

INSTITUT FRANCOPHONE INTERNATIONAL



PROJET DE GENIE LOGICIEL AVANCE « GESTION DE CAFETERIA »

Réalisé par : **BIAYA TSHIBANGU** Emmanuel
MILORME PIERRE Rubens

Superviseur: **Mr. HO Tuong Vinh**

Hanoi, 2017

I. Introduction

Le domaine de la restauration est en constante évolution, les gens sont toujours à la recherche d'innovation. C'est pourquoi nous avons choisie d'effectuer ce projet avec les connaissances acquises pendant les enseignements du cours de génie logiciel. La cafétéria s'adresse surtout à des étudiants, des passants et des internés du dortoir KTX My Dinh qui aiment manger de la cuisine moderne et goûter des mets nouveaux. La cafétéria propose une spécialité de la cuisine vietnamienne dans toutes ces formules. Ainsi partant de tout ce qui précède, étant qu'informaticien, nous allons mettre en place une application web dynamique afin de faciliter la visibilité de la cafétéria sur le net et de faciliter la gestion de transaction interne de vente des plats dans la cafétéria.

II. Objectifs

La conception de notre application tourne autour de quelques objectifs à savoir :

1. Le site devra s'afficher de la même façon sur tous les navigateurs sans aucune contrainte ;
2. La page d'accueil devra fournir des informations sous forme d'image des plats de la journée en défilement de droite à gauche ;
3. Sur la page doit s'afficher l'authentification permettant à l'administrateur et aux utilisateurs d'y accéder ;
4. L'administrateur détient le total contrôle du système sur les fonctions telle que (la suppression, la modification, création des abonnés et des utilisateurs).

III. Spécification des exigences

Notre système consiste à fournir des annonces des différents plats dans la cafétéria sous forme d'image défilent sur la page d'accueil. Sur la même interface nous retrouvons un système de sécurité (authentification) permettant la manipulation de différentes fonctionnalités de notre système. Seul l'administrateur du système et l'utilisateur qui est le caissier peuvent y accéder.

Les spécifications des exigences fonctionnelles du système se tourne sous trois axes principaux dont qui constitue le cœur de notre système, il s'agit donc de la :

- Gestion des utilisateurs ;
- Gestion de paiement ;
- Gestion des abonnés.

III.1. Liste des exigences

Nous fournissons d'une part les exigences fonctionnelles d'internaute au niveau interne de la cafétéria et de la visibilité au niveau externe des clients constituant les fonctionnalités non-fonctionnel.

Au niveau interne nous avons :

- Les utilisateurs : capacité pour un utilisateur d'exécuter une tâche dans un temps déterminé
 - Performance : Temps d'attente inférieur à 20 seconde ;
 - Sécurité : Accès personnalisés, connexions sécurisées ;
 - Portabilité : Utilisable avec plusieurs systèmes d'exploitation.

III.2. Besoins

Dans ce projet nous avons comme finalité :

- Enregistrer les abonnés en leur fournissant la carte avec un numéro ID ;
- Créer les différents utilisateurs pour la vente ;
- Gérer la transaction de paiement des clients abonné et non abonné ;

IV. CONCEPTION ET IMPLEMENTATION

IV.1. Introduction

Nous ne pourrions pas aborder les étapes de notre système sans pour autant démontré les pratiques du développement conceptuel de l'application Web. Un modèle conceptuel est en effet une représentation abstraite d'un système destiné à en faciliter l'étude et à le documenter. Le modèle présente notamment l'atout de faciliter la traçabilité du système, à savoir la possibilité de partir d'un de ses éléments et de suivre ses interactions et ses liens avec d'autres parties du modèle.

Une bonne conception réside en un bon choix de la méthode de modélisation puisqu'un modèle est établi pour traduire les besoins du concepteur et il prend sa véritable dimension lorsqu'il permet de communiquer sans ambiguïté.

Nous allons détailler la conception et l'analyse de chaque module afin de faciliter le développement de projet.

IV.2. Les acteurs

Un acteur représente le rôle d'une entité externe (utilisateur humain ou non) interagissant avec le système. Il est représenté par un bonhomme en fil de fer (en anglais stick man). Pour notre application, nous n'aurons que deux acteurs principaux qui vont pouvoir utiliser le système. Nous aurons l'administrateur qui prend le contrôle total du système, puis l'utilisateur du système qui a droit à quelques fonctionnalités partielles du système pour pouvoir accéder à l'interface protégée avec des tâches réduites.

IV.3. Diagramme de cas d'utilisation

Le diagramme ci-dessous représente une vue générale du projet, nous allons présenter les différentes tâches pour l'administrateur et l'utilisateur qui ont une main direct dans le système.



Figure 1 : Diagramme de cas d'utilisation

IV.4. Diagramme des séquences

Le diagramme des séquences permet de représenter des collaborations entre objets selon un point de vue temporel, en tenant compte de la chronologie des envois de messages.

Nous allons présenter le diagramme des séquences afin de bien illustrer tous les cas d'utilisations. Ainsi comme la taille de tous ces cas d'utilisation et le fait qu'ils effectuent les mêmes opérations, nous allons illustrer qu'un seul cas d'utilisation du système et le diagramme des séquences pour les autres cas d'utilisation restent identiques à celui-ci. Cas d'utilisation de gestion des annonces :

