# DEDICACES

A ma Mère

A mon frère et mes sœurs

A tous mes collègues de classe

A tous mes amis

**Jonathan NKURUNZIZA**

# REMERCIEMENT

Ce rapport est l'aboutissement de quelques années de formation consacrées à la programmation et faites à l’Université du Lac Tanganyika. La réalisation de ce projet a pu être menée à son terme grâce à la coopération d’une multitude de personnes que Je ne saurais citer tous les noms mais auxquelles Je voudrais, en cette occasion, témoigner l’expression de Mon profonde gratitude. Je voudrais, remercier, au préalable Jéhovah, le créateur, qui n’a pas d’égal lui qui sauves et qui fait échapper.

Je voudrais spécialement exprimer ma gratitude envers ma mère qui m’a guidé pas à pas durant mon parcours scolaire et académique. Son soutien et ses encouragements m’ont été précieux et j’en suis redevable d’une très bonne éducation. Qu’elle trouve ici la marque de ma profonde reconnaissance.

Je tiens en outre bien sûr à remercier les autorités de l’Université du Lac Tanganyika qui m’ont accueilli en leur sein pendant 3ans. L’enseignement de qualité dispensé par mes professeurs a également su nourrir ma réflexion et m’a représenté une profonde satisfaction intellectuelle, merci donc aux professeurs de l’ULT.

J’ai aussi une très grande dette à l’égard de tous mes enseignants depuis l’école primaire jusqu’à l’obtention de mon diplôme d’humanité technique dont les critiques et les éclairages m’ont été précieux.

Que Madame GAKUBA Joëlle trouve dans ce rapport, sa part académique pour son encadrement. Outre ses conseils et sa grande disponibilité, elle m’a accordé sa totale confiance et son soutien indéfectible tout au long de ce travail.

Je remercie, en particulier, Madame INGABIRE Christella gerant de Best Outlook Hotel pour l’intérêt qu’elle n’a pas cessé de témoigner à mon travail en m’accordant sa disponibilité, mais aussi pour les discussions enrichissantes que Nous avons partagées ; fondées sur une connaissance exceptionnelle de l’entreprise. J’en suis très reconnaissant pour sa confiance et la richesse de nos échanges. Qu’elle trouve ici le modeste fruit de ma très sincère reconnaissance.

Je voudrais également associer à mes remerciements aux membres de ma famille, mon frère, mes trois sœurs, mes oncles, mes beaux-frères, ma belle-sœur, mes tantes, mes neveux ainsi que mes nièces pour leur implication personnelle ou leurs encouragements et pour leurs efforts en vue du bon avancement de mon travail sur place.

  Enfin, je souhaite spécialement exprimer ma gratitude envers celles et ceux qui m’ont apporté leur aide et témoigné toute leur amitié ; mais aussi à tous ceux qui, de près ou de loin, directement ou indirectement, m’ont apporté leur contribution. Je ne saurais citer chacun son nom. Que tous trouvent ici l’expression de Mon franche reconnaissance.

**NKURUNZIZA Jonathan**

# LISTE DE SIGLES ET ABREVIATIONS

AJAX : Asynchronous Javascript And Xml

CSRF : Cross-site request forgery

CSV : Comma-Separated Values

CSS : Cascading Style Sheets

DRF : Django Rest Framework

DRY : Don’t Repeat Yourself

HTML : Hyper Text Markup Language

HTTP : Hyper Text Transfer Protocol

IA ou AI : Intelligence Artificiel

JS/Js : JavaScript

JSON : JavaScript Objects Notation

kMeans : k Moyennes

kNN : k Nearest Neighbours (k plus Proche Voisin)

MATCH : Méthode d’Analyse et de Conception Hiérarchisée

ML : Machine Learning (apprentissage automatique)

MTV : Model-Template-View

ORM : Object Relational Mapping

PSF : Python Software Foundation

REST : Representational State Transfert

SGBD : Système de Gestion de Base de Données

SQL : Structured Query Language

ULT : Université du Lac Tanganyika

UML : Unified Modeling Language

XML : eXtensible Markup Language

# LISTE DE FIGURES

**Aucune entrée de table des matières n'a été trouvée.**

# TABLE DE MATIERE

Contenu

[DEDICACES i](#_Toc34305074)

[REMERCIEMENT ii](#_Toc34305075)

[LISTE DE SIGLES ET ABREVIATIONS iii](#_Toc34305076)

[LISTE DE FIGURES iv](#_Toc34305077)

[TABLE DE MATIERE v](#_Toc34305078)

[CHAPITRE I: INTRODUCTION GENERALE 9](#_Toc34305079)

[I.1. Introduction 9](#_Toc34305080)

[I.2. Problématique 9](#_Toc34305081)

[I.3. Objectifs de mon travail 10](#_Toc34305082)

[Objectif Global 10](#_Toc34305083)

[Objectifs spécifiques 10](#_Toc34305084)

[I. 4. Hypothèse 10](#_Toc34305085)

[I.5. Intérêt du sujet 10](#_Toc34305086)

[Pour moi 10](#_Toc34305087)

[Pour les gestionnaires des restaurants 10](#_Toc34305088)

[Pour les serveurs des restaurants 10](#_Toc34305089)

[Pour les clients des restaurants 11](#_Toc34305090)

[I.6. Délimitation du sujet 11](#_Toc34305091)

[Dans le temps 11](#_Toc34305092)

[Dans l’espace 11](#_Toc34305093)

[Dans le domaine 11](#_Toc34305094)

[I.7. Méthodes et Techniques de recherche utilisées 11](#_Toc34305095)

[I.8. Limite du systeme 11](#_Toc34305096)

[CHAPITRE II : APERCU SUR LE SITE DE STAGE 12](#_Toc34305097)

[II.1. Description du site 12](#_Toc34305098)

[II.1.1. Historique 12](#_Toc34305099)

[II.1.2. Missions de l’Université du Lac Tanganyika 12](#_Toc34305100)

[II.1.3. Domaines de formation 12](#_Toc34305101)

[II.1.4. Administration de l’Université du Lac Tanganyika 14](#_Toc34305102)

[Figure 1 : Organigramme de l'ULT 14](file:///D:\SCRIPT\Rapport_De_Stage\Rapport%20%20Jonathan%20Nkur.docx#_Toc34305103)

[II.1.5. Situation géographique 15](#_Toc34305104)

[II.2. Conduite du stage 15](#_Toc34305105)

[II.2.1. Déroulement 15](#_Toc34305106)

[II.2.2. Thème du stage 15](#_Toc34305107)

[II.2.3. Solutions proposées 15](#_Toc34305108)

[II.3. Description de la logique Métier de l’application 15](#_Toc34305109)

[II.3.1. L’administrateur 16](#_Toc34305110)

[II.3.2. Table 16](#_Toc34305111)

[II.3.3. chef de service 16](#_Toc34305112)

[II.3.4. le commercial 16](#_Toc34305113)

[II.3.5. le cuisinier 16](#_Toc34305114)

[CHAPITRE III : ANALYSE ET CONCEPTION DU SYSTEME 17](#_Toc34305115)

[III. 1. Langage de modélisation 17](#_Toc34305116)

[III. 1. 1. Définition 17](#_Toc34305117)

[III. 1. 2. Exemples 17](#_Toc34305118)

[III. 1. 3. Présentation du langage UML 17](#_Toc34305119)

[III. 2 Conception d’un Système informatique de gestion des commandes des clients d’un restaurant automatisant les principaux taches des gérants 18](#_Toc34305120)

[III.2.1 Analyse des cas d’utilisation 19](#_Toc34305121)

[Tableau 1 : Identification des acteurs 21](#_Toc34305122)

[*Figure 2 : Diagramme de cas d’utilisation pour administrateur* 21](#_Toc34305123)

[*Figure 3 : Diagramme de cas d’utilisation pour client(Table) et serveur* 22](#_Toc34305124)

[*Figure 4 : Diagramme de cas d’utilisation pour Cuisinier et Commercial* 23](#_Toc34305125)

[III.2.2. Diagramme d’activité 25](#_Toc34305126)

[*Figure 3 : Diagramme d’activité pour «authentification»* 25](#_Toc34305127)

[Figure 4 : Diagramme d’activité pour « ajouter au panier » 26](#_Toc34305128)

[Figure 5 : Diagramme d’activité pour « servir une commande » 27](#_Toc34305129)

[Figure 6 : Diagramme d’activité pour «requisitionner un produit » 28](#_Toc34305130)

[Figure 7 : Diagramme d’activité « payer  la consomation » 29](#_Toc34305131)

[Figure 8 : Diagramme d’activité « s’approvisionner » 30](#_Toc34305132)

[III.3. Diagramme de séquence 31](#_Toc34305133)

[III.3.1. Diagramme de séquence pour « Authentification» 31](#_Toc34305134)

[Figure 8 : Diagramme de séquence pour « Authentification» 31](#_Toc34305135)

[III.3.2. Diagramme de séquence pour « ajouter au panier» 32](#_Toc34305136)

[Figure 9 : Diagramme de séquence pour « ajouter au panier» 32](#_Toc34305137)

[III.3.3. Diagramme de séquence pour « servir une commande» 32](#_Toc34305138)

[Figure 10 : Diagramme de séquence pour « servire une commande» 33](#_Toc34305139)

[III.3.3. Diagramme de séquence pour « servir une commande» 33](#_Toc34305140)

[Figure 10 : Diagramme de séquence pour « servire une commande» 33](#_Toc34305141)

[III.3.3. Diagramme de séquence pour « servir une commande» 33](#_Toc34305142)

[Figure 10 : Diagramme de séquence pour « servire une commande» 33](#_Toc34305143)

[III.3.3. Diagramme de séquence pour « servir une commande» 33](#_Toc34305144)

[Figure 10 : Diagramme de séquence pour « servire une commande» 33](#_Toc34305145)

[III.3.4. Diagramme de séquence pour « Recherche Spécifique» 34](#_Toc34305146)

[Figure 11 : Diagramme de séquence pour « Recherche spécifique» 34](#_Toc34305147)

[III.3.5. Diagramme de séquence pour « Contrôle du permis» 35](#_Toc34305148)

[Figure 12 : Diagramme de séquence pour « Contrôle du permis» 35](#_Toc34305149)

[III.4. Diagramme de classe 36](#_Toc34305150)

[Figure 13 : Diagramme de classe 36](#_Toc34305151)

[III.5. Dictionnaire de données 37](#_Toc34305152)

[Tableau 2 : Dictionnaire de données 39](#_Toc34305153)

[III.6.Diagramme de déploiement 39](#_Toc34305154)

[Figure 14 : Diagramme de déploiement 40](#_Toc34305155)

# CHAPITRE I: INTRODUCTION GENERALE

## I.1. Introduction

A travers le monde, un plus gros développement, dans le domaine de la science est en train de se répandre avec une très grande ampleur basée principalement sur l’informatique; ce qui fait que, dans presque tous les domaines, les entreprises font de plus en plus recours à des logiciels en vue d’accélérer leur productivité voire même en vue de rendre efficaces leurs travaux. L’informatique en est, pour le moment, la base.

L'informatique est un domaine d'activité scientifique, technique et industriel concernant le traitement automatique de l'information par l'exécution des programmes informatiques par des machines : des systèmes embarqués, des ordinateurs, des robots, des automates, etc.[[1]](#footnote-1)

Un système informatique étant un ensemble des moyens de saisie, de traitement et de transmission de l’information mis en œuvre pour une application donnée.

Certaines entreprises, sont dotées de système de gestion de différent taches entre autres, la gestion du personnel, de stock, de vente… mais la plupart de méthodes utilisés sont vraiment très basiques et incomplet et nécessitent toujours une grande maitrise de gestion alors qu’elles peuvent être automatisé ce qui a pour effet que les utilisateur du système se trouvent obligé de faire recourt au application externe et au ressources supplémentaires inutilement.

## I.2. Problématique

Les contraintes sont toujours nombreuses lors de la gestion de stock et des commandes au sein de la quasi-totalité des restaurants qu’on retrouve à Bujumbura. Ce qui entraine notamment les problèmes suivants :

* Les serveurs sont obligé d’écrire les commandes des clients donc pas d’abonnement possible(ou difficile à gérer) même chose pour la fidélisation
* Les restaurants ont tellement de plats qu’un client non habitué se sentira vite perdu par les menus (pas d’image et des descriptions sur les menus)
* Crainte d’essayer un nouveau menu car un client des questions telles que « ce plat est fait avec quelle ingrédient ? ceux qui en ont mangé l’ont trouvé comment ? est-ce que ce n’est pas déconseillé aux gens qui souffrent tel ou telle maladie ? »
* Querelles apres que le client est servi, soit disant qu’on lui à amené ce qu’il n’avait pas commandé et vice versa
* Les gestionnaires de stock ne parviennent pas à prévoir à l’avance combien de temps va durer le stock,
* Quel sera la production optimale pour gagner le plus d’argent
* Difficulté de savoir les plats qui restent quand il y’a forte demande (ex quand il se passe 5 commandes à par le billet des 5 different serveur alors qu’il en restait 3).

## I.3. Objectifs de mon travail

### Objectif Global

Conception et réalisation d’une « **Système informatique de gestion des commandes des clients d’un restaurant automatisant les principaux taches des gérants** » pour répondre aux problèmes cités ci-haut.

### Objectifs spécifiques

* Automatiser les commandes ;
* Faciliter le client dans ses choix ;
* Eviter les malentendus entre client et serveurs
* Limiter les alergies du aux consommations des tel ou tel aliments ;
* Prevoir les ruptures de stock inattandues.
* Eclaircir les produtions optimales
* Eviter les cconflits de commande au moment ou le stock tend à sa rupture

## I. 4. Hypothèse

Après avoir médité sur plusieurs aspects des restaurants et leurs clients et après avoir recueilli certaines informations nécessaires auprès de certains personnels chargés de differents services dans des restaurants aux modalités de travail, aux difficultés qu’ils rencontrent et à l’efficacité du travail, l’hypothèse émise est la suivante : « la conception et la réalisation d’un système informatique permettant de contrôler tout élément lié aux activités qui sont exercées par un restaurant et ses clients et ainsi apporter une solution aux problèmes rencontrés par ces derniers »

## I.5. Intérêt du sujet

Mon travail intitulé « Système informatique de gestion des commandes des clients d’un restaurant automatisant les principaux taches des gérants » présente pour intérêt plusieurs niveaux :

Pour moi**:** Ce travail m’a donné l’occasion de mettre en pratique ce que j’ai appris dans le domaine de la conception et la réalisation des applications mobile à l’aide des langage Java, XML, Python, JavaScript, SQL, HTML/CSS mais aussi comment mieux me servir API pour faire communique une application client avec une application serveur.

Pour les gestionnaires des restaurants**:** ce travail de recherche apportera un plus sur leurs systèmes d’informations habituelles qui sont laplupart des fois non-automatisés (ou presque).

Pour les serveurs des restaurants**:** ce travail de recherche leurs permetra de ne pas se soucier de prendre par cœur ‘que ce qui reste/qu’est que les clients ont commandé’ et d’autre question de ce genre.

Pour les clients des restaurants**:** ce travail sera éfficaces pour les clients nouveaux car ils auront des informations claires sur une recette sans se faire ridiculisé.

**Intérêt académique** : Ce travail pourra servir de références pour les futures recherches dans le domaine.

## I.6. Délimitation du sujet

Ce travail se délimite dans le temps, dans l’espace et dans le domaine :

Dans le temps**:** Mon travail est délimite sur une période allant du mois de décembre 2019 jusqu’ au mois de février 2020

Dans l’espace**:** Mon travail se limite aux services chargés de gestion et de service dans un restaurant.

Dans le domaine: Le système peut facilement s’adapter à la quasi-totalité des restaurants pour les services d’approvissionement, de presentations des recettes aux clients et de gestion des commandes efféctués mais aussi ce système peut facilement etre réadapter pour fonctionner dans les entreprises de le fonctionnement est proche de celle d’un restaurant.

## I.7. Méthodes et Techniques de recherche utilisées

**-** Documentation **:** J’ai consulté les ouvrages en rapport avec mon sujet et les informations sur internet ;

**-** Observations **:** Cette technique m’a permis d’observer comment s’effectue d’habitude les activités d’un resstaurant et les contraintes rencontrées souvent par les serveurs et les clients.

**-** Interview**:** Cette technique m’a permis d’échanger avec les serveurs des restaurants, les clients des restaurants, et les gerants.

