**eatSafe**

*The meal planner*

Table des matières

[*I – Acteurs & Avant-Propos* 4](#_Toc30026676)

[*II – Contexte* 4](#_Toc30026677)

[**1.** **Brief** 4](#_Toc30026678)

[**2.** **Objectifs** 4](#_Toc30026679)

[**3.** **Contraintes** 4](#_Toc30026680)

[**4.** **Livrables** 5](#_Toc30026681)

[*III – Conclusion* 5](#_Toc30026682)

[**1.** **En bref** 5](#_Toc30026683)

[**2.** **Reformulation du brief (contraintes, objectifs)** 5](#_Toc30026684)

[**3.** **Recommandation stratégique** 5](#_Toc30026685)

[**4.** **Liste des livrables définitifs** 6](#_Toc30026686)

[**IV – Structure et hiérarchisation de l’information, ergonomie, UX** 7](#_Toc30026687)

[**V – Réponse graphique** 8](#_Toc30026688)

[**1.** **Moodboard** 8](#_Toc30026689)

[**2.** **Charte graphique** 9](#_Toc30026690)

[**3.** **Maquettes, Webdesigns** 11](#_Toc30026691)

[**VI – Réponse éditoriale** 12](#_Toc30026692)

[**VII – Réponse fonctionnelle et technique** 13](#_Toc30026693)

[**1.** **Généralités** 13](#_Toc30026694)

[**2.** **Description détaillée page par page de l’ensemble des fonctionnalités** 13](#_Toc30026695)

[a. Architecture 13](#_Toc30026696)

[b. Backend 13](#_Toc30026697)

[c. Base de données 14](#_Toc30026698)

[d. Front-end 15](#_Toc30026699)

[e. Versionning 15](#_Toc30026700)

[f. Hébergement 15](#_Toc30026701)

[g. PWA (Progressive Web App) 16](#_Toc30026702)

[h. Framework & API 17](#_Toc30026703)

[i. Le site en version Bureau & version mobile non-installée 19](#_Toc30026704)

[**IX – Test et recettage** 20](#_Toc30026705)

[**1.** **Recettage** 20](#_Toc30026706)

[**2.** **AB Testing** 20](#_Toc30026707)

[**X – Mise en production** 20](#_Toc30026708)

[**1.** **Hébergement** 20](#_Toc30026709)

[**2.** **Serveur** 20](#_Toc30026710)

[**3.** **Mise en ligne** 20](#_Toc30026711)

[**4.** **Nom de domaine** 20](#_Toc30026712)

[**XI – Règles juridiques** 20](#_Toc30026713)

[**1.** **CGV / CGU** 20](#_Toc30026714)

[**2.** **Mentions légales** 21](#_Toc30026715)

[**3.** **Cookies** 21](#_Toc30026716)

[**4.** **Données personnelles** 21](#_Toc30026717)

[**5.** **Accessibilité** 21](#_Toc30026718)

[**XII – Enveloppe budgétaire** 21](#_Toc30026719)

# *I – Acteurs & Avant-Propos*

L’entreprise « eatSafe » créée en novembre 2019 une start-up portant le projet de création d’un assistant personnel de régulation nutritionnelle.

L’entreprise et l’application ne possédant pas de communauté, elle se chargera de la mise en application de l’acquisition et fidélisation d’une communauté autour du service et proposera diverses méthodes et recommandations stratégiques de communication pour atteindre le public souhaité.

L’entreprise réalisera des batteries de test en internes et publiera chaque version importante de l’application à une partie restreinte de la communauté pour obtenir des retours utilisateurs ainsi qu’un retour sur les bugs potentiellement présents dans l’application avant la mise à disposition de l’application à la totalité de la communauté.

Ce document est destiné aux acteurs étant susceptibles de se joindre au projet ainsi qu’aux potentiels clients souhaitant prendre connaissance des moindres détails de ce projet. Il sera constamment mis à jour en fonction de la direction que prendra le projet.

# *II – Contexte*

1. **Brief**

L’entreprise souhaite, sur le long terme, créer un service d’assistant à la régulation nutritionnelle destiné au quotidien dans l’optique d’améliorer le régime alimentaire des usagers ainsi que les conseiller, analyser leur consommation et leurs écarts pour ainsi enrichir leur expérience et prodiguer des conseils personnalisés.

1. **Objectifs**

Nos objectifs vont être de réaliser un MVP (Produit Minimum Viable) comprenant les fonctionnalités essentielles à une rapide mise sur le marché du produit afin de pouvoir commencer à créer une communauté autour de ce dernier.

Nous visons aussi un objectif B2B à destination des cantines, qu’elles soient en entreprises ou en écoles.

1. **Contraintes**

Nos contraintes sont multiples, l’entreprise souhaite sortir une première version du service courant Février 2020 pour à la fois commencer à créer une communauté, potentiellement démarcher des « Business Angels » (Il s’agit d’une personne physique, souvent un cadre d'entreprise en activité ou un ancien entrepreneur, qui décide d'investir une partie de son patrimoine financier dans des entreprises innovantes et à fort potentiel.) et pour commencer par la suite à récolter des données sur les parcours utilisateurs (pour de l’A/B Testing par exemple).

1. **Livrables**

Nos livrables seront tout simplement un produit minimum viable sous la forme d’une PWA (Progressive Web App ou Application web destinée aux smartphones) indiquant des recommandations nutritionnelles aux utilisateurs à chaque jour et une visibilité des recommandations générales par organisme. Et si possible y implémenter un système de création de recommandations personnalisé.