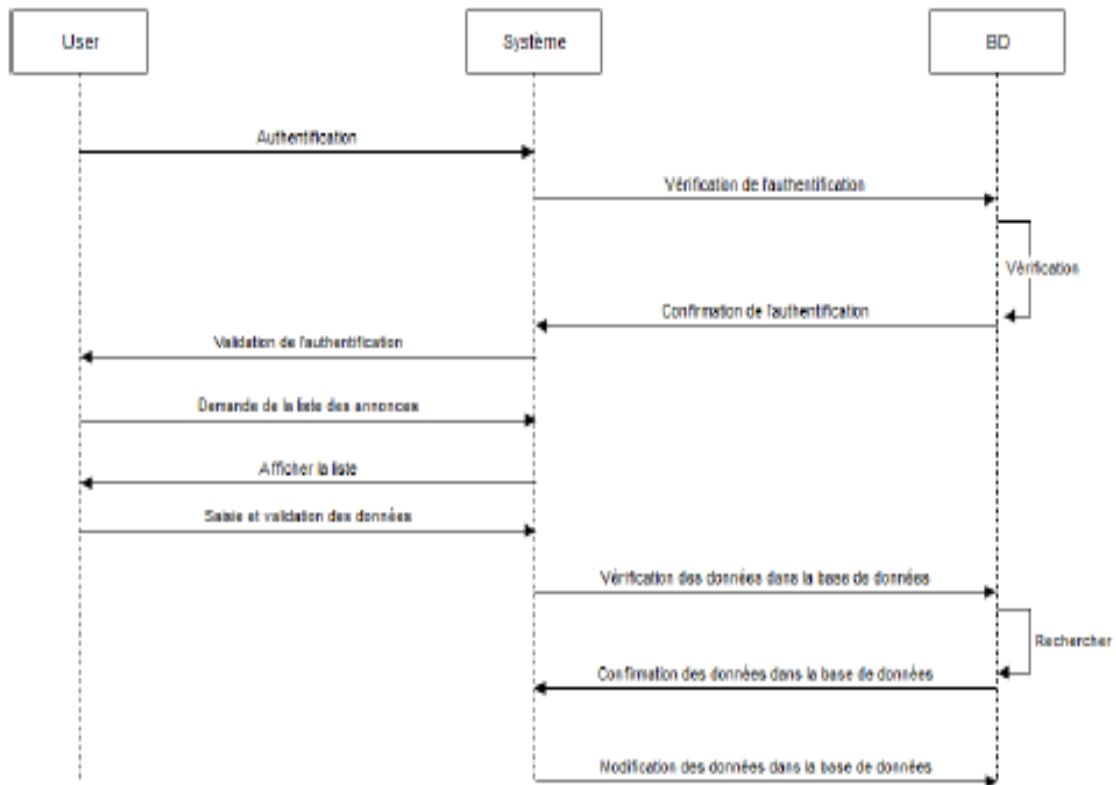


Figure 2 : Diagramme de séquence

IV.5. Diagramme de classe

Le diagramme de classes est généralement considéré comme le plus important dans un développement orienté objet. Il représente l'architecture conceptuelle du système : il décrit les classes que le système utilise, ainsi que leurs liens, que celles-ci représentent soit une relation d'héritage ou soit une relation d'agrégation ou soit une simple relation.

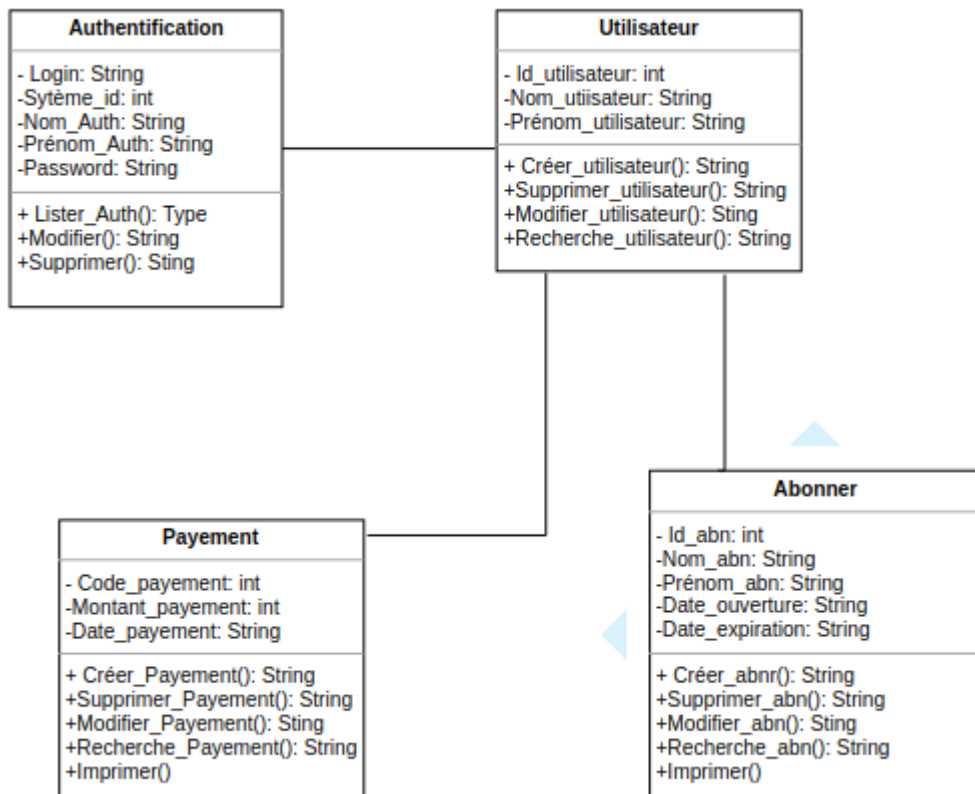


Figure 3 : Diagramme de classe

IV.6. La base de données du Projet

Le passage du modèle objet vers le modèle relationnel peut se faire automatiquement avec le SGBD PostgreSQL. Le résultat obtenu doit être finalisé ensuite selon l'application. Ce passage peut se faire avec plus de détails et précision manuellement.

Les règles de passage sont :

- Les classes sont transformées en modèle conceptuel (entités associations).
- chaque classe donne lieu à un type d'entité ou un type d'association, les attributs de la classe sont les propriétés des entités.
- les méthodes de la classe sont enlevées, il y a un traitement des identifiants. Si a une clé candidate, elle devient l'identifiant de l'entité.
- Si non un identifiant arbitraire est ajouté aux propriétés de l'entité ni l'héritage se concrétise par la création d'une

table par classe. Dans une relation d'héritage, l'objet père est représenté en table.

- L'objet fils hérite de la clé primaire et des attributs de l'objet père y rajoutant ses propres attributs.
- Le modèle conceptuel est traduit en modèle logique (modèle relationnel).
- Traduire chaque entité en association en table selon les cardinalités.
- Définir la série de contraintes d'intégrité référentielles.
- Le modèle relationnel est implémenté physiquement.
- Implémenter la base.

En vérifiant les contraintes d'intégrité référentielles et on obtient le schéma relationnel de la base de données suivant :

- o **Authentication** (id_Authen, log, nom, prénom, password) ;
- o **Utilisateur** (id_utilisateur, nom, prénom, ville, adresse, email) ;
- o **Payement** (id_payement, code_payement, date_payement) ;
- o **Abonné** (id_abonné, code_Abonné, prénom, date_d'ouverture, date_fermeture) ;

IV.7. Architecture du système

IV.7.1. Modèle MVC

Pour réaliser notre projet nous avons choisi d'utiliser l'architecture du modèle MVC.

