

Groupe2:
CAHIER D'ANALYSE ET DE CONCEPTION DU
LOGICIEL de Camerpedia



Sous la supervision de : Dr JIOMEKONG FIDEL



Listes des participants :

<i>NOMS et prenom</i>	<i>Matricules</i>
TONG Samuel Quentin	16U2085
NGUEJIP MUKETE Yves Jordan	17Q2742
KITIO AZANFACK sage excelle	15T2364
NJOUONKOU NKANJONE Maxime Anicet	17Q2772

Listes des participants :	2
I) ANALYSE DU LOGICIEL	4
1. Spécifications fonctionnelles	4
2. spécifications non fonctionnelles	4
3. Identification des acteurs	4
4. Diagramme d'activité de notre système	6
5. Cas d'utilisations et acteurs	7
6. Description textuelle de cas utilisations	7
7. Diagramme de séquence du système	9
fig 1: Diagramme de séquence décrivant l'interaction de l'administrateur avec le système	10
fig 2: Diagramme de séquence décrivant l'interaction du nutritionniste avec le système	11
fig 3: Diagramme de séquence décrivant l'interaction d'un utilisateur grand public avec le système	11
8. Diagramme de cas d'utilisation des utilisateurs	12
9. Définition du Product backlog	12
10. Proposition de quelques maquettes utilisateurs	12
fig 1: interface principale	13
fig 2: interface présentant le résultat d'une recherche	13
fig 3: interface wiki qui presente les informations sur une nourriture	14
fig 4: interface d'enregistrement d'un utilisateur	14
fig 5: interface de connexion d'un utilisateur	15
fig 6: interface de gestion des utilisateurs	15
fig 7: interface d'ajout d'un triplet	16
fig 8: interface d'ajout d'un objet (instance)	16
11. Fragmentation du Product backlog en Sprints	17
a) sprint 1 : Installation des outils de manipulation pour l'ingénierie et l'intégration d'ontologies	17
b) sprint 2 : Développement de l'ontologie	17
c) sprint 3 : Intégration du triple store	17
d) sprint 4 : intégration de l'ontologie	17
II) CONCEPTION DU LOGICIEL	18
Architecture Devops de développement du système	18
Architecture du système	19
Environnement de développement	20

Abstract:

Camerpedia est une application web dont l'objectif est de mettre à la disposition de toutes les personnes les nourritures camerounaises, leurs compositions, leurs valeurs énergétiques, leurs vertus. Ceci dans le but d'aider les gens à mieux se nourrir en fonction de mieux se nourrir en fonction de l'objectif recherché. Elle est basée sur une base de données graphe utilisant des ontologies qui sont peuplées au fur et à mesure du temps avec nos connaissances ainsi que les connaissances des utilisateurs.

I) ANALYSE DU LOGICIEL

1. Spécifications fonctionnelles

- Les utilisateurs s'enregistrent et ont un tableau de bord pour enregistrer les informations
- Un utilisateur remplit son profil qui correspond à sa ressource
- Un utilisateur peut inviter par mail une personne à s'inscrire en précisant son profil (grand public, professionnel de la santé, nutritionniste, etc.)
- Un utilisateur peut remplir les connaissances sur un met
- Un utilisateur peut recherche un met et compléter la connaissance ajoutée par d'autres utilisateurs
- Un utilisateur peut rechercher tous les repas consommés dans un pays, une région, une ville, etc.
- Un utilisateur peut voir la composition d'un plat
- Un user peut voir les vertus d'un plat
- L'ontologie doit évoluer, (la possibilité d'enrichir)
- La fiabilité des information (la possibilité d'accepter uniquement les information fiable et vérifié)

2. spécifications non fonctionnelles

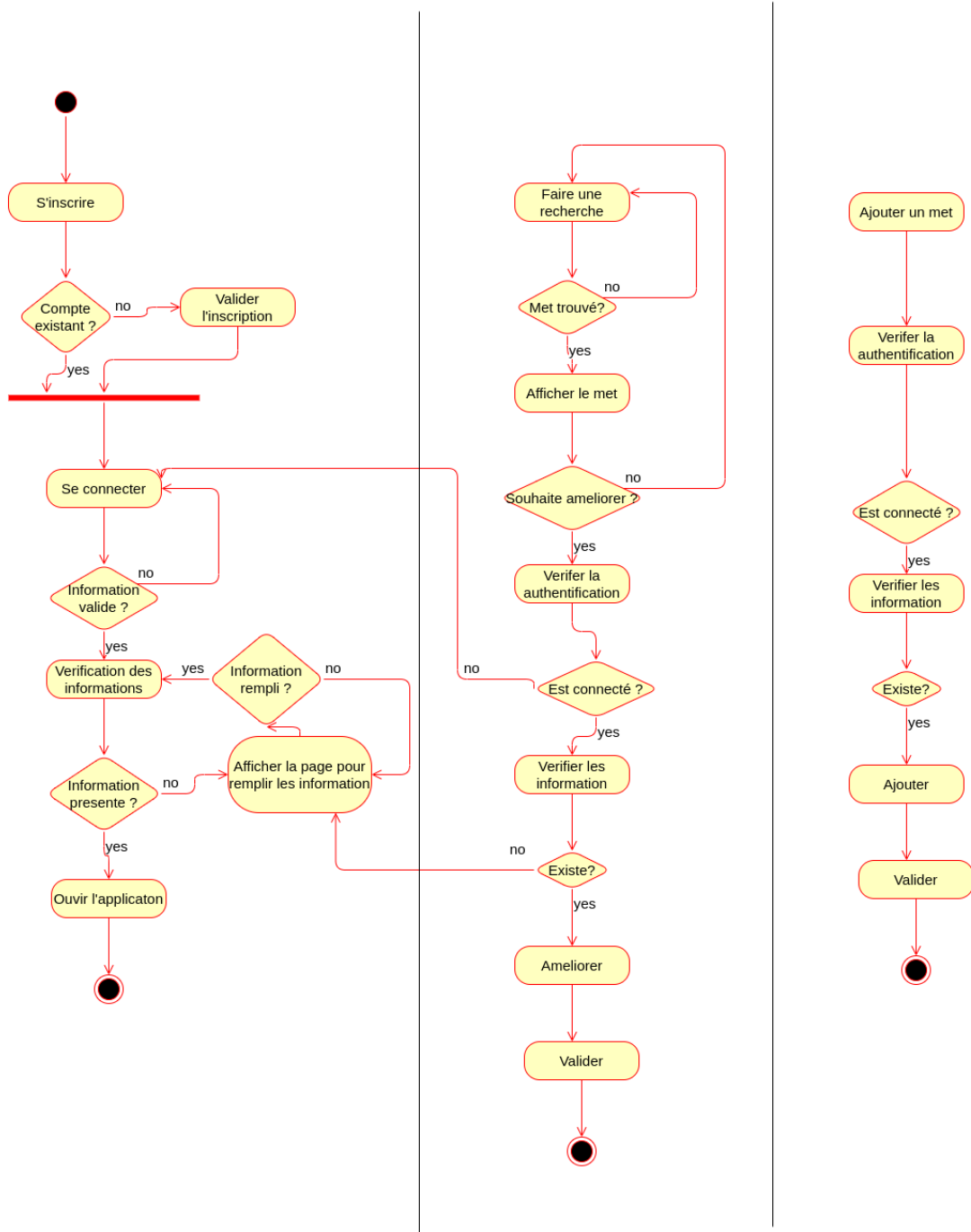
- l'application doit être web
- la vitesse des recherches: toutes les recherches s'effectueront rapidement et retourneront un resultat
- disponibilité: Camerpedia sera disponible à 98,6% du temps
- L'application doit être bien documenté
- l'ontologie doit être modulée afin qu'on puisse juste réutiliser un morceau de l'ontologie en cas de besoin
- L'application doit être intuitive et facile à utiliser
- le moteur de recherche doit être flexible c' est à dire on doit pouvoir compléter le résultat de notre recherche en cas d'erreur lors d'une entrée dans la barre de recherche