# *III – Conclusion*

1. **En bref**

L’entreprise souhaite donc s’établir en tant que leader de l’accompagnement nutritionnel au quotidien, personnaliser au maximum l’expérience utilisateur et créer une communauté autour de l’application sur le long terme.

1. **Reformulation du brief (contraintes, objectifs)**

Nos objectifs sont donc multiples. Créer un produit ou service répondant à ces attentes et développer dans un second temps une communauté autour du projet, le tout avec plusieurs contraintes : nous disposons uniquement de développeurs web et nous avons un temps imparti pour mettre en place un MVP : courant Février 2020.

1. **Recommandation stratégique**

**Identification des phases du projet :**

*(Information : Ces phases seront récurrentes, sous forme itérative par le biais d’une méthodologie SCRUM appliquée aux testeurs. On ne peut donc pas précisément les dater, néanmoins de grands axes s’en dégagent).*

PHASE 1 : Lancement du projet

Cette phase a pour but de démontrer la faisabilité du projet en fonction des idées émergeantes au postulat formé lors des premiers brainstorming.

* Brainstorming
* Analyse Marketing
* Étude de faisabilité

PHASE 2 : Création

Cette phase a pour but de définir concrètement les fonctionnalités qui formeront le cœur de l’application et ainsi de pouvoir permettre une meilleure vision globale du projet ainsi que de son statut actuel par rapport au statu attendu et de mieux visualiser la progression du projet.

* Arborescence
* Wireframes
* UI Kit
* Designs
* Assets optimisés
* Test d’assimilation utilisateurs (Ce test a pour but de prototyper l’application et de voir comment réagissent les utilisateurs face à cette dernière et s’ils comprennent bien la signification de chaque élément de l’application).

PHASE 3 : Recherche & Développement

Cette phase a pour but de démontrer quelle approche technique sera la plus viable pour l’application. De plus, une fois la démonstration établie, le lancement du développement et de l’intégration pourra avoir lieu.

* Étude des possibilités techniques
* Choix des méthodes et outils de travail
* Structuration globale de l’arborescence technique de l’application
* Lancement du développement du MVP

PHASE 4 : Testing & Feedback

Cette phase a pour but d’analyser et améliorer au mieux l’expérience utilisateur en prenant compte de leur comportement sur l’application et de leurs retours.

* Bug Bounty (Mise en place de récompenses pour les utilisateurs rapportant des bugs).
* Retours après utilisation
* Métriques et analyses du comportement des utilisateurs

PHASE 5 : Communication

Cette phase a pour but d’agrandir la communauté en augmentant la notoriété de la marque et de l’application.

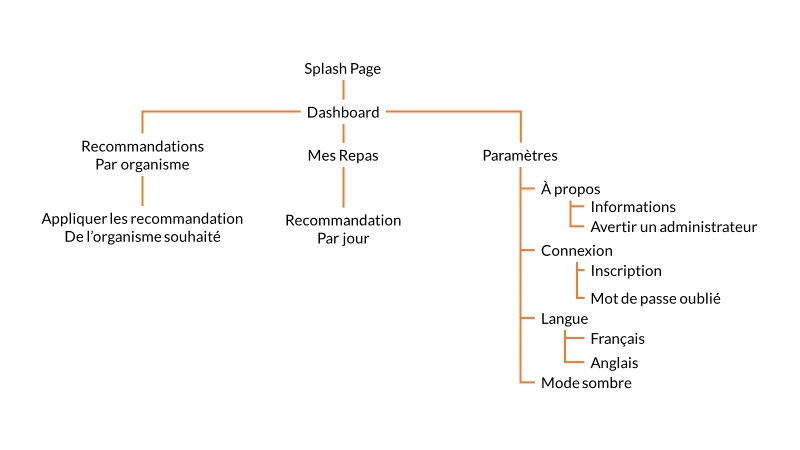
* Mise en place d’actions de communication (Géré par un tiers afin que nous nous concentrions sur le développement et mise en service de l’application).

1. **Liste des livrables définitifs**

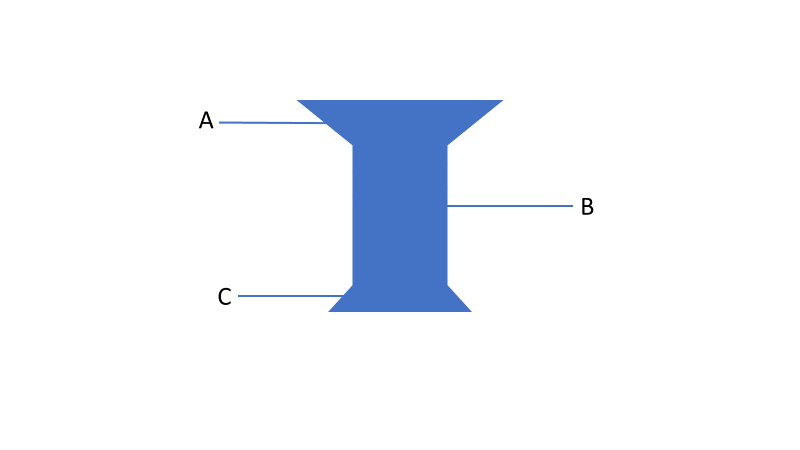
* Un cahier des charges complet
* Identité graphique complète
* Une application MVP fonctionnelle

# **IV – Structure et hiérarchisation de l’information, ergonomie, UX**

Arborescence & architecture de la donnée



Comme nous avons pu le voir précédemment nous utilisons une architecture de l’information que nous qualifierons de « lecture en double entonnoir ».

