ShareCookBook

# Introduction

## Description rapide

Ce projet vise à créer un site internet, plus précisément un réseau social. L’objectif de ce réseau est de créer et partager des recettes de cuisine avec d’autres utilisateurs.

## Plan d’action

En premier lieu, il est important de faire le cahier des charges, ensuite un plan d’action des tâches à faire sous forme de diagramme de Gantt, et pour finir, lancer l’exécution des différentes tâches afin de créer ce que contient le cahier des charges.

# Cahier des charges

## Introduction

### 2.1.1 Idée initiale

C’est lorsque je me suis rendu compte que ma grand-mère rédigeait tout le temps ses recettes à la main, pour nous les envoyer quand on en avait besoin, que me vint cette idée : créer une application dans laquelle toute la famille, les amis, et autres proches pourraient écrire leurs recettes, afin de pouvoir les partager et faciliter les échanges autour de cette transmission culinaire.

### 2.1.2 Public cible

Le public cible serait plutôt les personnes âgées, et les passionnés de cuisine. L’application ne s’adresse pas nécessairement à une personne lambda qui cherche une recette rapide pour son dîner. Ce serait plutôt une application de passionnés qui discutent, créent, et cuisinent les recettes qu'ils se partagent, entre les recettes de grand-mère, les créations et expérimentations de chacun et les recettes très connues, revisitées par chaque personne voulant rajouter sa petite touche.

### 2.1.3 Objectifs

Créer un endroit de partage de passions culinaires pour tous les utilisateurs, mais surtout être à l'écoute des utilisateurs qui font remonter des besoins concrets en terme de fonctionnalités de l’application.

### 2.1.4 Vision du produit

Deux utilisations seront possibles : premièrement les personnes aimant simplement la bonne cuisine, qui aimeraient partager leurs recettes à leurs amis / famille, et deuxièmement les personnes aimant expérimenter, car l'application leur permettra de trouver d'autres personnes partageant la même passion, cela leur permettra de s'entraider ou de trouver de l'inspiration.

## Description du projet

### 2.2.1 Fonctionnalités clés

Les fonctionnalités les plus importantes et à créer en priorité dans le cas d’une première version sont :

* Créer et publier des recettes
* Avoir un feed
* Pouvoir commenter et liker des recettes
* Pouvoir effectuer une recherche

D’autres fonctionnalités plus secondaires seront implantées dans les versions suivantes de l’application.

### 2.2.2 Plateformes cibles

Pour le moment, il est prévu que l’application soit disponible sur mobile, étant donné que la majorité des utilisateurs vont se servir de l'application avec leur téléphone, mais il est prévu d'avoir une application web accessible via n'importe quel appareil. Il faudra alors adapter l’UI pour desktop.

### 2.2.3 Monétisation

Mettre en place un système de publicités, obligatoirement, soit en rapport avec des magasins tels que Grand Frais ou Fresh, soit en rapport avec des ustensiles et autres éléments de cuisine.

### 2.2.4 Evolutivité

Une fois l'application sortie, les futures évolutions se baseront massivement sur l'avis des utilisateurs. Plus une fonctionnalité ou évolution sera demandée, plus elle sera en haut dans la liste des priorités. Bien sûr, l'équipe de développement peut et doit choisir les plus pertinentes.

## Exigences fonctionnelles

### 2.3.1 Gestion des utilisateurs

**Création de compte :**

Lors de la création de compte, l'utilisateur sera amené à choisir un User Name, qui devra être unique. Le code devra donc vérifier en base si cet User Name existe déjà ou non, de même pour l’adresse mail. L’utilisateur sera aussi invité à entrer un Nom, qui ne sera pas obligatoirement unique, ainsi qu’un mot de passe, avec la confirmation de ce dernier.

**Connexion/Déconnexion :**

L’utilisateur pourra alors se connecter grâce à son User Name et son mot de passe. Apparaîtra également une Checkbox lui proposant de rester connecté. Il aura aussi la possibilité de se déconnecter depuis la page Paramètres.

