis de cuisine	Aide le chef de cuisine dans la préparation des plats.	- Préparer les ingrédients - Suivre les instructions des chefs
Plongeur	Chargé de la vaisselle et de la propreté	-Nettoyer les ustencils

III- DIAGRAMME DE SEQUENCE

1. SALLE DE RESTAURATION

Dès leur arrivée, les clients sont pris en charge par le maître d'hôtel, qui leur assigne une table convenant à leurs besoins. Le chef de rang s'occupe ensuite de leur installation et leur présente les différentes options de menus. Après que les clients ont fait leur choix, le chef de rang saisit leur commande et l'envoie en cuisine pour la préparation. Pendant ce temps, le serveur anticipe leurs besoins en disposant les indispensables tels que le pain et l'eau, tout en restant attentif aux éventuelles requêtes. Durant le repas, il veille au bon déroulement en débarrassant la table après chaque plat et en répondant aux attentes des convives. Enfin, lorsque le repas touche à sa fin, le maître d'hôtel gère le règlement de l'addition. Ce schéma de fonctionnement met en lumière une organisation rigoureuse où chaque membre de l'équipe contribue activement à garantir un service fluide et satisfaisant.

2. CUISISNE

Le diagramme de séquence modélise l'ordre d'exécution des interactions entre les objets du système dans un scénario spécifique. Il met en évidence le temps et la chronologie des échanges.

Tableau 2: Etape du Diagramme de Sequence

Elément	Description
Acteur	Client, Serveur, Chef de rang, Maitre d'hôtel, Systèmes de gestions de la cuisine
Processus	Passer une commande

❓ **Acteurs** : Les acteurs impliqués dans le processus sont le client, le serveur, le chef de rang, et le système de gestion. Chaque acteur joue un rôle spécifique dans le passage de la commande.

❓ **Flux de Messages** : Les interactions sont représentées par des flèches qui indiquent l'ordre dans lequel les messages sont échangés. Chaque message représente une action ou une demande entre les acteurs et le système.

❓ **Chronologie**: Le tableau suit une séquence chronologique, montrant comment chaque action est liée à la précédente et comment elle affecte le flux global de la commande.

IV- DIAGRAMME DES COMPOSANTS

Le diagramme de composants illustre l'architecture physique du système de gestion d'un système d'organisation. Il montre comment les différents composants logiciels interagissent entre eux et comment ils sont déployés dans l'environnement.

Table 3: Etape du Diagramme de Composant

Composant	Role
Maitre d'hotel	Supervise l'accueil et le service, gère les réservations, assure la satisfaction des clients.
Chef de rang	Responsable d'une section, prend les commandes, supervise le service des serveurs.
Serveur	Sert les plats, prend les commandes, gère les paiements des clients.

Commis de salle	Aide à préparer les tables, soutient les serveurs pendant le service.
Chef partie cuisine	Prépare les plats dans sa section, supervise les commis de cuisine.
Plongeur	Gère la propreté de la vaisselle et des ustensiles dans la cuisine.
Chef de cuisine	Responsable de l'ensemble de la cuisine, élabore le menu, supervise le personnel de cuisine.
Commis de cuisine	Aide à la préparation des plats, suit les instructions du chef de cuisine.

🔍 **Vision d'ensemble** : Ce diagramme permet de visualiser comment chaque rôle contribue au fonctionnement du restaurant.

🔍 **Coordination** : Il montre les relations de supervision et de collaboration entre les différents membres du personnel, facilitant ainsi la gestion des opérations.

🔍 **Clarté des Rôles** : En définissant clairement les responsabilités, le diagramme aide à éviter les confusions et à assurer une communication efficace au sein de l'équipe.

Table 4: Etapes Diagramme d'activités

Etapes	Description
1- Client	Le client arrive au restaurant.
2- Consulte le menu	Le client examine les options disponibles.
3- Prends une décision	Le client décide de commander ou de poser des questions.
4- Serveur prend la commande	Le serveur note la commande du client.
5- Envoie la commande	Le serveur transmet la commande au système de gestion.
6- Enregistre la commande	Le système enregistre les détails de la commande.
7- Notifie le chef de rang	Le système informe le chef de rang de la nouvelle commande.
8- Transmet à la cuisine	Le chef de rang envoie la commande à la cuisine.