3. Identification des acteurs

comme acteurs nous avons recensé ici les acteurs suivants:

- le visiteur : est un utilisateur grand public simple, l'utilisateur grand public régulier qui peut être soit une personne malade
- un planificateur de cuisine (une personne qui cherche à composer des plats pour un événement)
- expert dans le domaine de la nutrition qui peut être un cuisinier ou un nutritionniste
- nutritionniste : est un expert du domaine qui se charge de la validation des apports des utilisateurs dans le système .
- l'administrateur du système qui a pour rôle de veiller sur les différentes activités du système .
- personne sous régime : est toute personne qui vient sur la plateforme pour avoir des astuces pour suivre un régime
- personne malade : une personne qui en fonction de sa maladie souhaiterait avoir des informations sur quel aliment consommé ou pas .

4. Diagramme d'activité de notre système



5. Cas d'utilisations et acteurs

Acteurs	Cas d'utilisations
planificateur de cuisine	se connecter, créer un compte, rechercher un met, inviter une personne, rechercher les plats consommés au cameroun
administrateur du système	se connecter, créer un compte, valider la mise à jour d'un met, valide le compte des experts, bloque des utilisateurs, invalide certains comptes, supprimer un compte
chef cuisinier	se connecter, créer un compte, créer un met, recherchez un met, voir la liste des mets, ajouter un régime
expert du domaine	se connecter, créer un compte, créer un met, recherchez un met, voir la liste des mets, peut valider des prescriptions fait par le système, peut valider un régime, peut exporter une ontologie, peut télécharger une ontologie qu'il a enrichi, définir un régime, donner les valeur d'un plat
nutritionniste	valider les mises à jour des experts du domaines
Personne malade	proposer un régime, s'enregistrer comme une personne malade, créer un compte, se connecter, avoir une recommandation de régime
Visiteur	ajouter un regime, peut télécharger une ontologie, rechercher un met, voir la liste des mets , voir les informations sur un met
personne pour le régime	proposer un régime, s'enregistrer comme une personne personne pour un régime, créer un compte, se connecter, avoir une recommandation de régime,

6. Description textuelle de cas utilisations

1. se connecter

☐ **Nom:** s'authentifier

- ☐ **Objectifs:** pouvoir se connecter à l'application
- ☐ **Acteurs principales:** les utilisateurs
- ☐ **Acteurs secondaires :** administrateur du système
- ☐ **Pré-conditions:** démarrer l'application
- ☐ **Scénario nominal:**
 - entrer son email et mot de passe au niveau du formulaire
 - vérifier que les informations sont exactes
 - accéder à l'application
- ☐ **Scénario alternatifs:**
 - reconnexion à la base de données
- ☐ **Post-conditions:**

2. Créer un met

- ☐ **Nom:** Créer un Met
- ☐ **Objectifs:** créer un met
- ☐ **Acteurs principales:** chef cuisinier , expert du domaine, nutritionniste
- ☐ **Acteurs secondaires :**
- ☐ **Pré-conditions:**
- ☐ **Scénario nominal:**
 - le système renvoie un formulaire
 - l'utilisateur renseigne les informations du met
 - l'utilisateur clique sur le bouton créer
- ☐ **Scénario alternatifs:**
 - choisir des éléments existant les informations
 - cliquez pour ajouter un élément
- ☐ **Post-conditions:**
 - nouveau met créé

3. Enrichir un met

- ☐ **Nom:** enrichir un met
- ☐ **Objectifs:** ajouter les informations sur un met
- ☐ **Acteurs principales:** expert du domaine/ chef cuisinier
- ☐ **Acteurs secondaires :** administrateur; super nutritionniste
- ☐ **Pré-conditions:**
 - le compte expert de domaine doit être créé
 - le met à enrichir doit être créé
- ☐ **Scénario nominal:**
 - cliquez sur rechercher le met à enrichir
 - entrer les données à ajouter pour la modification
- ☐ **Scénario alternatifs:**
 - cliquez sur sélectionner le met à enrichir
 - re entrer les informations à modifier

