One Minute

APPLICATION POUR LA RESTAURATION

DOCUMENT POUR LA MOA

Auteurs

- Anthony Slimani
- Laurine Drodzinski
- Noura Mares
- Sami Barchid
- Zoé Canoen

Introduction

Ce document récapitule la conception de l'application "One Minute". Le but de cette application est de faire gagner du temps au restaurant. Nous nous sommes mis à la place de tous les acteurs du restaurant, mais davantage sur la partie service, cuisine et menu du restaurant.

Sommaire

1. Résumé du problème Page 3
2. Scénarios concrets Page 3
3. Diagrammes de cas d'utilisation Page 15
4. Hiérarchie des cas d'utilisation Page 20
5. Descriptions textuelles Page 21
6. Diagrammes de séquences système Page 46
7. Diagrammes de séquences détaillé Page 61
8. Diagrammes d'états Page 74
9. Diagrammes de classes Page 75
10. Description de l'architecture logicielle Page 80
11. Glossaire de l'ingénierie des besoins Page 82
12. Premières versions des maquettes Page 84
13. Glossaire métier Page 86
14. Annexes Page 87

Résumé du problème

Le but de notre projet est d'informatiser le processus de gestion des commandes d'un restaurant. L'application doit permettre de prendre en charge une commande à partir de la demande du client jusqu'au règlement de l'addition en passant par le traitement en cuisine.

Scénarios concrets

Liste des scénarios

Service

Prendre une commande

Un serveur prend la commande d'un client.

Ajouts de préparations dans une commande

Un serveur ajoute une ou des préparations dans une commande en cours.

Suivre une commande

Un membre du restaurant peut suivre l'état d'avancement des commandes.

Servir une commande

Un serveur amène une partie de commande au client.

Paiement

Le client a terminé son repas et veut payer l'addition.

Le client prend sa commande

Un client peut prendre lui-même sa commande via l'application (optionnel).

Cuisine

Cuisiner une préparation

Un préparateur cuisine une préparation attendue par un client.

Préparation indisponible

Une préparation ne peut pas être réalisée.

Menu

Créer un menu

Le patron crée un menu.

Administrer un menu

Le patron modifie un menu existant.

Description des scénarios

Acteurs :

- ***Bob*** est un client
- ***Alice*** est une serveuse
- ***Roger*** est un préparateur
- ***Albert*** est le patron

Prendre une commande

Description :

Alice propose à ***Bob*** de prendre sa première commande.

Scénario :

- ***Alice*** propose le menu à ***Bob***.
- ***Bob*** choisit ce qu'il souhaite commander.
- ***Alice***, à partir de sa tablette, transmet la commande et le numéro de table aux préparateurs.
- Les préparateurs sont au courant de la commande.

Ajouts de préparations dans une commande

Description :

Alice peut ajouter des préparations dans une commande déjà en cours si ***Bob*** le désire.

Scénario :

- ***Bob*** veut ajouter une préparation à sa commande en cours.
- ***Bob*** appelle ***Alice*** pour changer sa commande.
    - ***Alice*** n'est pas disponible (alternatif).
- ***Alice*** prend sa tablette.
- ***Alice*** ajoute la nouvelle préparation à la commande.
    - La préparation n'est plus disponible (alternatif).
- L'équipe de cuisine est avertie de l'ajout.
    - Problème de notification si problème de connexion à internet (exception).

Suivre une commande

Description :

Le personnel du restaurant peut suivre l'état d'avancement d'une commande en cours ou passée.

Scénario :

- ***Alice*** veut savoir si la commande de spaghettis de ***Bob*** est prête.
- ***Alice*** sélectionne la commande de ***Bob*** pour voir son état.
- ***Alice*** voit que les spaghettis de ***Bob*** sont prêtes.
- ***Alice*** voit que son mojito fraise n'est pas prêt.
- ***Alice*** va chercher les spaghettis en cuisine.
- ***Alice*** amène les spaghettis à ***Bob***.
- ***Alice*** indique que les spaghettis ont été servies.

Servir une commande

Description :

Lorsque qu'une ***partie de la commande*** est prête, il faut la servir à ***Bob***.

Scénario :

- ***Alice*** est mise au courant qu'une ***partie de la commande*** est prête.
    - Problème de notification si problème de connexion à internet (exception).
- ***Alice*** indique sur l'application qu'elle va la chercher.
- ***Alice*** récupère la ***partie de la commande***.
- ***Alice*** la ramène à ***Bob***.

Paiement

Description :

Bob a fini de manger et souhaite quitter le restaurant mais avant il doit régler ***l'addition*** de sa commande.

Scénario :

- ***Bob*** demande ***l'addition*** à ***Alice.***
  - ***Alice*** n'est pas disponible (alternatif).
- ***Alice*** recherche ***l'addition*** à partir du numéro de table de ***Bob*** par l'intermédiaire de l'application.
  - ***Alice*** n'a pas accès au service de commande sur l'application (exception).
- L'application affiche ***l'addition.***
- ***Alice*** informe le montant de ***l'addition*** à ***Bob.***
- ***Bob*** paie ***l'addition*** par le moyen de paiement choisi.
- ***Alice*** indique à l'application que ***l'addition*** a bien été réglée.
- L'application archive la commande de ***Bob.***

Questions :

- Faut-il générer ***l'addition*** de manière physique à ***Bob*** ?
  - ***L'addition*** a besoin d'être générée de manière physique notamment pour les notes de frais.

Le client prend sa commande

La fonctionnalité étant optionnelle, elle sera décrite et travaillée ultérieurement. Nous la prévoyons pour la V3. Nous ferons néanmoins le nécessaire pour pouvoir garder l'application la plus généraliste possible pour pouvoir incorporer cette fonctionnalité plus tard.

Cuisiner une préparation

Description :

Roger désire faire une préparation, et doit informer l'application sur quelle(s) préparation(s) il est en train de cuisiner.

Scénario :

- ***Roger*** veut cuisiner une (ou plusieurs) préparation(s).
- ***Roger*** choisi une (ou plusieurs) préparation(s) qu'il s'apprête à cuisiner.
- ***Roger*** valide et peut commencer à cuisiner.
- Une fois que ***Roger*** a terminé sa préparation, il l'indique sur l'application.
- Ensuite, SOIT
    - sa préparation permet de terminer une ***partie de la commande***, dans ce cas une notification est envoyée aux serveurs pour venir chercher la préparation.
    - il manque une préparation pour la commande, la commande est alors en attente de la préparation manquante. Et attend qu'un préparateur cuisine la préparation manquante.

Préparation indisponible

Description :

Roger s'aperçoit que la réalisation d'une préparation ne peut être réalisée et doit donc rendre ***indisponible*** cette préparation.

Scénario :

- ***Roger*** indique que la préparation est ***indisponible*** sur l'application
    - Des commandes comprenant cette préparation sont en cours
        - L'application annule les commandes comportant cette préparation
        - L'application notifie les serveurs des commandes annulées
- L'application bloque la possibilité de sélectionner cette préparation lors d'une commande

[En attente de ***réapprovisionnement*** des ingrédients]

- ***Roger*** indique que la préparation est de nouveau disponible
- L'application débloque la possibilité de sélectionner cette préparation lors d'une commande

---

Créer un menu

Description :

Albert crée ou modifie la carte du restaurant, avant l'ouverture ou après la fermeture du restaurant.

Scénario :

- ***Albert*** commence à créer ou modifier la carte dans l'application.
- ***Albert*** crée ou modifie divers éléments.
- ***Albert*** termine la création ou la modification.
- La carte s'actualise sur les diverses tablettes.
    - Problème d'actualisation si les tablettes ne sont pas connectées à internet.

Administrer un menu

Description :

Albert ajoute, modifie et supprime les préparations disponibles dans l'application pour que celles-ci soient à jour par rapport aux menus.

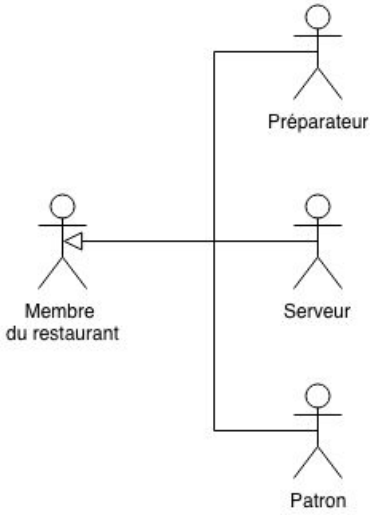
Scénario :

- ***Albert*** désire ajouter une préparation dans l'application.
- ***Albert*** indique à l'application qu'il veut ajouter cette préparation.
- L'application l'invite à décrire la préparation.
- ***Albert*** décrit la préparation.
- ***Albert*** veut supprimer une autre préparation.
- L'application propose les préparations déjà créées.
- ***Albert*** choisit la préparation à supprimer.
- La préparation choisie est supprimée.
- ***Albert*** veut modifier une dernière préparation.
- L'application propose les préparations existantes.
- ***Albert*** choisit la préparation à modifier.
- ***Albert*** modifie la préparation.
- ***Albert*** enregistre les modifications.
- La préparation est modifiée.

Diagrammes de cas d'utilisation

Acteurs

Notre application se voulant généraliste, il est important de noter que le terme préparateur inclut le rôle de cuisinier, de barman, de glacier etc. .



Nous avons choisi ces acteurs en fonction du sujet. Nous avons bien conscience que dans certains restaurants, il n'y a pas obligatoirement un barman ou un glacier, mais plutôt des serveurs en tant que préparateurs. De plus dans les restaurants, il y a souvent des managers, comme dans la restauration rapide par exemple.

Membre du restaurant

- Cet acteur est la généralisation des acteurs suivants :

Patron

- Gérant du restaurant.
- Possède un compte particulier qui lui permet de :
    - créer de nouveaux menus
    - modifier des menus existants
    - superviser l'état global du restaurant (avancement des commandes etc.)
- Doit être connecté à internet.

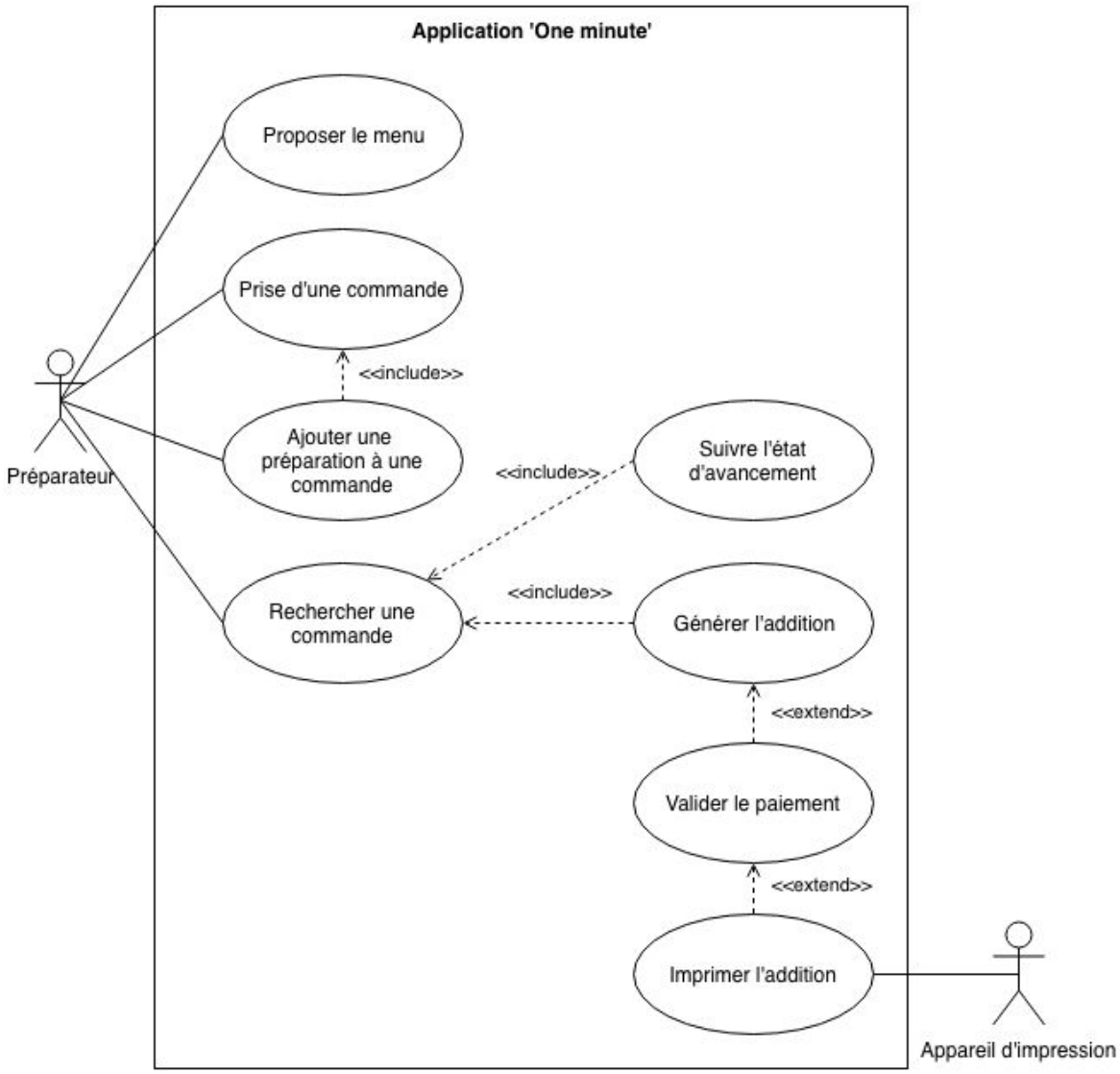
Serveur

- Est en contact avec les clients.
- Possède une tablette lui permettant de :
    - proposer le menu aux clients
    - prendre des commandes
    - suivre l'avancement des commandes
    - générer l'addition et valider le paiement
- Doit être connecté à internet.

Préparateur

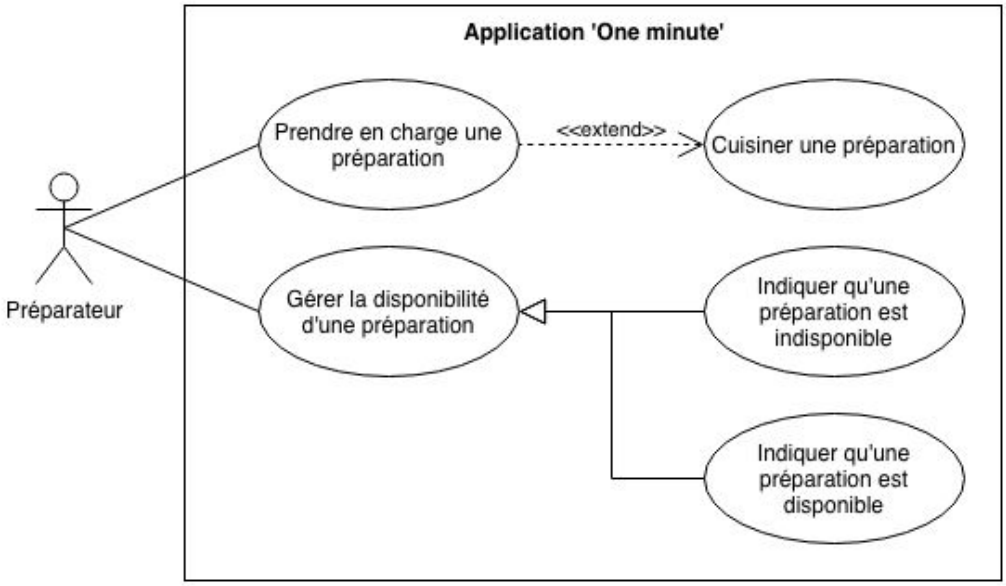
- Peut-être un barman, un glacier, ou simplement un cuisinier etc.
- Ne possède pas une tablette à lui seul (une ou deux tablettes pour l'ensemble des cuisiniers, une pour les barmans, glaciers etc.).
- Celle-ci lui permet de prendre en charge une préparation et également de gérer la disponibilité des plats.
- Doit être connecté à internet.

Service



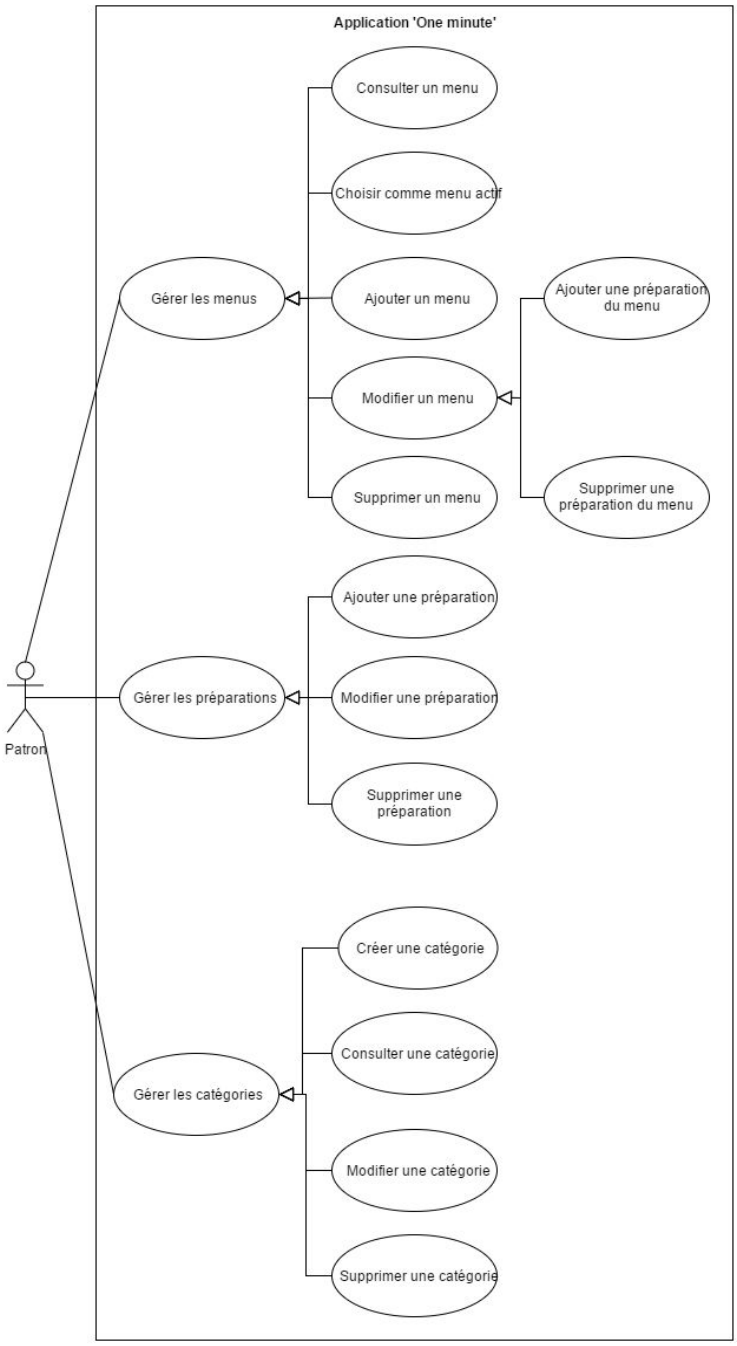
Le serveur a la possibilité de proposer le menu au client. Il pourra notamment prendre la commande de celui-ci. Pour cela, il devra saisir le numéro de table ainsi qu'ajouter une ou plusieurs préparations à la commande.
Lorsqu'une préparation est ajoutée dans une commande, les préparateurs sont directement notifiés. Les serveurs ont également la possibilité de servir une partie de la commande dès lors qu'ils ont été notifiés par les préparateurs.
Il peut également suivre l'état d'avancement de la commande. Dès que celle-ci est finie, le serveur pourra générer l'addition puis indiquer si le règlement a bien été réalisé.

Cuisine



Le préparateur a la possibilité d'indiquer sur l'application la prise en charge d'une préparation. Dès qu'il a fini, il doit indiquer l'accomplissement de cette dernière. Cela a comme conséquence de notifier tous les serveurs. Il a également la possibilité d'indiquer qu'une préparation est disponible ou indisponible.

Menu



Le patron a la possibilité de gérer le menu et les préparations, c'est-à-dire d'ajouter, de modifier ou de supprimer menus et préparations.

Hiérarchie des cas d'utilisation

Service

MOA	Nom CU	MOE
1	Prendre une commande	3
1	Servir une commande	1
1	Paiement	4
2	Suivre une commande	3
3	Imprimer l'addition	1
3	Ajouter une préparation à la commande	5
3	Proposer les plats disponibles du menu	3
5	Préparation Indisponible	2

Cuisine

MOA	Nom CU	MOE
1	Prendre en charge une préparation	1
1	Cuisinier une préparation	1
5	Préparation Indisponible	2

Menu

MOA	Nom CU	MOE
3	Créer une préparation	4
3	Supprimer une préparation	4
3	Modifier une préparation	4
3	Ajouter préparation au menu	5
3	Retirer une préparation du menu	5

Descriptions textuelles

Général

Conditions et exceptions s'appliquant à chacunes des descriptions textuelles.

Pré-conditions :

- Le serveur est disponible à tout moment
- Le membre du restaurant est authentifié au système par son rôle / qualification

Scénarios d'exceptions :

- L'appareil technique est hors-service
- Communication entre l'appareil et le serveur impossible
- Erreur de communication entre l'application et le serveur

Service

Proposer le menu

Nom du cas : Proposer le menu
But : Proposer le menu au client
Acteur principal : Le serveur
Date de création : 10/11/2018
Nom du responsable de création : Zoé Canoen
Version : 1.0

Pré-conditions :

- Le client est assis à une table

Déclenchement :

- Le client souhaite voir le menu du restaurant.

Scénario nominal :

1. Début du scénario
2. Le serveur est à la table du client.
3. Le serveur sélectionne un menu.
4. Le système affiche le menu sélectionné.
5. Le serveur montre le support au client.
6. Fin du scénario

Continuité du scénario :

- Le client peut prendre une commande ou ajouter une préparation à sa commande.

Prise d'une commande

Nom du cas : Prise d'une commande
But : Le serveur prend une commande à une table pour un client
Acteur principal : Serveur
Date de création : 06/11/2018
Nom du responsable de création : Zoé CANOEN
Dernière date de mise à jour : 09/11/2018
Nom du responsable de la dernière modification : Anthony SLIMANI
Version : 2.2

Pré-conditions :

- Le client est assis à une table.

Déclenchement :

- Le client appelle le serveur pour prendre sa commande
- Le serveur décide d'aller prendre la commande d'une table

Scénario nominal :

1. Début du scénario
2. Le serveur propose le menu au client (Voir Proposer menu)
3. Le serveur crée une nouvelle commande avec le numéro de table
4. Le serveur demande au client ce qu'il a choisi
5. Le client énonce les diverses préparations qu'il souhaite commander
6. Le serveur sélectionne les préparations parmis les préparations disponibles
7. Le serveur valide la commande
8. Fin du scénario

Post-condition :

- La commande est validée et est prête à être cuisinée

Scénarios alternatifs :

- 6.a [Une des préparations énoncées n'est plus disponible]
    - 6.a.1 Le serveur annonce que la préparation n'est plus disponible
    - 6.a.2 Le serveur invite le client à choisir une autre préparation
    - Reprise à l'étape 5

Ajouter une préparation à une commande

Nom du cas : Ajouter une préparation à la commande
But : Ajouter une préparation à une commande existante
Acteur principal : Serveur
Date de création : 06/11/2018
Nom du responsable de création : Zoé CANOEN
Dernière date de mise à jour : 10/11/2018
Nom du responsable de la dernière modification : Laurine DRODZINSKI
Version : 2.0

Pré-conditions :

- Le client a déjà commandé une fois auparavant.

