{ API:REST'O }

L'equipe :
Fabien DELBECQUE
Florian POURRY
Guillaume LEPRETRE
Thomas MOREAU
Emeline DIEU

Version : 3

Date de redaction : 06/11/2018

Table des matières

1. Introduction

L'API Resto est un logiciel permettant la synchronisation de toute l'équipe de préparation d'un restaurant. Nous allons définir les besoins d'un client à partir de divers documents tels que des scripts, des cas d'utilisation, un diagramme de classes etc.

1.1 Pourquoi ce système ?

La principale problématique est de faire collaborer efficacement différentes équipes aux rôles bien distincts autour d'un mêmes objectifs : servir les clients le plus rapidement possible dans un restaurant.

1.2 Portée du système

Le logiciel est dans un premier temps destiné au patron et aux employés du restaurant. Il pourra par la suite être utilisé par les clients.

1.3 Objectifs et critères de succès pour le projet

L'objectif est la réalisation d'un ensemble de documents permettant de répondre à tous les besoins du client.

2. Glossaires

2.1. Glossaire métier

Ce glossaire définit les termes utilisés par les utilisateurs, à savoir le personnel du restaurant.

Terme	Définition
Addition	Ensemble des dépenses d'une table
Bar	Lieu du restaurant où l'on réalise l'ensemble des boissons
Barman	Membre du personnel dans le bar réalisant l'ensemble des boissons
Client	Personne qui achète ou requiert des services moyennant rétribution
Cuisine	Lieu du restaurant où l'on réalise l'ensemble des entrées/plats
Cuisinier	Membre du personnel présent dans la cuisine réalisant l'ensemble des entrées/plats
Glacier	Membre du personnel présent dans la cuisine réalisant l'ensemble des desserts
Libérer table	Quand un serveur a nettoyé et dressé une table qui avait été assignée
Table	Lieu où sont placés les clients
Table libre	Quand une table est dressée et qu'il n'y a pas de client qui l'occupe
Serveur	Membre du personnel qui sert les clients
Préparateur	Membre du personnel réalisant les commandes du restaurant
Membre du Personnel	Personne travaillant dans le restaurant
Commande	Une commande est une demande d'un client sur un produit qui sera préparé et servis par les membres du personnel
Produit	Un produit est consommation demandée par les clients

Figure 1 : Glossaire métier

2.2. Glossaire technique

Ce glossaire définit les termes utilisés par l'équipe de développeurs. Les termes décrits permettent de comprendre le fonctionnement de l'application.

Terme	Définition
Addition	Opération consistant à réunir l'ensemble des dépenses d'une table
Bar	Lieu de réception et de préparation des boissons
Barman	Préparateur dans le bar réalisant l'ensemble des boissons
Client	Personne qui occupe une table, consomme des préparations et paye l'addition
Cuisine	Lieu de réception et de préparation des entrées/plats
Cuisinier	Préparateur présent dans la cuisine réalisant l'ensemble des entrées/plats
Glacier	Préparateur présent dans la cuisine réalisant l'ensemble des desserts
Libérer table	C'est lorsque l'on assigne la table au statut vide
Préparation	Un produit préparé par un préparateur
Table	Objet permettant de prendre commande et d'appeler un serveur
Table libre	Une table ayant le statut vide
Serveur	Personne qui s'occupe de servir les clients et transmet les informations aux préparateurs
Préparateur	Membre du Personnel réalisant l'ensemble des commandes
Membre du Personnel	Objet permettant d'étendre les Préparateurs et les Serveurs
Commande	Objet composé de produits, ayant un Etat d'avancement, assignée à une Table, réalisée par un Preparateur et servie aux clients par le Serveur
Produit	Objet permettant d'étendre les différents plats et boissons

Figure 2 : Glossaire technique

3. Système proposé

3.1 Modèles

3.1.1 Scénarios concrets

Nom du cas : ***Installation du Client.***
But : Trouver une table et y installer le client.
Acteur Principal : Le serveur.
Date de création : 09/10/2018.
Nom des responsables : Fabien et Florian.
Version : 1
Pré-condition : Le serveur a accès à l'application.
Déclenchement : Un client (seul ou en groupe) se présente.
Post-condition : Le client (seul ou en groupe) est installé à une table.
Scénario nominal :

1. Le serveur accueille le client.
2. Le serveur consulte l'application pour savoir si une table est libre.
3. Le serveur installe les clients à la table.
4. Le serveur indique à l'application que la table est occupée.

Scénario Alternatif :
2. a Aucune table libre
  2.a.1 : Le serveur s'aperçoit que toute les tables sont occupées.
  2.a.2 : Le serveur indique aux clients un temps d'attente estimé par l'application.
  2.a.3 : Après ce temps d'attente, retour à 3

Nom du cas : ***Prise de Commande par le serveur.***
But : Prendre la commande du client.
Acteur Principal : Le serveur.
Date de création : 09/10/2018.
Nom des responsables : Guillaume.
Version : 1.0
Pré-condition : Le serveur a accès à l'application.
Déclenchement : Un client (seul ou en groupe) est installé sur une table.
Post-condition : Le préparateur est notifié de la commande.
Scénario nominal :

1. Le serveur demande le choix du client
2. Le client choisit sa commande
3. Le serveur saisit la commande sur l'application
4. Le serveur valide la commande

Scénario Alternatif :
3. a Le client change d'avis et souhaite autre chose
3.a.1 : Le serveur modifie la commande du client
3.a.2 : Après ce temps d'attente, retour à 4

Nom du cas : ***Prise de Commande par le client.***
But : Prendre la commande du client.
Acteur Principal : Le client.
Date de création : 09/10/2018.
Nom des responsables : Guillaume.
Version : 1.0
Pré-condition : Le client a accès à l'application.
Déclenchement : Un client (seul ou en groupe) est installé sur une table.
Post-condition : Le préparateur est notifié de la commande.
Scénario nominal :

1. Le client regarde le menu sur l'application
2. Le client saisit sa commande sur l'application
3. Le client valide sa commande

Nom du cas: ***Préparation de la commande.***

But : Réaliser toutes les préparations d'une commande.
Acteur Principal : Les préparateurs.
Date de création: 09/10/2018.
Nom des responsables: Emeline et Thomas**.**
Version :** 1.0
Pre-condition: Les préparateurs ont une commande à réaliser.
Déclenchement : Le serveur notifie les préparateurs qu'une commande a été prise.
Post-condition: Le serveur vient chercher la commande.
Scénario nominal:

1. Un préparateur est notifié d'une nouvelle commande.
2. Un préparateur réalise la commande.
3. Le préparateur notifie le serveur que la commande est prête.