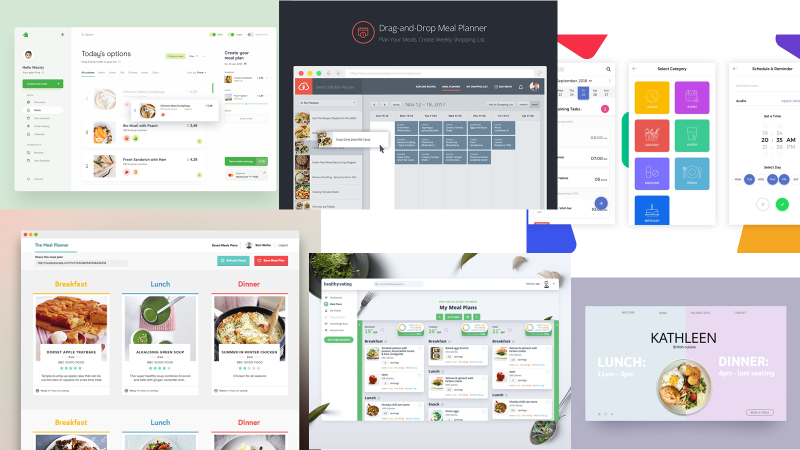
Dans ce tunnel de lecture, nous avons en premier lieu le point **A** qui regroupe les grands axes de l’application, c’est-à-dire ses principales fonctionnalités (Dashboard / Recommandations / Repas / Paramètres). Elles sont accessibles à tout instant par l’utilisateur et en un clic.

Ensuite vient le point **B** qui va axer l’expérience de l’utilisateur dans la continuité de la page au sein de laquelle il se situe et lui proposer des actions en rapport avec le contenu de cette page. Cette partie est accessible en un à deux clics et donc facilement atteignable.

Il pourra interagir au sein de ces fonctionnalités pour atteindre d’autres fonctionnalités plus avancées qui seront liées uniquement à la fonctionnalité parente observée au point B. Nous serons alors dans la partie **C** de ce double entonnoir qui serait plus semblable à une bulle regroupant toutes sortes d’informations non-essentielles pour un utilisateur lambda puisque plus difficilement accessible. En effet, les fonctionnalités requérant deux à trois clics (ou plus) seront considérées comme appartenant au point C.

# **V – Réponse graphique**

1. **Moodboard**



1. **Charte graphique**



**La signature :**

La signature est une véritable identité visuelle qui permet à la marque d’être reconnue instantanément, car elle est facilement visible et lisible. Celle-ci pourra être déclinée pour les signatures de mails, de courriers ou encore sur des cartes de visites. Elle servira aussi en web à la création du favicon du site.

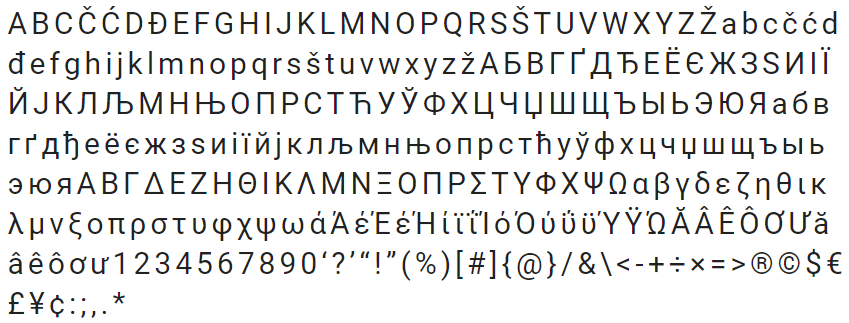
Ici, nous avons choisi de réaliser un logotype bien spécifique pour représenter le logo :

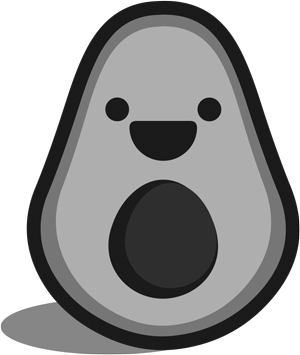
Le logo représente premièrement un avocat sympathique mais sur le fond expose le thème du « healthy » à la fois par le biais du vert et le biais du fruit.

**La charte typographique :**

Nous utiliserons la police Roboto qui est fine et élégante, dotée de nombreuses graisses mais ne se classe pas comme une police trop classieuse et reste sur un ton léger, ce qui s’accorde parfaitement au ton de la marque.

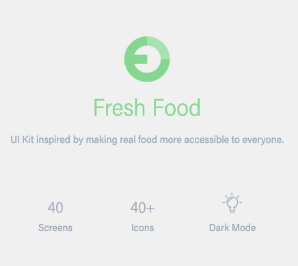
De plus, elle possède une vaste gamme de caractères compatibles et se place donc en tant que police « multi-Lang-friendly ». C’est aussi une « Variable Font ». Nous allons ici pouvoir observer notre kit de couleur, formes et déclinaisons.