Déclenchement :

- Le cas commence lorsque le client souhaite ajouter des préparations à sa commande.

Scénario nominal :

1. Début du scénario
2. Le client fait appel à un serveur afin de changer sa commande.
3. Le serveur arrive à la table du client.
4. Le serveur prend sa tablette.
5. Le serveur sélectionne la liste des commandes en cours.
6. Le système affiche la liste des commandes en cours.
7. Le serveur sélectionne la commande du client.
8. Le système affiche le résumé de la commande sélectionnée.
9. Le serveur demande au client ce qu'il souhaite commander.
10. Le serveur ajoute la ou les nouvelles préparations à la commande.
11. Le serveur confirme l'ajout de préparations.
12. Le système notifie l'équipe de cuisine que la commande a été modifiée.
13. Fin du scénario

Post-condition :

- L'équipe de cuisine est notifiée de la modification de la commande en question.

Scénarios alternatifs :

- 3a. [Aucun serveur n'est disponible]
    - 3a.1 Le client attend qu'un serveur soit de nouveau disponible.
    - 3a.2 Renvoie à 2.

- 10a. [Une ou plusieurs préparations demandées ne sont plus disponibles]
    - 10a.1 Le système affiche que la ou les préparations ne sont plus disponibles.
    - 10a.2 Le serveur ne peut pas les sélectionner.
    - 10a.3 Renvoie à 9.

Scénario d'exception :

- 6a. [Le système n'est pas connecté à internet]
    - 6a.1 Le système prévient le serveur qu'il ne parvient pas à se connecter à internet.
    - 6a.2 Le système met fin à la session.

Suivre une commande

Nom du cas : Suivre une commande
But : Le serveur souhaite suivre l'état d'avancement d'une commande
Acteur principal : Serveur
Date de création : 11/09/2018
Nom du responsable de création : Laurine DRODZINSKI
Dernière date de mise à jour : 06/11/2018
Nom du responsable de la dernière modification : Anthony SLIMANI
Version : 2.2

Déclenchement :

- Le serveur souhaite vérifier l'état d'avancement d'une commande

Scénario nominal :

1. Début du scénario
2. Le serveur indique qu'il souhaite afficher la liste des commandes en cours
3. Le système affiche la liste des commandes en cours
4. Le serveur sélectionne la commande du client
5. Le système affiche le résumé et l'état d'avancement de chaque partie de la commande sélectionnée
6. Le serveur consulte la commande du client
7. Fin du scénario

Post-condition :

- Le serveur a consulté l'état d'avancement d'une commande.

Continuité du scénario :

- Lorsqu'une partie de la commande est prête, le scénario Servir une commande peut se réaliser.
- Lorsque toutes les parties de la commande ont été servie, le scénario Paiement peut se réaliser.

Scénario d'exception :

- 3.a. [Aucune commande n'est en cours]
    - 3.a.1 Le système affiche qu'aucune commande est en cours
    - 3.a.2 Fin du scénario

Servir une partie de commande

Nom du cas : Servir une partie de commande
But : Le serveur sert une partie de la commande d'un client qui a été préparée par un préparateur
Acteur principal : Serveur
Date de création : 11/09/2018
Nom du responsable de création : Sami BARCHID
Dernière date de mise à jour : 09/11/2018
Nom du responsable de la dernière modification : Zoé CANOEN
Version : 2.0

Pré-conditions :

- La partie de la commande à servir est prête.
- Un cuisinier a terminé une partie de commande, voir scénario "Cuisiner une préparation"

Déclenchement :

- Le serveur consulte les parties de commandes prêtes à servir.

Scénario nominal :

1. Début du scénario
2. Le serveur consulte les parties de commandes prêtes à servir.
3. Le système affiche les parties de commandes prêtes à servir.
4. Le serveur sélectionne une partie de commande.
5. Le système supprime la partie de commande dans la liste des commandes à servir.
6. Le serveur va chercher la partie de commande à servir.
7. Le serveur sert la partie de commande au client.
8. Fin du scénario

Post-condition :

- La partie de commande est servie au client.

Remarque :

- Le serveur a reçu une notification de la cuisine avant le scénario. Il n'est pas obligé de servir directement la commande et peut faire d'autres choses entre deux. C'est pour cela que nous ne mettons pas cela en post condition.

Scénarios alternatifs :

- 2.a. [Aucune partie de commande prête à servir]
    - Rien ne se passe.

Scénario d'exception :

- 4.a. [La partie de commande sélectionnée est annulée/terminée]
    - 4.a.1. Le système affiche un message d'erreur au serveur.
    - 4.a.2. Le système retourne à l'état "afficher les parties de commandes".
- *.a. [Panne de réseau]
    - Le système indique que le réseau a été coupé.
    - Le système retourne à son état d'avant le déclenchement.

Paiement

Nom du cas : Paiement d'une commande
But : Le client souhaite régler l'addition de sa commande
Acteur principal : Serveur
Date de création : 19/10/2018
Nom du responsable de création : Anthony SLIMANI
Dernière date de mise à jour : 06/11/2018
Nom du responsable de la dernière modification : Anthony SLIMANI
Version : 2.2

Pré-conditions :

- Le serveur consulte le suivi de la commande
- L'état de réglementation de la commande est à "Non réglé"

Déclenchement :

- Le client appelle le serveur pour demander à régler sa commande

Scénario nominal :

1. Début du scénario
2. Le serveur choisit de générer l'addition
3. Le système affiche l'addition de la commande du client
4. Le serveur saisit le montant de la commande sur le système de paiement
5. Le client paie sa commande
6. Le serveur indique au système que le règlement a bien été effectué
7. Le système retire la commande de la liste des commandes en cours
8. Fin du scénario

Post-condition :

- L'état de réglementation de la commande est à "Réglé"

Continuité du scénario :

- Si le client le souhaite, il peut demander au serveur le détail de sa commande sous forme physique. Le scénario Imprimer l'addition d'une commande peut se réaliser.

Scénarios alternatifs :

- 5.a [Le paiement du client est refusé]

Scénario d'exception :

- 1.a [Le client n'est plus présent au restaurant]
    - 1.a.1 Le serveur annule la commande

- 1.a.2 Le système change l'état de la commande à "Fini"
- 1.a.3 Fin du scénario

Imprimer l'addition d'une commande

Nom du cas : Imprimer l'addition d'une commande
But : Le client souhaite avoir le détails de sa commande sous forme physique
Acteur principal : Serveur
Date de création : 19/10/2018
Nom du responsable de création : Anthony SLIMANI
Dernière date de mise à jour : 06/11/2018
Nom du responsable de la dernière modification : Anthony SLIMANI
Version : 2.2

Pré-conditions :

- Le serveur consulte la commande qui doit être imprimée sur le support technique
- L'état de réglementation de la commande est à "Réglé"

Déclenchement :

- Le client demande au serveur l'impression du détail de sa commande

Scénario nominal :

1. Début du scénario
2. Le serveur indique au système que le détail d'une commande doit être imprimé
3. Le système envoie le détail de la commande à l'appareil d'impression
4. L'appareil d'impression effectue l'impression
5. Le système affiche au serveur que l'impression est terminée
6. Le serveur apporte l'impression du détail de la commande au client
7. Fin du scénario

Post-condition :

- Le client a reçu le détail de sa commande sous forme physique.

Scénario d'exception :

- 3.a [L'appareil d'impression n'a pas la possibilité d'imprimer]
    - 3.a.1 Le système affiche un message d'erreur avec un message indiquant la cause
    - 3.a.2 Fin du scénario

Préparation indisponible

Nom du cas : Préparation indisponible
But : Lorsqu'une préparation devient indisponible il faut prévenir le client afin qu'il réadapte sa commande.
Acteur principal : Serveur
Date de création : 06/11/2018
Nom du responsable de création : Zoé CANOEN
Dernière date de mise à jour : 06/11/2018
Nom du responsable de la dernière modification : Laurine DRODZINSKI
Version : 2.0

Pré-conditions :

- Le client a passé une commande.

Déclenchement :

- Une préparation de la commande du client n'est plus disponible
- La préparation a été indiquée comme indisponible par la cuisine

Scénario nominal :

1. Début du scénario
2. Le système change l'état de la ligne de commande en "annulée" sur les lignes de commandes contenant des préparations indisponibles.
3. Le serveur va à la table où le plat est indisponible.
4. Le client peut choisir à ce moment un nouveau plat disponible.
5. Fin du scénario

Scénarios alternatifs :

- La préparation est de nouveau disponible car il y a eu un réapprovisionnement, dans ce cas, on ne fait rien.

Cuisine

Prendre en charge une préparation

Nom du cas : Prendre en charge une préparation
But : Ce cas montre les étapes permettant au préparateur de s'occuper d'une préparation
Acteur principal : Préparateur
Date de création : 03/11/2018
Nom du responsable de création : Zoé CANOEN
Dernière date de mise à jour : 08/11/2018
Nom du responsable de la dernière modification : Anthony SLIMANI
Version : 2.1

Pré-conditions :

- Des commandes sont en cours

Déclenchement :

- Le préparateur souhaite s'occuper d'une ou plusieurs préparation.s.

Scénario nominal :

1. Début du scénario
2. Le préparateur sélectionne la liste des préparations non prises en charge
3. Le système affiche la liste des préparations à réaliser
4. Le préparateur choisit une ou plusieurs préparations qu'il s'apprête à préparer
5. Le préparateur sélectionne son nom dans la liste des préparateurs
6. Le préparateur valide son choix
7. Le système réserve la ou les préparation.s choisie.s afin qu'aucun autre préparateur ne puisse les sélectionner
8. Fin du scénario

Post-condition :

- La ou les préparation.s choisie.s ne peuvent plus être pris en charge par d'autres préparateurs

Continuité du scénario :

- Le scénario Cuisiner une préparation peut être réalisé.

Scénario d'exception :

- 3.a [Aucune préparation dans la liste des préparations à réaliser]

    - 3.a.1 Le système affiche un message indiquant qu'aucune préparation n'est à réaliser
    - 3.a.2 Fin du scénario
- 7.a [La préparation a été déjà été prise en charge]
    - 7.a.1 Le système affiche un message indiquant que la préparation a été déjà été prise en charge
    - 7.a.2 Fin du scénario

Cuisiner une préparation

Nom du cas : Cuisiner une préparation
But : Ce cas montre les étapes permettant au préparateur de s'occuper d'une préparation
Acteur principal : Préparateur
Date de création : 11/09/2018
Nom du responsable de création : Laurine DRODZINSKI
Dernière date de mise à jour : 08/11/2018
Nom du responsable de la dernière modification : Anthony SLIMANI
Version : 2.3

Pré-conditions :

- Le préparateur a prit en charge une préparation

Déclenchement :

- Le préparateur souhaite s'occuper d'une ou plusieurs préparations

Scénario nominal :

1. Début du scénario
2. Le préparateur commence à préparer
3. Le préparateur termine la préparation
4. Le préparateur indique au système qu'il a terminé les préparations
5. Le préparateur indique qu'une partie de commande est terminée
6. Le système modifie l'état de la partie de commande terminée
7. Le système notifie les serveurs
8. Fin du scénario

Continuité du scénario :

- Le scénario Servir une commande peut être réalisé.

Post-condition :

- Les serveurs sont notifiés afin qu'ils puissent servir une partie de commande.

Scénarios alternatifs :

- 2.a [La préparation ne peut être réalisée]
    - Réalisation du scénario Préparation indisponible

Préparation indisponible

Nom du cas : *Préparation* indisponible
But : Ce cas montre les étapes permettant au *préparateur* de rendre *indisponible* une *préparation*
Acteur principal : Préparateur
Date de création : 11/09/2018
Nom du responsable de création : Laurine DRODZINSKI
Dernière date de mise à jour : 08/11/2018
Nom du responsable de la dernière modification : Anthony SLIMANI
Version : 2.3

Pré-conditions :

- Il n'y a plus assez d'ingrédients pour préparer un type de *préparation*.

Déclenchement :

- Le cas commence lorsque le *préparateur* se rend compte qu'une *préparation* ne peut pas être cuisinée.

Scénario nominal :

1. Début du scénario
2. Le *préparateur* sélectionne la *préparation* en question pour la rendre *indisponible*
3. Le système notifie les serveurs qu'une *préparation* ne peut pas être réalisée
4. Le système bloque la possibilité de sélectionner cette *préparation* lors de la prise d'une commande
5. Fin du scénario

Post-condition :

- La *préparation* reste indisponible tant qu'il n'y a pas eu réapprovisionnement.

Scénarios alternatifs :

- 2.a [Réapprovisionnement des ingrédients]
    - 2.a.1 Le *préparateur* indique que la *préparation* est de nouveau disponible.
    - 2.a.2 Le système débloque la possibilité de sélectionner cette *préparation* lors de la prise d'une commande.
    - 2.a.3 Fin du scénario.

Menu

Créer une préparation

Nom du cas : Créer une préparation
But : Le patron ajoute une préparation
Acteur principal : Patron
Date de création : 19/10/2018
Nom du responsable de création : Sami BARCHID
Dernière date de mise à jour : 05/11/2018
Nom du responsable de la dernière modification : Sami BARCHID
Version : 1.0

Déclenchement :

- Le patron indique qu'il veut créer une préparation.

Scénario nominal :

1. Début du scénario
2. Le patron indique qu'il veut créer une préparation.
3. Le système demande au patron d'entrer les données de la préparation.
4. Le patron entre les données de la préparation.
5. Le système valide les données de la préparation.
6. Le système enregistre la nouvelle préparation.
7. Le système indique au patron que la préparation est créée.
8. Fin du scénario.

Post-conditions :

- La nouvelle préparation est créée dans le système.
- La nouvelle préparation peut être créée dans des menus.

Scénarios alternatifs :

- 5.a. [Les données ne sont pas valides / La préparation entrée existe déjà]
    - 5.a.1. Le système affiche un message d'erreur.
    - 5.a.2. Retour à l'étape 3.

Scénarios d'exception :

- *.a. [Panne de réseau]
    - *.a.1 Le système indique que le réseau a été coupé.
    - *.a.2 Le système retourne à son état d'avant le déclenchement.
    - *.a.3 Fin du scénario.

Ajouter une préparation au menu

Nom du cas : Ajouter une préparation au menu
But : Le patron ajoute une préparation à un menu qu'il a choisi
Acteur principal : Patron
Date de création : 19/10/2018
Nom du responsable de création : Sami BARCHID
Dernière date de mise à jour : 05/11/2018
Nom du responsable de la dernière modification : Sami BARCHID
Version : 1.0

Pré-conditions :

- Le patron a choisi le menu sur lequel il veut ajouter une préparation.

Déclenchement :

- Le patron indique qu'il veut ajouter une préparation au menu choisi.

Scénario nominal :

1. Début du scénario
2. Le patron indique qu'il veut ajouter une préparation au menu choisi.
3. Le système affiche les préparations qui peuvent être ajoutées au menu.
4. Le patron choisit la préparation à ajouter.
5. Le système valide la préparation choisie.
6. Le système enregistre la préparation dans le menu.
7. Le système confirme l'ajout au patron.
8. Fin du scénario

Post-conditions :

- La préparation choisie est ajoutée au menu.

Scénarios alternatifs :

- 4.a. [La préparation à ajouter n'existe pas / existe déjà dans le menu]
    - 4.a.1. La patron indique qu'il veut
    - 4.a.2. Exécuter le CU : "Ajouter une préparation"
    - 4.a.3. Aller à l'étape 5.
- 5.a. [La préparation choisie n'est pas valide]
    - 5.a.1. Le système affiche un message d'erreur.
    - 5.a.2. Retour à l'étape 3.

Scénarios d'exception :

- 4.b. [Aucune préparation n'existe]

    - 4.b.1. Le système affiche un message d'erreur.
    - 4.b.2. Le système revient à son état d'avant le déclenchement.
    - 4.b.3. Fin du scénario.
- *.a. [Panne de réseau]
    - *.a.1 Le système indique que le réseau a été coupé.
    - *.a.2 Le système retourne à son état d'avant le déclenchement.
    - *.a.3 Fin du scénario.

Modifier une préparation

Nom du cas : Modifier une préparation
But : La patron modifie une préparation existante
Acteur principal : Patron
Date de création : 19/10/2018
Nom du responsable de création : Sami BARCHID
Dernière date de mise à jour : 05/11/2018
Nom du responsable de la dernière modification : Sami BARCHID
Version : 1.0

Déclenchement :

- Le patron indique qu'il veut modifier une préparation.

Scénario nominal :

1. Début du scénario
2. Le patron indique qu'il veut modifier une préparation.
3. Le système affiche les préparations existantes.
4. Le patron choisit la préparation à modifier.
5. Le système valide la préparation à modifier.
6. Le système demande au patron d'entrer les modifications.
7. Le patron entre les données de modification de la préparation.
8. Le système valide la préparation modifiée.
9. Le système modifie la préparation.
10. Le système indique au patron que la préparation est modifiée.
11. Fin du scénario.

Post-conditions :

- La préparation est modifiée.

Scénarios alternatifs :

- 5.a. [La préparation choisie n'est pas valide/n'existe pas]
    - 5.a.1. Le système affiche un message d'erreur.
    - 5.a.2. Retour à l'étape 3.
- 8.a. [Données invalides]
    - 8.a.1. Le système affiche un message d'erreur.
    - 8.a.2. Retour à l'étape 6.

Scénarios d'exception :

- 3.a. [Il n'y a aucune préparation existante]
    - 3.a.1. Le système affiche un message d'erreur.
    - 3.a.2. Le système revient à son état d'avant le déclenchement.

    - 3.a.3. Fin du scénario.
- *.a. [Panne de réseau]
    - *.a.1 Le système indique que le réseau a été coupé.
    - *.a.2 Le système retourne à son état d'avant le déclenchement.
    - *.a.3 Fin du scénario.

Retirer une préparation au menu

Nom du cas : Retirer une préparation au menu
But : Le patron retire une préparation à un menu qu'il a choisi
Acteur principal : Patron
Date de création : 19/10/2018
Nom du responsable de création : Sami BARCHID
Dernière date de mise à jour : 05/11/2018
Nom du responsable de la dernière modification : Sami BARCHID
Version : 1.0

Pré-conditions :

- Le patron consulte en détails le menu sur lequel il veut retirer une préparation.

Déclenchement :

- Le patron indique qu'il veut retirer une préparation au menu.

Scénario nominal :

1. Début du scénario
2. Le patron indique qu'il veut retirer une préparation au menu.
3. Le système affiche les préparations du menu.
4. Le patron choisit la préparation à retirer.
5. Le système demande la confirmation au patron.
6. Le patron confirme le choix de la préparation à retirer.
7. Le système valide la préparation choisie.
8. Le système supprime la préparation du menu.
9. Le système enregistre la suppression par le patron.
10. Fin du scénario.

Post-condition :

- La préparation choisie est retirée du menu.

Scénarios alternatifs :

- 6.a. [Le patron infirme la préparation à retirer]
    - 6.a.1. Le système affiche un message d'erreur.
    - 6.a.2. Le système revient à son état d'avant le déclenchement.
    - 6.a.3. Retour à l'étape 3.
- 7.a. [La préparation choisie n'est pas valide / est déjà retirée]
    - 7.a.1. Le système affiche un message d'erreur.
    - 7.a.2. Le système revient à son état d'avant le déclenchement.

    - 7.a.3. Retour à l'étape 3.

Scénarios d'exception :

- 3.b. [Aucune préparation n'existe dans le menu]
    - 3.b.1. Le système affiche un message d'erreur.
    - 3.b.2. Le système revient à son état d'avant le déclenchement.
    - 3.b.3. Fin du scénario.
- *.a. [Panne de réseau]
    - *.a.1 Le système indique que le réseau a été coupé.
    - *.a.2 Le système retourne à son état d'avant le déclenchement.
    - *.a.3 Fin du scénario.

Supprimer une préparation

Nom du cas : Supprimer une préparation
But : Le patron supprime une préparation existante
Acteur principal : Patron
Date de création : 19/10/2018
Nom du responsable de création : Sami BARCHID
Dernière date de mise à jour : 05/11/2018
Nom du responsable de la dernière modification : Sami BARCHID
Version : 1.0

Déclenchement :

- Le patron indique qu'il veut supprimer une préparation.

Scénario nominal :

1. Début du scénario
2. Le patron indique qu'il veut supprimer une préparation.
3. Le système affiche les préparations existantes.
4. Le patron choisit la préparation à supprimer.
5. Le système demande la confirmation du choix.
6. Le patron confirme le choix de la préparation à supprimer.
7. Le système valide la préparation à supprimer.
8. Le système supprime la préparation.
9. Le système indique au patron que la préparation est supprimée.
10. Fin du scénario.

Post-conditions :

- La préparation est retirée du système

Scénarios alternatifs :

- 6.a. [Le patron infirme la suppression de la préparation choisie]
    - 6.a.1. Le système affiche un message d'erreur.
    - 6.a.2. Retour à l'étape 3.
- 7.a. [La préparation choisie n'est pas valide/est déjà supprimée]
    - 7.a.1. Le système affiche un message d'erreur.
    - 7.a.2. Retour à l'étape 3.
- 7.b. [La préparation choisie est utilisée dans un/des menu(s)]
    - 7.b.1. Le système indique au patron que la préparation est utilisée dans un/des menu(s).
    - 7.b.2. Le système affiche les menus où la préparation est utilisée.
    - 7.b.3. Le patron confirme la suppression de la préparation.

    - 7.b.4. Aller à l'étape 7.

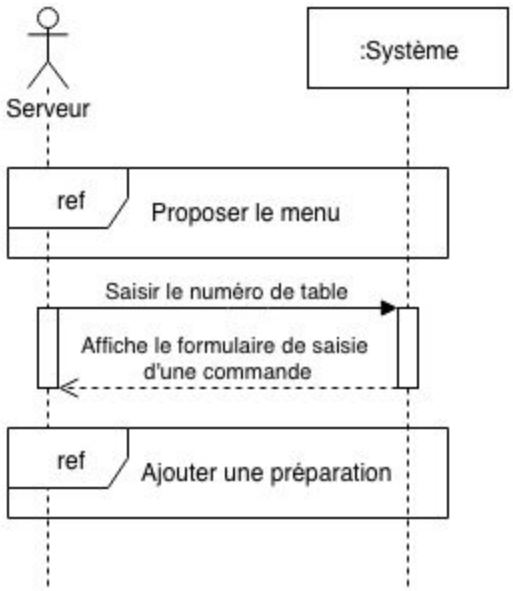
Scénarios d'exception :

- 7.b.3.a. [Le patron refuse la suppression de la préparation]
    - 7.b.3.a.1. Le système annule la suppression de la préparation.
    - 7.b.3.a.2. Le système retourne à son état d'avant le déclenchement.
    - 7.b.3.a.3. Fin du scénario.
- *.a. [Panne de réseau]
    - *.a.1 Le système indique que le réseau a été coupé.
    - *.a.2 Le système retourne à son état d'avant le déclenchement.
    - *.a.3 Fin du scénario.