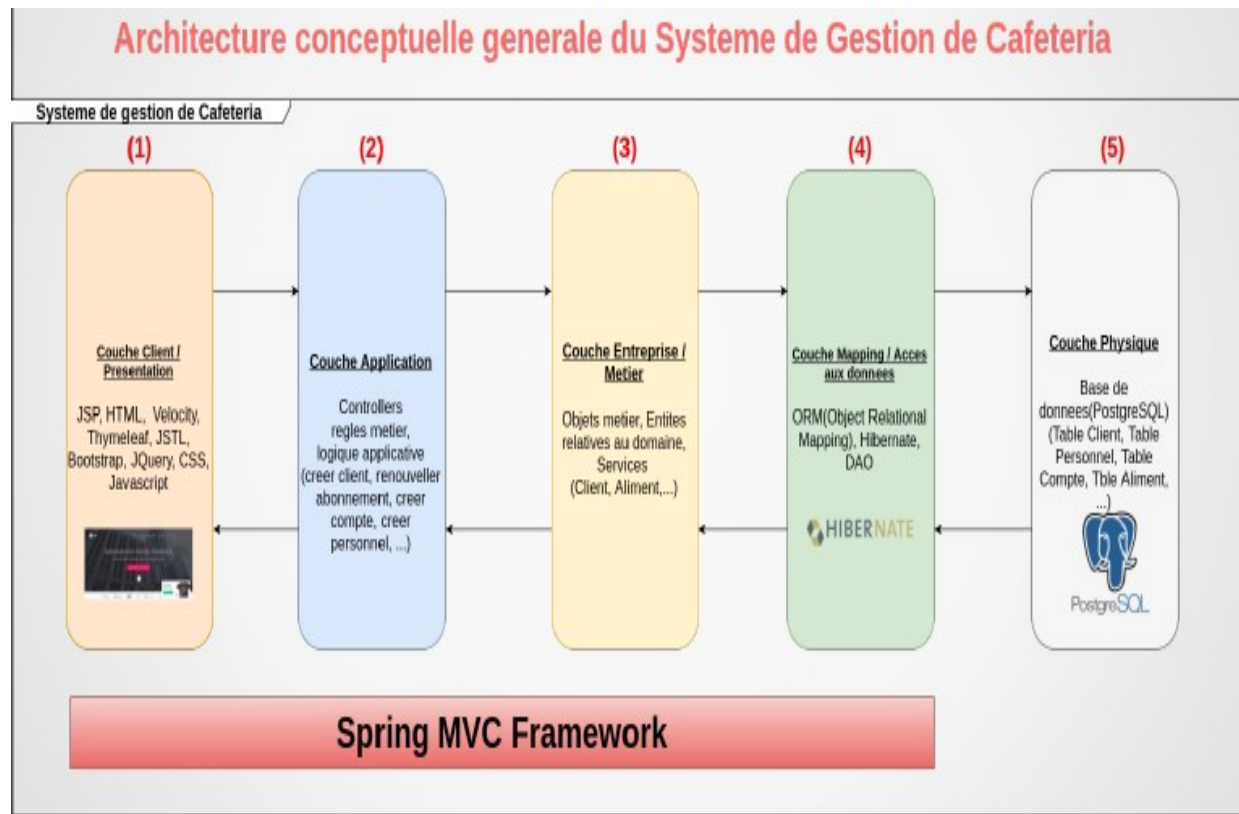


Figure 4 : Architecture de l'application suivant le modèle MVC

IV.7.2. Rôle des couches

COUCHES	TECHNOLOGIES	Obeservation
CLIENT	<ul style="list-style-type: none"> - JSP, HTML, JSTL, - Bootstrap, JQuery, CSS, javascript 	Cette couche est l'apparence visuelle de l'application, telle qu'elle sera perçue par l'utilisateur finale. C'est ici que vous aurez les images, les pages JSP, etc..
APPLICATION	<ul style="list-style-type: none"> - Controllers - Règles métier et logique applicative 	Ici nous avons les contrôleurs du pattern MVC. C'est à dire des composants qui interceptent les interactions de l'utilisateur et qui gère la cinématique (ou logique) des écrans
ENTREPRISE	<ul style="list-style-type: none"> - Objets metiers, entités relatives au domaine (objet du domaine manipule chaque jour), - Services 	La couche Service regroupe tous les services, ou réalisation de cas d'utilisation. L'idée est d'exposer ces services pour différentes applications clientes.
MAPPING	<ul style="list-style-type: none"> - ORM - Hibernate - DAO 	La couche Persistance est la couche qui sert de communicateur avec la base de données. Son avantage est le rôle de sas de sécurité entre l'application et la base de données. Si la structure de données change, seule (en théorie) cette couche est à modifier.
PHYSIQUE	<ul style="list-style-type: none"> - SGBDR - PostgreSQL - MySQL 	La couche physique contient les données à afficher

Tableau 1 : Diagramme de cas d'utilisation

IV.7.4. Modèle client serveur

La communication dans notre application se fait suivant le mode client-serveur pour toutes transactions et requêtes possible.

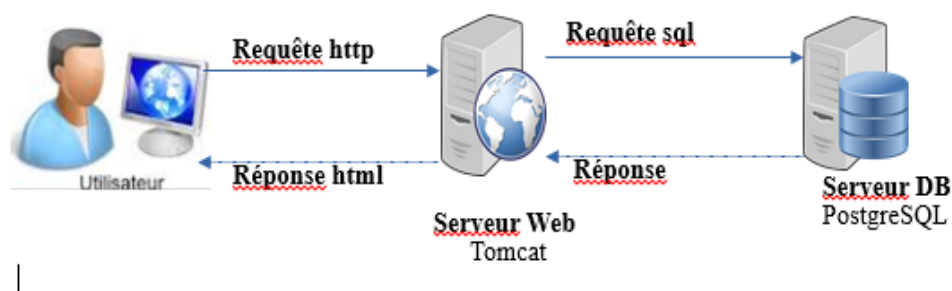


Figure 5 : Architecture client-serveur

V. Démonstration et test de validation du Système

V.1. Page d'accueil

En tapant l'adresse URL suivante <http://localhost:8080/gestionCafeteria/> dans la barre d'adresse du navigateur l'interface ci-dessus nous est présentée : il s'agit de la page d'accueil de l'application **gestionCafeteria** accessible par tous les utilisateurs y compris les utilisateurs anonymes non connectés.

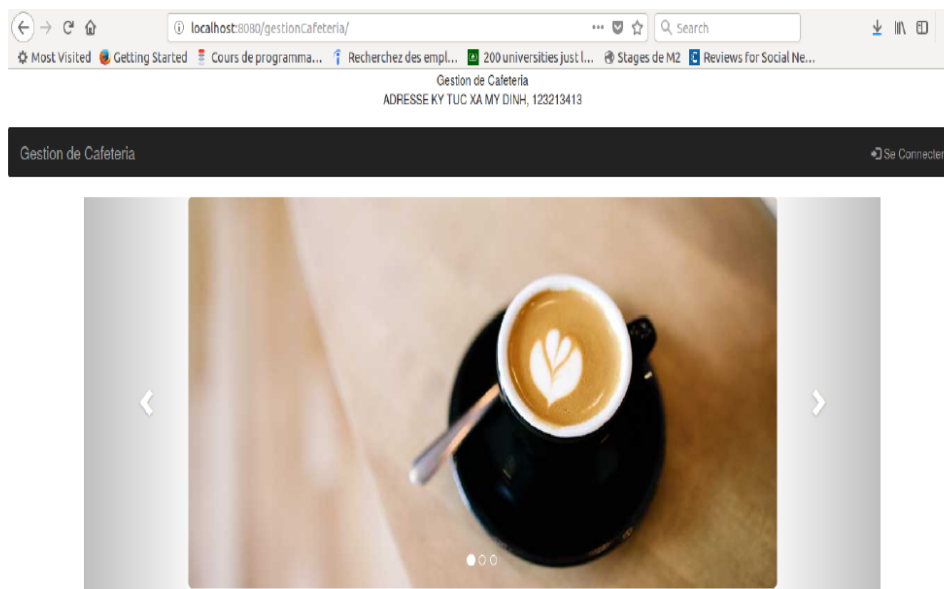


Image 1 : Accueil pour la visibilité

Cette page présente déjà des fonctionnalités et une utilité intéressantes. En effet, un utilisateur du système peut consulter à travers cette page la planification qui a été faite concernant les plats retenus pour les jours à venir et aussi consulter dans le temps ceux qui ont été préparés dans le passé comme on peut le voir et illustré dans la figure suivante sous forme de tableau et d'une frise chronologique.