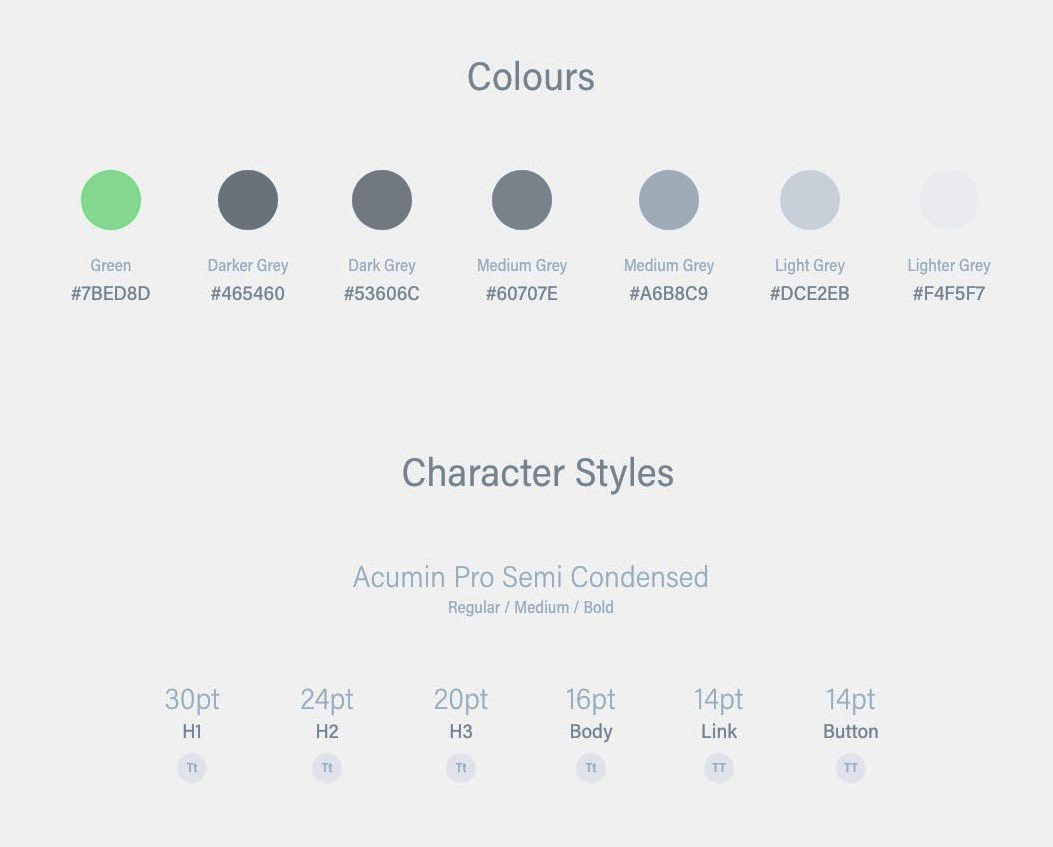


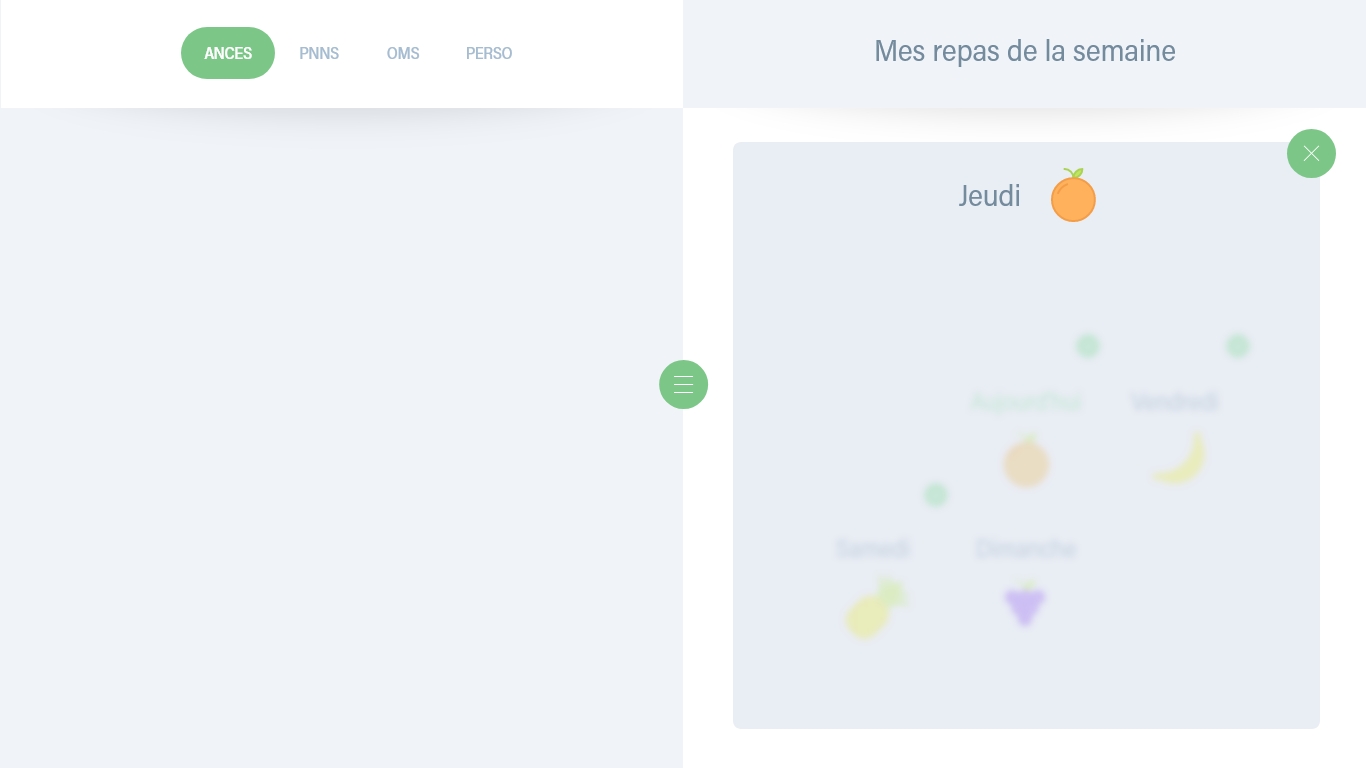
**Les documents administratifs :**

Il arrivera, au cours de l’impressions de certains documents que nous ne puissions pas mettre le logotype au format couleurs. (ex : document juridique). Pour palier cela, nous avons mis au point un typogramme plus sobre pour garder un air sérieux en interne ou face à des partenaires potentiels.



Nous avons choisi d’utiliser l’UI Kit « Smart Home » qui nous permet d’avoir des assets déjà définis ainsi qu’une ligne directrice pour la partie graphique du projet et ainsi maintenir une certaine cohérence globale. En voici quelques extraits.



1. **Maquettes, Webdesigns**

# **VI – Réponse** **éditoriale**

Nous axerons le ton et la communication autour d’un ton léger. Il est important d’éviter tout sentiment d’urgence, d’obligation et de générer la moindre frustration auprès de l’utilisateur.

En accord avec le ton et le style rédactionnel présenté ci-dessus, nous travaillerons donc autour de cinématiques légères et agréables avec une communication directe et personnalisée à l’utilisateur ou utilisatrice tout en s’adaptant au milieu digital actuel.

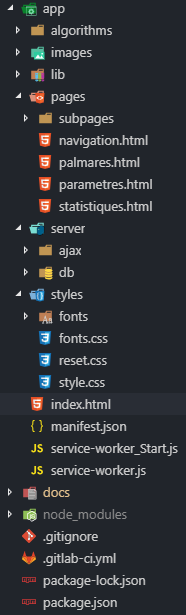
Nous recommandons ainsi l’utilisation éventuelle d’emojis à petite dose pour agrémenter les réseaux sociaux. Nous recommandons aussi l’utilisation de phrases exclamatives qui impacteront le lecteur, couplées à l’utilisation de questions incitant la réaction de la communauté.

# **VII – Réponse fonctionnelle et technique**

1. **Généralités**

L’application sera développée from scratch (à partir de rien) avec notre propre architecture et sera une PWA (Progressive Web App).

1. **Description détaillée page par page de l’ensemble des fonctionnalités**
   1. Architecture