Scénario Alternatif:

2. a Retard sur la commande
2 .a.1 : Le préparateur s'aperçoit qu'il y aura du retard sur la commande.
2 .a.2 : Le préparateur notifie le serveur de ce retard.
2 .a.3 : Après ce temps d'attente, retour au 2

3. a Changement de dernière minute
3 .a.1 : Le préparateur reçoit une notification qui indique un changement sur une commande.
3 .a.2 : Le préparateur prends en compte le changement de commande.
3 .a.3 : Retour en 2.

Nom du cas : ***Commande servie par le serveur.***

But : Prendre la commande du client.
Acteur Principal : Le Serveur.
Date de création : 09/10/2018.
Nom des responsables : Guillaume.
Version: 1.0
Pré-condition : La commande a été préparée.
Déclenchement :Le serveur est notifié de la commande.
Post-condition : Les client sont servis.
Scénario nominal :

1. Le serveur récupère la commande chez le préparateur
2. Le client reçoit la commande apportée par le serveur
3. Le serveur passe la commande à l'état « servie »

Scénario Alternatif :
2.a Le client remarque qu'il manque quelque chose sur la commande
2.a.1 Le serveur notifie le problème sur l'application.
2.a.2 Le préparateur corrige le problème.
2.a.3 Le préparateur notifie le Serveur
2.a.3 Retour à 2

Nom du cas : ***Encaissement des clients.***
But : Faire payer les clients pour les services demandés.
Acteur Principal : Le serveur.
Date de creation : 09/10/2018.
Nom des responsables : Emeline et Thomas.
Version : 1.0
Pre-condition : Les clients ont fini de consommer et s'apprêtent à partir.
Declenchement : Le serveur récupère l'addition par le biais de l'Application.
Post-condition : Les clients partent du restaurant.
Scenario nominal :

1. Le serveur donne l'addition aux clients.
2. Les clients paient l'addition .
3. Les clients partent du restaurant.

Scenario Alternatif :
2. a Ristourne
2 .a.1 : Le serveur applique la ristourne sur l'addition.
2 .a.2 : Le serveur donne la nouvelle addition.
2 .a.3 : Retour au 2

2.b Le client ne peut pas payer
2 .b.1 : Le serveur fait signer au client une reconnaissance de dettes.
2 .b.2 : Retour en 3