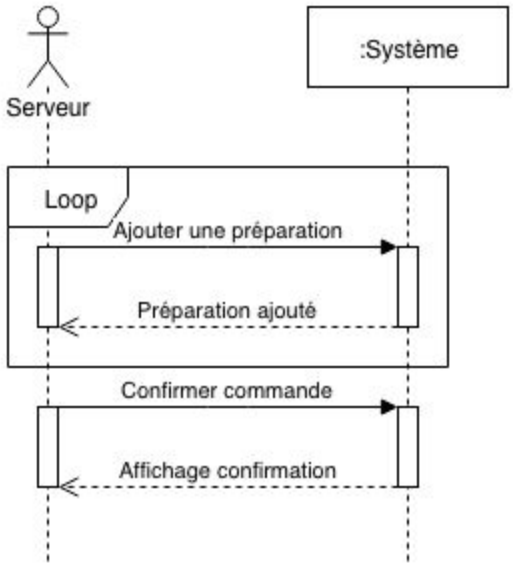
Diagrammes de séquence système

Service

Prise d'une commande



Ajouter une préparation à une commande



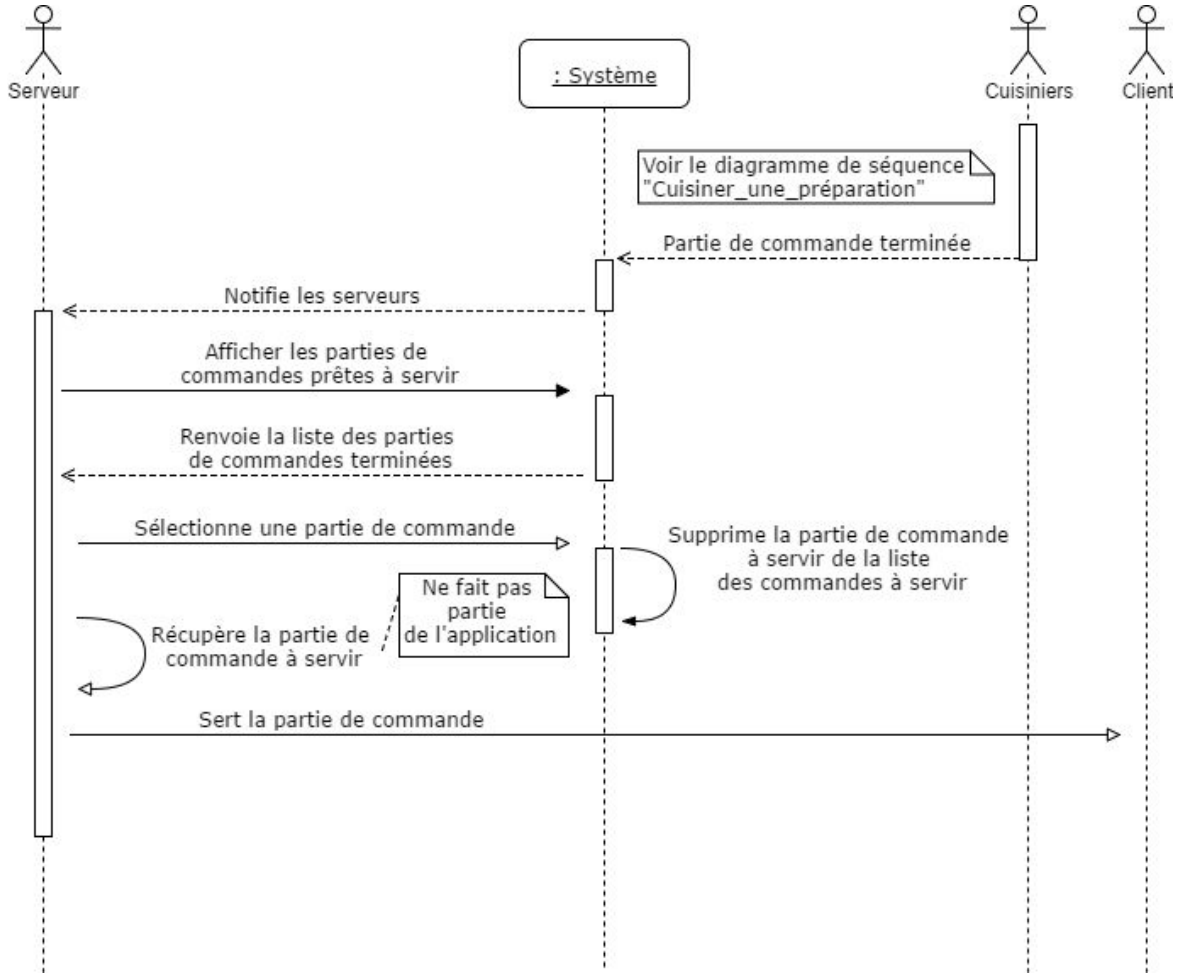
Suivre une commande

XXXXXXXXXXXXXXXXXXXXXXXXXXXX

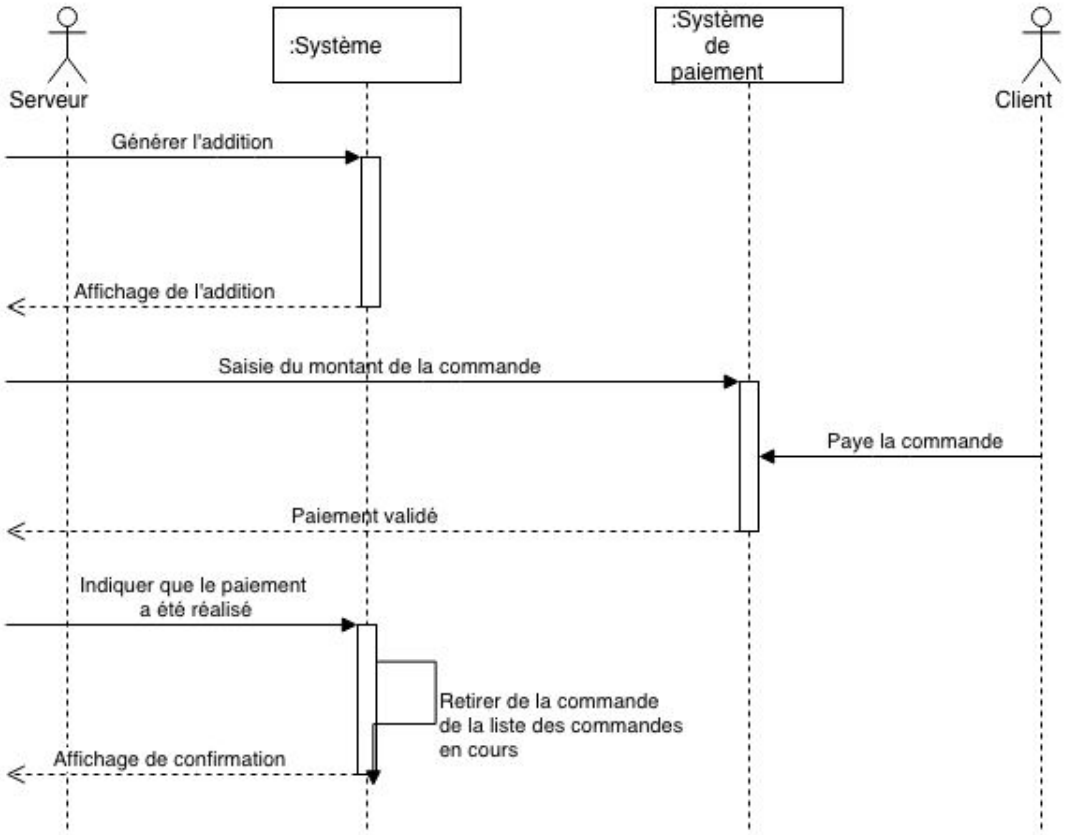
Sera réalisé à la prochaine itération.

XXXXXXXXXXXXXXXXXXXXXXXXXXXX

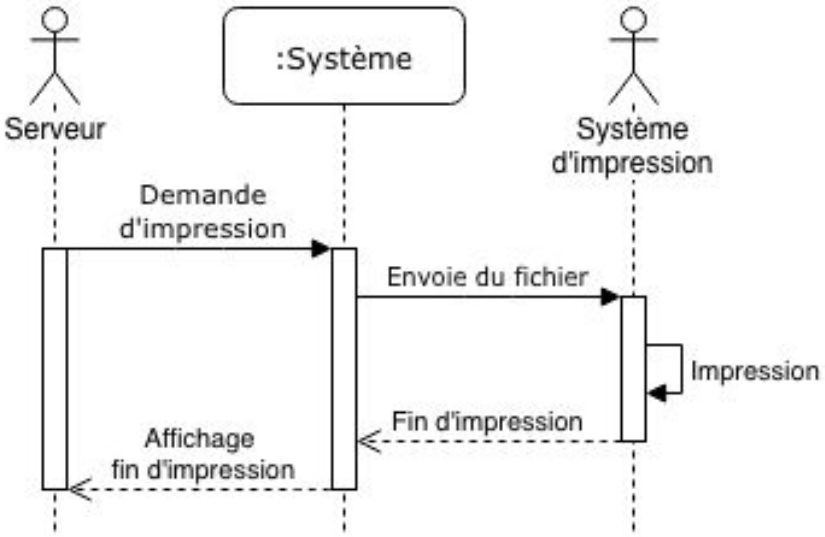
Servir une partie de commande



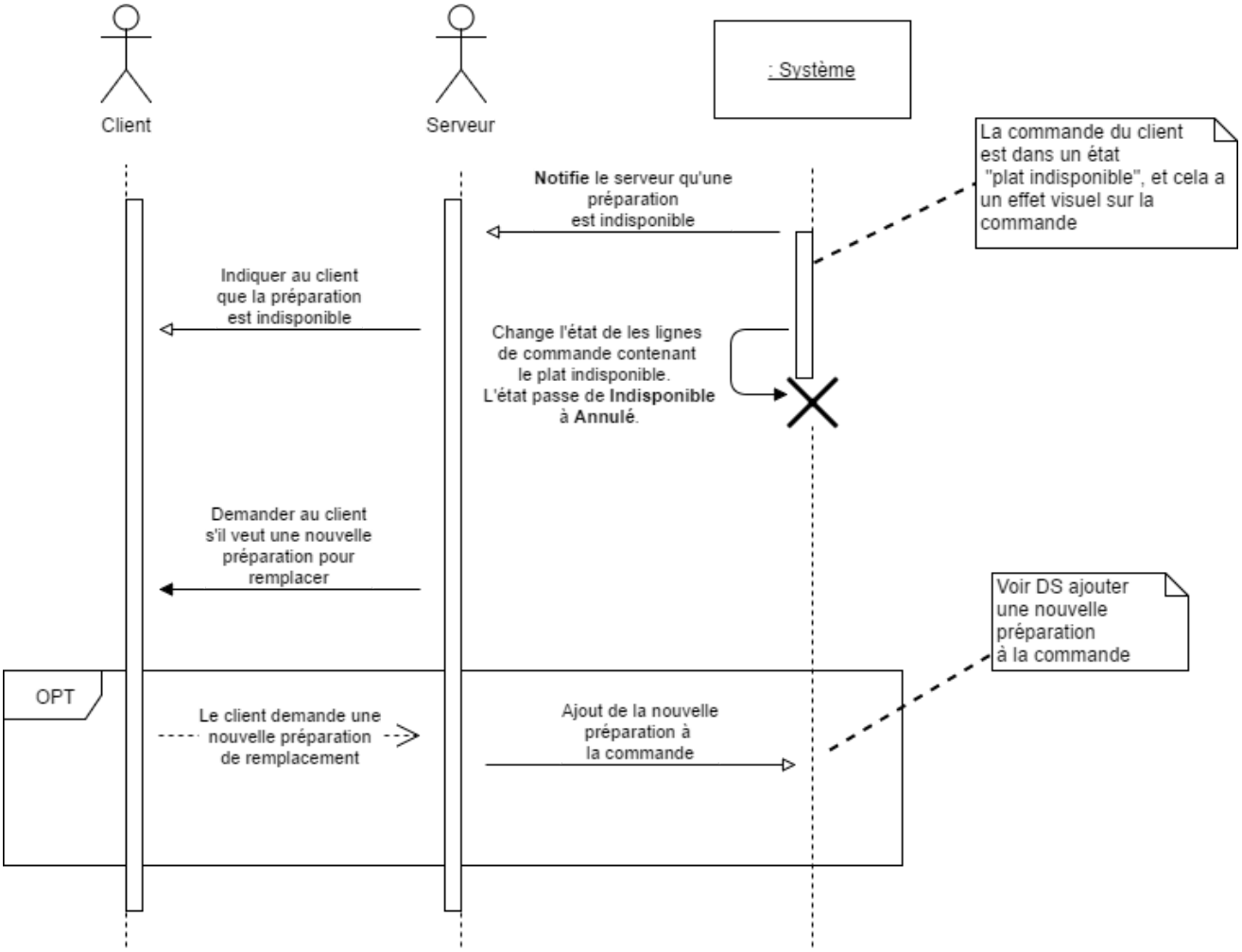
Paiement


Serveur
:Système
:Système de paiement
Client
Générer l'addition
Affichage de l'addition
Saisie du montant de la commande
Paye la commande
Paiement validé
Indiquer que le paiement a été réalisé
Retirer de la commande de la liste des commandes en cours
Affichage de confirmation