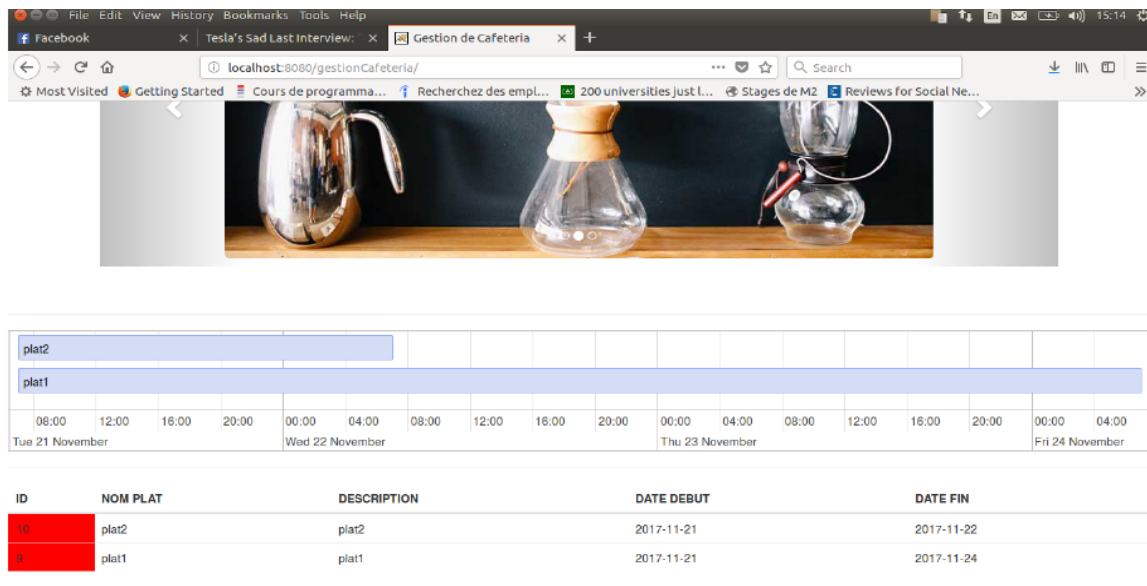


Image 2 : Diagramme de cas d'utilisation

V.2. Interface de connexion

Cliquer sur le lien Se connecter sur la page d'accueil permet d'être redirigé sur la page de connexion qui est très classique avec un nom d'utilisateur et un mot de passe. Deux rôles ou deux types d'utilisateur ont été identifiés dans le système à savoir l'administrateur responsable de la gestion et l'utilisateur dont la tâche principale est de supporter le processus des transactions tels que la vente des plats, la validation des cartes, les actions créditer et renouvellement des cartes. En utilisant un nom d'utilisateur déjà inscrit dans le système et le bon mot de passe (ici nous avons seulement deux utilisateurs pour les cas de tes **bill** (utilisateur administrateur) ayant pour mot de passe **newbensur3190** et **senburbens** (utilisateur simple) ayant comme mot de passe **senburbensnew**) nous sommes redirigés vers la page associée à ce rôle que seuls les utilisateurs ayant ce privilège en sont capables.

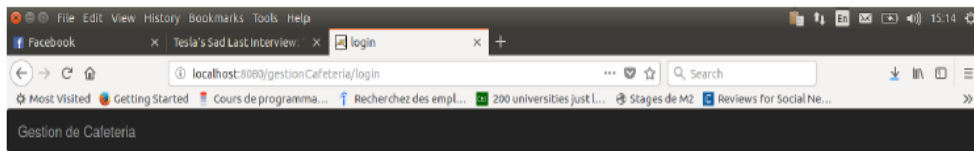


Image 3 : Page d'Authentification

V.3. Test de validation : Utilisateur Administrateur

Nous allons démontrer dans cette section les fonctionnalités et les tâches qu'il est possible d'effectuer après s'être connecté en tant qu'utilisateur ayant un rôle d'administrateur du système pour les options **Gérer Plat**, **Gérer Client** et **Gérer Carte**. La logique d'utilisation étant la même nous allons seulement présenter les scénarios de la fonctionnalité **Gérer Plat** : ce scénario est donc similaire pour les autres fonctionnalités susmentionnées. Après s'être connecté, l'utilisateur administrateur se retrouve sur la page dédiée aux utilisateurs ayant ce rôle. Cette page est illustrée avec la capture ci-dessous. D'ici l'utilisateur pourra effectuer toutes les tâches de gestion et d'administration nécessaire au bon fonctionnement des systèmes tels que la gestion des plats, des clients, des cartes et des utilisateurs.

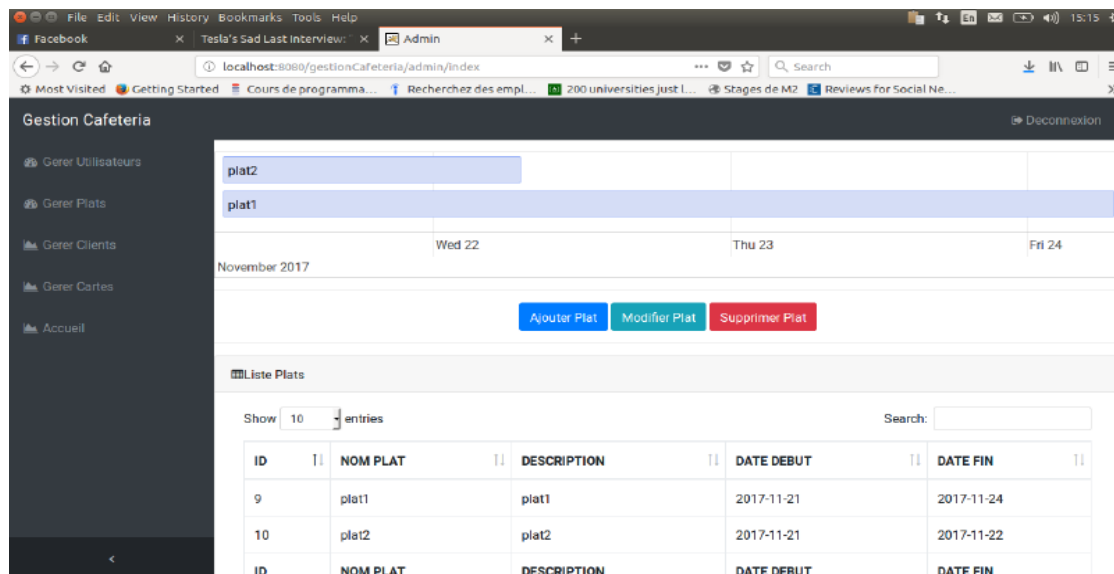


Image 4 : Page d'exercice des fonctionnalités

Démontrons à présent les différentes opérations que l'on peut effectuer en choisissant comme exemple de scenario la gestion des plats. Les trois opérations possibles sont **Ajouter Plat**, **Modifier Plat** et **Supprimer Plat**.