3.1.2 Diagramme d'utilisation

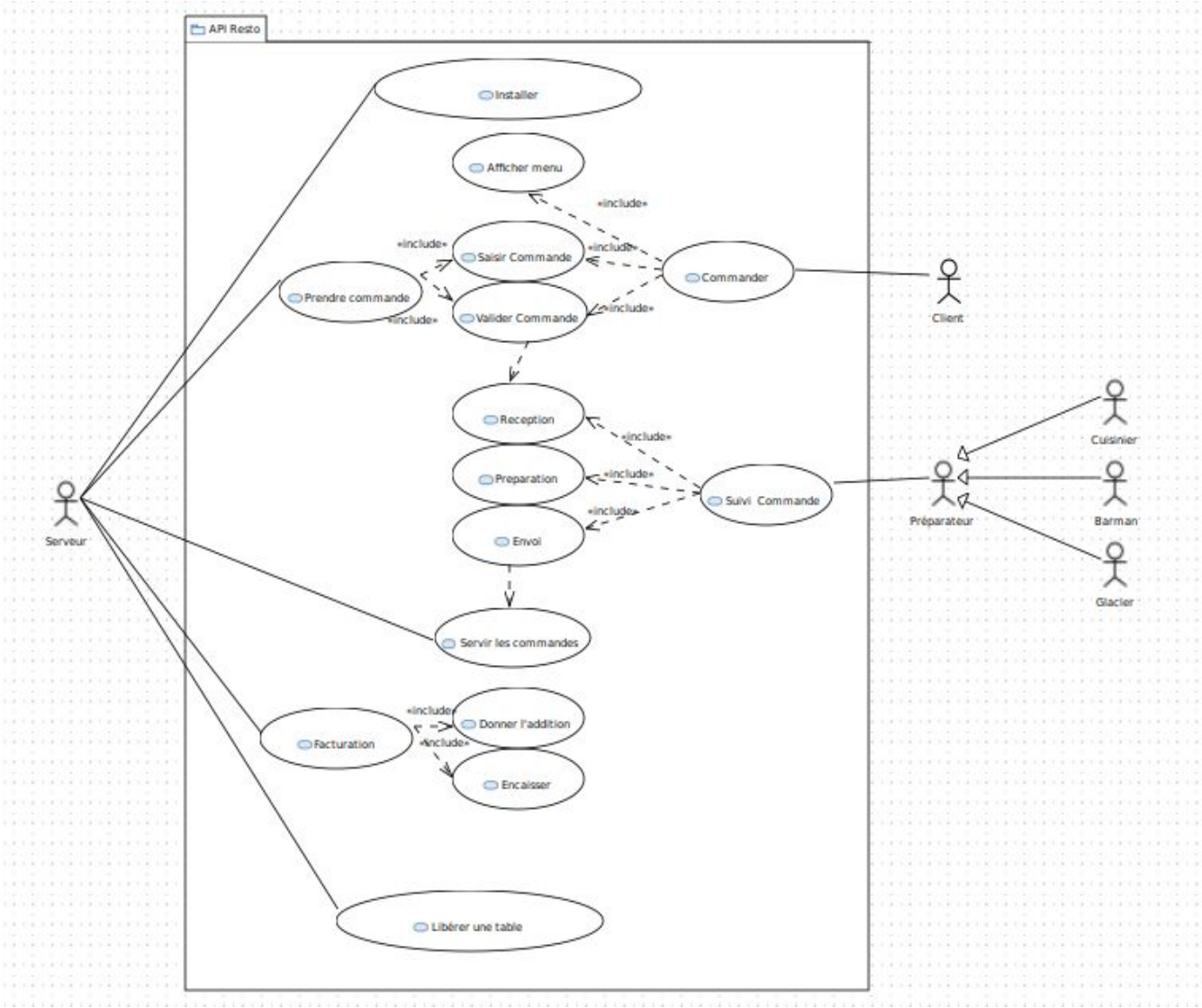


Figure 3 : Diagramme des cas d'utilisations

3.1.2.1 Description

Dans ce diagramme de cas d'utilisation, nous pouvons voir le déroulement complet d'une commande, depuis l'installation des clients jusqu'à la libération de la table.

3.1.2.2 Commander

Le client est capable de consulter le menu sur une tablette à sa disposition.
Ensuite, soit il commande seul, à l'aide de cette même tablette, soit le serveur la prends pour lui (i.e. : saisir la commande puis la valider.).

3.1.2.3 Suivi de commande

Une fois la commande envoyée, celle ci est réceptionnée par un préparateur qui peut être un barman, un cuisinier ou un glacier.
Celui-ci en commence la préparation, puis lorsque celle-ci est terminée, il informe le serveur grâce à l'application, qui peut alors la récupérer.
La réception d'une commande mène a une notification envoyée au préparateur concerné. De même que le fait de terminer la préparation envoie une notification au serveur afin que celui ci vienne la récupérer.

3.1.2.4 Facturation

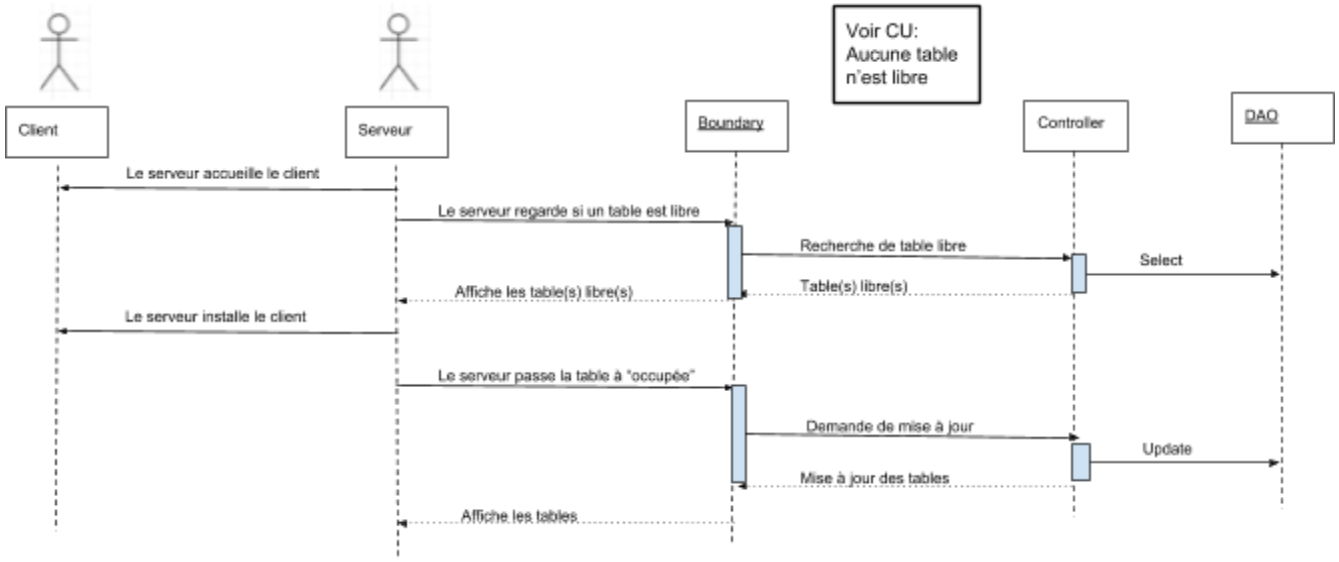
La facturation est constituée de deux étapes.
La première consiste en l'établissement d'une addition, répertoriant l'ensemble des consommations, leurs prix respectifs, ainsi que la somme de ces prix.
Ensuite, l'encaissement consiste à collecter la somme requise par l'addition, que ce soit par carte bancaire, espèce ou encore ticket restaurant, etc ...