Imprimer l'addition d'une commande

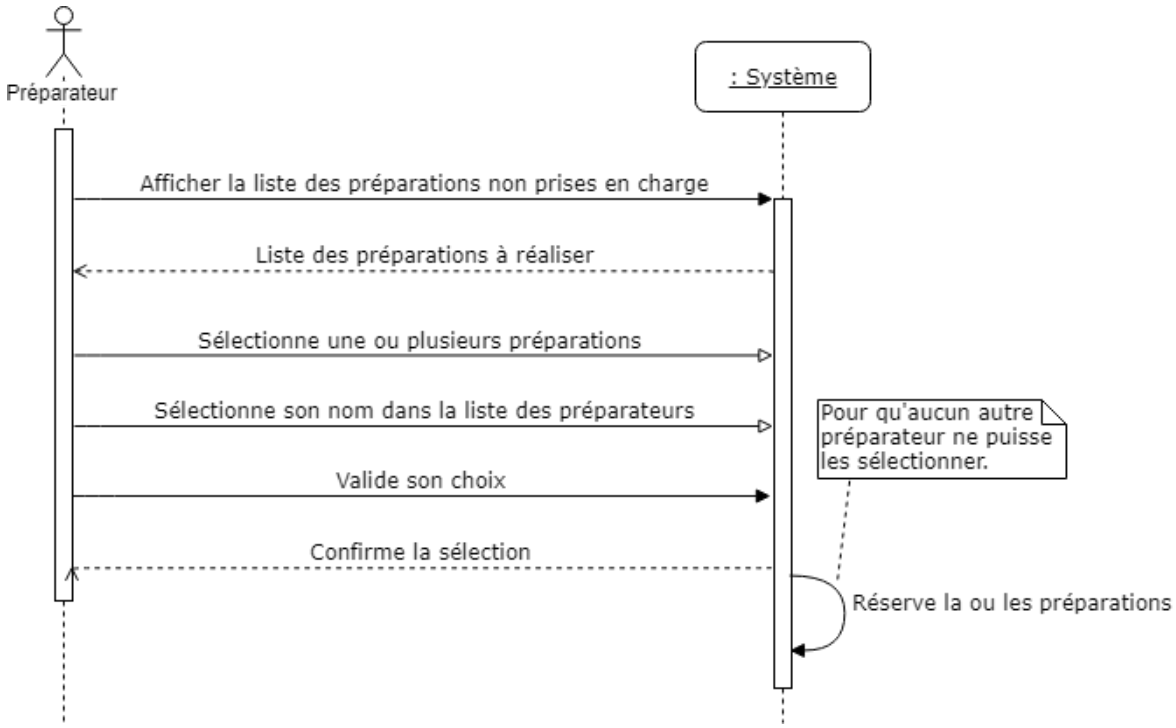


Préparation indisponible

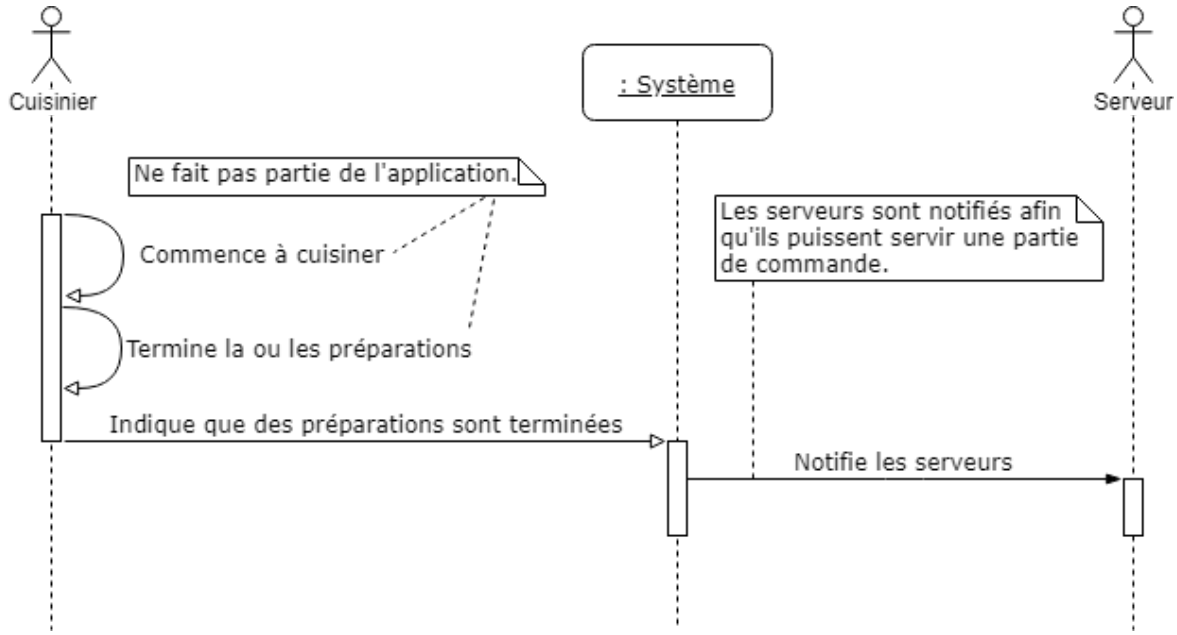


Cuisine

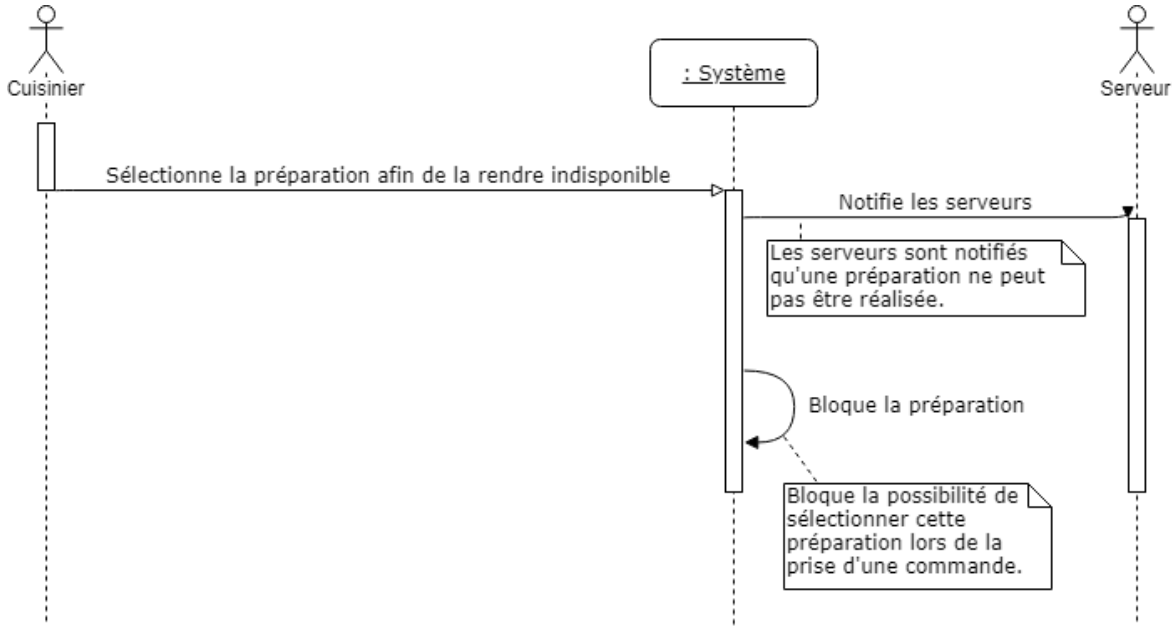
Prendre en charge une préparation



Cuisiner une préparation

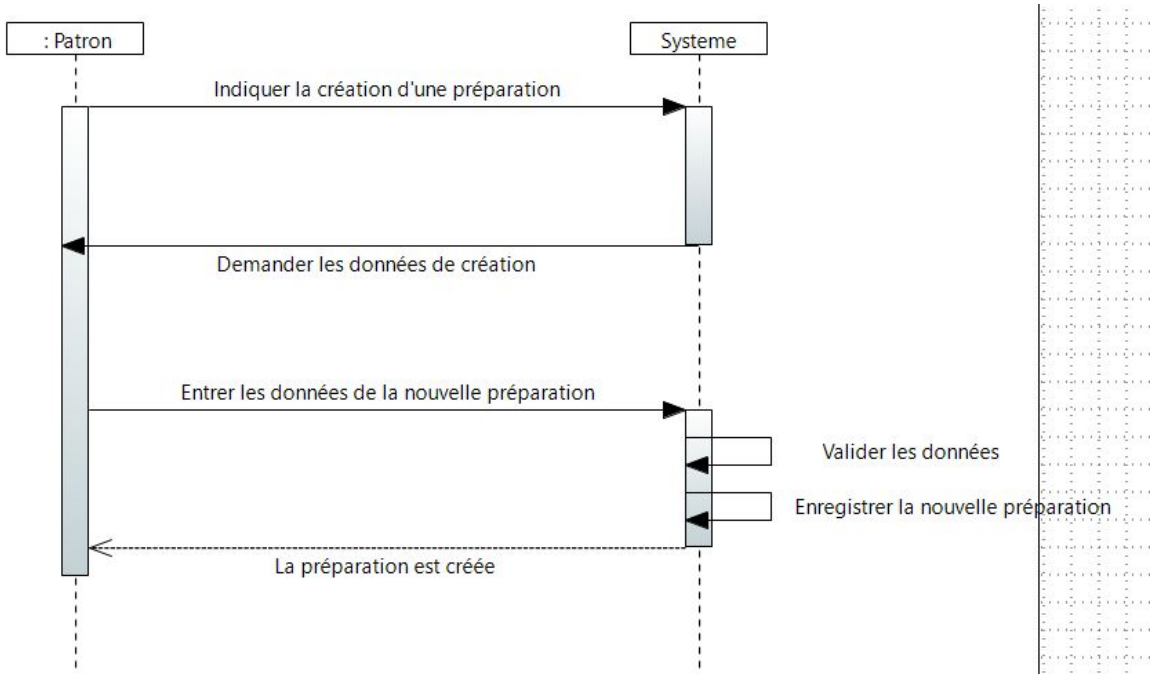


Préparation indisponible

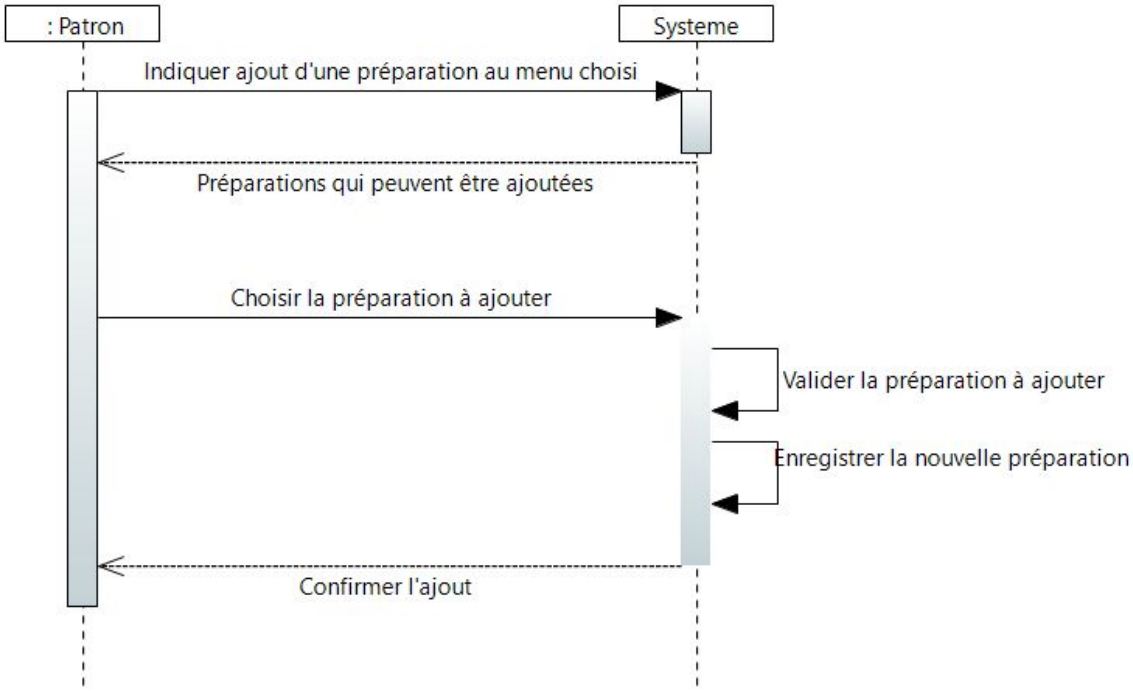


Menu

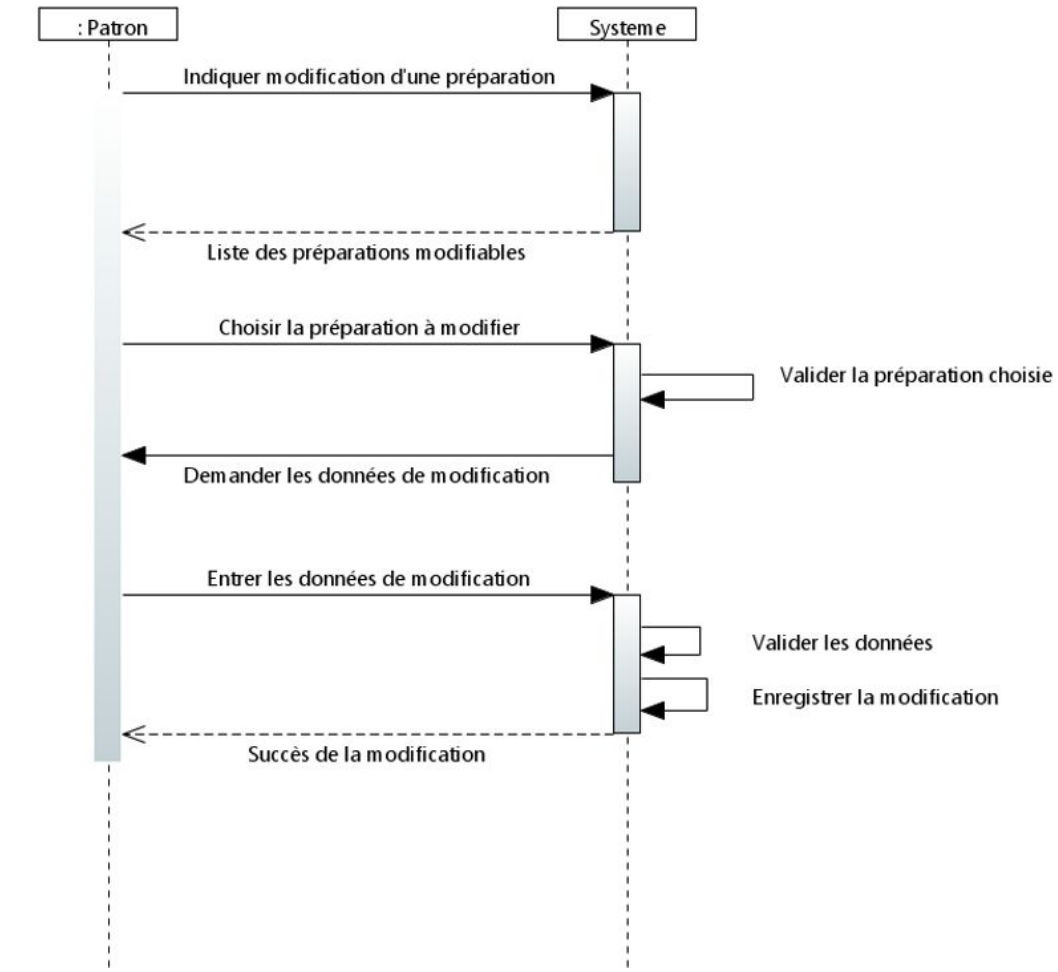
Créer une préparation



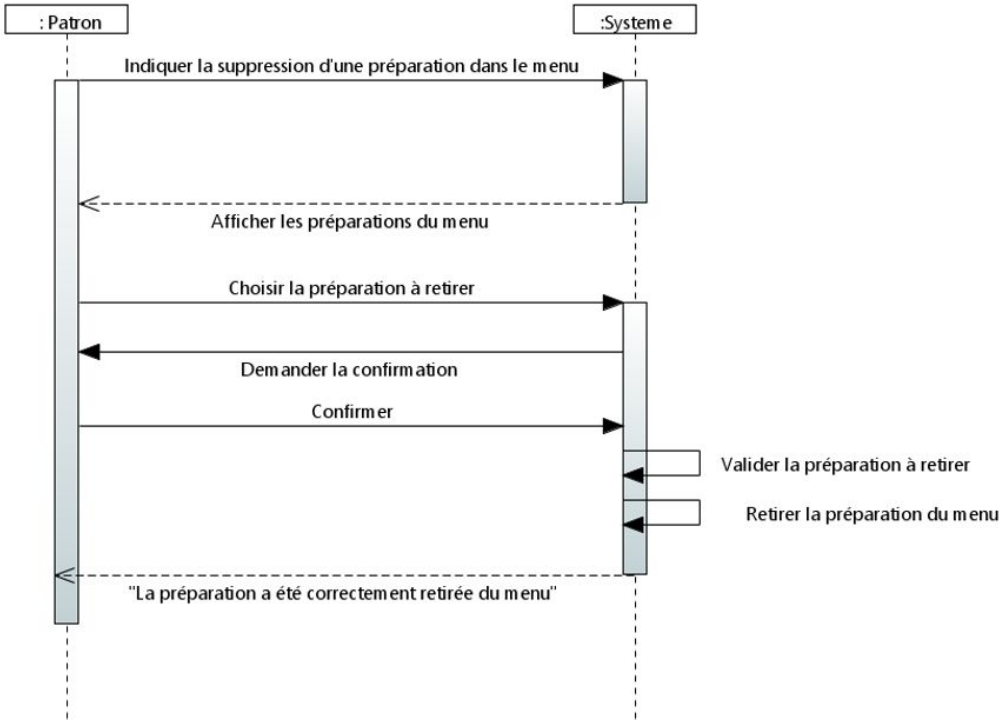
Ajouter une préparation au menu



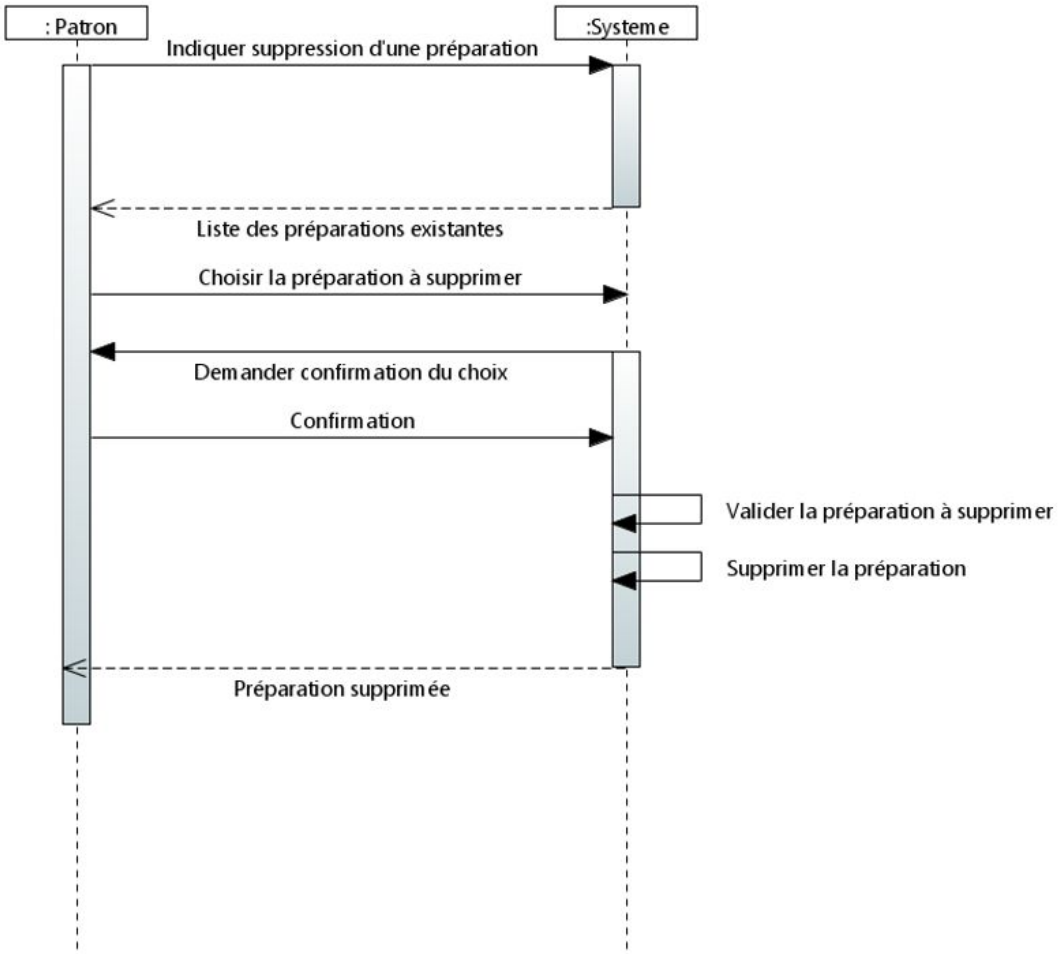
Modifier une préparation



Retirer une préparation au menu

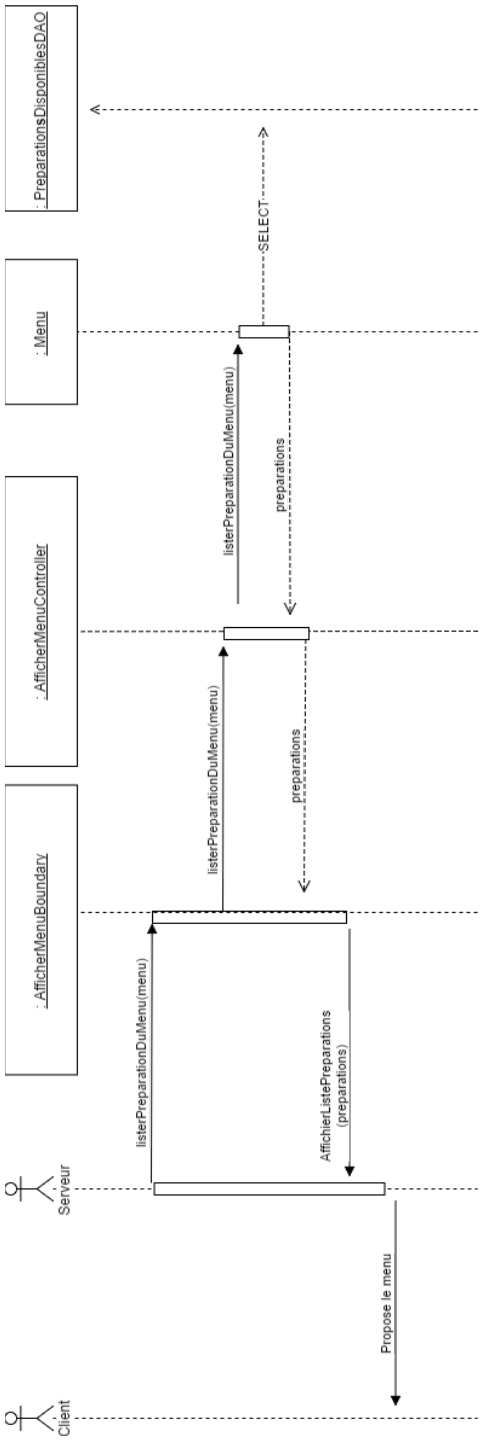


Supprimer une préparation

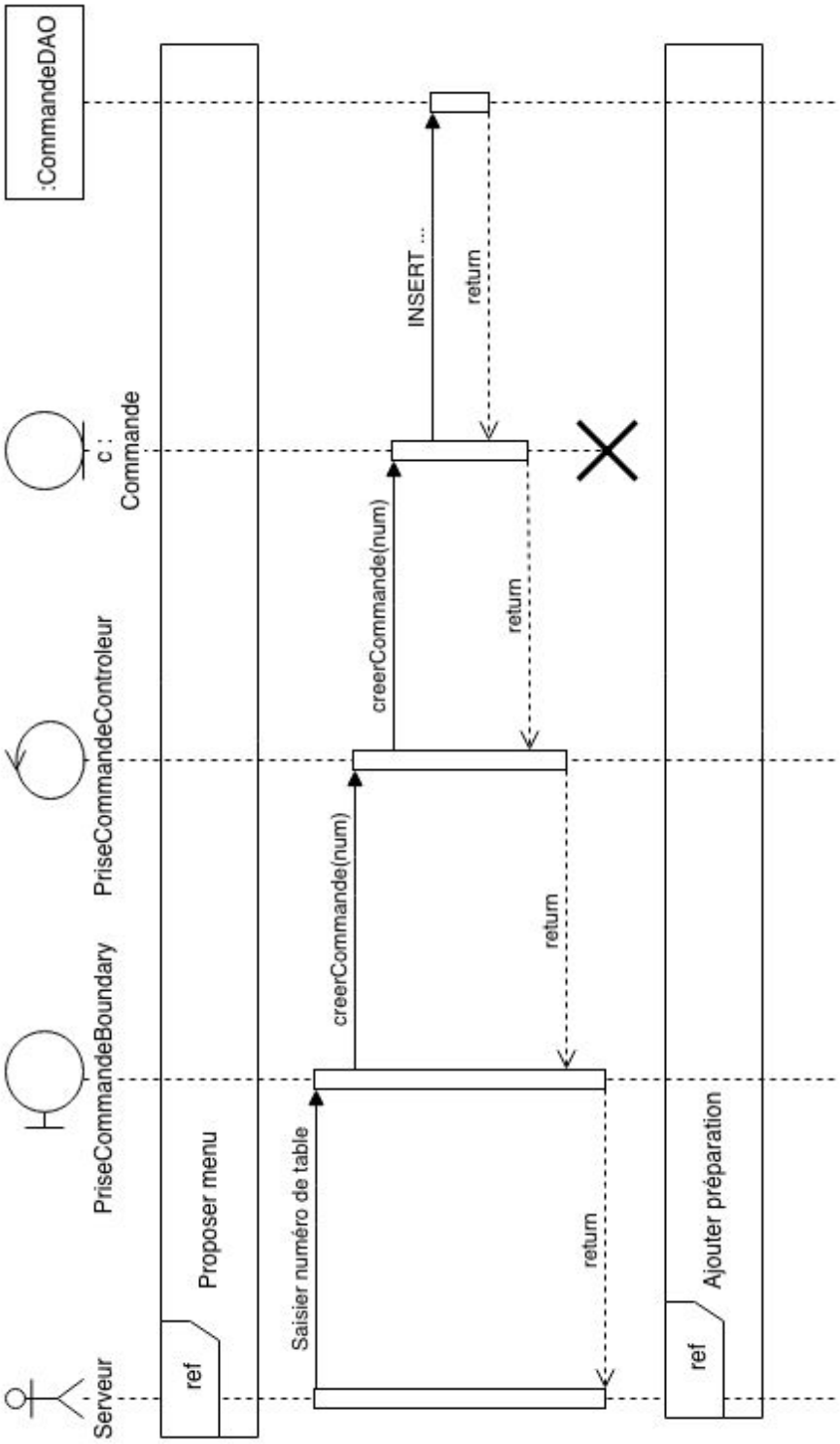


Diagrammes de séquence détaillés

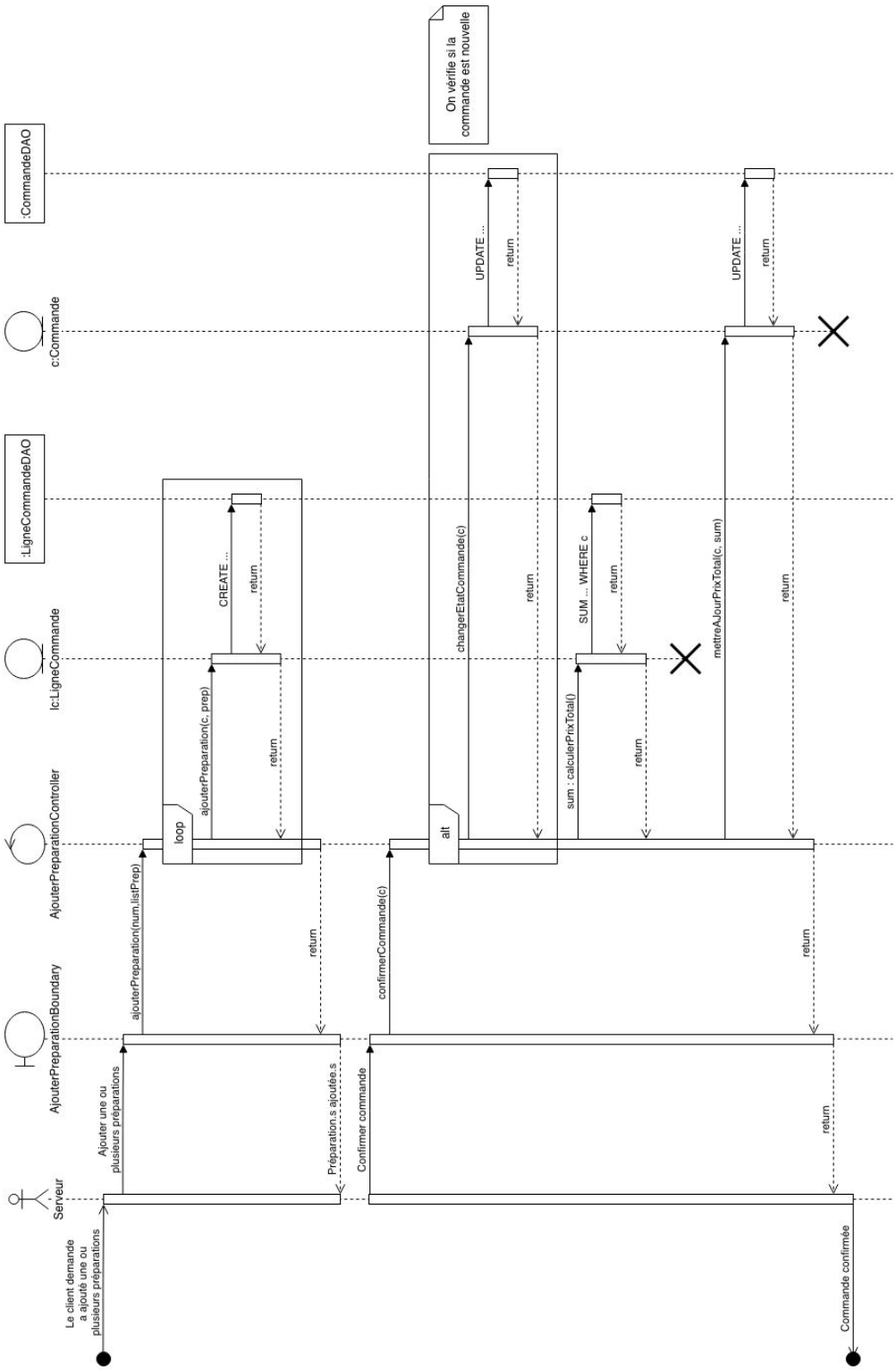
Proposer menu



Prise d'une commande



Ajouter une préparation à une commande



Suivre une commande

XXXXXXXXXXXXXXXXXXXXXXXXXXXX

Sera réalisé à la prochaine itération.