**Gestion de profil :**

L’utilisateur aura accès à une page profil où sera affiché son User Name, son Nom et son mot de passe dans une section. Les recettes qu’il a publiées seront accessibles dans une autre section.

### 2.3.2 Feed et recherche

**Feed :**

Une page dédiée au Feed sera disponible. Elle servira de page d'accueil. Sur cette page, l'utilisateur pourra trouver les dernières recettes postées les plus pertinentes.

**Recherche :**

Sur une page séparée, l'utilisateur pourra effectuer une recherche par mots clés, en tapant le titre de la recette ou l'intitulé du plat (exemple : taper « crêpes » donnera à l'utilisateur les recettes de crêpes bretonnes, crêpes classiques, crêpes salées, etc.)

### 2.3.3 Publications

L'utilisateur aura accès à une section "nouvelle recette" qui pourra lui servir à créer une nouvelle publication. La page sera composée d’un système d’onglets, servant à afficher à la fois la page de création, mais aussi une page de prévisualisation de la recette. Dans la page de création, il y aura une section « informations » où l’utilisateur pourra entrer le titre, le temps de préparation ainsi que le nombre de personnes pour lesquelles la recette est prévue. Dans la seconde section « ingrédients » où l’utilisateur pourra ajouter les ingrédients avec une seule et même commande. Ils apparaîtront en-dessous. Enfin une dernière section comportera un champ vide supportant du Markdown. Pour finir, figurera un bouton de validation et un autre pour réinitialiser la recette.

# 3. Etude de marché/concurrence

## 3.1 Marché

Mes observations du marché, effectuées sur une centaine de comptes Instagram, ainsi qu’une trentaine de comptes Twitter, révèlent que ces comptes postent au moins une fois tous les deux jours une recette de cuisine. La lecture de certains posts en réponse à ces publications montrent que ces réseaux ne sont pas adaptés.

## 3.2 Concurrence

### 3.2.1 Applications de recettes

Il existe de nombreuses applications de recettes, parmi lesquelles nous pouvons citer les applications des magasins de type Carrefour, qui proposent des recettes à réaliser à partir des produits disponibles dans leurs magasins, ou encore les applications de partage de recettes (750g, Marmiton, SuperCook ou encore Jow). Ces applications ne sont cependant pas du tout communautaires. Les recettes proposées sont uniquement écrites par l'éditeur et ne permettent pas une utilisation personnelle, il s’agit seulement d’un gros livre de recettes numériques.

D’autres applications de recettes de cuisine sont destinées aux personnes qui suivent un régime particulier. L'application proposera à l'utilisateur de choisir son régime alimentaire et lui donnera des recettes adaptées à ses besoins.

La plus-value de l’application que je créé est d’être uniquement communautaire (bien que l'éditeur ait la possibilité de mettre en avant certaines recettes les plus pertinentes). Son utilisation sera donc suffisamment spécifique pour ne pas entrer en concurrence directe avec les outils déjà présents sur le marché.

### 3.2.2 Réseaux sociaux

**Twitter :**

Coté réseaux sociaux, Twitter est pour l'instant le concurrent le plus connu dans l’édition de recettes. Il présente cependant de nombreux inconvénients dans la mesure où cet espace numérique n’est pas destiné au partage de recettes. Cet état de fait engendre différents désagréments :

Premièrement, Twitter n'est pas axé sur le partage de recettes, ce qui fait que bien souvent sur les posts d’influenceurs culinaires tels que [*Jigmé*](https://twitter.com/ClichesDeJigme), on peut lire de nombreux commentaires déplacés, des contenus haineux sans lien avec le post initial. Ce comportement rend le réseau social inadapté à cette pratique.