3.1.3 Priorisation des CU

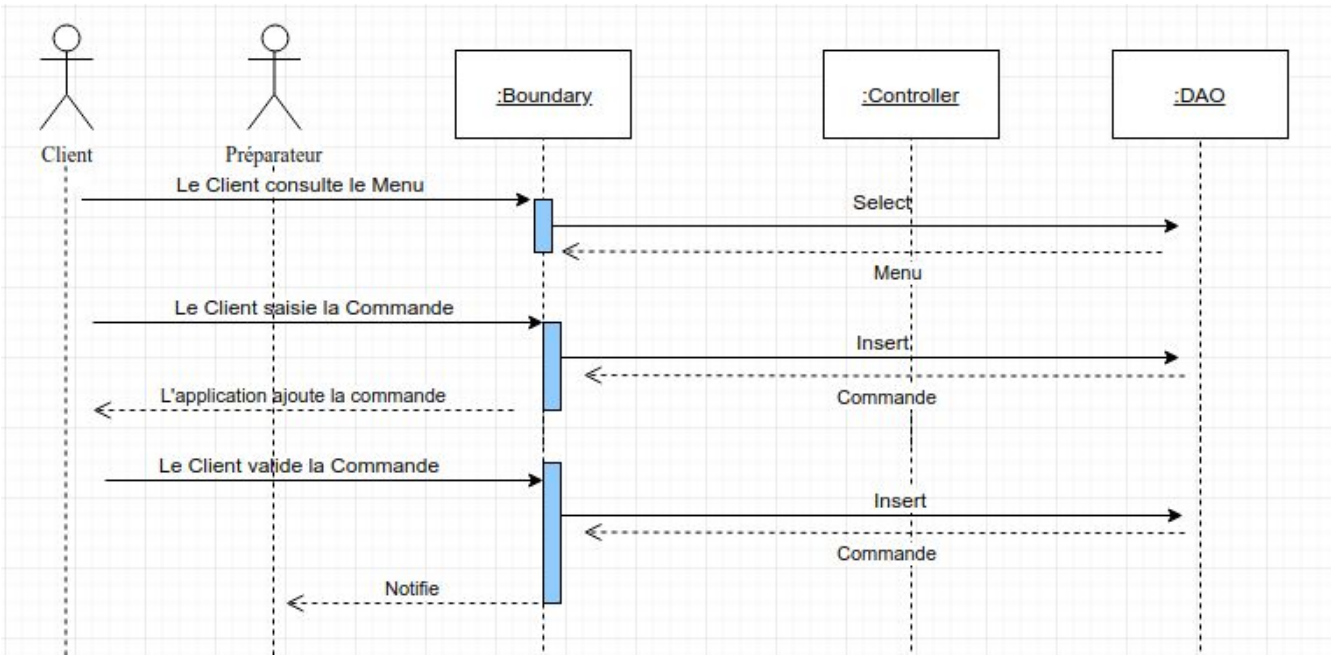
CU	Importance (MOA)	Complexité (MOE)	Argumentation
Installer	3	1	Concret pour le client
Afficher Menu	5	2	Demande de la MOA, sert de base à l'application
Saisir commande	4	3	Opération principale
Valider commande	3	4	Suite directe de la saisie, interactions fortes
Réception commande	2	2	/
Préparation	1	1	Simple changement d'état
Envoi	2	1	/
Servir commande	2	1	/
Donner l'addition	3	2	/
Encaisser	1	5	Risque technologique, sécurité
Libérer table	2	1	/

