

LIVRABLE MOE n°4 - GESTION DE PROJET



Photographie du Restaurant Universitaire de Vannes

PROJET Gestion et réservation de l'alimentation universitaire - CROUS

Groupe 588 MIAGE-Classique

Groupe n°4 : Guillaume COLIN, Joanna GARELLE, Mathieu LE MAROUILLE, Viet Hung NGUYEN

TABLE DES MATIÈRES

I. SYNTHÈSE DU COPIL.....	3
II. MODÉLISATION DES DONNÉES.....	5
A. Le Modèle Conceptuel des Données.....	5
B. Le Modèle Relationnel des Données.....	7
III. INSTANCES DE LA BASE DE DONNÉES.....	7
IV. CONTRAINTES.....	7
A. Contraintes sur les entités.....	8
B. Contraintes de domaine.....	8
V. FONCTIONNALITÉS.....	8
1. Affichage d'informations.....	8
2. Gestion des réservations.....	9
3. Gestion des utilisateurs.....	9
4. Gestion des menus.....	9
5. Gestion des stocks.....	9
6. Paiement des menus.....	9
VI. CONCLUSION.....	10
VII. ANNEXES.....	11

I. SYNTHÈSE DU COPIL

Date : Le vendredi 27 octobre 2023

Participants :

- MOE (Groupe 4) : Guillaume COLIN, Joanna GARELLE, Mathieu LE MAROUILLE, Viet Hung NGUYEN
- MOA (Groupe 8) : Michaël BESILY, Maël JOUFFROY, Erwan GEFFRAY, Matéo SOLER

Nous avons pu échanger à propos du projet de « Gestion et réservation de l'alimentation universitaire - CROUS » avec l'équipe MOA lors du COPIL ayant eu lieu le 27 octobre 2023.

Ce comité de pilotage avait pour but d'apporter des corrections à la MOE après la livraison du deuxième livrable, ce qui nous permettra par la suite de répondre plus précisément aux attentes de la MOA.

Lors de cet entretien, il a été convenu avec la maîtrise d'oeuvre, de mettre en place les correctifs suivants :

○ **Page de gestion des stocks :**

La page de gestion des stocks ne servirait pas uniquement à gérer les stocks (ajouter/modifier/supprimer) les produits disponibles pour le service. Cette page de gestion permettrait également au personnel de pouvoir gérer des menus. Menus qui seraient aussitôt modifiés sur l'application de réservation. Les utilisateurs seront ainsi notifiés si le menu qu'ils avaient réservé a subi une modification, et auront alors le choix d'annuler ou modifier leur réservation par la suite.

○ **Le type de réservation dépend du type de restaurant :**

Les Restaurants Universitaires (RU) n'ayant pas de limite de stock dans ce modèle, si l'utilisateur réserve dans un RU, l'application affichera alors le nombre de places disponibles sur un créneau s'étalant sur 30 minutes, temps choisi par la MOA, car jugé suffisant pour le repas. Le but recherché lors de la réservation dans un RU est le gain de temps pour les utilisateurs, et d'alléger la charge de travail aux heures de pointe pour mieux répartir cette charge sur l'ensemble du service pour les personnels du restaurant.

Si l'utilisateur a décidé de réserver dans une cafétéria, ce type de restaurant ayant des stocks limités, la réservation à l'aide de l'application aura pour but de mettre à jour les stocks disponibles, afin d'informer l'utilisateur lorsque certains menus ne sont plus disponibles à la vente. La gestion de places dans une cafétéria n'aurait pas de sens, car il peut choisir d'emporter son repas dans un autre espace dans l'Université ou à l'extérieur.

- **Un menu se compose uniquement d'une entrée, d'un plat ou d'un dessert :**

Nous avons ajouté, dans notre modèle précédent, des accompagnements (boisson, condiments), que l'utilisateur pouvait réserver à côté de son menu. Mais cette notion de supplément étant superficielle et non voulue par la MOA, nous devons la supprimer dans notre nouveau MCD.

- **L'application mettra à jour automatiquement les stocks après réservation de repas :**

Nous avons commis une autre erreur, celle de considérer qu'un membre du personnel, en plus de confirmer la venue des utilisateurs ayant réservé un créneau, mettrait à jour les stocks en prenant en compte les menus payés. Mais c'est l'application de réservation qui mettra à jour automatiquement les stocks après qu'une réservation aura été confirmée, ou qu'un porteur de carte n'ayant pas réservé, commande un menu sur place. Cette gestion des stocks se fera grâce à une API (Interface des programmation), et uniquement sur les cafétérias comme expliqué plus haut.