9- Cuisine les plats	L'équipe de cuisine commence à préparer les plats commandés.
10- Préparation terminée	Les plats sont prêts à être servis.
11- Récupération des plats	Les plats sont prêts à être servis.
12- Serveur sert les plats	Les plats sont prêts à être servis.
13- Le client déguste le repas	Le client apprécie son repas
14- Addition	Le client apprécie son repas
15- Prépare l'addition	Le serveur prépare et calcule l'addition.
16- Règle l'addition	Le client paye pour son repas.
17- Quitte le restaurant	Le client quitte le restaurant, complétant ainsi le processus.

V-MODELE CONCEPTUEL DE DONNEES

La base de données **restaurantmanagement** est conçue pour gérer efficacement les informations liées à la gestion d'un restaurant, y compris les clients, les employés, les ingrédients, les commandes et les recettes. Voici une explication détaillée des différentes tables et de leur rôle dans le système.

Clients de table

La table clients stocke les informations relatives aux clients du restaurant. Chaque client est identifié par un **ClientID** unique, qui est une clé primaire. Les colonnes incluent le nom complet (**FullName**), l'identifiant de réservation (**ReservationID**), ainsi que les heures d'arrivée et de départ (**ArrivalTime**, **DepartureTime**). Ce tableau permet de suivre les détails des visites des clients, facilitant ainsi la gestion des réservations et des interactions personnalisées.

Employés de table

La table **employees** est utilisée pour gérer les informations des employés. Chaque employé est identifié par un **EmployeeID** unique. Les colonnes stockent des détails tels que le prénom (**FirstName**), le nom (**LastName**), le poste occupé (**Position**), le salaire (**Salary**) et la date

d'embauche (**HireDate**). Cette structure permet une gestion centralisée des ressources humaines, avec un suivi clair des rôles et des rémunérations.

Ingrédients de la table

La table **ingredients** centralise les informations relatives aux ingrédients disponibles dans la cuisine. Chaque ingrédient est identifié par un **IngredientsID** unique. Les informations incluent le nom de l'ingrédient (**NameItem**), l'unité de mesure (**Unit**), et la quantité disponible (**Quantity**). Ce tableau est essentiel pour suivre l'inventaire et garantir que les stocks sont suffisants pour la préparation des plats.

Table de cuisine

La table **kitchen** gère les informations liées à l'équipement de cuisine. Chaque équipement est identifié par un **KitchenID** unique. Les colonnes incluent le nom de l'équipement (**EquipmentName**), la capacité maximale (**MaxCapacity**), et le statut de disponibilité (**Status**) qui peut être "Available", "In Use", ou "Maintenance". Cette table facilite la gestion des équipements et leur maintenance.

Commandes à table

La table **orders** permet de suivre les commandes des clients. Chaque commande est identifiée par un **OrderID** unique. Les colonnes incluent l'identifiant de la table (**TableID**), l'identifiant de l'employé ayant pris la commande (**EmployeeID**), l'heure de la commande (**OrderTime**), le statut de la commande (**Status**) et le montant total (**TotalAmount**). Cette table est fondamentale pour la gestion des services en salle et l'établissement des factures.

Détails de la commande de table

La table **orderdetails** est une table de liaison qui relie les commandes (**orders**) aux recettes (**recipes**). Chaque détail de commande est identifié par un **OrderDetailID** unique. Les colonnes incluent l'identifiant de la commande (**OrderID**), l'identifiant de la recette (**RecipeID**), la quantité (**Quantity**), et le sous-total (**Subtotal**). Ce tableau détaille les éléments spécifiques de chaque commande.

Recettes de table

La table **recipes** contient les informations sur les recettes proposées par le restaurant. Chaque recette est identifiée par un **RecipeID** unique. Les colonnes incluent le nom de la recette (**RecipeName**), le type de recette (**RecipeType**, comme "Entrée", "Main Course", ou "Dessert"), le temps de préparation et de cuisson (**PreparationTime**, **CookingTime**), ainsi qu'une colonne apparaît automatiquement le temps total (**TotalTime**). Cette table permet une gestion centralisée des recettes disponibles au menu.