- Fonctionnalité *Ajouter Plat*

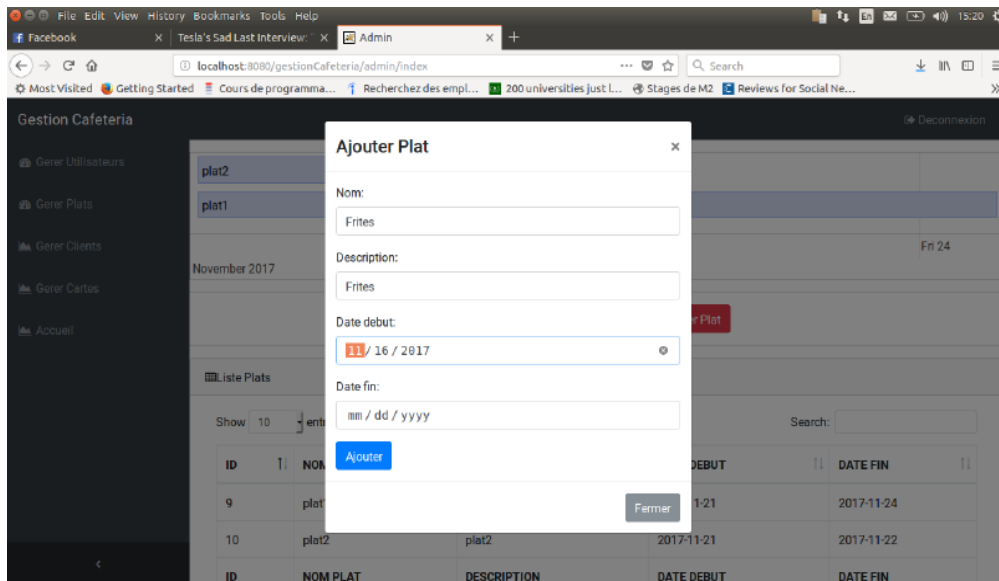


Image 5 : Actionner le bouton intitulé Ajouter Plat

- Fonctionnalité **Modifier Plat**

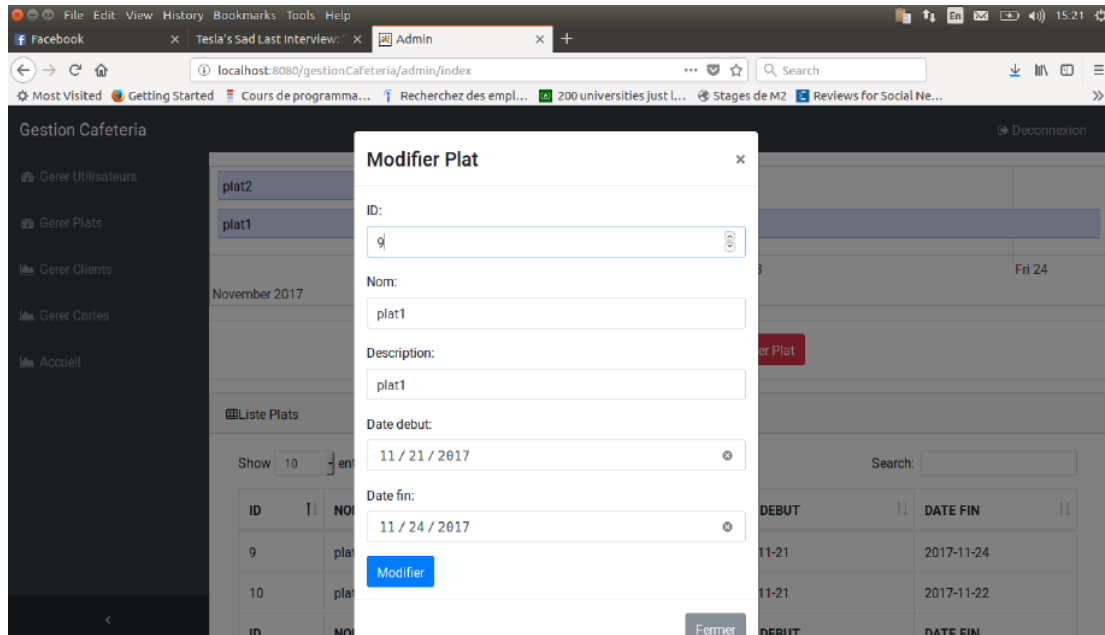


Image 6 : Actionner le bouton intitulé Modifier Plat

- Fonctionnalité **Supprimer Plat**

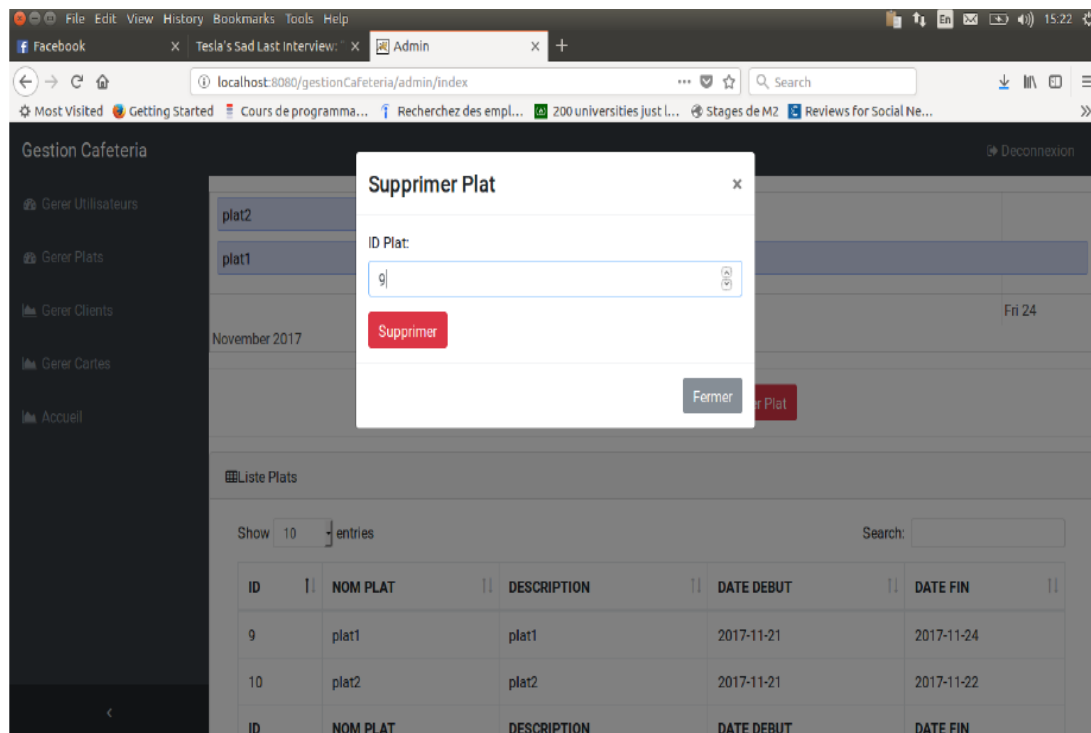


Image 7 : Actionner le bouton intitulé Supprimer Plat

L'ajout de nouveaux utilisateurs pourra se faire en cliquant sur le lien ou l'option **Gérer Utilisateurs**. La capture d'écran ci-après représente les fonctionnalités au niveau de cette section de l'application.