- ☐ **Post-conditions:**
-enrichissement terminé

4. Rechercher un met

- ☐ **Nom:** chercher un met
- ☐ **Objectifs:** cherchez un met dans le site
- ☐ **Acteurs principales:** tous les acteurs
- ☐ **Acteurs secondaires :**
- ☐ **Pré-conditions:**
-le met doit exister
- ☐ **Scénario nominal:**
 - entrer les informations pour chercher un met
- ☐ **Scénario alternatifs:**
 - entrez à nouveau les informations du met à chercher
- ☐ **Post-conditions:**
 - met trouvé

7. Diagramme de séquence du système

Les diagrammes de séquence de notre système sont représentés ci-dessous:

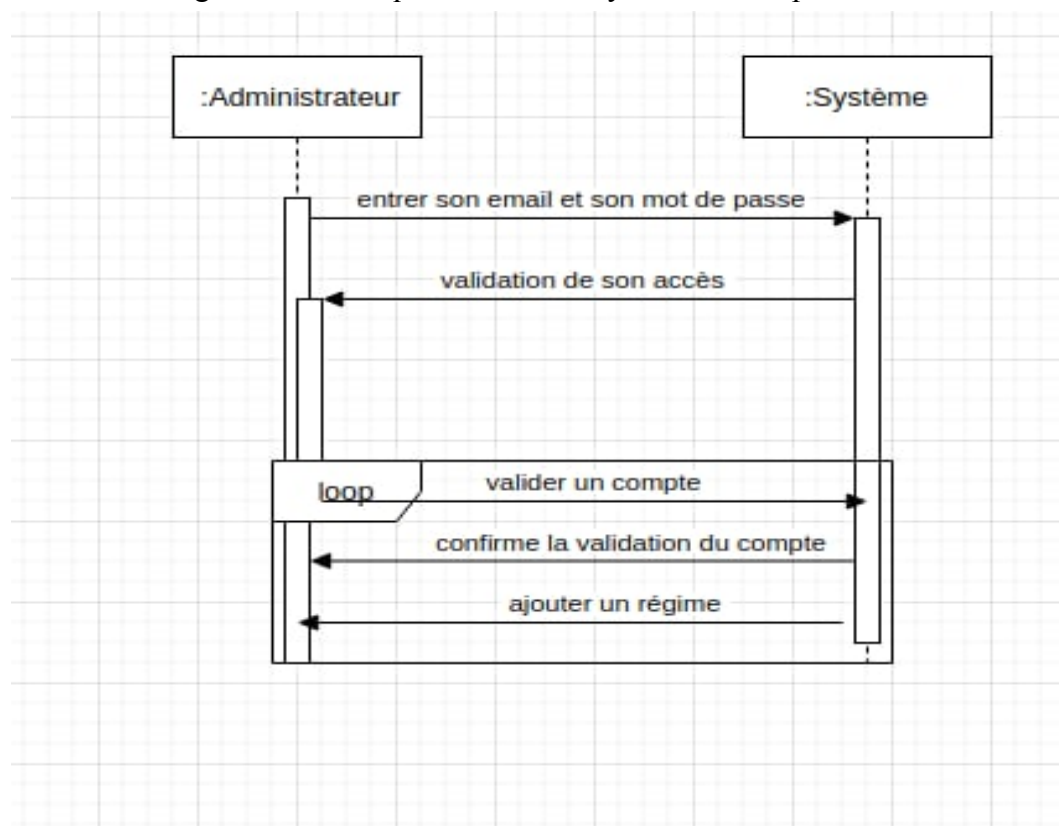


fig 1: Diagramme de séquence décrivant l'interaction de l'administrateur avec le système