## I.8. Limite du systeme

En dépit de la pertinence de Mon application, Mon projet est assujetti à certaines limites comme : l’obligation de disposer d’un serveur, d’une connexion et des téléphones (ou tablettes) Android et au moins deux ordinateurs pour pouvoir utiliser cette application. Mon travail se délimite dans le temps. En effet il a été développée depuis le mois de Janvier jusqu’au mois Févier de l’année 2020. De ce fait plusieurs autres modules dont je desiré tant intégré tel que la gestion des remunerations, les primes, la fidelisations des clients… je les ai pas integrées.

# CHAPITRE II : APERCU SUR LE SITE DE STAGE

## II.1. Description du site

### II.1.1. Historique

La création de l’Université du Lac Tanganyika remonte des années 1999 au mois de Mars par une équipe des 16 cadres (Professeurs d’universités, Administratifs et/ou  
Professionnels) burundais dans le but de promouvoir l’évolution du système éducatif burundais, l’avenir de ses serviteurs et bénéficiaires dans un environnement national et international de plus en plus compétitif et par conséquent impitoyable face à la médiocrité scientifique et professionnelle. Cette initiative visait 3 principaux objectifs ci-après :

✓ Répondre (Contribuer) qualitativement et quantitativement aux demandes éducatives intérieures ;  
✓Contribuer à la diversification des filières et à l’innovation ;  
✓Promouvoir la compétitivité au sein des établissements d’enseignement supérieur

### II.1.2. Missions de l’Université du Lac Tanganyika

L’Université du Lac Tanganyika s’offre plusieurs missions dont les principales sont :   
➢ Dispenser un enseignement scientifique de haut niveau ;

➢ Promouvoir une recherche scientifique orientée sur les priorités du développement et en assurer la diffusion des résultats ;

➢ Organiser la formation en cours d’emploi ;

➢ Promouvoir la coopération interuniversitaire par le biais des échanges de publications, des étudiants et des personnels scientifiques ;

➢ Répondre à la demande du marché de l’emploi et assurer l’adéquation formation-emploi [1].

### II.1.3. Domaines de formation

#### II.1. 3.1. Dans l’ancien système (LMD : Licence-Maitrise-Doctorat)

Les programmes s’étaient largement inspirés de ceux de l’Université du Burundi pour les filières similaires. Pour les filières non organisées par l’UB, les programmes s’étaient inspirés de ceux des écoles et universités étrangères, spécialement Européennes. Pour toutes ses facultés, les enseignements s’étalaient sur deux cycles de deux ans chacun, débouchant sur un diplôme des candidatures pour le premier cycle et un diplôme de licence pour le second.  
Dans l’ancien système, quatre facultés avaient pu être organisées :

➢ La faculté des Sciences de Gestion et d’Economie Appliquée  
➢ La faculté des Sciences Sociales, Politique et Administrative  
➢ La faculté de Droit  
➢La faculté d’Informatique

#### II.1.3.2. Dans le nouveau système (BMD : Baccalauréat-Maitrise-Doctorat)

Depuis l’entrée en vigueur du système BMD avec la loi No 1/022 du 31 décembre 2011, l’Université du Lac Tanganyika a tout mis en œuvre pour s’intégrer, en commençant par les facultés et instituts qui existaient déjà dans l’ancien système. C’est ainsi que tous les enseignements débouchant sur le Diplôme de Baccalauréat sont désormais organisés en six semestres comptant pour 30 crédits chacun.

En outre, l’offre de formation est désormais divisée en enseignement universitaire et enseignement postsecondaire professionnel.

L’enseignement universitaire comprend :

➢ La faculté de droit  
➢La faculté des Sciences de Gestion et d’Economie Appliquée qui se répartit en quatre options dont :

• Audit et contrôle de gestion  
• Finance et Comptabilité  
• Economie et Commerce International  
• Marketing et Management

➢La faculté des Sciences Sociales et Politiques  
➢La faculté d’Informatique repartit en deux options en Bac III :

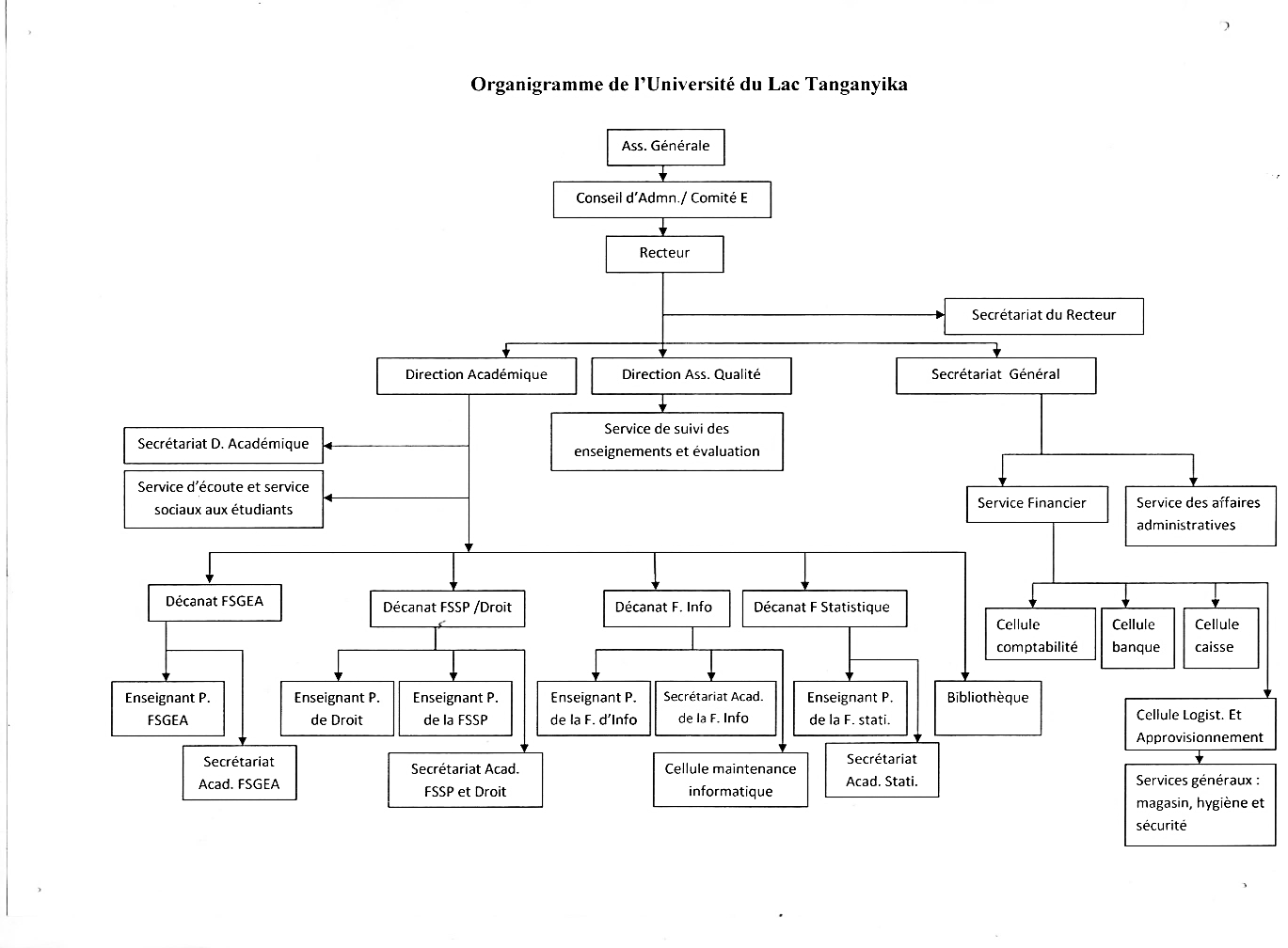
• Génie logiciel  
• Ingénierie des Réseaux

➢L’Institut Inter-facultaire de Statistique

L’enseignement postsecondaire professionnel

• L’Institut de maintenance Informatique et Réseaux  
• L’Institut de Comptabilité et Fiscalité.

### II.1.4. Administration de l’Université du Lac Tanganyika

Elle est représentée par l’organigramme ci-après :

# Figure  : Organigramme de l'ULT

### II.1.5. Situation géographique

Les locaux de l’Université de Lac Tanganyika sont situés dans la partie nord de la capitale burundaise (Bujumbura) en commune Ntahangwa dans quartier Gihosha.

## II.2. Conduite du stage

### II.2.1. Déroulement

Notre stage à l’ULT a pris une durée d’un mois allant du 04 Novembre 2019 au 3 Janvier 2020. Au cours de cette période les stagiaires dont Je fais partie étaient divisés en groupes allant de 1 à 3 personnes au maximum qui travaillaient sur des sujets validés par le jury de la faculté d’informatique. Chaque groupe était supervisé par un enseignant qualifié dans le domaine de la programmation. Le travail consistait aux analyses, conceptions et réalisations des applications informatiques en langages de programmations diverses suivant le choix de chaque groupe.

Pour mettre les stagiaires dans un climat favorable au travail, la faculté nous a dopté d’une salle de stage équipée d’une connexion internet pour les recherches.

### II.2.2. Thème du stage

En tant que programmeurs, tous les groupes travaillaient leurs sujets sous le même thème de stage : « **Analyses, conceptions et réalisations des applications informatiques** ».

### II.2.3. Solutions proposées

Par rapport à l’hypothèse émise je propose de concevoir et développer un système multi-plateforme qui permettra de :

* Automatiser les commandes ;
* Faciliter le client dans ses choix ;
* Eviter les malentendus entre client et serveurs
* Limiter les alergies du aux consommations des tel ou tel aliments ;
* Prevoir les ruptures de stock inattandues.
* Eclaircir les produtions optimales
* Eviter les conflits de commande au moment ou le stock tend à sa rupture

## II.3. Description de la logique Métier de l’application

Le « **Système informatique de gestion des commandes des clients d’un restaurant automatisant les principaux taches des gérants** » possède cinq types d’utilisateurs à savoir les tables, le chef de service, le commercial, l’administrateur, et le cuisinier.

Les droits d’utilisateur permettent de spécifier pour chaque type d’utilisateur les tâches auxquelles il a droit, et ainsi disponibiliser un nombre de pages de l’application auxquels il pourra accéder suivant son statut (table, administrateur, chef de service…). Pour accéder aux tâches auxquelles on a droit dans le systeme, tout utilisateur passe par la page d’authentification où il saisit ses informations de connexion à son compte utilisateur (pseudo et mot de passe) afin d’avoir la possibilité d’accéder à toutes les pages auxquelles il a droit.

### II.3.1. L’administrateur

Après l’authentification le personnel administrateur est dirigé à la page principale pour ajouter les utilisateurs, leur attribuer les droits, créer les tables visualiser l’historique du stock et l’historique des revenues.

### II.3.2. Table

Chaque table est pris pour un utilisateur ce qui a une double mission à savoir identifier d’où vient une commnade et eviter que les clients choissisent eux mene les tables où ils sont, ce qui permet d’éviter qu’un client se fasse passer pour un autre client sur une autre table.

Dans le restaurant chaque table est muni d’un appareil android (tablette de préference) et chaque table étant pris pour un utilisateur l’utilisateur pourra passer ses commandes il n’aura pas besoins de s’authentifier.

### II.3.3. chef de service

Le chef de service est un utilisateur qui une fois connécté est redirigé sur une interface qui est prevues pour visualiser les commandes par tables à temps réel. Pour chaque commande il pourra 1) l’approuver pour la mettre dans la listes des commande en cours de preparation et signal par ce fait au client que sa commade et approuver. La commande approuvé ne peut plus etre modifié par le client. 2) la valider quand le serveur vais la livrer au client et par cette acte une reçu (facture) est imprimé et lui-même ne peut plus la modifier.

### II.3.4. le commercial

Le caissier est un utilisateur qui une fois connécté est redirigé sur une interface ou il peut valider les payements, visualiser l’etat du stock pour faire des approvissionnements, visualiser la recette model, le meilleur client, et autres statistiques.

### II.3.5. le cuisinier

Le cuisinier est un utilisateur qui une fois connécté est redirigé sur une interface ou il peut visualiser à temp reéls les details des commandes du plus ancien au plus récent. Le cuisinier valide chaque fois les recettes qu’il a fini de préparer. Une fois que les recettes qui compose une commandes sont tous validé le systemè marque automatiquement la commande comme disponibles et ainsi le chef de service reçoit un signal et ainsi il peut dire aux serveurs de l’apporter au client.

# CHAPITRE III : ANALYSE ET CONCEPTION DU SYSTEME

## III. 1. Langage de modélisation

### III. 1. 1. Définition

* Un langage de modélisation est un langage artificiel qui peut être utilisé pour exprimer de l'[information](https://fr.wikipedia.org/wiki/Information) ou de la [connaissance](https://fr.wikipedia.org/wiki/Connaissance) ou des [systèmes](https://fr.wikipedia.org/wiki/Syst%C3%A8me) dans une structure qui est définie par un ensemble cohérent de règles. [5]

### III. 1. 2. Exemples

MATCH: Méthode d’Analyse et de Conception Hiérarchisée,

MERISE : Méthode d’Etude et de la Réalisation Informatique de Système d’Entreprise,

MBT: [Model-based Testing](https://fr.wikipedia.org/wiki/Model-based_testing) ,

UML (Unified Modeling Language):

Dans mon travail j’ai choisi UML.

### III. 1. 3. Présentation du langage UML

Le langage de modélisation unifié (UML) est un langage graphique permettant de visualiser, spécifier, construire et documenter les artefacts d'un système gourmand en logiciels. Le langage UML offre un moyen standard d'écrire les plans d'un système, y compris des éléments conceptuels tels que des fonctions système ainsi que des éléments concrets tels que des instructions de langage de programmation, des schémas de base de données et des composants logiciels réutilisables.

Adopté et standardisé par l’Object Management Group depuis 1997, UML est aujourd’hui un outil de communication incontournable, utilisé sur des centaines de projets par le monde.

En effet UML présente beaucoup d’avantages dont les points forts sont :

* gain de précision
* gage de stabilité
* encourage l'utilisation d'outils
* support de communication performant
* Il cadre l'analyse.
* Il facilite la compréhension de représentations abstraites complexes.
* Son caractère polyvalent et sa souplesse en font un langage universel.

**3.2. Importance de l’UML**

UML sert principalement à:

* Décomposer le processus de développement;
* Mettre en relation les experts métiers et les analystes ;
* Coordonner les équipes d’analystes et de conception;
* Séparer l’analyse de la réalisation;
* Prendre en compte l’évolution de l’analyse et du développement;
* Migrer facilement vers une architecture objet d’un point de vue statique et dynamique.

## III. 2 Conception d’un Système informatique de gestion des commandes des clients d’un restaurant automatisant les principaux taches des gérants

Mon système est modélisé par ces cinq diagrammes :

➢***Diagramme de cas d’utilisation*** : Sont ceux qui donnent une description sur les acteurs du système et l’utilisation du système. Les Diagrammes de cas d’utilisation modélisent à quoi sert le système en organisant les interactions possibles avec les acteurs.

Un acteur dans le système représente une personne ou un périphérique qui interagit avec le système tandis qu’un cas d’utilisation exprime une suite d’interactions entre acteur et l’Application, c’est à dire il définit une fonctionnalité utilisable par un acteur.

➢***Diagramme de Séquences*** : permet de décrire comment les éléments du système interagissent entre eux et avec leurs acteurs.

➢***Diagramme de Classes*** : Ce Diagramme va nous permettre de donner une représentation statique du système que nous allons développer. Il est centré sur la présentation des classes et les différentes relations qui existent entre elles.

➢***Diagramme d’activités*** : est une variante des machines à états dans laquelle les états correspondent à l’exécution d’actions ou d’activités (i.e. sont des états action) et où les transitions sont automatiques. Un diagramme d’activités s’utilise pour montrer les événements correspondant à des actions internes (i.e. des flots de contrôle de procédures). Un diagramme d’activité est attaché dans sa globalité soit à une classe, soit à un cas d’utilisation, soit à un paquetage, soit à l’implémentation d’une opération.

