

## 1 - Conception du site (fonctionnalité) :

1.1 Calcul de l'IMC, avec les données biométriques du l'utilisateur.

1.2 Calcul du besoin calorique.

1.3 Traçage et calcul des macros des aliments : Notre application permettra aux utilisateurs de suivre et de calculer le sucre total des aliments consommés au cours de la journée et les macros des aliments.

1.4 Enregistrement des aliments avec leurs Kcal et leurs macros : Les utilisateurs pourront enregistrer les aliments consommés avec leurs quantités respectives, ce qui leur permettra de visualiser dans leur compte la quantité nécessaire pour atteindre leur indicateurs macros maximal quotidien.

1.5 Dashboard complet : L'application disposera d'un tableau de bord comprenant certaines informations personnelles de l'utilisateur, des statistiques détaillées sur les aliments consommés (valeurs nutritionnelles, macros) et une vue globale des valeurs nutritionnelles respectées par l'utilisateur, la possibilité d'éditer son poids (afin d'avoir des nouveaux indices, IMC et macros à respecter, si le poids évolue).

1.6 Suggestion de réinitialisation quotidienne : Chaque jour, l'application interrogera l'utilisateur pour savoir s'il souhaite conserver les mêmes aliments enregistrés dans le tableau de bord ou s'il préfère recommencer avec une feuille blanche, elle doit lui demander aussi s'il souhaite conserver le même poids pour les calculs.

1.7 Vue d'ensemble sur les tendances à long terme (statistiques graphiques) : Les utilisateurs auront accès à une vue sur une semaine, un mois et une année, offrant une analyse approfondie de leur consommation quotidienne et de la conformité et le respect des macros et consommation de sucre.

Ces fonctionnalités visent à offrir une expérience utilisateur complète et personnalisée, aidant les personnes qui veulent suivre leurs consommations caloriques et leurs macros à mieux gérer leur alimentation et leur santé. En termes de design, nous souhaitons privilégier une interface intuitive, simple, facile à naviguer et centrée sur la facilité d'utilisation pour une audience variée.

## 2 - Suivi du projet :

### 2.1 Analyse et conception :

2.1.1 Réalisation du diagramme de cas d'utilisation UML pour définir l'architecture et les interactions entre les composants.

2.1.2 Élaboration d'un MCD (Merise) pour modéliser la structure des données.

2.1.3 Création d'un MLD pour formaliser le modèle logique de la base de données.

### 2.2 Sécurité à tous les niveaux :

2.2.1 Mise en place d'une check list pour respecter les normes de sécurité au niveau du front-end, back-end et de la base de données.

2.2.2 Utilisation de protocoles et de bonnes pratiques de sécurité pour garantir la protection des données sensibles des utilisateurs.

## 2.3 Structure MVC :

2.3.1 Implémentation d'une structure MVC (Modèle-Vue-Contrôleur) surtout pour la partie back-end du projet afin de garantir une organisation claire et modulaire du code.

## 3- Stratégies et technologies de développement :

3.1 HTML, CSS, JavaScript, PHP (framework symfony + Twig pour le front), SQL.

3.2 Utilisation d'une API open source pour récupérer les données des aliments.

## 4- Public cible:

4.1 Sportifs.

4.2 Les personnes qui souhaitent perdre du poids.

4.3 Les personnes qui souhaitent suivre leur consommation calorique.

4.4 Les personnes qui souhaitent suivre le total des macros consommées dans la journée (lipides, glucides, protéines, sucre).

4.5 Les personnes avec des problèmes chroniques, telle que les diabétiques (elles pourront voir le taux de sucre total consommé dans la journée), les personnes avec un cholestérol élevé.

UML:

Classe Utilisateur :

Utilisateur

1. id : int
2. pseudo : String
3. email : String
4. password : String
5. sexe : char
6. nom : String
7. prenom : String
8. naissance : Date
9. taille : float
10. tauxActivite : float

- Associations : A une relation "1, N" (1 à plusieurs) avec la classe **Suivi**, ce qui signifie qu'un **utilisateur** peut avoir plusieurs **suivis**.

Suivi

1. idSuivi : int
2. date : Date
3. kcalReference : int
4. poids : float

- Associations : A une relation "1" (exactement 1) avec la classe **Utilisateur**, et une relation "0, N" (0 à plusieurs) avec la classe d'association **Contient**.

Contient

1. quantite : int

- Attributs : quantité (représentant la quantité d'un **produit** dans un **suivi**).
- Associations : Fait le lien entre **Suivi** et **Produit** avec des cardinalités "1" côté **Suivi** et "0, N\*" côté **Produit**.

Produit

1. idProduit : int
2. libelle : String
3. kcal : int

- Associations : **Produit** une relation "0, N" (0 à plusieurs) avec la classe **Suivi** et la classe d'association **Contient**.

MCD:

Utilisateur

Entité Utilisateur :

1. ID (clé primaire)
2. Pseudo
3. Email
4. Password (crypté pour la sécurité)
5. Sexe
6. Nom
7. Prénom
8. Date\_Naissance
9. Taille (en cm)
10. Taux\_Activité\_Physique

Entité Suivi:

1. ID Suivi (clé primaire)
2. ID\_Utilisateur (clé étrangère)
3. Date
4. Calories\_Calculées\_Reference (avec les données biométriques)
5. Poids (s'il n'est pas mis à jour quotidiennement, garder le poids du jour d'avant)

Entité Contient:

1. ID\_Suivi (clé étrangère)
2. ID\_Produit (clé étrangère)
3. Quantité

Entité Produit:

1. ID Produit (clé primaire)
2. Libellé
3. Calories\_Produit (par 100gr)

Relations et Cardinalités

1. Relation Utilisateur - Profil
  - Un **Utilisateur** peut avoir un ou plusieurs **Suivis**.
  - **Chaque** Suivi est associé à un seul **Utilisateur**.
  - Cardinalité : 1 **Utilisateur** (1) peut avoir plusieurs **Suivis** (1,N).
2. Relation Suivi - Contient :
  - Un **Suivi** peut contenir plusieurs **Produit** à travers l'entité **Contient**.
  - La cardinalité serait de "1,N" pour **Suivi** vers **Contient** et de "0,N" pour **Produit** vers **Contient**.
3. Relation Contient - Produit :
  - Un **Produit** peut être lié à zéro ou plusieurs **Contient**.
  - La cardinalité serait de "0,N" pour **Produit** vers **Contient**

## MLD

**\*\*Respecte les 3 premières formes normales.**

### utilisateur

1. id\_utilisateur (clé primaire, int)
2. pseudo (varchar, 10)
3. email (varchar, 254)
4. password (varchar, 300) (SHA-256: Produit un hachage de 64 caractères hexadécimaux.)
5. sexe (varchar, 5)
6. nom (varchar, 15)
7. prénom (varchar, 15)
8. date\_naissance (date)
9. taille (int, 3)
10. taux\_activité\_physique (int, 1) (un chiffre compris entre 1 et 4 selon le niveau d'intensité)

### suivi\_quotidien

1. id\_suivi (clé primaire, int)
2. id\_utilisateur (clé étrangère, int, référence id\_utilisateur de la table utilisateur)
3. date (date)
4. calories\_calculées (int, 5)
5. poids\_du\_jour (int, 3)

### contient

1. id\_suivi (clé étrangère, int, référence id\_suivi de la table suivi\_quotidien)
2. id\_produit (clé étrangère, int, référence id\_produit de la table produit)
3. quantité (int, 4)

### produit

4. id\_produit (clé primaire, int)
5. libellé\_produit (varchar, 50)
6. calories\_produit (int, 5)