Recette de table

La table **recipestock** est utilisée pour détailler les stocks et les étapes nécessaires à la préparation des recettes. Chaque entrée est identifiée par un **RecipeStockID** unique. Les colonnes incluent l'identifiant de la recette (**RecipeID**), la description de la préparation (**Preparation**), et des détails supplémentaires (**Details**). Ce tableau fournit des informations détaillées sur les exigences en matière de stock et les instructions de préparation.

Ce modèle conceptuel de données est organisé pour permettre une gestion optimale des opérations du restaurant, en dépendant des clients, des employés, des commandes et des stocks. Les relations entre les tables assurent une cohérence et une flexibilité dans la gestion des données, facilitant ainsi l'optimisation des processus et la prise de décisions.

PARTIE II : DESCRIPTION DES DESIGN PATTERN UTILISES

1- SINGLETON

Le pattern Singleton a pour rôle de garantir l'unicité d'une instance d'une classe tout en fournissant un accès global à cette dernière. Pour y parvenir, le constructeur de la classe est rendu privé, empêchant ainsi sa création depuis l'extérieur. Une méthode statique, généralement appelée **GetInstance**, est utilisée pour récupérer cette instance unique. Si elle existe déjà, la méthode la retourne ; dans le cas contraire, elle en crée une nouvelle.

Dans notre application, nous avons employé le Singleton pour la gestion de la connexion à la base de données. Cela assure qu'une seule connexion est maintenue, évitant ainsi les problèmes liés à des accès multiples, qui pourraient compromettre l'intégrité des données. Le Singleton a également été utilisé pour le module de journalisation. Cela permet de centraliser les accès aux fichiers de logs et d'éviter que des écritures concurrentes n'endommagent ces derniers.

Ce design pattern est particulièrement utile dans les systèmes nécessitant une gestion rigoureuse des ressources partagées ou critiques.

2- FACTORY

Le besoin de créer des objets différents en fonction de certains paramètres est fréquent dans les applications. Par exemple, une usine pourrait produire divers types de produits en fonction des spécifications qu'on lui fournit. Pour répondre à cette exigence, le design pattern **Factory** est utilisé, permettant de centraliser et de simplifier le processus de création d'objets en déléguant cette responsabilité à des classes spécialisées, appelées "usines".

L'objectif principal de ce pattern est de fournir une interface commune pour la création d'objets tout en laissant les sous-classes ou des méthodes spécifiques décider quel type exact d'objet doit être instancié. Ainsi, le code principal n'a pas besoin de connaître les détails des classes à instancier, ce qui améliore sa lisibilité, sa modularité, et facilite les modifications ou extensions futures.

Dans notre application, ce concept a été appliqué pour gérer les différents membres de l'équipe de la salle du restaurant, représentés par la classe Staff dans notre diagramme UML. Cette équipe inclut des rôles tels que Serveur, Maître d'hôtel, et autres. Grâce au design pattern **Factory**, nous avons transféré la responsabilité de créer chaque type de membre du personnel à des classes ou méthodes spécifiques. Cela permet de gérer aisément l'ajout de nouveaux rôles ou la modification des existants sans altérer le code principal.

En résumé, le **Factory** permet de simplifier et de structurer le processus de création d'objets complexes tout en maintenant un couplage faible entre les composants.

3- OBSERVATEUR

Certaines classes dans une application contiennent des données ou des états susceptibles de changer fréquemment. Lorsque ces modifications se produisent, il peut être nécessaire de notifier automatiquement d'autres classes dépendantes afin qu'elles puissent réagir en conséquence. Le design pattern Observateur répond précisément à ce besoin en établissant une relation entre un sujet (objet observé) et un ensemble d'observateurs.

Le principe de ce pattern repose sur un mécanisme d'abonnement : les observateurs s'enregistrent auprès du sujet pour recevoir des notifications à chaque mise à jour. Cela garantit une synchronisation automatique entre l'état du sujet et les actions des observateurs, tout en maintenant un faible couplage entre les différentes classes.