➢***Diagramme de Déploiement*** : Le diagramme de déploiement définit l’architecture matérielle du système ou de l’application. Il représente les différents processeurs, périphériques et la répartition du système sur ces différents éléments. [3]

### III.2.1 Analyse des cas d’utilisation

#### 1. Identification des acteurs et cas d’utilisation

|  |  |
| --- | --- |
| Acteur | Rôles |
| Administrateur | * Ajouter, modifier, supprimer le personnel * Ajouter, modifier supprimer les permissions, * Ajouter, modifier, supprimer les fournisseurs, * Ajouter, modifier, supprimer Offres, * ajouter, modifier, supprimer Places, * Ajouter, modifier, supprimer Produits * Ajouter, modifier, supprimer Recettes * Ajouter, modifier, supprimer Stocks, * Ajouter, modifier, supprimer Tables, * visualiser l’histogramme statistique en courbe de service * faire le filtrage par date l’ l’histogramme statistique en courbe de service * visualiser l’histogramme en barres de service * visualiser l’histogramme statistique en courbe des depenses * visualiser l’histogramme en barres de des depenses * visualiser l’histogramme en barres des votes des clients sur les recettes, * visualiser l’histogramme en barres des recettes modales * faire le filtrage par date l’ l’histogramme statistique en barres des recettes modale |
| Client (Table) | * visualier tout les recettes disponibles * ajouter les recettes dans le panier * ajouter, modifier, supprimer les élements qu’il a mis dans son panier avant de passer la commande * passer une commande (valider le panier) * visualier toutes ses commandes * faire un vote sur les recettes commandées * donner son feed back |
| Commercial | * Ajouter, modifier, supprimer les fournisseurs, * Ajouter, modifier, supprimer Offres, * Ajouter, modifier, supprimer Produits * S’approvisionner * Visualier l’historique d’un element spécifique du stock * Approuver la requisistion |
| Serveur | * Visualier les tables (clients) selon les places ou ils sont situées * Visualiser les commandes d’une table specifique * Personnaliser la position des tables dans une place specifique * Se charger d’une commade specifique * Recevoir une notification quand une commande est prête à etre servi |
| Cuisinier | * Visualier les recettes commandées * Visualier l’etat du stock * Enlever une recette de la liste des recettes commandables en cas de rupture d’un produit composante de la recette * Effectuer les requisitions |

### Tableau 1 : Identification des acteurs

#### 2. Diagramme de cas d’utilisation

Les cas d’utilisation de notre système sont représentés par le diagramme des cas d’utilisation suivant :

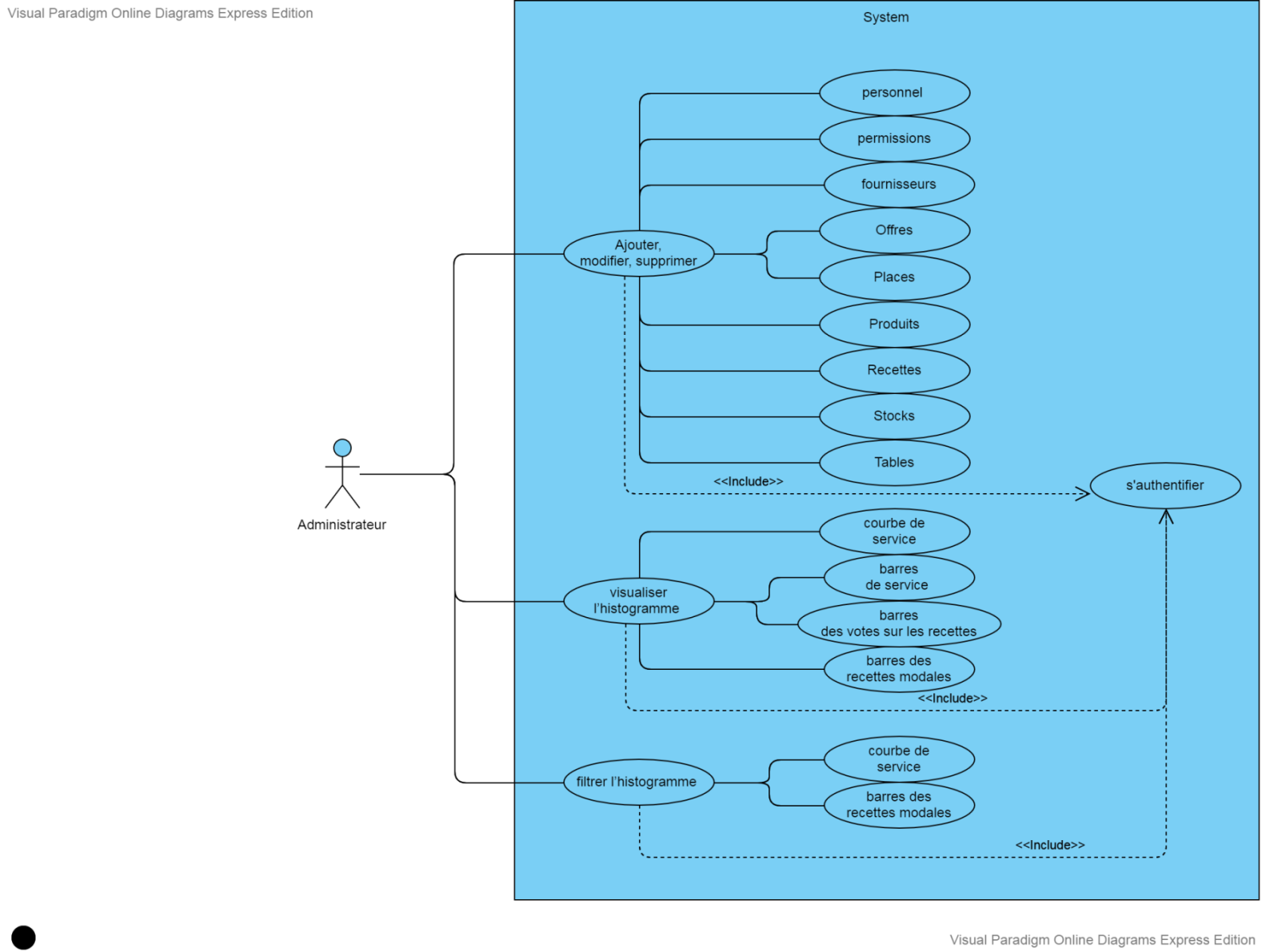


Figure : Diagramme de cas d’utilisation pour administrateur

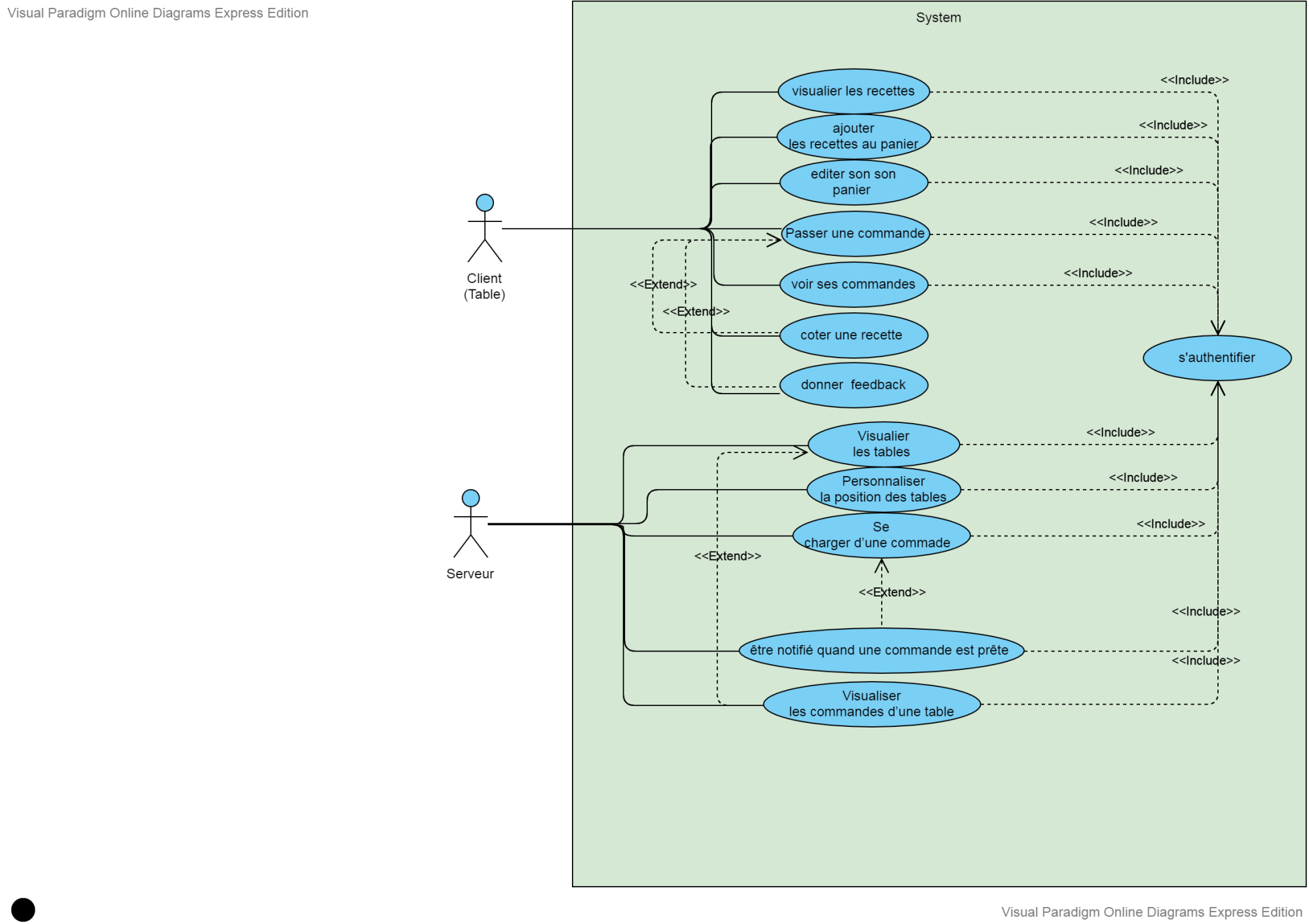


Figure : Diagramme de cas d’utilisation pour client(Table) et serveur

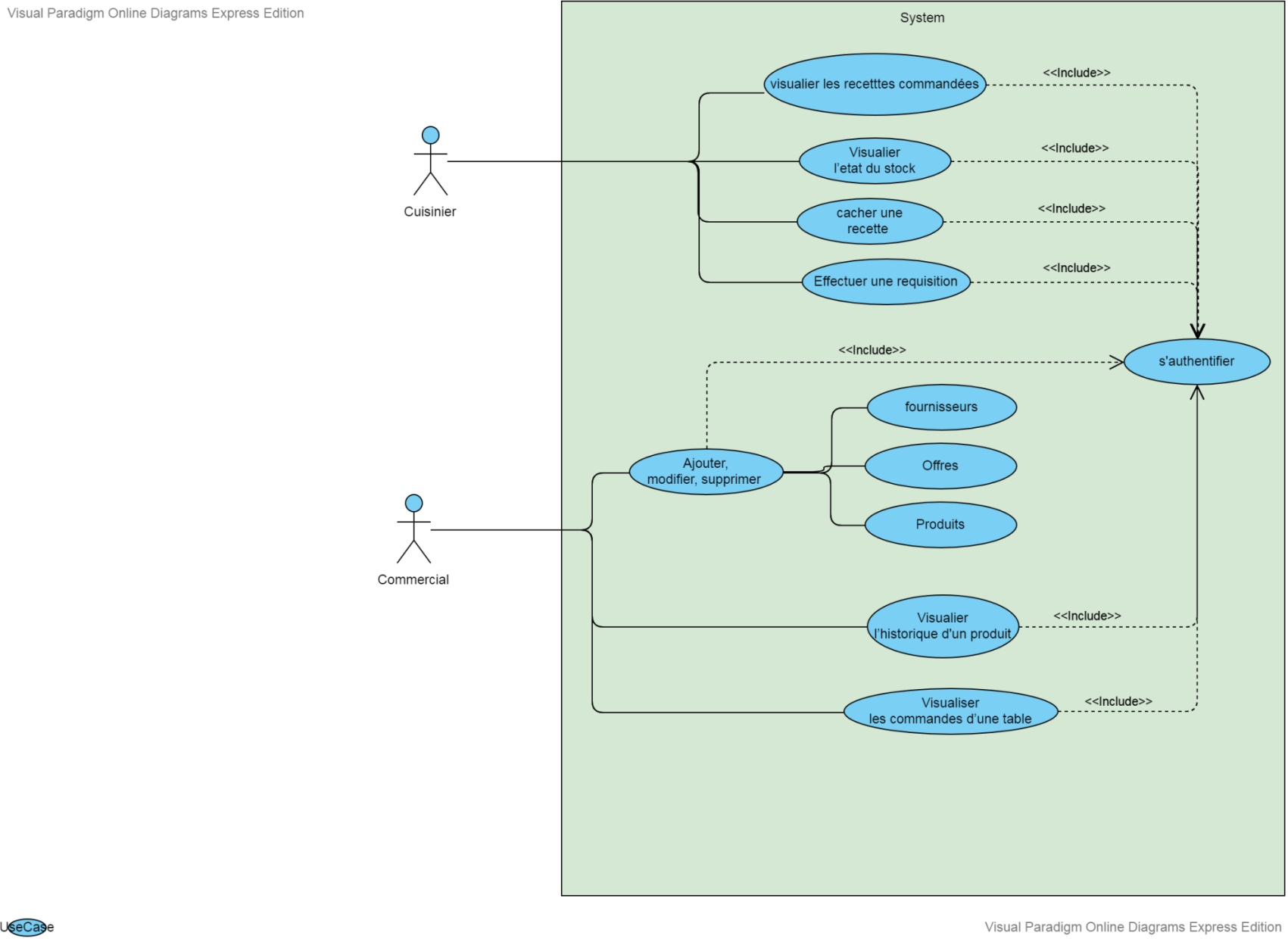


Figure : Diagramme de cas d’utilisation pour Cuisinier et Commercial

#### 3. Description textuelle des cas d’utilisation

**➢Authentification :**

Objectif : Permettre aux utilisateurs de l’application de s’authentifier pour accéder au système.

Acteur concerné : Tous les utilisateurs du present système.

Pré condition : Avoir acces à l’addresse à laquelle l’application est hebergée.

* ***Scénario normal :***

1. L’utilisateur saisit son nom d’utilisateur et son mot de passe.
2. L’utilisateur valide en cliquant sur le bouton « valider »
3. le système ouvre la page d’acceuil correspondant au dans lequel l’utilisateur

* ***Scénario alternatif :***
* Si à l’étape 2 le système réalise que le pseudo et/ou le mot de passe ne correspond pas on reste à l’étape 1 avec un message d’erreur. appartient. Si l’utilisateur est dans plusieurs groupe il’a la possibilité de changer d’interface de travail.

**➢ Ajouter au panier :**

* Objectif : metter à part des recettes à commander pour les editer avant de les commader et les commander.
* Acteurs concernés : les clients qui sont sur une table.
* pré condition : Avoir une péripherique ayant un navigateur capable d’utiliser le service http (ou https).
* **Scénario normal :**

1. Le client clique sur le boutton «ajouter au panier » qui est sur chaque recette dans la liste des recettes sur la page d’accueil.
2. l’utilisateur ouvre l’onglet « panier »
3. l’utilisateur appuis sur bouton commander

* **Scénario alternatif :**
* Lors de l’etape 1 l’utilisateur ne touve pas directement la liste de recettes par defaut, c’est qu’il n’est pas sur la page d’accueil. Il devra d’abord choisir l’onglet « dashboard »
* Lorque l’utilisateur passe une commande, le syteme lui demande son numero de télephone ainsi il a la possibilité de donner son numero de télephone et de laisser un message qui accompange sa commande (comme par exemple « sans sucre SVP ») le numero permettra de se faire fideliser si le restaurant fait la fidelisation.

**➢ Servir une commande:**

* Objectif : se charger de servir une table et recevoir la mise à jour de l’etat de la commande.
* Acteurs concernées : membre du groupe service.
* pré condition : Etre authentifié.
* **Scénario normal :**

1. L’utilisateur

* **Scénario alternatif :**
* Si à l’étape 1 le système réalise que l’identifiant trouvé

**➢ Effectuer une requisition:**

* Objectif : prendre une petite quantité d’un produit du stock à la cuisine.
* Acteurs concernées : un cuisinier et un commercial.
* pré conditions : Etre authentifié.
* **Scénario normal :**

1. Le cuisinier ouvre l’interface «requisition» et trouve un formulaire
2. Le cuisinier choisi un produit parmi les produits disponibles dans le stock, et la quantité qu’il veut requisitionner.
3. Le commercial quant à lui, il ouvre l’onglet, réquisition à partir de son paneau. Il retrouve un tableau

* **Scénario alternatif :**

1. Si à l’étape 2 le cuisinier veux requisitionner plus d’un produit a aussi la posibilité d’ajouter d’autres formulaires en cliquant sur «[ajouter un autre produit](javascript:void(0)) ».