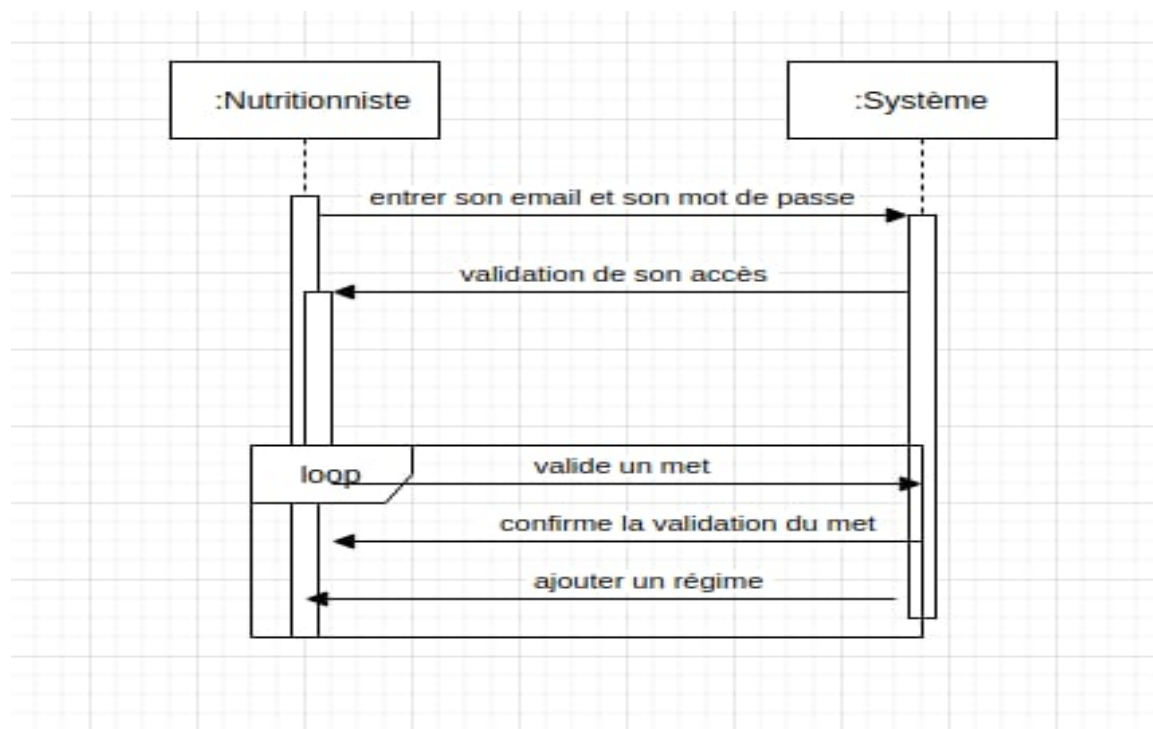


fig 2: Diagramme de séquence décrivant l'interaction du nutritionniste avec le système

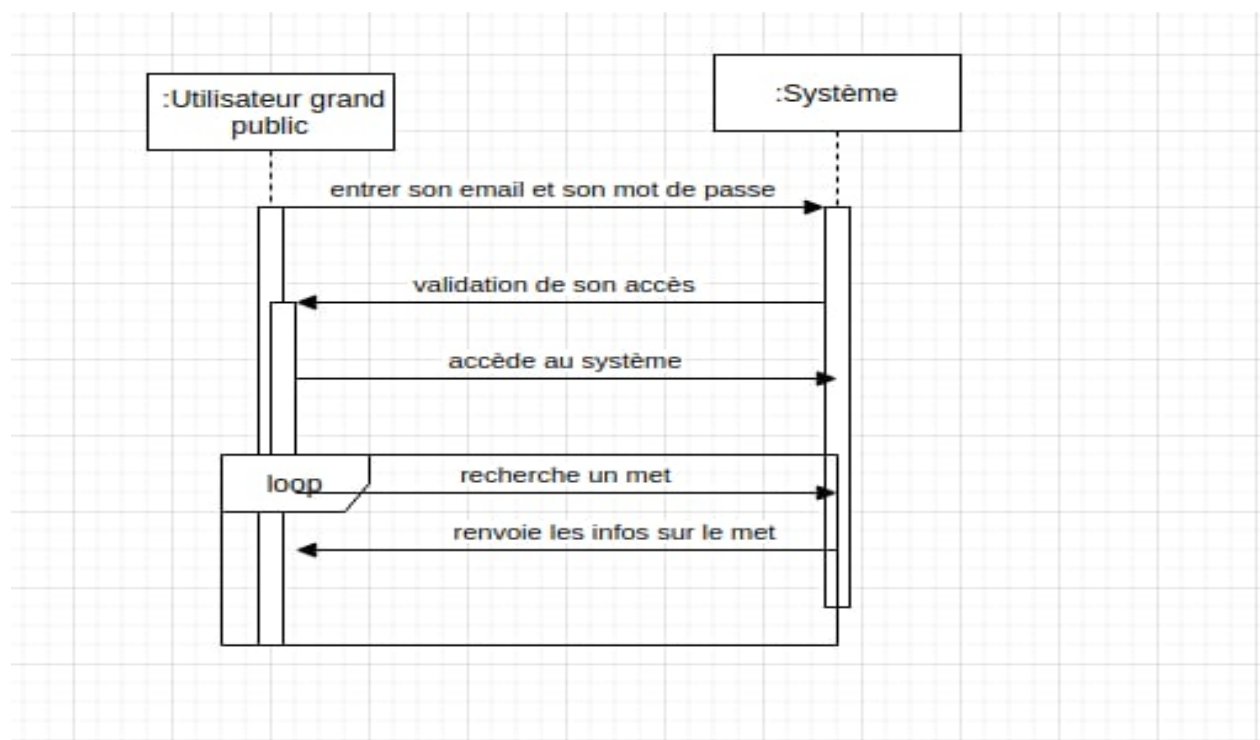
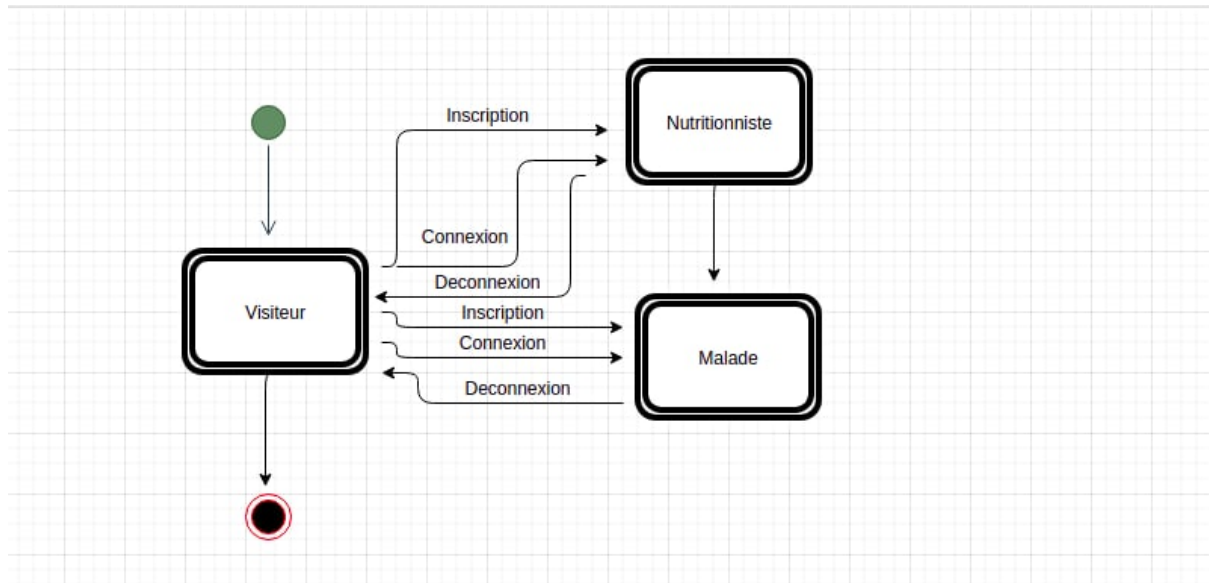


fig 3: Diagramme de séquence décrivant l'interaction d'un utilisateur grand public avec le système