3.1.4 Diagrammes de Séquence

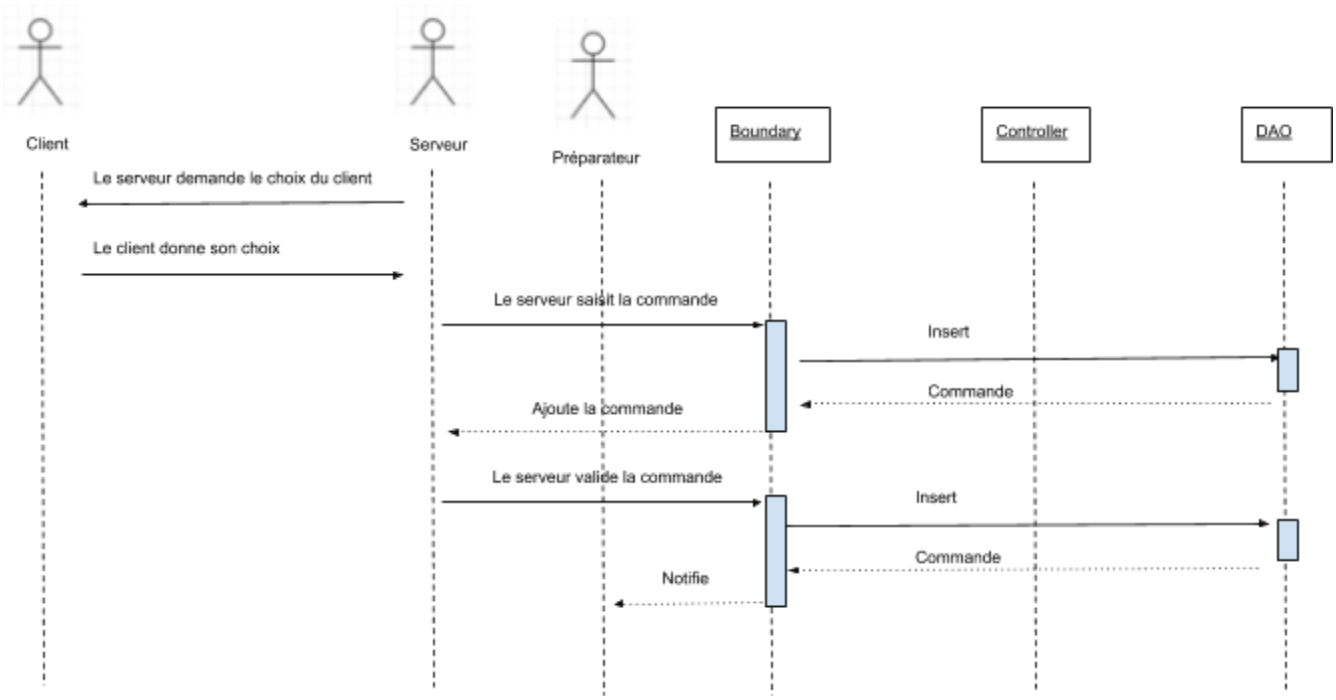
3.1.4.1 Installation Client



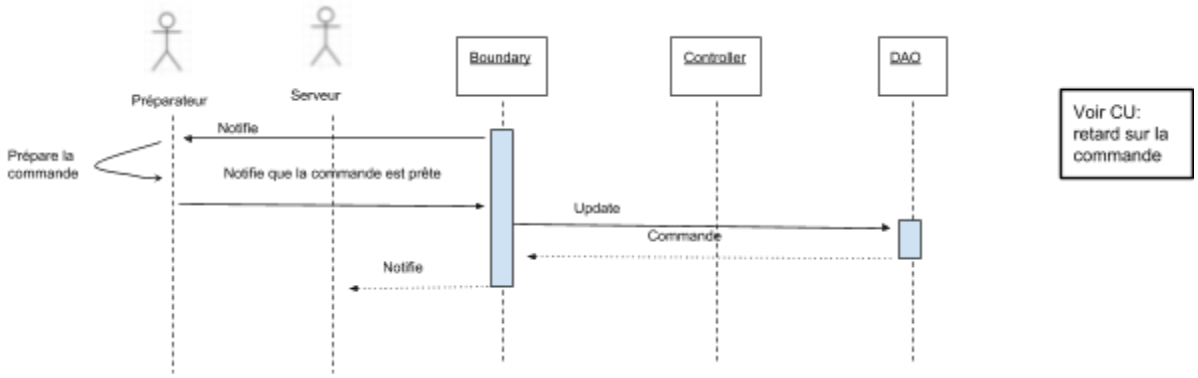
3.1.4.2 Prise de Commande du Client



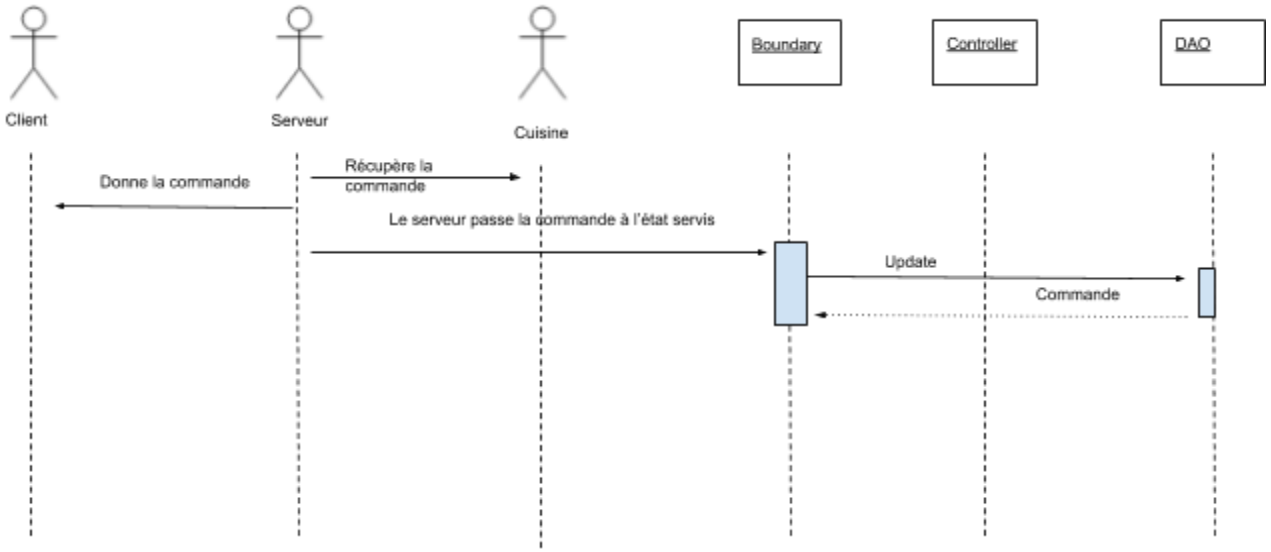
3.1.4.3 Prise de Commande du Serveur



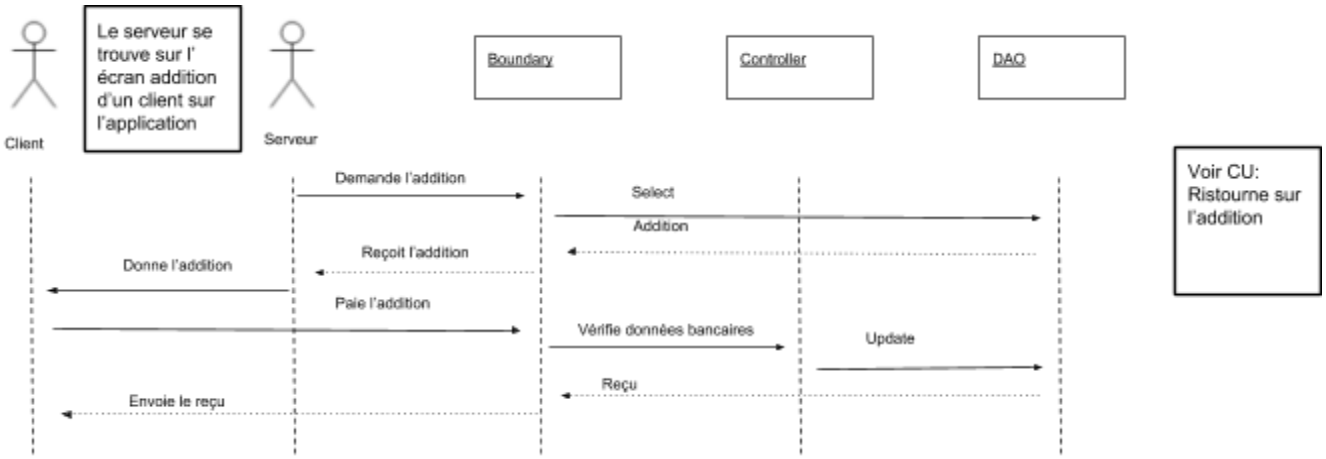
3.1.4.4 Preparation d'une Commande



3.1.4.5 Servir Commande



3.1.4.6 Encaissement



3.2 Architecture du logiciel

3.2.1 Choix de l'environnement

3.2.1.1 Couche serveur

Au niveau du choix de la couche serveur de notre logiciel. Il paraît évident, en vue de notre nom "API:REST'O" de choisir un serveur faisant des requêtes HTTP via une **API REST**. Ainsi, nous aurons notre serveur qui communiquera avec une base de données et qui enverra les différentes informations à notre client.