XXXXXXXXXXXXXXXXXXXXXXXXXXXX

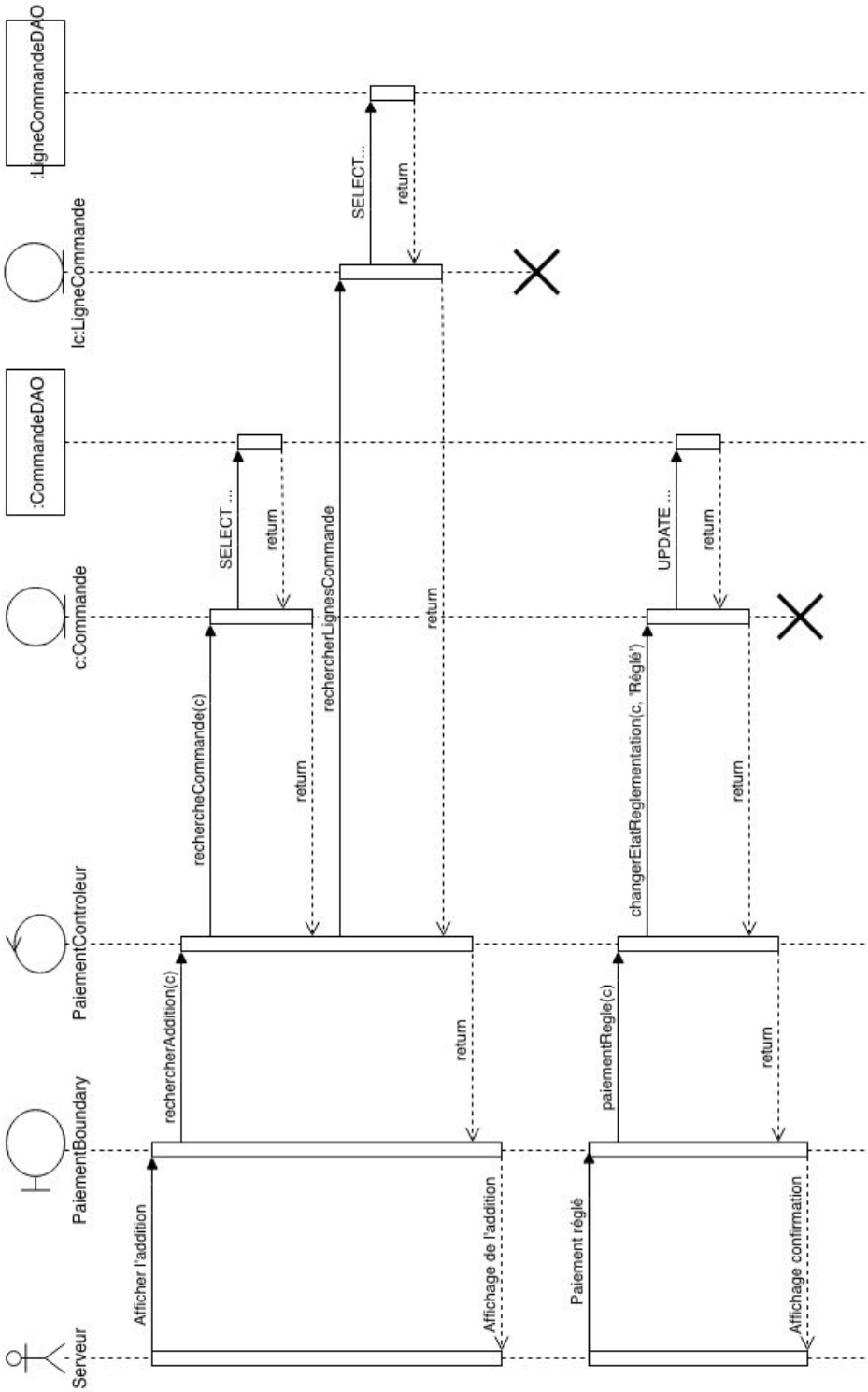
Servir une partie de commande

XXXXXXXXXXXXXXXXXXXXXXXXXXXX

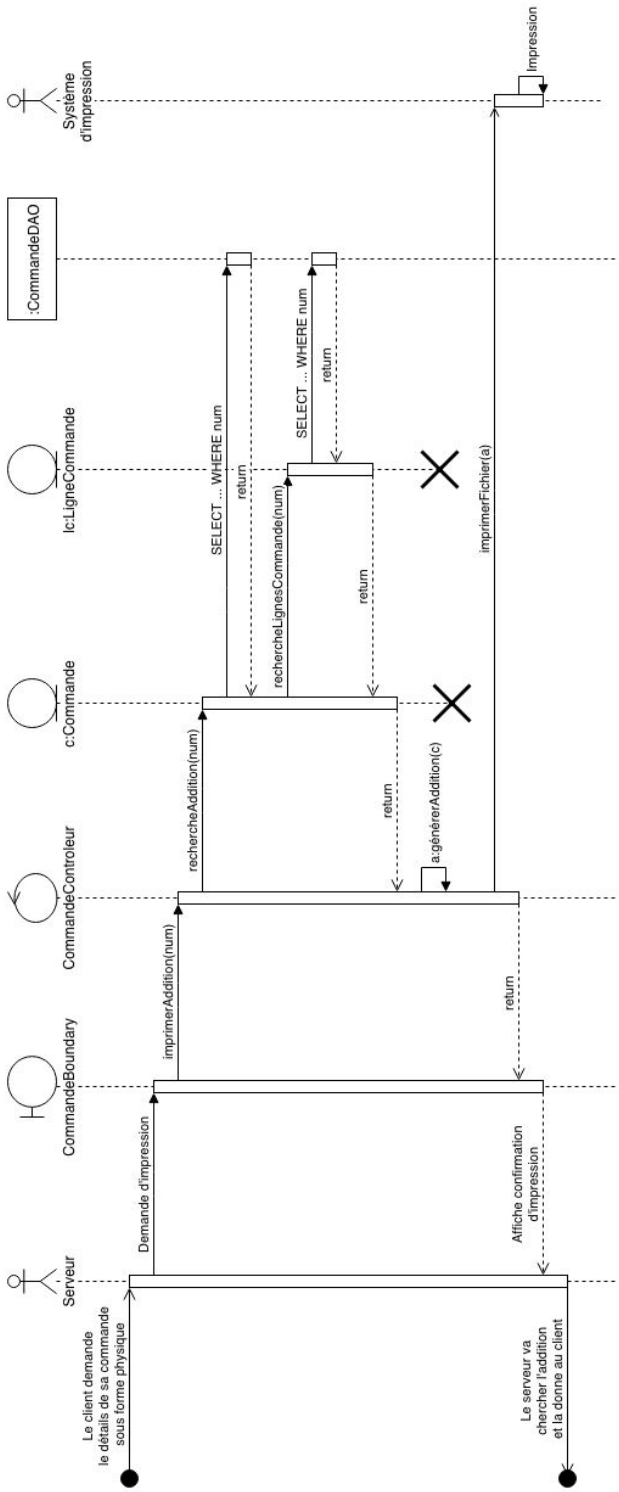
Sera réalisé à la prochaine itération.

XXXXXXXXXXXXXXXXXXXXXXXXXXXX

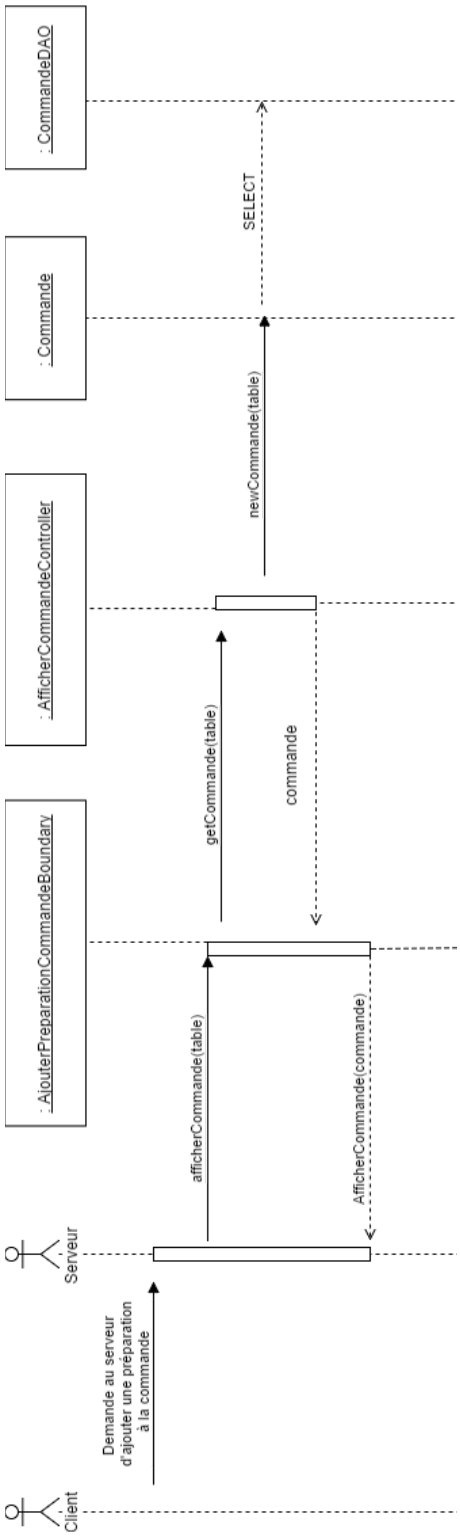
Paiement


:LigneCommandeDAO
lc:LigneCommande
:CommandeDAO
c:Commande
PaiementControleur
PaiementBoundary
Serveur
SELECT...
return
SELECT ...
return
rechercherLignesCommande
return
rechercheCommande(c)
return
UPDATE ...
return
changerEtatReglementation(c, 'Réglé')
return
rechercherAddition(c)
return
paiementRegle(c)
return
Afficher l'addition
Affichage de l'addition
Paiement réglé
Affichage confirmation