Deuxièmement, le partage de recettes sur Twitter est compliqué car les publications se font par des posts de vidéo ou de l'écrit. Ces modalités ne sont pas pratiques car elles impliquent un nombre de caractères limité par post et contraignent le partage de photos ou la mise en forme du texte.

**Threads :**

Threads présente tous les inconvénients de Twitter étant donné que c'est quasiment un copier-coller. Les contenus haineux sont toutefois moins présents.

**Instagram :**

Instagram est encore moins adapté à la publication de recettes.

Premièrement, les posts sont souvent des photos. Impossible donc de poster la recette écrite sans faire de montages photos qui vont forcer les utilisateurs à swiper et zoomer chaque fois que nécessaire. La longueur de la recette s’en trouve également limitée.

Ensuite, mettre le contenu de la recette en description de la photo n’est pas optimal. Comme sur Twitter et Threads, le texte ne pourra pas être mis en page et le nombre de caractères sera limité, mais sur Instagram, il est également impossible de rajouter un autre post en-dessous pour compléter.

# 4. Réponse au cahier des charges

## 4.1 UI et UX

Toutes les maquettes seront faites grâce à figma. Cela permettra de ne pas perdre de temps lors de la création de l'application côté code, étant donné que tout le design sera déjà créé.

### 4.1.1 UI

### **Style graphique :**

Pour cette application, il faudra un style graphique plutôt simple et très lisse avec quelques légers dégradés de couleurs pour mettre un peu de texture / profondeur, des boutons plutôt classiques rectangulaires avec des arrondis sur les bords, sans que les côtés soient des demi-cercles pour autant.

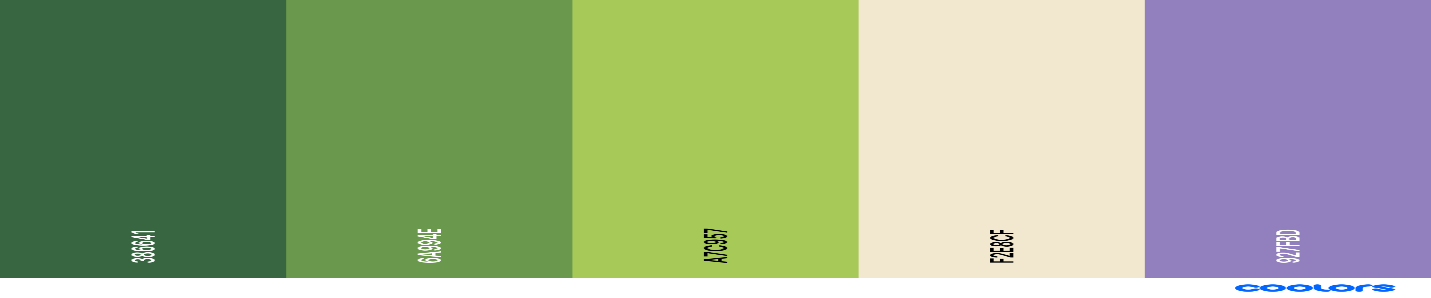
Il est inutile de mettre des effets de Hover étant donné que l'application sera pour le moment disponible uniquement sur mobile. Les effets de Hover des autres mouvements seront implémentés dans la version web, accessible sur Desktop.

Selon moi, il est aussi inutile de mettre des effets de mouvements sur les boutons. Il faut rester dans une interface lisse sans trop d'effets qui pourraient peut-être trop renforcer le côté application / futuriste. Le but de l’application est bien de rester au plus près d'un cahier de notes connecté à internet, plutôt que d’une application dans un téléphone.

### **Couleurs :**

En ce qui concerne les couleurs, j'opterai plutôt pour des couleurs style vert pâle et blanc cassé, pour amener un côté naturel qui renforce la volonté de créer une application sobre et moderne.