Pour le choix des technologies, beaucoup de choix sont possibles. Nous avons pensé à un serveur **node JS** avec une base de donnée **MongoDB** ou bien de travailler avec le framework **Spring boot**. Ces technologies sont récentes et "à la mode" et fonctionne parfaitement pour une API REST.

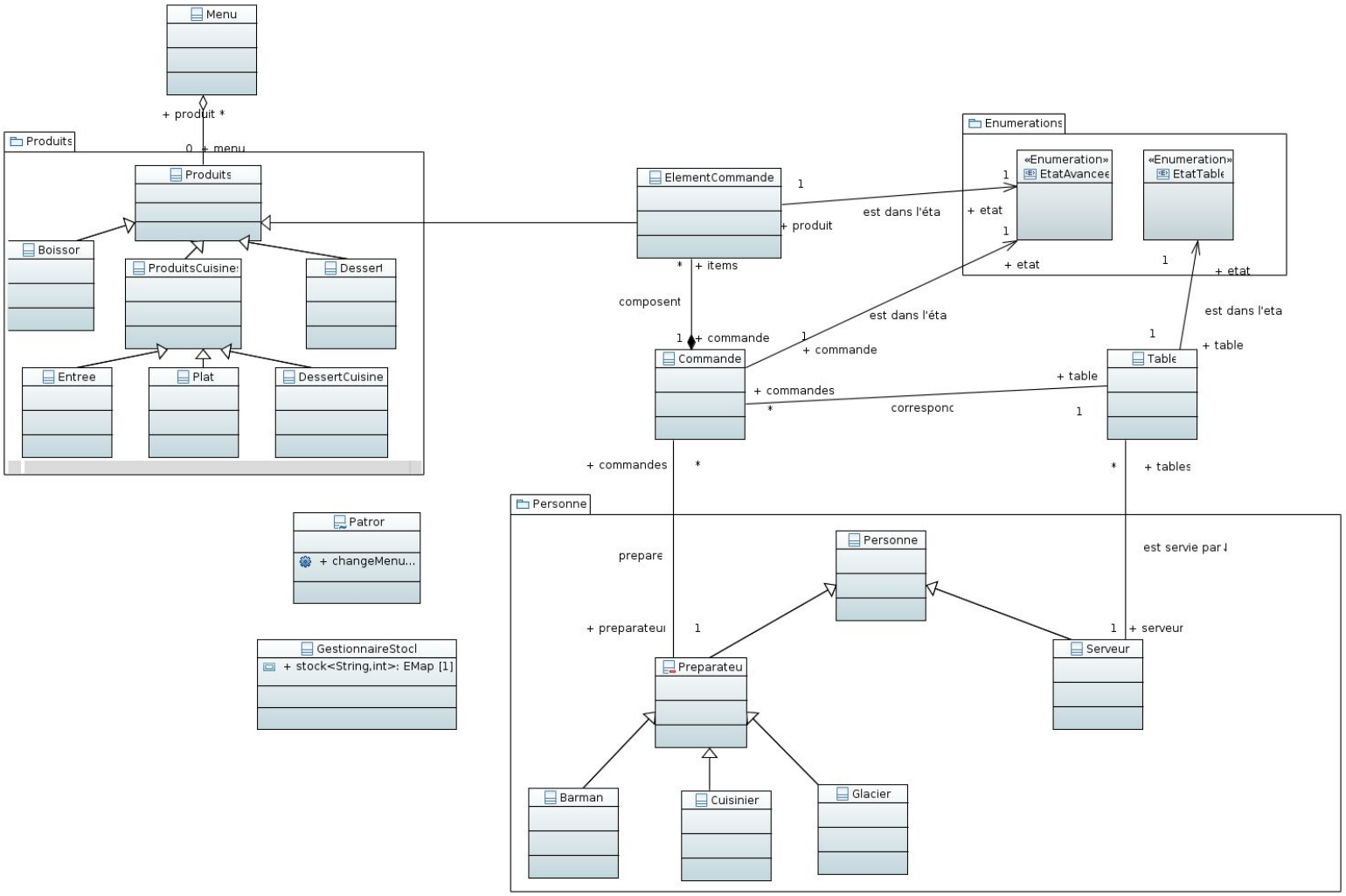
3.2.1.1 Couche client

L'idée est d'avoir une application multiplateforme. En effet, nous aurons les serveurs du restaurant qui utilisent des tablettes afin de créer les commandes et de savoir le montant de l'addition. Ensuite, nous avons des écrans interactifs pour les préparateurs afin de savoir ce qu'ils doivent préparer et de le notifier aux serveurs du restaurant. Enfin, les menus et les produits sont configurable via une IHM par le directeur.