8. Diagramme de cas d'utilisation des utilisateurs

les changements des états des utilisateurs est représenté dans le diagramme d'état transition suivant:



9. Définition du Product backlog

Le Product-Backlog comprend les éléments du backlog que nous allons suivre pour développer et valider notre ontologie. Ces éléments sont :

- **Spécification de l'ontologie** : il est question ici de définir le domaine et la portée de l'ontologie O4FN, qui ont été choisis, discutés et validés par notre équipe au cours de nombreux scrums meeting. Notre ontologie devra couvrir le domaine de la Santé et de la restauration et permettra aux utilisateurs de mieux choisir leurs repas.
- **Acquisition des connaissances** : il est question dans cette tâche d'acquérir des connaissances dans le domaine de la nutrition en se référant à des sources pertinentes. Dans notre cas, nous nous sommes servi d'Internet pour acquérir des connaissances, Mais aussi des réponses des cuisiniers dans notre entourage
- **Conceptualisation** : ici il est question de construire le modèle conceptuel de l'ontologie. Dans notre cas nous avons séparé utiliser une approche top-down pour structurer les concepts de l'ontologie
- **Implémentation** : ici, nous représentons notre ontologie O4FN comme un réseau de classes, propriétés et objets. Pour cela, **Protegé** qui est un logiciel open source ayant été développé par l'Université de Stanford a été utilisé comme outil de choix pour développer notre ontologie en OWL-DL (Description Logics)
- **peuplement de l'ontologie** : il sera question ici d'ajouter progressivement des instances à notre ontologie via une interface que nous développeront pour cet effet

10. Proposition de quelques maquettes utilisateurs

Nous avons proposé ci après une interface de gestion de principale par laquelle sera enrichi progressivement notre ontologie:

[Camerpedia](#) [Sparql](#) [Resources](#)

[Comptes](#)