Imprimer l'addition d'une commande

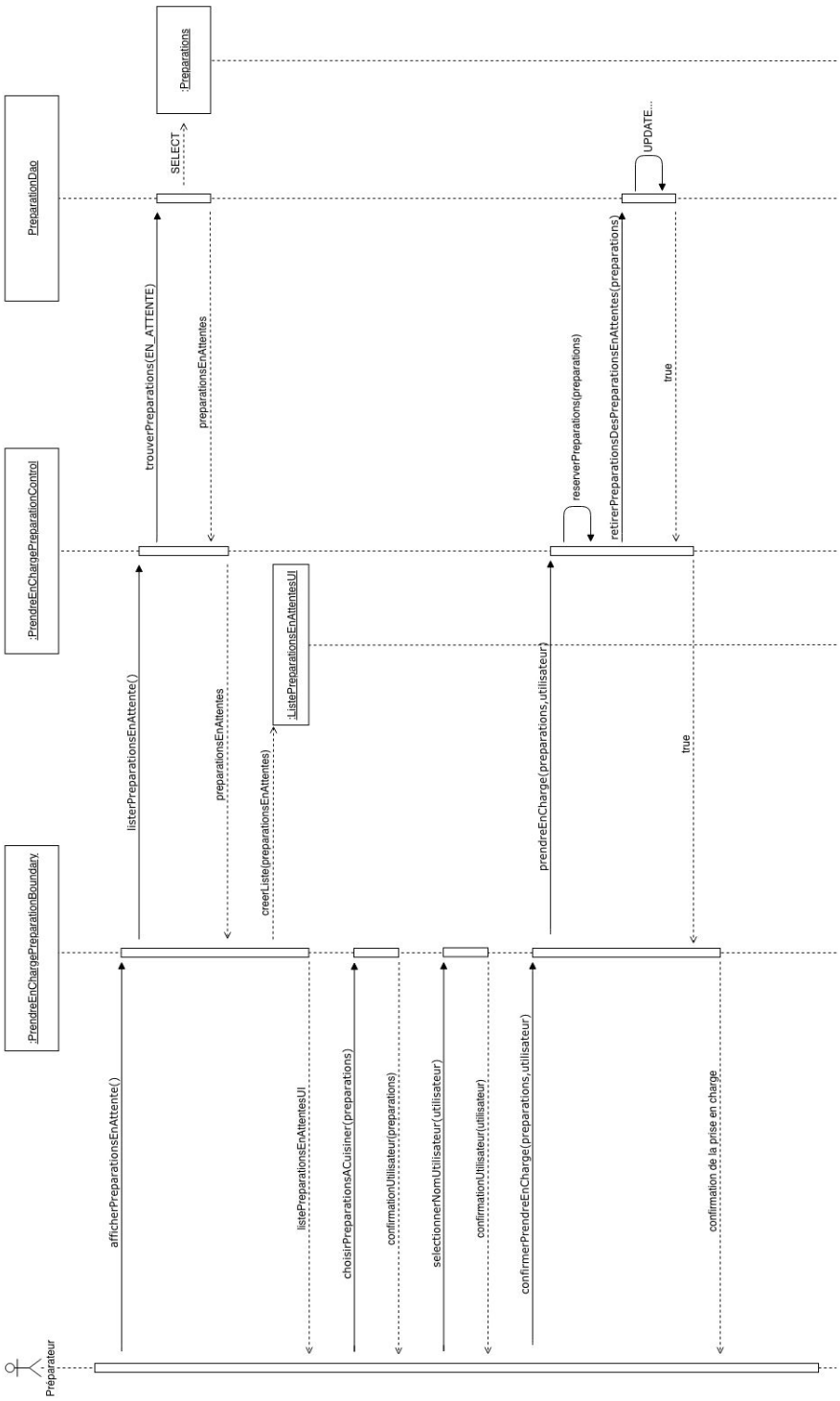


Afficher une commande

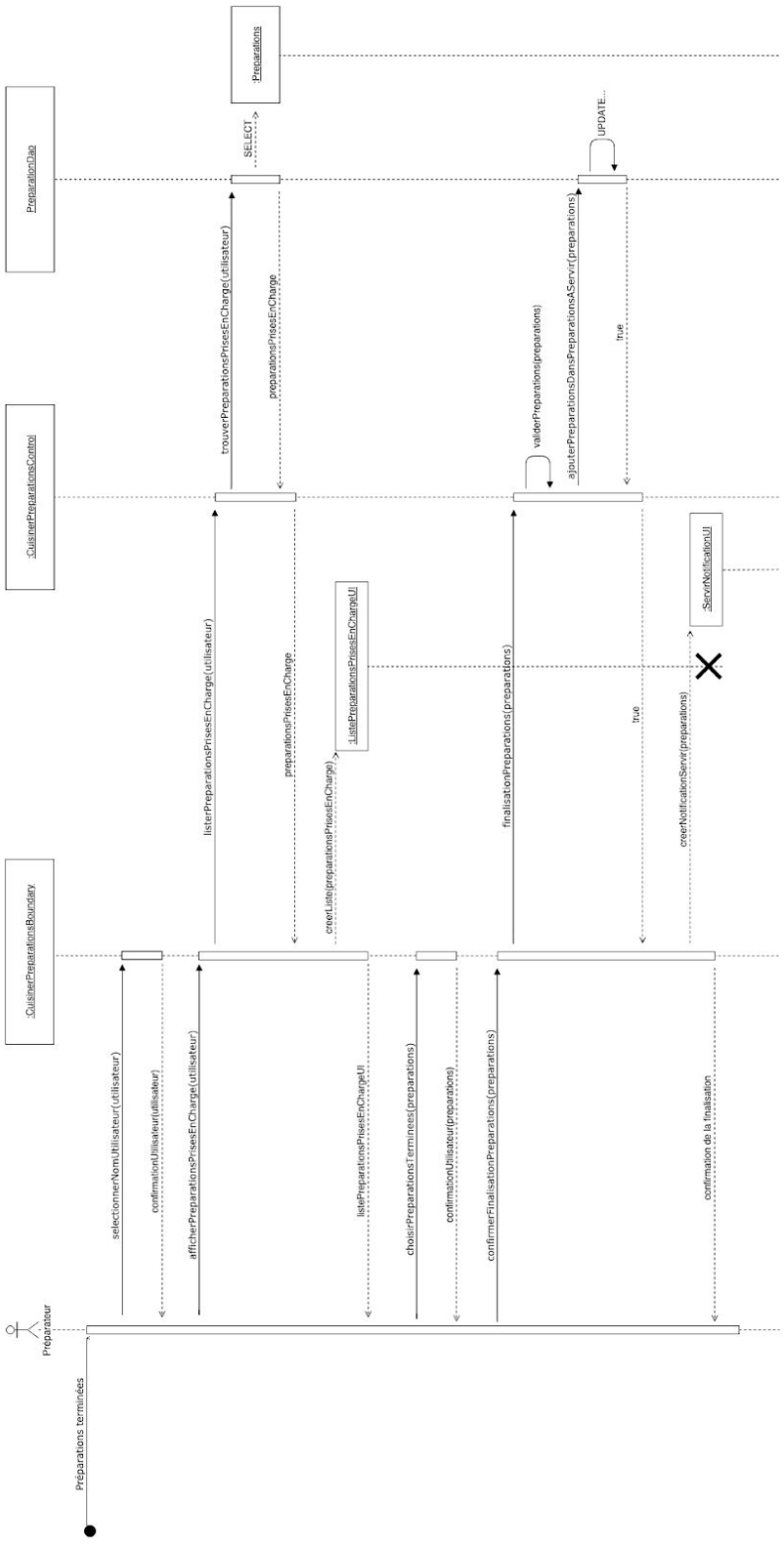


Cuisine

Prendre en charge une préparation

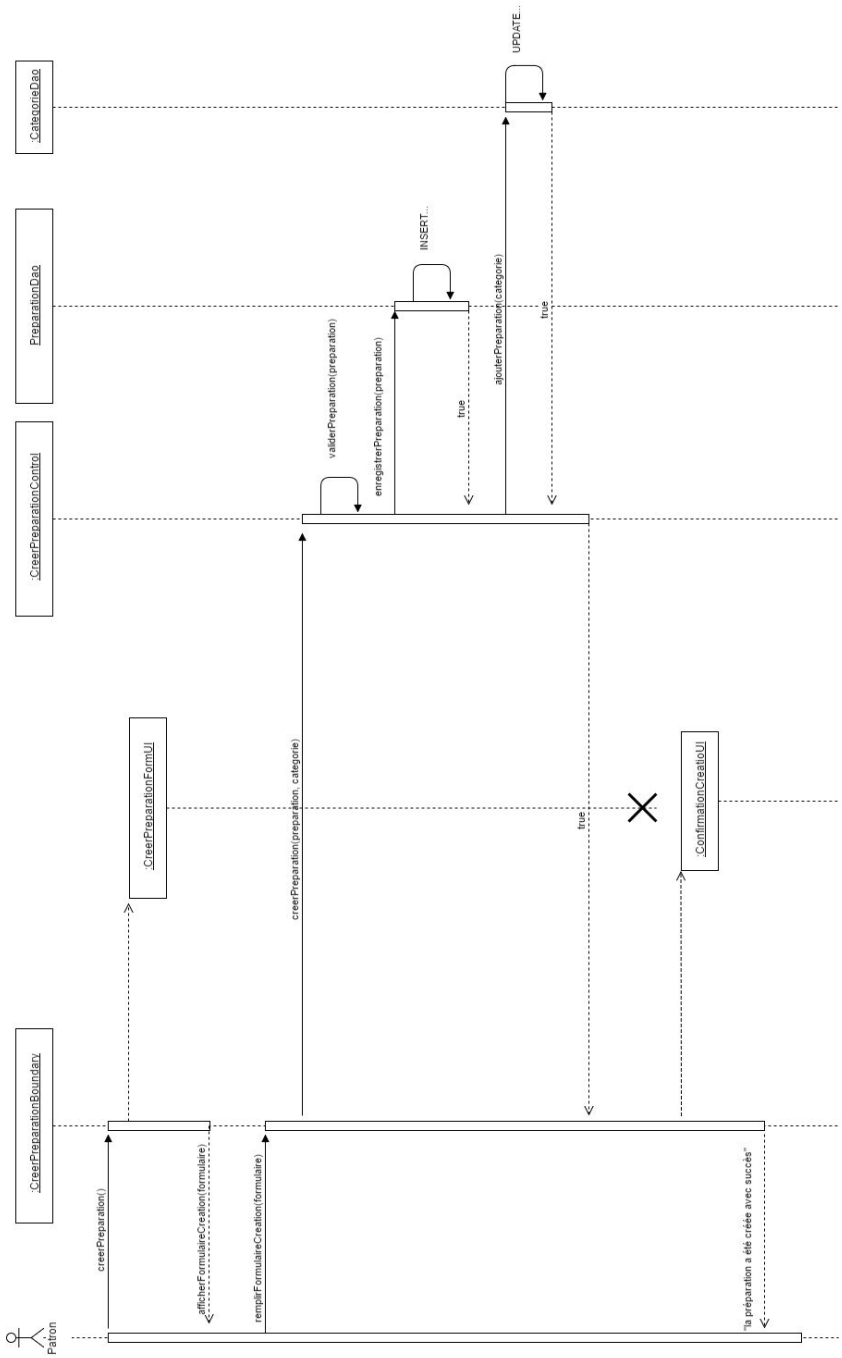


Cuisiner une préparation

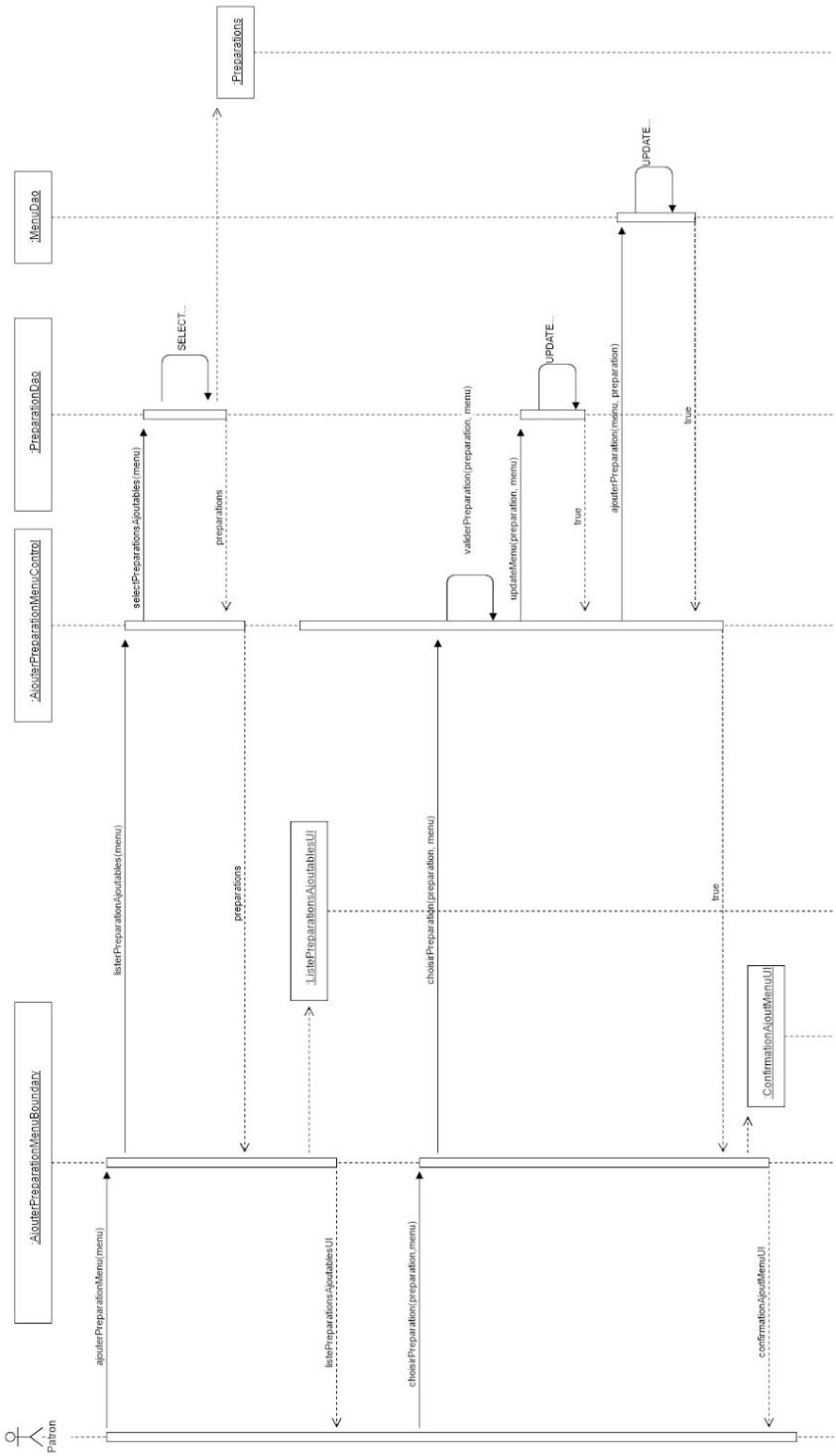


Menu

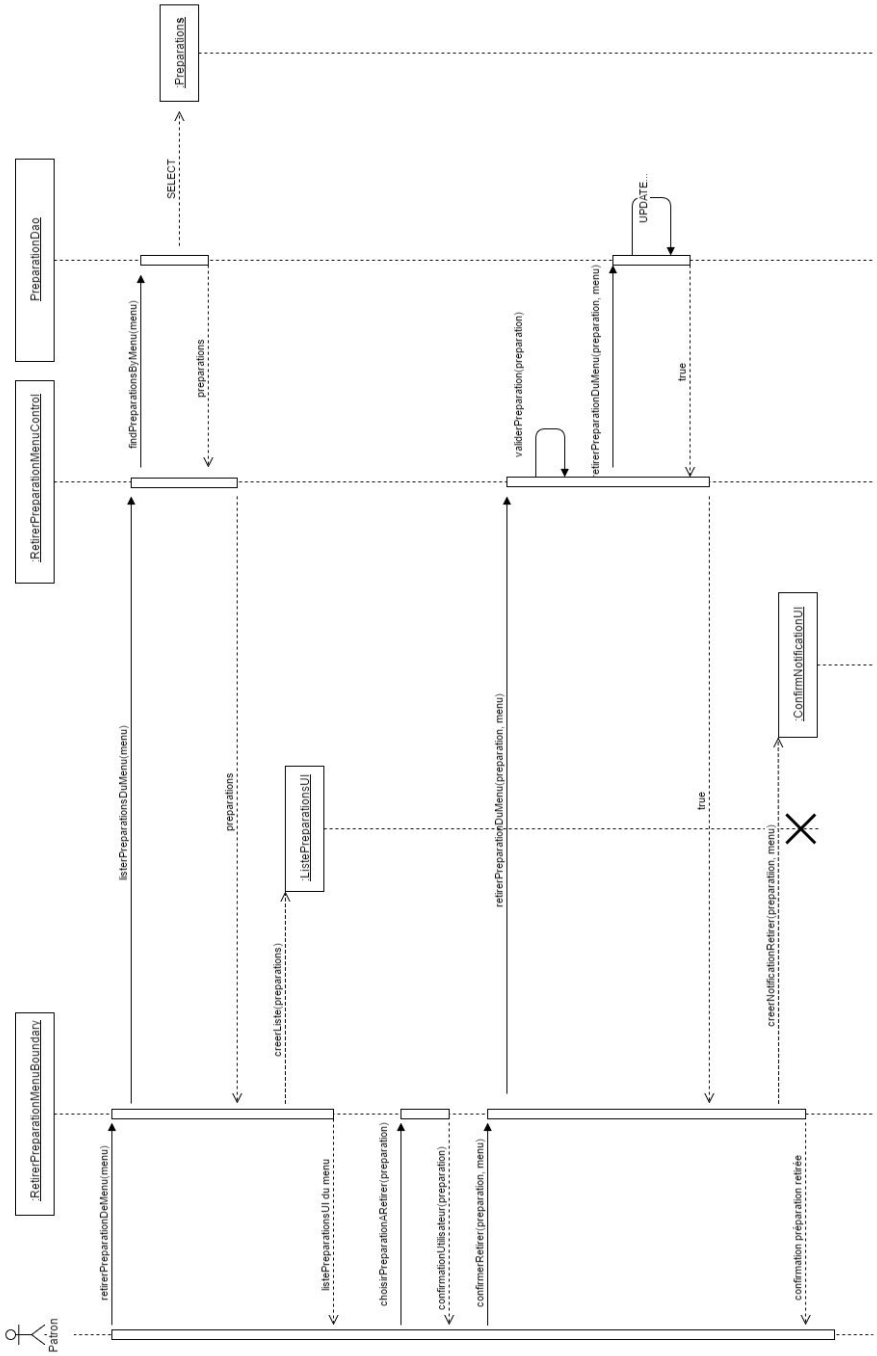
Créer une préparation



OUTER UNE PRÉPARATION AU MENU

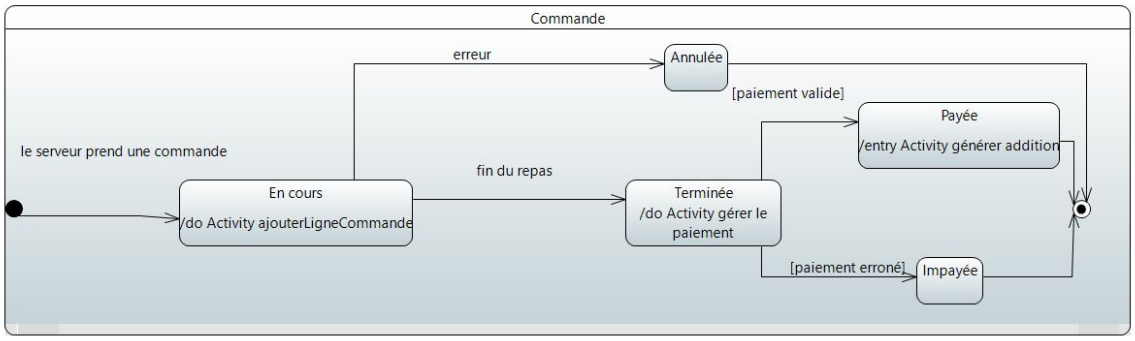


Retirer une préparation au menu



Diagrammes d'états

Les diagrammes ci-dessous décrivent les différents états par lesquels passent les commandes et les lignes de commandes à travers leurs lignes de vie :



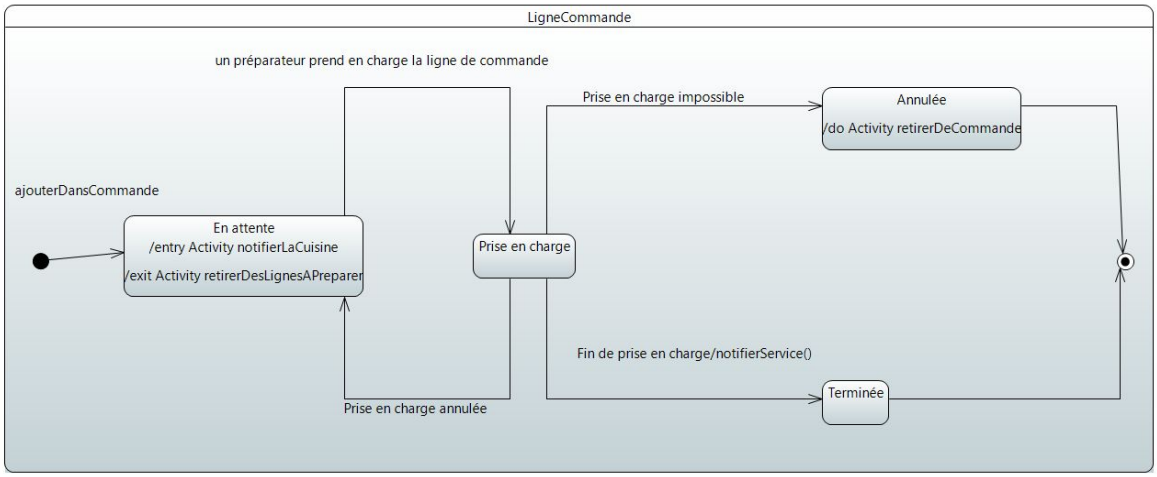
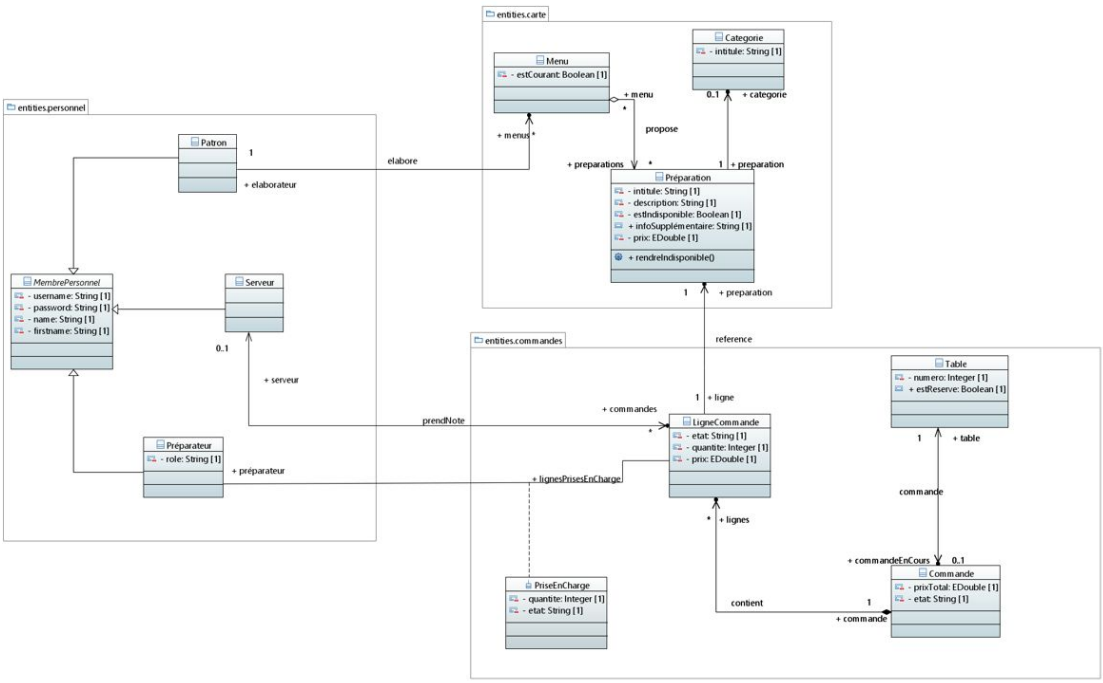


Diagramme de classes

Avertissement : les diagrammes de classes qui suivent sont encore en construction suivant l'archiecture logicielle mise en place. C'est pourquoi ce qui suit est une ébauche du diagramme de classes de l'architecture.

Ci-dessous est détaillé le diagramme de classe des entités :



Explication du diagramme de classe

Les utilisateurs

- **MembrePersonnel** représente tout utilisateur connecté de l'application et contient les infos de ces utilisateurs.
- **Patron**, **Serveur** et **Préparateur** représentent les différents utilisateurs pouvant se connecter et héritent donc de **MembrePersonnel**
- **Préparateur** représente tout membre du restaurant faisant partie des cuisines
    - Exemple : glacier, barman, cuisinier…

Classe "Table"

- Les clients, qui ne manipulent pas directement l'application, s'installe à des **Tables** qui seront utilisées pour la gestion des commandes dans le système.
- Les clients n'ont donc pas de représentation dans le système puisqu'ils n'interagissent pas avec ce dernier.
- Chaque table possède un **numéro** unique.
- Lorsque le client commence sa commande, celle-ci devient la **commandeEnCours** de la table.
- Lorsque l'addition est réglée, la **commandeEnCours** s'ajoute aux **commandesPassées** de la table.
    - Le champ **commandesPassées** est additionnel et existe à des fins d'historique non nécessaires dans l'immédiat.

Gestion du menu

- Une **Préparation** représente un plat, une entrée, un dessert, etc
- Une **Préparation** peut être reliée à une **Catégorie**.
- **Catégorie** représente une catégorie de préparation telle que Poisson, viande, dessert, boisson, etc
- Le **prix** d'une **Préparation** représente le prix courant de cette préparation à la carte.
- Un **Menu** propose plusieurs **Préparations** et une **Préparation** peut être proposée par plusieurs **Menus** différents.
- Il n'existe qu'un seul **Menu** courant qui est celui sur lequel on peut commander durant le service.
- Un **Patron** a tous les droits sur tous les **Menus**, il peut donc les créer/modifier/supprimer/etc.
- Une **Préparation** peut être temporairement **indisponible** (on ne peut plus la commander).

Gestion de la commande

- Une **Commande** possède plusieurs **états** :
    - Créée, En cours, Annulée, Terminée
- Afin d'éviter des calculs inutiles, la **Commande** retient son **prix total**
- Une **Commande** possède plusieurs **LigneCommande**.
- Une **LigneCommande** représente une **Préparation** choisie par le client ainsi que la quantité de cette **Préparation** à faire (ex : 2 spaghettis)

    - **Explication** : comme un client ne commande pas toujours tout ce qu'il veut en un coup au restaurant, il est plus sage de décomposer la commande en plusieurs lignes correspondant à ses demandes.

- Une **LigneCommande** possède elle-même plusieurs états :
    - En attente, En cours, Terminée, Servie, Annulée

- La **LigneCommande** est la donnée la plus manipulée par le système puisqu'elle sera :
    - créée par un serveur quand il **prend note** des demandes d'une table
    - notifiée en cuisine et en attente d'une prise en charge en cuisine.
    - en cours quand elle sera prise en charge en cuisine.
    - notifiée en service et terminée quand la prise en charge en cuisine est finie.
    - servie à la table par le serveur.
    - annulée si un problème survient.

- Une **LigneCommande** possède un **prix** (le prix total de la ligne s'obtient en faisant `prix * quantité`)
    - **Explication** : on ne peut pas se référer au prix courant dans la classe **Préparation** car ce prix courant change au cours du temps. Cependant, si je veux consulter l'état d'une commande d'il y a un mois, peut-être que le prix de mes spaghettis ont changé. Ceci m'amènerait à voir un prix qui n'est pas celui qui était correct un mois auparavant.

- Un **Préparateur** qui prend en charge une **LigneCommande** va générer une **PriseEnCharge**.
    - **Explication** : si une **LigneCommande** demande 64 spaghettis, il est peu probable qu'un seul **Préparateur** s'en charge. Donc chaque **Préparateur** choisit la **quantité** de spaghettis dont il s'occupera.

- Une **PriseEnCharge** a trois **états** :
    - En cours, Terminée, Annulée

Diagramme de classe pour la partie cuisine

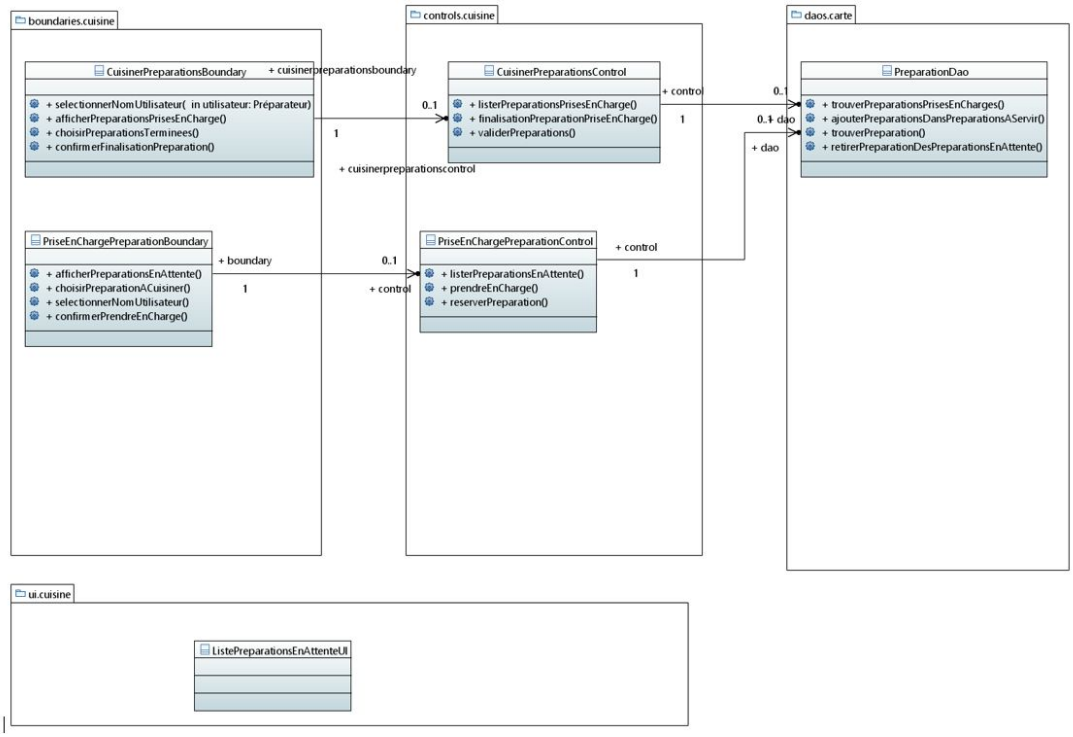


Diagramme de classe pour la partie carte

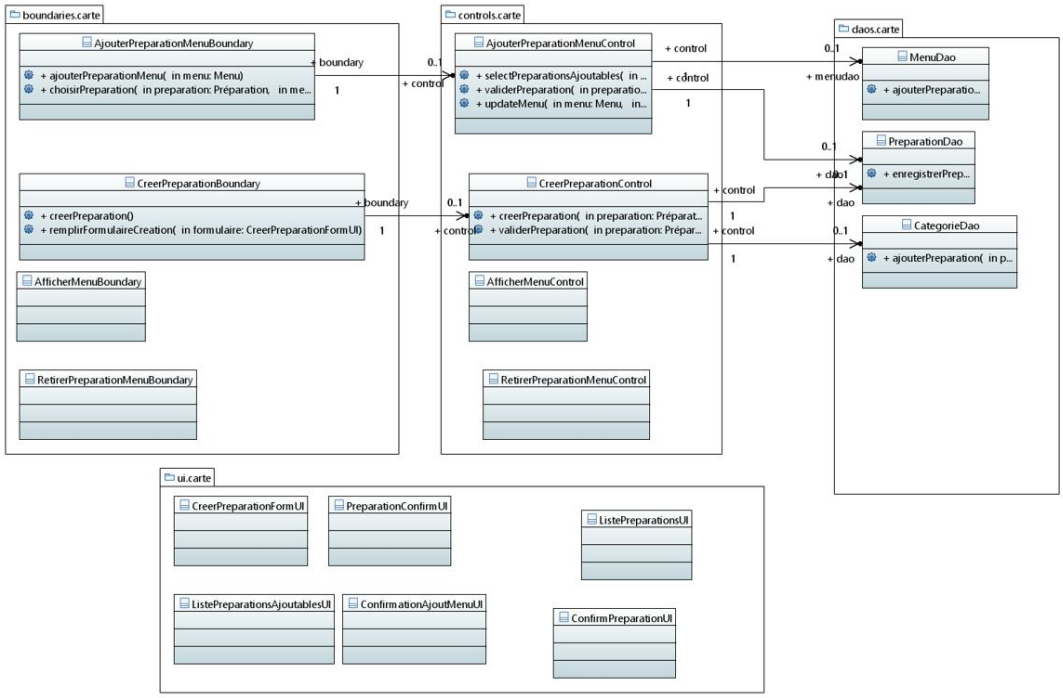
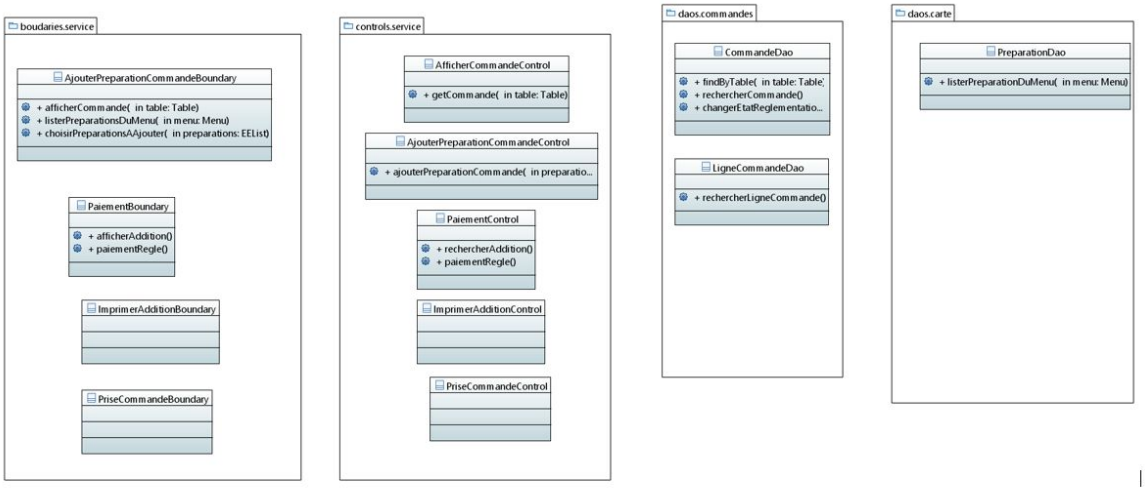
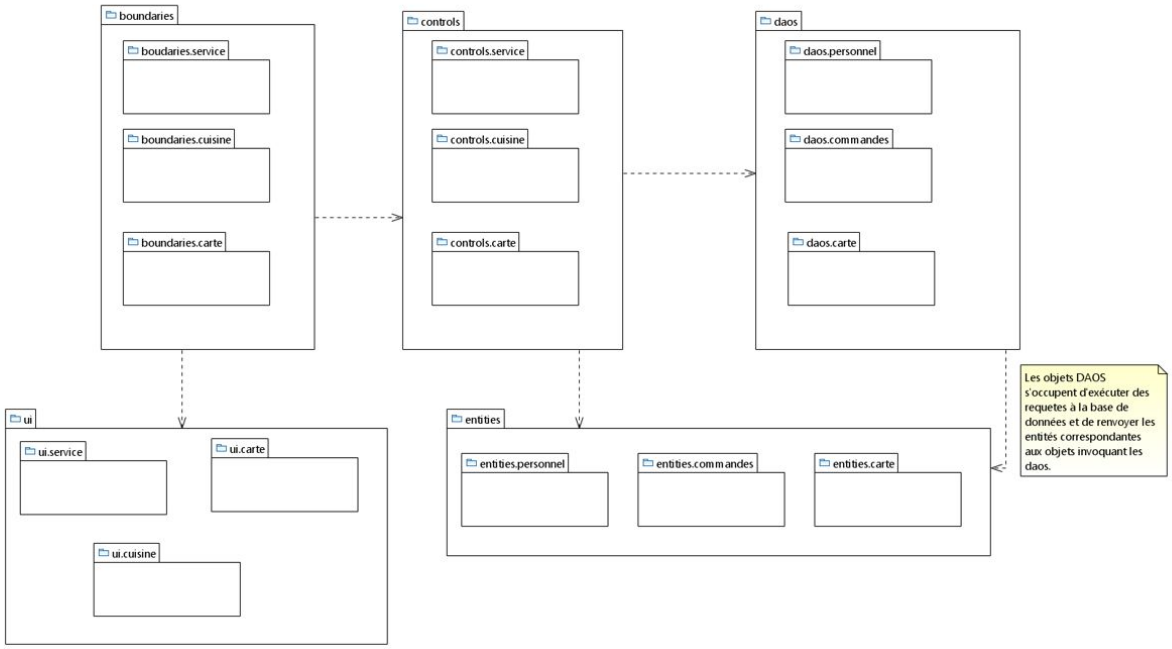


Diagramme de classe pour la partie service



Description de l'architecture logicielle

Ci-dessous se trouve le diagramme de package, représentant la manière dont notre application fonctionne.



Organisation

L'application est composée de 5 principaux packages :

- **boundaries** : contient les classes Boundary du système. Ces classes s'occupent de gérer l'interaction avec l'utilisateur.
- **ui** : contient les classes UI du système. Ces classes s'occupent de l'aspect visuel du système.
- **controls** : contient les classes Control du système. Ces classes implémentent la logique métier.
- **daos** : contient les classes DAO (Data Access Object). Ces classes s'occupent d'accéder et de gérer les données persistées dans la base de données.
- **entities** : contient les classes Entity du système. Ces classes sont des classes représentant les données persistées dans la base de données.

On distingue trois grandes logiques à laquelle répondent les packages :

Interaction avec l'utilisateur

Représente toute la logique d'interaction avec l'utilisateur (affichage, etc).