- **Le paiement se faisant à l'avance ou avec un TPE sur place, le paiement n'est pas géré par l'application :**

Pour faciliter la gestion des stocks dans notre application, l'API dont nous avons parlé précédemment, comptabilise les menus payés en avance sur l'application et sur place. L'application de réservation de repas ne gèrera pas l'aspect du règlement par les utilisateurs.

une cafétéria donnée. Un menu qui n'aura jamais été proposé à la carte n'apparaîtra jamais dans la table "Menu". Un menu est composé soit d'une entrée, soit d'un plat, soit d'un dessert, soit d'un ensemble de ces trois propositions. Il doit donc être composé au minimum d'un article qui doit être présent dans les stocks de la cafétéria qui propose le menu. Si un des articles qui compose le menu, n'est plus dans les stocks, le menu sera alors supprimé de la carte du jour.

- **Les contraintes sur le MCD**

Nous avons abordé quelques contraintes dans notre précédent modèle.

Contrainte de Totalité sur "Personne" : Toutes les personnes enregistrées dans la base étaient soit personnel, soit porteur de carte ou les deux à la fois.

Contrainte de Partition sur "Resto Crous" : Tous les Restos du Crous sont soit des RU, soit des Cafétérias, mais de manière exclusive.

Contrainte sur l'heure de réservation, car les porteurs de cartes ne peuvent réserver que sur les horaires compris entre l'heure d'ouverture et celui de fermeture du restaurant choisi.

En plus, nous avons ajouté une contrainte : le booléen "deuxième_reserve". Il ne prend que deux valeurs et est clé étrangère de "reserve", ce qui permet aux utilisateurs de prendre les deux repas auxquels ils ont droit au même endroit et sur le même créneau.

- **Le nombre de places d'un RU**

La gestion de l'attente et du service aux heures de pointe au RU étant cruciale, nous avons rajouté en attribut dans l'entité "RU" le nombre de places afin de savoir la capacité d'accueil.

- **Les suppléments supprimés**

Comme expliqué précédemment, la MOA nous a demandé de supprimer les suppléments que peuvent réserver les élèves, car un menu reservable ne doit pas en contenir.

B. Le Modèle Relationnel des Données

Voici le Modèle Relationnel obtenu à partir du Modèle Conceptuel de Données grâce à Looping.