Nous utilisons un « emballage d’application » nommé « app ». Nous avons aussi un dossier de documentation qui contient les outils nécessaires à la compréhension des fonctionnalités de l’application et à l’ajout de contenu au sein de cette dernière ; ce contenu reste privé et uniquement accessible au personnel de la production de l’application.

« app » contient plusieurs dossiers dédiés à l’organisation des modules de l’application :

« algorithmes » contient tous les scripts nécessaires au bon fonctionnement des calculs de l’application.

« images » permet d’organiser les médias de l’application.

« lib » contient toutes les bibliothèques dont l’application a besoin pour fonctionner.

« pages » contient tous les templates de page aussi appelé vues de l’application.

« server » contient toutes les fonctionnalités BDD et ajax nécessaires à la sauvegarde des données des utilisateurs.

« styles » ient la partie visuelle de l’application.

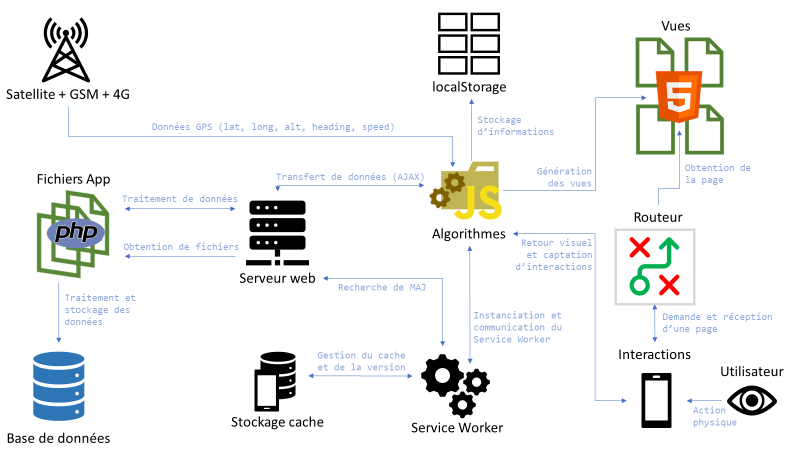
* 1. Backend

Tous les algorithmes sont traités côté utilisateur pour optimiser les allers-retours serveur et permettre aux usagers de pouvoir bénéficier de la majorité des fonctionnalités du projet sans pour autant avoir à être connecté ou dépenser de la bande passante.

Néanmoins, toutes les fonctionnalités communautaires ou impliquant la sauvegarde des données de conduite des usagers devront passer par un serveur distant.

Pour cela nous utilisons un système de communication en Ajax permettant la communication avec des pages PHP qui envoient toutes les données nécessaires au serveur.

Ainsi, l’usager voit au sein de son application des pages html et peut donc les ouvrir hors ligne sans que cela cause de problème à contrario d’une page PHP, qui ne peut pas être ouverte directement en tant que document dans un navigateur.