Les packages **ui** et **boundaries** sont chargés d'assurer l'interaction avec l'utilisateur.

- **ui** : gère l'affichage.
- **boundaries :** gère les actions que peut effectuer un utilisateur sur le système.

Logique métier (business)

Représente toute la logique métier derrière les cas d'utilisation.

Cette logique est implémentée par le package **controls**.

Logique de gestion des données

Représente tout ce qui est lié à la gestion des données persistées en base de données par le système.

Cette logique est implémentée par les package **daos** et **entities**.

- **entities** : renferme les classes qui contiennent les données brutes en base de données. En bref, ce package renferme la représentation orientée objet des données persistées.
- **daos** : permet au système d'interagir avec la base de données.

Les sous-packages

Les classes ne sont pas directement placées dans les packages principaux et sont divisées en plusieurs sous-packages, afin d'augmenter la granularité.

boundaries, uis et controls

En ce qui concerne les packages boundaries, uis et controls, les classes sont divisées en sous-packages représentées par les trois contextes fonctionnels que nous avons définis dans nos diagrammes de cas d'utilisation :

- **carte :** gestion de la carte des menus/préparations/...
- **cuisine** : gestion de la logique pour les préparateurs (cuisiniers, barmar, etc)
- **service** : gestion de la logique pour les serveurs.

Ainsi, les classes de ui.service s'occupent de l'affichage pour les UC liés au contexte du service.

entities et daos

Décomposer les entités et daos suivant les contextes fonctionnels des UC n'étaient pas une bonne idée pour les entités et daos puisque ces packages s'occupent essentiellement de la **logique de gestion de données**.

Trois types de sous-packages sont définis :

- **personnel** : gestion des données des préparateurs/serveurs/etc
- **commandes** : gestion des données des commandes
- **carte :** gestion des données de la carte.

Dépendances entre packages

- Les objets **boundaries***créent et renvoient les objets uis.
- Les objets boundaries appellent la logique métier implémentée dans les objets de controls.
- Les objets de controls appellent la gestion de données des objets daos.
- Les objets daos, après avoir interagit avec la base de données, manipulent les objets entities.
- Les objets controls récupèrent les objets entities pour appliquer la logique métier sur ceux-ci.

Glossaire de l'ingénierie des besoins

Portée

Ce glossaire présente et détaille les termes et concepts liés au vocabulaire technique de la solution développée.

Avertissement

Il est supposé ici que le lecteur possède déjà des connaissances techniques de base dans les domaines de l'orientée objet et des diagrammes UML.

Organisation

Le glossaire s'organise en une suite de définitions des différents termes, divisée en sous-catégories relatives aux différentes phases d'analyse.

Chaque terme est décrit en trois parties : un énoncé du terme, sa définition et sa traduction anglaise.

Définitions

Modèle fonctionnel

Terme	Définition	Traduction
Exigences fonctionnelles	Exigences à prendre en compte lors de la création du système et qui est liées aux besoins fonctionnels du client.	Functional requirements
Exigences non-fonctionnelles	Exigences à prendre en compte lors de la création du système et qui n'est pas liées aux besoins fonctionnels du client.	Nonfunctional requirements
Scénario concret	Scénario sus forme libre (texte, vidéo, bande dessinée,...) qui sert à décrire, à travers un récit, une manière d'utiliser le système à développer. Un scénario concret est abstrait de toutes les notions techniques du système.	Scenario

Modèle objet (architecture)

Terme	Définition	Traduction
Objet frontière	Objet du système dont le rôle est de gérer l'interaction avec l'utilisateur (affichage, etc).|	Boundary object
Objet de contrôle	Objet du système dont le rôle est de gérer la logique métier du système.	Control Object
Objet entité	Objet du système chargé de représenter les données persistées dans la base de données.	Entity object
Objet d'accès aux données	Objet du système dont le rôle est de gérer la logique de gestion des données.	Data Access Object (DAO)
Objet d'interface utilisateur	Objet du système dont le rôle est de représenter les interfaces affichées à l'utilisateur.	User Interface Object (UI)

Modèle dynamique

Terme	Définition	Traduction
Diagramme de séquence système (DSS)	Diagramme de séquence dont le rôle est de représenter les interactions entre les acteurs et le système. Ce dernier est vu sous forme de boîte noire (pas de spécification de son modèle objet).	System Sequence Diagram
Description textuelle/Scénario abstrait	Reprise de chaque cas d'utilisation sous forme textuelle pour décrire une interaction entre le système et l'acteur.	

Premières versions des maquettes

Maquettes interactives en ligne à cette adresse. (nous avons créé un dépôt git permettant au site de se mettre à jour automatiquement à chaque push).

L'application aura plusieurs vues en fonction des utilisateurs. Chaque vue sera sélectionnable à gauche de l'écran. Nous souhaitons laisser à tous les membres du restaurant la possibilité de voir les différentes parties de l'application. A priori, les accès seront donnés par des managers. Le mieux sera de le faire en début de chaque jour, avant le service.

Le but de l'application est de gagner du temps. Notre application ne doit donc pas alourdir la charge de travail des préparateurs. Nous ne connaissons pas le nombre de tablettes disponibles dans le restaurant, nous en comptons une par serveur pour le moment, et une ou deux en cuisine. Il sera tout à fait possible d'en mettre moins, car nous sommes bien conscients qu'une tablette coûte assez chère. Cela pose donc certains problèmes, en effet, les préparateurs ne doivent pas perdre de temps en se connectant. C'est pourquoi, nous vous proposons d'utiliser notre application avec un compte pour la cuisine. Chaque préparateur doit juste s'identifier en cliquant sur un bouton avec son nom pour que l'on sache qui cuisine quoi. Cela leur permet de ne pas avoir à rentrer leur mot de passe.

Maquette de la partie cuisine

Maquette coté Cuisine

Dans cette maquette pour pouvons observer qu'il y aura deux vues pour les préparateurs. Une première vue avec les catégories de plats, et une seconde en fonction des commandes.

Le but des catégories était de faire gagner du temps aux préparateurs. Etant donné le fait que la vue des commandes est aussi importante, nous allons aussi la laisser.

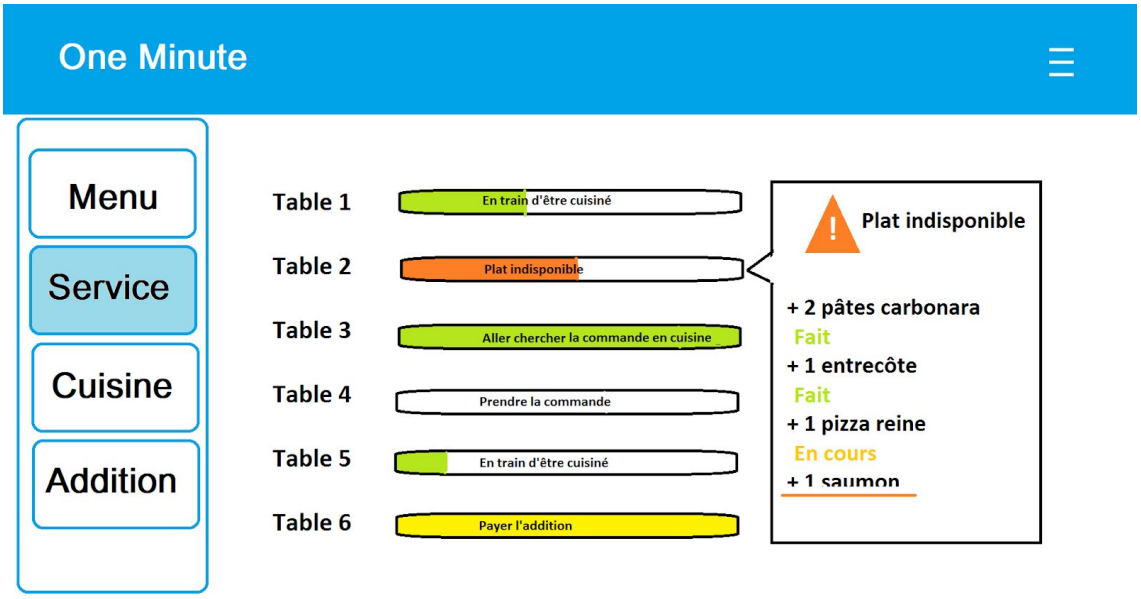
Suite à la création de cette maquette nous ai venu l'idée de laisser la possibilité aux préparateurs de ne pas perdre de temps en s'identifiant. C'est pourquoi ils auront juste à appuyer sur leur photo à droite, valider leur nom et choisir le plat qu'ils veulent s'approprier ou valider. Pourrons pas exemple tous se connecter sur le même compte, un compte spécialement réservé à la partie cuisine.

Maquette de la partie service

Dans cette première version, l'idée était de permettre aux serveurs de recréer le restaurant sous la forme d'une vue. Ceci avait pour but de savoir quelle table était laquelle. Malheureusement cette solution était assez technique et allait prendre du temps.

Première version de la maquette Service

Maquette coté Service



Nous pouvons observer ici la fonctionnalité de voir chaque commande, leurs états, les plats indisponibles s'il y en a, et s'il est temps de payer l'addition. Nous allons également trier les commandes, non plus par table, mais par ordre de priorité.

Glossaire métier

- ***Addition*** : Facture de la commande indiquant la liste des préparations commandées ainsi que le montant total de la commande.
- ***Bar*** *:* Endroit où sont préparées les boissons.
- ***Boisson*** : Préparation.
- ***Commande*** : Note qui sert à savoir ce que le client a choisi. Elle est prise par un serveur, et est transmise aux cuisines.
- ***Cuisine*** : Endroit où sont préparées les préparations.
- ***État*** : État d'avancement de la commande.
- ***Indisponible*** : État d'une préparation qui ne peut pas/plus être réalisée par manque de un (ou plusieurs) ingrédient(s) la constituant.
- ***Note*** : Liste des préparations qu'écrit le serveur.
- ***Partie de la commande*** : pack de boissons, de plats ou de desserts etc.
- ***Préparation*** : Plutôt que plat, nous employons le terme préparation qui est plus générique. Il contient les plats, les boissons, les desserts...
- ***Réapprovisionner*** : Action de recevoir un nouvel apport de marchandises afin de renouveler les stocks.
- ***Service*** : Fonction à gérer pour les serveurs du restaurant. Cela consiste à prendre la commande, amener les plats, et régler la commande.
- ***Stock*** : Ingrédients disponibles pour préparer les commandes.

ANNEXES

Les annexes sont les documents dont la MOA n'a pas besoin. En effet, pour vous permettre de juger au mieux le travail effectué, nous vous invitons à lire nos annexes qui contiennent :

- Notre méthode de travail
- Un retroplanning du rendu 1
- Un retroplanning du rendu 2
- Les synthèses de nos séances
- Le tableau types-concepts

Méthode de travail

Ce document sert à expliquer la manière générale dont on travaille.

En séance

L'objectif de nos deux heures de TPs par semaine est principalement de débattre de nos manières de voir la solution au problème posé. Nous ne nous concentrons donc pas directement sur la réalisation des tâches mais préférons prendre le temps d'entrechoquer nos idées afin de produire un travail plus efficace.

Nous pensons qu'en faisant cela, nous gagnons finalement du temps en ne repassant pas plusieurs fois sur notre travail.

En dehors des séances

- Pour organiser le travail à faire pour chaque semaine, il a été décidé d'utiliser **Trello**.
- Nous travaillons sur un dépôt GitHub à cette adresse.
- Pour faciliter les discussions moins formelles entre nous, nous utilisons un réseau de chat en ligne (Facebook Messenger).
- La complétion du document de suivi pour le groupe (document Excel) est réalisée avec Google Drive, qui offre un système de modifications collaboratives.
- Google Drive est utilisé pour partager les diverses ressources annexes.

Retroplanning du rendu 1

Retroplanning de Laurine Drodzinski

Travail effectué

Au cours de cette première itération j'ai eu l'occasion de travailler sur les scénarios. Après en avoir réalisé, j'ai pu retravailler le reste des scénarios afin qu'ils soient cohérents tant au niveau des acteurs, de la présentation, du vocabulaire etc.
J'ai également travaillé sur le tableau regroupant divers termes utilisés au cours de cette itération. Ensuite, j'ai décrit les différents acteurs de l'application en indiquant leurs possibilités d'actions ainsi que quelques contraintes les concernant. Par ailleurs, j'ai pu accompagner Anthony sur les diagrammes de cas d'utilisation. Puis, pour finir, j'ai corrigé les fautes d'orthographes des différents fichiers du git ainsi que remis en forme divers tableaux et dossiers.

Difficultés rencontrées

J'ai rencontré quelques problèmes de compréhension au début de projet concernant l'organisation du projet. En effet nous avions éparpillé des données sur diverses plateformes. Cependant ce problème fut très rapidement réglé car dès la deuxième séance nous nous sommes réorganisés et nous avons regroupés les informations sur git.Le second souci a été une confusion à propos de certaines informations du cours. Notamment au niveau des diagrammes de cas d'utilisation et des différentes flèches à utiliser. Afin de régler ce problème nous nous sommes rapprochés de Cédric Dumoulin qui nous a réexpliqué leur sens.

Retroplanning de BARCHID Sami

Tâches faites

- Élaboration des diagrammes de classe
- Élaboration du diagramme de package
- Élaboration de scénarios concrets

Méthode de travail

Ma manière de travailler personnelle dans le projet a été de profiter des TPs pour mixer un maximum des jugements de chacun sur tous les travaux à réaliser (diagrammes, scénarios, etc) afin d'être directement fixé sur la façon de remplir mes tâches. Personnellement, je pense que les deux heures de TPs par semaine sont plus utiles si on les utilise pour entrechoquer nos idées que réellement écrire quelque chose qui ne satisfait pas tout le monde.

Difficultés

La principale difficulté vue pour ce projet est l'absence totale de client à qui poser des questions, laissant les ambiguïtés de "l'appel d'offre" à notre appréhension personnelle.

Une difficulté, plus secondaire, est la différence de compréhension des standards UML entre tous (élèves, professeurs et manuels compris), qui souligne bien l'importance d'établir des conventions entre les différents membres de l'équipe. Cette difficulté est surmontée grâce aux différents débats entre nous lorsqu'une ambiguïté était rencontrée.

Retroplanning de Zoé Canoen

Tâches faites

- Mise en place des différents outils lors du début du projet
- Gestion de la répartition des charges au sein de l'équipe
- Synthèse des séances
- Rendu du lot 1 (format markdown et pdf)
- Premières maquettes de l'application (format fixe)

Méthode de travail

Pour moi il est important d'avoir une vision globale du sujet. Cela permet de gérer au mieux les délais. La répartition des tâches et la confiance en chacun des membres de l'équipe est primordiale. J'aime apporter des nouvelles idées et je pense que la communication dans l'équipe est importante. Au début je parlais à chacun, mais maintenant nous faisons des réunions tous ensemble pour une mise en commun.

Difficultés

Je pense que nous avançons très bien, et que nous faisons face à tous les évènements sans trop de difficultés. La capacité d'adaptation du groupe est très importante. Je suis bien consciente que ce n'est pas le seul projet sur lequel nous devons travailler, et que parfois certaines personnes dans le groupe n'ont pas le temps de travailler GL. Mais notre groupe se rééquilibre toujours pour continuer à avancer. Si une personne travaille beaucoup plus que les autres, je veille à ce que ces tâches soient minimisées. Et à l'inverse, si une personne n'a pas le temps de réaliser une tâche, je m'arrange pour que la tâche soit redistribuée si les délais sont trop courts. Nous nous faisons confiance sur le travail effectué, chaque personne travaille dans la mesure de son temps disponible, et finit toujours par rattraper son retard quand elle le peut. Nous pouvons tous compter sur les autres si besoin, et donc avancer à notre rythme.

Ressenti

GL est réellement une matière fortement intéressante. Au début nous devions trouver notre rythme de croisière et je crois que nous l'avons trouvé. Les réunions de groupe lors des tps sont vraiment très utiles et nous avançons beaucoup. Chacun peut ainsi avoir une vision globale du projet.

Retroplanning of Noura

Done tasks

- Description of case of use Package diagram elaboration of one version

Work's Method

For me, working within a team is a greatfull experience ; during the "TP", I try usually to understand the ideas of my colleagues and mainly try to explain mine. The most important thing that I try to get it after each meeting is to gather what Mr Michaël Launay tells us and what we sugggess to do as a srategy. Even between us, we have really many points of view, so that to make a collective decision is really so interesting.

Difficulties

If I will speak about difficulties , i will start by the difference of language with our responsible and with my colleagues. After each meetiig , and when I come back to home I try to remember the most important suggession, sentences and even words and I start "traduction service", it takes much time. Besides, one of the most things that was difficult for me, is that our project seems to be clear in his glabal view:"take an order inside a restaurant", but some details about at which level we are limited to designing this application ; should we just take into consideration "the service give by the waiter" or may be be so selective when we had to choose the performance evaluation criterion (good service/client satisfaction/price/time taken by the application), make it a bit complicated.

Retroplanning de Anthony SLIMANI

Tâches faites

- Élaboration de scénarios concrets & un alternatif
- Élaboration des diagrammes des cas d'utilisations
- Participation à l'élaboration du diagramme de classes

Difficultés

On a pu constater un début difficile au niveau de l'organisation mais nous avons pu nous harmoniser au fil des semaines. Nous avons des idées intéressantes mais parfois qui s'écarte un peu trop du sujet principal ou encore n'a pas la nécessité d'être élaboré pour cette première itération. Nous avons eu quelques difficultés sur les standards UML en constatant diverses différences avec le cours, les livres de référence ou encore des documents sur Internet.

Ressenti

Cette itération m'a montré que toute l'équipe était motivée pour ce début de projet. J'espère que cette motivation va perdurer durant les prochaines itérations. Une bonne cohésion d'équipe s'est formée durant ces quelques semaines. Tout le monde essaye de s'adapter aux autres, à son rythme. J'aime beaucoup la manière que chacun appréhende le sujet, de voir la diversité de nos idées et l'ouverture d'esprit de chacun lors des échanges de nos idées pour parvenir à une solution qui correspond le plus aux besoins du client.

Retroplanning du rendu 2

Retroplanning de Laurine Drodzinski

Tâches faites

- Descriptions textuelles :
    - Ajout de préparation dans une commande
    - Suivre une commande
    - Cuisiner une préparation
    - Préparation indisponible (cuisine)
- Diagrammes de séquences système :
    - Servir une commande
    - Prendre en charge une préparation
    - Cuisiner une préparation
    - Préparation indisponible (cuisine)
- Diagrammes de séquences détaillés:
    - Prendre en charge une préparation
    - Cuisiner une préparation
- Rédaction du rapport de la v2
- Correction des fautes

Méthode de travail

Pour ce rendu j'ai essayé de réaliser le travail qui m'était assigné en temps et en heure. J'ai rapidement terminé les descriptions textuelles afin de pouvoir me concentrer sur d'autres problématiques. En effet les diagrammes de séquences m'ont tout d'abord fait un peu peur et je ne savais pas où donner de la tête. Cependant grâce aux autres membres du projet qui ont pris le temps de me réexpliquer le fonctionnement de ces diagrammes, j'ai réalisé que ce n'était pas si difficile que ça.

Ensuite, j'ai beaucoup apprécié les réunions que nous avons effectué à chaque cours. Elles m'ont permis de mieux identifier les différents problèmes ainsi que les différentes tâches qui m'étaient destinées. Nous nous sommes correctement répartis le travail à faire et cela a permis à chacun d'apporter sa pierre au projet. De plus, afin de ne pas perdre le fil du projet et de réaliser des documents cohérents. Nous avons chargé des membres du groupe de vérifier la cohérence de chaque document.

Difficultés

Comme dit ci-dessus, le majeur problème que j'ai rencontré était les diagrammes de séquences (détaillés). Mais grâce aux membres du groupe j'ai tout de même pu réaliser ce travail.

Ressenti

Je suis fière de notre groupe. Chacun participe et il est agréable de travailler ensemble malgré les moments de stress lors de l'approche du rendu. Chacun est là pour aider l'autre et nous savons que nous pouvons compter sur chacun d'entre nous.

Retroplanning de Sami Barchid

Tâches faites

- Descriptions textuelles :
    - Ajouter une préparation au menu
    - Créer une préparation
    - Modifier une préparation
    - Retirer une préparation du menu
- Diagrammes de séquences système :
    - Prise d'une commande - aide
    - Retirer une préparation du menu (sur papyrus)
    - Modifier une préparation (sur papyrus)
    - Supprimer une préparation (sur Papyrus)
    - Créer une préparation
- Diagrammes de séquences détaillés:
    - Retirer une préparation du menu
    - Créer une préparation
    - Prise d'une commande - aide
    - Ajouter une préparation au menu
- Mise à jour du diagrammes des cas d'utilisation
- Diagramme de packages + texte explicatif
- Mise à jour des diagrammes de classes
- Glossaire d'ingénierie des besoins
- Mise en place d'une integration continue pour la mise en ligne de la maquette interactive

Méthode de travail

Pour ce rendu, j'ai aidé mes camarades en leur décrivant la manière de faire des diagrammes de séquence et en essayant de leur référer un maximum nos cours théoriques.

Difficultés

La grosse difficulté de ce rendu a été la quantité de travail à fournir. En effet, beaucoup des choses demandées prenaient du temps (diagrammes de séquence).

L'autre difficulté a été le fait que le diagramme de classe ne pouvait être fait avant que les diagrammes de séquence ne soient tous terminés, à cause du fait que ce sont les diagrammes de séquence qui décident des noms de méthodes, des classes, etc qui seront présents dans le diagramme de classe. De ce fait, cette partie a été compliquée à faire.

Un autre problème est le fait qu'un de nos membres, Noura Mares, a été malheureusement forcée de prendre des jours de repos dû à un gros problème de santé. Nous avons donc fait en sorte de nous répartir son travail tandis qu'elle restait au courant de ce que nous faisions.

Ressenti

- L'équipe est très soudée et est très motivée dans ce projet.

- Bonne ambiance

- Je trouve dommage le fait que nous devons produire uniquement des diagrammes de séquence, en une si grande quantité. En effet, je ne trouve pas que réaliser des diagrammes de séquence soit si judicieux étant donné que c'est ce type de diagramme qui va demander le plus de temps à une équipe de développement. À chaque changement dans notre processus, il faut perdre du temps à maintenir nos diagrammes de séquence pour éviter qu'ils deviennent obsolètes. De plus, je trouve que le niveau de détail très élevé de ce genre de diagramme rend la compréhension beaucoup moins facile pour un nouveau développeur qui arriverait sur notre projet. Établir une séquence générale pour notre architecture aurait été, selon moi, bien plus intéressant, permettant de comprendre comment elle s'articulait.

- Je trouve dommage aussi que nous n'avions pas eu à créer des diagrammes d'états dans ce travail. En effet, ils aident souvent bien plus à la compréhension de la logique business du système que des diagrammes de séquence.

Retroplanning de Zoé Canoen

Tâches faites

- Gestion de la répartition des charges au sein de l'équipe
- Synthèse des séances
- Rendu du lot 1 (format markdown et pdf)
- Premier aperçu de la maquette interactive
- Diagramme de séquences détaillé de Ajouter une Préparation
- Description textuelle de :
    - Prendre en charge une préparations,
    - Ajouter une préparation à la commande,
    - Préparation indisponible (côté cuisine),
    - Prendre une commande
- recherche sur l'architecture MVC / 3 couches / DAO
- Corrections et validation des travaux effectués

Méthode de travail

Pour moi il est important d'avoir une vision globale du projet. Cela permet de gérer au mieux les délais. La répartition des tâches et la confiance en chacun des membres de l'équipe est primordiale. J'aime apporter des nouvelles idées et je pense que la communication dans l'équipe est importante. Au début je parlais à chacun, mais maintenant nous faisons des réunions tous ensemble pour une mise en commun.