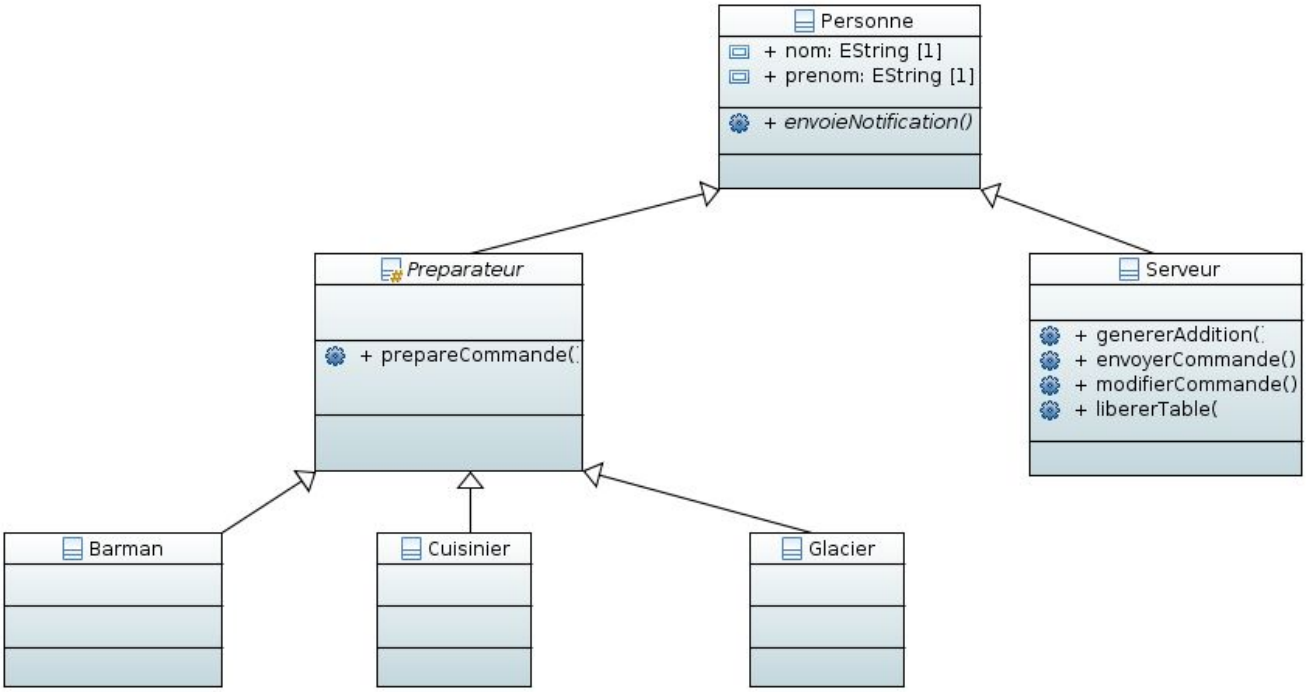
Par conséquent, il faut que l'application puisse s'utiliser sur smartphone/tablette et sur ordinateur via une IHM. Le framework Ionic serait un choix intéressant. En effet, ce framework open source conviendrait pour une application multiplateforme. De plus, il communique très facilement avec une API REST.