### III.2.2. Diagramme d’activité

#### 1. Diagramme d’activité pour « Authentification »

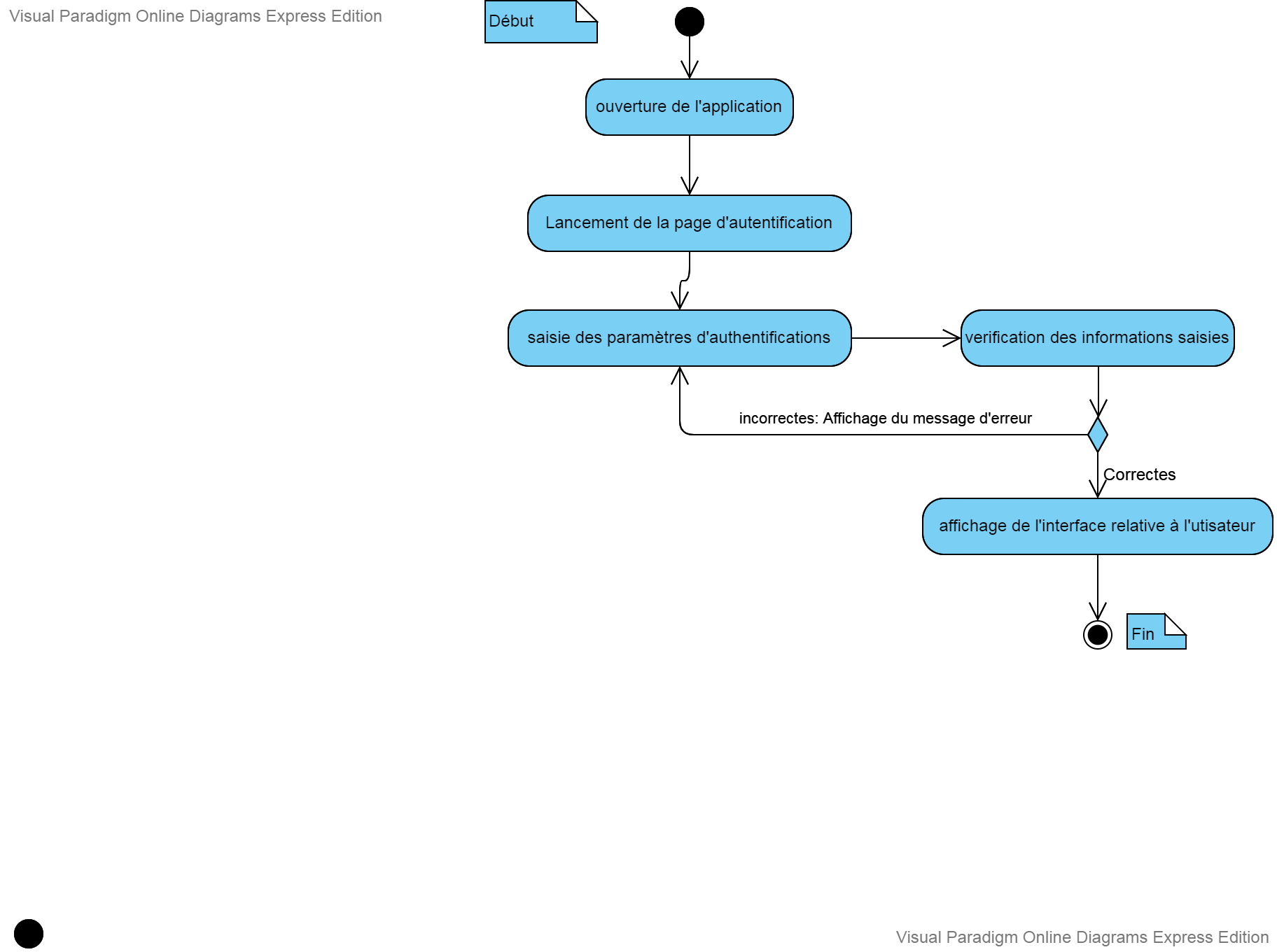
******

Figure : Diagramme d’activité pour «authentification»

#### 2. Diagramme d’activité pour « ajouter au panier »

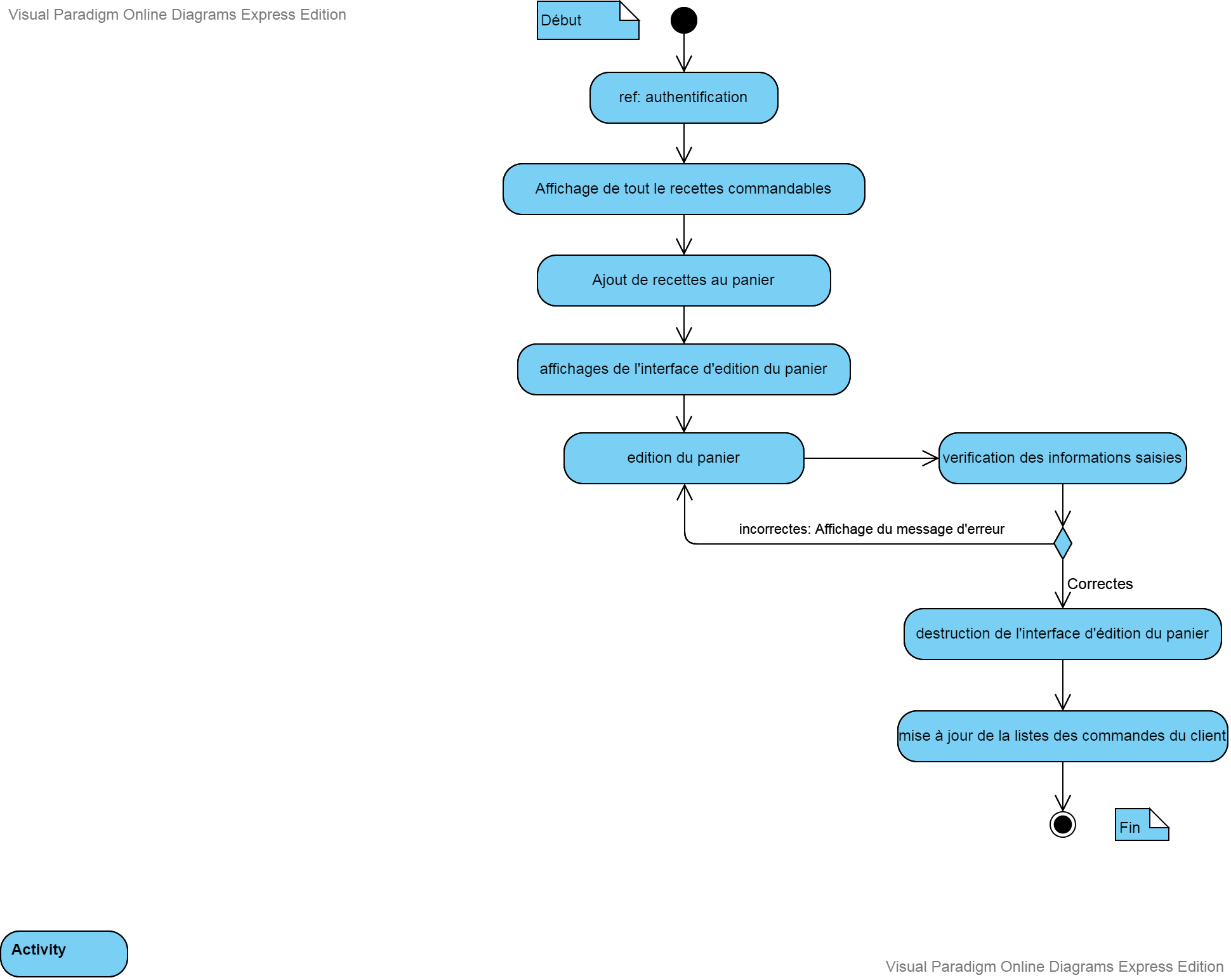
****

Figure : Diagramme d’activité pour « ajouter au panier »

#### 3. Diagramme d’activité pour « servir une commande

Figure : Diagramme d’activité pour « servir une commande »

#### 4. Diagramme d’activité pour « requisitionner un produit » C:\Users\KonstrIctor\Downloads\cuisine requisition.png

Figure : Diagramme d’activité pour «requisitionner un produit »

#### 5. Diagramme d’activité pour « payer  la consomation»

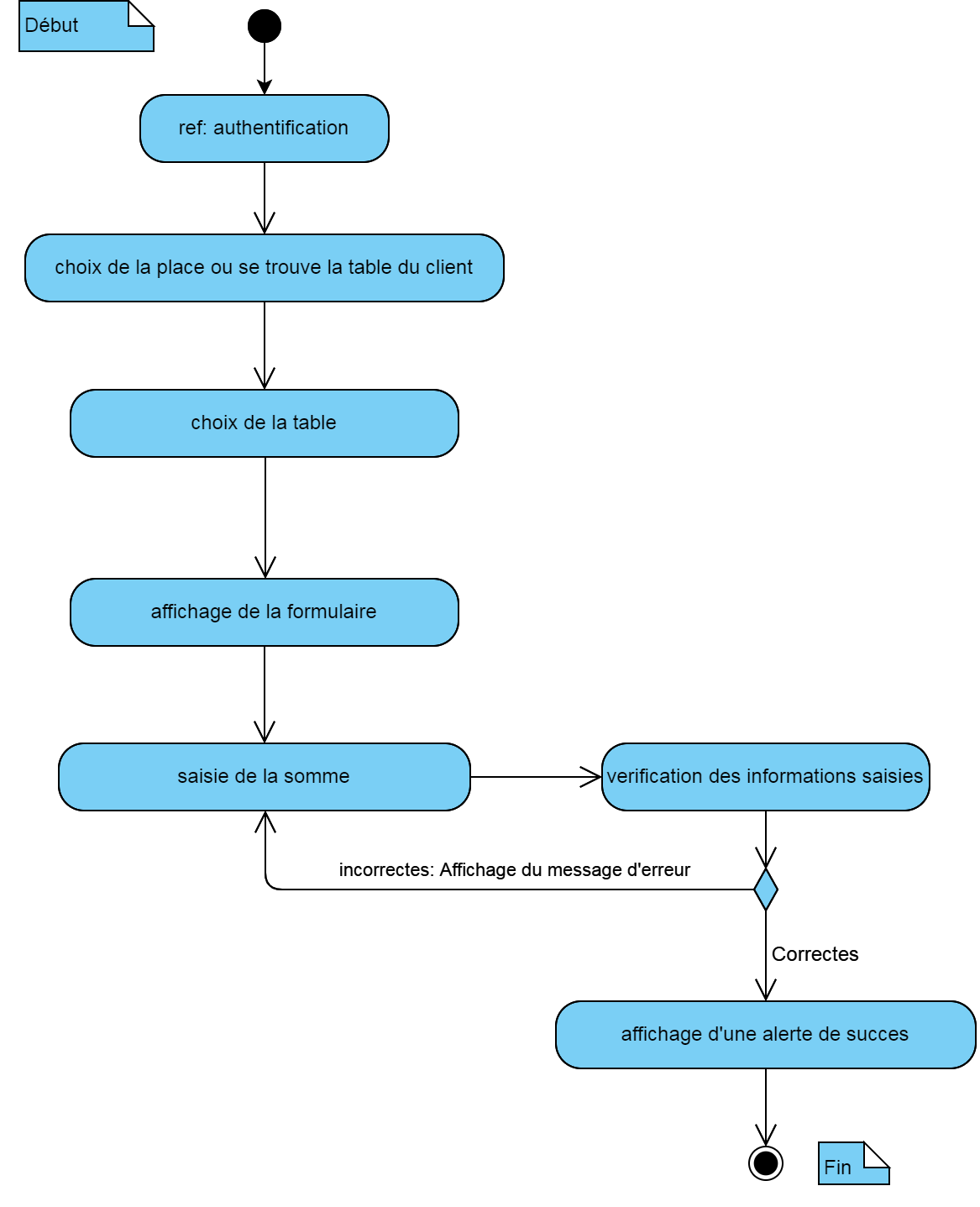


Figure : Diagramme d’activité « payer la consomation »

#### 6. Diagramme d’activité pour « s’approvisionner»

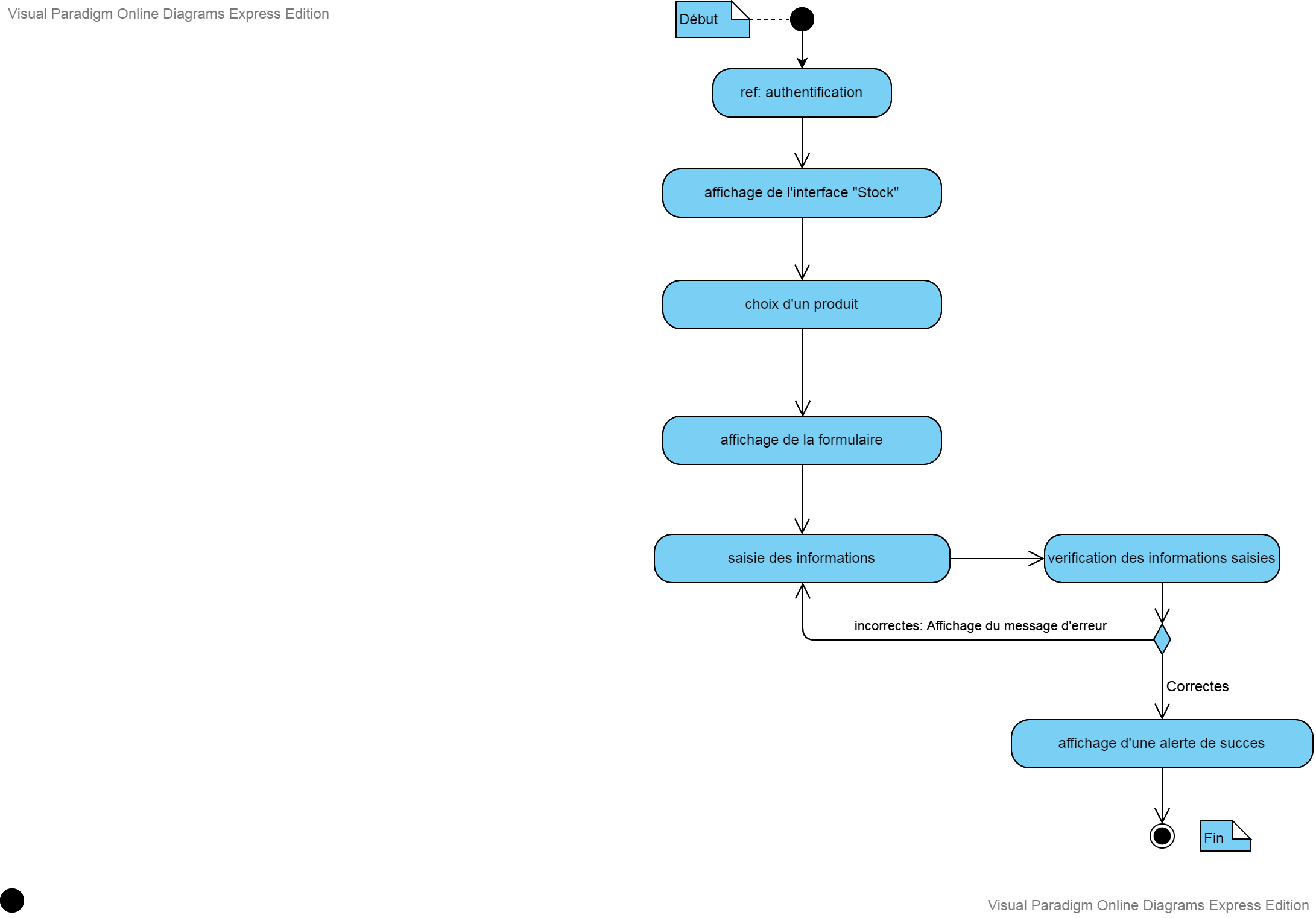


Figure : Diagramme d’activité « s’approvisionner »

## III.3. Diagramme de séquence

### III.3.1. Diagramme de séquence pour « Authentification»

****

Figure : Diagramme de séquence pour « Authentification»