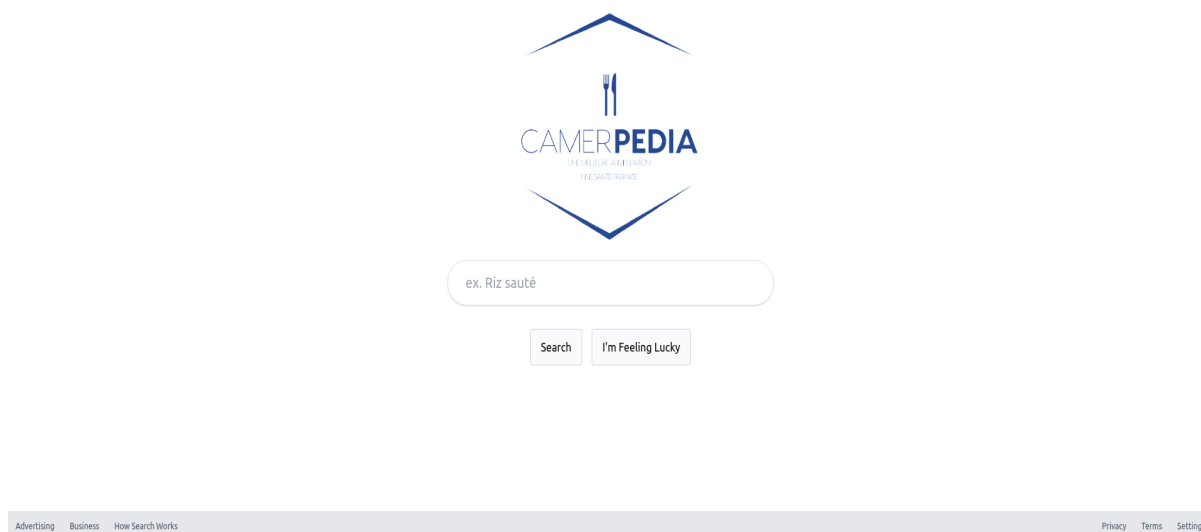


fig 1: interface principale

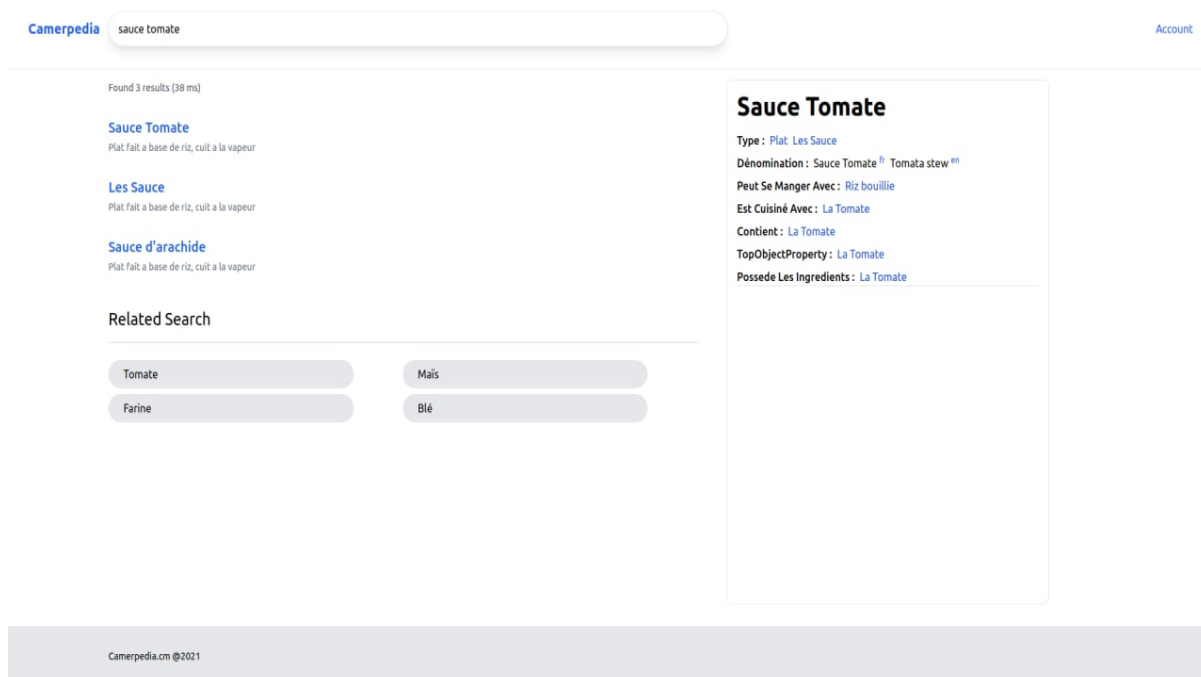


fig 2: interface présentant le résultat d'une recherche

Rice

Source from knowledge base, The food ontology

Lorem ipsum dolor sit amet consectetur adipiscing elit. Sunt accusantium maiores ab nisi voluptatibus in libero velit vitae corrupti, tenetur, dolores ad molestias magni architecto quod autem beatae sequi, eos labore ducimus. Rerum, natus earum eligendi in inventore dignissimos ipsa sint hic omnis doloremque quos? Perferendis voluptatibus odit praesentium delectus! Suscipit quam sapiente modi laudantium tempora, velit dolorem quo animi. Nobis a corporis quibusdam nisi inventore quisquam enim aut asperiores dolorum iure repellendus rerum veritatis vel modi saepe earum, aspernatur provident quasi, mollitia nam! Cum nulla nobis, harum aperiam adipisci exercitationem dolor repudiandae corporis, reprehenderit ex accusantium nam culpa? Alias minus vero ab repudiandae facere sit reprehenderit nulla, aut sunt accusantium molestias tempora illum ad beatae dolor, reiciendis culpa veniam omnis esse. Sapiente aut vitae odio adipisci sed, suscipit aspernatur corporis minus ad nemo voluptates architecto itaque praesentium accusantium neque numquam eveniet obcaecati nam tenetur autem quas eaque pariat illo magni? Voluptates, nostrum similique exercitationem atque accusamus unde consequuntur aliquid fuga cumque vitae voluptas cupiditate est fugiat, omnis dignissimos minus. Iste eum, cumque voluptate molestiae voluptatum, eos, quam facilis omnis perspicatis unde quod possimus non natus aut itaque eaque ex. Aut atque beatae explicabo corporis quia assumenda repellendus neque fugit laudantium odio repellat eaque ut iste quod molestias vero fuga, similique facilis voluptatem autem commodi! Magnam eveniet, culpa dolore dolor labore sit molestias voluptate accusantium aliquam possimus totam. Totam quod, doloribus porro quibusdam nostrum dolore maxime vitae adipisci, id consequatur voluptas. Nemo, nostrum optio, velit corrupti ad adipisci temporibus consectetur asperiores molestiae nihil maxime animi tempora neque? A ullam sequi beatae. Voluptates expedita maiores quia placeat ipsam, praesentium sit. Aliquam beatae explicabo voluptatem alias sed ipsa illo totam reprehenderit perferendis quas, iure consequuntur aspernatur! Vero est voluptatem sed omnis ratione repellendus doloremque, praesentium nam velit reprehenderit, error distinctio, laborum eos.

Nutrient

Ce repas contient xxx protéine et xxx yy

Nom local

Ce repas contient xxx protéine et xxx yy

Region d'origine

Ce repas contient xxx protéine et xxx yy

Recette

Riz sauté chez les bamenda

Lorem ipsum dolor sit amet consectetur adipiscing elit. Atque itaque mollitia deleniti cupiditate earum magnam a architecto iure nesciunt ab?

Camerpedia Wiki @2021

fig 3: interface wiki qui presente les informations sur une nourriture



Créer Un Compte!

Nom

Prénom

Entrez votre nom

Entrez votre prénom

Email

Entrez votre email

Type Compte

☐ Malade
 ☐ Nutritioniste
 ☐ Administrateur

Mot de passe

Confirmer mot de passe

Enregistrer

[Mot de passe oublié?](#)
[Avez-vous déjà un compte? connectez-vous!](#)

