

Rapport du projet Web :

# *Health·Care*

**Groupe R9** : Rémi HUGUET, Johan CHAGNON,  
Matthis PILLONEL, Nicolas ETHEVE, Valentin LE STRAT

Mardi 22 mai 2018



## Table des matières

<b>I</b>	<b>Avant-Propos</b>	<b>3</b>
I.1	Pourquoi HealthCare	3
I.2	La solution HealthCare	3
<b>II</b>	<b>Fonctionnalités du site</b>	<b>3</b>
II.1	Gestion des utilisateurs	3
II.1.1	La table Users . . . . .	3
II.1.2	Inscription . . . . .	4
II.1.3	Édition du profil . . . . .	4
II.2	Nutri-Sélecteur	4
II.2.1	Description du reste de la base de donnée . . . . .	4
II.2.2	Calcul du Nutri-Score . . . . .	5
II.2.2.1	Choix du nutriment . . . . .	5
II.2.2.2	Trie des aliments . . . . .	5
II.2.2.3	Affichage des aliments . . . . .	5
II.3	Interface administrateur	5
<b>III</b>	<b>Front-End</b>	<b>6</b>
<b>IV</b>	<b>Sécurité</b>	<b>6</b>
<b>V</b>	<b>Amélioration</b>	<b>6</b>
V.1	SocialCare	6
<b>VI</b>	<b>Conclusion</b>	<b>7</b>

## Première partie

# Avant-Propos

### I.1 Pourquoi HealthCare

Notre projet part d'un constat simple : avoir une alimentation saine n'est pas facile pour tout le monde et beaucoup de personnes ont des intolérances ou des carences. Nous avons donc décidé de créer un outil pour permettre à tous d'avoir une alimentation qui correspond à ses besoins. Cet outil s'appelle «HealthCare». Nos objectifs de départ étaient donc les suivants :

1. Connaître les aliments adaptés à ses spécifications alimentaires en quelques clics
2. Pouvoir connaître le taux de nutriments présents dans les aliments
3. Recevoir des conseils personnalisés en fonction de sa situation

Notre outil s'adresse aux personnes ayant des carences, aux personnes ayant des intolérances, aux personnes souffrant de diabète mais aussi aux sportifs ou aux végétariens plus généralement toute personnes ayant des besoins nutritionnels particuliers.

### I.2 La solution HealthCare

Notre solution est la suivante : Après avoir donné les informations relatives à ses besoins, l'utilisateur se voit proposer les aliments qui y sont le plus adaptés. Les aliments sont regroupés dans une base de donnée avec des informations sur les différents nutriments qui les composent. Lorsque l'utilisateur rentre ses données, on affiche les éléments correspondant à sa demande, en les triant du plus pertinent au moins pertinent. Les administrateurs peuvent quant à eux ajouter des aliments manuellement pour enrichir la base de donnée. Notre site pose également les bases d'un réseau sociale basé autour de la nutrition, en fournissant un profil éditable et personnalisable.

## Deuxième partie

# Fonctionnalités du site

### II.1 Gestion des utilisateurs

#### II.1.1 La table Users

Chaque utilisateur enregistré par le site est associé à une instance de la table `Users` de la base de donnée. Cette classe comprend :

<code>user_id</code>	Le numéro d'identifiant de l'utilisateur, généré automatiquement et faisant office de clef primaire.
<code>name</code>	Son nom
<code>email</code>	Son e-mail
<code>username</code>	Son pseudo (fait office de clef secondaire)
<code>password</code>	Son mot de passe
<code>role</code>	(soit member, soit admin)
<code>adress</code>	Son adresse, city sa ville, country son pays, zip son code postal
<code>bio</code>	Un court texte où l'utilisateur se décrit lui-même dans le formulaire

### II.1.2 Inscription

L'inscription d'un utilisateur se fait via un champ `<form>` et des `<input type = "text"/>`. Pour l'inscription, seuls les champs `name`, `email`, `username` et `password` sont requis. Après l'envoi du formulaire, `Register.php` se charge de vérifier que les données entrées sont valides, et veille à ce que le `username` et l'`email` n'apparaissent pas dans la base de données. Si c'est le cas on echo un message d'erreur, sinon on fait appel à la fonction `Register` qui exécute la requête SQL :

```
INSERT INTO users (name, email, username, password, roles) VALUES ($name, $email, $username, $password, $member);
```

### II.1.3 Édition du profil

D'une manière sensiblement proche, une fois connecté la page `profile.php` permet à l'utilisateur de changer ses paramètres. Cette fois le formulaire est plus long (adresse, biographie...) mais reste facultatif, et au lieu de `INSERT` la requête SQL correspondante donne :

```
UPDATE users SET username=$valeur WHERE user_id=$user_id;
```

Et on procède ainsi pour chaque variable à changer (sauf `user_id`).