Les couleurs pourraient reprendre l'idée de l'application Seek (application servant à scanner et déterminer les espèces d'animaux, insectes et plantes à travers le monde et à les localiser). Les couleurs seraient celles-ci :



Le violet sera utilisé pour ajouter du contraste ou différencier deux boutons, plutôt que d’utiliser du rouge comme dans la plupart des applications.

### 4.1.2 UX

### **Bottom bar :**

Tout se passera dans une bottom bar qui comportera 5 boutons. De gauche à droite : le feed ou page d’accueil, la page de recherche, la création d’une nouvelle recette, le profil utilisateur et enfin les paramètres.

### **Liste des recettes :**

Dans le feed ou dans la page de recherche, les recettes seront présentées sous forme de liste. Chaque élément de la liste sera dans une boîte divisée en trois tiers : le premier à gauche sur fond violet comportera le nombre de likes, le temps de préparation et le nombre de personnes, et dans les deux tiers restants sur fond vert, le titre de la recette. La transition entre les deux parties est marquée par un dégradé du violet au vert.

### **Recette :**

Lors de l'ouverture d'une recette, la page sera composée :

- d'un Header contenant le titre de la recette suivi, dans trois sections différentes, du temps de préparation, du nombre de personnes et du nombre de likes,

- du corps de la page, où sera affiché le texte de la recette, grâce à un interpréteur, transformant le Markdown en HTML,

- puis tout en bas de page, un Footer contenant, si on le divise en 5 : deux parties affichant l’User Name de la personne ayant posté la recette, un bouton de like, un bouton permettant d’afficher et de poster des commentaires, et enfin un bouton, pour l’instant non fonctionnel, servant à enregistrer la recette dans son profil.

## 4.2 Monétisation

Je travaille actuellement chez Prosol, exploitant des magasins Grand Frais et Fresh. J’ai entamé des discussions sur un possible financement contre l’affichage dans l’application de publicités pour les magasins. Pour le moment, l’équipe chargée du marketing et des publicités étudie la proposition. Des discussions concernant les objectifs de trafic et le financement sont envisagées.

## 4.3 Architecture

### 4.3.1 Frontend

Pour le frontend, le large choix m'a fait me poser beaucoup de questions entre les Framework tels que react, angular, vue... et les langages tels que PHP offrant un choix varié.

Mon choix s'est tourné vers les Framework.

Premièrement, je suis plus à l’aise dans l’utilisation des packages Node dont j’ai l'habitude de me servir plutôt que ceux de composer de PHP.

Deuxièmement, bien que Angular était un choix pertinent, je préférais me servir de React pour renforcer mes connaissances. J'ai donc utilisé ce Framework dans sa configuration TSX .

En ce qui concerne les packages installés, j'ai choisi d'utiliser les icônes de Bootstrap que j'ai utilisés dans le package "react-icons". Il intègre les icônes directement dans des composants. "React-markdown" me permet d'avoir un composant prenant en paramètre une variable contenant un string, le transforme en markdown afin d'afficher les recettes correctement, les packages classiques tels que react-dom, react-router-dom, etc...

### 4.3.2 Backend

Le choix pour le backend a été beaucoup plus rapide, étant donné que je n'en avais encore jamais fait. Il n’était pas nécessaire de faire beaucoup de traitement, le backend servant uniquement à renvoyer les données de la base. Aussi mon choix s'est porté sur l’utilisation de Node, écrit en JS.

De ce fait, j'ai utilisé des packages Node tels que :

- Express pour la connexion à la base de données

- Cors pour cloisonner l'accès (aux requêtes, et donc à la base de données) uniquement à l'adresse IP du VPS

- ainsi que Winston pour me créer un fichier de logs afin de savoir ce qui s'est passé en cas de bugs et autres problèmes remontés par les utilisateurs (même si le back ne considère pas ça comme des erreurs).