fig 4: interface d'enregistrement d'un utilisateur



fig 5: interface de connexion d'un utilisateur

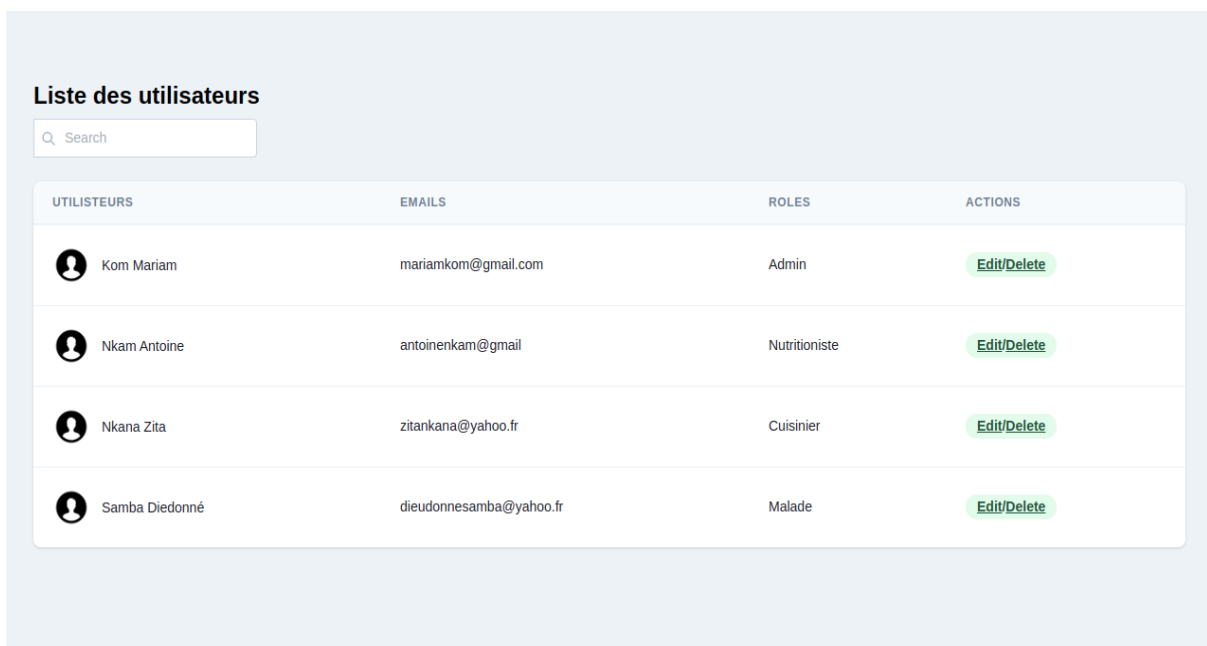



fig 6: interface de gestion des utilisateurs



Nouveau Triplet Ajouter un objet


Objets

liens

sujet

Ajouter

fig 7: interface d'ajout d'un triplet



Nouveau Triplet **Ajouter un objet**

URI

Rdfs:Label

Rdfs:Description

subClassof2

Foodsource2

Ajouter

fig 8: interface d'ajout d'un objet (instance)

11. Fragmentation du Product backlog en Sprints

a) sprint 1 : Installation des outils de manipulation pour l'ingénierie et l'intégration d'ontologies

- installer un Outil pour l'ingénierie des ontologies : dans notre cas nous choisirons le logiciel libre **Protege**
- installer un Outil pour le triple store : nous porterons un choix sur GraphDB
- installation d' une bibliothèque moteur de recherche : nous choisirons Apache Solr, une bibliothèque de logiciels de moteur de recherche gratuite et open source.

b) sprint 2 : Développement de l'ontologie

- définition du domaine et de la portée de l'ontologie
- recherche des ontologies existants
- Identification les sources de connaissances à utiliser pour construire l'ontologie
- Description sous forme de tableau des classes et les propriétés de chaque classe
- Définition de la hiérarchie des classes et les relations entre les classes
- Définir les axiomes et les règles.

c) sprint 3 : Intégration du triple store

- Importez et intégrez O4FN dans le triple store de notre choix
- Remplir O4FN avec des données imaginaires (au moins 100 triplets)
- Faire des requêtes SPARQL sur l'ontologie en considérant l'inférence

d) sprint 4 : intégration de l'ontologie

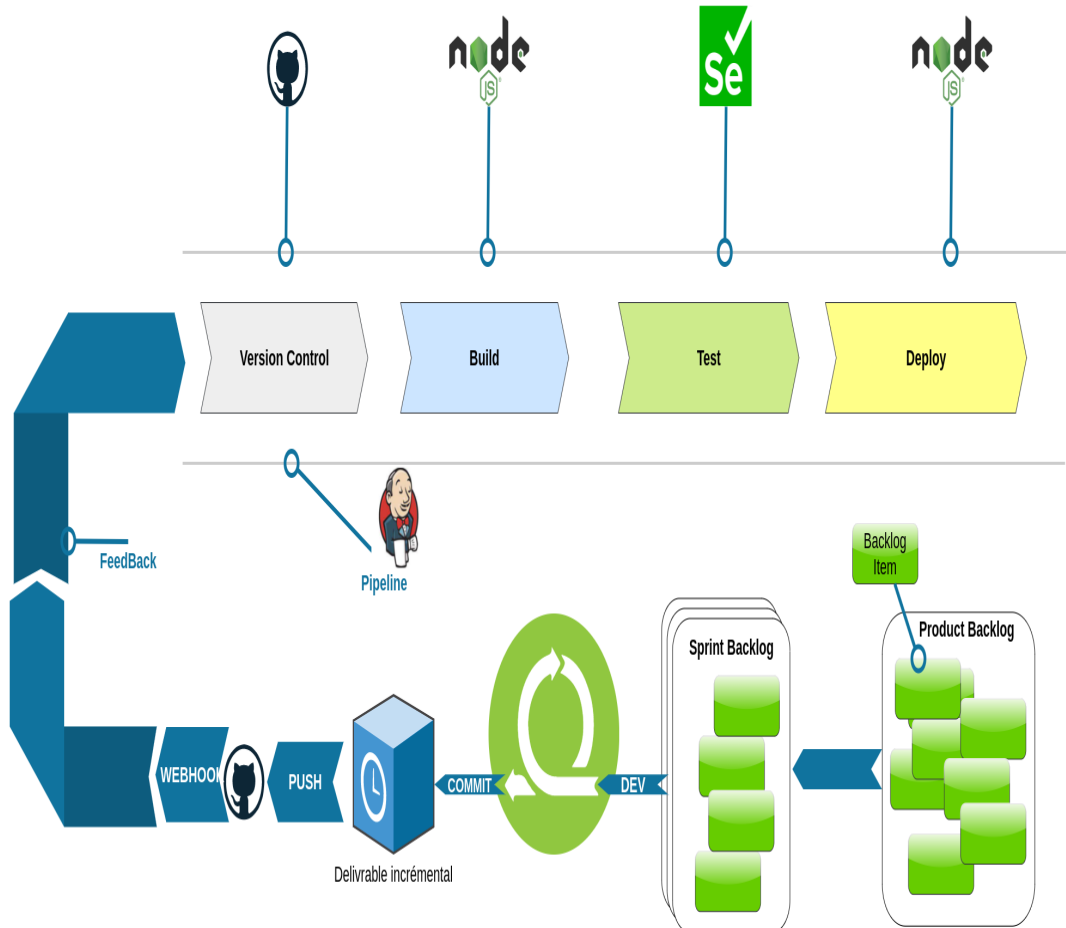
- Proposer une interface pour l'enrichissement de l'ontologie
- Proposer l'interface pour le peuplement et la recherche dans l'ontologie
- Proposer une interface pour la validation de l'ontologie