## II.2 Nutri-Sélecteur

### II.2.1 Description du reste de la base de donnée

Nous cherchons à ce qu'un utilisateur puisse obtenir des informations sur un certain aliment. Pour cela il nous faut donc une table `Aliments`. Elle se présente sous la forme suivante :

<code>alim_id</code>	Le numéro d'identifiant de l'aliment, généré automatiquement et faisant office de clef primaire.
<code>a_name</code>	Son nom (en français)

Comme il convient de jouer sur les nutriments pour obtenir la liste d'aliments, on a logiquement une table `nutriment`, qui se présente ainsi :

<code>nutri_id</code>	Le numéro d'identifiant du nutriment, généré automatiquement et faisant office de clef primaire.
<code>n_name</code>	Son nom (en français)

Enfin comme à un aliment on peut faire correspondre plusieurs nutriments et à un nutriment on peut faire correspondre plusieurs aliments, on a entre ces deux tables une

relation  $(n - n)$ , ce qui en SQL nous contraint à créer une troisième table faisant la liaison entre les deux, appelée `Aliments_Nutriments` :

<code>alim_id</code>	Le numéro d'identifiant de l'aliment concerné
<code>nutri_id</code>	Le numéro d'identifiant du nutriment concerné
<code>quantity</code>	La quantité de nutriment que possède l'aliment en question

Ici, `(alim_id, nutri_id)` forme la clef primaire de la table, `alim_id` référence `Aliments` et `nutri_id` référence `Nutriments`.

## II.2.2 Calcul du Nutri-Score

### II.2.2.1 Choix du nutriment

Le site affiche la liste des nutriments à ajouter à la sélection de l'utilisateur via un `<select>` qui contient la liste des nutriments répertoriés dans la base de données. Lorsque qu'un utilisateur ajoute un nutriment et appuie sur le bouton "Ajouter", le nom du nutriment est alors affiché en début de page ainsi qu'un `<input slider />` qui lui est associé. Le nutriment en question est alors supprimé de la liste des nutriments ajoutables.

Ajouter un nutriment revient à dire que l'on veut choisir à quel point on veut que la présence de ce nutriment impacte les résultats. Plus le slider est placé à droite et plus la teneur en ce nutriment est prise en compte. Si le curseur est placé tout à gauche, l'algorithme de tri des aliments est indifférent à la présence d'un tel nutriment.

### II.2.2.2 Trie des aliments

La valeur contenue dans les slider de chaque nutriment  $N$  est un coefficient entre 0 et 1. Les coefficients non spécifiés par l'utilisateur sont nuls. Pour chaque aliment  $A$ , il convient maintenant de calculer leur Nutri-Score. Pour cela, on applique la formule suivante afin que chaque nutriment soit considéré équitablement :

$$NutriScore_{user}(A) = \frac{1}{nbNutriment} \sum_{k=1}^{nbNutriment} coeff_{user}(k) \cdot \frac{q_k(A)}{max(q_k)}$$

On utilise ensuite la fonction `krsort()` de PHP qui permet de trier par ordre décroissant de nutri-score les aliments.

### II.2.2.3 Affichage des aliments

Muni de cette liste, il ne reste plus qu'à afficher les aliments après le formulaire par ordre décroissant de nutri-score. Les premiers aliments à apparaître sont ceux qui sont le plus pertinents en vertu des besoins spécifiés par l'utilisateur.

## II.3 Interface administrateur

Les droits d'un administrateur du site par rapport aux simples membres sont de pouvoir alimenter la base de données des aliments via une interface, appelée "Admin Dashboard" et accessible seulement par les utilisateurs ayant le rôle `admin`.

Là encore, la procédure d'ajout d'un aliment dans la base de données est similaire à celle de l'ajout d'un utilisateur, à ceci près qu'il faut ici réaliser également l'opération dans la table `Aliments_Nutriments` en fonction des données saisies.

## Troisième partie

# Front-End

La partie page d'accueil organisée construite avec HTML/CSS. Pour structurer la page nous avons utilisé des sections et un assez grand nombre de `div` pour bien gérer la modularité et l'organisation de nos contenus. De la même façon le CSS est très modulaire et la plupart des classes d'un tag sont composées de plusieurs sous-classes CSS qui se complètent.

Pour la partie application à savoir les pages de profils, d'admin, de Login et d'inscription sont en partie gérées grâce à bootstrap.

Enfin, le site propose un contenu Responsive grâce à media-query pour la page d'accueil et grâce à bootstrap pour les autres pages.

## Quatrième partie

# Sécurité

Le fait d'avoir des rôles admin et utilisateur nous oblige bien sûr à sécuriser les différentes pages du site. De ce fait, si un utilisateur essaye de se connecter à la page administrateur il sera immédiatement redirigé vers la page d'accueil.

## Cinquième partie

# Amélioration

### V.1 SocialCare

Une des améliorations prévues pour HealthCare est SocialCare. L'idée est de permettre à différents utilisateurs de se rencontrer via le site afin de pouvoir échanger sur leurs modes d'alimentation.

L'idée est de rassembler les utilisateurs ayant des besoins alimentaires similaires afin que ceux-ci puissent s'entraider pour leurs choix alimentaires.

Pour cela il faudrait comparer les préférences alimentaires des utilisateurs afin de pouvoir fournir une liste des profils similaires.

Cela permettrait d'ajouter un système d'amis sur le site qui pourrait éventuellement être complété par un forum où les utilisateurs pourraient discuter de leurs régimes alimentaires.

## Sixième partie

# Conclusion

Bien que nous n'ayons pas eu le temps de paufiner et étendre les fonctionnalités de notre site à d'autres service, le but premier du site est fonctionnel et offre une réelle réponse aux problèmes de nutrition que peuvent avoir certaines perosnnes. Ce fut pour nous l'occasion de mettre en place nos connaissances sur un projet de grande envergure et à partir d'un sujet choisi conjointement.