### 4.3.3 Base de données

Le choix de la base de données était initialement porté sur MySQL, car c'était le seul SGBD que j'avais déjà utilisé. Cependant, après avoir manipulé PostgreSQL dans le cadre professionnel, et ma base de données n'étant pas encore très imposante, j'ai rapidement changé de SGBD afin de passer sur PostgreSQL. Cela me permettait de m'en servir plus facilement dans le cadre professionnel, mais aussi de renforcer son utilisation dans le reste des projets que j'entreprendrai (tout comme le backend en JS, qui après utilisation, correspond totalement à ce dont j'ai besoin).

### 4.3.4 Hébergement

Pour l'Hébergement, j'ai choisi Hostinger. Il va me permettre non seulement de gérer à la fois le nom de domaine et le VPS, mais aussi de créer un sous-domaine dédié à mon application. J'avais la possibilité d’utiliser d'autres solutions, mais Hostinger m'a semblé être un choix judicieux : je savais qu'en cas de soucis je pourrais avoir l'aide d’un ami qui m'a beaucoup parlé de cette solution étant donné qu'il l’utilise personnellement (je n’ai finalement eu besoin d’aucune aide !).

# 5. Développement

## 5.1 Frontend

Le frontend se compose, excepté pour les écrans de connexion et de création de compte :

- d'un header dont le contenu diffère en fonction de l'écran, qui accueille notamment le titre de l'écran sur lequel nous nous trouvons, et un bouton retour si nécessaire

- du corps de l'écran, qui dépend a 100% de l'écran sélectionné mais qui respecte quand même le style de base de l'application

- d’une bottom bar contenant 5 boutons, servant à naviguer entre les écrans les plus importants tels que :

* L'écran d'accueil : contenant la prévisualisation des dernières publications postées de la plus récente à la plus ancienne
* L'écran de recherche : contenant une barre de recherche, avec en-dessous les résultats, ou un message si le résultat de la recherche est vide
* L'écran de création : composé de deux onglets, le premier sert à entrer toutes les informations relatives à la nouvelle recette que l'utilisateur désire poster, et le deuxième onglet qui sert à prévisualiser sa recette, reprenant exactement le même style que l'écran affiché lors du clic sur une prévisualisation de recette
* L'écran de compte : affiche en premier lieu les informations relatives au compte sur lequel nous sommes connectés, tels que le nom d'utilisateur, l'adresse mail ainsi que le nom complet, et en second lieu la prévisualisation des recettes que l'utilisateur a postées dans l'ordre de la plus récente à la plus ancienne
* L'écran de paramètres : ne servant pour l'instant qu'à avoir accès à un bouton "Déconnexion"

La bottom bar grise légèrement le bouton correspondant à l'écran sur lequel nous sommes, excepté pour deux écrans. Les seuls deux écrans qui ne sont pas accessibles depuis la bottom bar sont :

* L'écran de recette : affiché après le clic sur une prévisualisation de recette, l'écran est composé d'un bloc d'informations au-dessus, contenant les informations : titre, temps de préparation, nombre de personnes ainsi que le nombre de likes, suivi d'un espace où la recette est renseignée, d'abord une liste d'ingrédients, ensuite le corps de la recette, et juste au-dessus de la bottom bar, un autre bloc contenant le nom d'utilisateur de la personne ayant posté la recette, et trois boutons d'action : liker, commenter et enregistrer (malheureusement le bouton enregistrer ne fonctionne pas encore)
* L'écran de commentaires : accessible depuis l'écran de recette, il affiche tous les commentaires les uns à la suite des autres, le plus récent en bas, à la manière d'un chat, les commentaires contiennent le nom d'utilisateur de la personne ayant commenté, ainsi que le contenu du commentaire

## 5.2 Backend

Le backend se compose de plusieurs fichiers :