### III.3.2. Diagramme de séquence pour « ajouter au panier»

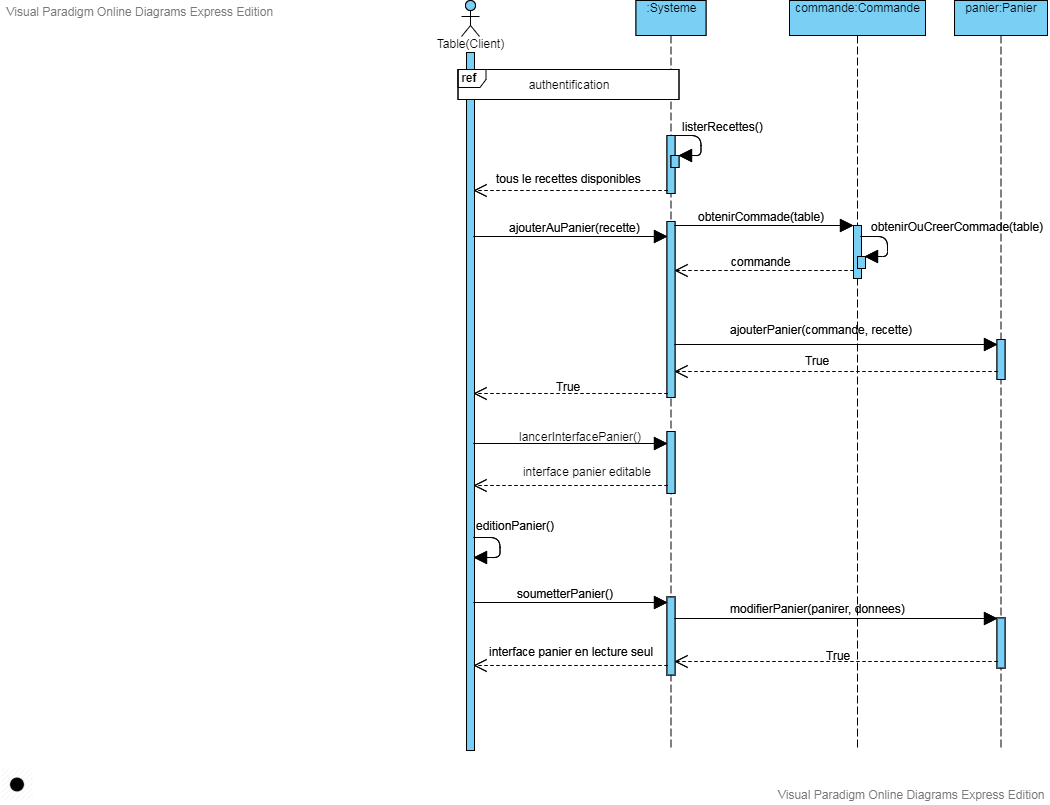


Figure : Diagramme de séquence pour « ajouter au panier»

### III.3.3. Diagramme de séquence pour « servir une commande»

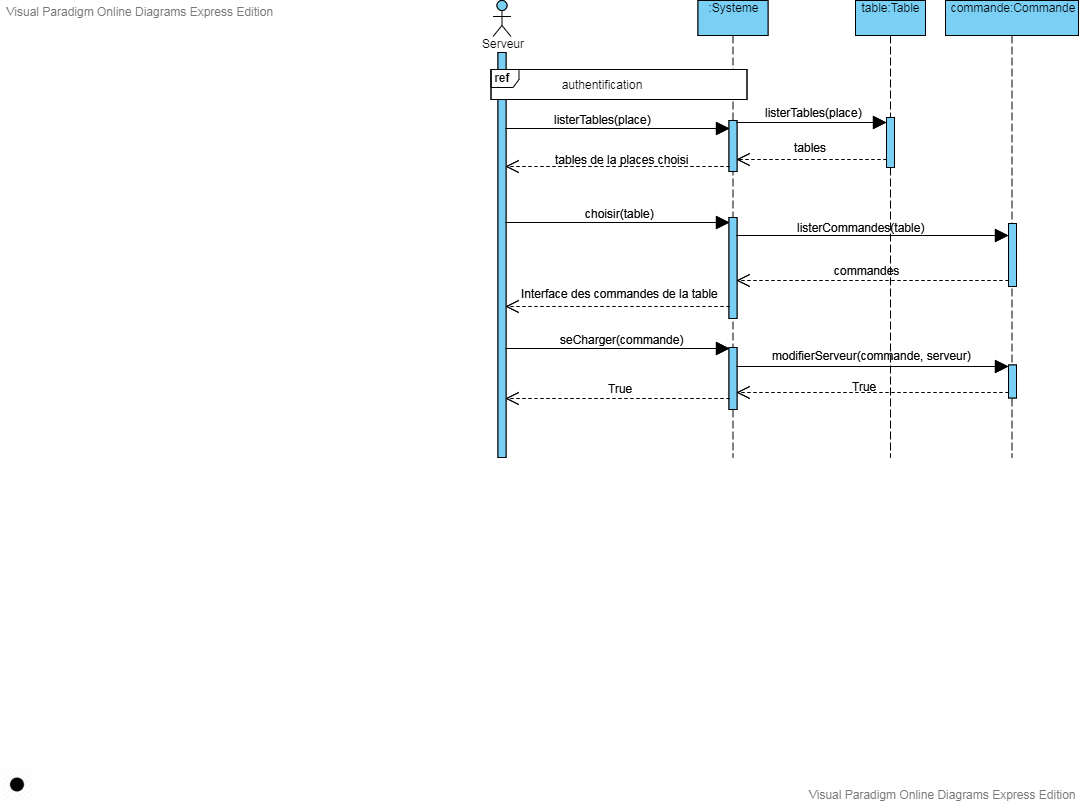


Figure : Diagramme de séquence pour « servire une commande»

### III.3.4. Diagramme de séquence pour « requisitionner un produit»

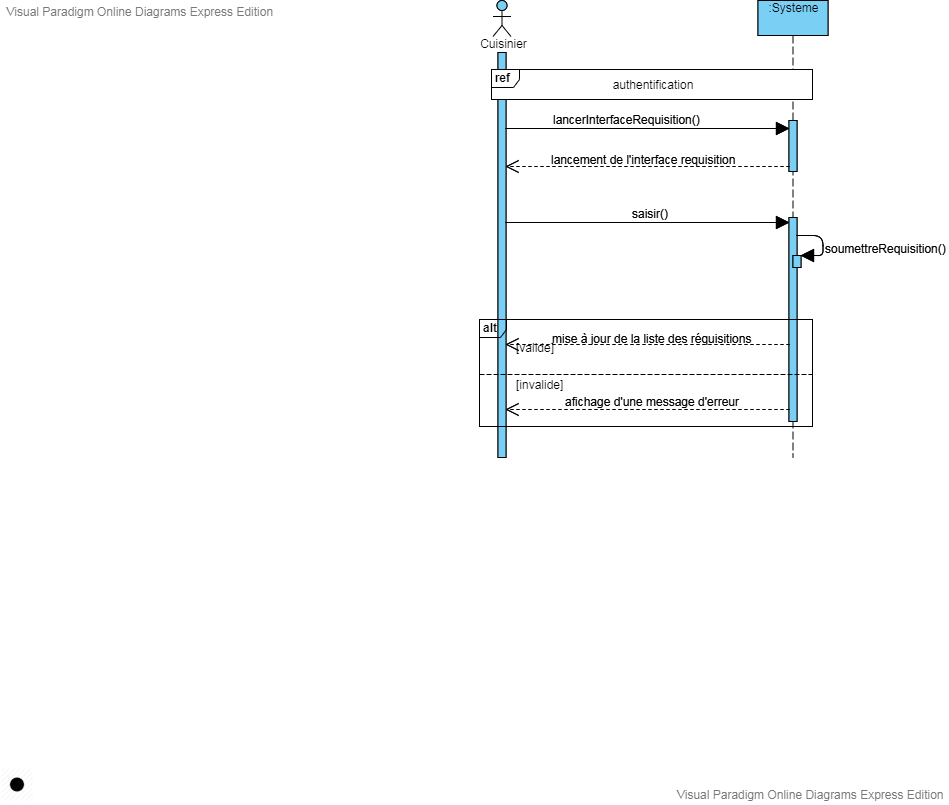


Figure : Diagramme de séquence pour « requisitionner un produit»

### III.3.5. Diagramme de séquence pour « payer la consomation»

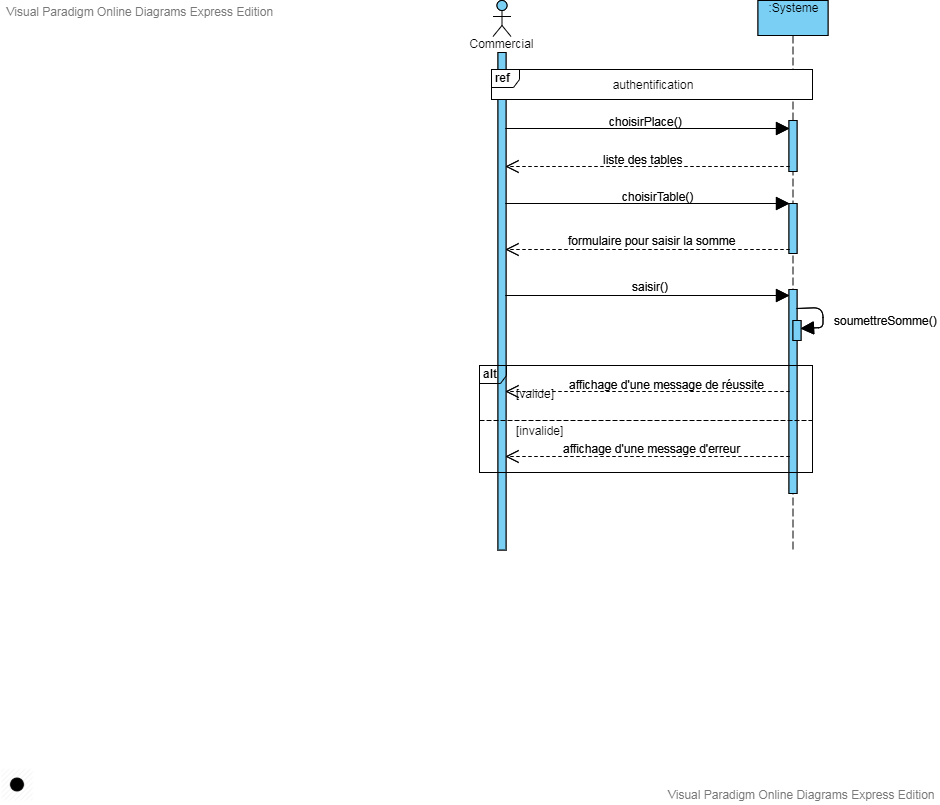


Figure : Diagramme de séquence pour « payer la consomation»

### III.3.6. Diagramme de séquence pour « s’approvisionner»

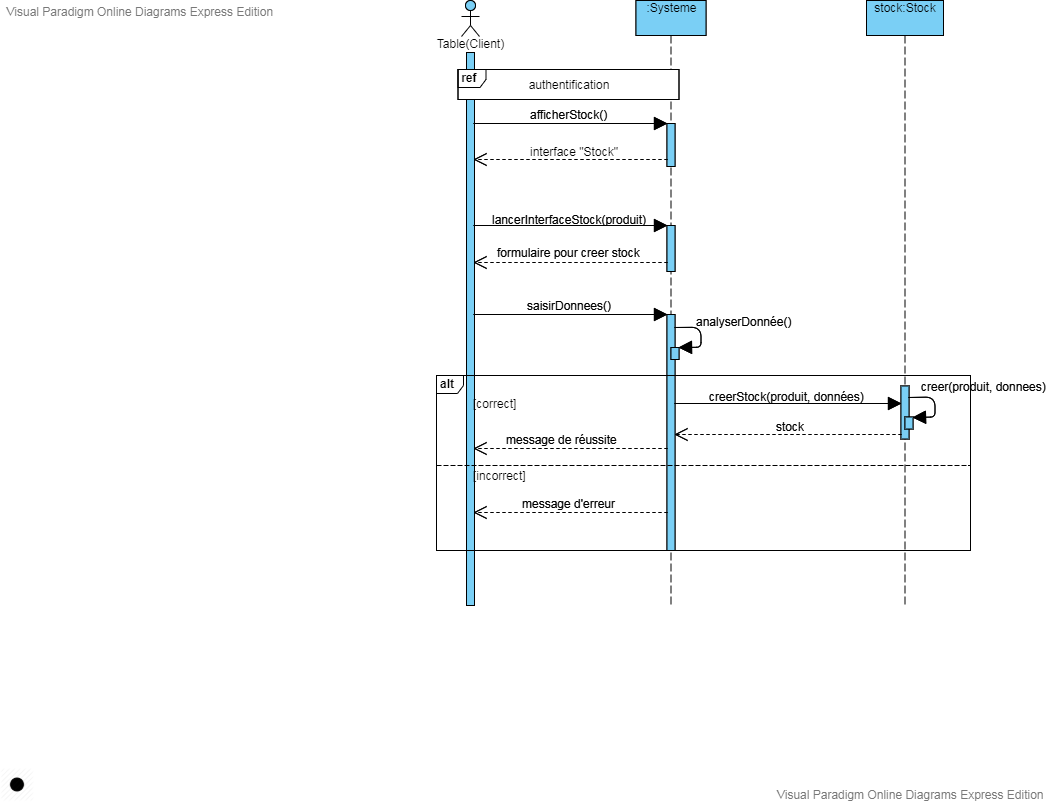


Figure : Diagramme de séquence pour « s’approvisionner»