* 1. Base de données

Concernant le système de bases de données, nous utilisons plusieurs systèmes : En première couche nous agissons sur la partie offline et côté utilisateur par le biais de l’API native localStorage qui agit en tant que base de données propre à l’utilisateur et qui ne s’efface que quand l’utilisateur vide le stockage de son navigateur ainsi, même si l’utilisateur vide son cache navigateur, ces informations ne seront pas perdues.

Néanmoins nous ne stockons pas beaucoup d’informations dedans. Nous stockons simplement les préférences de l’utilisateur dans les paramètres de l’application ainsi que, s’il en a, ses informations de connexion hachées et donc indécryptables.

Dès que l’utilisateur aura accès à internet et qu’il sera inscrit et connecté, nous pourrons stocker toutes les informations liées à son profil utilisateur ainsi que ses données de conduite par le biais d’une base de données externe hébergée chez 1&1 Ionos et au format MySQL regroupant à ce jour deux tables distinctes qui sont la table utilisateurs et la table trajets.

* 1. Front-end

La totalité de la partie front-end de l’application est rédigée en Javascript et en CSS néanmoins un passage en SASS est prévu courant août afin d’optimiser le temps d’intégration et permettre une meilleure compatibilité navigateur ainsi qu’une propreté du code bien nette est accessible et ergonomique à une personne souhaitant rapidement le modifier.

De plus nous mettons l’accessibilité en avant en fournissant un mode clair et un mode sombre ainsi que la possibilité de changer les préférences d’unité de temps, d’unité de distance et vitesse ainsi que la langue d’affichage des dates et la langue d’affichage du texte.

* 1. Versionning

Le versionning est géré par git, envoyé par le biais de Visual Studio Code et hébergé puis versionné sur GitHub.

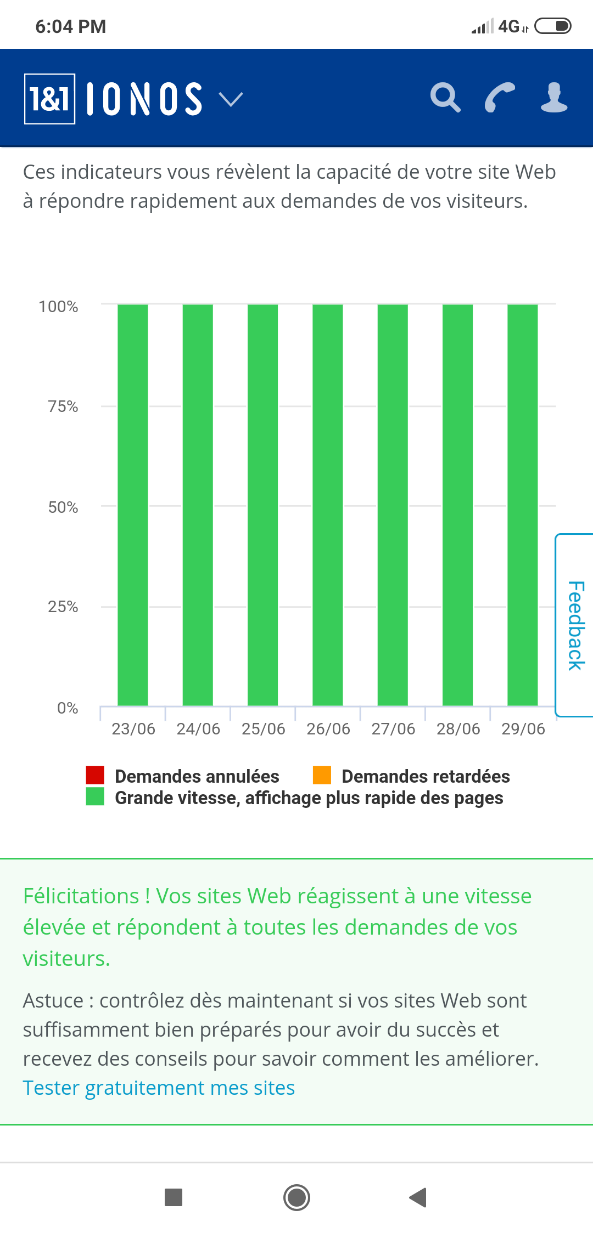
* 1. Hébergement

L’hébergement actuel est un hébergement web chez 1&1 Ionos avec un système de BDD sous **MySQL 5.5** avec une allocation stockage jusqu’à 1024 MB (1GB) proposant jusqu’à 25 BDD disponibles (si nécessaire).

Concernant l’allocation Espace Web, nous sommes limités à 262 144 fichiers pour une taille de 250GB.

Le stockage peut être géré par les biais d’une connexion **SSH** ou **SFTP** pour une totalité de 20 utilisateurs max différenciables.

La version **PHP** du serveur est en 7.3 pour être à jour avec les dernières mises à jour PHP et permettre un support 24h/24 7j/7 par les services d’assistance 1&1 Ionos.

En **hardware** (partie physique), le serveur est mutualisé est dispose du niveau de performance 1 de chez 1&1 Ionos qui comprend une RAM MAX de 2.5Go, 300 Process/mn, une mémoire limite à 512 Mo et après nos tests, nous avons **0% de perte ou de ralentissement du trafic**. Néanmoins, si cela s’avérait nécessaire, plusieurs options sont possibles comme prendre le niveau de performance supérieur pour améliorer les temps de réponse du site. L’option d’un serveur dédié serait envisageable si jamais le trafic devenait trop important ou que des ralentissements survenaient alors qu’il n’y aucune cause apparente ou étant à notre charge.