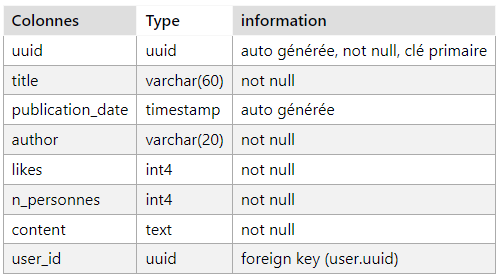
* index.js, contenant tous les chemins d'appel et faisant la liaison avec les deux fichiers de méthodes
* PublicationQuerries.js, contenant toutes les requêtes relatives aux publications tels que les likes, le post, etc.
* UserQuerries.js, contenant toutes les requêtes relatives aux utilisateurs telles que connexion, création de compte, accès aux informations
* Logger.js, contenant les méthodes pour mettre des informations et erreurs dans le fichier de log, avec la date, le nom de la méthode, et si elle a réussi, ou si elle a levé une erreur avec le contenu de l'erreur dans ce cas
* DateUtils.js : servant simplement à récupérer la date actuelle ainsi que traiter la donnée reçue pour lui donner ce format : dd-MM-yyyy HH:mm:ss

## 5.3 Base de données

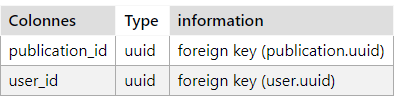
La base de données comporte actuellement quatre tables :

* publications
* users
* comments
* publication\_likes

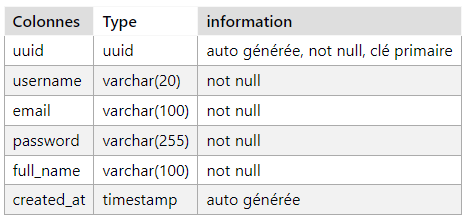
Publications :



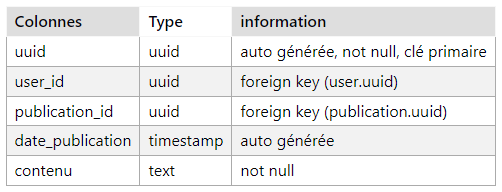
Publication\_likes :



Accounts :



commentaires :



## 5.4 Hébergement

Une fois mon compte Hostinger créé et l'achat d'un VPS effectué (j'ai pris le moins cher, n’ayant pas encore besoin de beaucoup de ressources, le premier plan me suffisait amplement), j'ai alors choisi le système d'exploitation Ubuntu 23.04 64bit sans interface graphique. Etant donné que mon VPS n'a pas beaucoup de ressources, ça ne servait à rien d'avoir une interface.

Alors, j'ai ajouté les fichiers de mon application, backend et frontend, dans le dossier root afin de lancer les deux. J'ai utilisé screens afin de gérer les deux applications en même temps.

En ce qui concerne le backend, je l'ai simplement lancé comme je le faisais sur ma machine, en faisant node index.js, puis je l'ai laissé tourner dans mon screen.

Pour le frontend, j'ai build mon application, et envoyé les fichiers du build dans le dossier /var/www/html.

En ce qui concerne le nom de domaine, j'ai acheté ebasson.fr depuis l'interface de Hostinger, avant de créer un sous domaine pointant sur l'adresse IP du VPS que je venais d'installer.

Après avoir lié mon nom de domaine à mon VPS, j'ai suivi un tutoriel pour installer et utiliser correctement Nginx afin qu'il redirige tout trafic http du port 80 vers les fichiers de mon application, dans le dossier /var/www/html

J'ai ensuite automatisé grâce à une fonction .bashrc afin de récupérer automatiquement le dernier commit, et refaire toute la séquence afin de mettre à jour l'application, ce qui me permet de faire des mises à jour et correctifs très rapidement après les retours des premiers utilisateurs.

En ce qui concerne la base de données, j'ai installé PostgreSQL sur la machine et configuré correctement le fichier pg\_hba.conf pour autoriser mon adresse IP personnelle à accéder aux bases, afin de me connecter avec dbeaver, rendant les évolutions sur la base de données aussi simplement que la mise à jour de l'application de production.