II) CONCEPTION DU LOGICIEL

1. Architecture Devops de développement du système

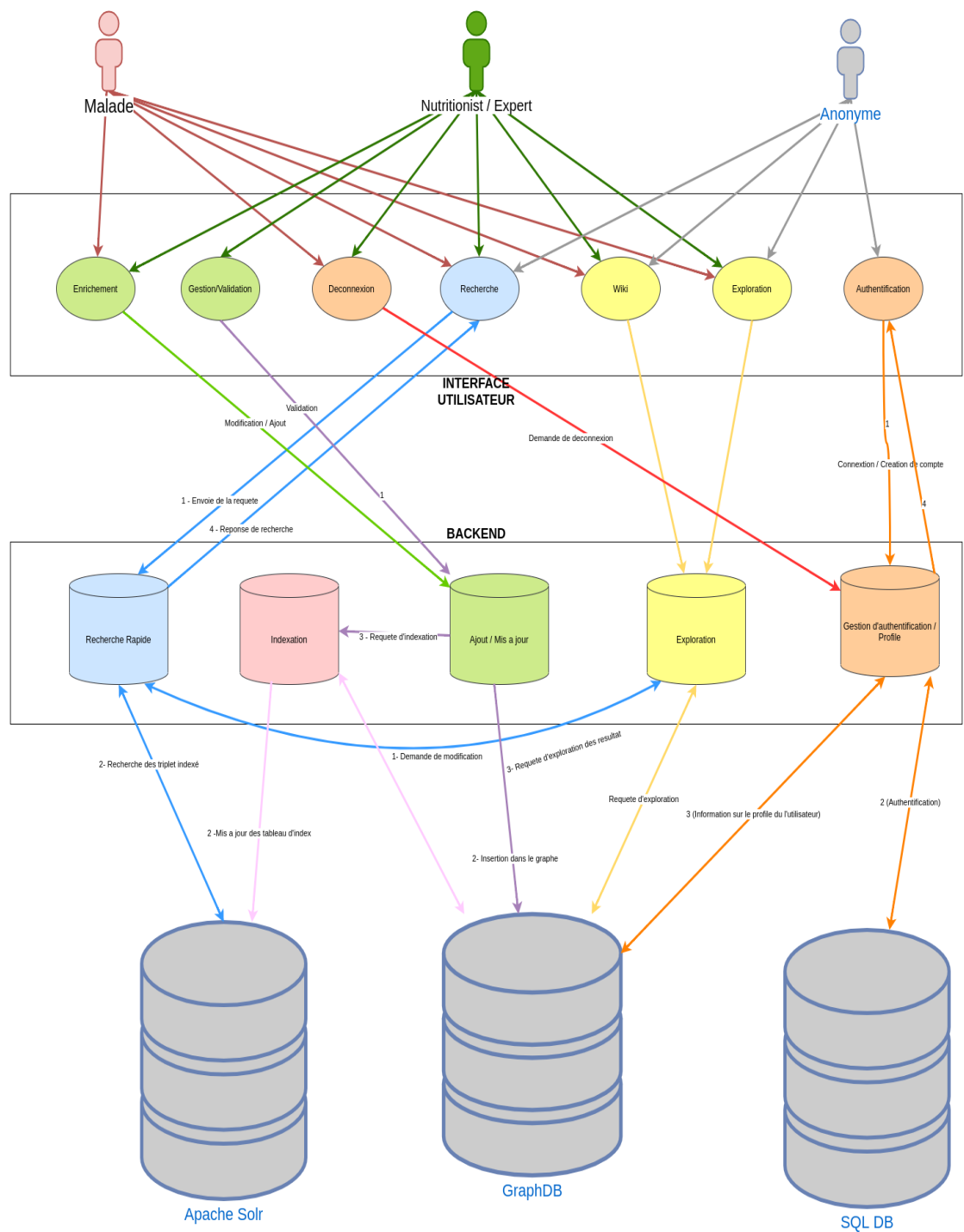
L'architecture devops de notre système est présenté ci dessous:

ARCHITECTURE DEV-OPS



2. Architecture du système

L'architecture du système CamerPedia est présentée comme suit:



Compte tenu du fait que le schéma de l'architecture n'est pas très claire, vous pouvez trouver une meilleure version de l'image à travers le lien suivant: <https://ibb.co/pRV8KhV>

3. Environnement de développement

Nous avons composé notre environnement de la manière suivante:

- **Protege:** c'est un logiciel open source qui a été développé par l'Université de Stanford a été utilisé comme outil de choix pour développer notre ontologie en OWL-DL (Description Logics). Nous utilisons sa version 5.50
- **GraphDB:** c'est un triple store compatible avec RDF et SPARQL et développé en JAVA. Nous utilisons sa version gratuite 9.8.0
- **Solr:** c'est un moteur de recherche textuelle développé également en JAVA. Il est connecter à GraphDB pour permettre de faire la recherche textuelle sur les données de la base de données graphe
- **Visual studio code:** c'est un IDE qui nous permet d'écrire notre code HTML/CSS et NodeJS
- **Langage de programmation:** nous avons utilisé pour le frontend **HTML**, **Tailwind CSS** (qui est un framework CSS), **JavaScript** et pour le backend nous avons utilisé **NodeJS** et **Koa** pour nous connecter au triple store GraphDB