L’hébergement bénéficie d’un système Site Lock qui protège le site contre les logiciels malveillants, les accès non-autorisés et les tentatives de piratage

Il bénéficie également d’un système de CDN (Content Delivery Network) qui accélère le temps de chargement de tous les médias, scripts et bibliothèques du site.

* 1. PWA (Progressive Web App)

L’application est développée sous la forme d’une Application Web Progressive, il s’agit tout simplement d’un site web combiné à certaines technologies lui permettant ainsi de s’enregistrer sur un écran d’accueil et de s’ouvrir comme une application native sur l’écran de l’usager. Ainsi les PWA ont les avantages d’une application native sans les inconvénients.

Le site s’affiche en plein écran, sans la barre du navigateur. Le site fonctionne parfaitement hors ligne.

Son coût et temps de développement par rapport à une application native son infimes.

La maintenance constante des packages de l’application n’existe pas puisque nous n’utilisons pas les systèmes de dépendances propres à des langages comme Java Android.

Son poids et donc occupation de l’espace de stockage est infime. Pour exemple : Twitter dans sa version native occupe plus de 100 Mo et à contrario, dans sa version PWA moins de 1 Mo.

L’actualisation et mise à jour de la PWA se fait automatiquement sans que l’utilisateur n’ait le besoin de mettre à jour manuellement. Nous pouvons donc nous passer de la longue et inutile phase d’approbation du Google Play Store ou du Apple Store, nous ne sommes donc plus dépendants d’eux.

L’utilisateur a le choix d’utiliser l’application sous forme d’un site responsive ou de l’installer directement dans son téléphone sous la forme d’une application.

La PWA dispose aussi des fonctionnalités avancées des applications natives en ayant par exemple accès à la plupart des fonctionnalités intrinsèques au téléphone, comme les différents capteurs par exemple.

Le fait que la PWA soit Google Friendly elle contribue au meilleur référencement du projet (SEO Friendly). De plus, une PWA, pour fonctionner doit obligatoirement être en HTTPS donc sécurisé pour l’utilisateur.

L’application peut être installée sur n’importe quel support, que ce soit Android, iOS, Windows, Linux, Raspberry, etc...

Nous partons néanmoins de l’affirmation suivante : L’application sera installée à partir d’un navigateur Chrome dans 95% des cas, nous garantissons un fonctionnement total si l’application est installée par le biais de Google Chrome mais supportons tout de même l’application sous les deux dernières versions de Chrome ainsi que Mozilla (Firefox, Nightly, Developer).

* 1. Framework & API

Un Framework javascript est une bibliothèque de fonctions pré écrites en javascript facilitant le développement d’applications fondées sur javascript.

Nous utiliserons ici Piodjio un framework CSS permettant d’optimiser le temps de développement par le biais de mise en place de classes CSS.

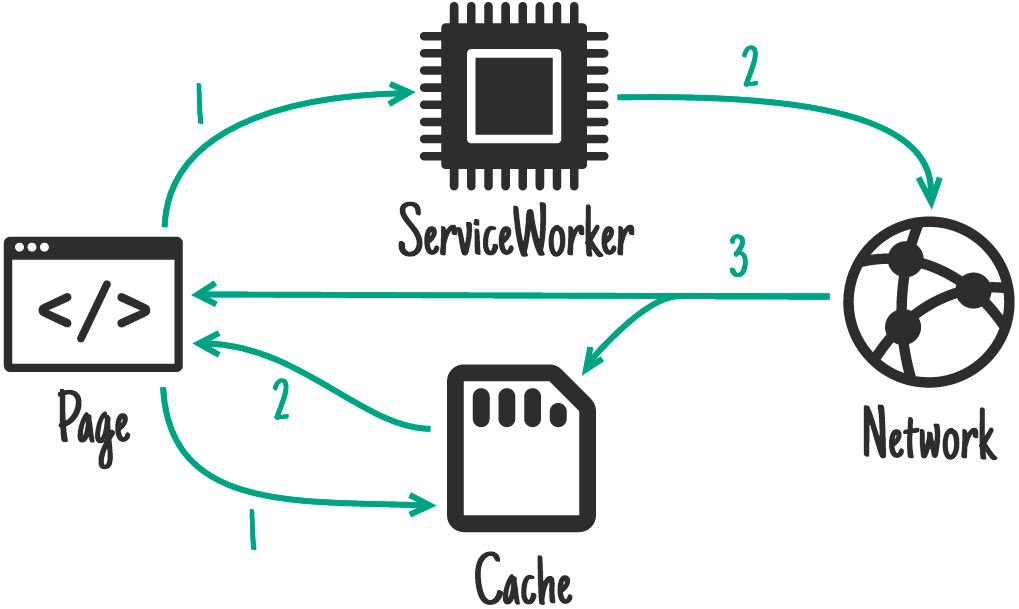
API est un acronyme pour Applications Programming Interface. Une API est une interface de programmation qui permet de se « brancher » sur une application pour échanger des données. Une API est ouverte et proposée par le propriétaire du programme.

En l’occurrence, nous utilisons ici uniquement des API dites W3CR c’est-à-dire recommandées par le W3C (le groupe décidant des fonctionnalités qui doivent être implémentées dans les navigateurs web).

Toutes ces API font donc l’objet d’une présence directement implémentées dans les navigateurs et ne requérant donc aucune installation ou téléchargement puisque directement accessibles.