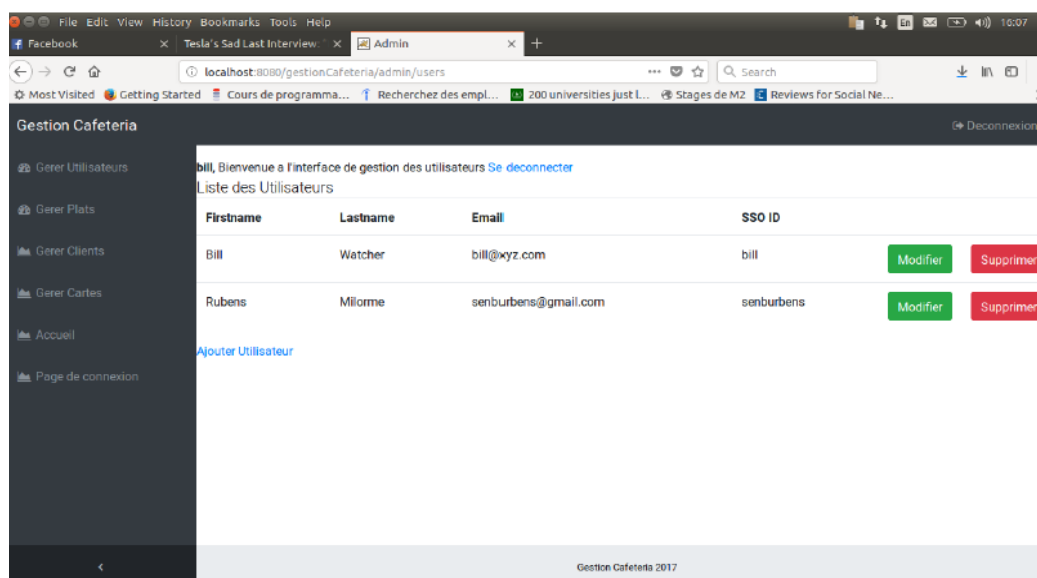
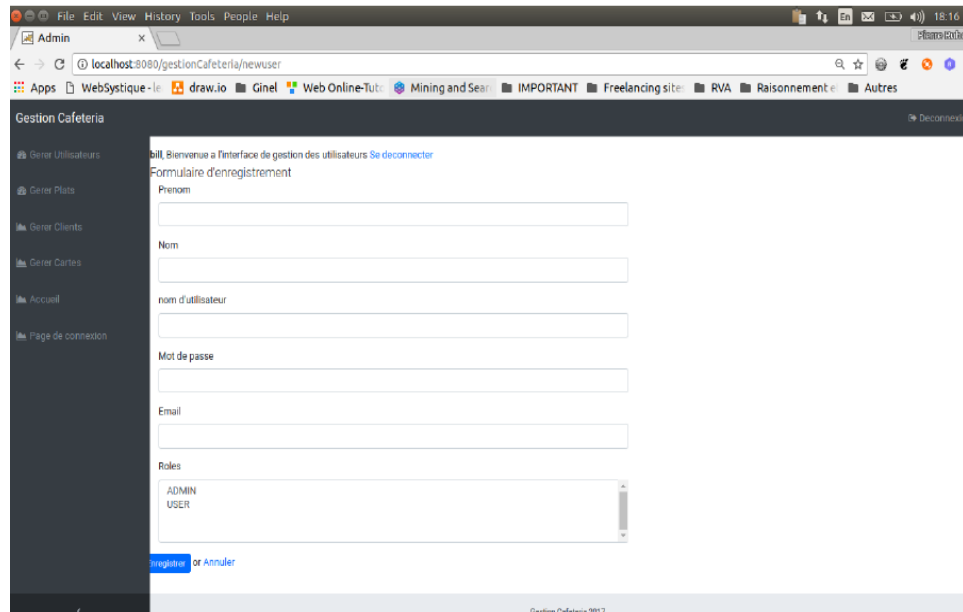


Image 8 : Ajout d'un utilisateur

En effet il est possible à ce niveau d'ajouter de nouveaux utilisateurs en cliquant sur le lien en bas à gauche intitulé **Ajouter Utilisateur**, de supprimer ceux qui existe déjà ou de les modifier. Les différentes zones d'interaction principalement des boutons et colores sont assez visibles.

- Ajouter Utilisateurs



The screenshot displays a web browser window with the address bar showing 'localhost:8080/gestionCafeteria/newuser'. The page title is 'Gestion Cafeteria'. On the left, a dark sidebar contains a menu with the following items: 'Gestion Utilisateurs', 'Gestion Plats', 'Gestion Clients', 'Gestion Cartes', 'Accueil', and 'Page de connexion'. The main content area is titled 'Bienvenue à l'interface de gestion des utilisateurs Se déconnecter' and 'Formulaire d'enregistrement'. It contains several input fields: 'Prénom', 'Nom', 'nom d'utilisateur', 'Mot de passe', and 'Email'. Below these is a 'Roles' section with a dropdown menu showing 'ADMIN' and 'USER'. At the bottom of the form are two buttons: 'S'inscrire' (highlighted in blue) and 'ou Annuler'.