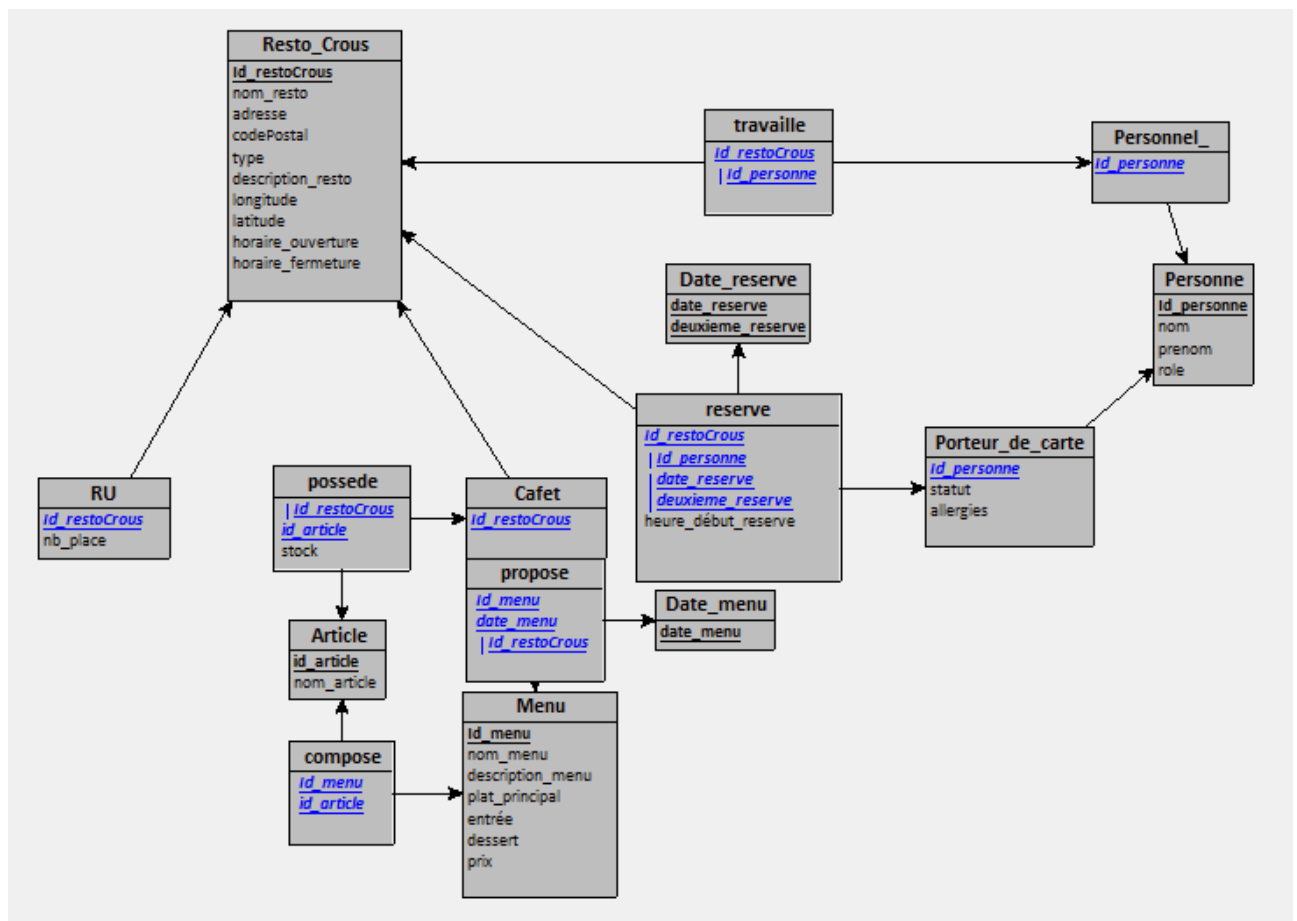


Figure 2- Deuxième version du Modèle Relationnel de Conception des Données sur la gestion des réservations

III. INSTANCES DE LA BASE DE DONNÉES

Si vous souhaitez voir nos instances pour notre base de données, nous avons ajouté en pièce jointe, lors du rendu deux, fichiers contenant nos créations de tables et insertions de valeurs à l'intérieur de celles-ci.

Ce sont les fichiers "Insert_CROUS.sql" et "Creation_tables_Resto_Crous.sql".

IV. CONTRAINTES

Dans le contexte de la base de données, une contrainte est comme une règle que l'on met en place pour s'assurer que les données stockées sont correctes et cohérentes. Voici nos contraintes en plus de celles déjà modélisées dans le MCD.

A. Contraintes sur les entités

Pour des raisons de simplicité pour l'utilisation des fonctionnalités, mais aussi de non-sens si ces données étaient nulles, nous considérons que les attributs suivants ne peuvent être nuls, à savoir : `nom_resto`, `adresse`, `codePostal`, `type`, `longitude`, `latitude`, `horaire_ouverture`, `horaire_fermeture`, `nb_place`, `nom_menu`, `prix`, `statut`, `heure_debut_reserve`, `nom`, `prenom` et `role`.

B. Contraintes de domaine

Contrainte de domaine sur	Leurs valeurs doivent être :
<code>type</code>	“Restaurant” ou “Cafétéria”.
<code>stock</code>	Supérieur ou égal à 0.
<code>nb_place</code>	Supérieur ou égal à 0.
<code>statut</code>	“alternant” ou “ boursier” ou “non boursier”
<code>prix</code>	Supérieur ou égal à 0.
<code>heure_debut_reserve</code>	compris entre <code>horaire_ouverture</code> et <code>horaire_fermeture</code>
<code>role</code>	“étudiant” ou “personnel” ou “administrateur” ou “professeur” ou “exterieur”

V. FONCTIONNALITÉS

Les fonctionnalités d'une application sont les différentes actions et services qu'elle offre aux utilisateurs. Ces fonctionnalités permettent l'accès et la manipulation des informations stockées dans la base de données instanciée précédemment.