## III.4. Diagramme de classe

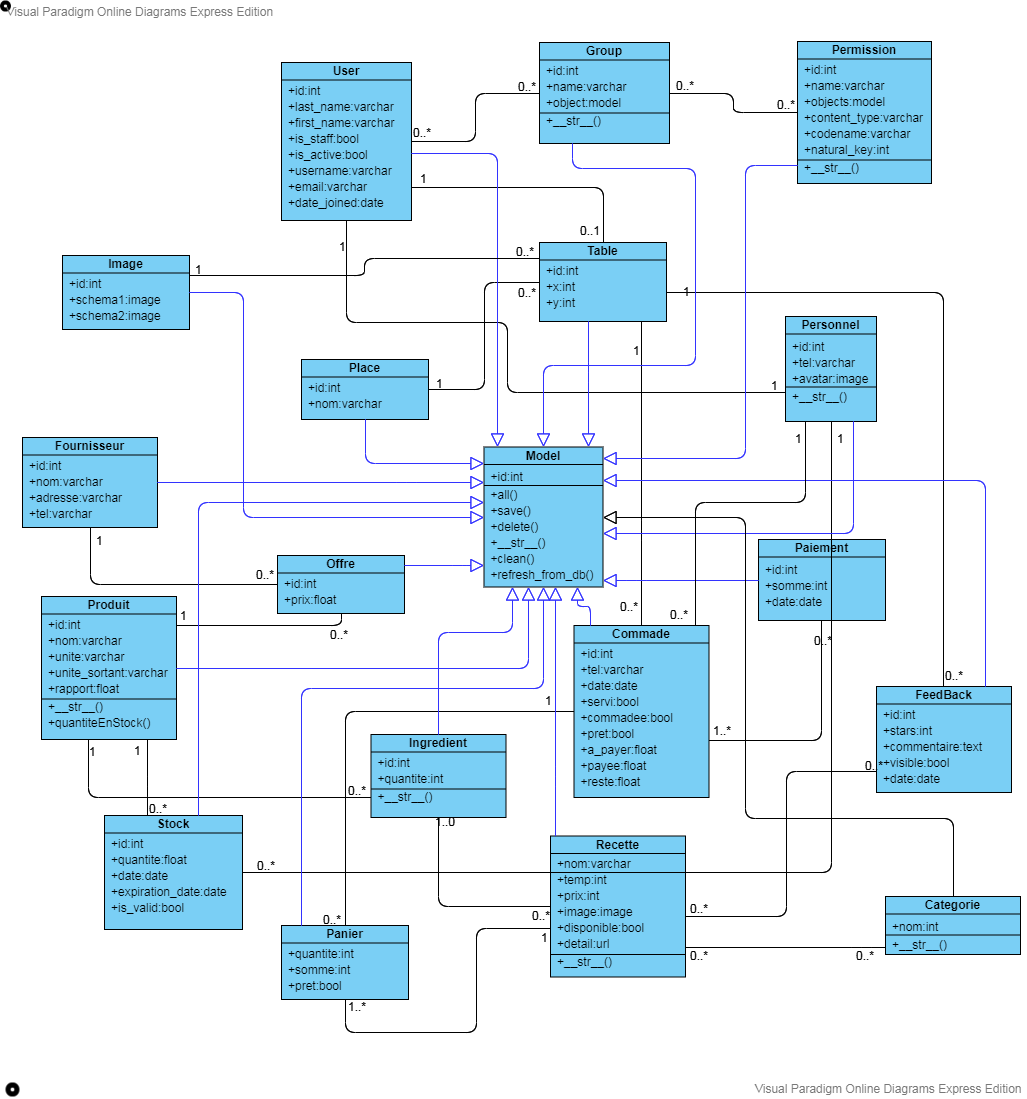
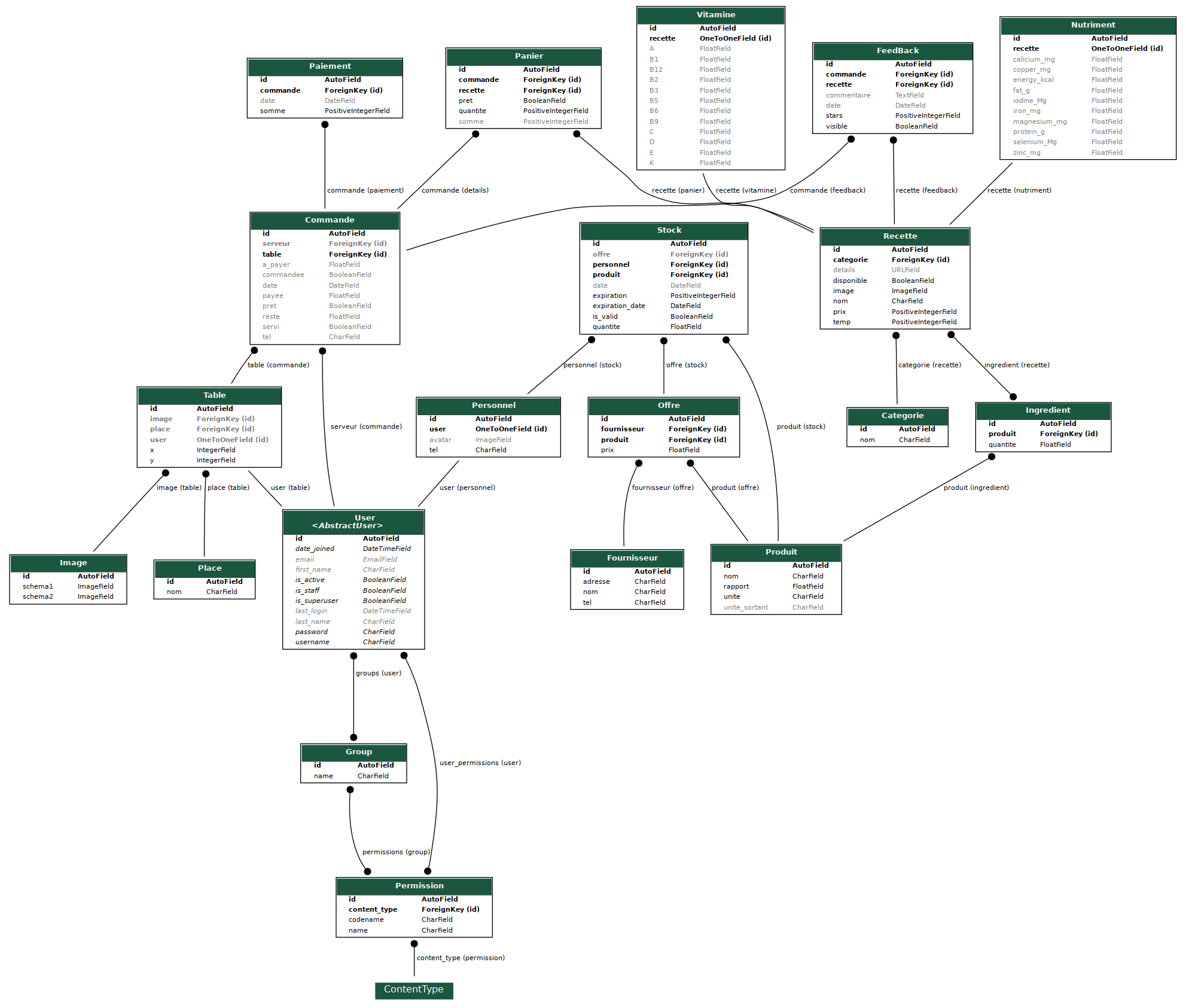


Figure : Diagramme de classe

## III.5. Model physique des donnés

****

### Tableau 2 : Dictionnaire de données

## III.6.Diagramme de déploiement

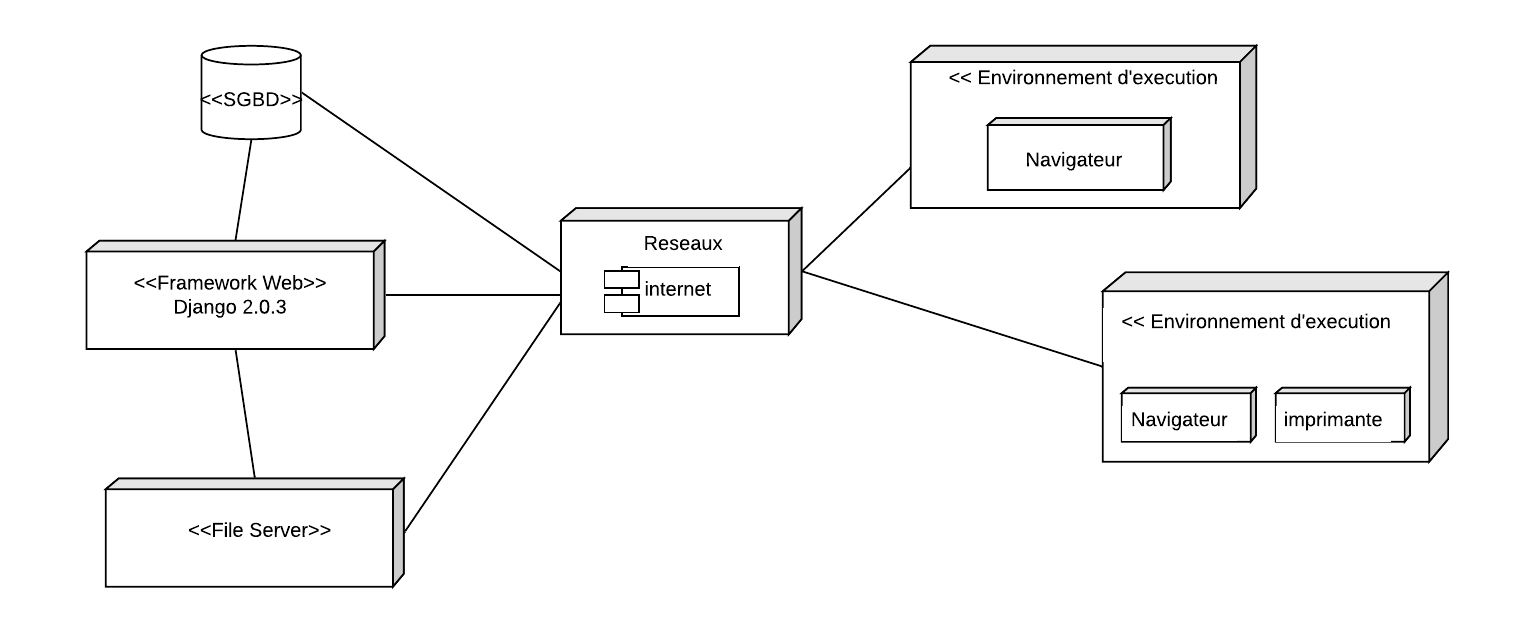


Figure : Diagramme de déploiement

# CHAPITRE IV: PRESENTATION DE QUELQUES PAGES DU SYSTEME …

## IV.1.Introduction

Dans ce chapitre, J’essaie de montrer quelques pages utilisateurs montrant le fonctionnement des applications composantes de systeme … Je tiens aussi à montrer les outils que j’ai utilisés au cours du développement de ce dernière et pourquoi.

## IV.2. Choix des outils et technologies utilisés

### IV.2.1. Au niveau conceptuel

Pour la conception de notre système d’information j’ai utilisé UML (expliqué au troisième chapitre) comme langage de modélisation vu que c’est un langage qui a pour but de faciliter les transitions, lors du développement d’une application informatique, du besoin originel à la phase d’implémentation, et visual paradigm online diagrams (Un logiciel de création de diagrammes tout-en-un qui prend en charge une grande variété de diagrammes commerciaux et techniques) accessible sur online.visual-paradigm.com.

### IV.2.2. Au niveau développement back-end

* + - 1. **Python**

Python créer par Guido van Rossum en 1991 est un langage de programmation interprété, multi-paradigme et multiplateformes. Il favorise la programmation impérative structurée, fonctionnelle et orientée objet. Il est doté d'un typage dynamique fort, d'une gestion automatique de la mémoire par ramasse-miettes et d'un système de gestion d'exceptions ; il est ainsi similaire à Perl, Ruby, Scheme, Smalltalk et Tcl.

* + - 1. **Le framework Django**

Un framework est un ensemble d'outils qui simplifie le travail d'un développeur. Traduit littéralement de l'anglais, un framework est un « cadre de travail ». Il apporte les bases communes à la majorité des programmes ou des sites web. Celles-ci étant souvent identiques, un développeur peut les réutiliser simplement et se concentrer sur les particularités de son projet.

Il s'agit donc d'un ensemble de bibliothèques coordonnées, qui permettent à un développeur d'éviter de réécrire plusieurs fois une même fonctionnalité, et ainsi d'éviter de réinventer constamment la roue. Ainsi le gain en énergie et en temps est considérable.

Django est un cadre de développement (framework) web open source en Python. Il a pour but de rendre le développement web 2.0 simple et rapide. Pour cette raison, le projet a pour slogan « Le framework pour les perfectionnistes avec des deadlines. ».

**Pourquoi Django?**

* Django prône le principe du « Don't repeat yourself », c'est-à-dire en français « Ne vous répétez pas », et permet le développement rapide de meilleures et plus performantes applications web, tout en conservant un code élégant
* Il est utilisé par de nombreux sites et organisations dont quelques sites majeurs d’internet citons : Instagram, Pinterest, disqus, bitbucket, l’admin d’open stak, , le département des sciences de la NASA, The Onion, Le New York Times, le Washington Post, Certaines parties de Google, International Watch Company, Art Gallery of NSW, Le Musée d'Art Contemporain d’Australie, Certaines parties d’IKEA, …
* Il est soutenu par une communauté active et expérimentée, qui publie régulièrement de nouvelles versions du framework avec de nouvelles fonctionnalités, des corrections de bugs, etc.
* Son architecture M.T.V (Model Template Vue, légèrement différent de MVC). La principale différence entre les deux architectures est que Django s'occupe lui-même de la partie contrôleur (code logiciel qui contrôle les interactions entre le modèle et la vue), nous laissant avec le template. Le template etant un fichier HTML mélangé avec Django Template Language (DTL). Le fonctionnement se resume avec le schema suivant :

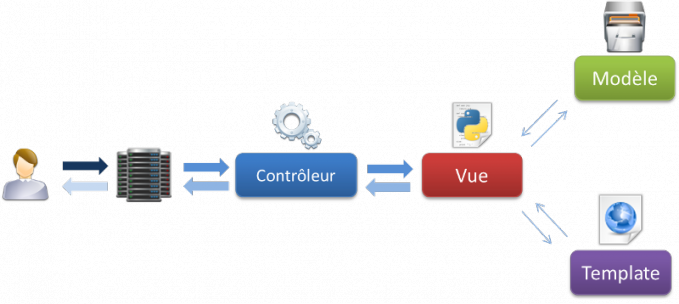


Figure : Django MVT

* Son ORM (Object relationnel mapper c’est-à-dire le programme qui se place en interface entre un programme applicatif et une base de données relationnelle pour simuler une base de données orientée objet.)
* Les formulaires automatiques et donc l’administration Django.

Et j’ajouterai que la conséquence de tout cela c’est : la rapidité du codage, un déploiement et une maintenance aisée.

Bien sur ce ne sont pas les uniques raisons d’utiliser Django, mais celles ci valent à elles seules vraiment la peine de regarder à quoi Django ressemble.

* + - 1. Django Rest Framework (DRF)

Le framework Django REST est une boîte à outils puissante et flexible pour la création d'API Web. Quelques raisons pour lesquelles j’ai utilisé le framework DRF:

* L'API navigable sur le Web est un énorme gain de convivialité pour vos développeurs.
* Stratégies d'authentification, y compris les packages pour OAuth1a et OAuth2.
* Sérialisation prenant en charge les sources de données ORM et non ORM (dans mon cas j’ai utilisé l’ORM native de Django).
* Personnalisable jusqu'en bas (possibilité d’utilisez simplement des vues basées sur les fonctions si vous n'avez pas besoin des fonctionnalités les plus puissantes).
* Documentation complète et grande assistance communautaire.

Utilisé et approuvé par des sociétés internationalement reconnues telles que Mozilla, Red Hat, Heroku et Eventbrite.

### IV.2.2. Au niveau développement front-end (client)

1. HTML/CSS

Ce sont les langages les plus utilisées pour permettre la création des sites web. Tous les sites web sont basés sur ces langages, ils sont incontournables et universels aujourd’hui.

Les langages HTML (HyperText Markup Langage) et CSS (Cascading Style Sheets, aussi appelées Feuilles de style) sont à la base du fonctionnement de tous les sites web, le navigateur fera la traduction entre ces langages informatiques et ce que nous voyons affiche à l’écran. Ces deux langages se complètent et ont des rôles différents à jouer pour pouvoir construire une page web.

1. Twitter Bootstrap

Bootstrap est un framework CSS qui comporte un système de grille simple et efficace pour mettre en ordre l'aspect visuel d'une page web. Il apporte du style pour les boutons, les formulaires, la navigation… Il permet ainsi de concevoir un site web rapidement et avec peu de lignes de code ajoutées

Car les navigateurs n'adoptent pas tous les mêmes valeurs par défaut pour les styles des éléments HTML. Ce qui peut générer quelques surprises au rendu des pages web selon le navigateur utilisé. D'autre part certains navigateurs présentent des défauts de prise en compte de certains éléments.

Bootstrap propose :

* Une mise en page basée sur une grille de 12 colonnes bien pratique. Bien sûr, si vous avez besoin de plus de 12 colonnes, ou de moins, il est toujours possible de changer la configuration ;
* La normalisation ;
* Du code fondé sur HTML5 et CSS3 ;
* Une bibliothèque totalement open source sous [license MIT](https://github.com/twbs/bootstrap/blob/master/LICENSE);
* Du code qui tient compte du format d'affichage des principaux outils de navigation (responsive design) : smartphones, tablettes… ;
* Des plugins jQuery de qualité ;
* Un résultat cross-browser (la prise en charge de Internet Explorer 7 a été abandonnée avec la version 3), donc une garantie de compatibilité maximale ;
* Une bonne documentation ;
* La garantie d'une évolution permanente ;
* Une mine de ressources variées sur le web ;

1. JavaScript et jQuery

**JavaScript** est avant tout le langage de programmation du Web. Il a été inventé en 1995 par [Brendan Eich](https://fr.wikipedia.org/wiki/Brendan_Eich), qui travaillait à l'époque pour la société [Netscape](https://fr.wikipedia.org/wiki/Netscape_Communications), créatrice du premier navigateur Web populaire (l'ancêtre de Firefox). Le langage JavaScript a été [inventé](https://fr.wikipedia.org/wiki/JavaScript#Histoire)  pour dynamiser les sites Internet coté client. Il a beaucoup évolué depuis, au point de [devenir essentiel](https://medium.com/@bpesquet/pourquoi-javascript-est-devenu-essentiel-5eb7f82cc6f4#.yxtyaolpj), mais il a toujours gardé ce rôle central de langage de programmation du web.

**jQuery** est une bibliothèque qui permet d'agir sur le code HTML, CSS, JavaScript et AJAX. jQuery permet de manipuler les éléments mis en place en HTML (textes, images, liens, vidéos, etc.) et mis en forme en CSS (position, taille, couleur, transparence, etc.) en utilisant des instructions simples qui donnent accès aux immenses possibilités de JavaScript et d'AJAX.