Sur la fin, nous avions beaucoup de travail à effectuer en peu de temps car les autres cours ont empiétés sur notre travail de GL. Nous avons tous travaillé en équipe en communiquant sur chaque modification. Je m'occupais de valider au fil de l'eau chaque modification.

Nous avons dû travailler à distance, et nous communiquions par Messenger sur les différentes répartitions des tâches.

Difficultés

La charge de travail avait été mal évalué, car les diagrammes de séquences détaillés ont prit beaucoup de temps.

Ressenti

Je suis contente de notre équipe. Nous faisons un bon travail sérieux, et nous sommes critiques sur ce que nous produisons. Nous pouvons toujours améliorer, voilà notre devise.

Retroplanning de Anthony Slimani

Tâches faites

- Descriptions textuelles :
    - Paiement
    - Imprimer l'addition d'une commande
    - Rétification de divers descriptions
- Diagrammes de séquences système :
    - Paiement
    - Imprimer addition
    - Prise d'une commande avec Sami
    - Ajouter une préparation - Service avec Zoé
- Diagrammes de séquences détaillés:
    - Paiement
    - Imprimer addition
    - Prise d'une commande avec Sami
    - Ajouter une préparation - Service avec Zoé
- Mise à jours des diagrammes de cas d'utilisation

Méthode de travail

Nous avons repris les même méthodes de travails que l'itération précédente. Durant chaque séance, nous avons réalisés des réunions de groupe pour permettre d'expliquer les problèmes rencontrés et le travail effectué. Chacun avait pour mission d'au moins effectuer deux descriptions textuelles, deux diagrammes de séquences et deux diagrammes de séquences détaillés, chose qui a été effectué avec succès.

Difficultés

J'ai eu beaucoup de mal à concevoir les diagrammes de séquences détaillés. J'ai pu être confu à de nombreuses reprises de la différences des standards d'UML que j'ai appris les années précédentes et de cette année entraînant parfois l'impossibilité de trancher entre mes "anciennes" et "nouvelles connaissances". Nous avons eu du mal à savoir qui était en train de faire quoi durant la fin de cette itération. Le suivi du projet et le Trello n'était pas mis à jour. De ce fait, il y a pu avoir deux membres du groupes qui travailler sur la même chose sans le savoir. Chose à améliorer durant la prochaine itération.

Ressenti

- Bon travail de la part de notre groupe
- Ambiance de groupe convivial et agréable
- Un moment de stress est venu lors de la deadline

Synthèse des séances

Synthèse de séance 01

Semaine du 10 Septembre

15 minutes de lecture du sujet 15 minutes pour poser le problème 2h15 pour faire les scénarios ensemble 15 minutes pour savoir qui fera quoi hors séance, et comment nous allons travailler (github, trello..)

Méthode de travail

En séance

Après une rapide lecture du sujet, la majorité des idées concordaient entre nous au niveau du choix des scénarios. Il a été décidé de répartir alors les différents scénarios trouvés entre nous. Zoé CANOEN avait pour rôle de noter nos diverses réflexions sur papier et de répartir les autres membres du groupe sur les scénarios à faire.

En parallèle, lorsque l'un de nous avait une question, nous essayions d'y répondre ensemble pour clarifier au mieux le scénario et trouver des problèmes auxquels nous n'aurions pas pensé au départ (voir les points discutés au chapitre référé).

L'écriture des scénarios

L'écriture des scénarios a été faite par chacun, mais nous avons tout regroupé vers le milieu de la séance pour voir ce que les autres avaient fait. Nous nous sommes aperçus qu'il y avait des points à discuter. En effet, le sujet ayant des zones d'ombre, nous devions pouvoir en parler avec le client. Mais puisque nous ne pouvions pas parler au client, nous avons allions devoir trancher, maintenant ou plus tard.

Il est important de noter que chaque scénario a été créé atomiquement, car le sujet rend assez bien compte des fonctionnalités de l'application pour chaque acteur. C'est pourquoi les scénarios se limitent au point de vue d'un acteur.

La caisse

Nous ne sommes pas sûrs que l'API doivent gérer le paiement ou si elle doit juste servir d'affichage de l'addition. Nous nous préoccuperons de cela plus tard. Pour le moment elle ne fera qu'afficher la note.

**Le visuel du client / serveur / cuisinier / patron **

L'application sera à priori fournie soit sous 4 applications différentes, soit sur une seule avec 4 visuels différents. En effet, le client ne doit pas pouvoir modifier les plats, avoir accès à l'addition, et les cuisiniers ne peuvent pas modifier une commande.

Ce que peut faire le client :

- Remplir le numéro de sa table
- Prendre une commande
- Suivre la commande

Ce que peut faire le serveur :

- Gérer l'ensemble des tables du restaurant (ajouter ou enlever des clients d'une table, ajouter ou supprimer une table)
- Créer une commande pour un client
- Modifier une commande pour un client
- Gérer l'addition d'une table pour régler la commande

Ce que peut faire un cuisinier :

- Voir la liste des plats qui le concerne
- Réserver un plat qu'il est en train de faire, pour ne pas que d'autres personnes le prenne
- Informer le client qu'un plat est en train d'être fait, ou est fini
- Donner une information sur le fait qu'un plat est en rupture de stock

La sélection des plats par les cuisiniers

Nous avons commencé à voir comment serait l'affichage des plats pour les cuisiniers.

Hors séance

- Pour organiser le travail à faire pour chaque semaine, il a été décidé d'utiliser **Trello**.
- Les diagrammes seront réalisés sur le logiciel "**draw.io**", permettant de créer des diagrammes collaboratifs (utilise des fichiers XML).
- Le dépôt Git de l'école étant encore indisponible pour la majorité des alternants suite à des retards administratifs, un deuxième `remote` sur **Github** a été installé temporairement à cette adresse.
- Pour faciliter les discussions moins formelles entre nous, nous utilisons un réseau de chat en ligne.
- Les scénarios alternatifs seront à faire plus tard.

Synthèse de séance 02

Semaine du 18 Septembre

- (1h50 de TD sur le sujet TinCar)
- 1h10 de TP sur notre projet

Méthode de travail

Avant la séance

Anthony s'est renseigné sur le fonctionnement d'une cuisine de restaurant.

- Liens :
- http://technoresto.org/environnement-technologique/lorganisation-du-travail-au-restaurant/
- http://technoresto.org/tr/brigade/

En séance

Nous avons chacun écrit sur un Google Doc le glossaire, pour mettre en commun plus facilement ce que nous avions. Nous avons pu redéfinir les termes que nous allions utiliser pour l'application. Ceci a permis de mettre en évidence les différents futurs **cas d'utilisation**. Nous avons donc pu prendre en main le logiciel **Eclipse Papyrus** qui servira aux diagrammes d'utilisation.

Dans un premier temps, le glossaire a été rempli à la manière d'un brainstorming, permettant par la suite de débattre ensemble pour définir les termes que nous allions utiliser pour la suite du projet.

Nous avons également déjà pensé à quelques classes pour élaborer une première ébauche d'un diagramme de classes.

Le début du glossaire

Nous avons commencé un glossaire, en trouvant les mots importants qu'il fallait définir, ainsi que les cas d'utilisation qu'il allait falloir faire.

Après la séance

- Les diagrammes ne sont plus réalisés sur "**draw.io**" mais **Eclipse Papyrus**. Chacun va réaliser des diagrammes de cas d'utilisation, qui se rapprochent des scénarios associés.
- ***Sami*** et ***Anthony*** vont commencer les diagrammes de classes
- ***Laurine*** et ***Noura*** vont répertorier toutes les spécifications de chaque acteur de l'application.
- ***Zoé*** va se renseigner sur comment fonctionne techniquement l'envoie des notifications sur une application et trouver tous les cas d'utilisation à faire.

Justification des choix

Diagrammes de classes

- Tous les employés du restaurant auront, à l'avenir, plusieurs attributs similaires. C'est pourquoi il est décidé de faire hériter tout membre du personnel du restaurant de la classe abstraite : **MembrePersonnel**.
- Les préparateurs du restaurant peuvent avoir plusieurs rôles différents : barman, glacier, cuisinier etc. Pour l'instant, chaque rôle de préparateur est une classe héritant de la classe **Préparateur**. Par la suite, il pourrait s'avérer qu'une distinction en classes de tous les préparateurs ne soit pas utile. C'est pourquoi ce choix est temporaire.

Synthèse de séance 03

Semaine du 25 Septembre

- 2h00 de TD
- 1h00 de TP

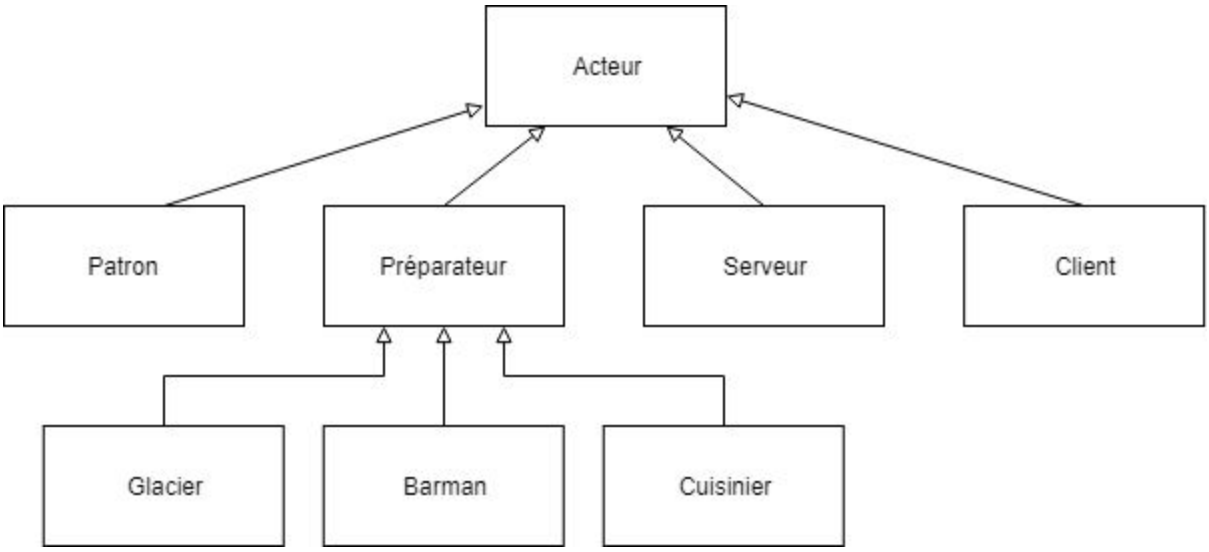
Méthode de travail

En séance

Nous avons fait un bilan de tout ce qui a été fait. Et nous nous sommes séparés en deux groupes. Sami et Noura se sont occupés des diagrammes de classes. Laurine et Anthony se sont occupés de finir les diagrammes de CU.

Bilan de ce qui a été fait

Lors des précédentes séances nous avions pensé à faire cela :



Mais ! Nous ne nous sommes pas rendu compte que les classes comme "glacier" ou "barman" ne sont que des instances de la classe. Donc nous allons simplement avoir des "préparateurs", qui seront la classe qui s'occupera de préparer les commandes en cuisine. Nous avons donc 3 classes membres du restaurant :

- Le patron, qui a la gestion administrative de l'application. C'est lui qui peut gérer le menu.

- Le serveur, qui s'occupe de prendre la commande au client. Il pourra modifier la commande du client.
- Les préparateurs, qui s'occuperont de valider les plats effectués, et indiquer les plats indisponibles.

Il faut savoir que nous allons vouloir faire une interface graphique différente pour chaque partie du restaurant. En effet, les besoins de toutes les équipes sont différents. Il faut que l'application soit pratique pour chaque personne. Des maquettes ont déjà été discutées lors de la première séance.

Diagrammes de CU

Anthony s'est occupé du diagramme de CU, avec tous les CU de notre projet

Diagrammes de classe

Sami s'est occupé de faire les diagrammes pour toutes les classes de notre projet. Il a été aidé par Noura qui s'est occupé de remplir les classes.

Synthèse de séance 04

Semaine du 2 Octobre

- 1h30 de TD
- 1h30 de TP

Méthode de travail

En séance

Les séances servent de plus en plus à faire le bilan des semaines tous ensemble. Cela est mieux comme ça, nous avançons tous ensemble. De plus, nous avons pu voir le cas de Noura, qui commence réellement à s'intégrer au groupe. Son arrivée a été compliquée pour elle car il y avait la barrière de la langue. Avec ce fonctionnement, cela lui permet de suivre tout ce qu'on fait et de réagir et de proposer ses idées. Nous arrêtons de nous séparer en petits groupes pour qu'elle puisse au maximum avoir une vision globale du sujet.

De plus, il est maintenant plus facile de garder les idées et remarques de chacun.

Choses à faire pour le rendu de VENDREDI 05 SEPTEMBRE A 20H

Voici toutes les remarques que nous avons notées rapidement :

Sami

- DIAGRAMME DE CLASSES :
    - Rajouter les packages sur le diagramme de classes
    - Modifier des flèches droites
- DIAGRAMME DE PACKAGES :
    - Rajouter "prise en charge", comme l'a fait Noura.
- DES PETITS DIAGRAMMES DE CLASSES :
    - Des petites images / diagrammes de classes : découpage des packages (en plus)

Anthony

- DIAGRAMMES CU :
    - Modifier les titres de type "Service", "Cuisine" en "Application"
    - Supprimer les notifications des diagrammes **OU** modifier en commentaire
    - Ajouter une description "notification" dans les descriptions des CU, si ce n'est pas fait.
    - CUISINE : supprimer notification, mettre un commentaire pour l'include ("il faut avoir fini la commande")
    - Générer l'addition PROBLEME d'include (à supprimer), rajouter la fonctionnalité "Générer l'addition"

Noura

- Rapport_final.md
    - Add this in the glossary :
        - "service" (for the waiters)
        - "cuisine" (for the cookers)

Zoé

- MAQUETTES :
    - ajouter les acteurs cuisiniers dans la cuisine
    - modifier l'interface du service. Faire une liste de table plutôt qu'une sorte de modélisation.
    - Modifier le texte : "voir les avancements" => "voir les commandes"
- DOCUMENT PDF
    - Mise en forme du Markdown sous Word avec un joli Template

Laurine

- changer les scénarios : de plat en préparation
- Mettre à jour le tableau suite aux modifications faites
- Vérifier chaque fichier pour les accents et les fautes d'orthographe

Discussions

Une discussion a commencé à émerger à la fin de la séance. Notre application est très peu permissive. Nous avons besoin d'avoir le point de vue du client. En effet, dans cette conception, les serveurs n'ont pas accès à la cuisine. Nous allons chercher, dans le futur, à la rendre plus permissive. Mais il ne faut peut-être pas arriver au cas extrême où nous n'avons aucune spécification sur serveur / préparateur / patron.

Définition des rôles

Au fur et à mesure, chacun a obtenu un rôle assez précis. Cela est visible dans le récapitulatif de ce que chacun doit faire. **Anthony** et **Sami** sont devenus nos experts de Papyrus. Nous avons choisi de fonctionner ainsi pour éviter les conflits entre tous les ordinateurs.

Anthony a également la vision de faisabilité technique. **Zoé** aime vouloir ajouter des nouvelles fonctionnalités. **Anthony** juge régulièrement ce qui est faisable rapidement ou ce qui va prendre du temps.

Laurine s'occupe de la vision avec la MOA et la livraison à faire. Et elle s'occupe de la faisabilité en termes de temps. Elle n'a pas de rôle défini, ce qui nous permet d'avoir toujours de l'aide si besoin. Elle a la capacité de s'adapter à toute situation si on lui assigne une tâche, ou de prendre les tâches elle même si elle n'a rien à faire.

Sami a le rôle d'expert technique, et est plus notre architecte technique. C'est lui qui produit les différents documents techniques, avec les fluxs de données. Et c'est lui qui est expert pour savoir quelle classe communique avec quoi.

Noura s'intègre de plus en plus. Les choses qui lui sont demandées peuvent paraître faible. Cependant, il faut savoir qu'elle traduit tout ce que nous faisons. Car nous produisons tous nos documents en français. C'est pourquoi, nous essayerons d'écrire au maximum en anglais pour l'aider. Son travail lui prend du temps en plus des cours, donc elle a eu beaucoup de mal au début, mais elle reste toujours active en séance.

Zoé a la vision d'équipe pour que chacun puisse faire quelque chose. Elle se fait aider de **Sami** pour savoir les tâches faisables pour **Noura** ou **Laurine** en fonction de l'aide dont il a besoin. Elle doit limiter le travail de Sami et Anthony qui sont moteurs de l'équipe. Donc elle doit dispatcher leur travail sur **Noura** et **Laurine**. Elle s'occupe aussi du rendu, d'écrire les synthèses des séances, et de regrouper tout ce qui a été fait. Elle doit être à l'écoute de tout le monde.

Synthèse de séance 05

Semaine du 15 Octobre

- 1h30 de TD
- 1h30 de TP

Méthode de travail

En séance

Lors de ce cours nous avons présenté le projet et les documents réalisés aux professeur. C'est Noura qui a présenté le projet et elle s'en est très bien sortie. Grâce à elle, le professeur a pu comprendre l'objectif de notre application ainsi que les étapes de conception par lesquelles nous sommes passés.

Synthèse de séance 06

Semaine du 22 Octobre

- 1h30 de TD
- 1h30 de TP

Méthode de travail

En séance

Nous avons du faire un point sur ce qui a été fait. Laurine et Sami ont travaillé sur les scénarios abstraits. Ils ont fait en particulier les descriptions textuelles de ceux ci. Zoé a commencé la maquette interactive avec les maquettes effectuées la précédente fois.

Voici les descriptions textuelles effectuées :

Le menu

* Ajouter une préparation au menu

* Créer une préparation

* Modifier une préparation

* Retirer une préparation au menu

* Supprimer une préparation

Le service

* Ajouter une préparation à la commande

* Plat indisponible

* Suivre l'avancement d'une commande

* Servir une partie d'une commande

La cuisine

* Cuisiner une préparation

* Préparation indisponible

Choses à faire pour le rendu de VENDREDI 09 NOVEMBRE A 20H

Sami

* Réarranger les descrption textuelles effectuée en utilisant le même template que Laurine

* Finir les scénarios abstraits du menu

C'est à dire faire les Diagramme de Séquence Système (DSS) pour chaque CU et Diagramme de séquence détaillé pour chaque DSS.

Anthony

* Scénario abstrait de : Générer l'addition

* Scénario abstrait de : Valider le paiement d'une commande

Anthony est face à un problème et a besoin d'avoir l'avis du client sur l'addition. En effet, la

Noura

* Scénario abstrait de : Prise d'une commande (Serveur)

* Scénario abstrait de : Proposer le menu du restaurant (Serveur)

* Retranscrire au propre les hiérarchies des CUs (attention à ne pas confondre CUs et scénarios).

Zoé

* Rangement des fichiers

* Synthèses des séances

* Description de l'architecture

* Ajout du nouveau scénario abstrait de : Préparation indisponible (Service)

* Scénario abstrait de : Prendre en charge une préparation (Cuisine)

Laurine

* Finir les scénarios abstraits :

  * Ajouter une préparation à la commande

  * Plat indisponible

  * Suivre l'avancement d'une commande

  * Servir une partie d'une commande

C'est à dire faire les Diagramme de Séquence Système (DSS) pour chaque CU et Diagramme de séquence détaillé pour chaque DSS.

Rapport final du rendu 3

Cette partie concerne les parties qu'il nous manque pour le rapport final du rendu 3.

Etude du modèle existant (étude du besoin)

Il est important d'expliciter mieux notre étude du besoin.

Maquette interactive

Il faudra terminer la maquette interactive (ainsi que les évènements)

Diaporama final

Nous avions déjà fait un diaporama pour la présentation du rendu 1. Il suffira de reprendre celui-ci.

Synthèse de séance 07

Semaine du 05 Novembre

- 1h30 de TD
- 1h30 de TP

Méthode de travail

En séance

Lors de cette séance nous avons effectué une petite réunion afin de résumer le travail restant à faire pour le rendu ainsi qu'assigner les tâches restantes.

Nous en avons conclu que le gros du travail qui restait à réaliser était les différents diagrammes de séquences (système et détaillés).

C'est donc ce sur quoi nous avons travaillé durant la séance.

Tableau TYPES-CONCEPTS

SUJET	TYPE	CONCEPT
Bob	Acteur / (Classe dans le futur)	Client
Alice	Acteur / Classe	Serveur du restaurant
Albert	Acteur / Classe	Patron du restaurant
Roger	Acteur / Classe	Préparateur / Cuisinier
L'équipe de cuisine	Acteur**s**	Ensemble des préparateurs
Table	Classe	Table du client
Menu	Classe	Carte du restaurant
Préparation	Classe	Entrée / Plat / Dessert / Boisson
Commande	Classe	Ensemble de toutes les préparations, commandées par le client
Partie de commande	Donnée / Classe	Ensemble de boissons, entrées, plats ou desserts. Donnée de la commande
Addition	Donnée (Décimal)	Donnée de la commande
Numéro de table	Donnée (Entier)	Donnée de la table
Glacier	Objet	Préparateur
Barman	Objet	Préparateur
Proposer le menu	Action / Objectif / Cas d'utilisation	Le serveur propose le menu au client
Ajouter une préparation	Action / Objectif / Cas d'utilisation	Le serveur ajoute une ou des préparations dans une commande en cours, sous demande du client
Saisir le numéro de table	Action / Cas d'utilisation	Le serveur saisi le numéro d'une table pour référencer une commande

Servir une partie de commande	Action / Cas d'utilisation	Le serveur amène au client une partie de la commande et indique que la livraison va être réalisée
Valider le paiement	Action / Objectif / Cas d'utilisation	Le serveur indique sur l'application que l'addition a bien été réglée
Suivre l'avancement d'une commande	Action / Objectif / Cas d'utilisation	Les serveurs peuvent suivre l'état et l'avancement d'une commande
Créer/Modifier le menu	Action / Objectif / Cas d'utilisation	Le patron apporte des modifications au menu ou en crée un nouveau
Valider le menu	Action / Cas d'utilisation	Le patron valide le menu après avoir effectué des modifications
Prendre en charge une préparation	Action / Objectif / Cas d'utilisation	Le préparateur sélectionne la préparation dont il souhaite s'occuper
Valider une préparation après l'avoir cuisiné	Action / Cas d'utilisation	Le préparateur valide la préparation une fois qu'il a terminé de la cuisiner
Rendre une préparation indisponible	Action / Cas d'utilisation	Le préparateur indique qu'une préparation est indisponible si les ingrédients ne sont pas suffisants
Demander l'addition	Action / Objectif	Le client demande l'addition à l'un des serveurs
Régler l'addition	Action	Le client paie l'addition quand il souhaite partir
Notification	Objet de médiation	Message destiné aux serveurs ou aux préparateurs
Affiche l'addition	Objet de médiation	Message destiné au client
Tablette	Support	Support utilisé par les membres du restaurant
OneMinute	Application	Application utilisée par les membres du restaurant