Image 9 : Ajout d'un utilisateur

- Modifier Utilisateurs

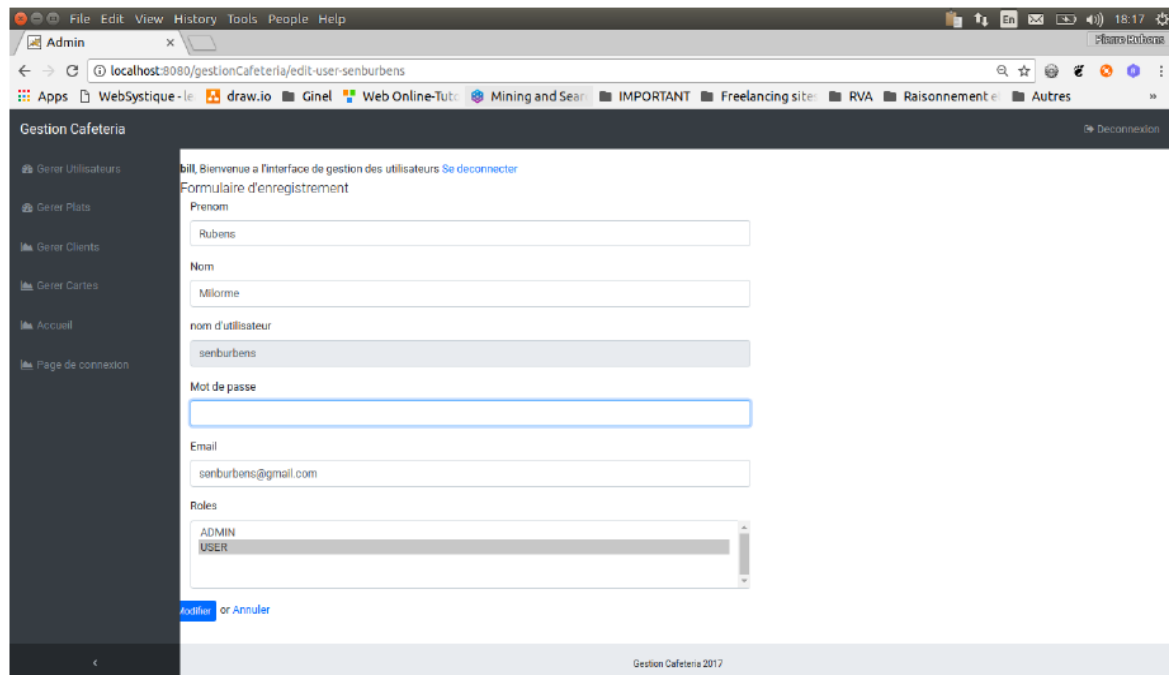


Image 10 : Modification d'un utilisateur

a) Test de validation : Utilisateur Simple

En se connectant avec un compte d'utilisateur simple, les deux interfaces suivantes sont celles que nous apercevons à travers