Les groupes de fonctionnalités que nous voulions implémenter, proposés ci-dessous, permettront de répondre aux besoins des usagers lors de la réservation de leur repas aux restaurants du Crous. Certaines ont pu faire l'objet de requêtes SQL, montrées en Annexes, d'autres sont listées de manière exhaustive.

1. Affichage d'informations

Menus dans une cafétéria : Affiche les menus disponibles pour le jour même dans une cafétéria donnée. → *figure 1 : Annexe page 11*

Carte géographique des restaurants : Option complexe mais possible pour afficher les emplacements des restaurants sur une carte.

Description des restaurants (RU et cafétéria) : Affiche le nom et la description des restaurants universitaires et cafétérias. → *figure 2 Annexes page 11*

Articles d'un menu dans une cafétéria (sans proposition) : Affiche les informations sur les articles composant un menu dans une cafétéria spécifique. → *figure 3 Annexes page 12*

Stock d'un article donné : Affiche le stock disponible pour un article particulier.
→ *figure 4 : Annexes page 12*

2. Gestion des réservations

Places disponibles pour un créneau dans un RU : Affiche le nombre de places disponibles pour un créneau donné dans un restaurant universitaire. → *figure 5 : Annexes page 13*

Créneaux possibles de réservation dans un RU : Découpe les horaires d'ouverture du restaurant donné par tranche de trente minutes pour donner en sortie les différents créneaux de réservation.

Visualiser les informations de la réservation : Affiche les détails de la réservation pour le personnel et le réservant. → *figure 6 : Annexes page 13*

Restaurants déjà réservés par l'utilisateur : Affiche les restaurants déjà réservés par un utilisateur donné. → *figure 7 : Annexes page 13*

3. Gestion des utilisateurs

Ajout d'utilisateurs : Ajoute un nouvel utilisateur avec des informations spécifiques.
→ *figure 8 : Annexes page 13*

Modification d'utilisateur : Modifie le rôle d'un utilisateur existant.
→ *figure 9 : Annexes page 13*

Suppression d'utilisateur : Supprime un utilisateur spécifié.
→ *figure 10: Annexes page 13*

4. Gestion des menus

Ajout de menu : Ajoute un nouveau menu avec des détails spécifiques.
→ *figure 11 : Annexes page 14*

Prix du menu : Affiche le prix du menu, ajusté selon le statut du réservant et du type de restaurant .
→ *figure 12: Annexes page 14*

5. Gestion des stocks

Modification du stock d'un article : Modifie le nombre d'exemplaires en stock d'un article dans une cafétéria donnée → *figure 13 : Annexes page 15*

6. Paiement des menus

Payer sur place un menu : Affiche les détails pour payer un menu sur place.
→ *figure 14 : Annexes page 15*

VI. CONCLUSION

Avec les modifications ajoutées au Modèle Conceptuel de Données et la réalisation du Modèle Relationnel des Données ainsi que de la base de données, nous avons pu approfondir notre sujet et mieux comprendre les attentes de la Maîtrise d'Ouvrage.

Comme piste d'amélioration, nous avons pensé à ajouter un graphique d'affluence, qui servirait de repère visuel pour les personnes souhaitant manger dans un Restaurant du CROUS et voulant éviter un temps d'attente trop long. Ce graphique permettrait de voir le niveau d'occupation d'un Restaurant Universitaire lors de ses horaires d'ouverture, sur plusieurs créneaux.

Cela s'inscrirait dans la continuité du but de l'application qui est d'économiser du temps aux utilisateurs de l'application.

L'objectif du prochain livrable sera d'implémenter plusieurs fonctionnalités et créer le site qui héberge l'application de réservation de repas et de places aux restaurants du CROUS.