* + 1. Service Worker API

Le Service Worker (ou opérateur de services) permet le bon fonctionnement du site en tant qu’application, il permet de mettre en cache les contenus du site et de choisir la méthode de chargement de l’application.



* + 1. Service Worker API

Nous utiliserons aussi une API non validée distribuant des datasets liés à la planification de repas prodiguée avec le projet initial.

Voici ci-dessus le fonctionnement de notre Service Worker :

Au chargement la page on appelle à la fois le cache (qui a préalablement sauvegardé nos pages pour y accéder hors-ligne) ainsi que le Service Worker qui lui est accessible en ligne.

Donc par défaut l’application ouvre la dernière version enregistrée de l’application et si l’application parvient à atteindre en arrière-plan le Service Worker, elle va voir si l’application a eu des nouveauté ou changements en se connectant au réseau.

S’il n’y a pas de changement, rien ne change. En revanche s’il détecte un changement de version du Service Worker une fois connecté au réseau, le réseau injectera en arrière-plan les nouvelles données dans le cache de l’application et pourra ensuite mettre à jour l’affichage de l’application (au prochain lancement de l’application dans notre cas).

* + 1. Web Storage API

Cette API fournit des mécanismes par lesquels les navigateurs peuvent stocker des paires (clé, valeur intrinsèque), d’une manière bien plus intuitive que celle des cookies et permettant un stockage et un accès après nettoyage du navigateur.

Nous nous en servons pour utiliser les fonctionnalités de stockage local qui servent à stocker les informations de connexion de l’utilisateur ainsi que ces préférences en matière de paramètres.

Le tout en gardant les données même hors connexion ou après le nettoyage des cookies et fichiers temporaires.

* + 1. Cache API

Nous utilisons cette API au sein du Service Worker.

Elle permet de sauvegarder les pages, scripts et assets (styles et médias) pour permettre l’ouverture de l’application instantanément et sans connexion internet.

* 1. Le site en version Bureau & version mobile non-installée

Le site en version bureau sera une page vitrine proposant l’installation de l’application et sera une simple page HTML stylisée en CSS.

Pour cela nous chargeons quand même l’application (afin de bénéficier de son effet SEO friendly) mais instaurons une « media query » qui au-delà de telle ou telle valeur fait que nous affichons cette page.

Concernant la version mobile, si la PWA est installée, nous redirigeons dans l’application, sinon nous proposons au travers de la page vitrine de d’installer directement ou bien de tester avant d’installer.

# **IX – Test et recettage**

1. **Recettage**

Un système de Bug Bounty va être implémenté pour permettre aux « découvreurs de bugs » de bénéficier d’avantages dès lors qu’ils découvrent un nouveau bug.

Cela nous permet d’assurer l’absence de bug et la remontée de ces derniers dans les plus brefs délais.

Les rapports de bugs se feront par le biais d’un bouton dans la page « À propos » de l’application.

1. **AB Testing**

Aucune phase d’AB Testing n’est à ce jour définie.

# **X – Mise en production**

1. **Hébergement**

L’application est hébergée (pour le moment en sous-domaine) chez 1&1 Ionos.

Leurs prix sont attractifs. Le support est réactif et disponible 24H/24 et 7J/7. De plus il s’agit d’un des hébergeurs français majeurs (Concurrent direct de OVH).

1. **Serveur**

Nous sommes actuellement sous serveur mutualisé car nous ne ressentons pas le besoin d’avoir des machines avec une configuration très spécifique et n’avons actuellement pas de recrue d’essence du trafic moyen. Néanmoins si le besoin se ressentait (augmentation drastique du trafic ou voisin interne au serveur ralentissant le trafic), il serait intéressant de se tourner vers une hébergement dédié mais attention : le coût d’un tel serveur peut s’avérer très élevé.

1. **Mise en ligne**

Le site est déjà en ligne depuis plusieurs semaines et des mises en ligne mensuelles de nouvelles fonctionnalités, correctifs ou mise à jour événementielles sont à prévoir.

1. **Nom de domaine**

Le nom de domaine va être réservé en premier ainsi que par la suite les noms de domaines s’en rapprochant pour éviter le cybersquatting et générer des redirections supplémentaires.

# **XI – Règles juridiques**

1. **CGV / CGU**

Aucun système d’achat In-App n’est pour l’instant implémenté, des CGV seront nécessaires lors de la mise en place de telles fonctionnalités.

Les CGU sont actuellement en cours de rédaction en interne.

1. **Mentions légales**

Les mentions légales sont accessibles depuis la page « À propos » de l’application.

1. **Cookies**

L’application n’utilise pas de système de cookie.

1. **Données personnelles**

Un hachage et une anonymisation des données sont effectués afin de garantir le respect des obligations en vigueur selon le RGPD. De plus, les données des utilisateurs ne s’étants pas connecté depuis 2 ans, 11 mois et 29 jours sont supprimés.

1. **Accessibilité**

Le site est accessible aux astigmates avec la possibilité de choisir entre un mode sombre et un mode clair de l’application.

Il est possible de zoomer dans l’application et il est possible d’utiliser la fonctionnalité « dualApps » implémentée par les derniers smartphones (possibilité de diviser l’écran en deux avec deux applications distinctes).

Un mode daltonien sera bientôt mis en place.

Le support est disponible en anglais comme en français.

L’utilisateur a la possibilité de choisir les unités de distances, de vitesse, de choisir le format de la date ainsi que son écriture en fonction de la région souhaitée.

# **XII – Enveloppe budgétaire**