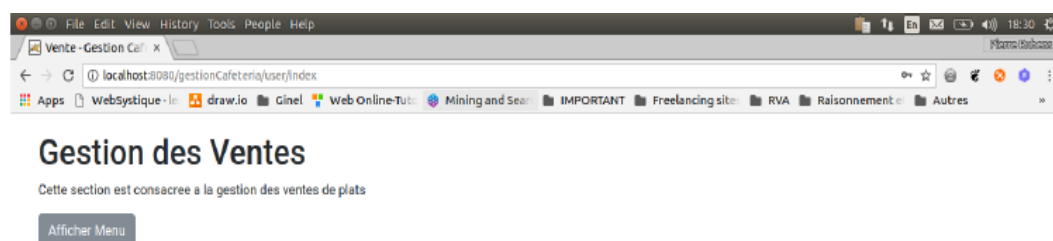


Image 11 : Connexion de l'application sur le navigateur

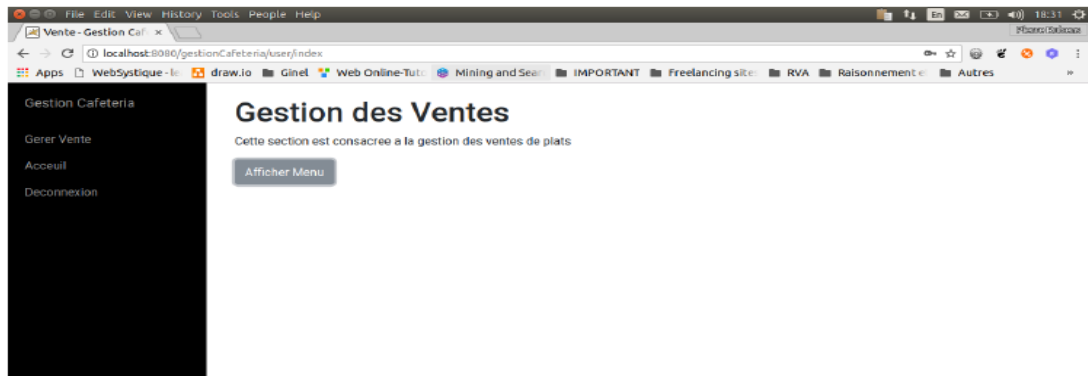


Image 12 : Affichage du Menu

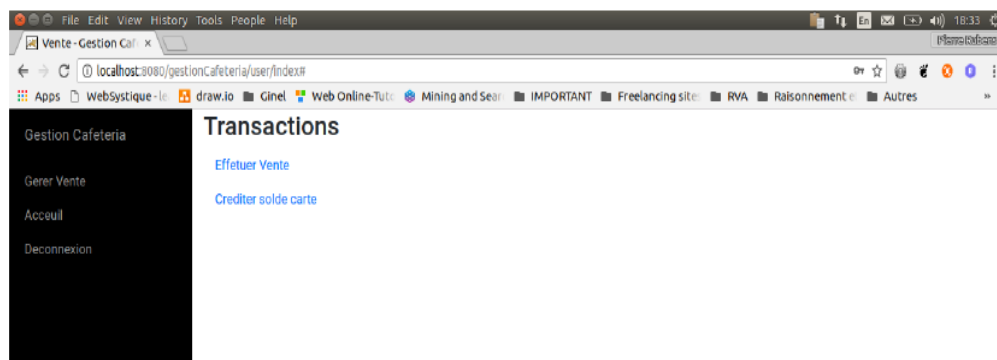


Image 13 : Affichage de transaction

- Opération Créditer Compte

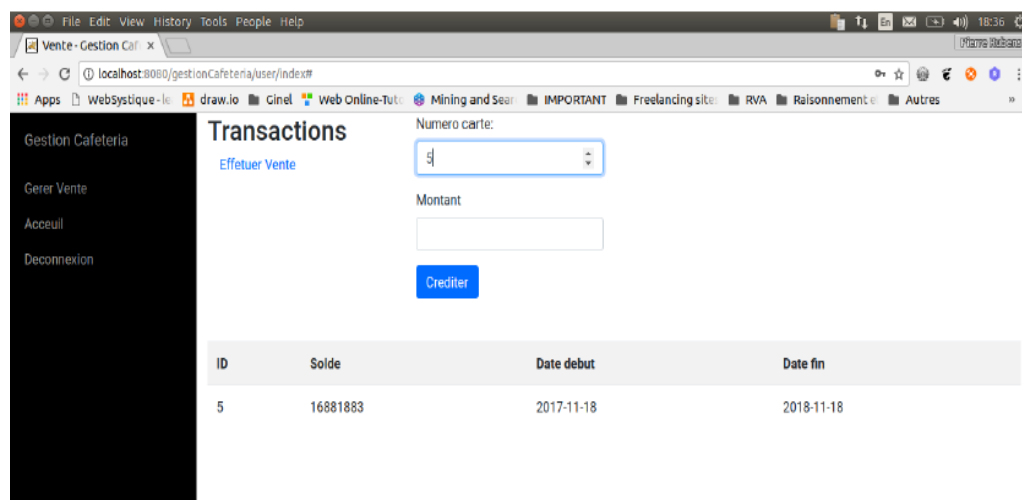


Image 14 : Transaction des données dans la base de données

VI. Organisation et structure de la base de données

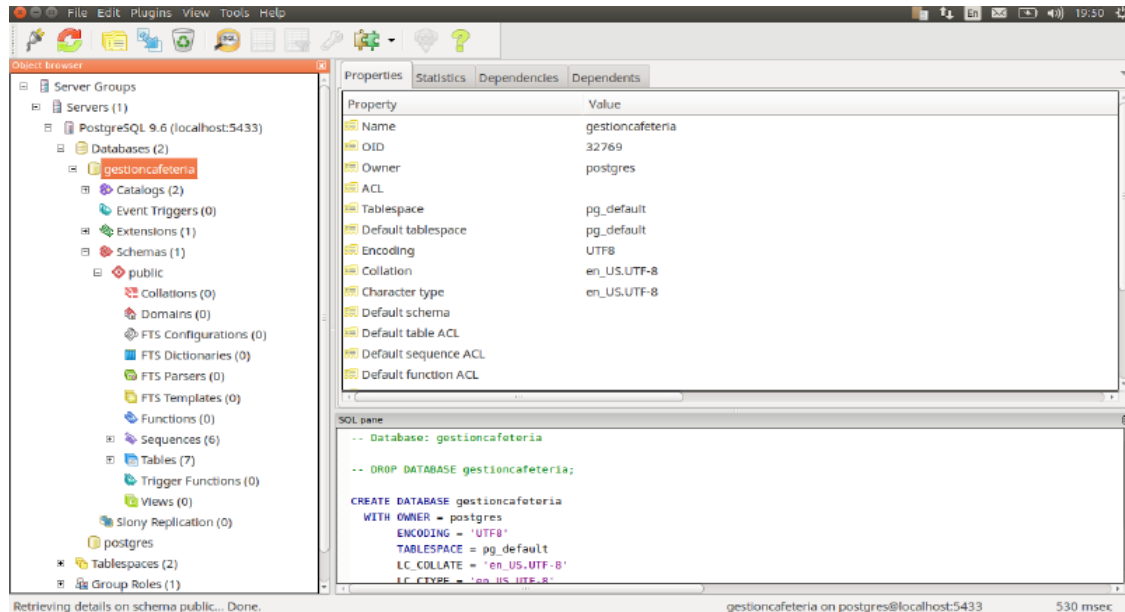


Image 15 : Base de données gestion Cafétéria

Tables

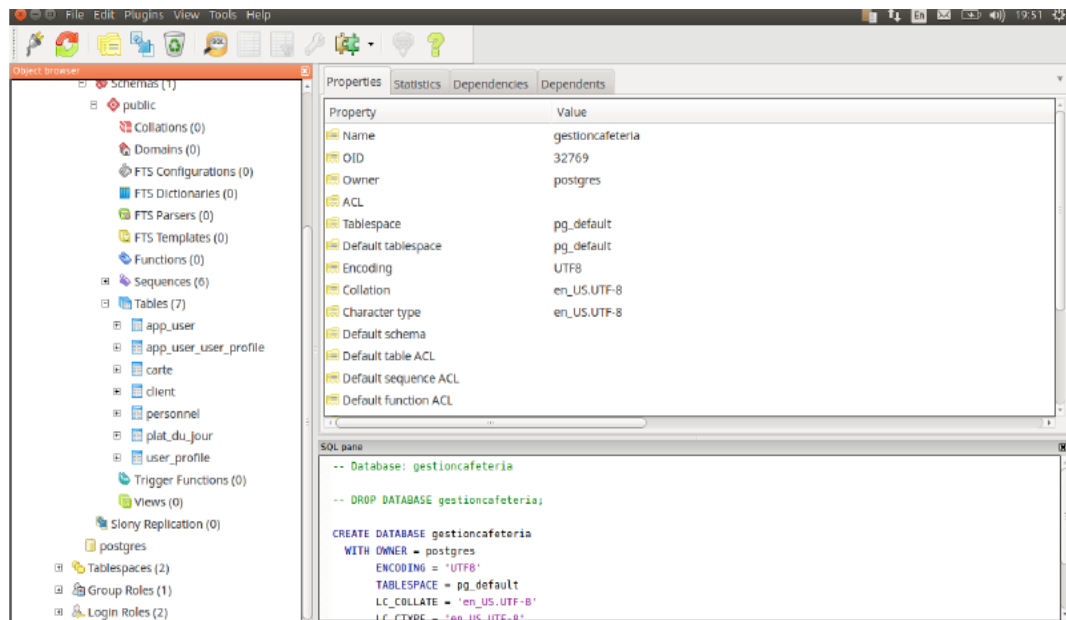


Image 16 : Base de données avec PostgreSQL

Table Utilisateur

	user_id [PK] bigint	user_profile_id [PK] bigint
1	1	2
2	6	1
*		

Client

	id [PK] serial	nom text	prenom text	email text	adresse text	statut text
1	4	pp3	pp3	pp3.com3	dwfdfs3	sgfffg3
2	5	nh	hh	hh@gmail.com	sdf	pp
3	6	llll	ll	ll@gmail.com	sgd	qw
*						

Table

Table Plat

Table Carte

	user_id [PK] bigint	user_profile_id [PK] bigint
1	1	2
2	6	1
*		

	numero [PK] serial	solde integer	start_date date	end_date date
1	5	17981883	2017-11-18	2018-11-18
2	7	900	2017-11-10	2018-11-10
*				

Image 17 : Différentes tables utilisées dans la base de données

VII. Limites

Tout projet n'étant toujours pas atteint à cent pourcent, possède quelques limites vu qu'il est réalisé dans le temps et dans l'espace, ce qui nous a poussés à ne pas finaliser l'implémentation du module de transaction liée au paiement des abonnés.

Conclusion

En guise de conclusion, la gestion de cafétéria nous a permis de mettre en œuvre les notions théoriques sur la modélisation UML passant par la pratique qui est la phase d'implémentation. Nous avons appliqué les nouvelles techniques de la mise en œuvre des applications Web en utilisant des outils performants et professionnels. Nous pouvons dire que la conception des applications web est un domaine techniquement en pleine croissance, ceci requière un bon apprentissage des outils afin d'adapter les connaissances anciennes à la nouvelle connaissance et cela nous a pris du

temps pour l'adaptation. Ainsi nos perceptive s'étend sur la finition de la partie transaction liée à la fonctionnalité de paiement de l'application.

Référence

- [1] https://github.com/postgresqlfr/pgdocs_fr
- [2] https://www.tutorialspoint.com/spring/spring_overview.htm
- [3] <http://yannart.developpez.com/java/spring/tutoriel/>
- [4] <https://www.w3schools.com/js/>
- [5] <https://www.enable-javascript.com/fr/>
- [6] <https://openclassrooms.com/courses/simplifiez-vos-developpements- javascript-avec-jquery/premiers-pas-avec-ajax>
- [7] https://sutterlity.gitbooks.io/apprendre-jquery/content/la_methode_ajax.html
- [8] https://sutterlity.gitbooks.io/apprendre-jquery/content/les_variables.html
- [9] <http://api.jquery.com/jquery.ajax/>