VII. ANNEXES

figure 1 - Afficher menus disponibles et proposés pour le jour même dans une cafétéria donnée

```
SELECT nom_resto AS Nom_Cafet, nom_menu AS Nom_Menu
From Menu Join propose On Menu.id_menu = propose.id_menu Join Cafet On
propose.id_restoCrous = Cafet.Id_restoCrous
Where nom_resto = 'Cafétéria Staps' AND propose.date_menu = Date('now');
```

Nom_Cafet	Nom_Menu
Cafétéria Staps	Menu Végan
Cafétéria Staps	Menu Dégustation

figure 2 - Affichage description du restaurant (RU et cafétéria) :

```
SELECT nom_resto, description_resto
FROM RU
UNION
SELECT nom_resto, description_resto
FROM Cafet;
```

nom_resto	description_resto
Resto U' La Lombarderie	Restauration traditionnelle. Cuisine du monde. Acc...
Resto U' Le Tertre	Restauration traditionnelle. Grillades. Pôle Vente...
Salle administrative - Resto U' Ricordeau	Restauration traditionnelle. Accessible aux person...
Resto U' Le Tertre	Campus Droit-Lettres. Du lundi au vendredi de 11h0...
Resto U' La Lombarderie	Campus sciences. Du lundi au vendredi de 11h15 à 1...
Resto U' La Chantreie	Campus Atlantech-Chantreie. Ouvert du lundi au ve...
Resto U Le Coudon	Campus universitaire de La Garde. Ouvert de 11h30 ...
Cafétéria St Symphorien	Située en face de l'IUT. RDC du restaurant univers...
RU du Technopôle	Situé près de l'UFR Droit et Polytech Tours. Cuisi...
Cafet' Pôle étudiant	Restauration diversifiée. Prestations chaudes et f...
Cafet' Pôle étudiant	Campus Droit-Lettres. Du lundi au jeudi de 8h30 à ...
Cafétéria L'Escarpe	Cafétéria du campus universitaire de Toulon. Ouver...
Cafétéria Staps	Campus Staps. Ouvert de 8h à 16h. Accessible aux p...
Cafétéria Lash	Campus Carlone. Ouvert de 7h45 à 16h30. Accessible...
Cafétéria des Tanneurs	28 places assises. Bar. Sandwichs variés, salades ...

figure 3 - Affichage des informations des articles qui compose un menu dans une cafétéria :

```
SELECT Cafet.nom_resto AS Nom_Cafeteria, Menu.nom_menu AS Nom_Menu,
Article.nom_article AS Nom_Article
FROM Cafet INNER JOIN propose ON Cafet.Id_restoCrous = propose.id_restoCrous
INNER JOIN Menu ON propose.id_menu = Menu.id_menu
INNER JOIN compose ON Menu.id_menu = compose.id_menu
INNER JOIN Article ON compose.id_article = Article.id_article
WHERE Cafet.nom_resto = 'Cafétéria Staps';
```

Nom_Cafeteria	Nom_Menu	Nom_Article
Cafétéria Staps	Menu Végan	Curry de pois chiches et légumes
Cafétéria Staps	Menu Végan	Rouleaux de printemps aux légumes
Cafétéria Staps	Menu Végan	Cheesecake végétalien
Cafétéria Staps	Menu Dégustation	Assortiment de fruits de mer
Cafétéria Staps	Menu Dégustation	Carpaccio de bœuf truffé
Cafétéria Staps	Menu Dégustation	Fondant au chocolat

figure 4 - Afficher stocks d'un article donné :

```
SELECT Article.nom_article, possede.stock FROM Article JOIN possede ON
Article.id_article = possede.id_article
WHERE Article.nom_article = 'Assortiment de fruits de mer';
```

nom_article	stock
Assortiment de fruits de mer	50

figure 5 - Affichage du nombre de places disponibles pour un créneau donné dans un RU donné :

```
SELECT RU.nom_resto AS Nom_RU, reserve.date_reserve AS Date_Reserve,
reserve.heure_debut_reserve AS Heure_Reserve, RU.nb_place -
COUNT(reserve.id_personne) AS Places_Disponibles
FROM reserve JOIN RU ON reserve.id_restoCrous = RU.Id_restoCrous
WHERE reserve.id_restoCrous = 17
AND reserve.date_reserve = '2023-12-05'
AND reserve.heure_debut_reserve = '13:30:00'
GROUP BY RU.nom_resto, reserve.date_reserve, reserve.heure_debut_reserve,
RU.nb_place;
```