3.2.2 Diagramme de classes

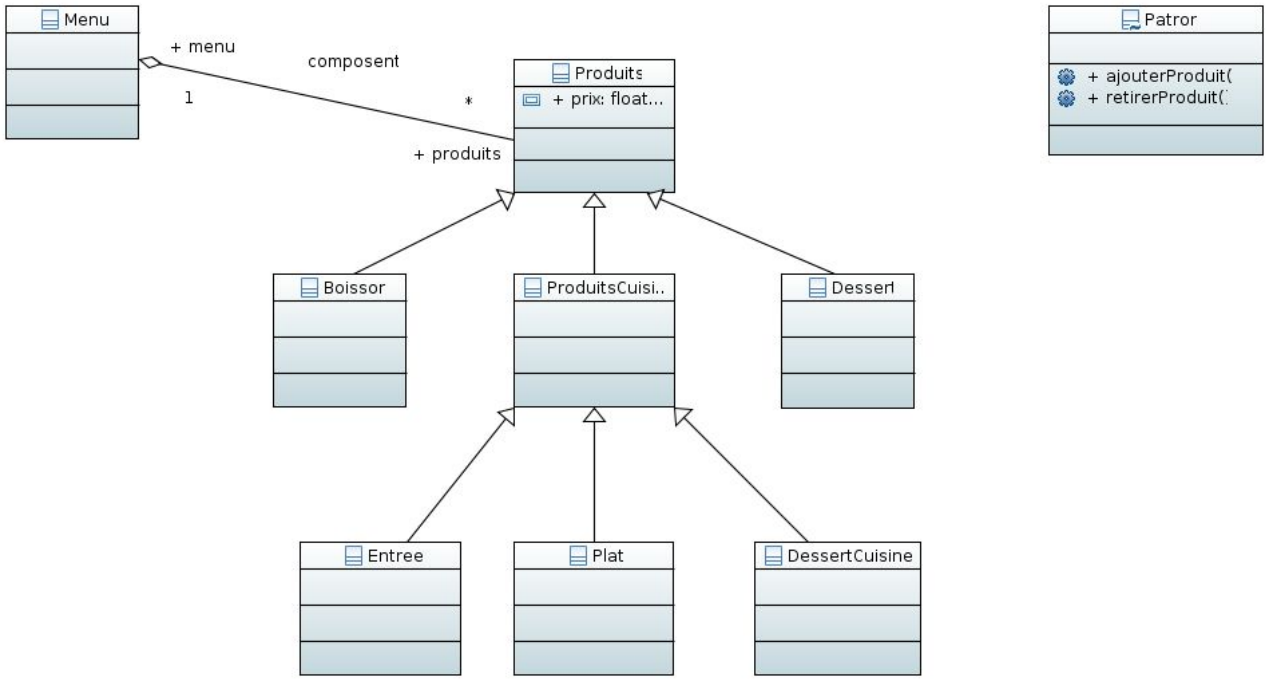
3.2.2.1 Vue d'ensemble



3.2.2.2 UML Personne



3.2.2.3 UML Produits



3.3 Exemple d'écrans sur l'application

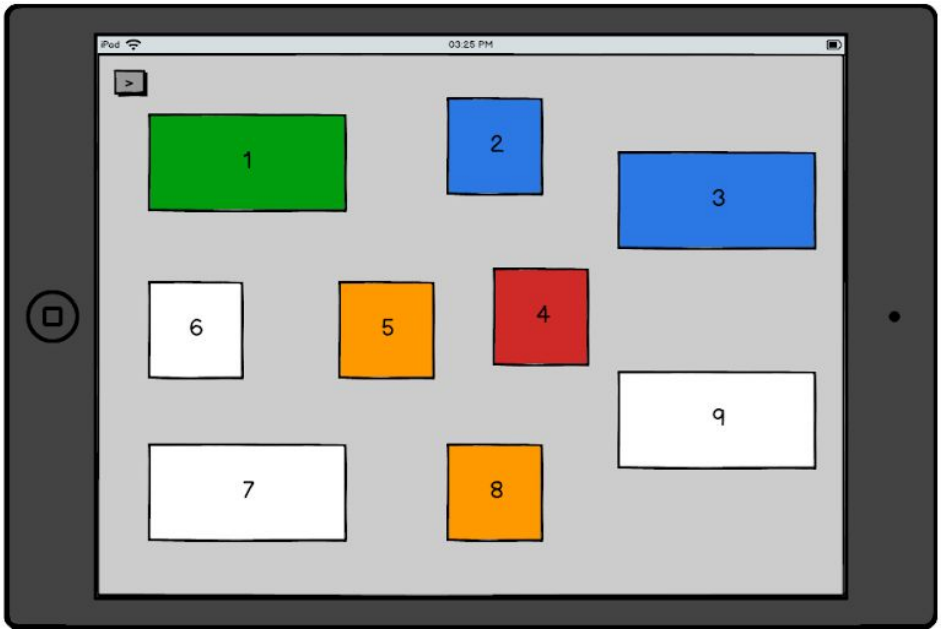


Figure 4 : Écran des tables

Figure 5 : Écran détail commande d'un table

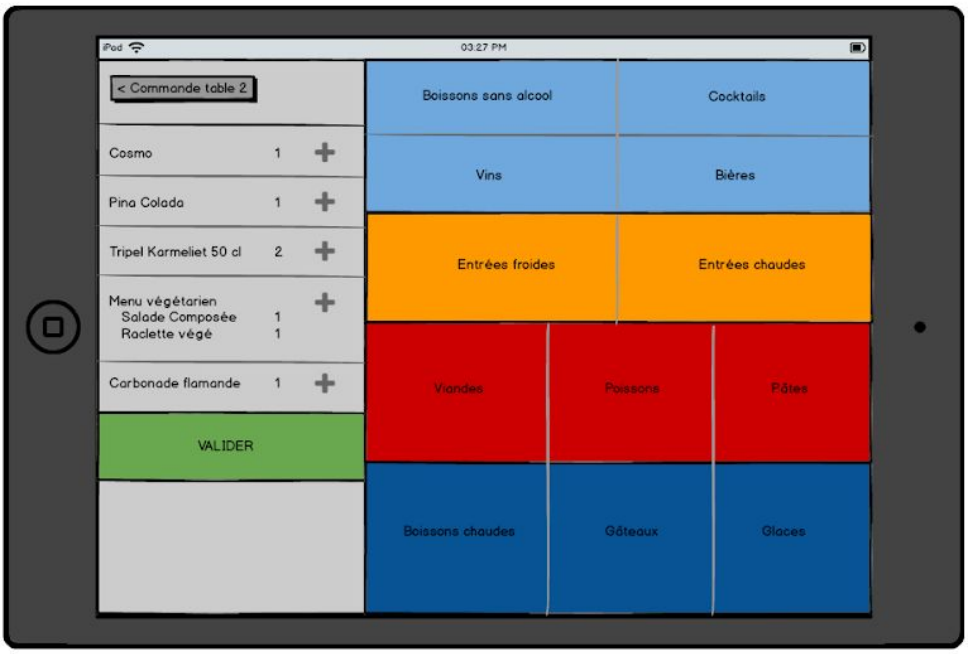


Figure 6 : Écran commander produits

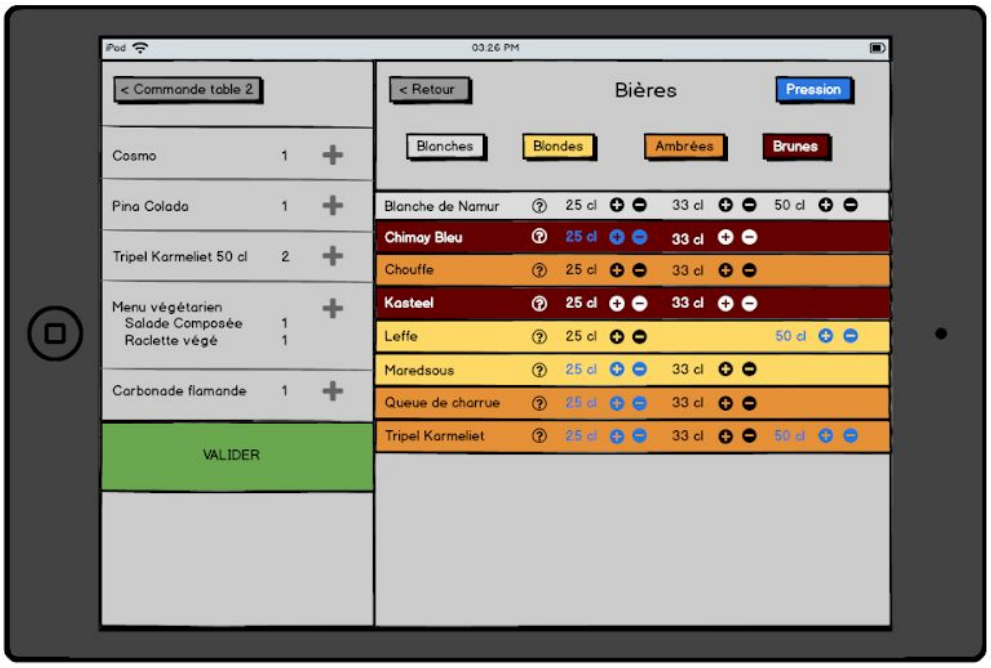


Figure 7 : Écran commander une biere