Dans notre application, ce modèle est utilisé pour synchroniser les changements effectués dans le modèle métier avec l'interface utilisateur. Par exemple, lorsqu'un état évolue dans le backend, la vue est immédiatement mise à jour pour refléter ces changements. Par ailleurs, ce pattern est exploité pour permettre aux clients de signaler leurs besoins à l'équipe en salle, en envoyant des notifications aux membres concernés.

Dans le diagramme de classes, l'implémentation du pattern Observateur repose sur des *Events*, qui agissent comme des types de délégués en C# .NET. Ces délégués facilitent la communication entre le sujet et ses observateurs, offrant un moyen efficace de gérer les notifications tout en maintenant la modularité et la maintenabilité du code.

4- STRATEGY

Le design pattern **Strategy** est utilisé pour permettre à une classe de changer dynamiquement son comportement ou celui d'une de ses méthodes en fonction d'une situation spécifique. Ce modèle repose sur l'encapsulation de différentes stratégies, chacune représentant une manière distincte de réaliser une opération, dans des classes séparées. Ainsi, la classe principale peut interagir avec une interface commune, ce qui facilite la sélection et la modification des stratégies à exécuter.

Dans notre application, ce pattern est employé pour gérer les modes de commande des clients. Chaque client peut adopter une méthode de commande différente selon ses préférences : certains choisissent de passer leur commande complète en une seule fois, tandis que d'autres préfèrent commander leurs plats progressivement, au fur et à mesure de leurs envies. Grâce au design pattern **Strategy**, ces différentes approches sont encapsulées dans des stratégies distinctes, permettant au système d'adapter le processus de commande de manière flexible.

Ce modèle améliore la modularité du code en isolant les différentes logiques de commande et simplifie l'ajout ou la modification d'une nouvelle stratégie sans impact sur la classe principale. Il garantit également une meilleure maintenabilité en réduisant les dépendances et en favorisant la réutilisation des stratégies dans d'autres contextes si nécessaire.

5- BRIDGE

Le design pattern **Bridge** vise à séparer une abstraction de son implémentation afin qu'elles puissent évoluer indépendamment. Plutôt que d'imposer une relation rigide entre l'interface et les différentes implémentations, ce modèle introduit un pont permettant de connecter ces deux aspects de manière flexible. L'interface agit alors comme un médiateur, garantissant une interaction cohérente tout en offrant la liberté de varier ou étendre les implémentations sans affecter la structure de l'application.

Dans notre cas, ce pattern est appliqué pour gérer les modes de déplacement des membres de l'équipe du restaurant. Chaque membre du personnel, qu'il s'agisse d'un serveur, d'un maître d'hôtel ou d'un commis, peut adopter une manière spécifique de se déplacer (par exemple, un déplacement rapide pour répondre à une commande urgente ou un déplacement discret pour ne pas perturber l'ambiance de la salle). Grâce à *Bridge*, l'interface définissant les déplacements est découplée de ses implémentations concrètes. Cela permet de facilement adapter ou introduire de nouveaux comportements de déplacement, indépendamment de la logique principale.

Ce design favorise une évolutivité remarquable, car les variations d'abstractions (les rôles du personnel) et d'implémentations (les modes de déplacement) peuvent être gérées séparément. Cela garantit aussi un code plus clair, limitant les doublons et renforçant la capacité à répondre à des besoins changeants, comme l'ajout d'un nouveau membre du staff ou d'une méthode de déplacement inédite.

CONCLUSION

En conclusion, l'architecture logicielle proposée dans ce rapport traduit une modélisation rigoureuse des processus opérationnels d'un restaurant. À travers les différents diagrammes présentés, chaque acteur et chaque composante du système a été soigneusement analysé pour assurer une représentation fidèle des interactions et des flux de données. Cette approche méthodique permet non seulement d'optimiser les tâches des employés, mais aussi d'améliorer l'expérience des clients en offrant un service fluide et structuré. Les modèles présentés constituent ainsi une base solide pour la mise en œuvre d'une solution logicielle capable de répondre aux exigences opérationnelles et d'évoluer en fonction des besoins futurs.