JavaScript étant certes puissants son syntaxe et vraiment très verbeux ce qui fait que toute erreur insignifiante dans la syntaxe provoque généralement la non-exécution de l'instruction correspondante et ainsi souvent nécessaire d'écrire de nombreuses lignes pour faire un simple petit traitement. Aussi les navigateurs n'adoptent pas tous les mêmes valeurs par défaut pour interpreter le code JS. Ce qui peut générer quelques surprises au rendu des pages web selon le navigateur utilisé. D'autre part certains navigateurs présentent des défauts de prise en compte de certains éléments.

La devise de jQuery etant « *Write less, do more* », ce qui signifie « Écrivez moins pour faire plus ». Cela devrait convenir à bon nombre de programmeurs dont moi. En effet, en écrivant moins de code, les erreurs sont moins fréquentes. Sans oublier aussi que jQuery permet aussi d’obtenir un résultat cross-browser, donc une garantie de compatibilité maximale.

## IV.3. Présentation de quelques pages de l’application

### Page pour authentification

Cette page permet aux différents utilisateurs de s’authentifier notamment les administrateurs, les cuisiniers, les serveurs et les comercials simples afin que chacun d’eux soit orienté à la page qui lui concerne.

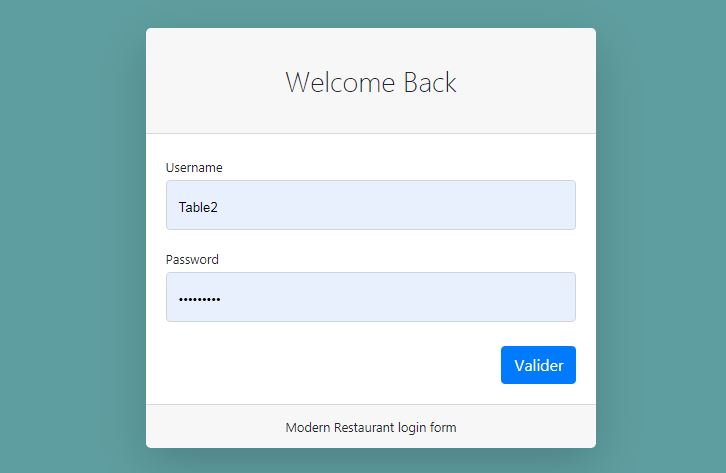


Figure : Page d’authentification

### Page pour administrateurs

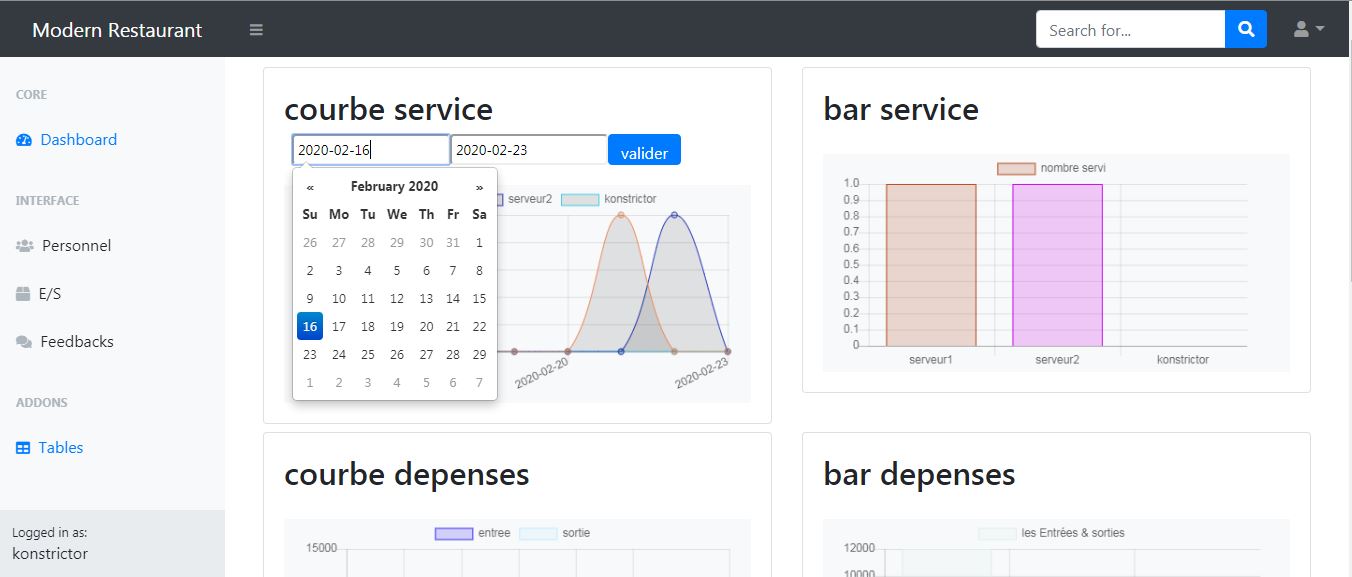


Figure : page d'administration 1

Cette page est une interface pour visualiser differents courbes statistique pour avoir l’idée claire de toute l’activité du restaurant

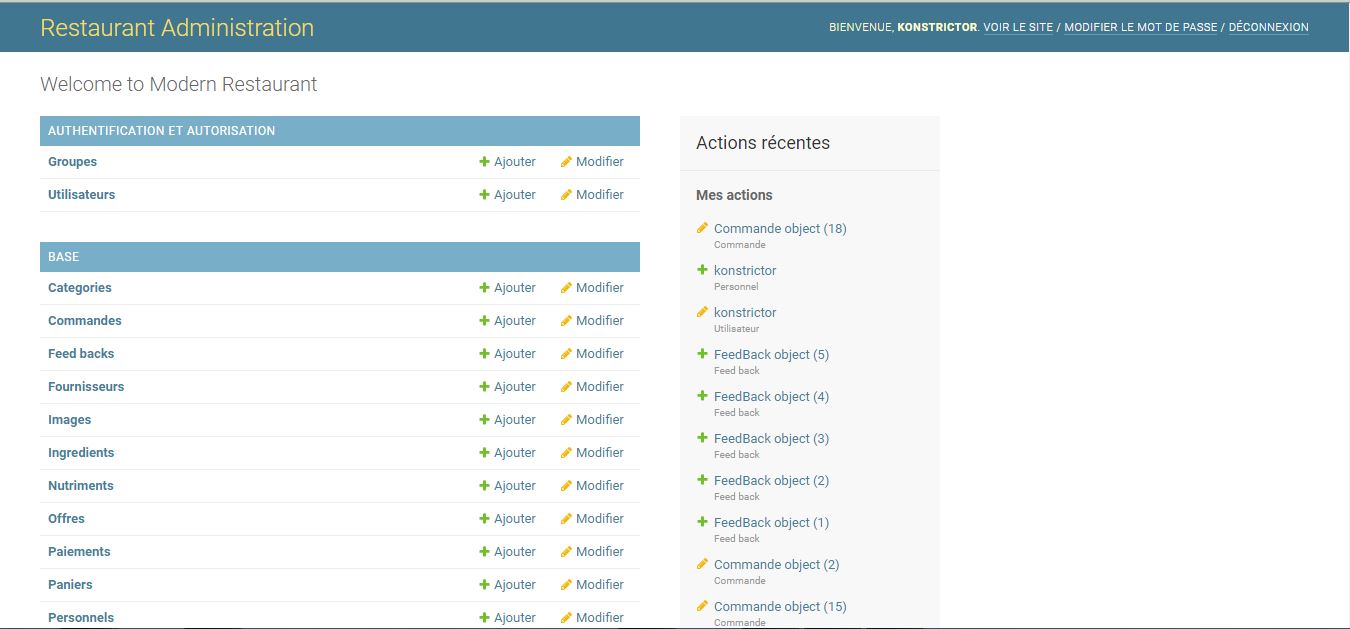


Figure : page d'administration 2

Cette page est une interface génerée par django à base des personalisations faites dans le fichier admin.py pour chaque application. Et c’est ici ou l’administrateur fait le reste tout ce à quoi il a droit, à savoir Ajouter, modifier, supprimer le personnel, les permissions, les fournisseurs, les offres, les places, les produits, les recettes, les stocks et tables.

### Page regler une facture

C’est sur cette page le commercial choisi une table et voit la somme total de ce que le client a consomé. Le commercial n’aura qu’à saissir la somme payée sans tenir compte du reste. Le système lui-même calcule la somme restante.

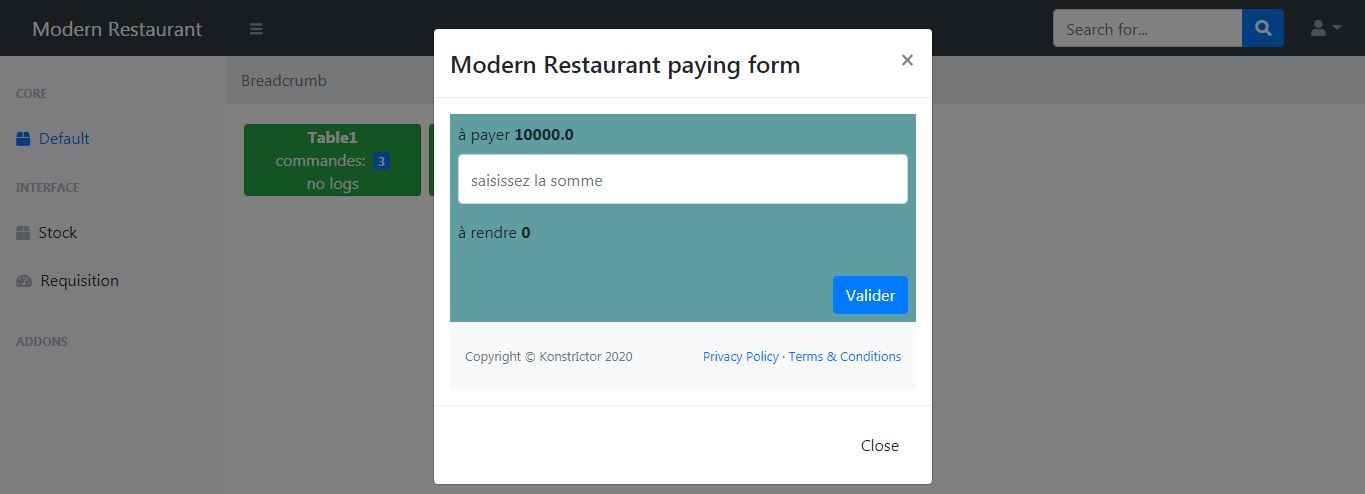


Figure : Page pour regler une facture

### Page pour Stock

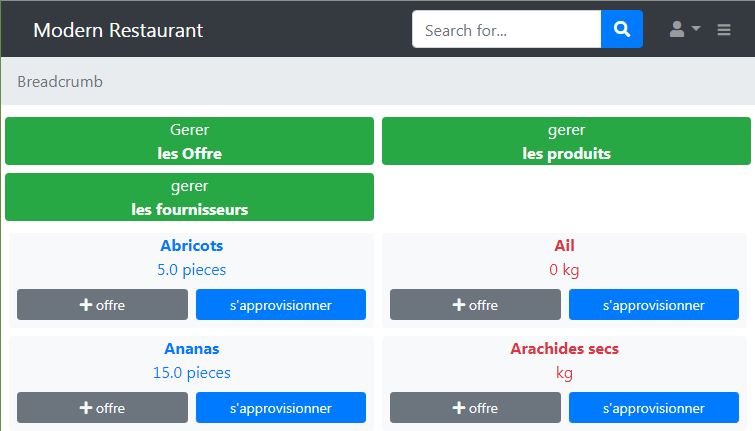


Figure : Page pour Stock

Cette affiche l’état du stock. Si le commercial clique sur un produit l’historique d’entrée et de sorti s’affiche, si il clique sur le formulaire d’ajout d’offre s’affiche. S’il clique sur la page d’aprovissionnement s’affiche.

+offre

s’approvisionner

### Page pour approvisionnement

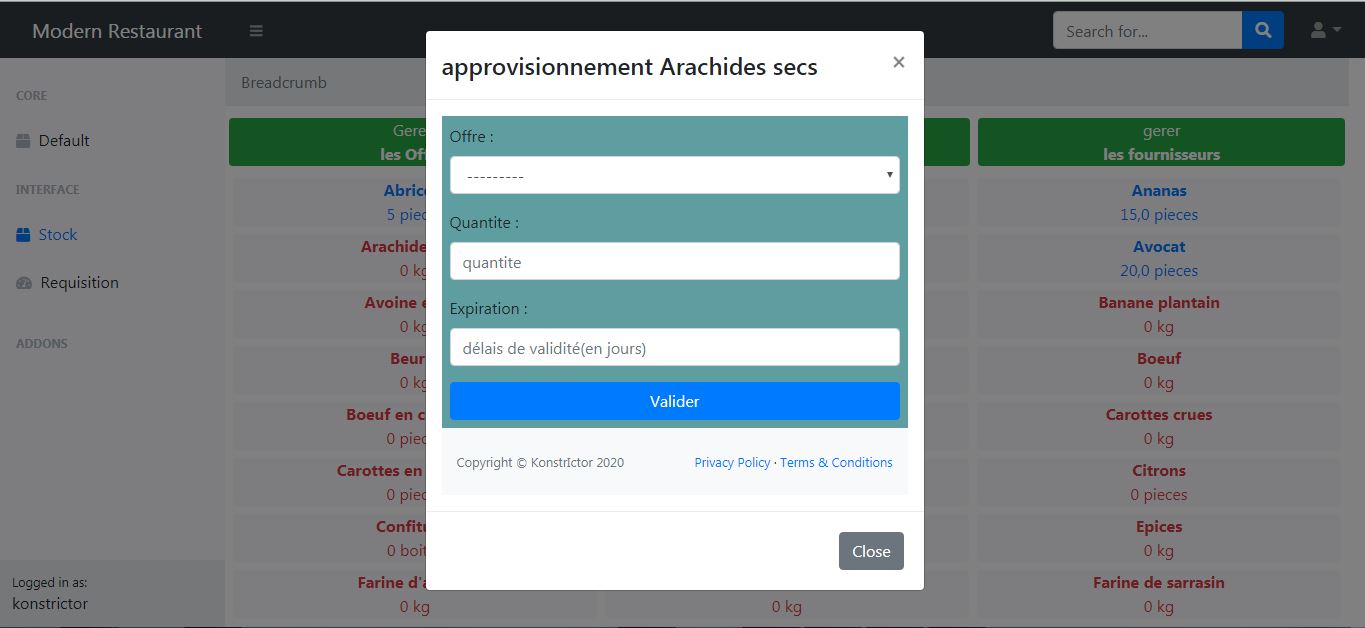


Figure : Page pour approvisionnement

C’est sur cette page que le commercial saisisse les informations sur l’aprovivionnement.

### Page pour effectuer une requisition

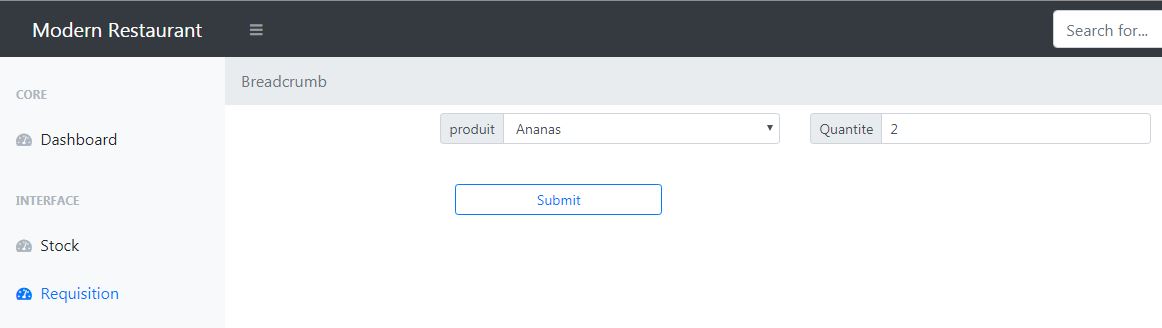


Figure : Page pour effectuer une requisition

Cette interface est une interface se trouvant dans l’aplication du cuisinier. Elle est faite d’un formulaire pour choisir un produit et saisir la quantité qu’il veut prendre du stock. . Il faut noter noter que le cuisinier peut ajouter d’autres formulaires. Elle comporte aussi une liste des requisition n’ayant pas encore été approuvé.