Nom_RU	Date_Reserve	Heure_Reserve	Places_Disponibles
Resto U' La Lombarderie	2023-12-05	13:30:00	399

figure 6 -Visualiser les informations de la réservation (pour personnel et celui qui a réservé)

```
SELECT rc.nom_resto, rc.adresse, rc.codePostal,rc.description_resto, r.date_reserve,
r.deuxieme_reserve, r.heure_debut_reserve
FROM Reserve r INNER JOIN Personne p ON r.id_personne = p.id_personne
INNER JOIN Resto_Crous rc ON r.id_restoCrous = rc.Id_restoCrous
WHERE p.id_personne = 'E223029G'
AND r.date_reserve = Date('now')
```

figure 7- Affichage des restaurants qui ont déjà été réservés avant par l'utilisateur

```
SELECT DISTINCT RC.Id_restoCrous, RC.nom_resto, RC.adresse, RC.codePostal
FROM Reserve R JOIN Resto_Crous RC ON R.id_restoCrous = RC.Id_restoCrous
WHERE R.id_personne = 'E223029G'
```

Id_restoCrous	nom_resto	adresse	codePostal
16	Cafet' Pôle étudiant	Chemin de la Censive du Tertre	44300

figure 8 - Ajout de l'Etudiant Pierre Dupont :

```
INSERT INTO Personne (id_personne, nom, prenom, role)
VALUES ('E224586D', 'Dupont', 'Pierre', 'Etudiant');
```

figure 9 -Modification de l'ID pour que Pierre Dupont devienne professeur :

```
UPDATE Personne
SET id_personne = 'P224586D', nom = 'Durant', prenom = 'Jean', role = 'Professeur'
WHERE id_personne = 'E224586D';
```

figure 10 - Suppression de Pierre Dupont :

```
DELETE FROM Personne
WHERE id_personne = 'P224586D';
```

figure 11 - Ajouter un Menu :

```
INSERT INTO Menu (id_menu, nom_menu, description_menu, plat_principal, entree,
dessert, prix) VALUES (16, 'Menu Mexicain', 'Éveillez vos papilles avec les saveurs
épicées et vibrantes du Mexique.', 'Tacos au poisson', 'Guacamole frais avec nachos',
'Churros avec sauce au chocolat', 5.00);
```

14	Menu Mexicain	Éveillez vos papilles avec les saveurs épicées et ...	Tacos au poisson	Guacamole frais avec nachos	Churros avec sauce au chocolat	5.50
15	Menu Fusion	Une fusion créative de cuisines du monde pour une ...	Tartare de saumon aux saveurs asiatiques	Maki au foie gras et mangue	Millefeuille exotique	6.50
16	Menu Mexicain	Éveillez vos papilles avec les saveurs épicées et ...	Tacos au poisson	Guacamole frais avec nachos	Churros avec sauce au chocolat	5.00

figure 12 - Voir prix du menu

prix inscrit pour cafétéria, sinon si RU, prix en fonction du statut du porteur de carte (non boursier => 3,30€, boursier => 1€, alternant => 0,30€)

```
SELECT nom_resto AS Nom_Cafet, nom_menu AS Nom_Menu, prix
From Menu Join propose On Menu.id_menu = propose.id_menu
Join Cafet On propose.id_restoCrous = Cafet.Id_restoCrous
Where nom_resto = 'Cafétéria des Tanneurs'
```

Nom_Cafet	Nom_Menu	prix
Cafétéria des Tanneurs	Menu Végan	5.00
Cafétéria des Tanneurs	Menu Italien	4.25

figure 13 - Modifier le stock d'un article (ajouter 50 à l'article Fondant au chocolat) et afficher la modification :

Table Article :

id_article	nom_article
3	Fondant au chocolat

Table possede :

id_restoCrous	id_article	stock
16	3	40

```
UPDATE possede SET stock = stock + 50
WHERE id_restoCrous = 16 AND id_article = 3;
SELECT id_restoCrous, id_article, stock
FROM possede
WHERE id_restoCrous = 16 AND id_article = 3;
```

id_restoCrous	id_article	stock
16	3	90

figure 14 -Payer sur place un menu :

```
SELECT RC.nom_resto, M.nom_menu, M.prix
FROM Resto_Crous RC INNER JOIN propose P ON RC.Id_restoCrous = P.id_restoCrous
INNER JOIN Menu M ON P.id_menu = M.id_menu
WHERE M.nom_menu = 'Menu Végan'
AND RC.nom_resto = 'Resto U' Le Tertre';
```

nom_resto	nom_menu	prix
Resto U' Le Tertre	Menu Végan	5.00