### Page pour approuver une requisition

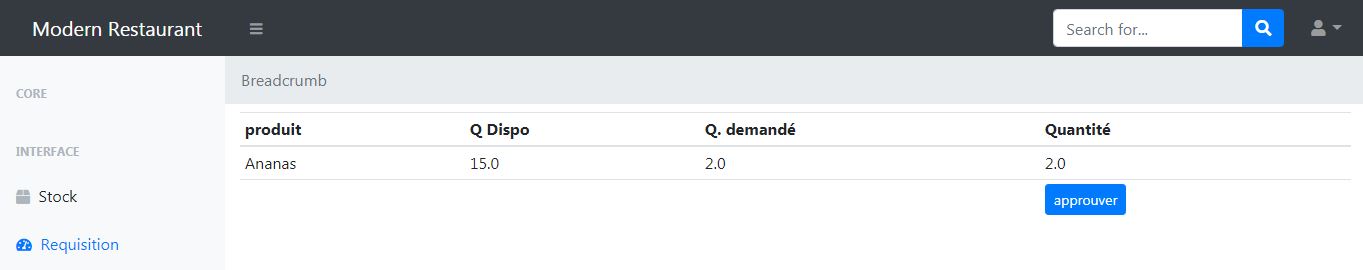


Figure : Page pour approuver une requisition

C’est sur cette page que le commercial valide les requisitions éffectuées par les cuisiniers. La requisition ici étant l’acte de prendre les produits du stock vers la cuisine.

### Page pour visualier les recettes et les ajouter au panier

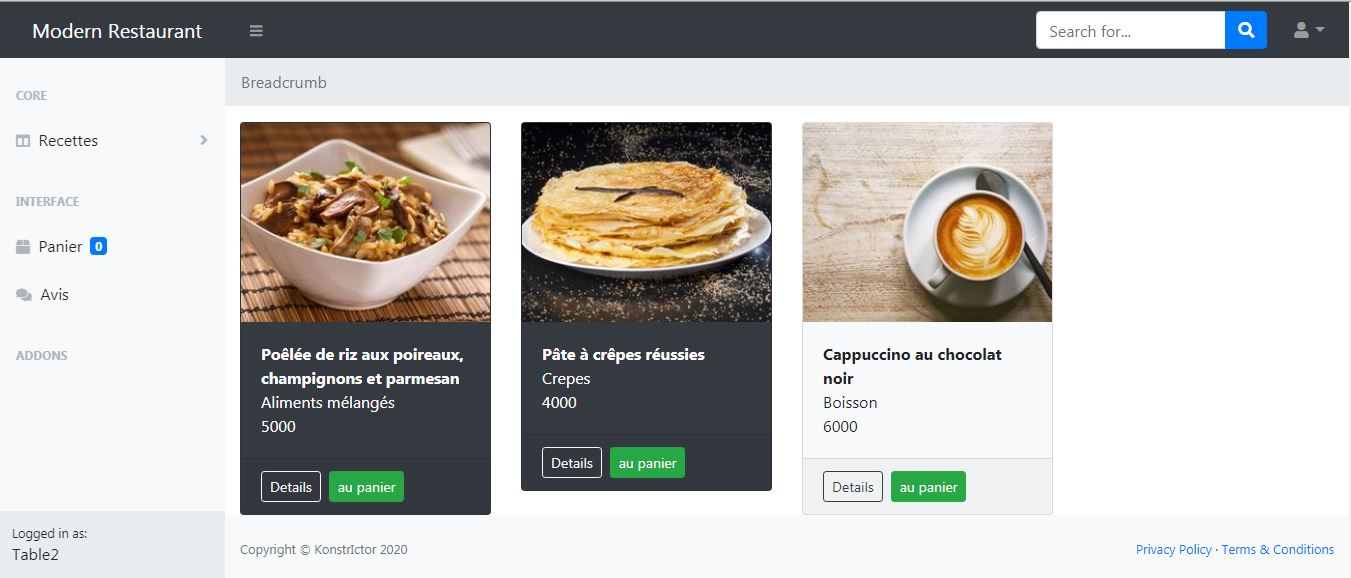


Figure : Page pour visualier les recettes et les ajouter au panier

Cette page est la page d’accueil pour les tables. Et permet au client sur la table de mettre les recettes voulues au panier mais aussi de visualiser les details s’il veut.

### Page pour personnaliser le panier

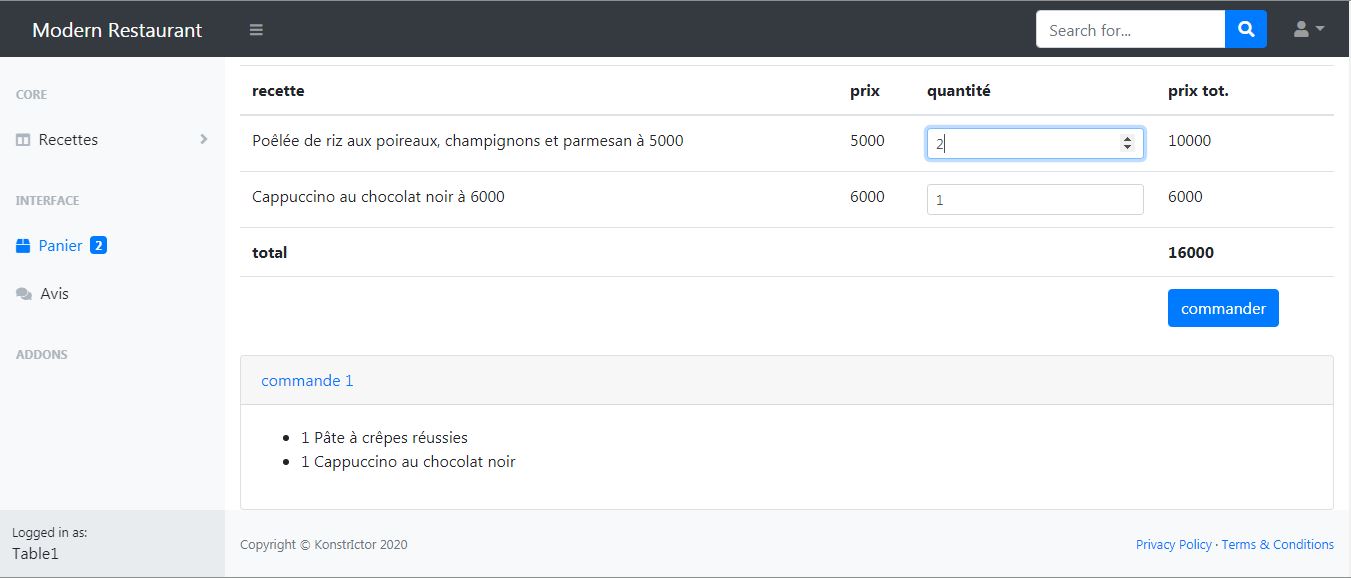


Figure : page pour personnaliser le panier

C’est sur cette page que le client retrouve la possibilité de regler son panier à savoir modifier la quantité et enlever les recettes non voulues ou ajouté au panier par erreur. Il n’y a pas de boutton supprimer, pour ce l’autisateur met la quantité voulue à zero et le système saura qu’il ne plus la recette. Si l’utilisateur decide de commander, il n’aura plus la possibilité de changer la commande, et les recettes composantes de la commande apparaissent separement directement dans la page d’accueil du cuisinier.

### Page d’accueil pour les serveurs

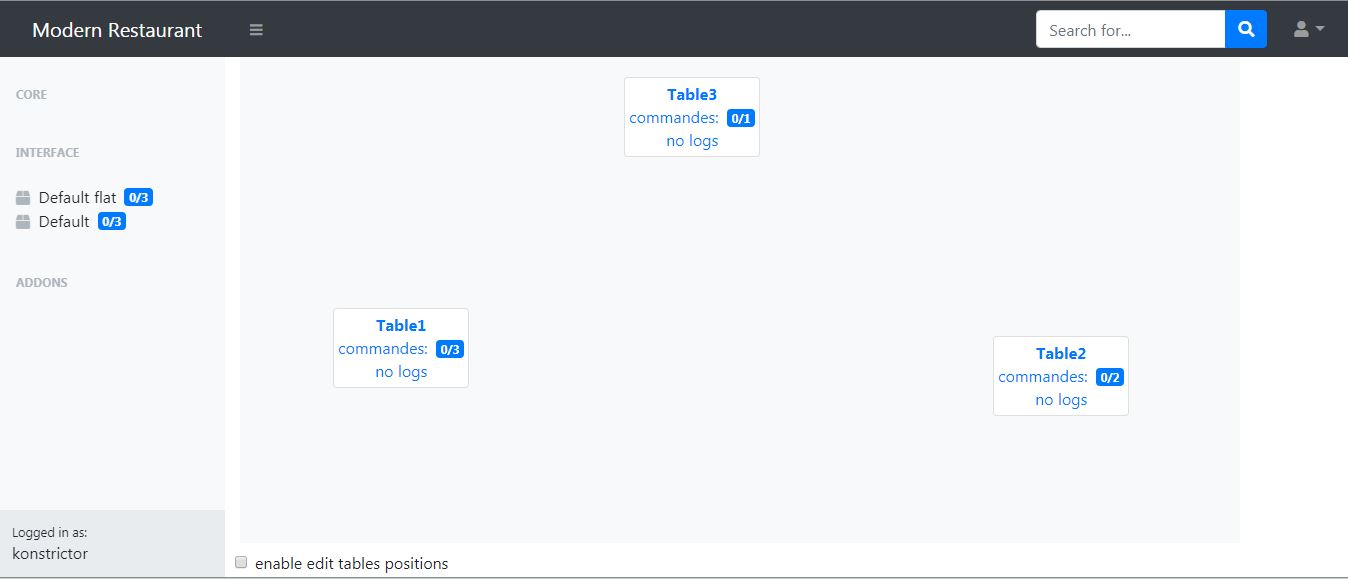


Figure : Page d’accueil pour les serveurs

Sur cette page le serveur voit dans le menu à gauche la liste des places disponible. Chaque place est dupliquée : un pour menu donne les tables sans tenir compte de leurs disposions, une autre tient compte de leurs emplacement par rapport à la disposion des tables dans la place donné, les positions sont editable par glisser-deposer.

### Page pour se charger de servir une commande

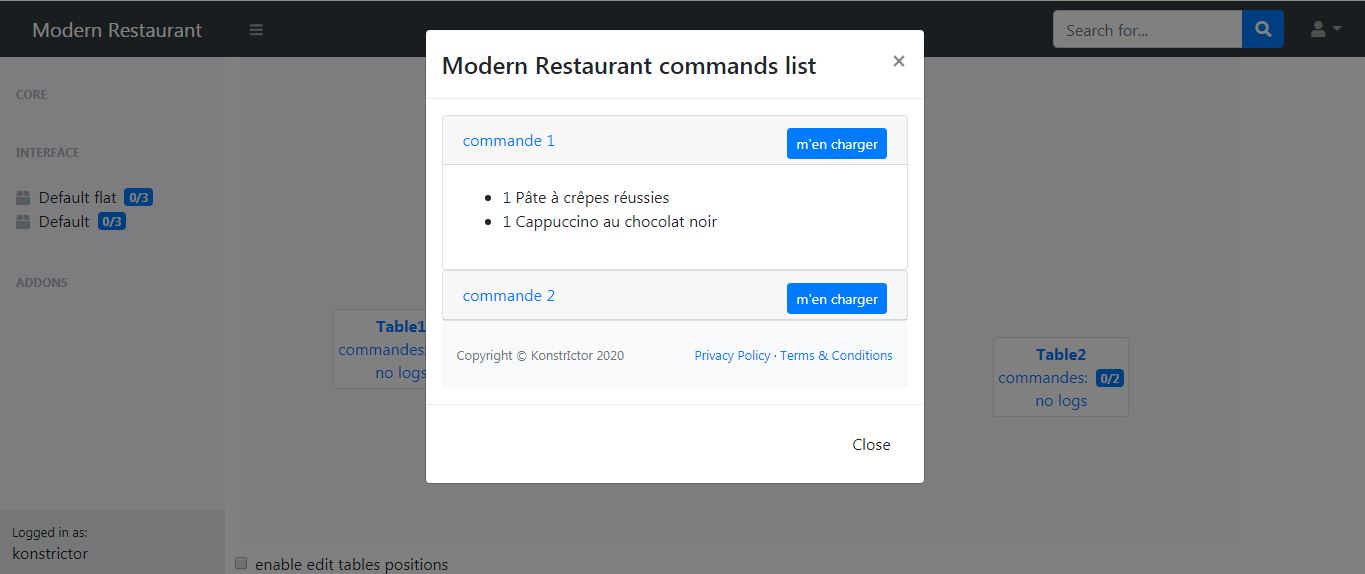


Figure : Page pour se charger de servir une commande

L’interface presente est une interface qui apparait quand un serveur clique sur une table. Dans cette interface le serveur remarque la liste des commandes éfectuées par le client avec la possibilité de visualiser la liste des recettes qui les compose. Ici alors, le client peut se charger de servir une commande, et aissi il recevra un signal lorque la commande sera prête.

### Page d’acceil pour les cuisiniers

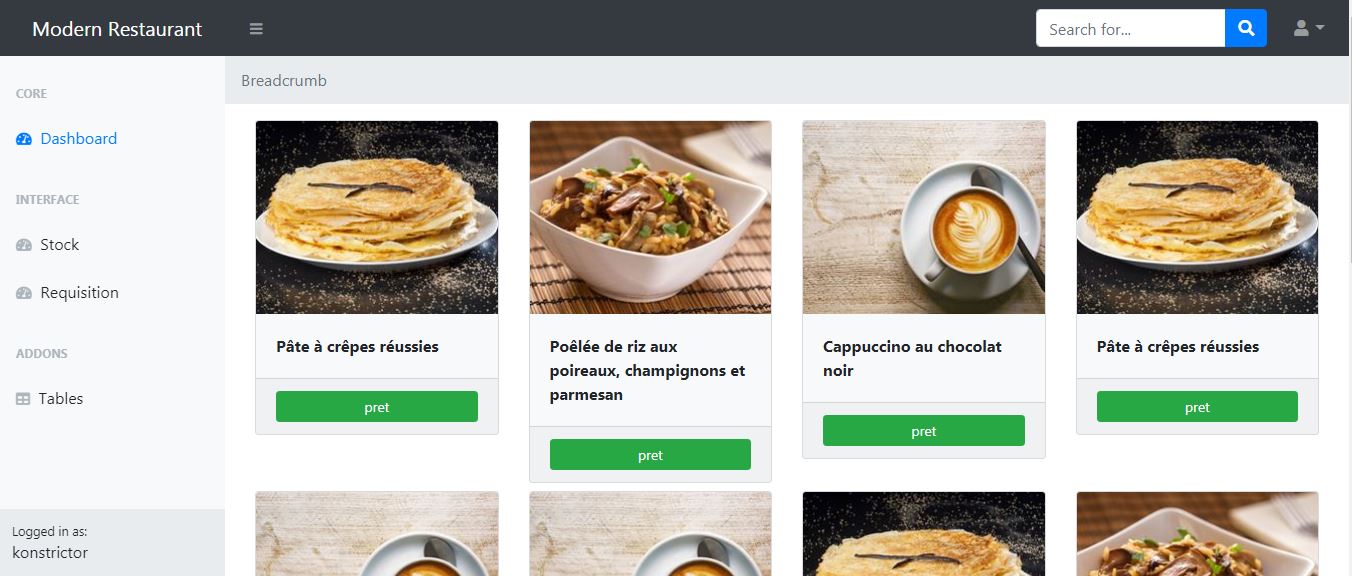


Figure : Page d’acceil pour les cuisiniers

Cette page est une page à deux roles principaux. C’est ici qu’un cuisinier trouve la liste des recettes commadées en partant de la commande le plus ancien au plus recent. C’est aussi ici que le cuisinier retrouve la possibilité de se charger de preparer une recette et de marquer qu’elle est prête. Si toutes les recettes composantes d’une commande sont prêtes le système se chargera de notifier le serveur qui s’en est chargé. Si personne ne s’en est chargé tout les serveurs sont notifés.

# IV.4. Critiques sur les résultats obtenus

Une application n’étant jamais achevée, nous nous battons toujours jusqu’au fond avec fermeté pour avoir abouti à la conception et réalisation de l’application répondant aux attentes que nous nous sommes fixées tout au début de ce travail. Les erreurs de programmation étant toujours omniprésentes dans toute application, nous admettons que notre application ne fait pas exception à cette règle qui tend à se généraliser bien que jusqu’à présent nous n’avons pas encore observé de disfonctionnements.

1. https://fr.wikipedia.org/wiki/Informatique [↑](#footnote-ref-1)